SMART ARM-based Microcontrollers

SAM L21E / SAM L21G / SAM L21J

DATASHEET COMPLETE

Introduction

Atmel® | SMART SAM L21 is a series of Ultra low-power microcontrollers using the 32-bit ARM® Cortex®-M0+ processor at max. 48MHz (2.46 CoreMark®/MHz) and up to 256KB Flash and 40KB of SRAM in a 32, 48, and 64 pin package. The sophisticated power management technologies, such as power domain gating, SleepWalking, Ultra low-power peripherals and more, allow for very low current consumptions. The highly configurable peripherals include a touch controller supporting capacitive interfaces with proximity sensing.

Features

- Processor
  - ARM Cortex-M0+ CPU running at up to 48MHz
    - Single-cycle hardware multiplier
    - Micro Trace Buffer
- Memories
  - 32/64/128/256KB in-system self-programmable Flash
  - 1/2/4/8KB Flash Read-While-Write section
  - 4/8/16/32KB SRAM Main Memory
  - 2/4/8/8KB SRAM Low power Memory
- System
  - Power-on reset (POR) and brown-out detection (BOD)
  - Internal and external clock options
  - External Interrupt Controller (EIC)
  - 16 external interrupts
  - One non-maskable interrupt
  - Two-pin Serial Wire Debug (SWD) programming, test and debugging interface
- Low Power
  - Idle, Standby, Backup, and Off sleep modes
  - SleepWalking peripherals

  - Static and Dynamic Power Gating Architecture
  - Battery backup support
  - Two Performance Levels
  - Embedded Buck/LDO regulator supporting on-the-fly selection
- Peripherals
  - 16-channel Direct Memory Access Controller (DMAC)
  - 12-channel Event System
  - Up to five 16-bit Timer/Counters (TC) including one low-power TC, each configurable as:
    - 16-bit TC with two compare/capture channels
    - 8-bit TC with two compare/capture channels
    - 32-bit TC with two compare/capture channels, by using two TCs
  - Two 24-bit and one 16-bit Timer/Counters for Control (TCC), with extended functions:
    - Up to four compare channels with optional complementary output
    - Generation of synchronized pulse width modulation (PWM) pattern across port pins
    - Deterministic fault protection, fast decay and configurable dead-time between complementary output
    - Dithering that increase resolution with up to 5 bit and reduce quantization error
  - 32-bit Real Time Counter (RTC) with clock/calendar function
  - Watchdog Timer (WDT)
  - CRC-32 generator
  - One full-speed (12Mbps) Universal Serial Bus (USB) 2.0 interface
    - Embedded host and device function
    - Eight endpoints
  - Up to six Serial Communication Interfaces (SERCOM) including one low-power SERCOM, each configurable to operate as either:
    - USART with full-duplex and single-wire half-duplex configuration
    - I^2C up to 3.4MHz
    - SPI
    - LIN slave
  - One AES encryption engine
  - One True Random Generator (TRNG)
  - One Configurable Custom Logic (CCL)
  - One 12-bit, 1MSPS Analog-to-Digital Converter (ADC) with up to 20 channels
    - Differential and single-ended input
    - Automatic offset and gain error compensation
    - Oversampling and decimation in hardware to support 13-, 14-, 15-, or 16-bit resolution
  - Two 12-bit, 1MSPS Dual Output Digital-to-Analog Converter (DAC)
  - Two Analog Comparators (AC) with window compare function
  - Three Operational Amplifiers (OPAMP)
  - Peripheral Touch Controller (PTC)
    - 169-Channel capacitive touch and proximity sensing
- Oscillators
  - 32.768kHz crystal oscillator (XOSC32K)
  - 0.4-32MHz crystal oscillator (XOSC)

  - 32.768kHz internal oscillator (OSC32K)
  - 32.768kHz ultra-low-power internal oscillator (OSCULP32K)
  - 16/12/8/4MHz high-accuracy internal oscillator (OSC16M)
  - 48MHz Digital Frequency Locked Loop (DFLL48M)
  - 96MHz Fractional Digital Phased Locked Loop (FDPLL96M)
- I/O
  - Up to 51 programmable I/O pins
- Easy migration from SAM D family
- Packages
  - 64-pin TQFP, QFN, WLCSP
  - 48-pin TQFP, QFN
  - 32-pin TQFP, QFN
- Operating Voltage
  - 1.62V – 3.63V

Table of Contents

1. Description

Atmel® | SMART SAM L21 is a series of Ultra low-power microcontrollers using the 32-bit ARM® Cortex®-M0+ processor, and ranging from 32- to 64-pins with up to 256KB Flash and 40KB of SRAM. The SAM L21 devices operate at a maximum frequency of 48MHz and reach 2.46 CoreMark®/MHz. They are designed for simple and intuitive migration with identical peripheral modules, hex compatible code, identical linear address map and pin compatible migration paths between all devices in the product series. All devices include intelligent and flexible peripherals, Atmel Event System for inter-peripheral signaling, and support for capacitive touch button, slider and wheel user interfaces.

The Atmel SAM L21 devices provide the following features: In-system programmable Flash, 16-channel direct memory access (DMA) controller, 12-channel Event System, programmable interrupt controller, up to 51 programmable I/O pins, 32-bit real-time clock and calendar, up to five 16-bit Timer/Counters (TC) and three Timer/Counters for Control (TCC) where each TC/TCC can be configured to perform frequency and waveform generation, accurate program execution timing or input capture with time and frequency measurement of digital signals. The TCs can operate in 8- or 16-bit mode, selected TCs can be cascaded to form a 32-bit TC, and three timer/counters have extended functions optimized for motor, lighting and other control applications. Two TCC can operate in 24-bit mode, the third TCC can operate in 16-bit mode. The series provide one full-speed USB 2.0 embedded host and device interface; up to six Serial Communication Modules (SERCOM) that each can be configured to act as an USART, UART, SPI, I^2C up to 3.4MHz, SMBus, PMBus, and LIN slave; up to twenty channel 1MSPS 12-bit ADC with programmable gain and optional oversampling and decimation supporting up to 16-bit resolution, two 12-bit 1MSPS DACs, two analog comparators with window mode, three independent cascadable OPAMPs supporting internal connection with others analog features, Peripheral Touch Controller supporting up to 192 buttons, sliders, wheels and proximity sensing; programmable Watchdog Timer, brown-out detector and power-on reset and two-pin Serial Wire Debug (SWD) program and debug interface.

All devices have accurate and low-power external and internal oscillators. All oscillators can be used as a source for the system clock. Different clock domains can be independently configured to run at different frequencies, enabling power saving by running each peripheral at its optimal clock frequency, and thus maintaining a high CPU frequency while reducing power consumption.

The SAM L21 devices have four software-selectable sleep modes, idle, standby, backup and off. In idle mode the CPU is stopped while all other functions can be kept running. In standby all clocks and functions are stopped except those selected to continue running. In this mode all RAMs and logic contents are retained. The device supports SleepWalking. This feature allows the peripheral to wake up from sleep based on predefined conditions, and thus allows some internal operation like DMA transfer and/or the CPU to wake up only when needed, e.g. when a threshold is crossed or a result is ready. The Event System supports synchronous and asynchronous events, allowing peripherals to receive, react to and send events even in standby mode.

The SAM L21 devices have two software-selectable performance levels (PL0 and PL2) allowing the user to scale the lowest core voltage level that will support the operating frequency. To further minimize consumption, specifically leakage dissipation, the SAM L21 devices utilizes power domain gating technique with retention to turn off some logic area while keeping its logic state. This technique is fully handled by hardware.

The Flash program memory can be reprogrammed in-system through the SWD interface. The same interface can be used for nonintrusive on-chip debugging of application code. A boot loader running in the device can use any communication interface to download and upgrade the application program in the Flash memory.

The Atmel SAM L21 devices are supported with a full suite of programs and system development tools, including C compilers, macro assemblers, program debugger/simulators, programmers and evaluation kits.

2. Configuration Summary

	SAM L21J	SAM L21G	SAM L21E
Pins	64	48	32
General Purpose I/O-pins (GPIOs)	51	37	25
Flash	256/128/64KB	256/128/64KB	256/128/64/32KB
Flash RWW section	8/4/2KB	8/4/2KB	8/4/2/1KB
System SRAM	32/16/8KB	32/16/8KB	32/16/8/4KB
Low Power SRAM	8/8/4KB	8/8/4KB	8/8/4/2KB
Timer Counter (TC) instances(1)	5	3	3
Waveform output channels per TC instance	2	2	2
Timer Counter for Control (TCC) instances	3	3	3
Waveform output channels per TCC	8/4/2	8/4/2	6/4/2
DMA channels	16	16	16
USB interface	1	1	1
AES engine	1	1	1
Configurable Custom Logic (CCL) (LUTs)	4	4	4
True Random Generator (TRNG)	1	1	1
Serial Communication Interface (SERCOM) instances	6	6	6
Analog-to-Digital Converter (ADC) channels	20	14	10
Analog Comparators (AC)	2	2	2
Digital-to-Analog Converter (DAC) channels	2	2	2
Operational Amplifier (OPAMP)	3	3	3

	SAM L21J	SAM L21G	SAM L21E
Real-Time Counter (RTC)	Yes	Yes	Yes
RTC alarms	1	1	1
RTC compare values	One 32-bit value or two 16-bit values	One 32-bit value or two 16-bit values	One 32-bit value or two 16-bit values
External Interrupt lines	16	16	16
Peripheral Touch Controller (PTC) channels (X- x Y-Lines) for mutual capacitance (2)	169 (13x13)	81 (9x9)	42 (7x6)
Peripheral Touch Controller (PTC) channels for self capacitance (Y-Lines only) (3)	16	10	7
Maximum CPU frequency	48MHz		
Packages	QFN TQFP WLCSP	QFN TQFP	QFN TQFP
Oscillators	32.768kHz crystal oscillator (XOSC32K) 0.4-32MHz crystal oscillator (XOSC) 32.768kHz internal oscillator (OSC32K) 32KHz ultra-low-power internal oscillator (OSCULP32K) 16/12/8/4MHz high-accuracy internal oscillator (OSC16M) 48MHz Digital Frequency Locked Loop (DFLL48M) 96MHz Fractional Digital Phased Locked Loop (FDPLL96M)		
Event System channels	12	12	12
SW Debug Interface	Yes	Yes	Yes
Watchdog Timer (WDT)	Yes	Yes	Yes

Note:

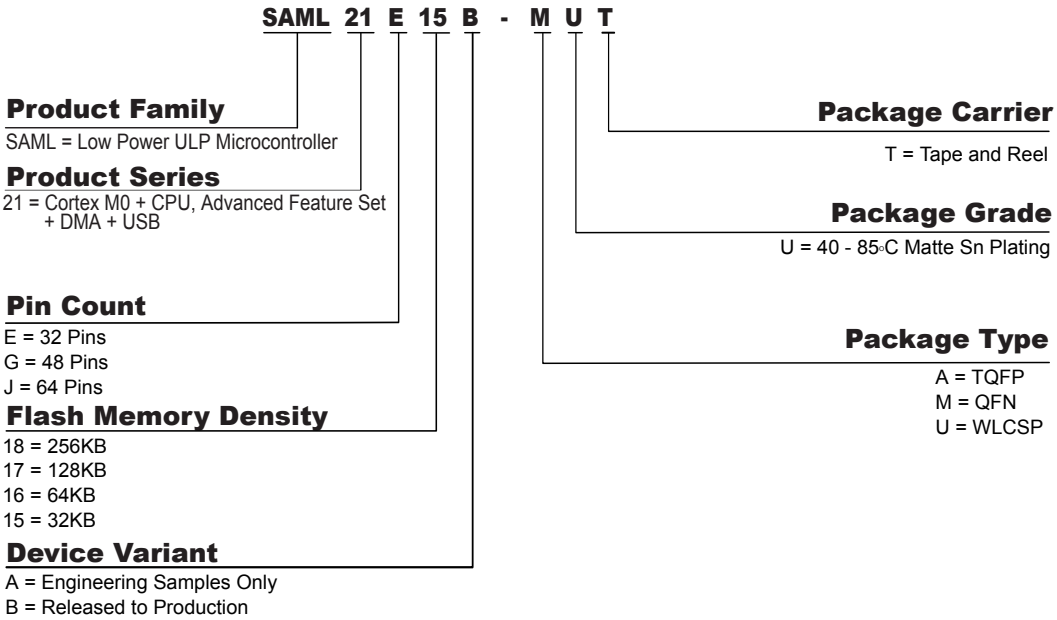
1. For SAM L21E and SAM L21G, only TC0, TC1 and TC4 are available.
2. The number of X- and Y-lines depends on the configuration of the device, as some I/O lines can be configured as either X-lines or Y-lines. Refer to *Multiplexed Signals* for details. The number in the Configuration Summary is the maximum number of channels that can be obtained.

3. The number of Y-lines depends on the configuration of the device, as some I/O lines can be configured as either X-lines or Y-lines. The number given here is the maximum number of Y-lines that can be obtained.

Related Links

Multiplexed Signals on page 29

3. Ordering Information



3.1. SAM L21J

Table 3-1. SAM L21J Ordering Codes

Ordering Code	FLASH (bytes)	SRAM (bytes)	Package	Carrier Type
ATSAML21J16B-AUT	64K	8K	TQFP64	Tape & Reel
ATSAML21J16B-MUT			QFN64	
ATSAML21J17B-AUT	128K	16K	TQFP64	Tape & Reel
ATSAML21J17B-MUT			QFN64	
ATSAML21J18B-AUT	256K	32K	TQFP64	Tape & Reel
ATSAML21J18B-MUT			QFN64	
ATSAML21J18B-UUT			WLCSP64	

3.2. SAM L21G

Table 3-2. SAM L21G Ordering Codes

Ordering Code	FLASH (bytes)	SRAM (bytes)	Package	Carrier Type
ATSAML21G16B-AUT	64K	8K	TQFP48	Tape & Reel
ATSAML21G16B-MUT			QFN48	

Ordering Code	FLASH (bytes)	SRAM (bytes)	Package	Carrier Type
ATSAML21G17B-AUT	128K	16K	TQFP48	Tape & Reel
ATSAML21G17B-MUT			QFN48	
ATSAML21G18B-AUT	256K	32K	TQFP48	Tape & Reel
ATSAML21G18B-MUT			QFN48	

3.3. SAM L21E

Table 3-3. SAM L21E

Ordering Code	FLASH (bytes)	SRAM (bytes)	Package	Carrier Type
ATSAML21E15B-AUT	32K	4K	TQFP32	Tape & Reel
ATSAML21E15B-MUT			QFN32	
ATSAML21E16B-AUT	64K	8K	TQFP32	Tape & Reel
ATSAML21E16B-MUT			QFN32	
ATSAML21E17B-AUT	128K	16K	TQFP32	Tape & Reel
ATSAML21E17B-MUT			QFN32	
ATSAML21E18B-AUT	256K	32K	TQFP32	Tape & Reel
ATSAML21E18B-MUT			QFN32	

3.4. Device Identification

The DSU - Device Service Unit peripheral provides the Device Selection bits in the Device Identification register (DID.DSEL) in order to identify the device by software. The SAM L21 variants have a reset value of DID=0x1081drxx, with the LSB identifying the die number ('d'), the die revision ('r') and the device selection ('xx').

Table 3-4. SAM L21 Device Identification Values

DSEL (DID[7:0])	Device
0x00	SAML21J18A
0x01	SAML21J17A
0x02	SAML21J16A
0x03-0x04	Reserved
0x05	SAML21G18A
0x06	SAML21G17A
0x07	SAML21G16A
0x08-0x09	Reserved
0x0A	SAML21E18A

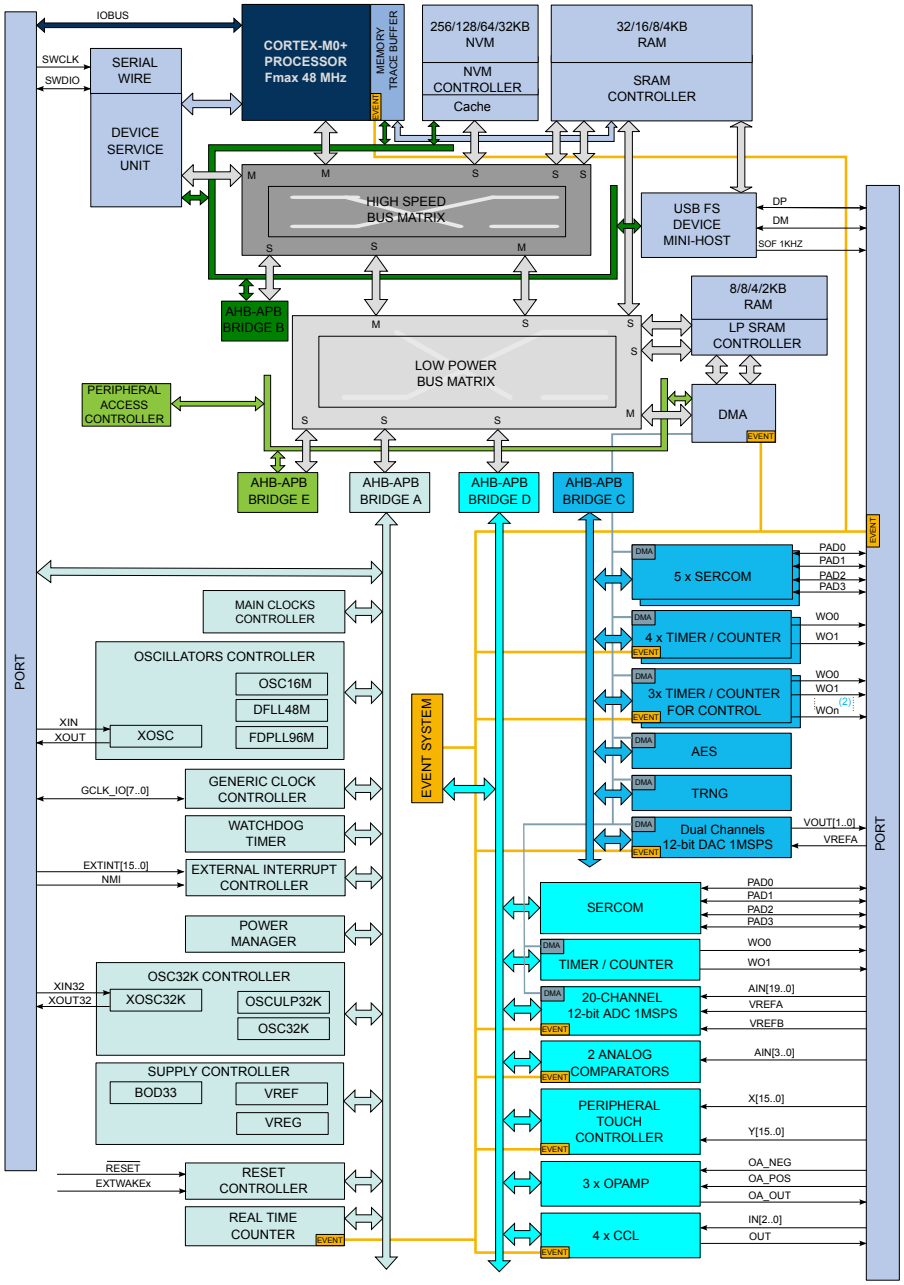
DSEL (DID[7:0])	Device
0x0B	SAML21E17A
0x0C	SAML21E16A
0x0D	SAML21E15A
0x0E	Reserved
0x0F	SAML21J18B
0x10	SAML21J17B
0x11	SAML21J16B
0x12-0x13	Reserved
0x14	SAML21G18B
0x15	SAML21G17B
0x16	SAML21G16B
0x17-0x18	Reserved
0x19	SAML21E18B
0x1A	SAML21E17B
0x1B	SAML21E16B
0x1C	SAML21E15B
0x1D-0xFF	Reserved

Note: The device variant (last letter of the ordering number) is independent of the die revision. The device variant denotes functional differences, whereas the die revision marks evolution of the die.

Related Links

DSU - Device Service Unit on page 87
DID on page 111

4. Block Diagram



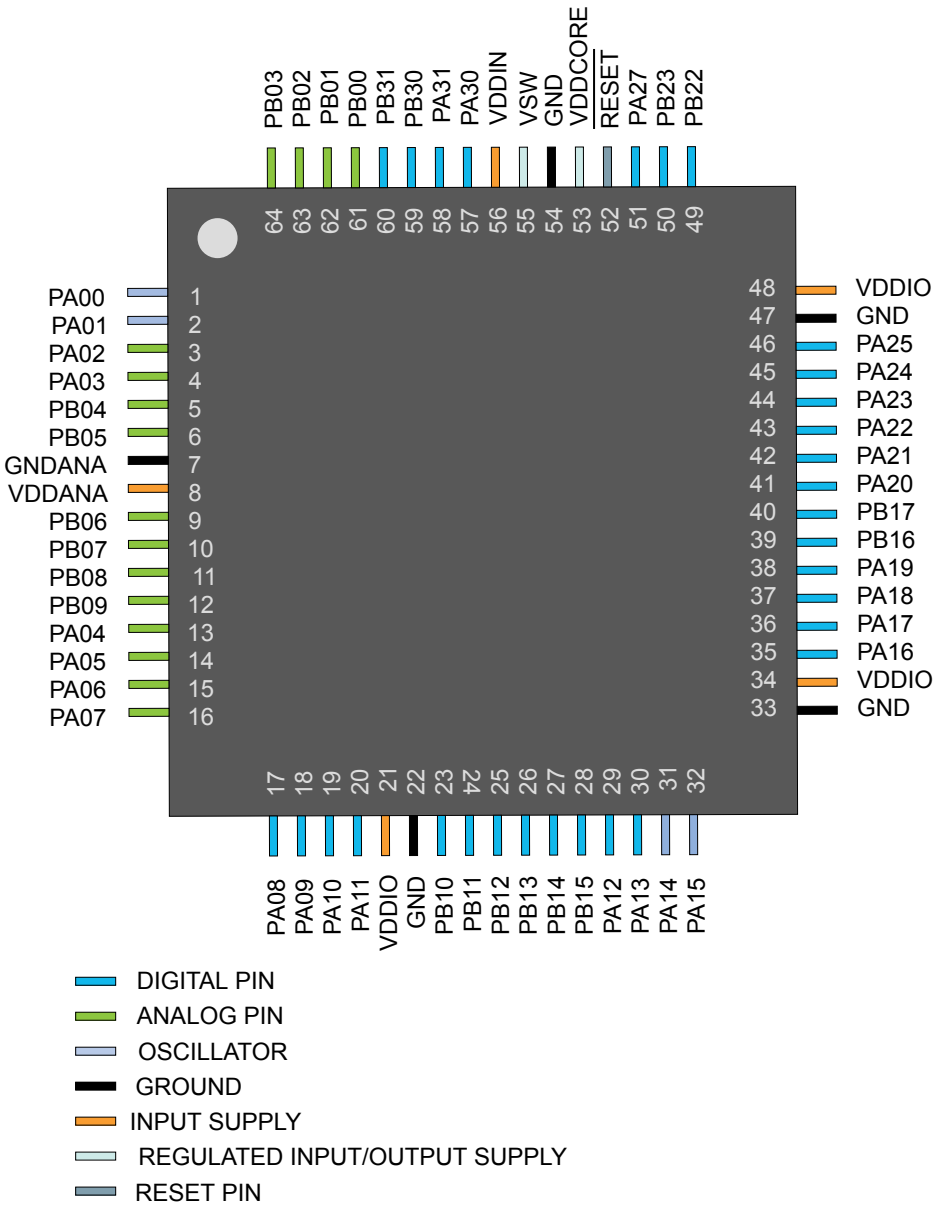
Note:

1. Some products have different number of SERCOM instances, Timer/Counter instances, PTC signals and ADC signals. Refer to Peripherals Configuration Summary on page 84 for details.

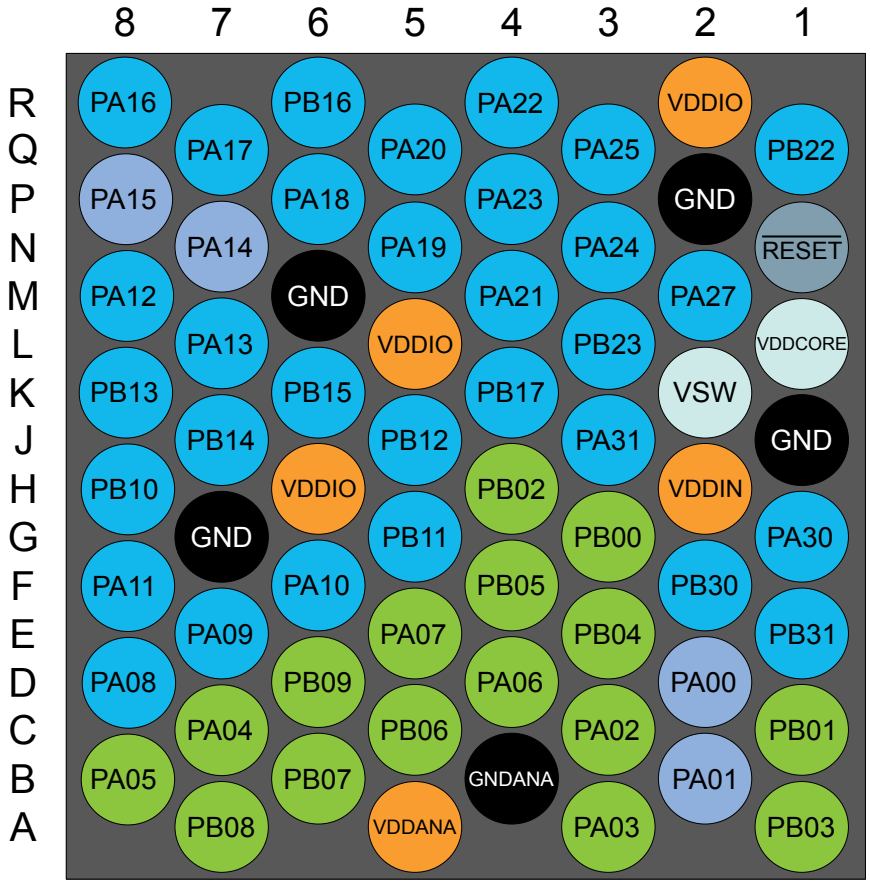
2. The three TCC instances have different configurations, including the number of Waveform Output (WO) lines.

5. Pinout

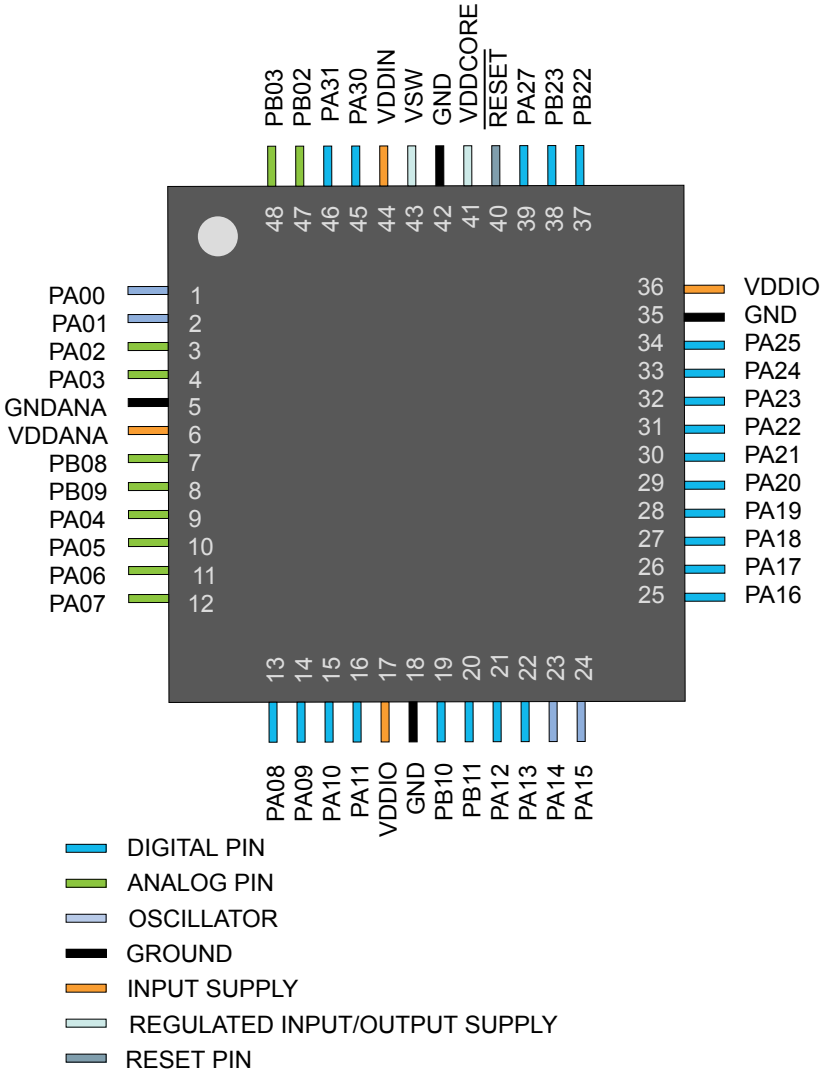
5.1. SAM L21J



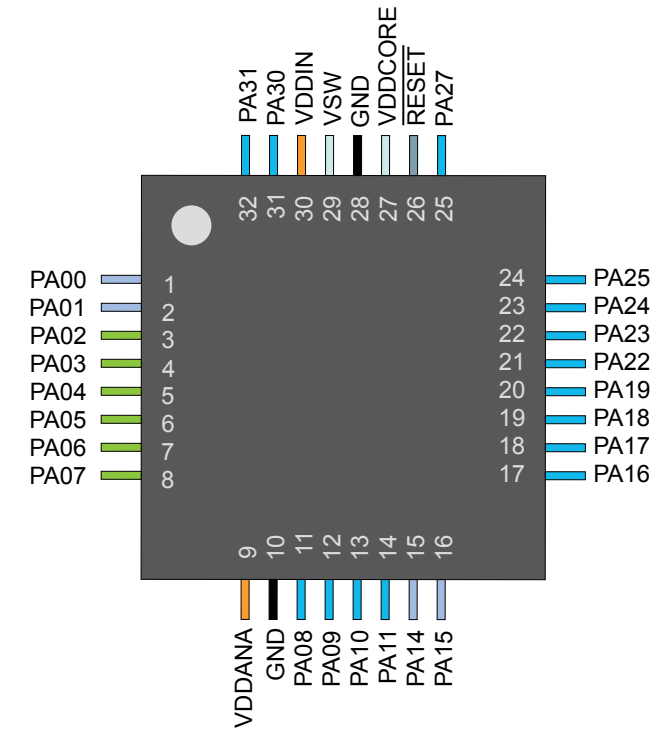
5.2. SAM L21J WLCSP64



5.3. SAM L21G


PB03
PB02
PA31
PA30
VDDIN
VSW
GND
VDDCORE
RESET
PA27
PB23
PB22
48
47
46
45
44
43
42
41
40
39
38
37
PA00 1
PA01 2
PA02 3
PA03 4
GNDANA 5
VDDANA 6
PB08 7
PB09 8
PA04 9
PA05 10
PA06 11
PA07 12
36 VDDIO
35 GND
34 PA25
33 PA24
32 PA23
31 PA22
30 PA21
29 PA20
28 PA19
27 PA18
26 PA17
25 PA16
13
14
15
16
17
18
19
20
21
22
23
24
PA08
PA09
PA10
PA11
VDDIO
GND
PB10
PB11
PA12
PA13
PA14
PA15
DIGITAL PIN
ANALOG PIN
OSCILLATOR
GROUND
INPUT SUPPLY
REGULATED INPUT/OUTPUT SUPPLY
RESET PIN

5.4. SAM L21E



- DIGITAL PIN
- ANALOG PIN
- OSCILLATOR
- GROUND
- INPUT SUPPLY
- REGULATED INPUT/OUTPUT SUPPLY
- RESET PIN

6. Signal Descriptions List

The following table gives details on signal names classified by peripheral.

Table 6-1. Signal Descriptions List

Signal Name	Function	Type	Active Level
Analog Comparators - AC			
AIN[3:0]	AC Analog Inputs	Analog	
CMP[1:0]	AC Comparator Outputs	Digital	
Analog Digital Converter - ADC			
AIN[19:0]	ADC Analog Inputs	Analog	
VREFA	ADC Voltage External Reference A	Analog	
VREFB	ADC Voltage External Reference B	Analog	
Digital Analog Converter - DAC			
VOUT[1:0]	DAC Voltage output	Analog	
VREFA	DAC Voltage External Reference	Analog	
Operational Amplifier - OPAMP			
OANEG[2:0]	OPAMP Analog Negative Inputs	Analog	
OAPOS[2:0]	OPAMP Analog Positive Inputs	Analog	
OAOUT[2:0]	OPAMP Analog outputs	Analog	
External Interrupt Controller - EIC			
EXTINT[15:0]	External Interrupts inputs	Digital	
NMI	External Non-Maskable Interrupt input	Digital	
Reset Controller - RSTC			
EXTWAKE[7:0]	External wake-up inputs	Digital	
Generic Clock Generator - GCLK			
GCLK_IO[7:0]	Generic Clock (source clock inputs or generic clock generator output)	Digital	
Custom Control Logic - CCL			
IN[11:0]	Logic Inputs	Digital	
OUT[3:0]	Logic Outputs	Digital	
Supply Controller - SUPC			
VBAT	External battery supply Inputs	Analog	

Signal Name	Function	Type	Active Level
PSOK	Main Power Supply OK input	Digital	
OUT[1:0]	Logic Outputs	Digital	
Power Manager - PM			
RESETN	Reset input	Digital	Low
Serial Communication Interface - SERCOMx			
PAD[3:0]	SERCOM Inputs/Outputs Pads	Digital	
Oscillators Control - OSCCTRL			
XIN	Crystal or external clock Input	Analog/Digital	
XOUT	Crystal Output	Analog	
32KHz Oscillators Control - OSC32KCTRL			
XIN32	32KHz Crystal or external clock Input	Analog/Digital	
XOUT32	32KHz Crystal Output	Analog	
Timer Counter - TCx			
WO[1:0]	Waveform Outputs	Digital	
Timer Counter - TCCx			
WO[7:0]	Waveform Outputs	Digital	
Peripheral Touch Controller - PTC			
X[15:0]	PTC Input	Analog	
Y[15:0]	PTC Input	Analog	
General Purpose I/O - PORT			
PA25 - PA00	Parallel I/O Controller I/O Port A	Digital	
PA27	Parallel I/O Controller I/O Port A	Digital	
PA31 - PA30	Parallel I/O Controller I/O Port A	Digital	
PB17 - PB00	Parallel I/O Controller I/O Port B	Digital	
PB23 - PB22	Parallel I/O Controller I/O Port B	Digital	
PB31 - PB30	Parallel I/O Controller I/O Port B	Digital	
Universal Serial Bus - USB			
DP	DP for USB	Digital	
DM	DM for USB	Digital	
SOF 1kHz	USB Start of Frame	Digital	

7. I/O Multiplexing and Considerations

7.1. Multiplexed Signals

Each pin is by default controlled by the PORT as a general purpose I/O and alternatively it can be assigned to one of the peripheral functions A, B, C, D, E, F, G, H or I. To enable a peripheral function on a pin, the Peripheral Multiplexer Enable bit in the Pin Configuration register corresponding to that pin (PINCFGn.PMUXEN, n = 0..31) in the PORT must be written to '1'. The selection of peripheral function A to H is done by writing to the Peripheral Multiplexing Odd and Even bits in the Peripheral Multiplexing register (PMUXn.PMUXE/O) in the PORT.

This table describes the peripheral signals multiplexed to the PORT I/O pins.

Table 7-1. PORT Function Multiplexing

Pin			I/O pin	Supply	A	B(1)(2)						C	D	E	F	G	H	I
SAML21E	SAML21G	SAML21J			EIC/RSTC	REF	ADC	AC	PTC	DAC	OPAMP	SERCOM (1)(2)	SERCOM-ALT	TC/TCC (3)	TCC	COM	AC/GCLK/SUPC	CCL
1	1	1	PA00	VSWOUT	EXTINT[0]/ EXTWAKE[0]								SERCOM1/ PAD[0]	TCC2/ WO[0]				
2	2	2	PA01	VSWOUT	EXTINT[1]/ EXTWAKE[1]								SERCOM1/ PAD[1]	TCC2/ WO[1]				
3	3	3	PA02 (5)	VDDANA	EXTINT[2]/ EXTWAKE[2]		AIN[0]		Y[0]	VOUT[0]	OA_NEG[0]							
4	4	4	PA03	VDDANA	EXTINT[3]/ EXTWAKE[3]	VREFA	AIN[1]		Y[1]									
		5	PB04	VDDANA	EXTINT[4]		AIN[12]		Y[10]									
		6	PB05	VDDANA	EXTINT[5]		AIN[13]		Y[11]		OA_NEG[1]							
		9	PB06	VDDANA	EXTINT[6]		AIN[14]		Y[12]		OA_NEG[2]							CCL2 IN[0]
		10	PB07	VDDANA	EXTINT[7]		AIN[15]		Y[13]									CCL2 IN[1]
	7	11	PB08	VDDANA	EXTINT[8]		AIN[2]				OA_OUT[1]		SERCOM4/ PAD[0]	TC0/ WO[0]				CCL2 IN[2]
	8	12	PB09	VDDANA	EXTINT[9]		AIN[3]		Y[15]		OA_POS[1]		SERCOM4/ PAD[1]	TC0/ WO[1]				CCL2 OUT
5	9	13	PA04 (5)	VDDANA	EXTINT[4]/ EXTWAKE[4]	VREFB	AIN[4]	AIN[0]			OA_OUT[2]		SERCOM0/ PAD[0]	TCC0/ WO[0]				CCL0 IN[0]
6	10	14	PA05 (5)	VDDANA	EXTINT[5]/ EXTWAKE[5]		AIN[5]	AIN[1]		VOUT[1]	OA_POS[2]		SERCOM0/ PAD[1]	TCC0/ WO[1]				CCL0 IN[1]
7	11	15	PA06	VDDANA	EXTINT[6]/ EXTWAKE[6]		AIN[6]	AIN[2]	Y[4]		OA_POS[0]		SERCOM0/ PAD[2]	TCC1/ WO[0]				CCL0 IN[2]
8	12	16	PA07 (5)	VDDANA	EXTINT[7]/ EXTWAKE[7]		AIN[7]	AIN[3]			OA_OUT[0]		SERCOM0/ PAD[3]	TCC1/ WO[1]				CCL0 OUT
11	13	17	PA08 (5)	VDDIO(6)	NMI		AIN[16]		X[0] Y[6]			SERCOM0/ PAD[0]	SERCOM2/ PAD[0]	TCC0/ WO[0]	TCC1/ WO[2]			CCL1 IN[0]
12	14	18	PA09	VDDIO(6)	EXTINT[9]		AIN[17]		X[1] Y[7]			SERCOM0/ PAD[1]	SERCOM2/ PAD[1]	TCC0/ WO[1]	TCC1/ WO[3]			CCL1 IN[1]
13	15	19	PA10	VDDIO(6)	EXTINT[10]		AIN[18]		X[2] Y[8]			SERCOM0/ PAD[2]	SERCOM2/ PAD[2]	TCC1/ WO[0]	TCC0/ WO[2]		GCLK_IO[4]	CCL1 IN[2]
14	16	20	PA11	VDDIO(6)	EXTINT[11]		AIN[19]		X[3] Y[9]			SERCOM0/ PAD[3]	SERCOM2/ PAD[3]	TCC1/ WO[1]	TCC0/ WO[3]		GCLK_IO[5]	CCL1 OUT
	19	23	PB10	VDDIO	EXTINT[10]				Y[2]				SERCOM4/ PAD[2]	TC1/ WO[0]	TCC0/ WO[4]		GCLK_IO[4]	CCL1 IN[2]
	20	24	PB11	VDDIO	EXTINT[11]				Y[3]				SERCOM4/ PAD[3]	TC1/ WO[1]	TCC0/ WO[5]		GCLK_IO[5]	CCL1 OUT
		25	PB12	VDDIO	EXTINT[12]				X[12] Y[5]			SERCOM4/ PAD[0		TC0/ WO[0]	TCC0/ WO[6]		GCLK_IO[6]	
		26	PB13	VDDIO	EXTINT[13]				X[13] Y[14]			SERCOM4/ PAD[1]		TC0/ WO[1]	TCC0/ WO[7]		GCLK_IO[7]	

Pin			I/O pin	Supply	A	B[1][2]						C	D	E	F	G	H	I
SAML21E	SAML21G	SAML21J			EIC/RSTC	REF	ADC	AC	PTC	DAC	OPAMP	SERCOM [1][2]	SERCOM-ALT	TC/TCC [3]	TCC	COM	AC/GCLK/SUPC	CCL
		27	PB14	VDDIO	EXTINT[14]				X[14]			SERCOM4/PAD[2]		TC1/WO[0]			GCLK_IO[0]	CCL3 IN[0]
		28	PB15	VDDIO	EXTINT[15]				X[15]			SERCOM4/PAD[3]		TC1/WO[1]			GCLK_IO[1]	CCL3 IN[1]
	21	29	PA12	VDDIO	EXTINT[12]							SERCOM2/PAD[0]	SERCOM4/PAD[0]	TCC2/WO[0]	TCC0/WO[6]		AC/CMP[0]	
	22	30	PA13	VDDIO	EXTINT[13]							SERCOM2/PAD[1]	SERCOM4/PAD[1]	TCC2/WO[1]	TCC0/WO[7]		AC/CMP[1]	
15	23	31	PA14	VDDIO[6]	EXTINT[14]							SERCOM2/PAD[2]	SERCOM4/PAD[2]	TC4/WO[0]	TCC0/WO[4]		GCLK_IO[0]	
16	24	32	PA15	VDDIO[6]	EXTINT[15]							SERCOM2/PAD[3]	SERCOM4/PAD[3]	TC4/WO[1]	TCC0/WO[5]		GCLK_IO[1]	
17	25	35	PA16	VDDIO[6]	EXTINT[0]				X[4]			SERCOM1/PAD[0]	SERCOM3/PAD[0]	TCC2/WO[0]	TCC0/WO[6]		GCLK_IO[2]	CCL0 IN[0]
18	26	36	PA17	VDDIO[6]	EXTINT[1]				X[5]			SERCOM1/PAD[1]	SERCOM3/PAD[1]	TCC2/WO[1]	TCC0/WO[7]		GCLK_IO[3]	CCL0 IN[1]
19	27	37	PA18	VDDIO[6]	EXTINT[2]				X[6]			SERCOM1/PAD[2]	SERCOM3/PAD[2]	TC4/WO[0]	TCC0/WO[2]		AC/CMP[0]	CCL0 IN[2]
20	28	38	PA19	VDDIO[6]	EXTINT[3]				X[7]			SERCOM1/PAD[3]	SERCOM3/PAD[3]	TC4/WO[1]	TCC0/WO[3]		AC/CMP[1]	CCL0 OUT
		39	PB16	VDDIO	EXTINT[0]							SERCOM5/PAD[0]		TC2/WO[0]	TCC0/WO[4]		GCLK_IO[2]	CCL3 IN[2]
		40	PB17	VDDIO	EXTINT[1]							SERCOM5/PAD[1]		TC2/WO[1]	TCC0/WO[5]		GCLK_IO[3]	CCL3 OUT
	29	41	PA20	VDDIO	EXTINT[4]				X[8]			SERCOM5/PAD[2]	SERCOM3/PAD[2]	TC3/WO[0]	TCC0/WO[6]		GCLK_IO[4]	
	30	42	PA21	VDDIO	EXTINT[5]				X[9]			SERCOM5/PAD[3]	SERCOM3/PAD[3]	TC3/WO[1]	TCC0/WO[7]		GCLK_IO[5]	
21	31	43	PA22	VDDIO[6]	EXTINT[6]				X[10]			SERCOM3/PAD[0]	SERCOM5/PAD[0]	TC0/WO[0]	TCC0/WO[4]		GCLK_IO[6]	CCL2 IN[0]
22	32	44	PA23	VDDIO[6]	EXTINT[7]				X[11]			SERCOM3/PAD[1]	SERCOM5/PAD[1]	TC0/WO[1]	TCC0/WO[5]	USB/SOF 1kHz	GCLK_IO[7]	CCL2 IN[1]
23	33	45	PA24	VDDIO[6]	EXTINT[12]							SERCOM3/PAD[2]	SERCOM5/PAD[2]	TC1/WO[0]	TCC1/WO[2]	USB/DM		CCL2 IN[2]
24	34	46	PA25	VDDIO[6]	EXTINT[13]							SERCOM3/PAD[3]	SERCOM5/PAD[3]	TC1/WO[1]	TCC1/WO[3]	USB/DP		CCL2 OUT
	37	49	PB22	VDDIN	EXTINT[6]								SERCOM5/PAD[2]	TC3/WO[0]			GCLK_IO[0]	CCL0 IN[0]
	38	50	PB23	VDDIN	EXTINT[7]								SERCOM5/PAD[3]	TC3/WO[1]			GCLK_IO[1]	CCL0 OUT
25	39	51	PA27	VDDIN	EXTINT[15]												GCLK_IO[0]	
31	45	57	PA30	VDDIN	EXTINT[10]								SERCOM1/PAD[2]	TCC1/WO[0]		CORTEX_M0P/SWCLK	GCLK_IO[0]	CCL1 IN[0]
32	46	58	PA31	VDDIN	EXTINT[11]								SERCOM1/PAD[3]	TCC1/WO[1]		SWDIO [4]		CCL1 OUT
		59	PB30	VDDIN	EXTINT[14]								SERCOM5/PAD[0]	TCC0/WO[0]	TCC1/WO[2]			
		60	PB31	VDDIN	EXTINT[15]								SERCOM5/PAD[1]	TCC0/WO[1]	TCC1/WO[3]			
		61	PB00	VSWOUT	EXTINT[0]		AIN[8]						SERCOM5/PAD[2]	TC3/WO[0]			SUPC/PSOK	CCL0 IN[1]
		62	PB01	VSWOUT	EXTINT[1]		AIN[9]						SERCOM5/PAD[3]	TC3/WO[1]			SUPC/OUT[0]	CCL0 IN[2]
	47	63	PB02	VSWOUT	EXTINT[2]		AIN[10]						SERCOM5/PAD[0]	TC2/WO[0]			SUPC/OUT[1]	CCL0 OUT
	48	64	PB03	VSWOUT	EXTINT[3]		AIN[11]						SERCOM5/PAD[1]	TC2/WO[1]			SUPC/VBAT	

Note:

1. All analog pin functions are on peripheral function B. Peripheral function B must be selected to disable the digital control of the pin.

2. Only some pins can be used in SERCOM I^2C mode. See also SERCOM I2C Pins on page 32.
3. TC2 and TC3 are not supported on SAM L21G. Refer to Configuration Summary on page 15 for details.
4. This function is only activated in the presence of a debugger.
5. When an analog peripheral is enabled, the analog output of the peripheral will interfere with the alternative functions of this pin. This is also true even when the peripheral is used for internal purposes.
6. On SAM L21E, VDDIO is electrically connected to the VDDANA domain and supplied through VDDANA.
7. Clusters of multiple GPIO pins are sharing the same supply pin. See GPIO Clusters on page 32.

Related Links

Electrical Characteristics on page 1140

7.2. Other Functions

7.2.1. Oscillator Pinout

The oscillators are not mapped to the normal PORT functions and their multiplexing are controlled by registers in the Oscillators Controller (OSCCTRL) and in the 32KHz Oscillators Controller (OSC32KCTRL).

Table 7-2. Oscillator Pinout

Oscillator	Supply	Signal	I/O pin
XOSC	VDDIO	XIN	PA14
		XOUT	PA15
XOSC32K	VSWOUT	XIN32	PA00
		XOUT32	PA01

Note: To improve the cycle-to-cycle jitter of XOSC32, it is recommended to keep the neighboring pins of XIN32 and XOUT32 following pins as static as possible.

Table 7-3. XOSC32 Jitter Minimization

Package Pin Count	Static Signal Recommended
64	PB00, PB01, PB02, PB03, PA02, PA03
48	PB02, PB03, PA02, PA03
32	PA02, PA03

Related Links

External Real Time Oscillator on page 1195

7.2.2. Serial Wire Debug Interface Pinout

Only the SWCLK pin is mapped to the normal PORT functions. A debugger cold-plugging or hot-plugging detection will automatically switch the SWDIO port to the SWDIO function.

Table 7-4. Serial Wire Debug Interface Pinout

Signal	Supply	I/O pin
SWCLK	VDDIN	PA30
SWDIO	VDDIN	PA31

7.2.3. SERCOM I²C Pins

Table 7-5. SERCOM Pins Supporting I²C

Device	Pins Supporting I²C Hs mode
SAML21E	PA08, PA09, PA16, PA17, PA22, PA23
SAML21G	PA08, PA09, PA12, PA13, PA16, PA17, PA22, PA23, PB16, PB17
SAML21J	PA08, PA09, PA12, PA13, PA16, PA17, PA22, PA23, PB12, PB13, PB16, PB17, PB30, PB31

7.2.4. GPIO Clusters

Table 7-6. GPIO Clusters

PACKAGE	CLUSTER	GPIO																SUPPLIES PINS CONNECTED TO THE CLUSTER
64pins	1	PB31	PB30	PA31	PA30	PA27												V_{DDIN} pin56/GND pin54
	2	PB23	PB22	PA25	PA24	PA23	PA22	PA21	PA20	PB17	PB16	PA19	PA18	PA17	PA16			V_{DDIO} pin 48/GND pin47 and V_{DDIO} pin34/GND pin33
	4	PA15	PA14	PA13	PA12	PB15	PB14	PB13	PB12	PB11	PB10							V_{DDIO} pin 34/GND pin33 and V_{DDIO} pin21/GND pin22
	5	PA11	PA10	PA09	PA08													V_{DDIO} pin21/GND pin22
	6	PA07	PA06	PA05	PA04	PB09	PB08	PB07	PB06	PB05	PB04	PA03	PA02					V_{DDANA} pin 8/GNDANA pin7
	7	PA01	PA00	PB03	PB02	PB01	PB00											VSWOUT

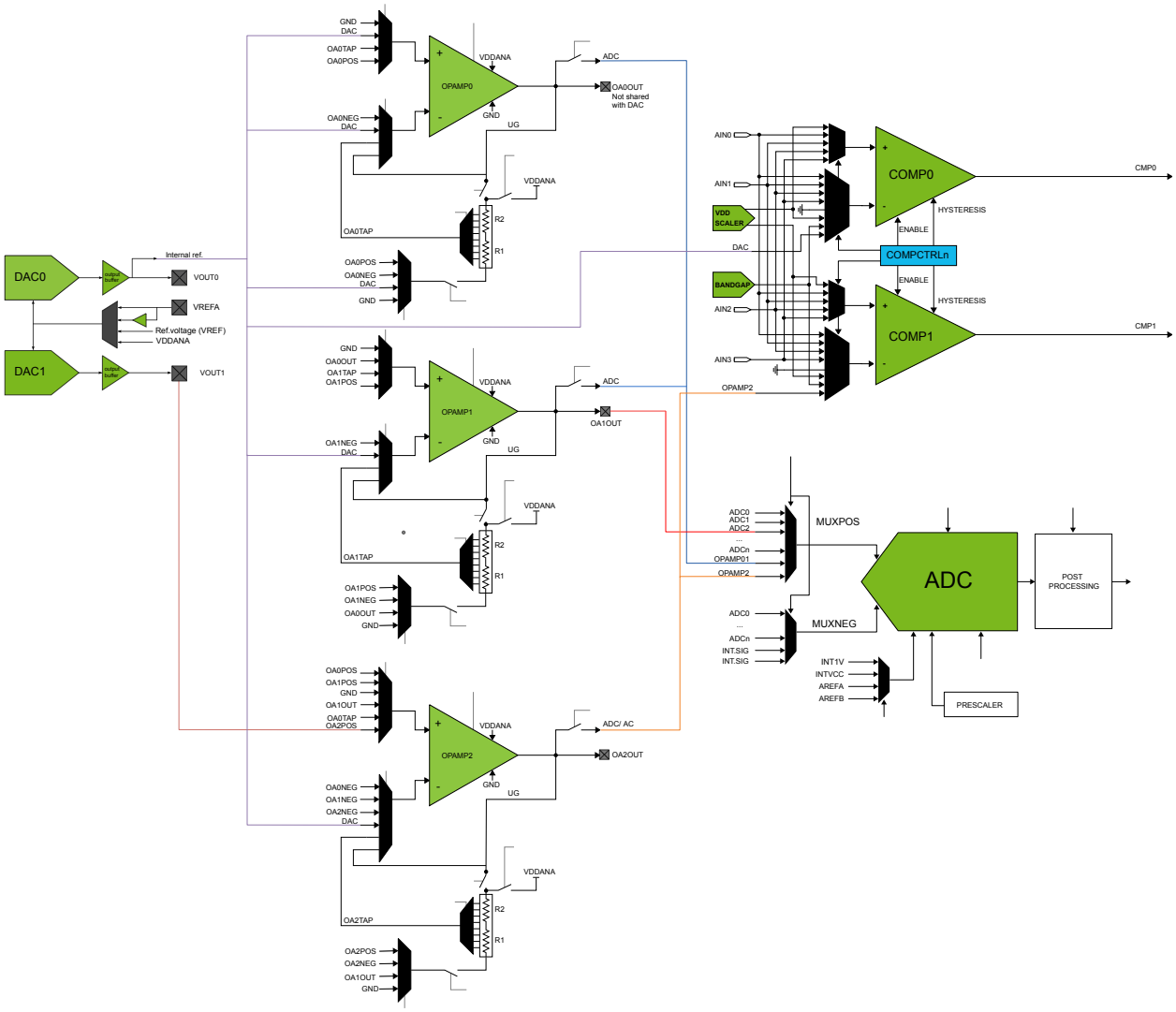
PACKAGE	CLUSTER	GPIO																SUPPLIES PINS CONNECTED TO THE CLUSTER
48pins	1	PA31	PA30															V_{DDIN} pin44/GND pin42
	2	PA28	PA27	PB23	PB22													V_{DDIN} pin44/GND pin42 and V_{DDIO} pin36/GND pin35
	3	PA25	PA24	PA23	PA22	PA21	PA20	PA19	PA18	PA17	PA16	PA15	PA14	PA13	PA12	PB11	PB10	V_{DDIO} pin36/GND pin35 and V_{DDIO} pin17/GND pin18
	4	PA11	PA10	PA09	PA08													VDDIO pin17/GND pin18
	5	PA07	PA06	PA05	PA04	PB09	PB08											V_{DDANA} pin6/ GNDANA pin5
	6	PA03	PA02	PA01	PA00	PB03	PB02											V_{DDANA} pin6/ GNDANA pin5
32pins	1	PA31	PA30	PA27	PA25	PA24	PA23	PA22										V_{DDIN} pin30/GND pin 28
	2	PA18	PA17	PA16	PA15	PA14	PA11	PA10	PA09	PA08								V_{DDANA} pin9/GND pin 10
	3	PA07	PA06	PA05	PA04	PA03	PA02											V_{DDANA} pin9/GND pin10
	4	PA01	PA00															VSWOUT

8. Analog Connections of Peripherals

This chapter provides a global view of the analog system, the signal interconnections, and the ONDEMAND function of the peripherals that process analog signals: AC, ADC, DAC, OPAMP.

8.1. Block Diagram

Figure 8-1. Interconnections of Analog Signal Components



8.2. Analog Connections

The analog peripherals can be connected to each other according to the block diagram above. To configure a particular peripheral refer to the corresponding description in this datasheet.

Peripherals can be connected through a pad. In this case, the digital functionality of the pad is lost and its configuration must not interfere with the analog connection.

Important:
When an analog peripheral is enabled, the analog output of the peripheral will interfere with the alternative functions of the output pads. This is also true even when the peripheral is used for internal purposes.

Analog inputs do not interfere with alternative pad functions.

8.3. Analog ONDEMAND Function

General Function
The analog ONDEMAND feature allows other analog peripherals to request the OPAMP.
Note: The analog ONDEMAND is independent of the ONDEMAND bit located in each source clock controller, used for requesting source clocks.

The OPAMP can be enabled by requests from ADC or AC.

The request mechanism is activated by writing a '1' to the OPAMP.OPAMPCTRLx.ONDEMAND bit. When a request is sent by one of the peripherals to OPAMPx, the OPAMPx will start up and acknowledge the request as soon as it is fully enabled.

If the OPAMP.OPAMPCTRLx.ONDEMAND bit is '0' but OPAMPx is enabled already, a request will be immediately acknowledged. If OPAMPCTRLx.ONDEMAND=0 and the OPAMPx is disabled, requests will not be acknowledged: requests are handled only when the OPAMP output is active (OPAMPCTRLx.ANAOUT=1).

In Standby sleep mode, the ONDEMAND operation is still active if OPAMPCTRLx.ONDEMAND=1. If OPAMPCTRLx.ONDEMAND=0, the OPAMPx is disabled.

The OPAMP controller peripheral must be configured appropriately before being requested.

For the ADC peripheral, ONDEMAND requests to the OPAMP are enabled by writing the ADC.CTRLA.ONDEMAND bit to '1'.

For the AC peripheral,there is no explicit ONDEMAND bit in the registers. ONDEMAND requests to OPAMPx are issued either when the AC is used in single-shot mode, or when comparisons are triggered by events from the Event System. The OPAMP must be selected as input of the AC previously.

> When the Negative Input MUX Selection bit field of the Comparator 1 Control register is set to DAC/OPAMP (AC.COMPCTRL1.MUXNEG=0x7), the AC will start issuing ONDEMAND requests to OPAMP.

Alternative Requests
When OPAMPx is set to accept ONDEMAND requests (OPAMP.OPAMCTRLx.ONDEMAND=1) but the ADC is not configured to issue requests to it (ADC.CTRLA.ONDEMAND=0), the ADC will send continuous requests to the receiver selected by ADC.INPUTCTRL.MUXPOS.

> If ADC.INPUTCTRL.MUXPOS=0x1E, OPAMP0 and OPAMP1 will receive requests.
>
> If ADC.INPUTCTRL.MUXPOS=0x1F, only OPAMP2 will receive requests.

When OPAMPx is set to accept ONDEMAND requests (OPAMP.OPAMCTRLx.ONDEMAND=1) but the AC is not configured to issue requests to it (AC.CTRLA.ONDEMAND=0), the AC will send continuous requests to the receiver selected by AC.COMPCTRLx.MUXNEG.

If AC.COMPCTRL1.MUXNEG=0x7, OPAMP2 will receive requests.

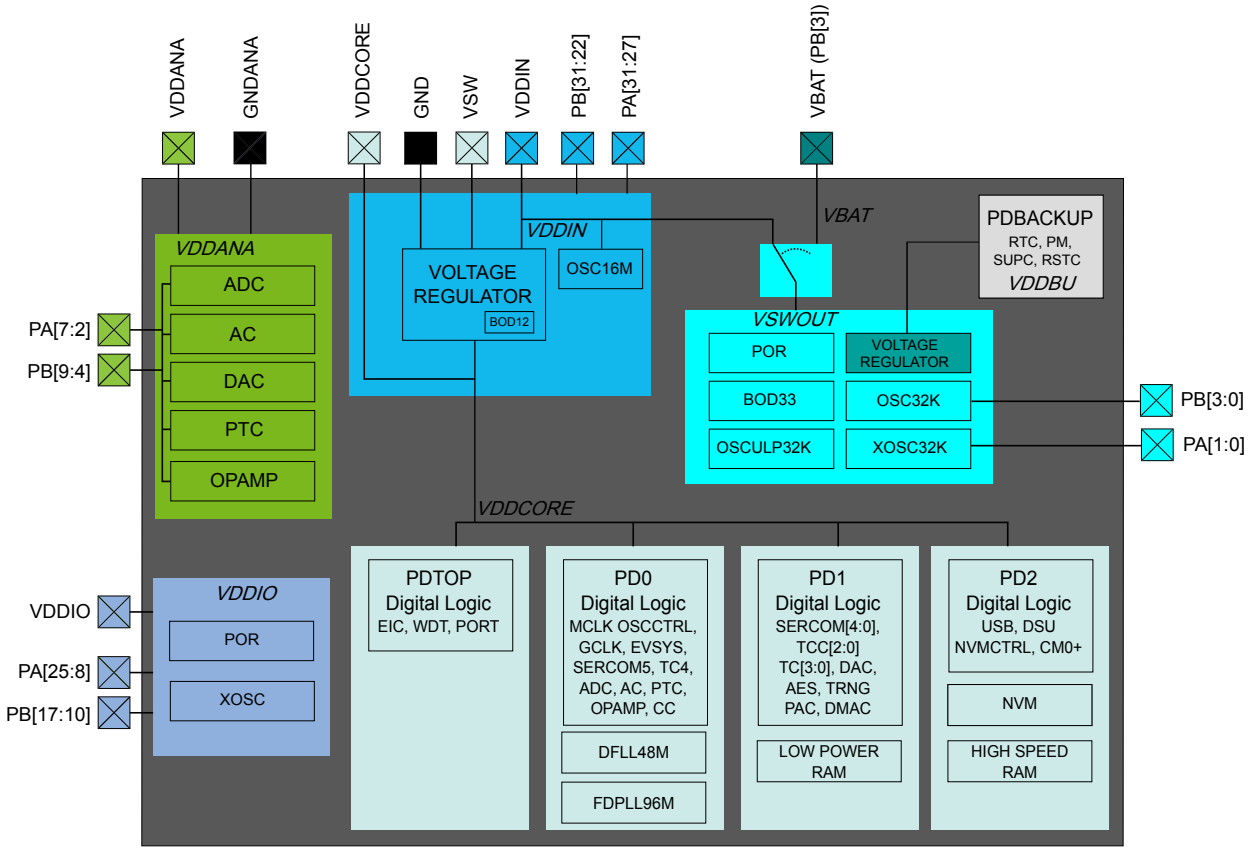
If AC.COMPCTRL0.MUXNEG=0x7, DAC0 will receive requests.

Related Links

OPAMP – Operational Amplifier Controller on page 1007
ADC – Analog-to-Digital Converter on page 1030
AC – Analog Comparators on page 1072

9. Power Supply and Start-Up Considerations

9.1. Power Domain Overview



The Atmel SAM L21 power domains are not independent of each other:

- VDDCORE, VDDIO and VDDIN share GND, whereas VDDANA refers to GNDANA.
- VDDANA and VDDIN must share the main supply, VDD.
- VDDCORE serves as the internal voltage regulator output. It powers the core, memories, peripherals, DFLL48M and FDPLL96M.
- VSWOUT and VDDBU are internal power domains.
- On SAM L21E, VDDIO is electrically connected to the VDDANA domain and supplied through VDDANA.

9.2. Power Supply Considerations

9.2.1. Power Supplies

The Atmel SAM L21 has several different power supply pins:

- VDDIO powers I/O lines and XOSC. Voltage is 1.62V to 3.63V
- VDDIN powers I/O lines, OSC16M, the internal regulator for VDDCORE and the Automatic Power Switch. Voltage is 1.62V to 3.63V
- VDDANA powers I/O lines and the ADC, AC, DAC, PTC and OPAMP. Voltage is 1.62V to 3.63V

- VBAT powers the Automatic Power Switch. Voltage is 1.62V to 3.63V
- VDDCORE serves as the internal voltage regulator output. It powers the core, memories, peripherals, DFLL48M and FDPLL96M. Voltage is 0.9V to 1.2V typical.
- The Automatic Power Switch is a configurable switch that selects between VDDIN and VBAT as supply for the internal output VSWOUT, see the figure in Power Domain Overview.

The same voltage must be applied to both VDDIN and VDDANA. This common voltage is referred to as VDD in the datasheet.

When the Peripheral Touch Controller (PTC) is used, VDDIO must be equal to VDD. When the PTC is not used by the user application, VDDIO may be lower than VDD.

The ground pins, GND, are common to VDDCORE, VDDIO and VDDIN. The ground pin for VDDANA is GNDANA.

For decoupling recommendations for the different power supplies, refer to the schematic checklist.

Related Links

Schematic Checklist on page 1189

9.2.2. Voltage Regulator

The SAM L21 internal Voltage Regulator has four different modes:

- Linear mode: This is the default mode when CPU and peripherals are running. It does not require an external inductor.
- Switching mode. This is the most efficient mode when the CPU and peripherals are running. This mode can be selected by software on the fly.
- Low Power (LP) mode. This is the default mode used when the chip is in standby mode.
- Shutdown mode. When the chip is in backup mode, the internal regulator is off.

Note that the Voltage Regulator modes are controlled by the Power Manager.

9.2.3. Typical Powering Schematic

The SAM L21uses a single supply from 1.62V to 3.63V.

The following figure shows the recommended power supply connection.

Figure 9-1. Power Supply Connection for Linear Mode Only

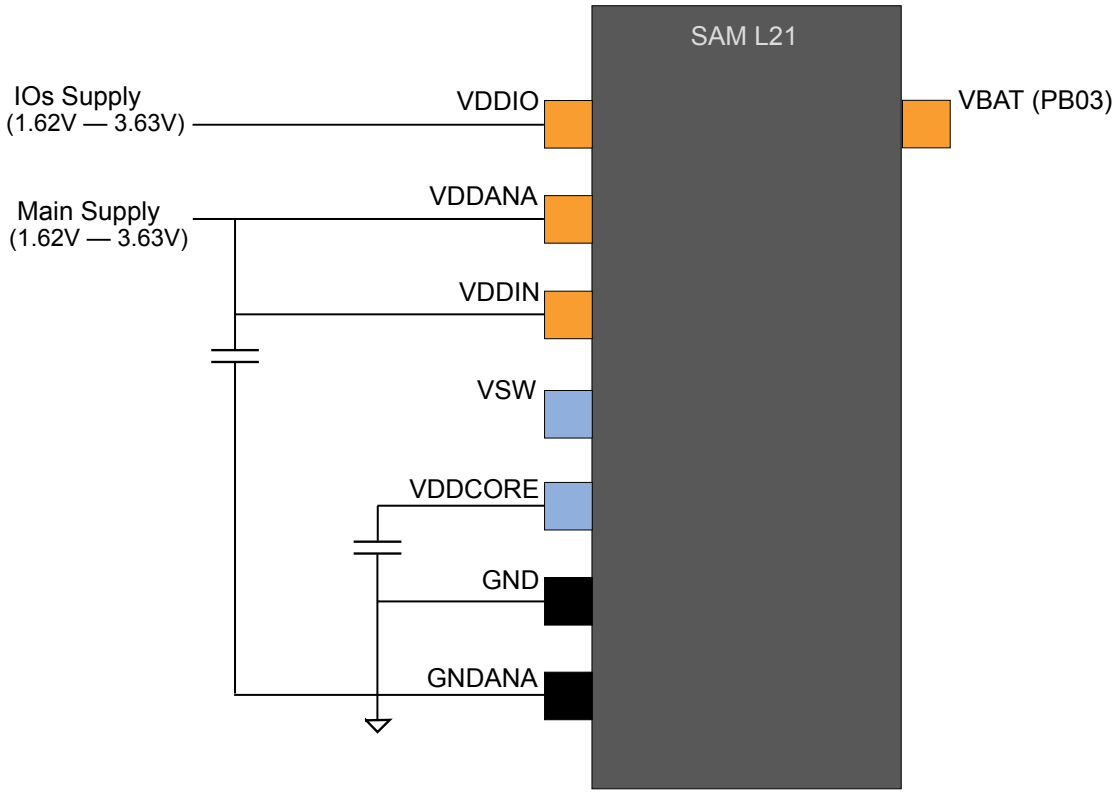


Figure 9-2. Power Supply Connection for Switching/Linear Mode

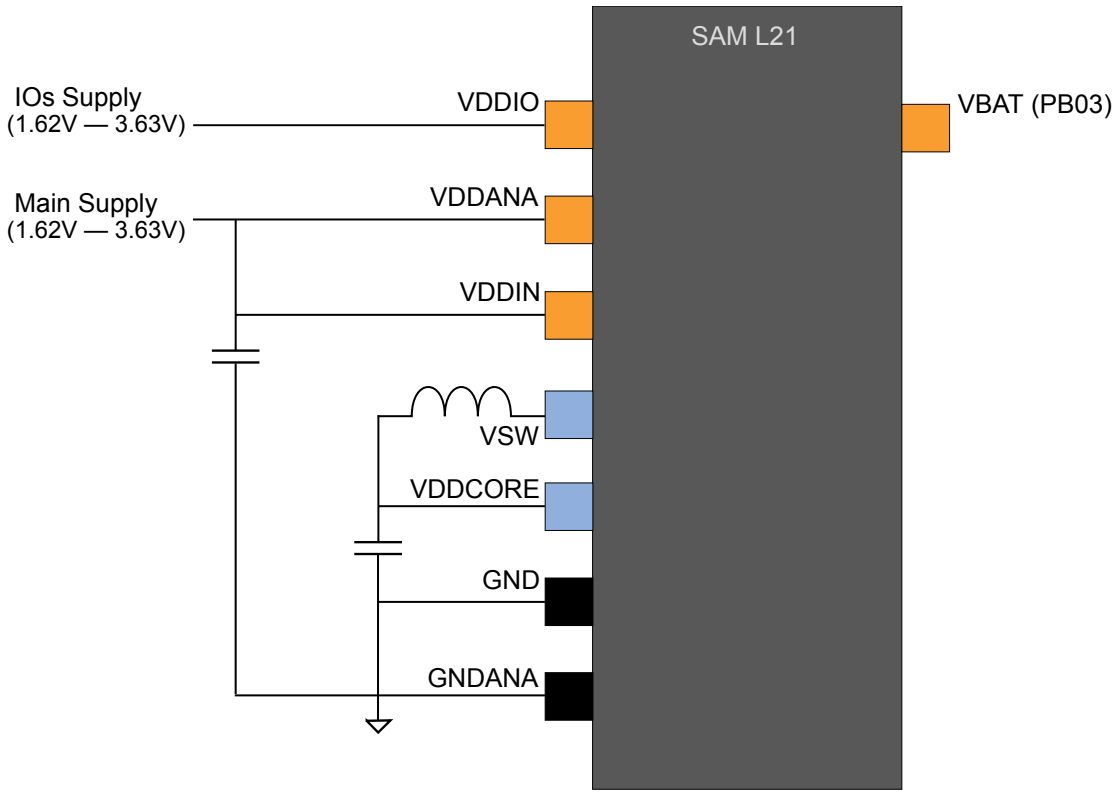
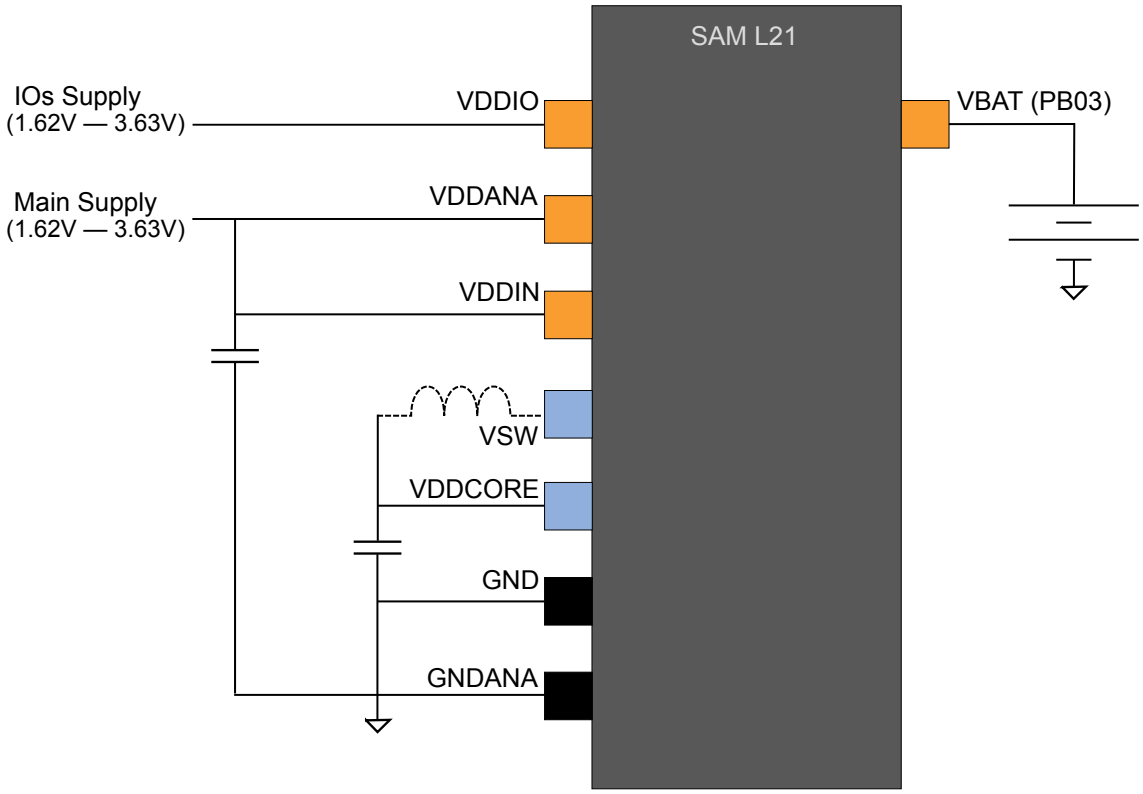


Figure 9-3. Power Supply Connection for Battery Backup



9.2.4. Power-Up Sequence

9.2.4.1. Supply Order

VDDIN and VDDANA must have the same supply sequence. Ideally, they must be connected together.

VDDIO can rise before or after VDDIN and VDDANA. Note that VDDIO supplies the XOSC, so VDDIO must be present before the application uses the XOSC feature. This is also applicable to all digital features present on pins supplied by VDDIO.

9.2.4.2. Minimum Rise Rate

The two integrated power-on reset (POR) circuits monitoring VDDIN and VDDIO require a minimum rise rate.

Related Links

Electrical Characteristics on page 1140

9.2.4.3. Maximum Rise Rate

The rise rate of the power supplies must not exceed the values described in Electrical Characteristics.

Related Links

Electrical Characteristics on page 1140

9.3. Power-Up

This section summarizes the power-up sequence of the SAM L21. The behavior after power-up is controlled by the Power Manager.

Related Links

PM – Power Manager on page 186

9.3.1. Starting of Internal Regulator

After power-up, the device is set to its initial state and kept in Reset, until the power has stabilized throughout the device. The default performance level after power-up is PL0. See section on *PM-Power Manager* for details. The internal regulator provides the internal VDDCORE corresponding to this performance level. Once the external voltage VDDIN and the internal VDDCORE reach a stable value, the internal Reset is released.

Related Links

PM – Power Manager on page 186

9.3.2. Starting of Clocks

Once the power has stabilized and the internal Reset is released, the device will use a 4MHz clock by default. The clock source for this clock signal is OSC16M, which is enabled and configured at 4MHz after a reset by default. This is also the default time base for Generic Clock Generator 0. In turn, Generator 0 provides the main clock GCLK_MAIN which is used by the Power Manager (PM).

Some synchronous system clocks are active after Start-Up, allowing software execution. Synchronous system clocks that are running receive the 4MHz clock from Generic Clock Generator 0. Other generic clocks are disabled.

Related Links

PM – Power Manager on page 186

Peripheral Clock Masking on page 151

9.3.3. I/O Pins

After power-up, the I/O pins are tri-stated except PA30, which is pull-up enabled and configured as input.

Related Links

PM – Power Manager on page 186

9.3.4. Fetching of Initial Instructions

After Reset has been released, the CPU starts fetching PC and SP values from the Reset address, 0x00000000. This points to the first executable address in the internal Flash memory. The code read from the internal Flash can be used to configure the clock system and clock sources. Refer to the ARM Architecture Reference Manual for more information on CPU startup (http://www.arm.com).

Related Links

PM – Power Manager on page 186

GCLK - Generic Clock Controller on page 131

OSC32KCTRL – 32KHz Oscillators Controller on page 264

9.4. Power-On Reset and Brown-Out Detector

The SAM L21 embeds three features to monitor, warn and/or reset the device:

- POR: Power-on Reset on VDDIN, VSWOUT and VDDIO
- BOD33: Brown-out detector on VSWOUT/VBAT
- Brown-out detector internal to the voltage regulator for VDDCORE. BOD12 is calibrated in production and its calibration parameters are stored in the NVM User Row. This data should not be changed if the User Row is written to in order to assure correct behavior.

9.4.1. Power-On Reset on VDDIN

VDDIN is monitored by POR. Monitoring is always activated, including startup and all sleep modes. If VDDIN goes below the threshold voltage, the entire chip is reset.

9.4.2. Power-On Reset on VSWOUT

VSWOUT is monitored by POR. Monitoring is always activated, including startup and all sleep modes. If VSWOUT goes below the threshold voltage, the entire chip is reset.

9.4.3. Power-On Reset on VDDIO

VDDIO is monitored by POR. Monitoring is always activated, including startup and all sleep modes. If VDDIO goes below the threshold voltage, all I/Os supplied by VSWOUT are reset.

9.4.4. Brown-Out Detector on VSWOUT/VBAT

BOD33 monitors VSWOUT or VBAT depending on configuration.

Related Links

SUPC – Supply Controller on page 283
Battery Backup Power Switch on page 288

9.4.5. Brown-Out Detector on VDDCORE

Once the device has started up, BOD12 monitors the internal VDDCORE.

Related Links

SUPC – Supply Controller on page 283
Battery Backup Power Switch on page 288

9.5. Performance Level Overview

By default, the device will start in Performance Level 0. This PL0 is aiming for the lowest power consumption by limiting logic speeds and the CPU frequency. As a consequence, all GCLK will have limited capabilities, and some peripherals and clock sources will not work or with limited capabilities:

List of peripherals/clock sources not available in PL0:

- USB (limited by logic frequency)
- DFLL48M

List of peripherals/clock sources with limited capabilities in PL0:

- All AHB/APB peripherals are limited by CPU frequency
- DPLL96M: may be able to generate 48MHz internally, but the output cannot be used by logic
- GCLK: the maximum frequency is by factor 4 compared to PL2
- SW interface: the maximum frequency is by factor 4 compared to PL2
- TC: the maximum frequency is by factor 4 compared to PL2
- TCC:the maximum frequency is by factor 4 compared to PL2
- SERCOM: the maximum frequency is by factor 4 compared to PL2

List of peripherals/clock sources with full capabilities in PL0:

- AC
- ADC
- DAC
- EIC

- OPAMP
- OSC16M
- PTC
- All 32KHz clock sources and peripherals

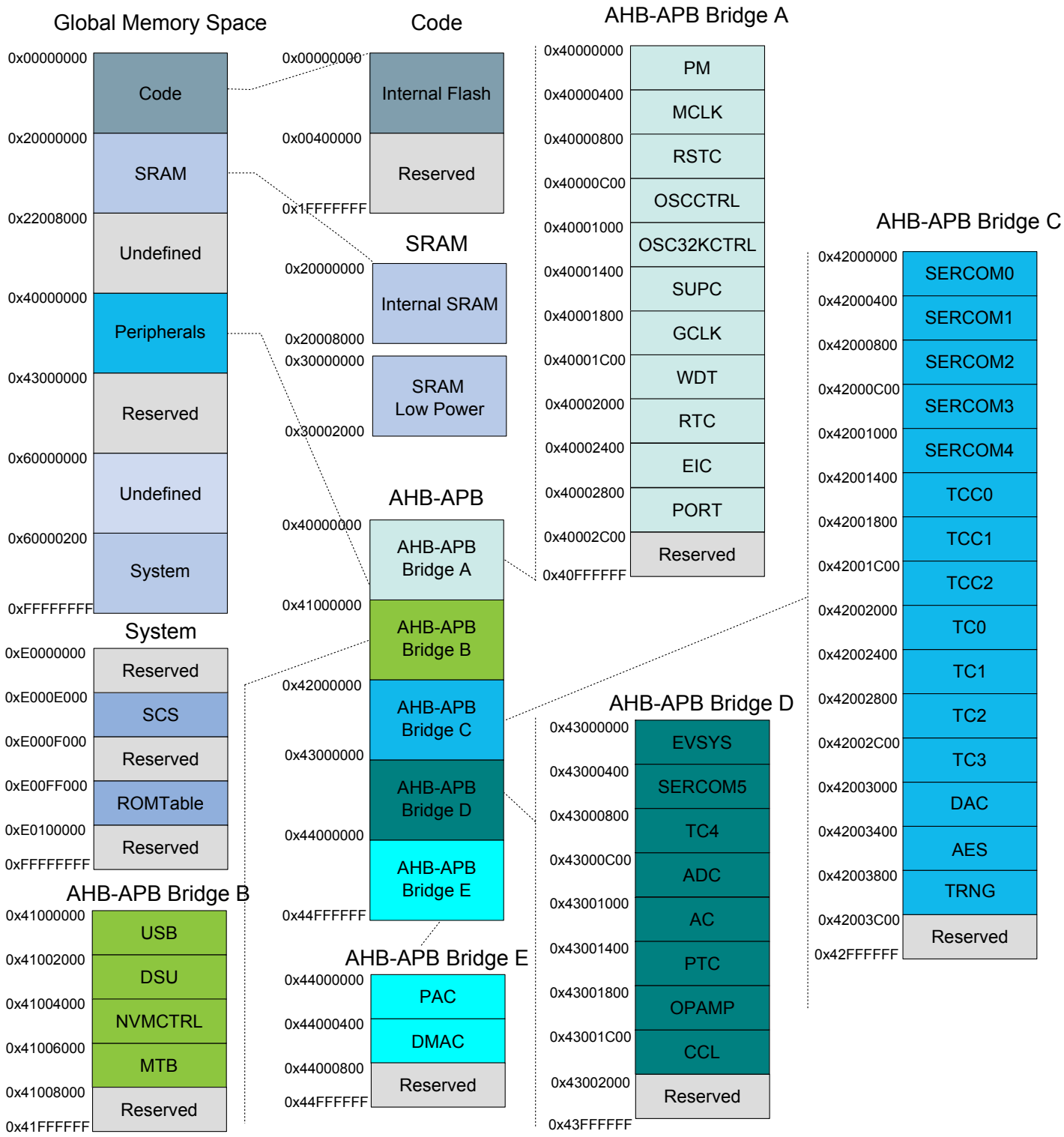
Full functionality and capability will be ensured in PL2. When transitioning between performance levels, the Supply Controller (SUPC) will provide a configurable smooth voltage scaling transition.

Related Links

PM – Power Manager on page 186
SUPC – Supply Controller on page 283
Block Diagram on page 21

10. Product Mapping

Figure 10-1. Atmel SAM L21 Product Mapping



11. Memories

11.1. Embedded Memories

- Internal high-speed Flash with Read-While-Write (RWW) capability on a section of the array
- Internal high-speed RAM, single-cycle access at full speed
- Internal low-power RAM, single-cycle access at full speed

11.2. Physical Memory Map

The high-speed bus is implemented as a bus matrix. All high-speed bus addresses are fixed, and they are never remapped in any way, even during boot. The 32-bit physical address space is mapped as follows:

Table 11-1. SAM L21 Physical Memory Map(1)

Memory	Start address	Size [KB]			
		SAML21x18	SAML21x17	SAML21x16	SAML21E15
Embedded Flash	0x00000000	256	128	64	32
Embedded RWW section	0x00400000	8	4	2	1
Embedded SRAM	0x20000000	32	16	8	4
Embedded low-power SRAM	0x30000000	8	8	4	2
Peripheral Bridge A	0x40000000	64	64	64	64
Peripheral Bridge B	0x41000000	64	64	64	64
Peripheral Bridge C	0x42000000	64	64	64	64
Peripheral Bridge D	0x43000000	64	64	64	64
Peripheral Bridge E	0x44000000	64	64	64	64
IOBUS	0x60000000	0.5	0.5	0.5	0.5

Note: 1. x = G, J, or E.

Table 11-2. Flash Memory Parameters(1)

Device	Flash size [KB]	Number of pages	Page size [Bytes]
SAML21x18	256	4096	64
SAML21x17	128	2048	64
SAML21x16	64	1024	64
SAML21E15	32	512	64

Note: 1. x = G, J, or E.

Table 11-3. RWW Section Parameters[1]

Device	Flash size [KB]	Number of pages	Page size [Bytes]
SAML21x18	8	128	64
SAML21x17	4	64	64
SAML21x16	2	32	64
SAML21E15	1	16	64

Note: 1. x = G, J, or E.

11.3. NVM User Row Mapping

The Non Volatile Memory (NVM) User Row contains calibration data that are automatically read at device power-on.

The NVM User Row can be read at address 0x00804000.

Note: When writing to the user row, the new values do not get loaded by the other modules on the device until a device reset occurs.

Table 11-4. NVM User Row Mapping

Bit Position	Name	Usage	Affected Settings
2:0	BOOTPROT	Used to select one of eight different bootloader sizes.	NVMCTRL - Non-Volatile Memory Controller
3	Reserved	—	—
6:4	EEPROM	Used to select one of eight different EEPROM sizes.	NVMCTRL - Non-Volatile Memory Controller
7	Reserved	—	—
13:8	BOD33 Level	BOD33 threshold level at power-on.	Supply Controller, register BOD33
14	BOD33 Disable	BOD33 Disable at power-on.	Supply Controller, register BOD33
16:15	BOD33 Action	BOD33 Action at power-on.	Supply Controller, register BOD33
22:17	BOD12 Level	BOD12 threshold level at power-on.	Supply Controller, register BOD12
23	BOD12 Disable	BOD12 Disable at power-on.	Supply Controller, register BOD12
25:24	BOD12 Action	BOD12 Action at power-on.	Supply Controller, register BOD12
26	WDT Enable	WDT Enable at power-on.	Watchdog Timer, register CTRLA
27	WDT Always-On	WDT Always-On at power-on.	Watchdog Timer, register CTRLA

Bit Position	Name	Usage	Affected Settings
31:28	WDT Period	WDT Period at power-on.	Watchdog Timer, register CONFIG
35:32	WDT Window	WDT Window mode time-out at power-on.	Watchdog Timer, register CONFIG
39:36	WDT EWOFFSET	WDT Early Warning Interrupt Time Offset at power-on.	Watchdog Timer, register EWCTRL
40	WDT WEN	WDT Timer Window Mode Enable at power-on	Watchdog Timer, register CTRLA
41	BOD33 Hysteresis	BOD33 Hysteresis configuration at power-on.	Supply Controller, register BOD33
42	BOD12 Hysteresis	BOD12 Hysteresis configuration at power-on.	Supply Controller, register BOD12
47:43	Reserved	—	—
63:48	LOCK	NVM Region Lock Bits.	NVMCTRL - Non-Volatile Memory Controller

Related Links

NVMCTRL – Non-Volatile Memory Controller on page 483

SUPC – Supply Controller on page 283

WDT – Watchdog Timer on page 321

11.4. NVM Software Calibration Area Mapping

The NVM Software Calibration Area contains calibration data that are determined and written during production test. These calibration values should be read by the application software and written back to the corresponding register.

The NVM Software Calibration Area can be read at address 0x00806020.

The NVM Software Calibration Area can not be written.

Table 11-5. NVM Software Calibration Area Mapping

Bit Position	Name	Description
2:0	BIASREFBUF	ADC linearity. To be written to ADC CALIB.BIASREFBUF
5:3	BIASCOMP	ADC bias calibration. To be written to ADC CALIB.BIASCOMP
12:6	OSC32KCAL	OSC32K calibration. To be written to OSC32KCTRL OSC32K.CALIB
17:13	USB_TRANSN	USB pad calibration. To be written to USB PADCAL.TRANSN
22:18	USB_TRANSP	USB pad calibration. To be written to USB PADCAL.TRANSP

Bit Position	Name	Description
25:23	USB_TRIM	USB pad calibration. To be written to USB PADCAL.TRIM
31:26	DFLL48M_COARSE_CAL	DFLL48M coarse calibration. To be written to OSCCTRL DFLLVAL.COARSE

Related Links

CALIB on page 1071

PADCAL on page 921

DFLLVAL on page 254

11.5. Serial Number

Each device has a unique 128-bit serial number which is a concatenation of four 32-bit words contained at the following addresses:

Word 0: 0x0080A00C

Word 1: 0x0080A040

Word 2: 0x0080A044

Word 3: 0x0080A048

The uniqueness of the serial number is guaranteed only when using all 128 bits.

12. Processor and Architecture

12.1. Cortex M0+ Processor

The Atmel SAM L21 implements the ARM®Cortex™-M0+ processor, based on the ARMv6 Architecture and Thumb®-2 ISA. The Cortex M0+ is 100% instruction set compatible with its predecessor, the Cortex-M0 core, and upward compatible to Cortex-M3 and M4 cores. The implemented ARM Cortex-M0+ is revision r0p1. For more information refer to http://www.arm.com

12.1.1. Cortex M0+ Configuration

Table 12-1. Cortex M0+ Configuration in Atmel SAM L21

Features	Cortex M0+ options	Atmel SAM L21 configuration
Interrupts	External interrupts 0-32	29
Data endianness	Little-endian or big-endian	Little-endian
SysTick timer	Present or absent	Present
Number of watchpoint comparators	0, 1, 2	2
Number of breakpoint comparators	0, 1, 2, 3, 4	4
Halting debug support	Present or absent	Present
Multiplier	Fast or small	Fast (single cycle)
Single-cycle I/O port	Present or absent	Present
Wake-up interrupt controller	Supported or not supported	Not supported
Vector Table Offset Register	Present or absent	Present
Unprivileged/Privileged support	Present or absent	Absent - All software run in privileged mode only
Memory Protection Unit	Not present or 8-region	Not present
Reset all registers	Present or absent	Absent
Instruction fetch width	16-bit only or mostly 32-bit	32-bit

The ARM Cortex-M0+ core has two bus interfaces:

- Single 32-bit AMBA-3 AHB-Lite system interface that provides connections to peripherals and all system memory including Flash memory and RAM
- Single 32-bit I/O port bus interfacing to the PORT with 1-cycle loads and stores

12.1.1.1. Cortex M0+ Peripherals

- System Control Space (SCS)
  - The processor provides debug through registers in the SCS. Refer to the Cortex-M0+ Technical Reference Manual for details (http://www.arm.com)
- Nested Vectored Interrupt Controller (NVIC)

- External interrupt signals connect to the NVIC, and the NVIC prioritizes the interrupts. Software can set the priority of each interrupt. The NVIC and the Cortex-M0+ processor core are closely coupled, providing low latency interrupt processing and efficient processing of late arriving interrupts. Refer to the Cortex-M0+ Technical Reference Manual for details (http://www.arm.com).
  Note: When the CPU frequency is much higher than the APB frequency it is recommended to insert a memory read barrier after each CPU write to registers mapped on the APB. Failing to do so in such conditions may lead to unexpected behavior such as e.g. re-entering a peripheral interrupt handler just after leaving it.

- System Timer (SysTick)
  - The System Timer is a 24-bit timer clocked by CLK_CPU that extends the functionality of both the processor and the NVIC. Refer to the Cortex-M0+ Technical Reference Manual for details (http://www.arm.com).
- System Control Block (SCB)
  - The System Control Block provides system implementation information, and system control. This includes configuration, control, and reporting of the system exceptions. Refer to the Cortex-M0+ Devices Generic User Guide for details (http://www.arm.com)
- Micro Trace Buffer (MTB)
  - The CoreSight MTB-M0+ (MTB) provides a simple execution trace capability to the Cortex-M0+ processor. Refer to section MTB-Micro Trace Buffer and the CoreSight MTB-M0+ Technical Reference Manual for details (http://www.arm.com).

Related Links

Nested Vector Interrupt Controller on page 51

12.1.1.2. Cortex M0+ Address Map

Table 12-2. Cortex-M0+ Address Map

Address	Peripheral
0xE000E000	System Control Space (SCS)
0xE000E010	System Timer (SysTick)
0xE000E100	Nested Vectored Interrupt Controller (NVIC)
0xE000ED00	System Control Block (SCB)
0x41006000	Micro Trace Buffer (MTB)

12.1.1.3. I/O Interface

The device allows direct access to PORT registers. Accesses to the AMBA® AHB-Lite™ and the single cycle I/O interface can be made concurrently, so the Cortex M0+ processor can fetch the next instructions while accessing the I/Os. This enables single cycle I/O access to be sustained for as long as necessary.

Related Links

PORT: IO Pin Controller on page 506
CPU Local Bus on page 509

12.2. Nested Vector Interrupt Controller

12.2.1. Overview

The Nested Vectored Interrupt Controller (NVIC) in the SAM L21 supports 32 interrupt lines with four different priority levels. For more details, refer to the Cortex-M0+ Technical Reference Manual (http://www.arm.com).

12.2.2. Interrupt Line Mapping

Each of the 28 interrupt lines is connected to one peripheral instance, as shown in the table below. Each peripheral can have one or more interrupt flags, located in the peripheral's Interrupt Flag Status and Clear (INTFLAG) register.

An interrupt flag is set when the interrupt condition occurs. Each interrupt in the peripheral can be individually enabled by writing a 1 to the corresponding bit in the peripheral's Interrupt Enable Set (INTENSET) register, and disabled by writing 1 to the corresponding bit in the peripheral's Interrupt Enable Clear (INTENCLR) register.

An interrupt request is generated from the peripheral when the interrupt flag is set and the corresponding interrupt is enabled.

The interrupt requests for one peripheral are ORed together on system level, generating one interrupt request for each peripheral. An interrupt request will set the corresponding interrupt pending bit in the NVIC interrupt pending registers (SETPEND/CLRPEND bits in ISPR/ICPR).

For the NVIC to activate the interrupt, it must be enabled in the NVIC interrupt enable register (SETENA/CLRENA bits in ISER/ICER). The NVIC interrupt priority registers IPR0-IPR7 provide a priority field for each interrupt.

Table 12-3. Interrupt Line Mapping

Peripheral source	NVIC line
EIC NMI – External Interrupt Controller	NMI
PM – Power Manager MCLK - Main Clock OSCCTRL - Oscillators Controller OSC32KCTRL - 32KHz Oscillators Controller SUPC - Supply Controller PAC - Protecion Access Controller	0
WDT – Watchdog Timer	1
RTC – Real Time Counter	2
EIC – External Interrupt Controller	3
NVMCTRL – Non-Volatile Memory Controller	4
DMAC - Direct Memory Access Controller	5
USB - Universal Serial Bus	6
EVSYS – Event System	7

Peripheral source	NVIC line
SERCOM0 – Serial Communication Interface 0	8
SERCOM1 – Serial Communication Interface 1	9
SERCOM2 – Serial Communication Interface 2	10
SERCOM3 – Serial Communication Interface 3	11
SERCOM4 – Serial Communication Interface 4	12
SERCOM5 – Serial Communication Interface 5	13
TCC0 – Timer Counter for Control 0	14
TCC1 – Timer Counter for Control 1	15
TCC2 – Timer Counter for Control 2	16
TC0 – Timer Counter 0	17
TC1 – Timer Counter 1	18
TC2 – Timer Counter 2	19
TC3 – Timer Counter 3	20
TC4 – Timer Counter 4	21
ADC – Analog-to-Digital Converter	22
AC – Analog Comparator	23
DAC – Digital-to-Analog Converter	24
PTC – Peripheral Touch Controller	25
AES - Advanced Encrytpion Standard module	26
TRNG - True Random Number Generator	27

12.3. Micro Trace Buffer

12.3.1. Features

- Program flow tracing for the Cortex-M0+ processor
- MTB SRAM can be used for both trace and general purpose storage by the processor
- The position and size of the trace buffer in SRAM is configurable by software
- CoreSight compliant

12.3.2. Overview

When enabled, the MTB records the changes in program flow that are reported by the Cortex-M0+ processor over the execution trace interface. This interface is shared between the Cortex-M0+ processor and the CoreSight MTB-M0+. The information is stored by the MTB in the SRAM as trace packets. An off-chip debugger can extract the trace information using the Debug Access Port to read the trace information from the SRAM. The debugger can then reconstruct the program flow from this information.

The MTB stores trace information into the SRAM and gives the processor access to the SRAM simultaneously. The MTB ensures that trace write accesses have priority over processor accesses.

An execution trace packet consists of a pair of 32-bit words that the MTB generates when it detects a non-sequential change of the program pounter (PC) value. A non-sequential PC change can occur during branch instructions or during exception entry. See the CoreSight MTB-M0+ Technical Reference Manual for more details on the MTB execution trace packet format.

Tracing is enabled when the MASTER.EN bit in the Master Trace Control Register is 1. There are various ways to set the bit to 1 to start tracing, or to 0 to stop tracing. See the CoreSight Cortex-M0+ Technical Reference Manual for more details on the Trace start and stop and for a detailed description of the MTB's MASTER register. The MTB can be programmed to stop tracing automatically when the memory fills to a specified watermark level or to start or stop tracing by writing directly to the MASTER.EN bit. If the watermark mechanism is not being used and the trace buffer overflows, then the buffer wraps around overwriting previous trace packets.

The base address of the MTB registers is 0x41006000; this address is also written in the CoreSight ROM Table. The offset of each register from the base address is fixed and as defined by the CoreSight MTB-M0+ Technical Reference Manual. The MTB has four programmable registers to control the behavior of the trace features:

- POSITION: Contains the trace write pointer and the wrap bit
- MASTER: Contains the main trace enable bit and other trace control fields
- FLOW: Contains the WATERMARK address and the AUTOSTOP and AUTOHALT control bits
- BASE: Indicates where the SRAM is located in the processor memory map. This register is provided to enable auto discovery of the MTB SRAM location by a debug agent

See the CoreSight MTB-M0+ Technical Reference Manual for a detailed description of these registers.

12.4. High-Speed Bus System

12.4.1. Features

High-Speed Bus Matrix has the following features:

- Symmetric crossbar bus switch implementation
- Allows concurrent accesses from different masters to different slaves
- 32-bit data bus
- Operation at a one-to-one clock frequency with the bus masters

H2LBRIDGE has the following features:

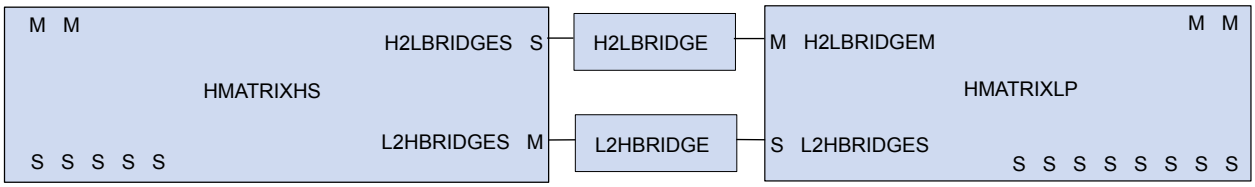
- LP clock division support
- Write: Posted-write FIFO of 3 words, no bus stall until it is full
- Write: 1 cycle bus stall when full when LP clock is not divided
- 2 stall cycles on read when LP clock is not divided
- Ultra low latency mode:
  - Suitable when the HS clock frequency is not above half the maximum device clock frequency
  - Removes all intrinsic bridge stall cycles (except those needed for LP clock ratio adaptation)
  - Enabled by writing a '1' in 0x41008120 using a 32-bit write access

L2HBRIDGE has the following features:

- LP clock division support
- Write: Posted-write FIFO of 1 word, no bus stall until it is full
- Write: 1 cycle bus stall when full when LP clock is not divided
- 2 stall cycles on read when LP clock is not divided

- ultra low latency mode:
  - Suitable when the HS clock frequency is not above half the maximum device clock frequency
  - Removes all intrinsic bridge stall cycles (except those needed for LP clock ratio adaptation)
  - Enabled by writing a '1' in 0x41008120 using a 32-bit write access

Figure 12-1. High-Speed Bus System Components



12.4.2. Configuration

Figure 12-2. Master-Slave Relations High-Speed Bus Matrix

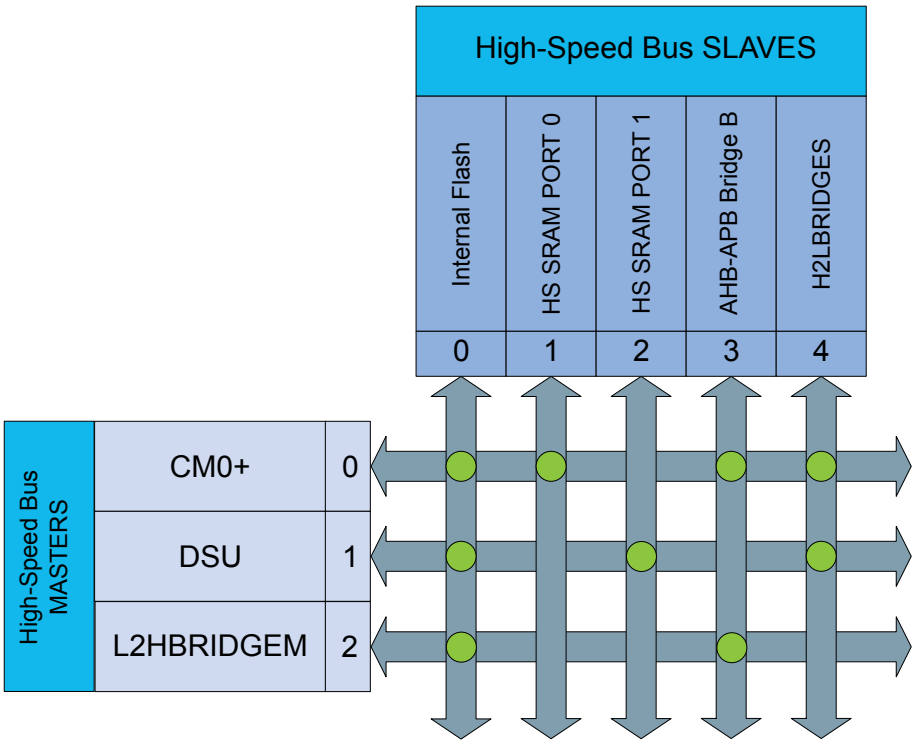


Figure 12-3. Master-Slave Relations Low-Power Bus Matrix

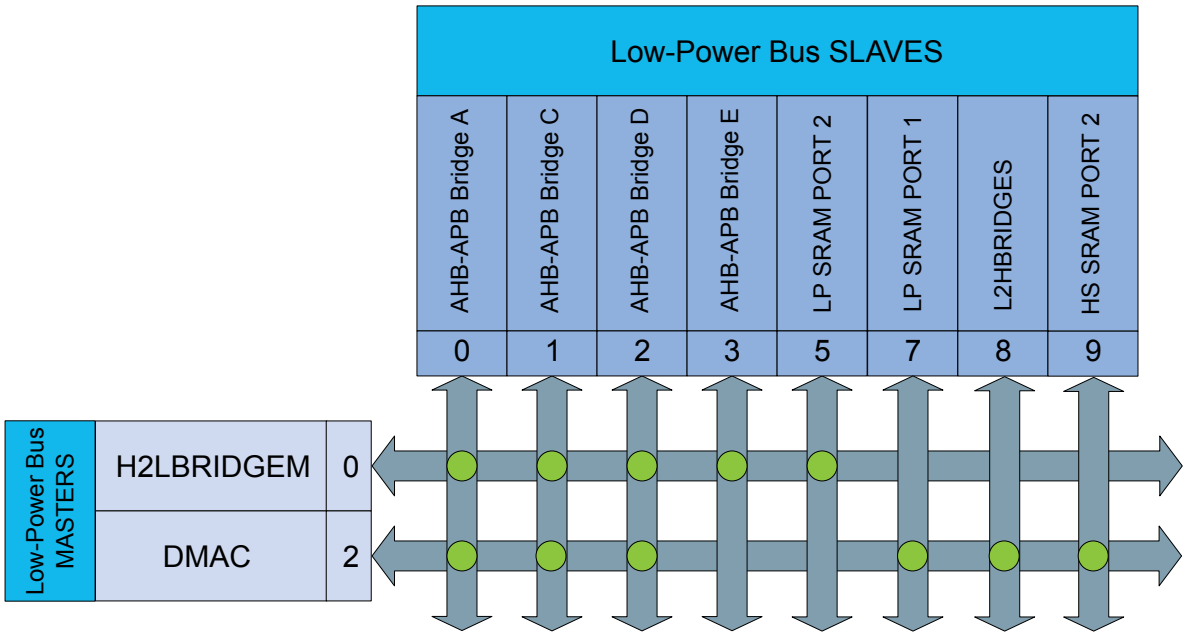


Table 12-4. High-Speed Bus Matrix Masters

High-Speed Bus Matrix Masters	Master ID
CM0+ - Cortex M0+ Processor	0
DSU - Device Service Unit	1
L2HBRIDGEM - Low-Power to High-Speed bus matrix AHB to AHB bridge	2

Table 12-5. High-Speed Bus Matrix Slaves

High-Speed Bus Matrix Slaves	Slave ID
Internal Flash Memory	0
HS SRAM Port 0 - CM0+ Access	1
HS SRAM Port 1 - DSU Access	2
AHB-APB Bridge B	3
H2LBRIDGES - High-Speed to Low-Power bus matrix AHB to AHB bridge	4

Table 12-6. Low-Power Bus Matrix Masters

Low-Power Bus Matrix Masters	Master ID
H2LBRIDGEM - High-Speed to Low-Power bus matrix AHB to AHB bridge	0
DMAC - Direct Memory Access Controller - Data Access	2

Table 12-7. Low-Power Bus Matrix Slaves

Low-Power Bus Matrix Slaves	Slave ID
AHB-APB Bridge A	0
AHB-APB Bridge C	1
AHB-APB Bridge D	2
AHB-APB Bridge E	3
LP SRAM Port 2- H2LBRIDGEM access	5
LP SRAM Port 1- DMAC access	7
L2HBRIDGES - Low-Power to High-Speed bus matrix AHB to AHB bridge	8
HS SRAM Port 2- HMATRIXLP access	9

12.4.3. SRAM Quality of Service

To ensure that masters with latency requirements get sufficient priority when accessing RAM, priority levels can be assigned to the masters for different types of access.

The Quality of Service (QoS) level is independently selected for each master accessing the RAM. For any access to the RAM, the RAM also receives the QoS level. The QoS levels and their corresponding bit values for the QoS level configuration are shown in the following table.

Table 12-8. Quality of Service

Value	Name	Description
0x0	DISABLE	Background (no sensitive operation)
0x1	LOW	Sensitive Bandwidth
0x2	MEDIUM	Sensitive Latency
0x3	HIGH	Critical Latency

If a master is configured with QoS level DISABLE (0x0) or LOW (0x1) there will be a minimum latency of one cycle for the RAM access.

The priority order for concurrent accesses are decided by two factors. First, the QoS level for the master and second, a static priority given by the port ID. The lowest port ID has the highest static priority. See the tables below for details.

The MTB has a fixed QoS level HIGH (0x3).

The CPU QoS level can be written/read, using 32-bit access only, at address 0x41008114 bits [1:0]. Its reset value is 0x3.

Refer to different master QOSCTRL registers for configuring QoS for the other masters (USB, DMAC).

Table 12-9. HS SRAM Port Connections QoS

HS SRAM Port Connection	Port ID	Connection Type	QoS	default QoS
MTB - Micro Trace Buffer	4	Direct	STATIC-3	0x3
USB - Universal Serial Bus	3	Direct	IP-QOSCTRL	0x3
HMATRIXLP - Low-Power Bus Matrix	2	Bus Matrix	0x44000934(1), bits[1:0]	0x2
DSU - Device Service Unit	1	Bus Matrix	0x4100201C(1)	0x2
CM0+ - Cortex M0+ Processor	0	Bus Matrix	0x41008114(1), bits[1:0]	0x3

Note:

1. Using 32-bit access only.

Table 12-10. LP SRAM Port Connections QoS

LP SRAM Port Connection	Port ID	Connection Type	QoS	default QoS
DMAC - Direct Memory Access Controller - Write-Back Access	5, 6	Direct	IP-QOSCTRL.WRBQOS	0x2
DMAC - Direct Memory Access Controller - Fetch Access	3, 4	Direct	IP-QOSCTRL.FQOS	0x2
H2LBRIDGEM - HS to LP bus matrix AHB to AHB bridge	2	Bus Matrix	0x44000924(1), bits[1:0]	0x2
DMAC - Direct Memory Access Controller - Data Access	1	Bus Matrix	IP-QOSCTRL.DQOS	0x2

Note:

1. Using 32-bit access only.

13. PAC - Peripheral Access Controller

13.1. Overview

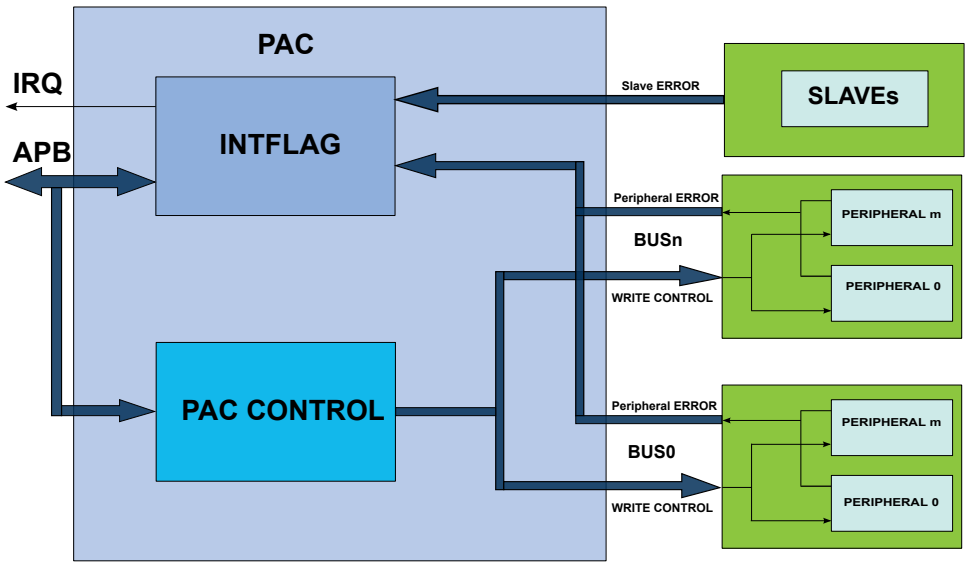
The Peripheral Access Controller provides an interface for the locking and unlocking of peripheral registers within the device. It reports all violations that could happen when accessing a peripheral: write protected access, illegal access, enable protected access, access when clock synchronization or software reset is on-going. These errors are reported in a unique interrupt flag for a peripheral. The PAC module also reports errors occurring at the slave bus level, when an access to a non-existing address is detected.

13.2. Features

- Manages write protection access and reports access errors for the peripheral modules or bridges

13.3. Block Diagram

Figure 13-1. PAC Block Diagram



13.4. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

13.4.1. IO Lines

Not applicable.

13.4.2. Power Management

The PAC can continue to operate in any sleep mode where the selected source clock is running. The PAC interrupts can be used to wake up the device from sleep modes. The events can trigger other operations in the system without exiting sleep modes.

Related Links

PM – Power Manager on page 186

13.4.3. Clocks

The PAC bus clock (CLK_PAC_APB) can be enabled and disabled in the Main Clock module. The default state of CLK_PAC_APB can be found in the related links.

Related Links

MCLK – Main Clock on page 148

Peripheral Clock Masking on page 151

13.4.4. DMA

Not applicable.

13.4.5. Interrupts

The interrupt request line is connected to the Interrupt Controller. Using the PAC interrupt requires the Interrupt Controller to be configured first.

Table 13-1. Interrupt Lines

Instances	NVIC Line
PAC	PACERR

Related Links

Nested Vector Interrupt Controller on page 51

13.4.6. Events

The events are connected to the Event System, which may need configuration.

Related Links

EVSYS – Event System on page 536

13.4.7. Debug Operation

When the CPU is halted in debug mode, write protection of all peripherals is disabled and the PAC continues normal operation.

13.4.8. Register Access Protection

All registers with write-access can be write-protected optionally by the Peripheral Access Controller (PAC), except the following registers:

- Write Control (WRCTRL) register
- AHB Slave Bus Interrupt Flag Status and Clear (INTFLAGAHB) register
- Peripheral Interrupt Flag Status and Clear n (INTFLAG A/B/C...) registers

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

PAC write-protection does not apply to accesses through an external debugger.

13.5. Functional Description

13.5.1. Principle of Operation

The Peripheral Access Control module allows the user to set a write protection on peripheral modules and generate an interrupt in case of a peripheral access violation. The peripheral's protection can be set,

cleared or locked for user convenience. A set of Interrupt Flag and Status registers informs the user on the status of the violation in the peripherals. In addition, slaves bus errors can be also reported in the cases where reserved area is accessed by the application.

13.5.2. Basic Operation

13.5.2.1. Initialization

After reset, the PAC is enabled.

13.5.2.2. Enabling and Resetting

The PAC is always enabled after reset.

Only a hardware reset will reset the PAC module.

13.5.2.3. Operations

The PAC module allows the user to set, clear or lock the write protection status of all peripherals on all Peripheral Bridges.

If a peripheral register violation occurs, the Peripheral Interrupt Flag n registers (INTFLAGn) are updated to inform the user on the status of the violation in the peripherals connected to the Peripheral Bridge n (n = A,B,C ...). The corresponding Peripheral Write Control Status n register (STATUSn) gives the state of the write protection for all peripherals connected to the corresponding Peripheral Bridge n. Refer to the Peripheral Access Errors on page 60 for details.

The PAC module reports also the errors occurring at slave bus level when an access to reserved area is detected. AHB Slave Bus Interrupt Flag register (INTFLAGAHB) informs the user on the status of the violation in the corresponding slave. Refer to the AHB Slave Bus Errors on page 61 for details.

13.5.2.4. Peripheral Access Errors

The following events will generate a Peripheral Access Error:

- Protected write: To avoid unexpected writes to a peripheral's registers, each peripheral can be write protected. Only the registers denoted as “Write Protected” in the module’s datasheet will be protected. If a peripheral is not write protected, write data accesses are performed normally. If a peripheral is write protected and if a write access is attempted, data will not be written and peripheral returns an access error. The corresponding interrupt flag bit in the INTFLAGn register will be set.
- Illegal access: Access to an unimplemented register within the module.
- Synchronized write error: For write-synchronized registers an error will be reported if the register is written while a synchronization is ongoing.

When any of the INTFLAGn registers bit are set, an interrupt will be requested if the PAC interrupt enable bit is set.

Related Links

Register Synchronization on page 127

13.5.2.5. Write Access Protection Management

Peripheral access control can be enabled or disabled by writing to the WRCTRL register.

The data written to the WRCTRL register is composed of two fields; WRCTRL.PERID and WRCTRL.KEY. The WRCTRL.PERID is an unique identifier corresponding to a peripheral. The WRCTRL.KEY is a key value that defines the operation to be done on the control access bit. These operations can be “clear protection”, “set protection” and “set and lock protection bit”.

The “clear protection” operation will remove the write access protection for the peripheral selected by WRCTRL.PERID. Write accesses are allowed for the registers in this peripheral.

The “set protection” operation will set the write access protection for the peripheral selected by WRCTRL.PERID. Write accesses are not allowed for the registers with write protection property in this peripheral.

The “set and lock protection” operation will permanently set the write access protection for the peripheral selected by WRCTRL.PERID. The write access protection will only be cleared by a hardware reset.

The peripheral access control status can be read from the corresponding STATUSn register.

13.5.2.6. Write Access Protection Management Errors

Only word-wise writes to the WRCTRL register will effectively change the access protection. Other type of accesses will have no effect and will cause a PAC write access error. This error is reported in the INTFLAGn.PAC bit corresponding to the PAC module.

PAC also offers an additional safety feature for correct program execution with an interrupt generated on double write clear protection or double write set protection. If a peripheral is write protected and a subsequent set protection operation is detected then the PAC returns an error, and similarly for a double clear protection operation. In addition, an error is generated when writing a “set and lock” protection to a write-protected peripheral or when a write access is done to a locked set protection.

This can be used to ensure that the application follows the intended program flow by always following a write protect with an unprotect and conversely. However in applications where a write protected peripheral is used in several contexts, e.g. interrupt, care should be taken so that either the interrupt can not happen while the main application or other interrupt levels manipulates the write protection status or when the interrupt handler needs to unprotect the peripheral based on the current protection status by reading the STATUS register.

The errors generated while accessing the PAC module registers (eg. key error, double protect error...) will set the INTFLAGn.PAC flag.

13.5.2.7. AHB Slave Bus Errors

The PAC module reports errors occurring at the Slave bus level. These errors are generated when an access is performed at an address where no slave (bridge or peripheral) is mapped. These errors are reported in the INTFLAGAHB register.

13.5.2.8. Generating Events

The PAC module can also generate an event when any of the Interrupt Flag registers bit are set. To enable the PAC event generation, the control bit EVCTRL.ERREO must be set.

13.5.3. DMA Operation

Not applicable.

13.5.4. Interrupts

The PAC has the following interrupt source:

- Error (ERR): Indicates that a peripheral access violation occurred in one of the peripherals controlled by the PAC module, or a bridge error occurred in one of the bridges reported by the PAC
  - This interrupt is a synchronous wake-up source.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAGAHB and INTFLAGn) registers is set when the interrupt condition occurs. Each interrupt can be individually enabled by writing a one to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a one to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register. An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the PAC is reset. All interrupt requests from the peripheral are ORed together

on system level to generate one combined interrupt request to the NVIC. The user must read the INTFLAGAHB and INTFLAGn registers to determine which interrupt condition is present.

Note that interrupts must be globally enabled for interrupt requests to be generated.

Related Links

Nested Vector Interrupt Controller on page 51
Sleep Mode Controller on page 191

13.5.5. Events

The PAC can generate the following output event:

- Error (ERR): Generated when one of the interrupt flag registers bits is set

Writing a one to an Event Output bit in the Event Control Register (EVCTRL.ERREO) enables the corresponding output event. Writing a zero to this bit disables the corresponding output event.

13.5.6. Sleep Mode Operation

In Sleep mode, the PAC is kept enabled if an available master (CPU, DMA) is running. The PAC will continue to catch access errors from module and generate interrupts or events.

13.5.7. Synchronization

Not applicable.

13.6. Register Summary

Offset	Name	Bit Pos.								
0x00	WRCTRL	7:0	PERID[7:0]							
0x01		15:8	PERID[15:8]							
0x02		23:16	KEY[7:0]							
0x03		31:24								
0x04	EVCTRL	7:0								ERREO
0x05 ... 0x07	Reserved									
0x08	INTENCLR	7:0								ERR
0x09	INTENSET	7:0								ERR
0x0A ... 0x0F	Reserved									
0x10	INTFLAGAHB	7:0				H2LBRIDGES	HPB1	HSRAMDSU	HSRAMCM0P	FLASH
0x11		15:8								
0x12		23:16	LPRAMDMAC		LPRAMHS		HPB4	HPB3	HPB2	HPB0
0x13		31:24							HSRAMLP	L2HBRIDGES
0x14	INTFLAGA	7:0	WDT	GCLK	SUPC	OSC32KCTRL	OSCCTRL	RSTC	MCLK	PM
0x15		15:8				DSUSTANDBY		PORT	EIC	RTC
0x16		23:16								
0x17		31:24								
0x18	INTFLAGB	7:0				HMATRIXHS	MTB	NVMCTRL	DSU	USB
0x19		15:8								
0x1A		23:16								
0x1B		31:24								
0x1C	INTFLAGC	7:0	TCC2	TCC1	TCC0	SERCOM4	SERCOM3	SERCOM2	SERCOM1	SERCOM0
0x1D		15:8		TRNG	AES	DAC	TC3	TC2	TC1	TC0
0x1E		23:16								
0x1F		31:24								
0x20	INTFLAGD	7:0	CCL	OPAMP	PTC	AC	ADC	TC4	SERCOM5	EVSYS
0x21		15:8								
0x22		23:16								
0x23		31:24								
0x24	INTFLAGE	7:0						HMATRIXLP	DMAC	PAC
0x25		15:8								
0x26		23:16								
0x27		31:24								
0x28 ... 0x33	Reserved									

Offset	Name	Bit Pos.								
0x34	STATUSA	7:0	WDT	GCLK	SUPC	OSC32KCTRL	OSCCTRL	RSTC	MCLK	PM
0x35		15:8				DSUSTANDBY		PORT	EIC	RTC
0x36		23:16								
0x37		31:24								
0x38	STATUSB	7:0				HMATRIXHS	MTB	NVMCTRL	DSU	USB
0x39		15:8								
0x3A		23:16								
0x3B		31:24								
0x3C	STATUSC	7:0	TCC2	TCC1	TCC0	SERCOM4	SERCOM3	SERCOM2	SERCOM1	SERCOM0
0x3D		15:8		TRNG	AES	DAC	TC3	TC2	TC1	TC0
0x3E		23:16								
0x3F		31:24								
0x40	STATUSD	7:0	CCL	OPAMP	PTC	AC	ADC	TC4	SERCOM5	EVSYS
0x41		15:8								
0x42		23:16								
0x43		31:24								
0x44	STATUSE	7:0						HMATRIXLP	DMAC	PAC
0x45		15:8								
0x46		23:16								
0x47		31:24								

13.7. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to the related links.

Related Links

Register Synchronization on page 127

13.7.1. Write Control

Name: WRCTRL
Offset: 0x0
Reset: 0x00000000
Property: –

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	KEY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	PERID[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PERID[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – KEY[7:0]: Peripheral Access Control Key
These bits define the peripheral access control key:

Value	Name	Description
0x0	OFF	No action
0x1	CLEAR	Clear the peripheral write control
0x2	SET	Set the peripheral write control
0x3	LOCK	Set and lock until the next hardware reset the peripheral write control

Bits 15:0 – PERID[15:0]: Peripheral Identifier
The PERID represents the peripheral whose control is changed using the WRCTRL.KEY. The Peripheral Identifier is calculated following formula:

$$PERID = 32^{*}\,\mathrm{BridgeNumber} + \mathrm{N}$$

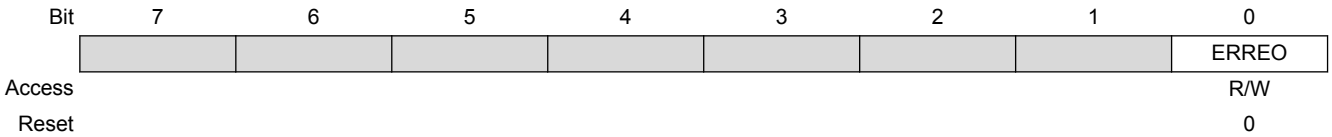
Where BridgeNumber represents the Peripheral Bridge Number (0 for Peripheral Bridge A, 1 for Peripheral Bridge B, etc). N represents the peripheral index from the respective Bridge Number:

Table 13-2. PERID Values

Periph. Bridge Name	BridgeNumber	PERID Values
A	0	0+N
B	1	32+N
C	2	64+N
D	3	96+N
E	4	128+N

13.7.2. Event Control

Name: EVCTRL
Offset: 0x04
Reset: 0x00
Property: PAC Write-Protection



Bit 0 – ERREO: Peripheral Access Error Event Output
This bit indicates if the Peripheral Access Error Event Output is enabled or not. When enabled, an event will be generated when one of the interrupt flag registers bits (INTFLAGAHB, INTFLAGn) is set:

Value	Description
0	Peripheral Access Error Event Output is disabled.
1	Peripheral Access Error Event Output is enabled.

13.7.3. Interrupt Enable Clear

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Name: INTENCLR
Offset: 0x08
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								ERR
Access								R/W
Reset								0

Bit 0 – ERR: Peripheral Access Error Interrupt Enable

This bit indicates that the Peripheral Access Error Interrupt is enabled and an interrupt request will be generated when one of the interrupt flag registers bits (INTFLAGAHB, INTFLAGn) is set:

Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Peripheral Access Error interrupt Enable bit and disables the corresponding interrupt request.

Value	Description
0	Peripheral Access Error interrupt is disabled.
1	Peripheral Access Error interrupt is enabled.

13.7.4. Interrupt Enable Set

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENCLR).

Name: INTENSET
Offset: 0x09
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								ERR
Access								R/W
Reset								0

Bit 0 – ERR: Peripheral Access Error Interrupt Enable

This bit indicates that the Peripheral Access Error Interrupt is enabled and an interrupt request will be generated when one of the interrupt flag registers bits (INTFLAGAHB, INTFLAGn) is set:

Writing a zero to this bit has no effect.

Writing a one to this bit will set the Peripheral Access Error interrupt Enable bit and enables the corresponding interrupt request.

Value	Description
0	Peripheral Access Error interrupt is disabled.
1	Peripheral Access Error interrupt is enabled.

13.7.5. AHB Slave Bus Interrupt Flag Status and Clear

This flag is cleared by writing a one to the flag.

This flag is set when an access error is detected by the SLAVE n, and will generate an interrupt request if INTENCLR/SET.ERR is one.

Writing a zero to this bit has no effect.

Writing a one to this bit will clear the corresponding INTFLAGAHB interrupt flag.

Name: INTFLAGAHB
Offset: 0x10
Reset: 0x000000
Property: –

Bit	31	30	29	28	27	26	25	24
							HSRAMLP	L2HBRIDGES
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
	LPRAMDMAC		LPRAMHS		HPB4	HPB3	HPB2	HPB0
Access	R/W		R/W		R/W	R/W	R/W	R/W
Reset	0		0		0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				H2LBRIDGES	HPB1	HSRAMDSU	HSRAMCM0P	FLASH
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit 25 – HSRAMLP: Interrupt Flag for SLAVE HSRAMLP

Bit 24 – L2HBRIDGES: Interrupt Flag for SLAVE L2HBRIDGES

Bit 23 – LPRAMDMAC: Interrupt Flag for SLAVE LPRAMDMAC

Bit 21 – LPRAMHS: Interrupt Flag for SLAVE LPRAMHS

Bit 19 – HPB4: Interrupt Flag for SLAVE HPB4

Bit 18 – HPB3: Interrupt Flag for SLAVE HPB3

Bit 17 – HPB2: Interrupt Flag for SLAVE HPB2

Bit 16 – HPB0: Interrupt Flag for SLAVE HPB0

Bit 4 – H2LBRIDGES: Interrupt Flag for SLAVE H2LBRIDGES

Bit 3 – HPB1: Interrupt Flag for SLAVE HPB1

Bit 2 – HSRAMDSU: Interrupt Flag for SLAVE HSRAMDSU

Bit 1 – HSRAMCM0P: Interrupt Flag for SLAVE HSRAMCM0P

Bit 0 – FLASH: Interrupt Flag for SLAVE FLASH

13.7.6. Peripheral Interrupt Flag Status and Clear A

This flag is cleared by writing a one to the flag.

This flag is set when a Peripheral Access Error occurs while accessing the peripheral associated with the respective INTFLAGA bit, and will generate an interrupt request if INTENCLR/SET.ERR is one.

Writing a zero to this bit has no effect.

Writing a one to this bit will clear the corresponding INTFLAGA interrupt flag.

Name: INTFLAGA
Offset: 0x14
Reset: 0x000000
Property: –

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				DSUSTANDBY		PORT	EIC	RTC
Access				R/W		R/W	R/W	R/W
Reset				0		0	0	0

Bit	7	6	5	4	3	2	1	0
	WDT	GCLK	SUPC	OSC32KCTRL	OSCCTRL	RSTC	MCLK	PM
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 12 – DSUSTANDBY: Interrupt Flag for DSUSTANDBY

Bit 10 – PORT: Interrupt Flag for PORT

Bit 9 – EIC: Interrupt Flag for EIC

Bit 8 – RTC: Interrupt Flag for RTC

Bit 7 – WDT: Interrupt Flag for WDT

Bit 6 – GCLK: Interrupt Flag for GCLK

Bit 5 – SUPC: Interrupt Flag for SUPC

Bit 4 – OSC32KCTRL: Interrupt Flag for OSC32KCTRL

Bit 3 – OSCCTRL: Interrupt Flag for OSCCTRL

Bit 2 – RSTC: Interrupt Flag for RSTC

Bit 1 – MCLK: Interrupt Flag for MCLK

Bit 0 – PM: Interrupt Flag for PM

13.7.7. Peripheral Interrupt Flag Status and Clear B

This flag is cleared by writing a one to the flag.

This flag is set when a Peripheral Access Error occurs while accessing the peripheral associated with the respective INTFLAGB bit, and will generate an interrupt request if INTENCLR/SET.ERR is one.

Writing a zero to this bit has no effect.

Writing a one to this bit will clear the corresponding INTFLAGB interrupt flag.

Name: INTFLAGB
Offset: 0x18
Reset: 0x000000
Property: –

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				HMATRIXHS	MTB	NVMCTRL	DSU	USB
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit 4 – HMATRIXHS: Interrupt Flag for HMATRIXHS

Bit 3 – MTB: Interrupt Flag for MTB

Bit 2 – NVMCTRL: Interrupt Flag for NVMCTRL

Bit 1 – DSU: Interrupt Flag for DSU

Bit 0 – USB: Interrupt Flag for USB

13.7.8. Peripheral Interrupt Flag Status and Clear C

This flag is cleared by writing a one to the flag.

This flag is set when a Peripheral Access Error occurs while accessing the peripheral associated with the respective INTFLAGC bit, and will generate an interrupt request if INTENCLR/SET.ERR is one.

Writing a zero to this bit has no effect.

Writing a one to this bit will clear the corresponding INTFLAGC interrupt flag.

Name: INTFLAGC
Offset: 0x1C
Reset: 0x000000
Property: –

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		TRNG	AES	DAC	TC3	TC2	TC1	TC0
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TCC2	TCC1	TCC0	SERCOM4	SERCOM3	SERCOM2	SERCOM1	SERCOM0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 14 – TRNG: Interrupt Flag for TRNG

Bit 13 – AES: Interrupt Flag for AES

Bit 12 – DAC: Interrupt Flag for DAC

Bits 11:8 – TCn: Interrupt Flag for TCn [n = 3..0]

Bits 7:5 – TCCn: Interrupt Flag for TCCn [n = 2..0]

Bits 4:0 – SERCOMn: Interrupt Flag for SERCOMn [n = 4..0]

13.7.9. Peripheral Interrupt Flag Status and Clear D

This flag is cleared by writing a '1' to the flag.

This flag is set when a Peripheral Access Error occurs while accessing the peripheral associated with the respective INTFLAGD bit, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.

Writing a '0' to this bit has no effect.

writing a '1' to this bit will clear the corresponding INTFLAGD interrupt flag.

Name: INTFLAGD
Offset: 0x20
Reset: 0x000000
Property: –

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	CCL	OPAMP	PTC	AC	ADC	TC4	SERCOM5	EVSYS
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 7 – CCL: Interrupt Flag for CCL

Bit 6 – OPAMP: Interrupt Flag for OPAMP

Bit 5 – PTC: Interrupt Flag for PTC

Bit 4 – AC: Interrupt Flag for AC

Bit 3 – ADC: Interrupt Flag for ADC

Bit 2 – TC4: Interrupt Flag for TC4

Bit 1 – SERCOM5: Interrupt Flag for SERCOM5

Bit 0 – EVSYS: Interrupt Flag for EVSYS

13.7.10. Peripheral Interrupt Flag Status and Clear E

This flag is cleared by writing a one to the flag.

This flag is set when a Peripheral Access Error occurs while accessing the peripheral associated with the respective INTFLAGE bit, and will generate an interrupt request if INTENCLR/SET.ERR is one.

Writing a zero to this bit has no effect.

Writing a one to this bit will clear the corresponding INTFLAGE interrupt flag.

Name: INTFLAGE
Offset: 0x24
Reset: 0x000000
Property: –

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						HMATRIXLP	DMAC	PAC
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – HMATRIXLP: Interrupt Flag for HMATRIXLP

Bit 1 – DMAC: Interrupt Flag for DMAC

Bit 0 – PAC: Interrupt Flag for PAC

13.7.11. Peripheral Write Protection Status A

Writing to this register has no effect.

Reading STATUS register returns peripheral write protection status:

Value	Description
0	Peripheral is not write protected.
1	Peripheral is write protected.

Name: STATUSA
Offset: 0x34
Reset: 0x000000
Property: –

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				DSUSTANDBY		PORT	EIC	RTC
Access				R		R	R	R
Reset				0		0	0	0

Bit	7	6	5	4	3	2	1	0
	WDT	GCLK	SUPC	OSC32KCTRL	OSCCTRL	RSTC	MCLK	PM
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 12 – DSUSTANDBY: Peripheral DSUSTANDBY Write Protection Status

Bit 10 – PORT: Peripheral PORT Write Protection Status

Bit 9 – EIC: Peripheral EIC Write Protection Status

Bit 8 – RTC: Peripheral RTC Write Protection Status

Bit 7 – WDT: Peripheral WDT Write Protection Status

Bit 6 – GCLK: Peripheral GCLK Write Protection Status

Bit 5 – SUPC: Peripheral SUPC Write Protection Status

Bit 4 – OSC32KCTRL: Peripheral OSC32KCTRL Write Protection Status

Bit 3 – OSCCTRL: Peripheral OSCCTRL Write Protection Status

Bit 2 – RSTC: Peripheral RSTC Write Protection Status

Bit 1 – MCLK: Peripheral MCLK Write Protection Status

Bit 0 – PM: Peripheral ATW Write Protection Status

13.7.12. Peripheral Write Protection Status B

Writing to this register has no effect.

Reading STATUS register returns peripheral write protection status:

Value	Description
0	Peripheral is not write protected.
1	Peripheral is write protected.

Name: STATUSB
Offset: 0x38
Reset: 0x000000
Property: –

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				HMATRIXHS	MTB	NVMCTRL	DSU	USB
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit 4 – HMATRIXHS: Peripheral HMATRIXHS Write Protection Status

Bit 3 – MTB: Peripheral MTB Write Protection Status

Bit 2 – NVMCTRL: Peripheral NVMCTRLWrite Protection Status

Bit 1 – DSU: Peripheral DSU Write Protection Status

Bit 0 – USB: Peripheral USB Write Protection Status

13.7.13. Peripheral Write Protection Status C

Writing to this register has no effect.

Reading STATUS register returns peripheral write protection status:

Value	Description
0	Peripheral is not write protected.
1	Peripheral is write protected.

Name: STATUSC
Offset: 0x3C
Reset: 0x000000
Property: –

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		TRNG	AES	DAC	TC3	TC2	TC1	TC0
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TCC2	TCC1	TCC0	SERCOM4	SERCOM3	SERCOM2	SERCOM1	SERCOM0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 14 – TRNG: Peripheral TRNG Write Protection Status

Bit 13 – AES: Peripheral AES Write Protection Status

Bit 12 – DAC: Peripheral DAC Write Protection Status

Bits 11:8 – TCn: Peripheral TCn Write Protection Status [n = 3..0]

Bits 7:5 – TCCn: Peripheral TCCn Write Protection Status [n = 2..0]

Bits 4:0 – SERCOMn: Peripheral SERCOMn Write Protection Status [n = 4..0]

13.7.14. Peripheral Write Protection Status D

Writing to this register has no effect.

Reading STATUS register returns peripheral write protection status:

Value	Description
0	Peripheral is not write protected.
1	Peripheral is write protected.

Name: STATUSD
Offset: 0x40
Reset: 0x000000
Property: –

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	CCL	OPAMP	PTC	AC	ADC	TC4	SERCOM5	EVSYS
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 7 – CCL: Peripheral CCL Write Protection Status

Bit 6 – OPAMP: Peripheral OPAMP Write Protection Status

Bit 5 – PTC: Peripheral PTC Write Protection Status

Bit 4 – AC: Peripheral AC Write Protection Status

Bit 3 – ADC: Peripheral ADC Write Protection Status

Bit 2 – TC4: Peripheral TC4 Write Protection Status

Bit 1 – SERCOM5: Peripheral SERCOM5 Write Protection Status

Bit 0 – EVSYS: Peripheral EVSYS Write Protection Status

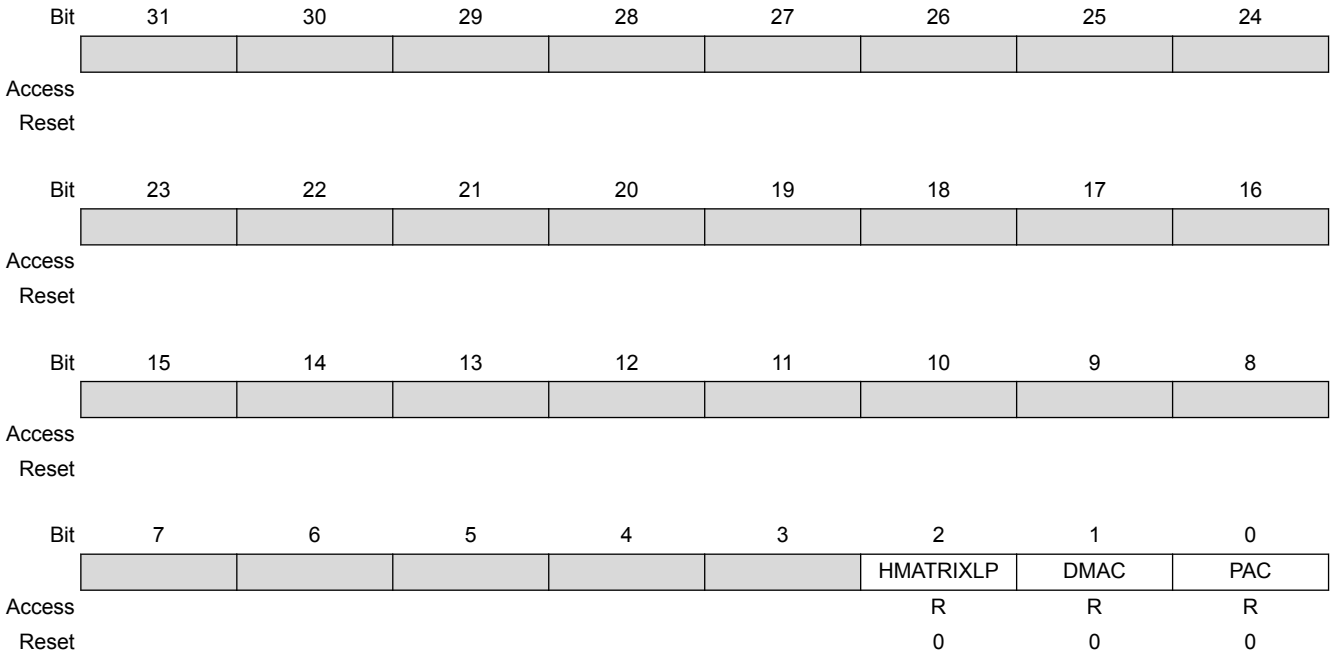
13.7.15. Peripheral Write Protection Status E

Writing to this register has no effect.

Reading STATUS register returns peripheral write protection status:

Value	Description
0	Peripheral is not write protected.
1	Peripheral is write protected.

Name: STATUSE
Offset: 0x44
Reset: 0x000000
Property: –



Bit 2 – HMATRIXLP: Peripheral HMATRIXLP Write Protection Status

Bit 1 – DMAC: Peripheral DMAC Write Protection Status

Bit 0 – PAC: Peripheral PAC Write Protection Status

14. Peripherals Configuration Summary

Table 14-1. Peripherals Configuration Summary

Peripheral name	Base address	IRQ line	AHB clock		APB clock		Bus Clock Domain	Generic Clock	PAC		Events		DMA		Power domain
			Index	Enabled at Reset	Index	Enabled at Reset	Name	Index	Index	Prot at Reset	User	Generator	Index	Sleep Walking	Name
AHB-APB Bridge A	0x40000000	—	0	Y	—	—	Low Power	—	—	—	—	—	—	N/A	PD1
PM	0x40000000	0	—	—	0	Y	Backup	—	0	N	—	—	—	N/A	PDBACKUP
MCLK	0x40000400	0	—	—	1	Y	Low Power	—	1	N	—	—	—	Y	PD0
RSTC	0x40000800	—	—	—	2	Y	Backup	—	2	N	—	—	—	N/A	PDBACKUP
OSCCTRL	0x40000C00	0	—	—	3	Y	Low Power	0: DFLL48M reference 1: FDPLL96M clk source 2: FDPLL96M 32kHz	3	N	—	—	—	Y	PD0
OSC32KCTRL	0x40001000	0	—	—	4	Y	Backup	—	4	N	—	—	—	—	PDBACKUP
SUPC	0x40001400	0	—	—	5	Y	Backup	—	5	N	—	—	—	N/A	PDBACKUP
GCLK	0x40001800	—	—	—	6	Y	Low Power	—	6	N	—	—	—	N/A	PD0
WDT	0x40001C00	1	—	—	7	Y	Low Power	—	7	N	—	—	—	Y	PDTOP
RTC	0x40002000	2	—	—	8	Y	Backup	—	8	N	—	1: CMP0/ ALARM0 2: CMP1 3: OVF 4-11: PER0-7	—	Y	PDBACKUP
EIC	0x40002400	3, NMI	—	—	9	Y	Low Power	3	9	N	—	12-27: EXTINT0-15	—	Y	PDTOP
PORT	0x40002800	—	—	—	10	Y	Low Power	—	10	N	0-3 : EV0-3	—	—	Y	PDTOP
AHB-APB Bridge B	0x41000000	—	1	Y	—	—	CPU	—	—	—	—	—	—	N/A	PD2
USB	0x41000000	6	12	Y	0	Y	CPU	4	0	N	—	—	—	Y	PD2
DSU	0x41002000	—	5	Y	1	Y	CPU	—	1	Y	—	—	—	N/A	PD2
NVMCTRL	0x41004000	4	8	Y	2	Y	CPU	—	2	N	—	—	—	Y	PD2
MTB	0x41006000	—	—	—	—	—	CPU	—	—	—	43,44: Start, Stop	—	—	N/A	PD2
AHB-APB Bridge C	0x42000000	—	2	Y	—	—	Low Power	—	—	—	—	—	—	N/A	PD1
SERCOM0	0x42000000	8	—	—	0	Y	Low Power	18: CORE 17: SLOW	0	N	—	—	1: RX 2: TX	Y	PD1
SERCOM1	0x42000400	9	—	—	1	Y	Low Power	19: CORE 17: SLOW	1	N	—	—	3: RX 4: TX	Y	PD1
SERCOM2	0x42000800	10	—	—	2	Y	Low Power	20: CORE 17: SLOW	2	N	—	—	5: RX 6: TX	Y	PD1

Peripheral name	Base address	IRQ line	AHB clock		APB clock		Bus Clock Domain	Generic Clock	PAC		Events		DMA		Power domain
			Index	Enabled at Reset	Index	Enabled at Reset	Name	Index	Index	Prot at Reset	User	Generator	Index	Sleep Walking	Name
SERCOM3	0x42000C00	11	—	—	3	Y	Low Power	21: CORE 17: SLOW	3	N	—	—	7: RX 8: TX	Y	PD1
SERCOM4	0x42001000	12	—	—	4	Y	Low Power	22: CORE 17: SLOW	4	N	—	—	9: RX 10: TX	Y	PD1
TCC0	0x42001400	14	—	—	5	Y	Low Power	25	5	N	12-13: EV0-1 14-17: MC0-3	36: OVF 37: TRG 38: CNT 39-42: MC0-3	11: OVF 12-15: MC0-3	Y	PD1
TCC1	0x42001800	15	—	—	6	Y	Low Power	25	6	N	18-19: EV0-1 20-21: MC0-1	43: OVF 44: TRG 45: CNT 46-47: MC0-1	16: OVF 17-18: MC0-1	Y	PD1
TCC2	0x42001C00	16	—	—	7	Y	Low Power	26	7	N	22-23: EV0-1 24-25: MC0-1	48: OVF 49: TRG 50: CNT 51-52: MC0-1	19: OVF 20-21: MC0-1	Y	PD1
TC0	0x42002000	17	—	—	8	Y	Low Power	27	8	N	26: EVU	53: OVF 54-55: MC0-1	22: OVF 23-24: MC0-1	Y	PD1
TC1	0x42002400	18	—	—	9	Y	Low Power	27	9	N	27: EVU	56: OVF 57-58: MC0-1	25: OVF 26-27: MC0-1	Y	PD1
TC2	0x42002800	19	—	—	10	Y	Low Power	28	10	N	28: EVU	59: OVF 60-61: MC0-1	28: OVF 29-30: MC0-1	Y	PD1
TC3	0x42002C00	20	—	—	11	Y	Low Power	28	11	N	29: EVU	62: OVF 63-64: MC0-1	31: OVF 32-33: MC0-1	Y	PD1
DAC	0x42003000	24	—	—	12	Y	Low Power	32	12	N	35-36: START0-1	73-74: EMPTY0-1	38-39: EMPTY0-1	Y	PD1
AES	0x42003400	26	—	—	13	Y	Low Power	—	13	N	—	—	44 : WR 45 : RD	Y	PD1
TRNG	0x42003800	27	—	—	14	Y	Low Power	—	14	N	—	77 : READY	—	Y	PD1
AHB-APB Bridge D	0x43000000	—	3	Y	—	—	Low Power	—	—	—	—	—	—	N/A	PD1
EVSYS	0x43000000	7	—	—	0	Y	Low Power	5-16: one per CHANNEL	0	N	—	—	—	Y	PD0
SERCOM5	0x43000400	13	—	—	1	Y	Low Power	24: CORE 23: SLOW	1	N	—	—	—	Y	PD0
TC4	0x43000800	21	—	—	2	Y	Low Power	29	2	N	—	65: OVF 66-67: MC0-1	34: OVF 35-36: MC0-1	Y	PD0
ADC	0x43000C00	22	—	—	3	Y	Low Power	30	3	N	31: START 32: SYNC	68: RESRDY 69: WINMON	37: RESRDY	Y	PD0

Peripheral name	Base address	IRQ line	AHB clock		APB clock		Bus Clock Domain	Generic Clock	PAC		Events		DMA		Power domain
			Index	Enabled at Reset	Index	Enabled at Reset	Name	Index	Index	Prot at Reset	User	Generator	Index	Sleep Walking	Name
AC	0x43001000	23	—	—	4	Y	Low Power	31	4	N	33-34: SOC0-1	70-71: COMP0-1 72: WIN0	—	Y	PD0
PTC	0x43001400	25	—	—	5	Y	Low Power	33	5	N	37: STCONV	75: EOC 76: WCOMP	—	—	PD0
OPAMP	0x43001800	—	—	—	6	Y	Low Power	—	6	N	—	—	—	Y	PD0
CCL	0x43001C00	—	—	—	7	Y	Low Power	34	7	N	38 : LUTIN0 39 : LUTIN1 40: LUTIN2 41: LUTIN3	78 : LUTOUT0 79 : LUTOUT1 80: LUTOUT2 81: LUTOUT3	—	Y	PD0
AHB-APB Bridge E	0x44000000	—	4	Y	—	—	Low Power	—	—	—	—	—	—	N/A	PD1
PAC	0x44000000	0	14	Y	0	Y	Low Power	—	0	—	—	82 : ACCERR	—	N/A	PD1
DMAC	0x44000400	—	11	Y	—	—	Low Power	—	1	—	4-11: CH0-7	28-35: CH0-7	—	Y	PD1

15. DSU - Device Service Unit

15.1. Overview

The Device Service Unit (DSU) provides a means to detect debugger probes. This enables the ARM Debug Access Port (DAP) to have control over multiplexed debug pads and CPU reset. The DSU also provides system-level services to debug adapters in an ARM debug system. It implements a CoreSight Debug ROM that provides device identification as well as identification of other debug components within the system. Hence, it complies with the ARM Peripheral Identification specification. The DSU also provides system services to applications that need memory testing, as required for IEC60730 Class B compliance, for example. The DSU can be accessed simultaneously by a debugger and the CPU, as it is connected on the High-Speed Bus Matrix. For security reasons, some of the DSU features will be limited or unavailable when the device is protected by the NVMCTRL security bit.

Related Links

NVMCTRL – Non-Volatile Memory Controller on page 483

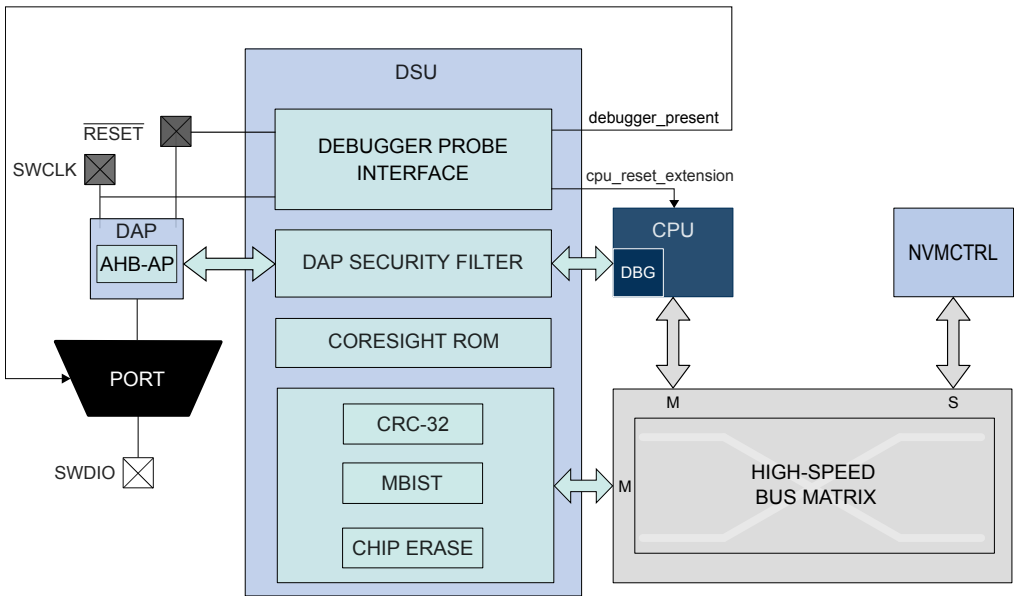
Security Bit on page 490

15.2. Features

- CPU reset extension
- Debugger probe detection (Cold- and Hot-Plugging)
- Chip-Erase command and status
- 32-bit cyclic redundancy check (CRC32) of any memory accessible through the bus matrix
- ARM® CoreSight™ compliant device identification
- Two debug communications channels
- Debug access port security filter
- Onboard memory built-in self-test (MBIST)

15.3. Block Diagram

Figure 15-1. DSU Block Diagram



15.4. Signal Description

The DSU uses three signals to function.

Signal Name	Type	Description
$\overline{\text{RESET}}$	Digital Input	External reset
SWCLK	Digital Input	SW clock
SWDIO	Digital I/O	SW bidirectional data pin

Related Links

I/O Multiplexing and Considerations on page 29

15.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

15.5.1. IO Lines

The SWCLK pin is by default assigned to the DSU module to allow debugger probe detection and to stretch the CPU reset phase. For more information, refer to Debugger Probe Detection on page 90. The Hot-Plugging feature depends on the PORT configuration. If the SWCLK pin function is changed in the PORT or if the PORT_MUX is disabled, the Hot-Plugging feature is disabled until a power-reset or an external reset.

15.5.2. Power Management

The DSU will continue to operate in any sleep mode where the selected source clock is running.

Related Links

PM – Power Manager on page 186

15.5.3. Clocks

The DSU bus clocks (CLK_DSU_APB and CLK_DSU_AHB) can be enabled and disabled by the Main Clock Controller.

Related Links

PM – Power Manager on page 186

MCLK – Main Clock on page 148

Peripheral Clock Masking on page 151

15.5.4. DMA

Not applicable.

15.5.5. Interrupts

Not applicable.

15.5.6. Events

Not applicable.

15.5.7. Register Access Protection

Registers with write-access can be optionally write-protected by the Peripheral Access Controller (PAC), except the following:

- Debug Communication Channel 0 register (DCC0)
- Debug Communication Channel 1 register (DCC1)

Note: Optional write-protection is indicated by the "PAC Write-Protection" property in the register description.

When the CPU is halted in debug mode, all write-protection is automatically disabled. Write-protection does not apply for accesses through an external debugger.

Related Links

PAC - Peripheral Access Controller on page 58

15.5.8. Analog Connections

Not applicable.

15.6. Debug Operation

15.6.1. Principle of Operation

The DSU provides basic services to allow on-chip debug using the ARM Debug Access Port and the ARM processor debug resources:

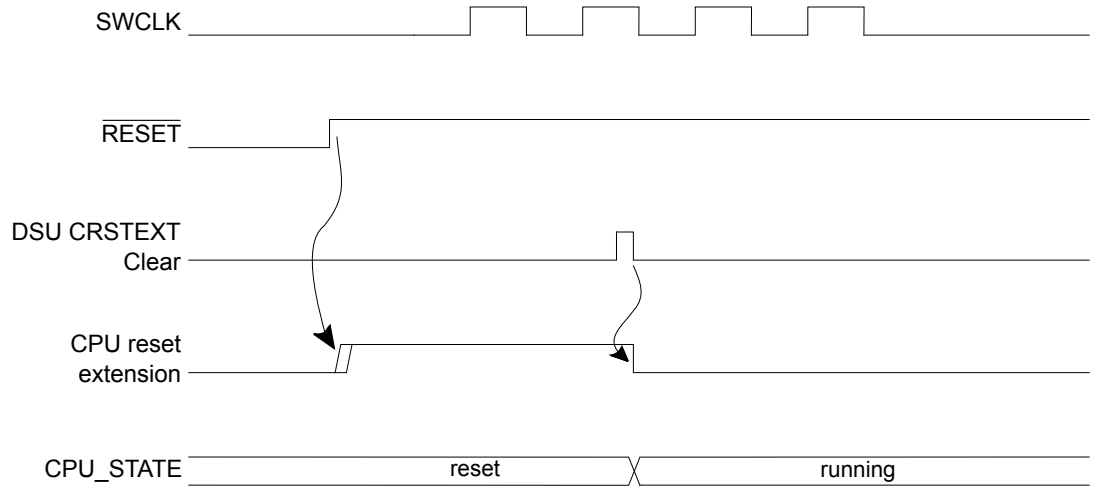
- CPU reset extension
- Debugger probe detection

For more details on the ARM debug components, refer to the ARM Debug Interface v5 Architecture Specification.

15.6.2. CPU Reset Extension

"CPU reset extension" refers to the extension of the reset phase of the CPU core after the external reset is released. This ensures that the CPU is not executing code at startup while a debugger connects to the system. It is detected on a $\overline{\text{RESET}}$ release event when SWCLK is low. At startup, SWCLK is internally pulled up to avoid false detection of a debugger if SWCLK is left unconnected. When the CPU is held in the reset extension phase, the CPU Reset Extension bit of the Status A register (STATUSA.CRSTEXT) is set. To release the CPU, write a '1' to STATUSA.CRSTEXT. STATUSA.CRSTEXT will then be set to zero. Writing a '0' to STATUSA.CRSTEXT has no effect. For security reasons, it is not possible to release the CPU reset extension when the device is protected by the NVMCTRL security bit. Trying to do so sets the Protection Error bit (PERR) of the Status A register (STATUSA.PERR).

Figure 15-2. Typical CPU Reset Extension Set and Clear Timing Diagram



Related Links

NVMCTRL – Non-Volatile Memory Controller on page 483

Security Bit on page 490

15.6.3. Debugger Probe Detection

15.6.3.1. Cold Plugging

Cold-Plugging is the detection of a debugger when the system is in reset. Cold-Plugging is detected when the CPU reset extension is requested, as described above.

15.6.3.2. Hot Plugging

Hot-Plugging is the detection of a debugger probe when the system is not in reset. Hot-Plugging is not possible under reset because the detector is reset when POR or $\overline{\text{RESET}}$ are asserted. Hot-Plugging is active when a SWCLK falling edge is detected. The SWCLK pad is multiplexed with other functions and the user must ensure that its default function is assigned to the debug system. If the SWCLK function is changed, the Hot-Plugging feature is disabled until a power-reset or external reset occurs. Availability of the Hot-Plugging feature can be read from the Hot-Plugging Enable bit of the Status B register (STATUSB.HPE).

Figure 15-3. Hot-Plugging Detection Timing Diagram

The presence of a debugger probe is detected when either Hot-Plugging or Cold-Plugging is detected. Once detected, the Debugger Present bit of the Status B register (STATUSB.DBGPRES) is set. For security reasons, Hot-Plugging is not available when the device is protected by the NVMCTRL security bit.

This detection requires that pads are correctly powered. Thus, at cold startup, this detection cannot be done until POR is released. If the device is protected, Cold-Plugging is the only way to detect a debugger probe, and so the external reset timing must be longer than the POR timing. If external reset is deasserted before POR release, the user must retry the procedure above until it gets connected to the device.

Related Links

NVMCTRL – Non-Volatile Memory Controller on page 483

Security Bit on page 490

15.7. Chip Erase

Chip-Erase consists of removing all sensitive information stored in the chip and clearing the NVMCTRL security bit. Therefore, all volatile memories and the Flash memory (including the EEPROM emulation area) will be erased. The Flash auxiliary rows, including the user row, will not be erased.

When the device is protected, the debugger must reset the device in order to be detected. This ensures that internal registers are reset after the protected state is removed. The Chip-Erase operation is triggered by writing a '1' to the Chip-Erase bit in the Control register (CTRL.CE). This command will be discarded if the DSU is protected by the Peripheral Access Controller (PAC). Once issued, the module clears volatile memories prior to erasing the Flash array. To ensure that the Chip-Erase operation is completed, check the Done bit of the Status A register (STATUSA.DONE).

The Chip-Erase operation depends on clocks and power management features that can be altered by the CPU. For that reason, it is recommended to issue a Chip- Erase after a Cold-Plugging procedure to ensure that the device is in a known and safe state.

The recommended sequence is as follows:

1. Issue the Cold-Plugging procedure (refer to Cold Plugging on page 90). The device then:
   1.1. Detects the debugger probe.
   1.2. Holds the CPU in reset.
2. Issue the Chip-Erase command by writing a '1' to CTRL.CE. The device then:
   2.1. Clears the system volatile memories.
   2.2. Erases the whole Flash array (including the EEPROM emulation area, not including auxiliary rows).

2.3. Erases the lock row, removing the NVMCTRL security bit protection.

3. Check for completion by polling STATUSA.DONE (read as one when completed).
4. Reset the device to let the NVMCTRL update fuses.

15.8. Programming

Programming the Flash or RAM memories is only possible when the device is not protected by the NVMCTRL security bit. The programming procedure is as follows:

1. At power up, $\overline{\text{RESET}}$ is driven low by a debugger. The on-chip regulator holds the system in a POR state until the input supply is above the POR threshold. The system continues to be held in this static state until the internally regulated supplies have reached a safe operating state.
2. The PM starts, clocks are switched to the slow clock (Core Clock, System Clock, Flash Clock and any Bus Clocks that do not have clock gate control). Internal resets are maintained due to the external reset.
3. The debugger maintains a low level on SWCLK. RESET is released, resulting in a debugger Cold-Plugging procedure.
4. The debugger generates a clock signal on the SWCLK pin, the Debug Access Port (DAP) receives a clock.
5. The CPU remains in Reset due to the Cold-Plugging procedure; meanwhile, the rest of the system is released.
6. A Chip-Erase is issued to ensure that the Flash is fully erased prior to programming.
7. Programming is available through the AHB-AP.
8. After the operation is completed, the chip can be restarted either by asserting $\overline{\text{RESET}}$, toggling power, or writing a '1' to the Status A register CPU Reset Phase Extension bit (STATUSA.CRSTEXT). Make sure that the SWCLK pin is high when releasing $\overline{\text{RESET}}$ to prevent extending the CPU reset.

Related Links

Electrical Characteristics on page 1140
NVMCTRL – Non-Volatile Memory Controller on page 483
Security Bit on page 490

15.9. Intellectual Property Protection

Intellectual property protection consists of restricting access to internal memories from external tools when the device is protected, and this is accomplished by setting the NVMCTRL security bit. This protected state can be removed by issuing a Chip-Erase (refer to Chip Erase on page 91). When the device is protected, read/write accesses using the AHB-AP are limited to the DSU address range and DSU commands are restricted. When issuing a Chip-Erase, sensitive information is erased from volatile memory and Flash.

The DSU implements a security filter that monitors the AHB transactions generated by the ARM AHB-AP inside the DAP. If the device is protected, then AHB-AP read/write accesses outside the DSU external address range are discarded, causing an error response that sets the ARM AHB-AP sticky error bits (refer to the ARM Debug Interface v5 Architecture Specification on http://www.arm.com).

The DSU is intended to be accessed either:

- Internally from the CPU, without any limitation, even when the device is protected
- Externally from a debug adapter, with some restrictions when the device is protected

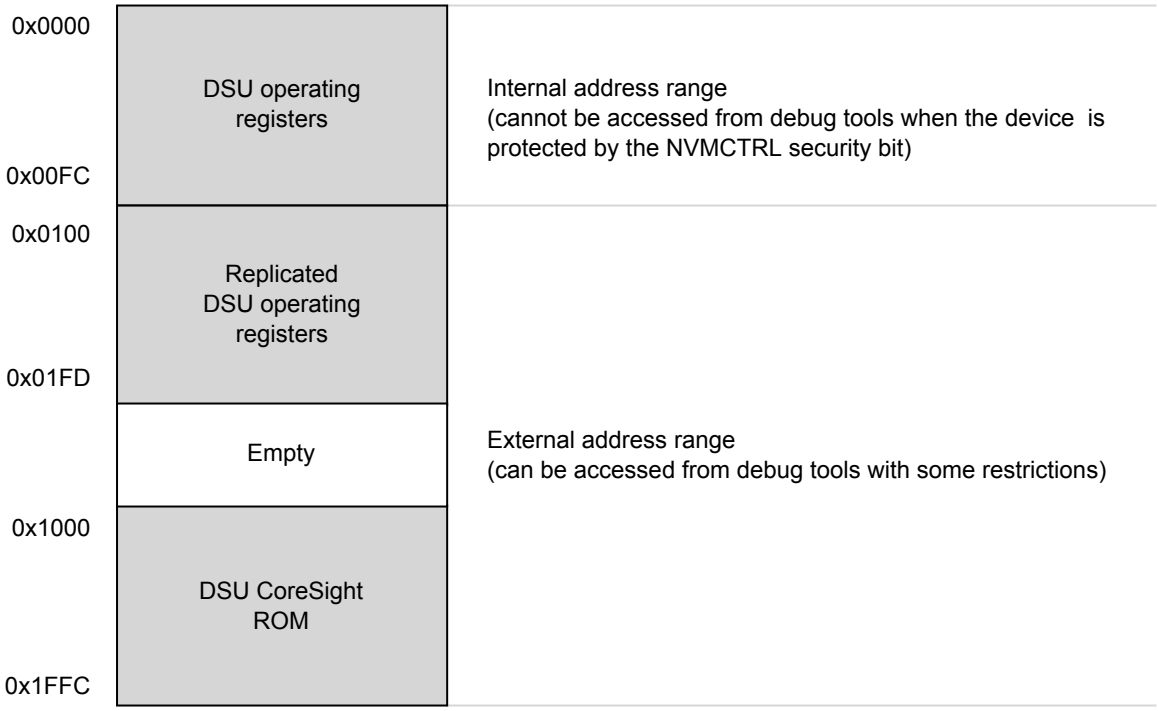
For security reasons, DSU features have limitations when used from a debug adapter. To differentiate external accesses from internal ones, the first 0x100 bytes of the DSU register map have been replicated at offset 0x100:

- The first 0x100 bytes form the internal address range
- The next 0x100 bytes form the external address range

When the device is protected, the DAP can only issue MEM-AP accesses in the DSU address range limited to the 0x100- 0x2000 offset range.

The DSU operating registers are located in the 0x00-0xFF area and remapped in 0x100-0x1FF to differentiate accesses coming from a debugger and the CPU. If the device is protected and an access is issued in the region 0x100-0x1FF, it is subject to security restrictions. For more information, refer to the Table 15-1 Feature Availability Under Protection on page 93.

Figure 15-4. APB Memory Mapping



Some features not activated by APB transactions are not available when the device is protected:

Table 15-1. Feature Availability Under Protection

Features	Availability when the device is protected
CPU Reset Extension	Yes
Clear CPU Reset Extension	No
Debugger Cold-Plugging	Yes
Debugger Hot-Plugging	No

Related Links

NVMCTRL – Non-Volatile Memory Controller on page 483
Security Bit on page 490

15.10. Device Identification

Device identification relies on the ARM CoreSight component identification scheme, which allows the chip to be identified as an Atmel device implementing a DSU. The DSU contains identification registers to differentiate the device.

15.10.1. CoreSight Identification

A system-level ARM CoreSight ROM table is present in the device to identify the vendor and the chip identification method. Its address is provided in the MEM-AP BASE register inside the ARM Debug Access Port. The CoreSight ROM implements a 64-bit conceptual ID composed as follows from the PID0 to PID7 CoreSight ROM Table registers:

Figure 15-5. Conceptual 64-bit Peripheral ID

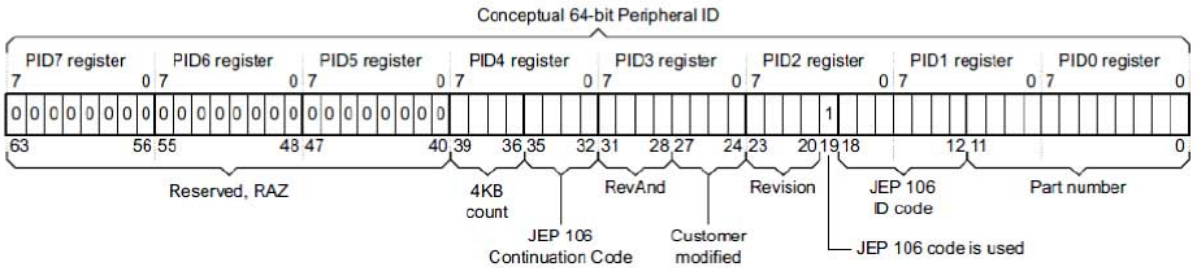


Table 15-2. Conceptual 64-Bit Peripheral ID Bit Descriptions

Field	Size	Description	Location
JEP-106 CC code	4	Atmel continuation code: 0x0	PID4
JEP-106 ID code	7	Atmel device ID: 0x1F	PID1+PID2
4KB count	4	Indicates that the CoreSight component is a ROM: 0x0	PID4
RevAnd	4	Not used; read as 0	PID3
CUSMOD	4	Not used; read as 0	PID3
PARTNUM	12	Contains 0xCD0 to indicate that DSU is present	PID0+PID1
REVISION	4	DSU revision (starts at 0x0 and increments by 1 at both major and minor revisions). Identifies DSU identification method variants. If 0x0, this indicates that device identification can be completed by reading the Device Identification register (DID)	PID3

For more information, refer to the ARM Debug Interface Version 5 Architecture Specification.

15.10.2. Chip Identification Method

The DSU DID register identifies the device by implementing the following information:

- Processor identification
- Product family identification
- Product series identification
- Device select

15.11. Functional Description

15.11.1. Principle of Operation

The DSU provides memory services such as CRC32 or MBIST that require almost the same interface. Hence, the Address, Length and Data registers (ADDR, LENGTH, DATA) are shared. These shared registers must be configured first; then a command can be issued by writing the Control register. When a command is ongoing, other commands are discarded until the current operation is completed. Hence, the user must wait for the STATUSA.DONE bit to be set prior to issuing another one.

15.11.2. Basic Operation

15.11.2.1. Initialization

The module is enabled by enabling its clocks. For more details, refer to Clocks on page 89. The DSU registers can be PAC write-protected.

Related Links

PAC - Peripheral Access Controller on page 58

15.11.2.2. Operation From a Debug Adapter

Debug adapters should access the DSU registers in the external address range 0x100 – 0x2000. If the device is protected by the NVMCTRL security bit, accessing the first 0x100 bytes causes the system to return an error. Refer to Intellectual Property Protection on page 92.

Related Links

NVMCTRL – Non-Volatile Memory Controller on page 483

Security Bit on page 490

15.11.2.3. Operation From the CPU

There are no restrictions when accessing DSU registers from the CPU. However, the user should access DSU registers in the internal address range (0x0 – 0x100) to avoid external security restrictions. Refer to Intellectual Property Protection on page 92.

15.11.3. 32-bit Cyclic Redundancy Check CRC32

The DSU unit provides support for calculating a cyclic redundancy check (CRC32) value for a memory area (including Flash and AHB RAM).

When the CRC32 command is issued from:

- The internal range, the CRC32 can be operated at any memory location
- The external range, the CRC32 operation is restricted; DATA, ADDR, and LENGTH values are forced (see below)

Table 15-3. AMOD Bit Descriptions when Operating CRC32

AMOD[1:0]	Short name	External range restrictions
0	ARRAY	CRC32 is restricted to the full Flash array area (EEPROM emulation area not included) DATA forced to 0xFFFFFFFF before calculation (no seed)
1	EEPROM	CRC32 of the whole EEPROM emulation area DATA forced to 0xFFFFFFFF before calculation (no seed)
2-3	Reserved	

The algorithm employed is the industry standard CRC32 algorithm using the generator polynomial 0xEDB88320 (reversed representation).

15.11.3.1. Starting CRC32 Calculation

CRC32 calculation for a memory range is started after writing the start address into the Address register (ADDR) and the size of the memory range into the Length register (LENGTH). Both must be word-aligned.

The initial value used for the CRC32 calculation must be written to the Data register (DATA). This value will usually be 0xFFFFFFFF, but can be, for example, the result of a previous CRC32 calculation if generating a common CRC32 of separate memory blocks.

Once completed, the calculated CRC32 value can be read out of the Data register. The read value must be complemented to match standard CRC32 implementations or kept non-inverted if used as starting point for subsequent CRC32 calculations.

If the device is in protected state by the NVMCTRL security bit, it is only possible to calculate the CRC32 of the whole flash array when operated from the external address space. In most cases, this area will be the entire onboard non-volatile memory. The Address, Length and Data registers will be forced to predefined values once the CRC32 operation is started, and values written by the user are ignored. This allows the user to verify the contents of a protected device.

The actual test is started by writing a '1' in the 32-bit Cyclic Redundancy Check bit of the Control register (CTRL.CRC). A running CRC32 operation can be canceled by resetting the module (writing '1' to CTRL.SWRST).

Related Links

NVMCTRL – Non-Volatile Memory Controller on page 483

Security Bit on page 490

15.11.3.2. Interpreting the Results

The user should monitor the Status A register. When the operation is completed, STATUSA.DONE is set. Then the Bus Error bit of the Status A register (STATUSA.BERR) must be read to ensure that no bus error occurred.

15.11.4. Debug Communication Channels

The Debug Communication Channels (DCCO and DCC1) consist of a pair of registers with associated handshake logic, accessible by both CPU and debugger even if the device is protected by the NVMCTRL security bit. The registers can be used to exchange data between the CPU and the debugger, during run time as well as in debug mode. This enables the user to build a custom debug protocol using only these registers.

The DCC0 and DCC1 registers are accessible when the protected state is active. When the device is protected, however, it is not possible to connect a debugger while the CPU is running (STATUSA.CRSTEXT is not writable and the CPU is held under Reset).

Two Debug Communication Channel status bits in the Status B registers (STATUS.DCCDx) indicate whether a new value has been written in DCC0 or DCC1. These bits, DCC0D and DCC1D, are located in the STATUSB registers. They are automatically set on write and cleared on read.

Note: The DCC0 and DCC1 registers are shared with the on-board memory testing logic (MBIST). Accordingly, DCC0 and DCC1 must not be used while performing MBIST operations.

Related Links

NVMCTRL – Non-Volatile Memory Controller on page 483

Security Bit on page 490

15.11.5. Testing of On-Board Memories MBIST

The DSU implements a feature for automatic testing of memory also known as MBIST (memory built-in self test). This is primarily intended for production test of on-board memories. MBIST cannot be operated from the external address range when the device is protected by the NVMCTRL security bit. If an MBIST command is issued when the device is protected, a protection error is reported in the Protection Error bit in the Status A register (STATUSA.PERR).

1. Algorithm
   The algorithm used for testing is a type of March algorithm called "March LR". This algorithm is able to detect a wide range of memory defects, while still keeping a linear run time. The algorithm is:
   1.1. Write entire memory to '0', in any order.
   1.2. Bit for bit read '0', write '1', in descending order.
   1.3. Bit for bit read '1', write '0', read '0', write '1', in ascending order.
   1.4. Bit for bit read '1', write '0', in ascending order.
   1.5. Bit for bit read '0', write '1', read '1', write '0', in ascending order.
   1.6. Read '0' from entire memory, in ascending order.

   The specific implementation used has a run time of O(14*n*), where *n* is the number of bits in the RAM. The detected faults are:
   - Address decoder faults
   - Stuck-at faults
   - Transition faults
   - Coupling faults
   - Linked Coupling faults
2. Starting MBIST
   To test a memory, you need to write the start address of the memory to the ADDR.ADDR bit field, and the size of the memory into the Length register.

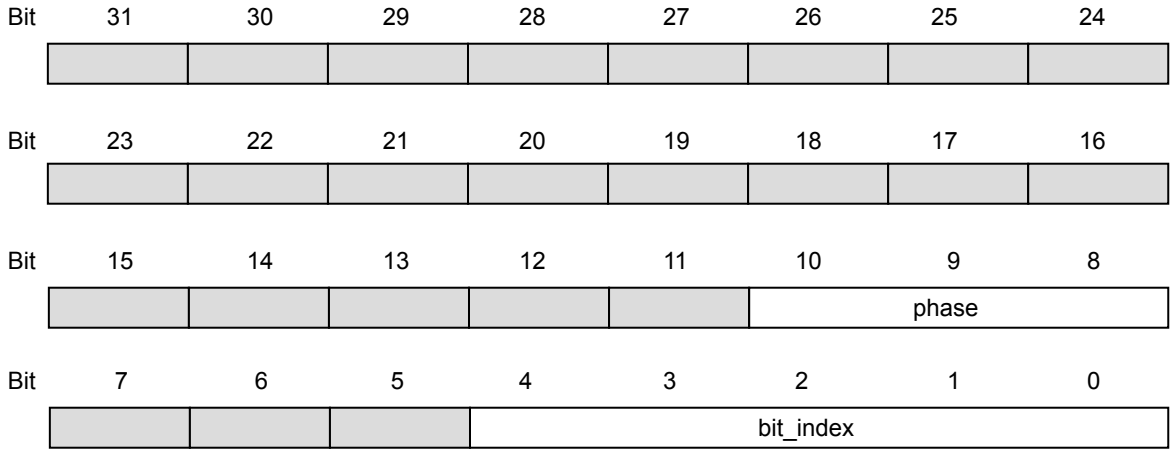
   For best test coverage, an entire physical memory block should be tested at once. It is possible to test only a subset of a memory, but the test coverage will then be somewhat lower.

   The actual test is started by writing a '1' to CTRL.MBIST. A running MBIST operation can be canceled by writing a '1' to CTRL.SWRST.
3. Interpreting the Results
   The tester should monitor the STATUSA register. When the operation is completed, STATUSA.DONE is set. There are two different modes:

- ADDR.AMOD=0: exit-on-error (default)
  In this mode, the algorithm terminates either when a fault is detected or on successful completion. In both cases, STATUSA.DONE is set. If an error was detected, STATUSA.FAIL will be set. User then can read the DATA and ADDR registers to locate the fault.
- ADDR.AMOD=1: pause-on-error
  In this mode, the MBIST algorithm is paused when an error is detected. In such a situation, only STATUSA.FAIL is asserted. The state machine waits for user to clear STATUSA.FAIL by writing a '1' in STATUSA.FAIL to resume. Prior to resuming, user can read the DATA and ADDR registers to locate the fault.

4. Locating Faults
   If the test stops with STATUSA.FAIL set, one or more bits failed the test. The test stops at the first detected error. The position of the failing bit can be found by reading the following registers:
   - ADDR: Address of the word containing the failing bit
   - DATA: contains data to identify which bit failed, and during which phase of the test it failed. The DATA register will in this case contains the following bit groups:

Figure 15-6. DATA bits Description When MBIST Operation Returns an Error



- bit_index: contains the bit number of the failing bit
- phase: indicates which phase of the test failed and the cause of the error, as listed in the following table.

Table 15-4. MBIST Operation Phases

Phase	Test actions
0	Write all bits to zero. This phase cannot fail.
1	Read '0', write '1', increment address
2	Read '1', write '0'
3	Read '0', write '1', decrement address
4	Read '1', write '0', decrement address
5	Read '0', write '1'
6	Read '1', write '0', decrement address
7	Read all zeros. bit_index is not used

Table 15-5. AMOD Bit Descriptions for MBIST

AMOD[1:0]	Description
0x0	Exit on Error
0x1	Pause on Error
0x2, 0x3	Reserved

Related Links

NVMCTRL – Non-Volatile Memory Controller on page 483

Security Bit on page 490

15.11.6. System Services Availability when Accessed Externally

External access: Access performed in the DSU address offset 0x200-0x1FFF range.

Internal access: Access performed in the DSU address offset 0x0-0x100 range.

Table 15-6. Available Features when Operated From The External Address Range and Device is Protected

Features	Availability From The External Address Range and Device is Protected
Chip-Erase command and status	Yes
CRC32	Yes, only full array or full EEPROM
CoreSight Compliant Device identification	Yes
Debug communication channels	Yes
Testing of onboard memories (MBIST)	Yes
STATUSA.CRSTEXT clearing	No (STATUSA.PERR is set when attempting to do so)

15.12. Register Summary

Offset	Name	Bit Pos.									
0x00	CTRL	7:0				CE	MBIST		CRC	SWRST	
0x01	STATUSA	7:0				PERR	FAIL	BERR	CRSTEXT	DONE	
0x02	STATUSB	7:0				HPE	DCCD1	DCCD0	DBGPRES	PROT	
0x03	Reserved										
0x04	ADDR	7:0							AMOD[1:0]		
0x05		15:8									
0x06		23:16									
0x07		31:24									
0x08	LENGTH	7:0	LENGTH[5:0]								
0x09		15:8	LENGTH[13:6]								
0x0A		23:16	LENGTH[21:14]								
0x0B		31:24	LENGTH[29:22]								
0x0C	DATA	7:0	DATA[7:0]								
0x0D		15:8	DATA[15:8]								
0x0E		23:16	DATA[23:16]								
0x0F		31:24	DATA[31:24]								
0x10	DCC0	7:0	DATA[7:0]								
0x11		15:8	DATA[15:8]								
0x12		23:16	DATA[23:16]								
0x13		31:24	DATA[31:24]								
0x14	DCC1	7:0	DATA[7:0]								
0x15		15:8	DATA[15:8]								
0x16		23:16	DATA[23:16]								
0x17		31:24	DATA[31:24]								
0x18	DID	7:0	DSEL[7:0]								
0x19		15:8	DIE[3:0]				REVISION[3:0]				
0x1A		23:16	FAMILY[0:0]		SERIES[5:0]						
0x1B		31:24	PROCESSOR[3:0]				FAMILY[4:1]				
0x1C ... 0x0FFF	Reserved										
0x1000	ENTRY0	7:0							FMT	EPRES	
0x1001		15:8	ADDOFF[3:0]								
0x1002		23:16	ADDOFF[11:4]								
0x1003		31:24	ADDOFF[19:12]								
0x1004	ENTRY1	7:0							FMT	EPRES	
0x1005		15:8	ADDOFF[3:0]								
0x1006		23:16	ADDOFF[11:4]								
0x1007		31:24	ADDOFF[19:12]								
0x1008	END	7:0	END[7:0]								
0x1009		15:8	END[15:8]								
0x100A		23:16	END[23:16]								
0x100B		31:24	END[31:24]								

Offset	Name	Bit Pos.								
0x100C ... 0x1FCB	Reserved									
0x1FCC	MEMTYPE	7:0								SMEMP
0x1FCD		15:8								
0x1FCE		23:16								
0x1FCF		31:24								
0x1FD0	PID4	7:0	FKBC[3:0]				JEPCC[3:0]			
0x1FD1		15:8								
0x1FD2		23:16								
0x1FD3		31:24								
0x1FD4 ... 0x1FDF	Reserved									
0x1FE0	PID0	7:0	PARTNBL[7:0]							
0x1FE1		15:8								
0x1FE2		23:16								
0x1FE3		31:24								
0x1FE4	PID1	7:0	JEPIDCL[3:0]				PARTNBH[3:0]			
0x1FE5		15:8								
0x1FE6		23:16								
0x1FE7		31:24								
0x1FE8	PID2	7:0	REVISION[3:0]				JEPU	JEPIDCH[2:0]		
0x1FE9		15:8								
0x1FEA		23:16								
0x1FEB		31:24								
0x1FEC	PID3	7:0	REVAND[3:0]				CUSMOD[3:0]			
0x1FED		15:8								
0x1FEE		23:16								
0x1FEF		31:24								
0x1FF0	CID0	7:0	PREAMBLEB0[7:0]							
0x1FF1		15:8								
0x1FF2		23:16								
0x1FF3		31:24								
0x1FF4	CID1	7:0	CCLASS[3:0]				PREAMBLE[3:0]			
0x1FF5		15:8								
0x1FF6		23:16								
0x1FF7		31:24								
0x1FF8	CID2	7:0	PREAMBLEB2[7:0]							
0x1FF9		15:8								
0x1FFA		23:16								
0x1FFB		31:24								
0x1FFC	CID3	7:0	PREAMBLEB3[7:0]							
0x1FFD		15:8								
0x1FFE		23:16								
0x1FFF		31:24								

15.13. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection on page 89.

15.13.1. Control

Name: CTRL
Offset: 0x0000
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
				CE	MBIST		CRC	SWRST
Access				W	W		W	W
Reset				0	0		0	0

Bit 4 – CE: Chip Erase
Writing a '0' to this bit has no effect.

Writing a '1' to this bit starts the Chip-Erase operation.

Bit 3 – MBIST: Memory Built-In Self-Test
Writing a '0' to this bit has no effect.

Writing a '1' to this bit starts the memory BIST algorithm.

Bit 1 – CRC: 32-bit Cyclic Redundancy Check
Writing a '0' to this bit has no effect.

Writing a '1' to this bit starts the cyclic redundancy check algorithm.

Bit 0 – SWRST: Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets the module.

15.13.2. Status A

Name: STATUSA
Offset: 0x0001
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
				PERR	FAIL	BERR	CRSTEXT	DONE
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit 4 – PERR: Protection Error
Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Protection Error bit.

This bit is set when a command that is not allowed in protected state is issued.

Bit 3 – FAIL: Failure
Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Failure bit.

This bit is set when a DSU operation failure is detected.

Bit 2 – BERR: Bus Error
Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Bus Error bit.

This bit is set when a bus error is detected.

Bit 1 – CRSTEXT: CPU Reset Phase Extension
Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the CPU Reset Phase Extension bit.

This bit is set when a debug adapter Cold-Plugging is detected, which extends the CPU reset phase.

Bit 0 – DONE: Done
Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Done bit.

This bit is set when a DSU operation is completed.

15.13.3. Status B

Name: STATUSB
Offset: 0x0002
Reset: 0x1X
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
				HPE	DCCD1	DCCD0	DBGPRES	PROT
Access				R	R	R	R	R
Reset				1	0	0	x	x

Bit 4 – HPE: Hot-Plugging Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit has no effect.

This bit is set when Hot-Plugging is enabled.

This bit is cleared when Hot-Plugging is disabled. This is the case when the SWCLK function is changed. Only a power-reset or a external reset can set it again.

Bit 1 – DBGPRES: Debugger Present
Writing a '0' to this bit has no effect.

Writing a '1' to this bit has no effect.

This bit is set when a debugger probe is detected.

This bit is never cleared.

Bit 0 – PROT: Protected
Writing a '0' to this bit has no effect.

Writing a '1' to this bit has no effect.

This bit is set at power-up when the device is protected.

This bit is never cleared.

Bits 3,2 – DCCDx: Debug Communication Channel x Dirty [x=1..0]
Writing a '0' to this bit has no effect.

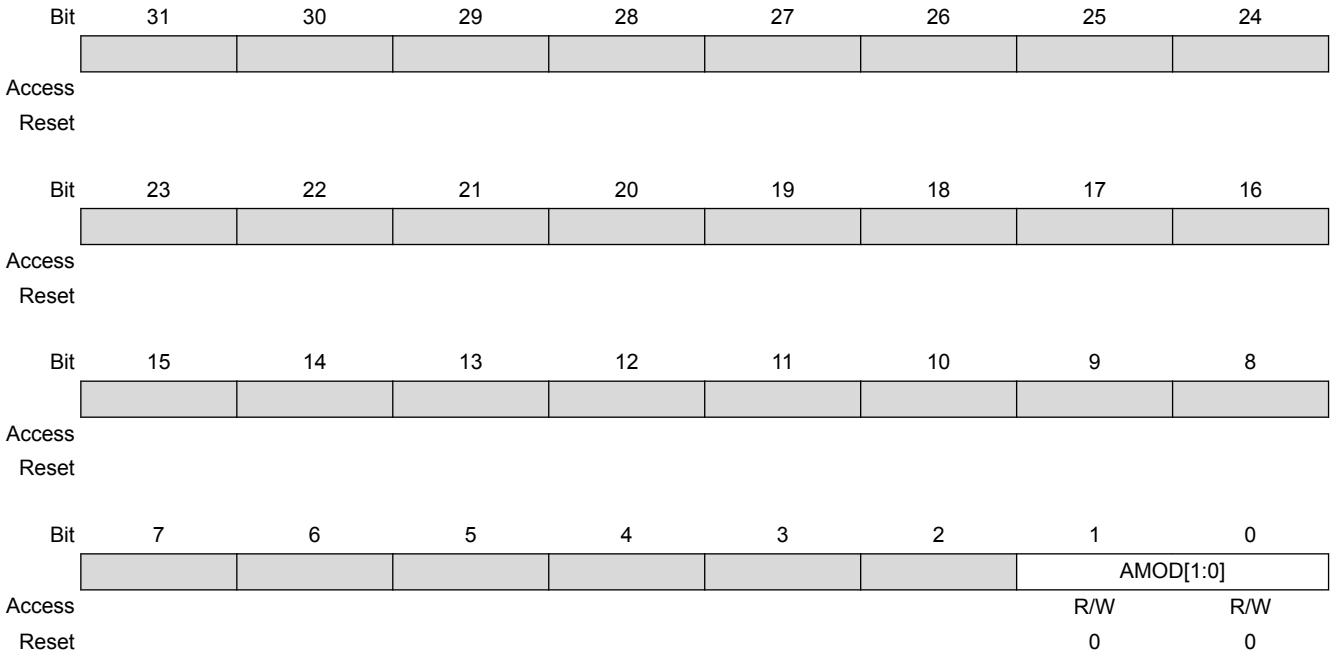
Writing a '1' to this bit has no effect.

This bit is set when DCCx is written.

This bit is cleared when DCCx is read.

15.13.4. Address

Name: ADDR
Offset: 0x0004
Reset: 0x00000000
Property: PAC Write-Protection



Bits 1:0 – AMOD[1:0]: Address
The functionality of these bits is dependent on the operation mode.

Bit description when operating CRC32: refer to 32-bit Cyclic Redundancy Check CRC32 on page 95

Bit description when testing onboard memories (MBIST): refer to Testing of On-Board Memories MBIST on page 97

15.13.5. Length

Name: LENGTH
Offset: 0x0008
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	LENGTH[29:22]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	LENGTH[21:14]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	LENGTH[13:6]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	LENGTH[5:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bits 31:2 – LENGTH[29:0]: Length
Length in words needed for memory operations.

15.13.6. Data

Name: DATA
Offset: 0x000C
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	DATA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DATA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DATA[31:0]: Data
Memory operation initial value or result value.

15.13.7. Debug Communication Channel 0

Name: DCC0
Offset: 0x0010
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	DATA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DATA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DATA[31:0]: Data
Data register.

15.13.8. Debug Communication Channel 1

Name: DCC1
Offset: 0x0014
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	DATA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DATA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DATA[31:0]: Data
Data register.

15.13.9. Device Identification

The information in this register is related to the *Ordering Information*.

Name: DID
Offset: 0x0018
Reset: see related links
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	PROCESSOR[3:0]				FAMILY[4:1]			
Access	R	R	R	R	R	R	R	R
Reset	p	p	p	p	f	f	f	f

Bit	23	22	21	20	19	18	17	16
	FAMILY[0:0]		SERIES[5:0]					
Access	R		R	R	R	R	R	R
Reset	f		s	s	s	s	s	s

Bit	15	14	13	12	11	10	9	8
	DIE[3:0]				REVISION[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	d	d	d	d	r	r	r	r

Bit	7	6	5	4	3	2	1	0
	DSEL[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

Bits 31:28 – PROCESSOR[3:0]: Processor
The value of this field defines the processor used on the device.

Bits 27:23 – FAMILY[4:0]: Product Family
The value of this field corresponds to the Product Family part of the ordering code.

Bits 21:16 – SERIES[5:0]: Product Series
The value of this field corresponds to the Product Series part of the ordering code.

Bits 15:12 – DIE[3:0]: Die Number
Identifies the die family.

Bits 11:8 – REVISION[3:0]: Revision Number
Identifies the die revision number. 0x0=rev.A, 0x1=rev.B etc.

Note: The device variant (last letter of the ordering number) is independent of the die revision.

Bits 7:0 – DSEL[7:0]: Device Selection
This bit field identifies a device within a product family and product series. Refer to the Ordering Information for device configurations and corresponding values for Flash memory density, pin count and device variant.

15.13.10. CoreSight ROM Table Entry 0

Name: ENTRY0
Offset: 0x1000
Reset: 0xXXXXX00X
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	ADDOFF[19:12]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

Bit	23	22	21	20	19	18	17	16
	ADDOFF[11:4]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

Bit	15	14	13	12	11	10	9	8
	ADDOFF[3:0]							
Access	R	R	R	R				
Reset	x	x	x	x				

Bit	7	6	5	4	3	2	1	0
							FMT	EPRES
Access							R	R
Reset							1	x

Bits 31:12 – ADDOFF[19:0]: Address Offset
The base address of the component, relative to the base address of this ROM table.

Bit 1 – FMT: Format
Always reads as '1', indicating a 32-bit ROM table.

Bit 0 – EPRES: Entry Present
This bit indicates whether an entry is present at this location in the ROM table.

This bit is set at power-up if the device is not protected indicating that the entry is not present.

This bit is cleared at power-up if the device is not protected indicating that the entry is present.

15.13.11. CoreSight ROM Table Entry 1

Name: ENTRY1
Offset: 0x1004
Reset: 0xXXXXX00X
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	ADDOFF[19:12]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

Bit	23	22	21	20	19	18	17	16
	ADDOFF[11:4]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

Bit	15	14	13	12	11	10	9	8
	ADDOFF[3:0]							
Access	R	R	R	R				
Reset	x	x	x	x				

Bit	7	6	5	4	3	2	1	0
							FMT	EPRES
Access							R	R
Reset							1	x

Bits 31:12 – ADDOFF[19:0]: Address Offset
The base address of the component, relative to the base address of this ROM table.

Bit 1 – FMT: Format
Always read as '1', indicating a 32-bit ROM table.

Bit 0 – EPRES: Entry Present
This bit indicates whether an entry is present at this location in the ROM table.

This bit is set at power-up if the device is not protected indicating that the entry is not present.

This bit is cleared at power-up if the device is not protected indicating that the entry is present.

15.13.12. CoreSight ROM Table End

Name: END
Offset: 0x1008
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	END[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	END[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	END[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	END[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – END[31:0]: End Marker
Indicates the end of the CoreSight ROM table entries.

15.13.13. CoreSight ROM Table Memory Type

Name: MEMTYPE
Offset: 0x1FCC
Reset: 0x0000000X
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								SMEMP
Access								R
Reset								x

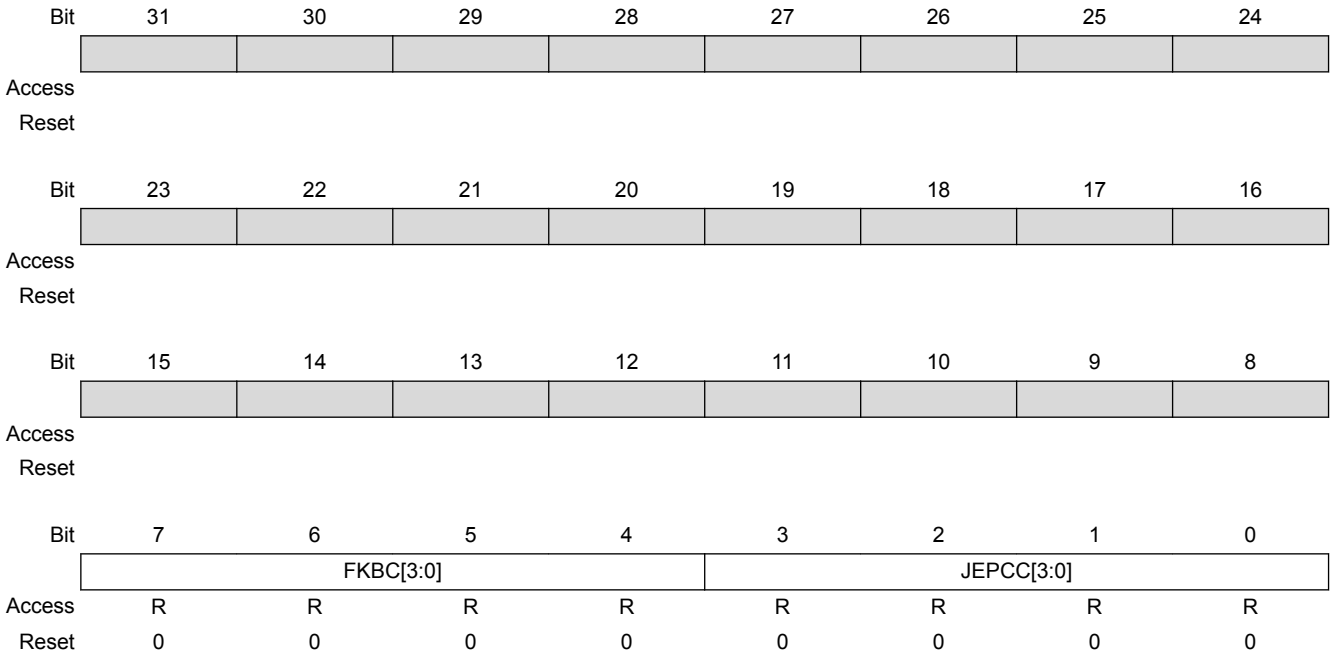
Bit 0 – SMEMP: System Memory Present
This bit indicates whether system memory is present on the bus that connects to the ROM table.

This bit is set at power-up if the device is not protected, indicating that the system memory is accessible from a debug adapter.

This bit is cleared at power-up if the device is protected, indicating that the system memory is not accessible from a debug adapter.

15.13.14. Peripheral Identification 4

Name: PID4
Offset: 0x1FD0
Reset: 0x00000000
Property: -



Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

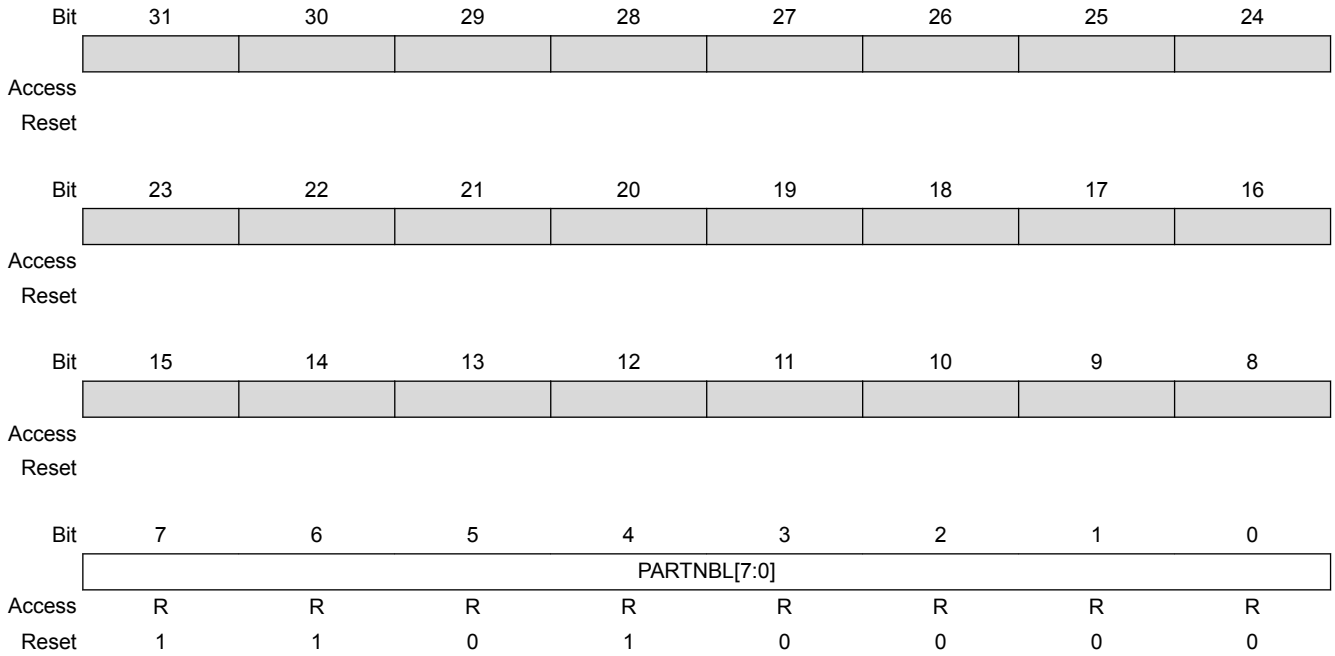
Bit	7	6	5	4	3	2	1	0
	FKBC[3:0]				JEPCC[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 7:4 – FKBC[3:0]: 4KB Count
These bits will always return zero when read, indicating that this debug component occupies one 4KB block.

Bits 3:0 – JEPCC[3:0]: JEP-106 Continuation Code
These bits will always return zero when read, indicating an Atmel device.

15.13.15. Peripheral Identification 0

Name: PID0
Offset: 0x1FE0
Reset: 0x000000D0
Property: -



Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	PARTNBL[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	1	1	0	1	0	0	0	0

Bits 7:0 – PARTNBL[7:0]: Part Number Low
These bits will always return 0xD0 when read, indicating that this device implements a DSU module instance.

15.13.16. Peripheral Identification 1

Name: PID1
Offset: 0x1FE4
Reset: 0x000000FC
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	JEPIDCL[3:0]				PARTNBH[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	1	1	1	1	1	1	0	0

Bits 7:4 – JEPIDCL[3:0]: Low part of the JEP-106 Identity Code
These bits will always return 0xF when read, indicating a Atmel device (Atmel JEP-106 identity code is 0x1F).

Bits 3:0 – PARTNBH[3:0]: Part Number High
These bits will always return 0xC when read, indicating that this device implements a DSU module instance.

15.13.17. Peripheral Identification 2

Name: PID2
Offset: 0x1FE8
Reset: 0x00000009
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	REVISION[3:0]				JEPU	JEPIDCH[2:0]		
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	1	0	0	1

Bits 7:4 – REVISION[3:0]: Revision Number
Revision of the peripheral. Starts at 0x0 and increments by one at both major and minor revisions.

Bit 3 – JEPU: JEP-106 Identity Code is used
This bit will always return one when read, indicating that JEP-106 code is used.

Bits 2:0 – JEPIDCH[2:0]: JEP-106 Identity Code High
These bits will always return 0x1 when read, indicating an Atmel device (Atmel JEP-106 identity code is 0x1F).

15.13.18. Peripheral Identification 3

Name: PID3
Offset: 0x1FEC
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	REVAND[3:0]				CUSMOD[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 7:4 – REVAND[3:0]: Revision Number
These bits will always return 0x0 when read.

Bits 3:0 – CUSMOD[3:0]: ARM CUSMOD
These bits will always return 0x0 when read.

15.13.19. Component Identification 0

Name: CID0
Offset: 0x1FF0
Reset: 0x0000000D
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	PREAMBLEB0[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	1	1	0	1

Bits 7:0 – PREAMBLEB0[7:0]: Preamble Byte 0
These bits will always return 0xD when read.

15.13.20. Component Identification 1

Name: CID1
Offset: 0x1FF4
Reset: 0x00000010
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	CCLASS[3:0]				PREAMBLE[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	1	0	0	0	0

Bits 7:4 – CCLASS[3:0]: Component Class
These bits will always return 0x1 when read indicating that this ARM CoreSight component is ROM table (refer to the ARM Debug Interface v5 Architecture Specification at http://www.arm.com).

Bits 3:0 – PREAMBLE[3:0]: Preamble
These bits will always return 0x0 when read.

15.13.21. Component Identification 2

Name: CID2
Offset: 0x1FF8
Reset: 0x00000005
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	PREAMBLEB2[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	1	0	1

Bits 7:0 – PREAMBLEB2[7:0]: Preamble Byte 2
These bits will always return 0x05 when read.

15.13.22. Component Identification 3

Name: CID3
Offset: 0x1FFC
Reset: 0x000000B1
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	PREAMBLEB3[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	1	0	1	1	0	0	0	1

Bits 7:0 – PREAMBLEB3[7:0]: Preamble Byte 3
These bits will always return 0xB1 when read.

16. Clock System

This chapter summarizes the clock distribution and terminology in the SAM L21 device. It will not explain every detail of its configuration. For in-depth documentation, see the respective peripherals descriptions and the *Generic Clock* documentation.

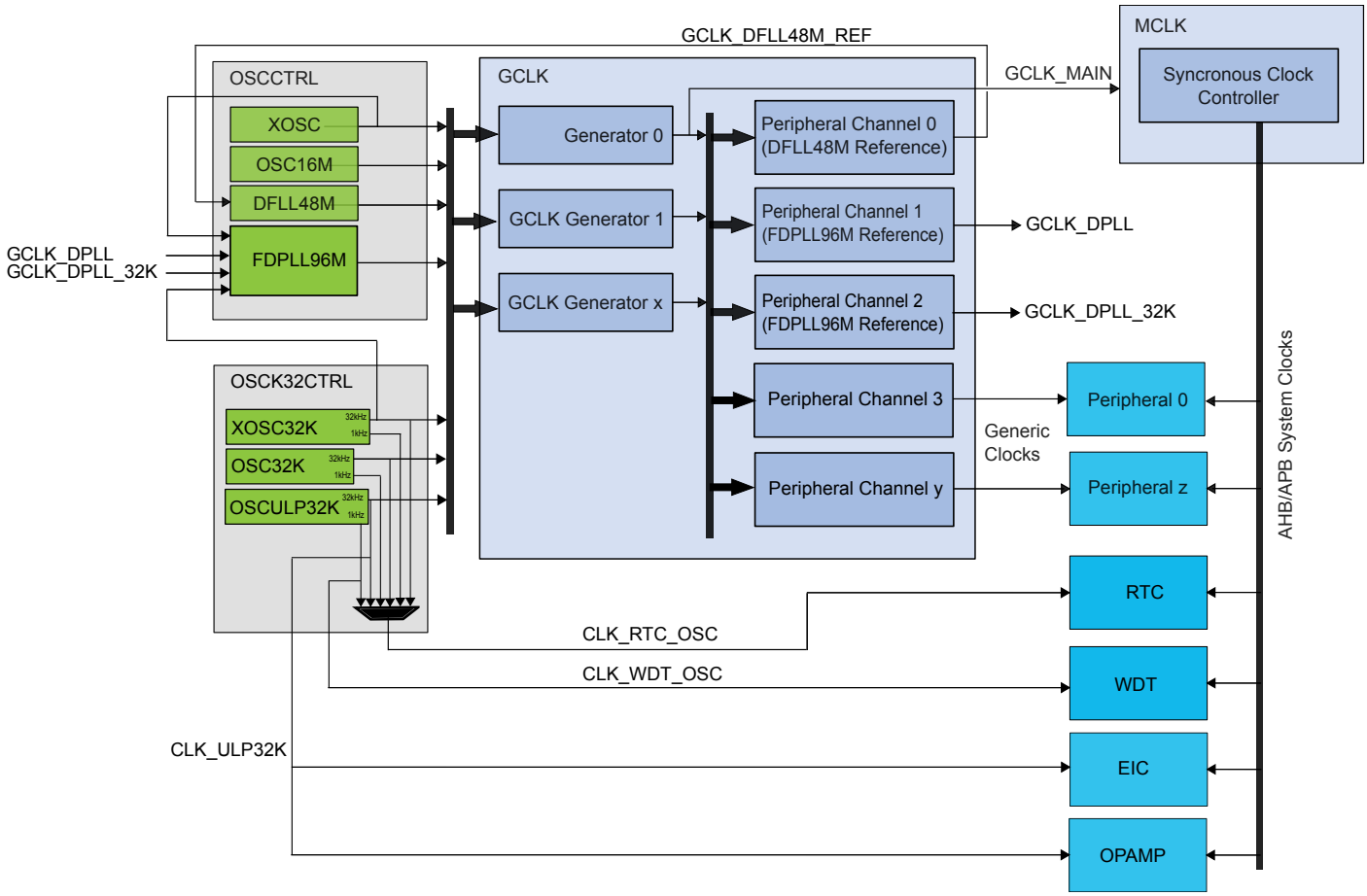
Related Links

GCLK - Generic Clock Controller on page 131

MCLK – Main Clock on page 148

16.1. Clock Distribution

Figure 16-1. Clock Distribution



The SAM L21 clock system consists of:

- *Clock sources*, controlled by OSCCTRL and OSC32KCTRL
  - A clock source provides a time base that is used by other components, such as Generic Clock Generators. Example clock sources are the internal 16MHz oscillator (OSC16M), External crystal oscillator (XOSC) and the Digital frequency locked loop (DFLL48M).
- *Generic Clock Controller (GCLK),* which generates, controls and distributes the asynchronous clock consisting of:
  - *Generic Clock Generators:* These are programmable prescalers that can use any of the system clock sources as a time base. The Generic Clock Generator 0 generates the clock

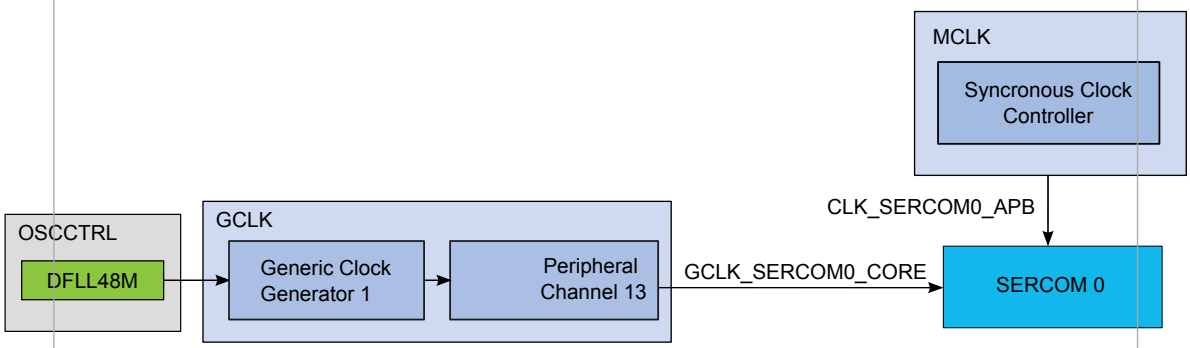
signal GCLK_MAIN, which is used by the Power Manager and the Main Clock (MCLK) module, which in turn generates synchronous clocks.

- *Generic Clocks:* These are clock signals generated by Generic Clock Generators and output by the Peripheral Channels, and serve as clocks for the peripherals of the system. Multiple instances of a peripheral will typically have a separate Generic Clock for each instance. Generic Clock 0 serves as the clock source for the DFLL48M clock input (when multiplying another clock source).

- *Main Clock Controller (MCLK)*
  - The MCLK generates and controls the synchronous clocks on the system. This includes the CPU, bus clocks (APB, AHB) as well as the synchronous (to the CPU) user interfaces of the peripherals. It contains clock masks that can turn on/off the user interface of a peripheral as well as prescalers for the CPU and bus clocks.

The next figure shows an example where SERCOM0 is clocked by the DFLL48M in open loop mode. The DFLL48M is enabled, the Generic Clock Generator 1 uses the DFLL48M as its clock source and feeds into Peripheral Channel 13. The Generic Clock 13, also called GCLK_SERCOM0_CORE, is connected to SERCOM0. The SERCOM0 interface, clocked by CLK_SERCOM0_APB, has been unmasked in the APBC Mask register in the MCLK.

Figure 16-2. Example of SERCOM Clock



16.2. Synchronous and Asynchronous Clocks

As the CPU and the peripherals can be in different clock domains, i.e. they are clocked from different clock sources and/or with different clock speeds, some peripheral accesses by the CPU need to be synchronized. In this case the peripheral includes a Synchronization Busy (SYNCBUSY) register that can be used to check if a sync operation is in progress.

For a general description, see Register Synchronization on page 127. Some peripherals have specific properties described in their individual sub-chapter “Synchronization”.

In the datasheet, references to Synchronous Clocks are referring to the CPU and bus clocks (MCLK), while asynchronous clocks are generated by the Generic Clock Controller (GCLK).

Related Links

Synchronization on page 138

16.3. Register Synchronization

16.3.1. Overview

All peripherals are composed of one digital bus interface connected to the APB or AHB bus and running from a corresponding clock in the Main Clock domain, and one peripheral core running from the peripheral Generic Clock (GCLK).

Communication between these clock domains must be synchronized. This mechanism is implemented in hardware, so the synchronization process takes place even if the peripheral generic clock is running from the same clock source and on the same frequency as the bus interface.

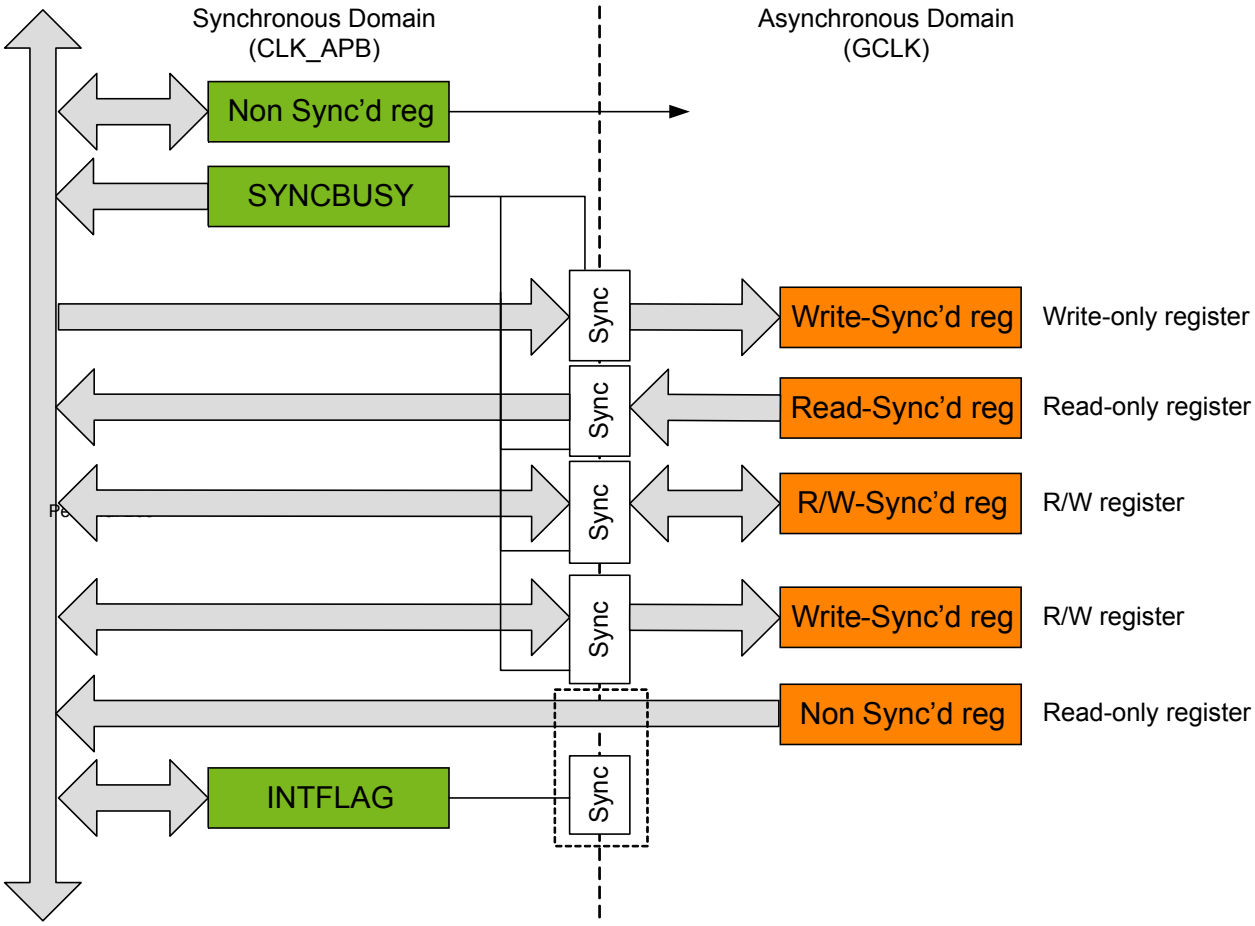
All registers in the bus interface are accessible without synchronization.

All registers in the peripheral core are synchronized when written. Some registers in the peripheral core are synchronized when read.

Each individual register description will have the properties "Read-Synchronized" and/or "Write-Synchronized" if a register is synchronized.

As shown in the figure below, each register that requires synchronization has its individual synchronizer and its individual synchronization status bit in the Synchronization Busy register (SYNCBUSY).
Note: For registers requiring both read- and write-synchronization, the corresponding bit in SYNCBUSY is shared.

Figure 16-3. Register Synchronization Overview



16.3.2. General Write Synchronization

Write-Synchronization is triggered by writing to a register in the peripheral clock domain. The respective bit in the Synchronization Busy register (SYNCBUSY) will be set when the write-synchronization starts and cleared when the write-synchronization is complete. Refer to Synchronization Delay on page 129 for details on the synchronization delay.

When write-synchronization is ongoing for a register, any subsequent write attempts to this register will be discarded, and an error will be reported.

Example:

REGA, REGB are 8-bit core registers. REGC is a 16-bit core register.

Offset	Register
0x00	REGA
0x01	REGB
0x02	REGC
0x03	

Synchronization is per register, so multiple registers can be synchronized in parallel. Consequently, after REGA (8-bit access) was written, REGB (8-bit access) can be written immediately without error.

REGC (16-bit access) can be written without affecting REGA or REGB. If REGC is written to in two consecutive 8-bit accesses without waiting for synchronization, the second write attempt will be discarded and an error is generated.

A 32-bit access to offset 0x00 will write all three registers. Note that REGA, REGB and REGC can be updated at different times because of independent write synchronization.

Related Links

PAC - Peripheral Access Controller on page 58

16.3.3. General Read Synchronization

Read-synchronized registers are synchronized each time the register value is updated but the corresponding SYNCBUSY bits are not set. Reading a read-synchronized register does not start a new synchronization, it returns the last synchronized value.

Note: The corresponding bits in SYNCBUSY will automatically be set when the device wakes up from sleep because read-synchronized registers need to be synchronized. Therefore reading a read-synchronized register before its corresponding SYNCBUSY bit is cleared will return the last synchronized value before sleep mode.

Moreover, if a register is also write-synchronized, any write access while SYNCBUSY bit is set will be discarded and generate an error.

16.3.4. Completion of Synchronization

In order to check if synchronization is complete, the user can either poll the relevant bits in SYNCBUSY or use the Synchronisation Ready interrupt (if available). The Synchronization Ready interrupt flag will be set when all ongoing synchronizations are complete, i.e. when all bits in SYNCBUSY are '0'.

16.3.5. Enable Write Synchronization

Setting the Enable bit in a module's Control A register (CTRLA.ENABLE) will trigger write-synchronization and set SYNCBUSY.ENABLE.

CTRLA.ENABLE will read its new value immediately after being written.

SYNCBUSY.ENABLE will be cleared by hardware when the operation is complete.

The Synchronisation Ready interrupt (if available) cannot be used to enable write-synchronization.

16.3.6. Software Reset Write-Synchronization

Setting the Software Reset bit in CTRLA (CTRLA.SWRST=1) will trigger write-synchronization and set SYNCBUSY.SWRST. When writing a '1' to the CTRLA.SWRST bit it will immediately read as '1'.

CTRL.SWRST and SYNCBUSY.SWRST will be cleared by hardware when the peripheral has been reset.

Writing a '0' to the CTRL.SWRST bit has no effect.

The Ready interrupt (if available) cannot be used for Software Reset write-synchronization.

16.3.7. Synchronization Delay

The synchronization will delay write and read accesses by a certain amount. This delay *D* is within the range of:

$5 \times P_{GCLK} + 2 \times P_{APB} < D < 6 \times P_{GCLK} + 3 \times P_{APB}$

Where P_{GCLK} is the period of the generic clock and P_{APB} is the period of the peripheral bus clock. A normal peripheral bus register access duration is $2 \times P_{APB}$.

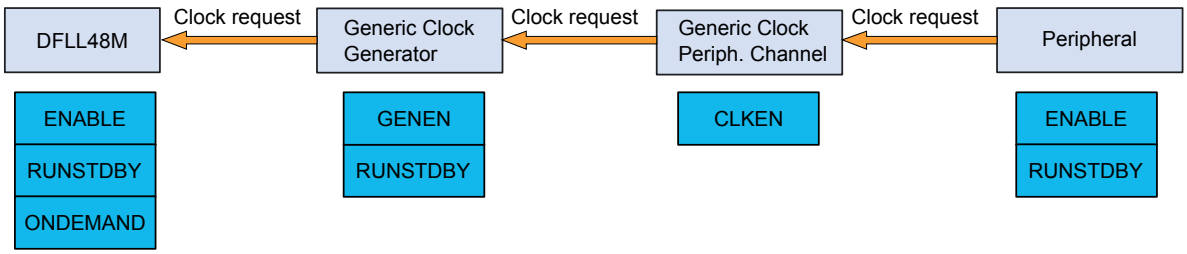
16.4. Enabling a Peripheral

In order to enable a peripheral that is clocked by a Generic Clock, the following parts of the system needs to be configured:

- A running Clock Source
- A clock from the Generic Clock Generator must be configured to use one of the running Clock Sources, and the Generator must be enabled.
- The Peripheral Channel that provides the Generic Clock signal to the peripheral must be configured to use a running Generic Clock Generator, and the Generic Clock must be enabled.
- The user interface of the peripheral needs to be unmasked in the PM. If this is not done the peripheral registers will read all 0's and any writing attempts to the peripheral will be discarded.

16.5. On Demand Clock Requests

Figure 16-4. Clock Request Routing



All clock sources in the system can be run in an on-demand mode: the clock source is in a stopped state unless a peripheral is requesting the clock source. Clock requests propagate from the peripheral, via the GCLK, to the clock source. If one or more peripheral is using a clock source, the clock source will be started/kept running. As soon as the clock source is no longer needed and no peripheral has an active request, the clock source will be stopped until requested again.

The clock request can reach the clock source only if the peripheral, the generic clock and the clock from the Generic Clock Generator in-between are enabled. The time taken from a clock request being asserted

to the clock source being ready is dependent on the clock source startup time, clock source frequency as well as the divider used in the Generic Clock Generator. The total startup time T_{start} from a clock request until the clock is available for the peripheral is between:

T_{start_max} = Clock source startup time + 2 × clock source periods + 2 × divided clock source periods

T_{start_min} = Clock source startup time + 1 × clock source period + 1 × divided clock source period

The time between the last active clock request stopped and the clock is shut down, T_{stop}, is between:

T_{stop_min} = 1 × divided clock source period + 1 × clock source period

T_{stop_max} = 2 × divided clock source periods + 2 × clock source periods

The On-Demand function can be disabled individually for each clock source by clearing the ONDEMAND bit located in each clock source controller. Consequently, the clock will always run whatever the clock request status is. This has the effect of removing the clock source startup time at the cost of power consumption.

The clock request mechanism can be configured to work in standby mode by setting the RUNSDTBY bits of the modules, see Figure 16-4 Clock Request Routing on page 129.

16.6. Power Consumption vs. Speed

When targeting for either a low-power or a fast acting system, some considerations have to be taken into account due to the nature of the asynchronous clocking of the peripherals:

If clocking a peripheral with a very low clock, the active power consumption of the peripheral will be lower. At the same time the synchronization to the synchronous (CPU) clock domain is dependent on the peripheral clock speed, and will take longer with a slower peripheral clock. This will cause worse response times and longer synchronization delays.

16.7. Clocks after Reset

On any Reset the synchronous clocks start to their initial state:

- OSC16M is enabled and configured to run at 4MHz
- Generic Generator 0 uses OSC16M as source and generates GCLK_MAIN
- CPU and BUS clocks are undivided

On a Power Reset, the 32KHz clock sources are reset and the GCLK module starts to its initial state:

- All Generic Clock Generators are disabled except
  - Generator 0 is using OSC16M at 4MHz as source and generates GCLK_MAIN
- All Peripheral Channels in GCLK are disabled.

On a User Reset the GCLK module starts to its initial state, except for:

- Generic Clocks that are write-locked, i.e., the according WRTLOCK is set to 1 prior to Reset

Related Links

RSTC – Reset Controller on page 174

17. GCLK - Generic Clock Controller

17.1. Overview

Depending on the application, peripherals may require specific clock frequencies to operate correctly. The Generic Clock controller GCLK provides nine Generic Clock Generators that can provide a wide range of clock frequencies.

Generators can be set to use different external and internal oscillators as source. The clock of each Generator can be divided . The outputs from the Generators are used as sources for the Peripheral Channels, which provide the Generic Clock (GCLK_PERIPH) to the peripheral modules, as shown in Figure 17-2 Generic Clock Controller Block Diagram on page 132. The number of Peripheral Clocks depends on how many peripherals the device has.

Note: The Generator 0 is always the direct source of the GCLK_MAIN signal.

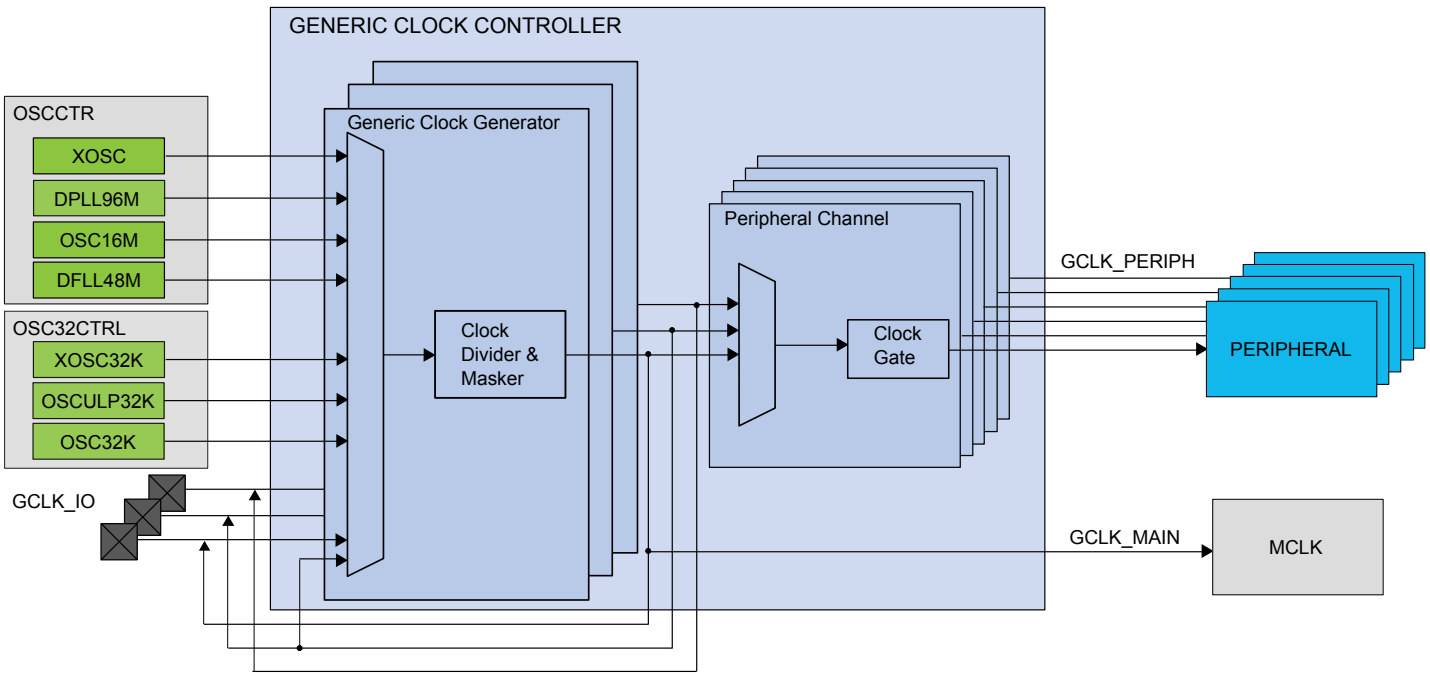
17.2. Features

- Provides a device-defined, configurable number of Peripheral Channel clocks
- Wide frequency range

17.3. Block Diagram

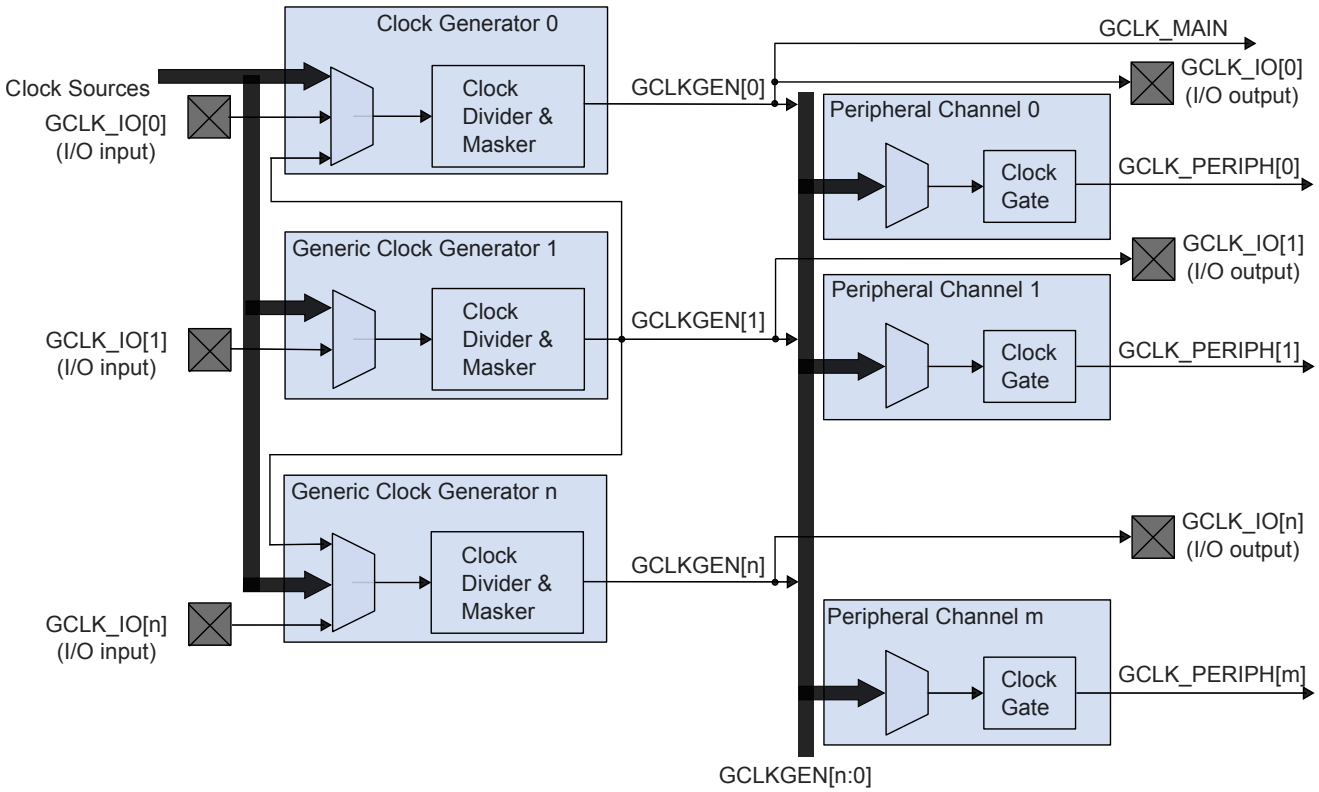
The generation of Peripheral Clock signals (GCLK_PERIPH) and the Main Clock (GCLK_MAIN) can be seen in Device Clocking Diagram.

Figure 17-1. Device Clocking Diagram



The GCLK block diagram is shown below:

Figure 17-2. Generic Clock Controller Block Diagram



17.4. Signal Description

Table 17-1. GCLK Signal Description

Signal name	Type	Description
GCLK_IO[n..0]	Digital I/O	Clock source for Generators when input Generic Clock signal when output

Note: One signal can be mapped on several pins.

Related Links

I/O Multiplexing and Considerations on page 29

17.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

17.5.1. I/O Lines

Using the GCLK I/O lines requires the I/O pins to be configured.

Related Links

PORT - I/O Pin Controller on page 506

17.5.2. Power Management

The GCLK can operate in all sleep modes, if required.

Related Links

PM – Power Manager on page 186

17.5.3. Clocks

The GCLK bus clock (CLK_GCLK_APB) can be enabled and disabled in the Main Clock Controller.

Related Links

Peripheral Clock Masking on page 151

OSC32KCTRL – 32KHz Oscillators Controller on page 264

17.5.4. DMA

Not applicable.

17.5.5. Interrupts

Not applicable.

17.5.6. Events

Not applicable.

17.5.7. Debug Operation

When the CPU is halted in debug mode the GCLK continues normal operation. If the GCLK is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

17.5.8. Register Access Protection

All registers with write-access can be optionally write-protected by the Peripheral Access Controller (PAC).

Note: Optional write-protection is indicated by the "PAC Write-Protection" property in the register description.

When the CPU is halted in debug mode, all write-protection is automatically disabled. Write-protection does not apply for accesses through an external debugger.

Related Links

PAC - Peripheral Access Controller on page 58

17.5.9. Analog Connections

Not applicable.

17.6. Functional Description

17.6.1. Principle of Operation

The GCLK module is comprised of nine Generic Clock Generators (Generators) sourcing up to 64 Peripheral Channels and the Main Clock signal GCLK_MAIN.

A clock source selected as input to a Generator can either be used directly, or it can be prescaled in the Generator. A generator output is used by one or more Peripheral Channels to provide a peripheral generic clock signal (GCLK_PERIPH) to the peripherals.

17.6.2. Basic Operation

17.6.2.1. Initialization

Before a Generator is enabled, the corresponding clock source should be enabled. The Peripheral clock must be configured as outlined by the following steps:

1. The Generator must be enabled (GENCTRLn.GENEN=1) and the division factor must be set (GENTRLn.DIVSEL and GENCTRLn.DIV) by performing a single 32-bit write to the Generator Control register (GENCTRLn).
2. The Generic Clock for a peripheral must be configured by writing to the respective Peripheral Channel Control register (PCHCTRLm). The Generator used as the source for the Peripheral Clock must be written to the GEN bit field in the Peripheral Channel Control register (PCHCTRLm.GEN).

Note: Each Generator n is configured by one dedicated register GENCTRLn.

Note: Each Peripheral Channel m is configured by one dedicated register PCHCTRLm.

17.6.2.2. Enabling, Disabling, and Resetting

The GCLK module has no enable/disable bit to enable or disable the whole module.

The GCLK is reset by setting the Software Reset bit in the Control A register (CTRLA.SWRST) to 1. All registers in the GCLK will be reset to their initial state, except for Peripheral Channels and associated Generators that have their Write Lock bit set to 1 (PCHCTRLm.WRTLOCK). For further details, refer to Configuration Lock on page 136.

17.6.2.3. Generic Clock Generator

Each Generator (GCLK_GEN) can be set to run from one of nine different clock sources except GCLK_GEN[1], which can be set to run from one of eight sources. GCLK_GEN[1] is the only Generator that can be selected as source to others Generators.

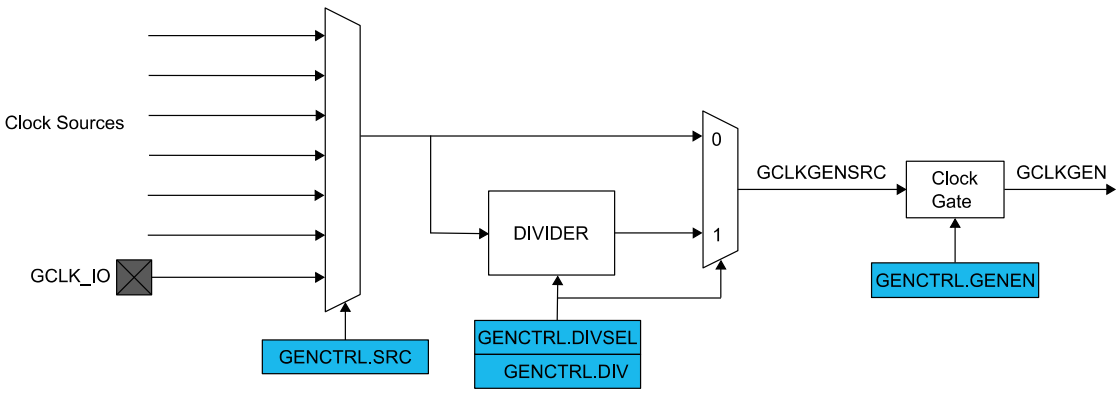
Each generator GCLK_GEN[x] can be connected to one specific pin (GCLK_IO[y]). The GCLK_IO[y] can be set to act as source to GCLK_GEN[x] or to output the clock signal generated by GCLK_GEN[x].

The selected source can be divided. Each Generator can be enabled or disabled independently.

Each GCLK_GEN clock signal can then be used as clock source for Peripheral Channels. Each Generator output is allocated to one or several Peripherals.

GCLK_GEN[0] is used as GCLK_MAIN for the synchronous clock controller inside the Main Clock Controller. Refer to the Main Clock Controller description for details on the synchronous clock generation.

Figure 17-3. Generic Clock Generator



Related Links

MCLK – Main Clock on page 148

17.6.2.4. Enabling a Generator

A Generator is enabled by writing a '1' to the Generator Enable bit in the Generator Control register (GENCTRLn.GENEN=1).

17.6.2.5. Disabling a Generator

A Generator is disabled by writing a '0' to GENCTRLn.GENEN. When GENCTRLn.GENEN=0, the GCLK_GEN[n] clock is disabled and gated.

17.6.2.6. Selecting a Clock Source for the Generator

Each Generator can individually select a clock source by setting the Source Select bit group in the Generator Control register (GENCTRLn.SRC).

Changing from one clock source, for example A, to another clock source, B, can be done on the fly: If clock source B is not ready, the Generator will continue using clock source A. As soon as source B is ready, the Generator will switch to it. During the switching operation, the Generator maintains clock requests to both clock sources A and B, and will release source A as soon as the switch is done. The according bit in SYNCBUSY register (SYNCBUSY.GENCTRLn) will remain '1' until the switch operation is completed.

The available clock sources are device dependent (usually the oscillators, RC oscillators, DPLL, and DFLL). Only Generator 1 can be used as a common source for all other generators.

17.6.2.7. Changing the Clock Frequency

The selected source for a Generator can be divided by writing a division value in the Division Factor bit field of the Generator Control register (GENCTRLn.DIV). How the actual division factor is calculated is depending on the Divide Selection bit (GENCTRLn.DIVSEL).

If GENCTRLn.DIVSEL=0 and GENCTRLn.DIV is either 0 or 1, the output clock will be undivided.

Note: The number of DIV bits for each Generator is device dependent.

17.6.2.8. Duty Cycle

When dividing a clock with an odd division factor, the duty-cycle will not be 50/50. Setting the Improve Duty Cycle bit of the Generator Control register (GENCTRLn.IDC) will result in a 50/50 duty cycle.

17.6.2.9. External Clock

The output clock (GCLK_GEN) of each Generator can be sent to I/O pins (GCLK_IO).

If the Output Enable bit in the Generator Control register is set (GENCTRLn.OE = 1) and the generator is enabled (GENCTRLn.GENEN=1), the Generator requests its clock source and the GCLK_GEN clock is output to an I/O pin.

If GENCTRLn.OE is 0, the according I/O pin is set to an Output Off Value, which is selected by GENCTRLn.OOV: If GENCTRLn.OOV is '0', the output clock will be low when turned off. If this bit is '1', the output clock will be high when turned off.

In Standby mode, if the clock is output (GENCTRLn.OE=1), the clock on the I/O pin is frozen to the OOV value if the Run In Standby bit of the Generic Control register (GENCTRLn.RUNSTDBY) is zero.
Note: With GENCTRLn.OE=1 and RUNSTDBY=0, entering the Standby mode can take longer due to a clock source dependent delay between turning off Power Domain 1 and 2. The maximum delay can be equal to the clock source period multiplied by the division factor.

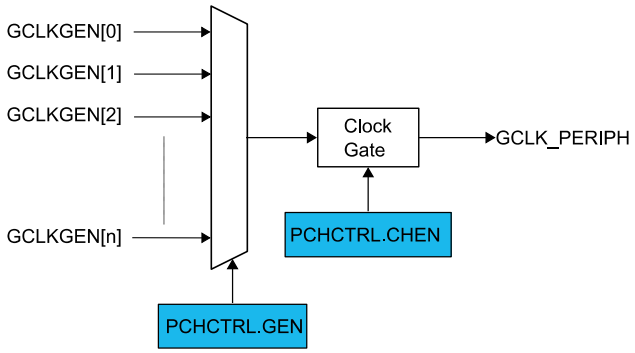
If GENCTRLn.RUNSTDBY is '1', the GCLKGEN clock is kept running and output to the I/O pin.

Related Links

Power Domain Controller on page 194

17.6.3. Peripheral Clock

Figure 17-4. Peripheral Clock



17.6.3.1. Enabling a Peripheral Clock

Before a Peripheral Clock is enabled, one of the Generators must be enabled (GENCTRLn.GENEN) and selected as source for the Peripheral Channel by setting the Generator Selection bits in the Peripheral Channel Control register (PCHCTRL.GEN). Any available Generator can be selected as clock source for each Peripheral Channel.

When a Generator has been selected, the peripheral clock is enabled by setting the Channel Enable bit in the Peripheral Channel Control register, PCHCTRLm.CHEN = 1. The PCHCTRLm.CHEN bit must be synchronized to the generic clock domain. PCHCTRLm.CHEN will continue to read as its previous state until the synchronization is complete.

17.6.3.2. Disabling a Peripheral Clock

A Peripheral Clock is disabled by writing PCHCTRLm.CHEN=0. The PCHCTRLm.CHEN bit must be synchronized to the Generic Clock domain. PCHCTRLm.CHEN will stay in its previous state until the synchronization is complete. The Peripheral Clock is gated when disabled.

Related Links

PCHCTRLm on page 145

17.6.3.3. Selecting the Clock Source for a Peripheral

When changing a peripheral clock source by writing to PCHCTRLm.GEN, the peripheral clock must be disabled before re-enabling it with the new clock source setting. This prevents glitches during the transition:

1. Disable the Peripheral Channel by writing PCHCTRLm.CHEN=0
2. Assert that PCHCTRLm.CHEN reads '0'
3. Change the source of the Peripheral Channel by writing PCHCTRLm.GEN
4. Re-enable the Peripheral Channel by writing PCHCTRLm.CHEN=1

Related Links

PCHCTRLm on page 145

17.6.3.4. Configuration Lock

The peripheral clock configuration can be locked for further write accesses by setting the Write Lock bit in the Peripheral Channel Control register PCHCTRLm.WRTLOCK=1). All writing to the PCHCTRLm register will be ignored. It can only be unlocked by a Power Reset.

The Generator source of a locked Peripheral Channel will be locked, too: The corresponding GENCTRLn register is locked, and can be unlocked only by a Power Reset.

There is one exception concerning the Generator 0. As it is used as GCLK_MAIN, it cannot be locked. It is reset by any Reset and will start up in a known configuration. The software reset (CTRLA.SWRST) can not unlock the registers.

In case of an external Reset, the Generator source will be disabled. Even if the WRTLOCK bit is written to '1' the peripheral channels are disabled (PCHCTRLm.CHEN set to '0') until the Generator source is enabled again. Then, the PCHCTRLm.CHEN are set to '1' again.

Related Links

CTRLA on page 140

PCHCTRLm on page 145

17.6.4. Additional Features

17.6.4.1. Peripheral Clock Enable after Reset

The Generic Clock Controller must be able to provide a generic clock to some specific peripherals after a Reset. That means that the configuration of the Generators and Peripheral Channels after Reset is device-dependent.

Refer to GENCTRLn.SRC for details on GENCTRLn reset.

Refer to PCHCTRLm.SRC for details on PCHCTRLm reset.

17.6.5. Sleep Mode Operation

17.6.5.1. SleepWalking

The GCLK module supports the SleepWalking feature.

If the system is in a sleep mode where the Generic Clocks are stopped, a peripheral that needs its clock in order to execute a process must request it from the Generic Clock Controller.

The Generic Clock Controller receives this request, determines which Generic Clock Generator is involved and which clock source needs to be awakened. It then wakes up the respective clock source, enables the Generator and Peripheral Channel stages successively, and delivers the clock to the peripheral.

The RUNSTDBY bit in the Generator Control register controls clock output to pin during standby sleep mode. If the bit is cleared, the Generator output is not available on pin. When set, the GCLK can continuously output the generator output to GCLK_IO. Refer to External Clock on page 135 for details.

Related Links

PM – Power Manager on page 186

17.6.5.2. Minimize Power Consumption in Standby

The following table identifies when a Clock Generator is off in Standby Mode, minimizing the power consumption:

Table 17-2. Clock Generator n Activity in Standby Mode

Request for Clock n present	GENCTRLn.RUNSTDBY	GENCTRLn.OE	Clock Generator n
yes	-	-	active
no	1	1	active
no	1	0	OFF

Request for Clock n present	GENCTRLn.RUNSTDBY	GENCTRLn.OE	Clock Generator n
no	0	1	OFF
no	0	0	OFF

17.6.5.3. Entering Standby Mode

There may occur a delay when the device is put into Standby, until the power is turned off. This delay is caused by running Clock Generators: if the Run in Standby bit in the Generator Control register (GENCTRLn.RUNSTDBY) is '0', GCLK must verify that the clock is turned of properly. The duration of this verification is frequency-dependent.

Related Links

PM – Power Manager on page 186

17.6.6. Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read. An exception is the Channel Enable bit in the Peripheral Channel Control registers (PCHCTRLm.CHEN). When changing this bit, the bit value must be read-back to ensure the synchronization is complete and to assert glitch free internal operation. Note that changing the bit value under ongoing synchronization will *not* generate an error.

An exception is the Channel Enable bit in the Peripheral Channel Control registers (PCHCTRLm.CHEN). When changing this bit, the bit value must be read-back to ensure the synchronization is complete and to assert glitch free internal operation. Note that changing the bit value under ongoing synchronization will *not* generate an error.

The following registers are synchronized when written:

- Generic Clock Generator Control register (GENCTRLn)
- Control A register (CTRLA)

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

Related Links

CTRLA on page 140
PCHCTRLm on page 145

17.7. Register Summary

17.7.1. Common Registers

Offset	Name	Bit Pos.								
0x0	CTRLA	7:0								SWRST
0x1	Reserved									
0x2	Reserved									
0x3	Reserved									

Offset	Name	Bit Pos.								
0x4	SYNCBUSY	7:0	GENCTRL5	GENCTRL4	GENCTRL3	GENCTRL2	GENCTRL1	GENCTRL0		SWRST
0x5		15:8	GENCTRL8	GENCTRL7	GENCTRL6					
0x6		23:16								
0x7		31:24								

17.7.2. Generic Clock Generator n

Offset[1]	Name	Bit Pos.									
0x20 + 0x4*n	GENCTRLn on page 142	7:0					SRC[4:0]				
0x21 + 0x4*n		15:8			RUNSTDBY	DIVSEL	OE	OOV	IDC	GENEN	
0x22 + 0x4*n		23:16	DIV[7:0]								
0x23 + 0x4*n		31:24	DIV[15:8]								

17.7.3. Peripheral Channel Control m

Offset[2]	Name	Bit Pos.								
0x80 + 0x4*m	PCHCTRLm	7:0	WRTLOCK	CHEN			GEN[3:0]			
0x81 + 0x4*m		15:8								
0x82 + 0x4*m		23:16								
0x83 + 0x4*m		31:24								

1. n is Generic Clock Generator number. Refer to GENCTRLn on page 142 for details.
2. m is Peripheral Channel number.

17.8. Register Description

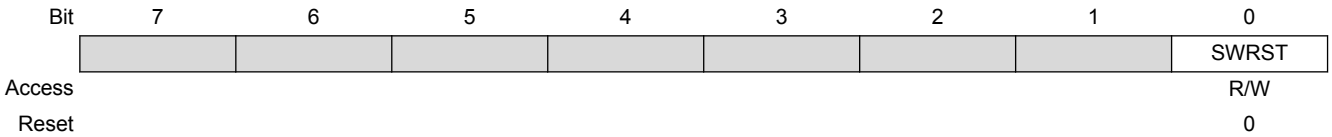
Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection on page 133.

Some registers are synchronized when read and/or written. Synchronization is denoted by the "Write-Synchronized" or the "Read-Synchronized" property in each individual register description. For details, refer to Synchronization on page 138.

17.8.1. Control A

Name: CTRLA
Offset: 0x0
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized



Bit 0 – SWRST: Software Reset
Writing a zero to this bit has no effect.

Setting this bit to 1 will reset all registers in the GCLK to their initial state after a Power Reset, except for generic clocks and associated Generators that have their WRTLOCK bit in PCHCTRLm set to 1.

Refer to GENCTRL Reset Value for details on GENCTRL register reset.

Refer to PCHCTRL Reset Value for details on PCHCTRL register reset.

Due to synchronization, there is a waiting period between setting CTRLA.SWRST and a completed Reset. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

Value	Description
0	There is no Reset operation ongoing.
1	A Reset operation is ongoing.

17.8.2. Synchronization Busy

Name: SYNCBUSY
Offset: 0x04
Reset: 0x00000000
Property: –

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						GENCTRL8	GENCTRL7	GENCTRL6
Access						R	R	R
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	GENCTRL5	GENCTRL4	GENCTRL3	GENCTRL2	GENCTRL1	GENCTRL0		SWRST
Access	R	R	R	R	R	R		R
Reset	0	0	0	0	0	0		0

Bit 0 – SWRST: SWRST Synchronization Busy
This bit is cleared when the synchronization of the CTRLA.SWRST register bit between clock domains is complete.

This bit is set when the synchronization of the CTRLA.SWRST register bit between clock domains is started.

Bits 2,3,4,5,6,7,8,9,10 – GENCTRLx: Generator Control x Synchronization Busy
This bit is cleared when the synchronization of the Generator Control n register (GENCTRLn) between clock domains is complete, or when clock switching operation is complete.

This bit is set when the synchronization of the Generator Control n register (GENCTRLn) between clock domains is started.

17.8.3. Generator Control

GENCTRLn controls the settings of Generic Generator n (n=0..8).

Name: GENCTRLn
Offset: 0x20+4*n, (n=0..8)
Reset: 0x00000106 for Generator n=0, else 0x00000000
Property: PAC Write-Protection, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
	DIV[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DIV[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
			RUNSTDBY	DIVSEL	OE	OOV	IDC	GENEN
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				SRC[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bits 31:16 – DIV[15:0]: Division Factor

These bits represent a division value for the corresponding Generator. The actual division factor is dependent on the state of DIVSEL. The number of relevant DIV bits for each Generator can be seen in this table. Written bits outside of the specified range will be ignored.

Table 17-3. Division Factor Bits

Generic Clock Generator	Division Factor Bits
Generator 0	8 division factor bits - DIV[7:0]
Generator 1	16 division factor bits - DIV[15:0]
Generator 2 - 8	8 division factor bits - DIV[7:0]

Bit 13 – RUNSTDBY: Run in Standby

This bit is used to keep the Generator running in Standby as long as it is configured to output to a dedicated GCLK_IO pin. If GENCTRLn.OE is zero, this bit has no effect and the generator will only be running if a peripheral requires the clock.

Value	Description
0	The Generator is stopped in Standby and the GCLK_IO pin state (one or zero) will be dependent on the setting in GENCTRL.OOV.
1	The Generator is kept running and output to its dedicated GCLK_IO pin during Standby mode.

Bit 12 – DIVSEL: Divide Selection
This bit determines how the division factor of the clock source of the Generator will be calculated from DIV. If the clock source should not be divided, DIVSEL must be 0 and the GENCTRLn.DIV value must be either 0 or 1.

Value	Description
0	The Generator clock frequency equals the clock source frequency divided by GENCTRLn.DIV.
1	The Generator clock frequency equals the clock source frequency divided by 2^(GENCTRLn.DIV+1).

Bit 11 – OE: Output Enable
This bit is used to output the Generator clock output to the corresponding pin (GCLK_IO), as long as GCLK_IO is not defined as the Generator source in the GENCTRLn.SRC bit field.

Value	Description
0	No Generator clock signal on pin GCLK_IO.
1	The Generator clock signal is output on the corresponding GCLK_IO, unless GCLK_IO is selected as a generator source in the GENCTRLn.SRC bit field.

Bit 10 – OOV: Output Off Value
This bit is used to control the clock output value on pin (GCLK_IO) when the Generator is turned off or the OE bit is zero, as long as GCLK_IO is not defined as the Generator source in the GENCTRLn.SRC bit field.

Value	Description
0	The GCLK_IO will be LOW when generator is turned off or when the OE bit is zero.
1	The GCLK_IO will be HIGH when generator is turned off or when the OE bit is zero.

Bit 9 – IDC: Improve Duty Cycle
This bit is used to improve the duty cycle of the Generator output to 50/50 for odd division factors.

Value	Description
0	Generator output clock duty cycle is not balanced to 50/50 for odd division factors.
1	Generator output clock duty cycle is 50/50.

Bit 8 – GENEN: Generator Enable
This bit is used to enable and disable the Generator.

Value	Description
0	Generator is disabled.
1	Generator is enabled.

Bits 4:0 – SRC[4:0]: Generator Clock Source Selection
These bits select the Generator clock source, as shown in this table.

Table 17-4. Generator Clock Source Selection

Value	Name	Description
0x00	XOSC	XOSC oscillator output
0x01	GCLK_IN	Generator input pad (GCLK_IO)
0x02	GCLK_GEN1	Generic clock generator 1 output
0x03	OSCULP32K	OSCULP32K oscillator output
0x04	OSC32K	OSC32K oscillator output
0x05	XOSC32K	XOSC32K oscillator output
0x06	OSC16M	OSC16M oscillator output
0x07	DFLL48M	DFLL48M output
0x08	DPLL96M	DPLL96M output
0x09-0x1F	Reserved	Reserved for future use

A Power Reset will reset all GENCTRLn registers. the Reset values of the GENCTRLn registers are shown in table below.

Table 17-5. GENCTRLn Reset Value after a Power Reset

GCLK Generator	Reset Value after a Power Reset
0	0x00000106
others	0x00000000

A User Reset will reset the associated GENCTRL register unless the Generator is the source of a locked Peripheral Channel (PCHCTRLm.WRTLOCK=1). The reset values of the GENCTRL register are as shown in the table below.

Table 17-6. GENCTRLn Reset Value after a User Reset

GCLK Generator	Reset Value after a User Reset
0	0x00000106
others	No change if the generator is used by a Peripheral Channel m with PCHCTRLm.WRTLOCK=1 else 0x00000000

17.8.4. Peripheral Channel Control

PCHTRLm controls the settings of Peripheral Channel number m (m=0..34).

Name: PCHCTRLm
Offset: 0x80 + 4*m, (m= 0..34)
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	WRTLOCK	CHEN			GEN[3:0]			
Access	R/W	R/W			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

Bit 7 – WRTLOCK: Write Lock

After this bit is set to '1', further writes to the PCHCTRLm register will be discarded. The control register of the corresponding Generator n (GENCTRLn), as assigned in PCHCTRLm.GEN, will also be locked. It can only be unlocked by a Power Reset.

Note that Generator 0 cannot be locked.

Value	Description
0	The Peripheral Channel register and the associated Generator register are not locked
1	The Peripheral Channel register and the associated Generator register are locked

Bit 6 – CHEN: Channel Enable

This bit is used to enable and disable a Peripheral Channel.

Value	Description
0	The Peripheral Channel is disabled
1	The Peripheral Channel is enabled

Bits 3:0 – GEN[3:0]: Generator Selection

This bit field selects the Generator to be used as the source of a peripheral clock, as shown in the table below:

Table 17-7. Generator Selection

Value	Description
0x0	Generic Clock Generator 0
0x1	Generic Clock Generator 1
0x2	Generic Clock Generator 2
0x3	Generic Clock Generator 3
0x4	Generic Clock Generator 4
0x5	Generic Clock Generator 5
0x6	Generic Clock Generator 6
0x7	Generic Clock Generator 7
0x8	Generic Clock Generator 8
0x9 - 0xF	Reserved

Table 17-8. Reset Value after a User Reset or a Power Reset

Reset	PCHCTRLm.GEN	PCHCTRLm.CHEN	PCHCTRLm.WRTLOCK
Power Reset	0x0	0x0	0x0
User Reset	If WRTLOCK = 0 : 0x0 If WRTLOCK = 1: no change	If WRTLOCK = 0 : 0x0 If WRTLOCK = 1: no change	No change

A Power Reset will reset all the PCHCTRLm registers.

A User Reset will reset a PCHCTRL if WRTLOCK=0, or else, the content of that PCHCTRL remains unchanged.

PCHCTRL register Reset values are shown in the table PCHCTRLm Mapping.

Table 17-9. PCHCTRLm Mapping

index(m)	Name	Description
0	GCLK_DFLL48M_REF	DFLL48M Reference
1	GCLK_DPLL	FDPLL96M input clock source for reference
2	GCLK_DPLL_32K	FDPLL96M 32kHz clock for FDPLL96M internal lock timer
3	GCLK_EIC	EIC
4	GCLK_USB	USB
5	GCLK_EVSYS_CHANNEL_0	EVSYS_CHANNEL_0
6	GCLK_EVSYS_CHANNEL_1	EVSYS_CHANNEL_1
7	GCLK_EVSYS_CHANNEL_2	EVSYS_CHANNEL_2
8	GCLK_EVSYS_CHANNEL_3	EVSYS_CHANNEL_3

index(m)	Name	Description
9	GCLK_EVSYS_CHANNEL_4	EVSYS_CHANNEL_4
10	GCLK_EVSYS_CHANNEL_5	EVSYS_CHANNEL_5
11	GCLK_EVSYS_CHANNEL_6	EVSYS_CHANNEL_6
12	GCLK_EVSYS_CHANNEL_7	EVSYS_CHANNEL_7
13	GCLK_EVSYS_CHANNEL_8	EVSYS_CHANNEL_8
14	GCLK_EVSYS_CHANNEL_9	EVSYS_CHANNEL_9
15	GCLK_EVSYS_CHANNEL_10	EVSYS_CHANNEL_10
16	GCLK_EVSYS_CHANNEL_11	EVSYS_CHANNEL_11
17	GCLK_SERCOM[0,1,2,3,4]_SLOW	SERCOM[0,1,2,3,4]_SLOW
18	GCLK_SERCOM0_CORE	SERCOM0_CORE
19	GCLK_SERCOM1_CORE	SERCOM1_CORE
20	GCLK_SERCOM2_CORE	SERCOM2_CORE
21	GCLK_SERCOM3_CORE	SERCOM3_CORE
22	GCLK_SERCOM4_CORE	SERCOM4_CORE
23	GCLK_SERCOM5_SLOW	SERCOM5_SLOW
24	GCLK_SERCOM5_CORE	SERCOM5_CORE
25	GCLK_TCC0, GCLK_TCC1	TCC0,TCC1
26	GCLK_TCC2	TCC2
27	GCLK_TC0, GCLK_TC1	TC0,TC1
28	GCLK_TC2, GCLK_TC3	TC2,TC3
29	GCLK_TC4	TC4
30	GCLK_ADC	ADC
31	GCLK_AC	AC
32	GCLK_DAC	DAC
33	GCLK_PTC	PTC
34	GCLK_CCL	CCL

18. MCLK – Main Clock

18.1. Overview

The Main Clock (MCLK) controls the synchronous clock generation of the device.

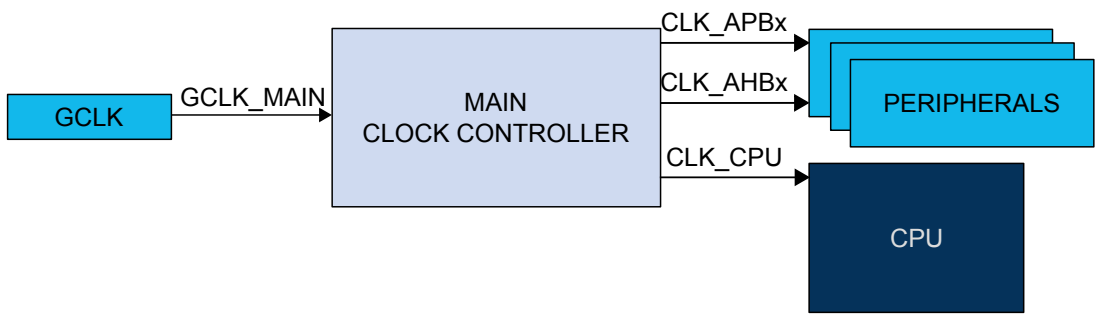
Using a clock provided by the Generic Clock Module (GCLK_MAIN), the Main Clock Controller provides synchronous system clocks to the CPU and the modules connected to the AHBx and the APBx bus. The synchronous system clocks are divided into a number of clock domains. Each clock domain can run at different frequencies, enabling the user to save power by running peripherals at a relatively low clock frequency, while maintaining high CPU performance or vice versa. In addition, the clock can be masked for individual modules, enabling the user to minimize power consumption.

18.2. Features

- Generates CPU, AHB, and APB system clocks
  - Clock source and division factor from GCLK
  - Clock prescaler with 1x to 128x division
- Safe run-time clock switching from GCLK
- Module-level clock gating through maskable peripheral clocks

18.3. Block Diagram

Figure 18-1. MCLK Block Diagram



18.4. Signal Description

Not applicable.

18.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

18.5.1. I/O Lines

Not applicable.

18.5.2. Power Management

The MCLK will operate in all sleep modes if a synchronous clock is required in these modes.

Related Links

PM – Power Manager on page 186

18.5.3. Clocks

The MCLK bus clock (CLK_MCLK_APB) can be enabled and disabled in the Main Clock module, and the default state of CLK_MCLK_APB can be found in the Peripheral Clock Masking section. If this clock is disabled, it can only be re-enabled by a reset.

The Generic Clock GCLK_MAIN is required to generate the Main Clocks. GCLK_MAIN is configured in the Generic Clock Controller, and can be re-configured by the user if needed.

Related Links

GCLK - Generic Clock Controller on page 131

Peripheral Clock Masking on page 151

18.5.3.1. Main Clock

The main clock GCLK_MAIN is the common source for the synchronous clocks. This is fed into the common 8-bit prescaler that is used to generate synchronous clocks to the CPU, AHBx, and APBx modules.

18.5.3.2. CPU Clock

The CPU clock (CLK_CPU) is routed to the CPU. Halting the CPU clock inhibits the CPU from executing instructions.

18.5.3.3. APBx and AHBx Clock

The APBx clocks (CLK_APBx) and the AHBx clocks (CLK_AHBx) are the root clock source used by modules requiring a clock on the APBx and the AHBx bus. These clocks are always synchronous to the CPU clock, but can be divided by a prescaler, and can run even when the CPU clock is turned off in sleep mode. A clock gater is inserted after the common APB clock to gate any APBx clock of a module on APBx bus, as well as the AHBx clock.

18.5.3.4. Clock Domains

The device has these synchronous clock domains:

- CPU synchronous clock domain (CPU Clock Domain). Frequency is f_{CPU}.
- Low Power synchronous clock domain (LP Clock Domain). Frequency is f_{LP}.

See also the related links for the clock domain partitioning.

Related Links

Peripheral Clock Masking on page 151

18.5.4. DMA

Not applicable.

18.5.5. Interrupts

The interrupt request line is connected to the Interrupt Controller. Using the MCLK interrupt requires the Interrupt Controller to be configured first.

18.5.6. Events

Not applicable.

18.5.7. Debug Operation

When the CPU is halted in debug mode, the MCLK continues normal operation. In sleep mode, the clocks generated from the MCLK are kept running to allow the debugger accessing any module. As a consequence, power measurements are incorrect in debug mode.

18.5.8. Register Access Protection

All registers with write-access can be write-protected optionally by the Peripheral Access Controller (PAC), except the following registers:

- Interrupt Flag register (INTFLAG)

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

PAC write-protection does not apply to accesses through an external debugger.

Related Links

PAC - Peripheral Access Controller on page 58

18.5.9. Analog Connections

Not applicable.

18.6. Functional Description

18.6.1. Principle of Operation

The GCLK_MAIN clock signal from the GCLK module is the source for the main clock, which in turn is the common root for the synchronous clocks for the CPU, APBx, and AHBx modules. The GCLK_MAIN is divided by an 8-bit prescaler. Each of the derived clocks can run from any divided or undivided main clock, ensuring synchronous clock sources for each clock domain. Each clock domain (CPU, LP) can be changed on the fly to respond to variable load in the application as long as $f_{CPU} \geq f_{LP} \geq f_{BUP}$. The clocks for each module in a clock domain can be masked individually to avoid power consumption in inactive modules. Depending on the sleep mode, some clock domains can be turned off.

18.6.2. Basic Operation

18.6.2.1. Initialization

After a Reset, the default clock source of the GCLK_MAIN clock is started and calibrated before the CPU starts running. The GCLK_MAIN clock is selected as the main clock without any prescaler division.

By default, only the necessary clocks are enabled.

Related Links

Peripheral Clock Masking on page 151

18.6.2.2. Enabling, Disabling, and Resetting

The MCLK module is always enabled and cannot be reset.

18.6.2.3. Selecting the Main Clock Source

Refer to the Generic Clock Controller description for details on how to configure the clock source of the GCLK_MAIN clock.

Related Links

GCLK - Generic Clock Controller on page 131

18.6.2.4. Selecting the Synchronous Clock Division Ratio

The main clock GCLK_MAIN feeds an 8-bit prescaler, which can be used to generate the synchronous clocks. By default, the synchronous clocks run on the undivided main clock. The user can select a prescaler division for the CPU clock domain by writing the Division (DIV) bits in the CPU Clock Division register CPUDIV, resulting in a CPU clock domain frequency determined by this equation:

$$f_{CPU} = \frac{f_{main}}{CPUDIV}$$

If the application attempts to write forbidden values in CPUDIV, LPDIV registers, registers are written but these bad values are not used and a violation is reported to the PAC module.

Division bits (DIV) can be written without halting or disabling peripheral modules. Writing DIV bits allows a new clock setting to be written to all synchronous clocks belonging to the corresponding clock domain at the same time.

Figure 18-2. Synchronous Clock Selection and Prescaler

Related Links

PAC - Peripheral Access Controller on page 58
Electrical Characteristics on page 1140

18.6.2.5. Clock Ready Flag

There is a slight delay between writing to CPUDIV, LPDIV, until the new clock settings become effective.

During this interval, the Clock Ready flag in the Interrupt Flag Status and Clear register (INTFLAG.CKRDY) will return zero when read. If CKRDY in the INTENSET register is set to '1', the Clock Ready interrupt will be triggered when the new clock setting is effective. The clock settings (CLKCFG) must not be re-written while INTFLAG. CKRDY reads '0'. The system may become unstable or hang, and a violation is reported to the PAC module.

Related Links

PAC - Peripheral Access Controller on page 58

18.6.2.6. Peripheral Clock Masking

It is possible to disable/enable the AHB or APB clock for a peripheral by writing the corresponding bit in the Clock Mask registers (APBxMASK) to '0'/'1'. The default state of the peripheral clocks is shown here.

Table 18-1. Peripheral Clock Default State

CPU Clock Domain	
Peripheral Clock	**Default State**
CLK_BRIDGE_B_AHB	Enabled
CLK_DSU_AHB	Enabled
CLK_DSU_APB	Enabled
CLK_USB_AHB	Enabled
CLK_USB_APB	Enabled
CLK_NVMCTRL_AHB	Enabled
CLK_NVMCTRL_APB	Enabled

Backup Clock Domain	
Peripheral Clock	**Default State**
CLK_OSC32KCTRL_APB	Enabled
CLK_PM_APB	Enabled
CLK_SUPC_APB	Enabled
CLK_RSTC_APB	Enabled
CLK_RTC_APB	Enabled

Low Power Clock Domain	
Peripheral Clock	**Default State**
CLK_AC_APB	Enabled
CLK_ADC_APB	Enabled
CLK_AES_APB	Enabled
CLK_BRIDGE_A_AHB	Enabled
CLK_BRIDGE_C_AHB	Enabled
CLK_BRIDGE_D_AHB	Enabled
CLK_BRIDGE_E_AHB	Enabled
CLK_CCL_APB	Enabled
CLK_DAC_APB	Enabled
CLK_DMAC_AHB	Enabled
CLK_EIC_APB	Enabled
CLK_EVSYS_APB	Enabled
CLK_GCLK_APB	Enabled
CLK_MCLK_APB	Enabled
CLK_OPAMP_APB	Enabled
CLK_OSCCTRL_APB	Enabled
CLK_PAC_AHB	Enabled
CLK_PAC_APB	Enabled
CLK_PORT_APB	Enabled
CLK_PTC_APB	Enabled
CLK_SERCOM0_APB	Enabled
CLK_SERCOM1_APB	Enabled
CLK_SERCOM2_APB	Enabled
CLK_SERCOM3_APB	Enabled

Low Power Clock Domain	
Peripheral Clock	**Default State**
CLK_SERCOM4_APB	Enabled
CLK_SERCOM5_APB	Enabled
CLK_TCC0_APB	Enabled
CLK_TCC1_APB	Enabled
CLK_TCC2_APB	Enabled
CLK_TC0_APB	Enabled
CLK_TC1_APB	Enabled
CLK_TC2_APB	Enabled
CLK_TC3_APB	Enabled
CLK_TC4_APB	Enabled
CLK_TRNG_APB	Enabled
CLK_WDT_APB	Enabled

When the APB clock is not provided to a module, its registers cannot be read or written. The module can be re-enabled later by writing the corresponding mask bit to '1'.

A module may be connected to several clock domains (for instance, AHB and APB), in which case it will have several mask bits.

Note that clocks should only be switched off if it is certain that the module will not be used: Switching off the clock for the NVM Controller (NVMCTRL) will cause a problem if the CPU needs to read from the Flash Memory. Switching off the clock to the MCLK module (which contains the mask registers) or the corresponding APBx bridge, will make it impossible to write the mask registers again. In this case, they can only be re-enabled by a system reset.

18.6.3. DMA Operation

Not applicable.

18.6.4. Interrupts

The peripheral has the following interrupt sources:

- Clock Ready (CKRDY): indicates that CPU, LP, clocks are ready. This interrupt is a synchronous wake-up source.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. Each interrupt can be enabled individually by writing a '1' to the corresponding enabling bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a '1' to the corresponding clearing bit in the Interrupt Enable Clear (INTENCLR) register.

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled or the peripheral is reset. An interrupt flag is cleared by writing a '1' to the corresponding bit in the INTFLAG register. Each peripheral can have one interrupt request line per interrupt source or one common interrupt request line for all the interrupt sources.If the peripheral has one common interrupt request line for all the

interrupt sources, the user must read the INTFLAG register to determine which interrupt condition is present.

Related Links

Overview on page 51

PM – Power Manager on page 186

Sleep Mode Controller on page 191

18.6.5. Events

Not applicable.

18.6.6. Sleep Mode Operation

In all IDLE sleep modes, the MCLK is still running on the selected main clock.

In STANDBY sleep mode, the MCLK is frozen if no synchronous clock is required.

18.7. Register Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0								
0x01	INTENCLR	7:0								CKRDY
0x02	INTENSET	7:0								CKRDY
0x03	INTFLAG	7:0								CKRDY
0x04	Reserved									
0x05	CPUDIV	7:0	CPUDIV[7:0]							
0x06 ... 0x0F	Reserved									
0x10	AHBMASK	7:0	Reserved	Reserved	DSU	APBE	APBD	APBC	APBB	APBA
0x11		15:8		PAC	Reserved	USB	DMAC	Reserved	Reserved	NVMCTRL
0x12		23:16								
0x13		31:24								
0x14	APBAMASK	7:0	WDT	GCLK	SUPC	OSC32KCTRL	OSCCTRL	RSTC	MCLK	PM
0x15		15:8	Reserved[3:0]					PORT	EIC	RTC
0x16		23:16	Reserved[11:4]							
0x17		31:24	Reserved[19:12]							
0x18	APBBMASK	7:0	Reserved[4:0]					NVMCTRL	DSU	USB
0x19		15:8	Reserved[12:5]							
0x1A		23:16	Reserved[20:13]							
0x1B		31:24	Reserved[28:21]							
0x1C	APBCMASK	7:0	TCC2	TCC1	TCC0	SERCOM4	SERCOM3	SERCOM2	SERCOM1	SERCOM0
0x1D		15:8		TRNG	AES	DAC	TC3	TC2	TC1	TC0
0x1E		23:16								
0x1F		31:24								
0x20	APBDMASK	7:0	CCL	OPAMP	PTC	AC	ADC	TC4	SERCOM5	EVSYS
0x21		15:8								
0x22		23:16								
0x23		31:24								
0x24	APBEMASK	7:0	Reserved[6:0]							PAC
0x25		15:8	Reserved[14:7]							
0x26		23:16	Reserved[22:15]							
0x27		31:24	Reserved[30:23]							

18.8. Register Description

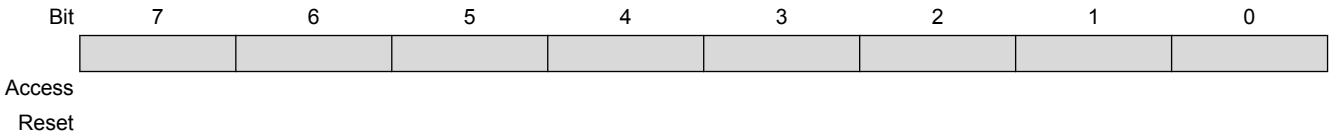
Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers can be write-protected optionally by the Peripheral Access Controller (PAC). This is denoted by the property "PAC Write-Protection" in each individual register description. Refer to the Register Access Protection on page 150 for details.

18.8.1. Control A

All bits in this register are reserved.

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection



18.8.2. Interrupt Enable Clear

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Name: INTENCLR
Offset: 0x01
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								CKRDY
Access								R/W
Reset								0

Bit 0 – CKRDY: Clock Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Clock Ready Interrupt Enable bit and the corresponding interrupt request.

Value	Description
0	The Clock Ready interrupt is disabled.
1	The Clock Ready interrupt is enabled and will generate an interrupt request when the Clock Ready Interrupt Flag is set.

18.8.3. Interrupt Enable Set

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Name: INTENSET
Offset: 0x02
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								CKRDY
Access								R/W
Reset								0

Bit 0 – CKRDY: Clock Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Clock Ready Interrupt Enable bit and enable the Clock Ready interrupt.

Value	Description
0	The Clock Ready interrupt is disabled.
1	The Clock Ready interrupt is enabled.

18.8.4. Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x03
Reset: 0x01
Property: –

Bit	7	6	5	4	3	2	1	0
								CKRDY
Access								R/W
Reset								1

Bit 0 – CKRDY: Clock Ready
This flag is cleared by writing a '1' to the flag.

This flag is set when the synchronous CPU, APBx, and AHBx clocks have frequencies as indicated in the CLKCFG registers and will generate an interrupt if INTENCLR/SET.CKRDY is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Clock Ready interrupt flag.

18.8.5. CPU Clock Division

Name: CPUDIV
Offset: 0x05
Reset: 0x01
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	CPUDIV[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	1

Bits 7:0 – CPUDIV[7:0]: CPU Clock Division Factor
These bits define the division ratio of the main clock prescaler related to the CPU clock domain.

To ensure correct operation, frequencies must be selected so that $F_{CPU} \geq F_{LP}$ (i.e. LPDIV ≥CPUDIV).

Frequencies must never exceed the specified maximum frequency for each clock domain.

Value	Name	Description
0x01	DIV1	Divide by 1
0x02	DIV2	Divide by 2
0x04	DIV4	Divide by 4
0x08	DIV8	Divide by 8
0x10	DIV16	Divide by 16
0x20	DIV32	Divide by 32
0x40	DIV64	Divide by 64
0x80	DIV128	Divide by 128
others	-	Reserved

18.8.6. Low Power Clock Division

Name: LPDIV
Offset: 0x05
Reset: 0x01
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	LPDIV[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	1

Bits 7:0 – LPDIV[7:0]: Low-Power Clock Division Factor
These bits define the division ratio of the main clock prescaler (2^n) related to the Low Power clock domain. To ensure correct operation, frequencies must be selected so that $F_{CPU} \geq F_{LP} \geq F_{\_BUP}$ (i.e. BUPDIV ≥ LPDIV ≥ CPUDIV). Also, frequencies must never exceed the specified maximum frequency for each clock domain.

Value	Name	Description
0x01	DIV1	Divide by 1
0x02	DIV2	Divide by 2
0x04	DIV4	Divide by 4
0x08	DIV8	Divide by 8
0x10	DIV16	Divide by 16
0x20	DIV32	Divide by 32
0x40	DIV64	Divide by 64
0x80	DIV128	Divide by 128
others	-	Reserved

18.8.7. AHB Mask

Name: AHBMASK
Offset: 0x10
Reset: 0x000FFFFF
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		PAC	Reserved	USB	DMAC	Reserved	Reserved	NVMCTRL
Access		R	R	R/W	R/W	R	R	R/W
Reset		1	1	1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	DSU	APBE	APBD	APBC	APBB	APBA
Access	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit 14 – PAC: PAC AHB Clock Enable

Value	Description
0	The AHB clock for the PAC is stopped.
1	The AHB clock for the PAC is enabled.

Bit 12 – USB: USB AHB Clock Enable

Value	Description
0	The AHB clock for the USB is stopped.
1	The AHB clock for the USB is enabled.

Bit 11 – DMAC: DMAC AHB Clock Enable

Value	Description
0	The AHB clock for the DMAC is stopped.
1	The AHB clock for the DMAC is enabled.

Bit 8 – NVMCTRL: NVMCTRL AHB Clock Enable

Value	Description
0	The AHB clock for the NVMCTRL is stopped.
1	The AHB clock for the NVMCTRL is enabled.

Bit 5 – DSU: DSU AHB Clock Enable

Value	Description
0	The AHB clock for the DSU is stopped.
1	The AHB clock for the DSU is enabled.

Bit 4 – APBE: APBE AHB Clock Enable

Value	Description
0	The AHB clock for the APBE is stopped.
1	The AHB clock for the APBE is enabled.

Bit 3 – APBD: APBD AHB Clock Enable

Value	Description
0	The AHB clock for the APBD is stopped.
1	The AHB clock for the APBD is enabled

Bit 2 – APBC: APBC AHB Clock Enable

Value	Description
0	The AHB clock for the APBC is stopped.
1	The AHB clock for the APBC is enabled

Bit 1 – APBB: APBB AHB Clock Enable

Value	Description
0	The AHB clock for the APBB is stopped.
1	The AHB clock for the APBB is enabled.

Bit 0 – APBA: APBA AHB Clock Enable

Value	Description
0	The AHB clock for the APBA is stopped.
1	The AHB clock for the APBA is enabled.

Bits 13,10,9,7,6 – Reserved: Reserved bits
Reserved bits are unused and reserved for future use. For compatibility with future devices, always write reserved bits to their reset value. If no reset value is given, write 0.

18.8.8. APBA Mask

Name: APBAMASK
Offset: 0x14
Reset: 0x00001FFF
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	Reserved[19:12]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	Reserved[11:4]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	Reserved[3:0]					PORT	EIC	RTC
Access	R	R	R	R		R	R	R
Reset	0	0	0	1		1	1	1

Bit	7	6	5	4	3	2	1	0
	WDT	GCLK	SUPC	OSC32KCTRL	OSCCTRL	RSTC	MCLK	PM
Access	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 31:12 – Reserved[19:0]: For future use
Reserved bits are unused and reserved for future use. For compatibility with future devices, always write reserved bits to their reset value. If no reset value is given, write 0.

Bit 10 – PORT: PORT APBA Clock Enable

Value	Description
0	The APBA clock for the PORT is stopped.
1	The APBA clock for the PORT is enabled.

Bit 9 – EIC: EIC APBA Clock Enable

Value	Description
0	The APBA clock for the EIC is stopped.
1	The APBA clock for the EIC is enabled.

Bit 8 – RTC: RTC APBA Clock Enable

Value	Description
0	The APBA clock for the RTC is stopped.
1	The APBA clock for the RTC is enabled.

Bit 7 – WDT: WDT APBA Clock Enable

Value	Description
0	The APBA clock for the WDT is stopped.
1	The APBA clock for the WDT is enabled.

Bit 6 – GCLK: GCLK APBA Clock Enable

Value	Description
0	The APBA clock for the GCLK is stopped.
1	The APBA clock for the GCLK is enabled.

Bit 5 – SUPC: SUPC APBA Clock Enable

Value	Description
0	The APBA clock for the SUPC is stopped.
1	The APBA clock for the SUPC is enabled.

Bit 4 – OSC32KCTRL: OSC32KCTRL APBA Clock Enable

Value	Description
0	The APBA clock for the OSC32KCTRL is stopped.
1	The APBA clock for the OSC32KCTRL is enabled.

Bit 3 – OSCCTRL: OSCCTRL APBA Clock Enable

Value	Description
0	The APBA clock for the OSCCTRL is stopped.
1	The APBA clock for the OSCCTRL is enabled.

Bit 2 – RSTC: RSTC APBA Clock Enable

Value	Description
0	The APBA clock for the RSTC is stopped.
1	The APBA clock for the RSTC is enabled.

Bit 1 – MCLK: MCLK APBA Clock Enable

Value	Description
0	The APBA clock for the MCLK is stopped.
1	The APBA clock for the MCLK is enabled.

Bit 0 – PM: PM APBA Clock Enable

Value	Description
0	The APBA clock for the PM is stopped.
1	The APBA clock for the PM is enabled.

18.8.9. APBB Mask

Name: APBBMASK
Offset: 0x18
Reset: 0x00000017
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	Reserved[28:21]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	Reserved[20:13]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	Reserved[12:5]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	Reserved[4:0]					NVMCTRL	DSU	USB
Access	R	R	R	R	R	R/W	R/W	R/W
Reset	0	0	0	1	0	1	1	1

Bits 31:3 – Reserved[28:0]: Reserved bits
Reserved bits are unused and reserved for future use. For compatibility with future devices, always write reserved bits to their reset value. If no reset value is given, write 0.

Bit 2 – NVMCTRL: NVMCTRL APBB Clock Enable

Value	Description
0	The APBB clock for the NVMCTRL is stopped
1	The APBB clock for the NVMCTRL is enabled

Bit 1 – DSU: DSU APBB Clock Enable

Value	Description
0	The APBB clock for the DSU is stopped
1	The APBB clock for the DSU is enabled

Bit 0 – USB: USB APBB Clock Enable

Value	Description
0	The APBB clock for the USB is stopped
1	The APBB clock for the USB is enabled

18.8.10. APBC Mask

Name: APBCMASK
Offset: 0x1C
Reset: 0x0000 7FFF
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		TRNG	AES	DAC	TC3	TC2	TC1	TC0
Access		R	R	R	R	R	R	R
Reset		1	1	1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
	TCC2	TCC1	TCC0	SERCOM4	SERCOM3	SERCOM2	SERCOM1	SERCOM0
Access	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit 14 – TRNG: TRNG APBC Mask Clock Enable

Value	Description
0	The APBC clock for the TRNG is stopped.
1	The APBC clock for the TRNG is enabled.

Bit 13 – AES: AES APBC Mask Clock Enable

Value	Description
0	The APBC clock for the AES is stopped.
1	The APBC clock for the AES is enabled.

Bit 12 – DAC: DAC APBC Mask Clock Enable

Value	Description
0	The APBC clock for the DAC is stopped.
1	The APBC clock for the DAC is enabled.

Bit 11 – TC3: TC3 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the TC3 is stopped.
1	The APBC clock for the TC3 is enabled.

Bit 10 – TC2: TC2 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the TC2 is stopped.
1	The APBC clock for the TC2 is enabled.

Bit 9 – TC1: TC1 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the TC1 is stopped.
1	The APBC clock for the TC1 is enabled.

Bit 8 – TC0: TC0 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the TC0 is stopped.
1	The APBC clock for the TC0 is enabled.

Bit 7 – TCC2: TCC2 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the TCC2 is stopped.
1	The APBC clock for the TCC2 is enabled.

Bit 6 – TCC1: TCC1 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the TCC1 is stopped.
1	The APBC clock for the TCC1 is enabled.

Bit 5 – TCC0: TCC0 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the TCC0 is stopped.
1	The APBC clock for the TCC0 is enabled.

Bit 4 – SERCOM4: SERCOM4 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the SERCOM4 is stopped.
1	The APBC clock for the SERCOM4 is enabled.

Bit 3 – SERCOM3: SERCOM3 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the SERCOM3 is stopped.
1	The APBC clock for the SERCOM3 is enabled.

Bit 2 – SERCOM2: SERCOM2 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the SERCOM2 is stopped.
1	The APBC clock for the SERCOM2 is enabled.

Bit 1 – SERCOM1: SERCOM1 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the SERCOM1 is stopped.
1	The APBC clock for the SERCOM1 is enabled.

Bit 0 – SERCOM0: SERCOM0 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the SERCOM0 is stopped.
1	The APBC clock for the SERCOM0 is enabled.

18.8.11. APBD Mask

Name: APBDMASK
Offset: 0x20
Reset: 0x000000FF
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	CCL	OPAMP	PTC	AC	ADC	TC4	SERCOM5	EVSYS
Access	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit 7 – CCL: CCL APBD Clock Enable

Value	Description
0	The APBD clock for the CCL is stopped.
1	The APBD clock for the CCL is enabled.

Bit 6 – OPAMP: OPAMP APBD Clock Enable

Value	Description
0	The APBD clock for the OPAMP is stopped.
1	The APBD clock for the OPAMP is enabled.

Bit 5 – PTC: PTC APBD Clock Enable

Value	Description
0	The APBD clock for the PTC is stopped.
1	The APBD clock for the PTC is enabled.

Bit 4 – AC: AC APBD Clock Enable

Value	Description
0	The APBD clock for the AC is stopped.
1	The APBD clock for the AC is enabled.

Bit 3 – ADC: ADC APBD Clock Enable

Value	Description
0	The APBD clock for the ADC is stopped.
1	The APBD clock for the ADC is enabled.

Bit 2 – TC4: TC4 APBD Clock Enable

Value	Description
0	The APBD clock for the TC4 is stopped.
1	The APBD clock for the TC4 is enabled.

Bit 1 – SERCOM5: SERCOM5 APBD Clock Enable

Value	Description
0	The APBD clock for the SERCOM5 is stopped.
1	The APBD clock for the SERCOM5 is enabled.

Bit 0 – EVSYS: EVSYS APBD Clock Enable

Value	Description
0	The APBD clock for the EVSYS is stopped.
1	The APBD clock for the EVSYS is enabled.

18.8.12. APBE Mask

Name: APBEMASK
Offset: 0x24
Reset: 0x0000 000D
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	Reserved[30:23]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	Reserved[22:15]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	Reserved[14:7]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	Reserved[6:0]							PAC
Access	R	R	R	R	R	R	R	R/W
Reset	0	0	0	0	1	1	0	1

Bits 31:1 – Reserved[30:0]: For future use
Reserved bits are unused and reserved for future use. For compatibility with future devices, always write reserved bits to their reset value. If no reset value is given, write 0.

Bit 0 – PAC: PAC APBE Clock Enable

Value	Description
0	The APBE clock for the PAC is stopped.
1	The APBE clock for the PAC is enabled.

19. RSTC – Reset Controller

19.1. Overview

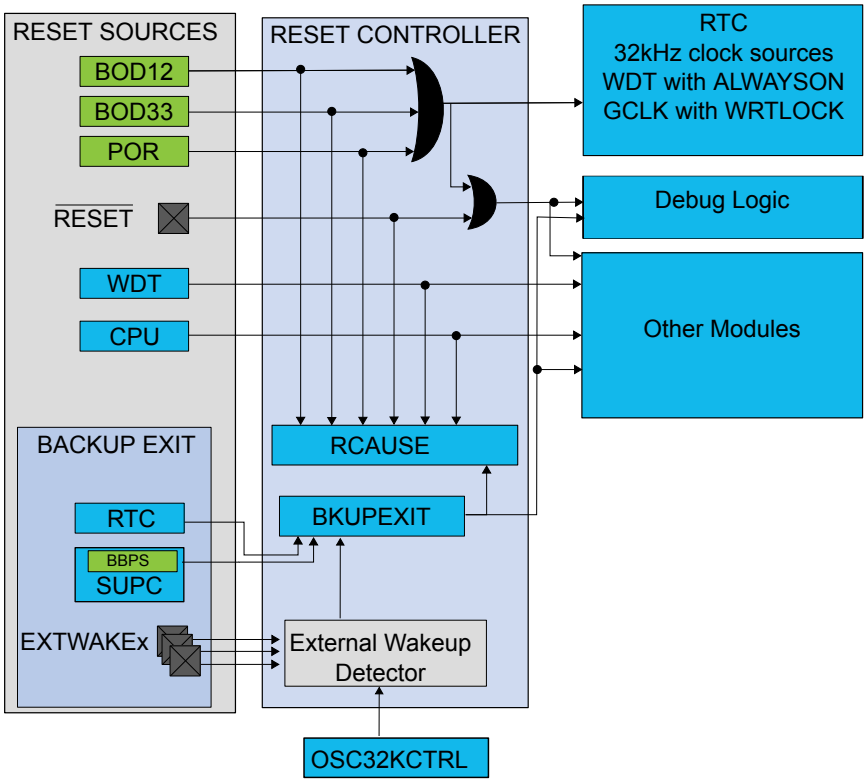
The Reset Controller (RSTC) manages the reset of the microcontroller. It issues a microcontroller reset, sets the device to its initial state and allows the reset source to be identified by software.

19.2. Features

- Reset the microcontroller and set it to an initial state according to the reset source
- Reset cause register for reading the reset source from the application code
- Multiple reset sources
  - Power supply reset sources: POR, BOD12, BOD33
  - User reset sources: External reset ($\overline{\text{RESET}}$), Watchdog reset, and System Reset Request
  - Backup exit sources: Real-Time Counter (RTC), External Wake-up (EXTWAKE), and Battery Backup Power Switch (BBPS)

19.3. Block Diagram

Figure 19-1. Reset System



19.4. Signal Description

Signal Name	Type	Description
$\overline{\text{RESET}}$	Digital input	External reset
EXTWAKE[7:0]	Digital input	External wakeup for backup mode

One signal can be mapped on several pins.

Related Links

I/O Multiplexing and Considerations on page 29

19.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

19.5.1. I/O Lines

Using the External Wake-up Lines requires the I/O pins to be configured in input mode before entering backup mode. External Wake-up function is active only in backup mode.

Caution: The EXTWAKE pins can not wake up the device after it has entered Battery Backup Mode, as the I/O pin configuration is lost in this mode.

19.5.2. Power Management

The Reset Controller module is always on.

19.5.3. Clocks

The RSTC bus clock (CLK_RSTC_APB) can be enabled and disabled in the Main Clock Controller.

A 32KHz clock is required to clock the RSTC if the debounce counter of the external wake-up detector is used.

Related Links

MCLK – Main Clock on page 148
Peripheral Clock Masking on page 151
OSC32KCTRL – 32KHz Oscillators Controller on page 264

19.5.4. DMA

Not applicable.

19.5.5. Interrupts

Not applicable.

19.5.6. Events

Not applicable.

19.5.7. Debug Operation

When the CPU is halted in debug mode, the RSTC continues normal operation.

19.5.8. Register Access Protection

All registers with write-access can be optionally write-protected by the Peripheral Access Controller (PAC).

Note: Optional write-protection is indicated by the "PAC Write-Protection" property in the register description.

When the CPU is halted in debug mode, all write-protection is automatically disabled. Write-protection does not apply for accesses through an external debugger.

19.5.9. Analog Connections

Not applicable.

19.6. Functional Description

19.6.1. Principle of Operation

The Reset Controller collects the various Reset sources and generates Reset for the device. External Wakeup lines causing a Backup Reset from the Backup Sleep Mode can be filtered using the debounce counter.

19.6.2. Basic Operation

19.6.2.1. Initialization

After a power-on Reset, the RSTC is enabled and the Reset Cause (RCAUSE) register indicates the POR source.

19.6.2.2. Enabling, Disabling, and Resetting

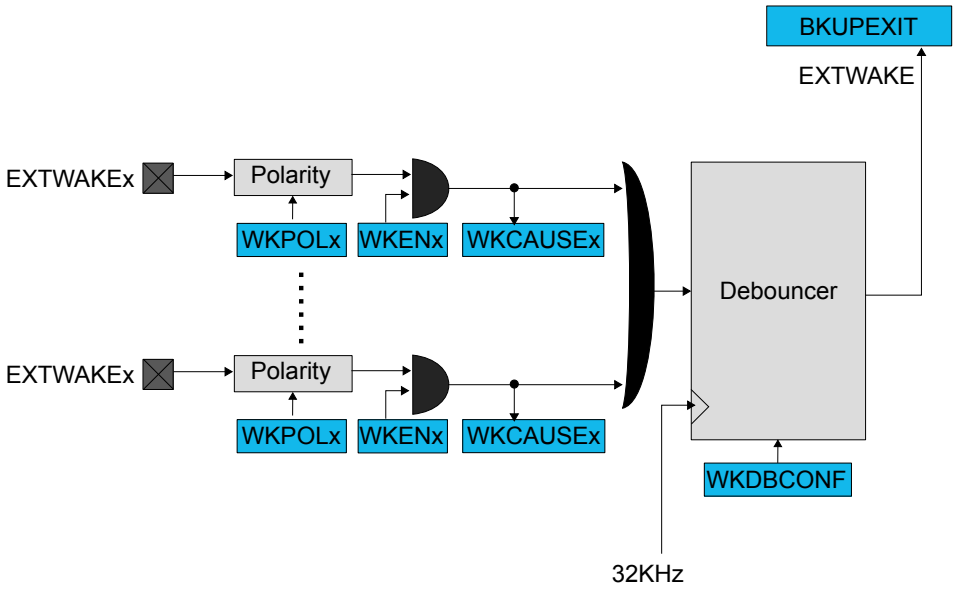
The RSTC module is always enabled.

19.6.2.3. External Wake-Up Detector

The External Wake-up detector is activated in Backup Sleep Mode only. In all other sleep modes, the debounce counter is stopped. Before entering Backup Mode, each external wake-up pin can be enabled by configuring the Wake-up Enable (WKEN) register. The corresponding I/O lines must also be configured in input mode using port configuration (PORT). The wake-up level can also be configured by using the Wake-up Polarity (WKPOL) register. If WKPOLx is written to 0 (default value), the input wake-up pin is active low. If WKPOLx=1 the pin is active high. All the resulting signals are wired-ORed to trigger a debounce counter which can be programmed with the Wake-up Debounce Configuration (WKDBCONF) register.

In Backup Mode, the debounce counter is running if at least one external wake-up pin is enabled and the WKDBCONF is configured to any other value than OFF. It is clocked by the OSCULP32K clock provided by the OSC32KCTRL module. If an enabled wake-up pin is asserted for a time longer than the debouncing period, the BKUPEXIT.EXTWAKE bit is set, and the value of each enable external wake-up pin is stored in the WKCAUSE register. This will allow the application to identify the external wake-up source when booting up from a backup exit reset. A backup reset is then applied. Refer to Reset Causes and Effects on page 177 for details.

Figure 19-2. External Wake-up Block Diagram



Related Links

OSC32KCTRL – 32KHz Oscillators Controller on page 264

19.6.2.4. Reset Causes and Effects

The latest Reset cause is available in RCAUSE register, and can be read during the application boot sequence in order to determine proper action.

These are the groups of Reset sources:

- Power supply Reset: Resets caused by an electrical issue. It covers POR and BODs Resets
- User Reset: Resets caused by the application. It covers external Resets, system Reset requests and watchdog Resets
- Backup reset: Resets caused by a Backup Mode exit condition

The following table lists the parts of the device that are reset, depending on the Reset type.

Table 19-1. Effects of the Different Reset Causes

	Power Supply Reset		User Reset		Backup Reset
	POR, BOD33	BOD12	External Reset	WDT Reset, System Reset Request	RTC, EXTWAKE, BBPS
RTC, OSC32KCTRL, RSTC, CTRLA.IORET bit of PM	Y	N	N	N	N
GCLK with WRTLOCK	Y	Y	N	N	Y
Debug logic	Y	Y	Y	N	Y
Others	Y	Y	Y	Y	Y

The external Reset is generated when pulling the $\overline{\text{RESET}}$ pin low.

The POR, BOD12, and BOD33 Reset sources are generated by their corresponding module in the Supply Controller Interface (SUPC).

The WDT Reset is generated by the Watchdog Timer.

The System Reset Request is a Reset generated by the CPU when asserting the SYSRESETREQ bit located in the Reset Control register of the CPU (for details refer to the ARM® Cortex™ Technical Reference Manual on http://www.arm.com).

From Backup Mode, the chip can be waken-up upon these conditions:

- Battery Backup Power Switch (BBPS): generated by the SUPC controller when the 3.3V VDDIO is restored.
- External wake up (EXTWAKEn): internally generated by the RSTC.
- Real-Time Counter interrupt. For details refer to the applicable INTFLAG in the RTC for details.

If one of these conditions is triggered in Backup Mode, the RCAUSE.BACKUP bit is set and the Backup Exit Register (BKUPEXIT) is updated.

Related Links

SUPC – Supply Controller on page 283
Battery Backup Power Switch on page 288

19.6.3. Additional Features

Not applicable.

19.6.4. DMA Operation

Not applicable.

19.6.5. Interrupts

Not applicable.

19.6.6. Events

Not applicable.

19.6.7. Sleep Mode Operation

The RSTC module is active in all sleep modes.

19.7. Register Summary

Offset	Name	Bit Pos.								
0x00	RCAUSE	7:0	BACKUP	SYST	WDT	EXT		BOD33	BOD12	POR
0x01	Reserved									
0x02	BKUPEXIT	7:0						BBPS	RTC	EXTWAKE
0x03	Reserved									
0x04	WKDBCONF	7:0				WKDBCNT[4:0]				
0x05 ... 0x07	Reserved									
0x08	WKPOL	7:0	WKPOL[7:0]							
0x09		15:8								
0x0A ... 0x0B	Reserved									
0x0C	WKEN	7:0	WKEN[7:0]							
0x0D		15:8								
0x0E ... 0x0F	Reserved									
0x10	WKCAUSE	7:0	WKCAUSE[7:0]							
0x11		15:8								

19.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection on page 176.

19.8.1. Reset Cause

When a Reset occurs, the bit corresponding to the Reset source is set to '1' and all other bits are written to '0'.

Name: RCAUSE
Offset: 0x00
Reset: Latest Reset Source
Property: –

Bit	7	6	5	4	3	2	1	0
	BACKUP	SYST	WDT	EXT		BOD33	BOD12	POR
Access	R	R	R	R		R	R	R
Reset	x	x	x	x		x	x	x

Bit 7 – BACKUP: Backup Reset
This bit is set if a Backup Reset has occurred. Refer to BKUPEXIT register to identify the source of the Backup Reset.

Bit 6 – SYST: System Reset Request
This bit is set if a System Reset Request has occurred. Refer to the Cortex processor documentation for more details.

Bit 5 – WDT: Watchdog Reset
This bit is set if a Watchdog Timer Reset has occurred.

Bit 4 – EXT: External Reset
This bit is set if an external Reset has occurred.

Bit 2 – BOD33: Brown Out 33 Detector Reset
This bit is set if a BOD33 Reset has occurred.

Bit 1 – BOD12: Brown Out 12 Detector Reset
This bit is set if a BOD12 Reset has occurred.

Bit 0 – POR: Power On Reset
This bit is set if a POR has occurred.

19.8.2. Backup Exit Source

When a Backup Reset occurs, the bit corresponding to the exit condition is set to '1', the other bits are written to '0'.

In some specific cases, the RTC and BBPS bits can be set together, e.g. when the device leaves the battery Backup Mode caused by a BBPS condition, and a RTC event was generated during the Battery Backup Mode period.

Name: BKUPEXIT
Offset: 0x02
Reset: Latest Backup Exit Source
Property: –

Bit	7	6	5	4	3	2	1	0
						BBPS	RTC	EXTWAKE
Access						R	R	R
Reset						x	x	x

Bit 2 – BBPS: Battery Backup Power Switch
This bit is set if the Battery Backup Power Switch of the Supply Controller changes back from battery mode to main power mode.

Bit 1 – RTC: Real Timer Counter Interrupt
This bit is set if an RTC interrupt flag is set in Backup Mode.

Bit 0 – EXTWAKE: External Wake-up
This bit is set if the wake-up detector has detected an external wake-up condition in Backup Mode.

19.8.3. Wakeup Debounce Configuration

Name: WKDBCONF
Offset: 0x04
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
				WKDBCNT[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bits 4:0 – WKDBCNT[4:0]: Wakeup Debounce Counter Value
These bits define the Debounce Mode used when waking up by external wakeup pin from Backup Mode.

WKDBCNT	Name	Description
0x00	OFF	No debouncing. Input pin is low or high level sensitive depending on its WKPOLx bit.
0x01	2CK32	Input pin shall be active for at least two 32KHz clock periods.
0x02	3CK32	Input pin shall be active for at least three 32KHz clock periods.
0x03	32CK32	Input pin shall be active for at least 32 32KHz clock periods.
0x04	512CK32	Input pin shall be active for at least 512 32KHz clock periods.
0x05	4096CK32	Input pin shall be active for at least 4096 32KHz clock periods.
0x06	32768CK32	Input pin shall be active for at least 32768 32KHz clock periods.
0x07	-	Reserved

19.8.4. Wakeup Polarity

Name: WKPOL
Offset: 0x08
Reset: 0x0000
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	WKPOL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

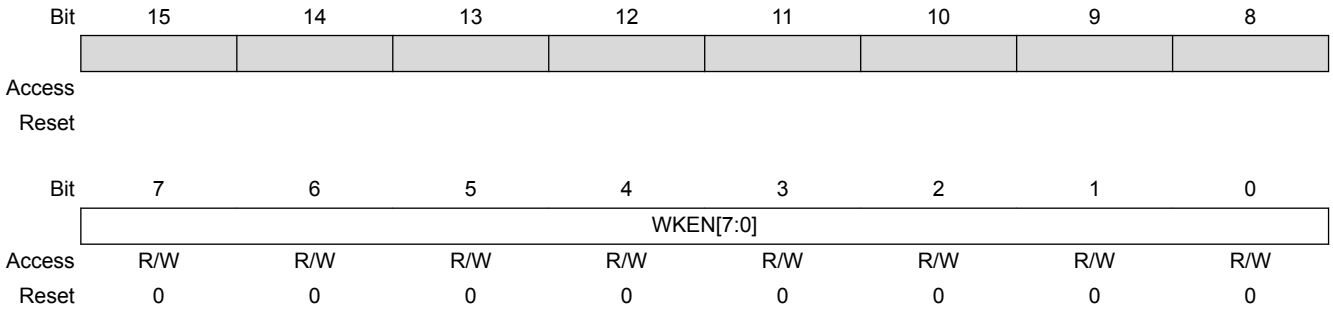
Bits 7:0 – WKPOL[7:0]: Wakeup Polarity
These bits define the polarity of each wakeup input pin.

Value	Description
0	Input pin x is active low.
1	Input pin x is active high.

19.8.5. Wakeup Enable

These bits enable wakeup for input pins from Backup Mode.

Name: WKEN
Offset: 0x0C
Reset: 0x0000
Property: PAC Write-Protection



Bits 7:0 – WKEN[7:0]: Wakeup Enable

Value	Description
0	The wakeup for input pin x from backup mode is disabled.
1	The wakeup for input pin x from backup mode is enabled.

19.8.6. Wakeup Cause

Name: WKCAUSE
Offset: 0x10
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	WKCAUSE[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – WKCAUSE[7:0]: Wakeup Cause x
This bit is updated when exiting Backup Mode.

Value	Description
0	Input pin x is not active or WKENx is written to '0'.
1	Input pin x is active and WKENx is written to '1'.

20. PM – Power Manager

20.1. Overview

The Power Manager (PM) controls the sleep modes and the power domain gating of the device.

Various sleep modes are provided in order to fit power consumption requirements. This enables the PM to stop unused modules in order to save power. In active mode, the CPU is executing application code. When the device enters a sleep mode, program execution is stopped and some modules and clock domains are automatically switched off by the PM according to the sleep mode. The application code decides which sleep mode to enter and when. Interrupts from enabled peripherals and all enabled reset sources can restore the device from a sleep mode to active mode.

Performance level technique consists of adjusting the regulator output voltage to reduce power consumption. The user can select on the fly the performance level configuration which best suits the application.

The power domain gating technique enables the PM to turn off unused power domain supplies individually, while keeping others powered up. Based on activity monitoring, power domain gating is managed automatically by hardware without software intervention. This technique is transparent for the application while minimizing the static consumption. The user can also manually control which power domains will be turned on and off in standby sleep mode.

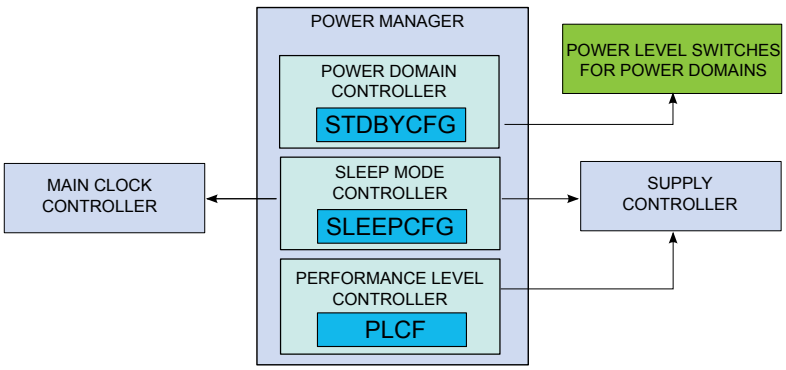
The internal state of the logic is retained (retention state) allowing the application context to be kept in non-active states.

20.2. Features

- Power management control
  - Sleep modes: Idle, Standby
  - Two performance levels
  - SleepWalking available in standby mode.
  - Full retention state in Standby mode

20.3. Block Diagram

Figure 20-1. PM Block Diagram



20.4. Signal Description

Not applicable.

20.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

20.5.1. I/O Lines

Not applicable.

20.5.2. Clocks

The PM bus clock (CLK_PM_APB) can be enabled and disabled in the Main Clock module. If this clock is disabled, it can only be re-enabled by a system reset.

20.5.3. DMA

Not applicable.

20.5.4. Interrupts

The interrupt request line is connected to the interrupt controller. Using the PM interrupt requires the interrupt controller to be configured first.

20.5.5. Events

Not applicable.

20.5.6. Debug Operation

When the CPU is halted in debug mode, the PM continues normal operation. If standby sleep mode is requested by the system while in debug mode, the power domains are not turned off. As a consequence, power measurements while in debug mode are not relevant.

Hot plugging in standby mode is supported except if the power domain PD2 is in retention state.

20.5.7. Register Access Protection

Registers with write-access can be write-protected optionally by the peripheral access controller (PAC).

PAC Write-Protection is not available for the following registers:

- Interrupt Flag register (INTFLAG). Refer to INTFLAG on page 214 for details

Optional PAC Write-Protection is denoted by the "PAC Write-Protection" property in each individual register description.

Write-protection does not apply to accesses through an external debugger.

Related Links

PAC - Peripheral Access Controller on page 58

20.5.8. Analog Connections

Not applicable.

20.6. Functional Description

20.6.1. Terminology

The following is a list of terms used to describe the Power Managemement features of this microcontroller.

20.6.1.1. Performance Levels

To help balance between performance and power consumption, the device has two performance levels. Each of the performance levels has a maximum operating frequency and a corresponding maximum consumption in µA/MHz.

It is the application's responsibility to configure the appropriate PL depending on the application activity level. When the application selects a new PL, the voltage applied on the full logic area moves from one value to another. This voltage scaling technique allows to reduce the active power consumption while decreasing the maximum frequency of the device.

PL0

Performance Level 0 (PL0) provides the maximum energy efficiency configuration.

Refer to Electrical Characteristics on page 1140 for details on energy consumption and maximum operating frequency.

PL2

Performance level 2 (PL2) provides the maximum operating frequency.

Refer to Electrical Characteristics on page 1140 for details on energy consumption and maximum operating frequency.

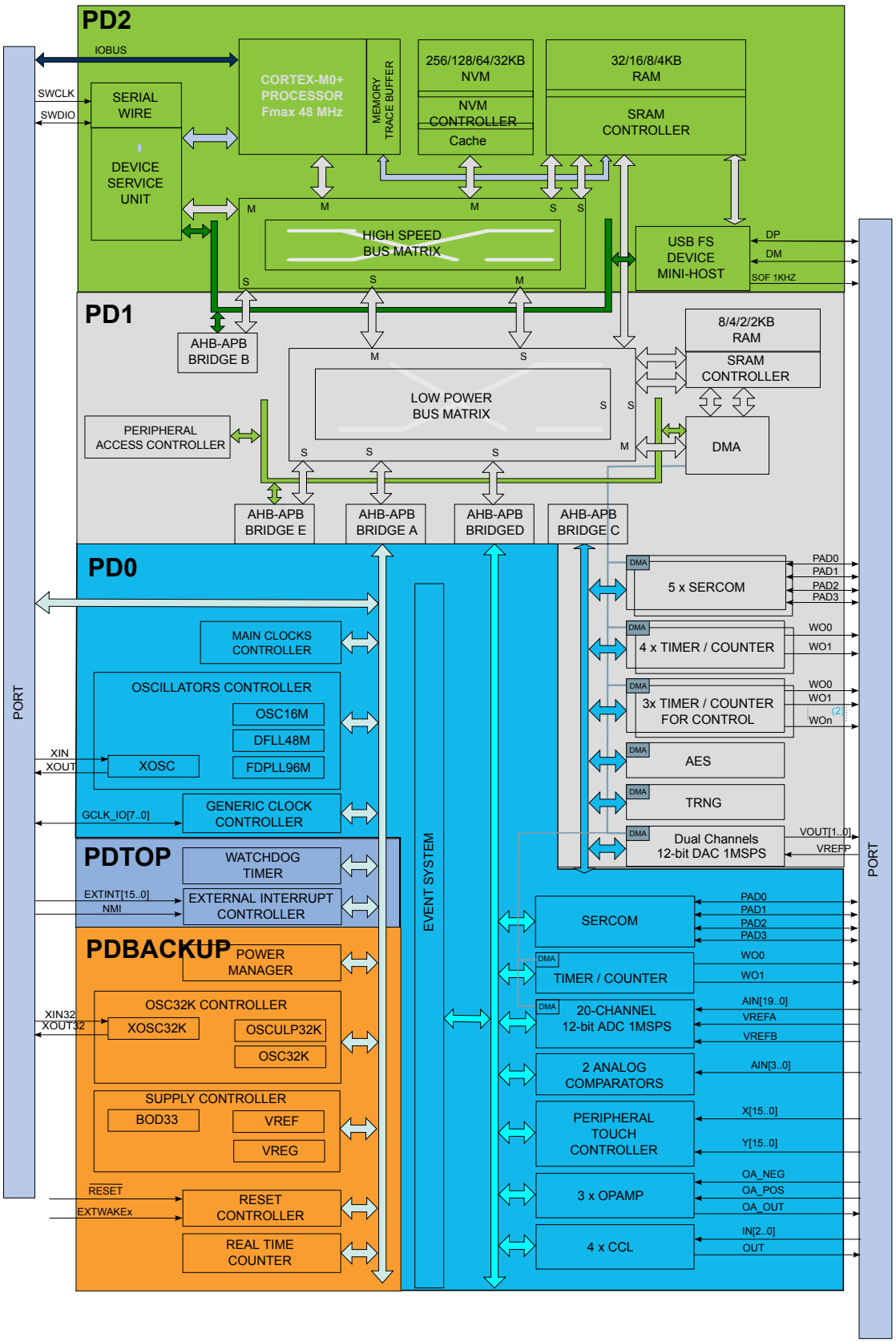
20.6.1.2. Power Domains

Leaving aside the supply domains, such as VDDIO, VDDIN and VDDANA, the device is split into these power domains: PD0, PD1, PD2, PDTOP, and PDBACKUP.

PD0, PD1 and PD2 are "switchable power domains". In standby or backup mode, they can be turned off to save leakage consumption according to user configuration.

The three peripheral domains, PD0, PD1, and PD2, can be in retention state when none of the contained peripherals are required, but if a peripheral power domain PD*n* is powered, lower power domains will be powered, too. For example, if no peripherals are being used in PD2 and one or several peripherals in PD1 are active, then PD2 will be powered down, PD1 will be powered, and PD0 will automatically be powered, even if no peripheral is being used.

Figure 20-2. Power Domain Partitioning



Related Links

Power Domain Overview on page 37

PD0
PD0 is the lowest Power Domain. It contains the Event System, the Generic Clock Controller, Oscillators Controller, the Main Clocks Controller. Additionally, PD0 contains a number of peripherals that allow the device to wake up from an interrupt: one SERCOM (SERCOM5), one Timer/Counter (TC4), ADC, AC, OPAMP, CCL, and the PTC. The PLL oscillator sources, DFLL48M and FDPLL96M, are in PD0 as well.

See also Power Domains on page 188.

This power domain will automatically be activated if either PD1 or PD2 are activated.

PD1
PD1 is the intermediate Power Domain. PD1 contains the DMA controller, the Peripheral Access Controller, and the Low Power Bus Matrix. It also contains the Timer/Counter for Control instances, the AES peripheral, the TRNG, the DAC and the low-power SRAM. PD1 contains the SERCOMs (except for SERCOM5, present in PD0), and the Timer Counters (except TC4, present in PD0).

When active, PD1 automatically activates PD0.

PD2
PD2 is the highest power domain. When activated, it will automatically activate both PD1 and PD0.

It contains the CM0+ core, the Non-Volatile Memory Controller, the Device Service Unit, USB, and the SRAM. See also Power Domains on page 188.

PDTOP
PDTOP contains all controllers located in the core domain. It is powered when in Active, Idle or Standby mode. It does not have a retention mode; it is either in an active state, or off. When in Backup of Off mode, this domain is completely powered down.

PDBACKUP
The Backup Power Domain (PDBACKUP) is always on, except in the off sleep mode. It contains the 32KHz oscillator sources, the Supply Controller, the Reset Controller, the Real Time Counter, and the Power Manager itself.

20.6.1.3. Sleep Modes

The device can be set in a sleep mode. In sleep mode, the CPU is stopped and the peripherals are either active or idle, according to the sleep mode depth:

- Idle sleep mode: The CPU is stopped. Synchronous clocks are stopped except when requested. The logic is retained.
- Standby sleep mode: The CPU is stopped as well as the peripherals. The logic is retained, and power domain gating can be used to reduce power consumption further.

20.6.1.4. Power Domain States and Gating

In Standby sleep mode, the Power Domain Gating technique allows for selecting the state of a PDn power domain automatically (e.g. for executing sleepwalking tasks) or manually:

Active State	The power domain is powered according to the performance level
Retention State	The main voltage supply for the power domain is switched off, while maintaining a secondary low-power supply for sequential cells. The logic context is restored when waking up.
Off State	The power domain is entirely powered off. The logic context is lost.

20.6.2. Principle of Operation

In active mode, all clock domains and power domains are active, allowing software execution and peripheral operation. The PM Sleep Mode Controller allows to save power by choosing between different sleep modes depending on application requirements, see Sleep Mode Controller on page 191.

The PM Performance Level Controller allows to optimize either for low power consumption or high performance.

The PM Power Domain Controller allows to reduce the power consumption in standby mode even further.

20.6.3. Basic Operation

20.6.3.1. Initialization

After a power-on reset, the PM is enabled, the device is in ACTIVE mode, the performance level is PL0 (the lowest power consumption) and all the power domains are in active state.

20.6.3.2. Enabling, Disabling and Resetting

The PM is always enabled and can not be reset.

20.6.3.3. Sleep Mode Controller

A Sleep mode is entered by executing the Wait For Interrupt instruction (WFI). The Sleep Mode bits in the Sleep Configuration register (SLEEPCFG.SLEEPMODE) select the level of the sleep mode.

Note: A small latency happens between the store instruction and actual writing of the SLEEPCFG register due to bridges. Software must ensure that the SLEEPCFG register reads the desired value before issuing a WFI instruction.

Note: After power-up, the MAINVREG low power mode takes some time to stabilize. Once stabilized, the INTFLAG.SLEEPRDY bit is set. Before entering Standby, software must ensure that the INTFLAG.SLEEPRDY bit is set.

Table 20-1. Sleep Mode Entry and Exit Table

Mode	Mode Entry	Wake-Up Sources
IDLE	SLEEPCFG.SLEEPMODE = IDLE _n	Synchronous (2) (APB, AHB), asynchronous (1)
STANDBY	SLEEPCFG.SLEEPMODE = STANDBY	Synchronous(3), Asynchronous

Note:

1. Asynchronous: interrupt generated on generic clock, external clock, or external event.
2. Synchronous: interrupt generated on the APB clock.
3. Synchronous interrupt only for peripherals configured to run in standby.

Note: The type of wake-up sources (synchronous or asynchronous) is given in each module interrupt section.

The sleep modes (idle, standby) and their effect on the clocks activity, the regulator and the NVM state are described in the table and the sections below. Refer to Power Domain Controller on page 194 for the power domain gating effect.

Table 20-2. Sleep Mode Overview

Mode	Main clock	CPU	AHBx and APBx clock	GCLK clocks	Oscillators		Regulator	NVM
					ONDEMAND = 0	ONDEMAND = 1		
Active	Run	Run	Run	Run(3)	Run	Run if requested	MAINVREG	active
IDLE	Run	Stop	Stop(1)	Run(3)	Run	Run if requested	MAINVREG	active
STANDBY	Stop(1)	Stop	Stop(1)	Stop(1)	Run if requested or RUNSTDBY=1	Run if requested	MAINVREG in low power mode	Ultra Low power

Note:

1. Running if requested by peripheral during SleepWalking.
2. Running during SleepWalking.
3. Following On-Demand Clock Request principle.

IDLE Mode

The IDLE mode allows power optimization with the fastest wake-up time.

The CPU is stopped, and peripherals are still working. As in active mode, the AHBx and APBx clocks for peripheral are still provided if requested. As the main clock source is still running, wake-up time is very fast.

- Entering IDLE mode: The IDLE mode is entered by executing the WFI instruction. Additionally, if the SLEEPONEXIT bit in the ARM Cortex System Control register (SCR) is set, the IDLE mode will be entered when the CPU exits the lowest priority ISR (Interrupt Service Routine, see ARM Cortex documentation for details). This mechanism can be useful for applications that only require the processor to run when an interrupt occurs. Before entering the IDLE mode, the user must select the idle Sleep Mode in the Sleep Configuration register (SLEEPCFG.SLEEPMODE=IDLE).
- Exiting IDLE mode: The processor wakes the system up when it detects any non-masked interrupt with sufficient priority to cause exception entry. The system goes back to the ACTIVE mode. The CPU and affected modules are restarted.

GCLK clocks, regulators and RAM are not affected by the idle sleep mode and operate in normal mode.

STANDBY Mode

The STANDBY mode is the lowest power configuration while keeping the state of the logic and the content of the RAM.

In this mode, all clocks are stopped except those configured to be running sleepwalking tasks. The clocks can also be active on request or at all times, depending on their on-demand and run-in-standby settings. Either synchronous (CLK_APBx or CLK_AHBx) or generic (GCLK_x) clocks or both can be involved in sleepwalking tasks. This is the case when for example the SERCOM RUNSTDBY bit is written to '1'.

- Entering STANDBY mode: This mode is entered by executing the WFI instruction after writing the Sleep Mode bit in the Sleep Configuration register (SLEEPCFG.SLEEPMODE=STANDBY). The SLEEPONEXIT feature is also available as in IDLE mode.
- Exiting STANDBY mode: Any peripheral able to generate an asynchronous interrupt can wake up the system. For example, a peripheral running on a GCLK clock can trigger an interrupt. When the enabled asynchronous wake-up event occurs and the system is woken up, the device will either execute the interrupt service routine or continue the normal program execution according to the Priority Mask Register (PRIMASK) configuration of the CPU.

Refer to Table 20-3 Sleep Mode versus Power Domain State Overview on page 195 for the RAM state.

The regulator operates in low-power mode by default and switches automatically to the normal mode in case of a sleepwalking task requiring more power. It returns automatically to low power mode when the sleepwalking task is completed.

HIBERNATE and BACKUP Mode

The BACKUP mode allows achieving the lowest power consumption aside from OFF. The device is entirely powered off except for the backup domain. All peripherals in backup domain are allowed to run, e.g. the RTC can be clocked by a 32.768kHz oscillator. All PM registers are reset except the CTRLA.IORET bit.

- Entering Backup mode: This mode is entered by executing the WFI instruction after selecting the Backup mode by writing the Sleep Mode bits in the Sleep Configuration register (SLEEPCFG.SLEEPMODE=BACKUP).
- Exiting Backup mode: is triggered when a Backup Reset is detected by the Reset Controller (RSTC).

OFF Mode

In OFF mode, the device is entirely powered-off.

- Entering OFF mode: This mode is entered by selecting the OFF mode in the Sleep Configuration register by writing the Sleep Mode bits (SLEEPCFG.SLEEPMODE=OFF), and subsequent execution of the WFI instruction.
- Exiting OFF mode: This mode is left by pulling the $\overline{\text{RESET}}$ pin low, or when a power Reset is done.

20.6.3.4. I/O Lines Retention in BACKUP Mode

When entering BACKUP mode, the PORT is powered off but the pin configuration is retained. When the device exits the BACKUP mode, the I/O line configuration can either be released or stretched, based on the I/O Retention bit in the CTRLA register (CTRLA.IORET).

- If IORET=0 when exiting BACKUP mode, the I/O lines configuration is released and driven by the reset value of the PORT.
- If the IORET=1 when exiting BACKUP mode, the configuration of the I/O lines is retained until the IORET bit is written to 0. It allows the I/O lines to be retained until the application has programmed the PORT.

20.6.3.5. Performance Level

The application can change the performance level on the fly writing to the by Performance Level Select bit in the Performance Level Configuration register (PLCFG.PLSEL).

When changing to a lower performance level, the bus frequency must be reduced before writing PLCFG.PLSEL in order to avoid exceeding the limit of the target performance level.

When changing to a higher performance level, the bus frequency can be increased only after the Performance Level Ready flag in the Interrupt Flag Status and Clear (INTFLAG.PLRDY) it set to '1', indicating that the performance level transition is complete.

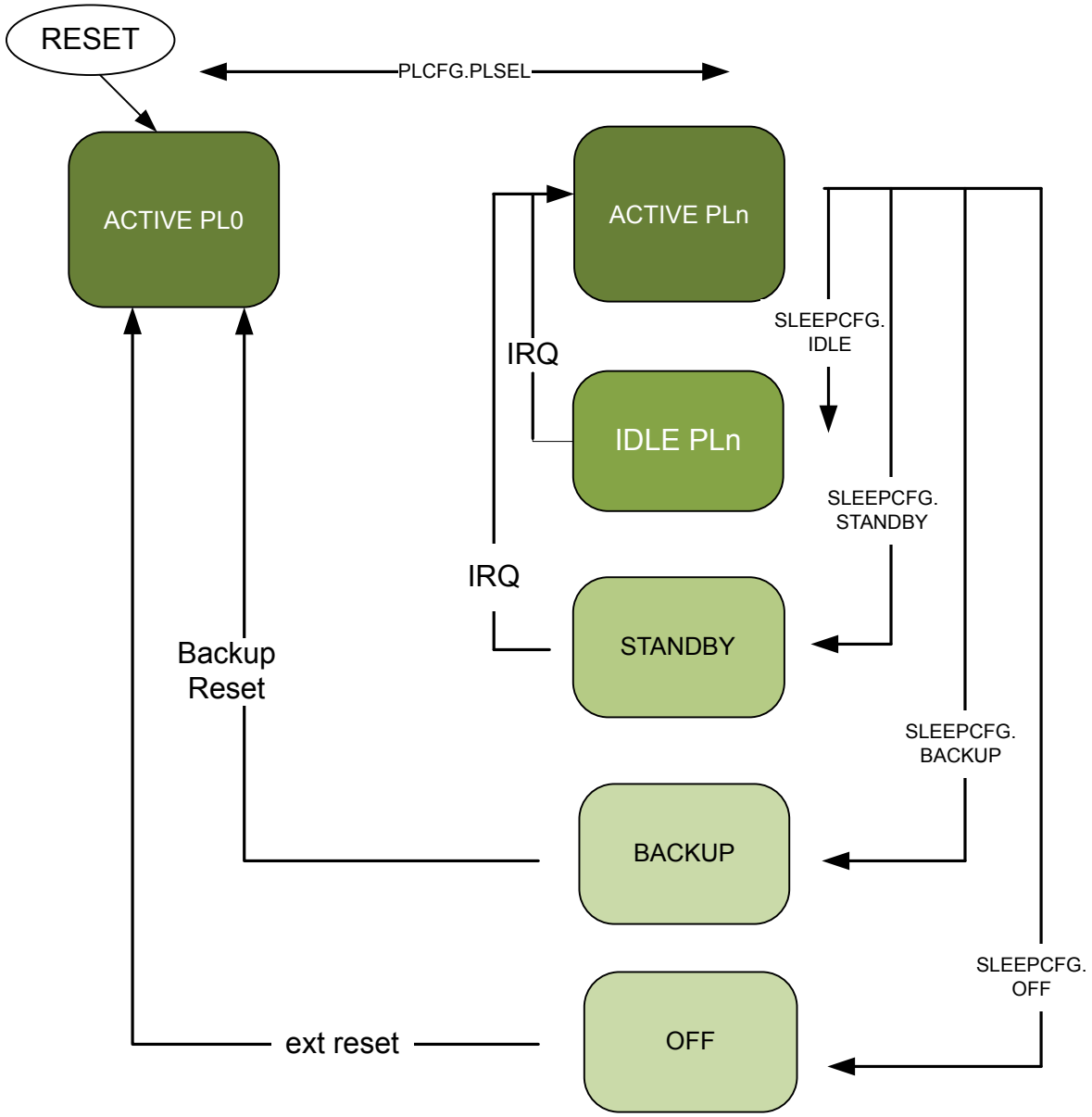
After a reset, the device starts in the lowest PL (lowest power consumption and lowest max frequency). The application can then switch to another PL at anytime without any stop in the code execution. As shown in Figure 20-3 Sleep Modes and Performance Level Transitions on page 194, performance level transition is possible only when the device is in active mode.

The Performance Level Disable bit in the Performance Level Configuration register (PLCFG.PLDIS) can be used to freeze the performance level to PL0. This disables the performance level hardware mechanism in order to reduce both the power consumption and the wake-up startup time from standby sleep mode.

Note: This bit PLCFG.PLDIS must be changed only when the current performance level is PL0.

Any attempt to modify this bit while the performance level is not PL0 is discarded and a violation is reported to the PAC module. Any attempt to change the performance level to PLn (with n>0) while PLCFG.PLDIS=1 is discarded and a violation is reported to the PAC module.

Figure 20-3. Sleep Modes and Performance Level Transitions



20.6.3.6. Power Domain Controller

The Power Domain Controller provides several ways of how power domains are handled while the device is in standby mode or entering standby mode:

- Default operation - all peripherals idle
  When entering standby mode, the power domains PD0, PD1, and PD2 are set in retention state. This allows for very low power consumption while retaining all the logic content of these power domains. When exiting standby mode, all power domains are set back to active state.
- Default operation - SleepWalking with static power gating (static SleepWalking)
  When a peripheral needs to remain running while the device is entering standby mode (e.g. to perform a sleepwalking task, or because of its RUNSTDBY bit written to '1') the power domain of

the peripheral (PDn) remains in active state as well as the inferior power domains (PDm with m<n). This is an extension of the SleepWalking applied to the power domain. At the end of the sleepwalking task, the device can either be woken up or remain in standby mode.

- SleepWalking with dynamic power gating (dynamic SleepWalking)
  A power domain PDn that is in active state due to static SleepWalking can wake up a superior power domain (PDm, with m<n) in order to perform a sleepwalking task. PDm is then automatically set to active state. At the end of the sleepwalking task, either the device can be woken up, or PDm can be set again to retention state.

The static and dynamic power gated SleepWalking features are fully transparent for the user. Which power domains are powered or not can also be configured manually, refer to Linked Power Domains on page 196 for details.

The table below illustrates these four cases to consider in standby mode:

1. SleepWalking is invoked on PD0, PD1, and PD2
2. SleepWalking is invoked on PD0 and PD1, while PD2 is in retention state
3. SleepWalking is invoked on PD0, while PD1 and PD2 are in retention state
4. This is the default mode where all PDs are in retention state

Table 20-3. Sleep Mode versus Power Domain State Overview

	Power Domain State				
Sleep Mode	**PD0**	**PD1**	**PD2**	**PDTOP**	**PDBACKUP**
Active	active	active	active	active	active
Idle	active	active	active	active	active
Standby - case 1	active	active	active	active	active
Standby - case 2	active	active	retention	active	active
Standby - case 3	active	retention	retention	active	active
Standby - case 4	retention	retention	retention	active	active
Backup	off	off	off	off	active
Off	off	off	off	off	off

20.6.3.7. Regulators, RAMs, and NVM State in Sleep Mode

By default, in standby sleep mode and backup sleep mode, the RAMs, NVM, and regulators are automatically set in low-power mode in order to reduce power consumption:

- The RAM is in low-power mode if its power domain is in retention or off state. Refer to RAM Automatic Low Power Mode on page 197 for details.
- Non-Volatile Memory - the NVM is located in the power domain PD2. By default, the NVM is automatically set in low power mode in these conditions:
  - When the power domain PD2 is in retention or off state.
  - When the device is in standby sleep mode and the NVM is not accessed. This behavior can be changed by software by configuring the SLEEPPRM bit group of the CTRLB register in the NVMCTRL peripheral.
  - When the device is in idle sleep mode and the NVM is not accessed. This behavior can be changed by software by configuring the SLEEPPRM bit group of the CTRLB register in the NVMCTRL peripheral.

- Regulators: by default, in standby sleep mode, the PM analyzes the device activity to use either the main or the low-power voltage regulator to supply the VDDCORE. Refer to Regulator Automatic Low Power Mode on page 197 for details.

GCLK clocks, regulators and RAM are not affected in idle sleep mode and operate in normal mode.

Table 20-4. Regulators, RAMs, and NVM state in Sleep Mode

Sleep Mode	Switchable Power Domains			RAMs mode[1]		NVM	Regulators		
	PD0	PD1	PD2	LP SRAM	SRAM		VDDCORE		VDDBU
							main	ulp	
Active	active	active	active	normal	normal	normal	on	on	on
Idle	active	active	active	normal	auto[2]	on	on	on	on
Standby - case 1	active	active	active	normal	normal	auto[2]	auto[3]	on	on
Standby - case 2	active	active	retention	normal	low power	low power	auto[3]	on	on
Standby - case 3	active	retention	retention	low power	low power	low power	auto[3]	on	on
Standby - case 4	retention	retention	retention	low power	low power	low power	off	on	on
Backup	off	off	off	off	off	off	off	off	on
OFF	off	off	off	off	off	off	off	off	off

Note:

1. RAMs mode by default: STDBYCFG.BBIAS bits are set to their default value.
2. auto: by default, NVM is in low-power mode if not accessed.
3. auto: by default, the main voltage regulator is on if GCLK, APBx, or AHBx clock is running during SleepWalking.
4. For a description of the cases, see Power Domain Controller on page 194.

20.6.4. Advanced Features

20.6.4.1. Power Domain Configuration

When entering standby sleep mode, a power domain is set automatically to retention state if no activity is required in it, refer to Power Domain Controller on page 194 for details. This behavior can be changed by writing the Power Domain Configuration bit group in the Standby Configuration register (STDBYCFG.PDCFG). For example, all power domains can be forced to remain in active state during standby sleep mode, this will accelerate wake-up time.

20.6.4.2. Linked Power Domains

Power domains can be linked to each other by using the Link Power Domain bit group in the Standby Configuration register (STDBYCFG.LINKPD). When PDn (n=0,1) is active, the linked power domain(s) of higher index PDm (m>n) will be in active state even if there is no activity in PDm.

When for example a static SleepWalking task is ongoing in PD0 while the device is in standby sleep mode and PD1 is linked to PD0 (LINKPD=PD01), then PD1 and PD0 are kept in active state. If dynamic SleepWalking is configured, the power state of PD1 will follow the state of PD0.

20.6.4.3. RAM Automatic Low Power Mode

The RAM is by default put in low power mode (back-biased) if its power domain is in retention state and the device is in standby sleep mode.

This behavior can be changed by configuring the Back Bias bit groups in the Standby Configuration register (STDBYCFG.BBIASxx), refer to the table below for details.

Note: in standby sleep mode, the DMAC can access the LP SRAM only when the power domain PD1 is not in retention and PM.STDBYCFG.BBIASLP=0x0. The DMAC can access the SRAM in standby sleep mode only when the power domain PD2 is not in retention and PM.STDBYCFG.BBIASHS=0x0.

Table 20-5. RAM Back-Biasing Mode

STBYCDFG.BBIASxx config		RAM
0x0	Retention Back Biasing mode	RAM is back-biased if its power domain is in retention state
0x1	Standby Back Biasing mode	RAM is back-biased if the device is in standby sleep mode
0x2	Standby OFF mode	RAM is OFF if the device is in standby sleep mode
0x3	Always OFF mode	RAM is OFF if its power domain is in retention state

20.6.4.4. Regulator Automatic Low Power Mode

In standby mode, the PM selects either the main or the low power voltage regulator to supply the VDDCORE. If all power domains are in retention state, the low power voltage regulator is used.

If a sleepwalking task is working on either asynchronous clocks (generic clocks) or synchronous clock (APB/AHB clocks), the main voltage regulator is used. This behavior can be changed by writing the Voltage Regulator Standby Mode bits in the Standby Configuration register (STDBYCFG.VREGSMOD). Refer to the following table for details.

Table 20-6. Regulator State in Sleep Mode

Sleep Modes	STDBYCFG. VREGSMOD	SleepWalking(1)	Regulator state for VDDCORE
Active	-	-	main voltage regulator
Idle	-	-	main voltage regulator
Standby (at least one PD is active)	0x0: AUTO	NO	low power regulator
		YES	main voltage regulator
	0x1: PERFORMANCE	-	main voltage regulator
	0x2: LP(2)	-(2)	low power regulator
Standby (all PD in retention)	-	-	low power regulator

Note:

1. SleepWalking is running on GCLK clock or synchronous clock. This is not related to OSC32K, XOSC32K or OSCULP32K clocks.
2. Must only be used when SleepWalking is running on GCLK with 32KHz source.

20.6.4.5. SleepWalking and Performance Level

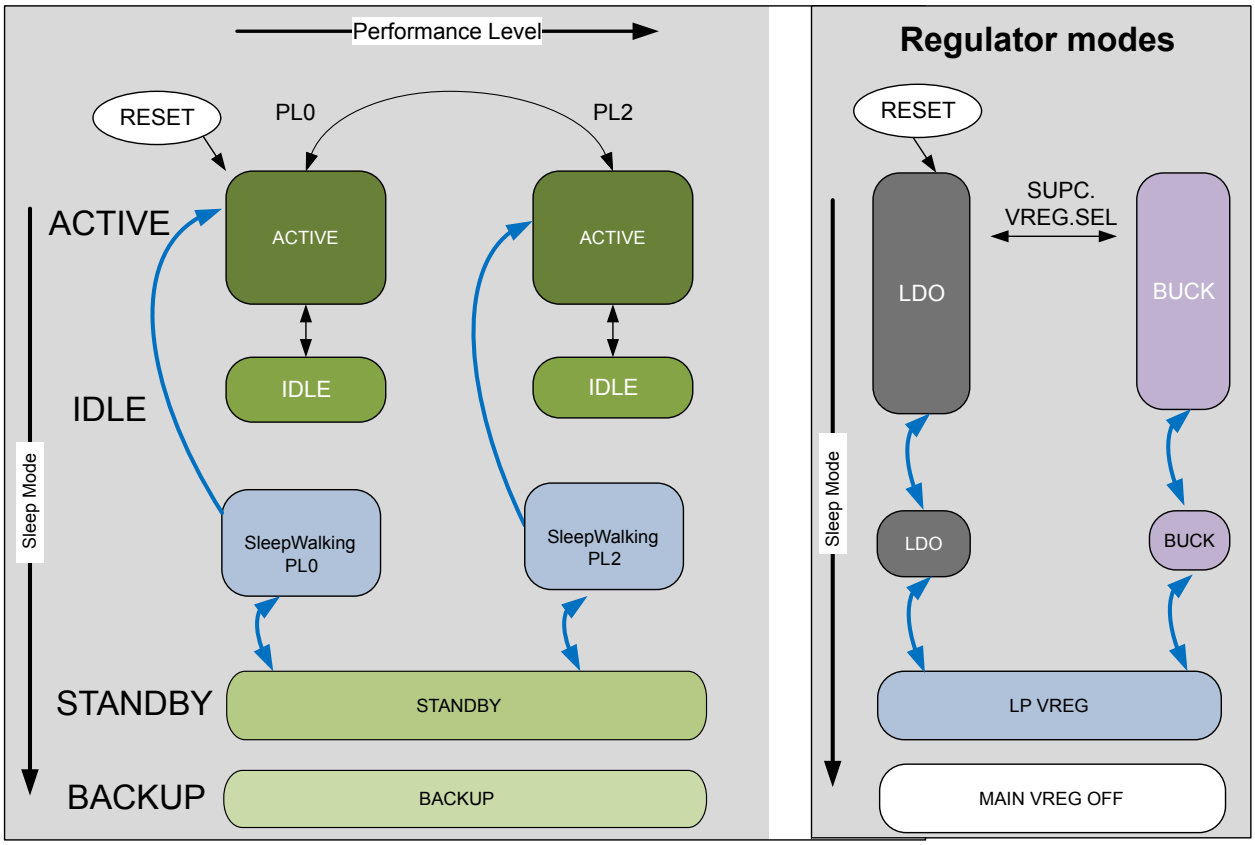
SleepWalking is the capability for a device to temporarily wake up clocks for a peripheral to perform a task without waking up the CPU from STANDBY sleep mode. At the end of the sleepwalking task, the device can either be woken p by an interrupt (from a peripheral involved in SleepWalking) or enter again

into STANDBY sleep mode. In this device, SleepWalking is supported only on GCLK clocks by using the on-demand clock principle of the clock sources.

In standby mode, when SleepWalking is ongoing, the performance level used to execute the sleepwalking task is the current configured performance level (used in active mode), and the main voltage regulator used to execute the sleepwalking task is the selected regulator used in active mode (LDO or Buck converter).

These are illustrated in the figure below.

Figure 20-4. Operating Conditions and SleepWalking



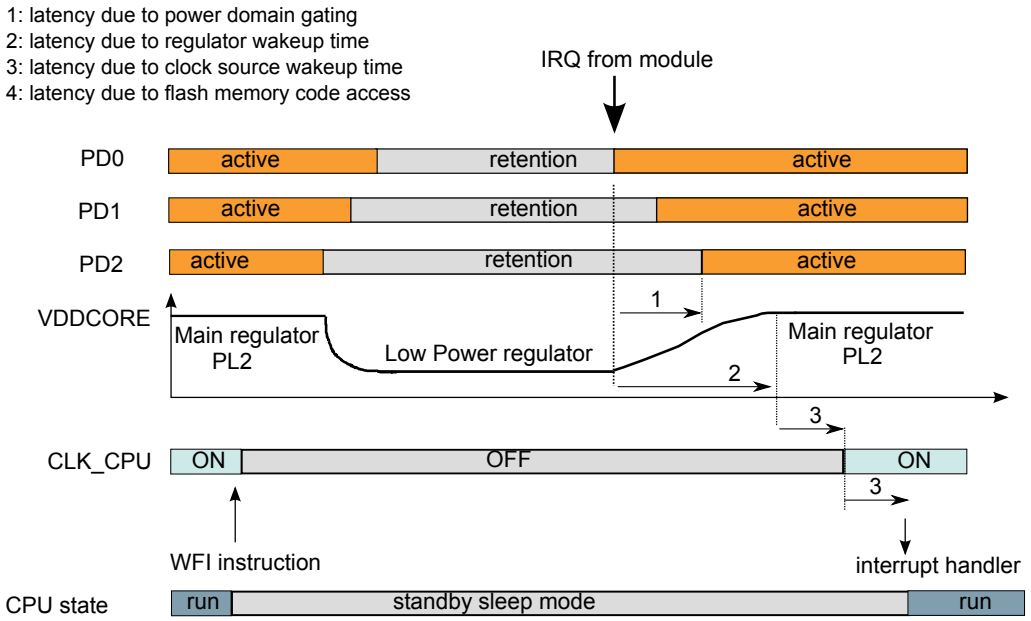
20.6.4.6. Wake-Up Time

As shown in the figure below, total wake-up time depends on:

- Latency due to Power Domain Gating:
  Usually, wake-up time is measured with the assumption that the power domains are already in active state. When using Power Domain Gating, changing a power domain from retention to active state will take a certain time, refer to Electrical Characteristics. If all power domains were already in active state in standby sleep mode, this latency is zero. If wake-up time is critical for the application, power domains can be forced to active state in standby sleep mode, refer to Power Domain Configuration on page 196 and Linked Power Domains on page 196 for details.
- Latency due to Performance Level and Regulator effect:
  Performance Level has to be taken into account for the global wake-up time. As example, if PL2 is selected and the device is in standby sleep mode, the voltage level supplied by the ULP voltage regulator is lower than the one used in active mode. When the device wakes up, it takes a certain amount of time for the main regulator to transition to the voltage level corresponding to PL2, causing additional wake-up time.

- Latency due to the CPU clock source wake-up time.
- Latency due to the NVM memory access.

Figure 20-5. Total Wake-up Time from Standby Sleep Mode



20.6.5. SleepWalking with Static Power Domain Gating in Details

In standby sleep mode, the power domain (PD) of a peripheral can remain in active state in order to perform sleepwalking tasks, whereas the other power domains are in retention state to reduce power consumption. This SleepWalking with static Power Domain Gating is supported by all peripherals. For some peripherals it must be enabled by writing a Run in Standby bit in the respective Control A register (CTRLA.RUNSTDBY) to '1'. Refer to each peripheral chapter for details.

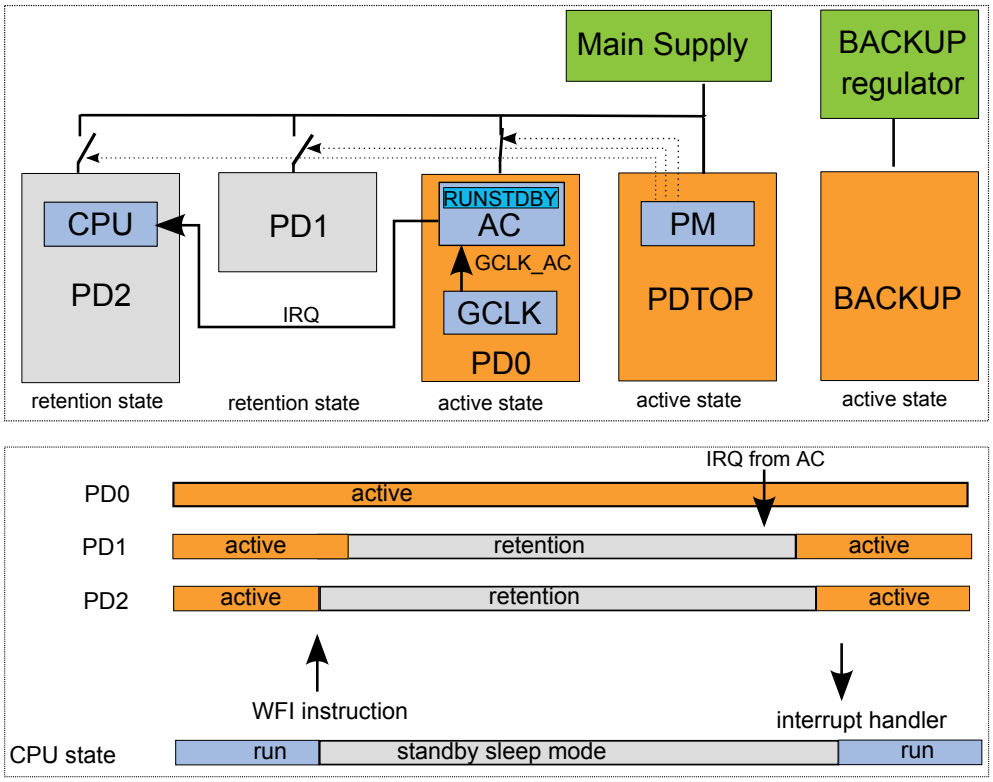
The following examples illustrate SleepWalking with static Power Domain Gating:

AC SleepWalking with Static PD Gating

The AC peripheral is used in continuous measurement mode to monitor voltage level on input pins. An AC interrupt is generated to wake up the device. To make the AC continue to run in standby sleep mode, the RUNSTDBY bit must be written to '1'.

- Entering standby mode: As shown in the next figure, PD0 (where the AC is located) remains active, whereas PD2 and PD1 are successively set to retention state by the Power Manager.

Figure 20-6. AC SleepWalking with Static PD Gating



- Exiting standby mode: When conditions are met, the AC peripheral generates an interrupt to wake up the device. Successively, the PM peripheral sets PD1 and PD2 to active state. Once PD2 is in active state, the CPU is able to operate normally and execute the AC interrupt handler accordingly.
- Wake-up time:
  - The required time to set PD1 and PD2 to active state has to be considered for the global wake-up time, refer to Wake-Up Time on page 198 for details.
  - In this case, the VDDCORE voltage is still supplied by the main voltage regulator, refer to Regulator Automatic Low Power Mode on page 197 for details. Thus, global wake-up time is not affected by the regulator.

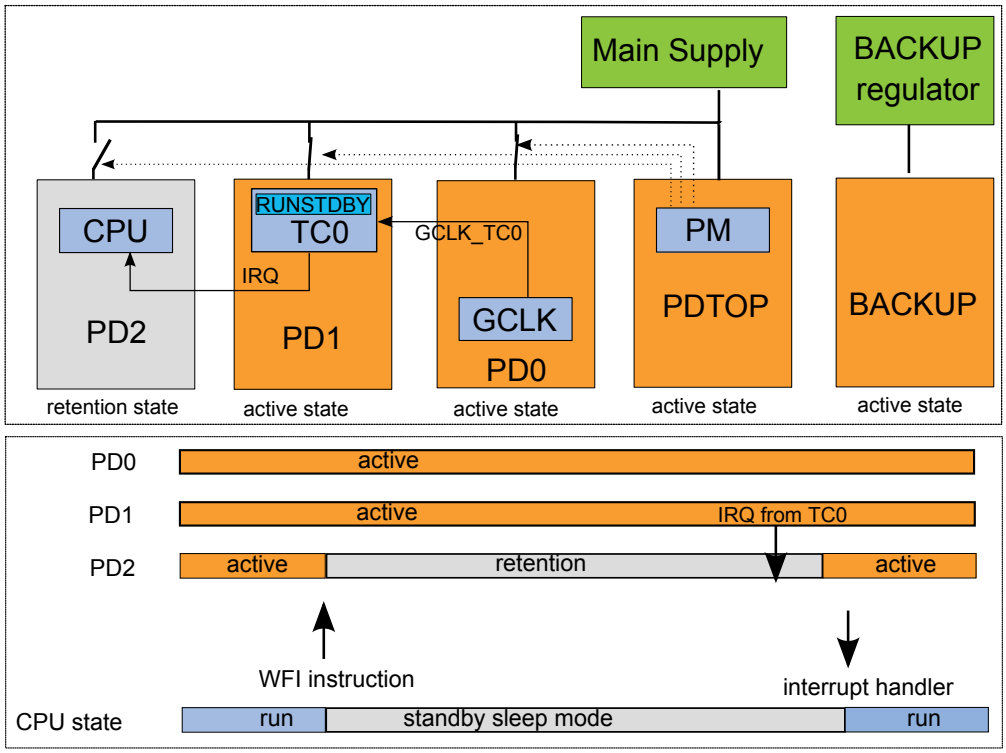
TC0 SleepWalking with Static PD Gating

TC0 peripheral is used in counter operation mode. An interrupt is generated to wake-up the device based on the TC0 peripheral configuration. To make the TC0 peripheral continue to run in standby sleep mode, the RUNSTDBY bit is written to '1'.

- Entering standby mode: As shown in Figure 20-7 TC0 SleepWalking with Static PD Gating on page 201, PD1 (where the TC0 is located) and PD0 (where the peripheral clock generator is located) remain active, whereas PD2 is set to retention state by the Power Manager peripheral. Refer to Power Domain Controller on page 194 for details.
- Exiting standby mode: When conditions are met, the TC0 peripheral generates an interrupt to wake-up the device. The PM peripheral sets PD2 to active state. Once PD2 is in active state, the CPU is able to operate normally and execute the TC0 interrupt handler accordingly.
- Wake-up time:
  - The required time to set PD2 to active state has to be considered for the global wake-up time, refer to Wake-Up Time on page 198 for details.

– In this case, the VDDCORE voltage is still supplied by the main voltage regulator, refer to Regulator Automatic Low Power Mode on page 197 for details. Thus, global wake-up time is not affected by the regulator.

Figure 20-7. TC0 SleepWalking with Static PD Gating

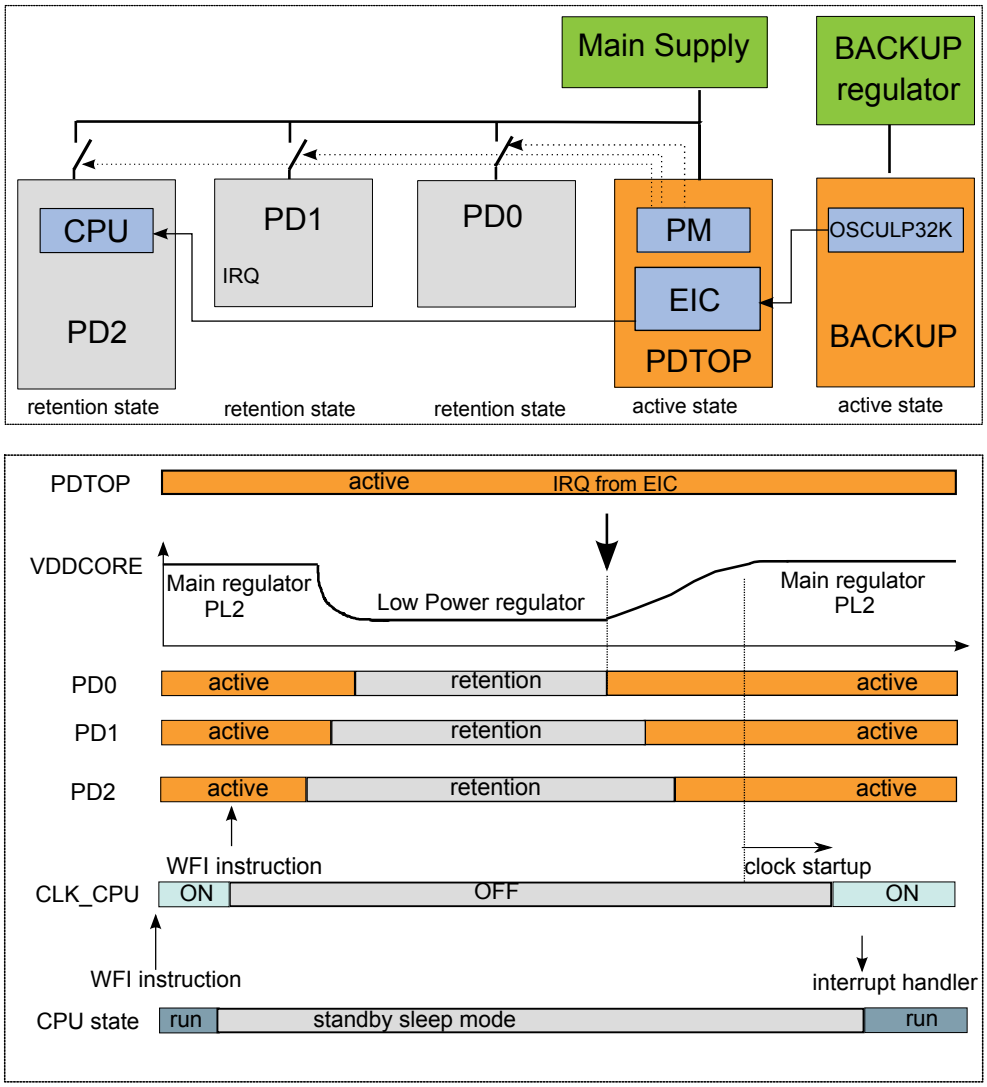


EIC SleepWalking with Static PD Gating

In this example, EIC peripheral is used to detect an edge condition to generate interrupt to the CPU. An External interrupt pin is filtered by the CLK_ULP32K clock, GCLK peripheral is not used. Refer to Chapter EIC – External Interrupt Controller on page 464 for details. As the EIC peripheral is located in the power domain PDTOP (which is not switchable), there is no RUNSTDBY bit in the EIC peripheral.

- Entering standby mode: As shown in Figure 20-8 EIC SleepWalking with Static PD Gating on page 202, all the switchable power domains are set in retention state by the Power Manager peripheral. The low power regulator supplies the VDDCORE voltage level.
- Exiting standby mode: When conditions are met, the EIC peripheral generates an interrupt to wake the device up. Successively, the PM peripheral sets PD0, PD1, and PD2 to active state, and the main voltage regulator restarts. Once PD2 is in active state and the main voltage regulator is ready, the CPU is able to operate normally and execute the EIC interrupt handler accordingly.
- Wake-up time:
  - The required time to set the switchable power domains to active state has to be considered for the global wake-up time, refer to Wake-Up Time on page 198 for details.
  - When in standby sleep mode, the GCLK peripheral is not used, allowing the VDDCORE to be supplied by the low power regulator to reduce consumption, see Regulator Automatic Low Power Mode on page 197. Consequently, main voltage regulator wake-up time has to be considered for the global wake-up time as shown in Figure 20-8 EIC SleepWalking with Static PD Gating on page 202.

Figure 20-8. EIC SleepWalking with Static PD Gating



20.6.6. Sleepwalking with Dynamic Power Domain Gating in Details

To reduce power consumption even further, Sleepwalking with dynamic Power Domain Gating (also referred to as "Dynamic Sleepwalking") is used to turn power domain state from retention to active and vice-versa, based on event or AHB bus transaction.

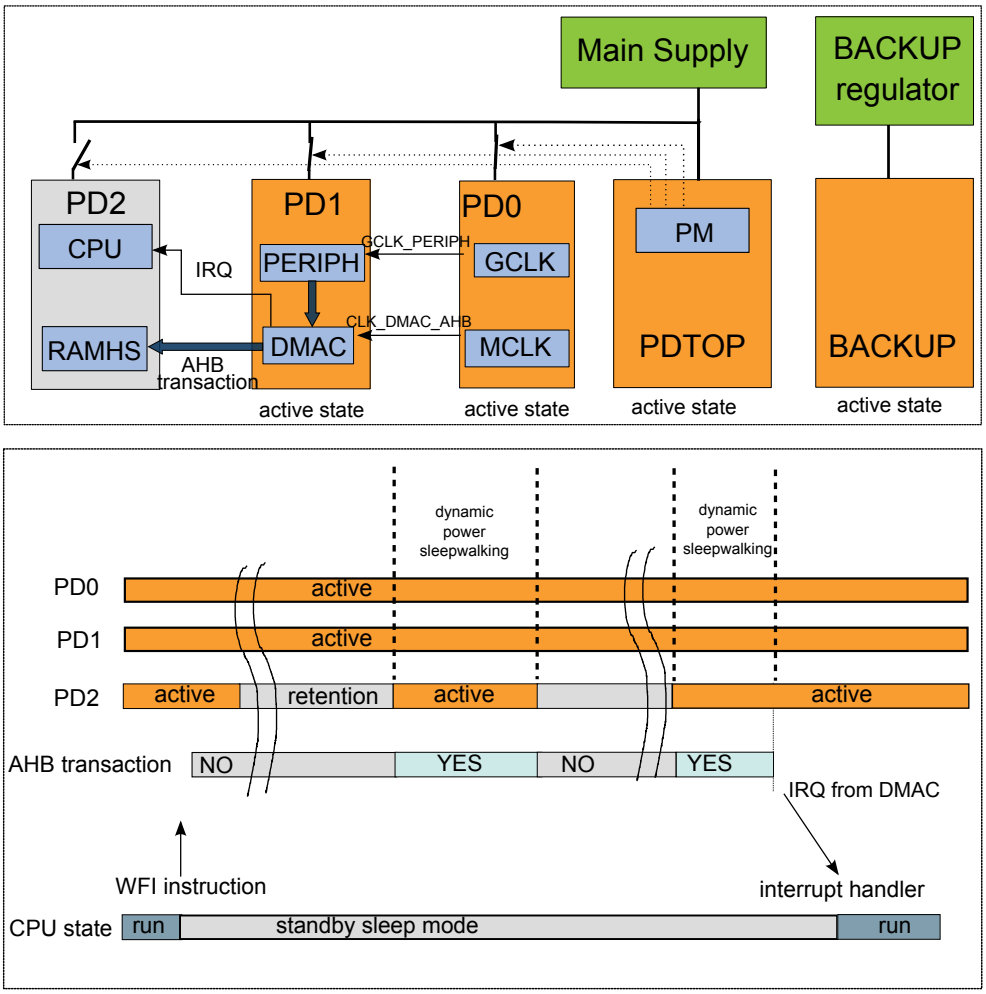
20.6.6.1. Dynamic SleepWalking on Bus Transaction

When in retention state, a power domain can be automatically set to active state by the PM if AHB bus transaction in direction to this power domain is detected. In this device, it concerns the AHB bus transaction from the DMAC to the modules located in power domain PD2.

Dynamic SleepWalking based on bus transaction is illustrated in the example below. By using the Run in Standby bit, the DMAC is configured to operate in standby sleep mode. A DMAC channel is configured to make peripheral-to-memory transfer from a module located in PD1 to the SRAM. Transfer request is triggered by the peripheral at periodic time. Refer to DMAC – Direct Memory Access Controller on page 399 for details. PD2 is set to active state only when AHB transaction is required before being set to retention state again to save power. Note that during this dynamic Sleepwalking period, the CPU is still

sleeping. The device can be woken up by an interrupt, for example at the end of a complete DMA block transfer.

Figure 20-9. Dynamic SleepWalking Based on Bus Transaction



20.6.6.2. Dynamic SleepWalking based on Event

To enable SleepWalking with dynamic power domain gating, the Dynamic Power Gating for Power Domain 0 and 1 bits in the Standby Configuration register (STDBYCFG.DPGPD0 and STDBYCFG.DPGPD1) have to be written to '1'.

When in retention state, a power domain can be automatically set to active state by the PM if an event is directed to this power domain. In this device, this concerns the event users located in power domains PD1 and PD0.

- When PD0, PD1 and PD2 are in retention state, dynamic SleepWalking can be triggered by:
  - RTC output event
  - EIC output event (if using the CLK_ULP32K clock)
- When PD0 is active whereas PD1 and PD2 are in retention state, dynamic SleepWalking can be triggered by:
  - RTC output event
  - EIC output event (if using CLK_ULP32K)
  - all peripheral within PD0 that are capable of generating events

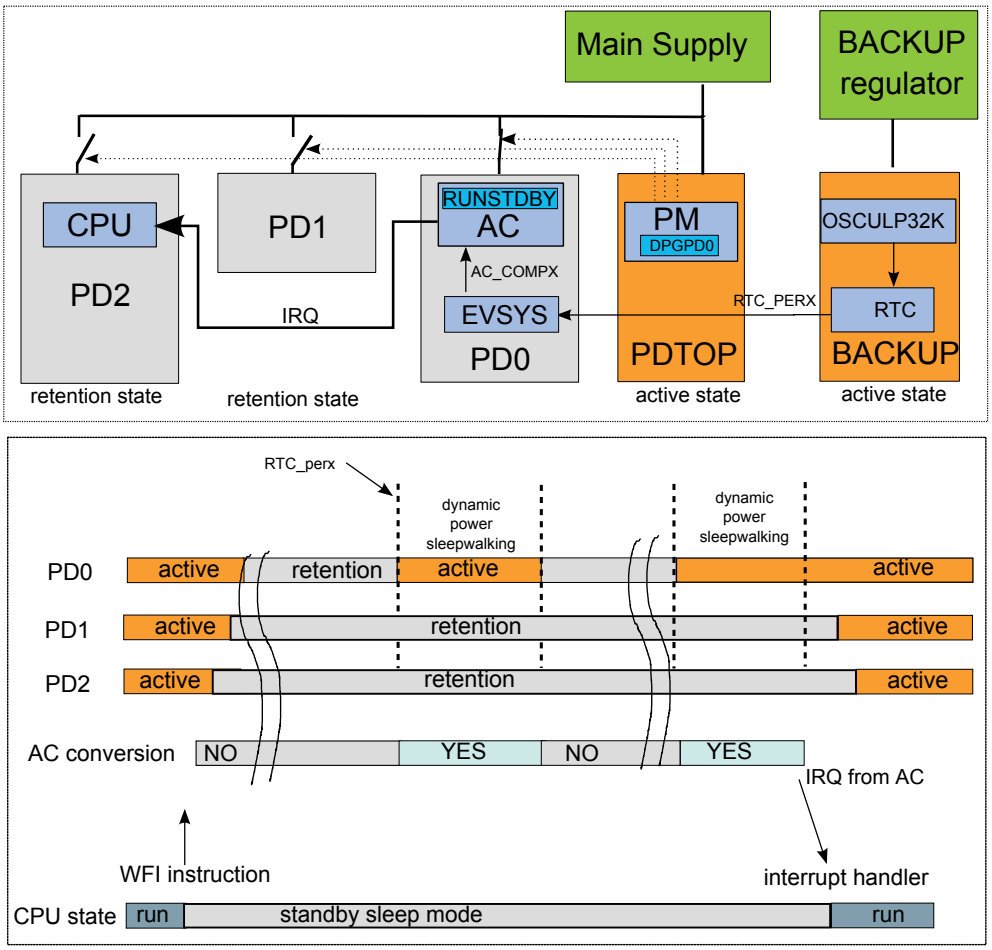
All modules located in PD0 are able to generate events. The EVSYS event generator must be configured to either synchronous or resynchronized path.

- When PD0 and PD1 are in retention, dynamic SleepWalking based on event is not useful.

Refer also to Power Domains on page 188.

Dynamic SleepWalking based on event is illustrated in the following example:

Figure 20-10. Dynamic SleepWalking based on Event: AC Periodic Comparison



The Analog Comparator (AC) peripheral is used in single shot mode to monitor voltage levels on input pins. A comparator interrupt, based on the AC peripheral configuration, is generated to wake up the device. In the GCLK module, the AC generic clock (GCLK_AC) source is routed a 32.768kHz oscillator (for low power applications, OSC32KULP is recommended). RTC and EVSYS modules are configured to generate periodic events to the AC. To make the comparator continue to run in standby sleep mode, the RUNSTDBY bit is written to '1'. To enable the dynamic SleepWalking for PD0 power domain, STDBYCFG.DPGPD0 must be written to '1'.

Entering standby mode: The Power Manager sets the PD0 power domain (where the AC module is located) in retention state, as well as PD1 and PD2. The AC comparators, COMPx, are OFF. The GCLK_AC clock is stopped. The VDDCORE is supplied by the low power regulator.

Dynamic SleepWalking: The RTC event (RTC_PERX) is routed by the Event System to the Analog Comparator to trigger a single-shot measurement. This event is detected by the Power Manager, which sets the PD0 power domain to active state and starts the main voltage regulator.

After enabling the AC comparator and starting the GCLK_AC, the single-shot measurement can be performed during sleep mode (sleepwalking task), refer to Single-Shot Measurement during Sleep on page 1083 for details. At the end of the conversion, if conditions to generate an interrupt are not met, the GCLK_AC clock is stopped again, as well as the AC comparator.

The low power regulator starts again and the PD0 power domain is set back to retention state by the PM. Note that during this dynamic SleepWalking period, the CPU is still sleeping.

Exiting standby mode: during the dynamic SleepWalking sequence, if conditions are met, the AC module generates an interrupt to wake up the device. Successively, the PD1 and PD2 power domain are set to active state by the PM.

Related Links

RTC – Real-Time Counter on page 341

EVSYS – Event System on page 536

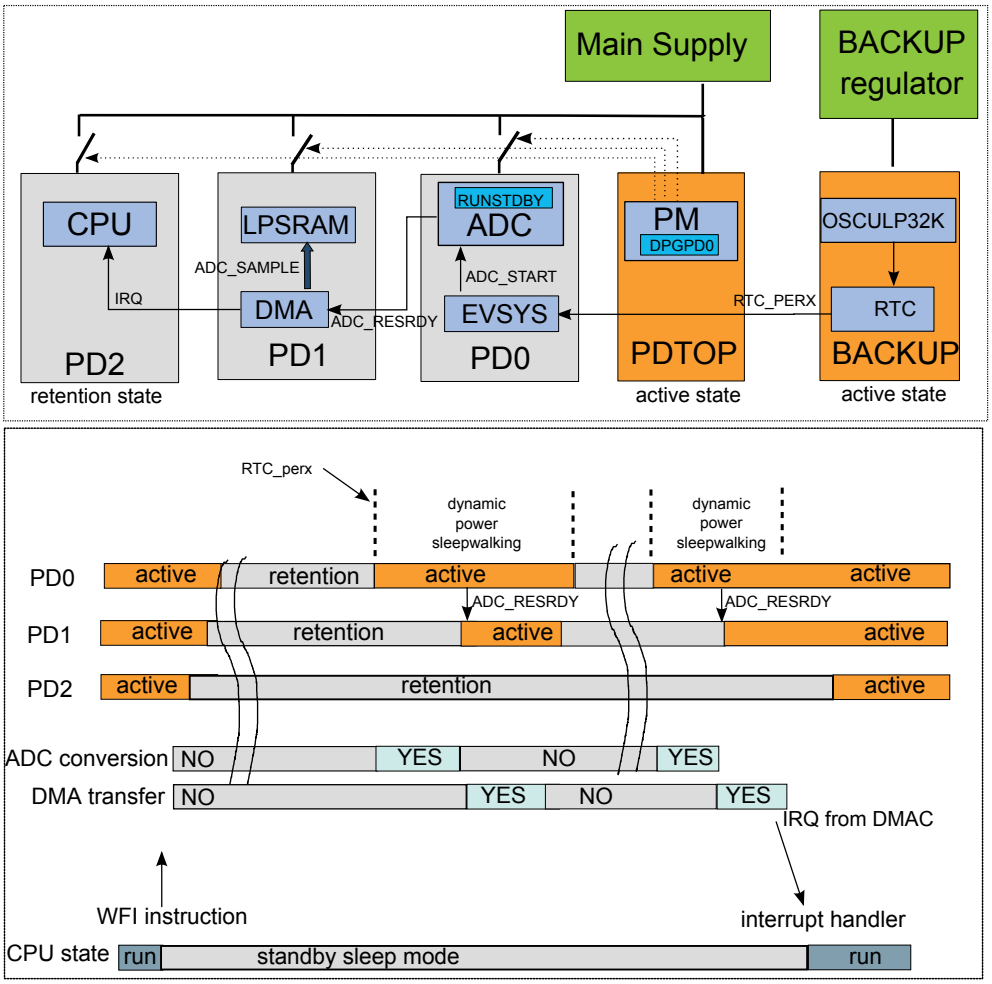
20.6.6.3. Dynamic SleepWalking Based on Peripheral DMA Trigger

To enable this advanced feature, the Dynamic Power Gating for Power Domain 0 and 1 bits in the Standby Configuration register (STDBYCFG.DPGPD0 and STDBYCFG.DPGPD1) have to be written to '1'.

When in retention state, the power domain PD1 (containing the DMAC) can be automatically set to active state if the PM detects a valid DMA trigger that is coming from a peripheral located in PD0. A peripheral DMA trigger is valid if the corresponding DMA channel is enabled and its Run in Standby bit (RUNSTDBY) is written to '1'.

This is illustrated in the following example:

Figure 20-11. Dynamic Sleepwalking based on Peripheral DMA Trigger



The Analog to Digital Converter (ADC) peripheral is used in one shot measurement mode to periodically convert a voltage level on input pins, and move the conversion result to RAM by DMA. After N conversions, an interrupt is generated by the DMA to wake up the device. In the GCLK module, the ADC generic clock (GCLK_ADC) source is routed to OSCULP32K. RTC and EVSYS modules are configured to generate periodic events to the ADC.

To make the ADC continue to run in standby sleep mode, its Run in Standby (RUNSTDBY) bit is written to '1'. The DMAC is configured to operate in standby sleep mode as well by using its respective RUNSTDBY bit. A DMAC channel is configured to enable peripheral-to-memory transfer from the ADC to the LP SRAM and to generate an interrupt when the block transfer is completed (after N beat transfers). The Run in Standby bit of this DMAC channel is written to '1' to allow it running in standby sleep mode.

Entering Standby mode: The Power Manager peripheral sets PD0 (where the ADC peripheral is located), PD1 (the DMAC is located here) and PD2 (CPU) to retention state. The ADC channels are OFF. The GCLK_ADC clock is stopped. The VDDCORE is supplied by the low power regulator.

Dynamic SleepWalking: based on RTC conditions, a RTC event (RTC_PERX) is routed by the Event System to the ADC controller to trigger a single-shot measurement.

This event is detected by the Power Manager which sets the PD0 power domain to active state and starts the main voltage regulator.

After enabling the ADC and starting the GCLK_ADC clock, the single-shot measurement during sleep mode can be performed as a sleepwalking task, refer to the ADC documentation for details. At the end of the comparison, a DMA transfer request (ADC_RESRDY) is triggered by the ADC.

This DMA transfer request is detected by the PM, which sets PD1 (containing the DMAC) to active state. The DMAC requests the CLK_DMAC_AHB clock and transfers the sample to the memory. When the DMA beat transfer is completed, the GCLK_ADC clock and the CLK_DMAC_AHB clock are stopped again, as well as the ADC peripheral.

The low power regulator starts again and the PD0 power domain is set back to retention state by the PM. Note that during this dynamic SleepWalking period, the CPU is still sleeping.

Exiting Standby mode: during SleepWalking with Dynamic Power Gating sequence, if conditions are met, the ADC peripheral generates an interrupt to wake up the device. Successively, the PD1 and PD2 power domain are set to active state by the PM.

Note: If the event trigger coming from PD0 is waking a peripheral in PD1 that does support SleepWalking, the PD1 will stay active until the follow-up task is finished. If the peripheral in PD1 does not support SleepWalking, the peripheral and PD1 will stay active after the task is finished.

Related Links

RTC – Real-Time Counter on page 341
EVSYS – Event System on page 536

20.6.7. DMA Operation

Not applicable.

20.6.8. Interrupts

The peripheral has the following interrupt sources:

- Performance Level Ready (PLRDY)
  This interrupt is a synchronous wake-up source. See Table 20-1 Sleep Mode Entry and Exit Table on page 191 for details.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register.

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled or the peripheral is reset.

An interrupt flag is cleared by writing a '1' to the corresponding bit in the INTFLAG register. Each peripheral can have one interrupt request line per interrupt source or one common interrupt request line for all the interrupt sources. Refer to Nested Vector Interrupt Controller on page 51 for details. If the peripheral has one common interrupt request line for all the interrupt sources, the user must read the INTFLAG register to determine which interrupt condition is present.

20.6.9. Events

Not applicable.

20.6.10. Sleep Mode Operation

The Power Manager is always active.

20.7. Register Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0						IORET		
0x01	SLEEPCFG	7:0						SLEEPMODE[2:0]		
0x02	PLCFG	7:0	PLDIS						PLSEL[1:0]	
0x03	Reserved									
0x04	INTENCLR	7:0								PLRDY
0x05	INTENSET	7:0								PLRDY
0x06	INTFLAG	7:0								PLRDY
0x07	Reserved									
0x08	STDBYCFG	7:0	VREGSMOD[1:0]		DPGPD1	DPGPD0			PDCFG[1:0]	
0x09		15:8			BBIASLP[1:0]		BBIASHS[1:0]		LINKPD[1:0]	

20.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection on page 187.

20.8.1. Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
						IORET		
Access						R/W		
Reset						0		

Bit 2 – IORET: I/O Retention
Note: This bit is not reset by a backup reset.

Value	Description
0	After waking up from Backup mode, I/O lines are not held.
1	After waking up from Backup mode, I/O lines are held until IORET is written to 0.

20.8.2. Sleep Configuration

Name: SLEEPCFG
Offset: 0x01
Reset: 0x2
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
						SLEEPMODE[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 2:0 – SLEEPMODE[2:0]: Sleep Mode
Note: A small latency happens between the store instruction and actual writing of the SLEEPCFG register due to bridges. Software has to make sure the SLEEPCFG register reads the wanted value before issuing WFI instruction.

Value	Name	Definition
0x0	Reserved	Reserved
0x1	Reserved	Reserved
0x2	IDLE	CPU, AHBx, and APBx clocks are OFF
0x3	Reserved	Reserved
0x4	STANDBY	ALL clocks are OFF, unless requested by sleepwalking peripheral
0x5	BACKUP	Only Backup domain is powered ON
0x6	OFF	All power domains are powered OFF
0x7	Reserved	Reserved

20.8.3. Performance Level Configuration

Name: PLCFG
Offset: 0x02
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	PLDIS						PLSEL[1:0]	
Access	R/W						R/W	R/W
Reset	0						0	0

Bit 7 – PLDIS: Performance Level Disable
Disabling the automatic PL selection forces the device to run in PL0, reducing the power consumption and the wake-up time from standby sleep mode.

Changing this bit whereas the current performance level is not PL0 is discarded an a violation is reported to the PAC module.

Value	Description
0	The Performance Level mechanism is enabled.
1	The Performance Level mechanism is disabled.

Bits 1:0 – PLSEL[1:0]: Performance Level Select

Value	Name	Definition
0x0	PL0	Performance Level 0
0x1	Reserved	Reserved
0x2	PL2	Performance Level 2
0x3	Reserved	Reserved

20.8.4. Interrupt Enable Clear

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Name: INTENCLR
Offset: 0x04
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								PLRDY
Access								R/W
Reset								0

Bit 0 – PLRDY: Performance Level Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Performance Ready Interrupt Enable bit and the corresponding interrupt request.

Value	Description
0	The Performance Ready interrupt is disabled.
1	The Performance Ready interrupt is enabled and will generate an interrupt request when the Performance Ready Interrupt Flag is set.

20.8.5. Interrupt Enable Set

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Name: INTENSET
Offset: 0x05
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								PLRDY
Access								R/W
Reset								0

Bit 0 – PLRDY: Performance Level Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Performance Ready Interrupt Enable bit and enable the Performance Ready interrupt.

Value	Description
0	The Performance Ready interrupt is disabled.
1	The Performance Ready interrupt is enabled.

20.8.6. Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x06
Reset: 0x00
Property: –

Bit	7	6	5	4	3	2	1	0
								PLRDY
Access								R/W
Reset								0

Bit 0 – PLRDY: Performance Level Ready
This flag is set when the performance level is ready and will generate an interrupt if INTENCLR/SET.PLRDY is '1'.

Writing a '1' to this bit has no effect.

Writing a '1' to this bit clears the Performance Ready interrupt flag.

20.8.7. Standby Configuration

Name: STDBYCFG
Offset: 0x08
Reset: 0x0000
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
			BBIASLP[1:0]		BBIASHS[1:0]		LINKPD[1:0]	
Access			R/W	R/W	R	R	R/W	R/W
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	VREGSMOD[1:0]		DPGPD1	DPGPD0			PDCFG[1:0]	
Access	R	R	R/W	R/W			R/W	R/W
Reset	0	0	0	0			0	0

Bits 13:12 – BBIASLP[1:0]: Back Bias for HMCRAMCLP
Refer to Table 20-5 RAM Back-Biasing Mode on page 197 for details.

Value	Description
0	Retention Back Biasing mode
1	Standby Back Biasing mode
2	Standby OFF mode
3	Always OFF mode

Bits 11:10 – BBIASHS[1:0]: Back Bias for HMCRAMCHS
Refer to Table 20-5 RAM Back-Biasing Mode on page 197 for details.

Value	Description
0	Retention Back Biasing mode
1	Standby Back Biasing mode
2	Standby OFF mode
3	Always OFF mode

Bits 9:8 – LINKPD[1:0]: Linked Power Domain
Refer to Linked Power Domains on page 196 for details.

Value	Name	Description
0x0	DEFAULT	Power domains PD0/PD1/PD2 are not linked.
0x1	PD01	Power domains PD0 and PD1 are linked. If PD0 is active, then PD1 is active even if there is no activity in PD1 .

Value	Name	Description
0x2	PD12	Power domains PD1 and PD2 are linked. If PD1 is active, then PD2 is active even if there is no activity in PD2.
0x3	PD012	All Power domains are linked. If PD0 is active, then PD1 and PD2 are active even if there is no activity in PD1 or PD2.

Bits 7:6 – VREGSMOD[1:0]: VREG Switching Mode
Refer to Regulator Automatic Low Power Mode on page 197 for details.

Value	Name	Description
0x0	AUTO	Automatic Mode
0x1	PERFORMANCE	Performance oriented
0x2	LP	Low Power consumption oriented

Bit 5 – DPGPD1: Dynamic Power Gating for Power Domain 1

Value	Description
0	Dynamic SleepWalking for power domain 1 is disabled.
1	Dynamic SleepWalking for power domain 1 is enabled.

Bit 4 – DPGPD0: Dynamic Power Gating for Power Domain 0

Value	Description
0	Dynamic SleepWalking for power domain 0 is disabled.
1	Dynamic SleepWalking for power domain 0 is enabled.

Bits 1:0 – PDCFG[1:0]: Power Domain Configuration

Value	Name	Description
0x0	DEFAULT	In standby mode, all power domain switching are handled by hardware.
0x1	PD0	In standby mode, power domain 0 (PD0) is forced ACTIVE. Other power domain switching is handled by hardware.
0x2	PD01	In standby mode, power domains PD0 and PD1 are forced ACTIVE. Power domain 2 switching is handled by hardware.
0x3	PD012	In standby mode, all power domains are forced ACTIVE.

21. OSCCTRL – Oscillators Controller

21.1. Overview

The Oscillators Controller (OSCCTRL) provides a user interface to the XOSC, OSC16M, DFLL48M, and FDPLL96M.

Through the interface registers, it is possible to enable, disable, calibrate, and monitor the OSCCTRL sub-peripherals.

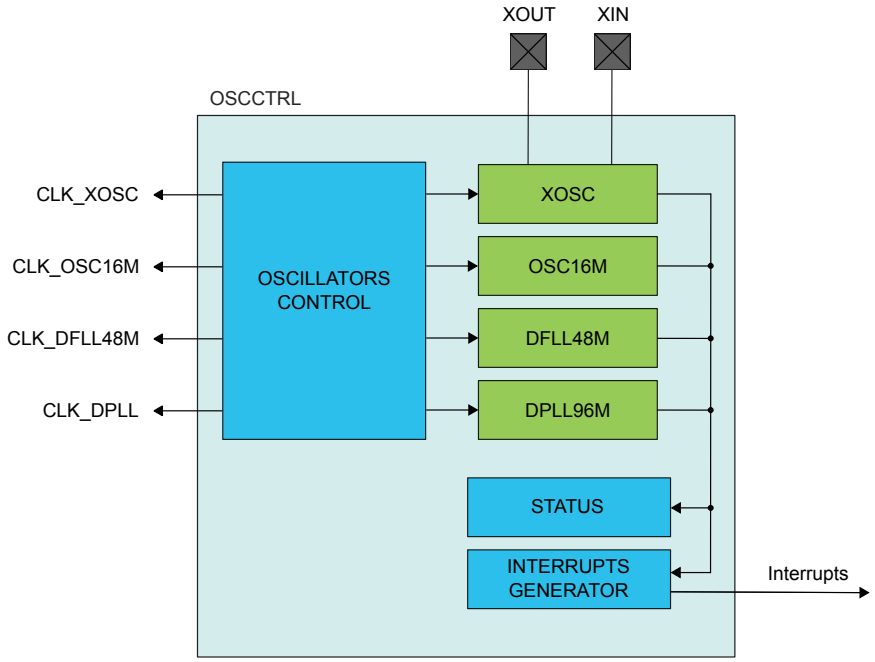
All sub-peripheral statuses are collected in the Status register (STATUS). They can additionally trigger interrupts upon status changes via the INTENSET, INTENCLR, and INTFLAG registers.

21.2. Features

- 0.4-32MHz Crystal Oscillator (XOSC)
  - Tunable gain control
  - Programmable start-up time
  - Crystal or external input clock on XIN I/O
- 16MHz Internal Oscillator (OSC16M)
  - Fast startup
  - 4/8/12/16MHz output frequencies available
- Digital Frequency Locked Loop (DFLL48M)
  - Internal oscillator with no external components
  - 48MHz output frequency
  - Operates stand-alone as a high-frequency programmable oscillator in open loop mode
  - Operates as an accurate frequency multiplier against a known frequency in closed loop mode
- Fractional Digital Phase Locked Loop (FDPLL96M)
  - 48MHz to 96MHz output frequency
  - 32kHz to 2MHz reference clock
  - A selection of sources for the reference clock
  - Adjustable proportional integral controller
  - Fractional part used to achieve 1/16th of reference clock step

21.3. Block Diagram

Figure 21-1. OSCCTRL Block Diagram



21.4. Signal Description

Signal	Description	Type
XIN	Multipurpose Crystal Oscillator or external clock generator input	Analog input
XOUT	Multipurpose Crystal Oscillator output	Analog output

The I/O lines are automatically selected when XOSC is enabled.

21.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

21.5.1. I/O Lines

I/O lines are configured by OSCCTRL when XOSC is enabled, and need no user configuration.

21.5.2. Power Management

The OSCCTRL can continue to operate in any sleep mode where the selected source clock is running. The OSCCTRL interrupts can be used to wake up the device from sleep modes. The events can trigger other operations in the system without exiting sleep modes.

Related Links

PM – Power Manager on page 186

21.5.3. Clocks

The OSCCTRL gathers controls for all device oscillators and provides clock sources to the Generic Clock Controller (GCLK). The available clock sources are: XOSC, OSC16M, DFLL48M, and FDPLL96M.

The OSCCTRL bus clock (CLK_OSCCTRL_APB) can be enabled and disabled in the Main Clock module (MCLK).

The DFLL48M control logic uses the DFLL oscillator output, which is also asynchronous to the user interface clock (CLK_OSCCTRL_APB). Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains. Refer to Synchronization on page 229 for further details.

Related Links

MCLK – Main Clock on page 148

Peripheral Clock Masking on page 151

21.5.4. DMA

Not applicable.

21.5.5. Interrupts

The interrupt request line is connected to the Interrupt Controller. Using the OSCCTRL interrupts requires the interrupt controller to be configured first.

Related Links

Nested Vector Interrupt Controller on page 51

21.5.6. Debug Operation

When the CPU is halted in debug mode the OSCCTRL continues normal operation. If the OSCCTRL is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

21.5.7. Register Access Protection

All registers with write-access can be write-protected optionally by the Peripheral Access Controller (PAC), except the following registers:

- Interrupt Flag Status and Clear register (INTFLAG)

Note: Optional write-protection is indicated by the "PAC Write-Protection" property in the register description.

When the CPU is halted in debug mode, all write-protection is automatically disabled. Write-protection does not apply for accesses through an external debugger.

21.5.8. Analog Connections

The 0.4-32MHz crystal must be connected between the XIN and XOUT pins, along with any required load capacitors.

21.6. Functional Description

21.6.1. Principle of Operation

XOSC, OSC16M, DFLL48M, and FDPLL96M are configured via OSCCTRL control registers. Through this interface, the sub-peripherals are enabled, disabled, or have their calibration values updated.

The Status register gathers different status signals coming from the sub-peripherals controlled by the OSCCTRL. The status signals can be used to generate system interrupts, and in some cases wake up the system from standby mode, provided the corresponding interrupt is enabled.

21.6.2. External Multipurpose Crystal Oscillator (XOSC) Operation

The XOSC can operate in two different modes:

- External clock, with an external clock signal connected to the XIN pin
- Crystal oscillator, with an external 0.4-32MHz crystal

The XOSC can be used as a clock source for generic clock generators. This is configured by the Generic Clock Controller.

At reset, the XOSC is disabled, and the XIN/XOUT pins can be used as General Purpose I/O (GPIO) pins or by other peripherals in the system. When XOSC is enabled, the operating mode determines the GPIO usage. When in crystal oscillator mode, the XIN and XOUT pins are controlled by the OSCCTRL, and GPIO functions are overridden on both pins. When in external clock mode, only the XIN pin will be overridden and controlled by the OSCCTRL, while the XOUT pin can still be used as a GPIO pin.

The XOSC is enabled by writing a '1' to the Enable bit in the External Multipurpose Crystal Oscillator Control register (XOSCCTRL.ENABLE).

To enable XOSC as an external crystal oscillator, the XTAL Enable bit (XOSCCTRL.XTALEN) must written to '1'. If XOSCCTRL.XTALEN is zero, the external clock input on XIN will be enabled.

When in crystal oscillator mode (XOSCCTRL.XTALEN=1), the External Multipurpose Crystal Oscillator Gain (XOSCCTRL.GAIN) must be set to match the external crystal oscillator frequency. If the External Multipurpose Crystal Oscillator Automatic Amplitude Gain Control (XOSCCTRL.AMPGC) is '1', the oscillator amplitude will be automatically adjusted, and in most cases result in a lower power consumption.

The XOSC will behave differently in different sleep modes, based on the settings of XOSCCTRL.RUNSTDBY, XOSCCTRL.ONDEMAND, and XOSCCTRL.ENABLE. If XOSCCTRL.ENABLE=0, the XOSC will be always stopped. For XOSCCTRL.ENABLE=1, this table is valid:

Table 21-1. XOSC Sleep Behavior

CPU Mode	XOSCCTRL.RUNSTDBY	XOSCCTRL.ONDEMAND	Sleep Behavior
Active or Idle	-	0	Always run
Active or Idle	-	1	Run if requested by peripheral
Standby	1	0	Always run
Standby	1	1	Run if requested by peripheral
Standby	0	-	Run if requested by peripheral

After a hard reset, or when waking up from a sleep mode where the XOSC was disabled, the XOSC will need a certain amount of time to stabilize on the correct frequency. This start-up time can be configured by changing the Oscillator Start-Up Time bit group (XOSCCTRL.STARTUP) in the External Multipurpose Crystal Oscillator Control register. During the start-up time, the oscillator output is masked to ensure that no unstable clock propagates to the digital logic.

The External Multipurpose Crystal Oscillator Ready bit in the Status register (STATUS.XOSCRDY) is set once the external clock or crystal oscillator is stable and ready to be used as a clock source. An interrupt is generated on a zero-to-one transition on STATUS.XOSCRDY if the External Multipurpose Crystal Oscillator Ready bit in the Interrupt Enable Set register (INTENSET.XOSCRDY) is set.

Related Links

GCLK - Generic Clock Controller on page 131

21.6.3. 16MHz Internal Oscillator (OSC16M) Operation

The OSC16M is an internal oscillator operating in open-loop mode and generating 4, 8, 12, or 16MHz frequency. The OSC16M frequency is selected by writing to the Frequency Select field in the OSC16M register (OSC16MCTRL.FSEL). OSC16M is enabled by writing '1' to the Oscillator Enable bit in the OSC16M Control register (OSC16MCTRL.ENABLE), and disabled by writing a '0' to this bit. Frequency selection must be done when OSC16M is disabled.

After enabling OSC16M, the OSC16M clock is output as soon as the oscillator is ready (STATUS.OSC16MRDY=1). User must ensure that the OSC16M is fully disabled before enabling it by reading STATUS.OSC16MRDY=0.

After reset, OSC16M is enabled and serves as the default clock source at 4MHz.

OSC16M will behave differently in different sleep modes based on the settings of OSC16MCTRL.RUNSTDBY, OSC16MCTRL.ONDEMAND, and OSC16MCTRL.ENABLE. If OSC16MCTRL.ENABLE=0, the OSC16M will be always stopped. For OSC16MCTRL.ENABLE=1, this table is valid:

Table 21-2. OSC16M Sleep Behavior

CPU Mode	OSC16MCTRL.RUNSTDBY	OSC16MCTRL.ONDEMAND	Sleep Behavior
Active or Idle	-	0	Always run
Active or Idle	-	1	Run if requested by peripheral
Standby	1	0	Always run
Standby	1	1	Run if requested by peripheral
Standby	0	-	Run if requested by peripheral

OSC16M is used as a clock source for the generic clock generators. This is configured by the Generic Clock Generator Controller.

Related Links

GCLK - Generic Clock Controller on page 131

21.6.4. Digital Frequency Locked Loop (DFLL48M) Operation

The DFLL48M can operate in both open-loop mode and closed-loop mode. In closed-loop mode, a low-frequency clock with high accuracy should be used as the reference clock to get high accuracy on the output clock (CLK_DFLL48M).

The DFLL48M can be used as a source for the generic clock generators.

Related Links

GCLK - Generic Clock Controller on page 131

21.6.4.1. Basic Operation

Open-Loop Operation

After any reset, the open-loop mode is selected. When operating in open-loop mode, the output frequency of the DFLL48M clock, CLK_DFLL48M, will be determined by the values written to the DFLL Coarse Value bit group and the DFLL Fine Value bit group (DFLLVAL.COARSE and DFLLVAL.FINE) in the DFLL Value register. Using "DFLL48M COARSE CAL" value from the Non Volatile Memory Software Calibration Area in DFLL.COARSE helps to output a frequency close to 48MHz.

It is possible to change the values of DFLLVAL.COARSE and DFLLVAL.FINE while the DFLL48M is enabled and in use, and thereby to adjust the output frequency of CLK_DFLL48M.

CLK_DFLL48M is ready to be used when STATUS.DFLLRDY is set after enabling the DFLL48M.

Closed-Loop Operation

In closed-loop operation, the DFLL48M output frequency is continuously regulated against a precise reference clock of relatively low frequency. This will improve the accuracy and stability of the CLK_DFLL48M clock in comparison to the open-loop (free-running) configuration.

Before closed-loop operation can be enabled, the DFLL48M must be enabled and configured in the following way:

1. Enable and select a reference clock (CLK_DFLL48M_REF). CLK_DFLL48M_REF is Generic Clock Channel 0 (DFLL48M_Reference).
2. Select the maximum step size allowed for finding the Coarse and Fine values by writing the appropriate values to the DFLL Coarse Maximum Step and DFLL Fine Maximum Step bit groups (DFLLMUL.CSTEP and DFLLMUL.FSTEP) in the DFLL Multiplier register.
   A small step size will ensure low overshoot on the output frequency, but it will typically take longer until locking is achieved. A high value might give a large overshoot, but will typically provide faster locking.

   DFLLMUL.CSTEP and DFLLMUL.FSTEP should not be higher than 50% of the maximum value of DFLLVAL.COARSE and DFLLVAL.FINE, respectively.
3. Select the multiplication factor in the DFLL Multiply Factor bit group (DFLLMUL.MUL) in the DFLL Multiplier register.
   Note: When choosing DFLLMUL.MUL, the output frequency must not exceed the maximum frequency of the device.
   If the target frequency is below the minimum frequency of the DFLL48M, the output frequency will be equal to the DFLL minimum frequency.
4. Start the closed loop mode by writing '1' to the DFLL Mode Selection bit in the DFLL Control register (DFLLCTRL.MODE). See Frequency Locking on page 222 for details.

The frequency of CLK_DFLL48M ($F_{clkdfll48m}$) is given by:

$$F_{clkdfll48m} = DFLLMUL \cdot MUL \times F_{clkdfll48m\_ref}$$

where $F_{clkdfll48m\_ref}$ is the frequency of the reference clock (CLK_DFLL48M_REF).

Related Links

GCLK - Generic Clock Controller on page 131

Frequency Locking

After enabling closed-loop operation by writing DFLLCTRL.MODE=1, the Coarse Value and the Fine Value bit fields in the DFLL48M Value register (DFLLVAL.COARSE and DFLLVAL.FINE) are used as starting parameters for the locking procedure.

Note: DFLLVAL.COARSE and DFLLVAL.FINE are read-only in closed-loop mode, and are controlled by the frequency tuner to meet user specified frequency.

The frequency locking is divided into two stages: coarse and fine lock.

Coarse Lock. Starting from the original DFLLVAL.COARSE and DFLLVAL.FINE, the control logic quickly finds the correct value for DFLLVAL.COARSE and sets the output frequency to a value close to the correct frequency. On coarse lock, the DFLL Locked on Coarse Value bit (STATUS.DFLLLCKC) in the Status register will be set.

Fine Lock. In this stage, the control logic tunes the value in DFLLVAL.FINE so that the output frequency is very close to the desired frequency. On fine lock, the DFLL Locked on Fine Value bit (STATUS.DFLLLCKF) in the Status register will be set.

Interrupts are generated by STATUS.DFLLLCKC and STATUS.DFLLLCKF, if INTENSET.DFLLLCKC or INTENSET.DFLLLCKF, respectively, are written to '1'.

The accuracy of the output frequency depends on which locks are set.

Note: Writing DFLLVAL.COARSE to a value close to the final value before entering closed-loop mode will reduce the time needed to get a lock on Coarse.

For a DFLL48M output frequency of 48MHz, the bit field "DFLL48M COARSE CAL" in the NVM Software Calibration Area provides a matching value for DFLL.COARSE, and will start DFLL with a frequency close to 48MHz.

This procedure will reduce the locking time to only the DFLL Fine Lock time:

1. Load the "DFLL48M COARSE CAL" value from the NVM Software Calibration Area into the DFLL.COARSE bit field.
2. Enable the Bypass Coarse Lock (DFLLCTRL.BPLCKC=1).
3. Start DFLL close loop (DFLLCTRL.MODE=1).

Frequency Error Measurement

The ratio between CLK_DFLL48M_REF and CLK48M_DFLL is measured automatically when the DFLL48M is in closed-loop mode. The difference between this ratio and the value in DFLLMUL.MUL is stored in the DFLL Multiplication Ratio Difference bit group (DFLLVAL.DIFF) in the DFLL Value register.

The relative error of CLK_DFLL48M with respect to the target frequency is calculated as follows:

$$ERROR = \frac{\text{DFLLVAL.DIFF}}{\text{DFLLMUL.MUL}}$$

Drift Compensation

If the Stable DFLL Frequency bit (DFLLCTRL.STABLE) in the DFLL Control register is '0', the frequency tuner will automatically compensate for drift in the CLK_DFLL48M without losing either of the locks.

Note: This means that DFLLVAL.FINE can change after every measurement of CLK_DFLL48M.

The DFLLVAL.FINE value may overflow or underflow in closed-loop mode due to large drift/instability of the clock source reference, and the DFLL Out Of Bounds bit (STATUS.DFLLOOB) in the Status register will be set. After an Out of Bounds error condition, the user must rewrite DFLLMUL.MUL to ensure correct CLK_DFLL48M frequency.

A zero-to-one transition of STATUS.DFLLOOB will generate an interrupt, if the DFLL Out Of Bounds bit in the Interrupt Enable Set register (INTENSET.DFLLOOB) is '1'. This interrupt will also be set if the tuner is not able to lock on the correct Coarse value.

To avoid this out-of-bounds error, the reference clock must be stable; an external oscillator XOSC32K is recommended.

Reference Clock Stop Detection

If CLK_DFLL48M_REF stops or is running at a very low frequency (slower than CLK_DFLL48M/(2 * MUL_{MAX})), the DFLL Reference Clock Stopped bit in the Status register (STATUS.DFLLRCS) will be set.

Detecting a stopped reference clock can take a long time, in the order of 217 CLK_DFLL48M cycles.

When the reference clock is stopped, the DFLL48M will operate as if in open-loop mode. Closed-loop mode operation will automatically resume when the CLK_DFLL48M_REF is restarted.

A zero-to-one transition of the DFLL Reference Clock Stopped bit in the Status register (STATUS.DFLLRCS) will generate an interrupt, if the DFLL Reference Clock Stopped bit in the Interrupt Enable Set register (INTENSET.DFLLRCS) is '1'.

21.6.4.2. Additional Features

Dealing with Settling Time in Closed-Loop Mode

The time from selecting a new CLK_DFLL48M output frequency until this frequency is output by the DFLL48M can be up to several microseconds. A small value in DFLLMUL.MUL can lead to instability in the DFLL48M locking mechanism, which can prevent the DFLL48M from achieving locks.

To avoid this, a chill cycle can be enabled, during which the CLK_DFLL48M frequency is not measured. The chill cycle is enabled by default, but can be disabled by writing '1' to the DFLL Chill Cycle Disable bit in the DFLL Control register (DFLLCTRL.CCDIS). Enabling chill cycles might double the lock time.

Another solution to this problem is using less strict lock requirements. This is called Quick Lock (QL). QL is enabled by default as well, but it can be disabled by writing '1' to the Quick Lock Disable bit in the DFLL Control register (DFLLCTRL.QLDIS). The Quick Lock might lead to a larger spread in the output frequency than chill cycles, but the average output frequency is the same.

USB Clock Recovery Module

USB Clock Recovery mode can be used to create the 48MHz USB clock from the USB Start Of Frame (SOF). This mode is enabled by writing a '1' to both the USB Clock Recovery Mode bit and the Mode bit in DFLL Control register (DFLLCTRL.USBCRM and DFLLCTRL.MODE).

The SOF signal from USB device will be used as reference clock (CLK_DFLL_REF), ignoring the selected generic clock reference. When the USB device is connected, a SOF will be sent every 1ms, thus DFLLVAL.MUX bits should be written to 0xBB80 to obtain a 48MHz clock.

In USB clock recovery mode, the DFLLCTRL.BPLCKC bit state is ignored, and the value stored in the DFLLVAL.COARSE will be used as final Coarse Value. The COARSE calibration value can be loaded from NVM OTP row by software. The locking procedure will also go instantaneously to the fine lock search.

The DFLLCTRL.QLDIS bit must be cleared and DFLLCTRL.CCDIS should be set to speed up the lock phase. The DFLLCTRL.STABLE bit state is ignored, an auto jitter reduction mechanism is used instead.

Wake from Sleep Modes

DFLL48M can optionally reset its lock bits when it is disabled. This is configured by the Lose Lock After Wake bit in the DFLL Control register (DFLLCTRL.LLAW).

If DFLLCTRL.LLAW is zero, the DFLL48M will be re-enabled and start running with the same configuration as before being disabled, even if the reference clock is not available. The locks will not be lost. After the reference clock has restarted, the fine lock tracking will quickly compensate for any frequency drift during sleep if DFLLCTRL.STABLE is zero.

If DFLLCTRL.LLAW is '1' when disabling the DFLL48M, the DFLL48M will lose all its locks, and needs to regain these through the full lock sequence.

Accuracy

There are three main factors that determine the accuracy of $F_{clkdfll48m}$. These can be tuned to obtain maximum accuracy when fine lock is achieved.

- Fine resolution. The frequency step between two Fine values. This is relatively smaller for higher output frequencies.
- Resolution of the measurement: If the resolution of the measured $F_{clkdfll48m}$ is low, i.e., the ratio between the CLK_DFLL48M frequency and the CLK_DFLL48M_REF frequency is small, the

DFLL48M might lock at a frequency that is lower than the targeted frequency. It is recommended to use a reference clock frequency of 32KHz or lower to avoid this issue for low target frequencies.

- The accuracy of the reference clock.

21.6.5. Digital Phase Locked Loop (DPLL) Operation

The task of the DPLL is to maintain coherence between the input (reference) signal and the respective output frequency, CLK_DPLL, via phase comparison. The DPLL controller supports three independent sources of reference clocks:

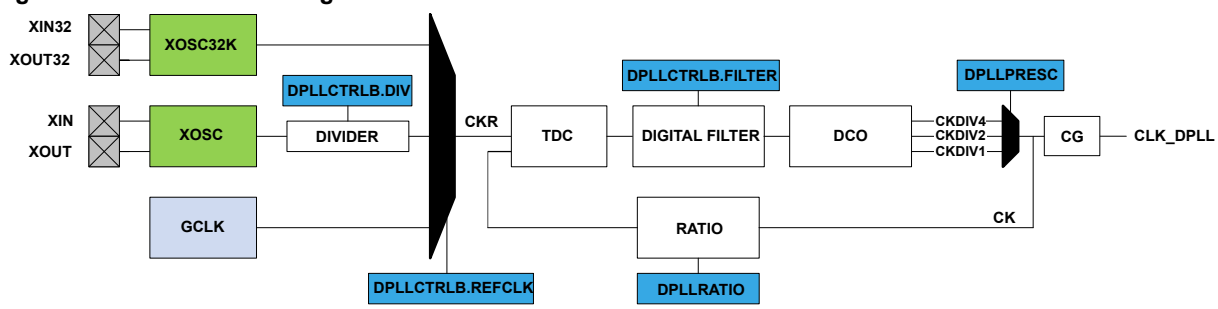
- XOSC32K: this clock is provided by the 32K External Crystal Oscillator (XOSC32K).
- XOSC: this clock is provided by the External Multipurpose Crystal Oscillator (XOSC).
- GCLK: this clock is provided by the Generic Clock Controller.

When the controller is enabled, the relationship between the reference clock frequency and the output clock frequency is:

$$f_{CK} = f_{CKR} \times \left(LDR + 1 + \frac{LDRFRAC}{16}\right) \times \frac{1}{2^{PRESC}}$$

Where f_{CK} is the frequency of the DPLL output clock, LDR is the loop divider ratio integer part, LDRFRAC is the loop divider ratio fractional part, f_{CKR} is the frequency of the selected reference clock, and PRESC is the output prescaler value.

Figure 21-2. DPLL Block Diagram



When the controller is disabled, the output clock is low. If the Loop Divider Ratio Fractional part bit field in the DPLL Ratio register (DPLLRATIO.LDRFRAC) is zero, the DPLL works in integer mode. Otherwise, the fractional mode is activated. Note that the fractional part has a negative impact on the jitter of the DPLL.

> Example (integer mode only): assuming F_{CKR} = 32kHz and F_{CK} = 48MHz, the multiplication ratio is 1500. It means that LDR shall be set to 1499.

> Example (fractional mode): assuming F_{CKR} = 32kHz and F_{CK} = 48.006MHz, the multiplication ratio is 1500.1875 (1500 + 3/16). Thus LDR is set to 1499 and LDRFRAC to 3.

Related Links

GCLK - Generic Clock Controller on page 131

OSC32KCTRL – 32KHz Oscillators Controller on page 264

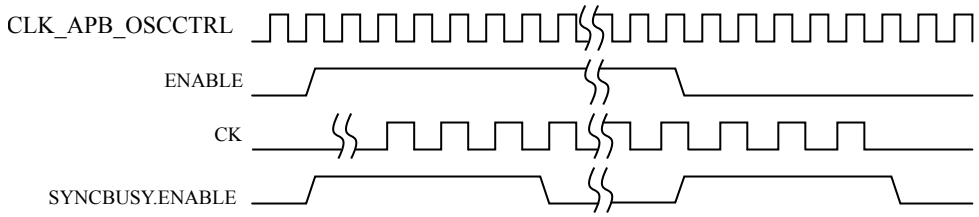
21.6.5.1. Basic Operation

Initialization, Enabling, Disabling, and Resetting

The DPLLC is enabled by writing a '1' to the Enable bit in the DPLL Control A register (DPLLCTRLA.ENABLE). The DPLLC is disabled by writing a zero to this bit.

The DPLLSYNCBUSY.ENABLE is set when the DPLLCTRLA.ENABLE bit is modified. It is cleared when the DPLL output clock CK has sampled the bit at the high level after enabling the DPLL. When disabling the DPLL, DPLLSYNCBUSY.ENABLE is cleared when the output clock is no longer running.

Figure 21-3. Enable Synchronization Busy Operation



The frequency of the DPLL output clock CK is stable when the module is enabled and when the Lock bit in the DPLL Status register is set (DPLLSTATUS.LOCK).

When the Lock Time bit field in the DPLL Control B register (DPLLCTRLB.LTIME) is non-zero, a user defined lock time is used to validate the lock operation. In this case the lock time is constant. If DPLLCTRLB.LTIME=0, the lock signal is linked with the status bit of the DPLL, and the lock time varies depending on the filter selection and the final target frequency.

When the Wake Up Fast bit (DPLLCTRLB.WUF) is set, the wake up fast mode is activated. In this mode the clock gating cell is enabled at the end of the startup time. At this time the final frequency is not stable, as it is still during the acquisition period, but it allows to save several milliseconds. After first acquisition, the Lock Bypass bit (DPLLCTRLB.LBYPASS) indicates if the lock signal is discarded from the control of the clock gater (CG) generating the output clock CLK_DPLL.

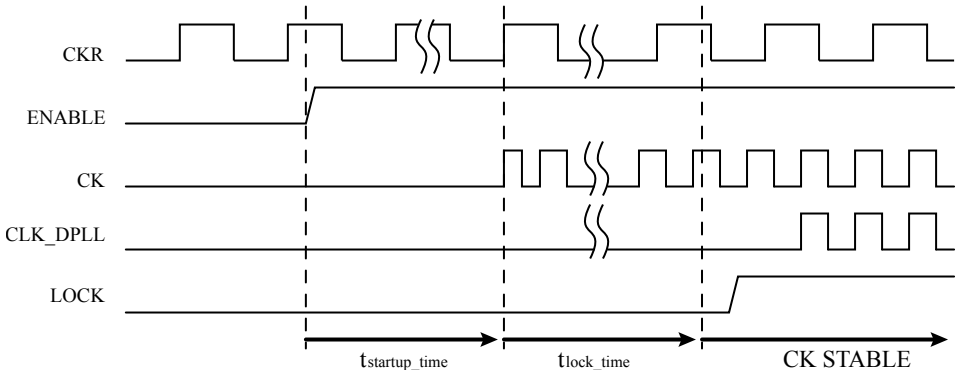
Table 21-3. CLK_DPLL Behavior from Startup to First Edge Detection

WUF	LTIME	CLK_DPLL Behavior
0	0	Normal Mode: First Edge when lock is asserted
0	Not Equal To Zero	Lock Timer Timeout mode: First Edge when the timer down-counts to 0.
1	X	Wake Up Fast Mode: First Edge when CK is active (startup time)

Table 21-4. CLK_DPLL Behavior after First Edge Detection

LBYPASS	CLK_DPLL Behavior
0	Normal Mode: the CLK_DPLL is turned off when lock signal is low.
1	Lock Bypass Mode: the CLK_DPLL is always running, lock is irrelevant.

Figure 21-4. CK and CLK_DPLL Output from DPLL Off Mode to Running Mode



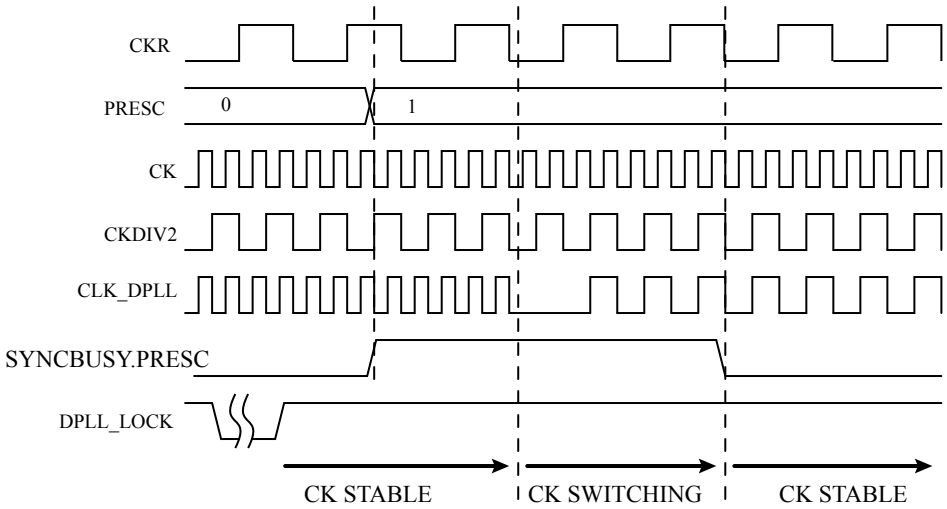
Reference Clock Switching

When a software operation requires reference clock switching, the recommended procedure is to turn the DPLL into the standby mode, modify the DPLLCTRLB.REFCLK to select the desired reference source, and activate the DPLL again.

Output Clock Prescaler

The DPLL controller includes an output prescaler. This prescaler provides three selectable output clocks CK, CKDIV2 and CKDIV4. The Prescaler bit field in the DPLL Prescaler register (DPLLPRESC.PRESC) is used to select a new output clock prescaler. When the prescaler field is modified, the DPLLSYNCBUSY.DPLLPRESC bit is set. It will be cleared by hardware when the synchronization is over.

Figure 21-5. Output Clock Switching Operation

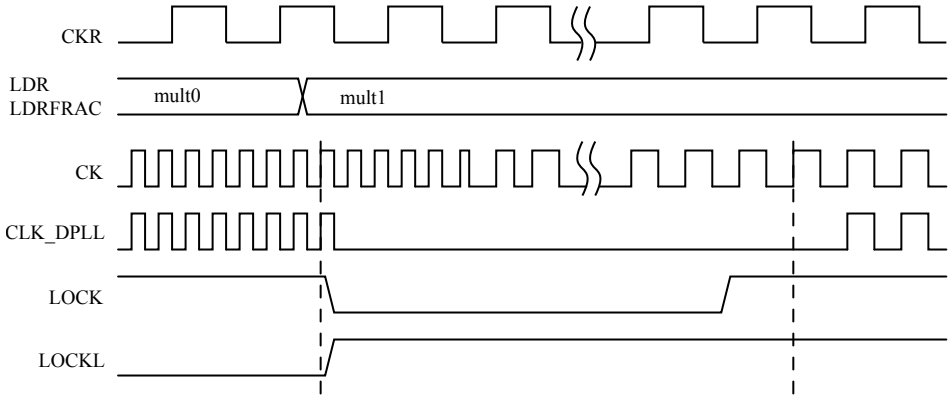


Loop Divider Ratio Updates

The DPLL Controller supports on-the-fly update of the DPLL Ratio Control (DPLLRATIO) register, allowing to modify the loop divider ratio and the loop divider ratio fractional part when the DPLL is enabled.

STATUS.DPLLLDRTO is set when the DPLLRATIO register has been modified and the DPLL analog cell has successfully sampled the updated value. At that time the DPLLSTATUS.LOCK bit is cleared and set again by hardware when the output frequency reached a stable state.

Figure 21-6. RATIOCTRL register update operation



Digital Filter Selection

The PLL digital filter (PI controller) is automatically adjusted in order to provide a good compromise between stability and jitter. Nevertheless a software operation can override the filter setting using the Filter bit field in the DPLL Control B register (DPLLCTRLB.FILTER). The Low Power Enable bit (DPLLCTRLB.LPEN) can be use to bypass the Time to Digital Converter (TDC) module.

21.6.6. DMA Operation

Not applicable.

21.6.7. Interrupts

The OSCCTRL has the following interrupt sources:

- XOSCRDY - Multipurpose Crystal Oscillator Ready: A 0-to-1 transition on the STATUS.XOSCRDY bit is detected
- CLKFAIL - Clock Failure. A 0-to-1 transition on the STATUS.CLKFAIL bit is detected
- OSC16MRDY - 16MHz Internal Oscillator Ready: A 0-to-1 transition on the STATUS.OSC16MRDY bit is detected
- DFLL-related:
  - DFLLRDY - DFLL48M Ready: A 0-to-1 transition of the STATUS.DFLLRDY bit is detected
  - DFLLOOB - DFLL48M Out Of Boundaries: A 0-to-1 transition of the STATUS.DFLLOOB bit is detected
  - DFLLLOCKF - DFLL48M Fine Lock: A 0-to-1 transition of the STATUS.DFLLLOCKF bit is detected
  - DFLLLOCKC - DFLL48M Coarse Lock: A 0-to-1 transition of the STATUS.DFLLLOCKC bit is detected
  - DFLLRCS - DFLL48M Reference Clock has Stopped: A 0-to-1 transition of the STATUS.DFLLRCS bit is detected
- DPLL-related:
  - DPLLLOCKR - DPLL Lock Rise: A 0-to-1 transition of the STATUS.DPLLLOCKR bit is detected
  - DPLLLOCKF - DPLL Lock Fall: A 0-to-1 transition of the STATUS.DPLLLOCKF bit is detected
  - DPLLLTTO - DPLL Lock Timer Time-out: A 0-to-1 transition of the STATUS.DPLLLTTO bit is detected
  - DPLLLDRTO - DPLL Loop Divider Ratio Update Complete. A 0-to-1 transition of the STATUS.DPLLLDRTO bit is detected

All these interrupts are synchronous wake-up source.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) is set when the interrupt condition occurs.

Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR).

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the OSCCTRL is reset. See the INTFLAG register for details on how to clear interrupt flags.

The OSCCTRL has one common interrupt request line for all the interrupt sources. The user must read the INTFLAG register to determine which interrupt condition is present. Refer to the INTFLAG register for details.

Note: The interrupts must be globally enabled for interrupt requests to be generated.

21.6.8. Events

Not applicable.

21.6.9. Synchronization

DFLL48M

Due to the multiple clock domains, values in the DFLL48M control registers need to be synchronized to other clock domains.

Once the DFLL is enabled, any read and write operation requires the DFLL Ready bit in the Status register (STATUS.DFLLRDY) to read '1'.
Note: Once the DFLL48M is enabled in on-demand mode (DFLLCTRL.ONDEMAND=1), the STATUS.DFLLRDY bit will keep to '0' until the DFLL48M is requested by a peripheral.

Before writing to any of the DFLL48M control registers, the user must check that the DFLL Ready bit (STATUS.DFLLRDY) is set to '1'. When this bit is set, the DFLL48M can be configured and CLK_DFLL48M is ready to be used. Any write to any of the DFLL48M control registers while DFLLRDY is '0' will be ignored.

In order to read from the DFLLVAL register in closed loop mode, the user must request a read synchronization by writing a '1' to the Read Request bit in the DFLL Synchronization register (DFLLSYNC.READREQ). This is required because the DFLL controller may change the content of the DFLLVAL register any time. If a read operation is issued while the DFLL controller is updating the DFLLVAL content, a zero will be returned.

Note: Issuing a read on any register while a write-synchronization is still on-going will return a zero.

Read-Synchronized registers using DFLLSYNC.READREQ:

- DFLL48M Value register (DFLLVAL)

Write-Synchronized registers:

- DFLL48M Control register (DFLLCTRL)
- DFLL48M Value register (DFLLVAL)
- DFLL48M Multiplier register (DFLLMUL)

DPLL96M

Due to the multiple clock domains, some registers in the DPLL96M must be synchronized when accessed.

When executing an operation that requires synchronization, the relevant synchronization bit in the Synchronization Busy register (DPLLSYNCBUSY) will be set immediately, and cleared when synchronization is complete.

The following bits need synchronization when written:

- Enable bit in control register A (DPLLCTRLA.ENABLE)
- DPLL Ratio register (DPLLRATIO)
- DPLL Prescaler register (DPLLPRESC)

Related Links

Register Synchronization on page 127

21.7. Register Summary

Offset	Name	Bit Pos.								
0x00	INTENCLR	7:0				OSC16MRDY				XOSCRDY
0x01		15:8				DFLLRCS	DFLLLCKC	DFLLLCKF	DFLLOOB	DFLLRDY
0x02		23:16					DPLLLDRTO	DPLLLTO	DPLLLCKF	DPLLLCKR
0x03		31:24								
0x04	INTENSET	7:0				OSC16MRDY				XOSCRDY
0x05		15:8				DFLLRCS	DFLLLCKC	DFLLLCKF	DFLLOOB	DFLLRDY
0x06		23:16					DPLLLDRTO	DPLLLTO	DPLLLCKF	DPLLLCKR
0x07		31:24								
0x08	INTFLAG	7:0				OSC16MRDY				XOSCRDY
0x09		15:8				DFLLRCS	DFLLLCKC	DFLLLCKF	DFLLOOB	DFLLRDY
0x0A		23:16					DPLLLDRTO	DPLLLTO	DPLLLCKF	DPLLLCKR
0x0B		31:24								
0x0C	STATUS	7:0				OSC16MRDY		CLKSW	CLKFAIL	XOSCRDY
0x0D		15:8				DFLLRCS	DFLLLCKC	DFLLLCKF	DFLLOOB	DFLLRDY
0x0E		23:16					DPLLLDRTO	DPLLLTO	DPLLLCKF	DPLLLCKR
0x0F		31:24								
0x10	XOSCCTRL	7:0	ONDEMAND	RUNSTDBY				XTALEN	ENABLE	
0x11		15:8	STARTUP[3:0]				AMPGC	GAIN[2:0]		
0x12 ... 0x13	Reserved									
0x14	OSC16MCTRL	7:0	ONDEMAND	RUNSTDBY			FSEL[1:0]		ENABLE	
0x15 ... 0x17	Reserved									
0x18	DFLLCTRL	7:0	ONDEMAND	RUNSTDBY	USBCRM	LLAW	STABLE	MODE	ENABLE	
0x19		15:8							QLDIS	CCDIS
0x1A ... 0x1B	Reserved									
0x1C	DFLLVAL	7:0	FINE[7:0]							
0x1D		15:8	COARSE[5:0]						FINE[9:8]	
0x1E		23:16	DIFF[7:0]							
0x1F		31:24	DIFF[15:8]							
0x20	DFLLMUL	7:0	MUL[7:0]							
0x21		15:8	MUL[15:8]							
0x22		23:16	FSTEP[7:0]							
0x23		31:24	CSTEP[5:0]						FSTEP[9:8]	
0x24	DFLLSYNC	7:0	READREQ							
0x25 ... 0x27	Reserved									
0x28	DPLLCTRLA	7:0	ONDEMAND	RUNSTDBY					ENABLE	

Offset	Name	Bit Pos.								
0x29 ... 0x2B	Reserved									
0x2C	DPLLRATIO	7:0	LDR[7:0]							
0x2D		15:8					LDR[11:8]			
0x2E		23:16					LDRFRAC[3:0]			
0x2F		31:24								
0x30	DPLLCTRLB	7:0			REFCLK[1:0]		WUF	LPEN	FILTER[1:0]	
0x31		15:8				LBYPASS		LTIME[2:0]		
0x32		23:16	DIV[7:0]							
0x33		31:24						DIV[10:8]		
0x34	DPLLPRESC	7:0							PRESC[1:0]	
0x35 ... 0x37	Reserved									
0x38	DPLLSYNCBUSY	7:0					DPLLPRESC	DPLLRATIO	ENABLE	
0x39 ... 0x3B	Reserved									
0x3C	DPLLSTATUS	7:0							CLKRDY	LOCK

21.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Write-protection is denoted by the "PAC Write-Protection" property in each individual register description. Refer to the Register Access Protection on page 219 section and the PAC - Peripheral Access Controller on page 58 chapter for details.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" or "Write.Synchronized" property in each individual register description. Refer to the Synchronization on page 229 section for details.

21.8.1. Interrupt Enable Set

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Name: INTENSET
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					DPLLLDRTO	DPLLLTO	DPLLLCKF	DPLLLCKR
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
				DFLLRCS	DFLLLCKC	DFLLLCKF	DFLLOOB	DFLLRDY
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
				OSC16MRDY				XOSCRDY
Access				R/W				R/W
Reset				0				0

Bit 19 – DPLLLDRTO: DPLL Loop Divider Ratio Update Complete Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the DPLL Loop Ratio Update Complete Interrupt Enable bit, which enables the DPLL Loop Ratio Update Complete interrupt.

Value	Description
0	The DPLL Loop Divider Ratio Update Complete interrupt is disabled.
1	The DPLL Loop Ratio Update Complete interrupt is enabled, and an interrupt request will be generated when the DPLL Loop Ratio Update Complete Interrupt flag is set.

Bit 18 – DPLLLTO: DPLL Lock Timeout Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the DPLL Lock Timeout Interrupt Enable bit, which enables the DPLL Lock Timeout interrupt.

Value	Description
0	The DPLL Lock Timeout interrupt is disabled.
1	The DPLL Lock Timeout interrupt is enabled, and an interrupt request will be generated when the DPLL Lock Timeout Interrupt flag is set.

Bit 17 – DPLLLCKF: DPLL Lock Fall Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the DPLL Lock Fall Interrupt Enable bit, which enables the DPLL Lock Fall interrupt.

Value	Description
0	The DPLL Lock Fall interrupt is disabled.
1	The DPLL Lock Fall interrupt is enabled, and an interrupt request will be generated when the DPLL Lock Fall Interrupt flag is set.

Bit 16 – DPLLLCKR: DPLL Lock Rise Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the DPLL Lock Rise Interrupt Enable bit, which enables the DPLL Lock Rise interrupt.

Value	Description
0	The DPLL Lock Rise interrupt is disabled.
1	The DPLL Lock Rise interrupt is enabled, and an interrupt request will be generated when the DPLL Lock Rise Interrupt flag is set.

Bit 12 – DFLLRCS: DFLL Reference Clock Stopped Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the DFLL Reference Clock Stopped Interrupt Enable bit, which enables the DFLL Reference Clock Stopped interrupt.

Value	Description
0	The DFLL Reference Clock Stopped interrupt is disabled.
1	The DFLL Reference Clock Stopped interrupt is enabled, and an interrupt request will be generated when the DFLL Reference Clock Stopped Interrupt flag is set.

Bit 11 – DFLLLCKC: DFLL Lock Coarse Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the DFLL Lock Coarse Interrupt Enable bit, which enables the DFLL Lock Coarse interrupt.

Value	Description
0	The DFLL Lock Coarse interrupt is disabled.
1	The DFLL Lock Coarse interrupt is enabled, and an interrupt request will be generated when the DFLL Lock Coarse Interrupt flag is set.

Bit 10 – DFLLLCKF: DFLL Lock Fine Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the DFLL Lock Fine Interrupt Disable/Enable bit, disable the DFLL Lock Fine interrupt and set the corresponding interrupt request.

Value	Description
0	The DFLL Lock Fine interrupt is disabled.
1	The DFLL Lock Fine interrupt is enabled, and an interrupt request will be generated when the DFLL Lock Fine Interrupt flag is set.

Bit 9 – DFLLOOB: DFLL Out Of Bounds Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the DFLL Out Of Bounds Interrupt Enable bit, which enables the DFLL Out Of Bounds interrupt.

Value	Description
0	The DFLL Out Of Bounds interrupt is disabled.
1	The DFLL Out Of Bounds interrupt is enabled, and an interrupt request will be generated when the DFLL Out Of Bounds Interrupt flag is set.

Bit 8 – DFLLRDY: DFLL Ready Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the DFLL Ready Interrupt Enable bit, which enables the DFLL Ready interrupt and set the corresponding interrupt request.

Value	Description
0	The DFLL Ready interrupt is disabled.
1	The DFLL Ready interrupt is enabled, and an interrupt request will be generated when the DFLL Ready Interrupt flag is set.

Bit 4 – OSC16MRDY: OSC16M Ready Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the OSC16M Ready Interrupt Enable bit, which enables the OSC16M Ready interrupt.

Value	Description
0	The OSC16M Ready interrupt is disabled.
1	The OSC16M Ready interrupt is enabled, and an interrupt request will be generated when the OSC16M Ready Interrupt flag is set.

Bit 0 – XOSCRDY: XOSC Ready Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the XOSC Ready Interrupt Enable bit, which enables the XOSC Ready interrupt.

Value	Description
0	The XOSC Ready interrupt is disabled.
1	The XOSC Ready interrupt is enabled, and an interrupt request will be generated when the XOSC Ready Interrupt flag is set.

21.8.2. Interrupt Enable Clear

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Name: INTENCLR
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					DPLLLDRTO	DPLLLTO	DPLLLCKF	DPLLLCKR
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
				DFLLRCS	DFLLLCKC	DFLLLCKF	DFLLOOB	DFLLRDY
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
				OSC16MRDY				XOSCRDY
Access				R/W				R/W
Reset				0				0

Bit 19 – DPLLLDRTO: DPLL Loop Divider Ratio Update Complete Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the DPLL Loop Divider Ratio Update Complete Interrupt Enable bit, which disables the DPLL Loop Divider Ratio Update Complete interrupt.

Value	Description
0	The DPLL Loop Divider Ratio Update Complete interrupt is disabled.
1	The DPLL Loop Divider Ratio Update Complete interrupt is enabled, and an interrupt request will be generated when the DPLL Loop Divider Ratio Update Complete Interrupt flag is set.

Bit 18 – DPLLLTO: DPLL Lock Timeout Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the DPLL Lock Timeout Interrupt Enable bit, which disables the DPLL Lock Timeout interrupt.

Value	Description
0	The DPLL Lock Timeout interrupt is disabled.
1	The DPLL Lock Timeout interrupt is enabled, and an interrupt request will be generated when the DPLL Lock Timeout Interrupt flag is set.

Bit 17 – DPLLLCKF: DPLL Lock Fall Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the DPLL Lock Fall Interrupt Enable bit, which disables the DPLL Lock Fall interrupt.

Value	Description
0	The DPLL Lock Fall interrupt is disabled.
1	The DPLL Lock Fall interrupt is enabled, and an interrupt request will be generated when the DPLL Lock Fall Interrupt flag is set.

Bit 16 – DPLLLCKR: DPLL Lock Rise Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the DPLL Lock Rise Interrupt Enable bit, which disables the DPLL Lock Rise interrupt.

Value	Description
0	The DPLL Lock Rise interrupt is disabled.
1	The DPLL Lock Rise interrupt is enabled, and an interrupt request will be generated when the DPLL Lock Rise Interrupt flag is set.

Bit 12 – DFLLRCS: DFLL Reference Clock Stopped Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the DFLL Reference Clock Stopped Interrupt Enable bit, which disables the DFLL Reference Clock Stopped interrupt.

Value	Description
0	The DFLL Reference Clock Stopped interrupt is disabled.
1	The DFLL Reference Clock Stopped interrupt is enabled, and an interrupt request will be generated when the DFLL Reference Clock Stopped Interrupt flag is set.

Bit 11 – DFLLLCKC: DFLL Lock Coarse Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the DFLL Lock Coarse Interrupt Enable bit, which disables the DFLL Lock Coarse interrupt.

Value	Description
0	The DFLL Lock Coarse interrupt is disabled.
1	The DFLL Lock Coarse interrupt is enabled, and an interrupt request will be generated when the DFLL Lock Coarse Interrupt flag is set.

Bit 10 – DFLLLCKF: DFLL Lock Fine Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the DFLL Lock Fine Interrupt Enable bit, which disables the DFLL Lock Fine interrupt.

Value	Description
0	The DFLL Lock Fine interrupt is disabled.
1	The DFLL Lock Fine interrupt is enabled, and an interrupt request will be generated when the DFLL Lock Fine Interrupt flag is set.

Bit 9 – DFLLOOB: DFLL Out Of Bounds Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the DFLL Out Of Bounds Interrupt Enable bit, which disables the DFLL Out Of Bounds interrupt.

Value	Description
0	The DFLL Out Of Bounds interrupt is disabled.
1	The DFLL Out Of Bounds interrupt is enabled, and an interrupt request will be generated when the DFLL Out Of Bounds Interrupt flag is set.

Bit 8 – DFLLRDY: DFLL Ready Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the DFLL Ready Interrupt Enable bit, which disables the DFLL Ready interrupt.

Value	Description
0	The DFLL Ready interrupt is disabled.
1	The DFLL Ready interrupt is enabled, and an interrupt request will be generated when the DFLL Ready Interrupt flag is set.

Bit 4 – OSC16MRDY: OSC16M Ready Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the OSC16M Ready Interrupt Enable bit, which disables the OSC16M Ready interrupt.

Value	Description
0	The OSC16M Ready interrupt is disabled.
1	The OSC16M Ready interrupt is enabled, and an interrupt request will be generated when the OSC16M Ready Interrupt flag is set.

Bit 0 – XOSCRDY: XOSC Ready Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the XOSC Ready Interrupt Enable bit, which disables the XOSC Ready interrupt.

Value	Description
0	The XOSC Ready interrupt is disabled.
1	The XOSC Ready interrupt is enabled, and an interrupt request will be generated when the XOSC Ready Interrupt flag is set.

21.8.3. Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x08
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					DPLLLDRTO	DPLLLTO	DPLLLCKF	DPLLLCKR
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
				DFLLRCS	DFLLLCKC	DFLLLCKF	DFLLOOB	DFLLRDY
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
				OSC16MRDY				XOSCRDY
Access				R/W				R/W
Reset				0				0

Bit 19 – DPLLLDRTO: DPLL Loop Divider Ratio Update Complete
This flag is cleared by writing '1' to it.

This flag is set on 0-to-1 transition of the DPLL Loop Divider Ratio Update Complete bit in the Status register (STATUS.DPLLLDRTO) and will generate an interrupt request if INTENSET.DPLLLDRTO is '1'.

Writing '0' to this bit has no effect.

Writing '1' to this bit clears the DPLL Loop Divider Ratio Update Complete interrupt flag.

Bit 18 – DPLLLTO: DPLL Lock Timeout
This flag is cleared by writing '1' to it.

This flag is set on 0-to-1 transition of the DPLL Lock Timeout bit in the Status register (STATUS.DPLLLTO) and will generate an interrupt request if INTENSET.DPLLLTO is '1'.

Writing '0' to this bit has no effect.

Writing '1' to this bit clears the DPLL Lock Timeout interrupt flag.

Bit 17 – DPLLLCKF: DPLL Lock Fall
This flag is cleared by writing '1' to it.

This flag is set on 0-to-1 transition of the DPLL Lock Fall bit in the Status register (STATUS.DPLLLCKF) and will generate an interrupt request if INTENSET.DPLLLCKF is '1'.

Writing '0' to this bit has no effect.

Writing '1' to this bit clears the DPLL Lock Fall interrupt flag.

Bit 16 – DPLLLCKR: DPLL Lock Rise
This flag is cleared by writing '1' to it.

This flag is set on 0-to-1 transition of the DPLL Lock Rise bit in the Status register (STATUS.DPLLLCKR) and will generate an interrupt request if INTENSET.DPLLLCKR is '1'.

Writing '0' to this bit has no effect.

Writing '1' to this bit clears the DPLL Lock Rise interrupt flag.

Bit 12 – DFLLRCS: DFLL Reference Clock Stopped
This flag is cleared by writing '1' to it.

This flag is set on 0-to-1 transition of the DFLL Reference Clock Stopped bit in the Status register (STATUS.DFLLRCS) and will generate an interrupt request if INTENSET.DFLLRCS is '1'.

Writing '0' to this bit has no effect.

Writing '1' to this bit clears the DFLL Reference Clock Stopped interrupt flag.

Bit 11 – DFLLLCKC: DFLL Lock Coarse
This flag is cleared by writing '1' to it.

This flag is set on 0-to-1 transition of the DFLL Lock Coarse bit in the Status register (STATUS.DFLLLCKC) and will generate an interrupt request if INTENSET.DFLLLCKC is '1'.

Writing '0' to this bit has no effect.

Writing '1' to this bit clears the DFLL Lock Coarse interrupt flag.

Bit 10 – DFLLLCKF: DFLL Lock Fine
This flag is cleared by writing '1' to it.

This flag is set on 0-to-1 transition of the DFLL Lock Fine bit in the Status register (STATUS.DFLLLCKF) and will generate an interrupt request if INTENSET.DFLLLCKF is '1'.

Writing '0' to this bit has no effect.

Writing '1' to this bit clears the DFLL Lock Fine interrupt flag.

Bit 9 – DFLLOOB: DFLL Out Of Bounds
This flag is cleared by writing '1' to it.

This flag is set on 0-to-1 transition of the DFLL Out Of Bounds bit in the Status register (STATUS.DFLLOOB) and will generate an interrupt request if INTENSET.DFLLOOB is '1'.

Writing '0' to this bit has no effect.

Writing '1' to this bit clears the DFLL Out Of Bounds interrupt flag.

Bit 8 – DFLLRDY: DFLL Ready
This flag is cleared by writing '1' to it.

This flag is set on 0-to-1 transition of the DFLL Ready bit in the Status register (STATUS.DFLLRDY) and will generate an interrupt request if INTENSET.DFLLRDY is '1'.

Writing '0' to this bit has no effect.

Writing '1' to this bit clears the DFLL Ready interrupt flag.

Bit 4 – OSC16MRDY: OSC16M Ready
This flag is cleared by writing '1' to it.

This flag is set on 0-to-1 transition of the OSC16M Ready bit in the Status register (STATUS.OSC16MRDY) and will generate an interrupt request if INTENSET.OSC16MRDY is '1'.

Writing '0' to this bit has no effect.

Writing '1' to this bit clears the OSC16M Ready interrupt flag.

Bit 0 – XOSCRDY: XOSC Ready
This flag is cleared by writing '1' to it.

This flag is set on a 0-to-1 transition of the XOSC Ready bit in the Status register (STATUS.XOSCRDY) and will generate an interrupt request if INTENSET.XOSCRDY is '1'.

Writing '0' to this bit has no effect.

Writing '1' to this bit clears the XOSC Ready interrupt flag.

21.8.4. Status

Name: STATUS
Offset: 0x0C
Reset: 0x00000100
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					DPLLLDRTO	DPLLLTO	DPLLLCKF	DPLLLCKR
Access					R	R	R	R
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
				DFLLRCS	DFLLLCKC	DFLLLCKF	DFLLOOB	DFLLRDY
Access				R	R	R	R	R
Reset				0	0	0	0	1

Bit	7	6	5	4	3	2	1	0
				OSC16MRDY		CLKSW	CLKFAIL	XOSCRDY
Access				R		R	R	R
Reset				0		0	0	0

Bit 19 – DPLLLDRTO: DPLL Loop Divider Ratio Update Complete

Value	Description
0	DPLL Loop Divider Ratio Update Complete not detected.
1	DPLL Loop Divider Ratio Update Complete detected.

Bit 18 – DPLLLTO: DPLL Lock Timeout

Value	Description
0	DPLL Lock time-out not detected.
1	DPLL Lock time-out detected.

Bit 17 – DPLLLCKF: DPLL Lock Fall

Value	Description
0	DPLL Lock fall edge not detected.
1	DPLL Lock fall edge detected.

Bit 16 – DPLLLCKR: DPLL Lock Rise

Value	Description
0	DPLL Lock rise edge not detected.
1	DPLL Lock fall edge detected.

Bit 12 – DFLLRCS: DFLL Reference Clock Stopped

Value	Description
0	DFLL reference clock is running.
1	DFLL reference clock has stopped.

Bit 11 – DFLLLCKC: DFLL Lock Coarse

Value	Description
0	No DFLL coarse lock detected.
1	DFLL coarse lock detected.

Bit 10 – DFLLLCKF: DFLL Lock Fine

Value	Description
0	No DFLL fine lock detected.
1	DFLL fine lock detected.

Bit 9 – DFLLOOB: DFLL Out Of Bounds

Value	Description
0	No DFLL Out Of Bounds detected.
1	DFLL Out Of Bounds detected.

Bit 8 – DFLLRDY: DFLL Ready

Value	Description
0	DFLL registers update is ongoing. Registers update is requested through DFLLSYNC.READREQ, or after a write access in DFLLCTRL, DFLLVAL or DFLLMUL register.
1	DFLL registers are stable and ready for read/write access.

Bit 4 – OSC16MRDY: OSC16M Ready

Value	Description
0	OSC16M is not ready.
1	OSC16M is stable and ready to be used as a clock source.

Bit 2 – CLKSW: XOSC Clock Switch

Value	Description
0	XOSC is not switched and provides the external clock or crystal oscillator clock.
1	XOSC is switched and provides the safe clock.

Bit 1 – CLKFAIL: XOSC Clock Failure

Value	Description
0	No XOSC failure detected.
1	A XOSC failure was detected.

Bit 0 – XOSCRDY: XOSC Ready

Value	Description
0	XOSC is not ready.
1	XOSC is stable and ready to be used as a clock source.

21.8.5. 16MHz Internal Oscillator (OSC16M) Control

Name: OSC16MCTRL
Offset: 0x14
Reset: 0x82
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY			FSEL[1:0]		ENABLE	
Access	R/W	R/W			R/W	R/W	R/W	
Reset	1	0			0	0	1	

Bit 7 – ONDEMAND: On Demand Control
The On Demand operation mode allows the oscillator to be enabled or disabled depending on peripheral clock requests.

If the ONDEMAND bit has been previously written to '1', the oscillator will only be running when requested by a peripheral. If there is no peripheral requesting the oscillator's clock source, the oscillator will be in a disabled state.

If On Demand is disabled the oscillator will always be running when enabled.

In standby sleep mode, the On Demand operation is still active.

Value	Description
0	The oscillator is always on, if enabled.
1	The oscillator is enabled when a peripheral is requesting the oscillator to be used as a clock source. The oscillator is disabled if no peripheral is requesting the clock source.

Bit 6 – RUNSTDBY: Run in Standby
This bit controls how the OSC16M behaves during standby sleep mode.

Value	Description
0	The OSC16M is disabled in standby sleep mode if no peripheral requests the clock.
1	The OSC16M is not stopped in standby sleep mode. If ONDEMAND=1, the OSC16M will be running when a peripheral is requesting the clock. If ONDEMAND=0, the clock source will always be running in standby sleep mode.

Bits 3:2 – FSEL[1:0]: Oscillator Frequency Selection
These bits control the oscillator frequency range.

Value	Description
0x00	4MHz
0x01	8MHz
0x10	12MHz
0x11	16MHz

Bit 1 – ENABLE: Oscillator Enable

Value	Description
0	The oscillator is disabled.
1	The oscillator is enabled.

21.8.6. External Multipurpose Crystal Oscillator (XOSC) Control

Name: XOSCCTRL
Offset: 0x10
Reset: 0x0080
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
	STARTUP[3:0]				AMPGC	GAIN[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY				XTALEN	ENABLE	
Access	R/W	R/W				R/W	R/W	
Reset	1	0				0	0	

Bits 15:12 – STARTUP[3:0]: Start-Up Time
These bits select start-up time for the oscillator.

The OSCULP32K oscillator is used to clock the start-up counter.

Table 21-5. Start-Up Time for External Multipurpose Crystal Oscillator

STARTUP[3:0]	Number of OSCULP32K Clock Cycles	Number of XOSC Clock Cycles	Approximate Equivalent Time [µs]
0x0	1	3	31
0x1	2	3	61
0x2	4	3	122
0x3	8	3	244
0x4	16	3	488
0x5	32	3	977
0x6	64	3	1953
0x7	128	3	3906
0x8	256	3	7813
0x9	512	3	15625
0xA	1024	3	31250
0xB	2048	3	62500µs
0xC	4096	3	125000
0xD	8192	3	250000
0xE	16384	3	500000
0xF	32768	3	1000000

Note:

1. Actual startup time is 1 OSCULP32K cycle + 3 XOSC cycles.
2. The given time neglects the three XOSC cycles before OSCULP32K cycle.

Bit 11 – AMPGC: Automatic Amplitude Gain Control
Note: This bit must be set only after the XOSC has settled, indicated by the XOSC Ready flag in the Status register (STATUS.XOSCRDY).

Value	Description
0	The automatic amplitude gain control is disabled.
1	The automatic amplitude gain control is enabled. Amplitude gain will be automatically adjusted during Crystal Oscillator operation.

Bits 10:8 – GAIN[2:0]: Oscillator Gain
These bits select the gain for the oscillator. The listed maximum frequencies are recommendations, and might vary based on capacitive load and crystal characteristics. Those bits must be properly configured even when the Automatic Amplitude Gain Control is active.

Value	Recommended Max Frequency [MHz]
0x0	2
0x1	4
0x2	8
0x3	16
0x4	30
0x5-0x7	Reserved

Bit 7 – ONDEMAND: On Demand Control
The On Demand operation mode allows the oscillator to be enabled or disabled, depending on peripheral clock requests.

If the ONDEMAND bit has been previously written to '1', the oscillator will be running only when requested by a peripheral. If there is no peripheral requesting the oscillator's clock source, the oscillator will be in a disabled state.

If On Demand is disabled, the oscillator will always be running when enabled.

In standby sleep mode, the On Demand operation is still active.

Value	Description
0	The oscillator is always on, if enabled.
1	The oscillator is enabled when a peripheral is requesting the oscillator to be used as a clock source. The oscillator is disabled if no peripheral is requesting the clock source.

Bit 6 – RUNSTDBY: Run in Standby
This bit controls how the XOSC behaves during standby sleep mode, together with the ONDEMAND bit:

Value	Description
0	The XOSC is not running in Standby sleep mode if no peripheral requests the clock.
1	The XOSC is running in Standby sleep mode. If ONDEMAND=1, the XOSC will be running when a peripheral is requesting the clock. If ONDEMAND=0, the clock source will always be running in Standby sleep mode.

Bit 2 – XTALEN: Crystal Oscillator Enable

This bit controls the connections between the I/O pads and the external clock or crystal oscillator:

Value	Description
0	External clock connected on XIN. XOUT can be used as general-purpose I/O.
1	Crystal connected to XIN/XOUT.

Bit 1 – ENABLE: Oscillator Enable

Value	Description
0	The oscillator is disabled.
1	The oscillator is enabled.

21.8.7. DFLL48M Control

Name: DFLLCTRL
Offset: 0x18
Reset: 0x0080
Property: PAC Write-Protection, Write-Synchronized using STATUS.DFLLRDY=1

Bit	15	14	13	12	11	10	9	8
							QLDIS	CCDIS
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY	USBCRM	LLAW	STABLE	MODE	ENABLE	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	1	0	0	0	0	0	0	

Bit 9 – QLDIS: Quick Lock Disable

Value	Description
0	Quick Lock is enabled.
1	Quick Lock is disabled.

Bit 8 – CCDIS: Chill Cycle Disable

Value	Description
0	Chill Cycle is enabled.
1	Chill Cycle is disabled.

Bit 7 – ONDEMAND: On Demand Control
The On Demand operation mode allows the DFLL to be enabled or disabled depending on peripheral clock requests.

If the ONDEMAND bit has been previously written to '1', the DFLL will only be running when requested by a peripheral. If there is no peripheral requesting the DFLL clock source, the DFLL will be in a disabled state.

If On Demand is disabled, the DFLL will always be running when enabled.

In standby sleep mode, the On Demand operation is still active.

Value	Description
0	The DFLL is always on, if enabled.
1	The DFLL is enabled when a peripheral is requesting the DFLL to be used as a clock source. The DFLL is disabled if no peripheral is requesting the clock source.

Bit 6 – RUNSTDBY: Run in Standby
This bit controls how the DFLL behaves during standby sleep mode:

Value	Description
0	The DFLL is disabled in standby sleep mode if no peripheral requests the clock.
1	The DFLL is not stopped in standby sleep mode. If ONDEMAND is one, the DFLL will be running when a peripheral is requesting the clock. If ONDEMAND is zero, the clock source will always be running in standby sleep mode.

Bit 5 – USBCRM: USB Clock Recovery Mode

Value	Description
0	USB Clock Recovery Mode is disabled.
1	USB Clock Recovery Mode is enabled.

Bit 4 – LLAW: Lose Lock After Wake

Value	Description
0	Locks will not be lost after waking up from sleep modes if the DFLL clock has been stopped.
1	Locks will be lost after waking up from sleep modes if the DFLL clock has been stopped.

Bit 3 – STABLE: Stable DFLL Frequency

Value	Description
0	FINE calibration tracks changes in output frequency.
1	FINE calibration register value will be fixed after a fine lock.

Bit 2 – MODE: Operating Mode Selection

Value	Description
0	The DFLL operates in open-loop operation.
1	The DFLL operates in closed-loop operation.

Bit 1 – ENABLE: DFLL Enable

Due to synchronization, there is delay from updating the register until the peripheral is enabled/disabled. The value written to DFLLCTRL.ENABLE will read back immediately after written.

Value	Description
0	The DFLL oscillator is disabled.
1	The DFLL oscillator is enabled.

21.8.8. DFLL48M Value

Name: DFLLVAL
Offset: 0x1C
Reset: 0x00000000
Property: PAC Write-Protection, Read-Synchronized using DFLLSYNC.READREQ, Write-Synchronized using STATUS.DFLLRDY=1

Bit	31	30	29	28	27	26	25	24
	DIFF[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DIFF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	COARSE[5:0]						FINE[9:8]	
Access	R	R	R	R	R	R	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FINE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – DIFF[15:0]: Multiplication Ratio Difference
In closed-loop mode (DFLLCTRL.MODE=1), this bit group indicates the difference between the ideal number of DFLL cycles and the counted number of cycles. In open-loop mode, this value is not updated and hence, invalid.

Bits 15:10 – COARSE[5:0]: Coarse Value
Set the value of the Coarse Calibration register. In closed-loop mode, this field is read-only.

Bits 9:0 – FINE[9:0]: Fine Value
Set the value of the Fine Calibration register. In closed-loop mode, this field is read-only.

21.8.9. DFLL48M Multiplier

Name: DFLLMUL
Offset: 0x20
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized using STATUS.DFLLRDY=1

Bit	31	30	29	28	27	26	25	24
	CSTEP[5:0]						FSTEP[9:8]	
Access	R	R	R	R	R	R	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	FSTEP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	MUL[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MUL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:26 – CSTEP[5:0]: Coarse Maximum Step
This bit group indicates the maximum step size allowed during coarse adjustment in closed-loop mode. When adjusting to a new frequency, the expected output frequency overshoot depends on this step size.

Bits 25:16 – FSTEP[9:0]: Fine Maximum Step
This bit group indicates the maximum step size allowed during fine adjustment in closed-loop mode. When adjusting to a new frequency, the expected output frequency overshoot depends on this step size.

Bits 15:0 – MUL[15:0]: DFLL Multiply Factor
This field determines the ratio of the CLK_DFLL output frequency to the CLK_DFLL_REF input frequency. Writing to the MUL bits will cause locks to be lost and the fine calibration value to be reset to its midpoint.

21.8.10. DFLL48M Synchronization

Name: DFLLSYNC
Offset: 0x24
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	READREQ							
Access	W							
Reset	0							

Bit 7 – READREQ: Read Request
To be able to read the current value of the DFLLVAL register in closed-loop mode, this bit must be written to '1'.

21.8.11. DPLL Control A

Name: DPLLCTRLA
Offset: 0x28
Reset: 0x80
Property: PAC Write-Protection, Write-Synchronized (ENABLE)

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY					ENABLE	
Access	R/W	R/W					R/W	
Reset	1	0					0	

Bit 7 – ONDEMAND: On Demand Clock Activation
The On Demand operation mode allows the DPLL to be enabled or disabled depending on peripheral clock requests.

If the ONDEMAND bit has been previously written to '1', the DPLL will only be running when requested by a peripheral. If there is no peripheral requesting the DPLL's clock source, the DPLL will be in a disabled state.

If On Demand is disabled the DPLL will always be running when enabled.

In standby sleep mode, the On Demand operation is still active.

Value	Description
0	The DPLL is always on, if enabled.
1	The DPLL is enabled when a peripheral is requesting the DPLL to be used as a clock source. The DPLL is disabled if no peripheral is requesting the clock source.

Bit 6 – RUNSTDBY: Run in Standby
This bit controls how the DPLL behaves during standby sleep mode:

Value	Description
0	The DPLL is disabled in standby sleep mode if no peripheral requests the clock.
1	The DPLL is not stopped in standby sleep mode. If ONDEMAND=1, the DPLL will be running when a peripheral is requesting the clock. If ONDEMAND=0, the clock source will always be running in standby sleep mode.

Bit 1 – ENABLE: DPLL Enable
The software operation of enabling or disabling the DPLL takes a few clock cycles, so the DPLLSYNCBUSY.ENABLE status bit indicates when the DPLL is successfully enabled or disabled.

Value	Description
0	The DPLL is disabled.
1	The DPLL is enabled.

21.8.12. DPLL Ratio Control

Name: DPLLRATIO
Offset: 0x2C
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					LDRFRAC[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
					LDR[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	LDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 19:16 – LDRFRAC[3:0]: Loop Divider Ratio Fractional Part
Writing these bits selects the fractional part of the frequency multiplier. Due to synchronization there is a delay between writing these bits and the effect on the DPLL output clock. The value written will read back immediately and the DPLLRATIO bit in the DPLL Synchronization Busy register (DPLLSYNCBUSY.DPLLRATIO) will be set. DPLLSYNCBUSY.DPLLRATIO will be cleared when the operation is completed.

Bits 11:0 – LDR[11:0]: Loop Divider Ratio
Writing these bits selects the integer part of the frequency multiplier. The value written to these bits will read back immediately, and the DPLLRATIO bit in the DPLL Synchronization busy register (DPLLSYNCBUSY.DPLLRATIO), will be set. DPLLSYNCBUSY.DPLLRATIO will be cleared when the operation is completed.

21.8.13. DPLL Control B

Name: DPLLCTRLB
Offset: 0x30
Reset: 0x00
Property: Enable-Protected, PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
						DIV[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
	DIV[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
				LBYPASS		LTIME[2:0]		
Access				R/W		R/W	R/W	R/W
Reset				0		0	0	0

Bit	7	6	5	4	3	2	1	0
			REFCLK[1:0]		WUF	LPEN	FILTER[1:0]	
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 26:16 – DIV[10:0]: Clock Divider
These bits set the XOSC clock division factor and can be calculated with following formula:

$$f_{DIV} = \frac{f_{XOSC}}{2x(DIV+1)}$$

Bit 12 – LBYPASS: Lock Bypass

Value	Description
0	DPLL Lock signal drives the DPLL controller internal logic.
1	DPLL Lock signal is always asserted.

Bits 10:8 – LTIME[2:0]: Lock Time
These bits select the lock time-out value:

Value	Name	Description
0x0	Default	No time-out. Automatic lock.
0x1	Reserved	
0x2	Reserved	
0x3	Reserved	

Value	Name	Description
0x4	8MS	Time-out if no lock within 8ms
0x5	9MS	Time-out if no lock within 9ms
0x6	10MS	Time-out if no lock within 10ms
0x7	11MS	Time-out if no lock within 11ms

Bits 5:4 – REFCLK[1:0]: Reference Clock Selection
Write these bits to select the DPLL clock reference:

Value	Name	Description
0x0	XOSC32K	XOSC32K clock reference
0x1	XOSC	XOSC clock reference
0x2	GCLK	GCLK clock reference
0x3	Reserved	

Bit 3 – WUF: Wake Up Fast

Value	Description
0	DPLL clock is output after startup and lock time.
1	DPLL clock is output after startup time.

Bit 2 – LPEN: Low-Power Enable

Value	Description
0	The low-power mode is disabled. Time to Digital Converter is enabled.
1	The low-power mode is enabled. Time to Digital Converter is disabled. This will improve power consumption but increase the output jitter.

Bits 1:0 – FILTER[1:0]: Proportional Integral Filter Selection
These bits select the DPLL filter type:

Value	Name	Description
0x0	DEFAULT	Default filter mode
0x1	LBFILT	Low bandwidth filter
0x2	HBFILT	High bandwidth filter
0x3	HDFILT	High damping filter

21.8.14. DPLL Prescaler

Name: DPLLPRESC
Offset: 0x34
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
							PRESC[1:0]	
Access							R/W	R/W
Reset							0	0

Bits 1:0 – PRESC[1:0]: Output Clock Prescaler
These bits define the output clock prescaler setting.

Value	Name	Description
0x0	DIV1	DPLL output is divided by 1
0x1	DIV2	DPLL output is divided by 2
0x2	DIV4	DPLL output is divided by 4
0x3	Reserved	

21.8.15. DPLL Synchronization Busy

Name: DPLLSYNCBUSY
Offset: 0x38
Reset: 0x00
Property: –

Bit	7	6	5	4	3	2	1	0
					DPLLPRESC	DPLLRATIO	ENABLE	
Access					R	R	R	
Reset					0	0	0	

Bit 3 – DPLLPRESC: DPLL Prescaler Synchronization Status

Value	Description
0	The DPLLRESC register has been synchronized.
1	The DPLLRESC register value has changed and its synchronization is in progress.

Bit 2 – DPLLRATIO: DPLL Loop Divider Ratio Synchronization Status

Value	Description
0	The DPLLRATIO register has been synchronized.
1	The DPLLRATIO register value has changed and its synchronization is in progress.

Bit 1 – ENABLE: DPLL Enable Synchronization Status

Value	Description
0	The DPLLCTRLA.ENABLE bit has been synchronized.
1	The DPLLCTRLA.ENABLE bit value has changed and its synchronization is in progress.

21.8.16. DPLL Status

Name: DPLLSTATUS
Offset: 0x3C
Reset: 0x00
Property: –

Bit	7	6	5	4	3	2	1	0
							CLKRDY	LOCK
Access							R	R
Reset							0	0

Bit 1 – CLKRDY: Output Clock Ready

Value	Description
0	The DPLL output clock is off.
1	The DPLL output clock in on.

Bit 0 – LOCK: DPLL Lock status bit

Value	Description
0	The DPLL Lock signal is cleared, when the DPLL is disabled or when the DPLL is trying to reach the target frequency.
1	The DPLL Lock signal is asserted when the desired frequency is reached.

22. OSC32KCTRL – 32KHz Oscillators Controller

22.1. Overview

The 32KHz Oscillators Controller (OSC32KCTRL) provides a user interface to the 32.768kHz oscillators: XOSC32K, OSC32K, and OSCULP32K.

The OSC32KCTRL sub-peripherals can be enabled, disabled, calibrated, and monitored through interface registers.

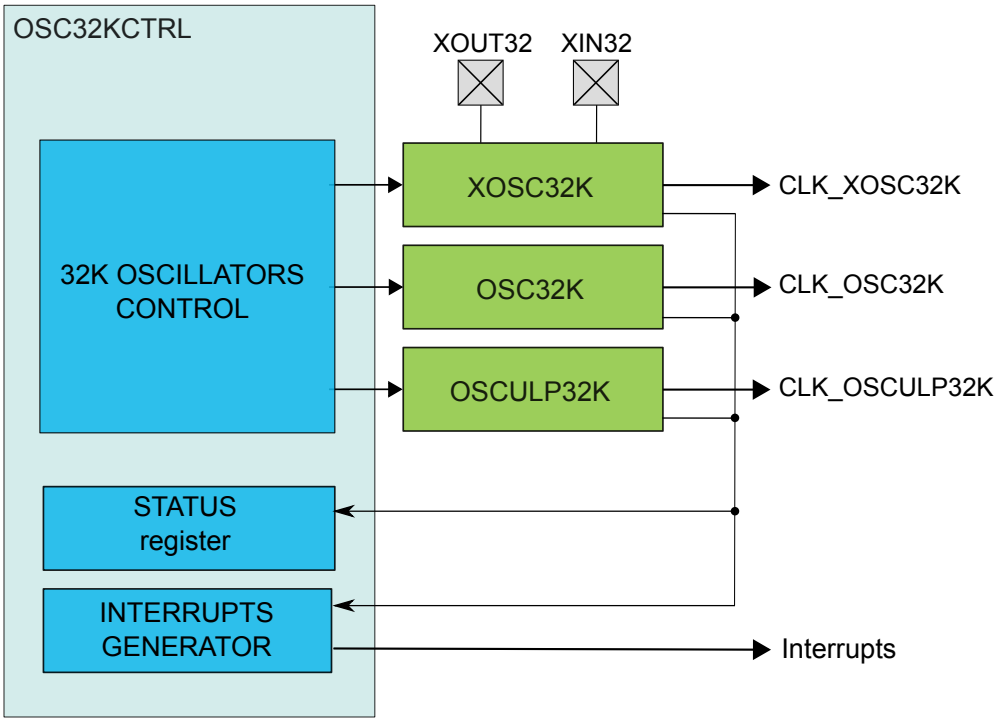
All sub-peripheral statuses are collected in the Status register (STATUS). They can additionally trigger interrupts upon status changes via the INTENSET, INTENCLR, and INTFLAG registers.

22.2. Features

- 32.768kHz Crystal Oscillator (XOSC32K)
  - Programmable start-up time
  - Crystal or external input clock on XIN32 I/O
- 32.768kHz High Accuracy Internal Oscillator (OSC32K)
  - Frequency fine tuning
  - Programmable start-up time
- 32.768kHz Ultra Low Power Internal Oscillator (OSCULP32K)
  - Ultra low power, always-on oscillator
  - Frequency fine tuning
- Calibration value loaded from Flash factory calibration at reset
- 1.024kHz clock outputs available

22.3. Block Diagram

Figure 22-1. OSC32KCTRL Block Diagram



22.4. Signal Description

Signal	Description	Type
XIN32	Analog Input	32.768kHz Crystal Oscillator or external clock generator input
XOUT32	Analog Output	32.768kHz Crystal Oscillator output

The I/O lines are automatically selected when XOSC32K is enabled.

Note: The signal of the external crystal oscillator may affect the jitter of neighboring pads.

22.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

22.5.1. I/O Lines

I/O lines are configured by OSC32KCTRL when XOSC32K is enabled, and need no user configuration.

22.5.2. Power Management

The OSC32KCTRL will continue to operate in any sleep mode where a 32KHz oscillator is running as source clock. The OSC32KCTRL interrupts can be used to wake up the device from sleep modes.

Related Links

PM – Power Manager on page 186

22.5.3. Clocks

The OSC32KCTRL gathers controls for all 32KHz oscillators and provides clock sources to the Generic Clock Controller (GCLK), Real-Time Counter (RTC), and Watchdog Timer (WDT).

The available clock sources are: XOSC32K, OSC32K, and OSCULP32K.

The OSC32KCTRL bus clock (CLK_OSC32KCTRL_APB) can be enabled and disabled in the Main Clock module (MCLK).

Related Links

Peripheral Clock Masking on page 151

22.5.4. Interrupts

The interrupt request lines are connected to the interrupt controller. Using the OSC32KCTRL interrupts requires the interrupt controller to be configured first.

Related Links

Nested Vector Interrupt Controller on page 51

22.5.5. Debug Operation

When the CPU is halted in debug mode, OSC32KCTRL will continue normal operation. If OSC32KCTRL is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

22.5.6. Register Access Protection

All registers with write-access can be write-protected optionally by the Peripheral Access Controller (PAC), except the following registers:

- Interrupt Flag Status and Clear (INTFLAG) register

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

PAC write-protection does not apply to accesses through an external debugger.

Related Links

PAC - Peripheral Access Controller on page 58

22.5.7. Analog Connections

The external 32.768kHz crystal must be connected between the XIN32 and XOUT32 pins, along with any required load capacitors. For details on recommended oscillator characteristics and capacitor load, refer to the related links.

Related Links

Electrical Characteristics on page 1140

22.5.8. Calibration

The OSC32K calibration value from the production test must be loaded from the NVM Software Calibration Area into the OSC32K register (OSC32K.CALIB) by software to achieve specified accuracy.

22.6. Functional Description

22.6.1. Principle of Operation

XOSC32K, OSC32K, and OSCULP32K are configured via OSC32KCTRL control registers. Through this interface, the sub-peripherals are enabled, disabled, or have their calibration values updated.

The STATUS register gathers different status signals coming from the sub-peripherals of OSC32KCTRL. The status signals can be used to generate system interrupts, and in some cases wake up the system from standby mode, provided the corresponding interrupt is enabled.

22.6.2. 32KHz External Crystal Oscillator (XOSC32K) Operation

The XOSC32K can operate in two different modes:

- External clock, with an external clock signal connected to XIN32
- Crystal oscillator, with an external 32.768kHz crystal connected between XIN32 and XOUT32

At reset, the XOSC32K is disabled, and the XIN32/XOUT32 pins can either be used as General Purpose I/O (GPIO) pins or by other peripherals in the system.

When XOSC32K is enabled, the operating mode determines the GPIO usage. When in crystal oscillator mode, the XIN32 and XOUT32 pins are controlled by the OSC32KCTRL, and GPIO functions are overridden on both pins. When in external clock mode, the only XIN32 pin will be overridden and controlled by the OSC32KCTRL, while the XOUT32 pin can still be used as a GPIO pin.

The XOSC32K is enabled by writing a '1' to the Enable bit in the 32KHz External Crystal Oscillator Control register (XOSC32K.ENABLE=1). The XOSC32K is disabled by writing a '0' to the Enable bit in the 32KHz External Crystal Oscillator Control register (XOSC32K.ENABLE=0).

To enable the XOSC32K as a crystal oscillator, the XTALEN bit in the 32KHz External Crystal Oscillator Control register must be set (XOSC32K.XTALEN=1). If XOSC32K.XTALEN is '0', the external clock input will be enabled.

The XOSC32K 32.768kHz output is enabled by setting the 32KHz Output Enable bit in the 32KHz External Crystal Oscillator Control register (XOSC32K.EN32K=1). The XOSC32K also has a 1.024kHz clock output. This is enabled by setting the 1KHz Output Enable bit in the 32KHz External Crystal Oscillator Control register (XOSC32K.EN1K=1).

It is also possible to lock the XOSC32K configuration by setting the Write Lock bit in the 32KHz External Crystal Oscillator Control register (XOSC32K.WRTLOCK=1). If set, the XOSC32K configuration is locked until a Power-On Reset (POR) is detected.

The XOSC32K will behave differently in different sleep modes based on the settings of XOSC32K.RUNSTDBY, XOSC32K.ONDEMAND, and XOSC32K.ENABLE. If XOSC32KCTRL.ENABLE=0, the XOSC32K will be always stopped. For XOS32KCTRL.ENABLE=1, this table is valid:

Table 22-1. XOSC32K Sleep Behavior

CPU Mode	XOSC32KCTRL.RUNSTDBY	XOSC32KCTRL.ONDEMAND	Sleep Behavior
Active or Idle	-	0	Always run
Active or Idle	-	1	Run if requested by peripheral
Standby	1	0	Always run

CPU Mode	XOSC32KCTRL.RUNSTDBY	XOSC32KCTRL.ONDEMAND	Sleep Behavior
Standby	1	1	Run if requested by peripheral
Standby	0	-	Run if requested by peripheral

As a crystal oscillator usually requires a very long start-up time, the 32KHz External Crystal Oscillator will keep running across resets when XOSC32K.ONDEMAND=0, except for power-on reset (POR). After a reset or when waking up from a sleep mode where the XOSC32K was disabled, the XOSC32K will need a certain amount of time to stabilize on the correct frequency. This start-up time can be configured by changing the Oscillator Start-Up Time bit group (XOSC32K.STARTUP) in the 32KHz External Crystal Oscillator Control register. During the start-up time, the oscillator output is masked to ensure that no unstable clock propagates to the digital logic.

Once the external clock or crystal oscillator is stable and ready to be used as a clock source, the XOSC32K Ready bit in the Status register is set (STATUS.XOSC32KRDY=1). The transition of STATUS.XOSC32KRDY from '0' to '1' generates an interrupt if the XOSC32K Ready bit in the Interrupt Enable Set register is set (INTENSET.XOSC32KRDY=1).

The XOSC32K can be used as a source for Generic Clock Generators (GCLK) or for the Real-Time Counter (RTC). Before enabling the GCLK or the RTC module, the corresponding oscillator output must be enabled (XOSC32K.EN32K or XOSC32K.EN1K) in order to ensure proper operation. In the same way, the GCLK or RTC modules must be disabled before the clock selection is changed. For details on RTC clock configuration, refer also to Real-Time Counter Clock Selection on page 270.

Related Links

GCLK - Generic Clock Controller on page 131

RTC – Real-Time Counter on page 341

22.6.3. 32KHz Internal Oscillator (OSC32K) Operation

The OSC32K provides a tunable, low-speed, and low-power clock source.

At reset, the OSC32K is disabled. It can be enabled by setting the Enable bit in the 32KHz Internal Oscillator Control register (OSC32K.ENABLE=1). The OSC32K is disabled by clearing the Enable bit in the 32KHz Internal Oscillator Control register (OSC32K.ENABLE=0).

The frequency of the OSC32K oscillator is controlled by OSC32K.CALIB, which is a calibration value in the 32KHz Internal Oscillator Calibration bits in the 32KHz Internal Oscillator Control register. The CALIB value must be must be loaded with production calibration values from the NVM Software Calibration Area. When writing the Calibration bits, the user must wait for the STATUS.OSC32KRDY bit to go high before the new value is committed to the oscillator.

The OSC32K has a 32.768kHz output which is enabled by setting the 32KHz Output Enable bit in the 32KHz Internal Oscillator Control register (OSC32K.EN32K=1). The OSC32K also has a 1.024kHz clock output. This is enabled by setting the 1KHz Output Enable bit in the 32KHz Internal Oscillator Control register (OSC32K.EN1K).

Before using the USB, the Pad Calibration register (PADCAL) must be loaded with production calibration values from the NVM Software Calibration Area.

The OSC32K will behave differently in different sleep modes based on the settings of OSC32K.RUNSTDBY, OSC32K.ONDEMAND, and OSC32K.ENABLE. If OSC32KCTRL.ENABLE=0, the OSC32K will be always stopped. For OS32KCTRL.ENABLE=1, this table is valid:

Table 22-2. OSC32K Sleep Behavior

CPU Mode	OSC32KCTRL.RUNSTDBY	OSC32KCTRL.ONDEMAND	Sleep Behavior
Active or Idle	-	0	Always run
Active or Idle	-	1	Run if requested by peripheral
Standby	1	0	Always run
Standby	1	1	Run if requested by peripheral
Standby	0	-	Run if requested by peripheral

The OSC32K requires a start-up time. For this reason, OSC32K will keep running across resets when OSC32K.ONDEMAND=0, except for power-on reset (POR).

After such a reset, or when waking up from a sleep mode where the OSC32K was disabled, the OSC32K will need a certain amount of time to stabilize on the correct frequency.

This startup time can be configured by changing the Oscillator Start-Up Time bit group (OSC32K.STARTUP) in the OSC32K Control register. During the start-up time, the oscillator output is masked to ensure that no unstable clock propagates to the digital logic.

Once the external clock or crystal oscillator is stable and ready to be used as a clock source, the OSC32K Ready bit in the Status register is set (STATUS.OSC32KRDY=1). The transition of STATUS.OSC32KRDY from '0' to '1' generates an interrupt if the OSC32K Ready bit in the Interrupt Enable Set register is set (INTENSET.OSC32KRDY=1).

The OSC32K can be used as a source for Generic Clock Generators (GCLK) or for the Real-Time Counter (RTC). Before enabling the GCLK or the RTC module, the corresponding oscillator output must be enabled (OSC32K.EN32K or OSC32K.EN1K) in order to ensure proper operation. In the same way, the GCLK or RTC modules must be disabled before the clock selection is changed.

Related Links

RTC – Real-Time Counter on page 341
Real-Time Counter Clock Selection on page 270
Electrical Characteristics on page 1140

22.6.4. 32KHz Ultra Low Power Internal Oscillator (OSCULP32K) Operation

The OSCULP32K provides a tunable, low-speed, and ultra-low-power clock source. The OSCULP32K is factory-calibrated under typical voltage and temperature conditions. The OSCULP32K should be preferred to the OSC32K whenever the power requirements are prevalent over frequency stability and accuracy.

The OSCULP32K is enabled by default after a power-on reset (POR) and will always run except during POR. The frequency of the OSCULP32K oscillator is controlled by the value in the 32KHz Ultra Low Power Internal Oscillator Calibration bits in the 32KHz Ultra Low Power Internal Oscillator Control register (OSCULP32K.CALIB). This data is used to compensate for process variations.

OSCULP32K.CALIB is automatically loaded from Flash Factory Calibration during start-up. The calibration value can be overridden by the user by writing to OSCULP32K.CALIB.

It is also possible to lock the OSCULP32K configuration by setting the Write Lock bit in the 32KHz Ultra Low Power Internal Oscillator Control register (OSCULP32K.WRTLOCK=1). If set, the OSCULP32K configuration is locked until a power-on reset (POR) is detected.

The OSCULP32K can be used as a source for Generic Clock Generators (GCLK) or for the Real-Time Counter (RTC). To ensure proper operation, the GCLK or RTC modules must be disabled before the clock selection is changed.

Related Links

RTC – Real-Time Counter on page 341

Real-Time Counter Clock Selection on page 270

GCLK - Generic Clock Controller on page 131

22.6.5. Watchdog Timer Clock Selection

The Watchdog Timer (WDT) uses the internal 1.024kHz OSCULP32K output clock. This clock is running all the time and internally enabled when requested by the WDT module.

Related Links

WDT – Watchdog Timer on page 321

22.6.6. Real-Time Counter Clock Selection

Before enabling the RTC module, the RTC clock must be selected first. All oscillator outputs are valid as RTC clock. The selection is done in the RTC Control register (RTCCTRL). To ensure a proper operation, it is highly recommended to disable the RTC module first, before the RTC clock source selection is changed.

Related Links

RTC – Real-Time Counter on page 341

22.6.7. Interrupts

The OSC32KCTRL has the following interrupt sources:

- XOSC32KRDY - 32KHz Crystal Oscillator Ready: A 0-to-1 transition on the STATUS.XOSC32KRDY bit is detected
- OSC32KRDY - 32KHz Internal Oscillator Ready: A 0-to-1 transition on the STATUS.OSC32KRDY bit is detected

All these interrupts are synchronous wake-up source.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) is set when the interrupt condition occurs. Each interrupt can be enabled individually by setting the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by setting the corresponding bit in the Interrupt Enable Clear register (INTENCLR). An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled or the OSC32KCTRL is reset. See the INTFLAG register for details on how to clear interrupt flags.

The OSC32KCTRL has one common interrupt request line for all the interrupt sources. The user must read the INTFLAG register to determine which interrupt condition is present. Refer to the INTFLAG register for details.

Note: Interrupts must be globally enabled for interrupt requests to be generated.

Related Links

PM – Power Manager on page 186

Nested Vector Interrupt Controller on page 51

22.7. Register Summary

Offset	Name	Bit Pos.								
0x00	INTENCLR	7:0							OSC32KRDY	XOSC32KRDY
0x01		15:8								
0x02		23:16								
0x03		31:24								
0x04	INTENSET	7:0							OSC32KRDY	XOSC32KRDY
0x05		15:8								
0x06		23:16								
0x07		31:24								
0x08	INTFLAG	7:0							OSC32KRDY	XOSC32KRDY
0x09		15:8								
0x0A		23:16								
0x0B		31:24								
0x0C	STATUS	7:0							OSC32KRDY	XOSC32KRDY
0x0D		15:8								
0x0E		23:16								
0x0F		31:24								
0x10	RTCCTRL	7:0						RTCSEL[2:0]		
0x11		15:8								
0x12		23:16								
0x13		31:24								
0x14	XOSC32K	7:0	ONDEMAND	RUNSTDBY		EN1K	EN32K	XTALEN	ENABLE	
0x15		15:8				WRTLOCK		STARTUP[2:0]		
0x16		23:16								
0x17		31:24								
0x18	OSC32K	7:0	ONDEMAND	RUNSTDBY			EN1K	EN32K	ENABLE	
0x19		15:8				WRTLOCK		STARTUP[2:0]		
0x1A		23:16		CALIB[6:0]						
0x1B		31:24								
0x1C	OSCULP32K	7:0								
0x1D		15:8	WRTLOCK			CALIB[4:0]				

22.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register and the 8-bit halves of a 16-bit register can be accessed directly.

All registers with write-access can be write-protected optionally by the peripheral access controller (PAC). Optional Write-Protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in the register description. Write-protection does not apply to accesses through an external debugger.

Related Links

PAC - Peripheral Access Controller on page 58

22.8.1. Interrupt Enable Clear

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Name: INTENCLR
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							OSC32KRDY	XOSC32KRDY
Access							R/W	R/W
Reset							0	0

Bit 1 – OSC32KRDY: OSC32K Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the OSC32K Ready Interrupt Enable bit, which disables the OSC32K Ready interrupt.

Value	Description
0	The OSC32K Ready interrupt is disabled.
1	The OSC32K Ready interrupt is enabled.

Bit 0 – XOSC32KRDY: XOSC32K Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the XOSC32K Ready Interrupt Enable bit, which disables the XOSC32K Ready interrupt.

Value	Description
0	The XOSC32K Ready interrupt is disabled.
1	The XOSC32K Ready interrupt is enabled.

22.8.2. Interrupt Enable Set

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Name: INTENSET
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							OSC32KRDY	XOSC32KRDY
Access							R/W	R/W
Reset							0	0

Bit 1 – OSC32KRDY: OSC32K Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the OSC32K Ready Interrupt Enable bit, which enables the OSC32K Ready interrupt.

Value	Description
0	The OSC32K Ready interrupt is disabled.
1	The OSC32K Ready interrupt is enabled.

Bit 0 – XOSC32KRDY: XOSC32K Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the XOSC32K Ready Interrupt Enable bit, which enables the XOSC32K Ready interrupt.

Value	Description
0	The XOSC32K Ready interrupt is disabled.
1	The XOSC32K Ready interrupt is enabled.

22.8.3. Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x08
Reset: 0x00000000
Property: –

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							OSC32KRDY	XOSC32KRDY
Access							R/W	R/W
Reset							0	0

Bit 1 – OSC32KRDY: OSC32K Ready
This flag is cleared by writing a '1' to it.

This flag is set by a zero-to-one transition of the OSC32K Ready bit in the Status register (STATUS.OSC32KRDY), and will generate an interrupt request if INTENSET.OSC32KRDY=1.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the OSC32K Ready interrupt flag.

Bit 0 – XOSC32KRDY: XOSC32K Ready
This flag is cleared by writing a '1' to it.

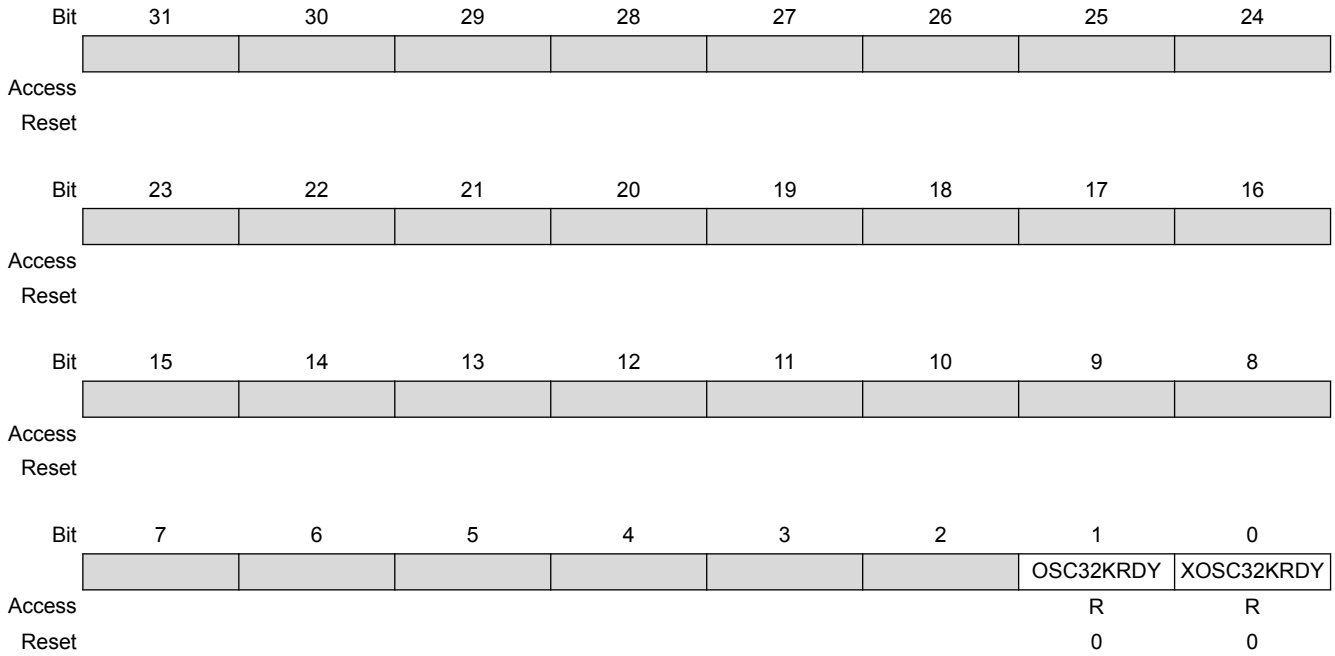
This flag is set by a zero-to-one transition of the XOSC32K Ready bit in the Status register (STATUS.XOSC32KRDY), and will generate an interrupt request if INTENSET.XOSC32KRDY=1.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the XOSC32K Ready interrupt flag.

22.8.4. Status

Name: STATUS
Offset: 0x0C
Reset: 0x00000000
Property: –



Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							OSC32KRDY	XOSC32KRDY
Access							R	R
Reset							0	0

Bit 1 – OSC32KRDY: OSC32K Ready

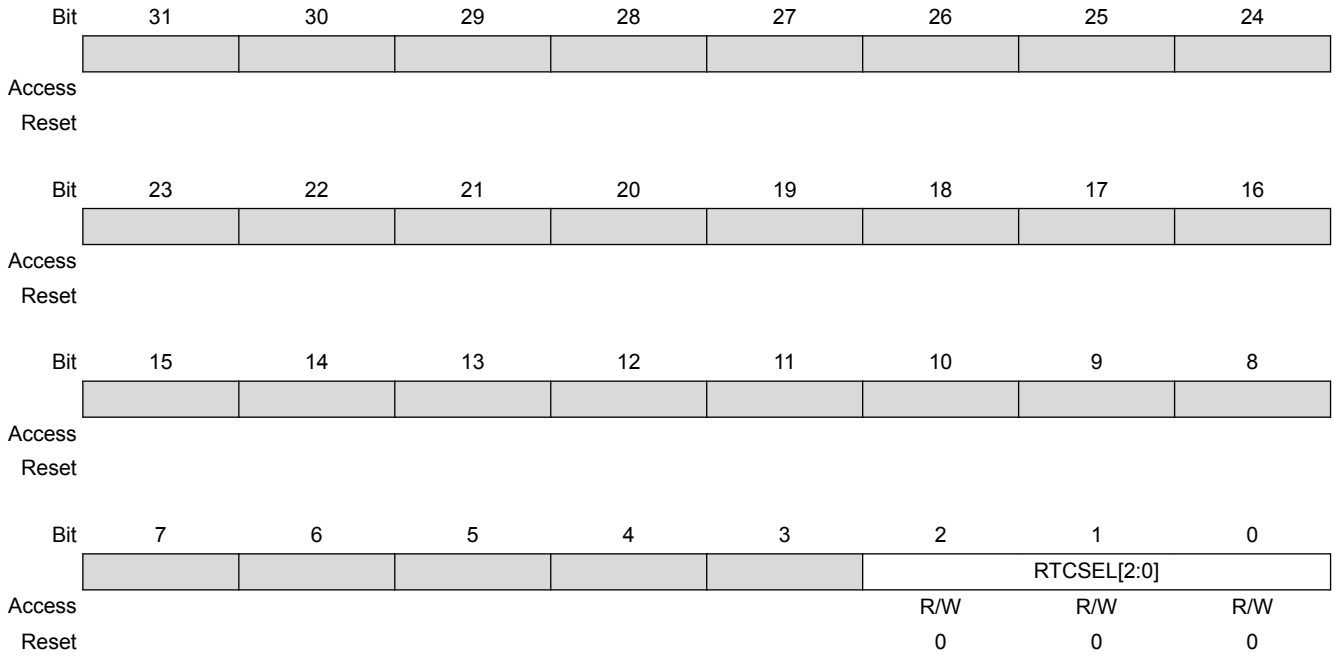
Value	Description
0	OSC32K is not ready.
1	OSC32K is stable and ready to be used as a clock source.

Bit 0 – XOSC32KRDY: XOSC32K Ready

Value	Description
0	XOSC32K is not ready.
1	XOSC32K is stable and ready to be used as a clock source.

22.8.5. RTC Clock Selection Control

Name: RTCCTRL
Offset: 0x10
Reset: 0x00000000
Property: PAC Write-Protection



Bits 2:0 – RTCSEL[2:0]: RTC Clock Source Selection
These bits select the source for the RTC.

Value	Name	Description
0x0	ULP1K	1.024kHz from 32KHz internal ULP oscillator
0x1	ULP32K	32.768kHz from 32KHz internal ULP oscillator
0x2	OSC1K	1.024kHz from 32KHz internal oscillator
0x3	OSC32K	32.768kHz from 32KHz internal oscillator
0x4	XOSC1K	1.024kHz from 32KHz external oscillator
0x5	XOSC32K	32.768kHz from 32KHz external crystal oscillator
0x6	Reserved	
0x7	Reserved	

22.8.6. 32KHz External Crystal Oscillator (XOSC32K) Control

Name: XOSC32K
Offset: 0x14
Reset: 0x00000080
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				WRTLOCK		STARTUP[2:0]		
Access				R/W		R/W	R/W	R/W
Reset				0		0	0	0

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY		EN1K	EN32K	XTALEN	ENABLE	
Access	R/W	R/W		R/W	R/W	R/W	R/W	
Reset	1	0		0	0	0	0	

Bit 12 – WRTLOCK: Write Lock
This bit locks the XOSC32K register for future writes, effectively freezing the XOSC32K configuration.

Value	Description
0	The XOSC32K configuration is not locked.
1	The XOSC32K configuration is locked.

Bits 10:8 – STARTUP[2:0]: Oscillator Start-Up Time
These bits select the start-up time for the oscillator.

The OSCULP32K oscillator is used to clock the start-up counter.

Table 22-3. Start-Up Time for 32KHz External Crystal Oscillator

STARTUP[2:0]	Number of OSCULP32K Clock Cycles	Number of XOSC32K Clock Cycles	Approximate Equivalent Time [s]
0x0	2048	3	0.06
0x1	4096	3	0.13
0x2	16384	3	0.5
0x3	32768	3	1

STARTUP[2:0]	Number of OSCULP32K Clock Cycles	Number of XOSC32K Clock Cycles	Approximate Equivalent Time [s]
0x4	65536	3	2
0x5	131072	3	4
0x6	262144	3	8
0x7	-	-	Reserved

Note:

1. Actual Start-Up time is 1 OSCULP32K cycle + 3 XOSC32K cycles.
2. The given time assumes an XTAL frequency of 32.768kHz.

Bit 7 – ONDEMAND: On Demand Control
This bit controls how the XOSC32K behaves when a peripheral clock request is detected. For details, refer to XOSC32K Sleep Behavior.

Bit 6 – RUNSTDBY: Run in Standby
This bit controls how the XOSC32K behaves during standby sleep mode. For details, refer to XOSC32K Sleep Behavior.

Bit 4 – EN1K: 1KHz Output Enable

Value	Description
0	The 1KHz output is disabled.
1	The 1KHz output is enabled.

Bit 3 – EN32K: 32KHz Output Enable

Value	Description
0	The 32KHz output is disabled.
1	The 32KHz output is enabled.

Bit 2 – XTALEN: Crystal Oscillator Enable
This bit controls the connections between the I/O pads and the external clock or crystal oscillator.

Value	Description
0	External clock connected on XIN32. XOUT32 can be used as general-purpose I/O.
1	Crystal connected to XIN32/XOUT32.

Bit 1 – ENABLE: Oscillator Enable

Value	Description
0	The oscillator is disabled.
1	The oscillator is enabled.

22.8.7. 32KHz Internal Oscillator (OSC32K) Control

Name: OSC32K
Offset: 0x18
Reset: 0x0000 0080 (Writing action by User required)
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
		CALIB[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		-	-	-	-	-	-	-

Bit	15	14	13	12	11	10	9	8
				WRTLOCK		STARTUP[2:0]		
Access				R/W		R/W	R/W	R/W
Reset				0		0	0	0

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY			EN1K	EN32K	ENABLE	
Access	R/W	R/W			R/W	R/W	R/W	
Reset	1	0			0	0	0	

Bits 22:16 – CALIB[6:0]: Oscillator Calibration
These bits control the oscillator calibration. The calibration values must be loaded by the user from the NVM Software Calibration Area.

Bit 12 – WRTLOCK: Write Lock
This bit locks the OSC32K register for future writes, effectively freezing the OSC32K configuration.

Value	Description
0	The OSC32K configuration is not locked.
1	The OSC32K configuration is locked.

Bits 10:8 – STARTUP[2:0]: Oscillator Start-Up Time
These bits select start-up time for the oscillator.

The OSCULP32K oscillator is used as input clock to the start-up counter.

Table 22-4. Start-Up Time for 32KHz Internal Oscillator

STARTUP[2:0]	Number of OSC32K clock cycles	Approximate Equivalent Time [ms]
0x0	3	0.092
0x1	4	0.122
0x2	6	0.183

STARTUP[2:0]	Number of OSC32K clock cycles	Approximate Equivalent Time [ms]
0x3	10	0.305
0x4	18	0.549
0x5	34	1.038
0x6	66	2.014
0x7	130	3.967

Note:

1. Start-up time is given by STARTUP + three OSC32K cycles.
2. The given time assumes an XTAL frequency of 32.768kHz.

Bit 7 – ONDEMAND: On Demand Control
This bit controls how the OSC32K behaves when a peripheral clock request is detected. For details, refer to OSC32K Sleep Behavior.

Bit 6 – RUNSTDBY: Run in Standby
This bit controls how the OSC32K behaves during standby sleep mode. For details, refer to OSC32K Sleep Behavior.

Bit 3 – EN1K: 1KHz Output Enable

Value	Description
0	The 1KHz output is disabled.
1	The 1KHz output is enabled.

Bit 2 – EN32K: 32KHz Output Enable

Value	Description
0	The 32KHz output is disabled.
1	The 32KHz output is enabled.

Bit 1 – ENABLE: Oscillator Enable

Value	Description
0	The oscillator is disabled.
1	The oscillator is enabled.

22.8.8. 32KHz Ultra Low Power Internal Oscillator (OSCULP32K) Control

Name: OSCULP32K
Offset: 0x1C
Reset: 0x0000XX06
Property: Write-Protected

Bit	15	14	13	12	11	10	9	8
	WRTLOCK			CALIB[4:0]				
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			0	0	0	0	x

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 15 – WRTLOCK: Write Lock
This bit locks the OSCULP32K register for future writes to fix the OSCULP32K configuration.

Value	Description
0	The OSCULP32K configuration is not locked.
1	The OSCULP32K configuration is locked.

Bits 12:8 – CALIB[4:0]: Oscillator Calibration
These bits control the oscillator calibration.

These bits are loaded from Flash Calibration at startup.

23. SUPC – Supply Controller

23.1. Overview

The Supply Controller (SUPC) manages the voltage reference, power supply, and supply monitoring of the device. It is also able to control two output pins.

The SUPC controls the voltage regulators for the core (VDDCORE) and backup (VDDBU) domains. It sets the voltage regulators according to the sleep modes, or the user configuration. In active mode, the voltage regulators can be selected on the fly between LDO (low-dropout) type regulator or Buck converter.

The SUPC supports connection of a battery backup to the VBAT power pin. It includes functionality that enables automatic power switching between main power and battery backup power. This ensures power to the backup domain when the main battery or power source is unavailable.

The SUPC embeds two Brown-Out Detectors. BOD33 monitors the voltage applied to the device (VDD or VBAT) and BOD12 monitors the internal voltage to the core (VDDCORE). The BOD can monitor the supply voltage continuously (continuous mode) or periodically (sampling mode).

The SUPC generates also a selectable reference voltage and a voltage dependent on the temperature which can be used by analog modules like the ADC.

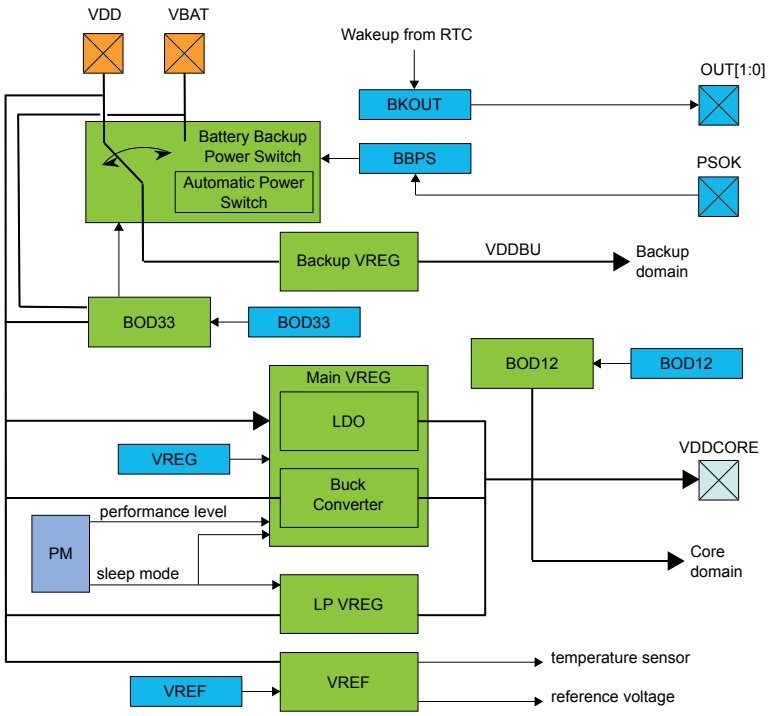
23.2. Features

- Voltage Regulator System
  - Main voltage regulator: LDO or Buck Converter in active mode (MAINVREG)
  - Low Power voltage regulator in standby mode (LPVREG)
  - Backup voltage regulator for backup domains
  - Controlled VDDCORE voltage slope when changing VDDCORE
- Battery Backup Power Switch
  - Automatic switching from main power to battery backup power
    - Automatic entry to backup mode when switched to battery backup power
  - Automatic switching from battery backup power to main power
    - Automatic exit from backup mode when switched back to main power
    - Stay in backup mode when switched back to main power
  - Main power request upon wake-up sources from backup mode
- Voltage Reference System
  - Reference voltage for ADC and DAC
  - Temperature sensor
- 3.3V Brown-Out Detector (BOD33)
  - Programmable threshold
  - Threshold value loaded from NVM User Row at startup
  - Triggers resets, interrupts, or Battery Backup Power Switch. Action loaded from NVM User Row
  - Operating modes:
    - Continuous mode

        - Sampled mode for low power applications with programmable sample frequency
    - Hysteresis value from Flash User Calibration
    - Monitor VDD or VBAT
- 1.2V Brown-Out Detector (BOD12)
    - Programmable threshold
    - Threshold value loaded from Flash User Calibration at startup
    - Triggers resets or interrupts
    - Operating modes:
        - Continuous mode
        - Sampled mode for low power applications with programmable sample frequency
    - Hysteresis value loaded from NVM User Row at startup
- Output pins
    - Pin toggling on RTC event

23.3. Block Diagram

Figure 23-1. SUPC Block Diagram



23.4. Signal Description

Signal Name	Type	Description
OUT[1:0]	Digital Output	SUPC Outputs
PSOK	Digital Input	Main Power Supply OK

One signal can be mapped on several pins.

Related Links

I/O Multiplexing and Considerations on page 29

23.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

23.5.1. I/O Lines

I/O lines are configured by SUPC either when the SUPC output (signal OUT) is enabled or when the PSOK input is enabled. The I/O lines need no user configuration.

23.5.2. Power Management

The SUPC can operate in all sleep modes except backup sleep mode. BOD33 and Battery backup Power Switch can operate in backup mode.

Related Links

PM – Power Manager on page 186

23.5.3. Clocks

The SUPC bus clock (CLK_SUPC_APB) can be enabled and disabled in the Main Clock module.

A 32KHz clock, asynchronous to the user interface clock (CLK_SUPC_APB), is required to run BOD33 and BOD12 in sampled mode. Due to this asynchronicity, writing to certain registers will require synchronization between the clock domains. Refer to Synchronization on page 293 for further details.

Related Links

OSC32KCTRL – 32KHz Oscillators Controller on page 264

Peripheral Clock Masking on page 151

23.5.4. DMA

Not applicable.

23.5.5. Interrupts

The interrupt request lines are connected to the interrupt controller. Using the SUPC interrupts requires the interrupt controller to be configured first.

Related Links

Nested Vector Interrupt Controller on page 51

23.5.6. Events

Not applicable.

23.5.7. Debug Operation

When the CPU is halted in debug mode, the SUPC continues normal operation. If the SUPC is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

If a debugger connection is detected by the system, BOD33 and BOD12 resets will be blocked.

23.5.8. Register Access Protection

Registers with write-access can be write-protected optionally by the peripheral access controller (PAC).

Note: Not all registers with write-access can be write-protected.

PAC Write-Protection is not available for the following registers:

- Interrupt Flag Status and Clear register (INTFLAG)

Optional PAC Write-Protection is denoted by the "PAC Write-Protection" property in each individual register description.

Related Links

PAC - Peripheral Access Controller on page 58

23.5.9. Analog Connections

Not applicable.

23.6. Functional Description

23.6.1. Voltage Regulator System Operation

23.6.1.1. Enabling, Disabling, and Resetting

The LDO main voltage regulator is enabled after any Reset. The main voltage regulator (MAINVREG) can be disabled by writing the Enable bit in the VREG register (VREG.ENABLE) to zero. The main voltage regulator output supply level is automatically defined by the performance level or the sleep mode selected in the Power Manager module.

Related Links

PM – Power Manager on page 186

23.6.1.2. Initialization

After a Reset, the LDO voltage regulator supplying VDDCORE is enabled.

23.6.1.3. Selecting a Voltage Regulator

In active mode, the type of the main voltage regulator supplying VDDCORE can be switched on the fly. The two alternatives are a LDO regulator and a Buck converter.

The main voltage regulator switching sequence:

- The user changes the value of the Voltage Regulator Selection bit in the Voltage Regulator System Control register (VREG.SEL)
- The start of the switching sequence is indicated by clearing the Voltage Regulator Ready bit in the STATUS register (STATUS.VREGRDY=0)
- Once the switching sequence is completed, STATUS.VREGRDY will read '1'

The Voltage Regulator Ready (VREGRDY) interrupt can also be used to detect a zero-to-one transition of the STATUS.VREGRDY bit.

23.6.1.4. Voltage Scaling Control

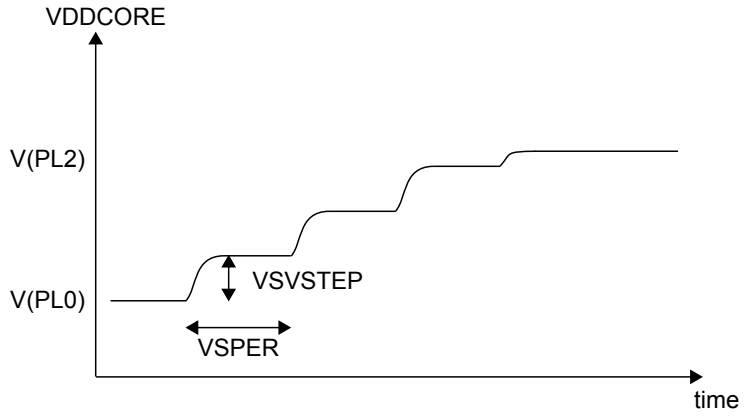
The VDDCORE supply will change under certain circumstances:

- When a new performance level (PL) is set
- When the standby sleep mode is entered or left
- When a sleepwalking task is requested in standby sleep mode

To prevent high peak current on the main power supply and to have a smooth transition of VDDCORE, both the voltage scaling step size and the voltage scaling frequency can be controlled: VDDCORE is changed by the selected step size of the selected period until the target voltage is reached.

The Voltage Scaling Voltage Step field is in the VREG register, VREG.VSVSTEP. The Voltage Scaling Period field is VREG.VSPER.

The following waveform shows an example of changing performance level from PL0 to PL2.



Setting VREG.VSVSTEP to the maximum value allows to transition in one voltage step.

The STATUS.VCORERDY bit is set to '1' as soon as the VDDCORE voltage has reached the target voltage. During voltage transition, STATUS.VCORERDY will read '0'. The Voltage Ready interrupt (VCORERDY) can be used to detect a 0-to-1 transition of STATUS.VCORERDY, see also Interrupts on page 293.

When entering the standby sleep mode and when no sleepwalking task is requested, the VDDCORE Voltage scaling control is not used.

23.6.1.5. Sleep Mode Operation

In standby mode, the low power voltage regulator (LPVREG) is used to supply VDDCORE.

When the Run in Standby bit in the VREG register (VREG.RUNSTDBY) is written to '1', VDDCORE is supplied by the main voltage regulator. Depending on the Standby in PL0 bit in the Voltage Regulator register (VREG.STDBYPL0), the VDDCORE level is either set to the PL0 voltage level, or remains in the current performance level.

Table 23-1. VDDCORE Level in Standby Mode

VREG.RUNSTDBY	VREG.STDBYPL0	VDDCORE Supply in Standby Mode
0	-	LPVREG
1	0	MAINVREG in current performance level(1)
1	1	MAINVREG in PL0

Note:

1. When the device is in PL0 but VREG.STDBYPL0=0, the MAINVREG is operating in normal power mode. To minimize power consumption, operate MAINVREG in PL0 mode by selecting VREG.STDBYPL0=1.

Related Links

Sleep Mode Controller on page 191

23.6.2. Voltage Reference System Operation

23.6.2.1. Initialization

The voltage reference output and the temperature sensor are disabled after any Reset.

23.6.2.2. Enabling, Disabling, and Resetting

The voltage reference output is enabled/disabled by setting/clearing the Voltage Reference Output Enable bit in the Voltage Reference register (VREF.VREFOE).

The temperature sensor is enabled/disabled by setting/clearing the Temperature Sensor Enable bit in the Voltage Reference register (VREF.TSEN).

Note: When VREF.ONDEMAND=0, it is not recommended to enable both voltage reference output and temperature sensor at the same time - only the voltage reference output will be present at both ADC inputs.

23.6.2.3. Selecting a Voltage Reference

The SEL bit group in the VREF register (VREF.SEL) selects the reference voltage to be applied to analog modules, e.g. the ADC.

23.6.2.4. Sleep Mode Operation

The Voltage Reference output and the Temperature Sensor output during sleep mode can be configured using the Run in Standby bit and the On Demand bit in the Voltage Reference register (VREF.RUNSTDBY, VREF.ONDEMAND), see the following table:

Table 23-2. VREF Sleep Mode Operation

VREF.ONDEMAND	VREF.RUNSTDBY	Voltage Reference Sleep behavior
-	-	Disable
0	0	Always run in all sleep modes *except* standby sleep mode
0	1	Always run in all sleep modes *including* standby sleep mode
1	0	Only run if requested by the ADC, in all sleep modes *except* standby sleep mode
1	1	Only run if requested by the ADC, in all sleep modes *including* standby sleep mode

23.6.3. Battery Backup Power Switch

23.6.3.1. Initialization

The Battery Backup Power Switch (BBPS) is disabled at power-up, and the backup domain is supplied by main power.

23.6.3.2. Forced Battery Backup Power Switch

The Backup domain is always supplied by the VBAT supply pin when the Configuration bit field in the Battery Backup Power Switch Control register (BBPS.CONF) is written to 0x2 (FORCED).

23.6.3.3. Automatic Battery Backup Power Switch

The supply of the backup domain can be switched automatically to VBAT supply pin by the Automatic Power Switch or by using the BOD33.

The supply of the backup domain can be switched automatically to VDD supply pin either by the Automatic Power Switch or the Main Power Pin when VDD and VDDCORE are restored.

Automatic Power Switch (APWS)

When the Configuration bit field in the Battery Backup Power Switch register (BBPS.CONF) is selecting the APWS, the Automatic Power Switch will function as Battery Backup Power Switch.

The Automatic Power switch allows to switch the supply of the backup domain from VDD to VBAT power and vice-versa.

When the Automatic Power Switch configuration is selected, the Automatic Power Switch Ready bit in the Status register (STATUS.APWSRDY) is set when the Automatic Power Switch is ready to operate. The Automatic Power Switch Ready bit in the Interrupt Flag Status and Clear (INTFLAG.APSWRDY) will be set at the same time.

Related Links

Electrical Characteristics on page 1140

BOD33 Power Switch

When the Configuration bit field in the Battery Backup Power Switch register (BBPS.CONF) are selecting the BOD33, BOD33 will function as Battery Backup Power Switch. In this case, when the VDD voltage is below the BOD33 threshold, the backup domain supply is switched to VBAT.

Main Power Supply OK (PSOK) Pin Enable

The state of the Main Power VDD can be used to switch between supply sources as long as the Battery Backup Power Switch is not configured as Automatic Power Switch (i.e., BBPS.CONF not set to APWS): when the Main Power Supply OK Pin Enable bit in the BBPS register is written to '1' (BBPS.PSOKEN), restoring VDD will form a low-to-high transition on the PSOK pin. This low-to-high transition will switch the Backup Power Supply back to VDD.

Note: With BBPS.PSOKEN=0 and BBPS.CONF not configured to APWS, the device can not be restarted.

Backup Battery Power Switch Status

The Battery Backup Power Switch bit in the Status register (STATUS.BBPS) indicates whether the backup domain is currently powered by VDD or VBAT.

23.6.3.4. Sleep Mode Operation

The Battery Backup Power Switch is not stopped in any sleep mode.

Entering Battery Backup Mode

Entering backup mode can be triggered by either:

- Wait-for-interrupt (WFI) instruction.
- Automatic Power Switch (BBPS.CONF=APWS). When the Automatic Power Switch detects loss of Main Power, the Backup Domain will be powered by battery and the device will enter the backup mode.
- BOD33 detection: When the BOD33 detects loss of Main Power, the Backup Domain will be powered by battery and the device will enter the backup mode. For this trigger, the following register configuration is required: BOD33.ACTION=BKUP, BOD33.VMON=VDD, and BBPS.CONF=BOD33.

Related Links

PM – Power Manager on page 186

Leaving Battery Backup Mode

Leaving backup mode can be triggered by either:

- RTC requests and externally triggered RSTC requests, under one of these conditions:
  - The Backup Domain is supplied by Main Power, and the Battery Backup Power Switch is *not* forced (BBPS.CONF not set to FORCED)
  - The Battery Backup Power Switch *is* forced (BBPS.CONF is FORCED)

  The device is kept in battery-powered backup mode until Main Power is restored to supply the device. Then, the backup domain will be powered by Main Power.

- Automatic Power Switch. Leaving backup mode will happen when Main Power is restored and the Battery Backup Power Switch configuration (BBPS.CONF) is set to APWS: When BBPS.WAKEEN=1, the device will leave backup mode and wake up.
  When BBPS.WAKEEN=0, the backup domain will be powered by Main Power, but the device will stay in backup mode.
- PSOK pin. A low-to-high transition on PSOK will wake up the device if BBPS.PSOKEN=1, BBPS.WAKEEN=1, and the Battery Backup Power Switch is different from APWS (BBPS.CONF is not APWS).
  When BBPS.WAKEEN=0, the backup domain will be powered by Main Power, but the device will stay in backup mode.

23.6.4. Output Pins

The SUPC can drive two outputs. By writing a '1' to the corresponding Output Enable bit in the Backup Output Control register (BKOUT.EN), the OUTx pin is driven by the SUPC.

The OUT pin can be set by writing a '1' to the corresponding Set Output bit in the Backup Output Control register (BKOUT.SETx).

The OUT pin can be cleared by writing a '1' to the corresponding CLR bit (BKOUT.CLRx).

If a RTC Toggle Enable bit is written to '1' (BKOUT.RTCTGLx), the corresponding OUTx pin will toggle when an RTC event occurs.

23.6.5. Brown-Out Detectors

23.6.5.1. Initialization

Before a Brown Out Detector (BOD12 or BOD33) is enabled, it must be configured, as outlined by the following:

- Set the BOD threshold level (BODnn.LEVEL)
- Set the configuration in active, standby, backup modes (BODnn.ACTCDG, BODnn.STDBYCFG, BODnn.BKUP)
- Set the prescaling value if the BOD will run in sampling mode (BODnn.PSEL)
- Set the action and hysteresis (BODnn.ACTION and BODnn.HYST)

The BOD12 and BOD33 registers are Enable-Protected, meaning that they can only be written when the respective BOD is disabled (BOD12.ENABLE=0 and SYNCBUSY.BOD12EN=0, or BOD33.ENABLE=0 and SYNCBUSY.BOD33EN=0, respectively). As long as the Enable bit is '1', any writes to Enable-Protected registers will be discarded, and an APB error will be generated. The Enable bits are not Enable-Protected.

23.6.5.2. Enabling, Disabling, and Resetting

After power or user reset, the BOD33 and BOD12 register values are loaded from the NVM User Row.

The BODs are enabled by writing a '1' to the Enable bit in the respective BOD control register (BOD12.ENABLE or BOD33.ENABLE). The BODs are disabled by writing a '0' to the respective BODnn.ENABLE.

Related Links

NVM User Row Mapping on page 46

23.6.5.3. 3.3V Brown-Out Detector (BOD33)

The 3.3V Brown-Out Detector (BOD33) is able to monitor either the VDD supply .

The Voltage Monitored bit in the BOD33 Control register (BOD33.VMON) selects which supply is monitored in active and standby mode. In backup mode, BOD33 will always monitor the supply of the backup domain, i.e. either VDD or VBAT.

If VDD is monitored, the BOD33 compares the voltage with the brown-out threshold level. This level is set in the BOD33 Level field in the BOD33 register (BOD33.LEVEL). This level is used in all modes except the backup sleep modes. In backup sleep modes, a different voltage reference is used, which is configured by the BOD33.BKUPLEVEL bits.

When VDD crosses below the brown-out threshold level, the BOD33 can generate either an interrupt, a Reset, or an Automatic Battery Backup Power Switch, depending on the BOD33 Action bit field (BOD33.ACTION).

If VBAT is monitored, the BOD33 compares the voltage with the brown-out threshold level set in the BOD33 Backup Level field in the BOD33 register (BOD33.BKUPLEVEL).

When VBAT crosses below the backup brown-out threshold level, the BOD33 can generate either an interrupt or a Reset.

The BOD33 detection status can be read from the BOD33 Detection bit in the Status register (STATUS.BOD33DET).

At start-up or at Power-On Reset (POR), the BOD33 register values are loaded from the NVM User Row.

Related Links

NVM User Row Mapping on page 46

23.6.5.4. 1.2V Brown-Out Detector (BOD12)

The 1.2V Brown-Out Detector (BOD12) monitors the VDDCORE supply and compares the voltage with the brown-out threshold level set in the BOD12 Threshold Level bit field in the BOD12 register (BOD12.LEVEL).

When VDDCORE crosses below the brown-out threshold level, the BOD12 can generate either an interrupt or a reset.

The BOD12 detection status can be read from the BOD12 Detection bit (STATUS.BOD12DET) in the STATUS register.

At start-up including power-on reset (POR), the BOD12 register values are loaded from the NVM User Row.

Related Links

NVM User Row Mapping on page 46

23.6.5.5. Continuous Mode

Continuous mode is the default mode for both BOD12 and BOD33.

The *BOD33* is continuously monitoring the supply voltage (VDD or VBAT, depending on BOD33.VMON) if it is enabled (BOD33.ENABLE=1) and if the BOD33 Configuration bit in the BOD33 register is cleared (BOD33.ACTCFG=0 for active mode, BOD33.STDBYCFG=0 for standby mode).

The *BOD12* is continuously monitoring the VDDCORE supply voltage if it is enabled (BOD12.ENABLE=1) and if the BOD12 Configuration bit in the BOD12 register is cleared (BOD12.ACTCFG=0 and/or BOD12.STDBYCFG, respectively).

23.6.5.6. Sampling Mode

The Sampling Mode is a low-power mode where the BOD33 or BOD12 are being repeatedly enabled on a sampling clock's ticks. The BOD33 (or BOD12) will monitor the supply voltage (or core voltage) for a short period of time and then go to a low-power disabled state until the next sampling clock tick.

Sampling mode is enabled in Active mode for each BOD by writing the respective ACTCFG bit (BOD12.ACTCFG=1 and BOD33.ACTCFG=1). Sampling mode is enabled in Standby mode by writing to the respective STDBYCFG bit (BODnn.STBYCFG=1). The frequency of the clock ticks ($F_{clksampling}$) is controlled by the Prescaler Select bit groups in the respective BOD registers (BOD33.PSEL and BOD12.PSEL, respectively).

$$F_{clksampling} = \frac{F_{clkprescaler}}{2^{(\mathrm{PSEL}+1)}}$$

The prescaler signal ($F_{clkprescaler}$) is a 1KHz clock, output by the 32KHz Ultra Low Power Oscillator OSCULP32K.

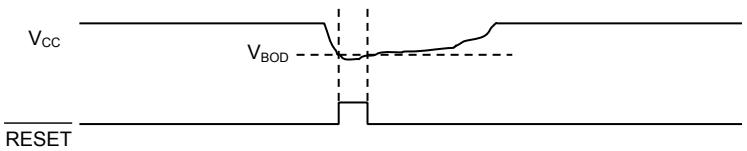
As the sampling clock is different from the APB clock domain, synchronization among the clocks is necessary. See also Synchronization on page 293.

23.6.5.7. Hysteresis

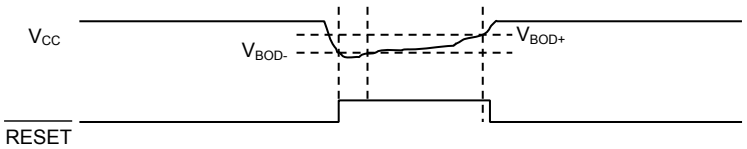
A hysteresis on the trigger threshold of a BOD will reduce the sensitivity to ripples on the monitored voltage: instead of switching $\overline{\text{RESET}}$ at each crossing of V_{BOD}, the thresholds for switching $\overline{\text{RESET}}$ on and off are separated (V_{BOD-} and V_{BOD+}, respectively).

Figure 23-2. BOD Hysteresis Principle

Hysteresis OFF:



Hysteresis ON:



Enabling the BOD33 hysteresis by writing the Hysteresis bit in the BOD33 register (BOD33.HYST) to '1' will add hysteresis to the BOD33 threshold level. BOD12.HYST=1 will add hysteresis to the BOD12 threshold level.

The hysteresis functionality can be used in both Continuous and Sampling Mode.

23.6.5.8. Sleep Mode Operation

Standby Mode

The BOD33 and the BOD12 can be used in standby mode if the BOD is enabled and the corresponding Run in Standby bit is written to '1' (BOD33.RUNSTDBY, BOD12.RUNSTDBY).

Both BODs can be configured to work in either Continuous or Sampling Mode by writing a '1' to the respective Configuration in Standby Sleep Mode bit (BOD33.STDBYCFG, BOD12.STDBYCFG).

Backup Mode

In Backup mode, the BOD12 is automatically disabled.

If the BOD33 is enabled and the Run in Backup sleep mode bit in the BOD33 register (BOD33.RUNBKUP) is written to '1', the BOD33 will operate in Sampling mode. In this state, the voltage monitored by BOD33 is always the supply of the backup domain, i.e. VDD or VBAT.

23.6.6. Interrupts

The SUPC has the following interrupt sources, which are either synchronous or asynchronous wake-up sources:

- VDDCORE Voltage Ready (VCORERDY), asynchronous
- Automatic Power Switch Ready Ready (APSWRDY), asynchronous
- Voltage Regulator Ready (VREGRDY) asynchronous
- BOD33 Ready (BOD33RDY), synchronous
- BOD33 Detection (BOD33DET), asynchronous
- BOD33 Synchronization Ready (B33SRDY), synchronous
- BOD12 Ready (BOD12RDY), synchronous
- BOD12 Detection (BOD12DET), asynchronous
- BOD12 Synchronization Ready (BOD12SRDY), synchronous

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) is set when the interrupt condition occurs.

Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR).

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until either the interrupt flag is cleared, the interrupt is disabled, or the SUPC is reset. See the INTFLAG register for details on how to clear interrupt flags. The SUPC has one common interrupt request line for all the interrupt sources. The user must read the INTFLAG register to determine which interrupt condition is present.

Note: Interrupts must be globally enabled for interrupt requests to be generated.

Related Links

Nested Vector Interrupt Controller on page 51
Sleep Mode Controller on page 191

23.6.7. Synchronization

The prescaler counters that are used to trigger brown-out detections operate asynchronously from the peripheral bus. As a consequence, the BOD12 and BOD33 Enable bits (BOD12.ENABLE and BOD33.ENABLE, respectively) need synchronization when written.

The Write-Synchronization of the Enable bit is triggered by writing a '1' to the Enable bit of the BOD12 or the BOD33 Control register. The Synchronization Ready bit (STATUS.B12SRDY or STATUS.B33SRDY) in the STATUS register will be cleared when the Write-Synchronization starts, and set again when the Write-Synchronization is complete. Writing to the same register while the Write-Synchronization is ongoing (STATUS.B33SRDY or STATUS.B12SRDY are '0') will generate an error without stalling the APB bus.

23.7. Register Summary

Offset	Name	Bit Pos.								
0x00	INTENCLR	7:0			B12SRDY	BOD12DET	BOD12RDY	B33SRDY	BOD33DET	BOD33RDY
0x01		15:8						VCORERDY	APWSRDY	VREGRDY
0x02		23:16								
0x03		31:24								
0x04	INTENSET	7:0			B12SRDY	BOD12DET	BOD12RDY	B33SRDY	BOD33DET	BOD33RDY
0x05		15:8						VCORERDY	APWSRDY	VREGRDY
0x06		23:16								
0x07		31:24								
0x08	INTFLAG	7:0			B12SRDY	BOD12DET	BOD12RDY	B33SRDY	BOD33DET	BOD33RDY
0x09		15:8						VCORERDY	APWSRDY	VREGRDY
0x0A		23:16								
0x0B		31:24								
0x0C	STATUS	7:0			B12SRDY	BOD12DET	BOD12RDY	B33SRDY	BOD33DET	BOD33RDY
0x0D		15:8					BBPS	VCORERDY	APWSRDY	VREGRDY
0x0E		23:16								
0x0F		31:24								
0x10	BOD33	7:0	RUNBKUP	RUNSTDBY	STDBYCFG	ACTION[1:0]		HYST	ENABLE	
0x11		15:8	PSEL[3:0]					VMON		ACTCFG
0x12		23:16			LEVEL[5:0]					
0x13		31:24			BKUPLEVEL[5:0]					
0x14	BOD12	7:0		RUNSTDBY	STDBYCFG	ACTION[1:0]		HYST	ENABLE	
0x15		15:8	PSEL[3:0]							ACTCFG
0x16		23:16			LEVEL[5:0]					
0x17		31:24								
0x18	VREG	7:0		RUNSTDBY	STDBYPL0			SEL	ENABLE	
0x19		15:8								LPEFF
0x1A		23:16					VSVSTEP[3:0]			
0x1B		31:24	VSPER[7:0]							
0x1C	VREF	7:0	ONDEMAND	RUNSTDBY				VREFOE	TSEN	
0x1D		15:8								
0x1E		23:16					SEL[3:0]			
0x1F		31:24								
0x20	BBPS	7:0					PSOKEN	WAKEEN	CONF[1:0]	
0x21		15:8								
0x22		23:16								
0x23		31:24								
0x24	BKOUT	7:0							EN[1:0]	
0x25		15:8							CLR[1:0]	
0x26		23:16							SET[1:0]	
0x27		31:24							RTCTGL[1:0]	
0x28	BKIN	7:0						BKIN[2:0]		
0x29		15:8								
0x2A		23:16								
0x2B		31:24								

23.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). PAC Write-protection is denoted by the "PAC Write-Protection" property in each individual register description. Refer to Register Access Protection on page 285 for details.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Write-Synchronized" or the "Read-Synchronized" property in each individual register description. Refer to Synchronization on page 293 for details.

23.8.1. Interrupt Enable Clear

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Name: INTENCLR
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						VCORERDY	APWSRDY	VREGRDY
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
			B12SRDY	BOD12DET	BOD12RDY	B33SRDY	BOD33DET	BOD33RDY
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 10 – VCORERDY: VDDCORE Voltage Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the VDDCORE Ready Interrupt Enable bit, which disables the VDDCORE Ready interrupt.

Value	Description
0	The VDDCORE Ready interrupt is disabled.
1	The VDDCORE Ready interrupt is enabled and an interrupt request will be generated when the VCORERDY Interrupt Flag is set.

Bit 9 – APWSRDY: Automatic Power Switch Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Automatic Power Switch Ready Interrupt Enable bit, which disables the Automatic Power Switch Ready interrupt.

Value	Description
0	The Automatic Power Switch Ready interrupt is disabled.
1	The Automatic Power Switch Ready interrupt is enabled and an interrupt request will be generated when the APWSRDY Interrupt Flag is set.

Bit 8 – VREGRDY: Voltage Regulator Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Voltage Regulator Ready Interrupt Enable bit, which disables the Voltage Regulator Ready interrupt.

Value	Description
0	The Voltage Regulator Ready interrupt is disabled.
1	The Voltage Regulator Ready interrupt is enabled and an interrupt request will be generated when the Voltage Regulator Ready Interrupt Flag is set.

Bit 5 – B12SRDY: BOD12 Synchronization Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the BOD12 Synchronization Ready Interrupt Enable bit, which disables the BOD12 Synchronization Ready interrupt.

Value	Description
0	The BOD12 Synchronization Ready interrupt is disabled.
1	The BOD12 Synchronization Ready interrupt is enabled, and an interrupt request will be generated when the BOD12 Synchronization Ready Interrupt flag is set.

Bit 4 – BOD12DET: BOD12 Detection Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the BOD12 Detection Interrupt Enable bit, which disables the BOD12 Detection interrupt.

Value	Description
0	The BOD12 Detection interrupt is disabled.
1	The BOD12 Detection interrupt is enabled, and an interrupt request will be generated when the BOD12 Detection Interrupt flag is set.

Bit 3 – BOD12RDY: BOD12 Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the BOD12 Ready Interrupt Enable bit, which disables the BOD12 Ready interrupt.

Value	Description
0	The BOD12 Ready interrupt is disabled.
1	The BOD12 Ready interrupt is enabled and an interrupt request will be generated when the BOD12 Ready Interrupt flag is set.

Bit 2 – B33SRDY: BOD33 Synchronization Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the BOD33 Synchronization Ready Interrupt Enable bit, which disables the BOD33 Synchronization Ready interrupt.

Value	Description
0	The BOD33 Synchronization Ready interrupt is disabled.
1	The BOD33 Synchronization Ready interrupt is enabled, and an interrupt request will be generated when the BOD33 Synchronization Ready Interrupt flag is set.

Bit 1 – BOD33DET: BOD33 Detection Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the BOD33 Detection Interrupt Enable bit, which disables the BOD33 Detection interrupt.

Value	Description
0	The BOD33 Detection interrupt is disabled.
1	The BOD33 Detection interrupt is enabled, and an interrupt request will be generated when the BOD33 Detection Interrupt flag is set.

Bit 0 – BOD33RDY: BOD33 Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the BOD33 Ready Interrupt Enable bit, which disables the BOD33 Ready interrupt.

Value	Description
0	The BOD33 Ready interrupt is disabled.
1	The BOD33 Ready interrupt is enabled, and an interrupt request will be generated when the BOD33 Ready Interrupt flag is set.

23.8.2. Interrupt Enable Set

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Name: INTENSET
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						VCORERDY	APWSRDY	VREGRDY
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
			B12SRDY	BOD12DET	BOD12RDY	B33SRDY	BOD33DET	BOD33RDY
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 10 – VCORERDY: VDDCORE Voltage Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the VDDCORE Ready Interrupt Enable bit, which enables the VDDCORE Ready interrupt.

Value	Description
0	The VDDCORE Ready interrupt is disabled.
1	The VDDCORE Ready interrupt is enabled and an interrupt request will be generated when the VCORERDY Interrupt Flag is set.

Bit 9 – APWSRDY: Automatic Power Switch Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Automatic Power Switch Ready Interrupt Enable bit, which enables the Automatic Power Switch Ready interrupt.

Value	Description
0	The Automatic Power Switch Ready interrupt is disabled.
1	The Automatic Power Switch Ready interrupt is enabled and an interrupt request will be generated when the Automatic Power Switch Ready Interrupt Flag is set.

Bit 8 – VREGRDY: Voltage Regulator Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Voltage Regulator Ready Interrupt Enable bit, which enables the Voltage Regulator Ready interrupt.

Value	Description
0	The Voltage Regulator Ready interrupt is disabled.
1	The Voltage Regulator Ready interrupt is enabled and an interrupt request will be generated when the Voltage Regulator Ready Interrupt Flag is set.

Bit 5 – B12SRDY: BOD12 Synchronization Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the BOD12 Synchronization Ready Interrupt Enable bit, which enables the BOD12 Synchronization Ready interrupt.

Value	Description
0	The BOD12 Synchronization Ready interrupt is disabled.
1	The BOD12 Synchronization Ready interrupt is enabled, and an interrupt request will be generated when the BOD12 Synchronization Ready Interrupt flag is set.

Bit 4 – BOD12DET: BOD12 Detection Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the BOD12 Detection Interrupt Enable bit, which enables the BOD12 Detection interrupt.

Value	Description
0	The BOD12 Detection interrupt is disabled.
1	The BOD12 Detection interrupt is enabled and an interrupt request will be generated when the BOD12 Detection Interrupt Flag is set.

Bit 3 – BOD12RDY: BOD12 Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the BOD12 Ready Interrupt Enable bit, which enables the BOD12 Ready interrupt.

Value	Description
0	The BOD12 Ready interrupt is disabled.
1	The BOD12 Ready interrupt is enabled, and an interrupt request will be generated when the BOD12 Ready Interrupt flag is set.

Bit 2 – B33SRDY: BOD33 Synchronization Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the BOD33 Synchronization Ready Interrupt Enable bit, which enables the BOD33 Synchronization Ready interrupt.

Value	Description
0	The BOD33 Synchronization Ready interrupt is disabled.
1	The BOD33 Synchronization Ready interrupt is enabled, and an interrupt request will be generated when the BOD33 Synchronization Ready Interrupt flag is set.

Bit 1 – BOD33DET: BOD33 Detection Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the BOD33 Detection Interrupt Enable bit, which enables the BOD33 Detection interrupt.

Value	Description
0	The BOD33 Detection interrupt is disabled.
1	The BOD33 Detection interrupt is enabled, and an interrupt request will be generated when the BOD33 Detection Interrupt flag is set.

Bit 0 – BOD33RDY: BOD33 Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the BOD33 Ready Interrupt Enable bit, which enables the BOD33 Ready interrupt.

Value	Description
0	The BOD33 Ready interrupt is disabled.
1	The BOD33 Ready interrupt is enabled, and an interrupt request will be generated when the BOD33 Ready Interrupt flag is set.

23.8.3. Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x08
Reset: 0x0000010X - X= determined from NVM User Row (0xX=0bx00y)
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						VCORERDY	APWSRDY	VREGRDY
Access						R/W	R/W	R/W
Reset						0	0	1

Bit	7	6	5	4	3	2	1	0
			B12SRDY	BOD12DET	BOD12RDY	B33SRDY	BOD33DET	BOD33RDY
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	x	0	0	y

Bit 10 – VCORERDY: VDDCORE Voltage Ready
This flag is cleared by writing a '1 to it.

This flag is set on a zero-to-one transition of the VDDCORE Ready bit in the Status register (STATUS.VCORERDY) and will generate an interrupt request if INTENSET.VCORERDY=1.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the VCORERDY interrupt flag.

Bit 9 – APWSRDY: Automatic Power Switch Ready
This flag is cleared by writing a '1' to it.

This flag is set on a zero-to-one transition of the Automatic Power Switch Ready bit in the Status register (STATUS.APWSRDY) and will generate an interrupt request if INTENSET.APWSRDY=1.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the APWSRDY interrupt flag.

Bit 8 – VREGRDY: Voltage Regulator Ready
This flag is cleared by writing a '1' to it.

This flag is set on a zero-to-one transition of the Voltage Regulator Ready bit in the Status register (STATUS.VREGRDY) and will generate an interrupt request if INTENSET.VREGRDY=1.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the VREGRDY interrupt flag.

Bit 5 – B12SRDY: BOD12 Synchronization Ready
This flag is cleared by writing a '1' to it.

This flag is set on a zero-to-one transition of the BOD12 Synchronization Ready bit in the Status register (STATUS.B12SRDY) and will generate an interrupt request if INTENSET.B12SRDY=1.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the BOD12 Synchronization Ready interrupt flag.

Bit 4 – BOD12DET: BOD12 Detection
This flag is cleared by writing a '1' to it.

This flag is set on a zero-to-one transition of the BOD12 Detection bit in the Status register (STATUS.BOD12DET) and will generate an interrupt request if INTENSET.BOD12DET=1.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the BOD12 Detection interrupt flag.

Bit 3 – BOD12RDY: BOD12 Ready
This flag is cleared by writing a '1' to it.

This flag is set on a zero-to-one transition of the BOD12 Ready bit in the Status register (STATUS.BOD12RDY) and will generate an interrupt request if INTENSET.BOD12RDY=1.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the BOD12 Ready interrupt flag.

The BOD12 can be enabled at startup from Flash User Row.

Bit 2 – B33SRDY: BOD33 Synchronization Ready
This flag is cleared by writing a '1' to it.

This flag is set on a zero-to-one transition of the BOD33 Synchronization Ready bit in the Status register (STATUS.B33SRDY) and will generate an interrupt request if INTENSET.B33SRDY=1.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the BOD33 Synchronization Ready interrupt flag.

Bit 1 – BOD33DET: BOD33 Detection
This flag is cleared by writing a '1' to it.

This flag is set on a zero-to-one transition of the BOD33 Detection bit in the Status register (STATUS.BOD33DET) and will generate an interrupt request if INTENSET.BOD33DET=1.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the BOD33 Detection interrupt flag.

Bit 0 – BOD33RDY: BOD33 Ready
This flag is cleared by writing a '1' to it.

This flag is set on a zero-to-one transition of the BOD33 Ready bit in the Status register (STATUS.BOD33RDY) and will generate an interrupt request if INTENSET.BOD33RDY=1.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the BOD33 Ready interrupt flag.

The BOD33 can be enabled.

23.8.4. Status

Name: STATUS
Offset: 0x0C
Reset: Determined from NVM User Row
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					BBPS	VCORERDY	APWSRDY	VREGRDY
Access					R	R	R	R
Reset					0	1	0	1

Bit	7	6	5	4	3	2	1	0
			B12SRDY	BOD12DET	BOD12RDY	B33SRDY	BOD33DET	BOD33RDY
Access			R	R	R	R	R	R
Reset			0	0	x	0	0	y

Bit 11 – BBPS: Battery Backup Power Switch

Value	Description
0	the backup domain is supplied by VDD.
1	the backup domain is supplied by VBAT.

Bit 10 – VCORERDY: VDDCORE Voltage Ready

Value	Description
0	the VDDCORE voltage is not as expected.
1	the VDDCORE voltage is the target voltage.

Bit 9 – APWSRDY: Automatic Power Switch Ready

Value	Description
0	The Automatic Power Switch is not ready.
1	The Automatic Power Switch is ready.

Bit 8 – VREGRDY: Voltage Regulator Ready

Value	Description
0	The selected voltage regulator in VREG.SEL is not ready.
1	The voltage regulator selected in VREG.SEL is ready and the core domain is supplied by this voltage regulator.

Bit 5 – B12SRDY: BOD12 Synchronization Ready

Value	Description
0	BOD12 synchronization is ongoing.
1	BOD12 synchronization is complete.

Bit 4 – BOD12DET: BOD12 Detection

Value	Description
0	No BOD12 detection.
1	BOD12 has detected that the core power supply is going below the BOD12 reference value.

Bit 3 – BOD12RDY: BOD12 Ready
The BOD12 can be enabled at start-up from NVM User Row.

Value	Description
0	BOD12 is not ready.
1	BOD12 is ready.

Bit 2 – B33SRDY: BOD33 Synchronization Ready

Value	Description
0	BOD33 synchronization is ongoing.
1	BOD33 synchronization is complete.

Bit 1 – BOD33DET: BOD33 Detection

Value	Description
0	No BOD33 detection.
1	BOD33 has detected that the I/O power supply is going below the BOD33 reference value.

Bit 0 – BOD33RDY: BOD33 Ready
The BOD33 can be enabled at start-up from NVM User Row.

Value	Description
0	BOD33 is not ready.
1	BOD33 is ready.

23.8.5. 3.3V Brown-Out Detector (BOD33) Control

Name: BOD33
Offset: 0x10
Reset: Determined from NVM User Row
Property: Write-Synchronized, Enable-Protected, PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
			BKUPLEVEL[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
			LEVEL[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			x	x	x	x	x	x

Bit	15	14	13	12	11	10	9	8
	PSEL[3:0]					VMON		ACTCFG
Access	R/W	R/W	R/W	R/W		R/W		R/W
Reset	0	0	0	0		0		0

Bit	7	6	5	4	3	2	1	0
	RUNBKUP	RUNSTDBY	STDBYCFG	ACTION[1:0]		HYST	ENABLE	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	y	y	0	z	

Bits 29:24 – BKUPLEVEL[5:0]: BOD33 Threshold Level on VBAT or in Backup Sleep Mode
These bits set the triggering voltage threshold for the BOD33 when the BOD33 monitors VBAT or in backup sleep mode.

This bit field is not synchronized.

Bits 21:16 – LEVEL[5:0]: BOD33 Threshold Level on VDD
These bits set the triggering voltage threshold for the BOD33 when the BOD33 monitors VDD except in backup sleep mode.

These bits are loaded from NVM User Row at start-up.

This bit field is not synchronized.

Bits 15:12 – PSEL[3:0]: Prescaler Select
Selects the prescaler divide-by output for the BOD33 sampling mode. The input clock comes from the OSCULP32K 1KHz output.

Value	Name	Description
0x0	DIV2	Divide clock by 2
0x1	DIV4	Divide clock by 4
0x2	DIV8	Divide clock by 8
0x3	DIV16	Divide clock by 16

Value	Name	Description
0x4	DIV32	Divide clock by 32
0x5	DIV64	Divide clock by 64
0x6	DIV128	Divide clock by 128
0x7	DIV256	Divide clock by 256
0x8	DIV512	Divide clock by 512
0x9	DIV1024	Divide clock by 1024
0xA	DIV2048	Divide clock by 2048
0xB	DIV4096	Divide clock by 4096
0xC	DIV8192	Divide clock by 8192
0xD	DIV16384	Divide clock by 16384
0xE	DIV32768	Divide clock by 32768
0xF	DIV65536	Divide clock by 65536

Bit 10 – VMON: Voltage Monitored in Active and Standby Mode
This bit is not synchronized.

Value	Description
0	The BOD33 monitors the VDD power pin in active and standby mode.
1	The BOD33 monitors the VBAT power pin in active and standby mode.

Bit 8 – ACTCFG: BOD33 Configuration in Active Sleep Mode
This bit is not synchronized.

Value	Description
0	In active mode, the BOD33 operates in continuous mode.
1	In active mode, the BOD33 operates in sampling mode.

Bit 7 – RUNBKUP: BOD33 Configuration in Backup Sleep Mode
This bit is not synchronized.

Value	Description
0	In backup sleep mode, the BOD33 is disabled.
1	In backup sleep mode, the BOD33 is enabled and configured in sampling mode.

Bit 6 – RUNSTDBY: Run in Standby
This bit is not synchronized.

Value	Description
0	In standby sleep mode, the BOD33 is disabled.
1	In standby sleep mode, the BOD33 is enabled.

Bit 5 – STDBYCFG: BOD33 Configuration in Standby Sleep Mode
If the RUNSTDBY bit is set to '1', the STDBYCFG bit sets the BOD33 configuration in standby sleep mode.

This bit is not synchronized.

Value	Description
0	In standby sleep mode, the BOD33 is enabled and configured in continuous mode.
1	In standby sleep mode, the BOD33 is enabled and configured in sampling mode.

Bits 4:3 – ACTION[1:0]: BOD33 Action
These bits are used to select the BOD33 action when the supply voltage crosses below the BOD33 threshold.

These bits are loaded from NVM User Row at start-up.

This bit field is not synchronized.

Value	Name	Description
0x0	NONE	No action
0x1	RESET	The BOD33 generates a reset
0x2	INT	The BOD33 generates an interrupt
0x3	BKUP	The BOD33 puts the device in backup sleep mode if VMON=0. No action if VMON=1.

Bit 2 – HYST: Hysteresis
This bit indicates whether hysteresis is enabled for the BOD33 threshold voltage.

This bit is loaded from NVM User Row at start-up.

This bit is not synchronized.

Value	Description
0	No hysteresis.
1	Hysteresis enabled.

Bit 1 – ENABLE: Enable
This bit is loaded from NVM User Row at start-up.

This bit is not enable-protected.

Value	Description
0	BOD33 is disabled.
1	BOD33 is enabled.

23.8.6. 1.2V Brown-Out Detector (BOD12) Control

Name: BOD12
Offset: 0x14
Reset: 0x00XX00YZ - determined by NVM User Row (0xXX=0bxxxxxx, 0xYZ=0x000yy0z0)
Property: Write-Synchronized, Enable-Protected,PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			LEVEL[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			x	x	x	x	x	x

Bit	15	14	13	12	11	10	9	8
	PSEL[3:0]							ACTCFG
Access	R/W	R/W	R/W	R/W				R/W
Reset	0	0	0	0				0

Bit	7	6	5	4	3	2	1	0
		RUNSTDBY	STDBYCFG	ACTION[1:0]		HYST	ENABLE	
Access		R/W	R/W	R/W	R/W	R/W	R/W	
Reset		0	0	y	y	0	z	

Bits 21:16 – LEVEL[5:0]: BOD12 Threshold Level
This field sets the triggering voltage threshold for the BOD12.

These bits are loaded from NVM User Row at start-up.

This field is not synchronized.

Bits 15:12 – PSEL[3:0]: Prescaler Select
Selects the prescaler divide-by output for the BOD12 Sampling mode. The input clock comes from the OSCULP32K 1KHz output.

Value	Name	Description
0x0	DIV2	Divide clock by 2
0x1	DIV4	Divide clock by 4
0x2	DIV8	Divide clock by 8
0x3	DIV16	Divide clock by 16
0x4	DIV32	Divide clock by 32
0x5	DIV64	Divide clock by 64
0x6	DIV128	Divide clock by 128
0x7	DIV256	Divide clock by 256

Value	Name	Description
0x8	DIV512	Divide clock by 512
0x9	DIV1024	Divide clock by 1024
0xA	DIV2048	Divide clock by 2048
0xB	DIV4096	Divide clock by 4096
0xC	DIV8192	Divide clock by 8192
0xD	DIV16384	Divide clock by 16384
0xE	DIV32768	Divide clock by 32768
0xF	DIV65536	Divide clock by 65536

Bit 8 – ACTCFG: BOD12 Configuration in Active Sleep Mode
This field is not synchronized.

Value	Description
0	In active mode, the BOD12 operates in continuous mode.
1	In active mode, the BOD12 operates in sampling mode.

Bit 6 – RUNSTDBY: Run in Standby
This bit is not synchronized.

Value	Description
0	In standby sleep mode, the BOD12 is disabled.
1	In standby sleep mode, the BOD12 is enabled.

Bit 5 – STDBYCFG: BOD12 Configuration in Standby Sleep Mode
If the RUNSTDBY bit is set to 1, the STDBYCFG bit sets the BOD12 configuration in standby sleep mode.

This field is not synchronized.

Value	Description
0	In standby sleep mode, the BOD12 is enabled and configured in continuous mode.
1	In standby sleep mode, the BOD12 is enabled and configured in sampling mode.

Bits 4:3 – ACTION[1:0]: BOD12 Action
These bits are used to select the BOD12 action when the supply voltage crosses below the BOD12 threshold.

These bits are loaded from NVM User Row at start-up.

This field is not synchronized.

Value	Name	Description
0x0	NONE	No action.
0x1	RESET	The BOD12 generates a reset.

Value	Name	Description
0x2	INT	The BOD12 generates an interrupt.
0x3	-	Reserved

Bit 2 – HYST: Hysteresis
This bit indicates whether hysteresis is enabled for the BOD12 threshold voltage:

This bit is not synchronized.

Value	Description
0	No hysteresis.
1	Hysteresis enabled.

Bit 1 – ENABLE: Enable
This bit is loaded from NVM User Row at start-up.

This bit is not enable-protected.

Value	Description
0	BOD12 is disabled.
1	BOD12 is enabled.

23.8.7. Voltage Regulator System (VREG) Control

Name: VREG
Offset: 0x18
Reset: 0x00000002
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	VSPER[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
					VSVSTEP[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
								LPEFF
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
		RUNSTDBY	STDBYPL0			SEL	ENABLE	
Access		R/W	R/W			R/W	R/W	
Reset		0	1			0	1	

Bits 31:24 – VSPER[7:0]: Voltage Scaling Period
This bitfield sets the period between the voltage steps when the VDDCORE voltage is changing in µs.

If VSPER=0, the period between two voltage steps is 1µs.

Bits 19:16 – VSVSTEP[3:0]: Voltage Scaling Voltage Step
This field sets the voltage step height when the VDDCORE voltage is changing to reach the target VDDCORE voltage.

The voltage step is equal to $2^{VSVSTEP*}$ min_step.

See the Electrical Characteristics chapter for the min_step voltage level.

Bit 8 – LPEFF: Low power Mode Efficiency

Value	Description
0	The voltage regulator in Low power mode has the default efficiency and supports the whole VDD range (1.62V to 3.6V).
1	The voltage regulator in Low power mode has the highest efficiency and supports a limited VDD range (2.5V to 3.6V).

Bit 6 – RUNSTDBY: Run in Standby

Value	Description
0	The voltage regulator is in low power mode in Standby sleep mode.
1	The voltage regulator is in normal mode in Standby sleep mode.

Bit 5 – STDBYPL0: Standby in PL0
This bit selects the performance level (PL) of the voltage regulator for the Standby sleep mode. This bit is only considered when RUNSTDBY=1.

Value	Description
0	In Standby sleep mode, the voltage regulator remains in the current performance level.
1	In Standby sleep mode, the voltage regulator is used in PL0.

Bit 2 – SEL: Voltage Regulator Selection
This bit is loaded from NVM User Row at start-up.

Value	Description
0	The voltage regulator in active mode is a LDO voltage regulator.
1	The voltage regulator in active mode is a buck converter.

Bit 1 – ENABLE: Enable

Value	Description
0	The voltage regulator is disabled.
1	The voltage regulator is enabled.

23.8.8. Voltage References System (VREF) Control

Name: VREF
Offset: 0x1C
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					SEL[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY				VREFOE	TSEN	
Access	R/W	R/W				R/W	R/W	
Reset	0	0				0	0	

Bits 19:16 – SEL[3:0]: Voltage Reference Selection
These bits select the Voltage Reference for the ADC/DAC.

Value	Name	Description
0x0	1V0	1.0V voltage reference typical value
0x1	1V1	1.1V voltage reference typical value
0x2	1V2	1.2V voltage reference typical value
0x3	1V25	1.25V voltage reference typical value
0x4	2V0	2.0V voltage reference typical value
0x5	2V2	2.2V voltage reference typical value
0x6	2V4	2.4V voltage reference typical value
0x7	2V5	2.5V voltage reference typical value
Others		Reserved

Bit 7 – ONDEMAND: On Demand Control
The On Demand operation mode allows to enable or disable the voltage reference depending on peripheral requests.

Value	Description
0	The voltage reference is always on, if enabled.
1	The voltage reference is enabled when a peripheral is requesting it. The voltage reference is disabled if no peripheral is requesting it.

Bit 6 – RUNSTDBY: Run In Standby
The bit controls how the voltage reference behaves during standby sleep mode.

Value	Description
0	The voltage reference is halted during standby sleep mode.
1	The voltage reference is not stopped in standby sleep mode. If VREF.ONDEMAND=1, the voltage reference will be running when a peripheral is requesting it. If VREF.ONDEMAND=0, the voltage reference will always be running in standby sleep mode.

Bit 2 – VREFOE: Voltage Reference Output Enable

Value	Description
0	The Voltage Reference output is not available as an ADC input channel.
1	The Voltage Reference output is routed to an ADC input channel.

Bit 1 – TSEN: Temperature Sensor Enable

Value	Description
0	Temperature Sensor is disabled.
1	Temperature Sensor is enabled and routed to an ADC input channel.

23.8.9. Battery Backup Power Switch (BBPS) Control

Name: BBPS
Offset: 0x20
Reset: 0x0000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					PSOKEN	WAKEEN	CONF[1:0]	
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 3 – PSOKEN: Power Supply OK Enable

Value	Description
0	The PSOK pin is not used.
1	The PSOK pin is used to determine the status of the Main Power Supply.

Bit 2 – WAKEEN: Wake Enable

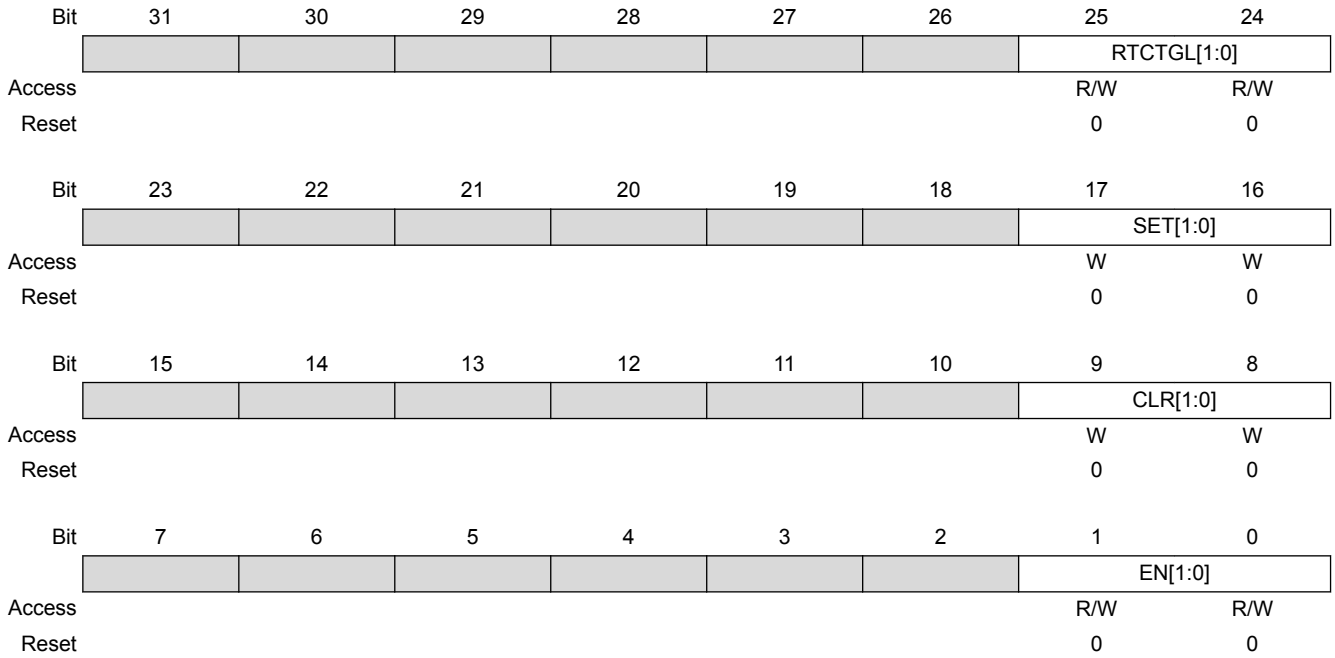
Value	Description
0	The device is not woken up when switched from battery backup power to Main Power.
1	The device is woken up when switched from battery backup power to Main Power.

Bits 1:0 – CONF[1:0]: Battery Backup Power Switch Configuration

Value	Name	Description
0x0	NONE	The backup domain is always supplied by Main Power.
0x1	APWS	The power switch is handled by the Automatic Power Switch.
0x2	FORCED	The backup domain is always supplied by Battery Backup Power.
0x3	BOD33	The power switch is handled by the BOD33.

23.8.10. Backup Output (BKOUT) Control

Name: BKOUT
Offset: 0x24
Reset: 0x0000
Property: PAC Write-Protection



Bits 25:24 – RTCTGL[1:0]: RTC Toggle Output

Value	Description
0	The output will not toggle on RTC event.
1	The output will toggle on RTC event.

Bits 17:16 – SET[1:0]: Set Output
Writing a '0' to a bit has no effect.

Writing a '1' to a bit will set the corresponding output.

Reading this bit returns '0'.

Bits 9:8 – CLR[1:0]: Clear Output
Writing a '0' to a bit has no effect.

Writing a '1' to a bit will clear the corresponding output.

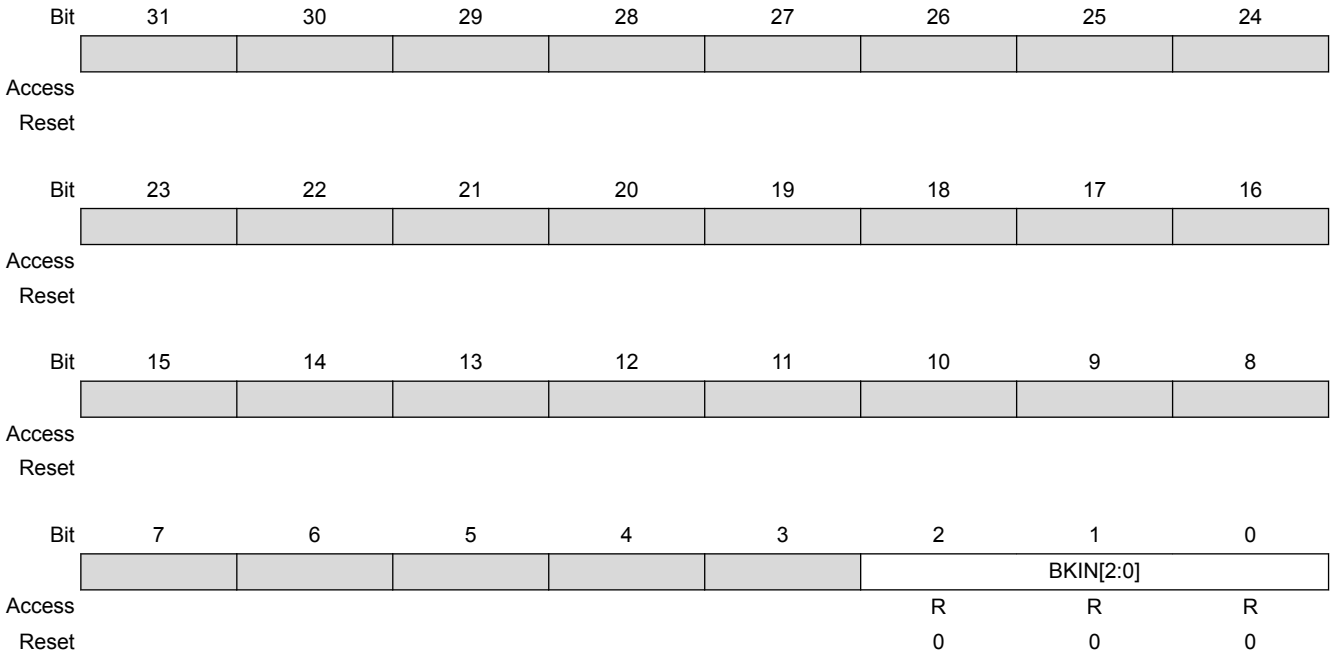
Reading this bit returns '0'.

Bits 1:0 – EN[1:0]: Enable Output

Value	Description
0	The output is not enabled.
1	The output is enabled and driven by the SUPC.

23.8.11. Backup Input (BKIN) Value

Name: BKIN
Offset: 0x28
Reset: 0x0000
Property: -



Bits 2:0 – BKIN[2:0]: Backup I/O Data Input Value
These bits are cleared when the corresponding backup I/O pin detects a logical low level on the input pin or when the backup I/O is not enabled.

These bits are set when the corresponding backup I/O pin detects a logical high level on the input pin when the backup I/O is enabled.

BKIN[2:0]	PAD	Description
BKIN[0]	PSOK	If BBPS.PSOKEN=1, BKIN[0] will give the input value of the PSOK pin
BKIN[1]	OUT[0]	If BKOUT.EN[0]=1, BKIN[1] will give the input value of the OUT[0] pin
BKIN[2]	OUT[1]	If BKOUT.EN[1]=1, BKIN[2] will give the input value of the OUT[1] pin

24. WDT – Watchdog Timer

24.1. Overview

The Watchdog Timer (WDT) is a system function for monitoring correct program operation. It makes it possible to recover from error situations such as runaway or deadlocked code. The WDT is configured to a predefined time-out period, and is constantly running when enabled. If the WDT is not cleared within the time-out period, it will issue a system reset. An early-warning interrupt is available to indicate an upcoming watchdog time-out condition.

The window mode makes it possible to define a time slot (or window) inside the total time-out period during which the WDT must be cleared. If the WDT is cleared outside this window, either too early or too late, a system reset will be issued. Compared to the normal mode, this can also catch situations where a code error causes the WDT to be cleared frequently.

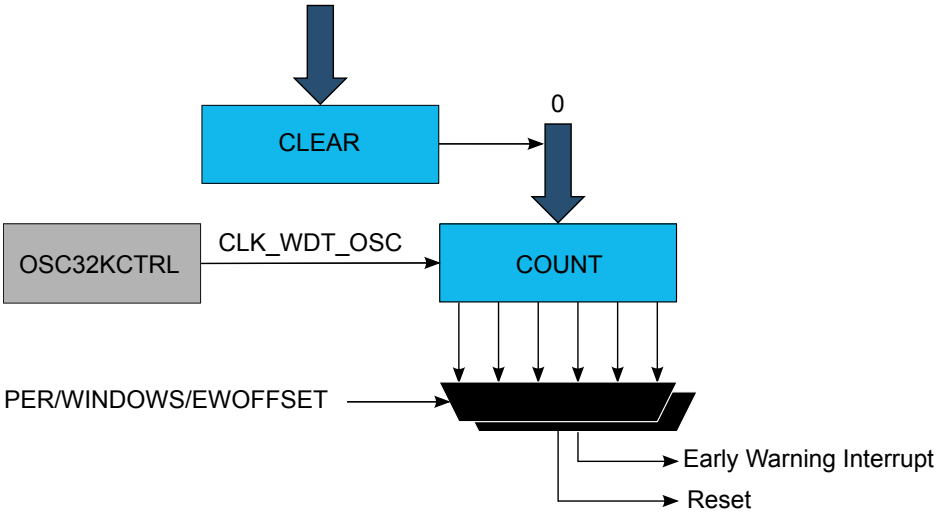
When enabled, the WDT will run in active mode and all sleep modes. It is asynchronous and runs from a CPU-independent clock source. The WDT will continue operation and issue a system reset or interrupt even if the main clocks fail.

24.2. Features

- Issues a system reset if the Watchdog Timer is not cleared before its time-out period
- Early Warning interrupt generation
- Asynchronous operation from dedicated oscillator
- Two types of operation
  - Normal
  - Window mode
- Selectable time-out periods
  - From 8 cycles to 16,384 cycles in Normal mode
  - From 16 cycles to 32,768 cycles in Window mode
- Always-On capability

24.3. Block Diagram

Figure 24-1. WDT Block Diagram



24.4. Signal Description

Not applicable.

24.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

24.5.1. I/O Lines

Not applicable.

24.5.2. Power Management

The WDT can continue to operate in any sleep mode where the selected source clock is running. The WDT interrupts can be used to wake up the device from sleep modes. The events can trigger other operations in the system without exiting sleep modes.

Related Links

PM – Power Manager on page 186

24.5.3. Clocks

The WDT bus clock (CLK_WDT_APB) can be enabled and disabled (masked) in the Main Clock module (MCLK).

A 1KHz oscillator clock (CLK_WDT_OSC) is required to clock the WDT internal counter. This clock must be configured and enabled in the 32KHz Oscillator Controller (OSC32KCTRL) before using the WDT.

CLK_WDT_OSC is normally sourced from the clock of the internal ultra-low-power oscillator, OSCULP32K. Due to the ultra-low-power design, the oscillator is not very accurate, and so the exact time-out period may vary from device to device. This variation must be kept in mind when designing software that uses the WDT to ensure that the time-out periods used are valid for all devices.

The counter clock CLK_WDT_OSC is asynchronous to the bus clock (CLK_WDT_APB). Due to this asynchronicity, writing to certain registers will require synchronization between the clock domains. Refer to Synchronization on page 327 for further details.

Related Links

Peripheral Clock Masking on page 151

OSC32KCTRL – 32KHz Oscillators Controller on page 264

Electrical Characteristics on page 1140

24.5.4. DMA

Not applicable.

24.5.5. Interrupts

The interrupt request line is connected to the interrupt controller. Using the WDT interrupt(s) requires the interrupt controller to be configured first.

Related Links

Nested Vector Interrupt Controller on page 51

Overview on page 51

24.5.6. Events

Not applicable.

24.5.7. Debug Operation

When the CPU is halted in debug mode the WDT will halt normal operation.

24.5.8. Register Access Protection

All registers with write-access can be write-protected optionally by the Peripheral Access Controller (PAC), except the following registers:

- Interrupt Flag Status and Clear (INTFLAG) register

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

PAC write-protection does not apply to accesses through an external debugger.

24.5.9. Analog Connections

Not applicable.

24.6. Functional Description

24.6.1. Principle of Operation

The Watchdog Timer (WDT) is a system for monitoring correct program operation, making it possible to recover from error situations such as runaway code, by issuing a Reset. When enabled, the WDT is a constantly running timer that is configured to a predefined time-out period. Before the end of the time-out period, the WDT should be set back, or else, a system Reset is issued.

The WDT has two modes of operation, Normal mode and Window mode. Both modes offer the option of Early Warning interrupt generation. The description for each of the basic modes is given below. The settings in the Control A register (CTRLA) and the Interrupt Enable register (handled by INTENCLR/INTENSET) determine the mode of operation:

Table 24-1. WDT Operating Modes

CTRLA.ENABLE	CTRLA.WEN	Interrupt Enable	Mode
0	x	x	Stopped
1	0	0	Normal mode
1	0	1	Normal mode with Early Warning interrupt
1	1	0	Window mode
1	1	1	Window mode with Early Warning interrupt

24.6.2. Basic Operation

24.6.2.1. Initialization

The following bits are enable-protected, meaning that they can only be written when the WDT is disabled (CTRLA.ENABLE=0):

- Control A register (CTRLA), except the Enable bit (CTRLA.ENABLE)
- Configuration register (CONFIG)
- Early Warning Interrupt Control register (EWCTRL)

Enable-protected bits in the CTRLA register can be written at the same time as CTRLA.ENABLE is written to '1', but not at the same time as CTRLA.ENABLE is written to '0'.

The WDT can be configured only while the WDT is disabled. The WDT is configured by defining the required Time-Out Period bits in the Configuration register (CONFIG.PER). If Window mode operation is desired, the Window Enable bit in the Control A register must be set (CTRLA.WEN=1) and the Window Period bits in the Configuration register (CONFIG.WINDOW) must be defined.

Enable-protection is denoted by the "Enable-Protected" property in the register description.

24.6.2.2. Configurable Reset Values

After a Power-on Reset, some registers will be loaded with initial values from the NVM User Row.

This includes the following bits and bit groups:

- Enable bit in the Control A register, CTRLA.ENABLE
- Always-On bit in the Control A register, CTRLA.ALWAYSON
- Watchdog Timer Windows Mode Enable bit in the Control A register, CTRLA.WEN
- Watchdog Timer Windows Mode Time-Out Period bits in the Configuration register, CONFIG.WINDOW
- Time-Out Period bits in the Configuration register, CONFIG.PER
- Early Warning Interrupt Time Offset bits in the Early Warning Interrupt Control register, EWCTRL.EWOFFSET

24.6.2.3. Enabling, Disabling, and Resetting

The WDT is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The WDT is disabled by writing a '0' to CTRLA.ENABLE.

The WDT can be disabled only if the Always-On bit in the Control A register (CTRLA.ALWAYSON) is '0'.

24.6.2.4. Normal Mode

In Normal mode operation, the length of a time-out period is configured in CONFIG.PER. The WDT is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). Once enabled, the

WDT will issue a system reset if a time-out occurs. This can be prevented by clearing the WDT at any time during the time-out period.

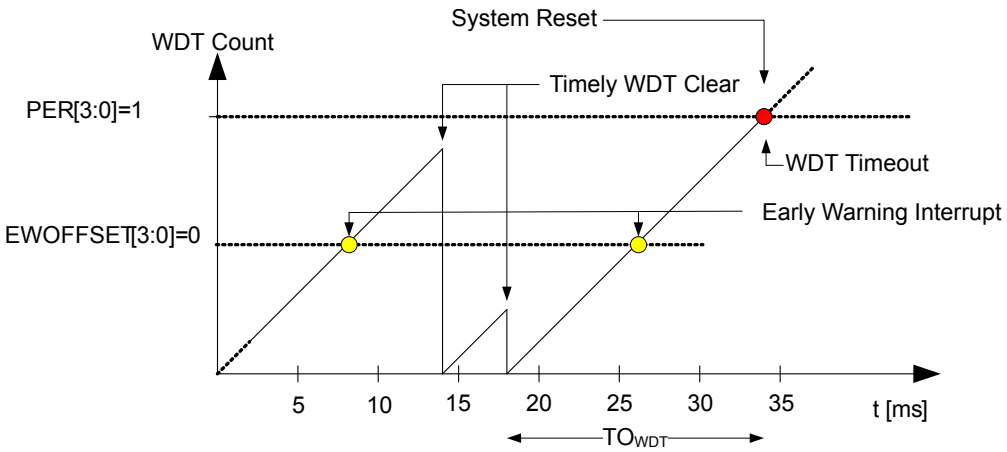
The WDT is cleared and a new WDT time-out period is started by writing 0xA5 to the Clear register (CLEAR). Writing any other value than 0xA5 to CLEAR will issue an immediate system reset.

There are 12 possible WDT time-out (TO_{WDT}) periods, selectable from 8ms to 16s.

By default, the early warning interrupt is disabled. If it is desired, the Early Warning Interrupt Enable bit in the Interrupt Enable register (INTENSET.EW) must be written to '1'. The Early Warning Interrupt is disabled again by writing a '1' to the Early Warning Interrupt bit in the Interrupt Enable Clear register (INTENCLR.EW).

If the Early Warning Interrupt is enabled, an interrupt is generated prior to a WDT time-out condition. In Normal mode, the Early Warning Offset bits in the Early Warning Interrupt Control register, EWCTRL.EWOFFSET, define the time when the early warning interrupt occurs. The Normal mode operation is illustrated in the figure Normal-Mode Operation.

Figure 24-2. Normal-Mode Operation



24.6.2.5. Window Mode

In Window mode operation, the WDT uses two different time specifications: the WDT can only be cleared by writing 0xA5 to the CLEAR register *after* the closed window time-out period (TO_{WDTW}), during the subsequent Normal time-out period (TO_{WDT}). If the WDT is cleared before the time window opens (before TO_{WDTW} is over), the WDT will issue a system reset.

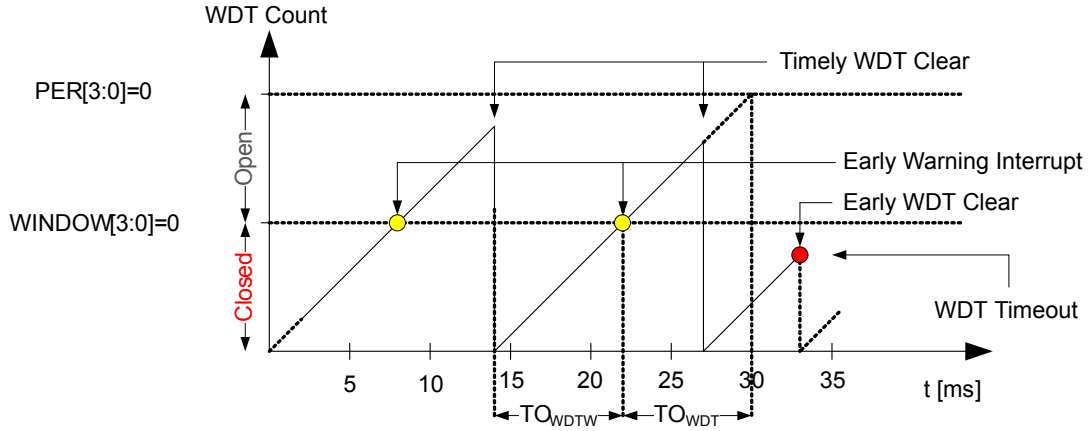
Both parameters TO_{WDTW} and TO_{WDT} are periods in a range from 8ms to 16s, so the total duration of the WDT time-out period is the sum of the two parameters.

The closed window period is defined by the Window Period bits in the Configuration register (CONFIG.WINDOW), and the open window period is defined by the Period bits in the Configuration register (CONFIG.PER).

By default, the Early Warning interrupt is disabled. If it is desired, the Early Warning Interrupt Enable bit in the Interrupt Enable register (INTENSET.EW) must be written to '1'. The Early Warning Interrupt is disabled again by writing a '1' to the Early Warning Interrupt bit in the Interrupt Enable Clear (INTENCLR.EW) register.

If the Early Warning interrupt is enabled in Window mode, the interrupt is generated at the start of the open window period, i.e. after TO_{WDTW}. The Window mode operation is illustrated in figure Window-Mode Operation.

Figure 24-3. Window-Mode Operation



24.6.3. DMA Operation

Not applicable.

24.6.4. Interrupts

The WDT has the following interrupt source:

- Early Warning (EW): Indicates that the counter is approaching the time-out condition.
  - This interrupt is an asynchronous wake-up source.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs.

Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register.

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the WDT is reset. See the INTFLAG on page 337 register description for details on how to clear interrupt flags. All interrupt requests from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. The user must read the INTFLAG register to determine which interrupt condition is present.

Note: Interrupts must be globally enabled for interrupt requests to be generated.

Related Links

Nested Vector Interrupt Controller on page 51

Overview on page 51

PM – Power Manager on page 186

Sleep Mode Controller on page 191

24.6.5. Events

Not applicable.

24.6.6. Sleep Mode Operation

The WDT will continue to operate in any sleep mode where the source clock is active except backup mode. The WDT interrupts can be used to wake up the device from a sleep mode. An interrupt request will be generated after the wake-up if the Interrupt Controller is configured accordingly. Otherwise the

CPU will wake up directly, without triggering an interrupt. In this case, the CPU will continue executing from the instruction following the entry into sleep.

24.6.7. Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read. An exception is the Channel Enable bit in the Peripheral Channel Control registers (PCHCTRLm.CHEN). When changing this bit, the bit value must be read-back to ensure the synchronization is complete and to assert glitch free internal operation. Note that changing the bit value under ongoing synchronization will *not* generate an error.

The following registers are synchronized when written:

- Enable bit in Control A register (CTRLA.ENABLE)
- Window Enable bit in Control A register (CTRLA.WEN)
- Always-On bit in control Control A (CTRLA.ALWAYSON)

The following registers are synchronized when read:

- Watchdog Clear register (CLEAR)

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

Required read-synchronization is denoted by the "Read-Synchronized" property in the register description.

24.6.8. Additional Features

24.6.8.1. Always-On Mode

The Always-On mode is enabled by setting the Always-On bit in the Control A register (CTRLA.ALWAYSON=1). When the Always-On mode is enabled, the WDT runs continuously, regardless of the state of CTRLA.ENABLE. Once written, the Always-On bit can only be cleared by a power-on reset. The Configuration (CONFIG) and Early Warning Control (EWCTRL) registers are read-only registers while the CTRLA.ALWAYSON bit is set. Thus, the time period configuration bits (CONFIG.PER, CONFIG.WINDOW, EWCTRL.EWOFFSET) of the WDT cannot be changed.

Enabling or disabling Window mode operation by writing the Window Enable bit (CTRLA.WEN) is allowed while in Always-On mode, but note that CONFIG.PER cannot be changed.

The Interrupt Clear and Interrupt Set registers are accessible in the Always-On mode. The Early Warning interrupt can still be enabled or disabled while in the Always-On mode, but note that EWCTRL.EWOFFSET cannot be changed.

Table WDT Operating Modes With Always-On shows the operation of the WDT for CTRLA.ALWAYSON=1.

Table 24-2. WDT Operating Modes With Always-On

WEN	Interrupt Enable	Mode
0	0	Always-on and normal mode
0	1	Always-on and normal mode with Early Warning interrupt
1	0	Always-on and window mode
1	1	Always-on and window mode with Early Warning interrupt

24.6.8.2. Early Warning

The Early Warning interrupt notifies that the WDT is approaching its time-out condition. The Early Warning interrupt behaves differently in Normal mode and in Window mode.

In Normal mode, the Early Warning interrupt generation is defined by the Early Warning Offset in the Early Warning Control register (EWCTRL.EWOFFSET). The Early Warning Offset bits define the number of CLK_WDT_OSC clocks before the interrupt is generated, relative to the start of the watchdog time-out period.

The user must take caution when programming the Early Warning Offset bits. If these bits define an Early Warning interrupt generation time greater than the watchdog time-out period, the watchdog time-out system reset is generated prior to the Early Warning interrupt. Consequently, the Early Warning interrupt will never be generated.

In window mode, the Early Warning interrupt is generated at the start of the open window period. In a typical application where the system is in sleep mode, the Early Warning interrupt can be used to wake up and clear the Watchdog Timer, after which the system can perform other tasks or return to sleep mode.

> If the WDT is operating in Normal mode with CONFIG.PER = 0x2 and EWCTRL.EWOFFSET = 0x1, the Early Warning interrupt is generated 16 CLK_WDT_OSC clock cycles after the start of the time-out period. The time-out system reset is generated 32 CLK_WDT_OSC clock cycles after the start of the watchdog time-out period.

24.7. Register Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0	ALWAYSON					WEN	ENABLE	
0x01	CONFIG	7:0	WINDOW[3:0]				PER[3:0]			
0x02	EWCTRL	7:0					EWOFFSET[3:0]			
0x03	Reserved									
0x04	INTENCLR	7:0								EW
0x05	INTENSET	7:0								EW
0x06	INTFLAG	7:0								EW
0x07	Reserved									
0x08	SYNCBUSY	7:0				CLEAR	ALWAYSON	WEN	ENABLE	
0x09		15:8								
0x0A		23:16								
0x0B		31:24								
0x0C	CLEAR	7:0	CLEAR[7:0]							

24.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection on page 323.

Some registers are synchronized when read and/or written. Synchronization is denoted by the "Write-Synchronized" or the "Read-Synchronized" property in each individual register description. For details, refer to Synchronization on page 327.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

24.8.1. Control A

Name: CTRLA
Offset: 0x00
Reset: N/A - Loaded from NVM User Row at startup
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	ALWAYSON					WEN	ENABLE	
Access	R/W					R/W	R/W	
Reset	-					-	-	

Bit 7 – ALWAYSON: Always-On
This bit allows the WDT to run continuously. After being set, this bit cannot be written to '0', and the WDT will remain enabled until a power-on Reset is received. When this bit is '1', the Control A register (CTRLA), the Configuration register (CONFIG) and the Early Warning Control register (EWCTRL) will be read-only, and any writes to these registers are not allowed.

Writing a '0' to this bit has no effect.

This bit is not Enable-Protected.

This bit is loaded from NVM User Row at startup.

Value	Description
0	The WDT is enabled and disabled through the ENABLE bit.
1	The WDT is enabled and can only be disabled by a power-on reset (POR).

Bit 2 – WEN: Watchdog Timer Window Mode Enable
This bit enables Window mode. It can only be written if the peripheral is disabled unless CTRLA.ALWAYSON=1. The initial value of this bit is loaded from Flash Calibration.

This bit is loaded from NVM User Row at startup.

Value	Description
0	Window mode is disabled (normal operation).
1	Window mode is enabled.

Bit 1 – ENABLE: Enable
This bit enables or disables the WDT. It can only be written if CTRLA.ALWAYSON=0.

Due to synchronization, there is delay between writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately, and the Enable bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

This bit is not Enable-Protected.

This bit is loaded from NVM User Row at startup.

Value	Description
0	The WDT is disabled.
1	The WDT is enabled.

24.8.2. Configuration

Name: CONFIG
Offset: 0x01
Reset: Loaded from NVM User Row at startup
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
	WINDOW[3:0]				PER[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	-	-	-	-	-	-	-	-

Bits 7:4 – WINDOW[3:0]: Window Mode Time-Out Period
In Window mode, these bits determine the watchdog closed window period as a number of cycles of the 1.024kHz CLK_WDT_OSC clock.

These bits are loaded from NVM User Row at startup.

Value	Name	Description
0x0	CYC8	8 clock cycles
0x1	CYC16	16 clock cycles
0x2	CYC32	32 clock cycles
0x3	CYC64	64 clock cycles
0x4	CYC128	128 clock cycles
0x5	CYC256	256 clock cycles
0x6	CYC512	512 clock cycles
0x7	CYC1024	1024 clock cycles
0x8	CYC2048	2048 clock cycles
0x9	CYC4096	4096 clock cycles
0xA	CYC8192	8192 clock cycles
0xB	CYC16384	16384 clock cycles
0xC - 0xF	-	Reserved

Bits 3:0 – PER[3:0]: Time-Out Period
These bits determine the watchdog time-out period as a number of 1.024kHz CLK_WDTOSC clock cycles. In Window mode operation, these bits define the open window period.

These bits are loaded from NVM User Row at startup.

Value	Name	Description
0x0	CYC8	8 clock cycles
0x1	CYC16	16 clock cycles
0x2	CYC32	32 clock cycles

Value	Name	Description
0x3	CYC64	64 clock cycles
0x4	CYC128	128 clock cycles
0x5	CYC256	256 clock cycles
0x6	CYC512	512 clock cycles
0x7	CYC1024	1024 clock cycles
0x8	CYC2048	2048 clock cycles
0x9	CYC4096	4096 clock cycles
0xA	CYC8192	8192 clock cycles
0xB	CYC16384	16384 clock cycles
0xC - 0xF	-	Reserved

24.8.3. Early Warning Control

Name: EWCTRL
Offset: 0x02
Reset: N/A - Loaded from NVM User Row at startup
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
					EWOFFSET[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					-	-	-	-

Bits 3:0 – EWOFFSET[3:0]: Early Warning Interrupt Time Offset
These bits determine the number of GCLK_WDT clock cycles between the start of the watchdog time-out period and the generation of the Early Warning interrupt. These bits are loaded from NVM User Row at startup.

Value	Name	Description
0x0	CYC8	8 clock cycles
0x1	CYC16	16 clock cycles
0x2	CYC32	32 clock cycles
0x3	CYC64	64 clock cycles
0x4	CYC128	128 clock cycles
0x5	CYC256	256 clock cycles
0x6	CYC512	512 clock cycles
0x7	CYC1024	1024 clock cycles
0x8	CYC2048	2048 clock cycles
0x9	CYC4096	4096 clock cycles
0xA	CYC8192	8192 clock cycles
0xB	CYC16384	16384 clock cycles
0xC - 0xF	-	Reserved

24.8.4. Interrupt Enable Clear

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Name: INTENCLR
Offset: 0x04
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								EW
Access								R/W
Reset								0

Bit 0 – EW: Early Warning Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Early Warning Interrupt Enable bit, which disables the Early Warning interrupt.

Value	Description
0	The Early Warning interrupt is disabled.
1	The Early Warning interrupt is enabled.

24.8.5. Interrupt Enable Set

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Name: INTENSET
Offset: 0x05
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								EW
Access								R/W
Reset								0

Bit 0 – EW: Early Warning Interrupt Enable

Writing a '0' to this bit has no effect.
Writing a '1' to this bit sets the Early Warning Interrupt Enable bit, which enables the Early Warning interrupt.

Value	Description
0	The Early Warning interrupt is disabled.
1	The Early Warning interrupt is enabled.

24.8.6. Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x06
Reset: 0x00
Property: N/A

Bit	7	6	5	4	3	2	1	0
								EW
Access								R/W
Reset								0

Bit 0 – EW: Early Warning
This flag is cleared by writing a '1' to it.

This flag is set when an Early Warning interrupt occurs, as defined by the EWOFFSET bit group in EWCTRL.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Early Warning interrupt flag.

24.8.7. Synchronization Busy

Name: SYNCBUSY
Offset: 0x08
Reset: 0x00000000
Property: Read-Only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				CLEAR	ALWAYSON	WEN	ENABLE	
Access				R	R	R	R	
Reset				0	0	0	0	

Bit 4 – CLEAR: Clear Synchronization Busy

Value	Description
0	Write synchronization of the CLEAR register is complete.
1	Write synchronization of the CLEAR register is ongoing.

Bit 3 – ALWAYSON: Always-On Synchronization Busy

Value	Description
0	Write synchronization of the CTRLA.ALWAYSON bit is complete.
1	Write synchronization of the CTRLA.ALWAYSON bit is ongoing.

Bit 2 – WEN: Window Enable Synchronization Busy

Value	Description
0	Write synchronization of the CTRLA.WEN bit is complete.
1	Write synchronization of the CTRLA.WEN bit is ongoing.

Bit 1 – ENABLE: Enable Synchronization Busy

Value	Description
0	Write synchronization of the CTRLA.ENABLE bit is complete.
1	Write synchronization of the CTRLA.ENABLE bit is ongoing.

24.8.8. Clear

Name: CLEAR
Offset: 0x0C
Reset: 0x00
Property: Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	CLEAR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – CLEAR[7:0]: Watchdog Clear
In Normal mode, writing 0xA5 to this register during the watchdog time-out period will clear the Watchdog Timer and the watchdog time-out period is restarted.

In Window mode, any writing attempt to this register before the time-out period started (i.e., during TO_{WDTW}) will issue an immediate system Reset. Writing 0xA5 during the time-out period TO_{WDT} will clear the Watchdog Timer and the complete time-out sequence (first TO_{WDTW} then TO_{WDT}) is restarted.

In both modes, writing any other value than 0xA5 will issue an immediate system Reset.

25. RTC – Real-Time Counter

25.1. Overview

The Real-Time Counter (RTC) is a 32-bit counter with a 10-bit programmable prescaler that typically runs continuously to keep track of time. The RTC can wake up the device from sleep modes using the alarm/compare wake up, periodic wake up, or overflow wake up mechanisms.

The RTC can generate periodic peripheral events from outputs of the prescaler, as well as alarm/compare interrupts and peripheral events, which can trigger at any counter value. Additionally, the timer can trigger an overflow interrupt and peripheral event, and can be reset on the occurrence of an alarm/compare match. This allows periodic interrupts and peripheral events at very long and accurate intervals.

The 10-bit programmable prescaler can scale down the clock source. By this, a wide range of resolutions and time-out periods can be configured. With a 32.768kHz clock source, the minimum counter tick interval is 30.5µs, and time-out periods can range up to 36 hours. For a counter tick interval of 1s, the maximum time-out period is more than 136 years.

25.2. Features

- 32-bit counter with 10-bit prescaler
- Multiple clock sources
- 32-bit or 16-bit counter mode
- Clock/Calendar mode
  - Time in seconds, minutes, and hours (12/24)
  - Date in day of month, month, and year
  - Leap year correction
- Digital prescaler correction/tuning for increased accuracy
- Overflow, alarm/compare match and prescaler interrupts and events
  - Optional clear on alarm/compare match
- 4 general purpose registers

25.3. Block Diagram

Figure 25-1. RTC Block Diagram (Mode 0 — 32-Bit Counter)

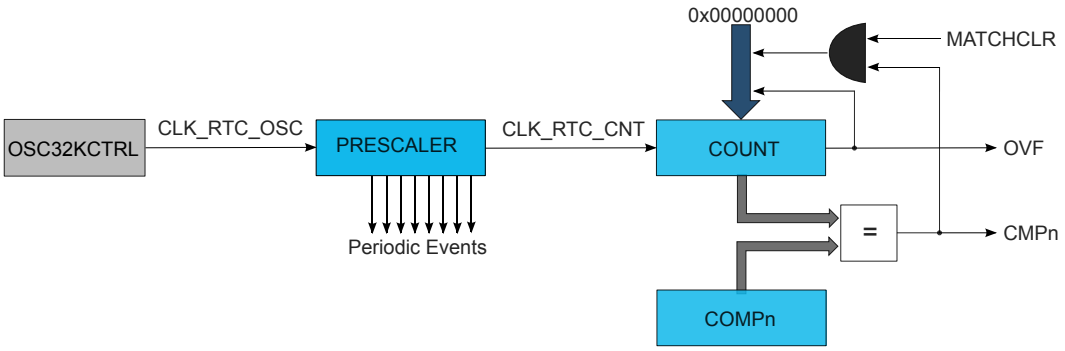


Figure 25-2. RTC Block Diagram (Mode 1 — 16-Bit Counter)

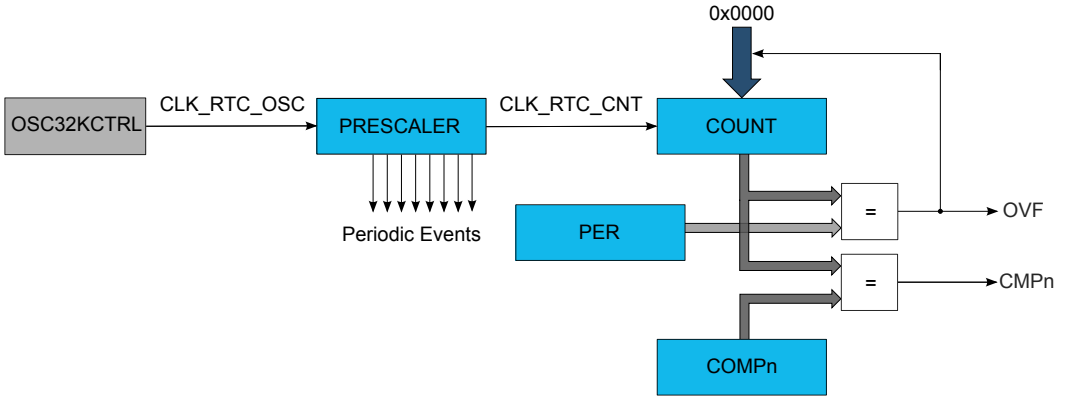
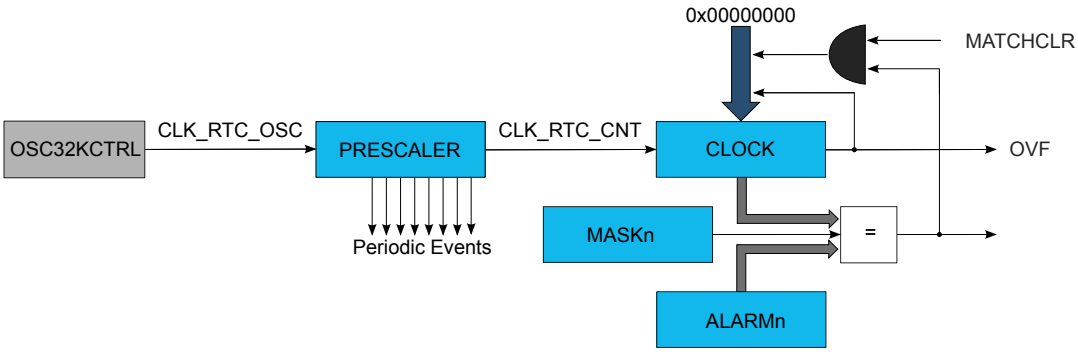


Figure 25-3. RTC Block Diagram (Mode 2 — Clock/Calendar)



25.4. Signal Description

Not applicable.

25.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

25.5.1. I/O Lines

Not applicable.

25.5.2. Power Management

The RTC will continue to operate in any sleep mode where the selected source clock is running. The RTC interrupts can be used to wake up the device from sleep modes. Events connected to the event system can trigger other operations in the system without exiting sleep modes. Refer to the *Power Manager* for details on the different sleep modes.

The RTC will be reset only at power-on (POR) or by setting the Software Reset bit in the Control A register (CTRLA.SWRST=1).

Related Links

PM – Power Manager on page 186

PM – Power Manager on page 186

25.5.3. Clocks

The RTC bus clock (CLK_RTC_APB) can be enabled and disabled in the Main Clock module MCLK, and the default state of CLK_RTC_APB can be found in Peripheral Clock Masking section.

A 32KHz or 1KHz oscillator clock (CLK_RTC_OSC) is required to clock the RTC. This clock must be configured and enabled in the 32KHz oscillator controller (OSC32KCTRL) before using the RTC.

This oscillator clock is asynchronous to the bus clock (CLK_RTC_APB). Due to this asynchronicity, writing to certain registers will require synchronization between the clock domains. Refer to Synchronization on page 347 for further details.

Related Links

OSC32KCTRL – 32KHz Oscillators Controller on page 264

Peripheral Clock Masking on page 151

25.5.4. DMA

Not applicable.

Related Links

DMAC – Direct Memory Access Controller on page 399

25.5.5. Interrupts

The interrupt request line is connected to the Interrupt Controller. Using the RTC interrupt requires the Interrupt Controller to be configured first.

Related Links

Nested Vector Interrupt Controller on page 51

25.5.6. Events

The events are connected to the *Event System*.

Related Links

EVSYS – Event System on page 536

25.5.7. Debug Operation

When the CPU is halted in debug mode the RTC will halt normal operation. The RTC can be forced to continue operation during debugging. Refer to DBGCTRL on page 358 for details.

25.5.8. Register Access Protection

All registers with write-access are optionally write-protected by the peripheral access controller (PAC), except the following registers:

- Interrupt Flag Status and Clear (INTFLAG) register
- General Purpose (GPx) registers

Write-protection is denoted by the "PAC Write-Protection" property in the register description.

Write-protection does not apply to accesses through an external debugger. Refer to the PAC - Peripheral Access Controller for details.

Related Links

PAC - Peripheral Access Controller on page 58

25.5.9. Analog Connections

A 32.768kHz crystal can be connected to the XIN32 and XOUT32 pins, along with any required load capacitors. For details on recommended crystal characteristics and load capacitors.

25.6. Functional Description

25.6.1. Principle of Operation

The RTC keeps track of time in the system and enables periodic events, as well as interrupts and events at a specified time. The RTC consists of a 10-bit prescaler that feeds a 32-bit counter. The actual format of the 32-bit counter depends on the RTC operating mode.

The RTC can function in one of these modes:

- Mode 0 - COUNT32: RTC serves as 32-bit counter
- Mode 1 - COUNT16: RTC serves as 16-bit counter
- Mode 2 - CLOCK: RTC serves as clock/calendar with alarm functionality

25.6.2. Basic Operation

25.6.2.1. Initialization

The following bits are enable-protected, meaning that they can only be written when the RTC is disabled (CTRLA.ENABLE=0):

- Operating Mode bits in the Control A register (CTRLA.MODE)
- Prescaler bits in the Control A register (CTRLA.PRESCALER)
- Clear on Match bit in the Control A register (CTRLA.MATCHCLR)
- Clock Representation bit in the Control A register (CTRLA.CLKREP)

The following register is enable-protected

- Event Control register (EVCTRL)

Enable-protected bits and registers can be changed only when the RTC is disabled (CTRLA.ENABLE=0). If the RTC is enabled (CTRLA.ENABLE=1), these operations are necessary: first write CTRLA.ENABLE=0 and check whether the write synchronization has finished, then change the desired bit field value. Enable-protected bits in can be written at the same time as CTRLA.ENABLE is written to '1', but not at the same time as CTRLA.ENABLE is written to '0'.

Enable-protection is denoted by the "Enable-Protected" property in the register description.

The RTC prescaler divides the source clock for the RTC counter.

Note: In Clock/Calendar mode, the prescaler must be configured to provide a 1Hz clock to the counter for correct operation.

The frequency of the RTC clock (CLK_RTC_CNT) is given by the following formula:

$$f_{\mathrm{CLK\_RTC\_CNT}} = \frac{f_{\mathrm{CLK\_RTC\_OSC}}}{2^{\mathrm{PRESCALER}}}$$

The frequency of the oscillator clock, CLK_RTC_OSC, is given by $f_{CLK\_RTC\_OSC}$, and $f_{CLK\_RTC\_CNT}$ is the frequency of the internal prescaled RTC clock, CLK_RTC_CNT.

25.6.2.2. Enabling, Disabling, and Resetting

The RTC is enabled by setting the Enable bit in the Control A register (CTRLA.ENABLE=1). The RTC is disabled by writing CTRLA.ENABLE=0.

The RTC is reset by setting the Software Reset bit in the Control A register (CTRLA.SWRST=1). All registers in the RTC, except DEBUG, will be reset to their initial state, and the RTC will be disabled. The RTC must be disabled before resetting it.

25.6.2.3. 32-Bit Counter (Mode 0)

When the RTC Operating Mode bits in the Control A register are zero (CTRLA.MODE=00), the counter operates in 32-bit Counter mode. The block diagram of this mode is shown in Figure 25-1 RTC Block Diagram (Mode 0 — 32-Bit Counter) on page 341. When the RTC is enabled, the counter will increment on every 0-to-1 transition of CLK_RTC_CNT. The counter will increment until it reaches the top value of 0xFFFFFFFF, and then wrap to 0x00000000. This sets the Overflow Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.OVF).

The RTC counter value can be read from or written to the Counter Value register (COUNT) in 32-bit format.

The counter value is continuously compared with the 32-bit Compare register (COMP0). When a compare match occurs, the Compare 0 Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.CMP0) is set on the next 0-to-1 transition of CLK_RTC_CNT.

If the Clear on Match bit in the Control A register (CTRLA.MATCHCLR) is '1', the counter is cleared on the next counter cycle when a compare match with COMP0 occurs. This allows the RTC to generate periodic interrupts or events with longer periods than the prescaler events. Note that when CTRLA.MATCHCLR is '1', INTFLAG.CMP0 and INTFLAG.OVF will both be set simultaneously on a compare match with COMP0.

25.6.2.4. 16-Bit Counter (Mode 1)

When the RTC Operating Mode bits in the Control A register (CTRLA.MODE) are 1, the counter operates in 16-bit Counter mode as shown in Figure 25-2 RTC Block Diagram (Mode 1 — 16-Bit Counter) on page 342. When the RTC is enabled, the counter will increment on every 0-to-1 transition of CLK_RTC_CNT. In 16-bit Counter mode, the 16-bit Period register (PER) holds the maximum value of the counter. The counter will increment until it reaches the PER value, and then wrap to 0x0000. This sets the Overflow Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.OVF).

The RTC counter value can be read from or written to the Counter Value register (COUNT) in 16-bit format.

The counter value is continuously compared with the 16-bit Compare registers (COMPn, n=0..1). When a compare match occurs, the Compare n Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.CMPn, n=0..1) is set on the next 0-to-1 transition of CLK_RTC_CNT.

25.6.2.5. Clock/Calendar (Mode 2)

When the RTC Operating Mode bit field in the Control A register (CTRLA.MODE) is '2', the counter operates in Clock/Calendar mode, as shown in Figure 25-3 RTC Block Diagram (Mode 2 — Clock/Calendar) on page 342. When the RTC is enabled, the counter will increment on every 0-to-1 transition of CLK_RTC_CNT. The selected clock source and RTC prescaler must be configured to provide a 1Hz clock to the counter for correct operation in this mode.

The time and date can be read from or written to the Clock Value register (CLOCK) in a 32-bit time/date format. Time is represented as:

- Seconds
- Minutes
- Hours

Hours can be represented in either 12- or 24-hour format, selected by the Clock Representation bit in the Control A register (CTRLA.CLKREP). This bit can be changed only while the RTC is disabled.

Date is represented as:

- Day as the numeric day of the month (starting at 1)
- Month as the numeric month of the year (1 = January, 2 = February, etc.)
- Year as a value counting the offset from a reference value that must be defined in software

The date is automatically adjusted for leap years, assuming every year divisible by 4 is a leap year. Therefore, the reference value must be a leap year, e.g. 2000. The RTC will increment until it reaches the top value of 23:59:59 December 31 of year 63, and then wrap to 00:00:00 January 1 of year 0. This will set the Overflow Interrupt flag in the Interrupt Flag Status and Clear registers (INTFLAG.OVF).

The clock value is continuously compared with the 32-bit Alarm register (ALARM0). When an alarm match occurs, the Alarm 0 Interrupt flag in the Interrupt Flag Status and Clear registers (INTFLAG.ALARM0) is set on the next 0-to-1 transition of CLK_RTC_CNT.

A valid alarm match depends on the setting of the Alarm Mask Selection bits in the Alarm 0 Mask register (MASK0.SEL). These bits determine which time/date fields of the clock and alarm values are valid for comparison and which are ignored.

If the Clear on Match bit in the Control A register (CTRLA.MATCHCLR) is set, the counter is cleared on the next counter cycle when an alarm match with ALARM0 occurs. This allows the RTC to generate periodic interrupts or events with longer periods than it would be possible with the prescaler events only (see Periodic Intervals on page 348).
Note: When CTRLA.MATCHCLR is 1, INTFLAG.ALARM0 and INTFLAG.OVF will both be set simultaneously on an alarm match with ALARM0.

25.6.3. DMA Operation

Not applicable.

25.6.4. Interrupts

The RTC has the following interrupt sources:

- Overflow (OVF): Indicates that the counter has reached its top value and wrapped to zero.
- Compare (CMPn): Indicates a match between the counter value and the compare register.
- Alarm (ALARM): Indicates a match between the clock value and the alarm register.
- Period n (PERn): The corresponding bit in the prescaler has toggled. Refer to Periodic Intervals on page 348 for details.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. Each interrupt can be individually enabled by setting the corresponding bit in the Interrupt Enable Set register (INTENSET=1), and disabled by setting the corresponding bit in the Interrupt Enable Clear register (INTENCLR=1).

An interrupt request is generated when the interrupt flag is raised and the corresponding interrupt is enabled. The interrupt request remains active until either the interrupt flag is cleared, the interrupt is disabled or the RTC is reset. See the description of the INTFLAG registers for details on how to clear interrupt flags.

All interrupt requests from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. Refer to the Nested Vector Interrupt Controller for details. The user must read the INTFLAG register to determine which interrupt condition is present.

Note: Interrupts must be globally enabled for interrupt requests to be generated. Refer to the Nested Vector Interrupt Controller for details.

Related Links

Nested Vector Interrupt Controller on page 51

25.6.5. Events

The RTC can generate the following output events:

- Overflow (OVF): Generated when the counter has reached its top value and wrapped to zero.
- Compare (CMPn): Indicates a match between the counter value and the compare register.
- Alarm (ALARM): Indicates a match between the clock value and the alarm register.
- Period n (PERn): The corresponding bit in the prescaler has toggled. Refer to Periodic Intervals on page 348 for details.

Setting the Event Output bit in the Event Control Register (EVCTRL.xxxEO=1) enables the corresponding output event. Writing a zero to this bit disables the corresponding output event. Refer to the EVSYS - Event System for details on configuring the event system.

Related Links

EVSYS – Event System on page 536

25.6.6. Sleep Mode Operation

The RTC will continue to operate in any sleep mode where the source clock is active. The RTC *interrupts* can be used to wake up the device from a sleep mode. RTC *events* can trigger other operations in the system without exiting the sleep mode.

An interrupt request will be generated after the wake-up if the Interrupt Controller is configured accordingly. Otherwise the CPU will wake up directly, without triggering any interrupt. In this case, the CPU will continue executing right from the first instruction that followed the entry into sleep.

25.6.7. Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read. An exception is the Channel Enable bit in the Peripheral Channel Control registers (PCHCTRLm.CHEN). When changing this bit, the bit value must be read-back to ensure the synchronization is complete and to assert glitch free internal operation. Note that changing the bit value under ongoing synchronization will *not* generate an error.

The following bits are synchronized when written:

- Software Reset bit in Control A register, CTRLA.SWRST
- Enable bit in Control A register, CTRLA.ENABLE

The following registers are synchronized when written:

- Counter Value register, COUNT
- Clock Value register, CLOCK
- Counter Period register, PER
- Compare n Value registers, COMPn
- Alarm n Value registers, ALARMn
- Frequency Correction register, FREQCORR
- Alarm n Mask register, MASKn

The following registers are synchronized when read:

- The Counter Value register, COUNT, if the Counter Read Sync Enable bit in CTRLA (CTRLA.COUNTSYNC) is '1'
- The Clock Value register, CLOCK, if the Clock Read Sync Enable bit in CTRLA (CTRLA.CLOCKSYNC) is '1'

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

Required read-synchronization is denoted by the "Read-Synchronized" property in the register description.

Related Links

Register Synchronization on page 127

25.6.8. Additional Features

25.6.8.1. Periodic Intervals

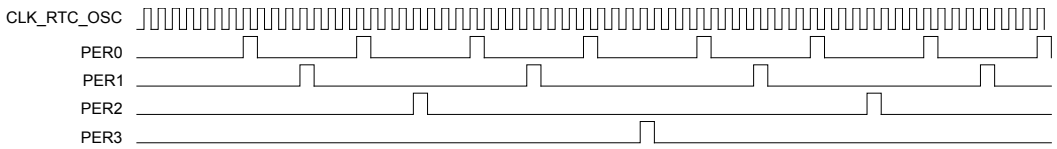
The RTC prescaler can generate interrupts and events at periodic intervals, allowing flexible system tick creation. Any of the upper eight bits of the prescaler (bits 2 to 9) can be the source of an interrupt/event. When one of the eight Periodic Event Output bits in the Event Control register (EVCTRL.PEREO[n=0..7]) is '1', an event is generated on the 0-to-1 transition of the related bit in the prescaler, resulting in a periodic event frequency of:

$$f_{PERIODIC(n)} = \frac{f_{CLK\_RTC\_OSC}}{2^{n+3}}$$

$f_{CLK\_RTC\_OSC}$ is the frequency of the internal prescaler clock CLK_RTC_OSC, and n is the position of the EVCTRL.PEREOn bit. For example, PER0 will generate an event every eight CLK_RTC_OSC cycles, PER1 every 16 cycles, etc. This is shown in the figure below.

Periodic events are independent of the prescaler setting used by the RTC counter, except if CTRLA.PRESCALER is zero. Then, no periodic events will be generated.

Figure 25-4. Example Periodic Events



25.6.8.2. Frequency Correction

The RTC Frequency Correction module employs periodic counter corrections to compensate for a too-slow or too-fast oscillator. Frequency correction requires that CTRLA.PRESCALER is greater than 1.

The digital correction circuit adds or subtracts cycles from the RTC prescaler to adjust the frequency in approximately 1ppm steps. Digital correction is achieved by adding or skipping a single count in the prescaler once every 4096 CLK_RTC_OSC cycles. The Value bit group in the Frequency Correction register (FREQCORR.VALUE) determines the number of times the adjustment is applied over 240 of these periods. The resulting correction is as follows:

$$\text{Correction in ppm} = \frac{\text{FREQCORR.VALUE}}{4096 \cdot 240} \cdot 10^6 \text{ppm}$$

This results in a resolution of 1.017ppm.

The Sign bit in the Frequency Correction register (FREQCORR.SIGN) determines the direction of the correction. A positive value will add counts and increase the period (reducing the frequency), and a negative value will reduce counts per period (speeding up the frequency).

Digital correction also affects the generation of the periodic events from the prescaler. When the correction is applied at the end of the correction cycle period, the interval between the previous periodic event and the next occurrence may also be shortened or lengthened depending on the correction value.

25.6.8.3. General Purpose Registers

The RTC includes up to eight General Purpose registers (GPx). These registers are reset only when the RTC is reset and remain powered while the RTC is powered. They can be used to store user-defined values while other parts of the system are powered off.

25.7. Register Summary - COUNT32

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0	MATCHCLR				MODE[1:0]		ENABLE	SWRST
0x01		15:8	COUNTSYNC				PRESCALER[3:0]			
0x02 ... 0x03	Reserved									
0x04	EVCTRL	7:0	PEREO7	PEREO6	PEREO5	PEREO4	PEREO3	PEREO2	PEREO1	PEREO0
0x05		15:8	OVFEO							CMPEO0
0x06		23:16								
0x07		31:24								
0x08	INTENCLR	7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
0x09		15:8	OVF							CMP0
0x0A	INTENSET	7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
0x0B		15:8	OVF							CMP0
0x0C	INTFLAG	7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
0x0D		15:8	OVF							CMP0
0x0E	DBGCTRL	7:0								DBGRUN
0x0F	Reserved									
0x10	SYNCBUSY	7:0			COMP0		COUNT	FREQCORR	ENABLE	SWRST
0x11		15:8	COUNTSYNC							
0x12		23:16								
0x13		31:24								
0x14	FREQCORR	7:0	SIGN		VALUE[5:0]					
0x15 ... 0x17	Reserved									
0x18	COUNT	7:0	COUNT[7:0]							
0x19		15:8	COUNT[15:8]							
0x1A		23:16	COUNT[23:16]							
0x1B		31:24	COUNT[31:24]							
0x1C ... 0x1F	Reserved									
0x20	COMP0	7:0	COMP[7:0]							
0x21		15:8	COMP[15:8]							
0x22		23:16	COMP[23:16]							
0x23		31:24	COMP[31:24]							
0x24	COMP1	7:0	COMP[7:0]							
0x25		15:8	COMP[15:8]							
0x26		23:16	COMP[23:16]							
0x27		31:24	COMP[31:24]							
0x28	COMP2	7:0	COMP[7:0]							
0x29		15:8	COMP[15:8]							
0x2A		23:16	COMP[23:16]							
0x2B		31:24	COMP[31:24]							

Offset	Name	Bit Pos.								
0x2C	COMP3	7:0	COMP[7:0]							
0x2D		15:8	COMP[15:8]							
0x2E		23:16	COMP[23:16]							
0x2F		31:24	COMP[31:24]							
0x30 ... 0x3F	Reserved									
0x40	GP0	7:0	GP[7:0]							
0x41		15:8	GP[15:8]							
0x42		23:16	GP[23:16]							
0x43		31:24	GP[31:24]							
0x44	GP1	7:0	GP[7:0]							
0x45		15:8	GP[15:8]							
0x46		23:16	GP[23:16]							
0x47		31:24	GP[31:24]							
0x48	GP2	7:0	GP[7:0]							
0x49		15:8	GP[15:8]							
0x4A		23:16	GP[23:16]							
0x4B		31:24	GP[31:24]							
0x4C	GP3	7:0	GP[7:0]							
0x4D		15:8	GP[15:8]							
0x4E		23:16	GP[23:16]							
0x4F		31:24	GP[31:24]							

25.8. Register Description - COUNT32

This Register Description section is valid if the RTC is in COUNT32 mode (CTRLA.MODE=0).

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

25.8.1. Control A in COUNT32 mode (CTRLA.MODE=0)

Name: CTRLA
Offset: 0x00
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	COUNTSYNC				PRESCALER[3:0]			
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MATCHCLR				MODE[1:0]		ENABLE	SWRST
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit 15 – COUNTSYNC: COUNT Read Synchronization Enable
The COUNT register requires synchronization when reading. Disabling the synchronization will prevent reading valid values from the COUNT register.
This bit is not enable-protected.

Value	Description
0	COUNT read synchronization is disabled
1	COUNT read synchronization is enabled

Bits 11:8 – PRESCALER[3:0]: Prescaler
These bits define the prescaling factor for the RTC clock source (GCLK_RTC) to generate the counter clock (CLK_RTC_CNT). Periodic events and interrupts are not available when the prescaler is off. These bits are not synchronized.

Value	Name	Description
0x0	OFF	CLK_RTC_CNT = GCLK_RTC/1
0x1	DIV1	CLK_RTC_CNT = GCLK_RTC/1
0x2	DIV2	CLK_RTC_CNT = GCLK_RTC/2
0x3	DIV4	CLK_RTC_CNT = GCLK_RTC/4
0x4	DIV8	CLK_RTC_CNT = GCLK_RTC/8
0x5	DIV16	CLK_RTC_CNT = GCLK_RTC/16
0x6	DIV32	CLK_RTC_CNT = GCLK_RTC/32
0x7	DIV64	CLK_RTC_CNT = GCLK_RTC/64
0x8	DIV128	CLK_RTC_CNT = GCLK_RTC/128
0x9	DIV256	CLK_RTC_CNT = GCLK_RTC/256
0xA	DIV512	CLK_RTC_CNT = GCLK_RTC/512

Value	Name	Description
0xB	DIV1024	CLK_RTC_CNT = GCLK_RTC/1024
0xC-0xF	-	Reserved

Bit 7 – MATCHCLR: Clear on Match
This bit defines if the counter is cleared or not on a match.

This bit is not synchronized.

Value	Description
0	The counter is not cleared on a Compare/Alarm 0 match
1	The counter is cleared on a Compare/Alarm 0 match

Bits 3:2 – MODE[1:0]: Operating Mode
This bit group defines the operating mode of the RTC.

This bit is not synchronized.

Value	Name	Description
0x0	COUNT32	Mode 0: 32-bit counter
0x1	COUNT16	Mode 1: 16-bit counter
0x2	CLOCK	Mode 2: Clock/calendar
0x3	-	Reserved

Bit 1 – ENABLE: Enable
Due to synchronization there is a delay between writing CTRLA.ENABLE and until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the Enable bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

Value	Description
0	The peripheral is disabled
1	The peripheral is enabled

Bit 0 – SWRST: Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the RTC (except DBGCTRL) to their initial state, and the RTC will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.

Due to synchronization there is a delay between writing CTRLA.SWRST and until the reset is complete. CTRLA.SWRST will be cleared when the reset is complete.

Value	Description
0	There is not reset operation ongoing
1	The reset operation is ongoing

25.8.2. Event Control in COUNT32 mode (CTRLA.MODE=0)

Name: EVCTRL
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	OVFEO							CMPEO0
Access	R/W							R/W
Reset	0							0

Bit	7	6	5	4	3	2	1	0
	PEREO7	PEREO6	PEREO5	PEREO4	PEREO3	PEREO2	PEREO1	PEREO0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVFEO: Overflow Event Output Enable

Value	Description
0	Overflow event is disabled and will not be generated.
1	Overflow event is enabled and will be generated for every overflow.

Bit 8 – CMPEO0: Compare 0 Event Output Enable

Value	Description
0	Compare 0 event is disabled and will not be generated.
1	Compare 0 event is enabled and will be generated for every compare match.

Bits 7:0 – PEREOn: Periodic Interval n Event Output Enable [n = 7..0]

Value	Description
0	Periodic Interval n event is disabled and will not be generated.
1	Periodic Interval n event is enabled and will be generated.

25.8.3. Interrupt Enable Clear in COUNT32 mode (CTRLA.MODE=0)

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Name: INTENCLR
Offset: 0x08
Reset: 0x0000
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
	OVF							CMP0
Access	R/W							R/W
Reset	0							0

Bit	7	6	5	4	3	2	1	0
	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF: Overflow Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Overflow Interrupt Enable bit, which disables the Overflow interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Bit 8 – CMP0: Compare 0 Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Compare 0 Interrupt Enable bit, which disables the Compare interrupt.

Value	Description
0	The Compare 0 interrupt is disabled.
1	The Compare 0 interrupt is enabled.

Bits 7:0 – PERn: Periodic Interval n Interrupt Enable [n = 7..0]
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Periodic Interval n Interrupt Enable bit, which disables the Periodic Interval n interrupt.

Value	Description
0	Periodic Interval n interrupt is disabled.
1	Periodic Interval n interrupt is enabled.

25.8.4. Interrupt Enable Set in COUNT32 mode (CTRLA.MODE=0)

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Name: INTENSET
Offset: 0x0A
Reset: 0x0000
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
	OVF							CMP0
Access	R/W							R/W
Reset	0							0

Bit	7	6	5	4	3	2	1	0
	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF: Overflow Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Overflow Interrupt Enable bit, which enables the Overflow interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Bit 8 – CMP0: Compare 0 Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Compare 0 Interrupt Enable bit, which enables the Compare 0 interrupt.

Value	Description
0	The Compare 0 interrupt is disabled.
1	The Compare 0 interrupt is enabled.

Bits 7:0 – PERn: Periodic Interval n Interrupt Enable [n = 7..0]
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Periodic Interval n Interrupt Enable bit, which enables the Periodic Interval n interrupt.

Value	Description
0	Periodic Interval n interrupt is disabled.
1	Periodic Interval n interrupt is enabled.

25.8.5. Interrupt Flag Status and Clear in COUNT32 mode (CTRLA.MODE=0)

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Name: INTFLAG
Offset: 0x0C
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
	OVF							CMP0
Access	R/W							R/W
Reset	0							0

Bit	7	6	5	4	3	2	1	0
	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF: Overflow
This flag is cleared by writing a '1' to the flag.

This flag is set on the next CLK_RTC_CNT cycle after an overflow condition occurs, and an interrupt request will be generated if INTENCLR/SET.OVF is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Overflow interrupt flag.

Bit 8 – CMP0: Compare 0
This flag is cleared by writing a '1' to the flag.

This flag is set on the next CLK_RTC_CNT cycle after a match with the compare condition, and an interrupt request will be generated if INTENCLR/SET.COMP0 is one.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Compare 0 interrupt flag.

Bits 7:0 – PERn: Periodic Interval n [n = 7..0]
This flag is cleared by writing a '1' to the flag.

This flag is set on the 0-to-1 transition of prescaler bit [n+2], and an interrupt request will be generated if INTENCLR/SET.PERx is one.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Periodic Interval n interrupt flag.

25.8.6. Debug Control

Name: DBGCTRL
Offset: 0x0E
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGRUN
Access								R/W
Reset								0

Bit 0 – DBGRUN: Debug Run
This bit is not reset by a software reset.

This bit controls the functionality when the CPU is halted by an external debugger.

Value	Description
0	The RTC is halted when the CPU is halted by an external debugger.
1	The RTC continues normal operation when the CPU is halted by an external debugger.

25.8.7. Synchronization Busy in COUNT32 mode (CTRLA.MODE=0)

Name: SYNCBUSY
Offset: 0x10
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	COUNTSYNC							
Access	R							
Reset	0							

Bit	7	6	5	4	3	2	1	0
			COMP0		COUNT	FREQCORR	ENABLE	SWRST
Access			R		R	R	R	R
Reset			0		0	0	0	0

Bit 15 – COUNTSYNC: Count Read Sync Enable Synchronization Busy Status

Value	Description
0	Write synchronization for CTRLA.COUNTSYNC bit is complete.
1	Write synchronization for CTRLA.COUNTSYNC bit is ongoing.

Bit 5 – COMP0: Compare 0 Synchronization Busy Status

Value	Description
0	Write synchronization for COMP0 register is complete.
1	Write synchronization for COMP0 register is ongoing.

Bit 3 – COUNT: Count Value Synchronization Busy Status

Value	Description
0	Read/write synchronization for COUNT register is complete.
1	Read/write synchronization for COUNT register is ongoing.

Bit 2 – FREQCORR: Frequency Correction Synchronization Busy Status

Value	Description
0	Read/write synchronization for FREQCORR register is complete.
1	Read/write synchronization for FREQCORR register is ongoing.

Bit 1 – ENABLE: Enable Synchronization Busy Status

Value	Description
0	Read/write synchronization for CTRLA.ENABLE bit is complete.
1	Read/write synchronization for CTRLA.ENABLE bit is ongoing.

Bit 0 – SWRST: Software Reset Synchronization Busy Status

Value	Description
0	Read/write synchronization for CTRLA.SWRST bit is complete.
1	Read/write synchronization for CTRLA.SWRST bit is ongoing.

25.8.8. Frequency Correlation

Name: FREQCORR
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	SIGN		VALUE[5:0]					
Access	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset	0		0	0	0	0	0	0

Bit 7 – SIGN: Correction Sign

Value	Description
0	The correction value is positive, i.e., frequency will be decreased.
1	The correction value is negative, i.e., frequency will be increased.

Bits 5:0 – VALUE[5:0]: Correction Value
These bits define the amount of correction applied to the RTC prescaler.

Value	Description
0	Correction is disabled and the RTC frequency is unchanged.
1 - 127	The RTC frequency is adjusted according to the value.

25.8.9. Counter Value in COUNT32 mode (CTRLA.MODE=0)

Name: COUNT
Offset: 0x18
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Bit	31	30	29	28	27	26	25	24
	COUNT[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	COUNT[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	COUNT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COUNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – COUNT[31:0]: Counter Value
These bits define the value of the 32-bit RTC counter in mode 0.

25.8.10. Compare n Value in COUNT32 mode (CTRLA.MODE=0)

Name: COMPn
Offset: 0x20 + n*0x04 [n=0..3]
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
	COMP[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	COMP[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	COMP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COMP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – COMP[31:0]: Compare Value
The 32-bit value of COMPn is continuously compared with the 32-bit COUNT value. When a match occurs, the Compare n interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.CMPn) is set on the next counter cycle, and the counter value is cleared if CTRLA.MATCHCLR is one.

25.8.11. General Purpose n

Name: GPn
Offset: 0x40 + n*0x04 [n=0..3]
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	GP[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	GP[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	GP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	GP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – GP[31:0]: General Purpose
These bits are for user-defined general purpose use.

25.9. Register Summary - COUNT16

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0					MODE[1:0]		ENABLE	SWRST
0x01		15:8	COUNTSYNC				PRESCALER[3:0]			
0x02 ... 0x03	Reserved									
0x04	EVCTRL	7:0	PEREO7	PEREO6	PEREO5	PEREO4	PEREO3	PEREO2	PEREO1	PEREO0
0x05		15:8	OVFEO						CMPEO1	CMPEO0
0x06		23:16								
0x07		31:24								
0x08	INTENCLR	7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
0x09		15:8	OVF						CMP1	CMP0
0x0A	INTENSET	7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
0x0B		15:8	OVF						CMP1	CMP0
0x0C	INTFLAG	7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
0x0D		15:8	OVF						CMP1	CMP0
0x0E	DBGCTRL	7:0								DBGRUN
0x0F	Reserved									
0x10	SYNCBUSY	7:0		COMP1	COMP0	PER	COUNT	FREQCORR	ENABLE	SWRST
0x11		15:8	COUNTSYNC							
0x12		23:16								
0x13		31:24								
0x14	FREQCORR	7:0	SIGN		VALUE[5:0]					
0x15 ... 0x17	Reserved									
0x18	COUNT	7:0	COUNT[7:0]							
0x19		15:8	COUNT[15:8]							
0x1A ... 0x1F	Reserved									
0x20	COMP0	7:0	COMP[7:0]							
0x21		15:8	COMP[15:8]							
0x22	COMP1	7:0	COMP[7:0]							
0x23		15:8	COMP[15:8]							
0x24	COMP2	7:0	COMP[7:0]							
0x25		15:8	COMP[15:8]							
0x26	COMP3	7:0	COMP[7:0]							
0x27		15:8	COMP[15:8]							
0x28	COMP4	7:0	COMP[7:0]							
0x29		15:8	COMP[15:8]							
0x2A	COMP5	7:0	COMP[7:0]							
0x2B		15:8	COMP[15:8]							
0x2C ... 0x3F	Reserved									

Offset	Name	Bit Pos.								
0x40	GP0	7:0	GP[7:0]							
0x41		15:8	GP[15:8]							
0x42		23:16	GP[23:16]							
0x43		31:24	GP[31:24]							
0x44	GP1	7:0	GP[7:0]							
0x45		15:8	GP[15:8]							
0x46		23:16	GP[23:16]							
0x47		31:24	GP[31:24]							
0x48	GP2	7:0	GP[7:0]							
0x49		15:8	GP[15:8]							
0x4A		23:16	GP[23:16]							
0x4B		31:24	GP[31:24]							
0x4C	GP3	7:0	GP[7:0]							
0x4D		15:8	GP[15:8]							
0x4E		23:16	GP[23:16]							
0x4F		31:24	GP[31:24]							

25.10. Register Description - COUNT16

This Register Description section is valid if the RTC is in COUNT16 mode (CTRLA.MODE=1).

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

25.10.1. Control A in COUNT16 mode (CTRLA.MODE=1)

Name: CTRLA
Offset: 0x00
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	COUNTSYNC				PRESCALER[3:0]			
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit	7	6	5	4	3	2	1	0
					MODE[1:0]		ENABLE	SWRST
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 15 – COUNTSYNC: COUNT Read Synchronization Enable
The COUNT register requires synchronization when reading. Disabling the synchronization will prevent reading valid values from the COUNT register.
This bit is not enable-protected.

Value	Description
0	COUNT read synchronization is disabled
1	COUNT read synchronization is enabled

Bits 11:8 – PRESCALER[3:0]: Prescaler
These bits define the prescaling factor for the RTC clock source (GCLK_RTC) to generate the counter clock (CLK_RTC_CNT). Periodic events and interrupts are not available when the prescaler is off. These bits are not synchronized.

Value	Name	Description
0x0	OFF	CLK_RTC_CNT = GCLK_RTC/1
0x1	DIV1	CLK_RTC_CNT = GCLK_RTC/1
0x2	DIV2	CLK_RTC_CNT = GCLK_RTC/2
0x3	DIV4	CLK_RTC_CNT = GCLK_RTC/4
0x4	DIV8	CLK_RTC_CNT = GCLK_RTC/8
0x5	DIV16	CLK_RTC_CNT = GCLK_RTC/16
0x6	DIV32	CLK_RTC_CNT = GCLK_RTC/32
0x7	DIV64	CLK_RTC_CNT = GCLK_RTC/64
0x8	DIV128	CLK_RTC_CNT = GCLK_RTC/128
0x9	DIV256	CLK_RTC_CNT = GCLK_RTC/256
0xA	DIV512	CLK_RTC_CNT = GCLK_RTC/512

Value	Name	Description
0xB	DIV1024	CLK_RTC_CNT = GCLK_RTC/1024
0xC-0xF	-	Reserved

Bits 3:2 – MODE[1:0]: Operating Mode
This field defines the operating mode of the RTC. This bit is not synchronized.

Value	Name	Description
0x0	COUNT32	Mode 0: 32-bit counter
0x1	COUNT16	Mode 1: 16-bit counter
0x2	CLOCK	Mode 2: Clock/calendar
0x3	-	Reserved

Bit 1 – ENABLE: Enable
Due to synchronization there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the Enable bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

Value	Description
0	The peripheral is disabled
1	The peripheral is enabled

Bit 0 – SWRST: Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the RTC (except DBGCTRL) to their initial state, and the RTC will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.

Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST will be cleared when the reset is complete.

Value	Description
0	There is not reset operation ongoing
1	The reset operation is ongoing

25.10.2. Event Control in COUNT16 mode (CTRLA.MODE=1)

Name: EVCTRL
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	OVFEO						CMPEO1	CMPEO0
Access	R/W						R/W	R/W
Reset	0						0	0

Bit	7	6	5	4	3	2	1	0
	PEREO7	PEREO6	PEREO5	PEREO4	PEREO3	PEREO2	PEREO1	PEREO0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVFEO: Overflow Event Output Enable

Value	Description
0	Overflow event is disabled and will not be generated.
1	Overflow event is enabled and will be generated for every overflow.

Bits 9:8 – CMPEOn: Compare n Event Output Enable [n = 1..0]

Value	Description
0	Compare n event is disabled and will not be generated.
1	Compare n event is enabled and will be generated for every compare match.

Bits 7:0 – PEREOn: Periodic Interval n Event Output Enable [n = 7..0]

Value	Description
0	Periodic Interval n event is disabled and will not be generated. [n = 7..0]
1	Periodic Interval n event is enabled and will be generated. [n = 7..0]

25.10.3. Interrupt Enable Clear in COUNT16 mode (CTRLA.MODE=1)

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Name: INTENCLR
Offset: 0x08
Reset: 0x0000
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
	OVF						CMP1	CMP0
Access	R/W						R/W	R/W
Reset	0						0	0

Bit	7	6	5	4	3	2	1	0
	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF: Overflow Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Overflow Interrupt Enable bit, which disables the Overflow interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Bits 9:8 – CMPn: Compare n Interrupt Enable [n = 1..0]
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Compare n Interrupt Enable bit, which disables the Compare n interrupt.

Value	Description
0	The Compare n interrupt is disabled.
1	The Compare n interrupt is enabled.

Bits 7:0 – PERn: Periodic Interval n Interrupt Enable [n = 7..0]
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Periodic Interval n Interrupt Enable bit, which disables the Periodic Interval n interrupt.

Value	Description
0	Periodic Interval n interrupt is disabled.
1	Periodic Interval n interrupt is enabled.

25.10.4. Interrupt Enable Set in COUNT16 mode (CTRLA.MODE=1)

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Name: INTENSET
Offset: 0x0A
Reset: 0x0000
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
	OVF						CMP1	CMP0
Access	R/W						R/W	R/W
Reset	0						0	0

Bit	7	6	5	4	3	2	1	0
	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF: Overflow Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Overflow Interrupt Enable bit, which enables the Overflow interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Bits 9:8 – CMPn: Compare n Interrupt Enable [n = 1..0]
Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Compare n Interrupt Enable bit, which and enables the Compare n interrupt.

Value	Description
0	The Compare n interrupt is disabled.
1	The Compare n interrupt is enabled.

Bits 7:0 – PERn: Periodic Interval n Interrupt Enable [n = 7..0]
Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Periodic Interval n Interrupt Enable bit, which enables the Periodic Interval n interrupt.

Value	Description
0	Periodic Interval n interrupt is disabled.
1	Periodic Interval n interrupt is enabled.

25.10.5. Interrupt Flag Status and Clear in COUNT16 mode (CTRLA.MODE=1)

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Name: INTFLAG
Offset: 0x0C
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
	OVF						CMP1	CMP0
Access	R/W						R/W	R/W
Reset	0						0	0

Bit	7	6	5	4	3	2	1	0
	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF: Overflow
This flag is cleared by writing a '1' to the flag.

This flag is set on the next CLK_RTC_CNT cycle after an overflow condition occurs, and an interrupt request will be generated if INTENCLR/SET.OVF is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Overflow interrupt flag.

Bits 9:8 – CMPn: Compare n [n = 1..0]
This flag is cleared by writing a '1' to the flag.

This flag is set on the next CLK_RTC_CNT cycle after a match with the compare condition, and an interrupt request will be generated if INTENCLR/SET.COMPx is one.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Compare n interrupt flag.

Bits 7:0 – PERn: Periodic Interval n [n = 7..0]
This flag is cleared by writing a '1' to the flag.

This flag is set on the 0-to-1 transition of prescaler bit [n+2], and an interrupt request will be generated if INTENCLR/SET.PERx is one.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Periodic Interval n interrupt flag.

25.10.6. Debug Control

Name: DBGCTRL
Offset: 0x0E
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGRUN
Access								R/W
Reset								0

Bit 0 – DBGRUN: Debug Run
This bit is not reset by a software reset.

This bit controls the functionality when the CPU is halted by an external debugger.

Value	Description
0	The RTC is halted when the CPU is halted by an external debugger.
1	The RTC continues normal operation when the CPU is halted by an external debugger.

25.10.7. Synchronization Busy in COUNT16 mode (CTRLA.MODE=1)

Name: SYNCBUSY
Offset: 0x10
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	COUNTSYNC							
Access	R							
Reset	0							

Bit	7	6	5	4	3	2	1	0
		COMP1	COMP0	PER	COUNT	FREQCORR	ENABLE	SWRST
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit 15 – COUNTSYNC: Count Read Sync Enable Synchronization Busy Status

Value	Description
0	Write synchronization for CTRLA.COUNTSYNC bit is complete.
1	Write synchronization for CTRLA.COUNTSYNC bit is ongoing.

Bits 6:5 – COMPn: Compare n Synchronization Busy Status [n = 1..0]

Value	Description
0	Write synchronization for COMPn register is complete.
1	Write synchronization for COMPn register is ongoing.

Bit 4 – PER: Period Synchronization Busy Status

Value	Description
0	Write synchronization for PER register is complete.
1	Write synchronization for PER register is ongoing.

Bit 3 – COUNT: Count Value Synchronization Busy Status

Value	Description
0	Read/write synchronization for COUNT register is complete.
1	Read/write synchronization for COUNT register is ongoing.

Bit 2 – FREQCORR: Frequency Correction Synchronization Busy Status

Value	Description
0	Read/write synchronization for FREQCORR register is complete.
1	Read/write synchronization for FREQCORR register is ongoing.

Bit 1 – ENABLE: Enable Synchronization Busy Status

Value	Description
0	Read/write synchronization for CTRLA.ENABLE bit is complete.
1	Read/write synchronization for CTRLA.ENABLE bit is ongoing.

Bit 0 – SWRST: Software Reset Synchronization Busy Status

Value	Description
0	Read/write synchronization for CTRLA.SWRST bit is complete.
1	Read/write synchronization for CTRLA.SWRST bit is ongoing.

25.10.8. Frequency Correlation

Name: FREQCORR
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	SIGN		VALUE[5:0]					
Access	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset	0		0	0	0	0	0	0

Bit 7 – SIGN: Correction Sign

Value	Description
0	The correction value is positive, i.e., frequency will be decreased.
1	The correction value is negative, i.e., frequency will be increased.

Bits 5:0 – VALUE[5:0]: Correction Value
These bits define the amount of correction applied to the RTC prescaler.

Value	Description
0	Correction is disabled and the RTC frequency is unchanged.
1 - 127	The RTC frequency is adjusted according to the value.

25.10.9. Counter Value in COUNT16 mode (CTRLA.MODE=1)

Name: COUNT
Offset: 0x18
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Bit	15	14	13	12	11	10	9	8
	COUNT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COUNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – COUNT[15:0]: Counter Value
These bits define the value of the 16-bit RTC counter in COUNT16 mode (CTRLA.MODE=1).

25.10.10. Counter Period in COUNT16 mode (CTRLA.MODE=1)

Name: PER
Offset: 0x18
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	PER[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PER[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – PER[15:0]: Counter Period
These bits define the value of the 16-bit RTC period in COUNT16 mode (CTRLA.MODE=1).

25.10.11. Compare n Value in COUNT16 mode (CTRLA.MODE=1)

Name: COMPn
Offset: 0x20 + n*0x02 [n=0..5]
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	COMP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COMP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – COMP[15:0]: Compare Value
The 16-bit value of COMPn is continuously compared with the 16-bit COUNT value. When a match occurs, the Compare n interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.CMPn) is set on the next counter cycle.

25.10.12. General Purpose n

Name: GPn
Offset: 0x40 + n*0x04 [n=0..3]
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	GP[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	GP[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	GP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	GP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – GP[31:0]: General Purpose
These bits are for user-defined general purpose use.

25.11. Register Summary - CLOCK

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0	MATCHCLR	CLKREP			MODE[1:0]		ENABLE	SWRST
0x01		15:8	CLOCKSYNC				PRESCALER[3:0]			
0x02 ... 0x03	Reserved									
0x04	EVCTRL	7:0	PEREO7	PEREO6	PEREO5	PEREO4	PEREO3	PEREO2	PEREO1	PEREO0
0x05		15:8	OVFEO							ALARMO0
0x06		23:16								
0x07		31:24								
0x08	INTENCLR	7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
0x09		15:8	OVF							ALARM0
0x0A	INTENSET	7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
0x0B		15:8	OVF							ALARM0
0x0C	INTFLAG	7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
0x0D		15:8	OVF							ALARM0
0x0E	DBGCTRL	7:0								DBGRUN
0x0F	Reserved									
0x10	SYNCBUSY	7:0			ALARM0		COUNT	FREQCORR	ENABLE	SWRST
0x11		15:8	CLOCKSYNC				MASK0			
0x12		23:16								
0x13		31:24								
0x14	FREQCORR	7:0	SIGN		VALUE[5:0]					
0x15 ... 0x17	Reserved									
0x18	CLOCK	7:0	MINUTE[1:0]		SECOND[5:0]					
0x19		15:8	HOUR[3:0]				MINUTE[5:2]			
0x1A		23:16	MONTH[1:0]		DAY[4:0]					HOUR[4:4]
0x1B		31:24	YEAR[5:0]						MONTH[3:2]	
0x1C ... 0x1F	Reserved									
0x20	ALARM0	7:0	MINUTE[1:0]		SECOND[5:0]					
0x21		15:8	HOUR[3:0]				MINUTE[5:2]			
0x22		23:16	MONTH[1:0]		DAY[4:0]					HOUR[4:4]
0x23		31:24	YEAR[5:0]						MONTH[3:2]	
0x24	ALARM1	7:0	MINUTE[1:0]		SECOND[5:0]					
0x25		15:8	HOUR[3:0]				MINUTE[5:2]			
0x26		23:16	MONTH[1:0]		DAY[4:0]					HOUR[4:4]
0x27		31:24	YEAR[5:0]						MONTH[3:2]	
0x28	ALARM2	7:0	MINUTE[1:0]		SECOND[5:0]					
0x29		15:8	HOUR[3:0]				MINUTE[5:2]			
0x2A		23:16	MONTH[1:0]		DAY[4:0]					HOUR[4:4]
0x2B		31:24	YEAR[5:0]						MONTH[3:2]	

Offset	Name	Bit Pos.								
0x2C	ALARM3	7:0	MINUTE[1:0]		SECOND[5:0]					
0x2D		15:8	HOUR[3:0]				MINUTE[5:2]			
0x2E		23:16	MONTH[1:0]		DAY[4:0]					HOUR[4:4]
0x2F		31:24	YEAR[5:0]						MONTH[3:2]	
0x30 ... 0x3F	Reserved									
0x40	GP0	7:0	GP[7:0]							
0x41		15:8	GP[15:8]							
0x42		23:16	GP[23:16]							
0x43		31:24	GP[31:24]							
0x44	GP1	7:0	GP[7:0]							
0x45		15:8	GP[15:8]							
0x46		23:16	GP[23:16]							
0x47		31:24	GP[31:24]							
0x48	GP2	7:0	GP[7:0]							
0x49		15:8	GP[15:8]							
0x4A		23:16	GP[23:16]							
0x4B		31:24	GP[31:24]							
0x4C	GP3	7:0	GP[7:0]							
0x4D		15:8	GP[15:8]							
0x4E		23:16	GP[23:16]							
0x4F		31:24	GP[31:24]							

25.12. Register Description - CLOCK

This Register Description section is valid if the RTC is in Clock/Calendar mode (CTRLA.MODE=2).

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

25.12.1. Control A in Clock/Calendar mode (CTRLA.MODE=2)

Name: CTRLA
Offset: 0x00
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	CLOCKSYNC				PRESCALER[3:0]			
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MATCHCLR	CLKREP			MODE[1:0]		ENABLE	SWRST
Access	R/W	R/W			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

Bit 15 – CLOCKSYNC: CLOCK Read Synchronization Enable
The CLOCK register requires synchronization when reading. Disabling the synchronization will prevent reading valid values from the CLOCK register.
This bit is not enable-protected.

Value	Description
0	CLOCK read synchronization is disabled
1	CLOCK read synchronization is enabled

Bits 11:8 – PRESCALER[3:0]: Prescaler
These bits define the prescaling factor for the RTC clock source (GCLK_RTC) to generate the counter clock (CLK_RTC_CNT). Periodic events and interrupts are not available when the prescaler is off. These bits are not synchronized.

Value	Name	Description
0x0	OFF	CLK_RTC_CNT = GCLK_RTC/1
0x1	DIV1	CLK_RTC_CNT = GCLK_RTC/1
0x2	DIV2	CLK_RTC_CNT = GCLK_RTC/2
0x3	DIV4	CLK_RTC_CNT = GCLK_RTC/4
0x4	DIV8	CLK_RTC_CNT = GCLK_RTC/8
0x5	DIV16	CLK_RTC_CNT = GCLK_RTC/16
0x6	DIV32	CLK_RTC_CNT = GCLK_RTC/32
0x7	DIV64	CLK_RTC_CNT = GCLK_RTC/64
0x8	DIV128	CLK_RTC_CNT = GCLK_RTC/128
0x9	DIV256	CLK_RTC_CNT = GCLK_RTC/256
0xA	DIV512	CLK_RTC_CNT = GCLK_RTC/512

Value	Name	Description
0xB	DIV1024	CLK_RTC_CNT = GCLK_RTC/1024
0xC-0xF	-	Reserved

Bit 7 – MATCHCLR: Clear on Match
This bit is valid only in Mode 0 (COUNT32) and Mode 2 (CLOCK). This bit can be written only when the peripheral is disabled. This bit is not synchronized.

Value	Description
0	The counter is not cleared on a Compare/Alarm 0 match
1	The counter is cleared on a Compare/Alarm 0 match

Bit 6 – CLKREP: Clock Representation
This bit is valid only in Mode 2 and determines how the hours are represented in the Clock Value (CLOCK) register. This bit can be written only when the peripheral is disabled. This bit is not synchronized.

Value	Description
0	24 Hour
1	12 Hour (AM/PM)

Bits 3:2 – MODE[1:0]: Operating Mode
This field defines the operating mode of the RTC. This bit is not synchronized.

Value	Name	Description
0x0	COUNT32	Mode 0: 32-bit counter
0x1	COUNT16	Mode 1: 16-bit counter
0x2	CLOCK	Mode 2: Clock/calendar
0x3	-	Reserved

Bit 1 – ENABLE: Enable
Due to synchronization there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the Enable bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

Value	Description
0	The peripheral is disabled
1	The peripheral is enabled

Bit 0 – SWRST: Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the RTC, except DBGCTRL, to their initial state, and the RTC will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.

Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST will be cleared when the reset is complete.

Value	Description
0	There is not reset operation ongoing
1	The reset operation is ongoing

25.12.2. Event Control in Clock/Calendar mode (CTRLA.MODE=2)

Name: EVCTRL
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	OVFEO							ALARMO0
Access	R/W							R/W
Reset	0							0

Bit	7	6	5	4	3	2	1	0
	PEREO7	PEREO6	PEREO5	PEREO4	PEREO3	PEREO2	PEREO1	PEREO0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVFEO: Overflow Event Output Enable

Value	Description
0	Overflow event is disabled and will not be generated.
1	Overflow event is enabled and will be generated for every overflow.

Bit 8 – ALARMO0: Alarm 0 Event Output Enable

Value	Description
0	Alarm 0 event is disabled and will not be generated.
1	Alarm 0 event is enabled and will be generated for every compare match.

Bits 7:0 – PEREOn: Periodic Interval n Event Output Enable [n = 7..0]

Value	Description
0	Periodic Interval n event is disabled and will not be generated.
1	Periodic Interval n event is enabled and will be generated.

25.12.3. Interrupt Enable Clear in Clock/Calendar mode (CTRLA.MODE=2)

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Name: INTENCLR
Offset: 0x08
Reset: 0x0000
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
	OVF							ALARM0
Access	R/W							R/W
Reset	0							0

Bit	7	6	5	4	3	2	1	0
	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF: Overflow Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Overflow Interrupt Enable bit, which disables the Overflow interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Bit 8 – ALARM0: Alarm 0 Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Alarm 0 Interrupt Enable bit, which disables the Alarm interrupt.

Value	Description
0	The Alarm 0 interrupt is disabled.
1	The Alarm 0 interrupt is enabled.

Bits 7:0 – PERn: Periodic Interval n Interrupt Enable [n = 7..0]
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Periodic Interval n Interrupt Enable bit, which disables the Periodic Interval n interrupt.

Value	Description
0	Periodic Interval n interrupt is disabled.
1	Periodic Interval n interrupt is enabled.

25.12.4. Interrupt Enable Set in Clock/Calendar mode (CTRLA.MODE=2)

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Name: INTENSET
Offset: 0x0A
Reset: 0x0000
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
	OVF							ALARM0
Access	R/W							R/W
Reset	0							0

Bit	7	6	5	4	3	2	1	0
	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF: Overflow Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Overflow Interrupt Enable bit, which enables the Overflow interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Bit 8 – ALARM0: Alarm 0 Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Alarm 0 Interrupt Enable bit, which enables the Alarm 0 interrupt.

Value	Description
0	The Alarm 0 interrupt is disabled.
1	The Alarm 0 interrupt is enabled.

Bits 7:0 – PERn: Periodic Interval n Interrupt Enable [n = 7..0]
Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Periodic Interval n Interrupt Enable bit, which enables the Periodic Interval n interrupt.

Value	Description
0	Periodic Interval n interrupt is disabled.
1	Periodic Interval n interrupt is enabled.

25.12.5. Interrupt Flag Status and Clear in Clock/Calendar mode (CTRLA.MODE=2)

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Name: INTFLAG
Offset: 0x0C
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
	OVF							ALARM0
Access	R/W							R/W
Reset	0							0

Bit	7	6	5	4	3	2	1	0
	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF: Overflow
This flag is cleared by writing a '1' to the flag.

This flag is set on the next CLK_RTC_CNT cycle after an overflow condition occurs, and an interrupt request will be generated if INTENCLR/SET.OVF is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Overflow interrupt flag.

Bit 8 – ALARM0: Alarm 0
This flag is cleared by writing a '1' to the flag.

This flag is set on the next CLK_RTC_CNT cycle after a match with the compare condition, and an interrupt request will be generated if INTENCLR/SET.ALARM0 is one.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Alarm 0 interrupt flag.

Bits 7:0 – PERn: Periodic Interval n [n = 7..0]
This flag is cleared by writing a '1' to the flag.

This flag is set on the 0-to-1 transition of prescaler bit [n+2], and an interrupt request will be generated if INTENCLR/SET.PERx is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Periodic Interval n interrupt flag.

25.12.6. Debug Control

Name: DBGCTRL
Offset: 0x0E
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGRUN
Access								R/W
Reset								0

Bit 0 – DBGRUN: Debug Run
This bit is not reset by a software reset.

This bit controls the functionality when the CPU is halted by an external debugger.

Value	Description
0	The RTC is halted when the CPU is halted by an external debugger.
1	The RTC continues normal operation when the CPU is halted by an external debugger.

25.12.7. Synchronization Busy in Clock/Calendar mode (CTRLA.MODE=2)

Name: SYNCBUSY
Offset: 0x10
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	CLOCKSYNC				MASK0			
Access	R				R			
Reset	0				0			

Bit	7	6	5	4	3	2	1	0
			ALARM0		COUNT	FREQCORR	ENABLE	SWRST
Access			R		R	R	R	R
Reset			0		0	0	0	0

Bit 15 – CLOCKSYNC: Clock Read Sync Enable Synchronization Busy Status

Value	Description
0	Write synchronization for CTRLA.CLOCKSYNC bit is complete.
1	Write synchronization for CTRLA.CLOCKSYNC bit is ongoing.

Bit 11 – MASK0: Mask 0 Synchronization Busy Status

Value	Description
0	Write synchronization for MASK0 register is complete.
1	Write synchronization for MASK0 register is ongoing.

Bit 5 – ALARM0: Alarm 0 Synchronization Busy Status

Value	Description
0	Write synchronization for ALARM0 register is complete.
1	Write synchronization for ALARM0 register is ongoing.

Bit 3 – COUNT: Count Value Synchronization Busy Status

Value	Description
0	Read/write synchronization for COUNT register is complete.
1	Read/write synchronization for COUNT register is ongoing.

Bit 2 – FREQCORR: Frequency Correction Synchronization Busy Status

Value	Description
0	Read/write synchronization for FREQCORR register is complete.
1	Read/write synchronization for FREQCORR register is ongoing.

Bit 1 – ENABLE: Enable Synchronization Busy Status

Value	Description
0	Read/write synchronization for CTRLA.ENABLE bit is complete.
1	Read/write synchronization for CTRLA.ENABLE bit is ongoing.

Bit 0 – SWRST: Software Reset Synchronization Busy Status

Value	Description
0	Read/write synchronization for CTRLA.SWRST bit is complete.
1	Read/write synchronization for CTRLA.SWRST bit is ongoing.

25.12.8. Frequency Correlation

Name: FREQCORR
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	SIGN		VALUE[5:0]					
Access	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset	0		0	0	0	0	0	0

Bit 7 – SIGN: Correction Sign

Value	Description
0	The correction value is positive, i.e., frequency will be decreased.
1	The correction value is negative, i.e., frequency will be increased.

Bits 5:0 – VALUE[5:0]: Correction Value
These bits define the amount of correction applied to the RTC prescaler.

Value	Description
0	Correction is disabled and the RTC frequency is unchanged.
1 - 127	The RTC frequency is adjusted according to the value.

25.12.9. Clock Value in Clock/Calendar mode (CTRLA.MODE=2)

Name: CLOCK
Offset: 0x18
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Bit	31	30	29	28	27	26	25	24
	YEAR[5:0]						MONTH[3:2]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	MONTH[1:0]		DAY[4:0]					HOUR[4:4]
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	HOUR[3:0]				MINUTE[5:2]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MINUTE[1:0]		SECOND[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:26 – YEAR[5:0]: Year
The year offset with respect to the reference year (defined in software).

The year is considered a leap year if YEAR[1:0] is zero.

Bits 25:22 – MONTH[3:0]: Month
1 – January

2 – February

...

12 – December

Bits 21:17 – DAY[4:0]: Day
Day starts at 1 and ends at 28, 29, 30, or 31, depending on the month and year.

Bits 16:12 – HOUR[4:0]: Hour
When CTRLA.CLKREP=0, the Hour bit group is in 24-hour format, with values 0-23. When CTRLA.CLKREP=1, HOUR[3:0] has values 1-12, and HOUR[4] represents AM (0) or PM (1).

Bits 11:6 – MINUTE[5:0]: Minute
0 – 59

Bits 5:0 – SECOND[5:0]: Second
0 – 59

25.12.10. Alarm n Value in Clock/Calendar mode (CTRLA.MODE=2)

The 32-bit value of ALARMn is continuously compared with the 32-bit CLOCK value, based on the masking set by MASKn.SEL. When a match occurs, the Alarm n interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.ALARMn) is set on the next counter cycle, and the counter is cleared if CTRLA.MATCHCLR is '1'.

Name: ALARMn
Offset: 0x20 + n*0x04 [n=0..3]
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
	YEAR[5:0]						MONTH[3:2]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	MONTH[1:0]		DAY[4:0]					HOUR[4:4]
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	HOUR[3:0]				MINUTE[5:2]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MINUTE[1:0]		SECOND[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:26 – YEAR[5:0]: Year
The alarm year. Years are only matched if MASKn.SEL is 6

Bits 25:22 – MONTH[3:0]: Month
The alarm month. Months are matched only if MASKn.SEL is greater than 4.

Bits 21:17 – DAY[4:0]: Day
The alarm day. Days are matched only if MASKn.SEL is greater than 3.

Bits 16:12 – HOUR[4:0]: Hour
The alarm hour. Hours are matched only if MASKn.SEL is greater than 2.

Bits 11:6 – MINUTE[5:0]: Minute
The alarm minute. Minutes are matched only if MASKn.SEL is greater than 1.

Bits 5:0 – SECOND[5:0]: Second
The alarm second. Seconds are matched only if MASKn.SEL is greater than 0.

25.12.11. Alarm n Mask in Clock/Calendar mode (CTRLA.MODE=2)

Name: MASKn
Offset: 0x24 + n*0x01 [n=0..3]
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
						SEL[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 2:0 – SEL[2:0]: Alarm Mask Selection
These bits define which bit groups of Alarm n are valid.

Value	Name	Description
0x0	OFF	Alarm Disabled
0x1	SS	Match seconds only
0x2	MMSS	Match seconds and minutes only
0x3	HHMMSS	Match seconds, minutes, and hours only
0x4	DDHHMMSS	Match seconds, minutes, hours, and days only
0x5	MMDDHHMMSS	Match seconds, minutes, hours, days, and months only
0x6	YYMMDDHHMMSS	Match seconds, minutes, hours, days, months, and years
0x7	-	Reserved

25.12.12. General Purpose n

Name: GPn
Offset: 0x40 + n*0x04 [n=0..3]
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	GP[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	GP[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	GP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	GP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – GP[31:0]: General Purpose
These bits are for user-defined general purpose use.

26. DMAC – Direct Memory Access Controller

26.1. Overview

The Direct Memory Access Controller (DMAC) contains both a Direct Memory Access engine and a Cyclic Redundancy Check (CRC) engine. The DMAC can transfer data between memories and peripherals, and thus off-load these tasks from the CPU. It enables high data transfer rates with minimum CPU intervention, and frees up CPU time. With access to all peripherals, the DMAC can handle automatic transfer of data between communication modules.

The DMA part of the DMAC has several DMA channels which all can receive different types of transfer triggers to generate transfer requests from the DMA channels to the arbiter, see also the Block Diagram. The arbiter will grant one DMA channel at a time to act as the active channel. When an active channel has been granted, the fetch engine of the DMAC will fetch a transfer descriptor from the low-power (LP) SRAM and store it in the internal memory of the active channel, which will execute the data transmission.

An ongoing data transfer of an active channel can be interrupted by a higher prioritized DMA channel. The DMAC will write back the updated transfer descriptor from the internal memory of the active channel to LP SRAM, and grant the higher prioritized channel to start transfer as the new active channel. Once a DMA channel is done with its transfer, interrupts and events can be generated optionally.

The DMAC has four bus interfaces:

- The *data transfer bus* is used for performing the actual DMA transfer.
- The *AHB/APB Bridge bus* is used when writing and reading the I/O registers of the DMAC.
- The *descriptor fetch bus* is used by the fetch engine to fetch transfer descriptors before data transfer can be started or continued.
- The *write-back bus* is used to write the transfer descriptor back to LP SRAM.

All buses are AHB master interfaces but the AHB/APB Bridge bus, which is an APB slave interface.

The CRC engine can be used by software to detect an accidental error in the transferred data and to take corrective action, such as requesting the data to be sent again or simply not using the incorrect data.

26.2. Features

- Data transfer from:
  - Peripheral to peripheral
  - Peripheral to memory
  - Memory to peripheral
  - Memory to memory
- Transfer trigger sources
  - Software
  - Events from Event System
  - Dedicated requests from peripherals
- SRAM based transfer descriptors
  - Single transfer using one descriptor
  - Multi-buffer or circular buffer modes by linking multiple descriptors
- Up to 16channels

  - Enable 16 independent transfers
  - Automatic descriptor fetch for each channel
  - Suspend/resume operation support for each channel
- Flexible arbitration scheme
  - 4 configurable priority levels for each channel
  - Fixed or round-robin priority scheme within each priority level
- From 1 to 256KB data transfer in a single block transfer
- Multiple addressing modes
  - Static
  - Configurable increment scheme
- Optional interrupt generation
  - On block transfer complete
  - On error detection
  - On channel suspend
- 4 event inputs
  - One event input for each of the 4 least significant DMA channels
  - Can be selected to trigger normal transfers, periodic transfers or conditional transfers
  - Can be selected to suspend or resume channel operation
- 4 event outputs
  - One output event for each of the 4 least significant DMA channels
  - Selectable generation on AHB, block, or transaction transfer complete
- Error management supported by write-back function
  - Dedicated Write-Back memory section for each channel to store ongoing descriptor transfer
- CRC polynomial software selectable to
  - CRC-16 (CRC-CCITT)
  - CRC-32 (IEEE® 802.3)

26.3. Block Diagram

Figure 26-1. DMAC Block Diagram

26.4. Signal Description

Not applicable.

26.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

26.5.1. I/O Lines

Not applicable.

26.5.2. Power Management

The DMAC will continue to operate in any sleep mode where the selected source clock is running. The DMAC's interrupts can be used to wake up the device from sleep modes. Events connected to the event system can trigger other operations in the system without exiting sleep modes. On hardware or software reset, all registers are set to their reset value.

Related Links

PM – Power Manager on page 186

26.5.3. Clocks

The DMAC bus clock (CLK_DMAC_APB) must be configured and enabled in the Main Clock module before using the DMAC.

This bus clock (CLK_DMAC_APB) is always synchronous to the module clock (CLK_DMAC_AHB), but can be divided by a prescaler and may run even when the module clock is turned off.

Related Links

Peripheral Clock Masking on page 151

26.5.4. DMA

Not applicable.

26.5.5. Interrupts

The interrupt request line is connected to the interrupt controller. Using the DMAC interrupt requires the interrupt controller to be configured first.

Related Links

Nested Vector Interrupt Controller on page 51

26.5.6. Events

The events are connected to the event system.

Related Links

EVSYS – Event System on page 536

26.5.7. Debug Operation

When the CPU is halted in debug mode the DMAC will halt normal operation. The DMAC can be forced to continue operation during debugging. Refer to DBGCTRL on page 432 for details.

26.5.8. Register Access Protection

All registers with write-access can be write-protected optionally by the Peripheral Access Controller (PAC), except the following registers:

- Interrupt Pending register (INTPEND)
- Channel ID register (CHID)
- Channel Interrupt Flag Status and Clear register (CHINTFLAG)

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

PAC write-protection does not apply to accesses through an external debugger.

Related Links

PAC - Peripheral Access Controller on page 58

26.5.9. Analog Connections

Not applicable.

26.6. Functional Description

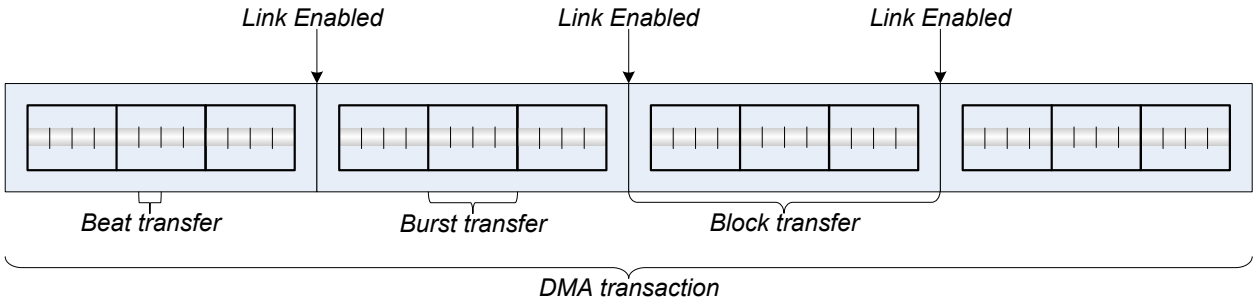
26.6.1. Principle of Operation

The DMAC consists of a DMA module and a CRC module.

26.6.1.1. DMA

The DMAC can transfer data between memories and peripherals without interaction from the CPU. The data transferred by the DMAC are called transactions, and these transactions can be split into smaller data transfers. Figure 'DMA Transfer Sizes' shows the relationship between the different transfer sizes:

Figure 26-2. DMA Transfer Sizes



- Beat transfer: The size of one data transfer bus access, and the size is selected by writing the Beat Size bit group in the Block Transfer Control register (BTCTRL.BEATSIZE)
- Burst transfer: Defined as n beat transfers, where n will differ from one device family to another. A burst transfer is atomic, cannot be interrupted and the length of the burst is selected by writing the Burst Length bit group in each Channel n Control A register (CHCTRLA.BURSTLEN).
- Block transfer: The amount of data one transfer descriptor can transfer, and the amount can range from 1 to 64k beats. In contrast to the burst transfer, a block transfer can be interrupted.
- Transaction: The DMAC can link several transfer descriptors by having the first descriptor pointing to the second and so forth, as shown in the figure above. A DMA transaction is the complete transfer of all blocks within a linked list.

A transfer descriptor describes how a block transfer should be carried out by the DMAC, and it must remain in Low-Power (LP) SRAM. For further details on the transfer descriptor refer to Transfer Descriptors on page 404.

The figure above shows several block transfers linked together, which are called linked descriptors. For further information about linked descriptors, refer to Linked Descriptors on page 412.

A DMA transfer is initiated by an incoming transfer trigger on one of the DMA channels. This trigger can be configured to be either a software trigger, an event trigger, or one of the dedicated peripheral triggers. The transfer trigger will result in a DMA transfer request from the specific channel to the arbiter. If there are several DMA channels with pending transfer requests, the arbiter chooses which channel is granted access to become the active channel. The DMA channel granted access as the active channel will carry out the transaction as configured in the transfer descriptor. A current transaction can be interrupted by a higher prioritized channel after each burst transfer, but will resume the block transfer when the according DMA channel is granted access as the active channel again.

For each beat transfer, an optional output event can be generated. For each block transfer, optional interrupts and an optional output event can be generated. When a transaction is completed, dependent of the configuration, the DMA channel will either be suspended or disabled.

26.6.1.2. CRC

The internal CRC engine supports two commonly used CRC polynomials: CRC-16 (CRC-CCITT) and CRC-32 (IEEE 802.3). It can be used on a selectable DMA channel, or on the I/O interface. Refer to CRC Operation on page 418 for details.

26.6.2. Basic Operation

26.6.2.1. Initialization

The following DMAC registers are enable-protected, meaning that they can only be written when the DMAC is disabled (CTRL.DMAENABLE=0):

- Descriptor Base Memory Address register (BASEADDR)
- Write-Back Memory Base Address register (WRBADDR)

The following DMAC bit is enable-protected, meaning that it can only be written when both the DMAC and CRC are disabled (CTRL.DMAENABLE=0 and CTRL.CRCENABLE=0):

- Software Reset bit in Control register (CTRL.SWRST)

The following DMA channel register is enable-protected, meaning that it can only be written when the corresponding DMA channel is disabled (CHCTRLA.ENABLE=0):

- Channel Control B (CHCTRLB) register, except the Command bit (CHCTRLB.CMD) and the Channel Arbitration Level bit (CHCTRLB.LVL)

The following DMA channel bit is enable-protected, meaning that it can only be written when the corresponding DMA channel is disabled:

- Channel Software Reset bit in Channel Control A register (CHCTRLA.SWRST)

The following CRC registers are enable-protected, meaning that they can only be written when the CRC is disabled (CTRL.CRCENABLE=0):

- CRC Control register (CRCCTRL)
- CRC Checksum register (CRCCHKSUM)

Enable-protection is denoted by the "Enable-Protected" property in the register description.

Before the DMAC is enabled it must be configured, as outlined by the following steps:

- The LP SRAM address of where the descriptor memory section is located must be written to the Description Base Address (BASEADDR) register
- The LP SRAM address of where the write-back section should be located must be written to the Write-Back Memory Base Address (WRBADDR) register
- Priority level x of the arbiter can be enabled by setting the Priority Level x Enable bit in the Control register (CTRL.LVLENx=1)

Before a DMA channel is enabled, the DMA channel and the corresponding first transfer descriptor must be configured, as outlined by the following steps:

- DMA channel configurations
  - The channel number of the DMA channel to configure must be written to the Channel ID (CHID) register
  - Trigger action must be selected by writing the Trigger Action bit group in the Channel Control B register (CHCTRLB.TRIGACT)
  - Trigger source must be selected by writing the Trigger Source bit group in the Channel Control B register (CHCTRLB.TRIGSRC)

- Transfer Descriptor
  - The size of each access of the data transfer bus must be selected by writing the Beat Size bit group in the Block Transfer Control register (BTCTRL.BEATSIZE)
  - The transfer descriptor must be made valid by writing a one to the Valid bit in the Block Transfer Control register (BTCTRL.VALID)
  - Number of beats in the block transfer must be selected by writing the Block Transfer Count (BTCNT) register
  - Source address for the block transfer must be selected by writing the Block Transfer Source Address (SRCADDR) register
  - Destination address for the block transfer must be selected by writing the Block Transfer Destination Address (DSTADDR) register

If CRC calculation is needed, the CRC engine must be configured before it is enabled, as outlined by the following steps:

- The CRC input source must selected by writing the CRC Input Source bit group in the CRC Control register (CRCCTRL.CRCSRC)
- The type of CRC calculation must be selected by writing the CRC Polynomial Type bit group in the CRC Control register (CRCCTRL.CRCPOLY)
- If I/O is selected as input source, the beat size must be selected by writing the CRC Beat Size bit group in the CRC Control register (CRCCTRL.CRCBEATSIZE)

26.6.2.2. Enabling, Disabling, and Resetting

The DMAC is enabled by writing the DMA Enable bit in the Control register (CTRL.DMAENABLE) to '1'. The DMAC is disabled by writing a '0' to CTRL.DMAENABLE.

A DMA channel is enabled by writing the Enable bit in the Channel Control A register (CHCTRLA.ENABLE) to '1', after writing the corresponding channel id to the Channel ID bit group in the Channel ID register (CHID.ID). A DMA channel is disabled by writing a '0' to CHCTRLA.ENABLE.

The CRC is enabled by writing a '1' to the CRC Enable bit in the Control register (CTRL.CRCENABLE). The CRC is disabled by writing a '0' to CTRL.CRCENABLE.

The DMAC is reset by writing a '1' to the Software Reset bit in the Control register (CTRL.SWRST) while the DMAC and CRC are disabled. All registers in the DMAC except DBGCTRL will be reset to their initial state.

A DMA channel is reset by writing a '1' to the Software Reset bit in the Channel Control A register (CHCTRLA.SWRST), after writing the corresponding channel id to the Channel ID bit group in the Channel ID register (CHID.ID). The channel registers will be reset to their initial state. The corresponding DMA channel must be disabled in order for the reset to take effect.

26.6.2.3. Transfer Descriptors

Together with the channel configurations the transfer descriptors decides how a block transfer should be executed. Before a DMA channel is enabled (CHCTRLA.ENABLE is written to one), and receives a transfer trigger, its first transfer descriptor has to be initialized and valid (BTCTRL.VALID). The first transfer descriptor describes the first block transfer of a transaction.

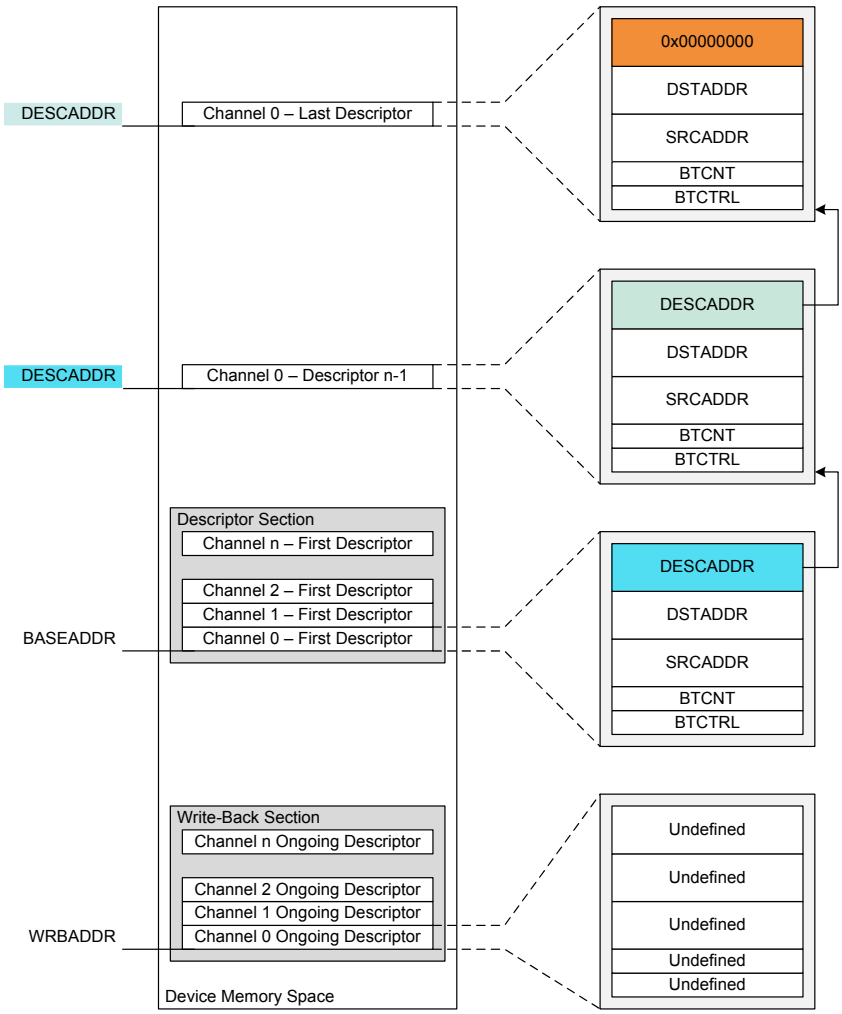
All transfer descriptors must reside in LP SRAM. The addresses stored in the Descriptor Memory Section Base Address (BASEADDR) and Write-Back Memory Section Base Address (WRBADDR) registers tell the DMAC where to find the descriptor memory section and the write-back memory section.

The descriptor memory section is where the DMAC expects to find the first transfer descriptors for all DMA channels. As BASEADDR points only to the first transfer descriptor of channel 0 (see figure below), all first transfer descriptors must be stored in a contiguous memory section, where the transfer descriptors

must be ordered according to their channel number. For further details on linked descriptors, refer to Linked Descriptors on page 412.

The write-back memory section is the section where the DMAC stores the transfer descriptors for the ongoing block transfers. WRBADDR points to the ongoing transfer descriptor of channel 0. All ongoing transfer descriptors will be stored in a contiguous memory section where the transfer descriptors are ordered according to their channel number. The figure below shows an example of linked descriptors on DMA channel 0. For further details on linked descriptors, refer to Linked Descriptors on page 412.

Figure 26-3. Memory Sections



The size of the descriptor and write-back memory sections is dependent on the number of the most significant enabled DMA channel *m*, as shown below:

$$Size = 128\text{bits} \cdot (m + 1)$$

For memory optimization, it is recommended to always use the less significant DMA channels if not all channels are required.

The descriptor and write-back memory sections can either be two separate memory sections, or they can share memory section (BASEADDR=WRBADDR). The benefit of having them in two separate sections, is that the same transaction for a channel can be repeated without having to modify the first transfer descriptor. The benefit of having descriptor memory and write-back memory in the same section is that it requires less LP SRAM. In addition, the latency from fetching the first descriptor of a transaction to the first burst transfer is executed, is reduced.

26.6.2.4. Arbitration

If a DMA channel is enabled and not suspended when it receives a transfer trigger, it will send a transfer request to the arbiter. When the arbiter receives the transfer request it will include the DMA channel in the queue of channels having pending transfers, and the corresponding Pending Channel x bit in the Pending Channels registers (PENDCH.PENDCHx) will be set. Depending on the arbitration scheme, the arbiter will choose which DMA channel will be the next active channel. The active channel is the DMA channel being granted access to perform its next burst transfer. When the arbiter has granted a DMA channel access to the DMAC, the corresponding bit PENDCH.PENDCHx will be cleared. See also the following figure.

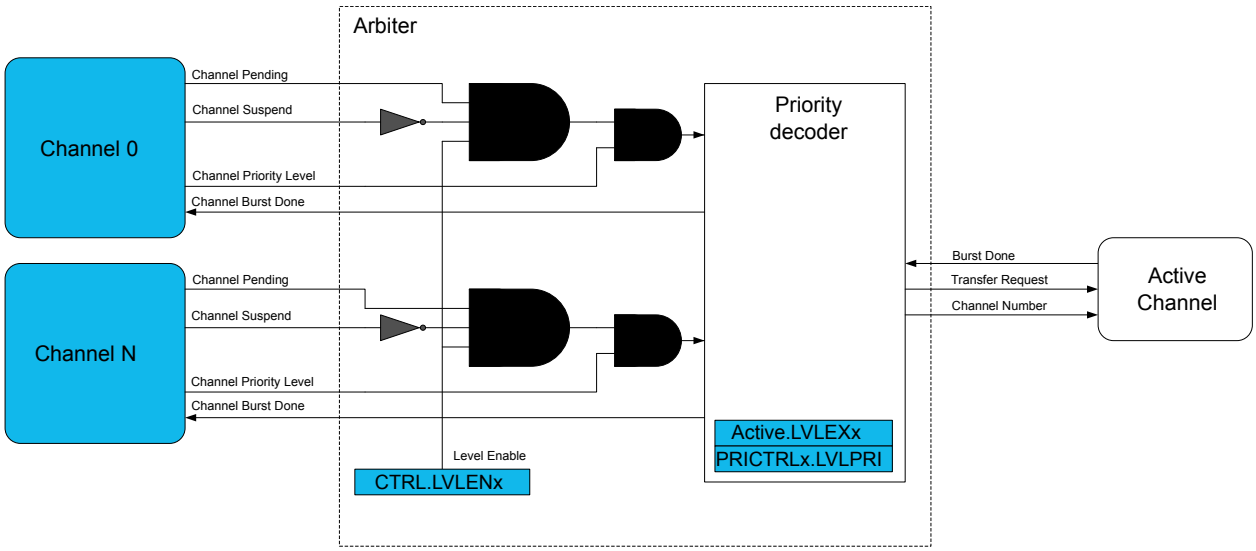
If the upcoming burst transfer is the first for the transfer request, the corresponding Busy Channel x bit in the Busy Channels register will be set (BUSYCH.BUSYCHx=1), and it will remain '1' for the subsequent granted burst transfers.

When the channel has performed its granted burst transfer(s) it will be either fed into the queue of channels with pending transfers, set to be waiting for a new transfer trigger, suspended, or disabled. This depends on the channel and block transfer configuration. If the DMA channel is fed into the queue of channels with pending transfers, the corresponding BUSYCH.BUSYCHx will remain '1'. If the DMA channel is set to wait for a new transfer trigger, suspended, or disabled, the corresponding BUSYCH.BUSYCHx will be cleared.

If a DMA channel is suspended while it has a pending transfer, it will be removed from the queue of pending channels, but the corresponding PENDCH.PENDCHx will remain set. When the same DMA channel is resumed, it will be added to the queue of pending channels again.

If a DMA channel gets disabled (CHCTRLA.ENABLE=0) while it has a pending transfer, it will be removed from the queue of pending channels, and the corresponding PENDCH.PENDCHx will be cleared.

Figure 26-4. Arbiter Overview



Priority Levels

When a channel level is pending or the channel is transferring data, the corresponding Level Executing bit is set in the Active Channel and Levels register (ACTIVE.LVLEXx).

Each DMA channel supports a 4-level priority scheme. The priority level for a channel is configured by writing to the Channel Arbitration Level bit group in the Channel Control B register (CHCTRLB.LVL). As long as all priority levels are enabled, a channel with a lower priority level number will have priority over a

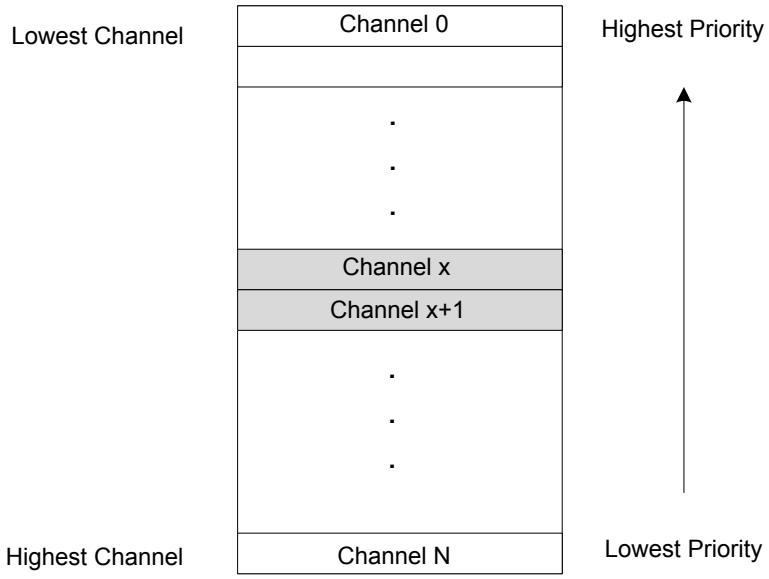
channel with a higher priority level number. Each priority level x is enabled by setting the corresponding Priority Level x Enable bit in the Control register (CTRL.LVLENx=1).

Within each priority level the DMAC's arbiter can be configured to prioritize statically or dynamically:

Static Arbitration within a priority level is selected by writing a '0' to the Level x Round-Robin Scheduling Enable bit in the Priority Control 0 register (PRICTRL0.RRLVLENx).

When static arbitration is selected, the arbiter will prioritize a low channel number over a high channel number as shown in the figure below. When using the static arbitration there is a risk of high channel numbers never being granted access as the active channel. This can be avoided using a dynamic arbitration scheme.

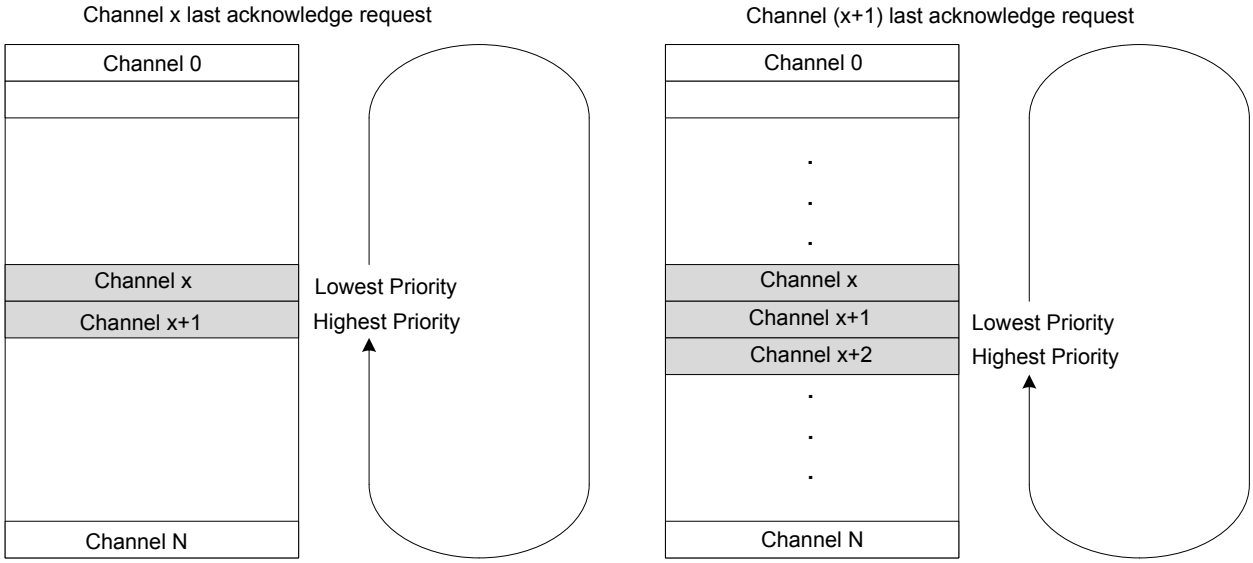
Figure 26-5. Static Priority Scheduling



Dynamic Arbitration within a priority level is selected by writing a '1' to PRICTRL0.RRLVLENx.

The dynamic arbitration scheme in the DMAC is round-robin. With the round-robin scheme, the channel number of the last channel being granted access will have the lowest priority the next time the arbiter has to grant access to a channel within the same priority level, as shown in Figure 26-6 Dynamic (Round-Robin) Priority Scheduling on page 408. The channel number of the last channel being granted access as the active channel is stored in the Level x Channel Priority Number bit group in the Priority Control 0 register (PRICTRL0.LVLPRIx) for the corresponding priority level.

Figure 26-6. Dynamic (Round-Robin) Priority Scheduling



26.6.2.5. Data Transmission

Before the DMAC can perform a data transmission, a DMA channel has to be configured and enabled, its corresponding transfer descriptor has to be initialized, and the arbiter has to grant the DMA channel access as the active channel.

Once the arbiter has granted a DMA channel access as the active channel (refer to Figure 26-1 DMAC Block Diagram on page 400) the transfer descriptor for the DMA channel will be fetched from LP SRAM using the fetch bus, and stored in the internal memory for the active channel. For a new block transfer, the transfer descriptor will be fetched from the descriptor memory section (BASEADDR); For an ongoing block transfer, the descriptor will be fetched from the write-back memory section (WRBADDR). By using the data transfer bus, the DMAC will read the data from the current source address and write it to the current destination address. For further details on how the current source and destination addresses are calculated, refer to the section on Addressing.

The arbitration procedure is performed after each burst transfer. If the current DMA channel is granted access again, the block transfer counter (BTCNT) of the internal transfer descriptor will be decremented by the number of beats in a burst, the optional output event Beat will be generated if configured and enabled, and the active channel will perform a new burst transfer. If a different DMA channel than the current active channel is granted access, the block transfer counter value will be written to the write-back section before the transfer descriptor of the newly granted DMA channel is fetched into the internal memory of the active channel.

When a block transfer has come to its end (BTCNT is zero), the Valid bit in the Block Transfer Control register will be cleared (BTCTRL.VALID=0) before the entire transfer descriptor is written to the write-back memory. The optional interrupts, Channel Transfer Complete and Channel Suspend, and the optional output event Block, will be generated if configured and enabled. After the last block transfer in a transaction, the Next Descriptor Address register (DESCADDR) will hold the value 0x00000000, and the DMA channel will either be suspended or disabled, depending on the configuration in the Block Action bit group in the Block Transfer Control register (BTCTRL.BLOCKACT). If the transaction has further block transfers pending, DESCADDR will hold the SRAM address to the next transfer descriptor to be fetched. The DMAC will fetch the next descriptor into the internal memory of the active channel and write its content to the write-back section for the channel, before the arbiter gets to choose the next active channel.

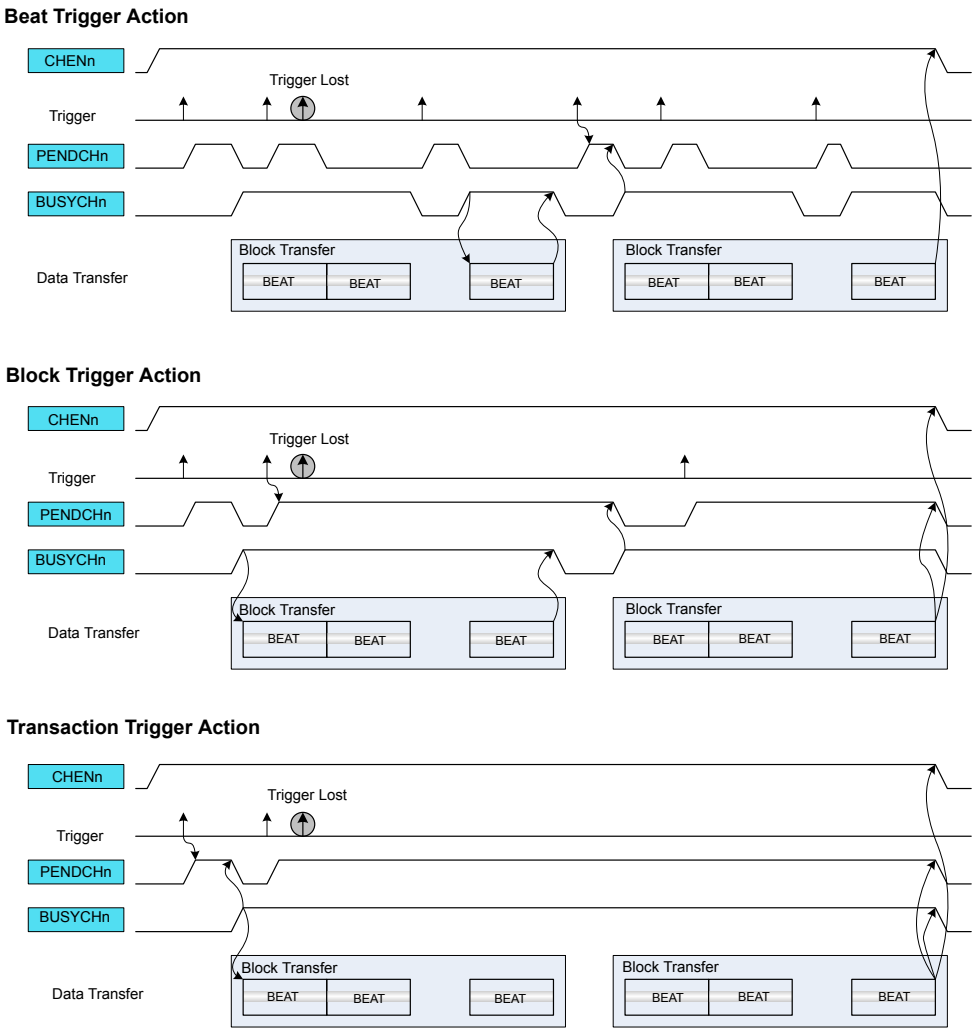
26.6.2.6. Transfer Triggers and Actions

A DMA transfer through a DMA channel can be started only when a DMA transfer request is detected, and the DMA channel has been granted access to the DMA. A transfer request can be triggered from software, from a peripheral, or from an event. There are dedicated Trigger Source selections for each DMA Channel Control B (CHCTRLB.TRIGSRC).

The trigger actions are available in the Trigger Action bit group in the Channel Control B register (CHCTRLB.TRIGACT). By default, a trigger generates a request for a block transfer operation. If a single descriptor is defined for a channel, the channel is automatically disabled when a block transfer has been completed. If a list of linked descriptors is defined for a channel, the channel is automatically disabled when the last descriptor in the list is executed. If the list still has descriptors to execute, the channel will be waiting for the next block transfer trigger. When enabled again, the channel will wait for the next block transfer trigger. The trigger actions can also be configured to generate a request for a beat transfer (CHCTRLB.TRIGACT=0x2) or transaction transfer (CHCTRLB.TRIGACT=0x3) instead of a block transfer (CHCTRLB.TRIGACT=0x0).

Figure 26-7 Trigger Action and Transfers on page 409 shows an example where triggers are used with two linked block descriptors.

Figure 26-7. Trigger Action and Transfers



If the trigger source generates a transfer request for a channel during an ongoing transfer, the new transfer request will be kept pending (CHSTATUS.PEND=1), and the new transfer can start after the

ongoing one is done. Only one pending transfer can be kept per channel. If the trigger source generates more transfer requests while one is already pending, the additional ones will be lost. All channels pending status flags are also available in the Pending Channels register (PENDCH).

When the transfer starts, the corresponding Channel Busy status flag is set in Channel Status register (CHSTATUS.BUSY). When the trigger action is complete, the Channel Busy status flag is cleared. All channel busy status flags are also available in the Busy Channels register (BUSYCH) in DMAC.

26.6.2.7. Addressing

Each block transfer needs to have both a source address and a destination address defined. The source address is set by writing the Transfer Source Address (SRCADDR) register, the destination address is set by writing the Transfer Destination Address (SRCADDR) register.

The addressing of this DMAC module can be static or incremental, for either source or destination of a block transfer, or both.

Incrementation for the source address of a block transfer is enabled by writing the Source Address Incrementation Enable bit in the Block Transfer Control register (BTCTRL.SRCINC=1). The step size of the incrementation is configurable and can be chosen by writing the Step Selection bit in the Block Transfer Control register (BTCTRL.STEPSEL=1) and writing the desired step size in the Address Increment Step Size bit group in the Block Transfer Control register (BTCTRL.STEPSIZE). If BTCTRL.STEPSEL=0, the step size for the source incrementation will be the size of one beat.

When source address incrementation is configured (BTCTRL.SRCINC=1), SRCADDR is calculated as follows:

If BTCTRL.STEPSEL=1:

$$\mathrm{SRCADDR} = \mathrm{SRCADDR}_{START} + BTCNT \cdot \left(BEATSIZE + 1\right) \cdot 2^{\mathrm{STEPSIZE}}$$

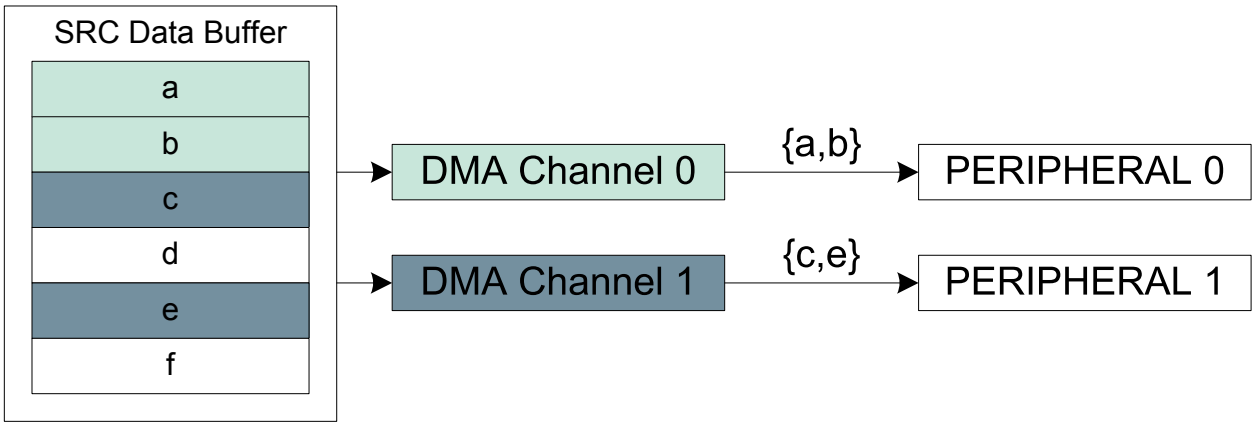
If BTCTRL.STEPSEL=0:

$$\mathrm{SRCADDR} = \mathrm{SRCADDR}_{START} + BTCNT \cdot \left(BEATSIZE + 1\right)$$

- $\mathrm{SRCADDR}_{\mathrm{START}}$ is the source address of the first beat transfer in the block transfer
- BTCNT is the initial number of beats remaining in the block transfer
- BEATSIZE is the configured number of bytes in a beat
- STEPSIZE is the configured number of beats for each incrementation

The following figure shows an example where DMA channel 0 is configured to increment the source address by one beat after each beat transfer (BTCTRL.SRCINC=1), and DMA channel 1 is configured to increment the source address by two beats (BTCTRL.SRCINC=1, BTCTRL.STEPSEL=1, and BTCTRL.STEPSIZE=0x1). As the destination address for both channels are peripherals, destination incrementation is disabled (BTCTRL.DSTINC=0).

Figure 26-8. Source Address Increment



Incrementation for the destination address of a block transfer is enabled by setting the Destination Address Incrementation Enable bit in the Block Transfer Control register (BTCTRL.DSTINC=1). The step size of the incrementation is configurable by clearing BTCTRL.STEPSEL=0 and writing BTCTRL.STEPSIZE to the desired step size. If BTCTRL.STEPSEL=1, the step size for the destination incrementation will be the size of one beat.

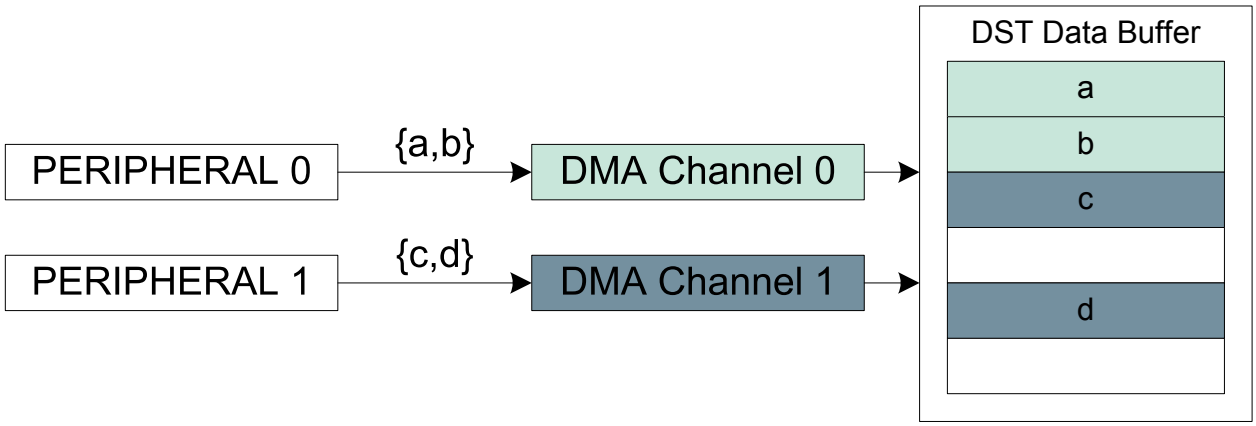
When the destination address incrementation is configured (BTCTRL.DSTINC=1), SRCADDR must be set and calculated as follows:

$DSTADDR = DSTADDR_{START} + BTCNT \bullet \left(BEATSIZE + 1\right) \bullet 2^{STEPSIZE}$	where BTCTRL.STEPSEL is zero
$DSTADDR = DSTADDR_{START} + BTCNT \bullet (BEATSIZE + 1)$	where BTCTRL.STEPSEL is one

- $DSTADDR_{START}$ is the destination address of the first beat transfer in the block transfer
- BTCNT is the initial number of beats remaining in the block transfer
- BEATSIZE is the configured number of bytes in a beat
- STEPSIZE is the configured number of beats for each incrementation

Figure 26-9 Destination Address Increment on page 412shows an example where DMA channel 0 is configured to increment destination address by one beat (BTCTRL.DSTINC=1) and DMA channel 1 is configured to increment destination address by two beats (BTCTRL.DSTINC=1, BTCTRL.STEPSEL=0, and BTCTRL.STEPSIZE=0x1). As the source address for both channels are peripherals, source incrementation is disabled (BTCTRL.SRCINC=0).

Figure 26-9. Destination Address Increment



26.6.2.8. Error Handling

If a bus error is received from an AHB slave during a DMA data transfer, the corresponding active channel is disabled and the corresponding Channel Transfer Error Interrupt flag in the Channel Interrupt Status and Clear register (CHINTFLAG.TERR) is set. If enabled, the optional transfer error interrupt is generated. The transfer counter will not be decremented and its current value is written-back in the write-back memory section before the channel is disabled.

When the DMAC fetches an invalid descriptor (BTCTRL.VALID=0) or when the channel is resumed and the DMA fetches the next descriptor with null address (DESCADDR=0x00000000), the corresponding channel operation is suspended, the Channel Suspend Interrupt Flag in the Channel Interrupt Flag Status and Clear register (CHINTFLAG.SUSP) is set, and the Channel Fetch Error bit in the Channel Status register (CHSTATUS.FERR) is set. If enabled, the optional suspend interrupt is generated.

26.6.3. Additional Features

26.6.3.1. Linked Descriptors

A transaction can consist of either a single block transfer or of several block transfers. When a transaction consist of several block transfers it is called linked descriptors.

Figure Figure 26-3 Memory Sections on page 405 illustrates how linked descriptors work. When the first block transfer is completed on DMA channel 0, the DMAC fetches the next transfer descriptor which is pointed to by the value stored in the Next Descriptor Address (DESCADDR) register of the first transfer descriptor. Fetching the next transfer descriptor (DESCADDR) is continued until the last transfer descriptor. When the block transfer for the last transfer descriptor is executed and DESCADDR=0x00000000, the transaction is terminated. For further details on how the next descriptor is fetched from LP SRAM, refer to section Data Transmission on page 408.

Adding Descriptor to the End of a List

To add a new descriptor at the end of the descriptor list, create the descriptor in LP SRAM, with DESCADDR=0x00000000 indicating that it is the new last descriptor in the list, and modify the DESCADDR value of the current last descriptor to the address of the newly created descriptor.

Modifying a Descriptor in a List

In order to add descriptors to a linked list, the following actions must be performed:

1. Enable the Suspend interrupt for the DMA channel.
2. Enable the DMA channel.
3. Reserve memory space in LP SRAM to configure a new descriptor.
4. Configure the new descriptor:

- Set the next descriptor address (DESCADDR)
- Set the destination address (DSTADDR)
- Set the source address (SRCADDR)
- Configure the block transfer control (BTCTRL) including
  - Optionally enable the Suspend block action
  - Set the descriptor VALID bit

5. Clear the VALID bit for the existing list and for the descriptor which has to be updated.
6. Read DESCADDR from the Write-Back memory.
   - If the DMA has not already fetched the descriptor which requires changes (i.e., DESCADDR is wrong):
     - Update the DESCADDR location of the descriptor from the List
     - Optionally clear the Suspend block action
     - Set the descriptor VALID bit to '1'
     - Optionally enable the Resume software command
   - If the DMA is executing the same descriptor as the one which requires changes:
     - Set the Channel Suspend software command and wait for the Suspend interrupt
     - Update the next descriptor address (DESCRADDR) in the write-back memory
     - Clear the interrupt sources and set the Resume software command
     - Update the DESCADDR location of the descriptor from the List
     - Optionally clear the Suspend block action
     - Set the descriptor VALID bit to '1'
7. Go to step 4 if needed.

Adding a Descriptor Between Existing Descriptors

To insert a new descriptor 'C' between two existing descriptors ('A' and 'B'), the descriptor currently executed by the DMA must be identified.

1. If DMA is executing descriptor B, descriptor C cannot be inserted.
2. If DMA has not started to execute descriptor A, follow the steps:
   2.1. Set the descriptor A VALID bit to '0'.
   2.2. Set the DESCADDR value of descriptor A to point to descriptor C instead of descriptor B.
   2.3. Set the DESCADDR value of descriptor C to point to descriptor B.
   2.4. Set the descriptor A VALID bit to '1'.
3. If DMA is executing descriptor A:
   3.1. Apply the software suspend command to the channel and
   3.2. Perform steps 2.1 through 2.4.
   3.3. Apply the software resume command to the channel.

26.6.3.2. Channel Suspend

The channel operation can be suspended at any time by software by writing a '1' to the Suspend command in the Command bit field of Channel Control B register (CHCTRLB.CMD). After the ongoing burst transfer is completed, the channel operation is suspended and the suspend command is automatically cleared.

When suspended, the Channel Suspend Interrupt flag in the Channel Interrupt Status and Clear register is set (CHINTFLAG.SUSP=1) and the optional suspend interrupt is generated.

By configuring the block action to suspend by writing Block Action bit group in the Block Transfer Control register (BTCTRL.BLOCKACT is 0x2 or 0x3), the DMA channel will be suspended after it has completed

a block transfer. The DMA channel will be kept enabled and will be able to receive transfer triggers, but it will be removed from the arbitration scheme.

If an invalid transfer descriptor (BTCTRL.VALID=0) is fetched from LP SRAM, the DMA channel will be suspended, and the Channel Fetch Error bit in the Channel Status register(CHASTATUS.FERR) will be set.

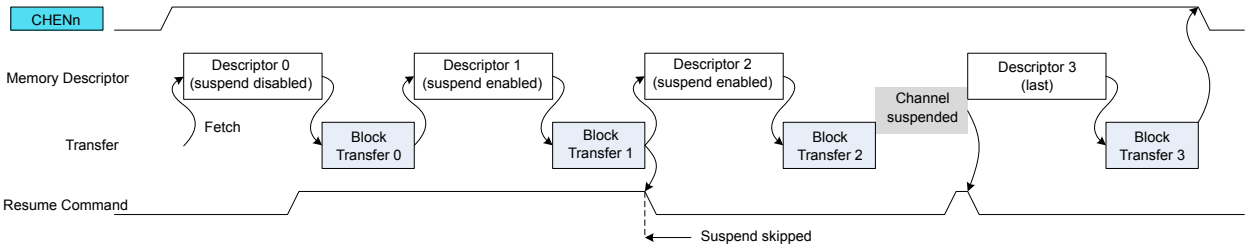
Note: Only enabled DMA channels can be suspended. If a channel is disabled when it is attempted to be suspended, the internal suspend command will be ignored.

For more details on transfer descriptors, refer to section Transfer Descriptors on page 404.

26.6.3.3. Channel Resume and Next Suspend Skip

A channel operation can be resumed by software by setting the Resume command in the Command bit field of the Channel Control B register (CHCTRLB.CMD). If the channel is already suspended, the channel operation resumes from where it previously stopped when the Resume command is detected. When the Resume command is issued before the channel is suspended, the next suspend action is skipped and the channel continues the normal operation.

Figure 26-10. Channel Suspend/Resume Operation



26.6.3.4. Event Input Actions

The event input actions are available only on the least significant DMA channels. For details on channels with event input support, refer to the in the Event system documentation.

Before using event input actions, the event controller must be configured first according to the following table, and the Channel Event Input Enable bit in the Channel Control B register (CHCTRLB.EVIE) must be written to '1'. Refer also to Events on page 420.

Table 26-1. Event Input Action

Action	CHCTRLB.EVACT	CHCTRLB.TRGSRC
None	NOACT	-
Normal Transfer	TRIG	DISABLE
Conditional Transfer on Strobe	TRIG	any peripheral
Conditional Transfer	CTRIG	
Conditional Block Transfer	CBLOCK	
Channel Suspend	SUSPEND	
Channel Resume	RESUME	
Skip Next Block Suspend	SSKIP	

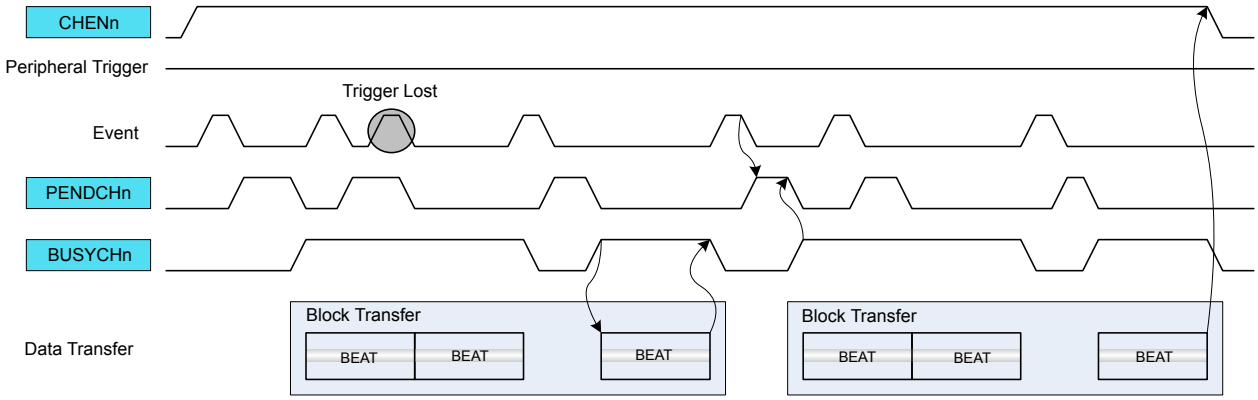
Normal Transfer

The event input is used to trigger a beat or burst transfer on peripherals.

The event is acknowledged as soon as the event is received. When received, both the Channel Pending status bit in the Channel Status register (CHSTATUS.PEND) and the corresponding Channel n bit in the Pending Channels register (PENDCH on page 442.PENDCHn) are set. If the event is received while the channel is pending, the event trigger is lost.

The figure below shows an example where beat transfers are enabled by internal events.

Figure 26-11. Beat Event Trigger Action



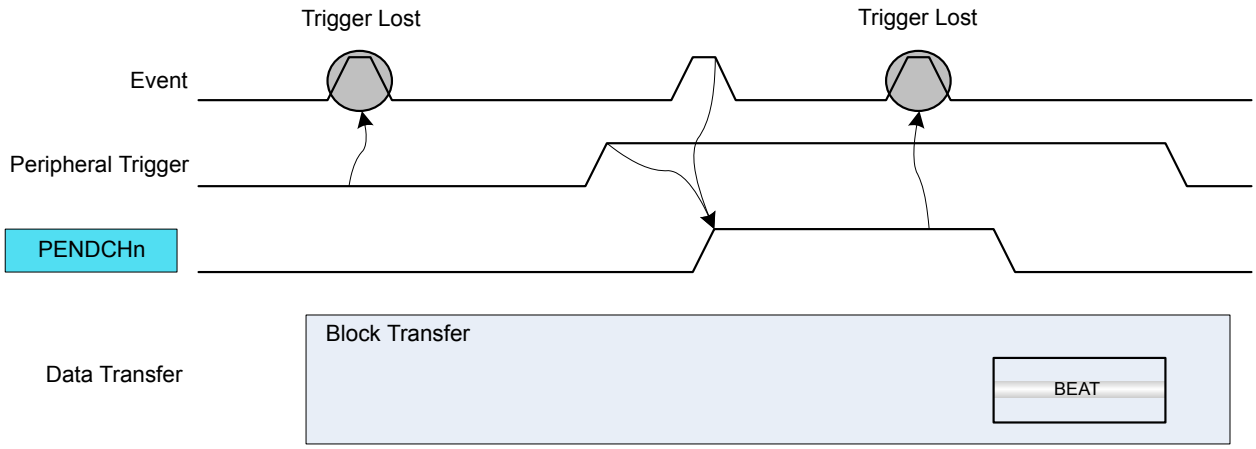
Conditional Transfer on Strobe

The event input is used to trigger a transfer on peripherals with pending transfer requests. This event action is intended to be used with peripheral triggers, e.g. for timed communication protocols or periodic transfers between peripherals: only when the peripheral trigger coincides with the occurrence of a (possibly cyclic) event the transfer is issued.

The event is acknowledged as soon as the event is received. The peripheral trigger request is stored internally when the previous trigger action is completed (i.e. the channel is not pending) and when an active event is received. If the peripheral trigger is active, the DMA will wait for an event before the peripheral trigger is internally registered. When both event and peripheral transfer trigger are active, both CHSTATUS.PEND and PENDCH on page 442.PENDCHn are set. A software trigger will now trigger a transfer.

The figure below shows an example where the peripheral beat transfer is started by a conditional strobe event action.

Figure 26-12. Periodic Event with Beat Peripheral Triggers



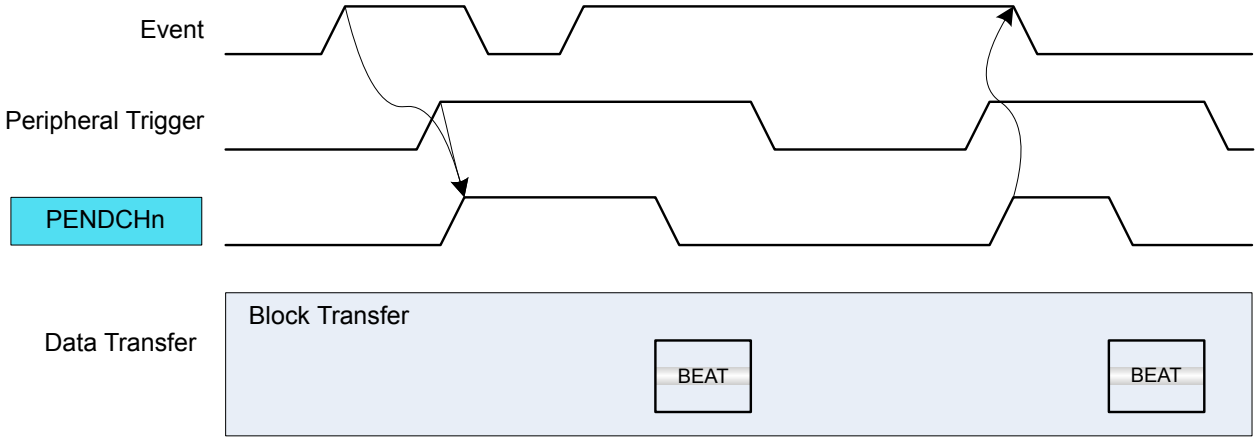
Conditional Transfer

The event input is used to trigger a conditional transfer on peripherals with pending transfer requests. As example, this type of event can be used for peripheral-to-peripheral transfers, where one peripheral is the source of event and the second peripheral is the source of the trigger.

Each peripheral trigger is stored internally when the event is received. When the peripheral trigger is stored internally, the Channel Pending status bit is set (CHSTATUS.PEND), the respective Pending Channel n Bit in the Pending Channels register is set (PENDCH on page 442.PENDCHn), and the event is acknowledged. A software trigger will now trigger a transfer.

The figure below shows an example where conditional event is enabled with peripheral beat trigger requests.

Figure 26-13. Conditional Event with Beat Peripheral Triggers



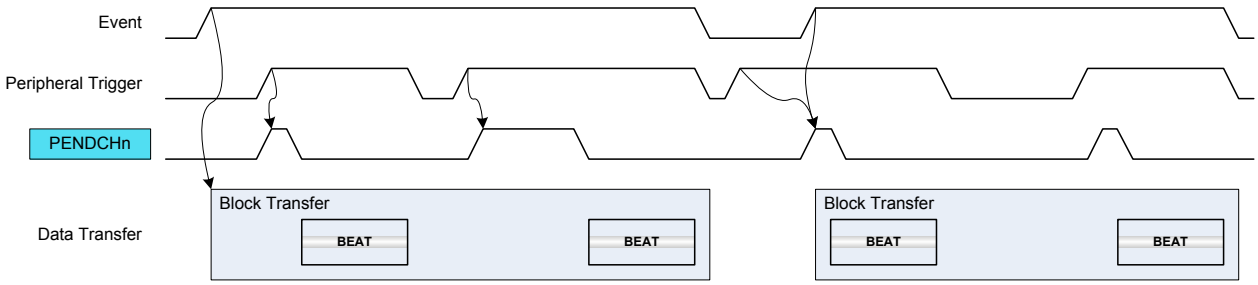
Conditional Block Transfer

The event input is used to trigger a conditional block transfer on peripherals.

Before starting transfers within a block, an event must be received. When received, the event is acknowledged when the block transfer is completed. A software trigger will trigger a transfer.

The figure below shows an example where conditional event block transfer is started with peripheral beat trigger requests.

Figure 26-14. Conditional Block Transfer with Beat Peripheral Triggers



Channel Suspend

The event input is used to suspend an ongoing channel operation. The event is acknowledged when the current AHB access is completed. For further details on Channel Suspend, refer to Channel Suspend on page 413.

Channel Resume

The event input is used to resume a suspended channel operation. The event is acknowledged as soon as the event is received and the Channel Suspend Interrupt Flag (CHINTFLAG.SUSP) is cleared. For further details refer to Channel Suspend on page 413.

Skip Next Block Suspend

This event can be used to skip the next block suspend action. If the channel is suspended before the event rises, the channel operation is resumed and the event is acknowledged. If the event rises before a suspend block action is detected, the event is kept until the next block suspend detection. When the block transfer is completed, the channel continues the operation (not suspended) and the event is acknowledged.

26.6.3.5. Event Output Selection

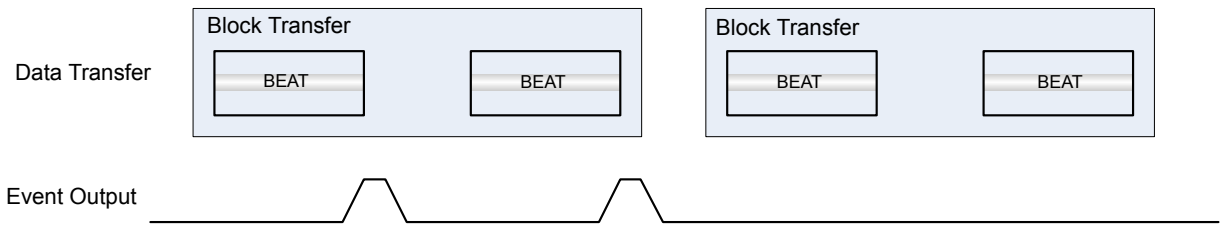
Event output selection is available only for the least significant DMA channels. The pulse width of an event output from a channel is one AHB clock cycle.

The output of channel events is enabled by writing a '1' to the Channel Event Output Enable bit in the Control B register (CHCTRLB.EVOE). The event output cause is selected by writing to the Event Output Selection bits in the Block Transfer Control register (BTCTRL.EVOSEL). It is possible to generate events after each block transfer (BTCTRL.EVOSEL=0x1) or beat transfer (BTCTRL.EVOSEL=0x3). To enable an event being generated when a transaction is complete, the block event selection must be set in the last transfer descriptor only.

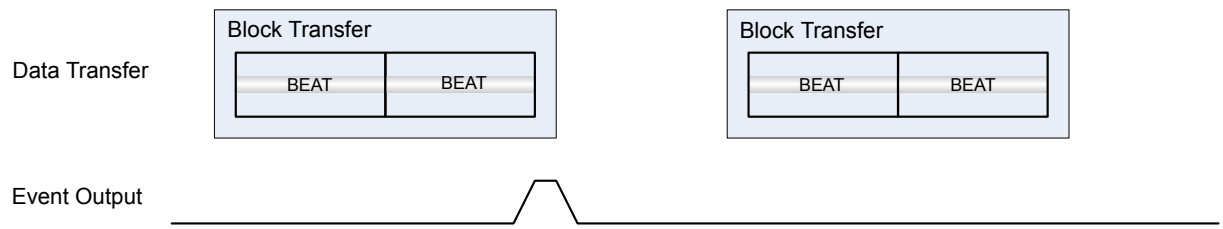
The figure Figure 26-15 Event Output Generation on page 417 shows an example where the event output generation is enabled in the first block transfer, and disabled in the second block.

Figure 26-15. Event Output Generation

Beat Event Output



Block Event Output



26.6.3.6. Aborting Transfers

Transfers on any channel can be aborted gracefully by software by disabling the corresponding DMA channel. It is also possible to abort all ongoing or pending transfers by disabling the DMAC.

When a DMA channel disable request or DMAC disable request is detected:

- Ongoing transfers of the active channel will be disabled when the ongoing beat transfer is completed and the write-back memory section is updated. This prevents transfer corruption before the channel is disabled.
- All other enabled channels will be disabled in the next clock cycle.

The corresponding Channel Enable bit in the Channel Control A register is cleared (CHCTRLA.ENABLE=0) when the channel is disabled.

The corresponding DMAC Enable bit in the Control register is cleared (CTRL.DMAENABLE=0) when the entire DMAC module is disabled.

26.6.3.7. CRC Operation

A cyclic redundancy check (CRC) is an error detection technique used to find errors in data. It is commonly used to determine whether the data during a transmission, or data present in data and program memories has been corrupted or not. A CRC takes a data stream or a block of data as input and generates a 16- or 32-bit output that can be appended to the data and used as a checksum.

When the data is received, the device or application repeats the calculation: If the new CRC result does not match the one calculated earlier, the block contains a data error. The application will then detect this and may take a corrective action, such as requesting the data to be sent again or simply not using the incorrect data.

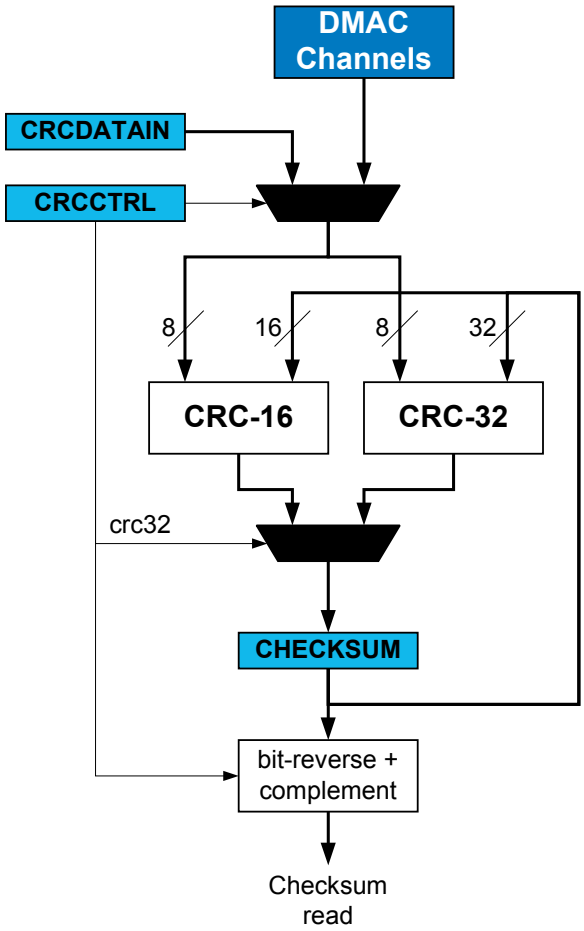
The CRC engine in DMAC supports two commonly used CRC polynomials: CRC-16 (CRC-CCITT) and CRC-32 (IEEE 802.3). Typically, applying CRC-n (CRC-16 or CRC-32) to a data block of arbitrary length will detect any single alteration that is ≤n bits in length, and will detect the fraction 1-2-n of all longer error bursts.

- CRC-16:
  - Polynomial: $x^{16}+ x^{12}+ x^5+ 1$
  - Hex value: 0x1021
- CRC-32:
  - Polynomial: $x^{32}+x^{26}+ x^{23}+ x^{22}+x^{16}+ x^{12}+ x^{11}+ x^{10}+ x^8+ x^7+ x^5+ x^4+ x^2+ x + 1$
  - Hex value: 0x04C11DB7

The data source for the CRC engine can either be one of the DMA channels or the APB bus interface, and must be selected by writing to the CRC Input Source bits in the CRC Control register (CRCCTRL.CRCSRC). The CRC engine then takes data input from the selected source and generates a checksum based on these data. The checksum is available in the CRC Checksum register (CRCCHKSUM). When CRC-32 polynomial is used, the final checksum read is bit reversed and complemented, as shown in Figure 26-16 CRC Generator Block Diagram on page 419.

The CRC polynomial is selected by writing to the CRC Polynomial Type bit in the CRC Control register (CRCCTRL.CRCPOLY), the default is CRC-16. The CRC engine operates on byte only. When the DMA is used as data source for the CRC engine, the DMA channel beat size setting will be used. When used with APB bus interface, the application must select the CRC Beat Size bit field of CRC Control register (CRCCTRL.CRCBEATSIZE). 8-, 16-, or 32-bit bus transfer access type is supported. The corresponding number of bytes will be written in the CRCDATAIN register and the CRC engine will operate on the input data in a byte by byte manner.

Figure 26-16. CRC Generator Block Diagram



CRC on DMA data CRC-16 or CRC-32 calculations can be performed on data passing through any DMA channel. Once a DMA channel is selected as the source, the CRC engine will continuously generate the CRC on the data passing through the DMA channel. The checksum is available for readout once the DMA transaction is completed or aborted. A CRC can also be generated on SRAM, Flash, or I/O memory by passing these data through a DMA channel. If the latter is done, the destination register for the DMA data can be the data input (CRCDATAIN) register in the CRC engine.

CRC using the I/O interface Before using the CRC engine with the I/O interface, the application must set the CRC Beat Size bits in the CRC Control register (CRCCTRL.CRCBEATSIZE). 8/16/32-bit bus transfer type can be selected.

CRC can be performed on any data by loading them into the CRC engine using the CPU and writing the data to the CRCDATAIN register. Using this method, an arbitrary number of bytes can be written to the register by the CPU, and CRC is done continuously for each byte. This means if a 32-bit data is written to the CRCDATAIN register the CRC engine takes four cycles to calculate the CRC. The CRC complete is signaled by a set CRCBUSY bit in the CRCSTATUS register. New data can be written only when CRCBUSY flag is not set.

26.6.4. DMA Operation

Not applicable.

26.6.5. Interrupts

The DMAC channels have the following interrupt sources:

- Transfer Complete (TCMPL): Indicates that a block transfer is completed on the corresponding channel. Refer to Data Transmission on page 408 for details.
- Transfer Error (TERR): Indicates that a bus error has occurred during a burst transfer, or that an invalid descriptor has been fetched. Refer to Error Handling on page 412 for details.
- Channel Suspend (SUSP): Indicates that the corresponding channel has been suspended. Refer to Channel Suspend on page 413 and Data Transmission on page 408 for details.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Channel Interrupt Flag Status and Clear (CHINTFLAG) register is set when the interrupt condition occurs. Each interrupt can be individually enabled by setting the corresponding bit in the Channel Interrupt Enable Set register (CHINTENSET=1), and disabled by setting the corresponding bit in the Channel Interrupt Enable Clear register (CHINTENCLR=1).

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, the DMAC is reset or the corresponding DMA channel is reset. See CHINTFLAG for details on how to clear interrupt flags. All interrupt requests are ORed together on system level to generate one combined interrupt request to the NVIC.

The user must read the Channel Interrupt Status (INTSTATUS) register to identify the channels with pending interrupts and must read the Channel Interrupt Flag Status and Clear (CHINTFLAG) register to determine which interrupt condition is present for the corresponding channel. It is also possible to read the Interrupt Pending register (INTPEND), which provides the lowest channel number with pending interrupt and the respective interrupt flags.

Note: Interrupts must be globally enabled for interrupt requests to be generated.

Related Links

Nested Vector Interrupt Controller on page 51

26.6.6. Events

The DMAC can generate the following output events:

- Channel (CH): Generated when a block transfer for a given channel has been completed, or when a beat transfer within a block transfer for a given channel has been completed. Refer to Event Output Selection on page 417 for details.

Setting the Channel Control B Event Output Enable bit (CHCTRLB.EVOE=1) enables the corresponding output event configured in the Event Output Selection bit group in the Block Transfer Control register (BTCTRL.EVOSEL). Clearing CHCTRLB.EVOE=0 disables the corresponding output event.

The DMAC can take the following actions on an input event:

- Transfer and Periodic Transfer Trigger (TRIG): normal transfer or periodic transfers on peripherals are enabled
- Conditional Transfer Trigger (CTRIG): conditional transfers on peripherals are enabled
- Conditional Block Transfer Trigger (CBLOCK): conditional block transfers on peripherals are enabled
- Channel Suspend Operation (SUSPEND): suspend a channel operation
- Channel Resume Operation (RESUME): resume a suspended channel operation
- Skip Next Block Suspend Action (SSKIP): skip the next block suspend transfer condition

Setting the Channel Control B Event Input Enable bit (CHCTRLB.EVIE=1) enables the corresponding action on input event. clearing this bit disables the corresponding action on input event. Note that several actions can be enabled for incoming events. If several events are connected to the peripheral, any enabled action will be taken for any of the incoming events. For further details on event input actions, refer to Event Input Actions on page 414.

Related Links

EVSYS – Event System on page 536

26.6.7. Sleep Mode Operation

Each DMA channel can be configured to operate in any sleep mode. To be able to run in standby, the RUNSTDBY bit in Channel Control A register (CHCTRLA.RUNSTDBY) must be written to '1'. The DMAC can wake up the device using interrupts from any sleep mode or perform actions through the Event System.

Note: In standby sleep mode, the DMAC can access the LP SRAM only when the power domain PD1 is not in retention and PM.STDBYCFG.BBIASLP=0x0. The DMAC can access the SRAM in standby sleep mode only when the power domain PD2 is not in retention and PM.STDBYCFG.BBIASHS=0x0.

26.6.8. Synchronization

Not applicable.

26.7. Register Summary

Offset	Name	Bit Pos.								
0x00	CTRL	7:0						CRCENABLE	DMAENABLE	SWRST
0x01		15:8					LVLEN3	LVLEN2	LVLEN1	LVLEN0
0x02	CRCCTRL	7:0					CRCPOLY[1:0]		CRCBEATSIZE[1:0]	
0x03		15:8			CRCSRC[5:0]					
0x04	CRCDATAIN	7:0	CRCDATAIN[7:0]							
0x05		15:8	CRCDATAIN[15:8]							
0x06		23:16	CRCDATAIN[23:16]							
0x07		31:24	CRCDATAIN[31:24]							
0x08	CRCCHKSUM	7:0	CRCCHKSUM[7:0]							
0x09		15:8	CRCCHKSUM[15:8]							
0x0A		23:16	CRCCHKSUM[23:16]							
0x0B		31:24	CRCCHKSUM[31:24]							
0x0C	CRCSTATUS	7:0							CRCZERO	CRCBUSY
0x0D	DBGCTRL	7:0								DBGRUN
0x0E	QOSCTRL	7:0			DQOS[1:0]		FQOS[1:0]		WRBQOS[1:0]	
0x0F	Reserved									
0x10	SWTRIGCTRL	7:0	SWTRIG7	SWTRIG6	SWTRIG5	SWTRIG4	SWTRIG3	SWTRIG2	SWTRIG1	SWTRIG0
0x11		15:8	SWTRIG15	SWTRIG14	SWTRIG13	SWTRIG12	SWTRIG11	SWTRIG10	SWTRIG9	SWTRIG8
0x12		23:16								
0x13		31:24								
0x14	PRICTRL0	7:0	RRLVLEN0				LVLPRI0[3:0]			
0x15		15:8	RRLVLEN1				LVLPRI1[3:0]			
0x16		23:16	RRLVLEN2				LVLPRI2[3:0]			
0x17		31:24	RRLVLEN3				LVLPRI3[3:0]			
0x18 ... 0x1F	Reserved									
0x20	INTPEND	7:0					ID[3:0]			
0x21		15:8	PEND	BUSY	FERR			SUSP	TCMPL	TERR
0x22 ... 0x23	Reserved									
0x24	INTSTATUS	7:0	CHINT7	CHINT6	CHINT5	CHINT4	CHINT3	CHINT2	CHINT1	CHINT0
0x25		15:8	CHINT15	CHINT14	CHINT13	CHINT12	CHINT11	CHINT10	CHINT9	CHINT8
0x26		23:16								
0x27		31:24								
0x28	BUSYCH	7:0	BUSYCH7	BUSYCH6	BUSYCH5	BUSYCH4	BUSYCH3	BUSYCH2	BUSYCH1	BUSYCH0
0x29		15:8	BUSYCH15	BUSYCH14	BUSYCH13	BUSYCH12	BUSYCH11	BUSYCH10	BUSYCH9	BUSYCH8
0x2A		23:16								
0x2B		31:24								
0x2C	PENDCH	7:0	PENDCH7	PENDCH6	PENDCH5	PENDCH4	PENDCH3	PENDCH2	PENDCH1	PENDCH0
0x2D		15:8	PENDCH15	PENDCH14	PENDCH13	PENDCH12	PENDCH11	PENDCH10	PENDCH9	PENDCH8
0x2E		23:16								
0x2F		31:24								

Offset	Name	Bit Pos.								
0x30	ACTIVE	7:0					LVLEX3	LVLEX2	LVLEX1	LVLEX0
0x31		15:8	ABUSY			ID[4:0]				
0x32		23:16	BTCNT[7:0]							
0x33		31:24	BTCNT[15:8]							
0x34	BASEADDR	7:0	BASEADDR[7:0]							
0x35		15:8	BASEADDR[15:8]							
0x36		23:16	BASEADDR[23:16]							
0x37		31:24	BASEADDR[31:24]							
0x38	WRBADDR	7:0	WRBADDR[7:0]							
0x39		15:8	WRBADDR[15:8]							
0x3A		23:16	WRBADDR[23:16]							
0x3B		31:24	WRBADDR[31:24]							
0x3C ... 0x3E	Reserved									
0x3F	CHID	7:0					ID[3:0]			
0x40	CHCTRLA	7:0		RUNSTDBY					ENABLE	SWRST
0x41 ... 0x43	Reserved									
0x44	CHCTRLB	7:0		LVL[1:0]		EVOE	EVIE	EVACT[2:0]		
0x45		15:8			TRIGSRC[5:0]					
0x46		23:16	TRIGACT[1:0]							
0x47		31:24							CMD[1:0]	
0x48 ... 0x4B	Reserved									
0x4C	CHINTENCLR	7:0						SUSP	TCMPL	TERR
0x4D	CHINTENSET	7:0						SUSP	TCMPL	TERR
0x4E	CHINTFLAG	7:0						SUSP	TCMPL	TERR
0x4F	CHSTATUS	7:0						FERR	BUSY	PEND

26.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection on page 401.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

26.8.1. Control

Name: CTRL
Offset: 0x00
Reset: 0x00X0
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
					LVLEN3	LVLEN2	LVLEN1	LVLEN0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
						CRCENABLE	DMAENABLE	SWRST
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – CRCENABLE: CRC Enable
Writing a '0' to this bit will disable the CRC calculation when the CRC Status Busy flag is cleared (CRCSTATUS. CRCBUSY). The bit is zero when the CRC is disabled.

Writing a '1' to this bit will enable the CRC calculation.

Value	Description
0	The CRC calculation is disabled.
1	The CRC calculation is enabled.

Bit 1 – DMAENABLE: DMA Enable
Setting this bit will enable the DMA module.

Writing a '0' to this bit will disable the DMA module. When writing a '0' during an ongoing transfer, the bit will not be cleared until the internal data transfer buffer is empty and the DMA transfer is aborted. The internal data transfer buffer will be empty once the ongoing burst transfer is completed.

This bit is not enable-protected.

Value	Description
0	The peripheral is disabled.
1	The peripheral is enabled.

Bit 0 – SWRST: Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit when both the DMAC and the CRC module are disabled (DMAENABLE and CRCENABLE are '0') resets all registers in the DMAC (except DBGCTRL) to their initial state. If either the DMAC or CRC module is enabled, the Reset request will be ignored and the DMAC will return an access error.

Value	Description
0	There is no Reset operation ongoing.
1	A Reset operation is ongoing.

Bits 11,10,9,8 – LVLENx: Priority Level x Enable [x=3..0]
When this bit is set, all requests with the corresponding level will be fed into the arbiter block. When cleared, all requests with the corresponding level will be ignored.

For details on arbitration schemes, refer to the Arbitration section.

These bits are not enable-protected.

Value	Description
0	Transfer requests for Priority level x will not be handled.
1	Transfer requests for Priority level x will be handled.

26.8.2. CRC Control

Name: CRCCTRL
Offset: 0x02
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
			CRCSRC[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
					CRCPOLY[1:0]		CRCBEATSIZE[1:0]	
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 13:8 – CRCSRC[5:0]: CRC Input Source
These bits select the input source for generating the CRC, as shown in the table below. The selected source is locked until either the CRC generation is completed or the CRC module is disabled. This means the CRCSRC cannot be modified when the CRC operation is ongoing. The lock is signaled by the CRCBUSY status bit. CRC generation complete is generated and signaled from the selected source when used with the DMA channel.

Value	Name	Description
0x00	NOACT	No action
0x01	IO	I/O interface
0x02-0x1F	-	Reserved
0x20	CHN	DMA channel 0
0x21	CHN	DMA channel 1
0x22	CHN	DMA channel 2
0x23	CHN	DMA channel 3
0x24	CHN	DMA channel 4
0x25	CHN	DMA channel 5
0x26	CHN	DMA channel 6
0x27	CHN	DMA channel 7
0x28	CHN	DMA channel 8
0x29	CHN	DMA channel 9
0x2A	CHN	DMA channel 10
0x2B	CHN	DMA channel 11
0x2C	CHN	DMA channel 12

Value	Name	Description
0x2D	CHN	DMA channel 13
0x2E	CHN	DMA channel 14
0x2F	CHN	DMA channel 15
0x30	CHN	DMA channel 16
0x31	CHN	DMA channel 17
0x32	CHN	DMA channel 18
0x33	CHN	DMA channel 19
0x34	CHN	DMA channel 20
0x35	CHN	DMA channel 21
0x36	CHN	DMA channel 22
0x37	CHN	DMA channel 23
0x38	CHN	DMA channel 24
0x39	CHN	DMA channel 25
0x3A	CHN	DMA channel 26
0x3B	CHN	DMA channel 27
0x3C	CHN	DMA channel 28
0x3D	CHN	DMA channel 29
0x3E	CHN	DMA channel 30
0x3F	CHN	DMA channel 31

Bits 3:2 – CRCPOLY[1:0]: CRC Polynomial Type
These bits define the size of the data transfer for each bus access when the CRC is used with I/O interface, as shown in the table below.

Value	Name	Description
0x0	CRC16	CRC-16 (CRC-CCITT)
0x1	CRC32	CRC32 (IEEE 802.3)
0x2-0x3		Reserved

Bits 1:0 – CRCBEATSIZE[1:0]: CRC Beat Size
These bits define the size of the data transfer for each bus access when the CRC is used with I/O interface.

Value	Name	Description
0x0	BYTE	8-bit bus transfer
0x1	HWORD	16-bit bus transfer
0x2	WORD	32-bit bus transfer

Value	Name	Description
0x3		Reserved

26.8.3. CRC Data Input

Name: CRCDATAIN
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	CRCDATAIN[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CRCDATAIN[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CRCDATAIN[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CRCDATAIN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CRCDATAIN[31:0]: CRC Data Input
These bits store the data for which the CRC checksum is computed. A new CRC Checksum is ready (CRCBEAT+ 1) clock cycles after the CRCDATAIN register is written.

26.8.4. CRC Checksum

The CRCCHKSUM represents the 16- or 32-bit checksum value and the generated CRC. The register is reset to zero by default, but it is possible to reset all bits to one by writing the CRCCHKSUM register directly. It is possible to write this register only when the CRC module is disabled. If CRC-32 is selected and the CRC Status Busy flag is cleared (i.e., CRC generation is completed or aborted), the bit reversed (bit 31 is swapped with bit 0, bit 30 with bit 1, etc.) and complemented result will be read from CRCCHKSUM. If CRC-16 is selected or the CRC Status Busy flag is set (i.e., CRC generation is ongoing), CRCCHKSUM will contain the actual content.

Name: CRCCHKSUM
Offset: 0x08
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
	CRCCHKSUM[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CRCCHKSUM[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CRCCHKSUM[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CRCCHKSUM[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CRCCHKSUM[31:0]: CRC Checksum

These bits store the generated CRC result. The 16 MSB bits are always read zero when CRC-16 is enabled.

26.8.5. CRC Status

Name: CRCSTATUS
Offset: 0x0C
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
							CRCZERO	CRCBUSY
Access							R	R/W
Reset							0	0

Bit 1 – CRCZERO: CRC Zero
This bit is cleared when a new CRC source is selected.

This bit is set when the CRC generation is complete and the CRC Checksum is zero.

When running CRC-32 and appending the checksum at the end of the packet (as little endian), the final checksum should be 0x2144df1c, and not zero. However, if the checksum is complemented before it is appended (as little endian) to the data, the final result in the checksum register will be zero. See the description of CRCCHKSUM to read out different versions of the checksum.

Bit 0 – CRCBUSY: CRC Module Busy
This flag is cleared by writing a one to it when used with I/O interface. When used with a DMA channel, the bit is set when the corresponding DMA channel is enabled, and cleared when the corresponding DMA channel is disabled. This register bit cannot be cleared by the application when the CRC is used with a DMA channel.

This bit is set when a source configuration is selected and as long as the source is using the CRC module.

26.8.6. Debug Control

Name: DBGCTRL
Offset: 0x0D
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGRUN
Access								R/W
Reset								0

Bit 0 – DBGRUN: Debug Run
This bit is not reset by a software reset.

This bit controls the functionality when the CPU is halted by an external debugger.

Value	Description
0	The DMAC is halted when the CPU is halted by an external debugger.
1	The DMAC continues normal operation when the CPU is halted by an external debugger.

26.8.7. Quality of Service Control

Name: QOSCTRL
Offset: 0x0E
Reset: 0x2A
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
			DQOS[1:0]		FQOS[1:0]		WRBQOS[1:0]	
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			1	0	1	0	1	0

Bits 5:4 – DQOS[1:0]: Data Transfer Quality of Service
These bits define the memory priority access during the data transfer operation.

DQOS[1:0]	Name	Description
0x0	DISABLE	Background (no sensitive operation)
0x1	LOW	Sensitive Bandwidth
0x2	MEDIUM	Sensitive Latency
0x3	HIGH	Critical Latency

Bits 3:2 – FQOS[1:0]: Fetch Quality of Service
These bits define the memory priority access during the fetch operation.

FQOS[1:0]	Name	Description
0x0	DISABLE	Background (no sensitive operation)
0x1	LOW	Sensitive Bandwidth
0x2	MEDIUM	Sensitive Latency
0x3	HIGH	Critical Latency

Bits 1:0 – WRBQOS[1:0]: Write-Back Quality of Service
These bits define the memory priority access during the write-back operation.

WRBQOS[1:0]	Name	Description
0x0	DISABLE	Background (no sensitive operation)
0x1	LOW	Sensitive Bandwidth
0x2	MEDIUM	Sensitive Latency
0x3	HIGH	Critical Latency

26.8.8. Software Trigger Control

Name: SWTRIGCTRL
Offset: 0x10
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	SWTRIG15	SWTRIG14	SWTRIG13	SWTRIG12	SWTRIG11	SWTRIG10	SWTRIG9	SWTRIG8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SWTRIG7	SWTRIG6	SWTRIG5	SWTRIG4	SWTRIG3	SWTRIG2	SWTRIG1	SWTRIG0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – SWTRIGn: Channel n Software Trigger [n = 15..0]
This bit is cleared when the Channel Pending bit in the Channel Status register (CHSTATUS.PEND) for the corresponding channel is either set, or by writing a '1' to it.

This bit is set if CHSTATUS.PEND is already '1' when writing a '1' to that bit.

Writing a '0' to this bit will clear the bit.

Writing a '1' to this bit will generate a DMA software trigger on channel x, if CHSTATUS.PEND=0 for channel x. CHSTATUS.PEND will be set and SWTRIGn will remain cleared.

26.8.9. Priority Control 0

Name: PRICTRL0
Offset: 0x14
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	RRLVLEN3				LVLPRI3[3:0]			
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RRLVLEN2				LVLPRI2[3:0]			
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RRLVLEN1				LVLPRI1[3:0]			
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RRLVLEN0				LVLPRI0[3:0]			
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit 31 – RRLVLEN3: Level 3 Round-Robin Arbitration Enable
This bit controls which arbitration scheme is selected for DMA channels with priority level 3. For details on arbitration schemes, refer to Arbitration on page 406.

Value	Description
0	Static arbitration scheme for channels with level 3 priority.
1	Round-robin arbitration scheme for channels with level 3 priority.

Bits 27:24 – LVLPRI3[3:0]: Level 3 Channel Priority Number
When round-robin arbitration is enabled (PRICTRL0.RRLVLEN3=1) for priority level 3, this register holds the channel number of the last DMA channel being granted access as the active channel with priority level 3.

When static arbitration is enabled (PRICTRL0.RRLVLEN3=0) for priority level 3, and the value of this bit group is non-zero, it will affect the static priority scheme. If the value of this bit group is x, channel x will have the highest priority. The priority will decrease as the channel number increases from x to n, where n is the maximum number of channels. Channel n has higher priority than channel 0, and the priority will continue to decrease from channel 0 to channel (x-1).

This bit group is not reset when round-robin arbitration gets disabled (PRICTRL0.RRLVLEN3 written to '0').

Bit 23 – RRLVLEN2: Level 2 Round-Robin Arbitration Enable
This bit controls which arbitration scheme is selected for DMA channels with priority level 2. For details on arbitration schemes, refer to Arbitration on page 406.

Value	Description
0	Static arbitration scheme for channels with level 2 priority.
1	Round-robin arbitration scheme for channels with level 2 priority.

Bits 19:16 – LVLPRI2[3:0]: Level 2 Channel Priority Number
When round-robin arbitration is enabled (PRICTRL0.RRLVLEN2=1) for priority level 2, this register holds the channel number of the last DMA channel being granted access as the active channel with priority level 2.

When static arbitration is enabled (PRICTRL0.RRLVLEN2=0) for priority level 2, and the value of this bit group is non-zero, it will affect the static priority scheme. If the value of this bit group is x, channel x will have the highest priority. The priority will decrease as the channel number increases from x to n, where n is the maximum number of channels. Channel n has higher priority than channel 0, and the priority will continue to decrease from channel 0 to channel (x-1).

This bit group is not reset when round-robin arbitration gets disabled (PRICTRL0.RRLVLEN2 written to '0').

Bit 15 – RRLVLEN1: Level 1 Round-Robin Scheduling Enable
For details on arbitration schemes, refer to Arbitration on page 406.

Value	Description
0	Static arbitration scheme for channels with level 1 priority.
1	Round-robin arbitration scheme for channels with level 1 priority.

Bits 11:8 – LVLPRI1[3:0]: Level 1 Channel Priority Number
When round-robin arbitration is enabled (PRICTRL0.RRLVLEN1=1) for priority level 1, this register holds the channel number of the last DMA channel being granted access as the active channel with priority level 1.

When static arbitration is enabled (PRICTRL0.RRLVLEN1=0) for priority level 1, and the value of this bit group is non-zero, it will affect the static priority scheme. If the value of this bit group is x, channel x will have the highest priority. The priority will decrease as the channel number increases from x to n, where n is the maximum number of channels. Channel n has higher priority than channel 0, and the priority will continue to decrease from channel 0 to channel (x-1).

This bit group is not reset when round-robin arbitration gets disabled (PRICTRL0.RRLVLEN1 written to '0').

Bit 7 – RRLVLEN0: Level 0 Round-Robin Scheduling Enable
For details on arbitration schemes, refer to Arbitration on page 406.

Value	Description
0	Static arbitration scheme for channels with level 0 priority.
1	Round-robin arbitration scheme for channels with level 0 priority.

Bits 3:0 – LVLPRI0[3:0]: Level 0 Channel Priority Number

When round-robin arbitration is enabled (PRICTRL0.RRLVLEN0=1) for priority level 0, this register holds the channel number of the last DMA channel being granted access as the active channel with priority level 0.

When static arbitration is enabled (PRICTRL0.RRLVLEN0=0) for priority level 0, and the value of this bit group is non-zero, it will affect the static priority scheme. If the value of this bit group is x, channel x will have the highest priority. The priority will decrease as the channel number increases from x to n, where n is the maximum number of channels. Channel n has higher priority than channel 0, and the priority will continue to decrease from channel 0 to channel (x-1).

This bit group is not reset when round-robin arbitration gets disabled (PRICTRL0.RRLVLEN0 written to '0').

26.8.10. Interrupt Pending

This register allows the user to identify the lowest DMA channel with pending interrupt.

Name: INTPEND
Offset: 0x20
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
	PEND	BUSY	FERR			SUSP	TCMPL	TERR
Access	R	R	R			R/W	R/W	R/W
Reset	0	0	0			0	0	0

Bit	7	6	5	4	3	2	1	0
					ID[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 15 – PEND: Pending
This bit will read '1' when the channel selected by Channel ID field (ID) is pending.

Bit 14 – BUSY: Busy
This bit will read '1' when the channel selected by Channel ID field (ID) is busy.

Bit 13 – FERR: Fetch Error
This bit will read '1' when the channel selected by Channel ID field (ID) fetched an invalid descriptor.

Bit 10 – SUSP: Channel Suspend
This bit will read '1' when the channel selected by Channel ID field (ID) has pending Suspend interrupt.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Channel ID (ID) Suspend interrupt flag.

Bit 9 – TCMPL: Transfer Complete
This bit will read '1' when the channel selected by Channel ID field (ID) has pending Transfer Complete interrupt.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Channel ID (ID) Transfer Complete interrupt flag.

Bit 8 – TERR: Transfer Error
This bit is read one when the channel selected by Channel ID field (ID) has pending Transfer Error interrupt.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Channel ID (ID) Transfer Error interrupt flag.

Bits 3:0 – ID[3:0]: Channel ID
These bits store the lowest channel number with pending interrupts. The number is valid if Suspend (SUSP), Transfer Complete (TCMPL) or Transfer Error (TERR) bits are set. The Channel ID field is

refreshed when a new channel (with channel number less than the current one) with pending interrupts is detected, or when the application clears the corresponding channel interrupt sources. When no pending channels interrupts are available, these bits will always return zero value when read.

When the bits are written, indirect access to the corresponding Channel Interrupt Flag register is enabled.

26.8.11. Interrupt Status

Name: INTSTATUS
Offset: 0x24
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	CHINT15	CHINT14	CHINT13	CHINT12	CHINT11	CHINT10	CHINT9	CHINT8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CHINT7	CHINT6	CHINT5	CHINT4	CHINT3	CHINT2	CHINT1	CHINT0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – CHINTn: Channel n Pending Interrupt [n=15..0]
This bit is set when Channel n has a pending interrupt/the interrupt request is received.

This bit is cleared when the corresponding Channel n interrupts are disabled or the interrupts sources are cleared.

26.8.12. Busy Channels

Name: BUSYCH
Offset: 0x28
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	BUSYCH15	BUSYCH14	BUSYCH13	BUSYCH12	BUSYCH11	BUSYCH10	BUSYCH9	BUSYCH8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BUSYCH7	BUSYCH6	BUSYCH5	BUSYCH4	BUSYCH3	BUSYCH2	BUSYCH1	BUSYCH0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – BUSYCHn: Busy Channel n [x=15..0]
This bit is cleared when the channel trigger action for DMA channel n is complete, when a bus error for DMA channel n is detected, or when DMA channel n is disabled.

This bit is set when DMA channel n starts a DMA transfer.

26.8.13. Pending Channels

Name: PENDCH
Offset: 0x2C
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	PENDCH15	PENDCH14	PENDCH13	PENDCH12	PENDCH11	PENDCH10	PENDCH9	PENDCH8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PENDCH7	PENDCH6	PENDCH5	PENDCH4	PENDCH3	PENDCH2	PENDCH1	PENDCH0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – PENDCHn: Pending Channel n [n=15..0]
This bit is cleared when trigger execution defined by channel trigger action settings for DMA channel n is started, when a bus error for DMA channel n is detected or when DMA channel n is disabled. For details on trigger action settings, refer to CHCTRLB.TRIGACT.

This bit is set when a transfer is pending on DMA channel n.

26.8.14. Active Channel and Levels

Name: ACTIVE
Offset: 0x30
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	BTCNT[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	BTCNT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ABUSY			ID[4:0]				
Access	R			R	R	R	R	R
Reset	0			0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
					LVLEX3	LVLEX2	LVLEX1	LVLEX0
Access					R	R	R	R
Reset					0	0	0	0

Bits 31:16 – BTCNT[15:0]: Active Channel Block Transfer Count
These bits hold the 16-bit block transfer count of the ongoing transfer. This value is stored in the active channel and written back in the corresponding Write-Back channel memory location when the arbiter grants a new channel access. The value is valid only when the active channel active busy flag (ABUSY) is set.

Bit 15 – ABUSY: Active Channel Busy
This bit is cleared when the active transfer count is written back in the write-back memory section.

This bit is set when the next descriptor transfer count is read from the write-back memory section.

Bits 12:8 – ID[4:0]: Active Channel ID
These bits hold the channel index currently stored in the active channel registers. The value is updated each time the arbiter grants a new channel transfer access request.

Bits 3,2,1,0 – LVLEXx: Level x Channel Trigger Request Executing [x=3..0]
This bit is set when a level-x channel trigger request is executing or pending.

This bit is cleared when no request is pending or being executed.

26.8.15. Descriptor Memory Section Base Address

Name: BASEADDR
Offset: 0x34
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
	BASEADDR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	BASEADDR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	BASEADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BASEADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – BASEADDR[31:0]: Descriptor Memory Base Address
These bits store the Descriptor memory section base address. The value must be 128-bit aligned.

26.8.16. Write-Back Memory Section Base Address

Name: WRBADDR
Offset: 0x38
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
	WRBADDR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WRBADDR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WRBADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	WRBADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – WRBADDR[31:0]: Write-Back Memory Base Address
These bits store the Write-Back memory base address. The value must be 128-bit aligned.

26.8.17. Channel ID

Name: CHID
Offset: 0x3F
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
					ID[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 3:0 – ID[3:0]: Channel ID
These bits define the channel number that will be affected by the channel registers (CH*). Before reading or writing a channel register, the channel ID bit group must be written first.

26.8.18. Channel Control A

This register affects the DMA channel that is selected in the Channel ID register (CHID.ID).

Name: CHCTRLA
Offset: 0x40
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
		RUNSTDBY					ENABLE	SWRST
Access		R/W					R/W	R/W
Reset		0					0	0

Bit 6 – RUNSTDBY: Channel run in standby
This bit is used to keep the DMAC channel running in standby mode.

This bit is not enable-protected.

Value	Description
0	The DMAC channel is halted in standby.
1	The DMAC channel continues to run in standby.

Bit 1 – ENABLE: Channel Enable
Writing a '0' to this bit during an ongoing transfer, the bit will not be cleared until the internal data transfer buffer is empty and the DMA transfer is aborted. The internal data transfer buffer will be empty once the ongoing burst transfer is completed.

Writing a '1' to this bit will enable the DMA channel.

This bit is not enable-protected.

Value	Description
0	DMA channel is disabled.
1	DMA channel is enabled.

Bit 0 – SWRST: Channel Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets the channel registers to their initial state. The bit can be set when the channel is disabled (ENABLE=0). Writing a '1' to this bit will be ignored as long as ENABLE=1. This bit is automatically cleared when the reset is completed.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

26.8.19. Channel Control B

This register affects the DMA channel that is selected in the Channel ID register (CHID.ID).

Name: CHCTRLB
Offset: 0x44
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
							CMD[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
	TRIGACT[1:0]							
Access	R/W	R/W						
Reset	0	0						

Bit	15	14	13	12	11	10	9	8
			TRIGSRC[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		LVL[1:0]		EVOE	EVIE	EVACT[2:0]		
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 25:24 – CMD[1:0]: Software Command

These bits define the software commands. Refer to Channel Suspend on page 413 and Channel Resume and Next Suspend Skip on page 414.

These bits are not enable-protected.

CMD[1:0]	Name	Description
0x0	NOACT	No action
0x1	SUSPEND	Channel suspend operation
0x2	RESUME	Channel resume operation
0x3	-	Reserved

Bits 23:22 – TRIGACT[1:0]: Trigger Action

These bits define the trigger action used for a transfer.

TRIGACT[1:0]	Name	Description
0x0	BLOCK	One trigger required for each block transfer
0x1	-	Reserved

TRIGACT[1:0]	Name	Description
0x2	BEAT	One trigger required for each beat transfer
0x3	TRANSACTION	One trigger required for each transaction

Bits 13:8 – TRIGSRC[5:0]: Trigger Source
These bits define the peripheral trigger which is source of the transfer. For details on trigger selection and trigger modes, refer to Transfer Triggers and Actions and CHCTRLB.TRIGACT.

Value	Name	Description
0x00	DISABLE	Only software/event triggers
0x01	SERCOM0 RX	SERCOM0 RX Trigger
0x02	SERCOM0 TX	SERCOM0 TX Trigger
0x03	SERCOM1 RX	SERCOM1 RX Trigger
0x04	SERCOM1 TX	SERCOM1 TX Trigger
0x05	SERCOM2 RX	SERCOM2 RX Trigger
0x06	SERCOM2 TX	SERCOM2 TX Trigger
0x07	SERCOM3 RX	SERCOM3 RX Trigger
0x08	SERCOM3 TX	SERCOM3 TX Trigger
0x09	SERCOM4 RX	SERCOM4 RX Trigger
0x0A	SERCOM4 TX	SERCOM4 TX Trigger
0x0B	TCC0 OVF	TCC0 Overflow Trigger
0x0C	TCC0 MC0	TCC0 Match/Compare 0 Trigger
0x0D	TCC0 MC1	TCC0 Match/Compare 1 Trigger
0x0E	TCC0 MC2	TCC0 Match/Compare 2 Trigger
0x0F	TCC0 MC3	TCC0 Match/Compare 3 Trigger
0x10	TCC1 OVF	TCC1 Overflow Trigger
0x11	TCC1 MC0	TCC1 Match/Compare 0 Trigger
0x12	TCC1 MC1	TCC1 Match/Compare 1 Trigger
0x13	TCC2 OVF	TCC2 Overflow Trigger
0x14	TCC2 MC0	TCC2 Match/Compare 0 Trigger
0x15	TCC2 MC1	TCC2 Match/Compare 1 Trigger
0x16	TC0 OVF	TC0 Overflow Trigger
0x17	TC0 MC0	TC0 Match/Compare 0 Trigger
0x18	TC0 MC1	TC0 Match/Compare 1 Trigger
0x19	TC1 OVF	TC1 Overflow Trigger
0x1A	TC1 MC0	TC1 Match/Compare 0 Trigger

Value	Name	Description
0x1B	TC1 MC1	TC1 Match/Compare 1 Trigger
0x1C	TC2 OVF	TC2 Overflow Trigger
0x1D	TC2 MC0	TC2 Match/Compare 0 Trigger
0x1E	TC2 MC1	TC2 Match/Compare 1 Trigger
0x1F	TC3 OVF	TC3 Overflow Trigger
0x20	TC3 MC0	TC3 Match/Compare 0 Trigger
0x21	TC3 MC1	TC3 Match/Compare 1 Trigger
0x22	TC4 OVF	TC4 Overflow Trigger
0x23	TC4 MC0	TC4 Match/Compare 0 Trigger
0x24	TC4 MC1	TC4 Match/Compare 1 Trigger
0x25	ADC RESRDY	ADC Result Ready Trigger
0x26	DAC0 EMPTY	DAC0 Empty Trigger
0x27	DAC1 EMPTY	DAC1 Empty Trigger
0x28 - 0x2B	-	Reserved
0x2C	AES WR	AES Write Trigger
0x2D	AES RD	AES Read Trigger

Bits 6:5 – LVL[1:0]: Channel Arbitration Level
These bits define the arbitration level used for the DMA channel, where a high level has priority over a low level. For further details on arbitration schemes, refer to Arbitration on page 406.

These bits are not enable-protected.

TRIGACT[1:0]	Name	Description
0x0	LVL0	Channel Priority Level 0
0x1	LVL1	Channel Priority Level 1
0x2	LVL2	Channel Priority Level 2
0x3	LVL3	Channel Priority Level 3

Bit 4 – EVOE: Channel Event Output Enable
This bit indicates if the Channel event generation is enabled. The event will be generated for every condition defined in the descriptor Event Output Selection (BTCTRL.EVOSEL).

This bit is available only for the least significant DMA channels. Refer to table: *User Multiplexer Selection* and *Event Generator Selection* of the Event System for details.

Value	Description
0	Channel event generation is disabled.
1	Channel event generation is enabled.

Bit 3 – EVIE: Channel Event Input Enable
This bit is available only for the least significant DMA channels. Refer to table: *User Multiplexer Selection* and *Event Generator Selection* of the Event System for details.

Value	Description
0	Channel event action will not be executed on any incoming event.
1	Channel event action will be executed on any incoming event.

Bits 2:0 – EVACT[2:0]: Event Input Action
These bits define the event input action, as shown below. The action is executed only if the corresponding EVIE bit in CHCTRLB register of the channel is set.

These bits are available only for the least significant DMA channels. Refer to table: *User Multiplexer Selection* and *Event Generator Selection* of the Event System for details.

EVACT[2:0]	Name	Description
0x0	NOACT	No action
0x1	TRIG	Normal Transfer and Conditional Transfer on Strobe trigger
0x2	CTRIG	Conditional transfer trigger
0x3	CBLOCK	Conditional block transfer
0x4	SUSPEND	Channel suspend operation
0x5	RESUME	Channel resume operation
0x6	SSKIP	Skip next block suspend action
0x7	-	Reserved

26.8.20. Channel Interrupt Enable Clear

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Channel Interrupt Enable Set (CHINTENSET) register.
This register affects the DMA channel that is selected in the Channel ID register (CHID.ID).

Name: CHINTENCLR
Offset: 0x4C
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
						SUSP	TCMPL	TERR
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – SUSP: Channel Suspend Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Channel Suspend Interrupt Enable bit, which disables the Channel Suspend interrupt.

Value	Description
0	The Channel Suspend interrupt is disabled.
1	The Channel Suspend interrupt is enabled.

Bit 1 – TCMPL: Channel Transfer Complete Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Channel Transfer Complete Interrupt Enable bit, which disables the Channel Transfer Complete interrupt.

Value	Description
0	The Channel Transfer Complete interrupt is disabled. When block action is set to none, the TCMPL flag will not be set when a block transfer is completed.
1	The Channel Transfer Complete interrupt is enabled.

Bit 0 – TERR: Channel Transfer Error Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Channel Transfer Error Interrupt Enable bit, which disables the Channel Transfer Error interrupt.

Value	Description
0	The Channel Transfer Error interrupt is disabled.
1	The Channel Transfer Error interrupt is enabled.

26.8.21. Channel Interrupt Enable Set

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Channel Interrupt Enable Clear (CHINTENCLR) register.
This register affects the DMA channel that is selected in the Channel ID register (CHID.ID).

Name: CHINTENSET
Offset: 0x4D
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
						SUSP	TCMPL	TERR
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – SUSP: Channel Suspend Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Channel Suspend Interrupt Enable bit, which enables the Channel Suspend interrupt.

Value	Description
0	The Channel Suspend interrupt is disabled.
1	The Channel Suspend interrupt is enabled.

Bit 1 – TCMPL: Channel Transfer Complete Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Channel Transfer Complete Interrupt Enable bit, which enables the Channel Transfer Complete interrupt.

Value	Description
0	The Channel Transfer Complete interrupt is disabled.
1	The Channel Transfer Complete interrupt is enabled.

Bit 0 – TERR: Channel Transfer Error Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Channel Transfer Error Interrupt Enable bit, which enables the Channel Transfer Error interrupt.

Value	Description
0	The Channel Transfer Error interrupt is disabled.
1	The Channel Transfer Error interrupt is enabled.

26.8.22. Channel Interrupt Flag Status and Clear

This register affects the DMA channel that is selected in the Channel ID register (CHID.ID).

Name: CHINTFLAG
Offset: 0x4E
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
						SUSP	TCMPL	TERR
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – SUSP: Channel Suspend
This flag is cleared by writing a '1' to it.

This flag is set when a block transfer with suspend block action is completed, when a software suspend command is executed, when a suspend event is received or when an invalid descriptor is fetched by the DMA.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Channel Suspend interrupt flag for the corresponding channel.

For details on available software commands, refer to CHCTRLB.CMD.

For details on available event input actions, refer to CHCTRLB.EVACT.

For details on available block actions, refer to BTCTRL.BLOCKACT.

Bit 1 – TCMPL: Channel Transfer Complete
This flag is cleared by writing a '1' to it.

This flag is set when a block transfer is completed and the corresponding interrupt block action is enabled.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Transfer Complete interrupt flag for the corresponding channel.

Bit 0 – TERR: Channel Transfer Error
This flag is cleared by writing a '1' to it.

This flag is set when a bus error is detected during a beat transfer or when the DMAC fetches an invalid descriptor.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Transfer Error interrupt flag for the corresponding channel.

26.8.23. Channel Status

This register affects the DMA channel that is selected in the Channel ID register (CHID.ID).

Name: CHSTATUS
Offset: 0x4F
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
						FERR	BUSY	PEND
Access						R	R	R
Reset						0	0	0

Bit 2 – FERR: Channel Fetch Error
This bit is cleared when a software resume command is executed.

This bit is set when an invalid descriptor is fetched.

Bit 1 – BUSY: Channel Busy
This bit is cleared when the channel trigger action is completed, when a bus error is detected or when the channel is disabled.

This bit is set when the DMA channel starts a DMA transfer.

Bit 0 – PEND: Channel Pending
This bit is cleared when the channel trigger action is started, when a bus error is detected or when the channel is disabled. For details on trigger action settings, refer to CHCTRLB.TRIGACT.

This bit is set when a transfer is pending on the DMA channel, as soon as the transfer request is received.

26.9. Register Summary - LP SRAM

Offset	Name	Bit Pos.								
0x00	BTCTRL	7:0				BLOCKACT[1:0]		EVOSEL[1:0]		VALID
0x01		15:8	STEPSIZE[2:0]			STEPSEL	DSTINC	SRCINC	BEATSIZE[1:0]	
0x02	BTCNT	7:0	BTCNT[7:0]							
0x03		15:8	BTCNT[15:8]							
0x04	SRCADDR	7:0	SRCADDR[7:0]							
0x05		15:8	SRCADDR[15:8]							
0x06		23:16	SRCADDR[23:16]							
0x07		31:24	SRCADDR[31:24]							
0x08	DSTADDR	7:0	DSTADDR[7:0]							
0x09		15:8	DSTADDR[15:8]							
0x0A		23:16	DSTADDR[23:16]							
0x0B		31:24	DSTADDR[31:24]							
0x0C	DESCADDR	7:0	DESCADDR[7:0]							
0x0D		15:8	DESCADDR[15:8]							
0x0E		23:16	DESCADDR[23:16]							
0x0F		31:24	DESCADDR[31:24]							

26.10. Register Description - LP SRAM

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection on page 401.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

26.10.1. Block Transfer Control

The BTCTRL register offset is relative to (BASEADDR or WRBADDR) + Channel Number * 0x10

Name: BTCTRL
Offset: 0x00
Reset: -
Property: -

Bit	15	14	13	12	11	10	9	8
	STEPSIZE[2:0]			STEPSEL	DSTINC	SRCINC	BEATSIZE[1:0]	
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				BLOCKACT[1:0]		EVOSEL[1:0]		VALID
Access								
Reset								

Bits 15:13 – STEPSIZE[2:0]: Address Increment Step Size

These bits select the address increment step size. The setting apply to source or destination address, depending on STEPSEL setting.

STEPSIZE[2:0]	Name	Description
0x0	X1	Next ADDR = ADDR + (BEATSIZE+1) * 1
0x1	X2	Next ADDR = ADDR + (BEATSIZE+1) * 2
0x2	X4	Next ADDR = ADDR + (BEATSIZE+1) * 4
0x3	X8	Next ADDR = ADDR + (BEATSIZE+1) * 8
0x4	X16	Next ADDR = ADDR + (BEATSIZE+1) * 16
0x5	X32	Next ADDR = ADDR + (BEATSIZE+1) * 32
0x6	X64	Next ADDR = ADDR + (BEATSIZE+1) * 64
0x7	X128	Next ADDR = ADDR + (BEATSIZE+1) * 128

Bit 12 – STEPSEL: Step Selection

This bit selects if source or destination addresses are using the step size settings.

STEPSEL	Name	Description
0x0	DST	Step size settings apply to the destination address
0x1	SRC	Step size settings apply to the source address

Bit 11 – DSTINC: Destination Address Increment Enable

Writing a '0' to this bit will disable the destination address incrementation. The address will be kept fixed during the data transfer.

Writing a '1' to this bit will enable the destination address incrementation. By default, the destination address is incremented by 1. If the STEPSEL bit is cleared, flexible step-size settings are available in the STEPSIZE register.

Value	Description
0	The Destination Address Increment is disabled.
1	The Destination Address Increment is enabled.

Bit 10 – SRCINC: Source Address Increment Enable
Writing a '0' to this bit will disable the source address incrementation. The address will be kept fixed during the data transfer.

Writing a '1' to this bit will enable the source address incrementation. By default, the source address is incremented by 1. If the STEPSEL bit is set, flexible step-size settings are available in the STEPSIZE register.

Value	Description
0	The Source Address Increment is disabled.
1	The Source Address Increment is enabled.

Bits 9:8 – BEATSIZE[1:0]: Beat Size
These bits define the size of one beat. A beat is the size of one data transfer bus access, and the setting apply to both read and write accesses.

BEATSIZE[1:0]	Name	Description
0x0	BYTE	8-bit bus transfer
0x1	HWORD	16-bit bus transfer
0x2	WORD	32-bit bus transfer
0x3		Reserved

Bits 4:3 – BLOCKACT[1:0]: Block Action
These bits define what actions the DMAC should take after a block transfer has completed.

BLOCKACT[1:0]	Name	Description
0x0	NOACT	Channel will be disabled if it is the last block transfer in the transaction
0x1	INT	Channel will be disabled if it is the last block transfer in the transaction and block interrupt
0x2	SUSPEND	Channel suspend operation is completed
0x3	BOTH	Both channel suspend operation and block interrupt

Bits 2:1 – EVOSEL[1:0]: Event Output Selection
These bits define the event output selection.

EVOSEL[1:0]	Name	Description
0x0	DISABLE	Event generation disabled
0x1	BLOCK	Event strobe when block transfer complete
0x2		Reserved
0x3	BEAT	Event strobe when beat transfer complete

Bit 0 – VALID: Descriptor Valid
Writing a '0' to this bit in the Descriptor or Write-Back memory will suspend the DMA channel operation when fetching the corresponding descriptor.

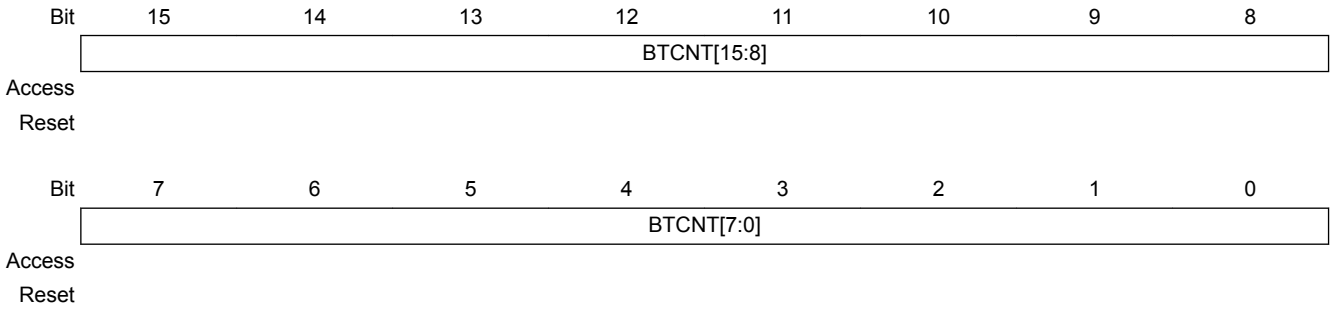
The bit is automatically cleared in the Write-Back memory section when channel is aborted, when an error is detected during the block transfer, or when the block transfer is completed.

Value	Description
0	The descriptor is not valid.
1	The descriptor is valid.

26.10.2. Block Transfer Count

The BTCNT register offset is relative to (BASEADDR or WRBADDR) + Channel Number * 0x10

Name: BTCNT
Offset: 0x02
Reset: -
Property: -



Bits 15:0 – BTCNT[15:0]: Block Transfer Count

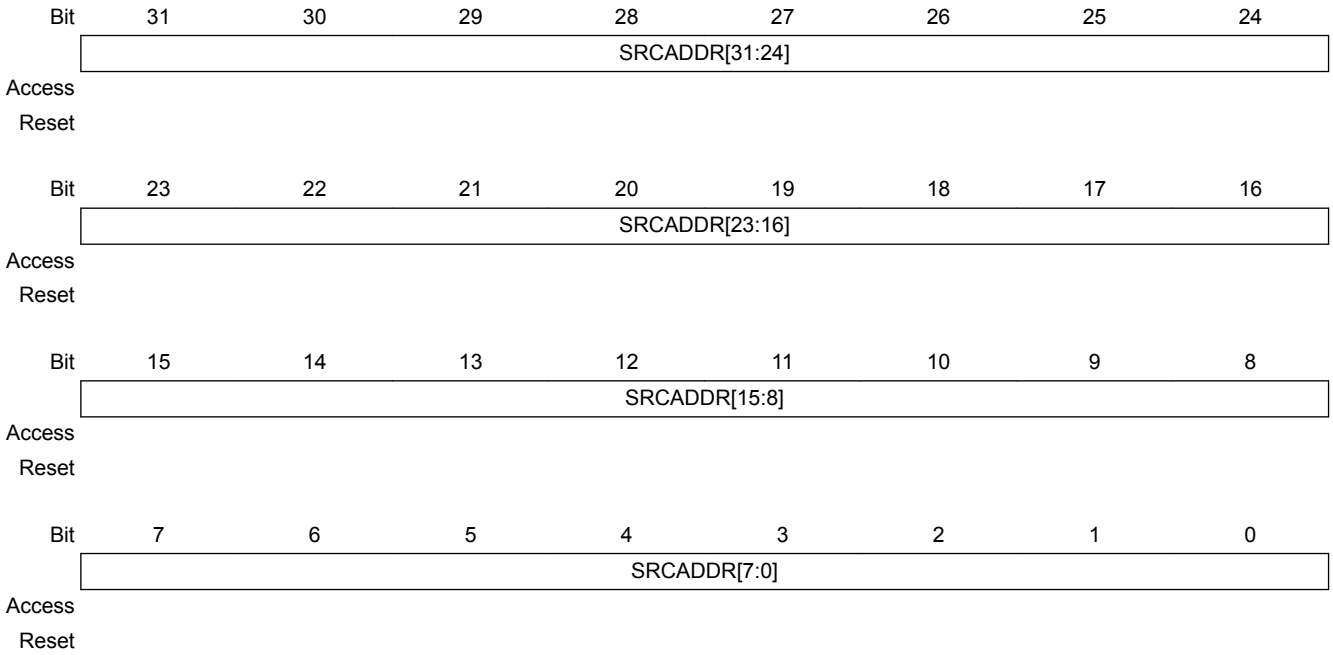
This bit group holds the 16-bit block transfer count.

During a transfer, the internal counter value is decremented by one after each beat transfer. The internal counter is written to the corresponding write-back memory section for the DMA channel when the DMA channel loses priority, is suspended or gets disabled. The DMA channel can be disabled by a complete transfer, a transfer error or by software.

26.10.3. Block Transfer Source Address

The SRCADDR register offset is relative to (BASEADDR or WRBADDR) + Channel Number * 0x10

Name: SRCADDR
Offset: 0x04
Reset: -
Property: -



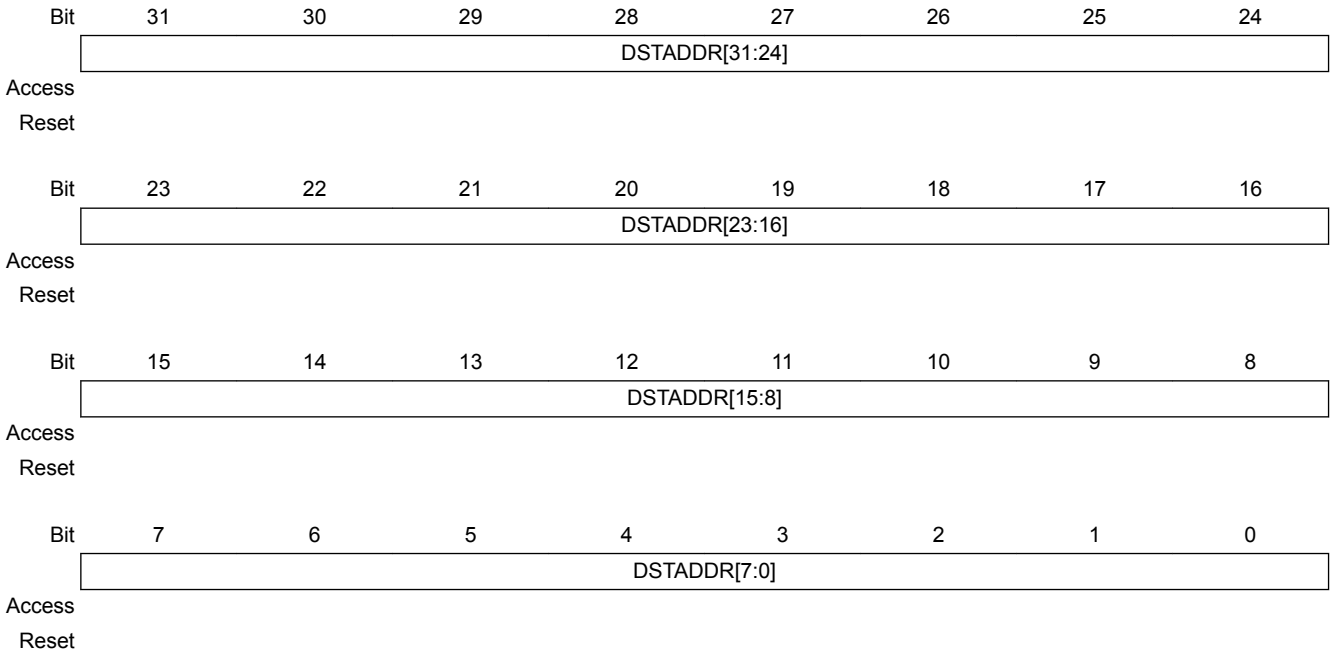
Bits 31:0 – SRCADDR[31:0]: Transfer Source Address

This bit group holds the source address corresponding to the last beat transfer address in the block transfer.

26.10.4. Block Transfer Destination Address

The DSTADDR register offset is relative to (BASEADDR or WRBADDR) + Channel Number * 0x10

Name: DSTADDR
Offset: 0x08
Reset: -
Property: -



Bits 31:0 – DSTADDR[31:0]: Transfer Destination Address

This bit group holds the destination address corresponding to the last beat transfer address in the block transfer.

26.10.5. Next Descriptor Address

The DESCADDR register offset is relative to (BASEADDR or WRBADDR) + Channel Number * 0x10

Name: DESCADDR
Offset: 0x0C
Reset: -
Property: -

Bit	31	30	29	28	27	26	25	24
	DESCADDR[31:24]							
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	DESCADDR[23:16]							
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	DESCADDR[15:8]							
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	DESCADDR[7:0]							
Access								
Reset								

Bits 31:0 – DESCADDR[31:0]: Next Descriptor Address

This bit group holds the LP SRAM address of the next descriptor. The value must be 128-bit aligned. If the value of this LP SRAM register is 0x00000000, the transaction will be terminated when the DMAC tries to load the next transfer descriptor.

27. EIC – External Interrupt Controller

27.1. Overview

The External Interrupt Controller (EIC) allows external pins to be configured as interrupt lines. Each interrupt line can be individually masked and can generate an interrupt on rising, falling, or both edges, or on high or low levels. Each external pin has a configurable filter to remove spikes. Each external pin can also be configured to be asynchronous in order to wake up the device from sleep modes where all clocks have been disabled. External pins can also generate an event.

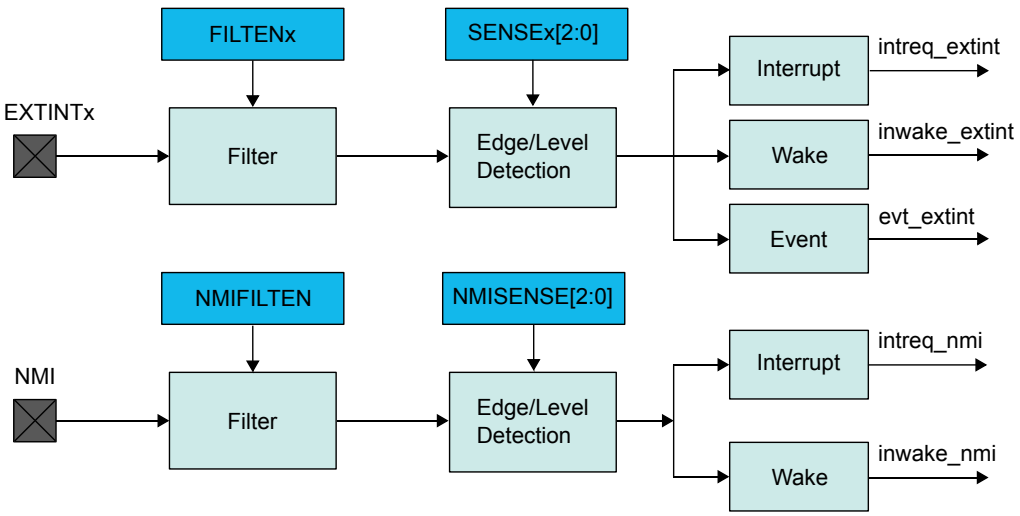
A separate non-maskable interrupt (NMI) is also supported. It has properties similar to the other external interrupts, but is connected to the NMI request of the CPU, enabling it to interrupt any other interrupt mode.

27.2. Features

- Up to 16 external pins, plus one non-maskable pin
- Dedicated, individually maskable interrupt for each pin
- Interrupt on rising, falling, or both edges
- synchronous or asynchronous edge detection mode
- Interrupt on high or low levels
- Asynchronous interrupts for sleep modes without clock
- Filtering of external pins
- Event generation

27.3. Block Diagram

Figure 27-1. EIC Block Diagram



27.4. Signal Description

Signal Name	Type	Description
EXTINT[15..0]	Digital Input	External interrupt pin
NMI	Digital Input	Non-maskable interrupt pin

One signal can be mapped on several pins.

27.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

27.5.1. I/O Lines

Using the EIC's I/O lines requires the I/O pins to be configured.

Related Links

PORT - I/O Pin Controller on page 506

27.5.2. Power Management

All interrupts are available in all sleep modes, but the EIC can be configured to automatically mask some interrupts in order to prevent device wake-up.

The EIC will continue to operate in any sleep mode where the selected source clock is running. The EIC's interrupts can be used to wake up the device from sleep modes. Events connected to the Event System can trigger other operations in the system without exiting sleep modes.

Related Links

PM – Power Manager on page 186

27.5.3. Clocks

The EIC bus clock (CLK_EIC_APB) can be enabled and disabled by the Main Clock Controller, the default state of CLK_EIC_APB can be found in the Peripheral Clock Masking section.

Some optional functions need a peripheral clock, which can either be a generic clock (GCLK_EIC, for wider frequency selection) or a Ultra Low Power 32KHz clock (CLK_ULP32K, for highest power efficiency). One of the clock sources must be configured and enabled before using the peripheral:

GCLK_EIC is configured and enabled in the Generic Clock Controller.

CLK_ULP32K is provided by the internal ultra-low-power (OSCULP32K) oscillator in the OSC32KCTRL module.

Both GCLK_EIC and CLK_ULP32K are asynchronous to the user interface clock (CLK_EIC_APB). Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains. Refer to Synchronization for further details.

Related Links

MCLK – Main Clock on page 148

Peripheral Clock Masking on page 151

GCLK - Generic Clock Controller on page 131

OSC32KCTRL – 32KHz Oscillators Controller on page 264

27.5.4. DMA

Not applicable.

27.5.5. Interrupts

There are two interrupt request lines, one for the external interrupts (EXTINT) and one for non-maskable interrupt (NMI).

The EXTINT interrupt request line is connected to the interrupt controller. Using the EIC interrupt requires the interrupt controller to be configured first.

The NMI interrupt request line is also connected to the interrupt controller, but does not require the interrupt to be configured.

Related Links

Nested Vector Interrupt Controller on page 51

27.5.6. Events

The events are connected to the Event System. Using the events requires the Event System to be configured first.

Related Links

EVSYS – Event System on page 536

27.5.7. Debug Operation

When the CPU is halted in debug mode, the EIC continues normal operation. If the EIC is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

27.5.8. Register Access Protection

All registers with write-access can be write-protected optionally by the Peripheral Access Controller (PAC), except the following registers:

- Interrupt Flag Status and Clear register (INTFLAG)
- Non-Maskable Interrupt Flag Status and Clear register (NMIFLAG)

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

PAC write-protection does not apply to accesses through an external debugger.

Related Links

PAC - Peripheral Access Controller on page 58

27.5.9. Analog Connections

Not applicable.

27.6. Functional Description

27.6.1. Principle of Operation

The EIC detects edge or level condition to generate interrupts to the CPU interrupt controller or events to the Event System. Each external interrupt pin (EXTINT) can be filtered using majority vote filtering, clocked by GCLK_EIC or by CLK_ULP32K.

27.6.2. Basic Operation

27.6.2.1. Initialization

The EIC must be initialized in the following order:

1. Enable CLK_EIC_APB
2. If required, configure the NMI by writing the Non-Maskable Interrupt Control register (NMICTRL)
3. When the NMI is used or synchronous edge detection or filtering are required, enable GCLK_EIC or CLK_ULP32K.
   GCLK_EIC is used when a frequency higher than 32KHz is required for filtering, CLK_ULP32K is recommended when power consumption is the priority. For CLK_ULP32K write a '1' to the Clock Selection bit in the Control A register (CTRLA.CKSEL).
4. Configure the EIC input sense and filtering by writing the Configuration n register (CONFIGn on page 482).
5. Enable the EIC.

The following bits are enable-protected, meaning that it can only be written when the EIC is disabled (CTRLA.ENABLE=0):

- Clock Selection bit in Control A register (CTRLA.CKSEL)

The following registers are enable-protected:

- Event Control register (EVCTRL)
- Configuration n register (CONFIG0, CONFIG1...)

Enable-protected bits in the CTRLA register can be written at the same time when setting CTRLA.ENABLE to '1', but not at the same time as CTRLA.ENABLE is being cleared.

Enable-protection is denoted by the "Enable-Protected" property in the register description.

27.6.2.2. Enabling, Disabling, and Resetting

The EIC is enabled by writing a '1' the Enable bit in the Control A register (CTRLA.ENABLE). The EIC is disabled by writing CTRLA.ENABLE to '0'.

The EIC is reset by setting the Software Reset bit in the Control register (CTRLA.SWRST). All registers in the EIC will be reset to their initial state, and the EIC will be disabled.

Refer to the CTRLA register description for details.

27.6.3. External Pin Processing

Each external pin can be configured to generate an interrupt/event on edge detection (rising, falling or both edges) or level detection (high or low). The sense of external interrupt pins is configured by writing the Input Sense x bits in the Config n register (CONFIGn.SENSEx). The corresponding interrupt flag (INTFLAG.EXTINT[x]) in the Interrupt Flag Status and Clear register (INTFLAG) is set when the interrupt condition is met.

When the interrupt flag has been cleared in edge-sensitive mode, INTFLAG.EXTINT[x] will only be set if a new interrupt condition is met. In level-sensitive mode, when interrupt has been cleared, INTFLAG.EXTINT[x] will be set immediately if the EXTINTx pin still matches the interrupt condition.

Each external pin can be filtered by a majority vote filtering, clocked by GCLK_EIC or CLK_ULP32K. Filtering is enabled if bit Filter Enable x in the Configuration n register (CONFIGn.FILTENx) is written to '1'. The majority vote filter samples the external pin three times with GCLK_EIC or CLK_ULP32K and outputs the value when two or more samples are equal.

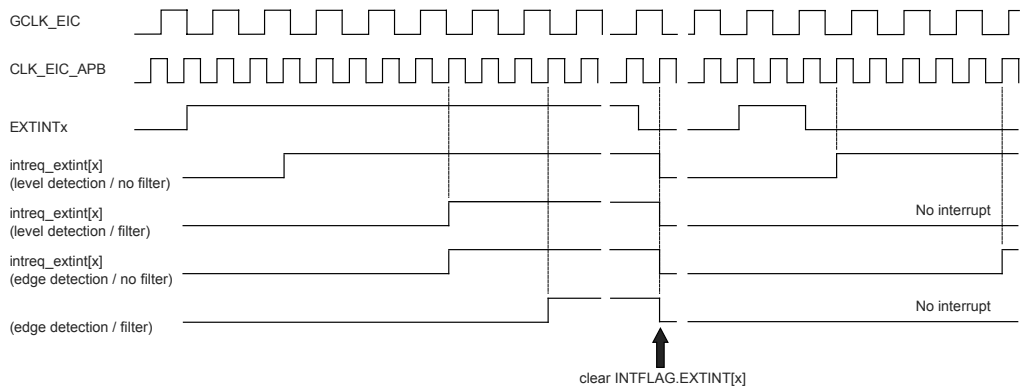
Table 27-1. Majority Vote Filter

Samples [0, 1, 2]	Filter Output
[0,0,0]	0
[0,0,1]	0
[0,1,0]	0
[0,1,1]	1
[1,0,0]	0
[1,0,1]	1
[1,1,0]	1
[1,1,1]	1

When an external interrupt is configured for level detection and when filtering is disabled, detection is done asynchronously. Asynchronuous detection does not require GCLK_EIC or CLK_ULP32K, but interrupt and events can still be generated.

If filtering or edge detection is enabled, the EIC automatically requests GCLK_EIC or CLK_ULP32K to operate. The selection between these two clocks is done by writing the Clock Selection bits in the Control A register (CTRLA.CKSEL). GCLK_EIC must be enabled in the GCLK module.

Figure 27-2. Interrupt Detections



The detection delay depends on the detection mode.

Table 27-2. Interrupt Latency

Detection mode	Latency (worst case)
Level without filter	Five CLK_EIC_APB periods
Level with filter	Four GCLK_EIC/CLK_ULP32K periods + five CLK_EIC_APB periods
Edge without filter	Four GCLK_EIC/CLK_ULP32K periods + five CLK_EIC_APB periods
Edge with filter	Six GCLK_EIC/CLK_ULP32K periods + five CLK_EIC_APB periods

Related Links

GCLK - Generic Clock Controller on page 131

27.6.4. Additional Features

27.6.4.1. Non-Maskable Interrupt (NMI)

The non-maskable interrupt pin can also generate an interrupt on edge or level detection, but it is configured with the dedicated NMI Control register (NMICTRL). To select the sense for NMI, write to the NMISENSE bit group in the NMI Control register (NMICTRL.NMISENSE). NMI filtering is enabled by writing a '1' to the NMI Filter Enable bit (NMICTRL.NMIFILTEN).

If edge detection or filtering is required, enable GCLK_EIC or CLK_ULP32K.

NMI detection is enabled only by the NMICTRL.NMISENSE value, and the EIC is not required to be enabled.

After reset, NMI is configured to no detection mode.

When an NMI is detected, the non-maskable interrupt flag in the NMI Flag Status and Clear register is set (NMIFLAG.NMI). NMI interrupt generation is always enabled, and NMIFLAG.NMI generates an interrupt request when set.

27.6.4.2. Asynchronous Edge Detection Mode

The EXTINT edge detection can be operated synchronously or asynchronously, selected by the Asynchronous Control Mode bit for external pin x in the External Interrupt Asynchronous Mode register (ASYNCH.ASYNCH[x]). The EIC edge detection is operated synchronously when the Asynchronous Control Mode bit (ASYNCH.ASYNCH[x]) is '0' (default value). It is operated asynchronously when ASYNCH.ASYNCH[x] is written to '1'.

In *Synchronous Edge Detection Mode*, the external interrupt (EXTINT) or the non-maskable interrupt (NMI) pins are sampled using the EIC clock as defined by the Clock Selection bit in the Control A register (CTRLA.CKSEL). The External Interrupt flag (INTFLAG.EXTINT[x]) or Non-Maskable Interrupt flag (NMIFLAG.NMI) is set when the last sampled state of the pin differs from the previously sampled state. In this mode, the EIC clock is required.

The Synchronous Edge Detection Mode can be used in Idle sleep mode.

In *Asynchronous Edge Detection Mode*, the external interrupt (EXTINT) pins or the non-maskable interrupt (NMI) pins set the External Interrupt flag or Non-Maskable Interrupt flag (INTFLAG.EXTINT[x] or NMIFLAG) directly. In this mode, the EIC clock is not requested.

The asynchronous edge detection mode can be used in all sleep modes.

27.6.5. DMA Operation

Not applicable.

27.6.6. Interrupts

The EIC has the following interrupt sources:

- External interrupt pins (EXTINTx). See Basic Operation on page 467.
- Non-maskable interrupt pin (NMI). See Additional Features on page 469.

Each interrupt source has an associated interrupt flag. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) is set when an interrupt condition occurs (NMIFLAG for NMI). Each interrupt, except NMI, can be individually enabled by setting the corresponding bit in the Interrupt Enable Set register (INTENSET=1), and disabled by setting the corresponding bit in the Interrupt Enable Clear register (INTENCLR=1).

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the EIC

is reset. See the INTFLAG register for details on how to clear interrupt flags. The EIC has one common interrupt request line for all the interrupt sources, and one interrupt request line for the NMI. The user must read the INTFLAG (or NMIFLAG) register to determine which interrupt condition is present.

Note: Interrupts must be globally enabled for interrupt requests to be generated.

Related Links

Processor and Architecture on page 49

27.6.7. Events

The EIC can generate the following output events:

- External event from pin (EXTINTx).

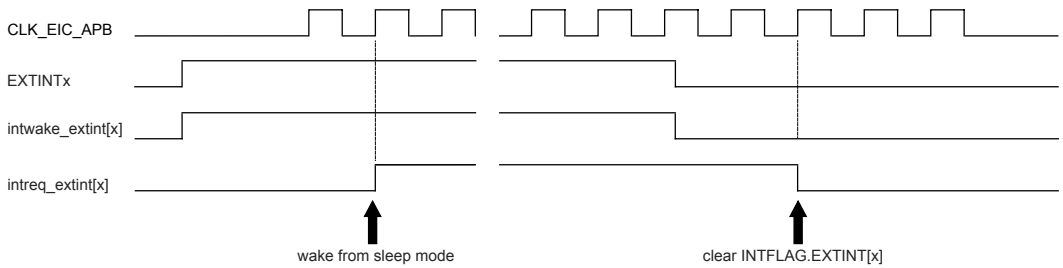
Setting an Event Output Control register (EVCTRL.EXTINTEO) enables the corresponding output event. Clearing this bit disables the corresponding output event. Refer to *Event System* for details on configuring the Event System.

When the condition on pin EXTINTx matches the configuration in the CONFIGn register, the corresponding event is generated, if enabled.

27.6.8. Sleep Mode Operation

In sleep modes, an EXTINTx pin can wake up the device if the corresponding condition matches the configuration in CONFIGn on page 482 register, and the corresponding bit in the Interrupt Enable Set register (INTENSET) is written to '1'.

Figure 27-3. Wake-up Operation Example (High-Level Detection, No Filter, Interrupt Enable Set)



27.6.9. Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read. An exception is the Channel Enable bit in the Peripheral Channel Control registers (PCHCTRLm.CHEN). When changing this bit, the bit value must be read-back to ensure the synchronization is complete and to assert glitch free internal operation. Note that changing the bit value under ongoing synchronization will *not* generate an error.

The following bits are synchronized when written:

- Software Reset bit in control register (CTRLA.SWRST)
- Enable bit in control register (CTRLA.ENABLE)

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

27.7. Register Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0				CKSEL			ENABLE	SWRST
0x01	NMICTRL	7:0				ASYNCH	NMIFILTEN	NMISENSE[2:0]		
0x02	NMIFLAG	7:0								NMI
0x03		15:8								
0x04	SYNCBUSY	7:0							ENABLE	SWRST
0x05		15:8								
0x06		23:16								
0x07		31:24								
0x08	EVCTRL	7:0	EXTINTEO[7:0]							
0x09		15:8	EXTINTEO[15:8]							
0x0A		23:16								
0x0B		31:24								
0x0C	INTENCLR	7:0	EXTINT[7:0]							
0x0D		15:8	EXTINT[15:8]							
0x0E		23:16								
0x0F		31:24								
0x10	INTENSET	7:0	EXTINT[7:0]							
0x11		15:8	EXTINT[15:8]							
0x12		23:16								
0x13		31:24								
0x14	INTFLAG	7:0	EXTINT[7:0]							
0x15		15:8	EXTINT[15:8]							
0x16		23:16								
0x17		31:24								
0x18	ASYNCH	7:0	ASYNCH[7:0]							
0x19		15:8	ASYNCH[15:8]							
0x1A		23:16								
0x1B		31:24								
0x1C	CONFIG0	7:0	FILTEN1	SENSE1[2:0]		FILTEN0	SENSE0[2:0]			
0x1D		15:8	FILTEN3	SENSE3[2:0]		FILTEN2	SENSE2[2:0]			
0x1E		23:16	FILTEN5	SENSE5[2:0]		FILTEN4	SENSE4[2:0]			
0x1F		31:24	FILTEN7	SENSE7[2:0]		FILTEN6	SENSE6[2:0]			
0x20	CONFIG1	7:0	FILTEN9	SENSE9[2:0]		FILTEN8	SENSE8[2:0]			
0x21		15:8	FILTEN11	SENSE11[2:0]		FILTEN10	SENSE10[2:0]			
0x22		23:16	FILTEN13	SENSE13[2:0]		FILTEN12	SENSE12[2:0]			
0x23		31:24	FILTEN15	SENSE15[2:0]		FILTEN14	SENSE14[2:0]			

27.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

27.8.1. Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized, Enable-Protected

Bit	7	6	5	4	3	2	1	0
				CKSEL			ENABLE	SWRST
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 4 – CKSEL: Clock Selection
The EIC can be clocked either by GCLK_EIC (when a frequency higher than 32KHz is required for filtering) or by CLK_ULP32K (when power consumption is the priority).

This bit is not Write-Synchronized.

Value	Description
0	The EIC is clocked by GCLK_EIC.
1	The EIC is clocked by CLK_ULP32K.

Bit 1 – ENABLE: Enable
Due to synchronization there is a delay between writing to CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the Enable bit in the Synchronization Busy register will be set (SYNCBUSY.ENABLE=1). SYNCBUSY.ENABLE will be cleared when the operation is complete.

This bit is not Enable-Protected.

Value	Description
0	The EIC is disabled.
1	The EIC is enabled.

Bit 0 – SWRST: Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the EIC to their initial state, and the EIC will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write operation will be discarded.

Due to synchronization there is a delay from writing CTRLA.SWRST until the Reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the Reset is complete.

This bit is not Enable-Protected.

Value	Description
0	There is no ongoing reset operation.
1	The reset operation is ongoing.

27.8.2. Non-Maskable Interrupt Control

Name: NMICTRL
Offset: 0x01
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
				ASYNCH	NMIFILTEN	NMISENSE[2:0]		
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit 4 – ASYNCH: Asynchronous Edge Detection Mode
The NMI edge detection can be operated synchronously or asynchronously to the EIC clock.

Value	Description
0	The NMI edge detection is synchronously operated.
1	The NMI edge detection is asynchronously operated.

Bit 3 – NMIFILTEN: Non-Maskable Interrupt Filter Enable

Value	Description
0	NMI filter is disabled.
1	NMI filter is enabled.

Bits 2:0 – NMISENSE[2:0]: Non-Maskable Interrupt Sense
These bits define on which edge or level the NMI triggers.

Value	Name	Description
0x0	NONE	No detection
0x1	RISE	Rising-edge detection
0x2	FALL	Falling-edge detection
0x3	BOTH	Both-edge detection
0x4	HIGH	High-level detection
0x5	LOW	Low-level detection
0x6 - 0x7	-	Reserved

27.8.3. Non-Maskable Interrupt Flag Status and Clear

Name: NMIFLAG
Offset: 0x02
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								NMI
Access								R/W
Reset								0

Bit 0 – NMI: Non-Maskable Interrupt
This flag is cleared by writing a '1' to it.

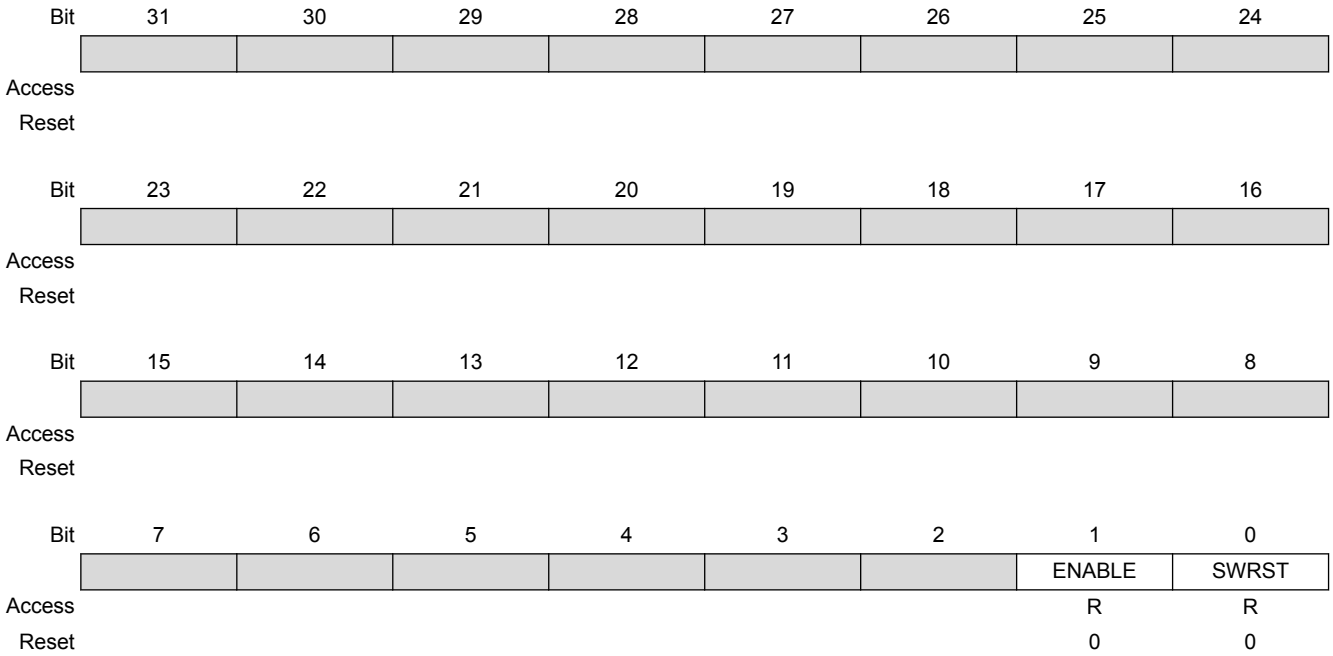
This flag is set when the NMI pin matches the NMI sense configuration, and will generate an interrupt request.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the non-maskable interrupt flag.

27.8.4. Synchronization Busy

Name: SYNCBUSY
Offset: 0x04
Reset: 0x00000000
Property: –



Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							ENABLE	SWRST
Access							R	R
Reset							0	0

Bit 1 – ENABLE: Enable Synchronization Busy Status

Value	Description
0	Write synchronization for CTRLA.ENABLE bit is complete.
1	Write synchronization for CTRLA.ENABLE bit is ongoing.

Bit 0 – SWRST: Software Reset Synchronization Busy Status

Value	Description
0	Write synchronization for CTRLA.SWRST bit is complete.
1	Write synchronization for CTRLA.SWRST bit is ongoing.

27.8.5. Event Control

Name: EVCTRL
Offset: 0x08
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	EXTINTEO[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	EXTINTEO[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – EXTINTEO[15:0]: External Interrupt x Event Output
These bits enable the event associated with the EXTINTx pin.

Value	Description
0	Event from pin EXTINTx is disabled.
1	Event from pin EXTINTx is enabled and will be generated when EXTINTx pin matches the external interrupt sensing configuration.

27.8.6. Interrupt Enable Clear

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Name: INTENCLR
Offset: 0x0C
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	EXTINT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	EXTINT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – EXTINT[15:0]: External Interrupt x Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the External Interrupt x Enable bit, which disables the external interrupt.

Value	Description
0	The external interrupt x is disabled.
1	The external interrupt x is enabled.

27.8.7. Interrupt Enable Set

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Name: INTENSET
Offset: 0x10
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	EXTINT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	EXTINT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – EXTINT[15:0]: External Interrupt x Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the External Interrupt x Enable bit, which enables the external interrupt.

Value	Description
0	The external interrupt x is disabled.
1	The external interrupt x is enabled.

27.8.8. Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x14
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	EXTINT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	EXTINT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – EXTINT[15:0]: External Interrupt x
This flag is cleared by writing a '1' to it.

This flag is set when EXTINTx pin matches the external interrupt sense configuration and will generate an interrupt request if INTENCLR/SET.EXTINT[x] is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the External Interrupt x flag.

27.8.9. External Interrupt Asynchronous Mode

Name: ASYNCH
Offset: 0x18
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	ASYNCH[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ASYNCH[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – ASYNCH[15:0]: Asynchronous Edge Detection Mode

Value	Description
0	The EXTINT edge detection is synchronously operated.
1	The EXTINT edge detection is asynchronously operated.

27.8.10. Configuration n

Name: CONFIGn
Offset: 0x1C+n*0x4 [n = 0..1]
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
	FILTEN7	SENSE7[2:0]			FILTEN6	SENSE6[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	FILTEN5	SENSE5[2:0]			FILTEN4	SENSE4[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	FILTEN3	SENSE3[2:0]			FILTEN2	SENSE2[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FILTEN1	SENSE1[2:0]			FILTEN0	SENSE0[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 3,7,11,15,19,23,27,31 – FILTENx: Filter x Enable [x = 7..0]

Value	Description
0	Filter is disabled for EXTINT[n*8+1] input.
1	Filter is enabled for EXTINT[n*8+1] input.

Bits 0:2,4:6,8:10,12:14,16:18,20:22,24:26,28:30 – SENSEx: Input Sense x Configuration
These bits define on which edge or level the interrupt or event for EXTINT[n*8+x] will be generated.

Value	Name	Description
0x0	NONE	No detection
0x1	RISE	Rising-edge detection
0x2	FALL	Falling-edge detection
0x3	BOTH	Both-edge detection
0x4	HIGH	High-level detection
0x5	LOW	Low-level detection
0x6 - 0x7	-	Reserved

28. NVMCTRL – Non-Volatile Memory Controller

28.1. Overview

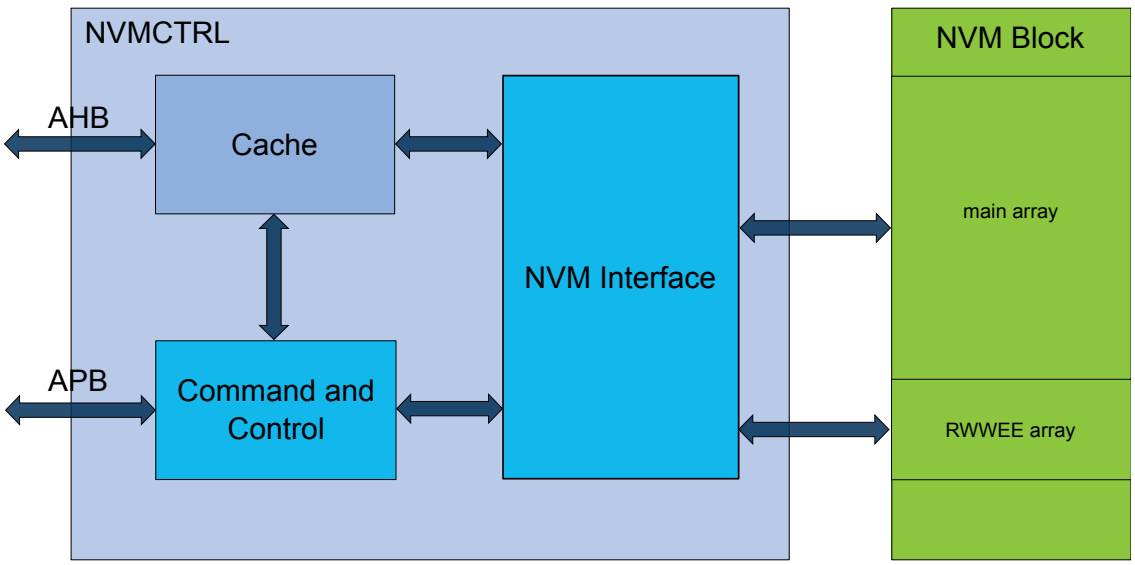
Non-Volatile Memory (NVM) is a reprogrammable Flash memory that retains program and data storage even with power off. It embeds a main array and a separate smaller array intended for EEPROM emulation (RWWEE) that can be programmed while reading the main array. The NVM Controller (NVMCTRL) connects to the AHB and APB bus interfaces for system access to the NVM block. The AHB interface is used for reads and writes to the NVM block, while the APB interface is used for commands and configuration.

28.2. Features

- 32-bit AHB interface for reads and writes
- Read While Write EEPROM emulation area
- All NVM sections are memory mapped to the AHB, including calibration and system configuration
- 32-bit APB interface for commands and control
- Programmable wait states for read optimization
- 16 regions can be individually protected or unprotected
- Additional protection for boot loader
- Supports device protection through a security bit
- Interface to Power Manager for power-down of Flash blocks in sleep modes
- Can optionally wake up on exit from sleep or on first access
- Direct-mapped cache

28.3. Block Diagram

Figure 28-1. Block Diagram



28.4. Signal Description

Not applicable.

28.5. Product Dependencies

In order to use this module, other parts of the system must be configured correctly, as described below.

28.5.1. Power Management

The NVMCTRL will continue to operate in any sleep mode where the selected source clock is running. The NVMCTRL interrupts can be used to wake up the device from sleep modes.

The Power Manager will automatically put the NVM block into a low-power state when entering sleep mode. This is based on the Control B register (CTRLB) SLEEPPRM bit setting. Refer to the CTRLB.SLEEPPRM register description for more details.

Related Links

PM – Power Manager on page 186

28.5.2. Clocks

Two synchronous clocks are used by the NVMCTRL. One is provided by the AHB bus (CLK_NVMCTRL_AHB) and the other is provided by the APB bus (CLK_NVMCTRL_APB). For higher system frequencies, a programmable number of wait states can be used to optimize performance. When changing the AHB bus frequency, the user must ensure that the NVM Controller is configured with the proper number of wait states. Refer to the Electrical Characteristics for the exact number of wait states to be used for a particular frequency range.

Related Links

Electrical Characteristics on page 1140

28.5.3. Interrupts

The NVM Controller interrupt request line is connected to the interrupt controller. Using the NVMCTRL interrupt requires the interrupt controller to be programmed first.

28.5.4. Debug Operation

When an external debugger forces the CPU into debug mode, the peripheral continues normal operation.

Access to the NVM block can be protected by the security bit. In this case, the NVM block will not be accessible. See the section on the NVMCTRL Security Bit on page 490 for details.

28.5.5. Register Access Protection

All registers with write-access are optionally write-protected by the Peripheral Access Controller (PAC), except the following registers:

- Interrupt Flag Status and Clear register (INTFLAG)
- Status register (STATUS)

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

PAC Write-Protection does not apply to accesses through an external debugger. Refer to the PAC - Peripheral Access Controller section for details.

Related Links

PAC - Peripheral Access Controller on page 58

28.5.6. Analog Connections

Not applicable.

28.6. Functional Description

28.6.1. Principle of Operation

The NVM Controller is a slave on the AHB and APB buses. It responds to commands, read requests and write requests, based on user configuration.

28.6.1.1. Initialization

After power up, the NVM Controller goes through a power-up sequence. During this time, access to the NVM Controller from the AHB bus is halted. Upon power-up completion, the NVM Controller is operational without any need for user configuration.

28.6.2. Memory Organization

Refer to the Physical Memory Map for memory sizes and addresses for each device.

The NVM is organized into rows, where each row contains four pages, as shown in the NVM Row Organization figure. The NVM has a row-erase granularity, while the write granularity is by page. In other words, a single row erase will erase all four pages in the row, while four write operations are used to write the complete row.

Figure 28-2. NVM Row Organization

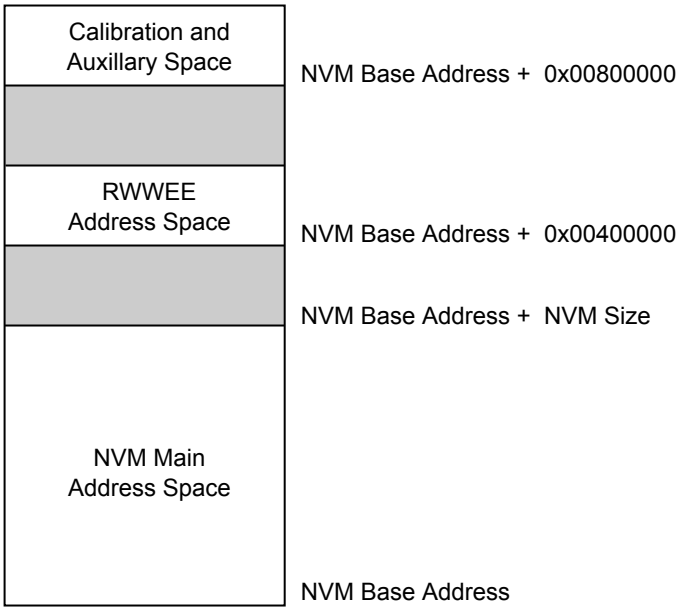
Row n	Page (n*4) + 3	Page (n*4) + 2	Page (n*4) + 1	Page (n*4) + 0

The NVM block contains a calibration and auxiliary space plus a dedicated EEPROM emulation space that are memory mapped. Refer to the NVM Organization figure below for details.

The calibration and auxiliary space contains factory calibration and system configuration information. These spaces can be read from the AHB bus in the same way as the main NVM main address space.

In addition, a boot loader section can be allocated at the beginning of the main array, and an EEPROM section can be allocated at the end of the NVM main address space.

Figure 28-3. NVM Memory Organization

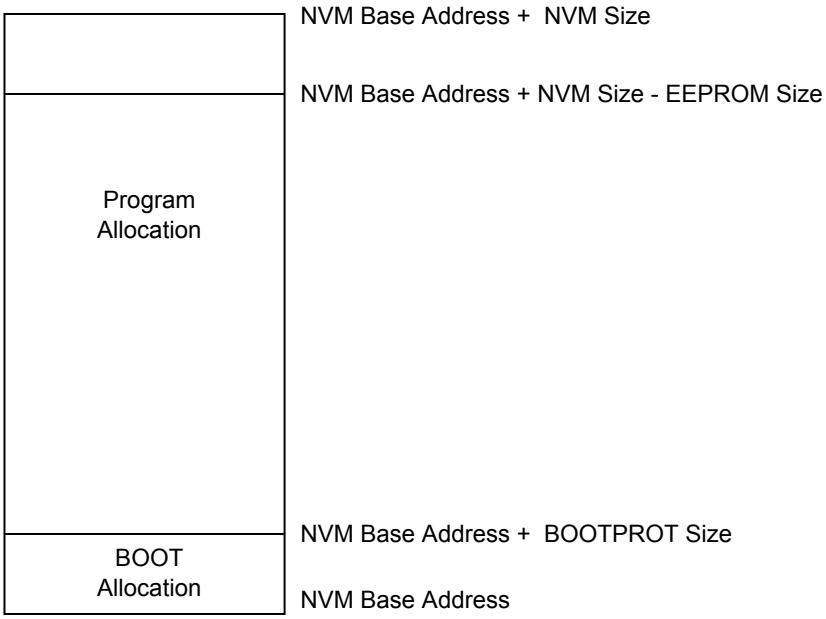


The lower rows in the NVM main address space can be allocated as a boot loader section by using the BOOTPROT fuses, and the upper rows can be allocated to EEPROM, as shown in the figure below.

The boot loader section is protected by the lock bit(s) corresponding to this address space and by the BOOTPROT[2:0] fuse. The EEPROM rows can be written regardless of the region lock status.

The number of rows protected by BOOTPROT is given in Boot Loader Size, the number of rows allocated to the EEPROM are given in EEPROM Size.

Figure 28-4. EEPROM and Boot Loader Allocation



Related Links

Physical Memory Map on page 45

28.6.3. Region Lock Bits

The NVM block is grouped into 16 equally sized regions. The region size is dependent on the Flash memory size, and is given in the table below. Each region has a dedicated lock bit preventing writing and erasing pages in the region. After production, all regions will be unlocked.

Table 28-1. Region Size

Memory Size [KB]	Region Size [KB]
256	16
128	8
64	4
32	2

To lock or unlock a region, the Lock Region and Unlock Region commands are provided. Writing one of these commands will temporarily lock/unlock the region containing the address loaded in the ADDR register. ADDR can be written by software, or the automatically loaded value from a write operation can be used. The new setting will stay in effect until the next Reset, or until the setting is changed again using the Lock and Unlock commands. The current status of the lock can be determined by reading the LOCK register.

To change the default lock/unlock setting for a region, the user configuration section of the auxiliary space must be written using the Write Auxiliary Page command. Writing to the auxiliary space will take effect after the next Reset. Therefore, a boot of the device is needed for changes in the lock/unlock setting to take effect. Refer to the Physical Memory Map for calibration and auxiliary space address mapping.

Related Links

Physical Memory Map on page 45

28.6.4. Command and Data Interface

The NVM Controller is addressable from the APB bus, while the NVM main address space is addressable from the AHB bus. Read and automatic page write operations are performed by addressing the NVM main address space or the RWWEE address space directly, while other operations such as manual page writes and row erases must be performed by issuing commands through the NVM Controller.

To issue a command, the CTRLA.CMD bits must be written along with the CTRLA.CMDEX value. When a command is issued, INTFLAG.READY will be cleared until the command has completed. Any commands written while INTFLAG.READY is low will be ignored.

Read the CTRLA register description for more details.

The CTRLB register must be used to control the power reduction mode, read wait states, and the write mode.

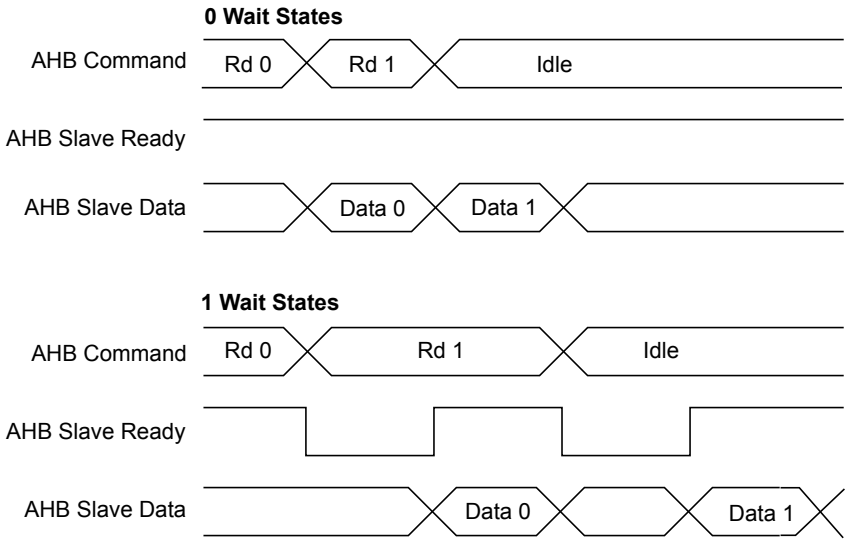
28.6.4.1. NVM Read

Reading from the NVM main address space is performed via the AHB bus by addressing the NVM main address space or auxiliary address space directly. Read data is available after the configured number of read wait states (CTRLB.RWS) set in the NVM Controller.

The number of cycles data are delayed to the AHB bus is determined by the read wait states. Examples of using zero and one wait states are shown in Figure Read Wait State Examples below.

Reading the NVM main address space while a programming or erase operation is ongoing on the NVM main array results in an AHB bus stall until the end of the operation. Reading the NVM main array does not stall the bus when the RWWEE array is being programmed or erased.

Figure 28-5. Read Wait State Examples



28.6.4.2. RWWEE Read

Reading from the RWW EEPROM address space is performed via the AHB bus by addressing the RWWEE address space directly. Refer to the figures in Memory Organization on page 485 for details.

Read timings are similar to regular NVM read timings when access size is Byte or half-Word. The AHB data phase is twice as long in case of full-Word-size access.

It is not possible to read the RWWEE area while the NVM main array is being written or erased, whereas the RWWEE area can be written or erased while the main array is being read.

The RWWEE address space is not cached, therefore it is recommended to limit access to this area for performance and power consumption considerations.

28.6.4.3. NVM Write

The NVM Controller requires that an erase must be done before programming. The entire NVM main address space and the RWWEE address space can be erased by a debugger Chip Erase command. Alternatively, rows can be individually erased by the Erase Row command or the RWWEE Erase Row command to erase the NVM main address space or the RWWEE address space, respectively.

After programming the NVM main array, the region that the page resides in can be locked to prevent spurious write or erase sequences. Locking is performed on a per-region basis, and so, locking a region will lock all pages inside the region.

Data to be written to the NVM block are first written to and stored in an internal buffer called the *page buffer*. The page buffer contains the same number of bytes as an NVM page. Writes to the page buffer must be 16 or 32 bits. 8-bit writes to the page buffer are not allowed and will cause a system exception.

Both the NVM main array and the RWWEE array share the same page buffer. Writing to the NVM block via the AHB bus is performed by a load operation to the page buffer. For each AHB bus write, the address is stored in the ADDR register. After the page buffer has been loaded with the required number of bytes, the page can be written to the NVM main array or the RWWEE array by setting CTRLA.CMD to 'Write Page' or 'RWWEE Write Page', respectively, and setting the key value to CMDEX. The LOAD bit in the STATUS register indicates whether the page buffer has been loaded or not. Before writing the page to memory, the accessed row must be erased.

Automatic page writes are enabled by writing the manual write bit to zero (CTRLB.MANW=0). This will trigger a write operation to the page addressed by ADDR when the last location of the page is written.

Because the address is automatically stored in ADDR during the I/O bus write operation, the last given address will be present in the ADDR register. There is no need to load the ADDR register manually, unless a different page in memory is to be written.

Procedure for Manual Page Writes (CTRLB.MANW=1)
The row to be written to must be erased before the write command is given.

- Write to the page buffer by addressing the NVM main address space directly
- Write the page buffer to memory: CTRL.CMD='Write Page' and CMDEX
- The READY bit in the INTFLAG register will be low while programming is in progress, and access through the AHB will be stalled

Procedure for Automatic Page Writes (CTRLB.MANW=0)
The row to be written to must be erased before the last write to the page buffer is performed.

Note that partially written pages must be written with a manual write.

- Write to the page buffer by addressing the NVM main address space directly.
  When the last location in the page buffer is written, the page is automatically written to NVM main address space.
- INTFLAG.READY will be zero while programming is in progress and access through the AHB will be stalled.

28.6.4.4. Page Buffer Clear

The page buffer is automatically set to all '1' after a page write is performed. If a partial page has been written and it is desired to clear the contents of the page buffer, the Page Buffer Clear command can be used.

28.6.4.5. Erase Row

Before a page can be written, the row containing that page must be erased. The Erase Row command can be used to erase the desired row in the NVM main address space. The RWWEE Erase Row can be used to erase the desired row in the RWWEE array. Erasing the row sets all bits to '1'. If the row resides in a region that is locked, the erase will not be performed and the Lock Error bit in the Status register (STATUS.LOCKE) will be set.

Procedure for Erase Row

- Write the address of the row to erase to ADDR. Any address within the row can be used.
- Issue an Erase Row command.

28.6.4.6. Lock and Unlock Region

These commands are used to lock and unlock regions as detailed in section Region Lock Bits on page 487.

28.6.4.7. Set and Clear Power Reduction Mode

The NVM Controller and block can be taken in and out of power reduction mode through the Set and Clear Power Reduction Mode commands. When the NVM Controller and block are in power reduction mode, the Power Reduction Mode bit in the Status register (STATUS.PRM) is set.

28.6.5. NVM User Configuration

The NVM user configuration resides in the auxiliary space. Refer to the Physical Memory Map of the device for calibration and auxiliary space address mapping.

The bootloader resides in the main array starting at offset zero. The allocated boot loader section is write-protected.

Table 28-2. Boot Loader Size

BOOTPROT [2:0]	Rows Protected by BOOTPROT	Boot Loader Size in Bytes
7	None	0
6	2	512
5	4	1024
4	8	2048
3	16	4096
2	32	8192
1	64	16384
0	128	32768

The EEPROM[2:0] bits indicate the EEPROM size, see the table below. The EEPROM resides in the upper rows of the NVM main address space and is writable, regardless of the region lock status.

Table 28-3. EEPROM Size

EEPROM[2:0]	Rows Allocated to EEPROM	EEPROM Size in Bytes
7	None	0
6	1	256
5	2	512
4	4	1024
3	8	2048
2	16	4096
1	32	8192
0	64	16384

Related Links

Physical Memory Map on page 45

28.6.6. Security Bit

The security bit allows the entire chip to be locked from external access for code security. The security bit can be written by a dedicated command, Set Security Bit (SSB). Once set, the only way to clear the security bit is through a debugger Chip Erase command. After issuing the SSB command, the PROGE error bit can be checked.

In order to increase the security level it is recommended to enable the internal BOD33 when the security bit is set.

Related Links

DSU - Device Service Unit on page 87

28.6.7. Cache

The NVM Controller cache reduces the device power consumption and improves system performance when wait states are required. Only the NVM main array address space is cached. It is a direct-mapped

cache that implements 64 lines of 64 bits (i.e., 512 Bytes). NVM Controller cache can be enabled by writing a '0' to the Cache Disable bit in the Control B register (CTRLB.CACHEDIS).

The cache can be configured to three different modes using the Read Mode bit group in the Control B register (CTRLB.READMODE).

The INVALL command can be issued using the Command bits in the Control A register to invalidate all cache lines (CTRLA.CMD=INVALL). Commands affecting NVM content automatically invalidate cache lines.

28.7. Register Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0		CMD[6:0]						
0x01		15:8	CMDEX[7:0]							
0x02 ... 0x03	Reserved									
0x04	CTRLB	7:0	MANW			RWS[3:0]				
0x05		15:8							SLEEPPRM[1:0]	
0x06		23:16						CACHEDIS	READMODE[1:0]	
0x07		31:24								
0x08	PARAM	7:0	NVMP[7:0]							
0x09		15:8	NVMP[15:8]							
0x0A		23:16	RWWEEP[3:0]					PSZ[2:0]		
0x0B		31:24	RWWEEP[11:4]							
0x0C	INTENCLR	7:0							ERROR	READY
0x0D ... 0x0F	Reserved									
0x10	INTENSET	7:0							ERROR	READY
0x11 ... 0x13	Reserved									
0x14	INTFLAG	7:0							ERROR	READY
0x15 ... 0x17	Reserved									
0x18	STATUS	7:0				NVME	LOCKE	PROGE	LOAD	PRM
0x19		15:8								SB
0x1A ... 0x1B	Reserved									
0x1C	ADDR	7:0	ADDR[7:0]							
0x1D		15:8	ADDR[15:8]							
0x1E		23:16			ADDR[21:16]					
0x1F		31:24								
0x20	LOCK	7:0	LOCK[7:0]							
0x21		15:8	LOCK[15:8]							

28.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

28.8.1. Control A

Name: CTRLA
Offset: 0x00
Reset: 0x0000
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
	CMDEX[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		CMD[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 15:8 – CMDEX[7:0]: Command Execution
When this bit group is written to the key value 0xA5, the command written to CMD will be executed. If a value different from the key value is tried, the write will not be performed and the PROGE status bit will set. PROGE is also set if a previously written command is not completed yet.

The key value must be written at the same time as CMD. If a command is issued through the APB bus on the same cycle as an AHB bus access, the AHB bus access will be given priority. The command will then be executed when the NVM block and the AHB bus are idle.

INTFLAG.READY must be '1' when the command is issued.

Bit 0 of the CMDEX bit group will read back as '1' until the command is issued.

Bits 6:0 – CMD[6:0]: Command
These bits define the command to be executed when the CMDEX key is written.

CMD[6:0]	Group Configuration	Description
0x00-0x01	-	Reserved
0x02	ER	Erase Row - Erases the row addressed by the ADDR register in the NVM main array.
0x03	-	Reserved
0x04	WP	Write Page - Writes the contents of the page buffer to the page addressed by the ADDR register.
0x05	EAR	Erase Auxiliary Row - Erases the auxiliary row addressed by the ADDR register. This command can be given only when the security bit is not set and only to the User Configuration Row.
0x06	WAP	Write Auxiliary Page - Writes the contents of the page buffer to the page addressed by the ADDR register. This command can be given only when the security bit is not set and only to the User Configuration Row.
0x07-0x0E	-	Reserved

CMD[6:0]	Group Configuration	Description
0x0F	WL	Write Lockbits- write the LOCK register
0x1A-0x19	-	Reserved
0x1A	RWWEEER	RWWEE Erase Row - Erases the row addressed by the ADDR register in the RWWEE array.
0x1B	-	Reserved
0x1C	RWWEEWP	RWWEE Write Page - Writes the contents of the page buffer to the page addressed by the ADDR register in the RWWEE array.
0x1D-0x3F	-	Reserved
0x40	LR	Lock Region - Locks the region containing the address location in the ADDR register.
0x41	UR	Unlock Region - Unlocks the region containing the address location in the ADDR register.
0x42	SPRM	Sets the Power Reduction Mode.
0x43	CPRM	Clears the Power Reduction Mode.
0x44	PBC	Page Buffer Clear - Clears the page buffer.
0x45	SSB	Set Security Bit - Sets the security bit by writing 0x00 to the first byte in the lockbit row.
0x46	INVALL	Invalidates all cache lines.
0x47-0x7F	-	Reserved

28.8.2. Control B

Name: CTRLB
Offset: 0x04
Reset: 0x00000080
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						CACHEDIS	READMODE[1:0]	
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
							SLEEPPRM[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	MANW			RWS[3:0]				
Access	R/W			R/W	R/W	R/W	R/W	
Reset	1			0	0	0	0	

Bit 18 – CACHEDIS: Cache Disable
This bit is used to disable the cache.

Value	Description
0	The cache is enabled
1	The cache is disabled

Bits 17:16 – READMODE[1:0]: NVMCTRL Read Mode

Value	Name	Description
0x0	NO_MISS_PENALTY	The NVM Controller (cache system) does not insert wait states on a cache miss. Gives the best system performance.
0x1	LOW_POWER	Reduces power consumption of the cache system, but inserts a wait state each time there is a cache miss. This mode may not be relevant if CPU performance is required, as the application will be stalled and may lead to increased run time.

Value	Name	Description
0x2	DETERMINISTIC	The cache system ensures that a cache hit or miss takes the same amount of time, determined by the number of programmed Flash wait states. This mode can be used for real-time applications that require deterministic execution timings.
0x3	Reserved	

Bits 9:8 – SLEEPPRM[1:0]: Power Reduction Mode during Sleep
Indicates the Power Reduction Mode during sleep.

Value	Name	Description
0x0	WAKEUPACCESS	NVM block enters low-power mode when entering sleep. NVM block exits low-power mode upon first access.
0x1	WAKEUPINSTANT	NVM block enters low-power mode when entering sleep. NVM block exits low-power mode when exiting sleep.
0x2	Reserved	
0x3	DISABLED	Auto power reduction disabled.

Bit 7 – MANW: Manual Write
Note that reset value of this bit is '1'.

Value	Description
0	Writing to the last word in the page buffer will initiate a write operation to the page addressed by the last write operation. This includes writes to memory and auxiliary rows.
1	Write commands must be issued through the CTRLA.CMD register.

Bits 4:1 – RWS[3:0]: NVM Read Wait States
These bits control the number of wait states for a read operation. '0' indicates zero wait states, '1' indicates one wait state, etc., up to 15 wait states.
This register is initialized to 0 wait states. Software can change this value based on the NVM access time and system frequency.

28.8.3. NVM Parameter

Name: PARAM
Offset: 0x08
Reset: 0x000XXXXX
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	RWWEEP[11:4]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RWWEEP[3:0]					PSZ[2:0]		
Access	R	R	R	R		R	R	R
Reset	0	0	0	0		x	x	x

Bit	15	14	13	12	11	10	9	8
	NVMP[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

Bit	7	6	5	4	3	2	1	0
	NVMP[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

Bits 31:20 – RWWEEP[11:0]: Read While Write EEPROM emulation area Pages
Indicates the number of pages in the RWW EEPROM emulation address space.

Bits 18:16 – PSZ[2:0]: Page Size
Indicates the page size. Not all devices of the device families will provide all the page sizes indicated in the table.

Value	Name	Description
0x0	8	8 bytes
0x1	16	16 bytes
0x2	32	32 bytes
0x3	64	64 bytes
0x4	128	128 bytes
0x5	256	256 bytes
0x6	512	512 bytes
0x7	1024	1024 bytes

Bits 15:0 – NVMP[15:0]: NVM Pages
Indicates the number of pages in the NVM main address space.

28.8.4. Interrupt Enable Clear

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Name: INTENCLR
Offset: 0x0C
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
							ERROR	READY
Access							R/W	R/W
Reset							0	0

Bit 1 – ERROR: Error Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the ERROR interrupt enable.

This bit will read as the current value of the ERROR interrupt enable.

Bit 0 – READY: NVM Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the READY interrupt enable.

This bit will read as the current value of the READY interrupt enable.

28.8.5. Interrupt Enable Set

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Name: INTENSET
Offset: 0x10
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
							ERROR	READY
Access							R/W	R/W
Reset							0	0

Bit 1 – ERROR: Error Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit sets the ERROR interrupt enable.

This bit will read as the current value of the ERROR interrupt enable.

Bit 0 – READY: NVM Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit sets the READY interrupt enable.

This bit will read as the current value of the READY interrupt enable.

28.8.6. Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x14
Reset: 0x00
Property: –

Bit	7	6	5	4	3	2	1	0
							ERROR	READY
Access							R/W	R
Reset							0	0

Bit 1 – ERROR: Error
This flag is set on the occurrence of an NVME, LOCKE or PROGE error.

This bit can be cleared by writing a '1' to its bit location.

Value	Description
0	No errors have been received since the last clear.
1	At least one error has occurred since the last clear.

Bit 0 – READY: NVM Ready

Value	Description
0	The NVM controller is busy programming or erasing.
1	The NVM controller is ready to accept a new command.

28.8.7. Status

Name: STATUS
Offset: 0x18
Reset: 0x0X00
Property: –

Bit	15	14	13	12	11	10	9	8
								SB
Access								R
Reset								x

Bit	7	6	5	4	3	2	1	0
				NVME	LOCKE	PROGE	LOAD	PRM
Access				R/W	R/W	R/W	R/W	R
Reset				0	0	0	0	0

Bit 8 – SB: Security Bit Status

Value	Description
0	The Security bit is inactive.
1	The Security bit is active.

Bit 4 – NVME: NVM Error
This bit can be cleared by writing a '1' to its bit location.

Value	Description
0	No programming or erase errors have been received from the NVM controller since this bit was last cleared.
1	At least one error has been registered from the NVM Controller since this bit was last cleared.

Bit 3 – LOCKE: Lock Error Status
This bit can be cleared by writing a '1' to its bit location.

Value	Description
0	No programming of any locked lock region has happened since this bit was last cleared.
1	Programming of at least one locked lock region has happened since this bit was last cleared.

Bit 2 – PROGE: Programming Error Status
This bit can be cleared by writing a '1' to its bit location.

Value	Description
0	No invalid commands or bad keywords were written in the NVM Command register since this bit was last cleared.
1	An invalid command and/or a bad keyword was/were written in the NVM Command register since this bit was last cleared.

Bit 1 – LOAD: NVM Page Buffer Active Loading
This bit indicates that the NVM page buffer has been loaded with one or more words. Immediately after an NVM load has been performed, this flag is set. It remains set until a page write or a page buffer clear (PBCLR) command is given.

This bit can be cleared by writing a '1' to its bit location.

Bit 0 – PRM: Power Reduction Mode
This bit indicates the current NVM power reduction state. The NVM block can be set in power reduction mode in two ways: through the command interface or automatically when entering sleep with SLEEPPRM set accordingly.

PRM can be cleared in three ways: through AHB access to the NVM block, through the command interface (SPRM and CPRM) or when exiting sleep with SLEEPPRM set accordingly.

Value	Description
0	NVM is not in power reduction mode.
1	NVM is in power reduction mode.

28.8.8. Address

Name: ADDR
Offset: 0x1C
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			ADDR[21:16]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 21:0 – ADDR[21:0]: NVM Address
ADDR drives the hardware (16-bit) address to the NVM when a command is executed using CMDEX. This register is also automatically updated when writing to the page buffer.

28.8.9. Lock Section

Name: LOCK
Offset: 0x20
Reset: 0xXXXX
Property: –

Bit	15	14	13	12	11	10	9	8
	LOCK[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	LOCK[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	x

Bits 15:0 – LOCK[15:0]: Region Lock Bits
In order to set or clear these bits, the CMD register must be used.

Default state after erase will be unlocked (0x0000).

Value	Description
0	The corresponding lock region is locked.
1	The corresponding lock region is not locked.

29. PORT - I/O Pin Controller

29.1. Overview

The IO Pin Controller (PORT) controls the I/O pins of the device. The I/O pins are organized in a series of groups, collectively referred to as a port group. Each group can have up to 32 pins that can be configured and controlled individually or as a group. Each pin may either be used for general-purpose I/O under direct application control or be assigned to an embedded device peripheral. When used for general-purpose I/O, each pin can be configured as input or output, with highly configurable driver and pull settings.

All I/O pins have true read-modify-write functionality when used for general-purpose I/O; the direction or the output value of one or more pins may be changed (set, reset or toggled) explicitly without unintentionally changing the state of any other pins in the same port group by a single, atomic 8-, 16- or 32-bit write.

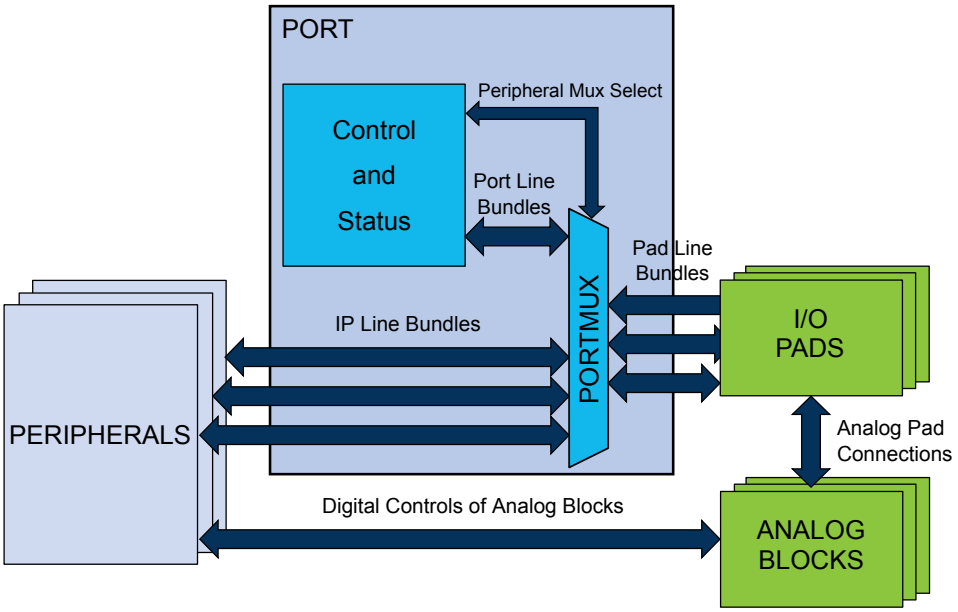
The PORT is connected to the high-speed bus matrix through an AHB/APB bridge. The Pin Direction, Data Output Value and Data Input Value registers may also be accessed using the low-latency CPU local bus (IOBUS; ARM® single-cycle I/O port).

29.2. Features

- Selectable input and output configuration for each individual pin
- Software-controlled multiplexing of peripheral functions on I/O pins
- Flexible pin configuration through a dedicated Pin Configuration register
- Configurable output driver and pull settings:
  - Totem-pole (push-pull)
  - Pull configuration
  - Driver strength
- Configurable input buffer and pull settings:
  - Internal pull-up or pull-down
  - Input sampling criteria
  - Input buffer can be disabled if not needed for lower power consumption
- Input event:
  - Up to four input event pins for each PORT group
  - SET/CLEAR/TOGGLE event actions for each event input on output value of a pin
  - Can be output to pin
- Power saving using STANDBY mode
  - No access to configuration registers
  - Possible access to data registers (DIR, OUT or IN)

29.3. Block Diagram

Figure 29-1. PORT Block Diagram



29.4. Signal Description

Table 29-1. Signal description for PORT

Signal name	Type	Description
Pxy	Digital I/O	General-purpose I/O pin y in group x

Refer to the *I/O Multiplexing and Considerations* for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

Related Links

I/O Multiplexing and Considerations on page 29

29.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly as following.

29.5.1. I/O Lines

The I/O lines of the PORT are mapped to pins of the physical device. The following naming scheme is used:

Each line bundle with up to 32 lines is assigned an identifier 'xy', with letter x=A, B, C… and two-digit number y=00, 01, …31. Examples: A24, C03.

PORT pins are labeled 'Pxy' accordingly, for example PA24, PC03. This identifies each pin in the device uniquely.

Each pin may be controlled by one or more peripheral multiplexer settings, which allow the pad to be routed internally to a dedicated peripheral function. When the setting is enabled, the selected peripheral

has control over the output state of the pad, as well as the ability to read the current physical pad state. Refer to *I/O Multiplexing and Considerations* for details.

Device-specific configurations may cause some lines (and the corresponding Pxy pin) not to be implemented.

Related Links

I/O Multiplexing and Considerations on page 29

29.5.2. Power Management

During reset, all PORT lines are configured as inputs with input buffers, output buffers and pull disabled.

If the PORT peripheral is shut down, the latches in the pad will keep their current configuration in any sleep mode, such as the output value and pull settings. However, the PORT configuration registers and input synchronizers will lose their contents, and these will not be restored when PORT is powered up again. Therefore, user must reconfigure the PORT peripheral at power-up to ensure it is in a well-defined state before use.

The PORT will continue operating in any sleep mode where the selected module source clock is running because the selected module source clock is still running.

29.5.3. Clocks

The PORT bus clock (CLK_PORT_APB) can be enabled and disabled in the Main Clock module, and the default state of CLK_PORT_APB can be found in the *Peripheral Clock Masking* section in *MCLK – Main Clock*.

The PORT is fed by two different clocks: a CPU main clock, which allows the CPU to access the PORT through the low latency CPU local bus (IOBUS); an APB clock, which is a divided clock of the CPU main clock and allows the CPU to access the registers of PORT through the high-speed matrix and the AHB/APB bridge.

The priority of IOBUS accesses is higher than event accesses and APB accesses. The EVSYS and APB will insert wait states in the event of concurrent PORT accesses.

The PORT input synchronizers use the CPU main clock so that the resynchronization delay is minimized with respect to the APB clock.

Related Links

MCLK – Main Clock on page 148

29.5.4. DMA

Not applicable.

29.5.5. Interrupts

Not applicable.

29.5.6. Events

The events of this peripheral are connected to the Event System.

Related Links

EVSYS – Event System on page 536

29.5.7. Debug Operation

When the CPU is halted in debug mode, this peripheral will continue normal operation. If the peripheral is configured to require periodical service by the CPU through interrupts or similar, improper operation or

data loss may result during debugging. This peripheral can be forced to halt operation during debugging - refer to the Debug Control (DBGCTRL) register for details.

29.5.8. Register Access Protection

All registers with write-access can be optionally write-protected by the Peripheral Access Controller (PAC).

Note: Optional write-protection is indicated by the "PAC Write-Protection" property in the register description.

When the CPU is halted in debug mode, all write-protection is automatically disabled. Write-protection does not apply for accesses through an external debugger.

Related Links

PAC - Peripheral Access Controller on page 58

29.5.9. Analog Connections

Analog functions are connected directly between the analog blocks and the I/O pads using analog buses. However, selecting an analog peripheral function for a given pin will disable the corresponding digital features of the pad.

29.5.10. CPU Local Bus

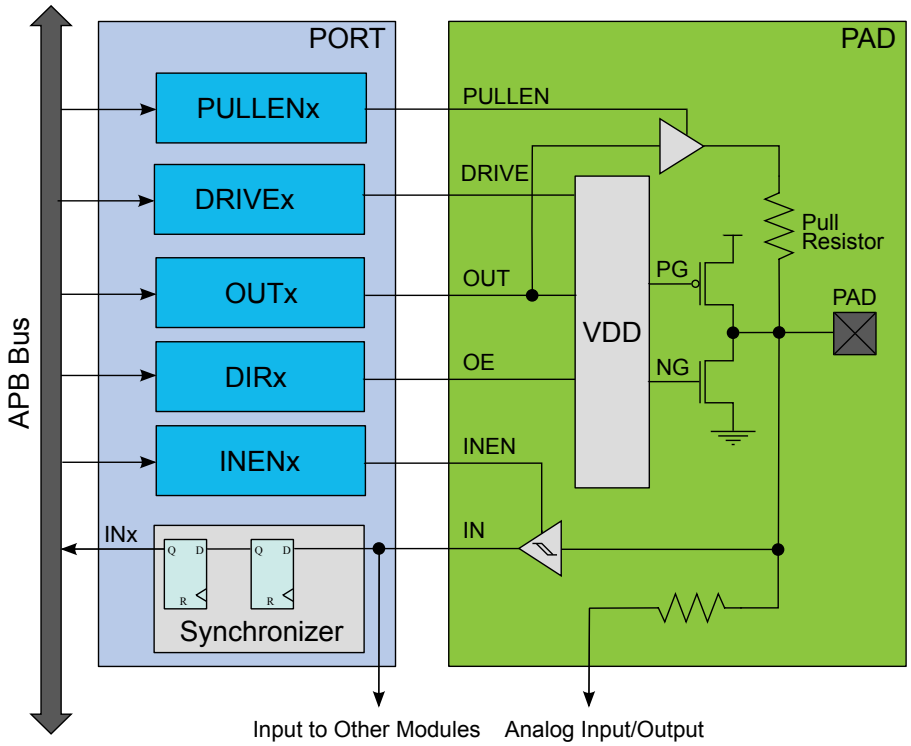
The CPU local bus (IOBUS) is an interface that connects the CPU directly to the PORT. It is a single-cycle bus interface, which does not support wait states. It supports 8-bit, 16-bit and 32-bit sizes.

This bus is generally used for low latency operation. The Data Direction (DIR) and Data Output Value (OUT) registers can be read, written, set, cleared or be toggled using this bus, and the Data Input Value (IN) registers can be read.

Since the IOBUS cannot wait for IN register resynchronization, the Control register (CTRL) must be configured to continuous sampling of all pins that need to be read via the IOBUS in order to prevent stale data from being read.

29.6. Functional Description

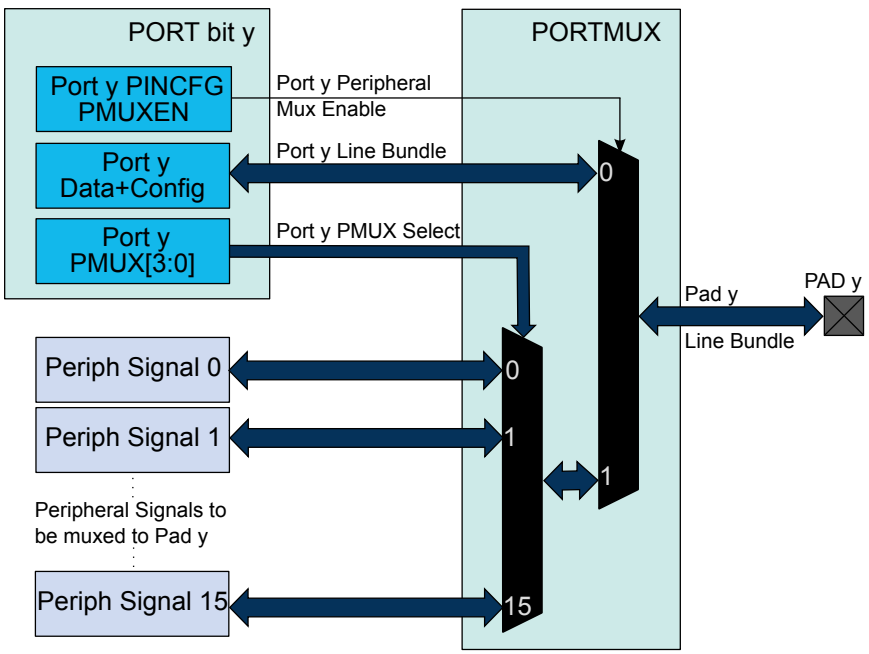
Figure 29-2. Overview of the PORT



29.6.1. Principle of Operation

Each pin group of up to 32 pins is controlled by the registers in PORT, as described in the figure. These registers in PORT are duplicated for each pin group, with increasing base addresses.

Figure 29-3. Overview of the peripheral functions multiplexing



The I/O pins of the device are controlled by PORT peripheral registers. Each port pin has a corresponding bit in the Data Direction (DIR) and Data Output Value (OUT) registers to enable that pin as an output and to define the output state.

The direction of each pin in a pin group is configured by the DIR register. If a bit in DIR is set to '1', the corresponding pin is configured as an output pin. If a bit in DIR is set to '0', the corresponding pin is configured as an input pin.

When the direction is set as output, the corresponding bit in the OUT register will set the level of the pin. If bit y in OUT is written to '1', pin y is driven HIGH. If bit y in OUT is written to '0', pin y is driven LOW. Pin configuration can be set by Pin Configuration (PINCFGy) registers, with y=00, 01, ..31 representing the bit position.

The Data Input Value (IN) is set as the input value of a port pin with resynchronization to the PORT clock. To reduce power consumption, these input synchronizers are clocked only when system requires reading the input value. The value of the pin can always be read, whether the pin is configured as input or output. If the Input Enable bit in the Pin Configuration registers (PINCFGy.INEN) is '0', the input value will not be sampled.

In PORT, the Peripheral Multiplexer Enable bit in the PINCFGy register (PINCFGY.PMUXEN) can be written to '1' to enable the connection between peripheral functions and individual I/O pins. The Peripheral Multiplexing (PMUXn) registers select the peripheral function for the corresponding pin. This will override the connection between the PORT and that I/O pin, and connect the selected peripheral signal to the particular I/O pin instead of the PORT line bundle.

29.6.2. Basic Operation

29.6.2.1. Initialization

After reset, all standard function device I/O pads are connected to the PORT with outputs tri-stated and input buffers disabled, even if there is no clock running.

However, specific pins, such as those used for connection to a debugger, may be configured differently, as required by their special function.

29.6.2.2. Operation

Each I/O pin y can be controlled by the registers in PORT. Each pin group has its own set of PORT registers, the base address of the register set for pin y is at byte address PORT + ([y] * 0x4). The index within that register set is [y].

To use pin number y as an *output*, write bit y of the DIR register to '1'. This can also be done by writing bit y in the DIRSET register to '1' - this will avoid disturbing the configuration of other pins in that group. The y bit in the OUT register must be written to the desired output value.

Similarly, writing an OUTSET bit to '1' will set the corresponding bit in the OUT register to '1'. Writing a bit in OUTCLR to '1' will set that bit in OUT to zero. Writing a bit in OUTTGL to '1' will toggle that bit in OUT.

To use pin y as an *input*, bit y in the DIR register must be written to '0'. This can also be done by writing bit y in the DIRCLR register to '1' - this will avoid disturbing the configuration of other pins in that group. The input value can be read from bit y in register IN as soon as the INEN bit in the Pin Configuration register (PINCFGy.INEN) is written to '1'. Refer to *I/O Multiplexing and Considerations* for details on pin configuration.

By default, the input synchronizer is clocked only when an input read is requested. This will delay the read operation by two CLK_PORT cycles. To remove the delay, the input synchronizers for each pin group of eight pins can be configured to be always active, but this will increase power consumption. This is enabled by writing '1' to the corresponding SAMPLINGn bit field of the CTRL register, see CTRL.SAMPLING for details.

To use pin y as one of the available peripheral functions, the corresponding PMUXEN bit of the PINCFGy register must be '1'. The PINCFGy register for pin y is at byte offset (PINCFG0 + [y]).

The peripheral function can be selected by setting the PMUXO or PMUXE in the PMUXn register. The PMUXO/PMUXE is at byte offset PMUX0 + (y/2). The chosen peripheral must also be configured and enabled.

Related Links

I/O Multiplexing and Considerations on page 29

29.6.3. I/O Pin Configuration

The Pin Configuration register (PINCFGy) is used for additional I/O pin configuration. A pin can be set in a totem-pole or pull configuration.

As pull configuration is done through the Pin Configuration register, all intermediate PORT states during switching of pin direction and pin values are avoided.

The I/O pin configurations are described further in this chapter, and summarized in Table 29-2 Pin Configurations Summary on page 512.

29.6.3.1. Pin Configurations Summary

Table 29-2. Pin Configurations Summary

DIR	INEN	PULLEN	OUT	Configuration
0	0	0	X	Reset or analog I/O: all digital disabled
0	0	1	0	Pull-down; input disabled
0	0	1	1	Pull-up; input disabled
0	1	0	X	Input
0	1	1	0	Input with pull-down
0	1	1	1	Input with pull-up
1	0	X	X	Output; input disabled
1	1	X	X	Output; input enabled

29.6.3.2. Input Configuration

Figure 29-4. I/O configuration - Standard Input

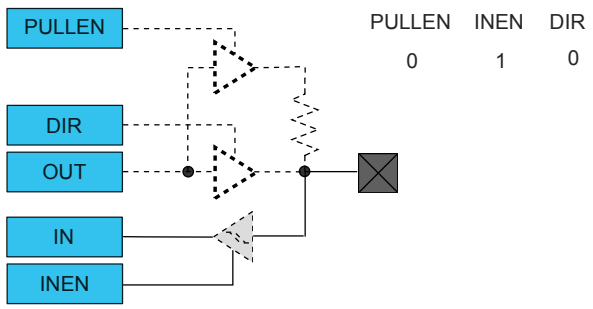
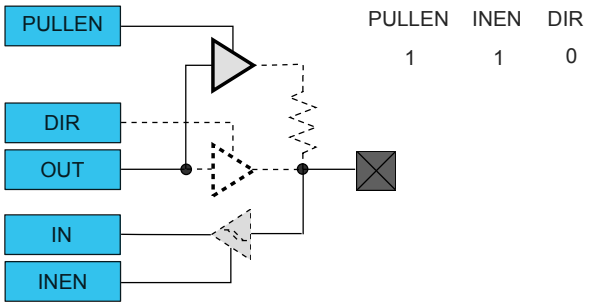


Figure 29-5. I/O Configuration - Input with Pull



Note: When pull is enabled, the pull value is defined by the OUT value.

29.6.3.3. Totem-Pole Output

When configured for totem-pole (push-pull) output, the pin is driven low or high according to the corresponding bit setting in the OUT register. In this configuration there is no current limitation for sink or source other than what the pin is capable of. If the pin is configured for input, the pin will float if no external pull is connected.

Note: Enabling the output driver will automatically disable pull.

Figure 29-6. I/O Configuration - Totem-Pole Output with Disabled Input

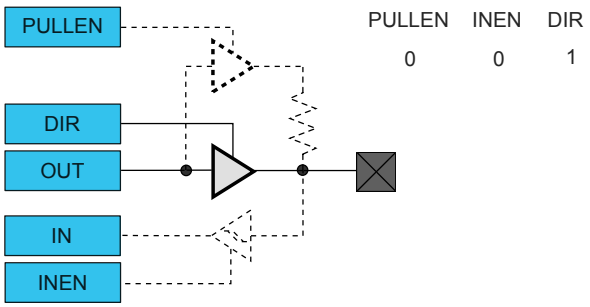


Figure 29-7. I/O Configuration - Totem-Pole Output with Enabled Input

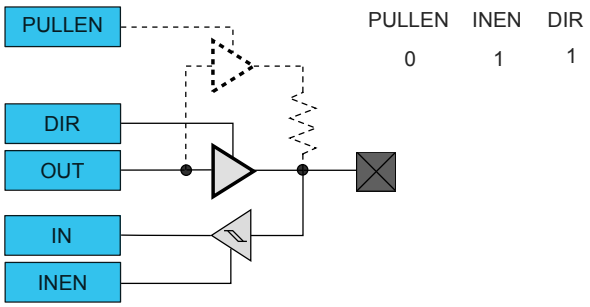
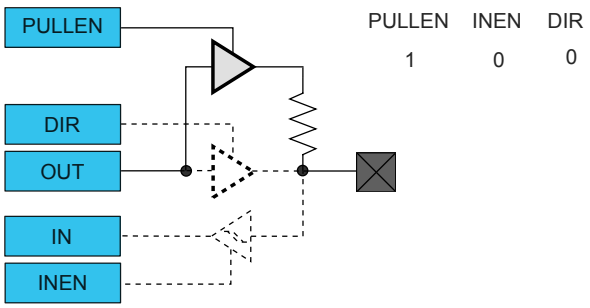


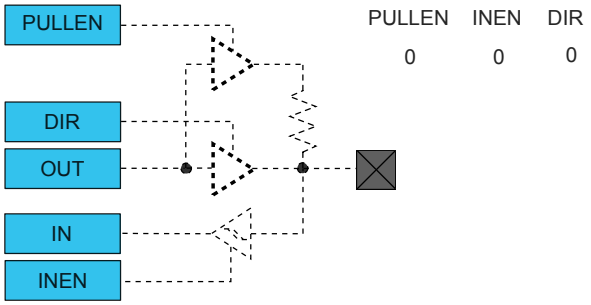
Figure 29-8. I/O Configuration - Output with Pull



29.6.3.4. Digital Functionality Disabled

Neither Input nor Output functionality are enabled.

Figure 29-9. I/O Configuration - Reset or Analog I/O: Digital Output, Input and Pull Disabled



29.6.4. Events

The PORT allows input events to control individual I/O pins. These input events are generated by the EVSYS module and can originate from a different clock domain than the PORT module.

The PORT can perform the following actions:

- Output (OUT): I/O pin will be set when the incoming event has a high level ('1') and cleared when the incoming event has a low-level ('0').
- Set (SET): I/O pin will be set when an incoming event is detected.
- Clear (CLR): I/O pin will be cleared when an incoming event is detected.
- Toggle (TGL): I/O pin will toggle when an incoming event is detected.

The event is output to pin without any internal latency. For SET, CLEAR and TOGGLE event actions, the action will be executed up to three clock cycles after a rising edge.

The event actions can be configured with the Event Action m bit group in the Event Input Control register(EVCTRL.EVACTm). Writing a '1' to a PORT Event Enable Input m of the Event Control register (EVCTRL.PORTEIm) enables the corresponding action on input event. Writing '0' to this bit disables the corresponding action on input event. Note that several actions can be enabled for incoming events. If several events are connected to the peripheral, any enabled action will be taken for any of the incoming events. Refer to *EVSYS – Event System*. for details on configuring the Event System.

Each event input can address one and only one I/O pin at a time. The selection of the pin is indicated by the PORT Event Pin Identifier of the Event Input Control register (EVCTR.PIDn). On the other hand, one I/O pin can be addressed by up to four different input events. To avoid action conflict on the output value of the register (OUT) of this particular I/O pin, only one action is performed according to the table below.

Note that this truth table can be applied to any SET/CLR/TGL configuration from two to four active input events.

Table 29-3. Priority on Simultaneous SET/CLR/TGL Event Actions

EVACT0	EVACT1	EVACT2	EVACT3	Executed Event Action
SET	SET	SET	SET	SET
CLR	CLR	CLR	CLR	CLR
All Other Combinations				TGL

Be careful when the event is output to pin. Due to the fact the events are received asynchronously, the I/O pin may have unpredictable levels, depending on the timing of when the events are received. When

several events are output to the same pin, the lowest event line will get the access. All other events will be ignored.

Related Links

EVSYS – Event System on page 536

29.6.5. PORT Access Priority

The PORT is accessed by different systems:

- The ARM® CPU through the ARM® single-cycle I/O port (IOBUS)
- The ARM® CPU through the high-speed matrix and the AHB/APB bridge (APB)
- EVSYS through four asynchronous input events

The following priority is adopted:

1. ARM® CPU IOBUS (No wait tolerated)
2. APB
3. EVSYS input events

For input events that require different actions on the same I/O pin, refer to Events on page 514.

29.7. Register Summary

The I/O pins are assembled in pin groups with up to 32 pins. Group 0 consists of the PA pins, and group 1 is for the PB pins, etc. Each pin group has its own PORT registers. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

Offset	Name	Bitpos.								
0x00	DIR	7:0	DIR[7:0]							
0x01		15:8	DIR[15:8]							
0x02		23:16	DIR[23:16]							
0x03		31:24	DIR[31:24]							
0x04	DIRCLR	7:0	DIRCLR[7:0]							
0x05		15:8	DIRCLR[15:8]							
0x06		23:16	DIRCLR[23:16]							
0x07		31:24	DIRCLR[31:24]							
0x08	DIRSET	7:0	DIRSET[7:0]							
0x09		15:8	DIRSET[15:8]							
0x0A		23:16	DIRSET[23:16]							
0x0B		31:24	DIRSET[31:24]							
0x0C	DIRTGL	7:0	DIRTGL[7:0]							
0x0D		15:8	DIRTGL[15:8]							
0x0E		23:16	DIRTGL[23:16]							
0x0F		31:24	DIRTGL[31:24]							
0x10	OUT	7:0	OUT[7:0]							
0x11		15:8	OUT[15:8]							
0x12		23:16	OUT[23:16]							
0x13		31:24	OUT[31:24]							

Offset	Name	Bitpos.								
0x14	OUTCLR	7:0	OUTCLR[7:0]							
0x15		15:8	OUTCLR[15:8]							
0x16		23:16	OUTCLR[23:16]							
0x17		31:24	OUTCLR[31:24]							
0x18	OUTSET	7:0	OUTSET[7:0]							
0x19		15:8	OUTSET[15:8]							
0x1A		23:16	OUTSET[23:16]							
0x1B		31:24	OUTSET[31:24]							
0x1C	OUTTGL	7:0	OUTTGL[7:0]							
0x1D		15:8	OUTTGL[15:8]							
0x1E		23:16	OUTTGL[23:16]							
0x1F		31:24	OUTTGL[31:24]							
0x20	IN	7:0	IN[7:0]							
0x21		15:8	IN[15:8]							
0x22		23:16	IN[23:16]							
0x23		31:24	IN[31:24]							
0x24	CTRL	7:0	SAMPLING[7:0]							
0x25		15:8	SAMPLING[15:8]							
0x26		23:16	SAMPLING[23:16]							
0x27		31:24	SAMPLING[31:24]							
0x28	WRCONFIG	7:0	PINMASK[7:0]							
0x29		15:8	PINMASK[15:8]							
0x2A		23:16		DRVSTR				PULLEN	INEN	PMUXEN
0x2B		31:24	HWSEL	WRPINCFG		WRPMUX	PMUX[3:0]			
0x2C	EVCTRL	7:0	PORTEI0	EVACT0[1:0]		PID0[4:0]				
0x2D		15:8	PORTEI1	EVACT1[1:0]		PID1[4:0]				
0x2E		23:16	PORTEI2	EVACT2[1:0]		PID2[4:0]				
0x2F		31:24	PORTEI3	EVACT3[1:0]		PID3[4:0]				
0x30	PMUX0	7:0	PMUXO[3:0]				PMUXE[3:0]			
0x31	PMUX1	7:0	PMUXO[3:0]				PMUXE[3:0]			
…	…	…								
0x3E	PMUX14	7:0	PMUXO[3:0]				PMUXE[3:0]			
0x3F	PMUX15	7:0	PMUXO[3:0]				PMUXE[3:0]			
0x40	PINCFG0	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x41	PINCFG1	7:0		DRVSTR				PULLEN	INEN	PMUXEN
…	…	…								
0x5E	PINCFG30	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x5F	PINCFG31	7:0		DRVSTR				PULLEN	INEN	PMUXEN

29.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection on page 509.

29.8.1. Data Direction

This register allows the user to configure one or more I/O pins as an input or output. This register can be manipulated without doing a read-modify-write operation by using the Data Direction Toggle (DIRTGL), Data Direction Clear (DIRCLR) and Data Direction Set (DIRSET) registers.

Name: DIR
Offset: 0x00 + n*0x80 [n=0,1]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	DIR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DIR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DIR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DIR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DIR[31:0]: Port Data Direction

These bits set the data direction for the individual I/O pins in the pin group.

Value	Description
0	The corresponding I/O pin in the pin group is configured as an input.
1	The corresponding I/O pin in the pin group is configured as an output.

29.8.2. Data Direction Clear

This register allows the user to set one or more I/O pins as an input, without doing a read-modify-write operation. Changes in this register will also be reflected in the Data Direction (DIR), Data Direction Toggle (DIRTGL) and Data Direction Set (DIRSET) registers.

Name: DIRCLR
Offset: 0x04 + n*0x80 [n=0..2]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	DIRCLR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DIRCLR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DIRCLR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DIRCLR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DIRCLR[31:0]: Port Data Direction Clear
Writing a '0' to a bit has no effect.

Writing a '1' to a bit will clear the corresponding bit in the DIR register, which configures the I/O pin as an input.

Value	Description
0	The corresponding I/O pin in the group will keep its configuration.
1	The corresponding I/O pin in the pin group is configured as input.

29.8.3. Data Direction Set

This register allows the user to set one or more I/O pins as an output, without doing a read-modify-write operation. Changes in this register will also be reflected in the Data Direction (DIR), Data Direction Toggle (DIRTGL) and Data Direction Clear (DIRCLR) registers.

Name: DIRSET
Offset: 0x08 + n*0x80 [n=0..2]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	DIRSET[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DIRSET[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DIRSET[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DIRSET[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DIRSET[31:0]: Port Data Direction Set
Writing '0' to a bit has no effect.

Writing '1' to a bit will set the corresponding bit in the DIR register, which configures the I/O pin as an output.

Value	Description
0	The corresponding I/O pin in the group will keep its configuration.
1	The corresponding I/O pin in the pin group is configured as an output.

29.8.4. Data Direction Toggle

This register allows the user to toggle the direction of one or more I/O pins, without doing a read-modify-write operation. Changes in this register will also be reflected in the Data Direction (DIR), Data Direction Set (DIRSET) and Data Direction Clear (DIRCLR) registers.

Name: DIRTGL
Offset: 0x0C + n*0x80 [n=0..2]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	DIRTGL[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DIRTGL[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DIRTGL[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DIRTGL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DIRTGL[31:0]: Port Data Direction Toggle
Writing '0' to a bit has no effect.

Writing '1' to a bit will toggle the corresponding bit in the DIR register, which reverses the direction of the I/O pin.

Value	Description
0	The corresponding I/O pin in the group will keep its configuration.
1	The direction of the corresponding I/O pin is toggled.

29.8.5. Data Output Value

This register sets the data output drive value for the individual I/O pins in the PORT.
This register can be manipulated without doing a read-modify-write operation by using the Data Output Value Clear (OUTCLR), Data Output Value Set (OUTSET), and Data Output Value Toggle (OUTTGL) registers.

Name: OUT
Offset: 0x10 + n*0x80 [n=0..2]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	OUT[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	OUT[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	OUT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	OUT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – OUT[31:0]: Port Data Output Value
For pins configured as outputs via the Data Direction register (DIR), these bits set the logical output drive level.

For pins configured as inputs via the Data Direction register (DIR) and with pull enabled via the Pull Enable bit in the Pin Configuration register (PINCFG.PULLEN), these bits will set the input pull direction.

Value	Description
0	The I/O pin output is driven low, or the input is connected to an internal pull-down.
1	The I/O pin output is driven high, or the input is connected to an internal pull-up.

29.8.6. Data Output Value Clear

This register allows the user to set one or more output I/O pin drive levels low, without doing a read-modify-write operation. Changes in this register will also be reflected in the Data Output Value (OUT), Data Output Value Toggle (OUTTGL) and Data Output Value Set (OUTSET) registers.

Name: OUTCLR
Offset: 0x14 + n*0x80[n=0..2]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	OUTCLR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	OUTCLR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	OUTCLR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	OUTCLR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – OUTCLR[31:0]: PORT Data Output Value Clear
Writing '0' to a bit has no effect.

Writing '1' to a bit will clear the corresponding bit in the OUT register. Pins configured as outputs via the Data Direction register (DIR) will be set to low output drive level. Pins configured as inputs via DIR and with pull enabled via the Pull Enable bit in the Pin Configuration register (PINCFG.PULLEN) will set set the input pull direction to an internal pull-down.

Value	Description
0	The corresponding I/O pin in the group will keep its configuration.
1	The corresponding I/O pin output is driven low, or the input is connected to an internal pull-down.

29.8.7. Data Output Value Set

This register allows the user to set one or more output I/O pin drive levels high, without doing a read-modify-write operation. Changes in this register will also be reflected in the Data Output Value (OUT), Data Output Value Toggle (OUTTGL) and Data Output Value Clear (OUTCLR) registers.

Name: OUTSET
Offset: 0x18 + n*0x80 [n=0..2]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	OUTSET[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	OUTSET[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	OUTSET[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	OUTSET[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – OUTSET[31:0]: PORT Data Output Value Set
Writing '0' to a bit has no effect.

Writing '1' to a bit will set the corresponding bit in the OUT register, which sets the output drive level high for I/O pins configured as outputs via the Data Direction register (DIR). For pins configured as inputs via Data Direction register (DIR) with pull enabled via the Pull Enable register (PULLEN), these bits will set the input pull direction to an internal pull-up.

Value	Description
0	The corresponding I/O pin in the group will keep its configuration.
1	The corresponding I/O pin output is driven high, or the input is connected to an internal pull-up.

29.8.8. Data Output Value Toggle

This register allows the user to toggle the drive level of one or more output I/O pins, without doing a read-modify-write operation. Changes in this register will also be reflected in the Data Output Value (OUT), Data Output Value Set (OUTSET) and Data Output Value Clear (OUTCLR) registers.

Name: OUTTGL
Offset: 0x1C + n*0x80 [n=0..2]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	OUTTGL[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	OUTTGL[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	OUTTGL[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	OUTTGL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – OUTTGL[31:0]: PORT Data Output Value Toggle

Writing '0' to a bit has no effect.

Writing '1' to a bit will toggle the corresponding bit in the OUT register, which inverts the output drive level for I/O pins configured as outputs via the Data Direction register (DIR). For pins configured as inputs via Data Direction register (DIR) with pull enabled via the Pull Enable register (PULLEN), these bits will toggle the input pull direction.

Value	Description
0	The corresponding I/O pin in the group will keep its configuration.
1	The corresponding OUT bit value is toggled.

29.8.9. Data Input Value

Name: IN
Offset: 0x20 + n*0x80 [n=0..2]
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	IN[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	IN[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	IN[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	IN[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – IN[31:0]: PORT Data Input Value
These bits are cleared when the corresponding I/O pin input sampler detects a logical low level on the input pin.

These bits are set when the corresponding I/O pin input sampler detects a logical high level on the input pin.

29.8.10. Control

Name: CTRL
Offset: 0x24 + n*0x80 [n=0..2]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	SAMPLING[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	SAMPLING[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SAMPLING[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SAMPLING[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – SAMPLING[31:0]: Input Sampling Mode
Configures the input sampling functionality of the I/O pin input samplers, for pins configured as inputs via the Data Direction register (DIR).

The input samplers are enabled and disabled in sub-groups of eight. Thus if any pins within a byte request continuous sampling, all pins in that eight pin sub-group will be continuously sampled.

Value	Description
0	The I/O pin input synchronizer is disabled.
1	The I/O pin input synchronizer is enabled.

29.8.11. Write Configuration

This write-only register is used to configure several pins simultaneously with the same configuration and/or peripheral multiplexing.

In order to avoid side effect of non-atomic access, 8-bit or 16-bit writes to this register will have no effect. Reading this register always returns zero.

Name: WRCONFIG
Offset: 0x28 + n*0x80 [n=0..2]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	HWSEL	WRPINCFG		WRPMUX	PMUX[3:0]			
Access	W	W		W	W	W	W	W
Reset	0	0		0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
		DRVSTR				PULLEN	INEN	PMUXEN
Access		W				W	W	W
Reset		0				0	0	0

Bit	15	14	13	12	11	10	9	8
	PINMASK[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PINMASK[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit 31 – HWSEL: Half-Word Select
This bit selects the half-word field of a a 32-pin group to be reconfigured in the atomic write operation.

This bit will always read as zero.

Value	Description
0	The lower 16 pins of the PORT group will be configured.
1	The upper 16 pins of the PORT group will be configured.

Bit 30 – WRPINCFG: Write PINCFG
This bit determines whether the atomic write operation will update the Pin Configuration register (PINCFGy) or not for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits.

Writing '0' to this bit has no effect.

Writing '1' to this bit updates the configuration of the selected pins with the written WRCONFIG.DRVSTR, WRCONFIG.PULLEN, WRCONFIG.INEN, WRCONFIG.PMUXEN and WRCONFIG.PINMASK values.

This bit will always read as zero.

Value	Description
0	The PINCFGy registers of the selected pins will not be updated.
1	The PINCFGy registers of the selected pins will be updated.

Bit 28 – WRPMUX: Write PMUX
This bit determines whether the atomic write operation will update the Peripheral Multiplexing register (PMUXn) or not for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits.

Writing '0' to this bit has no effect.

Writing '1' to this bit updates the pin multiplexer configuration of the selected pins with the written WRCONFIG. PMUX value.

This bit will always read as zero.

Value	Description
0	The PMUXn registers of the selected pins will not be updated.
1	The PMUXn registers of the selected pins will be updated.

Bits 27:24 – PMUX[3:0]: Peripheral Multiplexing
These bits determine the new value written to the Peripheral Multiplexing register (PMUXn) for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits, when the WRCONFIG.WRPMUX bit is set.

These bits will always read as zero.

Bit 22 – DRVSTR: Output Driver Strength Selection
This bit determines the new value written to PINCFGy.DRVSTR for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits, when the WRCONFIG.WRPINCFG bit is set.

This bit will always read as zero.

Bit 18 – PULLEN: Pull Enable
This bit determines the new value written to PINCFGy.PULLEN for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits, when the WRCONFIG.WRPINCFG bit is set.

This bit will always read as zero.

Bit 17 – INEN: Input Enable
This bit determines the new value written to PINCFGy.INEN for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits, when the WRCONFIG.WRPINCFG bit is set.

This bit will always read as zero.

Bit 16 – PMUXEN: Peripheral Multiplexer Enable
This bit determines the new value written to PINCFGy.PMUXEN for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits, when the WRCONFIG.WRPINCFG bit is set.

This bit will always read as zero.

Bits 15:0 – PINMASK[15:0]: Pin Mask for Multiple Pin Configuration
These bits select the pins to be configured within the half-word group selected by the WRCONFIG.HWSEL bit.

These bits will always read as zero.

Value	Description
0	The configuration of the corresponding I/O pin in the half-word group will be left unchanged.
1	The configuration of the corresponding I/O pin in the half-word pin group will be updated.

29.8.12. Event Input Control

There is one 32-bit Event Input Control register for each PORT group. Each byte of this register addresses a group of 32- bit I/O lines. The x denotes the number of the PORT group.

Name: EVCTRL
Offset: 0x2C + n*0x80 [n=0..2]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	PORTEI3	EVACT3[1:0]		PID3[4:0]				
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	PORTEI2	EVACT2[1:0]		PID2[4:0]				
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	PORTEI1	EVACT1[1:0]		PID1[4:0]				
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PORTEI0	EVACT0[1:0]		PID0[4:0]				
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31,23,15,7 – PORTEIx: PORT Event Input x Enable [x = 3..0]

Value	Description
0	The event action x (EVACTx) will not be triggered on any incoming event.
1	The event action x (EVACTx) will be triggered on any incoming event.

Bits 30:29, 22:21,14:13,6:5 – EVACTx: PORT Event Action x [x = 3..0]
These bits define the event action the PORT will perform on event input x. See also Table 29-4 PORT Event x Action (x = [3..0]) on page 531.

Bits 28:24,20:16,12:8,4:0 – PIDx: PORT Event Pin Identifier x [x = 3..0]
These bits define the I/O pin on which the event action will be performed, according to Table 29-5 PORT Event x Pin Identifier (x = [3..0]) on page 531.

Table 29-4. PORT Event x Action (x = [3..0])

Value	Name	Description
0x0	OUT	Output register of pin will be set to level of event.
0x1	SET	Set output register of pin on event.
0x2	CLR	Clear output register of pin on event.
0x3	TGL	Toggle output register of pin on event.

Table 29-5. PORT Event x Pin Identifier (x = [3..0])

Value	Name	Description
0x0	PIN0	Event action to be executed on PIN 0.
0x1	PIN1	Event action to be executed on PIN 1.
...	...	...
0x31	PIN31	Event action to be executed on PIN 31.

29.8.13. Peripheral Multiplexing n

There are up to 16 Peripheral Multiplexing registers in each group, one for every set of two subsequent I/O lines. The n denotes the number of the set of I/O lines, while the x denotes the number of the group.

Name: PMUXn
Offset: 0x30 + n [n=0..15] + m*0x80 [m=0..2]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	PMUXO[3:0]				PMUXE[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:4 – PMUXO[3:0]: Peripheral Multiplexing Odd
These bits select the peripheral function for odd-numbered pins (2*n + 1) of a PORT group, if the corresponding PINCFGy.PMUXEN bit is '1'.

Not all possible values for this selection may be valid. For more details, refer to the *I/O Multiplexing and Considerations*.

PMUXO[3:0]	Name	Description
0x0	A	Peripheral function A selected
0x1	B	Peripheral function B selected
0x2	C	Peripheral function C selected
0x3	D	Peripheral function D selected
0x4	E	Peripheral function E selected
0x5	F	Peripheral function F selected
0x6	G	Peripheral function G selected
0x7	H	Peripheral function H selected
0x8	I	Peripheral function I selected
0x9-0xF	-	Reserved

Bits 3:0 – PMUXE[3:0]: Peripheral Multiplexing Even
These bits select the peripheral function for even-numbered pins (2*n) of a PORT group, if the corresponding PINCFGy.PMUXEN bit is '1'.

Not all possible values for this selection may be valid. For more details, refer to the *I/O Multiplexing and Considerations.*

PMUXE[3:0]	Name	Description
0x0	A	Peripheral function A selected
0x1	B	Peripheral function B selected
0x2	C	Peripheral function C selected

PMUXE[3:0]	Name	Description
0x3	D	Peripheral function D selected
0x4	E	Peripheral function E selected
0x5	F	Peripheral function F selected
0x6	G	Peripheral function G selected
0x7	H	Peripheral function H selected
0x8	I	Peripheral function I selected
0x9-0xF	-	Reserved

29.8.14. Pin Configuration y

There are up to 32 Pin Configuration registers in each group, one for each I/O line. The n denotes the number of the I/O line, while the m denotes the number of the group.

Name: PINCFGy
Offset: 0x40 + n*0x1 [n=0..31] + m*0x80 [m=0..2]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
		DRVSTR				PULLEN	INEN	PMUXEN
Access		R/W				R/W	R/W	R/W
Reset		0				0	0	0

Bit 6 – DRVSTR: Output Driver Strength Selection
This bit controls the output driver strength of an I/O pin configured as an output.

Value	Description
0	Pin drive strength is set to normal drive strength.
1	Pin drive strength is set to stronger drive strength.

Bit 2 – PULLEN: Pull Enable
This bit enables the internal pull-up or pull-down resistor of an I/O pin configured as an input.

Value	Description
0	Internal pull resistor is disabled, and the input is in a high-impedance configuration.
1	Internal pull resistor is enabled, and the input is driven to a defined logic level in the absence of external input.

Bit 1 – INEN: Input Enable
This bit controls the input buffer of an I/O pin configured as either an input or output.

Writing a zero to this bit disables the input buffer completely, preventing read-back of the physical pin state when the pin is configured as either an input or output.

Value	Description
0	Input buffer for the I/O pin is disabled, and the input value will not be sampled.
1	Input buffer for the I/O pin is enabled, and the input value will be sampled when required.

Bit 0 – PMUXEN: Peripheral Multiplexer Enable
This bit enables or disables the peripheral multiplexer selection set in the Peripheral Multiplexing register (PMUXn) to enable or disable alternative peripheral control over an I/O pin direction and output drive value.

Writing a zero to this bit allows the PORT to control the pad direction via the Data Direction register (DIR) and output drive value via the Data Output Value register (OUT). The peripheral multiplexer value in PMUXn is ignored. Writing '1' to this bit enables the peripheral selection in PMUXn to control the pad. In this configuration, the physical pin state may still be read from the Data Input Value register (IN) if PINCFGy.INEN is set.

Value	Description
0	The peripheral multiplexer selection is disabled, and the PORT registers control the direction and output drive value.
1	The peripheral multiplexer selection is enabled, and the selected peripheral function controls the direction and output drive value.

30. EVSYS – Event System

30.1. Overview

The Event System (EVSYS) allows autonomous, low-latency and configurable communication between peripherals.

Several peripherals can be configured to generate and/or respond to signals known as events. The exact condition to generate an event, or the action taken upon receiving an event, is specific to each peripheral. Peripherals that respond to events are called event users. Peripherals that generate events are called event generators. A peripheral can have one or more event generators and can have one or more event users.

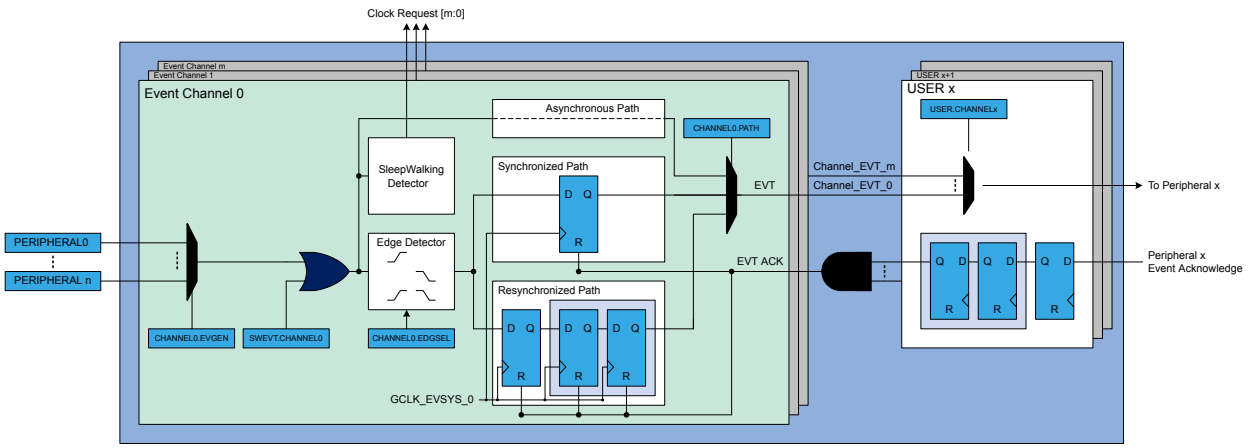
Communication is made without CPU intervention and without consuming system resources such as bus or RAM bandwidth. This reduces the load on the CPU and other system resources, compared to a traditional interrupt-based system.

30.2. Features

- 12 configurable event channels, where each channel can:
  - Be connected to any event generator.
  - Provide a pure asynchronous, resynchronized or synchronous path
- 82 event generators.
- 42 event users.
- Configurable edge detector.
- Peripherals can be event generators, event users, or both.
- SleepWalking and interrupt for operation in sleep modes.
- Software event generation.
- Each event user can choose which channel to respond to.

30.3. Block Diagram

Figure 30-1. Event System Block Diagram



30.4. Signal Description

Not applicable.

30.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

30.5.1. I/O Lines

Not applicable.

30.5.2. Power Management

The EVSYS can be used to wake up the CPU from all sleep modes, even if the clock used by the EVSYS channel and the EVSYS bus clock are disabled. Refer to the *PM – Power Manager* for details on the different sleep modes.

In all sleep modes, although the clock for the EVSYS is stopped, the device still can wake up the EVSYS clock. Some event generators can generate an event when their clocks are stopped. The generic clock for the channel (GCLK_EVSYS_CHANNEL_n) will be restarted if that channel uses a synchronized path or a resynchronized path. It does not need to wake the system from sleep.

Related Links

PM – Power Manager on page 186

30.5.3. Clocks

The EVSYS bus clock (CLK_EVSYS_APB) can be enabled and disabled in the Main Clock module, and the default state of CLK_EVSYS_APB can be found in *Peripheral Clock Masking*.

Each EVSYS channel has a dedicated generic clock (GCLK_EVSYS_CHANNEL_n). These are used for event detection and propagation for each channel. These clocks must be configured and enabled in the generic clock controller before using the EVSYS. Refer to *GCLK - Generic Clock Controller* for details.

Related Links

Peripheral Clock Masking on page 151
GCLK - Generic Clock Controller on page 131

30.5.4. DMA

Not applicable.

30.5.5. Interrupts

The interrupt request line is connected to the Interrupt Controller. Using the EVSYS interrupts requires the interrupt controller to be configured first. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

Nested Vector Interrupt Controller on page 51

30.5.6. Events

Not applicable.

30.5.7. Debug Operation

When the CPU is halted in debug mode, this peripheral will continue normal operation. If the peripheral is configured to require periodical service by the CPU through interrupts or similar, improper operation or

data loss may result during debugging. This peripheral can be forced to halt operation during debugging - refer to the Debug Control (DBGCTRL) register for details.

30.5.8. Register Access Protection

Registers with write-access can be optionally write-protected by the Peripheral Access Controller (PAC), except the following:

- Channel Status (CHSTATUS)
- Interrupt Flag Status and Clear register (INTFLAG)

Note: Optional write-protection is indicated by the "PAC Write-Protection" property in the register description.

When the CPU is halted in debug mode, all write-protection is automatically disabled. Write-protection does not apply for accesses through an external debugger.

30.5.9. Analog Connections

Not applicable.

30.6. Functional Description

30.6.1. Principle of Operation

The Event System consists of several channels which route the internal events from peripherals (generators) to other internal peripherals or IO pins (users). Each event generator can be selected as source for multiple channels, but a channel cannot be set to use multiple event generators at the same time.

A channel path can be configured in asynchronous, synchronous or re-synchronized mode of operation. The mode of operation must be selected based on the requirements of the application.

When using synchronous or resynchronized path, the Event System includes options to transfer events to users when rising, falling or both edges are detected on on event generators.

For further details, refer to “Channel Path on page 539” of this chapter.

30.6.2. Basic Operation

30.6.2.1. Initialization

Before enabling events routing within the system, the Event Users Multiplexer and Event Channels must be configured. The Event Users Multiplexer must be configured first.

For further details about the event user multiplexer configuration, refer to “User Multiplexer Setup on page 539”.

For further details about the event channels configuration, refer to “Event System Channel on page 539”.

30.6.2.2. Enabling, Disabling, and Resetting

The EVSYS is always enabled.

The EVSYS is reset by writing a ‘1’ to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the EVSYS will be reset to their initial state and all ongoing events will be canceled.

Refer to CTRLA.SWRST register for details.

30.6.2.3. User Multiplexer Setup

The user multiplexer defines the channel to be connected to which event user. Each user multiplexer is dedicated to one event user. A user multiplexer receives all event channels output and must be configured to select one of these channels, as shown in Figure 30-1 Event System Block Diagram on page 536. The channel is selected with the Channel bit group in the User register (USERm.CHANNEL).

The user multiplexer must always be configured before the channel. A list of all user multiplexers is found in the User (USERm) register description.

Related Links

USERn on page 555

30.6.2.4. Event System Channel

An event channel can select one event from a list of event generators. Depending on configuration, the selected event could be synchronized, resynchronized or asynchronously sent to the users. When synchronization or resynchronization is required, the channel includes an internal edge detector, allowing the Event System to generate internal events when rising, falling or both edges are detected on the selected event generator.

An event channel is able to generate internal events for the specific software commands. A channel block diagram is shown in Figure 30-1 Event System Block Diagram on page 536.

30.6.2.5. Event Generators

Each event channel can receive the events form all event generators. All event generators are listed in the Event Generator bit field in the Channel n register (CHANNELn.EVGEN). For details on event generation, refer to the corresponding module chapter. The channel event generator is selected by the Event Generator bit group in the Channel register (CHANNELn.EVGEN). By default, the channels are not connected to any event generators (ie, CHANNELn.EVGEN = 0)

30.6.2.6. Channel Path

There are three different ways to propagate the event from an event generator:

- Asynchronous path
- Synchronous path
- Resynchronized path

The path is decided by writing to the Path Selection bit group of the Channel register (CHANNELn.PATH).

Asynchronous Path

When using the asynchronous path, the events are propagated from the event generator to the event user without intervention from the Event System. The GCLK for this channel (GCLK_EVSYS_CHANNEL_n) is not mandatory, meaning that an event will be propagated to the user without any clock latency.

When the asynchronous path is selected, the channel cannot generate any interrupts, and the Channel Status register (CHSTATUS) is always zero. The edge detection is not required and must be disabled by software. Each peripheral event user has to select which event edge must trigger internal actions. For further details, refer to each peripheral chapter description.

Synchronous Path

The synchronous path should be used when the event generator and the event channel share the same generator for the generic clock. If they do not share the same clock, a logic change from the event generator to the event channel might not be detected in the channel, which means that the event will not be propagated to the event user. For details on generic clock generators, refer to *GCLK - Generic Clock Controller*.

When using the synchronous path, the channel is able to generate interrupts. The channel busy n bit in the Channel Status register (CHSTATUS.CHBUSYn) are also updated and available for use.

Resynchronized Path

The resynchronized path are used when the event generator and the event channel do not share the same generator for the generic clock. When the resynchronized path is used, resynchronization of the event from the event generator is done in the channel. For details on generic clock generators, refer to *GCLK - Generic Clock Controller*.

When the resynchronized path is used, the channel is able to generate interrupts. The channel busy n bits in the Channel Status register (CHSTATUS.CHBUSYn) are also updated and available for use.

Related Links

GCLK - Generic Clock Controller on page 131

30.6.2.7. Edge Detection

When synchronous or resynchronized paths are used, edge detection must be enabled. The event system can execute edge detection in three different ways:

- Generate an event only on the rising edge
- Generate an event only on the falling edge
- Generate an event on rising and falling edges.

Edge detection is selected by writing to the Edge Selection bit group of the Channel register (CHANNELn.EDGSEL).

30.6.2.8. Event Latency

An event from an event generator is propagated to an event user with different latency, depending on event channel configuration.

- Asynchronous Path: The maximum routing latency of an external event is related to the internal signal routing and it is device dependent.
- Synchronous Path: The maximum routing latency of an external event is one GCLK_EVSYS_CHANNEL_n clock cycle.
- Resynchronized Path: The maximum routing latency of an external event is three GCLK_EVSYS_CHANNEL_n clock cycles.

The maximum propagation latency of a user event to the peripheral clock core domain is three peripheral clock cycles.

The event generators, event channel and event user clocks ratio must be selected in relation with the internal event latency constraints. Events propagation or event actions in peripherals may be lost if the clock setup violates the internal latencies.

30.6.2.9. The Overrun Channel n Interrupt

The Overrun Channel n interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.OVRn) will be set, and the optional interrupt will be generated in the following cases:

- One or more event users on channel n is not ready when there is a new event.
- An event occurs when the previous event on channel m has not been handled by all event users connected to that channel.

The flag will only be set when using synchronous or resynchronized paths. In the case of asynchronous path, the INTFLAG.OVRn is always read as zero.

30.6.2.10. The Event Detected Channel n Interrupt

The Event Detected Channel n interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.EVDn) is set when an event coming from the event generator configured on channel n is detected.

The flag will only be set when using a synchronous or resynchronized paths. In the case of asynchronous path, the INTFLAG.EVDn is always zero.

30.6.2.11. Channel Status

The Channel Status register (CHSTATUS) shows the status of the channels when using a synchronous or resynchronized path. There are two different status bits in CHSTATUS for each of the available channels:

- The CHSTATUS.CHBUSYn bit will be set when an event on the corresponding channel n has not been handled by all event users connected to that channel.
- The CHSTATUS.USRRDYn bit will be set when all event users connected to the corresponding channel are ready to handle incoming events on that channel.

30.6.2.12. Software Event

A software event can be initiated on a channel by setting the Channel n bit in the Software Event register (SWEVT.CHANNELn) to '1'. Then the software event can be serviced as any event generator; i.e., when the bit is set to '1', an event will be generated on the respective channel.

30.6.3. Interrupts

The EVSYS has the following interrupt sources:

- Overrun Channel n interrupt (OVRn): for details, refer to The Overrun Channel n Interrupt on page 540.
- Event Detected Channel n interrupt (EVDn): for details, refer to The Event Detected Channel n Interrupt on page 541.

These interrupts events are asynchronous wake-up sources. See *Sleep Mode Controller*. Each interrupt source has an interrupt flag which is in the Interrupt Flag Status and Clear (INTFLAG) register. The flag is set when the interrupt is issued. Each interrupt event can be individually enabled by setting a '1' to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by setting a '1' to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register. An interrupt event is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt event works until the interrupt flag is cleared, the interrupt is disabled, or the Event System is reset. See INTFLAG on page 548 for details on how to clear interrupt flags.

All interrupt events from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. Refer to the *Nested Vector Interrupt Controller* for details. The event user must read the INTFLAG register to determine what the interrupt condition is.

Note that interrupts must be globally enabled for interrupt requests to be generated. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

Nested Vector Interrupt Controller on page 51

Sleep Mode Controller on page 191

30.6.4. Sleep Mode Operation

The EVSYS can generate interrupts to wake up the device from any sleep mode.

To be able to run in standby, the Run in Standby bit in the Channel register (CHANNELn.RUNSTDBY) must be set to '1'. When the Generic Clock On Demand bit in Channel register

(CHANNELn.ONDEMAND) is set to '1' and the event generator is detected, the event channel will request its clock (GCLK_EVSYS_CHANNEL_n). The event latency for a resynchronized channel path will increase by two GCLK_EVSYS_CHANNEL_n clock (i.e., up to five GCLK_EVSYS_CHANNEL_n clock cycles).

A channel will behave differently in different sleep modes regarding to CHANNELn.RUNSTDBY and CHANNELn.ONDEMAND, as shown in the table below:

Table 30-1. Event Channel Sleep Behavior

CHANNELn.ONDEMAND	CHANNELn.RUNSTDBY	Sleep Behavior
0	0	Only run in IDLE sleep modes if an event must be propagated. Disabled in STANDBY sleep mode.
0	1	Always run in IDLE and STANDBY sleep modes.
1	0	Only run in IDLE sleep modes if an event must be propagated. Disabled in STANDBY sleep mode. Two GCLK_EVSYS_n latency added in RESYNC path before the event is propagated internally.
1	1	Always run in IDLE and STANDBY sleep modes. Two GCLK_EVSYS_n latency added in RESYNC path before the event is propagated internally.

30.7. Register Summary

30.7.1. Common Registers

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0								SWRST
0x01..0x0B	Reserved									
0x0C	CHSTATUS	7:0	USRRDY7	USRRDY6	USRRDY5	USRRDY4	USRRDY3	USRRDY2	USRRDY1	USRRDY0
0x0D		15:8					USRRDY11	USRRDY10	USRRDY9	USRRDY8
0x0E		23:16	CHBUSY7	CHBUSY6	CHBUSY5	CHBUSY4	CHBUSY3	CHBUSY2	CHBUSY1	CHBUSY0
0x0F		31:24					CHBUSY11	CHBUSY10	CHBUSY9	CHBUSY8
0x10	INTENCLR	7:0	OVR7	OVR6	OVR5	OVR4	OVR3	OVR2	OVR1	OVR0
0x11		15:8					OVR11	OVR10	OVR9	OVR8
0x12		23:16	EVD7	EVD6	EVD5	EVD4	EVD3	EVD2	EVD1	EVD0
0x13		31:24					EVD11	EVD10	EVD9	EVD8
0x14	INTENSET	7:0	OVR7	OVR6	OVR5	OVR4	OVR3	OVR2	OVR1	OVR0
0x15		15:8					OVR11	OVR10	OVR9	OVR8
0x16		23:16	EVD7	EVD6	EVD5	EVD4	EVD3	EVD2	EVD1	EVD0
0x17		31:24					EVD11	EVD10	EVD9	EVD9
0x18	INTFLAG	7:0	OVR7	OVR6	OVR5	OVR4	OVR3	OVR2	OVR1	OVR0
0x19		15:8					OVR11	OVR10	OVR9	OVR8
0x1A		23:16	EVD7	EVD6	EVD5	EVD4	EVD3	EVD2	EVD1	EVD0
0x1B		31:24					EVD11	EVD10	EVD9	EVD9

Offset	Name	Bit Pos.								
0x1C	SWEVT	7:0	CHANNEL[7:0]							
0x1D		15:8					CHANNEL[11:8]			
0x1E		23:16								
0x1F		31:24								

30.7.2. CHANNELn

Offset	Name	Bit Pos.								
0x20 + 0x4*n	CHANNELn on page 550	7:0		EVGEN[6:0]						
0x21 + 0x4*n		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
0x22 + 0x4*n		23:16								
0x23 + 0x4*n		31:24								

30.7.3. USERm

Offset	Name	Bit Pos.								
0x80 + 0x4*m	USERm	7:0				CHANNEL[4:0]				
0x81 + 0x4*m		15:8								
0x82 + 0x4*m		23:16								
0x83 + 0x4*m		31:24								

30.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

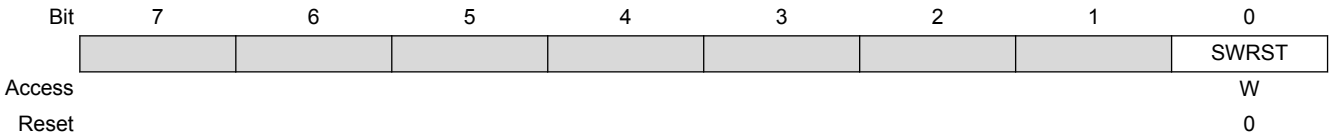
Refer to Register Access Protection on page 538 and *PAC - Peripheral Access Controller*.

Related Links

PAC - Peripheral Access Controller on page 58

30.8.1. Control

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection



Bit	7	6	5	4	3	2	1	0
								SWRST
Access								W
Reset								0

Bit 0 – SWRST: Software Reset
Writing '0' to this bit has no effect.
Writing '1' to this bit resets all registers in the EVSYS to their initial state.

30.8.2. Channel Status

Name: CHSTATUS
Offset: 0x0C
Reset: 0x000000FF
Property: –

Bit	31	30	29	28	27	26	25	24
					CHBUSY11	CHBUSY10	CHBUSY9	CHBUSY8
Access					R	R	R	R
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CHBUSY7	CHBUSY6	CHBUSY5	CHBUSY4	CHBUSY3	CHBUSY2	CHBUSY1	CHBUSY0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
					USRRDY11	USRRDY10	USRRDY9	USRRDY8
Access					R	R	R	R
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	USRRDY7	USRRDY6	USRRDY5	USRRDY4	USRRDY3	USRRDY2	USRRDY1	USRRDY0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	1

Bits 27:16 – CHBUSYn: Channel Busy n [n = 11..0]
This bit is cleared when channel n is idle.

This bit is set if an event on channel n has not been handled by all event users connected to channel n.

Bits 11:0 – USRRDYn: User Ready for Channel n [n = 11..0]
This bit is cleared when at least one of the event users connected to the channel is not ready.

This bit is set when all event users connected to channel n are ready to handle incoming events on channel n.

30.8.3. Interrupt Enable Clear

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Name: INTENCLR
Offset: 0x10
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
					EVD11	EVD10	EVD9	EVD8
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
	EVD7	EVD6	EVD5	EVD4	EVD3	EVD2	EVD1	EVD0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
					OVR11	OVR10	OVR9	OVR8
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	OVR7	OVR6	OVR5	OVR4	OVR3	OVR2	OVR1	OVR0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 27:16 – EVDn: Event Detected Channel n Interrupt Enable [n = 11..0]
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Event Detected Channel n Interrupt Enable bit, which disables the Event Detected Channel n interrupt.

Value	Description
0	The Event Detected Channel n interrupt is disabled.
1	The Event Detected Channel n interrupt is enabled.

Bits 11:0 – OVRn: Overrun Channel n Interrupt Enable[n = 11..0]
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Overrun Channel n Interrupt Enable bit, which disables the Overrun Channel n interrupt.

Value	Description
0	The Overrun Channel n interrupt is disabled.
1	The Overrun Channel n interrupt is enabled.

30.8.4. Interrupt Enable Set

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Name: INTENSET
Offset: 0x14
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
					EVD11	EVD10	EVD9	EVD8
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
	EVD7	EVD6	EVD5	EVD4	EVD3	EVD2	EVD1	EVD0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
					OVR11	OVR10	OVR9	OVR8
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	OVR7	OVR6	OVR5	OVR4	OVR3	OVR2	OVR1	OVR0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 27:16 – EVDn: Event Detected Channel n Interrupt Enable [n = 11..0]
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Event Detected Channel n Interrupt Enable bit, which enables the Event Detected Channel n interrupt.

Value	Description
0	The Event Detected Channel n interrupt is disabled.
1	The Event Detected Channel n interrupt is enabled.

Bits 11:0 – OVRn: Overrun Channel n Interrupt Enable [n = 11..0]
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Overrun Channel n Interrupt Enable bit, which disables the Overrun Channel n interrupt.

Value	Description
0	The Overrun Channel n interrupt is disabled.
1	The Overrun Channel n interrupt is enabled.

30.8.5. Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x18
Reset: 0x00000000
Property: –

Bit	31	30	29	28	27	26	25	24
					EVD11	EVD10	EVD9	EVD8
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
	EVD7	EVD6	EVD5	EVD4	EVD3	EVD2	EVD1	EVD0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
					OVR11	OVR10	OVR9	OVR8
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	OVR7	OVR6	OVR5	OVR4	OVR3	OVR2	OVR1	OVR0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 27:16 – EVDn: Event Detected Channel n Interrupt Enable [n=11..0]
This flag is set on the next CLK_EVSYS_APB cycle when an event is being propagated through the channel, and an interrupt request will be generated if INTENCLR/SET.EVDn is '1'.

When the event channel path is asynchronous, the EVDn interrupt flag will not be set.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Event Detected Channel n interrupt flag.

Bits 11:0 – OVRn: Overrun Channel n Interrupt Enable [n=11..0]
This flag is set on the next CLK_EVSYS_APB cycle when an event is being propagated through the channel, and an interrupt request will be generated if INTENCLR/SET.OVRn is '1'.

When the event channel path is asynchronous, the OVRn interrupt flag will not be set.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Overrun Detected Channel n interrupt flag.

30.8.6. Software Event

Name: SWEVT
Offset: 0x1C
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					CHANNEL11	CHANNEL10	CHANNEL9	CHANNEL8
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CHANNEL7	CHANNEL6	CHANNEL5	CHANNEL4	CHANNEL3	CHANNEL2	CHANNEL1	CHANNEL0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 11:0 – CHANNELn: Channel n Software [n=11..0] Selection
Writing '0' to this bit has no effect.

Writing '1' to this bit will trigger a software event for the channel n.

These bits will always return zero when read.

30.8.7. Channel

This register allows the user to configure channel n. To write to this register, do a single, 32-bit write of all the configuration data.

Name: CHANNELn
Offset: 0x20+n*0x4 [n=0..11]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
Access	R/W	R/W			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

Bit	7	6	5	4	3	2	1	0
		EVGEN[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit 15 – ONDEMAND: Generic Clock On Demand

Value	Description
0	Generic clock for a channel is always on, if the channel is configured and generic clock source is enabled.
1	Generic clock is requested on demand while an event is handled

Bit 14 – RUNSTDBY: Run in Standby

This bit is used to define the behavior during standby sleep mode.

Value	Description
0	The channel is disabled in standby sleep mode.
1	The channel is not stopped in standby sleep mode and depends on the CHANNEL.ONDEMAND

Bits 11:10 – EDGSEL[1:0]: Edge Detection Selection

These bits set the type of edge detection to be used on the channel.

These bits must be written to zero when using the asynchronous path.

Value	Name	Description
0x0	NO_EVT_OUTPUT	No event output when using the resynchronized or synchronous path
0x1	RISING_EDGE	Event detection only on the rising edge of the signal from the event generator
0x2	FALLING_EDGE	Event detection only on the falling edge of the signal from the event generator
0x3	BOTH_EDGES	Event detection on rising and falling edges of the signal from the event generator

Bits 9:8 – PATH[1:0]: Path Selection
These bits are used to choose which path will be used by the selected channel.

The path choice can be limited by the channel source, see the table in USERm on page 555.

Value	Name	Description
0x0	SYNCHRONOUS	Synchronous path
0x1	RESYNCHRONIZED	Resynchronized path
0x2	ASYNCHRONOUS	Asynchronous path
0x3	-	Reserved

Bits 6:0 – EVGEN[6:0]: Event Generator
These bits are used to choose the event generator to connect to the selected channel.

Value	Event Generator	Description
0x00	NONE	No event generator selected
0x01	RTC CMP0	Compare 0 (mode 0 and 1) or Alarm 0 (mode 2)
0x02	RTC CMP1	Compare 1
0x03	RTC OVF	Overflow
0x04	RTC PER0	Period 0
0x05	RTC PER1	Period 1
0x06	RTC PER2	Period 2
0x07	RTC PER3	Period 3
0x08	RTC PER4	Period 4
0x09	RTC PER5	Period 5
0x0A	RTC PER6	Period 6
0x0B	RTC PER7	Period 7
0x0C	EIC EXTINT0	External Interrupt 0
0x0D	EIC EXTINT1	External Interrupt 1
0x0E	EIC EXTINT2	External Interrupt 2

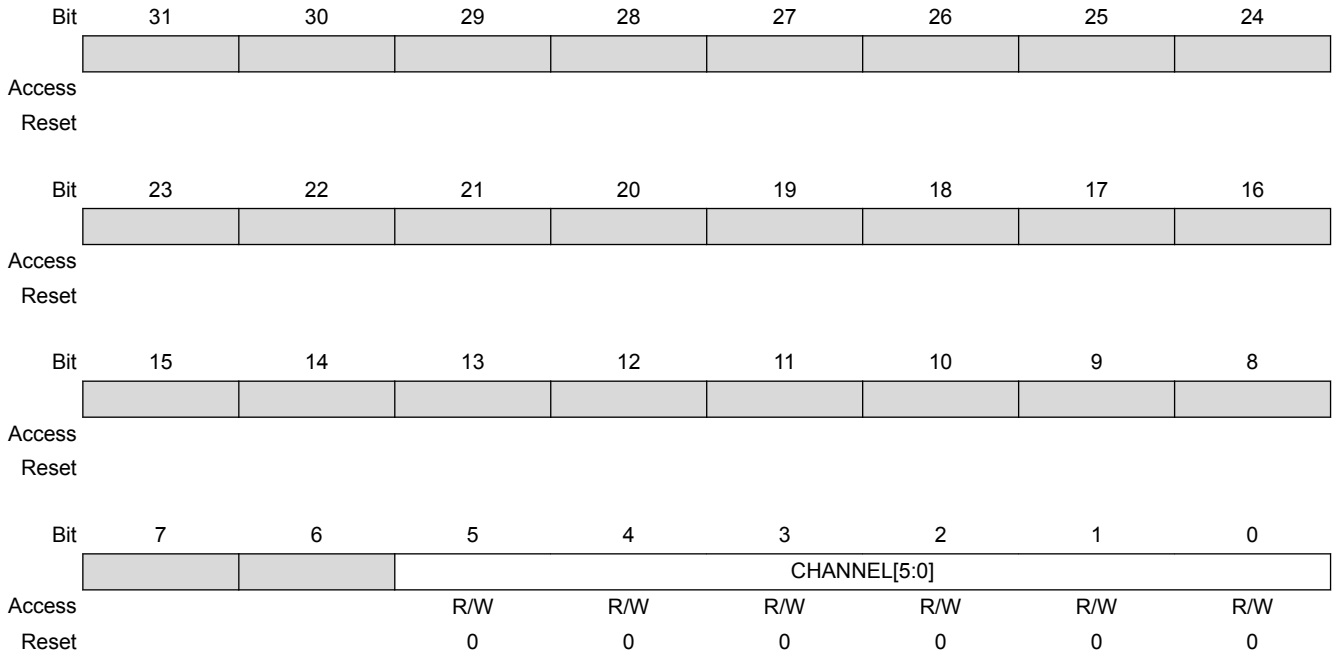
Value	Event Generator	Description
0x0F	EIC EXTINT3	External Interrupt 3
0x10	EIC EXTINT4	External Interrupt 4
0x11	EIC EXTINT5	External Interrupt 5
0x12	EIC EXTINT6	External Interrupt 6
0x13	EIC EXTINT7	External Interrupt 7
0x14	EIC EXTINT8	External Interrupt 8
0x15	EIC EXTINT9	External Interrupt 9
0x16	EIC EXTINT10	External Interrupt 10
0x17	EIC EXTINT11	External Interrupt 11
0x18	EIC EXTINT12	External Interrupt 12
0x19	EIC EXTINT13	External Interrupt 13
0x1A	EIC EXTINT14	External Interrupt 14
0x1B	EIC EXTINT15	External Interrupt 15
0x1C	DMAC CH0	Channel 0
0x1D	DMAC CH1	Channel 1
0x1E	DMAC CH2	Channel 2
0x1F	DMAC CH3	Channel 3
0x20	DMAC CH4	Channel 4
0x21	DMAC CH5	Channel 5
0x22	DMAC CH6	Channel 6
0x23	DMAC CH7	Channel 7
0x24	TCC0 OVF	Overflow
0x25	TCC0 TRG	Trig
0x26	TCC0 CNT	Counter
0x27	TCC0_MCX0	Match/Capture 0
0x28	TCC0_MCX1	Match/Capture 1
0x29	TCC0_MCX2	Match/Capture 2
0x2A	TCC0_MCX3	Match/Capture 3
0x2B	TCC1 OVF	Overflow
0x2C	TCC1 TRG	Trig
0x2D	TCC1 CNT	Counter
0x2E	TCC1_MCX0	Match/Capture 0
0x2F	TCC1_MCX1	Match/Capture 1

Value	Event Generator	Description
0x30	TCC2 OVF	Overflow
0x31	TCC2 TRG	Trig
0x32	TCC2 CNT	Counter
0x33	TCC2_MCX0	Match/Capture 0
0x34	TCC2_MCX1	Match/Capture 1
0x35	TC0 OVF	Overflow/Underflow
0x36	TC0 MC0	Match/Capture 0
0x37	TC0 MC1	Match/Capture 1
0x38	TC1 OVF	Overflow/Underflow
0x39	TC1 MC0	Match/Capture 0
0x3A	TC1 MC1	Match/Capture 1
0x3B	TC2 OVF	Overflow/Underflow
0x3C	TC2 MC0	Match/Capture 0
0x3D	TC2 MC1	Match/Capture 1
0x3E	TC3 OVF	Overflow/Underflow
0x3F	TC3 MC0	Match/Capture 0
0x40	TC3 MC1	Match/Capture 1
0x41	TC4 OVF	Overflow/Underflow
0x42	TC4 MC0	Match/Capture 0
0x43	TC4 MC1	Match/Capture 1
0x44	ADC RESRDY	Result Ready
0x45	ADC WINMON	Window Monitor
0x46	AC COMP0	Comparator 0
0x47	AC COMP1	Comparator 1
0x48	AC WIN0	Window 0
0x49	DAC EMPTY0	Data Buffer Empty
0x4A	EMPTY DAC 1	Data Buffer Empty
0x4B	PTC EOC	End of Conversion
0x4C	PTC WCOMP	Window Comparator
0x4D	TRNG READY	Data Ready
0x4E	CCL LUTOUT0	CCL output
0x4F	CCL LUTOUT1	CCL output
0x50	CCL LUTOUT2	CCL output

Value	Event Generator	Description
0x51	CCL LUTOUT3	CCL output
0x52	PAC ACCERR	Access Error
0x53-0x7F	Reserved	

30.8.8. Event User m

Name: USERm
Offset: 0x80+m*0x4 [m=0..41]
Reset: 0x00000000
Property: PAC Write-Protection



Bits 5:0 – CHANNEL[5:0]: Channel Event Selection
These bits are used to select the channel to connect to the event user.

Note that to select channel m, the value (m+1) must be written to the USER.CHANNEL bit group.

Value	Channel Number
0x00	No channel output selected
0x01	0
0x02	1
0x03	2
0x04	3
0x05	4
0x06	5
0x07	6
0x08	7
0x09	8
0x0A	9

Value	Channel Number
0x0B	10
0x0C	11
0x0D-0xFF	Reserved

Table 30-2. User Multiplexer Number

USERm	User Multiplexer	Description	Path Type
m = 0	PORT EV0	Event 0	Asynchronous, synchronous, and resynchronized paths
m = 1	PORT EV1	Event 1	Asynchronous, synchronous, and resynchronized paths
m = 2	PORT EV2	Event 2	Asynchronous, synchronous, and resynchronized paths
m = 3	PORT EV3	Event 3	Asynchronous, synchronous, and resynchronized paths
m = 4	DMAC CH0	Channel 0	Synchronous, and resynchronized paths
m = 5	DMAC CH1	Channel 1	Synchronous, and resynchronized paths
m = 6	DMAC CH2	Channel 2	Synchronous, and resynchronized paths
m = 7	DMAC CH3	Channel 3	Synchronous, and resynchronized paths
m = 8	DMAC CH4	Channel 4	Synchronous, and resynchronized paths
m = 9	DMAC CH5	Channel 5	Synchronous, and resynchronized paths
m = 10	DMAC CH6	Channel 6	Synchronous, and resynchronized paths
m = 11	DMAC CH7	Channel 7	Synchronous, and resynchronized paths
m = 12	TCC0 EV0	-	Asynchronous, synchronous, and resynchronized paths
m = 13	TCC0 EV1	-	Asynchronous, synchronous, and resynchronized paths

USERm	User Multiplexer	Description	Path Type
m = 14	TCC0 MC0	Match/Capture 0	Asynchronous, synchronous, and resynchronized paths
m = 15	TCC0 MC1	Match/Capture 1	Asynchronous, synchronous, and resynchronized paths
m = 16	TCC0 MC2	Match/Capture 2	Asynchronous, synchronous, and resynchronized paths
m = 17	TCC0 MC3	Match/Capture 3	Asynchronous, synchronous, and resynchronized paths
m = 18	TCC1 EV0	-	Asynchronous, synchronous, and resynchronized paths
m = 19	TCC1 EV1	-	Asynchronous, synchronous, and resynchronized paths
m = 20	TCC1 MC0	Match/Capture 0	Asynchronous, synchronous, and resynchronized paths
m = 21	TCC1 MC1	Match/Capture 1	Asynchronous, synchronous, and resynchronized paths
m = 22	TCC2 EV0	-	Asynchronous, synchronous, and resynchronized paths
m = 23	TCC2 EV1	-	Asynchronous, synchronous, and resynchronized paths
m = 24	TCC2 MC0	Match/Capture 0	Asynchronous, synchronous, and resynchronized paths
m = 25	TCC2 MC1	Match/Capture 1	Asynchronous, synchronous, and resynchronized paths
m = 26	TC0	-	Asynchronous, synchronous, and resynchronized paths
m = 27	TC1	-	Asynchronous, synchronous, and resynchronized paths

USERm	User Multiplexer	Description	Path Type
m = 28	TC2	-	Asynchronous, synchronous, and resynchronized paths
m = 29	TC3	-	Asynchronous, synchronous, and resynchronized paths
m = 30	TC4	-	Asynchronous, synchronous, and resynchronized paths
m = 31	ADC START	ADC start conversion	Asynchronous, synchronous, and resynchronized paths
m = 32	ADC SYNC	Flush ADC	Asynchronous, synchronous, and resynchronized paths
m = 33	AC COMP0	Start comparator 0	Asynchronous, synchronous, and resynchronized paths
m = 34	AC COMP1	Start comparator 1	Asynchronous, synchronous, and resynchronized paths
m = 35	DAC START0	DAC0 start conversion	Asynchronous, synchronous, and resynchronized paths
m = 36	DAC START1	DAC1S start conversion	Asynchronous, synchronous, and resynchronized paths
m = 37	PTC STCONV	PTC start conversion	Asynchronous, synchronous, and resynchronized paths
m = 38	CCL LUTIN 0	CCL input	Asynchronous, synchronous, and resynchronized paths
m = 39	CCL LUTIN 1	CCL input	Asynchronous, synchronous, and resynchronized paths
m = 40	CCL LUTIN 2	CCL input	Asynchronous, synchronous, and resynchronized paths
m = 41	CCL LUTIN 3	CCL input	Asynchronous, synchronous, and resynchronized paths

USERm	User Multiplexer	Description	Path Type
m = 42	Reserved	-	-
m = 43	MTB START	Tracing start	Asynchronous, synchronous, and resynchronized paths
m = 44	MTB STOP	Tracing stop	Asynchronous, synchronous, and resynchronized paths
others	Reserved	-	-

31. SERCOM – Serial Communication Interface

31.1. Overview

There are up to six instances of the serial communication interface (SERCOM) peripheral. Up to five (SERCOM[4:0]) are located in PD1, whereas SERCOM5, present in all device configurations, is always located in power domain PD0.

A SERCOM can be configured to support a number of modes: I^2C, SPI, and USART. When SERCOM is configured and enabled, all SERCOM resources will be dedicated to the selected mode.

The SERCOM serial engine consists of a transmitter and receiver, baud-rate generator and address matching functionality. It can use the internal generic clock or an external clock to operate in all sleep modes.

31.2. Features

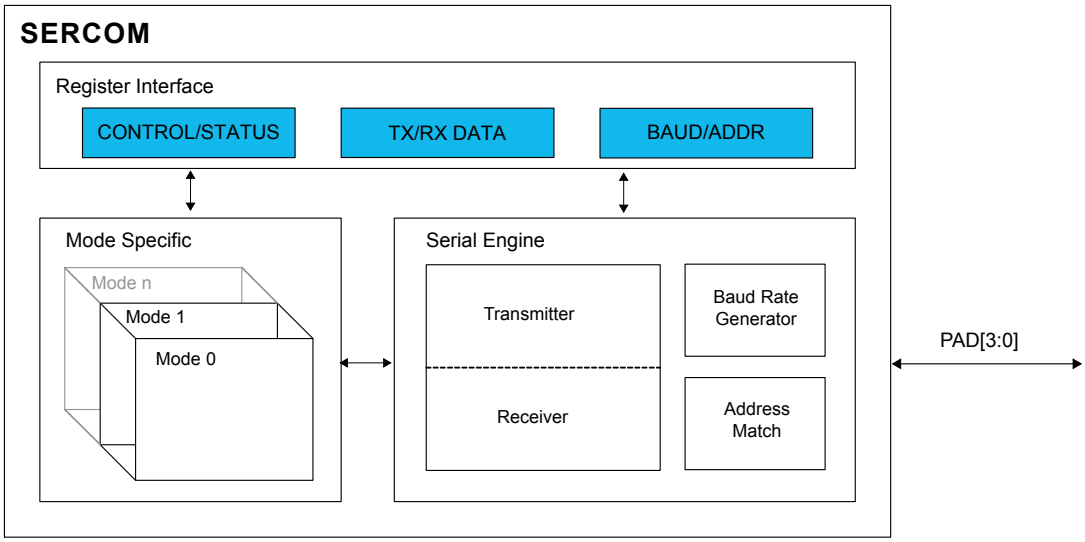
- Interface for configuring into one of the following:
  - I^2C – Two-wire serial interface
    SMBus™ compatible
  - SPI – Serial peripheral interface
  - USART – Universal synchronous and asynchronous serial receiver and transmitter
- Single transmit buffer and double receive buffer
- Baud-rate generator
- Address match/mask logic
- Operational in all sleep modes
- Can be used with DMA

Note: SERCOM5, due to its location in PD0, has a reduced feature set and does *not* support these features:

- General: DMA support
- USART:
  - 3x or 8x oversampling
  - Flow control (RTS/CTS)
  - IrDA
  - Single wire UART according to EN54
  - SOF/EOF function
- I^2C:
  - Fm+ and Hs modes
  - SMBus SCL low timeout
  - 10-bit addressing
  - PMBus Group command support
- SPI:
  - Single master chip select
  - Wake on $\overline{SS}$ assertion

31.3. Block Diagram

Figure 31-1. SERCOM Block Diagram



31.4. Signal Description

See the respective SERCOM mode chapters for details.

Related Links

SERCOM USART – SERCOM Universal Synchronous and Asynchronous Receiver and Transmitter on page 569

SERCOM SPI – SERCOM Serial Peripheral Interface on page 606

SERCOM I2C – SERCOM Inter-Integrated Circuit on page 639

31.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

31.5.1. I/O Lines

Using the SERCOM I/O lines requires the I/O pins to be configured using port configuration (PORT).

From *USART Block Diagram* one can see that the SERCOM has four internal pads, PAD[3:0]. The signals from I2C, SPI and USART are routed through these SERCOM pads via a multiplexer. The configuration of the multiplexer is available from the different SERCOM modes. Refer to the mode specific chapters for details.

Related Links

SERCOM USART – SERCOM Universal Synchronous and Asynchronous Receiver and Transmitter on page 569

SERCOM SPI – SERCOM Serial Peripheral Interface on page 606

SERCOM I2C – SERCOM Inter-Integrated Circuit on page 639

PORT: IO Pin Controller on page 506

Block Diagram on page 570

31.5.2. Power Management

The SERCOM can operate in any sleep mode where the selected clock source is running. SERCOM interrupts can be used to wake up the device from sleep modes.

Related Links

PM – Power Manager on page 186

31.5.3. Clocks

The SERCOM bus clock (CLK_SERCOMx_APB) is enabled by default, and can be enabled and disabled in the Main Clock.

The SERCOM uses two generic clocks: GCLK_SERCOMx_CORE and GCLK_SERCOMx_SLOW. The core clock (GCLK_SERCOMx_CORE) is required to clock the SERCOM while working as a master. The slow clock (GCLK_SERCOMx_SLOW) is only required for certain functions. See specific mode chapters for details.

These clocks must be configured and enabled in the Generic Clock Controller (GCLK) before using the SERCOM.

The generic clocks are asynchronous to the user interface clock (CLK_SERCOMx_APB). Due to this asynchronicity, writing to certain registers will require synchronization between the clock domains. Refer to Synchronization on page 568 for details.

Related Links

GCLK - Generic Clock Controller on page 131

MCLK – Main Clock on page 148

31.5.4. DMA

The DMA request lines are connected to the DMA Controller (DMAC). The DMAC must be configured before the SERCOM DMA requests are used.

Related Links

DMAC – Direct Memory Access Controller on page 399

31.5.5. Interrupts

The interrupt request line is connected to the Interrupt Controller (NVIC). The NVIC must be configured before the SERCOM interrupts are used.

Related Links

Nested Vector Interrupt Controller on page 51

31.5.6. Events

Not applicable.

31.5.7. Debug Operation

When the CPU is halted in debug mode, this peripheral will continue normal operation. If the peripheral is configured to require periodical service by the CPU through interrupts or similar, improper operation or data loss may result during debugging. This peripheral can be forced to halt operation during debugging - refer to the Debug Control (DBGCTRL) register for details.

31.5.8. Register Access Protection

All registers with write-access can be write-protected optionally by the Peripheral Access Controller (PAC), except the following registers:

- Interrupt Flag Clear and Status register (INTFLAG)
- Status register (STATUS)
- Data register (DATA)

- Address register (ADDR)

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

PAC write-protection does not apply to accesses through an external debugger.

Related Links

PAC - Peripheral Access Controller on page 58

31.5.9. Analog Connections

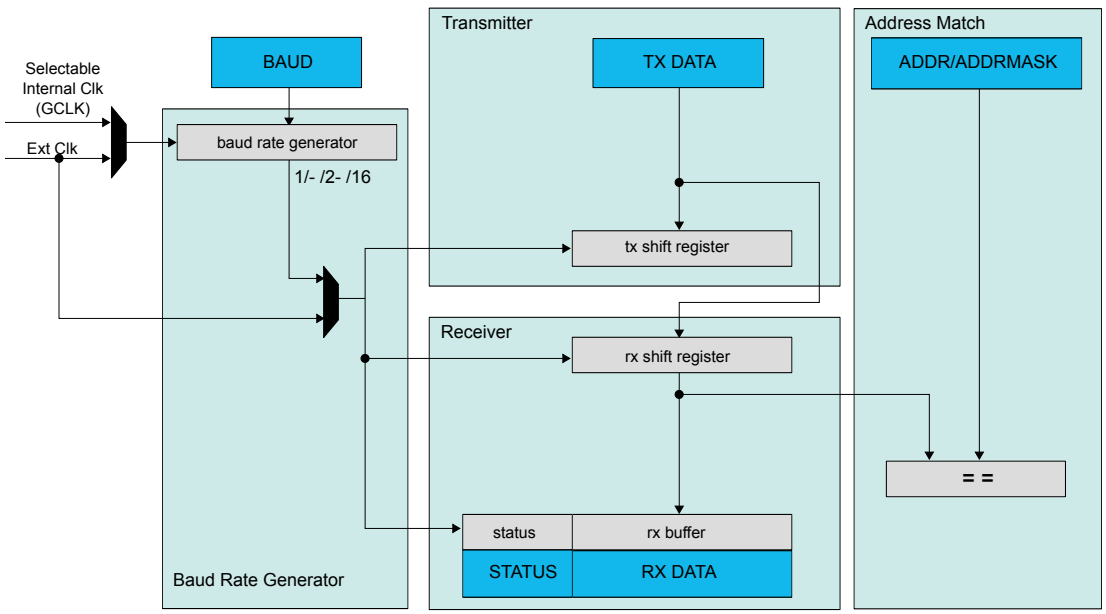
Not applicable.

31.6. Functional Description

31.6.1. Principle of Operation

The basic structure of the SERCOM serial engine is shown in Figure 31-2 SERCOM Serial Engine on page 563. Labels in capital letters are synchronous to the system clock and accessible by the CPU; labels in lowercase letters can be configured to run on the GCLK_SERCOMx_CORE clock or an external clock.

Figure 31-2. SERCOM Serial Engine



The transmitter consists of a single write buffer and a shift register.

The receiver consists of a two-level receive buffer and a shift register.

The baud-rate generator is capable of running on the GCLK_SERCOMx_CORE clock or an external clock.

Address matching logic is included for SPI and I^2C operation.

31.6.2. Basic Operation

31.6.2.1. Initialization

The SERCOM must be configured to the desired mode by writing the Operating Mode bits in the Control A register (CTRLA.MODE). Refer to table SERCOM Modes for details.

Table 31-1. SERCOM Modes

CTRLA.MODE	Description
0x0	USART with external clock
0x1	USART with internal clock
0x2	SPI in slave operation
0x3	SPI in master operation
0x4	I^2C slave operation
0x5	I^2C master operation
0x6-0x7	Reserved

For further initialization information, see the respective SERCOM mode chapters:

Related Links

SERCOM USART – SERCOM Universal Synchronous and Asynchronous Receiver and Transmitter on page 569

SERCOM SPI – SERCOM Serial Peripheral Interface on page 606

SERCOM I2C – SERCOM Inter-Integrated Circuit on page 639

31.6.2.2. Enabling, Disabling, and Resetting

This peripheral is enabled by writing '1' to the Enable bit in the Control A register (CTRLA.ENABLE), and disabled by writing '0' to it.

Writing ‘1’ to the Software Reset bit in the Control A register (CTRLA.SWRST) will reset all registers of this peripheral to their initial states, except the DBGCTRL register, and the peripheral is disabled.

Refer to the CTRLA register description for details.

31.6.2.3. Clock Generation – Baud-Rate Generator

The baud-rate generator, as shown in Figure 31-3 Baud Rate Generator on page 565, generates internal clocks for asynchronous and synchronous communication. The output frequency (f_{BAUD}) is determined by the Baud register (BAUD) setting and the baud reference frequency (f_{ref}). The baud reference clock is the serial engine clock, and it can be internal or external.

For asynchronous communication, the /16 (divide-by-16) output is used when transmitting, whereas the /1 (divide-by-1) output is used while receiving.

For synchronous communication, the /2 (divide-by-2) output is used.

This functionality is automatically configured, depending on the selected operating mode.

Figure 31-3. Baud Rate Generator

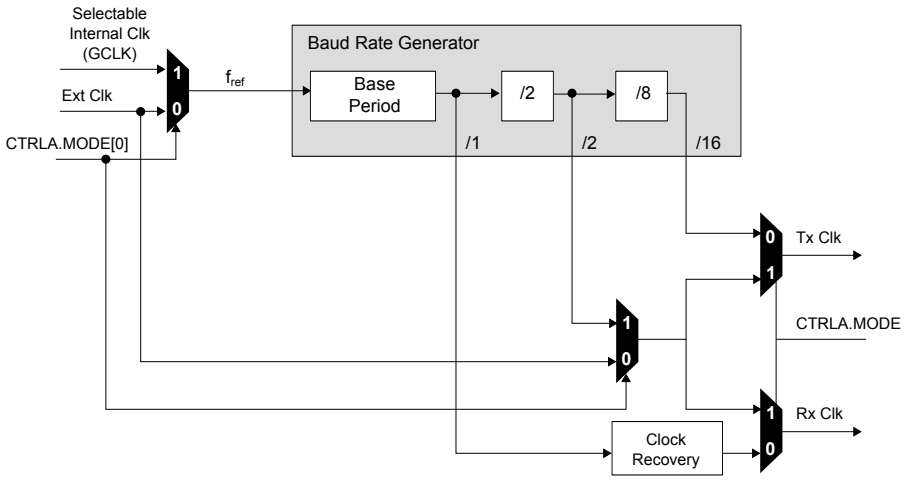


Table 31-2 Baud Rate Equations on page 565 contains equations for the baud rate (in bits per second) and the BAUD register value for each operating mode.

For asynchronous operation, there are two different modes: In *arithmetic mode*, the BAUD register value is 16 bits (0 to 65,535). In *fractional mode*, the BAUD register is 13 bits, while the fractional adjustment is 3 bits. In this mode the BAUD setting must be greater than or equal to 1.

For synchronous operation, the BAUD register value is 8 bits (0 to 255).

Table 31-2. Baud Rate Equations

Operating Mode	Condition	Baud Rate (Bits Per Second)	BAUD Register Value Calculation
Asynchronous Arithmetic	$f_{BAUD} \leq \frac{f_{ref}}{S}$	$f_{BAUD} = \frac{f_{ref}}{S}\left(1 - \frac{BAUD}{65536}\right)$	$BAUD = 65536 \cdot \left(1 - S \cdot \frac{f_{BAUD}}{f_{ref}}\right)$
Asynchronous Fractional	$f_{BAUD} \leq \frac{f_{ref}}{S}$	$f_{BAUD} = \frac{f_{ref}}{S \cdot \left(BAUD + \frac{FP}{8}\right)}$	$BAUD = \frac{f_{ref}}{S \cdot f_{BAUD}} - \frac{FP}{8}$
Synchronous	$f_{BAUD} \leq \frac{f_{ref}}{2}$	$f_{BAUD} = \frac{f_{ref}}{2 \cdot (BAUD + 1)}$	$BAUD = \frac{f_{ref}}{2 \cdot f_{BAUD}} - 1$

S - Number of samples per bit. Can be 16, 8, or 3.

The Asynchronous Fractional option is used for auto-baud detection.

The baud rate error is represented by the following formula:

$$\text{Error} = 1 - \left(\frac{\text{ExpectedBaudRate}}{\text{ActualBaudRate}}\right)$$

Asynchronous Arithmetic Mode BAUD Value Selection

The formula given for f_{BAUD} calculates the average frequency over 65536 f_{ref} cycles. Although the BAUD register can be set to any value between 0 and 65536, the actual average frequency of f_{BAUD} over a single frame is more granular. The BAUD register values that will affect the average frequency over a single frame lead to an integer increase in the cycles per frame (CPF)

$$CPF = \frac{f_{ref}}{f_{BAUD}}(D + S)$$

where

- *D* represent the data bits per frame
- *S* represent the sum of start and first stop bits, if present.

Table 31-3 BAUD Register Value vs. Baud Frequency on page 566 shows the BAUD register value versus baud frequency f_{BAUD} at a serial engine frequency of 48MHz. This assumes a *D* value of 8 bits and an *S* value of 2 bits (10 bits, including start and stop bits).

Table 31-3. BAUD Register Value vs. Baud Frequency

BAUD Register Value	Serial Engine CPF	f_{BAUD} at 48MHz Serial Engine Frequency (f_{REF})
0 – 406	160	3MHz
407 – 808	161	2.981MHz
809 – 1205	162	2.963MHz
...	...	...
65206	31775	15.11kHz
65207	31871	15.06kHz
65208	31969	15.01kHz

31.6.3. Additional Features

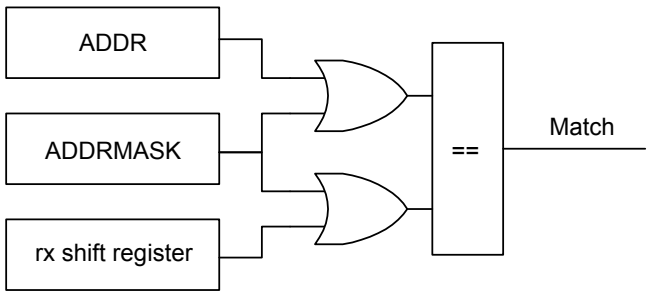
31.6.3.1. Address Match and Mask

The SERCOM address match and mask feature is capable of matching either one address, two unique addresses, or a range of addresses with a mask, based on the mode selected. The match uses seven or eight bits, depending on the mode.

Address With Mask

An address written to the Address bits in the Address register (ADDR.ADDR), and a mask written to the Address Mask bits in the Address register (ADDR.ADDRMASK) will yield an address match. All bits that are masked are not included in the match. Note that writing the ADDR.ADDRMASK to 'all zeros' will match a single unique address, while writing ADDR.ADDRMASK to 'all ones' will result in all addresses being accepted.

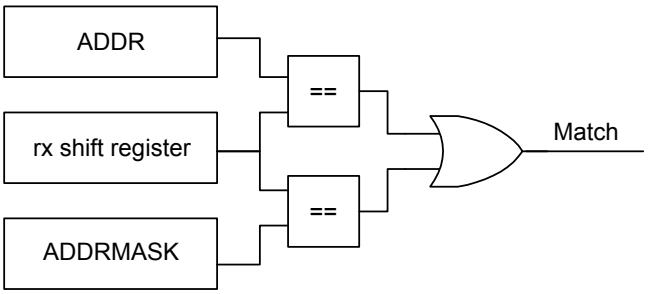
Figure 31-4. Address With Mask



Two Unique Addresses

The two addresses written to ADDR and ADDRMASK will cause a match.

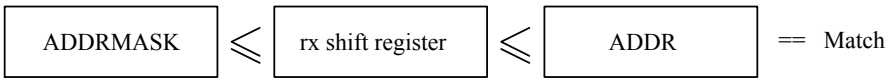
Figure 31-5. Two Unique Addresses



Address Range

The range of addresses between and including ADDR.ADDR and ADDR.ADDRMASK will cause a match. ADDR.ADDR and ADDR.ADDRMASK can be set to any two addresses, with ADDR.ADDR acting as the upper limit and ADDR.ADDRMASK acting as the lower limit.

Figure 31-6. Address Range



31.6.4. DMA Operation

Not applicable.

31.6.5. Interrupts

Interrupt sources are mode-specific. See the respective SERCOM mode chapters for details.

Each interrupt source has its own interrupt flag.

The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) will be set when the interrupt condition is met.

Each interrupt can be individually enabled by writing '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR).

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until either the interrupt flag is cleared, the interrupt is disabled, or the SERCOM is reset. For details on clearing interrupt flags, refer to the INTFLAG register description.

The SERCOM has one common interrupt request line for all the interrupt sources. The value of INTFLAG indicates which interrupt condition occurred. The user must read the INTFLAG register to determine which interrupt condition is present.

Note:

Note that interrupts must be globally enabled for interrupt requests.

Related Links

Nested Vector Interrupt Controller on page 51

31.6.6. Events

Not applicable.

31.6.7. Sleep Mode Operation

The peripheral can operate in any sleep mode where the selected serial clock is running. This clock can be external or generated by the internal baud-rate generator.

The SERCOM interrupts can be used to wake up the device from sleep modes. Refer to the different SERCOM mode chapters for details.

31.6.8. Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read. An exception is the Channel Enable bit in the Peripheral Channel Control registers (PCHCTRLm.CHEN). When changing this bit, the bit value must be read-back to ensure the synchronization is complete and to assert glitch free internal operation. Note that changing the bit value under ongoing synchronization will *not* generate an error.

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

Required read-synchronization is denoted by the "Read-Synchronized" property in the register description.

Related Links

Register Synchronization on page 127

32. SERCOM USART – SERCOM Universal Synchronous and Asynchronous Receiver and Transmitter

32.1. Overview

The Universal Synchronous and Asynchronous Receiver and Transmitter (USART) is one of the available modes in the Serial Communication Interface (SERCOM).

The USART uses the SERCOM transmitter and receiver, see Block Diagram on page 570. Labels in uppercase letters are synchronous to CLK_SERCOMx_APB and accessible for CPU. Labels in lowercase letters can be programmed to run on the internal generic clock or an external clock.

The transmitter consists of a single write buffer, a shift register, and control logic for different frame formats. The write buffer support data transmission without any delay between frames. The receiver consists of a two-level receive buffer and a shift register. Status information of the received data is available for error checking. Data and clock recovery units ensure robust synchronization and noise filtering during asynchronous data reception.

Related Links

SERCOM – Serial Communication Interface on page 560

32.2. USART Features

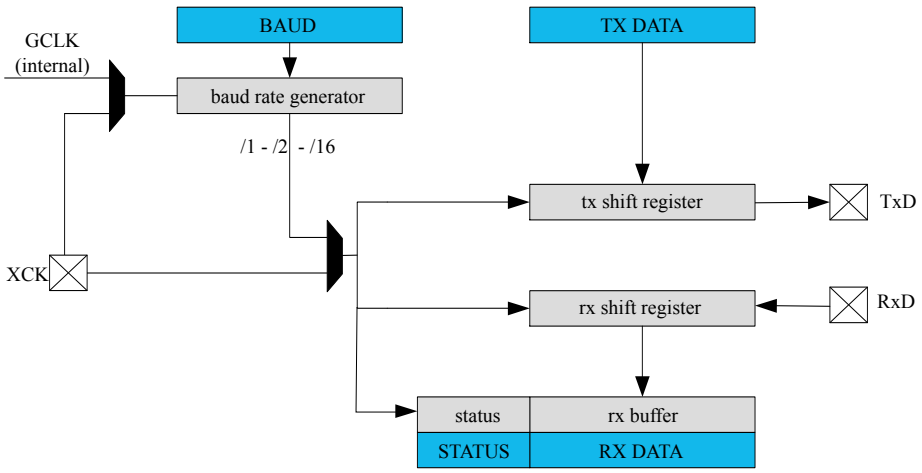
- Full-duplex operation
- Asynchronous (with clock reconstruction) or synchronous operation
- Internal or external clock source for asynchronous and synchronous operation
- Baud-rate generator
- Supports serial frames with 5, 6, 7, 8 or 9 data bits and 1 or 2 stop bits
- Odd or even parity generation and parity check
- Selectable LSB- or MSB-first data transfer
- Buffer overflow and frame error detection
- Noise filtering, including false start-bit detection and digital low-pass filter
- Collision detection
- Can operate in all sleep modes
- Operation at speeds up to half the system clock for internally generated clocks
- Operation at speeds up to the system clock for externally generated clocks
- RTS and CTS flow control
- IrDA modulation and demodulation up to 115.2kbps
- Start-of-frame detection
- Can work with DMA

Related Links

Features on page 560

32.3. Block Diagram

Figure 32-1. USART Block Diagram



32.4. Signal Description

Table 32-1. SERCOM USART Signals

Signal Name	Type	Description
PAD[3:0]	Digital I/O	General SERCOM pins

One signal can be mapped to one of several pins.

Related Links

I/O Multiplexing and Considerations on page 29

32.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

32.5.1. I/O Lines

Using the USART's I/O lines requires the I/O pins to be configured using the I/O Pin Controller (PORT).

When the SERCOM is used in USART mode, the SERCOM controls the direction and value of the I/O pins according to the table below. Both PORT control bits PINCFGn.PULLEN and PINCFGn.DRVSTR are still effective. If the receiver or transmitter is disabled, these pins can be used for other purposes.

Table 32-2. USART Pin Configuration

Pin	Pin Configuration
TxD	Output
RxD	Input
XCK	Output or input

The combined configuration of PORT and the Transmit Data Pinout and Receive Data Pinout bit fields in the Control A register (CTRLA.TXPO and CTRLA.RXPO, respectively) will define the physical position of the USART signals in Table 32-2 USART Pin Configuration on page 570.

Related Links

PORT: IO Pin Controller on page 506

32.5.2. Power Management

This peripheral can continue to operate in any sleep mode where its source clock is running. The interrupts can wake up the device from sleep modes. Refer to *PM – Power Manager* for details on the different sleep modes.

Related Links

PM – Power Manager on page 186

32.5.3. Clocks

The SERCOM bus clock (CLK_SERCOMx_APB) is enabled by default, and can be disabled and enabled in the Main Clock Controller. Refer to *Peripheral Clock Masking* for details.

A generic clock (GCLK_SERCOMx_CORE) is required to clock the SERCOMx_CORE. This clock must be configured and enabled in the Generic Clock Controller before using the SERCOMx_CORE. Refer to *GCLK - Generic Clock Controller* for details.

This generic clock is asynchronous to the bus clock (CLK_SERCOMx_APB). Therefore, writing to certain registers will require synchronization to the clock domains. Refer to Synchronization on page 582for further details.

Related Links

Peripheral Clock Masking on page 151

GCLK - Generic Clock Controller on page 131

32.5.4. DMA

The DMA request lines are connected to the DMA Controller (DMAC). In order to use DMA requests with this peripheral the DMAC must be configured first. Refer to *DMAC – Direct Memory Access Controller* for details.

Related Links

DMAC – Direct Memory Access Controller on page 399

32.5.5. Interrupts

The interrupt request line is connected to the Interrupt Controller. In order to use interrupt requests of this peripheral, the Interrupt Controller (NVIC) must be configured first. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

Nested Vector Interrupt Controller on page 51

32.5.6. Events

Not applicable.

32.5.7. Debug Operation

When the CPU is halted in debug mode, this peripheral will continue normal operation. If the peripheral is configured to require periodical service by the CPU through interrupts or similar, improper operation or data loss may result during debugging. This peripheral can be forced to halt operation during debugging - refer to the Debug Control (DBGCTRL) register for details.

32.5.8. Register Access Protection

Registers with write-access can be write-protected optionally by the peripheral access controller (PAC).

PAC Write-Protection is not available for the following registers:

- Interrupt Flag Clear and Status register (INTFLAG)
- Status register (STATUS)
- Data register (DATA)

Optional PAC Write-Protection is denoted by the "PAC Write-Protection" property in each individual register description.

Write-protection does not apply to accesses through an external debugger.

Related Links

PAC - Peripheral Access Controller on page 58

32.5.9. Analog Connections

Not applicable.

32.6. Functional Description

32.6.1. Principle of Operation

The USART uses the following lines for data transfer:

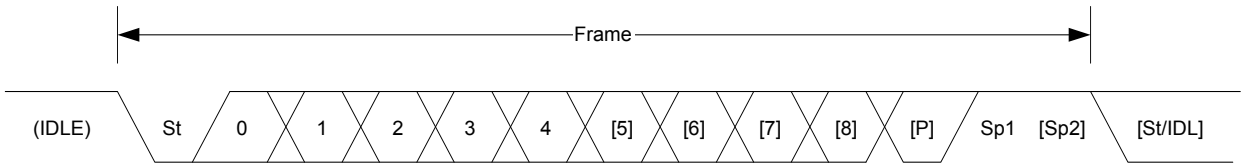
- RxD for receiving
- TxD for transmitting
- XCK for the transmission clock in synchronous operation

USART data transfer is frame based. A serial frame consists of:

- 1 start bit
- From 5 to 9 data bits (MSB or LSB first)
- No, even or odd parity bit
- 1 or 2 stop bits

A frame starts with the start bit followed by one character of data bits. If enabled, the parity bit is inserted after the data bits and before the first stop bit. After the stop bit(s) of a frame, either the next frame can follow immediately, or the communication line can return to the idle (high) state. The figure below illustrates the possible frame formats. Brackets denote optional bits.

Figure 32-2. Frame Formats



St Start bit. Signal is always low.

n, [n] Data bits. 0 to [5..9]

[P] Parity bit. Either odd or even.

Sp, [Sp] Stop bit. Signal is always high.

IDLE No frame is transferred on the communication line. Signal is always high in this state.

32.6.2. Basic Operation

32.6.2.1. Initialization

The following registers are enable-protected, meaning they can only be written when the USART is disabled (CTRL.ENABLE=0):

- Control A register (CTRLA), except the Enable (ENABLE) and Software Reset (SWRST) bits.
- Control B register (CTRLB), except the Receiver Enable (RXEN) and Transmitter Enable (TXEN) bits.
- Baud register (BAUD)

Any writes to these registers when the USART is enabled or is being enabled (CTRL.ENABLE is one) will be discarded. Writes to these registers while the peripheral is being disabled, will be completed after the disabling is complete.

When the USART is enabled or is being enabled (CTRLA.ENABLE=1), any writing attempt to these registers will be discarded. If the peripheral is being disabled, writing to these registers will be executed after disabling is completed. Enable-protection is denoted by the "Enable-Protection" property in the register description.

Before the USART is enabled, it must be configured by these steps:

1. Select either external (0x0) or internal clock (0x1) by writing the Operating Mode value in the CTRLA register (CTRLA.MODE).
2. Select either asynchronous (0) or or synchronous (1) communication mode by writing the Communication Mode bit in the CTRLA register (CTRLA.CMODE).
3. Select pin for receive data by writing the Receive Data Pinout value in the CTRLA register (CTRLA.RXPO).
4. Select pads for the transmitter and external clock by writing the Transmit Data Pinout bit in the CTRLA register (CTRLA.TXPO).
5. Configure the Character Size field in the CTRLB register (CTRLB.CHSIZE) for character size.
6. Set the Data Order bit in the CTRLA register (CTRLA.DORD) to determine MSB- or LSB-first data transmission.
7. To use parity mode:
   7.1. Enable parity mode by writing 0x1 to the Frame Format field in the CTRLA register (CTRLA.FORM).
   7.2. Configure the Parity Mode bit in the CTRLB register (CTRLB.PMODE) for even or odd parity.
8. Configure the number of stop bits in the Stop Bit Mode bit in the CTRLB register (CTRLB.SBMODE).
9. When using an internal clock, write the Baud register (BAUD) to generate the desired baud rate.
10. Enable the transmitter and receiver by writing '1' to the Receiver Enable and Transmitter Enable bits in the CTRLB register (CTRLB.RXEN and CTRLB.TXEN).

32.6.2.2. Enabling, Disabling, and Resetting

This peripheral is enabled by writing '1' to the Enable bit in the Control A register (CTRLA.ENABLE), and disabled by writing '0' to it.

Writing ‘1’ to the Software Reset bit in the Control A register (CTRLA.SWRST) will reset all registers of this peripheral to their initial states, except the DBGCTRL register, and the peripheral is disabled.

Refer to the CTRLA register description for details.

32.6.2.3. Clock Generation and Selection

For both synchronous and asynchronous modes, the clock used for shifting and sampling data can be generated internally by the SERCOM baud-rate generator or supplied externally through the XCK line.

The synchronous mode is selected by writing a '1' to the Communication Mode bit in the Control A register (CTRLA.CMODE), the asynchronous mode is selected by writing a zero to CTRLA.CMODE.

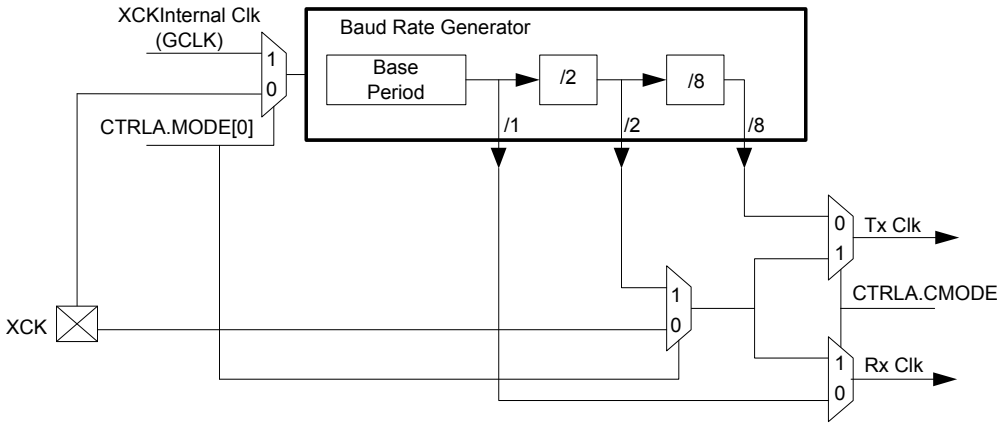
The internal clock source is selected by writing 0x1 to the Operation Mode bit field in the Control A register (CTRLA.MODE), the external clock source is selected by writing 0x0 to CTRLA.MODE.

The SERCOM baud-rate generator is configured as in the figure below.

In asynchronous mode (CTRLA.CMODE=0), the 16-bit Baud register value is used.

In synchronous mode (CTRLA.CMODE=1), the eight LSBs of the Baud register are used. Refer to *Clock Generation – Baud-Rate Generator* for details on configuring the baud rate.

Figure 32-3. Clock Generation



Related Links

Clock Generation – Baud-Rate Generator on page 564

Asynchronous Arithmetic Mode BAUD Value Selection on page 565

Synchronous Clock Operation

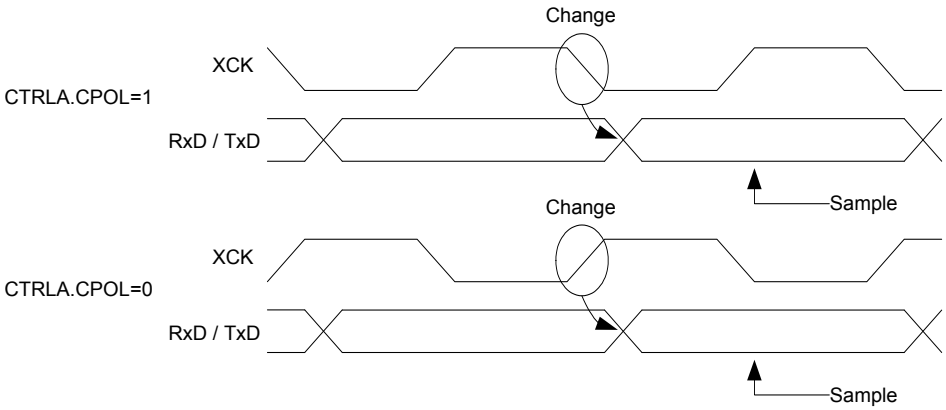
In synchronous mode, the CTRLA.MODE bit field determines whether the transmission clock line (XCK) serves either as input or output. The dependency between clock edges, data sampling, and data change is the same for internal and external clocks. Data input on the RxD pin is sampled at the opposite XCK clock edge when data is driven on the TxD pin.

The Clock Polarity bit in the Control A register (CTRLA.CPOL) selects which XCK clock edge is used for RxD sampling, and which is used for TxD change:

When CTRLA.CPOL is '0', the data will be changed on the rising edge of XCK, and sampled on the falling edge of XCK.

When CTRLA.CPOL is '1', the data will be changed on the falling edge of XCK, and sampled on the rising edge of XCK.

Figure 32-4. Synchronous Mode XCK Timing



When the clock is provided through XCK (CTRLA.MODE=0x0), the shift registers operate directly on the XCK clock. This means that XCK is not synchronized with the system clock and, therefore, can operate at frequencies up to the system frequency.

32.6.2.4. Data Register

The USART Transmit Data register (TxDATA) and USART Receive Data register (RxDATA) share the same I/O address, referred to as the Data register (DATA). Writing the DATA register will update the TxDATA register. Reading the DATA register will return the contents of the RxDATA register.

32.6.2.5. Data Transmission

Data transmission is initiated by writing the data to be sent into the DATA register. Then, the data in TxDATA will be moved to the shift register when the shift register is empty and ready to send a new frame. After the shift register is loaded with data, the data frame will be transmitted.

When the entire data frame including stop bit(s) has been transmitted and no new data was written to DATA, the Transmit Complete interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.TXC) will be set, and the optional interrupt will be generated.

The Data Register Empty flag in the Interrupt Flag Status and Clear register (INTFLAG.DRE) indicates that the register is empty and ready for new data. The DATA register should only be written to when INTFLAG.DRE is set.

Disabling the Transmitter

The transmitter is disabled by writing '0' to the Transmitter Enable bit in the CTRLB register (CTRLB.TXEN).

Disabling the transmitter will complete only after any ongoing and pending transmissions are completed, i.e., there is no data in the transmit shift register and TxDATA to transmit.

32.6.2.6. Data Reception

The receiver accepts data when a valid start bit is detected. Each bit following the start bit will be sampled according to the baud rate or XCK clock, and shifted into the receive shift register until the first stop bit of a frame is received. The second stop bit will be ignored by the receiver.

When the first stop bit is received and a complete serial frame is present in the receive shift register, the contents of the shift register will be moved into the two-level receive buffer. Then, the Receive Complete interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.RXC) will be set, and the optional interrupt will be generated.

The received data can be read from the DATA register when the Receive Complete interrupt flag is set.

Disabling the Receiver

Writing '0' to the Receiver Enable bit in the CTRLB register (CTRLB.RXEN) will disable the receiver, flush the two-level receive buffer, and data from ongoing receptions will be lost.

Error Bits

The USART receiver has three error bits in the Status (STATUS) register: Frame Error (FERR), Buffer Overflow (BUFOVF), and Parity Error (PERR). Once an error happens, the corresponding error bit will be set until it is cleared by writing ‘1’ to it. These bits are also cleared automatically when the receiver is disabled.

There are two methods for buffer overflow notification, selected by the Immediate Buffer Overflow Notification bit in the Control A register (CTRLA.IBON):

When CTRLA.IBON=1, STATUS.BUFOVF is raised immediately upon buffer overflow. Software can then empty the receive FIFO by reading RxDATA, until the receiver complete interrupt flag (INTFLAG.RXC) is cleared.

When CTRLA.IBON=0, the buffer overflow condition is attending data through the receive FIFO. After the received data is read, STATUS.BUFOVF will be set along with INTFLAG.RXC.

Asynchronous Data Reception

The USART includes a clock recovery and data recovery unit for handling asynchronous data reception.

The clock recovery logic can synchronize the incoming asynchronous serial frames at the RxD pin to the internally generated baud-rate clock.

The data recovery logic samples and applies a low-pass filter to each incoming bit, thereby improving the noise immunity of the receiver.

Asynchronous Operational Range

The operational range of the asynchronous reception depends on the accuracy of the internal baud-rate clock, the rate of the incoming frames, and the frame size (in number of bits). In addition, the operational range of the receiver is depending on the difference between the received bit rate and the internally generated baud rate. If the baud rate of an external transmitter is too high or too low compared to the internally generated baud rate, the receiver will not be able to synchronize the frames to the start bit.

There are two possible sources for a mismatch in baud rate: First, the reference clock will always have some minor instability. Second, the baud-rate generator cannot always do an exact division of the reference clock frequency to get the baud rate desired. In this case, the BAUD register value should be set to give the lowest possible error. Refer to *Clock Generation – Baud-Rate Generator* for details.

Recommended maximum receiver baud-rate errors for various character sizes are shown in the table below.

Table 32-3. Asynchronous Receiver Error for 16-fold Oversampling

D (Data bits+Parity)	R_{SLOW} [%]	R_{FAST} [%]	Max. total error [%]	Recommended max. Rx error [%]
5	94.12	107.69	+5.88/-7.69	±2.5
6	94.92	106.67	+5.08/-6.67	±2.0
7	95.52	105.88	+4.48/-5.88	±2.0
8	96.00	105.26	+4.00/-5.26	±2.0
9	96.39	104.76	+3.61/-4.76	±1.5
10	96.70	104.35	+3.30/-4.35	±1.5

The recommended maximum receiver baud-rate error assumes that the receiver and transmitter equally divide the maximum total error.

The following equations can calculate the ratio of the incoming data rate and internal receiver baud rate:

$$R_{SLOW} = \frac{(D+1)S}{S-1+D\cdot S+S_F} \quad , \qquad R_{FAST} = \frac{(D+2)S}{(D+1)S+S_M}$$

- R_{SLOW} is the ratio of the slowest incoming data rate that can be accepted in relation to the receiver baud rate
- R_{FAST} is the ratio of the fastest incoming data rate that can be accepted in relation to the receiver baud rate
- D is the sum of character size and parity size (D = 5 to 10 bits)
- S is the number of samples per bit (S = 16, 8 or 3)
- S_F is the first sample number used for majority voting (S_F = 7, 3, or 2) when CTRLA.SAMPA=0.
- S_M is the middle sample number used for majority voting (S_M = 8, 4, or 2) when CTRLA.SAMPA=0.

Related Links

Clock Generation – Baud-Rate Generator on page 564
Asynchronous Arithmetic Mode BAUD Value Selection on page 565

32.6.3. Additional Features

32.6.3.1. Parity

Even or odd parity can be selected for error checking by writing 0x1 to the Frame Format bit field in the Control A register (CTRLA.FORM).

If *even parity* is selected (CTRLB.PMODE=0), the parity bit of an outgoing frame is '1' if the data contains an odd number of bits that are '1', making the total number of '1' even.

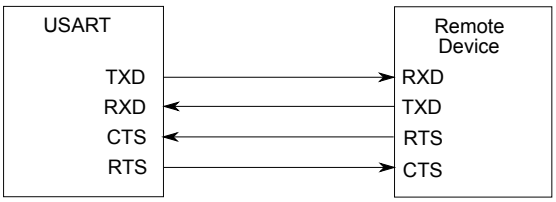
If *odd parity* is selected (CTRLB.PMODE=1), the parity bit of an outgoing frame is '1' if the data contains an even number of bits that are '0', making the total number of '1' odd.

When parity checking is enabled, the parity checker calculates the parity of the data bits in incoming frames and compares the result with the parity bit of the corresponding frame. If a parity error is detected, the Parity Error bit in the Status register (STATUS.PERR) is set.

32.6.3.2. Hardware Handshaking

The USART features an out-of-band hardware handshaking flow control mechanism, implemented by connecting the RTS and CTS pins with the remote device, as shown in the figure below.

Figure 32-5. Connection with a Remote Device for Hardware Handshaking

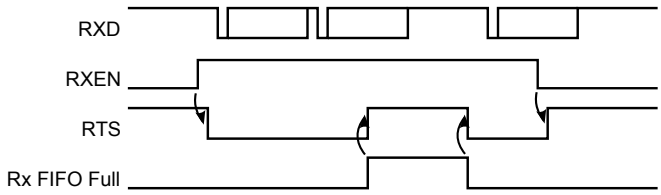


Hardware handshaking is only available in the following configuration:

- USART with internal clock (CTRLA.MODE=1),
- Asynchronous mode (CTRLA.CMODE=0),
- and Flow control pinout (CTRLA.TXPO=2).

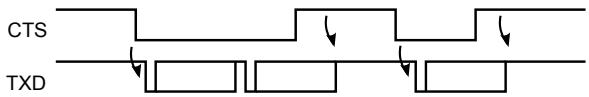
When the receiver is disabled or the receive FIFO is full, the receiver will drive the RTS pin high. This notifies the remote device to stop transfer after the ongoing transmission. Enabling and disabling the receiver by writing to CTRLB.RXEN will set/clear the RTS pin after a synchronization delay. When the receive FIFO goes full, RTS will be set immediately and the frame being received will be stored in the shift register until the receive FIFO is no longer full.

Figure 32-6. Receiver Behavior when Operating with Hardware Handshaking



The current CTS Status is in the STATUS register (STATUS.CTS). Character transmission will start only if STATUS.CTS=0. When CTS is set, the transmitter will complete the ongoing transmission and stop transmitting.

Figure 32-7. Transmitter Behavior when Operating with Hardware Handshaking



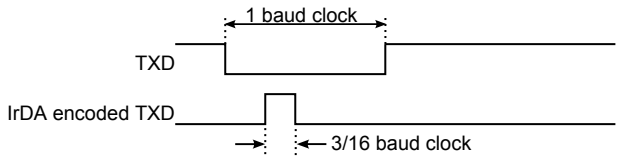
32.6.3.3. IrDA Modulation and Demodulation

Transmission and reception can be encoded IrDA compliant up to 115.2 kb/s. IrDA modulation and demodulation work in the following configuration:

- IrDA encoding enabled (CTRLB.ENC=1),
- Asynchronous mode (CTRLA.CMODE=0),
- and 16x sample rate (CTRLA.SAMPR[0]=0).

During transmission, each low bit is transmitted as a high pulse. The pulse width is 3/16 of the baud rate period, as illustrated in the figure below.

Figure 32-8. IrDA Transmit Encoding



The reception decoder has two main functions.

The first is to synchronize the incoming data to the IrDA baud rate counter. Synchronization is performed at the start of each zero pulse.

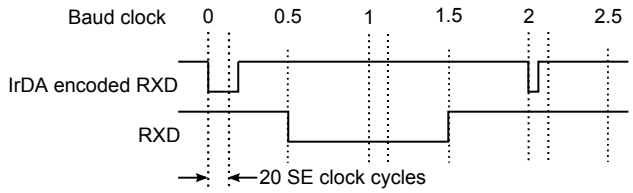
The second main function is to decode incoming Rx data. If a pulse width meets the minimum length set by configuration (RXPL.RXPL), it is accepted. When the baud rate counter reaches its middle value (1/2 bit length), it is transferred to the receiver.

Note: Note that the polarity of the transmitter and receiver are opposite: During transmission, a '0' bit is transmitted as a '1' pulse. During reception, an accepted '0' pulse is received as a '0' bit.

Example: The figure below illustrates reception where RXPL.RXPL is set to 19. This indicates that the pulse width should be at least 20 SE clock cycles. When using BAUD=0xE666 or 160 SE cycles per bit, this corresponds to 2/16 baud clock as minimum pulse width required. In this case the first bit is accepted as a '0', the second bit is a '1',

and the third bit is also a '1'. A low pulse is rejected since it does not meet the minimum requirement of 2/16 baud clock.

Figure 32-9. IrDA Receive Decoding



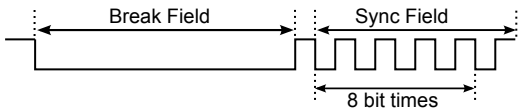
32.6.3.4. Break Character Detection and Auto-Baud

Break character detection and auto-baud are available in this configuration:

- Auto-baud frame format (CTRLA.FORM = 0x04 or 0x05),
- Asynchronous mode (CTRLA.CMODE = 0),
- and 16x sample rate using fractional baud rate generation (CTRLA.SAMPR = 1).

The auto-baud follows the LIN format. All LIN Frames start with a Break Field followed by a Sync Field. The USART uses a break detection threshold of greater than 11 nominal bit times at the configured baud rate. At any time, if more than 11 consecutive dominant bits are detected on the bus, the USART detects a Break Field. When a Break Field has been detected, the Receive Break interrupt flag (INTFLAG.RXBRK) is set and the USART expects the Sync Field character to be 0x55. This field is used to update the actual baud rate in order to stay synchronized. If the received Sync character is not 0x55, then the Inconsistent Sync Field error flag (STATUS.ISF) is set along with the Error interrupt flag (INTFLAG.ERROR), and the baud rate is unchanged.

Figure 32-10. LIN Break and Sync Fields



After a break field is detected and the start bit of the Sync Field is detected, a counter is started. The counter is then incremented for the next 8 bit times of the Sync Field. At the end of these 8 bit times, the counter is stopped. At this moment, the 13 most significant bits of the counter (value divided by 8) give the new clock divider (BAUD.BAUD), and the 3 least significant bits of this value (the remainder) give the new Fractional Part (BAUD.FP).

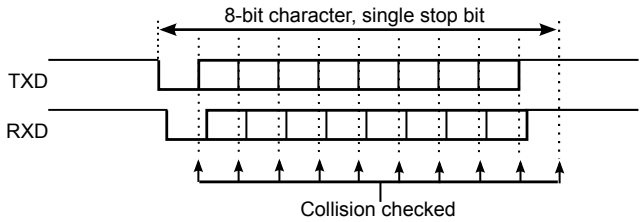
When the Sync Field has been received, the clock divider (BAUD.BAUD) and the Fractional Part (BAUD.FP) are updated after a synchronization delay. After the Break and Sync Fields are received, multiple characters of data can be received.

32.6.3.5. Collision Detection

When the receiver and transmitter are connected either through pin configuration or externally, transmit collision can be detected after selecting the Collision Detection Enable bit in the CTRLB register (CTRLB.COLDEN=1). To detect collision, the receiver and transmitter must be enabled (CTRLB.RXEN=1 and CTRLB.TXEN=1).

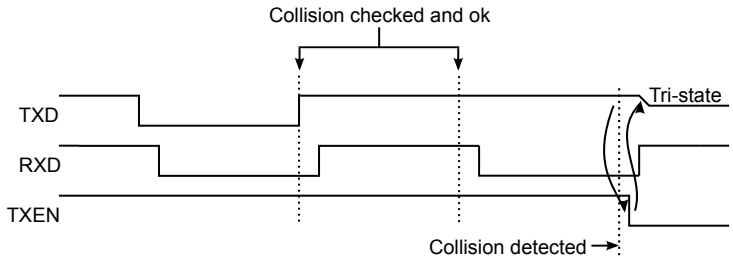
Collision detection is performed for each bit transmitted by comparing the received value with the transmit value, as shown in the figure below. While the transmitter is idle (no transmission in progress), characters can be received on RxD without triggering a collision.

Figure 32-11. Collision Checking



The next figure shows the conditions for a collision detection. In this case, the start bit and the first data bit are received with the same value as transmitted. The second received data bit is found to be different than the transmitted bit at the detection point, which indicates a collision.

Figure 32-12. Collision Detected



When a collision is detected, the USART follows this sequence:

1. Abort the current transfer.
2. Flush the transmit buffer.
3. Disable transmitter (CTRLB.TXEN=0)
   - This is done after a synchronization delay. The CTRLB Synchronization Busy bit (SYNCBUSY.CTRLB) will be set until this is complete.
   - After disabling, the TxD pin will be tri-stated.
4. Set the Collision Detected bit (STATUS.COLL) along with the Error interrupt flag (INTFLAG.ERROR).
5. Set the Transmit Complete interrupt flag (INTFLAG.TXC), since the transmit buffer no longer contains data.

After a collision, software must manually enable the transmitter again before continuing, after assuring that the CTRLB Synchronization Busy bit (SYNCBUSY.CTRLB) is not set.

32.6.3.6. Loop-Back Mode

For loop-back mode, configure the Receive Data Pinout (CTRLA.RXPO) and Transmit Data Pinout (CTRLA.TXPO) to use the same data pins for transmit and receive. The loop-back is through the pad, so the signal is also available externally.

32.6.3.7. Sample Adjustment

In asynchronous mode (CTRLA.CMODE=0), three samples in the middle are used to determine the value based on majority voting. The three samples used for voting can be selected using the Sample Adjustment bit field in Control A register (CTRLA.SAMPA). When CTRLA.SAMPA=0, samples 7-8-9 are used for 16x oversampling, and samples 3-4-5 are used for 8x oversampling.

32.6.4. DMA, Interrupts and Events

Table 32-4. Module Request for SERCOM USART

Condition	Request		
	DMA	Interrupt	Event
Data Register Empty (DRE)	Yes (request cleared when data is written)	Yes	NA
Receive Complete (RXC)	Yes (request cleared when data is read)	Yes	
Transmit Complete (TXC)	NA	Yes	
Receive Start (RXS)	NA	Yes	
Clear to Send Input Change (CTSIC)	NA	Yes	
Receive Break (RXBRK)	NA	Yes	
Error (ERROR)	NA	Yes	

32.6.4.1. DMA Operation

The USART generates the following DMA requests:

- Data received (RX): The request is set when data is available in the receive FIFO. The request is cleared when DATA is read.
- Data transmit (TX): The request is set when the transmit buffer (TX DATA) is empty. The request is cleared when DATA is written.

32.6.4.2. Interrupts

The USART has the following interrupt sources. These are asynchronous interrupts, and can wake up the device from any sleep mode:

- Data Register Empty (DRE)
- Receive Complete (RXC)
- Transmit Complete (TXC)
- Receive Start (RXS)
- Clear to Send Input Change (CTSIC)
- Received Break (RXBRK)
- Error (ERROR)

Each interrupt source has its own interrupt flag. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) will be set when the interrupt condition is met. Each interrupt can be individually enabled by writing '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR).

An interrupt request is generated when the interrupt flag is set and if the corresponding interrupt is enabled. The interrupt request remains active until either the interrupt flag is cleared, the interrupt is disabled, or the USART is reset. For details on clearing interrupt flags, refer to the INTFLAG register description.

The USART has one common interrupt request line for all the interrupt sources. The value of INTFLAG indicates which interrupt is executed. Note that interrupts must be globally enabled for interrupt requests. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

Nested Vector Interrupt Controller on page 51

32.6.4.3. Events

Not applicable.

32.6.5. Sleep Mode Operation

The behavior in sleep mode is depending on the clock source and the Run In Standby bit in the Control A register (CTRLA.RUNSTDBY):

- Internal clocking, CTRLA.RUNSTDBY=1: GCLK_SERCOMx_CORE can be enabled in all sleep modes. Any interrupt can wake up the device.
- External clocking, CTRLA.RUNSTDBY=1: The Receive Start and the Receive Complete interrupt(s) can wake up the device.
- Internal clocking, CTRLA.RUNSTDBY=0: Internal clock will be disabled, after any ongoing transfer was completed. The Receive Start and the Receive Complete interrupt(s) can wake up the device.
- External clocking, CTRLA.RUNSTDBY=0: External clock will be disconnected, after any ongoing transfer was completed. All reception will be dropped.

32.6.6. Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read. An exception is the Channel Enable bit in the Peripheral Channel Control registers (PCHCTRLm.CHEN). When changing this bit, the bit value must be read-back to ensure the synchronization is complete and to assert glitch free internal operation. Note that changing the bit value under ongoing synchronization will *not* generate an error.

The following bits are synchronized when written:

- Software Reset bit in the CTRLA register (CTRLA.SWRST)
- Enable bit in the CTRLA register (CTRLA.ENABLE)
- Receiver Enable bit in the CTRLB register (CTRLB.RXEN)
- Transmitter Enable bit in the Control B register (CTRLB.TXEN)

Note: CTRLB.RXEN is write-synchronized somewhat differently. See also CTRLB on page 589 for details.

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

32.7. Register Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
0x01		15:8	SAMPR[2:0]							IBON
0x02		23:16	SAMPA[1:0]		RXPO[1:0]				TXPO[1:0]	
0x03		31:24		DORD	CPOL	CMODE	FORM[3:0]			
0x04	CTRLB	7:0		SBMODE				CHSIZE[2:0]		
0x05		15:8			PMODE			ENC	SFDE	COLDEN
0x06		23:16							RXEN	TXEN
0x07		31:24								
0x08 ... 0x0B	Reserved									
0x0C	BAUD	7:0	BAUD[7:0]							
0x0D		15:8	BAUD[15:8]							
0x0E	RXPL	7:0	RXPL[7:0]							
0x0F ... 0x13	Reserved									
0x14	INTENCLR	7:0	ERROR		RXBRK	CTSIC	RXS	RXC	TXC	DRE
0x15	Reserved									
0x16	INTENSET	7:0	ERROR		RXBRK	CTSIC	RXS	RXC	TXC	DRE
0x17	Reserved									
0x18	INTFLAG	7:0	ERROR		RXBRK	CTSIC	RXS	RXC	TXC	DRE
0x19	Reserved									
0x1A	STATUS	7:0			COLL	ISF	CTS	BUFOVF	FERR	PERR
0x1B		15:8								
0x1C	SYNCBUSY	7:0						CTRLB	ENABLE	SWRST
0x1D		15:8								
0x1E		23:16								
0x1F		31:24								
0x20 ... 0x27	Reserved									
0x28	DATA	7:0	DATA[7:0]							
0x29		15:8								DATA[8:8]
0x2A ... 0x2F	Reserved									
0x30	DBGCTRL	7:0								DBGSTOP

32.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

32.8.1. Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
		DORD	CPOL	CMODE	FORM[3:0]			
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	SAMPA[1:0]		RXPO[1:0]				TXPO[1:0]	
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	0	0	0			0	0

Bit	15	14	13	12	11	10	9	8
	SAMPR[2:0]							IBON
Access	R/W	R/W	R/W					R
Reset	0	0	0					0

Bit	7	6	5	4	3	2	1	0
	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			0	0	0	0	0

Bit 30 – DORD: Data Order
This bit selects the data order when a character is shifted out from the Data register.

This bit is not synchronized.

Value	Description
0	MSB is transmitted first.
1	LSB is transmitted first.

Bit 29 – CPOL: Clock Polarity
This bit selects the relationship between data output change and data input sampling in synchronous mode.

This bit is not synchronized.

CPOL	TxD Change	RxD Sample
0x0	Rising XCK edge	Falling XCK edge
0x1	Falling XCK edge	Rising XCK edge

Bit 28 – CMODE: Communication Mode
This bit selects asynchronous or synchronous communication.

This bit is not synchronized.

Value	Description
0	Asynchronous communication.
1	Synchronous communication.

Bits 27:24 – FORM[3:0]: Frame Format
These bits define the frame format.

These bits are not synchronized.

FORM[3:0]	Description
0x0	USART frame
0x1	USART frame with parity
0x2-0x3	Reserved
0x4	Auto-baud - break detection and auto-baud.
0x5	Auto-baud - break detection and auto-baud with parity
0x6-0xF	Reserved

Bits 23:22 – SAMPA[1:0]: Sample Adjustment
These bits define the sample adjustment.

These bits are not synchronized.

SAMPA[1:0]	16x Over-sampling (CTRLA.SAMPR=0 or 1)	8x Over-sampling (CTRLA.SAMPR=2 or 3)
0x0	7-8-9	3-4-5
0x1	9-10-11	4-5-6
0x2	11-12-13	5-6-7
0x3	13-14-15	6-7-8

Bits 21:20 – RXPO[1:0]: Receive Data Pinout
These bits define the receive data (RxD) pin configuration.

These bits are not synchronized.

RXPO[1:0]	Name	Description
0x0	PAD[0]	SERCOM PAD[0] is used for data reception
0x1	PAD[1]	SERCOM PAD[1] is used for data reception
0x2	PAD[2]	SERCOM PAD[2] is used for data reception
0x3	PAD[3]	SERCOM PAD[3] is used for data reception

Bits 17:16 – TXPO[1:0]: Transmit Data Pinout
These bits define the transmit data (TxD) and XCK pin configurations.

This bit is not synchronized.

TXPO	TxD Pin Location	XCK Pin Location (When Applicable)	RTS	CTS
0x0	SERCOM PAD[0]	SERCOM PAD[1]	N/A	N/A
0x1	SERCOM PAD[2]	SERCOM PAD[3]	N/A	N/A
0x2	SERCOM PAD[0]	N/A	SERCOM PAD[2]	SERCOM PAD[3]
0x3	Reserved			

Bits 15:13 – SAMPR[2:0]: Sample Rate
These bits select the sample rate.

These bits are not synchronized.

SAMPR[2:0]	Description
0x0	16x over-sampling using arithmetic baud rate generation.
0x1	16x over-sampling using fractional baud rate generation.
0x2	8x over-sampling using arithmetic baud rate generation.
0x3	8x over-sampling using fractional baud rate generation.
0x4	3x over-sampling using arithmetic baud rate generation.
0x5-0x7	Reserved

Bit 8 – IBON: Immediate Buffer Overflow Notification
This bit controls when the buffer overflow status bit (STATUS.BUFOVF) is asserted when a buffer overflow occurs.

Value	Description
0	STATUS.BUFOVF is asserted when it occurs in the data stream.
1	STATUS.BUFOVF is asserted immediately upon buffer overflow.

Bit 7 – RUNSTDBY: Run In Standby
This bit defines the functionality in standby sleep mode.

This bit is not synchronized.

RUNSTDBY	External Clock	Internal Clock
0x0	External clock is disconnected when ongoing transfer is finished. All reception is dropped.	Generic clock is disabled when ongoing transfer is finished. The device can wake up on Receive Start or Transfer Complete interrupt.
0x1	Wake on Receive Start or Receive Complete interrupt.	Generic clock is enabled in all sleep modes. Any interrupt can wake up the device.

Bits 4:2 – MODE[2:0]: Operating Mode
These bits select the USART serial communication interface of the SERCOM.

These bits are not synchronized.

Value	Description
0x0	USART with external clock
0x1	USART with internal clock

Bit 1 – ENABLE: Enable
Due to synchronization, there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the Enable Synchronization Busy bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE is cleared when the operation is complete.

This bit is not enable-protected.

Value	Description
0	The peripheral is disabled or being disabled.
1	The peripheral is enabled or being enabled.

Bit 0 – SWRST: Software Reset
Writing '0' to this bit has no effect.

Writing '1' to this bit resets all registers in the SERCOM, except DBGCTRL, to their initial state, and the SERCOM will be disabled.

Writing '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded. Any register write access during the ongoing reset will result in an APB error. Reading any register will return the reset value of the register.

Due to synchronization, there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

This bit is not enable-protected.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

32.8.2. Control B

Name: CTRLB
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							RXEN	TXEN
Access							R/W	R/W
Reset							0	0

Bit	15	14	13	12	11	10	9	8
			PMODE			ENC	SFDE	COLDEN
Access			R/W			R/W	R/W	R/W
Reset			0			0	0	0

Bit	7	6	5	4	3	2	1	0
		SBMODE				CHSIZE[2:0]		
Access		R/W				R/W	R/W	R/W
Reset		0				0	0	0

Bit 17 – RXEN: Receiver Enable
Writing '0' to this bit will disable the USART receiver. Disabling the receiver will flush the receive buffer and clear the FERR, PERR and BUFOVF bits in the STATUS register.

Writing '1' to CTRLB.RXEN when the USART is disabled will set CTRLB.RXEN immediately. When the USART is enabled, CTRLB.RXEN will be cleared, and SYNCBUSY.CTRLB will be set and remain set until the receiver is enabled. When the receiver is enabled, CTRLB.RXEN will read back as '1'.

Writing '1' to CTRLB.RXEN when the USART is enabled will set SYNCBUSY.CTRLB, which will remain set until the receiver is enabled, and CTRLB.RXEN will read back as '1'.

This bit is not enable-protected.

Value	Description
0	The receiver is disabled or being enabled.
1	The receiver is enabled or will be enabled when the USART is enabled.

Bit 16 – TXEN: Transmitter Enable
Writing '0' to this bit will disable the USART transmitter. Disabling the transmitter will not become effective until ongoing and pending transmissions are completed.

Writing '1' to CTRLB.TXEN when the USART is disabled will set CTRLB.TXEN immediately. When the USART is enabled, CTRLB.TXEN will be cleared, and SYNCBUSY.CTRLB will be set and remain set until the transmitter is enabled. When the transmitter is enabled, CTRLB.TXEN will read back as '1'.

Writing '1' to CTRLB.TXEN when the USART is enabled will set SYNCBUSY.CTRLB, which will remain set until the receiver is enabled, and CTRLB.TXEN will read back as '1'.

This bit is not enable-protected.

Value	Description
0	The transmitter is disabled or being enabled.
1	The transmitter is enabled or will be enabled when the USART is enabled.

Bit 13 – PMODE: Parity Mode
This bit selects the type of parity used when parity is enabled (CTRLA.FORM is '1'). The transmitter will automatically generate and send the parity of the transmitted data bits within each frame. The receiver will generate a parity value for the incoming data and parity bit, compare it to the parity mode and, if a mismatch is detected, STATUS.PERR will be set.

This bit is not synchronized.

Value	Description
0	Even parity.
1	Odd parity.

Bit 10 – ENC: Encoding Format
This bit selects the data encoding format.

This bit is not synchronized.

Value	Description
0	Data is not encoded.
1	Data is IrDA encoded.

Bit 9 – SFDE: Start of Frame Detection Enable
This bit controls whether the start-of-frame detector will wake up the device when a start bit is detected on the RxD line.

This bit is not synchronized.

SFDE	INTENSET.RXS	INTENSET.RXC	Description
0	X	X	Start-of-frame detection disabled.
1	0	0	Reserved
1	0	1	Start-of-frame detection enabled. RXC wakes up the device from all sleep modes.
1	1	0	Start-of-frame detection enabled. RXS wakes up the device from all sleep modes.
1	1	1	Start-of-frame detection enabled. Both RXC and RXS wake up the device from all sleep modes.

Bit 8 – COLDEN: Collision Detection Enable
This bit enables collision detection.

This bit is not synchronized.

Value	Description
0	Collision detection is not enabled.
1	Collision detection is enabled.

Bit 6 – SBMODE: Stop Bit Mode
This bit selects the number of stop bits transmitted.

This bit is not synchronized.

Value	Description
0	One stop bit.
1	Two stop bits.

Bits 2:0 – CHSIZE[2:0]: Character Size
These bits select the number of bits in a character.

These bits are not synchronized.

CHSIZE[2:0]	Description
0x0	8 bits
0x1	9 bits
0x2-0x4	Reserved
0x5	5 bits
0x6	6 bits
0x7	7 bits

32.8.3. Baud

Name: BAUD
Offset: 0x0C
Reset: 0x0000
Property: Enable-Protected, PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
	BAUD[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BAUD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – BAUD[15:0]: Baud Value
Arithmetic Baud Rate Generation (`CTRLA.SAMPR[0]=0`):

These bits control the clock generation, as described in the SERCOM Baud Rate section.

If Fractional Baud Rate Generation (`CTRLA.SAMPR[0]=1`) bit positions 15 to 13 are replaced by FP[2:0] Fractional Part:

- **Bits 15:13 - FP[2:0]: Fractional Part**

  These bits control the clock generation, as described in the *SERCOM Clock Generation – Baud-Rate Generator* section.
- **Bits 12:0 - BAUD[21:0]: Baud Value**

  These bits control the clock generation, as described in the *SERCOM Clock Generation – Baud-Rate Generator* section.

32.8.4. Receive Pulse Length Register

Name: RXPL
Offset: 0x0E
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	RXPL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – RXPL[7:0]: Receive Pulse Length
When the encoding format is set to IrDA (CTRLB.ENC=1), these bits control the minimum pulse length that is required for a pulse to be accepted by the IrDA receiver with regards to the serial engine clock period SE_{per}.

$$PULSE \geq \left(\mathrm{RXPL} + 2\right) \cdot SE_{per}$$

32.8.5. Interrupt Enable Clear

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Name: INTENCLR
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	ERROR		RXBRK	CTSIC	RXS	RXC	TXC	DRE
Access	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset	0		0	0	0	0	0	0

Bit 7 – ERROR: Error Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Error Interrupt Enable bit, which disables the Error interrupt.

Value	Description
0	Error interrupt is disabled.
1	Error interrupt is enabled.

Bit 5 – RXBRK: Receive Break Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Receive Break Interrupt Enable bit, which disables the Receive Break interrupt.

Value	Description
0	Receive Break interrupt is disabled.
1	Receive Break interrupt is enabled.

Bit 4 – CTSIC: Clear to Send Input Change Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Clear To Send Input Change Interrupt Enable bit, which disables the Clear To Send Input Change interrupt.

Value	Description
0	Clear To Send Input Change interrupt is disabled.
1	Clear To Send Input Change interrupt is enabled.

Bit 3 – RXS: Receive Start Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Receive Start Interrupt Enable bit, which disables the Receive Start interrupt.

Value	Description
0	Receive Start interrupt is disabled.
1	Receive Start interrupt is enabled.

Bit 2 – RXC: Receive Complete Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Receive Complete Interrupt Enable bit, which disables the Receive Complete interrupt.

Value	Description
0	Receive Complete interrupt is disabled.
1	Receive Complete interrupt is enabled.

Bit 1 – TXC: Transmit Complete Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Transmit Complete Interrupt Enable bit, which disables the Receive Complete interrupt.

Value	Description
0	Transmit Complete interrupt is disabled.
1	Transmit Complete interrupt is enabled.

Bit 0 – DRE: Data Register Empty Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Data Register Empty Interrupt Enable bit, which disables the Data Register Empty interrupt.

Value	Description
0	Data Register Empty interrupt is disabled.
1	Data Register Empty interrupt is enabled.

32.8.6. Interrupt Enable Set

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Name: INTENSET
Offset: 0x16
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	ERROR		RXBRK	CTSIC	RXS	RXC	TXC	DRE
Access	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset	0		0	0	0	0	0	0

Bit 7 – ERROR: Error Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Error Interrupt Enable bit, which enables the Error interrupt.

Value	Description
0	Error interrupt is disabled.
1	Error interrupt is enabled.

Bit 5 – RXBRK: Receive Break Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Receive Break Interrupt Enable bit, which enables the Receive Break interrupt.

Value	Description
0	Receive Break interrupt is disabled.
1	Receive Break interrupt is enabled.

Bit 4 – CTSIC: Clear to Send Input Change Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Clear To Send Input Change Interrupt Enable bit, which enables the Clear To Send Input Change interrupt.

Value	Description
0	Clear To Send Input Change interrupt is disabled.
1	Clear To Send Input Change interrupt is enabled.

Bit 3 – RXS: Receive Start Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Receive Start Interrupt Enable bit, which enables the Receive Start interrupt.

Value	Description
0	Receive Start interrupt is disabled.
1	Receive Start interrupt is enabled.

Bit 2 – RXC: Receive Complete Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Receive Complete Interrupt Enable bit, which enables the Receive Complete interrupt.

Value	Description
0	Receive Complete interrupt is disabled.
1	Receive Complete interrupt is enabled.

Bit 1 – TXC: Transmit Complete Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Transmit Complete Interrupt Enable bit, which enables the Transmit Complete interrupt.

Value	Description
0	Transmit Complete interrupt is disabled.
1	Transmit Complete interrupt is enabled.

Bit 0 – DRE: Data Register Empty Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Data Register Empty Interrupt Enable bit, which enables the Data Register Empty interrupt.

Value	Description
0	Data Register Empty interrupt is disabled.
1	Data Register Empty interrupt is enabled.

32.8.7. Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x18
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	ERROR		RXBRK	CTSIC	RXS	RXC	TXC	DRE
Access	R/W		R/W	R/W	R/W	R	R/W	R
Reset	0		0	0	0	0	0	0

Bit 7 – ERROR: Error
This flag is cleared by writing '1' to it.

This bit is set when any error is detected. Errors that will set this flag have corresponding status flags in the STATUS register. Errors that will set this flag are COLL, ISF, BUFOVF, FERR, and PERR.Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the flag.

Bit 5 – RXBRK: Receive Break
This flag is cleared by writing '1' to it.

This flag is set when auto-baud is enabled (CTRLA.FORM) and a break character is received.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the flag.

Bit 4 – CTSIC: Clear to Send Input Change
This flag is cleared by writing a '1' to it.

This flag is set when a change is detected on the CTS pin.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the flag.

Bit 3 – RXS: Receive Start
This flag is cleared by writing '1' to it.

This flag is set when a start condition is detected on the RxD line and start-of-frame detection is enabled (CTRLB.SFDE is '1').

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Receive Start interrupt flag.

Bit 2 – RXC: Receive Complete
This flag is cleared by reading the Data register (DATA) or by disabling the receiver.

This flag is set when there are unread data in DATA.

Writing '0' to this bit has no effect.

Writing '1' to this bit has no effect.

Bit 1 – TXC: Transmit Complete
This flag is cleared by writing '1' to it or by writing new data to DATA.

This flag is set when the entire frame in the transmit shift register has been shifted out and there are no new data in DATA.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the flag.

Bit 0 – DRE: Data Register Empty
This flag is cleared by writing new data to DATA.

This flag is set when DATA is empty and ready to be written.

Writing '0' to this bit has no effect.

Writing '1' to this bit has no effect.

32.8.8. Status

Name: STATUS
Offset: 0x1A
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			COLL	ISF	CTS	BUFOVF	FERR	PERR
Access			R/W	R/W	R	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 5 – COLL: Collision Detected
This bit is cleared by writing '1' to the bit or by disabling the receiver.

This bit is set when collision detection is enabled (CTRLB.COLDEN) and a collision is detected.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear it.

Bit 4 – ISF: Inconsistent Sync Field
This bit is cleared by writing '1' to the bit or by disabling the receiver.

This bit is set when the frame format is set to auto-baud (CTRLA.FORM) and a sync field not equal to 0x55 is received.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear it.

Bit 3 – CTS: Clear to Send
This bit indicates the current level of the CTS pin when flow control is enabled (CTRLA.TXPO).

Writing '0' to this bit has no effect.

Writing '1' to this bit has no effect.

Bit 2 – BUFOVF: Buffer Overflow
Reading this bit before reading the Data register will indicate the error status of the next character to be read.

This bit is cleared by writing '1' to the bit or by disabling the receiver.

This bit is set when a buffer overflow condition is detected. A buffer overflow occurs when the receive buffer is full, there is a new character waiting in the receive shift register and a new start bit is detected.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear it.

Bit 1 – FERR: Frame Error
Reading this bit before reading the Data register will indicate the error status of the next character to be read.

This bit is cleared by writing '1' to the bit or by disabling the receiver.

This bit is set if the received character had a frame error, i.e., when the first stop bit is zero.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear it.

Bit 0 – PERR: Parity Error
Reading this bit before reading the Data register will indicate the error status of the next character to be read.

This bit is cleared by writing '1' to the bit or by disabling the receiver.

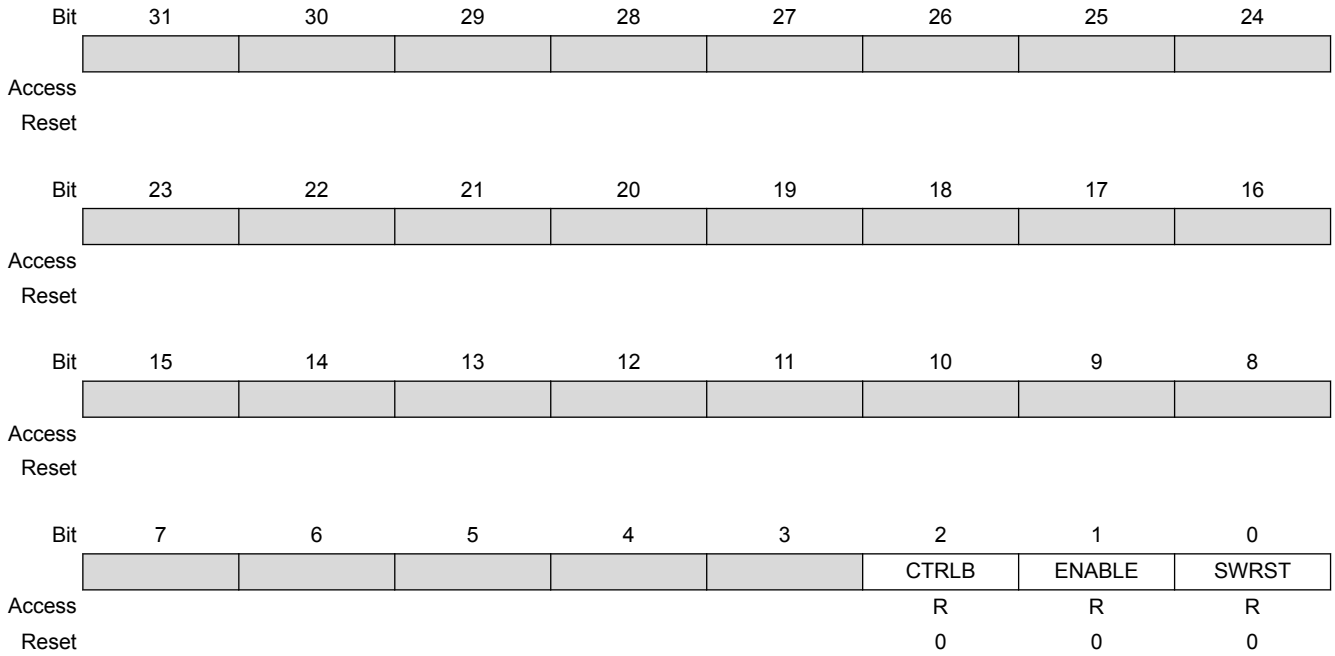
This bit is set if parity checking is enabled (CTRLA.FORM is 0x1, 0x5) and a parity error is detected.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear it.

32.8.9. Synchronization Busy

Name: SYNCBUSY
Offset: 0x1C
Reset: 0x00000000
Property: -



Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						CTRLB	ENABLE	SWRST
Access						R	R	R
Reset						0	0	0

Bit 2 – CTRLB: CTRLB Synchronization Busy
Writing to the CTRLB register when the SERCOM is enabled requires synchronization. When writing to CTRLB the SYNCBUSY.CTRLB bit will be set until synchronization is complete. If CTRLB is written while SYNCBUSY.CTRLB is asserted, an APB error will be generated.

Value	Description
0	CTRLB synchronization is not busy.
1	CTRLB synchronization is busy.

Bit 1 – ENABLE: SERCOM Enable Synchronization Busy
Enabling and disabling the SERCOM (CTRLA.ENABLE) requires synchronization. When written, the SYNCBUSY.ENABLE bit will be set until synchronization is complete.

Writes to any register (except for CTRLA.SWRST) while enable synchronization is on-going will be discarded and an APB error will be generated.

Value	Description
0	Enable synchronization is not busy.
1	Enable synchronization is busy.

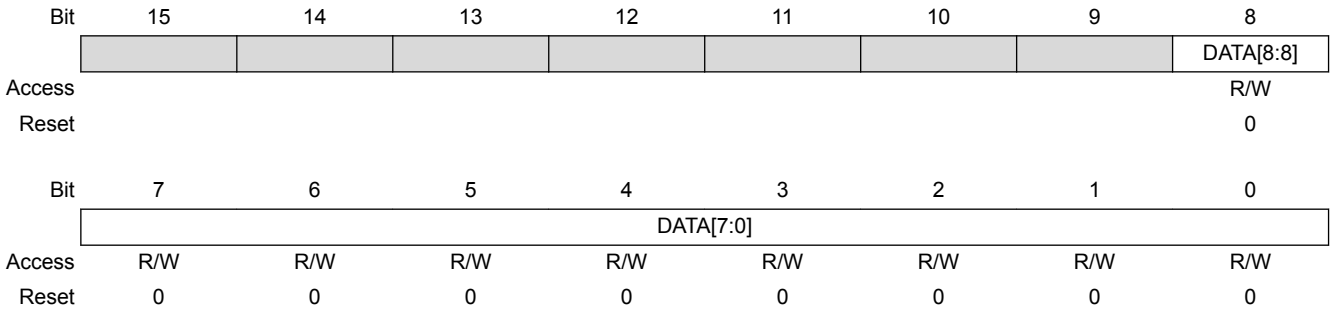
Bit 0 – SWRST: Software Reset Synchronization Busy
Resetting the SERCOM (CTRLA.SWRST) requires synchronization. When written, the SYNCBUSY.SWRST bit will be set until synchronization is complete.

Writes to any register while synchronization is on-going will be discarded and an APB error will be generated.

Value	Description
0	SWRST synchronization is not busy.
1	SWRST synchronization is busy.

32.8.10. Data

Name: DATA
Offset: 0x28
Reset: 0x0000
Property: -



Bits 8:0 – DATA[8:0]: Data

Reading these bits will return the contents of the Receive Data register. The register should be read only when the Receive Complete Interrupt Flag bit in the Interrupt Flag Status and Clear register (INTFLAG.RXC) is set. The status bits in STATUS should be read before reading the DATA value in order to get any corresponding error.

Writing these bits will write the Transmit Data register. This register should be written only when the Data Register Empty Interrupt Flag bit in the Interrupt Flag Status and Clear register (INTFLAG.DRE) is set.

32.8.11. Debug Control

Name: DBGCTRL
Offset: 0x30
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGSTOP
Access								R/W
Reset								0

Bit 0 – DBGSTOP: Debug Stop Mode
This bit controls the baud-rate generator functionality when the CPU is halted by an external debugger.

Value	Description
0	The baud-rate generator continues normal operation when the CPU is halted by an external debugger.
1	The baud-rate generator is halted when the CPU is halted by an external debugger.

33. SERCOM SPI – SERCOM Serial Peripheral Interface

33.1. Overview

The serial peripheral interface (SPI) is one of the available modes in the Serial Communication Interface (SERCOM).

The SPI uses the SERCOM transmitter and receiver configured as shown in Block Diagram on page 607. Each side, master and slave, depicts a separate SPI containing a shift register, a transmit buffer and two receive buffers. In addition, the SPI master uses the SERCOM baud-rate generator, while the SPI slave can use the SERCOM address match logic. Labels in capital letters are synchronous to CLK_SERCOMx_APB and accessible by the CPU, while labels in lowercase letters are synchronous to the SCK clock.

Related Links

SERCOM – Serial Communication Interface on page 560

33.2. Features

SERCOM SPI includes the following features:

- Full-duplex, four-wire interface (MISO, MOSI, SCK, $\overline{SS}$)
- Single-buffered transmitter, double-buffered receiver
- Supports all four SPI modes of operation
- Single data direction operation allows alternate function on MISO or MOSI pin
- Selectable LSB- or MSB-first data transfer
- Can be used with DMA
- Master operation:
  - Serial clock speed up to 12MHz
  - 8-bit clock generator
  - Hardware controlled $\overline{SS}$
- Slave operation:
  - Serial clock speed up to the system clock
  - Optional 8-bit address match operation
  - Operation in all sleep modes
  - Wake on $\overline{SS}$ transition

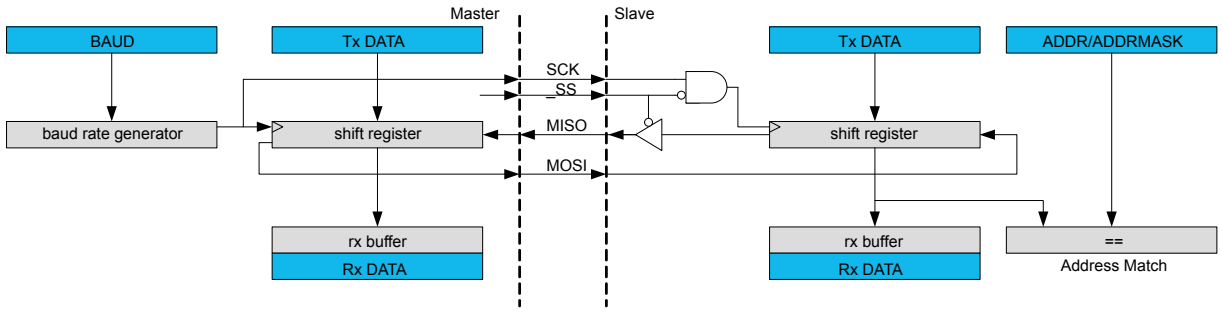
Related Links

SERCOM – Serial Communication Interface on page 560

Features on page 560

33.3. Block Diagram

Figure 33-1. Full-Duplex SPI Master Slave Interconnection



33.4. Signal Description

Table 33-1. SERCOM SPI Signals

Signal Name	Type	Description
PAD[3:0]	Digital I/O	General SERCOM pins

One signal can be mapped to one of several pins.

Related Links

I/O Multiplexing and Considerations on page 29

33.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

33.5.1. I/O Lines

In order to use the SERCOM's I/O lines, the I/O pins must be configured using the IO Pin Controller (PORT).

When the SERCOM is configured for SPI operation, the SERCOM controls the direction and value of the I/O pins according to the table below. Both PORT control bits PINCFGn.PULLEN and PINCFGn.DRVSTR are still effective. If the receiver is disabled, the data input pin can be used for other purposes. In master mode, the slave select line ($\overline{SS}$) is hardware controlled when the Master Slave Select Enable bit in the Control B register (CTRLB.MSSEN) is '1'.

Table 33-2. SPI Pin Configuration

Pin	Master SPI	Slave SPI
MOSI	Output	Input
MISO	Input	Output
SCK	Output	Input
$\overline{SS}$	Output (CTRLB.MSSEN=1)	Input

The combined configuration of PORT, the Data In Pinout and the Data Out Pinout bit groups in the Control A register (CTRLA.DIPO and CTRLA.DOPO) define the physical position of the SPI signals in the table above.

Related Links

PORT: IO Pin Controller on page 506

33.5.2. Power Management

This peripheral can continue to operate in any sleep mode where its source clock is running. The interrupts can wake up the device from sleep modes. Refer to *PM – Power Manager* for details on the different sleep modes.

Related Links

PM – Power Manager on page 186

33.5.3. Clocks

The SERCOM bus clock (CLK_SERCOMx_APB) is enabled by default, and can be enabled and disabled in the Main Clock.

A generic clock (GCLK_SERCOMx_CORE) is required to clock the SPI. This clock must be configured and enabled in the Generic Clock Controller before using the SPI.

This generic clock is asynchronous to the bus clock (CLK_SERCOMx_APB). Therefore, writes to certain registers will require synchronization to the clock domains.

Related Links

GCLK - Generic Clock Controller on page 131

Peripheral Clock Masking on page 151

Synchronization on page 617

33.5.4. DMA

The DMA request lines are connected to the DMA Controller (DMAC). In order to use DMA requests with this peripheral the DMAC must be configured first. Refer to *DMAC – Direct Memory Access Controller* for details.

Related Links

DMAC – Direct Memory Access Controller on page 399

33.5.5. Interrupts

The interrupt request line is connected to the Interrupt Controller. In order to use interrupt requests of this peripheral, the Interrupt Controller (NVIC) must be configured first. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

Nested Vector Interrupt Controller on page 51

33.5.6. Events

Not applicable.

33.5.7. Debug Operation

When the CPU is halted in debug mode, this peripheral will continue normal operation. If the peripheral is configured to require periodical service by the CPU through interrupts or similar, improper operation or data loss may result during debugging. This peripheral can be forced to halt operation during debugging - refer to the Debug Control (DBGCTRL) register for details.

33.5.8. Register Access Protection

Registers with write-access can be write-protected optionally by the peripheral access controller (PAC).

PAC Write-Protection is not available for the following registers:

- Interrupt Flag Clear and Status register (INTFLAG)
- Status register (STATUS)
- Data register (DATA)

Optional PAC Write-Protection is denoted by the "PAC Write-Protection" property in each individual register description.

Write-protection does not apply to accesses through an external debugger.

Related Links

PAC - Peripheral Access Controller on page 58

33.5.9. Analog Connections

Not applicable.

33.6. Functional Description

33.6.1. Principle of Operation

The SPI is a high-speed synchronous data transfer interface It allows high-speed communication between the device and peripheral devices.

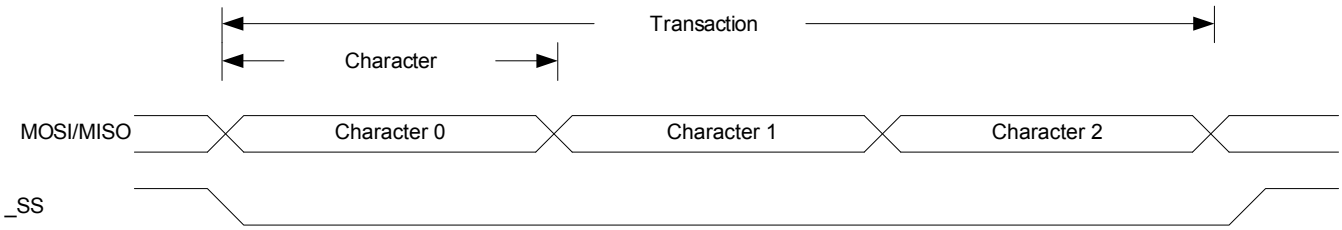
The SPI can operate as master or slave. As master, the SPI initiates and controls all data transactions. The SPI is single buffered for transmitting and double buffered for receiving.

When transmitting data, the Data register can be loaded with the next character to be transmitted during the current transmission.

When receiving, the data is transferred to the two-level receive buffer, and the receiver is ready for a new character.

The SPI transaction format is shown in SPI Transaction Format. Each transaction can contain one or more characters. The character size is configurable, and can be either 8 or 9 bits.

Figure 33-2. SPI Transaction Format



The SPI master must pull the slave select line ($\overline{SS}$) of the desired slave low to initiate a transaction. The master and slave prepare data to send via their respective shift registers, and the master generates the serial clock on the SCK line.

Data are always shifted from master to slave on the Master Output Slave Input line (MOSI); data is shifted from slave to master on the Master Input Slave Output line (MISO).

Each time character is shifted out from the master, a character will be shifted out from the slave simultaneously. To signal the end of a transaction, the master will pull the $\overline{SS}$ line high

33.6.2. Basic Operation

33.6.2.1. Initialization

The following registers are enable-protected, meaning that they can only be written when the SPI is disabled (CTRL.ENABLE=0):

- Control A register (CTRLA), except Enable (CTRLA.ENABLE) and Software Reset (CTRLA.SWRST)
- Control B register (CTRLB), except Receiver Enable (CTRLB.RXEN)
- Baud register (BAUD)
- Address register (ADDR)

When the SPI is enabled or is being enabled (CTRLA.ENABLE=1), any writing to these registers will be discarded.

when the SPI is being disabled, writing to these registers will be completed after the disabling.

Enable-protection is denoted by the Enable-Protection property in the register description.

Initialize the SPI by following these steps:

1. Select SPI mode in master / slave operation in the Operating Mode bit group in the CTRLA register (CTRLA.MODE= 0x2 or 0x3).
2. Select transfer mode for the Clock Polarity bit and the Clock Phase bit in the CTRLA register (CTRLA.CPOL and CTRLA.CPHA) if desired.
3. Select the Frame Format value in the CTRLA register (CTRLA.FORM).
4. Configure the Data In Pinout field in the Control A register (CTRLA.DIPO) for SERCOM pads of the receiver.
5. Configure the Data Out Pinout bit group in the Control A register (CTRLA.DOPO) for SERCOM pads of the transmitter.
6. Select the Character Size value in the CTRLB register (CTRLB.CHSIZE).
7. Write the Data Order bit in the CTRLA register (CTRLA.DORD) for data direction.
8. If the SPI is used in master mode:
   8.1. Select the desired baud rate by writing to the Baud register (BAUD).
   8.2. If Hardware SS control is required, write '1' to the Master Slave Select Enable bit in CTRLB register (CTRLB.MSSEN).
9. Enable the receiver by writing the Receiver Enable bit in the CTRLB register (CTRLB.RXEN=1).

33.6.2.2. Enabling, Disabling, and Resetting

This peripheral is enabled by writing '1' to the Enable bit in the Control A register (CTRLA.ENABLE), and disabled by writing '0' to it.

Writing ‘1’ to the Software Reset bit in the Control A register (CTRLA.SWRST) will reset all registers of this peripheral to their initial states, except the DBGCTRL register, and the peripheral is disabled.

Refer to the CTRLA register description for details.

33.6.2.3. Clock Generation

In SPI master operation (CTRLA.MODE=0x3), the serial clock (SCK) is generated internally by the SERCOM baud-rate generator.

In SPI mode, the baud-rate generator is set to synchronous mode. The 8-bit Baud register (BAUD) value is used for generating SCK and clocking the shift register. Refer to *Clock Generation – Baud-Rate Generator* for more details.

In SPI slave operation (CTRLA.MODE is 0x2), the clock is provided by an external master on the SCK pin. This clock is used to directly clock the SPI shift register.

Related Links

Clock Generation – Baud-Rate Generator on page 564

Asynchronous Arithmetic Mode BAUD Value Selection on page 565

33.6.2.4. Data Register

The SPI Transmit Data register (TxDATA) and SPI Receive Data register (RxDATA) share the same I/O address, referred to as the SPI Data register (DATA). Writing DATA register will update the Transmit Data register. Reading the DATA register will return the contents of the Receive Data register.

33.6.2.5. SPI Transfer Modes

There are four combinations of SCK phase and polarity to transfer serial data. The SPI data transfer modes are shown in SPI Transfer Modes (Table) and SPI Transfer Modes (Figure).

SCK phase is configured by the Clock Phase bit in the CTRLA register (CTRLA.CPHA). SCK polarity is programmed by the Clock Polarity bit in the CTRLA register (CTRLA.CPOL). Data bits are shifted out and latched in on opposite edges of the SCK signal. This ensures sufficient time for the data signals to stabilize.

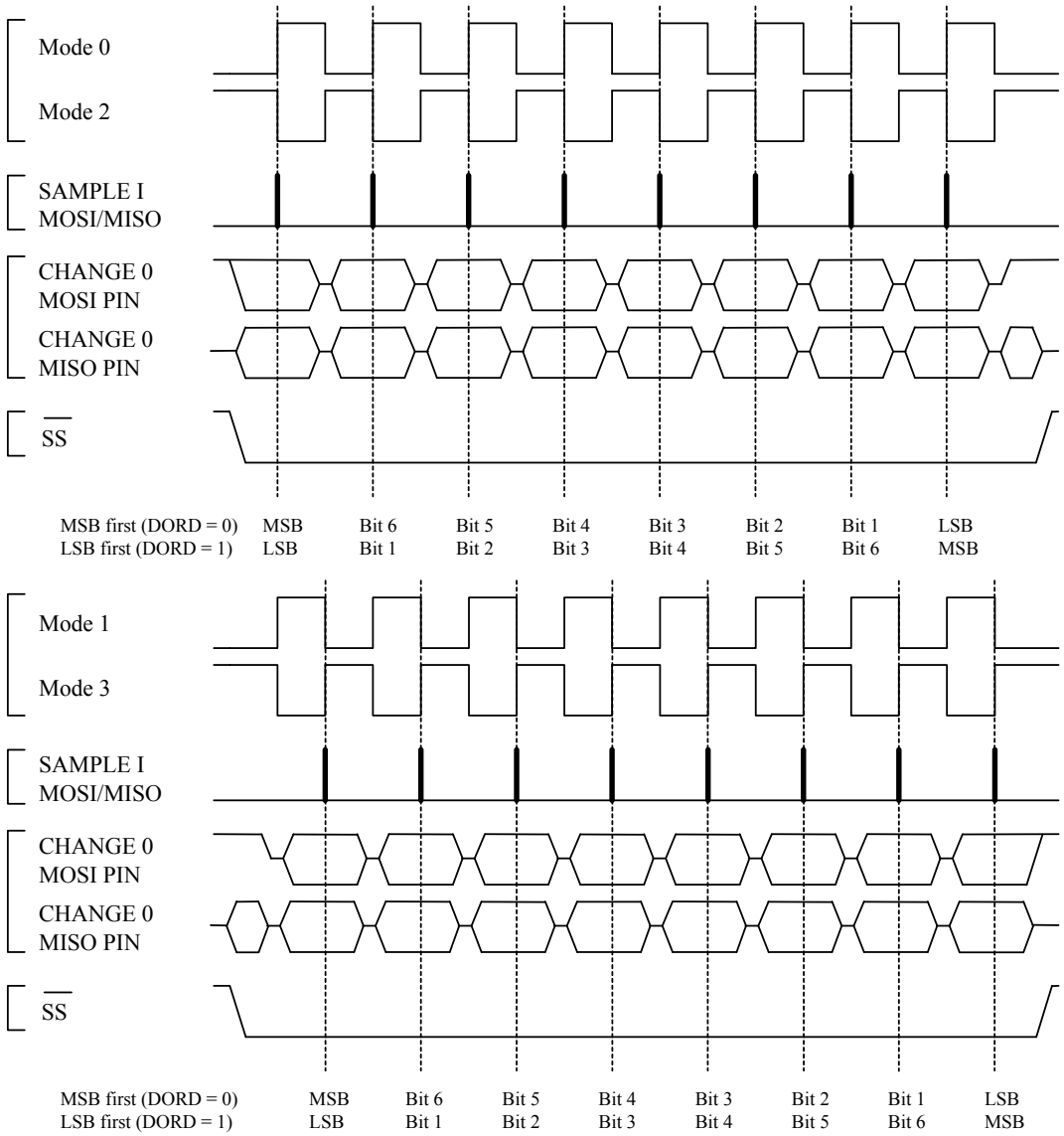
Table 33-3. SPI Transfer Modes

Mode	CPOL	CPHA	Leading Edge	Trailing Edge
0	0	0	Rising, sample	Falling, setup
1	0	1	Rising, setup	Falling, sample
2	1	0	Falling, sample	Rising, setup
3	1	1	Falling, setup	Rising, sample

Note:
Leading edge is the first clock edge in a clock cycle.

Trailing edge is the second clock edge in a clock cycle.

Figure 33-3. SPI Transfer Modes



33.6.2.6. Transferring Data

Master

In master mode (CTRLA.MODE=0x3), when Master Slave Enable Select (CTRLB.MSSEN) is '1', hardware will control the $\overline{SS}$ line.

When Master Slave Select Enable (CTRLB.MSSEN) is '0', the $\overline{SS}$ line must be configured as an output. $\overline{SS}$ can be assigned to any general purpose I/O pin. When the SPI is ready for a data transaction, software must pull the $\overline{SS}$ line low.

When writing a character to the Data register (DATA), the character will be transferred to the shift register. Once the content of TxDATA has been transferred to the shift register, the Data Register Empty flag in the Interrupt Flag Status and Clear register (INTFLAG.DRE) will be set. And a new character can be written to DATA.

Each time one character is shifted out from the master, another character will be shifted in from the slave simultaneously. If the receiver is enabled (CTRLA.RXEN=1), the contents of the shift register will be transferred to the two-level receive buffer. The transfer takes place in the same clock cycle as the last

data bit is shifted in. And the Receive Complete Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.RXC) will be set. The received data can be retrieved by reading DATA.

When the last character has been transmitted and there is no valid data in DATA, the Transmit Complete Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.TXC) will be set. When the transaction is finished, the master must pull the $\overline{SS}$ line high to notify the slave. If Master Slave Select Enable (CTRLB.MSSEN) is set to '0', the software must pull the $\overline{SS}$ line high.

Slave

In slave mode (CTRLA.MODE=0x2), the SPI interface will remain inactive with the MISO line tri-stated as long as the $\overline{SS}$ pin is pulled high. Software may update the contents of DATA at any time as long as the Data Register Empty flag in the Interrupt Status and Clear register (INTFLAG.DRE) is set.

When $\overline{SS}$ is pulled low and SCK is running, the slave will sample and shift out data according to the transaction mode set. When the content of TxDATA has been loaded into the shift register, INTFLAG.DRE will be set, and new data can be written to DATA.

Similar to the master, the slave will receive one character for each character transmitted. A character will be transferred into the two-level receive buffer within the same clock cycle its last data bit is received. The received character can be retrieved from DATA when the Receive Complete interrupt flag (INTFLAG.RXC) is set.

When the master pulls the $\overline{SS}$ line high, the transaction is done and the Transmit Complete Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.TXC) will be set.

After DATA is written it takes up to three SCK clock cycles until the content of DATA is ready to be loaded into the shift register on the next character boundary. As a consequence, the first character transferred in a SPI transaction will not be the content of DATA. This can be avoided by using the preloading feature. Refer to Preloading of the Slave Shift Register on page 614.

When transmitting several characters in one SPI transaction, the data has to be written into DATA register with at least three SCK clock cycles left in the current character transmission. If this criteria is not met, the previously received character will be transmitted.

Once the DATA register is empty, it takes three CLK_SERCOM_APB cycles for INTFLAG.DRE to be set.

33.6.2.7. Receiver Error Bit

The SPI receiver has one error bit: the Buffer Overflow bit (BUFOVF), which can be read from the Status register (STATUS). Once an error happens, the bit will stay set until it is cleared by writing '1' to it. The bit is also automatically cleared when the receiver is disabled.

There are two methods for buffer overflow notification, selected by the immediate buffer overflow notification bit in the Control A register (CTRLA.IBON):

If CTRLA.IBON=1, STATUS.BUFOVF is raised immediately upon buffer overflow. Software can then empty the receive FIFO by reading RxDATA until the receiver complete interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.RXC) goes low.

If CTRLA.IBON=0, the buffer overflow condition travels with data through the receive FIFO. After the received data is read, STATUS.BUFOVF and INTFLAG.ERROR will be set along with INTFLAG.RXC, and RxDATA will be zero.

33.6.3. Additional Features

33.6.3.1. Address Recognition

When the SPI is configured for slave operation (CTRLA.MODE=0x2) with address recognition (CTRLA.FORM is 0x2), the SERCOM address recognition logic is enabled: the first character in a transaction is checked for an address match.

If there is a match, the Receive Complete Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.RXC) is set, the MISO output is enabled, and the transaction is processed. If the device is in sleep mode, an address match can wake up the device in order to process the transaction.

If there is no match, the complete transaction is ignored.

If a 9-bit frame format is selected, only the lower 8 bits of the shift register are checked against the Address register (ADDR).

Preload must be disabled (CTRLB.PLOADEN=0) in order to use this mode.

Related Links

Address Match and Mask on page 566

33.6.3.2. Preloading of the Slave Shift Register

When starting a transaction, the slave will first transmit the contents of the shift register before loading new data from DATA. The first character sent can be either the reset value of the shift register (if this is the first transmission since the last reset) or the last character in the previous transmission.

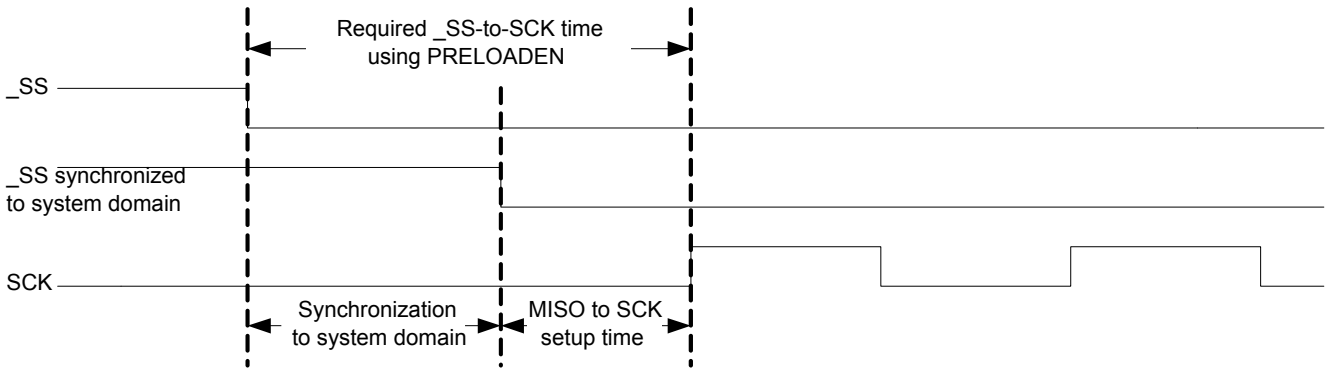
Preloading can be used to preload data into the shift register while $\overline{SS}$ is high: this eliminates sending a dummy character when starting a transaction. If the shift register is not preloaded, the current contents of the shift register will be shifted out.

Only one data character will be preloaded into the shift register while the synchronized $\overline{SS}$ signal is high. If the next character is written to DATA before $\overline{SS}$ is pulled low, the second character will be stored in DATA until transfer begins.

For proper preloading, sufficient time must elapse between $\overline{SS}$ going low and the first SCK sampling edge, as in Timing Using Preloading. See also *Electrical Characteristics* for timing details.

Preloading is enabled by writing '1' to the Slave Data Preload Enable bit in the CTRLB register (CTRLB.PLOADEN).

Figure 33-4. Timing Using Preloading



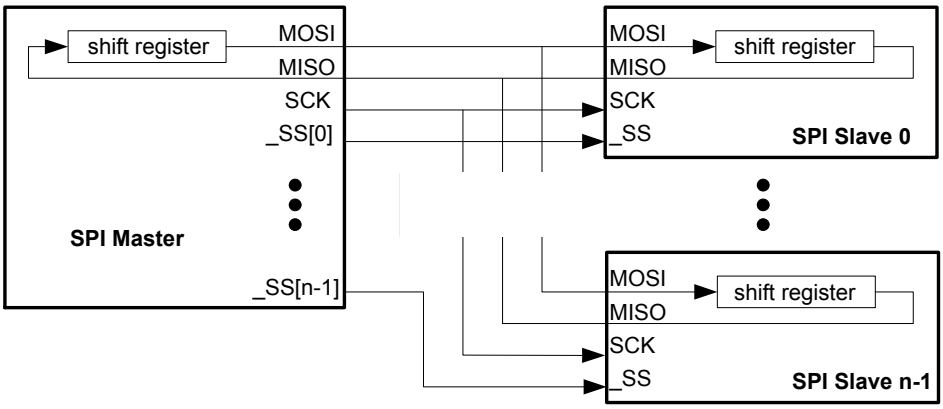
Related Links

Electrical Characteristics on page 1140

33.6.3.3. Master with Several Slaves

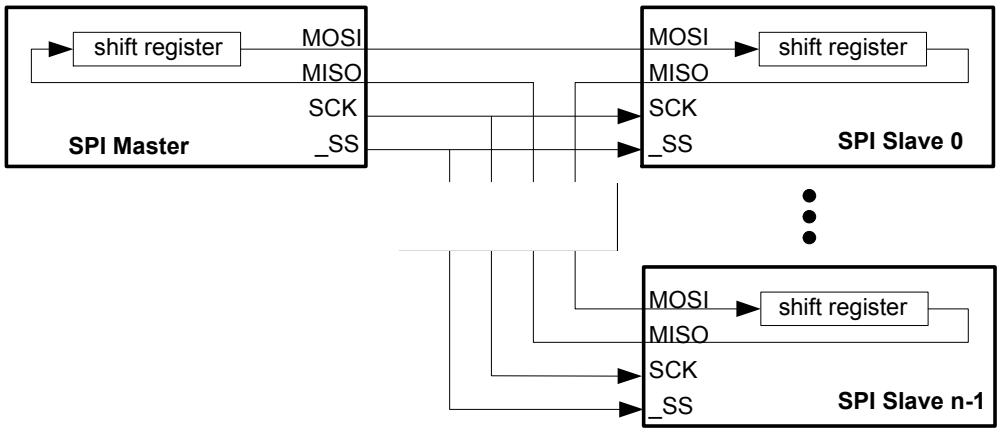
Master with multiple slaves in parallel is only available when Master Slave Select Enable (CTRLB.MSSEN) is set to zero and hardware $\overline{SS}$ control is disabled. If the bus consists of several SPI slaves, an SPI master can use general purpose I/O pins to control the $\overline{SS}$ line to each of the slaves on the bus, as shown in Multiple Slaves in Parallel. In this configuration, the single selected SPI slave will drive the tri-state MISO line.

Figure 33-5. Multiple Slaves in Parallel



Another configuration is multiple slaves in series, as in Multiple Slaves in Series. In this configuration, all n attached slaves are connected in series. A common $\overline{SS}$ line is provided to all slaves, enabling them simultaneously. The master must shift n characters for a complete transaction. Depending on the Master Slave Select Enable bit (CTRLB.MSSEN), the $\overline{SS}$ line can be controlled either by hardware or user software and normal GPIO.

Figure 33-6. Multiple Slaves in Series



33.6.3.4. Loop-Back Mode

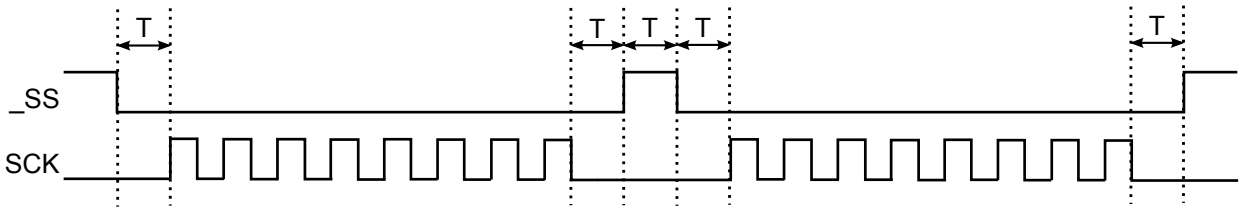
For loop-back mode, configure the Data In Pinout (CTRLA.DIPO) and Data Out Pinout (CTRLA.DOPO) to use the same data pins for transmit and receive. The loop-back is through the pad, so the signal is also available externally.

33.6.3.5. Hardware Controlled $\overline{SS}$

In master mode, a single $\overline{SS}$ chip select can be controlled by hardware by writing the Master Slave Select Enable (CTRLB.MSSEN) bit to '1'. In this mode, the $\overline{SS}$ pin is driven low for a minimum of one baud cycle before transmission begins, and stays low for a minimum of one baud cycle after transmission completes. If back-to-back frames are transmitted, the $\overline{SS}$ pin will always be driven high for a minimum of one baud cycle between frames.

In Hardware Controlled $\overline{SS}$, the time T is between one and two baud cycles depending on the SPI transfer mode.

Figure 33-7. Hardware Controlled $\overline{SS}$



T = 1 to 2 baud cycles

When CTRLB.MSSEN=0, the $\overline{SS}$ pin(s) is/are controlled by user software and normal GPIO.

33.6.3.6. Slave Select Low Detection

In slave mode, the SPI can wake the CPU when the slave select ($\overline{SS}$) goes low. When the Slave Select Low Detect is enabled (CTRLB.SSDE=1), a high-to-low transition will set the Slave Select Low interrupt flag (INTFLAG.SSL) and the device will wake up if applicable.

33.6.4. DMA, Interrupts, and Events

Table 33-4. Module Request for SERCOM SPI

Condition	Request		
	DMA	Interrupt	Event
Data Register Empty (DRE)	Yes (request cleared when data is written)	Yes	NA
Receive Complete (RXC)	Yes (request cleared when data is read)	Yes	
Transmit Complete (TXC)	NA	Yes	
Slave Select low (SSL)	NA	Yes	
Error (ERROR)	NA	Yes	

33.6.4.1. DMA Operation

The SPI generates the following DMA requests:

- Data received (RX): The request is set when data is available in the receive FIFO. The request is cleared when DATA is read.
- Data transmit (TX): The request is set when the transmit buffer (TX DATA) is empty. The request is cleared when DATA is written.

33.6.4.2. Interrupts

The SPI has the following interrupt sources. These are asynchronous interrupts, and can wake up the device from any sleep mode:

- Data Register Empty (DRE)
- Receive Complete (RXC)
- Transmit Complete (TXC)
- Slave Select Low (SSL)
- Error (ERROR)

Each interrupt source has its own interrupt flag. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) will be set when the interrupt condition is met. Each interrupt can be individually enabled by writing '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR).

An interrupt request is generated when the interrupt flag is set and if the corresponding interrupt is enabled. The interrupt request remains active until either the interrupt flag is cleared, the interrupt is disabled, or the SPI is reset. For details on clearing interrupt flags, refer to the INTFLAG register description.

The SPI has one common interrupt request line for all the interrupt sources. The value of INTFLAG indicates which interrupt is executed. Note that interrupts must be globally enabled for interrupt requests. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

Nested Vector Interrupt Controller on page 51

33.6.4.3. Events

Not applicable.

33.6.5. Sleep Mode Operation

The behavior in sleep mode is depending on the master/slave configuration and the Run In Standby bit in the Control A register (CTRLA.RUNSTDBY):

- Master operation, CTRLA.RUNSTDBY=1: The peripheral clock GCLK_SERCOM_CORE will continue to run in idle sleep mode and in standby sleep mode. Any interrupt can wake up the device.
- Master operation, CTRLA.RUNSTDBY=0: GLK_SERCOMx_CORE will be disabled after the ongoing transaction is finished. Any interrupt can wake up the device.
- Slave operation, CTRLA.RUNSTDBY=1: The Receive Complete interrupt can wake up the device.
- Slave operation, CTRLA.RUNSTDBY=0: All reception will be dropped, including the ongoing transaction.

33.6.6. Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read. An exception is the Channel Enable bit in the Peripheral Channel Control registers (PCHCTRLm.CHEN). When changing this bit, the bit value must be read-back to ensure the synchronization is complete and to assert glitch free internal operation. Note that changing the bit value under ongoing synchronization will *not* generate an error.

The following bits are synchronized when written:

- Software Reset bit in the CTRLA register (CTRLA.SWRST)
- Enable bit in the CTRLA register (CTRLA.ENABLE)
- Receiver Enable bit in the CTRLB register (CTRLB.RXEN)

Note: CTRLB.RXEN is write-synchronized somewhat differently. See also CTRLB on page 624 for details.

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

33.7. Register Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
0x01		15:8								IBON
0x02		23:16			DIPO[1:0]				DOPO[1:0]	
0x03		31:24		DORD	CPOL	CPHA	FORM[3:0]			
0x04	CTRLB	7:0		PLOADEN				CHSIZE[2:0]		
0x05		15:8	AMODE[1:0]		MSSEN				SSDE	
0x06		23:16							RXEN	
0x07		31:24								
0x08 ... 0x0B	Reserved									
0x0C	BAUD	7:0	BAUD[7:0]							
0x0D ... 0x13	Reserved									
0x14	INTENCLR	7:0	ERROR				SSL	RXC	TXC	DRE
0x15	Reserved									
0x16	INTENSET	7:0	ERROR				SSL	RXC	TXC	DRE
0x17	Reserved									
0x18	INTFLAG	7:0	ERROR				SSL	RXC	TXC	DRE
0x19	Reserved									
0x1A	STATUS	7:0						BUFOVF		
0x1B		15:8								
0x1C	SYNCBUSY	7:0						CTRLB	ENABLE	SWRST
0x1D		15:8								
0x1E		23:16								
0x1F		31:24								
0x20 ... 0x23	Reserved									
0x24	ADDR	7:0	ADDR[7:0]							
0x25		15:8								
0x26		23:16	ADDRMASK[7:0]							
0x27		31:24								
0x28	DATA	7:0	DATA[7:0]							
0x29		15:8								DATA[8:8]
0x2A ... 0x2F	Reserved									
0x30	DBGCTRL	7:0								DBGSTOP

33.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Refer to Synchronization on page 617

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

Refer to Register Access Protection on page 608.

33.8.1. Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
		DORD	CPOL	CPHA	FORM[3:0]			
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
			DIPO[1:0]				DOPO[1:0]	
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit	15	14	13	12	11	10	9	8
								IBON
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			0	0	0	0	0

Bit 30 – DORD: Data Order
This bit selects the data order when a character is shifted out from the shift register.

This bit is not synchronized.

Value	Description
0	MSB is transferred first.
1	LSB is transferred first.

Bit 29 – CPOL: Clock Polarity
In combination with the Clock Phase bit (CPHA), this bit determines the SPI transfer mode.

This bit is not synchronized.

Value	Description
0	SCK is low when idle. The leading edge of a clock cycle is a rising edge, while the trailing edge is a falling edge.
1	SCK is high when idle. The leading edge of a clock cycle is a falling edge, while the trailing edge is a rising edge.

Bit 28 – CPHA: Clock Phase
In combination with the Clock Polarity bit (CPOL), this bit determines the SPI transfer mode.

This bit is not synchronized.

Mode	CPOL	CPHA	Leading Edge	Trailing Edge
0x0	0	0	Rising, sample	Falling, change
0x1	0	1	Rising, change	Falling, sample
0x2	1	0	Falling, sample	Rising, change
0x3	1	1	Falling, change	Rising, sample

Value	Description
0	The data is sampled on a leading SCK edge and changed on a trailing SCK edge.
1	The data is sampled on a trailing SCK edge and changed on a leading SCK edge.

Bits 27:24 – FORM[3:0]: Frame Format
This bit field selects the various frame formats supported by the SPI in slave mode. When the 'SPI frame with address' format is selected, the first byte received is checked against the ADDR register.

FORM[3:0]	Name	Description
0x0	SPI	SPI frame
0x1	-	Reserved
0x2	SPI_ADDR	SPI frame with address
0x3-0xF	-	Reserved

Bits 21:20 – DIPO[1:0]: Data In Pinout
These bits define the data in (DI) pad configurations.

In master operation, DI is MISO.

In slave operation, DI is MOSI.

These bits are not synchronized.

DIPO[1:0]	Name	Description
0x0	PAD[0]	SERCOM PAD[0] is used as data input
0x1	PAD[1]	SERCOM PAD[1] is used as data input
0x2	PAD[2]	SERCOM PAD[2] is used as data input
0x3	PAD[3]	SERCOM PAD[3] is used as data input

Bits 17:16 – DOPO[1:0]: Data Out Pinout
This bit defines the available pad configurations for data out (DO) and the serial clock (SCK). In slave operation, the slave select line ($\overline{SS}$) is controlled by DOPO, while in master operation the $\overline{SS}$ line is controlled by the port configuration.

In master operation, DO is MOSI.

In slave operation, DO is MISO.

These bits are not synchronized.

DOPO	DO	SCK	Slave $\overline{SS}$	Master $\overline{SS}$
0x0	PAD[0]	PAD[1]	PAD[2]	System configuration
0x1	PAD[2]	PAD[3]	PAD[1]	System configuration
0x2	PAD[3]	PAD[1]	PAD[2]	System configuration
0x3	PAD[0]	PAD[3]	PAD[1]	System configuration

Bit 8 – IBON: Immediate Buffer Overflow Notification
This bit controls when the buffer overflow status bit (STATUS.BUFOVF) is set when a buffer overflow occurs.

This bit is not synchronized.

Value	Description
0	STATUS.BUFOVF is set when it occurs in the data stream.
1	STATUS.BUFOVF is set immediately upon buffer overflow.

Bit 7 – RUNSTDBY: Run In Standby
This bit defines the functionality in standby sleep mode.

These bits are not synchronized.

RUNSTDBY	Slave	Master
0x0	Disabled. All reception is dropped, including the ongoing transaction.	Generic clock is disabled when ongoing transaction is finished. All interrupts can wake up the device.
0x1	Ongoing transaction continues, wake on Receive Complete interrupt.	Generic clock is enabled while in sleep modes. All interrupts can wake up the device.

Bits 4:2 – MODE[2:0]: Operating Mode
These bits must be written to 0x2 or 0x3 to select the SPI serial communication interface of the SERCOM.

0x2: SPI slave operation

0x3: SPI master operation

These bits are not synchronized.

Bit 1 – ENABLE: Enable
Due to synchronization, there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRL.ENABLE will read back immediately and the Synchronization Enable Busy bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE is cleared when the operation is complete.

This bit is not enable-protected.

Value	Description
0	The peripheral is disabled or being disabled.
1	The peripheral is enabled or being enabled.

Bit 0 – SWRST: Software Reset
Writing '0' to this bit has no effect.

Writing '1' to this bit resets all registers in the SERCOM, except DBGCTRL, to their initial state, and the SERCOM will be disabled.

Writing "1' to CTRL.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded. Any register write access during the ongoing reset will result in an APB error. Reading any register will return the reset value of the register.

Due to synchronization, there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY. SWRST will both be cleared when the reset is complete.

This bit is not enable-protected.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

33.8.2. Control B

Name: CTRLB
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							RXEN	
Access							R/W	
Reset							0	

Bit	15	14	13	12	11	10	9	8
	AMODE[1:0]		MSSEN				SSDE	
Access	R/W	R/W	R/W				R/W	
Reset	0	0	0				0	

Bit	7	6	5	4	3	2	1	0
		PLOADEN				CHSIZE[2:0]		
Access		R/W				R/W	R/W	R/W
Reset		0				0	0	0

Bit 17 – RXEN: Receiver Enable
Writing '0' to this bit will disable the SPI receiver immediately. The receive buffer will be flushed, data from ongoing receptions will be lost and STATUS.BUFOVF will be cleared.

Writing '1' to CTRLB.RXEN when the SPI is disabled will set CTRLB.RXEN immediately. When the SPI is enabled, CTRLB.RXEN will be cleared, SYNCBUSY.CTRLB will be set and remain set until the receiver is enabled. When the receiver is enabled CTRLB.RXEN will read back as '1'.

Writing '1' to CTRLB.RXEN when the SPI is enabled will set SYNCBUSY.CTRLB, which will remain set until the receiver is enabled, and CTRLB.RXEN will read back as '1'.

This bit is not enable-protected.

Value	Description
0	The receiver is disabled or being enabled.
1	The receiver is enabled or it will be enabled when SPI is enabled.

Bits 15:14 – AMODE[1:0]: Address Mode
These bits set the slave addressing mode when the frame format (CTRLA.FORM) with address is used. They are unused in master mode.

AMODE[1:0]	Name	Description
0x0	MASK	ADDRMASK is used as a mask to the ADDR register
0x1	2_ADDRS	The slave responds to the two unique addresses in ADDR and ADDRMASK
0x2	RANGE	The slave responds to the range of addresses between and including ADDR and ADDRMASK. ADDR is the upper limit
0x3	-	Reserved

Bit 13 – MSSEN: Master Slave Select Enable
This bit enables hardware slave select ($\overline{SS}$) control.

Value	Description
0	Hardware $\overline{SS}$ control is disabled.
1	Hardware $\overline{SS}$ control is enabled.

Bit 9 – SSDE: Slave Select Low Detect Enable
This bit enables wake up when the slave select ($\overline{SS}$) pin transitions from high to low.

Value	Description
0	$\overline{SS}$ low detector is disabled.
1	$\overline{SS}$ low detector is enabled.

Bit 6 – PLOADEN: Slave Data Preload Enable
Setting this bit will enable preloading of the slave shift register when there is no transfer in progress. If the $\overline{SS}$ line is high when DATA is written, it will be transferred immediately to the shift register.

Bits 2:0 – CHSIZE[2:0]: Character Size

CHSIZE[2:0]	Name	Description
0x0	8BIT	8 bits
0x1	9BIT	9 bits
0x2-0x7	-	Reserved

33.8.3. Baud Rate

Name: BAUD
Offset: 0x0C
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
	BAUD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – BAUD[7:0]: Baud Register
These bits control the clock generation, as described in the *SERCOM Clock Generation – Baud-Rate Generator*.

33.8.4. Interrupt Enable Clear

Name: INTENCLR
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	ERROR				SSL	RXC	TXC	DRE
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit 7 – ERROR: Error Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Error Interrupt Enable bit, which disables the Error interrupt.

Value	Description
0	Error interrupt is disabled.
1	Error interrupt is enabled.

Bit 3 – SSL: Slave Select Low Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Slave Select Low Interrupt Enable bit, which disables the Slave Select Low interrupt.

Value	Description
0	Slave Select Low interrupt is disabled.
1	Slave Select Low interrupt is enabled.

Bit 2 – RXC: Receive Complete Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Receive Complete Interrupt Enable bit, which disables the Receive Complete interrupt.

Value	Description
0	Receive Complete interrupt is disabled.
1	Receive Complete interrupt is enabled.

Bit 1 – TXC: Transmit Complete Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Transmit Complete Interrupt Enable bit, which disable the Transmit Complete interrupt.

Value	Description
0	Transmit Complete interrupt is disabled.
1	Transmit Complete interrupt is enabled.

Bit 0 – DRE: Data Register Empty Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Data Register Empty Interrupt Enable bit, which disables the Data Register Empty interrupt.

Value	Description
0	Data Register Empty interrupt is disabled.
1	Data Register Empty interrupt is enabled.

33.8.5. Interrupt Enable Set

Name: INTENSET
Offset: 0x16
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	ERROR				SSL	RXC	TXC	DRE
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit 7 – ERROR: Error Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Error Interrupt Enable bit, which enables the Error interrupt.

Value	Description
0	Error interrupt is disabled.
1	Error interrupt is enabled.

Bit 3 – SSL: Slave Select Low Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Slave Select Low Interrupt Enable bit, which enables the Slave Select Low interrupt.

Value	Description
0	Slave Select Low interrupt is disabled.
1	Slave Select Low interrupt is enabled.

Bit 2 – RXC: Receive Complete Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Receive Complete Interrupt Enable bit, which enables the Receive Complete interrupt.

Value	Description
0	Receive Complete interrupt is disabled.
1	Receive Complete interrupt is enabled.

Bit 1 – TXC: Transmit Complete Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Transmit Complete Interrupt Enable bit, which enables the Transmit Complete interrupt.

Value	Description
0	Transmit Complete interrupt is disabled.
1	Transmit Complete interrupt is enabled.

Bit 0 – DRE: Data Register Empty Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Data Register Empty Interrupt Enable bit, which enables the Data Register Empty interrupt.

Value	Description
0	Data Register Empty interrupt is disabled.
1	Data Register Empty interrupt is enabled.

33.8.6. Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x18
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	ERROR				SSL	RXC	TXC	DRE
Access	R/W				R/W	R	R/W	R
Reset	0				0	0	0	0

Bit 7 – ERROR: Error
This flag is cleared by writing '1' to it.

This bit is set when any error is detected. Errors that will set this flag have corresponding status flags in the STATUS register. The BUFOVF error will set this interrupt flag.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the flag.

Bit 3 – SSL: Slave Select Low
This flag is cleared by writing '1' to it.

This bit is set when a high to low transition is detected on the _SS pin in slave mode and Slave Select Low Detect (CTRLB.SSDE) is enabled.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the flag.

Bit 2 – RXC: Receive Complete
This flag is cleared by reading the Data (DATA) register or by disabling the receiver.

This flag is set when there are unread data in the receive buffer. If address matching is enabled, the first data received in a transaction will be an address.

Writing '0' to this bit has no effect.

Writing '1' to this bit has no effect.

Bit 1 – TXC: Transmit Complete
This flag is cleared by writing '1' to it or by writing new data to DATA.

In master mode, this flag is set when the data have been shifted out and there are no new data in DATA.

In slave mode, this flag is set when the _SS pin is pulled high. If address matching is enabled, this flag is only set if the transaction was initiated with an address match.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the flag.

Bit 0 – DRE: Data Register Empty
This flag is cleared by writing new data to DATA.

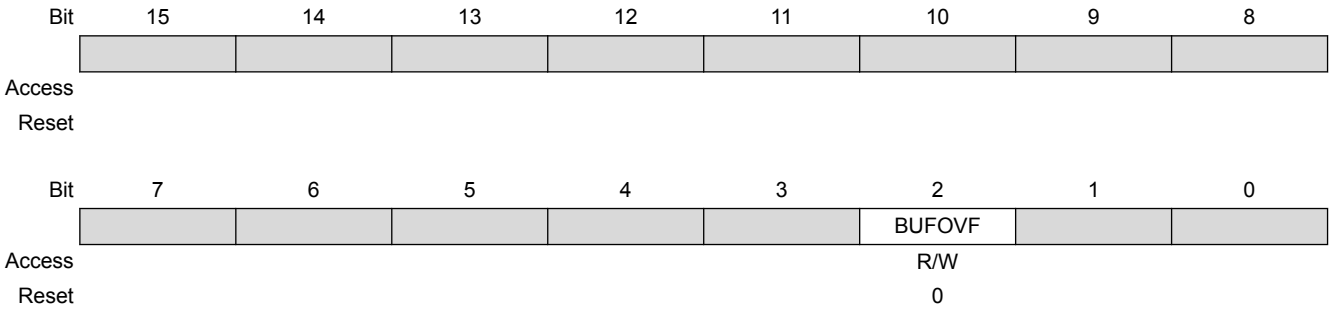
This flag is set when DATA is empty and ready for new data to transmit.

Writing '0' to this bit has no effect.

Writing '1' to this bit has no effect.

33.8.7. Status

Name: STATUS
Offset: 0x1A
Reset: 0x0000
Property: –



Bit 2 – BUFOVF: Buffer Overflow
Reading this bit before reading DATA will indicate the error status of the next character to be read.

This bit is cleared by writing '1' to the bit or by disabling the receiver.

This bit is set when a buffer overflow condition is detected. See also CTRLA.IBON for overflow handling.

When set, the corresponding RxDATA will be zero.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear it.

Value	Description
0	No Buffer Overflow has occurred.
1	A Buffer Overflow has occurred.

33.8.8. Synchronization Busy

Name: SYNCBUSY
Offset: 0x1C
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						CTRLB	ENABLE	SWRST
Access						R	R	R
Reset						0	0	0

Bit 2 – CTRLB: CTRLB Synchronization Busy
Writing to the CTRLB when the SERCOM is enabled requires synchronization. Ongoing synchronization is indicated by SYNCBUSY.CTRLB=1 until synchronization is complete. If CTRLB is written while SYNCBUSY.CTRLB=1, an APB error will be generated.

Value	Description
0	CTRLB synchronization is not busy.
1	CTRLB synchronization is busy.

Bit 1 – ENABLE: SERCOM Enable Synchronization Busy
Enabling and disabling the SERCOM (CTRLA.ENABLE) requires synchronization. Ongoing synchronization is indicated by SYNCBUSY.ENABLE=1 until synchronization is complete.

Writes to any register (except for CTRLA.SWRST) while enable synchronization is on-going will be discarded and an APB error will be generated.

Value	Description
0	Enable synchronization is not busy.
1	Enable synchronization is busy.

Bit 0 – SWRST: Software Reset Synchronization Busy

Resetting the SERCOM (CTRLA.SWRST) requires synchronization. Ongoing synchronization is indicated by SYNCBUSY.SWRST=1 until synchronization is complete.

Writes to any register while synchronization is on-going will be discarded and an APB error will be generated.

Value	Description
0	SWRST synchronization is not busy.
1	SWRST synchronization is busy.

33.8.9. Address

Name: ADDR
Offset: 0x24
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	ADDRMASK[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – ADDRMASK[7:0]: Address Mask
These bits hold the address mask when the transaction format with address is used (CTRLA.FORM, CTRLB.AMODE).

Bits 7:0 – ADDR[7:0]: Address
These bits hold the address when the transaction format with address is used (CTRLA.FORM, CTRLB.AMODE).

33.8.10. Data

Name: DATA
Offset: 0x28
Reset: 0x0000
Property: –

Bit	15	14	13	12	11	10	9	8
								DATA[8:8]
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 8:0 – DATA[8:0]: Data
Reading these bits will return the contents of the receive data buffer. The register should be read only when the Receive Complete Interrupt Flag bit in the Interrupt Flag Status and Clear register (INTFLAG.RXC) is set.

Writing these bits will write the transmit data buffer. This register should be written only when the Data Register Empty Interrupt Flag bit in the Interrupt Flag Status and Clear register (INTFLAG.DRE) is set.

33.8.11. Debug Control

Name: DBGCTRL
Offset: 0x30
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGSTOP
Access								R/W
Reset								0

Bit 0 – DBGSTOP: Debug Stop Mode
This bit controls the functionality when the CPU is halted by an external debugger.

Value	Description
0	The baud-rate generator continues normal operation when the CPU is halted by an external debugger.
1	The baud-rate generator is halted when the CPU is halted by an external debugger.

34. SERCOM I²C – SERCOM Inter-Integrated Circuit

34.1. Overview

The inter-integrated circuit (I²C) interface is one of the available modes in the serial communication interface (SERCOM).

The I²C interface uses the SERCOM transmitter and receiver configured as shown in Figure 34-1 I2C Single-Master Single-Slave Interconnection on page 640. Labels in capital letters are registers accessible by the CPU, while lowercase labels are internal to the SERCOM. Each master and slave have a separate I²C interface containing a shift register, a transmit buffer and a receive buffer. In addition, the I²C master uses the SERCOM baud-rate generator, while the I²C slave uses the SERCOM address match logic.

Related Links

SERCOM – Serial Communication Interface on page 560

34.2. Features

SERCOM I²C includes the following features:

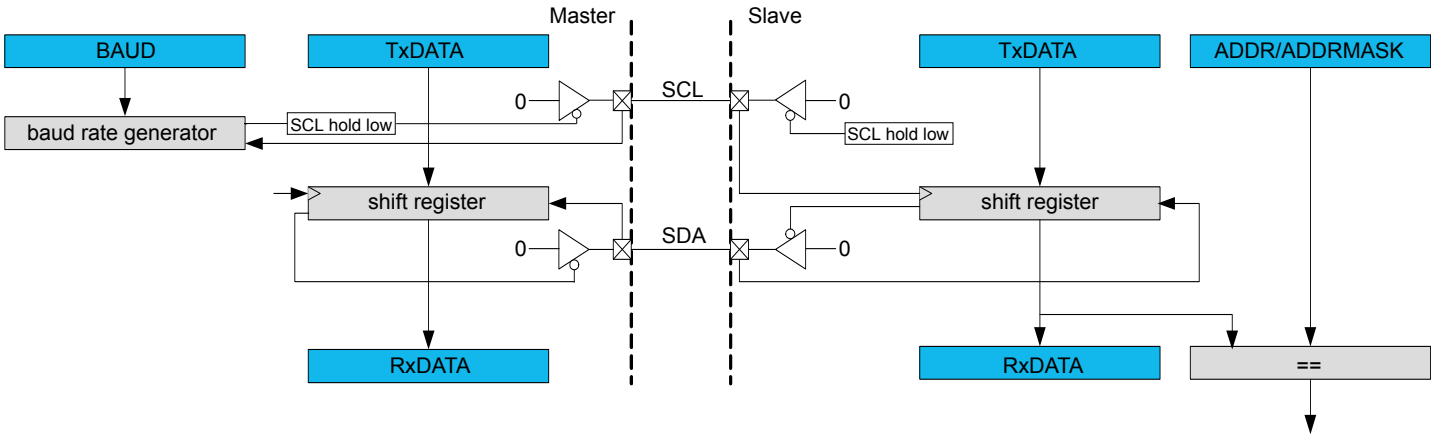
- Master or slave operation
- Can be used with DMA
- Philips I²C compatible
- SMBus™ compatible
- PMBus compatible
- Support of 100kHz and 400kHz, 1MHz and 3.4MHz I²C mode low system clock frequencies
- Physical interface includes:
  - Slew-rate limited outputs
  - Filtered inputs
- Slave operation:
  - Operation in all sleep modes
  - Wake-up on address match
  - 7-bit and 10-bit Address match in hardware for:
  - - Unique address and/or 7-bit general call address
    - Address range
    - Two unique addresses can be used with DMA

Related Links

Features on page 560

34.3. Block Diagram

Figure 34-1. I²C Single-Master Single-Slave Interconnection



34.4. Signal Description

Signal Name	Type	Description
PAD[0]	Digital I/O	SDA
PAD[1]	Digital I/O	SCL
PAD[2]	Digital I/O	SDA_OUT (4-wire)
PAD[3]	Digital I/O	SDC_OUT (4-wire)

One signal can be mapped on several pins.

Not all the pins are I²C pins.

Related Links

I/O Multiplexing and Considerations on page 29

34.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

34.5.1. I/O Lines

In order to use the I/O lines of this peripheral, the I/O pins must be configured using the I/O Pin Controller (PORT).

When the SERCOM is used in I²C mode, the SERCOM controls the direction and value of the I/O pins. Both PORT control bits PINCFGn.PULLEN and PINCFGn.DRVSTR are still effective. If the receiver or transmitter is disabled, these pins can be used for other purposes.

Related Links

PORT: IO Pin Controller on page 506

34.5.2. Power Management

This peripheral can continue to operate in any sleep mode where its source clock is running. The interrupts can wake up the device from sleep modes. Refer to *PM – Power Manager* for details on the different sleep modes.

Related Links

PM – Power Manager on page 186

34.5.3. Clocks

The SERCOM bus clock (CLK_SERCOMx_APB) is enabled by default, and can be enabled and disabled in the Main Clock Controller and the Power Manager.

Two generic clocks ared used by SERCOM, GCLK_SERCOMx_CORE and GCLK_SERCOM_SLOW. The core clock (GCLK_SERCOMx_CORE) can clock the I^2C when working as a master. The slow clock (GCLK_SERCOM_SLOW) is required only for certain functions, e.g. SMBus timing. These clocks must be configured and enabled in the Generic Clock Controller (GCLK) before using the I^2C.

These generic clocks are asynchronous to the bus clock (CLK_SERCOMx_APB). Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains. Refer to Synchronization on page 659 for further details.

Related Links

GCLK - Generic Clock Controller on page 131

Peripheral Clock Masking on page 151

PM – Power Manager on page 186

34.5.4. DMA

The DMA request lines are connected to the DMA Controller (DMAC). In order to use DMA requests with this peripheral the DMAC must be configured first. Refer to *DMAC – Direct Memory Access Controller* for details.

Related Links

DMAC – Direct Memory Access Controller on page 399

34.5.5. Interrupts

The interrupt request line is connected to the Interrupt Controller. In order to use interrupt requests of this peripheral, the Interrupt Controller (NVIC) must be configured first. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

Nested Vector Interrupt Controller on page 51

34.5.6. Events

Not applicable.

34.5.7. Debug Operation

When the CPU is halted in debug mode, this peripheral will continue normal operation. If the peripheral is configured to require periodical service by the CPU through interrupts or similar, improper operation or data loss may result during debugging. This peripheral can be forced to halt operation during debugging - refer to the Debug Control (DBGCTRL) register for details.

Refer to the DBGCTRL on page 703 register for details.

34.5.8. Register Access Protection

Registers with write-access can be write-protected optionally by the peripheral access controller (PAC).

PAC Write-Protection is not available for the following registers:

- Interrupt Flag Clear and Status register (INTFLAG)
- Status register (STATUS)
- Data register (DATA)
- Address register (ADDR)

Optional PAC Write-Protection is denoted by the "PAC Write-Protection" property in each individual register description.

Write-protection does not apply to accesses through an external debugger.

Related Links

PAC - Peripheral Access Controller on page 58

34.5.9. Analog Connections

Not applicable.

34.6. Functional Description

34.6.1. Principle of Operation

The I^2C interface uses two physical lines for communication:

- Serial Data Line (SDA) for packet transfer
- Serial Clock Line (SCL) for the bus clock

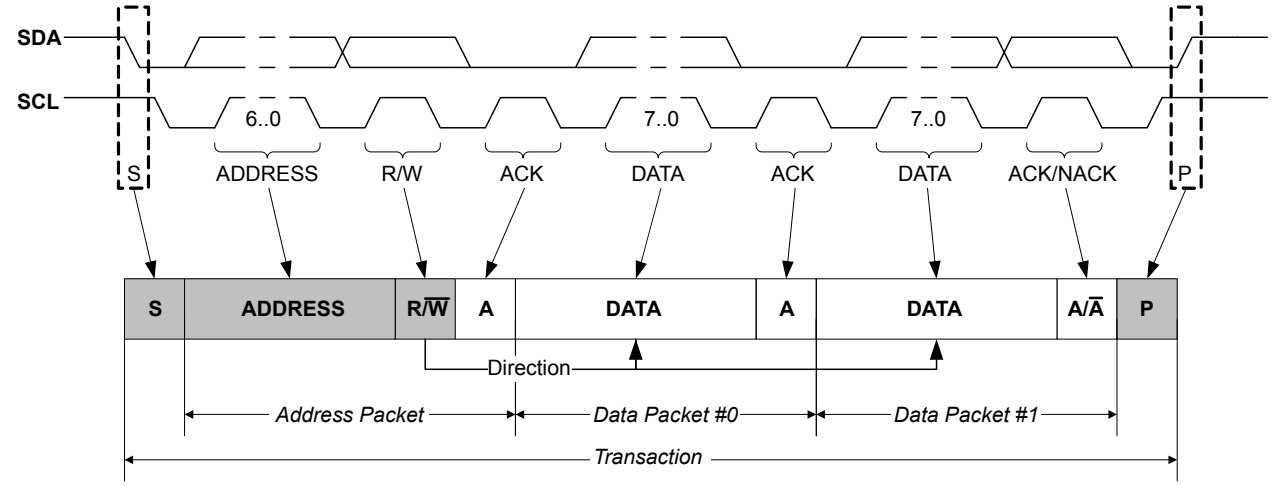
A transaction starts with the I^2C master sending the start condition, followed by a 7-bit address and a direction bit (read or write to/from the slave).

The addressed I^2C slave will then acknowledge (ACK) the address, and data packet transactions can begin. Every 9-bit data packet consists of 8 data bits followed by a one-bit reply indicating whether the data was acknowledged or not.

If a data packet is not acknowledged (NACK), whether by the I^2C slave or master, the I^2C master takes action by either terminating the transaction by sending the stop condition, or by sending a repeated start to transfer more data.

The figure below illustrates the possible transaction formats and Transaction Diagram Symbols explains the transaction symbols. These symbols will be used in the following descriptions.

Figure 34-2. Basic I²C Transaction Diagram



Transaction Diagram Symbols

Bus Driver

- Master driving bus
- Slave driving bus
- Either Master or Slave driving bus

Special Bus Conditions

- **S** START condition
- **Sr** repeated START condition
- **P** STOP condition

Data Package Direction

- **R** Master Read '1'
- **W** Master Write '0'

Acknowledge

- **A** Acknowledge (ACK) '0'
- **Ā** Not Acknowledge (NACK) '1'

34.6.2. Basic Operation

34.6.2.1. Initialization

The following registers are enable-protected, meaning they can be written only when the I²C interface is disabled (CTRLA.ENABLE is '0'):

- Control A register (CTRLA), except Enable (CTRLA.ENABLE) and Software Reset (CTRLA.SWRST) bits
- Control B register (CTRLB), except Acknowledge Action (CTRLB.ACKACT) and Command (CTRLB.CMD) bits
- Baud register (BAUD)
- Address register (ADDR) in slave operation.

When the I²C is enabled or is being enabled (CTRLA.ENABLE=1), writing to these registers will be discarded. If the I²C is being disabled, writing to these registers will be completed after the disabling.

Enable-protection is denoted by the "Enable-Protection" property in the register description.

Before the I²C is enabled it must be configured as outlined by the following steps:

1. Select I²C Master or Slave mode by writing 0x4 or 0x5 to the Operating Mode bits in the CTRLA register (CTRLA.MODE).
2. If desired, select the SDA Hold Time value in the CTRLA register (CTRLA.SDAHOLD).
3. If desired, enable smart operation by setting the Smart Mode Enable bit in the CTRLB register (CTRLB.SMEN).
4. If desired, enable SCL low time-out by setting the SCL Low Time-Out bit in the Control A register (CTRLA.LOWTOUT).
5. In Master mode:
   5.1. Select the inactive bus time-out in the Inactive Time-Out bit group in the CTRLA register (CTRLA.INACTOUT).
   5.2. Write the Baud Rate register (BAUD) to generate the desired baud rate.

   In Slave mode:
   5.1. Configure the address match configuration by writing the Address Mode value in the CTRLB register (CTRLB.AMODE).
   5.2. Set the Address and Address Mask value in the Address register (ADDR.ADDR and ADDR.ADDRMASK) according to the address configuration.

34.6.2.2. Enabling, Disabling, and Resetting

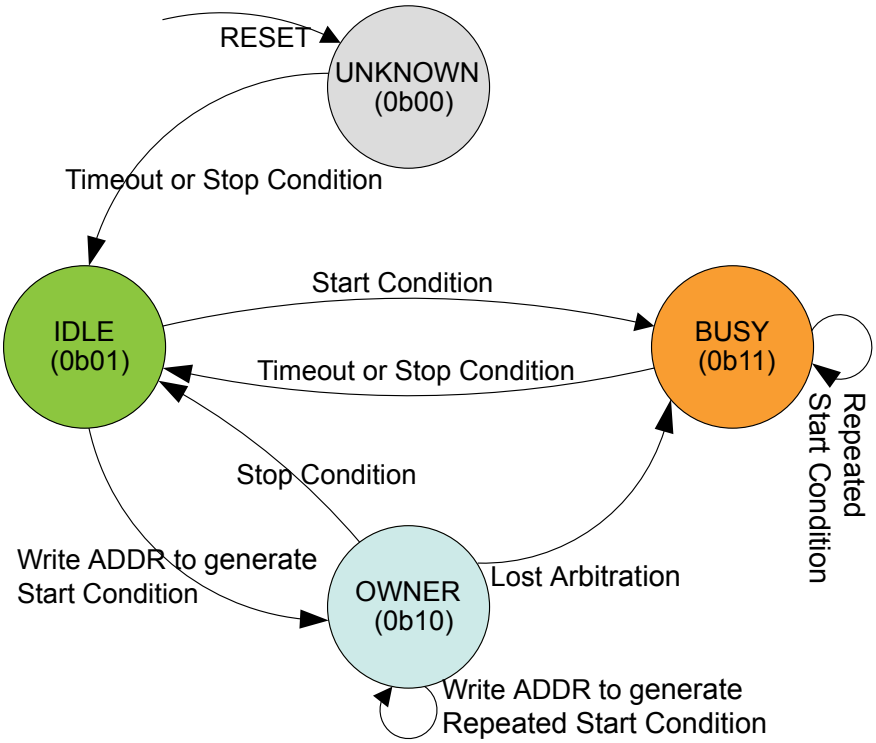
This peripheral is enabled by writing '1' to the Enable bit in the Control A register (CTRLA.ENABLE), and disabled by writing '0' to it.

Refer to CTRLA on page 683 for details.

34.6.2.3. I²C Bus State Logic

The bus state logic includes several logic blocks that continuously monitor the activity on the I²C bus lines in all sleep modes. The start and stop detectors and the bit counter are all essential in the process of determining the current bus state. The bus state is determined according to Bus State Diagram. Software can get the current bus state by reading the Master Bus State bits in the Status register (STATUS.BUSSTATE). The value of STATUS.BUSSTATE in the figure is shown in binary.

Figure 34-3. Bus State Diagram



The bus state machine is active when the I²C master is enabled.

After the I²C master has been enabled, the bus state is UNKNOWN (0b00). From the UNKNOWN state, the bus will transition to IDLE (0b01) by either:

- Forcing by by writing 0b01 to STATUS.BUSSTATE
- A stop condition is detected on the bus
- If the inactive bus time-out is configured for SMBus compatibility (CTRLA.INACTOUT) and a time-out occurs.

Note: Once a known bus state is established, the bus state logic will not re-enter the UNKNOWN state.

When the bus is IDLE it is ready for a new transaction. If a start condition is issued on the bus by another I²C master in a multi-master setup, the bus becomes BUSY (0b11). The bus will re-enter IDLE either when a stop condition is detected, or when a time-out occurs (inactive bus time-out needs to be configured).

If a start condition is generated internally by writing the Address bit group in the Address register (ADDR.ADDR) while IDLE, the OWNER state (0b10) is entered. If the complete transaction was performed without interference, i.e., arbitration was not lost, the I²C master can issue a stop condition, which will change the bus state back to IDLE.

However, if a packet collision is detected while in OWNER state, the arbitration is assumed lost and the bus state becomes BUSY until a stop condition is detected. A repeated start condition will change the bus state only if arbitration is lost while issuing a repeated start.

Regardless of winning or losing arbitration, the entire address will be sent. If arbitration is lost, only 'ones' are transmitted from the point of losing arbitration and the rest of the address length.

34.6.2.4. I²C Master Operation

The I²C master is byte-oriented and interrupt based. The number of interrupts generated is kept at a minimum by automatic handling of most events. The software driver complexity and code size are

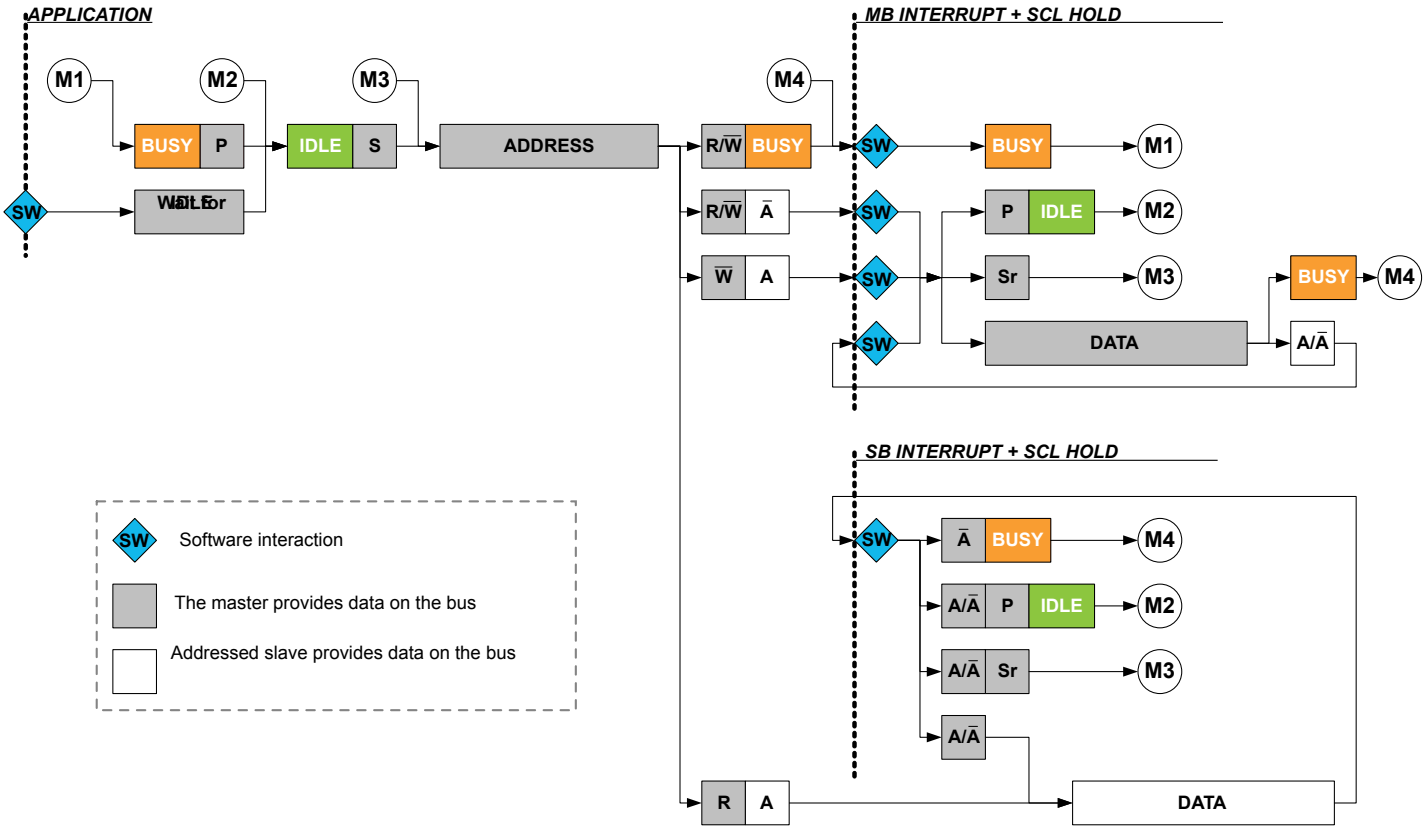
reduced by auto-triggering of operations, and a special smart mode, which can be enabled by the Smart Mode Enable bit in the Control A register (CTRLA.SMEN).

The I²C master has two interrupt strategies.

When SCL Stretch Mode (CTRLA.SCLSM) is '0', SCL is stretched before or after the acknowledge bit . In this mode the I²C master operates according to Master Behavioral Diagram (SCLSM=0). The circles labelled "M*n*" (M1, M2..) indicate the nodes the bus logic can jump to, based on software or hardware interaction.

This diagram is used as reference for the description of the I²C master operation throughout the document.

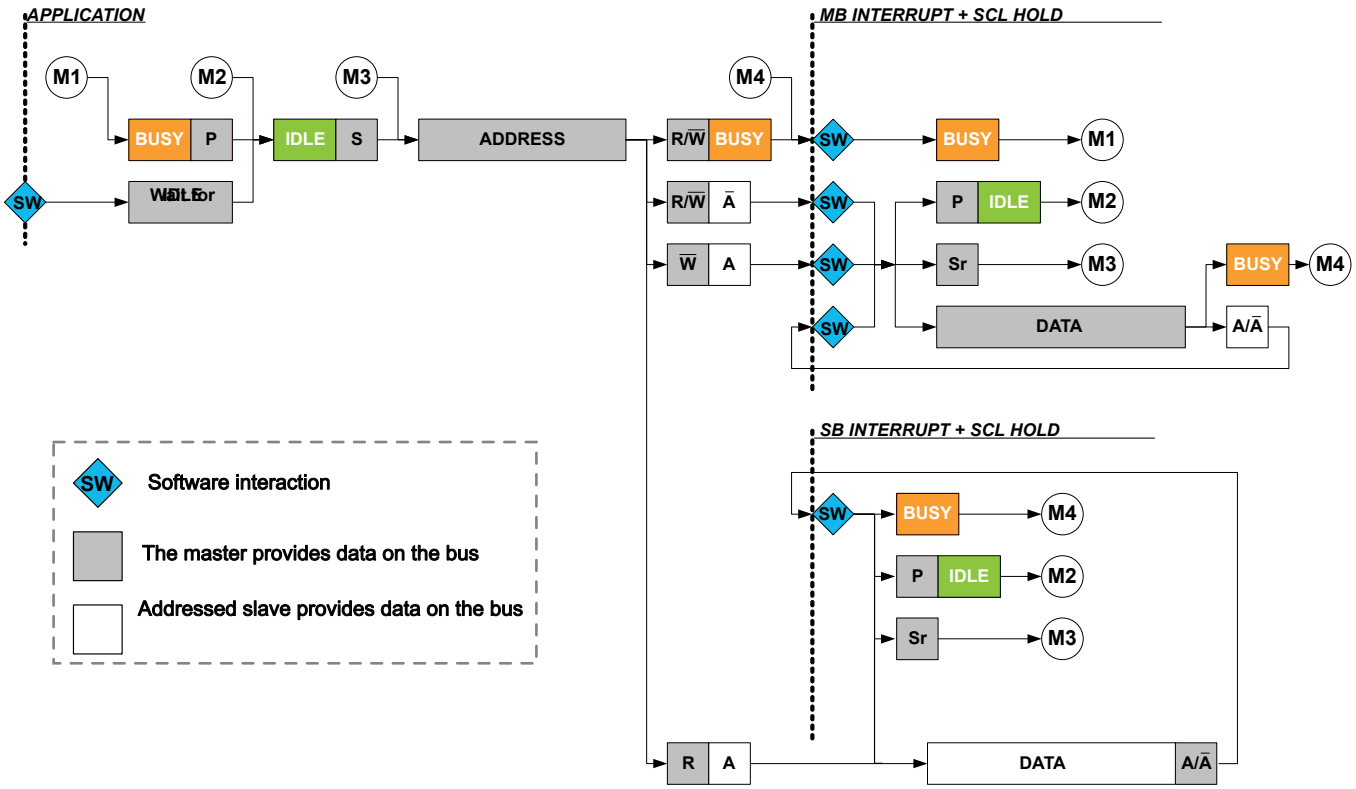
Figure 34-4. I²C Master Behavioral Diagram (SCLSM=0)



In the second strategy (CTRLA.SCLSM=1), interrupts only occur after the ACK bit, as in Master Behavioral Diagram (SCLSM=1). This strategy can be used when it is not necessary to check DATA before acknowledging.

Note: I²C High-speed (*Hs*) mode requires CTRLA.SCLSM=1.

Figure 34-5. I²C Master Behavioral Diagram (SCLSM=1)



Master Clock Generation

The SERCOM peripheral supports several I²C bi-directional modes:

- Standard mode (*Sm*) up to 100kHz
- Fast mode (*Fm*) up to 400kHz
- Fast mode Plus (*Fm+*) up to 1MHz
- High-speed mode (*Hs*) up to 3.4MHz

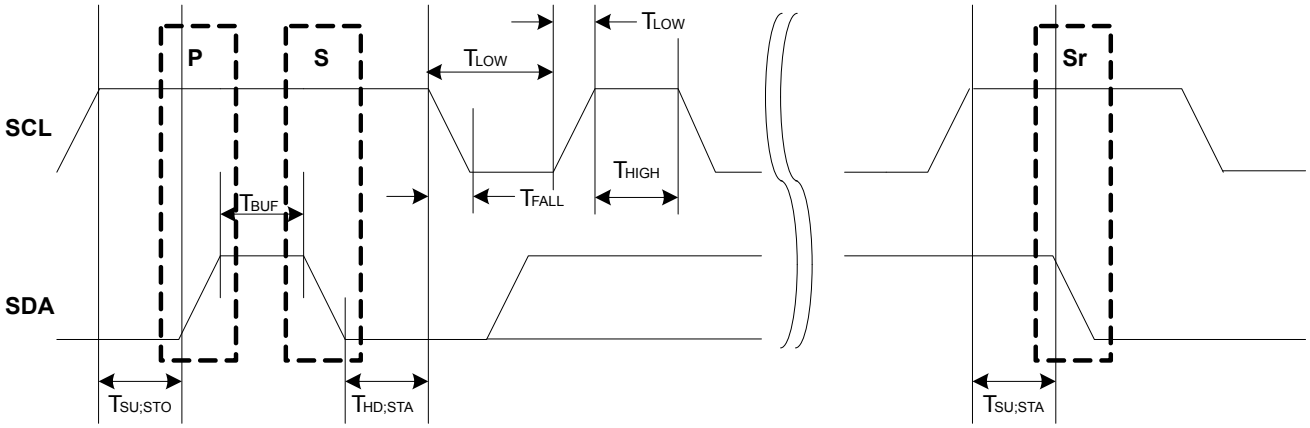
The Master clock configuration for *Sm, Fm,* and *Fm+* are described in Clock Generation (Standard-Mode, Fast-Mode, and Fast-Mode Plus) on page 647. For *Hs*, refer to Master Clock Generation (High-Speed Mode) on page 649.

Clock Generation (Standard-Mode, Fast-Mode, and Fast-Mode Plus)

In I²C *Sm, Fm*, and *Fm+* mode, the Master clock (SCL) frequency is determined as described in this section:

The low (T_{LOW}) and high (T_{HIGH}) times are determined by the Baud Rate register (BAUD), while the rise (T_{RISE}) and fall (T_{FALL}) times are determined by the bus topology. Because of the wired-AND logic of the bus, T_{FALL} will be considered as part of T_{LOW}. Likewise, T_{RISE} will be in a state between T_{LOW} and T_{HIGH} until a high state has been detected.

Figure 34-6. SCL Timing



The following parameters are timed using the SCL low time period T_{LOW}. This comes from the Master Baud Rate Low bit group in the Baud Rate register (BAUD.BAUDLOW). When BAUD.BAUDLOW=0, or the Master Baud Rate bit group in the Baud Rate register (BAUD.BAUD) determines it.

- T_{LOW} – Low period of SCL clock
- $T_{SU;STO}$ – Set-up time for stop condition
- T_{BUF} – Bus free time between stop and start conditions
- $T_{HD;STA}$ – Hold time (repeated) start condition
- $T_{SU;STA}$ – Set-up time for repeated start condition
- T_{HIGH} is timed using the SCL high time count from BAUD.BAUD
- T_{RISE} is determined by the bus impedance; for internal pull-ups. Refer to *Electrical Characteristics*.
- T_{FALL} is determined by the open-drain current limit and bus impedance; can typically be regarded as zero. Refer to *Electrical Characteristics* for details.

The SCL frequency is given by:

$$f_{SCL} = \frac{1}{T_{LOW} + T_{HIGH} + T_{RISE}}$$

When BAUD.BAUDLOW is zero, the BAUD.BAUD value is used to time both SCL high and SCL low. In this case the following formula will give the SCL frequency:

$$f_{SCL} = \frac{f_{GCLK}}{10 + 2BAUD + f_{GCLK} \cdot T_{RISE}}$$

When BAUD.BAUDLOW is non-zero, the following formula determines the SCL frequency:

$$f_{SCL} = \frac{f_{GCLK}}{10 + BAUD + BAUDLOW + f_{GCLK} \cdot T_{RISE}}$$

The following formulas can determine the SCL T_{LOW} and T_{HIGH} times:

$$T_{LOW} = \frac{BAUDLOW + 5}{f_{GCLK}}$$

$$T_{HIGH} = \frac{BAUD + 5}{f_{GCLK}}$$

Note: The I²C standard *Fm+* (Fast-mode plus) requires a nominal high to low SCL ratio of 1:2, and BAUD should be set accordingly. At a minimum, BAUD.BAUD and/or BAUD.BAUDLOW must be non-zero.

Startup Timing The minimum time between SDA transition and SCL rising edge is 6 APB cycles when the DATA register is written in smart mode. If a greater startup time is required due to long rise times, the time between DATA write and IF clear must be controlled by software.
Note: When timing is controlled by user, the Smart Mode cannot be enabled.

Related Links

Electrical Characteristics on page 1140

Master Clock Generation (High-Speed Mode)

For I²C *Hs* transfers, there is no SCL synchronization. Instead, the SCL frequency is determined by the GCLK_SERCOMx_CORE frequency (f_{GCLK}) and the High-Speed Baud setting in the Baud register (BAUD.HSBAUD). When BAUD.HSBAUDLOW=0, the HSBAUD value will determine both SCL high and SCL low. In this case the following formula determines the SCL frequency.

$$f_{SCL} = \frac{f_{GCLK}}{2 + 2 \cdot HS\,BAUD}$$

When HSBAUDLOW is non-zero, the following formula determines the SCL frequency.

$$f_{SCL} = \frac{f_{GCLK}}{2 + HS\,BAUD + HSBAUDLOW}$$

Note: The I²C standard *Hs* (High-speed) requires a nominal high to low SCL ratio of 1:2, and HSBAUD should be set accordingly. At a minimum, BAUD.HSBAUD and/or BAUD.HSBAUDLOW must be non-zero.

Transmitting Address Packets

The I²C master starts a bus transaction by writing the I²C slave address to ADDR.ADDR and the direction bit, as described in Principle of Operation on page 642. If the bus is busy, the I²C master will wait until the bus becomes idle before continuing the operation. When the bus is idle, the I²C master will issue a start condition on the bus. The I²C master will then transmit an address packet using the address written to ADDR.ADDR. After the address packet has been transmitted by the I²C master, one of four cases will arise according to arbitration and transfer direction.

Case 1: Arbitration lost or bus error during address packet transmission

If arbitration was lost during transmission of the address packet, the Master on Bus bit in the Interrupt Flag Status and Clear register (INTFLAG.MB) and the Arbitration Lost bit in the Status register (STATUS.ARBLOST) are both set. Serial data output to SDA is disabled, and the SCL is released, which disables clock stretching. In effect the I²C master is no longer allowed to execute any operation on the bus until the bus is idle again. A bus error will behave similarly to the arbitration lost condition. In this case, the MB interrupt flag and Master Bus Error bit in the Status register (STATUS.BUSERR) are both set in addition to STATUS.ARBLOST.

The Master Received Not Acknowledge bit in the Status register (STATUS.RXNACK) will always contain the last successfully received acknowledge or not acknowledge indication.

In this case, software will typically inform the application code of the condition and then clear the interrupt flag before exiting the interrupt routine. No other flags have to be cleared at this moment, because all flags will be cleared automatically the next time the ADDR.ADDR register is written.

Case 2: Address packet transmit complete – No ACK received

If there is no I²C slave device responding to the address packet, then the INTFLAG.MB interrupt flag and STATUS.RXNACK will be set. The clock hold is active at this point, preventing further activity on the bus.

The missing ACK response can indicate that the I²C slave is busy with other tasks or sleeping. Therefore, it is not able to respond. In this event, the next step can be either issuing a stop condition (recommended)

or resending the address packet by a repeated start condition. When using SMBus logic, the slave must ACK the address. If there is no response, it means that the slave is not available on the bus.

Case 3: Address packet transmit complete – Write packet, Master on Bus set

If the I²C master receives an acknowledge response from the I²C slave, INTFLAG.MB will be set and STATUS.RXNACK will be cleared. The clock hold is active at this point, preventing further activity on the bus.

In this case, the software implementation becomes highly protocol dependent. Three possible actions can enable the I²C operation to continue:

- Initiate a data transmit operation by writing the data byte to be transmitted into DATA.DATA.
- Transmit a new address packet by writing ADDR.ADDR. A repeated start condition will automatically be inserted before the address packet.
- Issue a stop condition, consequently terminating the transaction.

Case 4: Address packet transmit complete – Read packet, Slave on Bus set

If the I²C master receives an ACK from the I²C slave, the I²C master proceeds to receive the next byte of data from the I²C slave. When the first data byte is received, the Slave on Bus bit in the Interrupt Flag register (INTFLAG.SB) will be set and STATUS.RXNACK will be cleared. The clock hold is active at this point, preventing further activity on the bus.

In this case, the software implementation becomes highly protocol dependent. Three possible actions can enable the I²C operation to continue:

- Let the I²C master continue to read data by acknowledging the data received. ACK can be sent by software, or automatically in smart mode.
- Transmit a new address packet.
- Terminate the transaction by issuing a stop condition.

Note: An ACK or NACK will be automatically transmitted if smart mode is enabled. The Acknowledge Action bit in the Control B register (CTRLB.ACKACT) determines whether ACK or NACK should be sent.

Transmitting Data Packets

When an address packet with direction Master Write (see Figure 34-2 Basic I2C Transaction Diagram on page 643) was transmitted successfully , INTFLAG.MB will be set. The I²C master will start transmitting data via the I²C bus by writing to DATA.DATA, and monitor continuously for packet collisions. I

If a collision is detected, the I²C master will lose arbitration and STATUS.ARBLOST will be set. If the transmit was successful, the I²C master will receive an ACK bit from the I²C slave, and STATUS.RXNACK will be cleared. INTFLAG.MB will be set in both cases, regardless of arbitration outcome.

It is recommended to read STATUS.ARBLOST and handle the arbitration lost condition in the beginning of the I²C Master on Bus interrupt. This can be done as there is no difference between handling address and data packet arbitration.

STATUS.RXNACK must be checked for each data packet transmitted before the next data packet transmission can commence. The I²C master is not allowed to continue transmitting data packets if a NACK is received from the I²C slave.

Receiving Data Packets (SCLSM=0)

When INTFLAG.SB is set, the I²C master will already have received one data packet. The I²C master must respond by sending either an ACK or NACK. Sending a NACK may be unsuccessful when arbitration is lost during the transmission. In this case, a lost arbitration will prevent setting INTFLAG.SB. Instead, INTFLAG.MB will indicate a change in arbitration. Handling of lost arbitration is the same as for data bit transmission.

Receiving Data Packets (SCLSM=1)

When INTFLAG.SB is set, the I²C master will already have received one data packet and transmitted an ACK or NACK, depending on CTRLB.ACKACT. At this point, CTRLB.ACKACT must be set to the correct value for the next ACK bit, and the transaction can continue by reading DATA and issuing a command if not in the smart mode.

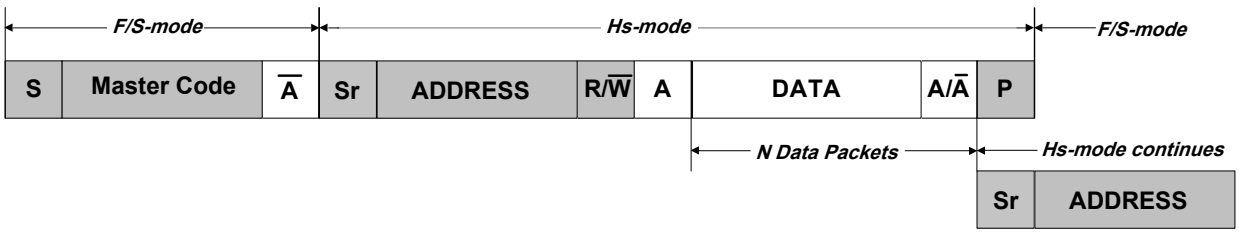
High-Speed Mode

High-speed transfers are a multi-step process, see High Speed Transfer.

First, a master code (0b00001nnn, where 'nnn' is a unique master code) is transmitted in Full-speed mode, followed by a NACK since no slaveshould acknowledge. Arbitration is performed only during the Full-speed Master Code phase. The master code is transmitted by writing the master code to the address register (ADDR.ADDR) and writing the high-speed bit (ADDR.HS) to '0'.

After the master code and NACK have been transmitted, the master write interrupt will be asserted. In the meanwhile, the slave address can be written to the ADDR.ADDR register together with ADDR.HS=1. Now in High-speed mode, the master will generate a repeated start, followed by the slave address with RW-direction. The bus will remain in High-speed mode until a stop is generated. If a repeated start is desired, the ADDR.HS bit must again be written to '1', along with the new address ADDR.ADDR to be transmitted.

Figure 34-7. High Speed Transfer



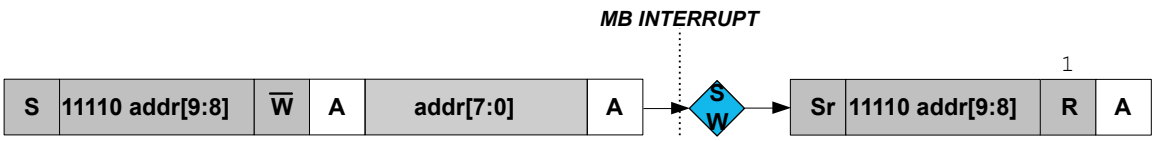
Transmitting in High-speed mode requires the I²C master to be configured in High-speed mode (CTRLA.SPEED=0x2) and the SCL clock stretch mode (CTRLA.SCLSM) bit set to '1'.

10-Bit Addressing

When 10-bit addressing is enabled by the Ten Bit Addressing Enable bit in the Address register (ADDR.TENBITEN=1) and the Address bit field ADDR.ADDR is written, the two address bytes will be transmitted, see 10-bit Address Transmission for a Read Transaction. The addressed slave acknowledges the two address bytes, and the transaction continues. Regardless of whether the transaction is a read or write, the master must start by sending the 10-bit address with the direction bit (ADDR.ADDR[0]) being zero.

If the master receives a NACK after the first byte, the write interrupt flag will be raised and the STATUS.RXNACK bit will be set. If the first byte is acknowledged by one or more slaves, then the master will proceed to transmit the second address byte and the master will first see the write interrupt flag after the second byte is transmitted. If the transaction direction is read-from-slave, the 10-bit address transmission must be followed by a repeated start and the first 7 bits of the address with the read/write bit equal to '1'.

Figure 34-8. 10-bit Address Transmission for a Read Transaction



This implies the following procedure for a 10-bit read operation:

1. Write the 10-bit address to ADDR.ADDR[10:1]. ADDR.TENBITEN must be '1', the direction bit (ADDR.ADDR[0]) must be '0' (can be written simultaneously with ADDR).

2. Once the Master on Bus interrupt is asserted, Write ADDR[7:0] register to '11110 `address[9:8]` 1'. ADDR.TENBITEN must be cleared (can be written simultaneously with ADDR).
3. Proceed to transmit data.

34.6.2.5. I²C Slave Operation

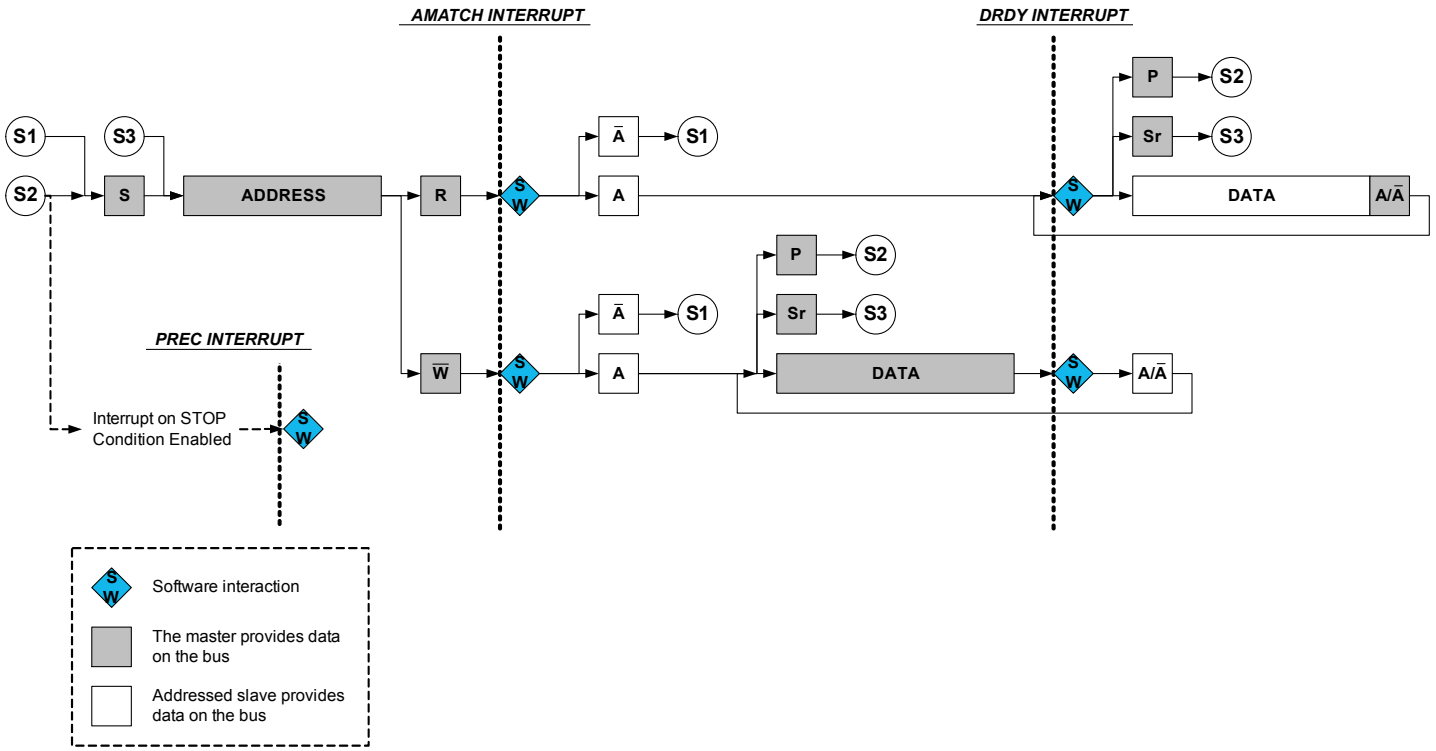
The I²C slave is byte-oriented and interrupt-based. The number of interrupts generated is kept at a minimum by automatic handling of most events. The software driver complexity and code size are reduced by auto-triggering of operations, and a special smart mode, which can be enabled by the Smart Mode Enable bit in the Control A register (CTRLA.SMEN).

The I²C slave has two interrupt strategies.

When SCL Stretch Mode bit (CTRLA.SCLSM) is '0', SCL is stretched before or after the acknowledge bit. In this mode, the I²C slave operates according to I²C Slave Behavioral Diagram (SCLSM=0). The circles labelled "S*n*" (S1, S2..) indicate the nodes the bus logic can jump to, based on software or hardware interaction.

This diagram is used as reference for the description of the I²C slave operation throughout the document.

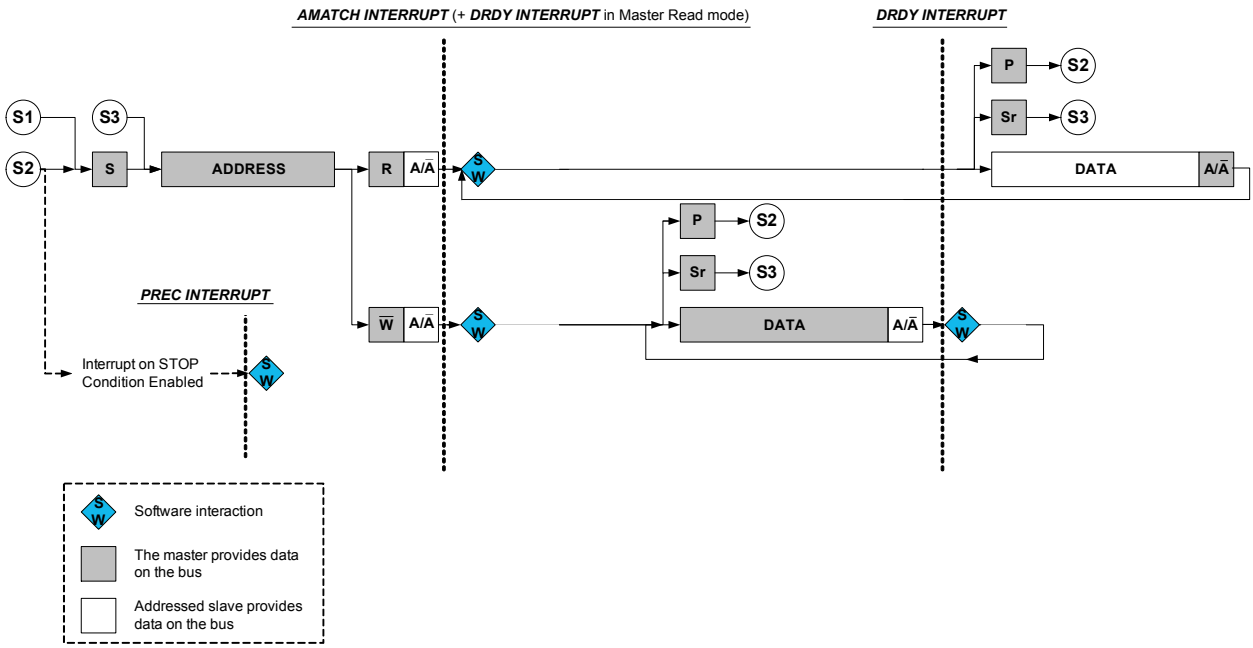
Figure 34-9. I²C Slave Behavioral Diagram (SCLSM=0)



In the second strategy (CTRLA.SCLSM=1), interrupts only occur after the ACK bit is sent as shown in Slave Behavioral Diagram (SCLSM=1). This strategy can be used when it is not necessary to check DATA before acknowledging. For master reads, an address and data interrupt will be issued simultaneously after the address acknowledge. However, for master writes, the first data interrupt will be seen after the first data byte has been received by the slave and the acknowledge bit has been sent to the master.

Note: For I²C High-speed mode (*Hs*), SCLSM=1 is required.

Figure 34-10. Slave Behavioral Diagram (SCLSM=1)



Receiving Address Packets (SCLSM=0)

When CTRLA.SCLSM=0, the I2C slave stretches the SCL line according to Figure 34-9 I2C Slave Behavioral Diagram (SCLSM=0) on page 652. When the I²C slave is properly configured, it will wait for a start condition.

When a start condition is detected, the successive address packet will be received and checked by the address match logic. If the received address is not a match, the packet will be rejected, and the I²C slave will wait for a new start condition. If the received address is a match, the Address Match bit in the Interrupt Flag register (INTFLAG.AMATCH) will be set.

SCL will be stretched until the I²C slave clears INTFLAG.AMATCH. As the I²C slave holds the clock by forcing SCL low, the software has unlimited time to respond.

The direction of a transaction is determined by reading the Read / Write Direction bit in the Status register (STATUS.DIR). This bit will be updated only when a valid address packet is received.

If the Transmit Collision bit in the Status register (STATUS.COLL) is set, this indicates that the last packet addressed to the I²C slave had a packet collision. A collision causes the SDA and SCL lines to be released without any notification to software. Therefore, the next AMATCH interrupt is the first indication of the previous packet's collision. Collisions are intended to follow the SMBus Address Resolution Protocol (ARP).

After the address packet has been received from the I²C master, one of two cases will arise based on transfer direction.

Case 1: Address packet accepted – Read flag set

The STATUS.DIR bit is '1', indicating an I²C master read operation. The SCL line is forced low, stretching the bus clock. If an ACK is sent, I²C slave hardware will set the Data Ready bit in the Interrupt Flag register (INTFLAG.DRDY), indicating data are needed for transmit. If a NACK is sent, the I²C slave will wait for a new start condition and address match.

Typically, software will immediately acknowledge the address packet by sending an ACK/NACK bit. The I²C slave Command bit field in the Control B register (CTRLB.CMD) can be written to '0x3' for both read

and write operations as the command execution is dependent on the STATUS.DIR bit. Writing '1' to INTFLAG.AMATCH will also cause an ACK/NACK to be sent corresponding to the CTRLB.ACKACT bit.

Case 2: Address packet accepted – Write flag set

The STATUS.DIR bit is cleared, indicating an I²C master write operation. The SCL line is forced low, stretching the bus clock. If an ACK is sent, the I²C slave will wait for data to be received. Data, repeated start or stop can be received.

If a NACK is sent, the I²C slave will wait for a new start condition and address match. Typically, software will immediately acknowledge the address packet by sending an ACK/NACK. The I²C slave command CTRLB.CMD = 3 can be used for both read and write operation as the command execution is dependent on STATUS.DIR.

Writing '1' to INTFLAG.AMATCH will also cause an ACK/NACK to be sent corresponding to the CTRLB.ACKACT bit.

Receiving Address Packets (SCLSM=1)

When SCLSM=1, the I²C slave will stretch the SCL line only after an ACK, see Slave Behavioral Diagram (SCLSM=1). When the I²C slave is properly configured, it will wait for a start condition to be detected.

When a start condition is detected, the successive address packet will be received and checked by the address match logic.

If the received address is not a match, the packet will be rejected and the I²C slave will wait for a new start condition.

If the address matches, the acknowledge action as configured by the Acknowledge Action bit Control B register (CTRLB.ACKACT) will be sent and the Address Match bit in the Interrupt Flag register (INTFLAG.AMATCH) is set. SCL will be stretched until the I²C slave clears INTFLAG.AMATCH. As the I²C slave holds the clock by forcing SCL low, the software is given unlimited time to respond to the address.

The direction of a transaction is determined by reading the Read/Write Direction bit in the Status register (STATUS.DIR). This bit will be updated only when a valid address packet is received.

If the Transmit Collision bit in the Status register (STATUS.COLL) is set, the last packet addressed to the I²C slave had a packet collision. A collision causes the SDA and SCL lines to be released without any notification to software. The next AMATCH interrupt is, therefore, the first indication of the previous packet's collision. Collisions are intended to follow the SMBus Address Resolution Protocol (*ARP*).

After the address packet has been received from the I²C master, INTFLAG.AMATCH be set to '1' to clear it.

Receiving and Transmitting Data Packets

After the I²C slave has received an address packet, it will respond according to the direction either by waiting for the data packet to be received or by starting to send a data packet by writing to DATA.DATA. When a data packet is received or sent, INTFLAG.DRDY will be set. After receiving data, the I²C slave will send an acknowledge according to CTRLB.ACKACT.

Case 1: Data received

INTFLAG.DRDY is set, and SCL is held low, pending for SW interaction.

Case 2: Data sent

When a byte transmission is successfully completed, the INTFLAG.DRDY interrupt flag is set. If NACK is received, indicated by STATUS.RXNACK=1, the I²C slave must expect a stop or a repeated start to be received. The I²C slave must release the data line to allow the I²C master to generate a stop or repeated

start. Upon detecting a stop condition, the Stop Received bit in the Interrupt Flag register (INTFLAG.PREC) will be set and the I²C slave will return to IDLE state.

High-Speed Mode

When the I²C slave is configured in High-speed mode (*Hs*, CTRLA.SPEED=0x2) and CTRLA.SCLSM=1, switching between Full-speed and High-speed modes is automatic. When the slave recognizes a START followed by a master code transmission and a NACK, it automatically switches to High-speed mode and sets the High-speed status bit (STATUS.HS). The slave will then remain in High-speed mode until a STOP is received.

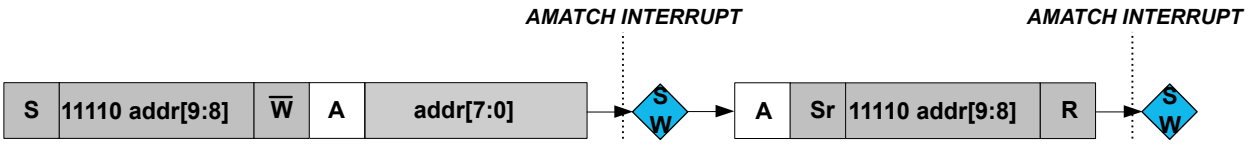
10-Bit Addressing

When 10-bit addressing is enabled (ADDR.TENBITEN=1), the two address bytes following a START will be checked against the 10-bit slave address recognition. The first byte of the address will always be acknowledged, and the second byte will raise the address interrupt flag, see 10-bit Addressing.

If the transaction is a write, then the 10-bit address will be followed by *N* data bytes.

If the operation is a read, the 10-bit address will be followed by a repeated START and reception of '11110 `ADDR[9:8]` 1', and the second address interrupt will be received with the DIR bit set. The slave matches on the second address as it it was addressed by the previous 10-bit address.

Figure 34-11. 10-bit Addressing



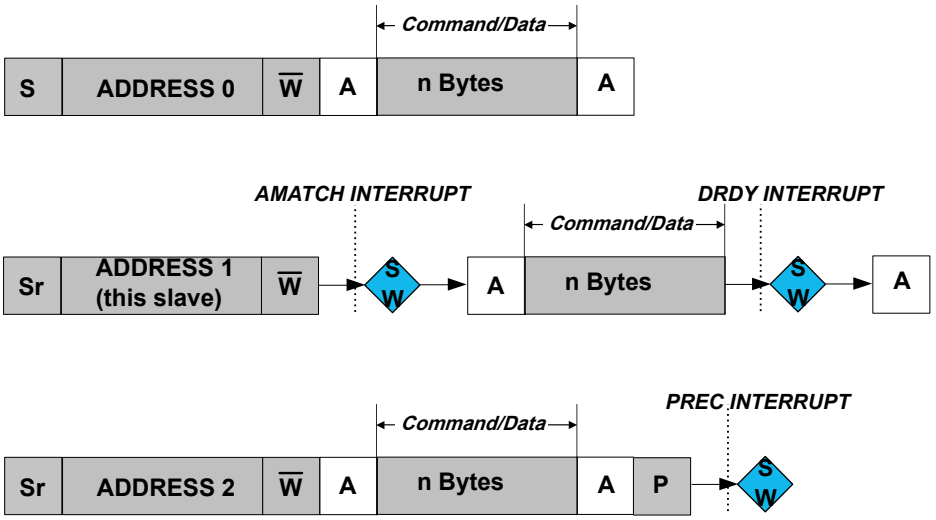
PMBus Group Command

When the PMBus Group Command bit in the CTRLB register is set (CTRLB.GCMD=1) and 7-bit addressing is used, INTFLAG.PREC will be set when a STOP condition is detected on the bus. When CTRLB.GCMD=0, a STOP condition without address match will not be set INTFLAG.PREC.

The group command protocol is used to send commands to more than one device. The commands are sent in one continuous transmission with a single STOP condition at the end. When the STOP condition is detected by the slaves addressed during the group command, they all begin executing the command they received.

PMBus Group Command Example shows an example where this slave, bearing ADDRESS 1, is addressed after a repeated START condition. There can be multiple slaves addressed before and after this slave. Eventually, at the end of the group command, a single STOP is generated by the master. At this point a STOP interrupt is asserted.

Figure 34-12. PMBus Group Command Example



34.6.3. Additional Features

34.6.3.1. SMBus

The I²C includes three hardware SCL low time-outs which allow a time-out to occur for SMBus SCL low time-out, master extend time-out, and slave extend time-out. This allows for SMBus functionality These time-outs are driven by the GCLK_SERCOM_SLOW clock. The GCLK_SERCOM_SLOW clock is used to accurately time the time-out and must be configured to use a 32KHz oscillator. The I²C interface also allows for a SMBus compatible SDA hold time.

- $T_{TIMEOUT}$: SCL low time of 25..35ms – Measured for a single SCL low period. It is enabled by CTRLA.LOWTOUTEN.
- $T_{LOW:SEXT}$: Cumulative clock low extend time of 25 ms – Measured as the cumulative SCL low extend time by a slave device in a single message from the initial START to the STOP. It is enabled by CTRLA.SEXTTOEN.
- $T_{LOW:MEXT}$: Cumulative clock low extend time of 10 ms – Measured as the cumulative SCL low extend time by the master device within a single byte from START-to-ACK, ACK-to-ACK, or ACK-to-STOP. It is enabled by CTRLA.MEXTTOEN.

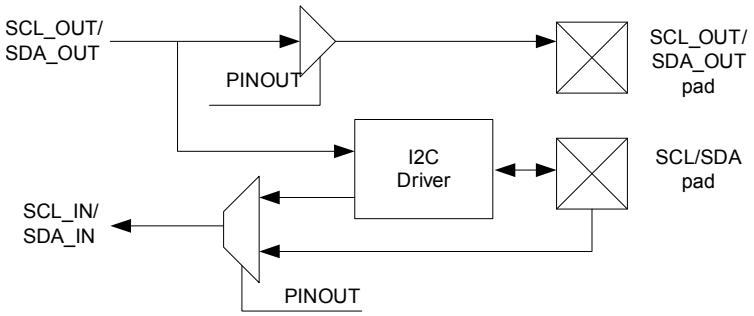
34.6.3.2. Smart Mode

The I²C interface has a smart mode that simplifies application code and minimizes the user interaction needed to adhere to the I²C protocol. The smart mode accomplishes this by automatically issuing an ACK or NACK (based on the content of CTRLB.ACKACT) as soon as DATA.DATA is read.

34.6.3.3. 4-Wire Mode

Selecting the Pin Usage bit in the Control A register (CTRLA.PINOUT) will enable 4-wire mode operation. In this mode, the internal I²C tri-state drivers are bypassed, and an external I²C compliant tri-state driver is needed when connecting to an I²C bus.

Figure 34-13. I²C Pad Interface



34.6.3.4. Quick Command

Setting the Quick Command Enable bit in the Control B register (CTRLB.QCEN) enables quick command. When quick command is enabled, the corresponding interrupt flag (INTFLAG.SB or INTFLAG.MB) is set immediately after the slave acknowledges the address. At this point, the software can either issue a stop command or a repeated start by writing CTRLB.CMD or ADDR.ADDR.

34.6.4. DMA, Interrupts and Events

Table 34-1. Module Request for SERCOM I²C Slave

Condition	Request		
	DMA	Interrupt	Event
Data needed for transmit (TX) (Slave transmit mode)	Yes (request cleared when data is written)		NA
Data received (RX) (Slave receive mode)	Yes (request cleared when data is read)		
Data Ready (DRDY)		Yes	
Address Match (AMATCH)		Yes	
Stop received (PREC)		Yes	
Error (ERROR)		Yes	

Table 34-2. Module Request for SERCOM I²C Master

Condition	Request		
	DMA	Interrupt	Event
Data needed for transmit (TX) (Master transmit mode)	Yes (request cleared when data is written)		NA
Data needed for transmit (RX) (Master transmit mode)	Yes (request cleared when data is read)		
Master on Bus (MB)		Yes	
Stop received (SB)		Yes	
Error (ERROR)		Yes	

34.6.4.1. DMA Operation

Smart mode must be enabled for DMA operation in the Control B register by writing CTRLB.SMEN=1.

Slave DMA

When using the I²C slave with DMA, an address match will cause the address interrupt flag (INTFLAG.ADDRMATCH) to be raised. After the interrupt has been serviced, data transfer will be performed through DMA.

The I²C slave generates the following requests:

- Write data received (RX): The request is set when master write data is received. The request is cleared when DATA is read.
- Read data needed for transmit (TX): The request is set when data is needed for a master read operation. The request is cleared when DATA is written.

Master DMA

When using the I²C master with DMA, the ADDR register must be written with the desired address (ADDR.ADDR), transaction length (ADDR.LEN), and transaction length enable (ADDR.LENEN). When ADDR.LENEN is written to 1 along with ADDR.ADDR, ADDR.LEN determines the number of data bytes in the transaction from 0 to 255. DMA is then used to transfer ADDR.LEN bytes followed by an automatically generated NACK (for master reads) and a STOP.

If a NACK is received by the slave for a master write transaction before ADDR.LEN bytes, a STOP will be automatically generated and the length error (STATUS.LENERR) will be raised along with the INTFLAG.ERROR interrupt.

The I²C master generates the following requests:

- Read data received (RX): The request is set when master read data is received. The request is cleared when DATA is read.
- Write data needed for transmit (TX): The request is set when data is needed for a master write operation. The request is cleared when DATA is written.

34.6.4.2. Interrupts

The I²C slave has the following interrupt sources. These are asynchronous interrupts. They can wake-up the device from any sleep mode:

- Error (ERROR)

- Data Ready (DRDY)
- Address Match (AMATCH)
- Stop Received (PREC)

The I²C master has the following interrupt sources. These are asynchronous interrupts. They can wake-up the device from any sleep mode:

- Error (ERROR)
- Slave on Bus (SB)
- Master on Bus (MB)

Each interrupt source has its own interrupt flag. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) will be set when the interrupt condition is meet. Each interrupt can be individually enabled by writing '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR). An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request active until the interrupt flag is cleared, the interrupt is disabled or the I²C is reset. See INTFLAG on page 693 register for details on how to clear interrupt flags.

The I²C has one common interrupt request line for all the interrupt sources. The value of INTFLAG indicates which interrupt is executed. Note that interrupts must be globally enabled for interrupt requests. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

Nested Vector Interrupt Controller on page 51

34.6.4.3. Events

Not applicable.

34.6.5. Sleep Mode Operation

I²C Master Operation

The generic clock (GCLK_SERCOMx_CORE) will continue to run in idle sleep mode. If the Run In Standby bit in the Control A register (CTRLA.RUNSTDBY) is '1', the GLK_SERCOMx_CORE will also run in standby sleep mode. Any interrupt can wake up the device.

If CTRLA.RUNSTDBY=0, the GLK_SERCOMx_CORE will be disabled after any ongoing transaction is finished. Any interrupt can wake up the device.

I²C Slave Operation

Writing CTRLA.RUNSTDBY=1 will allow the Address Match interrupt to wake up the device.

When CTRLA.RUNSTDBY=0, all receptions will be dropped.

34.6.6. Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read. An exception is the Channel Enable bit in the Peripheral Channel Control registers (PCHCTRLm.CHEN). When changing this bit, the bit value must be read-back to ensure the synchronization is complete and to assert glitch free internal operation. Note that changing the bit value under ongoing synchronization will *not* generate an error.

The following bits are synchronized when written:

- Software Reset bit in the CTRLA register (CTRLA.SWRST)
- Enable bit in the CTRLA register (CTRLA.ENABLE)

- Write to Bus State bits in the Status register (STATUS.BUSSTATE)
- Address bits in the Address register (ADDR.ADDR) when in master operation.

The following registers are synchronized when written:

- Data (DATA) when in master operation

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

34.7. Register Summary - I2C Slave

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
0x01		15:8								
0x02		23:16	SEXTTOEN		SDAHOLD[1:0]					PINOUT
0x03		31:24		LOWTOUT			SCLSM		SPEED[1:0]	
0x04	CTRLB	7:0								
0x05		15:8	AMODE[1:0]					AACKEN	GCMD	SMEN
0x06		23:16						ACKACT	CMD[1:0]	
0x07		31:24								
0x08 ... 0x13	Reserved									
0x14	INTENCLR	7:0	ERROR					DRDY	AMATCH	PREC
0x15	Reserved									
0x16	INTENSET	7:0	ERROR					DRDY	AMATCH	PREC
0x17	Reserved									
0x18	INTFLAG	7:0	ERROR					DRDY	AMATCH	PREC
0x19	Reserved									
0x1A	STATUS	7:0	CLKHOLD	LOWTOUT		SR	DIR	RXNACK	COLL	BUSERR
0x1B		15:8						LENERR	SEXTTOUT	
0x1C	SYNCBUSY	7:0							ENABLE	SWRST
0x1D		15:8								
0x1E		23:16								
0x1F		31:24								
0x20 ... 0x23	Reserved									
0x24	ADDR	7:0	ADDR[6:0]							GENCEN
0x25		15:8	TENBITEN					ADDR[9:7]		
0x26		23:16	ADDRMASK[6:0]							
0x27		31:24						ADDRMASK[9:7]		
0x28	DATA	7:0	DATA[7:0]							
0x29		15:8								

34.8. Register Description - I2C Slave

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection on page 642.

Some registers are synchronized when read and/or written. Synchronization is denoted by the "Write-Synchronized" or the "Read-Synchronized" property in each individual register description. For details, refer to Synchronization on page 659.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

34.8.1. Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
		LOWTOUT			SCLSM		SPEED[1:0]	
Access		R/W			R/W		R/W	R/W
Reset		0			0		0	0

Bit	23	22	21	20	19	18	17	16
	SEXTTOEN		SDAHOLD[1:0]					PINOUT
Access	R/W		R/W	R/W				R/W
Reset	0		0	0				0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			0	0	0	0	0

Bit 30 – LOWTOUT: SCL Low Time-Out
This bit enables the SCL low time-out. If SCL is held low for 25ms-35ms, the slave will release its clock hold, if enabled, and reset the internal state machine. Any interrupt flags set at the time of time-out will remain set.

Value	Description
0	Time-out disabled.
1	Time-out enabled.

Bit 27 – SCLSM: SCL Clock Stretch Mode
This bit controls when SCL will be stretched for software interaction.

This bit is not synchronized.

Value	Description
0	SCL stretch according to Figure 34-5 I2C Master Behavioral Diagram (SCLSM=1) on page 647
1	SCL stretch only after ACK bit according to Figure 34-9 I2C Slave Behavioral Diagram (SCLSM=0) on page 652

Bits 25:24 – SPEED[1:0]: Transfer Speed
These bits define bus speed.

These bits are not synchronized.

Value	Description
0x0	Standard-mode (Sm) up to 100 kHz and Fast-mode (Fm) up to 400 kHz
0x1	Fast-mode Plus (Fm+) up to 1 MHz
0x2	High-speed mode (Hs-mode) up to 3.4 MHz
0x3	Reserved

Bit 23 – SEXTTOEN: Slave SCL Low Extend Time-Out
This bit enables the slave SCL low extend time-out. If SCL is cumulatively held low for greater than 25ms from the initial START to a STOP, the slave will release its clock hold if enabled and reset the internal state machine. Any interrupt flags set at the time of time-out will remain set. If the address was recognized, PREC will be set when a STOP is received.

This bit is not synchronized.

Value	Description
0	Time-out disabled
1	Time-out enabled

Bits 21:20 – SDAHOLD[1:0]: SDA Hold Time
These bits define the SDA hold time with respect to the negative edge of SCL.

These bits are not synchronized.

Value	Name	Description
0x0	DIS	Disabled
0x1	75	50-100ns hold time
0x2	450	300-600ns hold time
0x3	600	400-800ns hold time

Bit 16 – PINOUT: Pin Usage
This bit sets the pin usage to either two- or four-wire operation:

This bit is not synchronized.

Value	Description
0	4-wire operation disabled
1	4-wire operation enabled

Bit 7 – RUNSTDBY: Run in Standby
This bit defines the functionality in standby sleep mode.

This bit is not synchronized.

Value	Description
0	Disabled – All reception is dropped.
1	Wake on address match, if enabled.

Bits 4:2 – MODE[2:0]: Operating Mode
These bits must be written to 0x04 to select the I^2C slave serial communication interface of the SERCOM.

These bits are not synchronized.

Bit 1 – ENABLE: Enable
Due to synchronization, there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRL.ENABLE will read back immediately and the Enable Synchronization Busy bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

This bit is not enable-protected.

Value	Description
0	The peripheral is disabled or being disabled.
1	The peripheral is enabled.

Bit 0 – SWRST: Software Reset
Writing '0' to this bit has no effect.

Writing '1' to this bit resets all registers in the SERCOM, except DBGCTRL, to their initial state, and the SERCOM will be disabled.

Writing '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded. Any register write access during the ongoing reset will result in an APB error. Reading any register will return the reset value of the register.

Due to synchronization, there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

This bit is not enable-protected.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

34.8.2. Control B

Name: CTRLB
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						ACKACT	CMD[1:0]	
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
	AMODE[1:0]					AACKEN	GCMD	SMEN
Access	R/W	R/W				R/W	R/W	R/W
Reset	0	0				0	0	0

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 18 – ACKACT: Acknowledge Action
This bit defines the slave's acknowledge behavior after an address or data byte is received from the master. The acknowledge action is executed when a command is written to the CMD bits. If smart mode is enabled (CTRLB.SMEN=1), the acknowledge action is performed when the DATA register is read.

This bit is not enable-protected.

Value	Description
0	Send ACK
1	Send NACK

Bits 17:16 – CMD[1:0]: Command
This bit field triggers the slave operation as the below. The CMD bits are strobe bits, and always read as zero. The operation is dependent on the slave interrupt flags, INTFLAG.DRDY and INTFLAG.AMATCH, in addition to STATUS.DIR.

All interrupt flags (INTFLAG.DRDY, INTFLAG.AMATCH and INTFLAG.PREC) are automatically cleared when a command is given.

This bit is not enable-protected.

Table 34-3. Command Description

CMD[1:0]	DIR	Action
0x0	X	(No action)
0x1	X	(Reserved)
0x2	Used to complete a transaction in response to a data interrupt (DRDY)	
	0 (Master write)	Execute acknowledge action succeeded by waiting for any start (S/Sr) condition
	1 (Master read)	Wait for any start (S/Sr) condition
0x3	Used in response to an address interrupt (AMATCH)	
	0 (Master write)	Execute acknowledge action succeeded by reception of next byte
	1 (Master read)	Execute acknowledge action succeeded by slave data interrupt
	Used in response to a data interrupt (DRDY)	
	0 (Master write)	Execute acknowledge action succeeded by reception of next byte
	1 (Master read)	Execute a byte read operation followed by ACK/NACK reception

Bits 15:14 – AMODE[1:0]: Address Mode
These bits set the addressing mode.

These bits are not write-synchronized.

Value	Name	Description
0x0	MASK	The slave responds to the address written in ADDR.ADDR masked by the value in ADDR.ADDRMASK. See *SERCOM – Serial Communication Interface* for additional information.
0x1	2_ADDRS	The slave responds to the two unique addresses in ADDR.ADDR and ADDR.ADDRMASK.
0x2	RANGE	The slave responds to the range of addresses between and including ADDR.ADDR and ADDR.ADDRMASK. ADDR.ADDR is the upper limit.
0x3	-	Reserved.

Bit 10 – AACKEN: Automatic Acknowledge Enable
This bit enables the address to be automatically acknowledged if there is an address match.

This bit is not write-synchronized.

Value	Description
0	Automatic acknowledge is disabled.
1	Automatic acknowledge is enabled.

Bit 9 – GCMD: PMBus Group Command
This bit enables PMBus group command support. When enabled, the Stop Recived interrupt flag (INTFLAG.PREC) will be set when a STOP condition is detected if the slave has been addressed since the last STOP condition on the bus.

This bit is not write-synchronized.

Value	Description
0	Group command is disabled.
1	Group command is enabled.

Bit 8 – SMEN: Smart Mode Enable
When smart mode is enabled, data is acknowledged automatically when DATA.DATA is read.

This bit is not write-synchronized.

Value	Description
0	Smart mode is disabled.
1	Smart mode is enabled.

34.8.3. Interrupt Enable Clear

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Name: INTENCLR
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	ERROR					DRDY	AMATCH	PREC
Access	R/W					R/W	R/W	R/W
Reset	0					0	0	0

Bit 7 – ERROR: Error Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Error Interrupt Enable bit, which disables the Error interrupt.

Value	Description
0	Error interrupt is disabled.
1	Error interrupt is enabled.

Bit 2 – DRDY: Data Ready Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Data Ready bit, which disables the Data Ready interrupt.

Value	Description
0	The Data Ready interrupt is disabled.
1	The Data Ready interrupt is enabled.

Bit 1 – AMATCH: Address Match Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Address Match Interrupt Enable bit, which disables the Address Match interrupt.

Value	Description
0	The Address Match interrupt is disabled.
1	The Address Match interrupt is enabled.

Bit 0 – PREC: Stop Received Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Stop Received Interrupt Enable bit, which disables the Stop Received interrupt.

Value	Description
0	The Stop Received interrupt is disabled.
1	The Stop Received interrupt is enabled.

34.8.4. Interrupt Enable Set

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Name: INTENSET
Offset: 0x16
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	ERROR					DRDY	AMATCH	PREC
Access	R/W					R/W	R/W	R/W
Reset	0					0	0	0

Bit 7 – ERROR: Error Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Error Interrupt Enable bit, which enables the Error interrupt.

Value	Description
0	Error interrupt is disabled.
1	Error interrupt is enabled.

Bit 2 – DRDY: Data Ready Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Data Ready bit, which enables the Data Ready interrupt.

Value	Description
0	The Data Ready interrupt is disabled.
1	The Data Ready interrupt is enabled.

Bit 1 – AMATCH: Address Match Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Address Match Interrupt Enable bit, which enables the Address Match interrupt.

Value	Description
0	The Address Match interrupt is disabled.
1	The Address Match interrupt is enabled.

Bit 0 – PREC: Stop Received Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Stop Received Interrupt Enable bit, which enables the Stop Received interrupt.

Value	Description
0	The Stop Received interrupt is disabled.
1	The Stop Received interrupt is enabled.

34.8.5. Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x18
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	ERROR					DRDY	AMATCH	PREC
Access	R/W					R/W	R/W	R/W
Reset	0					0	0	0

Bit 7 – ERROR: Error
This bit is set when any error is detected. Errors that will set this flag have corresponding status flags in the STATUS register. The corresponding bits in STATUS are SEXTTOUT, LOWTOUT, COLL, and BUSERR.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the flag.

Bit 2 – DRDY: Data Ready
This flag is set when a I^2C slave byte transmission is successfully completed.

The flag is cleared by hardware when either:

- Writing to the DATA register.
- Reading the DATA register with smart mode enabled.
- Writing a valid command to the CMD register.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Data Ready interrupt flag.

Bit 1 – AMATCH: Address Match
This flag is set when the I^2C slave address match logic detects that a valid address has been received.

The flag is cleared by hardware when CTRL.CMD is written.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Address Match interrupt flag. When cleared, an ACK/NACK will be sent according to CTRLB.ACKACT.

Bit 0 – PREC: Stop Received
This flag is set when a stop condition is detected for a transaction being processed. A stop condition detected between a bus master and another slave will not set this flag, unless the PMBus Group Command is enabled in the Control B register (CTRLB.GCMD=1).

This flag is cleared by hardware after a command is issued on the next address match.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Stop Received interrupt flag.

34.8.6. Status

Name: STATUS
Offset: 0x1A
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
						LENERR	SEXTTOUT	
Access						R/W	R/W	
Reset						0	0	

Bit	7	6	5	4	3	2	1	0
	CLKHOLD	LOWTOUT		SR	DIR	RXNACK	COLL	BUSERR
Access	R	R/W		R	R	R	R/W	R/W
Reset	0	0		0	0	0	0	0

Bit 10 – LENERR: Transaction Length Error
This bit is set when the length counter is enabled (LENGTH.LENEN) and a STOP or repeated START is received before or after the length in LENGTH.LEN is reached.
This bit is cleared automatically when responding to a new start condition with ACK or NACK (CTRLB.CMD=0x3) or when INTFLAG.AMATCH is cleared.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the status.

Bit 10 – HS: High-speed
This bit is set if the slave detects a START followed by a Master Code transmission.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the status. However, this flag is automatically cleared when a STOP is received.

Bit 9 – SEXTTOUT: Slave SCL Low Extend Time-Out
This bit is set if a slave SCL low extend time-out occurs.

This bit is cleared automatically if responding to a new start condition with ACK or NACK (write 3 to CTRLB.CMD) or when INTFLAG.AMATCH is cleared.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the status.

Value	Description
0	No SCL low extend time-out has occurred.
1	SCL low extend time-out has occurred.

Bit 7 – CLKHOLD: Clock Hold
The slave Clock Hold bit (STATUS.CLKHOLD) is set when the slave is holding the SCL line low, stretching the I2C clock. Software should consider this bit a read-only status flag that is set when INTFLAG.DRDY or INTFLAG.AMATCH is set.

This bit is automatically cleared when the corresponding interrupt is also cleared.

Bit 6 – LOWTOUT: SCL Low Time-out
This bit is set if an SCL low time-out occurs.

This bit is cleared automatically if responding to a new start condition with ACK or NACK (write 3 to CTRLB.CMD) or when INTFLAG.AMATCH is cleared.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the status.

Value	Description
0	No SCL low time-out has occurred.
1	SCL low time-out has occurred.

Bit 4 – SR: Repeated Start
When INTFLAG.AMATCH is raised due to an address match, SR indicates a repeated start or start condition.

This flag is only valid while the INTFLAG.AMATCH flag is one.

Value	Description
0	Start condition on last address match
1	Repeated start condition on last address match

Bit 3 – DIR: Read / Write Direction
The Read/Write Direction (STATUS.DIR) bit stores the direction of the last address packet received from a master.

Value	Description
0	Master write operation is in progress.
1	Master read operation is in progress.

Bit 2 – RXNACK: Received Not Acknowledge
This bit indicates whether the last data packet sent was acknowledged or not.

Value	Description
0	Master responded with ACK.
1	Master responded with NACK.

Bit 1 – COLL: Transmit Collision
If set, the I2C slave was not able to transmit a high data or NACK bit, the I2C slave will immediately release the SDA and SCL lines and wait for the next packet addressed to it.

This flag is intended for the SMBus address resolution protocol (ARP). A detected collision in non-ARP situations indicates that there has been a protocol violation, and should be treated as a bus error.

Note that this status will not trigger any interrupt, and should be checked by software to verify that the data were sent correctly. This bit is cleared automatically if responding to an address match with an ACK or a NACK (writing 0x3 to CTRLB.CMD), or INTFLAG.AMATCH is cleared.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the status.

Value	Description
0	No collision detected on last data byte sent.
1	Collision detected on last data byte sent.

Bit 0 – BUSERR: Bus Error
The Bus Error bit (STATUS.BUSERR) indicates that an illegal bus condition has occurred on the bus, regardless of bus ownership. An illegal bus condition is detected if a protocol violating start, repeated start or stop is detected on the I2C bus lines. A start condition directly followed by a stop condition is one example of a protocol violation. If a time-out occurs during a frame, this is also considered a protocol violation, and will set STATUS.BUSERR.

This bit is cleared automatically if responding to an address match with an ACK or a NACK (writing 0x3 to CTRLB.CMD) or INTFLAG.AMATCH is cleared.

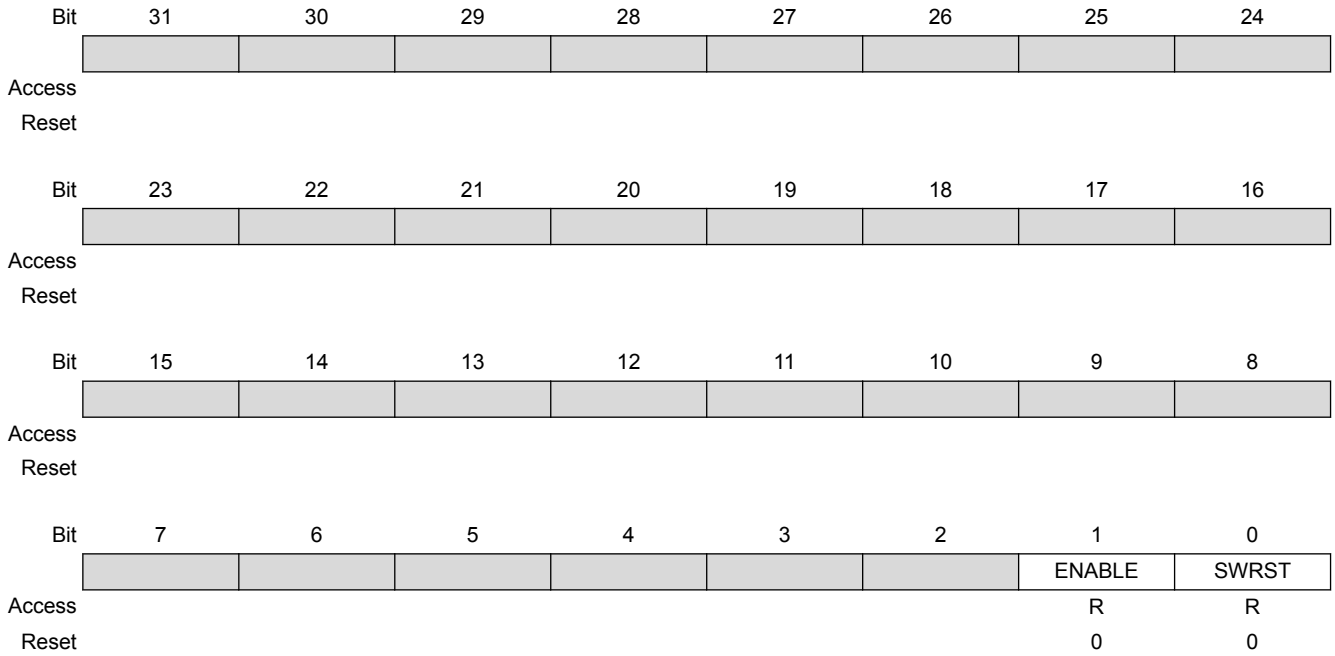
Writing a '1' to this bit will clear the status.

Writing a '0' to this bit has no effect.

Value	Description
0	No bus error detected.
1	Bus error detected.

34.8.7. Synchronization Busy

Name: SYNCBUSY
Offset: 0x1C
Reset: 0x00000000
Property:



Bit 1 – ENABLE: SERCOM Enable Synchronization Busy
Enabling and disabling the SERCOM (CTRLA.ENABLE) requires synchronization. When written, the SYNCBUSY.ENABLE bit will be set until synchronization is complete.

Writes to any register (except for CTRLA.SWRST) while enable synchronization is on-going will be discarded and an APB error will be generated.

Value	Description
0	Enable synchronization is not busy.
1	Enable synchronization is busy.

Bit 0 – SWRST: Software Reset Synchronization Busy
Resetting the SERCOM (CTRLA.SWRST) requires synchronization. When written, the SYNCBUSY.SWRST bit will be set until synchronization is complete.

Writes to any register while synchronization is on-going will be discarded and an APB error will be generated.

Value	Description
0	SWRST synchronization is not busy.
1	SWRST synchronization is busy.

34.8.8. Address

Name: ADDR
Offset: 0x24
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
						ADDRMASK[9:7]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
	ADDRMASK[6:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	

Bit	15	14	13	12	11	10	9	8
	TENBITEN					ADDR[9:7]		
Access	R/W					R/W	R/W	R/W
Reset	0					0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR[6:0]							GENCEN
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 26:17 – ADDRMASK[9:0]: Address Mask
These bits act as a second address match register, an address mask register or the lower limit of an address range, depending on the CTRLB.AMODE setting.

Bit 15 – TENBITEN: Ten Bit Addressing Enable

Value	Description
0	10-bit address recognition disabled.
1	10-bit address recognition enabled.

Bits 10:1 – ADDR[9:0]: Address
These bits contain the I^2C slave address used by the slave address match logic to determine if a master has addressed the slave.

When using 7-bit addressing, the slave address is represented by ADDR[6:0].

When using 10-bit addressing (ADDR.TENBITEN=1), the slave address is represented by ADDR[9:0]

When the address match logic detects a match, INTFLAG.AMATCH is set and STATUS.DIR is updated to indicate whether it is a read or a write transaction.

Bit 0 – GENCEN: General Call Address Enable
A general call address is an address consisting of all-zeroes, including the direction bit (master write).

Value	Description
0	General call address recognition disabled.
1	General call address recognition enabled.

34.8.9. Data

Name: DATA
Offset: 0x28
Reset: 0x0000
Property: Write-Synchronized, Read-Synchronized

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – DATA[7:0]: Data
The slave data register I/O location (DATA.DATA) provides access to the master transmit and receive data buffers. Reading valid data or writing data to be transmitted can be successfully done only when SCL is held low by the slave (STATUS.CLKHOLD is set). An exception occurs when reading the last data byte after the stop condition has been received.

Accessing DATA.DATA auto-triggers I^2C bus operations. The operation performed depends on the state of CTRLB.ACKACT, CTRLB.SMEN and the type of access (read/write).

Writing or reading DATA.DATA when not in smart mode does not require synchronization.

34.9. Register Summary - I2C Master

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
0x01		15:8								
0x02		23:16	SEXTTOEN	MEXTTOEN	SDAHOLD[1:0]					PINOUT
0x03		31:24		LOWTOUT	INACTOUT[1:0]		SCLSM		SPEED[1:0]	
0x04	CTRLB	7:0								
0x05		15:8							QCEN	SMEN
0x06		23:16						ACKACT	CMD[1:0]	
0x07		31:24								
0x08 ... 0x0B	Reserved									
0x0C	BAUD	7:0	BAUD[7:0]							
0x0D		15:8	BAUDLOW[7:0]							
0x0E		23:16	HSBAUD[7:0]							
0x0F		31:24	HSBAUDLOW[7:0]							
0x10 ... 0x13	Reserved									
0x14	INTENCLR	7:0	ERROR						SB	MB
0x15	Reserved									
0x16	INTENSET	7:0	ERROR						SB	MB
0x17	Reserved									
0x18	INTFLAG	7:0	ERROR						SB	MB
0x18	DATA	7:0	DATA[7:0]							
0x19		15:8								
0x1A	STATUS	7:0	CLKHOLD	LOWTOUT	BUSSTATE[1:0]			RXNACK	ARBLOST	BUSERR
0x1B		15:8						LENERR	SEXTTOUT	MEXTTOUT
0x1C	SYNCBUSY	7:0						SYSOP	ENABLE	SWRST
0x1D		15:8								
0x1E		23:16								
0x1F		31:24								
0x21 ... 0x23	Reserved									
0x24	ADDR	7:0								
0x25		15:8	TENBITEN	HS	LENEN			ADDR[2:0]		
0x26		23:16	LEN[7:0]							
0x27		31:24								
0x28 ... 0x2F	Reserved									
0x30	DBGCTRL	7:0								DBGSTOP

34.10. Register Description - I²C Master

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection on page 642.

Some registers are synchronized when read and/or written. Synchronization is denoted by the "Write-Synchronized" or the "Read-Synchronized" property in each individual register description. For details, refer to Synchronization on page 659.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

34.10.1. Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
		LOWTOUT	INACTOUT[1:0]		SCLSM		SPEED[1:0]	
Access		R/W	R/W	R/W	R/W		R/W	R/W
Reset		0	0	0	0		0	0

Bit	23	22	21	20	19	18	17	16
	SEXTTOEN	MEXTTOEN	SDAHOLD[1:0]					PINOUT
Access	R/W	R/W	R/W	R/W				R/W
Reset	0	0	0	0				0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			0	0	0	0	0

Bit 30 – LOWTOUT: SCL Low Time-Out
This bit enables the SCL low time-out. If SCL is held low for 25ms-35ms, the master will release its clock hold, if enabled, and complete the current transaction. A stop condition will automatically be transmitted.

INTFLAG.SB or INTFLAG.MB will be set as normal, but the clock hold will be released. The STATUS.LOWTOUT and STATUS.BUSERR status bits will be set.

This bit is not synchronized.

Value	Description
0	Time-out disabled.
1	Time-out enabled.

Bits 29:28 – INACTOUT[1:0]: Inactive Time-Out
If the inactive bus time-out is enabled and the bus is inactive for longer than the time-out setting, the bus state logic will be set to idle. An inactive bus arise when either an I^2C master or slave is holding the SCL low.

Enabling this option is necessary for SMBus compatibility, but can also be used in a non-SMBus set-up.

Calculated time-out periods are based on a 100kHz baud rate.

These bits are not synchronized.

Value	Name	Description
0x0	DIS	Disabled
0x1	55US	5-6 SCL cycle time-out (50-60µs)
0x2	105US	10-11 SCL cycle time-out (100-110µs)
0x3	205US	20-21 SCL cycle time-out (200-210µs)

Bit 27 – SCLSM: SCL Clock Stretch Mode
This bit controls when SCL will be stretched for software interaction.

This bit is not synchronized.

Value	Description
0	SCL stretch according to Figure 34-5 I2C Master Behavioral Diagram (SCLSM=1) on page 647.
1	SCL stretch only after ACK bit.

Bits 25:24 – SPEED[1:0]: Transfer Speed
These bits define bus speed.

These bits are not synchronized.

Value	Description
0x0	Standard-mode (Sm) up to 100 kHz and Fast-mode (Fm) up to 400 kHz
0x1	Fast-mode Plus (Fm+) up to 1 MHz
0x2	High-speed mode (Hs-mode) up to 3.4 MHz
0x3	Reserved

Bit 23 – SEXTTOEN: Slave SCL Low Extend Time-Out
This bit enables the slave SCL low extend time-out. If SCL is cumulatively held low for greater than 25ms from the initial START to a STOP, the master will release its clock hold if enabled, and complete the current transaction. A STOP will automatically be transmitted.

SB or MB will be set as normal, but CLKHOLD will be release. The MEXTTOUT and BUSERR status bits will be set.

This bit is not synchronized.

Value	Description
0	Time-out disabled
1	Time-out enabled

Bit 22 – MEXTTOEN: Master SCL Low Extend Time-Out
This bit enables the master SCL low extend time-out. If SCL is cumulatively held low for greater than 10ms from START-to-ACK, ACK-to-ACK, or ACK-to-STOP the master will release its clock hold if enabled, and complete the current transaction. A STOP will automatically be transmitted.

SB or MB will be set as normal, but CLKHOLD will be released. The MEXTTOUT and BUSERR status bits will be set.

This bit is not synchronized.

Value	Description
0	Time-out disabled
1	Time-out enabled

Bits 21:20 – SDAHOLD[1:0]: SDA Hold Time
These bits define the SDA hold time with respect to the negative edge of SCL.

These bits are not synchronized.

Value	Name	Description
0x0	DIS	Disabled
0x1	75NS	50-100ns hold time
0x2	450NS	300-600ns hold time
0x3	600NS	400-800ns hold time

Bit 16 – PINOUT: Pin Usage
This bit set the pin usage to either two- or four-wire operation:

This bit is not synchronized.

Value	Description
0	4-wire operation disabled.
1	4-wire operation enabled.

Bit 7 – RUNSTDBY: Run in Standby
This bit defines the functionality in standby sleep mode.

This bit is not synchronized.

Value	Description
0	GCLK_SERCOMx_CORE is disabled and the I^2C master will not operate in standby sleep mode.
1	GCLK_SERCOMx_CORE is enabled in all sleep modes.

Bits 4:2 – MODE[2:0]: Operating Mode
These bits must be written to 0x5 to select the I^2C master serial communication interface of the SERCOM.

These bits are not synchronized.

Bit 1 – ENABLE: Enable
Due to synchronization, there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRL.ENABLE will read back immediately and the Synchronization Enable Busy bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

This bit is not enable-protected.

Value	Description
0	The peripheral is disabled or being disabled.
1	The peripheral is enabled.

Bit 0 – SWRST: Software Reset
Writing '0' to this bit has no effect.

Writing '1' to this bit resets all registers in the SERCOM, except DBGCTRL, to their initial state, and the SERCOM will be disabled.

Writing '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded. Any register write access during the ongoing reset will result in an APB error. Reading any register will return the reset value of the register.

Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

This bit is not enable-protected.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

34.10.2. Control B

Name: CTRLB
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						ACKACT	CMD[1:0]	
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
							QCEN	SMEN
Access							R	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 18 – ACKACT: Acknowledge Action
This bit defines the I^2C master's acknowledge behavior after a data byte is received from the I^2C slave. The acknowledge action is executed when a command is written to CTRLB.CMD, or if smart mode is enabled (CTRLB.SMEN is written to one), when DATA.DATA is read.

This bit is not enable-protected.

This bit is not write-synchronized.

Value	Description
0	Send ACK.
1	Send NACK.

Bits 17:16 – CMD[1:0]: Command
Writing these bits triggers a master operation as described below. The CMD bits are strobe bits, and always read as zero. The acknowledge action is only valid in master read mode. In master write mode, a command will only result in a repeated start or stop condition. The CTRLB.ACKACT bit and the CMD bits can be written at the same time, and then the acknowledge action will be updated before the command is triggered.

Commands can only be issued when either the Slave on Bus interrupt flag (INTFLAG.SB) or Master on Bus interrupt flag (INTFLAG.MB) is '1'.

If CMD 0x1 is issued, a repeated start will be issued followed by the transmission of the current address in ADDR.ADDR. If another address is desired, ADDR.ADDR must be written instead of the CMD bits. This will trigger a repeated start followed by transmission of the new address.

Issuing a command will set the System Operation bit in the Synchronization Busy register (SYNCBUSY.SYSOP).

Table 34-4. Command Description

CMD[1:0]	Direction	Action
0x0	X	(No action)
0x1	X	Execute acknowledge action succeeded by repeated Start
0x2	0 (Write)	No operation
	1 (Read)	Execute acknowledge action succeeded by a byte read operation
0x3	X	Execute acknowledge action succeeded by issuing a stop condition

These bits are not enable-protected.

Bit 9 – QCEN: Quick Command Enable
This bit is not write-synchronized.

Value	Description
0	Quick Command is disabled.
1	Quick Command is enabled.

Bit 8 – SMEN: Smart Mode Enable
When smart mode is enabled, acknowledge action is sent when DATA.DATA is read.

This bit is not write-synchronized.

Value	Description
0	Smart mode is disabled.
1	Smart mode is enabled.

34.10.3. Baud Rate

Name: BAUD
Offset: 0x0C
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
	HSBAUDLOW[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	HSBAUD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	BAUDLOW[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BAUD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – HSBAUDLOW[7:0]: High Speed Master Baud Rate Low
HSBAUDLOW non-zero: HSBAUDLOW indicates the SCL low time in High-speed mode according to

$$\mathrm{HSBAUDLOW} = f_{\mathrm{GCLK}} \cdot T_{\mathrm{LOW}} - 1$$

HSBAUDLOW equal to zero: The HSBAUD register is used to time T_{LOW}, T_{HIGH}, $T_{SU;STO}$, $T_{HD;STA}$ and $T_{SU;STA}$. T_{BUF} is timed by the BAUD register.

Bits 23:16 – HSBAUD[7:0]: High Speed Master Baud Rate
This bit field indicates the SCL high time in High-speed mode according to the following formula. When HSBAUDLOW is zero, T_{LOW}, T_{HIGH}, $T_{SU;STO}$, $T_{HD;STA}$ and $T_{SU;STA}$ are derived using this formula. T_{BUF} is timed by the BAUD register.

$$\mathrm{HSBAUD} = f_{\mathrm{GCLK}} \cdot T_{\mathrm{HIGH}} - 1$$

Bits 15:8 – BAUDLOW[7:0]: Master Baud Rate Low
If this bit field is non-zero, the SCL low time will be described by the value written.

For more information on how to calculate the frequency, see SERCOM Clock Generation – Baud-Rate Generator on page 564.

Bits 7:0 – BAUD[7:0]: Master Baud Rate
This bit field is used to derive the SCL high time if BAUD.BAUDLOW is non-zero. If BAUD.BAUDLOW is zero, BAUD will be used to generate both high and low periods of the SCL.

For more information on how to calculate the frequency, see SERCOM Clock Generation – Baud-Rate Generator on page 564.

34.10.4. Interrupt Enable Clear

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Name: INTENCLR
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	ERROR						SB	MB
Access	R/W						R/W	R/W
Reset	0						0	0

Bit 7 – ERROR: Error Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Error Interrupt Enable bit, which disables the Error interrupt.

Value	Description
0	Error interrupt is disabled.
1	Error interrupt is enabled.

Bit 1 – SB: Slave on Bus Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Slave on Bus Interrupt Enable bit, which disables the Slave on Bus interrupt.

Value	Description
0	The Slave on Bus interrupt is disabled.
1	The Slave on Bus interrupt is enabled.

Bit 0 – MB: Master on Bus Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Master on Bus Interrupt Enable bit, which disables the Master on Bus interrupt.

Value	Description
0	The Master on Bus interrupt is disabled.
1	The Master on Bus interrupt is enabled.

34.10.5. Interrupt Enable Clear

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Name: INTENSET
Offset: 0x16
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	ERROR						SB	MB
Access	R/W						R/W	R/W
Reset	0						0	0

Bit 7 – ERROR: Error Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Error Interrupt Enable bit, which enables the Error interrupt.

Value	Description
0	Error interrupt is disabled.
1	Error interrupt is enabled.

Bit 1 – SB: Slave on Bus Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Slave on Bus Interrupt Enable bit, which enables the Slave on Bus interrupt.

Value	Description
0	The Slave on Bus interrupt is disabled.
1	The Slave on Bus interrupt is enabled.

Bit 0 – MB: Master on Bus Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Master on Bus Interrupt Enable bit, which enables the Master on Bus interrupt.

Value	Description
0	The Master on Bus interrupt is disabled.
1	The Master on Bus interrupt is enabled.

34.10.6. Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x18
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	ERROR						SB	MB
Access	R/W						R/W	R/W
Reset	0						0	0

Bit 7 – ERROR: Error
This flag is cleared by writing '1' to it.

This bit is set when any error is detected. Errors that will set this flag have corresponding status bits in the STATUS register. These status bits are LENERR, SEXTTOUT, MEXTTOUT, LOWTOUT, ARBLOST, and BUSERR.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the flag.

Bit 1 – SB: Slave on Bus
The Slave on Bus flag (SB) is set when a byte is successfully received in master read mode, i.e., no arbitration lost or bus error occurred during the operation. When this flag is set, the master forces the SCL line low, stretching the I^2C clock period. The SCL line will be released and SB will be cleared on one of the following actions:

- Writing to ADDR.ADDR
- Writing to DATA.DATA
- Reading DATA.DATA when smart mode is enabled (CTRLB.SMEN)
- Writing a valid command to CTRLB.CMD

Writing '1' to this bit location will clear the SB flag. The transaction will not continue or be terminated until one of the above actions is performed.

Writing '0' to this bit has no effect.

Bit 0 – MB: Master on Bus
This flag is set when a byte is transmitted in master write mode. The flag is set regardless of the occurrence of a bus error or an arbitration lost condition. MB is also set when arbitration is lost during sending of NACK in master read mode, or when issuing a start condition if the bus state is unknown. When this flag is set and arbitration is not lost, the master forces the SCL line low, stretching the I^2C clock period. The SCL line will be released and MB will be cleared on one of the following actions:

- Writing to ADDR.ADDR
- Writing to DATA.DATA
- Reading DATA.DATA when smart mode is enabled (CTRLB.SMEN)
- Writing a valid command to CTRLB.CMD

Writing '1' to this bit location will clear the MB flag. The transaction will not continue or be terminated until one of the above actions is performed.

Writing '0' to this bit has no effect.

34.10.7. Status

Name: STATUS
Offset: 0x1A
Reset: 0x0000
Property: Write-Synchronized

Bit	15	14	13	12	11	10	9	8
						LENERR	SEXTTOUT	MEXTTOUT
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	CLKHOLD	LOWTOUT	BUSSTATE[1:0]			RXNACK	ARBLOST	BUSERR
Access	R	R/W	R	R		R	R/W	R/W
Reset	0	0	0	0		0	0	0

Bit 10 – LENERR: Transaction Length Error
This bit is set when automatic length is used for a DMA transaction and the slave sends a NACK before ADDR.LEN bytes have been written by the master.

Writing '1' to this bit location will clear STATUS.LENERR. This flag is automatically cleared when writing to the ADDR register.

Writing '0' to this bit has no effect.

This bit is not write-synchronized.

Bit 9 – SEXTTOUT: Slave SCL Low Extend Time-Out
This bit is set if a slave SCL low extend time-out occurs.

This bit is automatically cleared when writing to the ADDR register.

Writing '1' to this bit location will clear SEXTTOUT. Normal use of the I^2C interface does not require the SEXTTOUT flag to be cleared by this method.

Writing '0' to this bit has no effect.

This bit is not write-synchronized.

Bit 8 – MEXTTOUT: Master SCL Low Extend Time-Out
This bit is set if a master SCL low time-out occurs.

Writing '1' to this bit location will clear STATUS.MEXTTOUT. This flag is automatically cleared when writing to the ADDR register.

Writing '0' to this bit has no effect.

This bit is not write-synchronized.

Bit 7 – CLKHOLD: Clock Hold
This bit is set when the master is holding the SCL line low, stretching the I^2C clock. Software should consider this bit when INTFLAG.SB or INTFLAG.MB is set.

This bit is cleared when the corresponding interrupt flag is cleared and the next operation is given.

Writing '0' to this bit has no effect.

Writing '1' to this bit has no effect.

This bit is not write-synchronized.

Bit 6 – LOWTOUT: SCL Low Time-Out
This bit is set if an SCL low time-out occurs.

Writing '1' to this bit location will clear this bit. This flag is automatically cleared when writing to the ADDR register.

Writing '0' to this bit has no effect.

This bit is not write-synchronized.

Bits 5:4 – BUSSTATE[1:0]: Bus State
These bits indicate the current I^2C bus state.

When in UNKNOWN state, writing 0x1 to BUSSTATE forces the bus state into the IDLE state. The bus state cannot be forced into any other state.

Writing BUSSTATE to idle will set SYNCBUSY.SYSOP.

Value	Name	Description
0x0	UNKNOWN	The bus state is unknown to the I^2C master and will wait for a stop condition to be detected or wait to be forced into an idle state by software
0x1	IDLE	The bus state is waiting for a transaction to be initialized
0x2	OWNER	The I^2C master is the current owner of the bus
0x3	BUSY	Some other I^2C master owns the bus

Bit 2 – RXNACK: Received Not Acknowledge
This bit indicates whether the last address or data packet sent was acknowledged or not.

Writing '0' to this bit has no effect.

Writing '1' to this bit has no effect.

This bit is not write-synchronized.

Value	Description
0	Slave responded with ACK.
1	Slave responded with NACK.

Bit 1 – ARBLOST: Arbitration Lost
This bit is set if arbitration is lost while transmitting a high data bit or a NACK bit, or while issuing a start or repeated start condition on the bus. The Master on Bus interrupt flag (INTFLAG.MB) will be set when STATUS.ARBLOST is set.

Writing the ADDR.ADDR register will automatically clear STATUS.ARBLOST.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear it.

This bit is not write-synchronized.

Bit 0 – BUSERR: Bus Error

This bit indicates that an illegal bus condition has occurred on the bus, regardless of bus ownership. An illegal bus condition is detected if a protocol violating start, repeated start or stop is detected on the I^2C bus lines. A start condition directly followed by a stop condition is one example of a protocol violation. If a time-out occurs during a frame, this is also considered a protocol violation, and will set BUSERR.

If the I^2C master is the bus owner at the time a bus error occurs, STATUS.ARBLOST and INTFLAG.MB will be set in addition to BUSERR.

Writing the ADDR.ADDR register will automatically clear the BUSERR flag.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear it.

This bit is not write-synchronized.

34.10.8. Synchronization Busy

Name: SYNCBUSY
Offset: 0x1C
Reset: 0x00000000
Property:

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						SYSOP	ENABLE	SWRST
Access						R	R	R
Reset						0	0	0

Bit 2 – SYSOP: System Operation Synchronization Busy
Writing CTRLB.CMD, STATUS.BUSSTATE, ADDR, or DATA when the SERCOM is enabled requires synchronization. When written, the SYNCBUSY.SYSOP bit will be set until synchronization is complete.

Value	Description
0	System operation synchronization is not busy.
1	System operation synchronization is busy.

Bit 1 – ENABLE: SERCOM Enable Synchronization Busy
Enabling and disabling the SERCOM (CTRLA.ENABLE) requires synchronization. When written, the SYNCBUSY.ENABLE bit will be set until synchronization is complete.

Writes to any register (except for CTRLA.SWRST) while enable synchronization is on-going will be discarded and an APB error will be generated.

Value	Description
0	Enable synchronization is not busy.
1	Enable synchronization is busy.

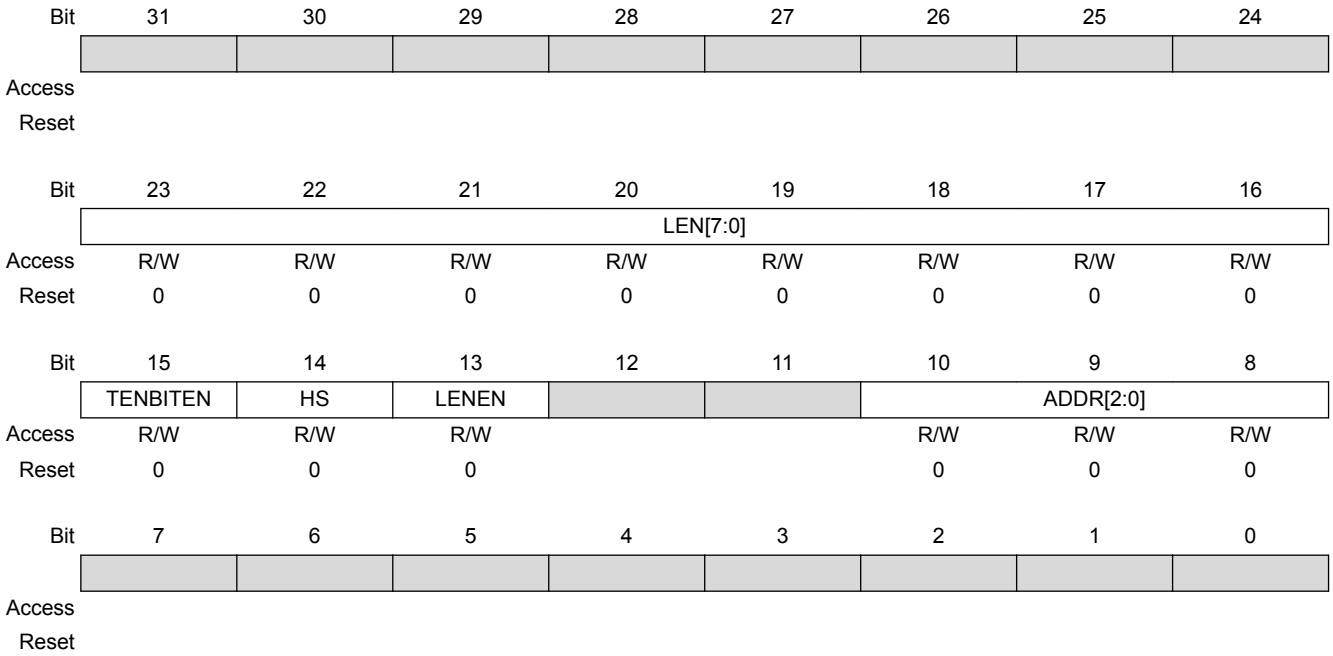
Bit 0 – SWRST: Software Reset Synchronization Busy
Resetting the SERCOM (CTRLA.SWRST) requires synchronization. When written, the SYNCBUSY.SWRST bit will be set until synchronization is complete.

Writes to any register while synchronization is on-going will be discarded and an APB error will be generated.

Value	Description
0	SWRST synchronization is not busy.
1	SWRST synchronization is busy.

34.10.9. Address

Name: ADDR
Offset: 0x24
Reset: 0x0000
Property: Write-Synchronized



Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	LEN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TENBITEN	HS	LENEN			ADDR[2:0]		
Access	R/W	R/W	R/W			R/W	R/W	R/W
Reset	0	0	0			0	0	0

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 23:16 – LEN[7:0]: Transaction Length
These bits define the transaction length of a DMA transaction from 0 to 255 bytes. The Transfer Length Enable (LENEN) bit must be written to '1' in order to use DMA.

Bit 15 – TENBITEN: Ten Bit Addressing Enable
This bit enables 10-bit addressing. This bit can be written simultaneously with ADDR to indicate a 10-bit or 7-bit address transmission.

Value	Description
0	10-bit addressing disabled.
1	10-bit addressing enabled.

Bit 14 – HS: High Speed
This bit enables High-speed mode for the current transfer from repeated START to STOP. This bit can be written simultaneously with ADDR for a high speed transfer.

Value	Description
0	High-speed transfer disabled.
1	High-speed transfer enabled.

Bit 13 – LENEN: Transfer Length Enable

Value	Description
0	Automatic transfer length disabled.
1	Automatic transfer length enabled.

Bits 10:8 – ADDR[2:0]: Address
When ADDR is written, the consecutive operation will depend on the bus state:

UNKNOWN: INTFLAG.MB and STATUS.BUSERR are set, and the operation is terminated.

BUSY: The I^2C master will await further operation until the bus becomes IDLE.

IDLE: The I^2C master will issue a start condition followed by the address written in ADDR. If the address is acknowledged, SCL is forced and held low, and STATUS.CLKHOLD and INTFLAG.MB are set.

OWNER: A repeated start sequence will be performed. If the previous transaction was a read, the acknowledge action is sent before the repeated start bus condition is issued on the bus. Writing ADDR to issue a repeated start is performed while INTFLAG.MB or INTFLAG.SB is set.

STATUS.BUSERR, STATUS.ARBLOST, INTFLAG.MB and INTFLAG.SB will be cleared when ADDR is written.

The ADDR register can be read at any time without interfering with ongoing bus activity, as a read access does not trigger the master logic to perform any bus protocol related operations.

The I^2C master control logic uses bit 0 of ADDR as the bus protocol's read/write flag (R/W); 0 for write and 1 for read.

34.10.10. Data

Name: DATA
Offset: 0x18
Reset: 0x0000
Property: Write-Synchronized, Read-Synchronized

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – DATA[7:0]: Data
The master data register I/O location (DATA) provides access to the master transmit and receive data buffers. Reading valid data or writing data to be transmitted can be successfully done only when SCL is held low by the master (STATUS.CLKHOLD is set). An exception is reading the last data byte after the stop condition has been sent.

Accessing DATA.DATA auto-triggers I^2C bus operations. The operation performed depends on the state of CTRLB.ACKACT, CTRLB.SMEN and the type of access (read/write).

Writing or reading DATA.DATA when not in smart mode does not require synchronization.

34.10.11. Debug Control

Name: DBGCTRL
Offset: 0x30
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGSTOP
Access								R/W
Reset								0

Bit 0 – DBGSTOP: Debug Stop Mode
This bit controls functionality when the CPU is halted by an external debugger.

Value	Description
0	The baud-rate generator continues normal operation when the CPU is halted by an external debugger.
1	The baud-rate generator is halted when the CPU is halted by an external debugger.

35. TC – Timer/Counter

35.1. Overview

There are up to five TC peripheral instances. Up to four TCs (TC[3:0]) are in PD1, whereas TC4, present in all device configurations, is always located in power domain PD0.

Each TC consists of a counter, a prescaler, compare/capture channels and control logic. The counter can be set to count events, or clock pulses. The counter, together with the compare/capture channels, can be configured to timestamp input events or IO pin edges, allowing for capturing of frequency and/or pulse width.

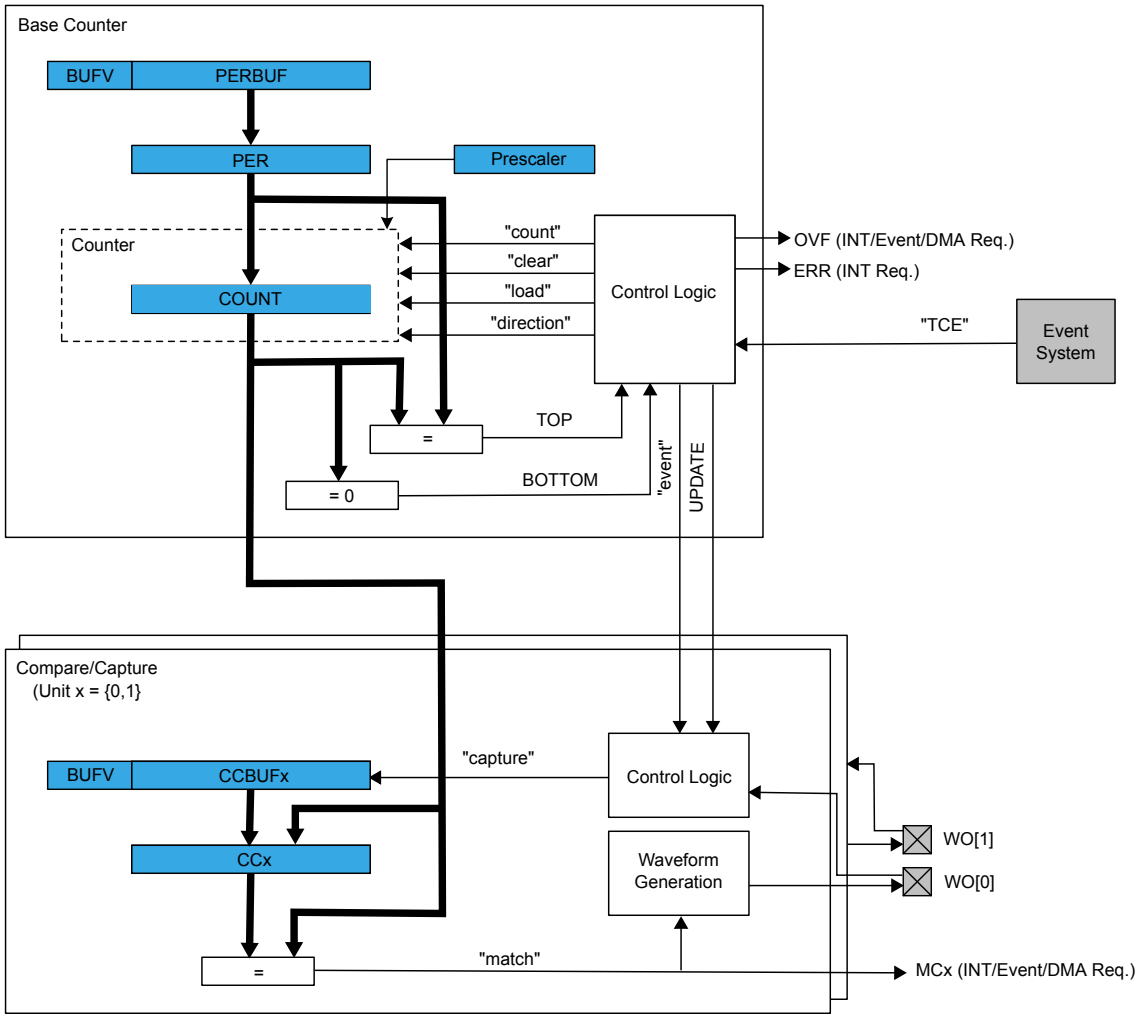
A TC can also perform waveform generation, such as frequency generation and pulse-width modulation.

35.2. Features

- Selectable configuration
  - 8-, 16- or 32-bit TC operation, with compare/capture channels
- 2 compare/capture channels (CC) with:
  - Double buffered timer period setting (in 8-bit mode only)
  - Double buffered compare channel
- Waveform generation
  - Frequency generation
  - Single-slope pulse-width modulation
- Input capture
  - Event / IO pin edge capture
  - Frequency capture
  - Pulse-width capture
  - Time-stamp capture
- One input event
- Interrupts/output events on:
  - Counter overflow/underflow
  - Compare match or capture
- Internal prescaler
- DMA support

35.3. Block Diagram

Figure 35-1. Timer/Counter Block Diagram



35.4. Signal Description

Table 35-1. Signal Description for TC.

Signal Name	Type	Description
WO[1:0]	Digital output	Waveform output
	Digital input	Capture input

Refer to *I/O Multiplexing and Considerations* for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

Related Links

I/O Multiplexing and Considerations on page 29

35.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

35.5.1. I/O Lines

In order to use the I/O lines of this peripheral, the I/O pins must be configured using the I/O Pin Controller (PORT).

Related Links

PORT: IO Pin Controller on page 506

35.5.2. Power Management

This peripheral can continue to operate in any sleep mode where its source clock is running. The interrupts can wake up the device from sleep modes. Events connected to the event system can trigger other operations in the system without exiting sleep modes. Refer to *PM – Power Manager* for details on the different sleep modes.

Related Links

PM – Power Manager on page 186

35.5.3. Clocks

The TC bus clocks (CLK_TCx_APB) can be enabled and disabled in the Power Manager. The default state of CLK_TCx_APB can be found in the *Peripheral Clock Masking*.

The generic clocks (GCLK_TCx) are asynchronous to the user interface clock (CLK_TCx_APB). Due to this asynchronicity, accessing certain registers will require synchronization between the clock domains. Refer to Synchronization for further details.

Note that TC0 and TC1 share a peripheral clock channel, as do TC2 and TC3. For this reason they cannot be set to different clock frequencies.

Related Links

Peripheral Clock Masking on page 151

35.5.4. DMA

The DMA request lines are connected to the DMA Controller (DMAC). In order to use DMA requests with this peripheral the DMAC must be configured first. Refer to *DMAC – Direct Memory Access Controller* for details.

Related Links

DMAC – Direct Memory Access Controller on page 399

35.5.5. Interrupts

The interrupt request line is connected to the Interrupt Controller. In order to use interrupt requests of this peripheral, the Interrupt Controller (NVIC) must be configured first. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

Nested Vector Interrupt Controller on page 51

35.5.6. Events

The events of this peripheral are connected to the Event System.

Related Links

EVSYS – Event System on page 536

35.5.7. Debug Operation

When the CPU is halted in debug mode, this peripheral will halt normal operation. This peripheral can be forced to continue operation during debugging - refer to the Debug Control (DBGCTRL) register for details.

35.5.8. Register Access Protection

Registers with write-access can be optionally write-protected by the Peripheral Access Controller (PAC), except the following:

- Interrupt Flag Status and Clear register (INTFLAG)
- Status register (STATUS)
- Count register (COUNT)
- Period and Period Buffer registers (PER, PERBUF)
- Compare/Capture Value registers and Compare/Capture Value Buffer registers (CCx, CCBUFx)

Note: Optional write-protection is indicated by the "PAC Write-Protection" property in the register description.

When the CPU is halted in debug mode, all write-protection is automatically disabled. Write-protection does not apply for accesses through an external debugger.

35.5.9. Analog Connections

Not applicable.

35.6. Functional Description

35.6.1. Principle of Operation

The following definitions are used throughout the documentation:

Table 35-2. Timer/Counter Definitions

Name	Description
TOP	The counter reaches TOP when it becomes equal to the highest value in the count sequence. The TOP value can be the same as Period (PER) or the Compare Channel 0 (CC0) register value depending on the waveform generator mode in Waveform Output Operations on page 711.
ZERO	The counter is ZERO when it contains all zeroes
MAX	The counter reaches MAX when it contains all ones
UPDATE	The timer/counter signals an update when it reaches ZERO or TOP, depending on the direction settings.
Timer	The timer/counter clock control is handled by an internal source

Name	Description
Counter	The clock control is handled externally (e.g. counting external events)
CC	For compare operations, the CC are referred to as “compare channels” For capture operations, the CC are referred to as “capture channels.”

Each TC instance has up to two compare/capture channels (CC0 and CC1).

The counter in the TC can either count events from the Event System, or clock ticks of the GCLK_TCx clock, which may be divided by the prescaler.

The counter value is passed to the CCx where it can be either compared to user-defined values or captured.

The Counter register (COUNT), compare and capture registers with buffers (CCx and CCBUFx) can be configured as 8-, 16- or 32-bit registers, with according MAX values. Mode settings determine the maximum range of the counter. Each buffer register has a buffer valid (BUFV) flag that indicates when the buffer contains a new value.

In 8-bit mode, Period Value (PER) and Period Buffer Value (PERBUF) registers are also available. The counter range and the operating frequency determine the maximum time resolution achievable with the TC peripheral.

The TC can be set to count up or down. Under normal operation, the counter value is continuously compared to the TOP or ZERO value to determine whether the counter has reached that value. On a comparison match the TC can request DMA transactions, or generate interrupts or events for the Event System.

In compare operation, the counter value is continuously compared to the values in the CCx registers. In case of a match the TC can request DMA transactions, or generate interrupts or events for the Event System. In waveform generator mode, these comparisons are used to set the waveform period or pulse width.

Capture operation can be enabled to perform input signal period and pulse width measurements, or to capture selectable edges from an IO pin or internal event from Event System.

35.6.2. Basic Operation

35.6.2.1. Initialization

The following registers are enable-protected, meaning that they can only be written when the TC is disabled (CTRLA.ENABLE =0):

- Control A register (CTRLA), except the Enable (ENABLE) and Software Reset (SWRST) bits
- Drive Control register (DRVCTRL)
- Wave register (WAVE)
- Event Control register (EVCTRL)

Enable-protected bits in the CTRLA register can be written at the same time as CTRLA.ENABLE is written to '1', but not at the same time as CTRLA.ENABLE is written to '0'. Enable-protection is denoted by the "Enable-Protected" property in the register description.

Before enabling the TC, the peripheral must be configured by the following steps:

1. Enable the TC bus clock (CLK_TCx_APB).
2. Select 8-, 16- or 32-bit counter mode via the TC Mode bit group in the Control A register (CTRLA.MODE). The default mode is 16-bit.

3. Select one wave generation operation in the Waveform Generation Operation bit group in the WAVE register (WAVE.WAVEGEN).
4. If desired, the GCLK_TCx clock can be prescaled via the Prescaler bit group in the Control A register (CTRLA.PRESCALER).
   – If the prescaler is used, select a prescaler synchronization operation via the Prescaler and Counter Synchronization bit group in the Control A register (CTRLA.PRESYNC).
5. If desired, select one-shot operation by writing a '1' to the One-Shot bit in the Control B Set register (CTRLBSET.ONESHOT).
6. If desired, configure the counting direction 'down' (starting from the TOP value) by writing a '1' to the Counter Direction bit in the Control B register (CTRLBSET.DIR).
7. For capture operation, enable the individual channels to capture in the Capture Channel x Enable bit group in the Control A register (CTRLA.CAPTEN).
8. If desired, enable inversion of the waveform output or IO pin input signal for individual channels via the Invert Enable bit group in the Drive Control register (DRVCTRL.INVEN).

35.6.2.2. Enabling, Disabling, and Resetting

The TC is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The TC is disbled by writing a zero to CTRLA.ENABLE.

The TC is reset by writing a '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the TC, except DBGCTRL, will be reset to their initial state. Refer to the CTRLA register for details.

The TC should be disabled before the TC is reset in order to avoid undefined behavior.

35.6.2.3. Prescaler Selection

The GCLK_TCx is fed into the internal prescaler.

The prescaler consists of a counter that counts up to the selected prescaler value, whereupon the output of the prescaler toggles.

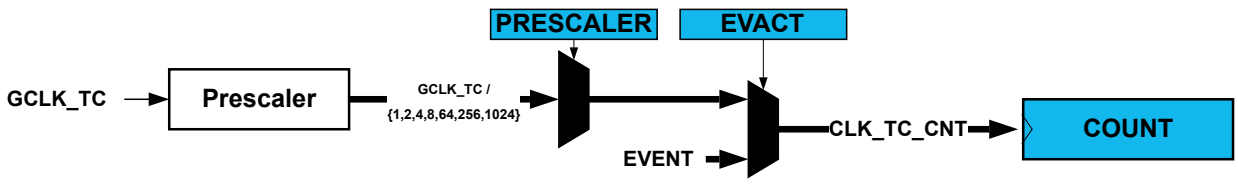
If the prescaler value is higher than one, the counter update condition can be optionally executed on the next GCLK_TCx clock pulse or the next prescaled clock pulse. For further details, refer to Prescaler (CTRLA.PRESCALER) and Counter Synchronization (CTRLA.PRESYNC) description.

Prescaler outputs from 1 to 1/1024 are available. For a complete list of available prescaler outputs, see the register description for the Prescaler bit group in the Control A register (CTRLA.PRESCALER).

Note: When counting events, the prescaler is bypassed.

The joint stream of prescaler ticks and event action ticks is called CLK_TC_CNT.

Figure 35-2. Prescaler



35.6.2.4. Counter Mode

The counter mode is selected by the Mode bit group in the Control A register (CTRLA.MODE). By default, the counter is enabled in the 16-bit counter resolution. Three counter resolutions are available:

- COUNT8: The 8-bit TC has its own Period Value and Period Buffer Value registers (PER and PERBUF).

- COUNT16: 16-bit is the default counter mode. There is no dedicated period register in this mode.
- COUNT32: This mode is achieved by pairing two 16-bit TC peripherals. TC0 is paired with TC1, and TC2 is paired with TC3. TC4 does not support 32-bit resolution.
  When paired, the TC peripherals are configured using the registers of the even-numbered TC (TC0 or TC2 respectively). The odd-numbered partner (TC1 or TC3 respectively) will act as slave, and the Slave bit in the Status register (STATUS.SLAVE) will be set. The register values of a slave will not reflect the registers of the 32-bit counter. Writing to any of the slave registers will not affect the 32-bit counter. Normal access to the slave COUNT and CCx registers is not allowed.

35.6.2.5. Counter Operations

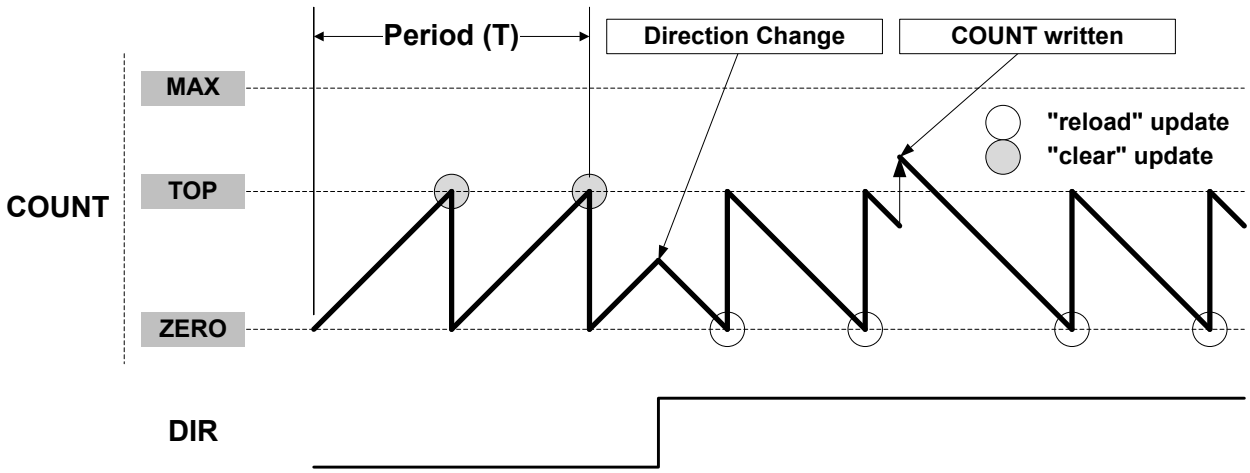
Depending on the mode of operation, the counter is cleared, reloaded, incremented, or decremented at each TC clock input (CLK_TC_CNT). A counter clear or reload marks the end of the current counter cycle and the start of a new one.

The counting direction is set by the Direction bit in the Control B register (CTRLB.DIR). If this bit is zero the counter is counting up, and counting down if CTRLB.DIR=1. The counter will count up or down for each tick (clock or event) until it reaches TOP or ZERO. When it is counting up and TOP is reached, the counter will be set to zero at the next tick (overflow) and the Overflow Interrupt Flag in the Interrupt Flag Status and Clear register (INTFLAG.OVF) will be set. When it is counting down, the counter is reloaded with the TOP value when ZERO is reached (underflow), and INTFLAG.OVF is set.

INTFLAG.OVF can be used to trigger an interrupt, a DMA request, or an event. An overflow/underflow occurrence (i.e. a compare match with TOP/ZERO) will stop counting if the One-Shot bit in the Control B register is set (CTRLBSET.ONESHOT).

It is possible to change the counter value (by writing directly in the COUNT register) even when the counter is running. When starting the TC, the COUNT value will be either ZERO or TOP (depending on the counting direction set by CTRLBSET.DIR or CTRLBCLR.DIR), unless a different value has been written to it, or the TC has been stopped at a value other than ZERO. The write access has higher priority than count, clear, or reload. The direction of the counter can also be changed during normal operation. See also the figure below.

Figure 35-3. Counter Operation



Due to asynchronous clock domains, the internal counter settings are written when the synchronization is complete. Normal operation must be used when using the counter as timer base for the capture channels.

Stop Command and Event Action

A Stop command can be issued from software by using Command bits in the Control B Set register (CTRLBSET.CMD = 0x2, STOP). When a Stop is detected while the counter is running, the counter will be loaded with the starting value (ZERO or TOP, depending on direction set by CTRLBSET.DIR or

CTRLBCLR.DIR). All waveforms are cleared and the Stop bit in the Status register is set (STATUS.STOP).

Re-Trigger Command and Event Action
A re-trigger command can be issued from software by writing the Command bits in the Control B Set register (CTRLBSET.CMD = 0x1, RETRIGGER), or from event when a re-trigger event action is configured in the Event Control register (EVCTRL.EVACT = 0x1, RETRIGGER).

When the command is detected during counting operation, the counter will be reloaded or cleared, depending on the counting direction (CTRLBSET.DIR or CTRLBCLR.DIR). When the re-trigger command is detected while the counter is stopped, the counter will resume counting from the current value in the COUNT register.

Note: When a re-trigger event action is configured in the Event Action bits in the Event Control register (EVCTRL.EVACT=0x1, RETRIGGER), enabling the counter will not start the counter. The counter will start on the next incoming event and restart on corresponding following event.

Count Event Action
The TC can count events. When an event is received, the counter increases or decreases the value, depending on direction settings (CTRLBSET.DIR or CTRLBCLR.DIR). The count event action can be selected by the Event Action bit group in the Event Control register (EVCTRL.EVACT=0x2, COUNT).

Start Event Action
The TC can start counting operation on an event when previously stopped. In this configuration, the event has no effect if the counter is already counting. When the peripheral is enabled, the counter operation starts when the event is received or when a re-trigger software command is applied.

The Start TC on Event action can be selected by the Event Action bit group in the Event Control register (EVCTRL.EVACT=0x3, START).

35.6.2.6. Compare Operations

By default, the Compare/Capture channel is configured for compare operations.

When using the TC and the Compare/Capture Value registers (CCx) for compare operations, the counter value is continuously compared to the values in the CCx registers. This can be used for timer or for waveform operation.

The Channel x Compare Buffer (CCBUFx) registers provide double buffer capability. The double buffering synchronizes the update of the CCx register with the buffer value at the UPDATE condition or a forced update command (CTRLBSET.CMD=UPDATE). For further details, refer to Double Buffering on page 714. The synchronization prevents the occurrence of odd-length, non-symmetrical pulses and ensures glitch-free output.

Waveform Output Operations
The compare channels can be used for waveform generation on output port pins. To make the waveform available on the connected pin, the following requirements must be fulfilled:

1. Choose a waveform generation mode in the Waveform Generation Operation bit in Waveform register (WAVE.WAVEGEN).
2. Optionally invert the waveform output WO[x] by writing the corresponding Output Waveform x Invert Enable bit in the Driver Control register (DRVCTRL.INVENx).
3. Configure the pins with the I/O Pin Controller. Refer to *PORT - I/O Pin Controller* for details.

The counter value is continuously compared with each CCx value. On a comparison match, the Match or Capture Channel x bit in the Interrupt Flag Status and Clear register (INTFLAG.MCx) will be set on the next zero-to-one transition of CLK_TC_CNT (see Normal Frequency Operation). An interrupt/and or event can be generated on comparison match if enabled. The same condition generates a DMA request.

There are four waveform configurations for the Waveform Generation Operation bit group in the Waveform register (WAVE.WAVEGEN). This will influence how the waveform is generated and impose restrictions on the top value. The configurations are:

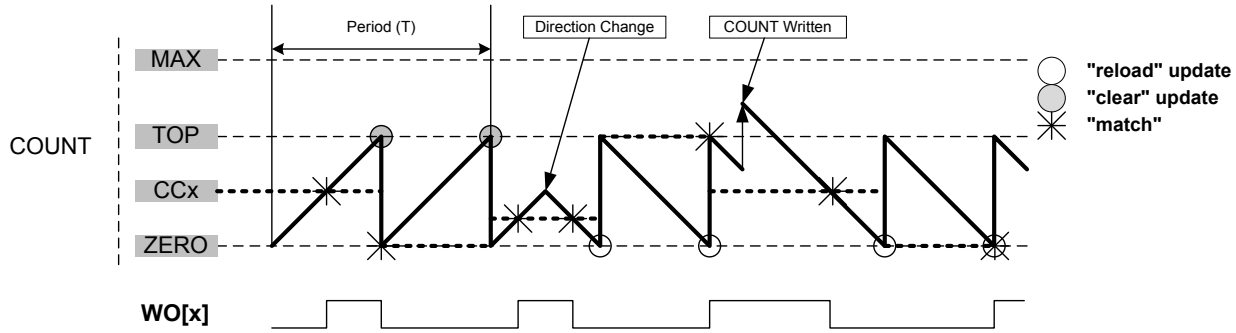
- Normal frequency (NFRQ)
- Match frequency (MFRQ)
- Normal pulse-width modulation (NPWM)
- Match pulse-width modulation (MPWM)

When using NPWM or NFRQ configuration, the TOP will be determined by the counter resolution. In 8-bit counter mode, the Period register (PER) is used as TOP, and the TOP can be changed by writing to the PER register. In 16- and 32-bit counter mode, TOP is fixed to the maximum (MAX) value of the counter.

Normal Frequency Generation (NFRQ)

For Normal Frequency Generation, the period time (T) is controlled by the period register (PER) for 8-bit counter mode and MAX for 16- and 32-bit mode. The waveform generation output (WO[x]) is toggled on each compare match between COUNT and CCx, and the corresponding Match or Capture Channel x Interrupt Flag (INTFLAG.MCx) will be set.

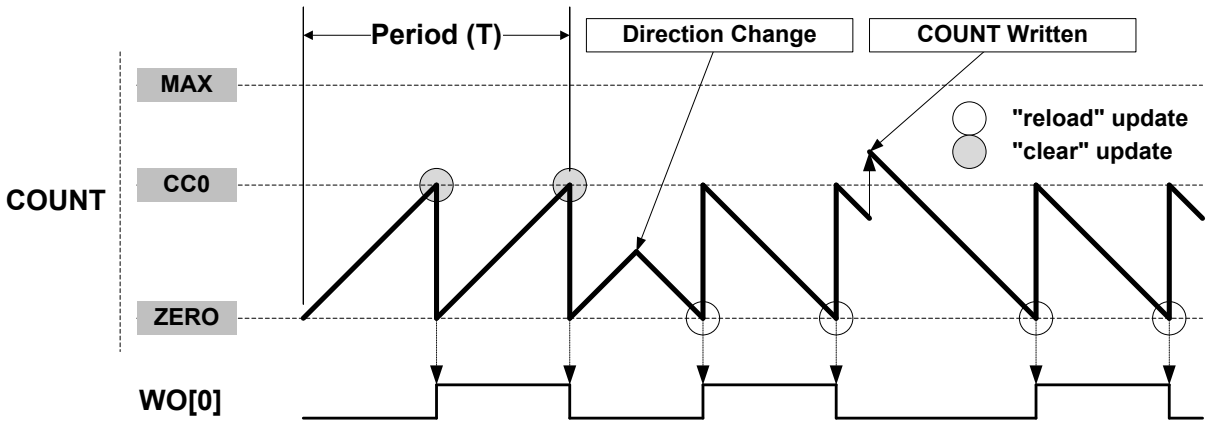
Figure 35-4. Normal Frequency Operation



Match Frequency Generation (MFRQ)

For Match Frequency Generation, the period time (T) is controlled by the CC0 register instead of PER or MAX. WO[0] toggles on each update condition.

Figure 35-5. Match Frequency Operation



Normal Pulse-Width Modulation Operation (NPWM)

NPWM uses single-slope PWM generation.

For single-slope PWM generation, the period time (T) is controlled by the TOP value, and CCx controls the duty cycle of the generated waveform output. When up-counting, the WO[x] is set at start or compare match between the COUNT and TOP values, and cleared on compare match between COUNT and CCx

register values. When down-counting, the WO[x] is cleared at start or compare match between the COUNT and ZERO values, and set on compare match between COUNT and CCx register values.

The following equation calculates the exact resolution for a single-slope PWM (R_{PWM_SS}) waveform:

$$R_{PWM\_SS} = \frac{\log(TOP+1)}{\log(2)}$$

The PWM frequency (f_{PWM_SS}) depends on TOP value and the peripheral clock frequency (f_{GCLK_TCC}), and can be calculated by the following equation:

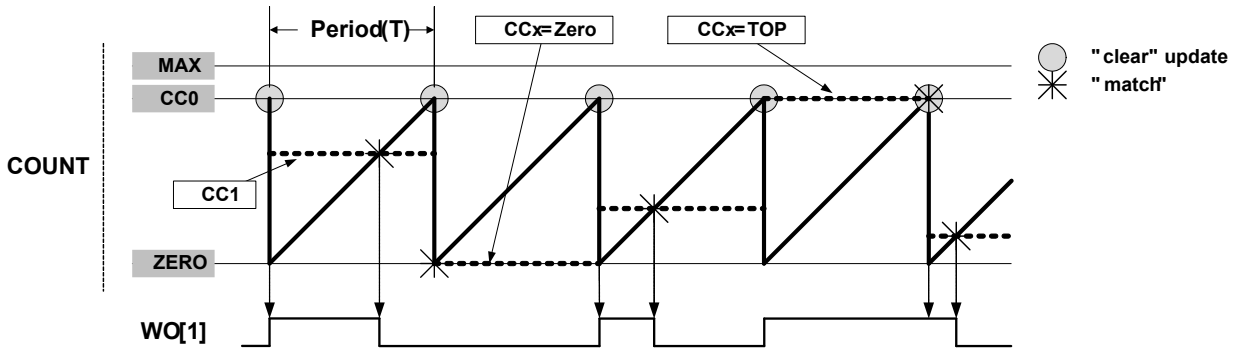
$$f_{PWM\_SS} = \frac{f_{GCLK\_TC}}{N(TOP+1)}$$

Where N represents the prescaler divider used (1, 2, 4, 8, 16, 64, 256, 1024).

Match Pulse-Width Modulation Operation (MPWM)

In MPWM, the output of WO[1] is depending on CC1 as shown in the figure below. On on every overflow/underflow, a one-TC-clock-cycle negative pulse is put out on WO[0] (not shown in the figure).

Figure 35-6. Match PWM Operation



The table below shows the update counter and overflow event/interrupt generation conditions in different operation modes.

Table 35-3. Counter Update and Overflow Event/interrupt Conditions in TC

Name	Operation	TOP	Update	Output Waveform		OVFIF/Event	
				On Match	On Update	Up	Down
NFRQ	Normal Frequency	PER	TOP/ ZERO	Toggle	Stable	TOP	ZERO
MFRQ	Match Frequency	CC0	TOP/ ZERO	Toggle	Stable	TOP	ZERO
NPWM	Single-slope PWM	PER	TOP/ ZERO	See description above.		TOP	ZERO
MPWM	Single-slope PWM	CC0	TOP/ ZERO	Toggle	Toggle	TOP	ZERO

Related Links

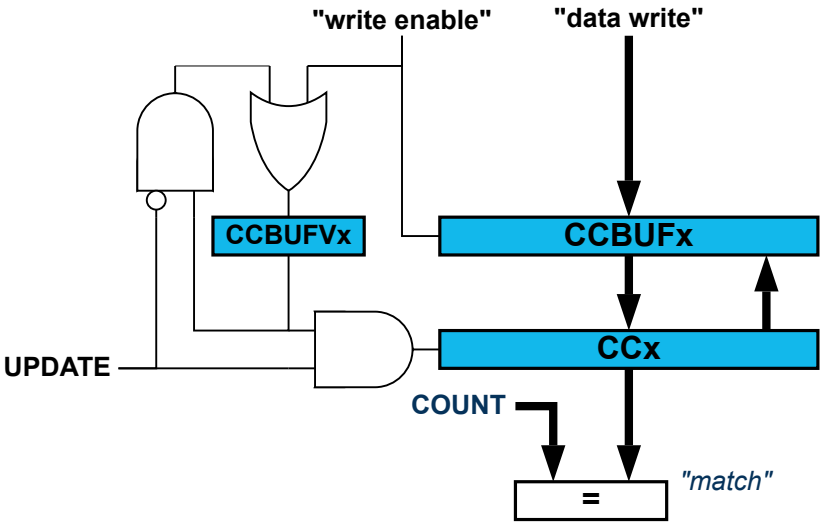
PORT: IO Pin Controller on page 506

35.6.2.7. Double Buffering

The Compare Channels (CCx) registers, and the Period (PER) register in 8-bit mode, are double buffered. Each buffer register has a buffer valid (CCBUFVx or PERBUFV) bit in the STATUS register, which indicates that the buffer register contains a new valid value that can be copied into the corresponding register. As long as the buffer valid status flag (PERBUFV or CCBUFVx) is set to '1', a write to the corresponding PER or CCx register will generate a Capture Overflow Error (ERR).

When the buffer valid flag bit in the STATUS register is '1' and the Lock Update bit in the CTRLB register is set to '0' by writing CTRLBCLR.LUPD to '1', double buffering is enabled: the data from buffer registers will be copied into the corresponding register under UPDATE condition (CTRLBSET.CMD=0x3), including the software update command. Then the buffer valid flags bit in the STATUS register are automatically cleared by hardware or software. Double buffering is not applied for capture mode. A compare register is double buffered as in the following figure.

Figure 35-7. Compare Channel Double Buffering



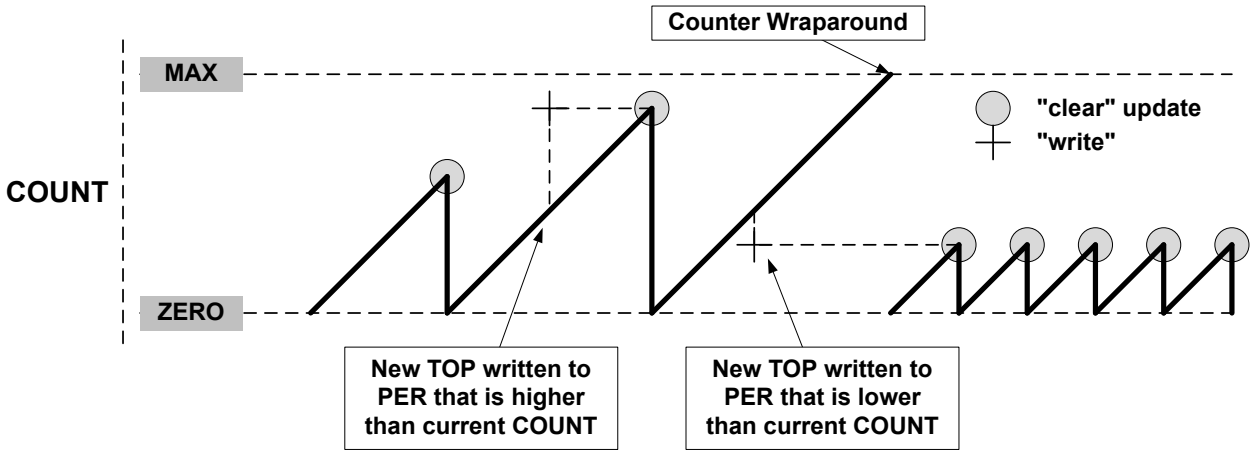
Both the registers (PER/CCx) and corresponding buffer registers (PERBUF/CCBUFx) are available in the I/O register map, and the double buffering feature is not mandatory. The double buffering is disabled by writing a '1' to CTRLBSET.LUPD. This allows initialization and bypassing of the buffer register and the double buffering feature.

Note: In NFRQ, MFRQ or PWM down-counting counter mode (CTRLBSET.DIR=1), the PER register and the PERBUF register are written simultaneously if double buffering is active or as soon as double buffering is activated (CTRLBCLR.LUPD=0).

Changing the Period

The counter period is changed by writing a new TOP value to the Period register (PER or CC0, depending on the waveform generation mode). If double buffering is off (CTRLBSET.LUPD=1), any period update is effective after the synchronization delay.

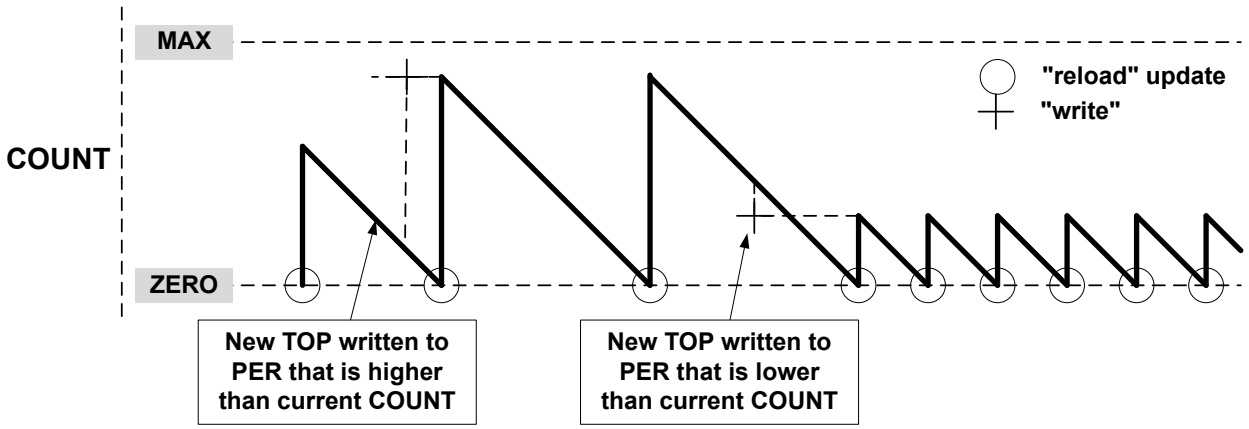
Figure 35-8. Unbuffered Single-Slope Up-Counting Operation



A counter wraparound can occur in any operation mode when up-counting without buffering, see Figure 35-8 Unbuffered Single-Slope Up-Counting Operation on page 715.

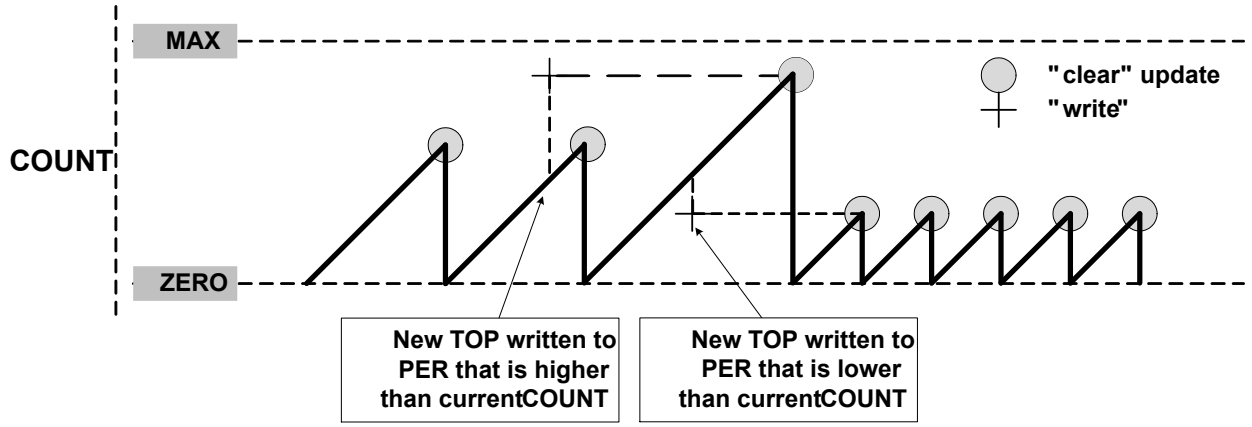
COUNT and TOP are continuously compared, so when a new TOP value that is lower than current COUNT is written to TOP, COUNT will wrap before a compare match.

Figure 35-9. Unbuffered Single-Slope Down-Counting Operation



When double buffering is used, the buffer can be written at any time and the counter will still maintain correct operation. The period register is always updated on the update condition, as shown in Figure 35-10 Changing the Period Using Buffering on page 716. This prevents wraparound and the generation of odd waveforms.

Figure 35-10. Changing the Period Using Buffering



35.6.2.8. Capture Operations

To enable and use capture operations, the corresponding Capture Channel x Enable bit in the Control A register (CTRLA.CAPTENx) must be written to '1'.

A capture trigger can be provided by asynchronous IO pin WO[x] for each capture channel or by a TC event. To enable the capture from the IO pin, the Capture On Pin x Enable bit in CTRLA register (CTRLA.COPENx) must be written to '1'.

Note: The RETRIGGER, COUNT and START event actions are available only on an event from the Event System.

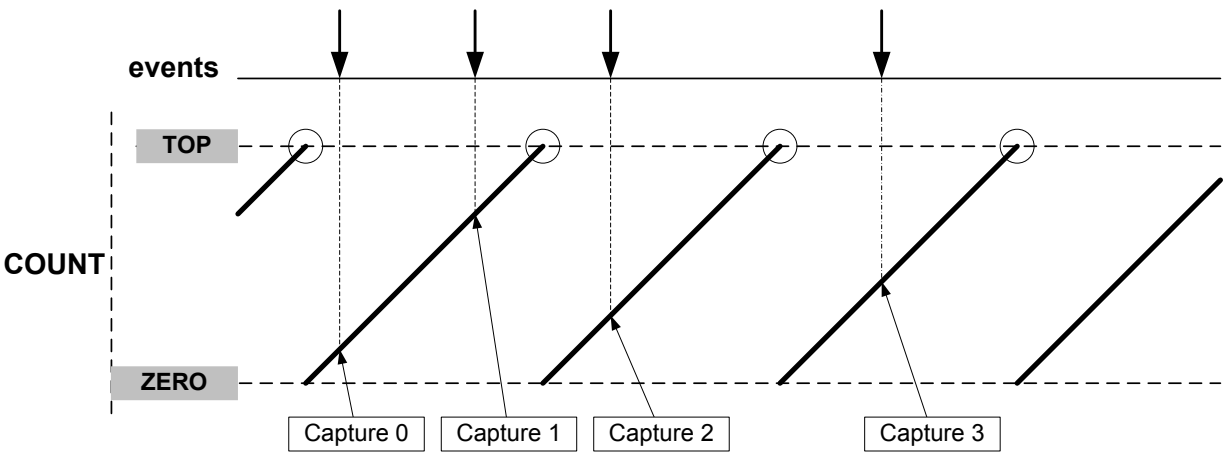
By default, a capture operation is done when a rising edge is detected on the input signal. Capture on falling edge is available, its activation is depending on the input source:

- When the channel is used with a IO pin, write a '1' to the corresponding Invert Enable bit in the Drive Control register (DRVCTRL.INVENx).
- When the channel is counting events from the Event System, write a '1' to the TC Event Input Invert Enable bit in Event Control register (EVCTRL.TCINV).

Event Capture Action

The compare/capture channels can be used as input capture channels to capture events from the Event System or from the corresponding IO pin, and give them a timestamp. The following figure shows four capture events for one capture channel.

Figure 35-11. Input Capture Timing



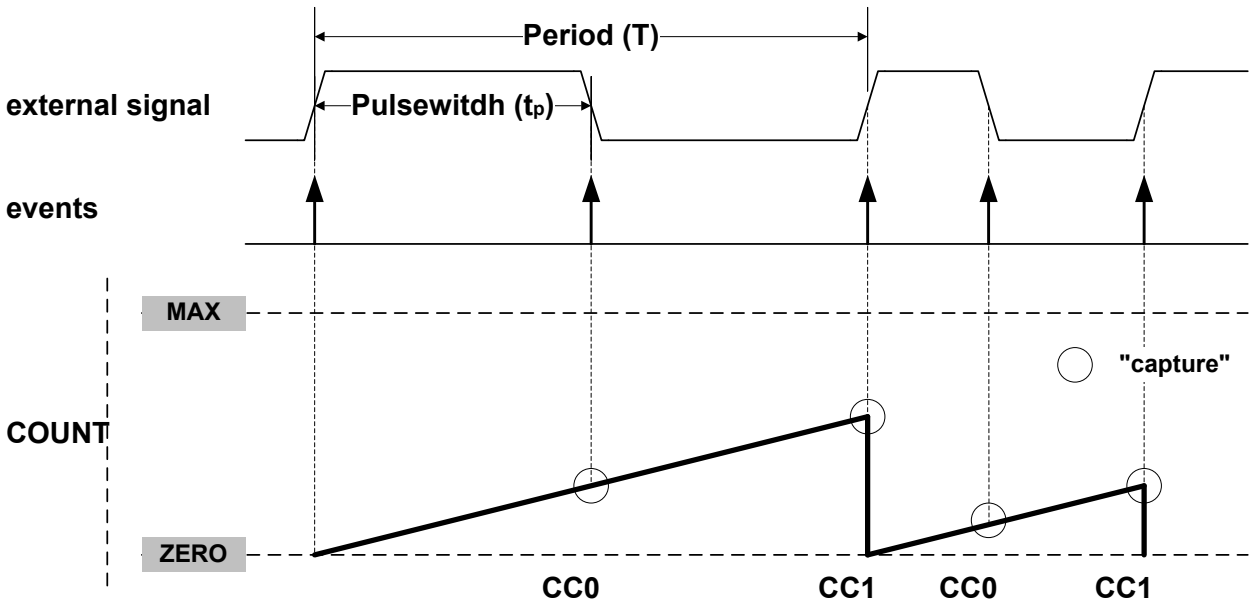
The TC can detect capture overflow of the input capture channels: When a new capture event is detected while the Capture Interrupt flag (INTFLAG.MCx) is still set, the new timestamp will not be stored and INTFLAG.ERR will be set.

Period and Pulse-Width (PPW) Capture Action

The TC can perform two input captures and restart the counter on one of the edges. This enables the TC to measure the pulse width and period and to characterize the frequency *f* and duty cycle of an input signal:

$$f = \frac{1}{T} \qquad \text{dutyCycle} = \frac{t_p}{T}$$

Figure 35-12. PWP Capture



Selecting PWP in the Event Action bit group in the Event Control register (EVCTRL.EVACT) enables the TC to perform one capture action on the rising edge and the other one on the falling edge. The period *T* will be captured into CC1 and the pulse width t_p in CC0. EVCTRL.EVACT=PPW (period and pulse-width)offers identical functionality, but will capture *T* into CC0 and t_p into CC1.

The TC Event Input Invert Enable bit in the Event Control register (EVCTRL.TCINV) is used to select whether the wraparound should occur on the rising edge or the falling edge. If EVCTRL.TCINV=1, the wraparound will happen on the falling edge. This also be for DRVCTRL.INVENx if pin capture is enabled.

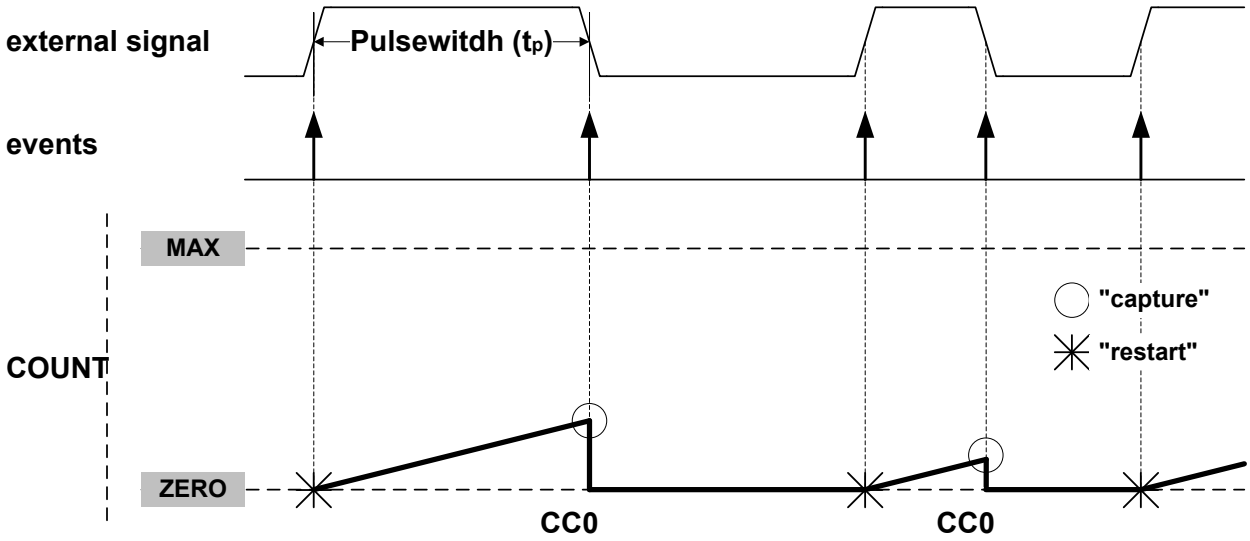
The TC can detect capture overflow of the input capture channels: When a new capture event is detected while the Capture Interrupt flag (INTFLAG.MCx) is still set, the new timestamp will not be stored and INTFLAG.ERR will be set.

Note: The corresponding capture is working only if the channel is enabled in capture mode (CTRLA.CAPTENx=1). If not, the capture action is ignored and the channel is enabled in compare mode of operation. Consequently, both channels must be enabled in order to fully characterize the input.

Pulse-Width Capture Action

The TC performs the input capture on the falling edge of the input signal. When the edge is detected, the counter value is cleared and the TC stops counting. When a rising edge is detected on the input signal, the counter restarts the counting operation. To enable the operation on opposite edges, the input signal to capture must be inverted (refer to DRVCTRL.INVEN or EVCTRL.TCEINV).

Figure 35-13. Pulse-Width Capture on Channel 0



The TC can detect capture overflow of the input capture channels: When a new capture event is detected while the Capture Interrupt flag (INTFLAG.MCx) is still set, the new timestamp will not be stored and INTFLAG.ERR will be set.

35.6.3. Additional Features

35.6.3.1. One-Shot Operation

When one-shot is enabled, the counter automatically stops on the next counter overflow or underflow condition. When the counter is stopped, the Stop bit in the Status register (STATUS.STOP) is automatically set and the waveform outputs are set to zero.

One-shot operation is enabled by writing a '1' to the One-Shot bit in the Control B Set register (CTRLBSET.ONESHOT), and disabled by writing a '1' to CTRLBCLR.ONESHOT. When enabled, the TC will count until an overflow or underflow occurs and stops counting operation. The one-shot operation can be restarted by a re-trigger software command, a re-trigger event, or a start event. When the counter restarts its operation, STATUS.STOP is automatically cleared.

35.6.3.2. Time-Stamp Capture

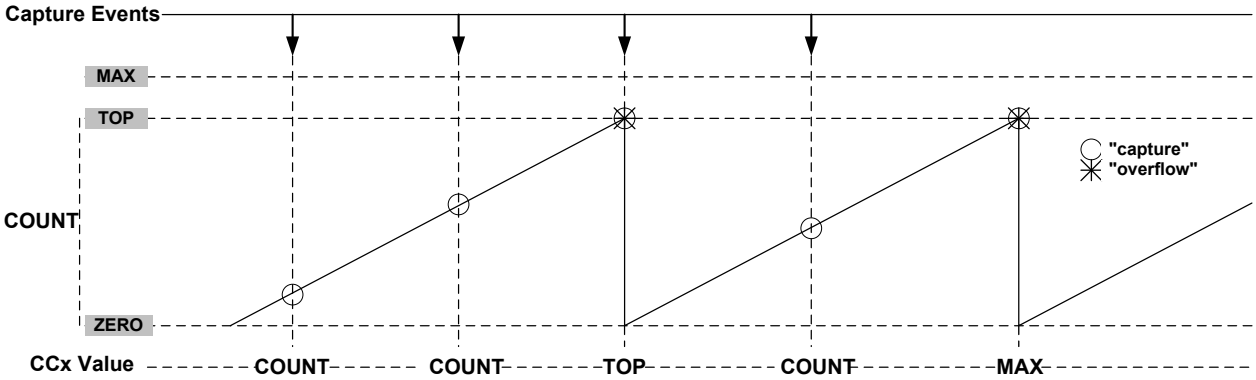
This feature is enabled when the Capture Time Stamp (STAMP) Event Action in Event Control register (EVCTRL.EVACT) is selected. The counter TOP value must be smaller than MAX.

When a capture event is detected, the COUNT value is copied into the corresponding Channel x Compare/Capture Value (CCx) register. In case of an overflow, the MAX value is copied into the corresponding CCx register.

When a valid captured value is present in the capture channel register, the corresponding Capture Channel x Interrupt Flag (INTFLAG.MCx) is set.

The timer/counter can detect capture overflow of the input capture channels: When a new capture event is detected while the Capture Channel interrupt flag (INTFLAG.MCx) is still set, the new time-stamp will not be stored and INTFLAG.ERR will be set.

Figure 35-14. Time-Stamp



35.6.4. DMA Operation

The TC can generate the following DMA requests:

- Overflow (OVF): the request is set when an update condition (overflow, underflow or re-trigger) is detected, the request is cleared by hardware on DMA acknowledge.
- Match or Capture Channel x (MCx): for a compare channel, the request is set on each compare match detection, the request is cleared by hardware on DMA acknowledge. For a capture channel, the request is set when valid data is present in the CCx register, and cleared when CCx register is read.

35.6.5. Interrupts

The TC has the following interrupt sources:

- Overflow/Underflow (OVF)
- Match or Capture Channel x (MCx)
- Capture Overflow Error (ERR)

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) is set when the interrupt condition occurs.

Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR).

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until either the interrupt flag is cleared, the interrupt is disabled, or the TC is reset. See INTFLAG for details on how to clear interrupt flags.

The TC has one common interrupt request line for all the interrupt sources. The user must read the INTFLAG register to determine which interrupt condition is present.

Note that interrupts must be globally enabled for interrupt requests to be generated. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

Nested Vector Interrupt Controller on page 51

35.6.6. Events

The TC can generate the following output events:

- Overflow/Underflow (OVF)
- Match or Capture Channel x (MCx)

Writing a '1' to an Event Output bit in the Event Control register (EVCTRL.MCEOx) enables the corresponding output event. The output event is disabled by writing EVCTRL.MCEOx=0.

One of the following event actions can be selected by the Event Action bit group in the Event Control register (EVCTRL.EVACT):

- Disable event action (OFF)
- Start TC (START)
- Re-trigger TC (RETRIGGER)
- Count on event (COUNT)
- Capture time stamp (STAMP)
- Capture Period (PPW and PWP)
- Capture Pulse Width (PW)

Writing a '1' to the TC Event Input bit in the Event Control register (EVCTRL.TCEI) enables input events to the TC. Writing a '0' to this bit disables input events to the TC. The TC requires only asynchronous event inputs. For further details on how configuring the asynchronous events, refer to *EVSYS - Event System*.

Related Links

EVSYS – Event System on page 536

35.6.7. Sleep Mode Operation

The TC can be configured to operate in any sleep mode. To be able to run in standby, the RUNSTDBY bit in the Control A register (CTRLA.RUNSTDBY) must be '1'. This peripheral can wake up the device from any sleep mode using interrupts or perform actions through the Event System.

If the On Demand bit in the Control A register (CTRLA.ONDEMAND) is written to '1', the module stops requesting its peripheral clock when the STOP bit in STATUS register (STATUS.STOP) is set to '1'. When a re-trigger or start condition is detected, the TC requests the clock before the operation starts.

35.6.8. Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read. An exception is the Channel Enable bit in the Peripheral Channel Control registers (PCHCTRLm.CHEN). When changing this bit, the bit value must be read-back to ensure the synchronization is complete and to assert glitch free internal operation. Note that changing the bit value under ongoing synchronization will *not* generate an error.

The following bits are synchronized when written:

- Software Reset and Enable bits in Control A register (CTRLA.SWRST and CTRLA.ENABLE)
- Capture Channel Buffer Valid bit in STATUS register (STATUS.CCBUFVx)

The following registers are synchronized when written:

- Control B Clear and Control B Set registers (CTRLBCLR and CTRLBSET)
- Count Value register (COUNT)
- Period Value register (PER)
- Channel x Compare/Capture Value registers (CCx)

The following registers are synchronized when read:

- Count Value register (COUNT): synchronization is done on demand through READSYNC command (CTRLBSET.CMD).

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

Required read-synchronization is denoted by the "Read-Synchronized" property in the register description.

35.7. Register Summary

Table 35-4. Register Summary – 8-bit Mode

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0	ONDEMAND	RUNSTDBY	PRESCSYNC[1:0]		MODE[1:0]		ENABLE	SWRST
0x01		15:8					ALOCK	PRESCALER[2:0]		
0x02		23:16			COPEN1	COPEN0			CAPTEN1	CAPTEN0
0x03		31:24								
0x04	CTRLBCLR	7:0	CMD[2:0]					ONESHOT	LUPD	DIR
0x05	CTRLBSET	7:0	CMD[2:0]					ONESHOT	LUPD	DIR
0x06	EVCTRL	7:0			TCEI	TCINV		EVACT[2:0]		
0x07		15:8			MCEO1	MCEO0				OVFEO
0x08	INTENCLR	7:0			MC1	MC0			ERR	OVF
0x09	INTENSET	7:0			MC1	MC0			ERR	OVF
0x0A	INTFLAG	7:0			MC1	MC0			ERR	OVF
0x0B	STATUS	7:0			CCBUFV1	CCBUFV0	PERBUFV		SLAVE	STOP
0x0C	WAVE	7:0							WAVEGEN[1:0]	
0x0D	DRVCTRL	7:0							INVEN1	INVEN0
0x0E	Reserved									
0x0F	DBGCTRL	7:0								DBGRUN
0x10	SYNCBUSY	7:0	CC1	CC0	PER	COUNT	STATUS	CTRLB	ENABLE	SWRST
0x11		15:8								
0x12		23:16								
0x13		31:24								
0x14	COUNT	7:0	COUNT[7:0]							
0x15	Reserved									
0x16	Reserved									
0x17	Reserved									
0x18	Reserved									
0x19	Reserved									
0x1A	Reserved									
0x1B	PER	7:0	PER[7:0]							
0x1C	CC0	7:0	CC[7:0]							
0x1D	CC1	7:0	CC[7:0]							
0x1E	Reserved									
0x1F	Reserved									
0x20	Reserved									
0x21	Reserved									
0x22	Reserved									
0x23	Reserved									
0x24	Reserved									
0x25	Reserved									

Offset	Name	Bit Pos.								
0x26	Reserved									
0x27	Reserved									
0x28	Reserved									
0x29	Reserved									
0x2A	Reserved									
0x2B	Reserved									
0x2C	Reserved									
0x2D	Reserved									
0x2E	Reserved									
0x2F	PERBUF	7:0	PERBUF[7:0]							
0x30	CCBUF0	7:0	CCBUF[7:0]							
0x31	CCBUF1	7:0	CCBUF[7:0]							
0x32	Reserved									
0x33	Reserved									

Table 35-5. Register Summary – 16-bit Mode

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0	ONDEMAND	RUNSTDBY	PRESCSYNC[1:0]		MODE[1:0]		ENABLE	SWRST
0x01		15:8					ALOCK	PRESCALER[2:0]		
0x02		23:16			COPEN1	COPEN0			CAPTEN1	CAPTEN0
0x03		31:24								
0x04	CTRLBCLR	7:0	CMD[2:0]					ONESHOT	LUPD	DIR
0x05	CTRLBSET	7:0	CMD[2:0]					ONESHOT	LUPD	DIR
0x06	EVCTRL	7:0			TCEI	TCINV		EVACT[2:0]		
0x07		15:8			MCEO1	MCEO0				OVFEO
0x08	INTENCLR	7:0			MC1	MC0			ERR	OVF
0x09	INTENSET	7:0			MC1	MC0			ERR	OVF
0x0A	INTFLAG	7:0			MC1	MC0			ERR	OVF
0x0B	STATUS	7:0			CCBUFV1	CCBUFV0			SLAVE	STOP
0x0C	WAVE								WAVEGEN[1:0]	
0x0D	DRVCTRL	7:0							INVEN1	INVEN0
0x0E	Reserved									
0x0F	DBGCTRL	7:0								DBGRUN
0x10	SYNCBUSY	7:0	CC1	CC0		COUNT	STATUS	CTRLB	ENABLE	SWRST
0x11		15:8								
0x12		23:16								
0x13		31:24								
0x14	COUNT	7:0	COUNT[7:0]							
0x15		15:8	COUNT[15:8]							
0x16	Reserved									
0x17	Reserved									
0x18	Reserved									
0x19	Reserved									
0x1A	Reserved									
0x1B	Reserved									
0x1C	CC0	7:0	CC[7:0]							
0x1D		15:8	CC[15:8]							

Offset	Name	Bit Pos.								
0x1E	CC1	7:0	CC[7:0]							
0x1F		15:8	CC[5:8]							
0x20	Reserved									
0x21	Reserved									
0x22	Reserved									
0x23	Reserved									
0x24	Reserved									
0x25	Reserved									
0x26	Reserved									
0x27	Reserved									
0x28	Reserved									
0x29	Reserved									
0x2A	Reserved									
0x2B	Reserved									
0x2C	Reserved									
0x2D	Reserved									
0x2E	Reserved									
0x2F	Reserved									
0x30	CCBUF0	7:0	CCBUF[7:0]							
0x31		15:8	CCBUF[15:8]							
0x32	CCBUF1	7:0	CCBUF[7:0]							
0x33		15:8	CCBUF[5:8]							
0x34	Reserved									
0x35	Reserved									
0x36	Reserved									
0x37	Reserved									

Table 35-6. Register Summary – 32-bit Mode

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0	ONDEMAND	RUNSTDBY	PRESCSYNC[1:0]		MODE[1:0]		ENABLE	SWRST
0x01		15:8					ALOCK	PRESCALER[2:0]		
0x02		23:16			COPEN1	COPEN0			CAPTEN1	CAPTEN0
0x03		31:24								
0x04	CTRLBCLR	7:0	CMD[2:0]					ONESHOT	LUPD	DIR
0x05	CTRLBSET	7:0	CMD[2:0]					ONESHOT	LUPD	DIR
0x06	EVCTRL	7:0			TCEI	TCINV		EVACT[2:0]		
0x07		15:8			MCEO1	MCEO0				OVFEO
0x08	INTENCLR	7:0			MC1	MC0			ERR	OVF
0x09	INTENSET	7:0			MC1	MC0			ERR	OVF
0x0A	INTFLAG	7:0			MC1	MC0			ERR	OVF
0x0B	STATUS	7:0			CCBUFV1	CCBUFV0			SLAVE	STOP
0x0C	WAVE	7:0							WAVEGEN[1:0]	
0x0D	DRVCTRL	7:0							INVEN1	INVEN0
0x0E	Reserved									
0x0F	DBGCTRL	7:0								DBGRUN

Offset	Name	Bit Pos.								
0x10	SYNCBUSY	7:0	CC1	CC0		COUNT	STATUS	CTRLB	ENABLE	SWRST
0x11		15:8								
0x12		23:16								
0x13		31:24								
0x14	COUNT	7:0	COUNT[7:0]							
0x15		15:8	COUNT[15:8]							
0x16		23:16	COUNT[23:16]							
0x17		31:24	COUNT[31:24]							
0x18	Reserved									
0x19	Reserved									
0x1A	Reserved									
0x1B	Reserved									
0x1C	CC0	7:0	CC[7:0]							
0x1D		15:8	CC[15:8]							
0x1E		23:16	CC[23:16]							
0x1F		31:24	CC[31:24]							
0x20	CC1	7:0	CC[7:0]							
0x21		15:8	CC[15:8]							
0x22		23:16	CC[23:16]							
0x23		31:24	CC[31:24]							
0x24	Reserved									
0x25	Reserved									
0x26	Reserved									
0x27	Reserved									
0x28	Reserved									
0x29	Reserved									
0x2A	Reserved									
0x2B	Reserved									
0x2C	Reserved									
0x2D	Reserved									
0x2E	Reserved									
0x2F	Reserved									
0x30	CCBUF0	7:0	CCBUF[7:0]							
0x31		15:8	CCBUF[15:8]							
0x32		23:16	CCBUF[23:16]							
0x33		31:24	CCBUF[31:24]							
0x34	CCBUF1	7:0	CCBUF[7:0]							
0x35		15:8	CCBUF[15:8]							
0x36		23:16	CCBUF[23:16]							
0x37		31:24	CCBUF[31:24]							
0x38	Reserved									
0x39	Reserved									
0x3A	Reserved									
0x3B	Reserved									
0x3C	Reserved									
0x3D	Reserved									

Offset	Name	Bit Pos.								
0x3E	Reserved									
0x3F	Reserved									

35.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection.

Some registers are synchronized when read and/or written. Synchronization is denoted by the "Write-Synchronized" or the "Read-Synchronized" property in each individual register description. For details, refer to Synchronization.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

35.8.1. Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			COPEN1	COPEN0			CAPTEN1	CAPTEN0
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit	15	14	13	12	11	10	9	8
					ALOCK	PRESCALER[2:0]		
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY	PRESCSYNC[1:0]		MODE[1:0]		ENABLE	SWRST
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 11 – ALOCK: Auto Lock
When this bit is set, Lock bit update (LUPD) is set to '1' on each overflow/underflow or re-trigger event.

This bit is not synchronized.

Value	Description
0	The LUPD bit is not affected on overflow/underflow, and re-trigger event.
1	The LUPD bit is set on each overflow/underflow or re-trigger event.

Bits 10:8 – PRESCALER[2:0]: Prescaler
These bits select the counter prescaler factor.

These bits are not synchronized.

Value	Name	Description
0x0	DIV1	Prescaler: GCLK_TC
0x1	DIV2	Prescaler: GCLK_TC/2
0x2	DIV4	Prescaler: GCLK_TC/4
0x3	DIV8	Prescaler: GCLK_TC/8
0x4	DIV16	Prescaler: GCLK_TC/16

Value	Name	Description
0x5	DIV64	Prescaler: GCLK_TC/64
0x6	DIV256	Prescaler: GCLK_TC/256
0x7	DIV1024	Prescaler: GCLK_TC/1024

Bit 7 – ONDEMAND: Clock On Demand
This bit selects the clock requirements when the TC is stopped.

In standby mode, if the Run in Standby bit (CTRLA.RUNSTDBY) is '0', ONDEMAND is forced to '0'.

This bit is not synchronized.

Value	Description
0	The On Demand is disabled. If On Demand is disabled, the TC will continue to request the clock when its operation is stopped (STATUS.STOP=1).
1	The On Demand is enabled. When On Demand is enabled, the stopped TC will not request the clock. The clock is requested when a software re-trigger command is applied or when an event with start/re-trigger action is detected.

Bit 6 – RUNSTDBY: Run in Standby
This bit is used to keep the TC running in standby mode.

This bit is not synchronized.

Value	Description
0	The TC is halted in standby.
1	The TC continues to run in standby.

Bits 5:4 – PRESCSYNC[1:0]: Prescaler and Counter Synchronization
These bits select whether the counter should wrap around on the next GCLK_TCx clock or the next prescaled GCLK_TCx clock. It also makes it possible to reset the prescaler.

These bits are not synchronized.

Value	Name	Description
0x0	GCLK	Reload or reset the counter on next generic clock
0x1	PRESC	Reload or reset the counter on next prescaler clock
0x2	RESYNC	Reload or reset the counter on next generic clock. Reset the prescaler counter
0x3	-	Reserved

Bits 3:2 – MODE[1:0]: Timer Counter Mode
These bits select the counter mode.

These bits are not synchronized.

Value	Name	Description
0x0	COUNT16	Counter in 16-bit mode
0x1	COUNT8	Counter in 8-bit mode

Value	Name	Description
0x2	COUNT32	Counter in 32-bit mode
0x3	-	Reserved

Bit 1 – ENABLE: Enable
Due to synchronization, there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately, and the ENABLE Synchronization Busy bit in the SYNCBUSY register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

Value	Description
0	The peripheral is disabled.
1	The peripheral is enabled.

Bit 0 – SWRST: Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the TC, except DBGCTRL, to their initial state, and the TC will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence; all other writes in the same write-operation will be discarded.

Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

Bits 21,20 – COPENx: Capture On Pin x Enable [x = 1..0]
This bit selects the trigger source for capture operation, either events or I/O pin input.

Value	Description
0	Event from Event System is selected as trigger source for capture operation on channel x.
1	I/O pin is selected as trigger source for capture operation on channel x.

Bits 17,16 – CAPTENx: Capture Channel x Enable [x = 1..0]
These bits are used to select whether channel x is a capture or a compare channel.

These bits are not synchronized.

Value	Description
0	CAPTENx disables capture on channel x.
1	CAPTENx enables capture on channel x.

35.8.2. Control B Clear

This register allows the user to clear bits in the CTRLB register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Set register (CTRLBSET).

Name: CTRLBCLR
Offset: 0x04
Reset: 0x00
Property: PAC Write-Protection, Read-Synchronized, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	CMD[2:0]					ONESHOT	LUPD	DIR
Access	R/W	R/W	R/W			R/W	R/W	R/W
Reset	0	0	0			0	0	0

Bits 7:5 – CMD[2:0]: Command
These bits are used for software control of the TC. The commands are executed on the next prescaled GCLK_TC clock cycle. When a command has been executed, the CMD bit group will be read back as zero.

Writing 0x0 to these bits has no effect.

Writing a '1' to any of these bits will clear the pending command.

Bit 2 – ONESHOT: One-Shot on Counter
This bit controls one-shot operation of the TC.

Writing a '0' to this bit has no effect

Writing a '1' to this bit will disable one-shot operation.

Value	Description
0	The TC will wrap around and continue counting on an overflow/underflow condition.
1	The TC will wrap around and stop on the next underflow/overflow condition.

Bit 1 – LUPD: Lock Update
This bit controls the update operation of the TC buffered registers.

When CTRLB.LUPD is set, no update of the buffered registers is performed, even though an UPDATE condition has occurred. Locking the update ensures that all buffer registers are valid before an update is performed.

This bit has no effect when input capture operation is enabled.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the LUPD bit.

Value	Description
0	The CCBUFx and PERBUF buffer registers value are copied into CCx and PER registers on counter update condition.
1	The CCBUFx and PERBUF buffer registers value are not copied into CCx and PER registers.

Bit 0 – DIR: Counter Direction
This bit is used to change the direction of the counter.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the bit and make the counter count up.

Value	Description
0	The timer/counter is counting up (incrementing).
1	The timer/counter is counting down (decrementing).

35.8.3. Control B Set

This register allows the user to set bits in the CTRLB register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Clear register (CTRLBCLR).

Name: CTRLBSET
Offset: 0x05
Reset: 0x00
Property: PAC Write-Protection, Read-synchronized, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	CMD[2:0]					ONESHOT	LUPD	DIR
Access	R/W	R/W	R/W			R/W	R/W	R/W
Reset	0	0	0			0	0	0

Bits 7:5 – CMD[2:0]: Command
These bits are used for software control of the TC. The commands are executed on the next prescaled GCLK_TC clock cycle. When a command has been executed, the CMD bit group will be read back as zero.

Writing 0x0 to these bits has no effect.

Writing a value different from 0x0 to these bits will issue a command for execution.

Value	Name	Description
0x0	NONE	No action
0x1	RETRIGGER	Force a start, restart or retrigger
0x2	STOP	Force a stop
0x3	UPDATE	Force update of double buffered registers
0x4	READSYNC	Force a read synchronization of COUNT

Bit 2 – ONESHOT: One-Shot on Counter
This bit controls one-shot operation of the TC.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will enable one-shot operation.

Value	Description
0	The TC will wrap around and continue counting on an overflow/underflow condition.
1	The TC will wrap around and stop on the next underflow/overflow condition.

Bit 1 – LUPD: Lock Update
This bit controls the update operation of the TC buffered registers.

When CTRLB.LUPD is set, no update of the buffered registers is performed, even though an UPDATE condition has occurred. Locking the update ensures that all buffer registers are valid before an update is performed.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the LUPD bit.

This bit has no effect when input capture operation is enabled.

Value	Description
0	The CCBUFx and PERBUF buffer registers value are copied into CCx and PER registers on counter update condition.
1	The CCBUFx and PERBUF buffer registers value are not copied into CCx and PER registers.

Bit 0 – DIR: Counter Direction
This bit is used to change the direction of the counter.

Writing a '0' to this bit has no effect

Writing a '1' to this bit will clear the bit and make the counter count up.

Value	Description
0	The timer/counter is counting up (incrementing).
1	The timer/counter is counting down (decrementing).

35.8.4. Event Control

Name: EVCTRL
Offset: 0x06
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
			MCEO1	MCEO0				OVFEO
Access			R/W	R/W				R/W
Reset			0	0				0

Bit	7	6	5	4	3	2	1	0
			TCEI	TCINV		EVACT[2:0]		
Access			R/W	R/W		R/W	R/W	R/W
Reset			0	0		0	0	0

Bit 8 – OVFEO: Overflow/Underflow Event Output Enable
This bit enables the Overflow/Underflow event. When enabled, an event will be generated when the counter overflows/underflows.

Value	Description
0	Overflow/Underflow event is disabled and will not be generated.
1	Overflow/Underflow event is enabled and will be generated for every counter overflow/ underflow.

Bit 5 – TCEI: TC Event Enable
This bit is used to enable asynchronous input events to the TC.

Value	Description
0	Incoming events are disabled.
1	Incoming events are enabled.

Bit 4 – TCINV: TC Inverted Event Input Polarity
This bit inverts the asynchronous input event source.

Value	Description
0	Input event source is not inverted.
1	Input event source is inverted.

Bits 2:0 – EVACT[2:0]: Event Action
These bits define the event action the TC will perform on an event.

Value	Name	Description
0x0	OFF	Event action disabled
0x1	RETRIGGER	Start, restart or retrigger TC on event
0x2	COUNT	Count on event

Value	Name	Description
0x3	START	Start TC on event
0x4	STAMP	Time stamp capture
0x5	PPW	Period captured in CC0, pulse width in CC1
0x6	PWP	Period captured in CC1, pulse width in CC0
0x7	PW	Pulse width capture

Bits 13,12 – MCEOx: Match or Capture Channel x Event Output Enable [x = 1..0]
These bits enable the generation of an event for every match or capture on channel x.

Value	Description
0	Match/Capture event on channel x is disabled and will not be generated.
1	Match/Capture event on channel x is enabled and will be generated for every compare/capture.

35.8.5. Interrupt Enable Clear

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Name: INTENCLR
Offset: 0x08
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
			MC1	MC0			ERR	OVF
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit 1 – ERR: Error Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Error Interrupt Enable bit, which disables the Error interrupt.

Value	Description
0	The Error interrupt is disabled.
1	The Error interrupt is enabled.

Bit 0 – OVF: Overflow Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Overflow Interrupt Enable bit, which disables the Overflow interrupt request.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Bits 5,4 – MCx: Match or Capture Channel x Interrupt Enable [x = 1..0]
Writing a '0' to these bits has no effect.

Writing a '1' to MCx will clear the corresponding Match or Capture Channel x Interrupt Enable bit, which disables the Match or Capture Channel x interrupt.

Value	Description
0	The Match or Capture Channel x interrupt is disabled.
1	The Match or Capture Channel x interrupt is enabled.

35.8.6. Interrupt Enable Set

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Name: INTENSET
Offset: 0x09
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
			MC1	MC0			ERR	OVF
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit 1 – ERR: Error Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Error Interrupt Enable bit, which enables the Error interrupt.

Value	Description
0	The Error interrupt is disabled.
1	The Error interrupt is enabled.

Bit 0 – OVF: Overflow Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Overflow Interrupt Enable bit, which enables the Overflow interrupt request.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Bits 5,4 – MCx: Match or Capture Channel x Interrupt Enable [x = 1..0]
Writing a '0' to these bits has no effect.

Writing a '1' to MCx will set the corresponding Match or Capture Channel x Interrupt Enable bit, which enables the Match or Capture Channel x interrupt.

Value	Description
0	The Match or Capture Channel x interrupt is disabled.
1	The Match or Capture Channel x interrupt is enabled.

35.8.7. Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x0A
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
			MC1	MC0			ERR	OVF
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit 1 – ERR: Error Interrupt Flag
This flag is set when a new capture occurs on a channel while the corresponding Match or Capture Channel x interrupt flag is set, in which case there is nowhere to store the new capture.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Error interrupt flag.

Bit 0 – OVF: Overflow Interrupt Flag
This flag is set on the next CLK_TC_CNT cycle after an overflow condition occurs, and will generate an interrupt request if INTENCLR.OVF or INTENSET.OVF is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Overflow interrupt flag.

Bits 5,4 – MCx: Match or Capture Channel x [x = 1..0]
This flag is set on a comparison match, or when the corresponding CCx register contains a valid capture value. This flag is set on the next CLK_TC_CNT cycle, and will generate an interrupt request if the corresponding Match or Capture Channel x Interrupt Enable bit in the Interrupt Enable Set register (INTENSET.MCx) is '1'.

Writing a '0' to one of these bits has no effect.

Writing a '1' to one of these bits will clear the corresponding Match or Capture Channel x interrupt flag

In capture operation, this flag is automatically cleared when CCx register is read.

35.8.8. Status

Name: STATUS
Offset: 0x0B
Reset: 0x01
Property: Write-Synchronized

Bit	7	6	5	4	3	2	1	0
			CCBUFV1	CCBUFV0	PERBUFV		SLAVE	STOP
Access			R/W	R/W	R/W		R	R
Reset			0	0	0		0	1

Bit 3 – PERBUFV: Period Buffer Valid
This bit is set when a new value is written to the PERBUF register. The bit is cleared by writing '1' to the corresponding location, or automatically cleared by hardware on UPDATE condition. This bit is available only in 8-bit mode and will always read zero in 16- and 32-bit modes.

Bit 1 – SLAVE: Slave Status Flag
This bit is only available in 32-bit mode on the slave TC (i.e., TC1 and/or TC3). The bit is set when the associated master TC (TC0 and TC2, respectively) is set to run in 32-bit mode.

Bit 0 – STOP: Stop Status Flag
This bit is set when the TC is disabled, on a Stop command, or on an overflow/underflow condition when the One-Shot bit in the Control B Set register (CTRLBSET.ONESHOT) is '1'.

Value	Description
0	Counter is running.
1	Counter is stopped.

Bits 5,4 – CCBUFVx: Channel x Compare or Capture Buffer Valid [x = 1..0]
For a compare channel x, the bit x is set when a new value is written to the corresponding CCBUFx register.

The bit x is cleared by writing a '1' to it, or it is cleared automatically by hardware on UPDATE condition.

For a capture channel x, the bit x is set when a valid capture value is stored in the CCBUFx register. The bit x is cleared automatically when the CCx register is read.

35.8.9. Waveform Generation Control

Name: WAVE
Offset: 0x0C
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
							WAVEGEN[1:0]	
Access							R/W	R/W
Reset							0	0

Bits 1:0 – WAVEGEN[1:0]: Waveform Generation Mode
These bits select the waveform generation operation. They affect the top value, as shown in Waveform Output Operations on page 711. They also control whether frequency or PWM waveform generation should be used. The waveform generation operations are explained in Waveform Output Operations on page 711.

These bits are not synchronized.

Value	Name	Operation	Top Value	Output Waveform on Match	Output Waveform on Wraparound
0x0	NFRQ	Normal frequency	PER[1] / Max	Toggle	No action
0x1	MFRQ	Match frequency	CC0	Toggle	No action
0x2	NPWM	Normal PWM	PER[1] / Max	Set	Clear
0x3	MPWM	Match PWM	CC0	Set	Clear

1) This depends on the TC mode: In 8-bit mode, the top value is the Period Value register (PER). In 16- and 32-bit mode it is the respective MAX value.

35.8.10. Driver Control

Name: DRVCTRL
Offset: 0x0D
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

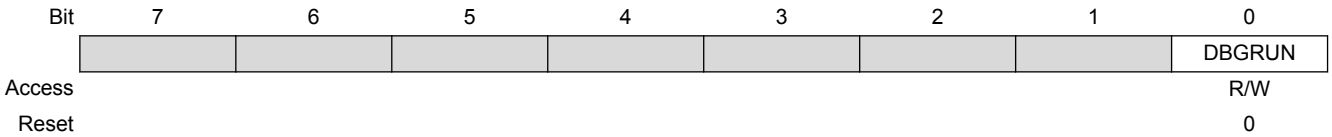
Bit	7	6	5	4	3	2	1	0
							INVEN1	INVEN0
Access							R/W	R/W
Reset							0	0

Bits 1,0 – INVENx: Output Waveform x Invert Enable [x = 1..0]
These bits are used to select inversion of the output or capture trigger input of channel x.

Value	Description
0	Disable inversion of the WO[x] output and IO input pin.
1	Enable inversion of the WO[x] output and IO input pin.

35.8.11. Debug Control

Name: DBGCTRL
Offset: 0x0F
Reset: 0x00
Property: PAC Write-Protection



Bit 0 – DBGRUN: Debug Run Mode
This bit is not affected by a software reset, and should not be changed by software while the TC is enabled.

Value	Description
0	The TC is halted when the device is halted in debug mode.
1	The TC continues normal operation when the device is halted in debug mode.

35.8.12. Synchronization Busy

Name: SYNCBUSY
Offset: 0x10
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	CC1	CC0	PER	COUNT	STATUS	CTRLB	ENABLE	SWRST
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 5 – PER: PER Synchronization Busy
This bit is cleared when the synchronization of PER between the clock domains is complete.

This bit is set when the synchronization of PER between clock domains is started.

This bit is also set when the PER is written, and cleared on update condition. The bit is automatically cleared when the STATUS.PERBUF bit is cleared.

Bit 4 – COUNT: COUNT Synchronization Busy
This bit is cleared when the synchronization of COUNT between the clock domains is complete.

This bit is set when the synchronization of COUNT between clock domains is started.

Bit 3 – STATUS: STATUS Synchronization Busy
This bit is cleared when the synchronization of STATUS between the clock domains is complete.

This bit is set when a '1' is written to the Capture Channel Buffer Valid status flags (STATUS.CCBUFVx) and the synchronization of STATUS between clock domains is started.

Bit 2 – CTRLB: CTRLB Synchronization Busy
This bit is cleared when the synchronization of CTRLB between the clock domains is complete.

This bit is set when the synchronization of CTRLB between clock domains is started.

Bit 1 – ENABLE: ENABLE Synchronization Busy
This bit is cleared when the synchronization of ENABLE bit between the clock domains is complete.

This bit is set when the synchronization of ENABLE bit between clock domains is started.

Bit 0 – SWRST: SWRST Synchronization Busy
This bit is cleared when the synchronization of SWRST bit between the clock domains is complete.

This bit is set when the synchronization of SWRST bit between clock domains is started.

Bits 7,6 – CCx: Compare/Capture Channel x Synchronization Busy
For details on CC channels number, refer to each TC feature list.

This bit is set when the synchronization of CCx between clock domains is started.

This bit is also set when the CCBUFx is written, and cleared on update condition. The bit is automatically cleared when the STATUS.CCBUFx bit is cleared.

35.8.13. Counter Value, 8-bit Mode

Name: COUNT
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Bit	7	6	5	4	3	2	1	0
	COUNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – COUNT[7:0]: Counter Value
These bits contain the current counter value.

35.8.14. Counter Value, 16-bit Mode

Name: COUNT
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Bit	15	14	13	12	11	10	9	8
	COUNT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COUNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – COUNT[15:0]: Counter Value
These bits contain the current counter value.

35.8.15. Counter Value, 32-bit Mode

Name: COUNT
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Bit	31	30	29	28	27	26	25	24
	COUNT[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	COUNT[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	COUNT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COUNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – COUNT[31:0]: Counter Value
These bits contain the current counter value.

35.8.16. Period Value, 8-bit Mode

Name: PER
Offset: 0x1B
Reset: 0xFF
Property: Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	PER[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	1

Bits 7:0 – PER[7:0]: Period Value
These bits hold the value of the Period Buffer register PERBUF. The value is copied to PER register on UPDATE condition.

35.8.17. Channel x Compare/Capture Value, 8-bit Mode

Name: CCx
Offset: 0x1C+i*0x1 [i=0..1]
Reset: 0x00
Property: Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	CC[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – CC[7:0]: Channel x Compare/Capture Value
These bits contain the compare/capture value in 8-bit TC mode. In Match frequency (MFRQ) or Match PWM (MPWM) waveform operation (WAVE.WAVEGEN), the CC0 register is used as a period register.

35.8.18. Channel x Compare/Capture Value, 16-bit Mode

Name: CCx
Offset: 0x1C+i*0x2 [i=0..1]
Reset: 0x0000
Property: Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	CC[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CC[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – CC[15:0]: Channel x Compare/Capture Value
These bits contain the compare/capture value in 16-bit TC mode. In Match frequency (MFRQ) or Match PWM (MPWM) waveform operation (WAVE.WAVEGEN), the CC0 register is used as a period register.

35.8.19. Channel x Compare/Capture Value, 32-bit Mode

Name: CCx
Offset: 0x1C+i*0x4 [i=0..1]
Reset: 0x00000000
Property: Write-Synchronized

Bit	31	30	29	28	27	26	25	24
	CC[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CC[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CC[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CC[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CC[31:0]: Channel x Compare/Capture Value
These bits contain the compare/capture value in 32-bit TC mode. In Match frequency (MFRQ) or Match PWM (MPWM) waveform operation (WAVE.WAVEGEN), the CC0 register is used as a period register.

35.8.20. Period Buffer Value, 8-bit Mode

Name: PERBUF
Offset: 0x2F
Reset: 0xFF
Property: -

Bit	7	6	5	4	3	2	1	0
	PERBUF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	1

Bits 7:0 – PERBUF[7:0]: Period Buffer Value
These bits hold the value of the period buffer register. The value is copied to PER register on UPDATE condition.

35.8.21. Channel x Compare Buffer Value, 8-bit Mode

Name: CCBUFx
Offset: 0x30+i*0x1 [i=0..1]
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	CCBUF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – CCBUF[7:0]: Channel x Compare Buffer Value
These bits hold the value of the Channel x Compare Buffer Value. When the buffer valid flag is '1' and double buffering is enabled (CTRLBCLR.LUPD=1), the data from buffer registers will be copied into the corresponding CCx register under UPDATE condition (CTRLBSET.CMD=0x3), including the software update command.

35.8.22. Channel x Compare Buffer Value, 16-bit Mode

Name: CCBUFx
Offset: 0x30+i*0x2 [i=0..1]
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
	CCBUF[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CCBUF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – CCBUF[15:0]: Channel x Compare Buffer Value
These bits hold the value of the Channel x Compare Buffer Value. When the buffer valid flag is '1' and double buffering is enabled (CTRLBCLR.LUPD=1), the data from buffer registers will be copied into the corresponding CCx register under UPDATE condition (CTRLBSET.CMD=0x3), including the software update command.

35.8.23. Channel x Compare Buffer Value, 32-bit Mode

Name: CCBUFx
Offset: 0x30+i*0x4 [i=0..1]
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	CCBUF[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CCBUF[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CCBUF[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CCBUF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CCBUF[31:0]: Channel x Compare Buffer Value
These bits hold the value of the Channel x Compare Buffer Value. When the buffer valid flag is '1' and double buffering is enabled (CTRLBCLR.LUPD=1), the data from buffer registers will be copied into the corresponding CCx register under UPDATE condition (CTRLBSET.CMD=0x3), including the software update command.

36. TCC – Timer/Counter for Control Applications

36.1. Overview

The device provides three instances of the Timer/Counter for Control applications (TCC) peripheral , TCC[2:0].

Each TCC instance consists of a counter, a prescaler, compare/capture channels and control logic. The counter can be set to count events or clock pulses. The counter together with the compare/capture channels can be configured to time stamp input events, allowing capture of frequency and pulse-width. It can also perform waveform generation such as frequency generation and pulse-width modulation.

Waveform extensions are intended for motor control, ballast, LED, H-bridge, power converters, and other types of power control applications. They allow for low- and high-side output with optional dead-time insertion. Waveform extensions can also generate a synchronized bit pattern across the waveform output pins. The fault options enable fault protection for safe and deterministic handling, disabling and/or shut down of external drivers.

Figure 36-1 Timer/Counter for Control Applications - Block Diagram on page 756 shows all features in TCC, whereas the table below lists the actual configuration of each of the TCC[2:0].

Table 36-1. TCC Configuration Summary

TCC#	Channels (CC_NUM)	Waveform Output (WO_NUM)	Counter size	Fault	Dithering	Output matrix	Dead Time Insertion (DTI)	SWAP	Pattern generation
0	4	8	24-bit	Yes	Yes	Yes	Yes	Yes	Yes
1	2	4	24-bit	Yes	Yes				Yes
2	2	2	16-bit	Yes					

Note: The number of CC registers (CC_NUM) for each TCC corresponds to the number of compare/capture channels, so that a TCC can have more Waveform Outputs (WO_NUM) than CC registers.

36.2. Features

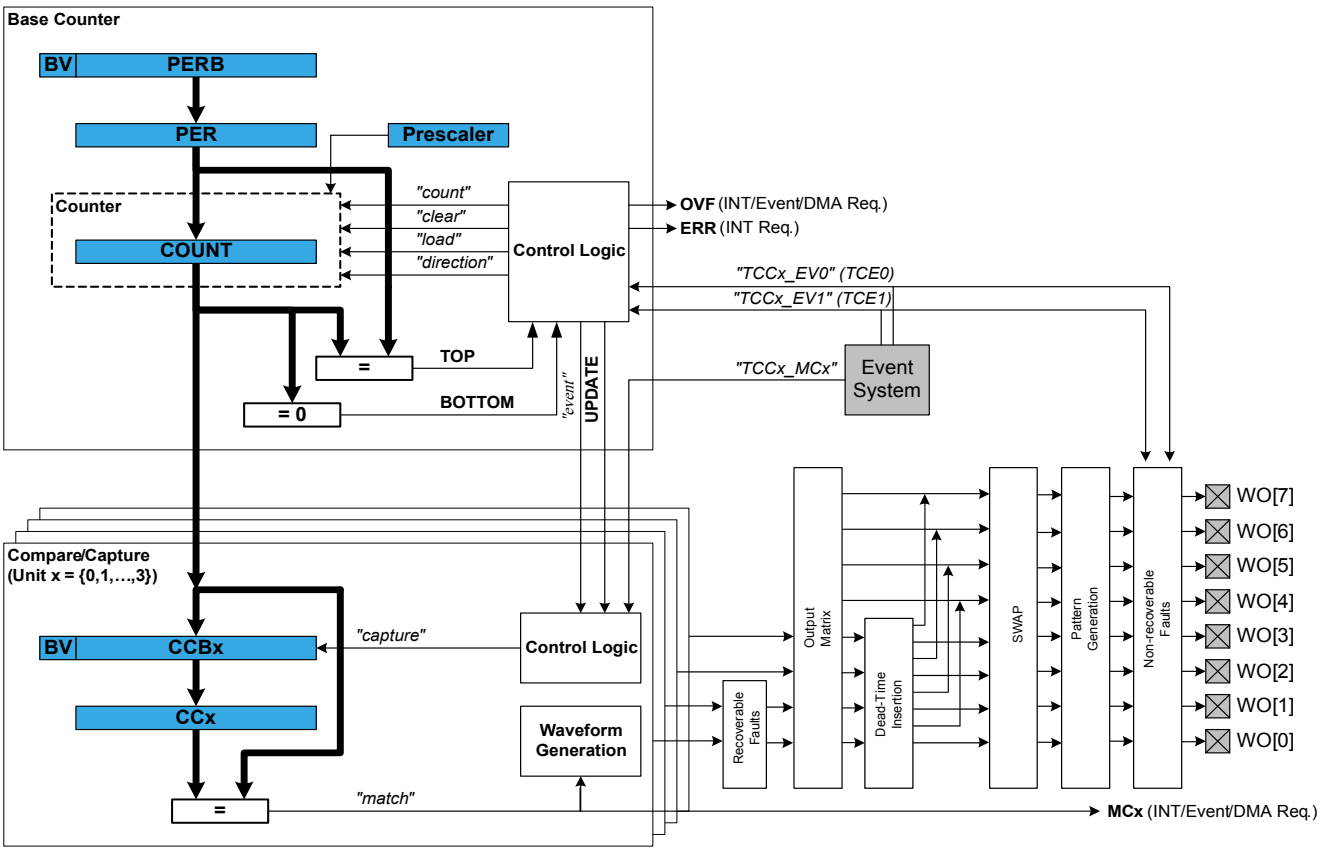
- Up to four compare/capture channels (CC) with:
  - Double buffered period setting
  - Double buffered compare or capture channel
  - Circular buffer on period and compare channel registers
- Waveform generation:
  - Frequency generation
  - Single-slope pulse-width modulation (PWM)
  - Dual-slope pulse-width modulation with half-cycle reload capability
- Input capture:
  - Event capture
  - Frequency capture
  - Pulse-width capture
- Waveform extensions:
  - Configurable distribution of compare channels outputs across port pins

- Low- and high-side output with programmable dead-time insertion
- Waveform swap option with double buffer support
- Pattern generation with double buffer support
- Dithering support

- Fault protection for safe disabling of drivers:
  - Two recoverable fault sources
  - Two non-recoverable fault sources
  - Debugger can be source of non-recoverable fault
- Input events:
  - Two input events for counter
  - One input event for each channel
- Output events:
  - Three output events (Count, Re-Trigger and Overflow) available for counter
  - One Compare Match/Input Capture event output for each channel
- Interrupts:
  - Overflow and Re-Trigger interrupt
  - Compare Match/Input Capture interrupt
  - Interrupt on fault detection
- Can be used with DMA and can trigger DMA transactions

36.3. Block Diagram

Figure 36-1. Timer/Counter for Control Applications - Block Diagram



36.4. Signal Description

Pin Name	Type	Description
TCCx/WO[0]	Digital output	Compare channel 0 waveform output
TCCx/WO[1]	Digital output	Compare channel 1 waveform output
…	...	...
TCCx/WO[WO_NUM-1]	Digital output	Compare channel n waveform output

Refer to *I/O Multiplexing and Considerations* for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

Related Links

I/O Multiplexing and Considerations on page 29

36.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

36.5.1. I/O Lines

In order to use the I/O lines of this peripheral, the I/O pins must be configured using the I/O Pin Controller (PORT).

Related Links

PORT: IO Pin Controller on page 506

36.5.2. Power Management

This peripheral can continue to operate in any sleep mode where its source clock is running. The interrupts can wake up the device from sleep modes. Events connected to the event system can trigger other operations in the system without exiting sleep modes. Refer to *PM – Power Manager* for details on the different sleep modes.

36.5.3. Clocks

The TCC bus clock (CLK_TCCx_APB, with x instance number of the TCCx) is enabled by default, and can be enabled and disabled in the Main Clock.

A generic clock (GCLK_TCCx) is required to clock the TCC. This clock must be configured and enabled in the generic clock controller before using the TCC. Note that TCC0 and TCC1 share a peripheral clock generator.

The generic clocks (GCLK_TCCx) are asynchronous to the bus clock (CLK_TCCx_APB). Due to this asynchronicity, writing certain registers will require synchronization between the clock domains. Refer to Synchronization on page 789 for further details.

Related Links

Peripheral Clock Masking on page 151

GCLK - Generic Clock Controller on page 131

36.5.4. DMA

The DMA request lines are connected to the DMA Controller (DMAC). In order to use DMA requests with this peripheral the DMAC must be configured first. Refer to *DMAC – Direct Memory Access Controller* for details.

Related Links

DMAC – Direct Memory Access Controller on page 399

36.5.5. Interrupts

The interrupt request line is connected to the Interrupt Controller. In order to use interrupt requests of this peripheral, the Interrupt Controller (NVIC) must be configured first. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

Nested Vector Interrupt Controller on page 51

36.5.6. Events

The events of this peripheral are connected to the Event System.

Related Links

EVSYS – Event System on page 536

36.5.7. Debug Operation

When the CPU is halted in debug mode, this peripheral will halt normal operation. This peripheral can be forced to continue operation during debugging - refer to the Debug Control (DBGCTRL) register for details.

Refer to DBGCTRL on page 810 register for details.

36.5.8. Register Access Protection

Registers with write-access can be optionally write-protected by the Peripheral Access Controller (PAC), except the following:

- Interrupt Flag register (INTFLAG)
- Status register (STATUS)
- Period and Period Buffer registers (PER, PERB)
- Compare/Capture and Compare/Capture Buffer registers (CCx, CCBx)
- Control Waveform register (WAVE)
- Pattern Generation Value and Pattern Generation Value Buffer registers (PATT, PATTB)

Note: Optional write-protection is indicated by the "PAC Write-Protection" property in the register description.

When the CPU is halted in debug mode, all write-protection is automatically disabled. Write-protection does not apply for accesses through an external debugger.

36.5.9. Analog Connections

Not applicable.

36.6. Functional Description

36.6.1. Principle of Operation

The following definitions are used throughout the documentation:

Table 36-2. Timer/Counter for Control Applications - Definitions

Name	Description
TOP	The counter reaches TOP when it becomes equal to the highest value in the count sequence. The TOP value can be the same as Period (PER) or the Compare Channel 0 (CC0) register value depending on the waveform generator mode in Waveform Output Generation Operations on page 764.
ZERO	The counter reaches ZERO when it contains all zeroes.
MAX	The counter reaches maximum when it contains all ones.
UPDATE	The timer/counter signals an update when it reaches ZERO or TOP, depending on the direction settings.
Timer	The timer/counter clock control is handled by an internal source.
Counter	The clock control is handled externally (e.g. counting external events).
CC	For compare operations, the CC are referred to as "compare channels." For capture operations, the CC are referred to as "capture channels."

Each TCC instance has up to four compare/capture channels (CCx).

The counter register (COUNT), period registers with buffer (PER and PERB), and compare and capture registers with buffers (CCx and CCBx) are 16- or 24-bit registers, depending on each TCC instance. Each buffer register has a buffer valid (BUFV) flag that indicates when the buffer contains a new value.

Under normal operation, the counter value is continuously compared to the TOP or ZERO value to determine whether the counter has reached TOP or ZERO. In either case, the TCC can generate interrupt requests, request DMA transactions, or generate events for the Event System. In waveform generator mode, these comparisons are used to set the waveform period or pulse width.

A prescaled generic clock (GCLK_TCCx) and events from the event system can be used to control the counter. The event system is also used as a source to the input capture.

The Recoverable Fault Unit enables event controlled waveforms by acting directly on the generated waveforms of the TCC compare channels output. These events can restart, halt the timer/counter period, shorten the output pulse active time, or disable waveform output as long as the fault condition is present. This can typically be used for current sensing regulation, and zero-crossing and demagnetization re-triggering.

The MCE0 and MCE1 event sources are shared with the Recoverable Fault Unit. Only asynchronous events are used internally when fault unit extension is enabled. For further details on how to configure asynchronous events routing, refer to *EVSYS – Event System*.

Recoverable fault sources can be filtered and/or windowed to avoid false triggering, for example from I/O pin glitches, by using digital filtering, input blanking, and qualification options. See also Recoverable Faults on page 776.

In addition, six optional independent and successive units primarily intended for use with different types of motor control, ballast, LED, H-bridge, power converter, and other types of power switching applications, are implemented in some of TCC instances. See also Figure 36-1 Timer/Counter for Control Applications - Block Diagram on page 756.

The output matrix (OTMX) can distribute and route out the TCC waveform outputs across the port pins in different configurations, each optimized for different application types. The Dead-Time Insertion (DTI) unit splits the four lower OTMX outputs into two non-overlapping signals: the non-inverted low side (LS) and inverted high side (HS) of the waveform output with optional dead-time insertion between LS and HS switching. The SWAP unit can swap the LS and HS pin outputs, and can be used for fast decay motor control.

The pattern generation unit can be used to generate synchronized waveforms with constant logic level on TCC UPDATE conditions. This is useful for easy stepper motor and full bridge control.

The non-recoverable fault module enables event controlled fault protection by acting directly on the generated waveforms of the timer/counter compare channel outputs. When a non-recoverable fault condition is detected, the output waveforms are forced to a safe and pre-configured value that is safe for the application. This is typically used for instant and predictable shut down and disabling high current or voltage drives.

The count event sources (TCE0 and TCE1) are shared with the non-recoverable fault extension. The events can be optionally filtered. If the filter options are not used, the non-recoverable faults provide an immediate asynchronous action on waveform output, even for cases where the clock is not present. For further details on how to configure asynchronous events routing, refer to section *EVSYS – Event System*.

Related Links

EVSYS – Event System on page 536

36.6.2. Basic Operation

36.6.2.1. Initialization

The following registers are enable-protected, meaning that they can only be written when the TCC is disabled(CTRLA.ENABLE=0):

- Control A (CTRLA) register, except Run Standby (RUNSTDBY), Enable (ENABLE) and Software Reset (SWRST) bits
- Recoverable Fault n Control registers (FCTRLA and FCTRLB)
- Waveform Extension Control register (WEXCTRL)
- Drive Control register (DRVCTRL)
- Event Control register (EVCTRL)

Enable-protected bits in the CTRLA register can be written at the same time as CTRLA.ENABLE is written to '1', but not at the same time as CTRLA.ENABLE is written to '0'. Enable-protection is denoted by the “Enable-Protected” property in the register description.

Before the TCC is enabled, it must be configured as outlined by the following steps:

1. Enable the TCC bus clock (CLK_TCCx_APB).
2. If Capture mode is required, enable the channel in capture mode by writing a '1' to the Capture Enable bit in the Control A register (CTRLA.CPTEN).

Optionally, the following configurations can be set before enabling TCC:

1. Select PRESCALER setting in the Control A register (CTRLA.PRESCALER).
2. Select Prescaler Synchronization setting in Control A register (CTRLA.PRESCSYNC).
3. If down-counting operation is desired, write the Counter Direction bit in the Control B Set register (CTRLBSET.DIR) to '1'.
4. Select the Waveform Generation operation in the WAVE register (WAVE.WAVEGEN).
5. Select the Waveform Output Polarity in the WAVE register (WAVE.POL).

6. The waveform output can be inverted for the individual channels using the Waveform Output Invert Enable bit group in the Driver register (DRVCTRL.INVEN).

36.6.2.2. Enabling, Disabling, and Resetting

The TCC is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The TCC is disabled by writing a zero to CTRLA.ENABLE.

The TCC is reset by writing '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the TCC, except DBGCTRL, will be reset to their initial state, and the TCC will be disabled. Refer to Control A (CTRLA on page 794) register for details.

The TCC should be disabled before the TCC is reset to avoid undefined behavior.

36.6.2.3. Prescaler Selection

The GCLK_TCCx clock is fed into the internal prescaler.

The prescaler consists of a counter that counts up to the selected prescaler value, whereupon the output of the prescaler toggles.

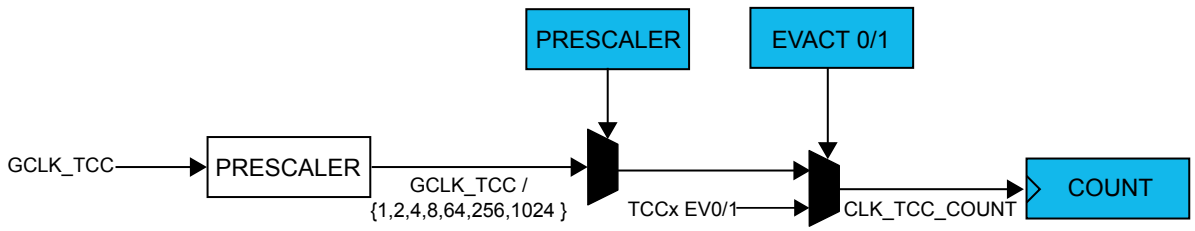
If the prescaler value is higher than one, the counter update condition can be optionally executed on the next GCLK_TCC clock pulse or the next prescaled clock pulse. For further details, refer to the Prescaler (CTRLA.PRESCALER) and Counter Synchronization (CTRLA.PRESYNC) descriptions.

Prescaler outputs from 1 to 1/1024 are available. For a complete list of available prescaler outputs, see the register description for the Prescaler bit group in the Control A register (CTRLA.PRESCALER).

Note: When counting events, the prescaler is bypassed.

The joint stream of prescaler ticks and event action ticks is called CLK_TCC_COUNT.

Figure 36-2. Prescaler



36.6.2.4. Counter Operation

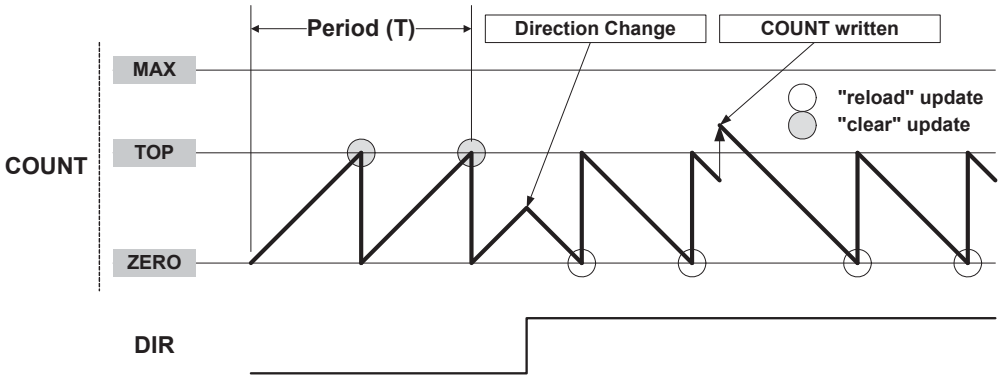
Depending on the mode of operation, the counter is cleared, reloaded, incremented, or decremented at each TCC clock input (CLK_TCC_COUNT). A counter clear or reload mark the end of current counter cycle and the start of a new one.

The counting direction is set by the Direction bit in the Control B register (CTRLB.DIR). If the bit is zero, it's counting up and one if counting down.

The counter will count up or down for each tick (clock or event) until it reaches TOP or ZERO. When it's counting up and TOP is reached, the counter will be set to zero at the next tick (overflow) and the Overflow Interrupt Flag in the Interrupt Flag Status and Clear register (INTFLAG.OVF) will be set. When down-counting, the counter is reloaded with the TOP value when ZERO is reached (underflow), and INTFLAG.OVF is set.

INTFLAG.OVF can be used to trigger an interrupt, a DMA request, or an event. An overflow/underflow occurrence (i.e. a compare match with TOP/ZERO) will stop counting if the One-Shot bit in the Control B register is set (CTRLBSET.ONESHOT).

Figure 36-3. Counter Operation



It is possible to change the counter value (by writing directly in the COUNT register) even when the counter is running. The COUNT value will always be ZERO or TOP, depending on direction set by CTRLBSET.DIR or CTRLBCLR.DIR, when starting the TCC, unless a different value has been written to it, or the TCC has been stopped at a value other than ZERO. The write access has higher priority than count, clear, or reload. The direction of the counter can also be changed during normal operation. See also Figure 36-3 Counter Operation on page 762.

Stop Command and Event Action

A stop command can be issued from software by using TCC Command bits in Control B Set register (CTRLBSET.CMD=0x2, STOP) or when the stop event action is configured in the Input Event Action 1 bits in Event Control register (EVCTRL.EVACT1=0x3, STOP).

When a stop is detected while the counter is running, the counter will maintain its current value. If the waveform generation (WG) is used, all waveforms are set to a state defined in Non-Recoverable State x Output Enable bit and Non- Recoverable State x Output Value bit in the Driver Control register (DRVCTRL.NREx and DRVCTRL.NRVx), and the Stop bit in the Status register is set (STATUS.STOP).

Re-Trigger Command and Event Action

A re-trigger command can be issued from software by using TCC Command bits in Control B Set register (CTRLBSET.CMD=0x1, RETRIGGER), or from event when the re-trigger event action is configured in the Input Event 0/1 Action bits in Event Control register (EVCTRL.EVACTn=0x1, RETRIGGER).

When the command is detected during counting operation, the counter will be reloaded or cleared, depending on the counting direction (CTRLBSET.DIR or CTRLBCLR.DIR). The Re-Trigger bit in the Interrupt Flag Status and Clear register will be set (INTFLAG.TRG). It is also possible to generate an event by writing a '1' to the Re-Trigger Event Output Enable bit in the Event Control register (EVCTRL.TRGEO). If the re-trigger command is detected when the counter is stopped, the counter will resume counting operation from the value in COUNT.

Note:
When a re-trigger event action is configured in the Event Action bits in the Event Control register (EVCTRL.EVACTn=0x1, RETRIGGER), enabling the counter will not start the counter. The counter will start on the next incoming event and restart on corresponding following event.

Start Event Action

The start action can be selected in the Event Control register (EVCTRL.EVACT0=0x3, START) and can start the counting operation when previously stopped. The event has no effect if the counter is already counting. When the module is enabled, the counter operation starts when the event is received or when a re-trigger software command is applied.

Note:

When a start event action is configured in the Event Action bits in the Event Control register (EVCTRL.EVACT0=0x3, START), enabling the counter will not start the counter. The counter will start on the next incoming event, but it will not restart on subsequent events.

Count Event Action

The TCC can count events. When an event is received, the counter increases or decreases the value, depending on direction settings (CTRLBSET.DIR or CTRLBCLR.DIR).

The count event action is selected by the Event Action 0 bit group in the Event Control register (EVCTRL.EVACT0=0x5, COUNT).

Direction Event Action

The direction event action can be selected in the Event Control register (EVCTRL.EVACT1=0x2, DIR). When this event is used, the asynchronous event path specified in the event system must be configured or selected. The direction event action can be used to control the direction of the counter operation, depending on external events level. When received, the event level overrides the Direction settings (CTRLBSET.DIR or CTRLBCLR.DIR) and the direction bit value is updated accordingly.

Increment Event Action

The increment event action can be selected in the Event Control register (EVCTRL.EVACT0=0x4, INC) and can change the counter state when an event is received. When the TCE0 event (TCCx_EV0) is received, the counter increments, whatever the direction setting (CTRLBSET.DIR or CTRLBCLR.DIR) is.

Decrement Event Action

The decrement event action can be selected in the Event Control register (EVCTRL.EVACT1=0x4, DEC) and can change the counter state when an event is received. When the TCE1 (TCCx_EV1) event is received, the counter decrements, whatever the direction setting (CTRLBSET.DIR or CTRLBCLR.DIR) is.

Non-recoverable Fault Event Action

Non-recoverable fault actions can be selected in the Event Control register (EVCTRL.EVACTn=0x7, FAULT). When received, the counter will be stopped and the output of the compare channels is overridden according to the Driver Control register settings (DRVCTRL.NREx and DRVCTRL.NRVx). TCE0 and TCE1 must be configured as asynchronous events.

Event Action Off

If the event action is disabled (EVCTRL.EVACTn=0x0, OFF), enabling the counter will also start the counter.

36.6.2.5. Compare Operations

By default, the Compare/Capture channel is configured for compare operations. To perform capture operations, it must be re-configured.

When using the TCC with the Compare/Capture Value registers (CCx) for compare operations, the counter value is continuously compared to the values in the CCx registers. This can be used for timer or for waveform operation.

The Channel x Compare/Capture Buffer Value (CCBx) registers provide double buffer capability. The double buffering synchronizes the update of the CCx register with the buffer value at the UPDATE condition or a force update command (CTRLBSET.CMD=0x3, UPDATE). For further details, refer to Double Buffering on page 768. The synchronization prevents the occurrence of odd-length, non-symmetrical pulses and ensures glitch-free output.

Waveform Output Generation Operations

The compare channels can be used for waveform generation on output port pins. To make the waveform available on the connected pin, the following requirements must be fulfilled:

1. Choose a waveform generation mode in the Waveform Generation Operation bit in Waveform register (WAVE.WAVEGEN).
2. Optionally invert the waveform output WO[x] by writing the corresponding Waveform Output x Inversion bit in the Driver Control register (DRVCTRL.INVENx).
3. Configure the pins with the I/O Pin Controller. Refer to *PORT - I/O Pin Controller* for details.

The counter value is continuously compared with each CCx value. On a comparison match, the Match or Capture Channel x bit in the Interrupt Flag Status and Clear register (INTFLAG.MCx) will be set on the next zero-to-one transition of CLK_TCC_COUNT (see Normal Frequency Operation). An interrupt and/or event can be generated on the same condition if Match/Capture occurs, i.e. INTENSET.MCx and/or EVCTRL.MCEOx is '1'. Both interrupt and event can be generated simultaneously. The same condition generates a DMA request.

There are seven waveform configurations for the Waveform Generation Operation bit group in the Waveform register (WAVE.WAVEGEN). This will influence how the waveform is generated and impose restrictions on the top value. The configurations are:

- Normal Frequency (NFRQ)
- Match Frequency (MFRQ)
- Normal Pulse-Width Modulation (NPWM)
- Dual-slope, interrupt/event at TOP (DSTOP)
- Dual-slope, interrupt/event at ZERO (DSBOTTOM)
- Dual-slope, interrupt/event at Top and ZERO (DSBOTH)
- Dual-slope, critical interrupt/event at ZERO (DSCRITICAL)

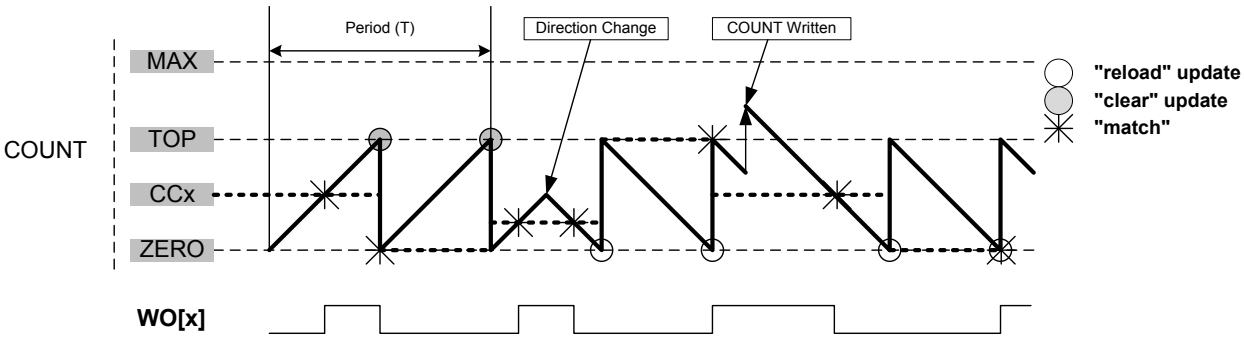
When using MFRQ configuration, the TOP value is defined by the CC0 register value. For the other waveform operations, the TOP value is defined by the Period (PER) register value.

For dual-slope waveform operations, the update time occurs when the counter reaches ZERO. For the other waveforms generation modes, the update time occurs on counter wraparound, on overflow, underflow, or re-trigger.

Normal Frequency (NFRQ)

For Normal Frequency generation, the period time (T) is controlled by the period register (PER). The waveform generation output (WO[x]) is toggled on each compare match between COUNT and CCx, and the corresponding Match or Capture Channel x Interrupt Flag (EVCTRL.MCEOx) will be set.

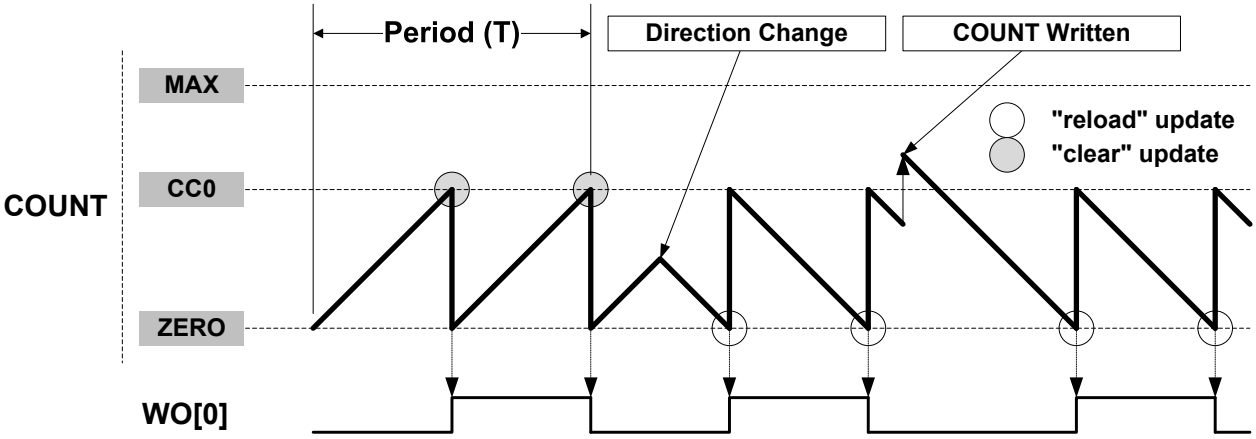
Figure 36-4. Normal Frequency Operation



Match Frequency (MFRQ)

For Match Frequency generation, the period time (T) is controlled by CC0 register instead of PER. WO[0] toggles on each update condition.

Figure 36-5. Match Frequency Operation



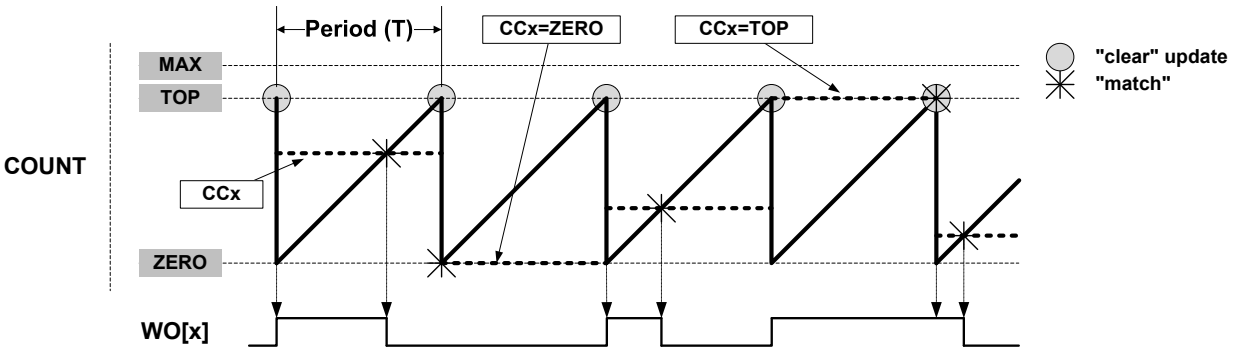
Normal Pulse-Width Modulation (NPWM)

NPWM uses single-slope PWM generation.

Single-Slope PWM Generation

For single-slope PWM generation, the period time (T) is controlled by Top value, and CCx controls the duty cycle of the generated waveform output. When up-counting, the WO[x] is set at start or compare match between the COUNT and TOP values, and cleared on compare match between COUNT and CCx register values. When down-counting, the WO[x] is cleared at start or compare match between the COUNT and ZERO values, and set on compare match between COUNT and CCx register values.

Figure 36-6. Single-Slope PWM Operation



The following equation calculates the exact resolution for a single-slope PWM (R_{PWM_SS}) waveform:

$$R_{PWM\_SS} = \frac{\log(TOP+1)}{\log(2)}$$

The PWM frequency depends on the Period register value (PER) and the peripheral clock frequency (f_{GCLK_TCC}), and can be calculated by the following equation:

$$f_{PWM\_SS} = \frac{f_{GCLK\_TCC}}{N(TOP+1)}$$

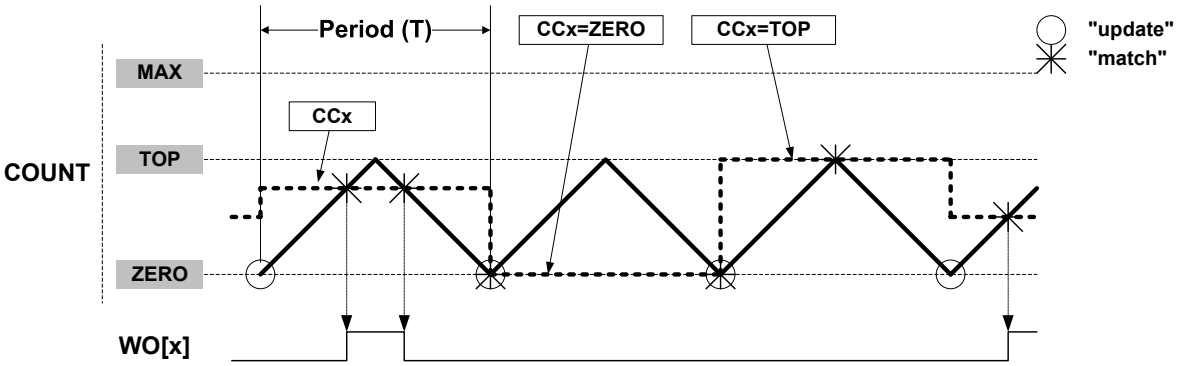
Where N represents the prescaler divider used (1, 2, 4, 8, 16, 64, 256, 1024).

Dual-Slope PWM Generation

For dual-slope PWM generation, the period setting (TOP) is controlled by PER, while CCx control the duty cycle of the generated waveform output. The figure below shows how the counter repeatedly counts from ZERO to PER and then from PER to ZERO. The waveform generator output is set on compare

match when up-counting, and cleared on compare match when down-counting. An interrupt/event is generated on TOP and/or ZERO, depend of Dual slope. In DSBOTH operation, a second update time occurs on TOP.

Figure 36-7. Dual-Slope Pulse Width Modulation



Using dual-slope PWM results in a lower maximum operation frequency compared to single-slope PWM generation. The period (TOP) defines the PWM resolution. The minimum resolution is 1 bit (TOP=0x00000001).

The following equation calculates the exact resolution for dual-slope PWM (R_{PWM_DS}):

$$R_{PWM\_DS} = \frac{\log(PER+1)}{\log(2)}.$$

The PWM frequency f_{PWM_DS} depends on the period setting (TOP) and the peripheral clock frequency f_{GCLK_TCC}, and can be calculated by the following equation:

$$f_{PWM\_DS} = \frac{f_{GCLK\_TCC}}{2N \cdot PER}$$

N represents the prescaler divider used. The waveform generated will have a maximum frequency of half of the TCC clock frequency (f_{GCLK_TCC}) when TOP is set to 0x00000001 and no prescaling is used.

The pulse width (P_{PWM_DS}) depends on the compare channel (CCx) register value and the peripheral clock frequency (f_{GCLK_TCC}), and can be calculated by the following equation:

$$P_{PWM\_DS} = \frac{2N \cdot (TOP - CCx)}{f_{GCLK\_TCC}}$$

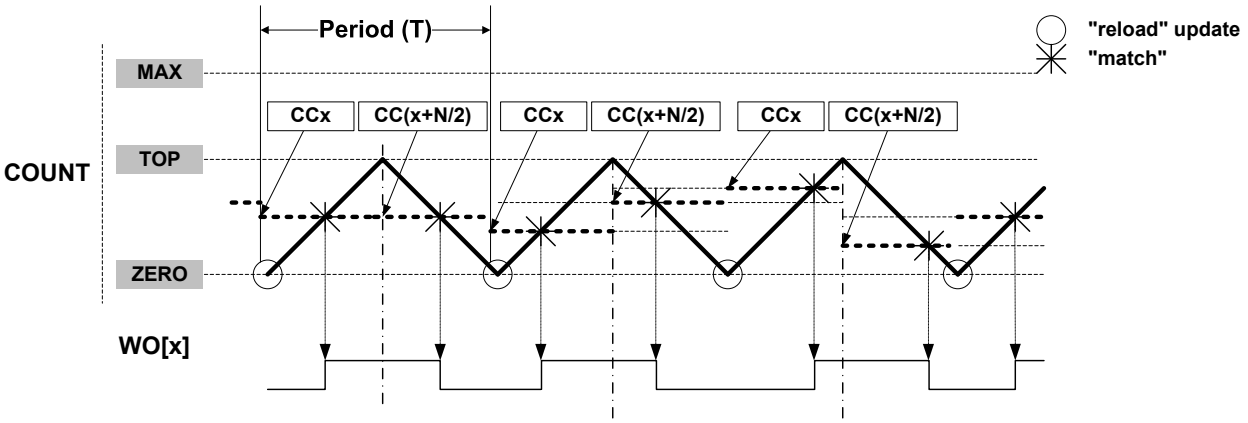
N represents the prescaler divider used.

Note: In DSTOP, DSBOTTOM and DSBOTH operation, when TOP is lower than MAX/2, the CCx MSB bit defines the ramp on which the CCx Match interrupt or event is generated. (Rising if CCx[MSB]=0, falling if CCx[MSB]=1.)

Dual-Slope Critical PWM Generation

Critical mode generation allows generation of non-aligned centered pulses. In this mode, the period time is controlled by PER while CCx control the generated waveform output edge during up-counting and CC(x +CC_NUM/2) control the generated waveform output edge during down-counting.

Figure 36-8. Dual-Slope Critical Pulse Width Modulation (N=CC_NUM)



The table below shows the update counter and overflow event/interrupt generation conditions in different operation modes.

Table 36-3. Counter Update and Overflow Event/interrupt Conditions

Name	Operation	TOP	Update	Output Waveform		OVFIF/Event	
				On Match	On Update	Up	Down
NFRQ	Normal Frequency	PER	TOP/ ZERO	Toggle	Stable	TOP	ZERO
MFRQ	Match Frequency	CC0	TOP/ ZERO	Toggle	Stable	TOP	ZERO
NPWM	Single-slope PWM	PER	TOP/ ZERO	See section 'Output Polarity' below		TOP	ZERO
DSCRITICAL	Dual-slope PWM	PER	ZERO			-	ZERO
DSBOTTOM	Dual-slope PWM	PER	ZERO			-	ZERO
DSBOTH	Dual-slope PWM	PER	TOP & ZERO			TOP	ZERO
DSTOP	Dual-slope PWM	PER	ZERO			TOP	–

Output Polarity

The polarity (WAVE.POLx) is available in all waveform output generation. In single-slope and dual-slope PWM operation, it is possible to invert the pulse edge alignment individually on start or end of a PWM cycle for each compare channels. The table below shows the waveform output set/clear conditions, depending on the settings of timer/counter, direction, and polarity.

Table 36-4. Waveform Generation Set/Clear Conditions

Waveform Generation operation	DIR	POLx	Waveform Generation Output Update	
			Set	Clear
Single-Slope PWM	0	0	Timer/counter matches TOP	Timer/counter matches CCx
		1	Timer/counter matches CC	Timer/counter matches TOP
	1	0	Timer/counter matches CC	Timer/counter matches ZERO
		1	Timer/counter matches ZERO	Timer/counter matches CC
Dual-Slope PWM	x	0	Timer/counter matches CC when counting up	Timer/counter matches CC when counting down
		1	Timer/counter matches CC when counting down	Timer/counter matches CC when counting up

In Normal and Match Frequency, the WAVE.POLx value represents the initial state of the waveform output.

Related Links

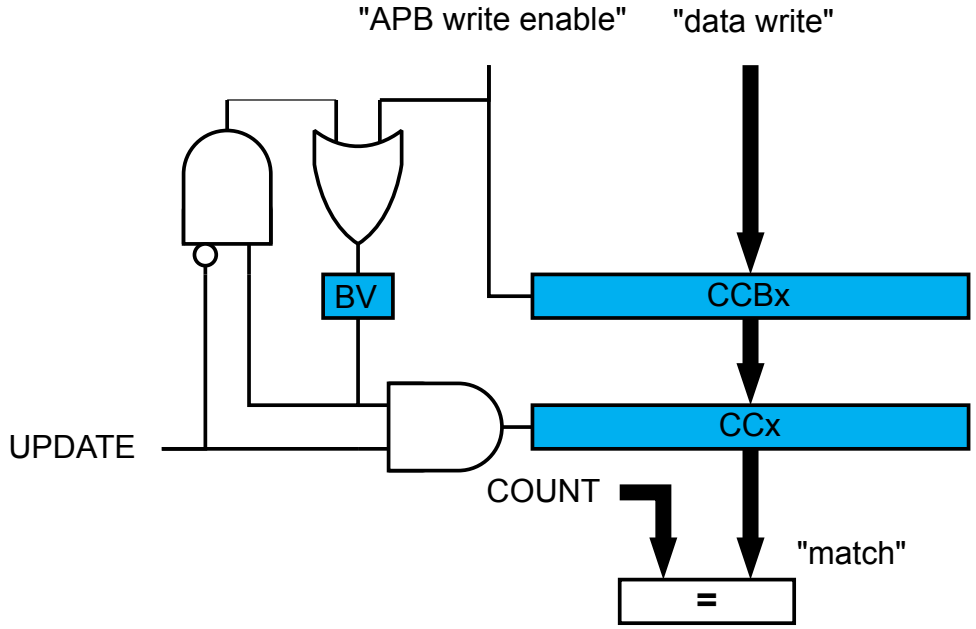
PORT: IO Pin Controller on page 506

36.6.2.6. Double Buffering

The Pattern (PATT), Period (PER) and Compare Channels (CCx) registers are all double buffered. Each buffer register has a buffer valid (PATTBUFV, PERBUFV or CCBUFVx) bit in the STATUS register, which indicates that the buffer register contains a valid value that can be copied into the corresponding register.

When the buffer valid flag bit in the STATUS register is '1' the and write a '0' to CTRLBCLR.LUPD to enable double buffering, the data from the buffer register will be copied into the corresponding register under UPDATE condition (CTRLBSET.CMD=0x3, UPDATE), including the software update command. Then the buffer valid flag bit in the STATUS register are automatically cleared by hardware or software. Double buffering is not applied for capture mode. A compare register is as the following figure.

Figure 36-9. Compare Channel Double Buffering



Both the registers (PATT/PER/CCx) and corresponding buffer registers (PATTBPERBUF/CCBUFx) are available in the I/O register map, and the double buffering feature is not mandatory. The double buffering is disabled by writing a '1' to CTRLSET.LUPD. This allows initialization and bypassing of the buffer register and the double buffering feature.

Note: In NFRQ, MFRQ or PWM down-counting counter mode (CTRLBSET.DIR=1), the PER register and the PERBUF register are written simultaneously if double buffering is active or as soon as double buffering is activated (CTRLBCLR.LUPD=0).

Changing the Period

The counter period is changed by writing a new Top value to the Period register (PER or CC0, depending on the waveform generation mode). If double buffering is off (CTRLBSET.LUPD=1), any period update is effective after the synchronization delay.

Figure 36-10. Unbuffered Single-Slope Up-Counting Operation

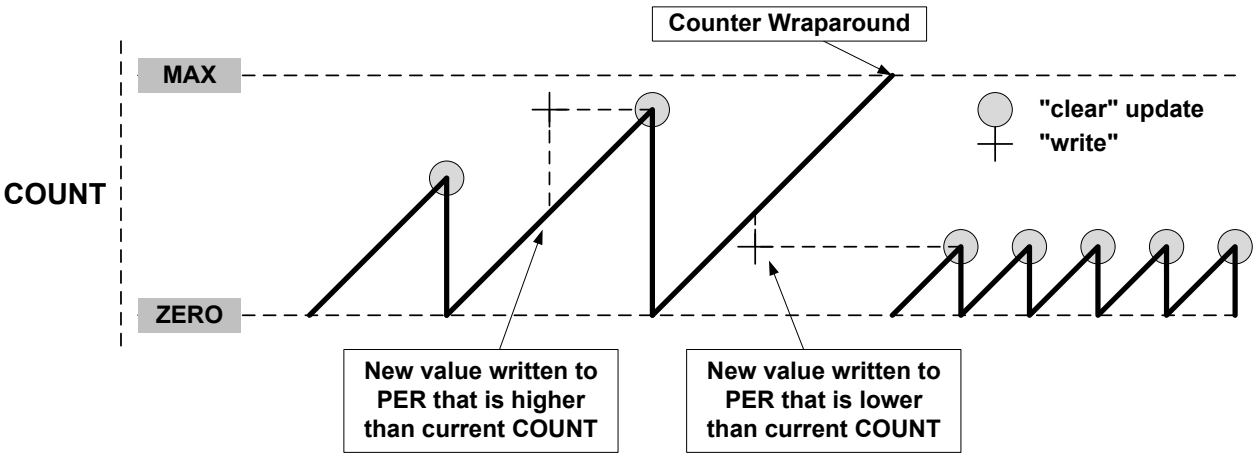
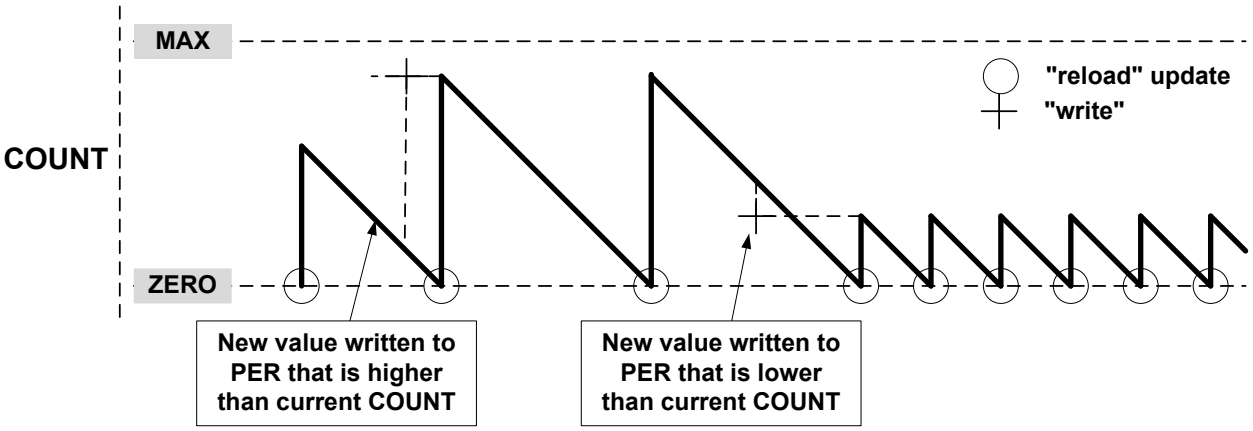
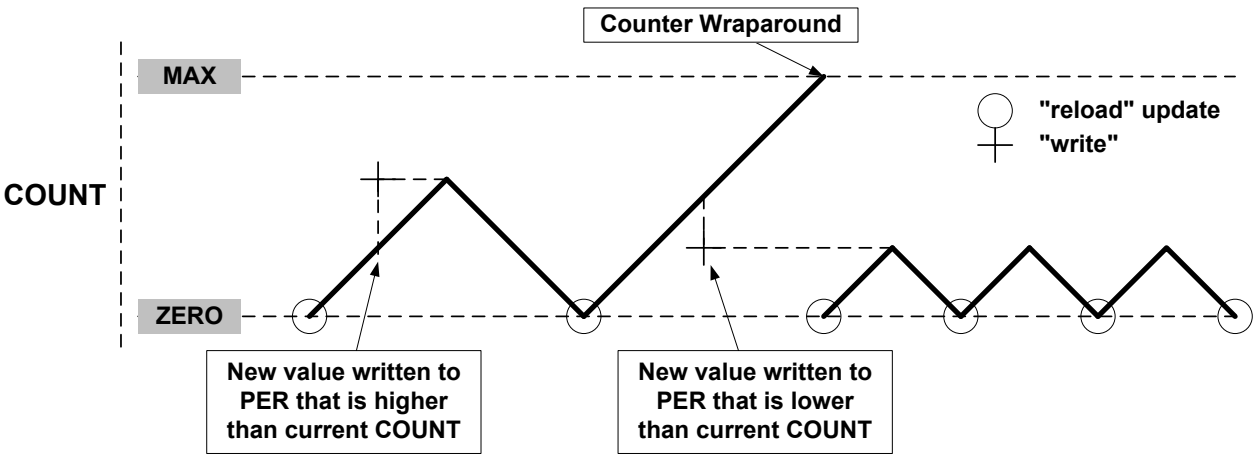


Figure 36-11. Unbuffered Single-Slope Down-Counting Operation



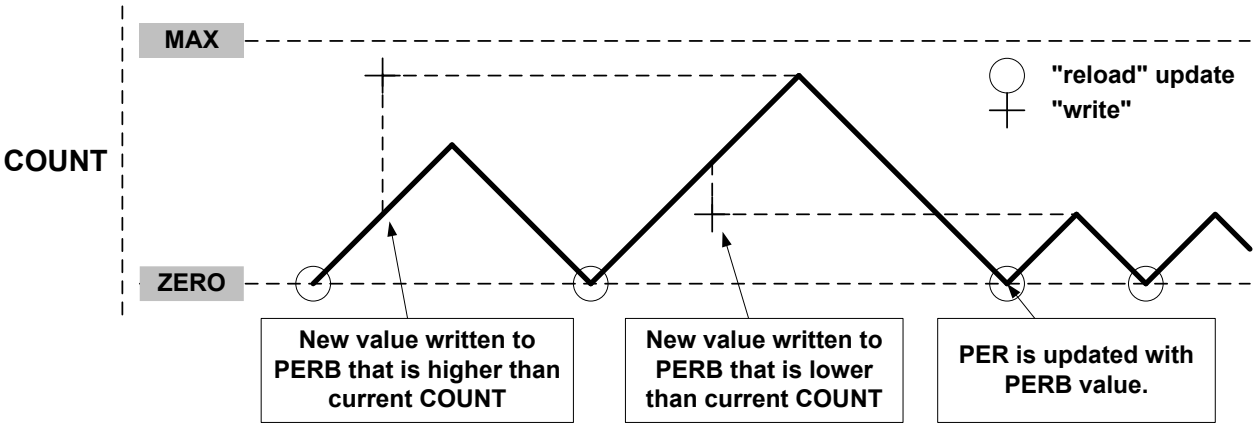
A counter wraparound can occur in any operation mode when up-counting without buffering, see Figure 36-10 Unbuffered Single-Slope Up-Counting Operation on page 769. COUNT and TOP are continuously compared, so when a new value that is lower than the current COUNT is written to TOP, COUNT will wrap before a compare match.

Figure 36-12. Unbuffered Dual-Slope Operation



When double buffering is used, the buffer can be written at any time and the counter will still maintain correct operation. The period register is always updated on the update condition, as shown in Figure 36-13 Changing the Period Using Buffering on page 770. This prevents wraparound and the generation of odd waveforms.

Figure 36-13. Changing the Period Using Buffering



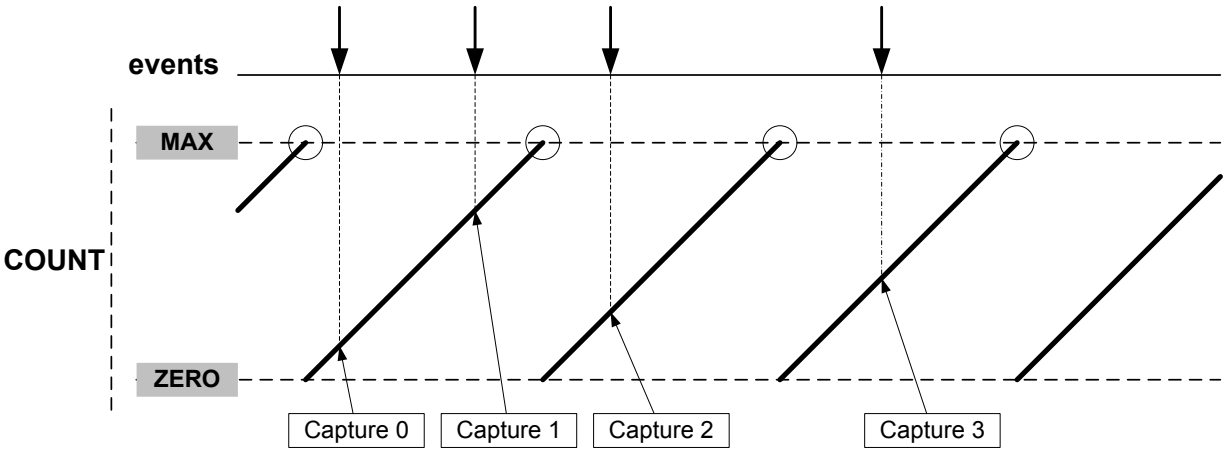
36.6.2.7. Capture Operations

To enable and use capture operations, the Match or Capture Channel x Event Input Enable bit in the Event Control register (EVCTRL.MCEIx) must be written to '1'. The capture channels to be used must also be enabled in the Capture Channel x Enable bit in the Control A register (CTRLA.CPTENx) before capturing can be performed.

Event Capture Action

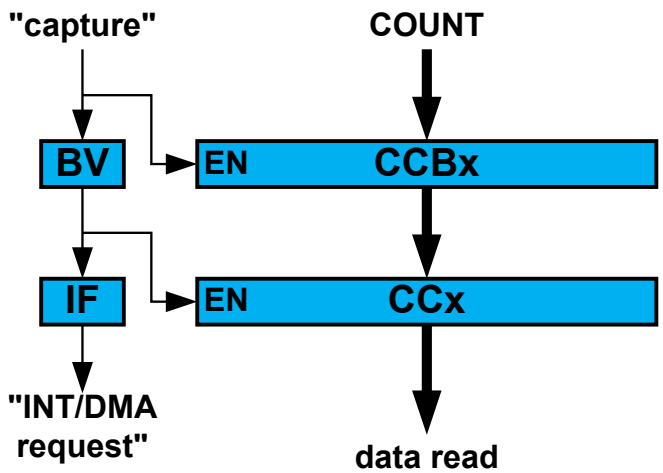
The compare/capture channels can be used as input capture channels to capture events from the Event System, and give them a timestamp. The following figure shows four capture events for one capture channel.

Figure 36-14. Input Capture Timing



For input capture, the buffer register and the corresponding CCx act like a FIFO. When CCx is empty or read, any content in CCBx is transferred to CCx. The buffer valid flag is passed to set the CCx interrupt flag (IF) and generate the optional interrupt, event or DMA request.

Figure 36-15. Capture Double Buffering



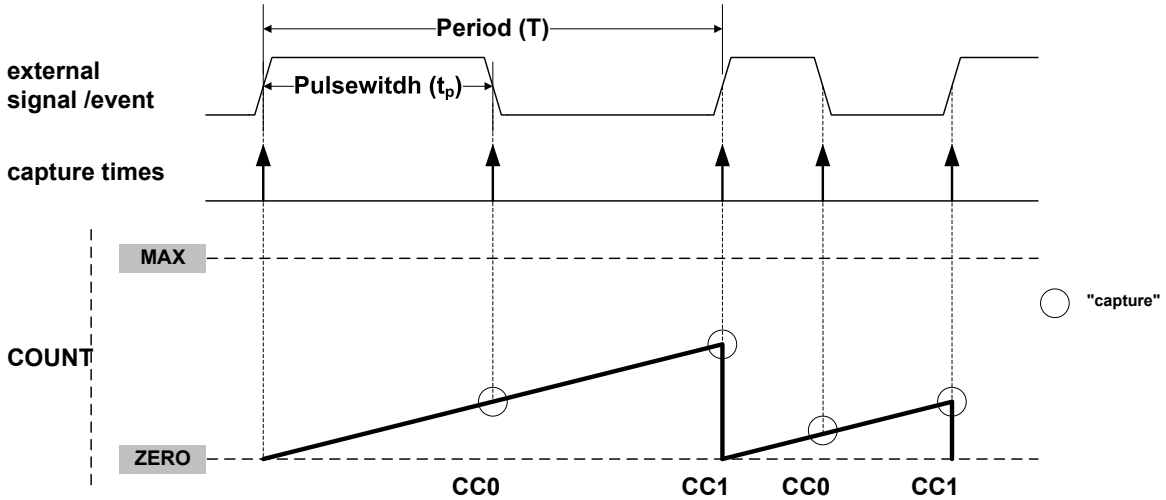
The TCC can detect capture overflow of the input capture channels: When a new capture event is detected while the Capture Buffer Valid flag (STATUS.CCBV) is still set, the new timestamp will not be stored and INTFLAG.ERR will be set.

Period and Pulse-Width (PPW) Capture Action

The TCC can perform two input captures and restart the counter on one of the edges. This enables the TCC to measure the pulse-width and period and to characterize the frequency *f* and dutyCycle of an input signal:

$$f = \frac{1}{T} \quad , \quad dutyCycle = \frac{t_p}{T}$$

Figure 36-16. PWP Capture



Selecting PWP or PPW in the Timer/Counter Event Input 1 Action bit group in the Event Control register (EVCTRL.EVACT1) enables the TCC to perform one capture action on the rising edge and the other one on the falling edge. When using PPW (period and pulse-width) event action, period *T* will be captured into CC0 and the pulse-width t_p into CC1. The PWP (Pulse-width and Period) event action offers the same functionality, but *T* will be captured into CC1 and t_p into CC0.

The Timer/Counter Event x Invert Enable bit in Event Control register (EVCTRL.TCEINVx) is used for event source x to select whether the wraparound should occur on the rising edge or the falling edge. If EVCTRL.TCEINVx=1, the wraparound will happen on the falling edge.

The corresponding capture is done only if the channel is enabled in capture mode (CTRLA.CPTENx=1). If not, the capture action will be ignored and the channel will be enabled in compare mode of operation. When only one of these channel is required, the other channel can be used for other purposes.

The TCC can detect capture overflow of the input capture channels: When a new capture event is detected while the INTFLAG.MCx is still set, the new timestamp will not be stored and INTFLAG.ERR will be set.

Note: In dual-slope PWM operation, and when TOP is lower than MAX/2, the CCx MSB captures the CTRLB.DIR state to identify the ramp on which the capture has been done. For rising ramps CCx[MSB] is zero, for falling ramps CCx[MSB]=1.

36.6.3. Additional Features

36.6.3.1. One-Shot Operation

When one-shot is enabled, the counter automatically stops on the next counter overflow or underflow condition. When the counter is stopped, the Stop bit in the Status register (STATUS.STOP) is set and the waveform outputs are set to the value defined by DRVCTRL.NREx and DRVCTRL.NRVx.

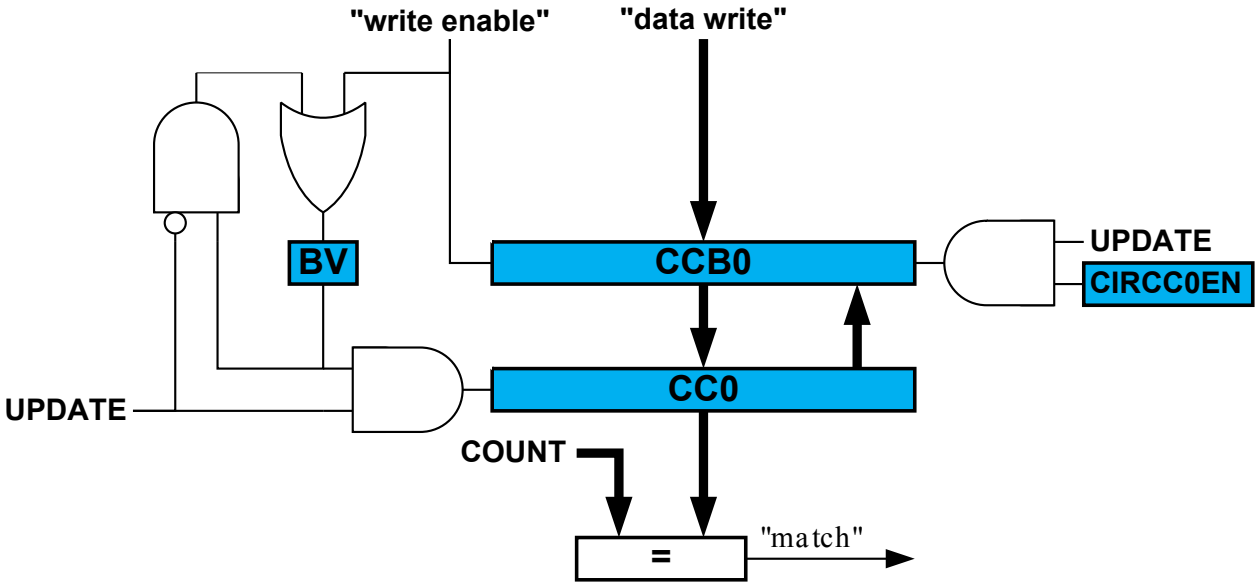
One-shot operation can be enabled by writing a '1' to the One-Shot bit in the Control B Set register (CTRLBSET.ONESHOT) and disabled by writing a '1' to CTRLBCLR.ONESHOT. When enabled, the TCC will count until an overflow or underflow occurs and stop counting. The one-shot operation can be

restarted by a re-trigger software command, a re-trigger event or a start event. When the counter restarts its operation, STATUS.STOP is automatically cleared.

36.6.3.2. Circular Buffer

The Period register (PER) and the compare channels register (CC0 to CC3) support circular buffer operation. When circular buffer operation is enabled, the PER or CCx values are copied into the corresponding buffer registers at each update condition. Circular buffering is dedicated to RAMP2, RAMP2A, and DSBOTH operations.

Figure 36-17. Circular Buffer on Channel 0



36.6.3.3. Dithering Operation

The TCC supports dithering on Pulse-width or Period on a 16, 32 or 64 PWM cycles frame.

Dithering consists in adding some extra clocks cycles in a frame of several PWM cycles, and can improve the accuracy of the *average* output pulse width and period. The extra clock cycles are added on some of the compare match signals, one at a time, through a "blue noise" process that minimizes the flickering on the resulting dither patterns.

Dithering is enabled by writing the corresponding configuration in the Enhanced Resolution bits in CTRLA register (CTRLA.RESOLUTION):

- DITH4 enable dithering every 16 PWM frames
- DITH5 enable dithering every 32 PWM frames
- DITH6 enable dithering every 64 PWM frames

The DITHERCY bits of COUNT, PER and CCx define the number of extra cycles to add into the frame (DITHERCY bits from the respective COUNT, PER or CCx registers). The remaining bits of COUNT, PER, CCx define the compare value itself.

The pseudo code, giving the extra cycles insertion regarding the cycle is:

```
int extra_cycle(resolution, dithercy, cycle){
  int MASK;
  int value
  switch (resolution){
    DITH4: MASK = 0x0f;
    DITH5: MASK = 0x1f;
    DITH6: MASK = 0x3f;
  }
```

```
  value = cycle * dithercy;
  if (((MASK & value) + dithercy) > MASK)
    return 1;
 return 0;
}
```

Dithering on Period

Writing DITHERCY in PER will lead to an average PWM period configured by the following formulas.

DITH4 mode:

$$PwmPeriod = \left(\frac{\mathrm{DITHERCY}}{16} + \mathrm{PER}\right)\left(\frac{1}{f_{\mathrm{GCLK\_TCC}}}\right)$$

Note: If DITH4 mode is enabled, the last 4 significant bits from PER/CCx or COUNT register correspond to the DITHERCY value, rest of the bits corresponds to PER/CCx or COUNT value.

DITH5 mode:

$$PwmPeriod = \left(\frac{\mathrm{DITHERCY}}{32} + \mathrm{PER}\right)\left(\frac{1}{f_{\mathrm{GCLK\_TCC}}}\right)$$

DITH6 mode:

$$PwmPeriod = \left(\frac{\mathrm{DITHERCY}}{64} + \mathrm{PER}\right)\left(\frac{1}{f_{\mathrm{GCLK\_TCC}}}\right)$$

Dithering on Pulse Width

Writing DITHERCY in CCx will lead to an average PWM pulse width configured by the following formula.

DITH4 mode:

$$PwmPulseWidth = \left(\frac{\mathrm{DITHERCY}}{16} + \mathrm{CCx}\right)\left(\frac{1}{f_{\mathrm{GCLK\_TCC}}}\right)$$

DITH5 mode:

$$PwmPulseWidth = \left(\frac{\mathrm{DITHERCY}}{32} + \mathrm{CCx}\right)\left(\frac{1}{f_{\mathrm{GCLK\_TCC}}}\right)$$

DITH6 mode:

$$PwmPulseWidth = \left(\frac{\mathrm{DITHERCY}}{64} + \mathrm{CCx}\right)\left(\frac{1}{f_{\mathrm{GCLK\_TCC}}}\right)$$

Note: The PWM period will remain static in this case.

36.6.3.4. Ramp Operations

Three ramp operation modes are supported. All of them require the timer/counter running in single-slope PWM generation. The ramp mode is selected by writing to the Ramp Mode bits in the Waveform Control register (WAVE.RAMP).

RAMP1 Operation

This is the default PWM operation, described in Single-Slope PWM Generation.

RAMP2 Operation

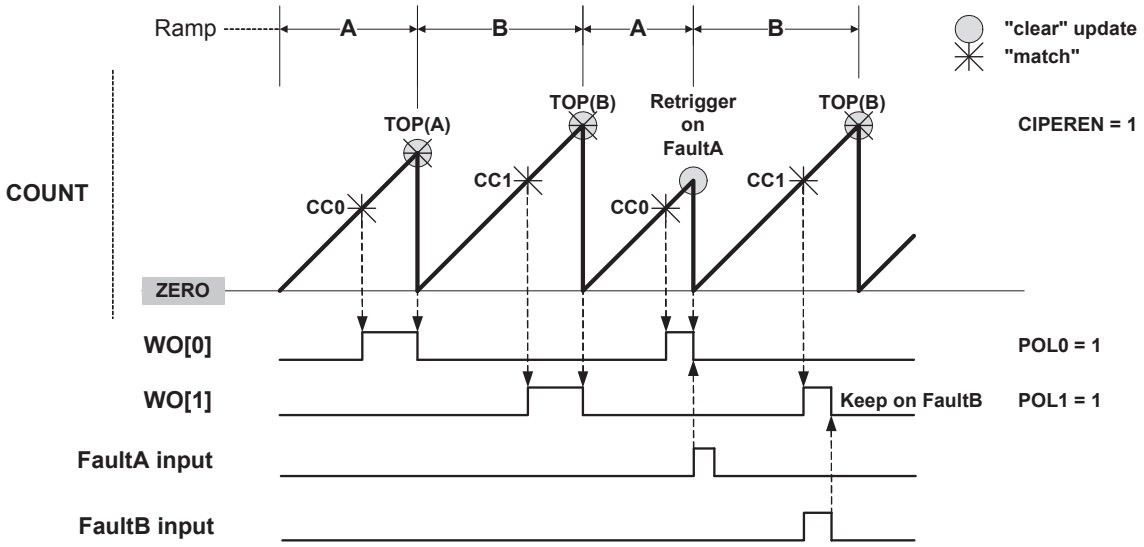
These operation modes are dedicated for power factor correction (PFC), Half-Bridge and Push-Pull SMPS topologies, where two consecutive timer/counter cycles are interleaved, see Figure 36-18 RAMP2

Standard Operation on page 775. In cycle A, odd channel output is disabled, and in cycle B, even channel output is disabled. The ramp index changes after each update, but can be software modified using the Ramp index command bits in Control B Set register (CTRLBSET.IDXCMD).

Standard RAMP2 (RAMP2) Operation

Ramp A and B periods are controlled by the PER register value. The PER value can be different on each ramp by the Circular Period buffer option in the Wave register (WAVE.CIPEREN=1). This mode uses a two-channel TCC to generate two output signals, or one output signal with another CC channel enabled in capture mode.

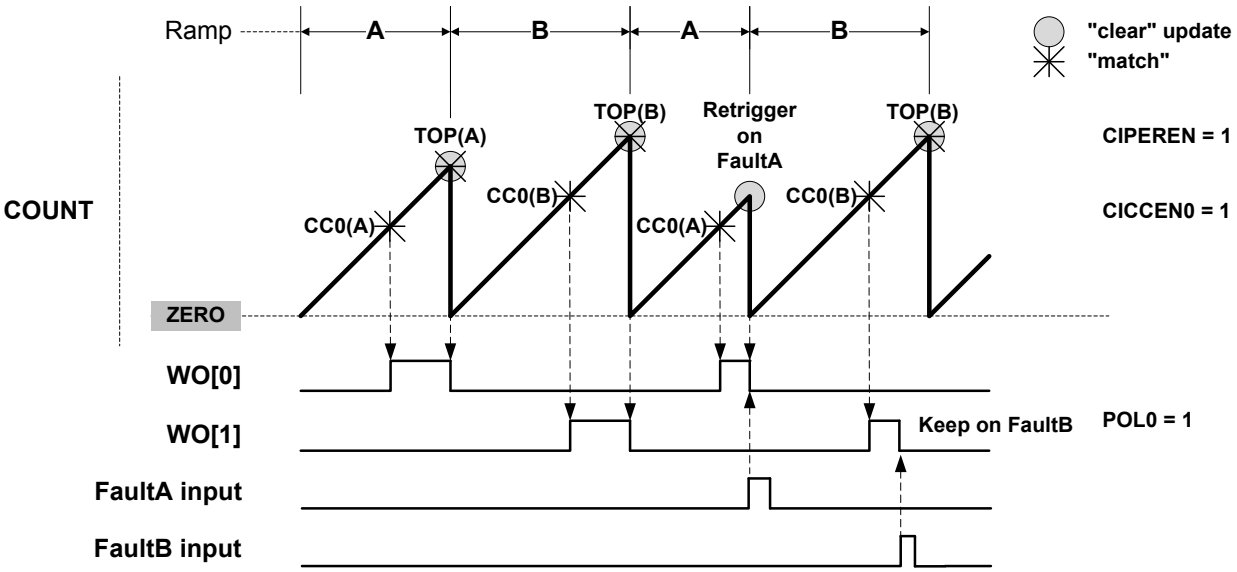
Figure 36-18. RAMP2 Standard Operation



Alternate RAMP2 (RAMP2A) Operation

Alternate RAMP2 operation is similar to RAMP2, but CC0 controls both WO[0] and WO[1] waveforms when the corresponding circular buffer option is enabled (CIPEREN=1). The waveform polarity is the same on both outputs. Channel 1 can be used in capture mode.

Figure 36-19. RAMP2 Alternate Operation



Critical RAMP2 (RAMP2C) Operation

Critical RAMP2 operation provides a way to cover RAMP2 operation requirements without the update constraint associated to the use of circular buffers. In this mode, CC0 is controlling the period of ramp A and PER is controlling the period of ramp B. When using more than two channels, WO[0] output is controlled by CC2 (HIGH) and CC0 (LOW). On TCC with 2 channels, a pulse on WO[0] will last the entire period of ramp A, if WAVE.POL0=0.

36.6.3.5. Recoverable Faults

Recoverable faults can restart or halt the timer/counter. Two faults, called Fault A and Fault B, can trigger recoverable fault actions on the compare channels CC0 and CC1 of the TCC. The compare channels' outputs can be clamped to inactive state either as long as the fault condition is present, or from the first valid fault condition detection on until the end of the timer/counter cycle.

Fault Inputs

The first two channel input events (TCCxMC0 and TCCxMC1) can be used as Fault A and Fault B inputs, respectively. Event system channels connected to these fault inputs must be configured as asynchronous. The TCC must work in a PWM mode.

Fault Filtering

There are three filters available for each input Fault A and Fault B. They are configured by the corresponding Recoverable Fault n Configuration registers (FCTRLA and FCTRLB). The three filters can either be used independently or in any combination.

Input Filtering

By default, the event detection is asynchronous. When the event occurs, the fault system will immediately and asynchronously perform the selected fault action on the compare channel output, also in device power modes where the clock is not available. To avoid false fault detection on external events (e.g. due to a glitch on an I/O port) a digital filter can be enabled and configured by the Fault B Filter Value bits in the Fault n Configuration registers (FCTRLn.FILTERVAL). If the event width is less than FILTERVAL (in clock cycles), the event will be discarded. A valid event will be delayed by FILTERVAL clock cycles.

Fault Blanking

This ignores any fault input for a certain time just after a selected waveform output edge. This can be used to prevent false fault triggering due to signal bouncing, as shown in the figure below. Blanking can be enabled by writing an edge triggering configuration to the Fault n Blanking Mode bits in the Recoverable Fault n Configuration register (FCTRLn.BLANK). The desired duration of the blanking must be written to the Fault n Blanking Time bits (FCTRLn.BLANKVAL).
The blanking time $t_b$is calculated by

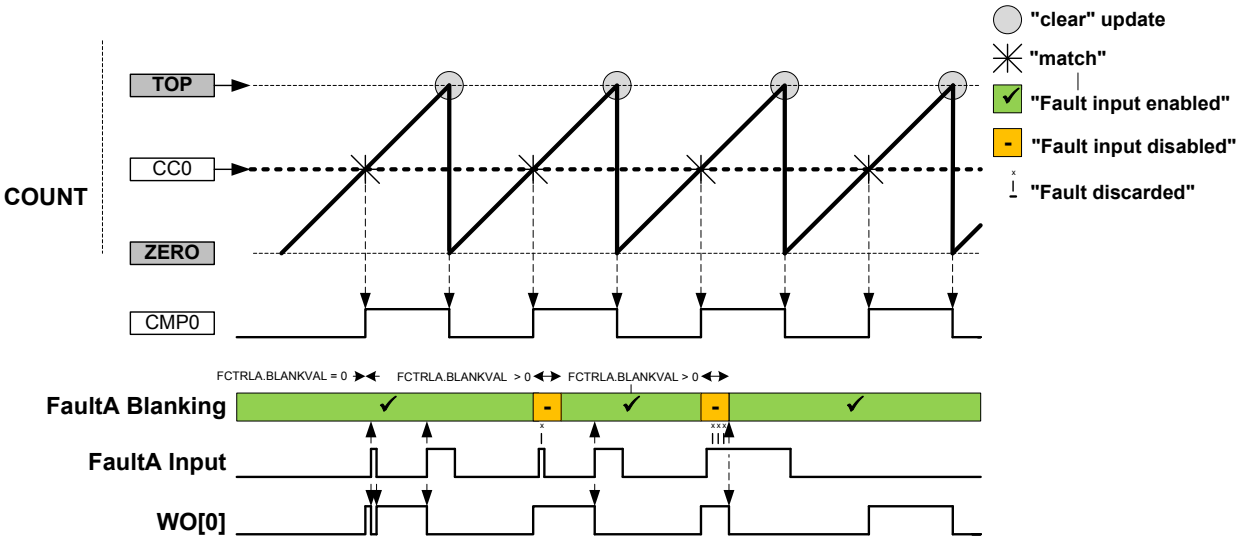
$$t_b = \frac{1 + \mathrm{BLANKVAL}}{f_{\mathrm{GCLK\_TCCx\_PRESC}}}$$

Here, $f_{GCLK\_TCCx\_PRESC}$ is the frequency of the prescaled peripheral clock frequency f_{GCLK_TCCx}.

The prescaler is enabled by writing '1' to the Fault n Blanking Prescaler bit (FCTRLn.BLANKPRESC). When disabled, $f_{GCLK\_TCCx\_PRESC}$=f_{GCLK_TCCx}. When enabled, $f_{GCLK\_TCCx\_PRESC}$=f_{GCLK_TCCx}/64.

The maximum blanking time (FCTRLn.BLANKVAL=
255) at f_{GCLK_TCCx}=96MHz is 2.67µs (no prescaler) or 170µs (prescaling). For f_{GCLK_TCCx}=1MHz, the maximum blanking time is either 170µs (no prescaling) or 10.9ms (prescaling enabled).

Figure 36-20. Fault Blanking in RAMP1 Operation with Inverted Polarity



Fault Qualification

This is enabled by writing a '1' to the Fault n Qualification bit in the Recoverable Fault n Configuration register (FCTRLn.QUAL). When the recoverable fault qualification is enabled (FCTRLn.QUAL=1), the fault input is disabled all the time the corresponding channel output has an inactive level, as shown in the figures below.

Figure 36-21. Fault Qualification in RAMP1 Operation

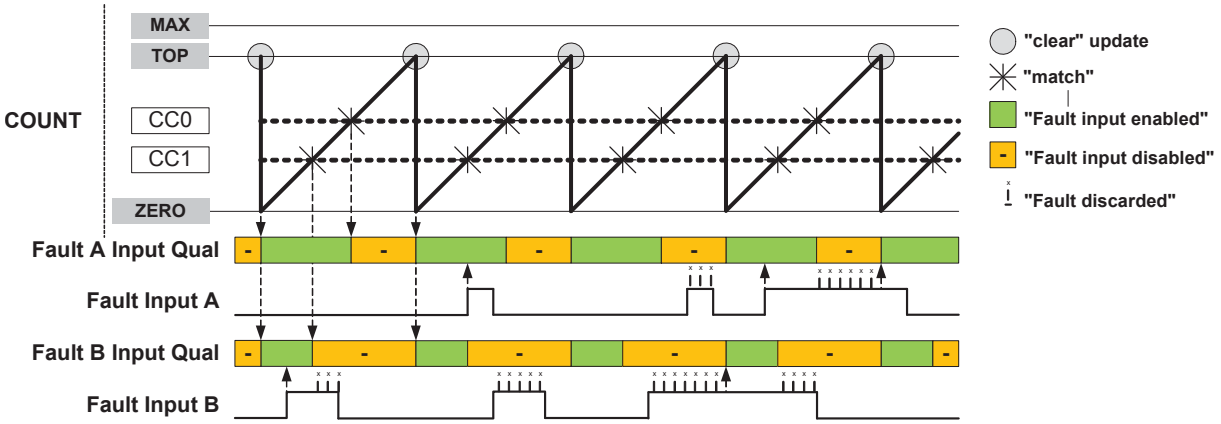
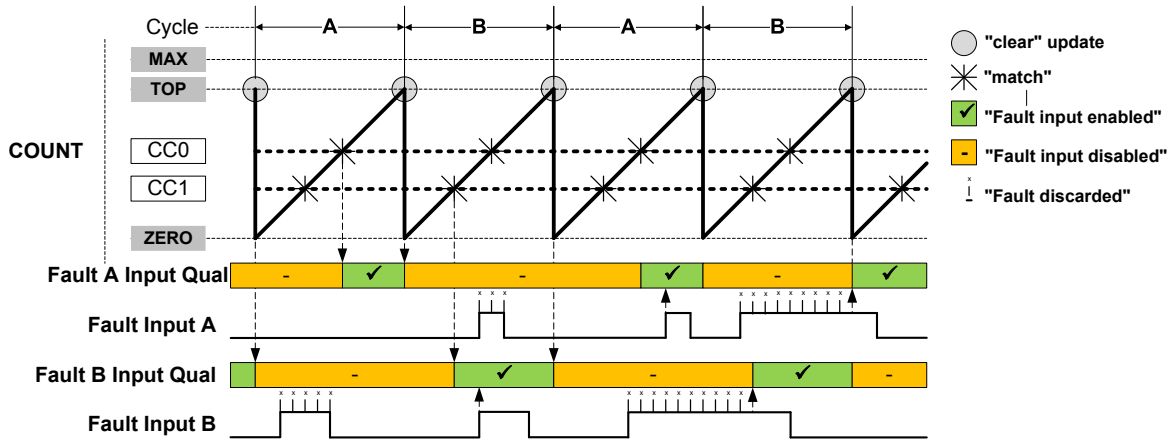


Figure 36-22. Fault Qualification in RAMP2 Operation with Inverted Polarity

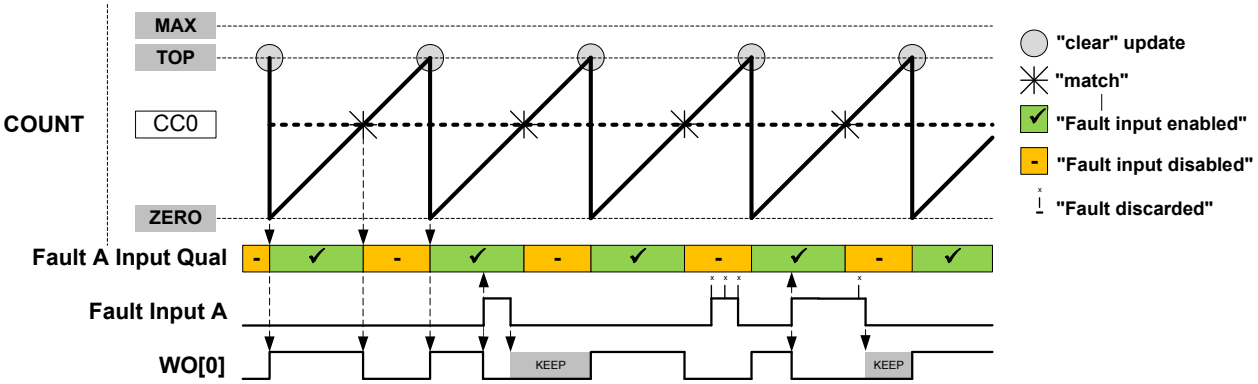


Fault Actions

Different fault actions can be configured individually for Fault A and Fault B. Most fault actions are not mutually exclusive; hence two or more actions can be enabled at the same time to achieve a result that is a combination of fault actions.

Keep Action This is enabled by writing the Fault n Keeper bit in the Recoverable Fault n Configuration register (FCTRLn.KEEP) to '1'. When enabled, the corresponding channel output will be clamped to zero as long as the fault condition is present. The clamp will be released on the start of the first cycle after the fault condition is no longer present, see next Figure.

Figure 36-23. Waveform Generation with Fault Qualification and Keep Action



Restart Action This is enabled by writing the Fault n Restart bit in Recoverable Fault n Configuration register (FCTRLn.RESTART) to '1'. When enabled, the timer/counter will be restarted as soon as the corresponding fault condition is present. The ongoing cycle is stopped and the timer/counter starts a new cycle, see Figure 36-24 Waveform Generation in RAMP1 mode with Restart Action on page 778. In Ramp 1 mode, when the new cycle starts, the compare outputs will be clamped to inactive level as long as the fault condition is present.

Note: For RAMP2 operation, when a new timer/counter cycle starts the cycle index will change automatically, see Figure 36-25 Waveform Generation in RAMP2 mode with Restart Action on page 779. Fault A and Fault B are qualified only during the cycle A and cycle B respectively: Fault A is disabled during cycle B, and Fault B is disabled during cycle A.

Figure 36-24. Waveform Generation in RAMP1 mode with Restart Action

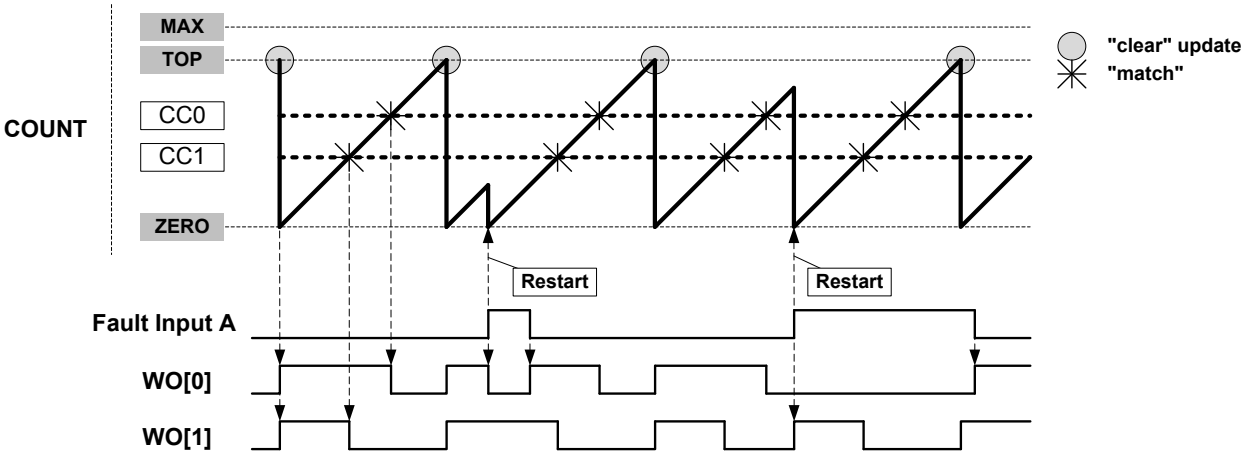
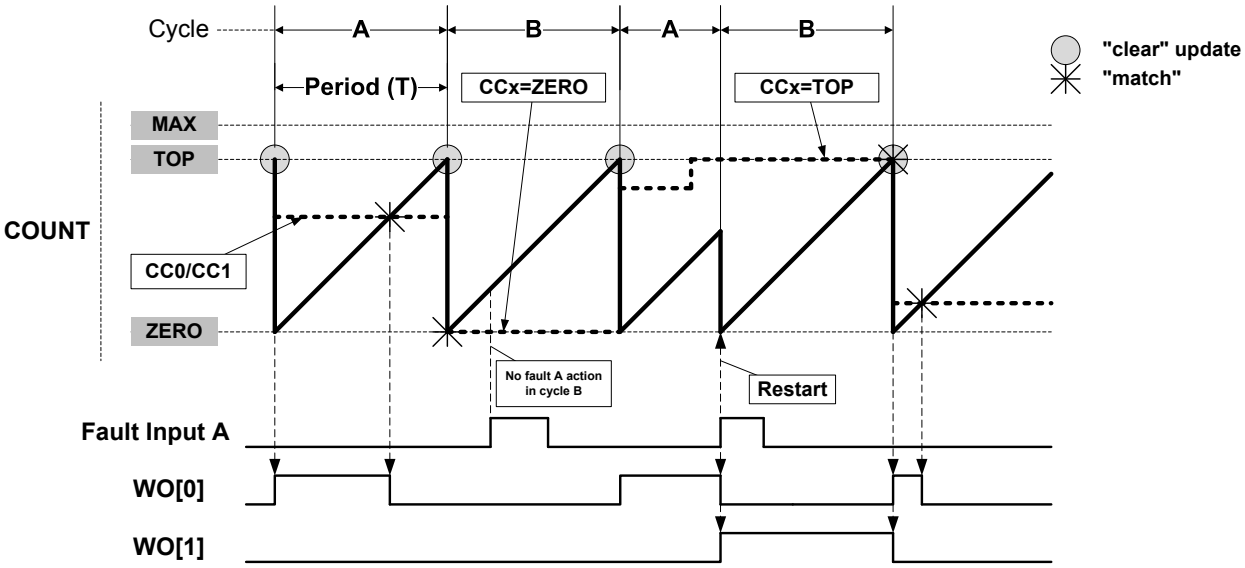


Figure 36-25. Waveform Generation in RAMP2 mode with Restart Action



Capture Action

Several capture actions can be selected by writing the Fault n Capture Action bits in the Fault n Control register (FCTRLn.CAPTURE). When one of the capture operations is selected, the counter value is captured when the fault occurs. These capture operations are available:

- CAPT - the equivalent to a standard capture operation, for further details refer to Capture Operations on page 771
- CAPTMIN - gets the minimum time stamped value: on each new local minimum captured value, an event or interrupt is issued.
- CAPTMAX - gets the maximum time stamped value: on each new local maximum captured value, an event or interrupt (IT) is issued, see Figure 36-26 Capture Action "CAPTMAX" on page 780.
- LOCMIN - notifies by event or interrupt when a local minimum captured value is detected.
- LOCMAX - notifies by event or interrupt when a local maximum captured value is detected.
- DERIV0 - notifies by event or interrupt when a local extreme captured value is detected, see Figure 36-27 Capture Action "DERIV0" on page 780.

CCx Content:

In CAPTMIN and CAPTMAX operations, CCx keeps the respective extremum values, see Figure 36-26 Capture Action "CAPTMAX" on page 780. In LOCMIN, LOCMAX or DERIV0 operation, CCx follows the counter value at fault time, see Figure 36-27 Capture Action "DERIV0" on page 780.

MCx Behaviour:

In LOCMIN and LOCMAX operation, capture is performed on each capture event. The MCx interrupt flag is set only when the captured value is lower (for LOCMIN) or higher (for LOCMAX) than the previous captured value. DERIV0 is equivalent to an OR function of (LOCMIN, LOCMAX).

In CAPT operation, capture is performed on each capture event. The MCx interrupt flag is set on each new capture.

In CAPTMIN and CAPTMAX operation, capture is performed only when a new lower (for CAPTMIN) or new higher (for CAPMAX) value is detected. The MCx interrupt flag is set on each new capture.

Interrupt Generation

In CAPT mode, an interrupt is generated on each filtered Fault n and each dedicated CCx channel capture counter value. In other modes, an interrupt is only generated on an extreme captured value.

Figure 36-26. Capture Action “CAPTMAX”

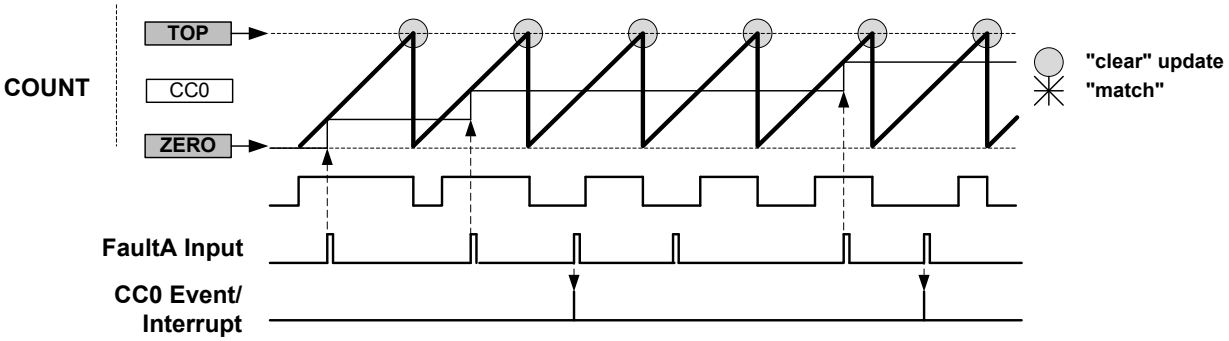
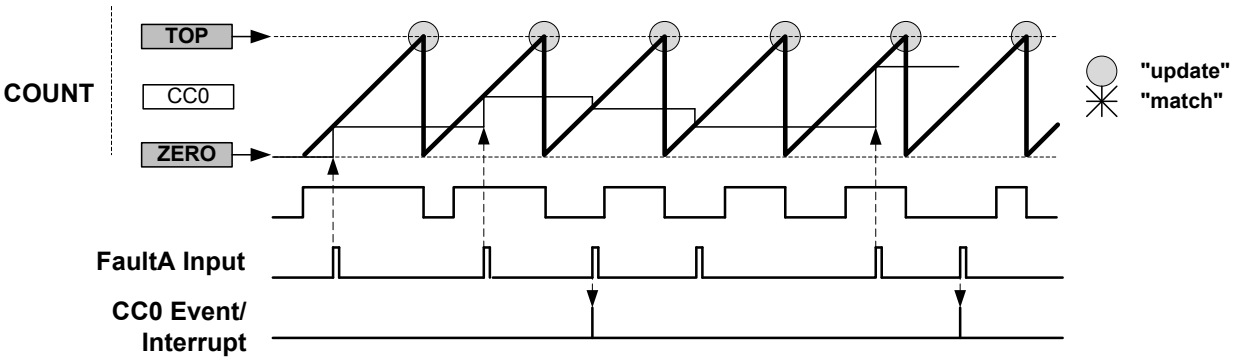


Figure 36-27. Capture Action “DERIV0”



Hardware Halt Action

This is configured by writing 0x1 to the Fault n Halt mode bits in the Recoverable Fault n Configuration register (FCTRLn.HALT). When enabled, the timer/counter is halted and the cycle is extended as long as the corresponding fault is present.

The next figure ('Waveform Generation with Halt and Restart Actions') shows an example where both restart action and hardware halt action are enabled for Fault A. The compare channel 0 output is clamped to inactive level as long as the timer/counter is halted. The timer/counter resumes the counting operation as soon as the fault condition is no longer present. As the restart action is enabled in this example, the timer/counter is restarted after the fault condition is no longer present.

The figure after that ('Waveform Generation with Fault Qualification, Halt, and Restart Actions') shows a similar example, but with additionally enabled fault qualification. Here, counting is resumed after the fault condition is no longer present.

Note that in RAMP2 and RAMP2A operations, when a new timer/counter cycle starts, the cycle index will automatically change.

Figure 36-28. Waveform Generation with Halt and Restart Actions

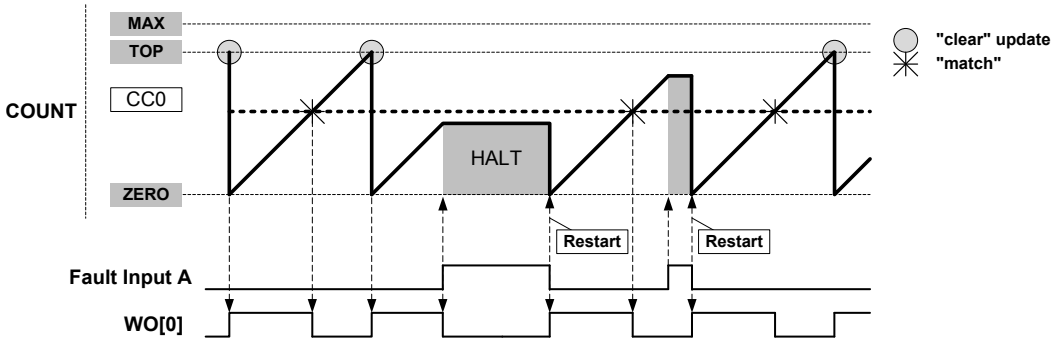
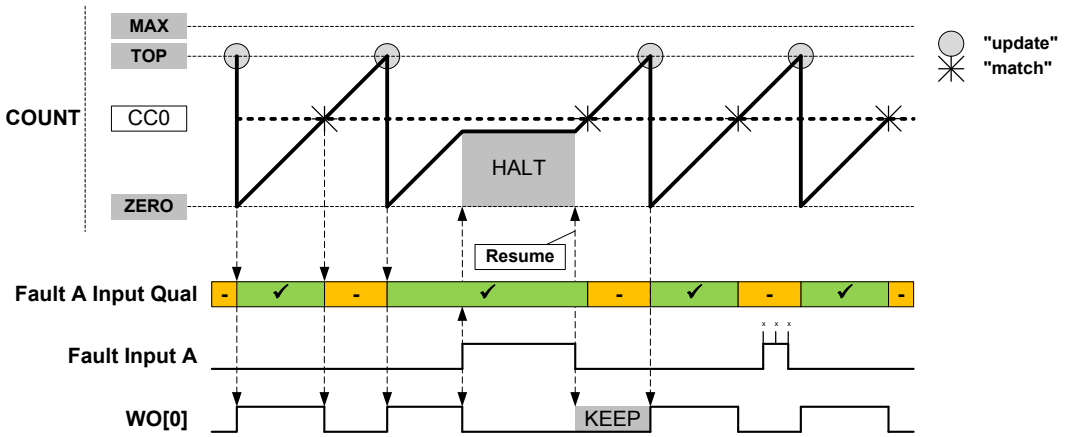


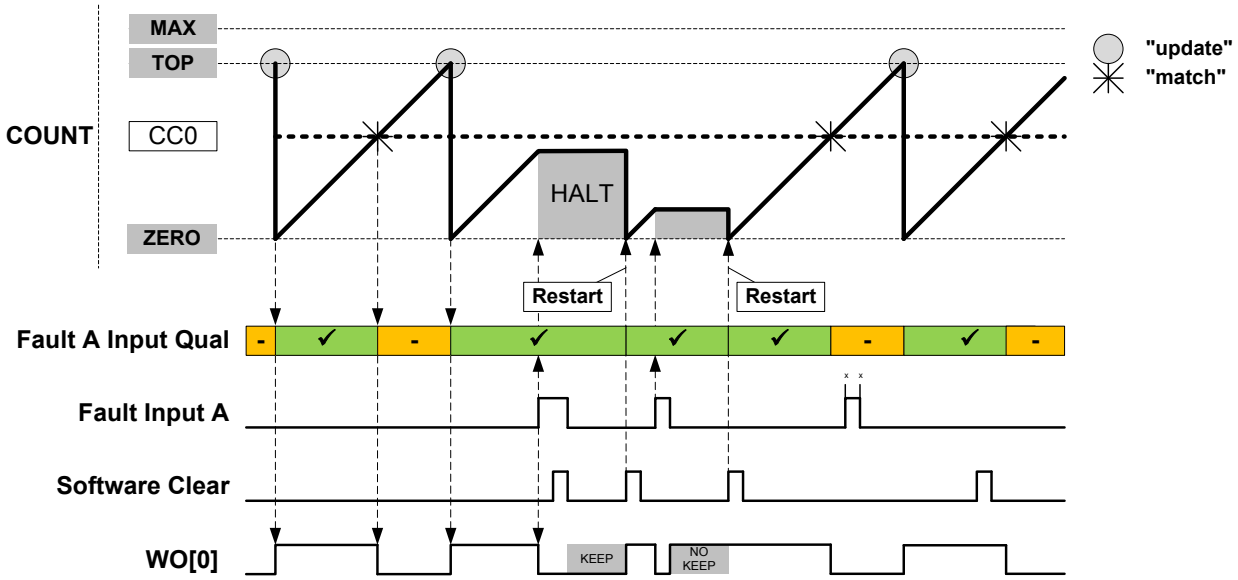
Figure 36-29. Waveform Generation with Fault Qualification, Halt, and Restart Actions



Software Halt Action

This is configured by writing 0x2 to the Fault n Halt mode bits in the Recoverable Fault n configuration register (FCTRLn.HALT). Software halt action is similar to hardware halt action, but in order to restart the timer/counter, the corresponding fault condition must not be present anymore, and the corresponding FAULT n bit in the STATUS register must be cleared by software.

Figure 36-30. Waveform Generation with Software Halt, Fault Qualification, Keep and Restart Actions



36.6.3.6. Non-Recoverable Faults

The non-recoverable fault action will force all the compare outputs to a pre-defined level programmed into the Driver Control register (DRVCTRL.NRE and DRVCTRL.NRV). The non-recoverable fault input (EV0 and EV1) actions are enabled in Event Control register (EVCTRL.EVACT0 and EVCTRL.EVACT1).

To avoid false fault detection on external events (e.g. a glitch on an I/O port) a digital filter can be enabled using Non-Recoverable Fault Input x Filter Value bits in the Driver Control register (DRVCTRL.FILTERVALn). Therefore, the event detection is synchronous, and event action is delayed by the selected digital filter value clock cycles.

When the Fault Detection on Debug Break Detection bit in Debug Control register (DGBCTRL.FDDBD) is written to '1', a non-recoverable Debug Faults State and an interrupt (DFS) is generated when the system goes in debug operation.

In RAMP2, RAMP2A, or DSBOTH operation, when the Lock Update bit in the Control B register is set by writing CTRLBSET.LUPD=1 and the ramp index or counter direction changes, a non-recoverable Update Fault State and the respective interrupt (UFS) are generated.

36.6.3.7. Time-Stamp Capture

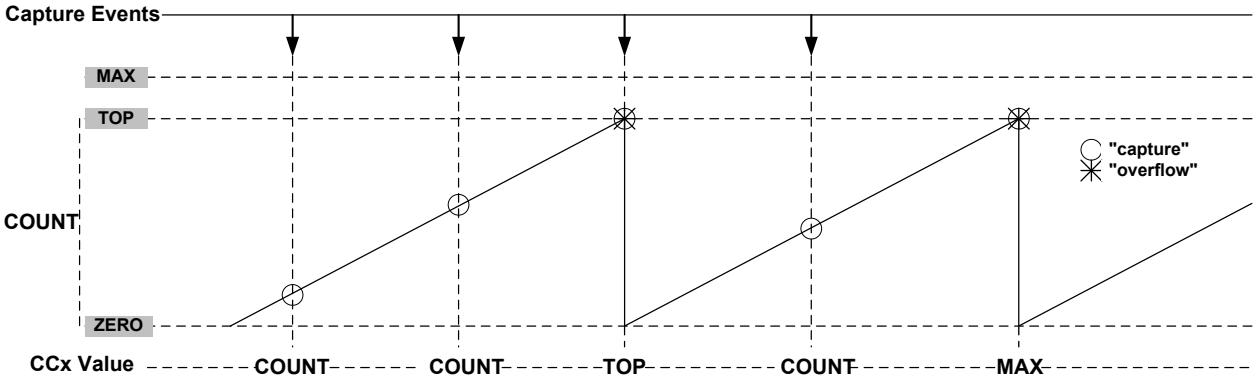
This feature is enabled when the Capture Time Stamp (STAMP) Event Action in Event Control register (EVCTRL.EVACT) is selected. The counter TOP value must be smaller than MAX.

When a capture event is detected, the COUNT value is copied into the corresponding Channel x Compare/Capture Value (CCx) register. In case of an overflow, the MAX value is copied into the corresponding CCx register.

When a valid captured value is present in the capture channel register, the corresponding Capture Channel x Interrupt Flag (INTFLAG.MCx) is set.

The timer/counter can detect capture overflow of the input capture channels: When a new capture event is detected while the Capture Channel interrupt flag (INTFLAG.MCx) is still set, the new time-stamp will not be stored and INTFLAG.ERR will be set.

Figure 36-31. Time-Stamp



36.6.3.8. Waveform Extension

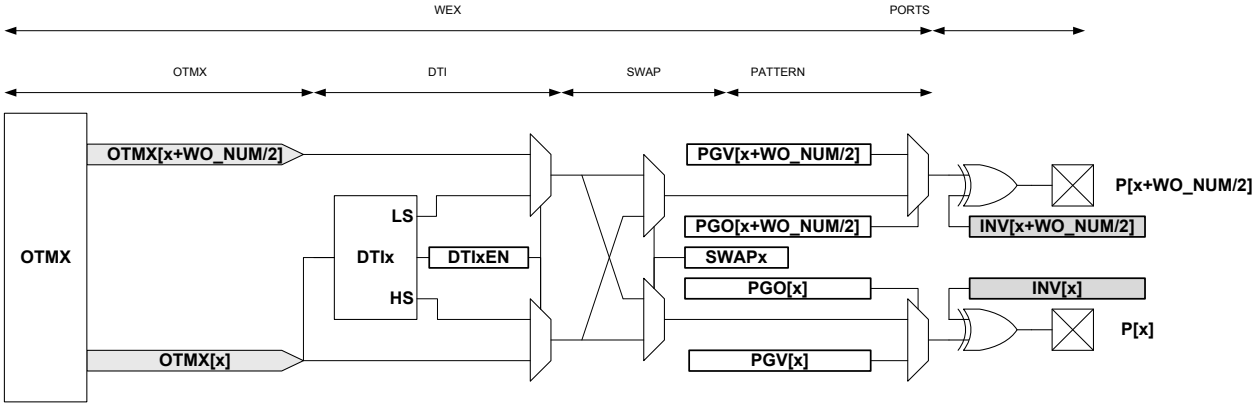
Figure 36-32 Waveform Extension Stage Details on page 783 shows a schematic diagram of actions of the four optional units that follow the recoverable fault stage on a port pin pair: Output Matrix (OTMX), Dead-Time Insertion (DTI), SWAP and Pattern Generation. The DTI and SWAP units can be seen as a four port pair slices:

- Slice 0 DTI0 / SWAP0 acting on port pins (WO[0], WO[WO_NUM/2 +0])
- Slice 1 DTI1 / SWAP1 acting on port pins (WO[1], WO[WO_NUM/2 +1])

And more generally:

- Slice n DTIx / SWAPx acting on port pins (WO[x], WO[WO_NUM/2 +x])

Figure 36-32. Waveform Extension Stage Details



The output matrix (OTMX) unit distributes compare channels, according to the selectable configurations in Table 36-5 Output Matrix Channel Pin Routing Configuration on page 783.

Table 36-5. Output Matrix Channel Pin Routing Configuration

Value	OTMX[x]							
0x0	CC3	CC2	CC1	CC0	CC3	CC2	CC1	CC0
0x1	CC1	CC0	CC1	CC0	CC1	CC0	CC1	CC0
0x2	CC0	CC0	CC0	CC0	CC0	CC0	CC0	CC0
0x3	CC1	CC1	CC1	CC1	CC1	CC1	CC1	CC0

Notes on Table 36-5 Output Matrix Channel Pin Routing Configuration on page 783:

- Configuration 0x0 is the default configuration. The channel location is the default one, and channels are distributed on outputs modulo the number of channels. Channel 0 is routed to the Output matrix output OTMX[0], and Channel 1 to OTMX[1]. If there are more outputs than channels, then channel 0 is duplicated to the Output matrix output OTMX[CC_NUM], channel 1 to OTMX[CC_NUM+1] and so on.
- Configuration 0x1 distributes the channels on output modulo half the number of channels. This assigns twice the number of output locations to the lower channels than the default configuration. This can be used, for example, to control the four transistors of a full bridge using only two compare channels.
  Using pattern generation, some of these four outputs can be overwritten by a constant level, enabling flexible drive of a full bridge in all quadrant configurations.
- Configuration 0x2 distributes compare channel 0 (CC0) to all port pins. With pattern generation, this configuration can control a stepper motor.
- Configuration 0x3 distributes the compare channel CC0 to the first output, and the channel CC1 to all other outputs. Together with pattern generation and the fault extension, this configuration can control up to seven LED strings, with a boost stage.

Table 36-6. Example: four compare channels on four outputs

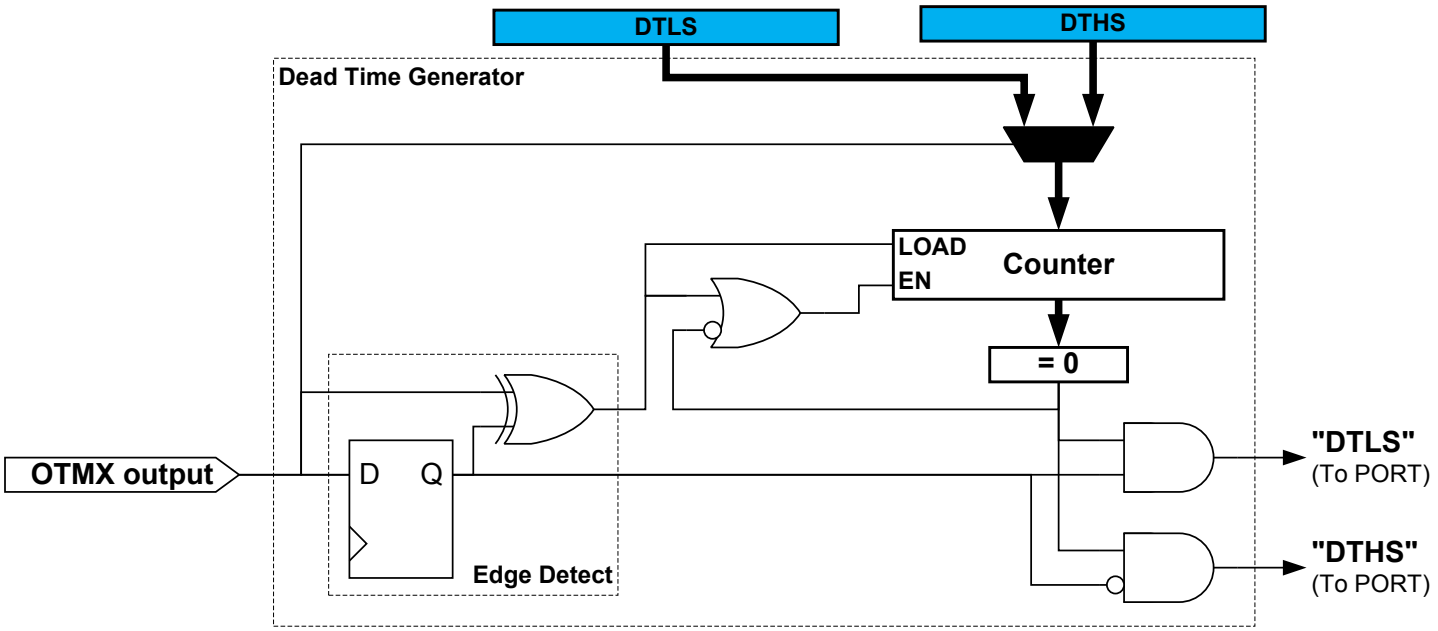
Value	OTMX[3]	OTMX[2]	OTMX[1]	OTMX[0]
0x0	CC3	CC2	CC1	CC0
0x1	CC1	CC0	CC1	CC0

Value	OTMX[3]	OTMX[2]	OTMX[1]	OTMX[0]
0x2	CC0	CC0	CC0	CC0
0x3	CC1	CC1	CC1	CC0

The dead-time insertion (DTI) unit generates OFF time with the non-inverted low side (LS) and inverted high side (HS) of the wave generator output forced at low level. This OFF time is called dead time. Dead-time insertion ensures that the LS and HS will never switch simultaneously.

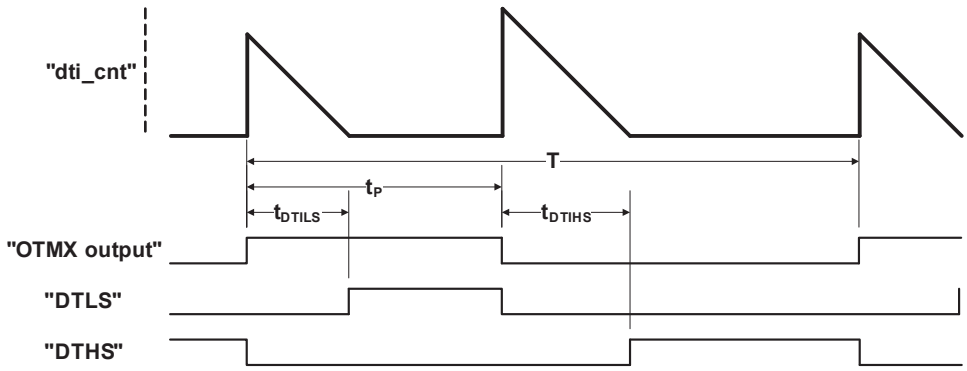
The DTI stage consists of four equal dead-time insertion generators; one for each of the first four compare channels. Figure 36-33 Dead-Time Generator Block Diagram on page 784 shows the block diagram of one DTI generator. The four channels have a common register which controls the dead time, which is independent of high side and low side setting.

Figure 36-33. Dead-Time Generator Block Diagram



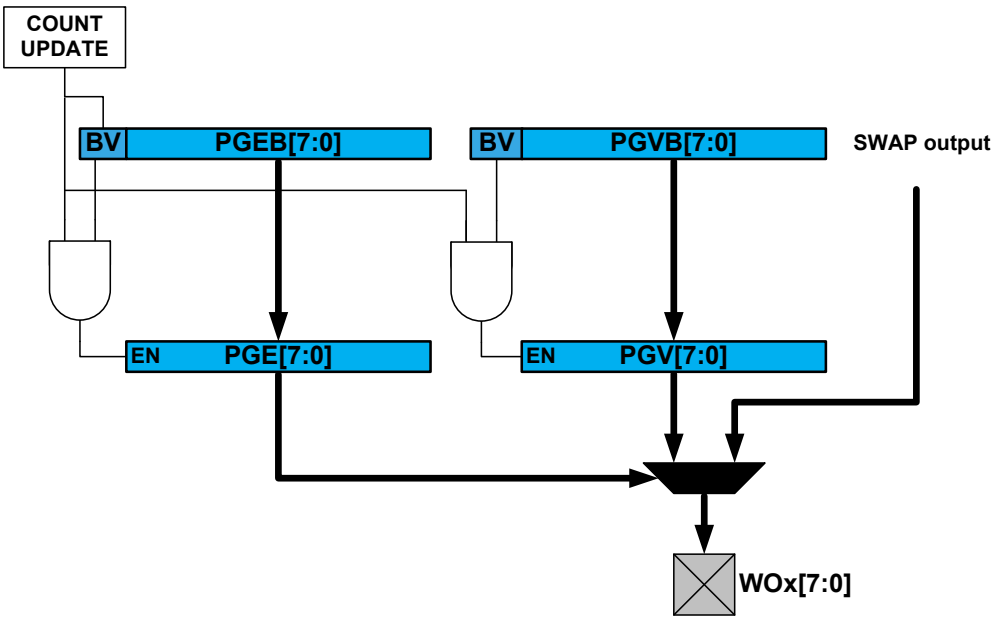
As shown in Figure 36-34 Dead-Time Generator Timing Diagram on page 784, the 8-bit dead-time counter is decremented by one for each peripheral clock cycle until it reaches zero. A non-zero counter value will force both the low side and high side outputs into their OFF state. When the output matrix (OTMX) output changes, the dead-time counter is reloaded according to the edge of the input. When the output changes from low to high (positive edge) it initiates a counter reload of the DTLS register. When the output changes from high to low (negative edge) it reloads the DTHS register.

Figure 36-34. Dead-Time Generator Timing Diagram



The pattern generator unit produces a synchronized bit pattern across the port pins it is connected to. The pattern generation features are primarily intended for handling the commutation sequence in brushless DC motors (BLDC), stepper motors, and full bridge control. See also Figure 36-35 Pattern Generator Block Diagram on page 785.

Figure 36-35. Pattern Generator Block Diagram



As with other double-buffered timer/counter registers, the register update is synchronized to the UPDATE condition set by the timer/counter waveform generation operation. If synchronization is not required by the application, the software can simply access directly the PATT.PGE, PATT.PGV bits registers.

36.6.4. DMA, Interrupts, and Events

Table 36-7. Module Requests for TCC

Condition	Interrupt request	Event output	Event input	DMA request	DMA request is cleared
Overflow / Underflow	Yes	Yes		Yes(1)	On DMA acknowledge
Channel Compare Match or Capture	Yes	Yes	Yes(2)	Yes(3)	For circular buffering: on DMA acknowledge For capture channel: when CCx register is read
Retrigger	Yes	Yes			
Count	Yes	Yes			
Capture Overflow Error	Yes				
Debug Fault State	Yes				
Recoverable Faults	Yes				
Non-Recoverable Faults	Yes				

Condition	Interrupt request	Event output	Event input	DMA request	DMA request is cleared
TCCx Event 0 input			Yes[4]		
TCCx Event 1 input			Yes[5]		

Notes:

1. DMA request set on overflow, underflow or re-trigger conditions.
2. Can perform capture or generate recoverable fault on an event input.
3. In capture or circular modes.
4. On event input, either action can be executed:
   - re-trigger counter
   - control counter direction
   - stop the counter
   - decrement the counter
   - perform period and pulse width capture
   - generate non-recoverable fault
5. On event input, either action can be executed:
   - re-trigger counter
   - increment or decrement counter depending on direction
   - start the counter
   - increment or decrement counter based on direction
   - increment counter regardless of direction
   - generate non-recoverable fault

36.6.4.1. DMA Operation

The TCC can generate the following DMA requests:

Counter overflow (OVF)	If the Ones-shot Trigger mode in the control A register (CTRLA.DMAOS) is written to '0', the TCC generates a DMA request on each cycle when an update condition (overflow, underflow or re-trigger) is detected. When an update condition (overflow, underflow or re-trigger) is detected while CTRLA.DMAOS=1, the TCC generates a DMA trigger on the cycle following the DMA One-Shot Command written to the Control B register (CTRLBSET.CMD=DMAOS). In both cases, the request is cleared by hardware on DMA acknowledge.
Channel Match (MCx)	A DMA request is set only on a compare match if CTRLA.DMAOS=0. The request is cleared by hardware on DMA acknowledge. When CTRLA.DMAOS=1, the DMA requests are not generated.
Channel Capture (MCx)	For a capture channel, the request is set when valid data is present in the CCx register, and cleared once the CCx register is read. In this operation mode, the CTRLA.DMAOS bit value is ignored.

DMA Operation with Circular Buffer

When circular buffer operation is enabled, the buffer registers must be written in a correct order and synchronized to the update times of the timer. The DMA triggers of the TCC provide a way to ensure a safe and correct update of circular buffers.

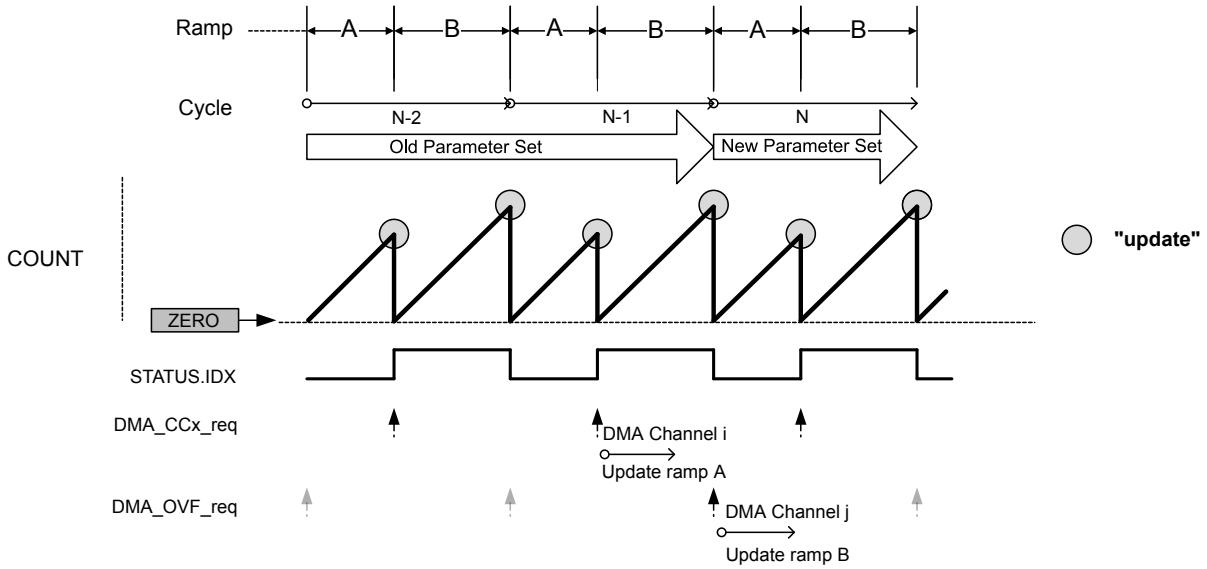
Note: Circular buffer are intended to be used with RAMP2, RAMP2A and DSBOTH operation only.

DMA Operation with Circular Buffer in RAMP and RAMP2A Mode
When a CCx channel is selected as a circular buffer, the related DMA request is not set on a compare match detection, but on start of ramp B.

If at least one circular buffer is enabled, the DMA overflow request is conditioned to the start of ramp A with an effective DMA transfer on previous ramp B (DMA acknowledge).

The update of all circular buffer values for ramp A can be done through a DMA channel triggered on a MC trigger. The update of all circular buffer values for ramp B, can be done through a second DMA channel triggered by the overflow DMA request.

Figure 36-36. DMA Triggers in RAMP and RAMP2 Operation Mode and Circular Buffer Enabled

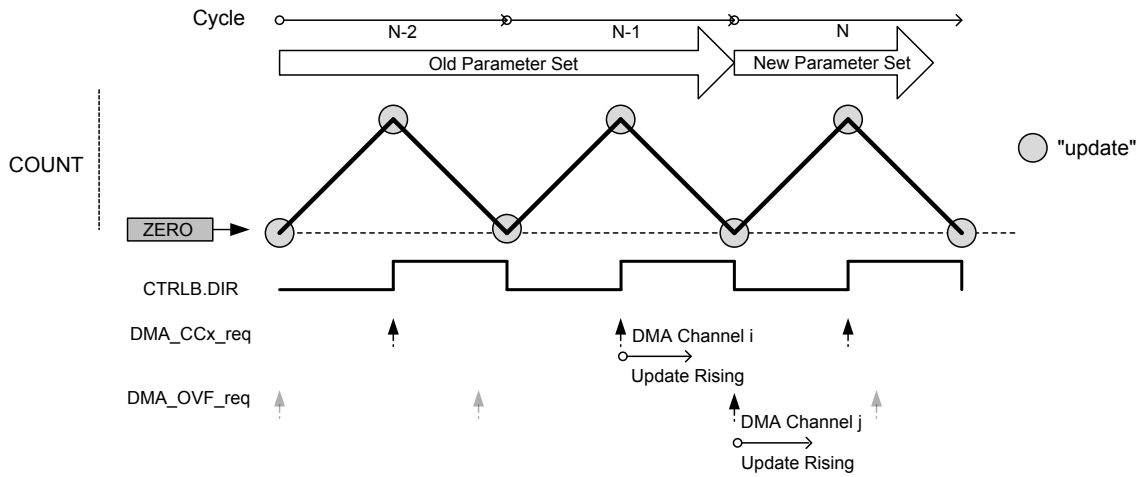


DMA Operation with Circular Buffer in DSBOTH Mode
When a CC channel is selected as a circular buffer, the related DMA request is not set on a compare match detection, but on start of down-counting phase.

If at least one circular buffer is enabled, the DMA overflow request is conditioned to the start of up-counting phase with an effective DMA transfer on previous down-counting phase (DMA acknowledge).

When up-counting, all circular buffer values can be updated through a DMA channel triggered by MC trigger. When down-counting, all circular buffer values can be updated through a second DMA channel, triggered by the OVF DMA request.

Figure 36-37. DMA Triggers in DSBOTH Operation Mode and Circular Buffer Enabled



36.6.4.2. Interrupts

The TCC has the following interrupt sources:

- Overflow/Underflow (OVF)
- Retrigger (TRG)
- Count (CNT) - refer also to description of EVCTRL.CNTSEL.
- Capture Overflow Error (ERR)
- Debug Fault State (DFS)
- Recoverable Faults (FAULTn)
- Non-recoverable Faults (FAULTx)
- Compare Match or Capture Channels (MCx)

These interrupts are asynchronous wake-up sources. See Sleep Mode Entry and Exit Table in PM/Sleep Mode Controller section for details.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register. An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the TCC is reset. See INTFLAG on page 820 for details on how to clear interrupt flags. The TCC has one common interrupt request line for all the interrupt sources. The user must read the INTFLAG register to determine which interrupt condition is present.

Note: Interrupts must be globally enabled for interrupt requests to be generated. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

Nested Vector Interrupt Controller on page 51

36.6.4.3. Events

The TCC can generate the following output events:

- Overflow/Underflow (OVF)
- Trigger (TRG)

- Counter (CNT) For further details, refer to EVCTRL.CNTSEL description.
- Compare Match or Capture on compare/capture channels: MCx

Writing a '1' ('0') to an Event Output bit in the Event Control Register (EVCTRL.xxEO) enables (disables) the corresponding output event. Refer also to *EVSYS – Event System*.

The TCC can take the following actions on a channel input event (MCx):

- Capture event
- Generate a recoverable or non-recoverable fault

The TCC can take the following actions on counter Event 1 (TCCx EV1):

- Counter re-trigger
- Counter direction control
- Stop the counter
- Decrement the counter on event
- Period and pulse width capture
- Non-recoverable fault

The TCC can take the following actions on counter Event 0 (TCCx EV0):

- Counter re-trigger
- Count on event (increment or decrement, depending on counter direction)
- Counter start - start counting on the event rising edge. Further events will not restart the counter; the counter will keep on counting using prescaled GCLK_TCCx, until it reaches TOP or ZERO, depending on the direction.
- Counter increment on event. This will increment the counter, irrespective of the counter direction.
- Count during active state of an asynchronous event (increment or decrement, depending on counter direction). In this case, the counter will be incremented or decremented on each cycle of the prescaled clock, as long as the event is active.
- Non-recoverable fault

The counter Event Actions are available in the Event Control registers (EVCTRL.EVACT0 and EVCTRL.EVACT1). For further details, refer to EVCTRL.

Writing a '1' ('0') to an Event Input bit in the Event Control register (EVCTRL.MCEIx or EVCTRL.TCEIx) enables (disables) the corresponding action on input event.

Note: When several events are connected to the TCC, the enabled action will apply for each of the incoming events. Refer to *EVSYS – Event System* for details on how to configure the event system.

Related Links

EVSYS – Event System on page 536

36.6.5. Sleep Mode Operation

The TCC can be configured to operate in any sleep mode. To be able to run in standby the RUNSTDBY bit in the Control A register (CTRLA.RUNSTDBY) must be '1'. The MODULE can in any sleep mode wake up the device using interrupts or perform actions through the Event System.

36.6.6. Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read. An exception is the Channel Enable bit in the Peripheral Channel Control registers (PCHCTRLm.CHEN). When changing this bit, the bit value must be read-back to ensure the synchronization is complete and to assert glitch free internal operation. Note that changing the bit value under ongoing synchronization will *not* generate an error.

The following bits are synchronized when written:

- Software Reset and Enable bits in Control A register (CTRLA.SWRST and CTRLA.ENABLE)

The following registers are synchronized when written:

- Control B Clear and Control B Set registers (CTRLBCLR and CTRLBSET)
- Status register (STATUS)
- Pattern and Pattern Buffer registers (PATT and PATTB)
- Waveform register (WAVE)
- Count Value register (COUNT)
- Period Value and Period Buffer Value registers (PER and PERB)
- Compare/Capture Channel x and Channel x Compare/Capture Buffer Value registers (CCx and CCBx)

The following registers are synchronized when read:

- Control B Clear and Control B Set registers (CTRLBCLR and CTRLBSET)
- Count Value register (COUNT)
- Pattern and Pattern Buffer registers (PATT and PATTB)
- Waveform register (WAVE)
- Period Value and Period Buffer Value registers (PER and PERB)
- Compare/Capture Channel x and Channel x Compare/Capture Buffer Value registers (CCx and CCBx)

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

Required read-synchronization is denoted by the "Read-Synchronized" property in the register description.

Related Links

Register Synchronization on page 127

36.7. Register Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0		RESOLUTION[1:0]					ENABLE	SWRST
0x01		15:8		ALOCK	PRESCSYNC[1:0]		RUNSTDBY	PRESCALER[2:0]		
0x02		23:16								
0x03		31:24					CPTEN3	CPTEN2	CPTEN1	CPTEN0
0x04	CTRLBCLR	7:0	CMD[2:0]			IDXCMD[1:0]		ONESHOT	LUPD	DIR
0x05	CTRLBSET	15:8	CMD[2:0]			IDXCMD[1:0]		ONESHOT	LUPD	DIR
0x06	Reserved									
0x07	Reserved									
0x08	SYNCBUSY	7:0	PER	WAVE	PATT	COUNT	STATUS	CTRLB	ENABLE	SWRST
0x09		15:8					CC3	CC2	CC1	CC0
0x0A		23:16		CCB3	CCB2	CCB1	CCB0	PERB		PATTB
0x0B		31:24								

Offset	Name	Bit Pos.								
0x0C	FCTRLA	7:0	RESTART	BLANK[1:0]		QUAL	KEEP		SRC[1:0]	
0x0D		15:8	BLANKPRESC	CAPTURE[2:0]			CHSEL[1:0]		HALT[1:0]	
0x0E		23:16	BLANKVAL[7:0]							
0x0F		31:24					FILTERVAL[3:0]			
0x10	FCTRLB	7:0	RESTART	BLANK[1:0]		QUAL	KEEP		SRC[1:0]	
0x11		15:8	BLANKPRESC	CAPTURE[2:0]			CHSEL[1:0]		HALT[1:0]	
0x12		23:16	BLANKVAL[7:0]							
0x13		31:24					FILTERVAL[3:0]			
0x14	WEXCTRL	7:0							OTMX[1:0]	
0x15		15:8					DTIEN3	DTIEN2	DTIEN1	DTIEN0
0x16		23:16	DTLS[7:0]							
0x17		31:24	DTHS[7:0]							
0x18	DRVCTRL	7:0	NRE7	NRE6	NRE5	NRE4	NRE3	NRE2	NRE1	NRE0
0x19		15:8	NRV7	NRV6	NRV5	NRV4	NRV3	NRV2	NRV1	NRV0
0x1A		23:16	INVEN7	INVEN6	INVEN5	INVEN4	INVEN3	INVEN2	INVEN1	INVEN0
0x1B		31:24	FILTERVAL1[3:0]				FILTERVAL0[3:0]			
0x1C	Reserved									
0x1D	Reserved									
0x1E	DBGCTRL	7:0						FDDBD		DBGRUN
0x1F	Reserved									
0x20	EVCTRL	7:0	CNTSEL[1:0]		EVACT1[2:0]			EVACT0[2:0]		
0x21		15:8	TCEI1	TCEI0	TCEINV1	TCEINV0		CNTEO	TRGEO	OVFEO
0x22		23:16					MCEI3	MCEI2	MCEI1	MCEI0
0x23		31:24					MCEO3	MCEO2	MCEO1	MCEO0
0x24	INTENCLR	7:0					ERR	CNT	TRG	OVF
0x25		15:8	FAULT1	FAULT0	FAULTB	FAULTA	DFS			
0x26		23:16					MC3	MC2	MC1	MC0
0x27	Reserved									
0x28	INTENSET	7:0					ERR	CNT	TRG	OVF
0x29		15:8	FAULT1	FAULT0	FAULTB	FAULTA	DFS			
0x2A		23:16					MC3	MC2	MC1	MC0
0x2B	Reserved									
0x2C	INTFLAG	7:0					ERR	CNT	TRG	OVF
0x2D		15:8	FAULT1	FAULT0	FAULTB	FAULTA	DFS			
0x2E		23:16					MC3	MC2	MC1	MC0
0x2F	Reserved									
0x30	STATUS	7:0	PERBV		PATTBV		DFS		IDX	STOP
0x31		15:8	FAULT1	FAULT0	FAULTB	FAULTA	FAULT1IN	FAULT0IN	FAULTBIN	FAULTAIN
0x32		23:16					CCBV3	CCBV2	CCBV1	CCBV0
0x33		31:24					CMP3	CMP2	CMP1	CMP0
0x34	COUNT	7:0	COUNT[7:0]							
0x35		15:8	COUNT[15:8]							
0x36		23:16	COUNT[23:16]							
0x37		31:24	COUNT[31:24]							

Offset	Name	Bit Pos.								
0x38	PATT	7:0	PGE7	PGE6	PGE5	PGE4	PGE3	PGE2	PGE1	PGE0
0x39		15:8	PGV7	PGV6	PGV5	PGV4	PGV3	PGV2	PGV1	PGV0
0x3A	Reserved									
0x3B	Reserved									
0x3C	WAVE	7:0	CIPEREN		RAMP[1:0]			WAVEGEN[2:0]		
0x3D		15:8					CICCEN3	CICCEN2	CICCEN1	CICCEN0
0x3E		23:16					POL3	POL2	POL1	POL0
0x3F		31:24					SWAP3	SWAP2	SWAP1	SWAP0
0x40	PER	7:0	PER[7:0]							
0x41		15:8	PER[15:8]							
0x42		23:16	PER[23:16]							
0x43		31:24	PER[31:24]							
0x44	CC0	7:0	CC0[7:0]							
0x45		15:8	CC0[15:8]							
0x46		23:16	CC0[23:16]							
0x47		31:24	CC0[31:24]							
0x48	CC1	7:0	CC1[7:0]							
0x49		15:8	CC1[15:8]							
0x4A		23:16	CC1[23:16]							
0x4B		31:24	CC1[31:24]							
0x4C	CC2	7:0	CC2[7:0]							
0x4D		15:8	CC2[15:8]							
0x4E		23:16	CC2[23:16]							
0x4F		31:24	CC2[31:24]							
0x50	CC3	7:0	CC3[7:0]							
0x51		15:8	CC3[15:8]							
0x52		23:16	CC3[23:16]							
0x53		31:24	CC3[31:24]							
0x54 ... 0x63	Reserved									
0x64	PATTB	7:0	PGEB[7:0]							
0x65		15:8	PGVB[7:0]							
0x66	Reserved									
0x67	Reserved									
0x68	Reserved									
0x69	Reserved									
0x6A	Reserved									
0x6B	Reserved									
0x6C	PERB	7:0	PERB[7:0]							
0x6D		15:8	PERB[15:8]							
0x6E		23:16	PERB[23:16]							
0x6F		31:24	PERB[31:24]							
0x70	CCBUF0	7:0	CCB0[7:0]							
0x71		15:8	CCB0[15:8]							
0x72		23:16	CCB0[23:16]							
0x73		31:24	CCB0[31:24]							

Offset	Name	Bit Pos.								
0x74	CCBUF1	7:0	CCB1[7:0]							
0x75		15:8	CCB1[15:8]							
0x76		23:16	CCB1[23:16]							
0x77		31:24	CCB1[31:24]							
0x78	CCBUF2	7:0	CCB2[7:0]							
0x79		15:8	CCB2[15:8]							
0x7A		23:16	CCB2[23:16]							
0x7B		31:24	CCB2[31:24]							
0x7C	CCBUF3	7:0	CCB3[7:0]							
0x7D		15:8	CCB3[15:8]							
0x7E		23:16	CCB3[23:16]							
0x7F		31:24	CCB3[31:24]							

36.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

36.8.1. Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized (ENABLE, SWRST)

Bit	31	30	29	28	27	26	25	24
					CPTEN3	CPTEN2	CPTEN1	CPTEN0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		ALOCK	PRESCYNC[1:0]		RUNSTDBY	PRESCALER[2:0]		
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		RESOLUTION[1:0]					ENABLE	SWRST
Access		R/W	R/W				R/W	R/W
Reset		0	0				0	0

Bit 14 – ALOCK: Auto Lock
This bit is not synchronized.

Value	Description
0	The Lock Update bit in the Control B register (CTRLB.LUPD) is not affected by overflow/underflow, and re-trigger events
1	CTRLB.LUPD is set to '1' on each overflow/underflow or re-trigger event.

Bits 13:12 – PRESCYNC[1:0]: Prescaler and Counter Synchronization
These bits select if on re-trigger event, the Counter is cleared or reloaded on either the next GCLK_TCCx clock, or on the next prescaled GCLK_TCCx clock. It is also possible to reset the prescaler on re-trigger event.

These bits are not synchronized.

Value	Name	Description	
		Counter Reloaded	Prescaler
0x0	GCLK	Reload or reset Counter on next GCLK	-
0x1	PRESC	Reload or reset Counter on next prescaler clock	-

Value	Name	Description	
		Counter Reloaded	Prescaler
0x2	RESYNC	Reload or reset Counter on next GCLK	Reset prescaler counter
0x3	Reserved		

Bit 11 – RUNSTDBY: Run in Standby
This bit is used to keep the TCC running in standby mode.

This bit is not synchronized.

Value	Description
0	The TCC is halted in standby.
1	The TCC continues to run in standby.

Bits 10:8 – PRESCALER[2:0]: Prescaler
These bits select the Counter prescaler factor.

These bits are not synchronized.

Value	Name	Description
0x0	DIV1	Prescaler: GCLK_TCC
0x1	DIV2	Prescaler: GCLK_TCC/2
0x2	DIV4	Prescaler: GCLK_TCC/4
0x3	DIV8	Prescaler: GCLK_TCC/8
0x4	DIV16	Prescaler: GCLK_TCC/16
0x5	DIV64	Prescaler: GCLK_TCC/64
0x6	DIV256	Prescaler: GCLK_TCC/256
0x7	DIV1024	Prescaler: GCLK_TCC/1024

Bits 6:5 – RESOLUTION[1:0]: Dithering Resolution
These bits increase the TCC resolution by enabling the dithering options.

These bits are not synchronized.

Table 36-8. Dithering

Value	Name	Description
0x0	NONE	The dithering is disabled.
0x1	DITH4	Dithering is done every 16 PWM frames. PER[3:0] and CCx[3:0] contain dithering pattern selection.

Value	Name	Description
0x2	DITH5	Dithering is done every 32 PWM frames. PER[4:0] and CCx[4:0] contain dithering pattern selection.
0x3	DITH6	Dithering is done every 64 PWM frames. PER[5:0] and CCx[5:0] contain dithering pattern selection.

Bit 1 – ENABLE: Enable
Due to synchronization there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the ENABLE bit in the SYNCBUSY register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

Value	Description
0	The peripheral is disabled.
1	The peripheral is enabled.

Bit 0 – SWRST: Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the TCC (except DBGCTRL) to their initial state, and the TCC will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence; all other writes in the same write-operation will be discarded.

Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

Bits 27,26,25,24 – CPTENx: Capture Channel x Enable
These bits are used to select the capture or compare operation on channel x.

Writing a '1' to CPTENx enables capture on channel x.

Writing a '0' to CPTENx disables capture on channel x.

36.8.2. Control B Clear

This register allows the user to change this register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Set (CTRLBSET) register.

Name: CTRLBCLR
Offset: 0x04
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Bit	7	6	5	4	3	2	1	0
	CMD[2:0]			IDXCMD[1:0]		ONESHOT	LUPD	DIR
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:5 – CMD[2:0]: TCC Command
These bits can be used for software control of re-triggering and stop commands of the TCC. When a command has been executed, the CMD bit field will read back zero. The commands are executed on the next prescaled GCLK_TCC clock cycle.

Writing zero to this bit group has no effect.

Writing a '1' to any of these bits will clear the pending command.

Value	Name	Description
0x0	NONE	No action
0x1	RETRIGGER	Clear start, restart or retrigger
0x2	STOP	Force stop
0x3	UPDATE	Force update of double buffered registers
0x4	READSYNC	Force COUNT read synchronization

Bits 4:3 – IDXCMD[1:0]: Ramp Index Command
These bits can be used to force cycle A and cycle B changes in RAMP2 and RAMP2A operation. On timer/counter update condition, the command is executed, the IDX flag in STATUS register is updated and the IDXCMD command is cleared.

Writing zero to these bits has no effect.

Writing a '1' to any of these bits will clear the pending command.

Value	Name	Description
0x0	DISABLE	DISABLE Command disabled: IDX toggles between cycles A and B
0x1	SET	Set IDX: cycle B will be forced in the next cycle
0x2	CLEAR	Clear IDX: cycle A will be forced in next cycle
0x3	HOLD	Hold IDX: the next cycle will be the same as the current cycle.

Bit 2 – ONESHOT: One-Shot
This bit controls one-shot operation of the TCC. When one-shot operation is enabled, the TCC will stop counting on the next overflow/underflow condition or on a stop command.

Writing a '0' to this bit has no effect

Writing a '1' to this bit will disable the one-shot operation.

Value	Description
0	The TCC will update the counter value on overflow/underflow condition and continue operation.
1	The TCC will stop counting on the next underflow/overflow condition.

Bit 1 – LUPD: Lock Update
This bit controls the update operation of the TCC buffered registers. When this bit is set, no update of the buffered registers is performed, even though an UPDATE condition has occurred. Locking the update ensures that all buffers registers are valid before an update is performed.

This bit has no effect when input capture operation is enabled.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will enable updating.

Value	Description
0	The CCBx, PERB, PGVB, PGOB, and SWAPBx buffer registers values *are* copied into the corresponding CCx, PER, PGV, PGO and SWAPx registers
1	The CCBx, PERB, PGVB, PGOB, and SWAPBx buffer registers values are *not* copied into the corresponding CCx, PER, PGV, PGO and SWAPx registers.

Bit 0 – DIR: Counter Direction
This bit is used to change the direction of the counter.

Writing a '0' to this bit has no effect

Writing a '1' to this bit will clear the bit and make the counter count up.

Value	Description
0	The timer/counter is counting up (incrementing).
1	The timer/counter is counting down (decrementing).

36.8.3. Control B Set

This register allows the user to change this register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Set (CTRLBCLR) register.

Name: CTRLBSET
Offset: 0x05
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Bit	7	6	5	4	3	2	1	0
	CMD[2:0]			IDXCMD[1:0]		ONESHOT	LUPD	DIR
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:5 – CMD[2:0]: TCC Command
These bits can be used for software control of re-triggering and stop commands of the TCC. When a command has been executed, the CMD bit field will be read back as zero. The commands are executed on the next prescaled GCLK_TCC clock cycle.

Writing zero to this bit group has no effect

Writing a valid value to this bit group will set the associated command.

Value	Name	Description
0x0	NONE	No action
0x1	RETRIGGER	Force start, restart or retrigger
0x2	STOP	Force stop
0x3	UPDATE	Force update of double buffered registers
0x4	READSYNC	Force a read synchronization of COUNT

Bits 4:3 – IDXCMD[1:0]: Ramp Index Command
These bits can be used to force cycle A and cycle B changes in RAMP2 and RAMP2A operation. On timer/counter update condition, the command is executed, the IDX flag in STATUS register is updated and the IDXCMD command is cleared.

Writing a zero to these bits has no effect.

Writing a valid value to these bits will set a command.

Value	Name	Description
0x0	DISABLE	Command disabled: IDX toggles between cycles A and B
0x1	SET	Set IDX: cycle B will be forced in the next cycle
0x2	CLEAR	Clear IDX: cycle A will be forced in next cycle
0x3	HOLD	Hold IDX: the next cycle will be the same as the current cycle.

Bit 2 – ONESHOT: One-Shot
This bit controls one-shot operation of the TCC. When in one-shot operation, the TCC will stop counting on the next overflow/underflow condition or a stop command.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will enable the one-shot operation.

Value	Description
0	The TCC will count continuously.
1	The TCC will stop counting on the next underflow/overflow condition.

Bit 1 – LUPD: Lock Update
This bit controls the update operation of the TCC buffered registers. When this bit is set, no update of the buffered registers is performed, even though an UPDATE condition has occurred. Locking the update can be used to ensure that all buffer registers are loaded with the desired values, before an update is performed. After all the buffer registers are loaded correctly, the buffered registers can be unlocked.

This bit has no effect when input capture operation is enabled.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will lock updating.

Value	Description
0	The CCBx, PERB, PGVB, PGOB, and SWAPBx buffer registers values *are* copied into the corresponding CCx, PER, PGV, PGO and SWAPx registers.
1	The CCBx, PERB, PGVB, PGOB, and SWAPBx buffer registers values are *not* copied into CCx, PER, PGV, PGO and SWAPx registers.

Bit 0 – DIR: Counter Direction
This bit is used to change the direction of the counter.

Writing a '0' to this bit has no effect

Writing a '1' to this bit will clear the bit and make the counter count up.

Value	Description
0	The timer/counter is counting up (incrementing).
1	The timer/counter is counting down (decrementing).

36.8.4. Synchronization Busy

Name: SYNCBUSY
Offset: 0x08
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
		CCB3	CCB2	CCB1	CCB0	PERB		PATTB
Access		R	R	R	R	R		R
Reset		0	0	0	0	0		0

Bit	15	14	13	12	11	10	9	8
					CC3	CC2	CC1	CC0
Access					R	R	R	R
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PER	WAVE	PATT	COUNT	STATUS	CTRLB	ENABLE	SWRST
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 18 – PERB: PERB Synchronization Busy
This bit is cleared when the synchronization of PERB register between the clock domains is complete.

This bit is set when the synchronization of PERB register between clock domains is started.

Bit 16 – PATTB: PATTB Synchronization Busy
This bit is cleared when the synchronization of PATTERNB register between the clock domains is complete.

This bit is set when the synchronization of PATTERNB register between clock domains is started.

Bit 7 – PER: PER Synchronization Busy
This bit is cleared when the synchronization of PER register between the clock domains is complete.

This bit is set when the synchronization of PER register between clock domains is started.

Bit 6 – WAVE: WAVE Synchronization Busy
This bit is cleared when the synchronization of WAVE register between the clock domains is complete.

This bit is set when the synchronization of WAVE register between clock domains is started.

Bit 5 – PATT: PATT Synchronization Busy
This bit is cleared when the synchronization of PATTERN register between the clock domains is complete.

This bit is set when the synchronization of PATTERN register between clock domains is started.

Bit 4 – COUNT: COUNT Synchronization Busy
This bit is cleared when the synchronization of COUNT register between the clock domains is complete.

This bit is set when the synchronization of COUNT register between clock domains is started.

Bit 3 – STATUS: STATUS Synchronization Busy
This bit is cleared when the synchronization of STATUS register between the clock domains is complete.

This bit is set when the synchronization of STATUS register between clock domains is started.

Bit 2 – CTRLB: CTRLB Synchronization Busy
This bit is cleared when the synchronization of CTRLB register between the clock domains is complete.

This bit is set when the synchronization of CTRLB register between clock domains is started.

Bit 1 – ENABLE: ENABLE Synchronization Busy
This bit is cleared when the synchronization of ENABLE bit between the clock domains is complete.

This bit is set when the synchronization of ENABLE bit between clock domains is started.

Bit 0 – SWRST: SWRST Synchronization Busy
This bit is cleared when the synchronization of SWRST bit between the clock domains is complete.

This bit is set when the synchronization of SWRST bit between clock domains is started.

Bits 22,21,20,19 – CCBx: Compare/Capture Buffer Channel x Synchronization Busy
This bit is cleared when the synchronization of Compare/Capture Buffer Channel x register between the clock domains is complete.

This bit is set when the synchronization of Compare/Capture Buffer Channel x register between clock domains is started.

CCBx bit is available only for existing Compare/Capture Channels. For details on CC channels number, refer to each TCC feature list.

Bits 11,10,9,8 – CCx: Compare/Capture Channel x Synchronization Busy
This bit is cleared when the synchronization of Compare/Capture Channel x register between the clock domains is complete.

This bit is set when the synchronization of Compare/Capture Channel x register between clock domains is started.

CCx bit is available only for existing Compare/Capture Channels. For details on CC channels number, refer to each TCC feature list.

This bit is set when the synchronization of CCx register between clock domains is started.

36.8.5. Fault Control A and B

Name: FCTRLn
Offset: 0x0C+4*x (x = 0,1)
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
					FILTERVAL[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
	BLANKVAL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	BLANKPRESC	CAPTURE[2:0]			CHSEL[1:0]		HALT[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RESTART	BLANK[1:0]		QUAL	KEEP		SRC[1:0]	
Access	R/W	R/W	R/W	R/W	R/W		R/W	R/W
Reset	0	0	0	0	0		0	0

Bits 27:24 – FILTERVAL[3:0]: Recoverable Fault n Filter Value
These bits define the filter value applied on MCEx (x=0,1) event input line. The value must be set to zero when MCEx event is used as synchronous event.

Bits 23:16 – BLANKVAL[7:0]: Recoverable Fault n Blanking Value
These bits determine the duration of the blanking of the fault input source. Activation and edge selection of the blank filtering are done by the BLANK bits (FCTRLn.BLANK).

When enabled, the fault input source is internally disabled for BLANKVAL* prescaled GCLK_TCC periods after the detection of the waveform edge.

Bit 15 – BLANKPRESC: Recoverable Fault n Blanking Value Prescaler
This bit enables a factor 64 prescaler factor on used as base frequency of the BLANKVAL value.

Value	Description
0	Blank time is BLANKVAL* prescaled GCLK_TCC.
1	Blank time is BLANKVAL* 64 * prescaled GCLK_TCC.

Bits 14:12 – CAPTURE[2:0]: Recoverable Fault n Capture Action
These bits select the capture and Fault n interrupt/event conditions.

Table 36-9. Fault n Capture Action

Value	Name	Description
0x0	DISABLE	Capture on valid recoverable Fault n is disabled
0x1	CAPT	On rising edge of a valid recoverable Fault n, capture counter value on channel selected by CHSEL[1:0]. INTFLAG.FAULTn flag rises on each new captured value.
0x2	CAPTMIN	On rising edge of a valid recoverable Fault n, capture counter value on channel selected by CHSEL[1:0], if COUNT value is lower than the last stored capture value (CC). INTFLAG.FAULTn flag rises on each local minimum detection.
0x3	CAPTMAX	On rising edge of a valid recoverable Fault n, capture counter value on channel selected by CHSEL[1:0], if COUNT value is higher than the last stored capture value (CC). INTFLAG.FAULTn flag rises on each local maximun detection.
0x4	LOCMIN	On rising edge of a valid recoverable Fault n, capture counter value on channel selected by CHSEL[1:0]. INTFLAG.FAULTn flag rises on each local minimum value detection.
0x5	LOCMAX	On rising edge of a valid recoverable Fault n, capture counter value on channel selected by CHSEL[1:0]. INTFLAG.FAULTn flag rises on each local maximun detection.
0x6	DERIV0	On rising edge of a valid recoverable Fault n, capture counter value on channel selected by CHSEL[1:0]. INTFLAG.FAULTn flag rises on each local maximun or minimum detection.

Bits 11:10 – CHSEL[1:0]: Recoverable Fault n Capture Channel
These bits select the channel for capture operation triggered by recoverable Fault n.

Value	Name	Description
0x0	CC0	Capture value stored into CC0
0x1	CC1	Capture value stored into CC1
0x2	CC2	Capture value stored into CC2
0x3	CC3	Capture value stored into CC3

Bits 9:8 – HALT[1:0]: Recoverable Fault n Halt Operation
These bits select the halt action for recoverable Fault n.

Value	Name	Description
0x0	DISABLE	Halt action disabled
0x1	HW	Hardware halt action
0x2	SW	Software halt action
0x3	NR	Non-recoverable fault

Bit 7 – RESTART: Recoverable Fault n Restart
Setting this bit enables restart action for Fault n.

Value	Description
0	Fault n restart action is disabled.
1	Fault n restart action is enabled.

Bits 6:5 – BLANK[1:0]: Recoverable Fault n Blanking Operation
These bits, select the blanking start point for recoverable Fault n.

Value	Name	Description
0x0	START	Blanking applied from start of the Ramp period
0x1	RISE	Blanking applied from rising edge of the waveform output
0x2	FALL	Blanking applied from falling edge of the waveform output
0x3	BOTH	Blanking applied from each toggle of the waveform output

Bit 4 – QUAL: Recoverable Fault n Qualification
Setting this bit enables the recoverable Fault n input qualification.

Value	Description
0	The recoverable Fault n input is not disabled on CMPx value condition.
1	The recoverable Fault n input is disabled when output signal is at inactive level (CMPx == 0).

Bit 3 – KEEP: Recoverable Fault n Keep
Setting this bit enables the Fault n keep action.

Value	Description
0	The Fault n state is released as soon as the recoverable Fault n is released.
1	The Fault n state is released at the end of TCC cycle.

Bits 1:0 – SRC[1:0]: Recoverable Fault n Source
These bits select the TCC event input for recoverable Fault n.

Event system channel connected to MCEx event input, must be configured to route the event asynchronously, when used as a recoverable Fault n input.

Value	Name	Description
0x0	DISABLE	Fault input disabled
0x1	ENABLE	MCEx (x=0,1) event input

Value	Name	Description
0x2	INVERT	Inverted MCEx (x=0,1) event input
0x3	ALTFAULT	Alternate fault (A or B) state at the end of the previous period.

36.8.6. Waveform Extension Control

Name: WEXCTRL
Offset: 0x14
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
	DTHS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DTLS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
					DTIEN3	DTIEN2	DTIEN1	DTIEN0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
							OTMX[1:0]	
Access							R/W	R/W
Reset							0	0

Bits 31:24 – DTHS[7:0]: Dead-Time High Side Outputs Value
This register holds the number of GCLK_TCC clock cycles for the dead-time high side.

Bits 23:16 – DTLS[7:0]: Dead-time Low Side Outputs Value
This register holds the number of GCLK_TCC clock cycles for the dead-time low side.

Bits 1:0 – OTMX[1:0]: Output Matrix
These bits define the matrix routing of the TCC waveform generation outputs to the port pins, according to Table 36-5 Output Matrix Channel Pin Routing Configuration on page 783.

Bits 11,10,9,8 – DTIENx : Dead-time Insertion Generator x Enable
Setting any of these bits enables the dead-time insertion generator for the corresponding output matrix. This will override the output matrix [x] and [x+WO_NUM/2], with the low side and high side waveform respectively.

Value	Description
0	No dead-time insertion override.
1	Dead time insertion override on signal outputs[x] and [x+WO_NUM/2], from matrix outputs[x] signal.

36.8.7. Driver Control

Name: DRVCTRL
Offset: 0x18
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
	FILTERVAL1[3:0]				FILTERVAL0[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	INVEN7	INVEN6	INVEN5	INVEN4	INVEN3	INVEN2	INVEN1	INVEN0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	NRV7	NRV6	NRV5	NRV4	NRV3	NRV2	NRV1	NRV0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	NRE7	NRE6	NRE5	NRE4	NRE3	NRE2	NRE1	NRE0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:28 – FILTERVAL1[3:0]: Non-Recoverable Fault Input 1 Filter Value
These bits define the filter value applied on TCE1 event input line. When the TCE1 event input line is configured as a synchronous event, this value must be 0x0.

Bits 27:24 – FILTERVAL0[3:0]: Non-Recoverable Fault Input 0 Filter Value
These bits define the filter value applied on TCE0 event input line. When the TCE0 event input line is configured as a synchronous event, this value must be 0x0.

Bits 23,22,21,20,19,18,17,16 – INVENx: Waveform Output x Inversion
These bits are used to select inversion on the output of channel x.

Writing a '1' to INVENx inverts output from WO[x].

Writing a '0' to INVENx disables inversion of output from WO[x].

Bits 15,14,13,12,11,10,9,8 – NRVx: NRVx Non-Recoverable State x Output Value
These bits define the value of the enabled override outputs, under non-recoverable fault condition.

Bits 7,6,5,4,3,2,1,0 – NREx: Non-Recoverable State x Output Enable
These bits enable the override of individual outputs by NRVx value, under non-recoverable fault condition.

Value	Description
0	Non-recoverable fault tri-state the output.
1	Non-recoverable faults set the output to NRVx level.

36.8.8. Debug control

Name: DBGCTRL
Offset: 0x1E
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
						FDDBD		DBGRUN
Access						R/W		R/W
Reset						0		0

Bit 2 – FDDBD: Fault Detection on Debug Break Detection
This bit is not affected by software reset and should not be changed by software while the TCC is enabled.

By default this bit is zero, and the on-chip debug (OCD) fault protection is enabled. OCD break request from the OCD system will trigger non-recoverable fault. When this bit is set, OCD fault protection is disabled and OCD break request will not trigger a fault.

Value	Description
0	No faults are generated when TCC is halted in debug mode.
1	A non recoverable fault is generated and FAULTD flag is set when TCC is halted in debug mode.

Bit 0 – DBGRUN: Debug Running State
This bit is not affected by software reset and should not be changed by software while the TCC is enabled.

Value	Description
0	The TCC is halted when the device is halted in debug mode.
1	The TCC continues normal operation when the device is halted in debug mode.

36.8.9. Event Control

Name: EVCTRL
Offset: 0x20
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
					MCEO3	MCEO2	MCEO1	MCEO0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
					MCEI3	MCEI2	MCEI1	MCEI0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TCEI1	TCEI0	TCINV1	TCINV0		CNTEO	TRGEO	OVFEO
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bit	7	6	5	4	3	2	1	0
	CNTSEL[1:0]		EVACT1[2:0]			EVACT0[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 10 – CNTEO: Timer/Counter Event Output Enable
This bit is used to enable the counter cycle event. When enabled, an event will be generated on begin or end of counter cycle depending of CNTSEL[1:0] settings.

Value	Description
0	Counter cycle output event is disabled and will not be generated.
1	Counter cycle output event is enabled and will be generated depend of CNTSEL[1:0] value.

Bit 9 – TRGEO: Retrigger Event Output Enable
This bit is used to enable the counter retrigger event. When enabled, an event will be generated when the counter retriggers operation.

Value	Description
0	Counter retrigger event is disabled and will not be generated.
1	Counter retrigger event is enabled and will be generated for every counter retrigger.

Bit 8 – OVFEO: Overflow/Underflow Event Output Enable
This bit is used to enable the overflow/underflow event. When enabled an event will be generated when the counter reaches the TOP or the ZERO value.

Value	Description
0	Overflow/underflow counter event is disabled and will not be generated.
1	Overflow/underflow counter event is enabled and will be generated for every counter overflow/underflow.

Bits 7:6 – CNTSEL[1:0]: Timer/Counter Interrupt and Event Output Selection
These bits define on which part of the counter cycle the counter event output is generated.

Value	Name	Description
0x0	BEGIN	An interrupt/event is generated at begin of each counter cycle
0x1	END	An interrupt/event is generated at end of each counter cycle
0x2	BETWEEN	An interrupt/event is generated between each counter cycle.
0x3	BOUNDARY	An interrupt/event is generated at begin of first counter cycle, and end of last counter cycle.

Bits 5:3 – EVACT1[2:0]: Timer/Counter Event Input 1 Action
These bits define the action the TCC will perform on TCE1 event input.

Value	Name	Description
0x0	OFF	Event action disabled.
0x1	RETRIGGER	Start restart or re-trigger TC on event
0x2	DIR (asynch)	Direction control
0x3	STOP	Stop TC on event
0x4	DEC	Decrement TC on event
0x5	PPW	Period captured into CC0 Pulse Width on CC1
0x6	PWP	Period captured into CC1 Pulse Width on CC0
0x7	FAULT	Non-recoverable Fault

Bits 2:0 – EVACT0[2:0]: Timer/Counter Event Input 0 Action
These bits define the action the TCC will perform on TCE0 event input 0.

Value	Name	Description
0x0	OFF	Event action disabled.
0x1	RETRIGGER	Start restart or re-trigger TC on event
0x2	COUNTEV	Count on event.
0x3	START	Start TC on event
0x4	INC	Increment TC on EVENT
0x5	COUNT (async)	Count on active state of asynchronous event
0x6	STAMP	Capture overflow times (Max value).
0x7	FAULT	Non-recoverable Fault

Bits 27,26,25,24 – MCEOx: Match or Capture Channel x Event Output Enable
These bits control if the Match/capture event on channel x is enabled and will be generated for every match or capture.

Value	Description
0	Match/capture x event is disabled and will not be generated.
1	Match/capture x event is enabled and will be generated for every compare/capture on channel x.

Bits 19,18,17,16 – MCEIx: Match or Capture Channel x Event Input Enable
These bits indicate if the Match/capture x incoming event is enabled

These bits are used to enable match or capture input events to the CCx channel of TCC.

Value	Description
0	Incoming events are disabled.
1	Incoming events are enabled.

Bits 15,14 – TCEIx: Timer/Counter Event Input x Enable
This bit is used to enable input event x to the TCC.

Value	Description
0	Incoming event x is disabled.
1	Incoming event x is enabled.

Bits 13,12 – TCINVx: Timer/Counter Event x Invert Enable
This bit inverts the event x input.

Value	Description
0	Input event source x is not inverted.
1	Input event source x is inverted.

36.8.10. Interrupt Enable Clear

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Name: INTENCLR
Offset: 0x24
Reset: 0x000000
Property: PAC Write-Protection

Bit	23	22	21	20	19	18	17	16
					MC3	MC2	MC1	MC0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	FAULT1	FAULT0	FAULTB	FAULTA	DFS			
Access	R/W	R/W	R/W	R/W	R/W			
Reset	0	0	0	0	0			

Bit	7	6	5	4	3	2	1	0
					ERR	CNT	TRG	OVF
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 13 – FAULTB: Recoverable Fault B Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Recoverable Fault B Interrupt Disable/Enable bit, which disables the Recoverable Fault B interrupt.

Value	Description
0	The Recoverable Fault B interrupt is disabled.
1	The Recoverable Fault B interrupt is enabled.

Bit 12 – FAULTA: Recoverable Fault A Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Recoverable Fault A Interrupt Disable/Enable bit, which disables the Recoverable Fault A interrupt.

Value	Description
0	The Recoverable Fault A interrupt is disabled.
1	The Recoverable Fault A interrupt is enabled.

Bit 11 – DFS: Debug Fault State Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Debug Fault State Interrupt Disable/Enable bit, which disables the Debug Fault State interrupt.

Value	Description
0	The Debug Fault State interrupt is disabled.
1	The Debug Fault State interrupt is enabled.

Bit 3 – ERR: Error Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Error Interrupt Disable/Enable bit, which disables the Compare interrupt.

Value	Description
0	The Error interrupt is disabled.
1	The Error interrupt is enabled.

Bit 2 – CNT: Counter Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Counter Interrupt Disable/Enable bit, which disables the Counter interrupt.

Value	Description
0	The Counter interrupt is disabled.
1	The Counter interrupt is enabled.

Bit 1 – TRG: Retrigger Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Retrigger Interrupt Disable/Enable bit, which disables the Retrigger interrupt.

Value	Description
0	The Retrigger interrupt is disabled.
1	The Retrigger interrupt is enabled.

Bit 0 – OVF: Overflow Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Overflow Interrupt Disable/Enable bit, which disables the Overflow interrupt request.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Bits 19,18,17,16 – MCx: Match or Capture Channel x Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the corresponding Match or Capture Channel x Interrupt Disable/Enable bit, which disables the Match or Capture Channel x interrupt.

Value	Description
0	The Match or Capture Channel x interrupt is disabled.
1	The Match or Capture Channel x interrupt is enabled.

Bits 15,14 – FAULTx: Non-Recoverable Fault x Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Non-Recoverable Fault x Interrupt Disable/Enable bit, which disables the Non-Recoverable Fault x interrupt.

Value	Description
0	The Non-Recoverable Fault x interrupt is disabled.
1	The Non-Recoverable Fault x interrupt is enabled.

36.8.11. Interrupt Enable Set

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Name: INTENSET
Offset: 0x28
Reset: 0x000000
Property: PAC Write-Protection

Bit	23	22	21	20	19	18	17	16
					MC3	MC2	MC1	MC0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	FAULT1	FAULT0	FAULTB	FAULTA	DFS			
Access	R/W	R/W	R/W	R/W	R/W			
Reset	0	0	0	0	0			

Bit	7	6	5	4	3	2	1	0
					ERR	CNT	TRG	OVF
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 13 – FAULTB: Recoverable Fault B Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Recoverable Fault B Interrupt Disable/Enable bit, which enables the Recoverable Fault B interrupt.

Value	Description
0	The Recoverable Fault B interrupt is disabled.
1	The Recoverable Fault B interrupt is enabled.

Bit 12 – FAULTA: Recoverable Fault A Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Recoverable Fault A Interrupt Disable/Enable bit, which enables the Recoverable Fault A interrupt.

Value	Description
0	The Recoverable Fault A interrupt is disabled.
1	The Recoverable Fault A interrupt is enabled.

Bit 11 – DFS: Debug Fault State Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Debug Fault State Interrupt Disable/Enable bit, which enables the Debug Fault State interrupt.

Value	Description
0	The Debug Fault State interrupt is disabled.
1	The Debug Fault State interrupt is enabled.

Bit 3 – ERR: Error Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Error Interrupt Disable/Enable bit, which enables the Compare interrupt.

Value	Description
0	The Error interrupt is disabled.
1	The Error interrupt is enabled.

Bit 2 – CNT: Counter Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Retrigger Interrupt Disable/Enable bit, which enables the Counter interrupt.

Value	Description
0	The Counter interrupt is disabled.
1	The Counter interrupt is enabled.

Bit 1 – TRG: Retrigger Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Retrigger Interrupt Disable/Enable bit, which enables the Retrigger interrupt.

Value	Description
0	The Retrigger interrupt is disabled.
1	The Retrigger interrupt is enabled.

Bit 0 – OVF: Overflow Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Overflow Interrupt Disable/Enable bit, which enables the Overflow interrupt request.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Bits 19,18,17,16 – MCx: Match or Capture Channel x Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the corresponding Match or Capture Channel x Interrupt Disable/Enable bit, which enables the Match or Capture Channel x interrupt.

Value	Description
0	The Match or Capture Channel x interrupt is disabled.
1	The Match or Capture Channel x interrupt is enabled.

Bits 15,14 – FAULTx: Non-Recoverable Fault x Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Non-Recoverable Fault x Interrupt Disable/Enable bit, which enables the Non-Recoverable Fault x interrupt.

Value	Description
0	The Non-Recoverable Fault x interrupt is disabled.
1	The Non-Recoverable Fault x interrupt is enabled.

36.8.12. Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x2C
Reset: 0x000000
Property: -

Bit	23	22	21	20	19	18	17	16
					MC3	MC2	MC1	MC0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	FAULT1	FAULT0	FAULTB	FAULTA	DFS			
Access	R/W	R/W	R/W	R/W	R/W			
Reset	0	0	0	0	0			

Bit	7	6	5	4	3	2	1	0
					ERR	CNT	TRG	OVF
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 13 – FAULTB: Recoverable Fault B Interrupt Flag
This flag is set on the next CLK_TCC_COUNT cycle after a Recoverable Fault B occurs.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Recoverable Fault B interrupt flag.

Bit 12 – FAULTA: Recoverable Fault A Interrupt Flag
This flag is set on the next CLK_TCC_COUNT cycle after a Recoverable Fault B occurs.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Recoverable Fault B interrupt flag.

Bit 11 – DFS: Debug Fault State Interrupt Flag
This flag is set on the next CLK_TCC_COUNT cycle after an Debug Fault State occurs.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Debug Fault State interrupt flag.

Bit 3 – ERR: Error Interrupt Flag
This flag is set if a new capture occurs on a channel when the corresponding Match or Capture Channel x interrupt flag is one. In which case there is nowhere to store the new capture.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the error interrupt flag.

Bit 2 – CNT: Counter Interrupt Flag
This flag is set on the next CLK_TCC_COUNT cycle after a counter event occurs.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the CNT interrupt flag.

Bit 1 – TRG: Retrigger Interrupt Flag
This flag is set on the next CLK_TCC_COUNT cycle after a counter retrigger occurs.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the re-trigger interrupt flag.

Bit 0 – OVF: Overflow Interrupt Flag
This flag is set on the next CLK_TCC_COUNT cycle after an overflow condition occurs.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Overflow interrupt flag.

Bits 19,18,17,16 – MCx: Match or Capture Channel x Interrupt Flag
This flag is set on the next CLK_TCC_COUNT cycle after a match with the compare condition or once CCx register contain a valid capture value.

Writing a '0' to one of these bits has no effect.

Writing a '1' to one of these bits will clear the corresponding Match or Capture Channel x interrupt flag

In Capture operation, this flag is automatically cleared when CCx register is read.

Bits 15,14 – FAULTx: Non-Recoverable Fault x Interrupt Flag
This flag is set on the next CLK_TCC_COUNT cycle after a Non-Recoverable Fault x occurs.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Non-Recoverable Fault x interrupt flag.

36.8.13. Status

Name: STATUS
Offset: 0x30
Reset: 0x00000001
Property: -

Bit	31	30	29	28	27	26	25	24
					CMP3	CMP2	CMP1	CMP0
Access					R	R	R	R
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
					CCBV3	CCBV2	CCBV1	CCBV0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	FAULT1	FAULT0	FAULTB	FAULTA	FAULT1IN	FAULT0IN	FAULTBIN	FAULTAIN
Access	R/W	R/W	R/W	R/W	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PERBV		PATTBV		DFS		IDX	STOP
Access	R/W		R/W		R/W		R	R
Reset	0		0		0		0	1

Bit 13 – FAULTB: Recoverable Fault B State
This bit is set by hardware as soon as recoverable Fault B condition occurs.

This bit can be clear by hardware when Fault B action is resumed, or by writing a '1' to this bit when the corresponding FAULTBIN bit is low. If software halt command is enabled (FAULTB.HALT=SW), clearing this bit will release the timer/counter.

Bit 12 – FAULTA: Recoverable Fault A State
This bit is set by hardware as soon as recoverable Fault A condition occurs.

This bit can be clear by hardware when Fault A action is resumed, or by writing a '1' to this bit when the corresponding FAULTAIN bit is low. If software halt command is enabled (FAULTA.HALT=SW), clearing this bit will release the timer/counter.

Bit 11 – FAULT1IN: Non-Recoverable Fault 1 Input
This bit is set while an active Non-Recoverable Fault 1 input is present.

Bit 10 – FAULT0IN: Non-Recoverable Fault 0 Input
This bit is set while an active Non-Recoverable Fault 0 input is present.

Bit 9 – FAULTBIN: Recoverable Fault B Input
This bit is set while an active Recoverable Fault B input is present.

Bit 8 – FAULTAIN: Recoverable Fault A Input
This bit is set while an active Recoverable Fault A input is present.

Bit 7 – PERBV: Period Buffer Valid
This bit is set when a new value is written to the PERB register. This bit is automatically cleared by hardware on UPDATE condition or by writing a '1' to this bit.

Bit 5 – PATTBV: Pattern Generator Value Buffer Valid
This bit is set when a new value is written to the PATTB register. This bit is automatically cleared by hardware on UPDATE condition or by writing a '1' to this bit.

Bit 3 – DFS: Debug Fault State
This bit is set by hardware in debug mode when DDBGCTRL.FDDBD bit is set. The bit is cleared by writing a '1' to this bit and when the TCC is not in debug mode.

When the bit is set, the counter is halted and the waveforms state depend on DRVCTRL.NRE and DRVCTRL.NRV registers.

Bit 1 – IDX: Ramp Index
In RAMP2 and RAMP2A operation, the bit is cleared during the cycle A and set during the cycle B. In RAMP1 operation, the bit always reads zero. For details on ramp operations, refer to Ramp Operations on page 774.

Bit 0 – STOP: Stop
This bit is set when the TCC is disabled either on a STOP command or on an UPDATE condition when One-Shot operation mode is enabled (CTRLBSET.ONESHOT=1).

This bit is clear on the next incoming counter increment or decrement.

Value	Description
0	Counter is running.
1	Counter is stopped.

Bits 27,26,25,24 – CMPx: Channel x Compare Value
This bit reflects the channel x output compare value.

Value	Description
0	Channel compare output value is 0.
1	Channel compare output value is 1.

Bits 19,18,17,16 – CCBVx: Channel x Compare or Capture Buffer Valid
For a compare channel, this bit is set when a new value is written to the corresponding CCBx register. The bit is cleared either by writing a '1' to the corresponding location or automatically on an UPDATE condition.

For a capture channel, the bit is set when a valid capture value is stored in the CCBx register. The bit is automatically cleared when the CCx register is read.

Bits 15,14 – FAULTx: Non-recoverable Fault x State
This bit is set by hardware as soon as non-recoverable Fault x condition occurs.

This bit is cleared by writing a one to this bit and when the corresponding FAULTxIN status bit is low.

Once this bit is clear, the timer/counter will restart from the last COUNT value. To restart the timer/counter from BOTTOM, the timer/counter restart command must be executed before clearing the corresponding STATEx bit. For further details on timer/counter commands, refer to available commands description (CTRLBSET on page 799.CMD).

36.8.14. Counter Value, CTRLA.RESOLUTION = NONE

Name: COUNT
Offset: 0x34
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Bit	31	30	29	28	27	26	25	24
	COUNT[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	COUNT[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	COUNT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COUNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – COUNT[31:0]: Counter Value
These bits hold the value of the counter register.

COUNT[31:23] will read zero when TCC is configured as 16- or 24-bit timer/counter.

COUNT[23:16] will read zero when TCC is configured as 16-bit timer/counter.

36.8.15. Counter Value, CTRLA.RESOLUTION = DITH4

Name: COUNT
Offset: 0x34
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Bit	31	30	29	28	27	26	25	24
	COUNT[27:20]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	COUNT[19:12]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	COUNT[11:4]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COUNT[3:0]							
Access	R/W	R/W	R/W	R/W				
Reset	0	0	0	0				

Bits 31:4 – COUNT[27:0]: Counter Value
These bits hold the value of the counter register.

COUNT[27:23] will read zero when TCC is configured as 16- or 24-bit timer/counter.

COUNT[23:16] will read zero when TCC is configured as 16-bit timer/counter.

36.8.16. Counter Value, CTRLA.RESOLUTION = DITH5

Name: COUNT
Offset: 0x34
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Bit	31	30	29	28	27	26	25	24
	COUNT[26:19]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	COUNT[18:11]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	COUNT[10:3]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COUNT[2:0]							
Access	R/W	R/W	R/W					
Reset	0	0	0					

Bits 31:5 – COUNT[26:0]: Counter Value
These bits hold the value of the counter register.

COUNT[26:23] will read zero when TCC is configured as 16- or 24-bit timer/counter.

COUNT[23:16] will read zero when TCC is configured as 16-bit timer/counter.

36.8.17. Counter Value, CTRLA.RESOLUTION = DITH6

Name: COUNT
Offset: 0x34
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Bit	31	30	29	28	27	26	25	24
	COUNT[25:18]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	COUNT[17:10]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	COUNT[9:2]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COUNT[1:0]							
Access	R/W	R/W						
Reset	0	0						

Bits 31:6 – COUNT[25:0]: Counter Value
These bits hold the value of the counter register.

COUNT[25:23] will read zero when TCC is configured as 16- or 24-bit timer/counter.

COUNT[23:16] will read zero when TCC is configured as 16-bit timer/counter.

36.8.18. Pattern

Name: PATT
Offset: 0x38
Reset: 0x0000
Property: Write-Synchronized, Read-Synchronized

Bit	15	14	13	12	11	10	9	8
	PGV7	PGV6	PGV5	PGV4	PGV3	PGV2	PGV1	PGV0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PGE7	PGE6	PGE5	PGE4	PGE3	PGE2	PGE1	PGE0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:8 – PGVn: Pattern Generation Output Value
This register holds the values of pattern for each waveform output.

Bits 7:0 – PGEn: Pattern Generation Output Enable
This register holds the enable status of pattern generation for each waveform output. A bit written to '1' will override the corresponding SWAP output with the corresponding PGVn value.

36.8.19. Waveform

Name: WAVE
Offset: 0x3C
Reset: 0x00000000
Property: Write-Synchronized, Read-Synchronized

Bit	31	30	29	28	27	26	25	24
					SWAP3	SWAP2	SWAP1	SWAP0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
					POL3	POL2	POL1	POL0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
					CIRCCEN3	CIRCCEN2	CIRCCEN1	CIRCCEN0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CIRPEREN		RAMP[1:0]			WAVEGEN[2:0]		
Access	R/W		R/W	R/W		R/W	R/W	R/W
Reset	0		0	0		0	0	0

Bit 7 – CIRPEREN: Circular Period Enable
Setting this bits enable the period circular buffer option. When the bit is set, the PER register value is copied-back into the PERB register on UPDATE condition.

Bits 5:4 – RAMP[1:0]: Ramp Operation
These bits select Ramp operation (RAMP). These bits are not synchronized.

Value	Name	Description
0x0	RAMP1	RAMP1 operation
0x1	RAMP2A	Alternative RAMP2 operation
0x2	RAMP2	RAMP2 operation
0x3	RAMP2C	Critical RAMP2 operation

Bits 2:0 – WAVEGEN[2:0]: Waveform Generation Operation
These bits select the waveform generation operation. The settings impact the top value and control if frequency or PWM waveform generation should be used. These bits are not synchronized.

Value	Name	Description						
		Operation	Top	Update	Waveform Output On Match	Waveform Output On Update	OVFIF/Event Up Down	
0x0	NFRQ	Normal Frequency	PER	TOP/Zero	Toggle	Stable	TOP	Zero
0x1	MFRQ	Match Frequency	CC0	TOP/Zero	Toggle	Stable	TOP	Zero
0x2	NPWM	Normal PWM	PER	TOP/Zero	Set	Clear	TOP	Zero
0x3	Reserved	–	–	–	–	–	–	–
0x4	DSCRITICAL	Dual-slope PWM	PER	Zero	~DIR	Stable	–	Zero
0x5	DSBOTTOM	Dual-slope PWM	PER	Zero	~DIR	Stable	–	Zero
0x6	DSBOTH	Dual-slope PWM	PER	TOP & Zero	~DIR	Stable	TOP	Zero
0x7	DSTOP	Dual-slope PWM	PER	Zero	~DIR	Stable	TOP	–

Bits 27,26,25,24 – SWAPx : Swap DTI Output Pair x
Setting these bits enables output swap of DTI outputs [x] and [x+WO_NUM/2]. Note the DTIxEN settings will not affect the swap operation.

Bits 19,18,17,16 – POLx: Channel Polarity x
Setting these bits enables the output polarity in single-slope and dual-slope PWM operations.

Value	Name	Description
0	(single-slope PWM waveform generation)	Compare output is initialized to ~DIR and set to DIR when TCC counter matches CCx value
1	(single-slope PWM waveform generation)	Compare output is initialized to DIR and set to ~DIR when TCC counter matches CCx value.
0	(dual-slope PWM waveform generation)	Compare output is set to ~DIR when TCC counter matches CCx value
1	(dual-slope PWM waveform generation)	Compare output is set to DIR when TCC counter matches CCx value.

Bits 11,10,9,8 – CIRCCENx: Circular CC Enable x
Setting this bits enables the compare circular buffer option on channel. When the bit is set, CCx register value is copied-back into the CCx register on UPDATE condition.

36.8.20. Period Value, CTRLA.RESOLUTION = NONE

Name: PER
Offset: 0x40
Reset: 0xFFFFFFFF
Property: Write-Synchronized, Read-Synchronized

Bit	31	30	29	28	27	26	25	24
	PER[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	23	22	21	20	19	18	17	16
	PER[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8
	PER[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
	PER[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 31:0 – PER[31:0]: Period Value
These bits hold the value of the period register.

PER[31:23] will read zero when TCC is configured as 16- or 24-bit timer/counter.

PER[23:16] will read zero when TCC is configured as 16-bit timer/counter.

36.8.21. Period Value, CTRLA.RESOLUTION = DITH4

Name: PER
Offset: 0x40
Reset: 0xFFFFFFFF
Property: Write-Synchronized, Read-Synchronized

Bit	31	30	29	28	27	26	25	24
	PER[27:20]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	23	22	21	20	19	18	17	16
	PER[19:12]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8
	PER[11:4]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
	PER[3:0]				DITHERCY[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 31:4 – PER[27:0]: Period Value
These bits hold the value of the period buffer register. The value is copied to PER register on UPDATE condition. The number of bits in this field corresponds to the size of the counter.

Bits 3:0 – DITHERCY[3:0]: Dithering Cycle Number
These bits hold the number of extra cycles that are added on the PWM period every 16 PWM frames.

36.8.22. Period Value, CTRLA.RESOLUTION = DITH5

Name: PER
Offset: 0x40
Reset: 0xFFFFFFFF
Property: Write-Synchronized, Read-Synchronized

Bit	31	30	29	28	27	26	25	24
	PER[26:19]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	23	22	21	20	19	18	17	16
	PER[18:11]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8
	PER[10:3]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
	PER[2:0]			DITHERCY[4:0]				
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 31:5 – PER[26:0]: Period Value
These bits hold the value of the period register.

PER[26:23] will read zero when TCC is configured as 16- or 24-bit timer/counter.

PER[23:16] will read zero when TCC is configured as 16-bit timer/counter.

Bits 4:0 – DITHERCY[4:0]: Dithering Cycle Number
These bits hold the number of extra cycles that are added on the PWM period every 32 PWM frames.

36.8.23. Period Value, CTRLA.RESOLUTION = DITH6

Name: PER
Offset: 0x40
Reset: 0xFFFFFFFF
Property: Write-Synchronized, Read-Synchronized

Bit	31	30	29	28	27	26	25	24
	PER[25:18]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	23	22	21	20	19	18	17	16
	PER[17:10]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8
	PER[9:2]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
	PER[1:0]		DITHERCY[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 31:6 – PER[25:0]: Period Value
These bits hold the value of the period register.

PER[25:23] will read zero when TCC is configured as 16- or 24-bit timer/counter.

PER[23:16] will read zero when TCC is configured as 16-bit timer/counter.

Bits 5:0 – DITHERCY[5:0]: Dithering Cycle Number
These bits hold the number of extra cycles that are added on the PWM period every 64 PWM frames.

36.8.24. Compare/Capture Channel x, CTRLA.RESOLUTION = NONE

The CCx register represents the 16-, 24- or 32-bit value, CCx. The register has two functions, depending of the mode of operation.

For capture operation, this register represents the second buffer level and access point for the CPU and DMA.

For compare operation, this register is continuously compared to the counter value. Normally, the output form the comparator is then used for generating waveforms.

CCx register is updated with the buffer value from their corresponding CCBx register when an UPDATE condition occurs.

In addition, in match frequency operation, the CC0 register controls the counter period.

Name: CCx
Offset: 0x44 + 4*x [x = 0..3]
Reset: 0x00000000
Property: Write-Synchronized, Read-Synchronized

Bit	31	30	29	28	27	26	25	24
	CC0[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CC0[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CC0[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CC0[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CCx: Channel x Compare/Capture Value
These bits hold the value of the Channel x compare/capture register.

Bit 31:24 (CCx[31:24]) are read zero when TCC is configured as 24-bits timer/counter.

Bit 31:16 (CCx[31:16]) are read zero when TCC is configured as 16-bits timer/counter.

36.8.25. Compare/Capture Channel x, CTRLA.RESOLUTION = DITH4

The CCx register represents the 16-, 24- or 32-bit value, CCx. The register has two functions, depending of the mode of operation.

For capture operation, this register represents the second buffer level and access point for the CPU and DMA.

For compare operation, this register is continuously compared to the counter value. Normally, the output form the comparator is then used for generating waveforms.

CCx register is updated with the buffer value from their corresponding CCBx register when an UPDATE condition occurs.

In addition, in match frequency operation, the CC0 register controls the counter period.

Name: CCx
Offset: 0x44 + 4*x [x = 0..3]
Reset: 0x00000000
Property: Write-Synchronized, Read-Synchronized

Bit	31	30	29	28	27	26	25	24
	CC0[27:20]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CC0[19:12]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CC0[11:4]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CC0[3:0]				DITHERCY[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:4 – CCx: Channel x Compare/Capture Value
These bits hold the value of the Channel x compare/capture register.

Bit 31:24 (CCx[27:20]) are read zero when TCC is configured as 24-bits timer/counter.

Bit 31:16 (CCx[27:12]) are read zero when TCC is configured as 16-bits timer/counter.

Bits 3:0 – DITHERCY[3:0]: Dithering Cycle Number
These bits hold the number of extra cycles that are added on the PWM pulse width every 16 PWM frames.

36.8.26. Compare/Capture Channel x, CTRLA.RESOLUTION = DITH5

The CCx register represents the 16-, 24- or 32-bit value, CCx. The register has two functions, depending of the mode of operation.

For capture operation, this register represents the second buffer level and access point for the CPU and DMA.

For compare operation, this register is continuously compared to the counter value. Normally, the output form the comparator is then used for generating waveforms.

CCx register is updated with the buffer value from their corresponding CCBx register when an UPDATE condition occurs.

In addition, in match frequency operation, the CC0 register controls the counter period.

Name: CCx
Offset: 0x44 + 4*x [x = 0..3]
Reset: 0x00000000
Property: Write-Synchronized, Read-Synchronized

Bit	31	30	29	28	27	26	25	24
	CC0[26:19]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CC0[18:11]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CC0[10:3]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CC0[2:0]			DITHERCY[4:0]				
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:5 – CCx: Channel x Compare/Capture Value
These bits hold the value of the Channel x compare/capture register.

Bit 31:24 (CCx[26:19]) are read zero when TCC is configured as 24-bits timer/counter.

Bit 31:16 (CCx[26:11]) are read zero when TCC is configured as 16-bits timer/counter.

Bits 4:0 – DITHERCY[4:0]: Dithering Cycle Number
These bits hold the number of extra cycles that are added on the PWM pulse width every 32 PWM frames.

36.8.27. Compare/Capture Channel x, CTRLA.RESOLUTION = DITH6

The CCx register represents the 16-, 24- or 32-bit value, CCx. The register has two functions, depending of the mode of operation.

For capture operation, this register represents the second buffer level and access point for the CPU and DMA.

For compare operation, this register is continuously compared to the counter value. Normally, the output form the comparator is then used for generating waveforms.

CCx register is updated with the buffer value from their corresponding CCBx register when an UPDATE condition occurs.

In addition, in match frequency operation, the CC0 register controls the counter period.

Name: CCx
Offset: 0x44 + 4*x [x = 0..3]
Reset: 0x00000000
Property: Write-Synchronized, Read-Synchronized

Bit	31	30	29	28	27	26	25	24
	CC0[25:18]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CC0[17:10]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CC0[9:2]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CC0[1:0]		DITHERCY[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:6 – CCx: Channel x Compare/Capture Value
These bits hold the value of the Channel x compare/capture register.

Bit 31:24 (CCx[25:23]) are read zero when TCC is configured as 24-bits timer/counter.

Bit 31:16 (CCx[25:10]) are read zero when TCC is configured as 16-bits timer/counter.

Bits 5:0 – DITHERCY[5:0]: Dithering Cycle Number
These bits hold the number of extra cycles that are added on the PWM pulse width every 64 PWM frames.

36.8.28. Pattern Buffer

Name: PATTBUF
Offset: 0x64
Reset: 0x0000
Property: Write-Synchronized, Read-Synchronized

Bit	15	14	13	12	11	10	9	8
	PGVB7	PGVB6	PGVB5	PGVB4	PGVB3	PGVB2	PGVB1	PGVB0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PGEB7	PGEB6	PGEB5	PGEB4	PGEB3	PGEB2	PGEB1	PGEB0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:8 – PGVBn: Pattern Generation Output Value Buffer
This register is the buffer for the PGV register. If double buffering is used, valid content in this register is copied to the PGV register on an UPDATE condition.

Bits 7:0 – PGEBn: Pattern Generation Output Enable Buffer
This register is the buffer of the PGE register. If double buffering is used, valid content in this register is copied into the PGE register at an UPDATE condition.

36.8.29. Period Buffer Value, CTRLA.RESOLUTION=NONE

Name: PERBUF
Offset: 0x6C
Reset: 0xFFFFFFFF
Property: Write-Synchronized, Read-Synchronized

Bit	31	30	29	28	27	26	25	24
	PERBBUF[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	23	22	21	20	19	18	17	16
	PERBBUF[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8
	PERBBUF[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
	PERBBUF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 31:0 – PERBBUF[31:0]: Period Buffer Value
These bits hold the value of the period buffer register. The value is copied to PER register on UPDATE condition.

Bits 31:24 (PERBUF[31:24]) are read zero when TCC is configured as 24-bit timer/counter.

Bits 31:16 (PERBUF[31:16]) are read zero when TCC is configured as 16-bit timer/counter.

36.8.30. Period Buffer Value, CTRLA.RESOLUTION=DITH4

Name: PERBUF
Offset: 0x6C
Reset: 0xFFFFFFFF
Property: Write-Synchronized, Read-Synchronized

Bit	31	30	29	28	27	26	25	24
					PERBUF[27:24]			
Access					R/W	R/W	R/W	R/W
Reset					1	1	1	1

Bit	23	22	21	20	19	18	17	16
	PERBUF[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8
	PERBUF[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
	PERBUF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 27:0 – PERBUF[27:0]: Period Buffer Value
These bits hold the value of the period buffer register. The value is copied to PER register on UPDATE condition.

Bits 31:24 (PERBUF[27:20]) are read zero when TCC is configured as 24-bit timer/counter.

Bits 31:16 (PERBUF[27:12]) are read zero when TCC is configured as 16-bit timer/counter.

Bits 3:0 – DITHERBUF[3:0]: Dithering Buffer Cycle Number
These bits represent the PER.DITHER bits buffer. When the double buffering is enable, DITHERBUF bits value is copied to the PER.DITHER bits on an UPDATE condition.

36.8.31. Period Buffer Value, CTRLA.RESOLUTION=DITH5

Name: PERBUF
Offset: 0x6C
Reset: 0xFFFFFFFF
Property: Write-Synchronized, Read-Synchronized

Bit	31	30	29	28	27	26	25	24
						PERBUF[26:24]		
Access						R/W	R/W	R/W
Reset						1	1	1

Bit	23	22	21	20	19	18	17	16
	PERBUF[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8
	PERBUF[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
	PERBUF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 26:0 – PERBUF[26:0]: Period Buffer Value
These bits hold the value of the period buffer register. The value is copied to PER register on UPDATE condition.

Bits 31:24 (PERBUF[26:19]) are read zero when TCC is configured as 24-bit timer/counter.

Bits 31:16 (PERBUF[26:11]) are read zero when TCC is configured as 16-bit timer/counter.

Bits 4:0 – DITHERBUF[4:0]: Dithering Buffer Cycle Number
These bits represent the PER.DITHER bits buffer. When the double buffering is enable, DITHERBUF bits value is copied to the PER.DITHER bits on an UPDATE condition.

36.8.32. Period Buffer Value, CTRLA.RESOLUTION=DITH6

Name: PERBUF
Offset: 0x6C
Reset: 0xFFFFFFFF
Property: Write-Synchronized, Read-Synchronized

Bit	31	30	29	28	27	26	25	24
							PERBUF[25:24]	
Access							R/W	R/W
Reset							1	1

Bit	23	22	21	20	19	18	17	16
	PERBUF[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8
	PERBUF[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
	PERBUF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bits 25:0 – PERBUF[25:0]: Period Buffer Value
These bits hold the value of the period buffer register. The value is copied to PER register on UPDATE condition.

Bits 31:24 (PERBUF[25:18]) are read zero when TCC is configured as 24-bit timer/counter.

Bits 31:16 (PERBUF[25:10]) are read zero when TCC is configured as 16-bit timer/counter.

Bits 5:0 – DITHERBUF[5:0]: Dithering Buffer Cycle Number
These bits represent the PER.DITHER bits buffer. When the double buffering is enable, DITHERBUF bits value is copied to the PER.DITHER bits on an UPDATE condition.

36.8.33. Channel x Compare/Capture Buffer Value, CTRLA.RESOLUTION=NONE

CCBUFx is copied into CCx at TCC update time

Name: CCBUFx
Offset: 0x70 + 4*x [x = 0..3]
Reset: 0x00000000
Property: Write-Synchronized, Read-Synchronized

Bit	31	30	29	28	27	26	25	24
	CCBUF0[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CCBUF0[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CCBUF0[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CCBUF0[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CCBUFx: Channel x Compare/Capture Value
These bits hold the value of the Channel x Compare/Capture Buffer Value register. The register serves as the buffer for the associated compare or capture registers (CCx). Accessing this register using the CPU or DMA will affect the corresponding CCBVx status bit.

Bit 31:24 (CCBUF[31:24]) will read zero when TCC is configured as 24-bit timer/counter.

Bit 31:16 (CCBUF[31:16]) will read zero when TCC is configured as 24-bit timer/counter.

36.8.34. Channel x Compare/Capture Buffer Value, CTRLA.RESOLUTION=DITH4

CCBUFx is copied into CCx at TCC update time

Name: CCBUFx
Offset: 0x70 + 4*x [x = 0..3]
Reset: 0x00000000
Property: Write-Synchronized, Read-Synchronized

Bit	31	30	29	28	27	26	25	24
	CCBUF0[27:20]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CCBUF0[19:12]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CCBUF0[11:4]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CCBUF0[3:0]				DITHERBUF[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:4 – CCBUFx: Channel x Compare/Capture Buffer Value
These bits hold the value of the Channel x Compare/Capture Buffer Value register. The register serves as the buffer for the associated compare or capture registers (CCx). Accessing this register using the CPU or DMA will affect the corresponding CCBVx status bit.

Bit 31:24 (CCBUF[27:20]) will read zero when TCC is configured as 24-bit timer/counter.

Bit 31:16 (CCBUF[27:12]) will read zero when TCC is configured as 16-bit timer/counter.

Bits 3:0 – DITHERBUF[3:0]: Dithering Buffer Cycle Number
These bits represent the CCx.DITHER bits buffer. When the double buffering is enable, DITHERBUF bits value is copied to the CCx.DITHER bits on an UPDATE condition.

36.8.35. Channel x Compare/Capture Buffer Value, CTRLA.RESOLUTION=DITH5

CCBUFx is copied into CCx at TCC update time

Name: CCBUFx
Offset: 0x70 + 4*x [x = 0..3]
Reset: 0x00000000
Property: Write-Synchronized, Read-Synchronized

Bit	31	30	29	28	27	26	25	24
	CCBUF0[26:19]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CCBUF0[18:11]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CCBUF0[10:3]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CCBUF0[2:0]			DITHERBUF[4:0]				
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:5 – CCBUFx: Channel x Compare/Capture Buffer Value
These bits hold the value of the Channel x Compare/Capture Buffer Value register. The register serves as the buffer for the associated compare or capture registers (CCx). Accessing this register using the CPU or DMA will affect the corresponding CCBVx status bit.

Bit 31:24 (CCBUF[26:19]) will read zero when TCC is configured as 24-bit timer/counter.

Bit 31:16 (CCBUF[26:11]) will read zero when TCC is configured as 16-bit timer/counter.

Bits 4:0 – DITHERBUF[4:0]: Dithering Buffer Cycle Number
These bits represent the CCx.DITHER bits buffer. When the double buffering is enable, DITHERBUF bits value is copied to the CCx.DITHER bits on an UPDATE condition.

36.8.36. Channel x Compare/Capture Buffer Value, CTRLA.RESOLUTION=DITH6

CCBUFx is copied into CCx at TCC update time

Name: CCBUFx
Offset: 0x70 + 4*x [x = 0..3]
Reset: 0x00000000
Property: Write-Synchronized, Read-Synchronized

Bit	31	30	29	28	27	26	25	24
	CCBUF0[25:18]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CCBUF0[17:10]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CCBUF0[9:2]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CCBUF0[1:0]		DITHERBUF[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:6 – CCBUFx: Channel x Compare/Capture Buffer Value
These bits hold the value of the Channel x Compare/Capture Buffer Value register. The register serves as the buffer for the associated compare or capture registers (CCx). Accessing this register using the CPU or DMA will affect the corresponding CCBVx status bit.

Bit 31:24 (CCBUF[25:18]) will read zero when TCC is configured as 24-bit timer/counter.

Bit 31:16 (CCBUF[25:10]) will read zero when TCC is configured as 16-bit timer/counter.

Bits 5:0 – DITHERBUF[5:0]: Dithering Buffer Cycle Number
These bits represent the CCx.DITHER bits buffer. When the double buffering is enable, DITHERBUF bits value is copied to the CCx.DITHER bits on an UPDATE condition.

37. TRNG – True Random Number Generator

37.1. Overview

The True Random Number Generator (TRNG) generates unpredictable random numbers that are not generated by an algorithm. It passes the American NIST Special Publication 800-22 and Diehard Random Tests Suites.

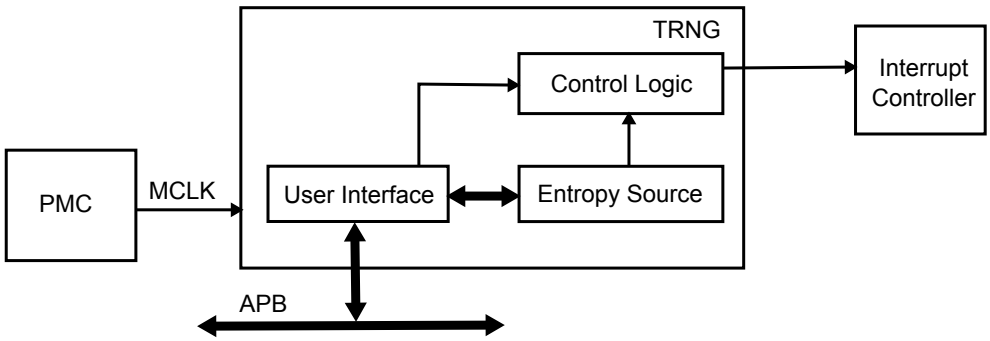
The TRNG may be used as an entropy source for seeding an NIST approved DRNG (Deterministic RNG) as required by FIPS PUB 140-2 and 140-3.

37.2. Features

- Passed NIST Special Publication 800-22 Tests Suite
- Passed Diehard Random Tests Suite
- May be used as Entropy Source for seeding an NIST approved DRNG (Deterministic RNG) as required by FIPS PUB 140-2 and 140-3
- Provides a 32-bit random number every 84 clock cycles

37.3. Block Diagram

Figure 37-1. TRNG Block Diagram.



37.4. Signal Description

Not applicable.

37.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

37.5.1. I/O Lines

Not applicable.

37.5.2. Power Management

The TRNG will continue to operate in any sleep mode, as long as its source clock is running. The TRNG interrupts can be used to wake up the device from sleep modes. Events connected to the event system can trigger other operations in the system without exiting sleep modes.

Related Links

PM – Power Manager on page 186

37.5.3. Clocks

The TRNG bus clock (CLK_TRNG_APB) can be enabled and disabled in the Main Clock module, and the default state of CLK_TRNG_APB can be found in *Peripheral Clock Masking*.

Related Links

Peripheral Clock Masking on page 151

37.5.4. DMA

Not applicable.

37.5.5. Interrupts

The interrupt request line is connected to the interrupt controller. Using the TRNG interrupt(s) requires the interrupt controller to be configured first. Refer to NVIC - Nested Interrupt *Nested Vector Interrupt Controller* for details.

Related Links

Nested Vector Interrupt Controller on page 51

37.5.6. Events

The events are connected to the Event System. Refer to *EVSYS – Event System* for details on how to configure the Event System.

Related Links

EVSYS – Event System on page 536

37.5.7. Debug Operation

When the CPU is halted in debug mode the TRNG continues normal operation. If the TRNG is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

37.5.8. Register Access Protection

All registers with write-access are optionally write-protected by the Peripheral Access Controller (PAC), except the following register:

Interrupt Flag Status and Clear (INTFLAG) register

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

37.5.9. Analog Connections

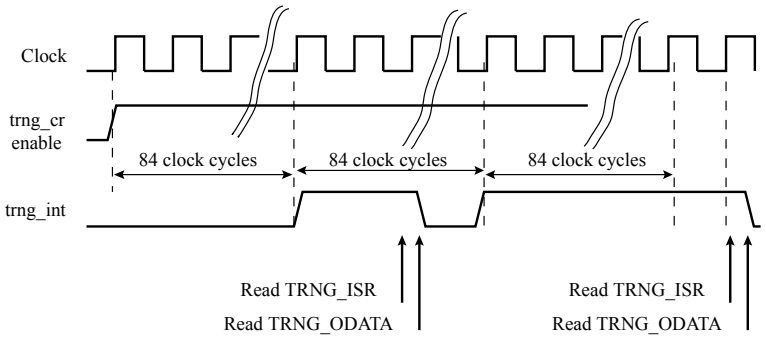
Not applicable.

37.6. Functional Description

37.6.1. Principle of Operation

As soon as the TRNG is enabled, the module automatically provides a new 32-bit random number every 84 CLK_TRNG_APB clock cycles. When new data is available, an optional interrupt or event can be generated.

Figure 37-2. TRNG Data Generation Sequence



37.6.2. Basic Operation

37.6.2.1. Initialization

The following register is enable-protected, meaning that it can only be written when the TRNG is disabled (CTRLA.ENABLE is zero):

Event Control register (EVCTRL)

Enable-protection is denoted by the Enable-Protected property in the register description.

37.6.2.2. Enabling, Disabling and Resetting

The TRNG is enabled by writing '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The TRNG is disabled by writing a zero to CTRLA.ENABLE.

37.6.3. Interrupts

The TRNG has the following interrupt source:

- Data Ready(DATARDY): Indicates that a new random data is available in DATA register and ready to be read.
  This interrupt is a synchronous wake-up source. See *Sleep Mode Controller* for details.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register.

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, or the interrupt is disabled. See INTFLAG on page 214 for details on how to clear interrupt flags. All interrupt requests from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. Refer to *Nested Vector Interrupt Controller* for details. The user must read the INTFLAG register to determine which interrupt condition is present.

Note that interrupts must be globally enabled for interrupt requests to be generated. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

Sleep Mode Controller on page 191

Nested Vector Interrupt Controller on page 51

37.6.4. Events

The TRNG can generate the following output event:

- Data Ready (DATARDY): Generated when a new random number is available in the DATA register.

Writing '1' to an Event Output bit in the Event Control Register (EVCTRL.xxEO) enables the corresponding output event. Writing a '0' to this bit disables the corresponding output event. Refer to *EVSYS – Event System* for details on configuring the Event System.

Related Links

EVSYS – Event System on page 536

37.6.5. Sleep Mode Operation

The Run in Standby bit in Control A register (CTRLA.RUNSTDBY) controls the behavior of the TRNG during standby sleep mode:

When this bit is '0', the TRNG is disabled during sleep, but maintains its current configuration.

When this bit is '1', the TRNG continues to operate during sleep and any enabled TRNG interrupt source can wake up the CPU.

37.6.6. Synchronization

Not applicable.

37.7. Register Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0		RUNSTDBY					ENABLE	
0x01 ... 0x03	Reserved									
0x04	EVCTRL	7:0								DATARDYEO
0x05 ... 0x07	Reserved									
0x08	INTENCLR	7:0								DATARDY
0x09	INTENSET	7:0								DATARDY
0x0A	INTFLAG	7:0								DATARDY
0x0B ... 0x1F	Reserved									
0x20	DATA	7:0	DATA[7:0]							
0x21		15:8	DATA[15:8]							
0x22		23:16	DATA[23:16]							
0x23		31:24	DATA[31:24]							

37.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

Refer to *PAC - Peripheral Access Controller* and Synchronization on page 852for details.

Related Links

PAC - Peripheral Access Controller on page 58

37.8.1. Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
		RUNSTDBY					ENABLE	
Access		R/W					R/W	
Reset		0					0	

Bit 6 – RUNSTDBY: Run in Standby
This bit controls how the ADC behaves during standby sleep mode:

Value	Description
0	The TRNG is halted during standby sleep mode.
1	The TRNG is not stopped in standby sleep mode.

Bit 1 – ENABLE: Enable

Value	Description
0	The TRNG is disabled.
1	The TRNG is enabled.

37.8.2. Event Control

Name: EVCTRL
Offset: 0x04
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
								DATARDYEO
Access								R/W
Reset								0

Bit 0 – DATARDYEO: Data Ready Event Output
This bit indicates whether the Data Ready event output is enabled or not and an output event will be generated when a new random value is completed.

Value	Description
0	Data Ready event output is disabled and an event will not be generated.
1	Data Ready event output is enabled and an event will be generated.

37.8.3. Interrupt Enable Clear

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Name: INTENCLR
Offset: 0x08
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DATARDY
Access								R/W
Reset								0

Bit 0 – DATARDY: Data Ready Interrupt Enable

Writing a '1' to this bit will clear the Data Ready Interrupt Enable bit, which disables the corresponding interrupt request.

Value	Description
0	The DATARDY interrupt is disabled.
1	The DATARDY interrupt is enabled.

37.8.4. Interrupt Enable Set

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Name: INTENSET
Offset: 0x09
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DATARDY
Access								R/W
Reset								0

Bit 0 – DATARDY: Data Ready Interrupt Enable

Writing a '1' to this bit will set the Data Ready Interrupt Enable bit, which enables the corresponding interrupt request.

Value	Description
0	The DATARDY interrupt is disabled.
1	The DATARDY interrupt is enabled.

37.8.5. Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x0A
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
								DATARDY
Access								R/W
Reset								0

Bit 0 – DATARDY: Data Ready
This flag is set when a new random value is generated, and an interrupt will be generated if INTENCLR/SET.DATARDY=1.

This flag is cleared by writing a '1' to the flag or by reading the DATA register.

Writing a '0' to this bit has no effect.

37.8.6. Output Data

Name: DATA
Offset: 0x20
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	DATA[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DATA[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DATA[31:0]: Output Data
These bits hold the 32-bit randomly generated output data.

38. AES – Advanced Encryption Standard

38.1. Overview

The Advanced Encryption Standard (AES) is a specification for the encryption of electronic data established by the U.S. National Institute of Standards and Technology (NIST) in 2001. AES operates on a 128-bit block of input data. The key size used for an AES cipher specifies the number of repetitions of transformation rounds that convert the input, called the plaintext, into the final output, called the ciphertext. The number of rounds of repetition is as follows:

- 10 rounds of repetition for 128-bit keys
- 12 rounds of repetition for 192-bit keys
- 14 rounds of repetition for 256-bit keys

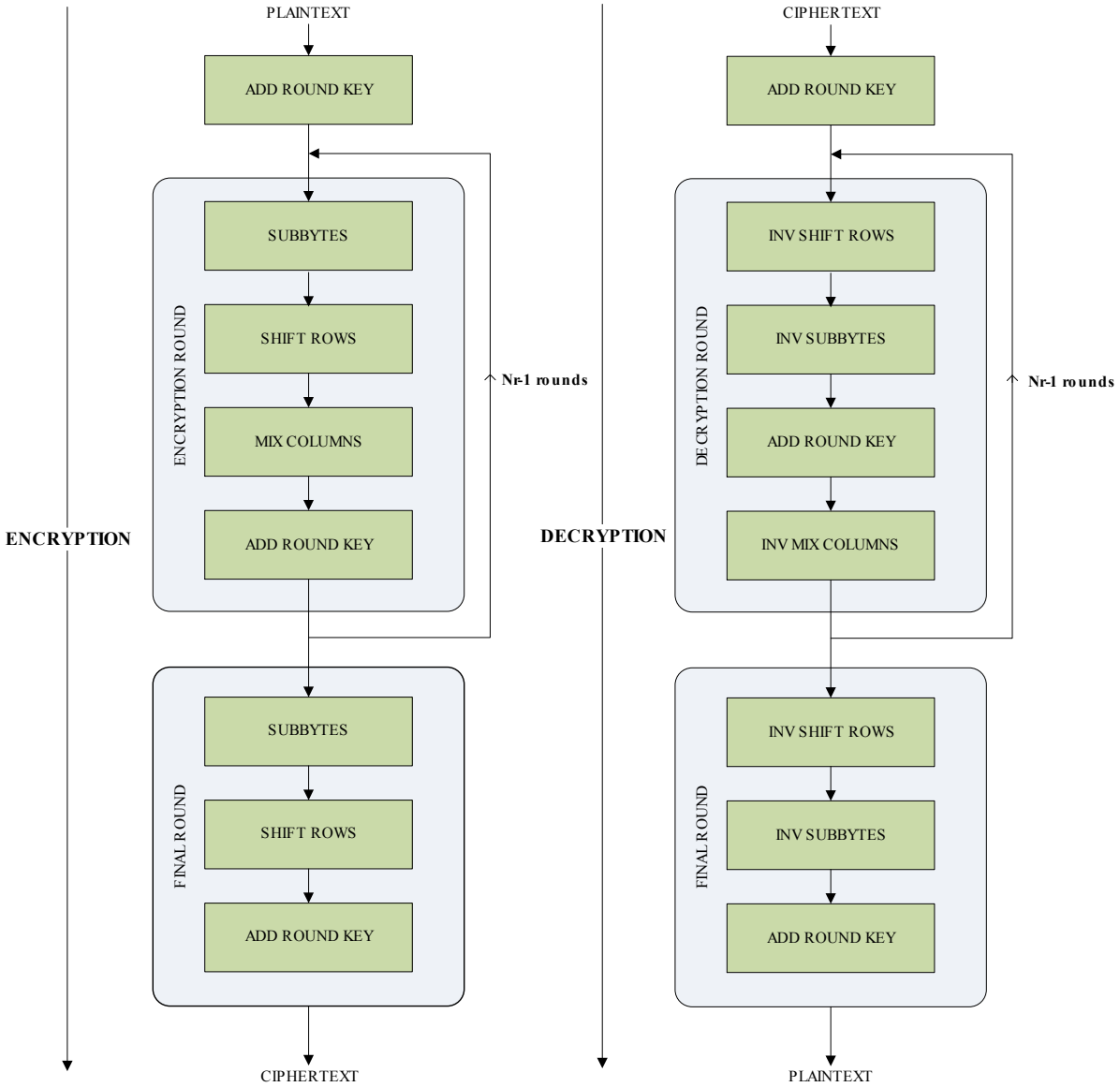
The AES works on a symmetric-key algorithm, meaning the same key is used for both encrypting and decrypting the data.

38.2. Features

- Compliant with FIPS Publication 197, Advanced Encryption Standard (AES)
- 128/192/256 bit cryptographic key supported
- Encryption time of 57/67/77 cycles with 128-bit/192-bit/256-bit cryptographic key
- Five confidentiality modes of operation as recommended in NIST Special Publication 800-38A
- Electronic Code Book (ECB)
- Cipher Block Chaining (CBC)
- Cipher Feedback (CFB)
- Output Feedback (OFB)
- Counter (CTR)
- Supports Counter with CBC-MAC (CCM/CCM*) mode for authenticated encryption
- 8, 16, 32, 64, 128-bit data sizes possible in CFB mode
- Optional (parameter) Galois Counter mode (GCM) encryption and authentication

38.3. Block Diagram

Figure 38-1. AES Block Diagram



38.4. Signal Description

Not applicable.

38.5. Product Dependencies

In order to use this AES module, other parts of the system must be configured correctly, as described below.

38.5.1. I/O Lines

Not applicable.

38.5.2. Power Management

The AES will continue to operate in any sleep mode, if it's source clock is running. The AES interrupts can be used to wake up the device from sleep modes. Refer to the Power Manager chapter for details on the different sleep modes.

AES is clocked only on the following conditions:

- Whenever there is an APB access for any read and write operation to the AES registers.
- When the AES is enabled & encryption/decryption is on.

38.5.3. Clocks

The AES bus clock (CLK_AES_APB) can be enabled and disabled in the Main Clock module, and the default state of CLK_AES_APB can be found in *Peripheral Clock Masking*. The module is fully clocked by CLK_AES_APB.

Related Links

Peripheral Clock Masking on page 151

38.5.4. DMA

The AES has two DMA request lines; one for input data, and one for output data. They are both connected to the DMA Controller (DMAC). These DMA request triggers will be acknowledged by the DMAC ACK signals. Using the AES DMA requests requires the DMA Controller to be configured first. Refer to the device DMA documentation.

38.5.5. Interrupts

The interrupt request line is connected to the interrupt controller. Using the AES interrupt requires the interrupt

controller to be configured first. Refer to the Processor and Architecture chapter for details.

All the AES interrupts are synchronous wake-up sources. See *Sleep Mode Controller* for details.

Related Links

Sleep Mode Controller on page 191

38.5.6. Events

Not applicable.

38.5.7. Debug Operation

When the CPU is halted in debug mode, the AES module continues normal operation. If the AES module is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar,

improper operation or data loss may result during debugging. The AES module can be forced to halt operation during debugging.

38.5.8. Register Access Protection

All registers with write-access are optionally write-protected by the peripheral access controller (PAC), except the

following register:

- Interrupt Flag Register (INTFLAG)

Write-protection is denoted by the Write-Protected property in the register description.

Write-protection does not apply to accesses through an external debugger. Refer to *PAC - Peripheral Access Controller* chapter for details.

Related Links

PAC - Peripheral Access Controller on page 58

38.5.9. Analog Connections

Not applicable.

38.6. Functional Description

38.6.1. Principle of Operation

The following is a high level description of the algorithm. These are the steps:

- KeyExpansion: Round keys are derived from the cipher key using Rijndael's key schedule.
- InitialRound:
  - AddRoundKey: Each byte of the state is combined with the round key using bitwise XOR.
- Rounds:
  - SubBytes: A non-linear substitution step where each byte is replaced with another according to a lookup table.
  - ShiftRows: A transposition step where each row of the state is shifted cyclically a certain number of steps.
  - MixColumns: A mixing operation which operates on the columns of the state, combining the four bytes in each column.
  - AddRoundKey
- Final Round (no MixColumns):
  - SubBytes
  - ShiftRows
  - AddRoundKey

The relationship between the module's clock frequency and throughput (in bytes per second) is given by:

Clock Frequency = (Throughput/2) x (Nr+1) for 2 byte parallel processing

Clock Frequency = (Throughput/4) x (Nr+1) for 4 byte parallel processing

where Nr is the number of rounds, depending on the key length.

38.6.2. Basic Operation

38.6.2.1. Initialization

The following register is enable-protected:

- Control A (CTRLA)

Enable-protection is denoted by the Enable-Protected property in the register description.

38.6.2.2. Enabling, Disabling, and Resetting

The AES module is enabled by writing a one to the Enable bit in the Control A register (CTRLA.ENABLE). The module is disabled by writing a zero to CTRLA.ENABLE. The module is reset by writing a one to the Software Reset bit in the Control A register (CTRLA.SWRST).

38.6.2.3. Basic Programming

The CIPHER bit in the Control A Register (CTRLA.CIPHER) allows selection between the encryption and the decryption processes. The AES is capable of using cryptographic keys of 128/192/256 bits to encrypt and decrypt data in blocks of 128 bits. The Key Size (128/192/256) can be programmed in the KEYSIZE field in the Control A Register (CTRLA.KEYSIZE). This 128-bit/192-bit/256-bit key is defined in the Key Word Registers (KEYWORDx). By setting the XORKEY bit of CTRLA register, keyword can be updated with the resulting XOR value of user keyword and previous keyword content.

The input data for processing is written to a data buffer consisting of four 32-bit registers through the Data register address. The data buffer register (note that input and output data shares the same data buffer register) that is written to when the next write is performed is indicated by the Data Pointer in the Data Buffer Pointer (DATABUFPTR) register. This field is incremented by one or wrapped by hardware when a write to the DATA register address is performed. This field can also be programmed, allowing the user direct control over which input buffer register to write to. Note that when AES module is in the CFB operation mode with the data segment size less than 128 bits, the input data must be written to the first (DATABUFPTR = 0) and/or second (DATABUFPTR = 1) input buffer registers (see Table 38-1 Relevant Input/Output Data Registers for different Confidentiality modes on page 865).

The input to the encryption processes of the CBC, CFB and OFB modes includes, in addition to the plaintext, a 128-bit data block called the Initialization Vector (IV), which must be set in the Initialization Vector Registers (INTVECTx). Additionally, the GCM mode 128-bit authentication data needs to be programmed. The Initialization Vector is used in the initial step in the encryption of a message and in the corresponding decryption of the message. The Initialization Vector Registers are also used by the Counter mode to set the counter value.

It is necessary to notify AES module whenever the next data block it is going to process is the beginning of a new message. This is done by writing a one to the New Message bit in the Control B register (CTRLB.NEWMSG).

The AES modes of operation are selected by setting the AESMODE field in the Control A Register (CTRLA.AESMODE). In Cipher Feedback Mode (CFB), five data sizes are possible (8, 16, 32, 64 or 128 bits), configurable by means of the CFBS field in the Control A Register (CTRLA.CFBS). In Counter mode, the size of the block counter embedded in the module is 16 bits. Therefore, there is a rollover after processing 1 megabyte of data. The data pre-processing, post-processing and data chaining for the concerned modes are automatically performed by the module.

When data processing has completed, the Encryption Complete bit in the Interrupt Flag register (INTFLAG.ENCCMP) is set by hardware (which triggers an interrupt request if the corresponding interrupt is enabled). The processed output data is read out through the Output Data register (DATA) address from the data buffer consisting of four 32-bit registers. The data buffer register that is read from when the next read is performed is indicated by the Data Pointer field in the Data Buffer Pointer register (DATABUFPTR). This field is incremented by one or wrapped by hardware when a read from the DATA

register address is performed. This field can also be programmed, giving the user direct control over which output buffer register to read from. Note that when AES module is in the CFB operation mode with the data segment size less than 128 bits, the output data must be read from the first (DATABUFPTR = 0) and/or second (DATABUFPTR = 1) output buffer registers (see see Table 38-1 Relevant Input/Output Data Registers for different Confidentiality modes on page 865). The Encryption Complete bit (INTFLAG.ENCCMP) is cleared by hardware after the processed data has been read from the relevant output buffer registers.

Table 38-1. Relevant Input/Output Data Registers for different Confidentiality modes

Confidentiality Mode	Relevant Input / Output Data Registers
ECB	All
CBC	All
OFB	All
128-bit CFB	All
64-bit CFB	First and Second
32-bit CFB	First
16-bit CFB	First
8-bit CFB	First
CTR	All

38.6.2.4. Start Modes

The Start mode field in the Control A Register (CTRLA.STARTMODE) allows the selection of encryption start mode.

1. Manual Start Mode
   In the Manual Start Mode the sequence is as follows:
   1.1. Write the 128/192/256 bit key in the Key Register (KEYWORDx)
   1.2. Write the initialization vector or counter in the Initialization Vector Register (INTVECTx). The initialization vector concerns all modes except ECB
   1.3. Enable interrupts in Interrupt Enable Set Register (INTENSET), depending on whether an interrupt is required or not at the end of processing.
   1.4. Write the data to be encrypted or decrypted in the Data Registers (DATA).
   1.5. Set the START bit in Control B Register (CTRLB.START) to begin the encryption or the decryption process.
   1.6. When the processing completes, the Encryption Complete bit in the Interrupt Flag Register (INTFLAG.ENCCMP) raises. If Encryption Complete interrupt has been enabled, the interrupt line of the AES is activated.
   1.7. When the software reads one of the Output Data Registers (DATA), INTFLAG.ENCCMP bit is automatically cleared.
2. Auto start Mode
   The Auto Start Mode is similar to the manual one, except in this mode, as soon as the correct number of input data registers is written, processing is automatically started without setting the START bit in the Control B Register. DMA operation uses this mode.
3. Last Output Data Mode (LOD)

This mode is used to generate message authentication code (MAC) on data in CCM mode of operation. The CCM mode combines counter mode for encryption and CBC-MAC generation for authentication.

When LOD is disabled in CCM mode then counter mode of encryption is performed on the input data block.

When LOD is enabled in CCM mode then CBC-MAC generation is performed. Zero block is used as the initialization vector by the hardware. Also software read from the Output Data Register (DATA) is not required to clear the ENCCMP flag. The ENCCMP flag is automatically cleared by writing into the Input Data Register (DATA). This allows retrieval of only the last data in several encryption/decryption processes. No output data register reads are necessary between each block of encryption/decryption process.

Note that assembling message depending on the security level identifier in CCM* has to be done in software.

38.6.2.5. Computation of last Nk words of expanded key

The AES algorithm takes the cryptographic key provided by the user and performs a Key Expansion routine to generate an expanded key. The expanded key contains a total of 4(Nr + 1) 32-bit words, where the first Nk (4/6/8 for a 128-/192-/256-bit key) words are the user-provided key. For data encryption, the expanded key is used in the forward direction, i.e., the first four words are used in the initial round of data processing, the second four words in the first round, the third four words in the second round, and so on. On the other hand, for data decryption, the expanded key is used in the reverse direction, i.e.,the last four words are used in the initial round of data processing, the last second four words in the first round, the last third four words in the second round, and so on.

To reduce gate count, the AES module does not generate and store the entire expanded key prior to data processing. Instead, it computes on-the-fly the round key (four 32-bit words) required for the current round of data processing. In general, the round key for the current round of data processing can be computed from the Nk words of the expanded key generated in the previous rounds. When AES module is operating in the encryption mode, the round key for the initial round of data processing is simply the user-provided key written to the KEY registers. On the other hand, when AES module is operating in the decryption mode, the round key for the initial round of data processing is the last four words of the expanded key, which is not available unless AES module has performed at least one encryption process prior to operating in the decryption mode.

In general, the last Nk words of the expanded key must be available before decryption can start. If desired, AES module can be instructed to compute the last Nk words of the expanded key in advance by writing a one to the Key Generate (KEYGEN) bit in the CTRLA register (CTRLA.KEYGEN). The computation takes Nr clock cycles. Alternatively, the last Nk words of the expanded key can be automatically computed by AES module when a decryption process is initiated if they have not been computed in advance or have become invalid. Note that this will introduce a latency of Nr clock cycles to the first decryption process.

38.6.2.6. Hardware Countermeasures against Differential Power Analysis Attacks

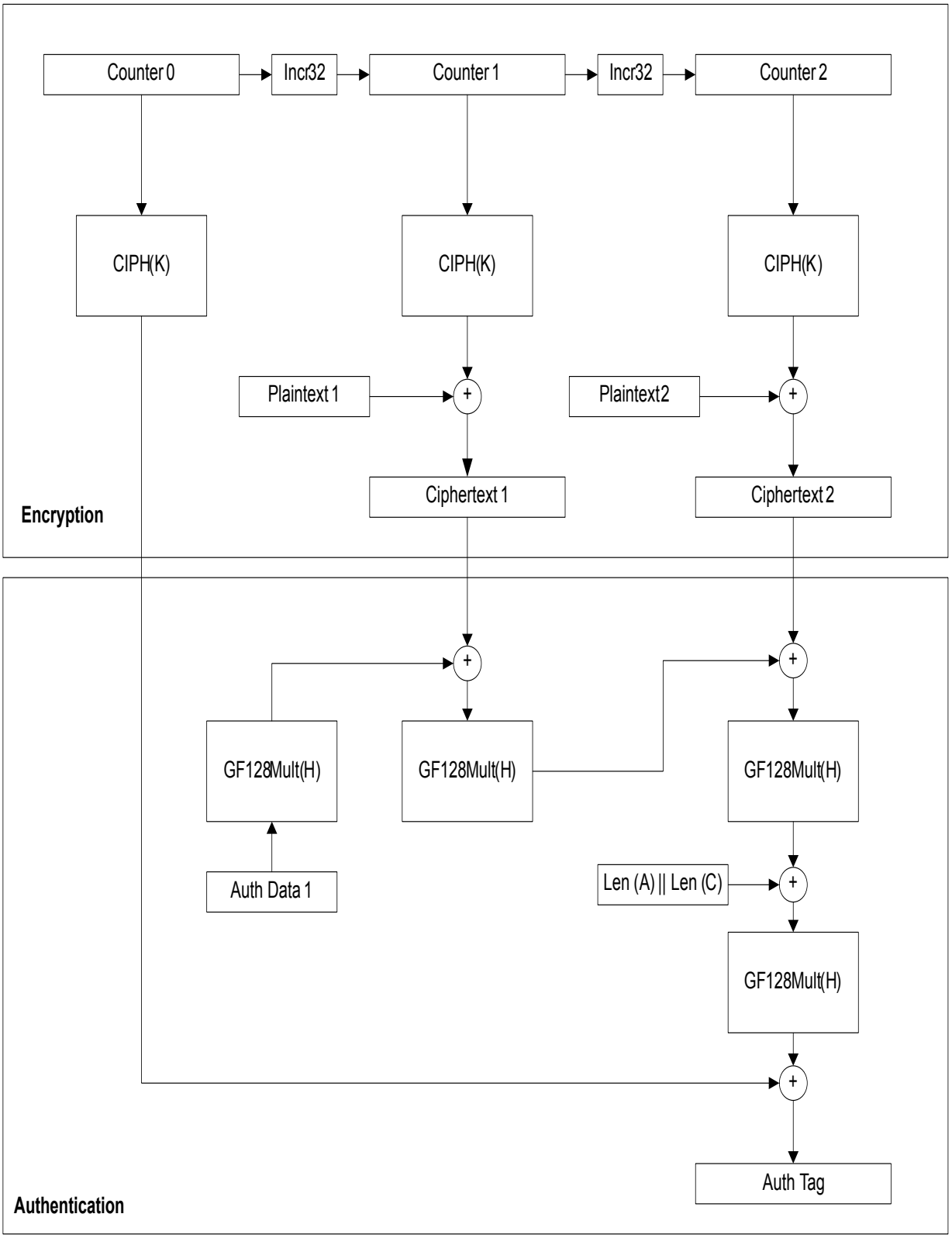
The AES module features four types of hardware countermeasures that are useful for protecting data against differential power analysis attacks:

- Type 1: Randomly add one cycle to data processing
- Type 2: Randomly add one cycle to data processing (other version)
- Type 3: Add a random number of clock cycles to data processing, subject to a maximum of 11/13/15 clock cycles for key sizes of 128/192/256 bits
- Type 4: Add random spurious power consumption during data processing

By default, all countermeasures are enabled. One or more of the countermeasures can be disabled by programming the Countermeasure Type field in the Control A (CTRLA.CTYPE) register. The countermeasures use random numbers generated by a deterministic random number generator embedded in AES module. The seed for the random number generator is written to the RANDSEED register. Note also that a new seed must be written after a change in the keysize. Note that enabling countermeasures reduces AES module's throughput. In short, the throughput is highest with all the countermeasures disabled. On the other hand, with all of the countermeasures enabled, the best protection is achieved but the throughput is worst.

38.6.3. Galois Counter Mode (GCM)

GCM is comprised of the AES engine in CTR mode along with a universal hash function (GHASH engine) that is defined over a binary Galois field to produce a message authentication tag. The GHASH engine processes data packets after the AES operation. GCM provides assurance of the confidentiality of data through the AES Counter mode of operation for encryption. Authenticity of the confidential data is assured through the GHASH engine. Refer to the NIST Special Publication 800-38D Recommendation for more complete information.


Counter 0
Incr32
Counter 1
Incr32
Counter 2
CIPH(K)
CIPH(K)
CIPH(K)
Plaintext 1
Plaintext2
Ciphertext 1
Ciphertext 2
Encryption
GF128Mult(H)
GF128Mult(H)
GF128Mult(H)
Auth Data 1
Len (A) || Len (C)
GF128Mult(H)
Auth Tag
Authentication

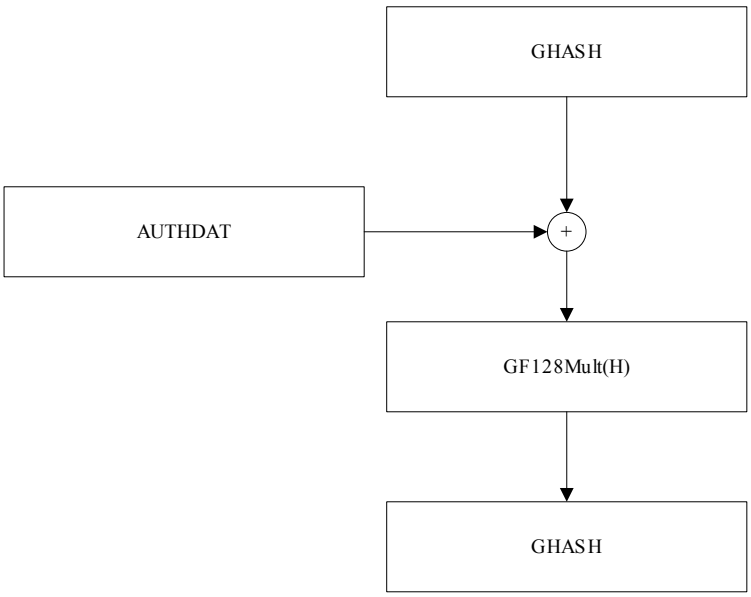
38.6.3.1. GCM Operation

Hashkey Generation

- Configure CTRLA register as follows:
  1.1. CTRLA.STARTMODE as Manual (Auto for DMAC)
  1.2. CTRLA.CIPHER as Encryption
  1.3. CTRLA.KEYSIZE as per the key used
  1.4. CTRLA.AESMODE as ECB
  1.5. CTRLA.CTYPE as per the countermeasures required.
- Set CTRLA.ENABLE
- Write zero to CIPLEN reg.
- Write the key in KEYWORDx register
- Write the zeros to DATA reg
- Set CTRLB.Start.
- Wait for INTFLAG.ENCCMP to be set
- AES Hardware generates Hash Subkey in HASHKEYx register.

Authentication Header Processing

- Configure CTRLA register as follows:
  1.1. CTRLA.STARTMODE as Manual
  1.2. CTRLA.CIPHER as Encryption
  1.3. CTRLA.KEYSIZE as per the key used
  1.4. CTRLA.AESMODE as GCM
  1.5. CTRLA.CTYPE as per the countermeasures required.
- Set CTRLA.ENABLE
- Write the key in KEYWORDx register
- Set CTRLB.GFMUL
- Write the Authdata to DATA reg
- Set CTRLB.START as1
- Wait for INTFLAG.GFMCMP to be set.
- AES Hardware generates output in GHASHx register
- Continue steps 4 to 7 for remaining Authentication Header.
  Note: If the Auth data is less than 128 bit, it has to be padded with zero to make it 128 bit aligned.



Plain text Processing

- Set CTRLB.NEWMSG for the new set of plain text processing.
- Load CIPLEN reg.
- Load (J0+1) in INTVECTx register.
- As described in NIST documentation J 0 = IV || 0 31 || 1 when len(IV)=96 and J0 =$GHASH_H$ (IV || 0 s+64 || [len(IV)] 64) (s is the minimum numer of zeroes that should be padded with the Initialization Vector to make it a multiple of 128) if len(IV) != 96.
- Load plain text in DATA register.
- Set CTRLB.START as 1.
- Wait for INTFLAG.ENCCMP to be set.
- AES Hardware generates output in DATA register.
- Intermediate GHASH is stored in GHASHx register and Cipher Text available in DATA register.
- Continue 3 to 6 till the input of plain text to get the cipher text and the Hash keys.
- At the last input, set CTRLB.EOM.
- Write last indata to DATA reg.
- Set CTRLB.START as 1.
- Wait for INTFLAG.ENCCMP to be set.
- AES Hardware generates output in DATA register and final Hash key in GHASHx register.
- Load [LEN(A)]64||[LEN(C)]64 in DATA register and set CTRLB.GFMUL and CTRLB.START as 1.
- Wait for INTFLAG.GFMCMP to be set.
- AES Hardware generates final GHASH value in GHASHx register.

Plain text processing with DMAC

- Set CTRLB.NEWMSG for the new set of plain text processing.
- Load CIPLEN reg.
- Load (J0+1) in INTVECTx register.
- Load plain text in DATA register.
- Wait for INTFLAG.ENCCMP to be set.
- AES Hardware generates output in DATA register.

- Intermediate GHASH is stored in GHASHx register and Cipher Text available in DATA register.
- Continue 3 to 5 till the input of plain text to get the cipher text and the Hash keys.
- At the last input, set CTRLB.EOM.
- Write last indata to DATA reg.
- Wait for INTFLAG.ENCCMP to be set.
- AES Hardware generates output in DATA register and final Hash key in GHASHx register.
- Load [LEN(A)]64||[LEN(C)]64 in DATA register and set CTRLB.GFMUL and CTRLB.START as 1.
- Wait for INTFLAG.GFMCMP to be set.
- AES Hardware generates final GHASH value in GHASHx register.

Tag Generation

- Configure CTRLA
  1.1. Set CTRLA.ENABLE to 0
  1.2. Set CTRLA.AESMODE as CTR
  1.3. Set CTRLA.ENABLE to 1
- Load J0 value to INITVECTVx reg.
- Load GHASHx value to DATA reg.
- Set CTRLB.NEWMSG and CTRLB.START to start the Counter mode operation.
- Wait for INTFLAG.ENCCMP to be set.
- AES Hardware generates the GCM Tag output in DATA register.

38.6.4. Synchronization

Not applicable.

38.7. Register Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0	CFBS[2:0]			AESMODE[2:0]			ENABLE	SWRST
0x01		15:8		XORKEY	KEYGEN		STARTMODE	CIPHER	KEYSIZE[1:0]	
0x02		23:16			CTYPE[3:0]					
0x3		31:24								
0x04	CTRLB	7:0						EOM	NEWMSG	START
0x05	INTENCLR	7:0							GFMCMP	ENCCMP
0x06	INTENSET	7:0							GFMCMP	ENCCMP
0x07	INTENFLAG	7:0							GFMCMP	ENCCMP
0x08	DATABUFPTR	7:0							DATAPTR[1:0]	
0x09	DBGCTRL									DBGRUN
0x0A,0x0B	Reserved									
0x0C, 0x10,0x14, 0x18,0x1C, 0x20,0x24, 0x28	KEYWORDx	7:0	KEYWORDx[7:0]							
		15:8	KEYWORDx[15:8]							
		23:16	KEYWORDx[23:16]							
		31:24	KEYWORDx[31:24]							

Offset	Name	Bit Pos.								
0x38	DATA	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x3C, 0x40,0x44, 0x48	INTVECTx	7:0	INTVECTx[7:0]							
		15:8	INTVECTx[15:8]							
		23:16	INTVECTx[23:16]							
		31:24	INTVECTx[31:24]							
0x4C, 0x50,0x54, 0x58	HASHKEYx	7:0	HASHKEYx[7:0]							
		15:8	HASHKEYx[15:8]							
		23:16	HASHKEYx[23:16]							
		31:24	HASHKEYx[31:24]							
0x5C, 0x60,0x64, 0x68	GHASHx	7:0	GHASHx[7:0]							
		15:8	GHASHx[15:8]							
		23:16	GHASHx[23:16]							
		31:24	GHASHx[31:24]							
0x70	CIPLEN	7:0	CIPLEN[7:0]							
		15:8	CIPLEN[15:8]							
		23:16	CIPLEN[23:16]							
		31:24	CIPLEN[31:24]							
0x74	RANDSEED	7:0	RANDSEED[7:0]							
		15:8	RANDSEED[15:8]							
		23:16	RANDSEED[23:16]							
		31:24	RANDSEED[31:24]							

38.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection on page 863.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

38.8.1. Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Enable-protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					CTYPE[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
		XORKEY	KEYGEN	LOD	STARTMODE	CIPHER	KEYSIZE[1:0]	
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CFBS[2:0]			AESMODE[2:0]			ENABLE	SWRST
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 19:16 – CTYPE[3:0]: Countermeasure type

Value	Name	Description
XXX0	CTYPE1 disabled	Countermeasure1 disabled
XXX1	CTYPE1 enabled	Countermeasure1 enabled
XX0X	CTYPE2 disabled	Countermeasure2 disabled
XX1X	CTYPE2 enabled	Countermeasure2 enabled
X0XX	CTYPE3 disabled	Countermeasure3 disabled
X1XX	CTYPE3 enabled	Countermeasure3 enabled
0XXX	CTYPE4 disabled	Countermeasure4 disabled
1XXX	CTYPE4 enabled	Countermeasure4 enabled

Bit 14 – XORKEY: XOR Key

Value	Description
0	No effect
1	The user keyword gets XORed with the previous keyword register content.

Bit 13 – KEYGEN: Key Generation

Value	Description
0	No effect
1	Start Computation of the last NK words of the expanded key

Bit 12 – LOD: Last Output Data Mode

Value	Description
0	No effect
1	Start encryption in Last Output Data mode

Bit 11 – STARTMODE: Start Mode Select

Value	Name	Description
0	Manual Mode	Start Encryption / Decryption in Manual mode
1	Auto Mode	Start Encryption / Decryption in Auto mode

Bit 10 – CIPHER: Encryption/ Decryption

Value	Description
0	Decryption
1	Encryption

Bits 9:8 – KEYSIZE[1:0]: Encryption Key Size

Value	Name	Description
0	128-bit Key	128-bit Key for Encryption / Decryption
1	192-bit Key	192-bit Key for Encryption / Decryption
2	256-bit Key	256-bit Key for Encryption / Decryption
3	Reserved	Reserved

Bits 7:5 – CFBS[2:0]: Cipher Feedback Block Size

Value	Name	Description
0	128-bit data block	128-bit Input data block for Encryption/Decryption in Cipher Feedback mode
1	64-bit data block	64-bit Input data block for Encryption/Decryption in Cipher Feedback mode
2	32-bit data block	32-bit Input data block for Encryption/Decryption in Cipher Feedback mode
3	16-bit data block	16-bit Input data block for Encryption/Decryption in Cipher Feedback mode
4	8-bit data block	8-bit Input data block for Encryption/Decryption in Cipher Feedback mode

Value	Name	Description
5-7	Reserved	Reserved

Bits 4:2 – AESMODE[2:0]: AES Modes of Operation

Value	Name	Description
0	ECB	Electronic code book mode
1	CBC	Cipher block chaining mode
2	OFB	Output feedback mode
3	CFB	Cipher feedback mode
4	Counter	Counter mode
5	CCM	CCM mode
6	GCM	Galois counter mode
7	Reserved	Reserved

Bit 1 – ENABLE: Enable

Value	Description
0	The peripheral is disabled
1	The peripheral is enabled

Bit 0 – SWRST: Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the AES module to their initial state, and the module will be disabled.

Writing a '1' to `SWRST` will always take precedence, meaning that all other writes in the same write operation will be discarded.

Value	Description
0	There is no reset operation ongoing
1	The reset operation is ongoing

38.8.2. Control B

Name: CTRLB
Offset: 0x04
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
					GFMUL	EOM	NEWMSG	START
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 3 – GFMUL: GF Multiplication
This bit is applicable only to GCM mode.

Value	Description
0	No action
1	Setting this bit calculates GF multiplication with data buffer content and hashkey register content.

Bit 2 – EOM: End of Message
This bit is applicable only to GCM mode.

Value	Description
0	No action
1	Setting this bit generates final GHASH value for the message.

Bit 1 – NEWMSG: New Message
This bit is used in cipher block chaining (CBC), cipher feedback (CFB) and output feedback (OFB), counter (CTR) modes to indicate the hardware to use Initialization vector for encrypting the first block of message.

Value	Description
0	No action
1	Setting this bit indicates start of new message to the module.

Bit 0 – START: Start Encryption/Decryption

Value	Description
0	No action
1	Start encryption / decryption in manual mode.

38.8.3. Interrupt Enable Clear

Name: INTENCLR
Offset: 0x05
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
							GFMCMP	ENCCMP
Access							R/W	R/W
Reset							0	0

Bit 1 – GFMCMP: GF Multiplication Complete Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the GF Multiplication Complete Interrupt Enable bit, which disables the GF Multiplication Complete interrupt.

Value	Description
0	The GF Multiplication Complete interrupt is disabled.
1	The GF Multiplication Complete interrupt is enabled.

Bit 0 – ENCCMP: Encryption Complete Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Encryption Complete Interrupt Enable bit, which disables the Encryption Complete interrupt.

Value	Description
0	The Encryption Complete interrupt is disabled.
1	The Encryption Complete interrupt is enabled.

38.8.4. Interrupt Enable Set

Name: INTENSET
Offset: 0x06
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
							GFMCMP	ENCCMP
Access							R/W	R/W
Reset							0	0

Bit 1 – GFMCMP: GF Multiplication Complete Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the GF Multiplication Complete Interrupt Enable bit, which enables the GF Multiplication Complete interrupt.

Value	Description
0	The GF Multiplication Complete interrupt is disabled.
1	The GF Multiplication Complete interrupt is enabled.

Bit 0 – ENCCMP: Encryption Complete Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Encryption Complete Interrupt Enable bit, which enables the Encryption Complete interrupt.

Value	Description
0	The Encryption Complete interrupt is disabled.
1	The Encryption Complete interrupt is enabled.

38.8.5. Interrupt Flag Status and Clear

Name: INTENFLAG
Offset: 0x07
Reset: 0x00
Property:

Bit	7	6	5	4	3	2	1	0
							GFMCMP	ENCCMP
Access							R/W	R/W
Reset							0	0

Bit 1 – GFMCMP: GF Multiplication Complete
This flag is cleared by writing a '1' to it.

This flag is set when GHASH value is available on the Galois Hash Registers (GHASHx) in GCM mode.

Writing a '0' to this bit has no effect.

This flag is also automatically cleared in the following cases.

1. Manual encryption/decryption occurs (START in CTRLB register).
2. Reading from the GHASHx register.

Bit 0 – ENCCMP: Encryption Complete
This flag is cleared by writing a '1' to it.

This flag is set when encryption/decryption is complete and valid data is available on the Data Register.

Writing a '0' to this bit has no effect.

This flag is also automatically cleared in the following cases:

1. Manual encryption/decryption occurs (START in CTRLA register). (This feature is needed only if we do not support double buffering of DATA registers).
2. Reading from the data register (DATAx) when LOD = 0.
3. Writing into the data register (DATAx) when LOD = 1.
4. Reading from the Hash Key register (HASHKEYx).

38.8.6. Data Buffer Pointer

Name: DATABUFPTR
Offset: 0x94
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
							DATAPTR[1:0]	
Access							R/W	R/W
Reset							0	0

Bits 1:0 – DATAPTR[1:0]: Data Pointer
Writing to this field changes the value of the data pointer, which determines which of the four data registers is written to/read from when the next write/read to the `DATA` register address is performed.

38.8.7. Debug

Name: DBGCTRL
Offset: 0x6C
Reset: 0x00
Property: PAC Write-Protection, Enable-protected

Bit	7	6	5	4	3	2	1	0
								DBGRUN
Access								W
Reset								0

Bit 0 – DBGRUN: Debug Run
Writing a '0' to this bit causes the AES to halt during debug mode.
Writing a '1' to this bit allows the AES to continue normal operation during debug mode. This bit can only be changed while the AES is disabled.

38.8.8. Keywordx

Name: KEYWORDx
Offset: 0X0C,0X10,0X14,0X18,0X1C,0X20,0X24,0X28
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	KEYWORD0[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	KEYWORD0[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	KEYWORD0[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	KEYWORD0[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – KEYWORDx: Key Word x Value
The four/six/eight 32-bit Key Word registers set the 128-bit/192-bit/256-bit cryptographic key used for encryption/decryption. `KEYWORD0` corresponds to the first word of the key and `KEYWORD3/KEYWORD5/KEYWORD7` to the last one.

Note: By setting the XORKEY bit of CTRLA register, keyword will update with the resulting XOR value of user keyword and previous keyword content.

38.8.9. Data

Name: DATA
Offset: 0X38
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	DATA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DATA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DATA[31:0]: Data Value
A write to or read from this register corresponds to a write to or read from one of the four data registers. The four 32-bit Data registers set the 128-bit data block used for encryption/decryption. The data register that is written to or read from is given by the `DATABUFPTR.DATPTR` field.

Note: Both input and output shares the same data buffer. Reading DATA register will return 0's when AES is performing encryption or decryption operation.

38.8.10. Initialization Vector x Register

Name: INTVECTx
Offset: 0X3C,0X40,0X44,0x48
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	INTVECT0[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	INTVECT0[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	INTVECT0[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	INTVECT0[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – INTVECTx: Initialization Vector x Value
The four 32-bit Initialization Vector registers set the 128-bit Initialization Vector data block that is used by some modes of operation as an additional initial input. `INTVECT0` corresponds to the first word of the Initialization Vector, `INTVECT3` to the last one. These registers are write-only to prevent the Initialization Vector from being read by another application. For CBC, OFB, and CFB modes, the Initialization Vector corresponds to the initialization vector. For CTR mode, it corresponds to the counter value.

38.8.11. Hash Key x (GCM mode only)

Name: HASHKEYx
Offset: 0X4C,0X50,0X54,0X58
Reset: 0x00000000
Property: PAC Write-protection

Bit	31	30	29	28	27	26	25	24
	HASHKEY0[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	HASHKEY0[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	HASHKEY0[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	HASHKEY0[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – HASHKEYx: Hash Key x Value
The four 32-bit `HASHKEY` registers contain the 128-bit Hash Key value computed from the AES KEY. The Hash Key value can also be programmed offering single GF128 multiplication possibilities.

38.8.12. Galois Hash x (GCM mode only)

Name: GHASHx
Offset: 0X5C,0X60,0X64,0X68
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	GHASH0[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	GHASH0[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	GHASH0[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	GHASH0[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – GHASHx: Galois Hash x Value
The four 32-bit Hash Word registers contain the `GHASH` value after GF128 multiplication in GCM mode. Writing a new key to `KEYWORDx` registers causes `GHASHx` to be initialized with zeroes. These registers can also be programmed.

38.8.13. Galois Hash x (GCM mode only)

Name: CIPLEN
Offset: 0X70
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	CIPLEN[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CIPLEN[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CIPLEN[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CIPLEN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CIPLEN[31:0]: Cipher Length
This register contains the length in bytes of the Cipher text that is to be processed. This is programmed by the user in GCM mode for Tag generation.

38.8.14. Random Seed

Name: RANDSEED
Offset: 0X74
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	RANDSEED[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RANDSEED[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RANDSEED[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RANDSEED[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – RANDSEED[31:0]: Random Seed
A write to this register corresponds to loading a new seed into the Random number generator.

39. USB – Universal Serial Bus

39.1. Overview

The Universal Serial Bus interface (USB) module complies with the Universal Serial Bus (USB) 2.1 specification supporting device modes.

The USB device mode supports 8 endpoint addresses. All endpoint addresses have one input and one output endpoint, for a total of 16 endpoints. Each endpoint is fully configurable in any of the four transfer types: control, interrupt, bulk or isochronous. The maximum data payload size is selectable up to 1023 bytes.

Internal SRAM is used to keep the configuration and data buffer for each endpoint. The memory locations used for the endpoint configurations and data buffers is fully configurable. The amount of memory allocated is dynamic according to the number of endpoints in use, and the configuration of these. The USB module has a built-in Direct Memory Access (DMA) and will read/write data from/to the system RAM when a USB transaction takes place. No CPU or DMA Controller resources are required.

To maximize throughput, an endpoint can be configured for ping-pong operation. When this is done the input and output endpoint with the same address are used in the same direction. The CPU or DMA Controller can then read/write one data buffer while the USB module writes/reads from the other buffer. This gives double buffered communication.

Multi-packet transfer enables a data payload exceeding the maximum packet size of an endpoint to be transferred as multiple packets without any software intervention. This reduces the number of interrupts and software intervention needed for USB transfers.

For low power operation the USB module can put the microcontroller in any sleep mode when the USB bus is idle and a suspend condition is given. Upon bus resume, the USB module can wake the microcontroller from any sleep mode.

39.2. Features

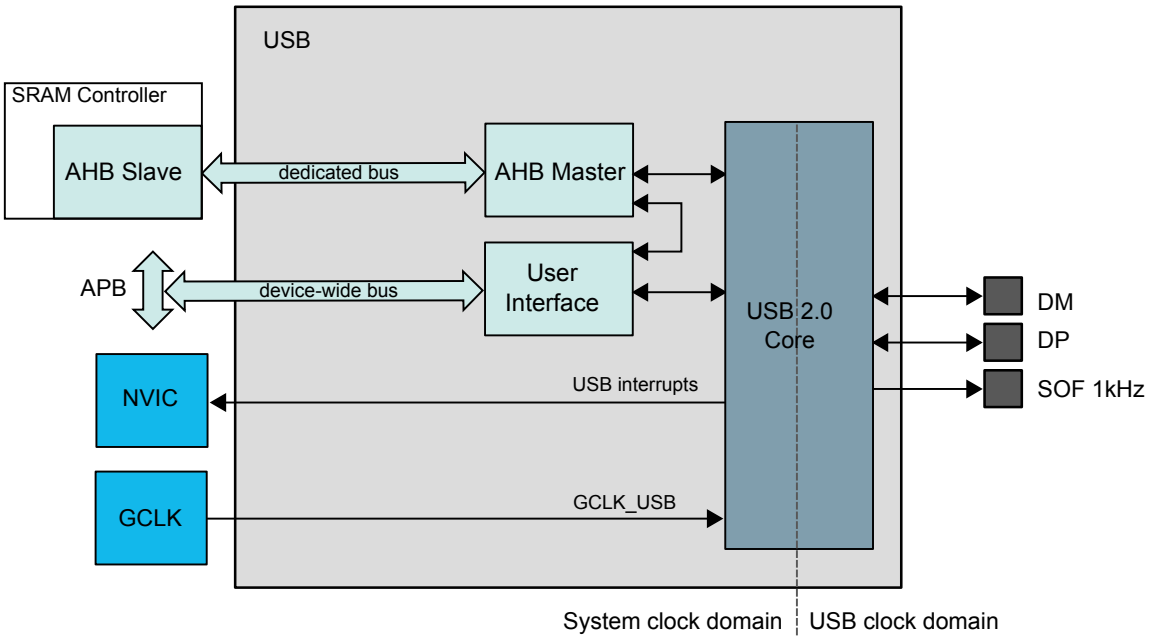
- Compatible with the USB 2.1 specification
- USB Embedded Host and Device mode
- Supports full (12Mbit/s) and low (1.5Mbit/s) speed communication
- Supports Link Power Management (LPM-L1) protocol
- On-chip transceivers with built-in pull-ups and pull-downs
- On-Chip USB serial resistors
- 1kHz SOF clock available on external pin
- Device mode
  - Supports 8 IN endpoints and 8 OUT endpoints
  - No endpoint size limitations
  - Built-in DMA with multi-packet and dual bank for all endpoints
  - Supports feedback endpoint
  - Supports crystal less clock
- Host mode
  - Supports 8 physical pipes
  - No pipe size limitations

- Supports multiplexed virtual pipe on one physical pipe to allow an unlimited USB tree
- Built-in DMA with multi-packet support and dual bank for all pipes
- Supports feedback endpoint
- Supports the USB 2.0 Phase-locked SOFs feature

39.3. USB Block Diagram

Figure 39-1. High-speed Implementation: USB Block Diagram

LS/FS Implementation: USB Block Diagram



39.4. Signal Description

Pin Name	Pin Description	Type
DM	Data -: Differential Data Line - Port	Input/Output
DP	Data +: Differential Data Line + Port	Input/Output
SOF 1kHZ	SOF Output	Output

Refer to *I/O Multiplexing and Considerations* for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

Related Links

I/O Multiplexing and Considerations on page 29

39.5. Product Dependencies

In order to use this peripheral module, other parts of the system must be configured correctly, as described below.

39.5.1. I/O Lines

The USB pins may be multiplexed with the I/O lines Controller. The user must first configure the I/O Controller to assign the USB pins to their peripheral functions.

A 1kHz SOF clock is available on an external pin. The user must first configure the I/O Controller to assign the 1kHz SOF clock to the peripheral function. The SOF clock is available for device and host mode.

39.5.2. Power Management

This peripheral can continue to operate in any sleep mode where its source clock is running. The interrupts can wake up the device from sleep modes. Events connected to the event system can trigger other operations in the system without exiting sleep modes. Refer to *PM – Power Manager* for details on the different sleep modes.

Related Links

PM – Power Manager on page 186

39.5.3. Clocks

The USB bus clock (CLK_USB_AHB) can be enabled and disabled in the Power Manager, and the default state of CLK_USB_AHB can be found in the *Peripheral Clock Masking*.

A generic clock (GCLK_USB) is required to clock the USB. This clock must be configured and enabled in the Generic Clock Controller before using the USB. Refer to *GCLK - Generic Clock Controller* for further details.

This generic clock is asynchronous to the bus clock (CLK_USB_AHB). Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains. Refer to *GCLK Synchronization* for further details.

The USB module requires a GCLK_USB of 48 MHz ± 0.25% clock for low speed and full speed operation. To follow the USB data rate at 12Mbit/s in full-speed mode, the CLK_USB_AHB clock should be at minimum 8MHz.

Clock recovery is achieved by a digital phase-locked loop in the USB module, which complies with the USB jitter specifications. If crystal-less operation is used in USB device mode, refer to *USB Clock Recovery Module*.

Related Links

GCLK - Generic Clock Controller on page 131
Synchronization on page 138
USB Clock Recovery Module on page 224

39.5.4. DMA

The USB has a built-in Direct Memory Access (DMA) and will read/write data to/from the system RAM when a USB transaction takes place. No CPU or DMA Controller resources are required.

39.5.5. Interrupts

The interrupt request line is connected to the Interrupt Controller. In order to use interrupt requests of this peripheral, the Interrupt Controller (NVIC) must be configured first. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

Nested Vector Interrupt Controller on page 51

39.5.6. Events

Not applicable.

39.5.7. Debug Operation

When the CPU is halted in debug mode the USB peripheral continues normal operation. If the USB peripheral is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

39.5.8. Register Access Protection

Registers with write-access can be optionally write-protected by the Peripheral Access Controller (PAC), except the following:

- Device Interrupt Flag (INTFLAG) register
- Endpoint Interrupt Flag (EPINTFLAG) register
- Host Interrupt Flag (INTFLAG) register
- Pipe Interrupt Flag (PINTFLAG) register

Note: Optional write-protection is indicated by the "PAC Write-Protection" property in the register description.

When the CPU is halted in debug mode, all write-protection is automatically disabled. Write-protection does not apply for accesses through an external debugger.

39.5.9. Analog Connections

Not applicable.

39.5.10. Calibration

The output drivers for the DP/DM USB line interface can be fine tuned with calibration values from production tests. The calibration values must be loaded from the NVM Software Calibration Area into the USB Pad Calibration register (PADCAL) by software, before enabling the USB, to achieve the specified accuracy. Refer to *NVM Software Calibration Area Mapping* for further details.

For details on Pad Calibration, refer to Pad Calibration (PADCAL on page 921) register.

Related Links

NVM Software Calibration Area Mapping on page 47

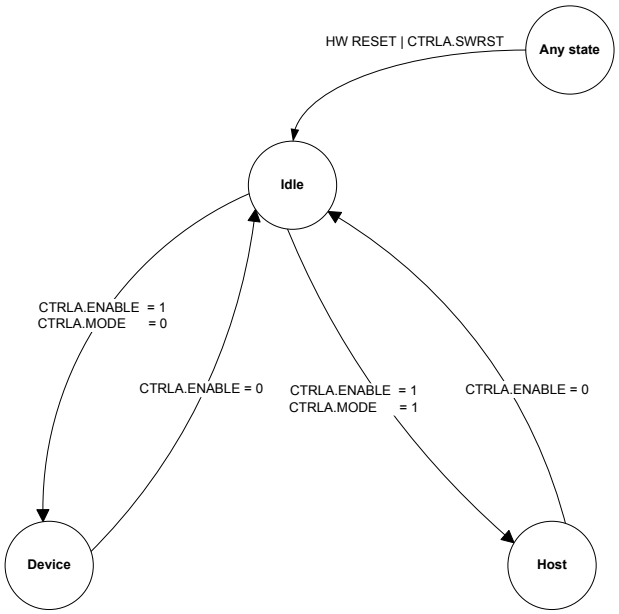
39.6. Functional Description

39.6.1. USB General Operation

39.6.1.1. Initialization

After a hardware reset, the USB is disabled. The user should first enable the USB (CTRLA.ENABLE) in either device mode or host mode (CTRLA.MODE).

Figure 39-2. General States



After a hardware reset, the USB is in the idle state. In this state:

- The module is disabled. The USB Enable bit in the Control A register (CTRLA.ENABLE) is reset.
- The module clock is stopped in order to minimize power consumption.
- The USB pad is in suspend mode.
- The internal states and registers of the device and host are reset.

Before using the USB, the Pad Calibration register (PADCAL) must be loaded with production calibration values from the NVM Software Calibration Area.

The USB is enabled by writing a '1' to CTRLA.ENABLE. The USB is disabled by writing a '0' to CTRLA.ENABLE.

The USB is reset by writing a '1' to the Software Reset bit in CTRLA (CTRLA.SWRST). All registers in the USB will be reset to their initial state, and the USB will be disabled. Refer to the CTRLA register for details.

The user can configure pads and speed before enabling the USB by writing to the Operating Mode bit in the Control A register (CTRLA.MODE) and the Speed Configuration field in the Control B register (CTRLB.SPDCONF). These values are taken into account once the USB has been enabled by writing a '1' to CTRLA.ENABLE.

After writing a '1' to CTRLA.ENABLE, the USB enters device mode or host mode (according to CTRLA.MODE).

The USB can be disabled at any time by writing a '0' to CTRLA.ENABLE.

Refer to USB Device Operations on page 894 for the basic operation of the device mode.

Refer to Host Operations on page 904 for the basic operation of the host mode.

Related Links

NVM Software Calibration Area Mapping on page 47

39.6.2. USB Device Operations

This section gives an overview of the USB module device operation during normal transactions. For more details on general USB and USB protocol, refer to the Universal Serial Bus specification revision 2.1.

39.6.2.1. Initialization

To attach the USB device to start the USB communications from the USB host, a zero should be written to the Detach bit in the Device Control B register (CTRLB.DETACH). To detach the device from the USB host, a one must be written to the CTRLB.DETACH.

After the device is attached, the host will request the USB device descriptor using the default device address zero. On successful transmission, it will send a USB reset. After that, it sends an address to be configured for the device. All further transactions will be directed to this device address. This address should be configured in the Device Address field in the Device Address register (DADD.DADD) and the Address Enable bit in DADD (DADD.ADDEN) should be written to one to accept communications directed to this address. DADD.ADDEN is automatically cleared on receiving a USB reset.

39.6.2.2. Endpoint Configuration

Endpoint data can be placed anywhere in the device RAM. The USB controller accesses these endpoints directly through the AHB master (built-in DMA) with the help of the endpoint descriptors. The base address of the endpoint descriptors needs to be written in the Descriptor Address register (DESCADD) by the user. Refer also to the Endpoint Descriptor structure in Endpoint Descriptor Structure on page 947.

Before using an endpoint, the user should configure the direction and type of the endpoint in Type of Endpoint field in the Device Endpoint Configuration register (EPCFG.EPTYPE0/1). The endpoint descriptor registers should be initialized to known values before using the endpoint, so that the USB controller does not read random values from the RAM.

The Endpoint Size field in the Packet Size register (PCKSIZE.SIZE) should be configured as per the size reported to the host for that endpoint. The Address of Data Buffer register (ADDR) should be set to the data buffer used for endpoint transfers.

The RAM Access Interrupt bit in Device Interrupt Flag register (INTFLAG.RAMACER) is set when a RAM access underflow error occurs during IN data stage.

When an endpoint is disabled, the following registers are cleared for that endpoint:

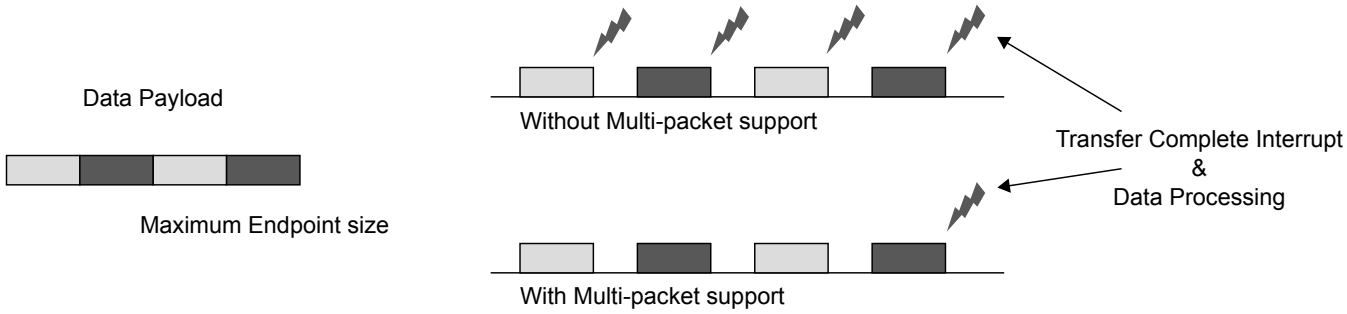
- Device Endpoint Interrupt Enable Clear/Set (EPINTENCLR/SET) register
- Device Endpoint Interrupt Flag (EPINTFLAG) register
- Transmit Stall 0 bit in the Endpoint Status register (EPSTATUS.STALLRQ0)
- Transmit Stall 1 bit in the Endpoint Status register (EPSTATUS.STALLRQ1)

39.6.2.3. Multi-Packet Transfers

Multi-packet transfer enables a data payload exceeding the endpoint maximum transfer size to be transferred as multiple packets without software intervention. This reduces the number of interrupts and software intervention required to manage higher level USB transfers. Multi-packet transfer is identical to the IN and OUT transactions described below unless otherwise noted in this section.

The application software provides the size and address of the RAM buffer to be proceeded by the USB module for a specific endpoint, and the USB module will split the buffer in the required USB data transfers without any software intervention.

Figure 39-3. Multi-Packet Feature - Reduction of CPU Overhead



39.6.2.4. USB Reset

The USB bus reset is initiated by a connected host and managed by hardware.

During USB reset the following registers are cleared:

- Device Endpoint Configuration (EPCFG) register - except for Endpoint 0
- Device Frame Number (FNUM) register
- Device Address (DADD) register
- Device Endpoint Interrupt Enable Clear/Set (EPINTENCLR/SET) register
- Device Endpoint Interrupt Flag (EPINTFLAG) register
- Transmit Stall 0 bit in the Endpoint Status register (EPSTATUS.STALLRQ0)
- Transmit Stall 1 bit in the Endpoint Status register (EPSTATUS.STALLRQ1)
- Endpoint Interrupt Summary (EPINTSMRY) register
- Upstream resume bit in the Control B register (CTRLB.UPRSM)

At the end of the reset process, the End of Reset bit is set in the Interrupt Flag register (INTFLAG.EORST).

39.6.2.5. Start-of-Frame

When a Start-of-Frame (SOF) token is detected, the frame number from the token is stored in the Frame Number field in the Device Frame Number register (FNUM.FNUM), and the Start-of-Frame interrupt bit in the Device Interrupt Flag register (INTFLAG.SOF) is set. If there is a CRC or bit-stuff error, the Frame Number Error status flag (FNUM.FNCERR) in the FNUM register is set.

39.6.2.6. Management of SETUP Transactions

When a SETUP token is detected and the device address of the token packet does not match DADD.DADD, the packet is discarded and the USB module returns to idle and waits for the next token packet.

When the address matches, the USB module checks if the endpoint is enabled in EPCFG. If the addressed endpoint is disabled, the packet is discarded and the USB module returns to idle and waits for the next token packet.

When the endpoint is enabled, the USB module then checks on the EPCFG of the addressed endpoint. If the EPCFG.EPTYPE0 is not set to control, the USB module returns to idle and waits for the next token packet.

When the EPCFG.EPTYPE0 matches, the USB module then fetches the Data Buffer Address (ADDR) from the addressed endpoint's descriptor and waits for a DATA0 packet. If a PID error or any other PID than DATA0 is detected, the USB module returns to idle and waits for the next token packet.

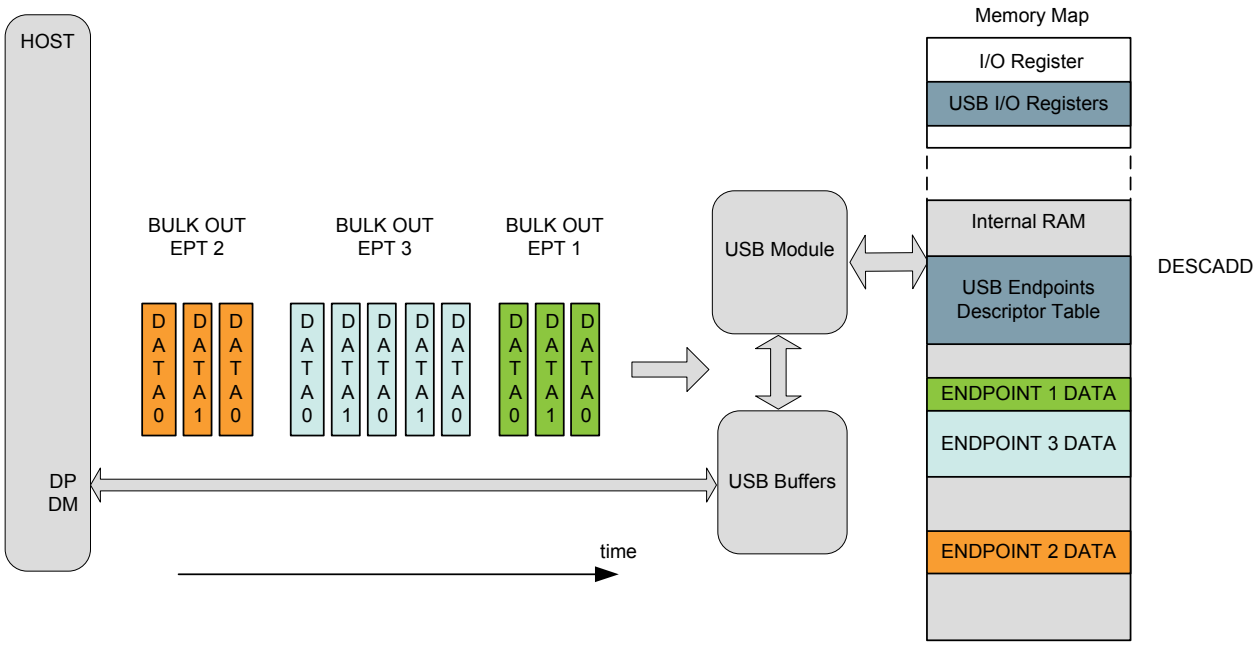
When the data PID matches and if the Received Setup Complete interrupt bit in the Device Endpoint Interrupt Flag register (EPINTFLAG.RXSTP) is equal to zero, ignoring the Bank 0 Ready bit in the Device Endpoint Status register (EPSTATUS.BK0RDY), the incoming data is written to the data buffer pointed to by the Data Buffer Address (ADDR). If the number of received data bytes exceeds the endpoint's maximum data payload size as specified by the PCKSIZE.SIZE, the remainders of the received data bytes are discarded. The packet will still be checked for bit-stuff and CRC errors. Software must never report a endpoint size to the host that is greater than the value configured in PCKSIZE.SIZE. If a bit-stuff or CRC error is detected in the packet, the USB module returns to idle and waits for the next token packet.

If data is successfully received, an ACK handshake is returned to the host, and the number of received data bytes, excluding the CRC, is written to the Byte Count (PCKSIZE.BYTE_COUNT). If the number of received data bytes is the maximum data payload specified by PCKSIZE.SIZE, no CRC data is written to the data buffer. If the number of received data bytes is the maximum data payload specified by PCKSIZE.SIZE minus one, only the first CRC data is written to the data buffer. If the number of received data is equal or less than the data payload specified by PCKSIZE.SIZE minus two, both CRC data bytes are written to the data buffer.

Finally the EPSTATUS is updated. Data Toggle OUT bit (EPSTATUS.DTGLOUT), the Data Toggle IN bit (EPSTATUS.DTGLIN), the current bank bit (EPSTATUS.CURRBK) and the Bank Ready 0 bit (EPSTATUS.BK0RDY) are set. Bank Ready 1 bit (EPSTATUS.BK1RDY) and the Stall Bank 0/1 bit (EPSTATUS.STALLQR0/1) are cleared on receiving the SETUP request. The RXSTP bit is set and triggers an interrupt if the Received Setup Interrupt Enable bit is set in Endpoint Interrupt Enable Set/Clear register (EPINTENSET/CLR.RXSTP).

39.6.2.7. Management of OUT Transactions

Figure 39-4. OUT Transfer: Data Packet Host to USB Device



When an OUT token is detected, and the device address of the token packet does not match DADD.DADD, the packet is discarded and the USB module returns to idle and waits for the next token packet.

If the address matches, the USB module checks if the endpoint number received is enabled in the EPCFG of the addressed endpoint. If the addressed endpoint is disabled, the packet is discarded and the USB module returns to idle and waits for the next token packet.

When the endpoint is enabled, the USB module then checks the Endpoint Configuration register (EPCFG) of the addressed output endpoint. If the type of the endpoint (EPCFG.EPTYPE0) is not set to OUT, the USB module returns to idle and waits for the next token packet.

The USB module then fetches the Data Buffer Address (ADDR) from the addressed endpoint's descriptor, and waits for a DATA0 or DATA1 packet. If a PID error or any other PID than DATA0 or DATA1 is detected, the USB module returns to idle and waits for the next token packet.

If EPSTATUS.STALLRQ0 in EPSTATUS is set, the incoming data is discarded. If the endpoint is not isochronous, a STALL handshake is returned to the host and the Transmit Stall Bank 0 interrupt bit in EPINTFLAG (EPINTFLAG.STALL0) is set.

For isochronous endpoints, data from both a DATA0 and DATA1 packet will be accepted. For other endpoint types the PID is checked against EPSTATUS.DTGLOUT. If a PID mismatch occurs, the incoming data is discarded, and an ACK handshake is returned to the host.

If EPSTATUS.BK0RDY is set, the incoming data is discarded, the bit Transmit Fail 0 interrupt bit in EPINTFLAG (EPINTFLAG.TRFAIL0) and the status bit STATUS_BK.ERRORFLOW are set. If the endpoint is not isochronous, a NAK handshake is returned to the host.

The incoming data is written to the data buffer pointed to by the Data Buffer Address (ADDR). If the number of received data bytes exceeds the maximum data payload specified as PCKSIZE.SIZE, the remainders of the received data bytes are discarded. The packet will still be checked for bit-stuff and CRC errors. If a bit-stuff or CRC error is detected in the packet, the USB module returns to idle and waits for the next token packet.

If the endpoint is isochronous and a bit-stuff or CRC error in the incoming data, the number of received data bytes, excluding CRC, is written to PCKSIZE.BYTE_COUNT. Finally the EPINTFLAG.TRFAIL0 and CRC Error bit in the Device Bank Status register (STATUS_BK.CRCERR) is set for the addressed endpoint.

If data was successfully received, an ACK handshake is returned to the host if the endpoint is not isochronous, and the number of received data bytes, excluding CRC, is written to PCKSIZE.BYTE_COUNT. If the number of received data bytes is the maximum data payload specified by PCKSIZE.SIZE no CRC data bytes are written to the data buffer. If the number of received data bytes is the maximum data payload specified by PCKSIZE.SIZE minus one, only the first CRC data byte is written to the data buffer If the number of received data is equal or less than the data payload specified by PCKSIZE.SIZE minus two, both CRC data bytes are written to the data buffer.

Finally in EPSTATUS for the addressed output endpoint, EPSTATUS.BK0RDY is set and EPSTATUS.DTGLOUT is toggled if the endpoint is not isochronous. The flag Transmit Complete 0 interrupt bit in EPINTFLAG (EPINTFLAG.TRCPT0) is set for the addressed endpoint.

39.6.2.8. Multi-Packet Transfers for OUT Endpoint

The number of data bytes received is stored in endpoint PCKSIZE.BYTE_COUNT as for normal operation. Since PCKSIZE.BYTE_COUNT is updated after each transaction, it must be set to zero when setting up a new transfer. The total number of bytes to be received must be written to PCKSIZE.MULTI_PACKET_SIZE. This value must be a multiple of PCKSIZE.SIZE, otherwise excess data may be written to SRAM locations used by other parts of the application.

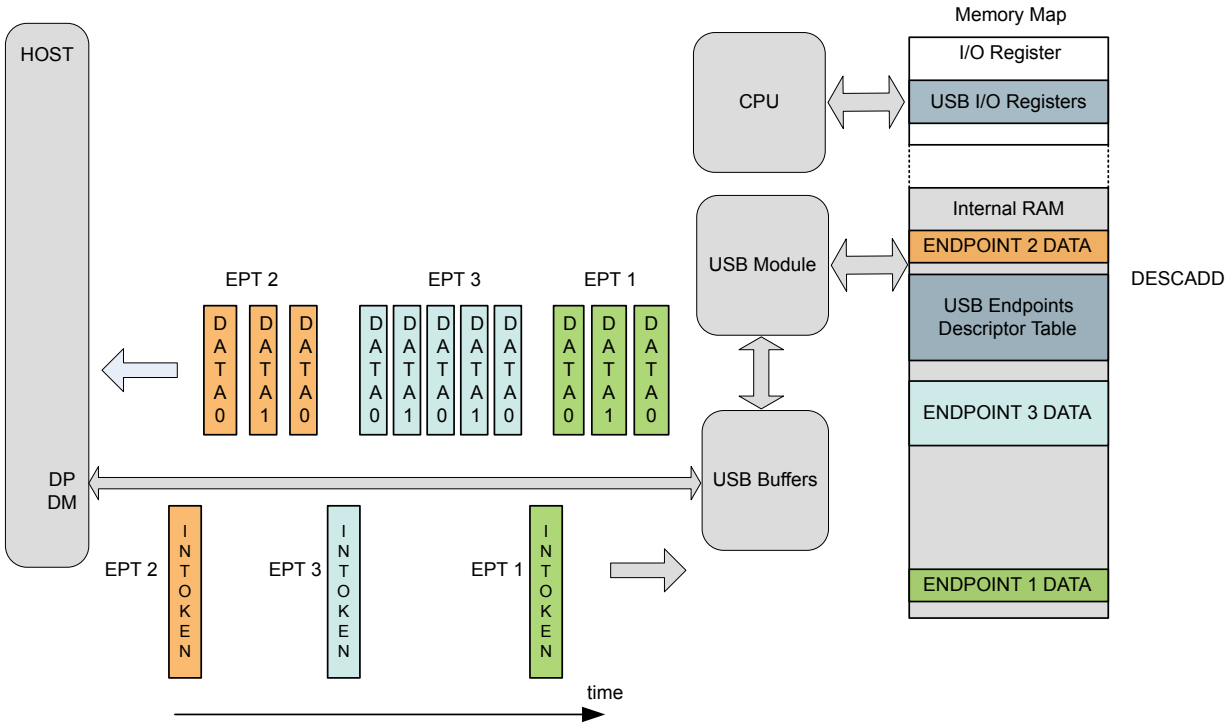
EPSTATUS.DTGLOUT management for non-isochronous packets and EPINTFLAG.BK1RDY/BK0RDY management are as for normal operation.

If a maximum payload size packet is received, PCKSIZE.BYTE_COUNT will be incremented by PCKSIZE.SIZE after the transaction has completed, and EPSTATUS.DTGLOUT will be toggled if the endpoint is not isochronous. If the updated PCKSIZE.BYTE_COUNT is equal to

PCKSIZE.MULTI_PACKET_SIZE (i.e. the last transaction), EPSTATUS.BK1RDY/BK0RDY, and EPINTFLAG.TRCPT0/TRCPT1 will be set.

39.6.2.9. Management of IN Transactions

Figure 39-5. IN Transfer: Data Packet USB Device to Host After Request from Host



When an IN token is detected, and if the device address of the token packet does not match DADD.DADD, the packet is discarded and the USB module returns to idle and waits for the next token packet.

When the address matches, the USB module checks if the endpoint received is enabled in the EPCFG of the addressed endpoint and if not, the packet is discarded and the USB module returns to idle and waits for the next token packet.

When the endpoint is enabled, the USB module then checks on the EPCFG of the addressed input endpoint. If the EPCFG.EPTYPE1 is not set to IN, the USB module returns to idle and waits for the next token packet.

If EPSTATUS.STALLRQ1 in EPSTATUS is set, and the endpoint is not isochronous, a STALL handshake is returned to the host and EPINTFLAG.STALL1 is set.

If EPSTATUS.BK1RDY is cleared, the flag EPINTFLAG.TRFAIL1 is set. If the endpoint is not isochronous, a NAK handshake is returned to the host.

The USB module then fetches the Data Buffer Address (ADDR) from the addressed endpoint's descriptor. The data pointed to by the Data Buffer Address (ADDR) is sent to the host in a DATA0 packet if the endpoint is isochronous. For non-isochronous endpoints a DATA0 or DATA1 packet is sent depending on the state of EPSTATUS.DTGLIN. When the number of data bytes specified in endpoint PCKSIZE.BYTE_COUNT is sent, the CRC is appended and sent to the host.

For isochronous endpoints, EPSTATUS.BK1RDY is cleared and EPINTFLAG.TRCPT1 is set.

For all non-isochronous endpoints the USB module waits for an ACK handshake from the host. If an ACK handshake is not received within 16 bit times, the USB module returns to idle and waits for the next token packet. If an ACK handshake is successfully received EPSTATUS.BK1RDY is cleared, EPINTFLAG.TRCPT1 is set and EPSTATUS.DTGLIN is toggled.

39.6.2.10. Multi-Packet Transfers for IN Endpoint

The total number of data bytes to be sent is written to PCKSIZE.BYTE_COUNT as for normal operation. The Multi-packet size register (PCKSIZE.MULTI_PACKET_SIZE) is used to store the number of bytes that are sent, and must be written to zero when setting up a new transfer.

When an IN token is received, PCKSIZE.BYTE_COUNT and PCKSIZE.MULTI_PACKET_SIZE are fetched. If PCKSIZE.BYTE_COUNT minus PCKSIZE.MULTI_PACKET_SIZE is less than the endpoint PCKSIZE.SIZE, endpoint BYTE_COUNT minus endpoint PCKSIZE.MULTI_PACKET_SIZE bytes are transmitted, otherwise PCKSIZE.SIZE number of bytes are transmitted. If endpoint PCKSIZE.BYTE_COUNT is a multiple of PCKSIZE.SIZE, the last packet sent will be zero-length if the AUTOZLP bit is set.

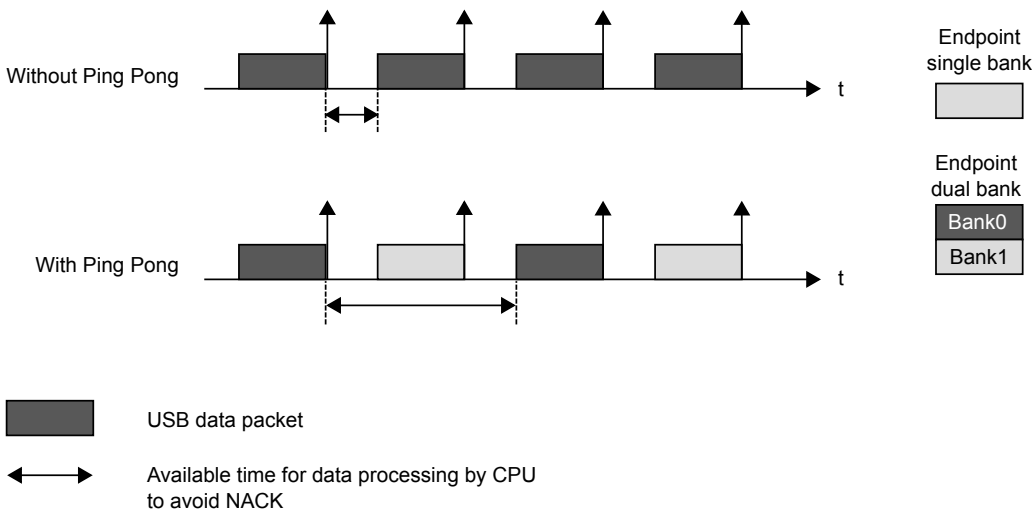
If a maximum payload size packet was sent (i.e. not the last transaction), MULTI_PACKET_SIZE will be incremented by the PCKSIZE.SIZE. If the endpoint is not isochronous the EPSTATUS.DTLGIN bit will be toggled when the transaction has completed. If a short packet was sent (i.e. the last transaction), MULTI_PACKET_SIZE is incremented by the data payload. EPSTATUS.BK0/1RDY will be cleared and EPINTFLAG.TRCPT0/1 will be set.

39.6.2.11. Ping-Pong Operation

When an endpoint is configured for ping-pong operation, it uses both the input and output data buffers (banks) for a given endpoint in a single direction. The direction is selected by enabling one of the IN or OUT direction in EPCFG.EPTYPE0/1 and configuring the opposite direction in EPCFG.EPTYPE1/0 as Dual Bank.

When ping-pong operation is enabled for an endpoint, the endpoint in the opposite direction must be configured as dual bank. The data buffer, data address pointer and byte counter from the enabled endpoint are used as Bank 0, while the matching registers from the disabled endpoint are used as Bank 1.

Figure 39-6. Ping-Pong Overview



The Bank Select flag in EPSTATUS.CURBK indicates which bank data will be used in the next transaction, and is updated after each transaction. According to EPSTATUS.CURBK, EPINTFLAG.TRCPT0 or EPINTFLAG.TRFAIL0 or EPINTFLAG.TRCPT1 or EPINTFLAG.TRFAIL1 in EPINTFLAG and Data Buffer 0/1 ready (EPSTATUS.BK0RDY and EPSTATUS.BK1RDY) are set. The EPSTATUS.DTGLOUT and EPSTATUS.DTGLIN are updated for the enabled endpoint direction only.

39.6.2.12. Feedback Operation

Feedback endpoints are endpoints with same the address but in different directions. This is usually used in explicit feedback mechanism in USB Audio, where a feedback endpoint is associated to one or more isochronous data endpoints to which it provides feedback service. The feedback endpoint always has the opposite direction from the data endpoint.

The feedback endpoint always has the opposite direction from the data endpoint(s). The feedback endpoint has the same endpoint number as the first (lower) data endpoint. A feedback endpoint can be created by configuring an endpoint with different endpoint size (PCKSIZE.SIZE) and different endpoint type (EPCFG.EPTYPE0/1) for the IN and OUT direction.

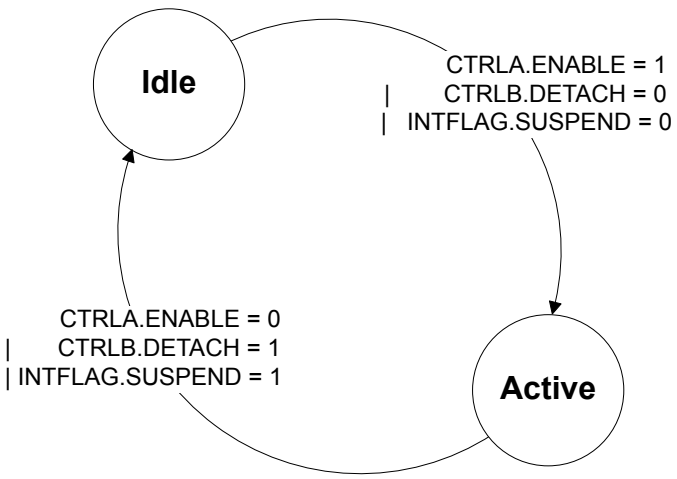
Example Configuration for Feedback Operation:

- Endpoint n / IN: EPCFG.EPTYPE1 = Interrupt IN, PCKSIZE.SIZE = 64.
- Endpoint n / OUT: EPCFG.EPTYPE0= Isochronous OUT, PCKSIZE.SIZE = 512.

39.6.2.13. Suspend State and Pad Behavior

The following figure, Pad Behavior, illustrates the behavior of the USB pad in device mode.

Figure 39-7. Pad Behavior

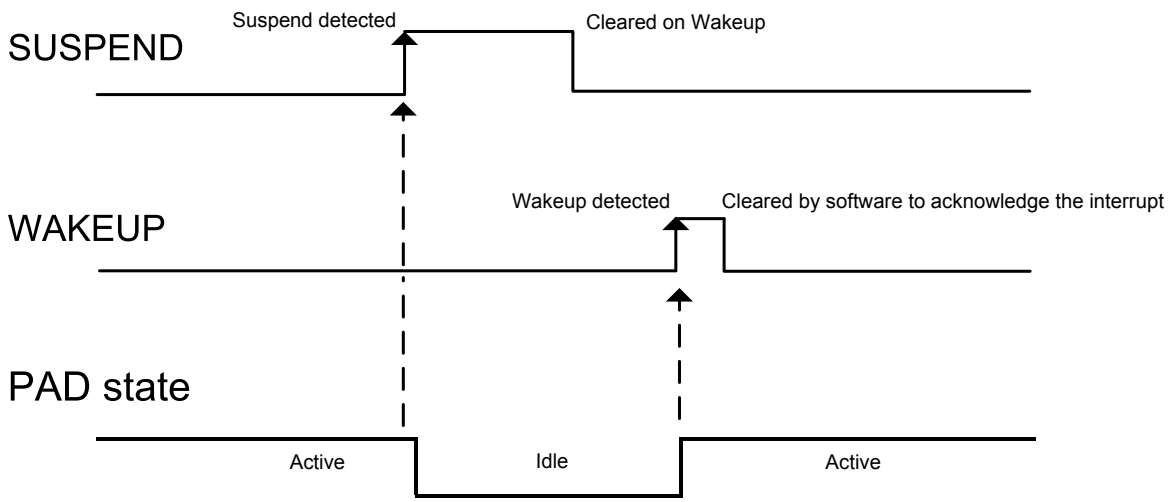


In Idle state, the pad is in low power consumption mode.

In Active state, the pad is active.

The following figure, Pad Events, illustrates the pad events leading to a PAD state change.

Figure 39-8. Pad Events



The Suspend Interrupt bit in the Device Interrupt Flag register (INTFLAG.SUSPEND) is set when a USB Suspend state has been detected on the USB bus. The USB pad is then automatically put in the Idle state. The detection of a non-idle state sets the Wake Up Interrupt bit in INTFLAG(INTFLAG.WAKEUP) and wakes the USB pad.

The pad goes to the Idle state if the USB module is disabled or if CTRLB.DETACH is written to one. It returns to the Active state when CTRLA.ENABLE is written to one and CTRLB.DETACH is written to zero.

39.6.2.14. Remote Wakeup

The remote wakeup request (also known as upstream resume) is the only request the device may send on its own initiative. This should be preceded by a DEVICE_REMOTE_WAKEUP request from the host.

First, the USB must have detected a “Suspend” state on the bus, i.e. the remote wakeup request can only be sent after INTFLAG.SUSPEND has been set.

The user may then write a one to the Remote Wakeup bit in CTRLB(CTRLB.UPRSM) to send an Upstream Resume to the host initiating the wakeup. This will automatically be done by the controller after 5 ms of inactivity on the USB bus.

When the controller sends the Upstream Resume INTFLAG.WAKEUP is set and INTFLAG.SUSPEND is cleared.

The CTRLB.UPRSM is cleared at the end of the transmitting Upstream Resume.

In case of a rebroadcast resume initiated by the host, the End of Resume bit in INTFLAG(INTFLAG.EORSM) flag is set when the rebroadcast resume is completed.

In the case where the CTRLB.UPRSM bit is set while a host initiated downstream resume is already started, the CTRLB.UPRSM is cleared and the upstream resume request is ignored.

39.6.2.15. Link Power Management L1 (LPM-L1) Suspend State Entry and Exit as Device

The LPM Handshake bit in CTRLB.LPMHDSK should be configured to accept the LPM transaction.

When a LPM transaction is received on any enabled endpoint n and a handshake has been sent in response by the controller according to CTRLB.LPMHDSK, the Device Link Power Manager (EXTREG) register is updated in the bank 0 of the addressed endpoint's descriptor. It contains information such as the Best Effort Service Latency (BESL), the Remote Wake bit (bRemoteWake), and the Link State parameter (bLinkState). Usually, the LPM transaction uses only the endpoint number 0.

If the LPM transaction was positively acknowledged (ACK handshake), USB sets the Link Power Management Interrupt bit in INTFLAG(INTFLAG.LPMSUSP) bit which indicates that the USB transceiver is suspended, reducing power consumption. This suspend occurs 9 microseconds after the LPM transaction according to the specification.

To further reduce consumption, it is recommended to stop the USB clock while the device is suspended.

The MCU can also enter in one of the available sleep modes if the wakeup time latency of the selected sleep mode complies with the host latency constraint (see the BESL parameter in EXTREG on page 951 register).

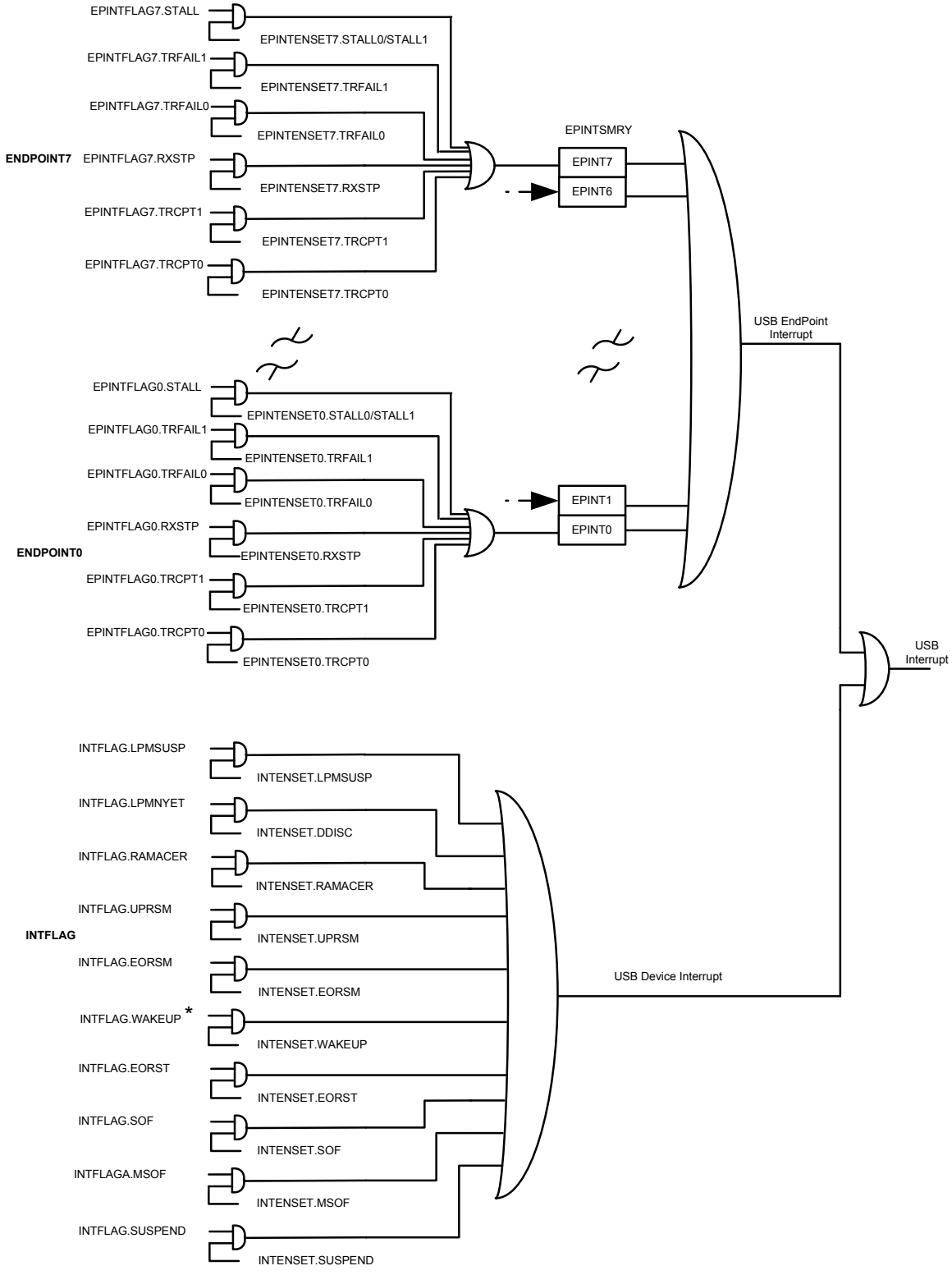
Recovering from this LPM-L1 suspend state is exactly the same as the Suspend state (see Section Suspend State and Pad Behavior on page 900) except that the remote wakeup duration initiated by USB is shorter to comply with the Link Power Management specification.

If the LPM transaction is responded with a NYET, the Link Power Management Not Yet Interrupt Flag INTFLAG(INTFLAG.LPMNYET) is set. This generates an interrupt if the Link Power Management Not Yet Interrupt Enable bit in INTENCLR/SET (INTENCLR/SET.LPMNYET) is set.

If the LPM transaction is responded with a STALL or no handshake, no flag is set, and the transaction is ignored.

39.6.2.16. USB Device Interrupt

Figure 39-9. Device Interrupt



* Asynchronous interrupt

The WAKEUP is an asynchronous interrupt and can be used to wake-up the device from any sleep mode.

39.6.3. Host Operations

This section gives an overview of the USB module Host operation during normal transactions. For more details on general USB and USB protocol, refer to Universal Serial Bus Specification revision 2.1.

39.6.3.1. Device Detection and Disconnection

Prior to device detection the software must set the VBUS is OK bit in CTRLB (CTRLB.VBUSOK) register when the VBUS is available. This notifies the USB host that USB operations can be started. When the bit CTRLB.VBUSOK is zero and even if the USB HOST is configured and enabled, host operation is halted. Setting the bit CTRLB.VBUSOK will allow host operation when the USB is configured.

The Device detection is managed by the software using the Line State field in the Host Status (STATUS.LINESTATE) register. The device connection is detected by the host controller when DP or DM is pulled high, depending of the speed of the device.

The device disconnection is detected by the host controller when both DP and DM are pulled down using the STATUS.LINESTATE registers.

The Device Connection Interrupt bit in INTFLAG (INTFLAG.DCONN) is set if a device connection is detected.

The Device Disconnection Interrupt bit in INTFLAG (INTFLAG.DDISC) is set if a device disconnection is detected.

39.6.3.2. Host Terminology

In host mode, the term pipe is used instead of endpoint. A host pipe corresponds to a device endpoint, refer to "Universal Serial Bus Specification revision 2.1." for more information.

39.6.3.3. USB Reset

The USB sends a USB reset signal when the user writes a one to the USB Reset bit in CTRLB (CTRLB.BUSRESET). When the USB reset has been sent, the USB Reset Sent Interrupt bit in the INTFLAG (INTFLAG.RST) is set and all pipes will be disabled.

If the bus was previously in a suspended state (Start of Frame Generation Enable bit in CTRLB (CTRLB.SOFE) is zero) the USB will switch it to the Resume state, causing the bus to asynchronously set the Host Wakeup Interrupt flag (INTFLAG.WAKEUP). The CTRLB.SOFE bit will be set in order to generate SOFs immediately after the USB reset.

During USB reset the following registers are cleared:

- All Host Pipe Configuration register (PCFG)
- Host Frame Number register (FNUM)
- Interval for the Bulk-Out/Ping transaction register (BINTERVAL)
- Host Start-of-Frame Control register (HSOFC)
- Pipe Interrupt Enable Clear/Set register (PINTENCLR/SET)
- Pipe Interrupt Flag register (PINTFLAG)
- Pipe Freeze bit in Pipe Status register (PSTATUS.FREEZE)

After the reset the user should check the Speed Status field in the Status register (STATUS.SPEED) to find out the current speed according to the capability of the peripheral.

39.6.3.4. Pipe Configuration

Pipe data can be placed anywhere in the RAM. The USB controller accesses these pipes directly through the AHB master (built-in DMA) with the help of the pipe descriptors. The base address of the pipe descriptors needs to be written in the Descriptor Address register (DESCADD) by the user. Refer also to Pipe Descriptor Structure on page 980.

Before using a pipe, the user should configure the direction and type of the pipe in Type of Pipe field in the Host Pipe Configuration register (PCFG.PTYPE). The pipe descriptor registers should be initialized to known values before using the pipe, so that the USB controller does not read the random values from the RAM.

The Pipe Size field in the Packet Size register (PCKSIZE.SIZE) should be configured as per the size reported by the device for the endpoint associated with this pipe. The Address of Data Buffer register (ADDR) should be set to the data buffer used for pipe transfers.

The Pipe Bank bit in PCFG (PCFG.BK) should be set to one if dual banking is desired. Dual bank is not supported for Control pipes.

The Ram Access Interrupt bit in Host Interrupt Flag register (INTFLAG.RAMACER) is set when a RAM access underflow error occurs during an OUT stage.

When a pipe is disabled, the following registers are cleared for that pipe:

- Interval for the Bulk-Out/Ping transaction register (BINTERVAL)
- Pipe Interrupt Enable Clear/Set register (PINTENCLR/SET)
- Pipe Interrupt Flag register (PINTFLAG)
- Pipe Freeze bit in Pipe Status register (PSTATUS.FREEZE)

39.6.3.5. Pipe Activation

A disabled pipe is inactive, and will be reset along with its context registers (pipe registers for the pipe n). Pipes are enabled by writing Type of the Pipe in PCFG (PCFG.PTYPE) to a value different than 0x0 (disabled).

When a pipe is enabled, the Pipe Freeze bit in Pipe Status register (PSTATUS.FREEZE) is set. This allow the user to complete the configuration of the pipe, without starting a USB transfer.

When starting an enumeration, the user retrieves the device descriptor by sending an GET_DESCRIPTOR USB request. This descriptor contains the maximal packet size of the device default control endpoint (bMaxPacketSize0) which the user should use to reconfigure the size of the default control pipe.

39.6.3.6. Pipe Address Setup

Once the device has answered the first host requests with the default device address 0, the host assigns a new address to the device. The host controller has to send a USB reset to the device and a SET_ADDRESS(addr) SETUP request with the new address to be used by the device. Once this SETUP transaction is complete, the user writes the new address to the Pipe Device Address field in the Host Control Pipe register (CTRL_PIPE.PDADDR) in Pipe descriptor. All following requests by this pipe will be performed using this new address.

39.6.3.7. Suspend and Wakeup

Setting CTRLB.SOFE to zero when in host mode will cause the USB to cease sending Start-of-Frames on the USB bus and enter the Suspend state. The USB device will enter the Suspend state 3ms later.

Before entering suspend by writing CTRLB.SOFE to zero, the user must freeze the active pipes by setting their PSTATUS.FREEZE bit. Any current on-going pipe will complete its transaction, and then all pipes will be inactive. The user should wait at least 1 complete frame before entering the suspend mode to avoid any data loss.

The device can awaken the host by sending an Upstream Resume (Remote Wakeup feature). When the host detects a non-idle state on the USB bus, it sets the INTFLAG.WAKEUP. If the non-idle bus state corresponds to an Upstream Resume (K state), the Upstream Resume Received Interrupt bit in INTFLAG (INTFLAG.UPRSM) is set and the user must generate a Downstream Resume within 1 ms and for at least 20 ms. It is required to first write a one to the Send USB Resume bit in CTRLB (CTRLB.RESUME)

to respond to the upstream resume with a downstream resume. Alternatively, the host can resume from a suspend state by sending a Downstream Resume on the USB bus (CTRLB.RESUME set to 1). In both cases, when the downstream resume is completed, the CTRLB.SOFE bit is automatically set and the host enters again the active state.

39.6.3.8. Phase-locked SOFs

To support the Synchronous Endpoints capability, the period of the emitted Start-of-Frame is maintained while the USB connection is not in the active state. This does not apply for the disconnected/connected/reset states. It applies for active/idle/suspend/resume states. The period of Start-of-Frame will be 1ms when the USB connection is in active state and an integer number of milli-seconds across idle/suspend/resume states.

To ensure the Synchronous Endpoints capability, the GCLK_USB clock must be kept running. If the GCLK_USB is interrupted, the period of the emitted Start-of-Frame will be erratic.

39.6.3.9. Management of Control Pipes

A control transaction is composed of three stages:

- SETUP
- Data (IN or OUT)
- Status (IN or OUT)

The user has to change the pipe token according to each stage using the Pipe Token field in PCFG (PCFG.PTOKEN).

For control pipes only, the token is assigned a specific initial data toggle sequence:

- SETUP: Data0
- IN: Data1
- OUT: Data1

39.6.3.10. Management of IN Pipes

IN packets are sent by the USB device controller upon IN request reception from the host. All the received data from the device to the host will be stored in the bank provided the bank is empty. The pipe and its descriptor in RAM must be configured.

The host indicates it is able to receive data from the device by clearing the Bank 0/1 Ready bit in PSTATUS (PSTATUS.BK0/1RDY), which means that the memory for the bank is available for new USB transfer.

The USB will perform IN requests as long as the pipe is not frozen by the user.

The generation of IN requests starts when the pipe is unfrozen (PSTATUS.PFREEZE is set to zero).

When the current bank is full, the Transmit Complete 0/1 bit in PINTFLAG (PINTFLAG.TRCPT0/1) will be set and trigger an interrupt if enabled and the PSTATUS.BK0/1RDY bit will be set.

PINTFLAG.TRCPT0/1 must be cleared by software to acknowledge the interrupt. This is done by writing a one to the PINTFLAG.TRCPT0/1 of the addressed pipe.

The user reads the PCKSIZE.BYTE_COUNT to know how many bytes should be read.

To free the bank the user must read the IN data from the address ADDR in the pipe descriptor and clear the PKSTATUS.BK0/1RDY bit. When the IN pipe is composed of multiple banks, a successful IN transaction will switch to the next bank. Another IN request will be performed by the host as long as the PSTATUS.BK0/1RDY bit for that bank is set. The PINTFLAG.TRCPT0/1 and PSTATUS.BK0/1RDY will be updated accordingly.

The user can follow the current bank looking at Current Bank bit in PSTATUS (PSTATUS.CURBK) and by looking at Data Toggle for IN pipe bit in PSTATUS (PSTATUS.DTGLIN).

When the pipe is configured as single bank (Pipe Bank bit in PCFG (PCFG.BK) is 0), only PINTFLAG.TRCPT0 and PSTATUS.BK0 are used. When the pipe is configured as dual bank (PCFG.BK is 1), both PINTFLAG.TRCPT0/1 and PSTATUS.BK0/1 are used.

39.6.3.11. Management of OUT Pipes

OUT packets are sent by the host. All the data stored in the bank will be sent to the device provided the bank is filled. The pipe and its descriptor in RAM must be configured.

The host can send data to the device by writing to the data bank 0 in single bank or the data bank 0/1 in dual bank.

The generation of OUT packet starts when the pipe is unfrozen (PSTATUS.PFREEZE is zero).

The user writes the OUT data to the data buffer pointer by ADDR in the pipe descriptor and allows the USB to send the data by writing a one to the PSTATUS.BK0/1RDY. This will also cause a switch to the next bank if the OUT pipe is part of a dual bank configuration.

PINTFLAGn.TRCPT0/1 must be cleared before setting PSTATUS.BK0/1RDY to avoid missing an PINTFLAGn.TRCPT0/1 event.

39.6.3.12. Alternate Pipe

The user has the possibility to run sequentially several logical pipes on the same physical pipe. It allows addressing of any device endpoint of any attached device on the bus.

Before switching pipe, the user should save the pipe context (Pipe registers and descriptor for pipe n).

After switching pipe, the user should restore the pipe context (Pipe registers and descriptor for pipe n) and in particular PCFG, and PSTATUS.

39.6.3.13. Data Flow Error

This error exists only for isochronous and interrupt pipes for both IN and OUT directions. It sets the Transmit Fail bit in PINTFLAG (PINTFLAG.TRFAIL), which triggers an interrupt if the Transmit Fail bit in PINTENCLR/SET(PINTENCLR/SET.TRFAIL) is set. The user must check the Pipe Interrupt Summary register (PINTSMRY) to find out the pipe which triggered the interrupt. Then the user must check the origin of the interrupt's bank by looking at the Pipe Bank Status register (STATUS_BK) for each bank. If the Error Flow bit in the STATUS_BK (STATUS_BK.ERRORFLOW) is set then the user is able to determine the origin of the data flow error. As the user knows that the endpoint is an IN or OUT the error flow can be deduced as OUT underflow or as an IN overflow.

An underflow can occur during an OUT stage if the host attempts to send data from an empty bank. If a new transaction is successful, the relevant bank descriptor STATUS_BK.ERRORFLOW will be cleared.

An overflow can occur during an IN stage if the device tries to send a packet while the bank is full. Typically this occurs when a CPU is not fast enough. The packet data is not written to the bank and is lost. If a new transaction is successful, the relevant bank descriptor STATUS_BK.ERRORFLOW will be cleared.

39.6.3.14. CRC Error

This error exists only for isochronous IN pipes. It sets the PINTFLAG.TRFAIL, which triggers an interrupt if PINTENCLR/SET.TRFAIL is set. The user must check the PINTSMRY to find out the pipe which triggered the interrupt. Then the user must check the origin of the interrupt's bank by looking at the bank descriptor STATUS_BK for each bank and if the CRC Error bit in STATUS_BK (STATUS_BK.CRCERR) is set then the user is able to determine the origin of the CRC error. A CRC error can occur during the IN stage if the USB detects a corrupted packet. The IN packet will remain stored in the bank and PINTFLAG.TRCPT0/1 will be set.

39.6.3.15. PERR Error

This error exists for all pipes. It sets the PINTFLAG.PERR Interrupt, which triggers an interrupt if PINTFLAG.PERR is set. The user must check the PINTSMRY register to find out the pipe which can cause an interrupt.

A PERR error occurs if one of the error field in the STATUS_PIPE register in the Host pipe descriptor is set and the Error Count field in STATUS_PIPE (STATUS_PIPE.ERCNT) exceeds the maximum allowed number of Pipe error(s) as defined in Pipe Error Max Number field in CTRL_PIPE (CTRL_PIPE.PERMAX). Refer to section STATUS_PIPE on page 987 register.

If one of the error field in the STATUS_PIPE register from the Host Pipe Descriptor is set and the STATUS_PIPE.ERCNT is less than the CTRL_PIPE.PERMAX, the STATUS_PIPE.ERCNT is incremented.

39.6.3.16. Link Power Management L1 (LPM-L1) Suspend State Entry and Exit as Host.

An EXTENDED LPM transaction can be transmitted by any enabled pipe. The PCFGn.PTYPE should be set to EXTENDED. Other fields as PCFG.PTOKEN, PCFG.BK and PCKSIZE.SIZE are irrelevant in this configuration. The user should also set the EXTREG.VARIABLE in the descriptor as described in EXTREG on page 984 register.

When the pipe is configured and enabled, an EXTENDED TOKEN followed by a LPM TOKEN are transmitted. The device responds with a valid HANDSHAKE, corrupted HANDSHAKE or no HANDSHAKE (TIME-OUT).

If the valid HANDSHAKE is an ACK, the host will immediately proceed to L1 SLEEP and the PINTFLAG.TRCT0 is set. The minimum duration of the L1 SLEEP state will be the TL1RetryAndResidency as defined in the reference document "ENGINEERING CHANGE NOTICE, USB 2.0 Link Power Management Addendum". When entering the L1 SLEEP state, the CTRLB.SOFE is cleared, avoiding Start-of-Frame generation.

If the valid HANDSHAKE is a NYET PINTFLAG.TRFAIL is set.

If the valid HANDSHAKE is a STALL the PINTFLAG.STALL is set.

If there is no HANDSHAKE or corrupted HANDSHAKE, the EXTENDED/LPM pair of TOKENS will be transmitted again until reaching the maximum number of retries as defined by the CTRL_PIPE.PERMAX in the pipe descriptor.

If the last retry returns no valid HANDSHAKE, the PINTFLAGn.PERR is set, and the STATUS_BK is updated in the pipe descriptor.

All LPM transactions, should they end up with a ACK, a NYET, a STALL or a PERR, will set the PSTATUS.PFREEZE bit, freezing the pipe before a succeeding operation. The user should unfreeze the pipe to start a new LPM transaction.

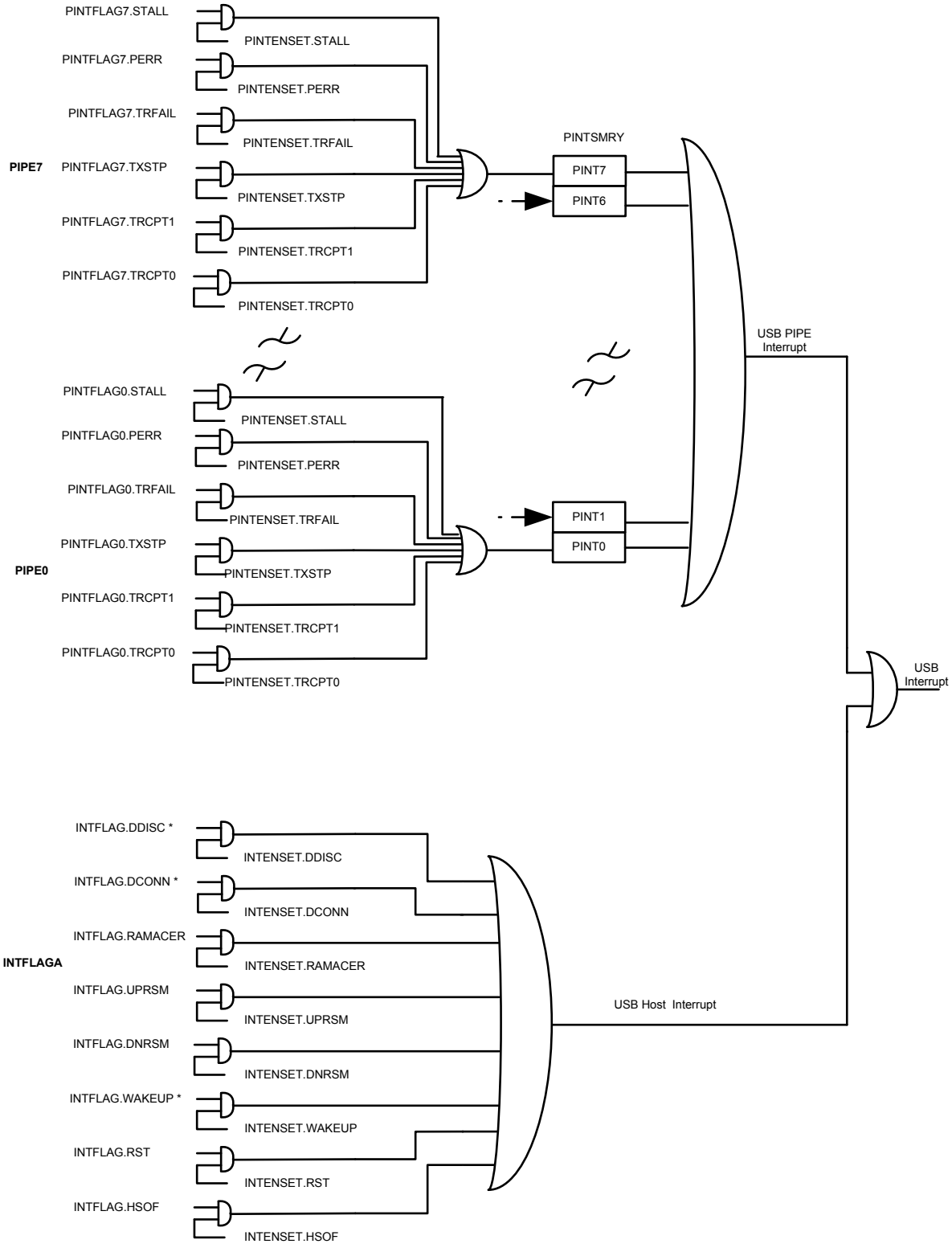
To exit the L1 STATE, the user initiate a DOWNSTREAM RESUME by setting the bit CTRLB.RESUME or a L1 RESUME by setting the Send L1 Resume bit in CTRLB (CTRLB.L1RESUME). In the case of a L1 RESUME, the K STATE duration is given by the BESL bit field in the EXTREG.VARIABLE field. See EXTREG on page 984.

When the host is in the L1 SLEEP state after a successful LPM transmitted, the device can initiate an UPSTREAM RESUME. This will set the Upstream Resume Interrupt bit in INTFLAG (INTFLAG.UPRSM). The host should proceed then to a L1 RESUME as described above.

After resuming from the L1 SLEEP state, the bit CTRLB.SOFE is set, allowing Start-of-Frame generation.

39.6.3.17. Host Interrupt

Figure 39-10. Host Interrupt



* Asynchronous interrupt

The WAKEUP is an asynchronous interrupt and can be used to wake-up the device from any sleep mode.

39.7. Register Summary

The register mapping depends on the Operating Mode field in the Control A register (CTRLA.MODE). The register summary is detailed below.

39.7.1. Common Device Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0	MODE					RUNSTBY	ENABLE	SWRST
0x01	Reserved									
0x02	SYNCBUSY	7:0							ENABLE	SWRST
0x03	QOSCTRL	7:0					DQOS[1:0]		CQOS[1:0]	
0x0D	FSMSTATUS	7:0		FSMSTATE[6:0]						
0x24	DESCADD	7:0	DESCADD[7:0]							
0x25		15:8	DESCADD[15:8]							
0x26		23:16	DESCADD[23:16]							
0x27		31:24	DESCADD[31:24]							
0x28	PADCAL	7:0	TRANSN[1:0]			TRANSP[4:0]				
0x29		15:8		TRIM[2:0]				TRANSN[4:2]		

39.7.2. Device Summary

Table 39-1. General Device Registers

Offset	Name	Bit Pos.								
0x04	Reserved									
0x05	Reserved									
0x06	Reserved									
0x07	Reserved									
0x08	CTRLB	7:0				NREPLY	SPDCONF[1:0]		UPRSM	DETACH
0x09		15:8					LPMHDSK[1:0]		GNAK	
0x0A	DADD		ADDEN	DADD[6:0]						
0x0B	Reserved									
0x0C	STATUS	7:0	LINESTATE[1:0]				SPEED[1:0]			
0x0E	Reserved									
0x0F	Reserved									
0x10	FNUM	7:0	FNUM[4:0]							
0x11		15:8	FNCERR		FNUM[10:5]					
0x12	Reserved									
0x14	INTENCLR	7:0	RAMACER	UPRSM	EORSM	WAKEUP	EORST	SOF		SUSPEND
0x15		15:8							LPMSUSP	LPMNYET
0x16	Reserved									
0x17	Reserved									
0x18	INTENSET	7:0	RAMACER	UPRSM	EORSM	WAKEUP	EORST	SOF		SUSPEND
0x19		15:8							LPMSUSP	LPMNYET
0x1A	Reserved									
0x1B	Reserved									
0x1C	INTFLAG	7:0	RAMACER	UPRSM	EORSM	WAKEUP	EORST	SOF		SUSPEND
0x1D		15:8							LPMSUSP	LPMNYET

Offset	Name	Bit Pos.								
0x1E	Reserved									
0x1F	Reserved									
0x20	EPINTSMRY	7:0	EPINT[7:0]							
0x21		15:8	EPINT[15:8]							
0x22	Reserved									
0x23	Reserved									

Table 39-2. Device Endpoint Register n

Offset	Name	Bit Pos.								
0x1m0	EPCFGn	7:0		EPTYPE1[1:0]				EPTYPE0[1:0]		
0x1m1	Reserved									
0x1m2	Reserved									
0x1m3	Reserved									
0x1m4	EPSTATUSCLRn	7:0	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
0x1m5	EPSTATUSSETn	7:0	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
0x1m6	EPSTATUSn	7:0	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
0x1m7	EPINTFLAGn	7:0		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0
0x1m8	EPINTENCLRn	7:0		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0
0x1m9	EPINTENSETn	7:0		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0
0x1mA	Reserved									
0x1mB	Reserved									

Table 39-3. Device Endpoint n Descriptor Bank 0

Offset 0x n0 + index	Name	Bit Pos.								
0x00	ADDR	7:0	ADD[7:0]							
0x01		15:8	ADD[15:8]							
0x02		23:16	ADD[23:16]							
0x03		31:24	ADD[31:24]							
0x04	PCKSIZE	7:0	BYTE_COUNT[7:0]							
0x05		15:8	MULTI_PACKET_SIZE[1:0]		BYTE_COUNT[13:8]					
0x06		23:16	MULTI_PACKET_SIZE[9:2]							
0x07		31:24	AUTO_ZLP	SIZE[2:0]			MULTI_PACKET_SIZE[13:10]			
0x08	EXTREG	7:0	VARIABLE[3:0]				SUBPID[3:0]			
0x09		15:8		VARIABLE[10:4]						
0x0A	STATUS_BK	7:0							ERRORFLOW	CRCERR
0x0B	Reserved	7:0								
0x0C	Reserved	7:0								
0x0D	Reserved	7:0								
0x0E	Reserved	7:0								
0x0F	Reserved	7:0								

Table 39-4. Device Endpoint n Descriptor Bank 1

Offset 0x n0 + 0x10 + index	Name	Bit Pos.								
0x00	ADDR	7:0	ADD[7:0]							
0x01		15:8	ADD[15:8]							
0x02		23:16	ADD[23:16]							
0x03		31:24	ADD[31:24]							
0x04	PCKSIZE	7:0	BYTE_COUNT[7:0]							
0x05		15:8	MULTI_PACKET_SIZE[1:0]		BYTE_COUNT[13:8]					
0x06		23:16	MULTI_PACKET_SIZE[9:2]							
0x07		31:24	AUTO_ZLP	SIZE[2:0]			MULTI_PACKET_SIZE[13:10]			
0x08	Reserved	7:0								
0x09	Reserved	15:8								
0x0A	STATUS_BK	7:0							ERRORFLOW	CRCERR
0x0B	Reserved	7:0								
0x0C	Reserved	7:0								
0x0D	Reserved	7:0								
0x0E	Reserved	7:0								
0x0F	Reserved	7:0								

39.7.3. Host Summary

Table 39-5. General Host Registers

Offset	Name	Bit Pos.								
0x04	Reserved									
0x05	Reserved									
0x06	Reserved									
0x07	Reserved									
0x08	CTRLB	7:0		TSTK	TSTJ		SPDCONF[1:0]		RESUME	
0x09		15:8					L1RESUME	VBUSOK	BUSRESET	SOFE
0x0A	HSOFC	7:0	FLENCE				FLENC[3:0]			
0x0B	Reserved									
0x0C	STATUS	7:0	LINESTATE[1:0]				SPEED[1:0]			
0x0E	Reserved									
0x0F	Reserved									
0x10	FNUM	7:0	FNUM[4:0]							
0x11		15:8			FNUM[10:5]					
0x12	FLENHIGH	7:0	FLENHIGH[7:0]							
0x14	INTENCLR	7:0	RAMACER	UPRSM	DNRSM	WAKEUP	RST	HSOF		
0x15		15:8							DDISC	DCONN
0x16	Reserved									
0x17	Reserved									
0x18	INTENSET	7:0	RAMACER	UPRSM	DNRSM	WAKEUP	RST	HSOF		
0x19		15:8							DDISC	DCONN
0x1A	Reserved									
0x1B	Reserved									

Offset	Name	Bit Pos.								
0x1C	INTFLAG	7:0	RAMACER	UPRSM	DNRSM	WAKEUP	RST	HSOF		
0x1D		15:8							DDISC	DCONN
0x1E	Reserved									
0x1F	Reserved									
0x20	PINTSMRY	7:0	PINT[7:0]							
0x21		15:8	PINT[15:8]							
0x22	Reserved									
0x23										

Table 39-6. Host Pipe Register n

Offset	Name	Bit Pos.								
0x1m0	PCFGn	7:0			PTYPE[2:0]			BK	PTOKEN[1:0]	
0x1m1	Reserved									
0x1m2	Reserved									
0x1m3	BINTERVAL	7:0	BINTERVAL[7:0]							
0x1m4	PSTATUSCLRn	7:0	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL
0x1m5	PSTATUSETn	7:0	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL
0x1m6	PSTATUSn	7:0	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL
0x1m7	PINTFLAGn	7:0			STALL	TXSTP	PERR	TRFAIL	TRCPT1	TRCPT0
0x1m8	PINTENCLRn	7:0			STALL	TXSTP	PERR	TRFAIL	TRCPT1	TRCPT0
0x1m9	PINTENSETn	7:0			STALL	TXSTP	PERR	TRFAIL	TRCPT1	TRCPT0
0x1mA	Reserved									
0x1mB	Reserved									

Table 39-7. Host Pipe n Descriptor Bank 0

Offset 0x n0 + index	Name	Bit Pos.								
0x00	ADDR	7:0	ADD[7:0]							
0x01		15:8	ADD[15:8]							
0x02		23:16	ADD[23:16]							
0x03		31:24	ADD[31:24]							
0x04	PCKSIZE	7:0	BYTE_COUNT[7:0]							
0x05		15:8	MULTI_PACKET_SIZE[1:0]		BYTE_COUNT[13:8]					
0x06		23:16	MULTI_PACKET_SIZE[9:2]							
0x07		31:24	AUTO_ZLP	SIZE[2:0]			MULTI_PACKET_SIZE[13:10]			
0x08	EXTREG	7:0	VARIABLE[3:0]				SUBPID[3:0]			
0x09		15:8		VARIABLE[10:4]						
0x0A	STATUS_BK	7:0							ERRORFLOW	CRCERR
0x0B		15:8								
0x0C	CTRL_PIPE	7:0		PDADDR[6:0]						
0x0D		15:8	PEPMAX[3:0]				PEPNUM[3:0]			
0x0E	STATUS_PIPE	7:0	ERCNT[2:0]			CRC16ER	TOUTER	PIDER	DAPIDER	DTGLER
0x0F		15:8								

Table 39-8. Host Pipe n Descriptor Bank 1

Offset 0x n0 +0x10 +index	Name	Bit Pos.								
0x00	ADDR	7:0	ADD[7:0]							
0x01		15:8	ADD[15:8]							
0x02		23:16	ADD[23:16]							
0x03		31:24	ADD[31:24]							
0x04	PCKSIZE	7:0	BYTE_COUNT[7:0]							
0x05		15:8	MULTI_PACKET_SIZE[1:0		BYTE_COUNT[13:8]					
0x06		23:16	MULTI_PACKET_SIZE[9:2]							
0x07		31:24	AUTO_ZLP	SIZE[2:0]			MULTI_PACKET_SIZE[13:10]			
0x08		7:0								
0x09		15:8								
0x0A	STATUS_BK	7:0							ERRORFLOW	CRCERR
0x0B		15:8								
0x0C		7:0								
0x0D		15:8								
0x0E	STATUS_PIPE	7:0		ERCNT[2:0]		CRC16ER	TOUTER	PIDER	DAPIDER	DTGLER
0x0F		15:8								

39.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

Refer to the Register Access Protection on page 892, *PAC - Peripheral Access Controller* and *GCLK Synchronization* for details.

Related Links

PAC - Peripheral Access Controller on page 58

39.8.1. Communication Device Host Registers

39.8.1.1. Control A

Name: CTRLA
Offset: 0x00
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronised

Bit	7	6	5	4	3	2	1	0
	MODE					RUNSDTBY	ENABLE	SWRST
Access	R/W					R/W	R/W	R/W
Reset	0					0	0	0

Bit 7 – MODE: Operating Mode
This bit defines the operating mode of the USB.

Value	Description
0	USB Device mode
1	USB Host mode

Bit 2 – RUNSDTBY: Run in Standby Mode
This bit is Enable-Protected.

Value	Description
0	USB clock is stopped in standby mode.
1	USB clock is running in standby mode

Bit 1 – ENABLE: Enable
Due to synchronization there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the Synchronization status enable bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

This bit is Write-Synchronized.

Value	Description
0	The peripheral is disabled or being disabled.
1	The peripheral is enabled or being enabled.

Bit 0 – SWRST: Software Reset
Writing a zero to this bit has no effect.

Writing a '1' to this bit resets all registers in the USB, to their initial state, and the USB will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.

Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

This bit is Write-Synchronized.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

39.8.1.2. Synchronization Busy

Name: SYNCBUSY
Offset: 0x02
Reset: 0x0000
Property: -

Bit	7	6	5	4	3	2	1	0
							ENABLE	SWRST
Access							R	R
Reset							0	0

Bit 1 – ENABLE: Synchronization Enable status bit
This bit is cleared when the synchronization of ENABLE register between the clock domains is complete.

This bit is set when the synchronization of ENABLE register between clock domains is started.

Bit 0 – SWRST: Synchronization Software Reset status bit
This bit is cleared when the synchronization of SWRST register between the clock domains is complete.

This bit is set when the synchronization of SWRST register between clock domains is started.

39.8.1.3. QOS Control

Name: QOSCTRL
Offset: 0x03
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
					DQOS[1:0]		CQOS[1:0]	
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	1

Bits 3:2 – DQOS[1:0]: Data Quality of Service
These bits define the memory priority access during the endpoint or pipe read/write data operation. Refer to *SRAM Quality of Service*.

Bits 1:0 – CQOS[1:0]: Configuration Quality of Service
These bits define the memory priority access during the endpoint or pipe read/write configuration operation. Refer to *SRAM Quality of Service*.

39.8.1.4. Finite State Machine Status

Name: FSMSTATUS
Offset: 0x0D
Reset: 0xXXXX
Property: Read only

Bit	7	6	5	4	3	2	1	0
		FSMSTATE[6:0]						
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	1

Bits 6:0 – FSMSTATE[6:0]: Fine State Machine Status
These bits indicate the state of the finite state machine of the USB controller.

Value	Name	Description
0x01	OFF (L3)	Corresponds to the powered-off, disconnected, and disabled state.
0x02	ON (L0)	Corresponds to the Idle and Active states.
0x04	SUSPEND (L2)	
0x08	SLEEP (L1)	
0x10	DNRESUME	Down Stream Resume.
0x20	UPRESUME	Up Stream Resume.
0x40	RESET	USB lines Reset.
Others		Reserved

39.8.1.5. Descriptor Address

Name: DESCADD
Offset: 0x24
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	DESCADD[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DESCADD[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DESCADD[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DESCADD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DESCADD[31:0]: Descriptor Address Value
These bits define the base address of the main USB descriptor in RAM. The two least significant bits must be written to zero.

39.8.1.6. Pad Calibration

The Pad Calibration values must be loaded from the NVM Software Calibration Area into the USB Pad Calibration register by software, before enabling the USB, to achieve the specified accuracy.
Refer to *NVM Software Calibration Area Mapping* for further details.

Refer to for further details.

Name: PADCAL
Offset: 0x28
Reset: 0x0000
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
		TRIM[2:0]				TRANSN[4:2]		
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bit	7	6	5	4	3	2	1	0
	TRANSN[1:0]			TRANSP[4:0]				
Access	R/W	R/W		R/W	R/W	R/W	R/W	R/W
Reset	0	0		0	0	0	0	0

Bits 14:12 – TRIM[2:0]: Trim bits for DP/DM
These bits calibrate the matching of rise/fall of DP/DM.

Bits 10:6 – TRANSN[4:0]: Trimmable Output Driver Impedance N
These bits calibrate the NMOS output impedance of DP/DM drivers.

Bits 4:0 – TRANSP[4:0]: Trimmable Output Driver Impedance P
These bits calibrate the PMOS output impedance of DP/DM drivers.

39.8.2. Device Registers - Common

39.8.2.1. Control B

Name: CTRLB
Offset: 0x08
Reset: 0x0001
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
					LPMHDSK[1:0]		GNAK	
Access					R/W	R/W	R/W	
Reset					0	0	0	

Bit	7	6	5	4	3	2	1	0
				NREPLY	SPDCONF[1:0]		UPRSM	DETACH
Access				R	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bits 11:10 – LPMHDSK[1:0]: Link Power Management Handshake
These bits select the Link Power Management Handshake configuration.

Value	Description
0x0	No handshake. LPM is not supported.
0x1	ACK
0x2	NYET
0x3	Reserved

Bit 9 – GNAK: Global NAK
This bit configures the operating mode of the NAK.

This bit is not synchronized.

Value	Description
0	The handshake packet reports the status of the USB transaction
1	A NAK handshake is answered for each USB transaction regardless of the current endpoint memory bank status

Bit 4 – NREPLY: No reply excepted SETUP Token
This bit is cleared by hardware when receiving a SETUP packet.

This bit has no effect for any other endpoint but endpoint 0.

Value	Description
0	Disable the “NO_REPLY” feature: Any transaction to endpoint 0 will be handled according to the USB2.0 standard.
1	Enable the “NO_REPLY” feature: Any transaction to endpoint 0 will be ignored except SETUP.

Bits 3:2 – SPDCONF[1:0]: Speed Configuration
These bits select the speed configuration.

Value	Description
0x0	LS: Low-speed
0x1	FS: Full-speed
0x2	Reserved
0x3	Reserved

Bit 1 – UPRSM: Upstream Resume
This bit is cleared when the USB receives a USB reset or once the upstream resume has been sent.

Value	Description
0	Writing a zero to this bit has no effect.
1	Writing a one to this bit will generate an upstream resume to the host for a remote wakeup.

Bit 0 – DETACH: Detach

Value	Description
0	The device is attached to the USB bus so that communications may occur.
1	It is the default value at reset. The internal device pull-ups are disabled, removing the device from the USB bus.

39.8.2.2. Device Address

Name: DADD
Offset: 0x0A
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	ADDEN	DADD[6:0]						
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 7 – ADDEN: Device Address Enable
This bit is cleared when a USB reset is received.

Value	Description
0	Writing a zero will deactivate the DADD field (USB device address) and return the device to default address 0.
1	Writing a one will activate the DADD field (USB device address).

Bits 6:0 – DADD[6:0]: Device Address
These bits define the device address. The DADD register is reset when a USB reset is received.

39.8.2.3. Status

Name: STATUS
Offset: 0x0C
Reset: 0x0000
Property: -

Bit	7	6	5	4	3	2	1	0
	LINESTATE[1:0]				SPEED[1:0]			
Access	R	R			R	R		
Reset	0	1			0	1		

Bits 7:6 – LINESTATE[1:0]: USB Line State Status
These bits define the current line state DP/DM.

LINESTATE[1:0]	USB Line Status
0x0	SE0/RESET
0x1	FS-J or LS-K State
0x2	FS-K or LS-J State

Bits 3:2 – SPEED[1:0]: Speed Status
These bits define the current speed used of the device

.

SPEED[1:0]	SPEED STATUS
0x0	Low-speed mode
0x1	Full-speed mode
0x2	Reserved
0x3	Reserved

39.8.2.4. Device Frame Number

Name: FNUM
Offset: 0x10
Reset: 0x0000
Property: Read only

Bit	15	14	13	12	11	10	9	8
	FNCERR		FNUM[10:5]					
Access	R		R	R	R	R	R	R
Reset	0		0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FNUM[4:0]					MFNUM[2:0]		
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 15 – FNCERR: Frame Number CRC Error
This bit is cleared upon receiving a USB reset.

This bit is set when a corrupted frame number (or micro-frame number) is received.

This bit and the SOF (or MSOF) interrupt bit are updated at the same time.

Bits 13:3 – FNUM[10:0]: Frame Number
These bits are cleared upon receiving a USB reset.

These bits are updated with the frame number information as provided from the last SOF packet even if a corrupted SOF is received.

Bits 2:0 – MFNUM[2:0]: Micro Frame Number
These bits are cleared upon receiving a USB reset or at the beginning of each Start-of-Frame (SOF interrupt).

These bits are updated with the micro-frame number information as provided from the last MSOF packet even if a corrupted MSOF is received.

39.8.2.5. Device Interrupt Enable Clear

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Name: INTENCLR
Offset: 0x14
Reset: 0x0000
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
							LPMSUSP	LPMNYET
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	RAMACER	UPRSM	EORSM	WAKEUP	EORST	SOF		SUSPEND
Access	R/W	R/W	R/W	R/W	R/W	R/W		R/W
Reset	0	0	0	0	0	0		0

Bit 9 – LPMSUSP: Link Power Management Suspend Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Link Power Management Suspend Interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Link Power Management Suspend interrupt is disabled.
1	The Link Power Management Suspend interrupt is enabled and an interrupt request will be generated when the Link Power Management Suspend interrupt Flag is set.

Bit 8 – LPMNYET: Link Power Management Not Yet Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Link Power Management Not Yet interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Link Power Management Not Yet interrupt is disabled.
1	The Link Power Management Not Yet interrupt is enabled and an interrupt request will be generated when the Link Power Management Not Yet interrupt Flag is set.

Bit 7 – RAMACER: RAM Access Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the RAM Access interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The RAM Access interrupt is disabled.
1	The RAM Access interrupt is enabled and an interrupt request will be generated when the RAM Access interrupt Flag is set.

Bit 6 – UPRSM: Upstream Resume Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Upstream Resume interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Upstream Resume interrupt is disabled.
1	The Upstream Resume interrupt is enabled and an interrupt request will be generated when the Upstream Resume interrupt Flag is set.

Bit 5 – EORSM: End Of Resume Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the End Of Resume interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The End Of Resume interrupt is disabled.
1	The End Of Resume interrupt is enabled and an interrupt request will be generated when the End Of Resume interrupt Flag is set.

Bit 4 – WAKEUP: Wake-Up Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Wake Up interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Wake Up interrupt is disabled.
1	The Wake Up interrupt is enabled and an interrupt request will be generated when the Wake Up interrupt Flag is set.

Bit 3 – EORST: End of Reset Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the End of Reset interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The End of Reset interrupt is disabled.
1	The End of Reset interrupt is enabled and an interrupt request will be generated when the End of Reset interrupt Flag is set.

Bit 2 – SOF: Start-of-Frame Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Start-of-Frame interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Start-of-Frame interrupt is disabled.
1	The Start-of-Frame interrupt is enabled and an interrupt request will be generated when the Start-of-Frame interrupt Flag is set.

Bit 0 – SUSPEND: Suspend Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Suspend Interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Suspend interrupt is disabled.
1	The Suspend interrupt is enabled and an interrupt request will be generated when the Suspend interrupt Flag is set.

39.8.2.6. Device Interrupt Enable Set

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Name: INTENSET
Offset: 0x18
Reset: 0x0000
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
							LPMSUSP	LPMNYET
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	RAMACER	UPRSM	EORSM	WAKEUP	EORST	SOF		SUSPEND
Access	R/W	R/W	R/W	R/W	R/W	R/W		R/W
Reset	0	0	0	0	0	0		0

Bit 9 – LPMSUSP: Link Power Management Suspend Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will set the Link Power Management Suspend Enable bit and enable the corresponding interrupt request.

Value	Description
0	The Link Power Management Suspend interrupt is disabled.
1	The Link Power Management Suspend interrupt is enabled.

Bit 8 – LPMNYET: Link Power Management Not Yet Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will set the Link Power Management Not Yet interrupt bit and enable the corresponding interrupt request.

Value	Description
0	The Link Power Management Not Yet interrupt is disabled.
1	The Link Power Management Not Yet interrupt is enabled.

Bit 7 – RAMACER: RAM Access Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will set the RAM Access Enable bit and enable the corresponding interrupt request.

Value	Description
0	The RAM Access interrupt is disabled.
1	The RAM Access interrupt is enabled.

Bit 6 – UPRSM: Upstream Resume Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will set the Upstream Resume Enable bit and enable the corresponding interrupt request.

Value	Description
0	The Upstream Resume interrupt is disabled.
1	The Upstream Resume interrupt is enabled.

Bit 5 – EORSM: End Of Resume Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will set the End Of Resume interrupt Enable bit and enable the corresponding interrupt request.

Value	Description
0	The End Of Resume interrupt is disabled.
1	The End Of Resume interrupt is enabled.

Bit 4 – WAKEUP: Wake-Up Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will set the Wake Up interrupt Enable bit and enable the corresponding interrupt request.

Value	Description
0	The Wake Up interrupt is disabled.
1	The Wake Up interrupt is enabled.

Bit 3 – EORST: End of Reset Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will set the End of Reset interrupt Enable bit and enable the corresponding interrupt request.

Value	Description
0	The End of Reset interrupt is disabled.
1	The End of Reset interrupt is enabled.

Bit 2 – SOF: Start-of-Frame Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will set the Start-of-Frame interrupt Enable bit and enable the corresponding interrupt request.

Value	Description
0	The Start-of-Frame interrupt is disabled.
1	The Start-of-Frame interrupt is enabled.

Bit 0 – SUSPEND: Suspend Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will set the Suspend interrupt Enable bit and enable the corresponding interrupt request.

Value	Description
0	The Suspend interrupt is disabled.
1	The Suspend interrupt is enabled.

39.8.2.7. Device Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x01C
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
							LPMSUSP	LPMNYET
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	RAMACER	UPRSM	EORSM	WAKEUP	EORST	SOF		SUSPEND
Access	R/W	R/W	R/W	R/W	R/W	R/W		R/W
Reset	0	0	0	0	0	0		0

Bit 9 – LPMSUSP: Link Power Management Suspend Interrupt Flag
This flag is cleared by writing a one to the flag.

This flag is set when the USB module acknowledge a Link Power Management Transaction (ACK handshake) and has entered the Suspended state and will generate an interrupt if INTENCLR/SET.LPMSUSP is one.

Writing a zero to this bit has no effect.

Writing a one to this bit clears the LPMSUSP Interrupt Flag.

Bit 8 – LPMNYET: Link Power Management Not Yet Interrupt Flag
This flag is cleared by writing a one to the flag.

This flag is set when the USB module acknowledges a Link Power Management Transaction (handshake is NYET) and will generate an interrupt if INTENCLR/SET.LPMNYET is one.

Writing a zero to this bit has no effect.

Writing a one to this bit clears the LPMNYET Interrupt Flag.

Bit 7 – RAMACER: RAM Access Interrupt Flag
This flag is cleared by writing a one to the flag.

This flag is set when a RAM access underflow error occurs during IN data stage. This bit will generate an interrupt if INTENCLR/SET.RAMACER is one.

Writing a zero to this bit has no effect.

Bit 6 – UPRSM: Upstream Resume Interrupt Flag
This flag is cleared by writing a one to the flag.

This flag is set when the USB sends a resume signal called “Upstream Resume” and will generate an interrupt if INTENCLR/SET.UPRSM is one.

Writing a zero to this bit has no effect.

Bit 5 – EORSM: End Of Resume Interrupt Flag
This flag is cleared by writing a one to the flag.

This flag is set when the USB detects a valid “End of Resume” signal initiated by the host and will generate an interrupt if INTENCLR/SET.EORSM is one.

Writing a zero to this bit has no effect.

Bit 4 – WAKEUP: Wake Up Interrupt Flag
This flag is cleared by writing a one to the flag.

This flag is set when the USB is reactivated by a filtered non-idle signal from the lines and will generate an interrupt if INTENCLR/SET.WAKEUP is one.

Writing a zero to this bit has no effect.

Bit 3 – EORST: End of Reset Interrupt Flag
This flag is cleared by writing a one to the flag.

This flag is set when a USB “End of Reset” has been detected and will generate an interrupt if INTENCLR/SET.EORST is one.

Writing a zero to this bit has no effect.

Bit 2 – SOF: Start-of-Frame Interrupt Flag
This flag is cleared by writing a one to the flag.

This flag is set when a USB “Start-of-Frame” has been detected (every 1 ms) and will generate an interrupt if INTENCLR/SET.SOF is one.

The FNUM is updated. In High Speed mode, the MFNUM register is cleared.

Writing a zero to this bit has no effect.

Bit 0 – SUSPEND: Suspend Interrupt Flag
This flag is cleared by writing a one to the flag.

This flag is set when a USB “Suspend” idle state has been detected for 3 frame periods (J state for 3 ms) and will generate an interrupt if INTENCLR/SET.SUSPEND is one.

Writing a zero to this bit has no effect.

39.8.2.8. Endpoint Interrupt Summary

Name: EPINTSMRY
Offset: 0x20
Reset: 0x00000000
Property: -

Bit	15	14	13	12	11	10	9	8
	EPINT[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	EPINT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – EPINT[15:0]: EndPoint Interrupt
The flag EPINT[n] is set when an interrupt is triggered by the EndPoint n. See EPINTFLAGn on page 941 register in the device EndPoint section.

This bit will be cleared when no interrupts are pending for EndPoint n.

39.8.3. Device Registers - Endpoint

39.8.3.1. Device Endpoint Configuration register n

Name: EPCFGn
Offset: 0x100 + (n x 0x20)
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
		EPTYPE1[2:0]				EPTYPE0[2:0]		
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bits 6:4 – EPTYPE1[2:0]: Endpoint Type for IN direction
These bits contains the endpoint type for IN direction.

Upon receiving a USB reset EPCFGn.EPTYPE1 is cleared except for endpoint 0 which is unchanged.

Value	Description
0x0	Bank1 is disabled.
0x1	Bank1 is enabled and configured as Control IN.
0x2	Bank1 is enabled and configured as Isochronous IN.
0x3	Bank1 is enabled and configured as Bulk IN.
0x4	Bank1 is enabled and configured as Interrupt IN.
0x5	Bank1 is enabled and configured as Dual-Bank OUT (Endpoint type is the same as the one defined in EPTYPE0)
0x6-0x7	Reserved

Bits 2:0 – EPTYPE0[2:0]: Endpoint Type for OUT direction
These bits contains the endpoint type for OUT direction.

Upon receiving a USB reset EPCFGn.EPTYPE0 is cleared except for endpoint 0 which is unchanged.

Value	Description
0x0	Bank0 is disabled.
0x1	Bank0 is enabled and configured as Control SETUP / Control OUT.
0x2	Bank0 is enabled and configured as Isochronous OUT.
0x3	Bank0 is enabled and configured as Bulk OUT.
0x4	Bank0 is enabled and configured as Interrupt OUT.
0x5	Bank0 is enabled and configured as Dual Bank IN (Endpoint type is the same as the one defined in EPTYPE1)
0x6-0x7	Reserved

39.8.3.2. EndPoint Status Clear n

Name: EPSTATUSCLRn
Offset: 0x104 + (n * 0x20)
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bit 7 – BK1RDY: Bank 1 Ready
Writing a zero to this bit has no effect.

Writing a one to this bit will clear EPSTATUS.BK1RDY bit.

Bit 6 – BK0RDY: Bank 0 Ready
Writing a zero to this bit has no effect.

Writing a one to this bit will clear EPSTATUS.BK0RDY bit.

Bit 5 – STALLRQ1: STALL bank 1 Request
Writing a zero to this bit has no effect.

Writing a one to this bit will clear EPSTATUS.STALLRQ1 bit.

Bit 4 – STALLRQ0: STALL bank 0 Request
Writing a zero to this bit has no effect.

Writing a one to this bit will clear EPSTATUS.STALLRQ0 bit.

Bit 2 – CURBK: Current Bank
Writing a zero to this bit has no effect.

Writing a one to this bit will clear EPSTATUS.CURBK bit.

Bit 1 – DTGLIN: Data Toggle IN
Writing a zero to this bit has no effect.

Writing a one to this bit will clear EPSTATUS.DTGLIN bit.

Bit 0 – DTGLOUT: Data Toggle OUT
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the EPSTATUS.DTGLOUT bit.

39.8.3.3. EndPoint Status Set n

Name: EPSTATUSSETn
Offset: 0x105 + (n x 0x20)
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bit 7 – BK1RDY: Bank 1 Ready
Writing a zero to this bit has no effect.

Writing a one to this bit will set EPSTATUS.BK1RDY bit.

Bit 6 – BK0RDY: Bank 0 Ready
Writing a zero to this bit has no effect.

Writing a one to this bit will set EPSTATUS.BK0RDY bit.

Bit 5 – STALLRQ1: STALL Request bank 1
Writing a zero to this bit has no effect.

Writing a one to this bit will set EPSTATUS.STALLRQ1 bit.

Bit 4 – STALLRQ0: STALL Request bank 0
Writing a zero to this bit has no effect.

Writing a one to this bit will set EPSTATUS.STALLRQ0 bit.

Bit 2 – CURBK: Current Bank
Writing a zero to this bit has no effect.

Writing a one to this bit will set EPSTATUS.CURBK bit.

Bit 1 – DTGLIN: Data Toggle IN
Writing a zero to this bit has no effect.

Writing a one to this bit will set EPSTATUS.DTGLIN bit.

Bit 0 – DTGLOUT: Data Toggle OUT
Writing a zero to this bit has no effect.

Writing a one to this bit will set the EPSTATUS.DTGLOUT bit.

39.8.3.4. EndPoint Status n

Name: EPSTATUSn
Offset: 0x106 + (n x 0x20)
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
Access	R	R	R	R		R	R	R
Reset	0	0	0	0		0	0	0

Bit 7 – BK1RDY: Bank 1 is ready
For Control/OUT direction Endpoints, the bank is empty.

Writing a one to the bit EPSTATUSCLR.BK1RDY will clear this bit.

Writing a one to the bit EPSTATUSSET.BK1RDY will set this bit.

Value	Description
0	The bank number 1 is not ready : For IN direction Endpoints, the bank is not yet filled in.
1	The bank number 1 is ready: For IN direction Endpoints, the bank is filled in. For Control/OUT direction Endpoints, the bank is full.

Bit 6 – BK0RDY: Bank 0 is ready
Writing a one to the bit EPSTATUSCLR.BK0RDY will clear this bit.

Writing a one to the bit EPSTATUSSET.BK0RDY will set this bit.

Value	Description
0	The bank number 0 is not ready : For IN direction Endpoints, the bank is not yet filled in. For Control/OUT direction Endpoints, the bank is empty.
1	The bank number 0 is ready: For IN direction Endpoints, the bank is filled in. For Control/OUT direction Endpoints, the bank is full.

Bit 5 – STALLRQ1: STALL bank 1 request
Writing a zero to the bit EPSTATUSCLR.STALLRQ1 will clear this bit.

Writing a one to the bit EPSTATUSSET.STALLRQ1 will set this bit.

This bit is cleared by hardware when receiving a SETUP packet.

Value	Description
0	Disable STALLRQ1 feature.
1	Enable STALLRQ1 feature: a STALL handshake will be sent to the host in regards to bank1.

Bit 4 – STALLRQ0: STALL bank 0 request
Writing a zero to the bit EPSTATUSCLR.STALLRQ0 will clear this bit.

Writing a one to the bit EPSTATUSSET.STALLRQ0 will set this bit.

This bit is cleared by hardware when receiving a SETUP packet.

Value	Description
0	Disable STALLRQ0 feature.
1	Enable STALLRQ0 feature: a STALL handshake will be sent to the host in regards to bank0.

Bit 2 – CURBK: Current Bank
Writing a zero to the bit EPSTATUSCLR.CURBK will clear this bit.

Writing a one to the bit EPSTATUSSET.CURBK will set this bit.

Value	Description
0	The bank0 is the bank that will be used in the next single/multi USB packet.
1	The bank1 is the bank that will be used in the next single/multi USB packet.

Bit 1 – DTGLIN: Data Toggle IN Sequence
Writing a zero to the bit EPSTATUSCLR.DTGLINCLR will clear this bit.

Writing a one to the bit EPSTATUSSET.DTGLINSET will set this bit.

Value	Description
0	The PID of the next expected IN transaction will be zero: data 0.
1	The PID of the next expected IN transaction will be one: data 1.

Bit 0 – DTGLOUT: Data Toggle OUT Sequence
Writing a zero to the bit EPSTATUSCLR.DTGLOUTCLR will clear this bit.

Writing a one to the bit EPSTATUSSET.DTGLOUTSET will set this bit.

Value	Description
0	The PID of the next expected OUT transaction will be zero: data 0.
1	The PID of the next expected OUR transaction will be one: data 1.

39.8.3.5. Device EndPoint Interrupt Flag n

Name: EPINTFLAGn
Offset: 0x107 + (n x 0x20)
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit 6 – STALL1: Transmit Stall 1 Interrupt Flag
This flag is cleared by writing a one to the flag.

This flag is set when a Transmit Stall occurs and will generate an interrupt if EPINTENCLR/SET.STALL1 is one.

EPINTFLAG.STALL1 is set for a single bank IN endpoint or double bank IN/OUT endpoint when current bank is "1".

Writing a zero to this bit has no effect.

Writing a one to this bit clears the STALL1 Interrupt Flag.

Bit 5 – STALL0: Transmit Stall 0 Interrupt Flag
This flag is cleared by writing a one to the flag.

This flag is set when a Transmit Stall occurs and will generate an interrupt if EPINTENCLR/SET.STALL0 is one.

EPINTFLAG.STALL0 is set for a single bank OUT endpoint or double bank IN/OUT endpoint when current bank is "0".

Writing a zero to this bit has no effect.

Writing a one to this bit clears the STALL0 Interrupt Flag.

Bit 4 – RXSTP: Received Setup Interrupt Flag
This flag is cleared by writing a one to the flag.

This flag is set when a Received Setup occurs and will generate an interrupt if EPINTENCLR/SET.RXSTP is one.

Writing a zero to this bit has no effect.

Writing a one to this bit clears the RXSTP Interrupt Flag.

Bit 3 – TRFAIL1: Transfer Fail 1 Interrupt Flag
This flag is cleared by writing a one to the flag.

This flag is set when a transfer fail occurs and will generate an interrupt if EPINTENCLR/SET.TRFAIL1 is one.

EPINTFLAG.TRFAIL1 is set for a single bank IN endpoint or double bank IN/OUT endpoint when current bank is "1".

Writing a zero to this bit has no effect.

Writing a one to this bit clears the TRFAIL1 Interrupt Flag.

Bit 2 – TRFAIL0: Transfer Fail 0 Interrupt Flag
This flag is cleared by writing a one to the flag.

This flag is set when a transfer fail occurs and will generate an interrupt if EPINTENCLR/SET.TRFAIL0 is one.

EPINTFLAG.TRFAIL0 is set for a single bank OUT endpoint or double bank IN/OUT endpoint when current bank is "0".

Writing a zero to this bit has no effect.

Writing a one to this bit clears the TRFAIL0 Interrupt Flag.

Bit 1 – TRCPT1: Transfer Complete 1 interrupt Flag
This flag is cleared by writing a one to the flag.

This flag is set when a Transfer Complete occurs and will generate an interrupt if EPINTENCLR/SET.TRCPT1 is one. EPINTFLAG.TRCPT1 is set for a single bank IN endpoint or double bank IN/OUT endpoint when current bank is "1".

Writing a zero to this bit has no effect.

Writing a one to this bit clears the TRCPT1 Interrupt Flag.

Bit 0 – TRCPT0: Transfer Complete 0 interrupt Flag
This flag is cleared by writing a one to the flag.

This flag is set when a Transfer complete occurs and will generate an interrupt if EPINTENCLR/SET.TRCPT0 is one. EPINTFLAG.TRCPT0 is set for a single bank OUT endpoint or double bank IN/OUT endpoint when current bank is "0".

Writing a zero to this bit has no effect.

Writing a one to this bit clears the TRCPT0 Interrupt Flag.

39.8.3.6. Device EndPoint Interrupt Enable n

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Endpoint Interrupt Enable Set (EPINTENSET) register.

Name: EPINTENCLRn
Offset: 0x108 + (n x 0x20)
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit 6 – STALL1: Transmit STALL 1 Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Transmit Stall 1 Interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Transmit Stall 1 interrupt is disabled.
1	The Transmit Stall 1 interrupt is enabled and an interrupt request will be generated when the Transmit Stall 1 Interrupt Flag is set.

Bit 5 – STALL0: Transmit STALL 0 Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Transmit Stall 0 Interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Transmit Stall 0 interrupt is disabled.
1	The Transmit Stall 0 interrupt is enabled and an interrupt request will be generated when the Transmit Stall 0 Interrupt Flag is set.

Bit 4 – RXSTP: Received Setup Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Received Setup Interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Received Setup interrupt is disabled.
1	The Received Setup interrupt is enabled and an interrupt request will be generated when the Received Setup Interrupt Flag is set.

Bit 3 – TRFAIL1: Transfer Fail 1 Interrupt Enable
The user should look into the descriptor table status located in ram to be informed about the error condition : ERRORFLOW, CRC.

Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Transfer Fail 1 Interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Transfer Fail 1 interrupt is disabled.
1	The Transfer Fail 1 interrupt is enabled and an interrupt request will be generated when the Transfer Fail 1 Interrupt Flag is set.

Bit 2 – TRFAIL0: Transfer Fail 0 Interrupt Enable
The user should look into the descriptor table status located in ram to be informed about the error condition : ERRORFLOW, CRC.

Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Transfer Fail 0 Interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Transfer Fail bank 0 interrupt is disabled.
1	The Transfer Fail bank 0 interrupt is enabled and an interrupt request will be generated when the Transfer Fail 0 Interrupt Flag is set.

Bit 1 – TRCPT1: Transfer Complete 1 Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Transfer Complete 1 Interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Transfer Complete 1 interrupt is disabled.
1	The Transfer Complete 1 interrupt is enabled and an interrupt request will be generated when the Transfer Complete 1 Interrupt Flag is set.

Bit 0 – TRCPT0: Transfer Complete 0 interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Transfer Complete 0 interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Transfer Complete bank 0 interrupt is disabled.
1	The Transfer Complete bank 0 interrupt is enabled and an interrupt request will be generated when the Transfer Complete 0 Interrupt Flag is set.

39.8.3.7. Device Interrupt EndPoint Set n

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Endpoint Interrupt Enable Set (EPINTENCLR) register. This register is cleared by USB reset or when EPEN[n] is zero.

Name: EPINTENSETn
Offset: 0x109 + (n x 0x20)
Reset: 0x0000
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit 6 – STALL1: Transmit Stall 1 Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will enable the Transmit bank 1 Stall interrupt.

Value	Description
0	The Transmit Stall 1 interrupt is disabled.
1	The Transmit Stall 1 interrupt is enabled.

Bit 5 – STALL0: Transmit Stall 0 Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will enable the Transmit bank 0 Stall interrupt.

Value	Description
0	The Transmit Stall 0 interrupt is disabled.
1	The Transmit Stall 0 interrupt is enabled.

Bit 4 – RXSTP: Received Setup Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will enable the Received Setup interrupt.

Value	Description
0	The Received Setup interrupt is disabled.
1	The Received Setup interrupt is enabled.

Bit 3 – TRFAIL1: Transfer Fail bank 1 Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will enable the Transfer Fail interrupt.

Value	Description
0	The Transfer Fail interrupt is disabled.
1	The Transfer Fail interrupt is enabled.

Bit 2 – TRFAIL0: Transfer Fail bank 0 Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will enable the Transfer Fail interrupt.

Value	Description
0	The Transfer Fail interrupt is disabled.
1	The Transfer Fail interrupt is enabled.

Bit 1 – TRCPT1: Transfer Complete bank 1 interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will enable the Transfer Complete 0 interrupt.

Value	Description
0	The Transfer Complete bank 1 interrupt is disabled.
1	The Transfer Complete bank 1 interrupt is enabled.

Bit 0 – TRCPT0: Transfer Complete bank 0 interrupt Enable
Writing a zero to this bit has no effect.

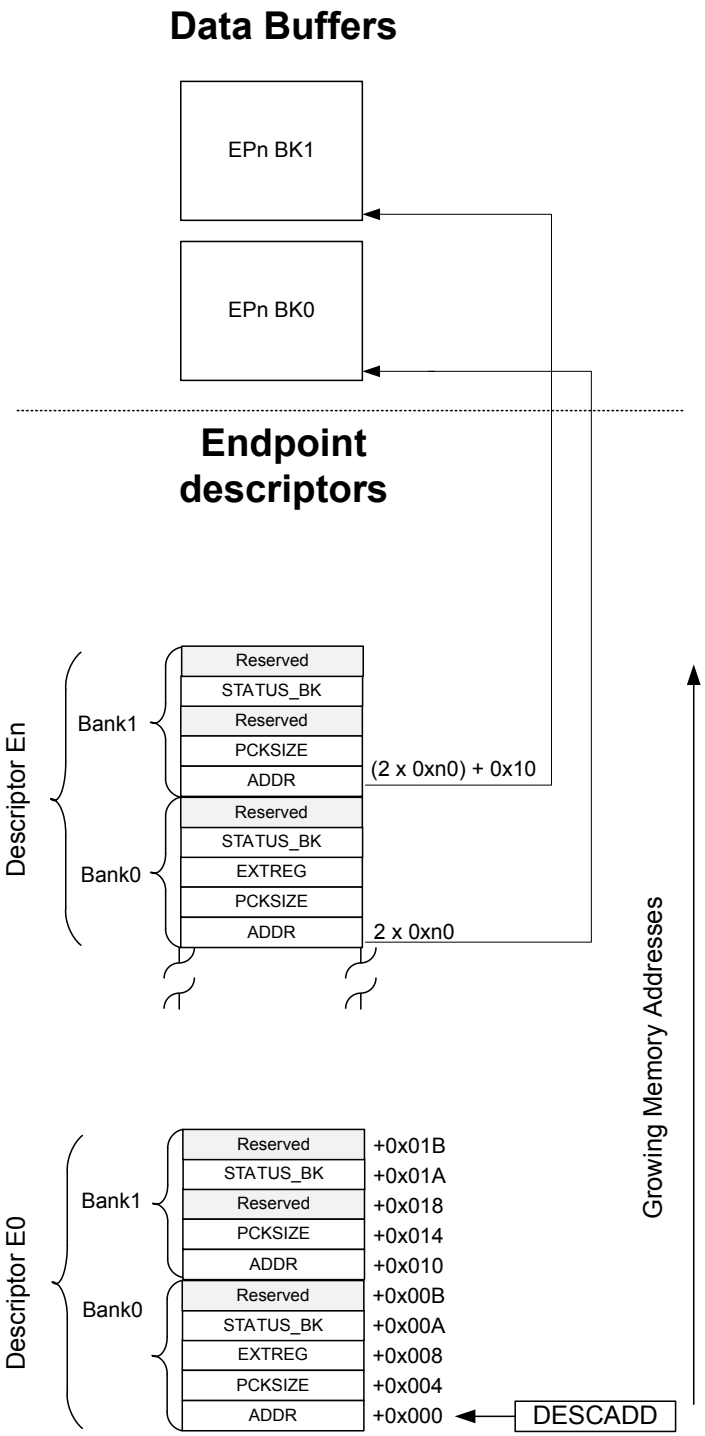
Writing a one to this bit will enable the Transfer Complete 1 interrupt.

0.2.4 Device Registers - Endpoint RAM

Value	Description
0	The Transfer Complete bank 0 interrupt is disabled.
1	The Transfer Complete bank 0 interrupt is enabled.

39.8.4. Device Registers - Endpoint RAM

39.8.4.1. Endpoint Descriptor Structure



39.8.4.2. Address of Data Buffer

Name: ADDR
Offset: 0x00 & 0x10
Reset: 0xxxxxxxx
Property: NA

Bit	31	30	29	28	27	26	25	24
	ADDR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	x	x	x	x	x	x	x

Bit	23	22	21	20	19	18	17	16
	ADDR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	x	x	x	x	x	x	x

Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	x	x	x	x	x	x	x

Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	x	x	x	x	x	x	x

Bits 31:0 – ADDR[31:0]: Data Pointer Address Value
These bits define the data pointer address as an absolute word address in RAM. The two least significant bits must be zero to ensure the start address is 32-bit aligned.

39.8.4.3. Packet Size

Name: PCKSIZE
Offset: 0x04 & 0x14
Reset: 0xxxxxxxxx
Property: NA

Bit	31	30	29	28	27	26	25	24
	AUTO_ZLP	SIZE[2:0]			MULTI_PACKET_SIZE[13:10]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	0	0	x	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	MULTI_PACKET_SIZE[9:2]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	MULTI_PACKET_SIZE[1:0]		BYTE_COUNT[13:8]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	x	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BYTE_COUNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	x

Bit 31 – AUTO_ZLP: Automatic Zero Length Packet
This bit defines the automatic Zero Length Packet mode of the endpoint.

When enabled, the USB module will manage the ZLP handshake by hardware. This bit is for IN endpoints only. When disabled the handshake should be managed by firmware.

Value	Description
0	Automatic Zero Length Packet is disabled.
1	Automatic Zero Length Packet is enabled.

Bits 30:28 – SIZE[2:0]: Endpoint size
These bits contains the maximum packet size of the endpoint.

Value	Description
0x0	8 Byte
0x1	16 Byte
0x2	32 Byte
0x3	64 Byte
0x4	128 Byte(1)
0x5	256 Byte(1)

Value	Description
0x6	512 Byte(1)
0x7	1023 Byte(1)

(1) for Isochronous endpoints only.

Bits 27:14 – MULTI_PACKET_SIZE[13:0]: Multiple Packet Size
These bits define the 14-bit value that is used for multi-packet transfers.

For IN endpoints, MULTI_PACKET_SIZE holds the total number of bytes sent. MULTI_PACKET_SIZE should be written to zero when setting up a new transfer.

For OUT endpoints, MULTI_PACKET_SIZE holds the total data size for the complete transfer. This value must be a multiple of the maximum packet size.

Bits 13:0 – BYTE_COUNT[13:0]: Byte Count
These bits define the 14-bit value that is used for the byte count.

For IN endpoints, BYTE_COUNT holds the number of bytes to be sent in the next IN transaction.

For OUT endpoint or SETUP endpoints, BYTE_COUNT holds the number of bytes received upon the last OUT or SETUP transaction.

39.8.4.4. Extended Register

Name: EXTREG
Offset: 0x08
Reset: 0xxxxxxxx
Property: NA

Bit	15	14	13	12	11	10	9	8
		VARIABLE[10:4]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	VARIABLE[3:0]				SUBPID[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	x	0	0	0	x

Bits 14:4 – VARIABLE[10:0]: VARIABLE
These bits define the VARIABLE field of a received extended token. These bits are updated when the USB has answered by an handshake token ACK to a LPM transaction. See Section 2.1.1 Protocol Extension Token in the reference document “ENGINEERING CHANGE NOTICE, USB 2.0 Link Power Management Addendum”.

To support the USB2.0 Link Power Management addition the VARIABLE field should be read as described below.

VARIABLES	Description
VARIABLE[3:0]	bLinkState (1)
VARIABLE[7:4]	BESL (2)
VARIABLE[8]	bRemoteWake (1)
VARIABLE[10:9]	Reserved

1. For a definition of LPM Token bRemoteWake and bLinkState fields, refer to "Table 2-3 in the reference document ENGINEERING CHANGE NOTICE, USB 2.0 Link Power Management Addendum".
2. For a definition of LPM Token BESL field, refer to "Table 2-3 in the reference document ENGINEERING CHANGE NOTICE, USB 2.0 Link Power Management Addendum" and "Table X-X1 in Errata for ECN USB 2.0 Link Power Management.

Bits 3:0 – SUBPID[3:0]: SUBPID
These bits define the SUBPID field of a received extended token. These bits are updated when the USB has answered by an handshake token ACK to a LPM transaction. See Section 2.1.1 Protocol Extension Token in the reference document “ENGINEERING CHANGE NOTICE, USB 2.0 Link Power Management Addendum”.

39.8.4.5. Device Status Bank

Name: STATUS_BK
Offset: 0x0A & 0x1A
Reset: 0xxxxxxxx
Property: NA

Bit	7	6	5	4	3	2	1	0
							ERROFLOW	CRCERR
Access							R/W	R/W
Reset							x	x

Bit 1 – ERROFLOW: Error Flow Status
This bit defines the Error Flow Status.

This bit is set when a Error Flow has been detected during transfer from/towards this bank.

For OUT transfer, a NAK handshake has been sent.

For Isochronous OUT transfer, an overrun condition has occurred.

For IN transfer, this bit is not valid. EPSTATUS.TRFAIL0 and EPSTATUS.TRFAIL1 should reflect the flow errors.

Value	Description
0	No Error Flow detected.
1	A Error Flow has been detected.

Bit 0 – CRCERR: CRC Error
This bit defines the CRC Error Status.

This bit is set when a CRC error has been detected in an isochronous OUT endpoint bank.

0.2.5 Host Registers - Common

Value	Description
0	No CRC Error.
1	CRC Error detected.

39.8.5. Host Registers - Common

39.8.5.1. Control B

Name: CTRLB
Offset: 0x08
Reset: 0x0000
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
					L1RESUME	VBUSOK	BUSRESET	SOFE
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
					SPDCONF[1:0]		RESUME	
Access					R/W	R/W	R/W	
Reset					0	0	0	

Bit 11 – L1RESUME: Send USB L1 Resume
Writing 0 to this bit has no effect.
1: Generates a USB L1 Resume on the USB bus. This bit should only be set when the Start-of-Frame generation is enabled (SOFE bit set). The duration of the USB L1 Resume is defined by the EXTREG.VARIABLE[7:4] bits field also known as BESL (See LPM ECN).See also EXTREG on page 984 Register.

This bit is cleared when the USB L1 Resume has been sent or when a USB reset is requested.

Bit 10 – VBUSOK: VBUS is OK
This notifies the USB HOST that USB operations can be started. When this bit is zero and even if the USB HOST is configured and enabled, HOST operation is halted. Setting this bit will allow HOST operation when the USB is configured and enabled.

Value	Description
0	The USB module is notified that the VBUS on the USB line is not powered.
1	The USB module is notified that the VBUS on the USB line is powered.

Bit 9 – BUSRESET: Send USB Reset

Value	Description
0	Reset generation is disabled. It is written to zero when the USB reset is completed or when a device disconnection is detected. Writing zero has no effect.
1	Generates a USB Reset on the USB bus.

Bit 8 – SOFE: Start-of-Frame Generation Enable

Value	Description
0	The SOF generation is disabled and the USB bus is in suspend state.
1	Generates SOF on the USB bus in full speed and keep it alive in low speed mode. This bit is automatically set at the end of a USB reset (INTFLAG.RST) or at the end of a downstream resume (INTFLAG.DNRSM) or at the end of L1 resume.

Bits 3:2 – SPDCONF[1:0]: Speed Configuration for Host
These bits select the host speed configuration as shown below

Value	Description
0x0	Low and Full Speed capable
0x1	Reserved
0x2	Reserved
0x3	Reserved

Bit 1 – RESUME: Send USB Resume
Writing 0 to this bit has no effect.
1: Generates a USB Resume on the USB bus.

This bit is cleared when the USB Resume has been sent or when a USB reset is requested.

39.8.5.2. Host Start-of-Frame Control

During a very short period just before transmitting a Start-of-Frame, this register is locked. Thus, after writing, it is recommended to check the register value, and write this register again if necessary. This register is cleared upon a USB reset.

Name: HSOFC
Offset: 0x0A
Reset: 0x0000
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	FLENCE				FLENC[3:0]			
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit 7 – FLENCE: Frame Length Control Enable

When this bit is '1', the time between Start-of-Frames can be tuned by up to +/-0.06% using FLENC[3:0].
Note: In Low Speed mode, FLENCE must be '0'.

Value	Description
0	Start-of-Frame is generated every 1ms.
1	Start-of-Frame generation depends on the signed value of FLENC[3:0]. USB Start-of-Frame period equals 1ms + (FLENC[3:0]/12000)ms

Bits 3:0 – FLENC[3:0]: Frame Length Control

These bits define the signed value of the 4-bit FLENC that is added to the Internal Frame Length when FLENCE is '1'. The internal Frame length is the top value of the frame counter when FLENCE is zero.

39.8.5.3. Status

Name: STATUS
Offset: 0x0C
Reset: 0x0000
Property: Read only

Bit	7	6	5	4	3	2	1	0
	LINESTATE[1:0]				SPEED[1:0]			
Access	R	R			R	R		
Reset	0	0			0	0		

Bits 7:6 – LINESTATE[1:0]: USB Line State Status
These bits define the current line state DP/DM.

LINESTATE[1:0]	USB Line Status
0x0	SE0/RESET
0x1	FS-J or LS-K State
0x2	FS-K or LS-J State

Bits 3:2 – SPEED[1:0]: Speed Status
These bits define the current speed used by the host.

SPEED[1:0]	Speed Status
0x0	Full-speed mode
0x1	Low-speed mode
0x2	Reserved
0x3	Reserved

39.8.5.4. Host Frame Number

Name: FNUM
Offset: 0x10
Reset: 0x0000
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
			FNUM[10:5]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FNUM[4:0]							
Access	R/W	R/W	R/W	R/W	R/W			
Reset	0	0	0	0	0			

Bits 13:3 – FNUM[10:0]: Frame Number
These bits contains the current SOF number.

These bits can be written by software to initialize a new frame number value. In this case, at the next SOF, the FNUM field takes its new value.

As the FNUM register lies across two consecutive byte addresses, writing byte-wise (8-bits) to the FNUM register may produce incorrect frame number generation. It is recommended to write FNUM register word-wise (32-bits) or half-word-wise (16-bits).

39.8.5.5. Host Frame Length

Name: FLENHIGH
Offset: 0x12
Reset: 0x0000
Property: Read-Only

Bit	7	6	5	4	3	2	1	0
	FLENHIGH[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – FLENHIGH[7:0]: Frame Length
These bits contains the 8 high-order bits of the internal frame counter.

Table 39-9. Counter Description vs. Speed

Host Register STATUS.SPEED	Description
Full Speed	With a USB clock running at 12MHz, counter length is 12000 to ensure a SOF generation every 1 ms.

39.8.5.6. Host Interrupt Enable Register Clear

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Name: INTENCLR
Offset: 0x14
Reset: 0x0000
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
							DDISC	DCONN
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	RAMACER	UPRSM	DNRSM	WAKEUP	RST	HSOF		
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bit 9 – DDISC: Device Disconnection Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Device Disconnection interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Device Disconnection interrupt is disabled.
1	The Device Disconnection interrupt is enabled and an interrupt request will be generated when the Device Disconnection interrupt Flag is set.

Bit 8 – DCONN: Device Connection Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Device Connection interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Device Connection interrupt is disabled.
1	The Device Connection interrupt is enabled and an interrupt request will be generated when the Device Connection interrupt Flag is set.

Bit 7 – RAMACER: RAM Access Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the RAM Access interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The RAM Access interrupt is disabled.
1	The RAM Access interrupt is enabled and an interrupt request will be generated when the RAM Access interrupt Flag is set.

Bit 6 – UPRSM: Upstream Resume from Device Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Upstream Resume interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Upstream Resume interrupt is disabled.
1	The Upstream Resume interrupt is enabled and an interrupt request will be generated when the Upstream Resume interrupt Flag is set.

Bit 5 – DNRSM: Down Resume Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Down Resume interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Down Resume interrupt is disabled.
1	The Down Resume interrupt is enabled and an interrupt request will be generated when the Down Resume interrupt Flag is set.

Bit 4 – WAKEUP: Wake Up Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Wake Up interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Wake Up interrupt is disabled.
1	The Wake Up interrupt is enabled and an interrupt request will be generated when the Wake Up interrupt Flag is set.

Bit 3 – RST: BUS Reset Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Bus Reset interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Bus Reset interrupt is disabled.
1	The Bus Reset interrupt is enabled and an interrupt request will be generated when the Bus Reset interrupt Flag is set.

Bit 2 – HSOF: Host Start-of-Frame Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Host Start-of-Frame interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Host Start-of-Frame interrupt is disabled.
1	The Host Start-of-Frame interrupt is enabled and an interrupt request will be generated when the Host Start-of-Frame interrupt Flag is set.

39.8.5.7. Host Interrupt Enable Register Set

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Name: INTENSET
Offset: 0x18
Reset: 0x0000
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
							DDISC	DCONN
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	RAMACER	UPRSM	DNRSM	WAKEUP	RST	HSOF		
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bit 9 – DDISC: Device Disconnection Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will set the Device Disconnection interrupt bit and enable the DDSIC interrupt.

Value	Description
0	The Device Disconnection interrupt is disabled.
1	The Device Disconnection interrupt is enabled.

Bit 8 – DCONN: Device Connection Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will set the Device Connection interrupt bit and enable the DCONN interrupt.

Value	Description
0	The Device Connection interrupt is disabled.
1	The Device Connection interrupt is enabled.

Bit 7 – RAMACER: RAM Access Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will set the RAM Access interrupt bit and enable the RAMACER interrupt.

Value	Description
0	The RAM Access interrupt is disabled.
1	The RAM Access interrupt is enabled.

Bit 6 – UPRSM: Upstream Resume from the device Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will set the Upstream Resume interrupt bit and enable the UPRSM interrupt.

Value	Description
0	The Upstream Resume interrupt is disabled.
1	The Upstream Resume interrupt is enabled.

Bit 5 – DNRSM: Down Resume Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will set the Down Resume interrupt Enable bit and enable the DNRSM interrupt.

Value	Description
0	The Down Resume interrupt is disabled.
1	The Down Resume interrupt is enabled.

Bit 4 – WAKEUP: Wake Up Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will set the Wake Up interrupt Enable bit and enable the WAKEUP interrupt request.

Value	Description
0	The WakeUp interrupt is disabled.
1	The WakeUp interrupt is enabled.

Bit 3 – RST: Bus Reset Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will set the Bus Reset interrupt Enable bit and enable the Bus RST interrupt.

Value	Description
0	The Bus Reset interrupt is disabled.
1	The Bus Reset interrupt is enabled.

Bit 2 – HSOF: Host Start-of-Frame Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will set the Host Start-of-Frame interrupt Enable bit and enable the HSOF interrupt.

Value	Description
0	The Host Start-of-Frame interrupt is disabled.
1	The Host Start-of-Frame interrupt is enabled.

39.8.5.8. Host Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x1C
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
							DDISC	DCONN
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	RAMACER	UPRSM	DNRSM	WAKEUP	RST	HSOF		
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bit 9 – DDISC: Device Disconnection Interrupt Flag
This flag is cleared by writing a one to the flag.

This flag is set when the device has been removed from the USB Bus and will generate an interrupt if INTENCLR/SET.DDISC is one.

Writing a zero to this bit has no effect.

Writing a one to this bit clears the DDISC Interrupt Flag.

Bit 8 – DCONN: Device Connection Interrupt Flag
This flag is cleared by writing a one to the flag.

This flag is set when a new device has been connected to the USB BUS and will generate an interrupt if INTENCLR/SET.DCONN is one.

Writing a zero to this bit has no effect.

Writing a one to this bit clears the DCONN Interrupt Flag.

Bit 7 – RAMACER: RAM Access Interrupt Flag
This flag is cleared by writing a one to the flag.

This flag is set when a RAM access error occurs during an OUT stage and will generate an interrupt if INTENCLR/SET.RAMACER is one.

Writing a zero to this bit has no effect.

Bit 6 – UPRSM: Upstream Resume from the Device Interrupt Flag
This flag is cleared by writing a one to the flag.

This flag is set when the USB has received an Upstream Resume signal from the Device and will generate an interrupt if INTENCLR/SET.UPRSM is one.

Writing a zero to this bit has no effect.

Bit 5 – DNRSM: Down Resume Interrupt Flag
This flag is cleared by writing a one to the flag.

This flag is set when the USB has sent a Down Resume and will generate an interrupt if INTENCLR/SET.DRSM is one.

Writing a zero to this bit has no effect.

Bit 4 – WAKEUP: Wake Up Interrupt Flag
This flag is cleared by writing a one.

This flag is set when:

l The host controller is in suspend mode (SOFE is zero) and an upstream resume from the device is detected.

l The host controller is in suspend mode (SOFE is zero) and an device disconnection is detected.

l The host controller is in operational state (VBUSOK is one) and an device connection is detected.

In all cases it will generate an interrupt if INTENCLR/SET.WAKEUP is one.

Writing a zero to this bit has no effect.

Bit 3 – RST: Bus Reset Interrupt Flag
This flag is cleared by writing a one to the flag.

This flag is set when a Bus “Reset” has been sent to the Device and will generate an interrupt if INTENCLR/SET.RST is one.

Writing a zero to this bit has no effect.

Bit 2 – HSOF: Host Start-of-Frame Interrupt Flag
This flag is cleared by writing a one to the flag.

This flag is set when a USB “Host Start-of-Frame” in Full Speed/High Speed or a keep-alive in Low Speed has been sent (every 1 ms) and will generate an interrupt if INTENCLR/SET.HSOF is one.

The value of the FNUM register is updated.

Writing a zero to this bit has no effect.

39.8.5.9. Pipe Interrupt Summary

Name: PINTSMRY
Offset: 0x20
Reset: 0x00000000
Property: Read-only

Bit	15	14	13	12	11	10	9	8
	PINT[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PINT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – PINT[15:0]
The flag PINT[n] is set when an interrupt is triggered by the pipe n. See PINTFLAG on page 974 register in the Host Pipe Register section.

This bit will be cleared when there are no interrupts pending for Pipe n.

Writing to this bit has no effect.

39.8.6. Host Registers - Pipe

39.8.6.1. Host Pipe n Configuration

Name: PCFGn
Offset: 0x100 + (n x 0x20)
Reset: 0x0000
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
			PTYPE[2:0]			BK	PTOKEN[1:0]	
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 5:3 – PTYPE[2:0]: Type of the Pipe
These bits contains the pipe type.

PTYPE[2:0]	Description
0x0	Pipe is disabled
0x1	Pipe is enabled and configured as CONTROL
0x2	Pipe is enabled and configured as ISO
0x3	Pipe is enabled and configured as BULK
0x4	Pipe is enabled and configured as INTERRUPT
0x5	Pipe is enabled and configured as EXTENDED
0x06-0x7	Reserved

These bits are cleared upon sending a USB reset.

Bit 2 – BK: Pipe Bank
This bit selects the number of banks for the pipe.

For control endpoints writing a zero to this bit is required as only Bank0 is used for Setup/In/Out transactions.

This bit is cleared when a USB reset is sent.

BK(1)	Description
0x0	Single-bank endpoint
0x1	Dual-bank endpoint

1. Bank field is ignored when PTYPE is configured as EXTENDED.

Value	Description
0	A single bank is used for the pipe.
1	A dual bank is used for the pipe.

Bits 1:0 – PTOKEN[1:0]: Pipe Token
These bits contains the pipe token.

PTOKEN[1:0](1)	Description
0x0	SETUP(2)
0x1	IN
0x2	OUT
0x3	Reserved

1. PTOKEN field is ignored when PTYPE is configured as EXTENDED.
2. Available only when PTYPE is configured as CONTROL

Theses bits are cleared upon sending a USB reset.

39.8.6.2. Interval for the Bulk-Out/Ping Transaction

Name: BINTERVAL
Offset: 0x103 + (n x 0x20)
Reset: 0x0000
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	BINTERVAL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – BINTERVAL[7:0]: BINTERVAL
These bits contains the Ping/Bulk-out period.

These bits are cleared when a USB reset is sent or when PEN[n] is zero.

BINTERVAL	Description
=0	Multiple consecutive OUT token is sent in the same frame until it is acked by the peripheral
>0	One OUT token is sent every BINTERVAL frame until it is acked by the peripheral

Depending from the type of pipe the desired period is defined as:

PTYPE	Description
Interrupt	1 ms to 255 ms
Isochronous	2^(Binterval) * 1 ms
Bulk or control	1 ms to 255 ms
EXT LPM	bInterval ignored. Always 1 ms when a NYET is received.

39.8.6.3. Pipe Status Clear n

Name: PSTATUSCLR
Offset: 0x104 + (n x 0x20)
Reset: 0x0000
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL
Access	R/W	R/W		R/W		R/W		R/W
Reset	0	0		0		0		0

Bit 7 – BK1RDY: Bank 1 Ready Clear
Writing a zero to this bit has no effect.

Writing a one to this bit will clear PSTATUS.BK1RDY bit.

Bit 6 – BK0RDY: Bank 0 Ready Clear
Writing a zero to this bit has no effect.

Writing a one to this bit will clear PSTATUS.BK0RDY bit.

Bit 4 – PFREEZE: Pipe Freeze Clear
Writing a zero to this bit has no effect.

Writing a one to this bit will clear PSTATUS.PFREEZE bit.

Bit 2 – CURBK: Current Bank Clear
Writing a zero to this bit has no effect.

Writing a one to this bit will clear PSTATUS.CURBK bit.

Bit 0 – DTGL: Data Toggle Clear
Writing a zero to this bit has no effect.

Writing a one to this bit will clear PSTATUS.DTGL bit.

39.8.6.4. Pipe Status Set Register n

Name: PSTATUSSET
Offset: 0x105 + (n x 0x20)
Reset: 0x0000
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL
Access	R/W	R/W		R/W		R/W		R/W
Reset	0	0		0		0		0

Bit 7 – BK1RDY: Bank 1 Ready Set
Writing a zero to this bit has no effect.

Writing a one to this bit will set the bit PSTATUS.BK1RDY.

Bit 6 – BK0RDY: Bank 0 Ready Set
Writing a zero to this bit has no effect.

Writing a one to this bit will set the bit PSTATUS.BK0RDY.

Bit 4 – PFREEZE: Pipe Freeze Set
Writing a zero to this bit has no effect.

Writing a one to this bit will set PSTATUS.PFREEZE bit.

Bit 2 – CURBK: Current Bank Set
Writing a zero to this bit has no effect.

Writing a one to this bit will set PSTATUS.CURBK bit.

Bit 0 – DTGL: Data Toggle Set
Writing a zero to this bit has no effect.

Writing a one to this bit will set PSTATUS.DTGL bit.

39.8.6.5. Pipe Status Register n

Name: PSTATUS
Offset: 0x106 + (n x 0x20)
Reset: 0x0000
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL
Access	R	R		R		R		R
Reset	0	0		0		0		0

Bit 7 – BK1RDY: Bank 1 is ready
Writing a one to the bit EPSTATUSCLR.BK1RDY will clear this bit.

Writing a one to the bit EPSTATUSSET.BK1RDY will set this bit.

This bank is not used for Control pipe.

Value	Description
0	The bank number 1 is not ready: For IN the bank is empty. For Control/OUT the bank is not yet fill in.
1	The bank number 1 is ready: For IN the bank is filled full. For Control/OUT the bank is filled in.

Bit 6 – BK0RDY: Bank 0 is ready
Writing a one to the bit EPSTATUSCLR.BK0RDY will clear this bit.

Writing a one to the bit EPSTATUSSET.BK0RDY will set this bit.

This bank is the only one used for Control pipe.

Value	Description
0	The bank number 0 is not ready: For IN the bank is not empty. For Control/OUT the bank is not yet fill in.
1	The bank number 0 is ready: For IN the bank is filled full. For Control/OUT the bank is filled in.

Bit 4 – PFREEZE: Pipe Freeze
Writing a one to the bit EPSTATUSCLR.PFREEZE will clear this bit.

Writing a one to the bit EPSTATUSSET.PFREEZE will set this bit.

This bit is also set by the hardware:

- When a STALL handshake has been received.
- After a PIPE has been enabled (rising of bit PEN.N).
- When an LPM transaction has completed whatever handshake is returned or the transaction was timed-out.
- When a pipe transfer was completed with a pipe error. See PINTFLAG on page 974 register.

When PFREEZE bit is set while a transaction is in progress on the USB bus, this transaction will be properly completed. PFREEZE bit will be read as “1” only when the ongoing transaction will have been completed.

Value	Description
0	The Pipe operates in normal operation.
1	The Pipe is frozen and no additional requests will be sent to the device on this pipe address.

Bit 2 – CURBK: Current Bank

Value	Description
0	The bank0 is the bank that will be used in the next single/multi USB packet.
1	The bank1 is the bank that will be used in the next single/multi USB packet.

Bit 0 – DTGL: Data Toggle Sequence
Writing a one to the bit EPSTATUSCLR.DTGL will clear this bit.

Writing a one to the bit EPSTATUSSET.DTGL will set this bit.

This bit is toggled automatically by hardware after a data transaction.

This bit will reflect the data toggle in regards of the token type (IN/OUT/SETUP).

Value	Description
0	The PID of the next expected transaction will be zero: data 0.
1	The PID of the next expected transaction will be one: data 1.

39.8.6.6. Host Pipe Interrupt Flag Register

Name: PINTFLAG
Offset: 0x107 + (n x 0x20)
Reset: 0x0000
Property: -

Bit	7	6	5	4	3	2	1	0
			STALL	TXSTP	PERR	TRFAIL	TRCPT1	TRCPT0
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 5 – STALL: STALL Received Interrupt Flag
This flag is cleared by writing a one to the flag.

This flag is set when a stall occurs and will generate an interrupt if PINTENCLR/SET.STALL is one.

Writing a zero to this bit has no effect.

Writing a one to this bit clears the STALL Interrupt Flag.

Bit 4 – TXSTP: Transmitted Setup Interrupt Flag
This flag is cleared by writing a one to the flag.

This flag is set when a Transfer Complete occurs and will generate an interrupt if PINTENCLR/SET.TXSTP is one.

Writing a zero to this bit has no effect.

Writing a one to this bit clears the TXSTP Interrupt Flag.

Bit 3 – PERR: Pipe Error Interrupt Flag
This flag is cleared by writing a one to the flag.

This flag is set when a pipe error occurs and will generate an interrupt if PINTENCLR/SET.PERR is one.

Writing a zero to this bit has no effect.

Writing a one to this bit clears the PERR Interrupt Flag.

Bit 2 – TRFAIL: Transfer Fail Interrupt Flag
This flag is cleared by writing a one to the flag.

This flag is set when a Transfer Fail occurs and will generate an interrupt if PINTENCLR/SET.TRFAIL is one.

Writing a zero to this bit has no effect.

Writing a one to this bit clears the TRFAIL Interrupt Flag.

Bit 1 – TRCPT1: Transfer Complete 1 interrupt Flag
This flag is cleared by writing a one to the flag.

This flag is set when a Transfer Complete occurs and will generate an interrupt if PINTENCLR/SET.TRCPT1 is one. PINTFLAG.TRCPT1 is set for a double bank IN/OUT pipe when current bank is 1.

Writing a zero to this bit has no effect.

Writing a one to this bit clears the TRCPT1 Interrupt Flag.

Bit 0 – TRCPT0: Transfer Complete 0 interrupt Flag
This flag is cleared by writing a one to the flag.

This flag is set when a Transfer complete occurs and will generate an interrupt if PINTENCLR/SET.TRCPT0 is one. PINTFLAG.TRCPT0 is set for a single bank IN/OUT pipe or a double bank IN/OUT pipe when current bank is 0.

Writing a zero to this bit has no effect.

Writing a one to this bit clears the TRCPT0 Interrupt Flag.

39.8.6.7. Host Pipe Interrupt Clear Register

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Pipe Interrupt Enable Set (PINTENSET) register.

This register is cleared by USB reset or when PEN[n] is zero.

Name: PINTENCLR
Offset: 0x108 + (n x 0x20)
Reset: 0x0000
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
			STALL	TXSTP	PERR	TRFAIL	TRCPT1	TRCPT0
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 5 – STALL: Received Stall Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Received Stall interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The received Stall interrupt is disabled.
1	The received Stall interrupt is enabled and an interrupt request will be generated when the received Stall interrupt Flag is set.

Bit 4 – TXSTP: Transmitted Setup Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Transmitted Setup interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Transmitted Setup interrupt is disabled.
1	The Transmitted Setup interrupt is enabled and an interrupt request will be generated when the Transmitted Setup interrupt Flag is set.

Bit 3 – PERR: Pipe Error Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Pipe Error interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Pipe Error interrupt is disabled.
1	The Pipe Error interrupt is enabled and an interrupt request will be generated when the Pipe Error interrupt Flag is set.

Bit 2 – TRFAIL: Transfer Fail Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Transfer Fail interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Transfer Fail interrupt is disabled.
1	The Transfer Fail interrupt is enabled and an interrupt request will be generated when the Transfer Fail interrupt Flag is set.

Bit 1 – TRCPT1: Transfer Complete Bank 1 interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Transfer Complete interrupt Enable bit 1 and disable the corresponding interrupt request.

Value	Description
0	The Transfer Complete Bank 1 interrupt is disabled.
1	The Transfer Complete Bank 1 interrupt is enabled and an interrupt request will be generated when the Transfer Complete interrupt Flag 1 is set.

Bit 0 – TRCPT0: Transfer Complete Bank 0 interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Transfer Complete interrupt Enable bit 0 and disable the corresponding interrupt request.

Value	Description
0	The Transfer Complete Bank 0 interrupt is disabled.
1	The Transfer Complete Bank 0 interrupt is enabled and an interrupt request will be generated when the Transfer Complete interrupt 0 Flag is set.

39.8.6.8. Host Interrupt Pipe Set Register

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Pipe Interrupt Enable Set (PINTENCLR) register.

This register is cleared by USB reset or when PEN[n] is zero.

Name: PINTENSET
Offset: 0x109 + (n x 0x20)
Reset: 0x0000
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
			STALL	TXSTP	PERR	TRFAIL	TRCPT1	TRCPT0
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 5 – STALL: Stall Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will enable the Stall interrupt.

Value	Description
0	The Stall interrupt is disabled.
1	The Stall interrupt is enabled.

Bit 4 – TXSTP: Transmitted Setup Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will enable the Transmitted Setup interrupt.

Value	Description
0	The Transmitted Setup interrupt is disabled.
1	The Transmitted Setup interrupt is enabled.

Bit 3 – PERR: Pipe Error Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will enable the Pipe Error interrupt.

Value	Description
0	The Pipe Error interrupt is disabled.
1	The Pipe Error interrupt is enabled.

Bit 2 – TRFAIL: Transfer Fail Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will enable the Transfer Fail interrupt.

Value	Description
0	The Transfer Fail interrupt is disabled.
1	The Transfer Fail interrupt is enabled.

Bit 1 – TRCPT1: Transfer Complete 1 interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will enable the Transfer Complete interrupt Enable bit 1.

Value	Description
0	The Transfer Complete 1 interrupt is disabled.
1	The Transfer Complete 1 interrupt is enabled.

Bit 0 – TRCPT0: Transfer Complete 0 interrupt Enable
Writing a zero to this bit has no effect.

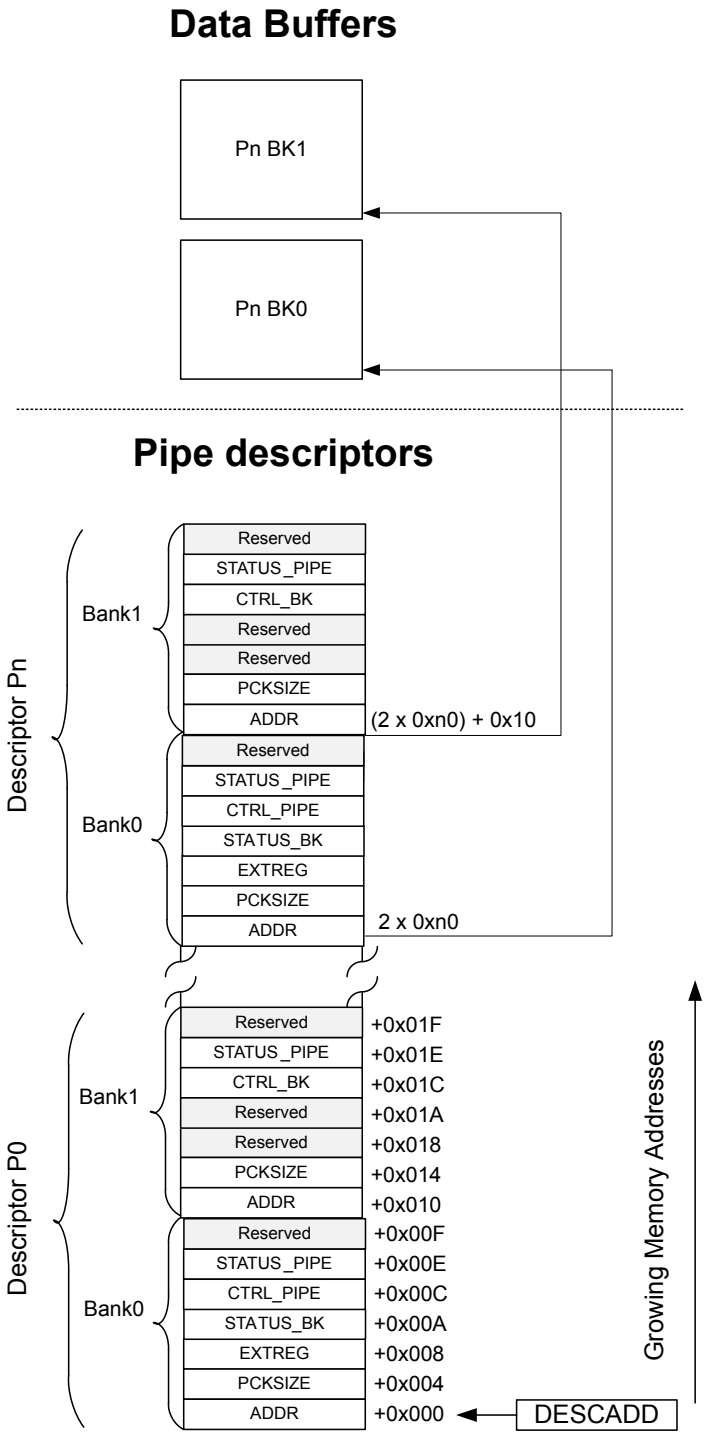
Writing a one to this bit will enable the Transfer Complete interrupt Enable bit 0.

0.2.7 Host Registers - Pipe RAM

Value	Description
0	The Transfer Complete 0 interrupt is disabled.
1	The Transfer Complete 0 interrupt is enabled.

39.8.7. Host Registers - Pipe RAM

39.8.7.1. Pipe Descriptor Structure



39.8.7.2. Address of the Data Buffer

Name: ADDR
Offset: 0x00 & 0x10
Reset: 0xxxxxxxx
Property: NA

Bit	31	30	29	28	27	26	25	24
	ADDR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ADDR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	x

Bits 31:0 – ADDR[31:0]: Data Pointer Address Value
These bits define the data pointer address as an absolute double word address in RAM. The two least significant bits must be zero to ensure the descriptor is 32-bit aligned.

39.8.7.3. Packet Size

Name: PCKSIZE
Offset: 0x04 & 0x14
Reset: 0xxxxxxxx
Property: NA

Bit	31	30	29	28	27	26	25	24
	AUTO_ZLP	SIZE[2:0]			MULTI_PACKET_SIZE[13:10]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	0	0	x	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	MULTI_PACKET_SIZE[9:2]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	MULTI_PACKET_SIZE[1:0]		BYTE_COUNT[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	x	0	0	0	0	0	x

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 31 – AUTO_ZLP: Automatic Zero Length Packet
This bit defines the automatic Zero Length Packet mode of the pipe.

When enabled, the USB module will manage the ZLP handshake by hardware. This bit is for OUT pipes only. When disabled the handshake should be managed by firmware.

Value	Description
0	Automatic Zero Length Packet is disabled.
1	Automatic Zero Length Packet is enabled.

Bits 30:28 – SIZE[2:0]: Pipe size
These bits contains the size of the pipe.

Theses bits are cleared upon sending a USB reset.

SIZE[2:0]	Description
0x0	8 Byte
0x1	16 Byte
0x2	32 Byte
0x3	64 Byte
0x4	128 Byte(1)

SIZE[2:0]	Description
0x5	256 Byte(1)
0x6	512 Byte(1)
0x7	1024 Byte in HS mode(1) 1023 Byte in FS mode(1)

1. For Isochronous pipe only.

Bits 27:14 – MULTI_PACKET_SIZE[13:0]: Multi Packet IN or OUT size
These bits define the 14-bit value that is used for multi-packet transfers.

For IN pipes, MULTI_PACKET_SIZE holds the total number of bytes sent. MULTI_PACKET_SIZE should be written to zero when setting up a new transfer.

For OUT pipes, MULTI_PACKET_SIZE holds the total data size for the complete transfer. This value must be a multiple of the maximum packet size.

Bits 13:8 – BYTE_COUNT[5:0]: Byte Count
These bits define the 14-bit value that contains number of bytes sent in the last OUT or SETUP transaction for an OUT pipe, or of the number of bytes to be received in the next IN transaction for an input pipe.

39.8.7.4. Extended Register

Name: EXTREG
Offset: 0x08
Reset: 0xxxxxxxx
Property: NA

Bit	15	14	13	12	11	10	9	8
		VARIABLE[10:4]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	VARIABLE[3:0]				SUBPID[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	x	0	0	0	x

Bits 14:4 – VARIABLE[10:0]: Extended variable
These bits define the VARIABLE field sent with extended token. See “Section 2.1.1 Protocol Extension Token in the reference document ENGINEERING CHANGE NOTICE, USB 2.0 Link Power Management Addendum.”

To support the USB2.0 Link Power Management addition the VARIABLE field should be set as described below.

VARIABLE	Description
VARIABLE[3:0]	bLinkState(1)
VARIABLE[7:4]	BESL (See LPM ECN)(2)
VARIABLE[8]	bRemoteWake(1)
VARIABLE[10:9]	Reserved

(1) for a definition of LPM Token bRemoteWake and bLinkState fields, refer to "Table 2-3 in the reference document ENGINEERING CHANGE NOTICE, USB 2.0 Link Power Management Addendum"
(2) for a definition of LPM Token BESL field, refer to "Table 2-3 in the reference document ENGINEERING CHANGE NOTICE, USB 2.0 Link Power Management Addendum" and "Table X-X1 in Errata for ECN USB 2.0 Link Power Management.

Bits 3:0 – SUBPID[3:0]: SUBPID
These bits define the SUBPID field sent with extended token. See “Section 2.1.1 Protocol Extension Token in the reference document ENGINEERING CHANGE NOTICE, USB 2.0 Link Power Management Addendum”.

To support the USB2.0 Link Power Management addition the SUBPID field should be set as described in “Table 2.2 SubPID Types in the reference document ENGINEERING CHANGE NOTICE, USB 2.0 Link Power Management Addendum”.

39.8.7.5. Host Status Bank

Name: STATUS_BK
Offset: 0x0A & 0x1A
Reset: 0xxxxxxxx
Property: NA

Bit	7	6	5	4	3	2	1	0
							ERROFLOW	CRCERR
Access							R/W	R/W
Reset							x	x

Bit 1 – ERROFLOW: Error Flow Status
This bit defines the Error Flow Status.

This bit is set when a Error Flow has been detected during transfer from/towards this bank.

For IN transfer, a NAK handshake has been received. For OUT transfer, a NAK handshake has been received. For Isochronous IN transfer, an overrun condition has occurred. For Isochronous OUT transfer, an underflow condition has occurred.

Value	Description
0	No Error Flow detected.
1	A Error Flow has been detected.

Bit 0 – CRCERR: CRC Error
This bit defines the CRC Error Status.

This bit is set when a CRC error has been detected in an isochronous IN endpoint bank.

Value	Description
0	No CRC Error.
1	CRC Error detected.

39.8.7.6. Host Control Pipe

Name: CTRL_PIPE
Offset: 0x0C
Reset: 0xxxxxxxx
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Bit	15	14	13	12	11	10	9	8
	PERMAX[3:0]				PEPNUM[3:0]			
Access	R	R	R	R	R/W	R/W	R/W	R/W
Reset	0	0	0	x	0	0	0	x

Bit	7	6	5	4	3	2	1	0
		PDADDR[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	x

Bits 15:12 – PERMAX[3:0]: Pipe Error Max Number
These bits define the maximum number of error for this Pipe before freezing the pipe automatically.

Bits 11:8 – PEPNUM[3:0]: Pipe EndPoint Number
These bits define the number of endpoint for this Pipe.

Bits 6:0 – PDADDR[6:0]: Pipe Device Address
These bits define the Device Address for this pipe.

39.8.7.7. Host Status Pipe

Name: STATUS_PIPE
Offset: 0x0E & 0x1E
Reset: 0xxxxxxxx
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	ERCNT[2:0]			CRC16ER	TOUTER	PIDER	DAPIDER	DTGLER
Access	R	R	R	R	R	R/W	R/W	R/W
Reset	0	0	x	x	x	x	x	x

Bits 7:5 – ERCNT[2:0]: Pipe Error Counter
These bits define the number of errors detected on the pipe.

Bit 4 – CRC16ER: CRC16 ERROR
This bit defines the CRC16 Error Status.

This bit is set when a CRC 16 error has been detected during a IN transactions.

Value	Description
0	No CRC 16 Error detected.
1	A CRC 16 error has been detected.

Bit 3 – TOUTER: TIME OUT ERROR
This bit defines the Time Out Error Status.

This bit is set when a Time Out error has been detected during a USB transaction.

Value	Description
0	No Time Out Error detected.
1	A Time Out error has been detected.

Bit 2 – PIDER: PID ERROR
This bit defines the PID Error Status.

This bit is set when a PID error has been detected during a USB transaction.

Value	Description
0	No PID Error detected.
1	A PID error has been detected.

Bit 1 – DAPIDER: Data PID ERROR
This bit defines the PID Error Status.

This bit is set when a Data PID error has been detected during a USB transaction.

Value	Description
0	No Data PID Error detected.
1	A Data PID error has been detected.

Bit 0 – DTGLER: Data Toggle Error
This bit defines the Data Toggle Error Status.

This bit is set when a Data Toggle Error has been detected.

Value	Description
0	No Data Toggle Error.
1	Data Toggle Error detected.

40. CCL – Configurable Custom Logic

40.1. Overview

The Configurable Custom Logic (CCL) is a programmable logic peripheral which can be connected to the device pins, to events, or to other internal peripherals. This allows the user to eliminate logic gates for simple glue logic functions on the PCB.

Each Lookup Table (LUT) consists of three inputs, a truth table, and as options synchronizer, filter and edge detector. Each LUT can generate an output as a user programmable logic expression with three inputs. Inputs can be individually masked.

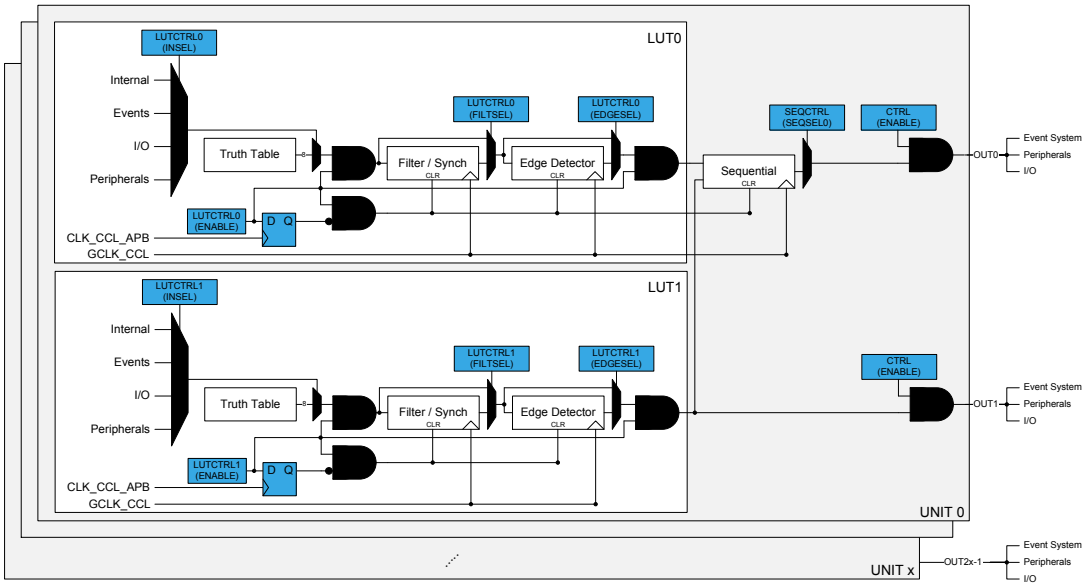
The output can be combinatorially generated from the inputs, and can be filtered to remove spikes. An optional sequential module can be enabled. The inputs of the sequential module are individually controlled by two independent, adjacent LUT (LUT0/LUT1, LUT2/LUT3 etc) outputs, enabling complex waveform generation.

40.2. Features

- Glue logic for general purpose PCB design
- Up to four Programmable LookUp Table (LUT)
- Combinatorial Logic Functions:
  AND, NAND, OR, NOR, XOR, XNOR, NOT
- Sequential Logic Functions:
  Gated D Flip-Flop, JK Flip-Flop, gated D Latch, RS Latch
- Flexible LookUp Table Inputs Selection:
  - I/Os
  - Events
  - Internal Peripherals
  - Subsequent LUT Output
- Output can be connected to IO pins or Event System
- Optional synchronizer, filter, or edge detector available on each LUT output

40.3. Block Diagram

Figure 40-1. Configurable Custom Logic



40.4. Signal Description

Pin Name	Type	Description
OUT[n]-OUT0	Digital output	Output from lookup table
IN[3n+2] - IN0	Digital input	Input to lookup table

Refer to *I/O Multiplexing and Considerations* for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

Related Links

I/O Multiplexing and Considerations on page 29

40.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

40.5.1. I/O Lines

Using the CCL I/O lines requires the I/O pins to be configured. Refer to *PORT - I/O Pin Controller* for details.

Related Links

PORT: IO Pin Controller on page 506

40.5.2. Power Management

This peripheral can continue to operate in any sleep mode where its source clock is running. Events connected to the event system can trigger other operations in the system without exiting sleep modes. Refer to *PM – Power Manager* for details on the different sleep modes.

Related Links

PM – Power Manager on page 186

40.5.3. Clocks

The CCL bus clock (CLK_CCL_APB) can be enabled and disabled in the power manager, and the default state of CLK_CCL_APB can be found in the *Peripheral Clock Masking*.

A generic clock (GCLK_CCL) is optionally required to clock the CCL. This clock must be configured and enabled in the Generic Clock Controller (GCLK) before using the sequential sub-module of CCL. GCLK_CCL is required when input events, a filter, an edge detector, or asequential sub-module is enabled. Refer to *GCLK - Generic Clock Controller* for details.

This generic clock is asynchronous to the user interface clock (CLK_CCL_APB).

Related Links

Peripheral Clock Masking on page 151

GCLK - Generic Clock Controller on page 131

40.5.4. DMA

Not applicable.

40.5.5. Interrupts

Not applicable.

40.5.6. Events

The events are connected to the Event System. Refer to *EVSYS – Event System* for details on how to configure the Event System.

Related Links

EVSYS – Event System on page 536

40.5.7. Debug Operation

When the CPU is halted in debug mode the CCL continues normal operation. If the CCL is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

40.5.8. Register Access Protection

All registers with write-access can be write-protected optionally by the peripheral access controller (PAC). Refer to *PAC - Peripheral Access Controller* for details.

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

PAC write-protection does not apply to accesses through an external debugger.

Related Links

PAC - Peripheral Access Controller on page 58

40.5.9. Analog Connections

Not applicable.

40.6. Functional Description

40.6.1. Principle of Operation

Configurable Custom Logic (CCL) is a programmable logic block that can use the device port pins and the internal Event System as both input and output channels. The CCL can serve as glue logic between the device and external devices. This increases the reliability of the PCB by reducing its complexity, and enables more powerful functions.

40.6.2. Basic Operation

40.6.2.1. Initialization

The following bits are enable-protected, meaning that they can only be written when the corresponding even LUT is disabled (LUTCTRL2x.ENABLE=0):

- Sequential Selection in Sequential Control x register (SEQCTRLx.SEQSEL)

The following registers are enable-protected, meaning that they can only be written when the corresponding LUT is disabled (LUTCTRLx.ENABLE=0):

- LUT Control x register, except ENABLE bit (LUTCTRLx)

Enable-protected bits in the LUTCTRLx registers can be written at the same time as LUTCTRLx.ENABLE is written to '1', but not at the same time as LUTCTRLx.ENABLE is written to '0'.

Enable-protection is denoted by the Enable-Protected property in the register description.

40.6.2.2. Enabling, Disabling, and Resetting

The CCL is enabled by writing a '1' to the Enable bit in the Control register (CTRL.ENABLE). The CCL is disabled by writing a '0' to CTRL.ENABLE.

Each LUT is enabled by writing a '1' to the Enable bit in the LUT Control x register (LUTCTRLx.ENABLE). Each LUT is disabled by writing a '0' to LUTCTRLx.ENABLE.

The CCL is reset by writing a '1' to the Software Reset bit in the Control register (CTRL.SWRST). All registers in the CCL will be reset to their initial state, and the CCL will be disabled. Refer to CTRL on page 1003 for details.

40.6.2.3. Lookup Table Logic

The lookup table in each LUT unit can generate any logic expression OUT as a function of three inputs (IN[2:0]), as shown in Figure 40-2 Truth Table Output Value Selection on page 992. One or more inputs can be masked. The truth table for the expression is defined by TRUTH bits in LUT Control x register (LUTCTRLx.TRUTH).

Figure 40-2. Truth Table Output Value Selection

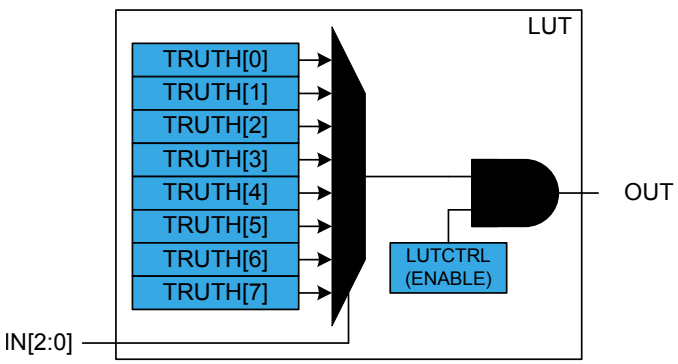


Table 40-1. Truth Table of LUT

IN[2]	IN[1]	IN[0]	OUT
0	0	0	TRUTH[0]
0	0	1	TRUTH[1]
0	1	0	TRUTH[2]
0	1	1	TRUTH[3]
1	0	0	TRUTH[4]
1	0	1	TRUTH[5]
1	1	0	TRUTH[6]
1	1	1	TRUTH[7]

40.6.2.4. Truth Table Inputs Selection

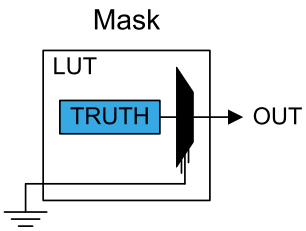
Input Overview
The inputs can be individually:

- Masked
- Driven by peripherals:
  - Analog comparator output (AC)
  - Timer/Counters waveform outputs (TC)
  - Serial Communication output transmit interface (SERCOM)
- Driven by internal events from Event System
- Driven by other CCL sub-modules

The Input Selection for each input y of LUT x is configured by writing the Input y Source Selection bit in the LUT x Control register (LUTCTRLx.INSELy).

Masked Inputs (MASK)
When a LUT input is masked (LUTCTRLx.INSELy=MASK), the corresponding TRUTH input (IN) is internally tied to zero, as shown in this figure:

Figure 40-3. Masked Input Selection



Internal Feedback Inputs (FEEDBACK)
When selected (LUTCTRLx.INSELy=FEEDBACK), the Sequential (SEQ) output is used as input for the corresponding LUT.

The output from an internal sequential sub-module can be used as input source for the LUT, see figure below for an example for LUT0 and LUT1. The sequential selection for each LUT follows the formula:

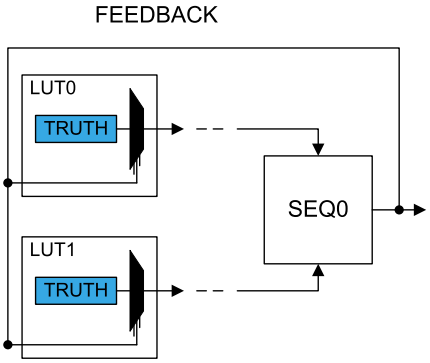
$$\mathrm{IN}[2\mathrm{N}][i] = \mathrm{SEQ}[N]$$

$\mathrm{IN}[2\mathrm{N}+1][i] = \mathrm{SEQ}[N]$

With *N* representing the sequencer number and *i*=0,1,2 representing the LUT input index.

For details, refer to Sequential Logic on page 998.

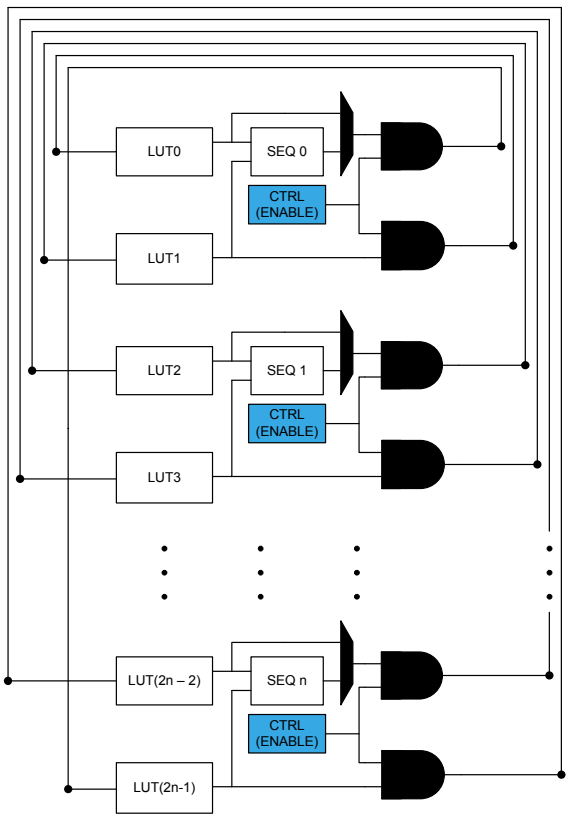
Figure 40-4. Feedback Input Selection



Linked LUT (LINK)

When selected (LUTCTRLx.INSELy=LINK), the subsequent LUT output is used as the LUT input (e.g., LUT2 is the input for LUT1), as shown in this figure:

Figure 40-5. Linked LUT Input Selection



Internal Events Inputs Selection(EVENT)

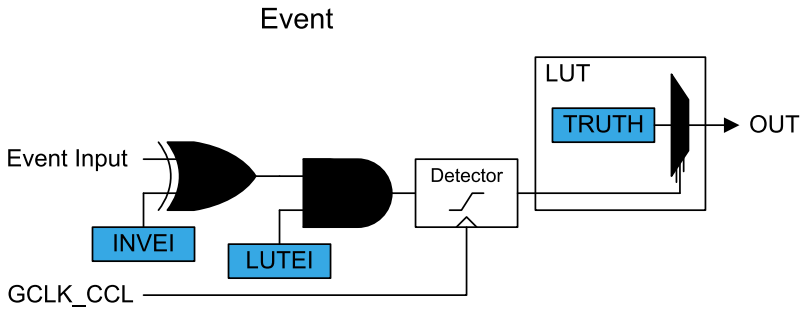
Asynchronous events from the Event System can be used as input selection, as shown in Figure 40-6 Event Input Selection on page 995. For each LUT, one event input line is available and can be selected

on each LUT input. Before enabling the event selection by writing LUTCTRLx.INSELy=EVENT, the Event System must be configured first.

The CCL includes an edge detector. When the event is received, an internal strobe is generated when a rising edge is detected. The pulse duration is one GCLK_CCL clock cycle. The following steps ensure proper operation:

1. Enable the GCLK_CCL clock
2. Configure the Event System to route the event asynchronously
3. If a strobe must be generated on the event input falling edge, write a '1' to the Inverted Event Input Enable bit in LUT Control register (LUTCTRLx.INVEI)
4. Enable the event input by writing the Event Input Enable bit in LUT Control register (LUTCTRLx.LUTEI) to '1'.

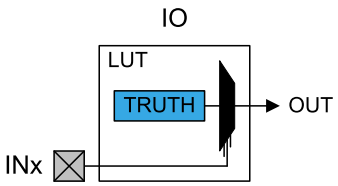
Figure 40-6. Event Input Selection



I/O Pin Inputs (IO)

When the IO pin is selected as LUT input (LUTCTRLx.INSELy=IO), the corresponding LUT input will be connected to the pin, as shown in the figure below.

Figure 40-7. I/O Pin Input Selection



Analog Comparator Inputs (AC)

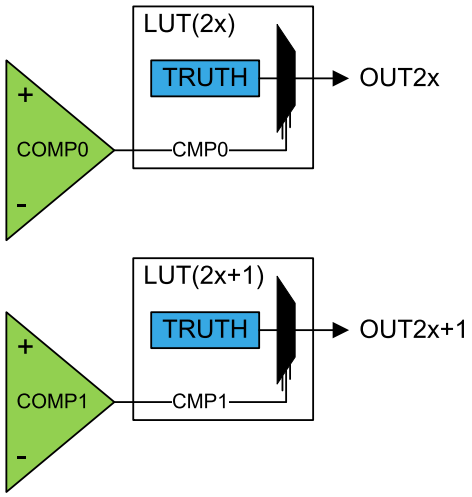
The AC outputs can be used as input source for the LUT (LUTCTRLx.INSELy=AC). The output of comparator 0 is available on even LUTs ("LUT(2x)": LUT0, LUT2) and the comparator 1 output is available on odd LUTs ("LUT(2x+1)": LUT1, LUT3), as shown in the figure below. More general, the analog comparator outputs are distributed following the formula:

$$\mathrm{IN}[N][i] = \mathrm{AC}[N \,\%\, \mathrm{ComparatorOutput\_Number}]$$

With *N* representing the LUT number and *i*=[0,1,2] representing the LUT input index.

Before selecting the comparator output, the AC must be configured first.

Figure 40-8. AC Input Selection



Timer/Counter Inputs (TC)

The TC waveform output WO[0] can be used as input source for the LUT (LUTCTRLx.INSELy=TC). Only consecutive instances of the TC, i.e. TCx and the subsequent TC(x+1), are available as default and alternative TC selections (e.g., TC0 and TC1 are sources for LUT0, TC1 and TC2 are sources for LUT1, etc). See the figure below for an example for LUT0. More general, the Timer/Counter selection for each LUT follows the formula:

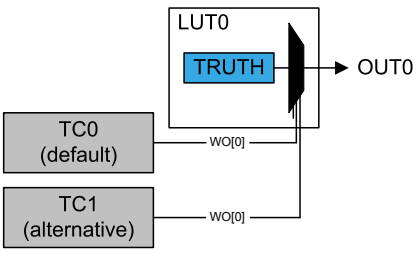
$$\mathrm{IN}[N][i] = DefaultTC[N \,\%\, \mathrm{TC\_Instance\_Number}]$$

$$\mathrm{IN}[N][i] = AlternativeTC[(N+1) \,\%\, \mathrm{TC\_Instance\_Number}]$$

Where N represents the LUT number and i represents the LUT input index (i=0,1,2).

Before selecting the waveform outputs, the TC must be configured first.

Figure 40-9. TC Input Selection



Timer/Counter for Control Application Inputs (TCC)

The TCC waveform outputs can be used as input source for the LUT. Only WO[2:0] outputs can be selected and routed to the respective LUT input (i.e., IN0 is connected to WO0, IN1 to WO1, and IN2 to WO2), as shown in the figure below.

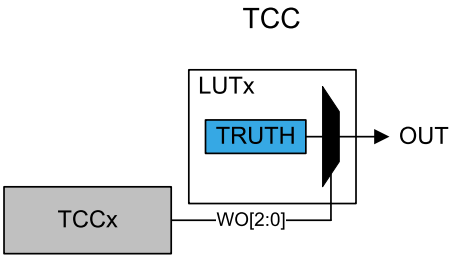
Note:
The TCC selection for each LUT follows the formula:

$$\mathrm{IN}[N][i] = TCC[N \,\%\, TCC\_\mathrm{Instance\_Number}]$$

Where *N* represents the LUT number.

Before selecting the waveform outputs, the TCC must be configured first.

Figure 40-10. TCC Input Selection



Serial Communication Output Transmit Inputs (SERCOM)

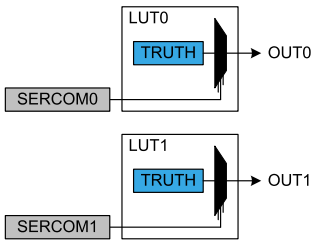
The serial engine transmitter output from Serial Communication Interface (SERCOM TX, TXd for USART, MOSI for SPI) can be used as input source for the LUT. The figure below shows an example for LUT0 and LUT1. The SERCOM selection for each LUT follows the formula:

$$\mathrm{IN}[N][i] = SERCOM[N \,\%\, \mathrm{SERCOM\_Instance\_Number}]$$

With *N* representing the LUT number and *i*=0,1,2 representing the LUT input index.

Before selecting the SERCOM as input source, the SERCOM must be configured first: the SERCOM TX signal must be output on SERCOMn/pad[0], which serves as input pad to the CCL.

Figure 40-11. SERCOM Input Selection



Related Links

I/O Multiplexing and Considerations on page 29
PORT: IO Pin Controller on page 506
GCLK - Generic Clock Controller on page 131
AC – Analog Comparators on page 1072
TC – Timer/Counter on page 704
TCC – Timer/Counter for Control Applications on page 755
SERCOM – Serial Communication Interface on page 560
Multiplexed Signals on page 29

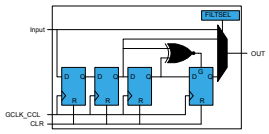
40.6.2.5. Filter

By default, the LUT output is a combinatorial function of the LUT inputs. This may cause some short glitches when the inputs change value. These glitches can be removed by clocking through filters, if demanded by application needs.

The Filter Selection bits in LUT Control register (LUTCTRLx.FILTSEL) define the synchronizer or digital filter options. When a filter is enabled, the OUT output will be delayed by two to five GCLK cycles. One APB clock after the corresponding LUT is disabled, all internal filter logic is cleared.

Note: Events used as LUT input will also be filtered, if the filter is enabled.

Figure 40-12. Filter



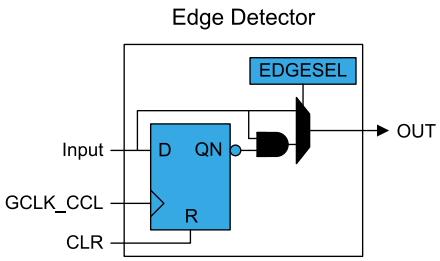
40.6.2.6. Edge Detector

The edge detector can be used to generate a pulse when detecting a rising edge on its input. To detect a falling edge, the TRUTH table should be programmed to provide the opposite levels.

The edge detector is enabled by writing '1' to the Edge Selection bit in LUT Control register (LUTCTRLx.EDGESEL). In order to avoid unpredictable behavior, a valid filter option must be enabled as well.

Edge detection is disabled by writing a '0' to LUTCTRLx.EDGESEL. After disabling a LUT, the corresponding internal Edge Detector logic is cleared one APB clock cycle later.

Figure 40-13. Edge Detector



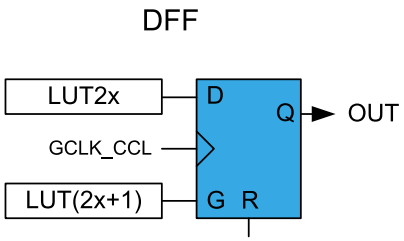
40.6.2.7. Sequential Logic

Each LUT pair can be connected to internal sequential logic: D flip flop, JK flip flop, gated D-latch or RS-latch can be selected by writing the corresponding Sequential Selection bits in Sequential Control x register (SEQCTRLx.SEQSEL). Before using sequential logic, the GCLK clock and optionally each LUT filter or edge detector, must be enabled.

Gated D Flip-Flop (DFF)

When the DFF is selected, the D-input is driven by the even LUT output (LUT2x), and the G-input is driven by the odd LUT output (LUT2x+1), as shown in Figure 40-14 D Flip Flop on page 998.

Figure 40-14. D Flip Flop



When the even LUT is disabled (LUTCTRL2x.ENABLE=0), the flip-flop is asynchronously cleared. The reset command (R) is kept enabled for one APB clock cycle. In all other cases, the flip-flop output (OUT) is refreshed on rising edge of the GCLK_CCL, as shown in Table 40-2 DFF Characteristics on page 999.

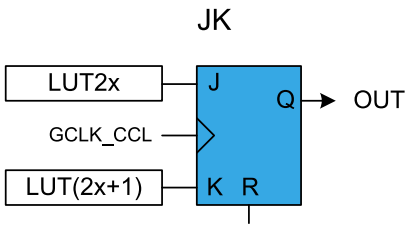
Table 40-2. DFF Characteristics

R	G	D	OUT
1	X	X	Clear
0	1	1	Set
		0	Clear
	0	X	Hold state (no change)

JK Flip-Flop (JK)

When this configuration is selected, the J-input is driven by the even LUT output (LUT2x), and the K-input is driven by the odd LUT output (LUT2x+1), as shown in Figure 40-15 JK Flip Flop on page 999.

Figure 40-15. JK Flip Flop



When the even LUT is disabled (LUTCTRL2x.ENABLE=0), the flip-flop is asynchronously cleared. The reset command (R) is kept enabled for one APB clock cycle. In all other cases, the flip-flop output (OUT) is refreshed on rising edge of the GCLK_CCL, as shown in Table 40-3 JK Characteristics on page 999.

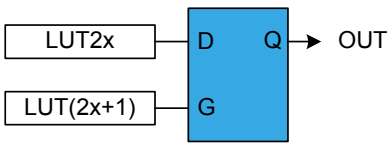
Table 40-3. JK Characteristics

R	J	K	OUT
1	X	X	Clear
0	0	0	Hold state (no change)
0	0	1	Clear
0	1	0	Set
0	1	1	Toggle

Gated D-Latch (DLATCH)

When the DLATCH is selected, the D-input is driven by the even LUT output (LUT2x), and the G-input is driven by the odd LUT output (LUT2x+1), as shown in Figure 40-14 D Flip Flop on page 998.

Figure 40-16. D-Latch



When the even LUT is disabled (LUTCTRL2x.ENABLE=0), the latch output will be cleared. The G-input is forced enabled for one more APB clock cycle, and the D-input to zero. In all other cases, the latch output (OUT) is refreshed as shown in Table 40-4 D-Latch Characteristics on page 1000.

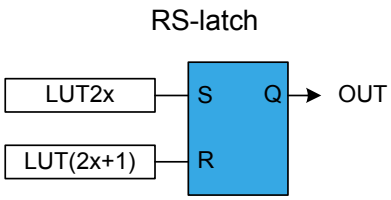
Table 40-4. D-Latch Characteristics

G	D	OUT
0	X	Hold state (no change)
1	0	Clear
1	1	Set

RS Latch (RS)

When this configuration is selected, the S-input is driven by the even LUT output (LUT2x), and the R-input is driven by the odd LUT output (LUT2x+1), as shown in Figure 40-17 RS-Latch on page 1000.

Figure 40-17. RS-Latch



When the even LUT is disabled (LUTCTRL2x.ENABLE=0), the latch output will be cleared. The R-input is forced enabled for one more APB clock cycle and S-input to zero. In all other cases, the latch output (OUT) is refreshed as shown in Table 40-5 RS-latch Characteristics on page 1000.

Table 40-5. RS-latch Characteristics

S	R	OUT
0	0	Hold state (no change)
0	1	Clear
1	0	Set
1	1	Forbidden state

40.6.3. Events

The CCL can generate the following output events:

- LUTOUTx: Lookup Table Output Value

Writing a '1' to the LUT Control Event Output Enable bit (LUTCTRL.LUTEO) enables the corresponding output event. Writing a '0' to this bit disables the corresponding output event. Refer to *EVSYS – Event System* for details on configuration.

The CCL can take the following actions on an input event:

- INx: The event is used as input for the TRUTH table. For further details refer to Events on page 991.

Writing a '1' to the LUT Control Event Input Enable bit (LUTCTRL.LUTEI) enables the corresponding action on input event. Writing a '0' to this bit disables the corresponding action on input event. Refer to *EVSYS – Event System* for details on configuration.

Related Links

EVSYS – Event System on page 536

40.6.4. Sleep Mode Operation

When using the GCLK_CCL internal clocking, writing the Run In Standby bit in the Control register (CTRL.RUNSTDBY) to '1' will allow GCLK_CCL to be enabled in all sleep modes.

If CTRL.RUNSTDBY=0, the GCLK_CCL will be disabled. If the Filter, Edge Detector or Sequential logic are enabled, the LUT output will be forced to zero in STANDBY mode. In all other cases, the TRUTH table decoder will continue operation and the LUT output will be refreshed accordingly.

40.7. Register Summary

Offset	Name	Bit Pos.								
0x00	CTRL	7:0		RUNSTDBY					ENABLE	SWRST
0x01 ... 0x03	Reserved									
0x04	SEQCTRL0	7:0					SEQSEL[3:0]			
0x05	SEQCTRL1	7:0					SEQSEL[3:0]			
0x06 ... 0x07	Reserved									
0x08	LUTCTRL0	7:0	EDGESEL		FILTSEL[1:0]				ENABLE	
0x09		15:8	INSEL1[3:0]				INSEL0[3:0]			
0x0A		23:16		LUTEO	LUTEI	INVEI	INSEL2[3:0]			
0x0B		31:24	TRUTH[7:0]							
0x0C	LUTCTRL1	7:0	EDGESEL		FILTSEL[1:0]				ENABLE	
0x0D		15:8	INSEL1[3:0]				INSEL0[3:0]			
0x0E		23:16		LUTEO	LUTEI	INVEI	INSEL2[3:0]			
0x0F		31:24	TRUTH[7:0]							
0x10	LUTCTRL2	7:0	EDGESEL		FILTSEL[1:0]				ENABLE	
0x11		15:8	INSEL1[3:0]				INSEL0[3:0]			
0x12		23:16		LUTEO	LUTEI	INVEI	INSEL2[3:0]			
0x13		31:24	TRUTH[7:0]							
0x14	LUTCTRL3	7:0	EDGESEL		FILTSEL[1:0]				ENABLE	
0x15		15:8	INSEL1[3:0]				INSEL0[3:0]			
0x16		23:16		LUTEO	LUTEI	INVEI	INSEL2[3:0]			
0x17		31:24	TRUTH[7:0]							

40.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection on page 991.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

40.8.1. Control

Name: CTRL
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
		RUNSTDBY					ENABLE	SWRST
Access		R/W					R/W	R/W
Reset		0					0	0

Bit 6 – RUNSTDBY: Run in Standby
This bit indicates if the GCLK_CCL clock must be kept running in standby mode. The setting is ignored for configurations where the generic clock is not required. For details refer to Sleep Mode Operation on page 1001.

Value	Description
0	Generic clock is not required in standby sleep mode.
1	Generic clock is required in standby sleep mode.

Bit 1 – ENABLE: Enable

Value	Description
0	The peripheral is disabled.
1	The peripheral is enabled.

Bit 0 – SWRST: Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the CCL to their initial state.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

40.8.2. Sequential Control x

Name: SEQCTRLx
Offset: 0x04 + n*0x01 [n=0..1]
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
					SEQSEL[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 3:0 – SEQSEL[3:0]: Sequential Selection
These bits select the sequential configuration:

Sequential Selection

Value	Name	Description
0x0	DISABLE	Sequential logic is disabled
0x1	DFF	D flip flop
0x2	JK	JK flip flop
0x3	LATCH	D latch
0x4	RS	RS latch
0x5 - 0xF		Reserved

40.8.3. LUT Control x

Name: LUTCTRLx
Offset: 0x08 + n*0x04 [n=0..3]
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected (except LUTEN)

Bit	31	30	29	28	27	26	25	24
	TRUTH[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
		LUTEO	LUTEI	INVEI	INSEL2[3:0]			
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	INSEL1[3:0]				INSEL0[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	EDGESEL		FILTSEL[1:0]				ENABLE	
Access	R/W		R/W	R/W			R/W	
Reset	0		0	0			0	

Bits 31:24 – TRUTH[7:0]: Truth Table
These bits define the value of truth logic as a function of inputs IN[2:0].

Bit 22 – LUTEO: LUT Event Output Enable

Value	Description
0	LUT event output is disabled.
1	LUT event output is enabled.

Bit 21 – LUTEI: LUT Event Input Enable

Value	Description
0	LUT incoming event is disabled.
1	LUT incoming event is enabled.

Bit 20 – INVEI: Inverted Event Input Enable

Value	Description
0	Incoming event is not inverted.
1	Incoming event is inverted.

Bit 7 – EDGESEL: Edge Selection

Value	Description
0	Edge detector is disabled.
1	Edge detector is enabled.

Bits 5:4 – FILTSEL[1:0]: Filter Selection
These bits select the LUT output filter options:

Filter Selection

Value	Name	Description
0x0	DISABLE	Filter disabled
0x1	SYNCH	Synchronizer enabled
0x2	FILTER	Filter enabled
0x3	Reserved	

Bit 1 – ENABLE: LUT Enable

Value	Description
0	The LUT is disabled.
1	The LUT is enabled.

Bits 19:16,15:12,11:8 – INSELx: LUT Input x Source Selection
These bits select the LUT input x source:

Value	Name	Description
0x0	MASK	Masked input
0x1	FEEDBACK	Feedback input source
0x2	LINK	Linked LUT input source
0x3	EVENT	Event input source
0x4	IO	I/O pin input source
0x5	AC	AC input source
0x6	TC	TC input source
0x7	ALTTC	Alternative TC input source
0x8	TCC	TCC input source
0x9	SERCOM	SERCOM input source
0xA - 0xF	Reserved	

41. OPAMP – Operational Amplifier Controller

41.1. Overview

The Operational Amplifier (OPAMP) Controller configures and controls three low power, general purpose operational amplifiers offering a high degree of flexibility and rail-to-rail inputs.

Most common inverting or non-inverting programmable gain and hysteresis configurations can be selected by software - no external components are required for these configurations.

The OPAMPs can be cascaded for both standalone mode and built-in configurations.

Each OPAMP can be used as a standalone amplifier. External pins are available for filter configurations or other applications. A reference can be generated from the DAC to be used as selectable reference for inverting PGA (programmable gain amplifier) or instrumentation amplifier. Each OPAMP can be used as buffer or PGA for the ADC or an AC. The OPAMP offset voltage can be compensated when it is used in combination with the ADC.

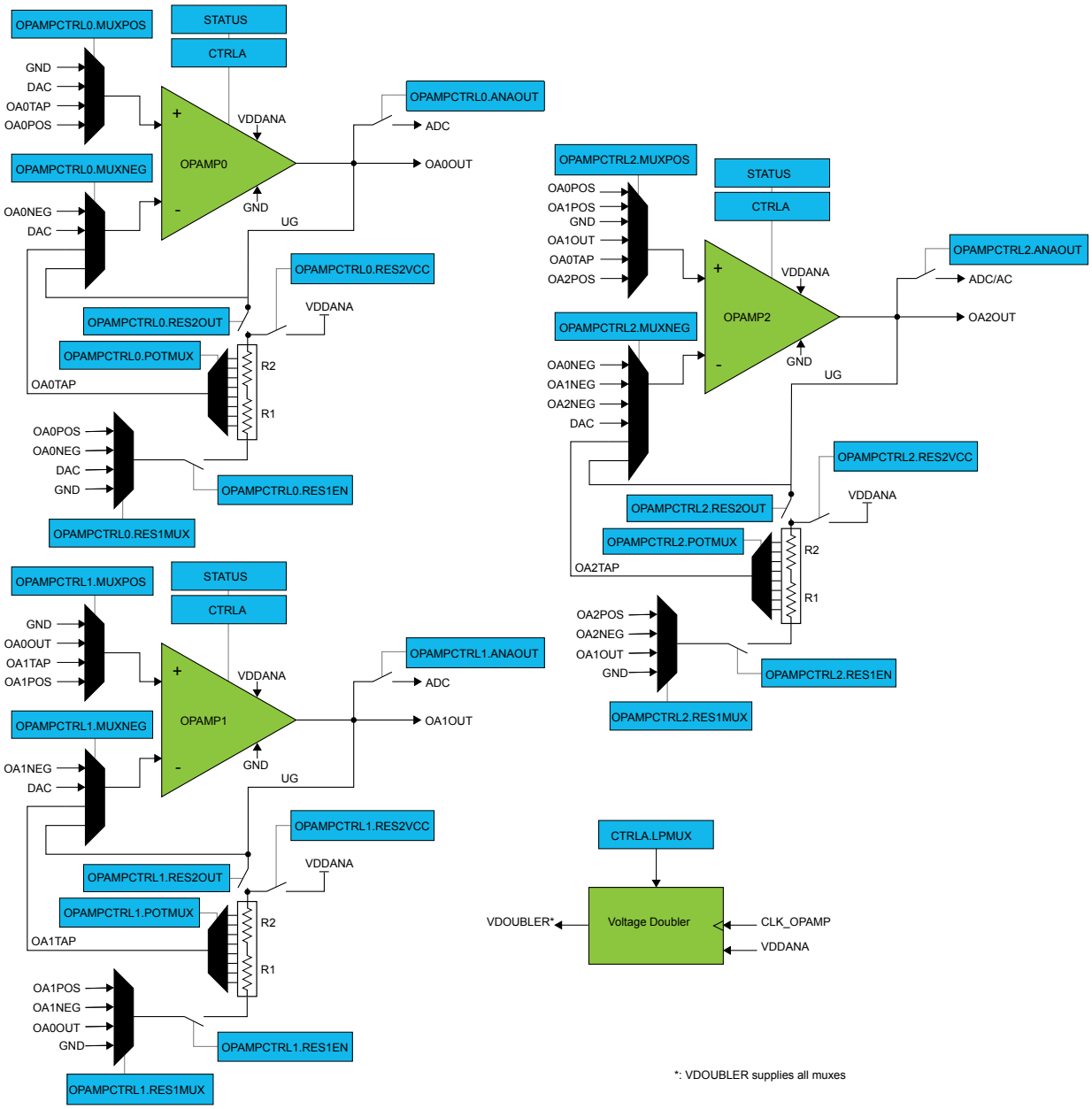
Four modes are available to select the trade-off between speed and power consumption to best fit the application requirements and optimize the power consumption.

41.2. Features

- Three individually configurable low power OPAMPs
- Rail-to-rail inputs
- Configurable resistor ladders for internal feedback
- Selectable configurations
  - Standalone OPAMP with flexible inputs
  - Unity gain amplifier
  - Non-inverting / inverting Programmable Gain Amplifier (PGA)
  - Cascaded PGAs
  - Instrumentation amplifier
  - Comparator with programmable hysteresis
- OPAMP output:
  - On I/O pins
  - As input for AC or ADC
- Flexible input selection:
  - I/O pins
  - DAC
  - Ground
- Low power options:
  - Selectable voltage doubler and propagation delay versus current consumption
  - On demand start-up for ADC and AC operations
- Offset/Gain measurement for calibration when used with the ADC

41.3. Block Diagram

Figure 41-1. OPAMP Block Diagram



41.4. Signal Description

Signal	Description	Type
OA0POS	OPAMP0 positive input	Analog input
OA0NEG	OPAMP0 negative input	Analog input
OA1POS	OPAMP1 positive input	Analog input

Signal	Description	Type
OA1NEG	OPAMP1 negative input	Analog input
OA2POS	OPAMP2 positive input	Analog input
OA2NEG	OPAMP2 negative input	Analog input
OA0OUT	OPAMP0 output	Analog output
OA1OUT	OPAMP1 output	Analog output
OA2OUT	OPAMP2 output	Analog output

One signal can be mapped on several pins.

Important:
When an analog peripheral is enabled, the analog output of the peripheral will interfere with the alternative functions of the output pads. This is also true even when the peripheral is used for internal purposes.

Analog inputs do not interfere with alternative pad functions.

Related Links

I/O Multiplexing and Considerations on page 29

41.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

41.5.1. I/O Lines

Using the OPAMP I/O lines requires the I/O pins to be configured. Refer to the *PORT - I/O Pin Controller* chapter for details.

41.5.2. Power Management

The OPAMP can operate in idle and standby sleep mode, according to the settings of the Run in Standby and On Demand bits in the OPAMP Control x registers (OPAMPCTRLx.RUNSTDBY and OPAMPCTRLx.ONDEMAND), as well as the Enable bit in the Control A register (CTRLA.ENABLE). Refer to *PM – Power Manager* for details on the different sleep modes.

Related Links

PM – Power Manager on page 186

41.5.3. Clocks

The OPAMP bus clock (CLK_OPAMP_APB) can be enabled and disabled in the Power Manager, and the default state of CLK_OPAMP_APB can be found in the *Peripheral Clock Masking*.

A clock (CLK_ULP32K) is required by the voltage doubler for low voltage operation (VCC < 2.5V). The CLK_ULP32K is a 32KHz clock which is provided by the OSCULP32K oscillator in the OSC32KCTRL module.

Related Links

Peripheral Clock Masking on page 151

41.5.4. DMA

Not applicable.

41.5.5. Interrupts

Not applicable.

41.5.6. Events

Not applicable.

41.5.7. Debug Operation

When the CPU is halted in debug mode the OPAMP continues normal operation. If the OPAMP is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

41.5.8. Register Access Protection

All registers with write-access can be write-protected optionally by the peripheral access controller (PAC). Refer to *PAC - Peripheral Access Controller* for details.

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

PAC write-protection does not apply to accesses through an external debugger.

Related Links

PAC - Peripheral Access Controller on page 58

41.5.9. Analog Connections

Each OPAMP has two I/O pins that can be used as analog inputs. These pins must be configured for analog operation before using them as OPAMP inputs.

If the DAC is to be used as OPAMP input, the DAC must be configured and enabled first.

Each OPAMP has one I/O pin that can be used as analog output. This pin must be configured for analog operation before using it as OPAMP output.

The analog signals of AC, ADC, DAC and OPAMP can be interconnected. The AC and ADC peripheral can request the OPAMP using an analog ONDEMAND functionality.

See *Analog Connections of Peripherals* for details.

Related Links

Analog Connections of Peripherals on page 34

41.5.10. Other dependencies

Not applicable.

41.6. Functional Description

41.6.1. Principle of Operation

Each OPAMP has one positive and one negative input. Each input may be chosen from either a selection of analog input pins, or internal inputs such as the DAC, the resistor ladder, and the ground and output of another OPAMP.

Each OPAMP can be configured with built-in feedback to support various functions with programmable or unity gain.

I/O pins are externally accessible so that the operational amplifier can be configured with external feedback.

All OPAMPs can be cascaded to support circuits such as differential amplifiers.

41.6.2. Basic Operation

Each operational amplifier can be configured in different modes, selected by the OPAMP Control x register (OPAMPCTRLx):

- Standalone operational amplifier
- Operational amplifier with built-in feedback

After being enabled, a start-up delay is added before the output of the operational amplifier is available. This start-up time is measured internally to account for environmental changes such as temperature or voltage supply level.

When the OPAMP is ready, the respective Ready x bit in the Status register is set (STATUS.READYx=1).

If the supply voltage is below 2.5V, the start-up time is also dependent on the voltage doubler. If the supply voltage is always above 2.5V, the voltage doubler can be disabled by setting the Low-Power Mux bit in the Control A Register (CTRLA.LPMUX).

41.6.2.1. Initialization

The OPAMP must be configured with the desired properties and inputs before it is enabled.

The asynchronous clocks CLK_OPAMP must be configured in the OSC32KCTRL module before enabling individual OPAMPs. See *OSC32KCTRL – 32KHz Oscillators Controller* for further details.

Related Links

OSC32KCTRL – 32KHz Oscillators Controller on page 264

41.6.2.2. Enabling, Disabling, and Resetting

The OPAMP is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The OPAMP is disabled by writing a '0' to CTRLA.ENABLE.

Each OPAMP sub-module is enabled by writing a '1' to the Enable bit in the OPAMP Control x register (OPAMPCTRLx.ENABLE). Each OPAMP sub-module is disabled by writing a '0'to OPAMPCTRLx.ENABLE.

The OPAMP module is reset by writing a '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the OPAMP will be reset to their initial state, and the OPAMP will be disabled. Refer to CTRLA on page 1024 for details.

41.6.3. DMA Operation

Not applicable.

41.6.4. Interrupts

Not applicable.

41.6.5. Events

Not applicable.

41.6.6. Sleep Mode Operation

The OPAMPs can also be used during sleep modes. The 32KHz clock source used by the voltage doubler must remain active. See Voltage Doubler for more details.

Each OPAMP x can be configured to behave differently in different sleep modes. The behavior is determined by the individual Run in Standby and On Demand bits in the OPAMP Control x registers (OPAMPCTRLx.RUNSTDBY, and OPAMPCTRLx.ONDEMAND), as well as the common Enable bit in the Control A register (CTRLA.ENABLE).

Table 41-1. Individual OPAMP Sleep Mode Operation

OPAMPCTRLx.RUNSTDBY	OPAMPCTRLx.ONDEMAND	CTRLA.ENABLE	Sleep Behavior
-	-	0	Disabled
0	0	1	Always run in all sleep modes except STANDBY sleep mode
0	1	1	Only run in all sleep modes except STANDBY sleep mode if requested by a peripheral.
1	0	1	Always run in all sleep mode
1	1	1	Only run in all sleep modes if requested by a peripheral.

Note:
When OPAMPCTRLx.ONDEMAND=1, the analog block is powered off for the lowest power consumption if it is not requested.

When requested, a start-up time delay is necessary when the system returns from sleep. The start-up time is depending on the Bias Selection bits in the OPAMP Control x register (OPAMPCTRLx.BIAS) and the corresponding speed/current consumption requirements.

41.6.7. Synchronization

Not applicable.

41.6.8. Configuring the Operational Amplifiers

Each individual operational amplifier is configured by its respective Operational Amplifier Control x register (OPAMPCTRLx). These settings must be configured before the amplifier is started.

- Select the positive input in OPAMPCTRLx.MUXPOS.
- Select the negative input in OPAMPCTRLx.MUXNEG.
- Select RES1EN if resistor ladder is used.
- Select the input for the resistor ladder in OPAMPCTRLx.RES1MUX.
- Select the potentiometer selection of the resistor ladder in OPAMPCTRLx.POTMUX.
- Select the VCC input for the resistor ladder in OPAMPCTRLx.RES2VCC.
- Connect the operational amplifier output to the resistor ladder using OPAMPCTRLx.RES2OUT.
- Select the trade-off between speed and energy consumption in OPAMPCTRLx.BIAS.

41.6.9. Standalone Mode

Each operational amplifier can be used as standalone amplifier. In this mode, positive input, negative input and the output are routed from/to external I/Os, requiring external feedback. OPAMPs can also be cascaded to support multiple OPAMP configurations. Refer to Operational Amplifier Control x register (OPAMPCTRLx) for further details on how to configure OPAMP I/Os.

41.6.10. Built-in Modes

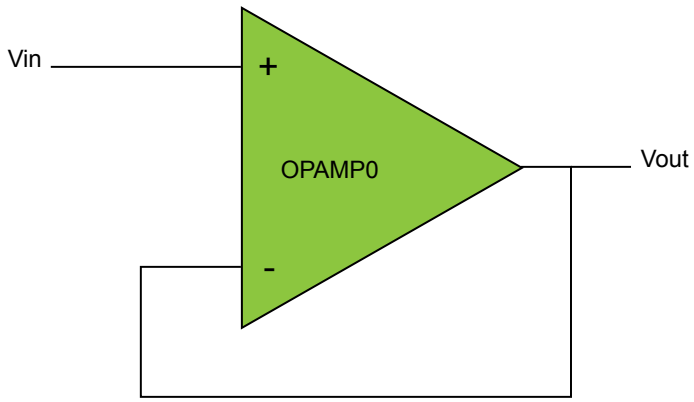
41.6.10.1. Voltage Follower

In this mode the unity gain path is selected for the negative input. The OPAMPCTRLx register can be configured as follows:

Table 41-2. Configuration - Three Independent Unitary Gain Followers

	MUXPOS	MUXNEG	RES1MUX	POTMUX	RES2VCC	RES2OUT	RES1EN	ANAOUT
OPAMP0	000	010	11	000	0	0	0	0
OPAMP1	000	010	11	000	0	0	0	0
OPAMP2	000	010	11	000	0	0	0	0

Figure 41-2. Voltage follower



41.6.10.2. Inverting PGA

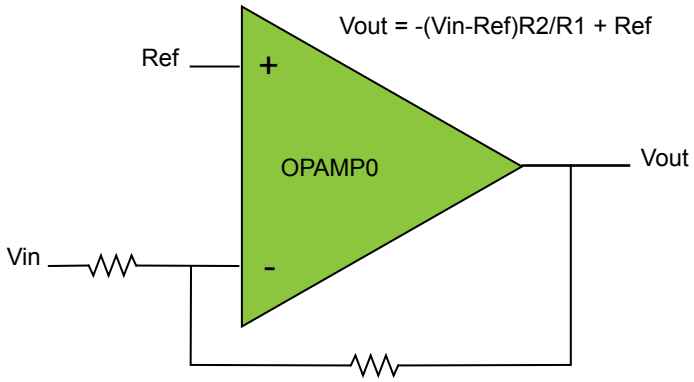
For inverting programmable gain amplifier operation, the OPAMPCTRLx registers can be configured as follows:

Table 41-3. Configuration - Three Independent Inverting PGAs

	MUXPOS	MUXNEG	RES1MUX	POTMUX	RES2VCC	RES2OUT	RES1EN	ANAOUT
OPAMP0	011	001	01	100	0	1	1	0
OPAMP1	011	001	01	100	0	1	1	0
OPAMP2	011	001	01	100	0	1	1	0

Inverting PGA (Example: Vout=-3.Vin, R1=4R, R2=12R)

Figure 41-3. Inverting PGA



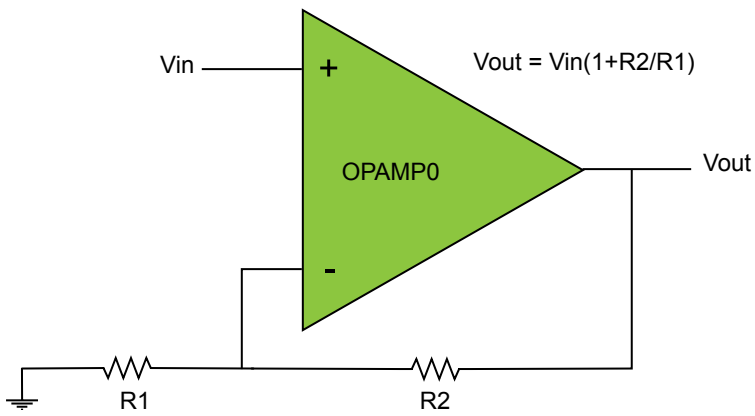
41.6.10.3. Non-Inverting PGA

For non-inverting programmable gain amplifier operation, the OPAMPCTRLx registers can be configured as follows:

Table 41-4. Configuration - Three Independent Non-Inverting PGAs (Example: Vout=4.Vin, R1=4R, R2=12R)

	MUXPOS	MUXNEG	RES1MUX	POTMUX	RES2VCC	RES2OUT	RES1EN	ANAOUT
OPAMP0	000	001	11	100	0	1	1	0
OPAMP1	000	001	11	100	0	1	1	0
OPAMP2	000	001	11	100	0	1	1	0

Figure 41-4. Non-Inverting PGA



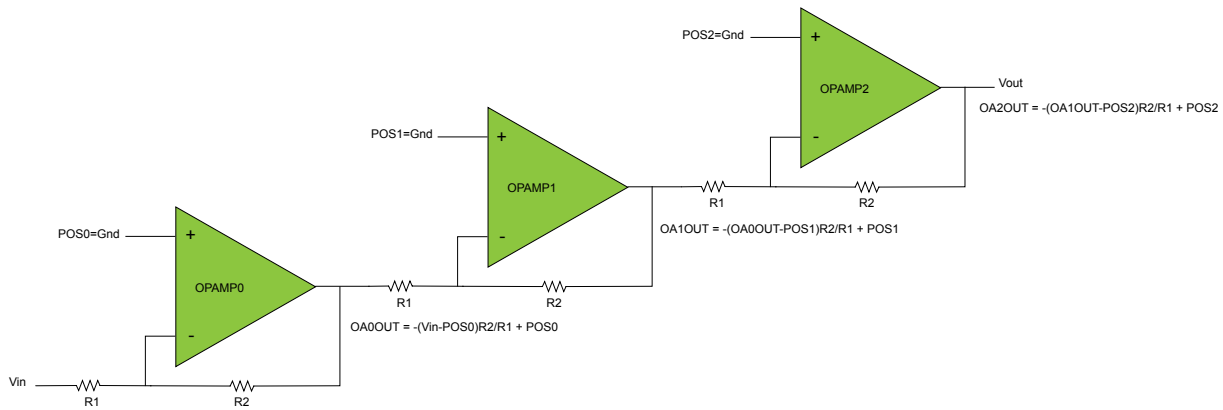
41.6.10.4. Cascaded Inverting PGA

The OPAMPs can be configured as three cascaded, inverting PGAs using these settings in OPAMPCTRLx:

Table 41-5. Cascade of three inverting PGAs (Example: R1=4R, R2=12R)

	MUXPOS	MUXNEG	RES1MUX	POTMUX	RES2VCC	RES2OUT	RES1EN	ANAOUT
OPAMP0	011	001	01	100	0	1	1	0
OPAMP1	011	001	10	100	0	1	1	0
OPAMP2	011	001	10	100	0	1	1	0

Figure 41-5. Cascaded Inverting PGA



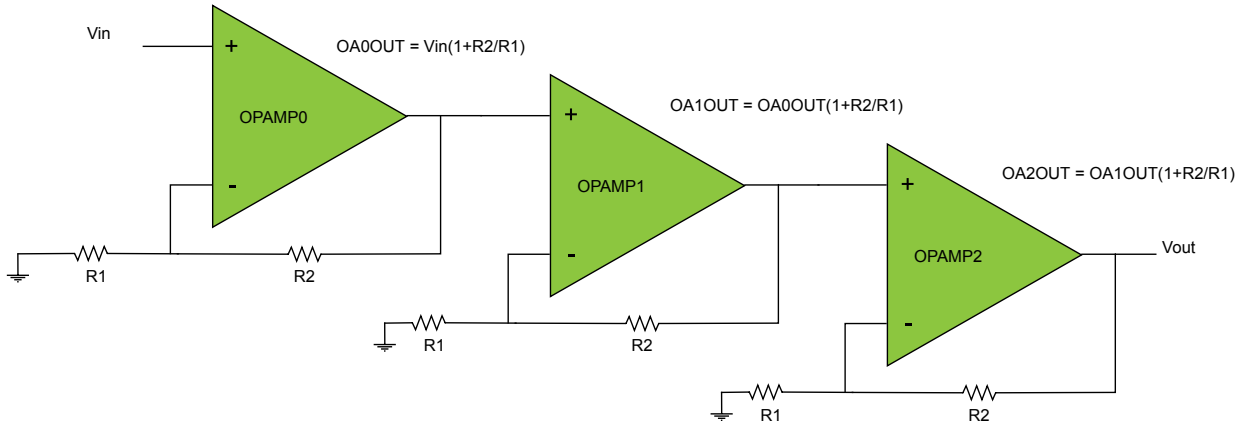
41.6.10.5. Cascaded Non-Inverting PGA

The OPAMPs can be configured as three cascaded, non-inverting PGAs using these settings in OPAMPCTRLx:

Table 41-6. Cascaded Non-Inverting PGA (Exemple: R1=4R, R2=12R)

	MUXPOS	MUXNEG	RES1MUX	POTMUX	RES2VCC	RES2OUT	RES1EN	ANAOUT
OPAMP0	000	001	11	100	0	1	1	0
OPAMP1	010	001	11	100	0	1	1	0
OPAMP2	010	001	11	100	0	1	1	0

Figure 41-6. Cascaded Non-Inverting PGA



41.6.10.6. Two OPAMPs Differential Amplifier

In this mode, OPAMP0 can be coupled with OPAMP1 or OPAMP1 with OPAMP2 in order to amplify a differtial signal.

To configure OPAMP0 and OPAMP1 as differential amplifier, the OPAMPCTRLx register can be configured as follows:

Table 41-7. OPAMP0 OPAMP1 Differential Amplifier (Example: R1=4R, R2=12R)

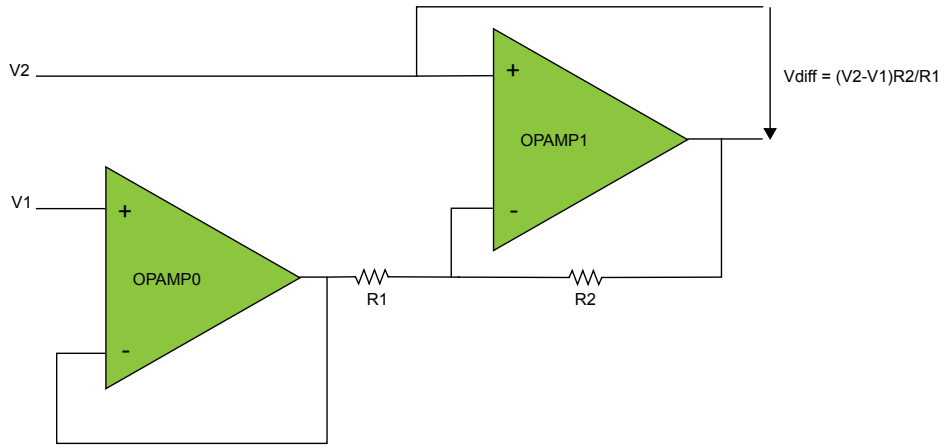
	MUXPOS	MUXNEG	RES1MUX	POTMUX	RES2VCC	RES2OUT	RES1EN	ANAOUT
OPAMP0	000	010	00	000	0	0	0	0
OPAMP1	000	001	10	100	0	1	1	0
OPAMP2	000	000	00	000	0	0	0	0

To configure OPAMP1 and OPAMP2 as differential amplifier, the OPAMPCTRLx register can be configured as follows:

Table 41-8. OPAMP1 OPAMP2 Differential Amplifier (Example: R1=4R, R2=12R)

	MUXPOS	MUXNEG	RES1MUX	POTMUX	RES2VCC	RES2OUT	RES1EN	ANAOUT
OPAMP0	000	000	00	000	0	0	0	0
OPAMP1	000	010	00	000	0	1	0	0
OPAMP2	000	001	10	100	0	1	1	0

Figure 41-7. OPAMP0 OPAMP1 Differential Amplifier



41.6.10.7. Instrumentation Amplifier

In this mode, OPAMP0 and OPAMP1 are configured as voltage followers. The OPAMPCTRLx register can be configured as follows:

Table 41-9. Instrumentation Amplifier Configuration

	MUXPOS	MUXNEG	RES1MUX	POTMUX	RES2VCC	RES2OUT	RES1EN	ANAOUT
OPAMP0	000	010	11	010	0	1	1	0
OPAMP1	000	010	11	000	0	0	0	0
OPAMP2	110	001	10	010	0	1	1	0

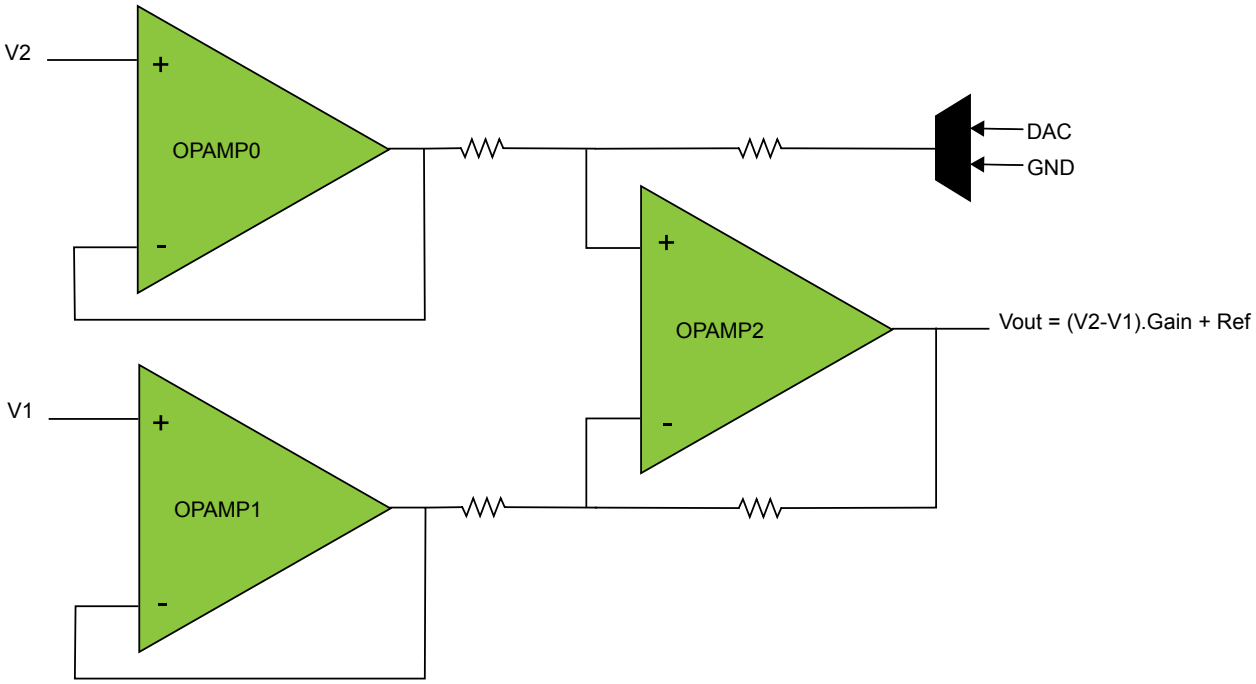
The resistor ladders associated with OPAMP0 and OPAMP2 must be configured as follows in order to select the appropriate gain:

Table 41-10. Instrumentation Amplifier Gain Selection

OPAMPCTRL0.POTMUX	OPAMPCTRL2.POTMUX	GAIN
0x7	Reserved	Reserved
0x6	0x0	1/7
0x5	Reserved	Reserved
0x4	0x1	1/3
0x3	Reserved	Reserved
0x2	0x2	1
0x1	0x4	3
0x0	0x6	7

Note: Either the DAC or GND must be the reference, selected by the OPAMPCTRL0.RES1MUX bits. Refer to OPAMPCTRL0, OPAMPCTRL1 and OPAMPCTRL2 for details.

Figure 41-8. Instrumentation amplifier



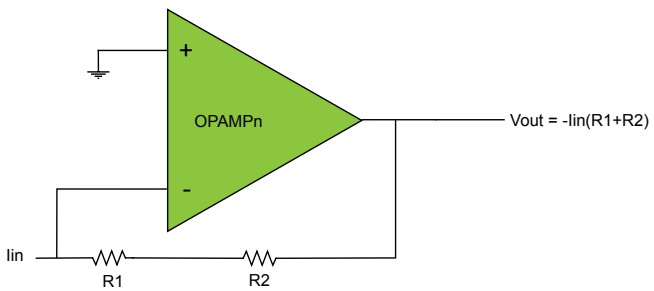
41.6.10.8. Transimpedance amplifier

Each OPAMP can be configured as a transimpedance amplifier (current to voltage converter). In this mode the positive input is connected to ground. The negative input is connected to the output through the resistor ladder. The OPAMPCTRLx register can be configured as follows:

Table 41-11. Transimpedance Amplifier

	MUXPOS	MUXNEG	RES1MUX	POTMUX	RES2VCC	RES2OUT	RES1EN	ANAOUT
OPAMP0	011	000	01	000	0	1	1	0
OPAMP1	011	000	01	000	0	1	1	0
OPAMP2	011	000	01	000	0	1	1	0

Figure 41-9. Transimpedance Amplifier



41.6.10.9. Programmable Hysteresis

Each OPAMP can be configured as an inverting or non-inverting comparator with programmable hysteresis. Applying hysteresis will prevent constant toggling of the output, caused by noise when the input signals are close to each other.

In both inverting and non-inverting comparator configurations the positive input is connected to the resistor ladder. When OPAMP is configured as an inverting comparator with programmable hysteresis, the input voltage must be applied to the negative input and RES1MUX must be connected to the ground. When an OPAMP is configured as a non-inverting comparator with programmable hysteresis, the input voltage must be applied to RES1MUX and the negative input must be connected to the ground.

To configure an OPAMP as an inverting comparator with programmable hysteresis, the OPAMPCTRLx register can be configured as follows:

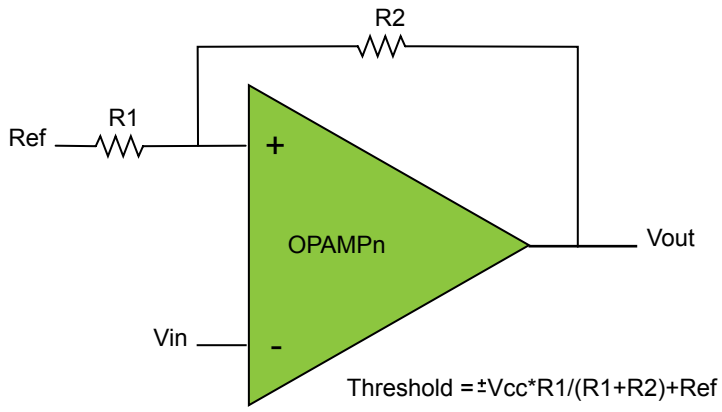
Table 41-12. Configuration of Input Muxes for OPAMP0 and OPAMP1 (Example: Vth = 3/4*Vcc, Ref = GND)

	MUXPOS	MUXNEG	RES1MUX	POTMUX	RES2VCC	RES2OUT	RES1EN	ANAOUT
OPAMP0	001	000	11	001	0	1	1	0
OPAMP1	001	000	11	001	0	1	1	0
OPAMP2	001	000	11	001	0	1	1	0

Table 41-13. POTMUX [2:0]: Potentiometer Selection

Value	R1	R2	Threshold = Vcc * R1 / (R1 + R2)
0x0	14R	2R	7/8 * Vcc
0x1	12R	4R	3/4 * Vcc
0x2	8R	8R	1/2 * Vcc
0x3	6R	10R	3/8 * Vcc
0x4	4R	12R	1/4 * Vcc
0x5	3R	13R	3/16 * Vcc
0x6	2R	14R	1/8 * Vcc
0x7	R	15R	1/16 * Vcc

Figure 41-10. Inverting comparator with programmable hysteresis



To configure an OPAMP as a non-inverting comparator with programmable hysteresis, the OPAMPCTRLx register can be configured as follows:

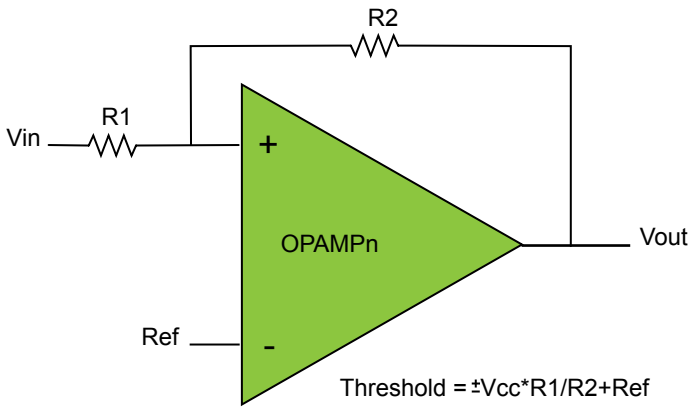
Table 41-14. Configuration of input muxes for OPAMP0 and OPAMP1 (Example: Vth = 1/3*Vcc, Ref = Gnd)

	MUXPOS	MUXNEG	RES1MUX	POTMUX	RES2VCC	RES2OUT	RES1EN	ANAOUT
OPAMP0	001	000	00	100	0	1	1	0
OPAMP1	001	000	00	100	0	1	1	0
OPAMP2	001	000	00	100	0	1	1	0

Table 41-15. POTMUX [2:0]: Potentiometer Selection

Value	R1	R2	Threshold = Vcc * R1 / R2
0x0	14R	2R	Vcc * 7 (unused)
0x1	12R	4R	Vcc * 3 (unused)
0x2	8R	8R	Vcc (unused)
0x3	6R	10R	0.6* Vcc
0x4	4R	12R	1/3 * Vcc
0x5	3R	13R	3/13 *Vcc
0x6	2R	14R	1/7 * Vcc
0x7	R	15R	1/15 * Vcc

Figure 41-11. Non-Inverting comparator with programmable hysteresis



41.6.11. ADC Driver

41.6.11.1. Buffer/PGA for ADC

Each OPAMP can be configured as a buffer or a PGA for the other modules (such as ADC or AC). OPAMPs can also be cascaded to increase the programmable gain.

The output to the OPAMP must be enabled by writing a '1' to the Analog Output bit in the Operational Amplifier x Control register (OPAMPCTRLx.ANAOUT). The ADC input mux must be configured to select OPAMP as input. Refer to *ADC – Analog-to-Digital Converter* for details on configuring the ADC.

Related Links

ADC – Analog-to-Digital Converter on page 1030

41.6.11.2. Offset and Gain Compensation

When the OPAMP is used in combination with the ADC, the OPAMP offset and gain errors can be compensated. To calculate offset and gain error compensation values

1. Configure OPAMP as Voltage Follower
2. Route the OPAMP output to the ADC:
   - Write a '1' to the Analog Output bit in the Operational Amplifier x Control register (OPAMPCTRLx.ANAOUT)
   - Select the OPAMP as input for the ADC, see *ADC – Analog-to-Digital Converter*.
3. Measure and set the Offset Correction value for the ADC OFFSETCORR register as in Offset Compensation on page 1021.
4. Measure and set the Gain Correction value for the ADC GAINCORR register as in Gain Compensation on page 1021.

The offset error compensation must be determined before gain error compensation.

The relation for offset and gain error compensation is shown in this equation:

Result = (converted value + OFFSETCORR)*GAINCORR

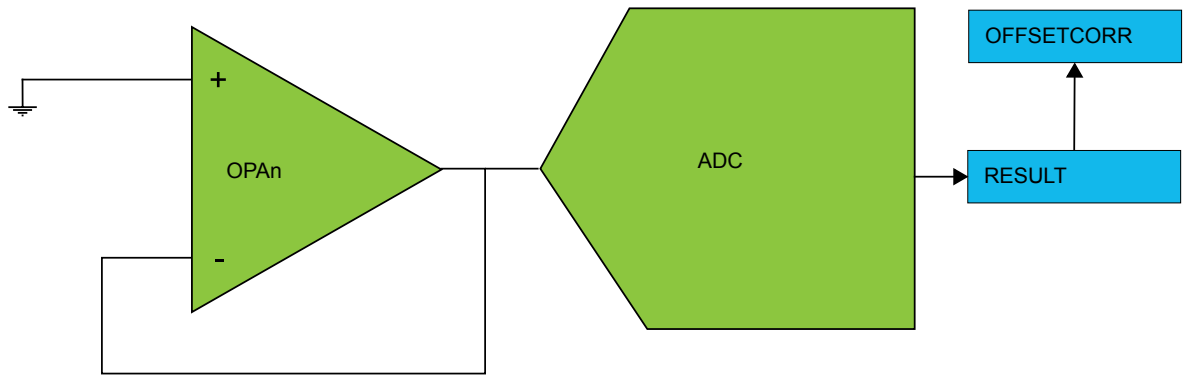
Related Links

ADC – Analog-to-Digital Converter on page 1030

41.6.11.3. Offset Compensation

To determine the offset compensation value, the positive input must be tied to ground. The result of the ADC conversion gives directly the offset compensation value that must be written in the ADC OFFSETCORR register.

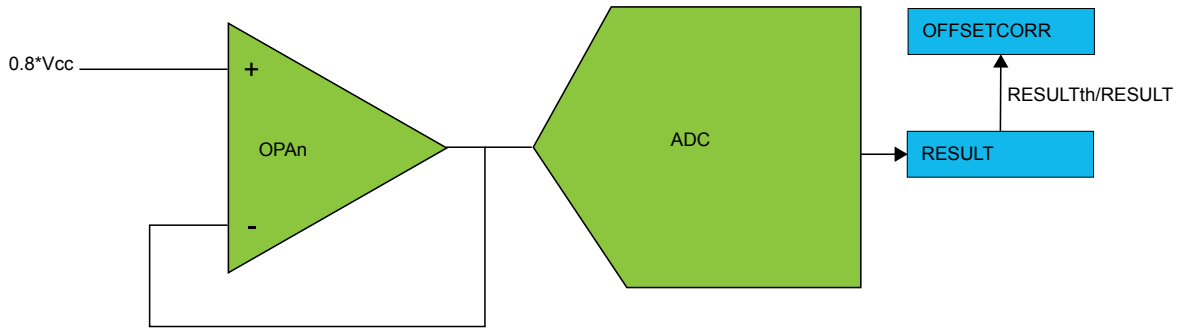
Figure 41-12. Offset Compensation



41.6.11.4. Gain Compensation

To perform gain compensation positive input must be close to VDD, but less (e.g. 0.8*Vref for instance) to avoid ADC saturation. The value for gain error compensation is obtained by dividing the theoretical ADC conversion result by the result from measurement. The obtained value for gain error compensation must be written in the ADC GAINCORR register.

Figure 41-13. Gain Compensation



41.6.12. AC Driver

One or several OPAMPs can be configured as input for the AC. The AC input mux must be appropriately configured to select OPAMP as input.

Related Links

AC – Analog Comparators on page 1072

41.6.13. Input Connection to DAC

The DAC can be used as a reference. This is configured by the corresponding OPAMPCTRLx.MUXPOS and OPAMPCTRLx.RES1MUX bits.

41.6.14. Voltage Doubler

The OPAMP peripheral contains a voltage doubler for the analog multiplexer switches to ensure proper operation for a supply voltage below 2.5V. Aside from the multiplexers, no other supply voltages are affected by the voltage doubler.

The voltage doubler is normally switched on/off automatically, based on the supply level. If the supply voltage is guaranteed to be above 2.5V, the voltage doubler can be completely disabled by writing the Low-Power Mux bit in the Control Register (CTRLA.LPMUX).

When enabling OPAMPs, additional start-up time is required for the voltage doubler to settle. Disabling the voltage doubler saves power and reduces the startup time.

41.6.15. Performance vs. Power Consumption

It is possible to tradeoff speed versus power efficiency to get the shortest possible propagation delay or the lowest power consumption.

The speed setting is configured for each amplifier individually by the Bias Control field in the Operational Amplifier x Control register (OPAMPCTRLx.BIAS). The BIAS bits select the amount of bias current provided to the operational amplifiers. This will also affect the start-up time.

41.7. Register Summary

Offset	Name	Bit pos.								
0x00	CTRLA	7:0	LPMUX						ENABLE	SWRST
0x01	Reserved	7:0								

Offset	Name	Bit pos.								
0x02	STATUS	7:0						READY2	READY1	READY0
0x03	Reserved									
0x04	OPAMPCTRL0	7:0	ONDEMAND	RUNSTDBY		BIAS[1:0]		ANAOUT	ENABLE	
0x05		15:8	POTMUX[2:0]			RES1MUX[1:0]		RES1EN	RES2VCC	RES2OUT
0x06		23:16		MUXNEG[2:0]				MUXPOS[2:0]		
0x07		31:24								
0x08	OPAMPCTRL1	7:0	ONDEMAND	RUNSTDBY		BIAS[1:0]		ANAOUT	ENABLE	
0x09		15:8	POTMUX[2:0]			RES1MUX[1:0]		RES1EN	RES2VCC	RES2OUT
0x0A		23:16		MUXNEG[2:0]				MUXPOS[2:0]		
0x0B		31:24								
0x0C	OPAMPCTRL2	7:0	ONDEMAND	RUNSTDBY		BIAS[1:0]		ANAOUT	ENABLE	
0x0D		15:8	POTMUX[2:0]			RES1MUX[1:0]		RES1EN	RES2VCC	RES2OUT
0x0E		23:16		MUXNEG[2:0]				MUXPOS[2:0]		
0x0F		31:24								

41.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection.

41.8.1. Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	LPMUX						ENABLE	SWRST
Access	R/W						R/W	R/W
Reset	0						0	0

Bit 7 – LPMUX: Low-Power Mux

Value	Description
0	The analog input muxes have low resistance, but consume more power at lower voltages (e.g., are driven by the voltage doubler).
1	The analog input muxes have high resistance, but consume less power at lower voltages (e.g., the voltage doubler is disabled).

Bit 1 – ENABLE: Enable

Value	Description
0	The peripheral is disabled.
1	The peripheral is enabled. Each OPAMP must also be enabled individually by the Enable bit in the corresponding OPAMP Control register (OPAMPCTRLx.ENABLE).

Bit 0 – SWRST: Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the MODULE to their initial state, and the OPAMP will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

41.8.2. Status

Name: STATUS
Offset: 0x02
Reset: 0x00
Property: –

Bit	7	6	5	4	3	2	1	0
						READY2	READY1	READY0
Access						R	R	R
Reset						0	0	0

Bits 2,1,0 – READYx: OPAMP x Ready

This bit is set when the OPAMPx output is ready.

This bit is cleared when the output of OPAMPx is not ready.

41.8.3. OPAMP Control x

Name: OPAMPCTRLx
Offset: 0x04+4*x, [x=0..2]
Reset: 0x00000080
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
		MUXNEG[2:0]				MUXPOS[2:0]		
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bit	15	14	13	12	11	10	9	8
	POTMUX[2:0]			RES1MUX[1:0]		RES1EN	RES2VCC	RES2OUT
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY		BIAS[1:0]		ANAOUT	ENABLE	
Access	R/W	R/W		R/W	R/W	R/W	R/W	
Reset	0	0		0	0	0	0	

Bits 22:20 – MUXNEG[2:0]: Negative Input Mux Selection
Selection on negative input for operational amplifier x.

Value	OPAMPx	Name	Description
0x0	x=0,1,2	OAxNEG	Negative I/O pin
0x1	x=0,1,2	OAxTAP	Resistor ladder x taps
0x2	x=0,1,2	OAxOUT	OPAMPx output
0x3	x=0,1	DAC	DAC output
	x=2	OA0NEG	Negative I/O pin OPA0
0x4	x=0,1	Reserved	
	x=2	OA1NEG	Negative I/O pin OPA1
0x5	x=0,1	Reserved	
	x=2	DAC	DAC output
0x6	x=0,1,2	Reserved	
0x7	x=0,1,2	Reserved	

Bits 18:16 – MUXPOS[2:0]: Positive Input Mux Selection
Selection on positive input for operational amplifier x.

Value	OPAMPx	Name	Description
0x0	x=0,1,2	OAxPOS	Positive I/O pin
0x1	x=0,1,2	OAxTAP	Resistor ladder x taps
0x2	x=0	DAC	DAC output
	x=1	OA0OUT	OPAMP0 output
	x=2	OA1OUT	OPAMP1 output
0x3	x=0,1,2	GND	Ground
0x4	x=0,1	Reserved	
	x=2	OA0POS	Positive I/O pin OPA0
0x5	x=0,1	Reserved	
	x=2	OA1POS	Positive I/O pin OPA1
0x6	x=0,1	Reserved	
	x=2	OA0TAP	Resistor ladder 0 taps
0x7	x=0,1,2	Reserved	

Bits 15:13 – POTMUX[2:0]: Potentiometer selection
Resistor selection bits control a numeric potentiometer with eight fixed values.

Value	R1	R2	Gain = R2/R1
0x0	14R	2R	1/7
0x1	12R	4R	1/3
0x2	8R	8R	1
0x3	6R	10R	1 + 2/3
0x4	4R	12R	3
0x5	3R	13R	4 + 1/3
0x6	2R	14R	7
0x7	R	15R	15

Bits 12:11 – RES1MUX[1:0]: Resistor 1 Mux
These bits select the connection of R1 resistor of the potentiometer.

Value	OPAMPx	Name	Description
0x0	x=0,1,2	OAxPOS	Positive inout of OPAMPx
0x1	x=0,1,2	OAxNEG	Negative input of OPAMPx

Value	OPAMPx	Name	Description
0x2	x=0	DAC	DAC output
	x=1	OA0OUT	OPAMP0 output
	x=2	OA1OUT	OPAMP1 output
0x3	x=0,1,2	GND	

Bit 10 – RES1EN: Resistor 1 Enable

Value	Description
0	R1 disconnected from RES1MUX.
1	R1 connected to RES1MUX.

Bit 9 – RES2VCC: Resistor ladder To VCC

Value	Description
0	Swith open.
1	Switch closed.

Bit 8 – RES2OUT: Resistor ladder To Output

Value	Description
0	Swith open.
1	Switch closed.

Bit 7 – ONDEMAND: On Demand Control
The On Demand operation mode allows the OPAMPx to be enabled or disabled, depending on other peripheral requests.

Value	Description
0	The OPAMPx is always on, if enabled.
1	The OPAMPx is enabled when a peripheral is requesting the OPAMPx to be used as an input. The OPAMPx is disabled if no peripheral is requesting it as an input.

Bit 6 – RUNSTDBY: Run in Standby
This bit controls how the OPAMPx behaves during standby sleep mode:

Value	Description
0	The OPAMPx is disabled in standby sleep mode.
1	The OPAMPx is not stopped in standby sleep mode. If OPAMPCTRLx.ONDEMAND=1, the OPAMPx will be running when a peripheral is requesting it as an input. If OPAMPCTRLx.ONDEMAND=0, OPAMPx will always be running in standby sleep mode.

Bits 4:3 – BIAS[1:0]: Bias Selection
These bits are used to select the bias mode.

Value	Name	Description
0x0	Mode 0	Minimum current consumption, but the slowest mode
0x1	Mode 1	Low current consumption, slow speed
0x2	Mode 2	High current consumption, fast speed
0x3	Mode 3	Maximum current consumption but the fastest mode

Bit 2 – ANAOUT: Analog Output
This bit controls a switch connected to the OPAMP output.

Value	Description
0	Swith open. No ADC or AC connection.
1	Switch closed. OPAMP output is connected to the ADC or AC input.

Bit 1 – ENABLE: Operational Amplifier Enable

Value	Description
0	The OPAMPx is disabled
1	The OPAMPx is enabled

42. ADC – Analog-to-Digital Converter

42.1. Overview

The Analog-to-Digital Converter (ADC) converts analog signals to digital values. The ADC has up to 12-bit resolution, and is capable of a sampling rate of up to 1MSPS. The input selection is flexible, and both differential and single-ended measurements can be performed. In addition, several internal signal inputs are available. The ADC can provide both signed and unsigned results.

ADC measurements can be started by either application software or an incoming event from another peripheral in the device. ADC measurements can be started with predictable timing, and without software intervention.

Both internal and external reference voltages can be used.

An integrated temperature sensor is available for use with the ADC. The bandgap voltage as well as the scaled I/O and core voltages can also be measured by the ADC.

The ADC has a compare function for accurate monitoring of user-defined thresholds, with minimum software intervention required.

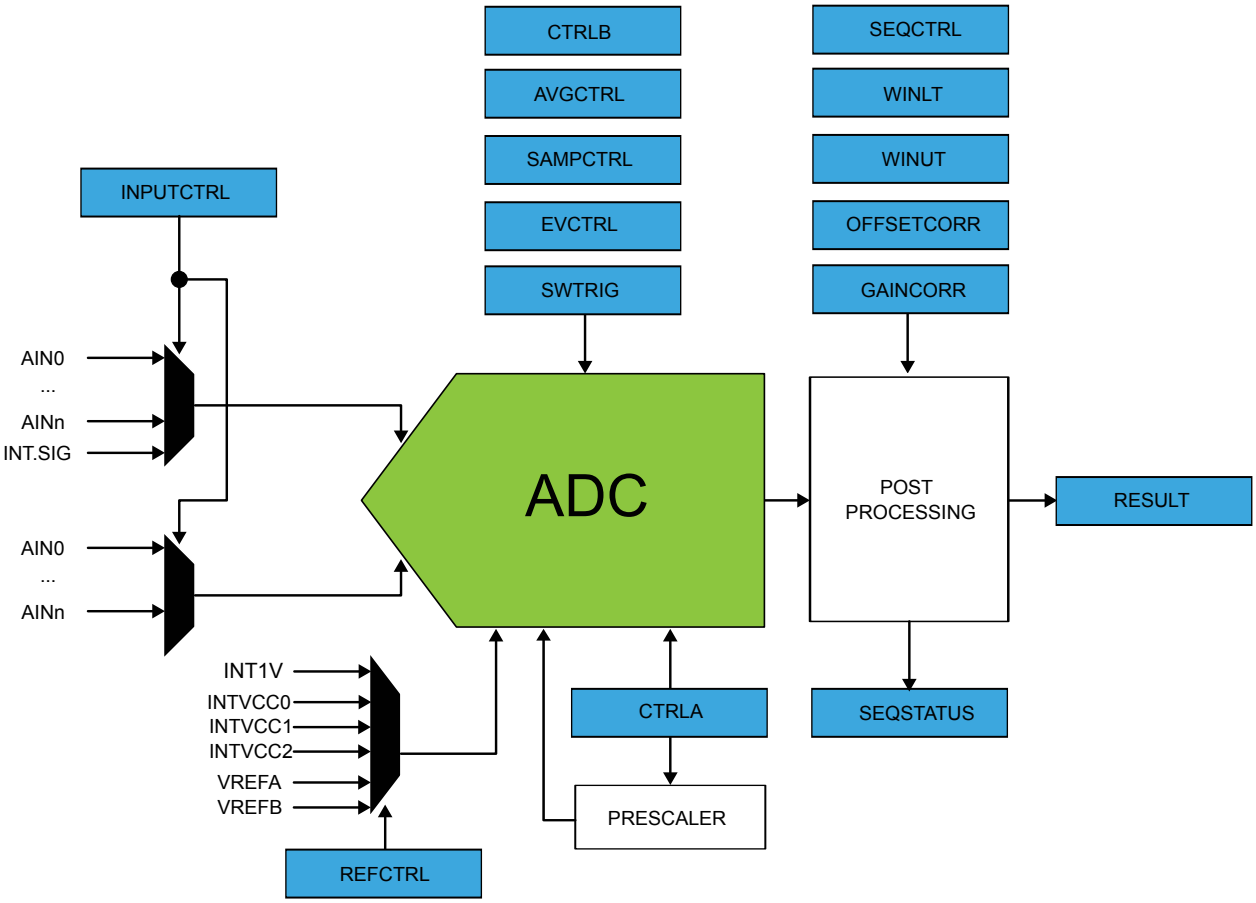
The ADC can be configured for 8-, 10- or 12-bit results. ADC conversion results are provided left- or right-adjusted, which eases calculation when the result is represented as a signed value. It is possible to use DMA to move ADC results directly to memory or peripherals when conversions are done.

42.2. Features

- 8-, 10- or 12-bit resolution
- Up to 1,000,000 samples per second (1 MSPS)
- Differential and single-ended inputs
  - Up to 20 analog inputs
    28 positive and 10 negative, including internal and external
- Internal inputs:
  - Internal temperature sensor
  - Bandgap voltage
  - Scaled core supply
  - Scaled I/O supply
  - DAC
- Single, continuous and sequencing options
- Windowing monitor with selectable channel
- Conversion range: V_{ref} = [1.0V to VDD_{ANA}]
- Built-in internal reference and external reference options
- Event-triggered conversion for accurate timing (one event input)
- Optional DMA transfer of conversion settings or result
- Hardware gain and offset compensation
- Averaging and oversampling with decimation to support up to 16-bit result
- Selectable sampling time
- Flexible Power / Throughput rate management

42.3. Block Diagram

Figure 42-1. ADC Block Diagram



42.4. Signal Description

Signal	Description	Type
VREFA	Analog input	External reference voltage A
VREFB	Analog input	External reference voltage B
AIN[19..0]	Analog input	Analog input channels

Note: One signal can be mapped on several pins.

Related Links

Configuration Summary on page 15
I/O Multiplexing and Considerations on page 29

42.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

42.5.1. I/O Lines

Using the ADC's I/O lines requires the I/O pins to be configured using the port configuration (PORT).

Related Links

PORT: IO Pin Controller on page 506

42.5.2. Power Management

The ADC will continue to operate in any sleep mode where the selected source clock is running. The ADC's interrupts can be used to wake up the device from sleep modes. Events connected to the event system can trigger other operations in the system without exiting sleep modes.

Related Links

PM – Power Manager on page 186

42.5.3. Clocks

The ADC bus clock (CLK_APB_ADCx) can be enabled in the Main Clock, which also defines the default state.

The ADC requires a generic clock (GCLK_ADC). This clock must be configured and enabled in the Generic Clock Controller (GCLK) before using the ADC.

A generic clock is asynchronous to the bus clock. Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains. Refer to Synchronization for further details.

Related Links

Peripheral Clock Masking on page 151

GCLK - Generic Clock Controller on page 131

42.5.4. DMA

The DMA request line is connected to the DMA Controller (DMAC). Using the ADC DMA requests requires the DMA Controller to be configured first.

Related Links

DMAC – Direct Memory Access Controller on page 399

42.5.5. Interrupts

The interrupt request line is connected to the interrupt controller. Using the ADC interrupt requires the interrupt controller to be configured first.

Related Links

Nested Vector Interrupt Controller on page 51

42.5.6. Events

The events are connected to the Event System.

Related Links

EVSYS – Event System on page 536

42.5.7. Debug Operation

When the CPU is halted in debug mode the ADC will halt normal operation. The ADC can be forced to continue operation during debugging. Refer to DBGCTRL on page 1066 for details.

42.5.8. Register Access Protection

All registers with write-access are optionally write-protected by the peripheral access controller (PAC), except the following register:

- Interrupt Flag Status and Clear (INTFLAG) register

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

PAC write-protection does not apply to accesses through an external debugger.

Related Links

PAC - Peripheral Access Controller on page 58

42.5.9. Analog Connections

I/O-pins (AINx), as well as the VREFA/VREFB reference voltage pins are analog inputs to the ADC.

The analog signals of AC, ADC, DAC and OPAMP can be interconnected. The AC and ADC peripheral can request the OPAMP using an analog ONDEMAND functionality.

See *Analog Connections of Peripherals* for details.

Related Links

Analog Connections of Peripherals on page 34

42.5.10. Calibration

The BIAS and LINEARITY calibration values from the production test must be loaded from the NVM Software Calibration Area into the ADC Calibration register (CALIB) by software to achieve specified accuracy.

Related Links

NVM Software Calibration Area Mapping on page 47

42.6. Functional Description

42.6.1. Principle of Operation

By default, the ADC provides results with 12-bit resolution. 8-bit or 10-bit results can be selected in order to reduce the conversion time, see Conversion Timing and Sampling Rate on page 1035.

The ADC has an oversampling with decimation option that can extend the resolution to 16 bits. The input values can be either internal (e.g., internal temperature sensor) or external (connected I/O pins). The user can also configure whether the conversion should be single-ended or differential.

42.6.2. Basic Operation

42.6.2.1. Initialization

The following registers are enable-protected, meaning that they can only be written when the ADC is disabled (CTRLA.ENABLE=0):

- Control B register (CTRLB)
- Reference Control register (REFCTRL)
- Event Control register (EVCTRL)
- Calibration register (CALIB)

Enable-protection is denoted by the "Enable-Protected" property in the register description.

42.6.2.2. Enabling, Disabling and Resetting

The ADC is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The ADC is disabled by writing CTRLA.ENABLE=0.

The ADC is reset by writing a '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the ADC, except DBGCTRL, will be reset to their initial state, and the ADC will be disabled. Refer to CTRLA on page 1045 for details.

42.6.2.3. Operation

In the most basic configuration, the ADC samples values from the configured internal or external sources (INPUTCTRL register). The rate of the conversion depends on the combination of the GCLK_ADC frequency and the clock prescaler.

To convert analog values to digital values, the ADC needs to be initialized first, as described in Initialization. Data conversion can be started either manually by setting the Start bit in the Software Trigger register (SWTRIG.START=1), or automatically by configuring an automatic trigger to initiate the conversions. A free-running mode can be used to continuously convert an input channel. When using free-running mode the first conversion must be started, while subsequent conversions will start automatically at the end of previous conversions.

The result of the conversion is stored in the Result register (RESULT) overwriting the result from the previous conversion.

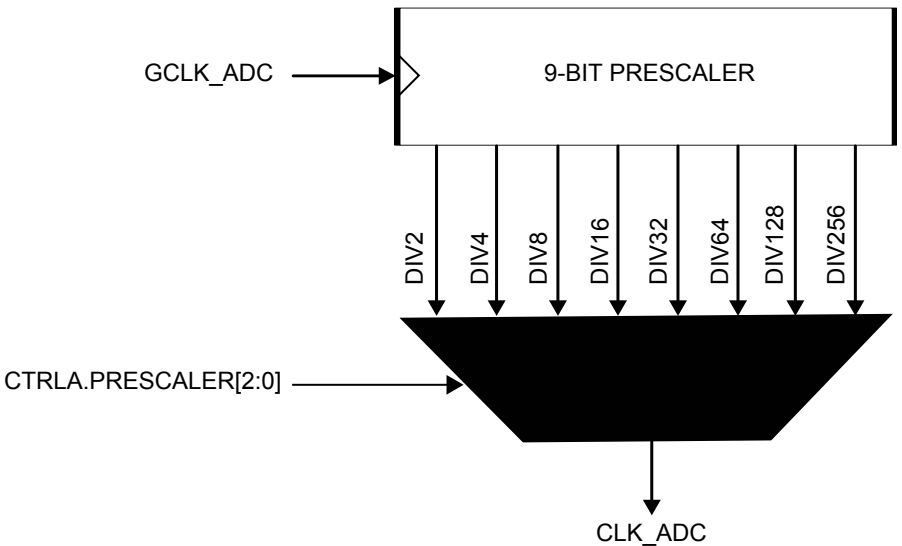
To avoid data loss if more than one channel is enabled, the conversion result must be read as soon as it is available (INTFLAG.RESRDY). Failing to do so will result in an overrun error condition, indicated by the OVERRUN bit in the Interrupt Flag Status and Clear register (INTFLAG.OVERRUN).

To enable one of the available interrupts sources, the corresponding bit in the Interrupt Enable Set register (INTENSET) must be written to '1'.

42.6.2.4. Prescaler Selection

The ADC is clocked by GCLK_ADC. There is also a prescaler in the ADC to enable conversion at lower clock rates. Refer to CTRLA for details on prescaler settings. Refer to Conversion Timing and Sampling Rate on page 1035 for details on timing and sampling rate.

Figure 42-2. ADC Prescaler



Note: The minimum prescaling factor is DIV2.

42.6.2.5. Reference Configuration

The ADC has various reference configuration options. By default, the internal bandgap voltage reference is selected. The REFSEL value in Reference Control register (REFCTRL) determines which reference will be selected. Based on customer application requirements, the external or internal reference can be selected. Two external references are available. The supply accepted on these pins is from 1.0V to VDD_{ANA}. Four internal inputs are also available. Refer to REFCTRL for further details on available selections.

42.6.2.6. ADC Resolution

The ADC supports 8-bit, 10-bit or 12-bit resolution. Resolution can be changed by writing the Resolution bit group in the Control C register (CTRLC.RESSEL). By default, the ADC resolution is set to 12 bits. The resolution affects the propagation delay, see also Conversion Timing and Sampling Rate on page 1035.

42.6.2.7. Differential and Single-Ended Conversions

The ADC has two conversion options: differential and single-ended:

If the positive input is always positive, the single-ended conversion should be used in order to have full 12-bit resolution in the conversion.

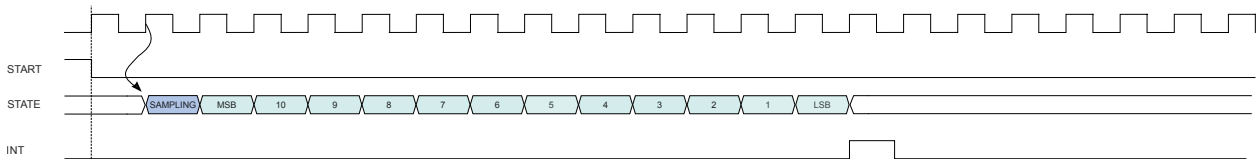
If the positive input may go below the negative input, the differential mode should be used in order to get correct results.

The differential mode is enabled by setting DIFFMODE bit in the Control C register (CTRLC.DIFFMODE). Both conversion types could be run in single mode or in free-running mode. When the free-running mode is selected, an ADC input will continuously sample the input and performs a new conversion. The INTFLAG.RESRDY bit will be set at the end of each conversion.

42.6.2.8. Conversion Timing and Sampling Rate

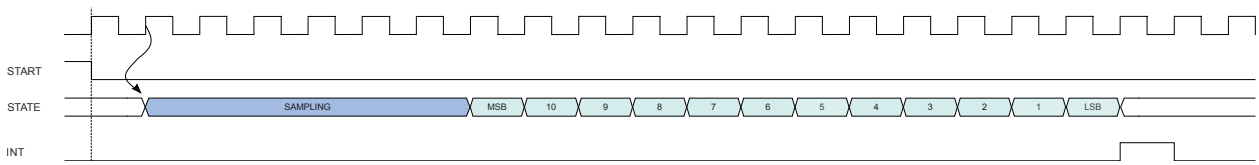
The following figure shows the ADC timing for one single conversion. A conversion starts after the software or event start are synchronized with the GCLK_ADC clock. The input channel is sampled in the first half CLK_ADC period.

Figure 42-3. ADC Timing for One Conversion in 12-bit Resolution



The sampling time can be increased by using the Sampling Time Length bit group in the Sampling Time Control register (SAMPCTRL.SAMPLEN). As example, the next figure is showing the timing conversion with sampling time increased to six CLK_ADC cycles.

Figure 42-4. ADC Timing for One Conversion with Increased Sampling Time, 12-bit



The ADC provides also offset compensation, see the following figure. The offset compensation is enabled by the Offset Compensation bit in the Sampling Control register (SAMPCTRL.OFFCOMP).

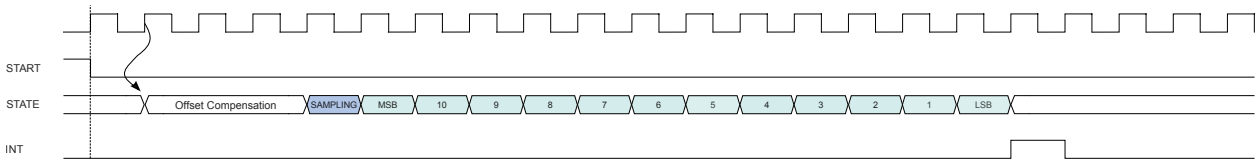
Note: If offset compensation is used, the sampling time must be set to one cycle of CLK_ADC.

In free running mode, the sampling rate R_S is calculated by

$R_S = f_{CLK\_ADC} / (n_{SAMPLING} + n_{OFFCOMP} + n_{DATA})$

Here, $n_{SAMPLING}$ is the sampling duration in CLK_ADC cycles, $n_{OFFCOMP}$ is the offset compensation duration in clock cycles, and n_{DATA} is the bit resolution. f_{CLK_ADC} is the ADC clock frequency from the internal prescaler: $f_{CLK\_ADC} = f_{GCLK\_ADC} / 2^{(1 + CTRLA.PRESCALER)}$

Figure 42-5. ADC Timing for One Conversion with Offset Compensation, 12-bit



The impact of resolution on the sampling rate is seen in the next two figures, where free-running sampling in 12-bit and 8-bit resolution are compared.

Figure 42-6. ADC Timing for Free Running in 12-bit Resolution

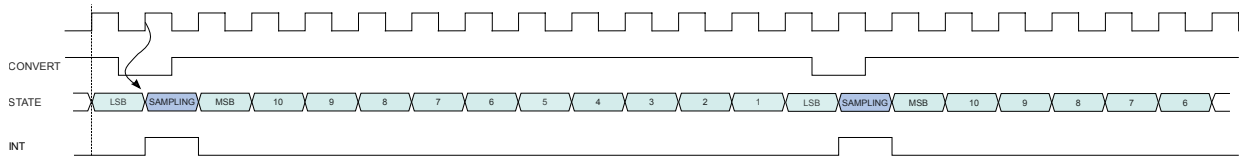
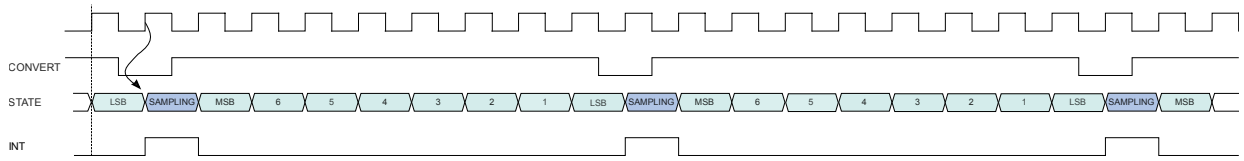


Figure 42-7. ADC Timing for Free Running in 8-bit Resolution



The propagation delay of an ADC measurement is given by:

$$\text{PropagationDelay} = \frac{1 + \text{Resolution}}{f_{\text{ADC}}}$$

> **Example.** In order to obtain 1MSPS in 12-bit resolution with a sampling time length of four CLK_ADC cycles, f_{CLK_ADC} must be 1MSPS * (4 + 12) = 16MHz. As the minimal division factor of the prescaler is 2, GCLK_ADC must be 32MHz.

42.6.2.9. Accumulation

The result from multiple consecutive conversions can be accumulated. The number of samples to be accumulated is specified by the Sample Number field in the Average Control register (AVGCTRL.SAMPLENUM). When accumulating more than 16 samples, the result will be too large to match the 16-bit RESULT register size. To avoid overflow, the result is right shifted automatically to fit within the available register size. The number of automatic right shifts is specified in the table below.

Note: To perform the accumulation of two or more samples, the Conversion Result Resolution field in the Control C register (CTRLC.RESSEL) must be set.

Table 42-1. Accumulation

Number of Accumulated Samples	AVGCTRL. SAMPLENUM	Number of Automatic Right Shifts	Final Result Precision	Automatic Division Factor
1	0x0	0	12 bits	0
2	0x1	0	13 bits	0
4	0x2	0	14 bits	0
8	0x3	0	15 bits	0
16	0x4	0	16 bits	0
32	0x5	1	16 bits	2
64	0x6	2	16 bits	4
128	0x7	3	16 bits	8
256	0x8	4	16 bits	16
512	0x9	5	16 bits	32
1024	0xA	6	16 bits	64
Reserved	0xB –0xF		12 bits	0

42.6.2.10. Averaging

Averaging is a feature that increases the sample accuracy, at the cost of a reduced sampling rate. This feature is suitable when operating in noisy conditions.

Averaging is done by accumulating m samples, as described in Accumulation on page 1036, and dividing the result by m. The averaged result is available in the RESULT register. The number of samples to be accumulated is specified by writing to AVGCTRL.SAMPLENUM as shown in Table 42-2 Averaging on page 1037.

The division is obtained by a combination of the automatic right shift described above, and an additional right shift that must be specified by writing to the Adjusting Result/Division Coefficient field in AVGCTRL (AVGCTRL.ADJRES), as described in Table 42-2 Averaging on page 1037.

Note: To perform the averaging of two or more samples, the Conversion Result Resolution field in the Control C register (CTRLC.RESSEL) must be set.

Averaging AVGCTRL.SAMPLENUM samples will reduce the un-averaged sampling rate by a factor $\frac{1}{\text{AVGCTRL.SAMPLENUM}}$.

When the averaged result is available, the INTFLAG.RESRDY bit will be set.

Table 42-2. Averaging

Number of Accumulated Samples	AVGCTRL. SAMPLENUM	Intermediate Result Precision	Number of Automatic Right Shifts	Division Factor	AVGCTRL. ADJRES	Total Number of Right Shifts	Final Result Precision	Automatic Division Factor
1	0x0	12 bits	0	1	0x0		12 bits	0
2	0x1	13	0	2	0x1	1	12 bits	0
4	0x2	14	0	4	0x2	2	12 bits	0

Number of Accumulated Samples	AVGCTRL. SAMPLENUM	Intermediate Result Precision	Number of Automatic Right Shifts	Division Factor	AVGCTRL. ADJRES	Total Number of Right Shifts	Final Result Precision	Automatic Division Factor
8	0x3	15	0	8	0x3	3	12 bits	0
16	0x4	16	0	16	0x4	4	12 bits	0
32	0x5	17	1	16	0x4	5	12 bits	2
64	0x6	18	2	16	0x4	6	12 bits	4
128	0x7	19	3	16	0x4	7	12 bits	8
256	0x8	20	4	16	0x4	8	12 bits	16
512	0x9	21	5	16	0x4	9	12 bits	32
1024	0xA	22	6	16	0x4	10	12 bits	64
Reserved	0xB –0xF				0x0		12 bits	0

42.6.2.11. Oversampling and Decimation

By using oversampling and decimation, the ADC resolution can be increased from 12 bits up to 16 bits, for the cost of reduced effective sampling rate.

To increase the resolution by n bits, 4n samples must be accumulated. The result must then be right-shifted by n bits. This right-shift is a combination of the automatic right-shift and the value written to AVGCTRL.ADJRES. To obtain the correct resolution, the ADJRES must be configured as described in the table below. This method will result in n bit extra LSB resolution.

Table 42-3. Configuration Required for Oversampling and Decimation

Result Resolution	Number of Samples to Average	AVGCTRL.SAMPLENUM[3:0]	Number of Automatic Right Shifts	AVGCTRL.ADJRES[2:0]
13 bits	$4^1 = 4$	0x2	0	0x1
14 bits	$4^2 = 16$	0x4	0	0x2
15 bits	$4^3 = 64$	0x6	2	0x1
16 bits	$4^4 = 256$	0x8	4	0x0

42.6.2.12. Automatic Sequences

The ADC has the ability to automatically sequence a series of conversions. This means that each time the ADC receives a start-of-conversion request, it can perform multiple conversions automatically. All of the 32 positive inputs can be included in a sequence by writing to corresponding bits in the Sequence Control register (SEQCTRL). The order of the conversion in a sequence is the lower positive MUX selection to upper positive MUX (AIN0, AIN1, AIN2 ...). In differential mode, the negative inputs selected by MUXNEG field, will be used for the entire sequence.

When a sequence starts, the Sequence Busy status bit in Sequence Status register (SEQSTATUS.SEQBUSY) will be set. When the sequence is complete, the Sequence Busy status bit will be cleared.

Each time a conversion is completed, the Sequence State bit in Sequence Status register (SEQSTATUS.SEQSTATE) will store the input number from which the conversion is done. The result will be stored in the RESULT register, and the Result Ready Interrupt Flag (INTFLAG.RESRDY) is set.

If additional inputs must be scanned, the ADC will automatically start a new conversion on the next input present in the sequence list.

Note that if SEQCTRL register has no bits set to '1', the conversion is done with the selected MUXPOS input.

42.6.2.13. Window Monitor

The window monitor feature allows the conversion result in the RESULT register to be compared to predefined threshold values. The window mode is selected by setting the Window Monitor Mode bits in the Control C register (CTRLC.WINMODE). Threshold values must be written in the Window Monitor Lower Threshold register (WINLT) and Window Monitor Upper Threshold register (WINUT).

If differential input is selected, the WINLT and WINUT are evaluated as signed values. Otherwise they are evaluated as unsigned values. The significant WINLT and WINUT bits are given by the precision selected in the Conversion Result Resolution bit group in the Control C register (CTRLC.RESSEL). This means that for example in 8-bit mode, only the eight lower bits will be considered. In addition, in differential mode, the eighth bit will be considered as the sign bit, even if the ninth bit is zero.

The INTFLAG.WINMON interrupt flag will be set if the conversion result matches the window monitor condition.

42.6.2.14. Offset and Gain Correction

Inherent gain and offset errors affect the absolute accuracy of the ADC.

The offset error is defined as the deviation of the actual ADC transfer function from an ideal straight line at zero input voltage. The offset error cancellation is handled by the Offset Correction register (OFFSETCORR). The offset correction value is subtracted from the converted data before writing the Result register (RESULT).

The gain error is defined as the deviation of the last output step's midpoint from the ideal straight line, after compensating for offset error. The gain error cancellation is handled by the Gain Correction register (GAINCORR).

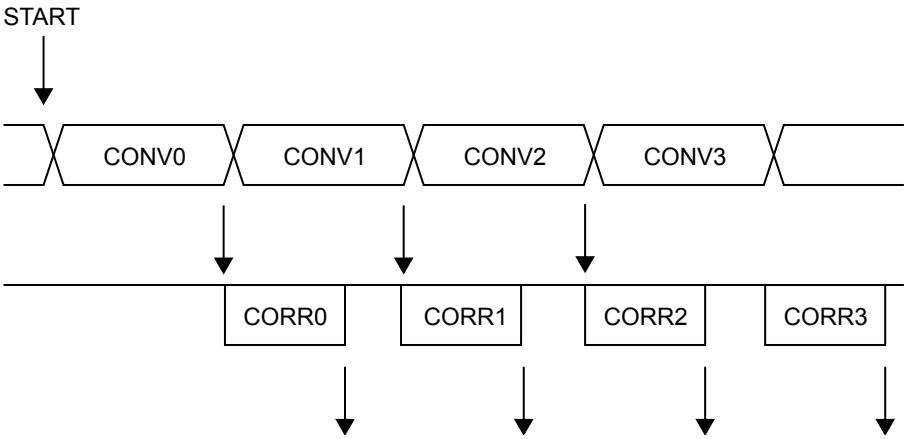
To correct these two errors, the Digital Correction Logic Enabled bit in the Control C register (CTRLC.CORREN) must be set.

Offset and gain error compensation results are both calculated according to:

$$\text{Result} = (\text{Conversion value} + - \text{OFFSETCORR}) \cdot \text{GAINCORR}$$

The correction will introduce a latency of 13 CLK_ADC clock cycles. In free running mode this latency is introduced on the first conversion only, since its duration is always less than the propagation delay. In single conversion mode this latency is introduced for each conversion.

Figure 42-8. ADC Timing Correction Enabled



42.6.3. Additional Features

42.6.4. DMA Operation

The ADC generates the following DMA request:

- Result Conversion Ready (RESRDY): the request is set when a conversion result is available and cleared when the RESULT register is read. When the averaging operation is enabled, the DMA request is set when the averaging is completed and result is available.

42.6.5. Interrupts

The ADC has the following interrupt sources:

- Result Conversion Ready: RESRDY
- Window Monitor: WINMON
- Overrun: OVERRUN

These interrupts are asynchronous wake-up sources. See *Sleep Mode Controller* for details.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. Each interrupt can be individually enabled by writing a one to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a one to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register. An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the ADC is reset. See INTFLAG on page 1053 for details on how to clear interrupt flags. All interrupt requests from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. Refer to *Nested Vector Interrupt Controller* for details. The user must read the INTFLAG register to determine which interrupt condition is present.

Note that interrupts must be globally enabled for interrupt requests to be generated. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

Nested Vector Interrupt Controller on page 51
Sleep Mode Controller on page 191

42.6.6. Events

The ADC can generate the following output events:

- Result Ready (RESRDY): Generated when the conversion is complete and the result is available. Refer to EVCTRL on page 1049 for details.
- Window Monitor (WINMON): Generated when the window monitor condition match. Refer to CTRLC on page 1057 for details.

Setting an Event Output bit in the Event Control Register (EVCTRL.xxEO=1) enables the corresponding output event. Clearing this bit disables the corresponding output event. Refer to the Event System chapter for details on configuring the event system.

The ADC can take the following actions on an input event:

- Start conversion (START): Start a conversion. Refer to SWTRIG on page 1065 for details.
- Conversion flush (FLUSH): Flush the conversion. Refer to SWTRIG on page 1065 for details.

Setting an Event Input bit in the Event Control register (EVCTRL.xxEI=1) enables the corresponding action on input event. Clearing this bit disables the corresponding action on input event.

The ADC uses only asynchronous events, so the asynchronous Event System channel path must be configured. By default, the ADC will detect a rising edge on the incoming event. If the ADC action must be performed on the falling edge of the incoming event, the event line must be inverted first. This is done by setting the corresponding Event Invert Enable bit in Event Control register (EVCTRL.xINV=1).

Note: If several events are connected to the ADC, the enabled action will be taken on any of the incoming events. If FLUSH and START events are available at the same time, the FLUSH event has priority.

Related Links

EVSYS – Event System on page 536

42.6.7. Sleep Mode Operation

The ONDEMAND and RUNSTDBY bits in the Control A register (CTRLA) control the behavior of the ADC during standby sleep mode, in cases where the ADC is enabled (CTRLA.ENABLE = 1). For further details on available options, refer to Table 42-4 ADC Sleep Behavior on page 1041.

Note: When CTRLA.ONDEMAND=1, the analog block is powered-off when the conversion is complete. When a start request is detected, the system returns from sleep and starts a new conversion after the start-up time delay.

Table 42-4. ADC Sleep Behavior

CTRLA.RUNSTDBY	CTRLA.ONDEMAND	CTRLA.ENABLE	Description
x	x	0	Disabled
0	0	1	Run in all sleep modes except STANDBY.
0	1	1	Run in all sleep modes on request, except STANDBY.
1	0	1	Run in all sleep modes.
1	1	1	Run in all sleep modes on request.

42.6.8. Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read. An exception is the Channel Enable bit in the Peripheral Channel Control registers (PCHCTRLm.CHEN). When changing this bit, the bit value must be read-back

to ensure the synchronization is complete and to assert glitch free internal operation. Note that changing the bit value under ongoing synchronization will *not* generate an error.

The following bits are synchronized when written:

- Software Reset bit in Control A register (CTRLA.SWRST)
- Enable bit in Control A register (CTRLA.ENABLE)

The following registers are synchronized when written:

- Input Control register (INPUTCTRL)
- Control C register (CTRLC)
- Average control register (AVGCTRL)
- Sampling time control register (SAMPCTRL)
- Window Monitor Lower Threshold register (WINLT)
- Window Monitor Upper Threshold register (WINUT)
- Gain correction register (GAINCORR)
- Offset Correction register (OFFSETCORR)
- Software Trigger register (SWTRIG)

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

Related Links

Register Synchronization on page 127

42.7. Register Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0	ONDEMAND	RUNSTDBY					ENABLE	SWRST
0x01	CTRLB	7:0						PRESCALER[2:0]		
0x02	REFCTRL	7:0	REFCOMP				REFSEL[3:0]			
0x03	EVCTRL	7:0			WINMONEO	RESRDYEO	STARTINV	FLUSHINV	STARTEI	FLUSHEI
0x04	INTENCLR	7:0						WINMON	OVERRUN	RESRDY
0x05	INTENSET	7:0						WINMON	OVERRUN	RESRDY
0x06	INTFLAG	7:0						WINMON	OVERRUN	RESRDY
0x07	SEQSTATUS	7:0	SEQBUSY			SEQSTATE[4:0]				
0x08	INPUTCTRL	7:0				MUXPOS[4:0]				
0x09		15:8				MUXNEG[4:0]				
0x0A	CTRLC	7:0			RESSEL[1:0]		CORREN	FREERUN	LEFTADJ	DIFFMODE
0x0B		15:8						WINMODE[2:0]		
0x0C	AVGCTRL	7:0		ADJRES[2:0]			SAMPLENUM[3:0]			
0x0D	SAMPCTRL	7:0	OFFCOMP		SAMPLEN[5:0]					
0x0E	WINLT	7:0	WINLT[7:0]							
0x0F		15:8	WINLT[15:8]							
0x10	WINUT	7:0	WINUT[7:0]							
0x11		15:8	WINUT[15:8]							
0x12	GAINCORR	7:0	GAINCORR[7:0]							
0x13		15:8					GAINCORR[11:8]			
0x14	OFFSETCORR	7:0	OFFSETCORR[7:0]							
0x15		15:8					OFFSETCORR[11:8]			
0x16 ... 0x17	Reserved									
0x18	SWTRIG	7:0							START	FLUSH
0x19 ... 0x1B	Reserved									
0x1C	DBGCTRL	7:0								DBGRUN
0x1D ... 0x1F	Reserved									
0x20	SYNCBUSY	7:0	WINUT	WINLT	SAMPCTRL	AVGCTRL	CTRLC	INPUTCTRL	ENABLE	SWRST
0x21		15:8						SWTRIG	OFFSETCOR R	GAINCORR
0x22 ... 0x23	Reserved									
0x24	RESULT	7:0	RESULT[7:0]							
0x25		15:8	RESULT[15:8]							
0x26 ... 0x27	Reserved									

Offset	Name	Bit Pos.								
0x28	SEQCTRL	7:0	SEQEN7	SEQEN6	SEQEN5	SEQEN4	SEQEN3	SEQEN2	SEQEN1	SEQEN0
0x29		15:8	SEQEN15	SEQEN14	SEQEN13	SEQEN12	SEQEN11	SEQEN10	SEQEN9	SEQEN8
0x2A		23:16	SEQEN23	SEQEN22	SEQEN21	SEQEN20	SEQEN19	SEQEN18	SEQEN17	SEQEN16
0x2B		31:24	SEQEN31	SEQEN30	SEQEN29	SEQEN28	SEQEN27	SEQEN26	SEQEN25	SEQEN24
0x2C	CALIB	7:0						BIASCOMP[2:0]		
0x2D		15:8						BIASREFBUF[2:0]		

42.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection on page 1033.

Some registers are synchronized when read and/or written. Synchronization is denoted by the "Write-Synchronized" or the "Read-Synchronized" property in each individual register description. For details, refer to Synchronization on page 1041.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

42.8.1. Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY					ENABLE	SWRST
Access	R/W	R/W					R/W	R/W
Reset	0	0					0	0

Bit 7 – ONDEMAND: On Demand Control
The On Demand operation mode allows the ADC to be enabled or disabled, depending on other peripheral requests.

In On Demand operation mode, i.e., if the ONDEMAND bit has been previously set, the ADC will only be running when requested by a peripheral. If there is no peripheral requesting the ADC will be in a disable state.

If On Demand is disabled the ADC will always be running when enabled.

In standby sleep mode, the On Demand operation is still active if the CTRLA.RUNSTDBY bit is '1'. If CTRLA.RUNSTDBY is '0', the ADC is disabled.

This bit is not synchronized.

Value	Description
0	The ADC is always on , if enabled.
1	The ADC is enabled, when a peripheral is requesting the ADC conversion. The ADC is disabled if no peripheral is requesting it.

Bit 6 – RUNSTDBY: Run in Standby
This bit controls how the ADC behaves during standby sleep mode.

This bit is not synchronized.

Value	Description
0	The ADC is halted during standby sleep mode.
1	The ADC is not stopped in standby sleep mode. If CTRLA.ONDEMAND=1, the ADC will be running when a peripheral is requesting it. If CTRLA.ONDEMAND=0, the ADC will always be running in standby sleep mode.

Bit 1 – ENABLE: Enable
Due to synchronization there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRL.ENABLE will read back immediately and the ENABLE bit in the SYNCBUSY register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

Value	Description
0	The ADC is disabled.
1	The ADC is enabled.

Bit 0 – SWRST: Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the ADC, except DBGCTRL, to their initial state, and the ADC will be disabled.

Writing a '1' to CTRL.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.

Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

42.8.2. Control B

Name: CTRLB
Offset: 0x01
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
						PRESCALER[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 2:0 – PRESCALER[2:0]: Prescaler Configuration
This field defines the ADC clock relative to the peripheral clock.

Value	Name	Description
0x0	DIV2	Peripheral clock divided by 2
0x1	DIV4	Peripheral clock divided by 4
0x2	DIV8	Peripheral clock divided by 8
0x3	DIV16	Peripheral clock divided by 16
0x4	DIV32	Peripheral clock divided by 32
0x5	DIV64	Peripheral clock divided by 64
0x6	DIV128	Peripheral clock divided by 128
0x7	DIV256	Peripheral clock divided by 256

42.8.3. Reference Control

Name: REFCTRL
Offset: 0x02
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
	REFCOMP				REFSEL[3:0]			
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit 7 – REFCOMP: Reference Buffer Offset Compensation Enable
The gain error can be reduced by enabling the reference buffer offset compensation. This will decrease the input impedance and thus increase the start-up time of the reference.

Value	Description
0	Reference buffer offset compensation is disabled.
1	Reference buffer offset compensation is enabled.

Bits 3:0 – REFSEL[3:0]: Reference Selection
These bits select the reference for the ADC.

Value	Name	Description
0x0	INTREF	internal bandgap reference
x01	INTVCC0	1/1.6 VDDANA
0x2	INTVCC1	1/2 VDDANA (only for VDDANA > 2.0V)
0x3	VREFA	External reference
0x4	VREFB	External reference
0x5	INTVCC2	VDDANA
0x6 - 0xF		Reserved

42.8.4. Event Control

Name: EVCTRL
Offset: 0x03
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
			WINMONEO	RESRDYEO	STARTINV	FLUSHINV	STARTEI	FLUSHEI
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 5 – WINMONEO: Window Monitor Event Out
This bit indicates whether the Window Monitor event output is enabled or not and an output event will be generated when the window monitor detects something.

Value	Description
0	Window Monitor event output is disabled and an event will not be generated.
1	Window Monitor event output is enabled and an event will be generated.

Bit 4 – RESRDYEO: Result Ready Event Out
This bit indicates whether the Result Ready event output is enabled or not and an output event will be generated when the conversion result is available.

Value	Description
0	Result Ready event output is disabled and an event will not be generated.
1	Result Ready event output is enabled and an event will be generated.

Bit 3 – STARTINV: Start Conversion Event Invert Enable

Value	Description
0	Start event input source is not inverted.
1	Start event input source is inverted.

Bit 2 – FLUSHINV: Flush Event Invert Enable

Value	Description
0	Flush event input source is not inverted.
1	Flush event input source is inverted.

Bit 1 – STARTEI: Start Conversion Event Input Enable

Value	Description
0	A new conversion will not be triggered on any incoming event.
1	A new conversion will be triggered on any incoming event.

Bit 0 – FLUSHEI: Flush Event Input Enable

Value	Description
0	A flush and new conversion will not be triggered on any incoming event.
1	A flush and new conversion will be triggered on any incoming event.

42.8.5. Interrupt Enable Clear

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Name: INTENCLR
Offset: 0x04
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
						WINMON	OVERRUN	RESRDY
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – WINMON: Window Monitor Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Window Monitor Interrupt Enable bit, which disables the corresponding interrupt request.

Value	Description
0	The window monitor interrupt is disabled.
1	The window monitor interrupt is enabled, and an interrupt request will be generated when the Window Monitor interrupt flag is set.

Bit 1 – OVERRUN: Overrun Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Overrun Interrupt Enable bit, which disables the corresponding interrupt request.

Value	Description
0	The Overrun interrupt is disabled.
1	The Overrun interrupt is enabled, and an interrupt request will be generated when the Overrun interrupt flag is set.

Bit 0 – RESRDY: Result Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Result Ready Interrupt Enable bit, which disables the corresponding interrupt request.

Value	Description
0	The Result Ready interrupt is disabled.
1	The Result Ready interrupt is enabled, and an interrupt request will be generated when the Result Ready interrupt flag is set.

42.8.6. Interrupt Enable Set

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Name: INTENSET
Offset: 0x05
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
						WINMON	OVERRUN	RESRDY
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – WINMON: Window Monitor Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Window Monitor Interrupt bit, which enables the Window Monitor interrupt.

Value	Description
0	The Window Monitor interrupt is disabled.
1	The Window Monitor interrupt is enabled.

Bit 1 – OVERRUN: Overrun Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Overrun Interrupt bit, which enables the Overrun interrupt.

Value	Description
0	The Overrun interrupt is disabled.
1	The Overrun interrupt is enabled.

Bit 0 – RESRDY: Result Ready Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Result Ready Interrupt bit, which enables the Result Ready interrupt.

Value	Description
0	The Result Ready interrupt is disabled.
1	The Result Ready interrupt is enabled.

42.8.7. Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x06
Reset: 0x00
Property: –

Bit	7	6	5	4	3	2	1	0
						WINMON	OVERRUN	RESRDY
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – WINMON: Window Monitor
This flag is cleared by writing a '1' to the flag or by reading the RESULT register.

This flag is set on the next GCLK_ADC cycle after a match with the window monitor condition, and an interrupt request will be generated if INTENCLR/SET.WINMON is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Window Monitor interrupt flag.

Bit 1 – OVERRUN: Overrun
This flag is cleared by writing a '1' to the flag.

This flag is set if RESULT is written before the previous value has been read by CPU, and an interrupt request will be generated if INTENCLR/SET.OVERRUN=1.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Overrun interrupt flag.

Bit 0 – RESRDY: Result Ready
This flag is cleared by writing a '1' to the flag or by reading the RESULT register.

This flag is set when the conversion result is available, and an interrupt will be generated if INTENCLR/SET.RESRDY=1.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Result Ready interrupt flag.

42.8.8. Sequence Status

Name: SEQSTATUS
Offset: 0x07
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	SEQBUSY			SEQSTATE[4:0]				
Access	R			R	R	R	R	R
Reset	0			0	0	0	0	0

Bit 7 – SEQBUSY: Sequence busy
This bit is set when the sequence start.

This bit is clear when the last conversion in a sequence is done.

Bits 4:0 – SEQSTATE[4:0]: Sequence State
These bit fields are the pointer of sequence. This value identifies the last conversion done in the sequence.

42.8.9. Input Control

Name: INPUTCTRL
Offset: 0x08
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
				MUXNEG[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
				MUXPOS[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bits 12:8 – MUXNEG[4:0]: Negative MUX Input Selection
These bits define the MUX selection for the negative ADC input.

Value	Name	Description
0x00	AIN0	ADC AIN0 pin
0x01	AIN1	ADC AIN1 pin
0x02	AIN2	ADC AIN2 pin
0x03	AIN3	ADC AIN3 pin
0x04	AIN4	ADC AIN4 pin
0x05	AIN5	ADC AIN5 pin
0x18	GND	Internal ground
0x19 - 0x1F	-	Reserved

Bits 4:0 – MUXPOS[4:0]: Positive MUX Input Selection
These bits define the MUX selection for the positive ADC input. If the internal bandgap voltage or temperature sensor input channel is selected, then the Sampling Time Length bit group in the Sampling Control register must be written with a corresponding value.

Value	Name	Description
0x00	AIN0	ADC AIN0 pin
0x01	AIN1	ADC AIN1 pin
0x02	AIN2	ADC AIN2 pin
0x03	AIN3	ADC AIN3 pin
0x04	AIN4	ADC AIN4 pin
0x05	AIN5	ADC AIN5 pin

Value	Name	Description
0x06	AIN6	ADC AIN6 pin
0x07	AIN7	ADC AIN7 pin
0x08	AIN8	ADC AIN8 pin
0x09	AIN9	ADC AIN9 pin
0x0A	AIN10	ADC AIN10 pin
0x0B	AIN11	ADC AIN11 pin
0x0C	AIN12	ADC AIN12 pin
0x0D	AIN13	ADC AIN13 pin
0x0E	AIN14	ADC AIN14 pin
0x0F	AIN15	ADC AIN15 pin
0x10	AIN16	ADC AIN16 pin
0x11	AIN17	ADC AIN17 pin
0x12	AIN18	ADC AIN18 pin
0x13	AIN19	ADC AIN19 pin
0x14 - 0x17	-	Reserved
0x18	TEMP	Temperature Sensor
0x19	BANDGAP	Bandgap Voltage
0x1A	SCALEDCOREVCC	1/4 Scaled Core Supply
0x1B	SCALEDIOVCC	1/4 Scaled I/O Supply

42.8.10. Control C

Name: CTRLC
Offset: 0x0A
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
						WINMODE[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
			RESSEL[1:0]		CORREN	FREERUN	LEFTADJ	DIFFMODE
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 10:8 – WINMODE[2:0]: Window Monitor Mode
These bits enable and define the window monitor mode.

Value	Name	Description
0x0	DISABLE	No window mode (default)
0x1	MODE1	RESULT > WINLT
0x2	MODE2	RESULT < WINUT
0x3	MODE3	WINLT < RESULT < WINUT
0x4	MODE4	WINUT < RESULT < WINLT
0x5 - 0x7		Reserved

Bits 5:4 – RESSEL[1:0]: Conversion Result Resolution
These bits define whether the ADC completes the conversion 12-, 10- or 8-bit result resolution.

Value	Name	Description
0x0	12BIT	12-bit result
0x1	16BIT	For averaging mode output
0x2	10BIT	10-bit result
0x3	8BIT	8-bit result

Bit 3 – CORREN: Digital Correction Logic Enabled

Value	Description
0	Disable the digital result correction.
1	Enable the digital result correction. The ADC conversion result in the RESULT register is then corrected for gain and offset based on the values in the GAINCORR and OFFSETCORR registers. Conversion time will be increased by 13 cycles according to the value in the Offset Correction Value bit group in the Offset Correction register.

Bit 2 – FREERUN: Free Running Mode

Value	Description
0	The ADC run in single conversion mode.
1	The ADC is in free running mode and a new conversion will be initiated when a previous conversion completes.

Bit 1 – LEFTADJ: Left-Adjusted Result

Value	Description
0	The ADC conversion result is right-adjusted in the RESULT register.
1	The ADC conversion result is left-adjusted in the RESULT register. The high byte of the 12-bit result will be present in the upper part of the result register. Writing this bit to zero (default) will right-adjust the value in the RESULT register.

Bit 0 – DIFFMODE: Differential Mode

Value	Description
0	The ADC is running in singled-ended mode.
1	The ADC is running in differential mode. In this mode, the voltage difference between the MUXPOS and MUXNEG inputs will be converted by the ADC.

42.8.11. Average Control

Name: AVGCTRL
Offset: 0x0C
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
		ADJRES[2:0]			SAMPLENUM[3:0]			
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 6:4 – ADJRES[2:0]: Adjusting Result / Division Coefficient
These bits define the division coefficient in 2n steps.

Bits 3:0 – SAMPLENUM[3:0]: Number of Samples to be Collected
These bits define how many samples are added together. The result will be available in the Result register (RESULT). Note: if the result width increases, CTRLC.RESSEL must be changed.

Value	Description
0x0	1 sample
0x1	2 samples
0x2	4 samples
0x3	8 samples
0x4	16 samples
0x5	32 samples
0x6	64 samples
0x7	128 samples
0x8	256 samples
0x9	512 samples
0xA	1024 samples
0xB - 0xF	Reserved

42.8.12. Sampling Time Control

Name: SAMPCTRL
Offset: 0x0D
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	OFFCOMP		SAMPLEN[5:0]					
Access	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset	0		0	0	0	0	0	0

Bit 7 – OFFCOMP: Comparator Offset Compensation Enable
Setting this bit enables the offset compensation for each sampling period to ensure low offset and immunity to temperature or voltage drift. This compensation increases the sampling time by three clock cycles.

This bit must be set to zero to validate the SAMPLEN value. It's not possible to use OFFCOMP=1 and SAMPLEN>0.

Bits 5:0 – SAMPLEN[5:0]: Sampling Time Length
These bits control the ADC sampling time in number of CLK_ADC cycles, depending of the prescaler value, thus controlling the ADC input impedance. Sampling time is set according to the equation:

$$\text{Sampling time} = (\text{SAMPLEN}+1) \cdot \left(\text{CLK}_{\text{ADC}}\right)$$

42.8.13. Window Monitor Lower Threshold

Name: WINLT
Offset: 0x0E
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	WINLT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	WINLT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – WINLT[15:0]: Window Lower Threshold
If the window monitor is enabled, these bits define the lower threshold value.

42.8.14. Window Monitor Upper Threshold

Name: WINUT
Offset: 0x10
Reset: 0x0000
Property: PAV Write-Protection, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	WINUT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	WINUT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – WINUT[15:0]: Window Upper Threshold
If the window monitor is enabled, these bits define the upper threshold value.

42.8.15. Gain Correction

Name: GAINCORR
Offset: 0x12
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
					GAINCORR[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	GAINCORR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 11:0 – GAINCORR[11:0]: Gain Correction Value
If CTRLC.CORREN=1, these bits define how the ADC conversion result is compensated for gain error before being written to the result register. The gain correction is a fractional value, a 1-bit integer plus an 11-bit fraction, and therefore ½ <= GAINCORR < 2. GAINCORR values range from 0.10000000000 to 1.11111111111.

42.8.16. Offset Correction

Name: OFFSETCORR
Offset: 0x14
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
					OFFSETCORR[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	OFFSETCORR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 11:0 – OFFSETCORR[11:0]: Offset Correction Value
If CTRLC.CORREN=1, these bits define how the ADC conversion result is compensated for offset error before being written to the Result register. This OFFSETCORR value is in two's complement format.

42.8.17. Software Trigger

Name: SWTRIG
Offset: 0x18
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
							START	FLUSH
Access							W	W
Reset							0	0

Bit 1 – START: ADC Start Conversion
Writing a '1' to this bit will start a conversion or sequence. The bit is cleared by hardware when the conversion has started. Writing a '1' to this bit when it is already set has no effect.

Writing a '0' to this bit will have no effect.

Bit 0 – FLUSH: ADC Conversion Flush
Writing a '1' to this bit will flush the ADC pipeline. A flush will restart the ADC clock on the next peripheral clock edge, and all conversions in progress will be aborted and lost. This bit is cleared until the ADC has been flushed.

After the flush, the ADC will resume where it left off; i.e., if a conversion was pending, the ADC will start a new conversion.

Writing this bit to '0' will have no effect.

42.8.18. Debug Control

Name: DBGCTRL
Offset: 0x1C
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGRUN
Access								R/W
Reset								0

Bit 0 – DBGRUN: Debug Run
This bit is not reset by a software reset.

This bit controls the functionality when the CPU is halted by an external debugger.

This bit should be written only while a conversion is not ongoing.

Value	Description
0	The ADC is halted when the CPU is halted by an external debugger.
1	The ADC continues normal operation when the CPU is halted by an external debugger.

42.8.19. Synchronization Busy

Name: SYNCBUSY
Offset: 0x20
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
						SWTRIG	OFFSETCORR	GAINCORR
Access						R	R	R
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	WINUT	WINLT	SAMPCTRL	AVGCTRL	CTRLC	INPUTCTRL	ENABLE	SWRST
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 10 – SWTRIG: Software Trigger Synchronization Busy
This bit is cleared when the synchronization of SWTRIG register between the clock domains is complete.

This bit is set when the synchronization of SWTRIG register between clock domains is started.

Bit 9 – OFFSETCORR: Offset Correction Synchronization Busy
This bit is cleared when the synchronization of OFFSETCORR register between the clock domains is complete.

This bit is set when the synchronization of OFFSETCORR register between clock domains is started.

Bit 8 – GAINCORR: Gain Correction Synchronization Busy
This bit is cleared when the synchronization of GAINCORR register between the clock domains is complete.

This bit is set when the synchronization of GAINCORR register between clock domains is started.

Bit 7 – WINUT: Window Monitor Lower Threshold Synchronization Busy
This bit is cleared when the synchronization of WINUT register between the clock domains is complete.

This bit is set when the synchronization of WINUT register between clock domains is started.

Bit 6 – WINLT: Window Monitor Upper Threshold Synchronization Busy
This bit is cleared when the synchronization of WINLT register between the clock domains is complete.

This bit is set when the synchronization of WINLT register between clock domains is started.

Bit 5 – SAMPCTRL: Sampling Time Control Synchronization Busy
This bit is cleared when the synchronization of SAMPCTRL register between the clock domains is complete.

This bit is set when the synchronization of SAMPCTRL register between clock domains is started.

Bit 4 – AVGCTRL: Average Control Synchronization Busy
This bit is cleared when the synchronization of AVGCTRL register between the clock domains is complete.

This bit is set when the synchronization of AVGCTRL register between clock domains is started.

Bit 3 – CTRLC: Control C Synchronization Busy
This bit is cleared when the synchronization of CTRLC register between the clock domains is complete.

This bit is set when the synchronization of CTRLC register between clock domains is started.

Bit 2 – INPUTCTRL: Input Control Synchronization Busy
This bit is cleared when the synchronization of INPUTCTRL register between the clock domains is complete.

This bit is set when the synchronization of INPUTCTRL register between clock domains is started.

Bit 1 – ENABLE: ENABLE Synchronization Busy
This bit is cleared when the synchronization of ENABLE register between the clock domains is complete.

This bit is set when the synchronization of ENABLE register between clock domains is started.

Bit 0 – SWRST: SWRST Synchronization Busy
This bit is cleared when the synchronization of SWRST register between the clock domains is complete.

This bit is set when the synchronization of SWRST register between clock domains is started

42.8.20. Result

Name: RESULT
Offset: 0x24
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
	RESULT[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RESULT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – RESULT[15:0]: Result Conversion Value
These bits will hold up to a 16-bit ADC conversion result, depending on the configuration.

In single conversion mode without averaging, the ADC conversion will produce a 12-bit result, which can be left- or right-shifted, depending on the setting of CTRLC.LEFTADJ.

If the result is left-adjusted (CTRLC.LEFTADJ), the high byte of the result will be in bit position [15:8], while the remaining 4 bits of the result will be placed in bit locations [7:4]. This can be used only if an 8-bit result is needed; i.e., one can read only the high byte of the entire 16-bit register.

If the result is not left-adjusted (CTRLC.LEFTADJ) and no oversampling is used, the result will be available in bit locations [11:0], and the result is then 12 bits long. If oversampling is used, the result will be located in bit locations [15:0], depending on the settings of the Average Control register.

42.8.21. Sequence Control

Name: SEQCTRL
Offset: 0x28
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	SEQEN31	SEQEN30	SEQEN29	SEQEN28	SEQEN27	SEQEN26	SEQEN25	SEQEN24
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	SEQEN23	SEQEN22	SEQEN21	SEQEN20	SEQEN19	SEQEN18	SEQEN17	SEQEN16
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SEQEN15	SEQEN14	SEQEN13	SEQEN12	SEQEN11	SEQEN10	SEQEN9	SEQEN8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SEQEN7	SEQEN6	SEQEN5	SEQEN4	SEQEN3	SEQEN2	SEQEN1	SEQEN0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – SEQENn: Enable Positive Input in the Sequence
For details on available positive mux selection, refer to INPUTCTRL.MUXENG.

The sequence start from the lowest input, and go to the next enabled input automatically when the conversion is done. If no bits are set the sequence is disabled.

Value	Description
0	Disable the positive input mux n selection from the sequence.
1	Enable the positive input mux n selection to the sequence.

42.8.22. Calibration

Name: CALIB
Offset: 0x2C
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
						BIASREFBUF[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
						BIASCOMP[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 10:8 – BIASREFBUF[2:0]: Bias Reference Buffer Scaling
This value from production test must be loaded from the NVM software calibration row into the CALIB register by software to achieve the specified accuracy.

The value must be copied only, and must not be changed.

Bits 2:0 – BIASCOMP[2:0]: Bias Comparator Scaling
This value from production test must be loaded from the NVM software calibration row into the CALIB register by software to achieve the specified accuracy.

The value must be copied only, and must not be changed

43. AC – Analog Comparators

43.1. Overview

The Analog Comparator (AC) supports two individual comparators. Each comparator (COMP) compares the voltage levels on two inputs, and provides a digital output based on this comparison. Each comparator may be configured to generate interrupt requests and/or peripheral events upon several different combinations of input change.

Hysteresis and propagation delay are two important properties of the comparators' dynamic behavior. Both parameters may be adjusted to achieve the optimal operation for each application.

The input selection includes four shared analog port pins and several internal signals. Each comparator output state can also be output on a pin for use by external devices.

The comparators are always grouped in pairs on each port. The AC peripheral implements one pair of comparators. These are called Comparator 0 (COMP0) and Comparator 1 (COMP1). They have identical behaviors, but separate control registers. The pair can be set in window mode to compare a signal to a voltage range instead of a single voltage level.

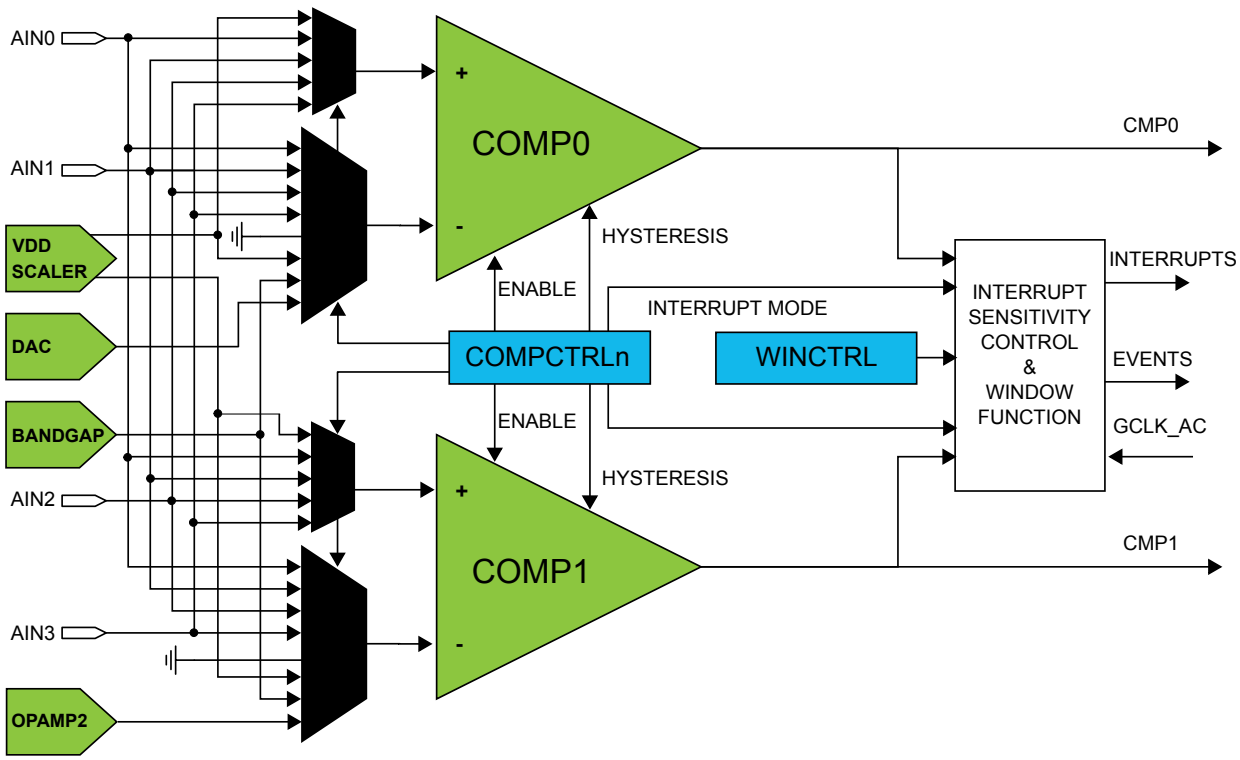
43.2. Features

- Two individual comparators
- Selectable propagation delay versus current consumption
- Selectable hysteresis
  - 4-levels or Off
- Analog comparator outputs available on pins
  - Asynchronous or synchronous
- Flexible input selection:
  - Four pins selectable for positive or negative inputs
  - Ground (for zero crossing)
  - Bandgap reference voltage
  - 64-level programmable VDD scaler per comparator
  - DAC
  - OPAMP2
- Interrupt generation on:
  - Rising or falling edge
  - Toggle
  - End of comparison
- Window function interrupt generation on:
  - Signal above window
  - Signal inside window
  - Signal below window
  - Signal outside window
- Event generation on:
  - Comparator output

- Window function inside/outside window
- Optional digital filter on comparator output
- Low-power option
  - Single-shot support

43.3. Block Diagram

Figure 43-1. Analog Comparator Block Diagram



43.4. Signal Description

Signal	Description	Type
AIN[3..0]	Analog input	Comparator inputs
CMP[1..0]	Digital output	Comparator outputs

Refer to *I/O Multiplexing and Considerations* for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

Related Links

I/O Multiplexing and Considerations on page 29

43.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

43.5.1. I/O Lines

Using the AC's I/O lines requires the I/O pins to be configured. Refer to *PORT - I/O Pin Controller* for details.

Related Links

PORT: IO Pin Controller on page 506

43.5.2. Power Management

The AC will continue to operate in any sleep mode where the selected source clock is running. The AC's interrupts can be used to wake up the device from sleep modes. Events connected to the event system can trigger other operations in the system without exiting sleep modes.

Related Links

PM – Power Manager on page 186

43.5.3. Clocks

The AC bus clock (CLK_AC_APB) can be enabled and disabled in the Power Manager, and the default state of CLK_AC_APB can be found in the Peripheral Clock Masking section in the Power Manager description.

A generic clock (GCLK_AC) is required to clock the AC. This clock must be configured and enabled in the generic clock controller before using the AC. Refer to the Generic Clock Controller chapter for details.

This generic clock is asynchronous to the bus clock (CLK_AC_APB). Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains. Refer to Synchronization for further details.

Related Links

PM – Power Manager on page 186

43.5.4. DMA

Not applicable.

43.5.5. Interrupts

The interrupt request lines are connected to the interrupt controller. Using the AC interrupts requires the interrupt controller to be configured first. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

Nested Vector Interrupt Controller on page 51

43.5.6. Events

The events are connected to the Event System. Refer to *EVSYS – Event System* for details on how to configure the Event System.

Related Links

EVSYS – Event System on page 536

43.5.7. Debug Operation

When the CPU is halted in debug mode, the AC will halt normal operation after any on-going comparison is completed. The AC can be forced to continue normal operation during debugging. Refer to DBGCTRL for details. If the AC is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

43.5.8. Register Access Protection

All registers with write-access can be write-protected optionally by the Peripheral Access Controller (PAC), except the following registers:

- Control B register (CTRLB)
- Interrupt Flag register (INTFLAG)

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

PAC write-protection does not apply to accesses through an external debugger.

Related Links

PAC - Peripheral Access Controller on page 58

43.5.9. Analog Connections

Each comparator has up to four I/O pins that can be used as analog inputs. Each pair of comparators shares the same four pins. These pins must be configured for analog operation before using them as comparator inputs.

Any internal reference source, such as a bandgap voltage reference, OPAMP2,or DAC must be configured and enabled prior to its use as a comparator input.

The analog signals of AC, ADC, DAC and OPAMP can be interconnected. The AC and ADC peripheral can request the OPAMP using an analog ONDEMAND functionality.

See *Analog Connections of Peripherals* for details.

Related Links

Analog Connections of Peripherals on page 34

43.6. Functional Description

43.6.1. Principle of Operation

Each comparator has one positive input and one negative input. Each positive input may be chosen from a selection of analog input pins. Each negative input may be chosen from a selection of both analog input pins and internal inputs, such as a bandgap voltage reference.

The digital output from the comparator is '1' when the difference between the positive and the negative input voltage is positive, and '0' otherwise.

The individual comparators can be used independently (normal mode) or paired to form a window comparison (window mode).

43.6.2. Basic Operation

43.6.2.1. Initialization

Some registers are enable-protected, meaning they can only be written when the module is disabled.

The following register is enable-protected:

- Event Control register (EVCTRL)

Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

43.6.2.2. Enabling, Disabling and Resetting

The AC is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The AC is disabled writing a '0' to CTRLA.ENABLE.

The AC is reset by writing a '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the AC will be reset to their initial state, and the AC will be disabled. Refer to *CTRLA* for details.

43.6.2.3. Comparator Configuration

Each individual comparator must be configured by its respective Comparator Control register (COMPCTRLx) before that comparator is enabled. These settings cannot be changed while the comparator is enabled.

- Select the desired measurement mode with COMPCTRLx.SINGLE. See Starting a Comparison for more details.
- Select the desired hysteresis with COMPCTRLx.HYSTEN and COMPCTRLx.HYST. See Input Hysteresis for more details.
- Select the comparator speed versus power with COMPCTRLx.SPEED. See Propagation Delay vs. Power Consumption for more details.
- Select the interrupt source with COMPCTRLx.INTSEL.
- Select the positive and negative input sources with the COMPCTRLx.MUXPOS and COMPCTRLx.MUXNEG bits. See Selecting Comparator Inputs for more details.
- Select the filtering option with COMPCTRLx.FLEN.
- Select standby operation with Run in Standby bit (COMPCTRLx.RUNSTDBY).

The individual comparators are enabled by writing a '1' to the Enable bit in the Comparator x Control registers (COMPCTRLx.ENABLE). The individual comparators are disabled by writing a '0' to COMPCTRLx.ENABLE. Writing a '0' to CTRLA.ENABLE will also disable all the comparators, but will not clear their COMPCTRLx.ENABLE bits.

43.6.2.4. Starting a Comparison

Each comparator channel can be in one of two different measurement modes, determined by the Single bit in the Comparator x Control register (COMPCTRLx.SINGLE):

- Continuous measurement
- Single-shot

After being enabled, a start-up delay is required before the result of the comparison is ready. This start-up time is measured automatically to account for environmental changes, such as temperature or voltage supply level, and is specified in *Electrical Characteristics*. During the start-up time, the COMP output is not available.

The comparator can be configured to generate interrupts when the output toggles, when the output changes from '0' to '1' (rising edge), when the output changes from '1' to '0' (falling edge) or at the end of the comparison. An end-of-comparison interrupt can be used with the single-shot mode to chain further events in the system, regardless of the state of the comparator outputs. The interrupt mode is set by the Interrupt Selection bit group in the Comparator Control register (COMPCTRLx.INTSEL). Events are generated using the comparator output state, regardless of whether the interrupt is enabled or not.

Related Links

Electrical Characteristics on page 1140

Continuous Measurement

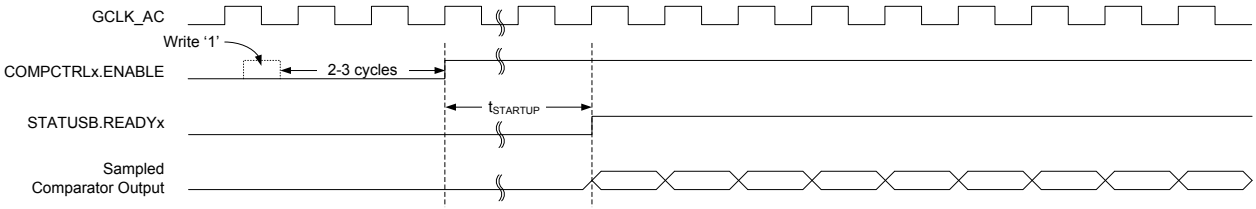
Continuous measurement is selected by writing COMPCTRLx.SINGLE to zero. In continuous mode, the comparator is continuously enabled and performing comparisons. This ensures that the result of the latest comparison is always available in the Current State bit in the Status A register (STATUSA.STATEx).

After the start-up time has passed, a comparison is done and STATUSA is updated. The Comparator x Ready bit in the Status B register (STATUSB.READYx) is set, and the appropriate peripheral events and

interrupts are also generated. New comparisons are performed continuously until the COMPCTRLx.ENABLE bit is written to zero. The start-up time applies only to the first comparison.

In continuous operation, edge detection of the comparator output for interrupts is done by comparing the current and previous sample. The sampling rate is the GCLK_AC frequency. An example of continuous measurement is shown in the Figure 43-2 Continuous Measurement Example on page 1077.

Figure 43-2. Continuous Measurement Example



For low-power operation, comparisons can be performed during sleep modes without a clock. The comparator is enabled continuously, and changes of the comparator state are detected asynchronously. When a toggle occurs, the Power Manager will start GCLK_AC to register the appropriate peripheral events and interrupts. The GCLK_AC clock is then disabled again automatically, unless configured to wake up the system from sleep.

Related Links

Electrical Characteristics on page 1140

Single-Shot

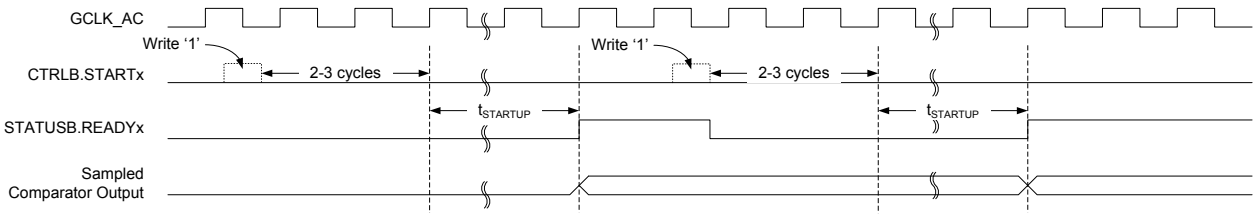
Single-shot operation is selected by writing COMPCTRLx.SINGLE to '1'. During single-shot operation, the comparator is normally idle. The user starts a single comparison by writing '1' to the respective Start Comparison bit in the write-only Control B register (CTRLB.STARTx). The comparator is enabled, and after the start-up time has passed, a single comparison is done and STATUSA is updated. Appropriate peripheral events and interrupts are also generated. No new comparisons will be performed.

Writing '1' to CTRLB.STARTx also clears the Comparator x Ready bit in the Status B register (STATUSB.READYx). STATUSB.READYx is set automatically by hardware when the single comparison has completed.

A single-shot measurement can also be triggered by the Event System. Setting the Comparator x Event Input bit in the Event Control Register (EVCTRL.COMPEIx) enables triggering on incoming peripheral events. Each comparator can be triggered independently by separate events. Event-triggered operation is similar to user-triggered operation; the difference is that a peripheral event from another hardware module causes the hardware to automatically start the comparison and clear STATUSB.READYx.

To detect an edge of the comparator output in single-shot operation for the purpose of interrupts, the result of the current measurement is compared with the result of the previous measurement (one sampling period earlier). An example of single-shot operation is shown in Figure 43-3 Single-Shot Example on page 1077.

Figure 43-3. Single-Shot Example



For low-power operation, event-triggered measurements can be performed during sleep modes. When the event occurs, the Power Manager will start GCLK_AC. The comparator is enabled, and after the

startup time has passed, a comparison is done and appropriate peripheral events and interrupts are also generated. The comparator and GCLK_AC are then disabled again automatically, unless configured to wake up the system from sleep.

Related Links

Electrical Characteristics on page 1140

43.6.3. Selecting Comparator Inputs

Each comparator has one positive and one negative input. The positive input is one of the external input pins (AINx). The negative input can be fed either from an external input pin (AINx) or from one of the several internal reference voltage sources common to all comparators. The user selects the input source as follows:

- The positive input is selected by the Positive Input MUX Select bit group in the Comparator Control register (COMPCTRLx.MUXPOS)
- The negative input is selected by the Negative Input MUX Select bit group in the Comparator Control register (COMPCTRLx.MUXNEG)

In the case of using an external I/O pin, the selected pin must be configured for analog use in the PORT Controller by disabling the digital input and output. The switching of the analog input multiplexers is controlled to minimize crosstalk between the channels. The input selection must be changed only while the individual comparator is disabled.

Note: For internal use of the comparison results by the CCL, this bit must be 0x1 or 0x2.

43.6.4. Window Operation

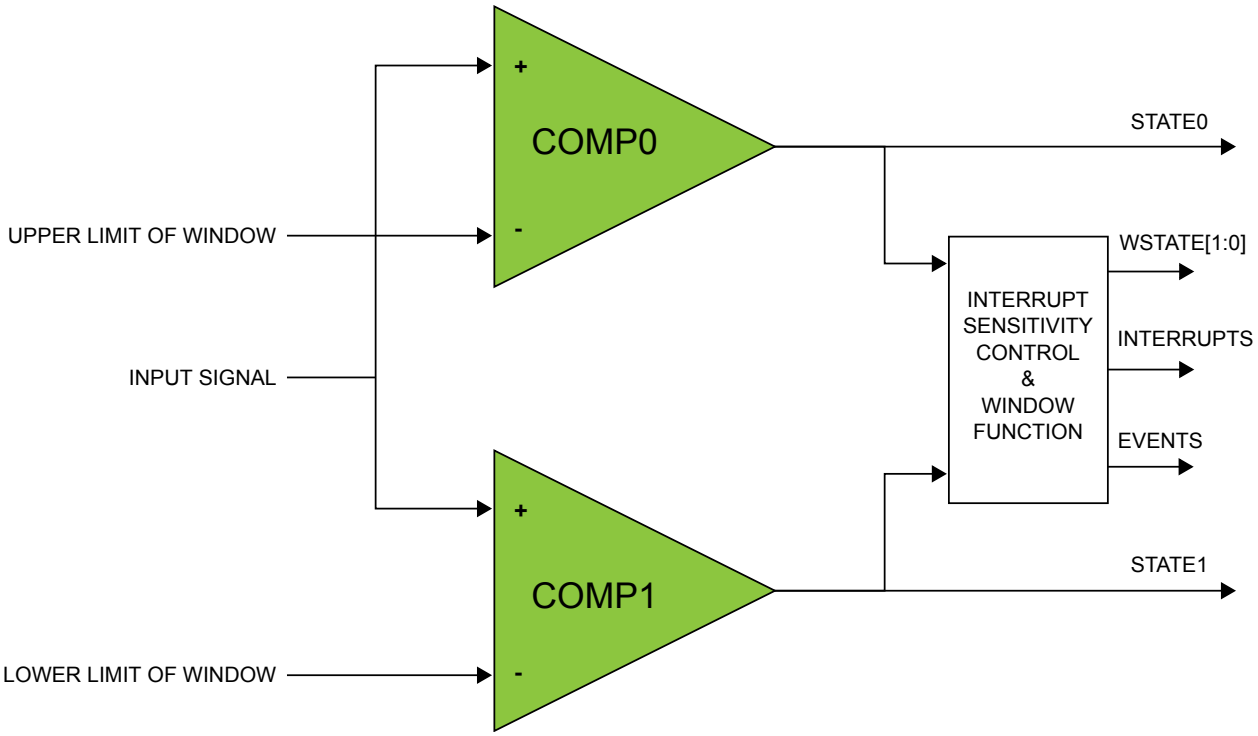
Each comparator pair can be configured to work together in window mode. In this mode, a voltage range is defined, and the comparators give information about whether an input signal is within this range or not. Window mode is enabled by the Window Enable x bit in the Window Control register (WINCTRL.WENx). Both comparators in a pair must have the same measurement mode setting in their respective Comparator Control Registers (COMPCTRLx.SINGLE).

To physically configure the pair of comparators for window mode, the same I/O pin must be chosen as positive input for each comparator, providing a shared input signal. The negative inputs define the range for the window. In Figure 43-4 Comparators in Window Mode on page 1079, COMP0 defines the upper limit and COMP1 defines the lower limit of the window, as shown but the window will also work in the opposite configuration with COMP0 lower and COMP1 higher. The current state of the window function is available in the Window x State bit group of the Status register (STATUS.WSTATEx).

Window mode can be configured to generate interrupts when the input voltage changes to below the window, when the input voltage changes to above the window, when the input voltage changes into the window or when the input voltage changes outside the window. The interrupt selections are set by the Window Interrupt Selection bit field in the Window Control register (WINCTRL.WINTSEL). Events are generated using the inside/outside state of the window, regardless of whether the interrupt is enabled or not. Note that the individual comparator outputs, interrupts and events continue to function normally during window mode.

When the comparators are configured for window mode and single-shot mode, measurements are performed simultaneously on both comparators. Writing '1' to either Start Comparison bit in the Control B register (CTRLB.STARTx) will start a measurement. Likewise either peripheral event can start a measurement.

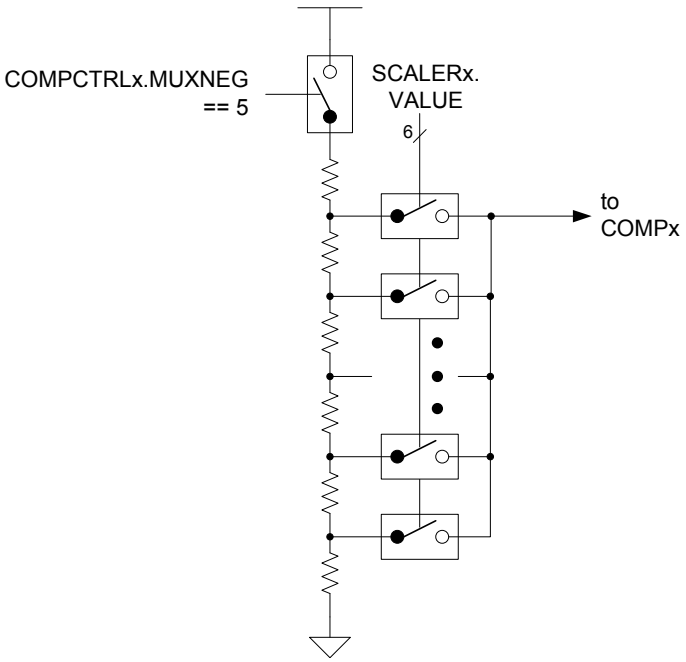
Figure 43-4. Comparators in Window Mode



43.6.5. VDD Scaler

The VDD scaler generates a reference voltage that is a fraction of the device's supply voltage, with 64 levels. One independent voltage channel is dedicated for each comparator. The scaler of a comparator is enabled when the Negative Input Mux bit field in the respective Comparator Control register (COMPCTRLx.MUXNEG) is set to 0x5 and the comparator is enabled. The voltage of each channel is selected by the Value bit field in the Scaler x registers (SCALERx.VALUE).

Figure 43-5. VDD Scaler



43.6.6. Input Hysteresis

Application software can selectively enable/disable hysteresis for the comparison. Applying hysteresis will help prevent constant toggling of the output, which can be caused by noise when the input signals are close to each other.

Hysteresis is enabled for each comparator individually by the Hysteresis Enable bit in the Comparator x Control register (COMPCTRLx.HYSTEN). Furthermore, when enabled, the level of hysteresis is programmable through the Hysteresis Level bits also in the Comparator x Control register (COMPCTRLx.HYST). Hysteresis is available only in continuous mode (COMPCTRLx.SINGLE=0).

43.6.7. Propagation Delay vs. Power Consumption

It is possible to trade off comparison speed for power efficiency to get the shortest possible propagation delay or the lowest power consumption. The speed setting is configured for each comparator individually by the Speed bit group in the Comparator x Control register (COMPCTRLx.SPEED). The Speed bits select the amount of bias current provided to the comparator, and as such will also affect the start-up time.

43.6.8. Filtering

The output of the comparators can be filtered digitally to reduce noise. The filtering is determined by the Filter Length bits in the Comparator Control x register (COMPCTRLx.FLEN), and is independent for each comparator. Filtering is selectable from none, 3-bit majority (N=3) or 5-bit majority (N=5) functions. Any change in the comparator output is considered valid only if N/2+1 out of the last N samples agree. The filter sampling rate is the GCLK_AC frequency.

Note that filtering creates an additional delay of N-1 sampling cycles from when a comparison is started until the comparator output is validated. For continuous mode, the first valid output will occur when the required number of filter samples is taken. Subsequent outputs will be generated every cycle based on the current sample plus the previous N-1 samples, as shown in Figure 43-6 Continuous Mode Filtering on page 1080. For single-shot mode, the comparison completes after the Nth filter sample, as shown in Figure 43-7 Single-Shot Filtering on page 1080.

Figure 43-6. Continuous Mode Filtering

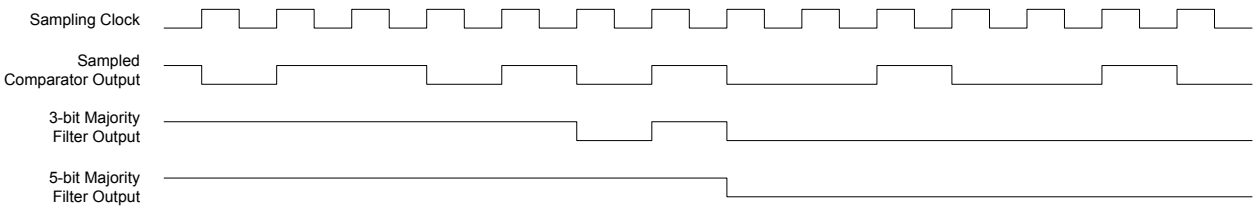
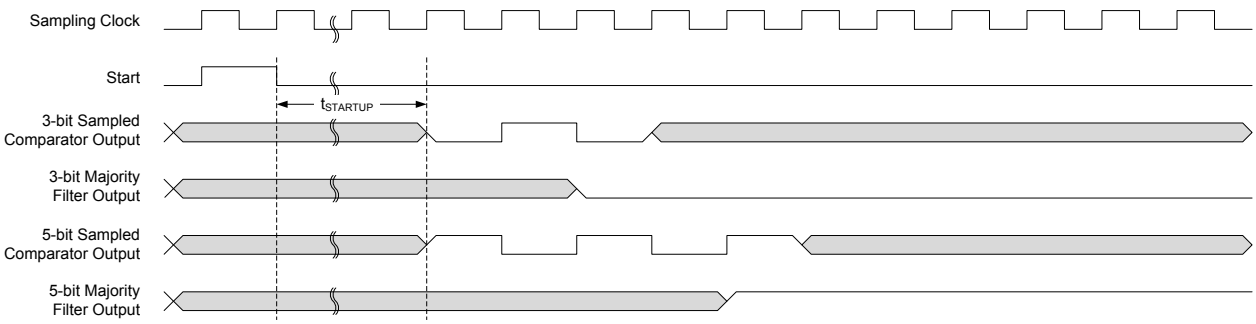


Figure 43-7. Single-Shot Filtering



During sleep modes, filtering is supported only for single-shot measurements. Filtering must be disabled if continuous measurements will be done during sleep modes, or the resulting interrupt/event may be generated incorrectly.

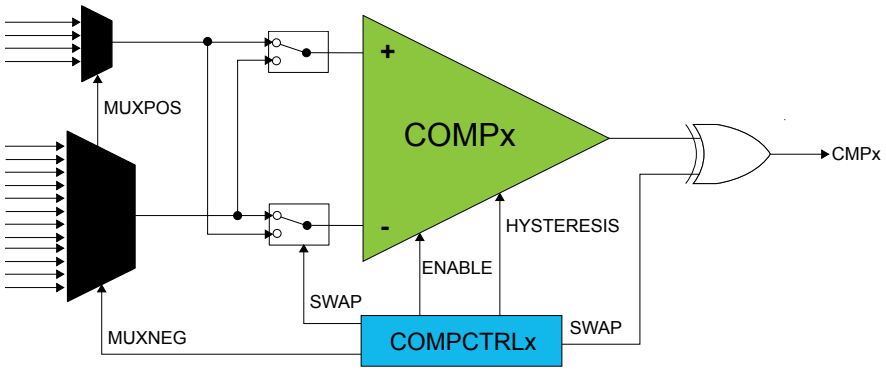
43.6.9. Comparator Output

The output of each comparator can be routed to an I/O pin by setting the Output bit group in the Comparator Control x register (COMPCTRLx.OUT). This allows the comparator to be used by external circuitry. Either the raw, non-synchronized output of the comparator or the CLK_AC-synchronized version, including filtering, can be used as the I/O signal source. The output appears on the corresponding CMP[x] pin.

43.6.10. Offset Compensation

The Swap bit in the Comparator Control registers (COMPCTRLx.SWAP) controls switching of the input signals to a comparator's positive and negative terminals. When the comparator terminals are swapped, the output signal from the comparator is also inverted, as shown in Figure 43-8 Input Swapping for Offset Compensation on page 1081. This allows the user to measure or compensate for the comparator input offset voltage. As part of the input selection, COMPCTRLx.SWAP can be changed only while the comparator is disabled.

Figure 43-8. Input Swapping for Offset Compensation



43.6.11. DMA Operation

Not applicable.

43.6.12. Interrupts

The AC has the following interrupt sources:

- Comparator (COMP0, COMP1): Indicates a change in comparator status.
- Window (WIN0): Indicates a change in the window status.

Comparator interrupts are generated based on the conditions selected by the Interrupt Selection bit group in the Comparator Control registers (COMPCTRLx.INTSEL). Window interrupts are generated based on the conditions selected by the Window Interrupt Selection bit group in the Window Control register (WINCTRL.WINTSEL[1:0]).

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. Each interrupt can be individually enabled by writing a one to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a one to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register. An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the AC is reset. See INFLAG for details on how to clear interrupt flags. All interrupt requests from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. The user must read the INTFLAG register to determine which interrupt condition is present.

Note that interrupts must be globally enabled for interrupt requests to be generated.

Related Links

Nested Vector Interrupt Controller on page 51

43.6.13. Events

The AC can generate the following output events:

- Comparator (COMP0, COMP1): Generated as a copy of the comparator status
- Window (WIN0): Generated as a copy of the window inside/outside status

Writing a one to an Event Output bit in the Event Control Register (EVCTRL.xxEO) enables the corresponding output event. Writing a zero to this bit disables the corresponding output event. Refer to the Event System chapter for details on configuring the event system.

The AC can take the following action on an input event:

- Start comparison (START0, START1): Start a comparison.

Writing a one to an Event Input bit into the Event Control register (EVCTRL.COMPEIx) enables the corresponding action on input event. Writing a zero to this bit disables the corresponding action on input event. Note that if several events are connected to the AC, the enabled action will be taken on any of the incoming events. Refer to the Event System chapter for details on configuring the event system.

When EVCTRL.COMPEIx is one, the event will start a comparison on COMPx after the start-up time delay. In normal mode, each comparator responds to its corresponding input event independently. For a pair of comparators in window mode, either comparator event will trigger a comparison on both comparators simultaneously.

43.6.14. Sleep Mode Operation

The Run in Standby bits in the Comparator x Control registers (COMPCTRLx.RUNSTDBY) control the behavior of the AC during standby sleep mode. Each RUNSTDBY bit controls one comparator. When the bit is zero, the comparator is disabled during sleep, but maintains its current configuration. When the bit is one, the comparator continues to operate during sleep. Note that when RUNSTDBY is zero, the analog blocks are powered off for the lowest power consumption. This necessitates a start-up time delay when the system returns from sleep.

For Window Mode operation, both comparators in a pair must have the same RUNSTDBY configuration.

When RUNSTDBY is one, any enabled AC interrupt source can wake up the CPU. The AC can also be used during sleep modes where the clock used by the AC is disabled, provided that the AC is still powered (not in shutdown). In this case, the behavior is slightly different and depends on the measurement mode, as listed in Table 43-1 Sleep Mode Operation on page 1082.

Table 43-1. Sleep Mode Operation

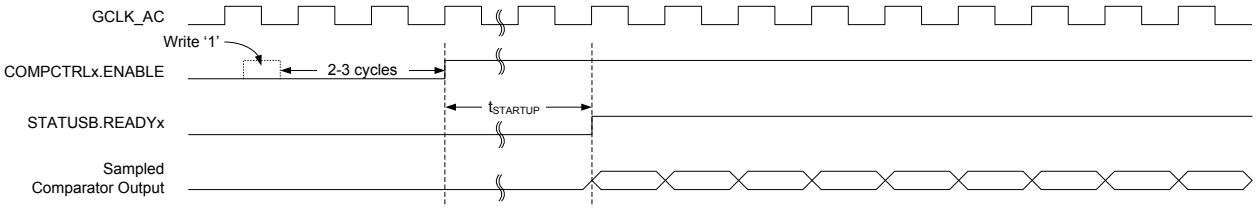
COMPCTRLx.MODE	RUNSTDBY=0	RUNSTDBY=1
0 (Continuous)	COMPx disabled	GCLK_AC stopped, COMPx enabled
1 (Single-shot)	COMPx disabled	GCLK_AC stopped, COMPx enabled only when triggered by an input event

43.6.14.1. Continuous Measurement during Sleep

When a comparator is enabled in continuous measurement mode and GCLK_AC is disabled during sleep, the comparator will remain continuously enabled and will function asynchronously. The current state of the comparator is asynchronously monitored for changes. If an edge matching the interrupt condition is found, GCLK_AC is started to register the interrupt condition and generate events. If the interrupt is enabled in the Interrupt Enable registers (INTENCLR/SET), the AC can wake up the device;

otherwise GCLK_AC is disabled until the next edge detection. Filtering is not possible with this configuration.

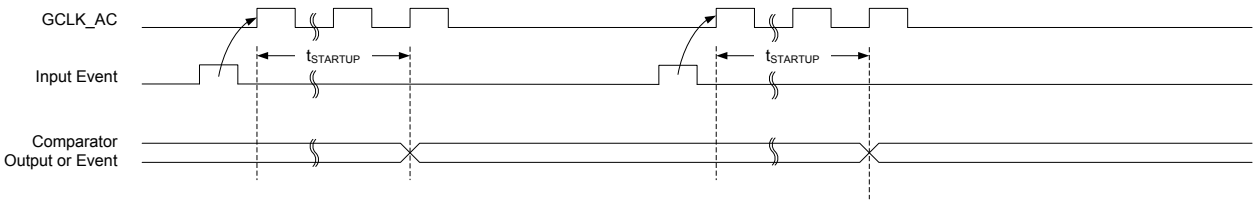
Figure 43-9. Continuous Mode SleepWalking



43.6.14.2. Single-Shot Measurement during Sleep

For low-power operation, event-triggered measurements can be performed during sleep modes. When the event occurs, the Power Manager will start GCLK_AC. The comparator is enabled, and after the start-up time has passed, a comparison is done, with filtering if desired, and the appropriate peripheral events and interrupts are also generated, as shown in Figure 43-10 Single-Shot SleepWalking on page 1083. The comparator and GCLK_AC are then disabled again automatically, unless configured to wake the system from sleep. Filtering is allowed with this configuration.

Figure 43-10. Single-Shot SleepWalking



43.6.15. Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read. An exception is the Channel Enable bit in the Peripheral Channel Control registers (PCHCTRLm.CHEN). When changing this bit, the bit value must be read-back to ensure the synchronization is complete and to assert glitch free internal operation. Note that changing the bit value under ongoing synchronization will *not* generate an error.

The following bits are synchronized when written:

- Software Reset bit in control register (CTRLA.SWRST)
- Enable bit in control register (CTRLA.ENABLE)
- Enable bit in Comparator Control register (COMPCTRLn.ENABLE)

The following registers are synchronized when written:

- Window Control register (WINCTRL)

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

Related Links

Register Synchronization on page 127

43.7. Register Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0							ENABLE	SWRST
0x01	CTRLB	7:0							START1	START0
0x02	EVCTRL	7:0				WINEO0			COMPEO1	COMPEO0
0x03		15:8			INVEI1	INVEI0			COMPEI1	COMPEI0
0x04	INTENCLR	7:0				WIN0			COMP1	COMP0
0x05	INTENSET	7:0				WIN0			COMP1	COMP0
0x06	INTFLAG	7:0				WIN0			COMP1	COMP0
0x07	STATUSA	7:0			WSTATE0[1:0]				STATE1	STATE0
0x08	STATUSB	7:0							READY1	READY0
0x09	DBGCTRL	7:0								DBGRUN
0x0A	WINCTRL	7:0						WINTSEL0[1:0]		WEN0
0x0B	Reserved									
0x0C	SCALER0	7:0			VALUE[5:0]					
0x0D	SCALER1	7:0			VALUE[5:0]					
0x0E ... 0x0F	Reserved									
0x10	COMPCTRL0	7:0		RUNSTDBY		INTSEL[1:0]		SINGLE	ENABLE	
0x11		15:8	SWAP	MUXPOS[2:0]				MUXNEG[2:0]		
0x12		23:16			HYST[1:0]		HYSTEN		SPEED[1:0]	
0x13		31:24			OUT[1:0]			FLEN[2:0]		
0x14	COMPCTRL1	7:0		RUNSTDBY		INTSEL[1:0]		SINGLE	ENABLE	
0x15		15:8	SWAP	MUXPOS[2:0]				MUXNEG[2:0]		
0x16		23:16			HYST[1:0]		HYSTEN		SPEED[1:0]	
0x17		31:24			OUT[1:0]			FLEN[2:0]		
0x18 ... 0x1F	Reserved									
0x20	SYNCBUSY	7:0				COMPCTRL1	COMPCTRL0	WINCTRL	ENABLE	SWRST
0x21		15:8								
0x22		23:16								
0x23		31:24								

43.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection.

Some registers are synchronized when read and/or written. Synchronization is denoted by the "Write-Synchronized" or the "Read-Synchronized" property in each individual register description. For details, refer to Synchronization.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

43.8.1. Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
							ENABLE	SWRST
Access							R/W	R/W
Reset							0	0

Bit 1 – ENABLE: Enable
Due to synchronization, there is delay from updating the register until the peripheral is enabled/disabled. The value written to CTRL.ENABLE will read back immediately and the corresponding bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE is cleared when the peripheral is enabled/disabled.

Value	Description
0	The AC is disabled.
1	The AC is enabled. Each comparator must also be enabled individually by the Enable bit in the Comparator Control register (COMPCTRLn.ENABLE).

Bit 0 – SWRST: Software Reset
Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the AC to their initial state, and the AC will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.

Due to synchronization, there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

43.8.2. Control B

Name: CTRLB
Offset: 0x01
Reset: 0x00
Property: –

Bit	7	6	5	4	3	2	1	0
							START1	START0
Access							R/W	R/W
Reset							0	0

Bits 1,0 – STARTx: Comparator x Start Comparison
Writing a '0' to this field has no effect.

Writing a '1' to STARTx starts a single-shot comparison on COMPx if both the Single-Shot and Enable bits in the Comparator x Control Register are '1' (COMPCTRLx.SINGLE and COMPCTRLx.ENABLE). If comparator x is not implemented, or if it is not enabled in single-shot mode, Writing a '1' has no effect.

This bit always reads as zero.

43.8.3. Event Control

Name: EVCTRL
Offset: 0x02
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
			INVEI1	INVEI0			COMPEI1	COMPEI0
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit	7	6	5	4	3	2	1	0
				WINEO0			COMPEO1	COMPEO0
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 4 – WINEO0: Window 0 Event Output Enable
These bits indicate whether the window 0 function can generate a peripheral event or not.

Value	Description
0	Window 0 Event is disabled.
1	Window 0 Event is enabled.

Bits 13,12 – INVEIx: Inverted Event Input Enable x

Value	Description
0	Incoming event is not inverted for comparator x.
1	Incoming event is inverted for comparator x.

Bits 9,8 – COMPEIx: Comparator x Event Input
Note that several actions can be enabled for incoming events. If several events are connected to the peripheral, the enabled action will be taken for any of the incoming events. There is no way to tell which of the incoming events caused the action.

These bits indicate whether a comparison will start or not on any incoming event.

Value	Description
0	Comparison will not start on any incoming event.
1	Comparison will start on any incoming event.

Bits 1,0 – COMPEOx: Comparator x Event Output Enable
These bits indicate whether the comparator x output can generate a peripheral event or not.

Value	Description
0	COMPx event generation is disabled.
1	COMPx event generation is enabled.

43.8.4. Interrupt Enable Clear

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Name: INTENCLR
Offset: 0x04
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
				WIN0			COMP1	COMP0
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 4 – WIN0: Window 0 Interrupt Enable
Reading this bit returns the state of the Window 0 interrupt enable.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit disables the Window 0 interrupt.

Value	Description
0	The Window 0 interrupt is disabled.
1	The Window 0 interrupt is enabled.

Bits 1,0 – COMPx: Comparator x Interrupt Enable
Reading this bit returns the state of the Comparator x interrupt enable.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit disables the Comparator x interrupt.

Value	Description
0	The Comparator x interrupt is disabled.
1	The Comparator x interrupt is enabled.

43.8.5. Interrupt Enable Set

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Name: INTENSET
Offset: 0x05
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
				WIN0			COMP1	COMP0
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 4 – WIN0: Window 0 Interrupt Enable
Reading this bit returns the state of the Window 0 interrupt enable.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit enables the Window 0 interrupt.

Value	Description
0	The Window 0 interrupt is disabled.
1	The Window 0 interrupt is enabled.

Bits 1,0 – COMPx: Comparator x Interrupt Enable
Reading this bit returns the state of the Comparator x interrupt enable.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Ready interrupt bit and enable the Ready interrupt.

Value	Description
0	The Comparator x interrupt is disabled.
1	The Comparator x interrupt is enabled.

43.8.6. Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x06
Reset: 0x00
Property: –

Bit	7	6	5	4	3	2	1	0
				WIN0			COMP1	COMP0
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 4 – WIN0: Window 0
This flag is set according to the Window 0 Interrupt Selection bit group in the WINCTRL register (WINCTRL.WINTSELx) and will generate an interrupt if INTENCLR/SET.WINx is also one.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Window 0 interrupt flag.

Bits 1,0 – COMPx: Comparator x
Reading this bit returns the status of the Comparator x interrupt flag. If comparator x is not implemented, COMPx always reads as zero.

This flag is set according to the Interrupt Selection bit group in the Comparator x Control register (COMPCTRLx.INTSEL) and will generate an interrupt if INTENCLR/SET.COMPx is also one.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Comparator x interrupt flag.

43.8.7. Status A

Name: STATUSA
Offset: 0x07
Reset: 0x00
Property: –

Bit	7	6	5	4	3	2	1	0
			WSTATE0[1:0]				STATE1	STATE0
Access			R	R			R	R
Reset			0	0			0	0

Bits 5:4 – WSTATE0[1:0]: Window 0 Current State
These bits show the current state of the signal if the window 0 mode is enabled.

Value	Name	Description
0x0	ABOVE	Signal is above window
0x1	INSIDE	Signal is inside window
0x2	BELOW	Signal is below window
0x3		Reserved

Bits 1,0 – STATEx: Comparator x Current State
This bit shows the current state of the output signal from COMPx. STATEx is valid only when STATUSB.READYx is one.

43.8.8. Status B

Name: STATUSB
Offset: 0x08
Reset: 0x00
Property: –

Bit	7	6	5	4	3	2	1	0
							READY1	READY0
Access							R	R
Reset							0	0

Bits 1,0 – READYx: Comparator x Ready
This bit is cleared when the comparator x output is not ready.
This bit is set when the comparator x output is ready.

43.8.9. Debug Control

Name: DBGCTRL
Offset: 0x09
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGRUN
Access								R/W
Reset								0

Bit 0 – DBGRUN: Debug Run
This bit is not reset by a software reset.

This bits controls the functionality when the CPU is halted by an external debugger.

Value	Description
0	The AC is halted when the CPU is halted by an external debugger. Any on-going comparison will complete.
1	The AC continues normal operation when the CPU is halted by an external debugger.

43.8.10. Window Control

Name: WINCTRL
Offset: 0x0A
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
						WINTSEL0[1:0]		WEN0
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 2:1 – WINTSEL0[1:0]: Window 0 Interrupt Selection
These bits configure the interrupt mode for the comparator window 0 mode.

Value	Name	Description
0x0	ABOVE	Interrupt on signal above window
0x1	INSIDE	Interrupt on signal inside window
0x2	BELOW	Interrupt on signal below window
0x3	OUTSIDE	Interrupt on signal outside window

Bit 0 – WEN0: Window 0 Mode Enable

Value	Description
0	Window mode is disabled for comparators 0 and 1.
1	Window mode is enabled for comparators 0 and 1.

43.8.11. Scaler n

Name: SCALERn
Offset: 0x0C + n*0x01 [n=0..1]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
			VALUE[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 5:0 – VALUE[5:0]: Scaler Value
These bits define the scaling factor for channel n of the V_{DD} voltage scaler. The output voltage, V_{SCALE}, is:

$$V_{SCALE} = \frac{V_{DD} \cdot (VALUE+1)}{64}$$

43.8.12. Comparator Control n

Name: COMPCTRLn
Offset: 0x10 + n*0x04 [n=0..1]
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
			OUT[1:0]			FLEN[2:0]		
Access			R/W	R/W		R/W	R/W	R/W
Reset			0	0		0	0	0

Bit	23	22	21	20	19	18	17	16
			HYST[1:0]		HYSTEN		SPEED[1:0]	
Access			R/W	R/W	R/W		R/W	R/W
Reset			0	0	0		0	0

Bit	15	14	13	12	11	10	9	8
	SWAP	MUXPOS[2:0]				MUXNEG[2:0]		
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bit	7	6	5	4	3	2	1	0
		RUNSTDBY		INTSEL[1:0]		SINGLE	ENABLE	
Access		R/W		R/W	R/W	R/W	R/W	
Reset		0		0	0	0	0	

Bits 29:28 – OUT[1:0]: Output
These bits configure the output selection for comparator n. COMPCTRLn.OUT can be written only while COMPCTRLn.ENABLE is zero.

Note: For internal use of the comparison results by the CCL, this bit must be 0x1 or 0x2.

These bits are not synchronized.

Value	Name	Description
0x0	OFF	The output of COMPn is not routed to the COMPn I/O port
0x1	ASYNC	The asynchronous output of COMPn is routed to the COMPn I/O port
0x2	SYNC	The synchronous output (including filtering) of COMPn is routed to the COMPn I/O port
0x3	N/A	Reserved

Bits 26:24 – FLEN[2:0]: Filter Length
These bits configure the filtering for comparator n. COMPCTRLn.FLEN can only be written while COMPCTRLn.ENABLE is zero.

These bits are not synchronized.

Value	Name	Description
0x0	OFF	No filtering
0x1	MAJ3	3-bit majority function (2 of 3)
0x2	MAJ5	5-bit majority function (3 of 5)
0x3-0x7	N/A	Reserved

Bits 21:20 – HYST[1:0]: Hysteresis Level
These bits indicate the hysteresis level of comparator n when hysteresis is enabled (COMPCTRLn.HYSTEN=1). Hysteresis is available only for continuous mode (COMPCTRLn.SINGLE=0). COMPCTRLn.HYST can be written only while COMPCTRLn.ENABLE is zero.

These bits are not synchronized.

Value	Name	Description
0x0	HYST50	50mV
0x1	HYST70	70mV
0x2	HYST90	90mV
0x3	HYST110	110mV

Bit 19 – HYSTEN: Hysteresis Enable
This bit indicates the hysteresis mode of comparator n. Hysteresis is available only for continuous mode (COMPCTRLn.SINGLE=0). COMPCTRLn.HYST can be written only while COMPCTRLn.ENABLE is zero.

This bit is not synchronized.

Value	Description
0	Hysteresis is disabled.
1	Hysteresis is enabled.

Bits 17:16 – SPEED[1:0]: Speed Selection
This bit indicates the speed/propagation delay mode of comparator n. COMPCTRLn.SPEED can be written only while COMPCTRLn.ENABLE is zero.

These bits are not synchronized.

Value	Name	Description
0x0	LOW	Low speed
0x1	MEDLOW	Medium low speed
0x2	MEDHIGH	Medium high speed
0x3	HIGH	High speed

Bit 15 – SWAP: Swap Inputs and Invert
This bit swaps the positive and negative inputs to COMPn and inverts the output. This function can be used for offset cancellation. COMPCTRLn.SWAP can be written only while COMPCTRLn.ENABLE is zero.

These bits are not synchronized.

Value	Description
0	The output of MUXPOS connects to the positive input, and the output of MUXNEG connects to the negative input.
1	The output of MUXNEG connects to the positive input, and the output of MUXPOS connects to the negative input.

Bits 14:12 – MUXPOS[2:0]: Positive Input Mux Selection
These bits select which input will be connected to the positive input of comparator n. COMPCTRLn.MUXPOS can be written only while COMPCTRLn.ENABLE is zero.

These bits are not synchronized.

Value	Name	Description
0x0	PIN0	I/O pin 0
0x1	PIN1	I/O pin 1
0x2	PIN2	I/O pin 2
0x3	PIN3	I/O pin 3
0x4	VSCALE	VDD scaler
0x5–0x7	-	Reserved

Bits 10:8 – MUXNEG[2:0]: Negative Input Mux Selection
These bits select which input will be connected to the negative input of comparator n. COMPCTRLn.MUXNEG can only be written while COMPCTRLn.ENABLE is zero.

These bits are not synchronized.

Value	Name	Description
0x0	PIN0	I/O pin 0
0x1	PIN1	I/O pin 1
0x2	PIN2	I/O pin 2
0x3	PIN3	I/O pin 3
0x4	GND	Ground
0x5	VSCALE	VDD scaler
0x6	BANDGAP	Internal bandgap voltage
0x7	DAC / OPAMP	DAC0 output (COMP0) OPAMP2 output (COMP1)

Bit 6 – RUNSTDBY: Run in Standby
This bit controls the behavior of the comparator during standby sleep mode.

This bit is not synchronized

Value	Description
0	The comparator is disabled during sleep.
1	The comparator continues to operate during sleep.

Bits 4:3 – INTSEL[1:0]: Interrupt Selection
These bits select the condition for comparator n to generate an interrupt or event. COMPCTRLn.INTSEL can be written only while COMPCTRLn.ENABLE is zero.

These bits are not synchronized.

Value	Name	Description
0x0	TOGGLE	Interrupt on comparator output toggle
0x1	RISING	Interrupt on comparator output rising
0x2	FALLING	Interrupt on comparator output falling
0x3	EOC	Interrupt on end of comparison (single-shot mode only)

Bit 2 – SINGLE: Single-Shot Mode
This bit determines the operation of comparator n. COMPCTRLn.SINGLE can be written only while COMPCTRLn.ENABLE is zero.

These bits are not synchronized.

Value	Description
0	Comparator n operates in continuous measurement mode.
1	Comparator n operates in single-shot mode.

Bit 1 – ENABLE: Enable
Writing a zero to this bit disables comparator n.
Writing a one to this bit enables comparator n.

Due to synchronization, there is delay from updating the register until the comparator is enabled/disabled. The value written to COMPCTRLn.ENABLE will read back immediately after being written. SYNCBUSY.COMPCTRLn is set. SYNCBUSY.COMPCTRLn is cleared when the peripheral is enabled/disabled.

Writing a one to COMPCTRLn.ENABLE will prevent further changes to the other bits in COMPCTRLn. These bits remain protected until COMPCTRLn.ENABLE is written to zero and the write is synchronized.

43.8.13. Synchronization Busy

Name: SYNCBUSY
Offset: 0x20
Reset: 0x00000000
Property: –

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				COMPCTRL1	COMPCTRL0	WINCTRL	ENABLE	SWRST
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit 2 – WINCTRL: WINCTRL Synchronization Busy
This bit is cleared when the synchronization of the WINCTRL register between the clock domains is complete.

This bit is set when the synchronization of the WINCTRL register between clock domains is started.

Bit 1 – ENABLE: Enable Synchronization Busy
This bit is cleared when the synchronization of the CTRLA.ENABLE bit between the clock domains is complete.

This bit is set when the synchronization of the CTRLA.ENABLE bit between clock domains is started.

Bit 0 – SWRST: Software Reset Synchronization Busy
This bit is cleared when the synchronization of the CTRLA.SWRST bit between the clock domains is complete.

This bit is set when the synchronization of the CTRLA.SWRST bit between clock domains is started.

Bits 4,3 – COMPCTRLx: COMPCTRLx Synchronization Busy
This bit is cleared when the synchronization of the COMPCTRLx register between the clock domains is complete.

This bit is set when the synchronization of the COMPCTRLx register between clock domains is started.

44. DAC – Digital-to-Analog Converter

44.1. Overview

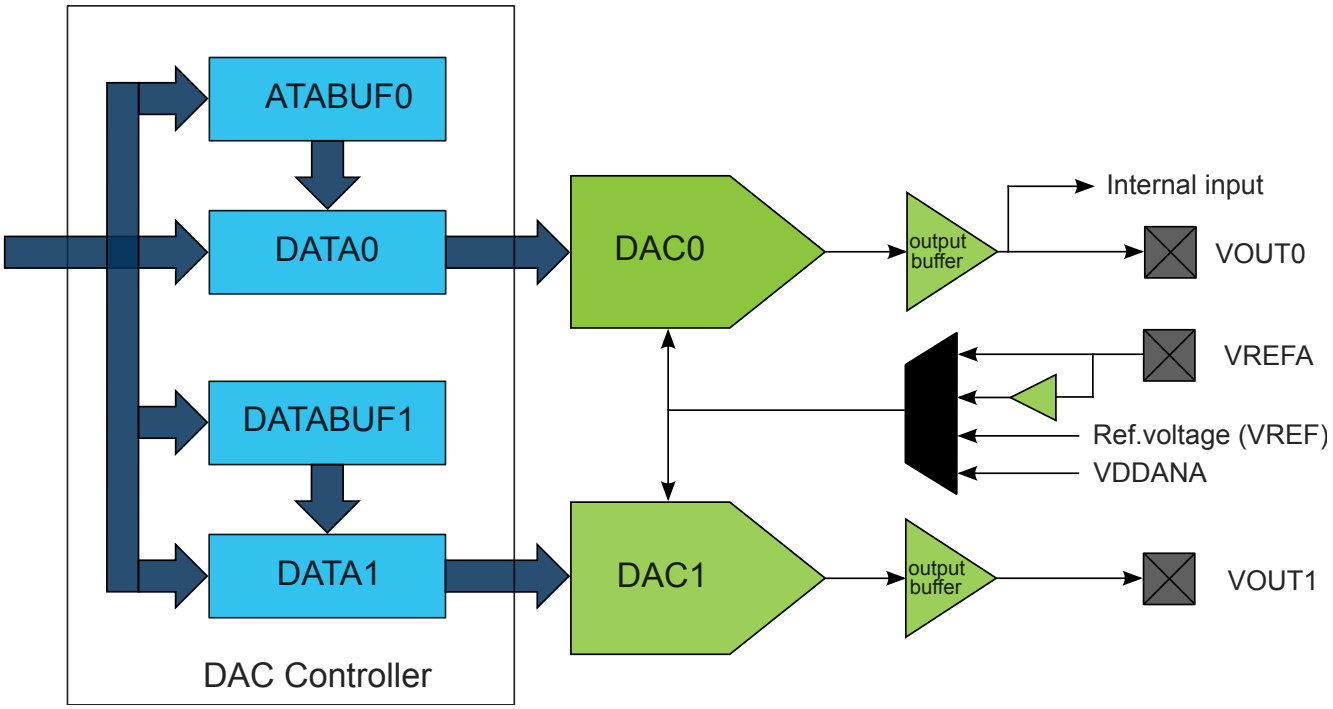
The Digital-to-Analog Converter (DAC) converts a digital value to a voltage. The DAC Controller controls two DACs, which can operate either as two independent DACs or as a single DAC in differential mode. Each DAC is 12-bit resolution and it is capable of converting up to 1,000,000 samples per second (MSPS).

44.2. Features

- Two independent DACs or single DAC in differential mode
- DAC with 12-bit resolution
- Up to 1MSPS conversion rate
- Hardware support for 16-bit using dithering
- Multiple trigger sources
- High-drive capabilities
- DAC0 used as internal input
- DMA support

44.3. Block Diagram

Figure 44-1. DAC Controller Block Diagram.



44.4. Signal Description

Signal	Description	Type
VOUT0	DAC0 output	Analog output
VOUT1	DAC1 output	Analog output
VREFA	External reference	Analog input

One signal can be mapped on several pins.

Important:
When an analog peripheral is enabled, the analog output of the peripheral will interfere with the alternative functions of the output pads. This is also true even when the peripheral is used for internal purposes.

Analog inputs do not interfere with alternative pad functions.

Related Links

I/O Multiplexing and Considerations on page 29

44.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

44.5.1. I/O Lines

Using the DAC Controller's I/O lines requires the I/O pins to be configured.

Table 44-1. I/O Lines

Instance	Signal	I/O Line	Peripheral Function
DAC	VOUT0	PAxx	A
DAC	VOUT1	PAxx	A
DAC	VREFA	PAxx	A

44.5.2. Power Management

The DAC Controller will continue to operate in any sleep mode where the selected source clock is running.

The DAC Controller interrupts can be used to wake up the device from sleep modes.

Events connected to the event system can trigger other operations in the system without exiting sleep modes.

Related Links

PM – Power Manager on page 186

44.5.3. Clocks

The DAC bus clock (CLK_DAC_APB) can be enabled and disabled in the Main Clock module, and the default state of CLK_DAC_APB can be found in *Peripheral Clock Masking*.

A generic clock (GCLK_DAC) is required to clock the DAC Controller. This clock must be configured and enabled in the generic clock controller before using the DAC Controller.

This generic clock is asynchronous to the bus clock (CLK_DAC_APB). Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains. Refer to Synchronization on page 1111 for further details.

Related Links

Peripheral Clock Masking on page 151

GCLK - Generic Clock Controller on page 131

44.5.4. DMA

The DMA request line is connected to the DMA Controller (DMAC). Using the DAC Controller DMA requests requires to configure the DMAC first.

Related Links

DMAC – Direct Memory Access Controller on page 399

44.5.5. Interrupts

The interrupt request line is connected to the interrupt controller. Using the DAC Controller interrupts requires the interrupt controller to be configured first.

Related Links

Nested Vector Interrupt Controller on page 51

Nested Vector Interrupt Controller on page 51

Sleep Mode Controller on page 191

44.5.6. Events

The events are connected to the Event System.

Related Links

EVSYS – Event System on page 536

44.5.7. Debug Operation

When the CPU is halted in debug mode the DAC will halt normal operation. Any on-going conversions will be completed. The DAC can be forced to continue normal operation during debugging. If the DAC is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

44.5.8. Register Access Protection

All registers with write-access can be write-protected optionally by the Peripheral Access Controller (PAC), except the following registers:

- Interrupt Flag Status and Clear (INTFLAG) register
- Data Buffer (DATABUFx) registers

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

PAC write-protection does not apply to accesses through an external debugger.

Related Links

PAC - Peripheral Access Controller on page 58

44.5.9. Analog Connections

The DAC has up to two analog output pins (VOUT0, VOUT1) and one analog input pin (VREFA) that must be configured first.

When an internal input is used, it must be enabled before DAC Controller is enabled.

The analog signals of AC, ADC, DAC and OPAMP can be interconnected.

See *Analog Connections of Peripherals* for details.

Related Links

Analog Connections of Peripherals on page 34

44.6. Functional Description

44.6.1. Principle of Operation

Each DAC converts the digital value located in the Data register (DATA0 or DATA1) into an analog voltage on the DAC output (VOUT0 or VOUT1, respectively).

A conversion is started when new data is loaded to the Data register. The resulting voltage is available on the DAC output after the conversion time. A conversion can also be started by input events from Event System.

44.6.2. Basic Operation

44.6.2.1. Initialization

The following registers are enable-protected, meaning they can only be written when the DAC Controller is disabled (CTRLA.ENABLE=0):

- Control B register (CTRLB)
- Event Control register (EVCTRL)
- DAC0 Control (DACCTRL0)
- DAC1 Control (DACCTRL1)

Enable-protection is denoted by the Enable-Protected property in the register description.

44.6.2.2. Enabling, Disabling and Resetting

The DAC Controller is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The DAC Controller is disabled by writing a '0' to CTRLA.ENABLE.

The DAC Controller is reset by writing '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the DAC will be reset to their initial state, and the DAC Controller will be disabled. Refer to CTRLA on page 1114 for details.

44.6.2.3. DAC Configuration

Each individual DAC is configured by its respective DAC Control register (DACCTRLx)). These settings are applied when DAC Controller is enabled and can be changed only when DAC Controller is disabled.

- Enable the selected DAC by writing a '1' to DACCTRLx.ENABLE.
- Select the data alignment with DACCCTRLx.LEFTADJ. Writing a '1' will left-align the data (DATABUFx/DATAx[31:20]). Writing a '0' to LEFTADJ will right-align the data (DATABUFx/DATAx[11:0]).
- If operation in standby mode is desired for DACx, write a '1' to the Run in Standby bit in the DAC Control register (DACCCTRLx.RUNSTDBY). If RUNSTDBY=1, DACx continues normal operation when system is in standby mode. If RUNSTDBY=0, DACx is halted in standby mode.

- Select dithering mode with DACCCTRLx.DITHER. Writing '1' to DITHER will enable dithering mode, writing a '0' will disable it. Refer to Dithering Mode on page 1109 for details.
- Select the refresh period with the Refresh Period bit field in DACCCTRLx.REFRESH[3:0]. Writing any value greater than '1' to the REFRESH bit field will enable and select the refresh mode. Refer to Conversion Refresh on page 1108 for details.
- Select the output buffer current according to data rate (for low power application) with the Current Control bit field DACCTRLx.CCTRL[1:0]. Refer to Output Buffer Current Control on page 1108 for details.

Once DAC Controller is enabled, DACx requires a startup time before the first conversion can start. The DACx Startup Ready bit in the Status register (STATUS.READYx) indicates that DACx is ready to convert a data when STATUS.READYx=1.

Conversions started while STATUS.READYx=0 are ignored.

VOUTx is at tri-state level if DACx is not enabled.

44.6.2.4. Digital to Analog Conversion

Each DAC converts a digital value (stored in DATAx register) into an analog voltage. The conversion range is between GND and the selected DAC voltage reference. The default voltage reference is the internal reference voltage. Other voltage reference options are the analog supply voltage (VDDANA) and the external voltage reference (VREFA). The voltage reference is selected by writing to the Reference Selection bits in the Control B register (CTRLB.REFSEL).

The output voltage from the DAC can be calculated using the following formula:

$$V_{\mathrm{OUTx}} = \frac{\mathrm{DATAx}}{4095} \times \mathrm{VREF}$$

A new conversion starts as soon as a new value is loaded into DATAx. DATAx can either be loaded via the APB bus during a CPU write operation, using DMA, or from the DATABUFx register when a STARTx event occurs.

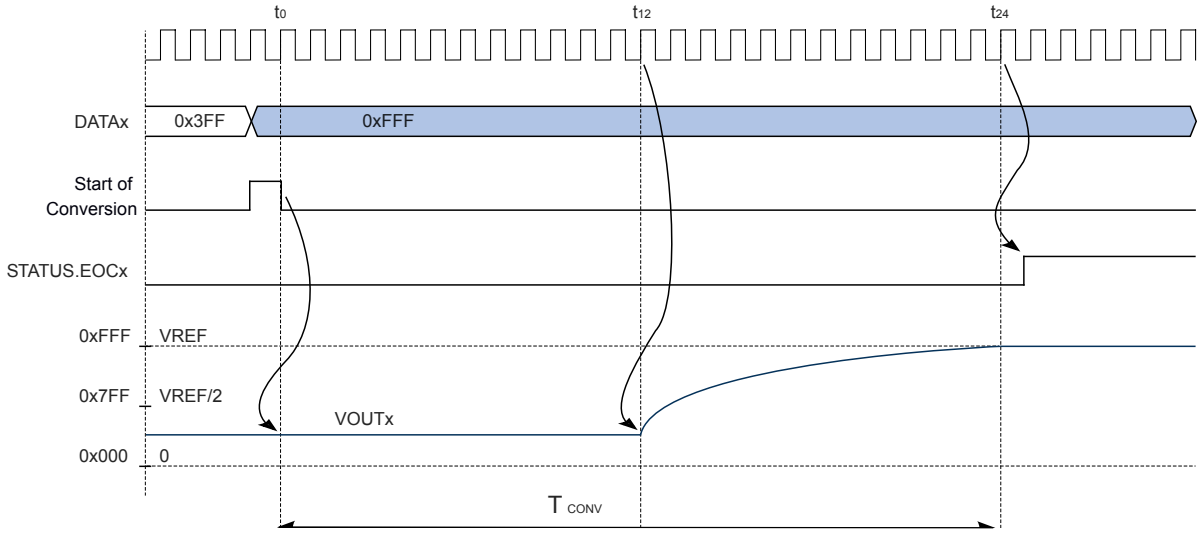
Refer to Events on page 1104 for details. Even if both DAC use the same GCLK, each data conversion can be started independently.

The conversion time is given by the period T_{GCLK} of the generic clock GCLK_DAC and the number of bits:

$$T_{\mathrm{CONV}} = 12 \times 2 \times T_{\mathrm{GCLK}}$$

The End Of Conversion bit in the Status register indicates that a conversion is completed (STATUS.EOCx=1). This means that VOUTx is stable.

Figure 44-2. Single DAC Conversion

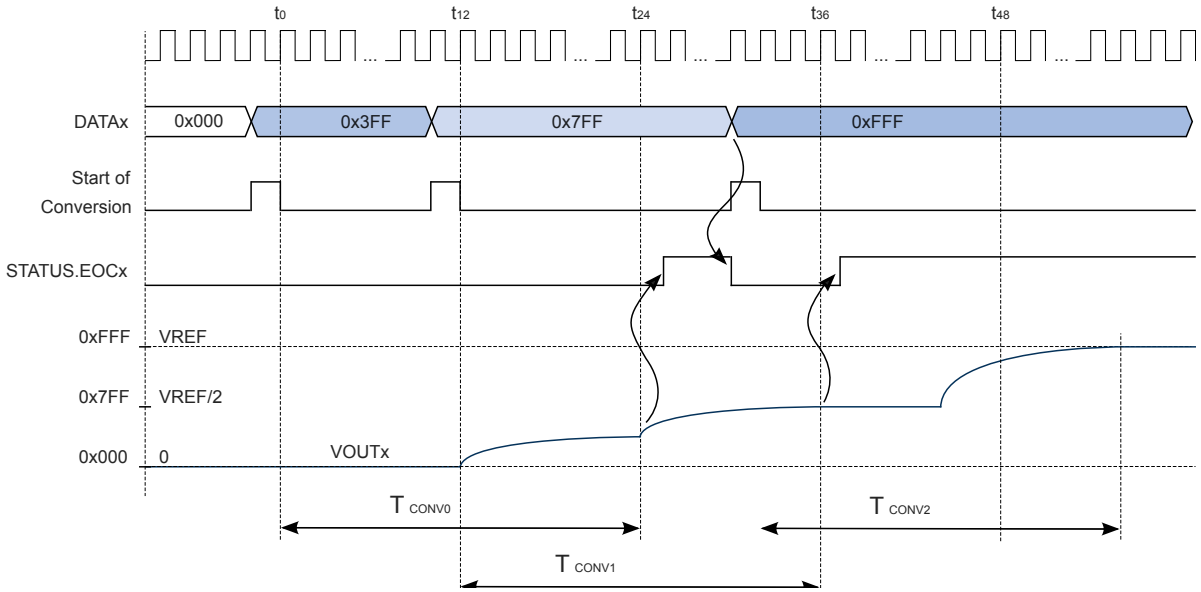


Since the DAC conversion is implemented as pipelined procedure, a new conversion can be started after only 12 GCLK_DAC periods. Therefore if DATAx is written while a conversion is ongoing, start of conversion is postponed until DACx is ready to start next conversion.

The maximum conversion rate (samples per second) is therefore:

$$CR_{max} = \frac{2}{T_{conv}}$$

Figure 44-3. Multiple DAC Conversions



Related Links

SUPC – Supply Controller on page 283

44.6.3. Additional Features

44.6.3.1. DAC0 as Internal Input

The analog output of DAC0, VOUT0, is internally available as input signal for other peripherals (AC, ADC, and OPAMP) when DAC0 is enabled.

Note: The pin VOUT0 will be dedicated as internal input and cannot be configured as alternate function.

44.6.3.2. Output Buffer Current Control

Power consumption can be reduced by controlling the output buffer current, according to conversion rate. Writing to the Current Control bits in DAC Control x register (DACCTRLx.[1:0]) will select an output buffer current.

44.6.3.3. Conversion Refresh

The DAC can only maintain its output within one LSB of the desired value for approximately 100µs. When a DAC is used to generate a static voltage or at a rate less than 20kSPS, the conversion must be refreshed periodically. The OSCULP32K clock can start new conversions automatically after a specified period. Write a value to the Refresh bit field in the DAC Control x register (DACCTRLx.REFRESH[3:0]) to select the refresh period according to the formula:

$$T_{\mathrm{REFRESH}} = \mathrm{REFRESH} \times T_{\mathrm{OSCULP32K}}$$

The actual period will depend on the tolerance of the OSCULP32K (see Electrical Characteristics).

If DACCTRLx.REFRESH=0, there is no conversion refresh. DACCTRLx.REFRESH=1 is Reserved.

If no new conversion is started before the refresh period is completed, DACx will convert the DATAx value again.

In standby sleep mode, the refresh mode remains enabled if DACCTRLx.RUNSTDBY=1.

If DATAx is written while a refresh conversion is ongoing, the conversion of the new content of DATAx is postponed until DACx is ready to start the next conversion.

Related Links

Electrical Characteristics on page 1140

44.6.3.4. Differential Mode

DAC0 and DAC1 can be configured to operate in differential mode, i.e. the combined output is a voltage balanced around VREF/2, see also the figure below.

In differential mode, DAC0 and DAC1 are converting synchronously the DATA0 value. DATA0 must therefore be a signed value, represented in two's complement format with DATA0[11] as the signed bit. DATA0 has therefore the range [-2047:2047].

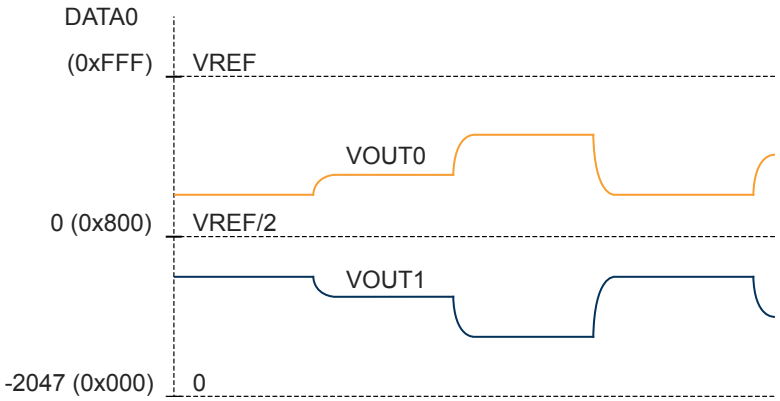
VOUT0 is the positive output and VOUT1 the negative output. The differential output voltage is therefore:

$$V_{\mathrm{OUT}} = \frac{\mathrm{DATA0}}{2047} \times \mathrm{VREF} = \left(V_{\mathrm{OUT0}} - V_{\mathrm{OUT1}}\right)$$

DACCTRL0 serves as the configuration register for both DAC0 and DAC1. Therefore DACCTRL1 does not need to be written.

The differential mode is enabled by writing a '1' to the Differential bit in the Control B register (CTRLB.DIFF).

Figure 44-4. DAC Conversions in Differential Mode



44.6.3.5. Dithering Mode

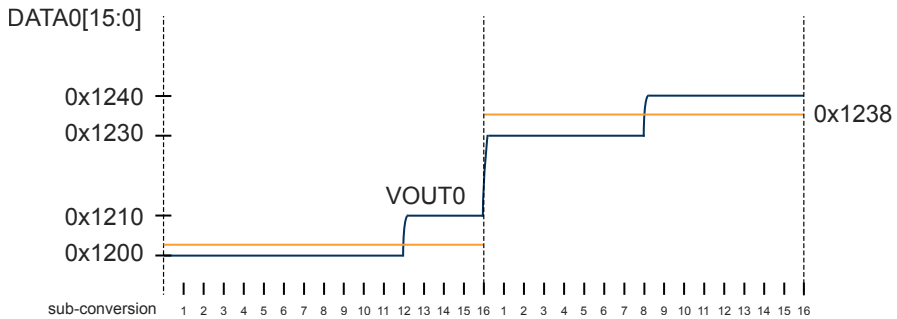
In dithering mode, DATAx is a 16-bit signed value where DATAx[15:4] is the 12-bit data converted by DAC and DATAx[3:0] represent the dither bits, used to minimize the quantization error.

The principle is to make 16 sub-conversions of the DATAx[15:4] value or the (DATAx[15:4] + 1) value, so that by averaging those two values, the conversion result of the 16-bit value (DATAx[15:0]) s accurate.

To operate, the STARTx event must be configured to generate 16 events for each DATAx[15:0] conversion, and DATABUFx must be loaded every 16 DAC conversions. EMPTYx event and DMA request are therefore generated every 16 DATABUFx to DATAx transfer. STATUS.EOCx still reports end of each sub-conversions.

Following timing diagram shows examples with DATA0[15:0] = 0x1204 followed by DATA0[15:0] = 0x1238.

Figure 44-5. DAC Conversions in Dithering Mode



44.6.4. Operating Conditions

- The DAC voltage reference must be below VDDANA.
- The maximum conversion rate of 1MSPS can be achieved only if VDDANA is above 2.4V.
- The frequency of GCLK_DAC must be equal or lower than 12MHz (corresponding to 1MSPS).

44.6.5. DMA Operation

In single mode (CTRLB.DIFF=0), DAC Controller generates the following DMA requests:

- Data Buffer 0 Empty (EMPTY0): The request is set when data is transferred from DATABUF0 or DATA0 to the internal data buffer of DAC0. The request is cleared when either DATA0 register or DATABUF0 register is written, or by writing a '1' to the EMPTY0 bit in the Interrupt Flag register (INTFLAG.EMPTY0).

- Data Buffer 1 Empty (EMPTY1): The request is set when data is transferred from DATABUF1 or DATA1 to the internal data buffer of DAC1. The request is cleared when either DATA0 register or DATABUF1 register is written, or by writing a one to the EMPTY1 bit in the Interrupt Flag register (INTFLAG.EMPTY1).

In differential mode (CTRLB.DIFF=1), DAC Controller generates the following DMA request:

- Data Buffer 0 Empty (EMPTY0): The request is set when data is transferred from DATABUF0 or DATA0 to the internal data buffer of DAC1. The request is cleared when either DATA0 register or DATABUF0 register is written, or by writing a one to the EMPTY0 bit in the Interrupt Flag register (INTFLAG.EMPTY0).

If the CPU accesses the registers which are source of DMA request set/clear condition, the DMA request can be lost or the DMA transfer can be corrupted, if enabled.

44.6.6. Interrupts

The DAC Controller has the following interrupt sources:

- DAC0 Data Buffer Empty (EMPTY0): Indicates that the internal data buffer of DAC0 is empty.
- DAC1 Data Buffer Empty (EMPTY1): Indicates that the internal data buffer of DAC1 is empty.
- DAC0 Underrun (UNDERRUN0): Indicates that the internal data buffer of DAC0 is empty and a DAC0 start of conversion event occurred. Refer to Events on page 1104 for details.
- DAC1 Underrun (UNDERRUN1): Indicates that the internal data buffer of DAC1 is empty and a DAC1 start of conversion event occurred. Refer to Events on page 1104 for details.

These interrupts are asynchronous wake-up sources. See *Sleep Mode Controller* for details.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register.

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the DAC Controller is reset. See INTFLAG on page 1122 for details on how to clear interrupt flags.

All interrupt requests from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. Refer to *Nested Vector Interrupt Controller* for details. The user must read the INTFLAG register to determine which interrupt condition is present.

Note that interrupts must be globally enabled for interrupt requests to be generated. Refer to *Nested Vector Interrupt Controller* for details.

44.6.7. Events

The DAC Controller can generate the following output events:

- Data Buffer 0 Empty (EMPTY0): Generated when the internal data buffer of DAC0 is empty. Refer to DMA Operation on page 1109 for details.
- Data Buffer 1 Empty (EMPTY1): Generated when the internal data buffer of DAC1 is empty. Refer to DMA Operation on page 1109 for details.

Writing a '1' to an Event Output bit in the Event Control Register (EVCTRL.EMPTYEOx) enables the corresponding output event. Writing a '0' to this bit disables the corresponding output event. Refer to the Event System chapter for details on configuring the event system.

The DAC Controller can take the following actions on an input event:

- DAC0 Start Conversion (START0): DATABUF0 value is transferred into DATA0 as soon as DAC0 is ready for the next conversion, and then conversion is started. START0 is considered as asynchronous to GCLK_DAC, thus it is resynchronized in the DAC Controller. Refer to Digital to Analog Conversion on page 1106 for details.
- DAC1 Start Conversion (START1): DATABUF1 value is transferred into DATA1 as soon as DAC1 is ready for the next conversion, and then conversion is started. START1 is considered as asynchronous to GCLK_DAC, thus it is resynchronized in the DAC Controller. Refer to Digital to Analog Conversion on page 1106 for details.

Writing a '1' to an Event Input bit in the Event Control register (EVCTRL.STARTEIx) enables the corresponding action on input event. Writing a '0' to this bit will disable the corresponding action on input event.

Note: When several events are connected to the DAC Controller, the enabled action will be taken on any of the incoming events. Refer to *EVSYS – Event System* for details on configuring the event system.

By default, DAC Controller detects rising edge events. Falling edge detection can be enabled by writing '1' to EVCTRL.INVEIx.

Related Links

EVSYS – Event System on page 536

44.6.8. Sleep Mode Operation

If the Run In Standby bit in the DAC Control x register DACCCTRLx.RUNSTDBY=1, the DACx will continue the conversions in standby sleep mode.

If DACCCTRLx.RUNSTDBY=0, the DACx will stop conversions in standby sleep mode.

If DACx conversion is stopped in standby sleep mode, DACx is also disabled to reduce power consumption. When exiting standby sleep mode, DACx is enabled again, therefore a certain startup time is required before starting a new conversion.

DAC Controller is compatible with SleepWalking: if RUNSTDBY=1, when an input event (STARTx) is detected in sleep mode, the DAC Controller will request GCLK_DAC in order to complete the conversion.

44.6.9. Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read. An exception is the Channel Enable bit in the Peripheral Channel Control registers (PCHCTRLm.CHEN). When changing this bit, the bit value must be read-back to ensure the synchronization is complete and to assert glitch free internal operation. Note that changing the bit value under ongoing synchronization will *not* generate an error.

The following bits are synchronized when written:

- Software Reset bit in control register (CTRLA.SWRST)
- Enable bit in control register (CTRLA.ENABLE)

The following registers are synchronized when written:

- DAC0 data register (DATA0)
- DAC1 data register (DATA1)
- DAC0 data buffer register (DATABUF0)
- DAC1 data buffer register (DATABUF1)

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

Related Links

Register Synchronization on page 127

44.7. Register Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0							ENABLE	SWRST
0x01	CTRLB	7:0						REFSEL[1:0]		DIFF
0x02	EVCTRL	7:0			INVEI1	INVEI0	EMPTYEO1	EMPTYEO0	STARTEI1	STARTEI0
0x03	Reserved									
0x04	INTENCLR	7:0					EMPTY1	EMPTY0	UNDERRUN1	UNDERRUN0
0x05	INTENSET	7:0					EMPTY1	EMPTY0	UNDERRUN1	UNDERRUN0
0x06	INTFLAG	7:0					EMPTY1	EMPTY0	UNDERRUN1	UNDERRUN0
0x07	STATUS	7:0					EOC1	EOC0	READY1	READY0
0x08	SYNCBUSY	7:0			DATABUF1	DATABUF0	DATA1	DATA0	ENABLE	SWRST
0x09		15:8								
0x0A		23:16								
0x0B		31:24								
0x0C	DACCTRL0	7:0	DITHER	RUNSTDBY			CCTRL[1:0][1:0]		ENABLE	LEFTADJ
0x0D		15:8					REFRESH[3:0]			
0x0E	DACCTRL1	7:0	DITHER	RUNSTDBY			CCTRL[1:0]		ENABLE	LEFTADJ
0x0F		15:8					REFRESH[3:0]			
0x10	DATA0	7:0	DATA[7:0]							
0x11		15:8	DATA[15:8]							
0x12	DATA1	7:0	DATA[7:0]							
0x13		15:8	DATA[15:8]							
0x14	DATABUF0	7:0	DATABUF[7:0]							
0x15		15:8	DATABUF[15:8]							
0x16	DATABUF1	7:0	DATABUF[7:0]							
0x17		15:8	DATABUF[15:8]							
0x18	DBGCTRL	7:0								DBGRUN

44.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection on page 1104.

Some registers are synchronized when read and/or written. Synchronization is denoted by the "Write-Synchronized" or the "Read-Synchronized" property in each individual register description. For details, refer to Synchronization on page 1111.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

44.8.1. Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
							ENABLE	SWRST
Access							R/W	R/W
Reset							0	0

Bit 1 – ENABLE: Enable DAC Controller
Due to synchronization there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the corresponding bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

Value	Description
0	The peripheral is disabled.
1	The peripheral is enabled.

Bit 0 – SWRST: Software Reset
Writing '0' to this bit has no effect.

Writing '1' to this bit resets all registers in the DAC to their initial state, and the DAC will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.

Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

44.8.2. Control B

Name: CTRLB
Offset: 0x01
Reset: 0x02
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
						REFSEL[1:0]		DIFF
Access						R/W	R/W	R/W
Reset						0	1	0

Bits 2:1 – REFSEL[1:0]: Reference Selection
This bit field selects the Reference Voltage for both DACs.

Value	Name	Description
00	VREFAU	Unbuffered external voltage reference (not buffered in DAC, direct connection)
01	VDDANA	Voltage supply
10	VREFAB	Buffered external voltage reference (buffered in DAC)
11	INTREF	Internal bandgap reference

Bit 0 – DIFF: Differential Mode Enable
This bit defines the conversion mode for both DACs.

Value	Description
0	Single mode
1	Differential mode

44.8.3. Event Control

Name: EVCTRL
Offset: 0x02
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
			INVEI1	INVEI0	EMPTYEO1	EMPTYEO0	STARTEI1	STARTEI0
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 5 – INVEI1: Enable Inversion of DAC1 input event
This bit defines the edge detection of the input event for DAC1 (START1).

Value	Description
0	Rising edge.
1	Falling edge.

Bit 4 – INVEI0: Enable Inversion Data Buffer Empty Event Output DAC0
This bit defines the edge detection of the input event for DAC0 (START0).

Value	Description
0	Rising edge.
1	Falling edge.

Bit 3 – EMPTYEO1: Data Buffer Empty Event Output DAC1
This bit indicates if the Data Buffer Empty Event output for DAC1 is enabled.

Value	Description
0	Data Buffer Empty event is disabled.
1	Data Buffer Empty event is enabled.

Bit 2 – EMPTYEO0: Data Buffer Empty Event Output DAC0
This bit indicates if the Data Buffer Empty Event output for DAC0 is enabled.

Value	Description
0	Data Buffer Empty event is disabled.
1	Data Buffer Empty event is enabled.

Bit 1 – STARTEI1: Start Conversion Event Input DAC1
This bit indicates if the Start input event for DAC1 is enabled.

Value	Description
0	A new conversion will not be triggered on any incoming event.
1	A new conversion will be triggered on any incoming event.

Bit 0 – STARTEI0: Start Conversion Event Input DAC0

This bit indicates if the Start input event for DAC0 is enabled.

Value	Description
0	A new conversion will not be triggered on any incoming event.
1	A new conversion will be triggered on any incoming event.

44.8.4. Interrupt Enable Clear

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Name: INTENCLR
Offset: 0x04
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
					EMPTY1	EMPTY0	UNDERRUN1	UNDERRUN0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 3 – EMPTY1: Data Buffer 1 Empty Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Data Buffer 1 Empty Interrupt Enable bit, which disables the Data Buffer 1 Empty interrupt.

Value	Description
0	The Data Buffer 1 Empty interrupt is disabled.
1	The Data Buffer 1 Empty interrupt is enabled.

Bit 2 – EMPTY0: Data Buffer 0 Empty Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Data Buffer 0 Empty Interrupt Enable bit, which disables the Data Buffer 0 Empty interrupt.

Value	Description
0	The Data Buffer 0 Empty interrupt is disabled.
1	The Data Buffer 0 Empty interrupt is enabled.

Bit 1 – UNDERRUN1: Underrun Interrupt Enable for DAC1
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Data Buffer 1 Underrun Interrupt Enable bit, which disables the Data Buffer 1 Underrun interrupt.

Value	Description
0	The Data Buffer 1 Underrun interrupt is disabled.
1	The Data Buffer 1 Underrun interrupt is enabled.

Bit 0 – UNDERRUN0: Underrun Interrupt Enable for DAC0
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Data Buffer 0 Underrun Interrupt Enable bit, which disables the Data Buffer 0 Underrun interrupt.

Value	Description
0	The Data Buffer 0 Underrun interrupt is disabled.
1	The Data Buffer 0 Underrun interrupt is enabled.

44.8.5. Interrupt Enable Set

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Name: INTENSET
Offset: 0x05
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
					EMPTY1	EMPTY0	UNDERRUN1	UNDERRUN0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 3 – EMPTY1: Data Buffer 1 Empty Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Data Buffer 1 Empty Interrupt Enable bit, which enables the Data Buffer 1 Empty interrupt.

Value	Description
0	The Data Buffer 1 Empty interrupt is disabled.
1	The Data Buffer 1 Empty interrupt is enabled.

Bit 2 – EMPTY0: Data Buffer 0 Empty Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Data Buffer 0 Empty Interrupt Enable bit, which enables the Data Buffer 0 Empty interrupt.

Value	Description
0	The Data Buffer 0 Empty interrupt is disabled.
1	The Data Buffer 0 Empty interrupt is enabled.

Bit 1 – UNDERRUN1: Underrun Interrupt Enable for DAC1
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Data Buffer 1 Underrun Interrupt Enable bit, which enables the Data Buffer 1 Underrun interrupt.

Value	Description
0	The Data Buffer 1 Underrun interrupt is disabled.
1	The Data Buffer 1 Underrun interrupt is enabled.

Bit 0 – UNDERRUN0: Underrun Interrupt Enable for DAC0
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Data Buffer 0 Underrun Interrupt Enable bit, which enables the Data Buffer 0 Underrun interrupt.

Value	Description
0	The Data Buffer 0 Underrun interrupt is disabled.
1	The Data Buffer 0 Underrun interrupt is enabled.

44.8.6. Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x06
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
					EMPTY1	EMPTY0	UNDERRUN1	UNDERRUN0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 3 – EMPTY1: Data Buffer 1 Empty
This flag is cleared by writing a '1' to it or by writing new data to DATA1 or DATABUF1.

This flag is set when the data buffer for DAC1 is empty and will generate an interrupt request if INTENCLR/INTENSET.EMPTY1=1.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Data Buffer 1 Empty interrupt flag.

Bit 2 – EMPTY0: Data Buffer 0 Empty
This flag is cleared by writing a '1' to it or by writing new data to DATA0 or DATABUF0.

This flag is set when the data buffer for DAC0 is empty and will generate an interrupt request if INTENCLR/INTENSET.EMPTY0=1.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Data Buffer 0 Empty interrupt flag.

Bit 1 – UNDERRUN1: DAC1 Underrun
This flag is cleared by writing a '1' to it.

This flag is set when a start conversion event (START1) occurred before new data is copied/written to the DAC1 data buffer and will generate an interrupt request if INTENCLR/INTENSET.UNDERRUN1=1.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the DAC1 Underrun interrupt flag.

Bit 0 – UNDERRUN0: DAC0 Underrun
This flag is cleared by writing a '1' to it.

This flag is set when a start conversion event (START0) occurred before new data is copied/written to the DAC) data buffer and will generate an interrupt request if INTENCLR/INTENSET.UNDERRUN0=1.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the DAC0 Underrun interrupt flag.

44.8.7. Status

Name: STATUS
Offset: 0x07
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
					EOC1	EOC0	READY1	READY0
Access					R	R	R	R
Reset					0	0	0	0

Bit 3 – EOC1: DAC1 End of Conversion
This bit is cleared when DATA1 register is written.

Value	Description
0	No conversion completed since last load of DATA1.
1	DAC1 conversion is complete, VOUT1 is stable.

Bit 2 – EOC0: DAC0 End of Conversion
This bit is cleared when DATA0 register is written.

Value	Description
0	No conversion completed since last load of DATA0.
1	DAC0 conversion is complete, VOUT0 is stable.

Bit 1 – READY1: DAC1 Startup Ready

Value	Description
0	DAC1 is not ready for conversion.
1	Startup time has elapsed, DAC1 is ready for conversion.

Bit 0 – READY0: DAC0 Startup Ready

Value	Description
0	DAC0 is not ready for conversion.
1	Startup time has elapsed, DAC0 is ready for conversion.

44.8.8. Synchronization Busy

Name: SYNCBUSY
Offset: 0x08
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			DATABUF1	DATABUF0	DATA1	DATA0	ENABLE	SWRST
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit 5 – DATABUF1: Data Buffer DAC1
This bit is set when DATABUF1 register is written.

This bit is cleared when DATABUF1 synchronization is completed.

Value	Description
0	No ongoing synchronized access.
1	Synchronized access is ongoing.

Bit 4 – DATABUF0: Data Buffer DAC0
This bit is set when DATABUF0 register is written.

This bit is cleared when DATABUF0 synchronization is completed.

Value	Description
0	No ongoing synchronized access.
1	Synchronized access is ongoing.

Bit 3 – DATA1: Data DAC1
This bit is set when DATA1 register is written.

This bit is cleared when DATA1 synchronization is completed.

Value	Description
0	No ongoing synchronized access.
1	Synchronized access is ongoing.

Bit 2 – DATA0: Data DAC0
This bit is set when DATA0 register is written.

This bit is cleared when DATA0 synchronization is completed.

Value	Description
0	No ongoing synchronized access.
1	Synchronized access is ongoing.

Bit 1 – ENABLE: DAC Enable Status
This bit is set when CTRLA.ENABLE bit is written.

This bit is cleared when CTRLA.ENABLE synchronization is completed.

Value	Description
0	No ongoing synchronization.
1	Synchronization is ongoing.

Bit 0 – SWRST: Software Reset
This bit is set when CTRLA.SWRST bit is written.

This bit is cleared when CTRLA.SWRST synchronization is completed.

Value	Description
0	No ongoing synchronization.
1	Synchronization is ongoing.

44.8.9. DAC0 Control

Name: DACCTRL0
Offset: 0x0C
Reset: 0x0000
Property: PAC Write-Protection, Enabled-Protected

Bit	15	14	13	12	11	10	9	8
					REFRESH[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DITHER	RUNSTDBY			CCTRL[1:0][1:0]		ENABLE	LEFTADJ
Access	R/W	R/W			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

Bits 11:8 – REFRESH[3:0]: Refresh period
This field defines the refresh period.

Value	Description
0x0	Refresh is disabled.
0x1	Reserved
0x2 to 0xF	$T_{\mathrm{REFRESH}} = \mathrm{REFRESH} \times 30.52\mu s$

Bit 7 – DITHER: Dithering Mode

Value	Description
0	Dithering mode is disabled.
1	Dithering mode is enabled.

Bit 6 – RUNSTDBY: Run in Standby
This bit controls the behavior of DAC0 during standby sleep mode.

Value	Description
0	DAC0 is disabled during standby sleep mode.
1	DAC0 continues to operate during standby sleep mode.

Bits 3:2 – CCTRL[1:0][1:0]: Current Control
This field defines the current in output buffer according to conversion rate.

Figure 44-6. Current Control

Value	Name	Description
0x0	CC100K	GCLK_DAC ≤ 1.2MHz (100kSPS)
0x1	CC1M	1.2MHz < GCLK_DAC ≤ 6MHz (500kSPS)

Value	Name	Description
0x2	CC12M	6MHz < GCLK_DAC ≤ 12MHz (1MSPS)
0x3	Reserved	

Bit 1 – ENABLE: Enable DAC0
This bit enables DAC0 when DAC Controller is enabled (CTRLA.ENABLE).

Value	Description
0	DAC0 is disabled.
1	DAC0 is enabled.

Bit 0 – LEFTADJ: Left Adjusted Data
This bit controls how the 12-bit conversion data is adjusted in the Data and Data Buffer registers.

Value	Description
0	DATA0 and DATABUF0 registers are right-adjusted.
1	DATA0 and DATABUF0 registers are left-adjusted.

44.8.10. DAC1 Control

Name: DACCTRL1
Offset: 0x0E
Reset: 0x0000
Property: PAC Write-Protection, Enabled-Protected

Bit	15	14	13	12	11	10	9	8
					REFRESH[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DITHER	RUNSTDBY			CCTRL[1:0]		ENABLE	LEFTADJ
Access	R/W	R/W			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

Bits 11:8 – REFRESH[3:0]: Refresh period
This field defines the refresh period.

Value	Description
0x0	Refresh is disabled.
0x1	Reserved
0x2 to 0xF	$T_{\text{REFRESH}} = \text{REFRESH} \times 30.52\mu s$

Bit 7 – DITHER: Dithering Mode

Value	Description
0	Dithering mode is disabled.
1	Dithering mode is enabled.

Bit 6 – RUNSTDBY: Run in Standby
This bit controls the behavior of DAC1 during standby sleep mode.

Value	Description
0	DAC1 is disabled during standby sleep mode.
1	DAC1 continues to operate during standby sleep mode.

Bits 3:2 – CCTRL[1:0]: Current Control
This field defines the current in output buffer.

Figure 44-7. Current Control

Value	Name	Description
0x0	CC100K	GCLK_DAC <= 1.2MHz (100kSPS)
0x1	CC1M	1.2MHz < GCLK_DAC <= 6MHz (500kSPS)

Value	Name	Description
0x2	CC12M	6MHz < GCLK_DAC <= 12MHz (1MSPS)
0x3		Reserved

Bit 1 – ENABLE: Enable DAC1
This bit enables DAC1 when DAC Controller is enabled (CTRLA.ENABLE).

Value	Description
0	DAC1 is disabled.
1	DAC1 is enabled.

Bit 0 – LEFTADJ: Left Adjusted Data
This bit controls how the 12-bit conversion data is adjusted in the Data and Data Buffer registers.

Value	Description
0	DATA1 and DATABUF1 registers are right-adjusted.
1	DATA1 and DATABUF1 registers are left-adjusted.

44.8.11. Data DAC0

Name: DATA0
Offset: 0x10
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – DATA[15:0]: DAC0 Data
DATA0 register contains the 12-bit value that is converted to a voltage by the DAC0. The adjustment of these 12 bits within the 16-bit register is controlled by DACCTRL0.LEFTADJ:

- DATA[11:0] when DACCTRL0.LEFTADJ=0.

- DATA[15:4] when DACCTRL0.LEFTADJ=1.

In dithering mode (whatever DACCTRL0.LEFTADJ value):

- DATA[15:4] are the 12-bit converted by DAC0.

- DATA[3:0] are the dither bits.

44.8.12. Data DAC1

Name: DATA1
Offset: 0x12
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – DATA[15:0]: DAC1 Data
DATA1 register contains the 12-bit value that is converted to a voltage by the DAC1. The adjustment of these 12 bits within the 16-bit register is controlled by DACCTRL1.LEFTADJ:

- DATA[11:0] when DACCTRL1.LEFTADJ=0.

- DATA[15:4] when DACCTRL1.LEFTADJ=1.

In dithering mode (whatever DACCTRL1.LEFTADJ value):

- DATA[15:4] are the 12-bit converted by DAC1.

- DATA[3:0] are the dither bits.

44.8.13. Data Buffer DAC0

Name: DATABUF0
Offset: 0x14
Reset: 0x0000
Property: Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	DATABUF[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DATABUF[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – DATABUF[15:0]: DAC0 Data Buffer
DATABUF0 contains the value to be transferred into DATA0 when a START0 event occurs.

44.8.14. Data Buffer DAC1

Name: DATABUF1
Offset: 0x16
Reset: 0x0000
Property: Write-Synchronized

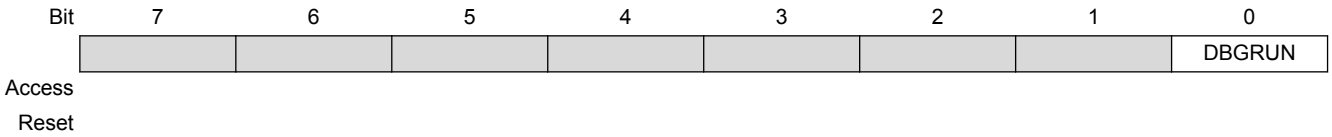
Bit	15	14	13	12	11	10	9	8
	DATABUF[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DATABUF[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – DATABUF[15:0]: DAC1 Data Buffer
DATABUF1 contains the value to be transferred into DATA1 when a START1 event occurs.

44.8.15. Debug Control

Name: DBGCTRL
Offset: 0x18
Reset: 0x00
Property: PAC Write-Protection



Bit 0 – DBGRUN: Debug Run
This bit is not reset by a software reset.

This bits controls the functionality when the CPU is halted by an external debugger.

Value	Description
0	The DAC is halted when the CPU is halted by an external debugger. Any ongoing conversion will complete.
1	The DAC continues normal operation when the CPU is halted by an external debugger.

45. PTC - Peripheral Touch Controller

45.1. Overview

The Peripheral Touch Controller (PTC) acquires signals in order to detect touch on capacitive sensors. The external capacitive touch sensor is typically formed on a PCB, and the sensor electrodes are connected to the analog front end of the PTC through the I/O pins in the device. The PTC supports both self- and mutual-capacitance sensors.

In mutual-capacitance mode, sensing is done using capacitive touch matrices in various X-Y configurations, including indium tin oxide (ITO) sensor grids. The PTC requires one pin per X-line and one pin per Y-line.

In self-capacitance mode, the PTC requires only one pin (Y-line) for each touch sensor.

45.2. Features

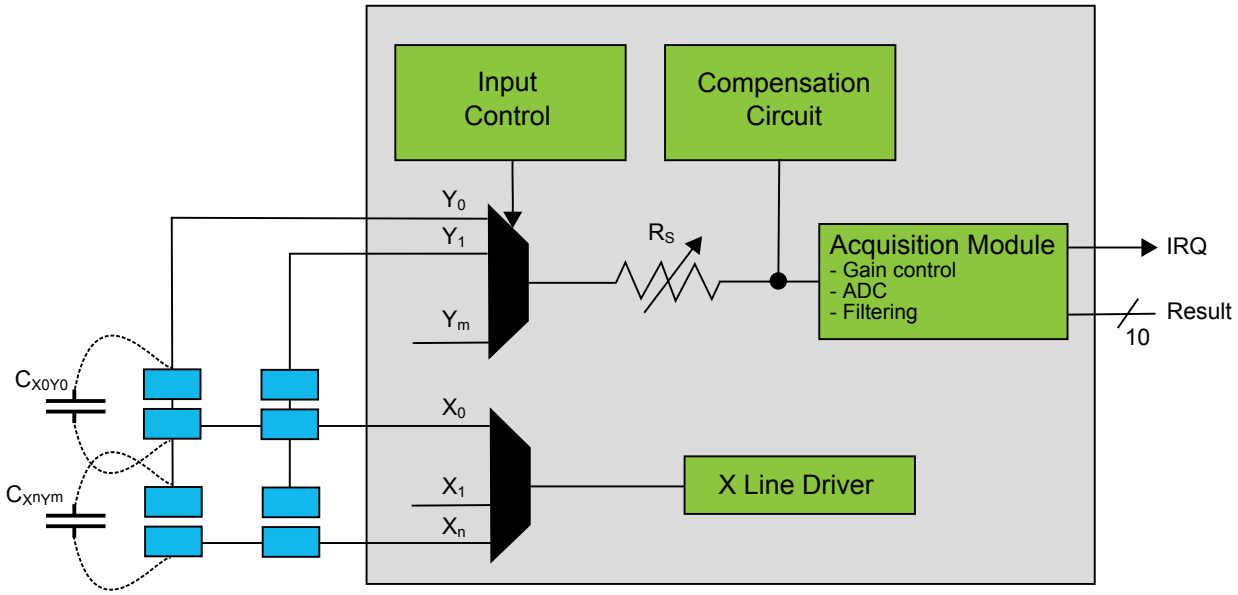
- Low-power, high-sensitivity, environmentally robust capacitive touch buttons, sliders, wheels and proximity sensing
- Supports mutual capacitance and self-capacitance sensing
  - Mix-and-match mutual-and self-capacitance sensors
- One pin per electrode – no external components
- Load compensating charge sensing
  - Parasitic capacitance compensation and adjustable gain for superior sensitivity
- Zero drift over the temperature and V_{DD} range
  - Auto calibration and re-calibration of sensors
- Single-shot and free-running charge measurement
- Hardware noise filtering and noise signal de-synchronization for high conducted immunity
- Selectable channel change delay
  - Allows choosing the settling time on a new channel, as required
- Acquisition-start triggered by command or through auto-triggering feature
- Low CPU utilization through interrupt on acquisition-complete
  - 5% CPU utilization scanning 10 channels at 50ms scan rate
- Supported by the Atmel® QTouch® Composer development tool, which comprises QTouch Library project builder and QTouch analyzer

Related Links

Configuration Summary on page 15
I/O Multiplexing and Considerations on page 29

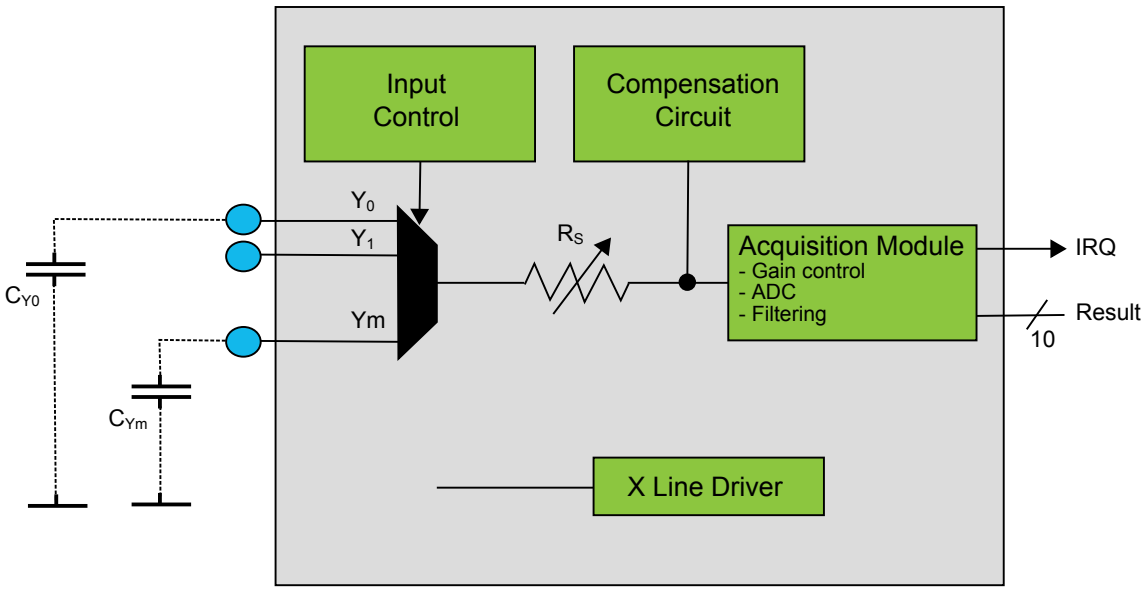
45.3. Block Diagram

Figure 45-1. PTC Block Diagram Mutual-Capacitance



Note: For SAM L21 the R_S = 20-100 KΩ.

Figure 45-2. PTC Block Diagram Self-Capacitance



Note: For SAM L21 the RS = 20-100 KΩ.

45.4. Signal Description

Name	Type	Description
X[n:0]	Digital	X-line (Output)
Y[m:0]	Analog	Y-line (Input/Output)

Note: The number of X and Y lines are device dependent. Refer to *Configuration Summary* for details.

Refer to *I/O Multiplexing and Considerations* for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

Related Links

Configuration Summary on page 15
I/O Multiplexing and Considerations on page 29

45.5. Product Dependencies

In order to use this Peripheral, configure the other components of the system as described in the following sections.

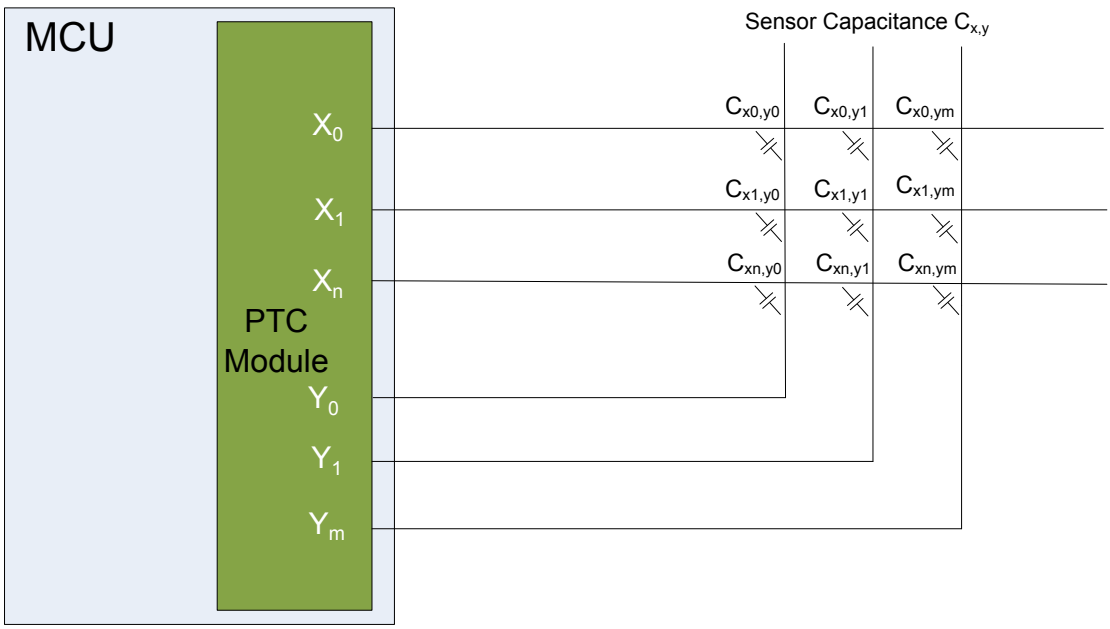
45.5.1. I/O Lines

The I/O lines used for analog X-lines and Y-lines must be connected to external capacitive touch sensor electrodes. External components are not required for normal operation. However, to improve the EMC performance, a series resistor of 1 KΩ can be used on X-lines and Y-lines.

45.5.1.1. Mutual-capacitance Sensor Arrangement

A mutual-capacitance sensor is formed between two I/O lines - an X electrode for transmitting and Y electrode for receiving. The mutual capacitance between the X and Y electrode is measured by the Peripheral Touch Controller.

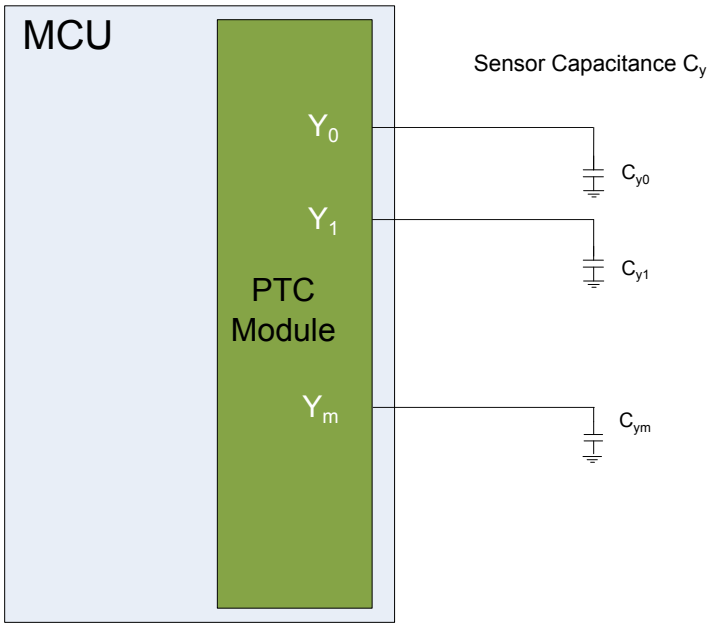
Figure 45-3. Mutual Capacitance Sensor Arrangement



45.5.1.2. Self-capacitance Sensor Arrangement

The self-capacitance sensor is connected to a single pin on the Peripheral Touch Controller through the Y electrode for receiving the signal. The sense electrode capacitance is measured by the Peripheral Touch Controller.

Figure 45-4. Self-capacitance Sensor Arrangement



For more information about designing the touch sensor, refer to Buttons, Sliders and Wheels Touch Sensor Design Guide on http://www.atmel.com.

45.5.2. Clocks

The PTC is clocked by the GCLK_PTC clock. The PTC operates from an asynchronous clock source and the operation is independent of the main system clock and its derivative clocks, such as the peripheral bus clock (CLK_APB). A number of clock sources can be selected as the source for the asynchronous GCLK_PTC. The clock source is selected by configuring the Generic Clock Selection ID in the Generic Clock Control register. For more information about selecting the clock sources, refer to *GCLK - Generic Clock Controller*.

The selected clock must be enabled in the Power Manager, before it can be used by the PTC. By default these clocks are disabled. The frequency range of GCLK_PTC is 400kHz to 4MHz.

For more details, refer to *PM – Power Manager*.

Related Links

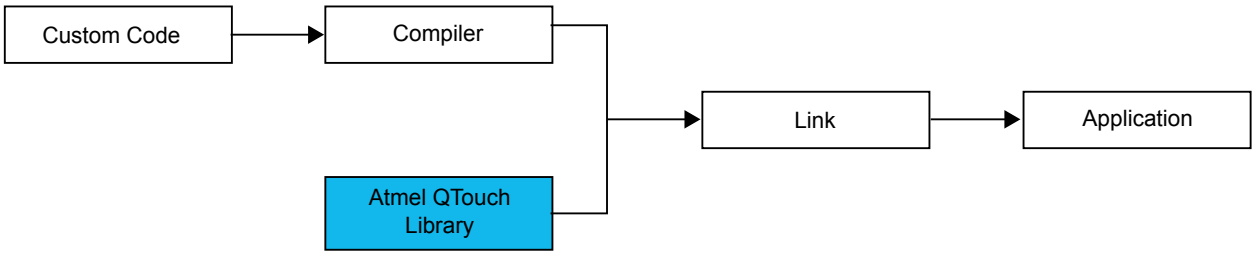
GCLK - Generic Clock Controller on page 131

PM – Power Manager on page 186

45.6. Functional Description

In order to access the PTC, the user must use the QTouch Composer tool to configure and link the QTouch Library firmware with the application code. QTouch Library can be used to implement buttons, sliders, wheels and proximity sensor in a variety of combinations on a single interface.

Figure 45-5. QTouch Library Usage



For more information about QTouch Library, refer to the Atmel QTouch Library Peripheral Touch Controller User Guide.

46. Electrical Characteristics

46.1. Disclaimer

All typical values are measured at T = 25°C unless otherwise specified. All minimum and maximum values are valid across operating temperature and voltage unless otherwise specified.

46.2. Absolute Maximum Ratings

Stresses beyond those listed in Table 46-1 Absolute Maximum Ratings on page 1140 may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Table 46-1. Absolute Maximum Ratings

Symbol	Description	Min.	Max.	Units
V_{DD}	Power supply voltage	0	3.8	V
I_{VDD}	Current into a V_{DD} pin	-	92(1)	mA
I_{GND}	Current out of a GND pin	-	130(1)	mA
V_{PIN}	Pin voltage with respect to GND and V_{DD}	GND-0.3V	VDD+0.3V	V
$T_{storage}$	Storage temperature	-60	150	°C

Note:
1. Maximum source current is 46mA and maximum sink current is 65mA per cluster. A cluster is a group of GPIOs as shown in section GPIO Clusters on page 32.

Also note that each VDD/GND pair is connected to two clusters, so current consumption through the pair will be a sum of the clusters' source/sink currents.

46.3. General Operating Ratings

The device must operate within the ratings listed in Table 46-2 General operating conditions on page 1140 in order for all other electrical characteristics and typical characteristics of the device to be valid.

Table 46-2. General operating conditions

Symbol	Description	Min.	Typ.	Max.	Units
V_{DDIN}	Power supply voltage	1.62	3.3	3.63	V
V_{DDIO}	IO Supply Voltage	1.62	3.3	3.63	V
V_{DDANA}	Analog supply voltage	1.62	3.3	3.63	V
T_A	Temperature range	-40	25	85	°C
T_J	Junction temperature	-	-	100	°C

46.4. Supply Characteristics

Table 46-3. Supply Characteristics

Symbol	Voltage		
	Min.	Max.	Units
V_{DDIO}	1.62	3.63	V
V_{DDIN}	1.62	3.63	V
V_{DDANA}	1.62	3.63	V
V_{BAT}	1.62	3.63	V

Table 46-4. Supply Slew Rates(1)

Symbol	Fall Rate	Rise Rate	Units
	Max.	Max.	
V_{DDIO}	0.05	0.1	V/µs
V_{DDIN}	0.05	0.1	V/µs
V_{DDANA}	0.05	0.1	V/µs
V_{BAT}	0.05	0.1	V/µs

Note: 1. These values are based on simulation. They are not covered by production test limits or characterization.

Related Links

Power Supply and Start-Up Considerations on page 37

46.5. Maximum Clock Frequencies

Table 46-5. Maximum GCLK Generator Output Frequencies(1)

Symbol	Description	Conditions	Fmax [MHz]		Units
			PL0	PL2	
$F_{gclkgen}[0:2]$	GCLK Generator output Frequency	-	24	96	Mhz
$F_{gclkgen}[3:8]$		undivided	24	96	Mhz
$F_{gclkgen}[3:8]$		divided	16	66	Mhz

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.

Table 46-6. Maximum Peripheral Clock Frequencies(1)

Symbol	Description	Conditions	Max.		Units
			PL0	PL2	
f_{CPU}	CPU clock frequency	-	12	48	MHz
f_{AHB}	AHB clock frequency	-	12	48	MHz
f_{APBA}	APBA clock frequency	Bus clock domain = BACKUP	6	6	MHz
f_{APBA}	APBA clock frequency	Bus clock domain = Low Power	12	48	MHz
f_{APBB}	APBB clock frequency	-	12	48	MHz
f_{APBC}	APBC clock frequency	-			
f_{APBD}	APBD clock frequency	-			
f_{APBE}	APBE clock frequency	-			
$f_{GCLK\_DFLL48M\_REF}$	DFLL48M Reference clock frequency	-	NA	33	MHz
f_{GCLK_DPLL}	FDPLL96M Reference clock frequency	-	2	2	MHz
$f_{GCLK\_DPLL\_32K}$	FDPLL96M 32k Reference clock frequency	-	32	100	kHz
f_{GCLK_EIC}	EIC input clock frequency	-	12	48	MHz
f_{GCLK_USB}	USB input clock frequency	-	NA	60	MHz
$f_{GCLK\_EVSYS\_CHANNEL\_0}$	EVSYS channel 0 input clock frequency	-	12	48	MHz
$f_{GCLK\_EVSYS\_CHANNEL\_1}$	EVSYS channel 1 input clock frequency	-			
$f_{GCLK\_EVSYS\_CHANNEL\_2}$	EVSYS channel 2 input clock frequency	-			
$f_{GCLK\_EVSYS\_CHANNEL\_3}$	EVSYS channel 3 input clock frequency	-			
$f_{GCLK\_EVSYS\_CHANNEL\_4}$	EVSYS channel 4 input clock frequency	-			
$f_{GCLK\_EVSYS\_CHANNEL\_5}$	EVSYS channel 5 input clock frequency	-			
$f_{GCLK\_EVSYS\_CHANNEL\_6}$	EVSYS channel 6 input clock frequency	-			
$f_{GCLK\_EVSYS\_CHANNEL\_7}$	EVSYS channel 7 input clock frequency	-			
$f_{GCLK\_EVSYS\_CHANNEL\_8}$	EVSYS channel 8 input clock frequency	-			
$f_{GCLK\_EVSYS\_CHANNEL\_9}$	EVSYS channel 9 input clock frequency	-			
$f_{GCLK\_EVSYS\_CHANNEL\_10}$	EVSYS channel 10 input clock frequency	-			
$f_{GCLK\_EVSYS\_CHANNEL\_11}$	EVSYS channel 11 input clock frequency	-			
$f_{GCLK\_SERCOMx\_SLOW}$	Common SERCOM slow input clock frequency	-	1	5	MHz

Symbol	Description	Conditions	Max.		Units
			PL0	PL2	
$f_{GCLK\_SERCOM0\_CORE}$	SERCOM0 input clock frequency	-	12	48	MHz
$f_{GCLK\_SERCOM1\_CORE}$	SERCOM1 input clock frequency	-			
$f_{GCLK\_SERCOM2\_CORE}$	SERCOM2 input clock frequency	-			
$f_{GCLK\_SERCOM3\_CORE}$	SERCOM3 input clock frequency	-			
$f_{GCLK\_SERCOM4\_CORE}$	SERCOM4 input clock frequency	-			
$f_{GCLK\_SERCOM5\_CORE}$	SERCOM5 input clock frequency	-			
$f_{GCLK\_TCC0,\ GCLK\_TCC1}$	TCC0,TCC1 input clock frequency	-	24	96	MHz
$f_{GCLK\_TCC2,\ GCLK\_TC0}$	TCC2,TC0 input clock frequency	-	12	48	MHz
$f_{GCLK\_TC1,\ GCLK\_TC2}$	TC1,TC2 input clock frequency	-			
$f_{GCLK\_TC3,\ GCLK\_TC4}$	TC3,TC4 input clock frequency	-			
f_{GCLK_ADC}	ADC input clock frequency	-	12	48	MHz
f_{GCLK_AC}	AC digital input clock frequency	-			
f_{GCLK_DAC}	DAC input clock frequency	-			
f_{GCLK_PTC}	PTC input clock frequency	-			
f_{GCLK_CCL}	CCL input clock frequency	-			

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.

46.6. Power Consumption

The values in this section are measured values of power consumption under the following conditions, except where noted:

- **Operating Conditions**
  - V_{VDDIN} = 3.3V or 1.8V
  - CPU is running on flash with 1 wait state for V_{DDIN}=3.3V, and 3 wait states for V_{DDIN}=1.8V
  - Low power cache is enabled
  - BOD33 is disabled
- **Oscillators**
  - XOSC (crystal oscillator) stopped
  - XOSC32K (32KHz crystal oscillator) running with external 32KHz crystal
  - When in active Performance Level 2 (PL2) mode, DFLL48M is running at 48MHz and using XOSC32K as reference
  - When in active PL0 mode, the internal Multi RC Oscillator is running at specified frequency
- **Clocks**
  - DFLL48M used as main clock source when in PL2 mode. In PL0 mode, OSC16M used at 4, 8, or 12MHz

- Clock masks and dividers at reset values: All AHB & APB clocks enabled, CPUDIV=1, BUPDIV=1, LPDIV=1
- I/Os are inactive input mode with input trigger disable

Table 46-7. Active Current Consumption(1)

Mode	Conditions	Regulator	PL	Clock	Vcc	Ta	Typ.	Max.	Units
ACTIVE	COREMARK	LDO	PL0	OSC 12MHz	1.8V	Max at 85°C Typ at 25°C	77	106	µA/MHz
					3.3V		79	101	
				OSC 8MHz	1.8V		80	111	
					3.3V		82	120	
				OSC 4MHz	1.8V		89	153	
					3.3V		92	166	
			PL2	DFLL 48MHz	1.8V		93	103	
					3.3V		95	105	
		BUCK	PL0	OSC 12MHz	1.8V		47	60	
					3.3V		32	43	
				OSC 8MHz	1.8V		50	71	
					3.3V		34	56	
				OSC 4MHz	1.8V		57	98	
					3.3V		42	81	
			PL2	DFLL 48MHz	1.8V		67	76	
					3.3V		40	45	
	FIBO	LDO	PL0	OSC 12MHz	1.8V		76	95	
					3.3V		78	98	
				OSC 8MHz	1.8V		79	111	
					3.3V		81	119	
				OSC 4MHz	1.8V		89	155	
					3.3V		91	173	
			PL2	DFLL 48MHz	1.8V		94	104	
					3.3V		95	104	
		BUCK	PL0	OSC 12MHz	1.8V		47	62	
					3.3V		31	42	
				OSC 8MHz	1.8V		50	71	
					3.3V		34	55	
				OSC 4MHz	1.8V		57	100	
					3.3V		42	88	
			PL2	DFLL 48MHz	1.8V		67	76	
					3.3V		40	45	

Mode	Conditions	Regulator	PL	Clock	Vcc	Ta	Typ.	Max.	Units
ACTIVE	WHILE1	LDO	PL0	OSC 12MHz	1.8V	Max at 85°C Typ at 25°C	60	80	µA/MHz
					3.3V		62	87	
				OSC 8MHz	1.8V		63	95	
					3.3V		65	106	
				OSC 4MHz	1.8V		72	138	
					3.3V		75	153	
			PL2	DFLL 48MHz	1.8V		73	80	
					3.3V		73	81	
		BUCK	PL0	OSC 12MHz	1.8V		38	51	
					3.3V		26	41	
				OSC 8MHz	1.8V		40	61	
					3.3V		29	51	
				OSC 4MHz	1.8V		47	90	
					3.3V		36	79	
			PL2	DFLL 48MHz	1.8V		52	59	
					3.3V		31	35	
IDLE			PL0	OSC 12MHz	1.8V		17	30	
					3.3V		13	24	

Note: 1. These values are based on characterization.

Table 46-8. Standby, Backup and Off Mode Current Consumption(1)

Mode	Conditions	Regulator Mode	Vcc	Ta	Typ.	Max.	Units
STANDBY	PD0, PD1, PD2 in retention state	LPEFF Disable	1.8V	25°C	1.3	3.1	µA
				85°C	23.9	60.8	
		LPEFF Enable	3.3V	25°C	1.2	2.7	
				85°C	20.7	45.8	
	PD0, PD1, PD2 in retention state with RTC running on OSCULP32K	LPEFF Disable	1.8V	25°C	1.5	3.2	
				85°C	24.0	60.8	
		LPEFF Enable	3.3V	25°C	1.4	2.9	
				85°C	20.8	45.9	
	PD1, PD2 in retention state and PD0 in active state	LPEFF Disable	1.8V	25°C	1.6	3.6	
				85°C	26.3	72.9	
		LPEFF Enable	3.3V	25°C	1.4	3.2	
				85°C	23.4	52.9	
	PD2 in retention state and PD0, PD1 in active state	LPEFF Disable	1.8V	25°C	2.3	5.6	
				85°C	54.7	126.2	
		LPEFF Enable	3.3V	25°C	2.0	4.8	
				85°C	36.7	82.0	
	PD0, PD1, PD2 in active state	LPEFF Disable	1.8V	25°C	3.3	7.9	
				85°C	83.9	185.5	
		LPEFF Enable	3.3V	25°C	2.8	6.3	
				85°C	52.4	147.7	

Mode	Conditions	Regulator Mode	Vcc	Ta	Typ.	Max.	Units
BACKUP	powered by VDDIN VDDIN+VDDANA+VDDIO consumption		1.8V	25°C	0.48	0.83	µA
				85°C	4.4	10.2	
			3.3V	25°C	0.59	1.1	
				85°C	5.9	14.0	
	powered by VDDIN VBAT consumption		1.8V	25°C	0.001	0.004	
				85°C	0.011	0.027	
			3.3V	25°C	0.007	0.026	
				85°C	0.036	0.076	
	powered by VDDIN with RTC running VDDIN+VDDANA+VDDIO consumption		1.8V	25°C	0.53	0.89	
				85°C	4.5	10.2	
			3.3V	25°C	0.64	1.2	
				85°C	5.9	14.1	
	powered by VDDIN with RTC running VBAT consumption		1.8V	25°C	0.001	0.005	
				85°C	0.011	0.028	
			3.3V	25°C	0.008	0.018	
				85°C	0.027	0.065	
	powered by VBAT VDDIN+VDDANA+VDDIO consumption		1.8V	25°C	0.49	0.85	
				85°C	4.5	10.2	
			3.3V	25°C	0.59	1.1	
				85°C	5.9	14.1	
	powered by VBAT VBAT consumption		1.8V	25°C	0.37	0.60	
				85°C	2.0	4.9	
			3.3V	25°C	0.40	0.64	
				85°C	2.1	5.1	
	powered by VBAT with RTC running VDDIN+VDDANA+VDDIO consumption		1.8V	25°C	0.56	0.91	
				85°C	4.5	10.3	
			3.3V	25°C	0.65	1.2	
				85°C	6.0	14.2	
	powered by VBAT with RTC running VBAT consumption		1.8V	25°C	0.42	0.66	
				85°C	2.1	5.0	
			3.3V	25°C	0.45	0.70	
				85°C	2.2	5.2	

Mode	Conditions	Regulator Mode	Vcc	Ta	Typ.	Max.	Units
OFF			1.8V	25°C	0.16	0.29	µA
				85°C	2.6	5.5	
			3.3V	25°C	0.21	0.51	
				85°C	3.9	9.3	

Note: 1. These values are based on characterization.

46.7. Wake-Up Time

Conditions:

- V_{DDIN} = 3.3V
- LDO Regulation mode
- CPU clock = OSC16M @12Mhz
- 1 Wait-state
- Cache enabled
- Flash Fast Wake up enabled (NVMCTRL.CTRLB.FWUP=1)
- Flash in WAKEUPINSTANT mode (NVMCTRL.CTRLB.SLEEPPRM=1)

For IDLE and STANDBY, CPU sets an IO by writing PORT->IOBUS without jumping in an interrupt handler (Cortex M0+ register PRIMASK=1). The wake-up time is measured between the edge of the wake-up input signal and the edge of the GPIO pin.

For Backup, the exit of mode is done through RTC wake-up. The wake-up time is measured between the toggle of the RTC pin and the set of the IO done by the first executed instructions after reset.

For OFF mode, the exit of mode is done through reset pin, the time is measured between the rising edge of the RESETN signal and the set of the IO done by the first executed instructions after reset.

Table 46-9. Wake-Up Timing

Sleep Mode	Condition		Typ.	Unit
IDLE	PL2 or PL0		1.2	µs
STANDBY	PL0 PM.PLSEL.PLDIS=1 (see errata 13674 for revision A)	PDCFG default	5.1	µs
		PD012 forced active PDCFG=3	2.1	
	PL2 Voltage scaling at default values: SUPC->VREG.VSVSTEP=0 SUPC->VREG.VSPER=0	PDCFG default	76	µs
		PD012 forced active PDCFG=3	75	
	PL2 Voltage scaling at fastest setting: SUPC->VREG.VSVSTEP=15 SUPC->VREG.VSPER=0	PDCFG default	16	µs
		PD012 forced active PDCFG=3	15	µs
BACKUP			90	µs
OFF			2200	µs

46.8. I/O Pin Characteristics

There are three different pin types with three different speeds: Backup, Normal, and High Sink[3,4].

The Drive Strength bit is located in the Pin Configuration register of the PORT (PORT.PINCFG.DRVSTR).

Table 46-10. I/O Pins Common Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
V_{IL}	Input low-level voltage	V_{DD}=1.62V-2.7V	-	-	0.25*V_{DD}	V
		V_{DD}=2.7V-3.63V	-	-	0.3*V_{DD}	
V_{IH}	Input high-level voltage	V_{DD}=1.62V-2.7V	0.7*V_{DD}	-	-	
		V_{DD}=2.7V-3.63V	0.55*V_{DD}	-	-	
V_{OL}	Output low-level voltage	V_{DD}>1.6V, I_{OL} max	-	0.1*V_{DD}	0.2*V_{DD}	
V_{OH}	Output high-level voltage	V_{DD}>1.6V, I_{OH} max	0.8*V_{DD}	0.9*V_{DD}	-	
R_{PULL}	Pull-up - Pull-down resistance	All pins except PA24, PA25	20	40	60	kΩ
		PA24, PA25(1)	50	100	150	
I_{LEAK}	Input leakage current	Pull-up resistors disabled	-1	+/-0.015	1	µA

Table 46-11. I/O Pins Maximum Output Current

Symbol	Parameter	Conditions	Backup Pins in Backup Mode(4)	Backup and Normal Pins(4)	High Sink Pins(3)	Backup and Normal Pins(4)	High Sink Pins(3)	Units
				DRVSTR=0		DRVSTR=1		
I_{OL}	Maximum Output low-level current	V_{DD}=1.62V-3V	0.005	1	2	2	4	mA
		V_{DD}=3V-3.63V	0.008	2.5	6	6	12	
I_{OH}	Maximum Output high-level current	V_{DD}=1.62V-3V	0.005	0.7	1.5	1.5	3	
		V_{DD}=3V-3.63V	0.008	2	5	5	10	

Table 46-12. I/O Pins Dynamic Characteristics(1)

Symbol	Parameter	Conditions	Backup Pins in Backup Mode(4)	Backup and Normal Pins(4)	High Sink Pins(3)	Backup and Normal Pins(4)	High Sink Pins(3)	Units
				DRVSTR=0		DRVSTR=1		
t_{RISE}	Maximum Rise time	V_{DD}=3.3V, load=20pF	2000	13	6	6	4.5	ns
t_{FALL}	Maximum Fall time	V_{DD}=3.3V, load=20pF	2000	12	7	7	4.5	

The pins with I^2C alternative mode available are compliant(2) with I^2C norms. All I^2C pins support Standard mode (Sm), Fast mode (Fm), Fast plus mode (Fm+), and High speed mode (Hs). Only I^2C High Speed pins support the High speed mode (*Hs*, up to 3.4MHz). The available I^2C pins are listed in the I/O Multiplexing section.

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.
2. The pins PA12, PA13, PB12, PB13, PB16, PB17, PB30, PB31 are limited on output low-level current in I^2C Standard mode (*Sm*) and Fast mode (*Fm*). The limitation is 2.5mA instead of 3mA for V_{OL}=0.4V, and 3mA instead of 6mA for V_{OL}=0.6V.
3. The following pins are High Sink pins and have different properties than normal pins: PA08, PA09, PA16, PA17, PA22, PA23, PA27, PA31.
4. The following pins are Backup pins and have different properties than normal pins: PA00, PA01, PB00, PB01, PB02, PB03.

Related Links

SERCOM I2C Pins on page 32

46.9. Analog Characteristics

46.9.1. Voltage Regulator Characteristics

46.9.1.1. Buck Converter

Table 46-13. Buck Converter Electrical Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
P_{EFF}	Power Efficiency(2)	I_{OUT} = 5mA	-	86	-	%
		I_{OUT} = 50mA	-	85	-	%
VREGSCAL	Voltage scaling(1)	min step size for PLx to PLy transition	-	5	-	mV
		Voltage Scaling Period	-	1	-	µs

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.
2. These values are based on characterization.

Table 46-14. External Components Requirements in Switching Mode(1)

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
C_{IN}	Input regulator capacitor		-	4.7	-	µF
		Ceramic dielectric	-	100	-	nF
C_{OUT}	Output regulator capacitor		-	1	-	µF
		Ceramic dielectric	-	100	-	nF
L_{EXT}	External inductance	Murata LQH3NPN100MJ0	-	10	-	µH

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
$R_{SERIE\_L_{EXT}}$	Serial resistance of Lext	-	-	-	0.7	Ω
$I_{SAT\_L_{EXT}}$	Saturation current	-	275	-	-	mA

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.

46.9.1.2. LDO Regulator

Table 46-15. LDO Regulator Electrical Characteristics

Symbol	Parameter	Conditions	Typ.	Units
VREGSCAL	Voltage scaling	min step size for PLx to Ply transistion	5	mV
		Voltage Scaling Period(1)	1	µs

Note: 1. These are based on simulation. These values are not covered by test or characterization

Table 46-16. External Components Requirements in Linear Mode

Symbol	Parameter	Conditions	Typ.	Units
C_{IN}	Input regulator capacitor		4.7	µF
		Ceramic dielectric X7R	100	nF
C_{OUT}	Output regulator capacitor		1	µF
		Ceramic dielectric X7R	100	nF

46.9.2. APWS

Table 46-17. Automatic Power Switch Characteristics

Symbol	Parameters	Min.	Typ.	Max.	Unit
CD	Decoupling capacitor	-	4.7	-	µF
T_{HUP}	V_{DDIN} threshold	-	1.84	-	V
T_{HDWN}		-	1.75	-	V
T_{HHYS}	V_{DDIN} hysteresis	-	90	-	mV

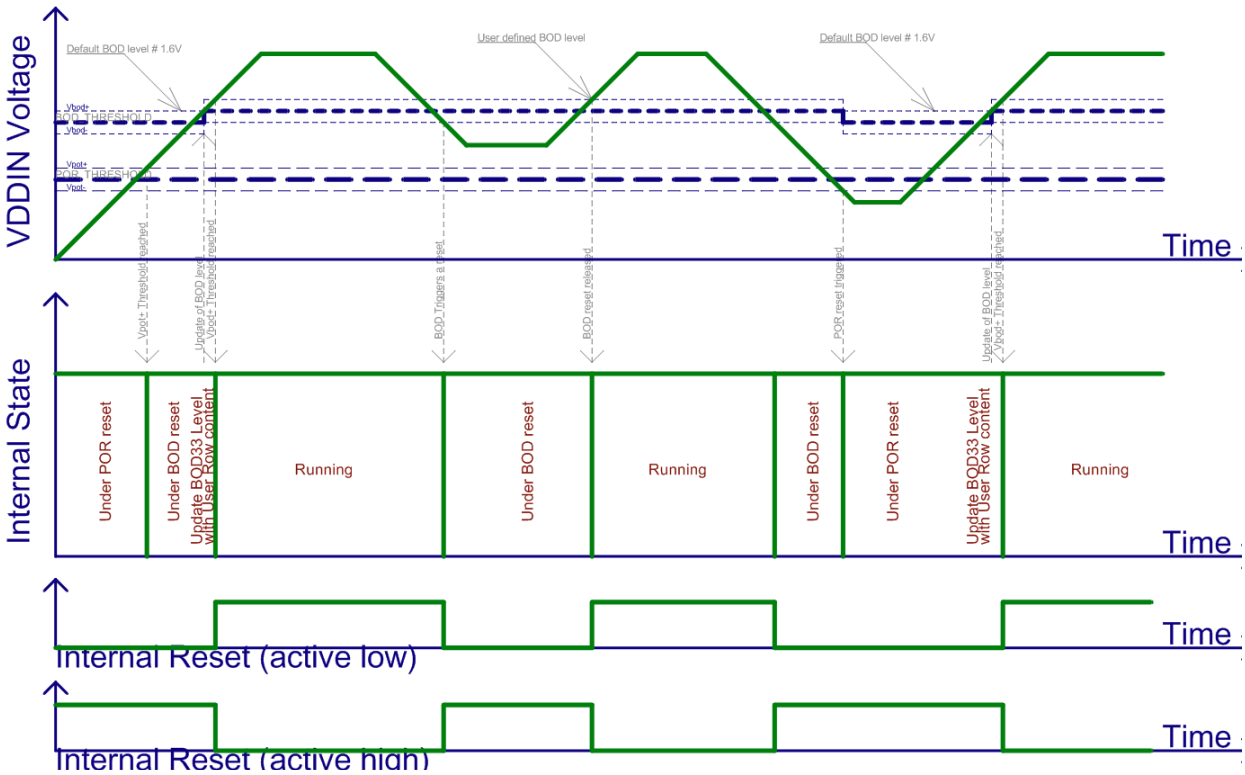
Note: With CDmin = Imax /(dV/dt)

46.9.3. Power-On Reset (POR) Characteristics

Table 46-18. POR33 Characteristics

Symbol	Parameters	Conditions	Min.	Typ.	Max.	Unit
V_{POT+}	Voltage threshold Level on V_{DDIN} rising		1.53	1.58	1.62	V
V_{POT-}	Voltage threshold Level on V_{DDIN} falling		0.6	1.04	1.39	V

Figure 46-1. BOD Reset Behavior at Startup and Default Levels



46.9.4. Brown-Out Detectors (BOD) Characteristics

Table 46-19. BOD33 Characteristics(1)

Symbol	Parameters	Conditions	Min.	Typ.	Max.	Unit
V_{BOD+}	BOD33 high threshold Level	V_{BAT}, 15	1.67	1.74	1.81	V
		V_{DDIN}, 7	1.75	1.75	1.80	
		V_{DDIN}, 6	1.66	1.72	1.75	
		V_{BAT}, 55	2.80	2.90	3.01	
		V_{DDIN}, 39	2.65	2.87	2.95	
		V_{BAT}, 63	3.02	3.14	3.26	
		V_{DDIN}, 48	03.12	3.20	3.29	
V_{BOD-} / V_{BOD}	BOD33 low threshold Level	V_{BAT}, 15	1.60	1.66	1.72	V
		V_{DDIN}, 7	1.63	1.673	1.71	
		V_{DDIN}, 6	1.60	1.65	1.68	
		V_{BAT}, 55	2.70	2.81	2.92	
		V_{DDIN}, 39	2.70	2.77	2.84	
		V_{BAT}, 63	2.92	3.04	3.16	
		V_{DDIN}, 48	3.00	3.08	3.16	
	Step size	-		34		mV

Symbol	Parameters	Conditions	Min.	Typ.	Max.	Unit
V_{HYS}	Hysteresis (V_{BOD+} - V_{BOD-}) BOD33.LEVEL= 0x0 to 0x3F	V_{BAT}	38	-	135	mV
		V_{DDIN}	47	-	155	
T_{START}	Startup time	time from enable to RDY	-	3.2	-	µs

Note:

1. These values are based on characterization.

Table 46-20. BOD33 Power Consumption(1)

Mode	Conditions	V_{CC}	Ta	Min.	Typ.	Max.	Units
BOD33	IDLE, mode CONT	1.8V	Max. at 85°C Typ. at 25°C	-	17.9	21.5	µA
		3.3V		-	28.8	33.1	
	IDLE, mode SAMPL	1.8V		-	0.020	0.306	
		3.3V		-	0.033	0.197	
	STDBY, mode SAMPL	1.8V		-	0.087	0.231	
		3.3V		-	0.114	0.347	

Note:

1. These values are based on characterization.

Related Links

Hysteresis on page 292

46.9.5. Analog-to-Digital Converter (ADC) Characteristics

Table 46-21. Operating Conditions

Symbol	Parameters	Conditions	Min.	Typ.	Max.	Unit
RES	Resolution	-	-	-	12	bits
	Conversion speed	-	10	-	1000	ksps
fs	Sampling clock	-	10	-	1000	kHz
clk	ADC Clock frequency	-	-	fs.16	-	Hz
T_S	Sampling time	OFFCOMP=1	250	-	25000	ns
	Conversion range	Diff mode	$-V_{REF}$	-	V_{REF}	V
		Single-Ended mode	0	-	V_{REF}	

Symbol	Parameters	Conditions	Min.	Typ.	Max.	Unit
V_{REF}	Reference input	REFCOMP=1	1	-	V_{DDANA}-0.6	V
		REFCOMP=0	V_{DDANA}	-	V_{DDANA}	
V_{IN}	Input channel range	-	0	-	V_{DDANA}	V
V_{CMIN}	Input common mode voltage	For V_{REF} > 1.0V	0.7	-	V_{REF}-0.7	V
		For V_{REF}=1.0V	0.3	-	V_{REF}-0.3	
CSAMPLE(1)	Input sampling capacitance	-	-	2.8	3.2	pF
RSAMPLE(1)	Input sampling on-resistance	-	-	-	1715	Ω
Rref(1)	Reference input source resistance	REFCOMP=1	-	-	5	kΩ

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.

Table 46-22. Power Consumption(1)

Symbol	Parameters	Conditions	Ta	Min.	Typ.	Max.	Unit
I_{DDANA}	Differential Mode	fs=1Msps / Reference buffer disabled BIASREFBUF='111', BIASCOMP='111' V_{DDANA}=V_{REF}=1.6V	Max.85°C Typ.25°C	-	105	128	µA
		V_{DDANA}=V_{REF}=3.6V		-	279	307	
	Differential Mode	fs=1Msps / Reference buffer enabled BIASREFBUF='111', BIASCOMP='111' V_{DDANA}=1.6V, V_{REF}=1.0V		-	175	231	µA
		V_{DDANA}=3.0V, V_{REF}=2.0V		-	300	374	
		V_{DDANA}=3.6V, V_{REF}=3.0V		-	356	438	
	Differential Mode	fs=10ksps / Reference buffer disabled BIASREFBUF='111', BIASCOMP='111' V_{DDANA}=V_{REF}=1.6V		-	30	41	µA
		V_{CC}=V_{REF}=3.6V		-	53	71	
	Differential Mode	fs=10ksps / Reference buffer enabled BIASREFBUF='111', BIASCOMP='111' V_{DDANA}=1.6V, V_{REF}=1.0V		-	95	139	µA
		V_{DDANA}=3.0V, V_{REF}=2.0V		-	115	178	
		V_{DDANA}=3.6V, V_{REF}=3.0V		-	122	187	

Symbol	Parameters	Conditions	Ta	Min.	Typ.	Max.	Unit
I_{DDANA}	Single-Ended Mode	fs=1Msps / Reference buffer disabled BIASREFBUF='111', BIASCOMP='111' $V_{DDANA}=V_{REF}=1.6V$	Max.85°C Typ.25°C	-	138	158	µA
		$V_{DDANA}=V_{REF}=3.6V$		-	321	359	
	Single-Ended Mode	fs=1Msps / Reference buffer enabled BIASREFBUF='111', BIASCOMP='111' $V_{DDANA}=1.6V$, $V_{REF}=1.0V$		-	203	257	µA
		$V_{DDANA}=3.0V$, $V_{REF}=2.0V$		-	331	413	
		$V_{DDANA}=3.6V$, $V_{REF}=3.0V$		-	388	482	
	Single-Ended Mode	fs=10ksps / Reference buffer disabled BIASREFBUF='111', BIASCOMP='111' $V_{CC}=V_{REF}=1.6V$		-	46	62	µA
		$V_{DDANA}=V_{REF}=3.6V$		-	89	120	
	Single-Ended Mode	fs=10ksps / Reference buffer enabled BIASREFBUF='111', BIASCOMP='111' $V_{CC}=1.6V$, $V_{REF}=1.0V$		-	109	157	µA
		$V_{DDANA}=3.0V$, $V_{REF}=2.0V$		-	138	211	
		$V_{DDANA}=3.6V$, $V_{REF}=3.0V$		-	148	228	

Note: 1. These values are based on characterization.

Table 46-23. Differential Mode(1)

Symbol	Parameters	Conditions	Min.	Typ.	Max.	Unit
ENOB	Effective Number of bits	$V_{DDANA}=3.0V$ / $V_{ref}=2.0V$	9.6	10.5	10.6	bits
		$V_{DDANA}=1.6V/3.6V$ $V_{ref}=1.0V$	8.9	9.7	9.9	
		$V_{DDANA}=V_{ref}=1.6V$	10	10.5	11.1	
		$V_{DDANA}=V_{ref}=3.6V$	10.5	10.9	11.0	
TUE	Total Unadjusted Error	$V_{DDANA}=3.0V$, $V_{ref}=2.0V$	-	7.5	11	LSB
INL	Integral Non Linearity	$V_{DDANA}=3.0V$, $V_{ref}=2.0V$	-	+/-1.5	+/-2.3	LSB

Symbol	Parameters	Conditions	Min.	Typ.	Max.	Unit
DNL	Differential Non Linearity	V_{DDANA}=3.0V, V_{ref}=2.0V	-	+/-0.8	+/-1.5	LSB
	Gain Error	External Reference voltage 1.0V	-	+/-0.7	+/-1.5	%
		External Reference voltage 3.0V	-	+/-0.2	+/-0.5	
		Reference bandgap voltage	-	+/-0.4	+/-4.4	
		VDDANA	-	+/-0.1	+/-0.4	
		VDDANA/2	-	+/-0.4	+/-1.3	
		VDDANA/1.6	-	+/-0.3	+/-0.9	
	Offset Error	External Reference voltage 1.0V	-	+/-1.1	+/-2.4	mV
		External Reference voltage 3.0V	-	+/-1.1	+/-3	
		Reference bandgap voltage	-	+/-2.3	+/-7.5	
		VDDANA	-	+/-0.9	+/-2.9	
		VDDANA/2	-	+/-1	+/-2.6	
		VDDANA/1.6	-	+/-1	+/-2.9	
SFDR	Spurious Free Dynamic Range	Fs=1MHz / Fin=13 kHz / Full range Input signal V_{DDANA}=3.0V, V_{ref}=2.0V	68	75	77	dB
SINAD	Signal to Noise and Distortion ratio		60	65	66	
SNR	Signal to Noise ratio		61	66	67	
THD	Total Harmonic Distortion		-74	-73	-67	
	Noise RMS	External Reference voltage	-	1.0	2.5	mV

Note: 1. These values are based on characterization.

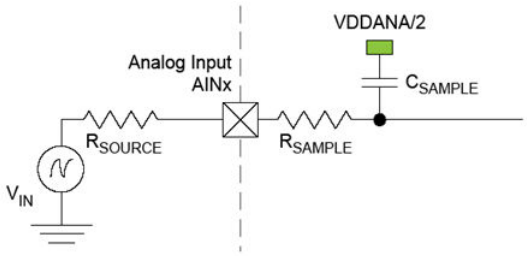
Table 46-24. Single-Ended Mode(1)

Symbol	Parameters	Conditions	Min.	Typ.	Max.	Unit
ENOB	Effective Number of bits	V_{DDANA}=3.0V / V_{ref} =2.0V	8.5	9.5	9.8	bits
		V_{DDANA}=1.6V/3.6V V_{ref}=1.0V	7.5	8.7	8.9	
		V_{DDANA}=V_{ref}=1.6V (2)	9.0	9.5	9.8	
		V_{DDANA}=V_{ref}=3.6V	9.2	9.8	9.9	
TUE	Total Unadjusted Error	V_{DDANA}=3.0V, V_{ref}=2.0V	-	17.4	31	LSB
INL	Integral Non Linearity	V_{DDANA}=3.0V, V_{ref}=2.0V	-	+/-2.2	+/-10.1	LSB
DNL	Differential Non Linearity	V_{DDANA}=3.0V, V_{ref}=2.0V	-	+/-0.8	+/-9.5	LSB
	Gain Error	External Reference voltage 1.0V	-	+/-1	+/-1.3	%
		External Reference voltage 3.0V	-	+/-0.3	+/-0.6	
		Reference bandgap voltage	-	+/-0.4	+/-3.2	
		V_{ref}=V_{DDANA} (2)	-	+/-0.1	+/-0.3	
		V_{ref}=V_{DDANA}/2	-	+/-0.6	+/-1.4	
		V_{ref}=V_{DDANA}/1.6	-	+/-0.4	+/-1	
	Offset Error	External Reference voltage 1.0V	-	+/-3.4	+/-13	mV
		External Reference voltage 3.0V	-	+/-3.6	+/-24	
		Reference bandgap voltage	-	+/-1	+/-14	
		V_{ref}=V_{DDANA} (2)	-	+/-4.2	+/-25	
		V_{ref}=V_{DDANA}/2	-	+/-5.7	+/-10	
		V_{ref}=V_{DDANA}/1.6	-	+/-6.3	+/-13	
SFDR	Spurious Free Dynamic Range	Fs=1MHz / Fin=13kHz / Full range Input signal V_{DDANA}=3.0V, V_{ref}=2.0V	65	71	78	dB
SINAD	Signal to Noise and Distortion ratio		53	59	61	
SNR	Signal to Noise ratio		53	59	61	
THD	Total Harmonic Distortion		-76	-70	-64	
	Noise RMS	External Reference voltage	-	2.0	7.0	mV

Note:

1. These values are based on characterization.
2. All parameters given above exclude the corner V_{DDANA} = V_{ref} = 1.6V, Ta = -40°C.

Figure 46-2. ADC Analog Input AINx



The minimum sampling time $t_{samplehold}$ for a given R_{source} can be found using this formula:

$$t_{samplehold} \geq \left(R_{sample} + R_{source}\right) \times C_{sample} \times \left(n + 2\right) \times \ln\left(2\right)$$

For 12-bit accuracy:

$$t_{samplehold} \geq \left(R_{sample} + R_{source}\right) \times C_{sample} \times 9.7$$

where $t_{samplehold} \geq \frac{1}{2 \times f_{ADC}}$.

46.9.6. Digital to Analog Converter (DAC) Characteristics

Table 46-25. Operating Conditions(1)

Symbol	Parameters	Conditions	Min.	Typ.	Max.	Unit
Res	Resolution	-	-	-	12	bits
clk	Internal DAC Clock frequency	-	-	-	12	MHz
fs_dac	Sampling frequency	clk/12, CCTRL=0x0 (Low Power)	-	-	10	ksps
		clk/12, CCTRL=0x2 (High Power)	-	-	1	Msps
V_{OUTmin}	Min Output Voltage	-	-	-	0.15	V
V_{OUTmax}	Max Output Voltage	-	V_{DDANA}-0.15	-	-	
V_{REF}	Reference input	CTRLB.REFSEL[1:0]=01	1 (2)	-	V_{DDANA}-0.15	V
		CTRLB.REFSEL[1:0]=10	1 (2)	-	V_{DDANA}	
C_{VREF}	External decoupling capacitor	-	-	220	-	nF
C_{LOAD}	Output capacitor load	-	-	-	50	pF
R_{LOAD}	Output resistance load	-	5	-	-	kΩ
t_s	Settling time	For reaching +/-1LSB of the final value. Step size < 500 LSB - C_{load} = 50pF	-	-	1	µs
t_{s_FS}	Settling time 0x080 to 0xF7F	For reaching +/-1LSB of the final value. Step size from 0% to 100% - C_{load} = 50pF	-	5	7	µs

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.
2. Internal Vref min value if used.

Table 46-26. Differential Mode(1)

Symbol	Parameters	Conditions	Min.	Typ.	Max.	Unit
INL	Integral Non Linearity, Best-fit curve from 0x080 to 0xF7F	clk=12MHz, V_{DDANA}=3.0V, External Ref.=2.0V, C_{Load}=50pF	-	+/-2.5	+/-3.5	LSB
		clk=12MHz, V_{DDANA}=2.0V, internal V_{DDANA}=2.0V, C_{Load}=50pF	-	+/-2.2	+/-3.0	
DNL	Differential Non Linearity, Best-fit curve from 0x080 to 0xF7F	clk=12MHz, V_{DDANA}=3.0V, External Ref.=2.0V, C_{Load}=50pF	-	+/-2.0	+/-2.5	LSB
		clk=12MHz, V_{DDANA}=2.0V, internal V_{DDANA}=2.0V, C_{Load}=50pF	-	+/-1.5	+/-2.5	
Gerr	Gain Error	External Reference voltage	-	+/-0.3	+/-0.8	% FSR
		Internal V_{DDANA} Reference	-	+/-0.2	+/-0.5	
		1.0V Internal Reference voltage	-	+/-1	+/-3.0	
Offerr	Offset Error	External Reference voltage	-	+/-5.0	+/-10.0	mV
		Internal V_{DDANA} Reference	-	+/-4.0	+/-10.0	
		1.0V Internal Reference voltage	-	+/-10.0	+/-30.0	
TC_g	Gain Drift		-20	-	20	ppm/°C
TC_o	Offset Drift		-0.05	-0.01	0.05	mV/°C
ENOB	Effective Number Of Bits	Fs=1Ms/s - External Ref - High Power	9.5	10.1	10.7	Bits
SNR	Signal to Noise ratio		58.0	67.0	71.0	dB
THD	Total Harmonic Distortion		-71.0	-64.0	-59.0	dB

Note:

1. These values are based on characterization.

Table 46-27. Single-Ended Mode(1)

Symbol	Parameters	Conditions	Min.	Typ.	Max.	Unit
INL	Integral Non Linearity, Best-fit curve from 0x080 to 0xF7F	clk=12MHz, V_{DDANA}=3.0V, External Ref.=2.0V, C_{Load}=50pF	-	+/-3	+/-4.5	LSB
		clk=12MHz, V_{DDANA}=2.0V, internal V_{DDANA}=2.0V, C_{Load}=50pF	-	+/-2.5	+/-3.5	
DNL(1)	Differential Non Linearity, Best-fit curve from 0x080 to 0xF7F	clk=12MHz, V_{DDANA}=3.0V, External Ref.=2.0V, C_{Load}=50pF	-	+/-3.0	+/-4.0	LSB
		clk=12MHz, V_{DDANA}=2.0V, internal V_{DDANA}=2.0V, C_{Load}=50pF	-	+/-2.5	+/-3.5	
Gerr	Gain Error	External Reference voltage	-	+/-0.3	+/-0.8	% FSR
		Internal V_{DDANA} Reference	-	+/-0.2	+/-0.5	
		1.0V Internal Reference voltage	-	+/-0.1	+/-3.0	

Symbol	Parameters	Conditions	Min.	Typ.	Max.	Unit
Offerr	Offset Error	External Reference voltage	-	+/-5.0	+/-10.0	mV
		Internal V_{DDANA} Reference	-	+/-7.0	+/-10.0	
		1.0V Internal Reference voltage	-	+/-10.0	+/-12.0	
TC_g	Gain Drift		-20	-	20	ppm/°C
TC_o	Offset Drift		-0.03	-0.05	0.02	mV/°C
ENOB	Effective Number Of Bits	Fs=1Ms/s - External Ref - High Power	9.0	10.0	10.3	Bits
SNR	Signal to Noise ratio		56.0	67.0	68.5	dB
THD	Total Harmonic Distortion		-69.0	-65.0	-55.0	dB

Note:

1. These values are based on characterization.

Table 46-28. Power Consumption(1)

Symbol	Parameters	Conditions	Ta	Min.	Typ.	Max.	Unit
I_{DDANA}	Differential Mode, DC supply current, 2 output channels - without load	fs=1Msps, CCTRL=0x2, V_{REF}>2.4V, V_{CC}=3.3V	Max.85°C Typ.25°C	-	401	587	µA
		fs=10ksps, CCTRL=0x0, V_{REF}<2.4V, V_{CC}=3.3V		-	84	174	
I_{DDANA}	Single-Ended Mode, DC supply current, 2 output channels - without load	fs=1Msps, CCTRL=0x2, V_{REF}>2.4V, V_{CC}=3.3V		-	297	395	µA
		fs=10ksps, CCTRL=0x0, V_{REF}<2.4V, V_{CC}=3.3V		-	53	112	

Note: 1.These values are based on characterization.

46.9.7. Analog Comparator (AC) Characteristics

Table 46-29. Electrical and Timing

Symbol	Parameters	Conditions	Min.	Typ	Max.	Unit
	Positive and Negative input range voltage		0	-	V_{DDANA}	V
ICMR	Input common mode range		0	-	V_{DDANA}-0.1	V
Off(2)	Offset	Xsel mode 0	-70	-4.5/+1.5	70	mV
		Xsel mode 1	-55	-4.5/+1.5	55	
		Xsel mode 2	-48	-4.5/+1.5	48	
		Xsel mode 3	-42	-4.5/+1.5	42	

Symbol	Parameters	Conditions	Min.	Typ	Max.	Unit
V_{Hys}(2)	Hysteresis	Xsel mode 0	10	45	74	mV
		Xsel mode 1	22	70	106	
		Xsel mode 2	37	90	116	
		Xsel mode 3	49	105	131	
T_{pd}(2)	Propagation Delay $V_{cm}=V_{ddana}/2$, V_{in}= +-100mV overdrive from V_{CM}	Xbias mode 0	-	4	12.3	µs
		Xbias mode 1	-	0.97	2.59	
		Xbias mode 2	-	0.56	1.41	
		Xbias mode 3	-	0.33	0.77	
T_{start}(1)	Start-up time	Xbias mode 0	-	17	56	µs
		Xbias mode 1	-	0.85	4.6	
		Xbias mode 2	-	0.55	3.2	
		Xbias mode 3	-	0.45	2.7	
V_{scale}(2)	INL			0.34		LSB
	DNL			0.06		
	Offset Error			0.1		
	Gain Error			1.22		

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.
2. These values are based on characterization.

Table 46-30. Power Consumption(1)

Symbol	Parameters	Conditions	Ta	Min.	Typ	Max.	Unit
I_{DDANA}	Current consumption $V_{CM}=V_{DDANA}/2$, +/-100mV overdrive from V_{CM} V_{CC}=3.3V Voltage scaler disabled	Xbias mode 0	Max.85°C Typ.25°C	-	50	1973	nA
		Xbias mode 1		-	156	2082	
		Xbias mode 2		-	289	2223	
		Xbias mode 3		-	549	2495	
	Current consumption Voltage Scaler only	V_{CC}=3.3V		-	13	17	µA

Note:

1. These values are based on characterization.

46.9.8. DETREF

Table 46-31. Reference Voltage Characteristics(1)

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
ADC/DAC Ref	ADC/DAC internal reference	nom. 1.0V, V_{CC}=3.0V, T= 25°C	0.967	1.0	1.017	V
		nom. 1.1V, V_{CC}=3.0V, T= 25°C	1.069	1.1	1.120	
		nom. 1.2V, V_{CC}=3.0V, T= 25°C	1.167	1.2	1.227	
		nom. 1.25V, V_{CC}=3.0V, T= 25°C	1.214	1.25	1.280	
		nom. 2.0V, V_{CC}=3.0V, T= 25°C	1.935	2.0	2.032	
		nom. 2.2V, V_{CC}=3.0V, T= 25°C	2.134	2.2	2.242	
		nom. 2.4V, V_{CC}=3.0V, T= 25°C	2.328	2.4	2.458	
		nom. 2.5V, V_{CC}=3.0V, T= 25°C	2.420	2.5	2.565	
	Ref Temperature coefficient	drift over [-40, +25]°C	-	-0.01/+0.015	-	%/°C
		drift over [+25, +85]°C	-	-0.01/0.005	-	
	Ref Supply coefficient	drift over [1.6, 3.6]V	-	+/-0.35	-	%/V
AC Ref	AC Ref Accuracy	V_{CC}=3.0V, T=25°C	1.073	1.1	1.123	V
	Ref Temperature coefficient	drift over [-40, +25]°C	-	+/-0.01		%/°C
		drift over [+25, +85]°C	-	-0.005/+0.001	-	%/°C
	Ref Supply coefficient	drift over [1.6, 3.6]V	-	-0.35/+0.35	-	%/V

Note: These values are based on characterization.

46.9.9. Temperature Sensor Characteristics

Table 46-32. Temperature Sensor Characteristics(1)

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
Ts_{Vout}	Temperature sensor output voltage	T= 25°C, V_{CC}=3.0V	-	656	-	mV
Ts_{Slope}	Temperature sensor slope	V_{CC} = 3.0V	-	2.24	-	mV/°C
$Ts_{V_{OUT}OV_{DDANA}}$	Variation over V_{DDANA} voltage	T= 25°C	-	1.1	-	mV/V

Note: 1.These values are based on characterization.

46.9.10. OPAMP

Table 46-33. Operating Conditions

Symbol	Parameters	Conditions	Min.	Typ.	Max.	Unit
V_{CC}	Power Supply	All power modes	1.6	3	3.63	V
V_{in}	Input voltage range		0	-	Vcc	V
V_{out}	Output voltage range		0.15	-	Vcc-0.15	V

Symbol	Parameters	Conditions	Min.	Typ.	Max.	Unit
C_{load}	Maximum capacitance load		-	-	50	pF
R_{load} (1)	Minimum resistive load	Output Range[0.15V;Vcc-0.15V]	3.5	-	-	kΩ
		Output Range[0.3V;Vcc-0.3V]	0.5	-	-	
I_{load}(1)	DC output current load	Output Range[0.15V;Vcc-0.15V]	-	-	1	mA
		Output Range[0.3V;Vcc-0.3V]	-	-	6.9	

Note: 1. These values are based on simulation. They are not covered by production test limits or characterization.

Table 46-34. Power Consumption(1)

Symbol	Parameters	Conditions	Ta	Min.	Typ.	Max.	Unit
IDD	DC supply current (Voltage Doubler OFF)	Mode 3,VCC =3.3V	Max 85°C Typ 25°C	-	184	312	uA
		Mode 2,VCC =3.3V		-	72	127	
		Mode 1,VCC =3.3V		-	21	33	
		Mode 0 ,VCC =3.3V		-	6	9	
	Voltage Doubler consumption	VCC =3.3V		-	0,69	1,5	

Note: 1. These values are based on characterization.

Table 46-35. Static Characteristics in 1X Gain(1)

Symbol	Parameters	Conditions	Min.	Typ.	Max.	Unit
G0	Open loop gain	Mode 3	-	114.5	-	dB
		Mode 2	-	117.6	-	
		Mode 1	-	116.8	-	
		Mode 0	-	108.5	-	
GBW	Gain Bandwidth	Mode 3	-	7.1	-	MHz
		Mode 2	-	2.8	-	
		Mode 1	-	0.85	-	
		Mode 0	-	0.2	-	
Φ_m	Phase margin	Mode 3	-	71.5	-	deg
		Mode 2	-	64	-	
		Mode 1	-	56	-	
		Mode 0	-	52	-	
T_{r1}	Response Time at 240µV (X1 gain)	Mode 3	-	1.3	-	µs
		Mode 2	-	3.3	-	
		Mode 1	-	13	-	
		Mode 0	-	52	-	

Symbol	Parameters	Conditions	Min.	Typ.	Max.	Unit
ΔT_{r1}	Response Time Variation for 10mV	Mode 3	-	100	-	ns
		Mode 2	-	-	-	
		Mode 1	-	-	-	
		Mode 0	-	-	-	
T_{start}	Start-up time (Enable to Ready), (Voltage Doubler OFF)	Mode 3	-	2.7	-	µs
		Mode 2	-	6.35	-	
		Mode 1	-	21.5	-	
		Mode 0	-	88.5	-	
Oe	Input Offset Voltage		-	-	+-3.5	mV
SR	Slew rate	Mode 3	-2.8	-	2.6	V/µs
		Mode 2	-1.2	-	1.1	
		Mode 1	-0.3	-	0.3	
		Mode 0	-0.09	-	0.07	
CMRR	1X gain	Mode 3	-	83	-	dB
		Mode 2	-	84	-	
		Mode 1	-	84	-	
		Mode 0	-	83	-	
PSRR	1X gain	Mode 3	-	76	-	dB
		Mode 2	-	76	-	
		Mode 1	-	76	-	
		Mode 0	-	75	-	
	Integrated Noise, BW=[0.1Hz-10kHz], x1 gain - V_{OUT}=1V	Mode 3	-	7.9	-	μV_{RMS}
		Mode 2	-	8.3	-	
		Mode 1	-	9.9	-	
		Mode 0	-	12.7	-	
	Integrated Noise, BW=[0.1Hz-1MHz], x1 gain - V_{OUT}=1V	Mode 3	-	18.2	-	μV_{RMS}
		Mode 2	-	22.8	-	
		Mode 1	-	36.7	-	
		Mode 0	-	44.4	-	

Note: 1.These values are based on simulation. They are not covered by production test limits or characterization.

Table 46-36. PGA Electrical Characteristics(1)

Symbol	Parameters	Conditions	Min.	Typ.	Max.	Unit
	Gain accuracy	16X Gain	-	-	-1.43	%
		4X Gain	-	-	+-0.4	
		1X Gain	-	-	+/-0.25	

Symbol	Parameters	Conditions	Min.	Typ.	Max.	Unit
THD	Total Harmonic Distortion @ 10kHz - mode 3	16X Gain	-	-77	-	dB
		4X Gain	-	-72.8	-	
		1X Gain	-	-82.6	-	
	Integrated Noise, BW=[0.1Hz-10 kHz], 16X gain - V_{OUT}=1V	Mode 3	-	147	-	µVrms
		Mode 2	-	147	-	
		Mode 1	-	162	-	
		Mode 0	-	191	-	
	Integrated Noise, BW=[0.1Hz-1MHz], 16X gain - V_{OUT}=1V	Mode 3	-	262	-	µVrms
		Mode 2	-	247	-	
		Mode 1	-	235	-	
		Mode 0	-	235	-	

Note: 1.These values are based on simulation. They are not covered by production test limits or characterization.

46.10. NVM Characteristics

Table 46-37. NVM Max Speed Characteristics

	Conditions	CPU Fmax (MHz)			
		0WS	1WS	2WS	3WS
PL0 (-40/85°C)	V_{DDIN}>1.6 V	6	12	12	12
	V_{DDIN}>2.7 V	7.5	12	12	12
PL2 (-40/85°C)	V_{DDIN}>1.6 V	14	28	42	48
	V_{DDIN}>2.7 V	24	45	48	48

Table 46-38. NVM Timings Characteristics

Symbol	Timings	Max	Units
t_{FPP}	Page Write(1)	2.5	ms
t_{FRE}	Row erase(1)	6	
t_{FSE}	Sector erase(1)	8	

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.
2. For this Flash technology, a maximum number of 8 consecutive writes is allowed per row. Once this number is reached, a row erase is mandatory.

Table 46-39. NVM Reliability Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
Ret_{NVM25k}	Retention after up to 25k	Average ambient 55°C	10	50	-	Years
$Ret_{NVM2.5k}$	Retention after up to 2.5k	Average ambient 55°C	20	100	-	Years
Ret_{NVM100}	Retention after up to 100	Average ambient 55°C	25	>100	-	Years
Cyc_{NVM}	Cycling Endurance(1)	-40°C < Ta < 85°C	25K	100K	-	Cycles

Note: 1. An endurance cycle is a write and an erase operation.

Table 46-40. EEPROM Emulation(1) Reliability characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
Ret_{EE100k}	Retention after up to 100k	Average ambient 55°C	10	50	-	Years
Ret_{EE10k}	Retention after up to 10k	Average ambient 55°C	20	100	-	Years
Cyc_{EE}	Cycling Endurance(2)	-40°C < Ta < 85°C	100K	400K	-	Cycles

Note:

(1) The EEPROM emulation is a software emulation described in the App note AT03265.

(2) An endurance cycle is a write and an erase operation.

46.11. Oscillators Characteristics

46.11.1. Crystal Oscillator (XOSC) Characteristics

Digital Clock Characteristics

The following table describes the characteristics for the oscillator when a digital clock is applied on XIN.

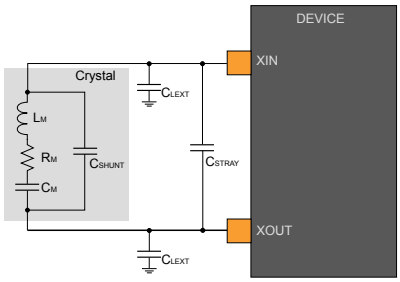
Table 46-41. Digital Clock Characteristics

Symbol	Parameter	Min.	Typ.	Max.	Units
F_{XIN}	XIN clock frequency	-	-	24	MHz
DC_{XIN}	XIN clock duty cycle	40	50	60	%

Chrystal Oscillator Characteristics

The following Table describes the characteristics for the oscillator when a crystal is connected between XIN and XOUT.

Figure 46-3. Oscillator Connection



The user must choose a crystal oscillator where the crystal load capacitance C_L is within the range given in the Table. The exact value of C_L can be found in the crystal datasheet. The capacitance of the external capacitors (C_{LEXT}) can then be computed as follows:

$C_{LEXT} = 2(C_L - C_{STRAY} - SHUNT)$,

where C_{STRAY} is the capacitance of the pins and the PCB, and C_{SHUNT} is the shunt capacity of the crystal.

Table 46-42. Multi Crystal Oscillator Electrical Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
F_{OUT}	Crystal oscillator frequency		0.4	-	32	MHz
ESR	Crystal Equivalent Series Resistance - SF=3	F=0.4MHz, C_L=100pF XOSC, GAIN=0	-	-	5.6K	Ω
		F=2MHz, C_L=20pF XOSC,GAIN=0	-	-	416	
		F=4MHz, C_L=20pF XOSC, GAIN=1	-	-	243	
		F=8MHz, C_L=20pF XOSC, GAIN=2	-	-	138	
		F=16MHz, C_L=20pF XOSC, GAIN=3	-	-	66	
		F=32MHz, C_L=20pF XOSC, GAIN=4	-	-	56	
C_{XIN}	Parasitic load capacitor		-	5.8	-	pF
C_{XOUT}			-	3.2	-	
T_{START}	Startup time	F=2MHz, C_L=20pF XOSC, GAIN=0	-	14k	48k	Cycles
		F=4MHz, C_L=20pF XOSC, GAIN=1	-	6.8k	19.5k	
		F=8MHz, C_L=20pF XOSC, GAIN=2	-	5.6k	13k	
		F=16MHz, C_L=20pF XOSC, GAIN=3	-	6.8k	14.5k	
		F=32MHz, C_L=20pF XOSC, GAIN=4	-	5.3K	9.6k	

Note: (1) These values are based on characterization.

Table 46-43. Power Consumption(1)

Symbol	Parameter	Conditions		Ta	Min.	Typ.	Max.	Units
I_{DD}	Current consumption	F=2MHz - CL=20pF XOSC,GAIN=0, VCC=3.3V	AGC=OFF	Max 85°C Typ 25°C	-	89	133	µA
			AGC=ON		-	82	130	
		F=4MHz - CL=20pF XOSC,GAIN=1, VCC=3.3V	AGC=OFF		-	140	194	
			AGC=ON		-	102	156	
		F=8MHz - CL=20pF XOSC,GAIN=2, VCC=3.3V	AGC=OFF		-	243	313	
			AGC=ON		-	166	232	
		F=16MHz - CL=20pF XOSC,GAIN=3, VCC=3.3V	AGC=OFF		-	493	605	
			AGC=ON		-	293	393	
		F=32MHz - CL=20pF XOSC,GAIN=4, VCC=3.3V	AGC=OFF		-	1467	1777	
			AGC=ON		-	651	1045	

Note: (1) These values are based on characterization.

46.11.2. External 32KHz Crystal Oscillator (XOSC32K) Characteristics

Digital Clock Characteristics

The following table describes the characteristics for the oscillator when a digital clock is applied on XIN32 pin.

Table 46-44. Digital Clock Characteristics(1)

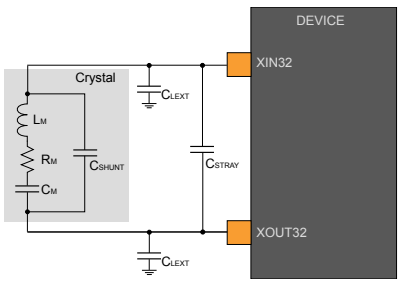
Symbol	Parameter	Min.	Typ.	Max.	Units
$f_{CPXIN32}$	XIN32 clock frequency		32.768	1000	kHz
DC_{XIN}	XIN32 clock duty cycle	40	50	60	%

Note: 1. These values are based on characterization.

Crystal Oscillator Characteristics

The following section describes the characteristics for the oscillator when a crystal is connected between XIN32 and XOUT32 pins.

Figure 46-4. Oscillator Crystal Connection



The user must choose a crystal oscillator where the crystal load capacitance C_L is within the range given in the table. The exact value of C_L can be found in the crystal datasheet. The capacitance of the external capacitors (C_{LEXT}) can then be computed as follows:

$C_{LEXT} = 2(C_L - C_{STRAY} - _{SHUNT})$,

where C_{STRAY} is the capacitance of the pins and PCB, C_{SHUNT} is the shunt capacitance of the crystal.

Table 46-45. 32KHz Crystal Oscillator Electrical Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
F_{OUT}	Crystal oscillator frequency	-	-	32.768	-	kHz
CL	Crystal load capacitance	-	7	9	12.5	pF
C_{SHUNT}	Crystal shunt capacitance	-	0.6	-	2	pF
C_M	Motional capacitance	-	0.6	-	3	fF
ESR	Crystal Equivalent Series Resistance - SF=3	f=32.768kHz, C_L=12.5pF	-	50	70	kΩ
C_{XIN32k}	Parasitic load capacitor(2)	-	-	2.31	-	pF
$C_{XOUT32k}$		-	-	2.53	-	
$t_{STARTUP}$	Startup time	f=32.768kHz, C_L=12.5pF	-	25k	82k	Cycles
P_{on}	Drive Level (1)	-	-	-	0.1	µW

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.
2. For device revision A, see Erratum 13425.

Table 46-46. Power Consumption(1)

Symbol	Parameter	Conditions	Ta	Min.	Typ.	Max.	Units
I_{DD}	Current consumption	VCC=3.3V	Max 85°C Typ 25°C	-	311	723	nA

Note: (1) These values are based on characterization.

46.11.3. 32.768kHz Internal Oscillator (OSC32K) Characteristics

Table 46-47. 32KHz RC Oscillator Electrical Characteristics(1)

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
F_{OUT}	Output frequency	at 25°C, at V_{DDIO}=3.3V	32.572	32.768	33.044	kHz
		at 25°C, over [1.62, 3.63]V	31.425	32.768	33.603	kHz
		over[-40,+85]°C, over [1.62, 3.63]V	28.581	32.768	34.716	kHz
I_{OSC32k}	Current consumption		-	0.54		µA

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
$T_{STARTUP}$	Startup time		-	1	2	cycles
Duty	Duty Cycle		-	50	-	%

Note: 1. These values are based on characterization.

Table 46-48. Power Consumption(1)

Symbol	Parameter	Conditions	Ta	Min.	Typ.	Max.	Units
I_{DD}	Current consumption	VCC=3.3V	Max 85°C Typ 25°C	-	0.54	1.10	µA

Note: (1) These values are based on characterization.

46.11.4. Internal Ultra Low Power 32KHz RC Oscillator (OSCULP32K) Characteristics

Table 46-49. Ultra Low Power Internal 32KHz RC Oscillator Electrical Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
F_{OUT}	Output frequency	at 25°C, at V_{DDIO}=3.3V	31.775	32.768	34.033	kHz
		at 25°C, over [1.62, 3.63]V	31.848	32.768	34.202	kHz
		over[-40,+85]°C, over [1.62, 3.63]V	26.296	32.768	38.384	kHz
Duty	Duty Cycle		-	50	-	%

46.11.5. Digital Frequency Locked Loop (DFLL48M) Characteristics

Table 46-50. DFLL48M Characteristics - Open Loop Mode(1,2)

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
$F_{OpenOUT}$	Output frequency	DFLLVAL.COARSE=DFLL48M_COARSE_CAL DFLLVAL.FINE=512	46.6	47.8	49	MHz
$T_{OpenSTARTUP}$	Startup time	DFLLVAL.COARSE=DFLL48M_COARSE_CAL DFLLVAL.FINE=512 F_{OUT} within 90% of final value	-	8.3	9.1	µs

Note:

1. DFLL48 in open loop can be used only with LDO regulator.
2. These values are based on characterization.

Table 46-51. DFLL48M Characteristics - Closed Loop Mode

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
$F_{CloseOUT}$	Average Output frequency	F_{REF} = 32.768kHz	47.2	48	48.8	MHz
F_{REF}(2,3)	Input reference frequency		732	32768	33000	Hz

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
$F_{CloseJitter}$(1)	Period Jitter	F_{REF} = 32.768kHz	-	-	0.51	ns
T_{Lock}(1)	Lock time	F_{REF} = 32.768kHz DFLLVAL.COARSE=DFLL48M_COARSE_CAL DFLLVAL.FINE = 512 DFLLCTRL.BPLCKC = 1 DFLLCTRL.QLDIS = 0 DFLLCTRL.CCDIS = 1 DFLLMUL.FSTEP = 10		200	700	µs

Note:

1. These values are based on characterization.
2. To insure that the device stays within the maximum allowed clock frequency, any reference clock for the DFLL in close loop must be within a 2% error accuracy.
3. These values are based on simulation. They are not covered by production test limits or characterization.

Table 46-52. DFLL48M Power Consumption(1)

Symbol	Parameter	Conditions	Ta	Min.	Typ.	Max.	Units
I_{DD}	Power consumption Open Loop	DFLLVAL.COARSE=DFLL48M_COARSE_CAL	Max.85°C Typ.25°C	-	286	-	µA
I_{DD}	Power consumption Closed Loop	F_{REF} = 32.768kHz, V_{CC}=3.3V		-	362	-	µA

Note:

1. These values are based on characterization.

46.11.6. 16MHz RC Oscillator (OSC16M) Characteristics

Table 46-53. Multi RC Oscillator Electrical Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
F_{OUT}	Output frequency	V_{CC}=3.3V, T=25°C	3.953	4	4.062	MHz
			7.877	8	8.112	
			11.857	12	12.139	
			15.754	16	16.235	
TempCo	Freq vs. temperature drift		-4		4	%
SupplyCo	Freq vs. supply drift		-2		2	

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
T_{WUP}(2)	Wake up time - 1st clock edge after enable	F_{OUT} = 4MHz	-	0.12	0.27	µs
		F_{OUT} = 8MHz	-	0.12	0.25	
		F_{OUT} = 12MHz	-	0.12	0.27	
		F_{OUT} = 16MHz	-	0.12	0.25	
$T_{STARTUP}$(2)	Startup time	F_{OUT} = 4MHz	-	1.2	2.9	µs
		F_{OUT} = 8MHz	-	1.3	2.6	
		F_{OUT} = 12MHz	-	1.3	2.8	
		F_{OUT} = 16MHz	-	1.4	3.1	
Duty(1)	Duty Cycle	-	45	50	55	%

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.
2. These values are based on characterization.

Table 46-54. Power Consumption(1)

Symbol	Parameter	Conditions	Ta	Min.	Typ.	Max.	Units
F_{OUT}	Output frequency	F_{out}=4MHz, V_{CC}=3.3V	Max.85°C Typ.25°C	-	64	96	µA
		F_{out}=8MHz, V_{CC}=3.3V		-	91	122	
		F_{out}=12MHz, V_{CC}=3.3V		-	114	144	
		F_{out}=16MHz, V_{CC}=3.3V		-	141	169	

Note:

1. These values are based on characterization.

46.11.7. Digital Phase Lock Loop (DPLL) Characteristics

Table 46-55. Fractional Digital Phase Lock Loop Characteristics

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
F_{IN}	Input Clock Frequency		32	-	2000	kHz
F_{OUT}	Output frequency	PL0	48	-	48	MHz
		PL2	48	-	96	MHz

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
J_P(2)	Period Jitter	F_{IN}=32kHz @ F_{OUT}=48MHz	-	1.9	5.0	%
		F_{IN}=32kHz @ F_{OUT}=96MHz	-	3.3	7.0	
		F_{IN}=2MHz @ F_{OUT}=48MHz	-	2.0	8.0	
		F_{IN}=2MHz @ F_{OUT}=96MHz	-	4.2	7.0	
T_{LOCK}(2)	Lock Time	After startup, time to get lock signal, F_{IN}=32kHz @ F_{OUT}=96MHz	-	1	2	ms
		After startup, time to get lock signal, F_{IN}=2MHz @ F_{OUT}=96MHz	-	25	35	µs
Duty(1)	Duty Cycle		40	50	60	%

Note:

1. These values are based on simulation. They are not covered by production test limits or characterization.
2. These values are based on characterization.

Table 46-56. Power Consumption(1)

Symbol	Parameter	Conditions	TA	Min.	Typ.	Max.	Units
I_{DD}	Current Consumption	Ck=48MHz (PL0)	Max.85°C Typ.25°C	-	454	548	µA
		Ck=96MHz (PL2)		-	934	1052	

Note:

1. These values are based on characterization.

46.12. Timing Characteristics

46.12.1. External Reset Pin

Table 46-57. External Reset Characteristics(1)

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
t_{EXT}	Minimum reset pulse width		1	-	-	µs

46.12.2. SERCOM in SPI Mode in PL0

Table 46-58. SPI Timing Characteristics and Requirements(1)

Symbol	Parameter	Conditions		Min.	Typ.	Max.	Units
t_{SCK}	SCK period	Master, VDD>2.70V		141	153		ns
		Master, VDD>1.62V		147	159		
t_{SCKW}	SCK high/low width	Master		-	0.5*tSCK	-	
t_{SCKR}	SCK rise time(2)	Master		-	0.25*tSCK	-	
t_{SCKF}	SCK fall time(2)	Master		-	0.25*tSCK	-	
t_{MIS}	MISO setup to SCK	Master, VDD>2.70V		141			
		Master, VDD>1.62V		147			
t_{MIH}	MISO hold after SCK	Master, VDD>2.70V		0			
		Master, VDD>1.62V		0			
t_{MOS}	MOSI setup SCK	Master, VDD>2.70V				30	
		Master, VDD>1.62V				30.6	
t_{MOH}	MOSI hold after SCK	Master, VDD>2.70V		-9			
		Master, VDD>1.62V		-8.5			
t_{SSCK}	Slave SCK Period	Slave, VDD>2.70V		220	250		
		Slave, VDD>1.62V		230	250	-	
t_{SSCKW}	SCK high/low width	Slave		-	0.5*tSCK	-	
t_{SSCKR}	SCK rise time(2)	Slave		-	0.25*tSCK	-	
t_{SSCKF}	SCK fall time(2)	Slave		-	0.25*tSCK	-	
t_{SIS}	MOSI setup to SCK	Slave, VDD>2.70V		42	-	-	
		Slave, VDD>1.62V		42			
t_{SIH}	MOSI hold after SCK	Slave, VDD>2.70V		0			
		Slave, VDD>1.62V		0			
t_{SSS}	SS setup to SCK	Slave	PRELOADEN=1				
			PRELOADEN=0				
t_{SSH}	SS hold after SCK	Slave					
t_{SOS}	MISO setup before SCK	Slave, VDD>2.70V				109	
		Slave, VDD>1.62V				115	
t_{SOH}	MISO hold after SCK	Slave, VDD>2.70V		17.3			
		Slave, VDD>1.62V		17.3			
t_{SOSS}	MISO setup after SS low	Slave, VDD>2.70V				95	
		Slave, VDD>1.62V				102	

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
t_{SOSH}	MISO hold after SS high	Slave, VDD>2.70V	10.2			ns
		Slave, VDD>1.62V	10.2			

Note: 1. These values are based on simulation. They are not covered by production test limits or characterization.

Figure 46-5. SPI Timing Requirements in Master Mode

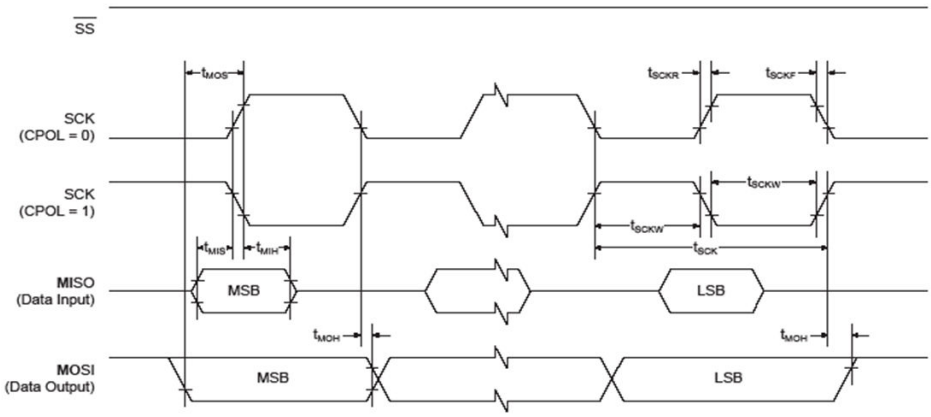
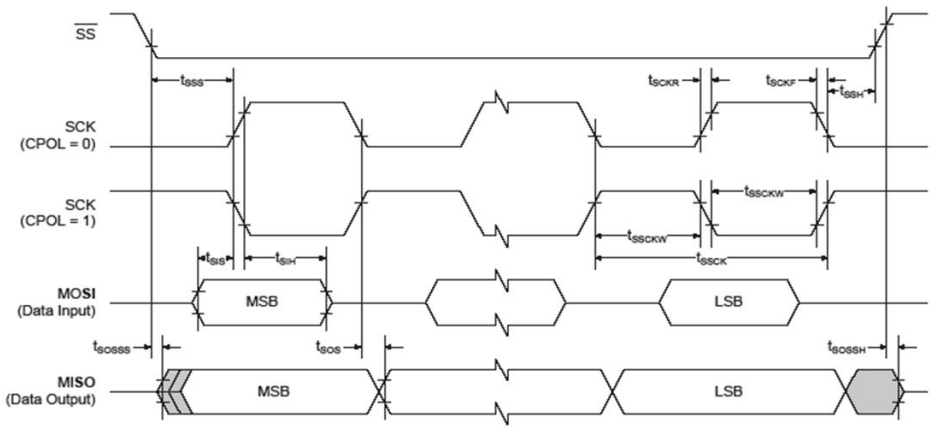


Figure 46-6. SPI Timing Requirements in Master Mode



46.12.3. SERCOM in SPI Mode in PL2

Table 46-59. SPI Timing Characteristics and Requirements(1)

Symbol	Parameter	Conditions		Min.	Typ.	Max.	Units
tSCK	SCK period	Master, VDD>2.70V		41.1	53		ns
		Master, VDD>1.62V	52.6	65			
tSCKW	SCK high/low width	Master		-	0.5*tSCK	-	
tSCKR	SCK rise time(2)	Master		-	0.25*tSCK	-	
tSCKF	SCK fall time(2)	Master		-	0.25*tSCK	-	
tMIS	MISO setup to SCK	Master, VDD>2.70V		41.1			
		Master, VDD>1.62V	52.6				
tMIH	MISO hold after SCK	Master, VDD>2.70V		0			
		Master, VDD>1.62V	0				
tMOS	MOSI setup SCK	Master, VDD>2.70V				8.5	
		Master, VDD>1.62V			13.1		
tMOH	MOSI hold after SCK	Master, VDD>2.70V		0.5			
		Master, VDD>1.62V	1				
tSSCK	Slave SCK Period	Slave, VDD>2.70V		74	90	-	
		Slave, VDD>1.62V	95	110	-		
tSSCKW	SCK high/low width	Slave		-	0.5*tSCK	-	
tSSCKR	SCK rise time(2)	Slave		-	0.25*tSCK	-	
tSSCKF	SCK fall time(2)	Slave		-	0.25*tSCK	-	
tSIS	MOSI setup to SCK	Slave, VDD>2.70V		10.3	-	-	
		Slave, VDD>1.62V	11.8	-	-		
tSIH	MOSI hold after SCK	Slave, VDD>2.70V		0	-	-	
		Slave, VDD>1.62V	0	-	-		
tSSS	SS setup to SCK	Slave	PRELOADEN=1				
			PRELOADEN=0				
tSSH	SS hold after SCK	Slave					

Symbol	Parameter	Conditions		Min.	Typ.	Max.	Units
tSOS	MISO setup before SCK	Slave, VDD>2.70V		-	-	36.9	ns
		Slave, VDD>1.62V	-	-	47.5		
tSOH	MISO hold after SCK	Slave, VDD>2.70V		11,5	-	-	
		Slave, VDD>1.62V	11,5	-	-		
tSOSS	MISO setup after SS low	Slave, VDD>2.70V		-	-	31	
		Slave, VDD>1.62V	-	-	41.3		
tSOSH	MISO hold after SS high	Slave, VDD>2.70V		6.2	-	-	
		Slave, VDD>1.62V	6.2	-	-		

Note: 1. These values are based on simulation. They are not covered by production test limits or characterization.

Figure 46-7. SPI Timing Requirements in Master Mode

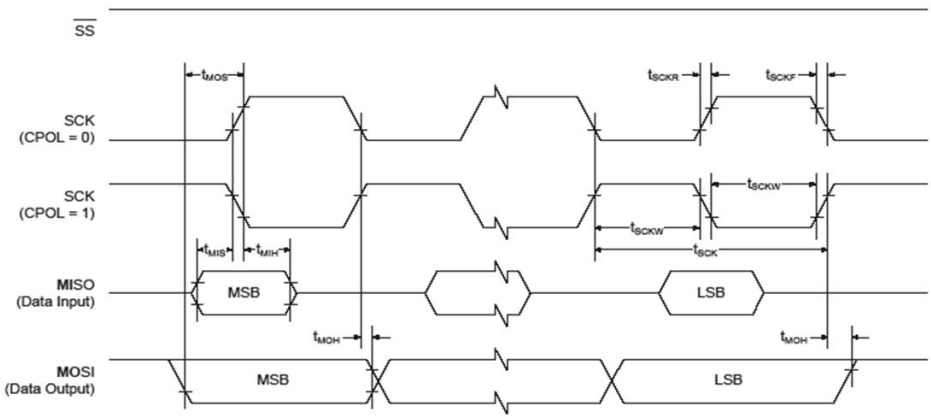
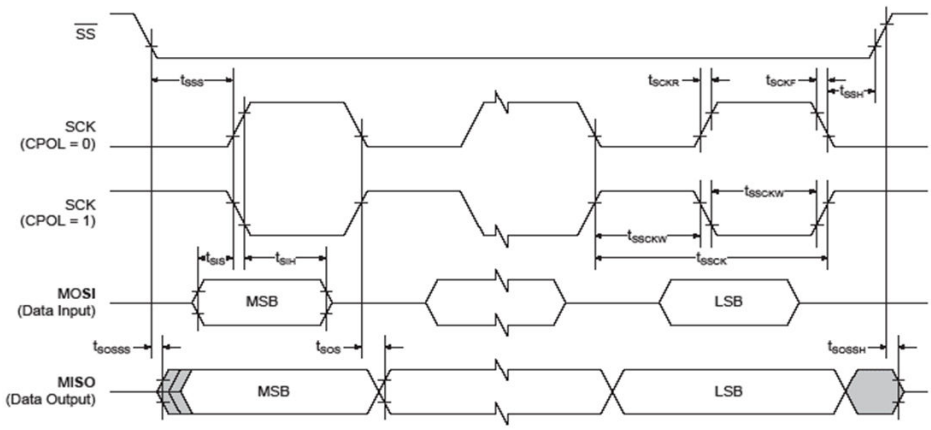


Figure 46-8. SPI Timing Requirements in Master Mode



46.13. USB Characteristics

The USB on-chip buffers comply with the Universal Serial Bus (USB) v2.0 standard. All AC parameters related to these buffers can be found within the USB 2.0 electrical specifications.

Electrical configuration required to be USB-compliant:

- the performance level must be PL2 only
- the CPU fequency must be higher 8MHz when USB is active (No contrain for USB suspend mode)
- the operating voltages must be 3.3V (Min. 3.0V, Max. 3.6V).
- the GCLK_USB frequency accuracy source must be less than:
  - in USB device mode, 48MHz +/-0.25%
  - in USB host mode, 48MHz +/-0.05%

Table 46-60. GCLK_USB Clock Setup Recommendations

Clock setup		USB Device	USB Host
DFLL48M	Open loop	No	No
	Close loop, Ref. internal OSC source	No	No
	Close loop, Ref. external XOSC source	Yes	No
	Close loop, Ref. SOF (USB recovery mode)(1)	Yes(2)	N/A
FDPLL	internal OSC (32K, 8M…)	No	No
	external OSC (<1MHz)	Yes	No
	external OSC (>1MHz)	Yes(3)	Yes

Note:

1. When using DFLL48M in USB recovery mode, the Fine Step value must be Ah to guarantee a USB clock at +/-0.25% before 11ms after a resume. Only usable in LDO regulator mode.
2. Very high signal quality and crystal less. It is the best setup for USB Device mode.
3. FDPLL lock time is short when the clock frequency source is high (> 1 MHz). Thus, FDPLL and external OSC can be stopped during USB suspend mode to reduce consumption and guarantee a USB wakeup time (See TDRSMDN in USB specification).

47. Packaging Information

47.1. Thermal Considerations

47.1.1. Thermal Resistance Data

The following table summarizes the thermal resistance data depending on the package.

Table 47-1. Thermal Resistance Data

Package Type	θ_{JA}	θ_{JC}
32-pin TQFP	68°C/W	25.8°C/W
48-pin TQFP	78.8°C/W	12.3°C/W
64-pin TQFP	66.7°C/W	11.9°C/W
32-pin QFN	37.2°C/W	3.1°C/W
48-pin QFN	31.6°C/W	10.3°C/W
64-pin QFN	32.2°C/W	10.1°C/W
64-pin WLCSP	36.8°C/W	5.0°C/W

Related Links

Junction Temperature on page 1180

47.1.2. Junction Temperature

The average chip-junction temperature, T_J, in °C can be obtained from the following:

1. $T_J = T_A + (P_D \times \theta_{JA})$
2. $T_J = T_A + (P_D \times (\theta_{HEATSINK} + \theta_{JC}))$

where:

- θ_{JA} = Package thermal resistance, Junction-to-ambient (°C/W), see Thermal Resistance Data
- θ_{JC} = Package thermal resistance, Junction-to-case thermal resistance (°C/W), see Thermal Resistance Data
- $\theta_{HEATSINK}$ = Thermal resistance (°C/W) specification of the external cooling device
- P_D = Device power consumption (W)
- T_A = Ambient temperature (°C)

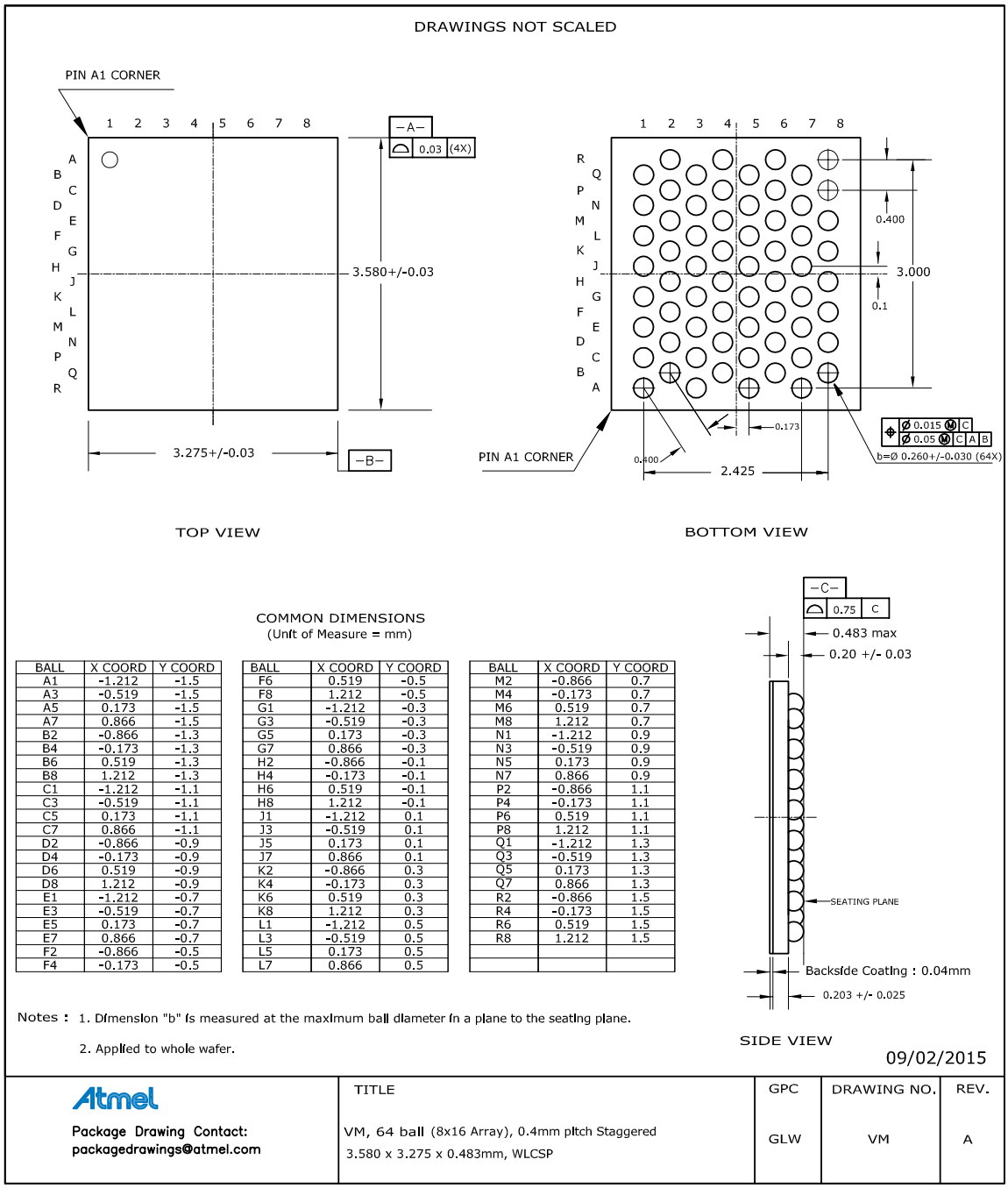
From the first equation, the user can derive the estimated lifetime of the chip and decide if a cooling device is necessary or not. If a cooling device is to be fitted on the chip, the second equation should be used to compute the resulting average chip-junction temperature T_J in °C.

Related Links

Thermal Resistance Data on page 1180

47.2. Package Drawings

47.2.1. 64-Ball WLCSP



BALL	X COORD	Y COORD
A1	-1.212	-1.5
A3	-0.519	-1.5
A5	0.173	-1.5
A7	0.866	-1.5
B2	-0.866	-1.3
B4	-0.173	-1.3
B6	0.519	-1.3
B8	1.212	-1.3
C1	-1.212	-1.1
C3	-0.519	-1.1
C5	0.173	-1.1
C7	0.866	-1.1
D2	-0.866	-0.9
D4	-0.173	-0.9
D6	0.519	-0.9
D8	1.212	-0.9
E1	-1.212	-0.7
E3	-0.519	-0.7
E5	0.173	-0.7
E7	0.866	-0.7
F2	-0.866	-0.5
F4	-0.173	-0.5

BALL	X COORD	Y COORD
F6	0.519	-0.5
F8	1.212	-0.5
G1	-1.212	-0.3
G3	-0.519	-0.3
G5	0.173	-0.3
G7	0.866	-0.3
H2	-0.866	-0.1
H4	-0.173	-0.1
H6	0.519	-0.1
H8	1.212	-0.1
J1	-1.212	0.1
J3	-0.519	0.1
J5	0.173	0.1
J7	0.866	0.1
K2	-0.866	0.3
K4	-0.173	0.3
K6	0.519	0.3
K8	1.212	0.3
L1	-1.212	0.5
L3	-0.519	0.5
L5	0.173	0.5
L7	0.866	0.5

BALL	X COORD	Y COORD
M2	-0.866	0.7
M4	-0.173	0.7
M6	0.519	0.7
M8	1.212	0.7
N1	-1.212	0.9
N3	-0.519	0.9
N5	0.173	0.9
N7	0.866	0.9
P2	-0.866	1.1
P4	-0.173	1.1
P6	0.519	1.1
P8	1.212	1.1
Q1	-1.212	1.3
Q3	-0.519	1.3
Q5	0.173	1.3
Q7	0.866	1.3
R2	-0.866	1.5
R4	-0.173	1.5
R6	0.519	1.5
R8	1.212	1.5

Table 47-2. Device and Package Maximum Weight

10	mg

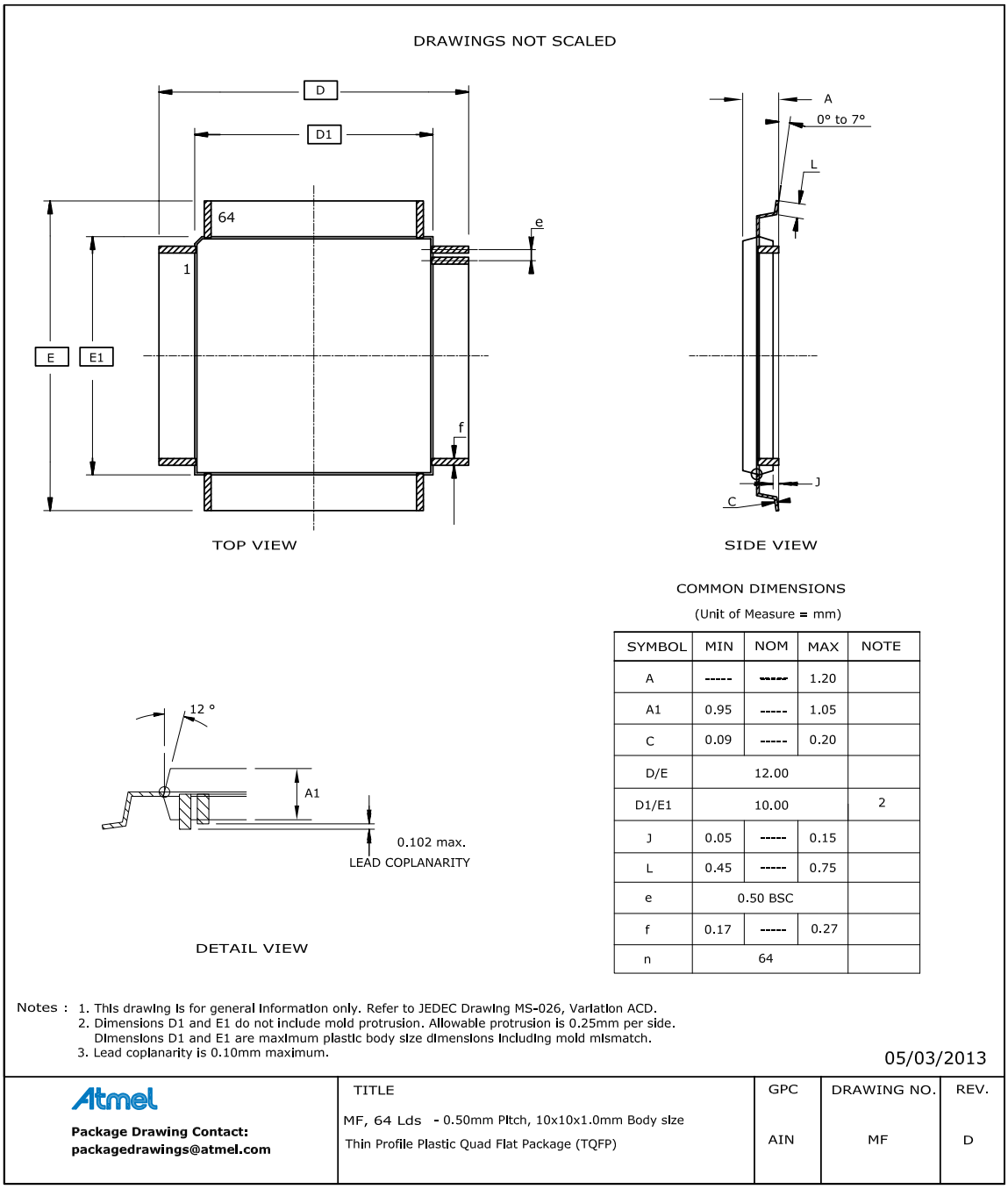
Table 47-3. Package Characteristics

Moisture Sensitivity Level	MSL1

Table 47-4. Package Reference

JEDEC Drawing Reference	N/A
JESD97 Classification	E1

47.2.2. 64 pin TQFP



COMMON DIMENSIONS

(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	-----	-----	1.20	
A1	0.95	-----	1.05	
C	0.09	-----	0.20	
D/E		12.00		
D1/E1		10.00		2
J	0.05	-----	0.15	
L	0.45	-----	0.75	
e		0.50 BSC		
f	0.17	-----	0.27	
n		64		

Notes : 1. This drawing is for general information only. Refer to JEDEC Drawing MS-026, Variation ACD.
2. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is 0.25mm per side. Dimensions D1 and E1 are maximum plastic body size dimensions including mold mismatch.
3. Lead coplanarity is 0.10mm maximum.

05/03/2013

	TITLE	GPC	DRAWING NO.	REV.
Atmel **Package Drawing Contact:** **packagedrawings@atmel.com**	MF, 64 Lds - 0.50mm Pitch, 10x10x1.0mm Body size Thin Profile Plastic Quad Flat Package (TQFP)	AIN	MF	D

Table 47-5. Device and Package Maximum Weight

300	mg

Table 47-6. Package Characteristics

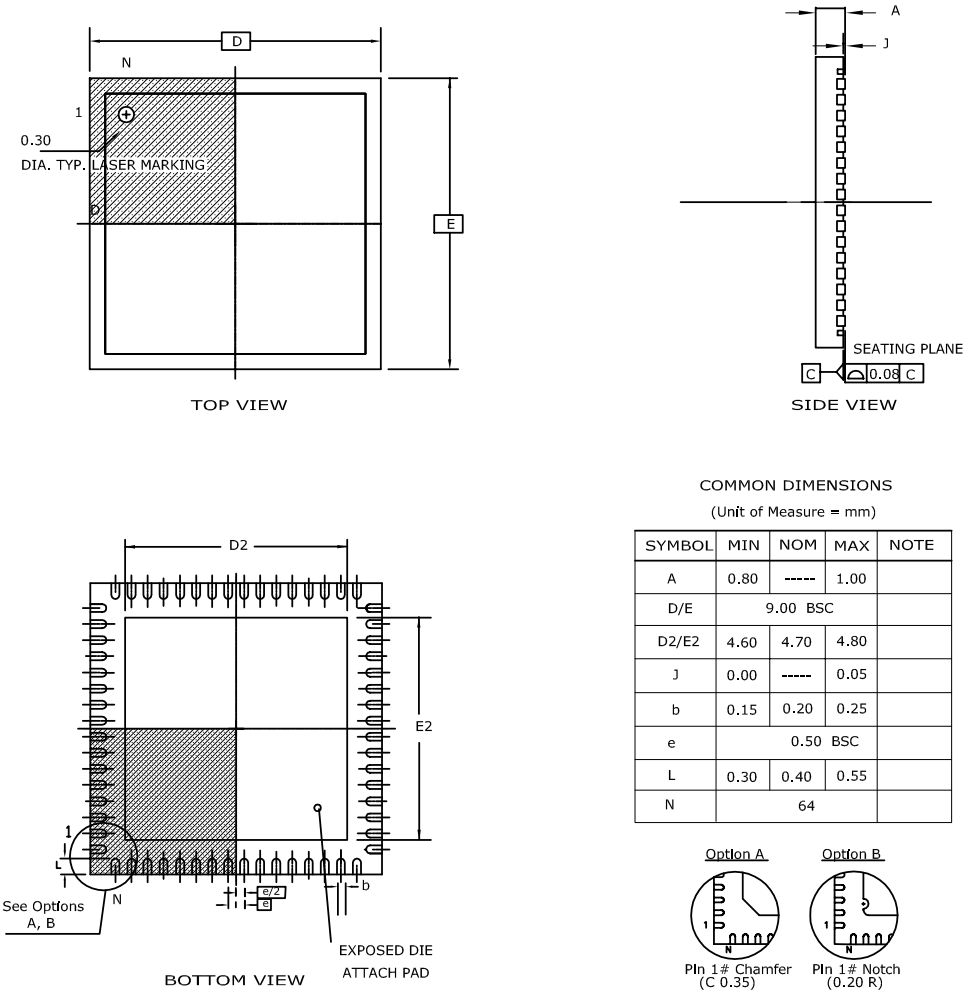
Moisture Sensitivity Level	MSL3

Table 47-7. Package Reference

JEDEC Drawing Reference	MS-026
JESD97 Classification	E3

47.2.3. 64 pin QFN

DRAWINGS NOT SCALED



COMMON DIMENSIONS

(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	0.80	-----	1.00	
D/E		9.00 BSC		
D2/E2	4.60	4.70	4.80	
J	0.00	-----	0.05	
b	0.15	0.20	0.25	
e		0.50 BSC		
L	0.30	0.40	0.55	
N		64		

Notes : 1. This drawing is for general information only. Refer to JEDEC Drawing MO-220
2. Dimension b applies to metallized terminal and is measured between 0.15mm and 0.30mm from the terminal tip.
If the terminal has the optical radius on the other end of the terminal, the dimension should not be measured in that radius area.

Note: The exposed die attach pad is not connected electrically inside the device.

Table 47-8. Device and Package Maximum Weight

200	mg

Table 47-9. Package Charateristics

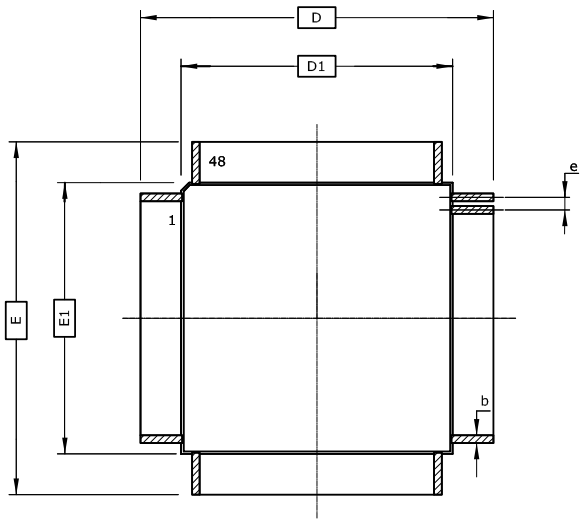
Moisture Sensitivity Level	MSL3

Table 47-10. Package Reference

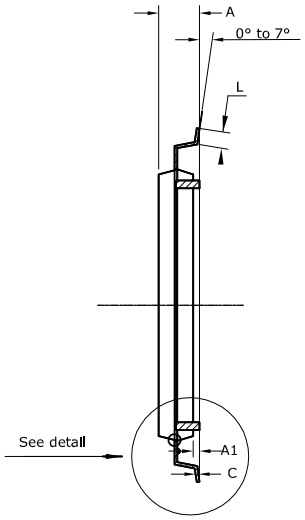
JEDEC Drawing Reference	MO-220
JESD97 Classification	E3

47.2.4. 48 pin TQFP

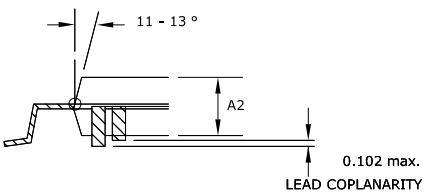
DRAWINGS NOT SCALED



TOP VIEW



SIDE VIEW



DETAIL VIEW

COMMON DIMENSIONS

(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	-----	-----	1.20	
A1	0.05	-----	0.15	
A2	0.95	-----	1.05	
C	0.09	-----	0.20	
D/E	9.00 BSC			
D1/E1	7.00 BSC			
L	0.45	-----	0.75	
b	0.17	-----	0.27	
e	0.50 BSC			

Notes : 1. This drawing is for general information only. Refer to JEDEC Drawing MS-026, Variation ABC.
2. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is 0.25mm per side. Dimensions D1 and E1 are maximum plastic body size dimensions including mold mismatch.
3. Lead coplanarity is 0.10mm maximum.

Table 47-11. Device and Package Maximum Weight

140	mg

Table 47-12. Package Characteristics

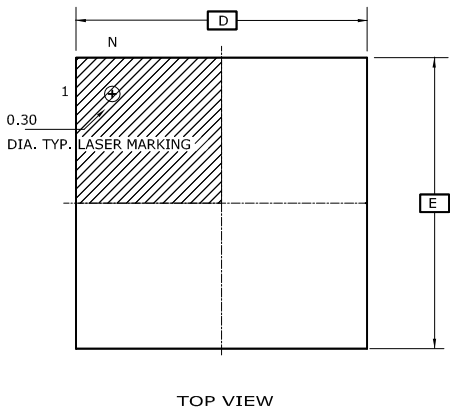
Moisture Sensitivity Level	MSL3

Table 47-13. Package Reference

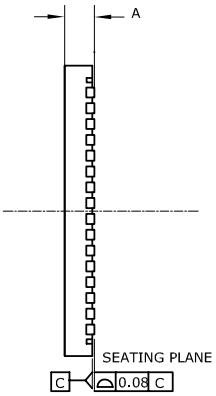
JEDEC Drawing Reference	MS-026
JESD97 Classification	E3

47.2.5. 48 pin QFN

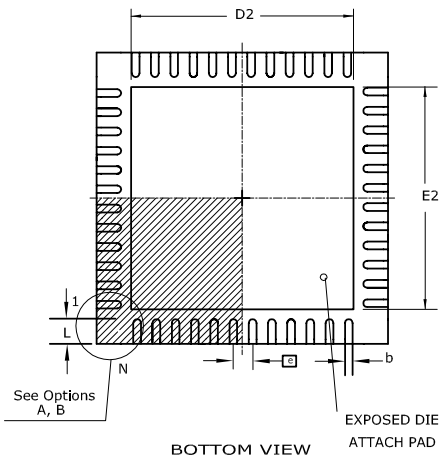
DRAWINGS NOT SCALED



TOP VIEW



SIDE VIEW

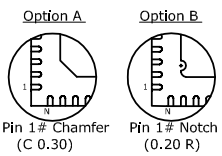


BOTTOM VIEW

COMMON DIMENSIONS

(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	0.80	0.85	0.90	
D/E		7.00 BSC		
D2/E2	5.05	5.15	5.25	
b	0.18	0.25	0.30	
e		0.50 BSC		
L	0.30	0.40	0.50	
N		48		



Notes : 1. This drawing is for general information only. Refer to JEDEC Drawing MO-220, Variation VKKD-4, for proper dimensions, tolerances, datums, etc.
2. Dimension b applies to metallized terminal and is measured between 0.15mm and 0.30mm from the terminal tip.
If the terminal has the optical radius on the other end of the terminal, the dimension should not be measured in that radius area.

Note: The exposed die attach pad is not connected electrically inside the device.

Table 47-14. Device and Package Maximum Weight

140	mg

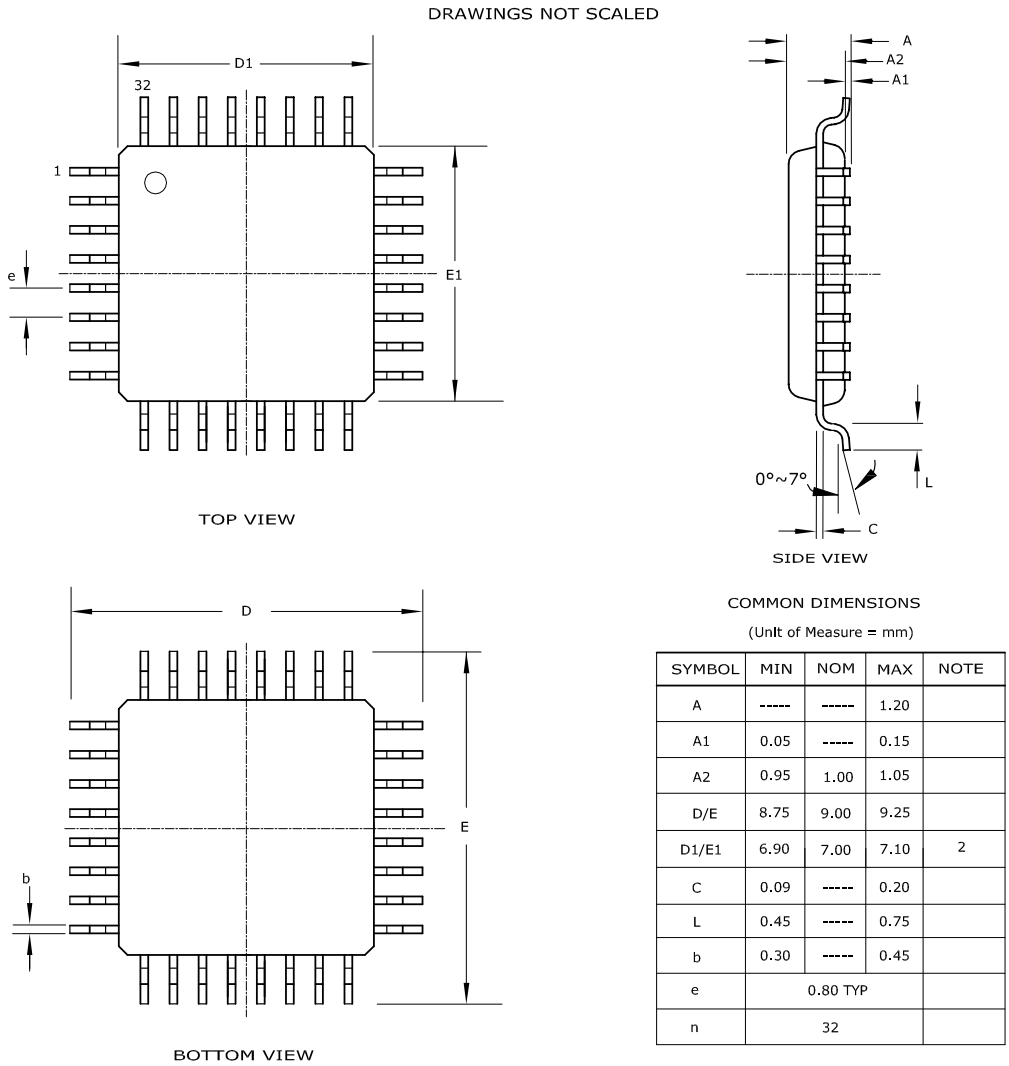
Table 47-15. Package Characteristics

Moisture Sensitivity Level	MSL3

Table 47-16. Package Reference

JEDEC Drawing Reference	MO-220
JESD97 Classification	E3

47.2.6. 32 pin TQFP



COMMON DIMENSIONS

(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	-----	------	1.20	
A1	0.05	-----	0.15	
A2	0.95	1.00	1.05	
D/E	8.75	9.00	9.25	
D1/E1	6.90	7.00	7.10	2
C	0.09	-----	0.20	
L	0.45	-----	0.75	
b	0.30	-----	0.45	
e	0.80 TYP			
n	32			

Notes : 1. This drawing is for general information only. Refer to JEDEC Drawing MS-026, Variation ABA.
2. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is 0.25mm per side. Dimensions D1 and E1 are maximum plastic body size dimensions including mold mismatch.
3. Lead coplanarity is 0.10mm maximum.

Table 47-17. Device and Package Maximum Weight

100	mg

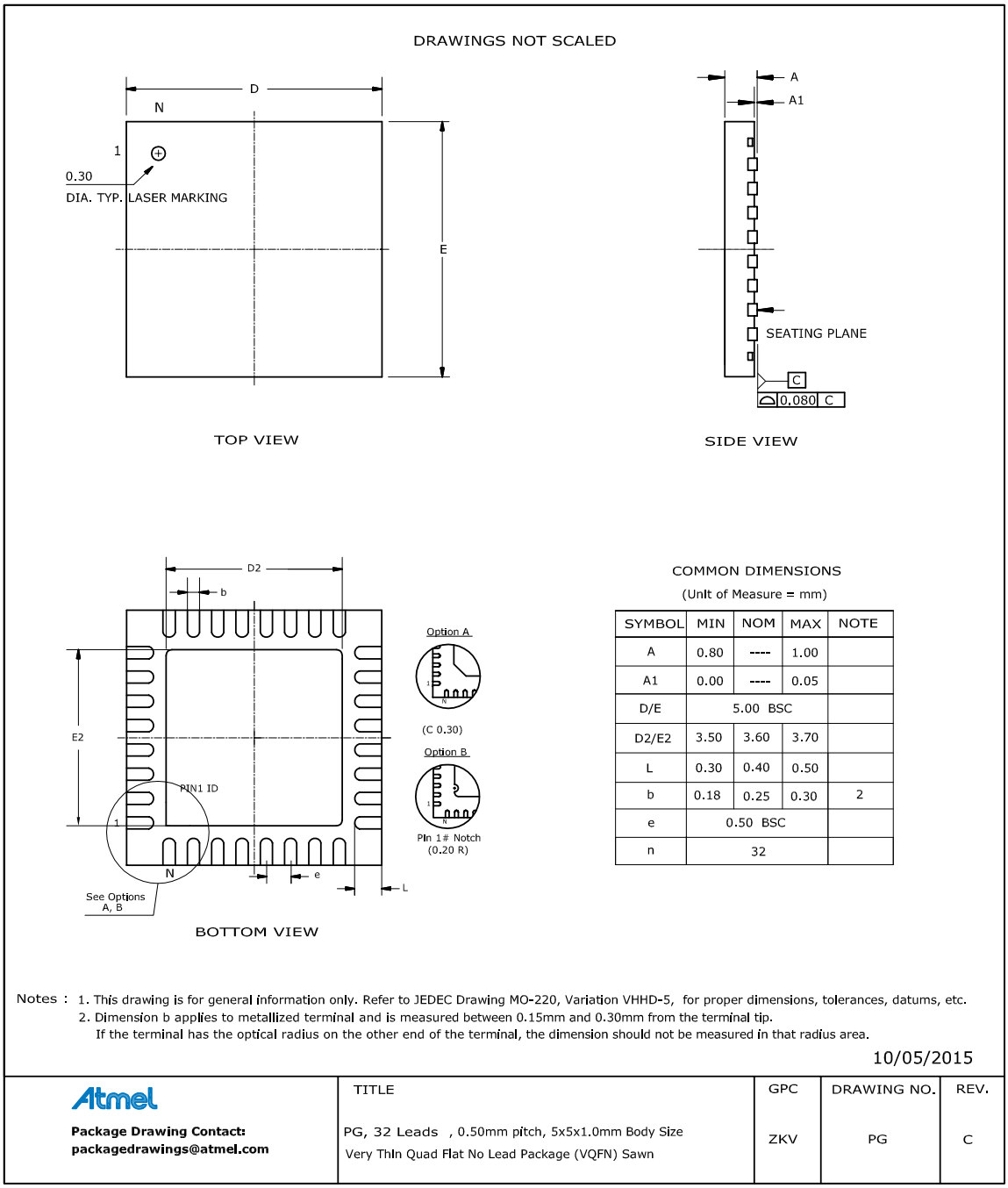
Table 47-18. Package Charateristics

Moisture Sensitivity Level	MSL3

Table 47-19. Package Reference

JEDEC Drawing Reference	MS-026
JESD97 Classification	E3

47.2.7. 32 pin QFN



Note: The exposed die attach pad is connected inside the device to GND and GNDANA.

Table 47-20. Device and Package Maximum Weight

90	mg

Table 47-21. Package Characteristics

Moisture Sensitivity Level	MSL3

Table 47-22. Package Reference

JEDEC Drawing Reference	MO-220
JESD97 Classification	E3

47.3. Soldering Profile

The following table gives the recommended soldering profile from J-STD-20.

Table 47-23.

Profile Feature	Green Package
Average Ramp-up Rate (217°C to peak)	3°C/s max.
Preheat Temperature 175°C ±25°C	150-200°C
Time Maintained Above 217°C	60-150s
Time within 5°C of Actual Peak Temperature	30s
Peak Temperature Range	260°C
Ramp-down Rate	6°C/s max.
Time 25°C to Peak Temperature	8 minutes max.

A maximum of three reflow passes is allowed per component.

48. Schematic Checklist

48.1. Introduction

This chapter describes a common checklist which should be used when starting and reviewing the schematics for a SAM L21 design. This chapter illustrates recommended power supply connections, how to connect external analog references, programmer, debugger, oscillator and crystal.

48.2. Power Supply

The SAM L21 supports a single or dual power supply from 1.62V to 3.63V. The same voltage must be applied to both VDDIN and VDDANA.

The internal voltage regulator has four different modes:

- Linear mode: this mode does not require any external inductor. This is the default mode when CPU and peripherals are running
- Switching mode (Buck): the most efficient mode when the CPU and peripherals are running.
- Low Power (LP) mode: This is the default mode used when the chip is in standby mode
- Shutdown mode: When the chip is in backup mode, the internal regulator is turned off

Selecting between switching mode and linear mode can be done by software on the fly, but the power supply must be designed according to which mode is to be used.

48.2.1. Power Supply Connections

The following figures shows the recommended power supply connections for switched/linear mode, linear mode only and with battery backup.

Figure 48-1. Power Supply Connection for Switching/Linear Mode

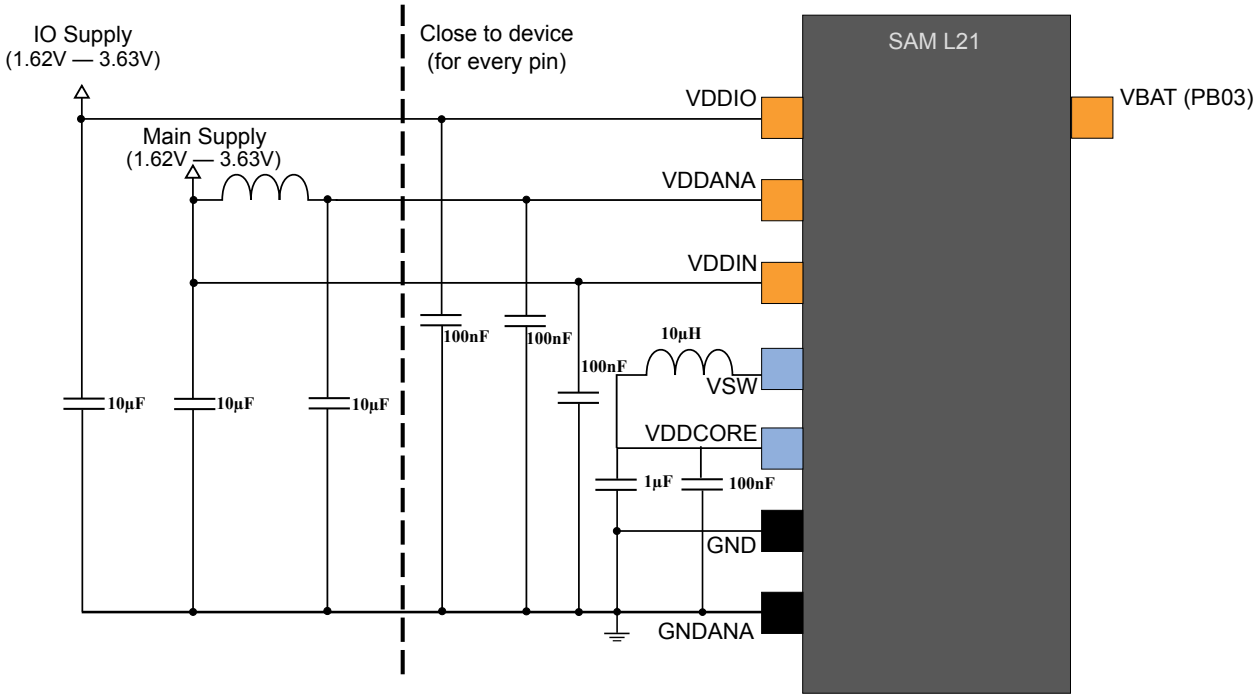


Figure 48-2. Power Supply Connection for Linear Mode Only

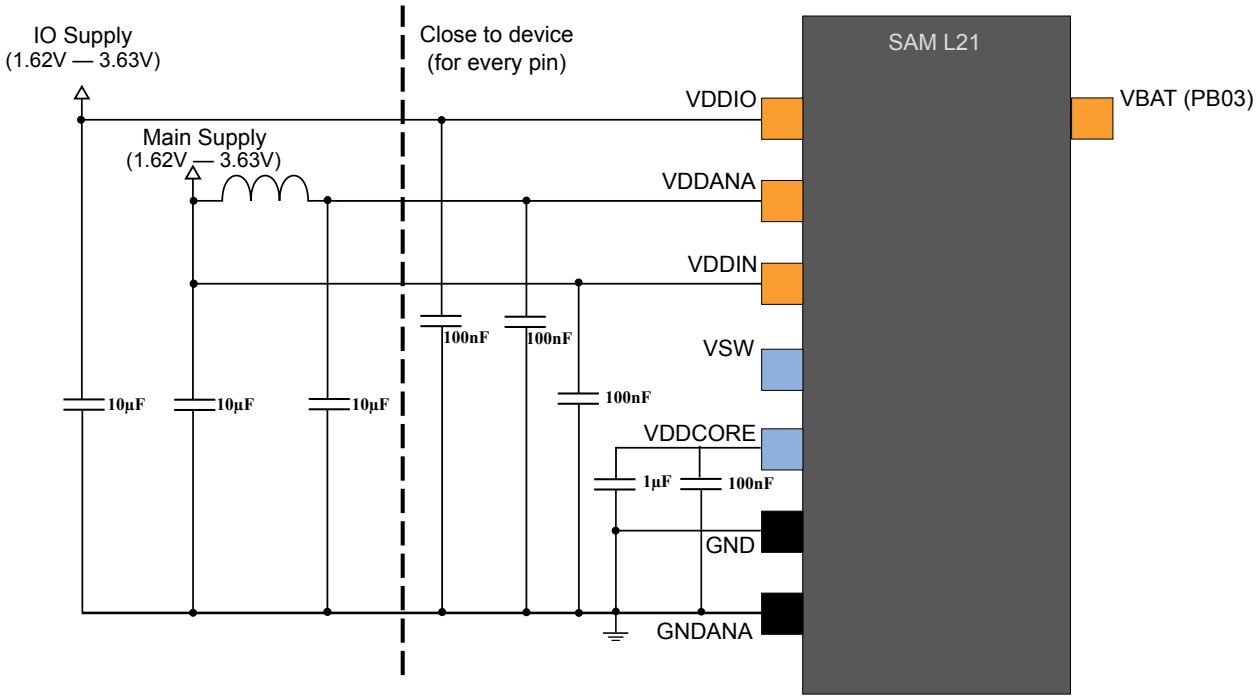


Figure 48-3. Power Supply Connection for Battery Backup

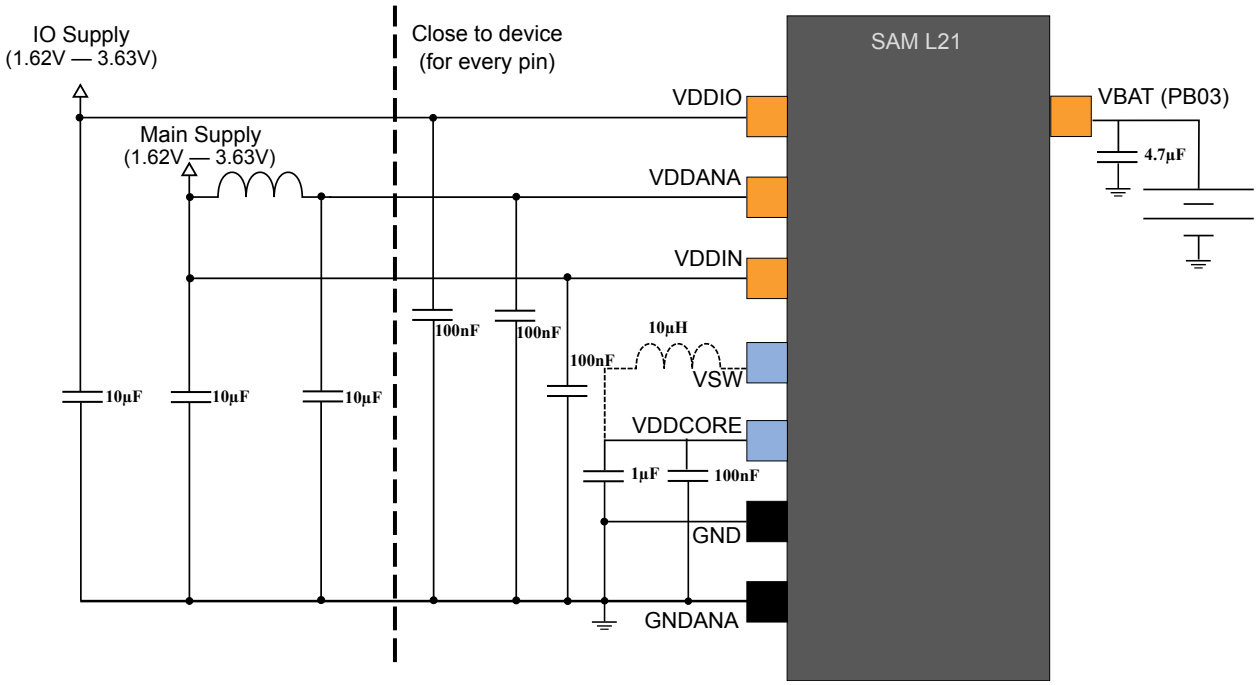


Table 48-1. Power Supply Connections, V_{DDCORE} or V_{SW} From Internal Regulator

Signal Name	Recommended Pin Connection	Description
V_{DDIO}	1.6V to 3.6V Decoupling/filtering capacitors 100nF(1)(2) and 10µF(1) Decoupling/filtering inductor 10µH(1)(3)	Digital supply voltage
V_{DDANA}	1.6V to 3.6V Decoupling/filtering capacitors 100nF(1)(2) and 10µF(1) Ferrite bead(4) prevents the V_{DD} noise interfering with V_{DDANA}	Analog supply voltage
V_{DDIN}	1.6V to 3.6V Decoupling/filtering capacitors 100nF(1)(2) and 10µF(1) Decoupling/filtering inductor 10µH(1)(3)	Digital supply voltage
V_{BAT}	1.6V to 3.6V when connected Decoupling/filtering capacitor 4.7µF(1)(2)	External battery supply input
V_{DDCORE}	0.9V to 1.2V typical Decoupling/filtering capacitors 100nF(1)(2) and 1µF(1)	Linear regulator mode: Core supply voltage output/ external decoupling pin Switched regulator mode: Core supply voltage input, must be connected to V_{SW} via inductor
V_{SW}	Switching regulator mode: 10 µH inductor with saturation current above 150mA and DCR<1Ω Linear regulator mode: Not connected	On-chip switching mode regulator output
GND		Ground
GND_{ANA}		Ground for the analog power domain

1. These values are only given as a typical example.

2. Decoupling capacitors should be placed close to the device for each supply pin pair in the signal group, low ESR capacitors should be used for better decoupling.

3. An inductor should be added between the external power and the V_{DD} for power filtering.

4. A ferrite bead has better filtering performance compared to standard inductor at high frequencies. A ferrite bead can be added between the main power supply (V_{DD}) and V_{DDANA} to prevent digital noise from entering the analog power domain. The bead should provide enough impedance (e.g. 50Ω at 20MHz and 220Ω at 100MHz) to separate the digital and analog power domains. Make sure to select a ferrite bead designed for filtering applications with a low DC resistance to avoid a large voltage drop across the ferrite bead.

48.2.2. Special Considerations for QFN Packages

The QFN package has an exposed paddle that must be connected to GND.

48.3. External Analog Reference Connections

The following schematic checklist is only necessary if the application is using one or more of the external analog references. If the internal references are used instead, the following circuits in Figure 48-4 External Analog Reference Schematic With Two References on page 1192 and Figure 48-5 External Analog Reference Schematic With One Reference on page 1192 are not necessary.

Figure 48-4. External Analog Reference Schematic With Two References

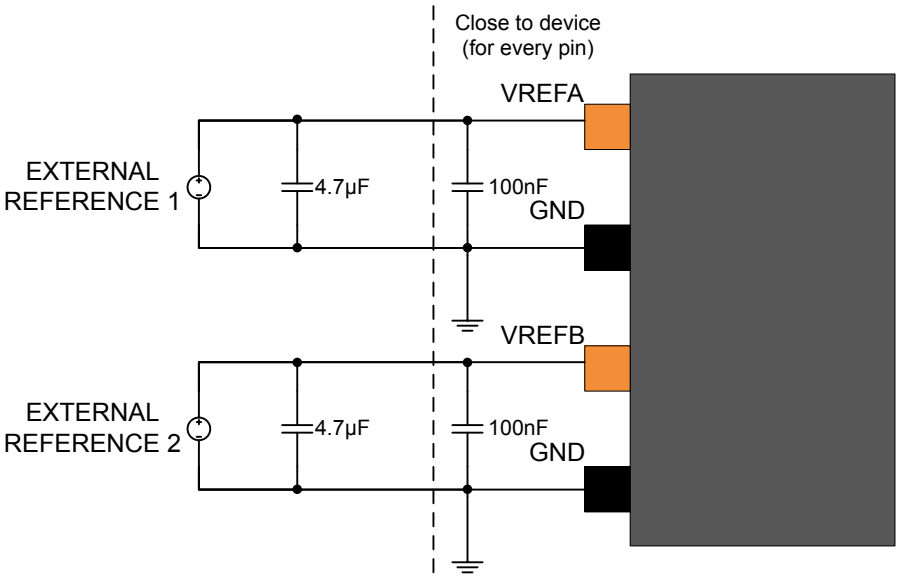


Figure 48-5. External Analog Reference Schematic With One Reference

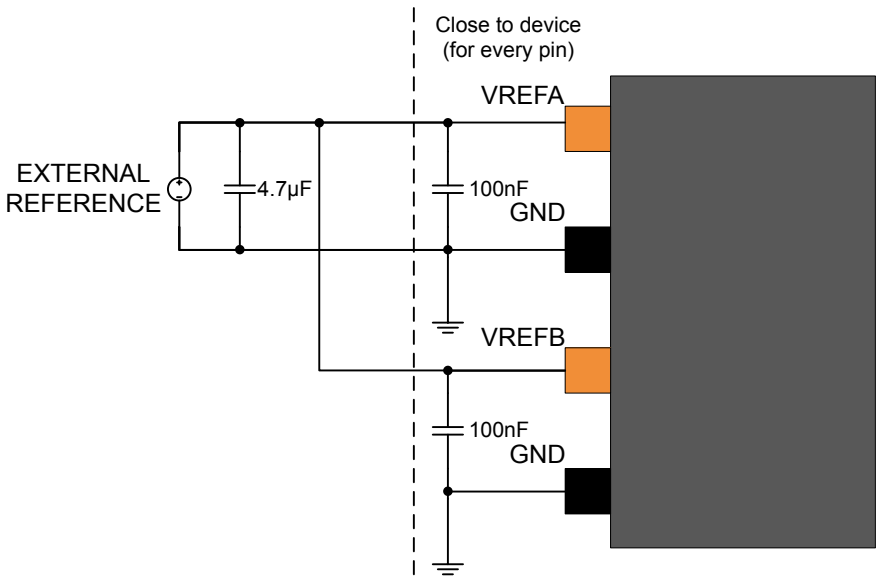


Table 48-2. External Analog Reference Connections

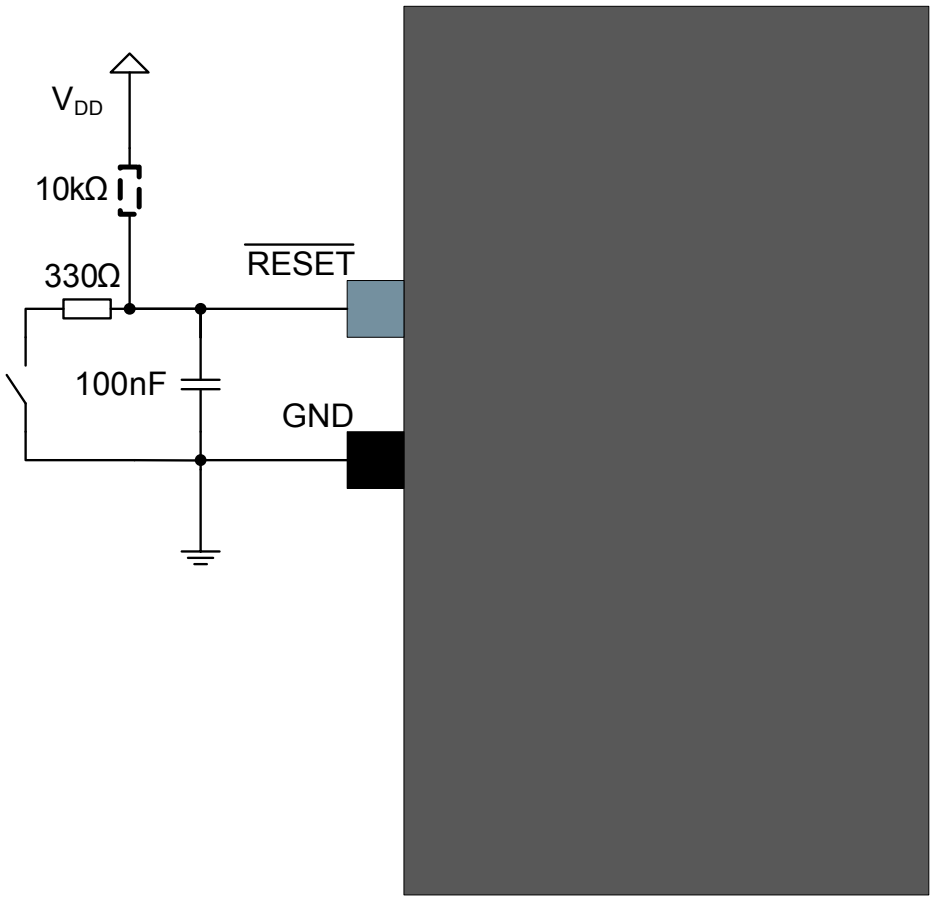
Signal Name	Recommended Pin Connection	Description
VREFx	1.0V to (V_{DDANA} - 0.6V) for ADC 1.0V to (V_{DDANA} - 0.6V) for DAC Decoupling/filtering capacitors 100nF[1][2] and 4.7µF[1]	External reference VREFx for the analog port
GND		Ground

1. These values are only given as a typical example.
2. Decoupling capacitor should be placed close to the device for each supply pin pair in the signal group.

48.4. External Reset Circuit

The external reset circuit should be connected to the $\overline{\text{RESET}}$ pin when the external reset function is used. If the external reset function has been disabled, the circuit is not necessary. The reset switch can also be removed, if the manual reset is not necessary. The $\overline{\text{RESET}}$ pin itself has an internal pull-up resistor, hence it is optional to add any external pull-up resistor.

Figure 48-6. External Reset Circuit Schematic



A pull-up resistor makes sure that the reset does not go low and unintentionally causing a device reset. An additional resistor has been added in series with the switch to safely discharge the filtering capacitor,

i.e. preventing a current surge when shorting the filtering capacitor which again can cause a noise spike that can have a negative effect on the system.

Table 48-3. Reset Circuit Connections

Signal Name	Recommended Pin Connection	Description
$\overline{\text{RESET}}$	Reset low level threshold voltage V_{DDIO} = 1.6V - 2.0V: Below 0.33 * V_{DDIO} V_{DDIO} = 2.7V - 3.6V: Below 0.36 * V_{DDIO} Decoupling/filter capacitor 100nF(1) Pull-up resistor 10kΩ(1)(2) Resistor in series with the switch 330Ω(1)	Reset pin

1. These values are only given as a typical example.

2. The SAM L21 features an internal pull-up resistor on the $\overline{\text{RESET}}$ pin, hence an external pull-up is optional.

48.5. Unused or Unconnected Pins

For unused pins the default state of the pins will give the lowest current leakage. There is thus no need to do any configuration of the unused pins in order to lower the power consumption.

48.6. Clocks and Crystal Oscillators

The SAM L21 can be run from internal or external clock sources, or a mix of internal and external sources. An example of usage can be to use the internal 16MHz oscillator as source for the system clock and an external 32.768kHz watch crystal as clock source for the Real-Time counter (RTC).

48.6.1. External Clock Source

Figure 48-7. External Clock Source Schematic

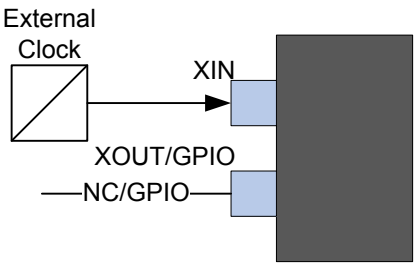
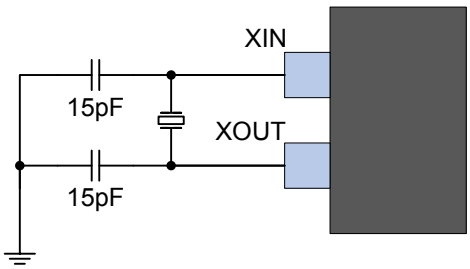


Table 48-4. External Clock Source Connections

Signal Name	Recommended Pin Connection	Description
XIN	XIN is used as input for an external clock signal	Input for inverting oscillator pin
XOUT/GPIO	Can be left unconnected or used as normal GPIO	NC/GPIO

48.6.2. Crystal Oscillator

Figure 48-8. Crystal Oscillator Schematic



The crystal should be located as close to the device as possible. Long signal lines may cause too high load to operate the crystal, and cause crosstalk to other parts of the system.

Table 48-5. Crystal Oscillator Checklist

Signal Name	Recommended Pin Connection	Description
XIN	Load capacitor 15pF(1)(2)	External crystal between 0.4 to 30MHz
XOUT	Load capacitor 15pF(1)(2)	

1. These values are only given as a typical example.

2. Decoupling capacitor should be placed close to the device for each supply pin pair in the signal group.

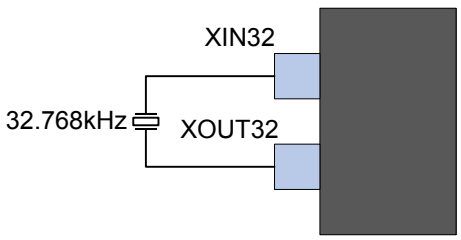
48.6.3. External Real Time Oscillator

The low frequency crystal oscillator is optimized for use with a 32.768kHz watch crystal. When selecting crystals, load capacitance and the crystal's Equivalent Series Resistance (ESR) must be taken into consideration. Both values are specified by the crystal vendor.

SAM L21 oscillator is optimized for very low power consumption, hence close attention should be made when selecting crystals.

The Low-frequency Crystal Oscillator provides an internal load capacitance of typical values available in the Electrical Characteristics section. This internal load capacitance and PCB capacitance can allow using a crystal inferior to 12.5pF load capacitance without external capacitors as shown in Figure 48-9 External Real Time Oscillator without Load Capacitor on page 1195.

Figure 48-9. External Real Time Oscillator without Load Capacitor



To improve Crystal accuracy and Safety Factor, the crystal datasheet can recommend adding external capacitors as shown in Figure 48-10 External Real Time Oscillator with Load Capacitor on page 1196.

To find suitable load capacitance for a 32.768kHz crystal, consult the crystal datasheet.

Figure 48-10. External Real Time Oscillator with Load Capacitor

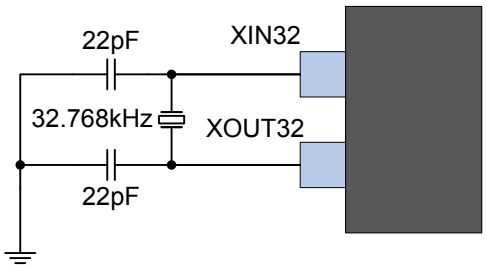


Table 48-6. External Real Time Oscillator Checklist

Signal Name	Recommended Pin Connection	Description
XIN32	Load capacitor 22pF(1)(2)	Timer oscillator input
XOUT32	Load capacitor 22pF(1)(2)	Timer oscillator output

1. These values are only given as typical examples.

2. Decoupling capacitor should be placed close to the device for each supply pin pair in the signal group.

Note: To improve the cycle-to-cycle jitter of XOSC32, it is recommended to keep the neighboring pins of XIN32 and XOUT32 as static as possible.

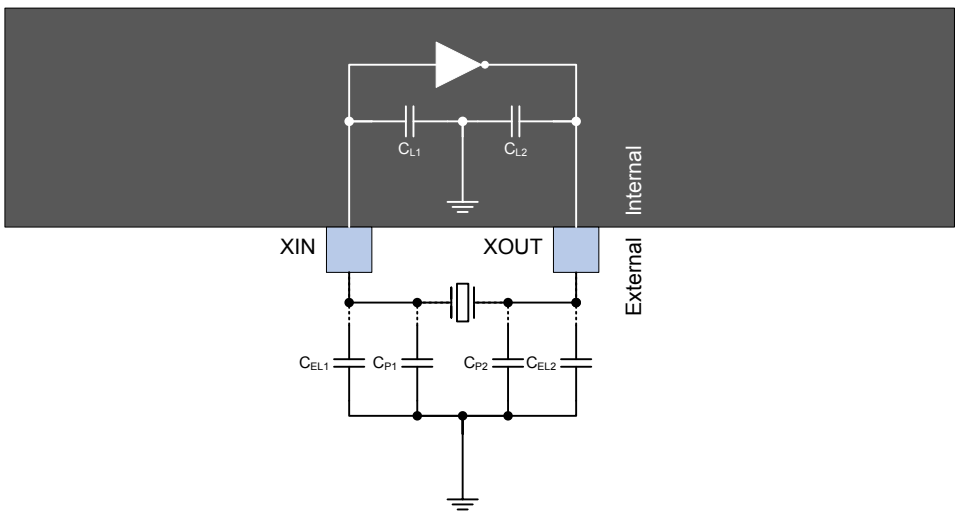
Related Links

Oscillator Pinout on page 31

48.6.4. Calculating the Correct Crystal Decoupling Capacitor

The model shown in Figure 48-11 Crystal Circuit With Internal, External and Parasitic Capacitance on page 1196 can be used to calculate correct load capacitor for a given crystal. This model includes internal capacitors C_{Ln}, external parasitic capacitance C_{ELn} and external load capacitance C_{Pn}.

Figure 48-11. Crystal Circuit With Internal, External and Parasitic Capacitance



Using this model the total capacitive load for the crystal can be calculated as shown in the equation below:

$$\sum C_{\text{tot}} = \frac{(C_{L1} + C_{P1} + C_{\text{EL1}})(C_{L2} + C_{P2} + C_{\text{EL2}})}{C_{L1} + C_{P1} + C_{\text{EL1}} + C_{L2} + C_{P2} + C_{\text{EL2}}}$$

where C_{tot} is the total load capacitance seen by the crystal. This value should be equal to the load capacitance value found in the crystal manufacturer datasheet.

The parasitic capacitance C_{ELn} can in most applications be disregarded as these are usually very small. If accounted for, these values are dependent on the PCB material and PCB layout.

For some crystal the internal capacitive load provided by the device itself can be enough. To calculate the total load capacitance in this case. C_{ELn} and C_{Pn} are both zero, $C_{L1} = C_{L2} = C_L$, and the equation reduces to the following:

$$\sum C_{\text{tot}} = \frac{C_L}{2}$$

See the related links for equivalent internal pin capacitance values.

Related Links

External 32KHz Crystal Oscillator (XOSC32K) Characteristics on page 1169

48.7. Programming and Debug Ports

For programming and/or debugging the SAM L21, the device should be connected using the Serial Wire Debug, SWD, interface. Currently the SWD interface is supported by several Atmel and third party programmers and debuggers, like the SAM-ICE, JTAGICE3 or SAM L21 Xplained Pro (SAM L21 evaluation kit) Embedded Debugger.

Refer to the SAM-ICE, JTAGICE3 or SAM L21 Xplained Pro user guides for details on debugging and programming connections and options. For connecting to any other programming or debugging tool, refer to that specific programmer or debugger's user guide.

The SAM L21 Xplained Pro evaluation board supports programming and debugging through the onboard embedded debugger so no external programmer or debugger is needed.

48.7.1. Cortex Debug Connector (10-pin)

For debuggers and/or programmers that support the Cortex Debug Connector (10-pin) interface the signals should be connected as shown in Figure 48-12 Cortex Debug Connector (10-pin) on page 1198 with details described in Table 48-7 Cortex Debug Connector (10-pin) on page 1198.

Figure 48-12. Cortex Debug Connector (10-pin)

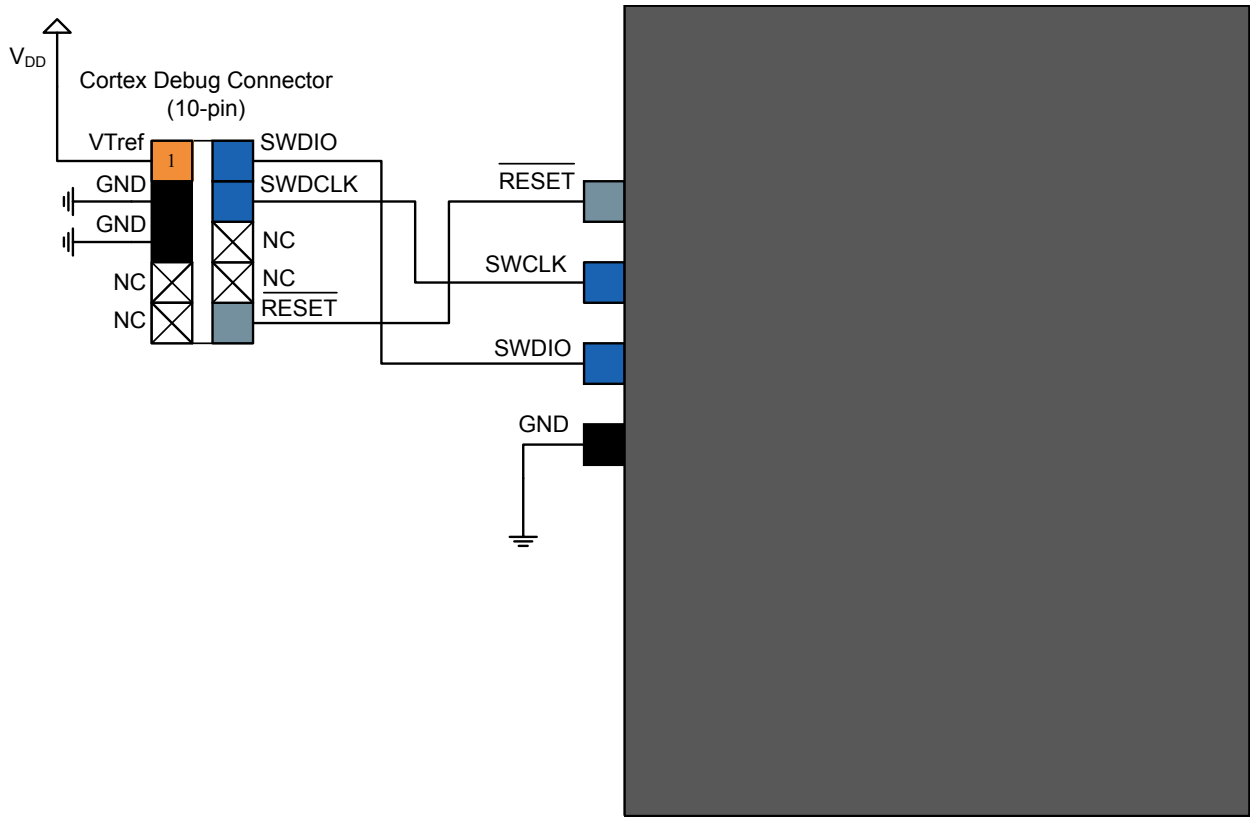


Table 48-7. Cortex Debug Connector (10-pin)

Header Signal Name	Description
SWDCLK	Serial wire clock pin
SWDIO	Serial wire bidirectional data pin
$\overline{\text{RESET}}$	Target device reset pin, active low
VTref	Target voltage sense, should be connected to the device V_{DD}
GND	Ground

48.7.2. 10-pin JTAGICE3 Compatible Serial Wire Debug Interface

The JTAGICE3 debugger and programmer does not support the Cortex Debug Connector (10-pin) directly, hence a special pinout is needed to directly connect the SAM L21 to the JTAGICE3, alternatively one can use the JTAGICE3 squid cable and manually match the signals between the JTAGICE3 and SAM L21. Figure 48-13 10-pin JTAGICE3 Compatible Serial Wire Debug Interface on page 1199 describes how to connect a 10-pin header that support connecting the JTAGICE3 directly to the SAM L21 without the need for a squid cable. This can also be used for the Atmel-ICE AVR connector port.

The JTAGICE3 squid cable or the JTACICE3 50mil cable can be used to connect the JTAGICE3 programmer and debugger to the SAM L21. 10-pin JTAGICE3 Compatible Serial Wire Debug Interface on page 1198 illustrates the correct pinout for the JTAGICE3 50 mil, and details are given in Table 48-8 10-pin JTAGICE3 Compatible Serial Wire Debug Interface on page 1199.

Figure 48-13. 10-pin JTAGICE3 Compatible Serial Wire Debug Interface

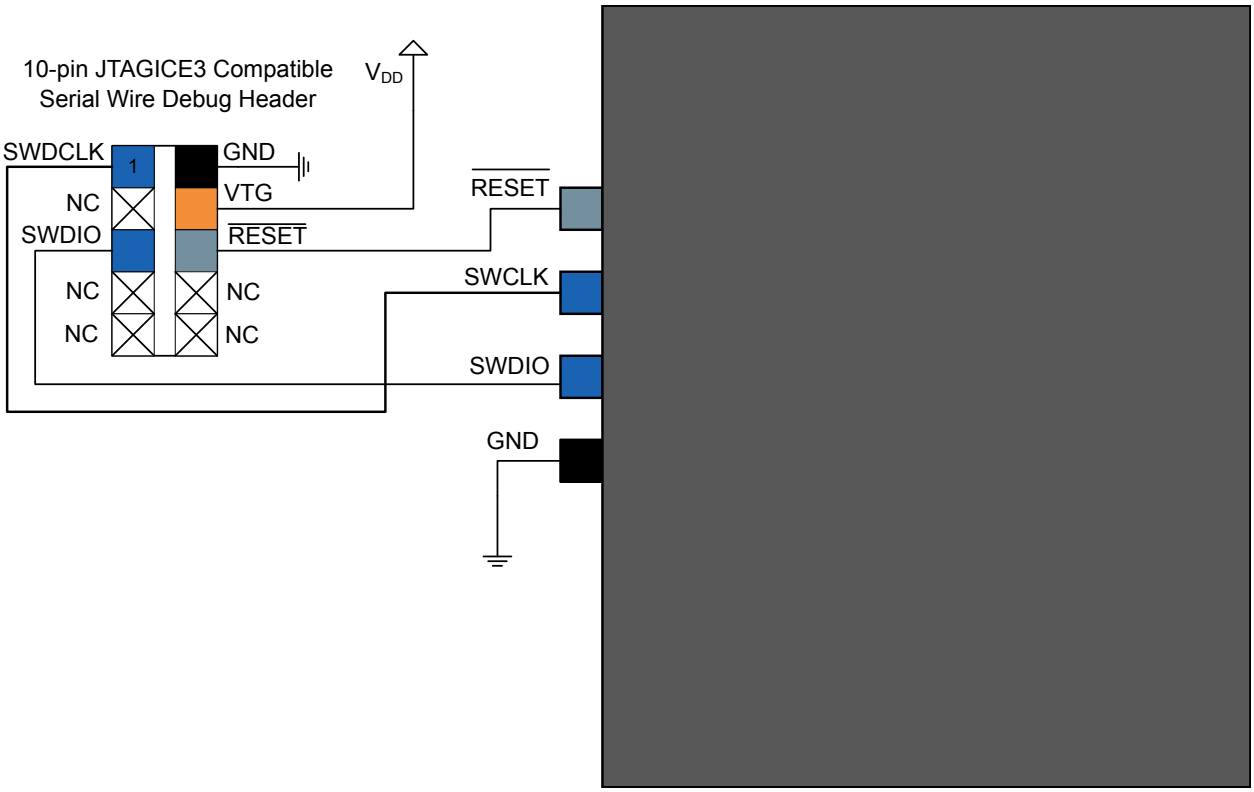


Table 48-8. 10-pin JTAGICE3 Compatible Serial Wire Debug Interface

Header Signal Name	Description
SWDCLK	Serial wire clock pin
SWDIO	Serial wire bidirectional data pin
$\overline{\text{RESET}}$	Target device reset pin, active low
VTG	Target voltage sense, should be connected to the device V_{DD}
GND	Ground

48.7.3. 20-pin IDC JTAG Connector

For debuggers and/or programmers that support the 20-pin IDC JTAG Connector, e.g. the SAM-ICE, the signals should be connected as shown in Figure 48-14 20-pin IDC JTAG Connector on page 1200 with details described in Table 48-9 20-pin IDC JTAG Connector on page 1200.

Figure 48-14. 20-pin IDC JTAG Connector

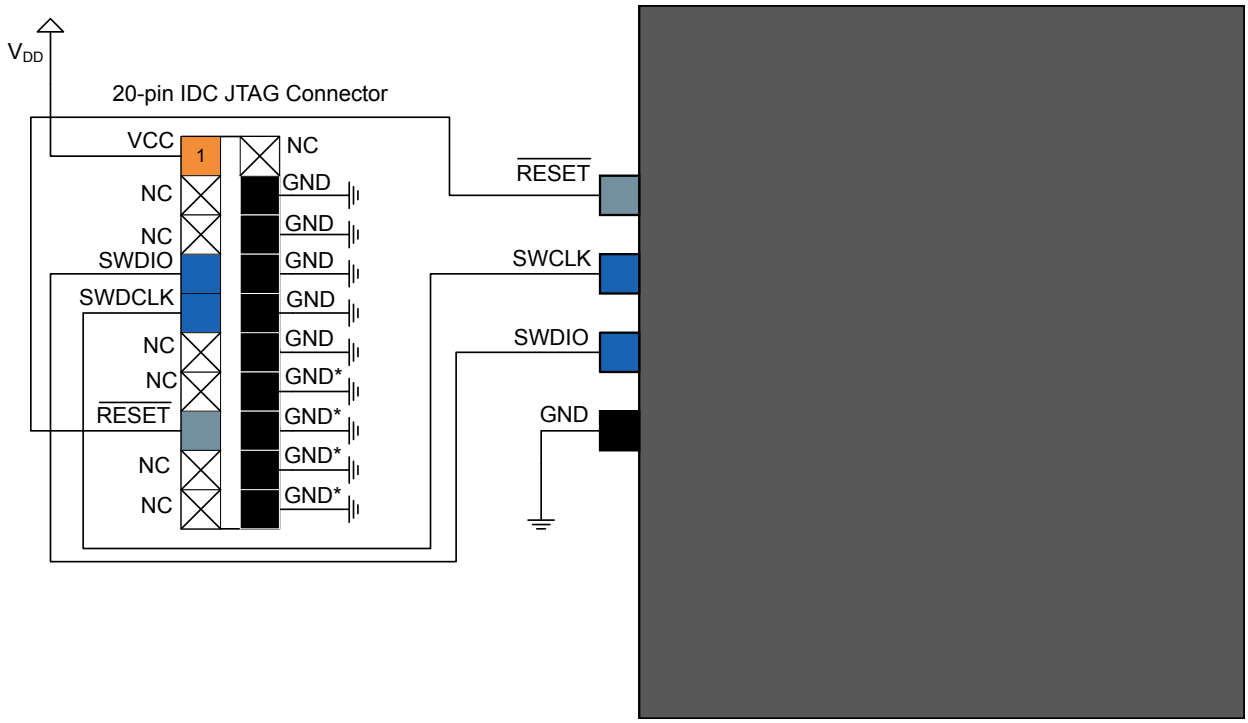


Table 48-9. 20-pin IDC JTAG Connector

Header Signal Name	Description
SWDCLK	Serial wire clock pin
SWDIO	Serial wire bidirectional data pin
$\overline{\text{RESET}}$	Target device reset pin, active low
VCC	Target voltage sense, should be connected to the device V_{DD}
GND	Ground
GND*	These pins are reserved for firmware extension purposes. They can be left unconnected or connected to GND in normal debug environment. They are not essential for SWD in general.

48.8. USB Interface

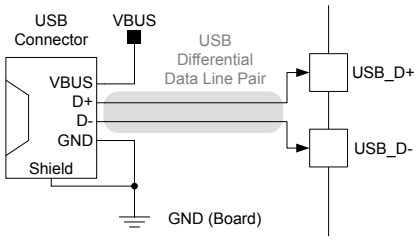
The USB interface consists of a differential data pair (D+/D-) and a power supply (VBUS, GND).

Refer to the Electrical Characteristics section for operating voltages which will allow USB operation.

Table 48-10. USB Interface Checklist

Signal Name	Recommended Pin Connection	Description
D+	• The impedance of the pair should be matched on the PCB to minimize reflections. • USB differential tracks should be routed with the same characteristics (length, width, number of vias, etc.) • Signals should be routed as parallel as possible, with a minimum number of angles and vias	USB full speed / low speed positive data upstream pin
D-		USB full speed / low speed negative data upstream pin

Figure 48-15. Low Cost USB Interface Example Schematic

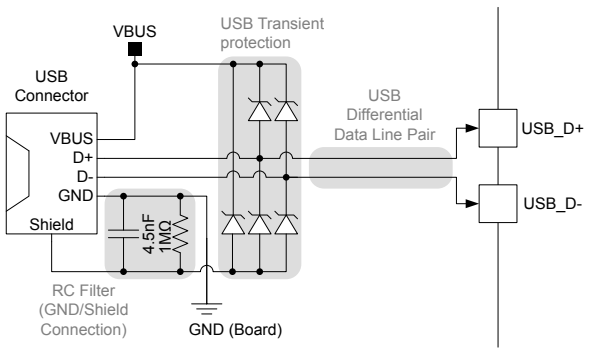


It is recommended to increase ESD protection on the USB D+, D-, and VBUS lines using dedicated transient suppressors. These protections should be located as close as possible to the USB connector to reduce the potential discharge path and reduce discharge propagation within the entire system.

The USB FS cable includes a dedicated shield wire that should be connected to the board with caution. Special attention should be paid to the connection between the board ground plane and the shield from the USB connector and the cable.

Tying the shield directly to ground would create a direct path from the ground plane to the shield, turning the USB cable into an antenna. To limit the USB cable antenna effect, it is recommended to connect the shield and ground through an RC filter.

Figure 48-16. Protected USB Interface Example Schematic



Related Links

Electrical Characteristics on page 1140

49. Errata

49.1. Revision A

49.1.1. PM

1 – If a RTC wakeup source is active (alarm, compare, periodic or overflow), the device cannot enter battery backup mode. Errata reference: 13947
Fix/Workaround:
None

2 – In standby mode, when running peripherals(like TC4 for example) from GCLK clock with 32Khz source, the main voltage regulator is used instead of the low power regulator, causing higher power consumption. Errata reference: 13901
Fix/Workaround:
Before entering standby mode, this code can be used: SUPC->VREF.reg |= (1 << 8).
After leaving standby mode, use this code: SUPC->VREF.reg &= ~(1 << 8).
This workaround must be used only in performance level 0 (PL0).

49.1.2. PORT

1 – When the PORT is defined as EVSYS.USER in a synch/resynch path, the first event is transmitted to the PORT but the acknowledgement coming from the PORT is not released. So next coming events are treated as overrun by EVSYS. Errata reference: 14317
Fix/Workaround:
None.
Do not use the synch/resynch path, only use asynchronous path.

49.1.3. SUPC

1 – The SUPC.LPEFF bit has no effect. Errata reference: 13463
Fix/Workaround:
None

2 – When VDDCORE is supplied by the BUCK converter in performance level 0, the chip cannot wake-up from standby mode because the VCORERDY status is stuck at 0. Errata reference: 13551
Fix/Workaround:
Before entering standby mode, switch the voltage regulator from the buck converter to the LDO

3 – The VREG.STDBYPL0 bit has no effect on standby power consumption when the buck regulator is used in standby mode (VREG.RUNSTDBY=1). Errata reference: 14066
Fix/Workaround:
None

4 – In PL0, Wake-up time from standby mode is two times longer than expected. Errata reference: 13674
Fix/Workaround:
Set bit 8 of SUPC.VREF to one before entering in standby mode and set it to zero after wake-up

49.1.4. TC

1 – The input capture on IO pins does not work. Errata reference: 14024
Fix/Workaround:
Use the input capture through TC event and use the EIC or CCL as event generators.

2 – In standby mode, when PD1 is in retention state and PD0 is in active state, if TC4 triggers a DMA request for sleepwalking purpose, the DMA transfer does not start. Errata reference: 13605
Fix/Workaround:
Before going to standby, set the PM->STDBYCFG.PDCFG to PD01 to force the power domain PD1 to active state.

49.1.5. RTC

1 – The COUNTSYNC/CLOCKSYNC bit of the RTC.CTRLA register has no effect. Read synchronization of the COUNT/CLOCK register is always enabled. Errata reference: 13714
Fix/Workaround:
None

49.1.6. DAC

1 – If the DAC is enabled, STATUS.READY is not cleared during standby and will remain one after waking up, even though the DAC needs to reinitialize. Errata reference: 14678
Fix/Workaround:
Wait for INTFLAG.EMPTY to be one after enabling the DAC instead of waiting for STATUS.READY.

2 – If CLK_APB_DAC is slower than GCLK_DAC the STATUS.READY bit may never be set. Errata reference: 14664
Fix/Workaround:
Wait for INTFLAG.EMPTY to be one after enabling the DAC instead of waiting for STATUS.READY.

3 – If refresh is enabled, the refresh cycle will not be started if data conversion happens too soon after DAC is enabled. Errata reference: 13844
Fix/Workaround:
Delay the first data conversion for longer than 60us, or issue a second conversion longer than 60us.

4 – For specific DAC configurations, the SYNCBUSY.DATA1 and SYNCBUSY.DATABUF1 may be stuck at 1. Errata reference: 14910
Fix/Workaround:
Don’t use the Refresh mode and Events at the same time for DAC1. If event is used, write data to DATABUF1 with no refresh. DAC0 is not limited by this restriction.

5 – The DAC cannot refresh automatically with accuracy. Errata reference: 13909
Fix/Workaround:
Manually issue data conversion every 100us to refresh and maintain accuracy.

6 – The reset value of the CTRLB register is 0x00. This value increases the loading capacitance of pin PA03 by about 5pF. This extra load can be an issue when this pin is used as a PTC Y line in self-cap mode. Errata reference: 14310
Fix/Workaround:
Write 0x02 in the CTRLB register.

49.1.7. PAC

1 – Any writing to PAC.INTFLAGAHB register clear all the flags in the register Errata reference: 13678
Fix/Workaround :
None

49.1.8. TRNG

1 – When the TRNG is enabled with CTRL.RUNSTDBY set to 0, an over-consumption (~50µA) could be present in standby mode. Errata reference: 14827
Fix/Workaround:
Disable the TRNG before entering standby mode.

49.1.9. RSTC

1 – When a System Reset Request is applied, the OSC16MCTRL register is not reset. Errata reference: 13416
Fix/Workaround:
None.

49.1.10. Device

1 – The H2LBRIDGE and H2LBRIDGE latency is high. Errata reference: 13414
For a write the latency is 5-cycles (cycles needed for a transaction to propagate from the bridge slave endpoint to the master endpoint). As the bridge supports posted-write, no stall cycles is seen as long as the FIFO is not full.
For a read the latency is 4-cycles for the address phase and again 4-cycles for the data to propagate back leading to a total of 8-cycle stall when there is no LP clock division and the accessed slave does not stall.
Fix/Workaround:
None

2 – In Standby mode, when Power Domain 1 is power gated, devices can show higher consumption than expected. Errata reference: 13599
Fix/Workaround:
Force the Power Domain 1 to remain active by setting PM.PDCFG to 0x2

3 – In IDLE sleep mode, the APB and AHB clocks are not stopped if the FDPLL is running as a GCLK clock source. Errata reference: 13401
Fix/Workaround:
Disable the FDPLL before entering IDLE sleep mode.

4 – The low latency mode cannot be enabled by writing a one to address 0x41008120. This bit has no effect. Errata reference: 13506
Fix/Workaround:
None

5 – XOSC32K contains load capacitor equivalent to Cload=7pf. External load capacitors must take this into account. Errata reference: 13425
Fix/Workaround:
None

6 – The SUPC/OUT pins toggle on the RTC Periodic Interval 0 (PER0) event only. Errata reference: 14151
Fix/Workaround:
None

7 – In IDLE sleep mode, the APB and AHB clocks are not stopped if the DFLL is running as a GCLK clock source. Errata reference: 13384
Fix/Workaround:
Disable the DFLL before entering IDLE sleep mode.

8 – The default TC selection as CCL input is not TC0 but TC4. Thus the TC selection is TC4/TC0/TC1/TC2 instead of TC0/TC1/TC2/TC3. And the TC alternate selection is TC0/TC1/TC2/TC3 instead of TC1/TC2/TC3/TC4. Errata reference: 13406
Fix/Workaround:
Use the TC input mapping described above.

9 – Pulldown functionality is not available on GPIO pin PA24 and PA25 Errata reference: 13915
Fix/Workaround:
None

10 – When BOD12 is configured to run in continuous mode (BOD12.STDBYCFG=0) in standby sleep mode, and configured to run in sampling mode (BOD12.ACTCFG=1) in active mode, the BOD12 detection does not work in standby sleep mode. Errata reference: 14150
Fix/Workaround:
Any other combination of BOD12.STDBYCFG and BOD12.ACTCFG works as expected.

11 – When internal voltage reference (SUPC.VREF.SEL) is selected as an input for the ADC and when SUPC.VREF.SEL > 0x4, then this reference is statically enabled and SUPC.VREF.ONDEMAND has no effect. Errata reference: 14588
Fix/Workaround :
None

12 – In I2C Slave mode, writing the CTRLB register when in the AMATCH or DRDY interrupt service routines can cause the state machine to reset. Errata reference: 13574
Fix/Workaround:

Write CTRLB.ACKACT to 0 using the following sequence:
// If higher priority interrupts exist, then disable so that the
// following two writes are atomic.
SERCOM - STATUS.reg = 0;
SERCOM - CTRLB.reg = 0;
// Re-enable interrupts if applicable.
Write CTRLB.ACKACT to 1 using the following sequence:
// If higher priority interrupts exist, then disable so that the
// following two writes are atomic.
SERCOM - STATUS.reg = 0;
SERCOM - CTRLB.reg = SERCOM_I2CS_CTRLB_ACKACT;
// Re-enable interrupts if applicable.
Otherwise, only write to CTRLB in the AMATCH or DRDY interrupts if it is to close out a transaction.
When not closing a transaction, clear the AMATCH interrupt by writing a 1 to its bit position instead of using CTRLB.CMD. The DRDY interrupt is automatically cleared by reading/writing to the DATA register in smart mode. If not in smart mode, DRDY should be cleared by writing a 1 to its bit position.
Code replacements examples:
Current:
SERCOM - CTRLB.reg |= SERCOM_I2CS_CTRLB_ACKACT;
Change to:
// If higher priority interrupts exist, then disable so that the
// following two writes are atomic.
SERCOM - STATUS.reg = 0;
SERCOM - CTRLB.reg = SERCOM_I2CS_CTRLB_ACKACT;
// Re-enable interrupts if applicable.
Current:
SERCOM - CTRLB.reg &= ~SERCOM_I2CS_CTRLB_ACKACT;
Change to:
// If higher priority interrupts exist, then disable so that the
// following two writes are atomic.
SERCOM - STATUS.reg = 0;
SERCOM - CTRLB.reg = 0;
// Re-enable interrupts if applicable.
Current:
/* ACK or NACK address */
SERCOM - CTRLB.reg |= SERCOM_I2CS_CTRLB_CMD(0x3);
Change to:
// CMD=0x3 clears all interrupts, so to keep the result similar,
// PREC is cleared if it was set.
if (SERCOM - INTFLAG.bit.PREC) SERCOM - INTFLAG.reg = SERCOM_I2CS_INTFLAG_PREC;
SERCOM - INTFLAG.reg = SERCOM_I2CS_INTFLAG_AMATCH;

13 – PTC Y-lines number 2,3,5 and 14 are not mapped on PB10, PB11, PB12 and PB13. Errata reference: 14046
OA_OUT[1] is not mapped on PB08
OA_NEG[2] is not mapped on PB06
Fix/Workaround :

PTC Y-lines number 2,3,5 and 14 are mapped on PA04, PA05, PA07 and PB08.
OA_OUT[1] is mapped on PB06
OA_NEG[2] is mapped on PB08

14 – When entering standby mode, the DFLL is still running even if not requested by any module causing extra consumption as the power domain PD0 is not turned off. Errata reference: 13984
Fix/Workaround:
DFLL must be disabled before entering in standby mode and re-enabled after wake-up.

15 – The SYSTICK calibration value is incorrect. Errata reference: 13995
Fix/Workaround:
The correct SYSTICK calibration value is 0x40000000. This value should not be used to initialize the Systick RELOAD value register, which should be initialized instead with a value depending on the main clock frequency and on the tick period required by the application. For a detailed description of the SYSTICK module, refer to the official ARM Cortex-M0+ documentation.

16 – When VDDIN < 2.7V, the consumption in STANDBY mode can be up to 3 times higher than expected. Errata reference: 13388
Fix/Workaround:
Use the device with VDDIN > 2.7V

17 – If the battery backup power switch is not set to wake-up from the battery backup mode (BBPS.WAKEEN=0), the SAM L21 wakes up when the main power is restored and BKUPEXIT.BBPS and RCAUSE.EXT is set to one(EXT reset seen as wakeup source) Errata reference: 13905
Fix/Workaround:
None

49.1.11. BOD33

1 – Switching the BOD33 from disable to enable triggers a BOD33 reset. Errata reference: 13918
Fix/Workaround:
If reset action is required, setup the BOD at power on using the user row.
For interrupt action,
- mask the interrupt (SUPC->INTENCLR.bit.BOD33DET=1)
- enable the BOD
- clear the flag (SUPC.INTFLAG.bit.BOD33DET)
- unmask the interrupt (SUPC->INTENSET.bit.BOD33DET=1)

49.1.12. BUCK

1 – When VCC < 2.7V, the main VREG in buck mode is not functional. Errata reference: 13657
Fix/Workaround:
When using the main VREG in buck mode, VCC must be > 2.7V.

49.1.13. ADC

1 – Overconsumption for up to 1.6 seconds on VDDANA when the ADC is disabled(manually or automatically) Errata reference: 14349

Fix/Workaround :
None

2 – The ADC Effective number of Bits (ENOB) is 9.2 in this revision Errata reference: 13850

49.1.14. DFLL48M

1 – The DFLL clock must be requested before being configured otherwise a write access to a DFLL register can freeze the device. Errata reference: 9905
Fix/Workaround:
Write a zero to the DFLL ONDEMAND bit in the DFLLCTRL register before configuring the DFLL module.

2 – If the DFLL48M reaches the maximum or minimum COARSE or FINE calibration values during the locking sequence, an out of bounds interrupt will be generated. These interrupts will be generated even if the final calibration values at DFLL48M lock are not at maximum or minimum, and might therefore be false out of bounds interrupts. Errata reference: 10669
Fix/Workaround:
Check that the lockbits: DFLLLCKC and DFLLLCKF in the SYSCTRL Interrupt Flag Status and Clear register (INTFLAG) are both set before enabling the DFLLOOB interrupt.

3 – The DFLL status bits in the PCLKSR register during the USB clock recovery mode can be wrong after a USB suspend state. Errata reference: 11938
Fix/Workaround:
Do not monitor the DFLL status bits in the PCLKSR register during the USB clock recovery mode.

49.1.15. FDPLL

1 – When entering standby mode, the FDPLL is still running even if not requested by any module causing extra consumption. Errata reference: 12244
Fix/Workaround:
FDPLL must be disabled before entering in standby mode and re-enabled after wake-up

49.1.16. SERCOM

1 – In USART autobaud mode, missing stop bits are not recognized as inconsistent sync (ISF) or framing (FERR) errors. Errata reference: 13852
Fix/Workaround:
None

2 – If the SERCOM is enabled in SPI mode with SSL detection enabled (CTRLB.SSDE) and CTRLB.RXEN=1, an erroneous slave select low interrupt (INTFLAG.SSL) can be generated. Errata reference: 13369
Fix/Workaround:
Enable the SERCOM first with CTRLB.RXEN=0. In a subsequent write, set CTRLB.RXEN=1.

49.1.17. NVMCTRL

1 – Default value of MANW in NVM.CTRLB is 0. Errata reference: 13134
This can lead to spurious writes to the NVM if a data write is done through a pointer with a wrong address corresponding to NVM area.
Fix/Workaround:
Set MANW in the NVM.CTRLB to 1 at startup

2 – When external reset is active it causes a high leakage current on VDDIO. Errata reference: 13446
Fix/Workaround:
Minimize the time external reset is active.

49.1.18. DSU

1 – The MBIST ""Pause-on-Error"" feature is not functional on this device. Errata reference: 14324
Fix/Workaround:
Do not use the ""Pause-on-Error"" feature.

49.1.19. TCC

1 – FCTRLX.CAPTURE[CAPTMARK] does not work as described in the datasheet. CAPTMARK cannot be used to identify captured values triggered by fault inputs source A or B on the same channel. Errata reference: 13316
Fix/Workaround:
Use two different channels to timestamp FaultA and FaultB.

2 – When the circular buffer is enabled, an APB clock is requested to update the corresponding APB register. If all masters in the system (CPU, DMA) are disabled, the APB clock is never provided to the TCC, making the circular buffer feature not functional in standby sleep mode. Errata reference: 12269
Fix/Workaround:
Keep a master enabled in the system (enable DMA, or do not enable standby sleep mode when circular buffer is enabled).

3 – In RAMP 2 mode with Fault keep, qualified and restart: Errata reference: 13262
If a fault occurred at the end of the period during the qualified state, the switch to the next ramp can have two restarts.
Fix/Workaround:
Avoid faults few cycles before the end or the beginning of a ramp.

4 – All DMA MCx triggers are not raised, on a CCx match. Errata reference: 13084
Fix/Workaround:
To update CC/CCBUF values through DMA, use DMA OVF triggers

49.1.20. DMAC

1 – If data is written to CRCDATAIN in two consecutive instructions, the CRC computation may be incorrect. Errata reference: 13507
Fix/Workaround:
Add a NOP instruction between each write to CRCDATAIN register.

49.1.21. CCL

1 – The reset of the RS latch is not functional. The latch can only be cleared by disabling the LUT. Errata reference: 14043
Fix/Workaround:
None

49.1.22. ADC

1 – Once set, the ADC.SWTRIG.START will not be cleared until the Microcontroller is reset. Errata reference: 14094
Fix/Workaround:
None

2 – When using the automatic sequencing feature, if the TEMP (temperature sensor) input and the BANDGAP input are selected, the BANDGAP will be converted twice: the first conversion will be stored in place of the TEMP conversion result and the second conversion will be stored in the BANDGAP conversion result. Errata reference: 14217
Fix/Workaround:
To be able to read the temperature sensor with the automatic sequencing feature, the bandgap input must be left out of the sequence.

3 – When window monitor is enabled and its output is 0, the ADC GCLK is kept running. Power consumption will be higher than expected in sleep modes Errata reference: 14449
Fix/Workaround:
None

4 – The LSB bit of ADC result is stuck at zero, in unipolar mode for 8-bit and 10-bit resolution. Errata reference: 14431
Fix/Workaround:
None

5 – ADC does not work in STANDBY mode when regulator is in Buck regulation mode with SUPC.VREG.RUNSTBDBY=1 and fast wakeup is enabled (SUPC.VREF.bit8=1). Errata reference: 14229
Fix/Workaround:
Set the device in IDLE or RUN mode to perform ADC conversions.

49.1.23. EIC

1 – Asynchronous edge detection is not supported in SAML21 revA. In other words, the EIC.NMICTRL.ASYNCH bit and the bits in the EIC.ASYNCH register have no effect. Errata reference: 12949
Fix/Workaround:
None

49.1.24. EVSYS

1 – The acknowledge between an event user and the EVSYS clears the CHSTATUS.CHBUSYn bit before this information is fully propagated in the EVSYS one GCLK_EVSYS_CHANNEL_n clock cycle later. As a consequence, any generator event occurring on that channel before that extra GCLK_EVSYS_CHANNEL_n clock cycle will trigger the overrun flag. Errata reference: 14835

Fix/Workaround:
For applications using event generators other than the software event, monitor the OVR flag.
For applications using the software event generator, wait one GCLK_EVSYS_CHANNEL_n clock cycle after the CHSTATUS.CHBUSYn bit is cleared before issuing a software event.

2 – Using synchronous, spurious overrun can appears with generic clock for the channel always on. Errata reference: 14532
Fix/Workaround:
- Request the generic clock on demand by setting the CHANNEL.ONDEMAND bit to one.
- No penalty is introduced.

49.2. Revision B

49.2.1. PM

1 – If the PM.STDBYCFG.VREGSMOD field is set to 2 (Low Power configuration), the oscillator source driving the GCLK_MAIN clock will still be running in standby mode causing extra consumption. Errata reference: 14539
Fix/Workaround:
Before entering in standby mode, switch the GCLK_MAIN to the OSCULP32K clock. After wakeup, switch back to the GCLK_MAIN clock.

49.2.2. DAC

1 – If the DAC is enabled, STATUS.READY is not cleared during standby and will remain one after waking up, even though the DAC needs to reinitialize. Errata reference: 14678
Fix/Workaround:
Wait for INTFLAG.EMPTY to be one after enabling the DAC instead of waiting for STATUS.READY.

2 – If CLK_APB_DAC is slower than GCLK_DAC the STATUS.READY bit may never be set. Errata reference: 14664
Fix/Workaround:
Wait for INTFLAG.EMPTY to be one after enabling the DAC instead of waiting for STATUS.READY.

3 – For specific DAC configurations, the SYNCBUSY.DATA1 and SYNCBUSY.DATABUF1 may be stuck at 1. Errata reference: 14910
Fix/Workaround:
Don’t use the Refresh mode and Events at the same time for DAC1. If event is used, write data to DATABUF1 with no refresh. DAC0 is not limited by this restriction.

4 – The SYNCBUSY.ENABLE bit is stuck at 1 after disabling and enabling the DAC when refresh is used. Errata reference: 14885
Fix/Workaround:
After the DAC is disabled in refresh mode, wait for at least 30us before re-enabling the DAC.

49.2.3. TRNG

1 – When the TRNG is enabled with CTRL.RUNSTDBY set to 0, an over-consumption (~50µA) could be present in standby mode. Errata reference: 14827
Fix/Workaround:
Disable the TRNG before entering standby mode.

49.2.4. ADC

1 – Overconsumption for up to 1.6 seconds on VDDANA when the ADC is disabled(manually or automatically) Errata reference: 14349
Fix/Workaround :
None

2 – The ADC Effective number of Bits (ENOB) is 9.2 in this revision Errata reference: 13850

49.2.5. DFLL48M

1 – The DFLL clock must be requested before being configured otherwise a write access to a DFLL register can freeze the device. Errata reference: 9905
Fix/Workaround:
Write a zero to the DFLL ONDEMAND bit in the DFLLCTRL register before configuring the DFLL module.

2 – If the DFLL48M reaches the maximum or minimum COARSE or FINE calibration values during the locking sequence, an out of bounds interrupt will be generated. These interrupts will be generated even if the final calibration values at DFLL48M lock are not at maximum or minimum, and might therefore be false out of bounds interrupts. Errata reference: 10669
Fix/Workaround:
Check that the lockbits: DFLLLCKC and DFLLLCKF in the SYSCTRL Interrupt Flag Status and Clear register (INTFLAG) are both set before enabling the DFLLOOB interrupt.

3 – The DFLL status bits in the PCLKSR register during the USB clock recovery mode can be wrong after a USB suspend state. Errata reference: 11938
Fix/Workaround:
Do not monitor the DFLL status bits in the PCLKSR register during the USB clock recovery mode.

49.2.6. DMAC

1 – A write from DMAC to a register in a module to disable a trigger from the module to DMAC, does not work in standby mode. Errata reference: 14648
(For example DAC, SLCD, SERCOM in transmission mode)
Fix/Workaround:
If the module generating the trigger also generates event, use event interface instead of triggers with DMAC (for example SLCD)

49.2.7. ADC

1 – When window monitor is enabled and its output is 0, the ADC GCLK is kept running. Power consumption will be higher than expected in sleep modes Errata reference: 14449
Fix/Workaround:
None

2 – The LSB bit of ADC result is stuck at zero, in unipolar mode for 8-bit and 10-bit resolution. Errata reference: 14431
Fix/Workaround:
None

49.2.8. EIC

1 – access to EIC_ASYNCH register in 8 or 16 bits mode is not functional. Errata reference: 14417
Writing the lowest 8 bits will also write the higher 8 bits.
Writing the highest 8 bits will also write the lower 8 bits.
Writing the lowest 16 bits will also write the higher 16 bits.
Writing the highest 16 bits will also write the lower 16 bits.
Fix/Workaround:
use only the 16 bit low part or 32 bits write modes

49.2.9. EVSYS

1 – The acknowledge between an event user and the EVSYS clears the CHSTATUS.CHBUSYn bit before this information is fully propagated in the EVSYS one GCLK_EVSYS_CHANNEL_n clock cycle later. As a consequence, any generator event occurring on that channel before that extra GCLK_EVSYS_CHANNEL_n clock cycle will trigger the overrun flag. Errata reference: 14835
Fix/Workaround:
For applications using event generators other than the software event, monitor the OVR flag.
For applications using the software event generator, wait one GCLK_EVSYS_CHANNEL_n clock cycle after the CHSTATUS.CHBUSYn bit is cleared before issuing a software event.

2 – Using synchronous, spurious overrun can appears with generic clock for the channel always on. Errata reference: 14532
Fix/Workaround:
- Request the generic clock on demand by setting the CHANNEL.ONDEMAND bit to one.
- No penalty is introduced.

49.3. Revision C

49.3.1. DAC

1 – For specific DAC configurations, the SYNCBUSY.DATA1 and SYNCBUSY.DATABUF1 may be stuck at 1. Errata reference: 14910
Fix/Workaround:

Don’t use the Refresh mode and Events at the same time for DAC1. If event is used, write data to DATABUF1 with no refresh. DAC0 is not limited by this restriction.

2 – The SYNCBUSY.ENABLE bit is stuck at 1 after disabling and enabling the DAC when refresh is used. Errata reference: 14885
Fix/Workaround:
After the DAC is disabled in refresh mode, wait for at least 30us before re-enabling the DAC.

49.3.2. TRNG

1 – When the TRNG is enabled with CTRL.RUNSTDBY set to 0, an over-consumption (~50µA) could be present in standby mode. Errata reference: 14827
Fix/Workaround:
Disable the TRNG before entering standby mode.

49.3.3. DFLL48M

1 – The DFLL clock must be requested before being configured otherwise a write access to a DFLL register can freeze the device. Errata reference: 9905
Fix/Workaround:
Write a zero to the DFLL ONDEMAND bit in the DFLLCTRL register before configuring the DFLL module.

2 – If the DFLL48M reaches the maximum or minimum COARSE or FINE calibration values during the locking sequence, an out of bounds interrupt will be generated. These interrupts will be generated even if the final calibration values at DFLL48M lock are not at maximum or minimum, and might therefore be false out of bounds interrupts. Errata reference: 10669
Fix/Workaround:
Check that the lockbits: DFLLLCKC and DFLLLCKF in the SYSCTRL Interrupt Flag Status and Clear register (INTFLAG) are both set before enabling the DFLLOOB interrupt.

3 – The DFLL status bits in the PCLKSR register during the USB clock recovery mode can be wrong after a USB suspend state. Errata reference: 11938
Fix/Workaround:
Do not monitor the DFLL status bits in the PCLKSR register during the USB clock recovery mode.

49.3.4. ADC

1 – The LSB bit of ADC result is stuck at zero, in unipolar mode for 8-bit and 10-bit resolution. Errata reference: 14431
Fix/Workaround:
None

49.3.5. EIC

1 – access to EIC_ASYNCH register in 8 or 16 bits mode is not functional. Errata reference: 14417

Writing the lowest 8 bits will also write the higher 8 bits.
Writing the highest 8 bits will also write the lower 8 bits.
Writing the lowest 16 bits will also write the higher 16 bits.
Writing the highest 16 bits will also write the lower 16 bits.
Fix/Workaround:
use only the 16 bit low part or 32 bits write modes

49.3.6. EVSYS

1 – The acknowledge between an event user and the EVSYS clears the CHSTATUS.CHBUSYn bit before this information is fully propagated in the EVSYS one GCLK_EVSYS_CHANNEL_n clock cycle later. As a consequence, any generator event occurring on that channel before that extra GCLK_EVSYS_CHANNEL_n clock cycle will trigger the overrun flag. Errata reference: 14835
Fix/Workaround:
For applications using event generators other than the software event, monitor the OVR flag.
For applications using the software event generator, wait one GCLK_EVSYS_CHANNEL_n clock cycle after the CHSTATUS.CHBUSYn bit is cleared before issuing a software event.

2 – Using synchronous, spurious overrun can appears with generic clock for the channel always on. Errata reference: 14532
Fix/Workaround:
- Request the generic clock on demand by setting the CHANNEL.ONDEMAND bit to one.
- No penalty is introduced.

50. Conventions

50.1. Numerical Notation

Table 50-1. Numerical Notation

Symbol	Description
165	Decimal number
0b0101	Binary number (example 0b0101 = 5 decimal)
'0101'	Binary numbers are given without prefix if unambiguous.
0x3B24	Hexadecimal number
X	Represents an unknown or don't care value
Z	Represents a high-impedance (floating) state for either a signal or a bus

50.2. Memory Size and Type

Table 50-2. Memory Size and Bit Rate

Symbol	Description
KB (kbyte)	kilobyte (2^{10} = 1024)
MB (Mbyte)	megabyte (2^{20} = 1024*1024)
GB (Gbyte)	gigabyte (2^{30} = 1024*1024*1024)
b	bit (binary '0' or '1')
B	byte (8 bits)
1kbit/s	1,000 bit/s rate (not 1,024 bit/s)
1Mbit/s	1,000,000 bit/s rate
1Gbit/s	1,000,000,000 bit/s rate
word	32 bit
half-word	16 bit

50.3. Frequency and Time

Symbol	Description
kHz	1kHz = 10^3Hz = 1,000Hz
KHz	1KHz = 1,024Hz, 32KHz = 32,768Hz
MHz	10^6 = 1,000,000Hz

Symbol	Description
GHz	10^9 = 1,000,000,000Hz
s	second
ms	millisecond
µs	microsecond
ns	nanosecond

50.4. Registers and Bits

Table 50-3. Register and Bit Mnemonics

Symbol	Description
R/W	Read/Write accessible register bit. The user can read from and write to this bit.
R	Read-only accessible register bit. The user can only read this bit. Writes will be ignored.
W	Write-only accessible register bit. The user can only write this bit. Reading this bit will return an undefined value.
BIT	Bit names are shown in uppercase. (Example ENABLE)
FIELD[n:m]	A set of bits from bit n down to m. (Example: PINA[3:0] = {PINA3, PINA2, PINA1, PINA0}
Reserved	Reserved bits are unused and reserved for future use. For compatibility with future devices, always write reserved bits to zero when the register is written. Reserved bits will always return zero when read.
PERIPHERAL*i*	If several instances of a peripheral exist, the peripheral name is followed by a number to indicate the number of the instance in the range 0-n. PERIPHERAL0 denotes one specific instance.
Reset	Value of a register after a power reset. This is also the value of registers in a peripheral after performing a software reset of the peripheral, except for the Debug Control registers.
SET/CLR	Registers with SET/CLR suffix allows the user to clear and set bits in a register without doing a read-modify-write operation. These registers always come in pairs. Writing a one to a bit in the CLR register will clear the corresponding bit in both registers, while writing a one to a bit in the SET register will set the corresponding bit in both registers. Both registers will return the same value when read. If both registers are written simultaneously, the write to the CLR register will take precedence.

51. Acronyms and Abbreviations

The below table contains acronyms and abbreviations used in this document.

Table 51-1. Acronyms and Abbreviations

Abbreviation	Description
AC	Analog Comparator
ADC	Analog-to-Digital Converter
ADDR	Address
AES	Advanced Encryption Standard
AHB	AMBA Advanced High-performance Bus
AMBA®	Advanced Microcontroller Bus Architecture
APB	AMBA Advanced Peripheral Bus
AREF	Analog reference voltage
BLB	Boot Lock Bit
BOD	Brown-out detector
CAL	Calibration
CC	Compare/Capture
CCL	Configurable Custom Logic
CLK	Clock
CRC	Cyclic Redundancy Check
CTRL	Control
DAC	Digital-to-Analog Converter
DAP	Debug Access Port
DFLL	Digital Frequency Locked Loop
DMAC	DMA (Direct Memory Access) Controller
DSU	Device Service Unit
EEPROM	Electrically Erasable Programmable Read-Only Memory
EIC	External Interrupt Controller
EVSYS	Event System
GCLK	Generic Clock Controller
GND	Ground
GPIO	General Purpose Input/Output
I^2C	Inter-Integrated Circuit
IF	Interrupt flag

Abbreviation	Description
INT	Interrupt
MBIST	Memory built-in self-test
MEM-AP	Memory Access Port
MTB	Micro Trace Buffer
NMI	Non-maskable interrupt
NVIC	Nested Vector Interrupt Controller
NVM	Non-Volatile Memory
NVMCTRL	Non-Volatile Memory Controller
OPAMP	Operation Amplifier
OSC	Oscillator
PAC	Peripheral Access Controller
PC	Program Counter
PER	Period
PM	Power Manager
POR	Power-on reset
PORT	I/O Pin Controller
PTC	Peripheral Touch Controller
PWM	Pulse Width Modulation
RAM	Random-Access Memory
REF	Reference
RTC	Real-Time Counter
RX	Receiver/Receive
SERCOM	Serial Communication Interface
SMBus™	System Management Bus
SP	Stack Pointer
SPI	Serial Peripheral Interface
SRAM	Static Random-Access Memory
SUPC	Supply Controller
SWD	Serial Wire Debug
TC	Timer/Counter
TCC	Timer/Counter for Control Applications
TRNG	True Random Number Generator
TX	Transmitter/Transmit

Abbreviation	Description
ULP	Ultra-low power
USART	Universal Synchronous and Asynchronous Serial Receiver and Transmitter
USB	Universal Serial Bus
V_{DD}	Common voltage to be applied to VDDIO, VDDIN and VDDANA
V_{DDIN}	Digital supply voltage
V_{DDIO}	Digital supply voltage
V_{DDANA}	Analog supply voltage
VREF	Voltage reference
WDT	Watchdog Timer
XOSC	Crystal Oscillator

52. Datasheet Revision History

52.1. Rev H - 12/2015

Configuration Summary on page 15 Ordering Information on page 18 Pinout on page 23 Packaging Information on page 1180	Package option WLCSP64 added
DSU - Device Service Unit on page 87	Bit CTRL.CRC is write-only.
GCLK - Generic Clock Controller on page 131	Bit field description for GENCTRL.SRC updated.
DMAC – Direct Memory Access Controller on page 399	• Bit field description CTRL.CRCENABLE added. • Bit field descriptions for PRICTRL0.LVLPRIn updated.
TCC – Timer/Counter for Control Applications on page 755	Editorial updates for CCBUFx registers.
Electrical Characteristics on page 1140	Timing Characteristics for SERCOM added.
Packaging Information on page 1180	Drawings updated for QFN32 and TQFP64.
Schematic Checklist on page 1189	Editorial updates for 'External Real Time Oscillator.'

52.2. Rev G - 11/2015

Processor and Architecture on page 49	• Accessing HMATRIX configuration must happen in 32-bit operations. • Editorial updates.
Clock System on page 125	• Editorial updates.
SUPC – Supply Controller on page 283	• Editorial updates.
OSCCTRL – Oscillators Controller on page 217	• STATUS.DFLLRDY indicates readiness of the DFLL48M registers for read/write access. • Editorial updates.
RTC – Real-Time Counter on page 341	• FREQCORR.SIGN affects period length
DAC – Digital-to-Analog Converter on page 1102	• Editorial updates.
CCL – Configurable Custom Logic on page 989	• Editorial updates.
Thermal Resistance Data on page 1180	• Values for QFN48 and QFN64 updated.

Electrical Characteristics on page 1140	• Entire section: Notes on values' base • Supply Characteristics – Table Supply Slew Rates updated – Table Supply Current Requirements removed • Maximum Clock Frequencies – DFLL, FDPLL reference clock frequencies updated • Power Consumption – Max. values added – Values updated – Editorial update • Wake-Up Time: editorial updates • Peripheral Power Consumption: Section removed. Data moved to tables 'Power Consumption' in respective subsections • I/O Pin Characteristics: Table I/O Pin Dynamic Characteristics updated

Electrical Characteristics on page 1140	• Voltage Regulator Characteristics – Table External Components Requirements in Switching Mode updated – Table External Components Requirements in Linear Mode updated – Table POR33 Characteristics updated – Table BOD33 Characteristics updated – Table BOD33 Power Consumption added • Analog-to-Digital Converter (ADC) Characteristics – Table Operating Conditions updated – Table Power Consumption added – Values updated, Min. and Max. added – Figure ADC Analog Input AINx added • Digital-to-Analog Converter (DAC) Characteristics – Table Operating Conditions updated – Values updated, Min. and Max. added – Table Power Consumption added • Analog Comparator (AC) Characteristics – Table Electrical and Timing updated – Values updated, Min. and Max. added – Table Power Consumption added • DETREF added • Temperature Sensor Characteristics: ADCTsAcc removed • OPAMP: Power Consumption added • NVM Characteristics: DSU Chip erase removed • External Reset Pin: section added • USB Characteristics: editorial update

Electrical Characteristics on page 1140 (Oscillator sections)	• Crystal Oscillator (XOSC) Characteristics – Digital Clock Characteristics added – Values updated and added – Power Consumption added • External 32KHz Crystal Oscillator (XOSC32K) Characteristics – Digital Clock Characteristics added – Conditions updated, values added – Power Consumption added • 32.768kHz Internal Oscillator (OSC32K) Characteristics – Values updated and added – Power Consumption added • Internal Ultra Low Power 32KHz RC Oscillator (OSCULP32K) Characteristics – Title changed – Values updated and added – $T_{STARTUP}$ removed – Power Consumption added • Digital Frequency Locked Loop (DFLL48M) Characteristics – Values added – Power Consumption added • 16MHz RC Oscillator (OSC16M) Characteristics – Values and conditions added – Power Consumption added • Digital Phase Lock Loop (DPLL) Characteristics – Values for jitter, lock time updated and added – Power Consumption added
Errata on page 1202	Added Errata: • **EVSYS**: CHSTATUS.CHBUSYn may be cleared too early under certain conditions. *Errata reference 14835* • **DAC**: Unexpected value of SYNCBUSY.ENABLE under certain conditions. *Errata reference 14885* • **DAC**: Unexpected value of SYNCBUSY.DATA1 and SYNCBUSY.DATABUF1 under certain conditions. *Errata reference 14910*

52.3. Rev F - 09/2015

Features on page 1	• Number of PTC channels change from 192 to 169.
Configuration Summary on page 15	• Added footnotes explaining which instances of TCs are available for the SAM L21E and G. • Number of PTC channels updated. Presentation split in mutual- and self-capacitance.
I/O Multiplexing and Considerations on page 29	• Notes added. • GPIO Cluster table added. • SERCOM Configurations added. • Oscillator Pinout: Recommendation for XOSC32 jitter optimization added.
DSU - Device Service Unit on page 87	• $\overline{\text{RESET}}$ is active low.
PM – Power Manager on page 186	• Register SLEEPCFG has reset value 0x2
OSCCTRL – Oscillators Controller on page 217	• Register OSC16MCTRL is 8 bit in size • DFLL48M drift compensation: Out-Of-Bounds is typically caused by drift of the reference clock.
RTC – Real-Time Counter on page 341	• No CTRLB register. • Editorial updates.
NVMCTRL – Non-Volatile Memory Controller on page 483	• Register PARAM has device-specific reset value
SERCOM USART – SERCOM Universal Synchronous and Asynchronous Receiver and Transmitter on page 569	• Register RXPL.RXPL: formula corrected.
TCC – Timer/Counter for Control Applications on page 755	• Registers CCBx renamed to CCBUFx • Register PATTB renamed to PATTBUF • Editorial updates.
ADC – Analog-to-Digital Converter on page 1030	• Features: Number of analog inputs updated. – Updated signal name from ADC to AIN. – AREFA/B naming updated to VREFA/B.
USB – Universal Serial Bus on page 889	• Size of register bit field PCKSIZE.BYTE_COUNT is 14 bit.

Electrical Characteristics on page 1140	• GPIO Cluster information moved to I/O Multiplexing and Considerations section. • Section IO Pin Characteristics consolidated.

Packaging Information on page 1180	• Junction Temperature: editorial updates.
Errata on page 1202	Errata for rev.C added. Added new Errata rev.B: • **PM**: Overconsumption under certain condition. *Errata reference 14539* • **DAC**: STATUS.READY not cleared during standby. *Errata reference 14678* • **DAC**: STATUS.READY not set under certain conditions. *Errata reference 14664* • **TRNG**: Power consumption in standby mode. *Errata reference 14827* • **EVSYS**: Spurious overrun can appear under certain conditions. *Errata reference 14532* • **ADC**: Short-term overconsumption on VDDNA under certain conditions. *Errata reference 14349* • **ADC**: ENOB is 9.2 in this revision. *Errata reference 13850* • **DMAC**: Under certain conditions, trigger disabling not functional in standby mode. *Errata reference 14648* • **EIC**: ASYNCH register access impaired unter certain conditions. *Errata reference 14417* Added new Errata rev.A: • **DAC**: STATUS.READY not cleared during standby. *Errata reference 14678* • **DAC**: STATUS.READY not set under certain conditions. *Errata reference 14664* • **EIC**: No asynchronous edge detection. *Errata reference 12949* • **TRNG**: Power consumption in standby mode. *Errata reference 14827* • **EVSYS**: Spurious overrun can appear under certain conditions. *Errata reference 14532* • **BOD33**: Reset occurs under certain conditions. *Errata reference 13918* • **ADC**: Short-term overconsumption on VDDNA under certain conditions. *Errata reference 14349* • **DSU**: MBIST Pause on error not functional. *Errata reference 14324* • **DMAC**: CRC computation may be incorrect. *Errata reference 14324* Changed Errata rev.A: • **Device**: SYSTICK calibration: workaround expanded.*Errata reference 13995*

52.4. Rev E - 07/2015

Ordering Information on page 18	• Device Variant B available.
Block Diagram on page 21	• Editorial updates.
Multiplexed Signals on page 29	• TC4 pins on PA14, PA15, PA18, PA19.
Analog Connections on page 34; analog output signals	• Restrictions for alternative pad functions apply.
OSC32KCTRL – 32KHz Oscillators Controller on page 264	• XOSC32 may affect jitter of neighboring pads.
SUPC – Supply Controller on page 283	• BOD33.HYST is loaded from NVM User Row at startup. • Editorial updates.
SERCOM SPI – SERCOM Serial Peripheral Interface on page 606	• Master Mode serial clock speed up to 12MHz. • Editorial updates.
TC – Timer/Counter on page 704	• DMA requests are cleared by hardware on DMA acknowledge.
USB – Universal Serial Bus on page 889	• LPM transaction endpoint number usually is 0.
CCL – Configurable Custom Logic on page 989	• Editorial updates.
DAC – Digital-to-Analog Converter on page 1102	• Conversion Refresh period updated.
Electrical Characteristics on page 1140	• In Voltage Regulator Characteristics, typical C_{in} is 10µF. • In Brown Out Detector Characteristics, hysteresis description updated. • In ADC Characteristics, RSAMPLE and Rref values are Min.
Analog Components	• Analog output will interfere with alternative functions of output pads.
ERRATA	• Section for Rev.B added. • Rev.A - changed errata: – **Device**: Erratum 14268 renamed to 14588.

52.5. Rev D - 06/2015

Introduction on page 1	• SRAM up to 40KB
Description on page 13	• SRAM up to 40KB • Editorial updates
NVM Software Calibration Area Mapping on page 47	• Updated
Block Diagram on page 21	• Event System connections added • AHB/APB Bridge B connections added
Analog Connections of Peripherals on page 34, OPAMP – Operational Amplifier Controller on page 1007, AC – Analog Comparators on page 1072ADC – Analog-to-Digital Converter on page 1030	• OPAMP, ADC connections updated • Analog ONDEMAND Function - "Alternative Requests" added
ADC – Analog-to-Digital Converter on page 1030	• INPUTCTRL.MUXPOS updated
Processor and Architecture on page 49	• I/O Interface description added • Ultra low latency mode at 0x41008120 • Editorial updates
Performance Level Overview on page 42	• New content
DSU - Device Service Unit on page 87	• ADDR.AMOD added
SUPC – Supply Controller on page 283	• PSOK pin is input • Temperature sensor is configurable for ONDEMAND • EXTWAKE can't wake device from Battery Backup Mode • Editorial updates
EVSYS – Event System on page 536	• USERm table updated • Editorial updates
GCLK - Generic Clock Controller on page 131	• GCLK_TC3 provided by peripheral channel 28 • GENCTRLn[15:8] description added • Editorial updates
RSTC – Reset Controller on page 174	• EXTWAKE can't wake device from Battery Backup Mode • Register BKUPEXIT description updated
PM – Power Manager on page 186	• Sleep Mode Overview - On-Demand functionality noted

NVMCTRL – Non-Volatile Memory Controller on page 483	• Register PARAM at offset 0x08
SERCOM – Serial Communication Interface on page 560	• Reduced feature set for SERCOM5
SERCOM I2C – SERCOM Inter-Integrated Circuit on page 639	• Editorial updates
OSCCTRL – Oscillators Controller on page 217	• Register XOSCCTRL.AMPGC must be set only when XOSC is ready • Editorial updates
OSC32KCTRL – 32KHz Oscillators Controller on page 264	• The XOSC32K signal may affect jitter of neighboring pads • Editorial updates
Packaging Information on page 1180	• QFN center pad notes added
Schematic Checklist on page 1189	• V_{SW}: recommended inductor with saturation current *above 150mA*.
ERRATA	Added new errata: • **ADC**: When window monitor is enabled and its output is 0, GCLK_ADC is kept running. Power consumption will be higher than expected in sleep modes. *Erratum reference: 14449* • **ADC**: The LSB bit of ADC result is stuck '0' in unipolar mode for 8-bit and 10-bit resolution. *Erratum reference: 14431* • **Device**: In IDLE sleep mode, the APB and AHB clocks are not stopped if the FDPLL is running as a GCLK clock source. *Erratum reference: 13401* Changed errata: • **Device**: The low latency mode cannot be enabled by writing '1' to 0x41008120. *Erratum reference: 13506* Removed errata: • Erratum 13918

52.6. Rev C - 03/2015

Configuration Summary on page 15	• Two ACs
NVM User Row Mapping on page 46	• 'BOD12 Enable' renamed to 'BOD12 Disable' • 'BOD33 Enable' renamed to 'BOD33 Disable'

Multiplexed Signals on page 29	• Editorial updates • PTC pins updated for die rev.B • OPAMP pins updated for die rev.B
PAC - Peripheral Access Controller on page 58	• No RFCTRL, ATW, TAL functionality
CCL – Configurable Custom Logic on page 989	• Updated Figure 40-5 Linked LUT Input Selection on page 994 • Filter delay two to five peripheral clock cycles
EIC – External Interrupt Controller on page 464	• Synchronous Edge Detection mode supports Idle sleep mode • Editorial updates
MCLK – Main Clock on page 148	• No TAL functionality • Reset value of APBCMASK register is 0x00007FFF
SERCOM SPI – SERCOM Serial Peripheral Interface on page 606	• Update block diagram
PM – Power Manager on page 186	• PLCFG.PLDIS added for die revision B • STDBYCFG.AVREGSD replaced by STBYCFG.VREGSMOD for die revision B • Power Domain Partition diagram added • Editorial updates
TCC – Timer/Counter for Control Applications on page 755	• Figure 36-28 Waveform Generation with Halt and Restart Actions on page 781 added
OSCCTRL – Oscillators Controller on page 217	• DPLLCTRLB register property 'Enable-Protected' added
• Features on page 1 • DMAC – Direct Memory Access Controller on page 399 • SUPC – Supply Controller on page 283 • TC – Timer/Counter on page 704 • OPAMP – Operational Amplifier Controller on page 1007	Editorial updates

Electrical Characteristics: • Supply Characteristics on page 1141 • Power Consumption on page 1143 • Wake-Up Time on page 1149 • Peripheral Power Consumption • APWS on page 1152 • Power-On Reset (POR) Characteristics on page 1152 • Brown-Out Detectors (BOD) Characteristics on page 1153 • OPAMP – Operational Amplifier Controller on page 1007 • SERCOM I2C – SERCOM Inter-Integrated Circuit on page 639 • Analog-to-Digital Converter (ADC) Characteristics on page 1154 • Digital to Analog Converter (DAC) Characteristics on page 1159 • Oscillators Characteristics on page 1167 • DETREF on page 1163 • OPAMP on page 1163 • NVM Characteristics on page 1166	Parameters updated
ERRATA	Added new errata: • **PTC, OPAMP**: PTC pins Y[1:3], Y[5], Y[12] relocated. OPAMP pins OA_OUT[1] and OA_NEG[2] swapped. (This document describes die rev.B.) *Errata reference 14046* • **SUPC**: Under certain conditions, bit VREF.ONDEMAND has no effect. *Errata reference 14268* • **DAC**: Default value of CTRLB increases loading capacitance of pin PA03. (Fixed for die rev.B.) *Errata reference 14310* • **EVSYS**: Synch/resynch path to PORT not functional. (Fixed for die rev.B.) *Errata reference 14317* Fixed errata: • **PM**: In die rev.A, the wrong voltage regulator was used under certain conditions. (Fixed for die rev.B.) *Errata reference 13901*

52.7. Rev B - 02/2015

Description on page 13	• Moved • CoreMark score updated from 2.14 to 2.46 CoreMark/MHz. • Two 24-bit TCC and one 16-bit TCC • 16 DMA channels • PTC supports up to 192 buttons • Two 12-bit 1MSPS DAC
Analog Connections of Peripherals on page 34	New
Electrical Characteristics on page 1140	New content
Introduction on page 1	New
Low Power Techniques	Removed, refer to PM – Power Manager on page 186
RTC – Real-Time Counter on page 341	Register Summary - COUNT16 and Register Description - COUNT16: Added Comparator 1 related bits to the EVCTRL, INTENCLR, INTENSET and INTFLAG registers.
EIC – External Interrupt Controller on page 464	• 16 pins/channels • Editorial updates
TCC – Timer/Counter for Control Applications on page 755	• Start Event Action operation note updated • No event on pins supported • Waveform double buffering not supported • No Synchronization Ready interrupt • RAMP2C operation added • Editorial updates
AC – Analog Comparators on page 1072, ADC, CCL – Configurable Custom Logic on page 989, DAC – Digital-to-Analog Converter on page 1102, OPAMP – Operational Amplifier Controller on page 1007, PAC - Peripheral Access Controller on page 58, PM – Power Manager on page 186, PORT - I/O Pin Controller on page 506, PTC - Peripheral Touch Controller on page 1135, SERCOM – Serial Communication Interface on page 560, TC – Timer/Counter on page 704	Editorial updates

Schematic Checklist on page 1189	Updated figures in Power Supply Connections on page 1189. Maximum ESR recommendation for 32.768kHz crystal removed from External Real Time Oscillator on page 1195.
ERRATA	Added new errata: • **CCL**: The reset of the RS latch is not functional. *Errata reference 14043* • **TC**: The input capture on IO pins doesn't work. *Errata reference 14024* • **SERCOM**: *Errata reference 14064*

52.8. Rev A - 01/2015

Initial revision.

Atmel | Enabling Unlimited Possibilities®

Atmel Corporation 1600 Technology Drive, San Jose, CA 95110 USA **T:** (+1)(408) 441.0311 **F:** (+1)(408) 436.4200 | **www.atmel.com**