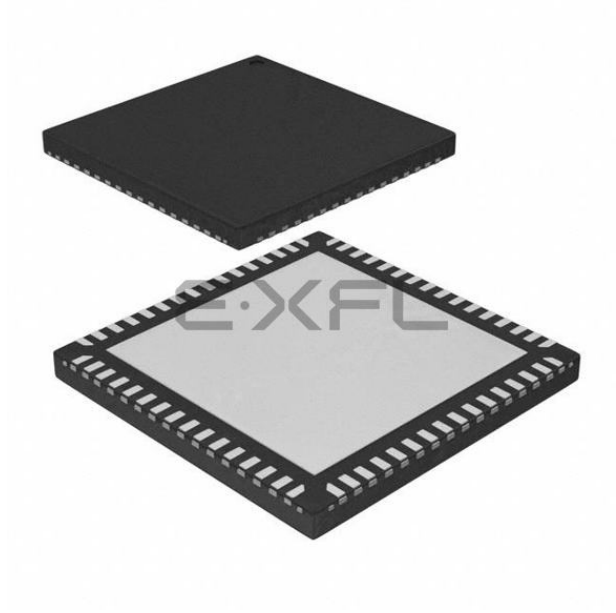


Welcome to E-XFL.COM

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
Peripherals	Brown-out Detect/Reset, DMA, POR, 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 ~ 5.5V
Data Converters	A/D 12x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	64-VFQFN Exposed Pad
Supplier Device Package	64-QFN (9x9)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsamc20j16a-mut

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

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

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

11.5.2.6 Write Access Protection Management Errors

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

PAC also offers an additional safety feature for correct program execution with an interrupt generated on double write clear protection or double write set protection. If a peripheral is write protected and a subsequent set protection operation is detected then the PAC returns an error, and similarly for a double clear protection operation.

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

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

11.5.2.7 AHB Slave Bus Errors

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

11.5.2.8 Generating Events

The PAC module can also generate an event when any of the Interrupt Flag registers bit are set. To enable the PAC event generation, the control bit EVCTRL.ERREO must be set a '1'.

11.5.3 DMA Operation

Not applicable.

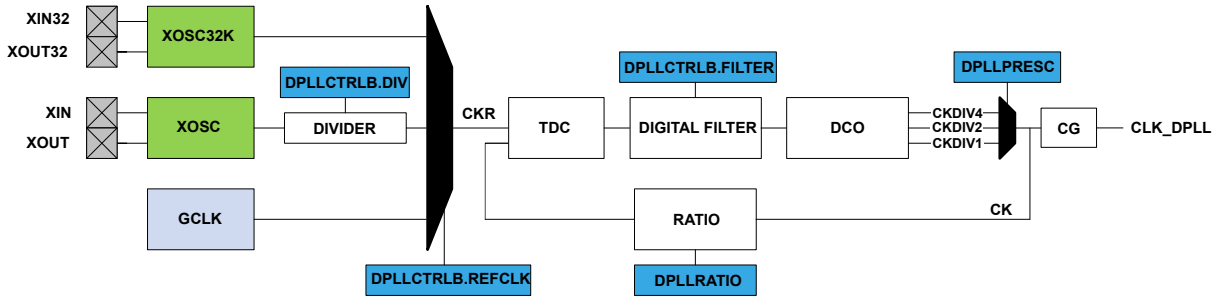
11.5.4 Interrupts

The PAC has the following interrupt source:

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

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAGAHB and INTFLAGn) registers is set when the interrupt condition occurs. Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register. An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared,

Figure 20-2. DPLL Block Diagram



When the controller is disabled, the output clock is low. If the Loop Divider Ratio Fractional part bit field in the DPLL Ratio register (DPLLRATIO.LDRFRAC) is zero, the DPLL works in integer mode. Otherwise, the fractional mode is activated. Note that the fractional part has a negative impact on the jitter of the DPLL.

Example (integer mode only): assuming F_{CKR} = 32kHz and F_{CK} = 48MHz, the multiplication ratio is 1500. It means that LDR shall be set to 1499.

Example (fractional mode): assuming F_{CKR} = 32kHz and F_{CK} = 48.006MHz, the multiplication ratio is 1500.1875 (1500 + 3/16). Thus LDR is set to 1499 and LDRFRAC to 3.

Related Links

GCLK - Generic Clock Controller

OSC32KCTRL – 32KHz Oscillators Controller

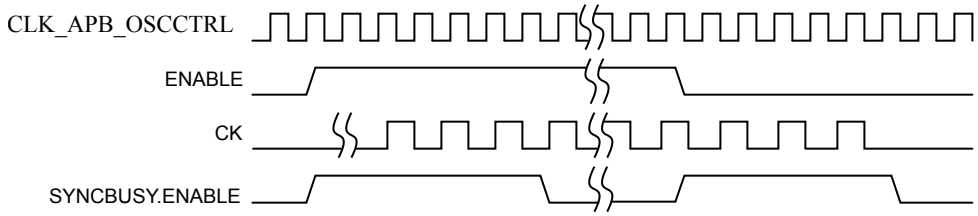
20.6.5.1 Basic Operation

Initialization, Enabling, Disabling, and Resetting

The DPLLC is enabled by writing a '1' to the Enable bit in the DPLL Control A register (DPLLCTRLA.ENABLE). The DPLLC is disabled by writing a zero to this bit.

The DPLLSYNCBUSY.ENABLE is set when the DPLLCTRLA.ENABLE bit is modified. It is cleared when the DPLL output clock CK has sampled the bit at the high level after enabling the DPLL. When disabling the DPLL, DPLLSYNCBUSY.ENABLE is cleared when the output clock is no longer running.

Figure 20-3. Enable Synchronization Busy Operation

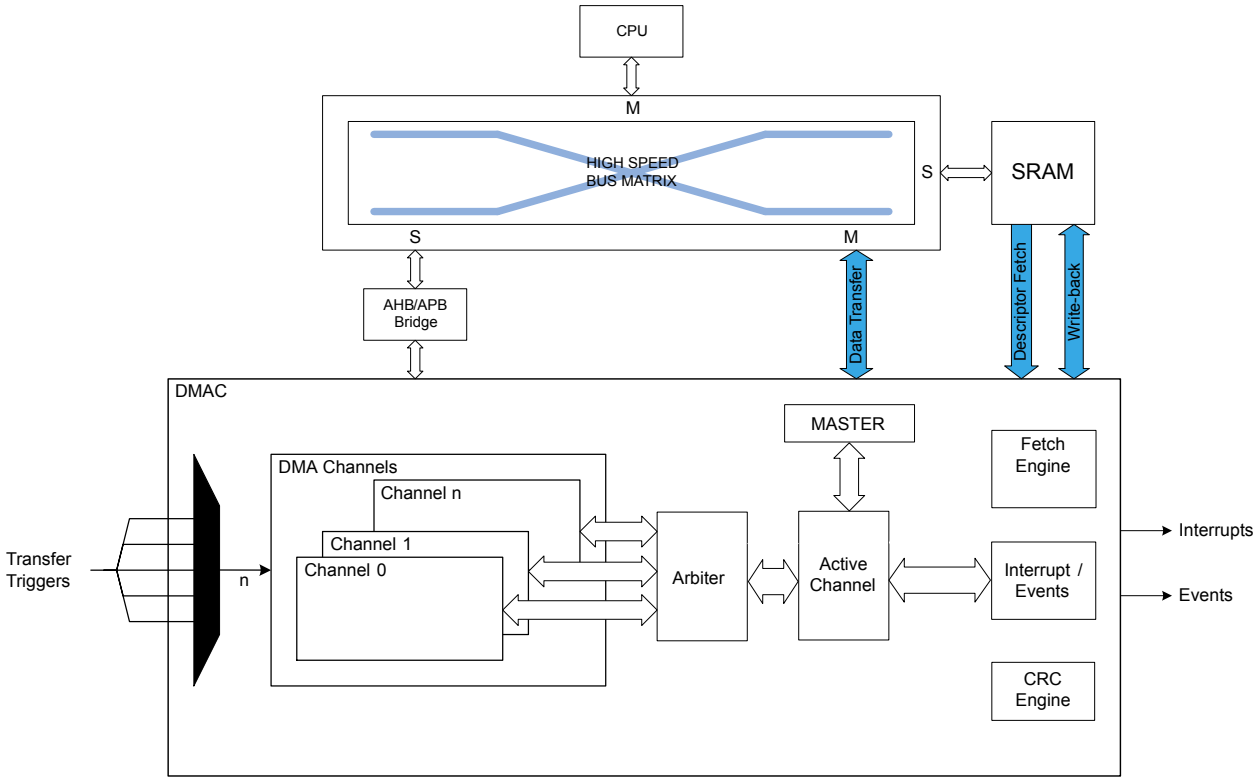


The frequency of the DPLL output clock CK is stable when the module is enabled and when the Lock bit in the DPLL Status register is set (DPLLSTATUS.LOCK).

When the Lock Time bit field in the DPLL Control B register (DPLLCTRLB.LTIME) is non-zero, a user defined lock time is used to validate the lock operation. In this case the lock time is constant. If DPLLCTRLB.LTIME=0, the lock signal is linked with the status bit of the DPLL, and the lock time varies depending on the filter selection and the final target frequency.

25.3 Block Diagram

Figure 25-1. DMAC Block Diagram



25.4 Signal Description

Not applicable.

25.5 Product Dependencies

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

25.5.1 I/O Lines

Not applicable.

25.5.2 Power Management

The DMAC will continue to operate in any sleep mode where the selected source clock is running. The DMAC's interrupts can be used to wake up the device from sleep modes. Events connected to the event system can trigger other operations in the system without exiting sleep modes. On hardware or software reset, all registers are set to their reset value.

Related Links

PM – Power Manager

25.5.3 Clocks

The DMAC bus clock (CLK_DMAC_APB) must be configured and enabled in the Main Clock module before using the DMAC.

Bit	31	30	29	28	27	26	25	24
	DIR[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DIR[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DIR[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DIR[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DIR[31:0]: Port Data Direction
These bits set the data direction for the individual I/O pins in the PORT group.

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.

28.9.2 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.

Tip: The I/O pins are assembled in pin groups (”PORT groups”) with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

Name: DIRCLR
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection

31.6.3.10 Sample Adjustment

In asynchronous mode (CTRLA.CMODE=0), three samples in the middle are used to determine the value based on majority voting. The three samples used for voting can be selected using the Sample Adjustment bit field in Control A register (CTRLA.SAMPA). When CTRLA.SAMPA=0, samples 7-8-9 are used for 16x oversampling, and samples 3-4-5 are used for 8x oversampling.

31.6.4 DMA, Interrupts and Events

31.6.4.1 DMA Operation

The USART generates the following DMA requests:

31.6.4.2 Interrupts

The USART has the following interrupt sources. These are asynchronous interrupts, and can wake up the device from any sleep mode:

- Data Register Empty (DRE)
- Receive Complete (RXC)
- Transmit Complete (TXC)
- Receive Start (RXS)
- Clear to Send Input Change (CTSIC)
- Received Break (RXBRK)
- Error (ERROR)

Each interrupt source has its own interrupt flag. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) will be set when the interrupt condition is met. Each interrupt can be individually enabled by writing '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR).

An interrupt request is generated when the interrupt flag is set and if the corresponding interrupt is enabled. The interrupt request remains active until either the interrupt flag is cleared, the interrupt is disabled, or the USART is reset. For details on clearing interrupt flags, refer to the INTFLAG register description.

The USART has one common interrupt request line for all the interrupt sources. The value of INTFLAG indicates which interrupt is executed. Note that interrupts must be globally enabled for interrupt requests. Refer to *Nested Vector Interrupt Controller* for details.

Related Links

Nested Vector Interrupt Controller

31.6.4.3 Events

Not applicable.

31.6.5 Sleep Mode Operation

The behavior in sleep mode is depending on the clock source and the Run In Standby bit in the Control A register (CTRLA.RUNSTDBY):

- Internal clocking, CTRLA.RUNSTDBY=1: GCLK_SERCOMx_CORE can be enabled in all sleep modes. Any interrupt can wake up the device.
- External clocking, CTRLA.RUNSTDBY=1: The Receive Start and the Receive Complete interrupt(s) can wake up the device.
- Internal clocking, CTRLA.RUNSTDBY=0: Internal clock will be disabled, after any ongoing transfer was completed. The Receive Start and the Receive Complete interrupt(s) can wake up the device.

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
					HDRDLY[1:0]		BRKLEN[1:0]	
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

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

Bits 11:10 – HDRDLY[1:0]: LIN Master Header Delay

These bits define the delay between break and sync transmission in addition to the delay between the sync and identifier (ID) fields when in LIN master mode (CTRLA.FORM=0x2).
This field is only valid when using the LIN header command (CTRLB.LINCMD=0x2).

Value	Description
0x0	Delay between break and sync transmission is 1 bit time. Delay between sync and ID transmission is 1 bit time.
0x1	Delay between break and sync transmission is 4 bit time. Delay between sync and ID transmission is 4 bit time.
0x2	Delay between break and sync transmission is 8 bit time. Delay between sync and ID transmission is 4 bit time.
0x3	Delay between break and sync transmission is 14 bit time. Delay between sync and ID transmission is 4 bit time.

Bits 9:8 – BRKLEN[1:0]: LIN Master Break Length

These bits define the length of the break field transmitted when in LIN master mode (CTRLA.FORM=0x2).

Value	Description
0x0	Break field transmission is 13 bit times
0x1	Break field transmission is 17 bit times
0x2	Break field transmission is 21 bit times
0x3	Break field transmission is 26 bit times

Bits 2:0 – GTIME[2:0]: Guard Time

These bits define the guard time when using RS485 mode (CTRLA.TXPO=0x3).

For RS485 mode, the guard time is programmable from 0-7 bit times and defines the time that the transmit enable pin (TE) remains high after the last stop bit is transmitted and there is no remaining data to be transmitted.

31.8.4 Baud

Name: BAUD
Offset: 0x0C [ID-00000fa7]
Reset: 0x0000
Property: Enable-Protected, PAC Write-Protection

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

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

Bits 15:0 – BAUD[15:0]: Baud Value
Arithmetic Baud Rate Generation (`CTRLA.SAMPR[0]=0`):

These bits control the clock generation, as described in the SERCOM Baud Rate section.

If Fractional Baud Rate Generation (`CTRLA.SAMPR[0]=1`) bit positions 15 to 13 are replaced by FP[2:0] Fractional Part:

- **Bits 15:13 - FP[2:0]: Fractional Part**

  These bits control the clock generation, as described in the *SERCOM Clock Generation – Baud-Rate Generator* section.
- **Bits 12:0 - BAUD[21:0]: Baud Value**

  These bits control the clock generation, as described in the *SERCOM Clock Generation – Baud-Rate Generator* section.

Related Links
Clock Generation – Baud-Rate Generator
Asynchronous Arithmetic Mode BAUD Value Selection

31.8.5 Receive Pulse Length Register

Name: RXPL
Offset: 0x0E [ID-00000fa7]
Reset: 0x00
Property: PAC Write-Protection

- Interrupt Flag Clear and Status register (INTFLAG)
- Status register (STATUS)
- Data register (DATA)

Optional PAC Write-Protection is denoted by the "PAC Write-Protection" property in each individual register description.

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

Related Links

PAC - Peripheral Access Controller

32.5.9 Analog Connections

Not applicable.

32.6 Functional Description

32.6.1 Principle of Operation

The SPI is a high-speed synchronous data transfer interface It allows high-speed communication between the device and peripheral devices.

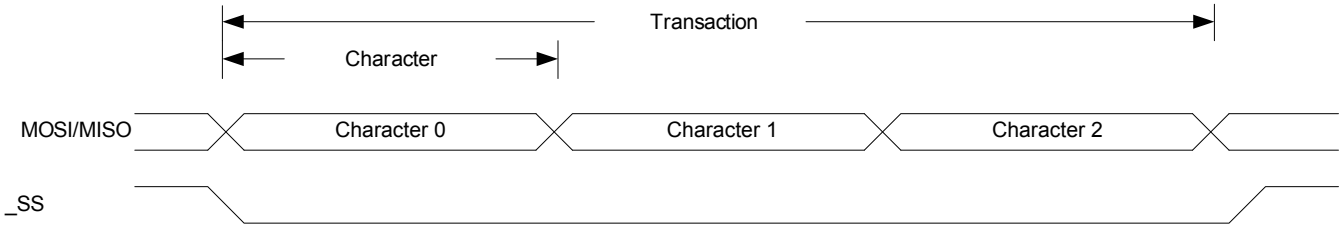
The SPI can operate as master or slave. As master, the SPI initiates and controls all data transactions. The SPI is single buffered for transmitting and double buffered for receiving.

When transmitting data, the Data register can be loaded with the next character to be transmitted during the current transmission.

When receiving, the data is transferred to the two-level receive buffer, and the receiver is ready for a new character.

The SPI transaction format is shown in SPI Transaction Format. Each transaction can contain one or more characters. The character size is configurable, and can be either 8 or 9 bits.

Figure 32-2. SPI Transaction Format



The SPI master must pull the slave select line ($\overline{SS}$) of the desired slave low to initiate a transaction. The master and slave prepare data to send via their respective shift registers, and the master generates the serial clock on the SCK line.

Data are always shifted from master to slave on the Master Output Slave Input line (MOSI); data is shifted from slave to master on the Master Input Slave Output line (MISO).

Each time character is shifted out from the master, a character will be shifted out from the slave simultaneously. To signal the end of a transaction, the master will pull the $\overline{SS}$ line high

32.6.2 Basic Operation

32.6.2.1 Initialization

The following registers are enable-protected, meaning that they can only be written when the SPI is disabled (CTRL.ENABLE=0):

Value	Description
0	Time-out disabled.
1	Time-out enabled.

Bits 29:28 – INACTOUT[1:0]: Inactive Time-Out
If the inactive bus time-out is enabled and the bus is inactive for longer than the time-out setting, the bus state logic will be set to idle. An inactive bus arise when either an I^2C master or slave is holding the SCL low.

Enabling this option is necessary for SMBus compatibility, but can also be used in a non-SMBus set-up.

Calculated time-out periods are based on a 100kHz baud rate.

These bits are not synchronized.

Value	Name	Description
0x0	DIS	Disabled
0x1	55US	5-6 SCL cycle time-out (50-60µs)
0x2	105US	10-11 SCL cycle time-out (100-110µs)
0x3	205US	20-21 SCL cycle time-out (200-210µs)

Bit 27 – SCLSM: SCL Clock Stretch Mode
This bit controls when SCL will be stretched for software interaction.

This bit is not synchronized.

Value	Description
0	SCL stretch according to Figure 33-5.
1	SCL stretch only after ACK bit, Figure 33-6.

Bits 25:24 – SPEED[1:0]: Transfer Speed
These bits define bus speed.

These bits are not synchronized.

Value	Description
0x0	Standard-mode (Sm) up to 100 kHz and Fast-mode (Fm) up to 400 kHz
0x1	Fast-mode Plus (Fm+) up to 1 MHz
0x2	High-speed mode (Hs-mode) up to 3.4 MHz
0x3	Reserved

Bit 23 – SEXTTOEN: Slave SCL Low Extend Time-Out
This bit enables the slave SCL low extend time-out. If SCL is cumulatively held low for greater than 25ms from the initial START to a STOP, the master will release its clock hold if enabled, and complete the current transaction. A STOP will automatically be transmitted.

SB or MB will be set as normal, but CLKHOLD will be release. The MEXTTOUT and BUSERR status bits will be set.

This bit is not synchronized.

Value	Description
0	Time-out disabled
1	Time-out enabled

34.8.5 Test

Name: TEST
Offset: 0x10 [ID-0000a4bb]
Reset: 0x00000000
Property: Read-only, Write-restricted

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
	RX	TX[1:0]		LBCK				
Access	R	R/W	R/W	R/W				
Reset	0	0	0	0				

Bit 7 – RX: Receive Pin
Monitors the actual value of pin CAN_RX

Value	Description
0	The CAN bus is dominant (CAN_RX = 0).
1	The CAN bus is recessive (CAN_RX = 1).

Bits 6:5 – TX[1:0]: Control of Transmit Pin
This field defines the control of the transmit pin.

Value	Name	Description
0x0	CORE	Reset value, CAN_TX controlled by CAN core, updated at the end of CAN bit time.
0x1	SAMPLE	Sample Point can be monitored at pin CAN_TX.
0x2	DOMINANT	Dominant (‘0’) level at pin CAN_TX.
0x3	RECESSIVE	Recessive (‘1’) level at pin CAN_TX.

Bit 4 – LBCK: Loop Back Mode

Value	Description
0	Loop Back Mode is disabled.
1	Loop Back Mode is enabled.

Bit 0 – OVF: Overflow Interrupt Flag
This flag is set on the next CLK_TC_CNT cycle after an overflow condition occurs, and will generate an interrupt request if INTENCLR.OVF or INTENSET.OVF is '1'.

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

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

35.7.2.8 Status

Name: STATUS
Offset: 0x0B
Reset: 0x01
Property: Read-Synchronized

Bit	7	6	5	4	3	2	1	0
				CCBUFVx	PERBUFV		SLAVE	STOP
Access				R/W	R/W		R	R
Reset				0	0		0	1

Bit 4 – CCBUFVx: Channel x Compare or Capture Buffer Valid
For a compare channel x, the bit x is set when a new value is written to the corresponding CCBUFx register.

The bit x is cleared by writing a '1' to it when CTRLB.LUPD is set, or it is cleared automatically by hardware on UPDATE condition.

For a capture channel x, the bit x is set when a valid capture value is stored in the CCBUFx register. The bit x is cleared automatically when the CCx register is read.

Bit 3 – PERBUFV: Period Buffer Valid
This bit is set when a new value is written to the PERBUF register. The bit is cleared by writing '1' to the corresponding location when CTRLB.LUPD is set, or automatically cleared by hardware on UPDATE condition. This bit is available only in 8-bit mode and will always read zero in 16- and 32-bit modes.

Bit 1 – SLAVE: Slave Status Flag
This bit is only available in 32-bit mode on the slave TC (i.e., TC1 and/or TC3). The bit is set when the associated master TC (TC0 and TC2, respectively) is set to run in 32-bit mode.

Bit 0 – STOP: Stop Status Flag
This bit is set when the TC is disabled, on a Stop command, or on an overflow/underflow condition when the One-Shot bit in the Control B Set register (CTRLBSET.ONESHOT) is '1'.

Value	Description
0	Counter is running.
1	Counter is stopped.

35.7.2.9 Waveform Generation Control

counting. When the module is enabled, the counter operation starts when the event is received or when a re-trigger software command is applied.

Note:
When a start event action is configured in the Event Action bits in the Event Control register (EVCTRL.EVACT0=0x3, START), enabling the counter will not start the counter. The counter will start on the next incoming event, but it will not restart on subsequent events.

Count Event Action

The TCC can count events. When an event is received, the counter increases or decreases the value, depending on direction settings (CTRLBSET.DIR or CTRLBCLR.DIR).

The count event action is selected by the Event Action 0 bit group in the Event Control register (EVCTRL.EVACT0=0x5, COUNT).

Direction Event Action

The direction event action can be selected in the Event Control register (EVCTRL.EVACT1=0x2, DIR). When this event is used, the asynchronous event path specified in the event system must be configured or selected. The direction event action can be used to control the direction of the counter operation, depending on external events level. When received, the event level overrides the Direction settings (CTRLBSET.DIR or CTRLBCLR.DIR) and the direction bit value is updated accordingly.

Increment Event Action

The increment event action can be selected in the Event Control register (EVCTRL.EVACT0=0x4, INC) and can change the counter state when an event is received. When the TCE0 event (TCCx_EV0) is received, the counter increments, whatever the direction setting (CTRLBSET.DIR or CTRLBCLR.DIR) is.

Decrement Event Action

The decrement event action can be selected in the Event Control register (EVCTRL.EVACT1=0x4, DEC) and can change the counter state when an event is received. When the TCE1 (TCCx_EV1) event is received, the counter decrements, whatever the direction setting (CTRLBSET.DIR or CTRLBCLR.DIR) is.

Non-recoverable Fault Event Action

Non-recoverable fault actions can be selected in the Event Control register (EVCTRL.EVACTn=0x7, FAULT). When received, the counter will be stopped and the output of the compare channels is overridden according to the Driver Control register settings (DRVCTRL.NREx and DRVCTRL.NRVx). TCE0 and TCE1 must be configured as asynchronous events.

Event Action Off

If the event action is disabled (EVCTRL.EVACTn=0x0, OFF), enabling the counter will also start the counter.

Related Links

One-Shot Operation

36.6.2.5 Compare Operations

By default, the Compare/Capture channel is configured for compare operations. To perform capture operations, it must be re-configured.

When using the TCC with the Compare/Capture Value registers (CCx) for compare operations, the counter value is continuously compared to the values in the CCx registers. This can be used for timer or for waveform operation.

The figure after that ('Waveform Generation with Fault Qualification, Halt, and Restart Actions') shows a similar example, but with additionally enabled fault qualification. Here, counting is resumed after the fault condition is no longer present.

Note that in RAMP2 and RAMP2A operations, when a new timer/counter cycle starts, the cycle index will automatically change.

Figure 36-30. Waveform Generation with Halt and Restart Actions

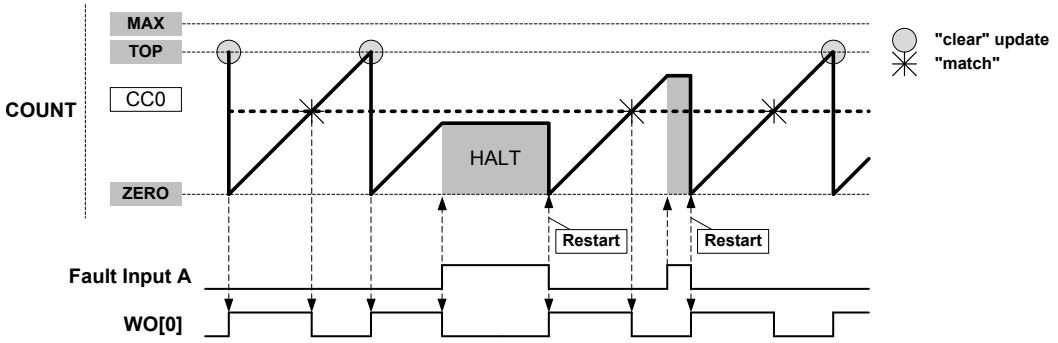
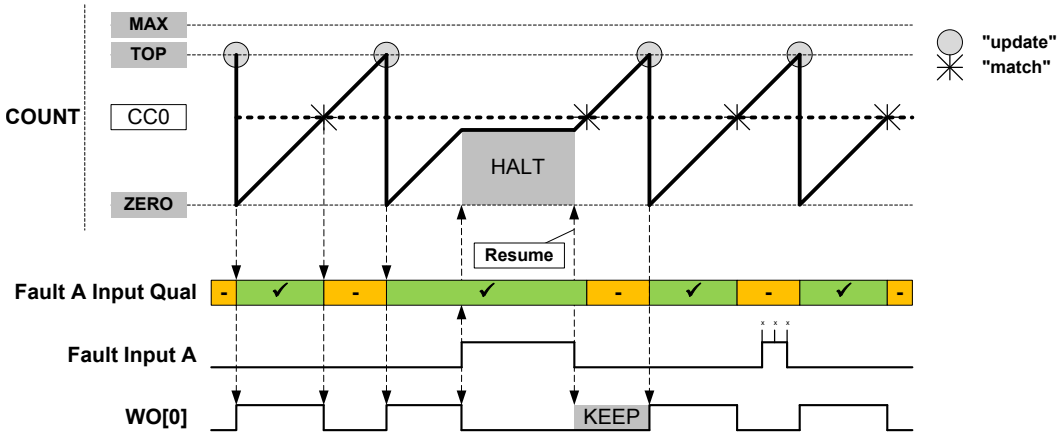


Figure 36-31. Waveform Generation with Fault Qualification, Halt, and Restart Actions



Software Halt Action

This is configured by writing 0x2 to the Fault n Halt mode bits in the Recoverable Fault n configuration register (FCTRLn.HALT). Software halt action is similar to hardware halt action, but in order to restart the timer/counter, the corresponding fault condition must not be present anymore, and the corresponding FAULT n bit in the STATUS register must be cleared by software.

Value	Description
0	The Non-Recoverable Fault x interrupt is disabled.
1	The Non-Recoverable Fault x interrupt is enabled.

Bit 13 – FAULTB: Recoverable Fault B Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Recoverable Fault B Interrupt Disable/Enable bit, which disables the Recoverable Fault B interrupt.

Value	Description
0	The Recoverable Fault B interrupt is disabled.
1	The Recoverable Fault B interrupt is enabled.

Bit 12 – FAULTA: Recoverable Fault A Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Recoverable Fault A Interrupt Disable/Enable bit, which disables the Recoverable Fault A interrupt.

Value	Description
0	The Recoverable Fault A interrupt is disabled.
1	The Recoverable Fault A interrupt is enabled.

Bit 11 – DFS: Non-Recoverable Debug Fault Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Debug Fault State Interrupt Disable/Enable bit, which disables the Debug Fault State interrupt.

Value	Description
0	The Debug Fault State interrupt is disabled.
1	The Debug Fault State interrupt is enabled.

Bit 10 – UFS: Non-Recoverable Update Fault Interrupt Enable
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Non-Recoverable Update Fault Interrupt Disable/Enable bit, which disables the Non-Recoverable Update Fault interrupt.

Value	Description
0	The Non-Recoverable Update Fault interrupt is disabled.
1	The Non-Recoverable Update Fault interrupt is enabled.

Bit 3 – ERR: Error Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Error Interrupt Disable/Enable bit, which disables the Compare interrupt.

Value	Description
0	The Error interrupt is disabled.
1	The Error interrupt is enabled.

Writing a '1' to this bit clears the Recoverable Fault B interrupt flag.

Bit 11 – DFS: Non-Recoverable Debug Fault State Interrupt Flag
This flag is set on the next CLK_TCC_COUNT cycle after an Debug Fault State occurs.

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

Writing a '1' to this bit clears the Debug Fault State interrupt flag.

Bit 10 – UFS: Non-Recoverable Update Fault
This flag is set when the RAMP index changes and the Lock Update bit is set (CTRLBSET.LUPD).

Writing a zero to this bit has no effect.

Writing a one to this bit clears the Non-Recoverable Update Fault interrupt flag.

Bit 3 – ERR: Error Interrupt Flag
This flag is set if a new capture occurs on a channel when the corresponding Match or Capture Channel x interrupt flag is one. In which case there is nowhere to store the new capture.

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

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

Bit 2 – CNT: Counter Interrupt Flag
This flag is set on the next CLK_TCC_COUNT cycle after a counter event occurs.

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

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

Bit 1 – TRG: Retrigger Interrupt Flag
This flag is set on the next CLK_TCC_COUNT cycle after a counter retrigger occurs.

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

Writing a '1' to this bit clears the re-trigger interrupt flag.

Bit 0 – OVF: Overflow Interrupt Flag
This flag is set on the next CLK_TCC_COUNT cycle after an overflow condition occurs.

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

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

36.8.13 Status

Name: STATUS
Offset: 0x30 [ID-00002e48]
Reset: 0x00000001
Property: -

Value	Name	Description
0x18	GND	Internal ground
0x19 - 0x1F	-	Reserved

Bits 4:0 – MUXPOS[4:0]: Positive MUX Input Selection
These bits define the MUX selection for the positive ADC input. If the internal bandgap voltage input channel is selected, then the Sampling Time Length bit group in the Sampling Control register must be written with a corresponding value.

Value	Name	Description
0x00	AIN0	ADC AIN0 pin
0x01	AIN1	ADC AIN1 pin
0x02	AIN2	ADC AIN2 pin
0x03	AIN3	ADC AIN3 pin
0x04	AIN4	ADC AIN4 pin
0x05	AIN5	ADC AIN5 pin
0x06	AIN6	ADC AIN6 pin
0x07	AIN7	ADC AIN7 pin
0x08	AIN8	ADC AIN8 pin
0x09	AIN9	ADC AIN9 pin
0x0A	AIN10	ADC AIN10 pin
0x0B	AIN11	ADC AIN11 pin
0xC - 0x17	-	Reserved
0x18	-	Reserved
0x19	BANDGAP	Bandgap Voltage
0x1C	DAC	DAC Output
0x1E	-	Reserved
0x1F	-	Reserved

38.8.10 Control C

Name: CTRLC
Offset: 0x0A [ID-0000120e]
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
	REFSEL[1:0]		DITHER		VPD	LEFTADJ	IOEN	EOEN
Access	R/W	R/W	R/W		R/W	R/W	R/W	R/W
Reset	0	0	0		0	0	0	0

Bits 7:6 – REFSEL[1:0]: Reference Selection
This bit field selects the Reference Voltage for the DAC.

Value	Name	Description
0x0	INTREF	Internal voltage reference
0x1	VDDANA	Analog voltage supply
0x2	VREFA	External reference
0x3		Reserved

Bit 5 – DITHER: Dithering Mode
This bit controls dithering operation according to Dithering mode.

Value	Description
0	Dithering mode is disabled.
1	Dithering mode is enabled.

Bit 3 – VPD: Voltage Pump Disabled
This bit controls the behavior of the voltage pump.

Value	Description
0	Voltage pump is turned on/off automatically
1	Voltage pump is disabled.

Bit 2 – LEFTADJ: Left-Adjusted Data
This bit controls how the 10-bit conversion data is adjusted in the Data and Data Buffer registers.

Value	Description
0	DATA and DATABUF registers are right-adjusted.
1	DATA and DATABUF registers are left-adjusted.

Bit 1 – IOEN: Internal Output Enable

Value	Description
0	Internal DAC output not enabled.
1	Internal DAC output enabled to be used by the AC or ADC.

Bit 0 – EOEN: External Output Enable

Value	Description
0	The DAC output is turned off.
1	The high-drive output buffer drives the DAC output to the V_{OUT} pin.

41.8.3 Event Control

Name: INTFLAG
Offset: 0x06 [ID-00001f13]
Reset: 0x00
Property: –

Bit	7	6	5	4	3	2	1	0
					OVF	WINMON	OVERRUN	RESRDY
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 3 – OVF: Overflow
This flag is cleared by writing a one to the flag.

This flag is set when the conversion result requires more than 24 bits and overflows the VALUE register, and an interrupt request will be generated if INTENCLR/SET.OVF is one.

Writing a zero to this bit has no effect.

Writing a one to this bit clears the Overflow interrupt flag.

Bit 2 – WINMON: Window Monitor
This flag is cleared by writing a one to the flag or by reading the VALUE register.

This flag is set on the next cycle after a match with the window monitor condition, and an interrupt request will be generated if INTENCLR/SET.WINMON is one.

Writing a zero to this bit has no effect.

Writing a one to this bit clears the Window Monitor interrupt flag.

Bit 1 – OVERRUN: Overrun
This flag is cleared by writing a one to the flag.

This flag is set if a valid VALUE is updated before the previous valid value has been read by the CPU, and an interrupt will be generated if INTENCLR/SET.OVERRUN is one.

Writing a zero to this bit has no effect.

Writing a one to this bit clears the Overrun interrupt flag.

Bit 0 – RESRDY: Result Ready
This flag is cleared by writing a one to the flag or by reading the VALUE register.

This flag is set when the conversion result is available, and an interrupt will be generated if INTENCLR/SET.RESRDY is one.

This flag will not set if an overflow occurs during the conversion.

Writing a zero to this bit has no effect.

Writing a one to this bit clears the Result Ready interrupt flag.

43.8.8 Status

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