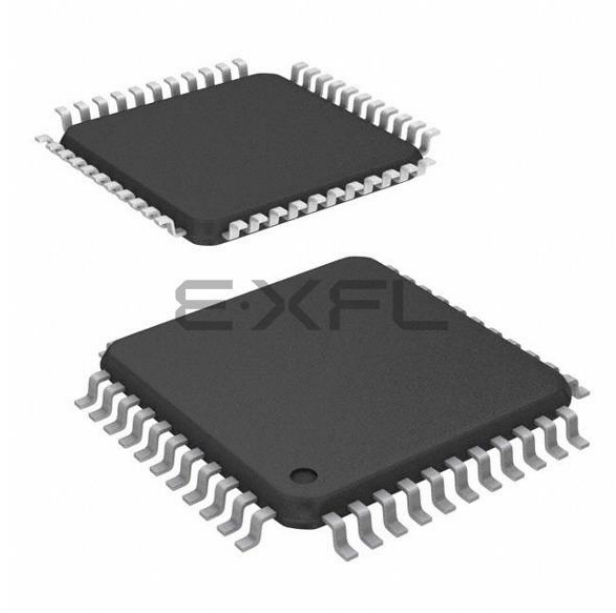


Welcome to E-XFL.COM

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"

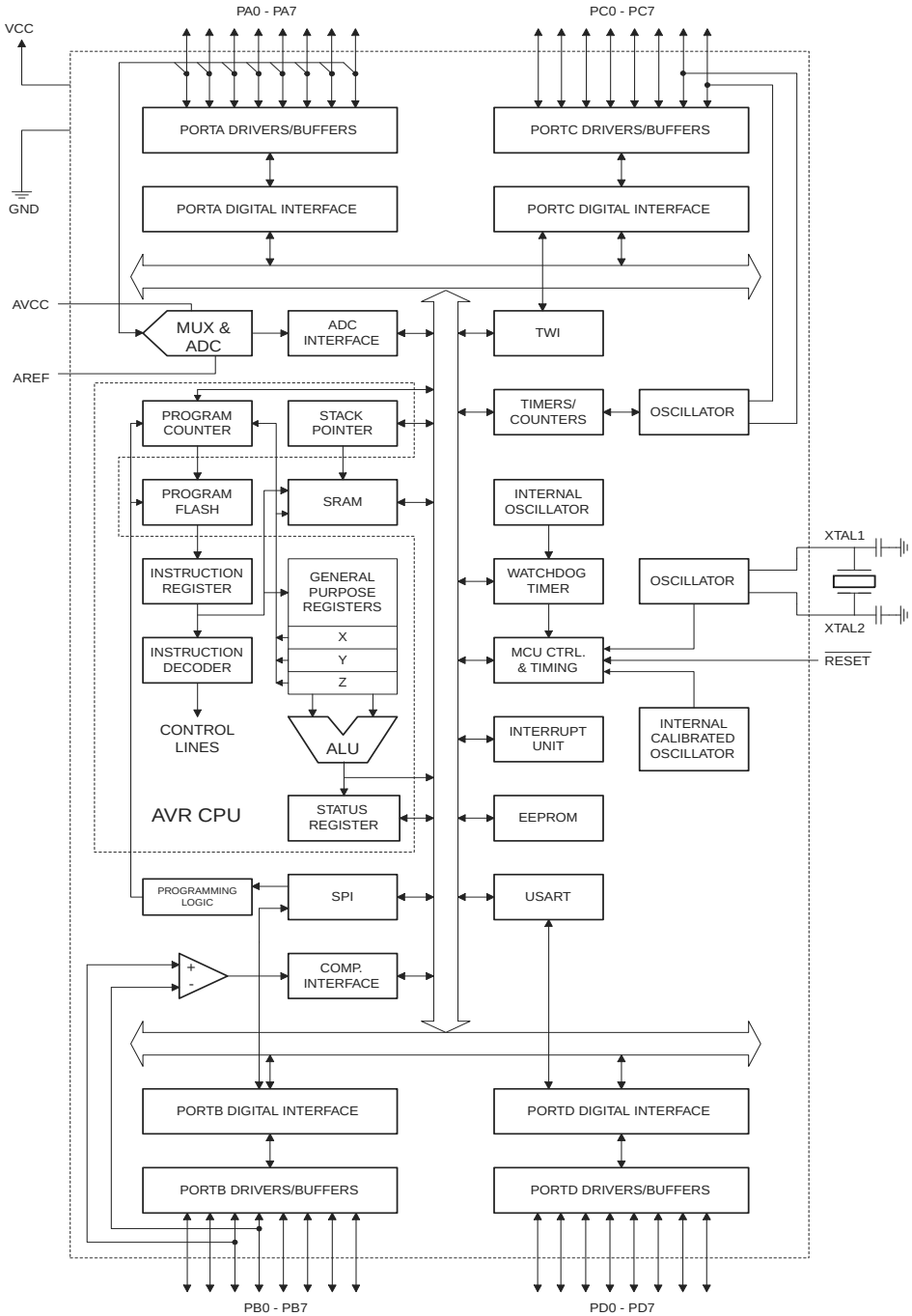
Details	
Product Status	Active
Core Processor	AVR
Core Size	8-Bit
Speed	16MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	32
Program Memory Size	16KB (8K x 16)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-TQFP
Supplier Device Package	44-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atmega16a-au

2. Overview

The ATmega16A is a low-power CMOS 8-bit microcontroller based on the Atmel AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega16A achieves throughputs approaching 1MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

2.1 Block Diagram

Figure 2-1. Block Diagram



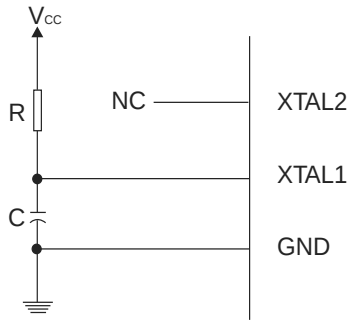
The Atmel AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega16A provides the following features: 16Kbytes of In-System Programmable Flash Program memory with Read-While-Write capabilities; 512bytes EEPROM; 1Kbyte SRAM; 32 general purpose I/O lines, 32 general purpose working registers; a JTAG interface for Boundary-scan; On-chip Debugging support and programming; three flexible Timer/Counters with compare modes; Internal and External Interrupts; a serial programmable USART; a byte oriented Two-wire Serial Interface, an 8-channel; 10-bit ADC with optional differential input stage with programmable gain (TQFP package only); a programmable Watchdog Timer with Internal Oscillator; an SPI serial port; and six software selectable power saving modes. The Idle mode stops the CPU while allowing the USART; Two-wire interface; A/D Converter; SRAM; Timer/Counters; SPI port; and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next External Interrupt or Hardware Reset. In Power-save mode, the Asynchronous Timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except Asynchronous Timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low-power consumption. In Extended Standby mode, both the main Oscillator and the Asynchronous Timer continue to run.

The device is manufactured using Atmels high density nonvolatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed in-system through an SPI serial interface, by a conventional nonvolatile memory programmer, or by an On-chip Boot program running on the AVR core. The boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega16A is a powerful microcontroller that provides a highly-flexible and cost-effective solution to many embedded control applications.

The ATmega16A is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, in-circuit emulators, and evaluation kits.

Figure 8-3. External RC Configuration



The Oscillator can operate in four different modes, each optimized for a specific frequency range. The operating mode is selected by the fuses CKSEL3:0 as shown in Table 8-6.

Table 8-6. External RC Oscillator Operating Modes

CKSEL3:0	Frequency Range (MHz)
0101	0.1 ≤ 0.9
0110	0.9 - 3.0
0111	3.0 - 8.0
1000	8.0 - 12.0

When this Oscillator is selected, start-up times are determined by the SUT Fuses as shown in Table 8-7.

Table 8-7. Start-up Times for the External RC Oscillator Clock Selection

SUT1:0	Start-up Time from Power-down and Power-save	Additional Delay from Reset (V_{CC} = 5.0V)	Recommended Usage
00	18 CK	–	BOD enabled
01	18 CK	4.1ms	Fast rising power
10	18 CK	65ms	Slowly rising power
11	6 CK(1)	4.1ms	Fast rising power or BOD enabled

Note: 1. This option should not be used when operating close to the maximum frequency of the device.

8.7 Calibrated Internal RC Oscillator

The Calibrated Internal RC Oscillator provides a fixed 1.0, 2.0, 4.0, or 8.0MHz clock. All frequencies are nominal values at 5V and 25°C. This clock may be selected as the sys-tem clock by programming the CKSEL Fuses as shown in Table 8-8. If selected, it will operate with no external components. The CKOPT Fuse should always be unpro-grammed when using this clock option. During Reset, hardware loads the calibration byte into the OSCCAL Register and thereby automatically calibrates the RC Oscillator. At 5V, 25°C and 1.0, 2.0, 4.0 or 8.0MHz Oscillator frequency selected, this calibration gives a frequency within ± 3% of the nominal frequency. Using calibration methods as described in application notes available at www.atmel.com/avr it is possible to achieve ±1% accuracy at any given V_{CC} and Temperature. When this Oscillator is used as the Chip Clock, the Watchdog Oscillator will still be used for the Watchdog Timer and for the reset time-out. For more information on the pre-programmed calibration value, see the section “Calibration Byte” on page 253.

```
.org $1C02
$1C02              jmp    EXT_INT0           ; IRQ0 Handler
$1C04              jmp    EXT_INT1           ; IRQ1 Handler
:.     :..         :                         ;
$1C28              jmp    SPM_RDY      ; Store Program Memory Ready Handler
```

When the BOOTRST Fuse is programmed and the Boot section size set to 2K bytes, the most typical and general program setup for the Reset and Interrupt Vector Addresses is:

```
Address Labels     Code                      Comments
.org $002
$002               jmp    EXT_INT0           ; IRQ0 Handler
$004               jmp    EXT_INT1           ; IRQ1 Handler
:.     :..         :                         ;
$028         jmp               SPM_RDY            ; Store Program Memory
Ready Handler
;
.org $1C00
$1C00 RESET:       ldi    r16,high(RAMEND)   ; Main program start
$1C01              out    SPH,r16            ; Set Stack Pointer to top of RAM
$1C02              ldi    r16,low(RAMEND)
$1C03              out    SPL,r16
$1C04              sei                       ; Enable interrupts
$1C05              <instr>  xxx
```

When the BOOTRST Fuse is programmed, the Boot section size set to 2K bytes and the IVSEL bit in the GICR Register is set before any interrupts are enabled, the most typical and general program setup for the Reset and Interrupt Vector Addresses is:

```
Address Labels     Code                      Comments
.org $1C00
$1C00              jmp    RESET              ; Reset handler
$1C02              jmp    EXT_INT0           ; IRQ0 Handler
$1C04              jmp    EXT_INT1           ; IRQ1 Handler
:.     :..         :                         ;
$1C28              jmp    SPM_RDY      ; Store Program Memory Ready Handler
;
$1C2A RESET:       ldi    r16,high(RAMEND)   ; Main program start
$1C2B              out    SPH,r16            ; Set Stack Pointer to top of RAM
$1C2C              ldi    r16,low(RAMEND)
$1C2D              out    SPL,r16
$1C2E              sei                       ; Enable interrupts
$1C2F              <instr>  xxx
```

11.2.1 Moving Interrupts Between Application and Boot Space

The General Interrupt Control Register controls the placement of the Interrupt Vector table.

11.2.2 GICR – General Interrupt Control Register

Bit	7	6	5	4	3	2	1	0	
	INT1	**INT0**	**INT2**	**–**	**–**	**–**	**IVSEL**	**IVCE**	**GICR**
Read/Write	R/W	R/W	R/W	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Switching between input with pull-up and output low generates the same problem. The user must use either the tri-state ({DDxn, PORTxn} = 0b00) or the output high state ({DDxn, PORTxn} = 0b11) as an intermediate step.

Table 12-1 summarizes the control signals for the pin value.

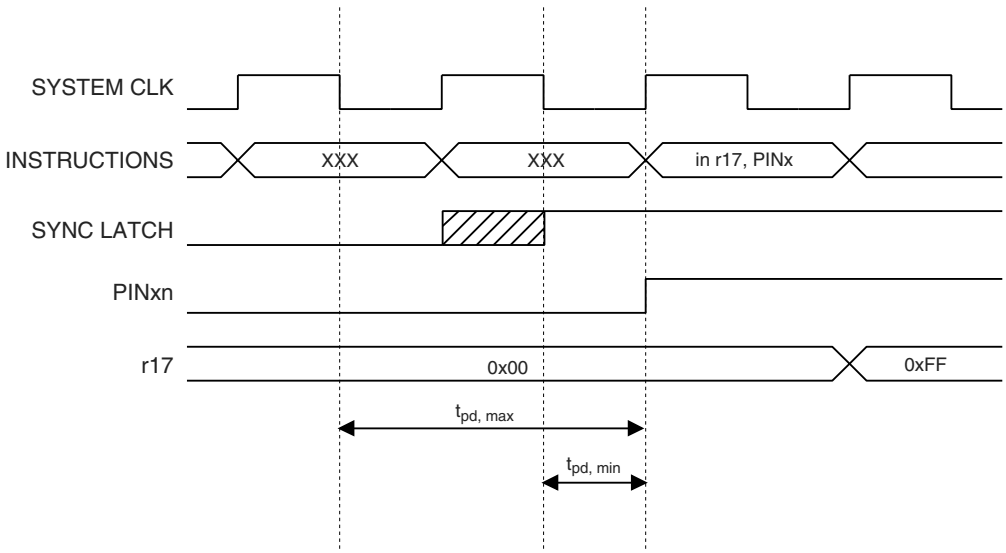
Table 12-1. Port Pin Configurations

DDxn	PORTxn	PUD (in SFIOR)	I/O	Pull-up	Comment
0	0	X	Input	No	Tri-state (Hi-Z)
0	1	0	Input	Yes	Pxn will source current if ext. pulled low.
0	1	1	Input	No	Tri-state (Hi-Z)
1	0	X	Output	No	Output Low (Sink)
1	1	X	Output	No	Output High (Source)

12.2.2 Reading the Pin Value

Independent of the setting of Data Direction bit DDxn, the port pin can be read through the PINxn Register bit. As shown in Figure 12-2, the PINxn Register bit and the preceding latch constitute a synchronizer. This is needed to avoid metastability if the physical pin changes value near the edge of the internal clock, but it also introduces a delay. Figure 12-3 shows a timing diagram of the synchronization when reading an externally applied pin value. The maximum and minimum propagation delays are denoted $t_{pd,max}$ and $t_{pd,min}$ respectively.

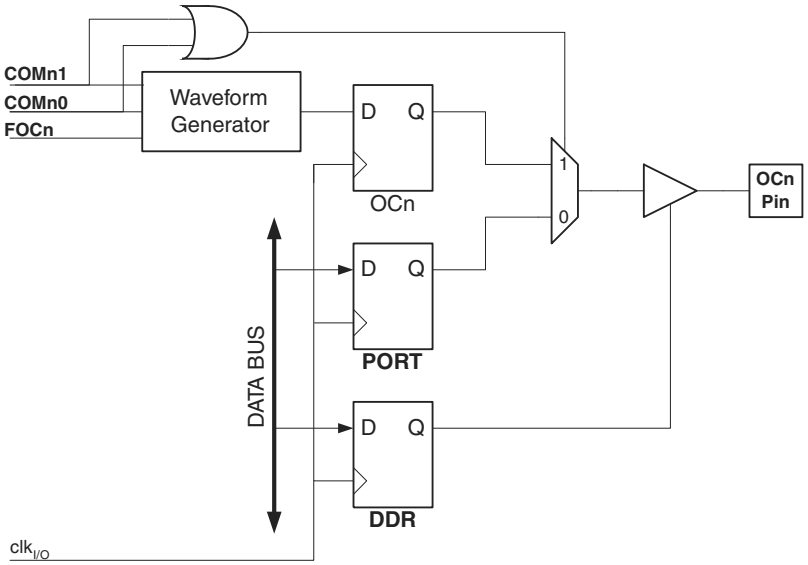
Figure 12-3. Synchronization when Reading an Externally Applied Pin Value



Consider the clock period starting shortly *after* the first falling edge of the system clock. The latch is closed when the clock is low, and goes transparent when the clock is high, as indicated by the shaded region of the “SYNC LATCH” signal. The signal value is latched when the system clock goes low. It is clocked into the PINxn Register at the succeeding positive clock edge. As indicated by the two arrows $t_{pd,max}$ and $t_{pd,min}$, a single signal transition on the pin will be delayed between ½ and 1½ system clock period depending upon the time of assertion.

When reading back a software assigned pin value, a *nop* instruction must be inserted as indicated in Figure 12-4. The *out* instruction sets the “SYNC LATCH” signal at the positive edge of the clock. In this case, the delay t_{pd} through the synchronizer is one system clock period.

Figure 14-4. Compare Match Output Unit, Schematic



The general I/O port function is overridden by the Output Compare (OC0) from the Waveform Generator if either of the COM01:0 bits are set. However, the OC0 pin direction (input or output) is still controlled by the Data Direction Register (DDR) for the port pin. The Data Direction Register bit for the OC0 pin (DDR_OC0) must be set as output before the OC0 value is visible on the pin. The port override function is independent of the Waveform Generation mode.

The design of the output compare pin logic allows initialization of the OC0 state before the output is enabled. Note that some COM01:0 bit settings are reserved for certain modes of operation. See “Register Description” on page 79.

14.6.1 Compare Output Mode and Waveform Generation

The Waveform Generator uses the COM01:0 bits differently in normal, CTC, and PWM modes. For all modes, setting the COM01:0 = 0 tells the waveform generator that no action on the OC0 Register is to be performed on the next compare match. For compare output actions in the non-PWM modes refer to Table 14-3 on page 80. For fast PWM mode, refer to Table 14-4 on page 80, and for phase correct PWM refer to Table 14-5 on page 81.

A change of the COM01:0 bits state will have effect at the first compare match after the bits are written. For non-PWM modes, the action can be forced to have immediate effect by using the FOC0 strobe bits.

14.7 Modes of Operation

The mode of operation, i.e., the behavior of the Timer/Counter and the Output Compare pins, is defined by the combination of the Waveform Generation mode (WGM01:0) and Compare Output mode (COM01:0) bits. The Compare Output mode bits do not affect the counting sequence, while the Waveform Generation mode bits do. The COM01:0 bits control whether the PWM output generated should be inverted or not (inverted or non-inverted PWM). For non-PWM modes the COM01:0 bits control whether the output should be set, cleared, or toggled at a compare match (See “Compare Match Output Unit” on page 72.).

For detailed timing information refer to Figure 14-8, Figure 14-9, Figure 14-10 and Figure 14-11 in “Timer/Counter Timing Diagrams” on page 77.

14.7.1 Normal Mode

The simplest mode of operation is the normal mode (WGM01:0 = 0). In this mode the counting direction is always up (incrementing), and no counter clear is performed. The counter simply overruns when it passes its

16.11.4 OCR1AH and OCR1AL – Output Compare Register 1 A

Bit	7	6	5	4	3	2	1	0	
	OCR1A[15:8]								OCR1AH
	OCR1A[7:0]								OCR1AL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

16.11.5 OCR1BH and OCR1BL – Output Compare Register 1 B

Bit	7	6	5	4	3	2	1	0	
	OCR1B[15:8]								OCR1BH
	OCR1B[7:0]								OCR1BL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The Output Compare Registers contain a 16-bit value that is continuously compared with the counter value (TCNT1). A match can be used to generate an output compare interrupt, or to generate a waveform output on the OC1x pin.

The Output Compare Registers are 16-bit in size. To ensure that both the high and Low bytes are written simultaneously when the CPU writes to these registers, the access is performed using an 8-bit temporary High Byte Register (TEMP). This temporary register is shared by all the other 16-bit registers. See “Accessing 16-bit Registers” on page 87.

16.11.6 ICR1H and ICR1L – Input Capture Register 1

Bit	7	6	5	4	3	2	1	0	
	ICR1[15:8]								ICR1H
	ICR1[7:0]								ICR1L
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The Input Capture is updated with the counter (TCNT1) value each time an event occurs on the ICP1 pin (or optionally on the analog comparator output for Timer/Counter1). The Input Capture can be used for defining the counter TOP value.

The Input Capture Register is 16-bit in size. To ensure that both the high and Low bytes are read simultaneously when the CPU accesses these registers, the access is performed using an 8-bit temporary High Byte Register (TEMP). This temporary register is shared by all the other 16-bit registers. See “Accessing 16-bit Registers” on page 87.

16.11.7 TIMSK – Timer/Counter Interrupt Mask Register[1]

Bit	7	6	5	4	3	2	1	0	
	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	TIMSK
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Note: 1. This register contains interrupt control bits for several Timer/Counters, but only Timer1 bits are described in this section. The remaining bits are described in their respective timer sections.

20.3.3 Address Packet Format

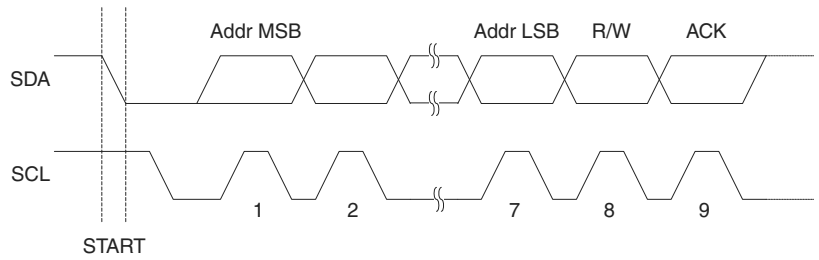
All address packets transmitted on the TWI bus are nine bits long, consisting of seven address bits, one READ/WRITE control bit and an acknowledge bit. If the READ/WRITE bit is set, a read operation is to be performed, otherwise a write operation should be performed. When a Slave recognizes that it is being addressed, it should acknowledge by pulling SDA low in the ninth SCL (ACK) cycle. If the addressed Slave is busy, or for some other reason can not service the Master's request, the SDA line should be left high in the ACK clock cycle. The Master can then transmit a STOP condition, or a REPEATED START condition to initiate a new transmission. An address packet consisting of a Slave address and a READ or a WRITE bit is called SLA+R or SLA+W, respectively.

The MSB of the address byte is transmitted first. Slave addresses can freely be allocated by the designer, but the address 0000 000 is reserved for a general call.

When a general call is issued, all Slaves should respond by pulling the SDA line low in the ACK cycle. A general call is used when a Master wishes to transmit the same message to several Slaves in the system. When the general call address followed by a Write bit is transmitted on the bus, all Slaves set up to acknowledge the general call will pull the SDA line low in the ack cycle. The following data packets will then be received by all the Slaves that acknowledged the general call. Note that transmitting the general call address followed by a Read bit is meaningless, as this would cause contention if several Slaves started transmitting different data.

All addresses of the format 1111 xxx should be reserved for future purposes.

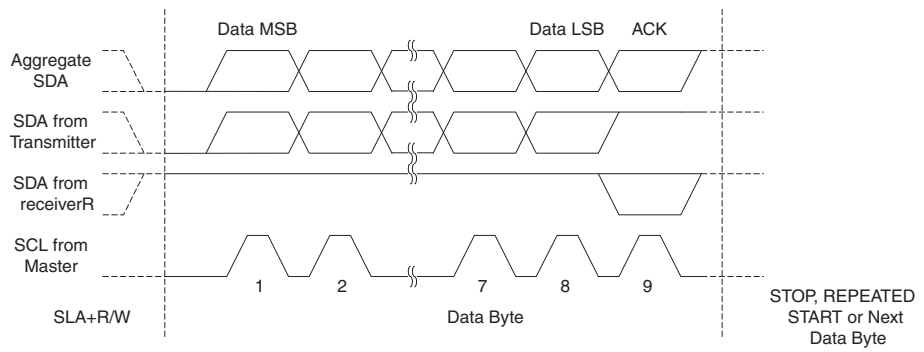
Figure 20-4. Address Packet Format



20.3.4 Data Packet Format

All data packets transmitted on the TWI bus are nine bits long, consisting of one data byte and an acknowledge bit. During a data transfer, the Master generates the clock and the START and STOP conditions, while the receiver is responsible for acknowledging the reception. An Acknowledge (ACK) is signalled by the receiver pulling the SDA line low during the ninth SCL cycle. If the receiver leaves the SDA line high, a NACK is signalled. When the receiver has received the last byte, or for some reason cannot receive any more bytes, it should inform the transmitter by sending a NACK after the final byte. The MSB of the data byte is transmitted first.

Figure 20-5. Data Packet Format

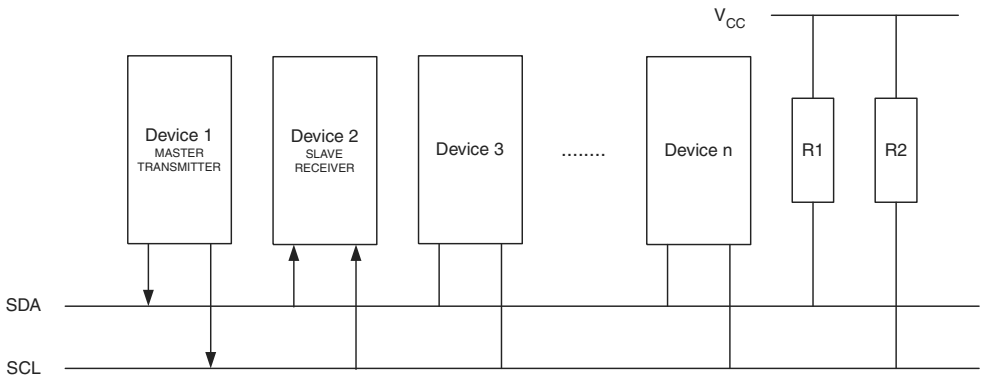


When the TWINT Flag is set, the status code in TWSR is used to determine the appropriate software action. For each status code, the required software action and details of the following serial transfer are given in Table 20-2 to Table 20-5. Note that the prescaler bits are masked to zero in these tables.

20.7.1 Master Transmitter Mode

In the Master Transmitter mode, a number of data bytes are transmitted to a Slave Receiver (see Figure 20-11). In order to enter a Master mode, a START condition must be transmitted. The format of the following address packet determines whether Master Transmitter or Master Receiver 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 20-11. Data Transfer in Master Transmitter Mode



A START condition is sent by writing the following value to TWCR:

TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
Value	1	X	1	0	X	1	0	X

TWEN must be set to enable the Two-wire Serial Interface, TWSTA must be written to one to transmit a START condition and TWINT must be written to one to clear the TWINT Flag. The TWI will then test the Two-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 TWSR will be $08 (See Table 20-2). In order to enter MT mode, SLA+W must be transmitted. This is done by writing SLA+W to TWDR. Thereafter the TWINT bit should be cleared (by writing it to one) to continue the transfer. This is accomplished by writing the following value to TWCR:

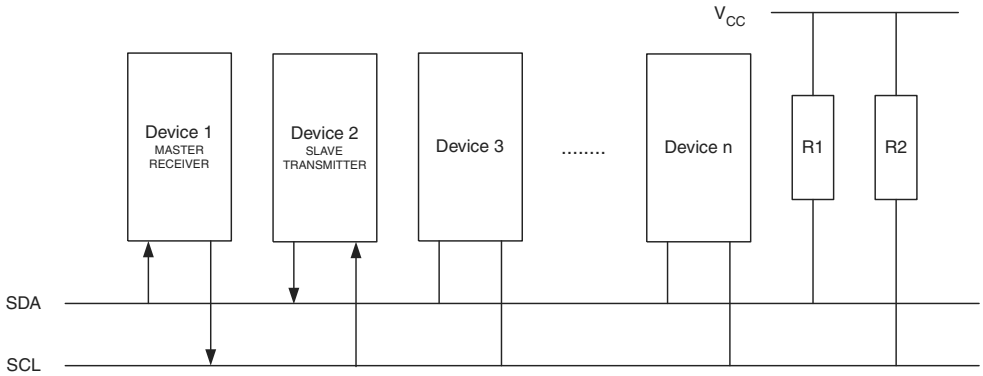
TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
Value	1	X	0	0	X	1	0	X

When SLA+W have been transmitted and an acknowledgement 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 $18, $20, or $38. The appropriate action to be taken for each of these status codes is detailed in Table 20-2.

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 TWCR Register. After updating TWDR, the TWINT bit should be cleared (by writing it to one) to continue the transfer. This is accomplished by writing the following value to TWCR:

TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
Value	1	X	0	0	X	1	0	X

Figure 20-13. Data Transfer in Master Receiver Mode



A START condition is sent by writing the following value to TWCR:

TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
Value	1	X	1	0	X	1	0	X

TWEN must be written to one to enable the Two-wire Serial Interface, TWSTA must be written to one to transmit a START condition and TWINT must be set to clear the TWINT Flag. The TWI will then test the Two-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 TWSR will be $08 (See Table 20-2). In order to enter MR mode, SLA+R must be transmitted. This is done by writing SLA+R to TWDR. Thereafter the TWINT bit should be cleared (by writing it to one) to continue the transfer. This is accomplished by writing the following value to TWCR:

TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
Value	1	X	0	0	X	1	0	X

When SLA+R have been transmitted and an acknowledgement 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 $38, $40, or $48. The appropriate action to be taken for each of these status codes is detailed in Table 20-3. Received data can be read from the TWDR Register when the TWINT Flag is set high by hardware. This scheme is repeated until the last byte has been received. After the last byte has been received, the MR should inform the ST by sending a NACK after the last received data byte. The transfer is ended by generating a STOP condition or a repeated START condition. A STOP condition is generated by writing the following value to TWCR:

TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
Value	1	X	0	1	X	1	0	X

A REPEATED START condition is generated by writing the following value to TWCR:

TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
Value	1	X	1	0	X	1	0	X

After a repeated START condition (state $10) the Two-wire Serial Interface can access the same Slave again, or a new Slave without transmitting a STOP condition. Repeated START enables the Master to switch between Slaves, Master Transmitter mode and Master Receiver mode without losing control over the bus.

20.9.3 TWSR – TWI Status Register

Bit	7	6	5	4	3	2	1	0	
	TWS7	TWS6	TWS5	TWS4	TWS3	–	TWPS1	TWPS0	TWSR
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	1	1	1	1	1	0	0	0	

- **Bits 7:3 – TWS: TWI Status**

These five bits reflect the status of the TWI logic and the Two-wire Serial Bus. The different status codes are described later in this section. Note that the value read from TWSR contains both the 5-bit status value and the 2-bit prescaler value. The application designer should mask the prescaler bits to zero when checking the Status bits. This makes status checking independent of prescaler setting. This approach is used in this datasheet, unless otherwise noted.

- **Bit 2 – Res: Reserved Bit**

This bit is reserved and will always read as zero.

- **Bits 1:0 – TWPS: TWI Prescaler Bits**

These bits can be read and written, and control the bit rate prescaler.

Table 20-7. TWI Bit Rate Prescaler

TWPS1	TWPS0	Prescaler Value
0	0	1
0	1	4
1	0	16
1	1	64

To calculate bit rates, see “Bit Rate Generator Unit” on page 172. The value of TWPS1:0 is used in the equation.

20.9.4 TWDR – TWI Data Register

Bit	7	6	5	4	3	2	1	0	
	TWD7	TWD6	TWD5	TWD4	TWD3	TWD2	TWD1	TWD0	TWDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	1	1	1	1	1	1	1	1	

In Transmit mode, TWDR contains the next byte to be transmitted. In Receive mode, the TWDR contains the last byte received. It is writable while the TWI is not in the process of shifting a byte. This occurs when the TWI Interrupt Flag (TWINT) is set by hardware. Note that the Data Register cannot be initialized by the user before the first interrupt occurs. The data in TWDR remains stable as long as TWINT is set. While data is shifted out, data on the bus is simultaneously shifted in. TWDR always contains the last byte present on the bus, except after a wake up from a sleep mode by the TWI interrupt. In this case, the contents of TWDR is undefined. In the case of a lost bus arbitration, no data is lost in the transition from Master to Slave. Handling of the ACK bit is controlled automatically by the TWI logic, the CPU cannot access the ACK bit directly.

- **Bits 7:0 – TWD: TWI Data Register**

These eight bits contain the next data byte to be transmitted, or the latest data byte received on the Two-wire Serial Bus.

ADC Data Register immediately, regardless of any ongoing conversions. For a complete description of this bit, see “ADCL and ADCH – The ADC Data Register” on page 211.

- **Bits 4:0 – MUX4:0: Analog Channel and Gain Selection Bits**

The value of these bits selects which combination of analog inputs are connected to the ADC. These bits also select the gain for the differential channels. See Table 22-4 for details. If these bits are changed during a conversion, the change will not go in effect until this conversion is complete (ADIF in ADCSRA is set).

Table 22-4. Input Channel and Gain Selections

MUX4:0	Single Ended Input	Positive Differential Input	Negative Differential Input	Gain
00000	ADC0	N/A		
00001	ADC1			
00010	ADC2			
00011	ADC3			
00100	ADC4			
00101	ADC5			
00110	ADC6			
00111	ADC7			
01000	N/A	ADC0	ADC0	10x
01001		ADC1	ADC0	10x
01010(1)		ADC0	ADC0	200x
01011(1)		ADC1	ADC0	200x
01100		ADC2	ADC2	10x
01101		ADC3	ADC2	10x
01110(1)		ADC2	ADC2	200x
01111(1)		ADC3	ADC2	200x
10000		ADC0	ADC1	1x
10001		ADC1	ADC1	1x
10010		ADC2	ADC1	1x
10011		ADC3	ADC1	1x
10100		ADC4	ADC1	1x
10101		ADC5	ADC1	1x
10110		ADC6	ADC1	1x
10111		ADC7	ADC1	1x
11000		ADC0	ADC2	1x
11001		ADC1	ADC2	1x
11010		ADC2	ADC2	1x
11011		ADC3	ADC2	1x
11100		ADC4	ADC2	1x

26.7.5 Programming the EEPROM

The EEPROM is organized in pages, see Table 26-6 on page 254. When programming the EEPROM, the program data is latched into a page buffer. This allows one page of data to be programmed simultaneously. The programming algorithm for the EEPROM data memory is as follows (refer to “Programming the Flash” on page 257 for details on Command, Address and Data loading):

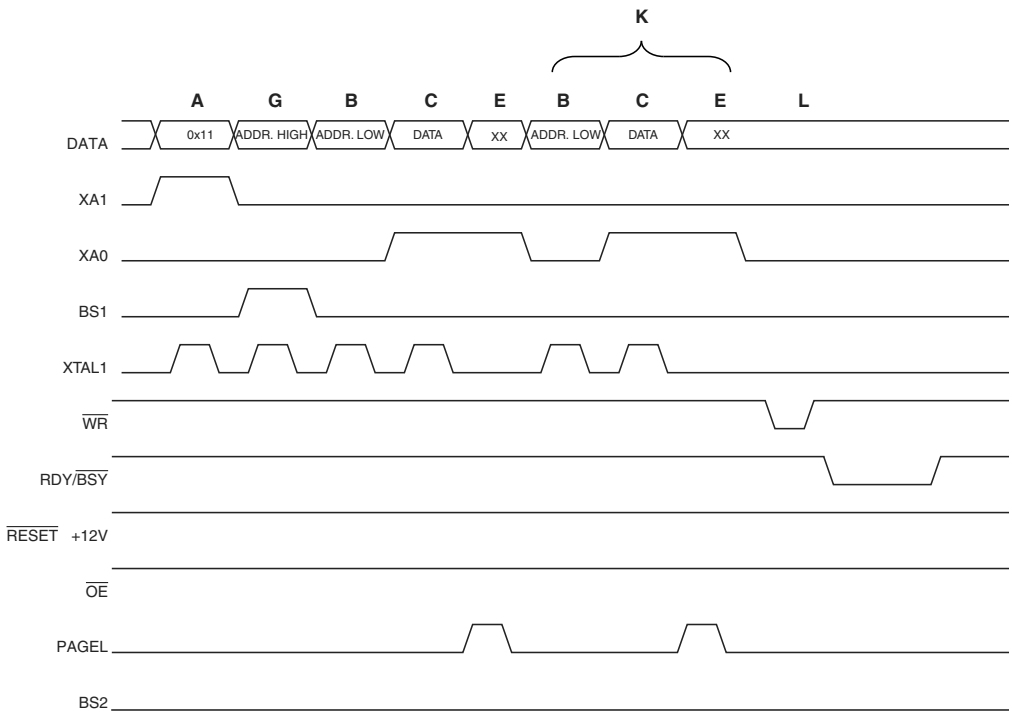
1. A: Load Command “0001 0001”.
2. G: Load Address High Byte ($00 - $FF)
3. B: Load Address Low Byte ($00 - $FF)
4. C: Load Data ($00 - $FF)
5. E: Latch data (give PAGEL a positive pulse)

K: Repeat 3 through 5 until the entire buffer is filled

L: Program EEPROM page

1. Set BS1 to “0”.
2. Give $\overline{WR}$ a negative pulse. This starts programming of the EEPROM page. RDY/$\overline{BSY}$ goes low.
3. Wait until to RDY/$\overline{BSY}$ goes high before programming the next page. (See Figure 26-4 for signal waveforms)

Figure 26-4. Programming the EEPROM Waveforms



26.7.6 Reading the Flash

The algorithm for reading the Flash memory is as follows (refer to “Programming the Flash” on page 257 for details on Command and Address loading):

1. A: Load Command “0000 0010”.
2. G: Load Address High Byte ($00 - $FF)
3. B: Load Address Low Byte ($00 - $FF)
4. Set $\overline{OE}$ to “0”, and BS1 to “0”. The Flash word Low byte can now be read at DATA.

T_A = -40°C to 85°C, V_{CC} = 2.7V to 5.5V (Unless Otherwise Noted) (Continued)

Symbol	Parameter	Condition	Min	Typ	Max	Units
I_{CC}	Power Supply Current	Active 1MHz, V_{CC} = 3V		0.6		mA
		Active 4MHz, V_{CC} = 3V		1.9	5	mA
		Active 8MHz, V_{CC} = 5V		7	15	mA
		Idle 1MHz, V_{CC} = 3V		0.2		mA
		Idle 4MHz, V_{CC} = 3V		0.6	2	mA
		Idle 8MHz, V_{CC} = 5V		2.7	7	mA
	Power-down Mode(5)	WDT enabled, V_{CC} = 3V		<8	15	µA
		WDT disabled, V_{CC} = 3V		< 1	4	µA
V_{ACIO}	Analog Comparator Input Offset Voltage	V_{CC} = 5V V_{in} = V_{CC}/2			40	mV
I_{ACLK}	Analog Comparator Input Leakage Current	V_{CC} = 5V V_{in} = V_{CC}/2	-50		50	nA
t_{ACPD}	Analog Comparator Propagation Delay	V_{CC} = 2.7V V_{CC} = 4.0V		750 500		ns

Notes:
1. “Max” means the highest value where the pin is guaranteed to be read as low
2. “Min” means the lowest value where the pin is guaranteed to be read as high
3. Although each I/O port can sink more than the test conditions (20mA at Vcc = 5V, 10mA at Vcc = 3V) under steady state conditions (non-transient), the following must be observed:
   PDIP Package:
   1] The sum of all IOL, for all ports, should not exceed 200mA.
   2] The sum of all IOL, for port A0 - A7, should not exceed 100mA.
   3] The sum of all IOL, for ports B0 - B7,C0 - C7, D0 - D7 and XTAL2, should not exceed 100mA.
   TQFP and QFN/MLF Package:
   1] The sum of all IOL, for all ports, should not exceed 400mA.
   2] The sum of all IOL, for ports A0 - A7, should not exceed 100mA.
   3] The sum of all IOL, for ports B0 - B7, should not exceed 100mA.
   4] The sum of all IOL, for ports C0 - C7, should not exceed 100mA.
   5] The sum of all IOL, for ports D0 - D7, xtal2, should not exceed 100mA.
   If IOL exceeds the test condition, VOL may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test condition.
4. Although each I/O port can source more than the test conditions (20mA at Vcc = 5V, 10mA at Vcc = 3V) under steady state conditions (non-transient), the following must be observed:
   PDIP Package:
   1] The sum of all IOH, for all ports, should not exceed 200mA.
   2] The sum of all IOH, for port A0 - A7, should not exceed 100mA.
   3] The sum of all IOH, for ports B0 - B7,C0 - C7, D0 - D7 and XTAL2, should not exceed 100mA.
   TQFP and QFN/MLF Package:
   1] The sum of all IOH, for all ports, should not exceed 400mA.
   2] The sum of all IOH, for ports A0 - A7, should not exceed 100mA.
   3] The sum of all IOH, for ports B0 - B7, should not exceed 100mA.
   4] The sum of all IOL, for ports C0 - C7, should not exceed 100mA.
   5] The sum of all IOL, for ports D0 - D7, xtal2, should not exceed 100mA..
5. Minimum V_{CC} for Power-down is 2.5V.

27.6 External Interrupts Characteristics

Table 27-3. Asynchronous External Interrupt Characteristics

Symbol	Parameter	Condition	Min	Typ	Max	Units
t_{INT}	Minimum pulse width for asynchronous external interrupt			50		ns

27.7 Two-wire Serial Interface Characteristics

Table 27-4 describes the requirements for devices connected to the Two-wire Serial Bus. The ATmega16A Two-wire Serial Interface meets or exceeds these requirements under the noted conditions.

Timing symbols refer to Figure 27-4.

Table 27-4. Two-wire Serial Bus Requirements

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low-voltage		-0.5	0.3 V_{CC}	V
V_{IH}	Input High-voltage		0.7 V_{CC}	V_{CC} + 0.5	V
V_{hys}(1)	Hysteresis of Schmitt Trigger Inputs		0.05 V_{CC}(2)	–	V
V_{OL}(1)	Output Low-voltage	3mA sink current	0	0.4	V
t_r(1)	Rise Time for both SDA and SCL		20 + 0.1C_b(3)(2)	300	ns
t_{of}(1)	Output Fall Time from V_{IHmin} to V_{ILmax}	10pF < C_b < 400pF(3)	20 + 0.1C_b(3)(2)	250	ns
t_{SP}(1)	Spikes Suppressed by Input Filter		0	50(2)	ns
I_i	Input Current each I/O Pin	0.1V_{CC} < V_i < 0.9V_{CC}	-10	10	µA
C_i(1)	Capacitance for each I/O Pin		–	10	pF
f_{SCL}	SCL Clock Frequency	f_{CK}(4) > max(16f_{SCL}, 250kHz)(5)	0	400	kHz
Rp	Value of Pull-up resistor	f_{SCL} ≤ 100kHz	$\frac{V_{CC} - 0,4V}{3mA}$	$\frac{1000ns}{C_b}$	Ω
		f_{SCL} > 100kHz	$\frac{V_{CC} - 0,4V}{3mA}$	$\frac{300ns}{C_b}$	Ω
$t_{HD;STA}$	Hold Time (repeated) START Condition	f_{SCL} ≤ 100kHz	4.0	–	µs
		f_{SCL} > 100kHz	0.6	–	µs
t_{LOW}	Low Period of the SCL Clock	f_{SCL} ≤ 100kHz	4.7	–	µs
		f_{SCL} > 100kHz	1.3	–	µs
t_{HIGH}	High period of the SCL clock	f_{SCL} ≤ 100kHz	4.0	–	µs
		f_{SCL} > 100kHz	0.6	–	µs
$t_{SU;STA}$	Set-up time for a repeated START condition	f_{SCL} ≤ 100kHz	4.7	–	µs
		f_{SCL} > 100kHz	0.6	–	µs
$t_{HD;DAT}$	Data hold time	f_{SCL} ≤ 100 kHz	0	3.45	µs
		f_{SCL} > 100 kHz	0	0.9	µs
$t_{SU;DAT}$	Data setup time	f_{SCL} ≤ 100 kHz	250	–	ns
		f_{SCL} > 100 kHz	100	–	ns

Notes: 1. Values are guidelines only.
2. Minimum for AVCC is 2.7V.
3. Maximum for AVCC is 5.5V.

27.10 Parallel Programming Characteristics

Figure 27-8. Parallel Programming Timing, Including some General Timing Requirements

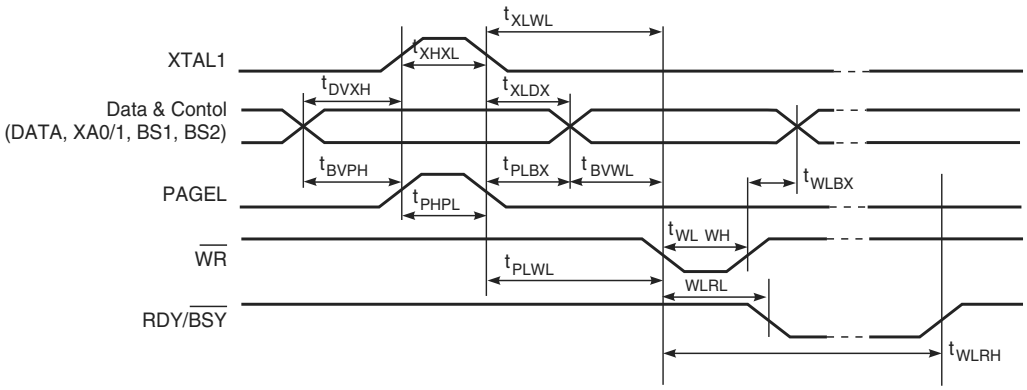
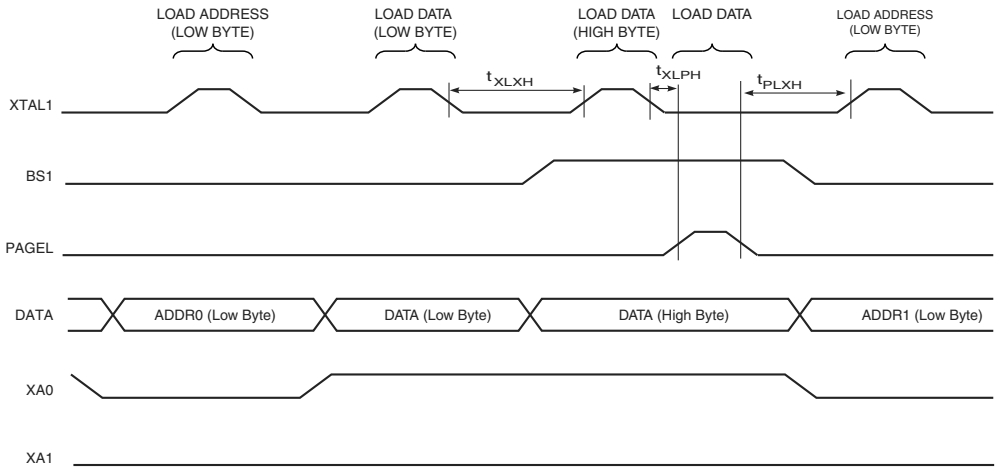


Figure 27-9. Parallel Programming Timing, Loading Sequence with Timing Requirements(1)



Note: 1. The timing requirements shown in Figure 27-8 (i.e., t_{DVXH}, t_{XHXL}, and t_{XLDX}) also apply to loading operation.

Figure 28-2. Active Supply Current vs. Frequency (1 - 16MHz)

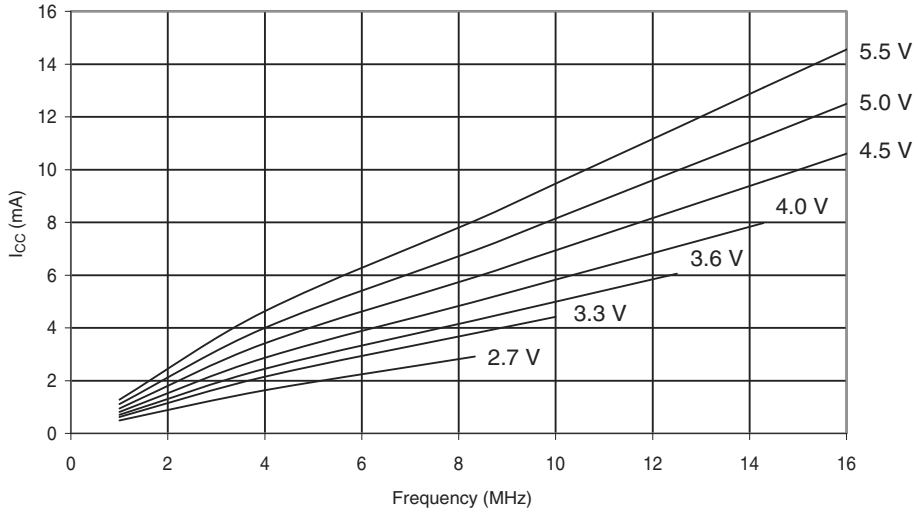


Figure 28-3. Active Supply Current vs. V_{CC} (Internal RC Oscillator, 8MHz)

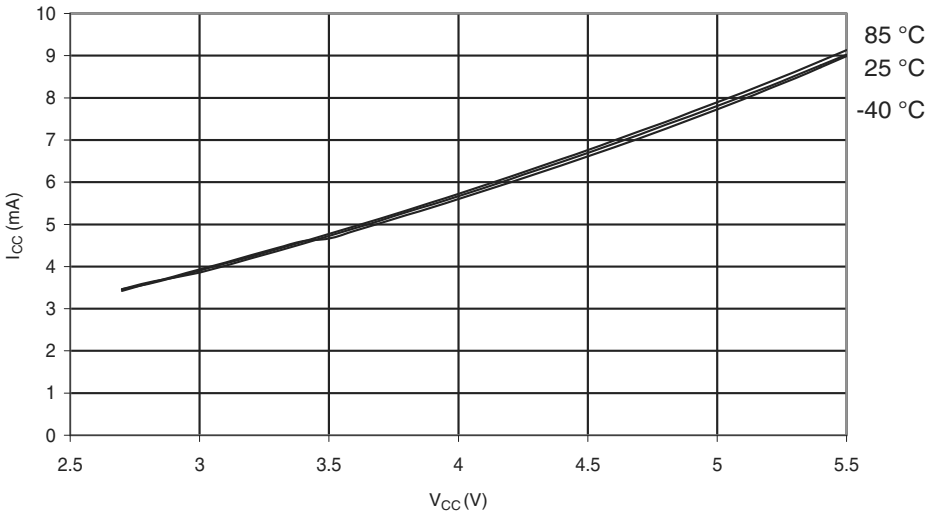
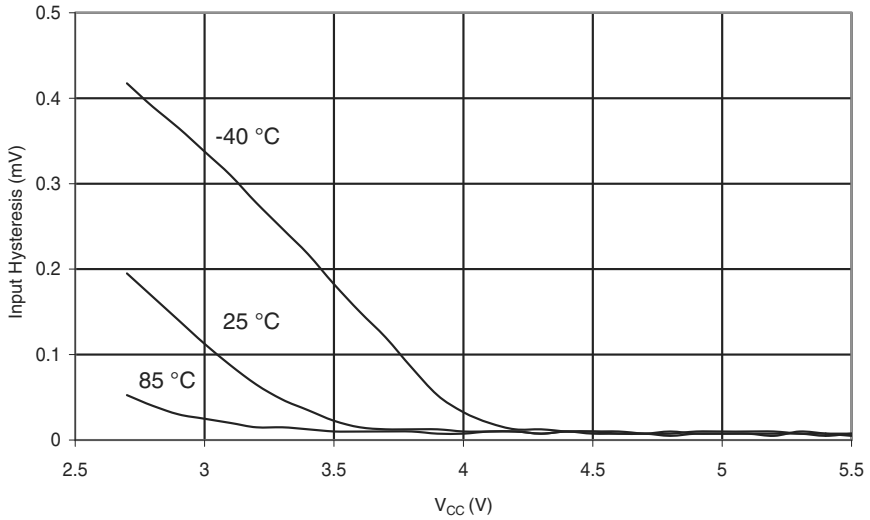


Figure 28-32. Reset Input Pin Hysteresis vs. V_{CC}



28.0.9 Bod Thresholds

Figure 28-33. Bod Thresholds vs. Temperature (Bodlevel is 4.0V)

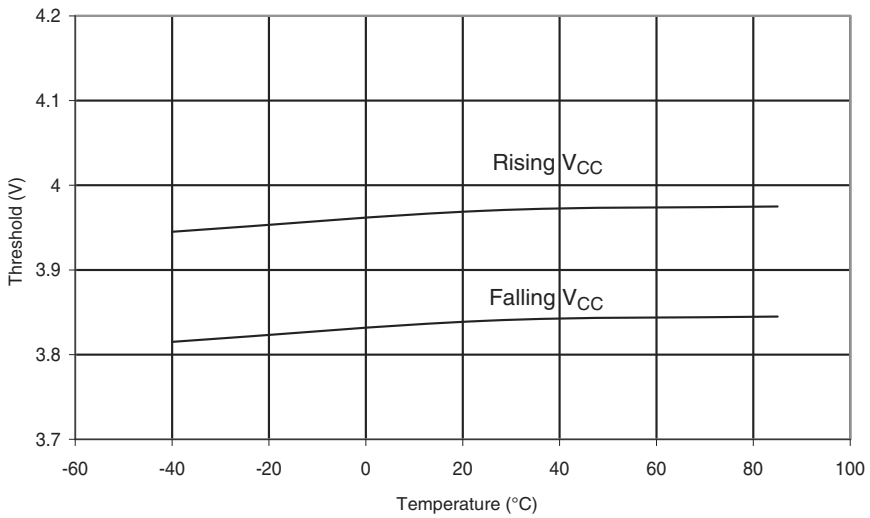


Figure 28-40. Calibrated 4 MHz RC Oscillator Frequency vs. Temperature

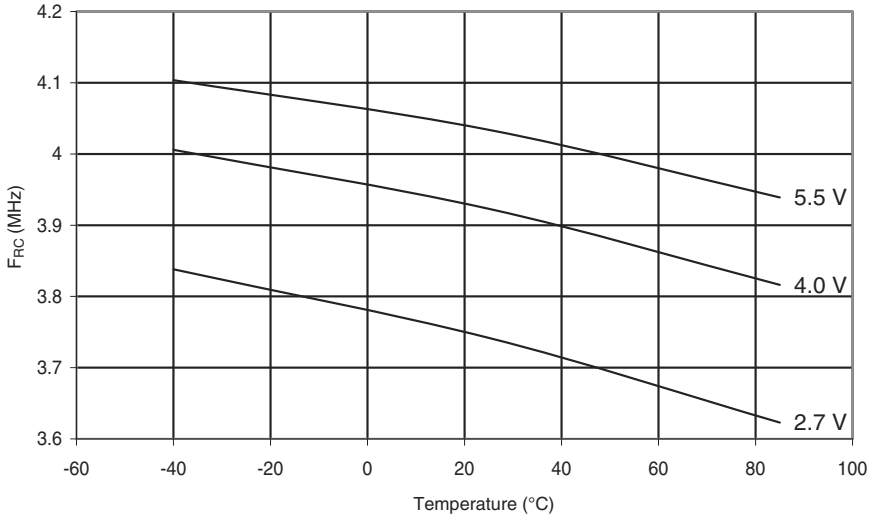
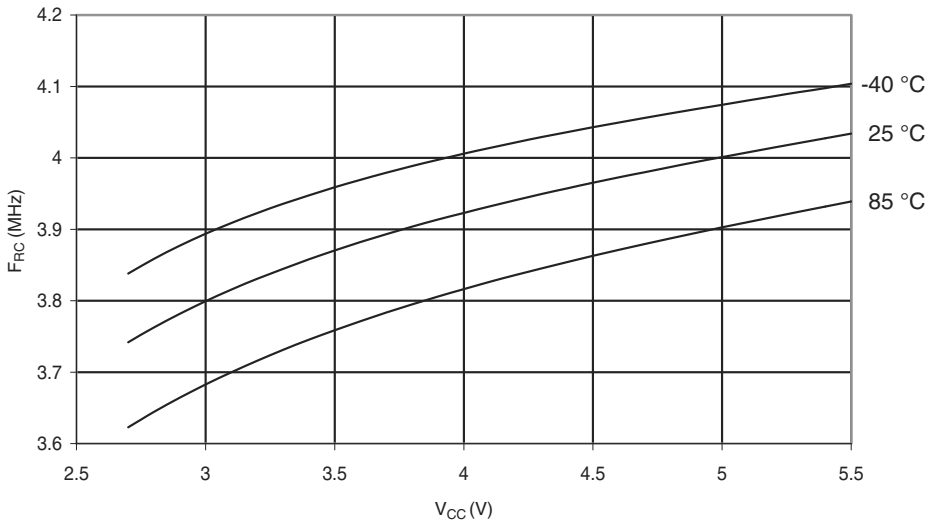


Figure 28-41. Calibrated 4MHz RC Oscillator Frequency vs. V_{CC}



29. Register Summary

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
$3F ($5F)	SREG	I	T	H	S	V	N	Z	C	9
$3E ($5E)	SPH	–	–	–	–	–	SP10	SP9	SP8	11
$3D ($5D)	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	11
$3C ($5C)	OCR0	Timer/Counter0 Output Compare Register								82
$3B ($5B)	GICR	INT1	INT0	INT2	–	–	–	IVSEL	IVCE	46, 67
$3A ($5A)	GIFR	INTF1	INTF0	INTF2	–	–	–	–	–	68
$39 ($59)	TIMSK	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	82, 109, 128
$38 ($58)	TIFR	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0	82, 110, 128
$37 ($57)	SPMCR	SPMIE	RWWSB	–	RWWSRE	BLBSET	PGWRT	PGERS	SPMEN	242
$36 ($56)	TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE	189
$35 ($55)	MCUCR	SM2	SE	SM1	SM0	ISC11	ISC10	ISC01	ISC00	35, 66
$34 ($54)	MCUCSR	JTD	ISC2	–	JTRF	WDRF	BORF	EXTRF	PORF	41, 67, 236
$33 ($53)	TCCR0	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00	79
$32 ($52)	TCNT0	Timer/Counter0 (8 Bits)								81
$31[1] ($51)[1]	OSCCAL	Oscillator Calibration Register								30
	OCDR	On-Chip Debug Register								218
$30 ($50)	SFIOR	ADTS2	ADTS1	ADTS0	–	ACME	PUD	PSR2	PSR10	64,84,129,194,212
$2F ($4F)	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	WGM11	WGM10	105
$2E ($4E)	TCCR1B	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	107
$2D ($4D)	TCNT1H	Timer/Counter1 – Counter Register High Byte								108
$2C ($4C)	TCNT1L	Timer/Counter1 – Counter Register Low Byte								108
$2B ($4B)	OCR1AH	Timer/Counter1 – Output Compare Register A High Byte								109
$2A ($4A)	OCR1AL	Timer/Counter1 – Output Compare Register A Low Byte								109
$29 ($49)	OCR1BH	Timer/Counter1 – Output Compare Register B High Byte								109
$28 ($48)	OCR1BL	Timer/Counter1 – Output Compare Register B Low Byte								109
$27 ($47)	ICR1H	Timer/Counter1 – Input Capture Register High Byte								109
$26 ($46)	ICR1L	Timer/Counter1 – Input Capture Register Low Byte								109
$25 ($45)	TCCR2	FOC2	WGM20	COM21	COM20	WGM21	CS22	CS21	CS20	125
$24 ($44)	TCNT2	Timer/Counter2 (8 Bits)								127
$23 ($43)	OCR2	Timer/Counter2 Output Compare Register								127
$22 ($42)	ASSR	–	–	–	–	AS2	TCN2UB	OCR2UB	TCR2UB	127
$21 ($41)	WDTCR	–	–	–	WDTOE	WDE	WDP2	WDP1	WDP0	41
$20[2] ($40)[2]	UBRRH	URSEL	–	–	–	UBRR[11:8]				162
	UCSRC	URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL	161
$1F ($3F)	EEARH	–	–	–	–	–	–	–	EEAR8	20
$1E ($3E)	EEARL	EEPROM Address Register Low Byte								20
$1D ($3D)	EEDR	EEPROM Data Register								20
$1C ($3C)	EECR	–	–	–	–	EERIE	EEMWE	EEWE	EERE	20
$1B ($3B)	PORTA	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	64
$1A ($3A)	DDRA	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	64
$19 ($39)	PINA	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	64
$18 ($38)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	64
$17 ($37)	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	64
$16 ($36)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	65
$15 ($35)	PORTC	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	65
$14 ($34)	DDRC	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	65
$13 ($33)	PINC	PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	65
$12 ($32)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	65
$11 ($31)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	65
$10 ($30)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	65
$0F ($2F)	SPDR	SPI Data Register								139
$0E ($2E)	SPSR	SPIF	WCOL	–	–	–	–	–	SPI2X	138
$0D ($2D)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	137
$0C ($2C)	UDR	USART I/O Data Register								158
$0B ($2B)	UCSRA	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM	159
$0A ($2A)	UCSRB	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8	160
$09 ($29)	UBRRL	USART Baud Rate Register Low Byte								162
$08 ($28)	ACSR	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	194
$07 ($27)	ADMUX	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	208
$06 ($26)	ADCSRA	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	210
$05 ($25)	ADCH	ADC Data Register High Byte								211
$04 ($24)	ADCL	ADC Data Register Low Byte								211
$03 ($23)	TWDR	Two-wire Serial Interface Data Register								191
$02 ($22)	TWAR	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE	192
$01 ($21)	TWSR	TWS7	TWS6	TWS5	TWS4	TWS3	–	TWPS1	TWPS0	191
$00 ($20)	TWBR	Two-wire Serial Interface Bit Rate Register								189