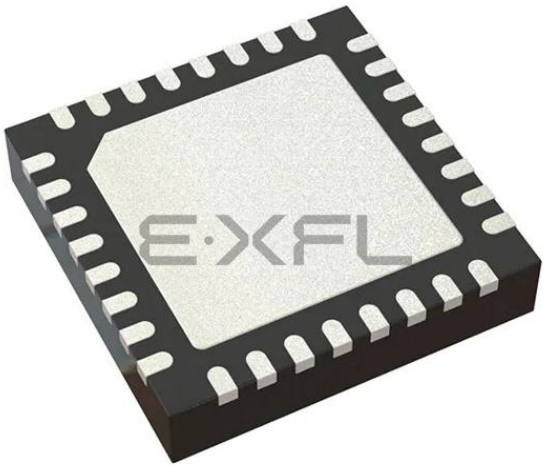


Welcome to E-XFL.COM

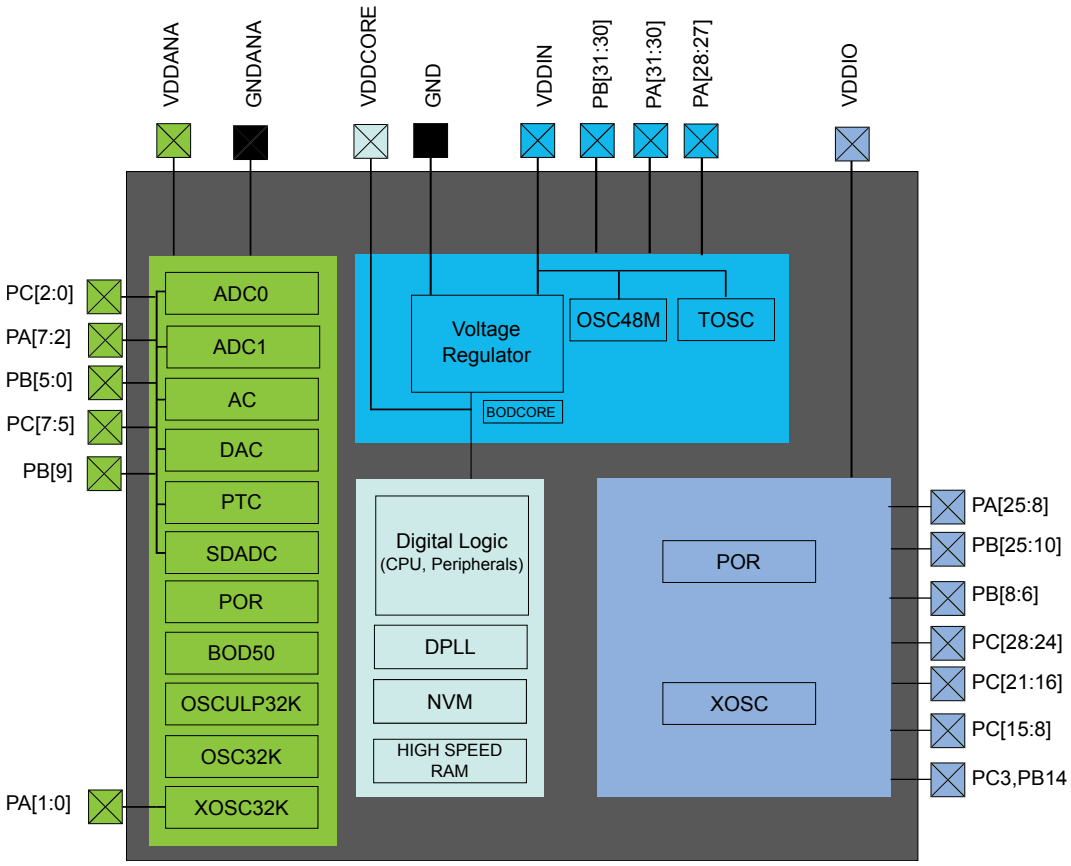
What is "Embedded - Microcontrollers"?

"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	CANbus, I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, WDT
Number of I/O	26
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 10x12b, 1x16b; D/A 1x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	32-VFQFN Exposed Pad
Supplier Device Package	32-VQFN (5x5)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsamc21e15a-mut

Figure 7-2. Power Domain Overview, SAM C20/C21 N



7.2 Power Supply Considerations

7.2.1 Power Supplies, SAM C21/SAM C20

The SAM C21 has the following power supply pins:

- VDDIO: Powers I/O lines and XOSC. Voltage is 2.70V to 5.50V.
- VDDIN: Powers I/O lines and the OSC48M, TOSC and internal regulator. Voltage is 2.70V to 5.50V.
- VDDANA: Powers I/O lines and the ADC, AC, DAC, PTC, SDADC, OSCULP32K, OSC32K, and XOSC32K. Voltage is 2.70V to 5.50V.
- VDDCORE: Internal regulated voltage output. Powers the core, memories, peripherals, and FDPLL96M. Voltage is 1.2V typical.

The same voltage must be applied to both the VDDIN and VDDANA pins. This common voltage is referred to as V_{DD} in the datasheet. VDDIO must always be less than or equal to VDDIN.

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.

The SAM C20 has the following power supply pins:

- VDDIO: Powers I/O lines and XOSC. Voltage is 2.70V to 5.50V.
- VDDIN: Powers I/O lines and the OSC48M, TOSC and internal regulator. Voltage is 2.70V to 5.50V.

11.7.7 Peripheral Interrupt Flag Status and Clear B

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 INTFLAGB 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 INTFLAGB interrupt flag.

Name: INTFLAGB
Offset: 0x18 [ID-00000a18]
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
				MTB	DMAC	NVMCTRL	DSU	PORT
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit 4 – MTB: Interrupt Flag for MTB

Bit 3 – DMAC: Interrupt Flag for DMAC

Bit 2 – NVMCTRL: Interrupt Flag for NVMCTRL

Bit 1 – DSU: Interrupt Flag for DSU

Bit 0 – PORT: Interrupt Flag for PORT

11.7.8 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.

Peripheral Name	Base Address	IRQ Line	AHB Clock		APB Clock		Generic Clock	PAC		Events		DMA	
			Index	Enabled at Reset	Index	Enabled at Reset	Index	Index	Prot at Reset	User	Generator	Index	Sleep Walking
SERCOM4	0x42001400	13			5	N	23: CORE 18: SLOW	5	N			10: RX 11: TX	Y
SERCOM5	0x42001800	14			6	N	25: CORE 24: SLOW	6	N			12: RX 13: TX	Y
CAN0	0x42001C00	15	8	N			26	7				14: DEBUG	N/A
CAN1	0x42002000	16	9	N			27	8				15: DEBUG	N/A
TCC0	0x42002400	17			9	N	28	9	N	9-10: EV0-1 11-14: MC0-3	34: OVF 35: TRG 36: CNT 37-40: MC0-3	16: OVF 17-20: MC0-3	Y
TCC1	0x42002800	18			10	N	28	10	N	15-16: EV0-1 17-18: MC0-1	41: OVF 42: TRG 43: CNT 44-45: MC0-1	21: OVF 22-23: MC0-1	Y
TCC2	0x42002C00	19			11	N	29	11	N	19-20: EV0-1 21-22: MC0-1	46: OVF 47: TRG 48: CNT 49-50: MC0-1	24: OVF 25-26: MC0-1	Y
TC0	0x42003000	20			12	N	30	12	N	23: EVU	51: OVF 52-53: MC0-1	27: OVF 28-29: MC0-1	Y
TC1	0x42003400	21			13	N	30	13	N	24: EVU	54: OVF 55-56: MC0-1	30: OVF 31-32: MC0-1	Y
TC2	0x42003800	22			14	N	31	14	N	25: EVU	57: OVF 58-59: MC0-1	33: OVF 34-35: MC0-1	Y
TC3	0x42003C00	23			15	N	31	15	N	26: EVU	60: OVF 61-62: MC0-1	36: OVF 37-38: MC0-1	Y
TC4	0x42004000	24			16	N	32	16	N	27: EVU	63: OVF 64-65: MC0-1	39: OVF 40-41: MC0-1	Y
ADC0	0x42004400	25			17	N	33	17	N	28: START 29: SYNC	66: RESRDY 67: WINMON	42: RESRDY	Y
ADC1	0x42004800	26			18	N	34	18	N	30: START 31: SYNC	68: RESRDY 69: WINMON	43: RESRDY	Y
SDADC	0x42004C00	29			19	N	35	19	N	32: START 33: FLUSH	70: RESRDY 71: WINMON	44: RESRDY	Y
AC	0x42005000	27			20	N	40	20	N	34-37: SOC0-3	72-75: COMP0-3 76-77: WIN0-1		Y
DAC	0x42005400	28			21	N	36	21	N	38: START	78: EMPTY	45: EMPTY	Y
PTC	0x42005800	30			22	N	37	22	N	39: STCONV	79: EOC 80: WCOMP	EOC: 46 WCOMP: 47 SEQ: 48	
CCL	0x42005C00				23	N	38	23	N	40-43 : LUTIN0-3	81-84: LUTOUT0-3		Y
AHB-APB Bridge D	0x43000000		13	Y	0								N/A
SERCOM6	0x43000000	9			0	N	41: CORE 18: SLOW	0	N			49: RX 50: TX	Y

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.

25.8.4 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

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

Related Links
SRAM Quality of Service

25.8.8 Software Trigger Control

Name: SWTRIGCTRL
Offset: 0x10
Reset: 0x00000000
Property: PAC Write-Protection

- 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 28-3. 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

28.6.5 PORT Access Priority

The PORT is accessed by different systems:

- 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:

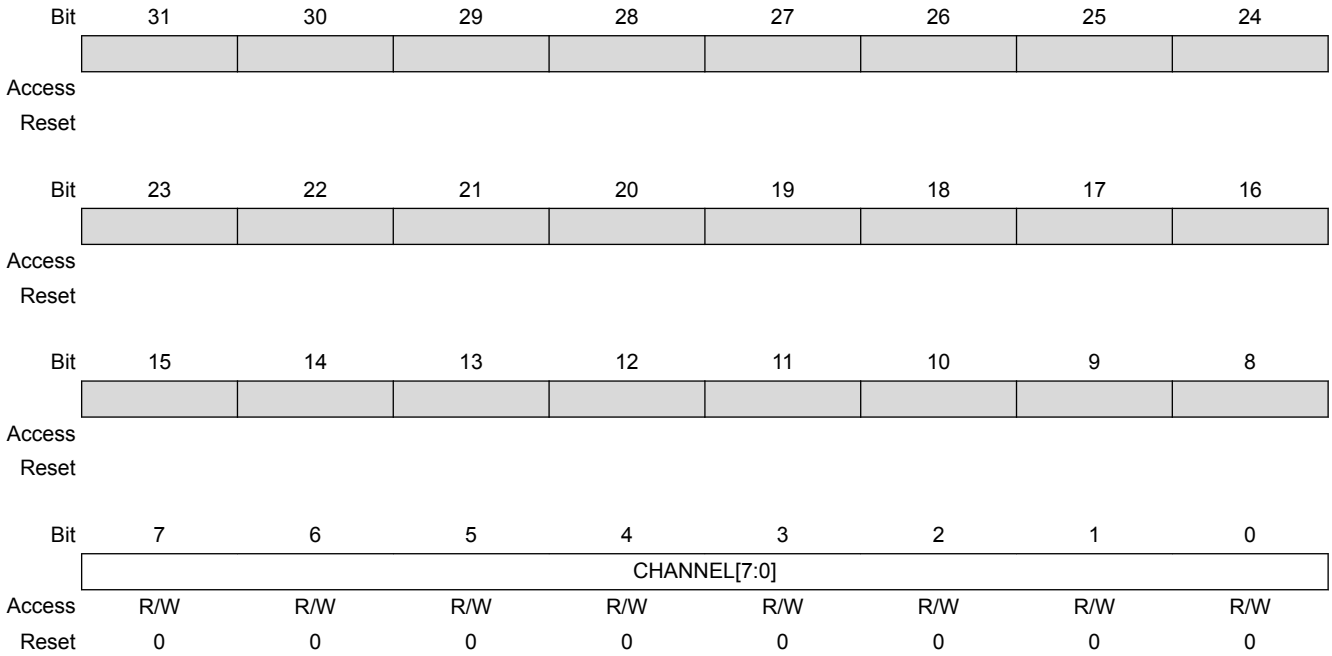
1. APB
2. EVSYS input events

For input events that require different actions on the same I/O pin, refer to Events.

Value	Event Generator	Description
0x58	TC5 OVF	Overflow/Underflow
0x59	TC5 MC0	Match/Capture 0
0x5A	TC5 MC1	Match/Capture 1
0x5B	TC6 OVF	Overflow/Underflow
0x5C	TC6 MC0	Match/Capture 0
0x5D	TC6 MC1	Match/Capture 1
0x5E	TC7 OVF	Overflow/Underflow
0x5F	TC7 MC0	Match/Capture 0
0x60	TC7 MC1	Match/Capture 1
0x61 - 0xFF	-	Reserved

29.8.8 Event User m

Name: USERm
Offset: 0x80+m*0x4 [m=0..460..46] [ID-0000120d]
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
	CHANNEL[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

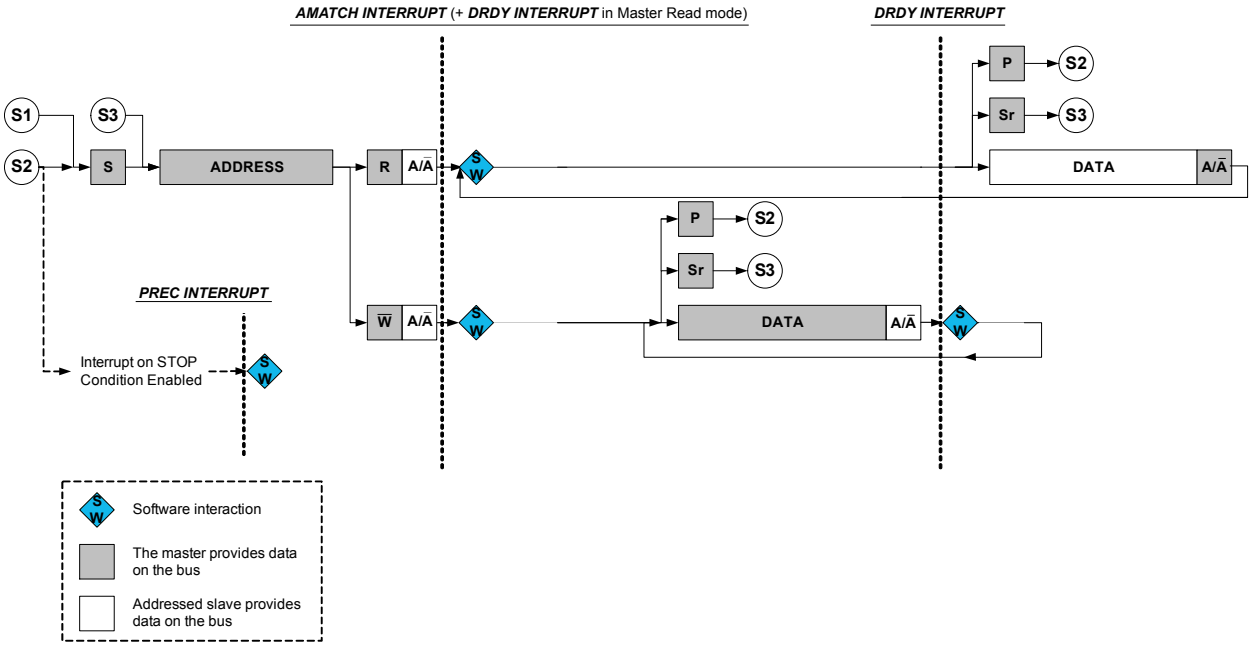
Bits 7:0 – CHANNEL[7:0]: Channel Event Selection
These bits are used to select the channel to connect to the event user.

Note that to select channel m, the value (m+1) must be written to the USER.CHANNEL bit group.

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 33-11. I²C Slave Behavioral Diagram (SCLSM=1)



Receiving Address Packets (SCLSM=0)

When CTRLA.SCLSM=0, the I2C slave stretches the SCL line according to Figure 33-10. 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

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.

33.10.2 Control B

Name: CTRLB
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

bus by waiting for the occurrence of a sequence of 11 consecutive ”recessive” bits (= Bus_Idle) before it can take part in bus activities and start the message transfer.

Access to the CAN configuration registers is only enabled when both bits CCCR.INIT and CCCR.CCE are set (protected write).

CCCR.CCE can only be set/reset while CCCR.INIT = ‘1’. CCCR.CCE is automatically reset when CCCR.INIT is reset.

The following registers are reset when CCCR.CCE is set

- HPMS - High Priority Message Status
- RXF0S - Rx FIFO 0 Status
- RXF1S - Rx FIFO 1 Status
- TXFQS - Tx FIFO/Queue Status
- TXBRP - Tx Buffer Request Pending
- TXBTO - Tx Buffer Transmission Occurred
- TXBCF - Tx Buffer Cancellation Finished
- TXEFS - Tx Event FIFO Status

The Timeout Counter value TOCV.TOC is preset to the value configured by TOCC.TOP when CCCR.CCE is set.

In addition the state machines of the Tx Handler and Rx Handler are held in idle state while CCCR.CCE = ‘1’.

The following registers are only writable while CCCR.CCE = ‘0’

- TXBAR - Tx Buffer Add Request
- TXBCR - Tx Buffer Cancellation Request

CCCR.TEST and CCCR.MON can only be set by the CPU while CCCR.INIT = ‘1’ and CCR.CCE = ‘1’. Both bits may be reset at any time. CCCR.DAR can only be set/reset while CCCR.INIT = ‘1’ and CCCR.CCE = ‘1’.

34.6.2.2 Normal Operation

Once the CAN is initialized and CCCR.INIT is reset to ‘0’, the CAN synchronizes itself to the CAN bus and is ready for communication.

After passing the acceptance filtering, received messages including Message ID and DLC are stored into a dedicated Rx Buffer or into Rx FIFO0 or Rx FIFO1.

For messages to be transmitted dedicated Tx Buffers and/or a Tx FIFO or a Tx Queue can be initialized or updated. Automated transmission on reception of remote frames is not implemented.

34.6.2.3 CAN FD Operation

There are two variants in the CAN FD frame format, first the CAN FD frame without bit rate switching where the data field of a CAN frame may be longer than 8 bytes. The second variant is the CAN FD frame where control field, data field, and CRC field of a CAN frame are transmitted with a higher bit rate than the beginning and the end of the frame.

The previously reserved bit in CAN frames with 11-bit identifiers and the first previously reserved bit in CAN frames with 29-bit identifiers will now be decoded as FDF bit. FDF = recessive signifies a CAN FD frame, FDF = dominant signifies a Classic CAN frame. In a CAN FD frame, the two bits following FDF, res and BRS, decide whether the bit rate inside of this CAN FD frame is switched. A CAN FD bit rate switch is signified by res = dominant and BRS = recessive. The coding of res = recessive is reserved for future expansion of the protocol. In case the CAN receives a frame with FDF = recessive and res = recessive, it

Queue Buffer the CPU has to write a ‘1’ to the corresponding bit position (=number of Tx Buffer) of register TXBCR. Transmit cancellation is not intended for Tx FIFO operation.

Successful cancellation is signaled by setting the corresponding bit of register TXBCF to ‘1’.

In case a transmit cancellation is requested while a transmission from a Tx Buffer is already ongoing, the corresponding TXBRP bit remains set as long as the transmission is in progress. If the transmission was successful, the corresponding TXBTO and TXBCF bits are set. If the transmission was not successful, it is not repeated and only the corresponding TXBCF bit is set.

Note: In case a pending transmission is canceled immediately before this transmission could have been started, there follows a short time window where no transmission is started even if another message is also pending in this node. This may enable another node to transmit a message which may have a lower priority than the second message in this node.

34.6.6.8 Tx Event Handling

To support Tx event handling the CAN has implemented a Tx Event FIFO. After the CAN has transmitted a message on the CAN bus, Message ID and timestamp are stored in a Tx Event FIFO element. To link a Tx event to a Tx Event FIFO element, the Message Marker from the transmitted Tx Buffer is copied into the Tx Event FIFO element.

The Tx Event FIFO can be configured to a maximum of 32 elements. The Tx Event FIFO element is described in Tx Event FIFO Element.

When a Tx Event FIFO full condition is signaled by IR.TEFF, no further elements are written to the Tx Event FIFO until at least one element has been read out and the Tx Event FIFO Get Index has been incremented. In case a Tx event occurs while the Tx Event FIFO is full, this event is discarded and interrupt flag IR.TEFL is set.

To avoid a Tx Event FIFO overflow, the Tx Event FIFO watermark can be used. When the Tx Event FIFO fill level reaches the Tx Event FIFO watermark configured by TXEFC.EFWM, interrupt flag IR.TEFW is set.

When reading from the Tx Event FIFO, two times the Tx Event FIFO Get Index TXEFS.EFGI has to be added to the Tx Event FIFO start address TXEFC.EFSA.

34.6.7 FIFO Acknowledge Handling

The Get Indexes of Rx FIFO 0, Rx FIFO 1 and the Tx Event FIFO are controlled by writing to the corresponding FIFO Acknowledge Index (refer to RXF0A, RXF1A and TXEFA). Writing to the FIFO Acknowledge Index will set the FIFO Get Index to the FIFO Acknowledge Index plus one and thereby updates the FIFO Fill Level. There are two use cases:

When only a single element has been read from the FIFO (the one being pointed to by the Get Index), this Get Index value is written to the FIFO Acknowledge Index.

When a sequence of elements has been read from the FIFO, it is sufficient to write the FIFO Acknowledge Index only once at the end of that read sequence (value: Index of the last element read), to update the FIFO’s Get Index.

Due to the fact that the CPU has free access to the CAN’s Message RAM, special care has to be taken when reading FIFO elements in an arbitrary order (Get Index not considered). This might be useful when reading a High Priority Message from one of the two Rx FIFOs. In this case the FIFO’s Acknowledge Index should not be written because this would set the Get Index to a wrong position and also alters the FIFO’s Fill Level. In this case some of the older FIFO elements would be lost.

Note: The application has to ensure that a valid value is written to the FIFO Acknowledge Index. The CAN does not check for erroneous values.

Bit 17 – MRAFL: Message RAM Access Failure Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 16 – TSWL: Timestamp Wraparound Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 15 – TEFLL: Tx Event FIFO Event Lost Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 14 – TEFFL: Tx Event FIFO Full Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 13 – TEFWL: Tx Event FIFO Watermark Reached Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 12 – TEFNL: Tx Event FIFO New Entry Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 11 – TFEL: Tx FIFO Empty Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 10 – TCFL: Transmission Cancellation Finished Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 9 – TCL: Transmission Completed Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit	31	30	29	28	27	26	25	24
			EFWM[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
			EFS[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	EFSA[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
	EFSA[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 29:24 – EFWM[5:0]: Event FIFO Watermark

Value	Description
0	Watermark interrupt disabled.
1 - 32	Level for Tx Event FIFO watermark interrupt (IR.TEFW).
>32	Watermark interrupt disabled.

Bits 21:16 – EFS[5:0]: Event FIFO Size
The Tx Event FIFO elements are indexed from 0 to EFS - 1.

Value	Description
0	Tx Event FIFO disabled
1 - 32	Number of Tx Event FIFO elements.
>32	Values greater than 32 are interpreted as 32.

Bits 15:0 – EFSA[15:0]: Event FIFO Start Address
Start address of Tx Event FIFO in Message RAM. When the CAN module addresses the Message RAM it addresses 32-bit words, not single bytes. The configurable start addresses are 32-bit word addresses, i.e. only bits 15 to 2 are evaluated, the two least significant bits are ignored. Bits 1 to 0 will always be read back as “00”.

34.8.46 Tx Event FIFO Status

Name: TXEFS
Offset: 0xF4 [ID-0000a4bb]
Reset: 0x00000000
Property: Read-only

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 any update of the registers with value of its buffered register is performed on hardware UPDATE condition. Locking the update ensures that all buffer registers are valid before an hardware 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 clear the LUPD bit.

Value	Description
0	The CCBUFx and PERBUF buffer registers value are copied into CCx and PER registers on hardware update condition.
1	The CCBUFx and PERBUF buffer registers value are not copied into CCx and PER registers on hardware update condition.

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.

35.7.1.14 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.

35.7.1.15 Channel x Compare/Capture Value, 8-bit Mode

Name: CCx
Offset: 0x1C + x*0x01 [x=0..1]
Reset: 0x00
Property: Write-Synchronized, Read-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.

35.7.1.16 Period Buffer Value, 8-bit Mode

Name: PERBUF
Offset: 0x2F
Reset: 0xFF
Property: Write-Synchronized

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.

The Channel x Compare/Capture Buffer Value (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 force update command (CTRLBSET.CMD=0x3, UPDATE). For further details, refer to Double Buffering. 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.

The table below shows the update counter and overflow event/interrupt generation conditions in different operation modes.

Table 36-2. 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

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	WINUT[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
	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 23:0 – WINUT[23:0]: Window Upper Threshold
If the window monitor is enabled, these bits define the upper threshold value.

39.8.14 Offset Correction

Name: OFFSETCORR
Offset: 0x14 [ID-0000243d]
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized

44.7 Register Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0							ENABLE	SWRST
0x01	CTRLB	7:0								START
0x02	CFGA	7:0	REFNUM[7:0]							
0x03		15:8	DIVREF							
0x04 ... 0x07	Reserved									
0x08	INTENCLR	7:0								DONE
0x09	INTENSET	7:0								DONE
0x0A	INTFLAG	7:0								DONE
0x0B	STATUS	7:0							OVF	BUSY
0x0C	SYNCBUSY	7:0							ENABLE	SWRST
0x0D		15:8								
0x0E		23:16								
0x0F		31:24								
0x10	VALUE	7:0	VALUE[7:0]							
0x11		15:8	VALUE[15:8]							
0x12		23:16	VALUE[23:16]							
0x13		31:24								

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 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.

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.

44.8.1 Control A

Name: CTRLA
Offset: 0x00 [ID-00000e03]
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Symbol	Condition	Max.	Units
$f_{GCLKGEN3}$	Divided	66	MHz
$f_{GCLKGEN4}$			
$f_{GCLKGEN5}$			
$f_{GCLKGEN6}$			
$f_{GCLKGEN7}$			
$f_{GCLKGEN8}$			

Table 45-8. Maximum Peripheral Clock Frequencies

Symbol	Description	Max.	Units
f_{CPU}	CPU clock frequency	48	MHz
f_{AHB}	AHB clock frequency	48	MHz
f_{APBA}	APBA clock frequency	48	MHz
f_{APBB}	APBB clock frequency	48	MHz
f_{APBC}	APBC clock frequency	48	MHz
f_{APBD}	APBD clock frequency	48	MHz
f_{GCLK_DPLL}	FDPLL96M Reference clock frequency	2	MHz
$f_{GCLK\_DPLL\_32K}$	FDPLL96M 32k Reference clock frequency	32	kHz
f_{GCLK_EIC}	EIC input clock frequency	48	MHz
$f_{GCLK\_FREQM\_MSR}$	FREQM Measure	96	MHz
$f_{GCLK\_FREQM\_REF}$	FREQM Reference	48	MHz
f_{GCLK_TSENS}	TSENS input clock frequency	48	MHz
$f_{GCLK\_EVSYS\_CHANNEL\_0}$	EVSYS channel 0 input clock frequency	48	MHz
$f_{GCLK\_EVSYS\_CHANNEL\_1}$	EVSYS channel 1 input clock frequency	48	MHz
$f_{GCLK\_EVSYS\_CHANNEL\_2}$	EVSYS channel 2 input clock frequency	48	MHz
$f_{GCLK\_EVSYS\_CHANNEL\_3}$	EVSYS channel 3 input clock frequency	48	MHz
$f_{GCLK\_EVSYS\_CHANNEL\_4}$	EVSYS channel 4 input clock frequency	48	MHz
$f_{GCLK\_EVSYS\_CHANNEL\_5}$	EVSYS channel 5 input clock frequency	48	MHz
$f_{GCLK\_EVSYS\_CHANNEL\_6}$	EVSYS channel 6 input clock frequency	48	MHz
$f_{GCLK\_EVSYS\_CHANNEL\_7}$	EVSYS channel 7 input clock frequency	48	MHz
$f_{GCLK\_EVSYS\_CHANNEL\_8}$	EVSYS channel 8 input clock frequency	48	MHz
$f_{GCLK\_EVSYS\_CHANNEL\_9}$	EVSYS channel 9 input clock frequency	48	MHz
$f_{GCLK\_EVSYS\_CHANNEL\_10}$	EVSYS channel 10 input clock frequency	48	MHz
$f_{GCLK\_EVSYS\_CHANNEL\_11}$	EVSYS channel 11 input clock frequency	48	MHz