



Welcome to [E-XFL.COM](https://www.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	I <sup>2</sup> C, SCI, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, I <sup>2</sup> S, POR, PWM, WDT
Number of I/O	52
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.63V
Data Converters	A/D 16x12b; D/A 1x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/atsamda1j16b-abt">https://www.e-xfl.com/product-detail/microchip-technology/atsamda1j16b-abt</a>

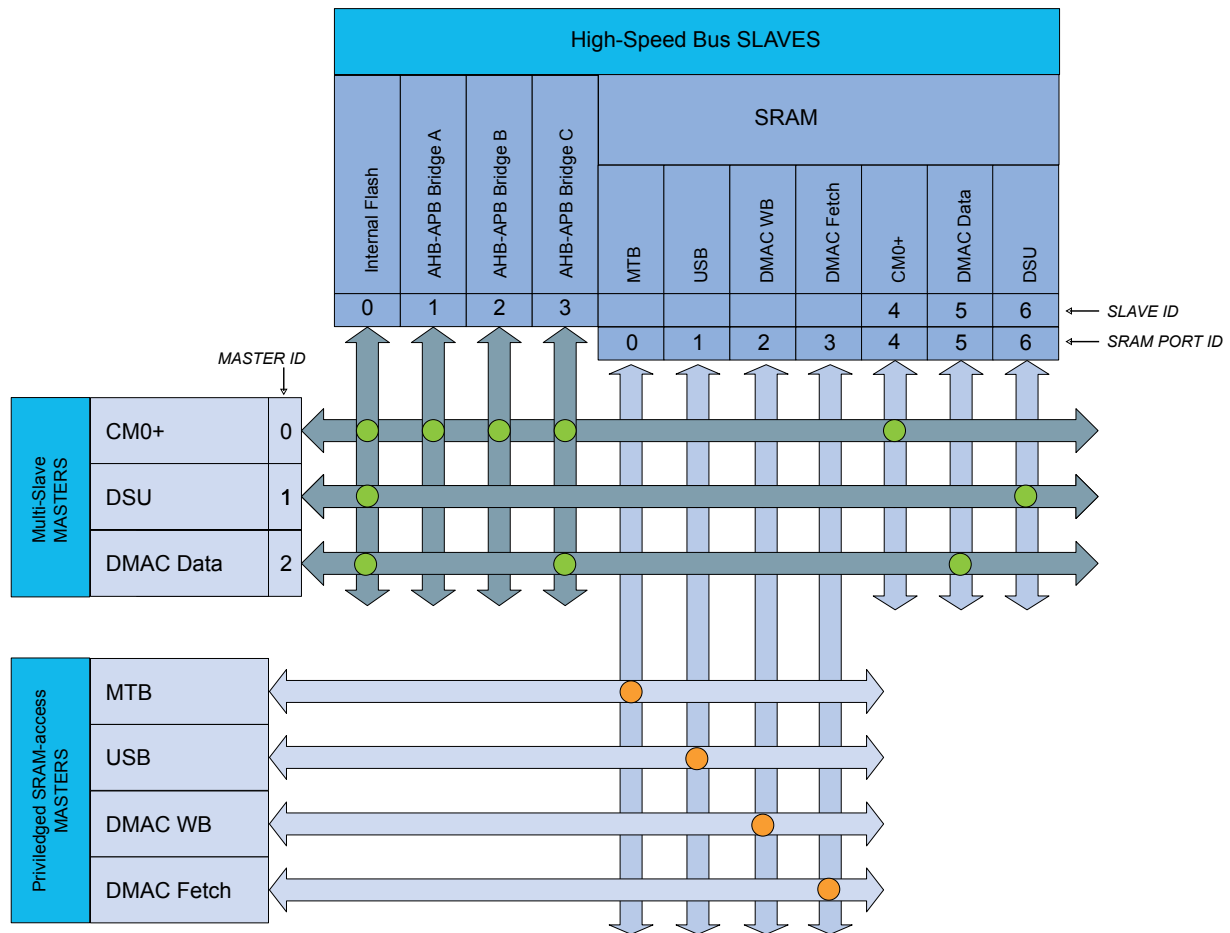
## 13.4 High-Speed Bus System

### 13.4.1 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

### 13.4.2 Configuration



**Table 13-4. Bus Matrix Masters**

Bus Matrix Masters	Master ID
CM0+ - Cortex M0+ Processor	0
DSU - Device Service Unit	1
DMAC - Direct Memory Access Controller / Data Access	2

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 [ID-00001c14]

**Reset:** see related links

**Property:** PAC Write-Protection

Writing a one to this bit will clear the OSC8M Ready Interrupt Enable bit, which disables the OSC8M Ready interrupt.

Value	Description
0	The OSC8M Ready interrupt is disabled.
1	The OSC8M Ready interrupt is enabled, and an interrupt request will be generated when the OSC8M Ready Interrupt flag is set.

#### Bit 2 – OSC32KRDY: OSC32K Ready Interrupt Enable

Writing a zero to this bit has no effect.

Writing a one 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, and an interrupt request will be generated when the OSC32K Ready Interrupt flag is set.

#### Bit 1 – XOSC32KRDY: XOSC32K Ready Interrupt Enable

Writing a zero to this bit has no effect.

Writing a one 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, and an interrupt request will be generated when the XOSC32K Ready Interrupt flag is set.

#### Bit 0 – XOSCRDY: XOSC Ready Interrupt Enable

Writing a zero to this bit has no effect.

Writing a one 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.

### 19.8.2 Interrupt Enable Set

**Name:** INTENSET  
**Offset:** 0x04 [ID-00003d5d]  
**Reset:** 0x00000000  
**Property:** Write-Protected

**Table 21-1. MODE0 - Mode Register Summary**

Offset	Name	Bit Pos.								
0x00	CTRL	7:0	MATCHCLR				MODE[1:0]		ENABLE	SWRST
0x01		15:8					PRESCALER[3:0]			
0x02	READREQ	7:0					ADDR[5:0]			
0x03		15:8	RREQ	RCONT						
0x04	EVCTRL	7:0	PEREO7	PEREO6	PEREO5	PEREO4	PEREO3	PEREO2	PEREO1	PEREO0
0x05		15:8	OVFEO							CMPEO0
0x06	INTENCLR	7:0	OVF	SYNCRDY						CMP0
0x07	INTENSET	7:0	OVF	SYNCRDY						CMP0
0x08	INTFLAG	7:0	OVF	SYNCRDY						CMP0
0x09	Reserved									
0x0A	STATUS	7:0	SYNCBUSY							
0x0B	DBGCTRL	7:0								DBGRUN
0x0C	FREQCORR	7:0	SIGN				VALUE[6:0]			
0x0D ... 0x0F	Reserved									
0x10	COUNT	7:0					COUNT[7:0]			
0x11		15:8					COUNT[15:8]			
0x12		23:16					COUNT[23:16]			
0x13		31:24					COUNT[31:24]			
0x14 ... 0x17	Reserved									
0x18	COMP0	7:0					COMP[7:0]			
0x19		15:8					COMP[15:8]			
0x1A		23:16					COMP[23:16]			
0x1B		31:24					COMP[31:24]			

**Table 21-2. MODE1 - Mode Register Summary**

Offset	Name	Bit Pos.								
0x00	CTRL	7:0					MODE[1:0]		ENABLE	SWRST
0x01		15:8					PRESCALER[3:0]			
0x02	READREQ	7:0					ADDR[5:0]			
0x03		15:8	RREQ	RCONT						
0x04	EVCTRL	7:0	PEREO7	PEREO6	PEREO5	PEREO4	PEREO3	PEREO2	PEREO1	PEREO0
0x05		15:8	OVFEO						CMPEO1	CMPEO0
0x06	INTENCLR	7:0	OVF	SYNCRDY					CMP1	CMP0
0x07	INTENSET	7:0	OVF	SYNCRDY					CMP1	CMP0
0x08	INTFLAG	7:0	OVF	SYNCRDY					CMP1	CMP0
0x09	Reserved									
0x0A	STATUS	7:0	SYNCBUSY							
0x0B	DBGCTRL	7:0								DBGRUN
0x0C	FREQCORR	7:0	SIGN				VALUE[6:0]			

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.

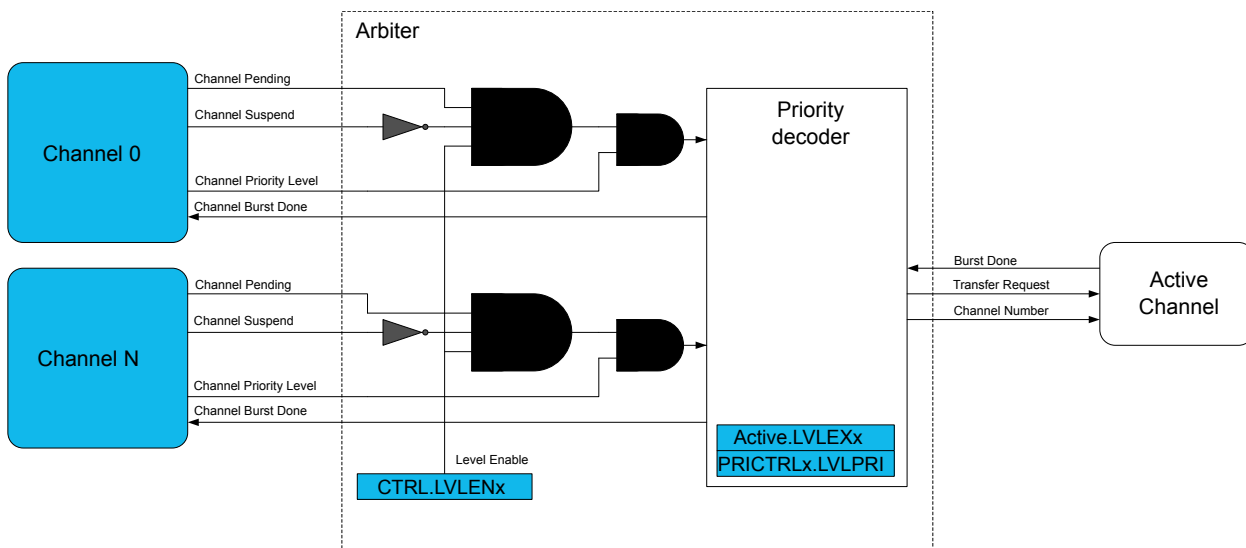
If the upcoming burst transfer is the first for the transfer request, the corresponding Busy Channel x bit in the Busy Channels register will be set (**BUSYCH**.BUSYCHx=1), and it will remain '1' for the subsequent granted burst transfers.

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 22-4. 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 higher priority level number will have priority over a channel with a lower 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).

## 22.8.10 Interrupt Pending

This register allows the user to identify the lowest DMA channel with pending interrupt.

**Name:** INTPEND

**Offset:** 0x20 [ID-00001ece]

**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

### 23.5.5 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](#)

### 23.5.6 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](#)

### 23.5.7 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.

### 23.5.8 Register Access Protection

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

### 23.5.9 Analog Connections

Not applicable.

## 23.6 Functional Description

### 23.6.1 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

### 23.6.2 Basic Operation

#### 23.6.2.1 Initialization

The EIC must be initialized in the following order:



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.

#### 24.6.4 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.

The CTRLB register must be used to control the power reduction mode, read wait states, and the write mode.

##### 24.6.4.1 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.

##### 24.6.4.2 RWWEE Read

Reading from the RWW EEPROM address space is performed via the AHB bus by addressing the RWWEE address space directly.

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.

##### 24.6.4.3 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.

### 26.6.3 Interrupts

The EVSYS has the following interrupt sources:

- Overrun Channel n (OVRn): for details, refer to *The Overrun Channel n Interrupt* section.
- Event Detected Channel n (EVDn): for details, refer to *The Event Detected Channel n Interrupt* section.

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 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](#)

[Sleep Mode Controller](#)

#### 26.6.3.1 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 resynchronized paths. In the case of asynchronous path, the INTFLAG.OVRn is always read as zero.

#### Related Links

[Nested Vector Interrupt Controller](#)

#### 26.6.3.2 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 resynchronized path. In the case of asynchronous path, the INTFLAG.EVDn is always zero.

#### Related Links

[Nested Vector Interrupt Controller](#)

### 26.6.4 Sleep Mode Operation

The EVSYS can generate interrupts to wake up the device from any sleep mode.

## 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.

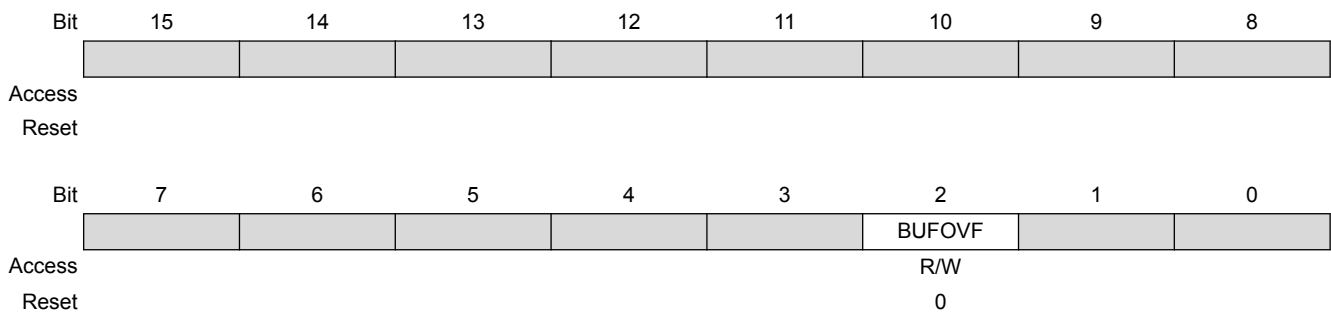
## 29.8.7 Status

**Name:** STATUS

**Offset:** 0x1A [ID-00000e74]

**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.

## 29.8.8 Synchronization Busy

**Name:** SYNCBUSY

**Offset:** 0x1C [ID-00000e74]

**Reset:** 0x00000000

**Property:** -

**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 <sup>2</sup> C 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<sup>2</sup>C 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.

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<sup>2</sup>C 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<sup>2</sup>C 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.

In mono format, Transmit mode, data written to the left channel is duplicated to the right output channel. In mono format, Receiver mode, data received from the right channel is ignored and data received from the left channel is duplicated in to the right channel.

In mono format, TDM Transmit mode with more than two slots, data written to the even-numbered slots is duplicated in to the following odd-numbered slot.

In mono format, TDM Receiver mode with more than two slots, data received from the even-numbered slots is duplicated in to the following odd-numbered slot.

Mono format can be enabled by writing a '1' to the MONO bit in the Serializer m Control register (SERCTRLm.MONO).

I<sup>2</sup>S support different data frame formats:

- 2-channel I<sup>2</sup>S with Word Select
- 1- to 8-slot Time Division Multiplexed (TDM) with Frame Sync and individually enabled slots
- 1- or 2-channel Pulse Density Modulation (PDM) reception for MEMS microphones
- 1-channel burst transfer with non-periodic Frame Sync

In 2 channel I<sup>2</sup>S mode, number of slots configured is one or two and successive data words corresponds to left and right channel. Left and right channel are identified by polarity of Word Select signal (FSn signal). Each frame consists of one or two data word(s). In the case of compact stereo format, the number of slots can be one. When 32-bit slot size is used, the number of slots can be two.

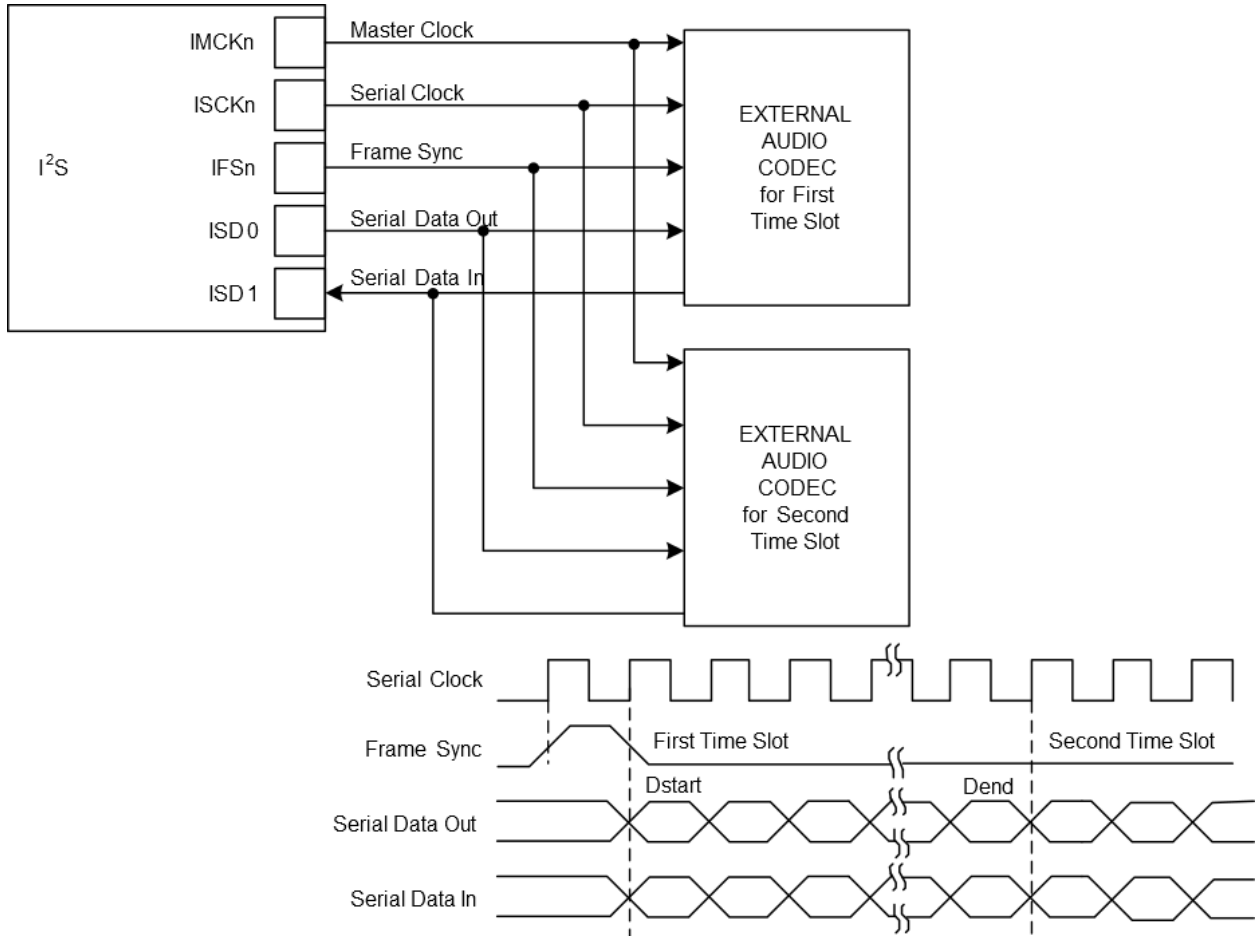
In TDM format, number slots can be configured up to 8 slots. If 4 slots are configured, each frame consists of 4 data words.

In PDM format, continuous 1-bit data samples are available on the SDm line for each SCKn rising and SCKn falling edge as in case of a MEMS microphone with PDM interface.

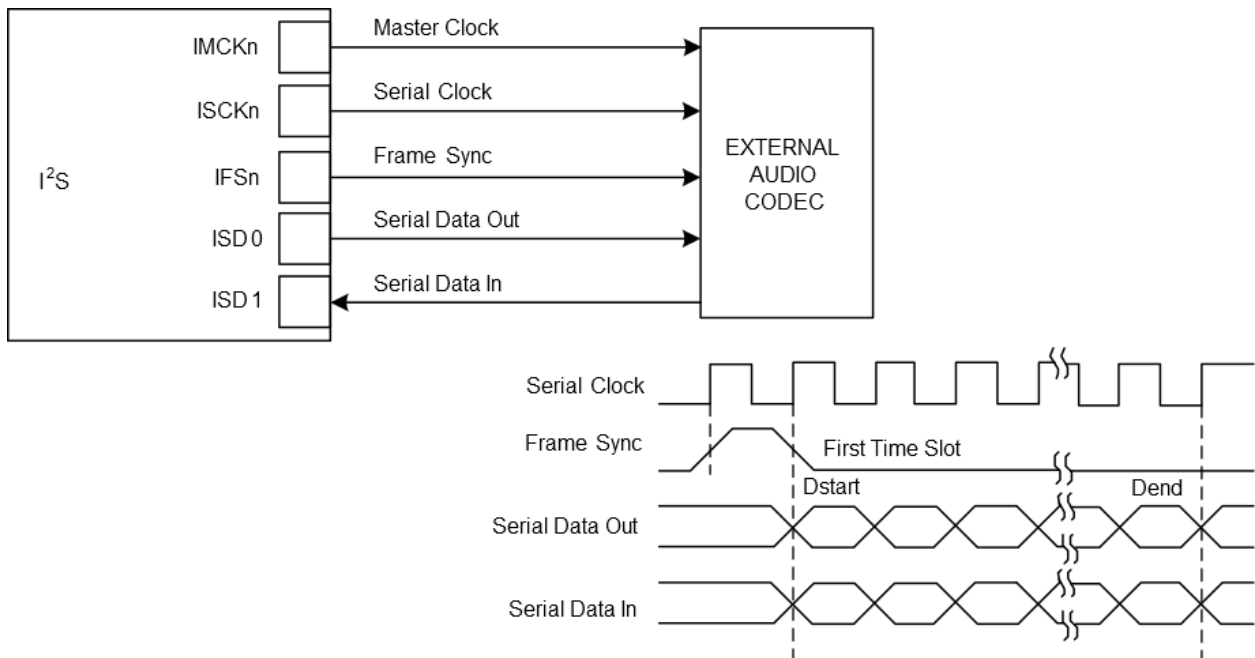
1-channel burst transfer with non-periodic Frame Sync mode is useful typically for passing control non-auto data as in case of DSP. In Burst mode, a single Data transfer starts at each Frame Sync pulse, and these pulses are 1-bit wide and occur only when a Data transfer is requested.

Sections [I<sup>2</sup>S Format - Reception and Transmission Sequence with Word Select](#), [TDM Format - Reception and Transmission Sequence](#) and [I<sup>2</sup>S Application Examples](#) describe more about frame/data formats and register settings required for different I<sup>2</sup>S applications.

**Figure 31-8. Time Slot Application Block Diagram**



**Figure 31-9. Codec Application Block Diagram**



- 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. 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 C register (CTRLC.CAPTEN).
- 8. If desired, enable inversion of the waveform output or IO pin input signal for individual channels via the Waveform Output Invert Enable bit group in the Control C register (CTRLC.INVEN).

### 32.6.2.2 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 disabled by writing a zero to CTRLA.ENABLE.

The TC is reset by writing a one 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, and the TC will be disabled. Refer to the CTRLA register for details.

The TC should be disabled before the TC is reset in order to avoid undefined behavior.

### 32.6.2.3 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.

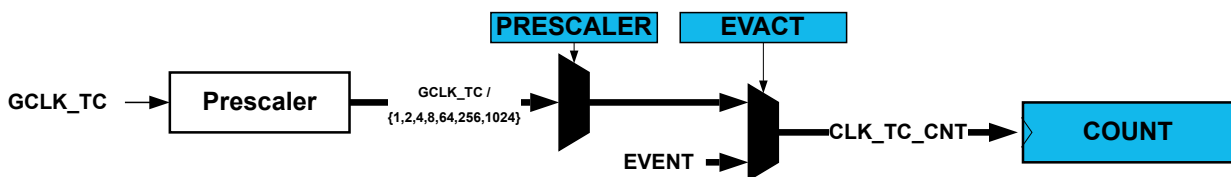
If the prescaler value is higher than one, the counter update condition can be optionally executed on the next GCLK\_TCx clock pulse or the next prescaled clock pulse. For further details, refer to Prescaler (CTRLA.PRESCALER) and Counter Synchronization (CTRLA.PRESYNC) description.

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 32-2. Prescaler**



### 32.6.2.4 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 register (PER). This register is used to store the period value that can be used as the top value for waveform generation.
- 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. TC3 is paired with TC4, and TC5 is paired with TC6. TC7 does not support 32-bit resolution.



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.

- 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](#)

Description	Symbol	Max.	Unit
I2S serial 0 input clock frequency	$f_{\text{GCLK\_I2S\_0}}$	13	MHz
I2S serial 1 input clock frequency	$f_{\text{GCLK\_I2S\_1}}$	13	MHz

## 39.5 Power Consumption

The values in this section are measured values of power consumption under the following conditions, except where noted:

- Operating conditions
  - $V_{\text{DDIN}} = 3.3\text{ V}$
  - $V_{\text{DDIN}} = 2.7\text{ V}$ , CPU is running on Flash with 1 wait state
- Wake up time from sleep mode is measured from the edge of the wakeup signal to the execution of the first instruction fetched in flash.
- Oscillators
  - XOSC (crystal oscillator) stopped
  - XOSC32K (32 kHz crystal oscillator) running with external 32kHz crystal
  - DFLL48M using XOSC32K as reference and running at 48 MHz
- Clocks
  - DFLL48M used as main clock source, except otherwise specified
  - CPU, AHB clocks undivided
  - APBA clock divided by 4
  - APBB and APBC bridges off
- The following AHB module clocks are running: NVMCTRL, APBA bridge
  - All other AHB clocks stopped
- The following peripheral clocks running: PM, SYSCTRL, RTC
  - All other peripheral clocks stopped
- I/Os are inactive with internal pull-up
- CPU is running on flash with 1 wait states
- Cache enabled
- BOD33 disabled

**Table 39-6. Current Consumption**

Mode	Conditions	$T_A$	$V_{\text{CC}}$	Typ.	Max.	Unit
ACTIVE	CPU running a While(1) algorithm	25°C	3.3V	3.32	3.63	mA
		105°C	3.3V	3.57	3.98	
	CPU running a While(1) algorithm, with GCLKIN as reference	25°C	3.3V	$64 \times \text{Freq} + 110$	$70 \times \text{Freq} + 131$	$\mu\text{A}$ (with freq in MHz)
		105°C	3.3V	$65 \times \text{Freq} + 342$	$65 \times \text{Freq} + 764$	
	CPU running a Fibonacci algorithm	25°C	3.3V	4.03	4.35	mA
		105°C	3.3V	4.29	4.76	

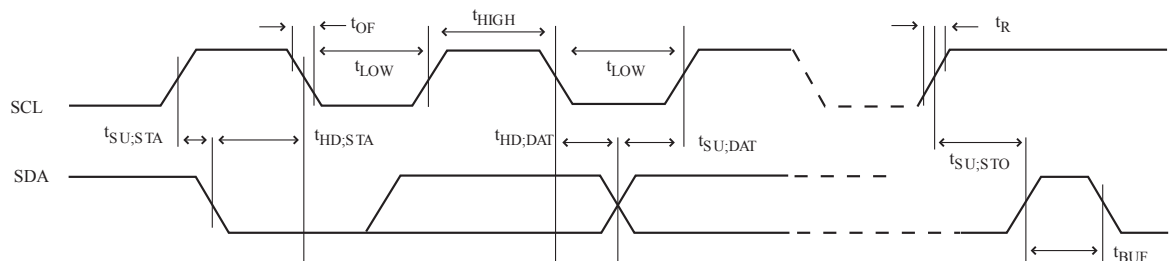
Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
$t_{SOSS}$	MISO setup after SS low	Slave	-	18	-	
$t_{SOSH}$	MISO hold after SS high	Slave	-	10	-	

1. These values are based on simulation. These values are not covered by test limits in production.
2. See [I/O Pin Characteristics](#).

### 39.14.3 SERCOM in I<sup>2</sup>C Mode Timing

This section describes the requirements for devices connected to the I<sup>2</sup>C Interface Bus.

**Figure 39-15. I<sup>2</sup>C Interface Bus Timing**



**Table 39-50. I<sup>2</sup>C Interface Timing**

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
$t_R$	Rise time for both SDA and SCL	Standard / Fast Mode	$C_b^{(2)} = 400\text{pF}$	-	230	350
		Fast Mode +	$C_b^{(2)} = 550\text{pF}$	60	100	
		High Speed Mode	$C_b^{(2)} = 100\text{pF}$	30	60	
$t_{OF}$	Output fall time from $V_{IHmin}$ to $V_{ILmax}$	Standard / Fast Mode	$10\text{pF} < C_b^{(2)} < 400\text{pF}$	25	50	
		Fast Mode +	$10\text{pF} < C_b^{(2)} < 550\text{pF}$	20	30	
		High Speed Mode	$10\text{pF} < C_b^{(2)} < 100\text{pF}$	10	20	
$t_{HD;STA}$	Hold time (repeated) START condition	$f_{SCL} > 100\text{kHz}$ , Master	$t_{LOW-9}$	-	-	
$t_{LOW}$	Low period of SCL Clock	$f_{SCL} > 100\text{kHz}$	113	-	-	

Name	Description	Mode	VDD=1.8V			VDD=3.3V			Units
			Min.	Typ.	Max.	Min.	Typ.	Max.	
t <sub>S_SDOH</sub>	Data output hold time	Slave transmitter	29.1			18.9			ns
t <sub>PDM2LS</sub>	Data input setup time	Master mode PDM2 Left	35.5			25.3			ns
t <sub>PDM2LH</sub>	Data input hold time	Master mode PDM2 Left	-8.2			-8.2			ns
t <sub>PDM2RS</sub>	Data input setup time	Master mode PDM2 Right	30.6			21.1			ns
t <sub>PDM2RH</sub>	Data input hold time	Master mode PDM2 Right	-7			-7			ns

1. All timing characteristics given for 15pF capacitive load.
2. These values are based on simulations and not covered by test limits in production.
3. See [I/O Pin Characteristics](#).