



Welcome to [E-XFL.COM](https://www.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                | PWM, WDT  |
| Number of I/O              | 32  |
| Program Memory Size        | 4KB (2K x 16)   |
| Program Memory Type        | FLASH   |
| EEPROM Size                | 256 x 8   |
| RAM Size                   | 256 x 8   |
| Voltage - Supply (Vcc/Vdd) | 2.7V ~ 6V   |
| Data Converters            | -   |
| Oscillator Type            | Internal  |
| Operating Temperature      | -40°C ~ 85°C  |
| Mounting Type              | Surface Mount   |
| Package / Case             | 44-TQFP   |
| Supplier Device Package    | 44-TQFP (10x10)   |
| Purchase URL               | <a href="https://www.e-xfl.com/product-detail/microchip-technology/at90s4414-4ai">https://www.e-xfl.com/product-detail/microchip-technology/at90s4414-4ai</a> |

**Figure 20.** The Parallel Instruction Fetches and Instruction Executions

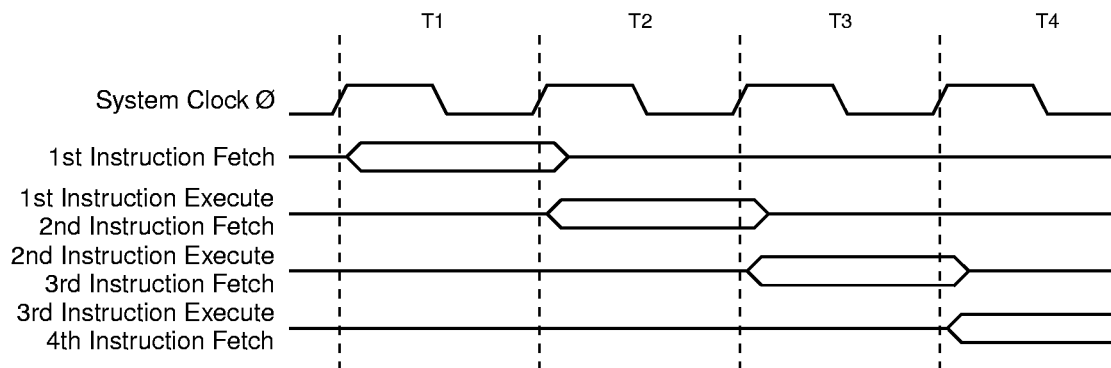
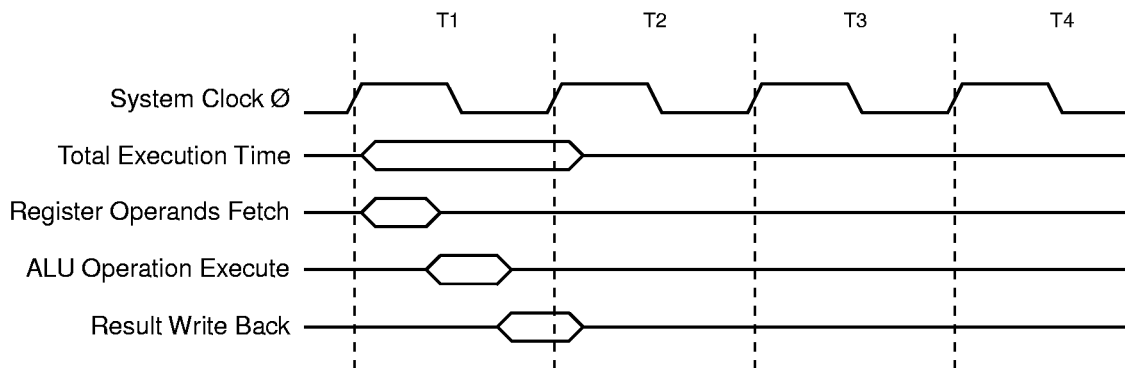


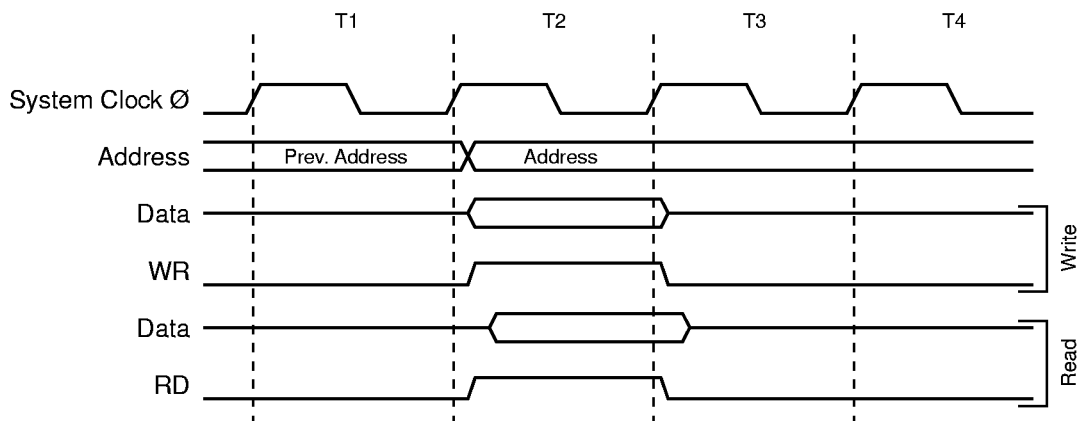
Figure 21 shows the internal timing concept for the register file. In a single clock cycle an ALU operation using two register operands is executed, and the result is stored back to the destination register.

**Figure 21.** Single Cycle ALU Operation



The internal data SRAM access is performed in two System Clock cycles as described in Figure 22.

**Figure 22.** On-chip Data SRAM Access Cycles



See "Interface to External SRAM" on page 53 for a description of the access to the external SRAM.

## I/O Memory

The I/O space definition of the AT90S4414/8515 is shown in the following table:

**Table 2.** AT90S4414/8515 I/O Space

| Address Hex | Name   | Function   |
|-------------|--------|--|
| \$3F (\$5F) | SREG   | Status Register                                    |
| \$3E (\$5E) | SPH    | Stack Pointer High                                 |
| \$3D (\$5D) | SPL    | Stack Pointer Low                                  |
| \$3B (\$5B) | GIMSK  | General Interrupt Mask register                    |
| \$3A (\$5A) | GIFR   | General Interrupt Flag Register                    |
| \$39 (\$59) | TIMSK  | Timer/Counter Interrupt Mask register              |
| \$38 (\$58) | TIFR   | Timer/Counter Interrupt Flag register              |
| \$35 (\$55) | MCUCR  | MCU general Control Register                       |
| \$33 (\$53) | TCCR0  | Timer/Counter0 Control Register                    |
| \$32 (\$52) | TCNT0  | Timer/Counter0 (8-bit)                             |
| \$2F (\$4F) | TCCR1A | Timer/Counter1 Control Register A                  |
| \$2E (\$4E) | TCCR1B | Timer/Counter1 Control Register B                  |
| \$2D (\$4D) | TCNT1H | Timer/Counter1 High Byte                           |
| \$2C (\$4C) | TCNT1L | Timer/Counter1 Low Byte                            |
| \$2B (\$4B) | OCR1AH | Timer/Counter1 Output Compare Register A High Byte |
| \$2A (\$4A) | OCR1AL | Timer/Counter1 Output Compare Register A Low Byte  |
| \$29 (\$49) | OCR1BH | Timer/Counter1 Output Compare Register B High Byte |
| \$28 (\$48) | OCR1BL | Timer/Counter1 Output Compare Register B Low Byte  |
| \$25 (\$45) | ICR1H  | T/C 1 Input Capture Register High Byte             |
| \$24 (\$44) | ICR1L  | T/C 1 Input Capture Register Low Byte              |
| \$21 (\$41) | WDTCR  | Watchdog Timer Control Register                    |
| \$1F (\$3E) | EEARH  | EEPROM Address Register High Byte (AT90S8515)      |
| \$1E (\$3E) | EEARL  | EEPROM Address Register Low Byte                   |
| \$1D (\$3D) | EEDR   | EEPROM Data Register                               |
| \$1C (\$3C) | EEDR   | EEPROM Control Register                            |
| \$1B (\$3B) | PORTA  | Data Register, Port A                              |
| \$1A (\$3A) | DDRA   | Data Direction Register, Port A                    |
| \$19 (\$39) | PINA   | Input Pins, Port A                                 |
| \$18 (\$38) | PORTB  | Data Register, Port B                              |
| \$17 (\$37) | DDRB   | Data Direction Register, Port B                    |
| \$16 (\$36) | PINB   | Input Pins, Port B                                 |
| \$15 (\$35) | PORTC  | Data Register, Port C                              |
| \$14 (\$34) | DDRC   | Data Direction Register, Port C                    |
| \$13 (\$33) | PINC   | Input Pins, Port C                                 |

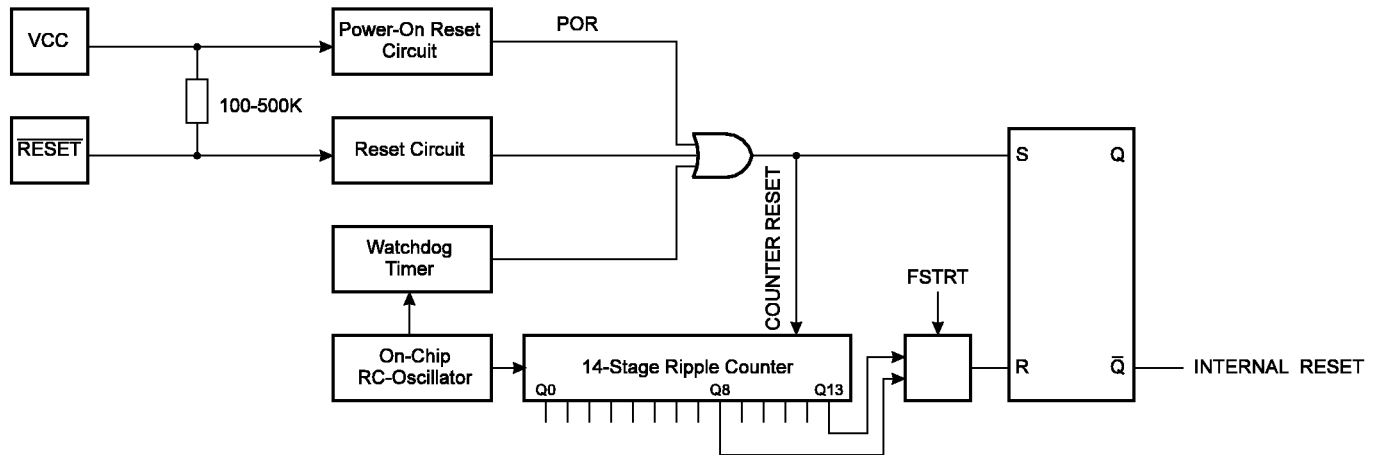
## Reset Sources

The AT90S4414/8515 has three sources of reset:

- Power-On Reset. The MCU is reset when the supply voltage is below the power-on reset threshold ( $V_{POT}$ ).
- External Reset. The MCU is reset when a low level is present on the  $\overline{RESET}$  pin for more than 50 ns.
- Watchdog Reset. The MCU is reset when the Watchdog timer period expires and the Watchdog is enabled.

During reset, all I/O registers are then set to their initial values, and the program starts execution from address \$000. The instruction placed in address \$000 must be an RJMP - relative jump - instruction to the reset handling routine. If the program never enables an interrupt source, the interrupt vectors are not used, and regular program code can be placed at these locations. The circuit diagram in Figure 23 shows the reset logic. Table 4 defines the timing and electrical parameters of the reset circuitry.

**Figure 23.** Reset Logic



**Table 4.** Reset Characteristics

| Symbol          | Parameter                                      | Min | Typ | Max          | Units |
|-----------------|--|-----|-----|--------------|-------|
| $V_{POT}^{(1)}$ | Power-on Reset Threshold Voltage (rising)      | 0.8 | 1.2 | 1.6          | V     |
|                 | Power-on Reset Threshold Voltage (falling)     | 0.2 | 0.4 | 0.6          | V     |
| $V_{RST}$       | $\overline{RESET}$ Pin Threshold Voltage       | -   | -   | $0.9 V_{CC}$ | V     |
| $t_{TOUT}$      | Reset Delay Time-out Period FSTRT Unprogrammed | 11  | 16  | 21           | ms    |
| $t_{TOUT}$      | Reset Delay Time-out Period FSTRT Programmed   | 1.0 | 1.1 | 1.2          | ms    |

Notes: 1. The Power-on Reset will not work unless the supply voltage has been below  $V_{POT}$  (falling).

The user can select the start-up time according to typical oscillator start-up. The number of WDT oscillator cycles used for each time-out is shown in Table 5. The frequency of the watchdog oscillator is voltage dependent as shown in "Typical Characteristics" on page 86.

**Table 5.** Number of Watchdog Oscillator Cycles

| FSTRT        | Time-out at $V_{CC} = 5V$ | Number of WDT cycles |
|--------------|---------------------------|----------------------|
| Programmed   | 1.1 ms                    | 1K                   |
| Unprogrammed | 16.0 ms                   | 16K                  |

## • Bit 6 - INTF0: External Interrupt Flag

When an event on the INT0 pin triggers an interrupt request, INTF0 becomes set (one). If the I-bit in SREG and the INT0 bit in GIMSK are set (one), the MCU will jump to the interrupt vector at address \$001. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it.

## • Bits 5..0 - Res: Reserved bits

These bits are reserved bits in the AT90S4414/8515 and always read as zero.

## Timer/counter Interrupt Mask Register - TMSK

| Bit           | 7     | 6      | 5      | 4 | 3      | 2 | 1     | 0 |      |
|---------------|-------|--------|--------|---|--------|---|-------|---|------|
| \$39 (\$59)   | TOIE1 | OCIE1A | OCIE1B | - | TICIE1 | - | TOIE0 | - | TMSK |
| Read/Write    | R/W   | R/W    | R/W    | R | R/W    | R | R/W   | R |      |
| Initial value | 0     | 0      | 0      | 0 | 0      | 0 | 0     | 0 |      |

## • Bit 7 - TOIE1: Timer/Counter1 Overflow Interrupt Enable

When the TOIE1 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter1 Overflow interrupt is enabled. The corresponding interrupt (at vector \$006) is executed if an overflow in Timer/Counter1 occurs, i.e., when the TOV1 bit is set in the Timer/Counter Interrupt Flag Register - TIFR.

## • Bit 6 - OCIE1A: Timer/Counter1 Output CompareA Match Interrupt Enable

When the OCIE1A bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter1 CompareA Match interrupt is enabled. The corresponding interrupt (at vector \$004) is executed if a CompareA match in Timer/Counter1 occurs, i.e., when the OCF1A bit is set in the Timer/Counter Interrupt Flag Register - TIFR.

## • 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 \$005) is executed if a CompareB match in Timer/Counter1 occurs, i.e., when the OCF1B bit is set in the Timer/Counter Interrupt Flag Register - TIFR.

## • Bit 4 - Res: Reserved bit

This bit is a reserved bit in the AT90S4414/8515 and always reads zero.

## • 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 \$003) 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 - Res: Reserved bit

This bit is a reserved bit in the AT90S4414/8515 and always reads zero.

## • 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 \$007) 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 - Res: Reserved bit

This bit is a reserved bit in the AT90S4414/8515 and always reads zero.

## Timer/Counter Interrupt Flag Register - TIFR

| Bit           | 7    | 6     | 5     | 4 | 3    | 2 | 1    | 0 |      |
|---------------|------|-------|-------|---|------|---|------|---|------|
| \$38 (\$58)   | TOV1 | OCF1A | OCF1B | - | ICF1 | - | TOV0 | - | TIFR |
| Read/Write    | R/W  | R/W   | R/W   | R | R/W  | R | R/W  | R |      |
| Initial value | 0    | 0     | 0     | 0 | 0    | 0 | 0    | 0 |      |

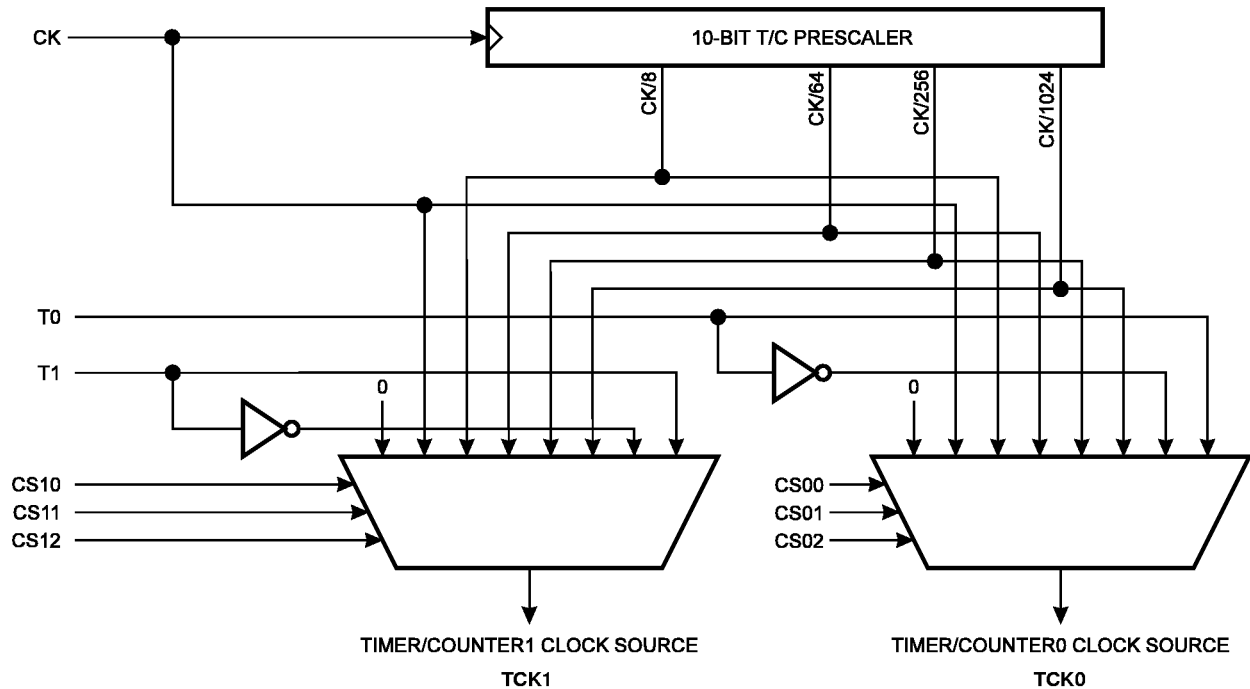
## • Bit 7 - TOV1: Timer/Counter1 Overflow Flag

The TOV1 is set (one) when an overflow occurs in Timer/Counter1. TOV1 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, TOV1 is cleared by writing a logic one to the flag. When the I-bit in SREG, and TOIE1 (Timer/Counter1 Overflow Interrupt Enable), and TOV1 are set (one), the Timer/Counter1 Overflow Interrupt is executed. In PWM mode, this bit is set when Timer/Counter1 changes counting direction at \$0000.

## Timer/Counter Prescaler

Figure 28 shows the general Timer/Counter prescaler.

**Figure 28.** Timer/Counter Prescaler



The four different prescaled selections are: CK/8, CK/64, CK/256 and CK/1024 where CK is the oscillator clock. For the two Timer/Counters, added selections as CK, external source and stop, can be selected as clock sources.

## 8-bit Timer/Counter0

Figure 29 shows the block diagram for Timer/Counter0.

The 8-bit Timer/Counter0 can select clock source from CK, prescaled CK, or an external pin. In addition it can be stopped as described in the specification for the Timer/Counter0 Control Register - TCCR0. The overflow status flag is found in the Timer/Counter Interrupt Flag Register - TIFR. Control signals are found in the Timer/Counter0 Control Register - TCCR0. The interrupt enable/disable settings for Timer/Counter0 are found in the Timer/Counter Interrupt Mask Register - TIMSK.

When Timer/Counter0 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 8-bit Timer/Counter0 features both a high resolution and a high accuracy usage with the lower prescaling opportunities. Similarly, the high prescaling opportunities make the Timer/Counter0 useful for lower speed functions or exact timing functions with infrequent actions.

## • TCNT1 Timer/Counter1 Read:

When the CPU reads the low byte TCNT1L, the data of the low byte TCNT1L is sent to the CPU and the data of the high byte TCNT1H is placed in the TEMP register. When the CPU reads the data in the high byte TCNT1H, the CPU receives the data in the TEMP register. Consequently, the low byte TCNT1L must be accessed first for a full 16-bit register read operation.

The Timer/Counter1 is realized as an up or up/down (in PWM mode) counter with read and write access. If Timer/Counter1 is written to and a clock source is selected, the Timer/Counter1 continues counting in the timer clock cycle after it is preset with the written value.

## Timer/Counter1 Output Compare Register - OCR1AH AND OCR1AL

| Bit           | 15         | 14  | 13  | 12  | 11  | 10  | 9   | 8          |                                |
|---------------|------------|-----|-----|-----|-----|-----|-----|------------|--------------------------------|
| \$2B (\$4B)   | <b>MSB</b> |     |     |     |     |     |     |            | <b>OCR1AH</b><br><b>OCR1AL</b> |
| \$2A (\$4A)   |            |     |     |     |     |     |     | <b>LSB</b> |                                |
|               | 7          | 6   | 5   | 4   | 3   | 2   | 1   | 0          |                                |
| Read/Write    | R/W        | R/W | R/W | R/W | R/W | R/W | R/W | R/W        |                                |
|               | 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          |                                |
|               | 0          | 0   | 0   | 0   | 0   | 0   | 0   | 0          |                                |

## Timer/Counter1 Output Compare Register - OCR1BH AND OCR1BL

| Bit           | 15         | 14  | 13  | 12  | 11  | 10  | 9   | 8          |                                |
|---------------|------------|-----|-----|-----|-----|-----|-----|------------|--------------------------------|
| \$29 (\$49)   | <b>MSB</b> |     |     |     |     |     |     |            | <b>OCR1BH</b><br><b>OCR1BL</b> |
| \$28 (\$48)   |            |     |     |     |     |     |     | <b>LSB</b> |                                |
|               | 7          | 6   | 5   | 4   | 3   | 2   | 1   | 0          |                                |
| Read/Write    | R/W        | R/W | R/W | R/W | R/W | R/W | R/W | R/W        |                                |
|               | 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          |                                |
|               | 0          | 0   | 0   | 0   | 0   | 0   | 0   | 0          |                                |

The output compare registers are 16-bit read/write registers.

The Timer/Counter1 Output Compare Registers contain the data to be continuously compared with Timer/Counter1. Actions on compare matches are specified in the Timer/Counter1 Control and Status register. A compare match does only occur if Timer/Counter1 counts to the OCR value. A software write that sets TCNT1 and OCR1A or OCR1B to the same value does not generate a compare match.

A compare match will set the compare interrupt flag in the CPU clock cycle following the compare event.

Since the Output Compare Registers - OCR1A and OCR1B - are 16-bit registers, a temporary register TEMP is used when OCR1A/B are written to ensure that both bytes are updated simultaneously. When the CPU writes the high byte, OCR1AH or OCR1BH, the data is temporarily stored in the TEMP register. When the CPU writes the low byte, OCR1AL or OCR1BL, the TEMP register is simultaneously written to OCR1AH or OCR1BH. Consequently, the high byte OCR1AH or OCR1BH must be written first for a full 16-bit register write operation.

The TEMP register is also used when accessing TCNT1, and ICR1. If the main program and also interrupt routines perform access to registers using TEMP, interrupts must be disabled during access from the main program (and from interrupt routines if interrupts are allowed from within interrupt routines)..

## Timer/Counter1 Input Capture Register - ICR1H AND ICR1L

| Bit           | 15  | 14 | 13 | 12 | 11 | 10 | 9 | 8 |       |
|---------------|-----|----|----|----|----|----|---|---|-------|
| \$25 (\$45)   | MSB |    |    |    |    |    |   |   | ICR1H |
| \$24 (\$44)   |     |    |    |    |    |    |   |   | ICR1L |
|               | 7   | 6  | 5  | 4  | 3  | 2  | 1 | 0 |       |
| Read/Write    | R   | R  | R  | R  | R  | R  | R | R |       |
|               | R   | R  | R  | R  | R  | R  | R | R |       |
| Initial value | 0   | 0  | 0  | 0  | 0  | 0  | 0 | 0 |       |
|               | 0   | 0  | 0  | 0  | 0  | 0  | 0 | 0 |       |

The input capture register is a 16-bit read-only register.

When the rising or falling edge (according to the input capture edge setting - ICES1) of the signal at the input capture pin - ICP - is detected, the current value of the Timer/Counter1 is transferred to the Input Capture Register - ICR1. At the same time, the input capture flag - ICF1 - is set (one).

Since the Input Capture Register - ICR1 - is a 16-bit register, a temporary register TEMP is used when ICR1 is read to ensure that both bytes are read simultaneously. When the CPU reads the low byte ICR1L, the data is sent to the CPU and the data of the high byte ICR1H is placed in the TEMP register. When the CPU reads the data in the high byte ICR1H, the CPU receives the data in the TEMP register. Consequently, the low byte ICR1L must be accessed first for a full 16-bit register read operation.

The TEMP register is also used when accessing TCNT1, OCR1A and OCR1B. If the main program and also interrupt routines perform access to registers using TEMP, interrupts must be disabled during access from the main program (and from interrupt routines if interrupts are allowed from within interrupt routines).

## Timer/Counter1 In PWM Mode

When the PWM mode is selected, Timer/Counter1 and the Output Compare Register1A - OCR1A and the Output Compare Register1B - OCR1B, form a dual 8, 9 or 10-bit, free-running, glitch-free and phase correct PWM with outputs on the PD5(OC1A) and OC1B pins. Timer/Counter1 acts as an up/down counter, counting up from \$0000 to TOP (see Table 12), where it turns and counts down again to zero before the cycle is repeated. When the counter value matches the contents of the 10 least significant bits of OCR1A or OCR1B, the PD5(OC1A)/OC1B pins are set or cleared according to the settings of the COM1A1/COM1A0 or COM1B1/COM1B0 bits in the Timer/Counter1 Control Register TCCR1A. Refer to Table 13 for details.

**Table 12.** Timer TOP Values and PWM Frequency

| PWM Resolution | Timer TOP value | Frequency       |
|----------------|-----------------|-----------------|
| 8-bit          | \$00FF (255)    | $f_{TCK1}/510$  |
| 9-bit          | \$01FF (511)    | $f_{TCK1}/1022$ |
| 10-bit         | \$03FF(1023)    | $f_{TCK1}/2046$ |

**Table 13.** Compare1 Mode Select in PWM Mode

| COM1X1 | COM1X0 | Effect on OCX1   |
|--------|--------|--|
| 0      | 0      | Not connected  |
| 0      | 1      | Not connected  |
| 1      | 0      | Cleared on compare match, up-counting. Set on compare match, down-counting (non-inverted PWM). |
| 1      | 1      | Cleared on compare match, down-counting. Set on compare match, up-counting (inverted PWM).     |

Note: X = A or B



## EEPROM Data Register - EEDR

|               |     |     |     |     |     |     |     |     |      |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|------|
| Bit           | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |      |
| \$1D (\$3D)   | MSB |     |     |     |     |     |     | LSB | EEDR |
| 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..0 - EEDR7..0: EEPROM Data

For the EEPROM write operation, the EEDR register contains the data to be written to the EEPROM in the address given by the EEAR register. For the EEPROM read operation, the EEDR contains the data read out from the EEPROM at the address given by EEAR.

## EEPROM Control Register - EECR

|               |   |   |   |   |   |       |      |      |      |
|---------------|---|---|---|---|---|-------|------|------|------|
| Bit           | 7 | 6 | 5 | 4 | 3 | 2     | 1    | 0    |      |
| \$1C (\$3C)   | - | - | - | - | - | EEMWE | EEWE | EERE | EECR |
| Read/Write    | R | R | R | R | R | R/W   | R/W  | R/W  |      |
| Initial value | 0 | 0 | 0 | 0 | 0 | 0     | 0    | 0    |      |

### • Bit 7..3 - Res: Reserved bits

These bits are reserved bits in the AT90S4414/8515 and will always read as zero.

### • Bit 2 - EEMWE: EEPROM Master Write Enable

The EEMWE bit determines whether setting EEWE to one causes the EEPROM to be written. When EEMWE is set(one) setting EEWE will write data to the EEPROM at the selected address. If EEMWE is zero, setting EEWE will have no effect. When EEMWE has been set (one) by software, hardware clears the bit to zero after four clock cycles. See the description of the EEWE bit for a EEPROM write procedure.

### • Bit 1 - EEWE: EEPROM Write Enable

The EEPROM Write Enable Signal EEWE is the write strobe to the EEPROM. When address and data are correctly set up, the EEWE bit must be set to write the value into the EEPROM. The EEMWE bit must be set when the logical one is written to EEWE, otherwise no EEPROM write takes place. The following procedure should be followed when writing the EEPROM (the order of steps 2 and 3 is unessential):

1. Wait until EEWE becomes zero.
2. Write new EEPROM address to EEARL and EEARH (optional)
3. Write new EEPROM data to EEDR (optional)
4. Write a logical one to the EEMWE bit in EECR
5. Within four clock cycles after setting EEMWE, write a logical one to EEWE.

When the write access time (typically 2.5 ms at  $V_{CC} = 5V$  or 4 ms at  $V_{CC} = 2.7V$ ) has elapsed, the EEWE bit is cleared (zero) by hardware. The user software can poll this bit and wait for a zero before writing the next byte. When EEWE has been set, the CPU is halted for two cycles before the next instruction is executed.

Caution: An interrupt between step 4 and step 5 will make the write cycle fail, since the EEPROM Master Write Enable will time-out. If an interrupt routine accessing the EEPROM is interrupting another EEPROM access, the EEAR or EEDR register will be modified, causing the interrupted EEPROM access to fail. It is recommended to have the global interrupt flag cleared during the 4 last steps to avoid these problems.

- **Bit 0 - EERE: EEPROM Read Enable**

The EEPROM Read Enable Signal EERE is the read strobe to the EEPROM. When the correct address is set up in the EEAR register, the EERE bit must be set. When the EERE bit is cleared (zero) by hardware, requested data is found in the EEDR register. The EEPROM read access takes one instruction and there is no need to poll the EERE bit. When EERE has been set, the CPU is halted for two cycles before the next instruction is executed.

The user should poll the EERE bit before starting the read operation. If a write operation is in progress when new data or address is written to the EEPROM I/O registers, the write operation will be interrupted, and the result is undefined.

## **Prevent EEPROM Corruption**

During periods of low  $V_{CC}$ , the EEPROM data can be corrupted because the supply voltage is too low for the CPU and the EEPROM to operate properly. These issues are the same as for board level systems using the EEPROM, and the same design solutions should be applied.

An EEPROM data corruption can be caused by two situations when the voltage is too low. First, a regular write sequence to the EEPROM requires a minimum voltage to operate correctly. Secondly, the CPU itself can execute instructions incorrectly, if the supply voltage for executing instructions is too low.

EEPROM data corruption can easily be avoided by following these design recommendations (one is sufficient):

1. Keep the AVR RESET active (low) during periods of insufficient power supply voltage. This is best done by an external low  $V_{CC}$  Reset Protection circuit, often referred to as a Brown-Out Detector (BOD). Please refer to application note AVR 180 for design considerations regarding power-on reset and low voltage detection.
2. Keep the AVR core in Power Down Sleep Mode during periods of low  $V_{CC}$ . This will prevent the CPU from attempting to decode and execute instructions, effectively protecting the EEPROM registers from unintentional writes.
3. Store constants in Flash memory if the ability to change memory contents from software is not required. Flash memory can not be updated by the CPU, and will not be subject to corruption.

## **Serial Peripheral Interface - SPI**

The Serial Peripheral Interface(SPI) allows high-speed synchronous data transfer between the AT90S4414/8515 and peripheral devices or between several AVR devices. The AT90S4414/8515 SPI features include the following:

- Full-duplex, 3-wire Synchronous Data Transfer
- Master or Slave Operation
- LSB First or MSB First Data Transfer
- Four Programmable Bit Rates
- End of Transmission Interrupt Flag
- Write Collision Flag Protection
- Wakeup from Idle Mode (Slave Mode Only)

## UART

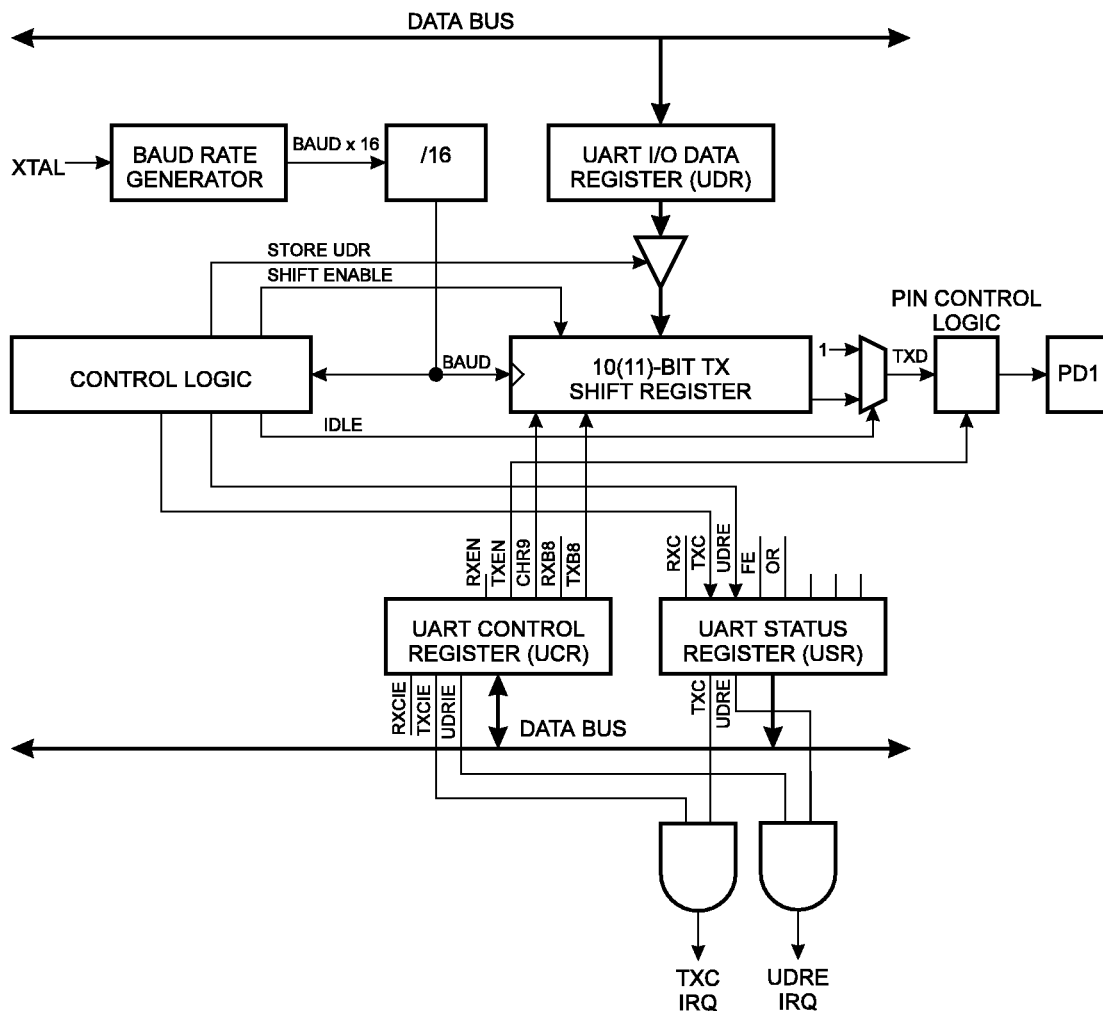
The AT90S4414/8515 features a full duplex (separate receive and transmit registers) Universal Asynchronous Receiver and Transmitter (UART). The main features are:

- Baud rate generator that can generate a large number of baud rates (bps)
- High baud rates at low XTAL frequencies
- 8 or 9 bits data
- Noise filtering
- Overrun detection
- Framing Error detection
- False Start Bit detection
- Three separate interrupts on TX Complete, TX Data Register Empty and RX Complete

### Data Transmission

A block schematic of the UART transmitter is shown in Figure 38.

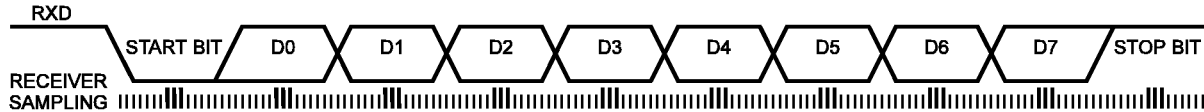
**Figure 38.** UART Transmitter



The receiver front-end logic samples the signal on the RXD pin at a frequency 16 times the baud rate. While the line is idle, one single sample of logical zero 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 RXD pin at samples 8, 9 and 10. If two or more of these three samples are found to be logical ones, 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 8, 9 and 10. 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 40.

**Figure 40.** Sampling Received Data



When the stop bit enters the receiver, the majority of the three samples must be one to accept the stop bit. If two or more samples are logical zeros, the Framing Error (FE) flag in the UART Status Register (USR) is set. Before reading the UDR register, the user should always check the FE bit to detect Framing Errors.

Whether or not a valid stop bit is detected at the end of a character reception cycle, the data is transferred to UDR and the RXC flag in USR is set. UDR is in fact two physically separate registers, one for transmitted data and one for received data. When UDR is read, the Receive Data register is accessed, and when UDR is written, the Transmit Data register is accessed. If 9 bit data word is selected (the CHR9 bit in the UART Control Register, UCR is set), the RXB8 bit in UCR is loaded with bit 9 in the Transmit shift register when data is transferred to UDR.

If, after having received a character, the UDR register has not been read since the last receive, the OverRun (OR) flag in UCR is set. This means that the last data byte shifted into to the shift register could not be transferred to UDR and has been lost. The OR bit is buffered, and is updated when the valid data byte in UDR is read. Thus, the user should always check the OR bit after reading the UDR register in order to detect any overruns if the baud rate is high or CPU load is high.

When the RXEN bit in the UCR register is cleared (zero), the receiver is disabled. This means that the PD0 pin can be used as a general I/O pin. When RXEN is set, the UART Receiver will be connected to PD0, which is forced to be an input pin regardless of the setting of the DDD0 bit in DDRD. When PD0 is forced to input by the UART, the PORTD0 bit can still be used to control the pull-up resistor on the pin.

When the CHR9 bit in the UCR register is set, transmitted and received characters are 9-bit long plus start and stop bits. The 9th data bit to be transmitted is the TXB8 bit in UCR register. This bit must be set to the wanted value before a transmission is initiated by writing to the UDR register. The 9th data bit received is the RXB8 bit in the UCR register.

## UART Control

### UART I/O Data Register - UDR

| Bit           | 7          | 6   | 5   | 4   | 3   | 2   | 1   | 0          |            |
|---------------|------------|-----|-----|-----|-----|-----|-----|------------|------------|
| \$0C (\$2C)   | <b>MSB</b> |     |     |     |     |     |     | <b>LSB</b> | <b>UDR</b> |
| 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 UDR register is actually two physically separate registers sharing the same I/O address. When writing to the register, the UART Transmit Data register is written. When reading from UDR, the UART Receive Data register is read.

• **Bit 2 - ACIC: Analog Comparator Input Capture Enable**

When set (one), this bit enables the Input Capture function in Timer/Counter1 to be triggered by the analog comparator. The comparator output is in this case directly connected to the Input Capture front-end logic, making the comparator utilize the noise canceler and edge select features of the Timer/Counter1 Input Capture interrupt. When cleared (zero), no connection between the analog comparator and the Input Capture function is given. To make the comparator trigger the Timer/Counter1 Input Capture interrupt, the TICIE1 bit in the Timer Interrupt Mask Register (TIMSK) must be set (one).

• **Bits 1,0 - ACIS1, ACIS0: Analog Comparator Interrupt Mode Select**

These bits determine which comparator events that trigger the Analog Comparator interrupt. The different settings are shown in Table 19.

**Table 19.** ACIS1/ACIS0 Settings

| ACIS1 | ACIS0 | Interrupt Mode                              |
|-------|-------|---|
| 0     | 0     | Comparator Interrupt on Output Toggle       |
| 0     | 1     | Reserved                                    |
| 1     | 0     | Comparator Interrupt on Falling Output Edge |
| 1     | 1     | Comparator Interrupt on Rising Output Edge  |

Note: When changing the ACIS1/ACIS0 bits, The Analog Comparator Interrupt must be disabled by clearing its Interrupt Enable bit in the ACSR register. Otherwise an interrupt can occur when the bits are changed.

## Interface to External SRAM

The interface to the SRAM 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 data SRAM is enabled by setting the SRE - External SRAM enable bit of the MCUCR - MCU control register, and will override the setting of the data direction register DDRA. When the SRE bit is cleared (zero), the external data SRAM is disabled, and the normal pin and data direction settings are used. When SRE is cleared (zero), the address space above the internal SRAM boundary is not mapped into the internal SRAM, as in AVR parts not having interface to the external SRAM.

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 SRAM only.

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

Figure 42 sketches how to connect an external SRAM to the AVR using 8 latches which are transparent when G is high.

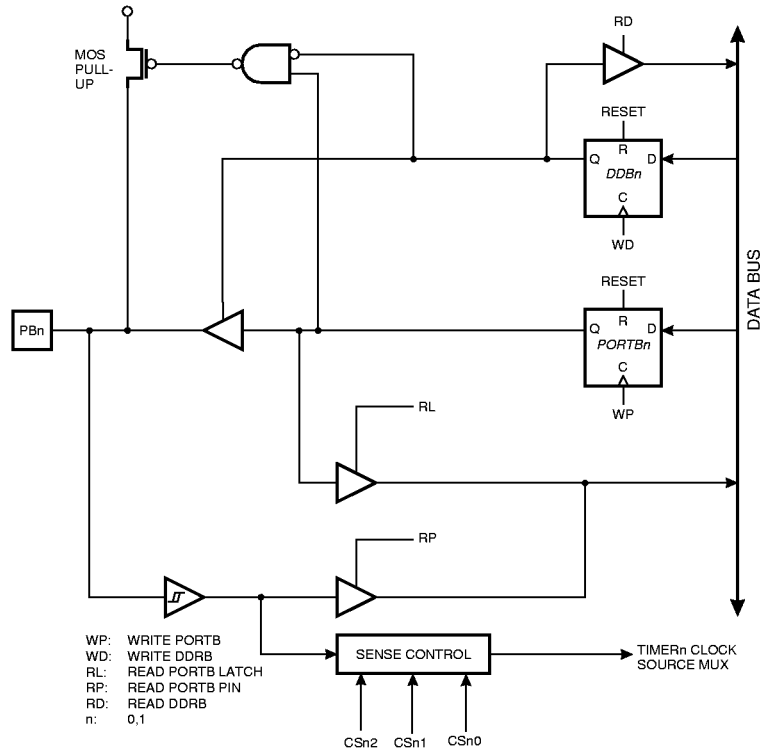
Default, the external SRAM access is a three-cycle scheme as depicted in Figure 43. When one extra wait state is needed in the access cycle, set the SRW bit (one) in the MCUCR register. The resulting access scheme is shown in Figure 44. In both cases, note that PORTA is data bus in one cycle only. As soon as the data access finishes, PORTA becomes a low order address bus again.

For details in the timing for the SRAM interface, please refer to Figure 68, Table 38, Table 39, Table 40, and Table 41 in section "Absolute Maximum Ratings\*" on page 81.

## Port B Schematics

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

**Figure 46.** Port B Schematic Diagram (Pins PB0 and PB1)



**Figure 47.** Port B Schematic Diagram (Pins PB2 and PB3)

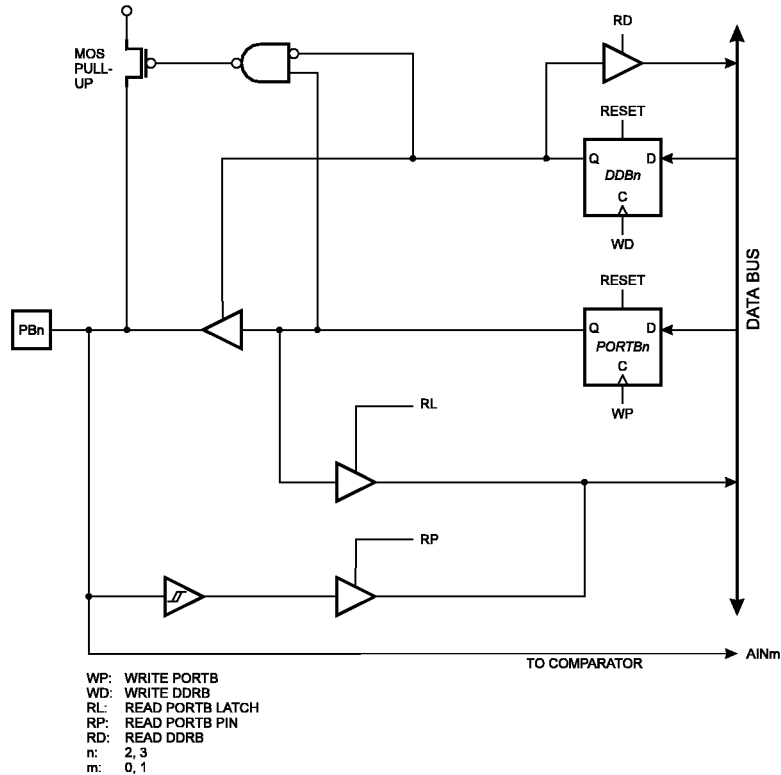


Figure 50. Port B Schematic Diagram (Pin PB6)

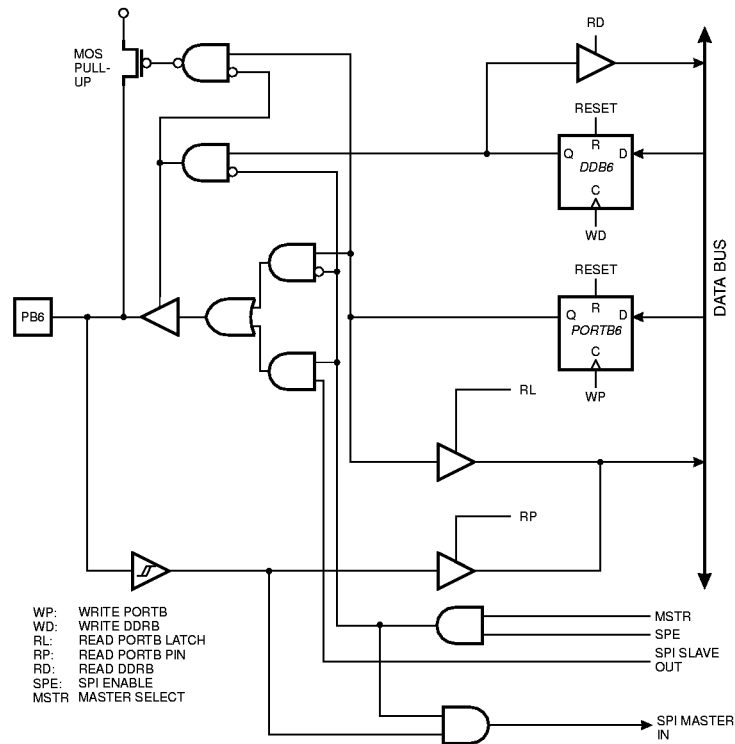
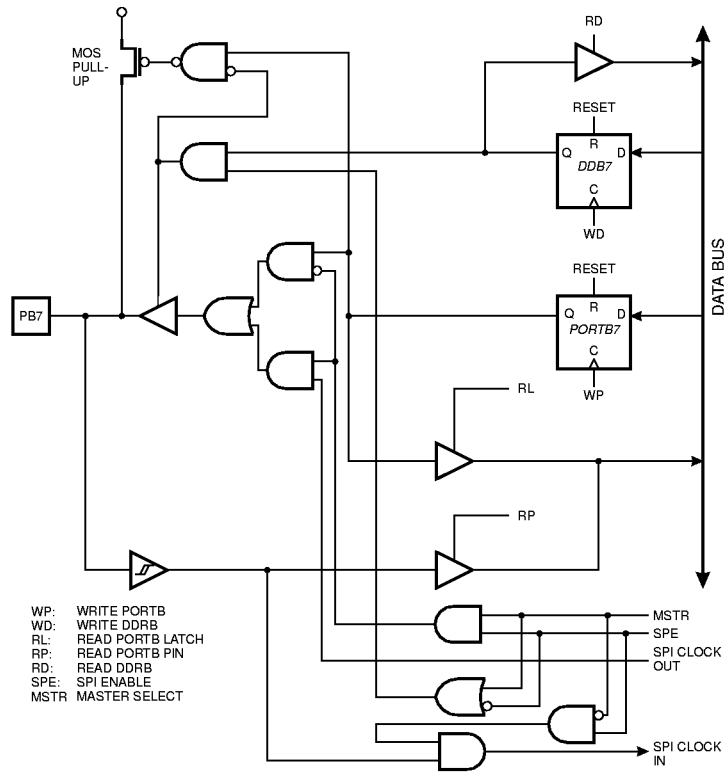


Figure 51. Port B Schematic Diagram (Pin PB7)



The Program and Data memory arrays on the AT90S4414/8515 are programmed byte-by-byte in either programming modes. For the EEPROM, an auto-erase cycle is provided within the self-timed write operation in the serial programming mode. During programming, the supply voltage must be in accordance with Table 27.

**Table 27.** Supply Voltage During Programming

| Part      | Serial Programming | Parallel Programming |
|-----------|--------------------|----------------------|
| AT90S4414 | 2.7 - 6.0V         | 4.5 - 5.5V           |
| AT90S8515 | 2.7 - 6.0V         | 4.5 - 5.5V           |

## Parallel Programming

This section describes how to parallel program and verify Flash Program memory, EEPROM Data memory, Lock bits and Fuse bits in the AT90S4414/8515.

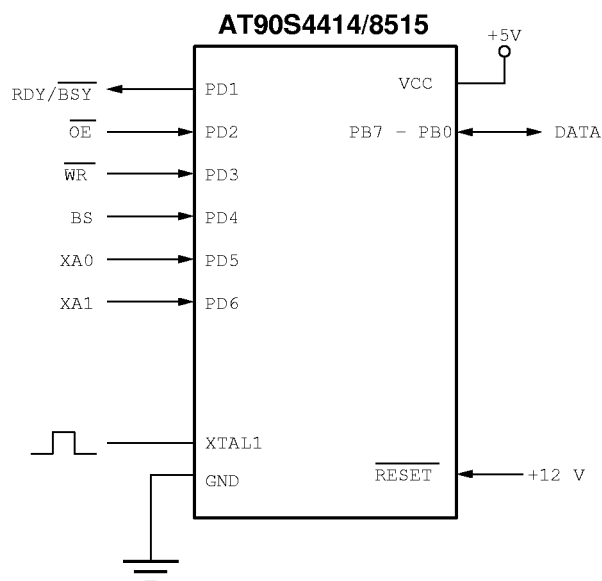
### Signal Names

In this section, some pins of the AT90S4414/AT908515 are referenced by signal names describing their function during parallel programming. See Figure 60 and Table 28. Pins not described in Table 28 are referenced by pin names.

The XA1/XA0 pins determine the action executed when the XTAL1 pin is given a positive pulse. The bit coding are shown in Table 29.

When pulsing  $\overline{WR}$  or  $\overline{OE}$ , the command loaded determines the action executed. The Command is a byte where the different bits are assigned functions as shown in Table 30.

**Figure 60.** Parallel Programming





## Chip Erase

The Chip Erase command will erase the Flash and EEPROM memories, and the Lock bits. The Lock bits are not reset until the Flash and EEPROM have been completely erased. The Fuse bits are not changed. Chip Erase must be performed before the Flash or EEPROM is reprogrammed.

Load Command "Chip Erase"

1. Set XA1, XA0 to "10". This enables command loading.
2. Set BS to "0".
3. Set DATA to '1000 0000'. This is the command for Chip erase.
4. Give XTAL1 a positive pulse. This loads the command.
5. Give  $\overline{WR}$  a  $t_{WLWH\_CE}$  wide negative pulse to execute Chip Erase. See Table 31 for  $t_{WLWH\_CE}$  value. Chip Erase does not generate any activity on the RDY/ $\overline{BSY}$  pin.

## Programming the Flash

A: Load Command "Write Flash"

1. Set XA1, XA0 to '10'. This enables command loading.
2. Set BS to '0'
3. Set DATA to '0001 0000'. This is the command for Write Flash.
4. Give XTAL1 a positive pulse. This loads the command.

B: Load Address High Byte

1. Set XA1, XA0 to "00". This enables address loading.
2. Set BS to "1". This selects high byte.
3. Set DATA = Address high byte (\$00 - \$07/\$0F)
4. Give XTAL1 a positive pulse. This loads the address high byte.

C: Load Address Low Byte

1. Set XA1, XA0 to "00". This enables address loading.
2. Set BS to "0". This selects low byte.
3. Set DATA = Address low byte (\$00 - \$FF)
4. Give XTAL1 a positive pulse. This loads the address low byte.

D: Load Data Low Byte

1. Set XA1, XA0 to "01". This enables data loading.
2. Set DATA = Data low byte (\$00 - \$FF)
3. Give XTAL1 a positive pulse. This loads the data low byte.

E: Write Data Low Byte

1. Set BS to "0". This selects low data.
2. Give  $\overline{WR}$  a negative pulse. This starts programming of the data byte. RDY/ $\overline{BSY}$  goes low.
3. Wait until RDY/ $\overline{BSY}$  goes high to program the next byte.

(See Figure 61 for signal waveforms.)

F: Load Data High Byte

1. Set XA1, XA0 to "01". This enables data loading.
2. Set DATA = Data high byte (\$00 - \$FF)
3. Give XTAL1 a positive pulse. This loads the data high byte.

G: Write Data High Byte

1. Set BS to "1". This selects high data.
2. Give  $\overline{WR}$  a negative pulse. This starts programming of the data byte. RDY/ $\overline{BSY}$  goes low.

3. Wait until RDY/BSY goes high to program the next byte.

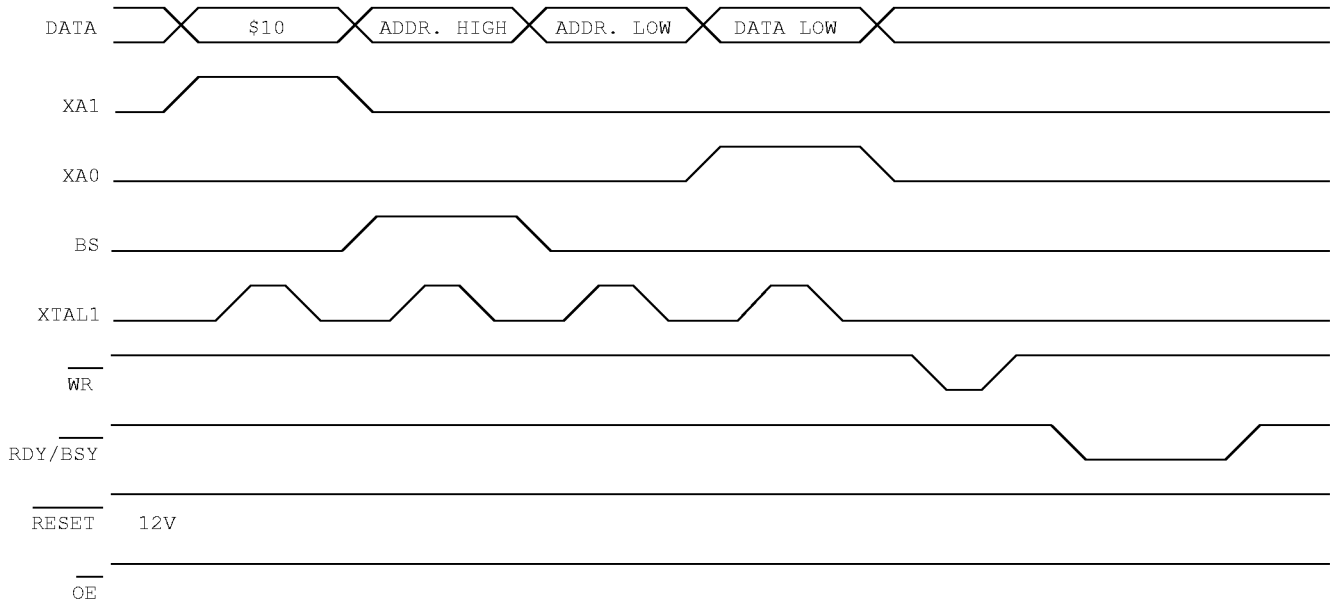
(See Figure 62 for signal waveforms.)

The loaded command and address are retained in the device during programming. For efficient programming, the following should be considered.

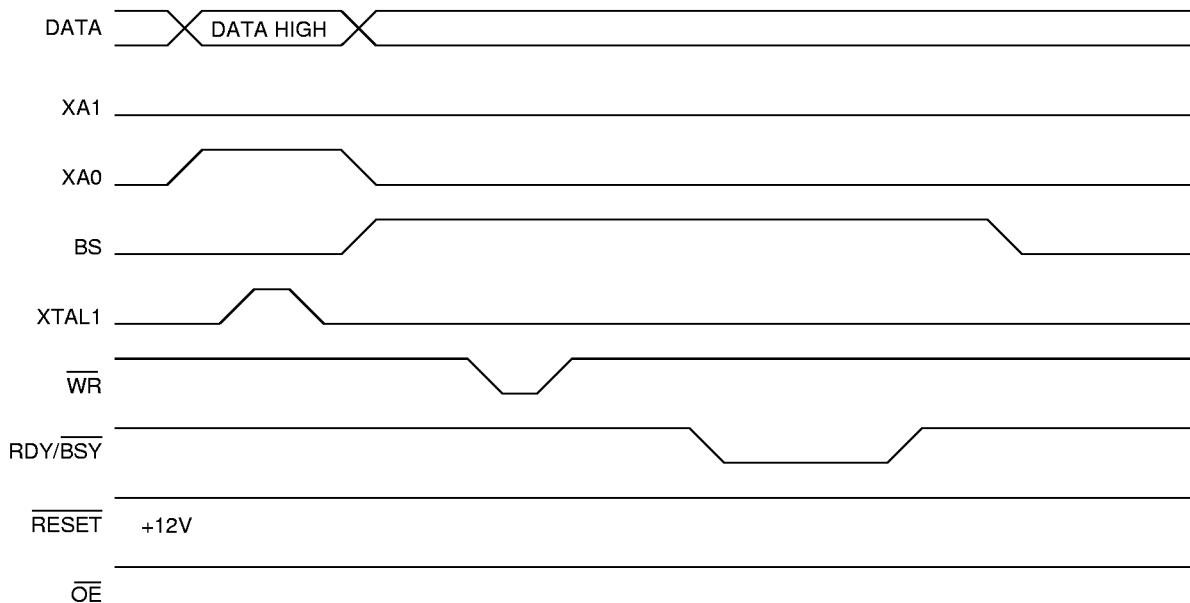
- The command needs only be loaded once when writing or reading multiple memory locations.
- Address high byte needs only be loaded before programming a new 256 word page in the Flash.
- Skip writing the data value \$FF, that is the contents of the entire Flash and EEPROM after a Chip Erase.

These considerations also applies to EEPROM programming, and Flash, EEPROM and Signature bytes reading.

**Figure 61. Programming the Flash Waveforms**



**Figure 62. Programming the Flash Waveforms (continued)**



**Table 40.** External Data Memory Characteristics, 2.7 - 4.0 Volts, No Wait State

|    | Symbol         | Parameter   | 4 MHz Oscillator |       | Variable Oscillator      |                          | Unit |
|----|----------------|---|------------------|-------|--------------------------|--------------------------|------|
|    |                |   | Min              | Max   | Min                      | Max                      |      |
| 0  | $1/t_{CLCL}$   | Oscillator Frequency                                |                  |       | 0.0                      | 4.0                      | MHz  |
| 1  | $t_{LHLL}$     | ALE Pulse Width                                     | 70.0             |       | $0.5t_{CLCL}-55.0^{(1)}$ |                          | ns   |
| 2  | $t_{AVLL}$     | Address Valid A to ALE Low                          | 60.0             |       | $0.5t_{CLCL}-65.0^{(1)}$ |                          | ns   |
| 3a | $t_{LLAX\_ST}$ | Address Hold After ALE Low, ST/STD/STS Instructions | 130.0            |       | $0.5t_{CLCL}+5.0^{(2)}$  |                          | ns   |
| 3b | $t_{LLAX\_LD}$ | Address Hold after ALE Low, LD/LDD/LDS Instructions | 15.0             |       | 15.0                     |                          | ns   |
| 4  | $t_{AVLLC}$    | Address Valid C to ALE Low                          | 60.0             |       | $0.5t_{CLCL}-65.0^{(1)}$ |                          | ns   |
| 5  | $t_{AVRL}$     | Address Valid to RD Low                             | 200.0            |       | $1.0t_{CLCL}-50.0$       |                          | ns   |
| 6  | $t_{AVWL}$     | Address Valid to WR Low                             | 325.0            |       | $1.5t_{CLCL}-50.0^{(1)}$ |                          | ns   |
| 7  | $t_{LLWL}$     | ALE Low to WR Low                                   | 230.0            | 270.0 | $1.0t_{CLCL}-20.0$       | $1.0t_{CLCL}+20.0$       | ns   |
| 8  | $t_{LLRL}$     | ALE Low to RD Low                                   | 105.0            | 145.0 | $0.5t_{CLCL}-20.0^{(2)}$ | $0.5t_{CLCL}+20.0^{(2)}$ | ns   |
| 9  | $t_{DVRH}$     | Data Setup to RD High                               | 95.0             |       | 95.0                     |                          | ns   |
| 10 | $t_{RLDV}$     | Read Low to Data Valid                              |                  | 170.0 |                          | $1.0t_{CLCL}-80.0$       | ns   |
| 11 | $t_{RHDX}$     | Data Hold After RD High                             | 0.0              |       | 0.0                      |                          | ns   |
| 12 | $t_{RLRH}$     | RD Pulse Width                                      | 230.0            |       | $1.0t_{CLCL}-20.0$       |                          | ns   |
| 13 | $t_{DVWL}$     | Data Setup to WR Low                                | 70.0             |       | $0.5t_{CLCL}-55.0^{(1)}$ |                          | ns   |
| 14 | $t_{WHDX}$     | Data Hold After WR High                             | 0.0              |       | 0.0                      |                          | ns   |
| 15 | $t_{DVWH}$     | Data Valid to WR High                               | 210.0            |       | $1.0t_{CLCL}-40.0$       |                          | ns   |
| 16 | $t_{WLWH}$     | WR Pulse Width                                      | 105.0            |       | $0.5t_{CLCL}-20.0^{(2)}$ |                          | ns   |

**Table 41.** External Data Memory Characteristics, 2.7 - 4.0 Volts, 1 Cycle Wait State

|    | Symbol       | Parameter              | 4 MHz Oscillator |        | Variable Oscillator      |                    | Unit |
|----|--------------|------------------------|------------------|--------|--------------------------|--------------------|------|
|    |              |                        | Min              | Max    | Min                      | Max                |      |
| 0  | $1/t_{CLCL}$ | Oscillator Frequency   |                  |        | 0.0                      | 4.0                | MHz  |
| 10 | $t_{RLDV}$   | Read Low to Data Valid |                  | 420.00 |                          | $2.0t_{CLCL}-80.0$ | ns   |
| 12 | $t_{RLRH}$   | RD Pulse Width         | 480.0            |        | $2.0t_{CLCL}-20.0$       |                    | ns   |
| 15 | $t_{DVWH}$   | Data Valid to WR High  | 460.0            |        | $2.0t_{CLCL}-40.0$       |                    | ns   |
| 16 | $t_{WLWH}$   | WR Pulse Width         | 355.0            |        | $1.5t_{CLCL}-20.0^{(2)}$ |                    | ns   |

Notes: 1. This assumes 50% clock duty cycle. The half period is actually the high time of the external clock, XTAL1.  
2. This assumes 50% clock duty cycle. The half period is actually the low time of the external clock, XTAL1.

## Typical Characteristics

The following charts show typical behavior. These data are characterized, but not tested. All current consumption measurements are performed with all I/O pins configured as inputs and with internal pull-ups enabled. ICP pulled high externally. A sine wave generator with rail to rail output is used as clock source.

The power consumption in power-down mode is independent of clock selection.

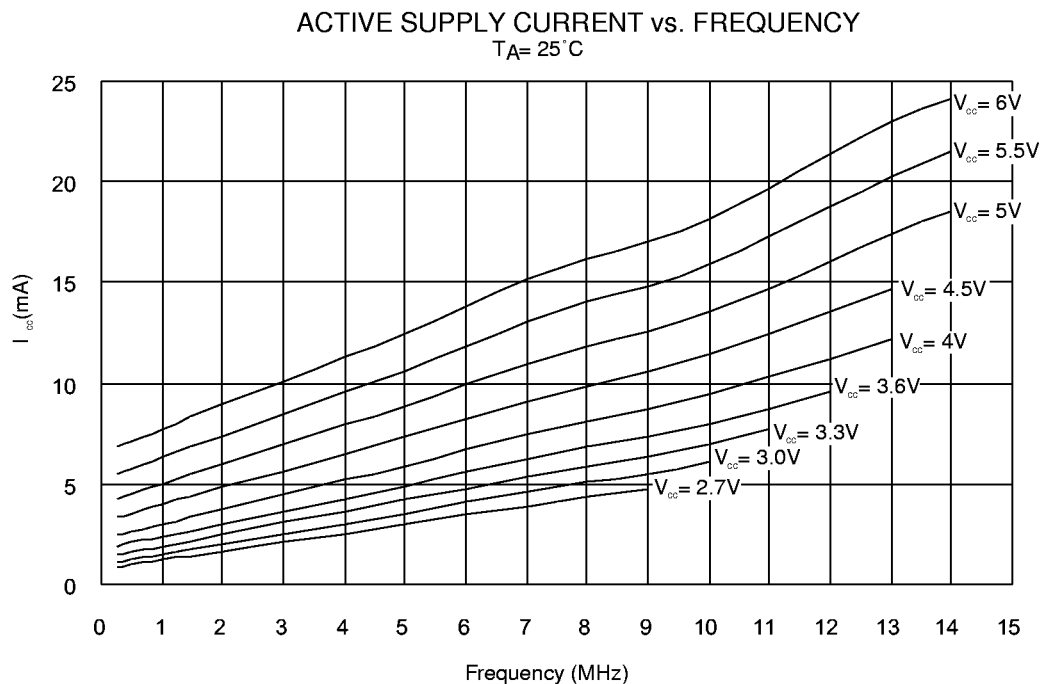
The current consumption is a function of several factors such as: operating voltage, operating frequency, loading of I/O pins, switching rate of I/O pins, code executed and ambient temperature. The dominating factors are operating voltage and frequency.

The current drawn from capacitive loaded pins may be estimated (for one pin) as  $C_L \cdot V_{CC} \cdot f$  where  $C_L$  = load capacitance,  $V_{CC}$  = operating voltage and  $f$  = average switching frequency of I/O pin.

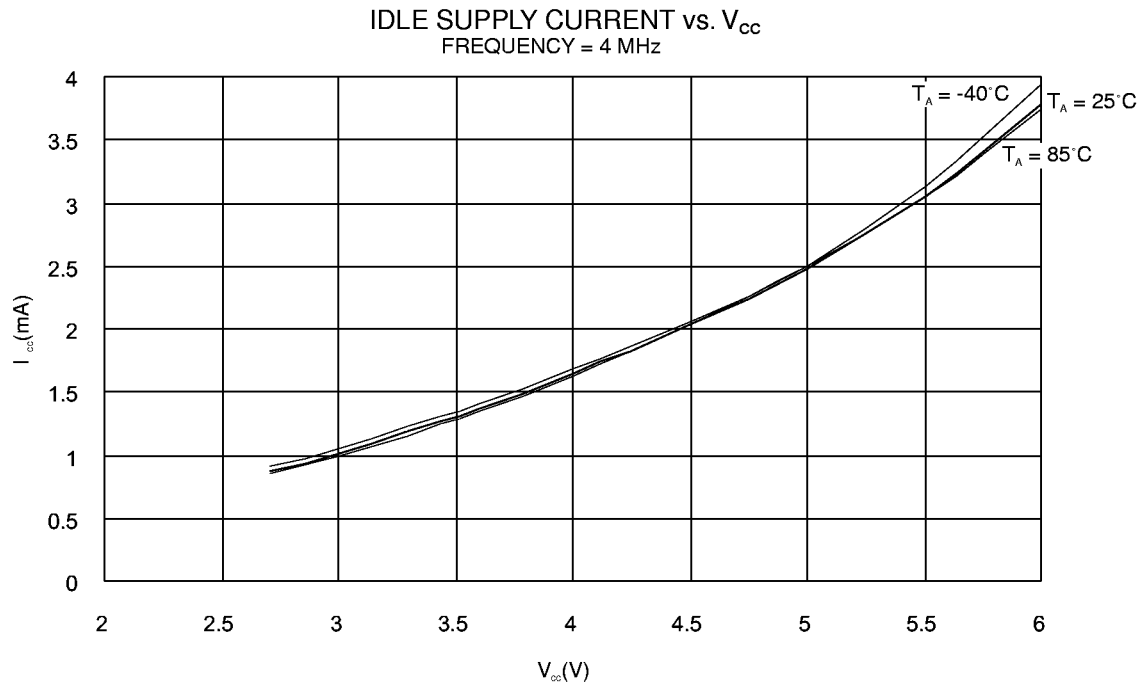
The parts are characterized at frequencies higher than test limits. Parts are not guaranteed to function properly at frequencies higher than the ordering code indicates.

The difference between current consumption in Power Down mode with Watchdog timer enabled and Power Down mode with Watchdog timer disabled represents the differential current drawn by the watchdog timer

**Figure 69.** Active Supply Current vs. Frequency



**Figure 72.** Idle Supply current vs.  $V_{CC}$



**Figure 73.** Power Down Supply Current vs.  $V_{CC}$

