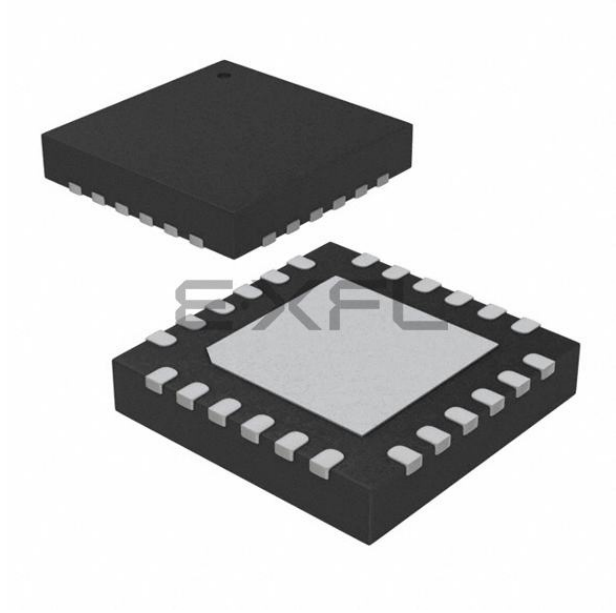


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

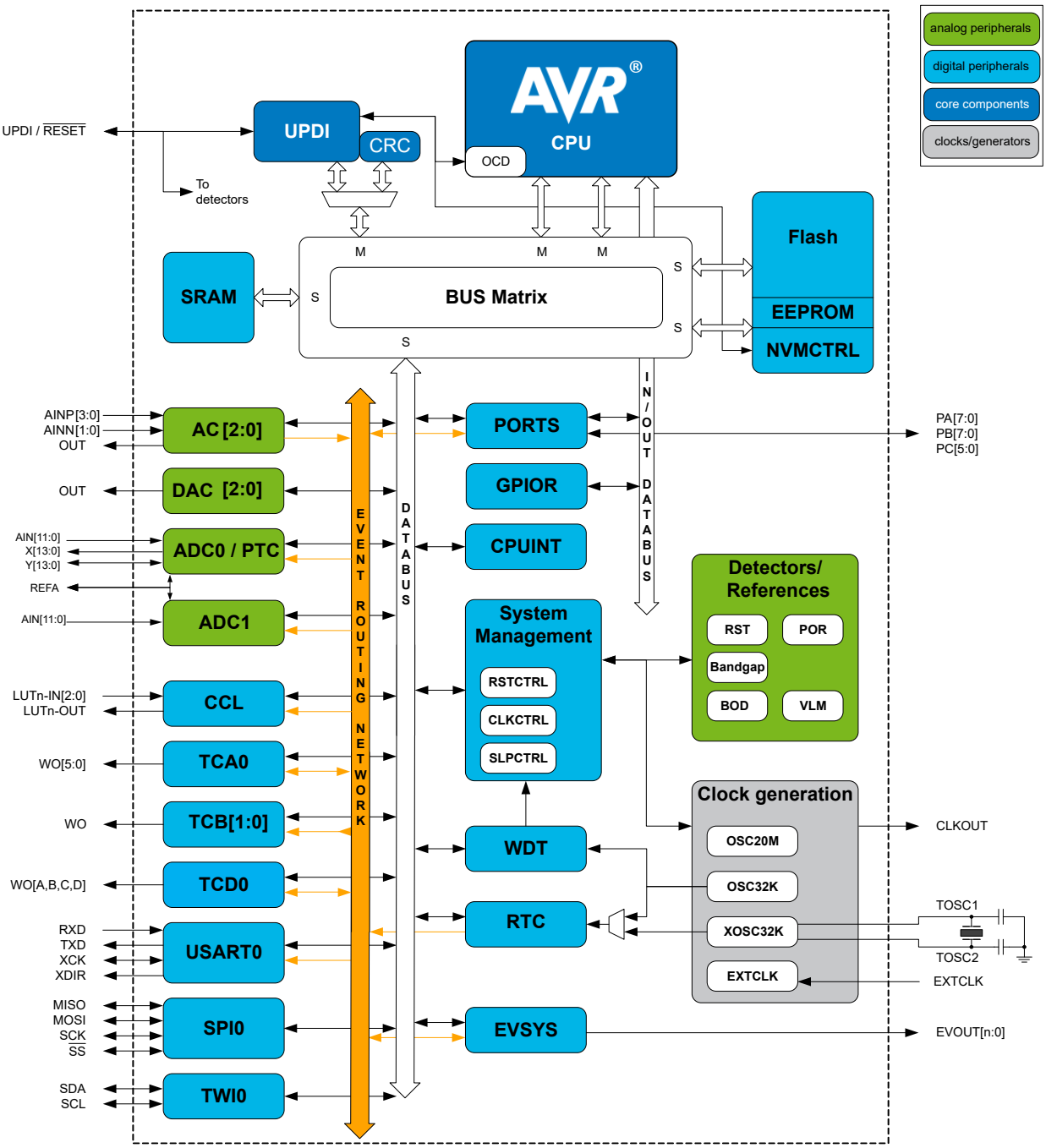
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded"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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	AVR
Core Size	8-Bit
Speed	16MHz
Connectivity	I²C, IrDA, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	22
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 24x10b; D/A 3x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	24-VFQFN Exposed Pad
Supplier Device Package	24-VQFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/attiny1617-mfr

3. Block Diagram

Figure 3-1. ATtiny3217/ATtiny1617 Block Diagram



6.10.4.9 Lockbits

Name: LOCKBIT
Offset: 0x0A
Reset: -
Property: -

Bit	7	6	5	4	3	2	1	0
	LOCKBIT[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 – LOCKBIT[7:0] Lockbits
When the part is locked, UPDI cannot access the system bus, so it cannot read out anything but CS-space.

Value	Description
0xC5	Valid key - the device is open
other	Invalid - the device is locked

Related Links

6.9 Memory Section Access from CPU and UPDI on Locked Device

9.5.2 Control B

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

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

Bit 1 – BOOTLOCK Boot Section Lock
Writing a ’1’ to this bit locks the boot section from read and instruction fetch.

If this bit is ’1’, a read from the boot section will return ’0’. A fetch from the boot section will also return ‘0’ as instruction.

This bit can be written from the boot section only. It can only be cleared to ’0’ by a Reset.

This bit will take effect only when the boot section is left the first time after the bit is written.

Bit 0 – APCWP Application Code Section Write Protection
Writing a ’1’ to this bit protects the application code section from further writes.

This bit can only be written to ’1’. It is cleared to ’0’ only by Reset.

Table 16-1. PORT System Dependencies

Dependency	Applicable	Peripheral
Clocks	Yes	CLKCTRL
I/O Lines and Connections	No	-
Interrupts	Yes	CPUINT
Events	Yes	EVSYS
Debug	No	-

Related Links

16.2.3.4 Events
16.2.3.1 Clocks
16.2.3.3 Interrupts

16.2.3.1 Clocks

This peripheral depends on the peripheral clock.

16.2.3.2 I/O Lines and Connections

Not applicable.

16.2.3.3 Interrupts

Using the interrupts of this peripheral requires the interrupt controller to be configured first.

Related Links

13. CPUINT - CPU Interrupt Controller
16.3.3 Interrupts
8.7.3 SREG

16.2.3.4 Events

The events of this peripheral are connected to the Event System.

Related Links

14. EVSYS - Event System

16.2.3.5 Debug Operation

This peripheral is unaffected by entering Debug mode.

16.3 Functional Description

16.3.1 Initialization

After Reset, all standard function device I/O pads are connected to the port with outputs tri-stated and input buffers enabled, even if there is no clock running.

Power consumption can be reduced by disabling digital input buffers for all unused pins and for pins used as analog inputs or outputs.

Specific pins, such as those used for connecting a debugger, may be configured differently, as required by their special function.

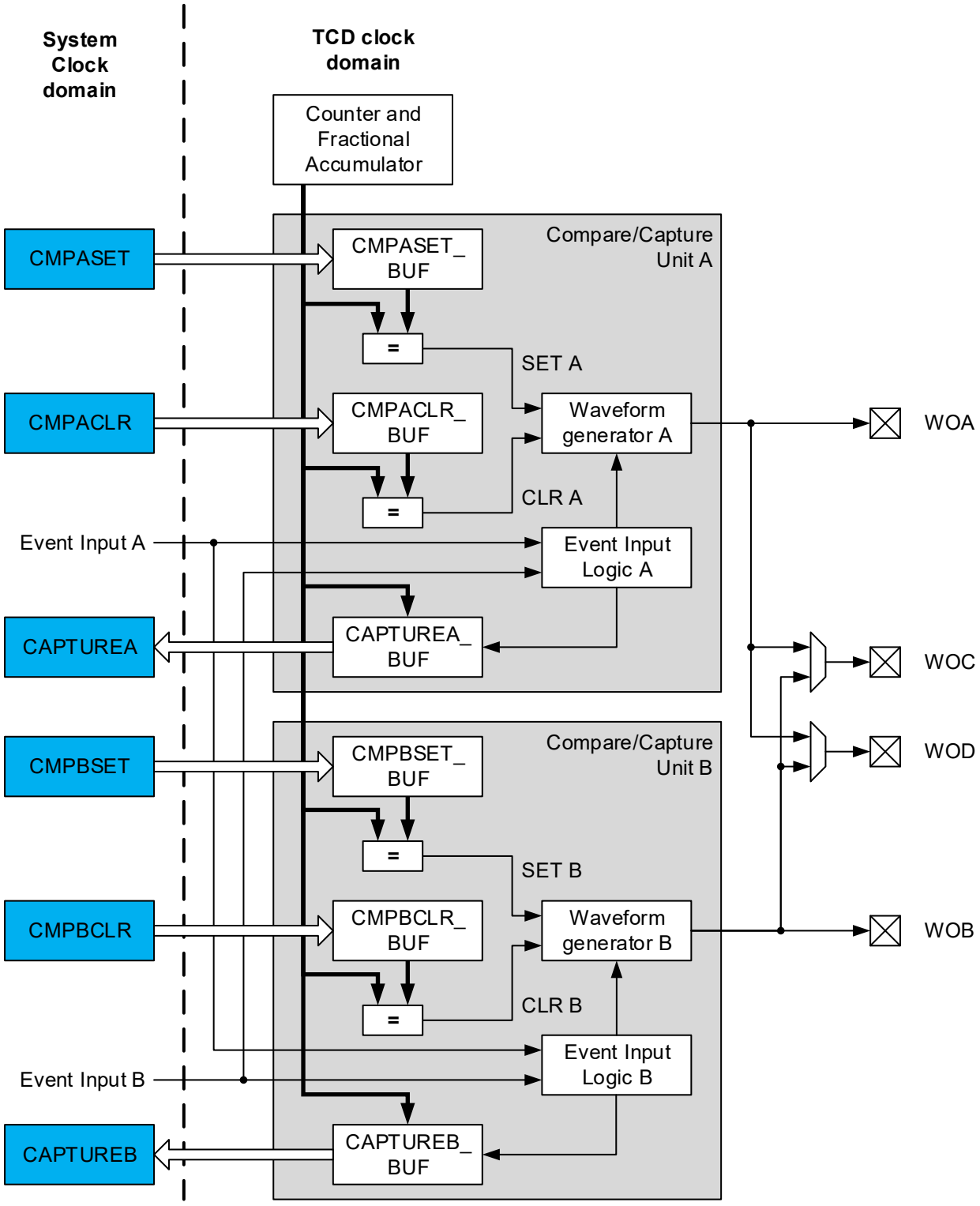
16.4 Register Summary - PORT

Offset	Name	Bit Pos.								
0x00	DIR	7:0	DIR[7:0]							
0x01	DIRSET	7:0	DIRSET[7:0]							
0x02	DIRCLR	7:0	DIRCLR[7:0]							
0x03	DIRTGL	7:0	DIRTGL[7:0]							
0x04	OUT	7:0	OUT[7:0]							
0x05	OUTSET	7:0	OUTSET[7:0]							
0x06	OUTCLR	7:0	OUTCLR[7:0]							
0x07	OUTTGL	7:0	OUTTGL[7:0]							
0x08	IN	7:0	IN[7:0]							
0x09	INTFLAGS	7:0	INT[7:0]							
0x0A ... 0x0F	Reserved									
0x10	PIN0CTRL	7:0	INVEN				PULLUPEN	ISC[2:0]		
0x11	PIN1CTRL	7:0	INVEN				PULLUPEN	ISC[2:0]		
0x12	PIN2CTRL	7:0	INVEN				PULLUPEN	ISC[2:0]		
0x13	PIN3CTRL	7:0	INVEN				PULLUPEN	ISC[2:0]		
0x14	PIN4CTRL	7:0	INVEN				PULLUPEN	ISC[2:0]		
0x15	PIN5CTRL	7:0	INVEN				PULLUPEN	ISC[2:0]		
0x16	PIN6CTRL	7:0	INVEN				PULLUPEN	ISC[2:0]		
0x17	PIN7CTRL	7:0	INVEN				PULLUPEN	ISC[2:0]		

16.5 Register Description - Ports

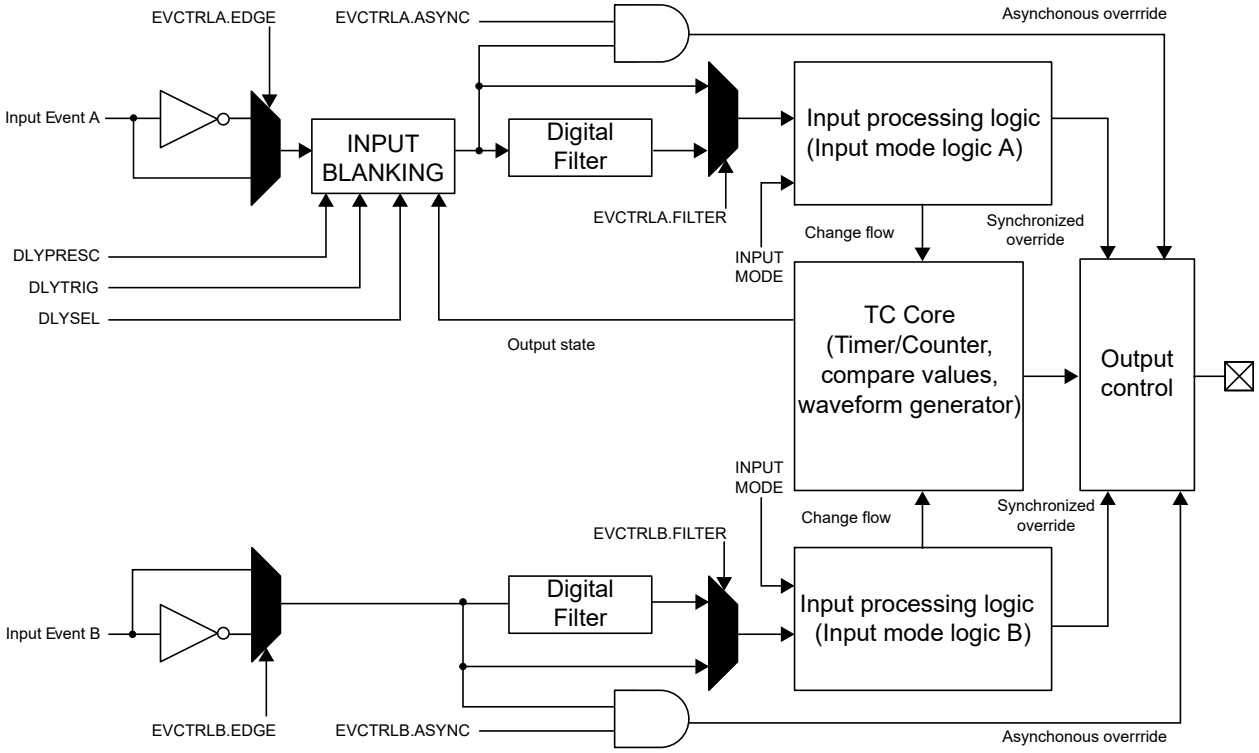
22.2.1 Block Diagram

Figure 22-1. Timer/Counter Block Diagram



The TCD core is asynchronous to the system clock. The timer/counter consist of two compare/capture units, each with a separate waveform output. In addition, there are two extra waveform outputs which can be equal to the output from one of the units. The compare registers CMPxSET and CMPxCLR are stored

Figure 22-9. TCD Input Overview



There is a delay of 2-3 clock cycles on the TCD synchronizer clock between receiving the input event, processing it, and overriding the outputs. If using the asynchronous event detection, the outputs will override instantly outside the input processing.

22.3.2.4.1 Input Blanking

Input blanking functionality is masking out the input events for a programmable time in a selectable part of the TCD cycle. Input blanking can be used to mask out 'false' input events that are triggered right after changes on the outputs.

To enable input blanking, write 0x1 to the Delay Select bit field in the Delay Control register (DLYSEL in TCDn.DLYCTRL). The trigger source is selected by the Delay Trigger bit field (DLYTRIG in TCDn.DLYCTRL).

Input blanking uses the Delay clock: after a trigger, a counter is counting up until the Delay Value (DLYVAL in TCDn.DLYVAL) is reached before input blanking is turned OFF. The TCD delay clock is a prescaled version of the Synchronization clock. The division factor is set by the Delay Prescaler bit field in the Delay Control register (DLYPRESC in TCDn.DLYCTRL). The duration of the input blanking is given by:

$$t_{\mathrm{BLANK}} = \frac{\mathrm{DLYPRESC\_division\_factor} \times \mathrm{DLYVAL}}{f_{\mathrm{CLK\_TCD\_SYNC}}}$$

Input blanking is using the same logic as the programmable output event. For this reason, it is not possible to use both at the same time.

22.3.2.4.2 Digital Filter

The digital filter for event input x is enabled by writing a '1' to the FILTER bit in the Event Control x register (TCDn.EVCTRLx). When the digital filter is enabled, any pulse lasting less than four counter clock cycles will be filtered out. Any change on the incoming event will, therefore, take four counter clock cycles before it will affect the input processing logic.

22.3.2.4.3 Asynchronous Event Detection

To enable asynchronous event detection on an input event, write a '1' to the Asynchronous Event Control bit (ASYNC) in the Event Control register (TCDn.EVCTRLx).

The asynchronous event detection makes it possible to asynchronously override the output when the input event occurs. What the Input event will do is depending on Input Mode for the event input. The outputs have direct override while the counter flow will be changed when the event is synchronized to the synchronization clock.

It is not possible to use both asynchronous event detection and digital filtering at the same time.

22.3.2.4.4 Input Modes

The user can select between 10 input modes. The selection is done by writing the Input Mode bit field (INPUTMODE) in the Input Control x register (TCDn.INPUTCTRLx).

Table 22-4. Input Mode Description

INPUTMODE	Description
0x0	Input has no action
0x1	Stop output, jump to opposite compare cycle and wait
0x2	Stop output, execute opposite compare cycle and wait
0x3	Stop output, execute opposite compare cycle while fault active
0x4	Stop all outputs, maintain frequency
0x5	Stop all outputs, execute dead time while fault active
0x6	Stop all outputs, jump to next compare cycle and wait
0x7	Stop all outputs, wait for software action
0x8	Stop output on edge, jump to next compare cycle
0x9	Stop output on edge, maintain frequency
0xA	Stop output at level, maintain frequency
other	Reserved

Not all input modes work in all Waveform Generation modes. Below is a table that shows what Waveform Generation modes the different input modes are valid in.

Table 22-5. Ramp Mode the Different Input Modes are Valid In

INPUTMODE	One Ramp Mode	Two Ramp Mode	Four Ramp Mode	Dual Slope Mode
0x1	Valid	Valid	Valid	Do not use
0x2	Do not use	Valid	Valid	Do not use
0x3	Do not use	Valid	Valid	Do not use
0x4	Valid	Valid	Valid	Valid
0x5	Do not use	Valid	Valid	Do not use
0x6	Do not use	Valid	Valid	Do not use

22.5.6 Event Control x

Name: EVCTRL
Offset: 0x08 + n*0x01 [n=0..1]
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	CFG[1:0]			EDGE		ACTION		TRIGEI
Access	R/W	R/W		R/W		R/W		R/W
Reset	0	0		0		0		0

Bits 7:6 – CFG[1:0] Event Configuration
When the input capture noise canceler is activated (FILTERON), the event input is filtered. The filter function requires four successive equal valued samples of the retrigger pin for changing its output. The input capture is therefore delayed by four clock cycles when the noise canceler is enabled.

When the Asynchronous Event is enabled (ASYNCON), the event input will qualify the output directly.

Value	Name	Description
0x0	NEITHER	Neither filter nor asynchronous event is enabled.
0x1	FILTERON	Input capture noise cancellation filter enabled.
0x2	ASYNCON	Asynchronous event output qualification enabled.
other	-	Reserved.

Bit 4 – EDGE Edge Selection
This bit is used to select the active edge or level for the event input.

Value	Name	Description
0	FALL_LOW	The falling edge or low level of the event input generates retrigger or fault action.
1	RISE_HIGH	The rising edge or high level of the event input generates retrigger or fault action.

Bit 2 – ACTION Event Action
This bit enables capture on event input. By default, the input will trigger a fault, depending on the Input x register Input mode (INPUTx). It is also possible to trigger a capture on the event input.

Value	Name	Description
0	FAULT	Event triggers a fault.
1	CAPTURE	Event triggers a fault and capture.

Bit 0 – TRIGEI Trigger Event Input Enable
Writing this bit to '1' enables event as trigger for input A.

23.10 Register Summary - RTC

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0	RUNSTDBY	PRESCALER[3:0]						RTCEN
0x01	STATUS	7:0					CMPBUSY	PERBUSY	CNTBUSY	CTRLABUSY
0x02	INTCTRL	7:0							CMP	OVF
0x03	INTFLAGS	7:0							CMP	OVF
0x04	TEMP	7:0	TEMP[7:0]							
0x05	DBGCTRL	7:0								DBGRUN
0x06	Reserved									
0x07	CLKSEL	7:0							CLKSEL[1:0]	
0x08	CNT	7:0	CNT[7:0]							
		15:8	CNT[15:8]							
0x0A	PER	7:0	PER[7:0]							
		15:8	PER[15:8]							
0x0C	CMP	7:0	CMP[7:0]							
		15:8	CMP[15:8]							
0x0E ... 0x0F	Reserved									
0x10	PITCTRLA	7:0		PERIOD[3:0]						PITEN
0x11	PITSTATUS	7:0								CTRLBUSY
0x12	PITINTCTRL	7:0								PI
0x13	PITINTFLAGS	7:0								PI
0x14	Reserved									
0x15	PITDBGCTRL	7:0								DBGRUN

23.11 Register Description

Offset	Name	Vector Description	Conditions
			• RXC: Receive Complete Interrupt

When an interrupt condition occurs, the corresponding interrupt flag is set in the Interrupt Flags register of the peripheral (*peripheral*.INTFLAGS).

An interrupt source is enabled or disabled by writing to the corresponding enable bit in the peripheral's Interrupt Control register (*peripheral*.INTCTRL).

An interrupt request is generated when the corresponding interrupt source is enabled and the interrupt flag is set. The interrupt request remains active until the interrupt flag is cleared. See the peripheral's INTFLAGS register for details on how to clear interrupt flags.

Related Links

8.7.3 SREG

13. CPUINT - CPU Interrupt Controller

25.3.4 Sleep Mode Operation

The SPI will continue working in Idle Sleep mode. When entering any deeper sleep mode, an active transaction will be stopped.

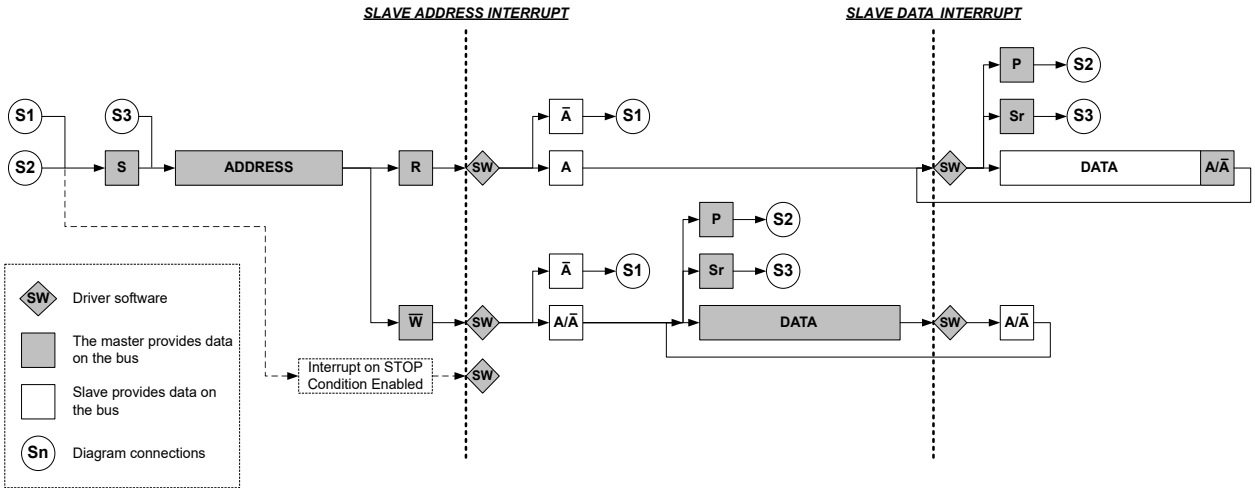
Related Links

11. SLPCTRL - Sleep Controller

25.3.5 Configuration Change Protection

Not applicable.

Figure 26-16. TWI Slave Operation



The number of interrupts generated is kept to a minimum by automatic handling of most conditions. Quick command can be enabled to auto-trigger operations and reduce software complexity.

Address Recognition mode can be enabled to allow the slave to respond to all received addresses.

26.3.4.3.1 Receiving Address Packets

When the TWI slave is properly configured, it will wait for a Start condition to be detected. When this happens, the successive address byte will be received and checked by the address match logic, and the slave will ACK a correct address and store the address in the TWIn.DATA register. If the received address is not a match, the slave will not acknowledge and store the address, but wait for a new Start condition.

The slave address/stop interrupt flag is set when a Start condition succeeded by a valid address byte is detected. A general call address will also set the interrupt flag.

A Start condition immediately followed by a Stop condition is an illegal operation and the bus error flag is set.

The R/$\overline{W}$ direction flag reflects the direction bit received with the address. This can be read by software to determine the type of operation currently in progress.

Depending on the R/$\overline{W}$ direction bit and bus condition, one of four distinct cases (S1 to S4) arises following the address packet. The different cases must be handled in software.

Case S1: Address Packet Accepted - Direction Bit Set

If the R/$\overline{W}$ direction flag is set, this indicates a master read operation. The SCL line is forced low by the slave, stretching the bus clock. If ACK is sent by the slave, the slave hardware will set the data interrupt flag indicating data is needed for transmit. Data, repeated Start, or Stop can be received after this. If NACK is sent by the slave, the slave will wait for a new Start condition and address match.

Case S2: Address Packet Accepted - Direction Bit Cleared

If the R/$\overline{W}$ direction flag is cleared, this indicates a master write operation. The SCL line is forced low, stretching the bus clock. If ACK is sent by the slave, the slave will wait for data to be received. Data, repeated Start, or Stop can be received after this. If NACK is sent, the slave will wait for a new Start condition and address match.

Case S3: Collision

If the slave is not able to send a high level or NACK, the collision flag is set, and it will disable the data and acknowledge output from the slave logic. The clock hold is released. A Start or repeated Start condition will be accepted.

26.5.2 Debug Control

Name: DBGCTRL
Offset: 0x02
Reset: 0x00
Property: -

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

Bit 0 – DBGRUN Debug Run

Value	Description
0	The peripheral is halted in Break Debug mode and ignores events
1	The peripheral will continue to run in Break Debug mode when the CPU is halted

28.5.4 Interrupt Flag

Name: INTFLAGS
Offset: 0x07
Reset: 0
Property: R/W

Bit	7	6	5	4	3	2	1	0
							INT1	INT0
Access								
Reset							0	0

Bits 0, 1 – INT Interrupt Flag
The INTn flag is set when LUTn output change matches the interrupt sense mode as defined in CCL.INTCTRLn. Writing a '1' to this flag's bit location will clear the flag.

28.5.8 TRUTHn

Name: TRUTH
Offset: 0x08 + n*0x04 [n=0..1]
Reset: 0x00
Property: Enable-Protected

Bit	7	6	5	4	3	2	1	0
	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

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

30.3.4 Interrupts

Table 30-2. Available Interrupt Vectors and Sources

Offset	Name	Vector Description	Conditions
0x00	RESRDY	Result Ready interrupt	The conversion result is available in the Result register (ADC.RES).
0x04	WCOMP	Window Comparator interrupt	As defined by WINCM in ADC.CTRLE.

When an interrupt condition occurs, the corresponding interrupt flag is set in the Interrupt Flags register of the peripheral (*peripheral*.INTFLAGS).

An interrupt source is enabled or disabled by writing to the corresponding enable bit in the peripheral's Interrupt Control register (*peripheral*.INTCTRL).

An interrupt request is generated when the corresponding interrupt source is enabled and the interrupt flag is set. The interrupt request remains active until the interrupt flag is cleared. See the peripheral's INTFLAGS register for details on how to clear interrupt flags.

30.3.5 Sleep Mode Operation

The ADC is by default disabled in Standby Sleep mode.

The ADC can stay fully operational in Standby Sleep mode if the Run in Standby bit (RUNSTDBY) in the Control A register (ADC.CTRLA) is written to '1'.

When the device is entering Standby Sleep mode when RUNSTDBY is '1', the ADC will stay active, hence any ongoing conversions will be completed and interrupts will be executed as configured.

In Standby Sleep mode an ADC conversion must be triggered via the Event System (EVSYS), or the ADC must be in free-running mode with the first conversion triggered by software before entering sleep. The peripheral clock is requested if needed and is turned OFF after the conversion is completed.

When an input event trigger occurs, the positive edge will be detected, the Start Conversion bit (STCONV) in the Command register (ADC.COMMAND) will be set, and the conversion will start. When the conversion is completed, the Result Ready Flag (RESRDY) in the Interrupt Flags register (ADC.INTFLAGS) is set and the STCONV bit in ADC.COMMAND is cleared.

The reference source and supply infrastructure need time to stabilize when activated in Standby Sleep mode. Configure a delay for the start of the first conversion by writing a non-zero value to the Initial Delay bits (INITDLY) in the Control D register (ADC.CTRLD).

In Power-Down Sleep mode, no conversions are possible. Any ongoing conversions are halted and will be resumed when going out of sleep. At the end of the conversion, the Result Ready Flag (RESRDY) will be set, but the content of the result registers (ADC.RES) is invalid since the ADC was halted in the middle of a conversion.

Related Links

11. SLPCTRL - Sleep Controller

30.3.6 Synchronization

Not applicable.

30.3.7 Configuration Change Protection

Not applicable.

30.5.7 MUXPOS

Name: MUXPOS
Offset: 0x06
Reset: 0x00
Property: -

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

Bits 4:0 – MUXPOS[4:0] MUXPOS
This bit field selects which single-ended analog input is connected to the ADC. If these bits are changed during a conversion, the change will not take effect until this conversion is complete.

Value	Name	Description
0x00	AIN0	ADC input pin 0
0x01	AIN1	ADC input pin 1
0x02	AIN2	ADC input pin 2
0x03	AIN3	ADC input pin 3
0x04	AIN4	ADC input pin 4
0x05	AIN5	ADC input pin 5
0x06	AIN6	ADC input pin 6
0x07	AIN7	ADC input pin 7
0x08	AIN8	ADC input pin 8
0x09	AIN9	ADC input pin 9
0x0A	AIN10	ADC input pin 10
0x0B	AIN11	ADC input pin 11
0x1B	PTC	ADC0: Reserved / ADC1: DAC2
0x1C	DAC0	DAC0
0x1D	INTREF	Internal reference (from VREF peripheral)
0x1E	TEMPSENSE	ADC0: Temperature sensor / ADC1: DAC1
0x1F	GND	0V (GND)
Other	-	Reserved

30.5.17 Calibration

Name: CALIB
Offset: 0x16
Reset: 0x01
Property: -

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

Bit 0 – DUTYCYC Duty Cycle
This bit determines the duty cycle of the ADC clock.

ADC_{clk} > 1.5 MHz requires a minimum operating voltage of 2.7V

Value	Description
0	50% Duty Cycle must be used if ADC_{clk} > 1.5 MHz
1	25% Duty Cycle (high 25% and low 75%) must be used for ADC_{clk} ≤ 1.5 MHz

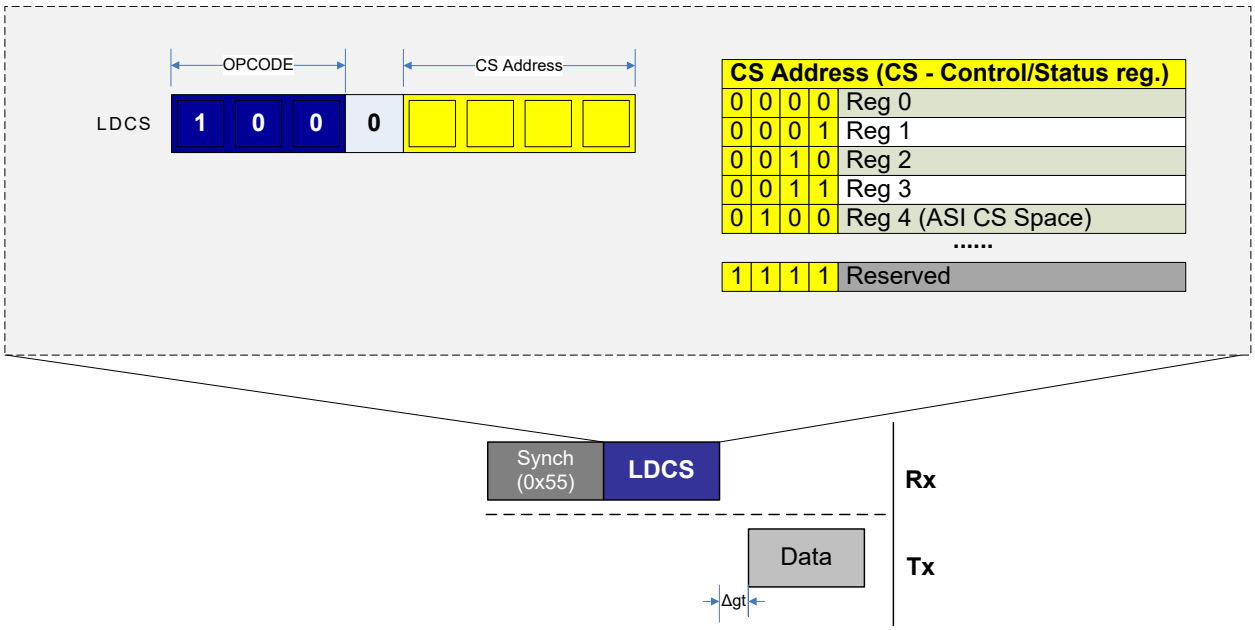
- Wait for the ACK character which signifies a successful write to the bus matrix

When used with the REPEAT, it is recommended to set up the address register with the start address for the block to be written and use the Pointer Post Increment register to automatically increase the address for each repeat cycle. When using REPEAT, the data frame of Size B data bytes can be sent after each received ACK.

33.3.3.5 LCDS - Load Data from Control and Status Register Space

The LCDS instruction is used to load data from the UPDI and ASI CS-space. LCDS is based on direct addressing, where the address is part of the instruction opcode. The total address space for LCDS is 16 bytes and can only access the internal UPDI register space. This instruction only supports byte access and the data size is not configurable.

Figure 33-13. LCDS Instruction Operation



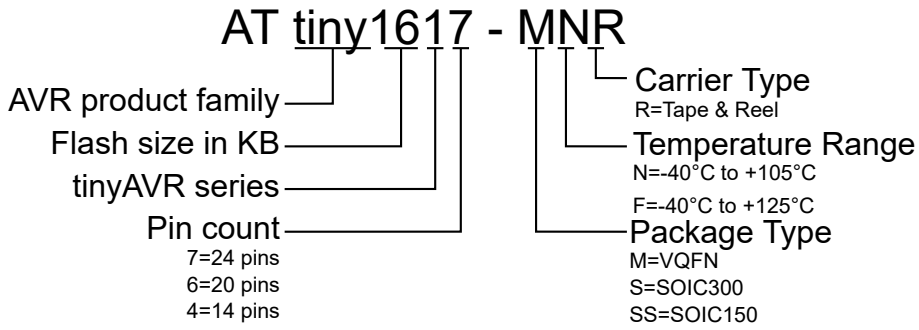
The figure above shows a typical example of LCDS data transmission. A data byte from the LCDS space is transmitted from the UPDI after the Guard Time is completed.

33.3.3.6 STCS (Store Data to Control and Status Register Space)

The STCS instruction is used to store data to the UPDI and ASI CS-space. STCS is based on direct addressing, where the address is part of the instruction opcode. The total address space for STCS is 16 bytes, and can only access the internal UPDI register space. This instruction only supports byte access, and data size is not configurable.

Product Identification System

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.



Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Legal Notice

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.