25.12.15. Timestamp Value

Name: TIMESTAMP
Offset: 0x64
Reset: 0
Property: -

Note: Once the value from TIMESTAMP register is read, INTFLAG.TAMPER bit must be cleared. The next value from this register should be read-only after the INTFLAG.TAMPER bit is set again.

Bit	31	30	29	28	27	26	25	24
	YEAR[5:0]						MONTH[3:2]	
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	MONTH[1:0]		DAY[4:0]					HOUR[4]
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	HOUR[3:0]				MINUTE[5:2]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

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

Bits 31:26 – YEAR[5:0] Year
The year value is captured by the TIMESTAMP when a tamper condition occurs.

Bits 25:22 – MONTH[3:0] Month
The month value is captured by the TIMESTAMP when a tamper condition occurs.

Bits 21:17 – DAY[4:0] Day
The day value is captured by the TIMESTAMP when a tamper condition occurs.

Bits 16:12 – HOUR[4:0] Hour
The hour value is captured by the TIMESTAMP when a tamper condition occurs.

Bits 11:6 – MINUTE[5:0] Minute
The minute value is captured by the TIMESTAMP when a tamper condition occurs.

Bits 5:0 – SECOND[5:0] Second
The second value is captured by the TIMESTAMP when a tamper condition occurs.

© 2025 Microchip Technology Inc. and its subsidiaries

25.12.16. Tamper ID

Name: TAMPID
Offset: 0x68
Reset: 0x00000000

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

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
					TAMPID3	TAMPID2	TAMPID1	TAMPID0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 31 – TAMPEVT Tamper Event Detected
Writing a '0' to this bit has no effect. Writing a '1' to this bit clears the tamper detection bit.

Value	Description
0	A tamper input event has not been detected
1	A tamper input event has been detected

Bits 0, 1, 2, 3 – TAMPIDn Tamper on Channel n Detected [n=0..3]
Writing a '0' to this bit has no effect. Writing a '1' to this bit clears the tamper detection bit.

Value	Description
0	A tamper condition has not been detected on Channel n
1	A tamper condition has been detected on Channel n

© 2025 Microchip Technology Inc. and its subsidiaries

25.12.17. Backup0

Name: BKUP0
Offset: 0x80
Reset: 0x00000000

Bit	31	30	29	28	27	26	25	24
	BKUP[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
	BKUP[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
	BKUP[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
	BKUP[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 – BKUP[31:0] Backup
These bits are user-defined for general purpose use in the Backup domain.

© 2025 Microchip Technology Inc. and its subsidiaries

26. Direct Memory Access Controller (DMAC)

26.1. Overview

The Direct Memory Access Controller (DMAC) contains both a Direct Memory Access engine and a Cyclic Redundancy Check (CRC) engine. The DMAC can transfer data between memories and peripherals, and, thus, off-load these tasks from the CPU. It enables high data transfer rates with minimum CPU intervention, and frees up CPU time. With access to all peripherals, the DMAC can handle automatic transfer of data between communication modules.

The DMA part of the DMAC has several DMA channels that can all receive different types of transfer triggers to generate transfer requests from the DMA channels to the arbiter (see *DMAC Block Diagram* in the *Block Diagram* from Related Links). The arbiter grants one DMA channel at a time to act as the active channel. When an active channel is granted, the fetch engine of the DMAC will fetch a transfer descriptor from the SRAM and store it in the internal memory of the active channel, which executes the data transmission.

An ongoing data transfer of an active channel can be interrupted by a higher prioritized DMA channel. The DMAC writes back the updated transfer descriptor from the internal memory of the active channel to SRAM, and grants the higher prioritized channel to start the transfer as the new active channel. When a DMA channel is done with its transfer, interrupts and events can be generated optionally.

The DMAC has four bus interfaces:

- The Data Transfer bus is used for performing the actual DMA transfer.
- The AHB/APB Bridge bus is used when writing and reading the I/O registers of the DMAC.
- The Descriptor Fetch bus is used by the fetch engine to fetch transfer descriptors before the data transfer can be started or continued.
- The Write-Back bus is used to write the transfer descriptor back to SRAM.

All buses are AHB Manager interfaces except for the AHB/APB Bridge bus, which is an APB Subordinate interface.

Burst transfer options, buffered active channel to pre-fetch descriptors and advance quality of service features ensure low-latency transfers for high-speed peripherals or high-speed operations.

The CRC engine can be used by software to detect an accidental error in the transferred data and to take corrective action, such as requesting the data to be sent again or simply not using the incorrect data.

Related Links
Block Diagram

26.2. Features

- Data Transfer From:
  - Peripheral to peripheral
  - Peripheral to memory
  - Memory to peripheral
  - Memory to memory
- Transfer Trigger Sources
  - Software
  - Events from event system
  - Dedicated requests from peripherals

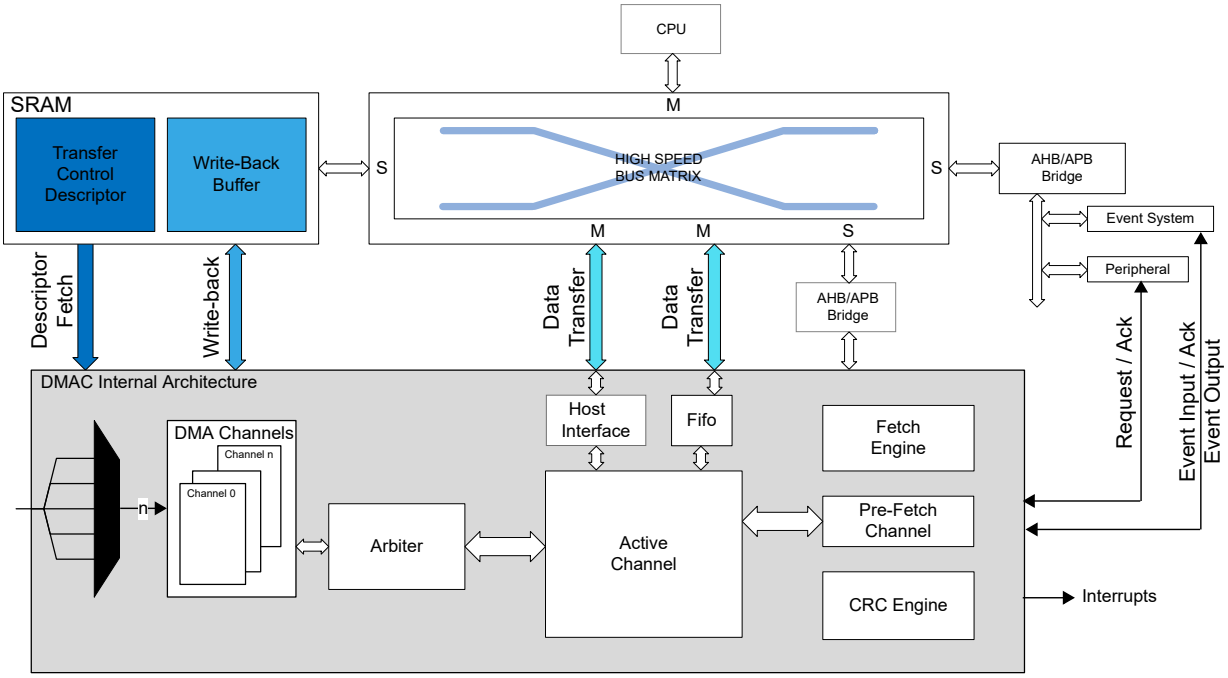
© 2025 Microchip Technology Inc. and its subsidiaries

- SRAM-Based Transfer Descriptors
  - Single transfer using one descriptor
  - Multi-buffer or circular buffer modes by linking multiple descriptors
- Up to 16 Channels
  - Enable 16 independent transfers
  - Automatic descriptor fetch for each channel
  - Suspend/resume operation support for each channel
- Flexible Arbitration Scheme
  - Four configurable priority levels for each channel
  - Fixed or round-robin priority scheme within each priority level
- From 1 to 256 KB Data Transfer in a Single Block Transfer
- Multiple Addressing Modes
  - Static
  - Configurable increment scheme
- Optional Interrupt Generation
  - On block transfer complete
  - On error detection
  - On channel suspend
- Eight Event Inputs
  - One event input for each of the eight least significant DMA channels
  - Can be selected to trigger normal transfers, periodic transfers or conditional transfers
  - Can be selected to suspend or resume channel operation
- Four Event Outputs
  - One output event for each of the four least significant DMA channels
  - Selectable generation on AHB, block or transaction transfer complete
- Error Management Supported by Write-Back Function
  - Dedicated Write-Back memory section for each channel to store ongoing descriptor transfer
- CRC Polynomial Software Selectable to
  - CRC-16 (CRC-CCITT)
  - CRC-32 (IEEE 802.3)

© 2025 Microchip Technology Inc. and its subsidiaries

26.3. Block Diagram

Figure 26-1. DMAC Block Diagram



26.4. Signal Description

Not applicable.

26.5. Product Dependencies

The following sections describe how the other parts of the system must be configured correctly to use this peripheral.

26.5.1. I/O Lines

Not applicable.

26.5.2. Power Management

The DMAC continues to operate in any sleep modes (Idle, Standby Sleep) where the selected source clock is running. The DMAC's interrupts can be used to wake up the device from the sleep modes. Events connected to the event system can trigger other operations in the system without exiting the sleep modes. On a hardware or software Reset, all registers are set to their Reset value.

26.5.3. Clocks

An AHB clock (CLK_DMAC_AHB) is required to clock the DMAC. This clock can be configured in the Main Clock peripheral (MCLK) before using the DMAC, and the default state of CLK_DMAC_AHB can be found in the MCLK.AHBMASK register.

26.5.4. DMA

Not applicable.

26.5.5. Interrupts

The interrupt request line is connected to the interrupt controller. Using the DMAC interrupt requires the interrupt controller to be configured first. See *Nested Vector Interrupt Controller (NVIC)* from Related Links.

© 2025 Microchip Technology Inc. and its subsidiaries

Related Links
Nested Vector Interrupt Controller (NVIC)

26.5.6. Events

The events are connected to the event system. See *Event System (EVSYS)* from Related Links.

Related Links
Event System (EVSYS)

26.5.7. Debug Operation

When the CPU is halted in Debug mode, the DMAC halts normal operation. The DMAC can be forced to continue operation during debugging. See *DBGCTRL* from Related Links.

Related Links
DBGCTRL
Debug Control

26.5.8. 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 Pending register (INTPEND)
- Channel ID register (CHID)
- Channel Interrupt Flag Status and Clear register (CHINTFLAG)

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

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

26.5.9. Analog Connections

Not applicable.

26.6. Functional Description

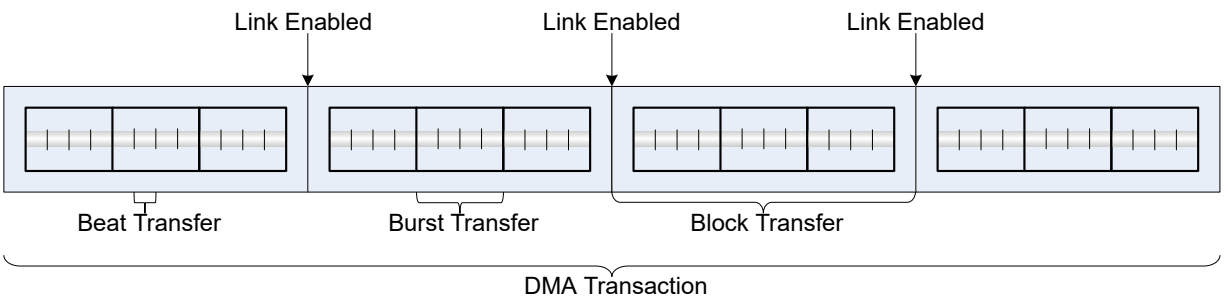
26.6.1. Principle of Operation

The DMAC consists of a DMA module and a CRC module.

26.6.1.1. DMA

The DMAC can transfer data between memories and peripherals without interaction from the CPU. The data transferred by the DMAC are called transactions, and these transactions can be split into smaller data transfers. The following figure illustrates the relationship between the different transfer sizes:

Figure 26-2. DMA Transfer Sizes



© 2025 Microchip Technology Inc. and its subsidiaries

- Beat transfer – The size of one data transfer bus access, and the size is selected by writing the Beat Size bit group in the Block Transfer Control register (BTCTRL.BEATSIZE).
- Burst transfer: Defined as n beat transfers, where n will differ from one device family to another. A burst transfer is atomic, cannot be interrupted and the length of the burst is selected by writing the Burst Length bit group in each Channel n Control A register (CHCTRLA.BURSTLEN).
- Block transfer – The amount of data one transfer descriptor can transfer, and the amount can range from 1 to 64k beats. A block transfer can be interrupted, in contrast to the burst transfer.
- Transaction – The DMAC can link several transfer descriptors by having the first descriptor pointing to the second and so forth, as shown in Figure 26-2. A DMA transaction is the complete transfer of all blocks within a linked list.

A transfer descriptor describes how a block transfer must be carried out by the DMAC, and it must remain in SRAM (see *Transfer Descriptors* from Related Links).

The Figure 26-2 illustrates several block transfers linked together, which are called linked descriptors (see *Linked Descriptors* from Related Links).

A DMA transfer is initiated by an incoming transfer trigger on one of the DMA channels. This trigger can be configured to be either a software trigger, an event trigger or one of the dedicated peripheral triggers. The transfer trigger results in a DMA transfer request from the specific channel to the arbiter. If there are several DMA channels with pending transfer requests, the arbiter chooses which channel is granted access to become the active channel. The DMA channel granted access as the active channel will carry out the transaction as configured in the transfer descriptor. A current transaction can be interrupted by a higher prioritized channel after each burst transfer, but resumes the block transfer when the related DMA channel is granted access as the active channel again.

For each beat transfer, an optional output event can be generated. For each block transfer, optional interrupts and an optional output event can be generated. When a transaction is completed, depending on the configuration, the DMA channel will either be suspended or disabled.

Related Links

Transfer Descriptors
Linked Descriptors

26.6.1.2. CRC

The internal CRC engine supports two commonly used CRC polynomials: CRC-16 (CRC-CCITT) and CRC-32 (IEEE 802.3). It can be used on a selectable DMA channel, or on the I/O interface. Refer to CRC Operation for details.

26.6.2. Basic Operation

26.6.2.1. Initialization

DMAC Initialization

Before DMAC is enabled, it must be configured as defined below:

- The SRAM address of where the descriptor memory section is located must be written to the Description Base Address (BASEADDR) register.
- The SRAM address of where the write-back section must be located must be written to the Write-Back Memory Base Address (WRBADDR) register.
- Priority level x of the arbiter can be enabled by setting the Priority Level x Enable bit in the Control register (CTRL.LVLENx=1)

DMA Channel Initialization

Before a DMA channel is enabled, the DMA channel and the corresponding first transfer descriptor must be configured, as defined below:

© 2025 Microchip Technology Inc. and its subsidiaries

- DMA Channel Configuration:
  - The channel number of the DMA channel to configure must be written to the Channel Control A (CHCTRLA) register.
  - Trigger action must be selected by writing the Trigger Action bit field in the Channel Control A (CHCTRLA.TRIGACT) register.
  - Trigger source must be selected by writing the Trigger Source bit field in the Channel Control A (CHCTRLA.TRIGSRC) register.
- Transfer Descriptor
  - The size of each access of the data transfer bus must be selected by writing the Beat Size bit group in the Block Transfer Control (BTCTRL.BEATSIZE) register.
  - The transfer descriptor must be made valid by writing a one to the Valid bit in the Block Transfer Control (BTCTRL.VALID) register.
  - Number of beats in the block transfer must be selected by writing the Block Transfer Count (BTCNT) register.
  - Source address for the block transfer must be selected by writing the Block Transfer Source Address (SRCADDR) register.
  - Destination address for the block transfer must be selected by writing the Block Transfer Destination Address (DSTADDR) register.

CRC Calculation

If CRC calculation is needed, the CRC engine must be configured before it is enabled, as described below:

- The CRC input source must selected by writing the CRC Input Source bit group in the CRC Control (CRCCTRL.CRCSRC) register.
- The type of CRC calculation must be selected by writing the CRC Polynomial Type bit group in the CRC Control (CRCCTRL.CRCPOLY) register.
- If I/O is selected as input source, the beat size must be selected by writing the CRC Beat Size bit group in the CRC Control (CRCCTRL.CRCBEATSIZE) register.

Register Properties

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

- The Descriptor Base Memory Address (BASEADDR) register
- The Write-Back Memory Base Address (WRBADDR) register

The following DMAC bit is enable-protected, that is, it can only be written when the DMAC and CRC are disabled (CTRL.DMAENABLE=0 and CRCCTRL.CRCSRC=0):

- The Software Reset bit in the Control (CTRL.SWRST) register

The following DMA channel bit is enable-protected, meaning that it can only be written when the corresponding DMA channel is disabled:

- The Channel Software Reset bit in the Channel Control A (CHCTRLA.SWRST) register

The following CRC registers are enable-protected, that is, they can only be written when the CRC is disabled (CRCCTRL.CRCSRC=0):

- The CRC Control (CRCCTRL) register
- CRC Checksum (CRCCHKSUM) register

Enable-protection is denoted by the 'Enable-Protected' property in the register description.

© 2025 Microchip Technology Inc. and its subsidiaries

26.6.2.2. Enabling, Disabling, and Resetting

The DMAC is enabled by writing the DMA Enable bit in the Control register (CTRL.DMAENABLE) to '1'. The DMAC is disabled by writing a '0' to the CTRL.DMAENABLE bit.

A DMA channel is enabled by writing the Enable bit in the Channel Control A register (CHCTRLA.ENABLE) to '1', after the corresponding channel ID to the channel is configured. A DMA channel is disabled by writing a '0' to CHCTRLAn.ENABLE.

The CRC is enabled by writing a '1' to the CRC Enable bit in the Control register (CTRL.CRCENABLE). The CRC is disabled by writing a '0' to CTRL.CRCENABLE.

The DMAC is reset by writing a '1' to the Software Reset bit in the Control register (CTRL.SWRST) while the DMAC and CRC are disabled. All registers in the DMAC except DBGCTRL will be reset to their initial state.

A DMA channel is reset by writing a '1' to the Software Reset bit in the Channel Control A register (CHCTRLAn.SWRST), after writing the corresponding channel ID to the Channel ID bit group in the Channel ID register (CHID.ID). The channel registers will be reset to their initial state. The corresponding DMA channel must be disabled in order for the Reset to take effect.

26.6.2.3. Transfer Descriptors

The transfer descriptors, together with the channel configurations, decide how a block transfer must be executed. Before a DMA channel is enabled (CHCTRLA.ENABLE is written to one) and receives a transfer trigger, its first transfer descriptor must be initialized and valid (BTCTRL.VALID). The first transfer descriptor describes the first block transfer of a transaction.

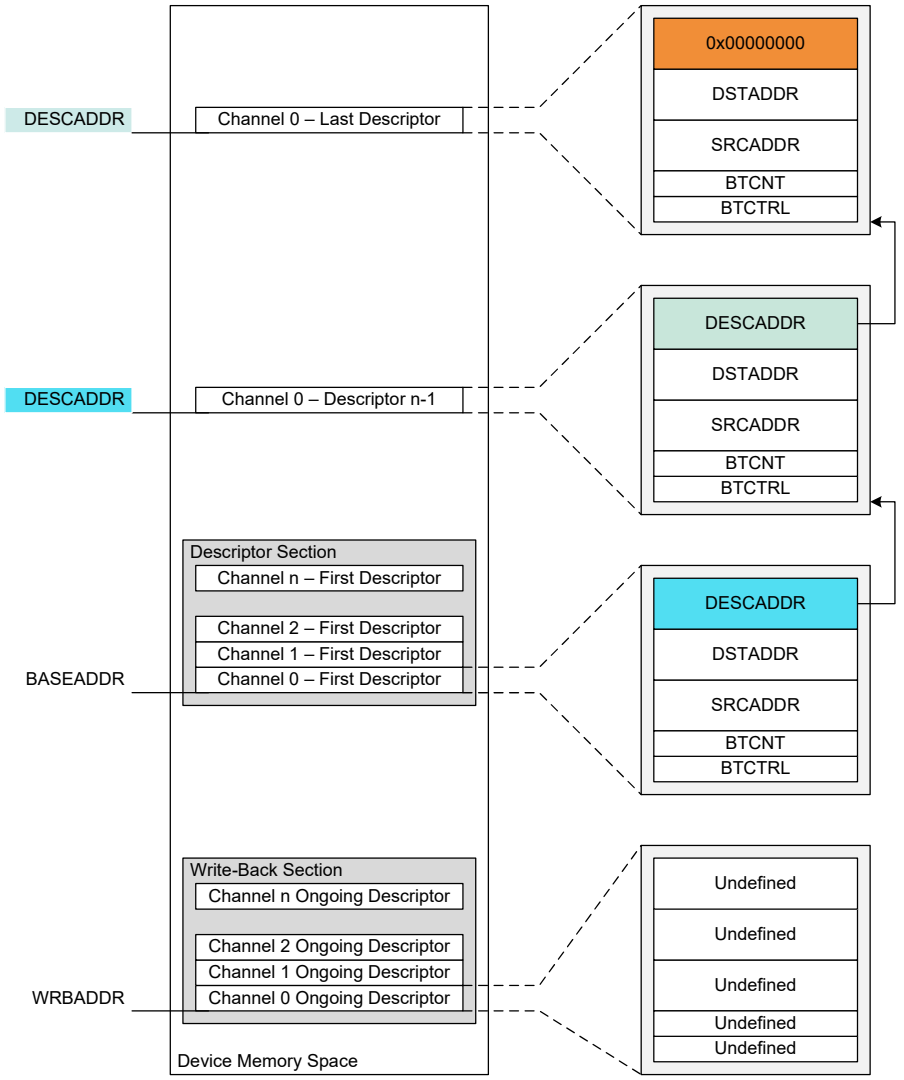
All transfer descriptors must reside in SRAM. The addresses stored in the Descriptor Memory Section Base Address (BASEADDR) and Write-Back Memory Section Base Address (WRBADDR) registers tell the DMAC where to find the descriptor memory section and the write-back memory section.

The descriptor memory section is where the DMAC expects to find the first transfer descriptors for all DMA channels. As BASEADDR points only to the first transfer descriptor of channel '0' (see the following figure). All first transfer descriptors must be stored in a contiguous memory section, where the transfer descriptors must be ordered according to their channel number (see *Linked Descriptors* from Related Links).

The write-back memory section is where the DMAC stores the transfer descriptors for the ongoing block transfers. WRBADDR points to the ongoing transfer descriptor of channel '0'. All ongoing transfer descriptors are stored in a contiguous memory section where the transfer descriptors are ordered according to their channel number. The figure below shows an example of linked descriptors on DMA channel '0' (see *Linked Descriptors* from Related Links).

© 2025 Microchip Technology Inc. and its subsidiaries

Figure 26-3. Memory Sections



The size of the descriptor and write-back memory sections are dependent on the number of the most significant enabled DMA channel *m*, as shown below:

$Size = 128\text{bits} \cdot (m + 1)$

For memory optimization, it is recommended to use the less significant DMA channels, if not all channels are required.

The descriptor and write-back memory sections can either be two separate memory sections, or they can share a memory section (BASEADDR=WRBADDR). The benefit of having them in two separate sections, is that the same transaction for a channel can be repeated without having to modify the first transfer descriptor.

Related Links

Linked Descriptors

© 2025 Microchip Technology Inc. and its subsidiaries

26.6.2.4. Arbitration

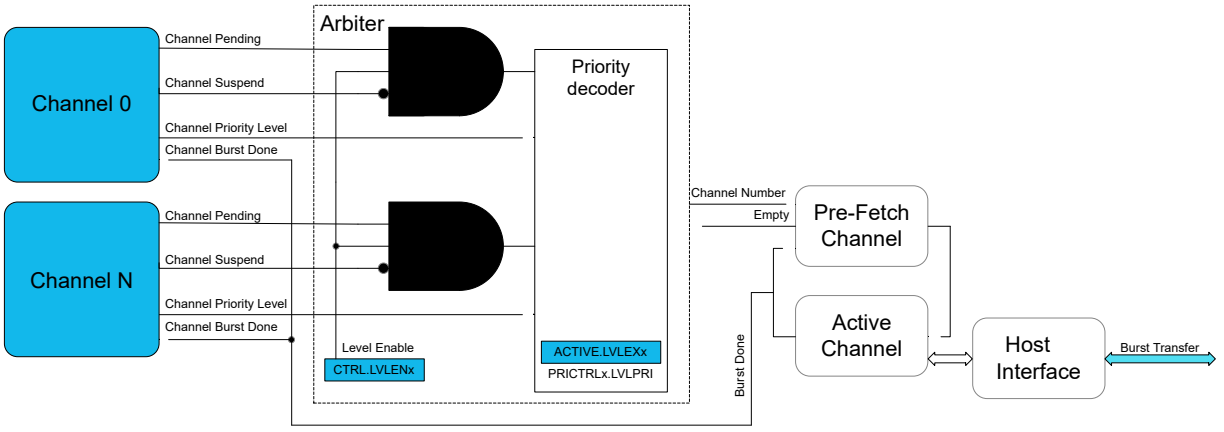
If a DMA channel is enabled and not suspended when it receives a transfer trigger, it will send a transfer request to the arbiter. When the arbiter receives the transfer request it will include the DMA channel in the queue of channels having pending transfers, and the corresponding Pending Channel x bit in the Pending Channels registers (PENDCH.PENDCHx) will be set. Depending on the arbitration scheme, the arbiter will choose which DMA channel will be the next active channel. The next transfer descriptor will be fetched from SRAM memory and stored internally in the Pre-Fetch Channel. The active channel is the DMA channel being granted access to perform its next burst transfer. When the Active Channel has completed a burst transfer, the descriptor stored in the Pre-Fetch Channel is transferred to the Active Channel and a new burst will take place.

When the descriptor stored in the Pre-Fetch Channel is transferred to the Active Channel, the corresponding PENDCH.PENDCHx will be cleared. In the same way, depending on trigger action settings and if the upcoming burst transfer is the first for the transfer request or not, the corresponding Busy Channel x bit in the Busy Channels register (BUSYCH.BUSYCHx), will either be set or remain '1'. 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 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. The status will also be indicated in CHINTFLAGn.SUSP. When the same DMA channel is resumed, it will be added to the queue of pending channels again.

If a DMA channel gets disabled (CHCTRLA.ENABLE=0) while it has a pending transfer, it will be removed from the queue of pending channels, and the corresponding PENDCH.PENDCHx will be cleared.

Figure 26-4. Arbiter Overview



Priority Levels

When a channel level is pending or the channel is transferring data, the corresponding Level Executing bit is set in the Active Channel and Levels register (ACTIVE.LVLEXx).

Each DMA channel supports up to 4-level priority scheme.

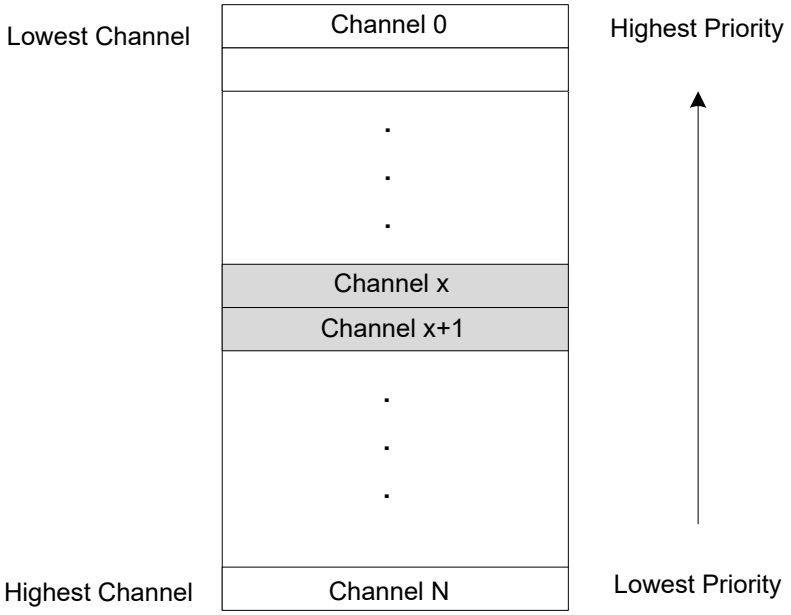
The priority level for a channel is configured by writing to the Channel Arbitration Level bit group in the Channel Priority Level register (CHPRILVL.PRILVL). As long as all priority levels are enabled, a channel with a higher priority level number will have priority over a channel with a lower priority

© 2025 Microchip Technology Inc. and its subsidiaries

level number. A priority level is enabled by writing the Priority Level x Enable bit in the Control register (CTRL.LVLENx) to '1', for the corresponding level.

Within each priority level, the DMAC's arbiter can be configured to prioritize statically or dynamically. For the arbiter to perform static arbitration within a priority level, the Level X Round-Robin Scheduling Enable bit in the Priority Control x register (PRICTRL0.RRLVLENx) has to be written to '0'. When static arbitration is enabled (PRICTRL0.RRLVLENx is '0'), the arbiter will prioritize a low channel number over a high channel number as shown in the following figure. When using the static scheme, there is a risk of high channel numbers never being granted access as the active channel. This can be avoided using a dynamic arbitration scheme.

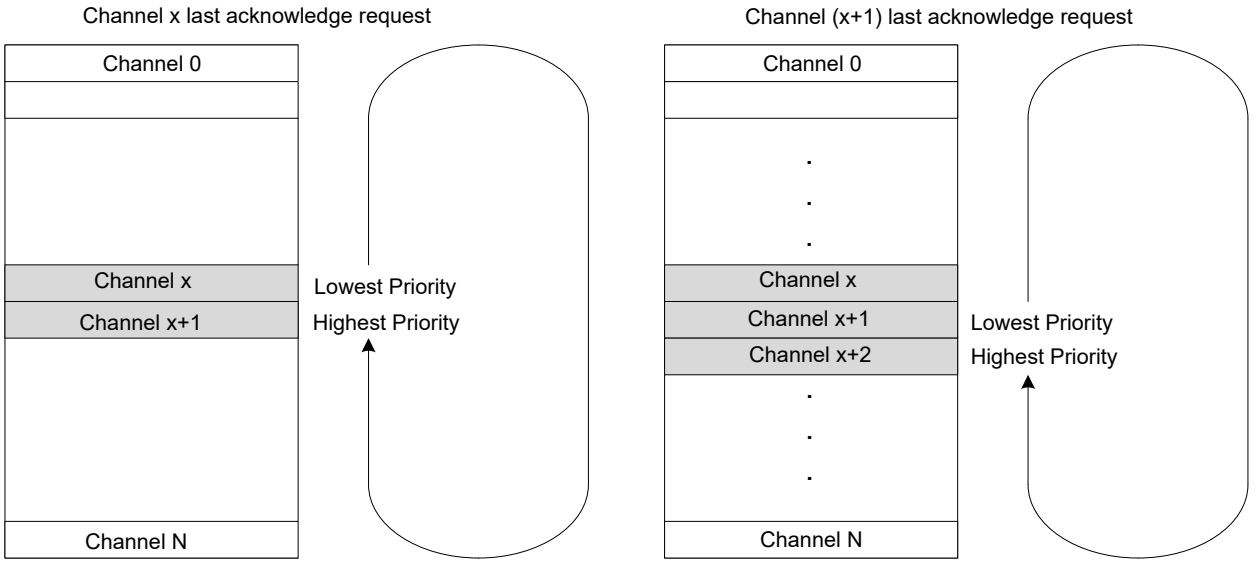
Figure 26-5. Static Priority Scheduling



The dynamic arbitration scheme in the DMAC is round-robin. Round-robin arbitration is enabled by writing PRICTRL0.RRLVLEN to '1', for a given priority level x. With the round-robin scheme, the channel number of the last channel being granted access will have the lowest priority the next time the arbiter has to grant access to a channel within the same priority level, as shown in the following figure. The channel number of the last channel being granted access as the active channel is stored in the Level x Channel Priority Number bit group in the Priority Control 0 register (PRICTRL0.LVLPRIx) for the corresponding priority level.

© 2025 Microchip Technology Inc. and its subsidiaries

Figure 26-6. Dynamic (Round-Robin) Priority Scheduling



26.6.2.5. Data Transmission

Before the DMAC can perform a data transmission, a DMA channel has to be configured and enabled, its corresponding transfer descriptor has to be initialized, and the arbiter has to grant the DMA channel access as the active channel.

Once the arbiter has granted a DMA channel access as the active channel (see *DMAC Block Diagram* in the *Block Diagram* from Related Links) the transfer descriptor for the DMA channel will be fetched from SRAM using the fetch bus, and stored in the internal memory for the active channel. For a new block transfer, the transfer descriptor will be fetched from the descriptor memory section (BASEADDR); For an ongoing block transfer, the descriptor will be fetched from the write-back memory section (WRBADDR). By using the data transfer bus, the DMAC will read the data from the current source address and write it to the current destination address. For further details on how the current source and destination addresses are calculated (see *Addressing* from the Related Links).

The arbitration procedure is performed after each transfer. If the current DMA channel is granted access again, the block transfer counter (BTCNT) of the internal transfer descriptor will be decremented by the number of beats in a transfer, the optional output event Beat will be generated if configured and enabled, and the active channel will perform a new transfer. If a different DMA channel than the current active channel is granted access, the block transfer counter value will be written to the write-back section before the transfer descriptor of the newly granted DMA channel is fetched into the internal memory of the active channel.

When a block transfer has come to its end (BTCNT is zero), the Valid bit in the Block Transfer Control register will be cleared (BTCTRL.VALID=0) before the entire transfer descriptor is written to the write-back memory. The optional interrupts, Channel Transfer Complete and Channel Suspend, and the optional output event Block, will be generated if configured and enabled. After the last block transfer in a transaction, the Next Descriptor Address register (DESCADDR) will hold the value 0x00000000, and the DMA channel will either be suspended or disabled, depending on the configuration in the Block Action bit group in the Block Transfer Control register (BTCTRL.BLOCKACT). If the transaction has further block transfers pending, DESCADDR will hold the SRAM address to the next transfer descriptor to be fetched. The DMAC will fetch the next descriptor into the internal memory of the active channel and write its content to the write-back section for the channel, before the arbiter gets to choose the next active channel.

© 2025 Microchip Technology Inc. and its subsidiaries

Related Links
Addressing
Block Diagram

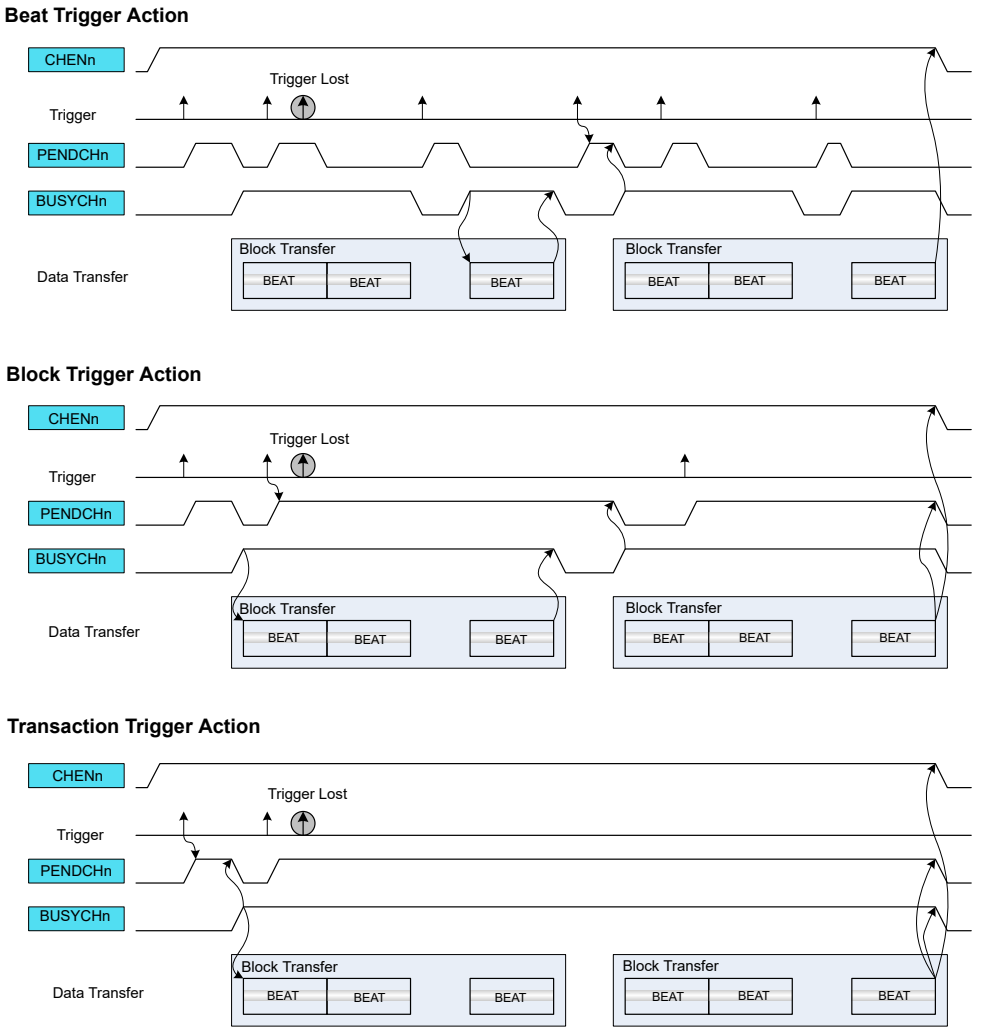
26.6.2.6. Transfer Triggers and Actions

A DMA transfer through a DMA channel can be started only when a DMA transfer request is detected, and the DMA channel has been granted access to the DMA. A transfer request can be triggered from software, from a peripheral, or from an event. There are dedicated Trigger Source selections for each DMA Channel n Control A (CHCTRLAn.TRIGSRC).

The trigger actions are available in the Trigger Action bit group in the Channel n Control A register (CHCTRLAn.TRIGACT). By default, a trigger generates a request for a block transfer operation. If a single descriptor is defined for a channel, the channel is automatically disabled when a block transfer has been completed. If a list of linked descriptors is defined for a channel, the channel is automatically disabled when the last descriptor in the list is executed. As long as the list still has descriptors to execute, the channel will be waiting for the next block transfer trigger. When enabled again, the channel will wait for the next block transfer trigger. The trigger actions can also be configured to generate a request for a burst transfer (CHCTRLAn.TRIGACT=0x2) or transaction transfer (CHCTRLAn.TRIGACT=0x3) instead of a block transfer (CHCTRLAn.TRIGACT=0x0).

The following figure shows an example where triggers are used with two linked block descriptors.

Figure 26-7. Trigger Action and Transfers



© 2025 Microchip Technology Inc. and its subsidiaries

If the trigger source generates a transfer request for a channel during an ongoing transfer, the new transfer request will be kept pending (CHSTATUSn.PEND=1), and the new transfer can start after the ongoing one is done. Only one pending transfer can be kept per channel. If the trigger source generates more transfer requests while one is already pending, the additional ones will be lost. All channels pending status flags are also available in the Pending Channels register (PENDCH).

When the transfer starts, the corresponding Channel Busy status flag is set in Channel n Status register (CHSTATUSn.BUSY). When the trigger action is complete, the Channel Busy status flag is cleared. All channel busy status flags are also available in the Busy Channels register (BUSYCH) in DMAC.

26.6.2.7. Addressing

Each block transfer needs to have both a source address and a destination address defined. The source address is set by writing the Transfer Source Address (SRCADDR) register, and the destination address is set by writing the Transfer Destination Address (DSTADDR) register.

The addressing of this DMAC module can be static or incremental, for either source or destination of a block transfer, or both.

Incrementation for the source address of a block transfer is enabled by writing the Source Address Incrementation Enable bit in the Block Transfer Control register (BTCTRL.SRCINC=1). The step size of the incrementation is configurable and can be chosen by writing the Step Selection bit in the Block Transfer Control register (BTCTRL.STEPSEL=1) and writing the desired step size in the Address Increment Step Size bit group in the Block Transfer Control register (BTCTRL.STEPSIZE). If BTCTRL.STEPSEL=0, the step size for the source incrementation will be the size of one beat.

When source address incrementation is configured (BTCTRL.SRCINC=1), SRCADDR is calculated as follows:

If BTCTRL.STEPSEL=1:

$$\mathrm{SRCADDR} = \mathrm{SRCADDR}_{START} + BTCNT \cdot \left(BEATSIZE + 1\right) \cdot 2^{\mathrm{STEPSIZE}}$$

If BTCTRL.STEPSEL=0:

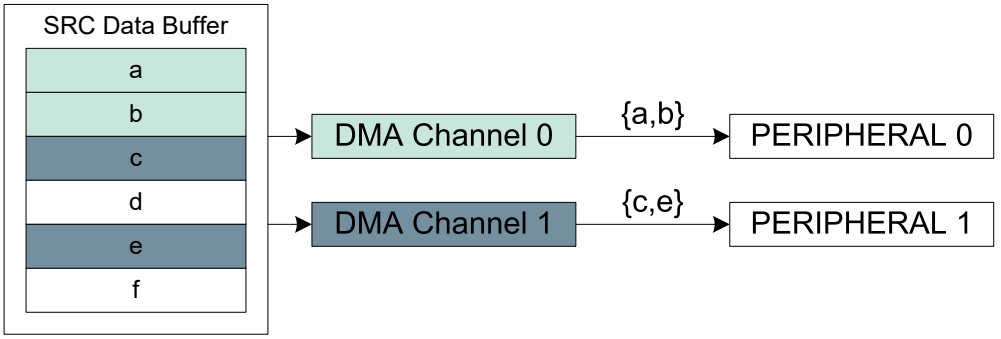
$$\mathrm{SRCADDR} = \mathrm{SRCADDR}_{START} + BTCNT \cdot \left(BEATSIZE + 1\right)$$

- $\mathrm{SRCADDR}_{START}$ is the source address of the first beat transfer in the block transfer
- BTCNT is the initial number of beats remaining in the block transfer
- BEATSIZE is the configured number of bytes in a beat
- STEPSIZE is the configured number of beats for each incrementation

The following figure shows an example where DMA channel 0 is configured to increment the source address by one beat after each beat transfer (BTCTRL.SRCINC=1), and DMA channel 1 is configured to increment the source address by two beats (BTCTRL.SRCINC=1, BTCTRL.STEPSEL=1 and BTCTRL.STEPSIZE=0x1). As the destination address for both channels are peripherals, destination incrementation is disabled (BTCTRL.DSTINC=0).

© 2025 Microchip Technology Inc. and its subsidiaries

Figure 26-8. Source Address Increment



Incrementation for the destination address of a block transfer is enabled by setting the Destination Address Incrementation Enable bit in the Block Transfer Control register (BTCTRL.DSTINC=1). The step size of the incrementation is configurable by clearing BTCTRL.STEPSEL=0 and writing BTCTRL.STEPSIZE to the desired step size. If BTCTRL.STEPSEL=1, the step size for the destination incrementation will be the size of one beat.

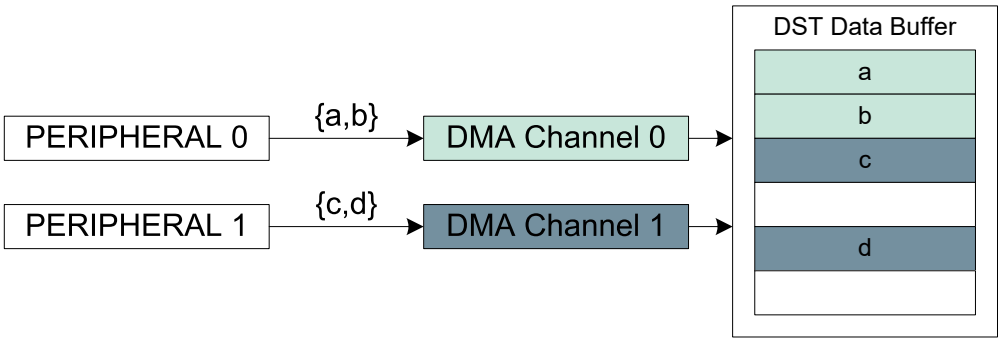
When the destination address incrementation is configured (BTCTRL.DSTINC=1), DSTADDR must be set and calculated as follows:

$DSTADDR = DSTADDR_{START} + BTCNT \bullet \left(BEATSIZE + 1\right) \bullet 2^{STEPSIZE}$	where BTCTRL.STEPSEL is zero
$DSTADDR = DSTADDR_{START} + BTCNT \bullet (BEATSIZE + 1)$	where BTCTRL.STEPSEL is one

- $DSTADDR_{START}$ is the destination address of the first beat transfer in the block transfer
- BTCNT is the initial number of beats remaining in the block transfer
- BEATSIZE is the configured number of bytes in a beat
- STEPSIZE is the configured number of beats for each incrementation

The following figure shows an example where DMA channel 0 is configured to increment destination address by one beat (BTCTRL.DSTINC=1) and DMA channel 1 is configured to increment destination address by two beats (BTCTRL.DSTINC=1, BTCTRL.STEPSEL=0 and BTCTRL.STEPSIZE=0x1). As the source address for both channels are peripherals, source incrementation is disabled (BTCTRL.SRCINC=0).

Figure 26-9. Destination Address Increment



26.6.2.8. Internal FIFO

To improve the bandwidth, the DMAC can support FIFO operation. When single-beat burst configuration is selected (CHCTRALx.BURSTLEN = SINGLE), the channel waits until the FIFO can

© 2025 Microchip Technology Inc. and its subsidiaries

transmit or accept a single beat transfer before it requests a bus access to write to the destination address. In all other cases, the channel waits until the FIFO threshold is reached before it requests a bus access to write to the destination address. The threshold is configurable and can be set by writing the THRESHOLD bits in the Channel x Control A register.

If the DMAC completes the read operations before the threshold is reached, the write to the destination is automatically enabled. If the FIFO is empty and the read from source is ongoing, the DMA will wait again until the FIFO threshold is reached before it requests a bus access to write the destination.

26.6.2.9. Error Handling

If a bus error is received from an AHB client during a DMA data transfer, the corresponding active channel is disabled and the corresponding Channel Transfer Error Interrupt flag in the Channel Interrupt Status and Clear register (CHINTFLAG.TERR) is set. If enabled, the optional transfer error interrupt is generated. The transfer counter will not be decremented and its current value is written-back in the write-back memory section before the channel is disabled.

When the DMAC fetches an invalid descriptor (BTCTRL.VALID=0) or when the channel is resumed and the DMA fetches the next descriptor with null address (DESCADDR=0x00000000), the corresponding channel operation is suspended, the Channel Suspend Interrupt Flag in the Channel Interrupt Flag Status and Clear register (CHINTFLAG.SUSP) is set, and the Channel Fetch Error bit in the Channel Status register (CHSTATUS.FERR) is set. If enabled, the optional suspend interrupt is generated.

26.6.3. Additional Features

26.6.3.1. Linked Descriptors

A transaction can consist of either a single block transfer or of several block transfers. When a transaction consists of several block transfers it is done with the help of linked descriptors.

Memory Sections illustrates how linked descriptors work (see *Memory Sections* figure in the *Transfer Descriptors* from Related Links). When the first block transfer is completed on DMA channel 0, the DMAC fetches the next transfer descriptor, which is pointed to by the value stored in the Next Descriptor Address (DESCADDR) register of the first transfer descriptor. Fetching the next transfer descriptor (DESCADDR) is continued until the last transfer descriptor. When the block transfer for the last transfer descriptor is executed and DESCADDR = 0x00000000, the transaction is terminated. For further details on how the next descriptor is fetched from SRAM (see *Data Transmission* from Related Links).

Related Links

Data Transmission

Transfer Descriptors

26.6.3.1.1. Adding Descriptor to the End of a List

To add a new descriptor at the end of the descriptor list, create the descriptor in SRAM, with DESCADDR = 0x00000000 indicating that it is the new last descriptor in the list, and modify the DESCADDR value of the current last descriptor to the address of the newly created descriptor.

26.6.3.1.2. Modifying a Descriptor in a List

In order to add descriptors to a linked list, the following actions must be performed:

1. Enable the Suspend interrupt for the DMA channel.
2. Enable the DMA channel.
3. Reserve memory space in SRAM to configure a new descriptor.
4. Configure the new descriptor:
   - Set the next descriptor address (DESCADDR)

© 2025 Microchip Technology Inc. and its subsidiaries

- Set the destination address (DESCADDR)
- Set the source address (SRCADDR)
- Configure the block transfer control (BTCTRL) including
  - Optionally enable the suspend block action
  - Set the descriptor VALID bit

5. Clear the VALID bit for the existing list and for the descriptor which has to be updated.
6. Read DESCADDR from the write-back memory.
   - If the DMA has not already fetched the descriptor that requires changes (in other words, DESCADDR is wrong):
     - Update the DESCADDR location of the descriptor from the list
     - Optionally clear the suspend block action
     - Set the descriptor VALID bit to '1'
     - Optionally enable the Resume Software command
   - If the DMA is executing the same descriptor as the one that requires changes:
     - Set the Channel Suspend Software command and wait for the suspend interrupt
     - Update the next descriptor address (DESCADDR) in the write-back memory
     - Clear the interrupt sources and set the Resume Software command
     - Update the DESCADDR location of the descriptor from the list
     - Optionally clear the suspend block action
     - Set the descriptor VALID bit to '1'
7. Go to step 4 if needed.

26.6.3.1.3. Adding a Descriptor Between Existing Descriptors

To insert a new descriptor 'C' between two existing descriptors ('A' and 'B'), the descriptor currently executed by the DMA must be identified.

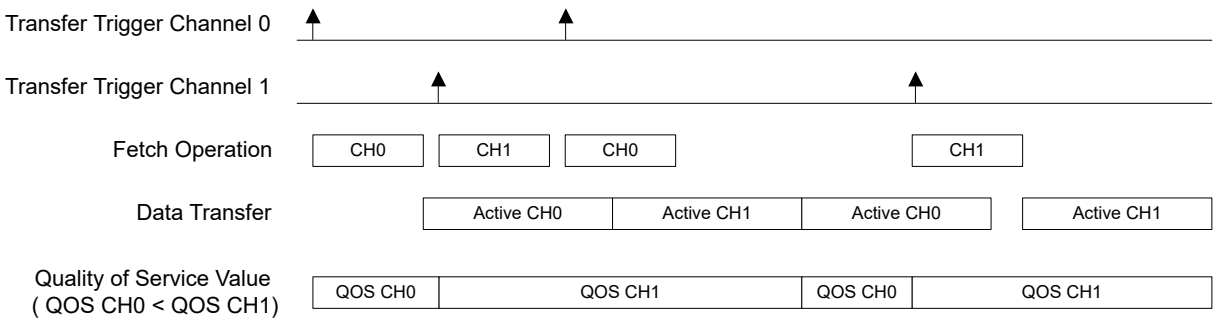
1. If DMA is executing descriptor B, descriptor C cannot be inserted.
2. If DMA has not started to execute descriptor A, follow the steps:
   a. Set the descriptor A VALID bit to '0'.
   b. Set the DESCADDR value of descriptor A to point to descriptor C instead of descriptor B.
   c. Set the DESCADDR value of descriptor C to point to descriptor B.
   d. Set the descriptor A VALID bit to '1'.
3. If DMA is executing descriptor A:
   a. Apply the software suspend command to the channel and
   b. Perform steps 2.1 through 2.4.
   c. Apply the software resume command to the channel.

26.6.3.2. Transfer Quality of Service

Each priority level group has dedicated quality of service settings. The setting can be written in the corresponding Quality of Service bit group in the Priority Control x register (PRICTRL0.QOSn).

© 2025 Microchip Technology Inc. and its subsidiaries

Figure 26-10. Quality of Service



When a channel is stored in the Pre-Fetch or Active Channel, the corresponding PRICTRLx.QOS bits value is stored in the respective channel. As shown in Quality of Service, the DMAC will select the highest QOS value between Active and Pre-Fetch channels. This value will apply to all DMAC buses.

26.6.3.3. Channel Suspend

The channel operation can be suspended at any time by software by writing a '1' to the Suspend command in the Command bit field of Channel Control B register (CHCTRLB.CMD). After the ongoing burst transfer is completed, the channel operation is suspended and the suspend command is automatically cleared.

When suspended, the Channel Suspend Interrupt flag in the Channel Interrupt Status and Clear register is set (CHINTFLAG.SUSP=1) and the optional suspend interrupt is generated.

By configuring the block action to suspend by writing Block Action bit group in the Block Transfer Control register (BTCTRL.BLOCKACT is 0x2 or 0x3), the DMA channel will be suspended after it has completed a block transfer. The DMA channel will be kept enabled and will be able to receive transfer triggers, but it will be removed from the arbitration scheme.

If an invalid transfer descriptor (BTCTRL.VALID=0) is fetched from SRAM, the DMA channel will be suspended, and the Channel Fetch Error bit in the Channel Status register (CHASTATUS.FERR) will be set.

Note: Only enabled DMA channels can be suspended. If a channel is disabled when it is attempted to be suspended, the internal suspend command will be ignored.

For more details on transfer descriptors (see *Transfer Descriptors* from Related Links).

Related Links

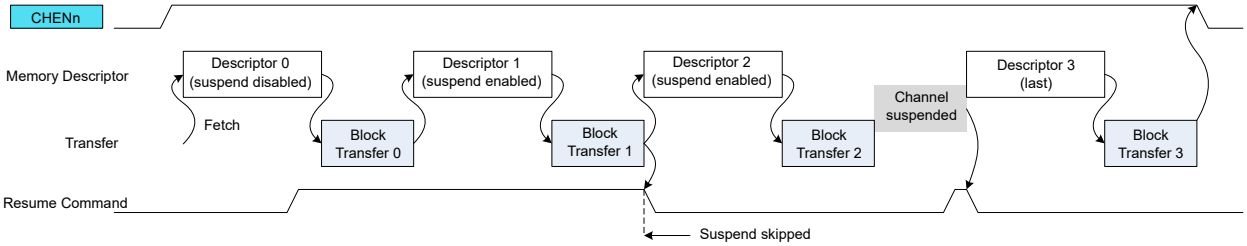
Transfer Descriptors

26.6.3.4. Channel Resume and Next Suspend Skip

A channel operation can be resumed by software by setting the Resume command in the Command bit field of the Channel Control B register (CHCTRLB.CMD). If the channel is already suspended, the channel operation resumes from where it previously stopped when the Resume command is detected. When the Resume command is issued before the channel is suspended, the next suspend action is skipped and the channel continues the normal operation.

© 2025 Microchip Technology Inc. and its subsidiaries

Figure 26-11. Channel Suspend/Resume Operation



26.6.3.5. Event Input Actions

The event input actions are available only on the least significant DMA channels. For more details on channels with event input support (see *Event System (EVSYS)* from Related Links).

Before using event input actions, the event controller must be configured first according to the following table, and the Channel Event Input Enable bit in the Channel Event Control register (CHEVCTRL.EVIE) must be written to '1'. See *Events* from Related Links.

Table 26-1. Event Input Action

Action	CHEVCTRL.EVACT	CHCTRLA.TRIGSRC
None	NOACT	—
Normal Transfer	TRIG	DISABLE
Conditional Transfer on Strobe	TRIG	Any peripheral
Conditional Transfer	CTRIG	
Conditional Block Transfer	CBLOCK	
Channel Suspend	SUSPEND	
Channel Resume	RESUME	
Skip Next Block Suspend	SSKIP	
Increase priority	INCPRI	

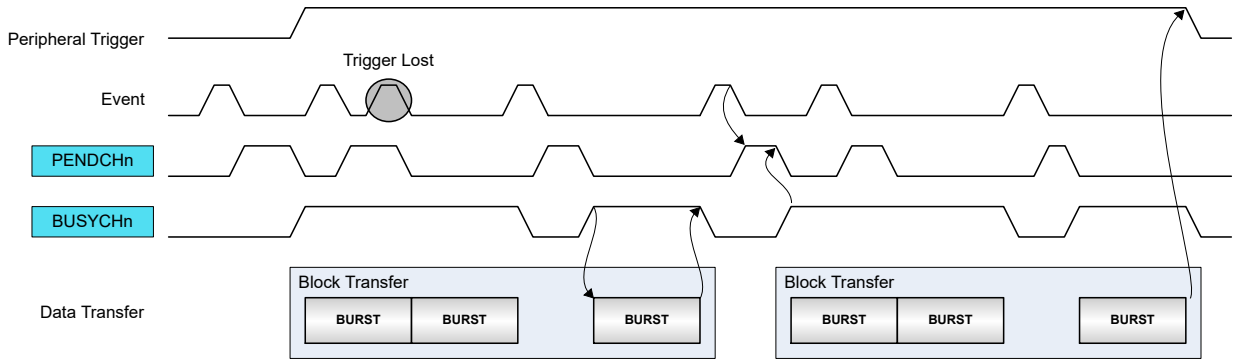
Normal Transfer

The event input is used to trigger a beat or burst transfer on peripherals.

The event is acknowledged as soon as the event is received. When received, both the Channel Pending status bit in the Channel Status register (CHSTATUS.PEND) and the corresponding Channel n bit in the Pending Channels register (PENDCH.PENDCHn) are set. If the event is received while the channel is pending, the event trigger is lost.

The following figure shows an example where beat transfers are enabled by internal events.

Figure 26-12. Burst Event Trigger Action



© 2025 Microchip Technology Inc. and its subsidiaries

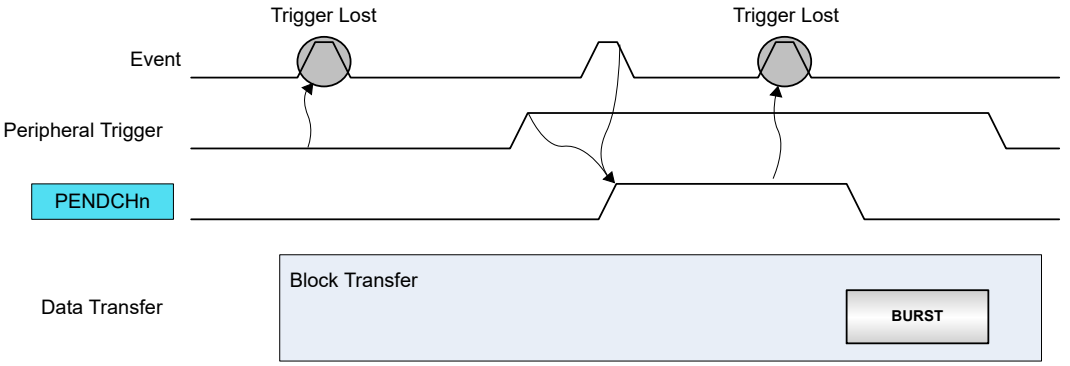
Conditional Transfer on Strobe

The event input is used to trigger a transfer on peripherals with pending transfer requests. This event action is intended to be used with peripheral triggers, for example, for timed communication protocols or periodic transfers between peripherals: only when the peripheral trigger coincides with the occurrence of a (possibly cyclic) event the transfer is issued.

The event is acknowledged as soon as the event is received. The peripheral trigger request is stored internally when the previous trigger action is completed (in other words, the channel is not pending) and when an active event is received. If the peripheral trigger is active, the DMA waits for an event before the peripheral trigger is internally registered. When both event and peripheral transfer trigger are active, both CHSTATUS.PEND and PENDCH.PENDCHn are set. A software trigger will now trigger a transfer.

The following figure shows an example where the peripheral beat transfer is started by a conditional strobe event action.

Figure 26-13. Periodic Event with Burst Peripheral Triggers



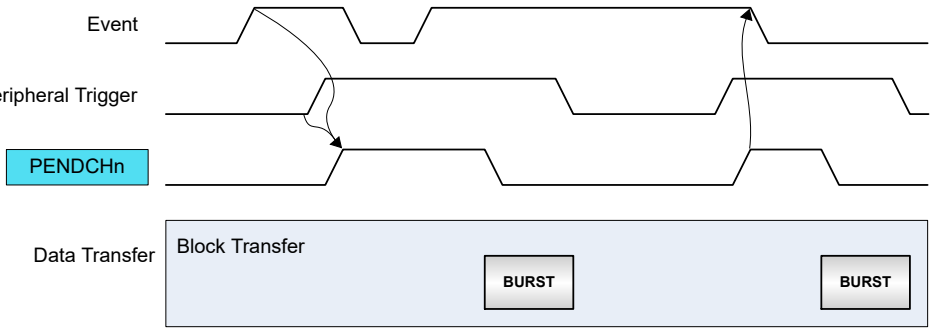
Conditional Transfer

The event input is used to trigger a conditional transfer on peripherals with pending transfer requests. As example, this type of event can be used for peripheral-to-peripheral transfers, where one peripheral is the source of event and the second peripheral is the source of the trigger.

Each peripheral trigger is stored internally when the event is received. When the peripheral trigger is stored internally, the Channel Pending status bit is set (CHSTATUS.PEND), the respective Pending Channel n Bit in the Pending Channels register is set (PENDCH.PENDCHn), and the event is acknowledged. A software trigger will now trigger a transfer.

The following figure shows an example where conditional event is enabled with peripheral beat trigger requests.

Figure 26-14. Conditional Event with Burst Peripheral Triggers



© 2025 Microchip Technology Inc. and its subsidiaries

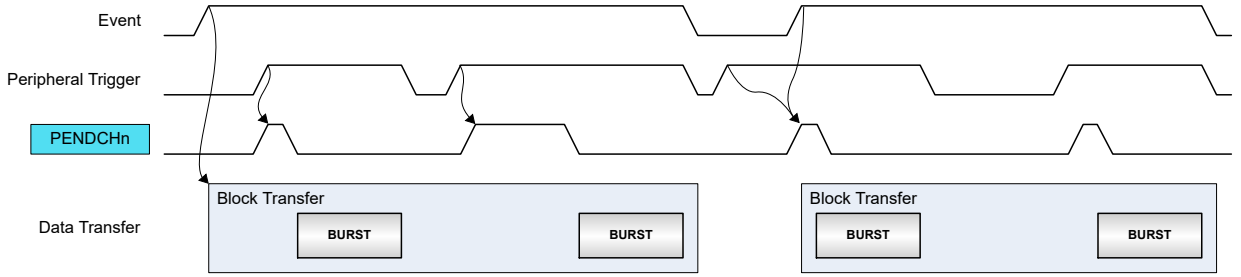
Conditional Block Transfer

The event input is used to trigger a conditional block transfer on peripherals.

Before starting transfers within a block, an event must be received. When received, the event is acknowledged when the block transfer is completed. A software trigger will trigger a transfer.

The following figure shows an example where conditional event block transfer is started with peripheral beat trigger requests.

Figure 26-15. Conditional Block Transfer with Burst Peripheral Triggers



Channel Suspend

The event input is used to suspend an ongoing channel operation. The event is acknowledged when the current AHB access is completed. For more details on Channel Suspend (see *Channel Suspend* from Related Links).

Channel Resume

The event input is used to resume a suspended channel operation. The event is acknowledged as soon as the event is received and the Channel Suspend Interrupt Flag (CHINTFLAG.SUSP) is cleared. See *Channel Suspend* from Related Links.

Skip Next Block Suspend

This event can be used to skip the next block suspend action. If the channel is suspended before the event rises, the channel operation is resumed and the event is acknowledged. If the event rises before a suspend block action is detected, the event is kept until the next block suspend detection. When the block transfer is completed, the channel continues the operation (not suspended) and the event is acknowledged.

Increase priority

This event can be used to increase a channel priority and to request higher quality of service (QOS), when critical transfers must be done. When the event is detected, the channel will have the highest priority and the output Quality of Service value is internally forced to the maximum value. The event is acknowledged when the trigger action execution is completed. When acknowledged, the channel will recover its initial priority level and quality of service settings.

Related Links

Channel Suspend
Event System (EVSYS)
Events

26.6.3.6. Event Output Selection

The event output selections are available only for channels supporting event outputs.

The Channel Event Output Enable can be set in the corresponding Channel n Event Control register (CHEVCTRL.EVOE). The Event Output Mode bits in the Channel n Event Control register (CHEVCTRL.EVOMODE) selects the event type the channel will generate.

© 2025 Microchip Technology Inc. and its subsidiaries

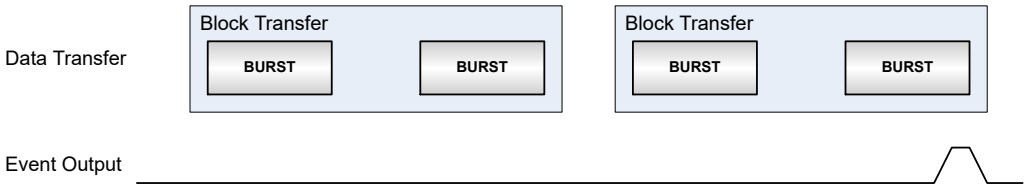
The transfer events (CHEVCTRL.EVOMODE = DEFAULT) are strobe events and their duration is one CLK_DMAC_AHB clock period. The transfer event type selection is available in each Descriptor Block Control location (BTCTRL.EVOSEL). Block or burst event output generation is supported.

The trigger action event (CHEVCTRL.EVOMODE = TRIGACT) is a level, active while the trigger action execution is not completed.

Block Event Output

When the block event output is selected, an event strobe is generated when the block transfer is completed. The pulse width of a block event output from a channel is one AHB clock cycle. It is also possible to use this event type to generate an event when the transaction is complete. For this type of application, the block event selection must be set in the last transfer descriptor only, as shown below.

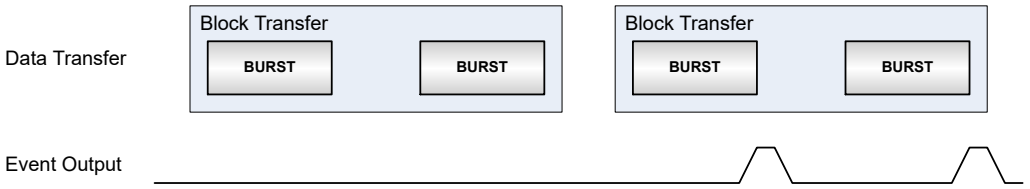
Figure 26-16. Block Event Output Generation



Burst Event Output

When the burst event output is selected, an event strobe is generated when each burst transfer within the corresponding block is completed. The pulse width of a burst event output from a channel is one AHB clock cycle. The figure below shows an example where the burst event output is set in the second descriptor of a linked list.

Figure 26-17. Burst Event Output Generation

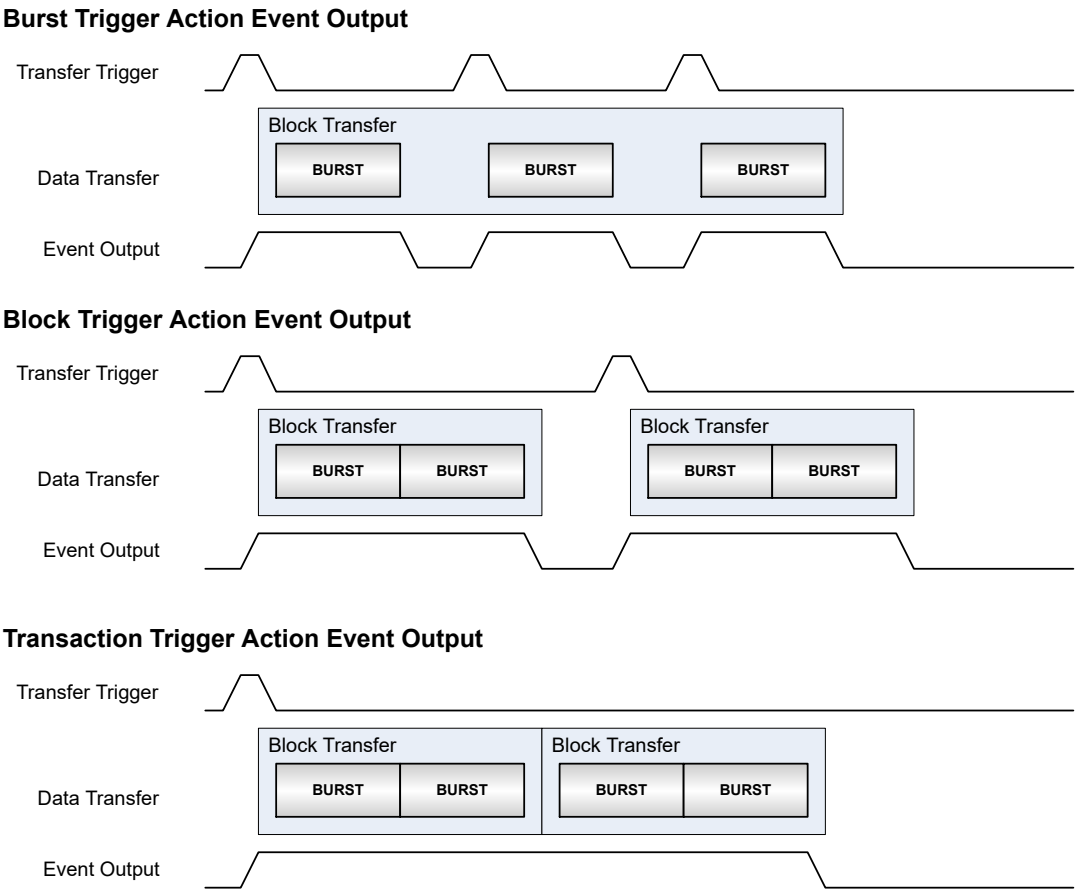


Trigger Action Event Output

When the trigger action event output is selected, an event level is generated. The event output is set when the transfer trigger occurred, and cleared when the corresponding trigger action is completed. The following figure shows an example for each trigger action type.

© 2025 Microchip Technology Inc. and its subsidiaries

Figure 26-18. Trigger Action Event Output Generation



26.6.3.7. Aborting Transfers

Transfers on any channel can be aborted gracefully by software by disabling the corresponding DMA channel. It is also possible to abort all ongoing or pending transfers by disabling the DMAC.

When a DMA channel disable request or DMAC disable request is detected:

- Ongoing transfers of the active channel will be disabled when the ongoing beat transfer is completed and the write-back memory section is updated. This prevents transfer corruption before the channel is disabled.
- All other enabled channels will be disabled in the next clock cycle.

The corresponding Channel Enable bit in the Channel Control A register is cleared (CHCTRLA.ENABLE=0) when the channel is disabled.

The corresponding DMAC Enable bit in the Control register is cleared (CTRL.DMAENABLE=0) when the entire DMAC module is disabled.

26.6.3.8. CRC Operation

A Cyclic Redundancy Check (CRC) is an error detection technique used to find errors in data. It is commonly used to determine whether the data during a transmission, or data present in data and program memories has been corrupted or not. A CRC takes a data stream or a block of data as input and generates a 16- or 32-bit output that can be appended to the data and used as a checksum.

When the data is received, the device or application repeats the calculation using the DSU's CRC engine. If the new CRC result does not match the one calculated earlier, the block contains a data error. The application will then detect this and may take a corrective action, such as requesting the data to be sent again or simply not using the incorrect data.

© 2025 Microchip Technology Inc. and its subsidiaries

The CRC engine in DMAC supports two commonly used CRC polynomials: CRC-16 (CRC-CCITT) and CRC-32 (IEEE 802.3). Typically, applying CRC-n (CRC-16 or CRC-32) to a data block of arbitrary length will detect any single alteration that is ≤n bits in length, and will detect the fraction 1-2-n of all longer error bursts.

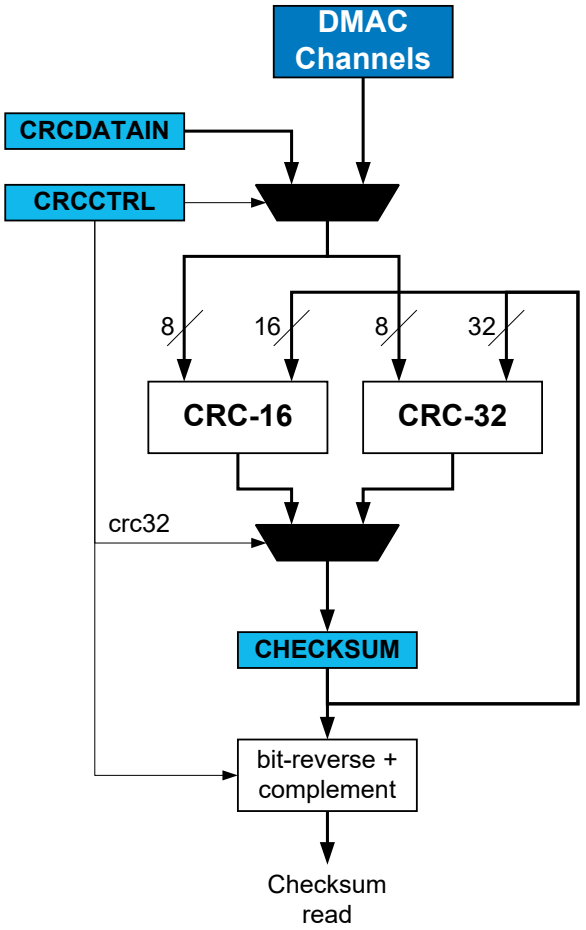
- CRC-16:
  - Polynomial: $x^{16}+ x^{12}+ x^5+ 1$
  - Hex value: 0x1021
- CRC-32:
  - Polynomial: $x^{32}+x^{26}+ x^{23}+ x^{22}+x^{16}+ x^{12}+ x^{11}+ x^{10}+ x^8+ x^7+ x^5+ x^4+ x^2+ x + 1$
  - Hex value: 0x04C11DB7

The data source for the CRC engine can either be one of the DMA channels or the APB bus interface, and must be selected by writing to the CRC Input Source bits in the CRC Control register (CRCCTRL.CRCSRC). The CRC engine then takes data input from the selected source and generates a checksum based on these data. The checksum is available in the CRC Checksum register (CRCCHKSUM). When CRC-32 polynomial is used, the final checksum read is bit reversed and complemented, as shown in *CRC Generator Block Diagram*.

The CRC polynomial is selected by writing to the CRC Polynomial Type bit in the CRC Control register (CRCCTRL.CRCPOLY), the default is CRC-16. The CRC engine operates on byte only. When the DMA is used as data source for the CRC engine, the DMA channel beat size setting will be used. When used with APB bus interface, the application must select the CRC Beat Size bit field of CRC Control register (CRCCTRL.CRCBEATSIZE). 8-, 16-, or 32-bit bus transfer access type is supported. The corresponding number of bytes will be written in the CRCDATAIN register and the CRC engine will operate on the input data in a byte by byte manner.

© 2025 Microchip Technology Inc. and its subsidiaries

Figure 26-19. CRC Generator Block Diagram



CRC on DMA data	CRC-16 or CRC-32 calculations can be performed on data passing through any DMA channel. Once a DMA channel is selected as the source, the CRC engine will continuously generate the CRC on the data passing through the DMA channel. The checksum is available for readout once the DMA transaction is completed or aborted. A CRC can also be generated on SRAM, Flash, or I/O memory by passing these data through a DMA channel. If the latter is done, the destination register for the DMA data can be the data input (CRCDATAIN) register in the CRC engine.
CRC using the I/O interface	Before using the CRC engine with the I/O interface, the application must set the CRC Beat Size bits in the CRC Control register (CRCCTRL.CRCBEATSIZE). 8/16/32-bit bus transfer type can be selected.

CRC can be performed on any data by loading them into the CRC engine using the CPU and writing the data to the CRCDATAIN register. Using this method, an arbitrary number of bytes can be written to the register by the CPU, and CRC is done continuously for each byte. This means if a 32-bit data is written to the CRCDATAIN register the CRC engine takes four cycles to calculate the CRC. The CRC complete is signaled by a set CRCBUSY bit in the CRCSTATUS register. New data can be written only when CRCBUSY flag is not set.

26.6.3.9. Memory CRC Generation

When enabled, it is possible to automatically calculate a memory block checksum. When the channel is enabled and the descriptor is fetched, the CRC Checksum register (CRCCHKSUM) is reloaded with the initial checksum value (CHKINIT) stored in the Block Transfer Destination Address register (DSTADDR). The DMA read and calculate the checksum over the data from the source address.When the checksum calculation is completed, the CRC value is stored in the CRC Checksum register

© 2025 Microchip Technology Inc. and its subsidiaries

(CRCCHKSUM), the Transfer Complete interrupt flag is set (CHINTFLAGn.TCMPL) and optional interrupt is generated.

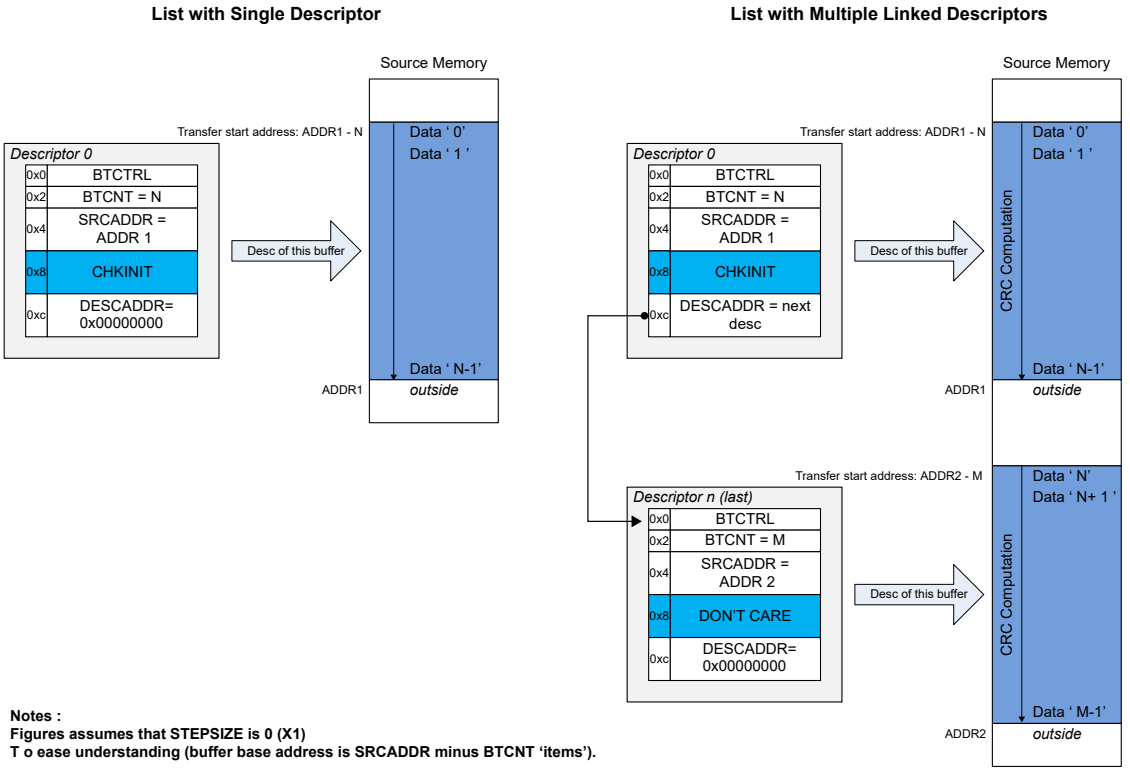
If linked descriptor is in the list (DESCADDR !=0), the DMA will fetch the next descriptor and CRC calculation continues as described above. When the last list descriptor is executed, the channel is automatically disabled.

In order to enable the memory CRC generation, the following actions must be performed:

1. The CRC module must be set to be used with a DMA channel (CRCCTRL.CRCSRC)
2. Reserve memory space addresses to configure a descriptor or a list of descriptors
3. Configure each descriptor:
   - Set the next descriptor address (DESCADDR)
   - Set the destination address with the initial checksum value (DSTADDR = CHKINIT) in the first descriptior in a list
   - Set the transfer source address (SRCADDR)
   - Set the block transfer count (BTCNT)
   - Set the memory CRC generation operation mode (CRCCTRL.CRCMODE = CRCGEN)
   - Enable optional interrupts
4. Enable the corresponding DMA channel (CHCTRLAn.ENABLE)

The figure below shows the CRC computation slots and descriptor configuration when single or linked-descriptors transfers are enabled.

Figure 26-20. CRC Computation with Single Linked Transfers



© 2025 Microchip Technology Inc. and its subsidiaries

26.6.3.10. Memory CRC Monitor

When enabled, it is possible to continuously check a a memory block data integrity by calculating and checking the CRC checksum. The following table shows the expected CRC checksum value that must be located in the last memory block location.

CRCCTRL.CRCPOLY	CRCCTRL.CRCBEATSIZE	Last Memory Block Byte Locations Value (MSB Byte First)	CHECKSUM Result
CRC-16	Byte	Expected CRC[7:0]	0x00000000
	Half-word	Expected CRC[15:8]	
	Word	0x00 0x00 Expected CRC[7:0] Expected CRC[15:8]	
CRC-32	Byte	Expected CRC[31:24]	CRC Magic Number (0x2144DF1C)
	Half-word	Expected CRC[23:16]	
	Word	Expected CRC[15:8] Expected CRC[7:0]	

When the channel is enabled and the descriptor is fetched, the CRC Checksum register (CRCCHKSUM) is reloaded with the initial checksum value (CHKINIT) stored in the DSTADDR location of the first descriptor. The DMA read and calculate the checksum over the entire data from the source address. When the checksum calculation is completed, the DMA read the last beat from the memory, and the calculated CRC value from the CRC Checksum register is compared to zero or the CRC magic number depending on CRC polynomial selection.

If the CHECKSUM does not match the comparison value, the DMA channel is disabled and both the CRC Error bit in the Channel n Status register (CHSTATUSn.CRCERR) and Transfer Error interrupt flag (CHINTFLAGn.TERR) are set. If enabled, the Transfer Error interrupt is generated.

If the calculated checksum value matches the compare value, the Transfer Complete Interrupt flag (CHINTFLAGn.TCMPL) is set, optional interrupt is generated and the DMA performs the following actions depending on the descriptor list settings:

- If the list has only one descriptor, the DMA re-fetches the descriptor.
- If the current descriptor is the last descriptor from the list, the DMA fetches the first descriptor from the list.

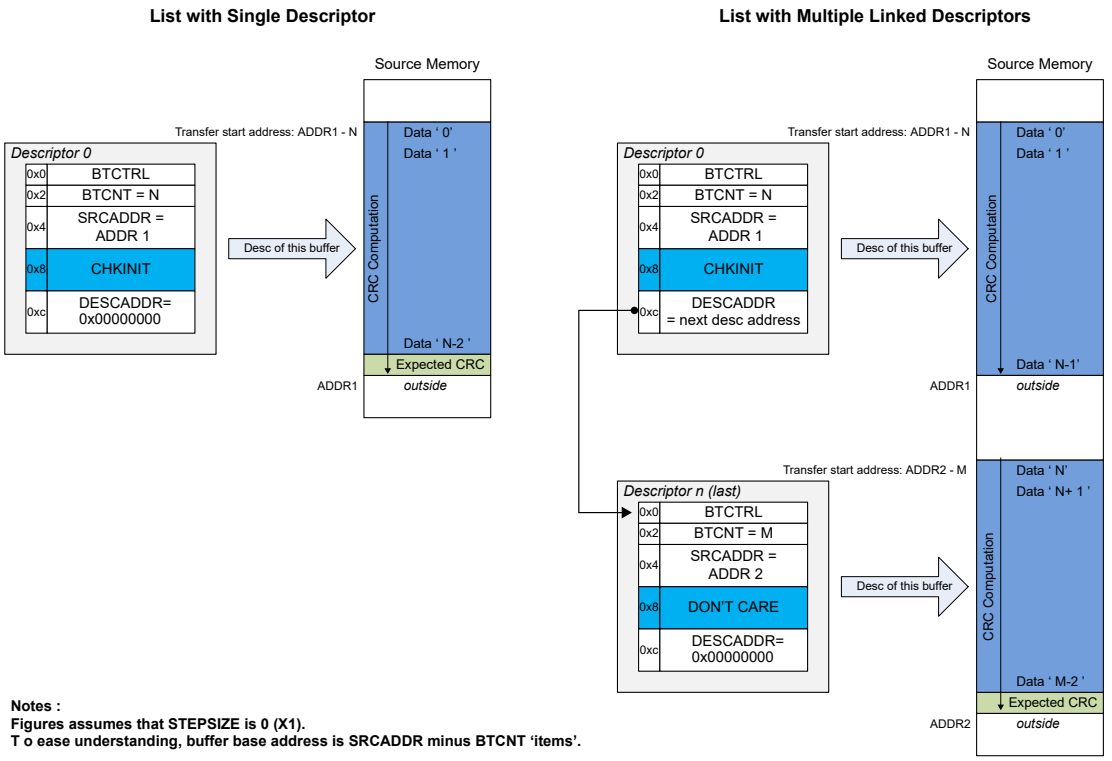
When the fetch is completed, the DMA restarts the operations described above when new triggers are detected.

To enable the memory CRC monitor, the following actions must be performed:

1. The CRC module must be set to be used with a DMA channel (CRCCTRL.CRCSRC).
2. Reserve memory space addresses to configure a descriptor or a list of descriptors.
3. Configure each descriptor.
   - Set the next descriptor address (DESCADDR).
   - In the first list descriptor, set the destination address with the initial checksum value (DSTADDR = CHKINIT).
   - Set the transfer source address (SRCADDR).
   - Set the block transfer count (BTCNT).
   - Set the memory CRC monitor operation mode (CRCCTRL.CRCMODE = CRCMON).
   - Enable optional interrupts.
4. Enable the corresponding DMA channel (CHCTRLAn.ENABLE).

© 2025 Microchip Technology Inc. and its subsidiaries

Figure 26-21. CRC Computation and Check with Single or Linked Transfers



Notes :
Figures assumes that STEPSIZE is 0 (X1).
T o ease understanding, buffer base address is SRCADDR minus BTCNT 'items'.

26.6.4. DMA Operation

Not applicable.

26.6.5. Interrupts

The DMAC channels have the following interrupt sources:

- Transfer Complete (TCMPL) – Indicates that a block transfer is completed on the corresponding channel. See *Data Transmission* from Related Links.
- Transfer Error (TERR) – Indicates that a bus error has occurred during a burst transfer or that an invalid descriptor was fetched. See *Error Handling* from Related Links.
- Channel Suspend (SUSP) – Indicates that the corresponding channel was suspended. See *Channel Suspend* and *Data Transmission* from Related Links.

Each interrupt source has an Interrupt flag associated with it. The Interrupt flag in the Channel Interrupt Flag Status and Clear (CHINTFLAG) register is set when the Interrupt condition occurs. Each interrupt can be individually enabled by setting the corresponding bit in the Channel Interrupt Enable Set register (CHINTENSET=1) and disabled by setting the corresponding bit in the Channel Interrupt Enable Clear register (CHINTENCLR=1). The status of enabled interrupts can be read from either INTENSET or INTENCLR.

An interrupt request is generated when the Interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the Interrupt flag is cleared, the interrupt is disabled, the DMAC is reset or the corresponding DMA channel is reset. For more details on how to clear Interrupt flags, see the *CHINTFLAG* register from Related Links. All interrupt requests are ORed together on system level to generate one combined interrupt request to the NVIC. See *Nested Vector Interrupt Controller (NVIC)* from Related Links.

The user must read the Channel Interrupt Status (INTSTATUS) register to identify the channels with pending interrupts and must read the Channel Interrupt Flag Status and Clear (CHINTFLAG) register

© 2025 Microchip Technology Inc. and its subsidiaries

to determine which Interrupt condition is present for the corresponding channel. It is also possible to read the Interrupt Pending register (INTPEND), which provides the lowest channel number with pending interrupt and the respective Interrupt flags.

Note: Interrupts must be globally enabled for interrupt requests to be generated.

Related Links

Nested Vector Interrupt Controller (NVIC)
Channel Suspend
Error Handling
Data Transmission
CHINTFLAG
Channel Interrupt Flag Status and Clear

26.6.6. Events

The DMAC can generate the following output events:

- Channel (CH) – Generated when a block transfer for a given channel completes or when a beat transfer within a block transfer for a given channel completes. See *Event Output Selection* from Related Links.

Setting the Channel Event Output Enable bit (CHEVCTRLx.EVOE=1) enables the corresponding output event configured in the Event Output Selection bit group in the Block Transfer Control register (BTCTRL.EVOSEL). Clearing CHEVCTRLx.EVOE=0 disables the corresponding output event.

The DMAC can take the following actions on an input event:

- Transfer and Periodic Transfer Trigger (TRIG) – Normal transfer or periodic transfers on peripherals are enabled
- Conditional Transfer Trigger (CTRIG) – Conditional transfers on peripherals are enabled
- Conditional Block Transfer Trigger (CBLOCK) – Conditional block transfers on peripherals are enabled
- Channel Suspend Operation (SUSPEND) – Suspend a channel operation
- Channel Resume Operation (RESUME) – Resume a suspended channel operation
- Skip Next Block Suspend Action (SSKIP) – Skip the next block suspend transfer condition
- Increase Priority (INCPRI) – Increase channel priority

Setting the Channel Event Input Enable bit (CHEVCTRLx.EVIE=1) enables the corresponding action on input event. Clearing this bit disables the corresponding action on input event. Note that several actions can be enabled for incoming events. If several events are connected to the peripheral, any enabled action is taken for any of the incoming events. See *Event Input Actions* from Related Links for more details on event input actions.
Note: Event input and outputs are not available for every channel. See *Features* from Related Links.

Related Links

Features
Event Input Actions
Event Output Selection

26.6.7. Sleep Mode Operation

Each DMA channel can be configured to operate in any sleep mode. To run in standby, the RUNSTDBY bit in Channel Control A register (CHCTRLA.RUNSTDBY) must be written to "1". The DMAC can wake-up the device using interrupts from any sleep mode or perform actions through the Event System.

© 2025 Microchip Technology Inc. and its subsidiaries

For channels with CHCTRLA.RUNSTDBY = 0, it is up to software to stop DMA transfers on these channels and wait for completion before going to Standby mode using the following sequence:

1. Suspend the DMAC channels for which CHCTRLA.RUNSTDBY = 0.
2. Check the SYNCBUSY bits of registers accessed by the DMAC channels being suspended.
3. Go to sleep.
4. When the device wakes-up, resume the suspended channels.

Note: In Stand-by Sleep mode, the DMAC can only access RAM when it is not back biased (PM.STDBYCFG.BBIASxx = 0x0)

26.6.8. Synchronization

Not applicable.

© 2025 Microchip Technology Inc. and its subsidiaries

26.7. Register Summary

Se the the *DMAC* module in the *Product Memory Mapping Overview* from Related Links for the base address.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRL	7:0						CRCENABLE	DMAENABLE	SWRST
		15:8					LVLENx3	LVLENx2	LVLENx1	LVLENx0
0x02	CRCCTRL	7:0					CRCPOLY[1:0]		CRCBEATSIZE[1:0]	
		15:8	CRCMODE[1:0]		CRCSRC[5:0]					
0x04	CRCDATAIN	7:0	CRCDATAIN[7:0]							
		15:8	CRCDATAIN[15:8]							
		23:16	CRCDATAIN[23:16]							
		31:24	CRCDATAIN[31:24]							
0x08	CRCCHKSUM	7:0	CRCCHKSUM[7:0]							
		15:8	CRCCHKSUM[15:8]							
		23:16	CRCCHKSUM[23:16]							
		31:24	CRCCHKSUM[31:24]							
0x0C	CRCSTATUS	7:0						CRCERR	CRCZERO	CRCBUSY
0x0D	DBGCTRL	7:0								DBGRUN
0x0E ... 0x0F	Reserved									
0x10	SWTRIGCTRL	7:0	SWTRIGn[7:0]							
		15:8	SWTRIGn[15:8]							
		23:16								
		31:24								
0x14	PRICTRL0	7:0	RRLVLEN0	QOS00[1:0]				LVLPRI0[4:0]		
		15:8	RRLVLEN1	QOS01[1:0]				LVLPRI1[4:0]		
		23:16	RRLVLEN2	QOS02[1:0]				LVLPRI2[4:0]		
		31:24	RRLVLEN3	QOS03[1:0]				LVLPRI3[4:0]		
0x18 ... 0x1F	Reserved									
0x20	INTPEND	7:0						ID[4:0]		
		15:8	PEND	BUSY	FERR	CRCERR		SUSP	TCMPL	TERR
0x22 ... 0x23	Reserved									
0x24	INTSTATUS	7:0	CHINTn[7:0]							
		15:8	CHINTn[15:8]							
		23:16								
		31:24								
0x28	BUSYCH	7:0	BUSYCHn[7:0]							
		15:8	BUSYCHn[15:8]							
		23:16								
		31:24								
0x2C	PENDCH	7:0	PENDCHn[7:0]							
		15:8	PENDCHn[15:8]							
		23:16								
		31:24								
0x30	ACTIVE	7:0					LVLEXx3	LVLEXx2	LVLEXx1	LVLEXx0
		15:8	ABUSY			ID[4:0]				
		23:16	BTCNT[7:0]							
		31:24	BTCNT[15:8]							
0x34	BASEADDR	7:0	BASEADDR[7:0]							
		15:8	BASEADDR[15:8]							
		23:16	BASEADDR[23:16]							
		31:24	BASEADDR[31:24]							
0x38	WRBADDR	7:0	WRBADDR[7:0]							
		15:8	WRBADDR[15:8]							
		23:16	WRBADDR[23:16]							
		31:24	WRBADDR[31:24]							

© 2025 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x3C ... 0x3F	Reserved									
0x40	CHCTRLA0	7:0		RUNSTDBY					ENABLE	SWRST
		15:8	TRIGSRC[7:0]							
		23:16			TRIGACT[1:0]					
		31:24			THRESHOLD[1:0]		BURSTLEN[3:0]			
0x44	CHCTRLB0	7:0							CMD[1:0]	
0x45	CHPRILVL0	7:0							PRILVL[1:0]	
0x46	CHEVCTRL0	7:0	EVOE	EVIE	EVOMODE[1:0]			EVACT[2:0]		
0x47 ... 0x4B	Reserved									
0x4C	CHINTENCLR0	7:0						SUSP	TCMPL	TERR
0x4D	CHINTENSET0	7:0						SUSP	TCMPL	TERR
0x4E	CHINTFLAG0	7:0						SUSP	TCMPL	TERR
0x4F	CHSTATUS0	7:0					CRCERR	FERR	BUSY	PEND
0x50	CHCTRLA1	7:0		RUNSTDBY					ENABLE	SWRST
		15:8	TRIGSRC[7:0]							
		23:16			TRIGACT[1:0]					
		31:24			THRESHOLD[1:0]		BURSTLEN[3:0]			
0x54	CHCTRLB1	7:0							CMD[1:0]	
0x55	CHPRILVL1	7:0							PRILVL[1:0]	
0x56	CHEVCTRL1	7:0	EVOE	EVIE	EVOMODE[1:0]			EVACT[2:0]		
0x57 ... 0x5B	Reserved									
0x5C	CHINTENCLR1	7:0						SUSP	TCMPL	TERR
0x5D	CHINTENSET1	7:0						SUSP	TCMPL	TERR
0x5E	CHINTFLAG1	7:0						SUSP	TCMPL	TERR
0x5F	CHSTATUS1	7:0					CRCERR	FERR	BUSY	PEND
0x60	CHCTRLA2	7:0		RUNSTDBY					ENABLE	SWRST
		15:8	TRIGSRC[7:0]							
		23:16			TRIGACT[1:0]					
		31:24			THRESHOLD[1:0]		BURSTLEN[3:0]			
0x64	CHCTRLB2	7:0							CMD[1:0]	
0x65	CHPRILVL2	7:0							PRILVL[1:0]	
0x66	CHEVCTRL2	7:0	EVOE	EVIE	EVOMODE[1:0]			EVACT[2:0]		
0x67 ... 0x6B	Reserved									
0x6C	CHINTENCLR2	7:0						SUSP	TCMPL	TERR
0x6D	CHINTENSET2	7:0						SUSP	TCMPL	TERR
0x6E	CHINTFLAG2	7:0						SUSP	TCMPL	TERR
0x6F	CHSTATUS2	7:0					CRCERR	FERR	BUSY	PEND
0x70	CHCTRLA3	7:0		RUNSTDBY					ENABLE	SWRST
		15:8	TRIGSRC[7:0]							
		23:16			TRIGACT[1:0]					
		31:24			THRESHOLD[1:0]		BURSTLEN[3:0]			
0x74	CHCTRLB3	7:0							CMD[1:0]	
0x75	CHPRILVL3	7:0							PRILVL[1:0]	
0x76	CHEVCTRL3	7:0	EVOE	EVIE	EVOMODE[1:0]			EVACT[2:0]		
0x77 ... 0x7B	Reserved									
0x7C	CHINTENCLR3	7:0						SUSP	TCMPL	TERR
0x7D	CHINTENSET3	7:0						SUSP	TCMPL	TERR
0x7E	CHINTFLAG3	7:0						SUSP	TCMPL	TERR
0x7F	CHSTATUS3	7:0					CRCERR	FERR	BUSY	PEND

© 2025 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x80	CHCTRLA4	7:0		RUNSTDBY					ENABLE	SWRST
		15:8	TRIGSRC[7:0]							
		23:16			TRIGACT[1:0]					
		31:24			THRESHOLD[1:0]		BURSTLEN[3:0]			
0x84	CHCTRLB4	7:0							CMD[1:0]	
0x85	CHPRILVL4	7:0							PRILVL[1:0]	
0x86	CHEVCTRL4	7:0	EVOE	EVIE	EVOMODE[1:0]			EVACT[2:0]		
0x87 ... 0x8B	Reserved									
0x8C	CHINTENCLR4	7:0						SUSP	TCMPL	TERR
0x8D	CHINTENSET4	7:0						SUSP	TCMPL	TERR
0x8E	CHINTFLAG4	7:0						SUSP	TCMPL	TERR
0x8F	CHSTATUS4	7:0					CRCERR	FERR	BUSY	PEND
0x90	CHCTRLA5	7:0		RUNSTDBY					ENABLE	SWRST
		15:8	TRIGSRC[7:0]							
		23:16			TRIGACT[1:0]					
		31:24			THRESHOLD[1:0]		BURSTLEN[3:0]			
0x94	CHCTRLB5	7:0							CMD[1:0]	
0x95	CHPRILVL5	7:0							PRILVL[1:0]	
0x96	CHEVCTRL5	7:0	EVOE	EVIE	EVOMODE[1:0]			EVACT[2:0]		
0x97 ... 0x9B	Reserved									
0x9C	CHINTENCLR5	7:0						SUSP	TCMPL	TERR
0x9D	CHINTENSET5	7:0						SUSP	TCMPL	TERR
0x9E	CHINTFLAG5	7:0						SUSP	TCMPL	TERR
0x9F	CHSTATUS5	7:0					CRCERR	FERR	BUSY	PEND
0xA0	CHCTRLA6	7:0		RUNSTDBY					ENABLE	SWRST
		15:8	TRIGSRC[7:0]							
		23:16			TRIGACT[1:0]					
		31:24			THRESHOLD[1:0]		BURSTLEN[3:0]			
0xA4	CHCTRLB6	7:0							CMD[1:0]	
0xA5	CHPRILVL6	7:0							PRILVL[1:0]	
0xA6	CHEVCTRL6	7:0	EVOE	EVIE	EVOMODE[1:0]			EVACT[2:0]		
0xA7 ... 0xAB	Reserved									
0xAC	CHINTENCLR6	7:0						SUSP	TCMPL	TERR
0xAD	CHINTENSET6	7:0						SUSP	TCMPL	TERR
0xAE	CHINTFLAG6	7:0						SUSP	TCMPL	TERR
0xAF	CHSTATUS6	7:0					CRCERR	FERR	BUSY	PEND
0xB0	CHCTRLA7	7:0		RUNSTDBY					ENABLE	SWRST
		15:8	TRIGSRC[7:0]							
		23:16			TRIGACT[1:0]					
		31:24			THRESHOLD[1:0]		BURSTLEN[3:0]			
0xB4	CHCTRLB7	7:0							CMD[1:0]	
0xB5	CHPRILVL7	7:0							PRILVL[1:0]	
0xB6	CHEVCTRL7	7:0	EVOE	EVIE	EVOMODE[1:0]			EVACT[2:0]		
0xB7 ... 0xBB	Reserved									
0xBC	CHINTENCLR7	7:0						SUSP	TCMPL	TERR
0xBD	CHINTENSET7	7:0						SUSP	TCMPL	TERR
0xBE	CHINTFLAG7	7:0						SUSP	TCMPL	TERR
0xBF	CHSTATUS7	7:0					CRCERR	FERR	BUSY	PEND
0xC0	CHCTRLA8	7:0		RUNSTDBY					ENABLE	SWRST
		15:8	TRIGSRC[7:0]							
		23:16			TRIGACT[1:0]					
		31:24			THRESHOLD[1:0]		BURSTLEN[3:0]			
0xC4	CHCTRLB8	7:0							CMD[1:0]	

© 2025 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xC5	CHPRILVL8	7:0							PRILVL[1:0]	
0xC6	CHEVCTRL8	7:0	EVOE	EVIE	EVOMODE[1:0]			EVACT[2:0]		
0xC7 ... 0xCB	Reserved									
0xCC	CHINTENCLR8	7:0						SUSP	TCMPL	TERR
0xCD	CHINTENSET8	7:0						SUSP	TCMPL	TERR
0xCE	CHINTFLAG8	7:0						SUSP	TCMPL	TERR
0xCF	CHSTATUS8	7:0					CRCERR	FERR	BUSY	PEND
0xD0	CHCTRLA9	7:0		RUNSTDBY					ENABLE	SWRST
		15:8	TRIGSRC[7:0]							
		23:16			TRIGACT[1:0]					
		31:24			THRESHOLD[1:0]		BURSTLEN[3:0]			
0xD4	CHCTRLB9	7:0							CMD[1:0]	
0xD5	CHPRILVL9	7:0							PRILVL[1:0]	
0xD6	CHEVCTRL9	7:0	EVOE	EVIE	EVOMODE[1:0]			EVACT[2:0]		
0xD7 ... 0xDB	Reserved									
0xDC	CHINTENCLR9	7:0						SUSP	TCMPL	TERR
0xDD	CHINTENSET9	7:0						SUSP	TCMPL	TERR
0xDE	CHINTFLAG9	7:0						SUSP	TCMPL	TERR
0xDF	CHSTATUS9	7:0					CRCERR	FERR	BUSY	PEND
0xE0	CHCTRLA10	7:0		RUNSTDBY					ENABLE	SWRST
		15:8	TRIGSRC[7:0]							
		23:16			TRIGACT[1:0]					
		31:24			THRESHOLD[1:0]		BURSTLEN[3:0]			
0xE4	CHCTRLB10	7:0							CMD[1:0]	
0xE5	CHPRILVL10	7:0							PRILVL[1:0]	
0xE6	CHEVCTRL10	7:0	EVOE	EVIE	EVOMODE[1:0]			EVACT[2:0]		
0xE7 ... 0xEB	Reserved									
0xEC	CHINTENCLR10	7:0						SUSP	TCMPL	TERR
0xED	CHINTENSET10	7:0						SUSP	TCMPL	TERR
0xEE	CHINTFLAG10	7:0						SUSP	TCMPL	TERR
0xEF	CHSTATUS10	7:0					CRCERR	FERR	BUSY	PEND
0xF0	CHCTRLA11	7:0		RUNSTDBY					ENABLE	SWRST
		15:8	TRIGSRC[7:0]							
		23:16			TRIGACT[1:0]					
		31:24			THRESHOLD[1:0]		BURSTLEN[3:0]			
0xF4	CHCTRLB11	7:0							CMD[1:0]	
0xF5	CHPRILVL11	7:0							PRILVL[1:0]	
0xF6	CHEVCTRL11	7:0	EVOE	EVIE	EVOMODE[1:0]			EVACT[2:0]		
0xF7 ... 0xFB	Reserved									
0xFC	CHINTENCLR11	7:0						SUSP	TCMPL	TERR
0xFD	CHINTENSET11	7:0						SUSP	TCMPL	TERR
0xFE	CHINTFLAG11	7:0						SUSP	TCMPL	TERR
0xFF	CHSTATUS11	7:0					CRCERR	FERR	BUSY	PEND
0x0100	CHCTRLA12	7:0		RUNSTDBY					ENABLE	SWRST
		15:8	TRIGSRC[7:0]							
		23:16			TRIGACT[1:0]					
		31:24			THRESHOLD[1:0]		BURSTLEN[3:0]			
0x0104	CHCTRLB12	7:0							CMD[1:0]	
0x0105	CHPRILVL12	7:0							PRILVL[1:0]	
0x0106	CHEVCTRL12	7:0	EVOE	EVIE	EVOMODE[1:0]			EVACT[2:0]		
0x0107 ... 0x010B	Reserved									

© 2025 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x010C	CHINTENCLR12	7:0						SUSP	TCMPL	TERR
0x010D	CHINTENSET12	7:0						SUSP	TCMPL	TERR
0x010E	CHINTFLAG12	7:0						SUSP	TCMPL	TERR
0x010F	CHSTATUS12	7:0					CRCERR	FERR	BUSY	PEND
0x0110	CHCTRLA13	7:0		RUNSTDBY					ENABLE	SWRST
		15:8	TRIGSRC[7:0]							
		23:16			TRIGACT[1:0]					
		31:24			THRESHOLD[1:0]		BURSTLEN[3:0]			
0x0114	CHCTRLB13	7:0							CMD[1:0]	
0x0115	CHPRILVL13	7:0							PRILVL[1:0]	
0x0116	CHEVCTRL13	7:0	EVOE	EVIE	EVOMODE[1:0]			EVACT[2:0]		
0x0117 ... 0x011B	Reserved									
0x011C	CHINTENCLR13	7:0						SUSP	TCMPL	TERR
0x011D	CHINTENSET13	7:0						SUSP	TCMPL	TERR
0x011E	CHINTFLAG13	7:0						SUSP	TCMPL	TERR
0x011F	CHSTATUS13	7:0					CRCERR	FERR	BUSY	PEND
0x0120	CHCTRLA14	7:0		RUNSTDBY					ENABLE	SWRST
		15:8	TRIGSRC[7:0]							
		23:16			TRIGACT[1:0]					
		31:24			THRESHOLD[1:0]		BURSTLEN[3:0]			
0x0124	CHCTRLB14	7:0							CMD[1:0]	
0x0125	CHPRILVL14	7:0							PRILVL[1:0]	
0x0126	CHEVCTRL14	7:0	EVOE	EVIE	EVOMODE[1:0]			EVACT[2:0]		
0x0127 ... 0x012B	Reserved									
0x012C	CHINTENCLR14	7:0						SUSP	TCMPL	TERR
0x012D	CHINTENSET14	7:0						SUSP	TCMPL	TERR
0x012E	CHINTFLAG14	7:0						SUSP	TCMPL	TERR
0x012F	CHSTATUS14	7:0					CRCERR	FERR	BUSY	PEND
0x0130	CHCTRLA15	7:0		RUNSTDBY					ENABLE	SWRST
		15:8	TRIGSRC[7:0]							
		23:16			TRIGACT[1:0]					
		31:24			THRESHOLD[1:0]		BURSTLEN[3:0]			
0x0134	CHCTRLB15	7:0							CMD[1:0]	
0x0135	CHPRILVL15	7:0							PRILVL[1:0]	
0x0136	CHEVCTRL15	7:0	EVOE	EVIE	EVOMODE[1:0]			EVACT[2:0]		
0x0137 ... 0x013B	Reserved									
0x013C	CHINTENCLR15	7:0						SUSP	TCMPL	TERR
0x013D	CHINTENSET15	7:0						SUSP	TCMPL	TERR
0x013E	CHINTFLAG15	7:0						SUSP	TCMPL	TERR
0x013F	CHSTATUS15	7:0					CRCERR	FERR	BUSY	PEND

Related Links

Product Memory Mapping Overview

26.8. Register Description

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

Some registers are optionally write-protected by the PAC. Optional PAC write protection is denoted by the PAC Write-Protection property in each individual register description. See *Register Access Protection* from Related Links.

© 2025 Microchip Technology Inc. and its subsidiaries

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the Enable-Protected property in each individual register description.

Related Links

Register Access Protection

© 2025 Microchip Technology Inc. and its subsidiaries

26.8.1. Control

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

Bit	15	14	13	12	11	10	9	8
					LVLENx3	LVLENx2	LVLENx1	LVLENx0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
						CRCENABLE	DMAENABLE	SWRST
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 8, 9, 10, 11 – LVLENxx Priority Level x Enable
When this bit is set, all requests with the corresponding level will be fed into the arbiter block. When cleared, all requests with the corresponding level will be ignored.
For details on arbitration schemes, see *Arbitration* from Related Links.
These bits are not enable-protected.

Value	Description
0	Transfer requests for Priority level x will not be handled.
1	Transfer requests for Priority level x will be handled.

Bit 2 – CRCENABLE CRC Enable
Writing a '0' to this bit will disable the CRC calculation when the CRC Status Busy flag is cleared (CRCSTATUS. CRCBUSY). The bit is zero when the CRC is disabled.
Writing a '1' to this bit will enable the CRC calculation.

Value	Description
0	The CRC calculation is disabled.
1	The CRC calculation is enabled.

Bit 1 – DMAENABLE DMA Enable
Setting this bit will enable the DMA module.
Writing a '0' to this bit will disable the DMA module. When writing a '0' during an ongoing transfer, the bit will not be cleared until the internal data transfer buffer is empty and the DMA transfer is aborted. The internal data transfer buffer will be empty once the ongoing burst transfer is completed.
This bit is not enable-protected.

Value	Description
0	The peripheral is disabled.
1	The peripheral is enabled.

Bit 0 – SWRST Software Reset
Writing a '0' to this bit has no effect.
Writing a '1' to this bit when both the DMAC and the CRC module are disabled (DMAENABLE and CRCENABLE are '0') resets all registers in the DMAC (except DBGCTRL) to their initial state. If either the DMAC or CRC module is enabled, the Reset request will be ignored and the DMAC will return an access error.

© 2025 Microchip Technology Inc. and its subsidiaries

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

Related Links

Arbitration

© 2025 Microchip Technology Inc. and its subsidiaries

26.8.2. CRC Control

Name: CRCCTRL
Offset: 0x02
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
	CRCMODE[1:0]		CRCSRC[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

Bits 15:14 – CRCMODE[1:0] CRC Operating Mode
These bits define the block transfer mode.

Value	Name	Description
0x0	DEFAULT	Default operating mode
0x1	—	Reserved
0x2	CRCMON	Memory CRC monitor operating mode
0x3	CRCGEN	Memory CRC generation operating mode

Bits 13:8 – CRCSRC[5:0] CRC Input Source
These bits select the input source for generating the CRC. The selected source is locked until either the CRC generation is completed or the CRC module is disabled. This means the CRCSRC cannot be modified when the CRC operation is ongoing. The lock is signaled by the CRCBUSY status bit. CRC generation complete is generated and signaled from the selected source when used with the DMA channel.

Value	Name	Description
0x00	NOACT	No action
0x01	IO	I/O interface
0x02 - 0x1F	—	Reserved
0x20	CH0	DMA channel 0
0x21	CH1	DMA channel 1
0x22	CH2	DMA channel 2
0x23	CH3	DMA channel 3
0x24	CH4	DMA channel 4
0x25	CH5	DMA channel 5
0x26	CH6	DMA channel 6
0x27	CH7	DMA channel 7
0x28	CH8	DMA channel 8
0x29	CH9	DMA channel 9
0x2A	CH10	DMA channel 10
0x2B	CH11	DMA channel 11
0x2C	CH12	DMA channel 12
0x2D	CH13	DMA channel 13
0x2E	CH14	DMA channel 14
0x2F	CH15	DMA channel 15

© 2025 Microchip Technology Inc. and its subsidiaries

Bits 3:2 – CRCPOLY[1:0] CRC Polynomial Type
These bits select the CRC polynomial type.

Value	Name	Description
0x0	CRC16	CRC-16 (CRC-CCITT)
0x1	CRC32	CRC32 (IEEE 802.3)
0x2-0x3	—	Reserved

Bits 1:0 – CRCBEATSIZE[1:0] CRC Beat Size
These bits define the size of the data transfer for each bus access when the CRC is used with I/O interface.

Value	Name	Description
0x0	BYTE	8-bit bus transfer
0x1	HWORD	16-bit bus transfer
0x2	WORD	32-bit bus transfer
0x3	—	Reserved

© 2025 Microchip Technology Inc. and its subsidiaries

26.8.3. CRC Data Input

Name: CRCDATAIN
Offset: 0x04
Reset: 0x00000000
Property: PAC Write Protection

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

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

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

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

Bits 31:0 – CRCDATAIN[31:0] CRC Data Input

These bits store the data for which the CRC checksum is computed. A new CRC checksum is ready (CRCBEAT+ 1) clock cycles after the CRCDATAIN register is written.

© 2025 Microchip Technology Inc. and its subsidiaries

26.8.4. CRC Checksum

Name: CRCCHKSUM
Offset: 0x08
Reset: 0x00000000
Property: PAC Write Protection, Enable-Protected

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.

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

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

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

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

Bits 31:0 – CRCCHKSUM[31:0] CRC Checksum
These bits store the generated CRC result. The 16 MSB bits are always read zero when CRC-16 is enabled.

© 2025 Microchip Technology Inc. and its subsidiaries

26.8.5. CRC Status

Name: CRCSTATUS
Offset: 0x0C
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
						CRCERR	CRCZERO	CRCBUSY
Access						R	R	R/W
Reset						0	0	0

Bit 2 – CRCERR CRC Error
This bit is read '1' when the memory CRC monitor detects data corruption.

Bit 1 – CRCZERO CRC Zero
This bit is cleared when a new CRC source is selected.
This bit is set when the CRC generation is complete and the CRC Checksum is zero.

Bit 0 – CRCBUSY CRC Module Busy
When used with an I/O interface (CRCCTRL.CRCSRC=0x1):

- This bit is cleared by writing a '1' to it
- This bit is set when the CRC Data Input (CRCDATAIN) register is written
- Writing a '1' to this bit will clear the CRC Module Busy bit
- Writing a '0' to this bit has no effect

When used with a DMA channel (CRCCTRL.CRCSRC=0x20..,0x3F):

- This bit is cleared when the corresponding DMA channel is disabled
- This bit is set when the corresponding DMA channel is enabled
- Writing a '1' to this bit has no effect
- Writing a '0' to this bit has no effect

© 2025 Microchip Technology Inc. and its subsidiaries

26.8.6. Debug Control

Name: DBGCTRL
Offset: 0x0D
Reset: 0x00
Property: PAC Write Protection

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

Bit 0 – DBGRUN Debug Run
This bit is not reset by a Software Reset.
This bit controls the functionality when the CPU is halted by an external debugger.

Value	Description
0	The DMAC is halted when the CPU is halted by an external debugger.
1	The DMAC continues normal operation when the CPU is halted by an external debugger.

© 2025 Microchip Technology Inc. and its subsidiaries

26.8.7. Software Trigger Control

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

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	SWTRIGn[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
	SWTRIGn[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 – SWTRIGn[15:0] Channel n Software Trigger [n = 15..0]
This bit is cleared when the Channel Pending bit in the Channel Status register (CHSTATUS.PEND) for the corresponding channel is either set or by writing a '1' to it.
This bit is set if CHSTATUS.PEND is already '1' when writing a '1' to that bit.
Writing a '0' to this bit clears the bit.
Writing a '1' to this bit generates a DMA software trigger on channel n, if CHSTATUS.PEND = 0 for channel n. CHSTATUS.PEND will be set and SWTRIGn remains cleared.

© 2025 Microchip Technology Inc. and its subsidiaries

26.8.8. Priority Control 0

Name: PRICTRL0
Offset: 0x14
Reset: 0x40404040
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	RRLVLEN3	QOS03[1:0]		LVLPRI3[4:0]				
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RRLVLEN2	QOS02[1:0]		LVLPRI2[4:0]				
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RRLVLEN1	QOS01[1:0]		LVLPRI1[4:0]				
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RRLVLEN0	QOS00[1:0]		LVLPRI0[4:0]				
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	0	0	0	0	0

Bits 7, 15, 23, 31 – RRLVLEN Level Round-Robin Scheduling Enable
For details on arbitration schemes, see *Arbitration* from Related Links.

Value	Description
0	Static arbitration scheme for channels with level 0 priority.
1	Round-robin arbitration scheme for channels with level 0 priority.

Bits 5:6, 13:14, 21:22, 29:30 – QOS Level Quality of Service

0x0	DISABLE Background (no sensitive operation)
0x1	LOW Sensitive to bandwidth
0x2	MEDIUM Sensitive to latency
0x3	Critical Latency

Bits 0:4, 8:12, 16:20, 24:28 – LVLPRI Level Channel Priority Number
When round-robin arbitration is enabled (PRICTRL0.RRLVLEN0=1) for priority level 0, this register holds the channel number of the last DMA channel being granted access as the active channel with priority level 0.
When static arbitration is enabled (PRICTRL0.RRLVLEN0=0) for priority level 0, and the value of this bit group is non-zero, it will not affect the static priority scheme.
This bit group is not reset when round-robin arbitration gets disabled (PRICTRL0.RRLVLEN0 written to '0').

Related Links
Arbitration

© 2025 Microchip Technology Inc. and its subsidiaries

26.8.9. Interrupt Pending

Name: INTPEND
Offset: 0x20
Reset: 0x0000
Property: -

This register allows the user to identify the lowest DMA channel with pending interrupt.
An interrupt that handles several channels must consult the INTPEND register to find out which channel number has priority (ignoring/filtering each channel that has its own interrupt line). An interrupt dedicated to only one channel must not use the INTPEND register.

Bit	15	14	13	12	11	10	9	8
	PEND	BUSY	FERR	CRCERR		SUSP	TCMPL	TERR
Access	R	R	R	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

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

Bit 15 – PEND Pending
This bit reads '1' when the channel selected by Channel ID field (ID) is pending.

Bit 14 – BUSY Busy
This bit reads '1' when the channel selected by Channel ID field (ID) is busy.

Bit 13 – FERR Fetch Error
This bit reads '1' when the channel selected by Channel ID field (ID) fetched an invalid descriptor.

Bit 12 – CRCERR CRC Error
This bit reads '1' when the channel selected by the Channel ID field (ID) has a CRC Error Status Flag bit set and is set when the CRC monitor detects data corruption.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears it. It also clears the corresponding flag in the Channel n Interrupt Flag Status and Clear register (CHINTFLAGn), where n is determined by the Channel ID bit field (ID).

Bit 10 – SUSP Channel Suspend
This bit reads '1' when the channel selected by the Channel ID field (ID) has a pending Suspend interrupt.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears it. It also clears the corresponding flag in the Channel n Interrupt Flag Status and Clear register (CHINTFLAGn), where n is determined by the Channel ID bit field (ID).

Bit 9 – TCMPL Transfer Complete
This bit reads '1' when the channel selected by Channel ID field (ID) has a pending Transfer Complete interrupt.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears it. It also clears the corresponding flag in the Channel n Interrupt Flag Status and Clear register (CHINTFLAGn), where n is determined by the Channel ID bit field (ID).

© 2025 Microchip Technology Inc. and its subsidiaries

Bit 8 – TERR Transfer Error

This bit reads '1' when the channel selected by the Channel ID field (ID) has a pending Transfer Error interrupt.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears it. It also clears the corresponding flag in the Channel n Interrupt Flag Status and Clear register (CHINTFLAGn), where n is determined by the Channel ID bit field (ID).

Bits 4:0 – ID[4:0] Channel ID

These bits store the lowest channel number with pending interrupts. The number is valid if Suspend (SUSP), Transfer Complete (TCMPL) or Transfer Error (TERR) bits are set. The Channel ID field is refreshed when a new channel (with a channel number less than the current one) with pending interrupts is detected or when the application clears the corresponding channel interrupt sources. When no pending channel interrupts are available, these bits always return a zero value when read. When the bits are written, indirect access to the corresponding Channel Interrupt Flag register is enabled.

© 2025 Microchip Technology Inc. and its subsidiaries

26.8.10. Interrupt Status

Name: INTSTATUS
Offset: 0x24
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

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

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

Bits 15:0 – CHINTn[15:0] Channel n Pending Interrupt [n=15..0]
This bit is set when Channel n has a pending interrupt/the interrupt request is received.
This bit is cleared when the corresponding Channel n interrupts are disabled or the interrupts sources are cleared.

© 2025 Microchip Technology Inc. and its subsidiaries

26.8.11. Busy Channels

Name: BUSYCH
Offset: 0x28
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

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

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

Bits 15:0 – BUSYCHn[15:0] Busy Channel n [n=15..0]
This bit is cleared when the channel trigger action for DMA channel n is complete, when a bus error for DMA channel n is detected, or when DMA channel n is disabled.
This bit is set when DMA channel n starts a DMA transfer.

© 2025 Microchip Technology Inc. and its subsidiaries

26.8.12. Pending Channels

Name: PENDCH
Offset: 0x2C
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

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

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

Bits 15:0 – PENDCHn[15:0] Pending Channel n [n=0..15]
This bit is cleared when a trigger execution defined by channel trigger action settings for DMA channel n is started, when a bus error for DMA channel n is detected or when DMA channel n is disabled. For details on trigger action settings, refer to CHCTRLB.TRIGACT.
This bit is set when a transfer is pending on DMA channel n.

© 2025 Microchip Technology Inc. and its subsidiaries

26.8.13. Active Channel and Levels

Name: ACTIVE
Offset: 0x30
Reset: 0x00000000
Property: -

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

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

Bit	15	14	13	12	11	10	9	8
	ABUSY			ID[4:0]				
Access	R			R	R	R	R	R
Reset	0			0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
					LVLEXx3	LVLEXx2	LVLEXx1	LVLEXx0
Access					R	R	R	R
Reset					0	0	0	0

Bits 31:16 – BTCNT[15:0] Active Channel Block Transfer Count
These bits hold the 16-bit block transfer count of the ongoing transfer. This value is stored in the active channel and written back in the corresponding Write-Back channel memory location when the arbiter grants a new channel access. The value is valid only when the active channel Active Busy flag (ABUSY) is set.

Bit 15 – ABUSY Active Channel Busy
This bit is cleared when the active transfer count is written back in the write-back memory section.
This bit is set when the next descriptor transfer count is read from the write-back memory section.

Bits 12:8 – ID[4:0] Active Channel ID
These bits hold the channel index currently stored in the active channel registers. The value is updated each time the arbiter grants a new channel transfer access request.

Bits 0, 1, 2, 3 – LVLEXxx Level x Channel Trigger Request Executing [x=3..0]
This bit is set when a level-x channel trigger request is executing or pending.
This bit is cleared when no request is pending or being executed.

© 2025 Microchip Technology Inc. and its subsidiaries

26.8.14. Descriptor Memory Section Base Address

Name: BASEADDR
Offset: 0x34
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

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

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

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

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

Bits 31:0 – BASEADDR[31:0] Descriptor Memory Base Address
These bits store the Descriptor memory section base address. The value must be 128-bit aligned.

Bits 31:0 – BASEADDR[31:0] Descriptor Memory Base Address
These bits store the Descriptor memory section base address. The value must be 64-bit aligned.

© 2025 Microchip Technology Inc. and its subsidiaries

26.8.15. Write-Back Memory Section Base Address

Name: WRBADDR
Offset: 0x38
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

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

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

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

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

Bits 31:0 – WRBADDR[31:0] Write-Back Memory Base Address
These bits store the Write-Back memory base address. The value must be 128-bit aligned.

Bits 31:0 – WRBADDR[31:0] Write-Back Memory Base Address
These bits store the Write-Back memory base address. The value must be 64-bit aligned.

© 2025 Microchip Technology Inc. and its subsidiaries

26.8.16. Channel Control A

Name: CHCTRLA
Offset: 0x40 + n*0x10 [n=0..15]
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
			THRESHOLD[1:0]		BURSTLEN[3:0]			
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
			TRIGACT[1:0]					
Access			R/W	R/W				
Reset			0	0				

Bit	15	14	13	12	11	10	9	8
	TRIGSRC[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		RUNSTDBY					ENABLE	SWRST
Access		R/W					R/W	R/W
Reset		0					0	0

Bits 29:28 – THRESHOLD[1:0] FIFO Threshold
These bits define the threshold from where the DMA starts to write to the destination. These bits have no effect in the case of single beat transfers.
These bits are not enable-protected.

Value	Name	Description
0x0	1BEAT	Destination write starts after each beat source addess read
0x1	2BEATS	Destination write starts after 2-beats source address read
0x2	4BEATS	Destination write starts after 4-beats source address read
0x3	8BEATS	Destination write starts after 8-beats source address read

Bits 27:24 – BURSTLEN[3:0] Burst Length
These bits define the burst mode.
These bits are not enable-protected.

Value	Name	Description
0x0	SINGLE	Single-beat burst
0x1	2BEAT	2-beats burst length
0x2	3BEAT	3-beats burst length
0x3	4BEAT	4-beats burst length
0x4	5BEAT	5-beats burst length
0x5	6BEAT	6-beats burst length
0x6	7BEAT	7-beats burst length
0x7	8BEAT	8-beats burst length
0x8	9BEAT	9-beats burst length
0x9	10BEAT	10-beats burst length
0xA	11BEAT	11-beats burst length

© 2025 Microchip Technology Inc. and its subsidiaries

Value	Name	Description
0xB	12BEAT	12-beats burst length
0xC	13BEAT	13-beats burst length
0xD	14BEAT	14-beats burst length
0xE	15BEAT	15-beats burst length
0xF	16BEAT	16-beats burst length

Bits 21:20 – TRIGACT[1:0] Trigger Action

These bits define the trigger action used for a transfer.
These bits are not enable-protected.

Value	Name	Description
0x0	BLOCK	One trigger required for each block transfer
0x1	—	Reserved
0x2	BURST	One trigger required for each burst transfer
0x3	TRANSACTION	One trigger required for each transaction

Bits 15:8 – TRIGSRC[7:0] Trigger Source

These bits define the peripheral that will be the source of a trigger.

Table 26-2. Triggers Map

TRIGSRC[7:0]	Trigger
0	DISABLE; Only software/event triggers
1	RTC_DMAC_ID_TIMESTAMP
2	DSU_DMAC_ID_DCC0
3	DSU_DMAC_ID_DCC1
4	SERCOM0_DMAC_ID_RX
5	SERCOM0_DMAC_ID_TX
6	SERCOM1_DMAC_ID_RX
7	SERCOM1_DMAC_ID_TX
8	SERCOM2_DMAC_ID_RX
9	SERCOM2_DMAC_ID_TX
10	SERCOM3_DMAC_ID_RX
11	SERCOM3_DMAC_ID_TX
12	SERCOM4_DMAC_ID_RX
13	SERCOM4_DMAC_ID_TX
14	SERCOM5_DMAC_ID_RX
15	SERCOM5_DMAC_ID_TX
16	SERCOM6_DMAC_ID_RX
17	SERCOM6_DMAC_ID_TX
18	TCC0_DMAC_ID_OVF
19	TCC0_DMAC_ID_MC_0
20	TCC0_DMAC_ID_MC_1
21	TCC0_DMAC_ID_MC_2
22	TCC0_DMAC_ID_MC_3
23	TCC0_DMAC_ID_MC_4
24	TCC0_DMAC_ID_MC_5
25	TCC1_DMAC_ID_OVF
26	TCC1_DMAC_ID_MC_0
27	TCC1_DMAC_ID_MC_1
28	TCC1_DMAC_ID_MC_2
29	TCC1_DMAC_ID_MC_3
30	TCC1_DMAC_ID_MC_4
31	TCC1_DMAC_ID_MC_5
32	TCC2_DMAC_ID_OVF

© 2025 Microchip Technology Inc. and its subsidiaries

Table 26-2. Triggers Map (continued)

TRIGSRC[7:0]	Trigger
33	TCC2_DMAC_ID_MC_0
34	TCC2_DMAC_ID_MC_1
35	TC0_DMAC_ID_OVF
36	TC0_DMAC_ID_MC_0
37	TC0_DMAC_ID_MC_1
38	TC1_DMAC_ID_OVF
39	TC1_DMAC_ID_MC_0
40	TC1_DMAC_ID_MC_1
41	TC2_DMAC_ID_OVF
42	TC2_DMAC_ID_MC_0
43	TC2_DMAC_ID_MC_1
44	TC3_DMAC_ID_OVF
45	TC3_DMAC_ID_MC_0
46	TC3_DMAC_ID_MC_1
47	TC4_DMAC_ID_OVF
48	TC4_DMAC_ID_MC_0
49	TC4_DMAC_ID_MC_1
50	TC5_DMAC_ID_OVF
51	TC5_DMAC_ID_MC_0
52	TC5_DMAC_ID_MC_1
53	TC6_DMAC_ID_OVF
54	TC6_DMAC_ID_MC_0
55	TC6_DMAC_ID_MC_1
56	TC7_DMAC_ID_OVF
57	TC7_DMAC_ID_MC_0
58	TC7_DMAC_ID_MC_1
59	TC8_DMAC_ID_OVF
60	TC8_DMAC_ID_MC_0
61	TC8_DMAC_ID_MC_1
62	TC9_DMAC_ID_OVF
63	TC9_DMAC_ID_MC_0
64	TC9_DMAC_ID_MC_1
65	QSPI_DMAC_ID_RX
66	QSPI_DMAC_ID_TX
67	CAN0_DMAC_ID_DBGMSG
68	CAN1_DMAC_ID_DBGMSG

Bit 6 – RUNSTDBY Channel run in standby

This bit is used to keep the DMAC channel running in Idle and Standby Sleep mode.
This bit is not enable-protected.

Value	Description
0	The DMAC channel is halted in Idle and Standby Sleep mode.
1	The DMAC channel continues to run in Idle and Standby Sleep mode.

Bit 1 – ENABLE Channel Enable

When writing a '0' to this bit during an ongoing transfer, the bit must not be cleared until the internal data transfer buffer is empty and the DMA transfer is aborted. The internal data transfer buffer is empty when the ongoing burst transfer is completed.
Writing a '1' to this bit enables the DMA channel.
This bit is not enable-protected.

© 2025 Microchip Technology Inc. and its subsidiaries

Value	Description
0	DMA channel is disabled.
1	DMA channel is enabled.

Bit 0 – SWRST Channel Software Reset

Writing a '0' to this bit has no effect.
Writing a '1' to this bit resets the channel registers to their initial state. The bit can be set when the channel is disabled (ENABLE=0). Writing a '1' to this bit is ignored as long as ENABLE=1. This bit is automatically cleared when the reset is completed.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

© 2025 Microchip Technology Inc. and its subsidiaries

26.8.17. Channel Control B

Name: CHCTRLB
Offset: 0x44 + n*0x10 [n=0..15]
Reset: 0x00
Property: PAC Write-Protection

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

Bits 1:0 – CMD[1:0] Software Command
These bits define the software commands. See *Channel Suspend* and *Channel Resume and Next Suspend Skip* from Related Links.
These bits are not enable-protected.

CMD[1:0]	Name	Description
0x0	NOACT	No action
0x1	SUSPEND	Channel suspend operation
0x2	RESUME	Channel resume operation
0x3	-	Reserved

Related Links
Channel Resume and Next Suspend Skip
Channel Suspend

© 2025 Microchip Technology Inc. and its subsidiaries

26.8.18. Channel Priority Level

Name: CHPRILVL
Offset: 0x45 + n*0x10 [n=0..15]
Reset: 0x00
Property: PAC Write-Protection

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

Bits 1:0 – PRILVL[1:0] Channel Priority Level
These bits define the priority level used for the DMA channel. The available levels are shown below, where a high level has priority over a low level. These bits are not enable-protected.

Value	Name	Description
0x0	LVL0	Channel Priority Level 0 (Lowest Level)
0x1	LVL1	Channel Priority Level 1
0x2	LVL2	Channel Priority Level 2
0x3	LVL3	Channel Priority Level 3 (Highest Level)

© 2025 Microchip Technology Inc. and its subsidiaries

26.8.19. Channel Event Control

Name: CHEVCTRL
Offset: 0x46 + n*0x10 [n=0..15]
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
	EVOE	EVIE	EVOMODE[1:0]			EVACT[2:0]		
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bit 7 – EVOE Channel Event Output Enable
This bit indicates if the Channel event generation is enabled. The event is generated for every condition defined in the Channel Event Output Selection bits (CHEVCTRL.EVOMODE).

Value	Description
0	Channel event generation is disabled.
1	Channel event generation is enabled.

Bit 6 – EVIE Channel Event Input Enable

Value	Description
0	Channel event action will not be executed on any incoming event.
1	Channel event action will be executed on any incoming event.

Bits 5:4 – EVOMODE[1:0] Channel Event Output Mode
These bits define the channel event output selection. For more details on event output generation, see *Event Output Selection* from Related Links.

Value	Name	Description
0x0	DEFAULT	Block event output selection. See BTCTRL.EVOSEL for available selections.
0x1	TRIGACT	Ongoing trigger action
0x2-0x3	—	Reserved

Bits 2:0 – EVACT[2:0] Channel Event Input Action
These bits define the event input action. The action is executed only if the corresponding EVIE bit in the CHEVCTRL register of the channel is set. For more details on event actions, see *Event Input Actions* from Related Links. These bits are available only for channels with event input support.

Value	Name	Description
0x0	NOACT	No action
0x1	TRIG	Transfer and periodic transfer trigger
0x2	CTRIG	Conditional transfer trigger
0x3	CBLOCK	Conditional block transfer
0x4	SUSPEND	Channel suspend operation
0x5	RESUME	Channel resume operation
0x6	SSKIP	Skip next block suspend action
0x7	INCPRI	Increase priority

Related Links
Event Input Actions
Event Output Selection

© 2025 Microchip Technology Inc. and its subsidiaries

26.8.20. Channel Interrupt Enable Clear

Name: CHINTENCLR
Offset: 0x4C + n*0x10 [n=0..15]
Reset: 0x00
Property: PAC Write-Protection

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

Bit	7	6	5	4	3	2	1	0
						SUSP	TCMPL	TERR
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – SUSP Channel Suspend Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Channel Suspend Interrupt Enable bit, which disables the Channel Suspend interrupt.

Value	Description
0	The Channel Suspend interrupt is disabled.
1	The Channel Suspend interrupt is enabled.

Bit 1 – TCMPL Channel Transfer Complete Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Channel Transfer Complete Interrupt Enable bit, which disables the Channel Transfer Complete interrupt.

Value	Description
0	The Channel Transfer Complete interrupt is disabled. When block action is set to none, the TCMPL flag will not be set when a block transfer is completed.
1	The Channel Transfer Complete interrupt is enabled.

Bit 0 – TERR Channel Transfer Error Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Channel Transfer Error Interrupt Enable bit, which disables the Channel Transfer Error interrupt.

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

© 2025 Microchip Technology Inc. and its subsidiaries

26.8.21. Channel Interrupt Enable Set

Name: CHINTENSET
Offset: 0x4D + n*0x10 [n=0..15]
Reset: 0x00
Property: PAC Write-Protection

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

Bit	7	6	5	4	3	2	1	0
						SUSP	TCMPL	TERR
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – SUSP Channel Suspend Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Channel Suspend Interrupt Enable bit, which enables the Channel Suspend interrupt.

Value	Description
0	The Channel Suspend interrupt is disabled.
1	The Channel Suspend interrupt is enabled.

Bit 1 – TCMPL Channel Transfer Complete Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Channel Transfer Complete Interrupt Enable bit, which enables the Channel Transfer Complete interrupt.

Value	Description
0	The Channel Transfer Complete interrupt is disabled.
1	The Channel Transfer Complete interrupt is enabled.

Bit 0 – TERR Channel Transfer Error Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Channel Transfer Error Interrupt Enable bit, which enables the Channel Transfer Error interrupt.

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

© 2025 Microchip Technology Inc. and its subsidiaries

26.8.22. Channel Interrupt Flag Status and Clear

Name: CHINTFLAG
Offset: 0x4E + n*0x10 [n=0..15]
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
						SUSP	TCMPL	TERR
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – SUSP Channel Suspend
This flag is cleared by writing a '1' to it.
This flag is set when a block transfer with suspend block action is completed, when a software suspend command is executed, when a suspend event is received or when an invalid descriptor is fetched by the DMA.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Channel Suspend interrupt flag for the corresponding channel.
For details on available software commands, see *CHCTRLB* in the *DMAC Register Summary* from Related Links.
For details on available event input actions, see *CHCTRLB* in the *DMAC Register Summary* from Related Links.
For details on available block actions, see *BTCTRL* in the *DMAC Register Summary (SRAM)* from Related Links.

Bit 1 – TCMPL Channel Transfer Complete
This flag is cleared by writing a '1' to it.
This flag is set when a block transfer is completed and the corresponding interrupt block action is enabled.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Transfer Complete interrupt flag for the corresponding channel.

Bit 0 – TERR Channel Transfer Error
This flag is cleared by writing a '1' to it.
This flag is set when a bus error is detected during a beat transfer or when the DMAC fetches an invalid descriptor.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Transfer Error interrupt flag for the corresponding channel.

Related Links
Register Summary
Register Summary - SRAM

© 2025 Microchip Technology Inc. and its subsidiaries

26.8.23. Channel Status

Name: CHSTATUS
Offset: 0x4F + n*0x10 [n=0..15]
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
					CRCERR	FERR	BUSY	PEND
Access					R/W	R	R	R
Reset					0	0	0	0

Bit 3 – CRCERR Channel CRC Error
This bit is set when the CRC monitor detects data corruption. This bit is cleared by writing '1' to it, or by clearing the CRC Error bit in the INTPEND register (INTPEND.CRCERR). See *INTPEND* in the *DMAC Register Summary* from Related Links.

Bit 2 – FERR Channel Fetch Error
This bit is cleared when a software resume command is executed.
This bit is set when an invalid descriptor is fetched.

Bit 1 – BUSY Channel Busy
This bit is cleared when the channel trigger action is completed, when a bus error is detected or when the channel is disabled.
This bit is set when the DMA channel starts a DMA transfer.

Bit 0 – PEND Channel Pending
This bit is cleared when the channel trigger action is started, when a bus error is detected or when the channel is disabled. For details on trigger action settings, see *CHCTRLB* in the *DMAC Register Summary* from Related Links.
This bit is set when a transfer is pending on the DMA channel, as soon as the transfer request is received.

Related Links
Register Summary
Register Summary - SRAM

© 2025 Microchip Technology Inc. and its subsidiaries

26.9. Register Summary - SRAM

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	BTCTRL	7:0				BLOCKACT[1:0]		EVOSEL[1:0]		VALID
		15:8	STEPSIZE[2:0]			STEPSEL	DSTINC	SRCINC	BEATSIZE[1:0]	
0x02	BTCNT	7:0	BTCNT[7:0]							
		15:8	BTCNT[15:8]							
0x04	SRCADDR	7:0	SRCADDR[7:0]							
		15:8	SRCADDR[15:8]							
		23:16	SRCADDR[23:16]							
		31:24	SRCADDR[31:24]							
0x08	DSTADDR	7:0	DSTADDR[7:0]							
		15:8	DSTADDR[15:8]							
		23:16	DSTADDR[23:16]							
		31:24	DSTADDR[31:24]							
0x0C	DESCADDR	7:0	DESCADDR[7:0]							
		15:8	DESCADDR[15:8]							
		23:16	DESCADDR[23:16]							
		31:24	DESCADDR[31:24]							

26.10. Register Description - SRAM

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

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

Some registers are enable protected, meaning they can only be written when the peripheral is disabled. Enable protection is denoted by the “Enable-Protected” property in each individual register description.

© 2025 Microchip Technology Inc. and its subsidiaries

26.10.1. Block Transfer Control

Name: BTCTRL
Offset: 0x00
Reset: 0x0000
Property: -

The BTCTRL register offset is relative to (BASEADDR or WRBADDR) + Channel Number * 0x10

Bit	15	14	13	12	11	10	9	8
	STEPSIZE[2:0]			STEPSEL	DSTINC	SRCINC	BEATSIZE[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

Bits 15:13 – STEPSIZE[2:0] Address Increment Step Size
These bits select the address increment step size. The setting apply to source or destination address, depending on STEPSEL setting.

Value	Name	Description
0x0	X1	Next ADDR = ADDR + (Beat size in byte) * 1
0x1	X2	Next ADDR = ADDR + (Beat size in byte) * 2
0x2	X4	Next ADDR = ADDR + (Beat size in byte) * 4
0x3	X8	Next ADDR = ADDR + (Beat size in byte) * 8
0x4	X16	Next ADDR = ADDR + (Beat size in byte) * 16
0x5	X32	Next ADDR = ADDR + (Beat size in byte) * 32
0x6	X64	Next ADDR = ADDR + (Beat size in byte) * 64
0x7	X128	Next ADDR = ADDR + (Beat size in byte) * 128

Bit 12 – STEPSEL Step Selection
This bit selects if source or destination addresses are using the step size settings.

Value	Name	Description
0x0	DST	Step size settings apply to the destination address
0x1	SRC	Step size settings apply to the source address

Bit 11 – DSTINC Destination Address Increment Enable
Writing a '0' to this bit will disable the destination address incrementation. The address will be kept fixed during the data transfer.
Writing a '1' to this bit will enable the destination address incrementation. By default, the destination address is incremented by 1. If the STEPSEL bit is cleared, flexible step-size settings are available in the STEPSIZE register.

Value	Description
0	The Destination Address Increment is disabled
1	The Destination Address Increment is enabled

Bit 10 – SRCINC Source Address Increment Enable
Writing a '0' to this bit will disable the source address incrementation. The address will be kept fixed during the data transfer.

© 2025 Microchip Technology Inc. and its subsidiaries

Writing a '1' to this bit will enable the source address incrementation. By default, the source address is incremented by 1. If the STEPSEL bit is set, flexible step-size settings are available in the STEPSIZE register.

Value	Description
0	The Source Address Increment is disabled
1	The Source Address Increment is enabled

Bits 9:8 – BEATSIZE[1:0] Beat Size

These bits define the size of one beat. A beat is the size of one data transfer bus access, and the setting apply to both read and write accesses.

Value	Name	Description
0x0	BYTE	8-bit bus transfer
0x1	HWORD	16-bit bus transfer
0x2	WORD	32-bit bus transfer
other		Reserved

Bits 4:3 – BLOCKACT[1:0] Block Action

These bits define what actions the DMAC should take after a block transfer has completed.

BLOCKACT[1:0]	Name	Description
0x0	NOACT	Channel will be disabled if it is the last block transfer in the transaction
0x1	INT	Channel will be disabled if it is the last block transfer in the transaction and block interrupt
0x2	SUSPEND	Channel suspend operation is completed
0x3	BOTH	Both channel suspend operation and block interrupt

Bits 2:1 – EVOSEL[1:0] Event Output Selection

These bits define the event output selection.

EVOSEL[1:0]	Name	Description
0x0	DISABLE	Event generation disabled
0x1	BLOCK	Event strobe when block transfer complete
0x2		Reserved
0x3	BEAT	Event strobe when beat transfer complete

Bit 0 – VALID Descriptor Valid

Writing a '0' to this bit in the Descriptor or Write-Back memory will suspend the DMA channel operation when fetching the corresponding descriptor.

The bit is automatically cleared in the Write-Back memory section when channel is aborted, when an error is detected during the block transfer, or when the block transfer is completed.

Value	Description
0	The descriptor is not valid
1	The descriptor is valid

© 2025 Microchip Technology Inc. and its subsidiaries

26.10.2. Block Transfer Count

Name: BTCNT
Offset: 0x02
Property: -

The BTCNT register offset is relative to (BASEADDR or WRBADDR) + Channel Number * 0x10

Bit	15	14	13	12	11	10	9	8
	BTCNT[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
	BTCNT[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 – BTCNT[15:0] Block Transfer Count

This bit group holds the 16-bit block transfer count.
During a transfer, the internal counter value is decremented by one after each beat transfer. The internal counter is written to the corresponding write-back memory section for the DMA channel when the DMA channel loses priority, is suspended or gets disabled. The DMA channel can be disabled by a complete transfer, a transfer error or by software.

© 2025 Microchip Technology Inc. and its subsidiaries

26.10.3. Block Transfer Source Address

Name: SRCADDR
Offset: 0x04
Property: -

The SRCADDR register offset is relative to (BASEADDR or WRBADDR) + Channel Number * 0x10

Bit	31	30	29	28	27	26	25	24
	SRCADDR[31:24]							
Access	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	SRCADDR[23:16]							
Access	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SRCADDR[15:8]							
Access	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SRCADDR[7:0]							
Access	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – SRCADDR[31:0] Transfer Source Address

This bit field holds the block transfer source address.
When source address incrementation is disabled (BTCTRL.SRCINC=0), SRCADDR corresponds to the last beat transfer address in the block transfer.
When source address incrementation is enabled (BTCTRL.SRCINC=1), SRCADDR is calculated as follows:
If BTCTRL.STEPSEL = 1:

$$\mathrm{SRCADDR} = \mathrm{SRCADDR}_{START} + BTCNT \cdot \left(BEATSIZE + 1\right) \cdot 2^{\mathrm{STEPSIZE}}$$

If BTCTRL.STEPSEL= 0:

$$\mathrm{SRCADDR} = \mathrm{SRCADDR}_{START} + BTCNT \cdot \left(BEATSIZE + 1\right)$$

- $\mathrm{SRCADDR}_{START}$ is the source address of the first beat transfer in the block transfer
- BTCNT is the initial number of beats remaining in the block transfer
- BEATSIZE is the configured number of bytes in a beat
- STEPSIZE is the configured number of beats for each incrementation

© 2025 Microchip Technology Inc. and its subsidiaries

26.10.4. Block Transfer Destination Address

Name: DSTADDR
Offset: 0x08
Property: -

The DSTADDR register offset is relative to (BASEADDR or WRBADDR) + Channel Number * 0x10

Bit	31	30	29	28	27	26	25	24
	DSTADDR[31:24]							
Access	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DSTADDR[23:16]							
Access	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DSTADDR[15:8]							
Access	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DSTADDR[7:0]							
Access	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DSTADDR[31:0] Transfer Destination Address

This bit field holds the block transfer destination address.
When destination address incrementation is disabled (BTCTRL.DSTINC = 0), DSTADDR corresponds to the last beat transfer address in the block transfer.
When destination address incrementation is enabled (BTCTRL.DSTINC = 1), DSTADDR is calculated as follows:
If BTCTRL.STEPSEL = 1:

$$DSTADDR = DSTADDR_{START} + BTCNT \bullet (BEATSIZE + 1)$$

If BTCTRL.STEPSEL = 0:

$$DSTADDR = DSTADDR_{START} + BTCNT \bullet \left(BEATSIZE + 1\right) \bullet 2^{STEPSIZE}$$

- $DSTADDR_{START}$ is the destination address of the first beat transfer in the block transfer
- BTCNT is the initial number of beats remaining in the block transfer
- BEATSIZE is the configured number of bytes in a beat
- STEPSIZE is the configured number of beats for each incrementation

© 2025 Microchip Technology Inc. and its subsidiaries

26.10.5. Next Descriptor Address

Name: DESCADDR
Offset: 0x0C
Reset: 0x00000000
Property: -

The DESCADDR register offset is relative to (BASEADDR or WRBADDR) + Channel Number * 0x10

Bit	31	30	29	28	27	26	25	24
	DESCADDR[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
	DESCADDR[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
	DESCADDR[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
	DESCADDR[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 – DESCADDR[31:0] Next Descriptor Address
This bit group holds the SRAM address of the next descriptor. The value must be 64-bit aligned. If the value of this SRAM register is 0x00000000, the transaction will be terminated when the DMAC tries to load the next transfer descriptor.

© 2025 Microchip Technology Inc. and its subsidiaries

27. External Interrupt Controller (EIC)

27.1. Overview

The External Interrupt Controller (EIC) allows external pins to be configured as interrupt lines. Each interrupt line can be individually masked and can generate an interrupt on rising, falling, both edges or on high or low levels. Each external pin has a configurable filter to remove spikes. Also, each external pin can be configured to be asynchronous to wake-up the device from Sleep modes where all clocks were disabled. External pins can generate an event.

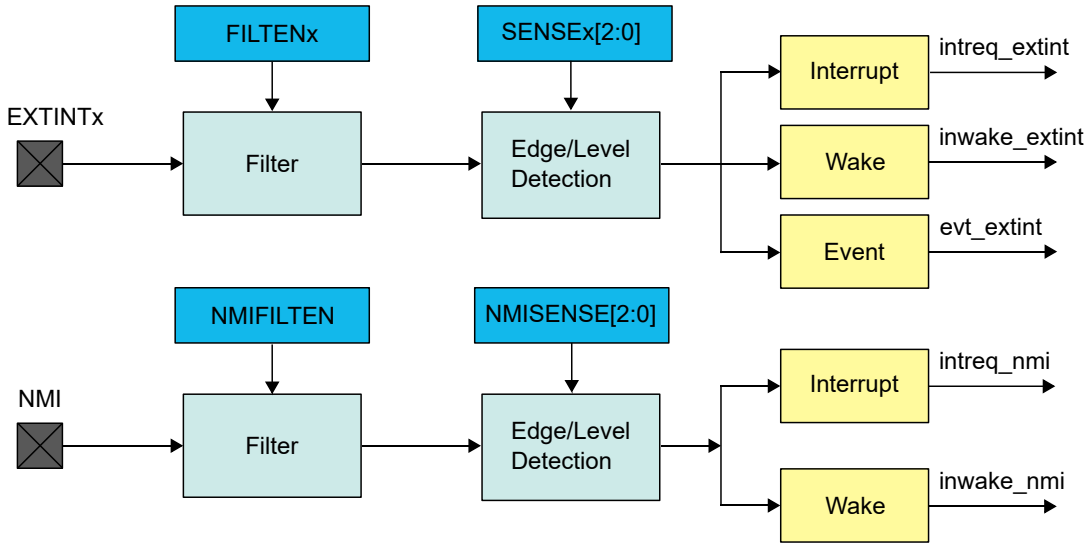
A separate Non-Maskable Interrupt (NMI) is also supported. It has properties similar to the other external interrupts but is connected to the NMI request of the NVIC.

27.2. Features

- Up to Four External Pins (EXTINTx) Plus One Non-Maskable Pin (NMI)
- Dedicated, Individually Maskable Interrupt for Each Pin
- Interrupt on Rising, Falling or Both Edges
- Synchronous or Asynchronous Edge Detection Mode
- Interrupt Pin Debouncing
- Interrupt on High or Low Levels
- Asynchronous Interrupts for Sleep Modes without Clock
- Filtering of External Pins
- Event Generation from EXTINTx

27.3. Block Diagram

Figure 27-1. EIC Block Diagram



27.4. Signal Description

Signal Name	Type	Description
EXTINT[3..0]	Digital Input	External interrupt pin
NMI	Digital Input	Non-maskable interrupt pin

One signal may be available on several pins.

© 2025 Microchip Technology Inc. and its subsidiaries

27.5. Product Dependencies

The following sections describe how the other parts of the system must be configured correctly to use this peripheral.

27.5.1. I/O Lines

To use EIC's I/O lines, configure the I/O pins using the I/O Peripheral Pin Select (PPS).

27.5.2. Power Management

The EIC continues to operate in any sleep modes (Standby Sleep, Idle) where the selected source clock is running. The EIC's interrupts can be used to wake up the device from sleep modes. Events connected to the Event System can trigger other operations in the system without exiting the sleep modes.

27.5.3. Clocks

The EIC bus clock (PB1_CLK) can be enabled and disabled by the CRU. The default state of PB1_CLK can be found in the CRU and PMD registers.

Some optional functions need a peripheral clock, which can either be a generic clock (GCLK_EIC, for wider frequency selection) or a Ultra Low-Power 32 KHz clock 32KHz_LPCLK, for the highest power efficiency). One of the clock sources must be configured and enabled before using the peripheral:

GCLK_EIC is configured and enabled in the CTRLA of CRU registers. For more details, see *Clock and Reset Unit (CRU)* from Related Links.

32KHz_LPCLK is provided by the various internal and external low power clock sources. For more details on configuration and selection of the clock, see *Clock and Reset Unit (CRU)* from Related Links.

Both GCLK_EIC and 32KHz_LPCLK are asynchronous to the user interface clock (PB1_CLK). Due to this asynchronicity, writes to certain registers require synchronization between the clock domains.

Related Links

Clock and Reset Unit (CRU)

27.5.4. DMA

Not applicable.

27.5.5. Interrupts

There are four external interrupts (EXTINT) and one Non-Maskable Interrupt (NMI).

All the EXTINT interrupt request lines are connected to a single interrupt in the interrupt controller. Using the EIC interrupt requires the interrupt controller to be configured first.

The NMI interrupt request line is connected to the non-maskable interrupt of the interrupt controller but does not require the interrupt to be configured.

27.5.6. Events

The events are connected to the Event System. Using the events requires the Event System to be configured first.

Related Links

Event System (EVSYS)

27.5.7. Debug Operation

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

© 2025 Microchip Technology Inc. and its subsidiaries

27.5.8. Register Access Protection

All registers with write access can be write protected optionally by the 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 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.

27.5.9. Analog Connections

Not applicable.

27.6. Functional Description

27.6.1. Principle of Operation

The EIC detects the edge or level condition to generate interrupts to the CPU interrupt controller or events to the Event System. Each external interrupt pin (EXTINT) can be filtered using majority vote filtering, clocked by GCLK_EIC or by 32KHz_LPCLK. For more details, see *External Pin Processing* from Related Links.

Related Links

External Pin Processing

27.6.2. Basic Operation

27.6.2.1. Initialization

The EIC must be initialized in the following order:

1. Enable CLK_EIC_APB.
2. If required, configure the NMI by writing the Non-Maskable Interrupt Control register (NMICTRL).
3. Enable GCLK_EIC or 32KHz_LPCLK when one of the following configurations is selected:
   - The NMI uses edge detection or filtering
   - One EXTINT uses filtering
   - One EXTINT uses synchronous edge detection
   - One EXTINT uses debouncing

   GCLK_EIC is used when a frequency higher than 32 KHz is required for filtering.

   32KHz_LPCLK is recommended when power consumption is the priority. For 32KHz_LPCLK, write a '1' to the Clock Selection bit in the Control A register (CTRLA.CKSEL).
4. Configure the EIC input sense and filtering by writing the Configuration register (CONFIG).
5. Optionally, enable the Asynchronous mode.
6. Optionally, enable the Debouncer mode.
7. Enable the EIC by writing a '1' to CTRLA.ENABLE.
8. Wait for SYNCBUSY.ENABLE = 0. Level detection is now functional. Edge detection will be functional after three cycles of the selected GCLK (GCLK_EIC or CLK_ULP32K).

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

- Clock Selection bit in Control A register (CTRLA.CKSEL)

The following registers are enable-protected:

© 2025 Microchip Technology Inc. and its subsidiaries

- Event Control register (EVCTRL)
- Configuration register (CONFIG)
- External Interrupt Asynchronous mode register (ASYNCH)
- Debouncer Enable register (DEBOUNCEN)
- Debounce Prescaler register (DPRESCALER)

Enable-protected bits in the CTRLA register can be written at the same time when setting CTRLA.ENABLE to '1' but not at the same time as CTRLA.ENABLE is being cleared.

Enable protection is denoted by the Enable-Protected property in the register description.

See the *NMICTRL*, *CTRLA*, *CONFIG*, *ASYNCH*, *DEBOUNCEN*, *DPRESCALER*, *EVCTRL* registers in the *EIC Register Summary* from Related Links.

Related Links
Register Summary

27.6.2.2. Enabling, Disabling and Resetting

The EIC is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The EIC is disabled by writing CTRLA.ENABLE to '0'.

The EIC is reset by setting the Software Reset bit in the Control register (CTRLA.SWRST). All registers in the EIC will be reset to their initial state, and the EIC will be disabled.

27.6.3. External Pin Processing

Each external pin can be configured to generate an interrupt/event on edge detection (rising, falling or both edges) or level detection (high or low). The sense of external interrupt pins is configured by writing the Input Sense x bits in the Config n register (CONFIG.SENSEx). The corresponding interrupt flag (INTFLAG.EXTINT[x]) in the Interrupt Flag Status and Clear register (INTFLAG) is set when the interrupt condition is met.

When the interrupt flag has been cleared in edge-sensitive mode, INTFLAG.EXTINT[x] will only be set if a new interrupt condition is met.

In level-sensitive mode, when the interrupt has been cleared, INTFLAG.EXTINT[x] will be set immediately if the EXTINTx pin still matches the interrupt condition.

Each external pin can be filtered by a majority vote filtering, clocked by GCLK_EIC or 32KHz_LPCLK. Filtering is enabled if the bit Filter Enable x in the Configuration n register (CONFIG.FILTENx) is written to '1'. The majority vote filter samples the external pin three times with GCLK_EIC or 32KHz_LPCLK and outputs the value when two or more samples are equal.

Table 27-1. Majority Vote Filter

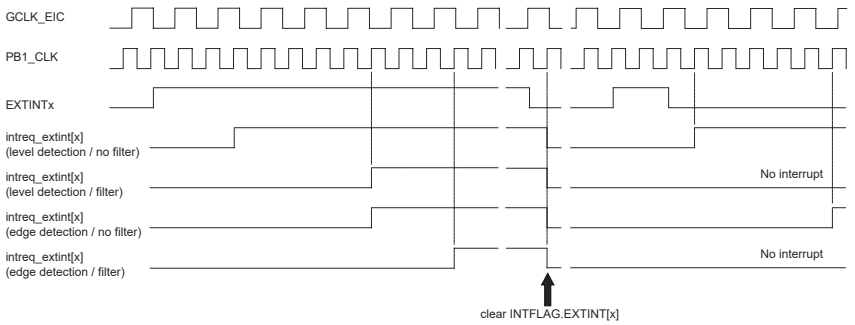
Samples [0, 1, 2]	Filter Output
[0,0,0]	0
[0,0,1]	0
[0,1,0]	0
[0,1,1]	1
[1,0,0]	0
[1,0,1]	1
[1,1,0]	1
[1,1,1]	1

When an external interrupt is configured for level detection and when filtering is disabled, detection is done asynchronously. Level detection and asynchronous edge detection does not require GCLK_EIC or 32KHz_LPCLK, but interrupt and events can still be generated.

© 2025 Microchip Technology Inc. and its subsidiaries

If filtering or synchronous edge detection or debouncing is enabled, the EIC automatically requests GCLK_EIC or 32KHz_LPCLK to operate. The selection between these two clocks is done by writing the Clock Selection bits in the Control A register (CTRLA.CKSEL). GCLK_EIC must be enabled in the CRU. In these modes the external pin is sampled at the EIC clock rate, thus pulses with duration lower than two EIC clock periods may not be properly detected.

Figure 27-2. Interrupt Detection Latency by Modes (Rising Edge)



The detection latency depends on the detection mode.

Table 27-2. Detection Latency

Detection Mode	Latency (Worst Case)
Level without filter	Five PB1_CLK periods
Level with filter	Four GCLK_EIC/32KHz_LPCLK periods + five PB1_CLK periods
Edge without filter	Four GCLK_EIC/32KHz_LPCLK periods + five PB1_CLK periods
Edge with filter	Six GCLK_EIC/32KHz_LPCLK periods + five PB1_CLK periods

27.6.4. Additional Features

27.6.4.1. Non-Maskable Interrupt (NMI)

The non-maskable interrupt pin can also generate an interrupt on edge or level detection, but it is configured with the dedicated NMI Control register (NMICTRL). To select the sense for NMI, write to the NMISENSE bit group in the NMI Control register (NMICTRL.NMISENSE). NMI filtering is enabled by writing a '1' to the NMI Filter Enable bit (NMICTRL.NMIFILTEN).

If edge detection or filtering is required, enable GCLK_EIC or 32KHz_LPCLK.

NMI detection is enabled only by the NMICTRL.NMISENSE value, and the EIC module is not required to be enabled.

When an NMI is detected, the Non-maskable Interrupt flag in the NMI Flag Status and Clear register is set (NMIFLAG.NMI). NMI interrupt generation is always enabled, and NMIFLAG.NMI generates an interrupt request when set.

27.6.4.2. Asynchronous Edge Detection Mode (No Debouncing)

The EXTINT edge detection operates synchronously or asynchronously, as selected by the Asynchronous Control Mode bit for external pin x in the External Interrupt Asynchronous Mode register (ASYNCH.ASYNCH[x]). The EIC edge detection is operated synchronously when the Asynchronous Control Mode bit (ASYNCH.ASYNCH[x]) is '0' (default value). It is operated asynchronously when ASYNCH.ASYNCH[x] is written to '1'.

In *Synchronous Edge Detection Mode*, the external interrupt (EXTINT) or the NMI pins are sampled using the EIC clock as defined by the Clock Selection bit in the Control A register (CTRLA.CKSEL). The External Interrupt flag (INTFLAG.EXTINT[x]) or Non-Maskable Interrupt flag (NMIFLAG.NMI) is set when the last sampled state of the pin differs from the previously sampled state. The EIC clock is needed in this mode.

© 2025 Microchip Technology Inc. and its subsidiaries

Note: The Synchronous Edge Detection Mode can be used in Idle and Standby Sleep modes.

In *Asynchronous Edge Detection Mode*, the external interrupt (EXTINT) pins or the NMI pins set the External Interrupt flag or Non-Maskable Interrupt flag (INTFLAG.EXTINT[x] or NMIFLAG) directly. The EIC clock is not needed in this mode.

Note:
The Asynchronous Edge Detection mode can be used in Idle and Standby Sleep modes.

27.6.4.3. Interrupt Pin Debouncing

The external interrupt pin (EXTINT) edge detection can use a debouncer to improve input noise immunity. When selected, the debouncer can work in the synchronous mode or the asynchronous mode, depending on the configuration of the ASYNCH.ASYNCH[x] bit for the pin. The debouncer uses the EIC clock as defined by the bit CTRLA.CKSEL to clock the debouncing circuitry. The debouncing timeframe is set with the debouncer prescaler DPRESCALER.PRESCALERn, which provides the low frequency clock tick that is used to reject higher frequency signals.

The Debouncing mode for pin EXTINT x can be selected only if the Sense bits in the Configuration y register (CONFIG.SENSEx) are set to RISE, FALL or BOTH. If the Debouncing mode for pin EXTINT x is selected, the Filter mode for that pin (CONFIG.FILTENx) cannot be selected.

The debouncer manages an internal valid pin state that depends on the external interrupt (EXTINT) pin transitions, the Debouncing mode and the debouncer prescaler frequency. The valid pin state reflects the pin value after debouncing. The external interrupt pin (EXTINT) is sampled continuously on the EIC clock. The sampled value is evaluated on each low frequency clock tick to detect a transitional edge when the sampled value is different from the current valid pin state. The sampled value is evaluated on each EIC clock when DPRESCALER.TICKON=0 or on each low frequency clock tick when DPRESCALER.TICKON=1, to detect a bounce when the sampled value is equal to the current valid pin state. Transitional edge detection increments the transition counter of the EXTINT pin, while bounce detection resets the transition counter. The transition counter must exceed the transition count threshold as defined by the DPRESCALER.STATESn bit field. In the Synchronous mode, the threshold is 4 when DPRESCALER.STATESn=0 or 8 when DPRESCALER.STATESn=1. In the asynchronous mode, the threshold is 4.

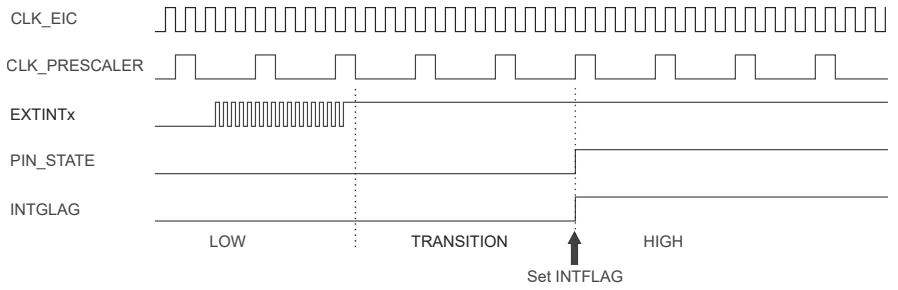
The valid pin state for the pins can be accessed by reading the register PINSTATE for either the Synchronous or Asynchronous Debouncing mode.

Synchronous Edge Detection mode – In this mode, the external interrupt (EXTINT) pin is sampled continuously on the EIC clock.

1. A pin edge transition is validated when the sampled value is consistently different from the current valid pin state for 4 (or 8 depending on bit DPRESCALER.STATESn) consecutive ticks of the low frequency clock.
2. Any pin sample at the low frequency clock tick rate with a value opposite to the current valid pin state increments the transition counter.
3. Any pin sample at the EIC clock rate (when DPRESCALER.TICKON=0) or the low frequency clock tick (when DPRESCALER.TICKON=1) with a value identical to the current valid pin state returns the transition counter to zero.
4. When the transition counter meets the count threshold, the pin edge transition is validated and the pin state PINSTATE.PINSTATE[x] is changed to the detected level.
5. The external interrupt flag (INTFLAG.EXTINT[x]) is set when the pin state PINSTATE.PINSTATE[x] is changed.

© 2025 Microchip Technology Inc. and its subsidiaries

Figure 27-3. EXTINT Pin Synchronous Debouncing (Rising Edge)

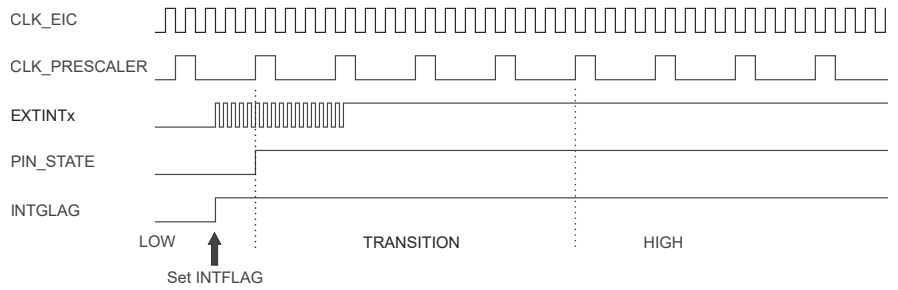


In the Synchronous Edge Detection mode, the EIC clock is required. The Synchronous Edge Detection mode can be used in Idle and Standby Sleep modes.

Asynchronous Edge Detection mode – In this mode, the external interrupt (EXTINT) pin directly drives an asynchronous edges detector, which triggers any rising or falling edge on the pin:

1. Any edge detected that indicates a transition from the current valid pin state immediately sets the valid pin state PINSTATE.PINSTATE[x] to the detected level.
2. The external interrupt flag (INTFLAG.EXTINT[x] is immediately changed.
3. The edge detector will then be idle until no other rising or falling edge transition is detected during four consecutive ticks of the low frequency clock.
4. Any rising or falling edge transition detected during the Idle state returns the transition counter to 0.
5. After four consecutive ticks of the low frequency clock without bounce detected, the edge detector is ready for a new detection.

Figure 27-4. EXTINT Pin Asynchronous Debouncing (Rising Edge)



In this mode, the EIC clock is requested. The Asynchronous Edge Detection mode can be used in Idle and Standby Sleep modes.

27.6.5. DMA Operation

Not applicable.

27.6.6. Interrupts

The EIC has the following interrupt sources:

- External interrupt (EXTINTx) pins. See *Basic Operation* from Related Links.
- Non-maskable interrupt (NMI) pin. See *Additional Features* from Related Links.

Each interrupt source has an associated Interrupt flag. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) is set when an Interrupt condition occurs (NMIFLAG for NMI). Each interrupt, except NMI, can be individually enabled by setting the corresponding bit in the Interrupt Enable Set register (INTENSET=1) and disabled by setting the corresponding bit in the

© 2025 Microchip Technology Inc. and its subsidiaries

Interrupt Enable Clear register (INTENCLR=1). The status of enabled interrupts can be read from either INTENSET or INTENCLR.

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled or the EIC is reset. For details on how to clear Interrupt flags, see the *INTFLAG* register from Related Links. The EIC has one interrupt request line for all external interrupts (EXTINTx) and one line for NMI. The user must read the INTFLAG (or NMIFLAG) register to determine which Interrupt condition is present.

Notes:

- Interrupts must be globally enabled for interrupt requests to be generated.
- If an external interrupt (EXTINT) is common on two or more I/O pins, only one will be active (the first one programmed).
- To clear the NMI interrupt, both the RNMICON.EXT and NMIFLAG.NMI bit must be cleared by writing a "1".

Related Links

Basic Operation
Additional Features
INTFLAG
Interrupt Flag Status and Clear

27.6.7. Events

The EIC can generate the following output events:

- External event from pin (EXTINT0-3)

Setting an Event Output Control register (EVCTRL.EXTINTEO) enables the corresponding output event. Clearing this bit disables the corresponding output event. For more details on configuring the event system, see *Event System (EVSYS)* from Related Links.

When the condition on pin EXTINTx matches the configuration in the CONFIG register, the corresponding event is generated, if enabled.

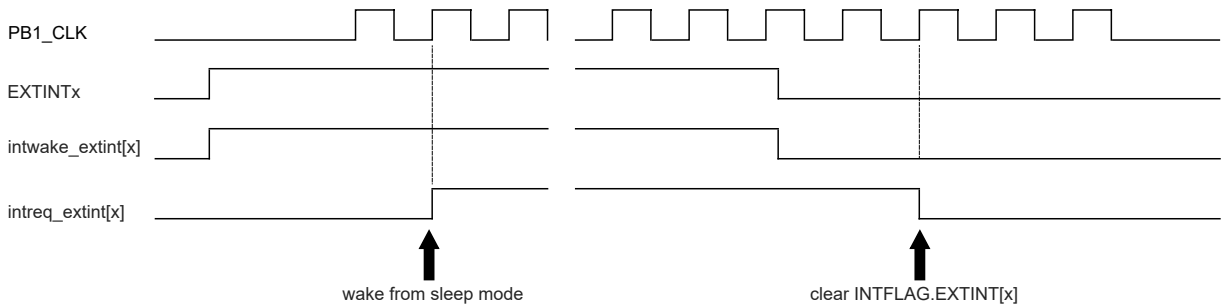
Related Links

Event System (EVSYS)

27.6.8. Sleep Mode Operation

In Sleep modes, an EXTINTx pin can wake up the device if the corresponding condition matches the configuration in the CONFIG register, and the corresponding bit in the Interrupt Enable Set register (INTENSET) is written to '1'.

Figure 27-5. Wake-up Operation Example (High-Level Detection, No Filter, Interrupt Enable Set)



© 2025 Microchip Technology Inc. and its subsidiaries

27.6.9. Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset bit in control register (CTRLA.SWRST)
- Enable bit in control register (CTRLA.ENABLE)

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

© 2025 Microchip Technology Inc. and its subsidiaries

27.7. Register Summary

See the *EIC* module in the *Product Memory Mapping Overview* from Related Links for base address.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0				CKSEL			ENABLE	SWRST
0x01	NMICTRL	7:0				NMIASYNCH	NMIFILTEN	NMISENSE[2:0]		
0x02	NMIFLAG	7:0								NMI
0x03	Reserved									
0x04	SYNCBUSY	7:0							ENABLE	SWRST
		15:8								
		23:16								
		31:24								
0x08	EVCTRL	7:0					EXTINTEO[3:0]			
		15:8								
		23:16								
		31:24								
0x0C	INTENCLR	7:0					EXTINT[3:0]			
		15:8								
		23:16								
		31:24								
0x10	INTENSET	7:0					EXTINT[3:0]			
		15:8								
		23:16								
		31:24								
0x14	INTFLAG	7:0					EXTINT[3:0]			
		15:8								
		23:16								
		31:24								
0x18	ASYNCH	7:0					ASYNCH[3:0]			
		15:8								
		23:16								
		31:24								
0x1C	CONFIG	7:0	FILTEN1	SENSE1[2:0]			FILTEN0	SENSE0[2:0]		
		15:8	FILTEN3	SENSE3[2:0]			FILTEN2	SENSE2[2:0]		
		23:16								
		31:24								
0x20 ... 0x2F	Reserved									
0x30	DEBOUNCEN	7:0					DEBOUNCEN[3:0]			
		15:8								
		23:16								
		31:24								
0x34	DPRESCALER	7:0					STATES0	PRESCALER0[2:0]		
		15:8								
		23:16								TICKON
		31:24								
0x38	PINSTATE	7:0					PINSTATE[3:0]			
		15:8								
		23:16								
		31:24								

Related Links

Product Memory Mapping Overview

27.8. Register Description

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

© 2025 Microchip Technology Inc. and its subsidiaries

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.

© 2025 Microchip Technology Inc. and its subsidiaries

27.8.1. Control A

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

Bit	7	6	5	4	3	2	1	0
				CKSEL			ENABLE	SWRST
Access				RW			RW	W
Reset				0			0	0

Bit 4 – CKSEL Clock Selection
The EIC can be clocked either by GCLK_EIC (when a frequency higher than 32.768 KHz is required for filtering) or by 32KHz_LPCLK (when power consumption is the priority).
This bit is not Write-Synchronized.

Value	Description
0	The EIC is clocked by GCLK_EIC.
1	The EIC is clocked by 32KHz_LPCLK.

Bit 1 – ENABLE Enable
Due to synchronization, there is a delay between writing to CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the Enable bit in the Synchronization Busy register is set (SYNCBUSY.ENABLE=1). SYNCBUSY.ENABLE is cleared when the operation is complete.
This bit is not Enable-Protected.
This bit is Write-Synchronized.

Value	Description
0	The EIC is disabled.
1	The EIC is enabled.

Bit 0 – SWRST Software Reset
Writing a '0' to this bit has no effect.
Writing a '1' to this bit resets all registers in the EIC to their initial state, and the EIC is disabled.
Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write operation are discarded.
Due to synchronization, there is a delay from writing CTRLA.SWRST until the Reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the Reset is complete.
This bit is not Enable-Protected.
This bit is Write-Synchronized.

Value	Description
0	There is no ongoing reset operation.
1	The reset operation is ongoing.

© 2025 Microchip Technology Inc. and its subsidiaries

27.8.2. Non-Maskable Interrupt Control

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

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

Bit 4 – NMIASYNCH Non-Maskable Interrupt Asynchronous Edge Detection Mode
The NMI edge detection can be operated synchronously or asynchronously to the EIC clock.
In Synchronous Edge Detection mode, the NMI pin is sampled using the EIC clock as defined by the bit CTRLA.CKSEL. The Non-Maskable Interrupt flag (NMIFLAG) is set when the pin and the pin sampler have a different value. In this mode, the EIC clock is required. The Synchronous Edge Detection mode can be used in all Sleep modes except STANDBY.
In Asynchronous Edge Detection mode, the NMI pins directly drive the set of the Non-Maskable Interrupt flag (NMIFLAG). In this mode, the EIC clock is not requested. The Asynchronous Edge Detection Mode can be used in all sleep modes.

Value	Description
0	The NMI edge detection is synchronously operated.
1	The NMI edge detection is asynchronously operated.

Bit 3 – NMIFILTEN Non-Maskable Interrupt Filter Enable

Value	Description
0	NMI filter is disabled.
1	NMI filter is enabled.

Bits 2:0 – NMISENSE[2:0] Non-Maskable Interrupt Sense Configuration
These bits define which edge or level the NMI triggers on.
Note: The NMI cannot be triggered based on level, but it is always based on edge.

Value	Name	Description
0x0	NONE	No detection
0x1	RISE	Rising-edge detection
0x2	FALL	Falling-edge detection
0x3	BOTH	Both-edge detection
0x4	HIGH	High-level detection
0x5	LOW	Low-level detection
0x6 - 0x7	—	Reserved

© 2025 Microchip Technology Inc. and its subsidiaries

27.8.3. Non-Maskable Interrupt Flag Status and Clear

Name: NMIFLAG
Offset: 0x2
Reset: 0x00

Bit	7	6	5	4	3	2	1	0
								NMI
Access								RW
Reset								0

Bit 0 – NMI Non-Maskable Interrupt
This flag is cleared by writing a '1' to it.
This flag is set when the NMI pin matches the NMI sense configuration and generates an interrupt request.
Writing a '0' to this bit has no effect.

© 2025 Microchip Technology Inc. and its subsidiaries

27.8.4. Synchronization Busy

Name: SYNCBUSY
Offset: 0x04
Reset: 0x00000000

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							ENABLE	SWRST
Access							R	R
Reset							0	0

Bit 1 – ENABLE Enable Synchronization Busy Status

Important: The SYNCBUSY.ENABLE bit is functional in level detection and asynchronous edge detection modes. In synchronous edge detection mode, it is necessary to wait for three cycles of the selected clock (GCLK_EIC) after enabling the EIC (CTRLA.ENABLE = 1).

Value	Description
0	Write synchronization for CTRLA.ENABLE bit is complete.
1	Write synchronization for CTRLA.ENABLE bit is ongoing.

Bit 0 – SWRST Software Reset Synchronization Busy Status

Value	Description
0	Write synchronization for CTRLA.SWRST bit is complete.
1	Write synchronization for CTRLA.SWRST bit is ongoing.

© 2025 Microchip Technology Inc. and its subsidiaries

27.8.5. Event Control

Name: EVCTRL
Offset: 0x08
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

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

Bits 3:0 – EXTINTEO[3:0] External Interrupt Event Output Enable
The bit x of EXTINTEO enables the event associated with the EXTINTx pin.

Value	Description
0	Event from pin EXTINTx is disabled.
1	Event from pin EXTINTx is enabled and will be generated when EXTINTx pin matches the external interrupt sensing configuration.

© 2025 Microchip Technology Inc. and its subsidiaries

27.8.6. Interrupt Enable Clear

Name: INTENCLR
Offset: 0x0C
Reset: 0x00000000
Property: PAC Write-Protection

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

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
					EXTINT[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 3:0 – EXTINT[3:0] External Interrupt Enable
The bit x of EXTINT disables the interrupt associated with the EXTINTx pin.
Writing a '0' to bit x has no effect.
Writing a '1' to bit x will clear the External Interrupt Enable bit x, which disables the external interrupt EXTINTx.

Value	Description
0	The external interrupt x is disabled.
1	The external interrupt x is enabled.

© 2025 Microchip Technology Inc. and its subsidiaries

27.8.7. Interrupt Enable Set

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

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

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
					EXTINT[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 3:0 – EXTINT[3:0] External Interrupt Enable
The bit x of EXTINT enables the interrupt associated with the EXTINTx pin.
Writing a '0' to bit x has no effect.
Writing a '1' to bit x will set the External Interrupt Enable bit x, which enables the external interrupt EXTINTx.

Value	Description
0	The external interrupt x is disabled.
1	The external interrupt x is enabled.

© 2025 Microchip Technology Inc. and its subsidiaries

27.8.8. Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x14
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

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

Bits 3:0 – EXTINT[3:0] External Interrupt
The flag bit x is cleared by writing a '1' to it.
This flag is set when EXTINTx pin matches the external interrupt sense configuration and will generate an interrupt request if INTENCLR/SET.EXTINT[x] is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the External Interrupt x flag.

© 2025 Microchip Technology Inc. and its subsidiaries

27.8.9. External Interrupt Asynchronous Mode

Name: ASYNCH
Offset: 0x18
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

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

Bits 3:0 – ASYNCH[3:0] Asynchronous Edge Detection Mode
The bit x of ASYNCH set the Asynchronous Edge Detection Mode for the interrupt associated with the EXTINTx pin.

Value	Description
0	The EXTINT x edge detection is synchronously operated.
1	The EXTINT x edge detection is asynchronously operated.

© 2025 Microchip Technology Inc. and its subsidiaries

27.8.10. External Interrupt Sense Configuration

Name: CONFIG
Offset: 0x1C
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	FILTEN3	SENSE3[2:0]			FILTEN2	SENSE2[2:0]		
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
	FILTEN1	SENSE1[2:0]			FILTEN0	SENSE0[2:0]		
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 3, 7, 11, 15 – FILTENx Filter Enable x [x=3..0]
Note: The filter must be disabled if the asynchronous detection is enabled.

Value	Description
0	Filter is disabled for EXTINT[x] input.
1	Filter is enabled for EXTINT[x] input.

Bits 0:2, 4:6, 8:10, 12:14 – SENSEx Input Sense Configuration x [x=3..0]
These bits define which edge or level the interrupt or event for EXTINT[x] is generated on.

Value	Name	Description
0x0	NONE	No detection
0x1	RISE	Rising-edge detection
0x2	FALL	Falling-edge detection
0x3	BOTH	Both-edge detection
0x4	HIGH	High-level detection
0x5	LOW	Low-level detection
0x6 - 0x7	—	Reserved

© 2025 Microchip Technology Inc. and its subsidiaries

27.8.11. Debouncer Enable

Name: DEBOUNCEN
Offset: 0x30
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

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

Bits 3:0 – DEBOUNCEN[3:0] Debouncer Enable
The bit x of DEBOUNCEN set the Debounce mode for the interrupt associated with the EXTINTx pin.

Value	Description
0	The EXTINT x edge input is not debounced.
1	The EXTINT x edge input is debounced.

© 2025 Microchip Technology Inc. and its subsidiaries

27.8.12. Debouncer Prescaler

Name: DPRESCALER
Offset: 0x34
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								RW
Reset								0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access					RW	RW	RW	RW
Reset					0	0	0	0

Bit 16 – TICKON Pin Sampler frequency selection
This bit selects the clock used for the sampling of bounce during transition detection.

Value	Description
0	The bounce sampler is using GCLK_EIC.
1	The bounce sampler is using the low frequency clock.

Bit 3 – STATES0 Debouncer number of states
This bit selects the number of samples by the debouncer low frequency clock needed to validate a transition from the current pin state to the next pin state in synchronous debouncing mode for pins EXTINT[3:0].

Value	Description
0	The number of low frequency samples is 3.
1	The number of low frequency samples is 7.

Bits 2:0 – PRESCALER0[2:0] Debouncer Prescaler
These bits select the debouncer low frequency clock for pins EXTINT[3:0].

Value	Name	Description
0x0	F/2	EIC clock divided by 2
0x1	F/4	EIC clock divided by 4
0x2	F/8	EIC clock divided by 8
0x3	F/16	EIC clock divided by 16
0x4	F/32	EIC clock divided by 32
0x5	F/64	EIC clock divided by 64
0x6	F/128	EIC clock divided by 128
0x7	F/256	EIC clock divided by 256

© 2025 Microchip Technology Inc. and its subsidiaries

27.8.13. Pin State

Name: PINSTATE
Offset: 0x38
Reset: 0x00000000

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
					PINSTATE[3:0]			
Access					R	R	R	R
Reset					0	0	0	0

Bits 3:0 – PINSTATE[3:0] Pin State
These bits return the valid pin state of the debounced external interrupt pin EXTINTx.

© 2025 Microchip Technology Inc. and its subsidiaries

28. Configurable Custom Logic (CCL)

28.1. Overview

The Configurable Custom Logic (CCL) is a programmable logic peripheral that can be connected to the device pins, to events or to other internal peripherals. This allows the user to eliminate logic gates for simple glue logic functions on the PCB.

Each Look-up Table (LUT) consists of three inputs:

- Truth table
- Optional synchronizer/filter
- Optional edge detector

Each LUT can generate an output as a user-programmable logic expression with three inputs. Inputs can be individually masked.

The output can be combinatorially generated from the inputs and can be filtered to remove spikes. Optional sequential logic can be used. The inputs of the sequential module are individually controlled by two independent, adjacent LUT (LUT0/LUT1) outputs, enabling complex waveform generation.

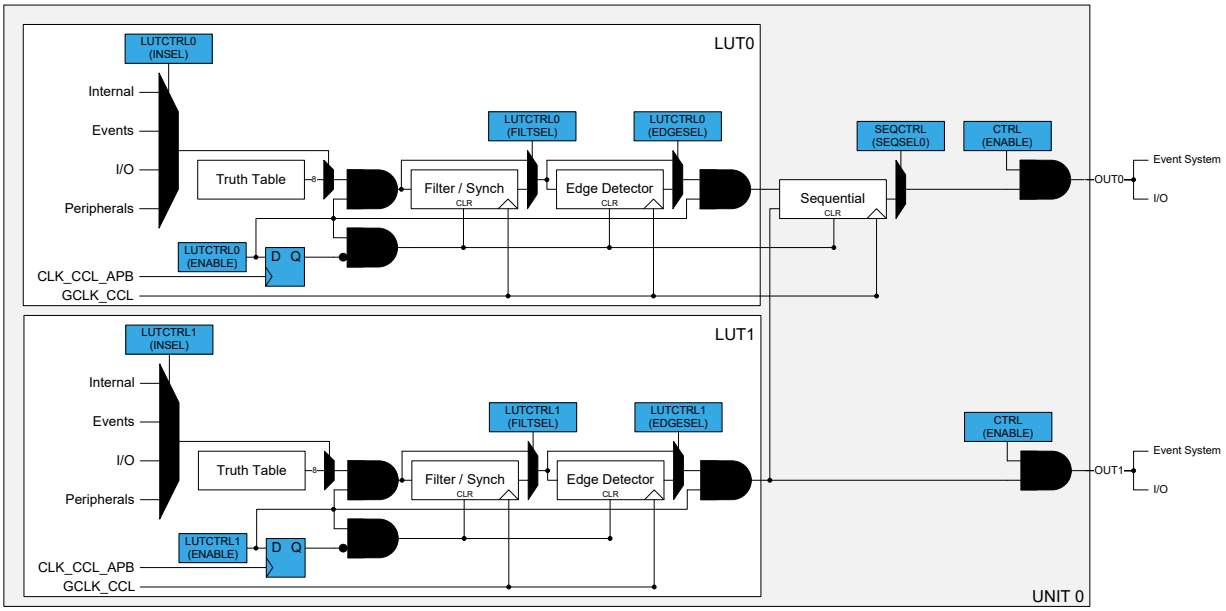
28.2. Features

- Glue Logic for General Purpose PCB Design
- Two Programmable Look-up Tables (LUTs)
- Combinatorial Logic Functions:
  - AND
  - NAND
  - OR
  - NOR
  - XOR
  - XNOR
  - NOT
- Sequential Logic Functions:
  - Gated D Flip-Flop
  - JK Flip-Flop
  - Gated D latch
  - RS latch
- Flexible LUT Inputs Selection:
  - I/Os
  - Events
  - Internal peripherals
  - Subsequent LUT output
- Output Can be Connected to the I/O Pins or the Event System
- Optional Synchronizer, Filter or Edge Detector Available on Each LUT Output

© 2025 Microchip Technology Inc. and its subsidiaries

28.3. Block Diagram

Figure 28-1. Configurable Custom Logic



28.4. Signal Description

Pin Name	Type	Description
OUT[n:0]	Digital output	Output from look-up table
IN[3n+2:0]	Digital input	Input to look-up table

1. n (n=1) is the number of CCL groups.

See *I/O Ports and Peripheral Pin Select (PPS)* from Related Links for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

Related Links

I/O Ports and Peripheral Pin Select (PPS)

28.5. Product Dependencies

The following sections describe how the other parts of the system must be configured correctly to use this peripheral.

28.5.1. I/O Lines

The CCL can take inputs and generate output through I/O pins. For this to function properly, the I/O pins must be configured to be used by a Look-up Table (LUT) using PPS configuration.

28.5.2. Power Management

This peripheral can continue to operate in any sleep modes (Standby Sleep, Idle) where its source clock is running. Events connected to the event system can trigger other operations in the system without exiting sleep modes.

© 2025 Microchip Technology Inc. and its subsidiaries

28.5.3. Clocks

The CCL bus clock PB2_CLK (CLK_CCL_APB) can be enabled and disabled in the CRU.

A generic clock (GCLK_CCL) is optionally required to clock the CCL. This clock must be configured and enabled in the Generic Clock Controller (GCLK) before using input events, filter, edge detection or sequential logic. GCLK_CCL is required when input events, a filter, an edge detector or a sequential sub-module is enabled.

This generic clock is asynchronous to the user interface clock.

See *Clock and Reset Unit (CRU)* from Related Links.

Related Links

Clock and Reset Unit (CRU)

28.5.4. DMA

Not applicable.

28.5.5. Interrupts

Not applicable.

28.5.6. Events

The CCL can use events from other peripherals and generate events that can be used by other peripherals. For this feature to function, the events have to be configured properly. See *Event System (EVSYS)* from Related Links.

Related Links

Event System (EVSYS)

28.5.7. Debug Operation

When the CPU is halted in Debug mode the CCL continues normal operation. However, the CCL cannot be halted when the CPU is halted in Debug mode. If the CCL is configured in a way that requires it to be periodically serviced by the CPU, improper operation or data loss may result during debugging.

28.5.8. Register Access Protection

All registers with write access can be write protected optionally by the PAC. See *Peripheral Access Controller (PAC)* from Related Links.

Optional write protection by the 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

Peripheral Access Controller (PAC)

28.5.9. Analog Connections

Not applicable.

28.6. Functional Description

28.6.1. Principle of Operation

Configurable Custom Logic (CCL) is a programmable logic block that can use the device port pins, internal peripherals, and the internal Event System as both input and output channels. The CCL can serve as glue logic between the device and external devices. The CCL can eliminate the need for external logic component and can also help the designer overcome challenging real-time constrains

© 2025 Microchip Technology Inc. and its subsidiaries

by combining core independent peripherals in clever ways to handle the most time critical parts of the application independent of the CPU.

28.6.2. Operation

28.6.2.1. Initialization

The following bits are enable-protected, meaning that they can only be written when CCL is disabled and the corresponding even LUT is disabled (CTRL.ENABLE=0 and LUTCTRLx.ENABLE=0):

- Sequential Selection bits in the Sequential Control (SEQCTRL.SEQSEL) register

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

- LUT Control x (LUTCTRLx) register, except the ENABLE bit

Enable-protected bits in the LUTCTRLx registers can be written at the same time as LUTCTRLx.ENABLE is written to '1' but not at the same time as LUTCTRLx.ENABLE is written to '0'.

Enable protection is denoted by the Enable-Protected property in the register description.

28.6.2.2. Enabling, Disabling and Resetting

The CCL is enabled by writing a '1' to the Enable bit in the Control register (CTRL.ENABLE). The CCL is disabled by writing a '0' to CTRL.ENABLE.

Each LUT is enabled by writing a '1' to the Enable bit in the LUT Control x register (LUTCTRLx.ENABLE). Each LUT is disabled by writing a '0' to LUTCTRLx.ENABLE.

The CCL is Reset by writing a '1' to the Software Reset bit in the Control register (CTRL.SWRST). All registers in the CCL are reset to their initial state, and the CCL is disabled.

28.6.2.3. Lookup Table Logic

The lookup table in each LUT unit can generate any logic expression OUT as a function of three inputs (IN[2:0]), as shown in Figure 28-2. One or more inputs can be masked. The truth table for the expression is defined by TRUTH bits in LUT Control x register (LUTCTRLx.TRUTH).

Figure 28-2. Truth Table Output Value Selection

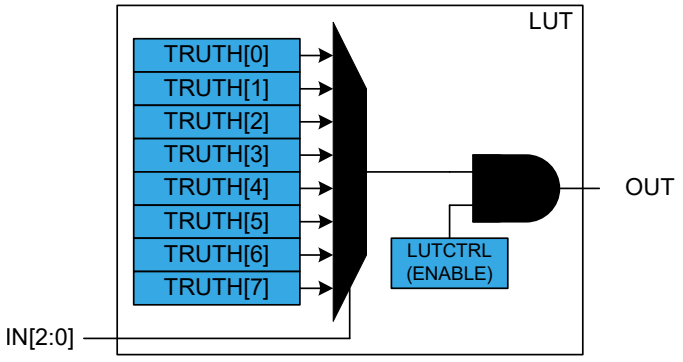


Table 28-1. Truth Table of LUT

IN[2]	IN[1]	IN[0]	OUT
0	0	0	TRUTH[0]
0	0	1	TRUTH[1]
0	1	0	TRUTH[2]
0	1	1	TRUTH[3]
1	0	0	TRUTH[4]
1	0	1	TRUTH[5]

© 2025 Microchip Technology Inc. and its subsidiaries

Table 28-1. Truth Table of LUT (continued)

IN[2]	IN[1]	IN[0]	OUT
1	1	0	TRUTH[6]
1	1	1	TRUTH[7]

28.6.2.4. Truth Table Inputs Selection

Input Overview

The inputs can be individually:

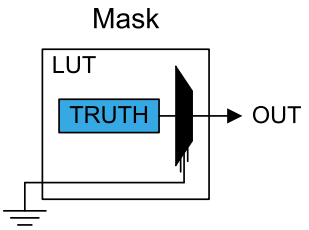
- Masked
- Driven by peripherals:
  - Analog Comparator output (AC)
  - Timer/Counters waveform outputs (TC)
  - Serial Communication output transmit interface (SERCOM)
- Driven by internal events from Event System
- Driven by other CCL sub-modules

The Input Selection for each input 'y' of LUT x is configured by writing the Input 'y' Source Selection bit in the LUT x Control register (LUTCTRLx.INSELy).

Masked Inputs (MASK)

When a LUT input is masked (LUTCTRLx.INSELy = MASK), the corresponding TRUTH input (IN) is internally tied to zero, as illustrated in this figure.

Figure 28-3. Masked Input Selection



Internal Feedback Inputs (FEEDBACK)

When selected (LUTCTRLx.INSELy = FEEDBACK), the Sequential (SEQ) output is used as input for the corresponding LUT.

The output from an internal sequential sub-module can be used as an input source for the LUT. See Figure 28-4 for an example for LUT0 and LUT1. The sequential selection for each LUT follows the formula:

$IN[0][i] = \mathrm{SEQ}$

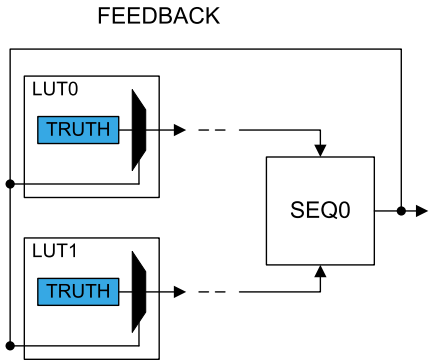
$IN[1][i] = \mathrm{SEQ}$

With *i* = [0,1,2] representing the LUT input index.

For additional information, see *Sequential Logic* from Related Links.

© 2025 Microchip Technology Inc. and its subsidiaries

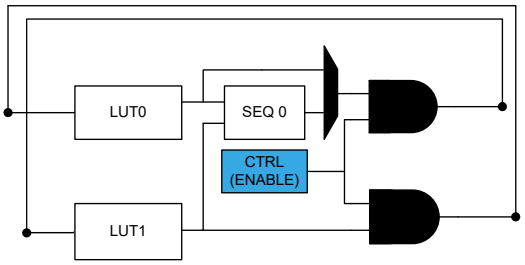
Figure 28-4. Feedback Input Selection



Linked LUT (LINK)

When selected (LUTCTRLx.INSELy=LINK), the subsequent LUT output is used as the LUT input (for example, LUT1 is the input for LUT0), as illustrated in the following figure.

Figure 28-5. Linked LUT Input Selection

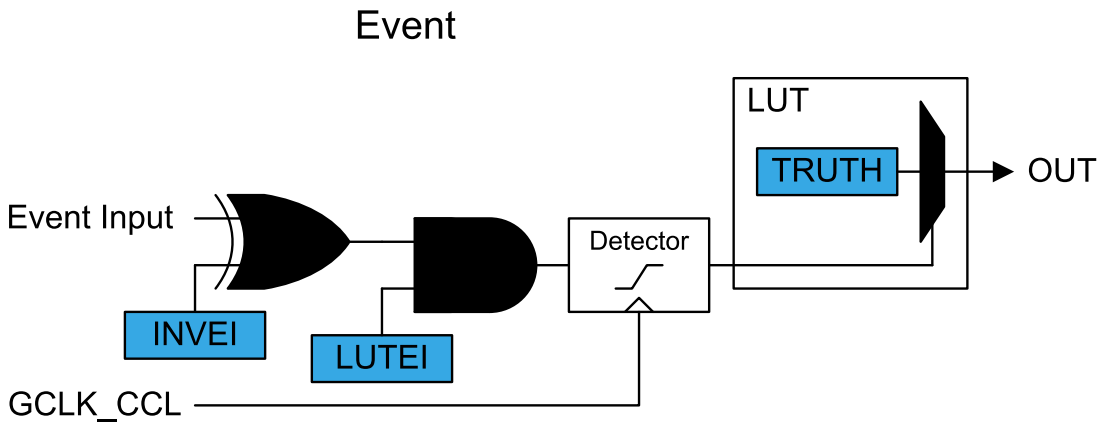


Internal Events Inputs Selection (EVENT)

Asynchronous events from the Event System can be used as an input selection, as illustrated in the following figure. For each LUT, one event input line is available and can be selected on each LUT input. Before enabling the event selection by writing LUTCTRLx.INSELy=EVENT, the Event System must be configured first.

By default, CCL includes an edge detector. When the event is received, an internal strobe is generated when a rising edge is detected. The pulse duration is one GCLK_CCL clock cycle.

Figure 28-6. Event Input Selection

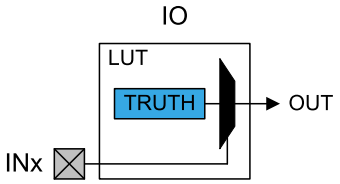


© 2025 Microchip Technology Inc. and its subsidiaries

I/O Pin Inputs (IO)

When the I/O pin is selected as the LUT input (LUTCTRLx.INSELy = IO), the corresponding LUT input is connected to the pin, as illustrated in the following figure.

Figure 28-7. I/O Pin Input Selection



Analog Comparator Inputs (AC)

The AC outputs can be used as an input source for the LUT (LUTCTRLx.INSELy=AC).

The analog comparator outputs are distributed following the formula:

$$IN[N][i] = AC[N \,\%\, ComparatorOutput\_Number]$$

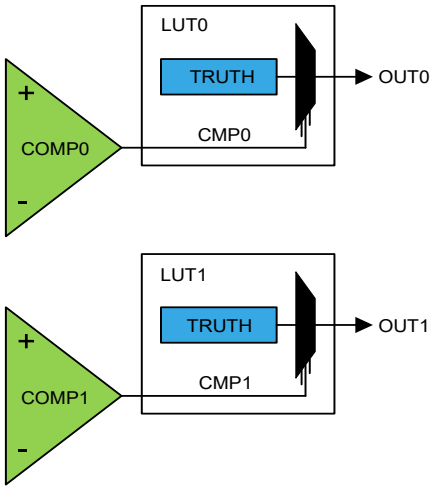
With N = [0,1] representing the LUT number and i = [0,1,2] representing the LUT input index.

Before selecting the comparator output, the AC must be configured first.

The output of comparator 0 is available on even LUTs ("LUT(2x)": LUT0) and the comparator 1 output is available on odd LUTs ("LUT(2x+1)": LUT1), as shown in the figure below.

The output of comparator 0 is available on even LUTs ("LUT(2x)": LUT0) and the comparator 1 output is available on odd LUTs ("LUT(2x+1)": LUT1), as illustrated in the following figure.

Figure 28-8. AC Input Selection



Timer/Counter Inputs (TC)

The TC waveform output WO[0] can be used as an input source for the LUT (LUTCTRLx.INSELy = TC). Only consecutive instances of the TC, that is, TCx and the subsequent TC(x+1), are available as default and alternative TC selections (for example, TC0 and TC1 are sources for LUT0, TC1 and TC2 are sources for LUT1). For an example for LUT0, see Figure 28-9. More generally, the Timer/Counter selection for each LUT follows the formula:

$$\mathrm{IN}[N][i] = DefaultTC[N \,\%\, \mathrm{TC\_Instance\_Number}]$$

$$\mathrm{IN}[N][i] = AlternativeTC[(N+1) \,\%\, \mathrm{TC\_Instance\_Number}]$$

With N = [0,1] representing the LUT number and *i* represents the LUT input index (i=0,1,2).

© 2025 Microchip Technology Inc. and its subsidiaries

Before selecting the waveform outputs, the TC must be configured first.

Figure 28-9. TC Input Selection

Timer/Counter for Control Application Inputs (TCC)

The TCC waveform outputs can be used as an input source for the LUT. Only WO[2:0] outputs can be selected and routed to the respective LUT input (that is, IN0 is connected to WO0, IN1 to WO1 and IN2 to WO2), as illustrated in the following figure.

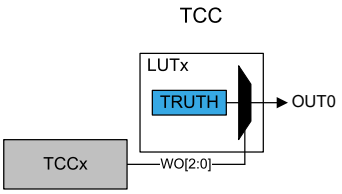
The TCC selection for each LUT follows the formula:

$\mathrm{IN}[N][i] = TCC[N\ \%\ \mathrm{TCC\_Instance\_Number}].\mathrm{WO}[\mathrm{i}]$

With N = [0,1] representing the LUT number and i represents the LUT input index (i = [0,1,2]).

Before selecting the waveform outputs, the TCC must be configured first.

Figure 28-10. TCC Input Selection



Serial Communication Output Transmit Inputs (SERCOM)

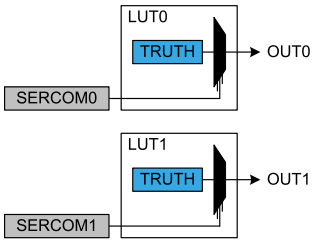
The serial engine transmitter output from the Serial Communication Interface (SERCOM TX, TXD for USART, MOSI for SPI) can be used as an input source for the LUT. The following figure illustrates an example for LUT0 and LUT1. The SERCOM selection for each LUT follows the formula:

$\mathrm{IN}[N][i] = SERCOM[N\ \%\ \mathrm{SERCOM\_Instance\_Number}]$

With N = [0,1] representing the LUT number and *i* = [0,1,2] representing the LUT input index.

Before selecting the SERCOM as an input source, the SERCOM must be configured first: the SERCOM TX signal must be output on SERCOMn/pad[0], which serves as the input pad to the CCL.

Figure 28-11. SERCOM Input Selection



Related Links

Sequential Logic

28.6.2.5. Filter

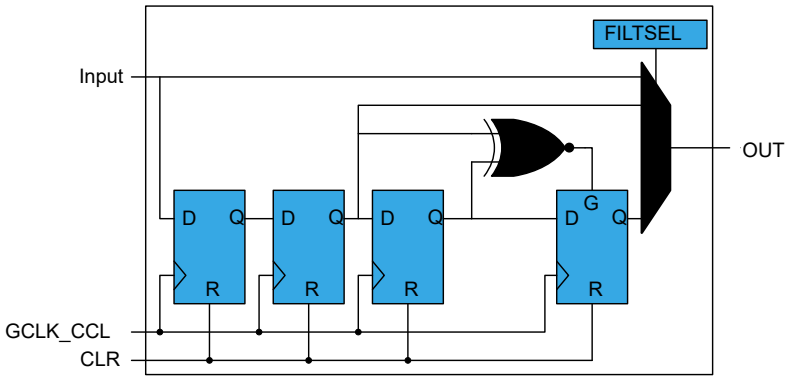
By default, the LUT output is a combinatorial function of the LUT inputs. This may cause some short glitches when the inputs change value. These glitches can be removed by clocking through filters, if demanded by application needs.

The Filter Selection bits in LUT Control register (LUTCTRLx.FILTSEL) define the synchronizer or digital filter options. When a filter is enabled, the OUT output will be delayed by two to five GCLK cycles. One APB clock after the corresponding LUT is disabled, all internal filter logic is cleared.

© 2025 Microchip Technology Inc. and its subsidiaries

Note: Events used as LUT input will also be filtered, if the filter is enabled.

Figure 28-12. Filter



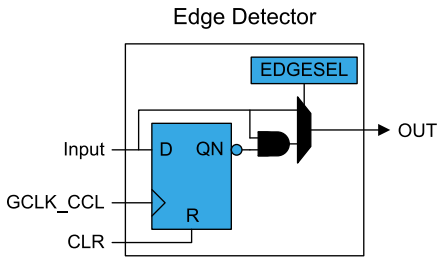
28.6.2.6. Edge Detector

The edge detector can be used to generate a pulse when detecting a rising edge on its input. To detect a falling edge, the TRUTH table must be inverted.

The edge detector is enabled by writing '1' to the Edge Selection bit in the LUT Control register (LUTCTRLx.EDGESEL). To avoid unpredictable behavior, either the filter or synchronizer must be enabled.

Edge detection is disabled by writing a '0' to LUTCTRLx.EDGESEL. After disabling a LUT, the corresponding internal Edge Detector logic is cleared one APB clock cycle later.

Figure 28-13. Edge Detector



28.6.2.7. Sequential Logic

Each LUT pair can be connected to the internal sequential logic, which can be configured to work as D flip flop, JK flip flop, gated D-latch or RS-latch by writing the Sequential Selection bits on the corresponding Sequential Control register (SEQCTRL.SEQSEL). Before using sequential logic, the GCLK_CCL clock and optionally each LUT filter or edge detector must be enabled.

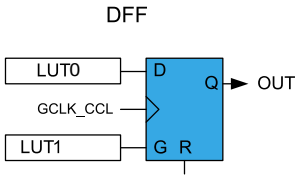
Note: While configuring the sequential logic, the even LUT must be disabled. When configured, the even LUT must be enabled.

Gated D Flip-Flop (DFF)

When the DFF is selected, the D-input is driven by the even LUT output LUT0, and the G-input is driven by the odd LUT output LUT1, as shown in the following figure.

© 2025 Microchip Technology Inc. and its subsidiaries

Figure 28-14. D Flip Flop



When the even LUT is disabled LUTCTRL0.ENABLE=0, the flip-flop is asynchronously cleared. The reset command (R) is kept enabled for one APB clock cycle. In all other cases, the flip-flop output (OUT) is refreshed on rising edge of the GCLK_CCL, as shown in the following table.

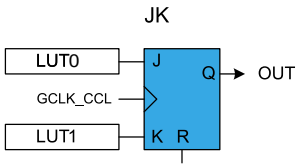
Table 28-2. DFF Characteristics

R	G	D	OUT
1	X	X	Clear
0	1	1	Set
		0	Clear
	0	X	Hold state (no change)

JK Flip-Flop (JK)

When this configuration is selected, the J-input is driven by the even LUT output LUT0, and the K-input is driven by the odd LUT output LUT1, as shown in the following figure.

Figure 28-15. JK Flip Flop



When the even LUT is disabled LUTCTRL0.ENABLE=0, the flip-flop is asynchronously cleared. The reset command (R) is kept enabled for one APB clock cycle. In all other cases, the flip-flop output (OUT) is refreshed on rising edge of the GCLK_CCL, as shown in the following table.

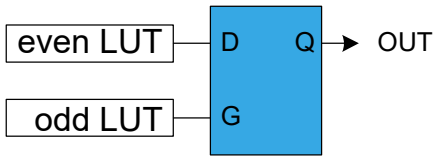
Table 28-3. JK Characteristics

R	J	K	OUT
1	X	X	Clear
0	0	0	Hold state (no change)
0	0	1	Clear
0	1	0	Set
0	1	1	Toggle

Gated D-Latch (DLATCH)

When the DLATCH is selected, the D-input is driven by the even LUT output LUT0, and the G-input is driven by the odd LUT output LUT1, as shown in the following figure.

Figure 28-16. D-Latch



© 2025 Microchip Technology Inc. and its subsidiaries

When the even LUT is disabled LUTCTRL0.ENABLE=0, the latch output will be cleared. The G-input is forced enabled for one more APB clock cycle, and the D-input to zero. In all other cases, the latch output (OUT) is refreshed as shown in the following table.

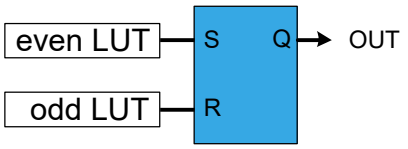
Table 28-4. D-Latch Characteristics

G	D	OUT
0	X	Hold state (no change)
1	0	Clear
1	1	Set

RS Latch (RS)

When this configuration is selected, the S-input is driven by the even LUT output LUT0, and the R-input is driven by the odd LUT output LUT1, as shown in the following figure.

Figure 28-17. RS-Latch



When the even LUT is disabled LUTCTRL0.ENABLE=0, the latch output will be cleared. The R-input is forced enabled for one more APB clock cycle and S-input to zero. In all other cases, the latch output (OUT) is refreshed as shown in the following table.

Table 28-5. RS-Latch Characteristics

S	R	OUT
0	0	Hold state (no change)
0	1	Clear
1	0	Set
1	1	Forbidden state

28.6.3. Events

The CCL can generate the following output events:

- LUTn where n=0-1 – Look-up Table Output Value

Writing a '1' to the LUT Control Event Output Enable bit (LUTCTRL.LUTEO) enables the corresponding output event. Writing a '0' to this bit disables the corresponding output event.

The CCL can take the following actions on an input event:

- INSELx where x=0-2 – The event is used as input for the TRUTH table. See *Event System (EVSYS)* from Related Links.

Writing a '1' to the LUT Control Event Input Enable bit (LUTCTRL.LUTEI) enables the corresponding action on input event. Writing a '0' to this bit disables the corresponding action on input event.

Related Links

Event System (EVSYS)

28.6.4. Sleep Mode Operation

When using the GCLK_CCL internal clocking, writing the Run In Standby bit in the Control register (CTRL.RUNSTDBY) to '1' allows GCLK_CCL to be enabled in Standby Sleep mode.

© 2025 Microchip Technology Inc. and its subsidiaries

If CTRL.RUNSTDBY=0, the GCLK_CCL is disabled in Standby Sleep mode. If the Filter, Edge Detector or Sequential logic is enabled, the LUT output is forced to zero in Standby mode. In all other cases, the TRUTH table decoder continues operation and the LUT output is refreshed accordingly.

© 2025 Microchip Technology Inc. and its subsidiaries

28.7. Register Summary

See the *CCL* module in the *Product Memory Mapping Overview* from Related Links for the base address.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRL	7:0		RUNSTDBY					ENABLE	SWRST
0x01 ... 0x03	Reserved									
0x04	SEQCTRL	7:0					SEQSEL[3:0]			
0x05 ... 0x07	Reserved									
0x08	LUTCTRL0	7:0	EDGESEL		FILTSEL[1:0]				ENABLE	
		15:8	INSEL1[3:0]				INSEL0[3:0]			
		23:16		LUTEO	LUTEI	INVEI	INSEL2[3:0]			
		31:24	TRUTH[7:0]							
0x0C	LUTCTRL1	7:0	EDGESEL		FILTSEL[1:0]				ENABLE	
		15:8	INSEL1[3:0]				INSEL0[3:0]			
		23:16		LUTEO	LUTEI	INVEI	INSEL2[3:0]			
		31:24	TRUTH[7:0]							

Related Links

Product Memory Mapping Overview

28.8. Register Description

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

Some registers are optionally write protected by the PAC. Optional PAC write protection is denoted by the PAC Write-Protection property in each individual register description. For details, see *Register Access Protection* from Related Links.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable protection is denoted by the Enable-Protected property in each individual register description.

Related Links

Register Access Protection

© 2025 Microchip Technology Inc. and its subsidiaries

28.8.1. Control

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

Note: CTRL register (except the bits ENABLE & SWRST) is Enable Protected when CCL.CTRL.ENABLE = 1.

Bit	7	6	5	4	3	2	1	0
		RUNSTDBY					ENABLE	SWRST
Access		R/W					R/W	W
Reset		0					0	0

Bit 6 – RUNSTDBY Run in Standby
This bit indicates if the GCLK_CCL clock must be kept running in Standby Sleep mode. The setting is ignored for configurations where the generic clock is not required. For details, see *Sleep Mode Operation* from Related Links.

Important: This bit must be written before enabling the CCL.

Value	Description
0	Generic clock is not required in Standby Sleep mode.
1	Generic clock is required in Standby Sleep mode.

Bit 1 – ENABLE Enable

Value	Description
0	The peripheral is disabled.
1	The peripheral is enabled.

Bit 0 – SWRST Software Reset
Writing a '0' to this bit has no effect.
Writing a '1' to this bit resets all registers in the CCL to their initial state.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

Related Links
Sleep Mode Operation

© 2025 Microchip Technology Inc. and its subsidiaries

28.8.2. Sequential Control

Name: SEQCTRL
Offset: 0x04
Reset: 0x00
Property: PAC Write-Protection, Enable-protected

Note: SEQCTRL register is Enable-protected. This register can only be written when "CCL.CTRL.ENABLE = 0 and CCL.LUTCTRL0.ENABLE = 0".

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

Bits 3:0 – SEQSEL[3:0] Sequential Selection
These bits select the sequential configuration:
Sequential Selection

Value	Name	Description
0x0	DISABLE	Sequential logic is disabled
0x1	DFF	D flip flop
0x2	JK	JK flip flop
0x3	LATCH	D latch
0x4	RS	RS latch
0x5 - 0xF	—	Reserved

© 2025 Microchip Technology Inc. and its subsidiaries

28.8.3. LUT Control n

Name: LUTCTRL
Offset: 0x08 + n*0x04 [n=0..1]
Reset: 0x00000000
Property: PAC Write-Protection, Enable-protected

Note: The LUTCTRLn register is Enable Protected except LUTCTRLn.ENABLE bit. This register can only be written when "CCL.CTRL.ENABLE = 0 and CCL.LUTCTRLn.ENABLE = 0".

Bit	31	30	29	28	27	26	25	24
	TRUTH[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
		LUTEO	LUTEI	INVEI	INSEL2[3:0]			
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	INSEL1[3:0]				INSEL0[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

Bits 31:24 – TRUTH[7:0] Truth Table
These bits define the value of truth logic as a function of inputs IN[2:0].

Bit 22 – LUTEO LUT Event Output Enable

Value	Description
0	LUT event output is disabled.
1	LUT event output is enabled.

Bit 21 – LUTEI LUT Event Input Enable

Value	Description
0	LUT incoming event is disabled.
1	LUT incoming event is enabled.

Bit 20 – INVEI Inverted Event Input Enable

Value	Description
0	Incoming event is not inverted.
1	Incoming event is inverted.

Bits 8:11, 12:15, 16:19 – INSELx LUT Input x Source Selection
These bits select the LUT input x source:

Value	Name	Description
0x0	MASK	Masked input

© 2025 Microchip Technology Inc. and its subsidiaries

Value	Name	Description
0x1	FEEDBACK	Feedback input source
0x2	LINK	Linked LUT input source
0x3	EVENT	Event input source
0x4	IO	I/O pin input source
0x5	AC	AC input source: CMP[0] (LUT0) / CMP[1] (LUT1)
0x6	TC	TC input source: TC0 WO[0] (LUT0) / TC1 WO[0] (LUT1)
0x7	ALTTC	Alternative TC input source: TC1 WO[0] (LUT0) / TC2 WO[0] (LUT1)
0x8	TCC	TCC input source: TCC0 (LUT0) / TCC1 (LUT1)
0x9	SERCOM	SERCOM input source: SERCOM0 PAD0 (LUT0) / SERCOM1 PAD0 (LUT1)
0xA-0xF	Reserved	Reserved

Bit 7 – EDGESEL Edge Selection

Value	Description
0	Edge detector is disabled.
1	Edge detector is enabled.

Bits 5:4 – FILTSEL[1:0] Filter Selection

These bits select the LUT output filter options:
Filter Selection

Value	Name	Description
0x0	DISABLE	Filter disabled
0x1	SYNCH	Synchronizer enabled
0x2	FILTER	Filter enabled
0x3	—	Reserved

Bit 1 – ENABLE LUT Enable

Value	Description
0	The LUT is disabled.
1	The LUT is enabled.

© 2025 Microchip Technology Inc. and its subsidiaries

29. Frequency Meter (FREQM)

29.1. Overview

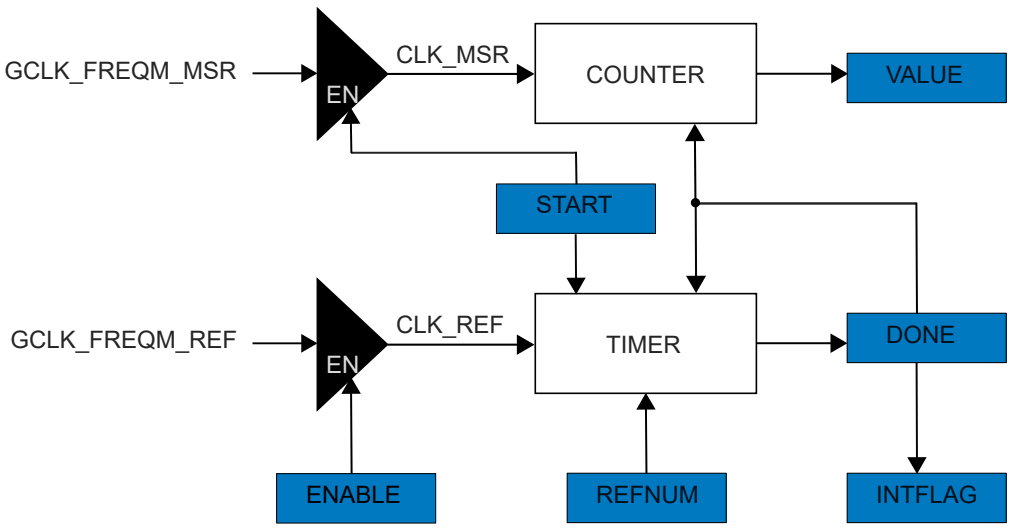
The Frequency Meter (FREQM) can be used to accurately measure the frequency of a clock by comparing it to a known reference clock.

29.2. Features

- Ratio can be measured with 24-bit accuracy
- Accurately measures the frequency of an input clock with respect to a reference clock
- Reference clock can be selected from the available GCLK_FREQM_REF sources
- Measured clock can be selected from the available GCLK_FREQM_MSR sources

29.3. Block Diagram

Figure 29-1. FREQM Block Diagram



29.4. Signal Description

Not applicable.

29.5. Product Dependencies

The following sections describe how the other parts of the system must be configured correctly to use this peripheral.

29.5.1. I/O Lines

Other than the internal GCLK sources, the clock provided on the REFI line is the external clock input source for GCLK_FREQM_REF/GCLK_FREQM_MSR, which can be used for measurement or reference clock sources. This requires the I/O pins to be configured using PPS configuration. See *I/O Ports and Peripheral Pin Select (PPS)* from Related Links.

© 2025 Microchip Technology Inc. and its subsidiaries

Related Links
I/O Ports and Peripheral Pin Select (PPS)

29.5.2. Power Management

The FREQM continues to operate in the Idle mode, where the selected source clock is running. The FREQM's interrupts can be used to wake up the device from the Idle mode. See *Power Management Unit (PMU)* from Related Links for more details on the different sleep modes.

Related Links
Power Management Unit (PMU)

29.5.3. Clocks

The clock for the FREQM bus interface (PB1_CLK) is enabled and disabled by the CRU generator.

Two generic clocks are used by the FREQM:

- Reference Clock (GCLK_FREQM_REF – GCLK_FREQM_REF is required to clock the internal reference timer, which acts as the frequency reference.
- Measurement Clock (GCLK_FREQM_MSR – GCLK_FREQM_MSR is required to clock a ripple counter for the frequency measurement. These clocks must be configured and enabled in the generic clock controller before using the FREQM.

See *Clock and Reset Unit (CRU)* from Related Links.

Related Links
Clock and Reset Unit (CRU)

29.5.4. DMA

Not applicable.

29.5.5. Interrupts

The interrupt request line is connected to the interrupt controller. Using FREQM interrupt requires the interrupt controller to be configured first.

29.5.6. Events

Not applicable.

29.5.7. Debug Operation

When the CPU is halted in Debug mode, the FREQM continues its normal operation. The FREQM cannot be halted when the CPU is halted in Debug mode. If the FREQM is configured in a way that requires it to be periodically serviced by the CPU, improper operation or data loss may result during debugging.

29.5.8. Register Access Protection

All registers with write access can be write protected optionally by the PAC, except the following registers:

- Control B register (CTRLB)
- Interrupt Flag Status and Clear register (INTFLAG)
- Status register (STATUS)

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

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

© 2025 Microchip Technology Inc. and its subsidiaries

29.6. Functional Description

29.6.1. Principle of Operation

FREQM counts the number of periods of the measured clock (GCLK_FREQM_MSR) with respect to the reference clock (GCLK_FREQM_REF). The measurement is done for a period of REFNUM/f_{CLK_REF} and stored in the Value register (VALUE.VALUE). REFNUM is the number of Reference clock cycles selected in the Configuration A register (CFGA.REFNUM).

The frequency of the measured clock, f_{CLK_MSR}, is calculated by

$f_{CLK\_MSR} = \left(\frac{VALUE}{REFNUM}\right) f_{CLK\_REF}$. The error can be maximum two measured clock cycles.

29.6.2. Basic Operation

29.6.2.1. Initialization

Before enabling FREQM, the device and peripheral must be configured:

- Each of the generic clocks (GCLK_FREQM_REF and GCLK_FREQM_MSR) must be configured and enabled.

> **Important:** The reference clock must be slower than the measurement clock.

- Write the number of Reference clock cycles for which the measurement is to be done in the Configuration A register (CFGA.REFNUM). This must be a non-zero number.

The following register is enable-protected, meaning that it can only be written when the FREQM is disabled (CTRLA.ENABLE=0):

- Configuration A register (CFGA)

Enable protection is denoted by the Enable-Protected property in the register description.

29.6.2.2. Enabling, Disabling and Resetting

The FREQM is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The peripheral is disabled by writing CTRLA.ENABLE=0.

The FREQM is Reset by writing a '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). On software Reset, all registers in the FREQM reset to their initial state, and the FREQM is disabled.

Then ENABLE and SWRST bits are write-synchronized.

29.6.2.3. Measurement

In the Configuration A register, the Number of Reference Clock Cycles field (CFGA.REFNUM) selects the duration of the measurement. The measurement is given in number of GCLK_FREQM_REF periods.
Note: The REFNUM field must be written before the FREQM is enabled.

After the FREQM is enabled, writing a '1' to the START bit in the Control B register (CTRLB.START) starts the measurement. The BUSY bit in Status register (STATUS.BUSY) is set when the measurement starts, and cleared when the measurement is complete.

There is also an interrupt request for Measurement Done: When the Measurement Done bit in Interrupt Enable Set register (INTENSET.DONE) is '1' and a measurement is finished, the Measurement Done bit in the Interrupt Flag Status and Clear register (INTFLAG.DONE) will be set and an interrupt request is generated.

The result of the measurement can be read from the Value register (VALUE.VALUE). The frequency of the measured clock GCLK_FREQM_MSR is then:

© 2025 Microchip Technology Inc. and its subsidiaries

$$f_{CLK\_MSR} = \left(\frac{VALUE}{REFNUM}\right) f_{CLK\_REF}$$

Notes:

1. In order to make sure the measurement result (VALUE.VALUE[23:0]) is valid, the overflow status (STATUS.OVF) must be checked.
2. Due to asynchronous operations, the VALUE Error measurement can be up to two samples.

If an overflow condition occurred, indicated by the overflow bit in the STATUS register (STATUS.OVF), either the number of reference clock cycles must be reduced (CFGA.REFNUM) or a faster reference clock must be configured. Once the configuration is adjusted, clear the overflow status by writing a '1' to STATUS.OVF. Then, another measurement can be started by writing a '1' to CTRLB.START.

Note: See *CFGA*, *CTRLB*, *STATUS*, *INTENSET*, *INTFLAG*, *VALUE* registers in the *Register Summary - FREQM* from Related Links.

Related Links

Register Summary

29.6.3. DMA Operation

Not applicable.

29.6.4. Interrupts

The FREQM has one interrupt source:

- DONE: A frequency measurement is done.

The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. The interrupt can be 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. The status of enabled interrupts can be read from either INTENSET or INTENCLR.

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled or the FREQM is reset. See the *INTFLAG* register Related Links for details on how to clear Interrupt flags. All interrupt requests from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. The user must read the INTFLAG register to determine which interrupt condition is present.

This interrupt is a synchronous wake-up source.

Note: Interrupts must be globally enabled for interrupt requests to be generated.

Related Links

INTFLAG

INTFLAG - Interrupt Flag Status and Clear Register

29.6.5. Events

Not applicable.

29.6.6. Sleep Mode Operation

The FREQM continues to operate in Idle mode or Sleep mode where the selected source clock is running. The FREQM's interrupts can be used to wake-up the device from Idle mode or Sleep mode.

For lowest chip power consumption in sleep modes, FREQM must be disabled before entering a Standby Sleep mode.

© 2025 Microchip Technology Inc. and its subsidiaries

29.6.7. Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits and registers are write-synchronized:

- Software Reset bit in Control A register (CTRLA.SWRST)
- Enable bit in Control A register (CTRLA.ENABLE)

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

© 2025 Microchip Technology Inc. and its subsidiaries

29.7. Register Summary

See the *FREQM* module in the *Product Memory Mapping Overview* from Related Links for the base address.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0							ENABLE	SWRST
0x01	CTRLB	7:0								START
0x02	CFGA	7:0	REFNUM[7:0]							
		15:8								
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
		15:8								
		23:16								
		31:24								
0x10	VALUE	7:0	VALUE[7:0]							
		15:8	VALUE[15:8]							
		23:16	VALUE[23:16]							
		31:24								

Related Links

Product Memory Mapping Overview

29.8. Register Description

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

Some registers 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 PAC. Optional PAC write protection is denoted by the PAC Write-Protection property in each individual register description.

© 2025 Microchip Technology Inc. and its subsidiaries

29.8.1. CTRLA - Control A Register

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

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

Bit 1 – ENABLE Enable

Due to synchronization, there is a delay from writing CTRLA.ENABLE until the peripheral is enabled or disabled. The value written to CTRLA.ENABLE reads back immediately, and the ENABLE bit in the Synchronization Busy register (SYNCBUSY.ENABLE) is set. SYNCBUSY.ENABLE is cleared when the operation is complete. This bit is not enable-protected.

Value	Description
0	The peripheral is disabled.
1	The peripheral is enabled.

Bit 0 – SWRST Software Reset

Writing a '0' to this bit has no effect.
Writing a '1' to this bit resets all registers in the FREQM to their initial state, and the FREQM is disabled. Writing a '1' to this bit will always take precedence, meaning that all other writes in the same write operation are discarded.
Due to synchronization, there is a delay from writing CTRLA.SWRST until the Reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST are cleared when the Reset is complete. This bit is not enable-protected.

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

© 2025 Microchip Technology Inc. and its subsidiaries

29.8.2. CTRLB - Control B Register

Name: CTRLB
Offset: 0x01
Reset: 0x00
Property: –

Bit	7	6	5	4	3	2	1	0
								START
Access								W
Reset								0

Bit 0 – START Start Measurement

Value	Description
0	Writing a '0' has no effect.
1	Writing a '1' starts a measurement.

© 2025 Microchip Technology Inc. and its subsidiaries

29.8.3. CFGA - Configuration A Register

Name: CFGA
Offset: 0x02
Reset: 0x0000
Property: PAC Write-Protection, Enable-protected

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	REFNUM[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 – REFNUM[7:0] Number of Reference Clock Cycles
Selects the duration of a measurement in the number of GCLK_FREQM_REF cycles. This must be a non-zero value, in other words, 0x01 (one cycle) to 0xFF (255 cycles).

© 2025 Microchip Technology Inc. and its subsidiaries

29.8.4. INTENCLR - Interrupt Enable Clear Register

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

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

Bit 0 – DONE Measurement Done Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Measurement Done Interrupt Enable bit, which disables the Measurement Done interrupt.

Value	Description
0	The Measurement Done interrupt is disabled.
1	The Measurement Done interrupt is enabled.

© 2025 Microchip Technology Inc. and its subsidiaries

29.8.5. INTENSET - Interrupt Enable Set Register

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

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

Bit 0 – DONE Measurement Done Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit sets the Measurement Done Interrupt Enable bit, which enables the Measurement Done interrupt.

Value	Description
0	The Measurement Done interrupt is disabled.
1	The Measurement Done interrupt is enabled.

© 2025 Microchip Technology Inc. and its subsidiaries

29.8.6. INTFLAG - Interrupt Flag Status and Clear Register

Name: INTFLAG
Offset: 0x0A
Reset: 0x00
Property: –

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

Bit 0 – DONE Measurement Done
This flag is cleared by writing a '1' to it.
This flag is set when a new measurement is completed.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the DONE Interrupt flag.

© 2025 Microchip Technology Inc. and its subsidiaries

29.8.7. STATUS - Status Register

Name: STATUS
Offset: 0x0B
Reset: 0x00
Property: –

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

Bit 1 – OVF Sticky Count Value Overflow
This bit is cleared by writing a '1' to it.
This bit is set when an overflow condition occurs to the value counter.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the OVF status.
Clearing the CTRLA.ENABLE register bit will clear the OVF bit.

Bit 0 – BUSY FREQM Status

Value	Description
0	No ongoing frequency measurement.
1	Frequency measurement is ongoing.

Note: If the measurement clock stalls (or is very slow) during a measurement slot, the STATUS.BUSY will never de-assert and the DONE interrupt will not be raised.

© 2025 Microchip Technology Inc. and its subsidiaries

29.8.8. SYNCBUSY - Synchronization Busy Register

Name: SYNCBUSY
Offset: 0x0C
Reset: 0x00000000
Property: –

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							ENABLE	SWRST
Access							R	R
Reset							0	0

Bit 1 – ENABLE Enable
This bit is cleared when the synchronization of CTRLA.ENABLE is complete.
This bit is set when the synchronization of CTRLA.ENABLE is started.

Bit 0 – SWRST Synchronization Busy
This bit is cleared when the synchronization of CTRLA.SWRST is complete.
This bit is set when the synchronization of CTRLA.SWRST is started.

© 2025 Microchip Technology Inc. and its subsidiaries

29.8.9. VALUE - Value Register

Name: VALUE
Offset: 0x10
Reset: 0x00000000
Property: –

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

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

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

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

Bits 23:0 – VALUE[23:0] Measurement Value
Result from measurement.

© 2025 Microchip Technology Inc. and its subsidiaries

30. Event System (EVSYS)

30.1. Overview

The Event System (EVSYS) allows autonomous, low-latency and configurable communication between peripherals.

Several peripherals can be configured to generate and/or respond to signals known as events. The exact condition to generate an event, or the action taken upon receiving an event, is specific to each peripheral. Peripherals that respond to events are called event users. Peripherals that generate events are called event generators. A peripheral can have one or more event generators and can have one or more event users.

Communication is made without CPU intervention and without consuming system resources such as bus or RAM bandwidth. This reduces the load on the CPU and other system resources, compared to a traditional interrupt-based system.

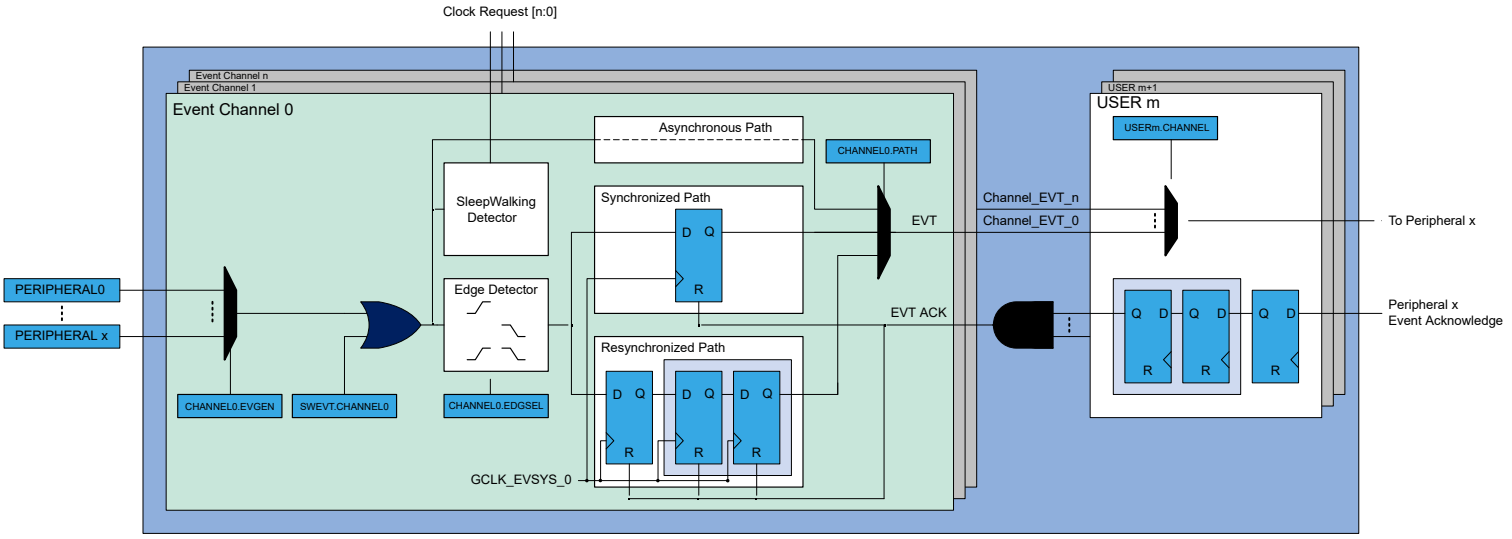
30.2. Features

- 32 Configurable Event Channels:
  - All channels can be connected to any event generator
  - All channels provide a pure asynchronous path
  - Twelve channels provide a resynchronized or synchronous path
- 89 Event Generators
- 59 Event Users
- Configurable Edge Detector
- Peripherals Can be Event Generators, Event Users or Both
- Sleep-Walking and Interrupt for Operation in Sleep Modes
- Software Event Generation
- Each Event User Can Choose which Channel to Respond to
- Optional Static or Round-Robin Interrupt Priority Arbitration

© 2025 Microchip Technology Inc. and its subsidiaries

30.3. Block Diagram

Figure 30-1. Event System Block Diagram



30.4. Product Dependencies

The following sections describe how the other parts of the system must be configured correctly to use this peripheral.

30.4.1. I/O Lines

Not applicable.

30.4.2. Power Management

The EVSYS can be used to wake up the CPU from the Idle and Standby Sleep modes, even if the clock used by the EVSYS channel and the EVSYS bus clock are disabled. See *Power Management Unit (PMU)* from Related Links for details on the different sleep modes.

In sleep modes, although the clock for the EVSYS is stopped, the device still can wake up the EVSYS clock. Some event generators can generate an event when their clocks are stopped. The generic clock for the channel (GCLK_EVSYS_CH_n) is restarted, if that channel uses a synchronized path or a resynchronized path. It does not need to wake the system from sleep.

Related Links

Power Management Unit (PMU)

30.4.3. Clocks

The EVSYS bus clock (PB2_CLK) can be enabled and disabled in the CRU. Each EVSYS channel, which can be configured as synchronous or resynchronized, has a dedicated generic clock (GCLK_EVSYS_CH_n). These are used for event detection and propagation for each channel. These clocks must be configured and enabled in the generic clock generator before using the EVSYS. See *Peripheral Clock Generation (GCLK)* from Related Links for more details on clock generation.

Important: Only EVSYS channel 0 to 11 can be configured as synchronous or resynchronized.

© 2025 Microchip Technology Inc. and its subsidiaries

Related Links
Peripheral Clock Generation (GCLK)

30.4.4. DMA
Not applicable.

30.4.5. Interrupts
The interrupt request line is connected to the interrupt controller. Using the EVSYS interrupts requires the interrupt controller to be configured first (see *Nested Vector Interrupt Controller (NVIC)* from Related Links).

Related Links
Nested Vector Interrupt Controller (NVIC)

30.4.6. Events
Not applicable.

30.4.7. Debug Operation
When the CPU is halted in Debug mode, this peripheral will continue normal operation. If the peripheral is configured to require periodical service by the CPU through interrupts or similar, improper operation or data loss may result during debugging. This peripheral can be forced to halt operation during debugging.

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

- Interrupt Flag (INTFLAG) register
- Channel Pending Interrupt (INTPEND)
- Channel n Interrupt Flag Status and Clear (CHINTFLAGn)

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

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

30.4.9. Analog Connections
Not applicable.

30.5. Functional Description

30.5.1. Principle of Operation
The Event System consists of channels which route the internal events from peripherals (generators) to other internal peripherals. Each event generator can be selected as source for multiple channels, but a channel cannot be set to use multiple event generators at the same time.

A channel path can be configured in asynchronous, synchronous or resynchronized mode of operation. The mode of operation must be selected based on the requirements of the application.

When using synchronous or resynchronized path, the Event System includes options to transfer events to users when rising, falling or both edges are detected on event generators.

See *Channel Path* from Related Links.

Related Links
Channel Path

© 2025 Microchip Technology Inc. and its subsidiaries

30.5.2. Basic Operation

30.5.2.1. Initialization

Before enabling event routing within the system, the Event Users Multiplexer and Event Channels must be selected in the Event System (EVSYS), and the two peripherals that generate and use the event must be configured. Follow these steps to configure the event:

1. In the peripheral generating the event, enable the output of the event by writing a '1' to the respective Event Output Enable bit (EO) in the peripheral's Event Control register, for example, AC.EVCTRL.WINEO0 or RTC.EVCTRL.OVFEO.
2. Configure the EVSYS:
   a. Configure the Event User multiplexer by writing the respective EVSYS.USERm register; see *User Multiplexer Setup* from Related Links.
   b. Configure the Event Channel by writing the respective EVSYS.CHANNELn register; see *Event System Channel* from Related Links.
3. Configure the action to be executed by the event user peripheral by writing to the Event Action bits (EVACT) in the respective Event control register, for example, TC.EVCTRL.EVACT.
   Note: This step is not applicable for all the peripherals.
4. In the event it is a user peripheral, enable event input by writing a '1' to the respective Event Input Enable bit ("EI") in the peripheral's Event Control register, for example, AC.EVCTRL.IVEI0.
   Note: The ADC, CVD and ZB modules do not have the EVCTRL register; therefore, the EVCTRL register configuration is not applicable for these modules.

Related Links

User Multiplexer Setup
Event System Channel

30.5.2.2. Enabling, Disabling and Resetting

The EVSYS is always enabled.

The EVSYS is reset by writing a '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the EVSYS reset to their initial state and all ongoing events are canceled.

See *CTRLA* from Related Links.

Related Links

CTRLA
Control A

30.5.2.3. User Multiplexer Setup

The user multiplexer defines the channel to be connected to which event user. Each user multiplexer is dedicated to one event user. A user multiplexer receives all event channels output and must be configured to select one of these channels, as shown in Block Diagram section. The channel is selected with the Channel bit group in the User register (USERm.CHANNEL).

The user multiplexer must always be configured before the channel. A list of all available event users is found in the User (USERm) register description.

30.5.2.4. Event System Channel

An event channel can select one event from a list of event generators. Depending on configuration, the selected event could be synchronized, resynchronized or asynchronously sent to the users. When synchronization or resynchronization is required, the channel includes an internal edge detector, allowing the Event System to generate internal events when rising, falling or both edges are detected on the selected event generator.

An event channel is able to generate internal events for the specific software commands. A channel block diagram is shown in *Block Diagram* section.

© 2025 Microchip Technology Inc. and its subsidiaries

30.5.2.5. Event Generators

Each event channel can receive the events from all event generators. All event generators are listed in the Event Generator bit field in the Channel n register (CHANNELn.EVGEN). For details on event generation, refer to the corresponding module chapter. The channel event generator is selected by the Event Generator bit group in the Channel register (CHANNELn.EVGEN). By default, the channels are not connected to any event generators (in other words, CHANNELn.EVGEN=0).

30.5.2.6. Channel Path

There are different ways to propagate the event from an event generator:

- Asynchronous path
- Resynchronized path

The path is decided by writing to the Path Selection bit group of the Channel register (CHANNELn.PATH).

Asynchronous Path

When using the asynchronous path, the events are propagated from the event generator to the event user without intervention from the Event System. The GCLK for this channel (GCLK_EVSYS_CH_n) is not mandatory, meaning that an event will be propagated to the user without any clock latency.

When the asynchronous path is selected, the channel cannot generate any interrupts, and the Channel x Status register (CHSTATUSx) is always zero. The edge detection is not required and must be disabled by software. Each peripheral event user has to select which event edge must trigger internal actions. For further details, refer to each peripheral chapter description.

Resynchronized Path

The resynchronized path are used when the event generator and the event channel do not share the same generator for the generic clock. When the resynchronized path is used, resynchronization of the event from the event generator is done in the channel.

When the resynchronized path is used, the channel is able to generate interrupts. The channel status bits in the Channel Status register (CHSTATUS) are also updated and available for use.

30.5.2.7. Edge Detection

When synchronous or resynchronized paths are used, edge detection must be enabled. The event system can execute edge detection in three different ways:

- Generate an event only on the rising edge
- Generate an event only on the falling edge
- Generate an event on rising and falling edges.

Edge detection is selected by writing to the Edge Selection bit group of the Channel register (CHANNELn.EDGSEL).

30.5.2.8. Event Latency

The latency from the event generator to the event user depends on the channel's configuration:

- Asynchronous Path – The maximum routing latency of an external event is related to the internal signal routing and it is device dependent.
- Synchronous Path – The maximum routing latency of an external event is one GCLK_EVSYS_CH_n clock cycle.
- Resynchronized Path – The maximum routing latency of an external event is three GCLK_EVSYS_CH_n clock cycles.

The maximum propagation latency of a user event to the peripheral clock core domain is three peripheral clock cycles.

© 2025 Microchip Technology Inc. and its subsidiaries

The event generators, event channel and event user clocks ratio must be selected in relation with the internal event latency constraints. Events propagation or event actions in peripherals may be lost if the clock setup violates the internal latencies.

30.5.2.9. The Overrun Channel n Interrupt

The Overrun Channel n Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAGn.OVR) 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 INTFLAGn.OVR is always read as zero.

30.5.2.10. The Event Detected Channel n Interrupt

The Event Detected Channel n Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAGn.EVD) 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 an asynchronous path, the INTFLAGn.EVD is always zero.

In synchronous mode, including RESYNC mode, spurious event detections can be generated when accessing peripherals used in conjunction with the EVSYS. To prevent spurious EVSYS detections, EVSYS must be write-protected by configuring the WRCTRL register in the PAC before use.

30.5.2.11. Channel Status

The Channel Status register (CHSTATUS) shows the status of the channels when using a synchronous or resynchronized path. There are two different status bits in CHSTATUS for each of the available channels:

- The CHSTATUSn.BUSYCH bit will be set when an event on the corresponding channel n has not been handled by all event users connected to that channel.
- The CHSTATUSn.RDYUSR bit will be set when all event users connected to the corresponding channel are ready to handle incoming events on that channel.

30.5.2.12. Software Event

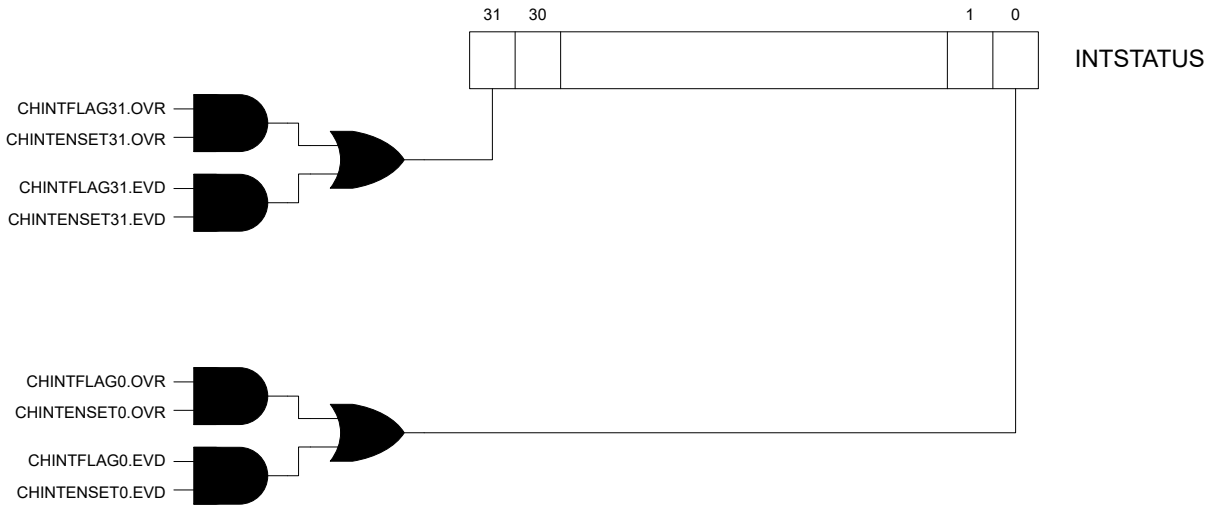
A software event can be initiated on a channel by writing a '1' to the Software Event bit in the Channel register (CHANNELm.SWEVT). Then, the software event can be serviced as any event generator; in other words, when the bit is set to '1', an event is generated on the respective channel.

30.5.2.13. Interrupt Status and Interrupts Arbitration

The Interrupt Status register stores all channels with pending interrupts, as illustrated in the following figure.

© 2025 Microchip Technology Inc. and its subsidiaries

Figure 30-2. Interrupt Status Register

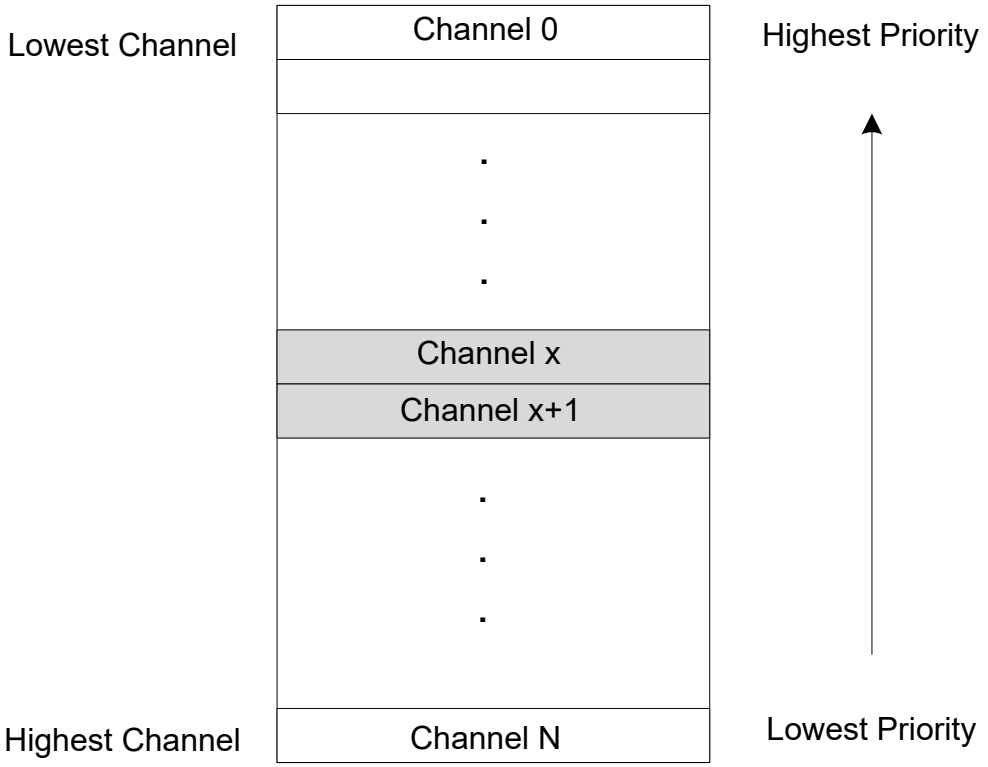


The Event System can arbitrate between all channels with pending interrupts. The arbiter can be configured to prioritize statically or dynamically the incoming events. The priority is evaluated each time a new channel has an interrupt pending or an interrupt is cleared. The Channel Pending Interrupt register (INTPEND) provides the channel number with the highest interrupt priority, the corresponding channel interrupt flags and status bits.

By default, static arbitration is enabled (PRICTRL.RREN is '0'), the arbiter will prioritize a low channel number over a high channel number as illustrated in Figure 30-3. When using the status scheme, there is a risk of high channel numbers never being granted access by the arbiter. This can be avoided using a dynamic arbitration scheme.

© 2025 Microchip Technology Inc. and its subsidiaries

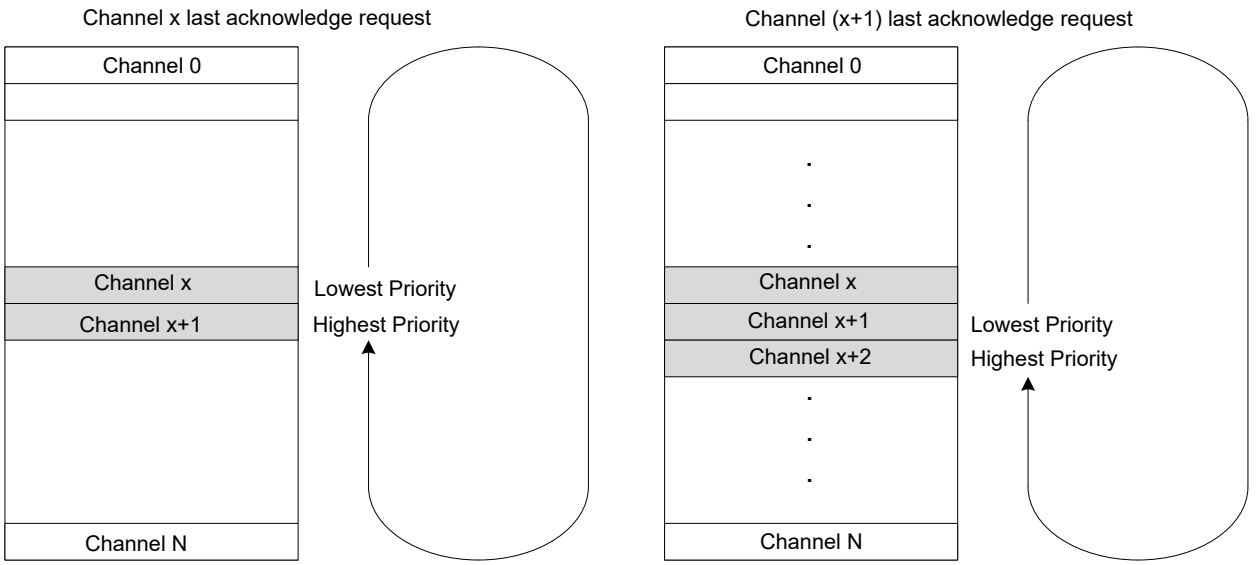
Figure 30-3. Static Priority



The dynamic arbitration scheme available in the Event System is round-robin. Round-robin arbitration is enabled by writing PRICTRL.RREN to one. With the round-robin scheme, the channel number of the last channel being granted access will have the lowest priority the next time the arbiter has to grant access to a channel, as illustrated in the following figure. The channel number of the last channel being granted access, will be stored in the Channel Priority Number bit group in the Priority Control register (PRICTRL.PRI).

© 2025 Microchip Technology Inc. and its subsidiaries

Figure 30-4. Round-Robin Scheduling



The Channel Pending Interrupt register (INTPEND) also offers the possibility to indirectly clear the interrupt flags of a specific channel. Writing a flag to one in this register, clears the corresponding interrupt flag of the channel specified by the INTPEND.ID bits.

30.5.3. Interrupts

The EVSYS has the following interrupt sources for each channel:

- Overrun Channel n interrupt (OVR)
- Event Detected Channel n interrupt (EVD)

These interrupts events are asynchronous wake-up sources.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the corresponding Channel n Interrupt Flag Status and Clear (CHINTFLAG) register is set when the interrupt condition occurs.

Note: Interrupts must be globally enabled to allow the generation of interrupt requests.

Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Channel n Interrupt Enable Set (CHINTENSET) register and disabled by writing a '1' to the corresponding bit in the Channel n Interrupt Enable Clear (CHINTENCLR) register. An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled or the Event System is reset. All interrupt requests are ORed together on system level to generate one combined interrupt request to the NVIC.

The user must read the Channel Interrupt Status (INTSTATUS) register to identify the channels with pending interrupts and must read the Channel n Interrupt Flag Status and Clear (CHINTFLAG) register to determine which interrupt condition is present for the corresponding channel. It is also possible to read the Interrupt Pending register (INTPEND), which provides the highest priority channel with the pending interrupt and the respective interrupt flags.

30.5.4. Sleep Mode Operation

The Event System can generate interrupts to wake up the device from the Idle or Standby Sleep mode.

© 2025 Microchip Technology Inc. and its subsidiaries

To be able to run in standby, the Run in Standby bit in the Channel register (CHANNELn.RUNSTDBY) must be set to '1'. When the Generic Clock On Demand bit in the Channel register (CHANNELn.ONDEMAND) is set to '1' and the event generator is detected, the event channel will request its clock (GCLK_EVSYS_CHANNEL_n). The event latency for a resynchronized channel path increases by two GCLK_EVSYS_CHANNEL_n clock (in other words, up to five GCLK_EVSYS_CHANNEL_n clock cycles).

A channel will behave differently in different sleep modes depending on whether using CHANNELn.RUNSTDBY or CHANNELn.ONDEMAND:

Table 30-1. Event Channel Sleep Behavior

CHANNELn.PATH	CHANNELn. ONDEMAND	CHANNELn. RUNSTDBY	Sleep Behavior
ASYNC	0	0	Only run in Idle modes if an event must be propagated. Disabled in Standby Sleep mode.
SYNC/RESYNC	0	1	Run in both Idle and Standby Sleep modes.
SYNC/RESYNC	1	0	Only run in Idle modes if an event must be propagated. Disabled in Standby Sleep mode. Two GCLK_EVSYS_n latency added in RESYNC path before the event is propagated internally.
SYNC/RESYNC	1	1	Run in both Idle and Standby Sleep modes. Two GCLK_EVSYS_n latency added in RESYNC path before the event is propagated internally.

© 2025 Microchip Technology Inc. and its subsidiaries

30.6. Register Summary

See the *EVSYS* module in the *Product Memory Mapping Overview* from Related Links for the base address.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0								SWRST
0x01 ... 0x03	Reserved									
0x04	SWEVT	7:0	CHANNEL7	CHANNEL6	CHANNEL5	CHANNEL4	CHANNEL3	CHANNEL2	CHANNEL1	CHANNEL0
		15:8	CHANNEL15	CHANNEL14	CHANNEL13	CHANNEL12	CHANNEL11	CHANNEL10	CHANNEL9	CHANNEL8
		23:16	CHANNEL23	CHANNEL22	CHANNEL21	CHANNEL20	CHANNEL19	CHANNEL18	CHANNEL17	CHANNEL16
		31:24	CHANNEL31	CHANNEL30	CHANNEL29	CHANNEL28	CHANNEL27	CHANNEL26	CHANNEL25	CHANNEL24
0x08	PRICTRL	7:0	RREN			PRI[4:0]				
0x09 ... 0x0F	Reserved									
0x10	INTPEND	7:0					ID[3:0]			
		15:8	BUSY	READY					EVD	OVR
0x12 ... 0x13	Reserved									
0x14	INTSTATUS	7:0	CHINT7	CHINT6	CHINT5	CHINT4	CHINT3	CHINT2	CHINT1	CHINT0
		15:8					CHINT11	CHINT10	CHINT9	CHINT8
		23:16								
		31:24								
0x18	BUSYCH	7:0	BUSYCH7	BUSYCH6	BUSYCH5	BUSYCH4	BUSYCH3	BUSYCH2	BUSYCH1	BUSYCH0
		15:8					BUSYCH11	BUSYCH10	BUSYCH9	BUSYCH8
		23:16								
		31:24								
0x1C	READYUSR	7:0	READYUSR7	READYUSR6	READYUSR5	READYUSR4	READYUSR3	READYUSR2	READYUSR1	READYUSR0
		15:8					READYUSR11	READYUSR10	READYUSR9	READYUSR8
		23:16								
		31:24								
0x20	CHANNEL0	7:0		EVGEN[6:0]						
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x24	CHINTENCLR0	7:0							EVD	OVR
0x25	CHINTENSET0	7:0							EVD	OVR
0x26	CHINTFLAG0	7:0							EVD	OVR
0x27	CHSTATUSn0	7:0							BUSYCH	RDYUSR
0x28	CHANNEL1	7:0		EVGEN[6:0]						
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x2C	CHINTENCLR1	7:0							EVD	OVR
0x2D	CHINTENSET1	7:0							EVD	OVR
0x2E	CHINTFLAG1	7:0							EVD	OVR
0x2F	CHSTATUSn1	7:0							BUSYCH	RDYUSR
0x30	CHANNEL2	7:0		EVGEN[6:0]						
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x34	CHINTENCLR2	7:0							EVD	OVR
0x35	CHINTENSET2	7:0							EVD	OVR
0x36	CHINTFLAG2	7:0							EVD	OVR
0x37	CHSTATUSn2	7:0							BUSYCH	RDYUSR
0x38	CHANNEL3	7:0		EVGEN[6:0]						
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								

© 2025 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x3C	CHINTENCLR3	7:0							EVD	OVR
0x3D	CHINTENSET3	7:0							EVD	OVR
0x3E	CHINTFLAG3	7:0							EVD	OVR
0x3F	CHSTATUSn3	7:0							BUSYCH	RDYUSR
0x40	CHANNEL4	7:0		EVGEN[6:0]						
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x44	CHINTENCLR4	7:0							EVD	OVR
0x45	CHINTENSET4	7:0							EVD	OVR
0x46	CHINTFLAG4	7:0							EVD	OVR
0x47	CHSTATUSn4	7:0							BUSYCH	RDYUSR
0x48	CHANNEL5	7:0		EVGEN[6:0]						
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x4C	CHINTENCLR5	7:0							EVD	OVR
0x4D	CHINTENSET5	7:0							EVD	OVR
0x4E	CHINTFLAG5	7:0							EVD	OVR
0x4F	CHSTATUSn5	7:0							BUSYCH	RDYUSR
0x50	CHANNEL6	7:0		EVGEN[6:0]						
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x54	CHINTENCLR6	7:0							EVD	OVR
0x55	CHINTENSET6	7:0							EVD	OVR
0x56	CHINTFLAG6	7:0							EVD	OVR
0x57	CHSTATUSn6	7:0							BUSYCH	RDYUSR
0x58	CHANNEL7	7:0		EVGEN[6:0]						
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x5C	CHINTENCLR7	7:0							EVD	OVR
0x5D	CHINTENSET7	7:0							EVD	OVR
0x5E	CHINTFLAG7	7:0							EVD	OVR
0x5F	CHSTATUSn7	7:0							BUSYCH	RDYUSR
0x60	CHANNEL8	7:0		EVGEN[6:0]						
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x64	CHINTENCLR8	7:0							EVD	OVR
0x65	CHINTENSET8	7:0							EVD	OVR
0x66	CHINTFLAG8	7:0							EVD	OVR
0x67	CHSTATUSn8	7:0							BUSYCH	RDYUSR
0x68	CHANNEL9	7:0		EVGEN[6:0]						
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x6C	CHINTENCLR9	7:0							EVD	OVR
0x6D	CHINTENSET9	7:0							EVD	OVR
0x6E	CHINTFLAG9	7:0							EVD	OVR
0x6F	CHSTATUSn9	7:0							BUSYCH	RDYUSR
0x70	CHANNEL10	7:0		EVGEN[6:0]						
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x74	CHINTENCLR10	7:0							EVD	OVR
0x75	CHINTENSET10	7:0							EVD	OVR
0x76	CHINTFLAG10	7:0							EVD	OVR
0x77	CHSTATUSn10	7:0							BUSYCH	RDYUSR

© 2025 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x78	CHANNEL11	7:0		EVGEN[6:0]						
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x7C	CHINTENCLR11	7:0							EVD	OVR
0x7D	CHINTENSET11	7:0							EVD	OVR
0x7E	CHINTFLAG11	7:0							EVD	OVR
0x7F	CHSTATUSn11	7:0							BUSYCH	RDYUSR
0x80	CHANNEL12	7:0		EVGEN[6:0]						
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x84 ... 0x87	Reserved									
0x88	CHANNEL13	7:0		EVGEN[6:0]						
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x8C ... 0x8F	Reserved									
0x90	CHANNEL14	7:0		EVGEN[6:0]						
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x94 ... 0x97	Reserved									
0x98	CHANNEL15	7:0		EVGEN[6:0]						
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x9C ... 0x9F	Reserved									
0xA0	CHANNEL16	7:0		EVGEN[6:0]						
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0xA4 ... 0xA7	Reserved									
0xA8	CHANNEL17	7:0		EVGEN[6:0]						
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0xAC ... 0xAF	Reserved									
0xB0	CHANNEL18	7:0		EVGEN[6:0]						
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0xB4 ... 0xB7	Reserved									
0xB8	CHANNEL19	7:0		EVGEN[6:0]						
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								

© 2025 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xBC ... 0xBF	Reserved									
0xC0	CHANNEL20	7:0					EVGEN[6:0]			
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0xC4 ... 0xC7	Reserved									
0xC8	CHANNEL21	7:0					EVGEN[6:0]			
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0xCC ... 0xCF	Reserved									
0xD0	CHANNEL22	7:0					EVGEN[6:0]			
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0xD4 ... 0xD7	Reserved									
0xD8	CHANNEL23	7:0					EVGEN[6:0]			
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0xDC ... 0xDF	Reserved									
0xE0	CHANNEL24	7:0					EVGEN[6:0]			
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0xE4 ... 0xE7	Reserved									
0xE8	CHANNEL25	7:0					EVGEN[6:0]			
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0xEC ... 0xEF	Reserved									
0xF0	CHANNEL26	7:0					EVGEN[6:0]			
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0xF4 ... 0xF7	Reserved									
0xF8	CHANNEL27	7:0					EVGEN[6:0]			
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0xFC ... 0xFF	Reserved									

© 2025 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0100	CHANNEL28	7:0					EVGEN[6:0]			
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x0104 ... 0x0107	Reserved									
0x0108	CHANNEL29	7:0					EVGEN[6:0]			
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x010C ... 0x010F	Reserved									
0x0110	CHANNEL30	7:0					EVGEN[6:0]			
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x0114 ... 0x0117	Reserved									
0x0118	CHANNEL31	7:0					EVGEN[6:0]			
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		23:16								
		31:24								
0x011C ... 0x011F	Reserved									
0x0120	USER0	7:0			CHANNEL[5:0]					
...										
0x015A	USER58	7:0			CHANNEL[5:0]					

Related Links

Product Memory Mapping Overview

30.7. Register Description

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

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

For more details, see *Register Access Protection* and *Peripheral Access Controller (PAC)* from Related Links.

Related Links

Register Access Protection

Peripheral Access Controller (PAC)

© 2025 Microchip Technology Inc. and its subsidiaries

30.7.1. Control A

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

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

Bit 0 – SWRST Software Reset
Writing '0' to this bit has no effect.
Writing '1' to this bit resets all registers in the EVSYS to their initial state.
Note: Before applying a Software Reset, disabling the event generators is recommended.

© 2025 Microchip Technology Inc. and its subsidiaries

30.7.2. Software Event

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

Bit	31	30	29	28	27	26	25	24
	CHANNEL31	CHANNEL30	CHANNEL29	CHANNEL28	CHANNEL27	CHANNEL26	CHANNEL25	CHANNEL24
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CHANNEL23	CHANNEL22	CHANNEL21	CHANNEL20	CHANNEL19	CHANNEL18	CHANNEL17	CHANNEL16
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CHANNEL15	CHANNEL14	CHANNEL13	CHANNEL12	CHANNEL11	CHANNEL10	CHANNEL9	CHANNEL8
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CHANNEL7	CHANNEL6	CHANNEL5	CHANNEL4	CHANNEL3	CHANNEL2	CHANNEL1	CHANNEL0
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – CHANNELx Channel x Software Selection [x=0..31]
Writing a '0' to this bit has no effect.
Writing a '1' to this bit triggers a software event for channel x.
These bits always return '0' when read.

© 2025 Microchip Technology Inc. and its subsidiaries

30.7.3. Priority Control

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

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

Bit 7 – RREN Round-Robin Scheduling Enable
For details on scheduling schemes, see *Interrupt Status and Interrupts Arbitration* from Related Links.

Value	Description
0	Static scheduling scheme for channels with level priority
1	Round-robin scheduling scheme for channels with level priority

Bits 4:0 – PRI[4:0] Channel Priority Number
When round-robin arbitration is enabled (PRICTRL.RREN=1) for the priority level, this register holds the channel number of the last EVSYS channel being granted access as the active channel with priority level. The value of this bit group is updated each time the INTPEND or any of CHINTFLAG registers are written.
When static arbitration is enabled (PRICTRL.RREN=0) for the priority level and the value of this bit group is nonzero, it does not affect the static priority scheme.
This bit group is not reset when round-robin scheduling gets disabled (PRICTRL.RREN written to zero).

Related Links
Interrupt Status and Interrupts Arbitration

© 2025 Microchip Technology Inc. and its subsidiaries

30.7.4. Channel Pending Interrupt

Name: INTPEND
Offset: 0x10
Reset: 0x4000

An interrupt that handles several channels must consult the INTPEND register to find out which channel number has priority (ignoring/filtering each channel that has its own interrupt line). An interrupt dedicated to only one channel must not use the INTPEND register.

Bit	15	14	13	12	11	10	9	8
	BUSY	READY					EVD	OVR
Access	R	R					RW	RW
Reset	0	1					0	0

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

Bit 15 – BUSY Busy
This bit is read as '1' when the event on a channel selected by the Channel ID field (ID) was not handled by all the event users connected to this channel.

Bit 14 – READY Ready
This bit is read as '1' when all event users connected to the channel selected by the Channel ID field (ID) are ready to handle incoming events on this channel.

Bit 9 – EVD Channel Event Detected
This flag is set on the next CLK_EVSYS_APB cycle when an event is being propagated through the channel, and an interrupt request is generated if CHINTENCLR/SET.EVD is '1'.
When the event channel path is asynchronous, the EVD bit will not be set.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears it. It also clears the corresponding flag in the Channel n Interrupt Flag Status and Clear register (CHINTFLAGn) of this peripheral, where n is determined by the Channel ID bit field (ID) in this register.

Bit 8 – OVR Channel Overrun
This flag is set on the next CLK_EVSYS cycle after an overrun channel condition occurs, and an interrupt request is generated if CHINTENCLR/SET.OVRx is '1'.
There are two possible overrun channel conditions:

- One or more of the event users on the channel selected by the Channel ID field (ID) are not ready when a new event occurs
- An event happens when all the event users have not yet handled the previous event on the channel selected by the Channel ID field (ID)

When the event channel path is asynchronous, the OVR interrupt flag will not be set.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears it. It also clears the corresponding flag in the Channel n Interrupt Flag Status and Clear register (CHINTFLAGn) of this peripheral, where n is determined by the Channel ID bit field (ID) in this register.

Bits 3:0 – ID[3:0] Channel ID
These bits store the channel number of the highest priority.

© 2025 Microchip Technology Inc. and its subsidiaries

When the bits are written, indirect access to the corresponding Channel Interrupt Flag register is enabled.

© 2025 Microchip Technology Inc. and its subsidiaries

30.7.5. Interrupt Status

Name: INTSTATUS
Offset: 0x14
Reset: 0x00000000

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
					CHINT11	CHINT10	CHINT9	CHINT8
Access					R	R	R	R
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CHINT7	CHINT6	CHINT5	CHINT4	CHINT3	CHINT2	CHINT1	CHINT0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 – CHINTx Channel x Pending Interrupt
This bit is set when Channel x has a pending interrupt.
This bit is cleared when the corresponding Channel x interrupts are disabled, or the source interrupt sources are cleared.

© 2025 Microchip Technology Inc. and its subsidiaries

30.7.6. Busy Channels

Name: BUSYCH
Offset: 0x18
Reset: 0x00000000

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
					BUSYCH11	BUSYCH10	BUSYCH9	BUSYCH8
Access					R	R	R	R
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BUSYCH7	BUSYCH6	BUSYCH5	BUSYCH4	BUSYCH3	BUSYCH2	BUSYCH1	BUSYCH0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 – BUSYCH Busy Channel x

This bit is set if an event occurs on channel x that was not handled by all event users connected to channel x.
This bit is cleared when channel x is idle.
When the event channel x path is asynchronous, this bit is always read as '0'.

© 2025 Microchip Technology Inc. and its subsidiaries

30.7.7. Ready Users

Name: READYUSR
Offset: 0x1C
Reset: 0x00000FFF

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
					READYUSR11	READYUSR10	READYUSR9	READYUSR8
Access					R	R	R	R
Reset					1	1	1	1

Bit	7	6	5	4	3	2	1	0
	READYUSR7	READYUSR6	READYUSR5	READYUSR4	READYUSR3	READYUSR2	READYUSR1	READYUSR0
Access	R	R	R	R	R	R	R	R
Reset	1	1	1	1	1	1	1	1

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 – READYUSRn Ready User for Channel n
This bit is set when all event users connected to channel n are ready to handle incoming events on channel n.
This bit is cleared when at least one of the event users connected to the channel is not ready.
When the event channel n path is asynchronous, this bit is always read zero.

© 2025 Microchip Technology Inc. and its subsidiaries

30.7.8. Channel n Control

Name: CHANNEL
Offset: 0x20 + n*0x08 [n=0..31]
Reset: 0x00008000
Property: PAC Write-Protection

This register allows the user to configure channel n. To write to this register, do a single, 32-bit write of all the configuration data.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
Access	RW	RW			RW	RW	RW	RW
Reset	1	0			0	0	0	0

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

Bit 15 – ONDEMAND Generic Clock On Demand

Value	Description
0	Generic clock for a channel is always on, if the channel is configured and generic clock source is enabled.
1	Generic clock is requested on demand while an event is handled

Bit 14 – RUNSTDBY Run in Standby
This bit is used to define the behavior during Idle and Standby Sleep mode.

Value	Description
0	The channel is disabled in Idle and Standby Sleep mode.
1	The channel is not stopped in Idle and Standby Sleep mode and depends on the CHANNEL.ONDEMAND bit.

Bits 11:10 – EDGSEL[1:0] Edge Detection Selection
These bits set the type of edge detection to be used on the channel.
These bits must be written to zero when using the asynchronous path.

Value	Name	Description
0x0	NO_EVT_OUTPUT	No event output when using the resynchronized path
0x1	RISING_EDGE	Event detection only on the rising edge of the signal from the event generator
0x2	FALLING_EDGE	Event detection only on the falling edge of the signal from the event generator
0x3	BOTH_EDGES	Event detection on rising and falling edges of the signal from the event generator

Bits 9:8 – PATH[1:0] Path Selection
These bits are used to choose which path is used by the selected channel.
Note: The path choice can be limited by the channel source; see *USERm* from Related Links.

© 2025 Microchip Technology Inc. and its subsidiaries

Important: Only EVSYS channel 0 to 11 can be configured as synchronous or resynchronized.

Value	Name	Description
0x1	RESYNCHRONIZED	Resynchronized path
0x2	ASYNCHRONOUS	Asynchronous path
Other	—	Reserved

Bits 6:0 – EVGEN[6:0] Event Generator Selection
These bits are used to choose the event generator to connect to the selected channel.

Table 30-2. Event Generator Selection

Value	Name	Description
0x00	None	No event generator selected
0x01-0x08	RTC_PERx	RTC period x=0..7
0x09-0x0C	RTC_CMPx	RTC comparison x=0..3
0x0D	RTC_TAMPER	RTC tamper detection
0x0E	RTC_OVF	RTC overflow
0x0F-0x12	EIC_EXTINTx	EIC external interrupt x=0..3
0x13-0x16	DMAC_CHx	DMA channel x=0..3
0x17	PAC_ACCERR	PAC Acc. error
0x18	TCC0_OVF	TCC0 overflow
0x19	TCC0_TRG	TCC0 trigger event
0x1A	TCC0_CNT	TCC0 counter
0x1B-0x20	TCC0_MCx	TCC0 match/compare x=0..5
0x21	TCC1_OVF	TCC1 overflow
0x22	TCC1_TRG	TCC1 trigger event
0x23	TCC1_CNT	TCC1 counter
0x24-0x29	TCC1_MCx	TCC1 match/compare x=0..5
0x2A	TCC2_OVF	TCC2 overflow
0x2B	TCC2_TRG	TCC2 trigger event
0x2C	TCC2_CNT	TCC2 counter
0x2D-0x2E	TCC2_MCx	TCC2 match/compare x=0..1
0x2F	TC0_OVF	TC0 overflow
0x30-0x31	TC0_MCx	TC0 match/compare x=0..1
0x32	TC1_OVF	TC1 overflow
0x33-0x34	TC1_MCx	TC1 match/compare x=0..1
0x35	TC2_OVF	TC2 overflow
0x36-0x37	TC2_MCx	TC2 match/compare x=0..1
0x38	TC3_OVF	TC3 overflow
0x39-0x3A	TC3_MCx	TC3 match/compare x=0..1
0x3B	TC4_OVF	TC4 overflow
0x3C-0x3D	TC4_MCx	TC4 match/compare x=0..1
0x3E	TC5_OVF	TC5 overflow
0x3F-0x40	TC5_MCx	TC5 match/compare x=0..1
0x41	TC6_OVF	TC6 overflow
0x42-0x43	TC6_MCx	TC6 match/compare x=0..1
0x44	TC7_OVF	TC7 overflow
0x45-0x46	TC7_MCx	TC7 match/compare x=0..1
0x47	TC8_OVF	TC8 overflow
0x48-0x49	TC8_MCx	TC8 match/compare x=0..1
0x4A	TC9_OVF	TC9 overflow
0x4B-0x4C	TC9_MCx	TC9 match/compare x=0..1

© 2025 Microchip Technology Inc. and its subsidiaries

Table 30-2. Event Generator Selection (continued)

Value	Name	Description
0x4D	ADC_UPBS_ADC_EVENT_16	No ADC early interrupts present on this device
0x4E	ADC_UPBS_ADC_EVENT_99	No ADC early interrupts present on this device
0x4F	ADC_UPBS_ADC_EVENT_100	No ADC early interrupts present on this device
0x50-0x51	AC_COMP_X	AC comparator, x=0..1
0x52	AC_WIN_0	AC0 window
0x53	TRNG_READY	Spare No TRNG READY event needed as trigger
0x54-0x55	CCL_LUTOUT_X	—
0x56	zb_tx_ts_active	ZB Transmit Packet Active time
0x57	zb_rx_ts_active	ZB Receive Packet Active time
0x58	GMAC.TSU_CMP	GMAC Timestamp CMP
0x59	QEI Interrupt	QEI Events

Related Links

USERm

Event User m

© 2025 Microchip Technology Inc. and its subsidiaries

30.7.9. Channel n Interrupt Enable Clear

Name: CHINTENCLR
Offset: 0x24 + n*0x08 [n=0..11]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
							EVD	OVR
Access							RW	RW
Reset							0	0

Bit 1 – EVD Channel Event Detected Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Event Detected Channel Interrupt Enable bit, which disables the Event Detected Channel interrupt.

Value	Description
0	The Event Detected Channel interrupt is disabled.
1	The Event Detected Channel interrupt is enabled.

Bit 0 – OVR Channel Overrun Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Overrun Channel Interrupt Enable bit, which disables the Overrun Channel interrupt.

Value	Description
0	The Overrun Channel interrupt is disabled.
1	The Overrun Channel interrupt is enabled.

© 2025 Microchip Technology Inc. and its subsidiaries

30.7.10. Channel n Interrupt Enable Set

Name: CHINTENSET
Offset: 0x25 + n*0x08 [n=0..11]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
							EVD	OVR
Access							RW	RW
Reset							0	0

Bit 1 – EVD Channel Event Detected Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Event Detected Channel Interrupt Enable bit, which enables the Event Detected Channel interrupt.

Value	Description
0	The Event Detected Channel interrupt is disabled.
1	The Event Detected Channel interrupt is enabled.

Bit 0 – OVR Channel Overrun Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Overrun Channel Interrupt Enable bit, which enables the Overrun Channel interrupt.

Value	Description
0	The Overrun Channel interrupt is disabled.
1	The Overrun Channel interrupt is enabled.

© 2025 Microchip Technology Inc. and its subsidiaries

30.7.11. Channel n Interrupt Flag Status and Clear

Name: CHINTFLAG
Offset: 0x26 + n*0x08 [n=0..11]
Reset: 0x00

Bit	7	6	5	4	3	2	1	0
							EVD	OVR
Access							RW	RW
Reset							0	0

Bit 1 – EVD Channel Event Detected
This flag is set on the next CLK_EVSYS_APB cycle when an event is being propagated through the channel, and an interrupt request will be generated if CHINTENCLR/SET.EVD is '1'.
When the event channel path is asynchronous, the EVD interrupt flag will not be set.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Event Detected Channel interrupt flag.

Bit 0 – OVR Channel Overrun
This flag is set on the next CLK_EVSYS cycle after an overrun channel condition occurs, and an interrupt request will be generated if CHINTENCLR/SET.OVR is '1'.
There are two possible overrun channel conditions:

- One or more of the event users on the channel are not ready when a new event occurs.
- An event happens when the previous event on channel has not yet been handled by all event users.

When the event channel path is asynchronous, the OVR interrupt flag will not be set.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Overrun Channel interrupt flag.

© 2025 Microchip Technology Inc. and its subsidiaries

30.7.12. Channel n Status

Name: CHSTATUSn
Offset: 0x27 + n*0x08 [n=0..11]
Reset: 0x01

Bit	7	6	5	4	3	2	1	0
							BUSYCH	RDYUSR
Access							R	R
Reset							0	0

Bit 1 – BUSYCH Busy Channel
This bit is cleared when the channel is idle.
This bit is set if an event on channel was not handled by all event users connected to the channel.
When the event channel path is asynchronous, this bit is always read as '0'.

Bit 0 – RDYUSR Ready User
This bit is cleared when at least one of the event users connected to the channel is not ready.
This bit is set when all event users connected to the channel are ready to handle incoming events on the channel.
When the event channel path is asynchronous, this bit is always read as '0'.

© 2025 Microchip Technology Inc. and its subsidiaries

30.7.13. Event User m

Name: USERm
Offset: 0x0120 + m*0x01 [m=0..58]
Reset: 0x0
Property: PAC Write-Protection

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

Bits 5:0 – CHANNEL[5:0] Channel Event Selection
These bits select channel n to connect to the event user m.
Note: A value x of this bit field selects channel n = x-1.

Table 30-3. User Multiplexer Number m

USER*m*	User Multiplexer	Description	Path Type(1)
m = 0	RTC_TAMPER	RTC Tamper	A, S, R
m = 1..8	DMAC_CH0..7	Channel 0..7	S, R
m = 9	CM4_TRACE_START	CM4 trace start	A, S, R
m = 10	CM4_TRACE_STOP	CM4 trace stop	A, S, R
m = 11	CM4_TRACE_TRIG	CM4 trace trigger	A, S, R
m = 12..13	TCC0 EV0..1	TCC0 EVx	A, S, R
m = 14..19	TCC0 MC0..5	TCC0 MCx	A, S, R
m = 20..21	TCC1 EV0..1	TCC1 EVx	A, S, R
m = 22..27	TCC1 MC0..5	TCC1 MCx	A, S, R
m = 28..29	TCC2 EV0..1	TCC2 EVx	A, S, R
m = 30..31	TCC2 MC0..1	TCC2 MCx	A, S, R
m = 32	TC0 EVU	TC0 EVU	A, S, R
m = 33	TC1 EVU	TC1 EVU	A, S, R
m = 34	TC2 EVU	TC2 EVU	A, S, R
m = 35	TC3 EVU	TC3 EVU	A, S, R
m = 36	TC4 EVU	TC4 EVU	A, S, R
m = 37	TC5 EVU	TC5 EVU	A, S, R
m = 38	TC6 EVU	TC6 EVU	A, S, R
m = 39	TC7 EVU	TC7 EVU	A, S, R
m=40	TC8_EVU	TC8_EVU	A, S, R
m=41	TC9_EVU	TC9_EVU	A, S, R
m=42	ADC_TRIGGER_5	ADC_TRIGGER_5	A
m=43	ADC_TRIGGER_6	ADC_TRIGGER_6	A
m=44	ADC_TRIGGER_7	ADC_TRIGGER_7	A
m=45	ADC_TRIGGER_8	ADC_TRIGGER_8	A
m=46	ADC_TRIGGER_9	ADC_TRIGGER_9	A
m=47	ADC_TRIGGER_10	ADC_TRIGGER_10	A
m=48	ADC_TRIGGER_11	ADC_TRIGGER_11	A
m=49	ADC_TRIGGER_12	ADC_TRIGGER_12	A
m=50	ADC_TRIGGER_13	ADC_TRIGGER_13	A
m=51	ADC_TRIGGER_14	ADC_TRIGGER_14	A
m=52	ADC_TRIGGER_15	ADC_TRIGGER_15	A
m=53	ADC_TRIGGER_16	ADC_TRIGGER_16	A
m=54	AC_SOC_0	AC_SOC_0	A, S, R
m=55	AC_SOC_1	AC_SOC_1	A, S, R
m=56	CCL_LUTIN_0	CCL_LUTIN_0	A, S, R

© 2025 Microchip Technology Inc. and its subsidiaries

Table 30-3. User Multiplexer Number m (continued)

USER*m*	User Multiplexer	Description	Path Type[1]
m=57	CCL_LUTIN_1	CCL_LUTIN_1	A, S, R
m=58	CVD_TRIGGER	CVD_TRIGGER	A

1) A = Asynchronous path, S = Synchronous path, R = Resynchronized path

© 2025 Microchip Technology Inc. and its subsidiaries

31. Serial Communication Interface (SERCOM)

31.1. Overview

The device supports up to seven SERCOM modules. Six SERCOM's (SERCOM0/1/2/3/4/5) can be configured to support a number of modes: I^2C, SPI and USART. One of the SERCOM (SERCOM6) has only I^2C functionality. When an instance of SERCOM is configured and enabled, all of the resources of that SERCOM instance is dedicated to the selected mode.

The SERCOM serial engine consists of a transmitter and receiver, baud-rate generator and address-matching functionality and mode-specific transmitter and receiver logic. It can use the internal generic clock or an external clock.

Table 31-1. SERCOM Instances

SERCOM	FIFO Depth	SPI	USART	I2C/SMBus	1.8V I2C/SMBus	Full Speed
SERCOM0(through PPS)	4	Yes	Yes	—	—	—
SERCOM0(Fixed pins)		Yes	Yes	Yes	—	Yes
SERCOM1(through PPS)	2	Yes	Yes	—	—	—
SERCOM1(Fixed pins)		Yes	Yes	Yes	—	Yes
SERCOM2(through PPS)	2	Yes	Yes	—	—	—
SERCOM2(Fixed pins)		Yes	Yes	Yes	Yes(1)	Yes
SERCOM3(through PPS)	2	Yes	Yes	—	—	—
SERCOM4(through PPS)	2	Yes	Yes	—	—	—
SERCOM5(through PPS)	2	Yes	Yes	—	—	—
SERCOM6(Fixed pins)	2	—	—	Yes	—	—

Note:
1. When used with external (off-chip) 1.8V pull-ups.

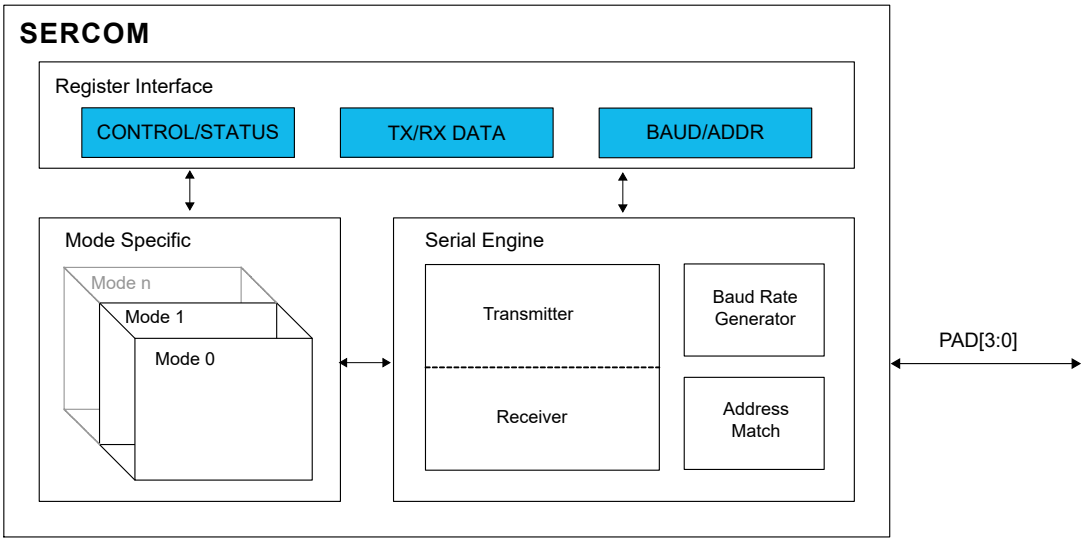
31.2. Features

- Interface for Configuring into One of the Following:
  - I^2C two-wire serial interface
  - SPI
  - USART
  - SMBus™ compatible
- Single Transmit Buffer and Double Receive Buffer
  Note: SERCOM0 has a FIFO depth of 4.
- Baud-Rate Generator
- Address Match/Mask Logic
- Operational in Idle and Standby Sleep Mode with an External Clock Source
- Can be Used with DMA

© 2025 Microchip Technology Inc. and its subsidiaries

31.3. Block Diagram

Figure 31-1. SERCOM Block Diagram



31.4. Signal Description

See the respective SERCOM mode chapters for details.

31.5. Product Dependencies

The following sections describe how the other parts of the system must be configured correctly to use this peripheral.

31.5.1. I/O Lines

Using the SERCOM I/O lines requires the I/O pins to be configured using the System Configuration registers or PPS.

The SERCOM has four internal pads, PAD[3:0], and the signals from I^2C, SPI and USART are routed through these SERCOM pads through a multiplexer. The configuration of the multiplexer is available from the different SERCOM modes.

See *I/O Ports and Peripheral Pin Select (PPS)* from Related Links.

Related Links

I/O Ports and Peripheral Pin Select (PPS)

31.5.2. Power Management

The SERCOM can operate in any Sleep mode (Idle, Standby Sleep) provided the selected clock source is running. SERCOM interrupts can be configured to wake the device from sleep modes.

31.5.3. Clocks

The SERCOM uses two generic clocks:

- GCLK_SERCOMx_CORE – The core clock (GCLK_SERCOMx_CORE) is required to clock the SERCOM while working as a host.
- GCLK_SERCOMx_SLOW – The slow clock (GCLK_SERCOMx_SLOW) is only required for certain functions.

These clocks must be configured and enabled in the Generic Clock Controller (GCLK) before using the SERCOM. See *Clock and Reset Unit (CRU)* from Related Links.

© 2025 Microchip Technology Inc. and its subsidiaries

The generic clocks are asynchronous to the bus clock (PBx_CLK). Therefore, writing to certain registers requireS synchronization between the clock domains.

Related Links

Clock and Reset Unit (CRU)

31.5.4. DMA

The DMA request lines are connected to the DMA Controller (DMAC). The DMAC must be configured before the SERCOM DMA requests are used. See *Direct Memory Access Controller (DMAC)* from Related Links.

Related Links

Direct Memory Access Controller (DMAC)

31.5.5. Interrupts

The interrupt request line is connected to the Interrupt Controller (NVIC). The NVIC must be configured before the SERCOM interrupts are used.

31.5.6. Events

Not applicable.

31.5.7. Debug Operation

When the CPU is halted in Debug mode, this peripheral continues normal operation. If the peripheral is configured to require periodical service by the CPU through interrupts or similar, improper operation or data loss may result during debugging. This peripheral can be forced to halt operation during debugging; see *DBGCTRL* register from Related Links.

Related Links

DBGCTRL

Debug Control

31.5.8. Register Access Protection

Registers with write access can be write protected optionally by the PAC.

Optional write protection by the 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.

31.5.9. Analog Connections

Not applicable.

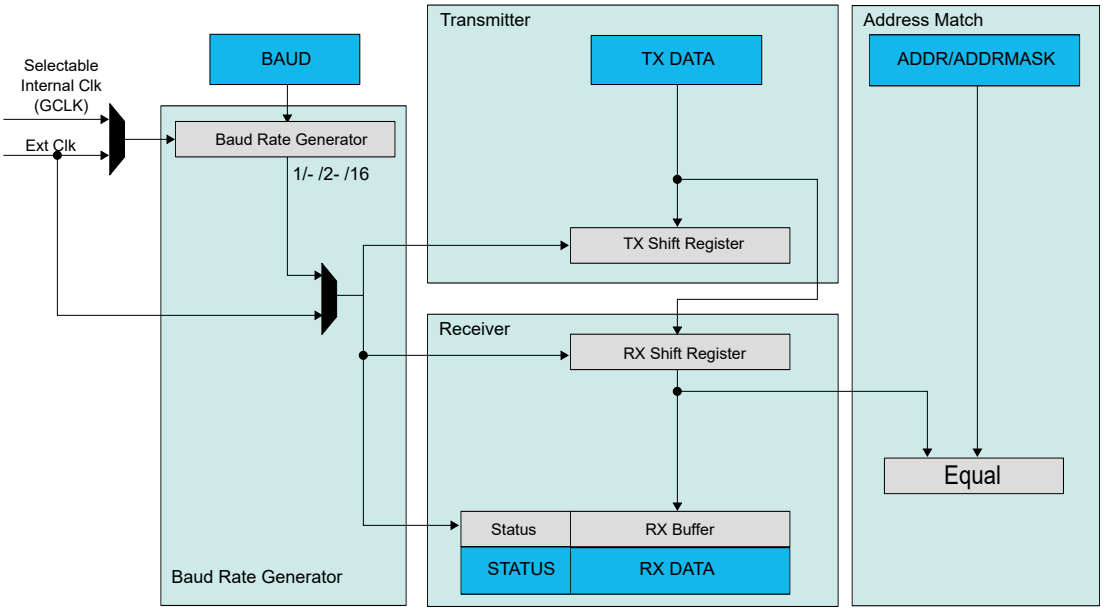
31.6. Functional Description

31.6.1. Principle of Operation

The basic structure of the SERCOM serial engine is shown in *SERCOM Serial Engine*. Labels in capital letters are synchronous to the system clock and accessible by the CPU; labels in lowercase letters can be configured to run on the GCLK_SERCOMx_CORE clock or an external clock.

© 2025 Microchip Technology Inc. and its subsidiaries

Figure 31-2. SERCOM Serial Engine



The transmitter consists of a single write buffer and a Shift register.

The receiver consists of a one-level (I^2C), or two-level (USART, SPI) receive buffer and a Shift register.

The baud-rate generator is capable of running on the GCLK_SERCOMx_CORE clock or an external clock.

Address matching logic is included for SPI and I^2C operation.

31.6.2. Basic Operation

31.6.2.1. Initialization

The SERCOM must be configured to the desired mode by writing the Operating Mode bits in the Control A register (CTRLA.MODE) as shown in the table below.

Table 31-2. SERCOM Modes

CTRLA.MODE	Description
0x0	USART with external clock
0x1	USART with internal clock
0x2	SPI in client operation
0x3	SPI in host operation
0x4	I^2C client operation
0x5	I^2C host operation
0x6-0x7	Reserved

For further initialization information, see the respective SERCOM mode chapters:

31.6.2.2. Enabling, Disabling and Resetting

This peripheral is enabled by writing '1' to the Enable bit in the Control A register (CTRLA.ENABLE) and disabled by writing '0' to it.

Writing '1' to the Software Reset bit in the Control A register (CTRLA.SWRST) resets all registers of this peripheral to their initial states, except the DBGCTRL register, and the peripheral is disabled.

See the *CTRLA* register from Related Links.

© 2025 Microchip Technology Inc. and its subsidiaries

Related Links

CTRLA

Control A

31.6.2.3. Clock Generation – Baud-Rate Generator

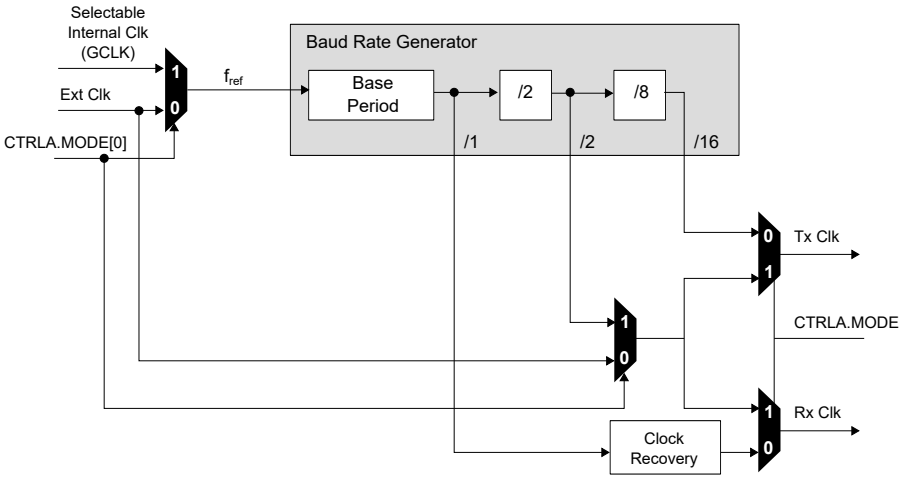
The baud-rate generator, as illustrated in the following figure, generates internal clocks for asynchronous and synchronous communication. The output frequency (f_{BAUD}) is determined by the Baud register (BAUD) setting and the baud reference frequency (f_{ref}). The baud reference clock is the serial engine clock, and it can be internal or external.

For asynchronous communication, the /16 (divide-by-16) output is used when transmitting, whereas, the /1 (divide-by-1) output is used while receiving.

For synchronous communication, the /2 (divide-by-2) output is used.

This functionality is automatically configured, depending on the selected operating mode.

Figure 31-3. Baud Rate Generator



The following table contains equations for the baud rate (in bits per second) and the BAUD register value for each operating mode.

For asynchronous operation, there are two modes:

- Arithmetic mode – The BAUD register value is 16 bits (0 to 65,535)
- Fractional mode – The BAUD register value is 13 bits, while the fractional adjustment is 3 bits. In this mode, the BAUD setting must be greater than or equal to 1.

For synchronous operation, the BAUD register value is 8 bits (0 to 255).

Table 31-3. Baud Rate Equations

Operating Mode	Condition	Baud Rate (Bits Per Second)	BAUD Register Value Calculation
Asynchronous Arithmetic	$f_{BAUD} \leq \frac{f_{ref}}{16}$	$f_{BAUD} = \frac{f_{ref}}{16}\left(1 - \frac{BAUD}{65536}\right)$	$BAUD = 65536 \cdot \left(1 - S \cdot \frac{f_{BAUD}}{f_{ref}}\right)$
Asynchronous Fractional	$f_{BAUD} \leq \frac{f_{ref}}{S}$	$f_{BAUD} = \frac{f_{ref}}{S \cdot \left(BAUD + \frac{FP}{8}\right)}$	$BAUD = \frac{f_{ref}}{S \cdot f_{BAUD}} - \frac{FP}{8}$
Synchronous	$f_{BAUD} \leq \frac{f_{ref}}{2}$	$f_{BAUD} = \frac{f_{ref}}{2 \cdot (BAUD + 1)}$	$BAUD = \frac{f_{ref}}{2 \cdot f_{BAUD}} - 1$

S - Number of samples per bit, which can be 16, 8 or 3.

© 2025 Microchip Technology Inc. and its subsidiaries

The Asynchronous Fractional option is used for auto-baud detection.

The baud rate error is represented by the following formula:

$$\text{Error} = 1 - \left(\frac{\text{ExpectedBaudRate}}{\text{ActualBaudRate}}\right)$$

31.6.3. Additional Features

31.6.3.1. Address Match and Mask

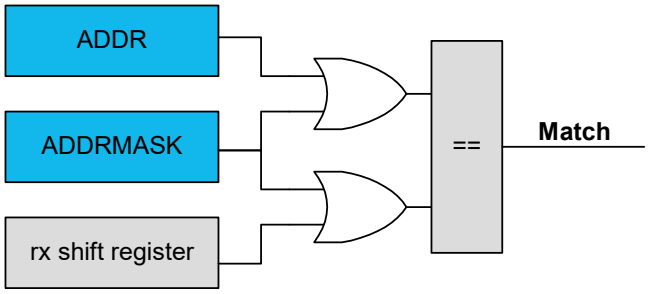
The SERCOM address match and mask feature is capable of matching either one address, two unique addresses, or a range of addresses with a mask, based on the mode selected. The match uses seven or eight bits, depending on the mode.

31.6.3.1.1. Address Mask

An address written to the Address bits in the Address register (ADDR.ADDR), and a mask written to the Address Mask bits in the Address register (ADDR.ADDRMASK) will yield an address match. All bits that are masked are not included in the match.

Note: Writing the ADDR.ADDRMASK to "all zeros" will match a single unique address, while writing ADDR.ADDRMASK to "all ones" will result in all addresses being accepted.

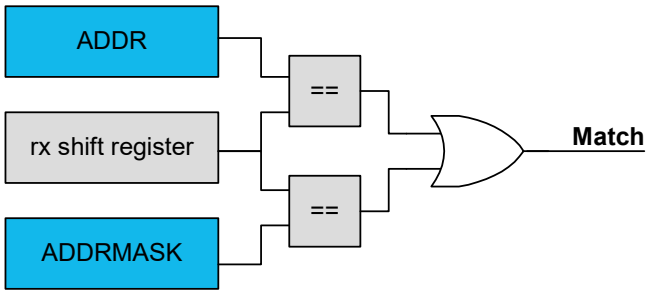
Figure 31-4. Address With Mask



31.6.3.1.2. Two Unique Addresses

The two addresses written to ADDR and ADDRMASK will cause a match.

Figure 31-5. Two Unique Addresses



31.6.3.1.3. Address Range

The range of addresses between and including ADDR.ADDR and ADDR.ADDRMASK will cause a match. ADDR.ADDR and ADDR.ADDRMASK can be set to any two addresses, with ADDR.ADDR acting as the upper limit and ADDR.ADDRMASK acting as the lower limit.

Figure 31-6. Address Range



© 2025 Microchip Technology Inc. and its subsidiaries

31.6.4. DMA Operation

The available DMA interrupts depend on the operation mode of the SERCOM peripheral. Refer to the Functional Description sections of the respective SERCOM mode.

31.6.5. Interrupts

Interrupt sources are mode specific. See the respective SERCOM mode chapters for details.

Each interrupt source has its own Interrupt flag.

The Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) will be set when the Interrupt condition is met.

Each interrupt can be individually enabled by writing '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR).

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

The value of INTFLAG indicates which Interrupt condition occurred. The user must read the INTFLAG register to determine which Interrupt condition is present.

Note: Interrupts must be globally enabled for interrupt requests to be generated.

31.6.6. Events

Not applicable.

31.6.7. Sleep Mode Operation

The peripheral can operate in any sleep mode (Idle, Standby Sleep) where the selected serial clock is running. This clock can be external or generated by the internal baud-rate generator.

The SERCOM interrupts can be used to wake up the device from sleep modes. Refer to the different SERCOM mode chapters for details.

31.6.8. Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

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

Required read synchronization is denoted by the Read-Synchronized property in the register description.

© 2025 Microchip Technology Inc. and its subsidiaries

32. SERCOM Synchronous and Asynchronous Receiver and Transmitter (SERCOM USART)

32.1. Overview

The Universal Synchronous and Asynchronous Receiver and Transmitter (USART) is one of the available modes in the Serial Communication Interface (SERCOM).

The USART uses the SERCOM transmitter and receiver (see Block Diagram). Labels in uppercase letters are synchronous to PBx_CLK and accessible for the CPU. Labels in lowercase letters can be programmed to run on the internal generic clock or an external clock.

The transmitter consists of a single write buffer, a Shift register and control logic for different frame formats. The write buffer supports data transmission without any delay between frames. The receiver consists of a two-level receive buffer and a Shift register. Status information of the received data is available for error checking. Data and clock recovery units ensure robust synchronization and noise filtering during asynchronous data reception.

32.2. USART Features

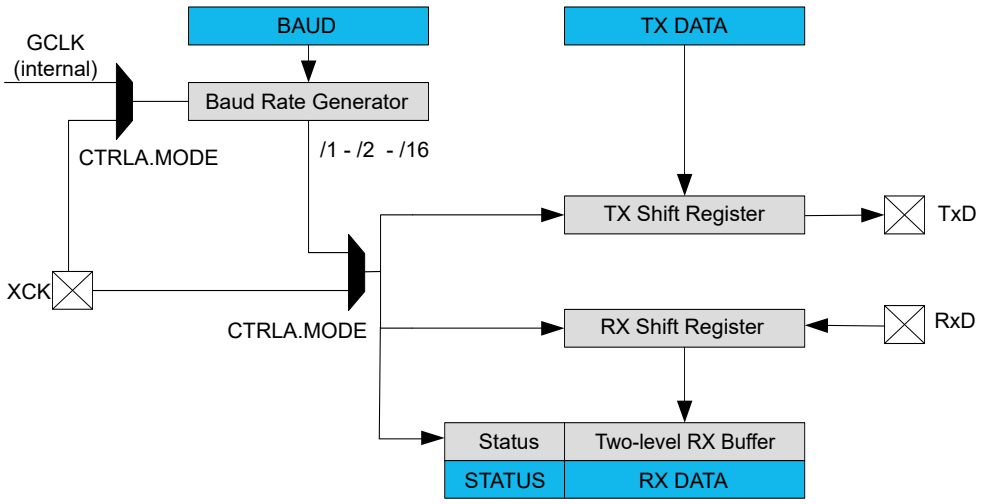
- Full-Duplex Operation
- Asynchronous (with Clock Reconstruction) or Synchronous Operation
- Internal or External Clock Source for Asynchronous and Synchronous Operation
- Baud-Rate Generator
- Supports Serial Frames with 5, 6, 7, 8 or 9 Data Bits and 1 or 2 Stop Bits
- Odd or Even Parity Generation and Parity Check
- Selectable LSB- or MSB- First Data Transfer
- Buffer Overflow and Frame Error Detection
- Noise Filtering, Including False Start-Bit Detection and Digital Low-Pass Filter
- Collision Detection
- Can Operate in All Sleep Modes
- Operation at Speeds Up to Half the System Clock for Internally-Generated Clocks
- Operation at Speeds Up to the System Clock for Externally-Generated Clocks
- RTS and CTS Flow Control
- IrDA Modulation and Demodulation Up to 115.2 kbps
- LIN Host Support
- LIN Client Support
  - Auto-baud and break character detection
- ISO 7816 T = 0 or T = 1 Protocols for Smart Card Interfacing
  • RS485 Support
- Start-of-Frame Detection
- Two-Level Receive Buffer
- Can Work with DMA
- 32-Bit Extension for Better System Bus Utilization

Note: SERCOM6 does not have USART functionality.

© 2025 Microchip Technology Inc. and its subsidiaries

32.3. Block Diagram

Figure 32-1. USART Block Diagram



32.4. Signal Description

Table 32-1. SERCOM USART Signals

Signal Name	Type	Description
PAD[3:0]	Digital I/O	General SERCOM pins

One signal can be mapped to one of several pins.

Note: For details on pin mapping, see *I/O Ports and Peripheral Pin Select (PPS)* from Related Links.

Related Links
I/O Ports and Peripheral Pin Select (PPS)

32.5. Product Dependencies

The following sections describe how the other parts of the system must be configured to correctly use this peripheral.

32.5.1. I/O Lines

Using the USART's I/O lines requires the I/O pins to be configured using the System Configuration registers (See *System Configuration and Register Locking (CFG)* from Related Links.) (SCOM_HSEN[2:0] of CFGCON1/DEVCFG1 register). If SERCOM pins are selected through PPS, the PPS registers have to be configured. See *I/O Ports and Peripheral Pin Select (PPS)* from Related Links.

If SCOMx_HSEN = 1, SERCOM uses dedicated pins.

If SCOMx_HSEN = 0, SERCOM uses the PPS path and I/O pins are multiplexed to pin groups defined in the PPS section.
Note: Only SERCOM0/1/2 can be configured using either dedicated pins or PPS path.

When the SERCOM is used in the USART mode, the SERCOM controls the direction and value of the I/O pins according to the following table. If the receiver or transmitter is disabled, these pins can be used for other purposes.

Table 32-2. USART Pin Configuration

Pin	Pin Configuration
TxD	Output

© 2025 Microchip Technology Inc. and its subsidiaries

Table 32-2. USART Pin Configuration (continued)

Pin	Pin Configuration
RxD	Input
XCK	Output or input

The configuration of the Transmit Data Pinout and Receive Data Pinout bit fields in the Control A register (CTRLA.TXPO and CTRLA.RXPO, respectively) defines the physical position of the USART signals in the above table.

Related Links

I/O Ports and Peripheral Pin Select (PPS)

System Configuration and Register Locking (CFG)

32.5.2. Power Management

This peripheral can continue to operate in any sleep mode (Idle, Standby Sleep) where its source clock is running. The interrupts can wake up the device from sleep modes.

32.5.3. Clocks

A generic clock (GCLK_SERCOMx_CORE) is required to clock the SERCOMx_CORE. This clock must be configured and enabled in the CRU registers before using it for SERCOMx_CORE. See *Clock and Reset Unit (CRU)* and *Overview* from Related Links.

This generic clock is asynchronous to the bus clock (PBx_CLK). Therefore, writing to certain registers requires synchronization to the clock domains.

Related Links

Clock and Reset Unit (CRU)

Overview

32.5.4. DMA

The DMA request lines are connected to the DMA Controller (DMAC). To use DMA requests with this peripheral, the DMAC must be configured first (see *Direct Memory Access Controller (DMAC)* from Related Links).

Related Links

Direct Memory Access Controller (DMAC)

32.5.5. Interrupts

The interrupt request line is connected to the Interrupt Controller. In order to use interrupt requests of this peripheral, the NVIC must be configured first. See *Nested Vector Interrupt Controller (NVIC)* from Related Links.

Related Links

Nested Vector Interrupt Controller (NVIC)

32.5.6. Events

Not applicable.

32.5.7. Debug Operation

When the CPU is halted in the Debug mode, this peripheral continues normal operation. If the peripheral is configured to require periodical service by the CPU through interrupts or similar, improper operation or data loss may result during debugging. This peripheral can be forced to halt operation during debugging. See *DBGCTRL* register from Related Links.

© 2025 Microchip Technology Inc. and its subsidiaries

Related Links

DBGCTRL

Debug Control

32.5.8. Register Access Protection

Registers with write access can be write-protected optionally by the PAC.

PAC write protection is not available for the following registers:

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

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

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

32.5.9. Analog Connections

Not applicable.

32.6. Functional Description

32.6.1. Principle of Operation

The USART uses the following lines for data transfer:

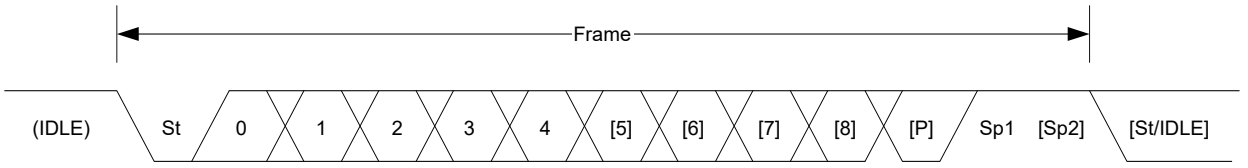
- RxD for receiving
- TxD for transmitting
- XCK for the transmission clock in synchronous operation

USART data transfer is frame based. A serial frame consists of:

- 1 start bit
- From 5 to 9 data bits (MSB or LSB first)
- No, even or odd parity bit
- 1 or 2 stop bits

A frame starts with the Start bit followed by one character of Data bits. If enabled, the parity bit is inserted after the Data bits and before the first Stop bit. After the stop bit(s) of a frame, either the next frame can follow immediately or the communication line can return to the Idle (high) state. The following figure illustrates the possible frame formats. Values inside brackets ([x]) denote optional bits.

Figure 32-2. Frame Formats



St	Start bit. Signal is always low.
n, [n]	Data bits. 0 to [4..8]
[P]	Parity bit. Either odd or even.
Sp, [Sp]	Stop bit. Signal is always high.

© 2025 Microchip Technology Inc. and its subsidiaries

IDLE No frame is transferred on the communication line. Signal is always high in this state.

32.6.2. Basic Operation

32.6.2.1. Initialization

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

- Control A register (CTRLA), except the Enable (ENABLE) and Software Reset (SWRST) bits
- Control B register (CTRLB), except the Receiver Enable (RXEN) and Transmitter Enable (TXEN) bits
- Baud register (BAUD)

Any writes to these registers when the USART is enabled or is being enabled (CTRL.ENABLE is one) are discarded. Writes to these registers while the peripheral is being disabled is completed after the disabling is complete.

When the USART is enabled or is being enabled (CTRLA.ENABLE=1), any writing attempt to these registers is discarded. If the peripheral is being disabled, writing to these registers is executed after disabling is completed. Enable protection is denoted by the Enable-Protection property in the register description.

Before the USART is enabled, it must be configured by these steps:

1. Select either the external (0x0) or internal clock (0x1) by writing the Operating Mode value in the CTRLA register (CTRLA.MODE).
2. Select either Asynchronous (0) or Synchronous (1) Communication mode by writing the Communication Mode bit in the CTRLA register (CTRLA.CMODE).
3. Select the pin for receive data by writing the Receive Data Pinout value in the CTRLA register (CTRLA.RXPO).
4. Select pads for the transmitter and external clock by writing the Transmit Data Pinout bit in the CTRLA register (CTRLA.TXPO).
5. Configure the Character Size field in the CTRLB register (CTRLB.CHSIZE) for the character size.
6. Set the Data Order bit in the CTRLA register (CTRLA.DORD) to determine MSB- or LSB-first data transmission.
7. To use Parity mode:
   a. Enable Parity mode by writing 0x1 to the Frame Format field in the CTRLA register (CTRLA.FORM).
   b. Configure the Parity Mode bit in the CTRLB register (CTRLB.PMODE) for even or odd parity.
8. Configure the number of stop bits in the Stop Bit Mode bit in the CTRLB register (CTRLB.SBMODE).
9. When using an internal clock, write the Baud register (BAUD) to generate the desired baud rate.
10. Enable the transmitter and receiver by writing '1' to the Receiver Enable and Transmitter Enable bits in the CTRLB register (CTRLB.RXEN and CTRLB.TXEN).

32.6.2.2. Enabling, Disabling and Resetting

This peripheral is enabled by writing '1' to the Enable bit in the Control A register (CTRLA.ENABLE) and disabled by writing '0' to it.

Writing '1' to the Software Reset bit in the Control A register (CTRLA.SWRST) resets all registers of this peripheral to their initial states, except the DBGCTRL register, and the peripheral is disabled.

See the *CTRLA* register from Related Links.

© 2025 Microchip Technology Inc. and its subsidiaries

32.6.2.3. Clock Generation and Selection

For both Synchronous and Asynchronous modes, the clock used for shifting and sampling data can be generated internally by the SERCOM baud-rate generator or supplied externally through the XCK line.

The Synchronous mode is selected by writing a '1' to the Communication Mode bit in the Control A register (CTRLA.CMODE), the Asynchronous mode is selected by writing '0' to CTRLA.CMODE.

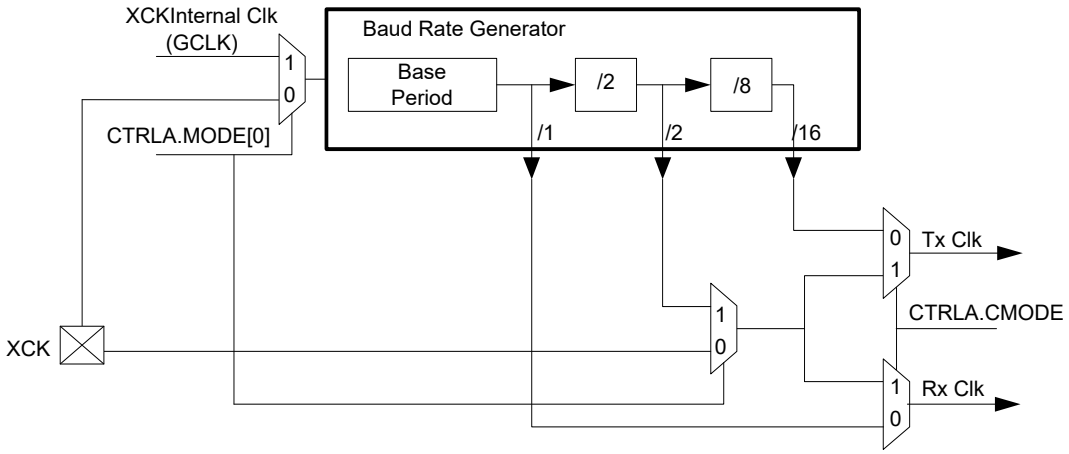
The internal clock source is selected by writing '1' to the Operation Mode bit field in the Control A register (CTRLA.MODE), the external clock source is selected by writing '0' to CTRLA.MODE.

The SERCOM baud-rate generator is configured as in the following figure.

In Asynchronous mode (CTRLA.CMODE=0), the 16-bit Baud register value is used.

In Synchronous mode (CTRLA.CMODE=1), the eight LSBs of the Baud register are used. For more details on configuring the baud rate (see *Clock Generation – Baud-Rate Generator* from Related Links).

Figure 32-3. Clock Generation



Related Links

Clock Generation – Baud-Rate Generator

32.6.2.3.1. Synchronous Clock Operation

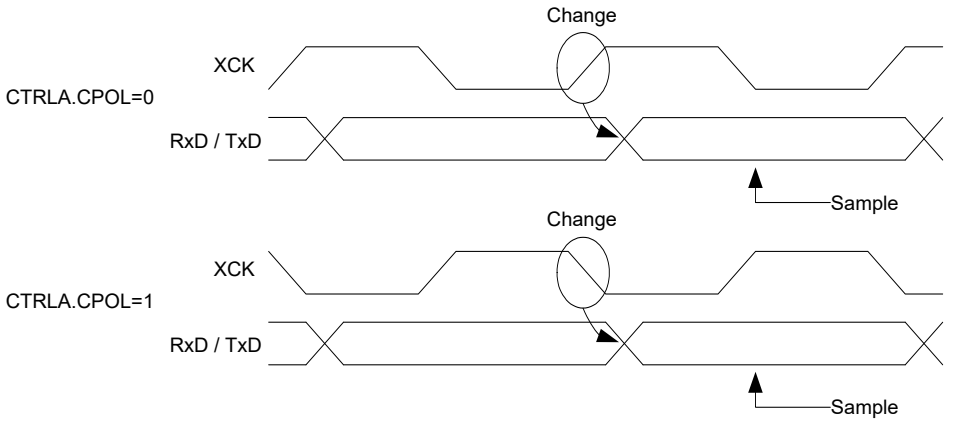
In Synchronous mode, the CTRLA.MODE bit field determines whether the transmission clock line (XCK) serves either as input or output. The dependency between clock edges, data sampling and data change is the same for internal and external clocks. Data input on the RxD pin is sampled at the opposite XCK clock edge when data is driven on the TxD pin.

The Clock Polarity bit in the Control A register (CTRLA.CPOL) selects which XCK clock edge is used for RxD sampling and which is used for TxD change:

- When CTRLA.CPOL is '0', the data is changed on the rising edge of XCK and sampled on the falling edge of XCK.
- When CTRLA.CPOL is '1', the data is changed on the falling edge of XCK and sampled on the rising edge of XCK.

© 2025 Microchip Technology Inc. and its subsidiaries

Figure 32-4. Synchronous Mode XCK Timing



When the clock is provided through XCK (CTRLA.MODE = 0x0), the Shift registers operate directly on the XCK clock. This means that XCK is not synchronized with the system clock and, therefore, can operate at frequencies up to the system frequency.

32.6.2.4. Data Register

The USART Transmit Data register (TxDATA) and USART Receive Data register (RxDATA) share the same I/O address, referred to as the Data register (DATA). Writing the DATA register will update the TxDATA register. Reading the DATA register will return the contents of the RxDATA register.

32.6.2.5. Data Transmission

Data transmission is initiated by writing the data to be sent into the DATA register. Then, the data in TxDATA will be moved to the shift register when the shift register is empty and ready to send a new frame. After the shift register is loaded with data, the data frame will be transmitted.

When the entire data frame including stop bit has been transmitted (both the Tx buffer and the shift register are empty) and no new data was written to DATA register, the Transmit Complete interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.TXC) will be set, and the optional interrupt will be generated.

The Data Register Empty flag in the Interrupt Flag Status and Clear register (INTFLAG.DRE) indicates that the register is empty and ready for new data. The DATA register should only be written to when INTFLAG.DRE is set.

32.6.2.5.1. Disabling the Transmitter

The transmitter is disabled by writing '0' to the Transmitter Enable bit in the CTRLB register (CTRLB.TXEN).

Disabling the transmitter completes only after any ongoing and pending transmissions are completed; in other words, there are no data in the Transmit Shift register and TxDATA to transmit.

32.6.2.6. Data Reception

The receiver accepts data when a valid Start bit is detected. Each bit following the Start bit is sampled according to the baud rate or XCK clock and shifted into the receive Shift register until the first Stop bit of a frame is received. The second Stop bit is ignored by the receiver.

When the first Stop bit is received and a complete serial frame is present in the Receive Shift register, the contents of the Shift register are moved into the two-level receive buffer. Then, the Receive Complete Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.RXC) is set and the optional interrupt is generated.

The received data can be read from the DATA register when the Receive Complete Interrupt flag is set.

© 2025 Microchip Technology Inc. and its subsidiaries

32.6.2.6.1. Disabling the Receiver

Writing '0' to the Receiver Enable bit in the CTRLB register (CTRLB.RXEN) disables the receiver, flushes the two-level receive buffer and data from ongoing receptions are lost.

32.6.2.6.2. Error Bits

The USART receiver has three error bits in the Status (STATUS) register:

- Frame Error (FERR)
- Buffer Overflow (BUFOVF)
- Parity Error (PERR)

When an error happens, the corresponding error bit is set until it is cleared by writing '1' to it. These bits are also cleared automatically when the receiver is disabled.

There are two methods for buffer overflow notification, selected by the Immediate Buffer Overflow Notification bit in the Control A register (CTRLA.IBON):

- When CTRLA.IBON = 1, STATUS.BUFOVF is raised immediately upon buffer overflow. Software can, then, empty the receive FIFO by reading RxDATA until the Receiver Complete Interrupt flag (INTFLAG.RXC) is cleared.
- When CTRLA.IBON = 0, the Buffer Overflow condition is attending data through the receive FIFO, which, then, sets the INTFLAG.ERROR bit. After the received data are read, STATUS.BUFOVF (and INTFLAG.ERROR) is set along with INTFLAG.RXC.

32.6.2.6.3. Asynchronous Data Reception

The USART includes a clock recovery and data recovery unit for handling asynchronous data reception.

The clock recovery logic can synchronize the incoming asynchronous serial frames at the RxD pin to the internally generated baud-rate clock.

The data recovery logic samples and applies a low-pass filter to each incoming bit, thereby improving the noise immunity of the receiver.

32.6.2.6.4. Asynchronous Operational Range

The operational range of the asynchronous reception depends on the accuracy of the internal baud-rate clock, the rate of the incoming frames and the frame size (in number of bits). In addition, the operational range of the receiver depends on the difference between the received bit rate and the internally-generated baud rate. If the baud rate of an external transmitter is too high or too low compared to the internally-generated baud rate, the receiver is not able to synchronize the frames to the start bit.

There are two possible sources for a mismatch in the baud rate:

- First, the reference clock always has some minor instability.
- Second, the baud-rate generator cannot always do an exact division of the reference clock frequency to get the baud rate desired.

In this case, the BAUD register value must be set to give the lowest possible error. See *Clock Generation – Baud-Rate Generator* from Related Links.

The following table provides an overview of the recommended maximum receiver baud-rate errors for various character sizes.

Table 32-3. Asynchronous Receiver Error for 16-fold Oversampling

D (Data Bits+Parity)	R_{SLOW} [%]	R_{FAST} [%]	Max. Total Error [%]	Recommended Max. Rx Error [%]
5	94.12	107.69	+5.88/-7.69	±2.5
6	94.92	106.67	+5.08/-6.67	±2.0
7	95.52	105.88	+4.48/-5.88	±2.0

© 2025 Microchip Technology Inc. and its subsidiaries

Table 32-3. Asynchronous Receiver Error for 16-fold Oversampling (continued)

D (Data Bits+Parity)	R_{SLOW} [%]	R_{FAST} [%]	Max. Total Error [%]	Recommended Max. Rx Error [%]
8	96.00	105.26	+4.00/-5.26	±2.0
9	96.39	104.76	+3.61/-4.76	±1.5
10	96.70	104.35	+3.30/-4.35	±1.5

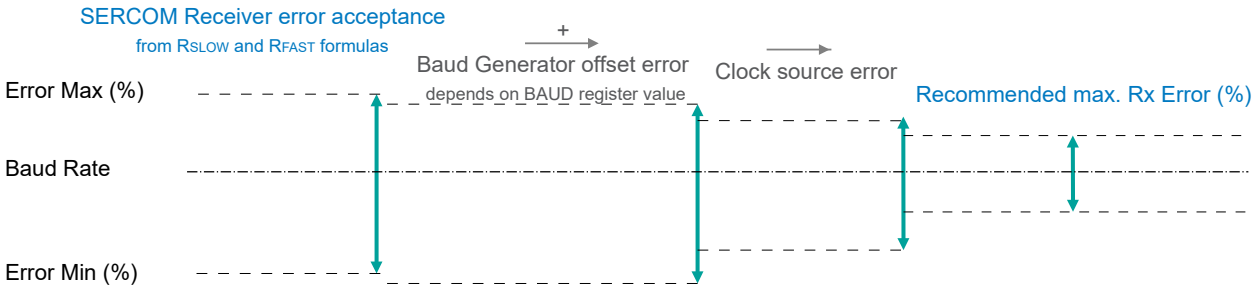
The following equations calculate the ratio of the incoming data rate and internal receiver baud rate:

$$R_{SLOW} = \frac{(D+1)S}{S-1+D\cdot S+S_F} \quad , \quad R_{FAST} = \frac{(D+2)S}{(D+1)S+S_M}$$

- R_{SLOW} is the ratio of the slowest incoming data rate that can be accepted in relation to the receiver baud rate.
- R_{FAST} is the ratio of the fastest incoming data rate that can be accepted in relation to the receiver baud rate.
- D is the sum of the character size and parity size (D = 5 to 10 bits).
- S is the number of samples per bit (S = 16, 8 or 3).
- S_F is the first sample number used for majority voting (S_F = 7, 3 or 2) when CTRLA.SAMPA = 0.
- S_M is the middle sample number used for majority voting (S_M = 8, 4 or 2) when CTRLA.SAMPA = 0.

The recommended maximum RX Error assumes that the receiver and transmitter equally divide the maximum total error. Its connection to the SERCOM Receiver error acceptance is depicted in this figure:

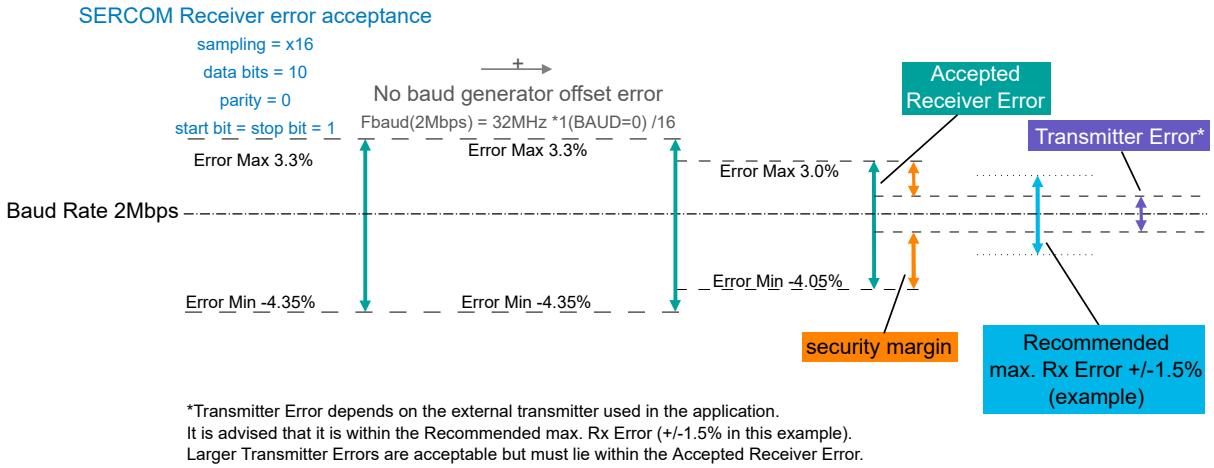
Figure 32-5. USART RX Error Calculation



The recommendation values in the table above accommodate errors of the clock source and the baud generator. The following figure illustrates an example for a baud rate of 3 Mbps:

© 2025 Microchip Technology Inc. and its subsidiaries

Figure 32-6. USART RX Error Calculation Example



Related Links

Clock Generation – Baud-Rate Generator

32.6.3. Additional Features

32.6.3.1. Parity

Even or odd parity can be selected for error checking by writing 0x1 to the Frame Format bit field in the Control A register (CTRLA.FORM).

If even parity is selected (CTRLB.PMODE = 0), the Parity bit of an outgoing frame is '1' if the data contains an odd number of bits that are '1', making the total number of '1' even.

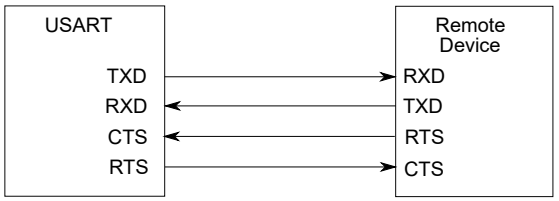
If odd parity is selected (CTRLB.PMODE = 1), the Parity bit of an outgoing frame is '1' if the data contains an even number of bits that are '1', making the total number of '1' odd.

When parity checking is enabled, the parity checker calculates the parity of the data bits in incoming frames and compares the result with the Parity bit of the corresponding frame. If a parity error is detected, the Parity Error bit in the Status register (STATUS.PERR) is set.

32.6.3.2. Hardware Handshaking

The USART features an out-of-band hardware handshaking flow control mechanism, implemented by connecting the RTS and CTS pins with the remote device, as illustrated in the following figure.

Figure 32-7. Connection with a Remote Device for Hardware Handshaking



Hardware handshaking is only available in the following configuration:

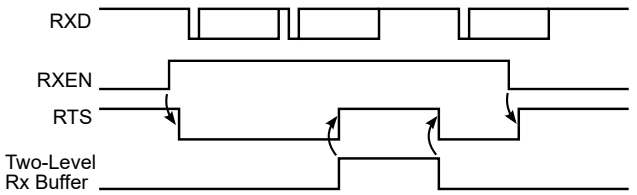
- USART with internal clock (CTRLA.MODE = 1)
- Asynchronous mode (CTRLA.CMODE = 0)
- Flow control pinout (CTRLA.TXPO = 2)

When the receiver is disabled or the receive FIFO is full, the receiver drives the RTS pin high. This notifies the remote device to stop the transfer after the ongoing transmission. Enabling and

© 2025 Microchip Technology Inc. and its subsidiaries

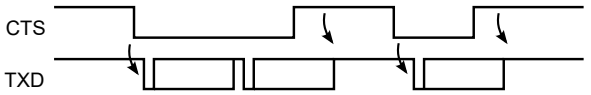
disabling the receiver by writing to CTRLB.RXEN will set/clear the RTS pin after a synchronization delay. When the receive FIFO goes full, RTS is set immediately and the frame being received is stored in the Shift register until the receive FIFO is no longer full.

Figure 32-8. Receiver Behavior when Operating with Hardware Handshaking



The current CTS Status is in the STATUS register (STATUS.CTS). Character transmission starts only if STATUS.CTS = 0. When CTS is set, the transmitter completes the ongoing transmission and stops transmitting.

Figure 32-9. Transmitter Behavior when Operating with Hardware Handshaking



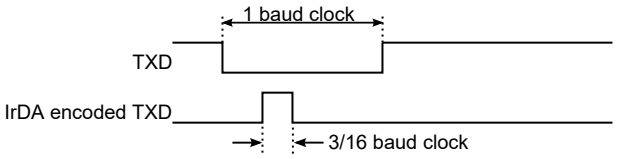
32.6.3.3. IrDA Modulation and Demodulation

Transmission and reception can be encoded IrDA compliant up to 115.2 kbps. IrDA modulation and demodulation work in the following configuration:

- IrDA encoding enabled (CTRLB.ENC = 1)
- Asynchronous mode (CTRLA.CMODE = 0)
- 16x sample rate (CTRLA.SAMPR[0] = 0)

During transmission, each low bit is transmitted as a high pulse. The pulse width is 3/16 of the baud rate period, as illustrated in the following figure.

Figure 32-10. IrDA Transmit Encoding



The reception decoder has two main functions:

- The first is to synchronize the incoming data to the IrDA baud rate counter. Synchronization is performed at the start of each zero pulse.
- The second main function is to decode incoming RX data. If a pulse width meets the minimum length set by configuration (RXPL.RXPL), it is accepted. When the baud rate counter reaches its middle value (1/2 bit length), it is transferred to the receiver.

Notes: The polarity of the transmitter and receiver are opposite:

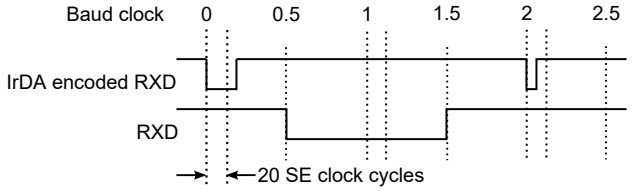
- During transmission, a '0' bit is transmitted as a '1' pulse
- During reception, an accepted '0' pulse is received as a '0' bit

> **Example:** The following figure illustrates reception where RXPL.RXPL is set to 19. This indicates that the pulse width must be at least 20 SE clock cycles. When using BAUD = 0xE666 or 160 SE cycles per bit, this corresponds to 2/16 baud clock as

© 2025 Microchip Technology Inc. and its subsidiaries

minimum pulse width required. In this case, the first bit is accepted as a '0', the second bit is a '1' and the third bit is also a '1'. A low pulse is rejected because it does not meet the minimum requirement of 2/16 baud clock.

Figure 32-11. IrDA Receive Decoding



32.6.3.4. Break Character Detection and Auto-Baud

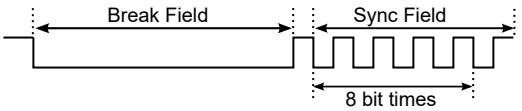
Break character detection and auto-baud are available in this configuration:

- Auto-baud frame format (CTRLA.FORM = 0x04 or 0x05)
- Asynchronous mode (CTRLA.CMODE = 0)
- 16x sample rate using fractional baud rate generation (CTRLA.SAMPR = 1)

The USART uses a break detection threshold of greater than 11 nominal bit times at the configured baud rate. At any time, if more than 11 consecutive dominant bits are detected on the bus, the USART detects a break field. When a break field is detected, the Receive Break Interrupt Flag (INTFLAG.RXBRK) is set and the USART expects the sync field character to be 0x55. This field is used to update the actual baud rate to stay synchronized. If the received sync character is not 0x55, then the Inconsistent Sync Field error flag (STATUS.ISF) is set along with the Error Interrupt Flag (INTFLAG.ERROR) and the baud rate is unchanged.

The auto-baud follows the LIN format. All LIN Frames start with a break field followed by a sync field.

Figure 32-12. LIN Break and Sync Fields



After a break field is detected and the Start bit of the sync field is detected, a counter is started. The counter is, then, incremented for the next 8 bit times of the sync field. At the end of these 8 bit times, the counter is stopped. At this moment, the 13 MSb of the counter (value divided by 8) gives the new clock divider (BAUD.BAUD) and the 3 LSb of this value (the remainder) gives the new Fractional Part (BAUD.FP).

When the sync field is received, the clock divider (BAUD.BAUD) and the Fractional Part (BAUD.FP) are updated after a synchronization delay. After the break and sync fields are received, multiple characters of data can be received.

32.6.3.5. LIN Host

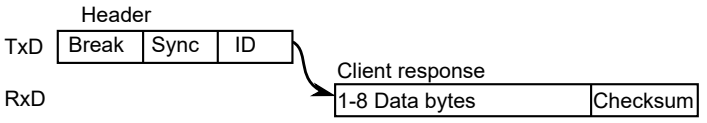
LIN Host is available with the following configuration:

- LIN Host format (CTRLA.FORM = 0x02)
- Asynchronous mode (CTRLA.CMODE = 0)
- 16x sample rate using fractional baud rate generation (CTRLA.SAMPR = 1)

LIN frames start with a header transmitted by the Host. The header consists of the break, sync and identifier fields. After the Host transmits the header, the addressed Client responds with 1-8 bytes of data plus checksum.

© 2025 Microchip Technology Inc. and its subsidiaries

Figure 32-13. LIN Frame Format



Using the LIN command field (CTRLB.LINCMD), the complete header can be automatically transmitted or software can control transmission of the various header components.

When CTRLB.LINCMD = 0x1, software controls transmission of the LIN header. In this case, software uses the following sequence:

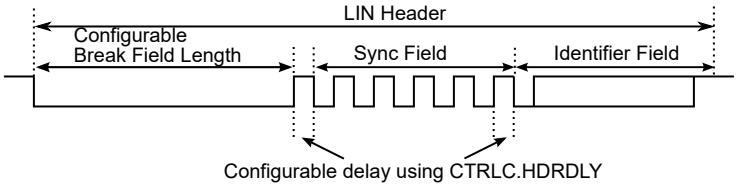
- CTRLB.LINCMD is written to 0x1.
- DATA register is written to 0x00. This triggers transmission of the break field by hardware.
  Note: Writing the DATA register with any other value is also result in the transmission of the break field by hardware.
- DATA register is written to 0x55. The 0x55 value (sync) is transmitted.
- DATA register is written to the identifier. The identifier is transmitted.

When CTRLB.LINCMD = 0x2, hardware controls transmission of the LIN header. In this case, software uses the following sequence:

- CTRLB.LINCMD is written to 0x2.
- DATA register is written to the identifier. This triggers the transmission of the complete header by hardware. First, the break field is transmitted. Next, the sync field is transmitted and, finally, the identifier is transmitted.

In LIN Host mode, the length of the break field is programmable using the break length field (CTRLC.BRKLEN). When the LIN header command is used (CTRLB.LINCMD = 0x2), the delay between the break and sync fields, in addition to the delay between the sync and ID fields, are configurable using the header delay field (CTRLC.HDRDLY). When manual transmission is used (CTRLB.LINCMD = 0x1), software controls the delay between break and sync.

Figure 32-14. LIN Header Generation



After header transmission is complete, the Client responds with 1-8 data bytes plus checksum.

32.6.3.6. RS485

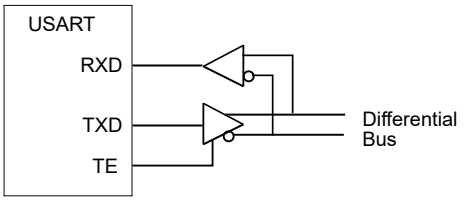
RS485 is available with the following configuration:

- USART frame format (CTRLA.FORM = 0x00 or 0x01)
- RS485 pinout (CTRLA.TXPO = 0x3)

The RS485 feature enables control of an external line driver as illustrated in the following figure. While operating in RS485 mode, the transmit enable pin (TE) is driven high when the transmitter is active.

© 2025 Microchip Technology Inc. and its subsidiaries

Figure 32-15. RS485 Bus Connection



The TE pin remains high for the complete frame including the stop bit(s). If a Guard Time is programmed in the Control C register (CTRLC.GTIME), the line remains driven after the last character completion. The following figure illustrates a transfer with one stop bit and CTRLC.GTIME = 3.

Figure 32-16. Example of TE Drive with Guard Time



The Transmit Complete interrupt flag (INTFLAG.TXC) is raised after the guard time is complete and TE goes low.

32.6.3.7. ISO 7816 for Smart Card Interfacing

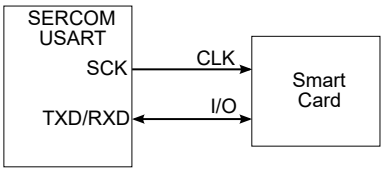
The SERCOM USART features an ISO/IEC 7816-compatible operating mode. This mode permits interfacing with smart cards and Security Access Modules (SAM) communicating through an ISO 7816 link. Both T = 0 and T = 1 protocols defined by the ISO 7816 specification are supported.

ISO 7816 is available with the following configuration:

- ISO 7816 format (CTRLA.FORM = 0x07)
- Inverse transmission and reception (CTRLA.RXINV = 1 and CTRLA.TXINV = 1)
- Single bidirectional data line (CTRLA.TXPO and CTRLA.RXPO configured to use the same data pin)
- Even parity (CTRLB.PMODE = 0)
- 8-bit character size (CTRLB.CHSIZE = 0)
- T = 0 (CTRLA.CMODE = 1) or T = 1 (CTRLA.CMODE = 0)

ISO 7816 is a half-duplex communication on a single bidirectional line. The USART connects to a smart card as shown below. The output is only driven when the USART is transmitting. The USART is considered the host of the communication as it generates the clock.

Figure 32-17. Connection of a Smart Card to the SERCOM USART



ISO 7816 characters are specified as 8 bits with even parity. The USART must be configured accordingly.

The USART cannot operate concurrently in both receiver and transmitter modes as the communication is unidirectional. It has to be configured according to the required mode by enabling or disabling either the receiver or the transmitter as desired. Enabling both the receiver and the transmitter at the same time in ISO 7816 mode may lead to unpredictable results.

© 2025 Microchip Technology Inc. and its subsidiaries

The ISO 7816 specification defines an inverse transmission format. Data bits of the character must be transmitted on the I/O line at their negative value (CTRLA.RXINV = 1 and CTRLA.TXINV = 1).

Protocol T = 0

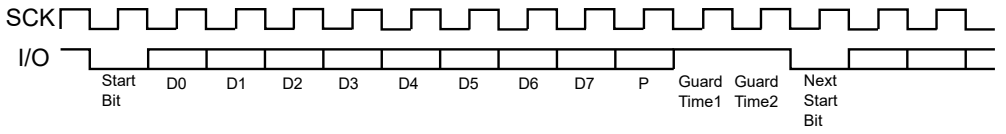
In T = 0 protocol, a character is made up of:

- One start bit
- Eight data bits
- One parity bit
- One guard time, which lasts two bit times

The transfer is synchronous (CTRLA.CMODE = 1). The transmitter shifts out the bits and does not drive the I/O line during the guard time. Additional guard time can be added by programming the Guard Time (CTRLC.GTIME).

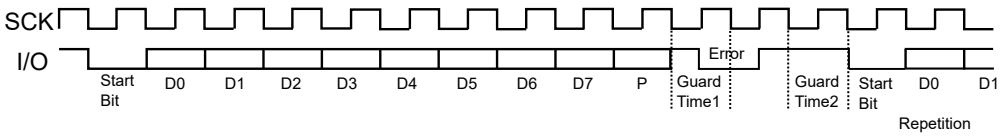
If no parity error is detected, the I/O line remains '1' during the guard time and the transmitter can continue with the transmission of the next character, as illustrated in the following figure.

Figure 32-18. T = 0 Protocol without Parity Error



If a parity error is detected by the receiver, it drives the I/O line to '0' during the guard time, as illustrated in the following figure. This error bit is also named NACK, for Non Acknowledge. In this case, the character lasts '1' bit time more, as the guard time length is the same and is added to the error bit time, which lasts '1' bit time.

Figure 32-19. T = 0 Protocol with Parity Error



When the USART is the receiver and it detects a parity error, the parity error bit in the Status Register (STATUS.PERR) is set and the character is not written to the receive FIFO.

Receive Error Counter

The receiver also records the total number of errors (receiver parity errors and NACKs from the remote transmitter) up to a maximum of 255. This can be read in the Receive Error Count (RXERRCNT) register. RXERRCNT is automatically cleared on read.

Receive NACK Inhibit

The receiver can also be configured to inhibit error generation. This can be achieved by setting the Inhibit Not Acknowledge (CTRLC.INACK) bit. If CTRLC.INACK is '1', no error signal is driven on the I/O line even if a parity error is detected. Moreover, if CTRLC.INACK is set, the erroneous received character is stored in the receive FIFO and the STATUS.PERR bit is set. Inhibit Not Acknowledge (CTRLC.INACK) takes priority over disable successive receive NACK (CTRLC.DSNACK).

Transmit Character Repetition

When the USART is transmitting a character and gets a NACK, it can automatically repeat the character before moving on to the next character. Repetition is enabled by writing the Maximum Iterations register (CTRLC.MAXITER) to a non-zero value. The USART repeats the character the number of times specified in CTRLC.MAXITER.

© 2025 Microchip Technology Inc. and its subsidiaries

When the USART repetition number reaches the programmed value in CTRLC.MAXITER, the STATUS.ITER bit is set and the internal iteration counter is Reset. If the repetition of the character is acknowledged by the receiver before the maximum iteration is reached, the repetitions are stopped and the iteration counter is cleared.

Disable Successive Receive NACK

The receiver can limit the number of successive NACKs sent back to the remote transmitter. This is programmed by setting the Disable Successive NACK bit (CTRLC.DSNACK). The maximum number of NACKs transmitted is programmed in the CTRLC.MAXITER field. As soon as the maximum is reached, the character is considered correct, an acknowledge is sent on the line, the STATUS.ITER bit is set and the internal iteration counter is Reset.

Protocol T = 1

When operating in ISO7816 protocol T = 1, the transmission is asynchronous (CTRL1.CMODE = 0) with one or two stop bits. After the stop bits are sent, the transmitter does not drive the I/O line.

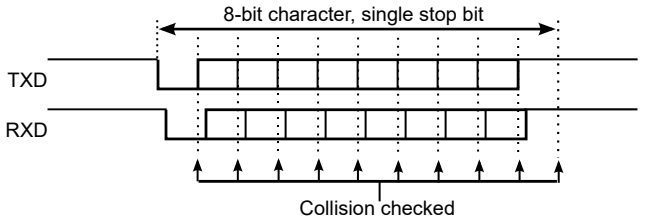
Parity is generated when transmitting and checked when receiving. Parity error detection sets the STATUS.PERR bit and the erroneous character is written to the receive FIFO. When using T = 1 protocol, the receiver does not signal errors on the I/O line and the transmitter does not retransmit.

32.6.3.8. Collision Detection

When the receiver and transmitter are connected either through pin configuration or externally, transmit collision can be detected after selecting the Collision Detection Enable bit in the CTRLB register (CTRLB.COLDEN = 1). To detect collision, the receiver and transmitter must be enabled (CTRLB.RXEN = 1 and CTRLB.TXEN = 1).

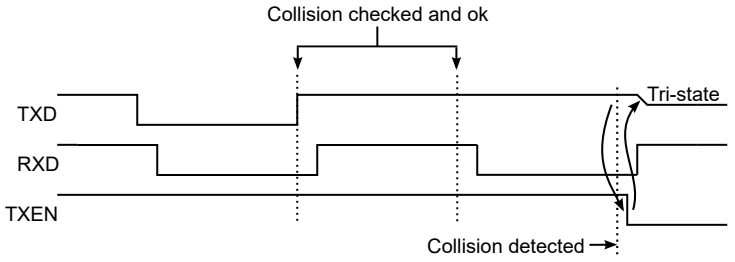
Collision detection is performed for each bit transmitted by comparing the received value with the transmit value, as illustrated in the following figure. While the transmitter is idle (no transmission in progress), characters can be received on RxD without triggering a collision.

Figure 32-20. Collision Checking



The following figure illustrates the conditions for a collision detection. In this case, the Start bit and the first Data bit are received with the same value as transmitted. The second received Data bit is found to be different than the transmitted bit at the detection point, which indicates a collision.

Figure 32-21. Collision Detected



When a collision is detected, the USART follows this sequence:

1. Abort the current transfer.

© 2025 Microchip Technology Inc. and its subsidiaries

2. Flush the transmit buffer.
3. Disable transmitter (CTRLB.TXEN = 0)
   - This is done after a synchronization delay. The CTRLB Synchronization Busy bit (SYNCBUSY.CTRLB) is set until this is complete.
   - After disabling, the TxD pin is tri-stated.
4. Set the Collision Detected bit (STATUS.COLL) along with the Error Interrupt Flag (INTFLAG.ERROR).
5. Set the Transmit Complete Interrupt Flag (INTFLAG.TXC) because the transmit buffer no longer contains data.

After a collision, software must manually enable the transmitter again before continuing, after confirming that the CTRLB Synchronization Busy bit (SYNCBUSY.CTRLB) is not set.

32.6.3.9. Loop-Back Mode

For Loop-Back mode, configure the Receive Data Pinout (CTRLA.RXPO) and Transmit Data Pinout (CTRLA.TXPO) to use the same data pins for transmit and receive. The loop-back is through the pad, so the signal is also available externally.

32.6.3.10. Start-of-Frame Detection

The USART start-of-frame detector can wake up the CPU when it detects a Start bit. In Standby Sleep mode, the internal fast start-up oscillator must be selected as the GCLK_SERCOMx_CORE source.

When a 1-to-0 transition is detected on RxD, the 8 MHz Internal Oscillator is powered up and the USART clock is enabled. After start-up, the rest of the data frame can be received, provided that the baud rate is slow enough in relation to the fast start-up internal oscillator start-up time. For more details, see *Electrical Characteristics* from Related Links. The start-up time of this oscillator varies with supply voltage and temperature.

The USART start-of-frame detection works both in Asynchronous and Synchronous modes. It is enabled by writing '1' to the Start of Frame Detection Enable bit in the Control B register (CTRLB.SFDE).

If the Receive Start Interrupt Enable bit in the Interrupt Enable Set register (INTENSET.RXS) is set, the Receive Start interrupt is generated immediately when a start is detected.

When using start-of-frame detection without the Receive Start interrupt, start detection will force the 8 MHz internal oscillator and USART clock active while the frame is being received. In this case, the CPU will not wake up until the receive complete interrupt is generated.

Related Links

Electrical Characteristics

32.6.3.11. 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 the Control A register (CTRLA.SAMPA). When CTRLA.SAMPA = 0, samples 7-8-9 are used for 16x oversampling and samples 3-4-5 are used for 8x oversampling.

32.6.3.12. 32-Bit Extension

For better system bus utilization, 32-bit data receive and transmit can be enabled separately by writing to the Data 32-bit bit field in the Control C register (CTRLC.DATA32B). When enabled, writes and/or reads to the DATA register are 32-bit in size.

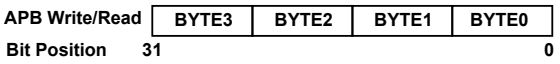
If frames are not multiples of 4 bytes, the length counter (LENGTH.LEN) and length enable (LENGTH.LENEN) must be configured before the data transfer begins, and LENGTH.LEN must be enabled only when CTRLC.DATA32B is enabled.

The following figure illustrates the order of transmit and receive when using a 32-bit extension. Bytes are transmitted or received and stored in order from 0 to 3. Only 8-bit and smaller character

© 2025 Microchip Technology Inc. and its subsidiaries

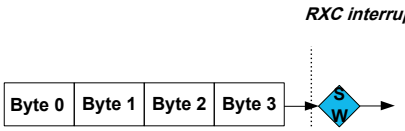
sizes are supported. If the character size is less than 8 bits, characters will still be 8-bit aligned within the 32-bit APB write or read. The unused bits within each byte is zero for received data and unused for transmit data.

Figure 32-22. 32-bit Extension Ordering



A receive transaction using a 32-bit extension is illustrated in the following figure. The Receive Complete flag (INTFLAG.RXC) is raised every four received bytes. For transmit transactions, the Data Register Empty flag (INTFLAG.DRE) is raised instead of INTFLAG.RXC.

Figure 32-23. 32-Bit Extension Receive Operation



Data Length Configuration

When the Data Length Enable bit field in the Length register (LENGTH.LENEN) is written to 0x1 or 0x2, the Data Length bit (LENGTH.LEN) determines the number of characters to be transmitted or received from 1 to 255.

Note: There is one internal length counter that can be used for either transmit (LENGTH.LENEN = 0x1) or receive (LENGTH.LENEN = 0x2) but not for both simultaneously.

The LENGTH register must be written before the frame begins. If LENGTH.LEN is not a multiple of 4 bytes, the final INTFLAG.RXC/DRE interrupt is raised when the last byte is received/sent. The internal length counter is reset when LENGTH.LEN is reached or when LENGTH.LENEN is written to 0x0.

Writing the LENGTH register while a frame is in progress produces unpredictable results. If LENGTH.LENEN is not set and a frame is not a multiple of 4 bytes, the remainder may be lost. Attempting to use the length counter for transmit and receive at the same time produces unpredictable results.

32.6.4. DMA, Interrupts and Events

Table 32-4. Module Request for SERCOM USART

Condition	Request		
	DMA	Interrupt	Event
Data Register Empty (DRE)	Yes (request cleared when data is written)	Yes	NA
Receive Complete (RXC)	Yes (request cleared when data is read)	Yes	
Transmit Complete (TXC)	NA	Yes	
Receive Start (RXS)	NA	Yes	
Clear to Send Input Change (CTSIC)	NA	Yes	
Receive Break (RXBRK)	NA	Yes	
Error (ERROR)	NA	Yes	

32.6.4.1. DMA Operation

The USART generates the following DMA requests:

- Data received (RX) – The request is set when data is available in the receive FIFO. The request is cleared when DATA is read.

© 2025 Microchip Technology Inc. and its subsidiaries

- Data transmit (TX) – The request is set when the transmit buffer (TX DATA) is empty. The request is cleared when DATA is written.

32.6.4.2. Interrupts

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

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

Each interrupt source has its own Interrupt flag. The Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) is 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). The status of enabled interrupts can be read from either INTENSET or 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, see *INTFLAG* from Related Links.

The value of INTFLAG indicates which interrupt is executed.
Note: Interrupts must be globally enabled for interrupt requests. See *Nested Vector Interrupt Controller (NVIC)* from Related Links.

Related Links

Nested Vector Interrupt Controller (NVIC)

INTFLAG

Interrupt Flag Status and Clear

32.6.4.3. Events

Not applicable.

32.6.5. Sleep Mode Operation

The behavior in Sleep mode is dependent 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 Complete interrupt(s) can wake up the device.
- Internal clocking, CTRLA.RUNSTDBY = 0 – The internal clock is disabled, after any ongoing transfer is complete. The Receive Complete interrupt(s) can wake up the device.
- External clocking, CTRLA.RUNSTDBY = 0 – The external clock is disconnected, after any ongoing transfer is complete. All reception is dropped.

32.6.6. Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

© 2025 Microchip Technology Inc. and its subsidiaries

The following bits are synchronized when written:

- Software Reset bit in the CTRLA register (CTRLA.SWRST)
- Enable bit in the CTRLA register (CTRLA.ENABLE)
- Receiver Enable bit in the CTRLB register (CTRLB.RXEN)
- Transmitter Enable bit in the Control B register (CTRLB.TXEN)

Note: CTRLB.RXEN is write-synchronized somewhat differently. See the *CTRLB* register from Related Links.

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

Related Links

CTRLB

Control B

© 2025 Microchip Technology Inc. and its subsidiaries

32.7. Register Summary

See the *SERCOM0/SERCOM1/SERCOM2/SERCOM3/SERCOM4/SERCOM5* module in the *Product Memory Mapping Overview* from Related Links for the base address based on the SERCOM instant used.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
		15:8	SAMPR[2:0]					TXINV	RXINV	IBON
		23:16	SAMPA[1:0]		RXPO[1:0]				TXPO[1:0]	
		31:24		DORD	CPOL	CMODE	FORM[3:0]			
0x04	CTRLB	7:0		SBMODE				CHSIZE[2:0]		
		15:8			PMODE			ENC		COLDEN
		23:16							RXEN	TXEN
		31:24							LINCMD[1:0]	
0x08	CTRLC	7:0						GTIME[2:0]		
		15:8					HDRDLY[1:0]		BRKLEN[1:0]	
		23:16		MAXITER[2:0]					DSNACK	INACK
		31:24							DATA32B[1:0]	
0x0C	BAUD	7:0	BAUD[7:0]							
		15:8	BAUD[15:8]							
0x0E	RXPL	7:0	RXPL[7:0]							
0x0F ... 0x13	Reserved									
0x14	INTENCLR	7:0	ERROR		RXBRK	CTSIC		RXC	TXC	DRE
0x15	Reserved									
0x16	INTENSET	7:0	ERROR		RXBRK	CTSIC		RXC	TXC	DRE
0x17	Reserved									
0x18	INTFLAG	7:0	ERROR		RXBRK	CTSIC		RXC	TXC	DRE
0x19	Reserved									
0x1A	STATUS	7:0	ITER	TXE	COLL	ISF	CTS	BUFOVF	FERR	PERR
		15:8								
0x1C	SYNCBUSY	7:0				LENGTH	RXERRCNT	CTRLB	ENABLE	SWRST
		15:8								
		23:16								
		31:24								
0x20	RXERRCNT	7:0	RXERRCNT[7:0]							
0x21	Reserved									
0x22	LENGTH	7:0	LEN[7:0]							
		15:8							LENEN[1:0]	
0x24 ... 0x27	Reserved									
0x28	DATA	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x2C ... 0x2F	Reserved									
0x30	DBGCTRL	7:0								DBGSTOP

Related Links

Product Memory Mapping Overview

32.8. Register Description

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

© 2025 Microchip Technology Inc. and its subsidiaries

Some registers require synchronization when read and/or written. Synchronization is denoted by the Read-Synchronized and/or Write-Synchronized property in each individual register description.

Optional write protection by the PAC is denoted by the PAC Write Protection property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable protection is denoted by the Enable-Protected property in each individual register description.

© 2025 Microchip Technology Inc. and its subsidiaries

32.8.1. Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
		DORD	CPOL	CMODE	FORM[3:0]			
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	SAMPA[1:0]		RXPO[1:0]				TXPO[1:0]	
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	0	0	0			0	0

Bit	15	14	13	12	11	10	9	8
	SAMPR[2:0]					TXINV	RXINV	IBON
Access	R/W	R/W	R/W			R/W	R/W	R/W
Reset	0	0	0			0	0	0

Bit	7	6	5	4	3	2	1	0
	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			0	0	0	0	0

Bit 30 – DORD Data Order
This bit selects the data order when a character is shifted out from the Data register.
This bit is not synchronized.

Value	Description
0	MSB is transmitted first.
1	LSB is transmitted first.

Bit 29 – CPOL Clock Polarity
This bit selects the relationship between data output change and data input sampling in the Synchronous mode.
This bit is not synchronized.

CPOL	TxD Change	RxD Sample
0x0	Rising XCK edge	Falling XCK edge
0x1	Falling XCK edge	Rising XCK edge

Bit 28 – CMODE Communication Mode
This bit selects asynchronous or synchronous communication.
This bit is not synchronized.

Value	Description
0	Asynchronous communication.
1	Synchronous communication.

Bits 27:24 – FORM[3:0] Frame Format
These bits define the frame format.
These bits are not synchronized.

© 2025 Microchip Technology Inc. and its subsidiaries

FORM[3:0]	Description
0x0	USART frame
0x1	USART frame with parity
0x2	LIN Host - Break and sync generation. See LIN Command (CTRLB.LINCMD).
0x3	Reserved
0x4	Auto-baud (LIN Client) - break detection and auto-baud.
0x5	Auto-baud - break detection and auto-baud with parity
0x6	Reserved
0x7	ISO 7816
0x8-0xF	Reserved

Bits 23:22 – SAMPA[1:0] Sample Adjustment
These bits define the sample adjustment.
These bits are not synchronized.

SAMPA[1:0]	16x Over-sampling (CTRLA.SAMPR=0 or 1)	8x Over-sampling (CTRLA.SAMPR=2 or 3)
0x0	7-8-9	3-4-5
0x1	9-10-11	4-5-6
0x2	11-12-13	5-6-7
0x3	13-14-15	6-7-8

Bits 21:20 – RXPO[1:0] Receive Data Pinout
These bits define the receive data (RxD) pin configuration.
These bits are not synchronized.

RXPO[1:0]	Name	Description
0x0	PAD[0]	SERCOM PAD[0] is used for data reception
0x1	PAD[1]	SERCOM PAD[1] is used for data reception
0x2	PAD[2]	SERCOM PAD[2] is used for data reception
0x3	PAD[3]	SERCOM PAD[3] is used for data reception

Bits 17:16 – TXPO[1:0] Transmit Data Pinout
These bits define the transmit data (TxD) and XCK pin configurations.
This bit is not synchronized.

TXPO	TxD Pin Location	XCK Pin Location (When Applicable)	RTS/TE	CTS
0x0	SERCOM PAD[0]	SERCOM PAD[1]	NA	NA
0x1	Reserved			
0x2	SERCOM PAD[0]	NA	SERCOM PAD[2]	SERCOM PAD[3]
0x3	SERCOM_PAD[0]	SERCOM_PAD[1]	SERCOM_PAD[2]	NA

Bits 15:13 – SAMPR[2:0] Sample Rate
These bits select the sample rate.
These bits are not synchronized.

SAMPR[2:0]	Description
0x0	16x over-sampling using arithmetic baud rate generation.
0x1	16x over-sampling using fractional baud rate generation.
0x2	8x over-sampling using arithmetic baud rate generation.
0x3	8x over-sampling using fractional baud rate generation.
0x4	3x over-sampling using arithmetic baud rate generation.
0x5-0x7	Reserved

Bit 10 – TXINV Transmit Data Invert
This bit controls whether the transmit data (TxD) is inverted or not.

© 2025 Microchip Technology Inc. and its subsidiaries

Note: Start, parity and stop bit(s) are unchanged. When enabled, parity is calculated on the inverted data.

This bit is not synchronized.

Value	Description
0	TxD is not inverted.
1	TxD is inverted.

Bit 9 – RXINV Receive Data Invert

This bit controls whether the receive data (RxD) is inverted or not.
Note: Start, parity and stop bit(s) are unchanged. When enabled, parity is calculated on the inverted data.

This bit is not synchronized.

Value	Description
0	RxD is not inverted.
1	RxD is inverted.

Bit 8 – IBON Immediate Buffer Overflow Notification

This bit controls when the Buffer Overflow Status bit (STATUS.BUFOVF) is asserted when a buffer overflow occurs.
This bit is not synchronized.

Value	Description
0	STATUS.BUFOVF is asserted when it occurs in the data stream.
1	STATUS.BUFOVF is asserted immediately upon buffer overflow.

Bit 7 – RUNSTDBY Run In Standby

This bit defines the functionality in Standby Sleep mode.
This bit is not synchronized.

RUNSTDBY	External Clock	Internal Clock
0x0	External clock is disconnected when ongoing transfer is finished. All reception is dropped.	Generic clock is disabled when ongoing transfer is finished. The device will not wake up on Transfer Complete interrupt unless the appropriate ONDEMAND bits are set in the clocking chain.
0x1	Wake on Receive Complete interrupt.	Generic clock is enabled in all sleep modes. Any interrupt can wake up the device.

Bits 4:2 – MODE[2:0] Operating Mode

These bits select the USART serial communication interface of the SERCOM.
These bits are not synchronized.

Value	Description
0x0	USART with external clock
0x1	USART with internal clock

Bit 1 – ENABLE Enable

Due to synchronization, there is a delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE reads back immediately and the Enable Synchronization Busy bit in the Synchronization Busy register (SYNCBUSY.ENABLE) is set. SYNCBUSY.ENABLE is cleared when the operation is complete.
This bit is not enable-protected.

Value	Description
0	The peripheral is disabled or being disabled.
1	The peripheral is enabled or being enabled.

© 2025 Microchip Technology Inc. and its subsidiaries

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 is disabled.
Writing '1' to CTRLA.SWRST always takes precedence, meaning that all other writes in the same write operation are discarded. Any register write access during the ongoing Reset results in an APB error.
Reading any register returns 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.

© 2025 Microchip Technology Inc. and its subsidiaries

32.8.2. Control B

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

Bit	31	30	29	28	27	26	25	24
							LINCMD[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
							RXEN	TXEN
Access							R/W	R/W
Reset							0	0

Bit	15	14	13	12	11	10	9	8
			PMODE			ENC		COLDEN
Access			R/W			R/W		R/W
Reset			0			0		0

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

Bits 25:24 – LINCMD[1:0] LIN Command

These bits define the LIN header transmission control. This field is only valid in LIN Host mode (CTRLA.FORM = LIN Host).
These are strobe bits and always reads back as '0'.
These bits are not enable-protected.

Value	Description
0x0	Normal USART transmission.
0x1	Break field is transmitted when DATA is written.
0x2	Break, sync and identifier are automatically transmitted when DATA is written with the identifier.
0x3	Reserved

Bit 17 – RXEN Receiver Enable

Writing '0' to this bit disables the USART receiver. Disabling the receiver flushes the receive buffer and clears the FERR, PERR and BUFOVF bits in the STATUS register.
Writing '1' to CTRLB.RXEN when the USART is disabled sets CTRLB.RXEN immediately. When the USART is enabled, CTRLB.RXEN is cleared and SYNCBUSY.CTRLB is set and remains set until the receiver is enabled. When the receiver is enabled, CTRLB.RXEN reads back as '1'.
Writing '1' to CTRLB.RXEN when the USART is enabled sets SYNCBUSY.CTRLB, which remains set until the receiver is enabled and CTRLB.RXEN reads back as '1'.
This bit is not enable-protected.

Value	Description
0	The receiver is disabled or being enabled.
1	The receiver is enabled or is enabled when the USART is enabled.

© 2025 Microchip Technology Inc. and its subsidiaries

Bit 16 – TXEN Transmitter Enable

Writing '0' to this bit disables the USART transmitter. Disabling the transmitter will not become effective until ongoing and pending transmissions are completed.
Writing '1' to CTRLB.TXEN when the USART is disabled sets CTRLB.TXEN immediately. When the USART is enabled, CTRLB.TXEN is cleared and SYNCBUSY.CTRLB is set and remains set until the transmitter is enabled. When the transmitter is enabled, CTRLB.TXEN reads back as '1'.
Writing '1' to CTRLB.TXEN when the USART is enabled sets SYNCBUSY.CTRLB, which remains set until the transmitter is enabled and CTRLB.TXEN reads back as '1'.
This bit is not enable-protected.

Value	Description
0	The transmitter is disabled or being enabled.
1	The transmitter is enabled or is enabled when the USART is enabled.

Bit 13 – PMODE Parity Mode

This bit selects the type of parity used when parity is enabled (CTRLA.FORM is '1'). The transmitter automatically generates and sends the parity of the transmitted data bits within each frame. The receiver generates a parity value for the incoming data and parity bit, compares it to the Parity mode and, if a mismatch is detected, sets STATUS.PERR.
This bit is not synchronized.

Value	Description
0	Even parity
1	Odd parity

Bit 10 – ENC Encoding Format

This bit selects the data encoding format.
This bit is not synchronized.

Value	Description
0	Data is not encoded.
1	Data is IrDA encoded.

Bit 8 – COLDEN Collision Detection Enable

This bit enables collision detection.
This bit is not synchronized.

Value	Description
0	Collision detection is not enabled.
1	Collision detection is enabled.

Bit 6 – SBMODE Stop Bit Mode

This bit selects the number of stop bits transmitted.
This bit is not synchronized.

Value	Description
0	One stop bit
1	Two stop bits

Bits 2:0 – CHSIZE[2:0] Character Size

These bits select the number of bits in a character.
These bits are not synchronized.

CHSIZE[2:0]	Description
0x0	8 bits
0x1	9 bits
0x2-0x4	Reserved

© 2025 Microchip Technology Inc. and its subsidiaries

CHSIZE (continued)	
CHSIZE[2:0]	**Description**
0x5	5 bits
0x6	6 bits
0x7	7 bits

© 2025 Microchip Technology Inc. and its subsidiaries

32.8.3. Control C

Name: CTRLC
Offset: 0x08
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
							DATA32B[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
		MAXITER[2:0]					DSNACK	INACK
Access		R/W	R/W	R/W			R/W	R/W
Reset		0	0	0			0	0

Bit	15	14	13	12	11	10	9	8
					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 25:24 – DATA32B[1:0] Data 32-Bit

These bits configure the 32-bit extension for read and write transactions to the DATA register. When disabled, access is according to CTRLB.CHSIZE.

Value	Description
0x0	DATA reads (for received data) and writes (for transmit data) according to CTRLB.CHSIZE.
0x1	DATA reads according to CTRLB.CHSIZE. DATA writes using 32-bit extension.
0x2	DATA reads using 32-bit extension. DATA writes according to CTRLB.CHSIZE.
0x3	DATA reads and writes using 32-bit extension.

Bits 22:20 – MAXITER[2:0] Maximum Iterations

These bits define the maximum number of retransmit iterations.
These bits also define the successive NACKs sent to the remote transmitter when CTRLC.DSNACK is set.
This field is only valid when using ISO7816 T = 0 mode (CTRLA.MODE = 0x7 and CTRLA.CMODE = 0).

Bit 17 – DSNACK Disable Successive Not Acknowledge

This bit controls how many times NACK is sent on parity error reception.
This bit is only valid in ISO7816 T = 0 mode and when CTRLC.INACK = 0.

Value	Description
0	NACK is sent on the ISO line for every parity error received.
1	Successive parity errors are counted up to the value specified in CTRLC.MAXITER. These parity errors generate a NACK on the ISO line. As soon as this value is reached, no additional NACK is sent on the ISO line.

© 2025 Microchip Technology Inc. and its subsidiaries

Bit 16 – INACK Inhibit Not Acknowledge

This bit controls whether a NACK is transmitted when a parity error is received.
This bit is only valid in ISO7816 T = 0 mode.

Value	Description
0	NACK is transmitted when a parity error is received.
1	NACK is not transmitted when a parity error is received.

Bits 11:10 – HDRDLY[1:0] LIN Host 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 Host mode (CTRLA.FORM = 0x2).
This field is only valid when using the LIN header command (CTRLB.LINCMD = 0x2).
Note: Ensure to select a correct setting for this bit field in order to follow the LIN specification.

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 8-bit time. **Note:** Non-conformal delay will be added between Break and Sync bits.
0x3	Delay between break and sync transmission is 14-bit time. Delay between sync and ID transmission is 14-bit time. **Note:** Non-conformal delay will be added between Break and Sync bits.

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

These bits define the length of the break field transmitted when in LIN Host mode (CTRLA.FORM = 0x2).
Note: Ensure to select a correct setting for this bit field in order to follow the LIN specification.

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.FORM = 0x0 or CTRLA.FORM = 0x1 and CTRLA.TXPO = 0x3) or ISO7816 mode (CTRLA.FORM = 0x7).
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.
For ISO7816 T = 0 mode, the guard time is programmable from 2-9-bit times and defines the guard time between each transmitted byte.

© 2025 Microchip Technology Inc. and its subsidiaries

32.8.4. Baud

Name: BAUD
Offset: 0x0C
Reset: 0x0000
Property: Enable-Protected, PAC Write-Protection

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

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

Bits 15:0 – BAUD[15:0] Baud Value

Arithmetic Baud Rate Generation (CTRLA.SAMPR[0] = 0).
These bits control the clock generation as described in the SERCOM Baud Rate section.
If Fractional Baud Rate Generation (CTRLA.SAMPR = 1 or = 3) 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; see *Clock Generation – Baud-Rate Generator* from Related Links.

- **Bits 12:0 - BAUD[12:0]: Baud Value**

  These bits control the clock generation; see *Clock Generation – Baud-Rate Generator* from Related Links.

Related Links

Clock Generation – Baud-Rate Generator

© 2025 Microchip Technology Inc. and its subsidiaries

32.8.5. Receive Pulse Length Register

Name: RXPL
Offset: 0x0E
Reset: 0x00
Property: Enable-Protected, PAC Write-Protection

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

Bits 7:0 – RXPL[7:0] Receive Pulse Length
When the encoding format is set to IrDA (CTRLB.ENC = 1), these bits control the minimum pulse length that is required for a pulse to be accepted by the IrDA receiver with regards to the serial engine clock period SE_{per}.
$PULSE \geq (\mathrm{RXPL} + 1) \cdot SE_{per}$

© 2025 Microchip Technology Inc. and its subsidiaries

32.8.6. Interrupt Enable Clear

Name: INTENCLR
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection

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

Bit	7	6	5	4	3	2	1	0
	ERROR		RXBRK	CTSIC		RXC	TXC	DRE
Access	R/W		R/W	R/W		R/W	R/W	R/W
Reset	0		0	0		0	0	0

Bit 7 – ERROR Error Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the Error Interrupt Enable bit, which disables the Error interrupt.

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

Bit 5 – RXBRK Receive Break Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the Receive Break Interrupt Enable bit, which disables the Receive Break interrupt.

Value	Description
0	Receive Break interrupt is disabled.
1	Receive Break interrupt is enabled.

Bit 4 – CTSIC Clear to Send Input Change Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the Clear To Send Input Change Interrupt Enable bit, which disables the Clear To Send Input Change interrupt.

Value	Description
0	Clear To Send Input Change interrupt is disabled.
1	Clear To Send Input Change interrupt is enabled.

Bit 2 – RXC Receive Complete Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the Receive Complete Interrupt Enable bit, which disables the Receive Complete interrupt.

Value	Description
0	Receive Complete interrupt is disabled.
1	Receive Complete interrupt is enabled.

Bit 1 – TXC Transmit Complete Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the Transmit Complete Interrupt Enable bit, which disables the Transmit Complete interrupt.

© 2025 Microchip Technology Inc. and its subsidiaries

Value	Description
0	Transmit Complete interrupt is disabled.
1	Transmit Complete interrupt is enabled.

Bit 0 – DRE Data Register Empty Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the Data Register Empty Interrupt Enable bit, which disables the Data Register Empty interrupt.

Value	Description
0	Data Register Empty interrupt is disabled.
1	Data Register Empty interrupt is enabled.

© 2025 Microchip Technology Inc. and its subsidiaries

32.8.7. Interrupt Enable Set

Name: INTENSET
Offset: 0x16
Reset: 0x00
Property: PAC Write-Protection

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

Bit	7	6	5	4	3	2	1	0
	ERROR		RXBRK	CTSIC		RXC	TXC	DRE
Access	R/W		R/W	R/W		R/W	R/W	R/W
Reset	0		0	0		0	0	0

Bit 7 – ERROR Error Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit sets the Error Interrupt Enable bit, which enables the Error interrupt.

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

Bit 5 – RXBRK Receive Break Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit sets the Receive Break Interrupt Enable bit, which enables the Receive Break interrupt.

Value	Description
0	Receive Break interrupt is disabled.
1	Receive Break interrupt is enabled.

Bit 4 – CTSIC Clear to Send Input Change Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit sets the Clear To Send Input Change Interrupt Enable bit, which enables the Clear To Send Input Change interrupt.

Value	Description
0	Clear To Send Input Change interrupt is disabled.
1	Clear To Send Input Change interrupt is enabled.

Bit 2 – RXC Receive Complete Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit sets the Receive Complete Interrupt Enable bit, which enables the Receive Complete interrupt.

Value	Description
0	Receive Complete interrupt is disabled.
1	Receive Complete interrupt is enabled.

Bit 1 – TXC Transmit Complete Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit sets the Transmit Complete Interrupt Enable bit, which enables the Transmit Complete interrupt.

© 2025 Microchip Technology Inc. and its subsidiaries

Value	Description
0	Transmit Complete interrupt is disabled.
1	Transmit Complete interrupt is enabled.

Bit 0 – DRE Data Register Empty Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit sets the Data Register Empty Interrupt Enable bit, which enables the Data Register Empty interrupt.

Value	Description
0	Data Register Empty interrupt is disabled.
1	Data Register Empty interrupt is enabled.

© 2025 Microchip Technology Inc. and its subsidiaries

32.8.8. Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x18
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	ERROR		RXBRK	CTSIC		RXC	TXC	DRE
Access	R/W		R/W	R/W		R	R/W	R
Reset	0		0	0		0	0	0

Bit 7 – ERROR Error
This flag is cleared by writing '1' to it.
This bit is set when any error is detected. Errors that set this flag have corresponding status flags in the STATUS register. Errors that set this flag are COLL, ISF, BUFOVF, FERR and PERR. Writing '0' to this bit has no effect.
Writing '1' to this bit clears the flag.

Bit 5 – RXBRK Receive Break
This flag is cleared by writing '1' to it.
This flag is set when auto-baud is enabled (CTRLA.FORM) and a break character is received.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the flag.

Bit 4 – CTSIC Clear to Send Input Change
This flag is cleared by writing '1' to it.
This flag is set when a change is detected on the CTS pin.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the flag.

Bit 2 – RXC Receive Complete
This flag is cleared by reading the Data register (DATA) or by disabling the receiver.
This flag is set when there are unread data in DATA.
Writing '0' to this bit has no effect.
Writing '1' to this bit has no effect.

Bit 1 – TXC Transmit Complete
This flag is cleared by writing '1' to it or by writing new data to DATA.
This flag is set after the entire frame in the Transmit Shift register is shifted out and there are no new data in DATA.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the flag.

Bit 0 – DRE Data Register Empty
This flag is cleared by writing new data to DATA.
This flag is set when DATA is empty and ready to be written.
Writing '0' to this bit has no effect.
Writing '1' to this bit has no effect.

© 2025 Microchip Technology Inc. and its subsidiaries

32.8.9. Status

Name: STATUS
Offset: 0x1A
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	ITER	TXE	COLL	ISF	CTS	BUFOVF	FERR	PERR
Access	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 7 – ITER Maximum Number of Repetitions Reached
This bit is set when the maximum number of NACK repetitions or retransmissions is met in the ISO7816 T = 0 mode.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears it.

Bit 6 – TXE Transmitter Empty
When CTRLA.FORM is set to LIN Host mode, this bit is set when any ongoing transmission is complete and TxDATA is empty.
When CTRLA.FORM is not set to LIN Host mode, this bit will always read back as '0'.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears it.

Bit 5 – COLL Collision Detected
This bit is cleared by writing '1' to the bit or by disabling the receiver.
This bit is set when collision detection is enabled (CTRLB.COLDEN) and a collision is detected.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears it.

Bit 4 – ISF Inconsistent Sync Field
This bit is cleared by writing '1' to the bit or by disabling the receiver.
This bit is set when the frame format is set to auto-baud (CTRLA.FORM) and a sync field not equal to 0x55 is received.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears it.

Bit 3 – CTS Clear to Send
This bit indicates the current level of the CTS pin when flow control is enabled (CTRLA.TXPO).
Writing '0' to this bit has no effect.
Writing '1' to this bit has no effect.

Bit 2 – BUFOVF Buffer Overflow
Reading this bit before reading the Data register indicates 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.

© 2025 Microchip Technology Inc. and its subsidiaries

This bit is set when a buffer overflow condition is detected. A buffer overflow occurs when the receive buffer is full, there is a new character waiting in the receive shift register and a new start bit is detected.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears it.

Bit 1 – FERR Frame Error

Reading this bit before reading the Data register indicates the error status of the next character to be read.
This bit is cleared by writing '1' to the bit or by disabling the receiver.
This bit is set if the received character had a frame error, in other words, when the first stop bit is '0'.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears it.

Bit 0 – PERR Parity Error

Reading this bit before reading the Data register indicates the error status of the next character to be read.
This bit is cleared by writing '1' to the bit or by disabling the receiver.
This bit is set if parity checking is enabled (CTRLA.FORM is 0x1, 0x5, or 0x7) and a parity error is detected.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears it.

© 2025 Microchip Technology Inc. and its subsidiaries

32.8.10. Synchronization Busy

Name: SYNCBUSY
Offset: 0x1C
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				LENGTH	RXERRCNT	CTRLB	ENABLE	SWRST
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit 4 – LENGTH LENGTH Synchronization Busy

Writing to the LENGTH register requires synchronization. When writing to LENGTH, SYNCBUSY.LENGTH is set until synchronization is complete. If the LENGTH register is written while SYNCBUSY.LENGTH is asserted, an APB error is generated.

Value	Description
0	LENGTH synchronization is not busy.
1	LENGTH synchronization is busy.

Bit 3 – RXERRCNT Receive Error Count Synchronization Busy

The RXERRCNT register is automatically synchronized to the APB domain upon error. When returning from sleep, this bit is raised until the new value is available to be read.

Value	Description
0	RXERRCNT synchronization is not busy.
1	RXERRCNT synchronization is busy.

Bit 2 – CTRLB CTRLB Synchronization Busy

Writing to the CTRLB register when the SERCOM is enabled requires synchronization. When writing to CTRLB the SYNCBUSY.CTRLB, the bit is set until synchronization is complete. If CTRLB is written while SYNCBUSY.CTRLB is asserted, an APB error is generated.

Value	Description
0	CTRLB synchronization is not busy.
1	CTRLB synchronization is busy.

© 2025 Microchip Technology Inc. and its subsidiaries

Bit 1 – ENABLE SERCOM Enable Synchronization Busy

Enabling and disabling the SERCOM (CTRLA.ENABLE) requires synchronization. When writing to CTRLA.ENABLE, the SYNCBUSY.ENABLE bit is set until synchronization is complete.

Value	Description
0	Enable synchronization is not busy.
1	Enable synchronization is busy.

Bit 0 – SWRST Software Reset Synchronization Busy

Resetting the SERCOM (CTRLA.SWRST) requires synchronization. When writing to CTRLA.SWRST, the SYNCBUSY.SWRST bit is set until synchronization is complete.

Value	Description
0	SWRST synchronization is not busy.
1	SWRST synchronization is busy.

© 2025 Microchip Technology Inc. and its subsidiaries

32.8.11. Receive Error Count

Name: RXERRCNT
Offset: 0x20
Reset: 0x00
Property: Read-Synchronized

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

Bits 7:0 – RXERRCNT[7:0] Receive Error Count
This register records the total number of parity errors and NACK errors combined in ISO7816 mode (CTRLA.FORM = 0x7).
This register is automatically cleared on read.

© 2025 Microchip Technology Inc. and its subsidiaries

32.8.12. Length

Name: LENGTH
Offset: 0x22
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
							LENEN[1:0]	
Access							R/W	R/W
Reset							0	0

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

Bits 9:8 – LENEN[1:0] Data Length Enable
In 32-bit Extension mode, this bit field configures the length counter either for transmit or receive transactions.

Value	Description
0x0	Length counter disabled
0x1	Length counter enabled for transmit
0x2	Length counter enabled for receive
0x3	Reserved

Bits 7:0 – LEN[7:0] Data Length
In 32-bit Extension mode, this bit field configures the data length, after which, the flags INTFLAG.RXC or INTFLAG.DRE are raised.

Value	Description
0x00	Reserved if LENEN = 0x1 or LENEN = 0x2
0x01-0xFF	Data Length

© 2025 Microchip Technology Inc. and its subsidiaries

32.8.13. Data

Name: DATA
Offset: 0x28
Reset: 0x0000
Property: -

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

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

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

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

Bits 31:0 – DATA[31:0] Data

Reading these bits returns the contents of the Receive Data register. The register must be read only when the Receive Complete Interrupt Flag bit in the Interrupt Flag Status and Clear register (INTFLAG.RXC) is set. The status bits in STATUS must be read before reading the DATA value to get any corresponding error.

Writing these bits writes the Transmit Data register. This register must be written only when the Data Register Empty Interrupt Flag bit in the Interrupt Flag Status and Clear register (INTFLAG.DRE) is set.

Reads and writes are 32-bit or CTLB.CHSIZE based on the CTRLC.DATA32B setting.

© 2025 Microchip Technology Inc. and its subsidiaries

32.8.14. Debug Control

Name: DBGCTRL
Offset: 0x30
Reset: 0x00
Property: PAC Write-Protection

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

Bit 0 – DBGSTOP Debug Stop Mode
This bit controls the baud-rate generator functionality when the CPU is halted by an external debugger.

Value	Description
0	The baud-rate generator continues normal operation when the CPU is halted by an external debugger.
1	The baud-rate generator is halted when the CPU is halted by an external debugger.

© 2025 Microchip Technology Inc. and its subsidiaries

33. SERCOM Serial Peripheral Interface (SERCOM SPI)

33.1. Overview

The Serial Peripheral Interface (SPI) is one of the available modes in the Serial Communication Interface (SERCOM).

The SPI uses the SERCOM transmitter and receiver configured as illustrated in Block Diagram. Each side, host and client, depicts a separate SPI containing a Shift register, a transmit buffer and a two-level receive buffer. In addition, the SPI host uses the SERCOM baud-rate generator, while the SPI client can use the SERCOM address match logic. Labels in capital letters are synchronous to PB1_CLK and accessible by the CPU, while labels in lowercase letters are synchronous to the SCK clock.

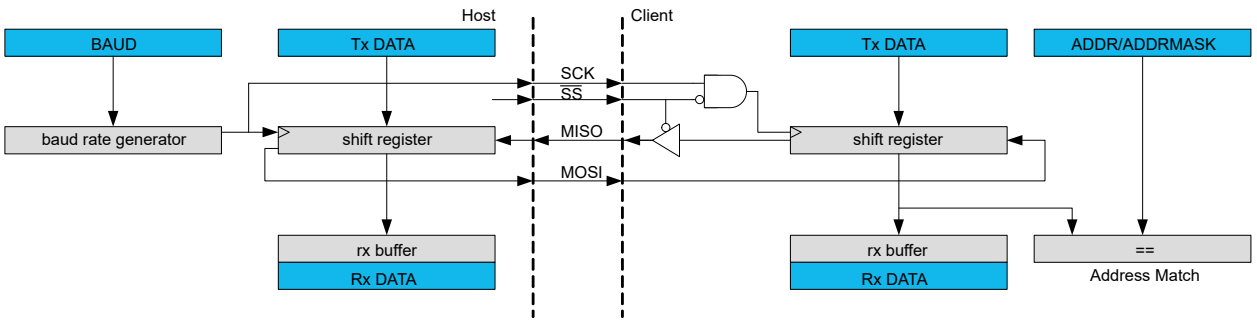
33.2. Features

- Full-Duplex, Four-Wire Interface (MISO, MOSI, SCK, $\overline{SS}$)
- One-Level Transmit Buffer, Two-Level Receive Buffer
- Supports All Four SPI Modes of Operation
- Single Data Direction Operation Allows Alternate Function on MISO or MOSI Pin
- Selectable LSB- or MSB- First Data Transfer
- Can Be Used with DMA
- 32-Bit Extension for Better System Bus Utilization
- Up to 16-Bytes Internal FIFO
- Host Operation:
  - Serial clock speed up to half the system clock
  - 8-bit clock generator
  - Hardware controlled $\overline{SS}$
  - Optional inter-character spacing
- Client Operation:
  - Serial clock speed up to half the system clock
  - Optional 8-bit address match operation
  - Operation in all sleep modes
  - Wake on $\overline{SS}$ transition

Note: SERCOM2 does not have SPI functionality.

33.3. Block Diagram

Figure 33-1. Full-Duplex SPI Host Client Interconnection



© 2025 Microchip Technology Inc. and its subsidiaries

33.4. Signal Description

Table 33-1. SERCOM SPI Signals

Signal Name	Type	Description
PAD[3:0]	Digital I/O	General SERCOM pins

One signal can be mapped to one of several pins. For more details on pin mapping, see *I/O Ports and Peripheral Pin Select (PPS)* from Related Links.

Related Links

I/O Ports and Peripheral Pin Select (PPS)

33.5. Product Dependencies

The following sections describe how the other parts of the system must be configured correctly to use this peripheral.

33.5.1. I/O Lines

To use the SERCOM's I/O lines, the I/O pins must be configured using the System Configuration registers (See *System Configuration and Register Locking (CFG)* from Related Links) (SERCOM_HSEN[2:0] of CFGCON1/DEVCFG1 register) for direct or PPS configuration. If SERCOM pins are selected through PPS, the PPS registers have to be configured. See *I/O Ports and Peripheral Pin Select (PPS)* from Related Links.

If SERCOM_HSEN[x] = 1, SERCOM uses dedicated pins.

If SERCOM_HSEN[x] = 0, SERCOM uses PPS path and I/O pins are multiplexed to pins groups defined in PPS section.

When the SERCOM is configured for SPI operation, the SERCOM controls the direction and value of the I/O pins according to the following table. If the receiver is disabled, the data input pin can be used for other purposes. In Host mode, the SPI Select line ($\overline{SS}$) is hardware-controlled when the Host SPI Select Enable bit in the Control B register (CTRLB.MSSEN) is '1'.

Table 33-2. SPI Pin Configuration

Pin	Host SPI	Client SPI
MOSI	Output	Input
MISO	Input	Output
SCK	Output	Input
$\overline{SS}$	Output (CTRLB.MSSEN = 1)	Input

The configuration of the Data In Pinout and the Data Out Pinout bit groups in the Control A register (CTRLA.DIPO and CTRLA.DOPO) define the physical position of the SPI signals in the Table 33-2.

Related Links

I/O Ports and Peripheral Pin Select (PPS)

System Configuration and Register Locking (CFG)

33.5.2. Power Management

This peripheral can continue to operate in any Sleep mode where its source clock is running. The interrupts can wake-up the device from Sleep modes.

33.5.3. Clocks

A generic clock (GCLK_SERCOMx_CORE) is required to clock the SPI. This clock must be configured and enabled in the Clock and Reset Unit (CRU) and Configuration (CFG.CFGPCLKGEN1) registers before using the SPI.

© 2025 Microchip Technology Inc. and its subsidiaries

This generic clock is asynchronous to the bus clock (PBx_CLK). Therefore, writes to certain registers require synchronization to the clock domains.

33.5.4. DMA

The DMA request lines are connected to the DMA Controller (DMAC). To use DMA requests with this peripheral, the DMAC must be configured first (see *Direct Memory Access Controller (DMAC)* from Related Links).

Related Links

Direct Memory Access Controller (DMAC)

33.5.5. Interrupts

The interrupt request line is connected to the Interrupt Controller. In order to use interrupt requests of this peripheral, the Interrupt Controller (NVIC) must be configured first. See *Nested Vector Interrupt Controller (NVIC)* from Related Links.

Related Links

Nested Vector Interrupt Controller (NVIC)

33.5.6. Events

Not applicable.

33.5.7. Debug Operation

When the CPU is halted in the Debug mode, this peripheral continues normal operation. If the peripheral is configured to require periodical service by the CPU through interrupts or similar, improper operation or data loss may result during debugging. This peripheral can be forced to halt operation during debugging. See *DBGCTRL* register from Related Links.

Related Links

DBGCTRL

Debug Control

33.5.8. Register Access Protection

Registers with write access can be write protected optionally by the PAC.

PAC write protection is not available for the following registers:

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

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

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

33.5.9. Analog Connections

Not applicable.

33.6. Functional Description

33.6.1. Principle of Operation

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

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

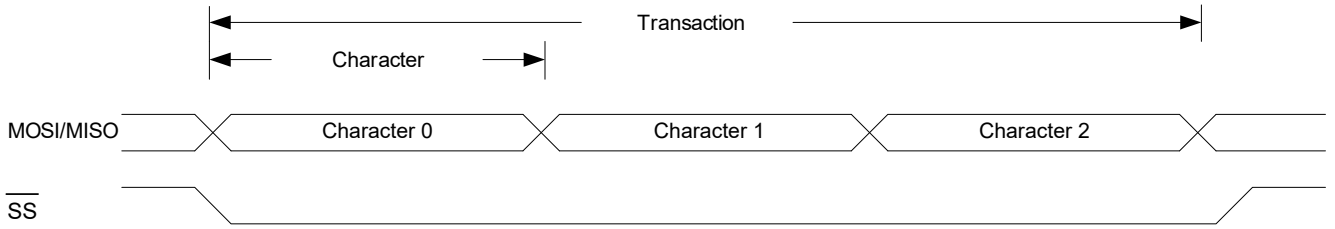
© 2025 Microchip Technology Inc. and its subsidiaries

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

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

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

Figure 33-2. SPI Transaction Format



The SPI Host must pull the SPI select line ($\overline{SS}$) of the desired Client low to initiate a transaction if multiple clients are connected to the bus. The Host and Client prepare data to send via their respective Shift registers, and the Host generates the serial clock on the SCK line.

Data are always shifted from Host to Client on the Host Output Client Input line (MOSI); data is shifted from Client to Host on the Host Input Client Output line (MISO).

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

33.6.2. Basic Operation

33.6.2.1. Initialization

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

- Control A register (CTRLA), except Enable (CTRLA.ENABLE) and Software Reset (CTRLA.SWRST)
- Control B register (CTRLB), except Receiver Enable (CTRLB.RXEN)
- Baud register (BAUD)
- Address register (ADDR)

When the SPI is enabled or is being enabled (CTRLA.ENABLE = 1), any writing to these registers is discarded.

When the SPI is being disabled, writing to these registers is completed after the disabling.

Enable protection is denoted by the Enable-Protection property in the register description.

Initialize the SPI by following these steps:

1. Select SPI mode in host/client operation in the Operating Mode bit group in the CTRLA register (CTRLA.MODE = 0x2 or 0x3).
2. Select Transfer mode for the Clock Polarity bit and the Clock Phase bit in the CTRLA register (CTRLA.CPOL and CTRLA.CPHA) if desired.
3. Select the Frame Format value in the CTRLA register (CTRLA.FORM).
4. Configure the Data In Pinout field in the Control A register (CTRLA.DIPO) for SERCOM pads of the receiver.
5. Configure the Data Out Pinout bit group in the Control A register (CTRLA.DOPO) for SERCOM pads of the transmitter.

© 2025 Microchip Technology Inc. and its subsidiaries

6. Select the Character Size value in the CTRLB register (CTRLB.CHSIZE).
7. Write the Data Order bit in the CTRLA register (CTRLA.DORD) for data direction.
8. If the SPI is used in Host mode:
   a. Select the desired baud rate by writing to the Baud register (BAUD).
   b. If Hardware $\overline{SS}$ control is required, write '1' to the Host SPI Select Enable bit in the CTRLB register (CTRLB.MSSEN).
9. Enable the receiver by writing the Receiver Enable bit in the CTRLB register (CTRLB.RXEN = 1).

33.6.2.2. Enabling, Disabling and Resetting

This peripheral is enabled by writing '1' to the Enable bit in the Control A register (CTRLA.ENABLE), and disabled by writing '0' to it.

Writing '1' to the Software Reset bit in the Control A register (CTRLA.SWRST) resets all registers of this peripheral to their initial states, except the DBGCTRL register, and the peripheral is disabled. See *CTRLA* register from Related Links.

Related Links

CTRLA

Control A

33.6.2.3. Clock Generation

In the SPI host operation (CTRLA.MODE = 0x3), the serial clock (SCK) is generated internally by the SERCOM Baud Rate Generator (BRG).

In the SPI mode, the BRG is set to Synchronous mode. The 8-bit Baud register (BAUD) value is used for generating SCK and clocking the Shift register (see *Clock Generation – Baud-Rate Generator* from Related Links).

In the SPI client operation (CTRLA.MODE = 0x2), the clock is provided by an external host on the SCK pin. This clock is used to clock the SPI Shift register.

Related Links

Clock Generation – Baud-Rate Generator

33.6.2.4. Data Register

The SPI Transmit Data register (TxDATA) and SPI Receive Data register (RxDATA) share the same I/O address, referred to as the SPI Data register (DATA). Writing DATA register will update the Transmit Data register. Reading the DATA register will return the contents of the Receive Data register.

33.6.2.5. SPI Transfer Modes

There are four combinations of SCK phase and polarity to transfer serial data. The SPI Data Transfer modes are shown in the following table and figure.

SCK phase is configured by the Clock Phase bit in the CTRLA register (CTRLA.CPHA). SCK polarity is programmed by the Clock Polarity bit in the CTRLA register (CTRLA.CPOL). Data bits are shifted out and latched in on opposite edges of the SCK signal. This ensures sufficient time for the data signals to stabilize.

Table 33-3. SPI Transfer Modes

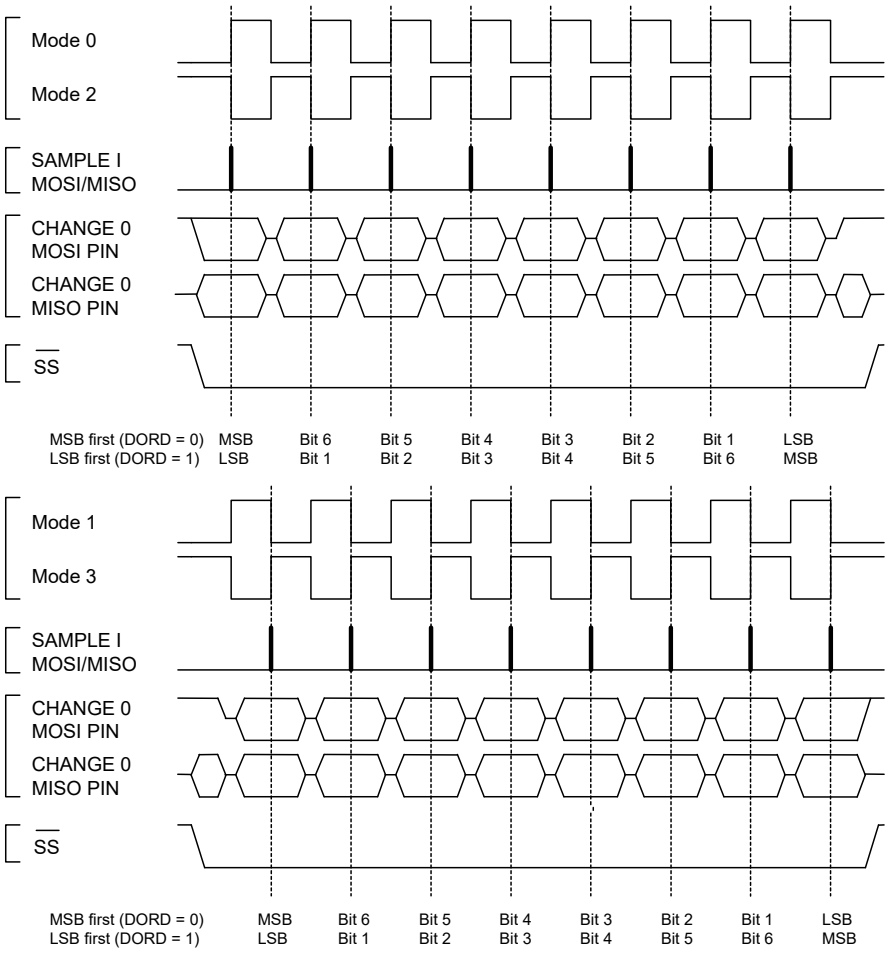
Mode	CPOL	CPHA	Leading Edge	Trailing Edge
0	0	0	Rising, sample	Falling, setup
1	0	1	Rising, setup	Falling, sample
2	1	0	Falling, sample	Rising, setup
3	1	1	Falling, setup	Rising, sample

© 2025 Microchip Technology Inc. and its subsidiaries

Notes:

- The leading edge is the first clock edge in a clock cycle.
- The trailing edge is the second clock edge in a clock cycle.

Figure 33-3. SPI Transfer Modes



33.6.2.6. Transferring Data

33.6.2.6.1. Host

In Host mode (CTRLA.MODE = 0x3), when Host SPI Select Enable (CTRLB.MSSEN) is '1', hardware controls the $\overline{SS}$ line.

When Host SPI Select Enable (CTRLB.MSSEN) is '0', the $\overline{SS}$ line must be configured as an output. $\overline{SS}$ can be assigned to any general purpose I/O pin. When the SPI is ready for a data transaction, software must pull the $\overline{SS}$ line low.

When writing a character to the Data register (DATA), the character is transferred to the Shift register when the shift register is empty. After the content of TxDATA is transferred to the Shift register, the Data Register Empty flag in the Interrupt Flag Status and Clear register (INTFLAG.DRE) is set. And a new character can be written to DATA.

Each time one character is shifted out from the host, another character is shifted in from the client simultaneously. If the receiver is enabled (CTRLA.RXEN = 1), the contents of the Shift register is transferred to the two-level receive buffer. The transfer takes place in the same clock cycle as the last data bit is shifted in. And the Receive Complete Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.RXC) is set. The received data can be retrieved by reading DATA.

© 2025 Microchip Technology Inc. and its subsidiaries

After the last character is transmitted and there are no valid data in DATA, the Transmit Complete Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.TXC) is set. When the transaction is finished, the host must pull the $\overline{SS}$ line high to notify the client. If Host SPI Select Enable (CTRLB.MSSEN) is set to '0', the software must pull the $\overline{SS}$ line high.

33.6.2.6.2. Client

In Client mode (CTRLA.MODE = 0x2), the SPI interface remains inactive with the MISO line tri-stated as long as the $\overline{SS}$ pin is pulled high. Software may update the contents of DATA at any time as long as the Data Register Empty flag in the Interrupt Status and Clear register (INTFLAG.DRE) is set.

When $\overline{SS}$ is pulled low and SCK is running, the client will sample and shift out data according to the Transaction mode set. After the content of TxDATA is loaded into the Shift register, INTFLAG.DRE is set and new data can be written to DATA.

Similar to the host, the client receives one character for each character transmitted. A character is transferred into the two-level receive buffer within the same clock cycle its last data bit received. The received character can be retrieved from DATA when the Receive Complete Interrupt flag (INTFLAG.RXC) is set.

When the host pulls the $\overline{SS}$ line high, the transaction is done and the Transmit Complete Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.TXC) is set.

After DATA is written, it takes up to three SCK clock cycles until the content of DATA is ready to be loaded into the Shift register on the next character boundary. As a consequence, the first character transferred in a SPI transaction will not be the content of DATA. This can be avoided by using the preloading feature. See *Preloading of the Client Shift Register* from Related Links.

When transmitting several characters in one SPI transaction, the data have to be written into the DATA register with at least three SCK clock cycles left in the current character transmission. If this criteria is not met, the previously received character is transmitted.

When the DATA register is empty, it takes three PBx_CLK cycles for INTFLAG.DRE to be set.

Related Links

Preloading of the Client Shift Register

33.6.2.7. Receiver Error Bit

The SPI receiver has one error bit: the Buffer Overflow bit (BUFOVF), which can be read from the Status register (STATUS). When an error happens, the bit stays set until it is cleared by writing '1' to it. The bit is also automatically cleared when the receiver is disabled.

There are two methods for buffer overflow notification, selected by the immediate Buffer Overflow Notification bit in the Control A register (CTRLA.IBON):

- If CTRLA.IBON = 1, STATUS.BUFOVF is raised immediately upon buffer overflow. Software can, then, empty the receive FIFO by reading RxDATA until the receiver complete Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.RXC) goes low.
- If CTRLA.IBON = 0, the Buffer Overflow condition travels with data through the receive FIFO. After the received data are read, STATUS.BUFOVF and INTFLAG.ERROR are set along with INTFLAG.RXC, and RxDATA is '0'.

33.6.3. Additional Features

33.6.3.1. Address Recognition

When the SPI is configured for client operation (CTRLA.MODE = 0x2) with address recognition (CTRLA.FORM = 0x2), the SERCOM address recognition logic is enabled: the first character in a transaction is checked for an address match.

If there is a match, the Receive Complete Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.RXC) is set, the MISO output is enabled and the transaction is processed. If the device is in Sleep mode, an address match can wake up the device to process the transaction.

© 2025 Microchip Technology Inc. and its subsidiaries

If there is no match, the complete transaction is ignored.

If a 9-bit frame format is selected, only the lower 8 bits of the Shift register are checked against the Address register (ADDR).

Preload must be disabled (CTRLB.PLOADEN = 0) to use this mode.

33.6.3.2. Preloading of the Client Shift Register

When starting a transaction, the client will first transmit the contents of the shift register before loading new data from DATA. The first character sent can be either the reset value of the shift register (if this is the first transmission since the last reset) or the last character in the previous transmission.

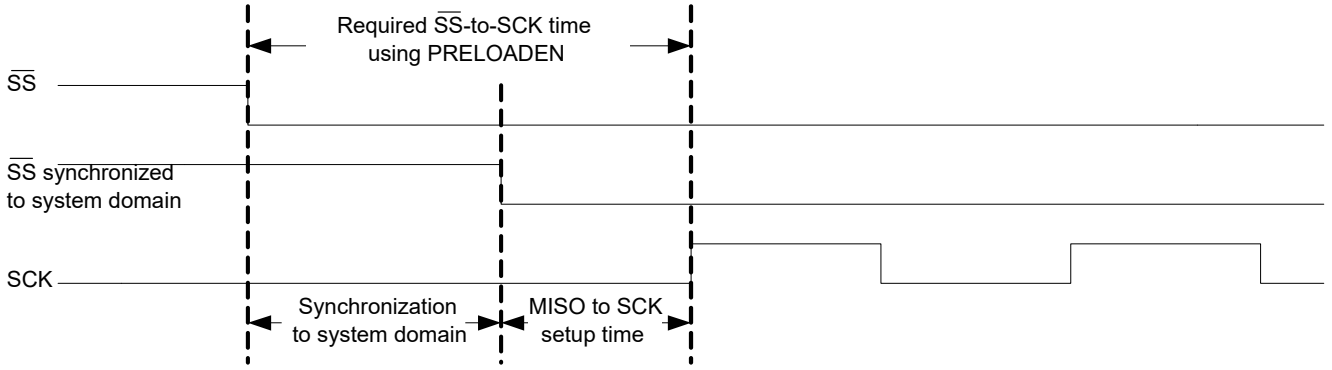
Preloading can be used to preload data into the shift register while $\overline{SS}$ is high: this eliminates sending a dummy character when starting a transaction. If the shift register is not preloaded, the current contents of the shift register will be shifted out.

Only one data character will be preloaded into the shift register while the synchronized $\overline{SS}$ signal is high. If the next character is written to DATA before $\overline{SS}$ is pulled low, the second character will be stored in DATA until transfer begins.

For proper preloading, sufficient time must elapse between $\overline{SS}$ going low and the first SCK sampling edge, as shown in the following figure. For timing details, see *Electrical Characteristics* from Related Links.

Preloading is enabled by writing '1' to the Client Data Preload Enable bit in the CTRLB register (CTRLB.PLOADEN).

Figure 33-4. Timing Using Preloading



Related Links

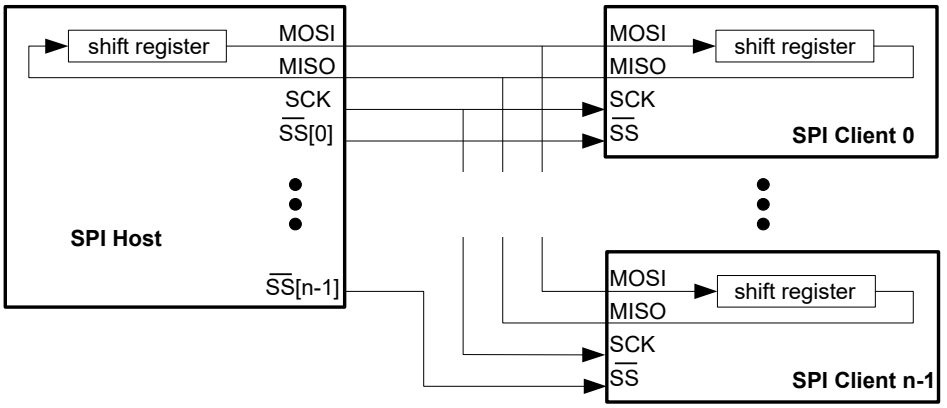
Electrical Characteristics

33.6.3.3. Host with Several Clients

Host with multiple clients in parallel is only available when Host SPI Select Enable (CTRLB.MSSEN) is set to zero and hardware $\overline{SS}$ control is disabled. If the bus consists of several SPI clients, a SPI host can use general purpose I/O pins to control the $\overline{SS}$ line to each of the clients on the bus, as shown in the following figure. In this configuration, the single selected SPI client will drive the tri-state MISO line.

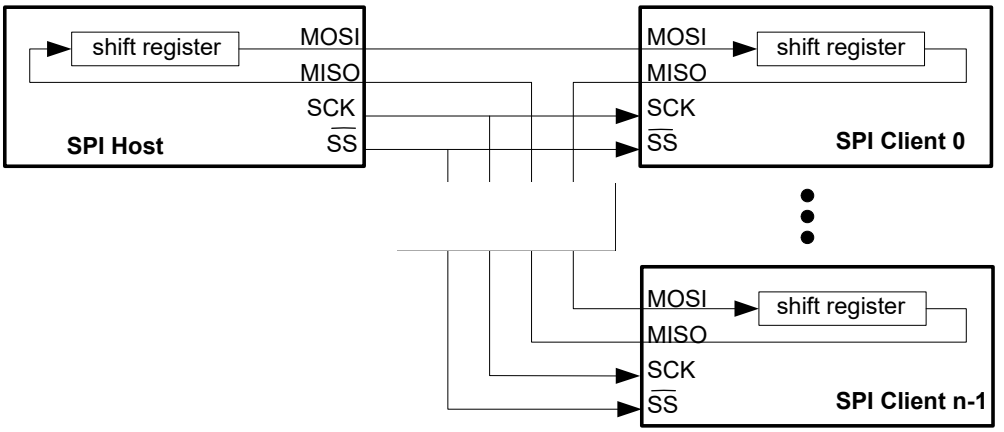
© 2025 Microchip Technology Inc. and its subsidiaries

Figure 33-5. Multiple Clients in Parallel



Another configuration is multiple clients in series, as shown in the following figure. In this configuration, all n attached clients are connected in series. A common $\overline{SS}$ line is provided to all clients, enabling them simultaneously. The host must shift n characters for a complete transaction. Depending on the Host SPI Select Enable bit (CTRLB.MSSEN), the $\overline{SS}$ line can be controlled either by hardware or user software and normal GPIO.

Figure 33-6. Multiple Clients in Series



33.6.3.4. Loop-Back Mode

For Loop-back mode, configure the Data In Pinout (CTRLA.DIPO) and Data Out Pinout (CTRLA.DOPO) to use the same data pins for transmit and receive. The loop-back is through the pad, so the signal is also available externally.

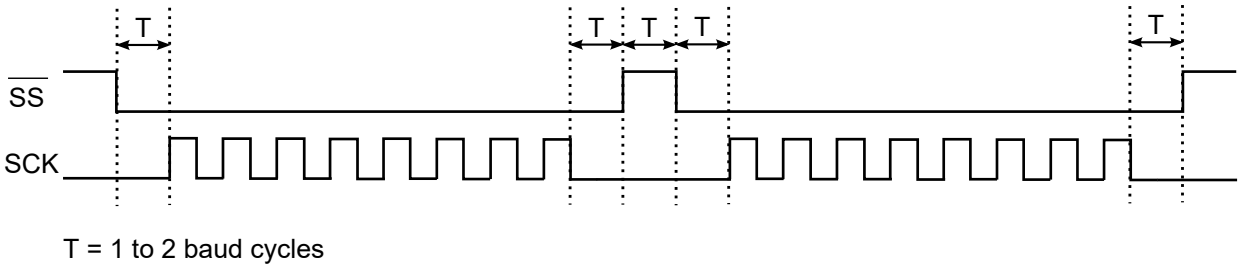
33.6.3.5. Hardware Controlled $\overline{SS}$

In Host mode, a single $\overline{SS}$ chip select can be controlled by hardware by writing the Host SPI Select Enable (CTRLB.MSSEN) bit to '1'. In this mode, the $\overline{SS}$ pin is driven low for a minimum of one baud cycle before transmission begins, and stays low for a minimum of one baud cycle after transmission completes. If back-to-back frames are transmitted, the $\overline{SS}$ pin will always be driven high for a minimum of one baud cycle between frames.

In the following figure, the time, T, is between one and two baud cycles depending on the SPI Transfer mode.

© 2025 Microchip Technology Inc. and its subsidiaries

Figure 33-7. Hardware Controlled $\overline{SS}$



When CTRLB.MSSEN = 0, the $\overline{SS}$ pin(s) is/are controlled by user software and normal GPIO.

33.6.3.6. SPI Select Low Detection

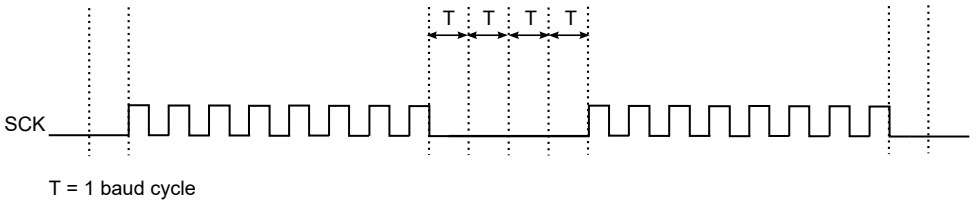
In Client mode, the SPI can wake the CPU when the SPI Select ($\overline{SS}$) goes low. When the SPI Select Low Detect is enabled (CTRLB.SSDE = 1), a high-to-low transition sets the SPI Select Low Interrupt flag (INTFLAG.SSL) and the device will wake up if applicable.

33.6.3.7. Host Inter-Character Spacing

When configured as the host, inter-character spacing can be increased by writing a non-zero value to the Inter-Character Spacing bit field in the Control C register (CTRLC.ICSPACE). When non-zero, CTRLC.ICSPACE represents the minimum number of baud cycles that the SCK clock line does not toggle and the next character is stalled.

The following figure illustrates an example for CTRLC.ICSPACE = 4; in this case, the SCK is inactive for four baud cycles.

Figure 33-8. Four Cycle Inter-Character Spacing Example



33.6.3.8. 32-Bit Extension

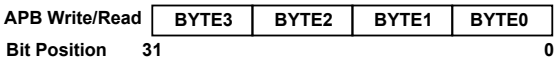
For better system bus utilization, 32-bit data receive and transmit can be enabled by writing to the Data 32-bit bit field in the Control C register (CTRLC.DATA32B = 1). When enabled, write and read transactions to/from the DATA register are 32 bits in size.

If frames are not multiples of 4 bytes, the Length Counter (LENGTH.LEN) and Length Enable (LENGTH.LENEN) must be configured before data transfer begins. LENGTH.LEN must be enabled only when CTRLC.DATA32B is enabled.

The following figure illustrates the order of transmit and receive when using 32-bit mode. Bytes are transmitted or received and stored in order from 0 to 3.

Only 8-bit character size is supported.

Figure 33-9. 32-Bit Extension Byte Ordering



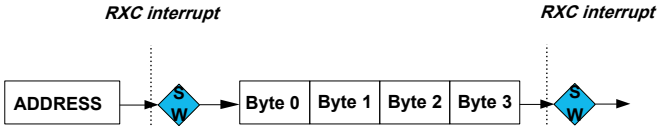
32-Bit Extension Client Operation

The following figure illustrates a transaction with 32-bit Extension enabled (CTRLC.DATA32B = 1). When address recognition is enabled (CTRLA.FORM = 0x2) and there is an address match, the

© 2025 Microchip Technology Inc. and its subsidiaries

address is loaded into the FIFO as Byte zero and data begins with Byte 1. INTFLAGS.RXC will, then, be raised for every 4 bytes transferred. For transmit, there is a 32-bit holding buffer in the core domain. After DATA is registered in the core domain, INTFLAG.DRE is raised so that the next 32 bits can be written to the DATA register.

Figure 33-10. 32-Bit Extension Client Operation



When utilizing the length counter, the LENGTH register must be written before the frame begins. If the frame length while $\overline{SS}$ is low is not a multiple of LENGTH.LEN bytes, the Length Error Status bit (STATUS.LENERR) is raised. If LENGTH.LEN is not a multiple of 4 bytes, the final INTFLAG.RXC interrupt is raised when the last byte is received.

The length count is based on the received bytes or the number of clocks if the receiver is not enabled. If pre-loading is disabled and DATA is written for transmit before SCK starts, transmitted data are delayed by one byte, but the length counter will still increment for the first (empty) byte transmission. When the counter reaches LENGTH.LEN, the internal length counter, RX byte counter and TX byte counter are reset. If multiple lengths are to be transmitted, INTFLAG.TXC must go high before writing DATA for subsequent lengths.

If there is a Length Error (STATUS.LENERR), the remaining bytes in the length are transmitted at the beginning of the next frame. If this is not desired, the SERCOM must be disabled and re-enabled to flush the TX and RX pipelines.

Writing the LENGTH register while a frame is in progress produces unpredictable results. If LENGTH.LENEN is not configured and a frame is not a multiple of 4 bytes (while $\overline{SS}$ is low), the remainder is transmitted in the next frame.

32-Bit Extension Host Operation

When using the SPI configured as the host, the Length and the Length Enable bit fields (LENGTH.LEN and LENGTH.LENEN) must be written before the frame begins. When LENGTH.LENEN is written to '1', the value of LENGTH.LEN determines the number of data bytes in the transaction from 1 to 255.

For receive data, INTFLAG.RXC is raised every 4 bytes received. If LENGTH.LEN is not a multiple of 4 bytes, the final INTFLAG.RXC is set when the final byte is received.

For transmit, there is a holding buffer for the 32-bit data in the core domain. After DATA is registered in the SCK domain, INTFLAG.DRE is raised so that the next 32 bits can be written to the DATA register.

If multiple lengths are to be transmitted, INTFLAG.TXC must go high before writing DATA for subsequent lengths.

© 2025 Microchip Technology Inc. and its subsidiaries

33.6.4. DMA, Interrupts and Events

Table 33-4. Module Request for SERCOM SPI

Condition	Request		
	DMA	Interrupt	Event
Standard (DRE) – Data Register Empty FIFO (DRE) – at least TXTRHOLD locations in TX FIFO are empty	Yes (request cleared when data are written)	Yes	NA
Standard (RXC) – Receive Complete FIFO (RXC) – at least RXTRHOLD data available in RX FIFO, or a last word available and length frame reception completed.	Yes (request cleared when data are read)	Yes	
Standard (TXC) – Transmit Complete FIFO (TXC) – Transmit Complete and TX FIFO is empty	NA	Yes	
SPI Select Low (SSL)	NA	Yes	
Error (ERROR)	NA	Yes	

33.6.4.1. DMA Operation

The SPI generates the following DMA requests:

- Data received (RX) – The request is set when data are available in the receive FIFO. The request is cleared when DATA is read.
- Data transmit (TX) – The request is set when the transmit buffer (TX DATA) is empty. The request is cleared when DATA is written.
- Data received (RX) – The request is set when data are available in the receive FIFO or if at least RXTRHOLD data are available in the RX FIFO when FIFO operation is enabled. The request is cleared when DATA is read.
- Data transmit (TX) – The request is set when the transmit buffer (TX DATA) is empty or if at least TXTRHOLD data locations are empty in the TX FIFO, when FIFO operation is enabled. The request is cleared when DATA is written.

33.6.4.2. Interrupts

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

- Data Register Empty (DRE)
- Receive Complete (RXC)
- Transmit Complete (TXC)
- SPI Select Low (SSL)
- Error (ERROR)

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

© 2025 Microchip Technology Inc. and its subsidiaries

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

The value of INTFLAG indicates which interrupt is executed. Note that interrupts must be globally enabled for interrupt requests. See *Nested Vector Interrupt Controller (NVIC)* from Related Links.

Related Links

Nested Vector Interrupt Controller (NVIC)

INTFLAG

Interrupt Flag Status and Clear

33.6.4.3. Events

Not applicable.

33.6.5. Sleep Mode Operation

The behavior in Sleep mode depends on the host/client configuration and the Run In Standby bit in the Control A register (CTRLA.RUNSTDBY):

- Host operation, CTRLA.RUNSTDBY = 1 – The peripheral clock GCLK_SERCOMx_CORE continues to run in Idle Sleep mode and in Standby Sleep mode. Any interrupt can wake up the device.
- Host operation, CTRLA.RUNSTDBY = 0 – GCLK_SERCOMx_CORE is disabled after the ongoing transaction is finished. Any interrupt can wake up the device.
- Client operation, CTRLA.RUNSTDBY = 1 – The Receive Complete interrupt can wake up the device.
- Client operation, CTRLA.RUNSTDBY = 0 – All reception is dropped, including the ongoing transaction.

33.6.6. Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read. When executing an operation that requires synchronization, the corresponding Synchronization Busy bit in the Synchronization register (SYNCBUSY) is set immediately and cleared when synchronization is complete. If an operation that requires synchronization is executed while the corresponding SYNCBUSY bit is one, a peripheral bus error is generated.

The following bits need to be synchronized when written:

- Software Reset bit in the CTRLA register (CTRLA.SWRST)
- Enable bit in the CTRLA register (CTRLA.ENABLE)
- Receiver Enable bit in the CTRLB register (CTRLB.RXEN)

Note: CTRLB.RXEN is write-synchronized somewhat differently. See *CTRLB* register from Related Links.

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

Related Links

CTRLB

Control B

© 2025 Microchip Technology Inc. and its subsidiaries

33.7. Register Summary

See the *SERCOMx (X=0,1,2)* module in the *Product Memory Mapping Overview* from Related Links for the base address based on the SERCOM instant used.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
		15:8								IBON
		23:16			DIPO[1:0]				DOPO[1:0]	
		31:24		DORD	CPOL	CPHA	FORM[3:0]			
0x04	CTRLB	7:0		PLOADEN				CHSIZE[2:0]		
		15:8	AMODE[1:0]		MSSEN				SSDE	
		23:16							RXEN	
		31:24								
0x08	CTRLC	7:0			ICSPACE[5:0]					
		15:8								
		23:16								
		31:24								DATA32B
0x0C	BAUD	7:0	BAUD[7:0]							
0x0D ... 0x13	Reserved									
0x14	INTENCLR	7:0	ERROR				SSL	RXC	TXC	DRE
0x15	Reserved									
0x16	INTENSET	7:0	ERROR				SSL	RXC	TXC	DRE
0x17	Reserved									
0x18	INTFLAG	7:0	ERROR				SSL	RXC	TXC	DRE
0x19	Reserved									
0x1A	STATUS	7:0						BUFOVF		
		15:8					LENERR			
0x1C	SYNCBUSY	7:0				LENGTH		CTRLB	ENABLE	SWRST
		15:8								
		23:16								
		31:24								
0x20 ... 0x21	Reserved									
0x22	LENGTH	7:0	LEN[7:0]							
		15:8								LENEN
0x24	ADDR	7:0	ADDR[7:0]							
		15:8								
		23:16	ADDRMASK[7:0]							
		31:24								
0x28	DATA	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x2C ... 0x2F	Reserved									
0x30	DBGCTRL	7:0								DBGSTOP

Related Links

Product Memory Mapping Overview

33.8. Register Description

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

© 2025 Microchip Technology Inc. and its subsidiaries

Some registers require synchronization when read and/or written. Synchronization is denoted by the Read-Synchronized and/or Write-Synchronized property in each individual register description.

Optional write protection by the PAC is denoted by the PAC Write Protection property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable protection is denoted by the Enable-Protected property in each individual register description.

© 2025 Microchip Technology Inc. and its subsidiaries

33.8.1. Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
		DORD	CPOL	CPHA	FORM[3:0]			
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
			DIPO[1:0]				DOPO[1:0]	
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit	15	14	13	12	11	10	9	8
								IBON
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			0	0	0	0	0

Bit 30 – DORD Data Order
This bit selects the data order when a character is shifted out from the shift register.
This bit is not synchronized.

Value	Description
0	MSB is transferred first.
1	LSB is transferred first.

Bit 29 – CPOL Clock Polarity
In combination with the Clock Phase bit (CPHA), this bit determines the SPI Transfer mode.
This bit is not synchronized.

Value	Description
0	SCK is low when idle. The leading edge of a clock cycle is a rising edge, while the trailing edge is a falling edge.
1	SCK is high when idle. The leading edge of a clock cycle is a falling edge, while the trailing edge is a rising edge.

Bit 28 – CPHA Clock Phase
In combination with the Clock Polarity bit (CPOL), this bit determines the SPI Transfer mode.
This bit is not synchronized.

Mode	CPOL	CPHA	Leading Edge	Trailing Edge
0x0	0	0	Rising, sample	Falling, change
0x1	0	1	Rising, change	Falling, sample
0x2	1	0	Falling, sample	Rising, change
0x3	1	1	Falling, change	Rising, sample

Value	Description
0	The data are sampled on a leading SCK edge and changed on a trailing SCK edge.

© 2025 Microchip Technology Inc. and its subsidiaries

Value	Description
1	The data are sampled on a trailing SCK edge and changed on a leading SCK edge.

Bits 27:24 – FORM[3:0] Frame Format

This bit field selects the various frame formats supported by the SPI in Client mode. When the SPI frame with address format is selected, the first byte received is checked against the ADDR register.

FORM[3:0]	Name	Description
0x0	SPI	SPI frame
0x1	—	Reserved
0x2	SPI_ADDR	SPI frame with address
0x3-0xF	—	Reserved

Bits 21:20 – DIPO[1:0] Data In Pinout

These bits define the data in (DI) pad configurations.
In host operation, DI is MISO.
In client operation, DI is MOSI.
These bits are not synchronized.

DIPO[1:0]	Name	Description
0x0	PAD[0]	SERCOM PAD[0] is used as data input
0x1	PAD[1]	SERCOM PAD[1] is used as data input
0x2	PAD[2]	SERCOM PAD[2] is used as data input
0x3	PAD[3]	SERCOM PAD[3] is used as data input

Bits 17:16 – DOPO[1:0] Data Out Pinout

These bits define the available pad configurations for Data Out (DO), the Serial Clock (SCK) and the SPI Select ($\overline{SS}$). In Client operation, the SPI Select line ($\overline{SS}$) is controlled by DOPO. In host operation, the SPI Select line ($\overline{SS}$) is either controlled by DOPO when CTRLB.MSSEN = 1, or by a GPIO driven by the application when CTRLB.MSSEN = 0.
In host operation, DO is MOSI.
In client operation, DO is MISO.
These bits are not synchronized.

DOPO	DO	SCK	Client $\overline{SS}$	Host $\overline{SS}$ (MSSEN = 1)	Host $\overline{SS}$ (MSSEN = 0)
0x0	PAD[0]	PAD[1]	PAD[2]	PAD[2]	Any GPIO configured by the application
0x1	Reserved				
0x2	PAD[3]	PAD[1]	PAD[2]	PAD[2]	Any GPIO configured by the application
0x3	Reserved				

Bit 8 – IBON Immediate Buffer Overflow Notification

This bit controls when the Buffer Overflow Status bit (STATUS.BUFOVF) is set when a buffer overflow occurs.
This bit is not synchronized.

Value	Description
0	STATUS.BUFOVF is set when it occurs in the data stream.
1	STATUS.BUFOVF is set immediately upon buffer overflow.

Bit 7 – RUNSTDBY Run In Standby

This bit defines the functionality in Standby Sleep mode.
This bit is not synchronized.

RUNSTDBY	Client	Host
0x0	Disabled. All reception is dropped, including the ongoing transaction.	Generic clock is disabled when ongoing transaction is finished. All interrupts can wake up the device.

© 2025 Microchip Technology Inc. and its subsidiaries

RUNSTDBY (continued)		
RUNSTDBY	**Client**	**Host**
0x1	Ongoing transaction continues, wake on Receive Complete interrupt.	Generic clock is enabled while in Sleep modes. All interrupts can wake up the device.

Bits 4:2 – MODE[2:0] Operating Mode

These bits must be written to 0x2 or 0x3 to select the SPI serial communication interface of the SERCOM.
0x2 – SPI client operation
0x3 – SPI host operation
These bits are not synchronized.

Bit 1 – ENABLE Enable

Due to synchronization, there is a delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRL.ENABLE reads back immediately and the Synchronization Enable Busy bit in the Synchronization Busy register (SYNCBUSY.ENABLE) is set. SYNCBUSY.ENABLE is cleared when the operation is complete.
This bit is not enable-protected.

Value	Description
0	The peripheral is disabled or being disabled.
1	The peripheral is enabled or being enabled.

Bit 0 – SWRST Software Reset

Writing '0' to this bit has no effect.
Writing '1' to this bit resets all registers in the SERCOM, except DBGCTRL, to their initial state and the SERCOM is disabled.
Writing '1' to CTRL.SWRST will always take precedence, meaning that all other writes in the same write operation is discarded. Any register write access during the ongoing Reset results in an APB error. Reading any register returns 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.

© 2025 Microchip Technology Inc. and its subsidiaries

33.8.2. Control B

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

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							RXEN	
Access							R/W	
Reset							0	

Bit	15	14	13	12	11	10	9	8
	AMODE[1:0]		MSSEN				SSDE	
Access	R/W	R/W	R/W				R/W	
Reset	0	0	0				0	

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

Bit 17 – RXEN Receiver Enable

Writing '0' to this bit disables the SPI receiver immediately. The receive buffer is flushed, data from ongoing receptions are lost and STATUS.BUFOVF is cleared.
Writing '1' to CTRLB.RXEN when the SPI is disabled sets CTRLB.RXEN immediately. When the SPI is enabled, CTRLB.RXEN is cleared, SYNCBUSY.CTRLB is set and remains set until the receiver is enabled. When the receiver is enabled, CTRLB.RXEN reads back as '1'.
Writing '1' to CTRLB.RXEN when the SPI is enabled sets SYNCBUSY.CTRLB, which remains set until the receiver is enabled, and CTRLB.RXEN reads back as '1'.
This bit is not enable-protected.

Value	Description
0	The receiver is disabled.
1	The receiver is enabled or it is enabled when SPI is enabled.

Bits 15:14 – AMODE[1:0] Address Mode

These bits set the Client Addressing mode when the frame format (CTRLA.FORM) with address is used. They are unused in Host mode.
These bits are not synchronized.

AMODE[1:0]	Name	Description
0x0	MASK	ADDRMASK is used as a mask to the ADDR register
0x1	2_ADDRS	The client responds to the two unique addresses in ADDR and ADDRMASK
0x2	RANGE	The client responds to the range of addresses between and including ADDR and ADDRMASK. ADDR is the upper limit
0x3	—	Reserved

© 2025 Microchip Technology Inc. and its subsidiaries

Bit 13 – MSSEN Host SPI Select Enable
This bit enables hardware SPI Select ($\overline{SS}$) control.
This bit is not synchronized.

Value	Description
0	Hardware $\overline{SS}$ control is disabled.
1	Hardware $\overline{SS}$ control is enabled.

Bit 9 – SSDE SPI Select Low Detect Enable
This bit enables wake-up when the SPI Select ($\overline{SS}$) pin transitions from high-to-low.
This bit is not synchronized.

Value	Description
0	$\overline{SS}$ low detector is disabled.
1	$\overline{SS}$ low detector is enabled.

Bit 6 – PLOADEN Client Data Preload Enable
Setting this bit enables preloading of the Client Shift register when there is no transfer in progress. If the $\overline{SS}$ line is high when DATA is written, it is transferred immediately to the Shift register.
This bit is not synchronized.

Bits 2:0 – CHSIZE[2:0] Character Size
These bits are not synchronized.

CHSIZE[2:0]	Name	Description
0x0	8BIT	8 bits
0x1	9BIT	9 bits
0x2-0x7	—	Reserved

© 2025 Microchip Technology Inc. and its subsidiaries

33.8.3. Control C

Name: CTRLC
Offset: 0x08
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
								DATA32B
Access								R/W
Reset								0

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
			ICSPACE[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 24 – DATA32B Data 32 Bit

This bit enables the 32-bit extension for read and write transactions to the DATA register. When disabled, access correlates to CTRLB.CHSIZE.

Value	Description
0	Transactions from and to the DATA register correlate to CTRLB.CHSIZE
1	Transactions from and to the DATA register are 32-bit

Bits 5:0 – ICSPACE[5:0] Inter-Character Spacing

When non-zero, CTRLC.ICSPACE selects the minimum number of baud cycles the SCK line will not toggle between characters.

Value	Description
0x00	Inter-Character Spacing is disabled
0x01-0x3F	The minimum Inter-Character Spacing

© 2025 Microchip Technology Inc. and its subsidiaries

33.8.4. Baud Rate

Name: BAUD
Offset: 0x0C
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

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

Bits 7:0 – BAUD[7:0] Baud Register
These bits control the clock generation, see *Clock Generation – Baud-Rate Generator* from Related Links.

Related Links
Clock Generation – Baud-Rate Generator

© 2025 Microchip Technology Inc. and its subsidiaries

33.8.5. Interrupt Enable Clear

Name: INTENCLR
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection

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

Bit	7	6	5	4	3	2	1	0
	ERROR				SSL	RXC	TXC	DRE
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit 7 – ERROR Error Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the Error Interrupt Enable bit, which disables the Error interrupt.

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

Bit 3 – SSL SPI Select Low Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the SPI Select Low Interrupt Enable bit, which disables the SPI Select Low interrupt.

Value	Description
0	SPI Select Low interrupt is disabled.
1	SPI Select Low interrupt is enabled.

Bit 2 – RXC Receive Complete Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the Receive Complete Interrupt Enable bit, which disables the Receive Complete interrupt.

Value	Description
0	Receive Complete interrupt is disabled.
1	Receive Complete interrupt is enabled.

Bit 1 – TXC Transmit Complete Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the Transmit Complete Interrupt Enable bit, which disables the Transmit Complete interrupt.

Value	Description
0	Transmit Complete interrupt is disabled.
1	Transmit Complete interrupt is enabled.

Bit 0 – DRE Data Register Empty Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the Data Register Empty Interrupt Enable bit, which disables the Data Register Empty interrupt.

© 2025 Microchip Technology Inc. and its subsidiaries

Value	Description
0	Data Register Empty interrupt is disabled.
1	Data Register Empty interrupt is enabled.

© 2025 Microchip Technology Inc. and its subsidiaries

33.8.6. Interrupt Enable Set

Name: INTENSET
Offset: 0x16
Reset: 0x00
Property: PAC Write-Protection

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

Bit	7	6	5	4	3	2	1	0
	ERROR				SSL	RXC	TXC	DRE
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit 7 – ERROR Error Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit sets the Error Interrupt Enable bit, which enables the Error interrupt.

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

Bit 3 – SSL SPI Select Low Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit sets the SPI Select Low Interrupt Enable bit, which enables the SPI Select Low interrupt.

Value	Description
0	SPI Select Low interrupt is disabled.
1	SPI Select Low interrupt is enabled.

Bit 2 – RXC Receive Complete Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit sets the Receive Complete Interrupt Enable bit, which enables the Receive Complete interrupt.

Value	Description
0	Receive Complete interrupt is disabled.
1	Receive Complete interrupt is enabled.

Bit 1 – TXC Transmit Complete Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit sets the Transmit Complete Interrupt Enable bit, which enables the Transmit Complete interrupt.

Value	Description
0	Transmit Complete interrupt is disabled.
1	Transmit Complete interrupt is enabled.

Bit 0 – DRE Data Register Empty Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit sets the Data Register Empty Interrupt Enable bit, which enables the Data Register Empty interrupt.

© 2025 Microchip Technology Inc. and its subsidiaries

Value	Description
0	Data Register Empty interrupt is disabled.
1	Data Register Empty interrupt is enabled.

© 2025 Microchip Technology Inc. and its subsidiaries

33.8.7. Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x18
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	ERROR				SSL	RXC	TXC	DRE
Access	R/W				R/W	R	R/W	R
Reset	0				0	0	0	0

Bit 7 – ERROR Error
This flag is cleared by writing '1' to it.
This bit is set when any error is detected. Errors that set this flag have corresponding Status flags in the STATUS register. The BUFOVF error and the LENERR error set this Interrupt flag.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the flag.

Bit 3 – SSL SPI Select Low
This flag is cleared by writing '1' to it.
This bit is set when a high-to-low transition is detected on the $\overline{SS}$ pin in Client mode and SPI Select Low Detect (CTRLB.SSDE) is enabled.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the flag.

Bit 2 – RXC Receive Complete
This flag is cleared by reading the Data (DATA) register or by disabling the receiver.
This flag is set when there are unread data in the receive buffer. If address matching is enabled, the first data received in a transaction are an address.
Writing '0' to this bit has no effect.
Writing '1' to this bit has no effect.

Bit 1 – TXC Transmit Complete
This flag is cleared by writing '1' to it or by writing new data to DATA.
In Host mode, this flag is set when the data are shifted out and there are no new data in DATA.
In Client mode, this flag is set when the $\overline{SS}$ pin is pulled high. If address matching is enabled, this flag is only set if the transaction was initiated with an address match.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the flag.

Bit 0 – DRE Data Register Empty
This flag is cleared by writing new data to DATA.
This flag is set when DATA is empty and ready for new data to transmit.
Writing '0' to this bit has no effect.
Writing '1' to this bit has no effect.

© 2025 Microchip Technology Inc. and its subsidiaries

33.8.8. Status

Name: STATUS
Offset: 0x1A
Reset: 0x0000
Property: –

Bit	15	14	13	12	11	10	9	8
					LENERR			
Access					R/W			
Reset					0			

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

Bit 11 – LENERR Transaction Length Error
This bit is set in the Client mode when the length counter is enabled (LENGTH.LENEN = 1) and the transfer length while $\overline{SS}$ is low is not a multiple of LENGTH.LEN.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears it.

Value	Description
0	No Length Error has occurred.
1	A Length Error has occurred.

Bit 2 – BUFOVF Buffer Overflow
Reading this bit before reading DATA indicates 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.
When set, the corresponding RxDATA is '0'.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears it.

Value	Description
0	No Buffer Overflow has occurred.
1	A Buffer Overflow has occurred.

© 2025 Microchip Technology Inc. and its subsidiaries

33.8.9. Synchronization Busy

Name: SYNCBUSY
Offset: 0x1C
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				LENGTH		CTRLB	ENABLE	SWRST
Access				R		R	R	R
Reset				0		0	0	0

Bit 4 – LENGTH LENGTH Synchronization Busy

Writing to the LENGTH register requires synchronization. When writing to LENGTH, SYNCBUSY.LENGTH is set until synchronization is complete. If the LENGTH register is written to while SYNCBUSY.LENGTH is asserted, an APB error is generated.

Note: In the Client mode, the clock is only running during data transfer, therefore, SYNCBUSY.LENGTH remains asserted until the next data transfer begins.

Value	Description
0	LENGTH synchronization is not busy.
1	LENGTH synchronization is busy.

Bit 2 – CTRLB CTRLB Synchronization Busy

Writing to the CTRLB when the SERCOM is enabled requires synchronization. Ongoing synchronization is indicated by SYNCBUSY.CTRLB = 1 until synchronization is complete. If CTRLB is written while SYNCBUSY.CTRLB = 1, an APB error is generated.

Value	Description
0	CTRLB synchronization is not busy.
1	CTRLB synchronization is busy.

Bit 1 – ENABLE SERCOM Enable Synchronization Busy

Enabling and disabling the SERCOM (CTRLA.ENABLE) requires synchronization. Ongoing synchronization is indicated by SYNCBUSY.ENABLE = 1 until synchronization is complete.

Value	Description
0	Enable synchronization is not busy.
1	Enable synchronization is busy.

© 2025 Microchip Technology Inc. and its subsidiaries

Bit 0 – SWRST Software Reset Synchronization Busy

Resetting the SERCOM (CTRLA.SWRST) requires synchronization. Ongoing synchronization is indicated by SYNCBUSY.SWRST = 1 until synchronization is complete.

Value	Description
0	SWRST synchronization is not busy.
1	SWRST synchronization is busy.

© 2025 Microchip Technology Inc. and its subsidiaries

33.8.10. Length

Name: LENGTH
Offset: 0x22
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
								LENEN
Access								R/W
Reset								0

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

Bit 8 – LENEN Data Length Enable
In 32-bit Extension mode, this bit field enables the length counter.

Value	Description
0	Length counter disabled
1	Length counter enabled

Bits 7:0 – LEN[7:0] Data Length
In 32-bit Extension mode, this bit field configures the data length, after which, the flags INTFLAG.RCX or INTFLAG.DRE are raised.

Value	Description
0x00	Reserved if LENEN = 0x1
0x01-0xFF	Data Length

© 2025 Microchip Technology Inc. and its subsidiaries

33.8.11. Address

Name: ADDR
Offset: 0x24
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

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

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

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

Bits 23:16 – ADDRMASK[7:0] Address Mask
These bits hold the address mask when the transaction format with address is used (CTRLA.FORM, CTRLB.AMODE).

Bits 7:0 – ADDR[7:0] Address
These bits hold the address when the transaction format with address is used (CTRLA.FORM, CTRLB.AMODE).

© 2025 Microchip Technology Inc. and its subsidiaries

33.8.12. Data

Name: DATA
Offset: 0x28
Reset: 0x0000
Property: –

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

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

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

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

Bits 31:0 – DATA[31:0] Data

Reading these bits returns the contents of the receive data buffer. The register must be read only when the Receive Complete Interrupt Flag bit in the Interrupt Flag Status and Clear register (INTFLAG.RXC) is set.

Writing these bits writes the transmit data buffer. This register must be written only when the Data Register Empty Interrupt Flag bit in the Interrupt Flag Status and Clear register (INTFLAG.DRE) is set. Reads and writes are 32-bit or CTLB.CHSIZE based on the CTRLC.DATA32B setting.

© 2025 Microchip Technology Inc. and its subsidiaries

33.8.13. Debug Control

Name: DBGCTRL
Offset: 0x30
Reset: 0x00
Property: PAC Write-Protection

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

Bit 0 – DBGSTOP Debug Stop Mode
This bit controls the functionality when the CPU is halted by an external debugger.

Value	Description
0	The baud-rate generator continues normal operation when the CPU is halted by an external debugger.
1	The baud-rate generator is halted when the CPU is halted by an external debugger.

© 2025 Microchip Technology Inc. and its subsidiaries

34. SERCOM Inter-Integrated Circuit (SERCOM I²C)

34.1. Overview

The Inter-Integrated Circuit (I²C) interface is one of the available modes in the serial communication interface (SERCOM).

The I²C interface uses the SERCOM transmitter and receiver configured as illustrated in Figure 34-1. Labels in capital letters are registers accessible by the CPU, while lowercase labels are internal to the SERCOM.

A SERCOM instance can be configured to be either an I²C host or an I²C client. Both host and client have an interface containing a shift register, a transmit buffer and a receive buffer. In addition, the I²C host uses the SERCOM baud-rate generator, while the I²C client uses the SERCOM address match logic.

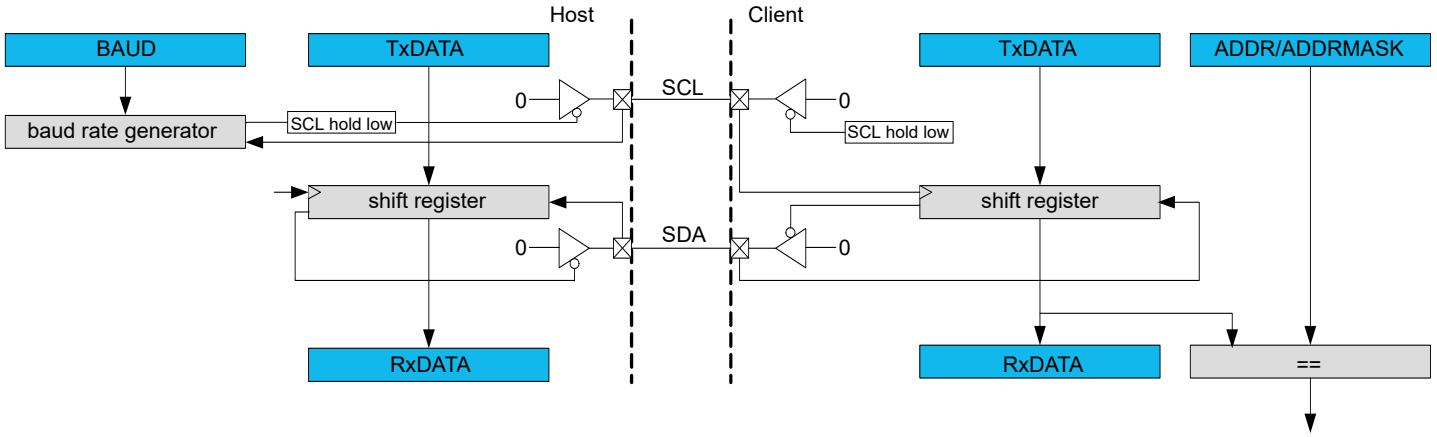
34.2. Features

- Host or Client Operation
- Can be used with DMA
- Philips I²C Compatible
- SMBus Compatible
- PMBus™ Compatible
- Support 100 kHz, 400 kHz and 1 MHz at Low System Clock Frequencies
- 32-Bit Data Extension for better System Bus Utilization
- Four-Wire Operation Supported
- Physical Interface Includes:
  - Slew-rate limited outputs
  - Filtered inputs
- Client Operation:
  - Operation in all Sleep modes
  - Wake-up on address match
  - 7-bit and 10-bit Address match in hardware for:
    - Unique address and/or 7-bit general call address
    - Address range
    - Two unique addresses can be used with DMA

© 2025 Microchip Technology Inc. and its subsidiaries

34.3. Block Diagram

Figure 34-1. I²C Single-Host Single-Client Interconnection



34.4. Signal Description

Table 34-1. Signal Description

Signal Name	Type	Description
PAD[0]	Digital I/O	SDA
PAD[1]	Digital I/O	SCL
PAD[2]	Digital I/O	SDA_OUT (Four-wire operation)
PAD[3]	Digital I/O	SCL_OUT (Four-wire operation)

I²C pins on SERCOM are fixed pins and not configurable through PPS functionality. See *I/O Ports and Peripheral Pin Select (PPS)* from Related Links.

Related Links

I/O Ports and Peripheral Pin Select (PPS)

34.5. Product Dependencies

The following sections describe how the other parts of the system must be configured correctly to use this peripheral.

34.5.1. I/O Lines

To use the SERCOM's I/O lines, the I/O pins must be configured as direct using the System Configuration registers. (See *System Configuration and Register Locking (CFG)* from Related Links.) I²C does not operate through PPS. (See the DEVCFG1 configuration bits SCOM_HSEN[x] in Configuration Bits Fuses and also the CFGCON1 SCOMn_HSEN in CFGCON1(L) register.)

Notes:

1. SERCOM6 has only I²C functionality and, therefore, the direct configuration using System configuration register is not required.
2. SERCOM Instances.

© 2025 Microchip Technology Inc. and its subsidiaries

SERCOM	FIFO Depth	SPI	USART	I2C/ SMBus	1.8V I2C/ SMBus	Full Speed	48-Pin	64-Pin	132-Pin
SERCOM0(through PPS)	4	Yes	Yes	—	—	—	Yes	Yes	Yes
SERCOM0(Fixed pins)		Yes	Yes	Yes	—	Yes	Yes	Yes	Yes
SERCOM1(through PPS)	2	Yes	Yes	—	—	—	Yes	Yes	Yes
SERCOM1(Fixed pins)		Yes	Yes	Yes	—	Yes	Yes	Yes	Yes
SERCOM2(through PPS)	2	Yes	Yes	—	—	—	Yes	Yes	Yes
SERCOM2(Fixed pins)		Yes	Yes	Yes	Yes[a]	Yes	—	Yes	Yes
SERCOM3(through PPS)	2	Yes	Yes	—	—	—	Yes	Yes	Yes
SERCOM4(through PPS)	2	Yes	Yes	—	—	—	—	Yes	Yes
SERCOM5(through PPS)	2	Yes	Yes	—	—	—	—	Yes	Yes
SERCOM6(Fixed pins)	2	—	—	Yes	—	—	Yes	Yes	Yes

Note:

1. When used with external (off-chip) 1.8V pull-ups

When the SERCOM is used in I^2C mode, the SERCOM controls the direction and value of the I/O pins. In I^2C mode, pull-up resistors are disabled. External pull-up resistors are required for proper function.

Related Links

System Configuration and Register Locking (CFG)

34.5.2. Power Management

This peripheral can continue to operate in any Sleep mode where its source clock is running. The interrupts can wake-up the device from Sleep modes.

34.5.3. Clocks

Two generic clocks are used by SERCOM, GCLK_SERCOMx_CORE and GCLK_SERCOMx_SLOW. The core clock (GCLK_SERCOMx_CORE) can clock the I^2C when working as a host. The slow clock (GCLK_SERCOMx_SLOW) is required only for certain functions, for example, for SMBus timing 32KHz_LPCLK must be configured as this is the source for GCLK_SERCOMx_SLOW clock. These two clocks must be configured and enabled in the CRU registers before using the I^2C.

These generic clocks are asynchronous to the bus clock (PBx_CLK). Due to this asynchronicity, writes to certain registers requires synchronization between the clock domains.

34.5.4. DMA

The DMA request lines are connected to the DMA Controller (DMAC). To use DMA requests with this peripheral, the DMAC must be configured first (see *Direct Memory Access Controller (DMAC)* from Related Links).

Related Links

Direct Memory Access Controller (DMAC)

34.5.5. Interrupts

The interrupt request line is connected to the Interrupt Controller. In order to use interrupt requests of this peripheral, the Interrupt Controller (NVIC) must be configured first. See *Nested Vector Interrupt Controller (NVIC)* from Related Links.

Related Links

Nested Vector Interrupt Controller (NVIC)

© 2025 Microchip Technology Inc. and its subsidiaries

34.5.6. Events

Not applicable.

34.5.7. Debug Operation

When the CPU is halted in Debug mode, this peripheral continues normal operation. If the peripheral is configured to require periodical service by the CPU through interrupts or similar, improper operation or data loss may result during debugging. This peripheral can be forced to halt operation during debugging. See the *DBGCTRL* register from Related Links.

Related Links

DBGCTRL

Debug Control

34.5.8. Register Access Protection

Registers with write access can be write protected optionally by the PAC.

PAC write protection is not available for the following registers:

- Interrupt Flag Clear and Status register (INTFLAG)
- Status register (STATUS)
- Data register (DATA)
- Address register (ADDR) in Host mode

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.

34.5.9. Analog Connections

Not applicable.

34.6. Functional Description

34.6.1. Principle of Operation

The I^2C interface uses two physical lines for communication:

- Serial Data Line (SDA) for data transfer
- Serial Clock Line (SCL) for the bus clock

A transaction starts with the I^2C host sending the Start condition, followed by a 7-bit address and a direction bit (read or write to/from the client).

The addressed I^2C client will then Acknowledge (ACK) the address, and data packet transactions can begin. Every 9-bit data packet consists of 8 data bits followed by a one-bit reply indicating whether the data was acknowledged or not.

If a data packet is Not Acknowledged (NACK), whether by the I^2C client or host, the I^2C host takes action by either terminating the transaction by sending the Stop condition, or by sending a repeated start to transfer more data.

The figure below illustrates the possible transaction formats and Transaction Diagram Symbols explains the transaction symbols. These symbols will be used in the following descriptions.

© 2025 Microchip Technology Inc. and its subsidiaries

Figure 34-2. Transaction Diagram Symbols

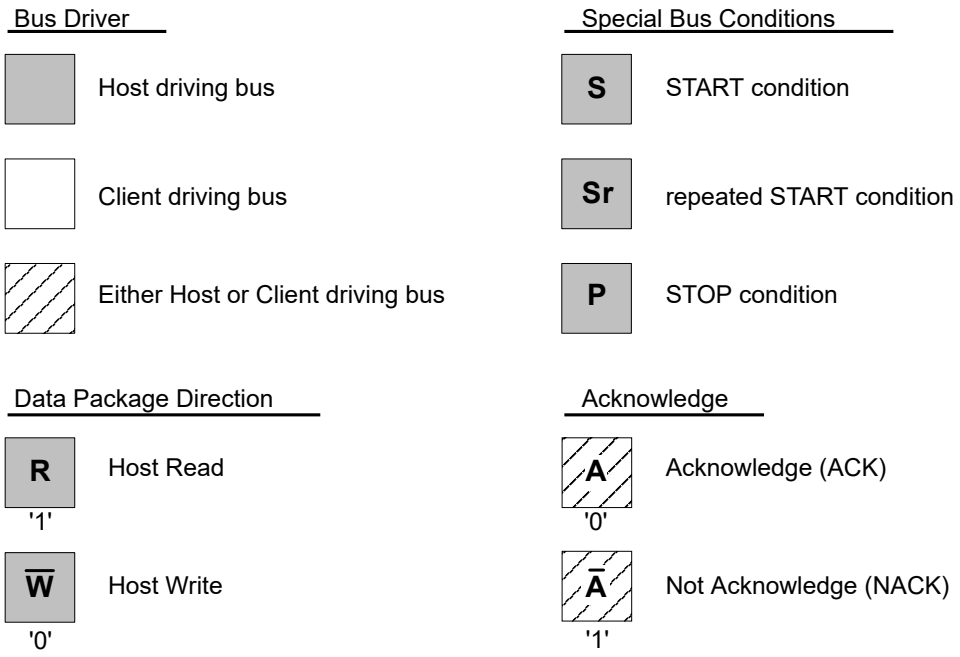
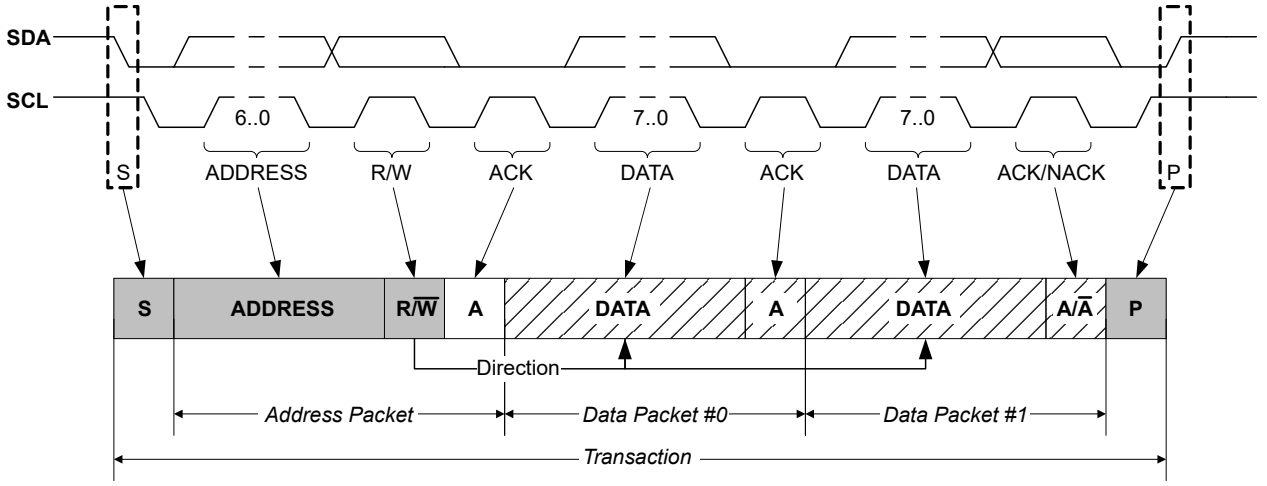


Figure 34-3. Basic I²C Transaction Diagram



34.6.2. Basic Operation

34.6.2.1. Initialization

The following registers are enable-protected, meaning they can be written only when the I²C interface is disabled (CTRLA.ENABLE = 0):

- Control A register (CTRLA), except Enable (CTRLA.ENABLE) and Software Reset (CTRLA.SWRST) bits
- Control B register (CTRLB), except Acknowledge Action (CTRLB.ACKACT) and Command (CTRLB.CMD) bits
- Baud register (BAUD)
- Address register (ADDR) in client operation

When the I²C is enabled or is being enabled (CTRLA.ENABLE = 1), writing to these registers is discarded. If the I²C is being disabled, writing to these registers is completed after the disabling.

© 2025 Microchip Technology Inc. and its subsidiaries

Enable protection is denoted by the Enable-Protection property in the register description.

Before the I²C is enabled, it must be configured as outlined by the following steps:

1. Select I²C Host or Client mode by writing 0x4 (Client mode) or 0x5 (Host mode) to the Operating Mode bits in the CTRLA register (CTRLA.MODE).
2. If desired, select the SDA Hold Time in the CFGCON1 register (CFGCON1.I2CDSELx).
3. In Client mode, the minimum client setup time for the SDA can be selected in the SDA Setup Time bit group in the Control C register (CTRLC.SDASETUP).
4. If desired, enable smart operation by setting the Smart Mode Enable bit in the CTRLB register (CTRLB.SMEN).
5. If desired, enable SCL low time-out by setting the SCL Low Time-Out bit in the Control A register (CTRLA.LOWTOUT).
6. In Host mode:
   a. Select the inactive bus time-out in the Inactive Time-Out bit group in the CTRLA register (CTRLA.INACTOUT).
   b. Write the Baud Rate register (BAUD) to generate the desired baud rate.

   In Client mode:
   a. Configure the address match configuration by writing the Address mode value in the CTRLB register (CTRLB.AMODE).
   b. Set the Address and Address Mask value in the Address register (ADDR.ADDR and ADDR.ADDRMASK) according to the address configuration.

34.6.2.2. Enabling, Disabling and Resetting

This peripheral is enabled by writing '1' to the Enable bit in the Control A register (CTRLA.ENABLE) and disabled by writing '0' to it.

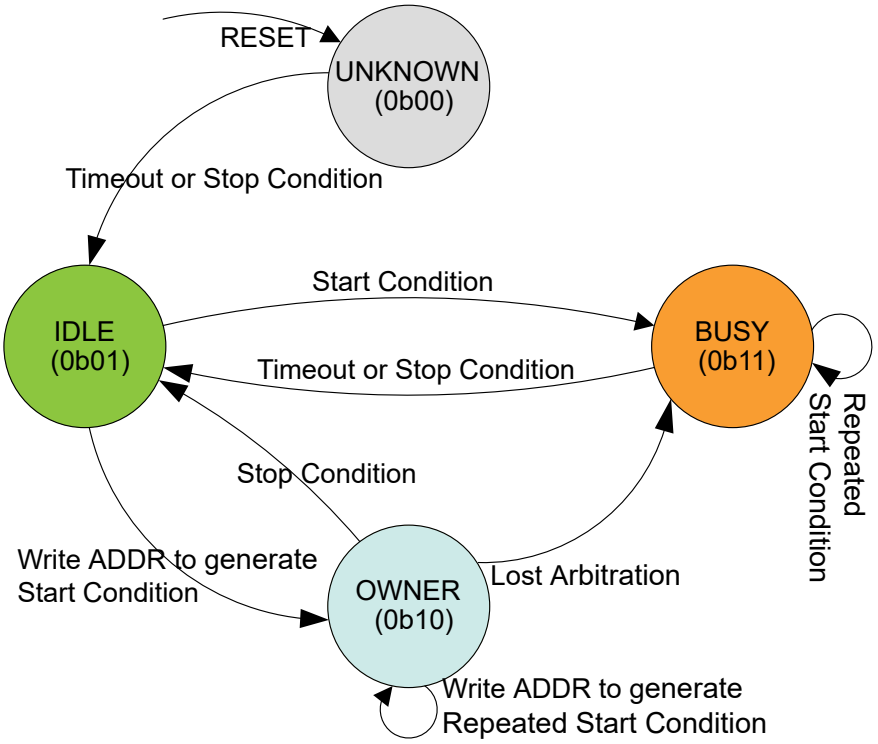
Writing '1' to the Software Reset bit in the Control A register (CTRLA.SWRST) resets all registers of this peripheral to their initial states, except the DBGCTRL register, and the peripheral is disabled.

34.6.2.3. I²C Bus State Logic

The Bus state logic includes several logic blocks that continuously monitor the activity on the I²C bus lines in all sleep modes with running GCLK_SERCOMx_CORE clocks. The start and stop detectors and the bit counter are all essential in the process of determining the current Bus state. The following figure illustrates how the Bus state is determined. Software can get the current Bus state by reading the Host Bus State bits in the Status register (STATUS.BUSSTATE). The value of STATUS.BUSSTATE in the following figure is shown in binary.

© 2025 Microchip Technology Inc. and its subsidiaries

Figure 34-4. Bus State Diagram



The Bus state machine is active when the I²C host is enabled.

After the I²C host is enabled, the Bus state is UNKNOWN (0b00). From the UNKNOWN state, the bus will transition to IDLE (0b01) by either:

- Forcing by writing 0b01 to STATUS.BUSSTATE
- A Stop condition is detected on the bus
- If the inactive bus time-out is configured for SMBus compatibility (CTRLA.INACTOUT) and a time-out occurs

Note: When a known Bus state is established, the Bus state logic will not re-enter the UNKNOWN state.

When the bus is IDLE, it is ready for a new transaction. If a Start condition is issued on the bus by another I²C host in a multi-host setup, the bus becomes BUSY (0b11). The bus re-enters IDLE either when a Stop condition is detected or when a time-out occurs (inactive bus time-out needs to be configured).

If a Start condition is generated internally by writing the Address bit group in the Address register (ADDR.ADDR) while IDLE, the OWNER state (0b10) is entered. If the complete transaction was performed without interference, in other words, arbitration was not lost, the I²C host can issue a Stop condition, which changes the Bus state back to IDLE.

However, if a packet collision is detected while in OWNER state, the arbitration is assumed lost and the Bus state becomes BUSY until a Stop condition is detected. A repeated Start condition changes the Bus state only if arbitration is lost while issuing a repeated start.

Note: Violating the protocol may cause the I²C to hang. If this happens, it is possible to recover from this state by a software Reset (CTRLA.SWRST = 1). See *CTRLA* (Client) or *CTRLA* (Host) register from Related Links.

© 2025 Microchip Technology Inc. and its subsidiaries

Related Links

CTRLA

Control A

CTRLA

Control A

34.6.2.4. I²C Host Operation

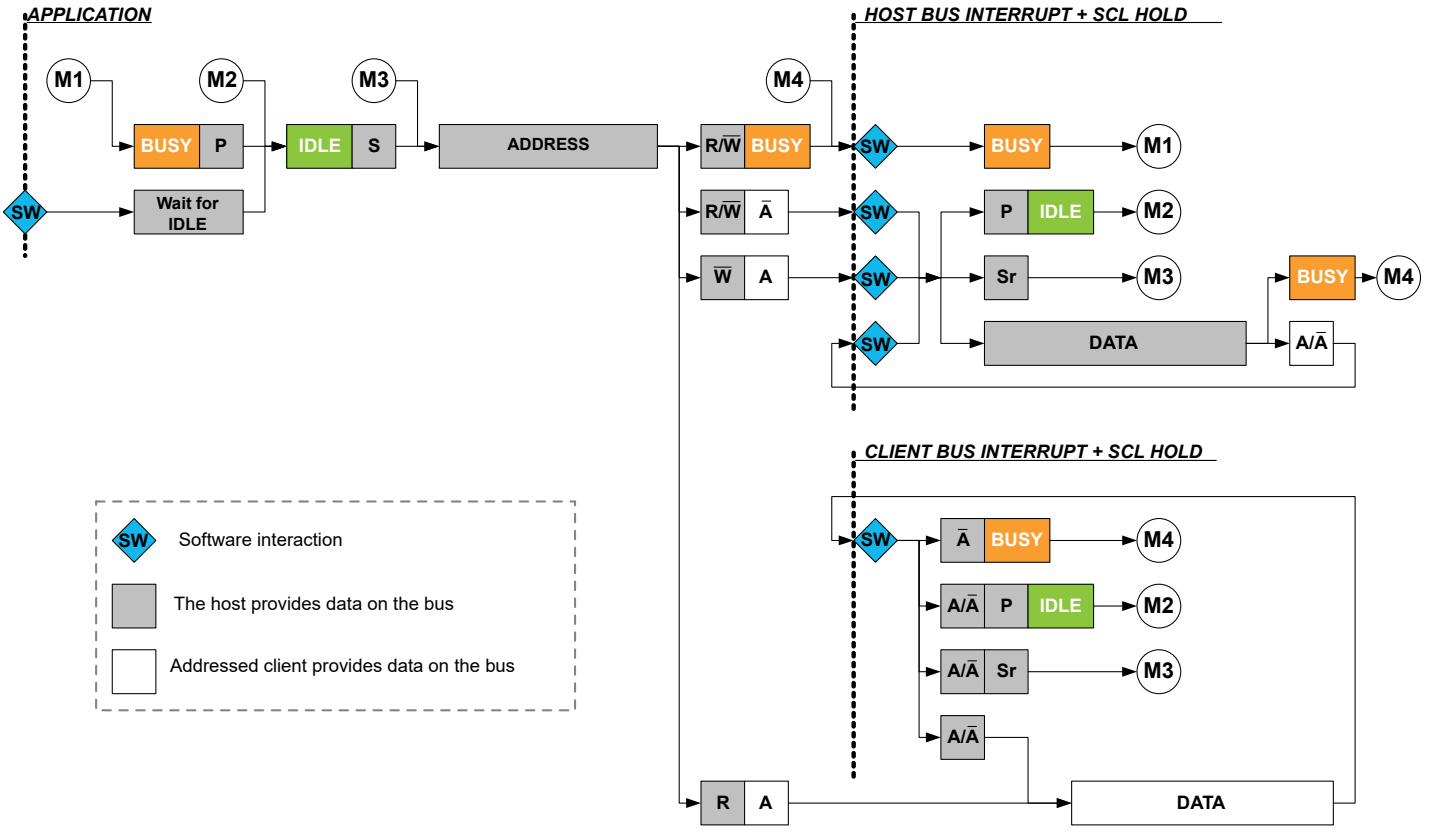
The I²C host is byte-oriented and interrupt-based. The number of interrupts generated is kept at a minimum by automatic handling of most incidents. The software driver complexity and code size are reduced by auto-triggering of operations and a Special Smart mode, which can be enabled by the Smart Mode Enable bit in the Control A register (CTRLA.SMEN).

The I²C host has two interrupt strategies.

When SCL Stretch Mode (CTRLA.SCLSM) is '0', SCL is stretched before or after the Acknowledge bit . In this mode, the I²C host operates according to Figure 34-5. The circles labeled M*n* (M1, M2..) indicate the nodes the bus logic can jump to, based on software or hardware interaction.

This diagram is used as a reference for the description of the I²C host operation throughout the document.

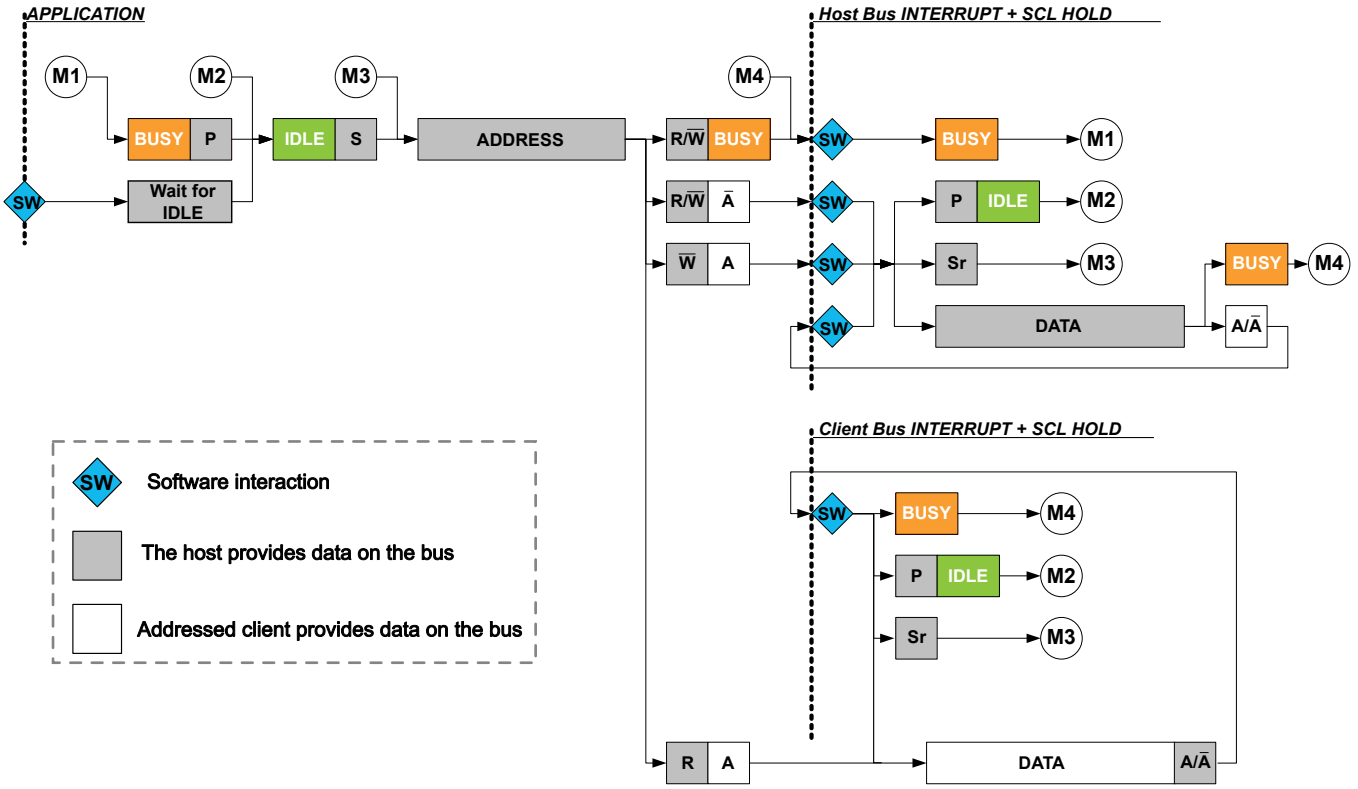
Figure 34-5. I²C Host Behavioral Diagram (SCLSM = 0)



In the second strategy (CTRLA.SCLSM = 1), interrupts only occur after the ACK bit, as illustrated in Figure 34-6. This strategy can be used when it is not necessary to check DATA before acknowledging.

© 2025 Microchip Technology Inc. and its subsidiaries

Figure 34-6. I²C Host Behavioral Diagram (SCLSM = 1)



34.6.2.4.1. Host Clock Generation

The SERCOM peripheral supports several I²C bidirectional modes:

- Standard mode (*Sm*) up to 100 kHz
- Fast mode (*Fm*) up to 400 kHz
- Fast mode Plus (*Fm+*) up to 1 MHz

The Host clock configuration for *Sm, Fm* and *Fm+* are described in *Clock Generation (Standard-Mode, Fast-Mode and Fast-Mode Plus)* as follows.

Related Links

Electrical Characteristics

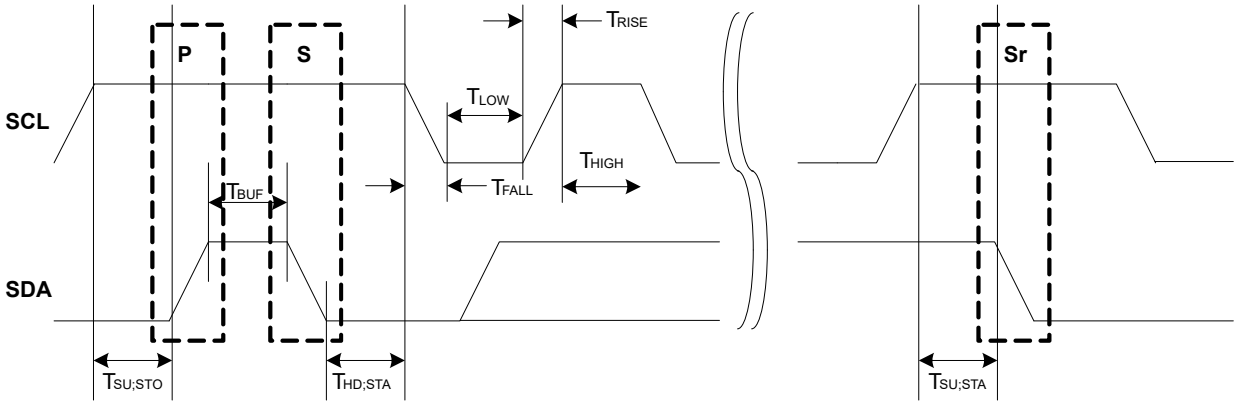
Clock Generation (Standard-Mode, Fast-Mode and Fast-Mode Plus)

In I²C *Sm, Fm* and *Fm+* modes, the Host clock (SCL) frequency is determined as described in this section:

The low (T_{LOW}) and high (T_{HIGH}) times are determined by the Baud Rate register (BAUD), while the rise (T_{RISE}) and fall (T_{FALL}) times are determined by the bus topology. Because of the wired-AND logic of the bus, T_{FALL} is considered as part of T_{LOW}. Likewise, T_{RISE} is in a state between T_{LOW} and T_{HIGH} until a high state is detected.

© 2025 Microchip Technology Inc. and its subsidiaries

Figure 34-7. SCL Timing



The following parameters are timed using the SCL low time period T_{LOW}. This comes from the Host Baud Rate Low bit group in the Baud Rate register (BAUD.BAUDLOW) when BAUD.BAUDLOW = 0 or the Host Baud Rate bit group in the Baud Rate register (BAUD.BAUD) determines it.

- T_{LOW} – Low period of SCL clock
- $T_{SU;STO}$ – Setup time for stop condition
- T_{BUF} – Bus free time between stop and start conditions
- $T_{HD;STA}$ – Hold time (repeated) start condition
- $T_{SU;STA}$ – Setup time for repeated start condition
- T_{HIGH} is timed using the SCL high time count from BAUD.BAUD
- T_{RISE} is determined by the bus impedance; for internal pull-ups
- T_{FALL} is determined by the open-drain current limit and bus impedance; can typically be regarded as zero
  See *Electrical Characteristics* from Related Links.

The SCL frequency is given by:

$$f_{SCL} = \frac{1}{T_{LOW} + T_{HIGH} + T_{RISE}}$$

When BAUD.BAUDLOW is zero, the BAUD.BAUD value is used to time both SCL high and SCL low. In this case, the following formula gives the SCL frequency:

$$f_{SCL} = \frac{f_{GCLK}}{10 + 2BAUD + f_{GCLK} \cdot T_{RISE}}$$

When BAUD.BAUDLOW is non-zero, the following formula determines the SCL frequency:

$$f_{SCL} = \frac{f_{GCLK}}{10 + BAUD + BAUDLOW + f_{GCLK} \cdot T_{RISE}}$$

The following formulas can determine the SCL T_{LOW} and T_{HIGH} times:

$$T_{LOW} = \frac{BAUDLOW + 5}{f_{GCLK}}$$

$$T_{HIGH} = \frac{BAUD + 5}{f_{GCLK}}$$

© 2025 Microchip Technology Inc. and its subsidiaries

Note: The I²C standard *Fm+* (Fast-mode plus) requires a nominal high to low SCL ratio of 1:2, and BAUD must be set accordingly. At a minimum, BAUD.BAUD and/or BAUD.BAUDLOW must be non-zero.

Start-up Timing: The minimum time between SDA transition and SCL rising edge is 6 APB cycles when the DATA register is written in smart mode. If a greater start-up time is required due to long rise times, the time between DATA write and IF clear must be controlled by software.
Note: When timing is controlled by the user, the Smart mode cannot be enabled.

34.6.2.4.2. Transmitting Address Packets

The I²C host starts a bus transaction by writing the I²C client address to ADDR.ADDR and the direction bit, as described in Principle of Operation, see *Principle of Operation* from Related Links. If the bus is busy, the I²C host will wait until the bus becomes idle before continuing the operation. When the bus is idle, the I²C host will issue a start condition on the bus. The I²C host will then transmit an address packet using the address written to ADDR.ADDR. After the address packet has been transmitted by the I²C host, one of four cases will arise according to arbitration and transfer direction.

Case 1: Arbitration lost or bus error during address packet transmission

If arbitration was lost during transmission of the address packet, the Host on Bus bit in the Interrupt Flag Status and Clear register (INTFLAG.MB) and the Arbitration Lost bit in the Status register (STATUS.ARBLOST) are both set. Serial data output to SDA is disabled, and the SCL is released, which disables clock stretching. In effect the I²C host is no longer allowed to execute any operation on the bus until the bus is idle again. A bus error will behave similarly to the Arbitration Lost condition. In this case, the MB Interrupt flag and Host Bus Error bit in the Status register (STATUS.BUSERR) are both set in addition to STATUS.ARBLOST.

The Host Received Not Acknowledge bit in the Status register (STATUS.RXNACK) will always contain the last successfully received acknowledge or not acknowledge indication.

In this case, software will typically inform the application code of the condition and then clear the Interrupt flag before exiting the interrupt routine. No other flags have to be cleared at this moment, because all flags will be cleared automatically the next time the ADDR.ADDR register is written.

Case 2: Address packet transmit complete – No ACK received

If there is no I²C client device responding to the address packet, then the INTFLAG.MB Interrupt flag and STATUS.RXNACK will be set. The clock hold is active at this point, preventing further activity on the bus.

The missing ACK response can indicate that the I²C client is busy with other tasks or sleeping. Therefore, it is not able to respond. In this event, the next step can be either issuing a Stop condition (recommended) or resending the address packet by a repeated Start condition. When using SMBus logic, the client must ACK the address. If there is no response, it means that the client is not available on the bus.

Case 3: Address packet transmit complete – Write packet, Host on Bus set

If the I²C host receives an acknowledge response from the I²C client, INTFLAG.MB will be set and STATUS.RXNACK will be cleared. The clock hold is active at this point, preventing further activity on the bus.

In this case, the software implementation becomes highly protocol dependent. Three possible actions can enable the I²C operation to continue:

- Initiate a data transmit operation by writing the data byte to be transmitted into DATA.DATA.
- Transmit a new address packet by writing ADDR.ADDR. A repeated Start condition will automatically be inserted before the address packet.
- Issue a Stop condition, consequently terminating the transaction.

© 2025 Microchip Technology Inc. and its subsidiaries

Case 4: Address packet transmit complete – Read packet, Client on Bus set

If the I²C host receives an ACK from the I²C client, the I²C host proceeds to receive the next byte of data from the I²C client. When the first data byte is received, the Client on Bus bit in the Interrupt Flag register (INTFLAG.SB) will be set and STATUS.RXNACK will be cleared. The clock hold is active at this point, preventing further activity on the bus.

In this case, the software implementation becomes highly protocol dependent. Three possible actions can enable the I²C operation to continue:

- Let the I²C host continue to read data by acknowledging the data received. ACK can be sent by software, or automatically in Smart mode.
- Transmit a new address packet.
- Terminate the transaction by issuing a Stop condition.

Note: An ACK or NACK will be automatically transmitted if Smart mode is enabled. The Acknowledge Action bit in the Control B register (CTRLB.ACKACT) determines whether ACK or NACK must be sent.

Related Links

Principle of Operation

34.6.2.4.3. Transmitting Data Packets

When an address packet with direction Host Write (see Figure 34-3) was transmitted successfully , INTFLAG.MB will be set. The I²C host will start transmitting data via the I²C bus by writing to DATA.DATA, and monitor continuously for packet collisions.

If a collision is detected, the I²C host will lose arbitration and STATUS.ARBLOST will be set. If the transmit was successful, the I²C host will receive an ACK bit from the I²C client, and STATUS.RXNACK will be cleared. INTFLAG.MB will be set in both cases, regardless of arbitration outcome.

It is recommended to read STATUS.ARBLOST and handle the arbitration lost condition in the beginning of the I²C Host on Bus interrupt. This can be done as there is no difference between handling address and data packet arbitration.

STATUS.RXNACK must be checked for each data packet transmitted before the next data packet transmission can commence. The I²C host is not allowed to continue transmitting data packets if a NACK is received from the I²C client.

34.6.2.4.4. Receiving Data Packets (SCLSM=0)

When INTFLAG.SB is set, the I²C host will already have received one data packet. The I²C host must respond by sending either an ACK or NACK. Sending a NACK may be unsuccessful when arbitration is lost during the transmission. In this case, a lost arbitration will prevent setting INTFLAG.SB. Instead, INTFLAG.MB will indicate a change in arbitration. Handling of lost arbitration is the same as for data bit transmission.

34.6.2.4.5. Receiving Data Packets (SCLSM=1)

When INTFLAG.SB is set, the I²C host will already have received one data packet and transmitted an ACK or NACK, depending on CTRLB.ACKACT. At this point, CTRLB.ACKACT must be set to the correct value for the next ACK bit, and the transaction can continue by reading DATA and issuing a command if not in the Smart mode.

34.6.2.5. I²C Client Operation

The I²C client is byte-oriented and interrupt-based. The number of interrupts generated is kept at a minimum by automatic handling of most events. The software driver complexity and code size are reduced by auto-triggering of operations and a special smart mode, which can be enabled by the Smart Mode Enable bit in the Control A register (CTRLA.SMEN).

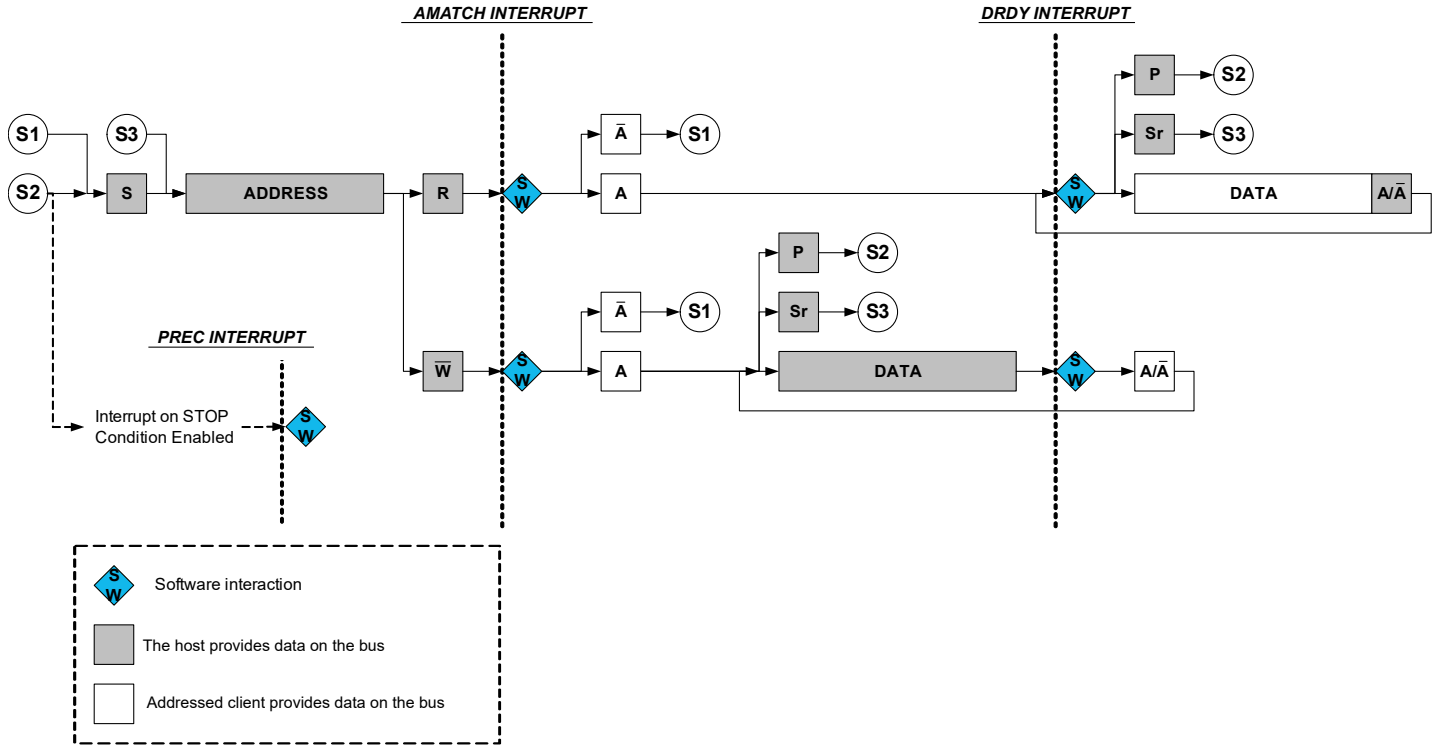
The I²C client has two interrupt strategies.

© 2025 Microchip Technology Inc. and its subsidiaries

When SCL Stretch Mode bit (CTRLA.SCLSM) is '0', SCL is stretched before or after the acknowledge bit. In this mode, the I²C client operates according to Figure 34-8. The circles labelled S*n* (S1, S2..) indicate the nodes the bus logic can jump to based on software or hardware interaction.

This diagram is used as a reference for the description of the I²C client operation throughout the document.

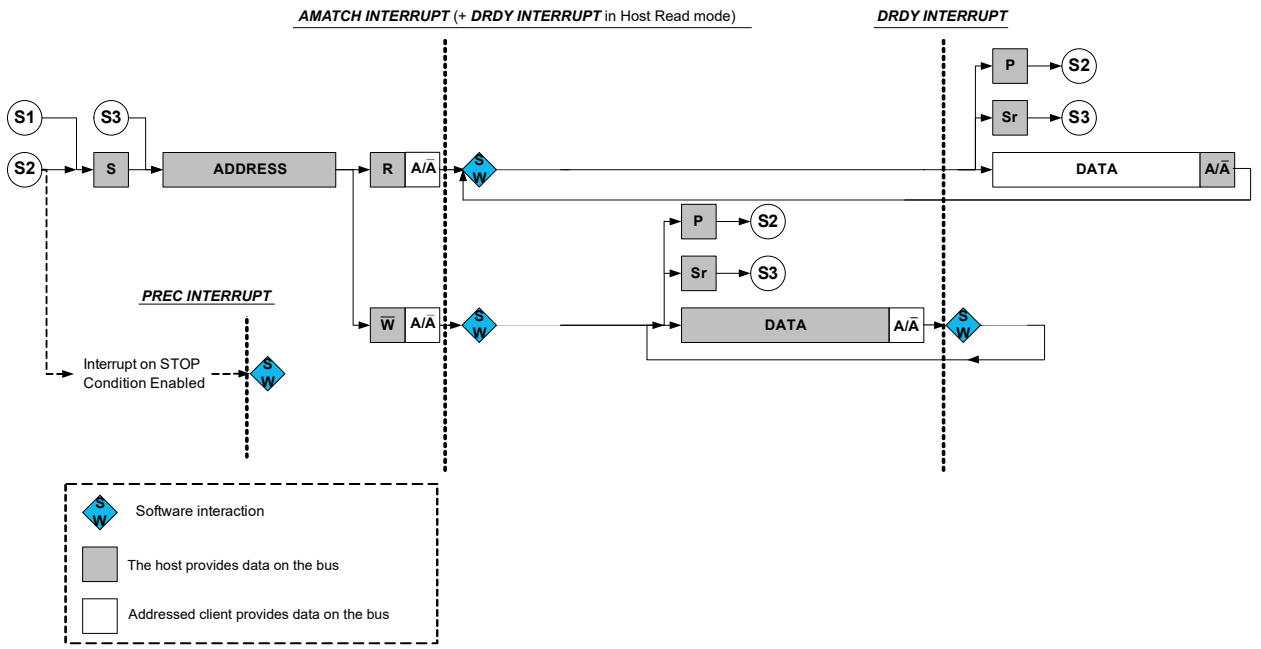
Figure 34-8. I²C Client Behavioral Diagram (SCLSM = 0)



In the second strategy (CTRLA.SCLSM = 1), interrupts only occur after the ACK bit is sent as illustrated Figure 34-9. This strategy can be used when it is not necessary to check DATA before acknowledging. For host reads, an address and data interrupt is issued simultaneously after the address acknowledge. However, for host writes, the first data interrupt is seen after the first data byte is received by the client and the acknowledge bit is sent to the host.

© 2025 Microchip Technology Inc. and its subsidiaries

Figure 34-9. I²C Client Behavioral Diagram (SCLSM = 1)



34.6.2.5.1. Receiving Address Packets (SCLSM = 0)

When CTRLA.SCLSM = 0, the I²C client stretches the SCL line according to Figure 34-8. When the I²C client is properly configured, it waits for a Start condition.

When a Start condition is detected, the successive address packet is received and checked by the address match logic. If the received address is not a match, the packet is rejected, and the I²C client waits for a new Start condition. If the received address is a match, the Address Match bit in the Interrupt Flag register (INTFLAG.AMATCH) is set.

SCL is stretched until the I²C client clears INTFLAG.AMATCH. As the I²C client 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 is 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 client 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 is received from the I²C host, one of two cases arises based on transfer direction.

Case 1: Address packet accepted – Read flag set

The STATUS.DIR bit is '1', indicating an I²C host read operation. The SCL line is forced low, stretching the bus clock. If an ACK is sent, I²C client hardware will set the Data Ready bit in the Interrupt Flag register (INTFLAG.DRDY), indicating data are needed for transmit. If a NACK is sent, the I²C client will wait for a new Start condition and address match.

Typically, software immediately acknowledges the address packet by sending an ACK/NACK bit. The I²C client Command bit field in the Control B register (CTRLB.CMD) can be written to 0x3 for both read and write operations, as the command execution is dependent on the STATUS.DIR bit. Writing '1' to INTFLAG.AMATCH will also cause an ACK/NACK to be sent corresponding to the CTRLB.ACKACT bit.

© 2025 Microchip Technology Inc. and its subsidiaries

Case 2: Address packet accepted – Write flag set

The STATUS.DIR bit is cleared, indicating an I²C host write operation. The SCL line is forced low, stretching the bus clock. If an ACK is sent, the I²C client waits for data to be received. Data, repeated start or stop can be received.

If a NACK is sent, the I²C client waits for a new Start condition and address match. Typically, software immediately acknowledges the address packet by sending an ACK/NACK. The I²C client command CTRLB.CMD = 3 can be used for both read and write operations, as the command execution is dependent on STATUS.DIR.

Writing '1' to INTFLAG.AMATCH also causes an ACK/NACK to be sent corresponding to the CTRLB.ACKACT bit.

34.6.2.5.2. Receiving Address Packets (SCLSM = 1)

When SCLSM = 1, the I²C client stretches the SCL line only after an ACK as illustrated in Figure 34-9. When the I²C client is properly configured, it waits for a Start condition to be detected.

When a Start condition is detected, the successive address packet is received and checked by the address match logic.

If the received address is not a match, the packet is rejected and the I²C client waits for a new Start condition.

If the address matches, the acknowledge action, as configured by the Acknowledge Action bit Control B register (CTRLB.ACKACT), is sent and the Address Match bit in the Interrupt Flag register (INTFLAG.AMATCH) is set. SCL is stretched until the I²C client clears INTFLAG.AMATCH. As the I²C client holds the clock by forcing SCL low, the software is given unlimited time to respond to the address.

The direction of a transaction is determined by reading the Read/Write Direction bit in the Status register (STATUS.DIR). This bit is updated only when a valid address packet is received.

If the Transmit Collision bit in the Status register (STATUS.COLL) is set, the last packet addressed to the I²C client had a packet collision. A collision causes the SDA and SCL lines to be released without any notification to software. The next AMATCH interrupt is, therefore, the first indication of the previous packet's collision. Collisions are intended to follow the SMBus Address Resolution Protocol (*ARP*).

After the address packet is received from the I²C host, INTFLAG.AMATCH can be set to '1' to clear it.

34.6.2.5.3. Receiving and Transmitting Data Packets

After the I²C client receives an address packet, it responds according to the direction either by waiting for the data packet to be received or by starting to send a data packet by writing to DATA.DATA. When a data packet is received or sent, INTFLAG.DRDY is set. After receiving data, the I²C client sends an acknowledge according to CTRLB.ACKACT.

Case 1: Data received

INTFLAG.DRDY is set and SCL is held low, pending software interaction.

Case 2: Data sent

When a byte transmission is successfully completed, the INTFLAG.DRDY Interrupt flag is set. If NACK is received, indicated by STATUS.RXNACK = 1, the I²C client must expect a stop or a repeated start to be received. The I²C client must release the data line to allow the I²C host to generate a stop or repeated start. Upon detecting a Stop condition, the Stop Received bit in the Interrupt Flag register (INTFLAG.PREC) is set and the I²C client returns to Idle state.

34.6.2.5.4. PMBus Group Command

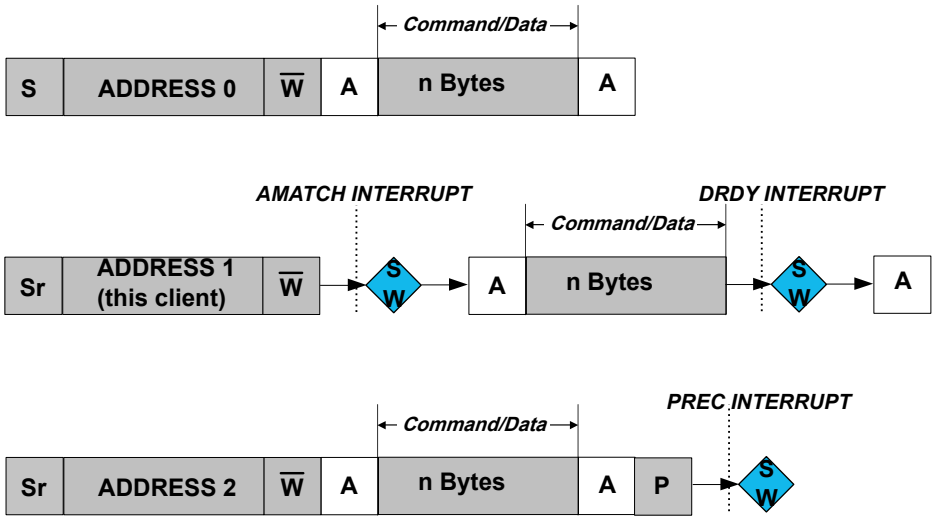
When the PMBus Group Command bit in the CTRLB register is set (CTRLB.GCMD = 1) and 7-bit addressing is used, INTFLAG.PREC is set if the client was addressed since the last STOP condition. When CTRLB.GCMD = 0, a STOP condition without an address match is not set INTFLAG.PREC.

© 2025 Microchip Technology Inc. and its subsidiaries

The group command protocol is used to send commands to more than one device. The commands are sent in one continuous transmission with a single STOP condition at the end. When the STOP condition is detected by the clients addressed during the group command, they all begin executing the command they received.

The following figure illustrates an example where this client, bearing ADDRESS 1, is addressed after a repeated START condition. There can be multiple clients addressed before and after this client. Eventually, at the end of the group command, a single STOP is generated by the host. At this point, a STOP interrupt is asserted.

Figure 34-10. PMBus Group Command Example



34.6.3. Additional Features

34.6.3.1. SMBus

The I^{2}C includes three hardware SCL low time-outs that allow a time-out to occur for SMBus SCL low time-out, host extend time-out and client extend time-out. This allows for SMBus functionality. These time-outs are driven by the GCLK_SERCOM_SLOW clock. The GCLK_SERCOM_SLOW clock is used to accurately time the time-out and must be configured to use a 32 KHz_LPCLK. The I^{2}C interface also allows for a SMBus compatible SDA hold time.

- $T_{TIMEOUT}$: SCL low time of 25-35 ms – Measured for a single SCL low period. It is enabled by CTRLA.LOWTOUTEN.
- $T_{LOW:SEXT}$: Cumulative clock low extend time of 25 ms – Measured as the cumulative SCL low extend time by a client device in a single message from the initial START to the STOP. It is enabled by CTRLA.SEXTTOEN.
- $T_{LOW:MEXT}$: Cumulative clock low extend time of 10 ms – Measured as the cumulative SCL low extend time by the host device within a single byte from START-to-ACK, ACK-to-ACK, or ACK-to-STOP. It is enabled by CTRLA.MEXTTOEN.

34.6.3.2. Smart Mode

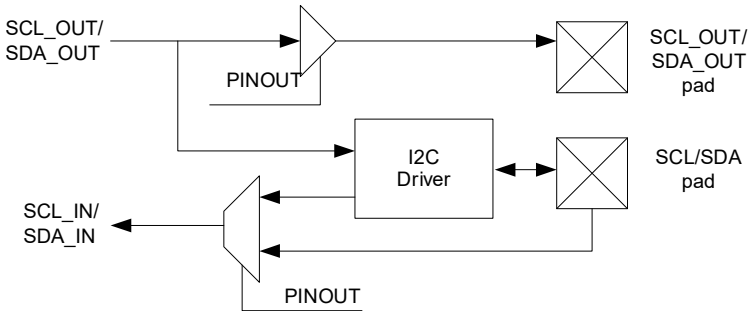
The I^{2}C interface has a Smart mode that simplifies application code and minimizes the user interaction needed to adhere to the I^{2}C protocol. The Smart mode accomplishes this by automatically issuing an ACK or NACK (based on the content of CTRLB.ACKACT) as soon as DATA.DATA is read.

© 2025 Microchip Technology Inc. and its subsidiaries

34.6.3.3. Four-Wire Mode

Writing a '1' to the Pin Usage bit in the Control A register (CTRLA.PINOUT) enables Four-Wire mode operation. In this mode, the internal I^2C tri-state drivers are bypassed and an external I^2C compliant tri-state driver is needed when connecting to an I^2C bus.

Figure 34-11. I^2C Pad Interface



34.6.3.4. Quick Command

Setting the Quick Command Enable bit in the Control B register (CTRLB.QCEN) enables quick command. When quick command is enabled, the corresponding Interrupt flag (INTFLAG.SB or INTFLAG.MB) is set immediately after the client acknowledges the address. At this point, the software can either issue a Stop command or a repeated start by writing CTRLB.CMD or ADDR.ADDR.

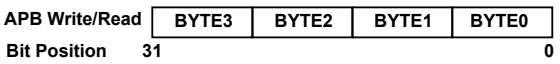
34.6.3.5. 32-Bit Extension

For better system bus utilization, 32-bit data receive and transmit can be enabled by writing to the Data 32-bit bit field in the Control C register (CTRLC.DATA32B = 1). When enabled, write and read transactions to/from the DATA register are 32 bits in size.

If frames are not multiples of 4 bytes, the Length Counter (LENGTH.LEN) and Length Enable (LENGTH.LENEN) must be configured before data transfer begins. LENGTH.LEN must be enabled only when CTRLC.DATA32B is enabled.

The following figure illustrates the order of transmit and receive when using 32-bit mode. Bytes are transmitted or received and stored in order from 0-3.

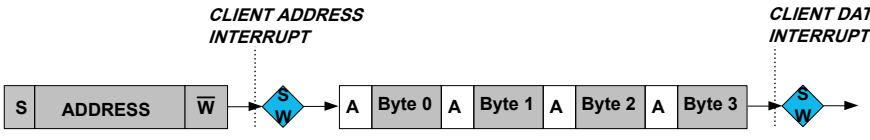
Figure 34-12. 32-Bit Extension Byte Ordering



32-Bit Extension Client Operation

The following figure illustrates a transaction with 32-bit Extension enabled (CTRLC.DATA32B = 1). In client operation, the Address Match interrupt in the Interrupt Flag Status and Clear register (INTFLAG.AMATCH) is set after the address is received and available in the DATA register. The Data Ready interrupt (INTFLAG.DRDY) will, then, be raised for every 4 bytes transferred.

Figure 34-13. 32-Bit Extension Client Operation



The LENGTH register can be written before the frame begins or when the AMATCH interrupt is set. If the frame size is not LENGTH.LEN bytes, the Length Error status bit (STATUS.LENERR) is raised.

© 2025 Microchip Technology Inc. and its subsidiaries

If LENGTH.LEN is not a multiple of 4 bytes, the final INTFLAG.DRDY interrupt is raised when the last byte is received for host reads. For host writes, the last data byte is automatically NACKed. On address recognition, the internal length counter is reset in preparation for the incoming frame.

When SCL clock stretch mode is selected (CTRLA.SCLSM = 1) and the transaction is a host write, the selected Acknowledge Action (CTRLB.ACKACT) will only be used to ACK/NACK each 4th byte. All other bytes are ACKed. This allows the user to write CTRLB.ACKACT = 1 in the final interrupt, so that the last byte in a 32-bit word is NACKed.

Writing to the LENGTH register while a frame is in progress produces unpredictable results. If LENGTH.LENEN is not set and a frame is not a multiple of 4 bytes, the remainder is lost.

32-Bit Extension Host Operation

When using the I^2C configured as host, the Address register must be written with the desired address (ADDR.ADDR) and, optionally, the transaction Length and transaction Length Enable bits (ADDR.LEN and ADDR.LENEN) can be written. When ADDR.LENEN is written to '1' along with ADDR.ADDR, ADDR.LEN determines the number of data bytes in the transaction from 0 to 255. Then, the ADDR.LEN bytes are transferred, followed by an automatically generated NACK (for host reads) and a STOP.

The INTFLAG.SB or INTFLAG.MB are raised for every 4 bytes transferred. If the transaction is a host read and ADDR.LEN is not a multiple of 4 bytes, the final INTFLAG.SB is set when the last byte is received.

When the SCL Clock Stretch mode is enabled (CTRLA.SCLSM = 1) and the transaction is a host read, the selected Acknowledge Action (CTRLB.ACKACT) is used to ACK/NACK each 4th byte. All other bytes are ACKed. This allows the user to set CTRLB.ACKACT = 1 in the final interrupt, so that the last byte in a 32-bit word is NACKed.

If a NACK is received by the client for a host write transaction before ADDR.LEN bytes, a STOP is automatically generated and the length error (STATUS.LENERR) is raised along with the INTFLAG.ERROR interrupt.

34.6.4. DMA, Interrupts and Events

Each interrupt source has its own interrupt flag. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) is set when the interrupt condition is met. Each interrupt can be individually enabled by writing '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET) and disabled by writing '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR). An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request is active until the interrupt flag is cleared, the interrupt is disabled or the I^2C is reset. See the *INTFLAG* (Client) or *INTFLAG* (Host) register from Related Links for details on how to clear interrupt flags.

Table 34-2. Module Request for SERCOM I^2C Client

Condition	Request		
	DMA	Interrupt	Event
Data needed for transmit (TX) (Client transmit mode)	Yes (request cleared when data is written)	—	NA
Data received (RX) (Client receive mode)	Yes (request cleared when data is read)	—	
Data Ready (DRDY)	—	Yes	
Address Match (AMATCH)	—	Yes	
Stop received (PREC)	—	Yes	
Error (ERROR)	—	Yes	

© 2025 Microchip Technology Inc. and its subsidiaries

Table 34-3. Module Request for SERCOM I²C Host

Condition	Request		
	DMA	Interrupt	Event
Data needed for transmit (TX) (Host transmit mode)	Yes (request cleared when data is written)	—	NA
Data needed for transmit (RX) (Host transmit mode)	Yes (request cleared when data is read)	—	
Host on Bus (MB)	—	Yes	
Stop received (SB)	—	Yes	
Error (ERROR)	—	Yes	

Related Links

INTFLAG

Interrupt Flag Status and Clear

INTFLAG

Interrupt Flag Status and Clear

34.6.4.1. DMA Operation

Smart mode must be enabled for DMA operation in the Control B register by writing CTRLB.SMEN = 1.

34.6.4.1.1. Host DMA

When using the I²C host with DMA, the ADDR register must be written with the desired address (ADDR.ADDR), transaction length (ADDR.LEN) and transaction length enable (ADDR.LENEN). When ADDR.LENEN is written to '1' along with ADDR.ADDR, ADDR.LEN determines the number of data bytes in the transaction from 0-255. DMA is, then, used to transfer ADDR.LEN bytes followed by an automatically generated NACK (for host reads) and a STOP.

If a NACK is received by the client for a host write transaction before ADDR.LEN bytes, a STOP is automatically generated and the length error (STATUS.LENERR) is raised along with the INTFLAG.ERROR interrupt.

The I²C host generates the following requests:

- Write data received (RX) – The request is set when host write data is received. The request is cleared when DATA is read.
- Read data needed for transmit (TX) – The request is set when data is needed for a host read operation. The request is cleared when DATA is written.
- Write data received (RX) – If the FIFO is disabled, the request is set when host write data is received. If the FIFO is enabled, the request is set when the RX FIFO threshold is reached (CTRLC.RXTRHOLD). The request is cleared when DATA is read.
- Read data needed for transmit (TX) – If the FIFO is disabled, the request is set when data are needed for a host read operation. If the FIFO is enabled, the request is set when the TX FIFO threshold is reached (CTRLC.TXTRHOLD). The request is cleared when DATA is written.

34.6.4.1.2. Client DMA

When using the I²C client with DMA, an address match causes the address Interrupt flag (INTFLAG.ADDRMATCH) to be raised. After the interrupt is serviced, the data transfer is performed through DMA.

The I²C client generates the following requests:

- Read data received (RX) – The request is set when host read data is received. The request is cleared when DATA is read.

© 2025 Microchip Technology Inc. and its subsidiaries

- Write data needed for transmit (TX) – The request is set when data are needed for a host write operation. The request is cleared when DATA is written.
- Read data received (RX) – If the FIFO is disabled, the request is set when host read data are received. If the FIFO is enabled, the request is set when the RX FIFO threshold is reached. The request is cleared when DATA is read.
- Write data needed for transmit (TX) – If the FIFO is disabled, the request is set when data are needed for a host write operation. If the FIFO is enabled, the request is set when the TX FIFO threshold is reached (CTRLC.TXTRHOLD). The request is cleared when DATA is written.

34.6.4.2. Interrupts

The I^{2}C client has the following interrupt sources. These are asynchronous interrupts. They can wake-up the device from any Sleep mode:

- Error (ERROR)
- Data Ready (DRDY)
- Address Match (AMATCH)
- Stop Received (PREC)

The I^{2}C host has the following interrupt sources. These are asynchronous interrupts. They can wake-up the device from any Sleep mode:

- Error (ERROR)
- Client on Bus (SB)
- Host on Bus (MB)

Each interrupt source has its own Interrupt flag. The Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) will be set when the Interrupt condition is 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).

The status of enabled interrupts can be read from either INTENSET or INTENCLR. An interrupt request is generated when the Interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the Interrupt flag is cleared, the interrupt is disabled or the I^{2}C is reset. For details on how to clear Interrupt flags, see *INTFLAG* register from Related Links.

The value of INTFLAG indicates which interrupt is executed. Note that interrupts must be globally enabled for interrupt requests. See *Nested Vector Interrupt Controller (NVIC)* from Related Links.

Related Links

Nested Vector Interrupt Controller (NVIC)

INTFLAG

Interrupt Flag Status and Clear

34.6.4.3. Events

Not applicable.

34.6.4.4. Sleep Mode Operation

I^{2}C Host Operation

The generic clock (GCLK_SERCOMx_CORE) continues to run in the Idle Sleep mode. If the Run In Standby bit in the Control A register (CTRLA.RUNSTDBY) is '1', the GLK_SERCOMx_CORE also runs in the Standby Sleep mode. Any interrupt can wake up the device.

If CTRLA.RUNSTDBY = 0, the GLK_SERCOMx_CORE is disabled after any ongoing transaction is finished. Any interrupt can wake up the device.

I^{2}C Client Operation

© 2025 Microchip Technology Inc. and its subsidiaries

Writing CTRLA.RUNSTDBY = 1 allows the Address Match interrupt to wake up the device.

When CTRLA.RUNSTDBY = 0, all receptions are dropped.

© 2025 Microchip Technology Inc. and its subsidiaries

34.7. Register Summary - I2C Client

See the *SERCOM0/SERCOM1/SERCOM2* module in the *Product Memory Mapping Overview* from Related Links for the base address based on the SERCOM instant used.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
		15:8								
		23:16	SEXTTOEN							PINOUT
		31:24		LOWTOUT			SCLSM		SPEED[1:0]	
0x04	CTRLB	7:0								
		15:8	AMODE[1:0]					AACKEN	GCMD	SMEN
		23:16						ACKACT	CMD[1:0]	
		31:24								
0x08	CTRLC	7:0					SDASETUP[3:0]			
		15:8								
		23:16								
		31:24								DATA32B
0x0C ... 0x13	Reserved									
0x14	INTENCLR	7:0	ERROR					DRDY	AMATCH	PREC
0x15	Reserved									
0x16	INTENSET	7:0	ERROR					DRDY	AMATCH	PREC
0x17	Reserved									
0x18	INTFLAG	7:0	ERROR					DRDY	AMATCH	PREC
0x19	Reserved									
0x1A	STATUS	7:0	CLKHOLD	LOWTOUT		SR	DIR	RXNACK	COLL	BUSERR
		15:8					LENERR		SEXTTOUT	
0x1C	SYNCBUSY	7:0				LENGTH			ENABLE	SWRST
		15:8								
		23:16								
		31:24								
0x20 ... 0x21	Reserved									
0x22	LENGTH	7:0	LEN[7:0]							
		15:8								LENEN
0x24	ADDR	7:0	ADDR[6:0]							GENCEN
		15:8						ADDR[9:7]		
		23:16	ADDRMASK[6:0]							
		31:24						ADDRMASK[9:7]		
0x28	DATA	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							

Related Links

Product Memory Mapping Overview

34.8. Register Description - I²C Client

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

Some registers are optionally write protected by the PAC. Optional PAC write protection is denoted by the PAC Write-Protection property in each individual register description.

Some registers are synchronized when read and/or written. Synchronization is denoted by the Write-Synchronized or the Read-Synchronized property in each individual register description.

© 2025 Microchip Technology Inc. and its subsidiaries

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable protection is denoted by the Enable-Protected property in each individual register description.

© 2025 Microchip Technology Inc. and its subsidiaries

34.8.1. Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
		LOWTOUT			SCLSM		SPEED[1:0]	
Access		R/W			R/W		R/W	R/W
Reset		0			0		0	0

Bit	23	22	21	20	19	18	17	16
	SEXTTOEN							PINOUT
Access	R/W							R/W
Reset	0							0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			0	0	0	0	0

Bit 30 – LOWTOUT SCL Low Time-Out

This bit enables the SCL low time-out. If SCL is held low for 25-35 ms, the client releases its clock hold, if enabled, and resets the internal state machine. Any interrupt flags set at the time of time-out remain set.
This bit is not synchronized.

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

Bit 27 – SCLSM SCL Clock Stretch Mode

This bit controls when SCL is stretched for software interaction.
This bit is not synchronized.

Value	Description
0	SCL stretch according to Figure 34-8.
1	SCL stretch only after ACK bit according to Figure 34-9.

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	Reserved
0x3	Reserved

© 2025 Microchip Technology Inc. and its subsidiaries

Bit 23 – SEXTTOEN Client SCL Low Extend Time-Out

This bit enables the client SCL low extend time-out. If SCL is cumulatively held low for greater than 25 ms from the initial START to a STOP, the client releases its clock hold, if enabled, and resets the internal state machine. Any interrupt flags set at the time of time-out remain set. If the address was recognized, PREC is set when a STOP is received.
This bit is not synchronized.

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

Bit 16 – PINOUT Pin Usage

This bit sets the pin usage to either two- or four-wire operation:
This bit is not synchronized.

Value	Description
0	Four-wire operation disabled
1	Four-wire operation enabled

Bit 7 – RUNSTDBY Run in Standby

This bit defines the functionality in the Standby Sleep mode.
This bit is not synchronized.

Value	Description
0	Disabled – All reception is dropped.
1	Wake on address match, if enabled.

Bits 4:2 – MODE[2:0] Operating Mode

These bits must be written to 0x04 to select the I^2C client 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 reads back immediately and the Enable Synchronization Busy bit in the Synchronization Busy register (SYNCBUSY.ENABLE) is set. SYNCBUSY.ENABLE is cleared when the operation is complete.
This bit is not enable-protected.

Value	Description
0	The peripheral is disabled or being disabled.
1	The peripheral is enabled.

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 is disabled.
Writing '1' to CTRLA.SWRST always takes precedence, meaning that all other writes in the same write operation are discarded. Any register write access during the ongoing Reset results in an APB error.
Reading any register returns 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.
Note: During SWRST, access to registers/bits without SWRST are disallowed until the hardware clears SYNCBUSY.SWRST.

© 2025 Microchip Technology Inc. and its subsidiaries

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

© 2025 Microchip Technology Inc. and its subsidiaries

34.8.2. Control B

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

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						ACKACT	CMD[1:0]	
Access						R/W	W	W
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
	AMODE[1:0]					AACKEN	GCMD	SMEN
Access	R/W	R/W				R/W	R/W	R/W
Reset	0	0				0	0	0

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 18 – ACKACT Acknowledge Action

This bit defines the client's acknowledge behavior after an address or data byte is received from the host. The acknowledge action is executed when a command is written to the CMD bits. If Smart mode is enabled (CTRLB.SMEN = 1), the acknowledge action is performed when the DATA register is read.
ACKACT might not be updated more than once between each peripheral interrupts request.
This bit is not enable-protected.

Value	Description
0	Send ACK
1	Send NACK

Bits 17:16 – CMD[1:0] Command

This bit field triggers the client operation as follows. The CMD bits are strobe bits and always read as '0'. The operation is dependent on the client interrupt flags, INTFLAG.DRDY and INTFLAG.AMATCH, in addition to STATUS.DIR.
All interrupt flags (INTFLAG.DRDY, INTFLAG.AMATCH and INTFLAG.PREC) are automatically cleared when a command is given.
This bit is not enable-protected.

Table 34-4. Command Description

CMD[1:0]	DIR	Action
0x0	X	(No action)
0x1	X	(Reserved)
0x2	Used to complete a transaction in response to a data interrupt (DRDY)	
	0 (Host write)	Execute acknowledge action succeeded by waiting for any start (S/Sr) condition
	1 (Host read)	Wait for any start (S/Sr) condition

© 2025 Microchip Technology Inc. and its subsidiaries

Table 34-4. Command Description (continued)

CMD[1:0]	DIR	Action
0x3	Used in response to an address interrupt (AMATCH)	
	0 (Host write)	Execute acknowledge action succeeded by reception of next byte
	1 (Host read)	Execute acknowledge action succeeded by client data interrupt
	Used in response to a data interrupt (DRDY)	
	0 (Host write)	Execute acknowledge action succeeded by reception of next byte
	1 (Host read)	Execute a byte read operation followed by ACK/NACK reception

Bits 15:14 – AMODE[1:0] Address Mode

These bits set the Addressing mode.

Value	Name	Description
0x0	MASK	The client responds to the address written in ADDR.ADDR masked by the value in ADDR.ADDRMASK.
0x1	2_ADDRS	The client responds to the two unique addresses in ADDR.ADDR and ADDR.ADDRMASK.
0x2	RANGE	The client responds to the range of addresses between and including ADDR.ADDR and ADDR.ADDRMASK. ADDR.ADDR is the upper limit.
0x3	—	Reserved.

Bit 10 – AACKEN Automatic Acknowledge Enable

This bit enables the address to be automatically acknowledged if there is an address match.

Value	Description
0	Automatic acknowledge is disabled.
1	Automatic acknowledge is enabled.

Bit 9 – GCMD PMBus Group Command

This bit enables PMBus group command support. When enabled, the Stop Received interrupt flag (INTFLAG.PREC) is set when a STOP condition is detected if the client was addressed since the last STOP condition on the bus.

Value	Description
0	Group command is disabled.
1	Group command is enabled.

Bit 8 – SMEN Smart Mode Enable

When Smart mode is enabled, data are acknowledged automatically when DATA.DATA is read.

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

© 2025 Microchip Technology Inc. and its subsidiaries

34.8.3. Control C

Name: CTRLC
Offset: 0x08
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
								DATA32B
Access								R/W
Reset								0

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
					SDASETUP[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 24 – DATA32B Data 32 Bit
This bit enables 32-bit data writes and reads to/from the DATA register.

Value	Description
0	Data transaction to/from DATA are 8-bit in size
1	Data transaction to/from DATA are 32-bit in size

Bits 3:0 – SDASETUP[3:0] SDA Setup Time
These bits select the minimum SDA-to-SCL setup time, measured from the release of SDA to the release of SCL:
$t_{\mathrm{SU:DAT}} = (GCLK\_SERCOMx \times \mathrm{APB\ period}\ (PBx\_CLK)) \times (6 + 16 \times \mathrm{SDASETUP})$

© 2025 Microchip Technology Inc. and its subsidiaries

34.8.4. Interrupt Enable Clear

Name: INTENCLR
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection

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

Bit	7	6	5	4	3	2	1	0
	ERROR					DRDY	AMATCH	PREC
Access	R/W					R/W	R/W	R/W
Reset	0					0	0	0

Bit 7 – ERROR Error Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the Error Interrupt Enable bit, which disables the Error interrupt.

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

Bit 2 – DRDY Data Ready Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the Data Ready bit, which disables the Data Ready interrupt.

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

Bit 1 – AMATCH Address Match Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the Address Match Interrupt Enable bit, which disables the Address Match interrupt.

Value	Description
0	The Address Match interrupt is disabled.
1	The Address Match interrupt is enabled.

Bit 0 – PREC Stop Received Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the Stop Received Interrupt Enable bit, which disables the Stop Received interrupt.

Value	Description
0	The Stop Received interrupt is disabled.
1	The Stop Received interrupt is enabled.

© 2025 Microchip Technology Inc. and its subsidiaries

34.8.5. Interrupt Enable Set

Name: INTENSET
Offset: 0x16
Reset: 0x00
Property: PAC Write-Protection

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

Bit	7	6	5	4	3	2	1	0
	ERROR					DRDY	AMATCH	PREC
Access	R/W					R/W	R/W	R/W
Reset	0					0	0	0

Bit 7 – ERROR Error Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit sets the Error Interrupt Enable bit, which enables the Error interrupt.

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

Bit 2 – DRDY Data Ready Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit sets the Data Ready Interrupt Enable bit, which enables the Data Ready interrupt.

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

Bit 1 – AMATCH Address Match Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit sets the Address Match Interrupt Enable bit, which enables the Address Match interrupt.

Value	Description
0	The Address Match interrupt is disabled.
1	The Address Match interrupt is enabled.

Bit 0 – PREC Stop Received Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit sets the Stop Received Interrupt Enable bit, which enables the Stop Received interrupt.

Value	Description
0	The Stop Received interrupt is disabled.
1	The Stop Received interrupt is enabled.

© 2025 Microchip Technology Inc. and its subsidiaries

34.8.6. Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x18
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	ERROR					DRDY	AMATCH	PREC
Access	R/W					R/W	R/W	R/W
Reset	0					0	0	0

Bit 7 – ERROR Error
This bit is set when any error is detected. Errors that will set this flag have corresponding status flags in the STATUS register. The corresponding bits in STATUS are LENERR, SEXTTOUT, LOWTOUT, COLL and BUSERR.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the flag.

Bit 2 – DRDY Data Ready
This flag is set when a I^2C client byte transmission is successfully completed.
The flag is cleared by hardware when either:

- Writing to the DATA register.
- Reading the DATA register with Smart mode enabled.
- Writing a valid command to the CMD register.

Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Data Ready Interrupt flag.

Bit 1 – AMATCH Address Match
This flag is set when the I^2C client address match logic detects that a valid address has been received.
The flag is cleared by hardware when CTRL.CMD is written.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Address Match Interrupt flag. When cleared, an ACK/NACK will be sent according to CTRLB.ACKACT.

Bit 0 – PREC Stop Received
This flag is set when a Stop condition is detected for a transaction being processed. A Stop condition detected between a bus host and another client will not set this flag, unless the PMBus Group Command is enabled in the Control B register (CTRLB.GCMD=1).
This flag is cleared by hardware after a command is issued on the next address match.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Stop Received Interrupt flag.

© 2025 Microchip Technology Inc. and its subsidiaries

34.8.7. Status

Name: STATUS
Offset: 0x1A
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
					LENERR		SEXTTOUT	
Access					R/W		R/W	
Reset					0		0	

Bit	7	6	5	4	3	2	1	0
	CLKHOLD	LOWTOUT		SR	DIR	RXNACK	COLL	BUSERR
Access	R/W	R/W		R	R	R	R/W	R/W
Reset	0	0		0	0	0	0	0

Bit 11 – LENERR Transaction Length Error
This bit is set when the length counter is enabled (LENGTH.LENEN) and a STOP or repeated START is received before or after the length in LENGTH.LEN is reached.
This bit is cleared automatically if responding to a new start condition with ACK or NACK (write 3 to CTRLB.CMD) or when INTFLAG.AMATCH is cleared.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the status.

Value	Description
0	No length error has occurred.
1	Length error has occurred.

Bit 9 – SEXTTOUT Client SCL Low Extend Time-Out
This bit is set if a client SCL low extend time-out occurs.
This bit is cleared automatically if responding to a new start condition with ACK or NACK (write 3 to CTRLB.CMD) or when INTFLAG.AMATCH is cleared.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the status.

Value	Description
0	No SCL low extend time-out has occurred.
1	SCL low extend time-out has occurred.

Bit 7 – CLKHOLD Clock Hold
The client Clock Hold bit (STATUS.CLKHOLD) is set when the client is holding the SCL line low, stretching the I2C clock. Software must consider this bit a read-only status flag that is set when INTFLAG.DRDY or INTFLAG.AMATCH is set.
This bit is automatically cleared when the corresponding interrupt is also cleared.

Bit 6 – LOWTOUT SCL Low Time-out
This bit is set if an SCL low time-out occurs.
This bit is cleared automatically if responding to a new start condition with ACK or NACK (write 3 to CTRLB.CMD) or when INTFLAG.AMATCH is cleared.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the status.

Value	Description
0	No SCL low time-out has occurred.

© 2025 Microchip Technology Inc. and its subsidiaries

Value	Description
1	SCL low time-out has occurred.

Bit 4 – SR Repeated Start

When INTFLAG.AMATCH is raised due to an address match, SR indicates a repeated start or start condition.
This flag is only valid while the INTFLAG.AMATCH flag is one.

Value	Description
0	Start condition on last address match
1	Repeated start condition on last address match

Bit 3 – DIR Read / Write Direction

The Read/Write Direction (STATUS.DIR) bit stores the direction of the last address packet received from a host .

Value	Description
0	Host write operation is in progress.
1	Host read operation is in progress.

Bit 2 – RXNACK Received Not Acknowledge

This bit indicates whether the last data packet sent was acknowledged or not.

Value	Description
0	Host responded with ACK.
1	Host responded with NACK.

Bit 1 – COLL Transmit Collision

If set, the I2C client was not able to transmit a high data or NACK bit, the I2C client will immediately release the SDA and SCL lines and wait for the next packet addressed to it.
This flag is intended for the SMBus address resolution protocol (ARP). A detected collision in non-ARP situations indicates that there has been a protocol violation, and must be treated as a bus error.
Note: This status will not trigger any interrupt, and must be checked by software to verify that the data were sent correctly. This bit is cleared automatically if responding to an address match with an ACK or a NACK (writing 0x3 to CTRLB.CMD), or INTFLAG.AMATCH is cleared.

Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the status.

Value	Description
0	No collision detected on last data byte sent.
1	Collision detected on last data byte sent.

Bit 0 – BUSERR Bus Error

The Bus Error bit (STATUS.BUSERR) indicates that an illegal bus condition has occurred on the bus, regardless of bus ownership. An illegal bus condition is detected if a protocol violating start, repeated start or stop is detected on the I2C bus lines. A start condition directly followed by a stop condition is one example of a protocol violation. If a time-out occurs during a frame, this is also considered a protocol violation, and will set STATUS.BUSERR.
This bit is cleared automatically if responding to an address match with an ACK or a NACK (writing 0x3 to CTRLB.CMD) or INTFLAG.AMATCH is cleared.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the status.

Value	Description
0	No bus error detected.
1	Bus error detected.

© 2025 Microchip Technology Inc. and its subsidiaries

34.8.8. Synchronization Busy

Name: SYNCBUSY
Offset: 0x1C
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				LENGTH			ENABLE	SWRST
Access				R			R	R
Reset				0			0	0

Bit 4 – LENGTH LENGTH Synchronization Busy
Writing LENGTH requires synchronization. When written, this bit will be set until synchronization is complete. If LENGTH is written while SYNCBUSY.LENGTH is asserted, an APB error will be generated.
Note: In client mode, the clock is only running during data transfer, so SYNCBUSY.LENGTH will remain asserted until the next data transfer begins.

Value	Description
0	LENGTH synchronization is not busy.
1	LENGTH synchronization is busy.

Bit 1 – ENABLE SERCOM Enable Synchronization Busy
Enabling and disabling the SERCOM (CTRLA.ENABLE) requires synchronization. Ongoing synchronization is indicated by SYNCBUSY.ENABLE = 1 until synchronization is complete.

Value	Description
0	Enable synchronization is not busy.
1	Enable synchronization is busy.

Bit 0 – SWRST Software Reset Synchronization Busy
Resetting the SERCOM (CTRLA.SWRST) requires synchronization. Ongoing synchronization is indicated by SYNCBUSY.SWRST = 1 until synchronization is complete.
Note: During a SWRST, access to registers/bits without SWRST are disallowed until SYNCBUSY.SWRST cleared by hardware.

Value	Description
0	SWRST synchronization is not busy.
1	SWRST synchronization is busy.

© 2025 Microchip Technology Inc. and its subsidiaries

34.8.9. Length

Name: LENGTH
Offset: 0x22
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
								LENEN
Access								R/W
Reset								0

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

Bit 8 – LENEN Data Length Enable
In 32-bit Extension mode (CTRLC.DATA32B = 1), this bit field enables the length counter.

Value	Description
0	Length counter is disabled.
1	Length counter is enabled.

Bits 7:0 – LEN[7:0] Data Length
In 32-bit Extension mode (CTRLC.DATA32B = 1) with Data Length counting enabled (LENGTH.LENEN), this bit field configures the data length from 0 to 255 bytes, after which, the flag INTFLAG.DRDY is raised.

© 2025 Microchip Technology Inc. and its subsidiaries

34.8.10. Address

Name: ADDR
Offset: 0x24
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
						ADDRMASK[9:7]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
	ADDRMASK[6:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	

Bit	15	14	13	12	11	10	9	8
						ADDR[9:7]		
Access						R/W	R/W	R/W
Reset						0	0	0

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

Bits 26:17 – ADDRMASK[9:0] Address Mask
These bits act as a second address match register, an address mask register or the lower limit of an address range, depending on the CTRLB.AMODE setting.

Bits 10:1 – ADDR[9:0] Address
These bits contain the I^2C client address used by the client address match logic to determine if a host has addressed the client.
When using 7-bit addressing, the client address is represented by ADDR[6:0].
When the address match logic detects a match, INTFLAG.AMATCH is set and STATUS.DIR is updated to indicate whether it is a read or a write transaction.

Bit 0 – GENCEN General Call Address Enable
A general call address is an address consisting of all-zeroes, including the direction bit (host write).

Value	Description
0	General call address recognition disabled.
1	General call address recognition enabled.

© 2025 Microchip Technology Inc. and its subsidiaries

34.8.11. Data

Name: DATA
Offset: 0x28
Reset: 0x00000000
Property: Read/Write

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

The client data register I/O location (DATA.DATA) provides access to the host transmit and receive data buffers. Reading valid data or writing data to be transmitted can be successfully done only when SCL is held low by the client (STATUS.CLKHOLD is set). An exception occurs when reading the last data byte after the stop condition has been received.

Accessing DATA.DATA auto-triggers I^2C bus operations. The operation performed depends on the state of CTRLB.ACKACT, CTRLB.SMEN and the type of access (read/write).

When CTRLC.DATA32B=1, read and write transactions from/to the DATA register are 32 bit in size. Otherwise, reads and writes are 8 bit.

© 2025 Microchip Technology Inc. and its subsidiaries

34.9. Register Summary - I2C Host

See the *SERCOM0/SERCOM1/SERCOM2/SERCOM6* module in the *Product Memory Mapping Overview* from Related Links for the base address based on the SERCOM instant used.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
		15:8								
		23:16	SEXTTOEN	MEXTTOEN						PINOUT
		31:24		LOWTOUT	INACTOUT[1:0]		SCLSM		SPEED[1:0]	
0x04	CTRLB	7:0								
		15:8							QCEN	SMEN
		23:16						ACKACT	CMD[1:0]	
		31:24								
0x08	CTRLC	7:0								
		15:8								
		23:16								
		31:24								DATA32B
0x0C	BAUD	7:0	BAUD[7:0]							
		15:8	BAUDLOW[7:0]							
		23:16								
		31:24								
0x10 ... 0x13	Reserved									
0x14	INTENCLR	7:0	ERROR						SB	MB
0x15	Reserved									
0x16	INTENSET	7:0	ERROR						SB	MB
0x17	Reserved									
0x18	INTFLAG	7:0	ERROR						SB	MB
0x19	Reserved									
0x1A	STATUS	7:0	CLKHOLD	LOWTOUT	BUSSTATE[1:0]			RXNACK	ARBLOST	BUSERR
		15:8						LENERR	SEXTTOUT	MEXTTOUT
0x1C	SYNCBUSY	7:0						SYSOP	ENABLE	SWRST
		15:8								
		23:16								
		31:24								
0x20 ... 0x23	Reserved									
0x24	ADDR	7:0	ADDR[7:0]							
		15:8	TENBITEN		LENEN			ADDR[10:8]		
		23:16	LEN[7:0]							
		31:24								
0x28	DATA	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x2C ... 0x2F	Reserved									
0x30	DBGCTRL	7:0								DBGSTOP

Related Links

Product Memory Mapping Overview

34.10. Register Description - I2C Host

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.

© 2025 Microchip Technology Inc. and its subsidiaries

Some registers are optionally write-protected by the PAC. Optional PAC write protection is denoted by the PAC Write-Protection property in each individual register description.

Some registers are synchronized when read and/or written. Synchronization is denoted by the Write-Synchronized or the Read-Synchronized property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable protection is denoted by the Enable-Protected property in each individual register description.

© 2025 Microchip Technology Inc. and its subsidiaries

34.10.1. Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
		LOWTOUT	INACTOUT[1:0]		SCLSM		SPEED[1:0]	
Access		R/W	R/W	R/W	R/W		R/W	R/W
Reset		0	0	0	0		0	0

Bit	23	22	21	20	19	18	17	16
	SEXTTOEN	MEXTTOEN						PINOUT
Access	R/W	R/W						R/W
Reset	0	0						0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			0	0	0	0	0

Bit 30 – LOWTOUT SCL Low Time-Out

This bit enables the SCL low time-out. If SCL is held low for 25-35 ms, the host releases its clock hold, if enabled, and complete the current transaction. A stop condition will automatically be transmitted. INTFLAG.SB or INTFLAG.MB is set as normal but the clock hold is released. The STATUS.LOWTOUT and STATUS.BUSERR status bits is set.
This bit is not synchronized.

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

Bits 29:28 – INACTOUT[1:0] Inactive Time-Out

If the inactive bus time-out is enabled and the bus is inactive for longer than the time-out setting, the bus state logic is set to Idle. An inactive bus arises when either an I^2C host or client is holding the SCL low.
Enabling this option is necessary for SMBus compatibility but can also be used in a non-SMBus setup.
Calculated time-out periods are based on a 100 kHz baud rate.
These bits are not synchronized.

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

© 2025 Microchip Technology Inc. and its subsidiaries

Bit 27 – SCLSM SCL Clock Stretch Mode

This bit controls when SCL is stretched for software interaction.
This bit is not synchronized.

Value	Description
0	SCL stretch according to Figure 34-5.
1	SCL stretch only after ACK bit according to Figure 34-6.

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	Reserved
0x3	Reserved

Bit 23 – SEXTTOEN Client SCL Low Extend Time-Out

This bit enables the client SCL low extend time-out. If SCL is cumulatively held low for greater than 25 ms from the initial START to a STOP, the client releases its clock hold, if enabled, and completes the current transaction. A STOP will automatically be transmitted.
SB or MB is set as normal but CLKHOLD is release. The MEXTTOUT and BUSERR status bits is set.
This bit is not synchronized.

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

Bit 22 – MEXTTOEN Host SCL Low Extend Time-Out

This bit enables the host SCL low extend time-out. If SCL is cumulatively held low for greater than 10 ms from START-to-ACK, ACK-to-ACK or ACK-to-STOP the host releases its clock hold if enabled and completes the current transaction. A STOP will automatically be transmitted.
SB or MB is set as normal but CLKHOLD is released. The MEXTTOUT and BUSERR status bits is set.
This bit is not synchronized.

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

Bit 16 – PINOUT Pin Usage

This bit sets the pin usage to either two- or four-wire operation:
This bit is not synchronized.

Value	Description
0	Four-wire operation disabled.
1	Four-wire operation enabled.

Bit 7 – RUNSTDBY Run in Standby

This bit defines the functionality in the Standby Sleep mode.
This bit is not synchronized.

Value	Description
0	GCLK_SERCOMx_CORE is disabled and the I^2C host will not operate in the Standby Sleep mode.
1	GCLK_SERCOMx_CORE is enabled in all Sleep modes.

© 2025 Microchip Technology Inc. and its subsidiaries

Bits 4:2 – MODE[2:0] Operating Mode
These bits must be written to 0x5 to select the I^2C host 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 reads back immediately and the Synchronization Enable Busy bit in the Synchronization Busy register (SYNCBUSY.ENABLE) is set. SYNCBUSY.ENABLE is cleared when the operation is complete.
This bit is not enable-protected.

Value	Description
0	The peripheral is disabled or being disabled.
1	The peripheral is enabled.

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 is disabled.
Writing '1' to CTRLA.SWRST always takes precedence, meaning that all other writes in the same write operation is discarded. Any register write access during the ongoing Reset results in an APB error.
Reading any register returns 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.
Note: During a SWRST, access to registers/bits without SWRST are disallowed until the hardware clears SYNCBUSY.SWRST.

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

© 2025 Microchip Technology Inc. and its subsidiaries

34.10.2. Control B

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

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						ACKACT	CMD[1:0]	
Access						R/W	W	W
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
							QCEN	SMEN
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 18 – ACKACT Acknowledge Action

This bit defines the I^2C host's acknowledge behavior after a data byte is received from the I^2C client. The acknowledge action is executed when a command is written to CTRLB.CMD or if Smart mode is enabled (CTRLB.SMEN is written to one) when DATA.DATA is read.
This bit is not enable-protected.
This bit is not write-synchronized.

Value	Description
0	Send ACK.
1	Send NACK.

Bits 17:16 – CMD[1:0] Command

Writing these bits triggers a host operation as described below. The CMD bits are strobe bits and always read as '0'. The acknowledge action is only valid in Host Read mode. In Host Write mode, a command will only result in a repeated Start or Stop condition. The CTRLB.ACKACT bit and the CMD bits can be written at the same time, and, then, the acknowledge action is updated before the command is triggered.
Commands can only be issued when either the Client on Bus interrupt flag (INTFLAG.SB) or Host on Bus interrupt flag (INTFLAG.MB) is '1'.
If CMD 0x1 is issued, a repeated start is issued followed by the transmission of the current address in ADDR.ADDR. If another address is desired, ADDR.ADDR must be written instead of the CMD bits. This triggers a repeated start followed by the transmission of the new address.
Issuing a command sets the System Operation bit in the Synchronization Busy register (SYNCBUSY.SYSOP).

© 2025 Microchip Technology Inc. and its subsidiaries

Table 34-5. Command Description

CMD[1:0]	Direction	Action
0x0	X	(No action)
0x1	X	Execute acknowledge action succeeded by repeated Start
0x2	0 (Write)	No operation
	1 (Read)	Execute acknowledge action succeeded by a byte read operation
0x3	X	Execute acknowledge action succeeded by issuing a stop condition

These bits are not enable-protected.

Bit 9 – QCEN Quick Command Enable
This bit is not write-synchronized.

Value	Description
0	Quick Command is disabled.
1	Quick Command is enabled.

Bit 8 – SMEN Smart Mode Enable
When Smart mode is enabled, acknowledge action is sent when DATA.DATA is read.
This bit is not write-synchronized.

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

© 2025 Microchip Technology Inc. and its subsidiaries

34.10.3. Control C

Name: CTRLC
Offset: 0x08
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
								DATA32B
Access								R/W
Reset								0

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
Access								
Reset								

Bit 24 – DATA32B Data 32 Bit
This bit enables 32-bit data writes and reads to/from the DATA register.

Value	Description
0	Data transactions to/from DATA are 8-bit in size
1	Data transactions to/from DATA are 32-bit in size

© 2025 Microchip Technology Inc. and its subsidiaries

34.10.4. Baud Rate

Name: BAUD
Offset: 0x0C
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	BAUDLOW[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

Bits 15:8 – BAUDLOW[7:0] Host Baud Rate Low

If this bit field is non-zero, the SCL low time will be described by the value written.
For more details on how to calculate the frequency, see *Clock Generation – Baud-Rate Generator* from Related Links.

Bits 7:0 – BAUD[7:0] Host Baud Rate

This bit field is used to derive the SCL high time if BAUD.BAUDLOW is non-zero. If BAUD.BAUDLOW is zero, BAUD will be used to generate both high and low periods of the SCL.
For more details on how to calculate the frequency, see *Clock Generation – Baud-Rate Generator* from Related Links.

Related Links

Clock Generation – Baud-Rate Generator

© 2025 Microchip Technology Inc. and its subsidiaries

34.10.5. Interrupt Enable Clear

Name: INTENCLR
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection

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

Bit	7	6	5	4	3	2	1	0
	ERROR						SB	MB
Access	R/W						R/W	R/W
Reset	0						0	0

Bit 7 – ERROR Error Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the Error Interrupt Enable bit, which disables the Error interrupt.

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

Bit 1 – SB Client on Bus Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the Client on Bus Interrupt Enable bit, which disables the Client on Bus interrupt.

Value	Description
0	The Client on Bus interrupt is disabled.
1	The Client on Bus interrupt is enabled.

Bit 0 – MB Host on Bus Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the Host on Bus Interrupt Enable bit, which disables the Host on Bus interrupt.

Value	Description
0	The Host on Bus interrupt is disabled.
1	The Host on Bus interrupt is enabled.

© 2025 Microchip Technology Inc. and its subsidiaries

34.10.6. Interrupt Enable Set

Name: INTENSET
Offset: 0x16
Reset: 0x00
Property: PAC Write-Protection

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

Bit	7	6	5	4	3	2	1	0
	ERROR						SB	MB
Access	R/W						R/W	R/W
Reset	0						0	0

Bit 7 – ERROR Error Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit sets the Error Interrupt Enable bit, which enables the Error interrupt.

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

Bit 1 – SB Client on Bus Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit sets the Client on Bus Interrupt Enable bit, which enables the Client on Bus interrupt.

Value	Description
0	The Client on Bus interrupt is disabled.
1	The Client on Bus interrupt is enabled.

Bit 0 – MB Host on Bus Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit sets the Host on Bus Interrupt Enable bit, which enables the Host on Bus interrupt.

Value	Description
0	The Host on Bus interrupt is disabled.
1	The Host on Bus interrupt is enabled.

© 2025 Microchip Technology Inc. and its subsidiaries

34.10.7. Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x18
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	ERROR						SB	MB
Access	R/W						R/W	R/W
Reset	0						0	0

Bit 7 – ERROR Error

This flag is cleared by writing '1' to it.
This bit is set when any error is detected. Errors that set this flag have corresponding status bits in the STATUS register. These status bits are LENERR, SEXTTOUT, MEXTTOUT, LOWTOUT, ARBLOST and BUSERR.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the flag.

Bit 1 – SB Client on Bus

The Client on Bus flag (SB) is set when a byte is successfully received in the Host Read mode, for example, no arbitration lost or bus error occurred during the operation. When this flag is set, the host forces the SCL line low, stretching the I^2C clock period. The SCL line is released and SB is cleared on one of the following actions:

- Writing to ADDR.ADDR
- Writing to DATA.DATA
- Reading DATA.DATA when Smart mode is enabled (CTRLB.SMEN)
- Writing a valid command to CTRLB.CMD

Writing '1' to this bit location clears the SB flag. The transaction will not continue or be terminated until one of the above actions is performed.
Writing '0' to this bit has no effect.

Bit 0 – MB Host on Bus

This flag is set when a byte is transmitted in Host Write mode. The flag is set regardless of the occurrence of a bus error or an Arbitration Lost condition. MB is also set when arbitration is lost during sending of a NACK in the Host Read mode or when issuing a Start condition if the bus state is unknown. When this flag is set and arbitration is not lost, the host forces the SCL line low, stretching the I^2C clock period. The SCL line is released and MB is cleared on one of the following actions:

- Writing to ADDR.ADDR
- Writing to DATA.DATA
- Reading DATA.DATA when Smart mode is enabled (CTRLB.SMEN)
- Writing a valid command to CTRLB.CMD

Writing '1' to this bit location clears the MB flag. The transaction will not continue or be terminated until one of the above actions is performed.
Writing '0' to this bit has no effect.

© 2025 Microchip Technology Inc. and its subsidiaries

34.10.8. Status

Name: STATUS
Offset: 0x1A
Reset: 0x0000
Property: Write-Synchronized

Bit	15	14	13	12	11	10	9	8
						LENERR	SEXTTOUT	MEXTTOUT
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	CLKHOLD	LOWTOUT	BUSSTATE[1:0]			RXNACK	ARBLOST	BUSERR
Access	R/W	R/W	R/W	R/W		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 and/or 32-bit transaction and the client sends a NACK before ADDR.LEN bytes are written by the host.
Writing '1' to this bit location clears 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 Client SCL Low Extend Time-Out
This bit is set if a client SCL low extend time-out occurs.
This bit is automatically cleared when writing to the ADDR register.
Writing '1' to this bit location clears SEXTTOUT. Normal use of the I^2C interface does not require the SEXTTOUT flag to be cleared by this method.
Writing '0' to this bit has no effect.
This bit is not write-synchronized.

Bit 8 – MEXTTOUT Host SCL Low Extend Time-Out
This bit is set if a Host SCL low time-out occurs.
Writing '1' to this bit location clears 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 host is holding the SCL line low, stretching the I^2C clock. Software must 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. Do not clear this bit to preserve the current Clock Hold state.

Bit 6 – LOWTOUT SCL Low Time-Out
This bit is set if an SCL low time-out occurs.
Writing '1' to this bit location clears this bit. This flag is automatically cleared when writing to the ADDR register.
Writing '0' to this bit has no effect.

© 2025 Microchip Technology Inc. and its subsidiaries

This bit is not write-synchronized.

Bits 5:4 – BUSSTATE[1:0] Bus State

These bits indicate the current I²C Bus state.
When in UNKNOWN state, writing 0x1 to BUSSTATE forces the bus state into the Idle state. The Bus state cannot be forced into any other state.
Writing BUSSTATE to Idle sets SYNCBUSY.SYSOP.

Value	Name	Description
0x0	UNKNOWN	The Bus state is unknown to the I²C host and waits for a Stop condition to be detected or wait to be forced into an Idle state by software
0x1	IDLE	The Bus state is waiting for a transaction to be initialized
0x2	OWNER	The I²C host is the current owner of the bus
0x3	BUSY	Some other I²C host owns the bus

Bit 2 – RXNACK Received Not Acknowledge

This bit indicates whether the last address or data packet sent was acknowledged or not.
Writing '0' to this bit has no effect.
Writing '1' to this bit has no effect.
This bit is not write-synchronized.

Value	Description
0	Client responded with ACK.
1	Client responded with NACK.

Bit 1 – ARBLOST Arbitration Lost

This bit is set if arbitration is lost while transmitting a high data bit or a NACK bit or while issuing a Start or Repeated Start condition on the bus. The Host on Bus Interrupt flag (INTFLAG.MB) is set when STATUS.ARBLOST is set.
Writing the ADDR.ADDR register automatically clears STATUS.ARBLOST.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears it.
This bit is not write-synchronized.

Bit 0 – BUSERR Bus Error

This bit indicates that an illegal Bus condition occurred on the bus, regardless of bus ownership. An illegal Bus condition is detected if a protocol violating start, repeated start or stop is detected on the I²C bus lines. A Start condition directly followed by a Stop condition is one example of a protocol violation. If a time-out occurs during a frame, this is also considered a protocol violation, and sets BUSERR.
If the I²C host is the bus owner at the time a bus error occurs, STATUS.ARBLOST and INTFLAG.MB are set in addition to BUSERR.
Writing the ADDR.ADDR register automatically clears the BUSERR flag.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears it.
This bit is not write-synchronized.

© 2025 Microchip Technology Inc. and its subsidiaries

34.10.9. Synchronization Busy

Name: SYNCBUSY
Offset: 0x1C
Reset: 0x00000000

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						SYSOP	ENABLE	SWRST
Access						R	R	R
Reset						0	0	0

Bit 2 – SYSOP System Operation Synchronization Busy
Writing CTRLB.CMD, STATUS.BUSSTATE, ADDR or DATA when the SERCOM is enabled requires synchronization. When written, the SYNCBUSY.SYSOP bit will be set until synchronization is complete.

Value	Description
0	System operation synchronization is not busy.
1	System operation synchronization is busy.

Bit 1 – ENABLE SERCOM Enable Synchronization Busy
Enabling and disabling the SERCOM (CTRLA.ENABLE) requires synchronization. When written, the SYNCBUSY.ENABLE bit will be set until synchronization is complete.

Value	Description
0	Enable synchronization is not busy.
1	Enable synchronization is busy.

Bit 0 – SWRST Software Reset Synchronization Busy
Resetting the SERCOM (CTRLA.SWRST) requires synchronization. When written, the SYNCBUSY.SWRST bit will be set until synchronization is complete.
Note: During a SWRST, access to registers/bits without SWRST are disallowed until SYNCBUSY.SWRST cleared by hardware.

Value	Description
0	SWRST synchronization is not busy.
1	SWRST synchronization is busy.

© 2025 Microchip Technology Inc. and its subsidiaries

34.10.10. Address

Name: ADDR
Offset: 0x24
Reset: 0x0000
Property: Write-Synchronized

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

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

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

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

Bits 23:16 – LEN[7:0] Transaction Length
These bits define the transaction length of a DMA and/or 32-bit transaction from 0-255 bytes. The Transfer Length Enable (LENEN) bit must be written to '1' to use DMA.

Bit 15 – TENBITEN Ten Bit Addressing Enable
This bit enables 10-bit addressing. This bit can be written simultaneously with ADDR to indicate a 10-bit or 7-bit address transmission.

Value	Description
0	10-bit addressing disabled.
1	10-bit addressing enabled.

Bit 13 – LENEN Transfer Length Enable

Value	Description
0	Automatic transfer length disabled.
1	Automatic transfer length enabled.

Bits 10:0 – ADDR[10:0] Address
When ADDR is written, the consecutive operation depends on the Bus state:

Table 34-6.

State	Description
UNKNOWN	INTFLAG.MB and STATUS.BUSERR are set and the operation is terminated.
BUSY	The I^2C host awaits further operation until the bus becomes IDLE.

© 2025 Microchip Technology Inc. and its subsidiaries

Table 34-6. (continued)

State	Description
IDLE	The I^2C host issues a start condition followed by the address written in ADDR. If the address is acknowledged, SCL is forced and held low and STATUS.CLKHOLD and INTFLAG.MB are set.
OWNER	A repeated start sequence is performed. If the previous transaction was a read, the acknowledge action is sent before the repeated start bus condition is issued on the bus. Writing ADDR to issue a repeated start is performed while INTFLAG.MB or INTFLAG.SB is set.
Note: STATUS.BUSERR, STATUS.ARBLOST, INTFLAG.MB and INTFLAG.SB is cleared when ADDR is written.	

The ADDR register can be read at any time without interfering with ongoing bus activity, as a read access does not trigger the host logic to perform any bus protocol related operations.
The I^2C host control logic uses bit '0' of ADDR as the bus protocol's read/write flag (R/W); '0' for write and '1' for read.

© 2025 Microchip Technology Inc. and its subsidiaries

34.10.11. Data

Name: DATA
Offset: 0x28
Reset: 0x00000000
Property: Read/Write

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

The host data register I/O location (DATA) provides access to the host transmit and receive data buffers. Reading valid data or writing data to be transmitted can be successfully done only when SCL is held low by the host (STATUS.CLKHOLD is set). An exception is reading the last data byte after the stop condition has been sent.

Accessing DATA.DATA auto-triggers I^2C bus operations. The operation performed depends on the state of CTRLB.ACKACT, CTRLB.SMEN and the type of access (read/write).

When CTRLC.DATA32B=1, read and write transactions from/to the DATA register are 32 bit in size. Otherwise, reads and writes are 8 bit.

© 2025 Microchip Technology Inc. and its subsidiaries

34.10.12. Debug Control

Name: DBGCTRL
Offset: 0x30
Reset: 0x00
Property: PAC Write-Protection

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

Bit 0 – DBGSTOP Debug Stop Mode
This bit controls functionality when the CPU is halted by an external debugger.

Value	Description
0	The baud-rate generator continues normal operation when the CPU is halted by an external debugger.
1	The baud-rate generator is halted when the CPU is halted by an external debugger.

© 2025 Microchip Technology Inc. and its subsidiaries

35. Quad Serial Peripheral Interface (QSPI)

35.1. Overview

The Quad SPI Interface (QSPI) circuit is a synchronous serial data link that provides communication with external devices in Host mode.

The QSPI can be used in SPI mode to interface serial peripherals, such as ADCs, DACs, LCD controllers and sensors or in Serial Memory mode to interface serial Flash memories.

The QSPI allows the system to execute code directly from a serial Flash memory (XIP) without code shadowing to SRAM. The serial Flash memory mapping is visible in the system as other memories (ROM, SRAM, embedded Flash memories and so on). XIP mode is not available for a secured or code protected device for security reasons. This is achieved by blocking off AHB accesses (only instruction) on the CPU side from QSPI. QSPI returns an error on the transactions where the CPU is requesting instructions from QSPI-connected external memories.

With the support of the Quad-SPI protocol, the QSPI allows the system to use high-performance serial Flash memories that are small and inexpensive in place of larger and more expensive parallel Flash memories.

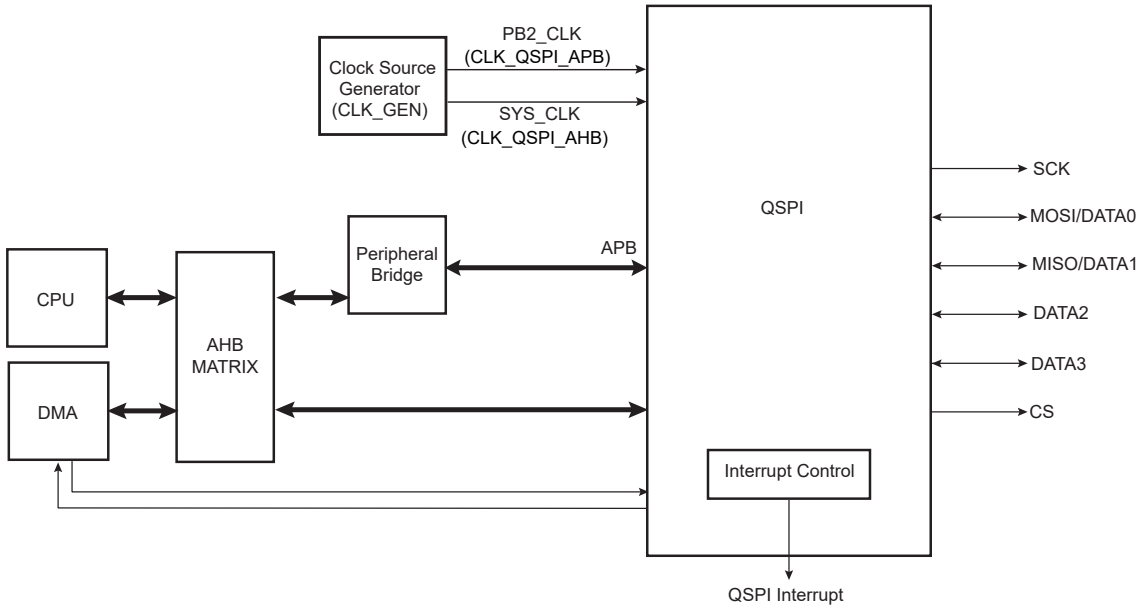
35.2. Features

- Host SPI Interface:
  - Programmable clock phase and clock polarity
  - Programmable transfer delays between consecutive transfers, between clock and data, between deactivation and activation of chip select
- SPI Mode:
  - To use serial peripherals, such as ADCs, DACs, LCD controllers, CAN controllers and sensors
  - 8-bit, 16-bit or 32-bit programmable data length
- Serial Memory Mode:
  - To use serial Flash memories operating in single-bit SPI, Dual SPI and Quad SPI
  - Supports Execute in Place (XIP). The system can execute code directly from a Serial Flash memory
  - Flexible instruction register to be compatible with all serial Flash memories
  - 32-bit Address mode (default is 24-bit address) to support serial Flash memories larger than 128 Mbit
  - Continuous Read mode
  - Scrambling/Unscrambling On-the-Fly
  - Double data rate support (Read only)
- Connection to DMA Channel Capabilities Optimizes Data Transfers
  - One channel for the receiver and one channel for the transmitter
- Register Write Protection

© 2025 Microchip Technology Inc. and its subsidiaries

35.3. Block Diagram

Figure 35-1. QSPI Block Diagram



35.4. Signal Description

Table 35-1. Quad-SPI Signals

Signal	Description	Type
SCK	Serial Clock	Output
CS	Chip Select	Output
MOSI(DATA0)	Data Output (Data Input Output 0)	Output (Input/Output)
MISO(DATA1)	Data Input (Data Input Output 1)	Input (Input/Output)
DATA2	Data Input Output 2	Input/Output
DATA3	Data Input Output 3	Input/Output

Notes:

1. MOSI and MISO are used for single-bit SPI operation.
2. DATA0-DATA1 are used for Dual SPI operation.
3. DATA0-DATA3 are used for Quad SPI operation.

For more details on the pin mapping of the QSPI peripheral, see *I/O Ports and Peripheral Pin Select (PPS)* from Related Links.

Related Links

I/O Ports and Peripheral Pin Select (PPS)

35.5. Product Dependencies

The following sections describe how the other parts of the system must be configured correctly to use this peripheral.

35.5.1. I/O Lines

Using the QSPI I/O lines requires the I/O pins to be configured using the System Configuration registers (see *System Configuration and Register Locking (CFG)* from Related Links) (QSPI_HSEN of

© 2025 Microchip Technology Inc. and its subsidiaries

CFGCON1/DEVCFG1 register) for direct or PPS. If QSPI pins are selected through PPS, the PPS registers have to be configured (see *I/O Ports and Peripheral Pin Select (PPS)* from Related Links).

If QSPI_HSEN = 1, QSPI uses dedicated pins.

If QSPI_HSEN = 0, QSPI uses PPS path and I/O pins are multiplexed to pins groups defined in PPS section.

Related Links

I/O Ports and Peripheral Pin Select (PPS)

System Configuration and Register Locking (CFG)

35.5.2. Power Management

The QSPI will continue to operate in any Sleep mode where the selected source clock is running. The QSPI interrupts can be used to wake up the device from sleep modes. See *Power Management Unit (PMU)* from Related Links for details on the different sleep modes.

Related Links

Power Management Unit (PMU)

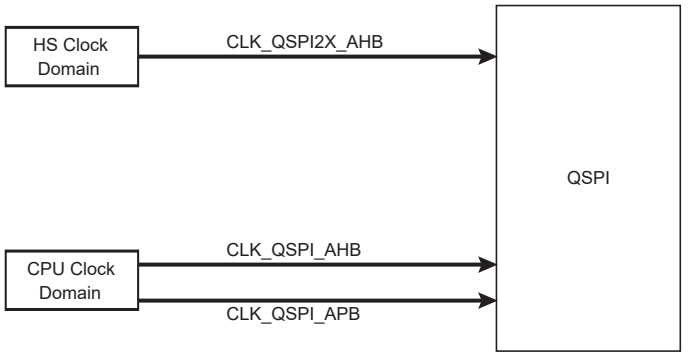
35.5.3. Clocks

An AHB clock (CLK_QSPI_AHB) is required to clock the QSPI. In PIC32CX-BZ6, SYS_CLK is the AHB clock and can be configured in the CRU.

A FAST clock (CLK_QSPI2X_AHB) is required to clock the QSPI. This clock can be enabled and disabled in the CFGCON1 register, bit 29 (CFGCON1.QSPIDDRM). When using QSPI DDR (Double Data Rate) mode, the System Clock (SYS_CLK) must be less than or equal to 64 MHz.

SYS_CLK is the CLK_QSPI_APB clock and can be configured in CRU registers.

Figure 35-2. QSPI Clock Organization



Important: The CLK_QSPI2x_AHB must be two times faster to CLK_QSPI_AHB when the QSPI is operated in the DDR mode. In Single Data Rate (SDR), the CLK_QSPI2x_AHB is not used.

The CLK_QSPI_APB, CLK_QSPI_AHB and CLK_QSPI2X_AHB, respectively, are all synchronous but can be divided by a prescaler.

35.5.4. DMA

The DMA request lines are connected to the DMA Controller (DMAC). Using the QSPI DMA requests requires the DMA Controller to be configured first.

© 2025 Microchip Technology Inc. and its subsidiaries

Note: DMAC write access must be 32-bit aligned. If a single byte is to be written in a 32-bit word, the rest of the word must be filled with 'ones'.

35.5.5. Interrupts

The interrupt request lines are connected to the interrupt controller. Using the QSPI interrupts requires the interrupt controller to be configured first. See *Nested Vector Interrupt Controller (NVIC)* from Related Links.

Related Links

Nested Vector Interrupt Controller (NVIC)

35.5.6. Events

Not applicable.

35.5.7. Debug Operation

When the CPU is halted in debug mode the QSPI continues normal operation. If the QSPI 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.

35.5.8. Register Access Protection

All registers with write access are optionally write protected by the PAC, except the following registers:

- Control A (CTRLA) register
- Transmit Data (TXDATA) register
- Interrupt Flag Status and Clear (INTFLAG) register
- Scrambling Key (SCRAMBKEY) register

PAC write protection is denoted by the PAC Write-Protection property in the register description.

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

35.6. Functional Description

35.6.1. Principle of Operation

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

The QSPI operates as a host. It initiates and controls all data transactions.

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

When receiving, the data is transferred to the RXDATA register, and the receiver is ready for a new character.

35.6.2. Basic Operation

35.6.2.1. Initialization

After Power-On Reset, this peripheral is enabled .

35.6.2.2. Enabling, Disabling and Resetting

The peripheral is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE).

The peripheral is disabled by writing a '0' to CTRLA.ENABLE.

The peripheral is reset by writing a '1' to the Software Reset bit (CTRLA.SWRST).

© 2025 Microchip Technology Inc. and its subsidiaries

35.6.3. Transfer Data Rate

By default, the QSPI module is enabled in single data rate mode. In this operating mode, the CLK_QSPI2X_AHB clock is not used and must be disabled.

The dual data rate operating mode is enabled by writing a '1' to the Double Data Rate Enable bit in the CFGCON1 register (CFGCON1.QSPIDDRM). This operating mode requires the CLK_QSPI2X_AHB clock and must be enabled before writing the DDREN bit.

35.6.4. Serial Clock Baud Rate

The QSPI Baud rate clock is generated by dividing the module clock (CLK_QSPI_AHB) by a value between 3 and 255.

This allows a maximum operating baud rate at up to Host Clock and a minimum operating baud rate of CLK_QSPI_AHB divided by 255.

At reset, BAUD = 0 and the user has to program a valid value before performing the first transfer.

35.6.5. Serial Clock Phase and Polarity

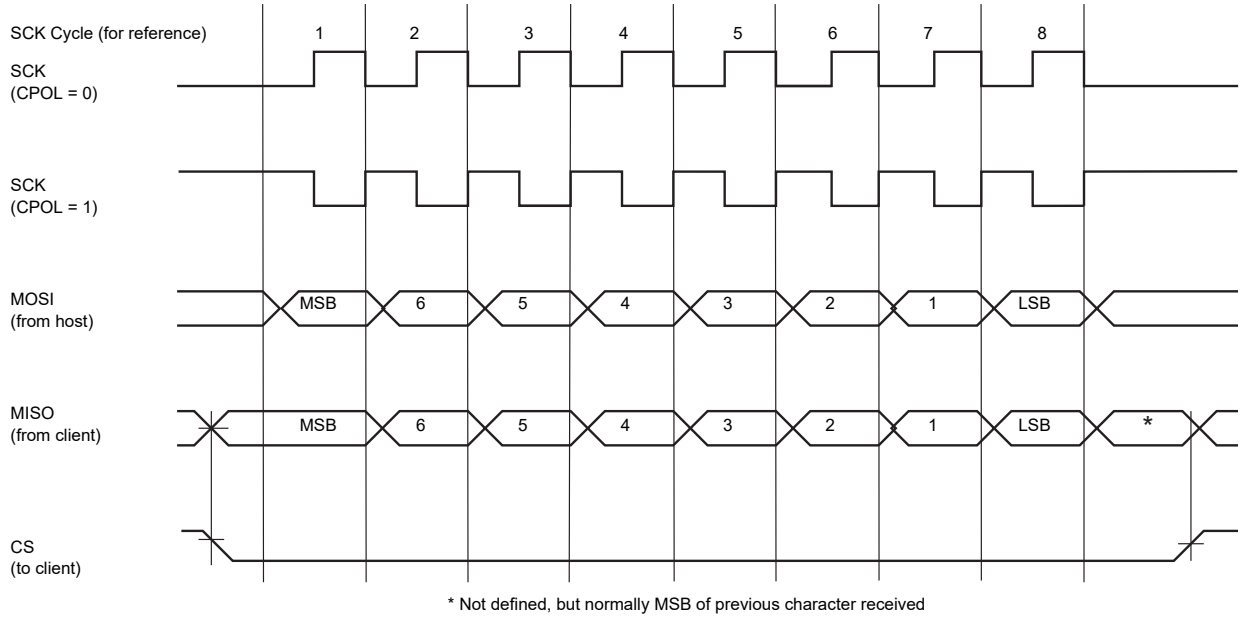
Four combinations of polarity and phase are available for data transfers. Writing the Clock Polarity bit in the QSPI Baud register (BAUD.CPOL) selects the polarity. The Clock Phase bit in the BAUD register programs the clock phase (BAUD.CPHA). These two parameters determine the edges of the clock signal where data is driven and sampled. Each of the two parameters has two possible states, resulting in four possible combinations.

Note: The polarity/phase combinations are incompatible. Thus, the interfaced client must use the same parameter values to communicate.

Table 35-2. SPI Transfer Mode

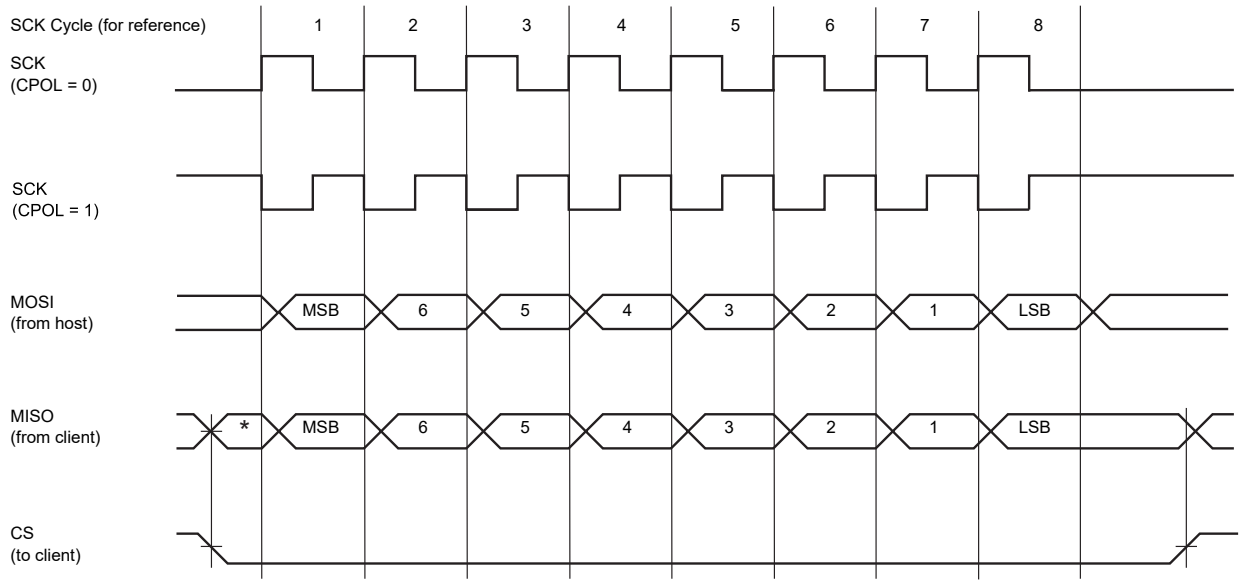
Clock Mode	BAUD.CPOL	BAUD.CPHA	Shift SCK Edge	Capture SCK Edge	SCK Inactive Level
0	0	0	Falling	Rising	Low
1	0	1	Rising	Falling	Low
2	1	0	Rising	Falling	High
3	1	1	Falling	Rising	High

Figure 35-3. QSPI Transfer Modes (BAUD.CPHA = 0, 8-bit transfer)



© 2025 Microchip Technology Inc. and its subsidiaries

Figure 35-4. QSPI Transfer Modes (BAUD.CPHA = 1, 8-bit transfer)



* Not defined, but normally LSB of previous character received

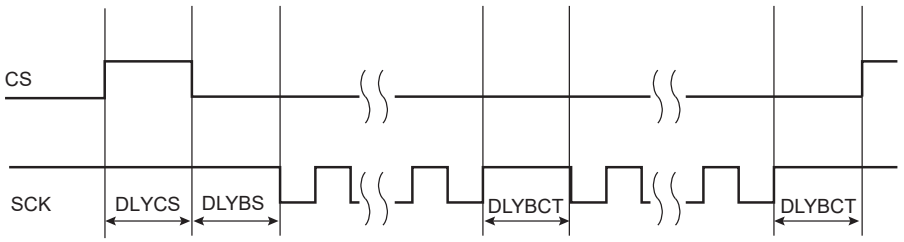
35.6.6. Transfer Delays

The QSPI supports several consecutive transfers while the chip select is active. Three delays can be programmed to modify the transfer waveforms:

- The delay between the inactivation and the activation of CS is programmed by writing the Minimum Inactive CS Delay bit field in the Control B register (CTRLB.DLYCS), allowing to tune the minimum time of CS at high level.
- The delay between consecutive transfers is programmed by writing the Delay Between Consecutive Transfers bit field in the Control B register (CTRLB.DLYBCT), allowing to insert a delay between two consecutive transfers. In Serial Memory mode, this delay is not programmable and DLYBCT settings are ignored.
- The delay before SCK is programmed by writing the Delay Before SCK bit field in the BAUD register (BAUD.DLYBS), allowing to delay the start of SPCK after the chip select has been asserted.

These delays allow the QSPI to be adapted to the interfaced peripherals and their speed and bus release time.

Figure 35-5. Programmable Delay



35.6.7. QSPI SPI Mode

In this mode, the QSPI acts as a regular SPI host.

To activate this mode, the MODE bit in the Control B register must be cleared (CTRLB.MODE = 0).

© 2025 Microchip Technology Inc. and its subsidiaries

35.6.7.1. SPI Mode Operations

The QSPI in standard SPI mode operates on the clock generated by the internal programmable baud rate generator. It fully controls the data transfers to and from the client connected to the SPI bus. The QSPI drives the chip select line to the client (CS) and the serial clock signal (SCK).

The QSPI features a single internal shift register and two holding registers: the Transmit Data Register (TXDATA) and the Receive Data Register (RXDATA). The holding registers maintain the data flow at a constant rate.

After enabling the QSPI, a data transfer begins when the processor writes to the TXDATA. The written data is immediately transferred into the internal shift register and transfer on the SPI bus starts. While the data in the internal shift register is shifted on the MOSI line, the MISO line is sampled and shifted into the internal shift register. Receiving data cannot occur without transmitting data.

If new data is written in TXDATA during the transfer, it stays in TXDATA until the current transfer is completed. Then, the received data is transferred from the internal shift register to the RXDATA, the data in TXDATA is loaded into the internal shift register, and a new transfer starts.

The transfer of data written in TXDATA in the internal shift register is indicated by the Transmit Data Register Empty (DRE) bit in the Interrupt Flag Status and Clear register (INTFLAG.DRE). When new data is written in TXDATA, this bit is cleared. The DRE bit is used to trigger the Transmit DMA channel.

The end of transfer is indicated by the Transmission Complete flag (INTFLAG.TXC). If the transfer delay for the last transfer was configured to be greater than 0 (CTRLB.DLYBCT), TXC is set after the completion of the delay. The module clock (CLK_QSPI_AHB) can be switched off at this time.

Ongoing transfer of received data from the internal shift register into RXDATA is indicated by the Receive Data Register Full flag (INTFLAG.RXC). When the received data is read, the RXC bit is cleared.

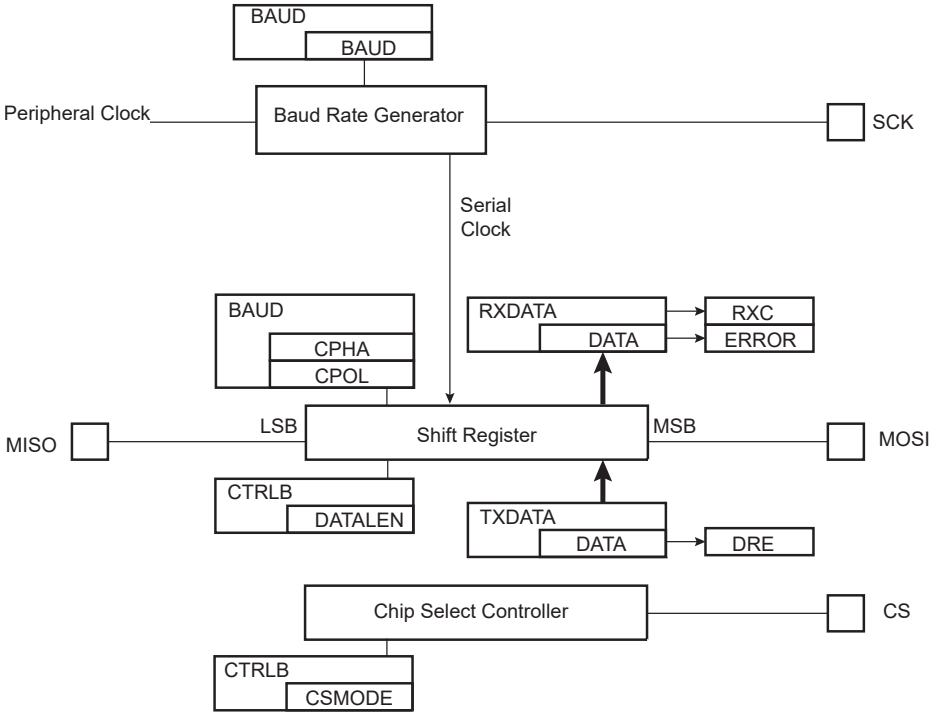
If the RXDATA has not been read before new data is received, the Overrun Error flag in INTFLAG register (INTFLAG.ERROR) is set. As long as this flag is set, data is loaded in RXDATA.

The SPI Mode Block Diagram shows a flow chart describing how transfers are handled.

© 2025 Microchip Technology Inc. and its subsidiaries

35.6.7.2. SPI Mode Block Diagram

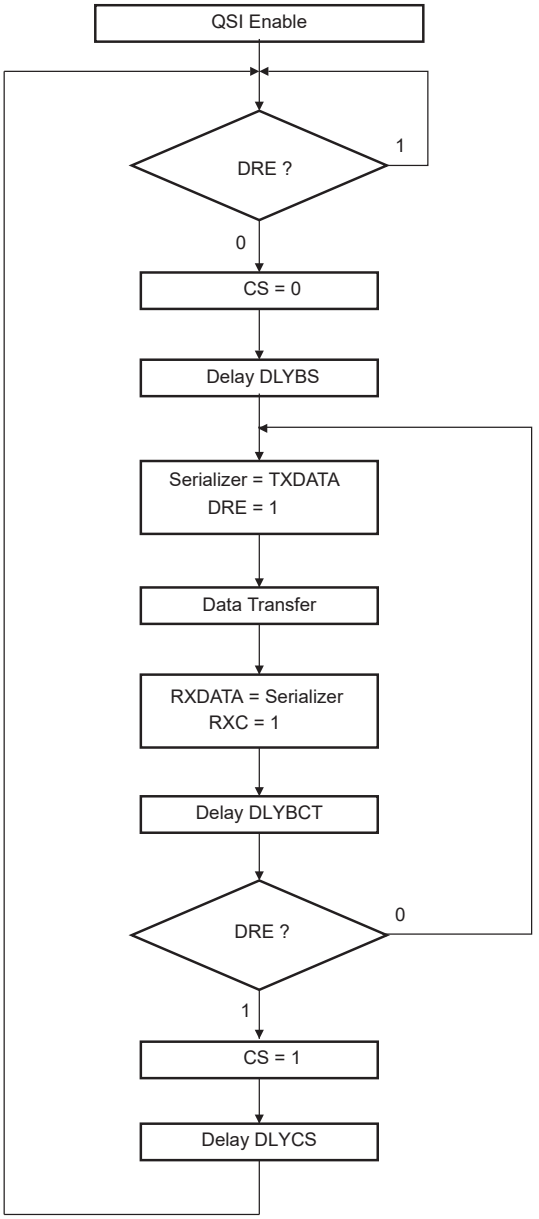
Figure 35-6. SPI Mode Block Diagram



© 2025 Microchip Technology Inc. and its subsidiaries

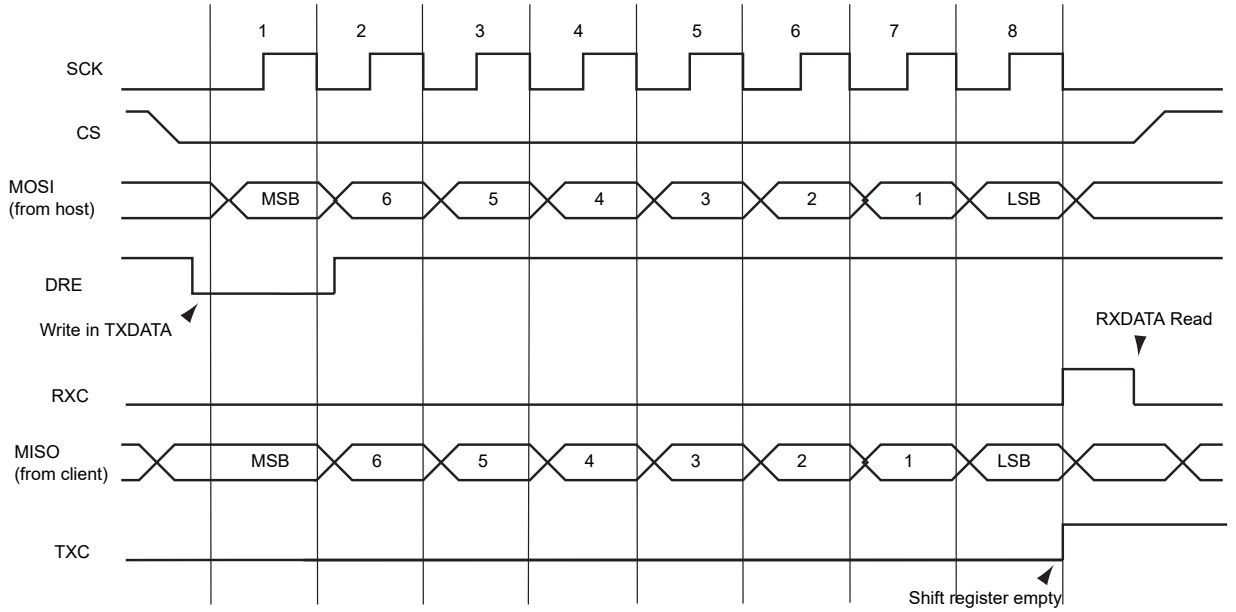
35.6.7.3. SPI Mode Flow Diagram

Figure 35-7. SPI Mode Flow Diagram



© 2025 Microchip Technology Inc. and its subsidiaries

Figure 35-8. Interrupt Flags Behaviour



35.6.7.4. Peripheral Deselection with DMA

When the Direct Memory Access Controller is used, the Chip Select line will remain low during the whole transfer since the Transmit Data Register Empty flag in the Interrupt Flag Status and Clear register (INTFLAG.DRE) is managed by the DMA itself. The reloading of the TXDATA by the DMA is done as soon as INTFLAG.DRE flag is set. In this case, setting the Chip Select Mode bit field in the Control B register (CTRLB.CSMODE) to 0x1 is not mandatory.

However, it may happen that when other DMA channels connected to other peripherals are in use as well, the QSPI DMA could be delayed by another DMA transfer with a higher priority on the bus. Having DMA buffers in slower memories like flash memory or SDRAM (compared to fast internal SRAM), may lengthen the reload time of the TXDATA by the DMA as well. This means that TXDATA might not be reloaded in time to keep the Chip Select line low. In this case the Chip Select line may toggle between data transfer and according to some SPI Client devices, and the communication might get lost. Writing CTRLB.CSMODE=0x1 can prevent this loss.

When CTRLB.CSMODE=0x0, the CS does not rise in all cases between two transfers on the same peripheral. During a transfer on a Chip Select, the INTFLAG.DRE flag is raised as soon as the content of the TXDATA is transferred into the internal shifter. When this flag is detected the TXDATA can be reloaded. if this reload occurs before the end of the current transfer and if the next transfer is performed on the same Chip Select as the current transfer, the Chip Select is not de-asserted between the two transfers. This may lead to difficulties for interfacing with some serial peripherals requiring the Chip Select to be de-asserted after each transfer. To facilitate interfacing with such devices, it is recommended to write CTRLB.CSMODE to 0x2.

35.6.7.5. Peripheral Deselection without DMA

During multiple data transfers on a Chip Select without the DMA, the TXDATA is loaded by the processor, and the Transmit Data Register Empty flag in the Interrupt Flag Status and Clear register (INTFLAG.DRE) rises as soon as the content of the RXDATA is transferred into the internal shift register. When this flag is detected high, the TXDATA can be reloaded. If this reload-by-processor occurs before the end of the current transfer and if the next transfer is performed on the same Chip Select as the current transfer, the Chip Select is not de-asserted between the two transfers.

Depending on the application software handling the flags or servicing other interrupts or other tasks, the processor may not reload the TXDATA in time to keep the Chip Select active (low). A null

© 2025 Microchip Technology Inc. and its subsidiaries

Delay Between Consecutive Transfer bit field value in the CTRLB register (CTRLB.DLYBCT) will give even less time for the processor to reload the TXDATA. With some SPI Client peripherals, requiring the Chip Select line to remain active (low) during a full set of transfers might lead to communication errors.

To facilitate interfacing with such devices, the Chip Select Mode bit field in the CTRLB register (CTRLB.CSMODE) can be written to 0x1. This allows the Chip Select lines to remain in their current state (low = active) until the end of transfer is indicated by the Last Transfer bit in the CTRLA register (CTRLA.LASTXFER). Even if the TXDATA is not reloaded the Chip Select will remain active. To have the Chip Select line rise at the end of the last data transfer, the LASTXFER bit in the CTRLA must be set before writing the last data to transmit into the TXDATA.

35.6.8. QSPI Serial Memory Mode

In this mode, the QSPI acts as a serial Flash memory controller. The QSPI can be used to read data from the serial Flash memory allowing the CPU to execute code from it (XIP execute in place). The QSPI can also be used to control the serial Flash memory (Program, Erase, Lock and so on) by sending specific commands. In this mode, the QSPI is compatible with single-bit SPI, Dual-SPI and Quad-SPI protocols.

To activate this mode, the MODE bit in Control B register must be set to one (CTRLB.MODE = 1).

In serial memory mode, data cannot be transferred by the TXDATA and the RXDATA but by writing or reading the QSPI memory space (0x0400 0000–0x0500 0000).

Caching can be enabled using the CMCC module along with configuring the CFGCON1.QSCHE_EN bit.

Important: The QSPI memory space region can be cached to improve data transfer speed.
However, external Flash devices that have command/status registers mapped in the QSPI memory space region must be managed carefully by applying any one of the following configurations:

- The data cache must be disabled.
- If the data cache is required, then the cache line must be invalidated before reading the status register.

35.6.8.1. Instruction Frame

To control serial Flash memories, the QSPI is able to send instructions by the SPI bus (for example, READ, PROGRAM, ERASE, LOCK and so on). The instruction set implemented in serial Flash memories is memory vendor dependent; therefore, the QSPI includes a complete instruction registers, which makes it very flexible and compatible with all serial Flash memories.

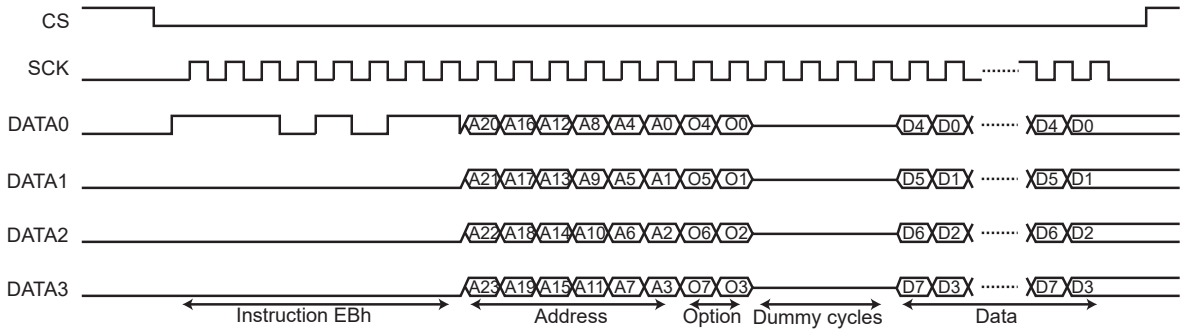
An instruction frame includes:

- **An instruction code (size: 8 bits):** The instruction can be optional in some cases.
- **An address (size: 24 bits or 32 bits):** The address is optional but is required by instructions such as READ, PROGRAM, ERASE, LOCK. By default, the address is 24 bits long but it can be 32 bits long to support serial Flash memories larger than 128 Mbit (16 Mbyte).
- **An option code (size: 1/2/4/8 bits):** The option code is optional but is useful for activating the XIP mode or the Continuous Read mode for READ instructions in some serial Flash memory devices. These modes permit data read latency improvement.
- **Dummy cycles:** Dummy cycles are optional but required by some READ instructions.
- **Data bytes are optional:** Data bytes are present for data transfer instructions such as READ or PROGRAM.

© 2025 Microchip Technology Inc. and its subsidiaries

The instruction code, the address/option and the data can be sent with Single-bit SPI, Dual SPI or Quad SPI protocols.

Figure 35-9. Instruction Frame



35.6.8.2. Instruction Frame Sending

To send an instruction frame, the user must first configure the address to send by writing the field ADDR in the Instruction Address Register (INSTRADDR.ADDR). This step is required if the instruction frame includes an address and no data. When data are present, the address of the instruction is defined by the address of the data accesses in the QSPI memory space and not by the INSTRADDR register.

If the instruction frame includes the instruction code and/or the option code, the user must configure the instruction code and/or the option code to send by writing the INST and OPTCODE bit fields in the Instruction Control Register (INSTRCTRL.OPTCODE, INSTRCTRL.INSTR).

Then, the user must write the Instruction Frame Register (INSTRFRAME) to configure the instruction frame depending on which instruction must be sent. If the instruction frame does not include data, writing in this register triggers the send of the instruction frame in the QSPI. If the instruction frame includes data, the send of the instruction frame is triggered by the first data access in the QSPI memory space.

The instruction frame is configured by the following bits and fields of INSTRFRAME:

- The WIDTH field is used to configure which data lanes are used to send the instruction code, the address, the option code and to transfer the data. It is possible to use two unidirectional data lanes (MISO-MOSI Single-bit SPI), two bidirectional data lanes (DATA0 - DATA1 Dual SPI) or four bidirectional data lanes (DATA0 – DATA3).

Table 35-3. WIDTH Encoding

INSTRFRAME	Instruction	Address/Option	Data
0	Single-bit SPI	Single-bit SPI	Single-bit SPI
1	Single-bit SPI	Single-bit SPI	Dual SPI
2	Single-bit SPI	Single-bit SPI	Quad SPI
3	Single-bit SPI	Dual SPI	Dual SPI
4	Single-bit SPI	Quad SPI	Quad SPI
5	Dual SPI	Dual SPI	Dual SPI
6	Quad SPI	Quad SPI	Quad SPI
7	Reserved		

- The INSTREN bit enables sending an instruction code
- The ADDREN bit enables sending of an address after the instruction code
- The OPTCODEEN bit enables sending of an option code after the address
- The DATAEN bit enables the transfer of data (READ or PROGRAM instruction)

© 2025 Microchip Technology Inc. and its subsidiaries

- The OPTCODELEN field configures the option code length (0 -> 1-bit / 1 -> 2-bit / 2 -> 4-bit / 3 -> 8-bit). The value written in OPTCODELEN must be consistent with the value written in the field WIDTH. For example, OPTCODELEN = 0 (1-bit option code) is not coherent with WIDTH = 6 (option code sent with QuadSPI protocol, thus the minimum length of the option code is 4-bit).
- The ADDRLEN bit configures the address length (0 -> 24 bits / 1-> 32 bits)
- The TFRTYPE field defines which type of data transfer must be performed
- The DUMMYLEN field configures the number of dummy cycles when reading data from the serial Flash memory. Between the address/option and the data, with some instructions, dummy cycles are inserted by the serial Flash memory.

If data transfer is enabled, the user can access the serial memory by reading or writing the QSPI memory space following these rules:

- Reading from the serial memory, but not memory data (for example reading the JEDEC-ID or the STATUS), requires TFRTYPE to be written to 0x0
- Reading from the serial memory, particularly memory data, requires TFRTYPE to be written to '1'
- Writing to the serial memory, but not memory data (for example, writing the configuration or STATUS), requires TFRTYPE to be written to 0x2
- Writing to the serial memory, particularly memory data, requires TFRTYPE to be written to 0x3

If TFRTYP has a value other than 0x1 and CTRLB.SMEMREG = 0, the address sent in the instruction frame is the address of the first system bus accesses. The addresses of the subsequent access actions are not used by the QSPI. At each system bus access, an SPI transfer is performed with the same size. For example, a half-word system bus access leads to a 16-bit SPI transfer and a byte system bus access leads to an 8-bit SPI transfer.

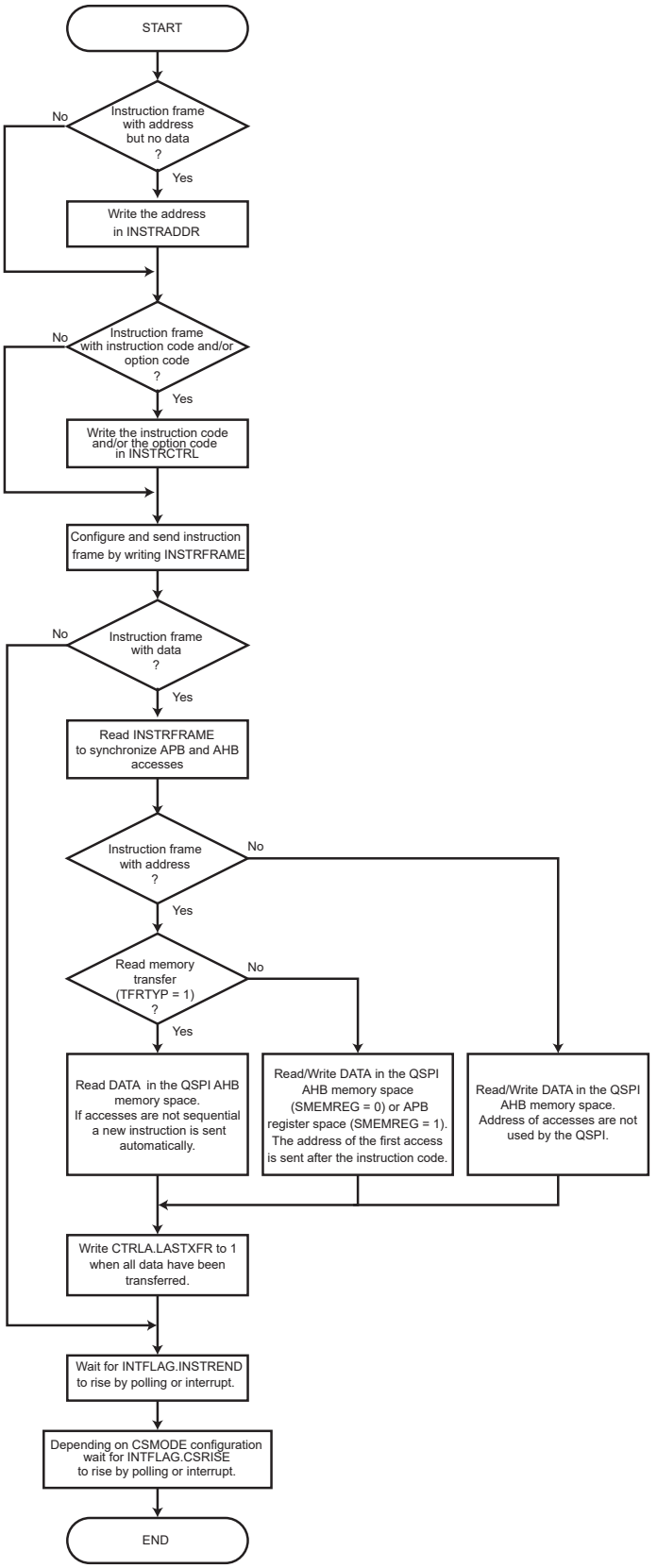
If CTRLB.SMEMREG = 1, accesses are made via the QSPI registers and the address sent in the instruction frame is the address defined in the INSTRADDR register. Each time the INSTRFRAME or TXDATA registers are written, an SPI transfer is performed with a byte size. Another byte is read each time the RXDATA register is read or written each time the TXDATA register is written. The SPI transfer ends by writing the LASTXFER bit in the Control A register (CTRLA.LASTXFER).

If TFRTYP = 0x1, the address of the first instruction frame is one of the first read access in the QSPI memory space. Each time the read accesses becomes non-sequential (addresses are not consecutive), a new instruction frame is sent with the last system bus access address. In this way, the system can read data at a random location in the serial memory. The size of the SPI transfers may differ from the size of the system bus read accesses.

When data transfer is not enabled, the end of the instruction frame is indicated when the INSTREND interrupt flag in the INTFLAG register is set. When data transfer is enabled, the user must indicate when the data transfer is complete in the QSPI memory space by setting the bit LASTXFR in the CTRLA. The end of the instruction frame is indicated when the INSTREND interrupt flag in the INTFLAG register is set.

© 2025 Microchip Technology Inc. and its subsidiaries

Figure 35-10. Instruction Transmission Flow Diagram



© 2025 Microchip Technology Inc. and its subsidiaries

35.6.8.3. Read Memory Transfer

The user can access the data of the serial memory by sending an instruction with DATAEN = 1 and TFRTYP = 0x1 in the Instruction Frame register (INSTRFRAME).

In this mode, the QSPI is able to read data at a random address into the serial Flash memory, allowing the CPU to execute code directly from it (XIP execute-in-place).

To fetch data, the user must first configure the instruction frame by writing the INSTRFRAME. Then, data can be read at any address in the QSPI address space mapping. The address of the system bus read accesses matches the address of the data inside the serial Flash memory.

When Fetch mode is enabled, several instruction frames can be sent before writing the bit LASTXFR in the CTRLA. Each time the system bus read accesses become non-sequential (addresses are not consecutive), a new instruction frame is sent with the corresponding address.

35.6.8.4. Continuous Read Mode

The QSPI is compatible with Continuous Read Mode (CRM), which is implemented in some Serial Flash memories.

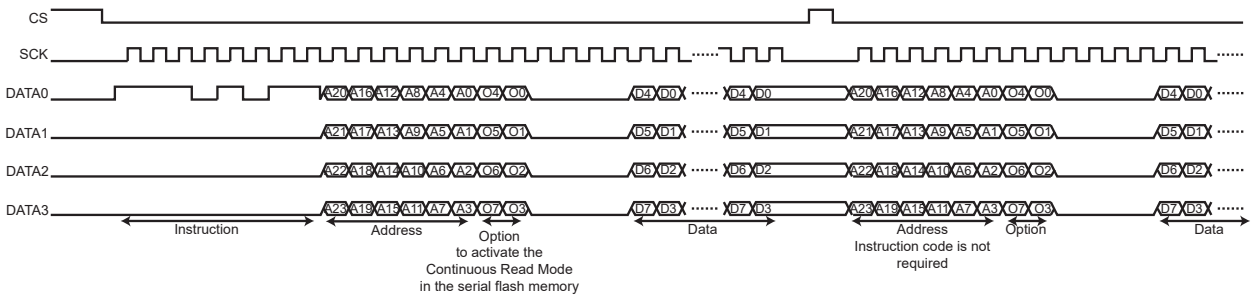
The CRM provides reduction in instruction overhead by excluding the instruction code from the instruction frame. When CRM is activated in a Serial Flash memory (by a specific option code), the instruction code is stored in the memory. For the next instruction frames, the instruction code is not required, as the memory uses the stored one.

In the QSPI, CRM is used when reading data from the memory (INSTFRAME.TFRTYPE = 0x1). The addresses of the system bus read accesses are often non-sequential; this leads to many instruction frames with the same instruction code. By disabling the sending of the instruction code, the CRM reduces the access time of the data.

To be functional, this mode must be enabled in both the QSPI and the Serial Flash memory. The CRM is enabled in the QSPI by setting the CRM bit in the INSTRFRAME register (INSTFRAME.CRMODE = 1, INSTFRAME.TFRTYPE must be 0x1). The CRM is enabled in the Serial Flash memory by sending a specific option code.

CAUTION If the CRM is not supported by the Serial Flash memory or disabled, the CRMODE bit must not be set. Otherwise, data read out from the Serial Flash memory is not valid.

Figure 35-11. Continuous Read Mode



35.6.8.5. Instruction Frame Transmission Examples

All waveforms in the following examples describe SPI transfers in SPI Clock mode 0 (BAUD.CPOL = 0 and BAUD.CPHA = 0). All system bus accesses described below refer to the system bus address phase. System bus wait cycles and system bus data phases are not shown.

© 2025 Microchip Technology Inc. and its subsidiaries

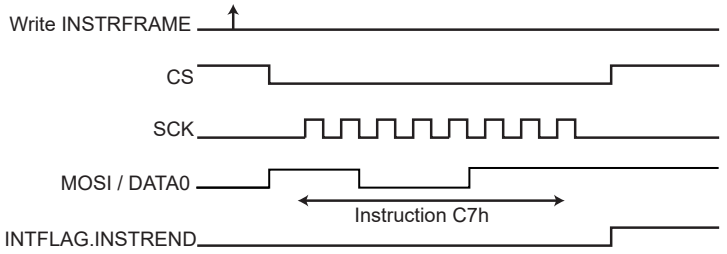
Example 35-1. Scenario 1

Instruction in Single-bit SPI, without address, without option, without data.

Command: CHIP ERASE (C7h).

- Write 0x0000_00C7 to INSTRCTRL register.
- Write 0x0000_0010 to INSTRFRAME register.
- Wait for INTFLAG.INSTREND to rise.

Figure 35-12. Instruction Transmission Waveform 1



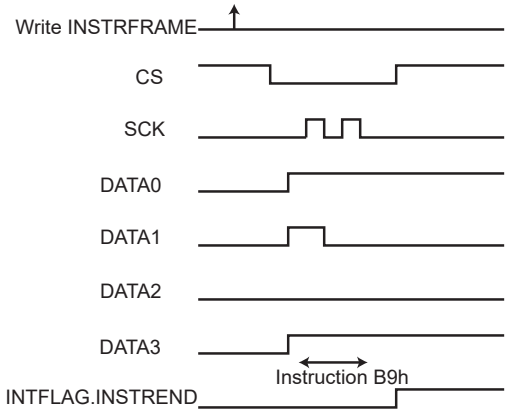
Example 35-2. Scenario 2

Instruction in Quad SPI, without address, without option, without data.

Command: POWER DOWN (B9h)

- Write 0x0000_00B9 to INSTRCTRL register.
- Write 0x0000_0016 to INSTRFRAME register.
- Wait for INTFLAG.INSTREND to rise.

Figure 35-13. Instruction Transmission Waveform 2



Example 35-3. Scenario 3

Instruction in Single-bit SPI, with address in Single-bit SPI, without option, without data.

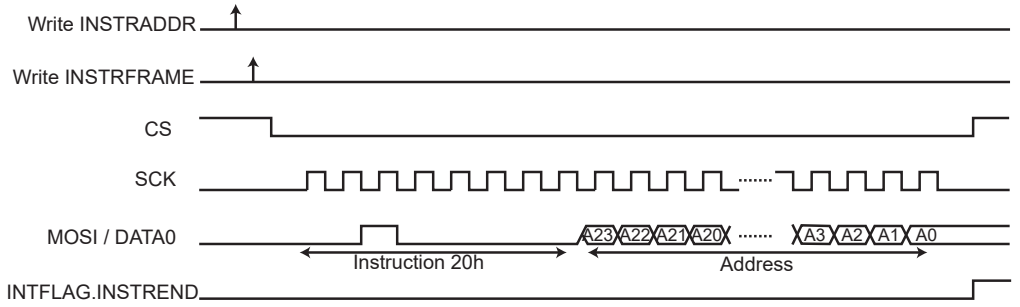
Command: BLOCK ERASE (20h)

- Write the address (of the block to erase) to QSPI_AR.
- Write 0x0000_0020 to INSTRCTRL register.

© 2025 Microchip Technology Inc. and its subsidiaries

- Write 0x0000_0030 to INSTRFRAME register.
- Wait for INTFLAG.INSTREND to rise.

Figure 35-14. Instruction Transmission Waveform 3



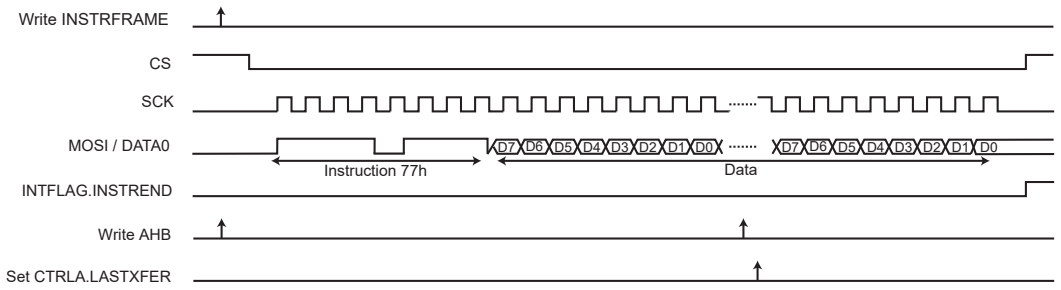
Example 35-4. Scenario 4

Instruction in Single-bit SPI, without address, without option, with data write in Single-bit SPI.

Command: SET BURST (77h)

- Write 0x0000_0077 to INSTRCTRL register.
- Write 0x0000_2090 to INSTRFRAME register.
- Read INSTRFRAME register (dummy read) to synchronize system bus accesses.
- Write data to the system bus memory space (0x0400_0000–0x0500_0000). The address of the system bus write accesses is not used.
- Write the LASTXFR bit in CTRLA register to '1'.
- Wait for INTFLAG.INSTREND to rise.

Figure 35-15. Instruction Transmission Waveform 4



Example 35-5. Scenario 5

Instruction in Single-bit SPI, with address in Dual SPI, without option, with data write in Dual SPI.

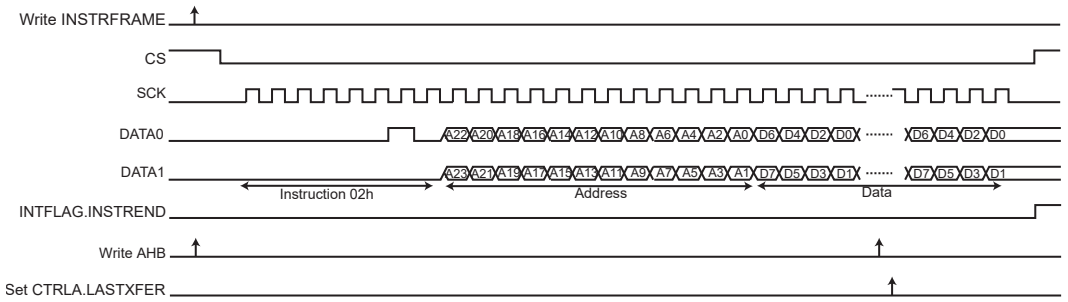
Command: BYTE/PAGE PROGRAM (02h)

- Write 0x0000_0002 to INSTRCTRL register.
- Write 0x0000_30B3 to INSTRFRAME register.
- Read INSTRFRAME register (dummy read) to synchronize system bus accesses.

© 2025 Microchip Technology Inc. and its subsidiaries

- Write data to the QSPI system bus memory space (0x040_00000–0x0500_0000). The address of the first system bus write access is sent in the instruction frame. The address of the next system bus write accesses is not used.
- Write LASTXFR bit in CTRLA register to '1'.
- Wait for INTFLAG.INSTREND to rise.

Figure 35-16. Instruction Transmission Waveform 5



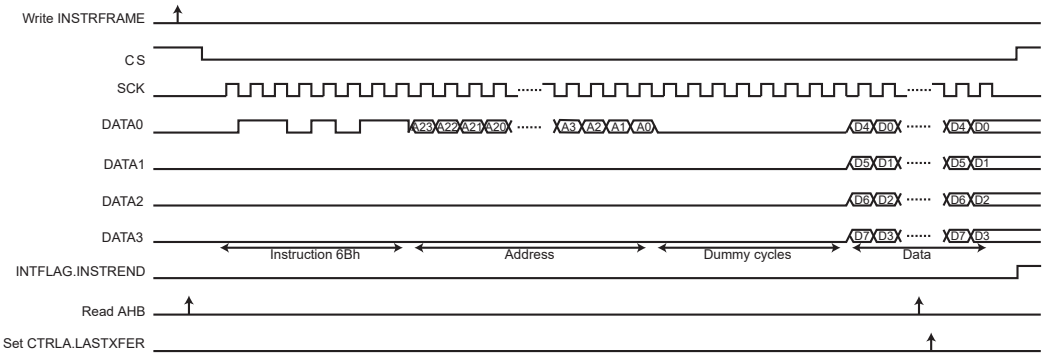
Example 35-6. Scenario 6

Instruction in Single-bit SPI, with address in Single-bit SPI, without option, with data read in Quad SPI, with eight dummy cycles.

Command: QUAD_OUTPUT READ ARRAY (6Bh)

- Write 0x0000_006B to INSTRCTRL register.
- Write 0x0008_10B2 to INSTRFRAME register.
- Read QSPI_IR (dummy read) to synchronize system bus accesses.
- Read data from the QSPI system bus memory space (0x040_00000–0x0500_0000). The address of the first system bus read access is sent in the instruction frame. The address of the next system bus read accesses is not used.
- Write the LASTXFR bit in CTRLA register to '1'.
- Wait for INTFLAG.INSTREND to rise.

Figure 35-17. Instruction Transmission Waveform 6



© 2025 Microchip Technology Inc. and its subsidiaries

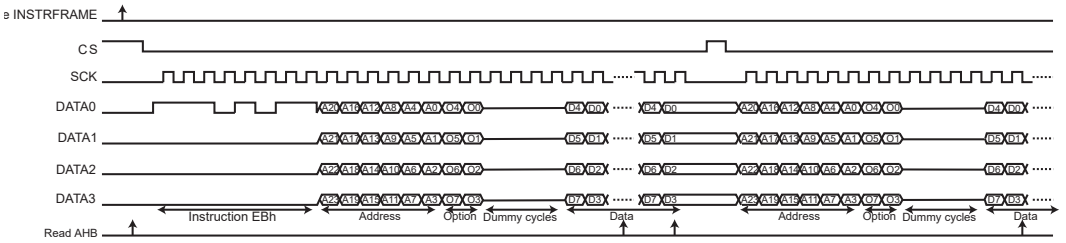
Example 35-7. Scenario 7

Instruction in Single-bit SPI, with address and option in Quad SPI, with data read from Quad SPI, with four dummy cycles, with fetch and continuous read.

Command: FAST READ QUAD I/O (EBh) - 8-BIT OPTION (0x30h)

- Write 0x0030_00EB to INSTRCTRL register.
- Write 0x0004_33F4 to INSTRFRAME register.
- Read INSTRFRAME register (dummy read) to synchronize system bus accesses.
- Read data from the QSPI system bus memory space (0x040_00000–0x0500_0000). Fetch is enabled, the address of the system bus read accesses is always used.
- Write LASTXFR bit in CTRLA register to '1'.
- Wait for INTFLAG.INSTREND to rise.

Figure 35-18. Instruction Transmission Waveform 7



Example 35-8. Scenario 8

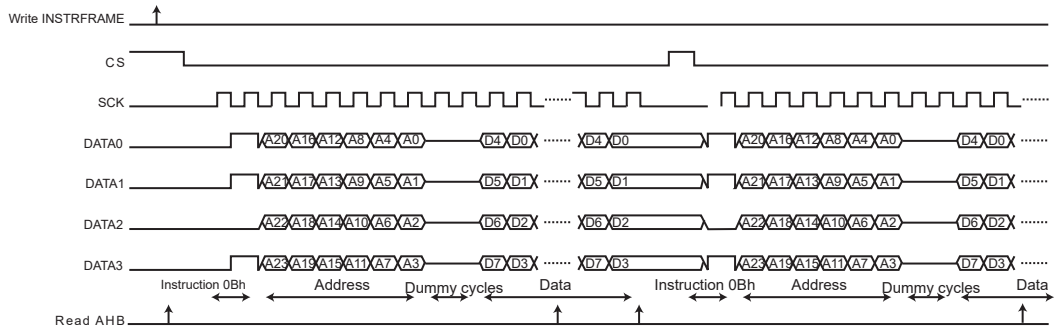
Instruction in Quad SPI, with address in Quad SPI, without option, with data read from Quad SPI, with two dummy cycles, with fetch.

Command: HIGH-SPEED READ (0Bh)

- Write 0x0000_000B to INSTRCTRL register.
- Write 0x0002_20B6 to INSTRFRAME register.
- Read INSTRFRAME register (dummy read) to synchronize system bus accesses.
- Read data in the QSPI system bus memory space (0x040_00000–0x0500_0000). Fetch is enabled, the address of the system bus read accesses is always used.
- Write LASTXFR bit in CTRLA register to '1'.
- Wait for INTFLAG.INSTREND to rise.

© 2025 Microchip Technology Inc. and its subsidiaries

Figure 35-19. Instruction Transmission Waveform 8



35.6.9. Scrambling/Unscrambling Function

The scrambling/unscrambling function cannot be performed on devices other than memories. Data are scrambled when written to memory and unscrambled when data are read.

The external data lines can be scrambled to prevent intellectual property data located in off-chip memories from being easily recovered by analyzing data at the package pin level of either the micro-controller or the QSPI client device (for example, memory).

The scrambling/unscrambling function can be enabled by writing a '1' to the ENABLE bit in the Scrambling Control register (SCRAMBCTRL.ENABLE).

The scrambling and unscrambling are performed on-the-fly without impacting the throughput.

The scrambling method depends on the user-configurable Scrambling User Key in the Scrambling Key register (SCRAMBKEY.KEY). This register is only accessible in the Write mode.

By default, the scrambling and unscrambling algorithm includes the scrambling user key, plus a device-dependent random value. This random value is not included when the Scrambling/Unscrambling Random Value Disable bit in the Scrambling Mode register (SCRAMBCTRL.RANDOMDIS) is written to '1'.

The random value is neither user-configurable nor readable. If SCRAMBCTRL.RANDOMDIS = 0, data scrambled by a given circuit cannot be unscrambled by a different circuit.

If SCRAMBCTRL.RANDOMDIS = 1, the scrambling/unscrambling algorithm includes only the scrambling user key, making it possible to manage data by different circuits.

The scrambling user key must be securely stored in a reliable Non-Volatile Memory to recover data from the off-chip memory. Any data scrambled with a given key cannot be recovered if the key is lost.

35.6.10. DMA Operation

The QSPI generates the following DMA requests:

- Data received (RX): The request is set when data is available in the RXDATA register, and cleared when RXDATA is read.
- Data transmit (TX): The request is set when the transmit buffer (TXDATA) is empty, and cleared when TXDATA is written.

Note: If DMA and RX memory modes are selected, a QSPI memory space read operation is required to force the first triggering.

If the CPU accesses the registers which are source of DMA request set/clear condition, the DMA request can be lost or the DMA transfer can be corrupted.

© 2025 Microchip Technology Inc. and its subsidiaries

35.6.11. Interrupts

The QSPI has the following interrupt source:

- Interrupt Request (INTREQ) – Indicates that at least one bit in the Interrupt Flag Status and Clear register (INTFLAG) is set to '1'

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set (INTENSET) register and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register. An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled or the QSPI is reset. All interrupt requests from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. The user must read the INTFLAG register to determine which interrupt condition is present.

Note: Interrupts must be globally enabled for interrupt requests to be generated.

© 2025 Microchip Technology Inc. and its subsidiaries

35.7. Register Summary

See the *QSPI* module in the *Product Memory Mapping Overview* from Related Links for the base address.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0							ENABLE	SWRST
		15:8								
		23:16								
		31:24								LASTXFER
0x04	CTRLB	7:0			CSMODE[1:0]		SMEMREG	WDRBT	LOOPEN	MODE
		15:8					DATALEN[3:0]			
		23:16	DLYBCT[7:0]							
		31:24	DLYCS[7:0]							
0x08	BAUD	7:0							CPHA	CPOL
		15:8	BAUD[7:0]							
		23:16	DLYBS[7:0]							
		31:24								
0x0C	RXDATA	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16								
		31:24								
0x10	TXDATA	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16								
		31:24								
0x14	INTENCLR	7:0					ERROR	TXC	DRE	RXC
		15:8						INSTREND		CSRISE
		23:16								
		31:24								
0x18	INTENSET	7:0					ERROR	TXC	DRE	RXC
		15:8						INSTREND		CSRISE
		23:16								
		31:24								
0x1C	INTFLAG	7:0					ERROR	TXC	DRE	RXC
		15:8						INSTREND		CSRISE
		23:16								
		31:24								
0x20	STATUS	7:0							ENABLE	
		15:8							CSSTATUS	
		23:16								
		31:24								
0x24 ... 0x2F	Reserved									
0x30	INSTRADDR	7:0	ADDR[7:0]							
		15:8	ADDR[15:8]							
		23:16	ADDR[23:16]							
		31:24	ADDR[31:24]							
0x34	INSTRCTRL	7:0	INSTR[7:0]							
		15:8								
		23:16	OPTCODE[7:0]							
		31:24								
0x38	INSTRFRAME	7:0	DATAEN	OPTCODEEN	ADDREN	INSTREN		WIDTH[2:0]		
		15:8	DDREN	CRMODE	TFRTYPE[1:0]			ADDRLEN	OPTCODELEN[1:0]	
		23:16				DUMMYLEN[4:0]				
		31:24								
0x3C ... 0x3F	Reserved									

© 2025 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x40	SCRAMBCTRL	7:0							RANDOMDIS	ENABLE
		15:8								
		23:16								
		31:24								
0x44	SCRAMBKEY	7:0	KEY[7:0]							
		15:8	KEY[15:8]							
		23:16	KEY[23:16]							
		31:24	KEY[31:24]							

35.8. Register Description

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

Optional write protection by the PAC is denoted by the PAC Write Protection property in each individual register description.

See *Peripheral Access Controller (PAC)* from Related Links.

Some registers are enable-protected, meaning they can only be written when the QSPI is disabled. Enable protection is denoted by the Enable-protected property in each individual register description.

Related Links

Peripheral Access Controller (PAC)

© 2025 Microchip Technology Inc. and its subsidiaries

35.8.1. Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
								LASTXFER
Access								W
Reset								0

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							ENABLE	SWRST
Access							W	W
Reset							0	0

Bit 24 – LASTXFER Last Transfer

Value	Description
0	No effect.
1	The chip select is de-asserted after the character written in TD has been transferred.

Bit 1 – ENABLE Enable

Writing a '0' to this bit disables the QSPI.
Writing a '1' to this bit enables the QSPI to transfer and receive data.
As soon as ENABLE is reset, QSPI finishes its transfer.
All pins are set in Input mode and no data are received or transmitted.
If a transfer is in progress, the transfer is finished before the QSPI is disabled.

Bit 0 – SWRST Software Reset

Writing a '0' to this bit has no effect.
Writing a '1' to this bit resets the QSPI. A software-triggered hardware Reset of the QSPI interface is performed.
DMAC channels are not affected by software Reset.

© 2025 Microchip Technology Inc. and its subsidiaries

35.8.2. Control B

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

Bit	31	30	29	28	27	26	25	24
	DLYCS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

Bit	15	14	13	12	11	10	9	8
					DATALEN[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
			CSMODE[1:0]		SMEMREG	WDRBT	LOOPEN	MODE
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 31:24 – DLYCS[7:0] Minimum Inactive CS Delay
This bit field defines the minimum delay between the inactivation and the activation of CS. The DLYCS time guarantees the client minimum deselect time.
If DLYCS is 0x00, one CLK_QSPI_AHB period is inserted by default.
Otherwise, the following equation determines the delay:
DLYCS = Minimum inactive × fperipheral clock

Bits 23:16 – DLYBCT[7:0] Delay Between Consecutive Transfers
This field defines the delay between two consecutive transfers with the same peripheral without removing the chip select. The delay is always inserted after each transfer and before removing the chip select if needed.
When DLYBCT = 0x00, no delay between consecutive transfers is inserted and the clock keeps its duty cycle over the character transfers. In Serial Memory mode (MODE = 1), DLYBCT is ignored and no delay is inserted. Otherwise, the following equation determines the delay:
DLYBCT = (Delay Between Consecutive Transfers × fperipheral clock)/32

Bits 11:8 – DATALEN[3:0] Data Length
The DATALEN field determines the number of data bits transferred. Reserved values must not be used.

Value	Name	Description
0x0	8BITS	8 bits transfer
0x1	9BITS	9 bits transfer
0x2	10BITS	10 bits transfer
0x3	11BITS	11 bits transfer
0x4	12BITS	12 bits transfer
0x5	13BITS	13 bits transfer

© 2025 Microchip Technology Inc. and its subsidiaries

Value	Name	Description
0x6	14BITS	14 bits transfer
0x7	15BITS	15 bits transfer
0x8	16BITS	16 bits transfer
0x9-0xF	—	Reserved

Bits 5:4 – CSMODE[1:0] Chip Select Mode

The CSMODE field determines how the chip select is de-asserted.

Value	Name	Description
0x0	NORELOAD	The chip select is de-asserted if TD was not reloaded before the end of the current transfer.
0x1	LASTXFER	The chip select is de-asserted when the bit LASTXFER is written at '1' and the character written in TD was transferred.
0x2	SYSTEMATICALLY	The chip select is de-asserted systematically after each transfer.
0x3	—	Reserved

Bit 3 – SMEMREG Serial Memory Register Mode

Value	Description
0	Serial memory registers are written via AHB access.
1	Serial memory registers are written via APB access. Reset the QSPI.

Bit 2 – WDRBT Wait Data Read Before Transfer

This bit determines the Wait Data Read Before Transfer option.

Bit 1 – LOOPEN Local Loopback Enable

This bit defines if the Local Loopback is enabled or disabled.
LOOPEN controls the local loopback on the data serializer for testing in SPI Mode only. (MISO is internally connected on MOSI).

Value	Description
0	Local Loopback is disabled.
1	Local Loopback is enabled.

Bit 0 – MODE Serial Memory Mode

This bit defines if the QSPI is in SPI mode or Serial Memory mode.

Value	Name	Description
0	SPI	SPI operating mode
1	MEMORY	Serial Memory operating mode

© 2025 Microchip Technology Inc. and its subsidiaries

35.8.3. Baud Rate

Name: BAUD
Offset: 0x08
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
	DLYBS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	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

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

Bits 23:16 – DLYBS[7:0] Delay Before SCK
This field defines the delay from CS valid to the first valid SCK transition.
When DLYBS equals zero, the CS valid to SCK transition is 1/2 the SCK clock period.
Otherwise, the following equation determines the delay:

Equation 35-1. Delay Before SCK

$$Delay\ \ Before\ SCK = \frac{DLYBS}{MCK}$$

Bits 15:8 – BAUD[7:0] Serial Clock Baud Rate
The QSPI uses a modulus counter to derive the SCK baud rate from the module clock (MCK) CLK_QSPI_AHB. The Baud rate is selected by writing a value from 1 to 255 in the BAUD field. The following equation determines the SCK baud rate:

Equation 35-2. SCK Baud Rate

$$SCK\ \ Baud\ \ Rate = \frac{MCK}{(BAUD + 1)}$$

Bit 1 – CPHA Clock Phase
CPHA determines which edge of SCK causes data to change and which edge causes data to be captured. CPHA is used with CPOL to produce the required clock/data relationship between host and client devices.

Value	Description
0	Data are captured on the leading edge of SCK and changed on the following edge of SCK.
1	Data are changed on the leading edge of SCK and captured on the following edge of SCK.

© 2025 Microchip Technology Inc. and its subsidiaries

Bit 0 – CPOL Clock Polarity

CPOL is used to determine the inactive state value of the serial clock (SCK). It is used with CPHA to produce the required clock/data relationship between host and client devices.

Value	Description
0	The inactive state value of SCK is logic level '0'.
0	The inactive state value of SCK is logic level '1'.

© 2025 Microchip Technology Inc. and its subsidiaries

35.8.4. Receive Data

Name: RXDATA
Offset: 0x0C
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

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

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

Bits 15:0 – DATA[15:0] Receive Data
Data received by the QSPI is stored in this register right-justified. Unused bits read zero.

© 2025 Microchip Technology Inc. and its subsidiaries

35.8.5. Transmit Data

Name: TXDATA
Offset: 0x10
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

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

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

Bits 15:0 – DATA[15:0] Transmit Data
Data to be transmitted by the QSPI is stored in this register. Information to be transmitted must be written to the transmit data register in a right-justified format.

© 2025 Microchip Technology Inc. and its subsidiaries

35.8.6. Interrupt Enable Clear

Name: INTENCLR
Offset: 0x14
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
						INSTREND		CSRISE
Access						R/W		R/W
Reset						0		0

Bit	7	6	5	4	3	2	1	0
					ERROR	TXC	DRE	RXC
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 10 – INSTREND Instruction End Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the corresponding interrupt request.

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

Bit 8 – CSRISE Chip Select Rise Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the corresponding interrupt request.

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

Bit 3 – ERROR Overrun Error Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the corresponding interrupt request.

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

Bit 2 – TXC Transmission Complete Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the corresponding interrupt request.

© 2025 Microchip Technology Inc. and its subsidiaries

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

Bit 1 – DRE Transmit Data Register Empty Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the corresponding interrupt request.

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

Bit 0 – RXC Receive Data Register Full Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the corresponding interrupt request.

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

© 2025 Microchip Technology Inc. and its subsidiaries

35.8.7. Interrupt Enable Set

Name: INTENSET
Offset: 0x18
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
						INSTREND		CSRISE
Access						R/W		R/W
Reset						0		0

Bit	7	6	5	4	3	2	1	0
					ERROR	TXC	DRE	RXC
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 10 – INSTREND Instruction End Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit sets the corresponding interrupt request.

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

Bit 8 – CSRISE Chip Select Rise Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit sets the corresponding interrupt request.

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

Bit 3 – ERROR Overrun Error Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit sets the corresponding interrupt request.

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

Bit 2 – TXC Transmission Complete Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit sets the corresponding interrupt request.

© 2025 Microchip Technology Inc. and its subsidiaries

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

Bit 1 – DRE Transmit Data Register Empty Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit sets the corresponding interrupt request.

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

Bit 0 – RXC Receive Data Register Full Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit sets the corresponding interrupt request.

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

© 2025 Microchip Technology Inc. and its subsidiaries

35.8.8. Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x1C
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						INSTREND		CSRISE
Access						R/W		R/W
Reset						0		0

Bit	7	6	5	4	3	2	1	0
					ERROR	TXC	DRE	RXC
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 10 – INSTREND Instruction End
This bit is set when an Instruction End was detected.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the flag.

Bit 8 – CSRISE Chip Select Rise
The bit is set when a Chip Select Rise was detected.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the flag.

Bit 3 – ERROR Overrun Error
This bit is set when an ERROR has occurred.
An ERROR occurs when RXDATA is loaded at least twice from the serializer.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the flag.

Bit 2 – TXC Transmission Complete

Value	Description
0	As soon as data are written in TXDATA.
1	TXDATA and internal shifter are empty. If a transfer delay was defined, TXC is set after the completion of such delay.

Bit 1 – DRE Transmit Data Register Empty
This bit is '0' when the QSPI is disabled or at Reset.
The bit is set as soon as ENABLE bit is set.

© 2025 Microchip Technology Inc. and its subsidiaries

Value	Description
0	Data were written to TXDATA and not yet transferred to the serializer.
1	The last data written in the TXDATA were transferred to the serializer.

Bit 0 – RXC Receive Data Register Full

Value	Description
0	No data were received since the last read of RXDATA.
1	Data were received and the received data were transferred from the serializer to RXDATA since the last read of RXDATA.

© 2025 Microchip Technology Inc. and its subsidiaries

35.8.9. Status

Name: STATUS
Offset: 0x20
Reset: 0x00000200
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
							CSSTATUS	
Access							R	
Reset							1	

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

Bit 9 – CSSTATUS Chip Select

Value	Description
0	Chip Select is asserted.
1	Chip Select is not asserted.

Bit 1 – ENABLE Enable

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

© 2025 Microchip Technology Inc. and its subsidiaries

35.8.10. Instruction Address

Name: INSTRADDR
Offset: 0x30
Reset: 0x00000000
Property: -

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

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

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

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

Bits 31:0 – ADDR[31:0] Instruction Address
Address to send to the serial Flash memory in the instruction frame.

© 2025 Microchip Technology Inc. and its subsidiaries

35.8.11. Instruction Code

Name: INSTRCTRL
Offset: 0x34
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

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

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

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

Bits 23:16 – OPTCODE[7:0] Option Code
These bits define the option code to send to the serial Flash memory.

Bits 7:0 – INSTR[7:0] Instruction Code
Instruction code to send to the serial Flash memory.

© 2025 Microchip Technology Inc. and its subsidiaries

35.8.12. Instruction Frame

Name: INSTRFRAME
Offset: 0x38
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
				DUMMYLEN[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DDREN	CRMODE	TFRTYPE[1:0]			ADDRLEN	OPTCODELEN[1:0]	
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bit	7	6	5	4	3	2	1	0
	DATAEN	OPTCODEEN	ADDREN	INSTREN		WIDTH[2:0]		
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bits 20:16 – DUMMYLEN[4:0] Dummy Cycles Length
The DUMMYLEN field defines the number of dummy cycles required by the serial Flash memory before data transfer.

Bit 15 – DDREN Double Data Rate Enable
Note: Double data rate operating is only supported in Read.

Value	Description
0	Double Data Rate operating mode is disabled.
1	Double Data Rate operating mode is enabled.

Bit 14 – CRMODE Continuous Read Mode
This bit defines if the Continuous Read mode is enabled or disabled.

Value	Description
0	Continuous Read Mode is disabled.
1	Continuous Read Mode is enabled.

Bits 13:12 – TFRTYPE[1:0] Data Transfer Type
These bits define the data type transfer.

Value	Name	Description
0x0	READ	Read transfer from the serial memory. Scrambling is not performed. Read at random location (fetch) in the serial Flash memory is not possible.
0x1	READMEMORY	Read data transfer from the serial memory. If enabled, scrambling is performed. Read at random location (fetch) in the serial Flash memory is possible.
0x2	WRITE	Write transfer into the serial memory. Scrambling is not performed.
0x3	WRITEMEMORY	Write data transfer into the serial memory. If enabled, scrambling is performed.

© 2025 Microchip Technology Inc. and its subsidiaries

Bit 10 – ADDRLEN Address Length

The ADDRLEN bit determines the length of the address.

Value	Name	Description
0x0	24BITS	24-bit address length
0x1	32BITS	32-bit address length

Bits 9:8 – OPTCODELEN[1:0] Option Code Length

The OPTCODELEN field determines the length of the option code. The value written in OPTCODELEN must be coherent with value written in the field WIDTH. For example, OPTCODELEN = 0 (1-bit option code) is not coherent with WIDTH = 6 (option code sent with QuadSPI protocol, thus, the minimum length of the option code is 4 bits).

Value	Name	Description
0x0	1BIT	1-bit length option code
0x1	2BITS	2-bit length option code
0x2	4BITS	4-bit length option code
0x3	8BITS	8-bit length option code

Bit 7 – DATAEN Data Enable

Value	Description
0	No data are sent/received to/from the serial Flash memory.
1	Data are sent/received to/from the serial Flash memory.

Bit 6 – OPTCODEEN Option Enable

Value	Description
0	The option is not sent to the serial Flash memory.
1	The option is sent to the serial Flash memory.

Bit 5 – ADDREN Address Enable

Value	Description
0	The transfer address is not sent to the serial Flash memory.
1	The transfer address is sent to the serial Flash memory.

Bit 4 – INSTREN Instruction Enable

Value	Description
0	The instruction is not sent to the serial Flash memory.
1	The instruction is sent to the serial Flash memory.

Bits 2:0 – WIDTH[2:0] Instruction Code, Address, Option Code and Data Width

This field defines the width of the instruction code, the address, the option and the data.

Value	Name	Description
0x0	SINGLE_BIT_SPI	Instruction: Single-bit SPI/Address-Option: Single-bit SPI/Data: Single-bit SPI
0x1	DUAL_OUTPUT	Instruction: Single-bit SPI/Address-Option: Single-bit SPI/Data: Dual SPI
0x2	QUAD_OUTPUT	Instruction: Single-bit SPI/Address-Option: Single-bit SPI/Data: Quad SPI
0x3	DUAL_IO	Instruction: Single-bit SPI/Address-Option: Dual SPI/Data: Dual SPI
0x4	QUAD_IO	Instruction: Single-bit SPI/Address-Option: Quad SPI/Data: Quad SPI
0x5	DUAL_CMD	Instruction: Dual SPI/Address-Option: Dual SPI/Data: Dual SPI
0x6	QUAD_CMD	Instruction: Quad SPI/Address-Option: Quad SPI/Data: Quad SPI
0x7	—	Reserved

© 2025 Microchip Technology Inc. and its subsidiaries

35.8.13. Scrambling Mode

Name: SCRAMBCTRL
Offset: 0x40
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
							RANDOMDIS	ENABLE
Access							R/W	R/W
Reset							0	0

Bit 1 – RANDOMDIS Scrambling/Unscrambling Random Value Disable

Value	Description
0	The scrambling/unscrambling algorithm includes the scrambling user key plus a random value that may differ from chip to chip.
1	The scrambling/unscrambling algorithm includes only the scrambling user key.

Bit 0 – ENABLE Scrambling/Unscrambling Enable
This bit defines if the scrambling/unscrambling is enabled or disabled.

Value	Description
0	Scrambling/unscrambling is disabled.
1	Scrambling/unscrambling is enabled.

© 2025 Microchip Technology Inc. and its subsidiaries

35.8.14. Scrambling Key

Name: SCRAMBKEY
Offset: 0x44
Reset: 0x00000000
Property: PAC Write-Protection

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

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

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

Bit	7	6	5	4	3	2	1	0
	KEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – KEY[31:0] Scrambling User Key
This field defines the user key value.

© 2025 Microchip Technology Inc. and its subsidiaries

36. Analog-to-Digital Converter (ADC)

36.1. Overview

The PIC32CX-BZ6 has two 12-bit ADC modules. This ADC module incorporates a multiplexer on the input to facilitate a group of inputs and provides a flexible automated scanning option through the input scan logic.

For the ADC module, the analog inputs are connected to the Sample and Hold (S&H) capacitor. The ADC module performs the conversion of the input analog signal based on the configurations set in the registers. When the conversion is complete, the final result is stored in the result buffer for the specific analog input and is passed to the digital filter and digital comparator if configured to use data from this particular sample.

CVD Analog Front End (AFE)

The CVD AFE module provides an analog front end for CVD touch screen applications. It has a selectable capacitance and the ability to pull the capacitor high or low under software command while an external touch load is connected to the common rail of the ADC.

36.2. Features

The PIC32CX-BZ6 12-bit High Speed Successive Approximation Register (SAR) Analog-to-Digital Converter (ADC) includes the following features:

- Two 12-Bit Resolution
- Two ADC Module, Up to 1 Msps Conversion Rate for Each SAR ADC
- Single-Ended and/or Differential Input
- Capacitive Voltage Divider (CVD) Controller
- Supported in Standby Sleep Mode
- Two Digital Comparators
- Two Digital Filters Supporting Two Modes:
  - Oversampling mode
  - Averaging mode
- Designed for Motor Control, Power Conversion and General Purpose Applications
- Two ADC SAR Cores Sample Up To Two Inputs at a Time and Then Convert the Sampled Inputs in Parallel.
  - First Class Channels: 1
    - Primary Input: AN0
    - Alternate Inputs: ANA0, ANB0
    - Negative Input: ANN0
    - Dedicated trigger per channel
  - Second Class Channels: 7
    - External Inputs: AN1 - AN7
    - Negative Input: ANN1
    - Dedicated trigger per channel
  - Third Class Channels: 15
    - External Inputs: AN8 - AN18
    - Negative Input: ANN1
    - Internal Inputs: Bandgap, Temp Sense, VBAT, VDDCORE

© 2025 Microchip Technology Inc. and its subsidiaries

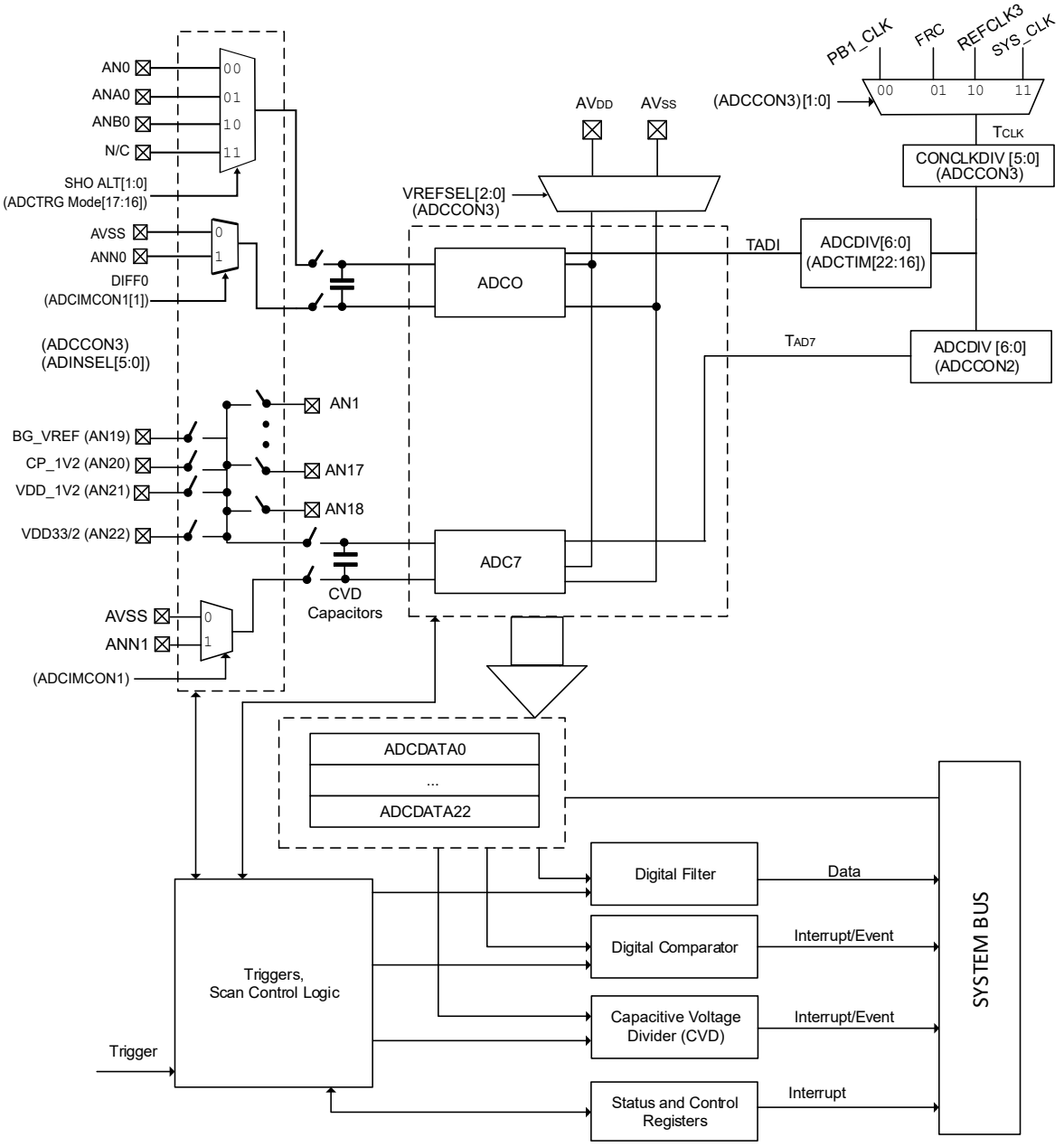
- 1 common scan trigger

– CVD Channels:
  - Rx Lines: CVDR1 - CVDR18
  - Tx Lines: CVDT0 - CVDT17

36.3. Block Diagram

A block diagram of the ADC module is illustrated in the following figure.

Figure 36-1. ADC Block Diagram



36.4. ADC Operation

The High-Speed Successive Approximation Register (SAR) ADC is designed to support power conversion and motor control applications and consists of two individual ADC modules. The

© 2025 Microchip Technology Inc. and its subsidiaries

dedicated ADC module has a single analog input (after alternate selection) connected to its S&H circuit. Since this ADC module samples a dedicated analog input, it is termed a “dedicated” ADC module. A dedicated ADC module is used to measure or capture time-sensitive or transitory analog signals. The shared ADC module has multiple analog input connected to its S&H circuit through a multiplexer. However, this ADC module is capable of up to 2 Msps sample rate.

The analog inputs connected to the ADC modules are Class 1, Class 2 and Class 3 inputs. The number of inputs designated for each class depends on the specific device. For the PIC32CX-BZ6, the following arrangement is provided.

- Class 1 = AN0
- Class 2 = AN1 to AN7
- Class 3 = AN8 to AN18

Note: “BG_VREF” is AN19 , “VDD_1V2” is AN20, “VDD_1V2” is AN21 and “VDD33/2” is AN22. These four channels are internal nodes.

The property of each class of analog input is described in the following table.

Table 36-1. Analog Input Class

ADC Module	Analog Input Class	Trigger	Trigger Action
Dedicated ADC Module	Class 1	Individual trigger source or scan trigger	Ends sampling and starts conversion
Shared ADC module	Class 2	Individual trigger source or scan trigger	Starts sampling sequence or begins scan sequence
Shared ADC module with input scan	Class 3	Scan trigger	Starts scan sequence

Class 1 analog input properties:

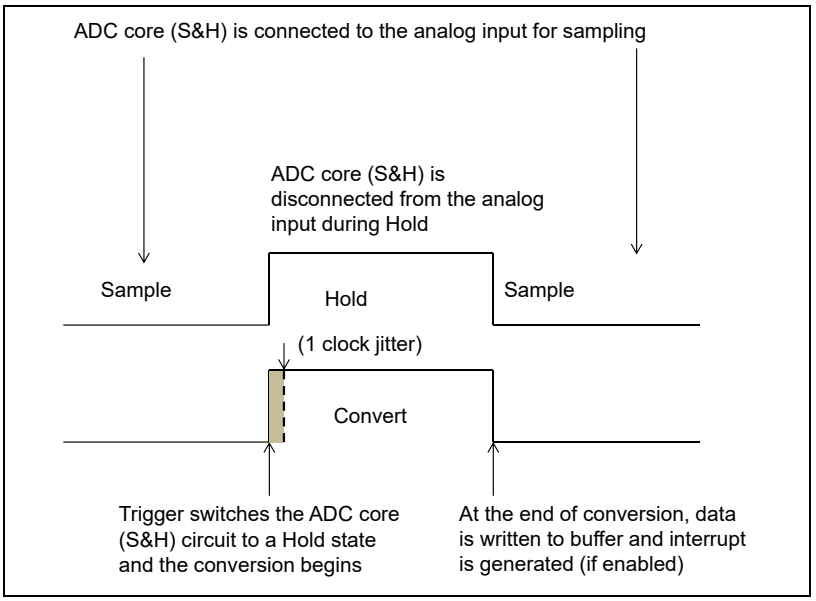
Class 1 inputs are associated with a dedicated ADC module. Dedicated ADC has a single Class 1 input associated with it at a given time. The (alternate) input selection is made through the SH0ALT bits in the ADCTRGMODE register. Regardless of the alternate input selection, the trigger source and the result register remains the same.

Class 1 input has a unique trigger (selected by the ADCTRGx register) and upon arrival of the trigger, ends sampling and starts conversion. Upon completion of conversion,the ADC module reverts back to sampling mode. When a Class 1 input is enabled and is not being converted, it is always sampled.

Class 1 inputs can be part of a scan list, triggered by the common scan trigger source.

© 2025 Microchip Technology Inc. and its subsidiaries

Figure 36-2. Sample and Conversion Sequence for Dedicated ADC Modules

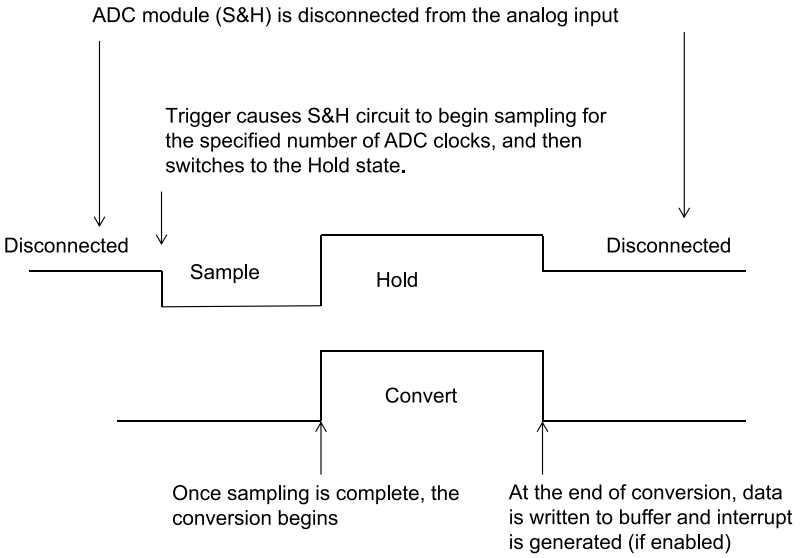


Class 2 and Class 3 analog input properties:

- Class 2 inputs are used on the shared ADC module, either individually triggered or as part of a scan list. When used individually, they are triggered by their unique trigger selected by the ADCTRGx register.
- The analog inputs on the shared ADC have a natural order of priority (for example, AN6 has a higher priority than AN7).
- Class 3 inputs are used exclusively for scanning and share a common trigger source (scan trigger).
- Class 3 analog inputs share both the ADC module and the trigger source; therefore, the only method possible to convert them is to scan them sequentially for each incoming scan trigger event, where scanning occurs in the natural order of priority.
- The arrival of a trigger in the shared ADC module only starts the sampling. When the trigger arrives, the ADC module goes into the Sampling mode for the sampling time decided by the SAMC[9:0] bits (ADCCON2[25:16]). At the end of sampling, the ADC starts conversion. Upon completion of conversion, the ADC module is used to convert the next in line Class 2 or Class 3 inputs according to the natural order of priority. When a shared analog input (Class 2 or Class 3) has completed all conversion and no trigger is pending, the ADC module is disconnected from all analog inputs

© 2025 Microchip Technology Inc. and its subsidiaries

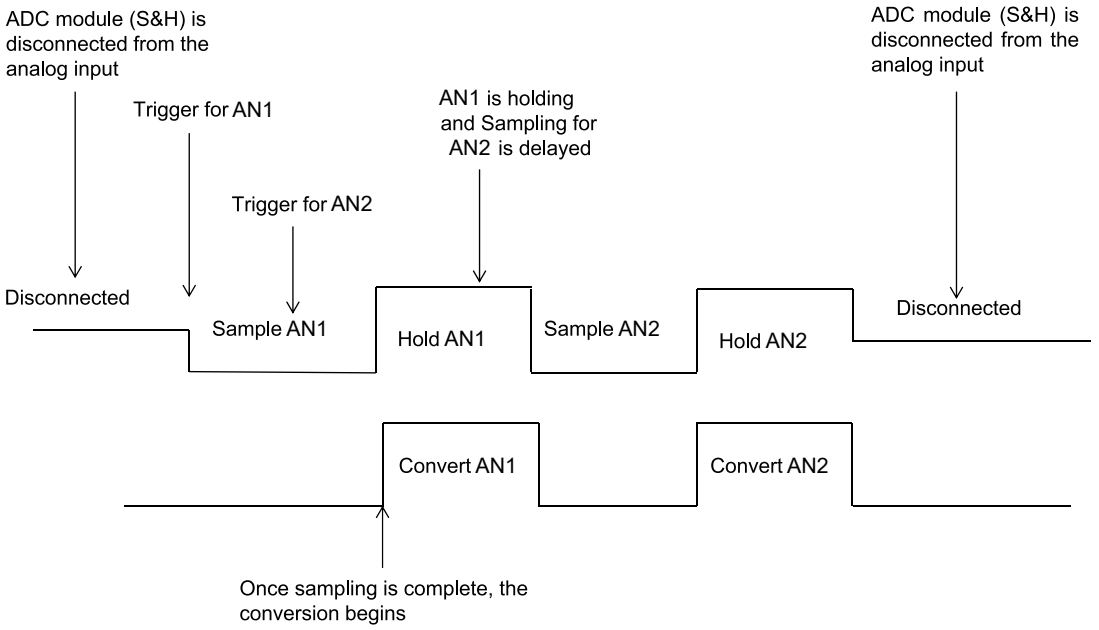
Figure 36-3. Sample and Conversion Sequence for Shared ADC Modules



36.4.1. Class 2 Triggering

When a single Class 2 input is triggered, it is sampled and converted by the shared S&H using the sequence illustrated in Sample and Conversion Sequence for the Shared ADC Modules figure; see *Sample and Conversion Sequence for Shared ADC Modules* figure in the *ADC Operation* from Related Links. When multiple Class 2 inputs are triggered, it is important to understand the consequences of trigger timing. If a conversion is underway and another Class 2 trigger occurs, then the sample-hold-conversion for the new trigger is stalled until the in-process, sample-hold cycle is complete, as shown in the following figure.

Figure 36-4. Multiple Independent Class 2 Trigger Conversion Sequence



© 2025 Microchip Technology Inc. and its subsidiaries

When multiple inputs to the shared S&H are triggered simultaneously, the processing order is determined by their natural priority (the lowest numbered input has the highest priority). As an example, if AN1, AN2 and AN3 are triggered simultaneously, AN1 is sampled and converted first, followed by AN2 and finally, AN3. When using the independent Class 2 triggering on the shared S&H, the SAMC[9:0] bits (ADCCON2[25:16]) determine the sample time for all inputs while the appropriate TRGSRC[4:0] bits in the ADCTRGx Register (see *ADCTRG1* register from Related Links) determine the trigger source for each input.

Related Links

ADC Operation

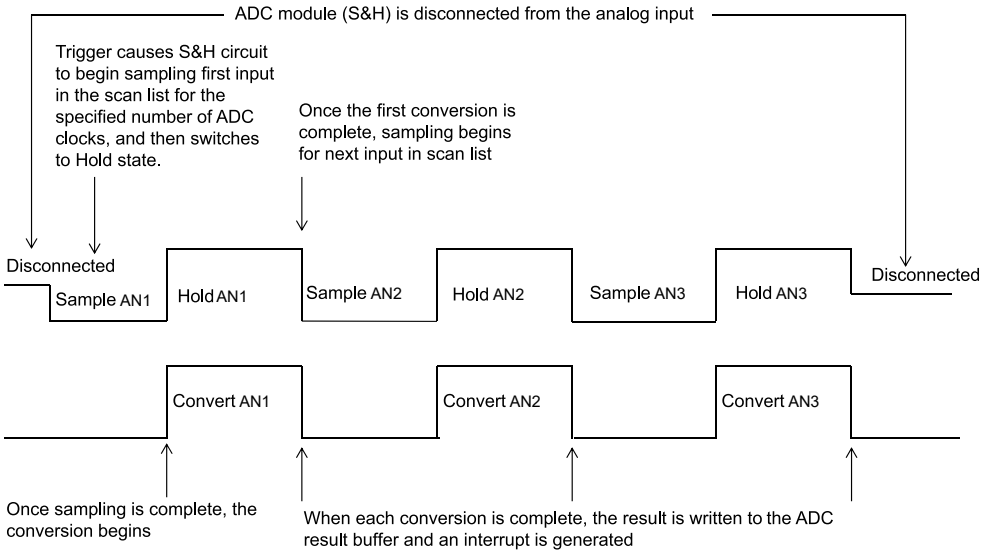
ADCTRG1

ADCTRG1 – ADC Trigger Source 1 Register

36.4.2. Input Scan

Input scanning is a feature that allows an automated scanning sequence of multiple Class 1, Class 2 or Class 3 inputs. All Class 2 and Class 3 inputs are scanned using the single shared S&H. The selection of analog inputs for scanning is done with the CSSx bits of the ADCCSS1 registers. Class 2 inputs are triggered using STRIG selection in the ADCTRGx register, and Class 3 inputs are triggered using the STRGSRC[4:0] of the ADCCON1[20:16] register. When a trigger occurs for Class 2 or Class 3 inputs, the sampling and conversion occur in the natural input order is used; lower number inputs are sampled before higher number inputs.

Figure 36-5. Input Scan Conversion Sequence for Three Class 2 Inputs



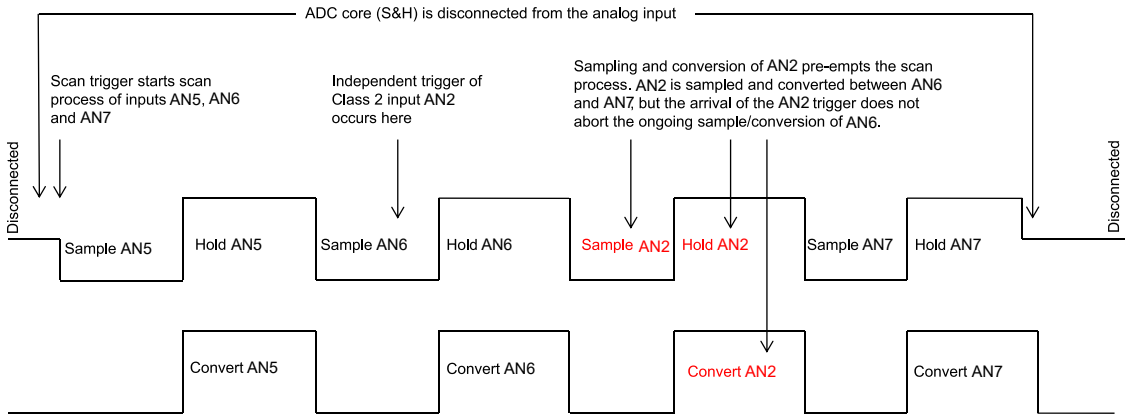
When using the shared analog inputs in scan mode, the SAMC[9:0] bits in the ADC Control Register 2 (ADCCON2[25:16]) determine the sample time for all inputs, while the Scan Trigger Source Selection bits (STRGSRC[4:0]) in the ADC Control Register 1 (ADCCON1[20:16]) determine the trigger source.

To ensure predictable results, a scan must not be retriggered until a sampling of all inputs is complete. Ensure system design to preclude retriggering a scan while a scan is in progress.

Individual Class 2 triggers that occur during a scan preempts the scan sequence if they are a higher priority than the sample currently being processed. In the following figure, a scan of AN5, AN6 and AN7 is underway when an independent trigger of Class 2 input AN2 takes place. The scan is interrupted for the sampling and conversion of AN2.

© 2025 Microchip Technology Inc. and its subsidiaries

Figure 36-6. Scan Conversion Pre-empted by Class 2 Input Trigger



36.5. ADC Module Configuration

Operation of the ADC module is directed through bit settings in the specific registers. The following instructions summarize the actions and the settings. The options and details for each configuration step are provided in the subsequent sections.

To configure the ADC module, perform the following steps:

1. Configure the analog port pins as described in *Configuring the Analog Port Pins*. See *Configuring the Analog Port Pins* from Related Links.
2. Select the analog inputs to the ADC multiplexers as described in *Selecting the ADC Multiplexer Analog Inputs*. See *Selecting the ADC Multiplexer Analog Inputs* from Related Links.
3. Select the format of the ADC result as described in Selecting the Format of the ADC Result. See Selecting the Format of the ADC Result from Related Links.
4. Select the conversion trigger source as described in *Selecting the Conversion Trigger Source*. See *Selecting the Conversion Trigger Source* from Related Links.
5. Select the voltage reference source as described in *Selecting the Voltage Reference Source*. See *Selecting the Voltage Reference Source* from Related Links.
6. Select the scanned inputs as described in *Selecting the Scanned Inputs*. See *Selecting the Scanned Inputs* from Related Links.
7. Select the ADC clock source and prescaler as described in *Selecting the Analog-to-Digital Conversion Clock Source and Prescaler*. See *Selecting the Analog-to-Digital Conversion Clock Source and Prescaler* from Related Links.
8. Specify any additional acquisition time (if required) as described in *Selecting the Analog to- Digital Conversion Clock Source and Prescaler*. See *Selecting the Analog to- Digital Conversion Clock Source and Prescaler* from Related Links.
9. Turn on the ADC module as described in *Turning ON the ADC*. See *Turning ON the ADC* from Related Links.
10. Poll (or wait for the interrupt) for the voltage reference to be ready as described in *Selecting the Voltage Reference Source*. See *Selecting the Voltage Reference Source* from Related Links.
11. Enable the analog and bias circuit for the required ADC modules and, after the ADC module wakes up, enable the digital circuit as described in *Low-Power Mode*. See *Low-Power Mode* from Related Links.
12. Configure the ADC interrupts (if required) as described in *Interrupts*. See *Interrupts* from Related Links.

© 2025 Microchip Technology Inc. and its subsidiaries

36.5.1. Configuring the Analog Port Pins

The ANSELx registers for the I/O ports associated with the analog inputs are used to configure the corresponding pin as an analog or a digital pin. A pin is configured as an analog input when the corresponding ANSELx bit is '1'. When the ANSELx bit is '0', the pin is set to digital control. The ANSELx registers are set when the device comes out of Reset, causing the ADC input pins to be configured as analog inputs by default.

The TRISx registers control the digital function of the port pins. The port pins that are required as analog inputs must have their corresponding bit set in the specific TRISx register, configuring the pin as an input. If the I/O pin associated with an ADC input is configured as an output by clearing the TRISx bit, the port's digital output level (V_{OH} or V_{OL}) is converted. After a device Reset, all of the TRISx bits are set. For more information on port pin configuration, see *I/O Ports and Peripheral Pin Select (PPS)* from Related Links.

Note: When reading a PORT register that shares pins with the ADC, any pin configured as an analog input reads as '0' when the PORT latch is read. Analog levels on any pin that is defined as a digital input but not configured as an analog input, may cause the input buffer to consume the current that exceeds the device specification.

Related Links

I/O Ports and Peripheral Pin Select (PPS)

36.5.2. Selecting the ADC Multiplexer Analog Inputs

The ADC module has two inputs, referred to as the positive and negative inputs. Input selection options vary as described in the following sections.

36.5.2.1. Selection of Positive Inputs

For dedicated ADC modules, one alternate selection is provided for each positive input. This alternate input can be chosen using the SH0ALT[1:0] bits in the ADC Triggering Mode Register (ADCTRGMODE) as follows:

- SH0ALT[1:0] (ADCTRGMODE[17:16])

For the shared ADC module, the positive input is shared among all Class 2 and Class 3 inputs. Input connection of the analog input ANx to the shared ADC is automatic for either the Class 2 input trigger or during a scan of Class 2 and or Class 3 inputs. Selecting inputs for scanning is described in *Selecting the Scanned Inputs* from Related Links.

36.5.2.2. Selection of Negative Inputs

Negative input selection is determined by the setting of the DIFFx bit of the ADCIMCON1 register. The DIFFx bit allows the inputs to be rail-to-rail and either single-ended or differential. The SIGNx and DIFFx bits in the ADCIMCON1 register scale the internal ADC analog inputs and reference voltages and configure the digital result to align with the expected full-scale output range.

For the shared ADC module, the analog inputs have individual settings for the DIFFx bit. Therefore, the user has the ability to select certain inputs as single-ended and others as differential while being connected to the same shared ADC module. While sampling, the signal changes on-the-fly as single-ended or differential according to its corresponding DIFFx bit setting.

Table 36-2. Negative Input Selection

ADCIMCON1		Input Configuration	Input Voltage		Output
DIFFx	SIGNx				
1	1	Differential 2's complement	Minimum input	$V_{IN}P - V_{IN}N = -V_{REF}$	-2048
			Maximum input	$V_{IN}P - V_{IN}N = V_{REF}$	+2047
1	0	Differential unipolar	Minimum input	$V_{IN}P - V_{IN}N = -V_{REF}$	0
			Maximum input	$V_{IN}P - V_{IN}N = V_{REF}$	+4095

© 2025 Microchip Technology Inc. and its subsidiaries

Table 36-2. Negative Input Selection (continued)

ADCIMCON1		Input Configuration	Input Voltage		Output
DIFFx	SIGNx				
0	1	Single-ended 2's complement	Minimum input	$V_{IN}P = V_{REF}$	-2048
			Maximum input	$V_{IN}P - V_{IN}N = V_{REF}$	+2047
0	0	Single-ended unipolar	Minimum input	$V_{IN}P = V_{REF}$	0
			Maximum input	$V_{IN}P - V_{IN}N = V_{REF}$	+4095

Legend:

- $V_{IN}P$ = Positive S&H input
- $V_{IN}N$ = Negative S&H input
- $V_{REF} = V_{REFH} - V_{REFL}$

Note: For proper operation and to prevent device damage, input voltage levels must not exceed the limits listed in the Electrical Specifications.

36.5.3. Selecting the Format of the ADC Result

The data in the ADC Result register can be read in any of the four supported data formats. The user can select from unsigned integer, signed integer, unsigned fractional or signed fractional. Integer data is right-justified and fractional data is left-justified.

- The integer or fractional data format selection is specified globally for all analog inputs using the Fractional Data Output Format bit, FRACT (ADCCON1[23]).
- The signed or unsigned data format selection can be independently specified for each individual analog input using the SIGNx bits in the ADCIMCONx registers.

The following table provides details on how a result is formatted.

Table 36-3. ADC Result Format

FRACT	SIGNx	Description	32-bit Output Data Format			
0	0	Unsigned integer	0000	0000	0000	0000
			0000	dddd	dddd	dddd
0	1	Signed integer	ssss	ssss	ssss	ssss
			ssss	sddd	dddd	dddd
1	0	Fractional	dddd	dddd	dddd	0000
			0000	0000	0000	0000
1	1	Signed fractional	sddd	dddd	dddd	0000
			0000	0000	0000	0000

The following code is an example for ADC Class 2 configuration and fractional format.

```
int main(int argc, char** argv) {
int result[3];

/* Initialize warm up time register */ ADCANCON = 0;
ADCANCONbits.WKUPCLKCNT = 5;  // Wakeup exponent = 32 * TAD7

/* Select ADC input mode */
ADCIMCON1bits.SIGN7 = 0;  // unsigned data format ADCIMCON1bits.DIFF7 = 0;

/* Configure ADCGIRQENx */
ADCGIRQEN1 = 0;           // No interrupts are used
ADCGIRQEN2 = 0;

/* Configure ADCCSSx */
```


© 2025 Microchip Technology Inc. and its subsidiaries

```
ADCCSS1 = 0;              // No scanning is used

/* Configure ADCCMPCONx */
ADCCMPCON1 = 0; // No digital comparators are used. Setting the ADCCMPCONx
ADCCMPCON2 = 0; // register to '0' ensures that the comparator is disabled.

/* Configure ADCFLTRx */
ADCFLTR1 = 0;             // No oversampling filters are used. ADCFLTR2 = 0;

/* Set up the trigger sources */
ADC1TRG2bits.TRGSRC7 = 1; // Set AN7 to trigger from software

/* Turn the ADC on */
ADCCON1bits.ON = 1;

/* Wait for voltage reference to be stable */
while(!ADCCON2bits.BGVRRDY); // Wait until the reference voltage is ready
while(ADCCON2bits.REFFLT);   // Wait if there is a fault with the reference voltage

/* Enable clock to analog circuit */
ADCANCONbits.ANEN7 = 1;       // Enable the clock to analog bias

/* Wait for ADC to be ready */
while(!ADCANCONbits.WKRDY7); // Wait until ADC7 is ready

/* Enable the ADC module */
ADCCON3bits.DIGEN7 = 1;       // Enable ADC7

while (1) {
/* Trigger a conversion */
ADCCON3bits.GSWTRG = 1;

/* Wait the conversions to complete */
while (ADCDSTAT1bits.ARDY7 == 0);
/* fetch the result */
result[0] = ADCDATA7;

/*
* Process results here
*
* Note 1: Loop time determines the sampling time since all inputs are Class 2.
* If the loop time happens is small and the next trigger happens before the
* completion of set sample time, the conversion will happen only after the
* sample time has elapsed.
*
* Note 2: Results are in fractional format
*
*/
}
return (1);
}
```

36.5.4. Selecting the Conversion Trigger Source

Class 1 and Class 2 inputs to the ADC module can be triggered for conversion either individually or as part of a scan sequence. Class 3 inputs can only be triggered as part of a scan sequence. Individual or scan triggers can originate from an event system (EVSYS), from external digital circuits connected to INT0, from external analog circuits connected to an analog comparator or through software by setting a trigger bit in an SFR.

Note: When conversion triggers for multiple Class 2 analog inputs occur simultaneously, they are prioritized according to a natural order priority scheme based on the analog input used. AN1 has the lowest priority, AN2 has the next higher priority and so on.

36.5.5. Selecting the Scanned Inputs

All available analog inputs can be configured for scanning. Class 1 inputs are sampled using their dedicated ADC module. Class 2 and Class 3 inputs are sampled using the shared ADC module. A single conversion trigger source is selected for all of the inputs selected for scanning using the STRGSRC[4:0] bits (ADCCON1[20:16]). On each conversion trigger, the ADC module starts converting (in the natural priority) all inputs specified in the user-specified scan list (ADCCSS1). For Class 2 and

© 2025 Microchip Technology Inc. and its subsidiaries

Class 3 inputs, the trigger initiates a sequential sample/conversion process in the natural priority order.

An analog input belongs to the scan if it is:

- A Class 3 input. For Class 3 inputs, scan is the only mechanism for conversion.
- A Class 1 and Class 2 input that has the scan trigger selected as the trigger source by selecting the STRIG option in the TRGSRCx[4:0] bits located in the ADCTRG1 and ADCTRG2 registers

The trigger options available for scan are identical to those available for independent triggering of Class 1 and Class 2 inputs. Any Class 2 inputs that are part of the scan must have the STRIG option selected as their trigger source in the TRGSRCx[4:0] bits.

Note: The end-of-scan (EOS) is generated only if the last shared input conversion has completed. Until this condition is met, the scan sequence is still in effect. Therefore, the EOS Interrupt can be used for any scan sequence with any combination of input types.

The following code is an example for ADC scanning multiple inputs.

```
int main(int argc, char** argv) {
int result[3];

/* Initialize warm up time register */ ADCANCON = 0;
ADCANCONbits.WKUPCLKCNT = 5; // Wakeup exponent = 32 * TAD7

/* Clock setting */
ADCCON3bits.ADCSEL = 0;       // Select input clock source
ADCCON3bits.CONCLKDIV = 1;    // Control clock frequency is half of input clock
ADCCON3bits.VREFSEL = 0;      // Select AVDD and AVSS as reference source

ADC0TIMEbits.ADCDIV = 1;      // ADC0 clock frequency is half of control clock = TAD7
ADC0TIMEbits.SAMC = 5;        // ADC0 sampling time = 5 * TAD7
ADC0TIMEbits.SELRES = 3;      // ADC0 resolution is 12 bits

/* Select ADC input mode */
ADCIMCON1bits.SIGN0 = 0;      // unsigned data format
ADCIMCON1bits.DIFF0 = 0;      // Single ended mode
ADCIMCON1bits.SIGN7 = 0;      // unsigned data format
ADCIMCON1bits.DIFF7 = 0;      // Single ended mode

/* Configure ADCGIRQENx */
ADCGIRQEN1 = 0;               // No interrupts are used.
ADCGIRQEN2 = 0;

/* Configure ADCCSSx */
ADCCSS1 = 0;                 // Clear all bits
ADCCSS1bits.CSS0 = 1;         // AN0 set for scan
ADCCSS1bits.CSS7 = 1;         // AN7 (Class 2) set for scan

/* Configure ADCCMPCONx */
ADCCMPCON1 = 0;               // No digital comparators are used. Setting the ADCCMPCONx
ADCCMPCON2 = 0;               // register to '0' ensures that the comparator is disabled.

/* Configure ADCFLTRx */
ADCFLTR1 = 0;                // No oversampling filters are used.
ADCFLTR2 = 0;

/* Set up the trigger sources */
ADCTRG1bits.TRGSRC0 = 3;      // Set AN0 (Class 2) to trigger from scan source
ADCTRG2bits.TRGSRC7 = 3;      // Set AN7 (Class 2) to trigger from scan source

/* Turn the ADC on */ ADCCON1bits.ON = 1;

/* Wait for voltage reference to be stable */
while(!ADCCON2bits.BGVRRDY); // Wait until the reference voltage is ready
while(ADCCON2bits.REFFLT);   // Wait if there is a fault with the reference voltage

/* Enable clock to analog circuit */
ADCANCONbits.ANEN7 = 1;       // Enable, ADC7

/* Wait for ADC to be ready */
while(!ADCANCONbits.WKRDY0);     // Wait until ADC0 is ready
while(!ADCANCONbits.WKRDY7);     // Wait until ADC7 is ready
```


© 2025 Microchip Technology Inc. and its subsidiaries

```
/* Enable the ADC module */
ADCCON3bits.DIGEN7 = 1;          // Enable ADC7

while (1) {
/* Trigger a conversion */
ADCCON3bits.GSWTRG = 1;

/* Wait the conversions to complete */
while (ADCDSTAT1bits.ARDY0 == 0);
/* fetch the result */
result[0] = ADCDATA0;

while (ADCDSTAT1bits.ARDY7 == 0);
/* fetch the result */
result[1] = ADCDATA7;

/*
* Process results here
*
*
*/
}
return (1);
}
```

36.5.6. Selecting the Analog-to-Digital Conversion Clock Source and Prescaler

The ADC module can use the internal Fast RC (FRC) oscillator output, system clock (SYS_CLK), reference clock (REFO3) or peripheral bus clock (PB1_CLK) as the conversion clock source (T_Q). See *ADCCON3* register from Related Links.

When the ADCSEL[1:0] bits (ADCCON2[31:30]) = 01, the internal FRC oscillator is used as the ADC clock source. When using the internal FRC oscillator, the ADC module can continue to function in Sleep mode and Idle mode.

Note: It is recommended that applications that require precise timing of ADC acquisitions use SYS_CLK as the clock source for the ADC.

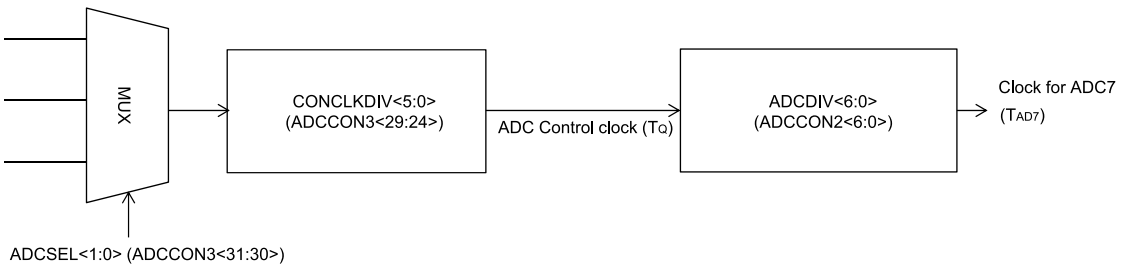
For correct ADC, the conversion clock limits must not be exceeded. Clock frequencies from 1 MHz to 32 MHz are supported by the ADC module.

The maximum rate that analog-to-digital conversions can be completed by the ADC module (effective conversion throughput) is 2 Msps. However, the maximum rate that a single input can be converted is dependent on the sampling time requirements. In addition, the sampling time depends on the output impedance of the analog signal source. For more information on sampling time, see *ADC Sampling Requirements* from Related Links.

The input clock source for the ADC is selected using the ADCSEL[1:0] bits (ADCCON3[31:30]). The input clock is further divided by the control clock divider CONCLKDIV[5:0] bits (ADCCON3[29:24]). The output clock is called the ADC control clock with a time period of T_Q.

The ADC control clock is divided before it is used for the shared ADC by the ADCDIV[6:0] bits (ADCCON2[6:0]). The time period for this clock is denoted as T_{AD7}.

Figure 36-7. Clock Derivation for Shared ADC Modules



© 2025 Microchip Technology Inc. and its subsidiaries

Equation 36-1. Sample Time for the Shared ADC Module

$$t_{SAMC} = ADCCON2 < 25:16 > T_{AD7}$$

$$t_{conversion} = 2 + ADCCON2 < 22:21 > T_{AD7}$$

Equation 36-2. ADC Throughput Rate

$$\text{FTP} = T_{AD7}/(T_{SAMC} + T_{CONV})$$

Where,

- T_{AD7} = The frequency of the individual ADC module

Related Links

ADCCON3

ADCCON3 – ADC Control Register 3

ADC Sampling Requirements

36.5.7. Turning ON the ADC

To turn ON the ADC module, follow these steps:

1. Set the ON bit (ADCCON1[15]) to '1'.
   This places the ADC module in Active mode, making it fully powered and functional.
   When the ON bit is set to '0', the ADC module is disabled.
2. Ensure the analog and digital circuits of the ADC are also turned ON.
   This step is required in addition to setting the ON bit to activate the entire module. See *Low-power Mode* from Related Links.

Note: When the ON bit is set to '0', the ADC module is disabled. In this state, both the digital and analog portions are turned OFF to maximize current savings.
It is not recommended to write to the ADC control bits, such as those that configure the ADC clock, input assignments, scanning, voltage reference selection, sample-and-hold (S&H) circuit operating modes, and interrupt configuration while the ADC module is enabled.

Related Links

Low-Power Mode

36.5.8. ADC Status Bits

The ADC module includes the WKRDY7 and WKDRY0 status bit in the ADCANCON register, which indicates the current state of ADC Analog and bias circuit. The user application must not perform any ADC operations until this bit is set.

36.6. Additional ADC Functions

This section describes some additional features of the ADC module, which includes:

- Digital comparator
- Oversampling filter

36.6.1. Digital Comparator

The ADC module features digital comparators that can be used to monitor selected analog input conversion results and generate interrupts when a conversion result is within the user-specified limits. Conversion triggers are still required to initiate conversions. The comparison occurs automatically once the conversion is complete. This feature is enabled by setting the Digital Comparator Module Enable bit, ENDCMP (ADCCMPCONx[7]).

The user application makes use of an interrupt that is generated when the analog-to-digital conversion result is higher or lower than the specified high and low limit values in the ADCCMPx

© 2025 Microchip Technology Inc. and its subsidiaries

register. The high and low limit values are specified in the DCMPHI[15:0] bits (ADCCMPx[31:16]) and the DCMPLO[15:0] bits (ADCCMPx[15:0]).

The CMPEx bits (x = 0 through 22) in the ADCCMPENx registers are used to specify which analog inputs are monitored by the digital comparator (for 23 analog inputs, ANx, where x = 0 through 22). For more details on mapping of AN19-22 to internal ADC, see *ADC Operations* from Related Links. The ADCCMPCONx register specifies the comparison conditions that generates an interrupt, as follows:

- When IEBTWN is '1', an interrupt is generated when DCMPLO ≤ ADCDATA < DCMPHI
- When IEHIHI is '1', an interrupt is generated when DCMPHI ≤ ADCDATA
- When IEHILO is '1', an interrupt is generated when ADCDATA < DCMPHI
- When IELOHI is '1', an interrupt is generated when DCMPLO ≤ ADCDATA
- When IELOLO is '1', an interrupt is generated when ADCDATA < DCMPLO

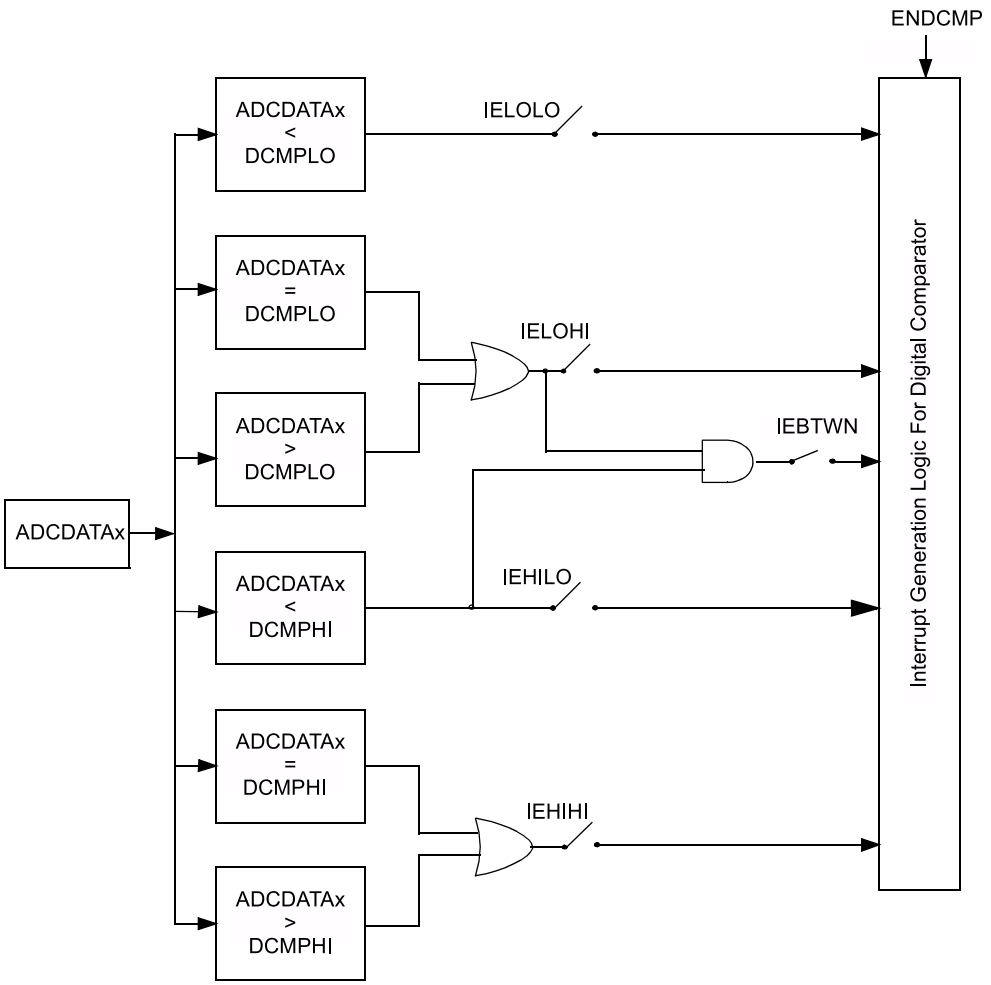
The comparator event generation is illustrated in the following figure. When the ADC module generates a conversion result, the conversion result is provided to the comparator. The comparator uses the DIFFx and SIGNx bits of the ADCIMCONx register (depending on the analog input used) to determine the data format used and to appropriately select whether the comparison must be signed or unsigned. The global ADC setting, which is specified by the FRACT bit (ADCCON1[23]), is also used to set the fractional or integer format. The digital comparator compares the ADC result with the high and low limit values (depending on the selected comparison criteria) in the ADCCMPx register.

Depending on the comparator results, a digital comparator interrupt event may be generated. If a comparator event occurs, the Digital Comparator Interrupt Event Detected status bit, DCMPED (ADCCMPCONx[5]), is set, and the Analog Input Identification (ID) bits, AINID[4:0] (ADCCMPCONx[12:8]), are automatically updated so that the user application knows which analog input generated the interrupt event.

Note: The user software must format the values contained in the ADCCMPx registers to match the converted data format as either signed or unsigned and fractional or integer.

© 2025 Microchip Technology Inc. and its subsidiaries

Figure 36-8. Digital Comparator



The following code is an example for the ADC digital comparator.

```
int main(int argc, char** argv) {
int result = 0, eventFlag = 0;

/* Configure ADCCON1 */
ADCCON1 = 0;                  // No ADCCON1 features are enabled including: Stop-in-Idle,
// turbo, CVD mode, Fractional mode and scan trigger source.
ADCCON1bits.SELRES = 3;        // ADC resolution is 12 bits
ADCCON1bits.STRGSRC = 0;       // No scan trigger.

/* Configure ADCCON2 */
ADCCON2bits.SAMC = 5;          // ADC7 sampling time = (2+SAMC) * TAD7
ADCCON2bits.ADCDIV = 1;        // ADC7 clock freq = TAD7

/* Initialize warm up time register */
ADCANCON = 0;

/* Select ADC input mode */
ADCIMCON1bits.SIGN7 = 0;       // unsigned data format
ADCIMCON1bits.DIFF7 = 0;       // Single ended mode

/* Configure ADCGIRQENx */
ADCGIRQEN1 = 0;                // No interrupts are used
ADCGIRQEN2 = 0;

/* Configure ADCCSSx */
ADCCSS1 = 0;                  // No scanning is used
```


© 2025 Microchip Technology Inc. and its subsidiaries

```
/* Configure ADCCMPCONx */
ADCCMP1 = 0;                  // Clear the register
ADCCMP1bits.DCMPHI = 0xC00;  // High limit is a 3072 result.
ADCCMP1bits.DCMPLO = 0x500;  // Low limit is a 1280 result.
ADCCMPCON1bits.IEBTWN = 1;    // Create an event when the measured result is // >= low limits
and < high limit.
ADCCMPEN1 = 0;                // Clear all enable bits
ADCCMPEN1bits.CMPE8 = 1;      // set the bit corresponding to AN8
ADCCMPCON1bits.ENDCMP = 1;    // enable comparator

/* Configure ADCFLTRx */
ADCFLTR1 = 0;                // No oversampling filters are used.
ADCFLTR2 = 0;
/* Set up the trigger sources */
ADCTRG2bits.TRGSRC7 = 3;         // Set AN7 (Class 2) to trigger from scan source

/* Turn the ADC on */
ADCCON1bits.ON = 1;

/* Wait for voltage reference to be stable */
while(!ADCCON2bits.BGVRRDY);     // Wait until the reference voltage is ready
while(ADCCON2bits.REFFLT);       // Wait if there is a fault with the reference voltage

/* Enable clock to analog circuit */
ADCANCONbits.ANEN7 = 1;          // Enable the clock to analog bias

/* Wait for ADC to be ready */
while(!ADCANCONbits.WKRDY7);     // Wait until ADC7 is ready

/* Enable the ADC module */
ADCCON3bits.DIGEN7 = 1;          // Enable ADC7

while (1) {
/* Trigger a conversion */
ADCCON3bits.GSWTRG = 1;

while (ADCDSTAT1bits.ARDY7 == 0);
/* fetch the result */
result = ADCDATA7;

/* Note: It is not necessary to fetch the result for the digital
* comparator to work. In this example we are triggering from
* software so we are using the ARDY8 to gate our loop. Reading the
* data clears the ARDY bit.
*/
/* See if we have a comparator event*/
if (ADCCMPCON1bits.DCMPED == 1) {
eventFlag = 1;
/*
* Process results here
*/
}
}
return (1);
}
```

Related Links

ADC Operation

36.6.2. Oversampling Digital Filter

The ADC module supports two oversampling digital filters. The oversampling digital filter consists of an accumulator and a decimator (down-sampler), which function together as a low-pass filter. By sampling an analog input at a higher-than-required sample rate, then, processing the data through the oversampling digital filter, the effective resolution of the ADC module can be increased at the expense of decreased conversion throughput.

To obtain x bits of extra resolution, the number of samples required (over and above the Nyquist rate) = $(2^x)^2$:

- 4x oversampling yields one extra bit of resolution (total 13 bits resolution)
- 16x oversampling yields two extra bits of resolution (total 14 bits resolution)

© 2025 Microchip Technology Inc. and its subsidiaries

- 64x oversampling provides three extra bits of resolution (total 15 bits resolution)
- 256x oversampling provides four extra bits of resolution (total 16 bits resolution)

The digital filter also has an averaging mode, where it accumulates the samples and divides it by the number of samples.

Note:

1. Only Class 1 and Class 2 analog inputs can engage the digital filter. Therefore, the FLTINIDx[2:0] bits are 3 bits wide (0 to 7).
2. During the burst conversion process (repeated trigger until all required data for oversampling is obtained), if filtering Class 2 input using the shared ADC module, higher priority ADC inputs may still process conversions; lower priority ADC conversion requests are held waiting until the filter burst sequence is completed.
3. If higher priority requests occur during the digital filter sequence, they delay the completion of the filtering process. This delay may affect the accuracy of the result because the multiple samples cannot be contiguous. The user must arrange the initiation trigger for the oversampling filters to occur while there are no expected interruptions from higher priority ADC conversion requests.

The user application must configure the following bits to perform an oversampling conversion:

- Select the amount of oversampling through the Oversampling Filter Oversampling Ratio (OVRSAM[2:0]) bits in the ADC Filter register (ADCFLTRx[28:26]).
- Set the filter mode to either Oversampling mode or Averaging mode using the DFMODE bit(ADCFLTRx[29]).
- If the filter is set to Averaging mode and the data format is set to fractional (FRACT bit), set or clear the DATA16EN bit (ADCFLTRx[30]) to set the output resolution.
- Set the sample time for subsequent samples:
  - If using Class 2 inputs, select the sample time using the SAMC[9:0] bits (ADCCON2[25:16]).
- Select the specific analog input to be oversampled by configuring the Analog Input ID Selection bits, FLTINIDx[4:0] (ADCFLTRx[20:16]).
- If needed, include the oversampling filter interrupt event in the global ADC interrupt by setting the Accumulator Filter Global Interrupt Enable bit, AFGIEN (ADCFLTRx[25]).
- Enable the oversampling filter by setting the Oversampling Filter Accumulator Enable bit, AFEN (ADCFLTRx[31]).

When the digital filter module is configured, the filter's control logic waits for an external trigger to initiate the process. The trigger signal for the analog input to be oversampled causes the accumulator to be cleared and initiates the first conversion. The trigger also forces the trigger sensitivity into level mode and forces the trigger itself to 1 as long as the filter needs to acquire the user-specified number of samples via the OVRSAM[2:0] bits (ADCFLTRx[28:26]). The time delay between each acquired sample is decided by the set sample time in the SAMC[9:0] bits in the ADCCON2 register for Class 2 and the time for conversion. When the required number set by OVRSAM[2:0] are received and processed, the data stored in the FLT_DATAx[15:0] bit (ADCFLTRx[15:0]) and the AFIFx bit (ADCFLTRx[24]) are set and the interrupt is generated (if enabled).

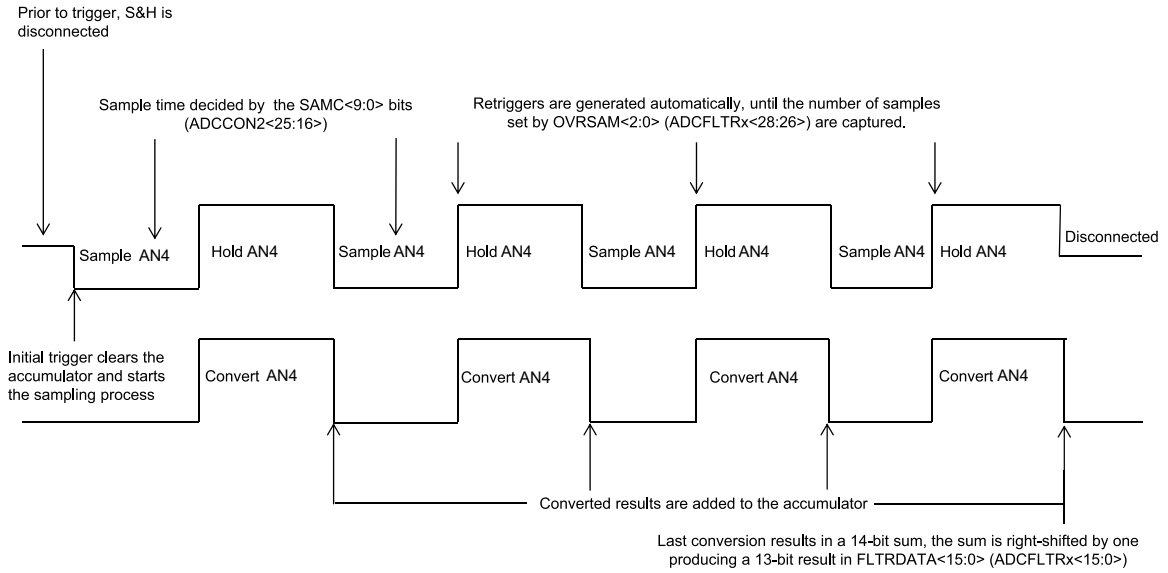
The following figure illustrates 4x oversampling using a Class 2 input. Triggering a Class 2 input initiates sampling for the length of time defined by the SAMC[9:0] bits. Retriggers generated by the oversampling logic use the SAMC[9:0] bits to set the sample time.

Class 2 inputs use the shared S&H; therefore, oversampling blocks lower priority Class 2 and Class 3 triggers. Higher priority Class 2 triggers completely disrupt the oversampling process; therefore, they must be avoided completely. The same priority rule applies to two Class 2 inputs that use two digital filters. In such a case, the higher priority input also uses the shared ADC module in Burst

© 2025 Microchip Technology Inc. and its subsidiaries

mode and prevents the lower priority input from using the shared ADC. Only after all required samples are obtained by the higher priority input can the lower priority input use the shared ADC to acquire samples for its own digital filtering.

Figure 36-9. 4x Oversampling of a Class 2 Input



The following code is an example for ADC digital oversampling filter:

```
int main(int argc, char** argv) {
int result;

/* Configure ADCCON1 */
ADCCON1 = 0;          // No ADCCON1 features are enabled including: Stop-in-Idle, turbo,
// CVD mode, Fractional mode and scan trigger source.

/* Initialize warm up time register */ ADCANCON = 0;
ADCANCONbits.WKUPCLKCNT = 5;  // Wake-up exponent = 32 * TAD7

/* Clock setting */ ADCCON3 = 0;
ADCCON3bits.ADCSEL = 0;       // Select input clock source
ADCCON3bits.CONCLKDIV = 1;    // Control clock frequency is half of input clock
ADCCON3bits.VREFSEL = 0;      // Select AVDD and AVSS as reference source

ADCCON2bits.ADCDIV = 1;      // ADC7 clock frequency is half of control clock = TAD7
ADCCON2bits.SAMC = 5;        // ADC7 sampling time = (2+SAMC) * TAD7
ADCCON1bits.SELRES = 3;      // ADC7 resolution is 12 bits


/* Select ADC input mode */
ADCIMCON1bits.SIGN0 = 0;      // unsigned data format
ADCIMCON1bits.DIFF0 = 0;      // Single ended mode

/* Configure ADCGIRQENx */
ADCGIRQEN1 = 0;               // No interrupts are used
ADCGIRQEN2 = 0;

/* Configure ADCCSSx */
ADCCSS1 = 0;                  // No scanning is used
/* Configure ADCCMPCONx */
ADCCMPCON1 = 0;               // No digital comparators are used. Setting the ADCCMPCONx

/* Configure ADCFLTRx */
ADCFLTR1 = 0;                 // Clear all bits
ADCFLTR1bits.CHNLID = 0;      // Use AN0 as the source
ADCFLTR1bits.OVRSAM = 3;      // 16x oversampling
ADCFLTR1bits.DFMODE = 0;      // Oversampling mode
```


© 2025 Microchip Technology Inc. and its subsidiaries

```
ADCFLTR1bits.AFEN = 1;        // Enable filter 1
ADCFLTR2 = 0;                 // Clear all bits

/* Set up the trigger sources */
ADCTRGSNSbits.LVL0 = 0;        // Edge trigger
ADCTRG1bits.TRGSRC0 = 1;     // Set AN0 to trigger from software.

/* Turn the ADC on */
ADCCON1bits.ON = 1;

/* Wait for voltage reference to be stable */
while(!ADCCON2bits.BGVRRDY); // Wait until the reference voltage is ready
while(ADCCON2bits.REFFLT);   // Wait if there is a fault with the reference voltage

/* Enable clock to analog circuit */
ADCANCONbits.ANEN0 = 1;       // Enable the clock to analog bias and digital control

/* Wait for ADC to be ready */
while(!ADCANCONbits.WKRDY7); // Wait until ADC7 is ready

/* Enable the ADC module */ ADCCON3bits.DIGEN7 = 1;       // Enable ADC7

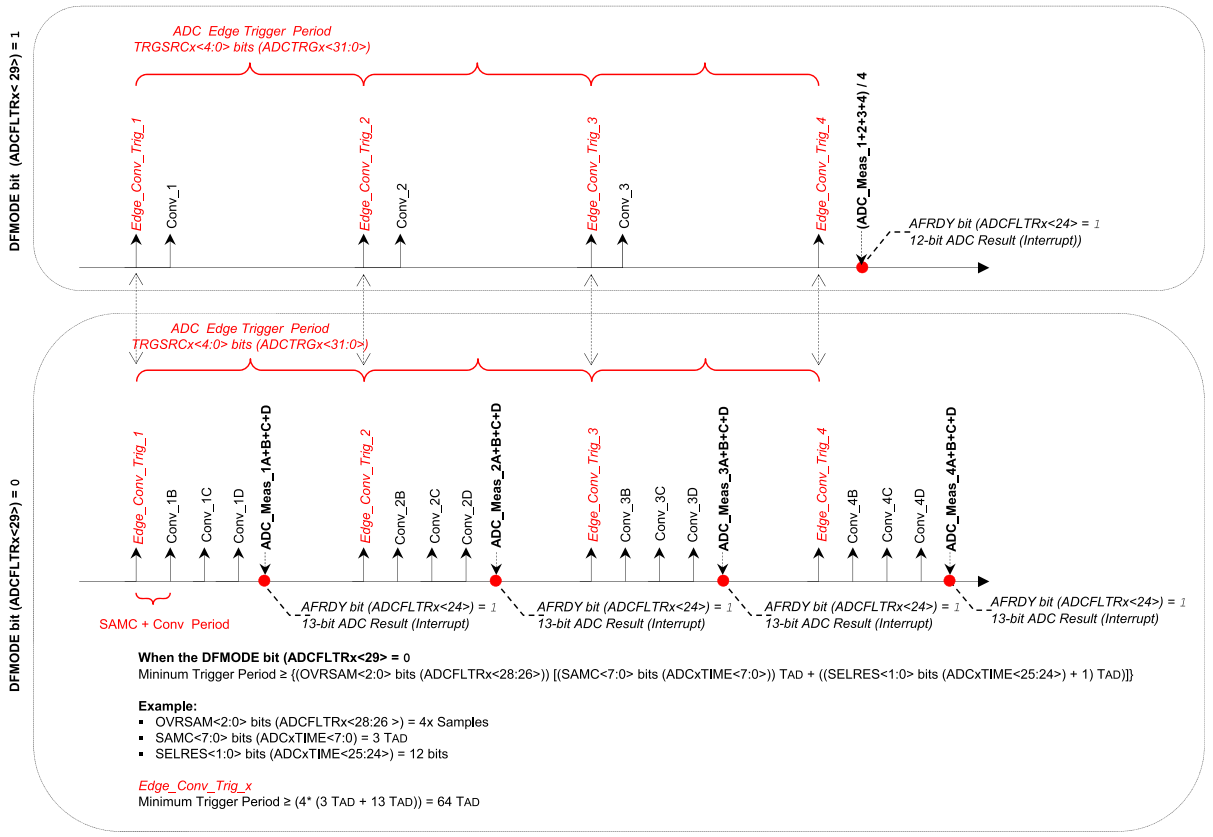
while (1) {
/* Trigger a conversion */ ADCCON3bits.GSWTRG = 1;

/* Wait for the oversampling process to complete */
while (ADCFLTR1bits.AFRDY == 0);
/* fetch the result */
result = ADCFLTR1bits.FLTRDATA;

/*
* Process result Here
*
* Note 1: Loop time determines the sampling time for the first sample.
* remaining samples sample time is determined by set sampling + conversion time.
*
* Note 2: The first 5 samples may have reduced accuracy.
*
*/
}
return (1);
}
```


© 2025 Microchip Technology Inc. and its subsidiaries

Figure 36-10. ADC Filter Comparisons Example



36.7. Interrupts

The ADC module supports interrupts triggered from a variety of sources that can be processed individually or globally. An early interrupt feature is also available to compensate for interrupt servicing latency.

After an enabled interrupt is generated, the CPU jumps to the vector assigned to that interrupt. The CPU begins executing code at the vector address. The user software at this vector address must perform the required operations, such as processing the data results, clearing the interrupt flag, then exiting. See *Nested Vector Interrupt Controller (NVIC)* from Related Links for more information on interrupts and the vector address table details.

Related Links

Nested Vector Interrupt Controller (NVIC)

36.7.1. Interrupt Sources

The ADC is capable of generating interrupts from the events listed in the following table.

Table 36-4. ADC Interrupt Sources

Interrupt Event	Description	Interrupt Enable Bit	Interrupt Status Bit
ANx Data Ready Event (ADC_GIRQ)	Interrupt is generated upon a completion of a conversion from an analog input source (ANx). Each of the ARDYx bits is capable of generating a unique interrupt when set using the ADCBASE register.	AGIENx of ADCGIRQEN1	ARDYx of ADCDSTAT1 register

© 2025 Microchip Technology Inc. and its subsidiaries

Table 36-4. ADC Interrupt Sources (continued)

Interrupt Event	Description	Interrupt Enable Bit	Interrupt Status Bit
Digital Comparator Event (ADC_DIRQ)	When a conversion's comparison criteria are met by a configured and enabled digital comparator. Each of the digital comparators is capable of generating a unique interrupt when its DCMPED bit is set.	DCMPGIEN of ADCCMPCONx register	DCMPED of ADCCMPCONx register
Oversampling Filter Data Ready Event (ADC_AIRQ)	When an oversampling filter completed the accumulation/decimation process and stored the result	AFGIEN of ADCFLTRx register	AFIFx of ADCFLTRx register
Both Band Gap Voltage and ADC Reference Voltage Ready Event (ADC_BGVR_RDY)	Interrupt is generated when both band gap voltage and ADC reference voltage are ready	BGVRIEN of ADCCON2 register	BGVRRDY of ADCCON2 register
Band Gap Fault/Reference Voltage Fault/AV_{DD} Brown-out Fault Event (ADC_FLT)	Interrupt is generated when Band Gap Fault/Reference Voltage Fault/AV_{DD} Brown-out occurs	REFFLTIEN of ADCCON2 register	REFFLT of ADCCON2 register
End of Scan Event (ADC_FCC)	Interrupt is generated when all the selected inputs completed scan	EOSIEN of ADCCON2 register	EOSRDY of ADCCON2 register
ADC Module Wake-up Event	Interrupt is generated when ADC wakes up after being enabled	WKIEN7 and WKIEN0 of ADCANCON register	WKIEN7 and WKIEN0 of ADCANCON register
Update Ready Event	Interrupt is generated when ADC SFRs are ready to be (and can be safely) updated with new values	UPDIEN of ADCCON3 register	UPDRDY of ADCCON3 register

36.7.2. ADC Base Register (ADCBASE) Usage

After conversion of ADC is complete, if the interrupt is vectored to a function that is common to all analog inputs, it takes some significant time to find the ADC input by evaluating the ARDYx bits in the ADCDSTATx. To avoid this time spent, the ADCBASE register is provided, which contains the base address of the user's ADC ISR jump table. When read, the ADCBASE register provides a sum of the contents of the ADCBASE register plus an encoding of the ARDYx bits set in the ADCDSTATx registers. This use of the ADCBASE register supports the creation of an interrupt vector address that can be used to improve the performance of an ISR.

The ARDYx bits are binary priority encoded with ARDY0 being the highest priority and ARDY22 being the lowest priority. The encoded priority result is, then, shifted left the amount specified by the number of bit positions specified by the IRQVS[2:0] bits in the ADCCON1 register, then, added to the contents of the ADCBASE register. If there are no ARDYx bits set, reading the ADCBASE register equals the value written into the ADCBASE register.

The ADCBASE register is typically loaded with the base address of a jump table that contains the address of the appropriate ISR. The kth interrupt request is enabled via the AGIENx bit (0-22) in ADCGIRQEN1.

The following codes are examples for the ADCBASE register usage.

Case 1:

```
ADCBASE = 0x1234; // Set the address
ADCCON1bits.IRQVS = 2; // left shift by 2
ADCGIRQEN1bits.AGIEN0 = 1; // enable interrupt when AN0 completion is done.
```

When the ADC conversion for AN0 is complete, bit `0` of ADCDSTAT1 = ARDY0 is set.

Read value of ADCBASE = 0x1234 + (0 << 2) = 0x1234.

Therefore, the ISR must be placed at address `0x1234` for AN0.

© 2025 Microchip Technology Inc. and its subsidiaries

Case 2:

```
ADCBASE = 0x1234; // Set the address
ADCCON1bits.IRQVS = 2; // left shift by 2
ADCGIRQEN1bits.AGIEN0 = 2; // enable interrupt when AN2 completion is done.
```

When the ADC conversion for AN2 is complete, bit 2 of ADCDSTAT1 = ARDY2 is set.

Read value of ADCBASE = 0x1234 + (2 << 2) = 0x123C.

Therefore, the ISR must be placed at address 0x123C for AN2.

Note: The contents of the ADCBASE register are not altered. Summation is performed when the ADCBASE register is read and the summation result is the returned read value from the ADCBASE SFR.

36.7.3. Interrupt Enabling, Priority and Vectoring

Each of the ADC events previously mentioned generates an interrupt when its associated peripheral Interrupt bits are enabled.. Each of the ADC events previously listed also has an associated interrupt vector. See *Nested Vector Interrupt Controller (NVIC)* from Related Links for more information on the vector location and control/status bits associated with each individual interrupt.

Related Links

Nested Vector Interrupt Controller (NVIC)

36.7.4. Individual and Global Interrupts

The use of the individual interrupts previously listed can significantly optimize the servicing of multiple ADC events by keeping each ISR focused on efficiently handling a specific event. In addition, different ISRs can be easily segregated according to the tasks performed, thereby making user software easier to implement and maintain. There may be cases where it is desirable to have a single ISR service multiple interrupt events. To facilitate this, each ADC event can be logically “ORed” to create a single global ADC interrupt. When an ADC event is enabled for a global interrupt, it vectors to a single interrupt routine. It is the responsibility of this single global ISR to determine the source of the interrupt through polling and process it accordingly.

Use of the Global Interrupt requires configuration of its own unique ISER, IPR0, INTFLAG and configuration of its interrupt vector as described in Interrupt Enabling, Priority and Vectoring. See *Nested Vector Interrupt Controller (NVIC)* from Related Links.

Interrupts for the ADC can be configured as individual or global or utilized as both where some are processed individually and others in the global ISR.

Related Links

Nested Vector Interrupt Controller (NVIC)

36.8. Power-Saving Modes of Operation

The Power-Saving mode, Standby Sleep mode and Idle mode are useful for reducing the conversion noise by minimizing the digital activity of the CPU, buses and other peripherals.

36.8.1. Standby Sleep Mode

When the device enters the Standby Sleep mode, the system clock (SYS_CLK) is halted. If an ADC module selects SYS_CLK as its clock source or selects REFO3 as its clock source (REFO3 is generated from SYS-CLK) and the Standby Sleep mode occurs during a conversion, the conversion is aborted. The converter cannot resume a partially completed conversion on exiting from the Sleep mode. The ADC register contents are not affected by the device entering or leaving the Sleep mode. The ADC module can operate during the Sleep mode if the ADC clock source is derived from a source other than SYS_CLK that is active during the Sleep mode. The FRC clock source is a logical choice for operation during Sleep; however, the REFO3 clock source can also be used, provided it has an input clock that is operational during the Sleep mode.

© 2025 Microchip Technology Inc. and its subsidiaries

ADC operation during the Sleep mode reduces the digital switching noise from the conversion. When the conversion is completed, the ARDYx status bit for that analog input is set and the result is loaded into the corresponding ADC Result register (ADCDATAx).

If any of the ADC interrupts are enabled, the device is woken up from the Sleep mode when the ADC interrupt occurs. The program execution resumes at the ADC ISR if the ADC interrupt is greater than the current CPU priority. Otherwise, execution continues from the instruction after the WFI instruction that placed the device in the Sleep mode.

To minimize the effects of digital noise on the ADC module operation, the user must select a conversion trigger source that ensures that the ADC take places in the Sleep mode. For example, the external interrupt pin (INT0) conversion trigger option (TRGSRC[4:0] = 00100) can be used for performing sampling and conversion while the device is in the Sleep mode.

Note: For the ADC module to operate in the Sleep mode, the ADC clock source must be set to Internal FRC (ADCSEL[1:0] bits (ADCCON3[31:30]) = 01). Alternately, the REFO3 source can be used; however, the clock source used for REFO3 must operate during Sleep mode. Any changes to the ADC clock configuration require that the ADC be disabled.

36.8.2. Operation During Idle Mode

For the ADC, the stop in Idle Mode bit, SIDL (ADCCON1[13]), specifies whether the ADC module stops on Idle or continues on Idle. If SIDL = 0, the ADC module continues normal operation when the device enters Idle mode. If any of the ADC interrupts are enabled, the device wakes up from Idle mode when the ADC interrupt occurs. The program execution resumes at the ADC ISR if the ADC interrupt is greater than the current CPU priority. Otherwise, execution continues from the instruction after the WAIT instruction that placed the device in Idle mode.

If SIDL = 1, the ADC module stops in Idle mode. If the device enters Idle mode during a conversion, the conversion is aborted. The converter cannot resume a partially completed conversion on exiting from Idle mode.

36.8.3. Low-Power Mode

The ADC module can be placed in a low-power state by disabling the digital circuit for individual ADC modules that are not running. This is possible by clearing the DIGEN7 bit in the ADCCON3 register. (See *ADCCON3* register from Related Links.)

An even lower power state is possible by disabling the analog and bias circuit for ADC module that is not running. This is possible by clearing the ANEN7 bit in the ADCANCON register. (See *ADCANCON* register from Related Links.) Disabling the digital circuit to achieve Low-Power mode provides a significantly faster module restart compared to disabling and re-enabling the analog and bias circuit of the ADC module. This is because disabling and re-enabling the analog and bias circuit using the ANEN7 bit requires a wake-up time (typical minimum wake-up time of 20 µs) for the ADC module before it can be used. See *Electrical Characteristics* from Related Links for more information on the stabilization time.

When the analog and bias circuit for an ADC module is enabled, the wake-up must be polled (or through an interrupt) using the wake-up ready bits, WKRDY7, which must be equal to '1'.

Related Links

ADCCON3

ADCCON3 – ADC Control Register 3

ADCANCON

ADCANCON – ADC Analog Warm-up Control Register

Electrical Characteristics

© 2025 Microchip Technology Inc. and its subsidiaries

36.9. Effects of Reset

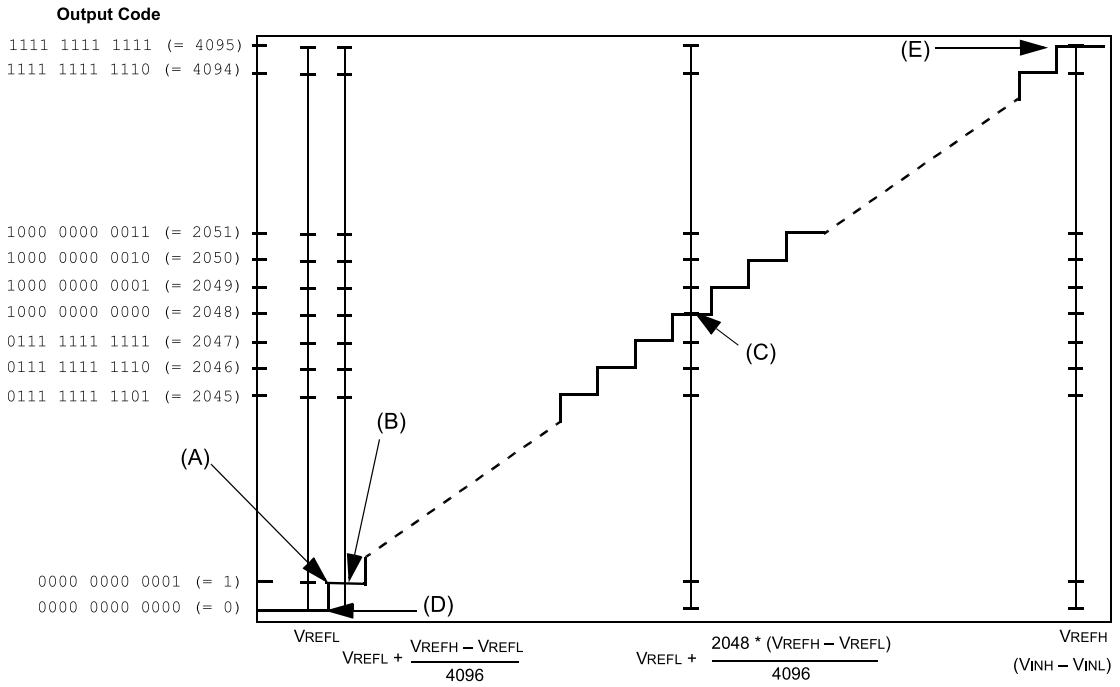
Following any Reset event, all the ADC control and status registers are reset to their default values with control bits in a non-active state. This disables the ADC module and sets the analog input pins to Analog Input mode. Any conversion that was in progress terminates, and the result cannot be written to the result buffer. The values in the ADCDATAx registers are initialized to '0x00000000' during a device Reset. The bias circuits are also turned OFF, so the ADC resuming operations wait for the bias circuits to stabilize by polling (or requesting to be interrupted by) the BGVRRDY bit (ADCCON2 register).

36.10. Transfer Function

A typical transfer function of the 12-bit ADC is illustrated in the following figure. The difference of the input voltages (V_{INH} - V_{INL}) is compared with the reference (V_{REFH} - V_{REFL}).

- The first code transition (A) occurs when the input voltage is (V_{REFH} - V_{REFL}/8192) or 0.5 LSb.
- The 0000 0000 0001 code is centered at (V_{REFH} - V_{REFL}/4096) or 1.0 LSb (B).
- The 1000 0000 0000 code is centered at (2048 * (V_{REFH} - V_{REFL})/4096) (C).
- An input voltage less than (1 * (V_{REFH} - V_{REFL})/8192) converts as 0000 0000 0000 (D).
- An input greater than (8192 * (V_{REFH} - V_{REFL})/8192) converts as 1111 1111 1111 (E).

Figure 36-11. Analog-to-Digital Transfer Function



36.11. ADC Sampling Requirements

The analog input model of the 12-bit ADC is illustrated in the following figure. The total acquisition time for the analog-to-digital conversion is a function of the internal circuit settling time and the holding capacitor charge time.

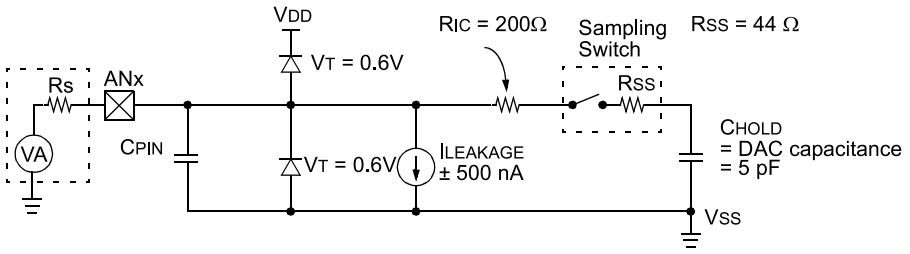
For the ADC module to meet its specified accuracy, the charge holding capacitor (C_{HOLD}) must be allowed to fully charge to the voltage level on the analog input pin. The analog output source impedance (R_S), the interconnect impedance (R_{IC}) and the internal sampling switch (R_{SS}) impedance combine to directly affect the time required to charge the C_{HOLD}. The combined impedance of the analog sources must, therefore, be small enough to fully charge (to within one-fourth LSB of

© 2025 Microchip Technology Inc. and its subsidiaries

the desired voltage) the holding capacitor within the selected sample time. The internal holding capacitor is in the discharged state prior to each sample operation.

At least 1 T_{AD7} time period must be allowed between conversions for the acquisition time. See *Electrical Characteristics* from Related Links.

Figure 36-12. 12-bit ADC Analog Input Model



Note: The C_{PIN} value depends on the device package and is not tested. The effect of the C_{PIN} is negligible if Rs 5 k.

Legend:

- C_{PIN} = Input capacitance
- R_{SS} = Sampling switch resistance
- R_S = Source resistance
- $I_{LEAKAGE}$ = Leakage current at the pin due to various junctions
- V_T = Threshold voltage
- R_{IC} = Interconnect resistance
- C_{HOLD} = Sample/hold capacitance

Related Links

Electrical Characteristics

36.11.1. Connection Considerations

Because the analog inputs employ Electrostatic Discharge (ESD) protection, they have diodes to V_{DD} and V_{SS}; therefore, the analog input must be between V_{DD} and V_{SS}. If the input voltage exceeds this range by greater than 0.3V (either direction), one of the diodes becomes forward biased, and it may damage the device if the input current specification is exceeded.

An external RC filter is sometimes added for antialiasing of the input signal. The R (resistive) component must be selected to ensure that the acquisition time is met. Any external components connected (through high-impedance) to an analog input pin (capacitor, Zener diode and so on) must have very little leakage current at the pin.

© 2025 Microchip Technology Inc. and its subsidiaries

36.12. Register Summary

The register offsets shown below are with respect to Base Address = `0x4400_1500`.

The PIC32CX-BZ6 12-bit High Speed SAR ADC module has the following Special Function Registers (SFRs):

Note: All registers in this table have corresponding CLR, SET and INV registers at its virtual address, plus an offset of 0x4, 0x8 and 0xC, respectively. See *CLR, SET and INV Registers* from Related Links.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	ADCCON1	7:0			IRQVS[2:0]		STRGLVL		DMABL[2:0]	
		15:8	ON	FRZ	SIDL		CVD_EN	FSYDMA	FSYUPB	SRTGDS
		23:16	FRACT	SELRES[1:0]				STRGSRC[4:0]		
		31:24								
0x04 ... 0x0F	Reserved									
0x10	ADCCON2	7:0		ADCDIV[6:0]						
		15:8	BGVRIEN	REFFLTIEN	EOSIEN		ENXCNVRT	ADCEIS[2:0]		
		23:16	SAMC[7:0]							
		31:24	BGVRRDY	REFFLT	EOSRDY	CVD_CPL[2:0]			SAMC[9:8]	
0x14 ... 0x1F	Reserved									
0x20	ADCCON3	7:0	GLSWTRG	GSWTRG	ADINSEL[5:0]					
		15:8	VREFSEL[2:0]			TRGSUSP	UPDIEN	UPDRDY	SAMP	RQCNVRT
		23:16	DIGEN7							DIGEN0
		31:24	ADCSEL[1:0]		CONCLKDIV[5:0]					
0x24 ... 0x2F	Reserved									
0x30	ADCTRGMODE	7:0								SSAMPEN0
		15:8								STRGEN0
		23:16							SH0ALT[1:0]	
		31:24								
0x34 ... 0x3F	Reserved									
0x40	ADCIMCON1	7:0	DIFF3	SIGN3	DIFF2	SIGN2	DIFF1	SIGN1	DIFF0	SIGN0
		15:8	DIFF7	SIGN7	DIFF6	SIGN6	DIFF5	SIGN5	DIFF4	SIGN4
		23:16	DIFF11	SIGN11	DIFF10	SIGN10	DIFF9	SIGN9	DIFF8	SIGN8
		31:24	DIFF15	SIGN15	DIFF14	SIGN14	DIFF13	SIGN13	DIFF12	SIGN12
0x44 ... 0x4F	Reserved									
0x50	ADCIMCON2	7:0	DIFF19	SIGN19	DIFF18	SIGN18	DIFF17	SIGN17	DIFF16	SIGN16
		15:8			DIFF22	SIGN22	DIFF21	SIGN21	DIFF20	SIGN20
		23:16								
		31:24								
0x54 ... 0x7F	Reserved									
0x80	ADCGIRQEN1	7:0	AGIEN7	AGIEN6	AGIEN5	AGIEN4	AGIEN3	AGIEN2	AGIEN1	AGIEN0
		15:8	AGIEN15	AGIEN14	AGIEN13	AGIEN12	AGIEN11	AGIEN10	AGIEN9	AGIEN8
		23:16		AGIEN22	AGIEN21	AGIEN20	AGIEN19	AGIEN18	AGIEN17	AGIEN16
		31:24								
0x84 ... 0x9F	Reserved									

© 2025 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xA0	ADCCSS1	7:0	CSS7	CSS6	CSS5	CSS4	CSS3	CSS2	CSS1	CSS0
		15:8	CSS15	CSS14	CSS13	CSS12	CSS11	CSS10	CSS9	CSS8
		23:16		CSS22	CSS21	CSS20	CSS19	CSS18	CSS17	CSS16
		31:24								
0xA4 ... 0xBF	Reserved									
0xC0	ADCDSTAT1	7:0	ARDY7	ARDY6	ARDY5	ARDY4	ARDY3	ARDY2	ARDY1	ARDY0
		15:8	ARDY15	ARDY14	ARDY13	ARDY12	ARDY11	ARDY10	ARDY9	ARDY8
		23:16		ARDY22	ARDY21	ARDY20	ARDY19	ARDY18	ARDY17	ARDY16
		31:24								
0xC4 ... 0xDF	Reserved									
0xE0	ADCCMPEN1	7:0	CMPE7	CMPE6	CMPE5	CMPE4	CMPE3	CMPE2	CMPE1	CMPE0
		15:8	CMPE15	CMPE14	CMPE13	CMPE12	CMPE11	CMPE10	CMPE9	CMPE8
		23:16		CMPE22	CMPE21	CMPE20	CMPE19	CMPE18	CMPE17	CMPE16
		31:24								
0xE4 ... 0xEF	Reserved									
0xF0	ADCCMP1	7:0	ADCMPLO[7:0]							
		15:8	ADCMPLO[15:8]							
		23:16	ADCMPHI[7:0]							
		31:24	ADCMPHI[15:8]							
0xF4 ... 0xFF	Reserved									
0x0100	ADCCMPEN2	7:0	CMPEx7	CMPEx6	CMPEx5	CMPEx4	CMPEx3	CMPEx2	CMPEx1	CMPEx0
		15:8	CMPEx15	CMPEx14	CMPEx13	CMPEx12	CMPEx11	CMPEx10	CMPEx9	CMPEx8
		23:16		CMPEx22	CMPEx21	CMPEx20	CMPEx19	CMPEx18	CMPEx17	CMPEx16
		31:24								
0x0104 ... 0x010F	Reserved									
0x0110	ADCCMP2	7:0	ADCMPLO[7:0]							
		15:8	ADCMPLO[15:8]							
		23:16	ADCMPHI[7:0]							
		31:24	ADCMPHI[15:8]							
0x0114 ... 0x019F	Reserved									
0x01A0	ADCFLTR1	7:0	FLT_DATA0[7:0]							
		15:8	FLT_DATA0[15:8]							
		23:16				FLTINID0[4:0]				
		31:24	AFEN0	DATA16EN0	ADFMODE0	OVRSAM0[2:0]			AFIEN0	AFIF0
0x01A4 ... 0x01AF	Reserved									
0x01B0	ADCFLTR2	7:0	FLT_DATA1[7:0]							
		15:8	FLT_DATA1[15:8]							
		23:16				FLTINID1[4:0]				
		31:24	AFEN1	DATA16EN1	ADFMODE1	OVRSAM1[2:0]			AFIEN1	AFIF1
0x01B4 ... 0x01FF	Reserved									
0x0200	ADCTRG1	7:0				TRGSRC0[4:0]				
		15:8				TRGSRC1[4:0]				
		23:16				TRGSRC2[4:0]				
		31:24				TRGSRC3[4:0]				

© 2025 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0204 ... 0x020F	Reserved									
0x0210	ADCTRG2	7:0				TRGSRC4[4:0]				
		15:8				TRGSRC5[4:0]				
		23:16				TRGSRC6[4:0]				
		31:24				TRGSRC7[4:0]				
0x0214 ... 0x027F	Reserved									
0x0280	ADCCMPCON1	7:0	ENDCMP	DCMPGIEN	DCMPED	IEBTWN	IEHIHI	IEHILO	IELOHI	IELOLO
		15:8			CMPINID0[5:0]					
		23:16	CVD_DATA[7:0]							
		31:24	CVD_DATA[15:8]							
0x0284 ... 0x028F	Reserved									
0x0290	ADCCMPCON2	7:0	ENDCMP	DCMPGIEN	DCMPED	IEBTWN	IEHIHI	IEHILO	IELOHI	IELOLO
		15:8				AINID[4:0]				
		23:16								
		31:24								
0x0294 ... 0x02FF	Reserved									
0x0300	ADCBASE	7:0	ADCBASE[7:0]							
		15:8	ADCBASE[15:8]							
		23:16								
		31:24								
0x0304 ... 0x030F	Reserved									
0x0310	ADCDMASTAT	7:0								RAF0
		15:8	DMA_CNT_EN							RAF_IEN0
		23:16	WR_OVF_ERR							RBF0
		31:24	DMAGEN			RBF_IEN0				
0x0314 ... 0x031F	Reserved									
0x0320	ADCCNTB	7:0	ADCCNTB[7:0]							
		15:8	ADCCNTB[15:8]							
		23:16	ADCCNTB[23:16]							
		31:24	ADCCNTB[31:24]							
0x0324 ... 0x032F	Reserved									
0x0330	ADCDMAB	7:0	ADDMAB[7:0]							
		15:8	ADDMAB[15:8]							
		23:16	ADDMAB[23:16]							
		31:24	ADDMAB[31:24]							
0x0334 ... 0x033F	Reserved									
0x0340	ADCTRGSNS	7:0	LVL7	LVL6	LVL5	LVL4	LVL3	LVL2	LVL1	LVL0
		15:8								
		23:16								
		31:24								
0x0344 ... 0x034F	Reserved									

© 2025 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0350	ADC0TIME	7:0	SAMCx[7:0]							
		15:8							SAMCx[9:8]	
		23:16	BCHENx	ADCDIVx[6:0]						
		31:24				EISx[2:0]			SELRESx[1:0]	
0x0354 ... 0x03FF	Reserved									
0x0400	ADCANCON	7:0	ANEN7							ANEN0
		15:8	WKRDY7							WKRDY0
		23:16	WKIEN7							WKIEN0
		31:24					WKUPCLKCNT[3:0]			
0x0404 ... 0x066F	Reserved									
0x0670	ADC7CFG	7:0	ADCCFG[7:0]							
		15:8	ADCCFG[15:8]							
		23:16	ADCCFG[23:16]							
		31:24	ADCCFG[31:24]							
0x0674 ... 0x06FF	Reserved									
0x0700	ADCSYSCFG0	7:0	AN[7:0]							
		15:8	AN[15:8]							
		23:16		AN[22:16]						
		31:24								
0x0704 ... 0x09FF	Reserved									
0x0A00	ADCDATA0	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0A04 ... 0x0A0F	Reserved									
0x0A10	ADCDATA1	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0A14 ... 0x0A1F	Reserved									
0x0A20	ADCDATA2	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0A24 ... 0x0A2F	Reserved									
0x0A30	ADCDATA3	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0A34 ... 0x0A3F	Reserved									
0x0A40	ADCDATA4	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							

© 2025 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0A44 ... 0x0A4F	Reserved									
0x0A50	ADCDATA5	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0A54 ... 0x0A5F	Reserved									
0x0A60	ADCDATA6	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0A64 ... 0x0A6F	Reserved									
0x0A70	ADCDATA7	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0A74 ... 0x0A7F	Reserved									
0x0A80	ADCDATA8	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0A84 ... 0x0A8F	Reserved									
0x0A90	ADCDATA9	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0A94 ... 0x0A9F	Reserved									
0x0AA0	ADCDATA10	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0AA4 ... 0x0AAF	Reserved									
0x0AB0	ADCDATA11	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0AB4 ... 0x0ABF	Reserved									
0x0AC0	ADCDATA12	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0AC4 ... 0x0ACF	Reserved									

© 2025 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0AD0	ADCDATA13	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0AD4 ... 0x0ADF	Reserved									
0x0AE0	ADCDATA14	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0AE4 ... 0x0AEF	Reserved									
0x0AF0	ADCDATA15	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0AF4 ... 0x0AFF	Reserved									
0x0B00	ADCDATA16	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0B04 ... 0x0B0F	Reserved									
0x0B10	ADCDATA17	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0B14 ... 0x0B1F	Reserved									
0x0B20	ADCDATA18	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0B24 ... 0x0B2F	Reserved									
0x0B30	ADCDATA19	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0B34 ... 0x0B3F	Reserved									
0x0B40	ADCDATA20	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							
0x0B44 ... 0x0B4F	Reserved									
0x0B50	ADCDATA21	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							

© 2025 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0B54 ... 0x0B5F	Reserved									
0x0B60	ADCDATA22	7:0	DATA[7:0]							
		15:8	DATA[15:8]							
		23:16	DATA[23:16]							
		31:24	DATA[31:24]							

Related Links
CLR, SET and INV Registers

36.13. Register Description

The following is the list of conventions available in the register description:

- R = Readable bit
- W = Writable bit
- U = Unimplemented bit, read as '0'
- -n = Value at POR
- 1 = Bit is set
- 0 = Bit is cleared
- x = Bit is unknown
- HS = Hardware Set
- HC = Hardware Cleared

© 2025 Microchip Technology Inc. and its subsidiaries

36.13.1. ADCCON1 – ADC Control Register 1

Name: ADCCON1
Offset: 0x00
Reset: 0x00601000
Property: -

This register controls the basic operation of the ADC module, including behavior in the Sleep and Idle modes and data formatting. This register also specifies the vector shift amounts for the Interrupt Controller.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	FRACT	SELRES[1:0]		STRGSRC[4:0]				
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ON	FRZ	SIDL		CVD_EN	FSYDMA	FSYUPB	SRTGDS
Access	R/W	R/W	R/W		R/W	R/W	R/W	R/W
Reset	0	0	0		0	0	0	0

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

Bit 23 – FRACT Fractional Data Output Format bit

Value	Description
0	Integer
1	Fractional

Bits 22:21 – SELRES[1:0] Shared ADC Resolution bits

Note: Changing the resolution of the ADC does not shift the result in the corresponding ADCDATAx register. The result occupies 12 bits with the corresponding lower unused bits set to '0'. For example, a resolution of 6 bits results in ADCDATAx[5:0] being set to '0' and ADCDATAx[11:6] holding the result.

Value	Description
11	12 bits (default)
10	10 bits
01	8 bits
00	6 bits

Bits 20:16 – STRGSRC[4:0] ScanTrigger Source Select bits

Value	Description
10001 - 11111	Reserved
10000	EVSYS_53
01111	EVSYS_52
01110	EVSYS_51

© 2025 Microchip Technology Inc. and its subsidiaries

Value	Description
01101	EVSYS_50
01100	EVSYS_49
01011	EVSYS_48
01010	EVSYS_47
01001	EVSYS_46
01000	EVSYS_45
00111	EVSYS_44
00110	EVSYS_43
00101	EVSYS_42
00100	INT0 External interrupt
00011	Reserved
00010	Global level software trigger (GLSWTRG)
00001	Global software edge trigger (GSWTRG)
00000	No Trigger

Bit 15 – ON ADC Module Enable bit

Note: The ON bit must be set only after the ADC module is configured.

Value	Description
0	ADC module is disabled
1	ADC module is enabled

Bit 14 – FRZ Freeze in Debug Mode

Value	Description
0	Do not freeze in the Debug mode
1	Freeze in the Debug mode

Bit 13 – SIDL Stop in Idle Mode bit

Value	Description
0	Continue module operation in the Idle mode
1	Discontinue module operation when device enters the Idle mode

Bit 11 – CVD_EN CVD Enable (Capacitive Voltage Division Enable) is the bit that enables and starts the CVD operation.

Notes:

1. The software must ensure that prior to enable CVD_EN, the shared ADC core is enabled and ready for conversions; in other words, ADCANCON.ANENx = 1'b1 and ADCANCON.WKRDYx = 1'b1 and ADCCON3.CHN_EN_SHR = 1'b1.
2. The software must disable all external triggers for all second class channels by setting the corresponding ADCTRGx.TRGSRCx[4:0] = 5'h00 and, also, the third class scan trigger by setting ADCCONx.STRGSRC[4:0] = 5'h00.
3. The software must enable the ADCCSS1.CSSx channel scan select bits of all the channels to be included in the CVD scan.
4. The software must set up the Digital Comparator 1 with the necessary comparison values in ADCCMP1 and the required setup in ADCCMPCON1 (enabling the comparator itself as well as its interrupt if desired). The register ADCCMPEN1 is irrelevant for the CVD operation. The Digital Comparator 1 updates its status field ADCCMPCON1.AINID[5:0] with the channel ID just finished for CVD only upon issuing an ADCCMPCON1.DCMPED interrupt signifying a detected touch event. When the CVD accomplishes its purpose (usually after an interrupt request from the Digital Comparator 1 signaling a touch event), the software must clear first the ADCCON3.DIGEN7 bit before clearing CVD_EN. After that, the software may set again

© 2025 Microchip Technology Inc. and its subsidiaries

ADCCON3.DIGEN7 and start normal A/D conversions on the shared ADC core for all second and third class channels.

Bit 10 – FSYDMA Fast Synchronous DMA System Clock

When set, it means "Fast Synchronous DMA System Clock" to the ADC control clock.

Value	Description
0	When this bit is cleared, it forces the ADC Controller to engage the aforementioned synchronizers, except the situation, where ADCON3.ADCSEL == 2'b11, in which case the aforementioned synchronizers are always bypassed. When ADCON3.ADCSEL == 2'b11, the ADC control clock source is the DMA system clock itself, therefore the ADC control clock is a divided down synchronous version of the DMA system clock, situation which satisfies all the conditions to force FSYDMA high and bypass the ADC control clock domain to DMA system clock domain synchronizers.
1	When this bit is set, it forces the ADC Controller to bypass the internal double flip-flop synchronizers of the signals from the ADC control clock domain into the DMA system clock domain. It is essential that the user enables FSYDMA only if the DMA system clock is synchronous (edge-aligned) with the ADC control clock and the DMA system clock is faster than or same frequency with the ADC control clock.

Bit 9 – FSYUPB Fast Synchronous UPB Clock bit

Note: ADCCON1.FSYUPB must be '0' when ADCCON1.ADCSEL[1:0] != 0.

Value	Description
0	Fast synchronous UPB clock is disabled
1	Fast synchronous UPB clock is enabled

Bit 8 – SRTGDS SCAN Re-trigger Disable

Value	Description
0	When this bit is cleared, the scan cycle is re-triggerable, which means that if an edge-sensitive scan trigger arrives in the middle of a current scan cycle, the current conversion completes but the rest of the current scan is aborted, the EOS_RDY status bit is asserted and the scan cycle starts again from the beginning with the first channel included in the scan cycle.
1	By setting this bit, the user disallows the scan trigger to re-start the current scan cycle and the current scan cycle completes with its last included channel before getting re-started.

Bits 6:4 – IRQVS[2:0] Interrupt Vector Shift bits

To determine the interrupt vector address, this bit specifies the amount of left-shift done to the ARDYx status bits in the ADCDSTAT1 and ADCDSTAT2 registers prior to adding with the ADCBASE register.

Interrupt Vector Address = Read Value of ADCBASE and Read Value of ADCBASE = Value written to ADCBASE + x << IRQVS[2:0], where 'x' is the smallest active input ID from the ADCDSTAT1 or ADCDSTAT2 registers (which has highest priority).

Value	Description
111	Shift x left 7 bit positions
110	Shift x left 6 bit positions
101	Shift x left 5 bit positions
100	Shift x left 4 bit positions
011	Shift x left 3 bit positions
010	Shift x left 2 bit positions
001	Shift x left 1 bit position
000	Shift x left 0 bit positions

Bit 3 – STRGLVL ScanTrigger High Level/Positive Edge Sensitivity bit

Value	Description
0	Scan trigger is positive edge sensitive. When the STRIG mode is selected (TRGSRCx[4:0] in the ADCTRGx register), only a single scan trigger is generated, which completes the scan of all selected analog inputs.

© 2025 Microchip Technology Inc. and its subsidiaries

Value	Description
1	Scan trigger is high level sensitive. When the STRIG mode is selected (TRGSRCx[4:0] in the ADCTRGx register), the scan trigger continues for all selected analog inputs, until the STRIG option is removed.

Bits 2:0 – DMABL[2:0] DMA Buffer Length Size bits

Note: Since each output data is 16-bit wide, one location consists of 2 bytes.

Value	Description
111	Allocates 128 locations in RAM to each analog input
1110	Allocates 64 locations in RAM to each analog input
101	Allocates 32 locations in RAM to each analog input
100	Allocates 16 locations in RAM to each analog input
011	Allocates 8 locations in RAM to each analog input
010	Allocates 4 locations in RAM to each analog input
001	Allocates 2 locations in RAM to each analog input
000	Allocates 1 locations in RAM to each analog input

© 2025 Microchip Technology Inc. and its subsidiaries

36.13.2. ADCCON2 – ADC Control Register 2

Name: ADCCON2
Offset: 0x10
Reset: 0x00000000
Property: -

This register controls the reference selection for the ADC module, the sample time for the shared ADC module, interrupt enable for reference, early interrupt selection and clock division selection for the shared ADC.

Bit	31	30	29	28	27	26	25	24
	BGVRRDY	REFFLT	EOSRDY	CVD_CPL[2:0]			SAMC[9:8]	
Access	R/HS/HC	R/HS/HC	R/HS/HC	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
	SAMC[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	BGVRIEN	REFFLTIEN	EOSIEN		ENXCNVRT	ADCEIS[2:0]		
Access	R/W	R/W	R/W		R/W	R/W	R/W	R/W
Reset	0	0	0		0	0	0	0

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

Bit 31 – BGVRRDY Band Gap Voltage/ADC Reference Voltage Status bit
Data processing is valid only after BGVRRDY is set by hardware, so the application code must check that the BGVRRDY bit is set to ensure data validity. This bit set to '0' when ON (ADCCON1[15]) = 0.

Value	Description
0	Either or both band gap voltage and ADC reference voltages (V_{REF}) are not ready
1	Both band gap voltage and ADC reference voltages (V_{REF}) are ready

Bit 30 – REFFLT Band Gap/V_{REF}/A_{VDD} BOR Fault Status bit
This bit is cleared when the ON bit (ADCCON1[15]) = 0 and the BGVRRDY bit = 1.

Value	Description
0	Band gap and V_{REF} voltage are working properly
1	Fault in band gap or the V_{REF} voltage while the ON bit (ADCCON1[15]) was set. Most likely a band gap or V_{REF} fault is caused by a BOR of the analog V_{DD} supply.

Bit 29 – EOSRDY End of Scan Interrupt Status bit
This bit is cleared when ADCCON2[31:24] are read in software.

Value	Description
0	Scanning is not complete
1	All analog inputs are considered for scanning through the scan trigger (all analog inputs specified in the ADCCSS1 register) completed scanning

Bits 28:26 – CVD_CPL[2:0] CVD Partly Line Capacitor Setting; Cpline = CVD_CPL[2:0] * 2.5 pF = 0 to 17.5 pF

© 2025 Microchip Technology Inc. and its subsidiaries

Bits 25:16 – SAMC[9:0] SampleTime for the Shared ADC bits
Where T_{AD7} = Period of the ADC conversion clock for the Shared ADC controlled by the ADCCON2.ADCDIV[6:0] bits

Value	Description
1111111111	1025 T_{AD7}
...	—
0000000001	3 T_{AD7}
0000000000	2 T_{AD7}

Bit 15 – BGVRIEN Band Gap/V_{REF} Voltage Ready Interrupt Enable bit

Value	Description
0	No interrupt is generated when the BGVRRDY bit is set
1	Interrupt is generated when the BGVRDDY bit is set

Bit 14 – REFFLTIEN Band Gap/V_{REF} Voltage Fault Interrupt Enable bit

Value	Description
0	No interrupt is generated when the REFFLT bit is set
1	Interrupt is generated when the REFFLT bit is set

Bit 13 – EOSIEN End of Scan Interrupt Enable bit

Value	Description
0	No interrupt is generated when the EOSRDY bit is set
1	Interrupt is generated when the EOSRDY bit is set

Bit 11 – ENXCNVRT Enable External Conversion Request Interface bit
Note: The external module (such as the PTG) is responsible for asserted only the proper trigger signals. This ADC module has no method to block specific trigger requests from the external module.

Value	Description
0	This bit will disable other modules from specifying and requesting conversion of an ADC input.
1	Setting this bit will enable another module (such as the PTG) to specify and request conversion of an ADC input.

Bits 10:8 – ADCEIS[2:0] Early Interrupt Select bits
These early interrupt select bits are for the Shared ADC Core; applies to all shared channels. These bits select the number of clocks prior to the arrival of valid data that the associated interrupt is generated.
The hardware will utilize the maximum allowed early interrupt setting if the user programs the ADCEIS bit-field to a non-allowed value.

Value	Description
111	The early ready interrupt is generated 8 ADC core clocks prior to end of conversion.
110	The early ready interrupt is generated 7 ADC core clocks prior to end of conversion.
101	The early ready interrupt is generated 6 ADC core clocks prior to end of conversion.
100	The early ready interrupt is generated 5 ADC core clocks prior to end of conversion.
011	The early ready interrupt is generated 4 ADC core clocks prior to end of conversion.
010	The early ready interrupt is generated 3 ADC core clocks prior to end of conversion.
001	The early ready interrupt is generated 2 ADC core clocks prior to end of conversion.
000	The early ready interrupt is generated 1 ADC core clocks prior to end of conversion.

© 2025 Microchip Technology Inc. and its subsidiaries

Bits 6:0 – ADCDIV[6:0] Division Ratio for the Shared SAR ADC Core Clock bits
The ADCDIV[6:0] bits divide the ADC control clock (T_Q) to generate the clock for the shared SAR ADC.

Value	Description
1111111	254 * T_Q = T_{AD7}
. . .	—
0000011	6 * T_Q = T_{AD7}
0000010	4 * T_Q = T_{AD7}
0000001	2 * T_Q = T_{AD7}
0000000	Reserved

© 2025 Microchip Technology Inc. and its subsidiaries

36.13.3. ADCCON3 – ADC Control Register 3

Name: ADCCON3
Offset: 0x20
Reset: 0x00000000
Property: -

This register enables ADC clock selection, enables/disables the digital feature for the shared ADC module and controls the manual (software) sampling and conversion.

Note:

1. The SAMP bit has the highest priority, and setting this bit keeps the S&H circuit in Sample mode until the bit is cleared. Also, usage of the SAMP bit causes settings of SAMC[9:0] bits (ADCCON2[25:16]) to be ignored.
2. The SAMP bit only connects Class 2 and Class 3 analog inputs to the shared ADC.
3. The SAMP bit is not a self-clearing bit and it is the responsibility of application software to first clear this bit and, only after setting the RQCNVRT bit, to start the ADC.
4. Normally, when the SAMP and RQCNVRT bits are used by software routines, all TRGSRCx[4:0] bits and STRGSRC[4:0] bits must be set to '00000' to disable all external hardware triggers and prevent them from interfering with the software-controlled sampling command signal SAMP and with the software-controlled trigger RQCNVRT.

Bit	31	30	29	28	27	26	25	24
	ADCSEL[1:0]		CONCLKDIV[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DIGEN7							DIGEN0
Access	R/W							R/W
Reset	0							0

Bit	15	14	13	12	11	10	9	8
	VREFSEL[2:0]			TRGSUSP	UPDIEN	UPDRDY	SAMP	RQCNVRT
Access	R/W	R/W	R/W	R/W	R/W	R/HS/HC	R/W	R/HS/HC
Reset	0	0	0	0	0	0	0	0

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

Bits 31:30 – ADCSEL[1:0] Analog-to-Digital Clock Source (T_{CLK}) bits

Value	Description
00	Peripheral Bus Clock (PB1_CLK)
01	FRC Clock
10	REFO3 Clock Output
11	System Clock (SYS_CLK)

Bits 29:24 – CONCLKDIV[5:0] Analog-to-Digital Control Clock (T_Q) Divider bits

Value	Description
111111	64 * T_{CLK}= T_Q
...	—

© 2025 Microchip Technology Inc. and its subsidiaries

Value	Description
000011	4 * T_{CLK}= T_Q
000010	3 * T_{CLK}= T_Q
000001	2 * T_{CLK}= T_Q
000000	T_{CLK}= T_Q

Bit 23 – DIGEN7 Shared ADC Digital Enable bit

Value	Description
1	ADC is digital enabled
0	ADC is digital disabled

Bit 16 – DIGEN0 Digital Enable First Class Input Channel 0 bit
This bit is dedicated to enable digital functionality for the ADC core. It does not control the core_clk[0]. Only ADANLCTL.ANLEN0 controls core_clk[0]. ADANLCTL.ANLEN0 also qualifies CHN_EN0 if ADANLCTL.ANLEN0 == 0, then also CHN_EN0 == 0.

Bits 15:13 – VREFSEL[2:0] Voltage Reference (V_{REF}) Input Selection bits

Table 36-5.

VREFSEL[2:0]	AD_{REF+}	AD_{REF-}
000	AV_{DD}	AV_{SS}
001–111	Reserved	

Bit 12 – TRGSUSP Trigger Suspend bit

Value	Description
1	Triggers are blocked from starting a new analog-to-digital conversion, but the ADC module is not disabled
0	Triggers are not blocked

Bit 11 – UPDIEN Update Ready Interrupt Enable bit

Value	Description
1	Interrupt is generated when the UPDRDY bit is set by hardware
0	No interrupt is generated

Bit 10 – UPDRDY ADC Update Ready Status bit
Note: This bit is only active while the TRGSUSP bit is set and there are no more running conversions of any ADC modules.

Value	Description
1	ADC SFRs can be updated
0	ADC SFRs cannot be updated

Bit 9 – SAMP Class 2 and Class 3 Analog Input Sampling Enable bit(1,2,3,4)

Value	Description
1	The ADC S&H amplifier is sampling
0	The ADC S&H amplifier is holding

Bit 8 – RQCNVRT Individual ADC Input Conversion Request bit
This bit and its associated ADINSEL[5:0] bits enable the user to individually request an ADC of an analog input through software.
Note: This bit is automatically cleared in the next ADC clock cycle.

Value	Description
1	Trigger the conversion of the selected ADC input as specified by the ADINSEL[5:0] bits

© 2025 Microchip Technology Inc. and its subsidiaries

Value	Description
0	Do not trigger the conversion

Bit 7 – GLSWTRG Global Level Software Trigger bit

Value	Description
1	Trigger conversion for ADC inputs that have selected the GLSWTRG bit as the trigger signal, either through the associated TRGSRC[4:0] bits in the ADCTRGx registers or through the STRGSRC[4:0] bits in the ADCCON1 register
0	Do not trigger an ADC

Bit 6 – GSWTRG Global Software Trigger bit

This bit is automatically cleared in the next ADC clock cycle.

Value	Description
0	Trigger conversion for ADC inputs that have selected the GSWTRG bit as the trigger signal, either through the associated TRGSRC[4:0] bits in the ADCTRGx registers or through the STRGSRC[4:0] bits in the ADCCON1 register
1	Do not trigger an ADC

Bits 5:0 – ADINSEL[5:0] Analog Input Select bits

These bits select the analog input to be converted when the RQCNVRT bit is set.

Value	Description
111111	Reserved
...	—
10110	VDD33/2(AN22) (Internal)
10101	VDD 1.2V(VDD_1V2(AN21)) (Internal)
10100	ADC Charge-pump 1.2V(CP_1V2(AN20)) (Internal)
10011	BandGap Reference (BG_VREF (AN19)) (Internal)
10010	AN18 is being monitored
...	—
000001	AN1 is being monitored
000000	AN0 is being monitored

© 2025 Microchip Technology Inc. and its subsidiaries

36.13.4. ADCTRGMODE - ADC Triggering Mode for Dedicated ADC Register

Name: ADCTRGMODE
Offset: 0x30
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							SH0ALT[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	15	14	13	12	11	10	9	8
								STRGEN0
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
								SSAMPEN0
Access								R/W
Reset								0

Bits 17:16 – SH0ALT[1:0] ADC0 Analog Input Select bit

Value	Description
11	Reserved
10	ANB0
01	ANA0
00	AN0

Bit 8 – STRGEN0 ADC0 Presynchronized Triggers bit

Value	Description
1	ADC0 uses presynchronized triggers
0	ADC0 does not use presynchronized triggers

Bit 0 – SSAMPEN0 ADC0 Synchronous Sampling bit

Value	Description
1	ADC0 uses synchronous sampling for the first sample after being idle or disabled
0	ADC0 does not use synchronous sampling

© 2025 Microchip Technology Inc. and its subsidiaries

36.13.5. ADCIMCON1 – ADC Input Mode Control Register 1

Name: ADCIMCON1
Offset: 0x40
Reset: 0x00000000
Property: -

This register enables the user to select between single-ended and differential operation as well as select between signed and unsigned data format.

Bit	31	30	29	28	27	26	25	24
	DIFF15	SIGN15	DIFF14	SIGN14	DIFF13	SIGN13	DIFF12	SIGN12
Access	R/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
	DIFF11	SIGN11	DIFF10	SIGN10	DIFF9	SIGN9	DIFF8	SIGN8
Access	R/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
	DIFF7	SIGN7	DIFF6	SIGN6	DIFF5	SIGN5	DIFF4	SIGN4
Access	R/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
	DIFF3	SIGN3	DIFF2	SIGN2	DIFF1	SIGN1	DIFF0	SIGN0
Access	R/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 31 – DIFF15 AN15 Mode bit

Value	Description
1	AN15 is using Differential mode
0	AN15 is using Single-ended mode

Bit 30 – SIGN15 AN15 Signed Data Mode bit

Value	Description
1	AN15 is using Signed Data mode
0	AN15 is using Unsigned Data mode

Bit 29 – DIFF14 AN14 Mode bit

Value	Description
1	AN14 is using Differential mode
0	AN14 is using Single-ended mode

Bit 28 – SIGN14 AN14 Signed Data Mode bit

Value	Description
1	AN14 is using Signed Data mode
0	AN14 is using Unsigned Data mode

Bit 27 – DIFF13 AN13 Mode bit

© 2025 Microchip Technology Inc. and its subsidiaries

Value	Description
1	AN13 is using Differential mode
0	AN13 is using Single-ended mode

Bit 26 – SIGN13 AN13 Signed Data Mode bit

Value	Description
1	AN13 is using Signed Data mode
0	AN13 is using Unsigned Data mode

Bit 25 – DIFF12 AN12 Mode bit

Value	Description
1	AN12 is using Differential mode
0	AN12 is using Single-ended mode

Bit 24 – SIGN12 AN12 Signed Data Mode bit

Value	Description
1	AN12 is using Signed Data mode
0	AN12 is using Unsigned Data mode

Bit 23 – DIFF11 AN11 Mode bit

Value	Description
1	AN11 is using Differential mode
0	AN11 is using Single-ended mode

Bit 22 – SIGN11 AN11 Signed Data Mode bit

Value	Description
1	AN11 is using Signed Data mode
0	AN11 is using Unsigned Data mode

Bit 21 – DIFF10 AN10 Mode bit

Value	Description
1	AN10 is using Differential mode
0	AN10 is using Single-ended mode

Bit 20 – SIGN10 AN10 Signed Data Mode bit

Value	Description
1	AN10 is using Signed Data mode
0	AN10 is using Unsigned Data mode

Bit 19 – DIFF9 AN9 Mode bit

Value	Description
1	AN9 is using Differential mode
0	AN9 is using Single-ended mode

Bit 18 – SIGN9 AN9 Signed Data Mode bit

Value	Description
1	AN9 is using Signed Data mode
0	AN9 is using Unsigned Data mode

Bit 17 – DIFF8 AN8 Mode bit

© 2025 Microchip Technology Inc. and its subsidiaries

Value	Description
1	AN8 is using Differential mode
0	AN8 is using Single-ended mode

Bit 16 – SIGN8 AN8 Signed Data Mode bit

Value	Description
1	AN8 is using Signed Data mode
0	AN8 is using Unsigned Data mode

Bit 15 – DIFF7 AN7 Mode bit

Value	Description
1	AN7 is using Differential mode
0	AN7 is using Single-ended mode

Bit 14 – SIGN7 AN7 Signed Data Mode bit

Value	Description
1	AN7 is using Signed Data mode
0	AN7 is using Unsigned Data mode

Bit 13 – DIFF6 AN6 Mode bit

Value	Description
1	AN6 is using Differential mode
0	AN6 is using Single-ended mode

Bit 12 – SIGN6 AN6 Signed Data Mode bit

Value	Description
1	AN6 is using Signed Data mode
0	AN6 is using Unsigned Data mode

Bit 11 – DIFF5 AN5 Mode bit

Value	Description
1	AN5 is using Differential mode
0	AN5 is using Single-ended mode

Bit 10 – SIGN5 AN5 Signed Data Mode bit

Value	Description
1	AN5 is using Signed Data mode
0	AN5 is using Unsigned Data mode

Bit 9 – DIFF4 AN4 Mode bit

Value	Description
1	AN4 is using Differential mode
0	AN4 is using Single-ended mode

Bit 8 – SIGN4 AN4 Signed Data Mode bit

Value	Description
1	AN4 is using Signed Data mode
0	AN4 is using Unsigned Data mode

Bit 7 – DIFF3 AN3 Mode bit

© 2025 Microchip Technology Inc. and its subsidiaries

Value	Description
1	AN3 is using Differential mode
0	AN3 is using Single-ended mode

Bit 6 – SIGN3 AN3 Signed Data Mode bit

Value	Description
1	AN3 is using Signed Data mode
0	AN3 is using Unsigned Data mode

Bit 5 – DIFF2 AN2 Mode bit

Value	Description
1	AN2 is using Differential mode
0	AN2 is using Single-ended mode

Bit 4 – SIGN2 AN2 Signed Data Mode bit

Value	Description
1	AN2 is using Signed Data mode
0	AN2 is using Unsigned Data mode

Bit 3 – DIFF1 AN1 Mode bit

Value	Description
1	AN1 is using Differential mode
0	AN1 is using Single-ended mode

Bit 2 – SIGN1 AN1 Signed Data Mode bit

Value	Description
1	AN1 is using Signed Data mode
0	AN1 is using Unsigned Data mode

Bit 1 – DIFF0 AN0 Mode bit

Value	Description
1	AN0 is using Differential mode
0	AN0 is using Single-ended mode

Bit 0 – SIGN0 AN0 Signed Data Mode bit

Value	Description
1	AN0 is using Signed Data mode
0	AN0 is using Unsigned Data mode

© 2025 Microchip Technology Inc. and its subsidiaries

36.13.6. ADCIMCON2 – ADC Input Mode Control Register 2

Name: ADCIMCON2
Offset: 0x50
Reset: 0x00000000
Property: -

This register enables the user to select between single-ended and differential operation as well as select between signed and unsigned data format.

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
			DIFF22	SIGN22	DIFF21	SIGN21	DIFF20	SIGN20
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DIFF19	SIGN19	DIFF18	SIGN18	DIFF17	SIGN17	DIFF16	SIGN16
Access	R/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 13 – DIFF22 AN22 Mode bit

Value	Description
1	AN22 is using Differential mode
0	AN22 is using Single-ended mode

Bit 12 – SIGN22 AN22 Signed Data Mode bit

Value	Description
1	AN22 is using Signed Data mode
0	AN22 is using Unsigned Data mode

Bit 11 – DIFF21 AN21 Mode bit

Value	Description
1	AN21 is using Differential mode
0	AN21 is using Single-ended mode

Bit 10 – SIGN21 AN21 Signed Data Mode bit

Value	Description
1	AN21 is using Signed Data mode
0	AN21 is using Unsigned Data mode

Bit 9 – DIFF20 AN20 Mode bit

© 2025 Microchip Technology Inc. and its subsidiaries

Value	Description
1	AN20 is using Differential mode
0	AN20 is using Single-ended mode

Bit 8 – SIGN20 AN20 Signed Data Mode bit

Value	Description
1	AN20 is using Signed Data mode
0	AN20 is using Unsigned Data mode

Bit 7 – DIFF19 AN19 Mode bit

Value	Description
1	AN19 is using Differential mode
0	AN19 is using Single-ended mode

Bit 6 – SIGN19 AN19 Signed Data Mode bit

Value	Description
1	AN19 is using Signed Data mode
0	AN19 is using Unsigned Data mode

Bit 5 – DIFF18 AN18 Mode bit

Value	Description
1	AN18 is using Differential mode
0	AN18 is using Single-ended mode

Bit 4 – SIGN18 AN18 Signed Data Mode bit

Value	Description
1	AN18 is using Signed Data mode
0	AN18 is using Unsigned Data mode

Bit 3 – DIFF17 AN17 Mode bit

Value	Description
1	AN17 is using Differential mode
0	AN17 is using Single-ended mode

Bit 2 – SIGN17 AN17 Signed Data Mode bit

Value	Description
1	AN17 is using Signed Data mode
0	AN17 is using Unsigned Data mode

Bit 1 – DIFF16 AN16 Mode bit

Value	Description
1	AN16 is using Differential mode
0	AN16 is using Single-ended mode

Bit 0 – SIGN16 AN16 Signed Data Mode bit

Value	Description
1	AN16 is using Signed Data mode
0	AN16 is using Unsigned Data mode

© 2025 Microchip Technology Inc. and its subsidiaries

36.13.7. ADCGIRQEN1 – ADC Global Interrupt Enable Register 1

Name: ADCGIRQEN1
Offset: 0x80
Reset: 0x00000000
Property: -

This register specifies which of the individual input conversion interrupts can generate the global ADC interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
		AGIEN22	AGIEN21	AGIEN20	AGIEN19	AGIEN18	AGIEN17	AGIEN16
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	AGIEN15	AGIEN14	AGIEN13	AGIEN12	AGIEN11	AGIEN10	AGIEN9	AGIEN8
Access	R/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
	AGIEN7	AGIEN6	AGIEN5	AGIEN4	AGIEN3	AGIEN2	AGIEN1	AGIEN0
Access	R/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 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22 – AGIEN ADC Global Interrupt Enable bits

Value	Description
1	Interrupts are enabled for the selected analog input. The interrupt is generated after the converted data is ready (indicated by the ARDYx bit ('x' = 8-1) of the ADCDSTAT1 register)
0	Interrupts are disabled

© 2025 Microchip Technology Inc. and its subsidiaries

36.13.8. ADCCSS1 – ADC Common Scan Select Register 1

Name: ADCCSS1
Offset: 0xA0
Reset: 0x00000000
Property: -

This register specifies the analog inputs to be scanned by the common scan trigger.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
		CSS22	CSS21	CSS20	CSS19	CSS18	CSS17	CSS16
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CSS15	CSS14	CSS13	CSS12	CSS11	CSS10	CSS9	CSS8
Access	R/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
	CSS7	CSS6	CSS5	CSS4	CSS3	CSS2	CSS1	CSS0
Access	R/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 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22 – CSS Analog Common Scan Select bits

Notes:

1. In addition to setting the appropriate bits in this register, Class 2 analog inputs must select the STRIG input as the trigger source if they are to be scanned through the CSSx bits. Refer to the bit descriptions in the ADCTRGx registers for selecting the STRIG option.
2. If a Class 2 input is included in the scan by setting the CSSx bit to '1' and by setting the TRGSRCx[4:0] bits to STRIG mode (0b11), the user application must ensure that no other triggers are generated for that input using the RQCNVRT bit in the ADCCON3 register or the hardware input or any digital filter. Otherwise, the scan behavior is unpredictable.

Value	Description
1	Select ANx for input scan
0	Skip ANx for input scan

© 2025 Microchip Technology Inc. and its subsidiaries

36.13.9. ADCDSTAT1 – ADC Data Ready Status Register 1

Name: ADCDSTAT1
Offset: 0xC0
Reset: 0x00000000
Property: -

This register contains the interrupt status of the individual analog input conversions. Each bit represents the data-ready status for its associated conversion result.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
		ARDY22	ARDY21	ARDY20	ARDY19	ARDY18	ARDY17	ARDY16
Access		R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ARDY15	ARDY14	ARDY13	ARDY12	ARDY11	ARDY10	ARDY9	ARDY8
Access	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ARDY7	ARDY6	ARDY5	ARDY4	ARDY3	ARDY2	ARDY1	ARDY0
Access	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22 – ARDY Conversion Data Ready for Corresponding Analog Input Ready bits

Value	Description
1	This bit is set when converted data is ready in the data register
0	This bit is cleared when the associated data register is read

© 2025 Microchip Technology Inc. and its subsidiaries

36.13.10. ADCCMPEN1 – ADC Digital Comparator 1 Enable Register

Name: ADCCMPEN1
Offset: 0xE0
Reset: 0x00000000
Property: -

These registers select which analog input conversion result is processed by the digital comparator.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
		CMPE22	CMPE21	CMPE20	CMPE19	CMPE18	CMPE17	CMPE16
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CMPE15	CMPE14	CMPE13	CMPE12	CMPE11	CMPE10	CMPE9	CMPE8
Access	R/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
	CMPE7	CMPE6	CMPE5	CMPE4	CMPE3	CMPE2	CMPE1	CMPE0
Access	R/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 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22 – CMPE ADC Digital Comparator x Enable bits

Note: CMPEx = where x stands for bit value from 0 to 22

These bits enable conversion results corresponding to the analog input to be processed by the digital comparator. CMPE0 enables AN0, CMPE1 enables AN1 and so on.

Notes:

1. CMPEx = ANx, where x = 0-22 (Digital Comparator inputs are limited to AN0 through AN22)
2. Changing the bits in this register while the Digital Comparator is enabled (ENDCMP = 1) can result in unpredictable behavior.

© 2025 Microchip Technology Inc. and its subsidiaries

36.13.11. ADCCMP1 – ADC Digital Comparator 1 Limit Value Register

Name: ADCCMP1
Offset: 0xF0
Reset: 0x00000000
Property: -

These registers contain the high and low digital comparison values for use by the digital comparator.
Notes:

1. Changing theses bits while the Digital Comparator is enabled (ENDCMP = 1) can result in unpredictable behavior.
2. The format of the limit values must match the format of the ADC converted value in terms of sign and fractional settings.
3. For Digital Comparator 1 used in the CVD mode, the DCMPHI[15:0] and DCMPLO[15:0] bits must always be specified in signed format as the CVD output data is differential and is always signed.

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

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

Bit	15	14	13	12	11	10	9	8
	ADCMPLO[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
	ADCMPLO[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – ADCMPHI[15:0] Digital Comparator x High Limit Value bits
These bits store the high limit value, which is used by the digital comparator for comparisons with ADC converted data.

Bits 15:0 – ADCMPLO[15:0] Digital Comparator x Low Limit Value bits
These bits store the low limit value, which is used by digital comparator for comparisons with ADC converted data.

© 2025 Microchip Technology Inc. and its subsidiaries

36.13.12. ADCCMPEN2 – ADC Digital Comparator 2 Enable Register

Name: ADCCMPEN2
Offset: 0x100
Reset: 0x00000000
Property: -

These registers select which analog input conversion result is processed by the digital comparator.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
		CMPEx22	CMPEx21	CMPEx20	CMPEx19	CMPEx18	CMPEx17	CMPEx16
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CMPEx15	CMPEx14	CMPEx13	CMPEx12	CMPEx11	CMPEx10	CMPEx9	CMPEx8
Access	R/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
	CMPEx7	CMPEx6	CMPEx5	CMPEx4	CMPEx3	CMPEx2	CMPEx1	CMPEx0
Access	R/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 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22 – CMPEx ADC Digital Comparator x Enable bits

Note: CMPEx = where x stands for the bit value from 0 to 22

These bits enable conversion results corresponding to the analog input to be processed by the digital comparator. CMPE0 enables AN0, CMPE1 enables AN1 and so on.

Notes:

1. CMPEx = ANx, where x = 0-22 (Digital Comparator inputs are limited to AN0 through AN22)
2. Changing the bits in this register while the Digital Comparator is enabled (ENDCMP = 1) can result in unpredictable behavior.

© 2025 Microchip Technology Inc. and its subsidiaries

36.13.13. ADCCMP2 – ADC Digital Comparator 2 Limit Value Register

Name: ADCCMP2
Offset: 0x110
Reset: 0x00000000
Property: -

These registers contain the high and low digital comparison values for use by the digital comparator.
Notes:

1. Changing theses bits while the Digital Comparator is enabled (ENDCMP = 1) can result in unpredictable behavior.
2. The format of the limit values must match the format of the ADC converted value in terms of sign and fractional settings.

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

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

Bit	15	14	13	12	11	10	9	8
	ADCMPLO[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
	ADCMPLO[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – ADCMPHI[15:0] Digital Comparator x High Limit Value bits
These bits store the high limit value, which is used by digital comparator for comparisons with ADC converted data.

Bits 15:0 – ADCMPLO[15:0] Digital Comparator x Low Limit Value bits
These bits store the low limit value, which is used by digital comparator for comparisons with ADC converted data.

© 2025 Microchip Technology Inc. and its subsidiaries

36.13.14. ADCFLTR1 – ADC Digital Filter 1 Register

Name: ADCFLTR1
Offset: 0x1A0
Reset: 0x00000000
Property: -

These registers provide control and status bits for the oversampling filter accumulator and also include the 16-bit filter output data.

Bit	31	30	29	28	27	26	25	24
	AFEN0	DATA16EN0	ADFMODE0	OVRSAM0[2:0]			AFIEN0	AFIF0
Access	R/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
				FLTINID0[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	FLT_DATA0[15:8]							
Access	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FLT_DATA0[7:0]							
Access	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC
Reset	0	0	0	0	0	0	0	0

Bit 31 – AFEN0 Digital Filter x Enable bit

Value	Description
1	Digital filter is enabled
0	Digital filter is disabled and the AFRDY status bit is cleared

Bit 30 – DATA16EN0 Filter Significant Data Length bit

Note: This bit is significant only if DFMODE = 1 (Averaging mode) and FRACT (ADCCON1[23]) = 1 (Fractional Output mode).

Value	Description
1	All 16 bits of the filter output data are significant
0	Only the first 12 bits are significant, followed by four zeros

Bit 29 – ADFMODE0 ADC Filter Mode bit

Value	Description
1	Filter x works in Averaging mode
0	Filter x works in Oversampling Filter mode (default)

Bits 28:26 – OVRSAM0[2:0] Oversampling Filter Ratio bits

Value	Description
	If DFMODE is '0'
111	128 samples (shift sum 3 bits to right, output data is in 15.1 format)
110	32 samples (shift sum 2 bits to right, output data is in 14.1 format)
101	8 samples (shift sum 1 bit to right, output data is in 13.1 format)

© 2025 Microchip Technology Inc. and its subsidiaries

Value	Description
100	2 samples (shift sum 0 bits to right, output data is in 12.1 format)
011	256 samples (shift sum 4 bits to right, output data is 16 bits)
010	64 samples (shift sum 3 bits to right, output data is 15 bits)
001	16 samples (shift sum 2 bits to right, output data is 14 bits)
000	4 samples (shift sum 1 bit to right, output data is 13 bits)
	If DFMODE is '1'
111	256 samples (256 samples to be averaged)
110	128 samples (128 samples to be averaged)
101	64 samples (64 samples to be averaged)
100	32 samples (32 samples to be averaged)
011	16 samples (16 samples to be averaged)
010	8 samples (8 samples to be averaged)
001	4 samples (4 samples to be averaged)
000	2 samples (2 samples to be averaged)

Bit 25 – AFIEN0 Digital Filter x Interrupt Enable bit

Value	Description
1	Digital filter interrupt is enabled and is generated by the AFRDY status bit
0	Digital filter is disabled

Bit 24 – AFIF0 Digital Filter x Data Ready Status bit

Note: This bit is cleared by reading the FLTRDATA[15:0] bits or by disabling the Digital Filter module (by setting AFEN to '0').

Value	Description
1	Data is ready in the FLTRDATA[15:0] bits
0	Data is not ready

Bits 20:16 – FLTINID0[4:0] Digital Filter Analog Input Selection bits

Note: Only the first 8 analog inputs, Class 2 (AN0 -AN7), can use a digital filter.

These bits specify the analog input to be used as the oversampling filter data source.

Value	Description
11111	Reserved
...	—
...	—
...	—
00111	AN7
...	—
...	—
...	—
00010	AN2
00001	AN1
00000	AN0

Bits 15:0 – FLT_DATA0[15:0] Digital Filter x Data Output Value bits

The filter output data is as per the fractional format set in the FRACT bit (ADCCON1[23]). The FRACT bit must not be changed while the filter is enabled. Changing the state of the FRACT bit after the operation of the filter ended must not update the value of the FLTRDATA[15:0] bits to reflect the new format.

© 2025 Microchip Technology Inc. and its subsidiaries

36.13.15. ADCFLTR2 – ADC Digital Filter 2 Register

Name: ADCFLTR2
Offset: 0x1B0
Reset: 0x00000000
Property: -

These registers provide control and status bits for the oversampling filter accumulator and also include the 16-bit filter output data.

Bit	31	30	29	28	27	26	25	24
	AFEN1	DATA16EN1	ADFMODE1	OVRSAM1[2:0]			AFIEN1	AFIF1
Access	R/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
				FLTINID1[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	FLT_DATA1[15:8]							
Access	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FLT_DATA1[7:0]							
Access	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC
Reset	0	0	0	0	0	0	0	0

Bit 31 – AFEN1 Digital Filter x Enable bit

Value	Description
1	Digital filter is enabled
0	Digital filter is disabled and the AFRDY status bit is cleared

Bit 30 – DATA16EN1 Filter Significant Data Length bit

Note: This bit is significant only if DFMODE = 1 (Averaging mode) and FRACT (ADCCON1[23]) = 1 (Fractional Output mode).

Value	Description
1	All 16 bits of the filter output data are significant
0	Only the first 12 bits are significant, followed by four '0'

Bit 29 – ADFMODE1 ADC Filter Mode bit

Value	Description
1	Filter x works in Averaging mode
0	Filter x works in Oversampling Filter mode (default)

Bits 28:26 – OVRSAM1[2:0] Oversampling Filter Ratio bits

Value	Description
	If DFMODE is '0'
111	128 samples (shift sum 3 bits to right, output data is in 15.1 format)
110	32 samples (shift sum 2 bits to right, output data is in 14.1 format)

© 2025 Microchip Technology Inc. and its subsidiaries

Value	Description
101	8 samples (shift sum 1 bit to right, output data is in 13.1 format)
100	2 samples (shift sum 0 bits to right, output data is in 12.1 format)
011	256 samples (shift sum 4 bits to right, output data is 16 bits)
010	64 samples (shift sum 3 bits to right, output data is 15 bits)
001	16 samples (shift sum 2 bits to right, output data is 14 bits)
000	4 samples (shift sum 1 bit to right, output data is 13 bits)
	If DFMODE is '1'
111	256 samples (256 samples to be averaged)
110	128 samples (128 samples to be averaged)
101	64 samples (64 samples to be averaged)
100	32 samples (32 samples to be averaged)
011	16 samples (16 samples to be averaged)
010	8 samples (8 samples to be averaged)
001	4 samples (4 samples to be averaged)
000	2 samples (2 samples to be averaged)

Bit 25 – AFIEN1 Digital Filter x Interrupt Enable bit

Value	Description
1	Digital filter interrupt is enabled and is generated by the AFRDY status bit
0	Digital filter is disabled

Bit 24 – AFIF1 Digital Filter x Data Ready Status bit

Note: This bit is cleared by reading the FLTRDATA[15:0] bits or by disabling the Digital Filter module (by setting AFEN to '0').

Value	Description
1	Data is ready in the FLTRDATA[15:0] bits
0	Data is not ready

Bits 20:16 – FLTINID1[4:0] Digital Filter Analog Input Selection bits

Note: Only the first 8 analog inputs, Class 2 (AN0-AN7), can use a digital filter.

These bits specify the analog input to be used as the oversampling filter data source.

Value	Description
11111	Reserved
...	—
...	—
...	—
01100	Reserved
0111	AN7
...	—
...	—
...	—
00010	AN2
00001	AN1
00000	AN0

Bits 15:0 – FLT_DATA1[15:0] Digital Filter x Data Output Value bits

The filter output data is as per the fractional format set in the FRACT bit (ADCCON1[23]). The FRACT bit must not be changed while the filter is enabled. Changing the state of the FRACT bit after the operation of the filter ended must not update the value of the FLTRDATA[15:0] bits to reflect the new format.

© 2025 Microchip Technology Inc. and its subsidiaries

36.13.16. ADCTRG1 – ADC Trigger Source 1 Register

Name: ADCTRG1
Offset: 0x200
Reset: 0x00000000
Property: -

This register controls the trigger source selection for AN0 through AN3 analog inputs.

Bit	31	30	29	28	27	26	25	24
				TRGSRC3[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
				TRGSRC2[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
				TRGSRC1[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
				TRGSRC0[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bits 28:24 – TRGSRC3[4:0] Trigger Source for Conversion of Analog Input AN3 Select bits
Note: For STRIG, in addition to setting the trigger, it also requires programming of the STRGSRC[4:0] bits (ADCCON1[20:16]) to select the trigger source and requires the appropriate CSS bits to be set in the ADCCSSx registers.

Value	Description
10001 – 11111	Reserved
10000	EVSYS_53
01111	EVSYS_52
01110	EVSYS_51
01101	EVSYS_50
01100	EVSYS_49
01011	EVSYS_48
01010	EVSYS_47
01001	EVSYS_46
01000	EVSYS_45
00111	EVSYS_44
00110	EVSYS_43
00101	EVSYS_42
00100	INT0 External interrupt
00011	STRIG
00010	Global level software trigger (GLSWTRG)
00001	Global software edge trigger (GSWTRG)
00000	No Trigger

© 2025 Microchip Technology Inc. and its subsidiaries

Bits 20:16 – TRGSRC2[4:0] Trigger Source for Conversion of Analog Input AN2 Select bits
Note: See Bits 28-24 for bit value definitions.

Bits 12:8 – TRGSRC1[4:0] Trigger Source for Conversion of Analog Input AN1 Select bits
Note: See Bits 28-24 for bit value definitions.

Bits 4:0 – TRGSRC0[4:0] Trigger Source for Conversion of Analog Input AN0 Select bits
Note: See Bits 28-24 for bit value definitions.

© 2025 Microchip Technology Inc. and its subsidiaries

36.13.17. ADCTRG2 – ADC Trigger Source 2 Register

Name: ADCTRG2
Offset: 0x210
Reset: 0x00000000
Property: -

This register controls the trigger source selection for AN4 through AN7 analog inputs.

Bit	31	30	29	28	27	26	25	24
				TRGSRC7[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
				TRGSRC6[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
				TRGSRC5[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
				TRGSRC4[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bits 28:24 – TRGSRC7[4:0] Trigger Source for Conversion of Analog Input AN7 Select bits
Note: For STRIG, in addition to setting the trigger, it also requires programming of the STRGSRC[4:0] bits (ADCCON1[20:16]) to select the trigger source and requires the appropriate CSS bits to be set in the ADCCSSx registers.

Value	Description
10001 – 11111	Reserved
10000	EVSYS_53
01111	EVSYS_52
01110	EVSYS_51
01101	EVSYS_50
01100	EVSYS_49
01011	EVSYS_48
01010	EVSYS_47
01001	EVSYS_46
01000	EVSYS_45
00111	EVSYS_44
00110	EVSYS_43
00101	EVSYS_42
00100	INT0 External interrupt
00011	STRIG
00010	Global level software trigger (GLSWTRG)
00001	Global software edge trigger (GSWTRG)
00000	No Trigger

© 2025 Microchip Technology Inc. and its subsidiaries

Bits 20:16 – TRGSRC6[4:0] Trigger Source for Conversion of Analog Input AN6 Select bits
Note: See Bits 28-24 for bit value definitions.

Bits 12:8 – TRGSRC5[4:0] Trigger Source for Conversion of Analog Input AN5 Select bits
Note: See Bits 28-24 for bit value definitions.

Bits 4:0 – TRGSRC4[4:0] Trigger Source for Conversion of Analog Input AN4 Select bits
Note: See Bits 28-24 for bit value definitions.

© 2025 Microchip Technology Inc. and its subsidiaries

36.13.18. ADCCMPCON1 – ADC Digital Comparator 1 Control Register

Name: ADCCMPCON1
Offset: 0x280
Reset: 0x00000000
Property: -

This register controls the operation of Digital Comparator 1, including the generation of interrupts, comparison criteria to be used and provides status when a comparator event occurs.

Bit	31	30	29	28	27	26	25	24
	CVD_DATA[15:8]							
Access	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CVD_DATA[7:0]							
Access	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
			CMPINID0[5:0]					
Access			R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ENDCMP	DCMPGIEN	DCMPED	IEBTWN	IEHIHI	IEHILO	IELOHI	IELOLO
Access	R/W	R/W	R/HS/HC	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – CVD_DATA[15:0] CVD Differential Output Data
In the CVD mode, this 16-bit field gets the CVD differential output data whenever a DCMPED interrupt is generated. The value in this field is ADCCON1.FRACT-compliant and always signed because it is the result of the subtraction between the CVD positive and negative measurements.

Bits 13:8 – CMPINID0[5:0] Comparator's Analog Input ID (Identification) bits
When a DCMP interrupt is generated, this read only bit field contains the identification number of the analog input being monitored by the digital comparator #0.
Note: In normal ADC mode, only analog inputs [31:0] can be processed by the digital comparator module #0. The digital comparator #0 also supports the CVD mode, in which all second and third class channel IDs may be stored in the CMPINID0[5:0] field (which therefore must be 6 bit wide).

Value	Description
11111	Reserved
. . .	—
. . .	—
. . .	—
10110	AN22 is being monitored
10101	AN21 is being monitored
10100	AN20 is being monitored
10011	AN19 is being monitored
10010	AN18 is being monitored
10001	AN17 is being monitored
10000	AN16 is being monitored
01111	AN15 is being monitored

© 2025 Microchip Technology Inc. and its subsidiaries

Value	Description
01110	AN14 is being monitored
01101	AN13 is being monitored
01100	AN12 is being monitored
01011	AN11 is being monitored
. . .	—
01000	AN8 is being monitored
00111	AN7 is being monitored
. . .	—
00001	AN1 is being monitored
00000	AN0 is being monitored

Bit 7 – ENDCMP Digital Comparator 1 Enable bit

Value	Description
1	Digital Comparator 1 is enabled
0	Digital Comparator 1 is not enabled, and the DCMPED status bit (ADCCMPCON[5]) is cleared

Bit 6 – DCMPGIEN Digital Comparator 1 Global Interrupt Enable bit

Value	Description
1	A Digital Comparator 1 interrupt is generated when the DCMPED status bit (ADCCMPCON[5]) is set
0	A Digital Comparator 1 interrupt is disabled

Bit 5 – DCMPED Digital Comparator 1 Output True Event Status bit

The logical conditions where the digital comparator becomes True are defined by the IEBTWN, IEHIHI, IEHILO, IELOHI and IELOLO bits.

Note: This bit is cleared by reading the AINID[4:0] bits or by disabling the Digital Comparator module (by setting ENDCMP to '0').

Value	Description
1	Digital Comparator 1 output true event has occurred (output of comparator is '1')
0	Digital Comparator 1 output is false (output of comparator is '0')

Bit 4 – IEBTWN Between Low/High Digital Comparator 1 Event bit

Value	Description
1	Generate a digital comparator event when DATA[31:0] is less than DCMPHI[15:0] and greater than DCMPLO[15:0]
0	Do not generate a digital comparator event

Bit 3 – IEHIHI High/High Digital Comparator 1 Event bit

Value	Description
1	Generate a Digital Comparator 1 event when DCMPHI[15:0] bits are less than or equal to DATA[31:0] bits
0	Do not generate an event

Bit 2 – IEHILO High/Low Digital Comparator 1 Event bit

Value	Description
1	Generate a Digital Comparator 1 event when DATA[31:0] bits are less than DCMPHI[15:0] bits
0	Do not generate an event

Bit 1 – IELOHI Low/High Digital Comparator 1 Event bit

Value	Description
1	Generate a Digital Comparator 1 event when DCMPLO[15:0] bits are less than or equal to DATA[31:0] bits
0	Do not generate an event

© 2025 Microchip Technology Inc. and its subsidiaries

Bit 0 – IELOLO Low/Low Digital Comparator 1 Event bit

Value	Description
1	Generate a Digital Comparator 1 event when DATA[31:0] bits are less than DCMPLO[15:0] bits
0	Do not generate an event

© 2025 Microchip Technology Inc. and its subsidiaries

36.13.19. ADCCMPCON2 – ADC Digital Comparator 2 Control Register

Name: ADCCMPCON2
Offset: 0x290
Reset: 0x00000000
Property: -

These registers control the operation of Digital Comparator 2, including the generation of interrupts and the comparison criteria to be used. This register also provides the status when a comparator event occurs.

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
				AINID[4:0]				
Access				R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ENDCMP	DCMPGIEN	DCMPED	IEBTWN	IEHIHI	IEHILO	IELOHI	IELOLO
Access	R/W	R/W	R/HS/HC	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 12:8 – AINID[4:0] Digital Comparator 2 Analog Input Identification (ID) bits
When a digital comparator event occurs (DCMPED = 1), these bits identify the analog input being monitored by the digital comparator.

Value	Description
11111	Reserved
10110	AN22 is being monitored
10101	AN21 is being monitored
10100	AN20 is being monitored
10011	AN19 is being monitored
10010	AN18 is being monitored
10001	AN17 is being monitored
10000	AN16 is being monitored
01111	AN15 is being monitored
01110	AN14 is being monitored
01101	AN13 is being monitored
01100	AN12 is being monitored
01011	AN11 is being monitored
...	—
...	—
...	—
00111	AN7 is being monitored
...	—
00001	AN1 is being monitored

© 2025 Microchip Technology Inc. and its subsidiaries

Value	Description
00000	AN0 is being monitored

Bit 7 – ENDCMP Digital Comparator 2 Enable bit

Value	Description
1	Digital Comparator 2 is enabled
0	Digital Comparator 2 is not enabled, and the DCMPED status bit is cleared

Bit 6 – DCMPGIEN Digital Comparator 2 Global Interrupt Enable bit

Value	Description
1	Digital Comparator 2 interrupt is generated when the DCMPED status bit is set
0	Digital Comparator 2 interrupt is disabled

Bit 5 – DCMPED Digital Comparator 2 Output True Event Status bit

The logical conditions where the digital comparator gets True are defined by the IEBTWN, IEHIHI, IEHILO, IELOHI and IELOLO bits.

Note: This bit is cleared by reading the AINID[4:0] bits or by disabling the Digital Comparator module (by setting ENDCMP to '0').

Value	Description
1	Digital Comparator 2 output true event has occurred (output of comparator is '1')
0	Digital Comparator 2 output is false (output of comparator is '0')

Bit 4 – IEBTWN Between Low/High Digital Comparator 2 Event bit

Value	Description
1	Generate a digital comparator event when DCMPLO[15:0] bits DATA[31:0] bits [DCMPHI[15:0] bits
0	Do not generate a digital comparator event

Bit 3 – IEHIHI High/High Digital Comparator 2 Event bit

Value	Description
1	Generate a Digital Comparator 2 event when DCMPHI[15:0] bits are less than or equal to DATA[31:0] bits
0	Do not generate an event

Bit 2 – IEHILO High/Low Digital Comparator 2 Event bit

Value	Description
1	Generate a Digital Comparator 2 event when DATA[31:0] bits are less than DCMPHI[15:0] bits
0	Do not generate an event

Bit 1 – IELOHI Low/High Digital Comparator 2 Event bit

Value	Description
1	Generate a Digital Comparator 2 event when DCMPLO[15:0] bits are less than or equal to DATA[31:0] bits
0	Do not generate an event

Bit 0 – IELOLO Low/Low Digital Comparator 2 Event bit

Value	Description
1	Generate a Digital Comparator 2 event when DATA[31:0] bits are less than DCMPLO[15:0] bits
0	Do not generate an event

© 2025 Microchip Technology Inc. and its subsidiaries

36.13.20. ADCBASE – ADC Base Register

Name: ADCBASE
Offset: 0x300
Reset: 0x00000000
Property: -

This register specifies the base address of the user ADC Interrupt Service Routine (ISR) jump table.

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
	ADCBASE[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
	ADCBASE[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 – ADCBASE[15:0] ADCISR Base Address bits

This register, when read, contains the base address of the user's ADC ISR jump table. The interrupt vector address is determined by the IRQVS[2:0] bits of the ADCCON1 register specifying the amount of left shift done to the ARDYx status bits in the ADCDSTAT1 register, prior to adding with ADCBASE register.

Interrupt vector address = Read value of ADCBASE

Read value of ADCBASE = Value written to ADCBASE + x << ADCCON1.IRQVS[2:0], where x is the smallest active analog input ID from the ADCDSTAT1 register (which has the highest priority).

© 2025 Microchip Technology Inc. and its subsidiaries

36.13.21. ADCDMASTAT – ADC DMA Status Register

Name: ADCDMASTAT
Offset: 0x310
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	DMAGEN			RBF_IEN0				
Access	R/W			R/W				
Reset	0			0				

Bit	23	22	21	20	19	18	17	16
	WR_OVF_ERR							RBF0
Access	R/W							R/W
Reset	0							0

Bit	15	14	13	12	11	10	9	8
	DMA_CNT_EN							RAF_IEN0
Access	R/W							R/W
Reset	0							0

Bit	7	6	5	4	3	2	1	0
								RAF0
Access								R/W
Reset								0

Bit 31 – DMAGEN DMA Global Enable bit

Bit 28 – RBF_IEN0 RAM Buffer B Full Interrupt Enable bit

Value	Description
1	Interrupts are enabled and generated when the RBFx Status bit is set
0	Interrupts are disabled

Bit 23 – WR_OVF_ERR DMA Write Overflow Error bit
Note: This bit is cleared by hardware after a software read of the ADCDMASTAT register.

Value	Description
1	DMA write overflow error has occurred (circular buffer)
0	DMA write overflow error has not occurred

Bit 16 – RBF0 RAM Buffer B Full Status bit
Note: These bits are self-clearing upon being a read by software.

Value	Description
1	RAM Buffer B is full
0	RAM Buffer B is not full

Bit 15 – DMA_CNT_EN DMA Write Overflow Error bit
Note: This bit is cleared by hardware after a software read of the ADCDMASTAT register.

Value	Description
1	DMA write overflow error has occurred (circular buffer)
0	DMA write overflow error has not occurred

© 2025 Microchip Technology Inc. and its subsidiaries

Bit 8 – RAF_IEN0 RAM Buffer A Full Interrupt Enable bit

Value	Description
1	Interrupts are enabled and generated when the RAFx status bit is set
0	Interrupts are disabled

Bit 0 – RAF0 RAM Buffer A Full Status bit

Note: These bits are self-clearing upon being a read by software.

Value	Description
1	RAM Buffer A is full
0	RAM Buffer A is not full

© 2025 Microchip Technology Inc. and its subsidiaries

36.13.22. ADCCNTB – ADC Sample Count Base Address Register

Name: ADCCNTB
Offset: 0x320
Reset: 0x00000000
Property: RW

Bit	31	30	29	28	27	26	25	24
	ADCCNTB[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
	ADCCNTB[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
	ADCCNTB[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
	ADCCNTB[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 – ADCCNTB[31:0] Analog-to-Digital Count Base Address bits

The ADCCNTB register contains the user-defined RAM address at which the DMA engine will start saving the current count of output samples (if the DMACNTEN bit (ADCDMASTAT[15]) is set), which is already written to each of the buffers in the System RAM for each ADC module. The ADCx module will have its Buffer A current sample count saved at the address ((ADCCNTB) + 2 * x) and its Buffer B current sample count saved at the address ((ADCCNTB) + (2 * x + 1)). Where 'x' is the dedicated ADC module ID.

© 2025 Microchip Technology Inc. and its subsidiaries

36.13.23. ADCDMAB – ADC DMA Base Address Register

Name: ADCDMAB
Offset: 0x330
Reset: 0x00000000
Property: RW

Bit	31	30	29	28	27	26	25	24
	ADDMAB[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
	ADDMAB[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
	ADDMAB[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
	ADDMAB[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 – ADDMAB[31:0] DMA Base Address bits

The ADCDMAB register contains the user-specified RAM address at which DMA engine will start saving the converted data. The address of saving each data is specified by the following relations:
Buffer A starting address at: ADCDMAB + (2 * x) * 2(ADCON1bits.DMABL + 1)
Buffer B starting at: ADCDMAB + (2 * (x + 1)) * 2(ADCON1bits.DMABL + 1)
Where, 'x' is the dedicated ADC module ID.

© 2025 Microchip Technology Inc. and its subsidiaries

36.13.24. ADCTRGSNS – ADC Trigger Level/Edge Sensitivity Register

Name: ADCTRGSNS
Offset: 0x340
Reset: 0x00000000
Property: -

This register contains the setting for trigger level for each ADC analog input.

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
	LVL7	LVL6	LVL5	LVL4	LVL3	LVL2	LVL1	LVL0
Access	R/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 0, 1, 2, 3, 4, 5, 6, 7 – LVL Trigger Level and Edge Sensitivity bits

Notes:

1. This register specifies the trigger level for analog inputs 0 to 7.
2. The higher analog input ID belongs to Class 3, and, therefore, is only scan triggered. All Class 3 analog inputs use the scan trigger, for which the level/edge is defined by the STRGLVL bit (ADCCON1[3]).

Value	Description
1	Analog input is sensitive to the high level of its trigger (level sensitivity implies retriggering as long as the trigger signal remains high)
0	Analog input is sensitive to the positive edge of its trigger (this is the value after a reset)

© 2025 Microchip Technology Inc. and its subsidiaries

36.13.25. ADC Timing for First Class Channel Register (x = 0)

Name: ADC0TIME
Offset: 0x350
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
				EISx[2:0]			SELRESx[1:0]	
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	BCHENx	ADCDIVx[6:0]						
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

Bit	7	6	5	4	3	2	1	0
	SAMCx[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 28:26 – EISx[2:0] Early Interrupt Select Bits for First Class Channel index x (x=0...6)
These bits select the number of clocks prior to the end of conversion that the early interrupt is generated.
Notes: Depending on the bit resolution selection field SELRESx[1:0], the allowed EISx[2:0] values are:

- For SELRESx[1:0] == `2'b11`, i.e. 12-bit Resolution, all 8 possible settings for Early interrupt are allowed, 0 to 7 (1 to 8 Core Clock periods early)
- For SELRESx[1:0] == `2'b10`, i.e. 10-bit Resolution, all 8 possible settings for Early interrupt are allowed, 0 to 7 (1 to 8 Core Clock periods early)
- For SELRESx[1:0] == `2'b01`, i.e. 8-bit Resolution, only the 6 lowest settings are allowed, 0 to 5 (1 to 6 Core Clock periods early)
- For SELRESx[1:0] == `2'b00`, i.e. 6-bit Resolution, only the 4 lowest settings are allowed, 0 to 3 (1 to 4 Core Clock periods early)

The hardware will utilize the maximum allowed Early Interrupt setting if the user programs the EISx bit-field to a nonallowed value.

Value	Description
111	The early ready interrupt is generated 8 ADC core clocks prior to end of conversion
110	The early ready interrupt is generated 7 ADC core clocks prior to end of conversion
101	The early ready interrupt is generated 6 ADC core clocks prior to end of conversion
100	The early ready interrupt is generated 5 ADC core clocks prior to end of conversion
011	The early ready interrupt is generated 4 ADC core clocks prior to end of conversion
010	The early ready interrupt is generated 3 ADC core clocks prior to end of conversion
001	The early ready interrupt is generated 2 ADC core clocks prior to end of conversion
000	The early ready interrupt is generated 1 ADC core clock prior to end of conversion

© 2025 Microchip Technology Inc. and its subsidiaries

Bits 25:24 – SELRESx[1:0] ADCx Resolution Select bits
Selects ADC Resolution for First Class Channel index x (x=0...6) as follows:
00 -> 6 bits / 01 -> 8 bits / 10 -> 10 bits / 11 -> 12 bits

Value	Description
00	Sets the resolution to 6 bits for the ADC
01	Sets the resolution to 8 bits for the ADC
10	Sets the resolution to 10 bits for the ADC
11	Sets the resolution to 12 bits for the ADC

Bit 23 – BCHENx Buffer Channel Enable bit
If set to 1 and if ADDMAST.DMAEN == 1, the output data of first class channel x, x=0...6, will be saved by the DMA interface to the System RAM. If set to 0, this first class channel output data can be retrieved only via the UPB interface.

Bits 22:16 – ADCDIVx[6:0] ADCx Clock Divisor bits
Division Ratio for the SAR ADC core clock core_clk[x] of the First Class Channel index x (x=0...6) from the ADC control clock ctl_clk.
The ADCDIVx bit field divides the ADC control clock ctl_clk with period TQ:

Value	Description
1111111	2TQ·(ADCDIVx[6:0]) = 254·TQ = TADx
.....	
0000011	2TQ·(ADCDIVx[6:0]) = 6·TQ = TADx
0000010	2TQ·(ADCDIVx[6:0]) = 4·TQ = TADx
0000001	2TQ·(ADCDIVx[6:0]) = 2·TQ = TADx
0000000	Reserved (but still implemented as 2TQ·(ADCDIVx[6:0]) = 2·TQ = TADx)

Bits 9:0 – SAMCx[9:0] ADCx Sample Time bits
Burst Mode Sampling Time for First Class Channel index x (x=0...6) applies to all samples of a burst of samples converted by the first channel index x (when the channel is set with trigger level sensitivity), with the following exception regarding the very first sample in the burst. If the bit ADTFCC.SYSMPLENx is set to zero (y=0...1, x=0...6), then the first sample of a burst is sampled in asynchronous mode, which means that the trigger event for the first sample signifies the end-of-sampling. If the bit ADTFCC.SYSMPLENx is set to one, then the first sample is made in synchronous mode, which means that the locallysynchronized (control clock edge-aligned) trigger event signifies the extension-of-sampling, which has a duration dictated by this bit-field SAMCx in the same way the subsequent samples of the burst have.
This bit field uses the encoding as follows:
SAMCx[9:0] is specified in TADX = period of the core_clk[x], which is the ADC core clock for the first class ADC core index x, controlled by ADCDIVx[6:0] defined above in this table.

Value	Description
1111111111	1025 TADX
.....	
0000000001	3 TADX
0000000000	2 TADX

© 2025 Microchip Technology Inc. and its subsidiaries

36.13.26. ADCANCON – ADC Analog Warm-up Control Register

Name: ADCANCON
Offset: 0x400
Reset: 0x00000000
Property: -

This register contains the warm-up control settings for the analog and bias circuit of the ADC module.

Bit	31	30	29	28	27	26	25	24
					WKUPCLKCNT[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WKIEN7							WKIEN0
Access	R/W							R/W
Reset	0							0

Bit	15	14	13	12	11	10	9	8
	WKRDY7							WKRDY0
Access	R/HS/HC							R/HS/HC
Reset	0							0

Bit	7	6	5	4	3	2	1	0
	ANEN7							ANEN0
Access	R/W							R/W
Reset	0							0

Bits 27:24 – WKUPCLKCNT[3:0] Wake-up Clock Count bits
These bits represent the number of ADC clocks required to warm-up the ADC module before it can perform conversion. Although the clocks are specific to each ADC, the WKUPCLKCNT bit is common to all ADC modules.

Value	Description
1111	2^{15}= 32,768 clocks
...	—
...	—
...	—
0110	2^{6}= 64 clocks
0101	2^{5}= 32 clocks
0100	2^{4}= 16 clocks
0011	2^{4}= 16 clocks
0010	2^{4}= 16 clocks
0001	2^{4}= 16 clocks
0000	2^{4}= 16 clocks

Bit 23 – WKIEN7 Shared ADC Wake-up Interrupt Enable bit

Value	Description
1	Enable interrupt and generate interrupt when the WKRDY7 status bit is set
0	Disable interrupt

Bit 16 – WKIEN0 Shared ADC Wake-up Interrupt Enable bit

© 2025 Microchip Technology Inc. and its subsidiaries

Value	Description
1	Enable interrupt and generate interrupt when the WKRDY7 status bit is set
0	Disable interrupt

Bit 15 – WKRDY7 Shared ADC Wake-up Status bit

Note: This bit is cleared by hardware when the ANEN7 bit is cleared.

Value	Description
1	ADC Analog and bias circuitry ready after the wake-up count number $2^{WKUPEXP}$ clocks after setting ANEN2 to '1'
0	ADC Analog and bias circuitry is not ready

Bit 8 – WKRDY0 Shared ADC Wake-up Status bit

Note: This bit is cleared by hardware when the ANEN7 bit is cleared.

Value	Description
1	ADC Analog and bias circuitry ready after the wake-up count number $2^{WKUPEXP}$ clocks after setting ANEN2 to '1'
0	ADC Analog and bias circuitry is not ready

Bit 7 – ANEN7 Shared ADC Analog and Bias Circuitry Enable bit

Value	Description
1	Analog and bias circuitry enabled. When the analog and bias circuit is enabled, the ADC module needs a warm-up time, as defined by the ADCANCON.WKUPCLKCNT[3:0] bits.
0	Analog and bias circuitry disabled

Bit 0 – ANEN0 Shared ADC Analog and Bias Circuitry Enable bit

Value	Description
1	Analog and bias circuitry enabled. When the analog and bias circuit is enabled, the ADC module needs a warm-up time, as defined by the ADCANCON.WKUPCLKCNT[3:0] bits.
0	Analog and bias circuitry disabled

© 2025 Microchip Technology Inc. and its subsidiaries

36.13.27. ADC7CFG – ADC7 Configuration Register

Name: ADC7CFG
Offset: 0x670
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	ADCCFG[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
	ADCCFG[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
	ADCCFG[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
	ADCCFG[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 – ADCCFG[31:0] ADC Module Configuration Data bits

© 2025 Microchip Technology Inc. and its subsidiaries

36.13.28. ADCSYSCFG0 – ADC System Configuration Register 0

Name: ADCSYSCFG0
Offset: 0x700
Reset: 0x00000000
Property: -

This register contains read-only bits corresponding to the analog input.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

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

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

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

Bits 22:0 – AN[22:0] ADC Analog Input bits
These bits reflect the system configuration and are updated during boot-up time. By reading these read-only bits, the user application can determine whether or not an analog input in the device is available.

© 2025 Microchip Technology Inc. and its subsidiaries

36.13.29. ADCDATAn – ADC Output Data Register ('n' = 0 to 22)

Name: ADCDATAn
Offset: 0x0A00 + n*0x10 [n=0..22]
Reset: 0x00000000
Property: -

These registers are the analog-to-digital conversion output data registers. The ADCDATAn register is associated with each external analog input, 0-18, plus internal analog inputs 19-22.

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

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

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

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

Bits 31:0 – DATA[31:0] ADC Converted Data Output bits
Note: Reading the ADCDATAn register value after changing the ADCCON1.FRACT bit converts the data into the format specified by the ADCCON1.FRACT bit.

© 2025 Microchip Technology Inc. and its subsidiaries

37. Automated Touch Sensing using the Enhance Capacitive Voltage Divider (CVD)

37.1. Overview

The enhanced CVD peripheral adds automated touch sensing capability to the PIC32CX-BZ6 device family. The touch peripheral available on these family of devices supports self- and mutual-capacitance measurement. The enhanced CVD controller offloads the CPU by automated touch scans with programmable phase timing and oversampling.

37.2. Features

- Self Measurement and Mutual Measurement for Touch Detection
  Note: The CVD Mutual only mode is not supported.
- Supports Up to 18 RX/TX Channels for Touch Measurements:
  - Supports up to 18 channels in self-measurement
  - Supports Mutual cap measurement
- AddCap Control to Optimize Sensitivity and Noise Performance especially for Larger Capacitive Loads
- Low Power supports for Lumping of Multiple Touch Channels
- Automated Oversampling to Increase the Signal-to-Noise Ratio
- Driven Shield+ for Superior Noise Immunity and Moisture Tolerance:
  - Any touch channel can be used for the driven shield
  - Driven Shield+ enables all touch sensor to work as driven shield
- Programmable Threshold to Detect Touch in Low-Power Measurements (Low-power Wake on Touch)
- Supported in Standby Sleep and Idle Mode

37.3. Configuration

Touch library support is available to configure the touch sensors like Button/Slider/Wheel/Surface. The touch library is configured using Microchip Code Configurator (MCC). For more details, refer to https://microchipdeveloper.com/touch:generate-touch-project-with-harmony.

The complete and recent list of touch features supported is listed in the "Getting Started with Microchip Touch".

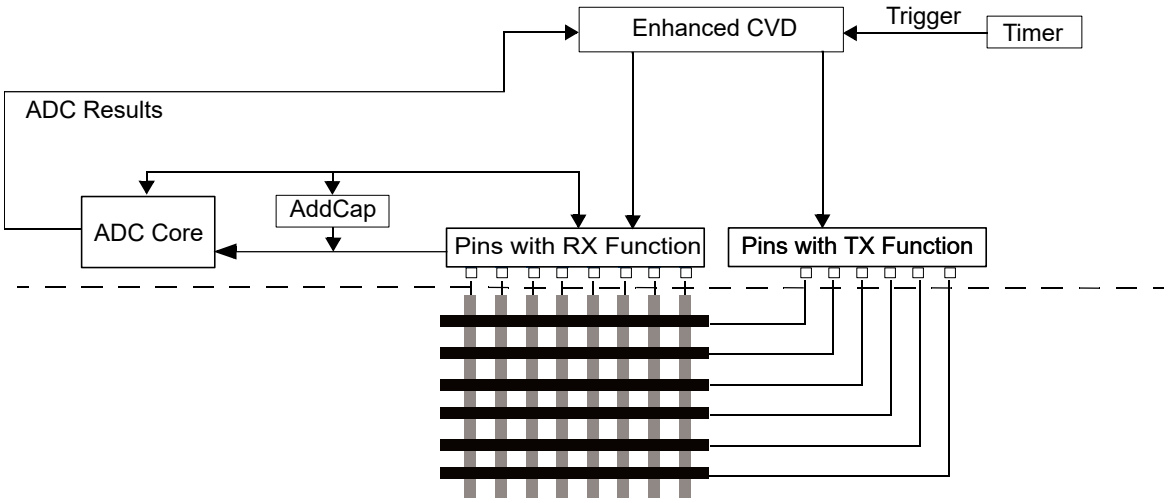
For more details on the example project, refer to https://discover.microchip.com/.

For more details on designing the touch sensor electrode, refer to *Capacitive Touch Sensor Design Guide*.

© 2025 Microchip Technology Inc. and its subsidiaries

37.4. Block Diagram

Figure 37-1. Enhanced Capacitive Voltage Divider (CVD) Controller Block Diagram



37.5. CVD Module Operation

The enhanced CVD uses the shared ADC SARCORE for its operations; see *Analog-to-Digital Converter (ADC)* from Related Links. The enhanced CVD controller controls the shared ADC SARCORE in a simplified mode that supports only the needs of CVD, performing the full sequence of an oversampled CVD touch scan.

The controller also controls pin functions needed for the CVD operations. Some of these pins provide RX functionality by having an analog. Some pins may have both RX and TX functions. The RX and TX pins connect to a matrix of button electrodes, a touch screen or touch pad electrode grid with horizontal and vertical structures. To determine a touch or an approach, measure the capacitance of these electrodes.

An AddCap module connects to the analog input net to augment the internal capacitance of the chip to match the base capacitance of the touch sensor. Measurements are triggered either by a CPU or a timer through EVSYS.

Related Links

Analog-to-Digital Converter (ADC)

37.5.1. Theory of Operation

This section describes the full sequence of an oversampled CVD touch scan for self and mutual mode.

37.5.1.1. CVD Self Sensing

A CVD operation begins with the internal Sample-and-Hold capacitor (S&H Cap) being disconnected from the path that connects it to the external capacitive sensor node. While disconnected, S&H Cap is precharged to V_{DD} or discharged to V_{SS}. The sensor node is either discharged or charged to V_{SS} or V_{DD}, respectively, to the opposite level of S&H Cap. When the precharge phase is complete, the V_{DD}/V_{SS} bias paths for the two nodes are disconnected and the paths between S&H Cap and the external sensor node is reconnected, at which time the acquisition phase of the CVD operation begins. During acquisition, a capacitive voltage divider is formed between the precharged S&H Cap and sensor nodes, which results in a final voltage level setting on S&H Cap, which is determined by the capacitances and precharge levels of the two nodes. After acquisition, the ADC converts the voltage level on S&H Cap. This process is, then, repeated with the selected precharge levels inverted

© 2025 Microchip Technology Inc. and its subsidiaries

for both the S&H Cap and the sensor nodes. The waveform for two CVD measurements, which is known as the differential CVD measurement, is illustrated in the following figure.

37.5.1.2. CVD Mutual Sensing

The mutual capacitance measurement is performed between two lines, namely transmit (TX) and receive (RX). The mutual TX line is driven in sync with the charge-sharing phase of self-cap measurement resulting in charge contribution from the TX pin. The resulting charge distribution between S&H Cap and the RX line is a combination of charge accumulated on S&H Cap and charge contributed by driving the TX line.

Figure 37-2. CVD Mutual Sensing Waveform

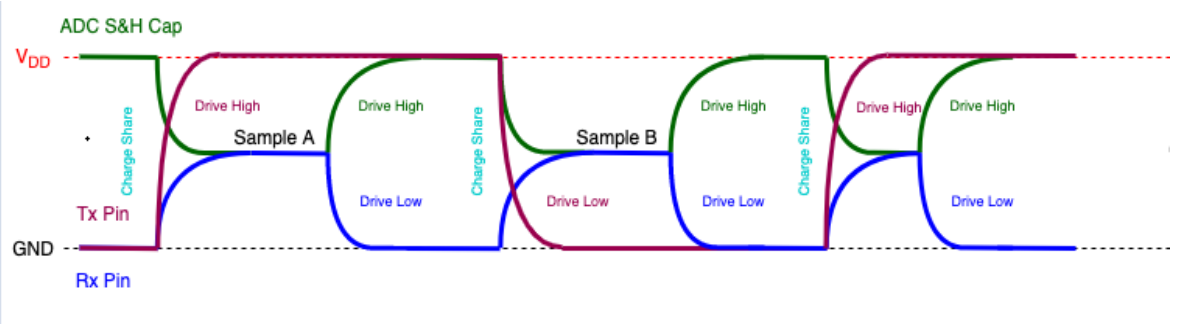
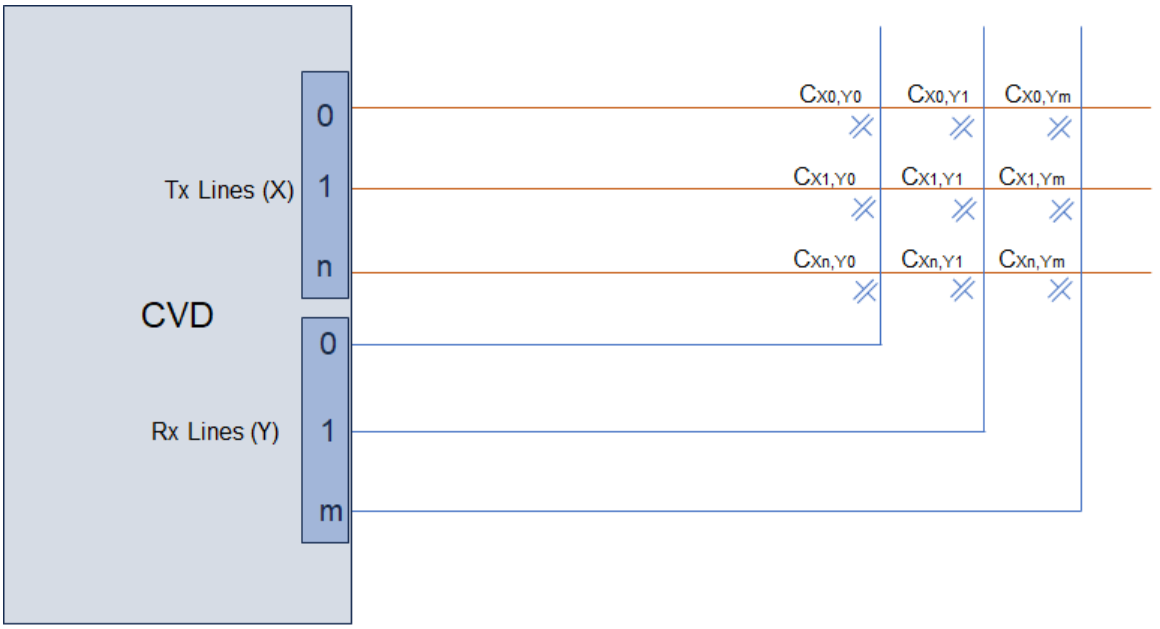


Figure 37-3. Typical Mutual Cap Sensor Arrangements



37.5.2. Channel Sets

The enhanced CVD module can control up to eight RX inputs and eight TX inputs. It enables the user to map up to eight specific CVDRx pins sequential RX indexes and up to eight specific CVDTx pins to sequential TX indexes via Special Function Registers (SFRs). The actual pins used can, therefore, be in any order and with any gaps required. This enables the PCB designer to map I/O pins to panel RX/TX functions with greater ease.

© 2025 Microchip Technology Inc. and its subsidiaries

37.5.2.1. Lumping/Channel Grouping/STRIDE

Each scan can perform touch measurement on multiple RX/TX pins via the STRIDE (also known as Lump and Channel Grouping) setting, enabling the user to scan multiple RX inputs in parallel and/or drive multiple TX outputs in parallel.

37.5.2.2. Scan Descriptors

The enhanced CVD module includes four scan descriptors to off-load the intervention requirements and timing dependencies of the CPU. Each scan descriptor includes the RX and TX indexes to be scanned and the strides for each, as well as the time for each measurement phase, the amount of oversampling and a threshold at which to act on the data.

Each descriptor indicates:

- RX indexes to be scanned
- TX indexes to be driven
- Number of RX indexes to scan at one time
- Number of TX indexes to scan at one time
- Independent enable for Self (RX-drive) and combination of Self and Mutual modes
- Channel timing control
- Oversampling or threshold support
- Provision to enable an interrupt when complete or when threshold exceeds
- The Enable mode settings to enable the single, continuous or continuous until threshold scanning

37.5.3. Oversampling

A scan descriptor can program each measurement for any amount of oversampling from 0-127 additional samples. This accumulates all the samples into a 19-bit result. Do not perform division or averaging to prevent data loss and to enable non-power-of-two oversampling.

37.5.4. Phase Timing

The user can tune the timing of the main six phases of the CVD measurement as per application needs.

The scan descriptor holds time values for each of the following phases of the measurement cycle:

- Charge
- Acquire
- Conversion
- Spacing between polarity measurements
- Spacing between oversampling of same point(s)
- Spacing between channels

37.5.5. Thresholds

Each scan descriptor specifies a threshold value. If the application wishes to receive all the accumulated oversample data, it can set the threshold to zero. Otherwise, the scan descriptor can be set to only store data that exceed the requested threshold or to store all data and assert an interrupt only when the threshold is met.

37.5.6. Interrupts

Configure the enhanced CVD to cause an interrupt when the FIFO exceeds a programmed threshold and/or when a scan descriptor exceeds a threshold and/or completes its function.
Note: FIFO threshold interrupt is not supported in the Standby Sleep mode.

© 2025 Microchip Technology Inc. and its subsidiaries

37.5.7. Triggers

The enhanced CVD can be set to start the enabled Scan Descriptors upon two possible trigger events. One is a software trigger SFR bit and another trigger is from any of EVSYS generators. See *Event System (EVSYS)* from Related Links.

If a scan descriptor is set to run in the continuous mode, it runs when it receives a trigger and stays enabled. The CVD module then waits for the next trigger to run the next enabled scan descriptor. Alternatively, the scan descriptor may instruct the CVD module to continue scanning the one descriptor on every trigger until a threshold is met. These modes allow the CVD module to run autonomously, being triggered by a system timer while the CPU sleeps (Idle mode or Standby Sleep mode), and interrupt the CPU to wake it if a tough threshold is reached. The first mode is useful for scanning various portions of the panel at regular intervals. The second mode is useful for scanning a low-resolution scan on every trigger, then, moving to a higher resolution scan descriptor if a threshold is met.

Related Links

Event System (EVSYS)

37.5.8. FIFO

Results are stored in a FIFO with a depth of 8 word. The FIFO contains the accumulated result data and the TX and RX indexes for that data. The user can configure the CVD module to store all accumulated results into the FIFO or only results that exceed the threshold specified in the scan descriptor. In either case, the user can generate an interrupt when the FIFO exceeds a watermark value.

The application has the choice of reading out:

- Only a single 32-bit word that contains the RX and TX indexes and the delta of the positive and negative accumulated CVD measurements
- A total of three 32-bit words to read the actual positive and negative CVD measurement along with delta

The FIFO uses following registers:

- CVDRESH – CVD Results POS FIFO Read Register
- CVDRESL – CVD Results NEG FIFO Read Register
- CVDRESD – CVD Results Descriptor FIFO Read Register

© 2025 Microchip Technology Inc. and its subsidiaries

37.6. Register Summary

See the *CVD* module in the *Product Memory Mapping Overview* from Related Links for the base address.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CVDCON	7:0			CLKSEL[1:0]		TRIGSEL[3:0]			
		15:8	FIFOTH[7:0]							
		23:16	THSTR				CVDIEN	FIFOIEN	FIFOTH[9:8]	
		31:24	ON	FRZ	SIDL	ORDER	SDWREN		ABORT	SWTRIG
0x04	CVDADC	7:0					ADCCON	ADCCON[2:0]		
		15:8					ADCCON	ADCCON	ADCCON[1:0]	
		23:16								
		31:24								
0x08	CVDSTAT	7:0		SD2INT	SD2DONE	SD2BUSY		SD1INT	SD1DONE	SD1BUSY
		15:8		SD4INT	SD4DONE	SD4BUSY		SD3INT	SD3DONE	SD3BUSY
		23:16	FIFOCNT[7:0]							
		31:24	FIFOFULL	FIFOWM	FIFOMT				FIFOCNT[9:8]	
0x0C ... 0x0F	Reserved									
0x10	CVDRESH	7:0	POS[7:0]							
		15:8	POS[15:8]							
		23:16	POS[23:16]							
		31:24								
0x14	CVDRESL	7:0	NEG[7:0]							
		15:8	NEG[15:8]							
		23:16	NEG[23:16]							
		31:24								
0x18	CVDRESD	7:0	DELTA[7:0]							
		15:8	DELTA[15:8]							
		23:16	RXINDEX[4:0]						DELTA[17:16]	
		31:24	TXINDEX[4:0]						SDNUM[1:0]	
0x1C ... 0x7F	Reserved									
0x80	CVDRX0	7:0	RXAN1[5:0]							
		15:8	RXAN2[5:0]							
		23:16	RXAN3[5:0]							
		31:24	RXAN4[5:0]							
0x84	CVDRX1	7:0	RXAN5[5:0]							
		15:8	RXAN6[5:0]							
		23:16	RXAN7[5:0]							
		31:24	RXAN8[5:0]							
0x88	CVDRX2	7:0	RXAN9[5:0]							
		15:8	RXAN10[5:0]							
		23:16	RXAN11[5:0]							
		31:24	RXAN12[5:0]							
0x8C	CVDRX3	7:0	RXAN13[5:0]							
		15:8	RXAN14[5:0]							
		23:16	RXAN15[5:0]							
		31:24	RXAN16[5:0]							
0x90	CVDRX4	7:0	RXAN17[5:0]							
		15:8	RXAN18[5:0]							
		23:16								
		31:24								
0x94 ... 0xBF	Reserved									
0xC0	CVDTX0	7:0	TXOUT0[5:0]							
		15:8	TXOUT1[5:0]							
		23:16	TXOUT2[5:0]							
		31:24	TXOUT3[5:0]							

© 2025 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xC4	CVDTX1	7:0			TXOUT4[5:0]					
		15:8			TXOUT5[5:0]					
		23:16			TXOUT6[5:0]					
		31:24			TXOUT7[5:0]					
0xC8	CVDTX2	7:0			TXOUT8[5:0]					
		15:8			TXOUT9[5:0]					
		23:16			TXOUT10[5:0]					
		31:24			TXOUT11[5:0]					
0xCC	CVDTX3	7:0			TXOUT12[5:0]					
		15:8			TXOUT13[5:0]					
		23:16			TXOUT14[5:0]					
		31:24			TXOUT15[5:0]					
0xD0	CVDTX4	7:0			TXOUT16[5:0]					
		15:8			TXOUT17[5:0]					
		23:16								
		31:24								
0xD4 ... 0xFF	Reserved									
0x0100	CVDSD0C1	7:0		SD0OVRSAMP[6:0]						
		15:8	SD0TH[7:0]							
		23:16	SD0TH[15:8]							
		31:24	SD0TH[23:16]							
0x0104	CVDSD0C2	7:0	SD0RXSTRIDE	SD0RXSTRIDE	SD0RXBEG[5:0]					
		15:8	SD0RXSTRIDE	SD0RXSTRIDE	SD0RXEND[5:0]					
		23:16	SD0TXSTRIDE	SD0TXSTRIDE	SD0TXBEG[5:0]					
		31:24	SD0TXSTRIDE	SD0TXSTRIDE	SD0TXEND[5:0]					
0x0108	CVDSD0C3	7:0		SD0CHGTIME[6:0]						
		15:8		SD0ACQTIME[6:0]						
		23:16	SDnADCCON[7:0]							
		31:24	SD0EN[1:0]				SD0BUF	SD0INTEN	SD0SELF	SD0MUT
0x010C	CVDSD0T2	7:0		SD0CONTIME[6:0]						
		15:8		SD0POLTIME[6:0]						
		23:16		SD0OVRTIME[6:0]						
		31:24		SD0CHNTIME[6:0]						
0x0110	CVDSD1C1	7:0		SD1OVRSAMP[6:0]						
		15:8	SD1TH[7:0]							
		23:16	SD1TH[15:8]							
		31:24	SD1TH[23:16]							
0x0114	CVDSD1C2	7:0	SD1RXSTRIDE	SD1RXSTRIDE	SD1RXBEG[5:0]					
		15:8	SD1RXSTRIDE	SD1RXSTRIDE	SD1RXEND[5:0]					
		23:16	SD1TXSTRIDE	SD1TXSTRIDE	SD1TXBEG[5:0]					
		31:24	SD1TXSTRIDE	SD1TXSTRIDE	SD1TXEND[5:0]					
0x0118	CVDSD1C3	7:0		SD1CHGTIME[6:0]						
		15:8		SD1ACQTIME[6:0]						
		23:16					CVDEN	CVDCPL[2:0]		
		31:24	SD1EN[1:0]				SD1BUF	SD1INTEN	SD1SELF	SD1MUT
0x011C	CVDSD1T2	7:0		SD1CONTIME[6:0]						
		15:8		SD1POLTIME[6:0]						
		23:16		SD1OVRTIME[6:0]						
		31:24		SD1CHNTIME[6:0]						
0x0120	CVDSD2C1	7:0		SD2OVRSAMP[6:0]						
		15:8	SD2TH[7:0]							
		23:16	SD2TH[15:8]							
		31:24	SD2TH[23:16]							
0x0124	CVDSD2C2	7:0	SD2RXSTRIDE	SD2RXSTRIDE	SD2RXBEG[5:0]					
		15:8	SD2RXSTRIDE	SD2RXSTRIDE	SD2RXEND[5:0]					
		23:16	SD2TXSTRIDE	SD2TXSTRIDE	SD2TXBEG[5:0]					
		31:24	SD2TXSTRIDE	SD2TXSTRIDE	SD2TXEND[5:0]					

© 2025 Microchip Technology Inc. and its subsidiaries

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0128	CVDSD2C3	7:0		SD2CHGTIME[6:0]						
		15:8		SD2ACQTIME[6:0]						
		23:16					CVDEN	CVDCPL[2:0]		
		31:24	SD2EN[1:0]				SD2BUF	SD2INTEN	SD2SELF	SD2MUT
0x012C	CVDSD2T2	7:0		SD2CONTIME[6:0]						
		15:8		SD2POLTIME[6:0]						
		23:16		SD2OVRTIME[6:0]						
		31:24		SD2CHNTIME[6:0]						
0x0130	CVDSD3C1	7:0		SD3OVRSAMP[6:0]						
		15:8	SD3TH[7:0]							
		23:16	SD3TH[15:8]							
		31:24	SD3TH[23:16]							
0x0134	CVDSD3C2	7:0	SD3RXSTRIDE	SD3RXSTRIDE	SD3RXBEG[5:0]					
		15:8	SD3RXSTRIDE	SD3RXSTRIDE	SD3RXEND[5:0]					
		23:16	SD3TXSTRIDE	SD3TXSTRIDE	SD3TXBEG[5:0]					
		31:24	SD3TXSTRIDE	SD3TXSTRIDE	SD3TXEND[5:0]					
0x0138	CVDSD3C3	7:0		SD3CHGTIME[6:0]						
		15:8		SD3ACQTIME[6:0]						
		23:16					CVDEN	CVDCPL[2:0]		
		31:24	SD3EN[1:0]				SD3BUF	SD3INTEN	SD3SELF	SD3MUT
0x013C	CVDSD3T2	7:0		SD3CONTIME[6:0]						
		15:8		SD3POLTIME[6:0]						
		23:16		SD3OVRTIME[6:0]						
		31:24		SD3CHNTIME[6:0]						

Related Links

Product Memory Mapping Overview

37.7. Register Description

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

Optional write protection by the PAC is denoted by the PAC Write Protection property in each individual register description. See *Peripheral Access Controller (PAC)* from Related Links.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable protection is denoted by the Enable-protected property in each individual register description.

The following are the list of conventions available in the register description:

- R = Readable bit
- W = Writable bit
- U = Unimplemented bit, read as '0'
- -n = Value at POR
- 1 = Bit is set
- 0 = Bit is cleared
- x = Bit is unknown
- HS = Hardware Set
- HC = Hardware Cleared

Related Links

Peripheral Access Controller (PAC)

© 2025 Microchip Technology Inc. and its subsidiaries

37.7.1. CVDCON - CVD Control Register

Name: CVDCON
Offset: 0x00
Reset: 0x00020000
Property: -

Bit	31	30	29	28	27	26	25	24
	ON	FRZ	SIDL	ORDER	SDWREN		ABORT	SWTRIG
Access	R/W	R/W	R/W	R/W	R/W		W/HC	W/HC
Reset	0	0	0	0	0		0	0

Bit	23	22	21	20	19	18	17	16
	THSTR				CVDIEN	FIFOIEN	FIFOTH[9:8]	
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	1	0

Bit	15	14	13	12	11	10	9	8
	FIFOTH[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

Bit 31 – ON Enables the State Machine to scan the enabled Scan Descriptors upon next trigger
Before turning ON bit from 1'b1 to 1'b0, the Scan Enable bits of all descriptors must be cleared and the CVD controller must either be allowed to finish any scan in progress or must be instructed to abort the scan with the ABORT bit.

Bit 30 – FRZ Freeze Mode

Value	Description
1	CVD controller stops in the Debugger mode
0	CVD controller runs in the Debugger mode

Bit 29 – SIDL Stop in Idle Mode bit

Value	Description
1	CVD controller stops when device enters the Idle mode
0	CVD controller continues running in the Idle mode

Bit 28 – ORDER RX/TX Loop Order

Value	Description
1	Scans all the requested TX indexes, then increments RX index and continues operation
0	Scans all the requested RX indexes, then increments TX index and continues operation

Bit 27 – SDWREN Scan Descriptor Write Enable

Value	Description
1	Enables writes to the scan descriptors
0	Prevents writes to the scan descriptors

© 2025 Microchip Technology Inc. and its subsidiaries

Bit 25 – ABORT Abort Current Scan

Note: The controller moves on to the next enabled Scan Descriptor if there is one; otherwise, it goes to idle state. Hardware clears this bit.

Value	Description
1	Aborts the current scan
0	CVD controller continues with the current scan

Bit 24 – SWTRIG Software Trigger control. Starts scan manually

Note: Hardware clears this bit.

Value	Description
1	Starts scan manually
0	Continues without the scan

Bit 23 – THSTR Threshold Store Mode

Value	Description
1	Stores only the results which exceed the programmed threshold for the Scan Descriptor
0	Stores all the results in FIFO

Bit 19 – CVDIEN Global Interrupt Enable

Value	Description
1	Enables the FIFO and scan descriptor interrupts
0	Disables the FIFO and scan descriptor interrupts

Bit 18 – FIFOIEN FIFO Threshold Interrupt Enable

Value	Description
1	Controller asserts an interrupt when the FIFO threshold is met
0	Controller does not assert an interrupt when the FIFO threshold is met

Bits 17:8 – FIFOTH[9:0] Threshold for the results FIFO

These bits contain threshold for the results FIFO that causes an interrupt and watermark FIFOWM status bit assertion.

Bits 5:4 – CLKSEL[1:0] Clock Select for CVD

Value	Description
00	PB1_CLK
01	POSC
10	LPRC
11	GCLK

Bits 3:0 – TRIGSEL[3:0] Trigger select for starting the scan

Value	Description
0000	SFR controlled software trigger
0001	EVSYS event. See *Event System (EVSYS)* from Related Links.
0010 – 1111	Reserved

Related Links

Event System (EVSYS)

© 2025 Microchip Technology Inc. and its subsidiaries

37.7.2. CVD ADC Configuration Register

Name: CVDADC
Offset: 0x04
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					ADCCON	ADCCON	ADCCON[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
					ADCCON	ADCCON[2:0]		
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 11 – ADCCON ANN1 Channel Enabled

Bit 10 – ADCCON Shared SARCORE Differential Enabled

Bits 9:8 – ADCCON[1:0] Shared SARCORE Resolution Select

Bit 3 – ADCCON CVD Enabled

Bits 2:0 – ADCCON[2:0] ADCON2.CVD_CPL[2:0] Register Field

© 2025 Microchip Technology Inc. and its subsidiaries

37.7.3. CVD Status Register

Name: CVDSTAT
Offset: 0x08
Reset: 0x20000000
Property: -

Bit	31	30	29	28	27	26	25	24
	FIFOFULL	FIFOWM	FIFOMT				FIFOCNT[9:8]	
Access	R	R	R				R	R
Reset	0	0	1				0	0

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

Bit	15	14	13	12	11	10	9	8
		SD4INT	SD4DONE	SD4BUSY		SD3INT	SD3DONE	SD3BUSY
Access		R/W/HS	R	R		R/W/HS	R	R
Reset		0	0	0		0	0	0

Bit	7	6	5	4	3	2	1	0
		SD2INT	SD2DONE	SD2BUSY		SD1INT	SD1DONE	SD1BUSY
Access		R/W/HS	R	R		R/W/HS	R	R
Reset		0	0	0		0	0	0

Bit 31 – FIFOFULL Results FIFO is Full

Value	Description
1	FIFO is full
0	Not full

Bit 30 – FIFOWM

Value	Description
1	FIFO reached the programmed FIFOTHRESH threshold
0	FIFO did not reach the programmed FIFOTHRESH threshold

Bit 29 – FIFOMT

Value	Description
1	FIFO is empty
0	FIFO is not empty

Bits 25:16 – FIFOCNT[9:0]
These bits indicate the number of words in the Results FIFO.

Bit 14 – SD4INT

Value	Description
1	Scan Descriptor 4 caused an interrupt
0	Scan Descriptor 4 did not cause an interrupt

© 2025 Microchip Technology Inc. and its subsidiaries

Bit 13 – SD4DONE

Note: The hardware clears this bit upon receiving next trigger for Scan Descriptor 4.

Value	Description
1	Scan Descriptor 4 completed at least once
0	Scan Descriptor 4 did not complete

Bit 12 – SD4BUSY

Value	Description
1	Scan Descriptor 4 is in progress
0	Scan Descriptor 4 is not in progress

Bit 10 – SD3INT

Value	Description
1	Scan Descriptor 3 caused an interrupt
0	Scan Descriptor 3 did not cause an interrupt

Bit 9 – SD3DONE

The controller sets this bit if Scan Descriptor 3 completes at least once. Core clears this bit upon receiving next trigger for Scan Descriptor 3.

Bit 8 – SD3BUSY

Value	Description
1	Scan Descriptor 3 is in progress
0	Scan Descriptor 3 is not in progress

Bit 6 – SD2INT

Value	Description
1	Scan Descriptor 2 caused an interrupt
0	Scan Descriptor 2 did not cause an interrupt

Bit 5 – SD2DONE The controller sets this bit if Scan Descriptor 2 completes at least once. Core clears this bit upon receiving next trigger for Scan Descriptor 2.

Bit 4 – SD2BUSY

Value	Description
1	Scan Descriptor 2 caused an interrupt
0	Scan Descriptor 2 did not cause an interrupt

Bit 2 – SD1INT Scan Descriptor 1 caused an interrupt

Value	Description
1	Scan Descriptor 1 caused an interrupt
0	Scan Descriptor 1 did not cause an interrupt

Bit 1 – SD1DONE The controller sets this bit if Scan Descriptor 1 completed at least once. Core clears this bit upon receiving next trigger for Scan Descriptor 1.

Bit 0 – SD1BUSY Scan Descriptor 1 is in progress

Value	Description
1	Scan Descriptor 1 is in progress
0	Scan Descriptor 1 is not in progress

© 2025 Microchip Technology Inc. and its subsidiaries

37.7.4. CVD Results POS FIFO Read Register

Name: CVDRESH
Offset: 0x010
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

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

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

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

Bits 23:0 – POS[23:0]

The accumulated result of the positive-side measurements Since the controller supports up to 128x oversampling, each polarity can accumulate up to 23 bits when using a 16-bit ADC. The accumulation is not shifted back down to create an average. Therefore, if oversampling was requested, the software will need to account for the left-shift of both the result returned.

© 2025 Microchip Technology Inc. and its subsidiaries

37.7.5. CVD Results NEG FIFO Read Register

Name: CVDRESL
Offset: 0x014
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

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

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

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

Bits 23:0 – NEG[23:0]

The accumulated result of the negative-side measurements Since the controller supports up to 128x oversampling, each polarity can accumulate up to 23 bits when using a 16-bit ADC. The accumulation is not shifted back down to create an average. Therefore, if oversampling was requested, the SW will need to account for the left-shift of both the result returned.

© 2025 Microchip Technology Inc. and its subsidiaries

37.7.6. CVD Results Descriptor FIFO Read Register

Name: CVDRESD
Offset: 0x018
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	TXINDEX[4:0]						SDNUM[1:0]	
Access	R	R	R	R	R		R	R
Reset	0	0	0	0	0		0	0

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

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

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

Bits 31:27 – TXINDEX[4:0] Transmit Index of the result
If the Stride of the Scan Descriptor is more than one, the Transmit Index indicates the first one of the group.

Bits 25:24 – SDNUM[1:0] Scan Descriptor Number
These bits has the Scan Descriptor Number that generated the result.

Bits 23:19 – RXINDEX[4:0] Receive Index of the result
If the Stride of the Scan Descriptor is more than one, the Receive Index indicates the first one of the group.

Bits 17:0 – DELTA[17:0] Delta of the accumulated results of the negative-side and positive-side measurements
The controller supports up to 128x oversampling; therefore, each polarity can accumulate up to 19 bits when using 12-bit ADC. The accumulation is not shifted back down to create an average. Therefore, if oversampling was requested, the software will need to account for the results returned.
The DELTA presented is the 18 MSBs (including sign bit) of a signed subtraction of the two accumulators. The data will be in 2's complement form if the delta is negative.
Width Needed: ADCBITS + 7 + 1 (sign)
Width available: 18
delta_pre[ADCBITS+7:0] = signed'({1'b0, POS[ADCBITS+7-1:0]} - {1'b0, NEG[ADCBITS+7-1:0]})
DELTA[17:0] = delta_pre[ADC_BITS+7:ADC_BITS+7-17]

© 2025 Microchip Technology Inc. and its subsidiaries

Note: Reading this register increments the FIFO read pointer, destroying the data in the previous two registers for NEG and POS absolute values. If the NEG and POS values are desired, those registers must be read before this register. If the absolute values are not required, bandwidth can be saved by reading only this descriptor register.

© 2025 Microchip Technology Inc. and its subsidiaries

37.7.7. CVD Receive Index 0 Configuration

Name: CVDRX0
Offset: 0x080
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
			RXAN4[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
			RXAN3[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
			RXAN2[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
			RXAN1[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 29:24 – RXAN4[5:0] ANx/CVDR channel to use for RX Index 4

Bits 21:16 – RXAN3[5:0] ANx/CVDR channel to use for RX Index 3

Bits 13:8 – RXAN2[5:0] ANx/CVDR channel to use for RX Index 2

Bits 5:0 – RXAN1[5:0] ANx/CVDR channel to use for RX Index 1

© 2025 Microchip Technology Inc. and its subsidiaries

37.7.8. CVD Receive Index 1 Configuration

Name: CVDRX1
Offset: 0x084
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
			RXAN8[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
			RXAN7[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
			RXAN6[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
			RXAN5[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 29:24 – RXAN8[5:0] ANx/CVDR channel to use for RX Index 8

Bits 21:16 – RXAN7[5:0] ANx/CVDR channel to use for RX Index 7

Bits 13:8 – RXAN6[5:0] ANx/CVDR channel to use for RX Index 6

Bits 5:0 – RXAN5[5:0] ANx/CVDR channel to use for RX Index 5

© 2025 Microchip Technology Inc. and its subsidiaries

37.7.9. CVD Receive Index 2 Configuration

Name: CVDRX2
Offset: 0x088
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
			RXAN12[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
			RXAN11[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
			RXAN10[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
			RXAN9[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 29:24 – RXAN12[5:0] ANx/CVDR channel to use for RX Index 12

Bits 21:16 – RXAN11[5:0] ANx/CVDR channel to use for RX Index 11

Bits 13:8 – RXAN10[5:0] ANx/CVDR channel to use for RX Index 10

Bits 5:0 – RXAN9[5:0] ANx/CVDR channel to use for RX Index 9

© 2025 Microchip Technology Inc. and its subsidiaries

37.7.10. CVD Receive Index 3 Configuration

Name: CVDRX3
Offset: 0x08C
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
			RXAN16[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
			RXAN15[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
			RXAN14[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
			RXAN13[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 29:24 – RXAN16[5:0] ANx/CVDR channel to use for RX Index 16

Bits 21:16 – RXAN15[5:0] ANx/CVDR channel to use for RX Index 15

Bits 13:8 – RXAN14[5:0] ANx/CVDR channel to use for RX Index 14

Bits 5:0 – RXAN13[5:0] ANx/CVDR channel to use for RX Index 13

© 2025 Microchip Technology Inc. and its subsidiaries

37.7.11. CVD Receive Index 4 Configuration

Name: CVDRX4
Offset: 0x090
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			RXAN18[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
			RXAN17[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 13:8 – RXAN18[5:0] ANx/CVDR channel to use for RX Index 18

Bits 5:0 – RXAN17[5:0] ANx/CVDR channel to use for RX Index 17

© 2025 Microchip Technology Inc. and its subsidiaries

37.7.12. CVD Transmit Index 0 Configuration

Name: CVDTX0
Offset: 0x0C0
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
			TXOUT3[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
			TXOUT2[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
			TXOUT1[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
			TXOUT0[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 29:24 – TXOUT3[5:0] CVDT channel to use for TX Index 3

Bits 21:16 – TXOUT2[5:0] CVDT channel to use for TX Index 2

Bits 13:8 – TXOUT1[5:0] CVDT channel to use for TX Index 1

Bits 5:0 – TXOUT0[5:0] CVDT channel to use for TX Index 0

© 2025 Microchip Technology Inc. and its subsidiaries

37.7.13. CVD Transmit Index 1 Configuration

Name: CVDTX1
Offset: 0x0C4
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
			TXOUT7[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
			TXOUT6[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
			TXOUT5[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
			TXOUT4[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 29:24 – TXOUT7[5:0] CVDT channel to use for TX Index 7

Bits 21:16 – TXOUT6[5:0] CVDT channel to use for TX Index 6

Bits 13:8 – TXOUT5[5:0] CVDT channel to use for TX Index 5

Bits 5:0 – TXOUT4[5:0] CVDT channel to use for TX Index 4

© 2025 Microchip Technology Inc. and its subsidiaries

37.7.14. CVD Transmit Index 2 Configuration

Name: CVDTX2
Offset: 0x0C8
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
			TXOUT11[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
			TXOUT10[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
			TXOUT9[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
			TXOUT8[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 29:24 – TXOUT11[5:0] CVDT channel to use for TX Index 11

Bits 21:16 – TXOUT10[5:0] CVDT channel to use for TX Index 10

Bits 13:8 – TXOUT9[5:0] CVDT channel to use for TX Index 9

Bits 5:0 – TXOUT8[5:0] CVDT channel to use for TX Index 8

© 2025 Microchip Technology Inc. and its subsidiaries

37.7.15. CVD Transmit Index 3 Configuration

Name: CVDTX3
Offset: 0x0CC
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
			TXOUT15[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
			TXOUT14[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
			TXOUT13[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
			TXOUT12[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 29:24 – TXOUT15[5:0] CVDT channel to use for TX Index 15

Bits 21:16 – TXOUT14[5:0] CVDT channel to use for TX Index 14

Bits 13:8 – TXOUT13[5:0] CVDT channel to use for TX Index 13

Bits 5:0 – TXOUT12[5:0] CVDT channel to use for TX Index 12

© 2025 Microchip Technology Inc. and its subsidiaries

37.7.16. CVD Transmit Index 4 Configuration

Name: CVDTX4
Offset: 0x0D0
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			TXOUT17[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
			TXOUT16[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 13:8 – TXOUT17[5:0] CVDT channel to use for TX Index 17

Bits 5:0 – TXOUT16[5:0] CVDT channel to use for TX Index 16

© 2025 Microchip Technology Inc. and its subsidiaries

37.7.17. CVD Scan Descriptor 0 Control 1

Name: CVDSD0C1
Offset: 0x100
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	SD0TH[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	23	22	21	20	19	18	17	16
	SD0TH[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	15	14	13	12	11	10	9	8
	SD0TH[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		SD0OVRSAMP[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 31:8 – SD0TH[23:0] Scan Descriptor Threshold
The controller subtracts the accumulators after all oversampling is complete. The controller compares the result of subtraction to this threshold to determine if the threshold exceeded asserts internally, which is used to generate an interrupt and/or store data to the FIFO, depending on the settings of other control bits.

Bits 6:0 – SD0OVRSAMP[6:0] Scan Descriptor Over Sampling
Determines the amount of oversampling done on each measurement.

Value	Description
0	Accumulates one measurement
1	Accumulates two measurements
...	—
127	Accumulates 128 measurements

© 2025 Microchip Technology Inc. and its subsidiaries

37.7.18. CVD Scan Descriptor 0 Control 2

Name: CVDSD0C2
Offset: 0x104
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	SD0TXSTRIDE	SD0TXSTRIDE	SD0TXEND[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	SD0TXSTRIDE	SD0TXSTRIDE	SD0TXBEG[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SD0RXSTRIDE	SD0RXSTRIDE	SD0RXEND[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SD0RXSTRIDE	SD0RXSTRIDE	SD0RXBEG[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31, 30 – SD0TXSTRIDE Scan Descriptor TX Index Stride
Determines the number of TX Indexes included in a single measurement.

- 30 – LSB
- 31 – MSB

Value	Description
4'b0	One TX Index
4'bF	16 TX Indexes

Bits 29:24 – SD0TXEND[5:0] Scan Descriptor TX Index End
Determines the last TX index to include in a scan. The SD0TXSTRIDE+1 value increments the TX index pointer, meets or exceeds this value, then, the TX loop of the scan is complete.

Bits 23, 22 – SD0TXSTRIDE Scan Descriptor TX Index Stride
Determines the number of TX Indexes included in a single measurement.

- 22 – LSB
- 23 – MSB

Value	Description
4'b0	One TX Index
4'bF	16 TX Indexes

Bits 21:16 – SD0TXBEG[5:0] Scan Descriptor TX Index Start
Determines the first TX index to include in a scan.

© 2025 Microchip Technology Inc. and its subsidiaries

Bits 15, 14 – SD0RXSTRIDE Scan Descriptor RX Index Stride
Determines the number of RX indices included in a single measurement.

Value	Description
4'b0	One RX Index
4'bF	16 RX Indexes

Bits 13:8 – SD0RXEND[5:0] Scan Descriptor RX Index End
Determines the last RX index to include in a scan. The SD0RXSTRIDE+1 value increments the RX index pointer, meets or exceeds this value, then, the RX loop of the scan is complete.

Bits 7, 6 – SD0RXSTRIDE Scan Descriptor RX Index Stride
Determines the number of RX indices included in a single measurement.

Value	Description
4'b0	One RX Index
4'bF	16 RX Indexes

Bits 5:0 – SD0RXBEG[5:0] Scan Descriptor RX Index Start
Determines the first RX index to include in a scan

© 2025 Microchip Technology Inc. and its subsidiaries

37.7.19. CVD Scan Descriptor 0 Control 3

Name: CVDSD0C3
Offset: 0x108
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	SD0EN[1:0]				SD0BUF	SD0INTEN	SD0SELF	SD0MUT
Access	R/W/HC	R/W/HC			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

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

Bit	15	14	13	12	11	10	9	8
		SD0ACQTIME[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		SD0CHGTIME[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 31:30 – SD0EN[1:0] Scan Descriptor Enable Mode

Value	Description
00	Disables the Scan Descriptor
01	Executes the Scan Descriptor one time only, then clears the enable
10	Executes the Scan Descriptor and keeps it enabled. Then, moves on to the next enabled descriptors.
11	Executes the Scan Descriptor in a loop until a threshold match is detected, then clears the enable mode and moves on to the next enabled descriptors.

Bit 27 – SD0BUF Scan Descriptor CVD Buffer Enable

Value	Description
1	Uses CVD buffer output as shared ADC input
0	Does not use CVD buffer output as shared ADC input

Bit 26 – SD0INTEN Scan Descriptor Interrupt Enable

Value	Description
1	Scan descriptor creates an interrupt if the accumulator threshold is met
0	Scan descriptor does not create an interrupt

Bit 25 – SD0SELF Scan Descriptor Self Measurement Mode

Value	Description
1	Enables the Self Measurement mode; RX outputs are part of CVD measurement and are driven
0	Disables the Self Measurement mode; RX outputs are not part of CVD measurements

Bit 24 – SD0MUT Scan Descriptor Mutual Mode

© 2025 Microchip Technology Inc. and its subsidiaries

Value	Description
1	Enables the Mutual Measurement mode; TX outputs are part of CVD measurement and are driven
0	Disables the Mutual Measurement mode; TX outputs are not part of CVD measurements

Bits 23:16 – SDnADCCON[7:0]
Configuration bits to pass to ADC when this scan descriptor is active. The system-level interconnect determines exactly what each of these bits controls. Examples is CPLEN (Capacitor Party Line Enable).

Bits 14:8 – SD0ACQTIME[6:0] Scan Descriptor Acquire Time
Specifies the time that CVD waits for the ADC voltage to settle.

Bits 6:0 – SD0CHGTIME[6:0] Scan Descriptor Charge Time
Specifies the time that CVD remains in the charging state for internal or external capacitors and for TX outputs

© 2025 Microchip Technology Inc. and its subsidiaries

37.7.20. CVD Scan Descriptor 0 Time 2

Name: CVDSD0T2
Offset: 0x10C
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
		SD0CHNTIME[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
		SD0OVRTIME[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
		SD0POLTIME[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		SD0CONTIME[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 30:24 – SD0CHNTIME[6:0] Scan Descriptor Channel Time
Controls the number of cycles the state machine waits in the RXCHAN or TXCHAN state before moving to the next RX/TX pair

Bits 22:16 – SD0OVRTIME[6:0] Scan Descriptor Oversample Time
Controls the number of cycles the state machine waits in the OVERSAMP state before taking the next oversampling measurement of an RX/TX pair

Bits 14:8 – SD0POLTIME[6:0] Scan Descriptor Polarity Time
Controls the number of cycles the state machine waits in the POLARITY state before taking the second polarity measurement of an RX/TX pair

Bits 6:0 – SD0CONTIME[6:0] Scan Descriptor Convert Time
Controls the number of cycles the state machine waits in the CONVERT state for the ADC sample data. User must ensure that the ADC asserts End-of-Convert (EOC) before the CONVERT state timer expires

© 2025 Microchip Technology Inc. and its subsidiaries

37.7.21. CVD Scan Descriptor 1 Control 1

Name: CVDSD1C1
Offset: 0x110
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	SD1TH[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	23	22	21	20	19	18	17	16
	SD1TH[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	15	14	13	12	11	10	9	8
	SD1TH[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		SD1OVRSAMP[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 31:8 – SD1TH[23:0] Scan Descriptor Threshold
The controller subtracts the accumulators after all oversampling is complete. The controller compares the result of subtraction to this threshold to determine if the threshold exceeded asserts internally, which is used to generate an interrupt and/or store data to the FIFO, depending on the settings of other control bits.

Bits 6:0 – SD1OVRSAMP[6:0] Scan Descriptor Over Sampling
Determines the amount of oversampling done on each measurement.

Value	Description
0	Accumulates one measurement
1	Accumulates two measurements
...	—
127	Accumulates 128 measurements

© 2025 Microchip Technology Inc. and its subsidiaries

37.7.22. CVD Scan Descriptor 1 Control 2

Name: CVDSD1C2
Offset: 0x114
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	SD1TXSTRIDE	SD1TXSTRIDE	SD1TXEND[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	SD1TXSTRIDE	SD1TXSTRIDE	SD1TXBEG[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SD1RXSTRIDE	SD1RXSTRIDE	SD1RXEND[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SD1RXSTRIDE	SD1RXSTRIDE	SD1RXBEG[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 29:24 – SD1TXEND[5:0] Scan Descriptor TX Index End
Determines the last TX index to include in a scan. The SD1TXSTRIDE+1 value increments the TX index pointer, meets or exceeds this value; then, the TX loop of the scan is complete.

Bits 22,23,30,31 – SD1TXSTRIDE Scan Descriptor TX Index Stride
Determines the number of TX indices included in a single measurement.

Value	Description
4'b0	One TX Index
4'bF	16 TX Indexes

Bits 21:16 – SD1TXBEG[5:0] Scan Descriptor TX Index Start
Determines the first TX index to include in a scan

Bits 13:8 – SD1RXEND[5:0] Scan Descriptor RX Index End
Determines the last RX index to include in a scan. The SD1RXSTRIDE+1 value increments the RX index pointer, meets or exceeds this value; then, the RX loop of the scan is complete.

Bits 6,7,14,15 – SD1RXSTRIDE Scan Descriptor RX Index Stride
Determines the number of RX indices included in a single measurement

Value	Description
4'b0	One RX Index
4'bF	16 RX Indexes

Bits 5:0 – SD1RXBEG[5:0] Scan Descriptor RX Index Start
Determines the first RX index to include in a scan

© 2025 Microchip Technology Inc. and its subsidiaries

37.7.23. CVD Scan Descriptor 1 Control 3

Name: CVDSD1C3
Offset: 0x118
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	SD1EN[1:0]				SD1BUF	SD1INTEN	SD1SELF	SD1MUT
Access	R/W/HC	R/W/HC			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

Bit	23	22	21	20	19	18	17	16
					CVDEN	CVDCPL[2:0]		
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
		SD1ACQTIME[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		SD1CHGTIME[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 31:30 – SD1EN[1:0] Scan Descriptor Enable Mode

Value	Description
00	Disables the Scan Descriptor.
01	Executes the Scan Descriptor one time only, then clears the enable.
10	Executes the Scan Descriptor, and keeps it enabled. Then, moves on to the next enabled descriptors.
11	Executes the Scan Descriptor in a loop until a threshold match is detected, then clears the enable mode and moves on to next enabled descriptors.

Bit 27 – SD1BUF Scan Descriptor CVD Buffer Enable

Value	Description
1	Uses CVD buffer output as shared ADC (ADC2) input.
0	Does not use CVD buffer output as shared ADC (ADC2) input.

Bit 26 – SD1INTEN Scan Descriptor Interrupt Enable

Value	Description
1	Scan Descriptor creates an interrupt if the accumulator threshold is met.
0	Scan descriptor does not create an interrupt.

Bit 25 – SD1SELF Scan Descriptor Self Measurement Mode

Value	Description
1	Enables the Self Measurement mode; RX outputs are part of CVD measurement and are driven.
0	Disables the Self Measurement mode; RX outputs are not part of CVD measurements.

Bit 24 – SD1MUT Scan Descriptor Mutual Mode

© 2025 Microchip Technology Inc. and its subsidiaries

Value	Description
1	Enables the Mutual Measurement mode; TX outputs are part of CVD measurement and are driven.
0	Disables the Mutual Measurement mode; TX outputs are not part of CVD measurements.

Bit 19 – CVDEN Capacitive Voltage Division Enable bit

Value	Description
1	CVD operation is enabled
0	CVD operation is disabled

Bits 18:16 – CVDCPL[2:0] Capacitor Voltage Divider (CVD) Setting bits

Value	Description
111	7 * 2.5 pF = 17.5 pF
110	6 * 2.5 pF = 15 pF
101	5 * 2.5 pF = 12.5 pF
100	4 * 2.5 pF = 10 pF
011	3 * 2.5 pF = 7.5 pF
010	2 * 2.5 pF = 5 pF
001	1 * 2.5 pF = 2.5 pF
000	0 * 2.5 pF = 0 pF

Bits 14:8 – SD1ACQTIME[6:0] Scan Descriptor Acquire Time
Specifies the time that CVD waits for ADC voltage to settle.

Bits 6:0 – SD1CHGTIME[6:0] Scan Descriptor Charge Time
Specifies the time that CVD remains in the charging state for internal or external capacitors and for TX outputs.

© 2025 Microchip Technology Inc. and its subsidiaries

37.7.24. CVD Scan Descriptor 1 Time 2

Name: CVDSD1T2
Offset: 0x11C
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
		SD1CHNTIME[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
		SD1OVRTIME[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
		SD1POLTIME[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		SD1CONTIME[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 30:24 – SD1CHNTIME[6:0] Scan Descriptor Channel Time
Controls the number of cycles the state machine waits in the RXCHAN or TXCHAN state before moving to the next RX/TX pair.

Bits 22:16 – SD1OVRTIME[6:0] Scan Descriptor Oversample Time
Controls the number of cycles the state machine waits in the OVERSAMP state before taking the next oversampling measurement of an RX/TX pair.

Bits 14:8 – SD1POLTIME[6:0] Scan Descriptor Polarity Time
Controls the number of cycles the state machine waits in the POLARITY state before taking the second polarity measurement of an RX/TX pair.

Bits 6:0 – SD1CONTIME[6:0] Scan Descriptor Convert Time
Controls the number of cycles the state machine waits in the CONVERT state for the ADC sample data. User must ensure that the ADC asserts End-of-Convert (EOC) before the CONVERT state timer expires.

© 2025 Microchip Technology Inc. and its subsidiaries

37.7.25. CVD Scan Descriptor 2 Control 1

Name: CVDSD2C1
Offset: 0x120
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	SD2TH[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	23	22	21	20	19	18	17	16
	SD2TH[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	15	14	13	12	11	10	9	8
	SD2TH[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		SD2OVRSAMP[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 31:8 – SD2TH[23:0] Scan Descriptor Threshold
The controller subtracts the accumulators after all oversampling is complete. The controller compares the result of subtraction to this threshold to determine if the threshold exceeded asserts internally, which is used to generate an interrupt and/or store data to the FIFO, depending on the settings of other control bits.

Bits 6:0 – SD2OVRSAMP[6:0] Scan Descriptor Over Sampling
Determines the amount of oversampling done on each measurement.

Value	Description
0	Accumulates one measurement
1	Accumulates two measurements
...	—
127	Accumulates 128 measurements

© 2025 Microchip Technology Inc. and its subsidiaries

37.7.26. CVD Scan Descriptor 2 Control 2

Name: CVDSD2C2
Offset: 0x124
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	SD2TXSTRIDE	SD2TXSTRIDE	SD2TXEND[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	SD2TXSTRIDE	SD2TXSTRIDE	SD2TXBEG[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SD2RXSTRIDE	SD2RXSTRIDE	SD2RXEND[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SD2RXSTRIDE	SD2RXSTRIDE	SD2RXBEG[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 29:24 – SD2TXEND[5:0] Scan Descriptor TX Index End
Determines the last TX index to include in a scan. The SD2TXSTRIDE+1 value increments the TX index pointer, meets or exceeds this value; then, the TX loop of the scan is complete.

Bits 22,23,30,31 – SD2TXSTRIDE Scan Descriptor TX Index Stride
Determines the number of TX indices included in a single measurement.

Value	Description
4'b0	One TX Index
4'bF	16 TX Indexes

Bits 21:16 – SD2TXBEG[5:0] Scan Descriptor TX Index Start
Determines the first TX index to include in a scan

Bits 13:8 – SD2RXEND[5:0] Scan Descriptor RX Index End
Determines the last RX index to include in a scan. The SD2RXSTRIDE+1 value increments the RX index pointer, meets or exceeds this value; then, the RX loop of the scan is complete.

Bits 6,7,14,15 – SD2RXSTRIDE Scan Descriptor RX Index Stride
Determines the number of RX indices included in a single measurement

Value	Description
4'b0	One RX Index
4'bF	16 RX Indexes

Bits 5:0 – SD2RXBEG[5:0] Scan Descriptor RX Index Start
Determines the first RX index to include in a scan

© 2025 Microchip Technology Inc. and its subsidiaries

37.7.27. CVD Scan Descriptor 2 Control 3

Name: CVDSD2C3
Offset: 0x128
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	SD2EN[1:0]				SD2BUF	SD2INTEN	SD2SELF	SD2MUT
Access	R/W/HC	R/W/HC			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

Bit	23	22	21	20	19	18	17	16
					CVDEN	CVDCPL[2:0]		
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
		SD2ACQTIME[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		SD2CHGTIME[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 31:30 – SD2EN[1:0] Scan Descriptor Enable Mode

Value	Description
00	Disables the Scan Descriptor.
01	Executes the Scan Descriptor one time only, then clears the enable.
10	Executes the Scan Descriptor, and keeps it enabled. Then, moves on to the next enabled descriptors.
11	Executes the Scan Descriptor in a loop until a threshold match is detected, then clears the enable mode and moves on to next enabled descriptors.

Bit 27 – SD2BUF Scan Descriptor CVD Buffer Enable

Value	Description
1	Uses CVD buffer output as shared ADC (ADC2) input.
0	Does not use CVD buffer output as shared ADC (ADC2) input.

Bit 26 – SD2INTEN Scan Descriptor Interrupt Enable

Value	Description
1	Scan Descriptor creates an interrupt if the accumulator threshold is met.
0	Scan descriptor does not create an interrupt.

Bit 25 – SD2SELF Scan Descriptor Self Measurement Mode

Value	Description
1	Enables the Self Measurement mode; RX outputs are part of CVD measurement and are driven.
0	Disables the Self Measurement mode; RX outputs are not part of CVD measurements.

Bit 24 – SD2MUT Scan Descriptor Mutual Mode

© 2025 Microchip Technology Inc. and its subsidiaries

Value	Description
1	Enables the Mutual Measurement mode; TX outputs are part of CVD measurement and are driven.
0	Disables the Mutual Measurement mode; TX outputs are not part of CVD measurements.

Bit 19 – CVDEN Capacitive Voltage Division Enable bit

Value	Description
1	CVD operation is enabled
0	CVD operation is disabled

Bits 18:16 – CVDCPL[2:0] Capacitor Voltage Divider (CVD) Setting bits

Value	Description
111	7 * 2.5 pF = 17.5 pF
110	6 * 2.5 pF = 15 pF
101	5 * 2.5 pF = 12.5 pF
100	4 * 2.5 pF = 10 pF
011	3 * 2.5 pF = 7.5 pF
010	2 * 2.5 pF = 5 pF
001	1 * 2.5 pF = 2.5 pF
000	0 * 2.5 pF = 0 pF

Bits 14:8 – SD2ACQTIME[6:0] Scan Descriptor Acquire Time
Specifies the time that CVD waits for ADC voltage to settle

Bits 6:0 – SD2CHGTIME[6:0] Scan Descriptor Charge Time
Specifies the time that CVD remains in the charging state for internal or external capacitors and for TX outputs

© 2025 Microchip Technology Inc. and its subsidiaries

37.7.28. CVD Scan Descriptor 2 Time 2

Name: CVDSD2T2
Offset: 0x12C
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
		SD2CHNTIME[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
		SD2OVRTIME[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
		SD2POLTIME[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		SD2CONTIME[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 30:24 – SD2CHNTIME[6:0] Scan Descriptor Channel Time
Controls the number of cycles the state machine waits in the RXCHAN or TXCHAN state before moving to the next RX/TX pair.

Bits 22:16 – SD2OVRTIME[6:0] Scan Descriptor Oversample Time
Controls the number of cycles the state machine waits in the OVERSAMP state before taking the next oversampling measurement of an RX/TX pair.

Bits 14:8 – SD2POLTIME[6:0] Scan Descriptor Polarity Time
Controls the number of cycles the state machine waits in the POLARITY state before taking the second polarity measurement of an RX/TX pair.

Bits 6:0 – SD2CONTIME[6:0] Scan Descriptor Convert Time
Controls the number of cycles the state machine waits in the CONVERT state for the ADC sample data. The user must ensure that the ADC asserts End-of-Convert (EOC) before the CONVERT state timer expires.

© 2025 Microchip Technology Inc. and its subsidiaries

37.7.29. CVD Scan Descriptor 3 Control 1

Name: CVDSD3C1
Offset: 0x130
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	SD3TH[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	23	22	21	20	19	18	17	16
	SD3TH[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	15	14	13	12	11	10	9	8
	SD3TH[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		SD3OVRSAMP[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 31:8 – SD3TH[23:0] Scan Descriptor Threshold
The controller subtracts the accumulators after all oversampling is complete. The controller compares the result of subtraction to this threshold to determine if the threshold exceeded asserts internally, which is used to generate an interrupt and/or store data to the FIFO, depending on the settings of other control bits.

Bits 6:0 – SD3OVRSAMP[6:0] Scan Descriptor Over Sampling
Determines the amount of oversampling done on each measurement

Value	Description
0	Accumulates one measurement
1	Accumulates two measurements
...	—
127	Accumulates 128 measurements

© 2025 Microchip Technology Inc. and its subsidiaries

37.7.30. CVD Scan Descriptor 3 Control 2

Name: CVDSD3C2
Offset: 0x134
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	SD3TXSTRIDE	SD3TXSTRIDE	SD3TXEND[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	SD3TXSTRIDE	SD3TXSTRIDE	SD3TXBEG[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SD3RXSTRIDE	SD3RXSTRIDE	SD3RXEND[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SD3RXSTRIDE	SD3RXSTRIDE	SD3RXBEG[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 29:24 – SD3TXEND[5:0] Scan Descriptor TX Index End
Determines the last TX index to include in a scan. The SD3TXSTRIDE+1 value increments the TX index pointer, meets or exceeds this value; then, the TX loop of the scan is complete.

Bits 22,23,30,31 – SD3TXSTRIDE Scan Descriptor TX Index Stride
Determines the number of TX Indexes included in a single measurement

Value	Description
4'b0	One TX Index
4'bF	16 TX Indexes

Bits 21:16 – SD3TXBEG[5:0] Scan Descriptor TX Index Start
Determines the first TX index to include in a scan

Bits 13:8 – SD3RXEND[5:0] Scan Descriptor RX Index End
Determines the last RX index to include in a scan. The SD3RXSTRIDE+1 value increments the RX index pointer, meets or exceeds this value; then, the RX loop of the scan is complete.

Bits 6,7,14,15 – SD3RXSTRIDE Scan Descriptor RX Index Stride
Determines the number of RX indexes included in a single measurement

Value	Description
4'b0	One RX Index
4'bF	16 RX Indexes

Bits 5:0 – SD3RXBEG[5:0] Scan Descriptor RX Index Start
Determines the first RX index to include in a scan

© 2025 Microchip Technology Inc. and its subsidiaries

37.7.31. CVD Scan Descriptor 3 Control 3

Name: CVDSD3C3
Offset: 0x138
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	SD3EN[1:0]				SD3BUF	SD3INTEN	SD3SELF	SD3MUT
Access	R/W/HC	R/W/HC			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

Bit	23	22	21	20	19	18	17	16
					CVDEN	CVDCPL[2:0]		
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
		SD3ACQTIME[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		SD3CHGTIME[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 31:30 – SD3EN[1:0] Scan Descriptor Enable Mode

Value	Description
00	Disables the Scan Descriptor.
01	Executes the Scan Descriptor one time only, then clears the enable.
10	Executes the Scan Descriptor, and keeps it enabled. Then, moves on to the next enabled descriptors.
11	Executes the Scan Descriptor in a loop until a threshold match is detected, then, clears the enable mode and moves on to next enabled descriptors.

Bit 27 – SD3BUF Scan Descriptor CVD Buffer Enable

Value	Description
1	Uses CVD buffer output as shared ADC (ADC2) input.
0	Does not use CVD buffer output as shared ADC (ADC2) input.

Bit 26 – SD3INTEN Scan Descriptor Interrupt Enable

Value	Description
1	Scan Descriptor creates an interrupt if the accumulator threshold is met.
0	Scan descriptor does not create an interrupt.

Bit 25 – SD3SELF Scan Descriptor Self Measurement Mode

Value	Description
1	Enables the Self Measurement mode; RX outputs are part of CVD measurement and are driven.
0	Disables the Self Measurement mode; RX outputs are not part of CVD measurements.

Bit 24 – SD3MUT Scan Descriptor Mutual Mode

© 2025 Microchip Technology Inc. and its subsidiaries

Value	Description
1	Enables the Mutual Measurement mode; TX outputs are part of CVD measurement and are driven.
0	Disables the Mutual Measurement mode; TX outputs are not part of CVD measurements.

Bit 19 – CVDEN Capacitive Voltage Division Enable bit

Value	Description
1	CVD operation is enabled
0	CVD operation is disabled

Bits 18:16 – CVDCPL[2:0] Capacitor Voltage Divider (CVD) Setting bits

Value	Description
111	7 * 2.5 pF = 17.5 pF
110	6 * 2.5 pF = 15 pF
101	5 * 2.5 pF = 12.5 pF
100	4 * 2.5 pF = 10 pF
011	3 * 2.5 pF = 7.5 pF
010	2 * 2.5 pF = 5 pF
001	1 * 2.5 pF = 2.5 pF
000	0 * 2.5 pF = 0 pF

Bits 14:8 – SD3ACQTIME[6:0] Scan Descriptor Acquire Time
Specifies the time that CVD waits for ADC voltage to settle

Bits 6:0 – SD3CHGTIME[6:0] Scan Descriptor Charge Time
Specifies the time that CVD remains in the Charging state for internal or external capacitors and for TX outputs

© 2025 Microchip Technology Inc. and its subsidiaries

37.7.32. CVD Scan Descriptor 3 Time 2

Name: CVDSD3T2
Offset: 0x13C
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
		SD3CHNTIME[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
		SD3OVRTIME[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
		SD3POLTIME[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		SD3CONTIME[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 30:24 – SD3CHNTIME[6:0] Scan Descriptor Channel Time
Controls the number of cycles the state machine waits in the RXCHAN or TXCHAN state before moving to the next RX/TX pair

Bits 22:16 – SD3OVRTIME[6:0] Scan Descriptor Oversample Time
Controls the number of cycles the state machine waits in the OVERSAMP state before taking the next oversampling measurement of an RX/TX pair

Bits 14:8 – SD3POLTIME[6:0] Scan Descriptor Polarity Time
Controls the number of cycles the state machine waits in the POLARITY state before taking the second polarity measurement of an RX/TX pair

Bits 6:0 – SD3CONTIME[6:0] Scan Descriptor Convert Time
Controls the number of cycles the state machine waits in the CONVERT state for the ADC sample data. The user must ensure that the ADC asserts End-of-Convert (EOC) before the CONVERT state timer expires.

© 2025 Microchip Technology Inc. and its subsidiaries

38. Analog Comparators (AC)

38.1. Overview

The Analog Comparator (AC) supports two individual comparators:

- One shared (with built-in supply voltage monitor (MVREF)) AC_CMP1
- One dedicated AC_CMP0

Each comparator (COMP) compares the voltage levels on two inputs and provides a digital output based on the comparison. Each comparator may be configured to generate interrupt requests and/or peripheral events upon several different combinations of input changes.

The input selection includes up to four shared analog port pins and several internal signals. Each Comparator Output state can also be output on a pin for use by external devices.

The comparators are grouped in pairs on each port. The AC peripheral implements one pair of comparators, Comparator 0 (AC_COMP0) and Comparator 1 (AC_COMP1) as a pair.

They have identical behaviors but separate Control registers. Each pair can be set in the Window mode to compare a signal to a voltage range instead of a single voltage level.

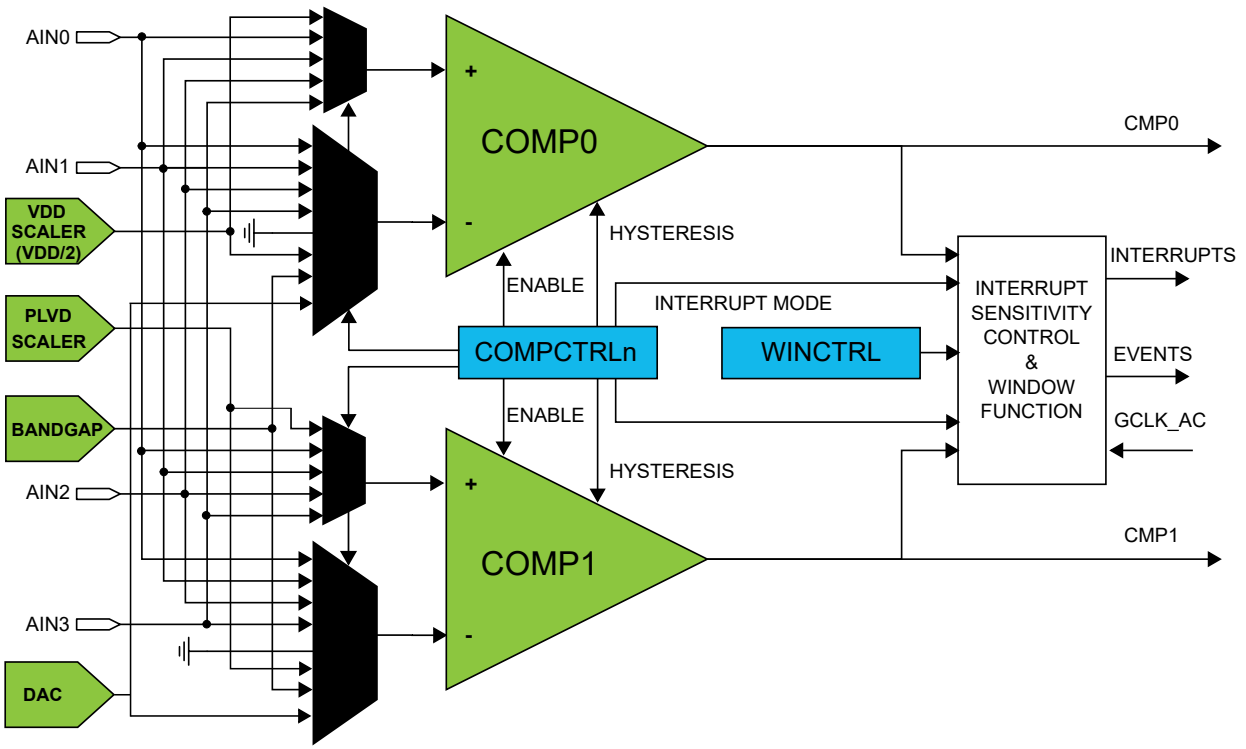
38.2. Features

- Two Individual Comparators (Single Pair Configuration)
- Analog Comparator Outputs Available on Pins
  - Asynchronous or Synchronous
- Flexible Input Selection:
  - Up to four pins selectable for positive or negative inputs
  - Ground (for zero crossing)
  - Bandgap reference voltage
  - Programmable VDD scaler for AC_CMP1 (Shared with Programmable Low Voltage Detector (PLVD)) and fixed VDD/2 for AC_CMP0
  - DAC output
- Interrupt Generation on:
  - Rising or falling edge
  - Toggle
  - End of comparison
- Window Function Interrupt Generation on:
  - Signal above window
  - Signal inside window
  - Signal below window
  - Signal outside window
- Event Generation on:
  - Comparator output
  - Window function inside/outside window
- Optional Digital Filter on Comparator Output

© 2025 Microchip Technology Inc. and its subsidiaries

38.3. Block Diagram

Figure 38-1. Analog Comparator Block Diagram



38.4. Product Dependencies

The following sections describe how the other parts of the system must be configured correctly to use this peripheral.

38.4.1. I/O Lines

Using the AC's I/O lines requires the I/O pins to be configured as analog pins for AC_AINx inputs. See *I/O Ports and Peripheral Pin Select (PPS)* from Related Links.

Table 38-1. I/O LINES

Signal	Peripheral Function
AC_AIN0	Comparator input
AC_AIN1	Comparator input
AC_AIN2	Comparator input
AC_AIN3	Comparator input
AC_CMP0	Comparator output
AC_CMP1	Comparator output

Note:

1. To get the analog comparator output on the I/O line, CFGCON1.CMP0_OE/CFGCON1.CMP1_OE needs to be set/enabled. See *System Configuration and Register Locking (CFG)* from Related Links.

Related Links

I/O Ports and Peripheral Pin Select (PPS)

System Configuration and Register Locking (CFG)

© 2025 Microchip Technology Inc. and its subsidiaries

38.4.2. Power Management

The AC continues to operate in any Sleep mode (Idle, Standby Sleep) where the selected source clock is running. The AC's interrupts can be used to wake up the device from the sleep modes. Events connected to the event system can trigger other operations in the system without exiting the Standby Sleep mode.

38.4.3. Clocks

The AC bus clock (PB2_CLK) can be enabled and disabled in the Main Clock module, CRU (see *Clock and Reset Unit (CRU)* from Related Links), and the Analog Comparator module can be enabled or disabled via the PMD1 register. See *Peripheral Module Disable Register (PMD)* from Related Links.

A generic clock (GCLK_AC) is required to clock the AC. This clock must be configured and enabled in the generic clock controller before using the AC.

This generic clock is asynchronous to the bus clock (PB2_CLK). Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains. See *Synchronization* from Related Links.

Related Links

Clock and Reset Unit (CRU)

Peripheral Module Disable (PMD)

Synchronization

38.4.4. DMA

Not applicable.

38.4.5. Interrupts

The interrupt request lines are connected to the interrupt controller. Using the AC interrupts requires the interrupt controller to be configured first. See *Nested Vector Interrupt Controller (NVIC)* from Related Links.

Related Links

Nested Vector Interrupt Controller (NVIC)

38.4.6. Events

The events are connected to the Event System. See *Event System (EVSYS)* from Related Links for details on how to configure the Event System.

Related Links

Event System (EVSYS)

38.4.7. Debug Operation

When the CPU is halted in debug mode, the AC will halt normal operation after any ongoing comparison is completed. The AC can be forced to continue normal operation during debugging. See *DBGCTRL* register from Related Links. If the AC is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

Related Links

DBGCTRL

Debug Control

38.4.8. Register Access Protection

All registers with write access can be write protected optionally by the PAC, except for the following registers:

© 2025 Microchip Technology Inc. and its subsidiaries

- Control B register (CTRLB)
- Interrupt Flag register (INTFLAG)

Optional write protection by the 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.

38.4.9. Analog Connections

Each comparator has up to four I/O pins that can be used as analog inputs. The pair of comparators shares the same four pins. These pins must be configured for analog operation before using them as comparator inputs.

Any internal reference source, such as a bandgap voltage reference, DAC must be configured and enabled prior to its use as a comparator input.

38.5. Functional Description

38.5.1. Principle of Operation

Each comparator has one positive input and one negative input. Each positive input may be chosen from a selection of analog input pins. Each negative input may be chosen from a selection of both analog input pins and internal inputs, such as bandgap voltage reference.

The digital output from the comparator is '1' when the difference between the positive and the negative input voltage is positive and '0' otherwise.

The individual comparators can be used independently (Normal mode) or paired to form a window comparison (Window mode).

38.5.2. Basic Operation

38.5.2.1. Initialization

Some registers are enable-protected, meaning they can only be written when the module is disabled.

The following register is enable-protected:

- Event Control register (EVCTRL)

Enable protection is denoted by the Enable-Protected property in each individual register description.

38.5.2.2. Enabling, Disabling and Resetting

The AC is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The AC is disabled by writing a '0' to CTRLA.ENABLE. In addition to this AC module enable configuration, AC must also be enabled in the PMD module.

The AC is reset by writing a '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the AC are reset to their initial state, and the AC is disabled. See *CTRLA* register from Related Links.

Related Links

CTRLA

Control A

38.5.2.3. Comparator Configuration

Each individual comparator must be configured by its respective Comparator Control register (COMPCTRLx) before that comparator is enabled. These settings cannot be changed while the comparator is enabled.

© 2025 Microchip Technology Inc. and its subsidiaries

- Select the desired measurement mode with COMPCTRLx.SINGLE. See *Starting a Comparison* from Related Links.
- Fixed hysteresis
- Fixed speed of operation
- Select the interrupt source with COMPCTRLx.INTSEL.
- Select the positive and negative input sources with the COMPCTRLx.MUXPOS and COMPCTRLx.MUXNEG bits. See *Selecting Comparator Inputs* from Related Links.
- Select the filtering option with COMPCTRLx.FLEN.
- Select the standby operation with the Run in Standby bit (COMPCTRLx.RUNSTDBY).

The individual comparators are enabled by writing a '1' to the Enable bit in the Comparator x Control registers (COMPCTRLx.ENABLE). The individual comparators are disabled by writing a '0' to COMPCTRLx.ENABLE. Writing a '0' to CTRLA.ENABLE will also disable all the comparators but will not clear their COMPCTRLx.ENABLE bits.

Related Links

Selecting Comparator Inputs

38.5.2.4. Starting a Comparison

Each comparator channel can be in one of two different measurement modes, which is determined by the COMPCTRLx.SINGLE bit:

- Continuous measurement
- Single-shot

After being enabled, a start-up delay is required before the result of the comparison is ready. This start-up time is measured automatically to account for environmental changes, such as temperature or voltage supply level, and is specified in the Electrical Specifications. During the start-up time, the COMP output is not predictable until the Comparator x Ready bit in the Status B register (STATUSB.READYx) is set.

The comparator can be configured to generate interrupts when the output toggles, when the output changes from '0' to '1' (rising edge), when the output changes from '1' to '0' (falling edge) or at the end of the comparison. An end-of-comparison interrupt can be used with the Single-Shot mode to chain further events in the system, regardless of the state of the comparator outputs. The Interrupt mode is set by the Interrupt Selection bit group in the Comparator Control register (COMPCTRLx.INTSEL). Events are generated using the comparator output state regardless of whether the interrupt is enabled or not.

38.5.2.4.1. Continuous Measurement

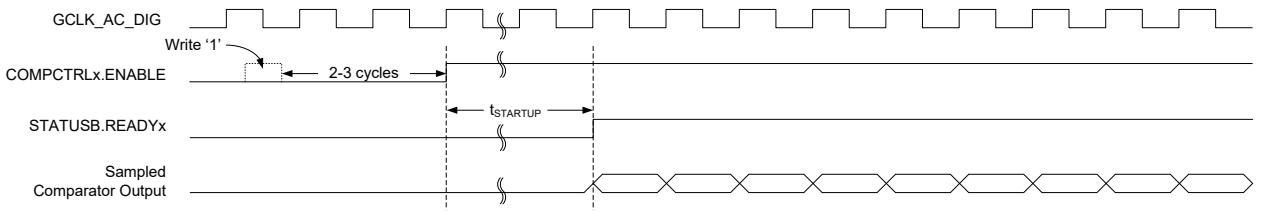
Continuous measurement is selected by writing COMPCTRLx.SINGLE to zero. In continuous mode, the comparator is continuously enabled and performing comparisons. This ensures that the result of the latest comparison is always available in the Current State bit in the Status A register (STATUSA.STATEx).

After the start-up time has passed, a comparison is done and STATUSA is updated. The Comparator x Ready bit in the Status B register (STATUSB.READYx) is set, and the appropriate peripheral events and interrupts are also generated. New comparisons are performed continuously until the COMPCTRLx.ENABLE bit is written to zero. The start-up time applies only to the first comparison.

In the continuous operation, edge detection of the comparator output for interrupts is done by comparing the current and previous sample. The sampling rate is the GCLK_AC frequency. An example of continuous measurement is shown in the following figure.

© 2025 Microchip Technology Inc. and its subsidiaries

Figure 38-2. Continuous Measurement Example



For low-power operation, comparisons can be performed during sleep mode without a clock. The comparator is enabled continuously, and changes of the comparator state are detected asynchronously. When a toggle occurs, the CRU will start GCLK_AC to register the appropriate peripheral events and interrupts. The GCLK_AC clock is, then, disabled again automatically unless configured to wake up the system from sleep.

38.5.2.4.2. Single-Shot

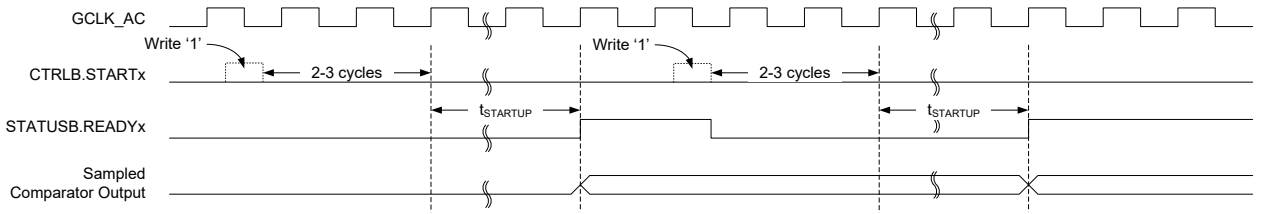
Single-shot operation is selected by writing COMPCTRLx.SINGLE to '1'. During single-shot operation, the comparator is normally idle. The user starts a single comparison by writing '1' to the respective Start Comparison bit in the write-only Control B register (CTRLB.STARTx). The comparator is enabled and, after the start-up time has passed, a single comparison is done and STATUSA is updated. Appropriate peripheral events and interrupts are also generated. No new comparisons will be performed.

Writing '1' to CTRLB.STARTx also clears the Comparator x Ready bit in the Status B register (STATUSB.READYx). STATUSB.READYx is set automatically by hardware when the single comparison has completed.

A single-shot measurement can also be triggered by the Event System. Setting the Comparator x Event Input bit in the Event Control Register (EVCTRL.COMPEIx) enables triggering on incoming peripheral events. Each comparator can be triggered independently by separate events. Event-triggered operation is similar to user-triggered operation; the difference is that a peripheral event from another hardware module causes the hardware to automatically start the comparison and will not clear STATUSB.READYx.

To detect an edge of the comparator output in single-shot operation for the purpose of interrupts, the result of the current measurement is compared with the result of the previous measurement (one sampling period earlier). An example of single-shot operation is shown in the following figure.

Figure 38-3. Single-Shot Example



For low-power operation, event-triggered measurements can be performed during sleep modes. When the event occurs, the CRU will start GCLK_AC. The comparator is enabled and, after the start-up time has passed, a comparison is done and appropriate peripheral events and interrupts are also generated. The comparator and GCLK_AC are, then, disabled again automatically unless configured to wake up the system from sleep.

38.5.3. Selecting Comparator Inputs

Each comparator has one positive and one negative input. The positive input is one of the external input pins (AINx). The negative input can be fed either from an external input pin (AINx) or from

© 2025 Microchip Technology Inc. and its subsidiaries

one of the internal reference voltage sources common to all comparators. The user selects the input source as follows:

- The positive input is selected by the Positive Input MUX Select bit group in the Comparator Control register (COMPCTRLx.MUXPOS).
- The negative input is selected by the Negative Input MUX Select bit group in the Comparator Control register (COMPCTRLx.MUXNEG).

In the case of using an external I/O pin, the selected pin must be configured for analog use in the GPIO by disabling the digital input and output. The switching of the analog input multiplexers is controlled to minimize crosstalk between the channels. The input selection must be changed only while the individual comparator is disabled.

Note: For internal use of the comparison results by the CCL module (see *Configurable Custom Logic (CCL)* from Related Links), COMPCTRLx.OUT must be `0x1` or `0x2`.

Related Links

Configurable Custom Logic (CCL)

38.5.4. Window Operation

The comparator pair can be configured to work together in Window mode. In this mode, a voltage range is defined and the comparators give information about whether an input signal is within this range or not. Window mode is enabled by the Window Enable x bit in the Window Control register (WINCTRL.WENx). Both comparators must have the same measurement mode setting in their respective Comparator Control Registers (COMPCTRLx.SINGLE).

Notes:

1. Both comparators must be enabled (COMPCTRLx.ENABLE = 1) before the Window mode is enabled (WINCTRL.WEN0 = 1).
2. Window mode must be first disabled (WINCTRL.WEN0 = 0) before both comparators are disabled (COMPCTRLx.ENABLE = 0).

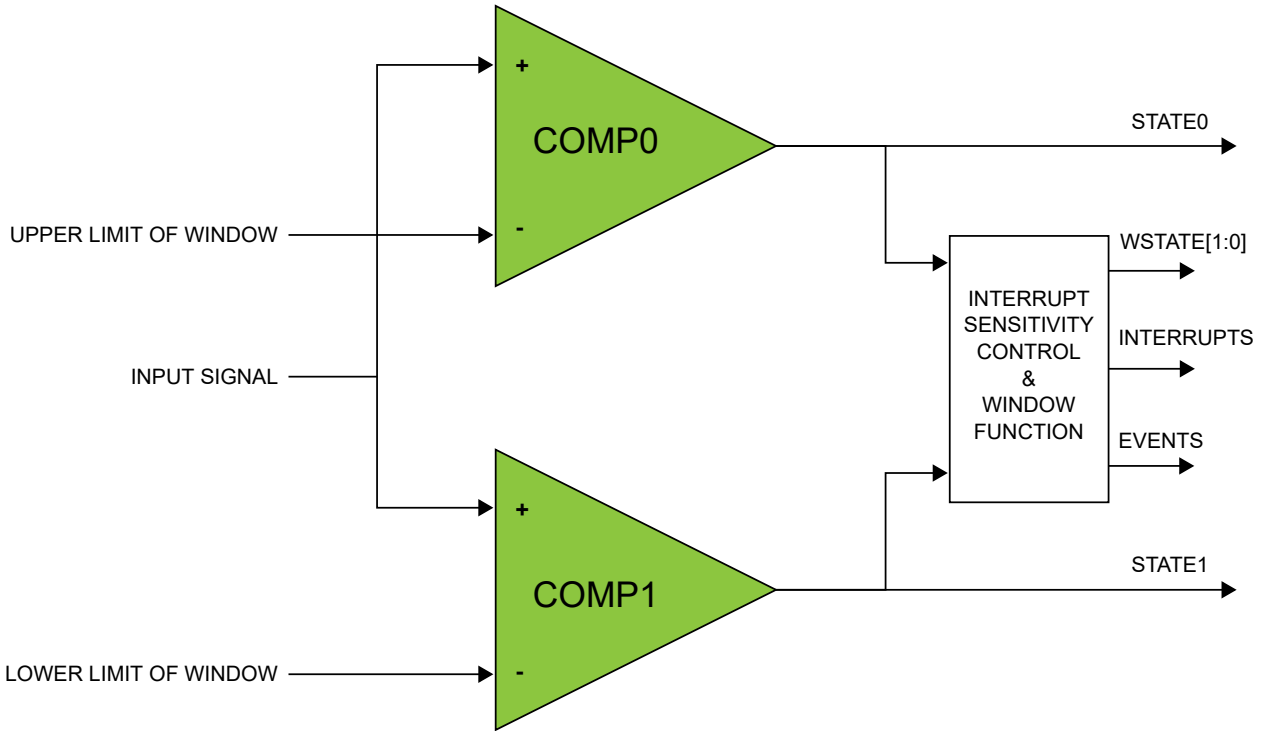
To physically configure the comparators for Window mode, the same I/O pin must be chosen as positive input for each comparator, providing a shared input signal. The negative inputs define the range for the window. In the figure below, COMP0 defines the upper limit and COMP1 defines the lower limit of the window as shown, but the window will also work in the opposite configuration with COMP0 lower and COMP1 higher. The current state of the window function is available in the Window x State bit group of the Status register (STATUS.WSTATEx).

Window mode can be configured to generate interrupts when the input voltage changes to below the window, when the input voltage changes to above the window, when the input voltage changes into the window or when the input voltage changes outside the window. The interrupt selections are set by the Window Interrupt Selection bit field in the Window Control register (WINCTRL.WINTSEL). Events are generated using the inside/outside state of the window, regardless of whether the interrupt is enabled or not. Note that the individual comparator outputs, interrupts and events continue to function normally during Window mode.

When the comparators are configured for Window mode and Single-shot mode, measurements are performed simultaneously on both comparators. Writing '1' to either Start Comparison bit in the Control B register (CTRLB.STARTx) will start a measurement. Likewise, either peripheral event can start a measurement. The STATUSA.WSTATE bit field must be read only after STATUSB.READY bits of both the related comparators are set.

© 2025 Microchip Technology Inc. and its subsidiaries

Figure 38-4. Comparators in Window Mode

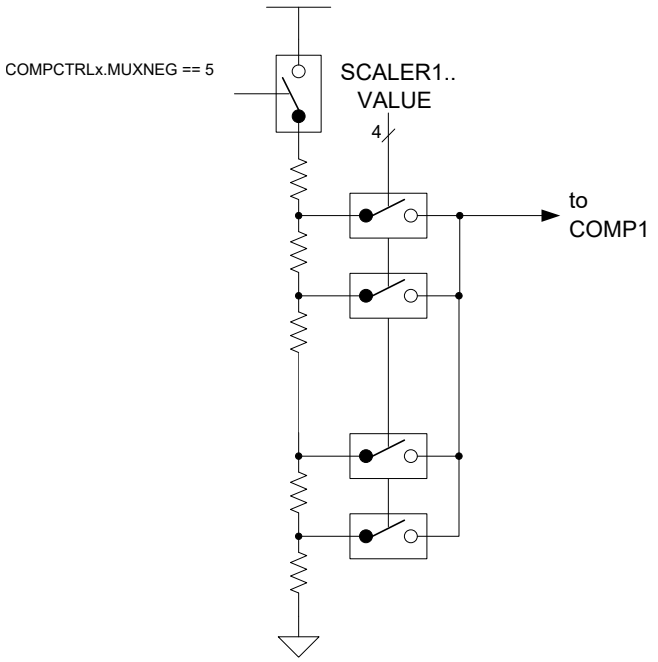


38.5.5. VDD Scaler

The VDD scaler generates a reference voltage that is a fraction of the device's supply voltage with 16 levels. The programmable VDD scaler (PLVD Scaler) is available for AC_CMP1 only. AC_CMP0 uses a fixed VDD/2 reference. The scaler of a comparator is enabled when the Negative Input Mux bit field or the Positive Input Mux in the respective Comparator Control register (COMPCTRLx.MUXNEG) is set to `0x5` and the comparator is enabled. The voltage of each channel is selected by the Value bit field in the SCALER1 registers (SCALER1.VALUE) using the VDD resistor ladder.

© 2025 Microchip Technology Inc. and its subsidiaries

Figure 38-5. PLVD Scaler

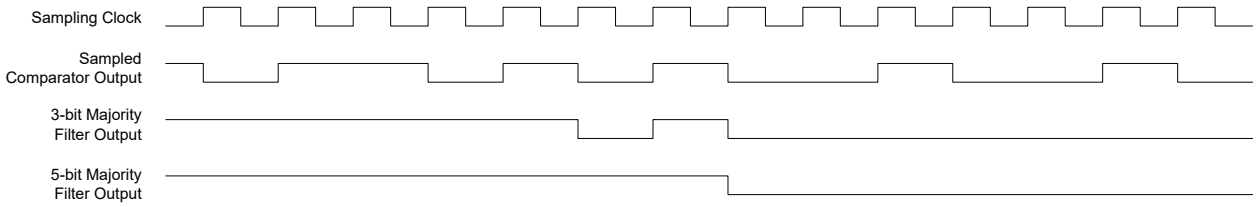


38.5.6. Filtering

The output of the comparators can be filtered digitally to reduce noise. The filtering is determined by the Filter Length bits in the Comparator Control x register (COMPCTRLx.FLEN), and is independent for each comparator. Filtering is selectable from none, 3-bit majority (N=3) or 5-bit majority (N=5) functions. Any change in the comparator output is considered valid only if N/2+1 out of the last N samples agree. The filter sampling rate is the GCLK_AC frequency.

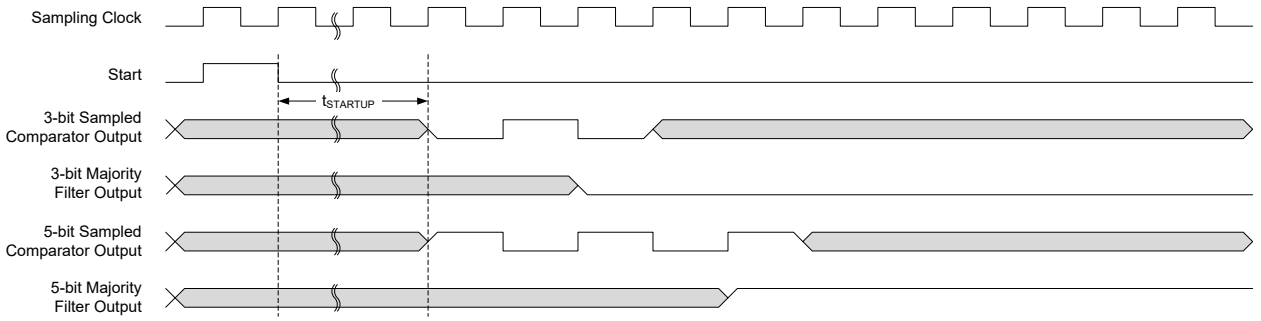
Note that filtering creates an additional delay of N-1 sampling cycles from when a comparison is started until the comparator output is validated. For Continuous mode, the first valid output will occur when the required number of filter samples is taken. Subsequent outputs will be generated every cycle based on the current sample plus the previous N-1 samples, as shown in Figure 38-6. For Single-shot mode, the comparison completes after the Nth filter sample, as shown in Figure 38-7.

Figure 38-6. Continuous Mode Filtering



© 2025 Microchip Technology Inc. and its subsidiaries

Figure 38-7. Single-Shot Filtering



During Sleep modes, filtering is supported only for single-shot measurements. Filtering must be disabled if continuous measurements will be done during Sleep modes, or the resulting interrupt/event may be generated incorrectly.

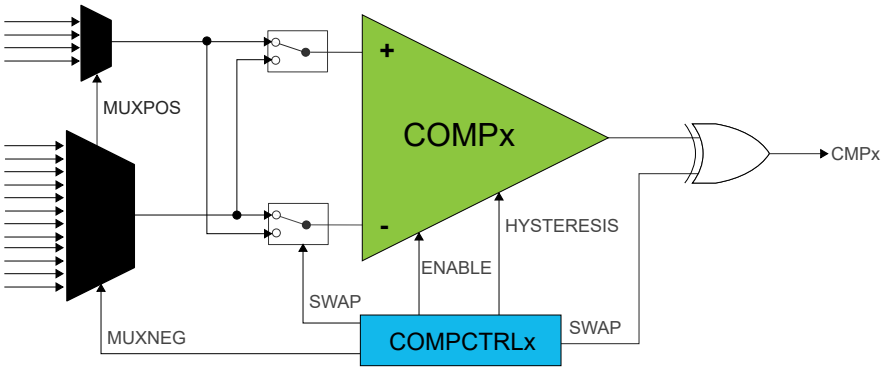
38.5.7. Comparator Output

The output of each comparator can be routed to an I/O pin by setting the Output bit group in the Comparator Control x register (COMPCTRLx.OUT) also set CFGCON1.CMP0_OE/CFGCON1.CMP1_OE. This allows the comparator to be used by external circuitry. Either the raw, non-synchronized output of the comparator or the CLK_AC-synchronized version, including filtering, can be used as the I/O signal source. The output appears on the corresponding CMP[x] pin.

38.5.8. Offset Compensation

The Swap bit in the Comparator Control registers (COMPCTRLx.SWAP) controls switching of the input signals to a comparator's positive and negative terminals. When the comparator terminals are swapped, the output signal from the comparator is also inverted, as shown in Figure 38-8. This allows the user to measure or compensate for the comparator input offset voltage. As part of the input selection, COMPCTRLx.SWAP can be changed only while the comparator is disabled.

Figure 38-8. Input Swapping for Offset Compensation



Note: The inputs are swapped only if the comparator inputs difference is within offset values.

38.5.9. DMA Operation

Not applicable.

38.5.10. Interrupts

The AC has the following interrupt sources:

- Comparator (COMP0, COMP1): Indicates a change in comparator status
- Window (WIN0): Indicates a change in the window status

© 2025 Microchip Technology Inc. and its subsidiaries

Comparator interrupts are generated based on the conditions selected by the Interrupt Selection bit group in the Comparator Control registers (COMPCTRLx.INTSEL). Window interrupts are generated based on the conditions selected by the Window Interrupt Selection bit group in the Window Control register (WINCTRL.WINTSEL[1:0]).

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

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled or the AC is Reset. See *INTFLAG* register from Related Links for details on how to clear interrupt flags. All interrupt requests from the peripheral are OR'ed together on the system level to generate one combined interrupt request to the NVIC. The user must read the INTFLAG register to determine which interrupt condition is present. See *Nested Vector Interrupt Controller (NVIC)* from Related Links.

Note: Interrupts must be globally enabled for interrupt requests to be generated.

Related Links

Nested Vector Interrupt Controller (NVIC)

INTFLAG

Interrupt Flag Status and Clear

38.5.11. Events

The AC can generate the following output events:

- Comparator (COMP0, COMP1): Generated as a copy of the comparator status
- Window (WIN0): Generated as a copy of the window inside/outside status

Writing a one to an Event Output bit in the Event Control Register (EVCTRL.xxEO) enables the corresponding output event. Writing a zero to this bit disables the corresponding output event. See *Event System (EVSYS)* from Related Links for details on how to configure the Event System.

The AC can take the following action on an input event:

- Start comparison (START0, START1): Start a comparison

Writing a one to an Event Input bit into the Event Control register (EVCTRL.COMPEIx) enables the corresponding action on an input event. Writing a zero to this bit disables the corresponding action on an input event. Note that if several events are connected to the AC, the enabled action will be taken on any of the incoming events. See *Event System (EVSYS)* from Related Links for details on how to configure the Event System.

When EVCTRL.COMPEIx is one, the event will start a comparison on COMPx after the start-up time delay. In normal mode, each comparator responds to its corresponding input event independently. For a pair of comparators in window mode, either comparator event will trigger a comparison on both comparators simultaneously.

Related Links

Event System (EVSYS)

38.5.12. Sleep Mode Operation

The Run in Standby bits in the Comparator x Control registers (COMPCTRLx.RUNSTDBY) control the behavior of the AC during standby sleep mode. Each RUNSTDBY bit controls one comparator. When the bit is zero, the comparator is disabled during sleep, but maintains its current configuration. When the bit is one, the comparator continues to operate during sleep. Note that when RUNSTDBY

© 2025 Microchip Technology Inc. and its subsidiaries

is zero, the analog blocks are powered off for the lowest power consumption. This necessitates a start-up time delay when the system returns from sleep.

For Window Mode operation, both comparators in a pair must have the same RUNSTDBY configuration.

When RUNSTDBY is one, any enabled AC interrupt source can wake up the CPU. The AC can also be used during sleep modes where the clock used by the AC is disabled, provided that the AC is still powered (not in shutdown). In this case, the behavior is slightly different and depends on the measurement mode, as listed in Table 38-2.

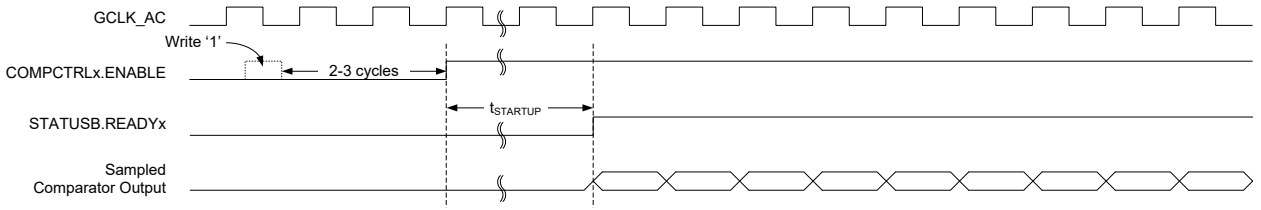
Table 38-2. Sleep Mode Operation

COMPCTRLx.MODE	RUNSTDBY=0	RUNSTDBY=1
0 (Continuous)	COMPx disabled	GCLK_AC stopped, COMPx enabled
1 (Single-shot)	COMPx disabled	GCLK_AC stopped, COMPx enabled only when triggered by an input event

38.5.12.1. Continuous Measurement during Sleep

When a comparator is enabled in continuous measurement mode and GCLK_AC is disabled during sleep, the comparator will remain continuously enabled and will function asynchronously. The current state of the comparator is asynchronously monitored for changes. If an edge matching the interrupt condition is found, GCLK_AC is started to register the interrupt condition and generate events. If the interrupt is enabled in the Interrupt Enable registers (INTENCLR/SET), the AC can wake up the device; otherwise GCLK_AC is disabled until the next edge detection. Filtering is not possible with this configuration.

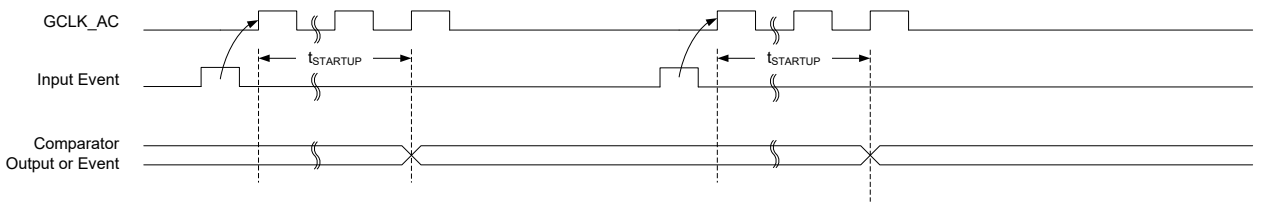
Figure 38-9. Continuous Mode SleepWalking



38.5.12.2. Single-Shot Measurement during Sleep

For low-power operation, event-triggered measurements can be performed during sleep modes. When the event occurs, the CRU will start GCLK_AC. The comparator is enabled and, after the start-up time has passed, a comparison is done, with filtering if desired, and the appropriate peripheral events and interrupts are also generated as shown in the following figure. The comparator and GCLK_AC are, then, disabled again automatically unless configured to wake the system from sleep. Filtering is allowed with this configuration.

Figure 38-10. Single-Shot SleepWalking



38.5.13. Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

© 2025 Microchip Technology Inc. and its subsidiaries

The following bits are synchronized when written:

- Software Reset bit in the Control register (CTRLA.SWRST)
- Enable bit in the Control register (CTRLA.ENABLE)
- Enable bit in the 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.

© 2025 Microchip Technology Inc. and its subsidiaries

38.6. Register Summary

See the *AC* module in the *Product Memory Mapping Overview* from Related Links for the base address.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0							ENABLE	SWRST
0x01	CTRLB	7:0							START1	START0
0x02	EVCTRL	7:0				WINEO0			COMPEO1	COMPEO0
		15:8			INVEI1	INVEI0			COMPEI1	COMPEI0
0x04	INTENCLR	7:0				WIN0			COMP1	COMP0
0x05	INTENSET	7:0				WIN0			COMP1	COMP0
0x06	INTFLAG	7:0				WIN0			COMP1	COMP0
0x07	STATUSA	7:0			WSTATE0[1:0]				STATE1	STATE0
0x08	STATUSB	7:0							READY1	READY0
0x09	DBGCTRL	7:0								DBGRUN
0x0A	WINCTRL	7:0						WINTSEL0[1:0]		WEN0
0x0B	Reserved									
0x0C	SCALER0	7:0			VALUE[5:0]					
0x0D	SCALER1	7:0					VALUE[3:0]			
0x0E ... 0x0F	Reserved									
0x10	COMPCTRL0	7:0		RUNSTDBY		INTSEL[1:0]		SINGLE	ENABLE	
		15:8	SWAP		MUXPOS[2:0]				MUXNEG[2:0]	
		23:16								
		31:24			OUT[1:0]				FLEN[2:0]	
0x14	COMPCTRL1	7:0		RUNSTDBY		INTSEL[1:0]		SINGLE	ENABLE	
		15:8	SWAP		MUXPOS[2:0]				MUXNEG[2:0]	
		23:16								
		31:24			OUT[1:0]				FLEN[2:0]	
0x18 ... 0x1F	Reserved									
0x20	SYNCBUSY	7:0				COMPCTRL1	COMPCTRL0	WINCTRL	ENABLE	SWRST
		15:8								
		23:16								
		31:24								

Related Links

Product Memory Mapping Overview

38.7. Register Description

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

Some registers are optionally write-protected by the PAC. Optional PAC write protection is denoted by the PAC Write-Protection property in each individual register description. See *Register Access Protection* from Related Links.

Some registers are synchronized when read and/or written. Synchronization is denoted by the Write-Synchronized or the Read-Synchronized property in each individual register description. See *Synchronization* from Related Links.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable protection is denoted by the Enable-Protected property in each individual register description.

Related Links

Register Access Protection

Synchronization

© 2025 Microchip Technology Inc. and its subsidiaries

38.7.1. Control A

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

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

Bit 1 – ENABLE Enable

Due to synchronization, there is a delay from updating the register until the peripheral is enabled/disabled. The value written to CTRL.ENABLE reads back immediately, and the corresponding bit in the Synchronization Busy register (SYNCBUSY.ENABLE) is set. SYNCBUSY.ENABLE is cleared when the peripheral is enabled/disabled.

Note: To avoid spurious interrupts from multiple enable/disable cycles of AC, use the SWRST bit to reset the comparator module.

Value	Description
0	The AC is disabled.
1	The AC is enabled. Each comparator must also be enabled individually by the Enable bit in the Comparator Control register (COMPCTRLn.ENABLE).

Bit 0 – SWRST Software Reset

Writing a '0' to this bit has no effect.
Writing a '1' to this bit resets all registers in the AC to their initial state and the AC is disabled.
Writing a '1' to CTRLA.SWRST always takes precedence, meaning that all other writes in the same write operation are discarded.
Due to synchronization, there is a delay from writing CTRLA.SWRST until the Reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the Reset is complete.

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

© 2025 Microchip Technology Inc. and its subsidiaries

38.7.2. Control B

Name: CTRLB
Offset: 0x01
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
							START1	START0
Access							R/W	R/W
Reset							0	0

Bits 0, 1 – STARTx Comparator x Start Comparison
Writing a '0' to this field has no effect.
Writing a '1' to STARTx starts a single-shot comparison on COMPx if both the Single-Shot and Enable bits in the Comparator x Control Register are '1' (COMPCTRLx.SINGLE and COMPCTRLx.ENABLE). If Comparator x is not implemented or if it is not enabled in Single-shot mode, writing a '1' has no effect.
This bit always reads as '0'.

© 2025 Microchip Technology Inc. and its subsidiaries

38.7.3. Event Control

Name: EVCTRL
Offset: 0x02
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
			INVEI1	INVEI0			COMPEI1	COMPEI0
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit	7	6	5	4	3	2	1	0
				WINEO0			COMPEO1	COMPEO0
Access				R/W			R/W	R/W
Reset				0			0	0

Bits 12, 13 – INVEIx Inverted Event Input Enable x

Value	Description
0	Incoming event is not inverted for comparator x.
1	Incoming event is inverted for comparator x.

Bits 8, 9 – COMPEIx Comparator x Event Input
Note that several actions can be enabled for incoming events. If several events are connected to the peripheral, the enabled action will be taken for any of the incoming events. There is no way to tell which of the incoming events caused the action.
These bits indicate whether a comparison will start or not on any incoming event.

Value	Description
0	Comparison will not start on any incoming event.
1	Comparison will start on any incoming event.

Bit 4 – WINEO0 Window 0 Event Output Enable
These bits indicate whether the window 0 function can generate a peripheral event or not.

Value	Description
0	Window 0 Event is disabled.
1	Window 0 Event is enabled.

Bits 0, 1 – COMPEOx Comparator x Event Output Enable
These bits indicate whether the comparator x output can generate a peripheral event or not.

Value	Description
0	COMPx event generation is disabled.
1	COMPx event generation is enabled.

© 2025 Microchip Technology Inc. and its subsidiaries

38.7.4. Interrupt Enable Clear

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

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

Bit	7	6	5	4	3	2	1	0
				WIN0			COMP1	COMP0
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 4 – WIN0 Window 0 Interrupt Enable
Reading this bit returns the state of the Window 0 interrupt enable.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit disables the Window 0 interrupt.

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

Bits 0, 1 – COMPx Comparator x Interrupt Enable
Reading this bit returns the state of the Comparator x interrupt enable.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit disables the Comparator x interrupt.

Value	Description
0	The Comparator x interrupt is disabled.
1	The Comparator x interrupt is enabled.

© 2025 Microchip Technology Inc. and its subsidiaries

38.7.5. Interrupt Enable Set

Name: INTENSET
Offset: 0x05
Reset: 0x00
Property: PAC Write-Protection

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

Bit	7	6	5	4	3	2	1	0
				WIN0			COMP1	COMP0
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 4 – WIN0 Window 0 Interrupt Enable
Reading this bit returns the state of the Window 0 interrupt enable.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit enables the Window 0 interrupt.

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

Bits 0, 1 – COMPx Comparator x Interrupt Enable
Reading this bit returns the state of the Comparator x interrupt enable.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit sets the Ready interrupt bit and enable the Ready interrupt.

Value	Description
0	The Comparator x interrupt is disabled.
1	The Comparator x interrupt is enabled.

© 2025 Microchip Technology Inc. and its subsidiaries

38.7.6. Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x06
Reset: 0x00
Property: –

Bit	7	6	5	4	3	2	1	0
				WIN0			COMP1	COMP0
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 4 – WIN0 Window 0
This flag is set according to the Window 0 Interrupt Selection bit group in the WINCTRL register (WINCTRL.WINTSELx) and generates an interrupt if INTENCLR/SET.WINx is also '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Window 0 interrupt flag.

Bits 0, 1 – COMPx Comparator x
Reading this bit returns the status of the Comparator x interrupt flag. If comparator x is not implemented, COMPx always reads as '0'.
This flag is set according to the Interrupt Selection bit group in the Comparator x Control register (COMPCTRLx.INTSEL) and generates an interrupt if INTENCLR/SET.COMPx is also '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Comparator x interrupt flag.

© 2025 Microchip Technology Inc. and its subsidiaries

38.7.7. Status A

Name: STATUSA
Offset: 0x07
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
			WSTATE0[1:0]				STATE1	STATE0
Access			R	R			R	R
Reset			0	0			0	0

Bits 5:4 – WSTATE0[1:0] Window 0 Current State
These bits show the current state of the signal if the Window 0 mode is enabled.
These values may change during start-up and measurement cycles. When polling for sample completion, use both the STATUSB.READYx bits to signal completion.

Value	Name	Description
0x0	ABOVE	Signal is above window
0x1	INSIDE	Signal is inside window
0x2	BELOW	Signal is below window
0x3	—	Reserved

Bits 0, 1 – STATEx Comparator x Current State
This bit shows the current state of the output signal from COMPx. STATEx is valid only when the STATUSB.READYx bit is one.

© 2025 Microchip Technology Inc. and its subsidiaries

38.7.8. Status B

Name: STATUSB
Offset: 0x08
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
							READY1	READY0
Access							R	R
Reset							0	0

Bits 0, 1 – READYx Comparator x Ready

Value	Description
0	This bit is cleared when the Comparator x output is not ready.
1	This bit is set when the Comparator x output is ready.

© 2025 Microchip Technology Inc. and its subsidiaries

38.7.9. Debug Control

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

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

Bit 0 – DBGRUN Debug Run
This bit is not reset by a software Reset.
This bits controls the functionality when the CPU is halted by an external debugger.

Value	Description
0	The AC is halted when the CPU is halted by an external debugger. Any on-going comparison completes before AC is halted.
1	The AC continues normal operation when the CPU is halted by an external debugger.

© 2025 Microchip Technology Inc. and its subsidiaries

38.7.10. Window Control

Name: WINCTRL
Offset: 0x0A
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Bit	7	6	5	4	3	2	1	0
						WINTSEL0[1:0]		WEN0
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 2:1 – WINTSEL0[1:0] Window 0 Interrupt Selection
These bits configure the Interrupt mode for the comparator Window 0 mode.

Value	Name	Description
0x0	ABOVE	Interrupt on signal above window
0x1	INSIDE	Interrupt on signal inside window
0x2	BELOW	Interrupt on signal below window
0x3	OUTSIDE	Interrupt on signal outside window

Bit 0 – WEN0 Window 0 Mode Enable

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

© 2025 Microchip Technology Inc. and its subsidiaries

38.7.11. Scaler 0

Name: SCALER0
Offset: 0x0C
Reset: 0x00
Property: PAC Write-Protection

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

Bits 5:0 – VALUE[5:0] Scaler Value
Bits [5:0] are unused. Fixed selection of VDD/2.

© 2025 Microchip Technology Inc. and its subsidiaries

38.7.12. Scaler 1

Name: SCALER1
Offset: 0x0D
Reset: 0x00
Property: PAC Write-Protection

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

Bits 3:0 – VALUE[3:0] Scaler Value
These bits define the scaling factor for channel 1 of the VDD voltage scaler. The output voltage, V_{SCALE}, is:

$$V_{\mathrm{SCALE}} = V_{DD} \times \left(\frac{R\_Bottom}{R\_Total}\right)$$

Where, R_Total = 900.
Refer to the following table for R_Bottom for different VALUE[3:0]
For example, V_{SCALE} for VALUE[3:0] = 0x02 at V_{DD} = 3.3V

$$V_{\mathrm{SCALE}} = 3.3 \times \left(\frac{598.5}{900}\right) = 2.1945V$$

Table 38-3. Scaler Value

Value[3:0]	R_Bottom
0x0	Reserved
0x1	Reserved
0x2	598.5
0x3	634.5
0x4	306
0x5	324
0x6	328.5
0x7	360
0x8	387
0x9	400.5
0xA	432
0xB	450
0xC	468
0xD	490.5
0xE	481.5
0xF	External Reference on LVDIN pin

© 2025 Microchip Technology Inc. and its subsidiaries

38.7.13. Comparator Control n

Name: COMPCTRL
Offset: 0x10 + n*0x04 [n=0..1]
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
			OUT[1:0]			FLEN[2:0]		
Access			R/W	R/W		R/W	R/W	R/W
Reset			0	0		0	0	0

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	SWAP	MUXPOS[2:0]				MUXNEG[2:0]		
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bit	7	6	5	4	3	2	1	0
		RUNSTDBY		INTSEL[1:0]		SINGLE	ENABLE	
Access		R/W		R/W	R/W	R/W	R/W	
Reset		0		0	0	0	0	

Bits 29:28 – OUT[1:0] Output
These bits configure the output selection for Comparator n. COMPCTRLn.OUT can be written only while COMPCTRLn.ENABLE is '0'.
Note: For internal use of the comparison results by the CCL, this must be 0x1 or 0x2.

These bits are not synchronized.

Value	Name	Description
0x0	OFF	The output of COMPn is not routed to the COMPn I/O port
0x1	ASYNC	The asynchronous output of COMPn is routed to the COMPn I/O port
0x2	SYNC	The synchronous output (including filtering) of COMPn is routed to the COMPn I/O port
0x3	N/A	Reserved

Bits 26:24 – FLEN[2:0] Filter Length
These bits configure the filtering for Comparator n. COMPCTRLn.FLEN can only be written while COMPCTRLn.ENABLE is '0'.
These bits are not synchronized.

Value	Name	Description
0x0	OFF	No filtering
0x1	MAJ3	3-bit majority function (2 of 3)
0x2	MAJ5	5-bit majority function (3 of 5)
0x3-0x7	N/A	Reserved

Bit 15 – SWAP Swap Inputs and Invert
This bit swaps the positive and negative inputs to COMPn and inverts the output. This function can be used for offset cancellation. COMPCTRLn.SWAP can be written only while COMPCTRLn.ENABLE is '0'.

© 2025 Microchip Technology Inc. and its subsidiaries

These bits are not synchronized.

Value	Description
0	The output of MUXPOS connects to the positive input, and the output of MUXNEG connects to the negative input.
1	The output of MUXNEG connects to the positive input, and the output of MUXPOS connects to the negative input.

Bits 14:12 – MUXPOS[2:0] Positive Input Mux Selection

These bits select which input is connected to the positive input of Comparator n.
COMPCTRLn.MUXPOS can be written only while COMPCTRLn.ENABLE is '0'.
These bits are not synchronized.

Value	Name	Description
0x0	PIN0	AC_AIN0
0x1	PIN1	AC_AIN1
0x2	PIN2	AC_AIN2
0x3	PIN3	AC_AIN3
0x4	VSCALE	VDD scaler
0x5-0x7	—	Reserved

Bits 10:8 – MUXNEG[2:0] Negative Input Mux Selection

These bits select which input is connected to the negative input of Comparator n.
COMPCTRLn.MUXNEG can only be written while COMPCTRLn.ENABLE is '0'.
These bits are not synchronized.

Value	Name	Description
0x0	PIN0	I/O pin 0
0x1	PIN1	I/O pin 1
0x2	PIN2	I/O pin 2
0x3	PIN3	I/O pin 3
0x4	GND	Ground
0x5	VSCALE	VDD scaler
0x6	BANDGAP	Internal bandgap voltage
0x7	DAC	DAC output

Bit 6 – RUNSTDBY Run in Standby

This bit controls the behavior of the comparator during Standby Sleep mode.
This bit is not synchronized

Value	Description
0	The comparator is disabled during sleep.
1	The comparator continues to operate during sleep.

Bits 4:3 – INTSEL[1:0] Interrupt Selection

These bits select the condition for Comparator n to generate an interrupt or event.
COMPCTRLn.INTSEL can be written only while COMPCTRLn.ENABLE is '0'.
These bits are not synchronized.

Value	Name	Description
0x0	TOGGLE	Interrupt on comparator output toggle
0x1	RISING	Interrupt on comparator output rising
0x2	FALLING	Interrupt on comparator output falling
0x3	EOC	Interrupt on end of comparison (Single-shot mode only)

© 2025 Microchip Technology Inc. and its subsidiaries

Bit 2 – SINGLE Single-Shot Mode

This bit determines the operation of Comparator n. COMPCTRLn.SINGLE can be written only while COMPCTRLn.ENABLE is '0'.
These bits are not synchronized.

Value	Description
0	Comparator n operates in continuous Measurement mode.
1	Comparator n operates in Single-shot mode.

Bit 1 – ENABLE Enable

Writing a '0' to this bit disables Comparator n.
Writing a '1' to this bit enables Comparator n.
Due to synchronization, there is a delay from updating the register until the comparator is enabled/disabled. The value written to COMPCTRLn.ENABLE reads back immediately after being written. SYNCBUSY.COMPCTRLn is set. SYNCBUSY.COMPCTRLn is cleared when the peripheral is enabled/disabled.
Writing a '1' to COMPCTRLn.ENABLE prevents further changes to the other bits in COMPCTRLn. These bits remain protected until COMPCTRLn.ENABLE is written to '0' and the write is synchronized.

© 2025 Microchip Technology Inc. and its subsidiaries

38.7.14. Synchronization Busy

Name: SYNCBUSY
Offset: 0x20
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				COMPCTRL1	COMPCTRL0	WINCTRL	ENABLE	SWRST
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bits 3, 4 – COMPCTRLx COMPCTRLx Synchronization Busy
This bit is cleared when the synchronization of the COMPCTRLx register between the clock domains is complete.
This bit is set when the synchronization of the COMPCTRLx register between clock domains is started.

Bit 2 – WINCTRL WINCTRL Synchronization Busy
This bit is cleared when the synchronization of the WINCTRL register between the clock domains is complete.
This bit is set when the synchronization of the WINCTRL register between clock domains is started.

Bit 1 – ENABLE Enable Synchronization Busy
This bit is cleared when the synchronization of the CTRLA.ENABLE bit between the clock domains is complete.
This bit is set when the synchronization of the CTRLA.ENABLE bit between clock domains is started.

Bit 0 – SWRST Software Reset Synchronization Busy
This bit is cleared when the synchronization of the CTRLA.SWRST bit between the clock domains is complete.
This bit is set when the synchronization of the CTRLA.SWRST bit between clock domains is started.

© 2025 Microchip Technology Inc. and its subsidiaries

39. Digital-to-Analog Converter (DAC)

39.1. Overview

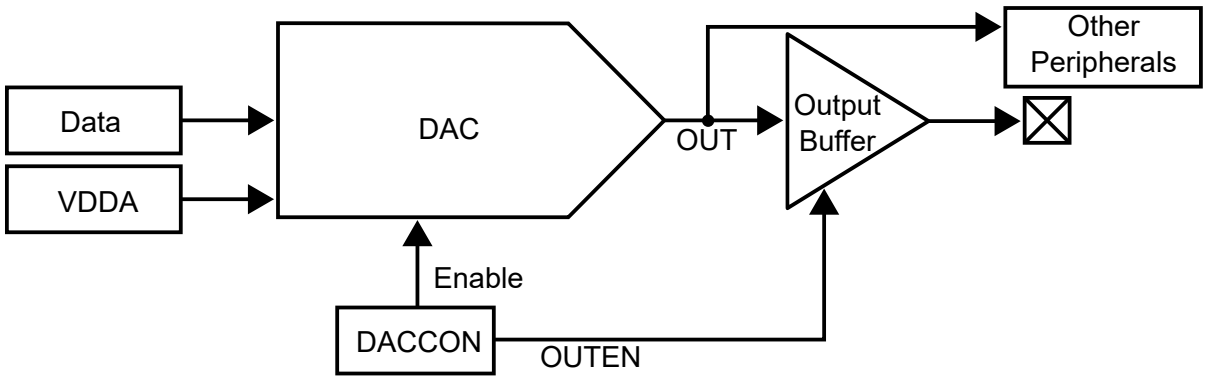
The PIC32CX-BZ6 7-bit Digital-to-Analog Converter (DAC), converts a digital value written to the Data (DACCODE.DATA) register to an analog voltage. The conversion range is between GND and AVDD. The DAC has one continuous time output with high drive capabilities. The application can start the DAC conversion by writing to the Data (DACCODE.DATA) register and enabling the DAC Module DACCON.EN bit to '1'.

39.2. Features

- Reference Voltage is Fixed to VDDA
- Seven Bits of Programmable Granularity Provided for Output Voltage with Six-Bit Accuracy
- Output is High Impedance in the Disabled State

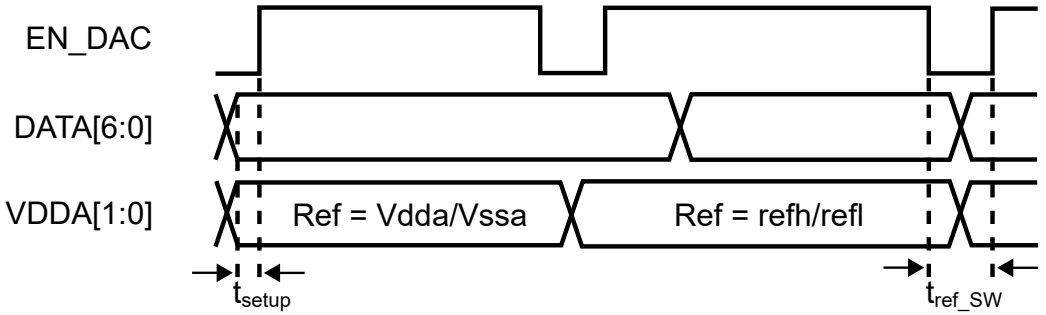
39.3. Block Diagram

Figure 39-1. Block Diagram



39.4. DAC Timing Diagram

Figure 39-2. DAC Timing Diagram



39.5. Functional Description

39.5.1. Initialization

The following steps are necessary to operate the DAC:

- The reference voltage fixed to VDDA.

© 2025 Microchip Technology Inc. and its subsidiaries

- Configure the usage of the DAC output:
  - Configure an internal peripheral to use the DAC output. Refer to the documentation of the respective peripherals.
  - Enable the output to a pin by writing a '1' to the Output Buffer Enable (OUTEN) bit. The Port peripheral needs to disable the input for the DAC pin.
- Write an initial digital value to the Data (DACCODE.DATA) register.
- Enable the DAC by writing a '1' to the ENABLE bit in the Control A (DACCON.EN) register.

39.5.2. Operation

39.5.2.1. Enabling, Disabling and Resetting

The user enables the DAC by writing a '1' to the ENABLE bit in the DACCON (DACCON.EN) register and disables it by writing a '0' to this bit.

39.5.2.2. Starting a Conversion

When the ENABLE bit in the DACCON (DACCON.EN) register is written to '1', a conversion starts as soon as the Data (DACCODE.DATA) register is written.

When the ENABLE bit in DACCON.EN is written to '0', writing to the Data register does not trigger a conversion. Instead, the conversion starts when the ENABLE bit in DACCON.EN is written to '1'.

39.5.2.3. DAC as a Source for Internal Peripheral

The analog output of the DAC can be internally connected to other peripherals when the ENABLE bit in the DACCON (DACCON.EN) register is written to '1'. When the DAC analog output is only being used internally, the Output Buffer Enable (DACCON.OUTEN) bit in DACCON can be '0'.

39.5.2.4. DAC Output on Pin

In the DACCON (DACCON.OUTEN) register, the user can connect the analog output of the DAC to a pin by writing a '1' to the Output Buffer Enable (OUTEN) bit. The pin used by the DAC must have the input disabled from the Port peripheral. There is an output buffer between the DAC output and the pin, which ensures the analog value does not depend on the load of the pin. The output buffer can only source current; it has very limited sinking capability.

39.5.2.5. Low Power Mode Operation

If DAC conversion stops in Low Power mode, DAC is also disabled to reduce power consumption. When exiting Low Power mode, DAC is enabled again; therefore, a certain start-up time is required before starting a new conversion.

- When LPRCEN = 1, the DACCON2 register is not used.
- To set a clock request for LPRC, write a '1' to the DACCON.LPRCEN bit in the DACCON register.
- SnH mode must be used with OUTEN. This mode does not support an internal peripheral.

© 2025 Microchip Technology Inc. and its subsidiaries

39.6. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	DACCON	7:0	SNH	OUTEN	LPRCEN(1)					
		15:8	EN							
		23:16								
		31:24								
0x04	DACCODE	7:0		DATA[6:0]						
		15:8								
		23:16								
		31:24								
0x08	DACCON2	7:0	WIDTH[7:0]							
		15:8							WIDTH[9:8]	
		23:16	PERIOD[7:0]							
		31:24	PRESCALAR[2:0]						PERIOD[9:8]	

© 2025 Microchip Technology Inc. and its subsidiaries

39.6.1. DACCON – Control Register

Name: DACCON
Offset: 0x00
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	EN							
Access	R/W							
Reset	0							

Bit	7	6	5	4	3	2	1	0
	SNH	OUTEN	LPRCEN(1)					
Access	R/W	R/W	R/W					
Reset	0	0	0					

Bit 15 – EN Enable

Value	Description
1	The peripheral is enabled
0	The peripheral is disabled

Bit 7 – SNH Sample and Hold Circuitry
To set the clock request for LPRC, write a '1' to this bit when the LPRCEN bit is set.

Value	Description
1	DAC operation in Low Power mode
0	DAC operation in Normal mode

Bit 6 – OUTEN Will Enable the Output Buffer

Value	Description
1	Output buffer enabled
0	Output buffer disabled

Bit 5 – LPRCEN(1) LPRC Clock Enabled for SnH Mode
Note:

1. The SnH clock to DAC can be either pb_clk or LPRC clk. Setting LPRCEN to '1' selects the LPRC clock as the SnH clock source when SNH = '1'. When LPRCEN='1', the DACCON2 register is not used.

Value	Description
1	LPRC clock enabled
0	LPRC clock not enabled

© 2025 Microchip Technology Inc. and its subsidiaries

39.6.2. DACCODE – DAC Code Register

Name: DACCODE
Offset: 0x04
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

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

Bits 6:0 – DATA[6:0] DAC Data
Digital value to be converted to analog voltage. 7-bit data that needs conversion is written here.

© 2025 Microchip Technology Inc. and its subsidiaries

39.6.3. DACCON2 – DAC Config Register(1)

Name: DACCON2
Offset: 0x08
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	PRESCALAR[2:0]						PERIOD[9:8]	
Access	R/W	R/W	R/W				R/W	R/W
Reset	0	0	0				0	0

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

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

Bit	7	6	5	4	3	2	1	0
	WIDTH[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 31:29 – PRESCALAR[2:0] Prescaling Factor for SnH Clock
0x0 - Sampling clock is SnH clock directly
0x1 - Sampling clock is SnH clock/2
0x2 - Sampling clock is SnH clock/4
0x3 - Sampling clock is SnH clock/8
0x4 - Sampling clock is SnH clock/16
0x5 - Sampling clock is SnH clock/32
0x6 - Sampling clock is SnH clock/64
0x7 - Sampling clock is SnH clock/128

Bits 25:16 – PERIOD[9:0] SnH Clock Period
Tperiod = prescaler_clk_period * (DACCON2[PERIOD] + 1)

Bits 9:0 – WIDTH[9:0] SnH Clock Width
Twidth = prescaler_clk_period * (DACCON2[WIDTH] + 1)

Note:

1. The DACCON2 register is applicable when SnH clock is pb_clk. It is not applicable when SnH = 0 or when LPRCEN = 0.

© 2025 Microchip Technology Inc. and its subsidiaries

40. Timer/Counter (TC)

40.1. Overview

There are up to 10 TC peripheral instances. Each TC consists of a counter, a prescaler, compare/capture channels and control logic. The counter can be set to count events or clock pulses. The counter, together with the compare/capture channels, can be configured to time stamp input events or I/O pin edges, allowing for capturing of frequency and/or pulse width.

A TC can also perform waveform generation, such as frequency generation and pulse-width modulation.

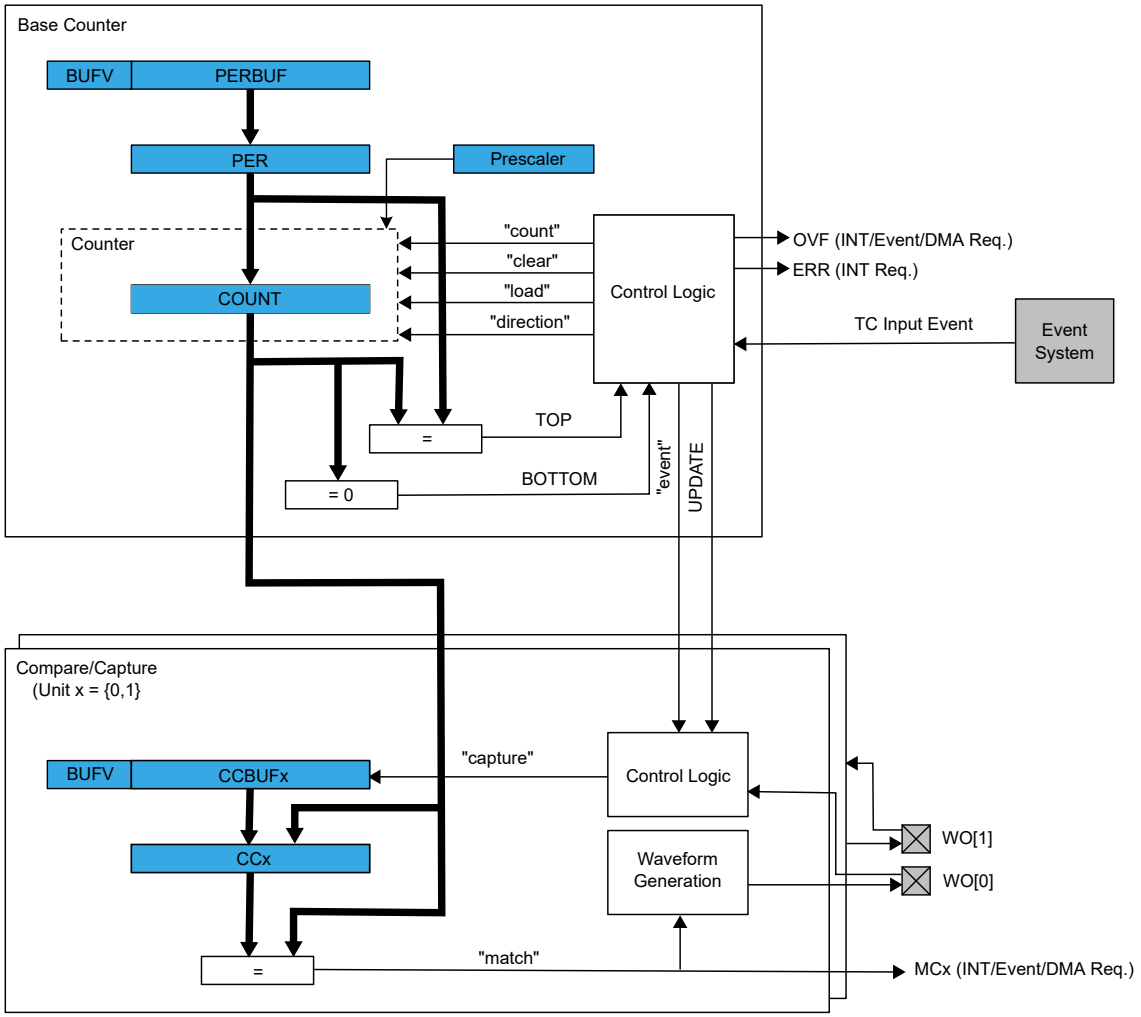
40.2. Features

- Selectable Configuration
  - 8-, 16- or 32-bit TC operation with compare/capture channels
- Two Compare/Capture Channels (CC) with:
  - Double buffered timer period setting (in 8-bit mode only)
  - Double buffered compare channel
- Waveform Generation
  - Frequency generation
  - Single-slope pulse-width modulation
- Input Capture
  - Event or I/O pin edge capture
  - Frequency capture
  - Pulse-width capture
  - Time-stamp capture
  - Minimum and maximum capture
- One Input Event
- Interrupts/Output Events on:
  - Counter overflow/underflow
  - Compare match or capture
- Internal Prescaler
- DMA support

© 2025 Microchip Technology Inc. and its subsidiaries

40.3. Block Diagram

Figure 40-1. Timer/Counter Block Diagram



40.4. Signal Description

Table 40-1. Signal Description for TC

Signal Name	Type	Description
WO[1:0]	Digital output	Waveform output
	Digital input	Capture input

For details on the pin mapping for this peripheral, see *I/O Ports and Peripheral Pin Select (PPS)* from Related Links. One signal can be mapped on several pins.

Related Links

I/O Ports and Peripheral Pin Select (PPS)

© 2025 Microchip Technology Inc. and its subsidiaries

40.5. Product Dependencies

The following sections describe how the other parts of the system must be configured correctly to use this peripheral.

40.5.1. I/O Lines

To use the I/O lines of this peripheral, the I/O pins must be configured using the I/O Pin Controller (PORT). See *Port Register Summary* from Related Links.

Related Links

Port Register Summary

40.5.2. Power Management

This peripheral can continue to operate in any sleep mode (Idle, Standby Sleep) where its source clock is running. The interrupts can wake up the device from sleep modes. Events connected to the event system can trigger other operations in the system without exiting sleep modes.

40.5.3. Clocks

The TC bus clocks (CLK_TCx_APB) can be enabled and disabled in the Clock and Reset Unit (CRU).

The generic clocks (GCLK_TCx) are asynchronous to the user interface clock (CLK_TCx_APB). Due to this asynchronicity, accessing certain registers will require synchronization between the clock domains.

Note: Two instances of the TC can share a peripheral clock channel. In this case, they cannot be set to different clock frequencies. See *Clock and Reset Unit (CRU)* from Related Links.

Related Links

Clock and Reset Unit (CRU)

40.5.4. DMA

The DMA request lines are connected to the DMA Controller (DMAC). In order to use DMA requests with this peripheral, the DMAC must be configured first (see *Direct Memory Access Controller (DMAC)* from Related Links).

Related Links

Direct Memory Access Controller (DMAC)

40.5.5. Interrupts

The interrupt request line is connected to the Interrupt Controller. In order to use interrupt requests of this peripheral, the Interrupt Controller (NVIC) must be configured first. See *Nested Vector Interrupt Controller (NVIC)* from Related Links.

Related Links

Nested Vector Interrupt Controller (NVIC)

Nested Vector Interrupt Controller (NVIC)

40.5.6. Events

The events of this peripheral are connected to the Event System.

Related Links

Event System (EVSYS)

40.5.7. Debug Operation

When the CPU is halted in Debug mode, this peripheral will halt normal operation. This peripheral can be forced to continue operation during debugging. For more details, see *DBGCTRL* from Related Links.

© 2025 Microchip Technology Inc. and its subsidiaries

Related Links

DBGCTRL

Debug Control

40.5.8. Register Access Protection

Registers with write access can be optionally write protected by the PAC, except for the following:

- Interrupt Flag Status and Clear register (INTFLAG)
- Status register (STATUS)
- Period and Period Buffer registers (PER, PERBUF)
- Compare/Capture Value registers and Compare/Capture Value Buffer registers (CCx, CCBUFx)

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

Write protection does not apply for accesses through an external debugger.

40.5.9. Analog Connections

Not applicable.

40.6. Functional Description

40.6.1. Principle of Operation

The following definitions are used throughout the documentation:

Table 40-2. Timer/Counter Definitions

Name	Description
TOP	The counter reaches TOP when it becomes equal to the highest value in the count sequence. The TOP value can be the same as Period (PER) or the Compare Channel 0 (CC0) register value depending on the waveform generator mode in Waveform Output Operations. See *Waveform Output Operations* from Related Links.
ZERO	The counter is ZERO when it contains all zeros.
MAX	The counter reaches MAX when it contains all ones.
UPDATE	The timer/counter signals an update when it reaches ZERO or TOP, depending on the direction settings.
Timer	The timer/counter clock control is handled by an internal source.
Counter	The clock control is handled externally (e.g., counting external events).
CC	For compare operations, the CC are referred to as *compare channels*. For capture operations, the CC are referred to as *capture channels*.

Each TC instance has up to two compare/capture channels (CC0 and CC1).

The counter in the TC can either count events from the Event System or clock ticks of the GCLK_TCx clock, which may be divided by the prescaler.

The counter value is passed to the CCx where it can be either compared to user-defined values or captured.

For optimized timing, the CCx and CCBUFx registers share a common resource. When writing into CCBUFx, lock the access to the corresponding CCx register (SYNCBUSY.CCX = `1`) until the CCBUFx register value is not loaded into the CCx register (BUFVx ==`1`). Each buffer register has a buffer valid (BUFV) flag that indicates when the buffer contains a new value.

The Counter register (COUNT) and the Compare and Capture registers with buffers (CCx and CCBUFx) can be configured as 8-, 16- or 32-bit registers with corresponding MAX values. Mode settings (CTRLA.MODE) determine the maximum range of the Counter register.

© 2025 Microchip Technology Inc. and its subsidiaries

In 8-bit mode, a Period Value (PER) register and its Period Buffer Value (PERBUF) register are also available. The counter range and the operating frequency determine the maximum time resolution achievable with the TC peripheral.

The TC can be set to count up or down. Under normal operation, the counter value is continuously compared to the TOP or ZERO value to determine whether the counter has reached that value. On a comparison match, the TC can request DMA transactions or generate interrupts or events for the Event System.

In a compare operation, the counter value is continuously compared to the values in the CCx registers. In the case of a match, the TC can request DMA transactions or generate interrupts or events for the Event System. In waveform generator mode, these comparisons are used to set the waveform period or pulse width.

Capture operation can be enabled to perform input signal period and pulse width measurements or to capture selectable edges from an I/O pin or internal event from Event System.

Related Links

Waveform Output Operations

40.6.2. Basic Operation

40.6.2.1. Initialization

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

- Control A register (CTRLA), except the Enable (ENABLE) and Software Reset (SWRST) bits
- Drive Control register (DRVCTRL)
- Wave register (WAVE)
- Event Control register (EVCTRL)

Writing to enable-protected bits and setting the CTRLA.ENABLE bit can be performed in a single 32-bit access of the CTRLA register. Writing to enable-protected bits and clearing the CTRLA.ENABLE bit cannot be performed in a single 32-bit access.

Before enabling the TC, the peripheral must be configured by the following steps:

1. Enable the TC bus clock if not already enabled by default (PB1_CLK).
2. Select 8-, 16- or 32-bit counter mode via the TC Mode bit group in the Control A register (CTRLA.MODE). The default mode is 16-bit.
3. Select one wave generation operation in the Waveform Generation Operation bit group in the WAVE register (WAVE.WAVEGEN).
4. If desired, the GCLK_TCx clock can be prescaled via the Prescaler bit group in the Control A register (CTRLA.PRESCALER).
   - If the prescaler is used, select a prescaler synchronization operation via the Prescaler and Counter Synchronization bit group in the Control A register (CTRLA.PRESYNC).
5. If desired, select one-shot operation by writing a '1' to the One-Shot bit in the Control B Set register (CTRLBSET.ONESHOT).
6. If desired, configure the counting direction down (starting from the TOP value) by writing a '1' to the Counter Direction bit in the Control B register (CTRLBSET.DIR).
7. For capture operation, enable the individual channels to capture in the Capture Channel x Enable bit group in the Control A register (CTRLA.CAPTEN).
8. If desired, enable inversion of the waveform output or I/O pin input signal for individual channels via the Invert Enable bit group in the Drive Control register (DRVCTRL.INVEN).

© 2025 Microchip Technology Inc. and its subsidiaries

40.6.2.2. 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 '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the TC, except DBGCTRL, will be reset to their initial state. See *CTRLA* from Related Links.

The TC must be disabled before the TC is reset to avoid undefined behavior.

Related Links

CTRLA

Control A

40.6.2.3. 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.

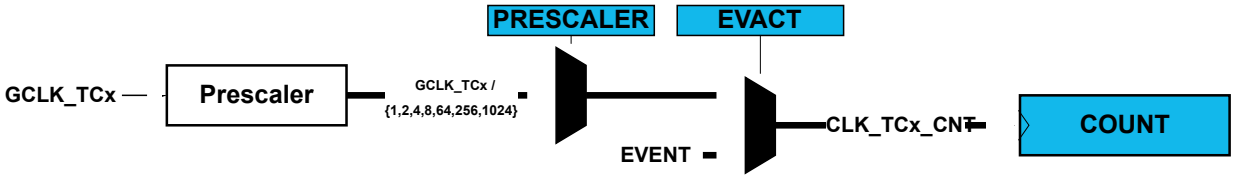
If the prescaler value is higher than one, the Counter Update condition can be optionally executed on the next GCLK_TCx clock pulse or the next prescaled clock pulse. For further details, refer to Prescaler (CTRLA.PRESCALER) and Counter Synchronization (CTRLA.PRESYNC) description.

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_TCx_CNT.

Figure 40-2. Prescaler



40.6.2.4. Counter Mode

The Counter mode is selected by the Mode bit group in the Control A register (CTRLA.MODE). By default, the counter is enabled in the 16-bit counter resolution. Three counter resolutions are available:

- COUNT8 – The 8-bit TC has its own Period Value and Period Buffer Value registers (PER and PERBUF).
- COUNT16 – 16-bit is the default Counter mode. There is no dedicated period register in this mode.
- COUNT32 – 32-bit mode is achieved by pairing two 16-bit TC peripherals.
  When paired, the TC peripherals are configured using the registers of the even-numbered TC (TC0, TC2, TC4, TC6 or TC8 respectively).

  The TC bus clocks (CLK_TCx_APB) for both host and client TCs need to be enabled.

  The odd-numbered partner (TC1, TC3, TC5, TC7 or TC9, respectively) acts as a client, and the Client bit in the Status register (STATUS.SLAVE) is set. The register values of a client will not reflect the registers of the 32-bit counter. Writing to any of the slave registers will not affect the 32-bit counter. Normal access to the slave COUNT and CCx registers is not allowed.

© 2025 Microchip Technology Inc. and its subsidiaries

40.6.2.5. Counter Operations

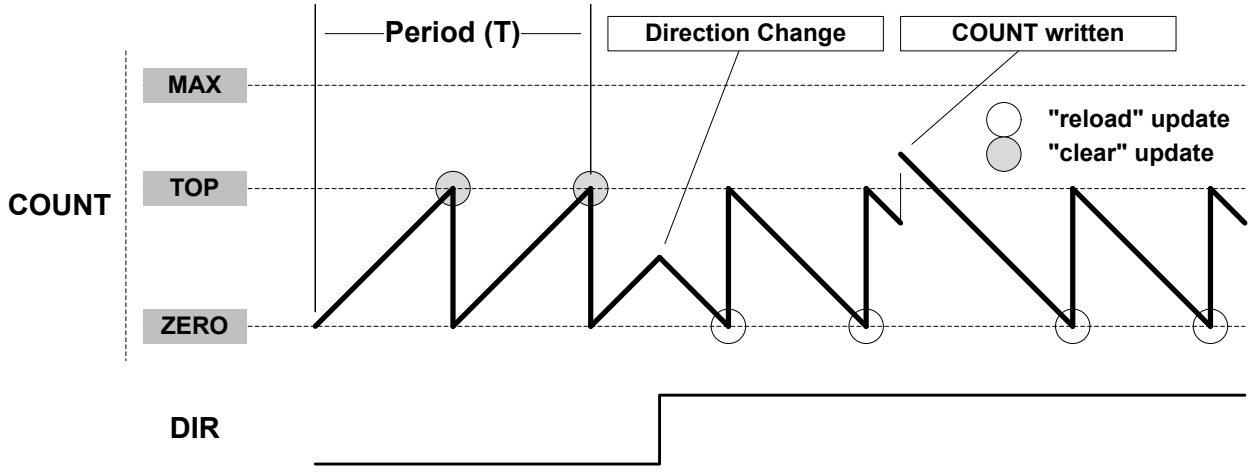
Depending on the mode of operation, the counter is cleared, reloaded, incremented, or decremented at each TC clock input (CLK_TCx_CNT). A counter clear or reload marks the end of the current counter cycle and the start of a new one.

The counting direction is set by the Direction bit in the Control B register (CTRLB.DIR). If this bit is zero the counter is counting up, and counting down if CTRLB.DIR=1. The counter will count up or down for each tick (clock or event) until it reaches TOP or ZERO. When it is counting up and TOP is reached, the counter will be set to zero at the next tick (overflow) and the Overflow Interrupt Flag in the Interrupt Flag Status and Clear register (INTFLAG.OVF) will be set. When it is counting down, the counter is reloaded with the TOP value when ZERO is reached (underflow), and INTFLAG.OVF is set.

INTFLAG.OVF can be used to trigger an interrupt, a DMA request, or an event. An overflow/underflow occurrence (i.e., a compare match with TOP/ZERO) will stop counting if the One-Shot bit in the Control B register is set (CTRLBSET.ONESHOT).

It is possible to change the counter value (by writing directly in the COUNT register) even when the counter is running. When starting the TC, the COUNT value will be either ZERO or TOP (depending on the counting direction set by CTRLBSET.DIR or CTRLBCLR.DIR), unless a different value has been written to it, or the TC has been stopped at a value other than ZERO. The write access has higher priority than count, clear, or reload. The direction of the counter can also be changed when the counter is running. See also the following figure.

Figure 40-3. Counter Operation



Due to asynchronous clock domains, the internal counter settings are written when the synchronization is complete. Normal operation must be used when using the counter as timer base for the capture channels.

40.6.2.5.1. Stop Command and Event Action

A Stop command can be issued from software by using Command bits in the Control B Set register (CTRLBSET.CMD = 0x2, STOP). When a Stop is detected while the counter is running, the counter is set to bottom/top depending on direction: up/down. All waveforms are cleared, and the Stop bit in the Status register is set (STATUS.STOP).

40.6.2.5.2. Re-Trigger Command and Event Action

A re-trigger command can be issued from software by writing the Command bits in the Control B Set register (CTRLBSET.CMD = 0x1, RETRIGGER) or from an event when a re-trigger event action is configured in the Event Control register (EVCTRL.EVACT = 0x1, RETRIGGER).

When the command is detected during the counting operation, the counter will be reloaded or cleared, depending on the counting direction (CTRLBSET.DIR or CTRLBCLR.DIR). When the re-trigger

© 2025 Microchip Technology Inc. and its subsidiaries

command is detected while the counter is stopped, the counter will resume counting from the current value in the COUNT register.

Note: When a re-trigger event action is configured in the Event Action bits in the Event Control register (EVCTRL.EVACT=0x1, RETRIGGER), enabling the counter will not start the counter. The counter will start on the next incoming event and restart on the corresponding following event.

40.6.2.5.3. Count Event Action

The TC can count events. When an event is received, the counter increases or decreases the value, depending on direction settings (CTRLBSET.DIR or CTRLBCLR.DIR). The count event action can be selected by the Event Action bit group in the Event Control register (EVCTRL.EVACT=0x2, COUNT).

Note: If this operation mode is selected, PWM generation is not supported.

40.6.2.5.4. Start Event Action

The TC can start counting operation on an event when previously stopped. In this configuration, the event has no effect if the counter is already counting. When the peripheral is enabled, the counter operation starts when the event is received or when a re-trigger software command is applied.

The Start TC on Event action can be selected by the Event Action bit group in the Event Control register (EVCTRL.EVACT=0x3, START).

40.6.2.6. Compare Operations

By default, the Compare/Capture channel is configured for compare operations.

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

The Channel x Compare Buffer (CCBUFx) registers provide double buffer capability. The double buffering synchronizes the update of the CCx register with the buffer value at the UPDATE condition or a forced update command (CTRLBSET.CMD=UPDATE). See *Double Buffering* from Related Links. The synchronization prevents the occurrence of odd-length, non-symmetrical pulses and ensures glitch-free output.

Related Links

Double Buffering

40.6.2.6.1. Waveform Output Operations

The compare channels can be used for waveform generation on output port pins. To make the waveform available on the connected pin, the following requirements must be fulfilled:

1. Choose a Waveform Generation mode in the Waveform Generation Operation bit in Waveform register (WAVE.WAVEGEN).
2. Optionally, invert the waveform output WO[x] by writing the corresponding Output Waveform x Invert Enable bit in the Driver Control register (DRVCTRL.INVENx).
3. Configure the pins with the I/O Peripheral Pin Select (PPS). See *I/O Ports and Peripheral Pin Select (PPS)* from Related Links.
   Note: The event must not be used when the compare channel is set in the waveform output operating mode.

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) is set on the next zero-to-one transition of CLK_TC_CNT (see Normal Frequency Operation). An interrupt and/or event can be generated on a comparison match if enabled. The same condition generates a DMA request.

There are four waveform configurations for the Waveform Generation Operation bit group in the Waveform register (WAVE.WAVEGEN). This influences how the waveform is generated and imposes restrictions on the top value. The configurations are:

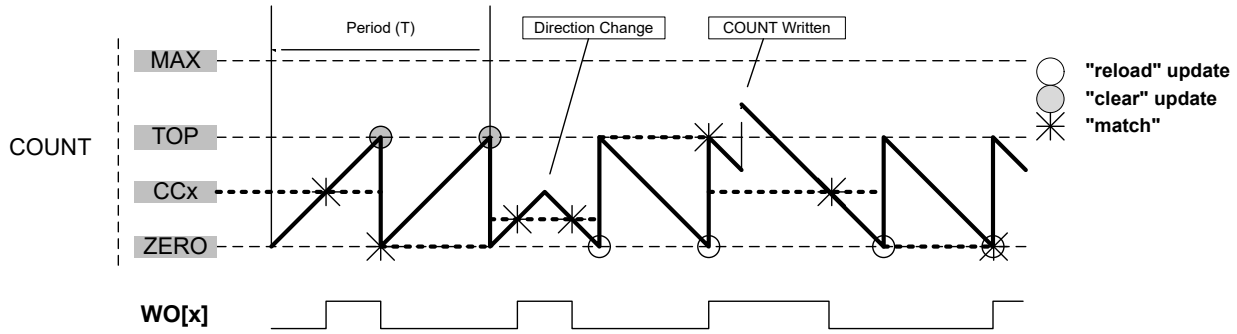
© 2025 Microchip Technology Inc. and its subsidiaries

- Normal frequency (NFRQ)
- Match frequency (MFRQ)
- Normal pulse-width modulation (NPWM)
- Match pulse-width modulation (MPWM)

When using NPWM or NFRQ configuration, TOP is determined by the Period register (PER). In the 8-bit Counter mode, the Period register (PER) is used as TOP, and the TOP can be changed by writing to the PER register. In the 16- and 32-Bit Counter modes, TOP is fixed to the maximum (MAX) value of the counter.

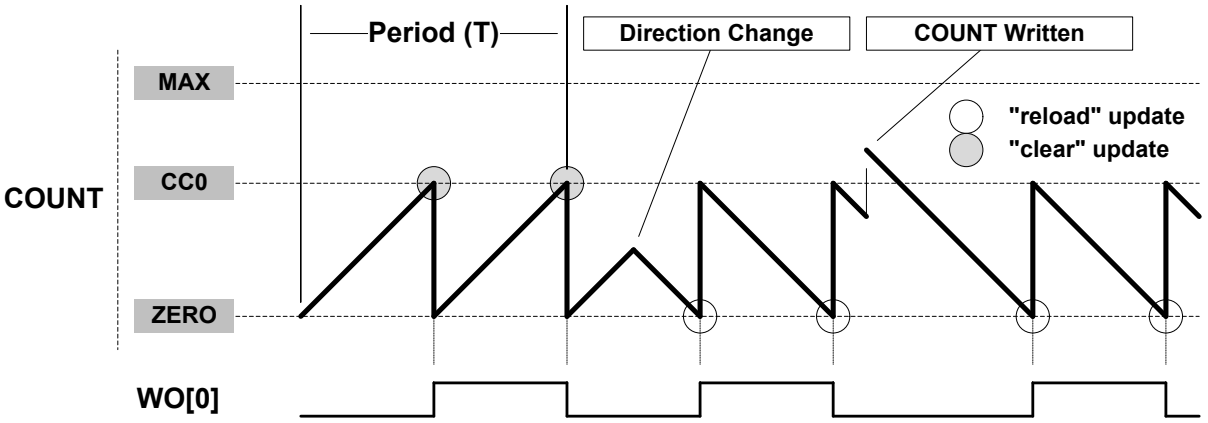
Normal Frequency Generation (NFRQ)
For Normal Frequency Generation, the period time (T) is controlled by the period register (PER) for the 8-bit Counter mode and MAX for the 16- and 32-Bit mode. The waveform generation output (WO[x]) is toggled on each compare match between COUNT and CCx, and the corresponding Match or Capture Channel x Interrupt Flag (INTFLAG.MCx) is set.

Figure 40-4. Normal Frequency Operation



Match Frequency Generation (MFRQ)
For Match Frequency Generation, the period time (T) is controlled by the CC0 register instead of PER or MAX. WO[0] toggles on each Update condition.

Figure 40-5. Match Frequency Operation



Normal Pulse-Width Modulation Operation (NPWM)
NPWM uses single-slope PWM generation.

For single-slope PWM generation, the period time (T) is controlled by the TOP value and CCx controls the duty cycle of the generated waveform output. When up-counting, the WO[x] is set at start or compare match between the COUNT and TOP values and cleared on compare match between

© 2025 Microchip Technology Inc. and its subsidiaries

COUNT and CCx register values. When down-counting, the WO[x] is cleared at the start or compare match between the COUNT and ZERO values and set on the compare match between the COUNT and CCx register values.

The following equation calculates the exact resolution for a single-slope PWM (R_{PWM_SS}) waveform:

$$R_{PWM\_SS} = \frac{\log(TOP+1)}{\log(2)}$$

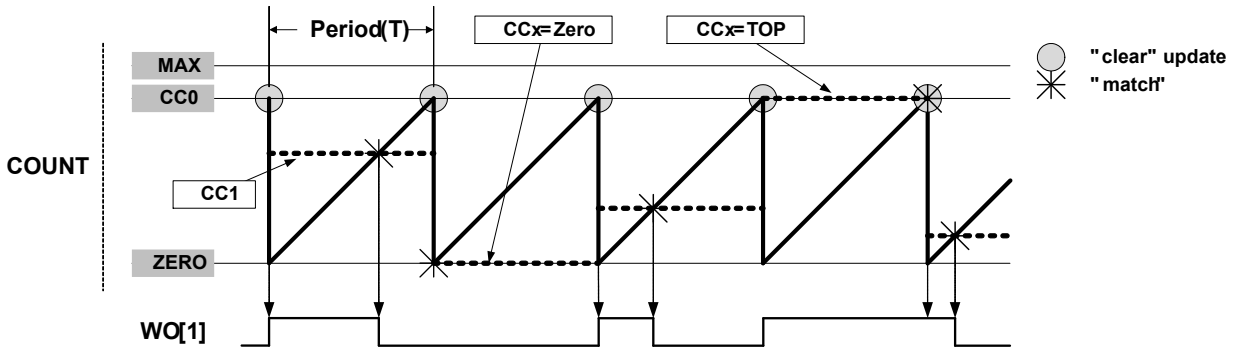
The PWM frequency (f_{PWM_SS}) depends on the TOP value and the peripheral clock frequency (f_{GCLK_TC}) and can be calculated by the following equation:

$$f_{PWM\_SS} = \frac{f_{GCLK\_TC}}{N(TOP+1)}$$

Where N represents the prescaler divider used (1, 2, 4, 8, 16, 64, 256, 1024).

Match Pulse-Width Modulation Operation (MPWM)
In MPWM, the output of WO[1] is depending on CC1 as illustrated in the following figure. On every overflow/underflow, a one-TC-clock-cycle negative pulse is put out on WO[0] (not shown in the figure).

Figure 40-6. Match PWM Operation



The following table lists the Update Counter and Overflow Event/Interrupt Generation conditions in different operation modes.

Table 40-3. Counter Update and Overflow Event/interrupt Conditions in TC

Name	Operation	TOP	Update	Output Waveform		OVFIF/Event	
				On Match	On Update	Up	Down
NFRQ	Normal Frequency	PER	TOP/ ZERO	Toggle	Stable	TOP	ZERO
MFRQ	Match Frequency	CC0	TOP/ ZERO	Toggle	Stable	TOP	ZERO
NPWM	Single-slope PWM	PER	TOP/ ZERO	See description above.		TOP	ZERO
MPWM	Single-slope PWM	CC0	TOP/ ZERO	Toggle	Toggle	TOP	ZERO

Related Links
I/O Ports and Peripheral Pin Select (PPS)

40.6.2.7. Double Buffering

The Compare Channels (CCx) registers and the Period (PER) register in 8-bit mode are double buffered. Each buffer register has a buffer valid bit (CCBUFVx or PERBUFV) in the STATUS register, which indicates that the buffer register contains a new valid value that can be copied into the corresponding register. As long as the respective buffer valid status flag (PERBUFV or CCBUFVx) are set to '1', related syncbusy bits are set (SYNCBUSY.PER or SYNCBUSY.CCx), a write to the respective

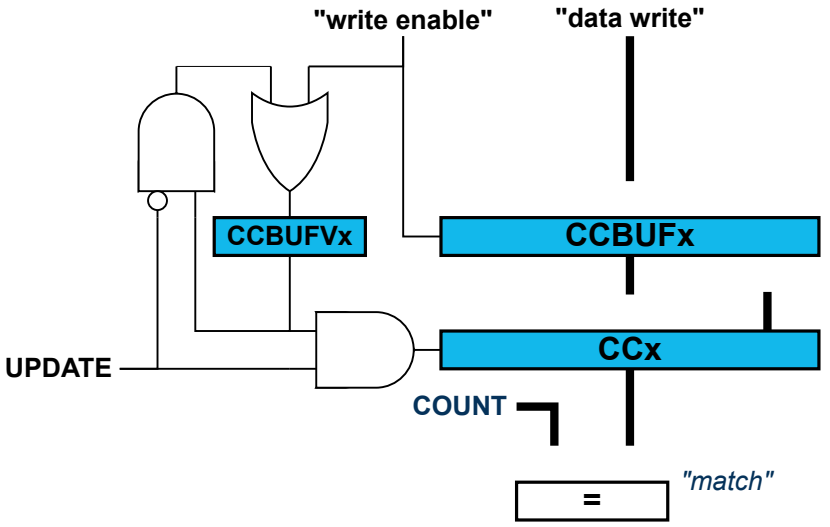
© 2025 Microchip Technology Inc. and its subsidiaries

PER/PERBUF or CCx/CCBUFx registers will generate a PAC error and access to the respective PER or CCx register is invalid.

When the buffer valid flag bit in the STATUS register is '1' and the Lock Update bit in the CTRLB register is set to '0' (writing CTRLBCLR.LUPD to '1'), double buffering is enabled: the data from buffer registers will be copied into the corresponding register under hardware UPDATE conditions, then, the buffer valid flags bit in the STATUS register are automatically cleared by hardware.
Note: The software update command (CTRLBSET.CMD=0x3) is acting independently of the LUPD value.

A compare register is double buffered as in the following figure.

Figure 40-7. Compare Channel Double Buffering



Both the registers (PER/CCx) and corresponding buffer registers (PERBUF/CCBUFx) are available in the I/O register map, and the double buffering feature is not mandatory. The double buffering is disabled by writing a '1' to CTRLBSET.LUPD.

Note: In NFRQ, MFRQ or PWM down-counting counter mode (CTRLBSET.DIR=1) when double buffering is enabled (CTRLBCLR.LUPD=1), the PERBUF register is continuously copied into the PER independently of update conditions.

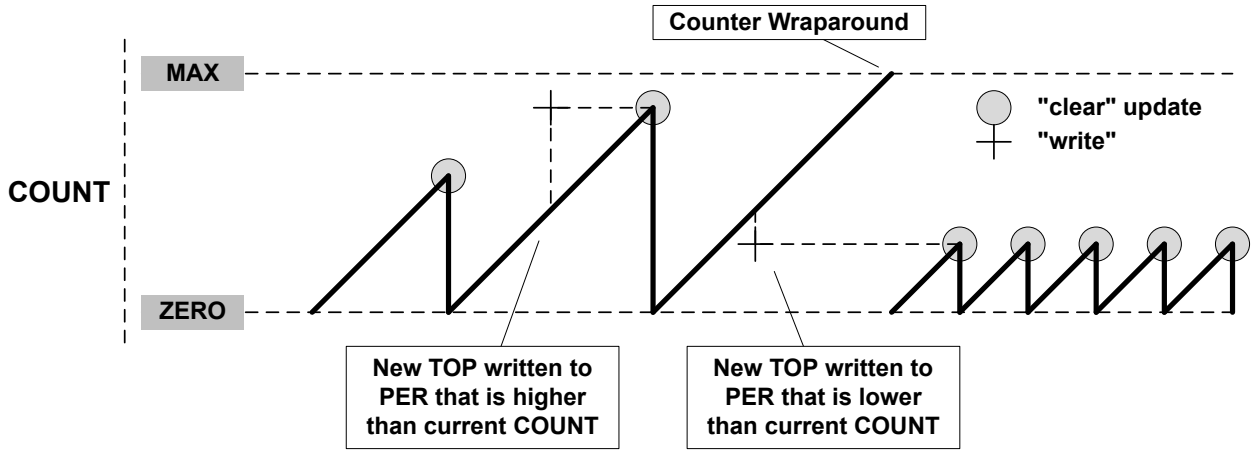
Changing the Period

The counter period can be changed by writing a new TOP value to the Period register (PER or CC0, depending on the waveform generation mode), which is available in 8-bit mode. Any period update on registers (PER or CCx) is effective after the synchronization delay.

A counter wraparound can occur in any operation mode when up-counting without buffering (see the following figure).

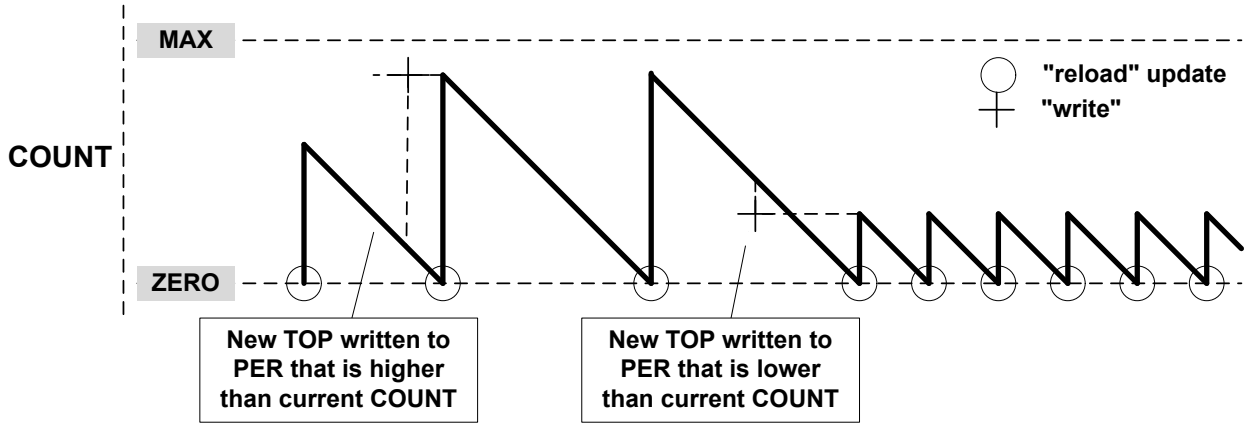
© 2025 Microchip Technology Inc. and its subsidiaries

Figure 40-8. Unbuffered Single-Slope Up-Counting Operation



COUNT and TOP are continuously compared, so when a new TOP value that is lower than the current COUNT is written to TOP, COUNT will wrap before a compare match.

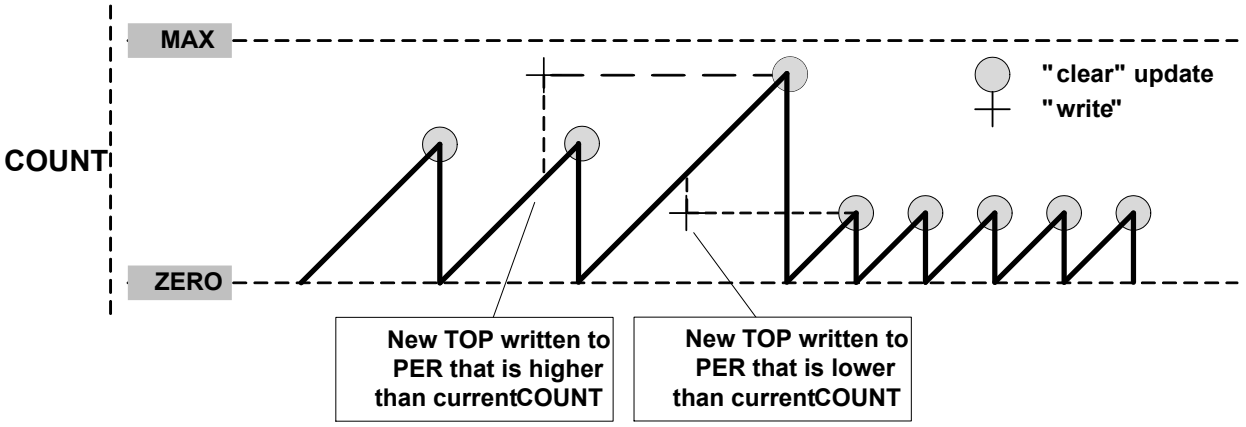
Figure 40-9. Unbuffered Single-Slope Down-Counting Operation



When double buffering is used, the buffer can be written at any time and the counter will still maintain the correct operation. The period register is always updated on the update condition, as shown in the following figure. This prevents wraparound and the generation of odd waveforms.

© 2025 Microchip Technology Inc. and its subsidiaries

Figure 40-10. Changing the Period Using Buffering



40.6.2.8. Capture Operations

To enable and use capture operations, the corresponding Capture Channel x Enable bit in the Control A register (CTRLA.CAPTENx) must be written to '1'.

A capture trigger can be provided by the input event line TC_EV or by the asynchronous I/O pin WO[x] for each capture channel or by a TC event. To enable the capture from the input event line, the Event Input Enable bit in the Event Control register (EVCTRL.TCEI) must be writtento '1'. To enable the capture from the I/O pin, the Capture On Pin x Enable bit in the CTRLA register (CTRLA.COPENx) must be written to '1'.

Notes:

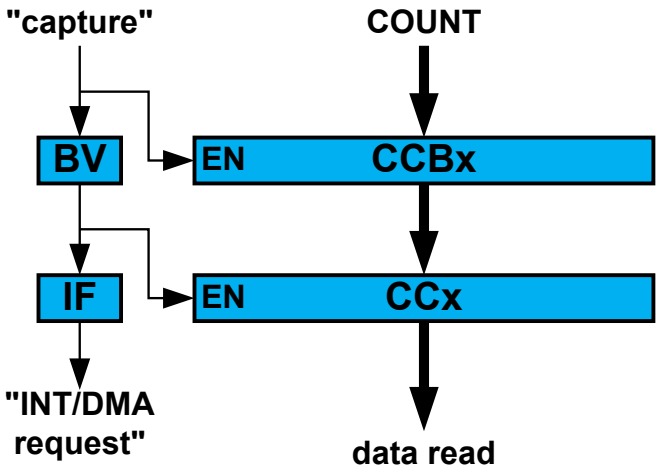
1. Capture on I/Os is only possible in the Event and Time-Stamp capture action modes. Other modes can only use internal events. (If I/O toggling is needed in other modes, the I/Os edge must be configured for generating internal events).
2. Capture on an event from the Event System is possible in Event, PPW/PWP/PW and Time-Stamp capture action modes. In this case, the event system channels must be configured to operate in an asynchronous mode of operation.
3. Depending on CTRLA.COPENx, channel x can be configured for I/Os or internal event capture (both are mutually exclusive). One channel can be configured for I/O capture, while the other uses internal event capture.

By default, a capture operation is done when a rising edge is detected on the input signal. Capture on the falling edge is available, its activation is dependent on the input source:

- When the channel is used with an I/O pin, write a '1' to the corresponding Invert Enable bit in the Drive Control register (DRVCTRL.INVENx).
- When the channel is counting events from the Event System, write a '1' to the TC Event Input Invert Enable bit in Event Control register (EVCTRL.TCINV).

© 2025 Microchip Technology Inc. and its subsidiaries

Figure 40-11. Capture Double Buffering



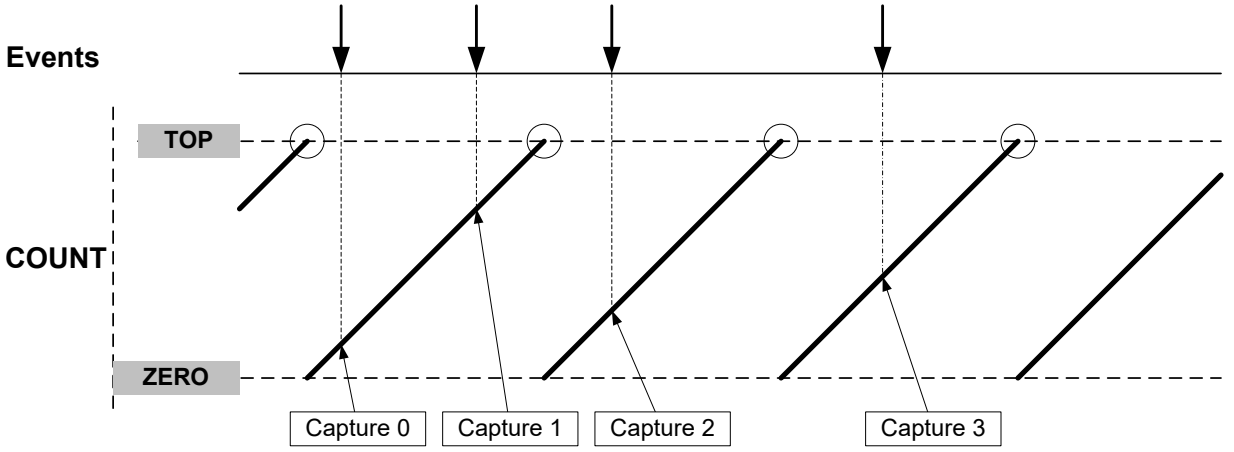
For input capture, the buffer register and the corresponding CCx act like a FIFO. When CCx is empty or read, any content in CCBUFx is transferred to CCx. The buffer valid flag is passed to set the CCx interrupt flag (IF) and generate the optional interrupt, event or DMA request. The CCBUFx register value cannot be read; all captured data must be read from the CCx register.

Note:
When up-counting (CTRLBSET.DIR = 0), counter values lower than '1' cannot be captured. To capture the full range including value '0', the TC must be in the Down-counting mode (CTRLBSET.DIR = 0).

40.6.2.8.1. Event Capture Action on Events or I/Os

The compare and capture channels can be used as input capture channels to capture events from the Event System or I/O pins and give them a timestamp. This mode is selected when EVTCTRL.EVACT is configured either as OFF, RETRIGGER, COUNT or START. The following figure shows four capture events for one capture channel.

Figure 40-12. Input Capture Timing



The TC can detect capture overflow of the input capture channels: When a new capture event is detected while the Capture Interrupt flag (INTFLAG.MCx) is still set, the new timestamp will not be stored and INTFLAG.ERR will be set.

© 2025 Microchip Technology Inc. and its subsidiaries

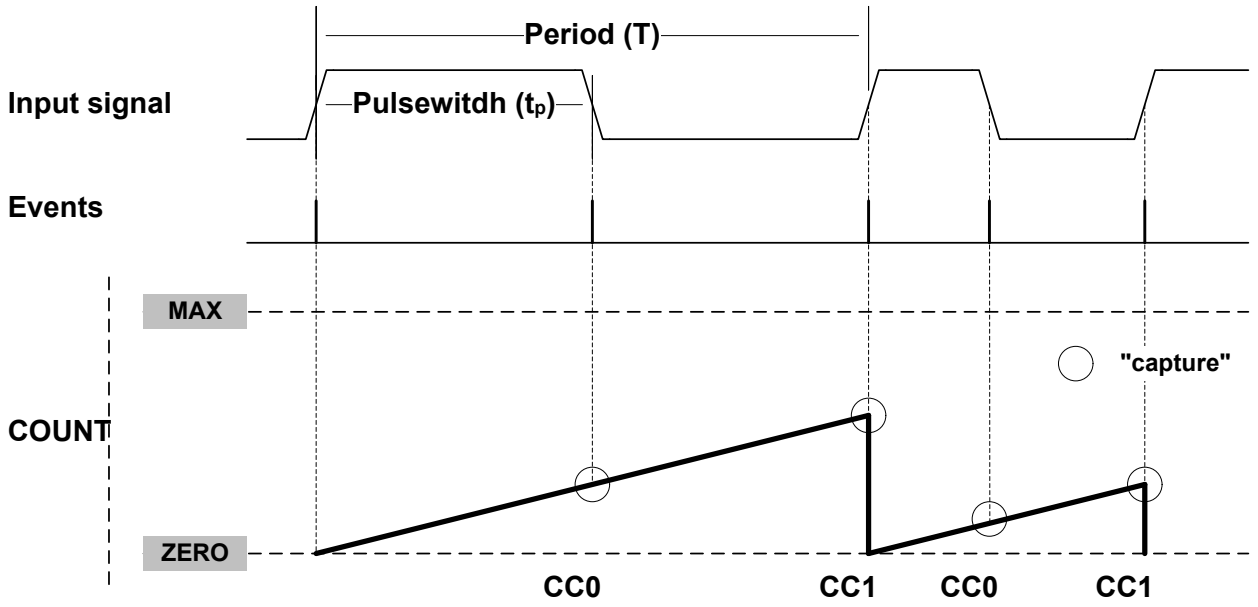
40.6.2.8.2. Period and Pulse-Width (PPW/PWP) Capture Action on Events

The TC can perform two input captures and restart the counter on one of the edges. This enables the TC to measure the pulse width and period and to characterize the frequency *f* and duty cycle of an input signal:

$$f = \frac{1}{T}$$

$$\text{dutyCycle} = \frac{t_p}{T}$$

Figure 40-13. PWP Capture



Selecting PWP in the Event Action bit group in the Event Control register (EVCTRL.EVACT) enables the TC to perform one capture action on the rising edge and another one on the falling edge. The period *T* will be captured into CC1 and the pulse width t_p in CC0. EVCTRL.EVACT = PPW (period and pulse-width) offers identical functionality, but will capture *T* into CC0 and t_p into CC1.

The TC Event Input Invert Enable bit in the Event Control register (EVCTRL.TCINV) is used to select whether the wraparound must occur on the rising edge or the falling edge. If EVCTRL.TCINV = 1, the wraparound will happen on the falling edge.

The TC can detect capture overflow of the input capture channels: When a new capture event is detected while the Capture Interrupt flag (INTFLAG.MCx) is still set, the new timestamp will not be stored and INTFLAG.ERR will be set.

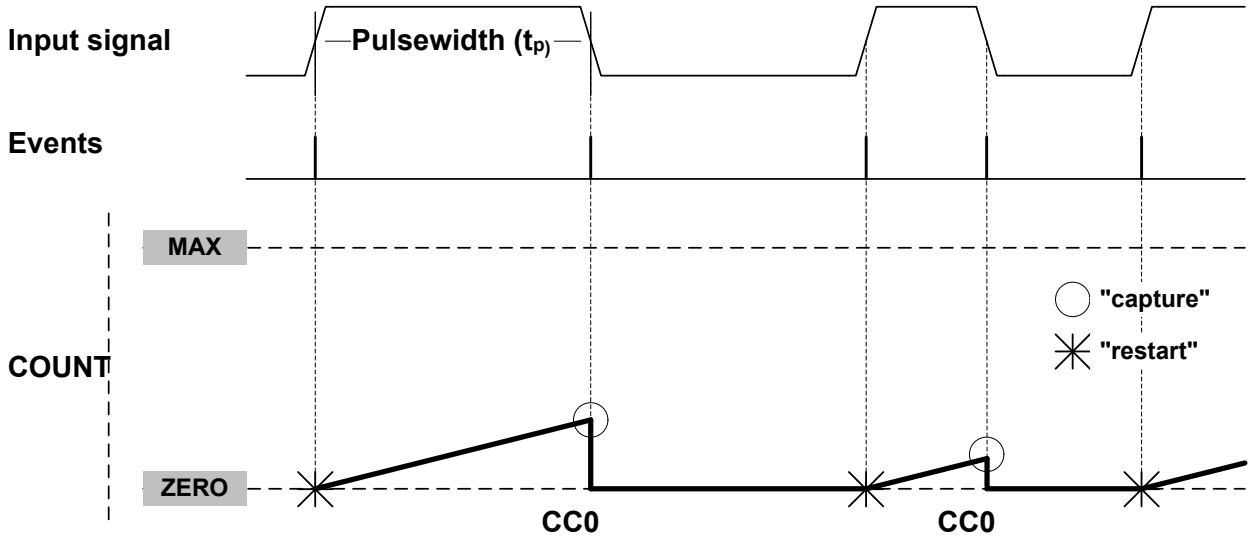
Note: The corresponding capture is working only if the channel is enabled in capture mode (CTRLA.CAPTENx = 1). Consequently, both channels must be enabled to fully characterize the input.

40.6.2.8.3. Pulse-Width (PW) Capture Action on Events

The TC performs the input capture on the falling edge of the input signal. When the edge is detected, the counter value is cleared and the TC stops counting. When a rising edge is detected on the input signal, the counter restarts the counting operation. To enable the operation on opposite edges, the input signal to capture must be inverted (refer to EVCTRL.TCINV).

© 2025 Microchip Technology Inc. and its subsidiaries

Figure 40-14. Pulse-Width Capture on Channel 0



The TC can detect capture overflow of the input capture channels: When a new capture event is detected while the Capture Interrupt flag (INTFLAG.MCx) is still set, the new timestamp will not be stored and INTFLAG.ERR will be set.

40.6.3. Additional Features

40.6.3.1. One-Shot Operation

When one-shot is enabled, the counter automatically stops on the next counter overflow or underflow condition. When the counter is stopped, the Stop bit in the Status register (STATUS.STOP) is automatically set and the waveform outputs are set to '0'.

One-shot operation is enabled by writing a '1' to the One-Shot bit in the Control B Set register (CTRLBSET.ONESHOT) and disabled by writing a '1' to CTRLBCLR.ONESHOT. When enabled, the TC counts until an overflow or underflow occurs and stops the counting operation. The one-shot operation can be restarted by a re-trigger software command, a re-trigger event or a start event. When the counter restarts its operation, STATUS.STOP is automatically cleared.

40.6.3.2. Time-Stamp Capture on Events or I/Os

This feature is enabled when the Capture Time Stamp (STAMP) Event Action in Event Control register (EVCTRL.EVACT) is selected. The counter TOP value must be smaller than MAX.

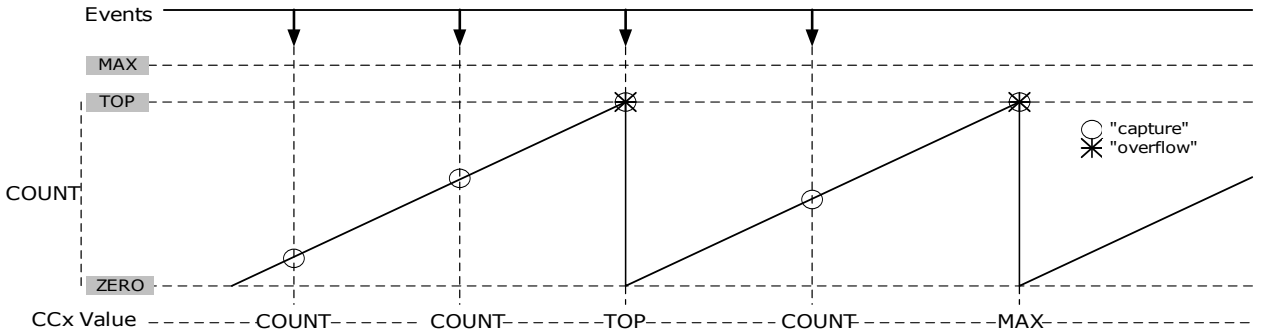
When a capture event from the Event System or the I/O pin is detected, the COUNT value is copied into the corresponding Channel x Compare/Capture Value (CCx) register. In case of an overflow, the MAX value is copied into the corresponding CCx register.

When a valid captured value is present in the capture channel register, the corresponding Capture Channel x Interrupt Flag (INTFLAG.MCx) is set.

The timer/counter can detect capture overflow of the input capture channels: When a new capture event is detected while the Capture Channel interrupt flag (INTFLAG.MCx) is still set, the new time-stamp will not be stored and INTFLAG.ERR will be set.

© 2025 Microchip Technology Inc. and its subsidiaries

Figure 40-15. Time Stamp



40.6.3.3. Minimum Capture

The minimum capture is enabled by writing the CAPTMIN mode in the Channel n Capture Mode bits in the Control A register (CTRLA.CAPTMODEn = CAPTMIN).

CCx Content:

In CAPTMIN operations, CCx keeps the minimum captured values. Before enabling this mode of capture, the user must initialize the corresponding CCx register value to a value different from zero. If the CCx register initial value is zero, no capture is performed using the corresponding channel.

MCx Behaviour:

In the CAPTMIN operation, the capture is performed only if, when on capture event time, the counter value is lower than the last captured value. The MCx interrupt flag is set only if, when on capture event time, the counter value is higher or equal to the value captured on the previous event. So the interrupt flag is set when a new absolute local minimum value is detected.

40.6.3.4. Maximum Capture

The maximum capture is enabled by writing the CAPTMAX mode in the Channel n Capture Mode bits in the Control A register (CTRLA.CAPTMODEn = CAPTMAX).

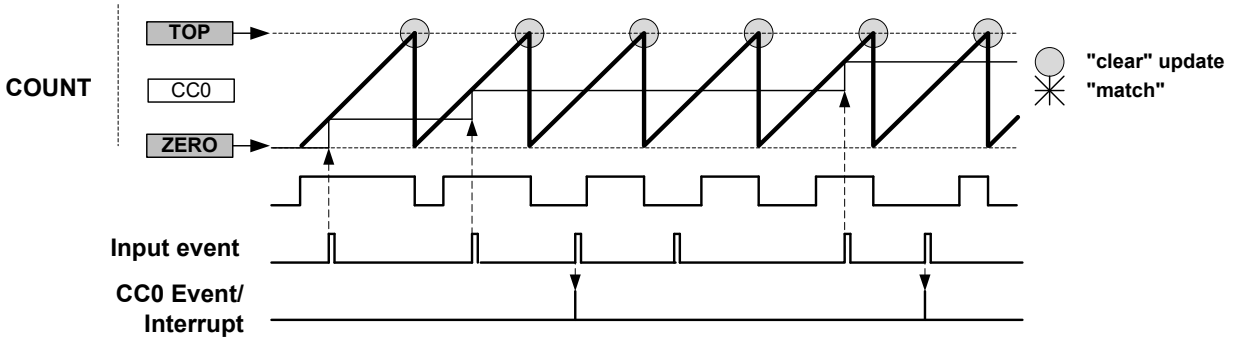
CCx Content:

In CAPTMAX operations, CCx keeps the maximum captured values. Before enabling this mode of capture, the user must initialize the corresponding CCx register value to a value different from TOP. If the CCx register initial value is TOP, no capture is performed using the corresponding channel.

MCx Behaviour:

In the CAPTMAX operation, the capture is performed only if, when on capture event time, the counter value is higher than the last captured value. The MCx Interrupt flag is set only if, when on capture event time, the counter value is lower or equal to the value captured on the previous event. So the Interrupt flag is set when a new absolute local maximum value is detected.

Figure 40-16. Maximum Capture Operation with CC0 Initialized with ZERO Value



© 2025 Microchip Technology Inc. and its subsidiaries

40.6.4. DMA Operation

The TC can generate the following DMA requests:

- Overflow (OVF): the request is set when an update condition (overflow, underflow or re-trigger) is detected, the request is cleared by hardware on DMA acknowledge.
- Match or Capture Channel x (MCx): for a compare channel, the request is set on each compare match detection, the request is cleared by hardware on DMA acknowledge. For a capture channel, the request is set when valid data is present in the CCx register, and cleared when CCx register is read.

40.6.5. Interrupts

The TC has the following interrupt sources:

- Overflow/Underflow (OVF)
- Match or Capture Channel x (MCx)
- Capture Overflow Error (ERR)

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) is set when the interrupt condition occurs.

Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR). The status of enabled interrupts can be read from either INTENSET or INTENCLR.

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until either the interrupt flag is cleared, the interrupt is disabled, or the TC is reset. See *INTFLAG* from Related Links for more details on how to clear the interrupt flags.

The TC has one common interrupt request line for all the interrupt sources. The user must read the INTFLAG register to determine which interrupt condition is present.

Note that interrupts must be globally enabled for interrupt requests to be generated. See *Nested Vector Interrupt Controller (NVIC)* from Related Links.

Related Links

Nested Vector Interrupt Controller (NVIC)

INTFLAG

Interrupt Flag Status and Clear

40.6.6. Events

The TC can generate the following output events:

- Overflow/Underflow (OVF)
- Match or Capture Channel x (MCx)

Writing a '1' to an Event Output bit in the Event Control register (EVCTRL.MCEOx) enables the corresponding output event. The output event is disabled by writing EVCTRL.MCEOx=0.

One of the following event actions can be selected by the Event Action bit group in the Event Control register (EVCTRL.EVACT):

- Disable event action (OFF)
- Start TC (START)
- Re-trigger TC (RETRIGGER)
- Count on event (COUNT)

© 2025 Microchip Technology Inc. and its subsidiaries

- Capture time stamp (STAMP)
- Capture Period (PPW and PWP)
- Capture Pulse Width (PW)

Writing a '1' to the TC Event Input bit in the Event Control register (EVCTRL.TCEI) enables input events to the TC. Writing a '0' to this bit disables input events to the TC. The TC requires only asynchronous event inputs. See *Event System (EVSYS)* from Related Links for additional information on configuring the asynchronous events.

Related Links
Event System (EVSYS)

40.6.7. Sleep Mode Operation

The TC can be configured to operate in any sleep mode (Idle, Standby Sleep). To be able to run in Standby Sleep mode, the RUNSTDBY bit in the Control A register (CTRLA.RUNSTDBY) must be '1'. This peripheral can wake up the device from any sleep mode using interrupts or perform actions through the Event System.

If the On Demand bit in the Control A register (CTRLA.ONDEMAND) is written to '1', the module stops requesting its peripheral clock when the STOP bit in the STATUS register (STATUS.STOP) is set to '1'. When a re-trigger or start condition is detected, the TC requests the clock before the operation starts.

40.6.8. Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset and Enable bits in Control A register (CTRLA.SWRST and CTRLA.ENABLE)
- Capture Channel Buffer Valid bit in STATUS register (STATUS.CCBUFVx)

The following registers are synchronized when written:

- Control B Clear and Control B Set registers (CTRLBCLR and CTRLBSET)
- Count Value register (COUNT)
- Period Value and Period Buffer Value registers (PER and PERBUF)
- Channel x Compare/Capture Value and Channel x Compare/Capture Buffer Value registers (CCx and CCBUFx)

The following registers are synchronized when read:

- Count Value register (COUNT): Synchronization is done on demand through READSYNC command (CTRLBSET.CMD).

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

Required read synchronization is denoted by the Read-Synchronized property in the register description.

© 2025 Microchip Technology Inc. and its subsidiaries

40.7. Register Summary - 8-bit Mode

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0	ONDEMAND	RUNSTDBY	PRESCSYNC[1:0]		MODE[1:0]		ENABLE	SWRST
		15:8	DMAOS				ALOCK	PRESCALER[2:0]		
		23:16			COPEN1	COPEN0			CAPTEN1	CAPTEN0
		31:24				CAPTMODE1[1:0]			CAPTMODE0[1:0]	
0x04	CTRLBCLR	7:0	CMD[2:0]					ONESHOT	LUPD	DIR
0x05	CTRLBSET	7:0	CMD[2:0]					ONESHOT	LUPD	DIR
0x06	EVCTRL	7:0			TCEI	TCINV		EVACT[2:0]		
		15:8			MCEO1	MCEO0				OVFEO
0x08	INTENCLR	7:0			MC1	MC0			ERR	OVF
0x09	INTENSET	7:0			MC1	MC0			ERR	OVF
0x0A	INTFLAG	7:0			MC1	MC0			ERR	OVF
0x0B	STATUS	7:0			CCBUFV1	CCBUFV0	PERBUFV		CLIENT	STOP
0x0C	WAVE	7:0							WAVEGEN[1:0]	
0x0D	DRVCTRL	7:0							INVEN1	INVEN0
0x0E	Reserved									
0x0F	DBGCTRL	7:0								DBGRUN
0x10	SYNCBUSY	7:0	CC1	CC0	PER	COUNT	STATUS	CTRLB	ENABLE	SWRST
0x11 ... 0x13	Reserved									
0x14	COUNT	7:0	COUNT[7:0]							
0x15 ... 0x1A	Reserved									
0x1B	PER	7:0	PER[7:0]							
0x1C	CC0	7:0	CC[7:0]							
0x1D	CC1	7:0	CC[7:0]							
0x1E ... 0x2E	Reserved									
0x2F	PERBUF	7:0	PERBUF[7:0]							
0x30	CCBUF0	7:0	CCBUF[7:0]							
0x31	CCBUF1	7:0	CCBUF[7:0]							

40.8. Register Description - 8-Bit Mode

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

Some registers are optionally write-protected by the PAC. Optional PAC write protection is denoted by the PAC Write-Protection property in each individual register description.

Some registers are synchronized when read and/or written. Synchronization is denoted by the Write-Synchronized or the Read-Synchronized property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable protection is denoted by the Enable-Protected property in each individual register description.

The following are the list of conventions available in the register description:

- R = Readable bit
- W = Writable bit
- U = Unimplemented bit, read as '0'
- -n = Value at POR
- 1 = Bit is set

© 2025 Microchip Technology Inc. and its subsidiaries

- 0 = Bit is cleared
- x = Bit is unknown
- HS = Hardware Set
- HC = Hardware Cleared

© 2025 Microchip Technology Inc. and its subsidiaries

40.8.1. Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized, Enable-Protected

Bit	31	30	29	28	27	26	25	24
				CAPTMODE1[1:0]			CAPTMODE0[1:0]	
Access				R/W	R/W		R/W	R/W
Reset				0	0		0	0

Bit	23	22	21	20	19	18	17	16
			COPEN1	COPEN0			CAPTEN1	CAPTEN0
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit	15	14	13	12	11	10	9	8
	DMAOS				ALOCK	PRESCALER[2:0]		
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY	PRESCSYNC[1:0]		MODE[1:0]		ENABLE	SWRST
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	W
Reset	0	0	0	0	0	0	0	0

Bits 28:27 – CAPTMODE1[1:0] Capture mode Channel 1
These bits select the channel 1 capture mode.

Value	Name	Description
0x0	DEFAULT	Default capture
0x1	CAPTMIN	Minimum capture
0x2	CAPTMAX	Maximum capture
0x3	—	Reserved

Bits 25:24 – CAPTMODE0[1:0] Capture mode Channel 0
These bits select the channel 0 capture mode.

Value	Name	Description
0x0	DEFAULT	Default capture
0x1	CAPTMIN	Minimum capture
0x2	CAPTMAX	Maximum capture
0x3	—	Reserved

Bits 20, 21 – COPENx Capture On Pin x Enable [x=1..0]
Bit x of COPEN[1:0] selects the trigger source for capture operation, either events or I/O pin input. This bit is not synchronized.

Value	Description
0	Event from Event System is selected as trigger source for capture operation on channel x.
1	I/O pin is selected as trigger source for capture operation on channel x.

Bits 16, 17 – CAPTENx Capture Channel x Enable [x=1..0]
Bit x of CAPTEN[1:0] selects whether channel x is a capture or a compare channel.

© 2025 Microchip Technology Inc. and its subsidiaries

These bits are not synchronized.

Value	Description
0	CAPTEN disables capture on channel x.
1	CAPTEN enables capture on channel x.

Bit 15 – DMAOS DMA One-Shot Trigger Mode

This bit enables the DMA One-shot Trigger Mode.
Writing a '1' to this bit generates a DMA trigger on TC cycle following a TC_CTRLBSET_CMD_DMAOS command.
Writing a '0' to this bit generates DMA triggers on each TC cycle.
This bit is not synchronized.

Bit 11 – ALOCK Auto Lock

When this bit is set, Lock bit update (LUPD) is set to '1' on each overflow/underflow or re-trigger event.
This bit is not synchronized.

Value	Description
0	The LUPD bit is not affected on overflow/underflow and re-trigger event.
1	The LUPD bit is set on each overflow/underflow or re-trigger event.

Bits 10:8 – PRESCALER[2:0] Prescaler

These bits select the counter prescaler factor.
These bits are not synchronized.

Value	Name	Description
0x0	DIV1	Prescaler: GCLK_TC
0x1	DIV2	Prescaler: GCLK_TC/2
0x2	DIV4	Prescaler: GCLK_TC/4
0x3	DIV8	Prescaler: GCLK_TC/8
0x4	DIV16	Prescaler: GCLK_TC/16
0x5	DIV64	Prescaler: GCLK_TC/64
0x6	DIV256	Prescaler: GCLK_TC/256
0x7	DIV1024	Prescaler: GCLK_TC/1024

Bit 7 – ONDEMAND Clock On Demand

This bit selects the clock requirements when the TC is stopped.
In Standby mode, if the Run in Standby bit (CTRLA.RUNSTDBY) is '0', ONDEMAND is forced to '0'.
This bit is not synchronized.

Value	Description
0	The On Demand is disabled. If On Demand is disabled, the TC continues to request the clock when its operation is stopped (STATUS.STOP = 1).
1	The On Demand is enabled. When On Demand is enabled, the stopped TC will not request the clock. The clock is requested when a software re-trigger command is applied or when an event with start/re-trigger action is detected.

Bit 6 – RUNSTDBY Run in Standby

This bit is used to keep the TC running in Standby mode.
This bit is not synchronized.

Value	Description
0	The TC is halted in standby.
1	The TC continues to run in standby.

© 2025 Microchip Technology Inc. and its subsidiaries

Bits 5:4 – PRESCSYNC[1:0] Prescaler and Counter Synchronization

These bits select whether the counter must wrap around on the next GCLK_TCx clock or the next prescaled GCLK_TCx clock. It also makes it possible to reset the prescaler.
These bits are not synchronized.

Value	Name	Description
0x0	GCLK	Reload or reset the counter on next generic clock
0x1	PRESC	Reload or reset the counter on next prescaler clock
0x2	RESYNC	Reload or reset the counter on next generic clock. Reset the prescaler counter
0x3	—	Reserved

Bits 3:2 – MODE[1:0] Timer Counter Mode

These bits select the Counter mode.
These bits are not synchronized.

Value	Name	Description
0x0	COUNT16	Counter in 16-bit mode
0x1	COUNT8	Counter in 8-bit mode
0x2	COUNT32	Counter in 32-bit mode
0x3	—	Reserved

Bit 1 – ENABLE Enable

Due to synchronization, there is a delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE reads back immediately and the ENABLE Synchronization Busy bit in the SYNCBUSY register (SYNCBUSY.ENABLE) is set. SYNCBUSY.ENABLE is cleared when the operation is complete.
This bit is not enable-protected.

Value	Description
0	The peripheral is disabled.
1	The peripheral is enabled.

Bit 0 – SWRST Software Reset

Writing a '0' to this bit has no effect.
Writing a '1' to this bit resets all registers in the TC, except DBGCTRL, to their initial state and the TC is disabled.
Writing a '1' to CTRLA.SWRST always takes precedence; all other writes in the same write-operation are discarded.
This bit is not enable-protected.

© 2025 Microchip Technology Inc. and its subsidiaries

40.8.2. Control B Clear

Name: CTRLBCLR
Offset: 0x04
Reset: 0x00
Property: PAC Write-Protection, Read-Synchronized, Write-Synchronized

This register allows the user to clear bits in the CTRLB register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Set register (CTRLBSET).

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
Writing '0x0' to these bits has no effect.
Writing a '1' to any of these bits clears 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 disables one-shot operation.

Value	Description
0	The TC will wrap around and continue counting on an overflow/underflow condition.
1	The TC will wrap around and stop on the next underflow/overflow condition.

Bit 1 – LUPD Lock Update
This bit controls the update operation of the TC buffered registers.
When CTRLB.LUPD is set, no update of the registers with the value of its buffered register is performed on the hardware UPDATE condition. Locking the update ensures that all buffer registers are valid before a hardware update is performed. After all the buffer registers are loaded correctly, the buffered registers can be unlocked.
This bit has no effect when the input capture operation is enabled.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears 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 clears the bit and make the counter count up.

Value	Description
0	The timer/counter is counting up (incrementing).
1	The timer/counter is counting down (decrementing).

© 2025 Microchip Technology Inc. and its subsidiaries

40.8.3. Control B Set

Name: CTRLBSET
Offset: 0x05
Reset: 0x00
Property: PAC Write-Protection, Read-synchronized, Write-Synchronized

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 is executed, the CMD bit group is read back as '0'.

Writing '0x0' to these bits has no effect.

Writing a value different from '0x0' to these bits issues a command for execution.

Important: This command requires synchronization before executing it. The following is a valid sequence:

- Issue CMD command (CTRLBSET.CMD = command)
- Wait for CMD synchronization (SYNCBUSY.CTRLB = 0)
- Wait for CMD read back as zero (CTRLBSET.CMD = 0)

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 is executed, the CMD bit group is read back as '0'.
Writing '0x0' to these bits has no effect.
Writing a value different from '0x0' to these bits issues a command for execution.

Value	Name	Description
0x0	NONE	No action
0x1	RETRIGGER	Force a start, restart or retrigger
0x2	STOP	Force a stop
0x3	UPDATE	Force update of double buffered registers
0x4	READSYNC	Force a read synchronization of COUNT

Bit 2 – ONESHOT One-Shot on Counter
This bit controls one-shot operation of the TC.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit enables one-shot operation.

Value	Description
0	The TC wraps around and continue counting on an overflow/underflow condition.
1	The TC wraps around and stop on the next underflow/overflow condition.

Bit 1 – LUPD Lock Update
This bit controls the update operation of the TC buffered registers.
When CTRLB.LUPD is set, no update of the registers with a value of its buffered register is performed on the hardware UPDATE condition. Locking the update ensures that all buffer registers are valid before a hardware update is performed. After all the buffer registers are loaded correctly, the buffered registers can be unlocked.

© 2025 Microchip Technology Inc. and its subsidiaries

Writing a '0' to this bit has no effect.
Writing a '1' to this bit sets the LUPD bit.
This bit has no effect when input capture operation is enabled.

Value	Description
0	The CCBUFx and PERBUF buffer registers value are copied into CCx and PER registers on 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 sets the bit and make the counter count down.

Value	Description
0	The timer/counter is counting up (incrementing).
1	The timer/counter is counting down (decrementing).

© 2025 Microchip Technology Inc. and its subsidiaries

40.8.4. Event Control

Name: EVCTRL
Offset: 0x06
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
			MCEO1	MCEO0				OVFEO
Access			R/W	R/W				R/W
Reset			0	0				0

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

Bits 12, 13 – MCEOx Match or Capture Channel x Event Output Enable [x = 1..0]
These bits enable the generation of an event for every match or capture on channel x.

Value	Description
0	Match/Capture event on channel x is disabled and will not be generated.
1	Match/Capture event on channel x is enabled and will be generated for every compare/capture.

Bit 8 – OVFEO Overflow/Underflow Event Output Enable
This bit enables the Overflow/Underflow event. When enabled, an event will be generated when the counter overflows/underflows.

Value	Description
0	Overflow/Underflow event is disabled and will not be generated.
1	Overflow/Underflow event is enabled and will be generated for every counter overflow/underflow.

Bit 5 – TCEI TC Event Enable
This bit is used to enable asynchronous input events to the TC.

Value	Description
0	Incoming events are disabled.
1	Incoming events are enabled.

Bit 4 – TCINV TC Inverted Event Input Polarity
This bit inverts the asynchronous input event source.

Value	Description
0	Input event source is not inverted.
1	Input event source is inverted.

Bits 2:0 – EVACT[2:0] Event Action
These bits define the event action the TC will perform on an event.

Value	Name	Description
0x0	OFF	Event action disabled
0x1	RETRIGGER	Start, restart or retrigger TC on event
0x2	COUNT	Count on event
0x3	START	Start TC on event
0x4	STAMP	Time stamp capture
0x5	PPW	Period captured in CC0, pulse width in CC1

© 2025 Microchip Technology Inc. and its subsidiaries

Value	Name	Description
0x6	PWP	Period captured in CC1, pulse width in CC0
0x7	PW	Pulse width capture

© 2025 Microchip Technology Inc. and its subsidiaries

40.8.5. Interrupt Enable Clear

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

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

Bit	7	6	5	4	3	2	1	0
			MC1	MC0			ERR	OVF
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bits 4, 5 – MCx Match or Capture Channel x Interrupt Enable [x = 1..0]
Writing a '0' to these bits has no effect.
Writing a '1' to MCx will clear the corresponding Match or Capture Channel x Interrupt Enable bit, which disables the Match or Capture Channel x interrupt.

Value	Description
0	The Match or Capture Channel x interrupt is disabled.
1	The Match or Capture Channel x interrupt is enabled.

Bit 1 – ERR Error Interrupt Disable
Writing a '0' to these bits has no effect.
Writing a '1' to this bit will clear the Error Interrupt Enable bit, which disables the Error interrupt.

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

Bit 0 – OVF Overflow Interrupt Disable
Writing a '0' to these bits has no effect.
Writing a '1' to this bit will clear the Overflow Interrupt Enable bit, which disables the Overflow interrupt request.

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

© 2025 Microchip Technology Inc. and its subsidiaries

40.8.6. Interrupt Enable Set

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

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

Bit	7	6	5	4	3	2	1	0
			MC1	MC0			ERR	OVF
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bits 4, 5 – MCx Match or Capture Channel x Interrupt Enable [x = 1..0]
Writing a '0' to these bits has no effect.
Writing a '1' to MCx will set the corresponding Match or Capture Channel x Interrupt Enable bit, which enables the Match or Capture Channel x interrupt.

Value	Description
0	The Match or Capture Channel x interrupt is disabled.
1	The Match or Capture Channel x interrupt is enabled.

Bit 1 – ERR Error Interrupt Enable
Writing a '0' to these bits has no effect.
Writing a '1' to this bit will set the Error Interrupt Enable bit, which enables the Error interrupt.

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

Bit 0 – OVF Overflow Interrupt Enable
Writing a '0' to these bits has no effect.
Writing a '1' to this bit will set the Overflow Interrupt Enable bit, which enables the Overflow interrupt request.

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

© 2025 Microchip Technology Inc. and its subsidiaries

40.8.7. Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x0A
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
			MC1	MC0			ERR	OVF
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bits 4, 5 – MCx Match or Capture Channel x [x = 1..0]
This flag is set on a comparison match, or when the corresponding CCx register contains a valid capture value. This flag is set on the next CLK_TC_CNT cycle, and will generate an interrupt request if the corresponding Match or Capture Channel x Interrupt Enable bit in the Interrupt Enable Set register (INTENSET.MCx) is '1'.
Writing a '0' to these bits has no effect.
Writing a '1' to one of these bits will clear the corresponding Match or Capture Channel x interrupt flag
In capture operation, this flag is automatically cleared when CCx register is read.

Bit 1 – ERR Error Interrupt Flag
This flag is set when a new capture occurs on a channel while the corresponding Match or Capture Channel x interrupt flag is set, in which case there is no place to store the new capture.
Writing a '0' to these bits has no effect.
Writing a '1' to this bit clears the Error interrupt flag.

Bit 0 – OVF Overflow Interrupt Flag
This flag is set on the next CLK_TC_CNT cycle after an overflow condition occurs, and will generate an interrupt request if INTENCLR.OVF or INTENSET.OVF is '1'.
Writing a '0' to these bits has no effect.
Writing a '1' to this bit clears the Overflow interrupt flag.

© 2025 Microchip Technology Inc. and its subsidiaries

40.8.8. Status

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

Bit	7	6	5	4	3	2	1	0
			CCBUFV1	CCBUFV0	PERBUFV		CLIENT	STOP
Access			R/W	R/W	R/W		R	R
Reset			0	0	0		0	1

Bits 4, 5 – CCBUFVx Channel x Compare or Capture Buffer Valid [x=1..0]
For a compare channel x, the bit x is set when a new value is written to the corresponding CCBUFx register.
The bit x is cleared by writing a '1' to it 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 the UPDATE condition. This bit is available only in 8-bit mode and will always read zero in 16- and 32-bit modes.

Bit 1 – CLIENT Client Status Flag
This bit is only available in 32-bit mode on the Client TC (in other words, TC1, TC3, TC5, TC7 and/or TC9). The bit is set when the associated Host TC (TC0, TC2, TC4, TC6 and/or TC8, 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.

© 2025 Microchip Technology Inc. and its subsidiaries

40.8.9. Waveform Generation Control

Name: WAVE
Offset: 0x0C
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

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

Bits 1:0 – WAVEGEN[1:0] Waveform Generation Mode
These bits select the waveform generation operation. They affect the top value, as shown in Waveform Output Operations. They also control whether frequency or PWM waveform generation must be used. The waveform generation operations are explained in Waveform Output Operations. See *Waveform Output Operations* from Related Links.
These bits are not synchronized.

Value	Name	Operation	Top Value	Output Waveform on Match	Output Waveform on Wraparound
0x0	NFRQ	Normal frequency	PER[1] / Max	Toggle	No action
0x1	MFRQ	Match frequency	CC0	Toggle	No action
0x2	NPWM	Normal PWM	PER[1] / Max	Set	Clear
0x3	MPWM	Match PWM	CC0	Set	Clear

1. This depends on the TC mode: In 8-bit mode, the top value is the Period Value register (PER). In 16- and 32-bit mode, it is the respective MAX value.

Related Links
Waveform Output Operations

© 2025 Microchip Technology Inc. and its subsidiaries