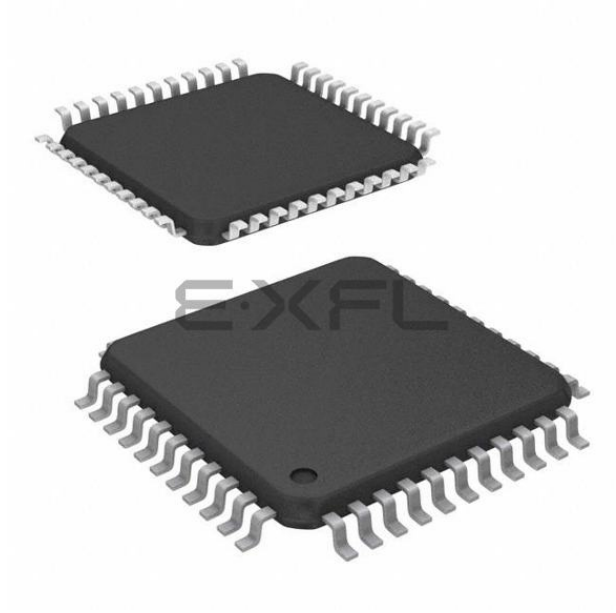


**Welcome to E-XFL.COM**

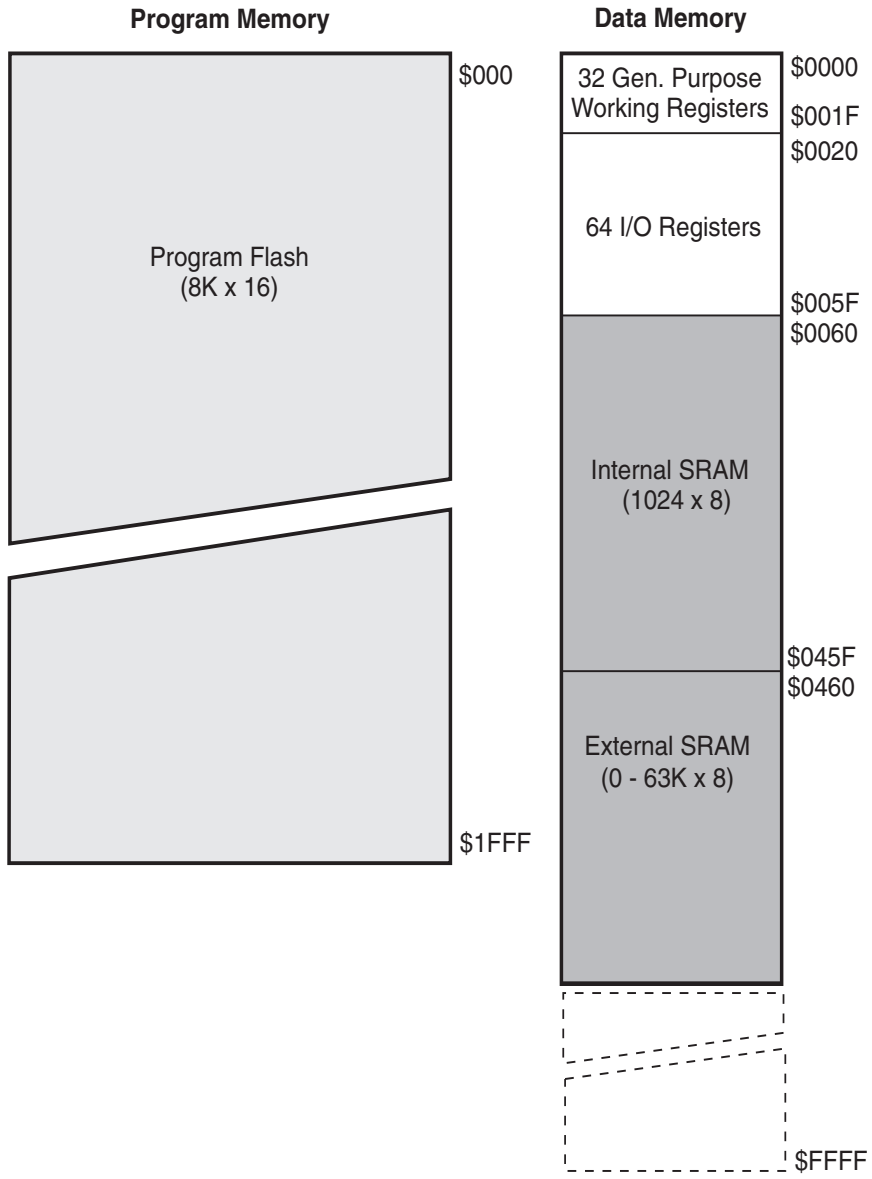
## What is "Embedded - Microcontrollers"?

"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 | Obsolete |
| Core Processor | AVR |
| Core Size | 8-Bit |
| Speed | 4MHz |
| Connectivity | SPI, UART/USART |
| Peripherals | POR, PWM, WDT |
| Number of I/O | 35 |
| 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 | - |
| Oscillator Type | External |
| 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/atmega161l-4ai |

**Figure 5.** Memory Maps




A flexible interrupt module has its control registers in the I/O space with an additional global interrupt enable bit in the Status Register. All the different interrupts have a separate Interrupt Vector in the Interrupt Vector table at the beginning of the Program memory. The different interrupts have priority in accordance with their Interrupt Vector position. The lower the Interrupt Vector address, the higher the priority.

- **Bit 5 – OCIE1B: Timer/Counter1 Output CompareB Match Interrupt Enable**

When the OCIE1B bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter1 CompareB Match interrupt is enabled. The corresponding interrupt (at Vector $010) is executed if a Compare B Match in Timer/Counter1 occurs, i.e., when the OCF1B bit is set in the Timer/Counter Interrupt Flag Register (TIFR).

- **Bit 4 – TOIE2: Timer/Counter2 Overflow Interrupt Enable**

When the TOIE2 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter2 Overflow interrupt is enabled. The corresponding interrupt (at Vector $00a) is executed if an overflow in Timer/Counter2 occurs, i.e., when the TOV2 bit is set in the Timer/Counter Interrupt Flag Register (TIFR).

- **Bit 3 – TICIE1: Timer/Counter1 Input Capture Interrupt Enable**

When the TICIE1 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter1 Input Capture Event Interrupt is enabled. The corresponding interrupt (at Vector $00C) is executed if a capture-triggering event occurs on pin 31, ICP, i.e., when the ICF1 bit is set in the Timer/Counter Interrupt Flag Register (TIFR).

- **Bit 2 – OCIE2: Timer/Counter2 Output Compare Match Interrupt Enable**

When the OCIE2 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter2 Compare Match interrupt is enabled. The corresponding interrupt (at Vector $008) is executed if a Compare2 match in Timer/Counter2 occurs, i.e., when the OCF2 bit is set in the Timer/Counter Interrupt Flag Register (TIFR).

- **Bit 1 – TOIE0: Timer/Counter0 Overflow Interrupt Enable**

When the TOIE0 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter0 Overflow interrupt is enabled. The corresponding interrupt (at Vector $016) is executed if an overflow in Timer/Counter0 occurs, i.e., when the TOV0 bit is set in the Timer/Counter Interrupt Flag Register (TIFR).

- **Bit 0 – OCIE0: Timer/Counter0 Output Compare Match Interrupt Enable**

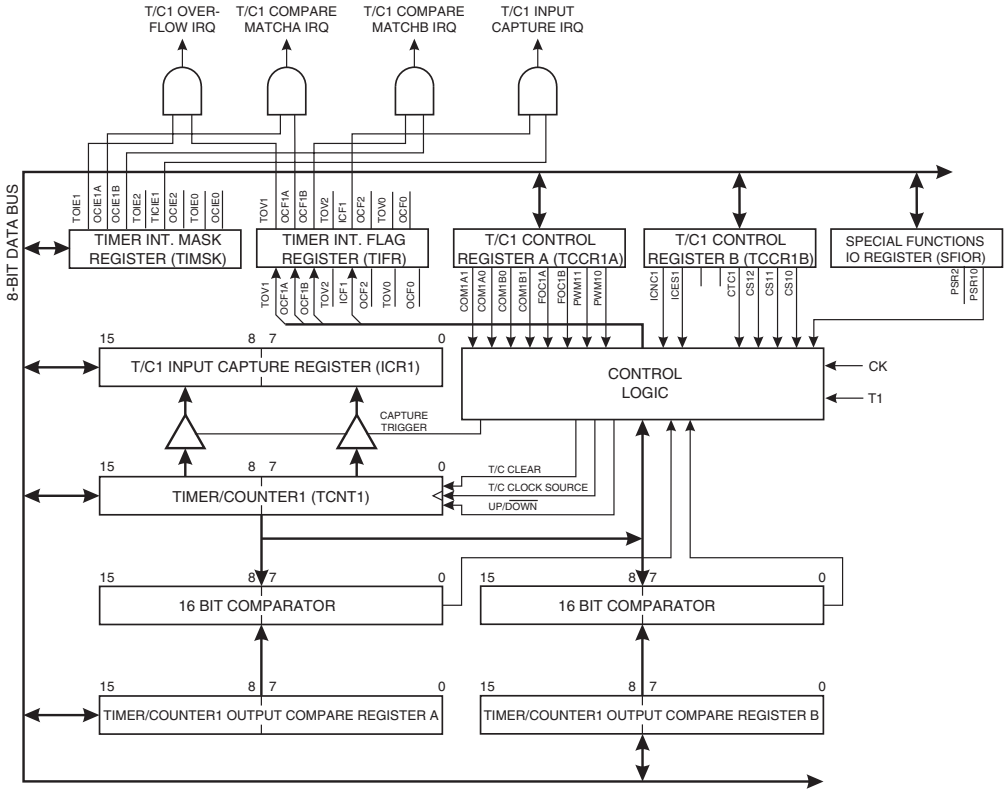
When the OCIE0 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter0 Compare Match interrupt is enabled. The corresponding interrupt (at Vector $014) is executed if a Compare 0 Match in Timer/Counter0 occurs, i.e., when the OCF0 bit is set in the Timer/Counter Interrupt Flag Register (TIFR).

- Description of wake-up from Power-save mode when the timer is clocked asynchronously: When the interrupt condition is met, the wake-up process is started on the following cycle of the timer clock, that is, the timer is always advanced by at least 1 before the processor can read the counter value. The Interrupt Flags are updated three processor cycles after the processor clock has started. During these cycles, the processor executes instructions, but the interrupt condition is not readable and the interrupt routine has not started yet.
- During asynchronous operation, the synchronization of the Interrupt Flags for the asynchronous timer takes three processor cycles plus one timer cycle. The timer is therefore advanced by at least 1 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.

## Timer/Counter1

Figure 35 shows the block diagram for Timer/Counter1.

**Figure 35.** Timer/Counter1 Block Diagram




The 16-bit Timer/Counter1 can select clock source from CK, prescaled CK or an external pin. In addition, it can be stopped as described in “Timer/Counter1 Control Register B – TCCR1B”. The different Status Flags (Overflow, Compare Match, and Capture Event) are found in the Timer/Counter Interrupt Flag Register (TIFR). Control signals are found in the Timer/Counter1 Control Registers (TCCR1A and TCCR1B). The interrupt enable/disable settings for Timer/Counter1 are found in the Timer/Counter Interrupt Mask Register (TIMSK).

When Timer/Counter1 is externally clocked, the external signal is synchronized with the Oscillator frequency of the CPU. To assure proper sampling of the external clock, the minimum time between two external clock transitions must be at least one internal CPU

clock period. The external clock signal is sampled on the rising edge of the internal CPU clock.

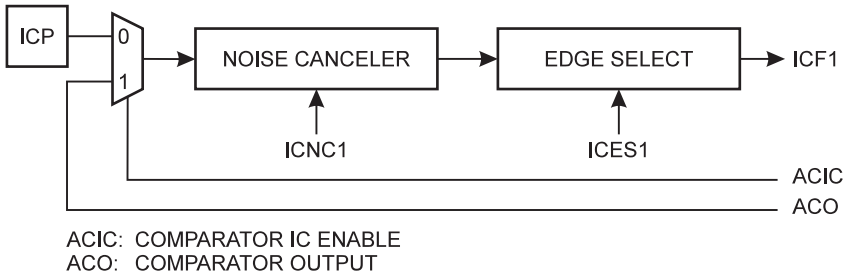
The 16-bit Timer/Counter1 features both a high-resolution and a high-accuracy usage with the lower prescaling opportunities. Similarly, the high-prescaling opportunities make the Timer/Counter1 useful for lower speed functions or exact timing functions with infrequent actions.

The Timer/Counter1 supports two Output Compare functions using the Output Compare Register 1 A and B (OCR1A and OCR1B) as the data sources to be compared to the Timer/Counter1 contents. The Output Compare functions include optional clearing of the counter on compareA match and actions on the Output Compare pins on both compare matches.

Timer/Counter1 can also be used as an 8-, 9-, or 10-bit Pulse Width Modulator. In this mode the counter and the OCR1A/OCR1B Registers serve as a dual glitch-free stand-alone PWM with centered pulses. Alternatively, the Timer/Counter1 can be configured to operate at twice the speed in PWM mode, but without centered pulses. Refer to page 55 for a detailed description of this function.

The Input Capture function of Timer/Counter1 provides a capture of the Timer/Counter1 contents to the Input Capture Register (ICR1), triggered by an external event on the Input Capture Pin (ICP). The actual capture event settings are defined by the Timer/Counter1 Control Register (TCCR1B). In addition, the Analog Comparator can be set to trigger the Input Capture. Refer to the section, “The Analog Comparator”, for details on this. The ICP pin logic is shown in Figure 36.

**Figure 36.** ICP Pin Schematic Diagram




If the noise canceler function is enabled, the actual trigger condition for the Capture Event is monitored over four samples, and all four must be equal to activate the Capture Flag.

**Timer/Counter1 Control Register A – TCCR1A**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $2F ($4F) | COM1A1 | COM1A0 | COM1B1 | COM1B0 | FOC1A | FOC1B | PWM11 | PWM10 | TCCR1A |
| 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 | |

- **Bits 7, 6 – COM1A1, COM1A0: Compare Output Mode1A, Bits 1 and 0**

The COM1A1 and COM1A0 control bits determine any output pin action following a compare match in Timer/Counter1. Any output pin actions affect pin OC1A (Output CompareA pin 1). This is an alternative function to an I/O port, and the corresponding direction control bit must be set (one) to control an output pin. The control configuration is shown in Table 14.

WDTOE bit is set (one). To disable an enabled Watchdog Timer, the following procedure must be followed:

1. In the same operation, write a logical “1” to WDTOE and WDE. A logical “1” must be written to WDE even though it is set to one before the disable operation starts.
2. Within the next four clock cycles, write a logical “0” to WDE. This disables the Watchdog.

- **Bits 2..0 – WDP2, WDP1, WDP0: Watchdog Timer Prescaler 2, 1 and 0**

The WDP2, WDP1 and WDP0 bits determine the Watchdog Timer prescaling when the Watchdog Timer is enabled. The different prescaling values and their corresponding time-out periods are shown in Table 20.

**Table 20.** Watchdog Timer Prescale Select (1)

| WDP2 | WDP1 | WDP0 | Number of WDT Oscillator Cycles | Typical Time-out at $V_{CC}$ = 3.0V | Typical Time-out at $V_{CC}$ = 5.0V |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 16K | 47 ms | 15 ms |
| 0 | 0 | 1 | 32K | 94 ms | 30 ms |
| 0 | 1 | 0 | 64K | 0.19 s | 60 ms |
| 0 | 1 | 1 | 128K | 0.38 s | 0.12 s |
| 1 | 0 | 0 | 256K | 0.75 s | 0.24 s |
| 1 | 0 | 1 | 512K | 1.5 s | 0.49 s |
| 1 | 1 | 0 | 1,024K | 3.0 s | 0.97 s |
| 1 | 1 | 1 | 2,048K | 6.0 s | 1.9 s |

Note: 1. The frequency of the Watchdog Oscillator is voltage-dependent, as shown in the Electrical Characteristics section.
The WDR (Watchdog Reset) instruction should always be executed before the Watchdog Timer is enabled. This ensures that the reset period will be in accordance with the Watchdog Timer prescale settings. If the Watchdog Timer is enabled without reset, the Watchdog Timer may not start counting from zero.
To avoid unintentional MCU Reset, the Watchdog Timer should be disabled or reset before changing the Watchdog Timer Prescale Select.

## $\overline{SS}$ Pin Functionality

When the SPI is configured as a Master (MSTR in SPCR is set), the user can determine the direction of the $\overline{SS}$ pin. If $\overline{SS}$ is configured as an output, the pin is a general output pin, which does not affect the SPI system. If $\overline{SS}$ is configured as an input, it must be held high to ensure Master SPI operation. If the $\overline{SS}$ pin is driven low by peripheral circuitry when the SPI is configured as Master with the $\overline{SS}$ pin defined as an input, the SPI system interprets this as another Master selecting the SPI as a Slave and starting to send data to it. To avoid bus contention, the SPI system takes the following actions:

1. The MSTR bit in SPCR is cleared and the SPI system becomes a Slave. As a result of the SPI becoming a Slave, the MOSI and SCK pins become inputs.
2. The SPIF Flag in SPSR is set, and if the SPI interrupt is enabled and the I-bit in SREG is set, the interrupt routine will be executed.

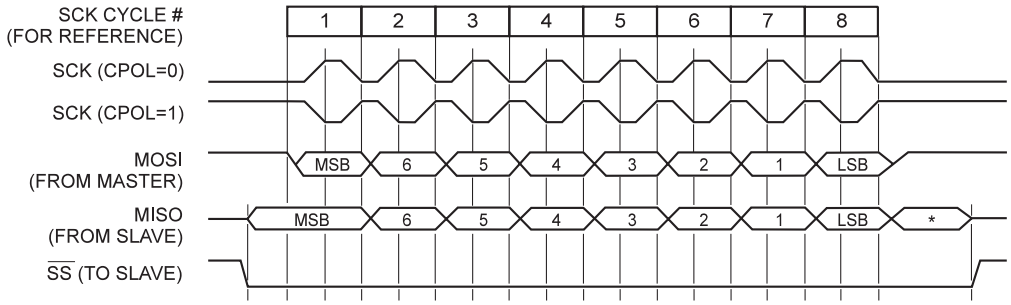
Thus, when interrupt-driven SPI transmission is used in Master mode, and there exists a possibility that $\overline{SS}$ is driven low. The interrupt should always check that the MSTR bit is still set. Once the MSTR bit has been cleared by a Slave select, it must be set by the user to re-enable SPI Master mode.

When the SPI is configured as a Slave, the $\overline{SS}$ pin is always input. When $\overline{SS}$ is held low, the SPI is activated and MISO becomes an output if configured so by the user. All other pins are inputs. When $\overline{SS}$ is driven high, all pins are inputs and the SPI is passive, which means that it will not receive incoming data. Note that the SPI logic will be reset once the $\overline{SS}$ pin is brought high. If the $\overline{SS}$ pin is brought high during a transmission, the SPI will stop sending and receiving immediately and both data received and data sent must be considered lost.

## Data Modes

There are four combinations of SCK phase and polarity with respect to serial data, which are determined by control bits CPHA and CPOL. The SPI data transfer formats are shown in Figure 42 and Figure 43.

**Figure 42.** SPI Transfer Format with CPHA = 0 and DORD = 0




* Not defined but normally MSB of character just received

### UART0 and UART1 High Byte Baud Rate Register UBRRHI

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $20 ($40) | MSB1 | | | LSB1 | MSB0 | | | LSB0 | UBRRHI |
| 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 UART Baud Register is a 12-bit register. The four most significant bits are located in a separate register, UBRRHI. Note that both UART0 and UART1 share this register. Bit 7 to bit 4 of UBRRHI contain the four most significant bits of the UART1 Baud Register. Bit 3 to bit 0 contain the four most significant bits of the UART0 Baud Register.

### UART0 Baud Rate Register Low Byte – UBRR0

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $09 ($29) | MSB | | | | | | | LSB | UBRR0 |
| 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 | |

### UART1 Baud Rate Register Low Byte – UBRR1

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $00 ($20) | MSB | | | | | | | LSB | UBRR1 |
| 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 | |

UBRRn stores the eight least significant bits of the UART Baud Rate Register.

## Double-speed Transmission

The ATmega161 provides a separate UART mode that allows the user to double the communication speed. By setting the U2X bit in UART Control and Status Register UCSRnA, the UART speed will be doubled. The data reception will differ slightly from Normal mode. Since the speed is doubled, the Receiver front-end logic samples the signals on RXDn pin at a frequency 8 times the baud rate. While the line is idle, one single sample of logical “0” will be interpreted as the falling edge of a start bit, and the start bit detection sequence is initiated. Let sample 1 denote the first zero-sample. Following the 1-to-0 transition, the Receiver samples the RXDn pin at samples 4, 5 and 6. If two or more of these three samples are found to be logical “1”s, the start bit is rejected as a noise spike and the Receiver starts looking for the next 1-to-0 transition.

If, however, a valid start bit is detected, sampling of the data bits following the start bit is performed. These bits are also sampled at samples 4, 5 and 6. The logical value found in at least two of the three samples is taken as the bit value. All bits are shifted into the Transmitter Shift Register as they are sampled. Sampling of an incoming character is shown in Figure 47.

**Figure 47.** Sampling Received Data when the Transmission Speed is Doubled

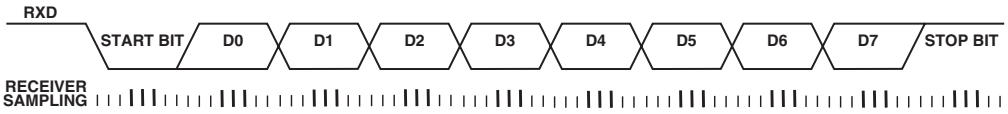


**Table 25.** UBR Settings at Various Crystal Frequencies in Double-speed Mode

| Baud Rate | 1.0000 MHz | % Error | 1.8432 MHz | % Error | 2.0000 MHz | % Error |
|---|---|---|---|---|---|---|
| **2400** | UBR = **51** | 0.2 | UBR = **95** | 0.0 | UBR = **103** | 0.2 |
| **4800** | UBR = **25** | 0.2 | UBR = **47** | 0.0 | UBR = **51** | 0.2 |
| **9600** | UBR = **12** | 0.2 | UBR = **23** | 0.0 | UBR = **25** | 0.2 |
| **14400** | **UBR = 8** | 3.7 | UBR = **15** | 0.0 | UBR = 16 | 2.1 |
| **19200** | **UBR = 6** | 7.5 | UBR = **11** | 0.0 | UBR = **12** | 0.2 |
| **28800** | **UBR = 3** | 7.8 | UBR = **7** | 0.0 | UBR = 8 | 3.7 |
| **38400** | **UBR = 2** | 7.8 | UBR = **5** | 0.0 | UBR = 6 | 7.5 |
| **57600** | **UBR = 1** | 7.8 | UBR = **3** | 0.0 | UBR = 3 | 7.8 |
| **76800** | **UBR = 1** | 22.9 | UBR = **2** | 0.0 | UBR = 2 | 7.8 |
| **115200** | **UBR = 0** | 84.3 | UBR = **1** | 0.0 | UBR = 1 | 7.8 |
| **230400** | **-** | - | UBR = **0** | 0.0 | UBR = 0 | 84.3 |
| Baud Rate | 3.2768 MHz | % Error | 3.6864 MHz | % Error | 4.0000 MHz | % Error |
| **2400** | UBR = **170** | 0.2 | UBR = **191** | 0.0 | UBR = **207** | 0.2 |
| **4800** | UBR = **84** | 0.4 | UBR = **95** | 0.0 | UBR = **103** | 0.2 |
| **9600** | UBR = **42** | 0.8 | UBR = **47** | 0.0 | UBR = **51** | 0.2 |
| **14400** | UBR = 27 | 1.6 | UBR = **31** | 0.0 | UBR = **34** | 0.8 |
| **19200** | UBR = 20 | 1.6 | UBR = **23** | 0.0 | UBR = **25** | 0.2 |
| **28800** | UBR = 13 | 1.6 | UBR = **15** | 0.0 | UBR = 16 | 2.1 |
| **38400** | UBR = 10 | 3.1 | UBR = **11** | 0.0 | UBR = **12** | 0.2 |
| **57600** | UBR = 6 | 1.6 | UBR = **7** | 0.0 | UBR = 8 | 3.7 |
| **76800** | UBR = 4 | 6.2 | UBR = **5** | 0.0 | UBR = 6 | 7.5 |
| **115200** | UBR = 3 | 12.5 | UBR = **3** | 0.0 | UBR = 3 | 7.8 |
| **230400** | UBR = 1 | 12.5 | UBR = **1** | 0.0 | UBR = 1 | 7.8 |
| **460800** | UBR = 0 | 12.5 | UBR = **0** | 0.0 | UBR = 0 | 7.8 |
| **912600** | - | - | - | - | UBR = 0 | 84.3 |
| Baud Rate | 7.3728 MHz | % Error | 8.0000 MHz | % Error | 9.2160 MHz | % Error |
| 2400 | UBR = **383** | 0.0 | UBR = **416** | 0.1 | UBR = **479** | 0.0 |
| 4800 | UBR = **191** | 0.0 | UBR = **207** | 0.2 | UBR = **239** | 0.0 |
| 9600 | UBR = **95** | 0.0 | UBR = **103** | 0.2 | UBR = **119** | 0.0 |
| 14400 | UBR = **63** | 0.0 | UBR = **68** | 0.6 | UBR = **79** | 0.0 |
| 19200 | UBR = **47** | 0.0 | UBR = **51** | 0.2 | UBR = **59** | 0.0 |
| 28800 | UBR = **31** | 0.0 | UBR = **34** | 0.8 | UBR = **39** | 0.0 |
| 38400 | UBR = **23** | 0.0 | UBR = **25** | 0.2 | UBR = **29** | 0.0 |
| 57600 | UBR = **15** | 0.0 | UBR = 16 | 2.1 | UBR = **19** | 0.0 |
| 76800 | UBR = **11** | 0.0 | UBR = 12 | 0.2 | UBR = **14** | 0.0 |
| 115200 | UBR = **7** | 0.0 | UBR = 8 | 3.7 | UBR = **9** | 0.0 |
| **230400** | UBR = **3** | 0.0 | UBR = 3 | 7.8 | UBR = **4** | 0.0 |
| **460800** | UBR = **1** | 0.0 | UBR = 1 | 7.8 | UBR = 2 | 20.0 |
| **912600** | UBR = **0** | 0.0 | UBR = 0 | 7.8 | UBR = 0 | 20.0 |

**Figure 49.** External Memory with Page Select

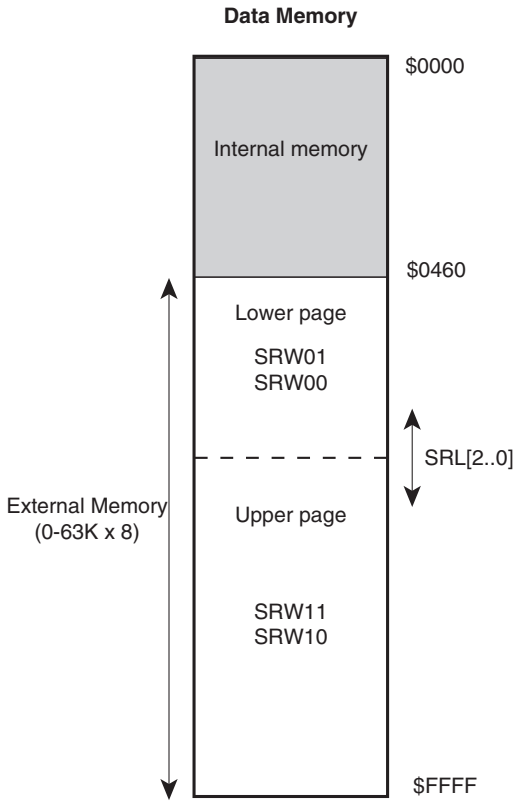
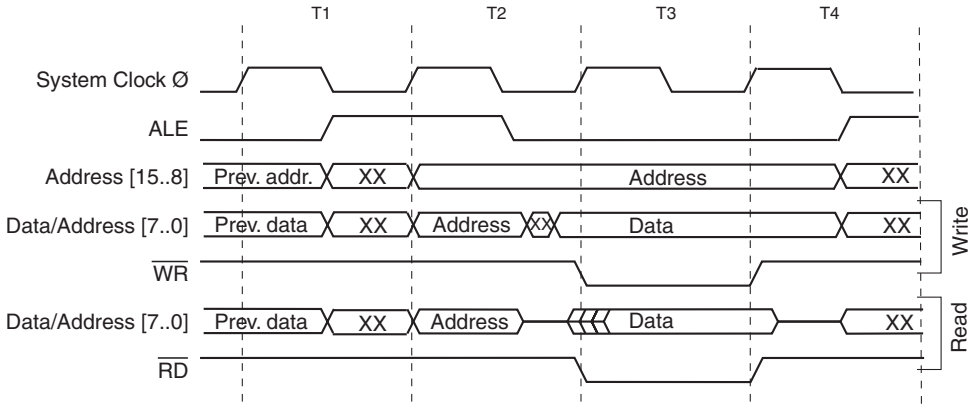
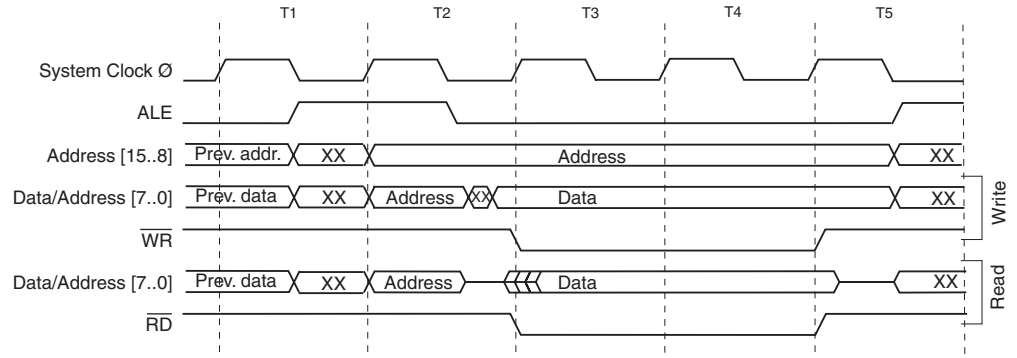



**Figure 50.** External Data Memory Cycles without Wait State (SRWn1 = 0 and SRWn0 =0)[1]



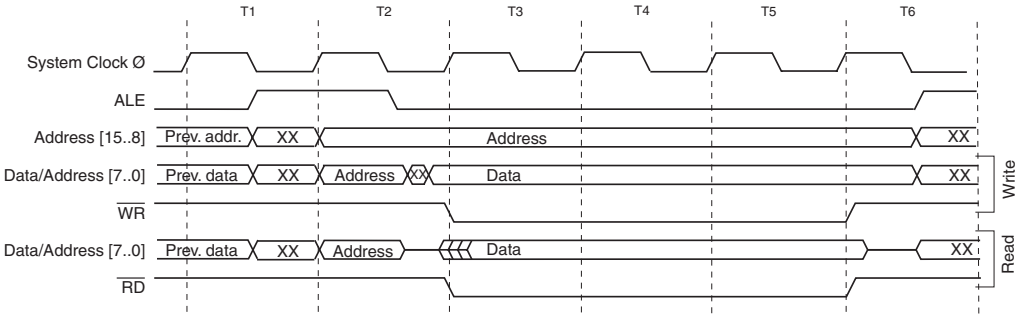

Note: 1. SRWn1 = SRW11 (upper page) or SRW01 (lower page), SRWn0 = SRW10 (upper page) or SRW00 (lower page).
The ALE pulse in period T4 is only present if the next instruction accesses the RAM (internal or external). The Data and Address will only change in T4 if ALE is present (the next instruction accesses the RAM).

**Figure 51.** External Data Memory Cycles with SRWn1 = 0 and SRWn0 = 1[1]



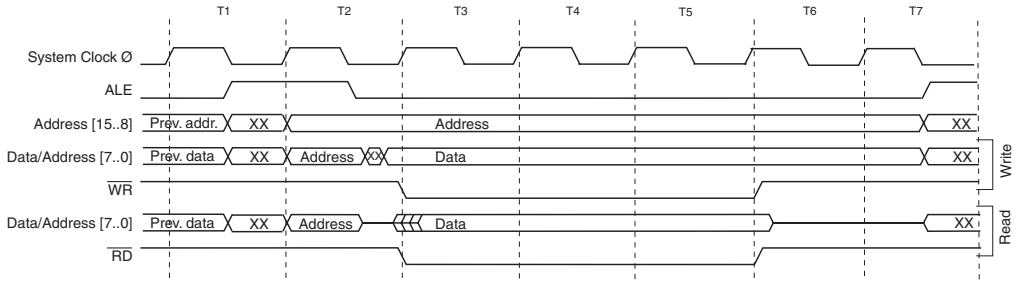

Note: 1. SRWn1 = SRW11 (upper page) or SRW01 (lower page), SRWn0 = SRW10 (upper page) or SRW00 (lower page). The ALE pulse in period T5 is only present if the next instruction accesses the RAM (internal or external). The Data and Address will only change in T5 if ALE is present (the next instruction accesses the RAM).

**Figure 52.** External Data Memory Cycles with SRWn1 = 1 and SRWn0 = 0[1]




Note: 1. SRWn1 = SRW11 (upper page) or SRW01 (lower page), SRWn0 = SRW10 (upper page) or SRW00 (lower page). The ALE pulse in period T6 is only present if the next instruction accesses the RAM (internal or external). The Data and Address will only change in T6 if ALE is present (the next instruction accesses the RAM).

**Figure 53.** External Data Memory Cycles with SRWn1 = 1 and SRWn0 = 1[1]




Note: 1. SRWn1 = SRW11 (upper page) or SRW01 (lower page), SRWn0 = SRW10 (upper page) or SRW00 (lower page). The ALE pulse in period T7 is only present if the next instruction accesses the RAM (internal or external). The Data and Address will only change in T7 if ALE is present (the next instruction accesses the RAM).

## Using the External Memory Interface

The interface consists of:

Port A: multiplexed low-order address bus and data bus

Port C: high-order address bus

The ALE pin: address latch enable

The $\overline{RD}$ and $\overline{WR}$ pin: read and write strobes

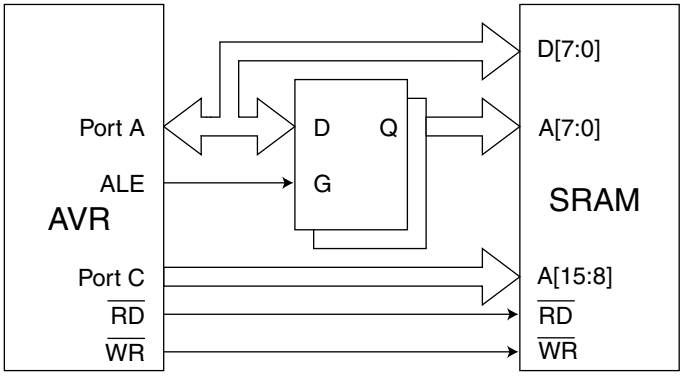
The external memory interface is enabled by setting the external SRAM enable bit (SRE) of the MCU Control Register (MCUCR) and will override the setting of the Data Direction Registers DDRA, DDRD and DDRE. When the SRE bit is cleared (zero), the external memory interface is disabled and the normal pin and data direction settings are used. When SRE is low, the address space above the internal SRAM boundary is not mapped into the internal SRAM, as in AVR parts do not have an external memory interface.

When ALE goes from high to low, there is a valid address on Port A. ALE is low during a data transfer. $\overline{RD}$ and $\overline{WR}$ are active when accessing the external memory only.

When the external memory interface is enabled, the ALE signal may have short pulses when accessing the internal RAM, but the ALE signal is stable when accessing the external memory.

Figure 54 sketches how to connect an external SRAM to the AVR using eight latches that are transparent when G is high.

**Figure 54.** External SRAM Connected to the AVR




For details on the timing for the SRAM interface, please see Figure 83 through Figure 86 and Table 51 through Table 58.

## I/O Ports

All AVR ports have true read-modify-write functionality when used as general digital I/O ports. This means that the direction of one port pin can be changed without unintentionally changing the direction of any other pin with the SBI and CBI instructions. The same applies for changing drive value (if configured as output) or enabling/disabling of pull-up resistors (if configured as input).

## Port A

Port A is an 8-bit bi-directional I/O port.

Three I/O memory address locations are allocated for the Port A, one each for the Data Register – PORTA, $1B($3B), Data Direction Register – DDRA, $1A($3A) and the Port A Input Pins – PINA, $19($39). The Port A Input Pins address is read-only, while the Data Register and the Data Direction Register are read/write.

All port pins have individually selectable pull-up resistors. The Port A output buffers can sink 20 mA and thus drive LED displays directly. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated.

The Port A pins have alternate functions related to the optional external memory interface. Port A can be configured to be the multiplexed low-order address/data bus during accesses to the external Data memory. In this mode, Port A has internal pull-up resistors.

When Port A is set to the alternate function by the SRE (External SRAM Enable) bit in the MCUCR (MCU Control Register), the alternate settings override the Data Direction Register.

### Port A Data Register – PORTA

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $1B ($3B) | PORTA7 | PORTA6 | PORTA5 | PORTA4 | PORTA3 | PORTA2 | PORTA1 | PORTA0 | PORTA |
| 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 | |

### Port A Data Direction Register – DDRA

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $1A ($3A) | DDA7 | DDA6 | DDA5 | DDA4 | DDA3 | DDA2 | DDA1 | DDA0 | DDRA |
| 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 | |

### Port A Input Pins Address – PINA

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $19 ($39) | PINA7 | PINA6 | PINA5 | PINA4 | PINA3 | PINA2 | PINA1 | PINA0 | PINA |
| Read/Write | R | R | R | R | R | R | R | R | |
| Initial Value | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | |

The Port A Input Pins address (PINA) is not a register; this address enables access to the physical value on each Port A pin. When reading PORTA, the Port A Data Latch is read and when reading PINA, the logical values present on the pins are read.

### Port A as General Digital I/O

All eight pins in Port A have equal functionality when used as digital I/O pins.

PAn, general I/O pin: The DDAn bit in the DDRA Register selects the direction of this pin. If DDAn is set (one), PAn is configured as an output pin. If DDAn is cleared (zero), PAn is configured as an input pin. If PORTAn is set (one) when the pin is configured as an input pin, the MOS pull-up resistor is activated. To switch the pull-up resistor off, the PORTAn has to be cleared (zero) or the pin has to be configured as an output pin. The

## Port B Schematics

Note that all port pins are synchronized. The synchronization latches are, however, not shown in the figures.

**Figure 56.** Port B Schematic Diagram (Pins PB0 and PB1)

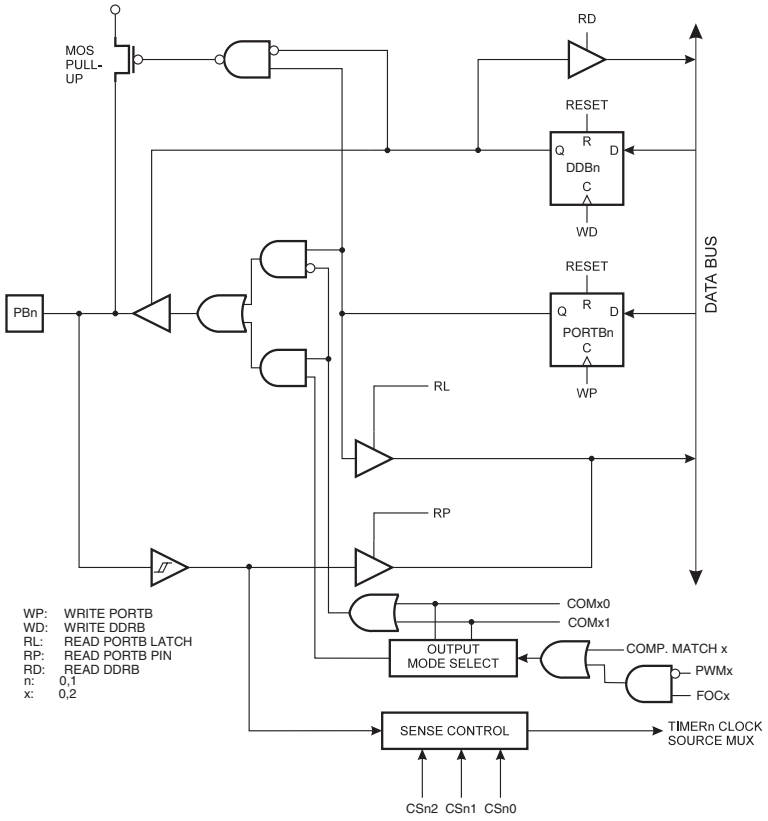


**Figure 61.** Port B Schematic Diagram (Pin PB6)

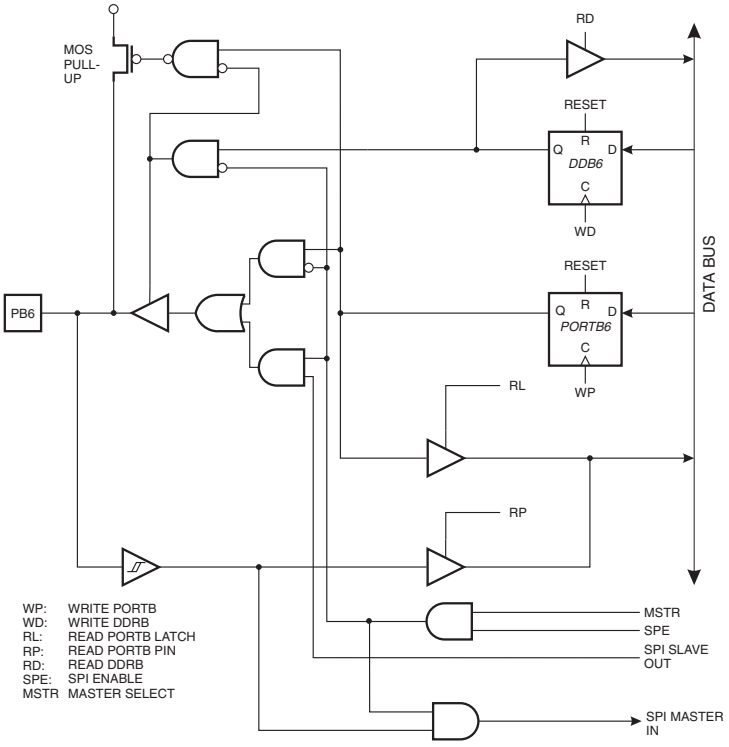



**Figure 62.** Port B Schematic Diagram (Pin PB7)

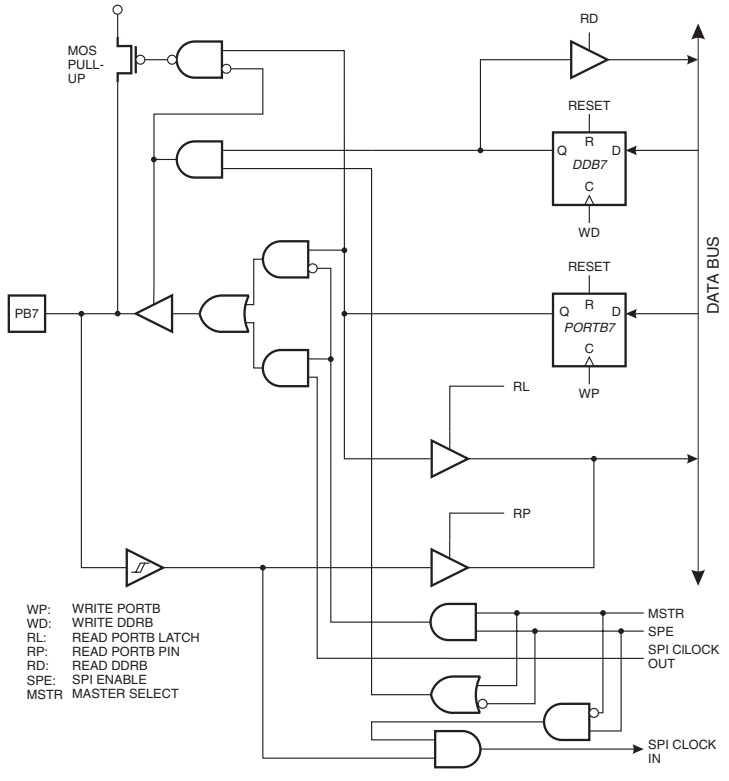


**Table 32.** DDCn Effects on Port C Pins[1]

| DDCn | PORTCn | I/O | Pull-up | Comment |
|---|---|---|---|---|
| 0 | 0 | Input | No | Tri-state (high-Z) |
| 0 | 1 | Input | Yes | PCn will source current if ext. pulled low |
| 1 | 0 | Output | No | Push-pull Zero Output |
| 1 | 1 | Output | No | Push-pull One Output |

Note: 1. n: 7, 6,…0, pin number

## Port C Schematics

Note that all port pins are synchronized. The synchronization latch is, however, not shown in the figure.

**Figure 63.** Port C Schematic Diagram (Pins PC0 - PC7)

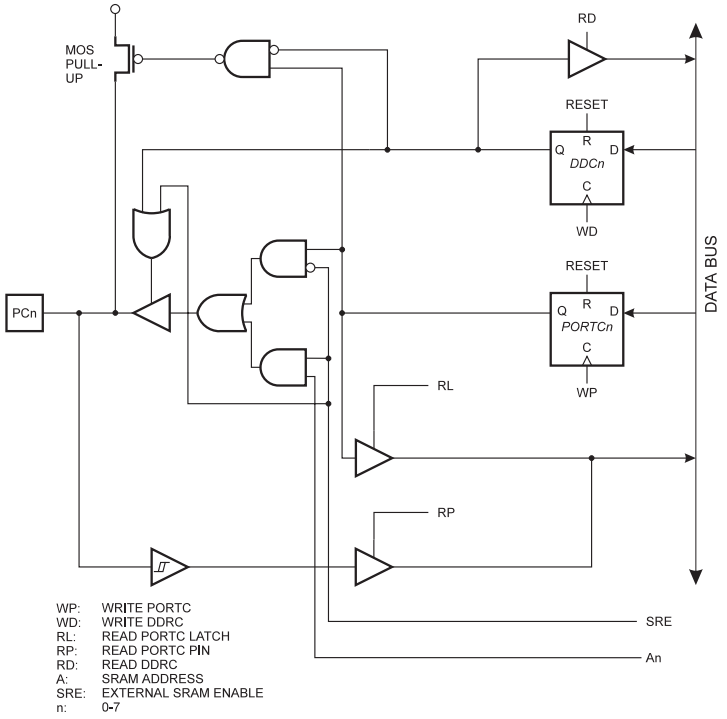


**Figure 68.** Port D Schematic Diagram (Pin PD5)

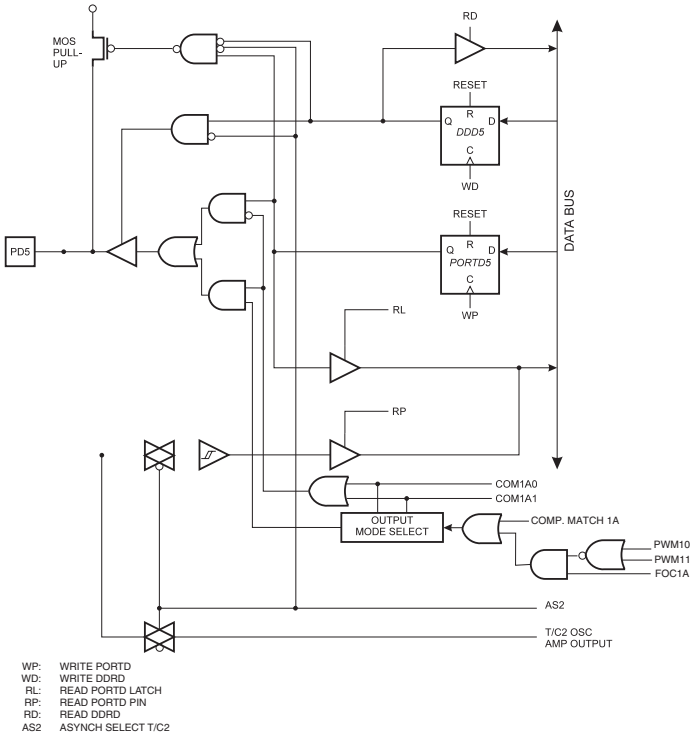



**Figure 69.** Port D Schematic Diagram (Pin PD6)

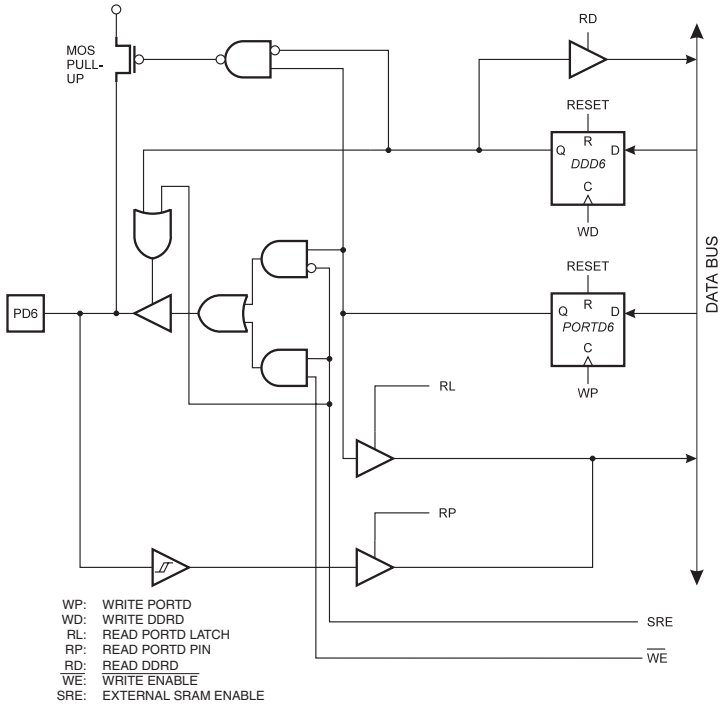


**Table 39.** Boot Reset Fuse, BOOTRST[1]

| BOOTRST | Reset Address |
|---|---|
| 1 | Reset Vector = Application Reset (address $0000) |
| 0 | Reset Vector = Boot Loader Reset (address $1E00) |

Note: 1. "1" means unprogrammed, "0" means programmed

## Capabilities of the Boot Loader

The program code within the Boot Loader section has the ability to read from and write into the entire Flash, including the Boot Loader Memory. This allows the user to update both the Application code and the Boot Loader code that handles the software update. The Boot Loader can thus even modify itself, and it can also erase itself from the code if the feature is not needed anymore. Special care must be taken if the user allows the Boot Loader section to be updated by leaving Boot Lock bit11 unprogrammed. An accidental write to the Boot Loader itself can corrupt the entire Boot Loader, and further software updates might be impossible. If it is not needed to change the Boot Loader software itself, it is recommended that the Boot Lock bit 11 be programmed to protect the Boot Loader software from software changes.

## Self-programming the Flash

Programming of the Flash is executed one page at a time. The Flash page must be erased first for correct programming. The general Write Lock (Lock Bit 2) does not control the programming of the Flash memory by SPM instruction. Similarly, the Read/Write Lock (Lock Bit 1) does not control reading or writing by LPM/SPM, if it is attempted.

The Program memory can only be updated page-by-page, not word by word. One page is 128 bytes (64 words). The Program memory will be modified by first performing page erase, then by filling the temporary page buffer one word at a time using SPM, and then by executing page write. If only part of the page needs to be changed, the other parts must be stored (for example, in the temporary page buffer) before the erase, and then be rewritten. The temporary page buffer can be accessed in a random sequence. The CPU is halted both during page erase and during page write and the SPMEN bit in the SPMCR Register will be auto-cleared. For future compatibility, however, it is recommended that the user software verify that the SPMEN bit is cleared before starting a new page erase, page write, or before writing the Lock bits command (see code examples below). It is essential that the page address used in both the page erase and page write operation is addressing the same page.

### Setting the Boot Loader Lock bits by SPM

To set the Boot Loader Lock bits, write the desired data to R0, write "1001" to SPMCR, and execute SPM within four clock cycles after writing SPMCR. The only accessible Lock bits are the Boot Lock bits that may prevent the Application Code and Boot Loader sections from any software update by the MCU. See Table 37 and Table 38 for how the different settings of the Boot Loader bits affect the Flash access.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| | – | – | BLB12 | BLB11 | BLB02 | BLB01 | – | – | R0 |

If bit5 - bit2 in R0 is cleared (zero), the corresponding Boot Lock bit will be programmed if an SPM instruction is executed within four cycles after BLBSET and SPMEN are set in SPMCR.

### Performing Page Erase by SPM

To execute a page erase, set up the address in the Z-pointer, write "0011" to SPMCR, and execute SPM within four clock cycles after writing SPMCR. The data in R1 and R0 are ignored. The page address must be written to Z13:Z7. Other bits in the Z-pointer will be ignored during this operation.

**Table 45.** Parallel Programming Characteristics, $T_A$ = 25°C ± 10%, $V_{CC}$ = 5V ± 10%[1][2][3]

| Symbol | Parameter | Min | Typ | 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 |
| $t_{XHXL}$ | XTAL1 Pulse Width High | 67 | | | ns |
| $t_{XLDX}$ | Data and Control Hold after XTAL1 Low | 67 | | | ns |
| $t_{XLWL}$ | XTAL1 Low to $\overline{WR}$ Low | 67 | | | ns |
| $t_{BVXH}$ | BS1 Valid before XTAL1 High | 67 | | | ns |
| $t_{PHPL}$ | PAGEL Pulse Width High | 67 | | | ns |
| $t_{PLBX}$ | BS1 Hold after PAGEL Low | 67 | | | ns |
| $t_{PLWL}$ | PAGEL Low to $\overline{WR}$ Low | 67 | | | ns |
| $t_{BVWL}$ | BS1 Valid to $\overline{WR}$ Low | 67 | | | ns |
| $t_{RHBX}$ | BS1 Hold after RDY/$\overline{BSY}$ High | 67 | | | ns |
| $t_{WLWH}$ | $\overline{WR}$ Pulse Width Low | 67 | | | ns |
| $t_{WLRL}$ | $\overline{WR}$ Low to RDY/$\overline{BSY}$ Low | 0 | | 2.5 | µs |
| $t_{WLRH}$ | $\overline{WR}$ Low to RDY/$\overline{BSY}$ High[1] | 1 | | 1.7 | ms |
| $t_{WLRH\_CE}$ | $\overline{WR}$ Low to RDY/$\overline{BSY}$ High for Chip Erase[2] | 16 | | 28 | ms |
| $t_{WLRH\_FLASH}$ | $\overline{WR}$ Low to RDY/$\overline{BSY}$ High for Write Flash[3] | 8 | | 14 | ms |
| $t_{XLOL}$ | XTAL1 Low to $\overline{OE}$ Low | 67 | | | ns |
| $t_{OLDV}$ | $\overline{OE}$ Low to DATA Valid | | 20 | | ns |
| $t_{OHDZ}$ | $\overline{OE}$ High to DATA Tri-stated | | | 20 | ns |

Notes: 1. $t_{WLRH}$ is valid for the Write EEPROM, Write Fuse bits and Write Lock bits commands.
2. $t_{WLRH\_CE}$ is valid for the Chip Erase command.
3. $t_{WLRH\_FLASH}$ is valid for the Write Flash command.

## Serial Downloading

Both the Flash and EEPROM memory arrays can be programmed using the serial SPI bus while $\overline{RESET}$ is pulled to GND. The serial interface consists of pins SCK, MOSI (input) and MISO (output). After $\overline{RESET}$ is set low, the Programming Enable instruction needs to be executed first, before program/erase operations can be executed.

When programming the EEPROM, an auto-erase cycle is built into the self-timed programming operation (in the Serial mode ONLY) and there is no need to first execute the Chip Erase instruction. The chip erase operation turns the contents of every memory location in both the program and EEPROM arrays into $FF.

The program and EEPROM memory arrays have separate address spaces:

$0000 to $1FFF for Program memory and $0000 to $01FF for EEPROM memory.

Either an external system clock is supplied at pin XTAL1 or a crystal needs to be connected across pins XTAL1 and XTAL2. The minimum low and high periods for the serial clock (SCK) input are defined as follows:

Low: > 2 XTAL1 clock cycles

High: > 2 XTAL1 clock cycles

**Figure 85.** External Memory Timing (SRWn1 = 1, SRWn0 = 0)

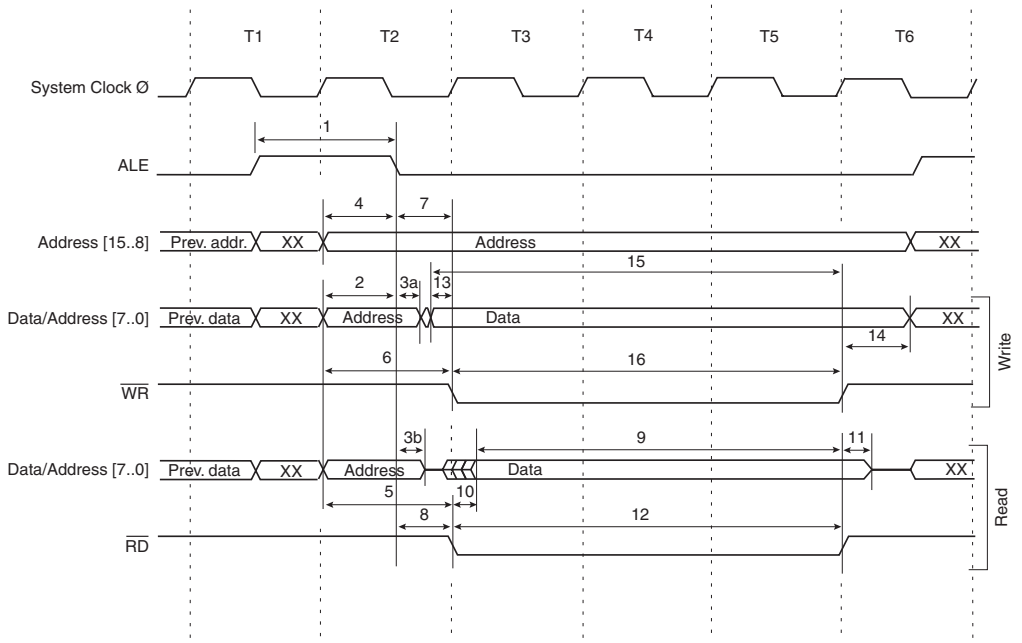
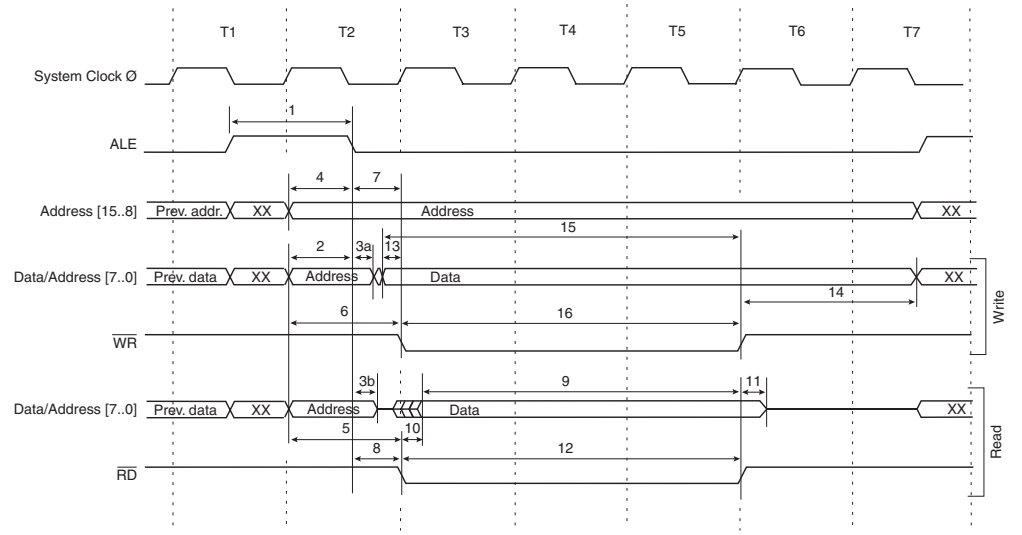


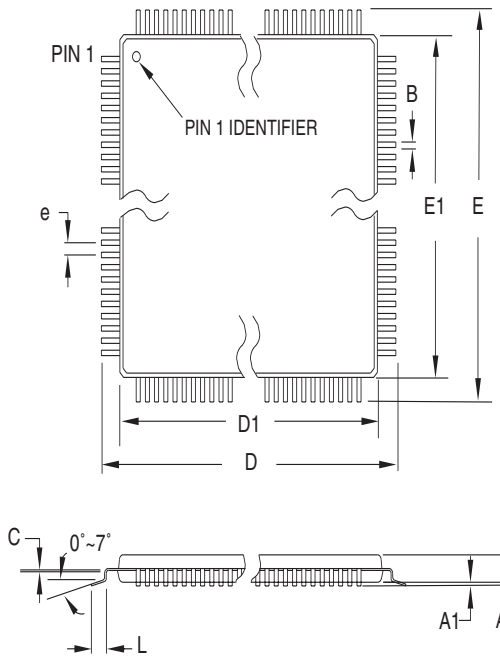

**Figure 86.** External Memory Timing (SRWn1 = 1, SRWn0 = 1)[1]




Note: 1. The ALE pulse in the last period (T4 - T7) is only present if the next instruction accesses the RAM (internal or external). The data and address will only change in T4 - T7 if ALE is present (the next instruction accesses the RAM).

## Packaging Information

### 44A




**COMMON DIMENSIONS**
(Unit of Measure = mm)

| SYMBOL | MIN | NOM | MAX | NOTE |
|---|---|---|---|---|
| A | – | – | 1.20 | |
| A1 | 0.05 | – | 0.15 | |
| A2 | 0.95 | 1.00 | 1.05 | |
| D | 11.75 | 12.00 | 12.25 | |
| D1 | 9.90 | 10.00 | 10.10 | Note 2 |
| E | 11.75 | 12.00 | 12.25 | |
| E1 | 9.90 | 10.00 | 10.10 | Note 2 |
| B | 0.30 | – | 0.45 | |
| C | 0.09 | – | 0.20 | |
| L | 0.45 | – | 0.75 | |
| e | 0.80 TYP | | | |

Notes:
1. This package conforms to JEDEC reference MS-026, Variation ACB.
2. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is 0.25 mm per side. Dimensions D1 and E1 are maximum plastic body size dimensions including mold mismatch.
3. Lead coplanarity is 0.10 mm maximum.

10/5/2001

| | TITLE | DRAWING NO. | REV. |
|---|---|---|---|
| 2325 Orchard Parkway<br>San Jose, CA 95131 | **44A,** 44-lead, 10 x 10 mm Body Size, 1.0 mm Body Thickness, 0.8 mm Lead Pitch, Thin Profile Plastic Quad Flat Package (TQFP) | 44A | B |