Microchip Technology - ATTINY1614-SSNR Datasheet

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	20MHz
Connectivity	I²C, IrDA, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	12
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)	1.8V ~ 5.5V
Data Converters	A/D 14x10b; D/A 3x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	14-SOIC (0.154", 3.90mm Width)
Supplier Device Package	14-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/attiny1614-ssnr

Base Address	Name	Description
0x06A8	DAC1	Digital to Analog Converter 1
0x06B0	DAC2	Digital to Analog Converter 2
0x0800	USART0	Universal Synchronous Asynchronous Receiver Transmitter
0x0810	TWI0	Two Wire Interface
0x0820	SPI0	Serial Peripheral Interface
0x0A00	TCA0	Timer/Counter Type A instance 0
0x0A40	TCB0	Timer/Counter Type B instance 0
0x0A50	TCB1	Timer/Counter Type B 1
0x0A80	TCD0	Timer/Counter Type D instance 0
0x0F00	SYSCFG	System Configuration
0x1000	NVMCTRL	Non Volatile Memory Controller
0x1100	SIGROW	Signature Row
0x1280	FUSES	Device specific fuses
0x1300	USERROW	User Row

7.2 Interrupt Vector Mapping

Each of the 26 interrupt vectors is connected to one peripheral instance, as shown in the table below. A peripheral can have one or more interrupt sources, see the 'Interrupt' section in the 'Functional Description' of the respective peripheral for more details on the available interrupt sources.

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

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

The naming of the registers may vary slightly in some peripherals.

An interrupt request is generated when the corresponding interrupt 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.

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

Table 7-2. Interrupt Vector Mapping

Vector Number	Base Address	Peripheral source
0	0x00	RESET
1	0x02	NMI - Non Maskable Interrupt from CRC
2	0x04	VLM - Voltage Level Monitor
3	0x06	PORTA - Port A

Value	Description
0	OSCULP32K is not stable
1	OSCULP32K is stable

Bit 4 – OSC20MS: OSC20M Status

The status bit will only be available if the source is requested as the main clock or by another module. If the oscillators RUNSTDBY bit is set but the oscillator is unused/not requested this bit will be 0.

Value	Description
0	OSC20M is not stable
1	OSC20M is stable

Bit 0 – SOSC: Main Clock Oscillator Changing

Value	Description
0	The clock source for CLK_MAIN is not undergoing a switch.
1	The clock source for CLK_MAIN is undergoing a switch, and will change as soon as the new source is stable.

10.5.5 16/20MHz Oscillator Control A

Name: OSC20MCTRLA
Offset: 0x10
Reset: 0x00
Property: Configuration Change Protection

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

Bit 1 – RUNSTDBY: Run Standby

This bit force the oscillator on in all modes, even when unused by the system. In standby sleep mode this can be used to ensure immediate wake-up and not waiting for oscillator start-up time.

When not requested by peripherals, no oscillator output is provided.

It takes 4 oscillator cycles to open the clock gate after a request but the oscillator analog start-up time will be removed when this bit is set.

10.5.6 16/20MHz Oscillator Calibration A

Name: OSC20MCALIBA
Offset: 0x11
Reset: Based on FREQSEL in FUSE.OSCCFG
Property: Configuration Change Protection

SREG

20.3.6 Sleep Mode Operation

The timer/counter will halt operation in all sleep modes.

20.3.7 Configuration Change Protection

Not applicable.

20.5 Register Description - Normal Mode

20.5.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: -

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

Bits 3:1 – CLKSEL[2:0]: Clock Select
These bits select the clock frequency for the timer/counter.

Value	Name	Description
0x0	DIV1	$f_{TCA} = f_{CLK\_PER}/1$
0x1	DIV2	$f_{TCA} = f_{CLK\_PER}/2$
0x2	DIV4	$f_{TCA} = f_{CLK\_PER}/4$
0x3	DIV8	$f_{TCA} = f_{CLK\_PER}/8$
0x4	DIV16	$f_{TCA} = f_{CLK\_PER}/16$
0x5	DIV64	$f_{TCA} = f_{CLK\_PER}/64$
0x6	DIV256	$f_{TCA} = f_{CLK\_PER}/256$
0x7	DIV1024	$f_{TCA} = f_{CLK\_PER}/1024$

Bit 0 – ENABLE: Enable

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

20.5.2 Control B - Normal Mode

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

Bit	7	6	5	4	3	2	1	0
		CMPnEN2	CMPnEN1	CMPnEN0	ALUPD	WGMODE[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 4, 5, 6 – CMPnEN: Compare n Enable
In the FRQ or PWM waveform generation mode, these bits will override the PORT output register for the corresponding pin.

21.4 Register Summary - TCB

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0		RUNSTDBY		SYNCUPD		CLKSEL[1:0]		ENABLE
0x01	CTRLB	7:0		ASYNC	CCMPINIT	CCMPEN		CNTMODE[2:0]		
0x02 ... 0x03	Reserved									
0x04	EVCTRL	7:0		FILTER		EDGE				CAPTEI
0x05	INTCTRL	7:0								CAPT
0x06	INTFLAGS	7:0								CAPT
0x07	STATUS	7:0								RUN
0x08	DBGCTRL	7:0								DBGRUN
0x09	TEMP	7:0	TEMP[7:0]							
0x0A	CNT	7:0	CNT[7:0]							
0x0B		15:8	CNT[15:8]							
0x0C	CCMP	7:0	CCMP[7:0]							
0x0D		15:8	CCMP[15:8]							

21.5 Register Description

21.5.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
		RUNSTDBY		SYNCUPD		CLKSEL[1:0]		ENABLE
Access		R/W		R/W		R/W	R/W	R/W
Reset		0		0		0	0	0

Bit 6 – RUNSTDBY: Run Standby
Writing a '1' to this bit will enable the peripheral to run in Standby sleep mode. Not applicable when CLKSEL is set to 0x2 (CLK_TCA).

Bit 4 – SYNCUPD: Synchronize Update
When this bit is written to '1', the TCB will restart whenever the TCA0 is restarted.

Bits 2:1 – CLKSEL[1:0]: Clock Select
Writing these bits selects the clock source for this peripheral.

Value	Description
0x0	CLK_PER
0x1	CLK_PER / 2

Value	Description
0x6	Single shot mode
0x7	8-bit PWM mode

21.5.3 Event Control

Name: EVCTRL
Offset: 0x04
Reset: 0x00
Property: -

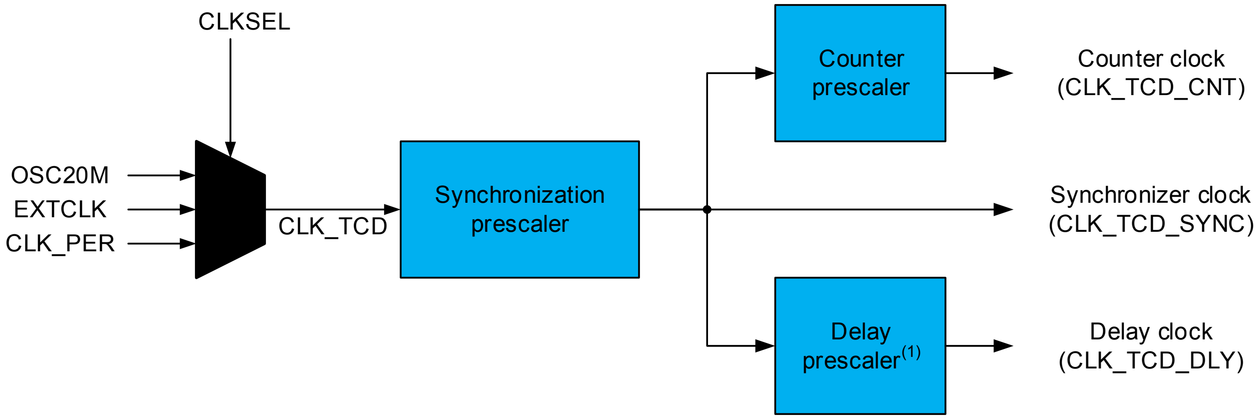
Bit	7	6	5	4	3	2	1	0
		FILTER		EDGE				CAPTEI
Access		R/W		R/W				R/W
Reset		0		0				0

Bit 6 – FILTER: Input Capture Noise Cancellation Filter
Writing this bit to '1' enables the input capture noise cancellation unit.

Bit 4 – EDGE: Event Edge
This bit is used to select the Event edge. The effect of this bit is dependent on the selected Count Mode (CNTMODE) in TCB.CTRLB.

Count Mode	EDGE	Positive Edge	Negative Edge
Periodic Interrupt Mode	0	Not Applicable	Not Applicable
	1	Not Applicable	Not Applicable
Timeout Check Mode	0	Start counter	Stop counter
	1	Stop counter	Start counter
Input Capture on Event Mode	0	Input capture frequency and pulse width measurement mode	Not Applicable
	1	Not Applicable	Capture = count
Input Capture Frequency Measurement Mode	0	Capture = count, initialize, interrupt	Not Applicable
	1	Not Applicable	Capture = count, initialize, interrupt
Input Capture Pulse Width Measurement Mode	0	Initialize	Capture = count, interrupt
	1	Capture = count, interrupt	Initialize

Figure 22-2. Clock Selection and Prescalers Overview


CLKSEL
OSC20M
EXTCLK
CLK_PER
CLK_TCD
Synchronization prescaler
Counter prescaler
Delay prescaler(1)
Counter clock (CLK_TCD_CNT)
Synchronizer clock (CLK_TCD_SYNC)
Delay clock (CLK_TCD_DLY)

1. Used by input blanking/delay event out

The TCD synchronizer clock is separate from the other module clocks enabling faster synchronization between the TCD domain and the IO domain. The total prescalingfor the counter is:

$$\mathrm{SYNCPRESC\_division\_factor} \times \mathrm{CNTPRESC\_division\_factor}$$

The Delay prescaler is used to prescale the clock used for the input blanking/ delayed event output functionality. The prescaler can be configured independently allowing separate range and accuracy settings from the counter functionality.

22.3.2.3 Waveform Generation Modes

The TCD provides four different waveform generation modes. The waveform generation modes determine how the counter is counting during a TCD cycle, and when the compare values are matching. A TCD cycle is split into these states:

- Dead time WOA (DTA)
- On time WOA (OTA)
- Dead time WOB (DTB)
- On time WOB (OTB)

In a standard configuration all states are present in the order they are listed and they are non-overlapping. The compare values Compare A Set (TCD.CMPASET), Compare A Clear (TCD.CMPACLR), Compare B Set (TCD.CMPBSET) and Compare B Clear (TCD.CMPBCLR) defines when each of the states are ending and the next is beginning. There are four different ways to go through a TCD cycle. The different ways are called Waveform Generation Modes. They are controlled by the Waveform Generation Mode bits (WGMODE) in the Control A register (TCD.CTRLA). The waveform generation modes are:

- One Ramp mode
- Two Ramp mode
- Four Ramp mode
- Dual Slope mode

The name indicates how the counter is operating during one TCDcycle.

One Ramp Mode

In One Ramp Mode, TCD counter counts up until it reaches the CMPBCLR value. Then the TCD cycle is done and the counter restarts from 0x000, beginning a new TCD cycle. The TCD cycle period is

$$T_{\mathrm{TCD\_cycle}} = \frac{(\mathrm{CMPBCLR} + 1)}{f_{\mathrm{CLK\_TCD\_CNT}}}$$

Offset	Name	Bit Pos.								
0x2A	CMPACLR	7:0	CMPCLR[7:0]							
0x2B		15:8					CMPCLR[11:8]			
0x2C	CMPBSET	7:0	CMPSET[7:0]							
0x2D		15:8					CMPSET[11:8]			
0x2E	CMPBCLR	7:0	CMPCLR[7:0]							
0x2F		15:8					CMPCLR[11:8]			

22.5 Register Description

22.5.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: Enable-protected

Bit	7	6	5	4	3	2	1	0
		CLKSEL[1:0]		CNTPRES[1:0]		SYNCPRES[1:0]		ENABLE
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 6:5 – CLKSEL[1:0]: Clock Select
The clock select bits select the clock source of the TCD clock.

Value	Description
0x0	OSC20M
0x1	Reserved
0x2	External clock
0x3	System clock

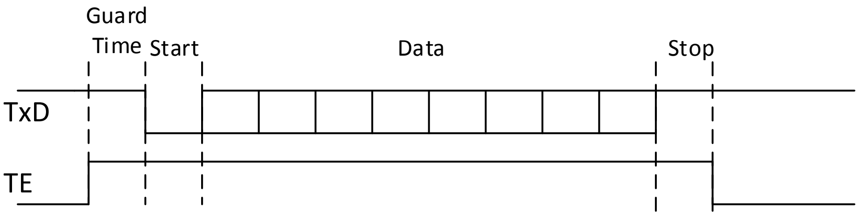
Bits 4:3 – CNTPRES[1:0]: Counter Prescaler
The Counter Prescaler bits select the division factor of the TCD counter clock.

Value	Description
0x0	Division factor 1
0x1	Division factor 4
0x2	Division factor 32
0x3	Reserved

Bits 2:1 – SYNCPRES[1:0]: Synchronization Prescaler
The synchronization prescaler bits select the division factor of the TCD clock.

Value	Description
0x0	Division factor 1
0x1	Division factor 2
0x2	Division factor 4
0x3	Division factor 8

Figure 24-11. TE Drive Timing



Related Links

Signal Description

24.3.2.7 Start Frame Detection

The start frame detection is supported in UART mode only. The UART start frame detector is limited to Standby sleep mode only and can wake up the system when a start bit is detected.

When a high-to-low transition is detected on RxDn, the oscillator is powered up and the UART 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 oscillator start-up time. Start-up time of the oscillators varies with supply voltage and temperature. For details on oscillator start-up time characteristics, refer to the Electrical Characteristics.

If a false start bit is detected and if the system has not been waken-up by another source, the clock will automatically be turned off and the UART waits for the next transition.

The UART start frame detection works in asynchronous mode only. It is enabled by writing the Start Frame Detection bit (SFDEN) in CTRLB. If the start bit is detected while the device is in Standby sleep mode, the UART Start Interrupt Flag (RXSIF) bit is set.

In Active, Idle and Power Down sleep modes, the asynchronous detection is automatically disabled.

The UART receive complete flag and UART start interrupt flag share the same interrupt line, but each has its dedicated interrupt settings. The Table 21-5 shows the USART start frame detection modes, depending of interrupt setting.

Table 24-7. USART Start Frame Detection Modes

SFDEN	RXSIF interrupt	RXCIF interrupt	Comment
0	x	x	Standard mode
1	Disabled	Disabled	Only the oscillator is powered during the frame reception. If the interrupts are disabled and buffer overflow is ignored, all incoming frames will be lost
1 [(1)]	Disabled	Enabled	System/all clocks waked-up on Receive Complete interrupt
1 [(1)]	Enabled	x	System/all clocks waked-up on UART Start Detection

Note:

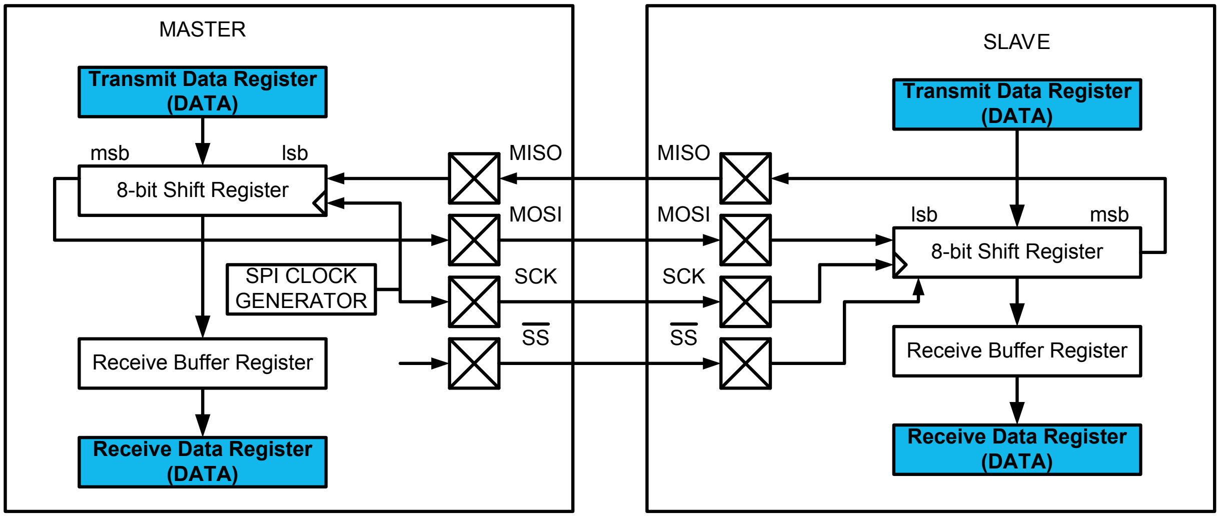
1. The SLEEP instruction will not shut down the oscillator if there is ongoing communication.

24.3.2.8 Break Character Detection and Auto-baud

When USART receive mode is set to LINAUTO mode (RXMODE in USART.CTRLB), it follows the LIN format. All LIN Frames start with a Break Field followed by a Sync Field. 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

25.2.1 Block Diagram

Figure 25-1. SPI Block Diagram



25.2.2 Signal Description

Table 25-1. Signals in Master and Slave Mode

Signal	Description	Type	
		Master Mode	Slave Mode
MOSI	Master Out Slave In	User defined output	Input
MISO	Master In Slave Out	Input	User defined output
SCK	Slave clock (generated by master)	User defined output	Input
$\overline{SS}$	Slave select (generated by master)	User defined output	Input

Related Links

I/O Multiplexing and Considerations

25.2.3 Product Dependencies

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

Table 25-2. SPI Product Dependencies

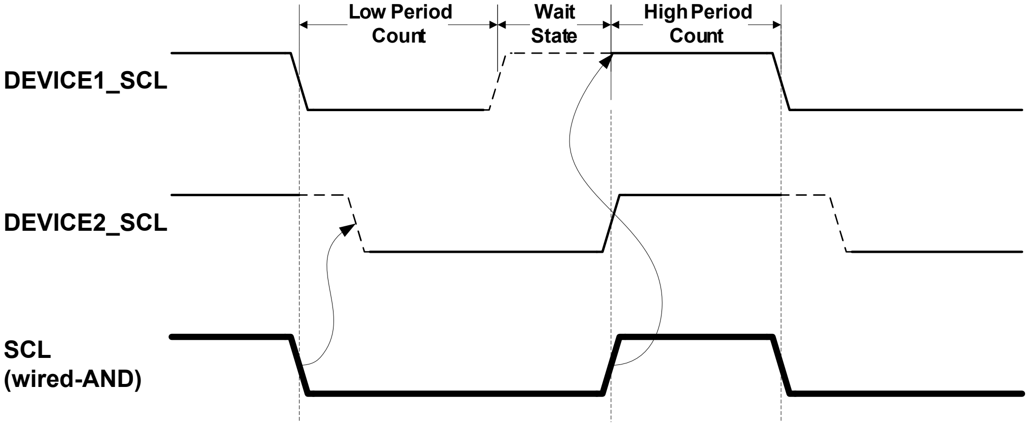
Dependency	Applicable	Peripheral
Clocks	Yes	CLKCTRL
I/O Lines and Connections	Yes	PORT
Interrupts	Yes	CPUINT
Events	No	-
Debug	Yes	UPDI

Related Links

26.3.2.8 Synchronization

A clock synchronization algorithm is necessary for solving situations where more than one master is trying to control the SCL line at the same time. The algorithm is based on the same principles used for the clock stretching previously described. Figure 26-11shows an example where two masters are competing for control over the bus clock. The SCL line is the wired-AND result of the two masters clock outputs.

Figure 26-11. Clock Synchronization



A high-to-low transition on the SCL line will force the line low for all masters on the bus, and they will start timing their low clock period. The timing length of the low clock period can vary among the masters. When a master (DEVICE1 in this case) has completed its low period, it releases the SCL line. However, the SCL line will not go high until all masters have released it. Consequently, the SCL line will be held low by the device with the longest low period (DEVICE2). Devices with shorter low periods must insert a wait state until the clock is released. All masters start their high period when the SCL line is released by all devices and has gone high. The device which first completes its high period (DEVICE1) forces the clock line low, and the procedure is then repeated. The result is that the device with the shortest clock period determines the high period, while the low period of the clock is determined by the device with the longest clock period.

26.3.3 TWI Bus State Logic

The bus state logic continuously monitors the activity on the TWI bus lines when the master is enabled. It continues to operate in all sleep modes, including power-down.

The bus state logic includes START and STOP condition detectors, collision detection, inactive bus timeout detection, and a bit counter. These are used to determine the bus state. Software can get the current bus state by reading the bus state bits in the master status register. The bus state can be unknown, idle, busy, or owner, and is determined according to the state diagram shown in Figure 26-12. The values of the bus state bits according to state are shown in binary in the figure.

Bit 7 – RIF: Read Interrupt Flag
This bit is set to '1' when the master byte read operation is successfully completed, i.e. no arbitration lost or bus error occurred during the operation. The read operation is triggered by software reading DATA or writing to ADDR registers with bit ADDR[0] written to '1'. A slave device must have responded with an ACK to the address and direction byte transmitted by the master for this flag to be set.

Writing a '1' to this bit will clear the RIF. However, normal use of the TWI does not require the flag to be cleared by this method.

Clearing the RIF bit will follow the same software interaction as the CLKHOLD flag.

The RIF flag can generate a master read interrupt (see description of the RIEN control bit in the TWI.MCTRLA register).

Bit 6 – WIF: Write Interrupt Flag
This bit is set when a master transmit address or byte write is completed, regardless of the occurrence of a bus error or an arbitration lost condition.

Writing a '1' to this bit will clear the WIF. However, normal use of the TWI does not require the flag to be cleared by this method.

Clearing the WIF bit will follow the same software interaction as the CLKHOLD flag.

The WIF flag can generate a master write interrupt (see description of the WIEN control bit in the TWI.MCTRLA register).

Bit 5 – CLKHOLD: Clock Hold
If read as '1', this bit indicates that the master is currently holding the TWI clock (SCL) low, stretching the TWI clock period.

Writing a '1' to this bit will clear the CLKHOLD flag. However, normal use of the TWI does not require the CLKHOLD flag to be cleared by this method, since the flag is automatically cleared when accessing several other TWI registers. The CLKHOLD flag can be cleared by:

1. Writing a '1' to it.
2. Writing to the TWI.MADDR register.
3. Writing to the TWI.MDATA register.
4. Reading the TWI.DATA register while the ACKACT control bits in TWI.MCTRLB are set to either send ACK or NACK.
5. Writing a valid command to the TWI.MCTRLB register.

Bit 4 – RXACK: Received Acknowledge
This bit is read-only and contains to the most recently received acknowledge bit from slave.

Bit 3 – ARBLOST: Arbitration Lost
If read as '1' this bit indicates that the master has lost arbitration while transmitting a high data or NACK bit, or while issuing a start or repeated start condition (S/Sr) on the bus.

Writing a '1' to it will clear the ARBLOST flag. However, normal use of the TWI does not require the flag to be cleared by this method. However, as for the CLKHOLD flag, clearing the ARBLOST flag is not required during normal use of the TWI.

Clearing the ARBLOST bit will follow the same software interaction as the CLKHOLD flag.

Given the condition where the bus ownership is lost to another master, the software must either abort operation or resend the data packet. Either way, the next required software interaction is in both cases to

Bits 7:0 – DATA[7:0]: Data
The bit field gives direct access to the masters physical shift register which is used both to shift data out onto the bus (write) and to shift in data received from the bus (read).

The direct access implies that the data register cannot be accessed during byte transmissions. Build-in logic prevents any write access to this register during the shift operations. Reading valid data or writing data to be transmitted can only be successfully done when the bus clock (SCL) is held low by the master, i.e. when the CLKHOLD bit in the Master Status register (TWI.MSTATUS) is set. However, it is not necessary to check the CLKHOLD bit in software before accessing this register if the software keeps track of the present protocol state by using interrupts or observing the interrupt flags.

Accessing this register assumes that the master clock hold is active, auto-triggers bus operations dependent of the state of the acknowledge action command bit (ACKACT) in TWI.MSTATUS and type of register access (read or write).

A write access to this register will, independent of ACKACT in TWI.MSTATUS, command the master to perform a byte transmit operation on the bus directly followed by receiving the acknowledge bit from the slave. When the acknowledge bit is received, the Master Write Interrupt Flag (WIF) in TWI.MSTATUS is set regardless of any bus errors or arbitration. If operating in a multi-master environment, the interrupt handler or application software must check the Arbitration Lost Status Flag (ARBLOST) in TWI.MSTATUS before continuing from this point. If the arbitration was lost, the application software must decide to either abort or to resend the packet by rewriting this register. The entire operation is performed (i.e. all bits are clocked), regardless of winning or losing arbitration before the write interrupt flag is set. When arbitration is lost, only '1's are transmitted for the remainder of the operation, followed by a write interrupt with ARBLOST flag set.

Both TWI master interrupt flags are cleared automatically when this register is written. However, the Master Arbitration Lost and Bus Error flags are left unchanged.

Reading this register triggers a bus operation, dependent on the setting of the acknowledge action command bit (ACKACT) in TWI.MSTATUS. Normally the ACKACT bit is preset to either ACK or NACK before the register read operation. If ACK or NACK action is selected, the transmission of the acknowledge bit precedes the release of the clock hold. The clock is released for one byte, allowing the slave to put one byte of data on the bus. The Master Read Interrupt flag RIF in TWI.MSTATUS is then set if the procedure was successfully executed. However, if arbitration was lost when sending NACK, or a bus error occurred during the time of operation, the Master Write Interrupt flag (WIF) is set instead. Observe that the two master interrupt flags are mutual exclusive, i.e. both flags will not be set simultaneously.

Both TWI master interrupt flags are cleared automatically if this register is read while ACKACT is set to either ACK or NACK. However, arbitration lost and bus error flags are left unchanged.

26.5.9 Slave Control A

Name: SCTRLA
Offset: 0x09
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	DIEN	APIEN	PIEN			PMEN	SMEN	ENABLE
Access	R/W	R/W	R/W			R/W	R/W	R/W
Reset	0	0	0			0	0	0

Bit 1 – DIR: Read/Write Direction
This bit is read only and indicates the current bus direction state. The DIR bit reflects the direction bit value from the last address packet received from a master TWI device. If this bit is read as '1', a master read operation is in progress. Consequently a '0' indicates that a master write operation is in progress.

Bit 0 – AP: Address or Stop
When the TWI slave address or stop interrupt flag (APIF) is set, this bit determines whether the interrupt is due to address detection or a stop condition.

Value	Name	Description
0	STOP	A stop condition generated the interrupt on APIF.
1	ADR	Address detection generated the interrupt on APIF.

26.5.12 Slave Address

Name: SADDR
Offset: 0x0C
Reset: 0x00
Property: -

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 7:0 – ADDR[7:0]: Address
The slave address register in combination with the slave address mask register (TWI.SADDRMASK) is used by the slave address match logic to determine if a master TWI device has addressed the TWI slave. The slave address interrupt flag (APIF) is set to one if the received address is recognized. The slave address match logic supports recognition of 7- and 10-bits addresses, and general call address.

When using 7-bit or 10-bit address recognition mode, the upper 7-bits of the address register (ADDR[7:1]) represents the slave address and the least significant bit (ADDR[0]) is used for general call address recognition. Setting the ADDR[0] bit in this case enables the general call address recognition logic. The TWI slave address match logic only support recognition of the first byte of a 10-bit address i.e. by setting ADDRA[7:1] = “0b11110aa” where “aa” represents bit 9 and 8 or the slave address. The second 10-bit address byte must be handled by software.

26.5.13 Slave Data

Name: SDATA
Offset: 0x0D
Reset: 0x00
Property: -

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

Gated D-Latch (DLATCH)

The D-input is driven by the even LUT output (LUT0), and the G-input is driven by the odd LUT output (LUT1).

Figure 28-9. D-Latch

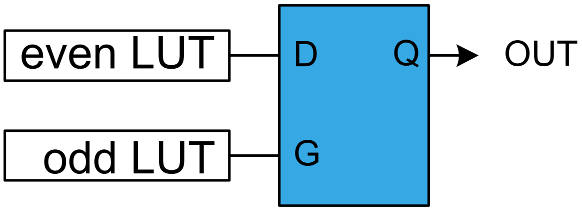


Table 28-5. D-Latch Characteristics

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

RS Latch (RS)

The S-input is driven by the even LUT output (LUT0), and the R-input is driven by the odd LUT output (LUT1).

Figure 28-10. RS-Latch

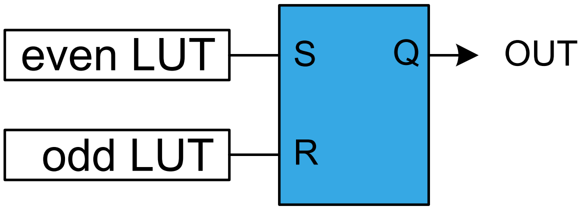


Table 28-6. RS-latch Characteristics

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

28.3.2.7 Clock Source Settings

The Filter, Edge Detector and Sequential logic are by default clocked by the system clock (CLK_PER). It is also possible to use the LUT input 2 (IN[2]) to clock these blocks (CLK_MUX_OUT in figure Figure 28-11). This is configured by writing the Clock Source bit (CLKSRC) in the LUT Control A register (CCL.LUTnCTRLA) to '1'.

Value	Name	Description
0x8	TCA0	TCA WO1 input source
0x9	TCD0	TCD WOB input source
0xA	USART0	USART TXD input source
0xB	SPI0	SPI MOSI input source
0xC	AC1	AC1 OUT input source
0xD	TCB1	TCB 1 W0 input source
0xE	AC2	AC2 OUT input source

Bits 3:0 – INSEL0[3:0]: LUT n Input 0 Source Selection
These bits select the source for input 0 of LUT n:

Value	Name	Description
0x0	MASK	Masked input
0x1	FEEDBACK	Feedback input
0x2	LINK	Linked other LUT as input source
0x3	EVENT0/EVENT1	Event 0 as input source for LUT0 / Event 1 as input source for LUT1
0x4	EVENT2/EVENT3	Event 2 as input source for LUT0 / Event 3 as input source for LUT1
0x5	IO	I/O pin LUTn-IN0 input source
0x6	AC0	AC0 OUT input source
0x7	TCB0	TCB WO input source
0x8	TCA0	TCA WO0 input source
0x9	TCD0	TCD WOA input source
0xA	USART0	USART XCK input source
0xB	SPI0	SPI SCK input source
0xC	AC1	AC1 OUT input source
0xD	TCB1	TCB 1 W0 input source
0xE	AC2	AC2 OUT input source
Other	-	Reserved

28.5.5 LUT n Control C

Name: LUT0CTRLC, LUT1CTRLC
Offset: 0x07 + n*0x04 [n=0..1]
Reset: 0x00
Property: Enable-Protected

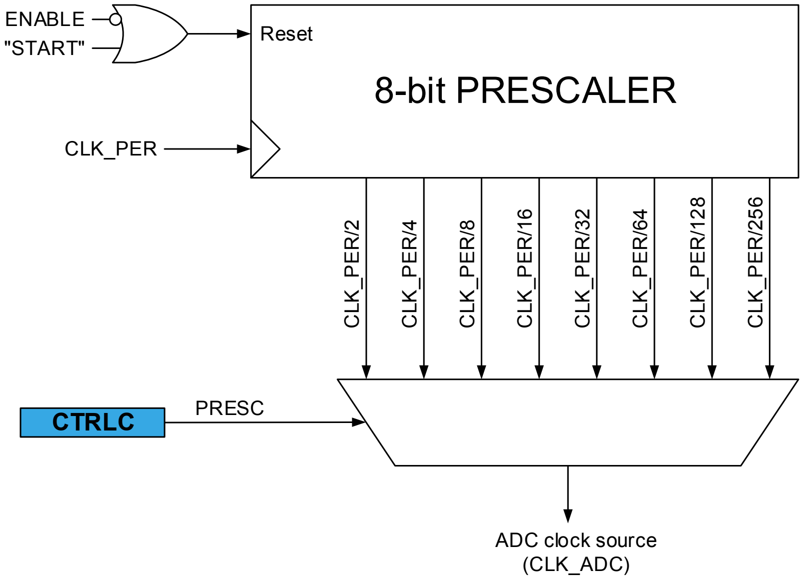
Bit	7	6	5	4	3	2	1	0
					INSEL2[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 3:0 – INSEL2[3:0]: LUT n Input 2 Source Selection
These bits select the source for input 2 of LUT n:

Value	Name	Description
0x0	MASK	Masked input
0x1	FEEDBACK	Feedback input
0x2	LINK	Linked other LUT as input source

30.3.2.2 Clock generation

Figure 30-6. ADC Prescaler



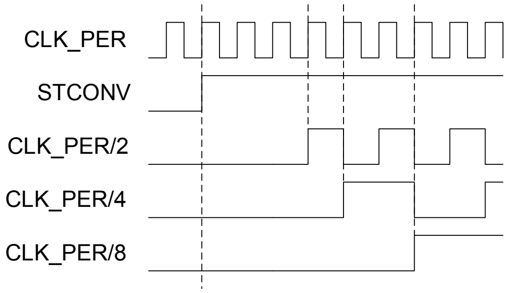
The ADC requires an input clock frequency between 50kHz and 2MHz for maximum resolution. If a lower resolution than 10 bits is selected, the input clock frequency to the ADC can be higher than 2MHz to get a higher sample rate.

The ADC module contains a prescaler which generates the ADC clock (CLK_ADC) from any CPU clock (CLK_PER) above 100kHz. The prescaling is selected by writing to the Prescaler bits (PRESC) in the Control C register (ADC.CTRLC). The prescaler starts counting from the moment the ADC is switched on by writing a '1' to the ENABLE bit in ADC.CTRLA. The prescaler keeps running as long as the ENABLE bit is one, the prescaler counter is reset to zero when the ENABLE bit is zero.

When initiating a conversion by writing a '1' to the Start Conversion bit (STCONV) in the Command register (ADC.COMMAND) or from event, the conversion starts at the following rising edge of the CLK_ADC clock cycle. The prescaler is kept reset as long as there is no ongoing conversion. This assures a fixed delay of of from the trigger to the actual start of a conversion given by in CLK_PER cycles are given by

$$\mathrm{StartDelay} = \frac{\mathrm{PRESC}_{\mathrm{factor}}}{2} + 2$$

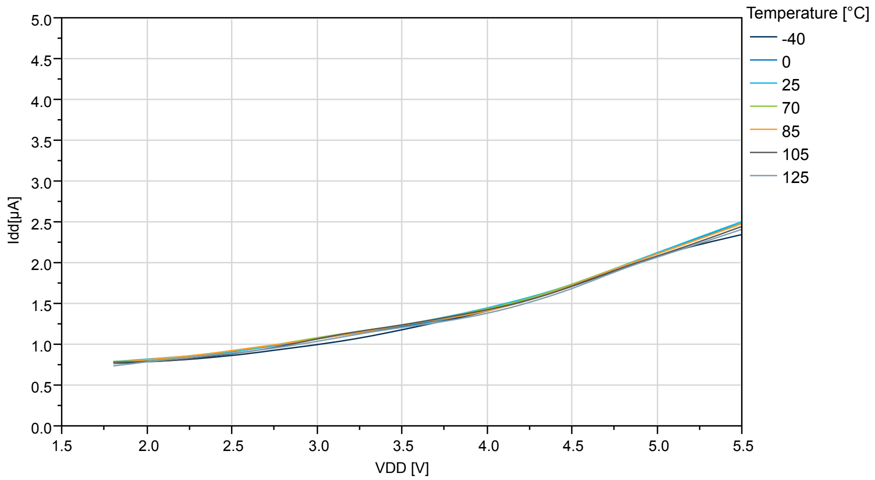
Figure 30-7. Start conversion and clock generation



30.3.2.3 Conversion timing

A normal conversion takes 13 CLK_ADC cycles. The actual sample-and-hold takes place 2 CLK_ADC cycles after the start of a conversion. Start of conversion is initiated by writing a '1' to the STCONV bit in ADC.COMMAND. When a conversion is complete, the result is available in the Result register (ADC.RES), and the Result Ready interrupt flag is set (RESRDY in ADC.INTFLAG). The interrupt flag will

Figure 38-37. ATtiny1614/1616/1617: BOD Current vs. V_{DD} (Sampled BOD at 1kHz)



BOD Threshold vs. Temperature

Figure 38-38. ATtiny1614/1616/1617: BOD Threshold vs. Temperature (Level 1.8V)

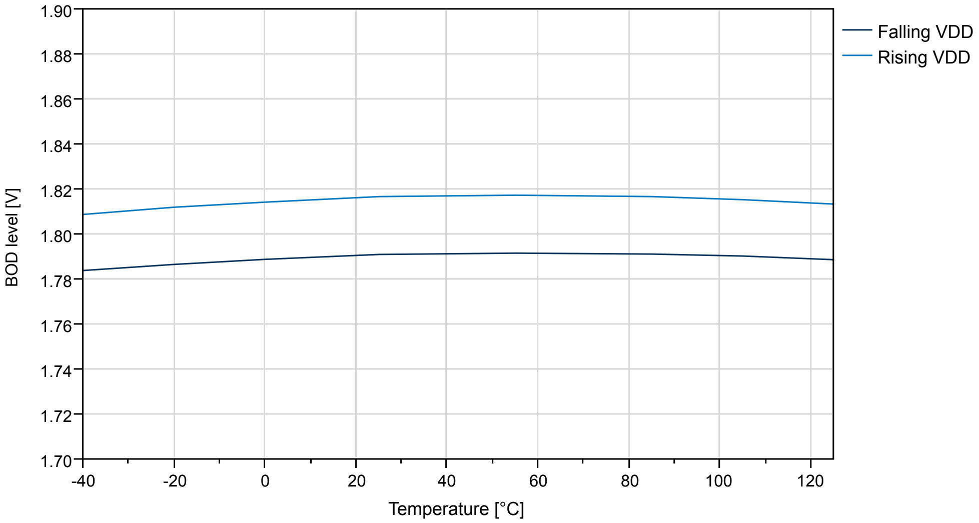
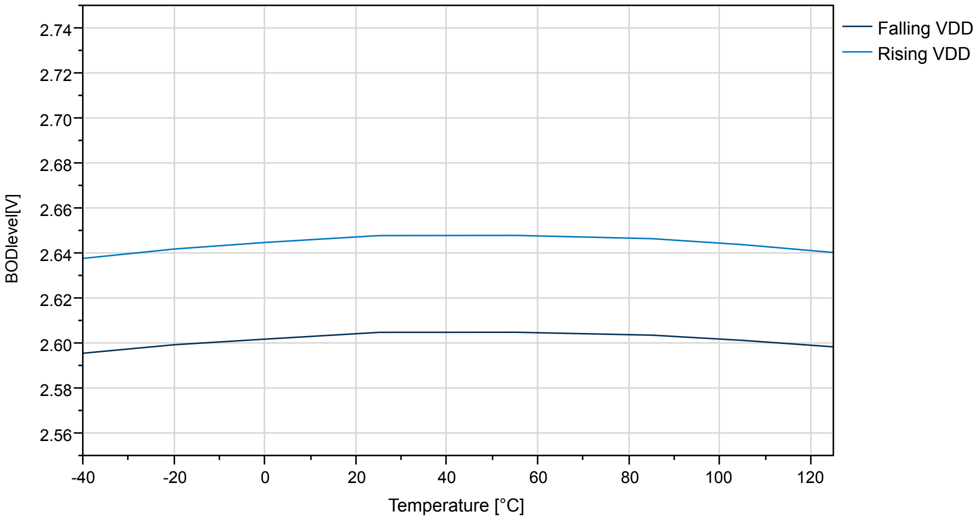


Figure 38-39. ATtiny1614/1616/1617: BOD Threshold vs. Temperature (Level 2.6V)



Trademarks

ISBN: 978-1-5224-1459-9

Quality Management System Certified by DNV

ISO/TS 16949
Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company’s quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip’s quality system for the design and manufacture of development systems is ISO 9001:2000 certified.