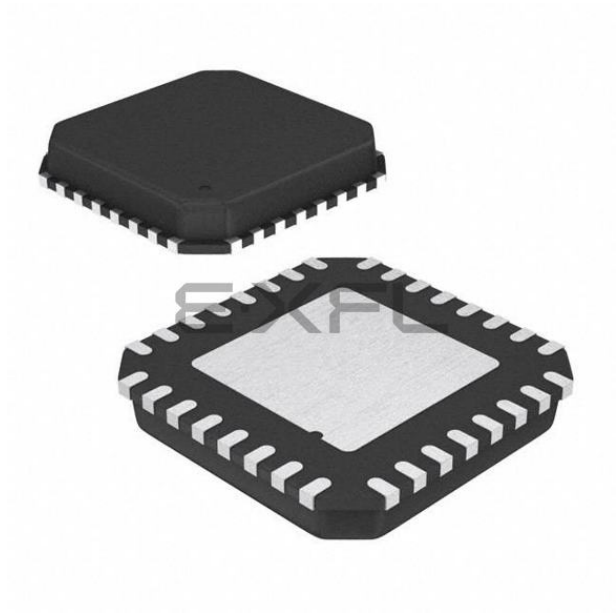


Welcome to E-XFL.COM

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, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	23
Program Memory Size	16KB (8K x 16)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	32-VFQFN Exposed Pad
Supplier Device Package	32-VQFN (5x5)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atmega168p-20mq

2. Configuration Summary

Features	ATmega48P/PV /88P/PV /168P/PV
Pin Count	28/32
Flash (Bytes)	4K/8K/16K
SRAM (Bytes)	512/1K/1K
EEPROM (Bytes)	256/512/512
Interrupt Vector Size (instruction word/vector)	1/1/2
General Purpose I/O Lines	23
SPI	2
TWI (I^2C)	1
USART	1
ADC	10-bit 15kSPS
ADC Channels	8
8-bit Timer/Counters	2
16-bit Timer/Counters	1

ATmega88P/PV and ATmega168P/PV support a real Read-While-Write Self-Programming mechanism. There is a separate Boot Loader Section, and the SPM instruction can only execute from there. In ATmega48P/PV, there is no Read-While-Write support and no separate Boot Loader Section. The SPM instruction can execute from the entire Flash.

Bit 0 – C: Carry Flag

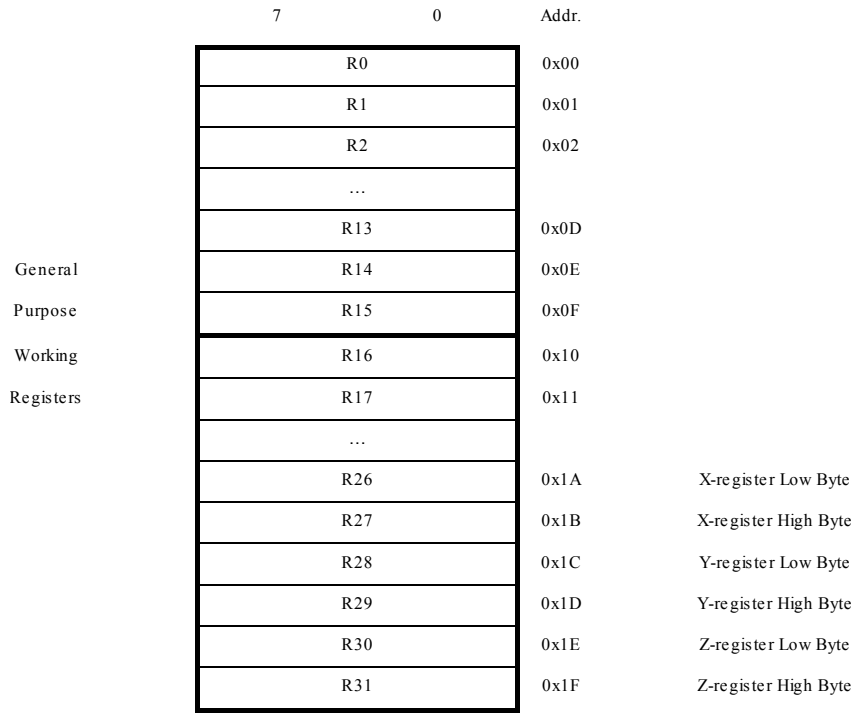
The Carry Flag C indicates a carry in an arithmetic or logic operation. See the *Instruction Set Description* for detailed information.

11.4. General Purpose Register File

The Register File is optimized for the AVR Enhanced RISC instruction set. In order to achieve the required performance and flexibility, the following input/output schemes are supported by the Register File:

- One 8-bit output operand and one 8-bit result input
- Two 8-bit output operands and one 8-bit result input
- Two 8-bit output operands and one 16-bit result input
- One 16-bit output operand and one 16-bit result input

Figure 11-2. AVR CPU General Purpose Working Registers

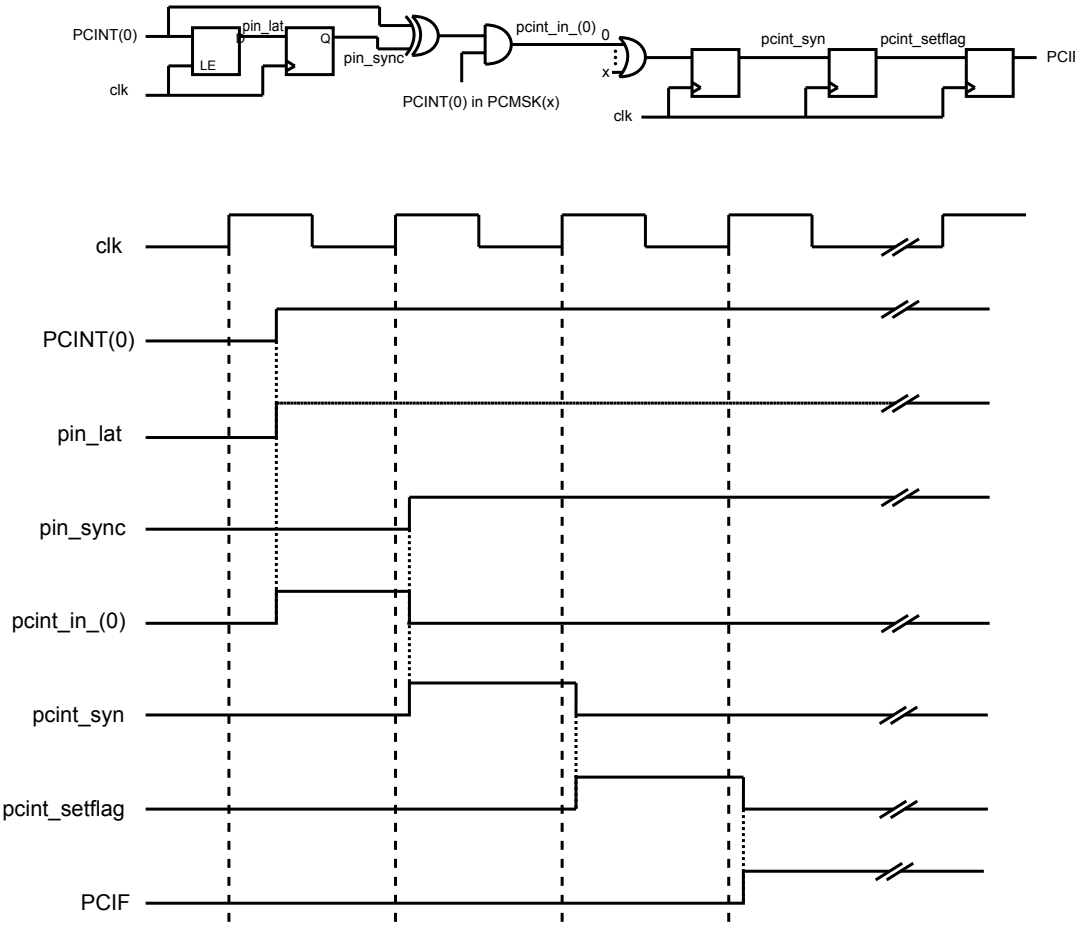


Most of the instructions operating on the Register File have direct access to all registers, and most of them are single cycle instructions. As shown in the figure, each register is also assigned a data memory address, mapping them directly into the first 32 locations of the user Data Space. Although not being physically implemented as SRAM locations, this memory organization provides great flexibility in access of the registers, as the X-, Y-, and Z-pointer registers can be set to index any register in the file.

11.4.1. The X-register, Y-register, and Z-register

The registers R26...R31 have some added functions to their general purpose usage. These registers are 16-bit address pointers for indirect addressing of the data space. The three indirect address registers X, Y, and Z are defined as described in the figure.

Figure 17-1. Timing of pin change interrupts



Related Links

System Control and Reset on page 73

Clock Systems and Their Distribution on page 50

System Clock and Clock Options on page 50

17.2. Register Description

17.2.1. External Interrupt Control Register A

The External Interrupt Control Register A contains control bits for interrupt sense control.

Name: EICRA
Offset: 0x69
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
					ISC11	ISC10	ISC01	ISC00
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 3:2 – ISC1n: Interrupt Sense Control 1 [n = 1:0]

The External Interrupt 1 is activated by the external pin INT1 if the SREG I-flag and the corresponding interrupt mask are set. The level and edges on the external INT1 pin that activate the interrupt are defined in the table below. The value on the INT1 pin is sampled before detecting edges. If edge or toggle interrupt is selected, pulses that last longer than one clock period will generate an interrupt. Shorter pulses are not guaranteed to generate an interrupt. If low level interrupt is selected, the low level must be held until the completion of the currently executing instruction to generate an interrupt.

Value	Description
00	The low level of INT1 generates an interrupt request.
01	Any logical change on INT1 generates an interrupt request.
10	The falling edge of INT1 generates an interrupt request.
11	The rising edge of INT1 generates an interrupt request.

Bits 1:0 – ISC0n: Interrupt Sense Control 0 [n = 1:0]

The External Interrupt 0 is activated by the external pin INT0 if the SREG I-flag and the corresponding interrupt mask are set. The level and edges on the external INT0 pin that activate the interrupt are defined in table below. The value on the INT0 pin is sampled before detecting edges. If edge or toggle interrupt is selected, pulses that last longer than one clock period will generate an interrupt. Shorter pulses are not guaranteed to generate an interrupt. If low level interrupt is selected, the low level must be held until the completion of the currently executing instruction to generate an interrupt.

Value	Description
00	The low level of INT0 generates an interrupt request.
01	Any logical change on INT0 generates an interrupt request.
10	The falling edge of INT0 generates an interrupt request.
11	The rising edge of INT0 generates an interrupt request.

- ADC2/PCINT10 – Port C, Bit 2
  - PC2 can also be used as ADC input Channel 2. The ADC input channel 2 uses analog power.
  - PCINT10: Pin Change Interrupt source 10. The PC2 pin can serve as an external interrupt source.
- ADC1/PCINT9 – Port C, Bit 1
  - PC1 can also be used as ADC input Channel 1. The ADC input channel 1 uses analog power.
  - PCINT9: Pin Change Interrupt source 9. The PC1 pin can serve as an external interrupt source.
- ADC0//CINT8 – Port C, Bit 0
  - PC0 can also be used as ADC input Channel 0. The ADC input channel 0 uses analog power.
  - PCINT8: Pin Change Interrupt source 8. The PC0 pin can serve as an external interrupt source.

The tables below relate the alternate functions of Port C to the overriding signals shown in Figure 18-5.

Table 18-7. Overriding Signals for Alternate Functions in PC6...PC4(1)

Signal Name	PC6/$\overline{\text{RESET}}$/PCINT14	PC5/SCL/ADC5/PCINT13	PC4/SDA/ADC4/PCINT12
PUOE	RSTDISBL	TWEN	TWEN
PUOV	1	PORTC5 • $\overline{\text{PUD}}$	PORTC4 • $\overline{\text{PUD}}$
DDOE	RSTDISBL	TWEN	TWEN
DDOV	0	SCL_OUT	SDA_OUT
PVOE	0	TWEN	TWEN
PVOV	0	0	0
DIEOE	RSTDISBL + PCINT14 • PCIE1	PCINT13 • PCIE1 + ADC5D	PCINT12 • PCIE1 + ADC4D
DIEOV	RSTDISBL	PCINT13 • PCIE1	PCINT12 • PCIE1
DI	PCINT14 INPUT	PCINT13 INPUT	PCINT12 INPUT
AIO	RESET INPUT	ADC5 INPUT / SCL INPUT	ADC4 INPUT / SDA INPUT

Note: 1. When enabled, the 2-wire Serial Interface enables slew-rate controls on the output pins PC4 and PC5. This is not shown in the figure. In addition, spike filters are connected between the AIO outputs shown in the port figure and the digital logic of the TWI module.

Table 18-8. Overriding Signals for Alternate Functions in PC3...PC0

Signal Name	PC3/ADC3/ PCINT11	PC2/ADC2/ PCINT10	PC1/ADC1/ PCINT9	PC0/ADC0/ PCINT8
PUOE	0	0	0	0
PUOV	0	0	0	0
DDOE	0	0	0	0
DDOV	0	0	0	0

19. TC0 - 8-bit Timer/Counter0 with PWM

Related Links

Timer/Counter0 and Timer/Counter1 Prescalers on page 192

19.1. Features

- Two independent Output Compare Units
- Double Buffered Output Compare Registers
- Clear Timer on Compare Match (Auto Reload)
- Glitch free, phase correct Pulse Width Modulator (PWM)
- Variable PWM period
- Frequency generator
- Three independent interrupt sources (TOV0, OCF0A, and OCF0B)

19.2. Overview

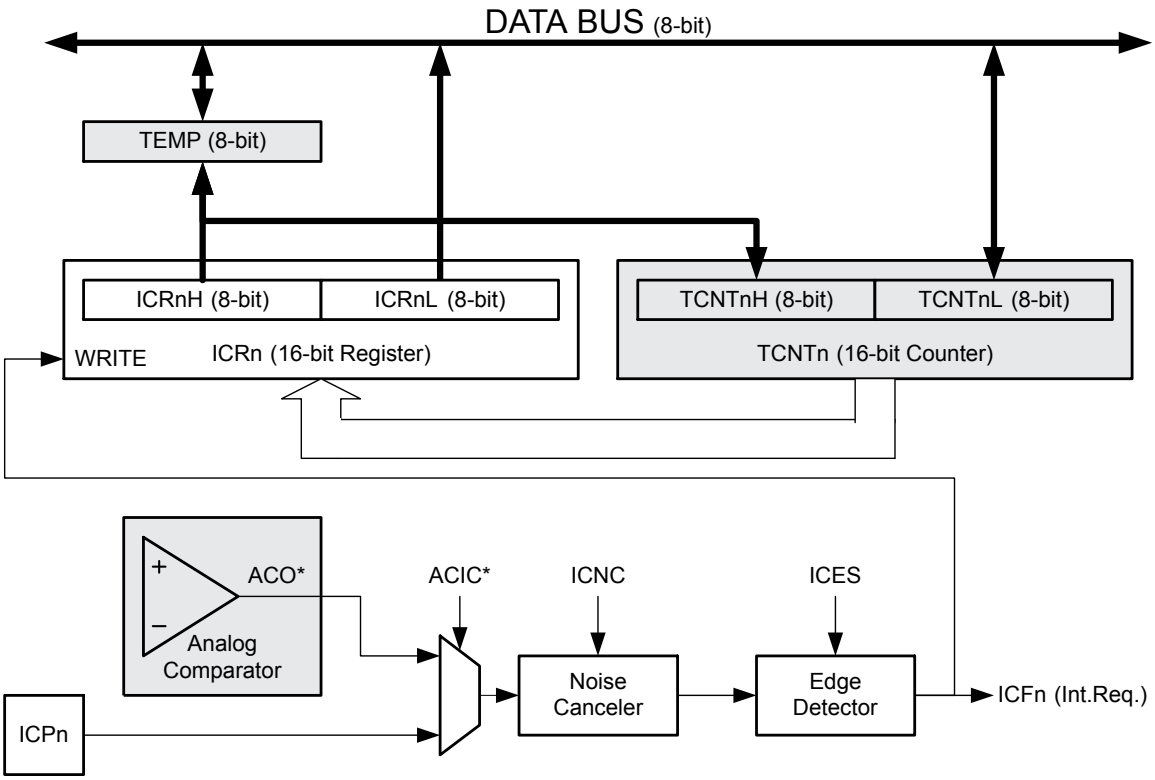
Timer/Counter0 (TC0) is a general purpose 8-bit Timer/Counter module, with two independent Output Compare Units, and PWM support. It allows accurate program execution timing (event management) and wave generation.

A simplified block diagram of the 8-bit Timer/Counter is shown below. CPU accessible I/O Registers, including I/O bits and I/O pins, are shown in bold. The device-specific I/O Register and bit locations are listed in the Register Description. For the actual placement of I/O pins, refer to the pinout diagram.

The TC0 is enabled by writing the PRTIM0 bit in ”Minimizing Power Consumption” to '0'.

The TC0 is enabled when the PRTIM0 bit in the Power Reduction Register (PRR.PRTIM0) is written to '1'.

Figure 20-3. Input Capture Unit Block Diagram for TC1



Note: The “n” in the register and bit names indicates the device number (n = 1 for Timer/Counter 1), and the “x” indicates Output Compare unit (A/B).

When a change of the logic level (an event) occurs on the Input Capture pin (ICP1), or alternatively on the Analog Comparator output (ACO), and this change confirms to the setting of the edge detector, a capture will be triggered: the 16-bit value of the counter (TCNT1) is written to the Input Capture Register (ICR1). The Input Capture Flag (ICF) is set at the same system clock cycle as the TCNT1 value is copied into the ICR1 Register. If enabled (TIMSK1.ICIE=1), the Input Capture Flag generates an Input Capture interrupt. The ICF1 Flag is automatically cleared when the interrupt is executed. Alternatively the ICF Flag can be cleared by software by writing '1' to its I/O bit location.

Reading the 16-bit value in the Input Capture Register (ICR1) is done by first reading the low byte (ICR1L) and then the high byte (ICR1H). When the low byte is read form ICR1L, the high byte is copied into the high byte temporary register (TEMP). When the CPU reads the ICR1H I/O location it will access the TEMP Register.

The ICR1 Register can only be written when using a Waveform Generation mode that utilizes the ICR1 Register for defining the counter’s TOP value. In these cases the Waveform Generation mode bits (WGM1[3:0]) must be set before the TOP value can be written to the ICR1 Register. When writing the ICR1 Register, the high byte must be written to the ICR1H I/O location before the low byte is written to ICR1L.

See also Accessing 16-bit Registers.

20.9.1. Input Capture Trigger Source

The main trigger source for the Input Capture unit is the Input Capture pin (ICP1). Timer/Counter1 can alternatively use the Analog Comparator output as trigger source for the Input Capture unit. The Analog Comparator is selected as trigger source by setting the Analog Comparator Input Capture (ACIC) bit in

Figure 22-2. Counter Unit Block Diagram

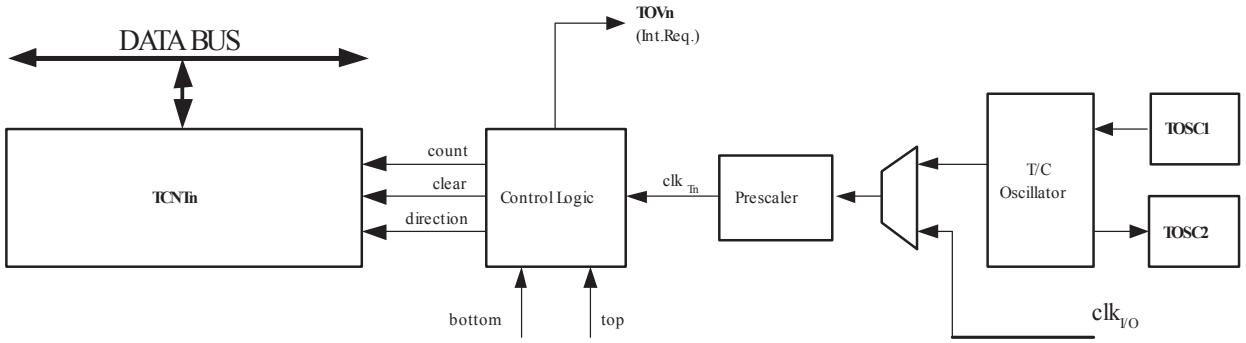


Table 22-2. Signal description (internal signals):

Signal name	Description
count	Increment or decrement TCNT2 by 1.
direction	Selects between increment and decrement.
clear	Clear TCNT2 (set all bits to zero).
clk_{Tn}	Timer/Counter clock, referred to as clk_{T2} in the following.
top	Signalizes that TCNT2 has reached maximum value.
bottom	Signalizes that TCNT2 has reached minimum value (zero).

Depending on the mode of operation used, the counter is cleared, incremented, or decremented at each timer clock (clk_{T2}). clk_{T2} can be generated from an external or internal clock source, selected by the Clock Select bits (CS2[2:0]). When no clock source is selected (CS2[2:0]=0x0) the timer is stopped. However, the TCNT2 value can be accessed by the CPU, regardless of whether clk_{T2} is present or not. A CPU write overrides (has priority over) all counter clear or count operations.

The counting sequence is determined by the setting of the WGM21 and WGM20 bits located in the Timer/Counter Control Register (TCCR2A) and the WGM22 bit located in the Timer/Counter Control Register B (TCCR2B). There are close connections between how the counter behaves (counts) and how waveforms are generated on the Output Compare outputs OC2A and OC2B. For more details about advanced counting sequences and waveform generation, see "Modes of Operation".

The Timer/Counter Overflow Flag (TOV2) is set according to the mode of operation selected by the TCC2B.WGM2[2:0] bits. TOV2 can be used for generating a CPU interrupt.

22.5. Output Compare Unit

The 8-bit comparator continuously compares TCNT2 with the Output Compare Register (OCR2A and OCR2B). Whenever TCNT2 equals OCR2A or OCR2B, the comparator signals a match. A match will set the Output Compare Flag (OCF2A or OCF2B) at the next timer clock cycle. If the corresponding interrupt is enabled, the Output Compare Flag generates an Output Compare interrupt. The Output Compare Flag is automatically cleared when the interrupt is executed. Alternatively, the Output Compare Flag can be cleared by software by writing a logical one to its I/O bit location. The Waveform Generator uses the match signal to generate an output according to operating mode set by the WGM2[2:0] bits and Compare Output mode (COM2x[1:0]) bits. The max and bottom signals are used by the Waveform Generator for handling the special cases of the extreme values in some modes of operation (See Modes of Operation).

The following figure shows a block diagram of the Output Compare unit.

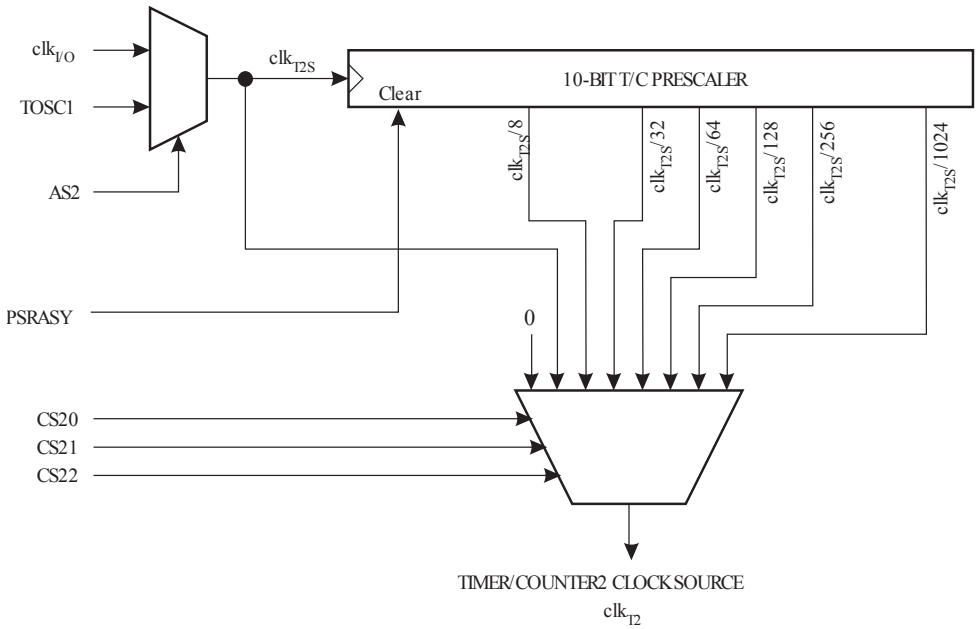
unpredictable, as it depends on the wake-up time. The recommended procedure for reading TCNT2 is thus as follows:

1. Wait for the corresponding Update Busy Flag to be cleared.
2. Read TCNT2.

- During asynchronous operation, the synchronization of the Interrupt Flags for the asynchronous timer takes 3 processor cycles plus one timer cycle. The timer is therefore advanced by at least one before the processor can read the timer value causing the setting of the Interrupt Flag. The Output Compare pin is changed on the timer clock and is not synchronized to the processor clock.

22.10. Timer/Counter Prescaler

Figure 22-12. Prescaler for TC2



The clock source for TC2 is named clk_{T2S}. It is by default connected to the main system I/O clock $clk_{I/O}$. By writing a '1' to the Asynchronous TC2 bit in the Asynchronous Status Register (ASSR.AS2), TC2 is asynchronously clocked from the TOSC1 pin. This enables use of TC2 as a Real Time Counter (RTC). When AS2 is set, pins TOSC1 and TOSC2 are disconnected from Port B. A crystal can then be connected between the TOSC1 and TOSC2 pins to serve as an independent clock source for TC2. The Oscillator is optimized for use with a 32.768kHz crystal.

For TC2, the possible prescaled selections are: $clk_{T2S}/8$, $clk_{T2S}/32$, $clk_{T2S}/64$, $clk_{T2S}/128$, $clk_{T2S}/256$, and $clk_{T2S}/1024$. Additionally, clk_{T2S} as well as 0 (stop) may be selected. The prescaler is reset by writing a '1' to the Prescaler Reset TC2 bit in the General TC2 Control Register (GTCCR.PSRASY). This allows the user to operate with a defined prescaler.

22.11. Register Description

BAUD	Baud rate (in bits per second, bps)
f_{OSC}	System oscillator clock frequency
UBRRn	Contents of the UBRRnH and UBRRnL Registers, (0-4095). Some examples of UBRRn values for some system clock frequencies are found in Examples of Baud Rate Settings.

24.4.2. Double Speed Operation (U2Xn)

The transfer rate can be doubled by setting the U2Xn bit in UCSRnA. Setting this bit only has effect for the asynchronous operation. Set this bit to zero when using synchronous operation.

Setting this bit will reduce the divisor of the baud rate divider from 16 to 8, effectively doubling the transfer rate for asynchronous communication. However, in this case, the Receiver will only use half the number of samples (reduced from 16 to 8) for data sampling and clock recovery, and therefore a more accurate baud rate setting and system clock are required when this mode is used.

For the Transmitter, there are no downsides.

24.4.3. External Clock

External clocking is used by the synchronous slave modes of operation. The description in this section refers to the Clock Generation Logic block diagram in the previous section.

External clock input from the XCKn pin is sampled by a synchronization register to minimize the chance of meta-stability. The output from the synchronization register must then pass through an edge detector before it can be used by the Transmitter and Receiver. This process introduces a two CPU clock period delay and therefore the maximum external XCKn clock frequency is limited by the following equation:

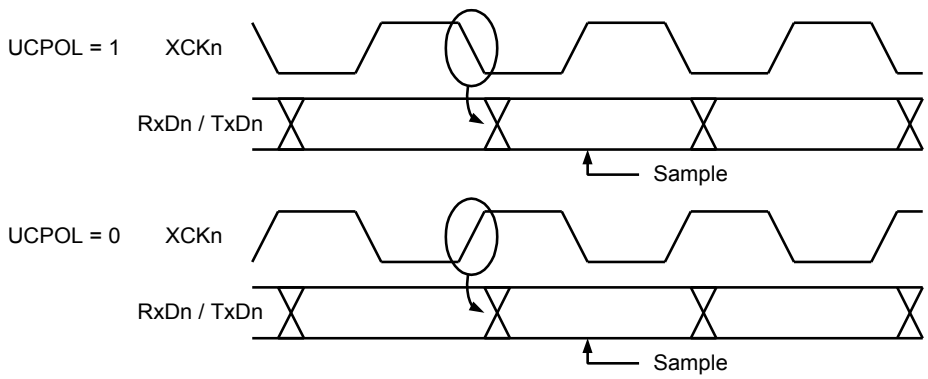
$$f_{\mathrm{XCKn}} < \frac{f_{\mathrm{OSC}}}{4}$$

The value of f_{osc} depends on the stability of the system clock source. It is therefore recommended to add some margin to avoid possible loss of data due to frequency variations.

24.4.4. Synchronous Clock Operation

When synchronous mode is used (UMSEL = 1), the XCKn pin will be used as either clock input (Slave) or clock output (Master). The dependency between the clock edges and data sampling or data change is the same. The basic principle is that data input (on RxDn) is sampled at the opposite XCKn clock edge of the edge the data output (TxDn) is changed.

Figure 24-3. Synchronous Mode XCKn Timing

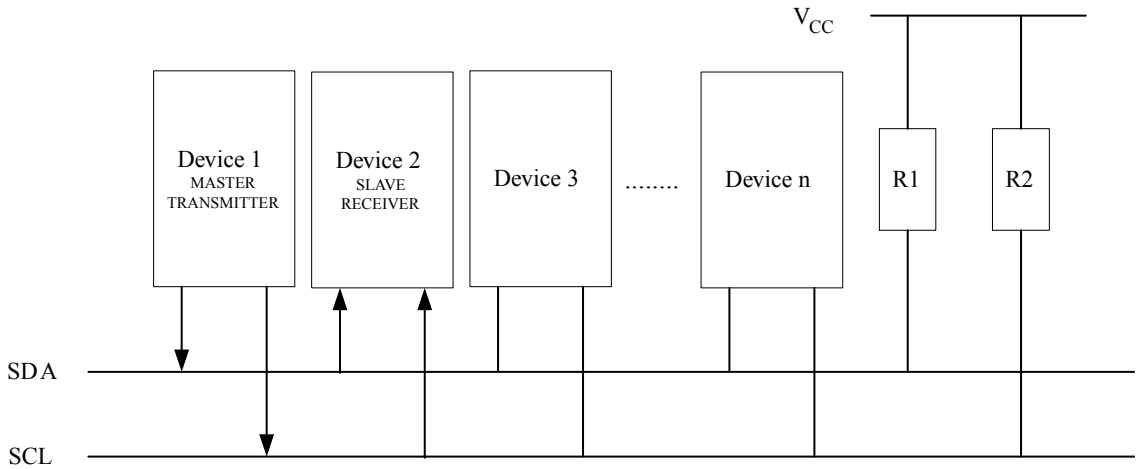


The UCPOL bit UCRSC selects which XCKn clock edge is used for data sampling and which is used for data change. As the above timing diagram shows, when UCPOL is zero, the data will be changed at

26.7.1. Master Transmitter Mode

In the Master Transmitter (MT) mode, a number of data bytes are transmitted to a Slave Receiver, see figure below. In order to enter a Master mode, a START condition must be transmitted. The format of the following address packet determines whether MT or Master Receiver (MR) mode is to be entered: If SLA +W is transmitted, MT mode is entered, if SLA+R is transmitted, MR mode is entered. All the status codes mentioned in this section assume that the prescaler bits are zero or are masked to zero.

Figure 26-11. Data Transfer in Master Transmitter Mode



A START condition is sent by writing a value to the TWI Control Register n (TWCRn) of the type TWCRn=1x10x10x:

- The TWI Enable bit (TWCRn.TWEN) must be written to '1' to enable the 2-wire Serial Interface
- The TWI Start Condition bit (TWCRn.TWSTA) must be written to '1' to transmit a START condition
- The TWI Interrupt Flag (TWCRn.TWINT) must be written to '1' to clear the flag.

The TWI n will then test the 2-wire Serial Bus and generate a START condition as soon as the bus becomes free. After a START condition has been transmitted, the TWINT Flag is set by hardware, and the status code in TWSRn will be 0x08 (see Status Code table below). In order to enter MT mode, SLA +W must be transmitted. This is done by writing SLA+W to the TWI Data Register (TWDRn). Thereafter, the TWCRn.TWINT Flag should be cleared (by writing a '1' to it) to continue the transfer. This is accomplished by writing a value to TWRC of the type TWCR=1x00x10x.

When SLA+W have been transmitted and an acknowledgment bit has been received, TWINT is set again and a number of status codes in TWSR are possible. Possible status codes in Master mode are 0x18, 0x20, or 0x38. The appropriate action to be taken for each of these status codes is detailed in the Status Code table below.

When SLA+W has been successfully transmitted, a data packet should be transmitted. This is done by writing the data byte to TWDR. TWDR must only be written when TWINT is high. If not, the access will be discarded, and the Write Collision bit (TWWC) will be set in the TWCRn Register. After updating TWDRn, the TWINT bit should be cleared (by writing '1' to it) to continue the transfer. This is accomplished by writing again a value to TWCRn of the type TWCRn=1x00x10x.

This scheme is repeated until the last byte has been sent and the transfer is ended, either by generating a STOP condition or a by a repeated START condition. A repeated START condition is accomplished by writing a regular START value TWCRn=1x10x10x. A STOP condition is generated by writing a value of the type TWCRn=1x01x10x.

After a repeated START condition (status code 0x10), the 2-wire Serial Interface can access the same Slave again, or a new Slave without transmitting a STOP condition. Repeated START enables the Master

```
          ld r1, Y+
          ldi spmcrval, (1<<SPMEN)
          rcall Do_spm
          adiw ZH:ZL, 2
          sbiw loophi:looplo, 2 ;use subi for PAGESIZEB<=256
          brne Wrloop

         ; execute Page Write
         subi ZL, low(PAGESIZEB)  ;restore pointer
         sbci ZH, high(PAGESIZEB) ;not required for PAGESIZEB<=256
         ldi spmcrval, (1<<PGWRT) | (1<<SPMEN)
 rcall Do_spm

         ; re-enable the RWW section
 ldi spmcrval, (1<<RWWSRE) | (1<<SPMEN)
         rcall Do_spm

         ; read back and check, optional
         ldi looplo, low(PAGESIZEB)  ;init loop variable
         ldi loophi, high(PAGESIZEB) ;not required for PAGESIZEB<=256
         subi YL, low(PAGESIZEB)     ;restore pointer
         sbci YH, high(PAGESIZEB)
Rdloop:
         lpm r0, Z+
         ld r1, Y+
         cpse r0, r1
         rjmp Error
         sbiw loophi:looplo, 1 ;use subi for PAGESIZEB<=256
         brne Rdloop

        ; return to RWW section
        ; verify that RWW section is safe to read
Return:
        in temp1, SPMCSR
        sbrs temp1, RWWSB ; If RWWSB is set, the RWW section is not ready yet
        ret
        ; re-enable the RWW section
        ldi spmcrval, (1<<RWWSRE) | (1<<SPMEN)
        rcall Do_spm
        rjmp Return
```

Bit 1 – PGERS: Page Erase

If this bit is written to one at the same time as SPMEN, the next SPM instruction within four clock cycles executes Page Erase. The page address is taken from the high part of the Z-pointer. The data in R1 and R0 are ignored. The PGERS bit will auto-clear upon completion of a Page Erase, or if no SPM instruction is executed within four clock cycles. The CPU is halted during the entire Page Write operation.

Bit 0 – SPMEN: Store Program Memory

This bit enables the SPM instruction for the next four clock cycles. If written to one together with either RWWSRE, BLBSET, PGWRT, or PGERS, the following SPM instruction will have a special meaning, see description above. If only SPMEN is written, the following SPM instruction will store the value in R1:R0 in the temporary page buffer addressed by the Z-pointer. The LSB of the Z-pointer is ignored. The SPMEN bit will auto-clear upon completion of an SPM instruction, or if no SPM instruction is executed within four clock cycles. During Page Erase and Page Write, the SPMEN bit remains high until the operation is completed.

Writing any other combination than “0x10001”, “0x01001”, “0x00101”, “0x00011” or “0x00001” in the lower five bits will have no effect.

Symbol	Parameter	Condition	Min.	Typ	Max	Units
SR_{ON}	Power-on Slope Rate		0.01	-	10	V/ms
V_{RST}	$\overline{\text{RESET}}$ Pin Threshold Voltage		0.2 V_{CC}	-	0.9 V_{CC}	V
t_{RST}	Minimum pulse width on $\overline{\text{RESET}}$ Pin		-	-	2.5	µs
V_{HYST}	Brown-out Detector Hysteresis		-	50	-	mV
t_{BOD}	Min. Pulse Width on Brown-out Reset		-	2	-	µs
V_{BG}	Bandgap reference voltage	V_{CC}=2.7 T_A=25°C	1.0	1.1	1.2	V
t_{BG}	Bandgap reference start-up time	V_{CC}=2.7 T_A=25°C	-	40	70	µs
I_{BG}	Bandgap reference current consumption	V_{CC}=2.7 T_A=25°C	-	10	-	µA

Note:

1. Values are guidelines only.
2. The Power-on Reset will not work unless the supply voltage has been below VPOT (falling)

Table 33-9. BODLEVEL Fuse Coding(1)

BODLEVEL [2:0] Fuses	Min. V_{BOT}	Typ. V_{BOT}	Max V_{BOT}	Units
111	BOD Disabled			
110	1.7	1.8	2.0	V
101	2.5	2.7	2.9	
100	4.1	4.3	4.5	
011	Reserved			
010				
001				
000				

Note: V_{BOT} may be below nominal minimum operating voltage for some devices. For devices where this is the case, the device is tested down to V_{CC} = V_{BOT} during the production test. This guarantees that a Brown-Out Reset will occur before V_{CC} drops to a voltage where correct operation of the microcontroller is no longer guaranteed. The test is performed using BODLEVEL = 101 and BODLEVEL = 100 for ATmega48P/88P/168P, and BODLEVEL = 110 and BODLEVEL = 101 for ATmega48PV/88PV/168PV.

Symbol	Parameter	Condition	Min.	Typ	Max	Units
	Input Bandwidth		-	38.5		kHz
V_{INT}	Internal Voltage Reference		1.0	1.1	1.2	V
R_{REF}	Reference Input Resistance		-	50	-	kΩ
R_{AIN}	Analog Input Resistance		-	100	-	MΩ

Note:

1. AV_{CC} absolute min./max: 1.8V/5.5V

33.9. Parallel Programming Characteristics

Table 33-13. Parallel Programming Characteristics, V_{CC} = 5V ± 10%

Symbol	Parameter	Min.	Max	Units
V_{PP}	Programming Enable Voltage	11.5	12.5	V
I_{PP}	Programming Enable Current	-	250	µA
t_{DVXH}	Data and Control Valid before XTAL1 High	67	-	ns
t_{XLXH}	XTAL1 Low to XTAL1 High	200	-	ns
t_{XHXL}	XTAL1 Pulse Width High	150	-	ns
t_{XLDX}	Data and Control Hold after XTAL1 Low	67	-	ns
t_{XLWL}	XTAL1 Low to $\overline{WR}$ Low	0	-	ns
t_{XLPH}	XTAL1 Low to PAGEL high	0	-	ns
t_{PLXH}	PAGEL low to XTAL1 high	150	-	ns
t_{BVPH}	BS1 Valid before PAGEL High	67	-	ns
t_{PHPL}	PAGEL Pulse Width High	150	-	ns
t_{PLBX}	BS1 Hold after PAGEL Low	67	-	ns
t_{WLBX}	BS2/1 Hold after RDY/$\overline{BSY}$ high	67	-	ns
t_{PLWL}	PAGEL Low to $\overline{WR}$ Low	67	-	ns
t_{BVWL}	BS1 Valid to $\overline{WR}$ Low	67	-	ns
t_{WLWH}	$\overline{WR}$ Pulse Width Low	150	-	ns
t_{WLRL}	$\overline{WR}$ Low to RDY/$\overline{BSY}$ Low	0	1	µs
t_{WLRH}	$\overline{WR}$ Low to RDY/$\overline{BSY}$ High(1)	3.7	4.5	ms
t_{WLRH_CE}	$\overline{WR}$ Low to RDY/$\overline{BSY}$ High for Chip Erase(2)	7.5	9	ms
t_{XLOL}	XTAL1 Low to $\overline{OE}$ Low	0	-	ns
t_{BVDV}	BS1 Valid to DATA valid	0	250	ns

Figure 34-78. BOD Thresholds vs. Temperature (BODLEVEL is 2.7V)

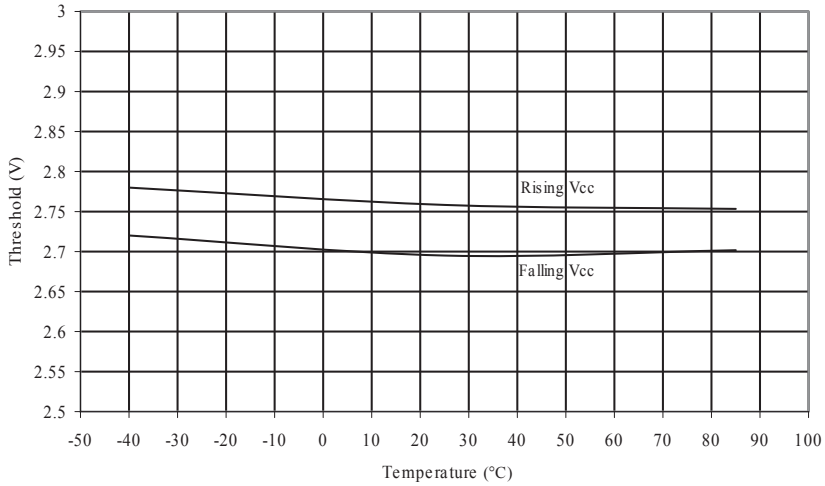
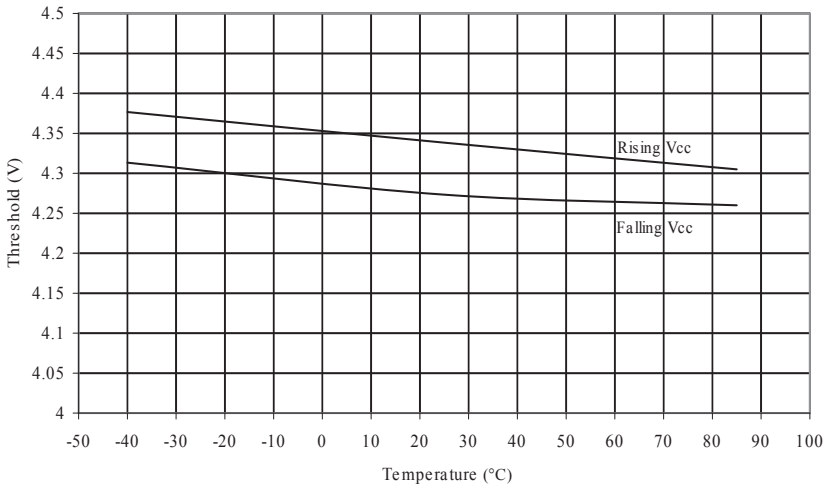


Figure 34-79. BOD Thresholds vs. Temperature (BODLEVEL is 4.3V)



34.2.11. Internal Oscillator Speed

Figure 34-80. Watchdog Oscillator Frequency vs. Temperature

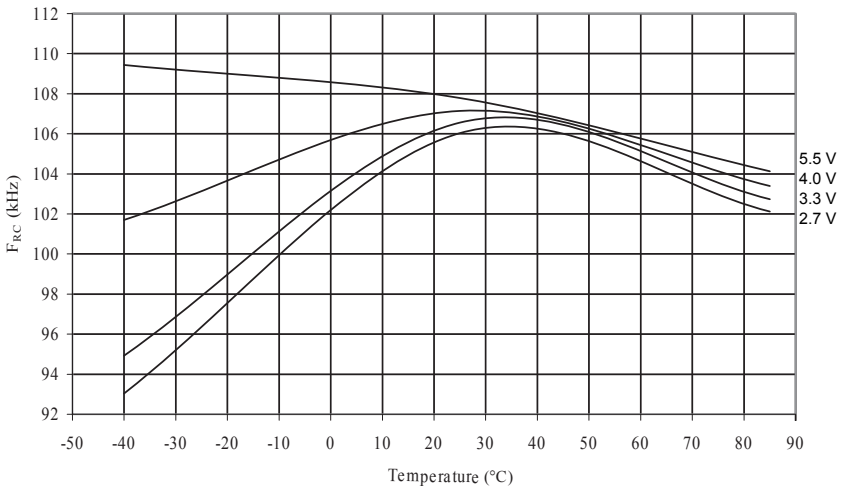
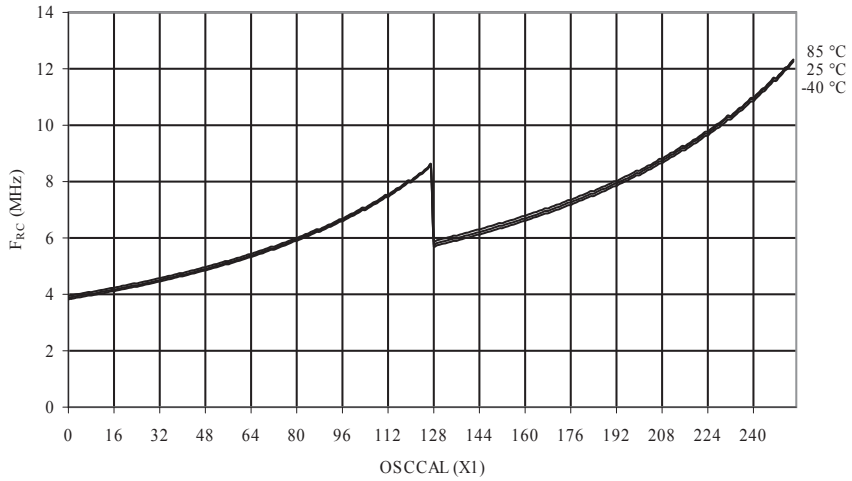


Figure 34-84. Calibrated 8MHz RC Oscillator Frequency vs. OSCCAL Value



34.2.12. Current Consumption of Peripheral Units

Figure 34-85. ADC Current vs. V_{CC} ($AREF = AV_{CC}$)

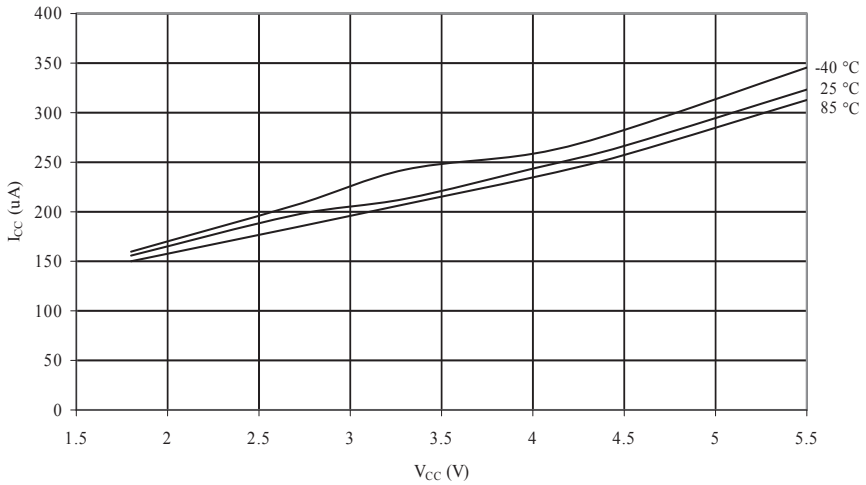
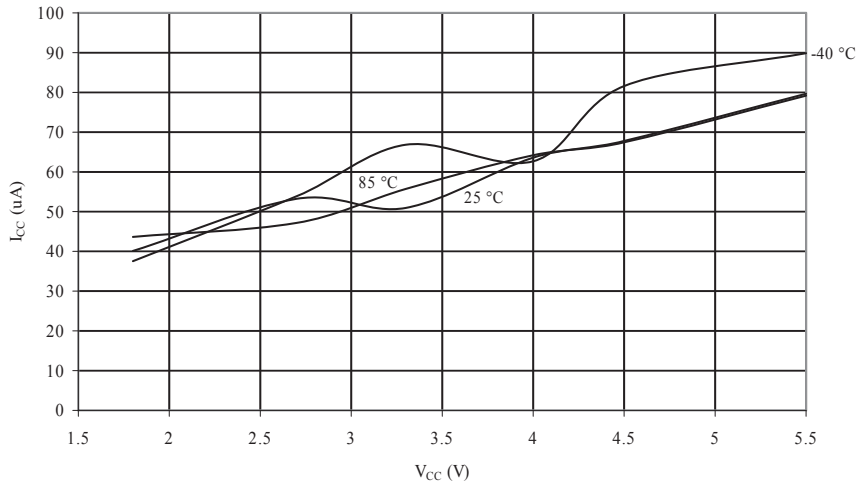


Figure 34-86. Analog Comparator Current vs. V_{CC}



34.3.2. Idle Supply Current

Figure 34-98. ATmega168P/PV: Idle Supply Current vs. Low Frequency (0.1MHz - 1.0MHz)

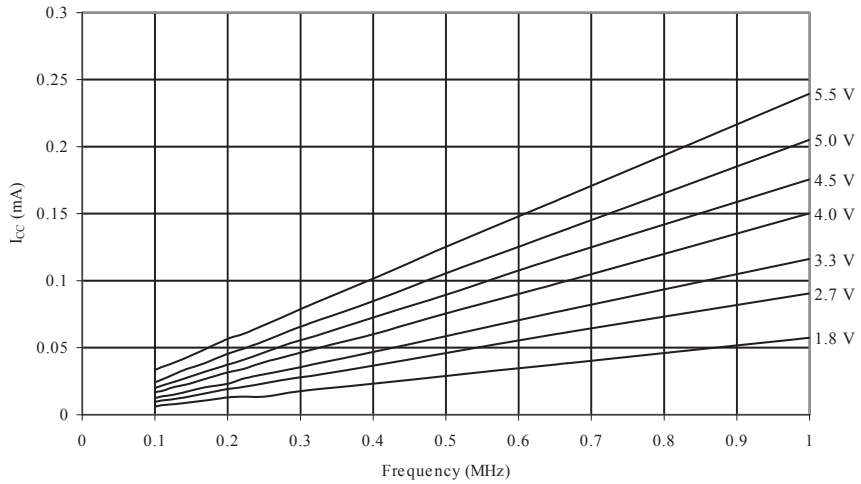


Figure 34-99. ATmega168P/PV: Idle Supply Current vs. Frequency (1MHz - 20MHz)

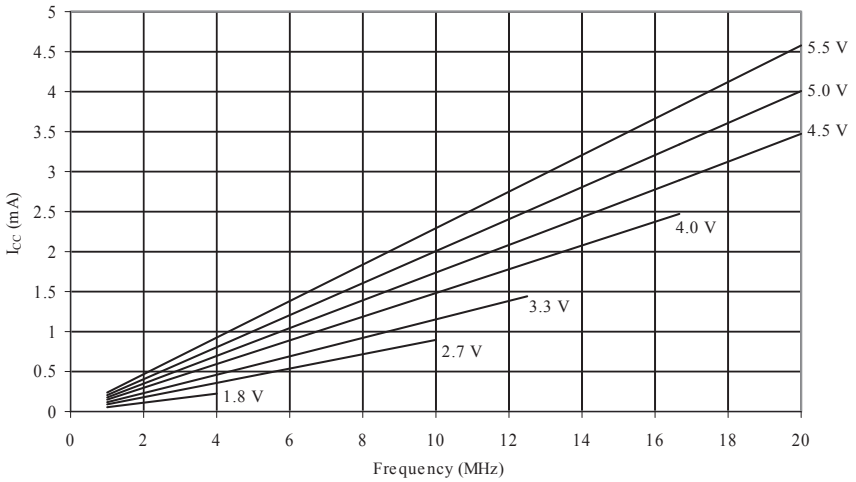
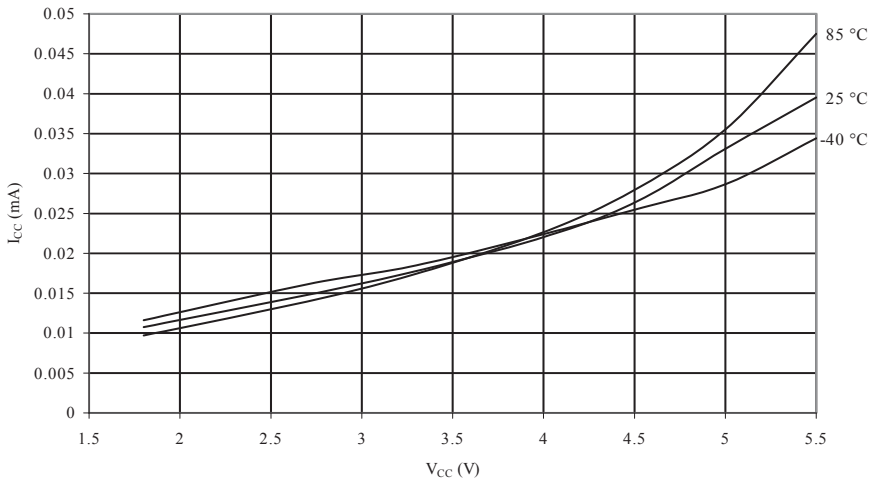


Figure 34-100. ATmega168P/PV: Idle Supply Current vs. V_{CC} (Internal RC Oscillator, 128kHz)

cc



39.12. Rev. 8025E-08/08

1. Updated description of Stack Pointer.
2. Updated description of use of external capacitors in Low Frequency Crystal Oscillator.
3. Updated Table 13-9 in Low Frequency Crystal Oscillator.
4. Added note to Address Match Unit.
5. Added section Reading the Signature Row from Software.
6. Updated Program And Data Memory Lock Bits to include ATmega328P in the description.
7. Added “ATmega328P DC Characteristics” on page 319.
8. Updated Speed Grades for ATmega328P.
9. Removed note 6 and 7 from Table 33-11.
10. Added figure Figure 34-46 for ATmega48P.
11. Added figure Figure 34-92 for ATmega88P.
12. Added figure Figure 34-138 for ATmega168P.
13. Added Register Summary.
14. Updated Ordering Information for Packaging Information.

39.13. Rev. 8025D-03/08

1. Updated figures in Speed Grades.
2. Updated note in Table 33-8 in System and Reset Characteristics.
3. Ordering codes for Packaging Information updated. - ATmega328P is offered in 20 MHz option only.
4. Added Errata for ATmega328P rev. B, “” on page 421.

39.14. Rev. 8025C-01/08

1. Updated Register Summary with Power-save numbers.
2. Added ATmega328P ”Standby Supply Current” on page 408.

39.15. Rev. 8025B-01/08

1. Updated Feature.
2. Added Data Retention.
3. Updated Table 13-2.
4. Removed “Low-frequency Crystal Oscillator Internal Load Capacitance“ table from Low Frequency Crystal Oscillator.
5. Removed JTD bit from MCUCR.
6. Updated typical and general program setup for Reset and Interrupt Vector Addresses in Interrupt Vectors in ATmega168P/PV and ”Interrupt Vectors in ATmega328P” on page 65.
7. Updated Interrupt Vectors Start Address in Table 16-5 and Table 11-7 on page 66.
8. Updated Temperature Measurement.
9. Updated ATmega328P Fuse Bits.
10. Removed VOL3/VOH3 rows from Common DC Characteristics.

39.16. Rev. 8025A-07/07

Initial revision.