Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

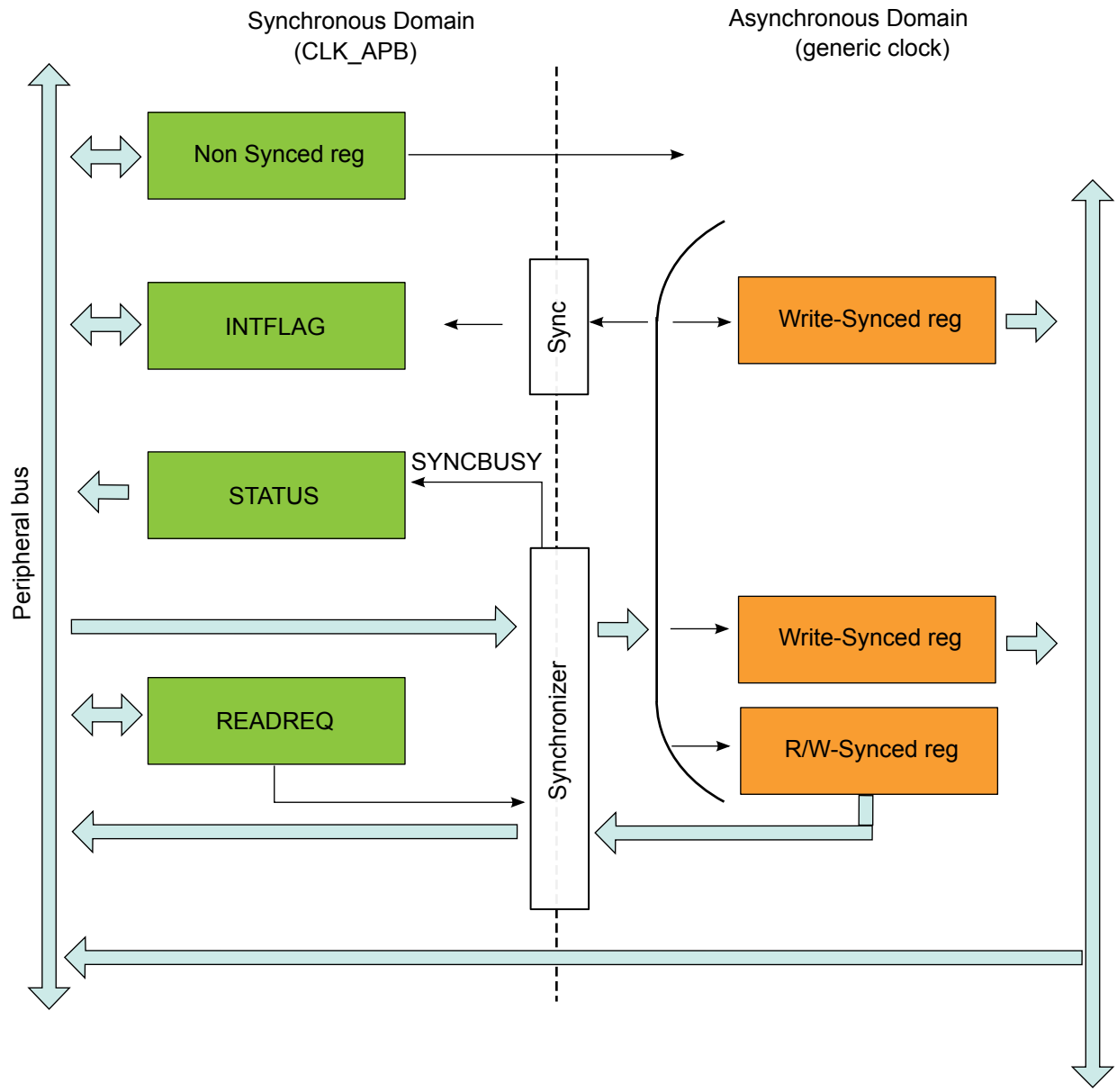
Details	
Product Status	Active
Core Processor	ARM® Cortex®-M0+
Core Size	32-Bit Single-Core
Speed	48MHz
Connectivity	I²C, LINbus, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, I²S, POR, PWM, WDT
Number of I/O	38
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	1.62V ~ 3.6V
Data Converters	A/D 14x12b; D/A 1x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	48-TQFP
Supplier Device Package	48-TQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsamd21g15b-af

3.3 SAM D21J

Table 3-6. Device Variant A

Ordering Code	FLASH (bytes)	SRAM (bytes)	Package	Carrier Type
ATSAMD21J15A-AU	32K	4K	TQFP64	Tray
ATSAMD21J15A-AUT				Tape & Reel
ATSAMD21J15A-AF				Tray
ATSAMD21J15A-AFT				Tape & Reel
ATSAMD21J15A-MU			QFN64	Tray
ATSAMD21J15A-MUT				Tape & Reel
ATSAMD21J15A-MF				Tray
ATSAMD21J15A-MFT				Tape & Reel
ATSAMD21J16A-AU	64K	8K	TQFP64	Tray
ATSAMD21J16A-AUT				Tape & Reel
ATSAMD21J16A-AF				Tray
ATSAMD21J16A-AFT				Tape & Reel
ATSAMD21J16A-MU			QFN64	Tray
ATSAMD21J16A-MUT				Tape & Reel
ATSAMD21J16A-MF				Tray
ATSAMD21J16A-MFT				Tape & Reel
ATSAMD21J16A-CU			UFBGA64	Tray
ATSAMD21J16A-CUT				Tape & Reel
ATSAMD21J17A-AU	128K	16K	TQFP64	Tray
ATSAMD21J17A-AUT				Tape & Reel
ATSAMD21J17A-AF				Tray
ATSAMD21J17A-AFT				Tape & Reel
ATSAMD21J17A-MU			QFN64	Tray
ATSAMD21J17A-MUT				Tape & Reel
ATSAMD21J17A-MF				Tray
ATSAMD21J17A-MFT				Tape & Reel
ATSAMD21J17A-CU			UFBGA64	Tray
ATSAMD21J17A-CUT				Tape & Reel

Figure 14-3. Synchronization



14.3.1.2 Write-Synchronization

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

When the write-synchronization is ongoing (STATUS.SYNCBUSY is one), any of the following actions will cause the peripheral bus to stall until the synchronization is complete:

- Writing a generic clock peripheral core register
- Reading a read-synchronized peripheral core register
- Reading the register that is being written (and thus triggered the synchronization)

Peripheral core registers without read-synchronization will remain static once they have been written and synchronized, and can be read while the synchronization is ongoing without causing the peripheral bus to

Value	Name	Description
0x1D	GCLK_TC6, GCLK_TC7	TC6,TC7
0x1E	GCLK_ADC	ADC
0x1F	GCLK_AC_DIG	AC_DIG
0x20		
0x21	GCLK_AC_ANA	AC_ANA
0x22		
0x23	GCLK_DAC	DAC
0x24	GCLK_PTC	PTC
0x25	GCLK_I2S_0	I2S_0
0x26	GCLK_I2S_1	I2S_1
0x27-0x3F	-	Reserved

15.8.4 Generic Clock Generator Control

Name: GENCTRL
Offset: 0x4
Reset: 0x00000000
Property: Write-Protected, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			RUNSTDBY	DIVSEL	OE	OOV	IDC	GENEN
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
				SRC[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
					ID[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 21 – RUNSTDBY: Run in Standby
This bit is used to keep the generic clock generator running when it is configured to be output to its dedicated GCLK_IO pin. If GENCTRL.OE is zero, this bit has no effect and the generic clock generator will only be running if a peripheral requires the clock.

XOSC.RUNSTDBY	XOSC.ONDEMAND	XOSC.ENABLE	Sleep Behavior
-	-	0	Disabled
0	0	1	Always run in IDLE sleep modes. Disabled in STANDBY sleep mode.
0	1	1	Only run in IDLE sleep modes if requested by a peripheral. Disabled in STANDBY sleep mode.
1	0	1	Always run in IDLE and STANDBY sleep modes.
1	1	1	Only run in IDLE or STANDBY sleep modes if requested by a 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 (XOSC.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 Power and Clock Status register (PCLKSR.XOSCRDY) is set when the user-selected start-up time is over. An interrupt is generated on a zero-to-one transition on PCLKSR.XOSCRDY if the External Multipurpose Crystal Oscillator Ready bit in the Interrupt Enable Set register (INTENSET.XOSCRDY) is set.

Note: Do not enter standby mode when an oscillator is in start-up:
Wait for the OSCxRDY bit in SYSCTRL.PCLKSR register to be set before going into standby mode.

Related Links

GCLK - Generic Clock Controller

17.6.3 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

The XOSC32K can be used as a source for generic clock generators, as described in the *GCLK – Generic Clock Controller*.

At power-on reset (POR) the XOSC32K is disabled, and the XIN32/XOUT32 pins can 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, XIN32 and XOUT32 are controlled by the SYSCTRL, and GPIO functions are overridden on both pins. When in external clock mode, only the XIN32 pin will be overridden and controlled by the SYSCTRL, while the XOUT32 pin can still be used as a GPIO pin.

The external clock or crystal oscillator is enabled by writing a one to the Enable bit (XOSC32K.ENABLE) in the 32kHz External Crystal Oscillator Control register. To enable the XOSC32K as a crystal oscillator, a one must be written to the XTAL Enable bit (XOSC32K.XTALEN). If XOSC32K.XTALEN is zero, external clock input will be enabled.

The oscillator is disabled by writing a zero to the Enable bit (XOSC32K.ENABLE) in the 32kHz External Crystal Oscillator Control register while keeping the other bits unchanged. Writing to the

Figure 17-3. CK and CLK_FDPLL96M Off Mode to Running Mode

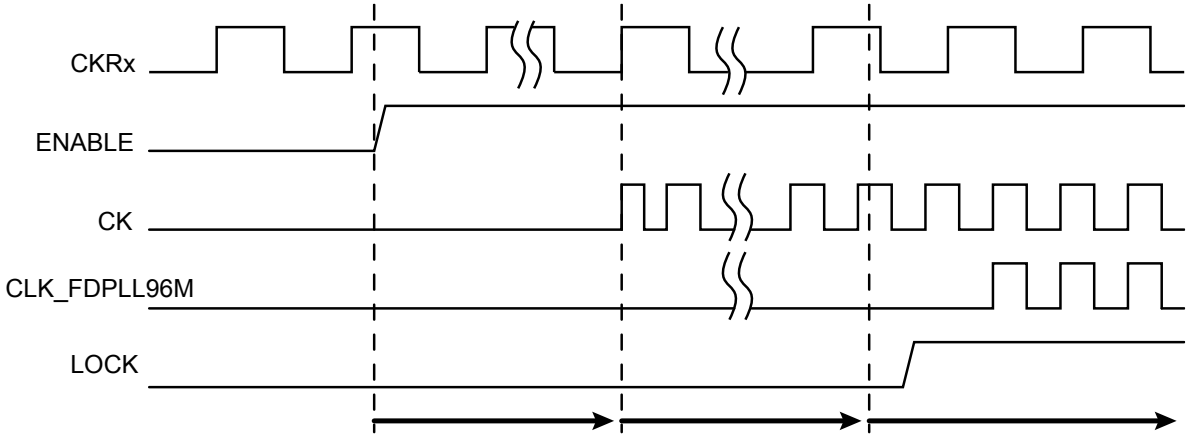


Figure 17-4. CK and CLK_FDPLL96M Off Mode to Running Mode when Wake-Up Fast is Activated

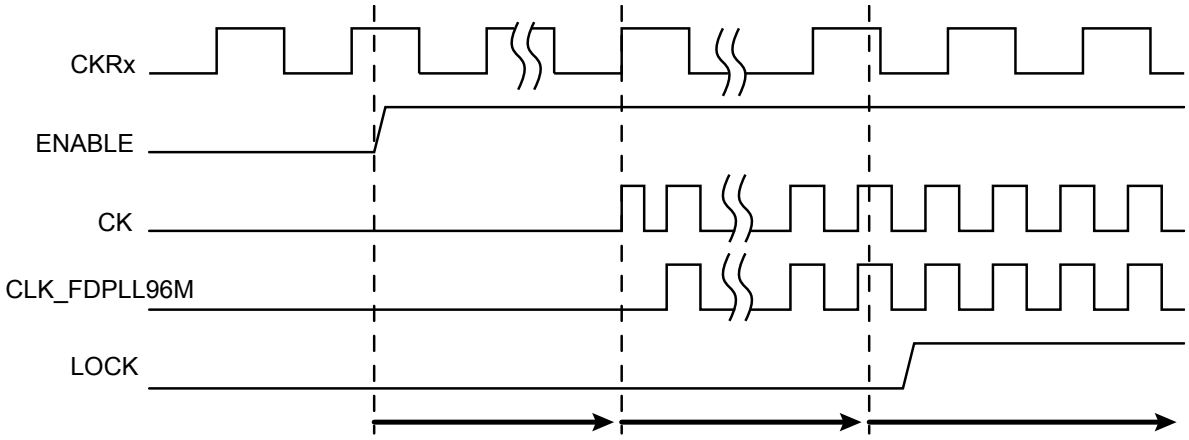
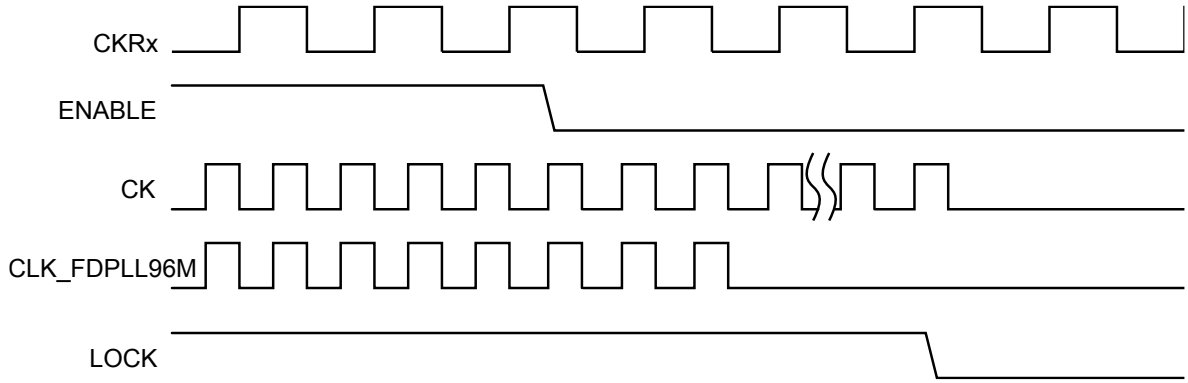


Figure 17-5. CK and CLK_FDPLL96M Running Mode to Off Mode



17.6.8.5 Reference Clock Switching

When a software operation requires reference clock switching, the normal operation is to disable the FDPLL96M, modify the DPLLCTRLB.REFCLK to select the desired reference source and activate the FDPLL96M again.

17.6.8.6 Loop Divider Ratio updates

The FDPLL96M supports on-the-fly update of the DPLLRATIO register, so it is allowed to modify the loop divider ratio and the loop divider ratio fractional part when the FDPLL96M is enabled. At that time, the DPLLSTATUS.LOCK bit is cleared and set again by hardware when the output frequency reached a stable state. The DPLL Lock Fail bit in the Interrupt Flag Status and Clear register (INTFLAG.DPLLLCK)

- DFLLRCS - DFLL48M Reference Clock has Stopped: A “0-to-1” transition on the PCLKSR.DFLLRCS bit is detected
- BOD33RDY - BOD33 Ready: A “0-to-1” transition on the PCLKSR.BOD33RDY bit is detected
- BOD33DET - BOD33 Detection: A “0-to-1” transition on the PCLKSR.BOD33DET bit is detected. This is an asynchronous interrupt and can be used to wake-up the device from any sleep mode.
- B33SRDY - BOD33 Synchronization Ready: A “0-to-1” transition on the PCLKSR.B33SRDY bit is detected
- BOD12RDY - BOD12 Ready: A “0-to-1” transition on the PCLKSR.BOD12RDY bit is detected
- BOD12DET - BOD12 Detection: A “0-to-1” transition on the PCLKSR.BOD12DET bit is detected
- B12SRDY - BOD12 Synchronization Ready: A “0-to-1” transition on the PCLKSR.B12SRDY bit is detected
- PLL Lock (LOCK): Indicates that the DPLL Lock bit is asserted.
- PLL Lock Lost (LOCKL): Indicates that a falling edge has been detected on the Lock bit during normal operation mode.
- PLL Lock Timer Timeout (LTTO): This interrupt flag indicates that the software defined time DPLLCTRLB.LTIME has elapsed since the start of the FDPLL96M.

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 SYSCTRL is reset. See Interrupt Flag Status and Clear (INTFLAG) register 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: 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

17.6.14 Synchronization

Due to the multiple clock domains, values in the DFLL48M control registers need to be synchronized to other clock domains. The status of this synchronization can be read from the Power and Clocks Status register (PCLKSR). Before writing to any of the DFLL48M control registers, the user must check that the DFLL Ready bit (PCLKSR.DFLLRDY) in PCLKSR is set to one. 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 zero will be ignored. An interrupt is generated on a zero-to-one transition of DFLLRDY if the DFLLRDY bit (INTENSET.DFLLDY) in the Interrupt Enable Set register is set.

In order to read from any of the DFLL48M configuration registers, the user must request a read synchronization by writing a one to DFLLSYNC.READREQ. The registers can be read only when PCLKSR.DFLLRDY is set. If DFLLSYNC.READREQ is not written before a read, a synchronization will be started, and the bus will be halted until the synchronization is complete. Reading the DFLL48M registers when the DFLL48M is disabled will not halt the bus.

Bit	7	6	5	4	3	2	1	0
	OVF	SYNCRDY						CMP0
Access	R/W	R/W						R/W
Reset	0	0						0

Bit 7 – OVF: Overflow Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Overflow Interrupt Enable bit and disable the corresponding interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled, and an interrupt request will be generated when the Overflow interrupt flag is set.

Bit 6 – SYNCRDY: Synchronization Ready Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Synchronization Ready Interrupt Enable bit and disable the corresponding interrupt.

Value	Description
0	The Synchronization Ready interrupt is disabled.
1	The Synchronization Ready interrupt is enabled, and an interrupt request will be generated when the Synchronization Ready interrupt flag is set.

Bit 0 – CMP0: Compare 0 Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Compare 0 Interrupt Enable bit and disable the corresponding interrupt.

Value	Description
0	The Compare 0 interrupt is disabled.
1	The Compare 0 interrupt is enabled, and an interrupt request will be generated when the Compare x interrupt flag is set.

19.8.9 Interrupt Enable Clear - MODE1

Name: INTENCLR
Offset: 0x06
Reset: 0x00
Property: Write-Protected

Bit	7	6	5	4	3	2	1	0
	OVF	SYNCRDY					CMP1	CMP0
Access	R/W	R/W					R/W	R/W
Reset	0	0					0	0

Bit 7 – OVF: Overflow Interrupt Enable
Writing a zero to this bit has no effect.

20.6.2 Basic Operation

20.6.2.1 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.

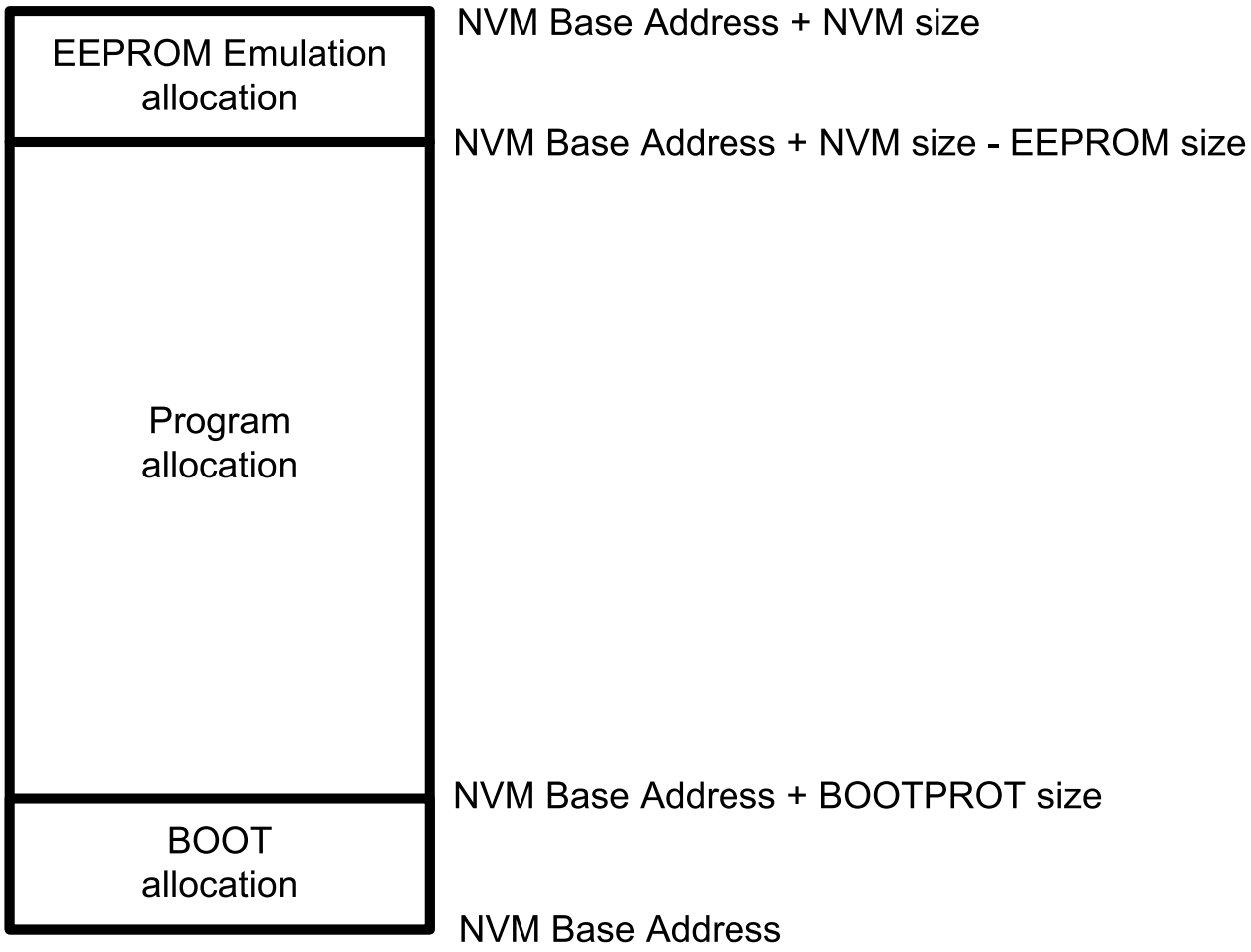
Before the DMAC is enabled it must be configured, as outlined by the following steps:

- The SRAM address of where the descriptor memory section is located must be written to the Description Base Address (BASEADDR) register
- The 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)

Figure 22-4. EEPROM and Boot Loader Allocation



Related Links

Physical Memory Map

22.6.3 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 22-1. 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

Value	Description
0	The corresponding I/O pin in the PORT group is configured as an input.
1	The corresponding I/O pin in the PORT group is configured as an output.

23.8.2 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
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 PORT group will keep its configuration.
1	The corresponding I/O pin in the PORT group is configured as input.

23.8.3 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

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 26-3. 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

27. SERCOM SPI – SERCOM Serial Peripheral Interface

27.1 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. 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

27.2 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, $f_{SCK}=1/t_{SCK}$(1)
  - 8-bit clock generator
  - Hardware controlled $\overline{SS}$
- Slave operation:
  - Serial clock speed, $f_{SCK}=1/t_{SSCK}$(1)
  - Optional 8-bit address match operation
  - Operation in all sleep modes
  - Wake on $\overline{SS}$ transition

1. For t_{SCK} and t_{SSCK} values, refer to SPI Timing Characteristics.

Related Links

SERCOM in SPI Mode Timing

SERCOM – Serial Communication Interface

Features

- Data (DATA) when in master operation

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

Related Links

Register Synchronization

Bit 31 – MCKOUTINV: Master Clock Output Invert

Value	Description
0	The Master Clock n is output without inversion.
1	The Master Clock n is inverted before being output.

Bit 30 – SCKOUTINV: Serial Clock Output Invert

Value	Description
0	The Serial Clock n is output without inversion.
1	The Serial Clock n is inverted before being output.

Bit 29 – FSOUTINV: Frame Sync Output Invert

Value	Description
0	The Frame Sync n is output without inversion.
1	The Frame Sync n is inverted before being output.

Bits 28:24 – MCKOUTDIV[4:0]: Master Clock Output Division Factor
The generic clock selected by MCKSEL is divided by (MCKOUTDIV + 1) to obtain the Master Clock n output.

Bits 23:19 – MCKDIV[4:0]: Master Clock Division Factor
The Master Clock n is divided by (MCKDIV + 1) to obtain the Serial Clock n.

Bit 18 – MCKEN: Master Clock Enable

Value	Description
0	The Master Clock n division and output is disabled.
1	The Master Clock n division and output is enabled.

Bit 16 – MCKSEL: Master Clock Select
This field selects the source of the Master Clock n.

MCKSEL	Name	Description
0x0	GCLK	GCLK_I2S_n is used as Master Clock n source
0x1	MCKPIN	MCKn input pin is used as Master Clock n source

Bit 12 – SCKSEL: Serial Clock Select
This field selects the source of the Serial Clock n.

SCKSEL	Name	Description
0x0	MCKDIV	Divided Master Clock n is used as Serial Clock n source
0x1	SCKPIN	SCKn input pin is used as Serial Clock n source

Bit 11 – FSINV: Frame Sync Invert

Value	Description
0	The Frame Sync n is used without inversion.
1	The Frame Sync n is inverted before being used.

PRESCALER[2:0]	Name	Description
0x2	DIV16	Peripheral clock divided by 16
0x3	DIV32	Peripheral clock divided by 32
0x4	DIV64	Peripheral clock divided by 64
0x5	DIV128	Peripheral clock divided by 128
0x6	DIV256	Peripheral clock divided by 256
0x7	DIV512	Peripheral clock divided by 512

Bits 5:4 – RESSEL[1:0]: Conversion Result Resolution
These bits define whether the ADC completes the conversion at 12-, 10- or 8-bit result resolution.

RESSEL[1:0]	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 GAINCAL and OFFSETCAL registers. Conversion time will be increased by X 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 is 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.

Table 37-15. I^2C Pins Characteristics in I^2C configuration

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
R_{PULL}	Pull-up - Pull-down resistance	I	20	40	60	kΩ
V_{IL}	Input low-level voltage	V_{DD}=1.62V-2.7VI	-	-	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-3.63VI	0.55*V_{DD}	-	-	
V_{HYS}	Hysteresis of Schmitt trigger inputs		0.08*V_{DD}	-	-	
V_{OL}	Output low-level voltage	V_{DD}> 2.0V, I_{OL}=3mA	-	-	0.4	
		V_{DD}≤2.0V , I_{OL}=2mA	-	-	0.2*V_{DD}	
I_{OL}	Output low-level current	V_{OL} =0.4V Standard, Fast and HS Modes	3			mA
		V_{OL} =0.4V Fast Mode +	20	-	-	
		V_{OL} =0.6V	6	-	-	
f_{SCL}	SCL clock frequency	I	-	-	3.4	MHz

I^2C pins timing characteristics can be found in SERCOM in I2C Mode Timing.

37.8.3 XOSC Pin

XOSC pins behave as normal pins when used as normal I/Os. Refer to table Normal I/O Pins.

37.8.4 XOSC32 Pin

XOSC32 pins behave as normal pins when used as normal I/Os. Refer to table Normal I/O Pins.

37.8.5 External Reset Pin

Reset pin has the same electrical characteristics as normal I/O pins. Refer to table Normal I/O Pins.

37.9 Injection Current

Stresses beyond those listed in the table below 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 37-28. Averaging Feature (Device Variant B and C)

Average Number	Conditions	SNR (dB)	SINAD (dB)	SFDR (dB)	ENOB (bits)
1	In differential mode, 1x gain, V_{DDANA}=3.0V, V_{REF}=1.0V, 350kSps at 25°C	66.0	65.0	72.8	10.5
8		67.6	65.8	75.1	10.62
32		69.7	67.1	75.3	10.85
128		70.4	67.5	75.5	10.91

37.10.4.2 Performance with the hardware offset and gain correction

Inherent gain and offset errors affect the absolute accuracy of the ADC. The offset error cancellation is handled by the Offset Correction register (OFFSETCORR) and the gain error cancellation, by the Gain Correction register (GAINCORR). The offset and gain correction value is subtracted from the converted data before writing the Result register (RESULT).

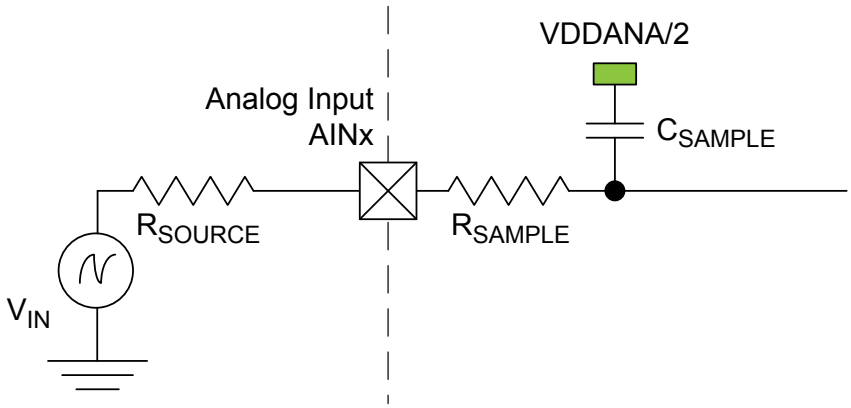
Table 37-29. Offset and Gain correction feature

Gain Factor	Conditions	Offset Error (mV)	Gain Error (mV)	Total Unadjusted Error (LSB)
0.5x	In differential mode, 1x gain, V_{DDANA}=3.0V, V_{REF}=1.0V, 350kSps at 25°C	0.25	1.0	2.4
1x		0.20	0.10	1.5
2x		0.15	-0.15	2.7
8x		-0.05	0.05	3.2
16x		0.10	-0.05	6.1

37.10.4.3 Inputs and Sample and Hold Acquisition Times

The analog voltage source must be able to charge the sample and hold (S/H) capacitor in the ADC in order to achieve maximum accuracy. Seen externally the ADC input consists of a resistor (R_{SAMPLE}) and a capacitor (C_{SAMPLE}). In addition, the source resistance (R_{SOURCE}) must be taken into account when calculating the required sample and hold time. The next figure shows the ADC input channel equivalent circuit.

Figure 37-5. ADC Input



To achieve n bits of accuracy, the C_{SAMPLE} capacitor must be charged at least to a voltage of

40.1.2.3 PM

1 – In debug mode, if a watchdog reset occurs, the debug session is lost.
Errata reference: 12196
Fix/Workaround:
A new debug session must be restart after a watchdog reset.

40.1.2.4 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 – 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.

3 – 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.

40.1.2.5 XOSC32K

1 – The automatic amplitude control of the XOSC32K does not work.
Errata reference: 10933
Fix/Workaround:
Use the XOSC32K with Automatic Amplitude control disabled (XOSC32K.AAMPEN = 0)

40.1.2.6 FDPLL

1 – When changing on-the-fly the FDPLL ratio in DPLLnRATIO register, STATUS.DPLLnLDRTO will not be set when the ratio update will be completed.
Errata reference: 15753
Fix/Workaround:
Wait for the interruption flag INTFLAG.DPLLnLDRTO instead.

40.1.2.7 DMAC

1 – When at least one channel using linked descriptors is already active, enabling another DMA channel (with or without linked descriptors) can result in a channel Fetch Error (FERR) or an incorrect descriptor fetch.

Errata reference: 12368
Fix/Workaround:
For pin PA24 and PA25, the GPIO pull-up and pull-down must be disabled before enabling alternative functions on them.

5 – If APB clock is stopped and GCLK clock is running, APB read access to read-synchronized registers will freeze the system. The CPU and the DAP AHB-AP are stalled, as a consequence debug operation is impossible.
Errata reference: 10416
Fix/Workaround:
Do not make read access to read-synchronized registers when APB clock is stopped and GCLK is running. To recover from this situation, power cycle the device or reset the device using the RESETN pin.

6 – 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

Symbol	Description	Mode	VDD=1.8V			VDD=3.3V			Units
			Min.	Typ.	Max.	Min.	Typ.	Max	
d_{M_MCKI}	I2S MCK duty cycle	Master mode, pin is input (1b)		50			50		%
t_{M_SCKOR}	I2S SCK rise time(2)	Master mode / Capacitive load CL = 15 pF			9.7			4.6	ns
t_{M_SCKOF}	I2S SCK fall time(3)	Master mode / Capacitive load CL = 15 pF			10.2			4.5	ns
d_{M_SCKO}	I2S SCK duty cycle	Master mode	45.6		50	45.6		50	%
f_{M_SCKO}, 1/ t_{M_SCKO}	I2S SCK frequency	Master mode, Supposing external device response delay is 30ns			7.8			9.5	MHz
f_{S_SCKI}, 1/ t_{S_SCKI}	I2S SCK frequency	Slave mode, Supposing external device response delay is 30ns			14.4			14.8	MHz
d_{S_SCKO}	I2S SCK duty cycle	Slave mode		50			50		%
t_{M_FSOV}	FS valid time	Master mode			4.1			4	ns
t_{M_FSOH}	FS hold time	Master mode	-0.9			-0.9			ns
t_{S_FSIS}	FS setup time	Slave mode	2.3			1.5			ns
t_{S_FSIH}	FS hold time	Slave mode	0			0			ns
t_{M_SDIS}	Data input setup time	Master mode	36.2			24.5			ns
t_{M_SDIH}	Data input hold time	Master mode	-8.2			-8.2			ns
t_{S_SDIS}	Data input setup time	Slave mode	4.8			3.9			ns
t_{S_SDIH}	Data input hold time	Slave mode	1.2			1.2			ns
t_{M_SDOV}	Data output valid time	Master transmitter			5.9			4.8	ns
t_{M_SDOH}	Data output hold time	Master transmitter	-0.5			-0.5			ns
t_{S_SDOV}	Data output valid time	Slave transmitter			37.8			25.9	ns
t_{S_SDOH}	Data output hold time	Slave transmitter	37.6			25.7			ns
t_{PDM2LS}	Data input setup time	Master mode PDM2 Left	36.2			24.5			ns
t_{PDM2LH}	Data input hold time	Master mode PDM2 Left	-8.2			-8.2			ns