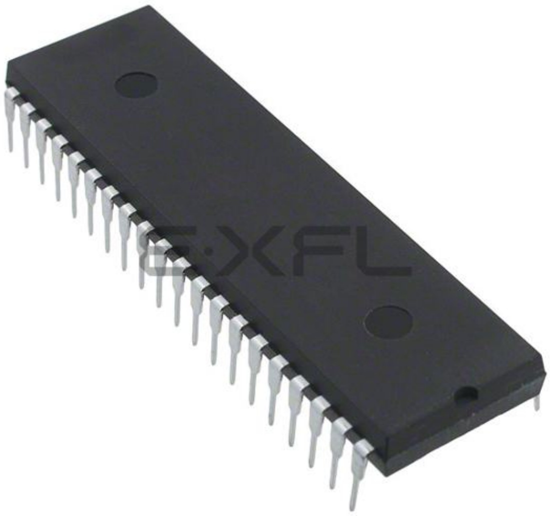


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

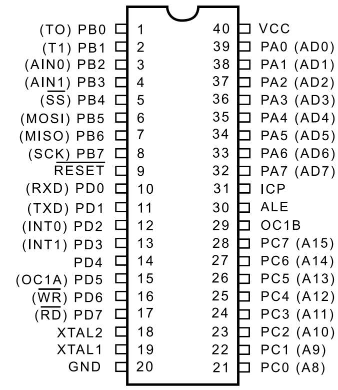
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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"

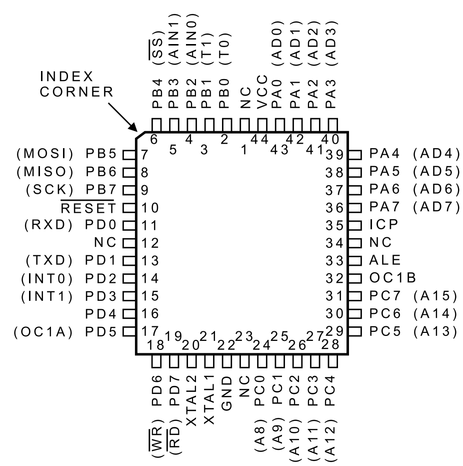
Details	
Product Status	Obsolete
Core Processor	AVR
Core Size	8-Bit
Speed	4MHz
Connectivity	SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	32
Program Memory Size	8KB (4K x 16)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C
Mounting Type	Through Hole
Package / Case	40-DIP (0.600", 15.24mm)
Supplier Device Package	40-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at90s8515a-4pi

Pin Configurations

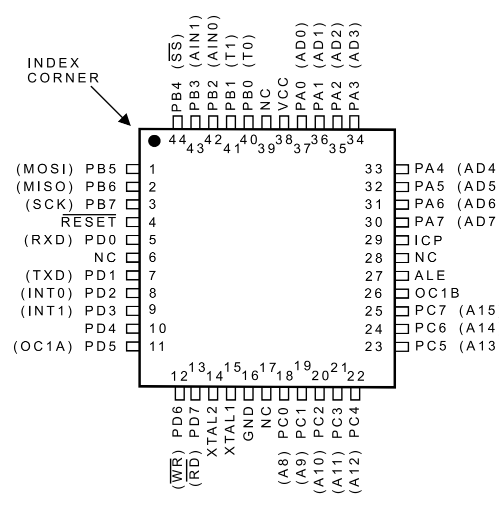
PDIP



PLCC



TQFP



Architectural Overview

The fast-access register file concept contains 32 x 8-bit general-purpose working registers with a single clock cycle access time. This means that during one single clock cycle, one ALU (Arithmetic Logic Unit) operation is executed. Two operands are output from the register file, the operation is executed and the result is stored back in the register file – in one clock cycle.

Six of the 32 registers can be used as three 16-bit indirect address register pointers for Data Space addressing, enabling efficient address calculations. One of the three address pointers is also used as the address pointer for the constant table look-up function. These added function registers are the 16-bit X-, Y-, and Z-register.

The ALU supports arithmetic and logic functions between registers or between a constant and a register. Single register operations are also executed in the ALU. Figure 4 shows the AT90S8515 AVR RISC microcontroller architecture.

In addition to the register operation, the conventional memory addressing modes can be used on the register file as well. This is enabled by the fact that the register file is assigned the 32 lowermost Data Space addresses ($00 - $1F), allowing them to be accessed as though they were ordinary memory locations.

The I/O memory space contains 64 addresses for CPU peripheral functions such as Control Registers, Timer/Counters, A/D converters and other I/O functions. The I/O memory can be accessed directly or as the Data Space locations following those of the register file, $20 - $5F.

The AVR uses a Harvard architecture concept – with separate memories and buses for program and data. The program memory is executed with a two-stage pipeline. While one instruction is being executed, the next instruction is pre-fetched from the program memory. This concept enables instructions to be executed in every clock cycle. The program memory is In-System Programmable Flash memory.

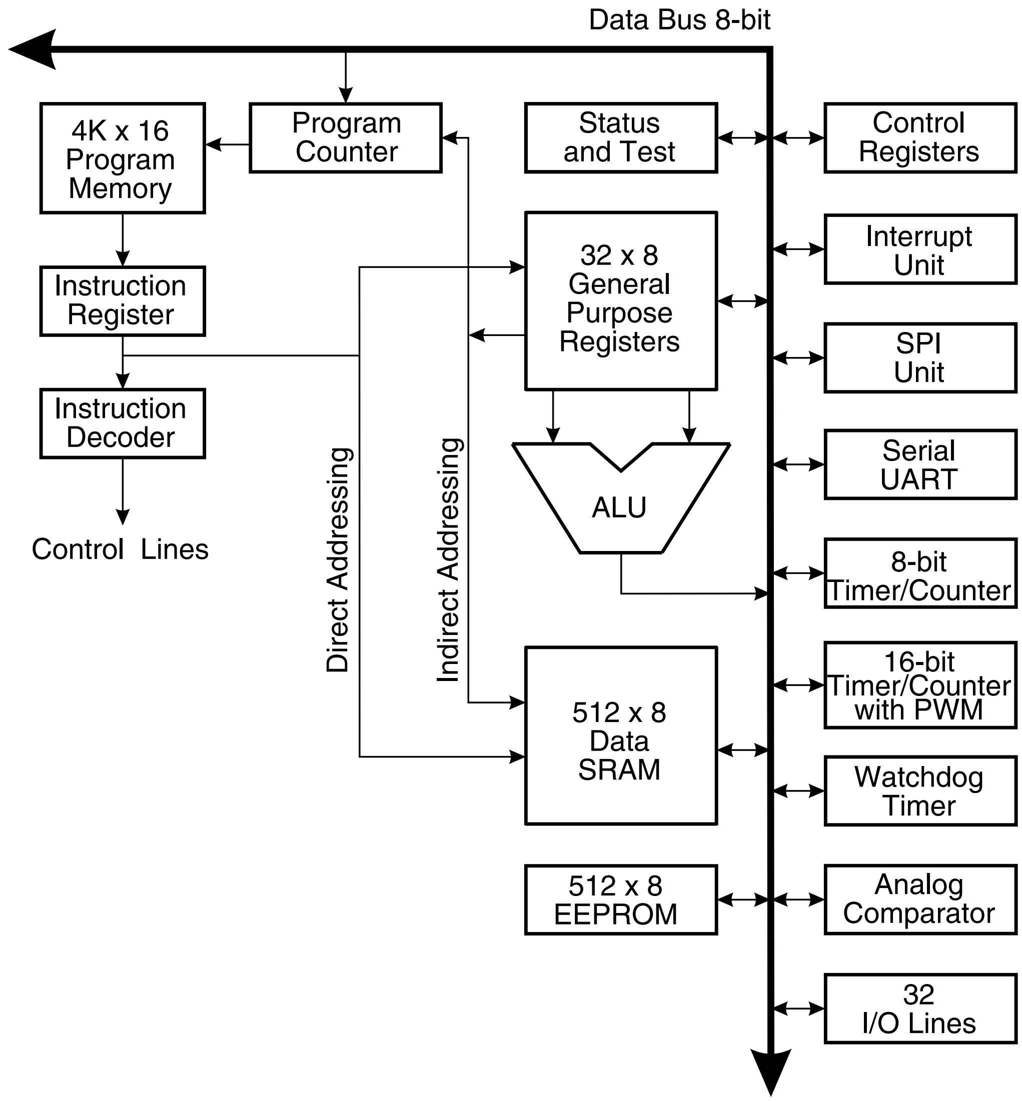
With the relative jump and call instructions, the whole 4K address space is directly accessed. Most AVR instructions have a single 16-bit word format. Every program memory address contains a 16- or 32-bit instruction.

During interrupts and subroutine calls, the return address Program Counter (PC) is stored on the stack. The stack is effectively allocated in the general data SRAM and consequently, the stack size is only limited by the total SRAM size and the usage of the SRAM. All user programs must initialize the SP in the reset routine (before subroutines or interrupts are executed). The 16-bit Stack Pointer (SP) is read/write-accessible in the I/O space.

The 512-byte data SRAM can be easily accessed through the five different addressing modes supported in the AVR architecture.

The memory spaces in the AVR architecture are all linear and regular memory maps.

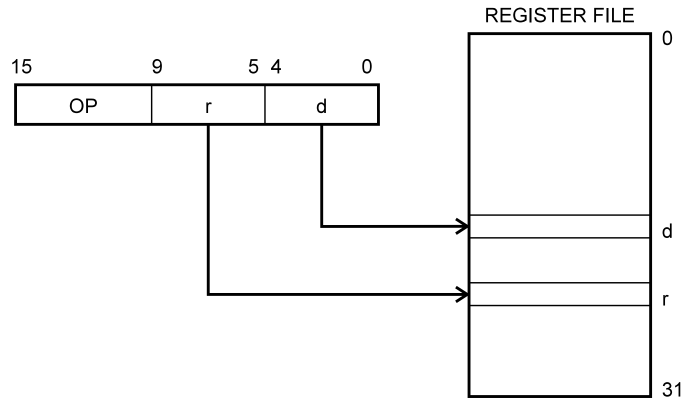
Figure 4. The AT90S8515 AVR RISC Architecture



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.

Register Direct, Two Registers Rd and Rr

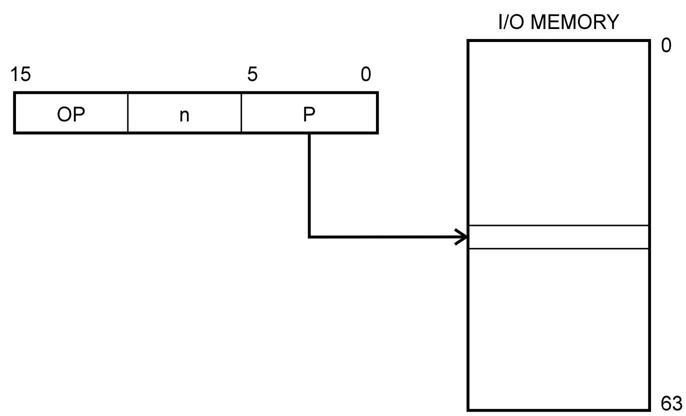
Figure 10. Direct Register Addressing, Two Registers



Operands are contained in register r (Rr) and d (Rd). The result is stored in register d (Rd).

I/O Direct

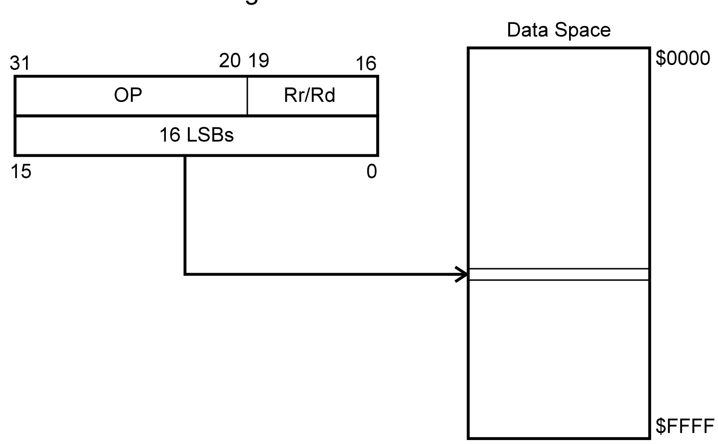
Figure 11. I/O Direct Addressing



Operand address is contained in six bits of the instruction word. n is the destination or source register address.

Data Direct

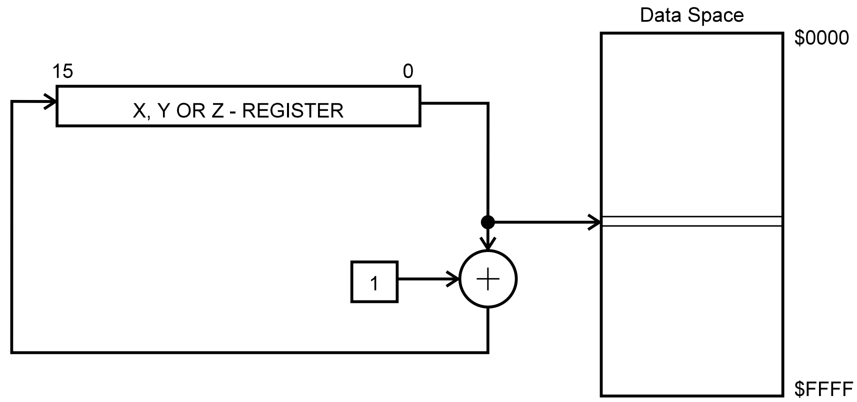
Figure 12. Direct Data Addressing



The X-, Y-, or the Z-register is decremented before the operation. Operand address is the decremented contents of the X-, Y-, or the Z-register.

Data Indirect with Post-increment

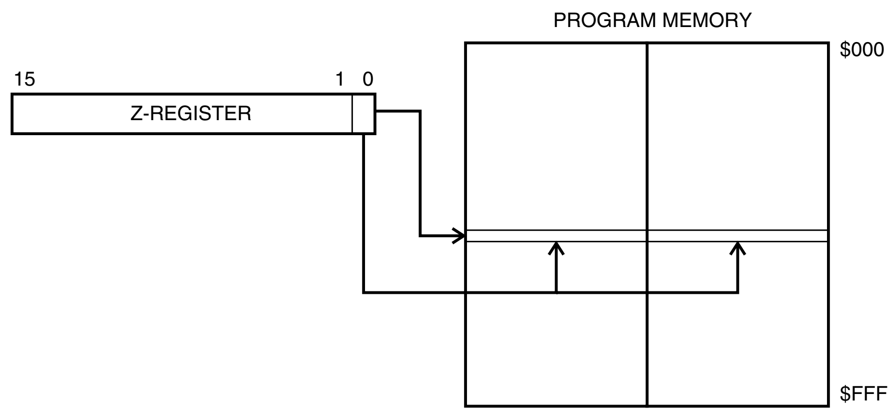
Figure 16. Data Indirect Addressing with Post-increment



The X-, Y-, or the Z-register is incremented after the operation. Operand address is the content of the X-, Y-, or the Z-register prior to incrementing.

Constant Addressing Using the LPM Instruction

Figure 17. Code Memory Constant Addressing

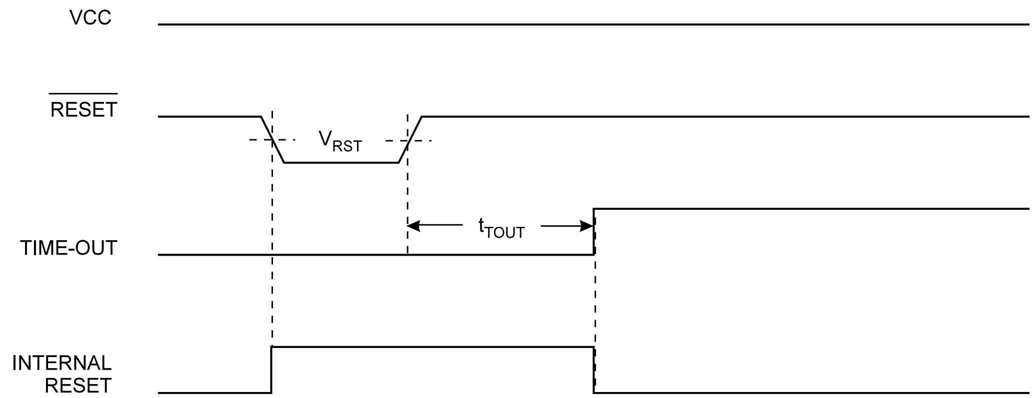


Constant byte address is specified by the Z-register contents. The 15 MSBs select word address (0 - 4K), the LSB selects low byte if cleared (LSB = 0) or high byte if set (LSB = 1).

External Reset

An external reset is generated by a low level on the $\overline{\text{RESET}}$ pin. Reset pulses longer than 50 ns will generate a reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a reset. When the applied signal reaches the Reset Threshold Voltage (V_{RST}) on its positive edge, the delay timer starts the MCU after the Time-out period t_{TOUT} has expired.

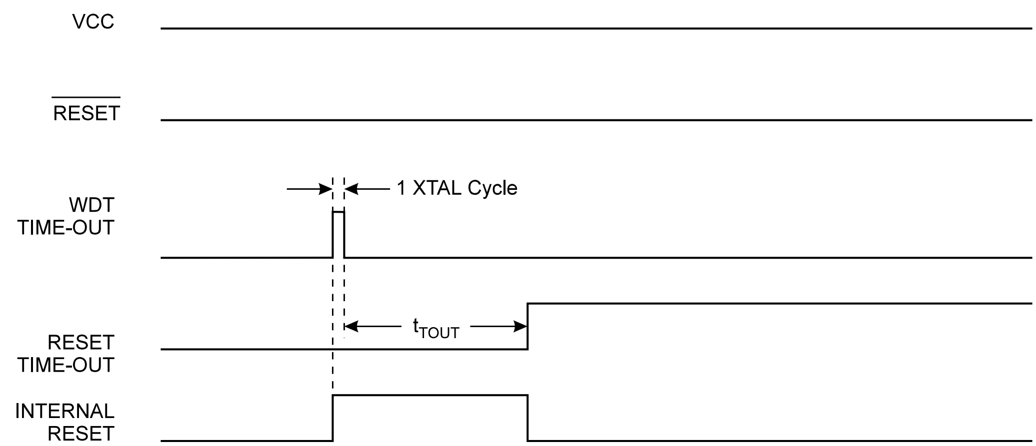
Figure 26. External Reset during Operation



Watchdog Reset

When the Watchdog times out, it will generate a short reset pulse of 1 XTAL cycle duration. On the falling edge of this pulse, the delay timer starts counting the Time-out period t_{TOUT}. Refer to page 42 for details on operation of the Watchdog.

Figure 27. Watchdog Reset during Operation



Interrupt Handling

The AT90S8515 has two 8-bit interrupt mask control registers; GIMSK (General Interrupt Mask register) and TIMSK (Timer/Counter Interrupt Mask register).

When an interrupt occurs, the Global Interrupt Enable I-bit is cleared (zero) and all interrupts are disabled. The user software can set (one) the I-bit to enable nested interrupts. The I-bit is set (one) when a Return from Interrupt instruction (RETI) is executed.

For interrupts triggered by events that can remain static (e.g., the Output Compare Register1 matching the value of Timer/Counter1), the interrupt flag is set when the event occurs. If the interrupt flag is cleared and the interrupt condition persists, the flag will not be set until the event occurs the next time.

When the Program Counter is vectored to the actual interrupt vector in order to execute the interrupt handling routine, hardware clears the corresponding flag that generated the

• **Bit 6 – INTF0: External Interrupt Flag0**

When an edge on the INT0 pin triggers an interrupt request, the corresponding interrupt flag, INTF0, becomes set (one). If the I-bit in SREG and the corresponding interrupt enable bit, INT0 in GIMSK are set (one), the MCU will jump to the interrupt vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag is cleared by writing a logical “1” to it. This flag is always cleared when INT0 is configured as level interrupt.

• **Bits 5..0 – Res: Reserved Bits**

These bits are reserved bits in the AT90S8515 and always read as zero.

Timer/Counter Interrupt Mask Register – TIMSK

Bit	7	6	5	4	3	2	1	0	
$39 ($59)	TOIE1	OCIE1A	OCIE1B	–	TICIE1	–	TOIE0	–	TIMSK
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 – OCE1A: 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 AT90S8515 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 AT90S8515 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 AT90S8515 and always reads zero.

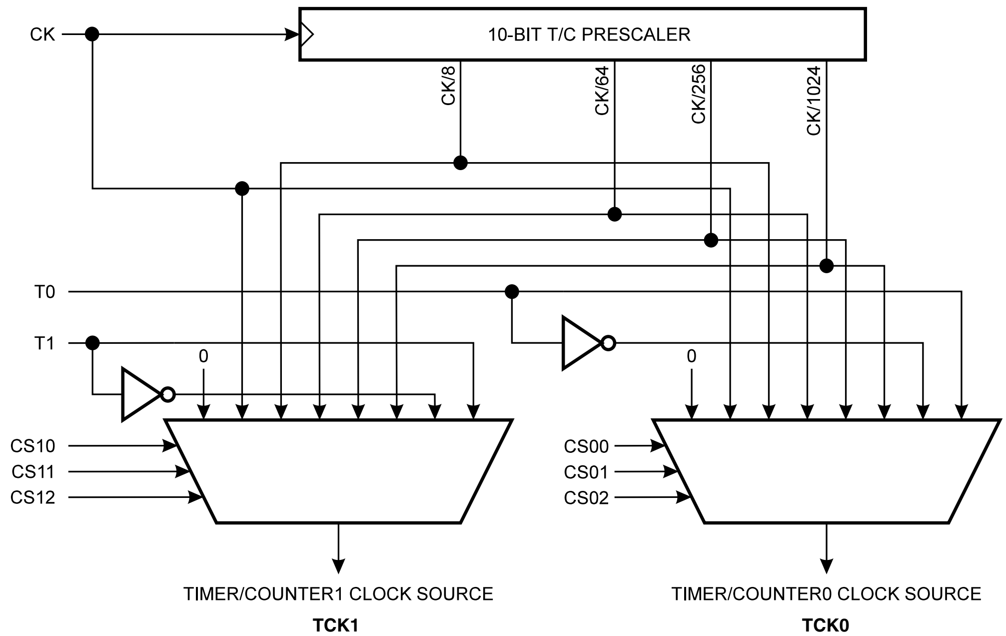
Timer/Counters

The AT90S8515 provides two general-purpose Timer/Counters – one 8-bit T/C and one 16-bit T/C. The Timer/Counters have individual prescaling selection from the same 10-bit prescaling timer. Both Timer/Counters can either be used as a timer with an internal clock time base or as a counter with an external pin connection that triggers the counting.

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 such 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 Insterrupt 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.

Timer/Counter1 Control Register B – TCCR1B

Bit	7	6	5	4	3	2	1	0	
$2E ($4E)	ICNC1	ICES1	–	–	CTC1	CS12	CS11	CS10	TCCR1B
Read/Write	R/W	R/W	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – ICNC1: Input Capture1 Noise Canceler (4 CKs)**

When the ICNC1 bit is cleared (zero), the input capture trigger noise canceler function is disabled. The input capture is triggered at the first rising/falling edge sampled on the ICP (input capture pin) as specified. When the ICNC1 bit is set (one), four successive samples are measured on the ICP, and all samples must be high/low according to the input capture trigger specification in the ICES1 bit. The actual sampling frequency is XTAL clock frequency.

- **Bit 6 – ICES1: Input Capture1 Edge Select**

While the ICES1 bit is cleared (zero), the Timer/Counter1 contents are transferred to the Input Capture Register (ICR1) on the falling edge of the input capture pin (ICP). While the ICES1 bit is set (one), the Timer/Counter1 contents are transferred to the ICR1 on the rising edge of the ICP.

- **Bits 5, 4 – Res: Reserved Bits**

These bits are reserved bits in the AT90S8515 and always read zero.

- **Bit 3 – CTC1: Clear Timer/Counter1 on Compare Match**

When the CTC1 control bit is set (one), the Timer/Counter1 is reset to $0000 in the clock cycle after a compareA match. If the CTC1 control bit is cleared, Timer/Counter1 continues counting and is unaffected by a compare match. Since the compare match is detected in the CPU clock cycle following the match, this function will behave differently when a prescaling higher than 1 is used for the timer. When a prescaling of 1 is used, and the compareA register is set to C, the timer will count as follows if CTC1 is set:

... | C-2 | C-1 | C | 0 | 1 | ...

When the prescaler is set to divide by 8, the timer will count like this:

... | C-2, C-2, C-2, C-2, C-2, C-2, C-2, C-2 | C-1, C-1, C-1, C-1, C-1, C-1, C-1, C-1 | C, 0, 0, 0, 0, 0, 0, 0 | ...

In PWM mode, this bit has no effect.

- **Bits 2, 1, 0 – CS12, CS11, CS10: Clock Select1, Bits 2, 1 and 0**

The Clock Select1 bits 2, 1 and 0 define the prescaling source of Timer/Counter1.

Table 10. Clock 1 Prescale Select

CS12	CS11	CS10	Description
0	0	0	Stop, the Timer/Counter1 is stopped.
0	0	1	CK
0	1	0	CK/8
0	1	1	CK/64
1	0	0	CK/256
1	0	1	CK/1024
1	1	0	External Pin T1, falling edge
1	1	1	External Pin T1, rising edge

The TEMP register is also used when accessing TCNT1, OCR1A and OCR1B. If the main program and 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, 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 11), 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 12 for details.

Table 11. 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 12. 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

Note that in the PWM mode, the 10 least significant OCR1A/OCR1B bits, when written, are transferred to a temporary location. They are latched when Timer/Counter1 reaches the value TOP. This prevents the occurrence of odd-length PWM pulses (glitches) in the event of an unsynchronized OCR1A/OCR1B write. See Figure 32 for an example.

- **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 "1" 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 "1" to the EEMWE bit in EECR (to be able to write a logical "1" to the EEMWE bit, the EEWE bit must be written to zero in the same cycle).
5. Within four clock cycles after setting EEMWE, write a logical "1" 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 registers will be modified, causing the interrupted EEPROM access to fail. It is recommended to have the global interrupt flag cleared during the four 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 four cycles before the next instruction is executed.

The user should poll the EEWE 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.

SPI Data Register – SPDR

Bit	7	6	5	4	3	2	1	0	
$0F ($2F)	MSB							LSB	SPDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	x	x	x	x	x	x	x	x	Undefined

The SPI Data Register is a read/write register used for data transfer between the register file and the SPI Shift Register. Writing to the register initiates data transmission. Reading the register causes the Shift Register Receive buffer to be read.

UART Control

UART I/O Data Register – UDR

Bit	7	6	5	4	3	2	1	0	
$0C ($2C)	MSB							LSB	UDR
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.

UART Status Register – USR

Bit	7	6	5	4	3	2	1	0	
$0B ($2B)	RXC	TXC	UDRE	FE	OR	–	–	–	USR
Read/Write	R	R/W	R	R	R	R	R	R	
Initial Value	0	0	1	0	0	0	0	0	

The USR register is a read-only register providing information on the UART status.

- **Bit 7 – RXC: UART Receive Complete**

This bit is set (one) when a received character is transferred from the Receiver Shift register to UDR. The bit is set regardless of any detected framing errors. When the RXCIE bit in UCR is set, the UART Receive Complete interrupt will be executed when RXC is set (one). RXC is cleared by reading UDR. When interrupt-driven data reception is used, the UART Receive Complete Interrupt routine must read UDR in order to clear RXC, otherwise a new interrupt will occur once the interrupt routine terminates.

- **Bit 6 – TXC: UART Transmit Complete**

This bit is set (one) when the entire character (including the stop bit) in the Transmit Shift register has been shifted out and no new data has been written to UDR. This flag is especially useful in half-duplex communications interfaces, where a transmitting application must enter receive mode and free the communications bus immediately after completing the transmission.

When the TXCIE bit in UCR is set, setting of TXC causes the UART Transmit Complete interrupt to be executed. TXC is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, the TXC bit is cleared (zero) by writing a logical "1" to the bit.

- **Bit 5 – UDRE: UART Data Register Empty**

This bit is set (one) when a character written to UDR is transferred to the Transmit Shift register. Setting of this bit indicates that the transmitter is ready to receive a new character for transmission.

When the UDRIE bit in UCR is set, the UART Transmit Complete interrupt to be executed as long as UDRE is set. UDRE is cleared by writing UDR. When interrupt-driven data transmittal is used, the UART Data Register Empty Interrupt routine must write UDR in order to clear UDRE, otherwise a new interrupt will occur once the interrupt routine terminates.

UDRE is set (one) during reset to indicate that the transmitter is ready.

- **Bit 4 – FE: Framing Error**

This bit is set if a Framing Error condition is detected, i.e., when the stop bit of an incoming character is zero.

BAUD Rate Generator

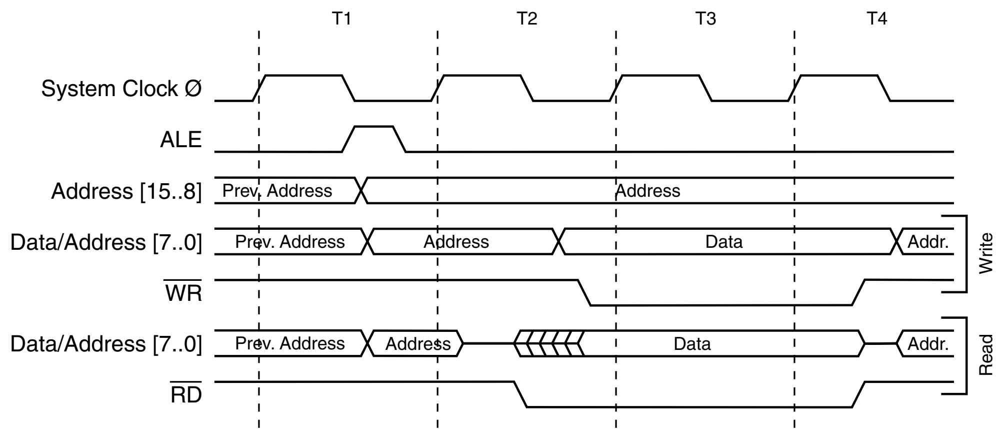
The baud rate generator is a frequency divider that generates baud rates according to the following equation:

$$\text{BAUD} = \frac{f_{\text{CK}}}{16(\text{UBRR} + 1)}$$

- BAUD = Baud rate
- f_{CK} = Crystal Clock frequency
- UBRR = Contents of the UART Baud Rate register, UBRR (0 - 255)

For standard crystal frequencies, the most commonly used baud rates can be generated by using the UBRR settings in Table 17. UBRR values that yield an actual baud rate differing less than 2% from the target baud rate are boldface in the table. However, using baud rates that have more than 1% error is not recommended. High error ratings give less noise immunity.

Figure 44. External Data SRAM Memory Cycles with Wait State



Port D Data Register – PORTD

Bit	7	6	5	4	3	2	1	0	
$12 ($32)	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	PORTD
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 D Data Direction Register – DDRD

Bit	7	6	5	4	3	2	1	0	
$11 ($31)	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	DDRD
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 D Input Pins Address – PIND

Bit	7	6	5	4	3	2	1	0	
$10 ($30)	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	PIND
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 D Input Pins address (PIND) is not a register; this address enables access to the physical value on each Port D pin. When reading PORTD, the Port D Data Latch is read and when reading PIND, the logical values present on the pins are read.

Port D as General Digital I/O

PDn, general I/O pin: The DDDn bit in the DDRD register selects the direction of this pin. If DDDn is set (one), PDn is configured as an output pin. If DDDn is cleared (zero), PDn is configured as an input pin. If PDn is set (one) when configured as an input pin, the MOS pull-up resistor is activated. To switch the pull-up resistor off the PDn has to be cleared (zero) or the pin has to be configured as an output pin. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not active.

Table 24. DDDn Bits on Port D Pins

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

Note: n: 7,6…0, pin number.

Alternate Functions of Port D

- **$\overline{\text{RD}}$ – Port D, Bit 7**

$\overline{\text{RD}}$ is the external data memory read control strobe. See "Interface to External SRAM" on page 60 for detailed information.

- **$\overline{\text{WR}}$ – Port D, Bit 6**

$\overline{\text{WR}}$ is the external data memory write control strobe. See "Interface to External SRAM" on page 60 for detailed information.

- **OC1A – Port D, Bit 5**

OC1A: Output compare match output. The PD5 pin can serve as an external output when the Timer/Counter1 compare matches. The PD5 pin has to be configured as an output (DDD5 set [one]) to serve this function. See the Timer/Counter1 description for further details and how to enable the output. The OC1A pin is also the output pin for the PWM mode timer function.

• **INT1 – Port D, Bit 3**

INT1: External Interrupt source 1. The PD3 pin can serve as an external interrupt source to the MCU. See the interrupt description for further details and how to enable the source.

• **INT0 – Port D, Bit 2**

INT0: External Interrupt source 0. The PD2 pin can serve as an external interrupt source to the MCU. See the interrupt description for further details and how to enable the source.

• **TXD – Port D, Bit 1**

Transmit Data (data output pin for the UART). When the UART transmitter is enabled, this pin is configured as an output, regardless of the value of DDRD1.

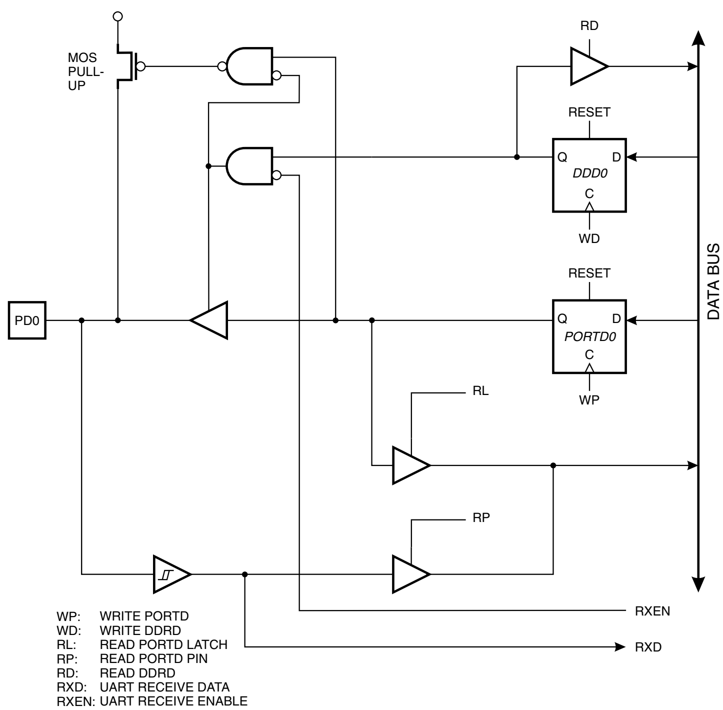
• **RXD – Port D, Bit 0**

Receive Data (data input pin for the UART). When the UART receiver is enabled, this pin is configured as an input, regardless of the value of DDRD0. When the UART forces this pin to be an input, a logical "1" in PORTD0 will turn on the internal pull-up.

Port D Schematics

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

Figure 53. Port D Schematic Diagram (Pin PD0)



Bit 5 = SPIEN Fuse bit

Bit 0 = FSTRT Fuse bit

Bit 7 - 6, 4 - 1 = “1”. These bits are reserved and should be left unprogrammed (“1”).

3. Give $\overline{WR}$ a t_{WLWH_PFB}-wide negative pulse to execute the programming, t_{WLWH_PFB} is found in Table 30. Programming the Fuse bits does not generate any activity on the RDY/$\overline{BSY}$ pin.

Programming the Lock Bits

The algorithm for programming the Lock bits is as follows (refer to “Programming the Flash” on page 81 for details on command and data loading):

1. A: Load Command “0010 0000”.
2. D: Load Data Low Byte. Bit n = “0” programs the Lock bit.

   Bit 2 = Lock Bit2

   Bit 1 = Lock Bit1

   Bit 7 - 3, 0 = “1”. These bits are reserved and should be left unprogrammed (“1”).
3. E: Write Data Low Byte.

The Lock bits can only be cleared by executing Chip Erase.

Reading the Fuse and Lock Bits

The algorithm for reading the Fuse and Lock bits is as follows (refer to “Programming the Flash” on page 81 for details on Command loading):

1. A: Load Command “0000 0100”.
2. Set $\overline{OE}$ to “0”, and BS to “1”. The status of the Fuse and Lock bits can now be read at DATA (“0” means programmed).

   Bit 7 = Lock Bit1

   Bit 6 = Lock Bit2

   Bit 5 = SPIEN Fuse bit

   Bit 0 = FSTRT Fuse bit
3. Set $\overline{OE}$ to “1”.

Observe that BS needs to be set to “1”.

Reading the Signature Bytes

The algorithm for reading the signature bytes is as follows (refer to “Programming the Flash” on page 81 for details on command and address loading):

1. A: Load Command “0000 1000”.
2. C: Load Address Low Byte ($00 - $02).

   Set $\overline{OE}$ to “0”, and BS to “0”. The selected signature byte can now be read at DATA.
3. Set $\overline{OE}$ to “1”.

Parallel Programming Characteristics

Figure 63. Parallel Programming Timing

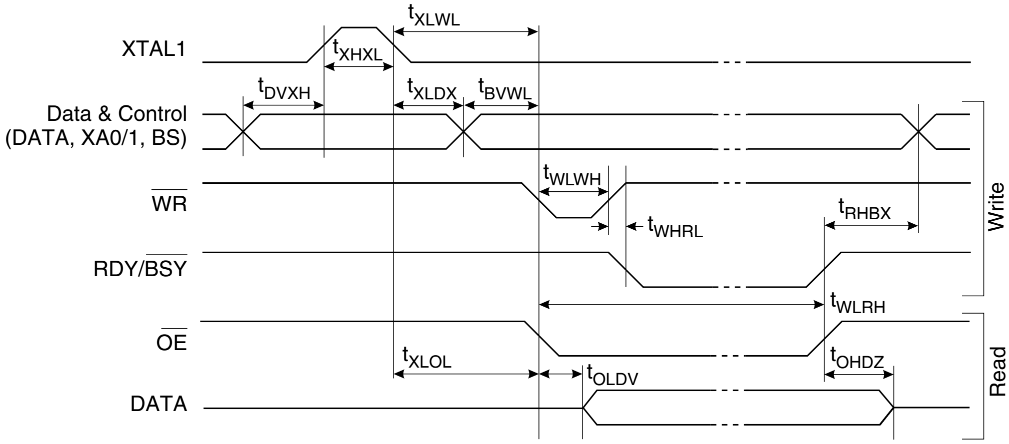


Table 30. Parallel Programming Characteristics, $T_A = 25°C \pm 10\%$, $V_{CC} = 5V \pm 10\%$

Symbol	Parameter	Min	Typ	Max	Units
V_{PP}	Programming Enable Voltage	11.5		12.5	V
I_{PP}	Programming Enable Current			250.0	µA
t_{DVXH}	Data and Control Setup before XTAL1 High	67.0			ns
t_{XHXL}	XTAL1 Pulse Width High	67.0			ns
t_{XLDX}	Data and Control Hold after XTAL1 Low	67.0			ns
t_{XLWL}	XTAL1 Low to $\overline{WR}$ Low	67.0			ns
t_{BVWL}	BS Valid to $\overline{WR}$ Low	67.0			ns
t_{RHBX}	BS Hold after RDY/$\overline{BSY}$ High	67.0			ns
t_{WLWH}	$\overline{WR}$ Pulse Width Low(1)	67.0			ns
t_{WHRL}	$\overline{WR}$ High to RDY/$\overline{BSY}$ Low(2)		20.0		ns
t_{WLRH}	$\overline{WR}$ Low to RDY/$\overline{BSY}$ High(2)	0.5	0.7	0.9	ms
t_{XLOL}	XTAL1 Low to $\overline{OE}$ Low	67.0			ns
t_{OLDV}	$\overline{OE}$ Low to DATA Valid		20.0		ns
t_{OHDZ}	$\overline{OE}$ High to DATA Tri-stated			20.0	ns
t_{WLWH_CE}	$\overline{WR}$ Pulse Width Low for Chip Erase	5.0	10.0	15.0	ms
t_{WLWH_PFB}	$\overline{WR}$ Pulse Width Low for Programming the Fuse Bits	1.0	1.5	1.8	ms

Notes: 1. Use t_{WLWH_CE} for Chip Erase and t_{WLWH_PFB} for programming the Fuse bits.
2. If t_{WLWH} is held longer than t_{WLRH}, no RDY/$\overline{BSY}$ pulse will be seen.

Electrical Characteristics

Absolute Maximum Ratings*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on Any Pin except $\overline{RESET}$ with Respect to Ground	-1.0V to V_{CC} + 0.5V
Voltage on $\overline{RESET}$ with Respect to Ground	-1.0V to +13.0V
Maximum Operating Voltage	6.6V
DC Current per I/O Pin	40.0 mA
DC Current V_{CC} and GND Pins	200.0 mA

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC Characteristics

T_A = -40°C to 85°C, V_{CC} = 2.7V to 6.0V (unless otherwise noted)

Symbol	Parameter	Condition	Min	Typ	Max	Units
V_{IL}	Input Low Voltage	(Except XTAL1)	-0.5		0.3 V_{CC}(1)	V
V_{IL1}	Input Low Voltage	(XTAL1)	-0.5		0.2 V_{CC}(1)	V
V_{IH}	Input High Voltage	(Except XTAL1, $\overline{RESET}$)	0.6 V_{CC}(2)		V_{CC} + 0.5	V
V_{IH1}	Input High Voltage	(XTAL1)	0.8 V_{CC}(2)		V_{CC} + 0.5	V
V_{IH2}	Input High Voltage	($\overline{RESET}$)	0.9 V_{CC}(2)		V_{CC} + 0.5	V
V_{OL}	Output Low Voltage(3) (Ports A, B, C, D)	I_{OL} = 20 mA, V_{CC} = 5V I_{OL} = 10 mA, V_{CC} = 3V			0.6 0.5	V V
V_{OH}	Output High Voltage(4) (Ports A, B, C, D)	I_{OH} = -3 mA, V_{CC} = 5V I_{OH} = -1.5 mA, V_{CC} = 3V	4.2 2.3			V V
I_{IL}	Input Leakage Current I/O Pin	V_{CC} = 6V, pin low (absolute value)			8.0	µA
I_{IH}	Input Leakage Current I/O Pin	V_{CC} = 6V, pin high (absolute value)			980.0	nA
RRST	Reset Pull-up Resistor		100.0		500.0	kΩ
$R_{I/O}$	I/O Pin Pull-up Resistor		35.0		120.0	kΩ
I_{CC}	Power Supply Current	Active Mode, V_{CC} = 3V, 4 MHz			3.0	mA
		Idle Mode V_{CC} = 3V, 4 MHz			1.2	mA
	Power-down mode(5)	WDT enabled, V_{CC} = 3V		9.0	15.0	µA
		WDT disabled, V_{CC} = 3V		<1.0	2.0	µA
V_{ACIO}	Analog Comparator Input Offset Voltage	V_{CC} = 5V V_{in} = V_{CC}/2			40.0	mV
I_{ACLK}	Analog Comparator Input Leakage Current	V_{CC} = 5V V_{in} = V_{CC}/2	-50.0		50.0	nA
t_{ACPD}	Analog Comparator Propagation Delay	V_{CC} = 2.7V V_{CC} = 4.0V		750.0 500.0		ns