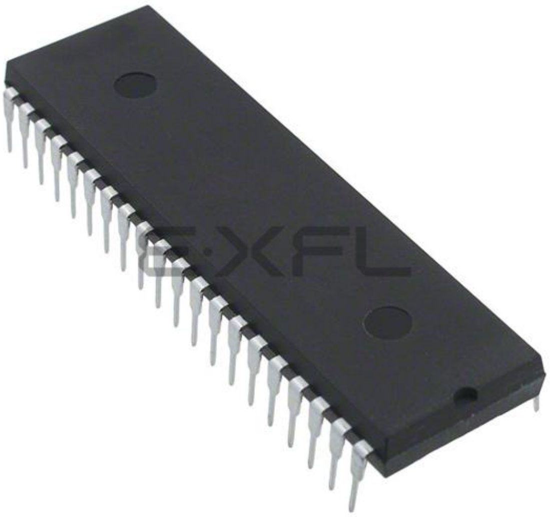
Microchip Technology - AT90S4414-8PC Datasheet



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	8MHz
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)	4V ~ 6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°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/at90s4414-8pc

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.

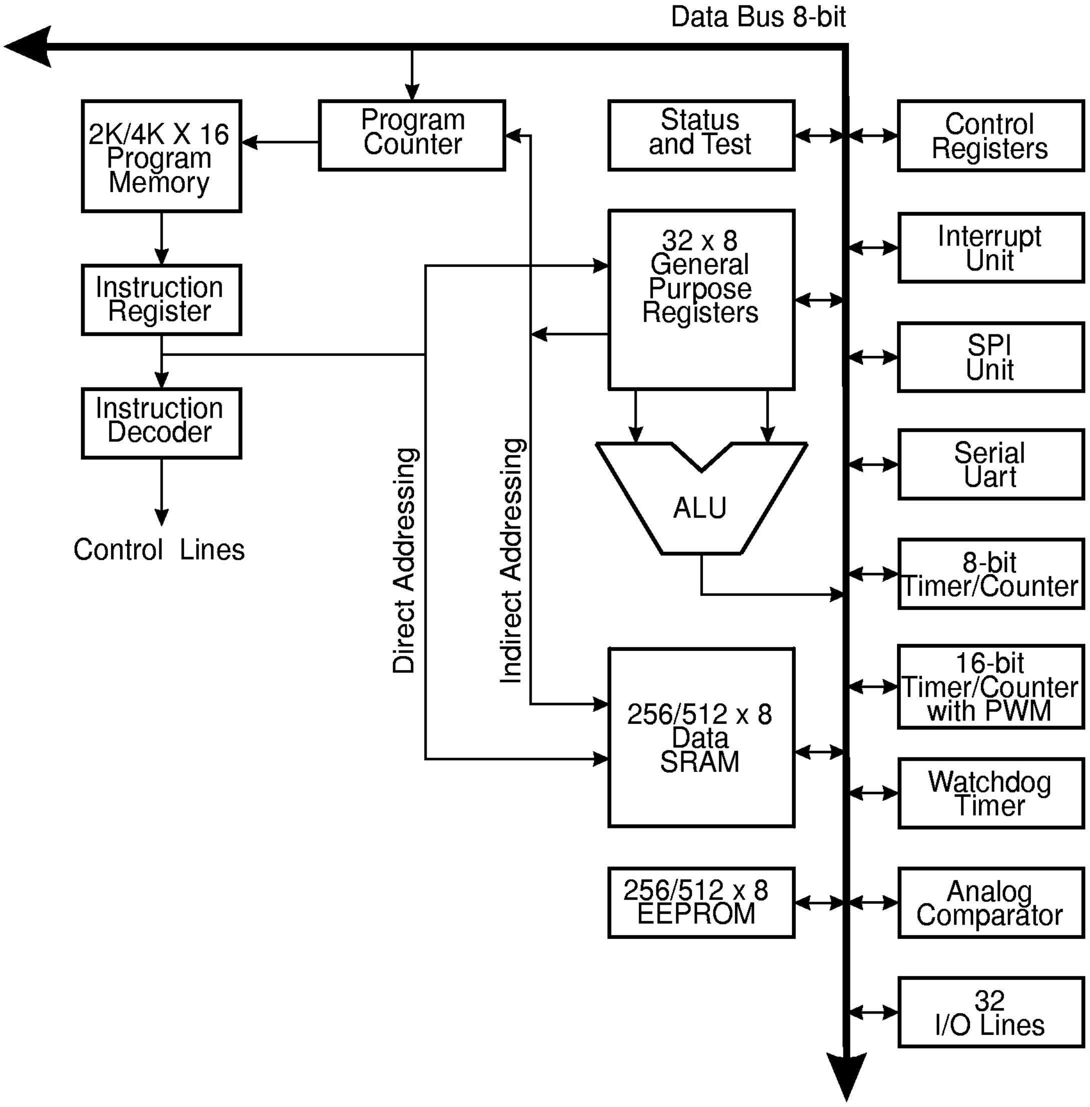
With the relative jump and call instructions, the whole 2K/4K address space is directly accessed. Most AVR instructions have a single 16-bit word format. Every program memory address contains a 16- or 32-bit instruction.

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 256/512 bytes 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.

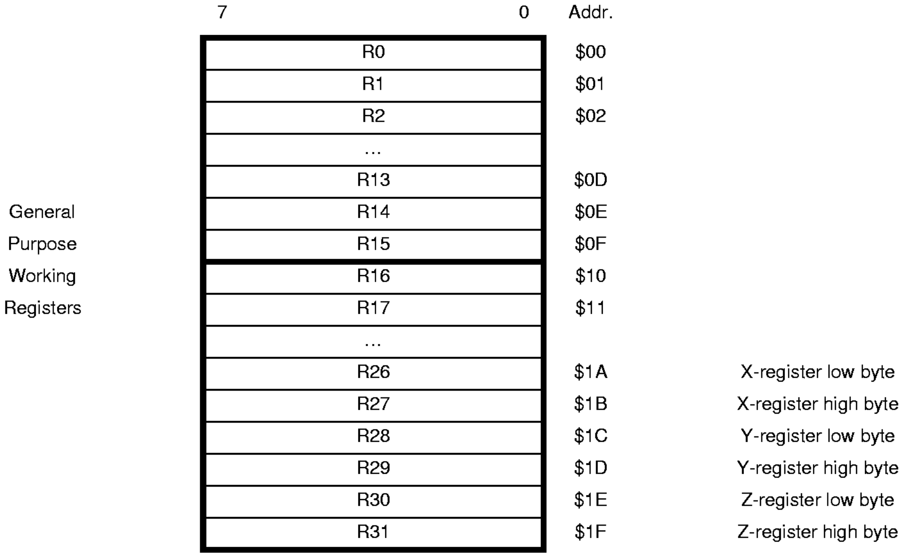
Figure 4. The AT90S4414/8515 AVR RISC Architecture



General Purpose Register File

Figure 6 shows the structure of the 32 general purpose working registers in the CPU.

Figure 6. AVR CPU General Purpose Working Registers



	7 0	Addr.	
	R0	$00	
	R1	$01	
	R2	$02	
	…		
	R13	$0D	
General	R14	$0E	
Purpose	R15	$0F	
Working	R16	$10	
Registers	R17	$11	
	…		
	R26	$1A	X-register low byte
	R27	$1B	X-register high byte
	R28	$1C	Y-register low byte
	R29	$1D	Y-register high byte
	R30	$1E	Z-register low byte
	R31	$1F	Z-register high byte

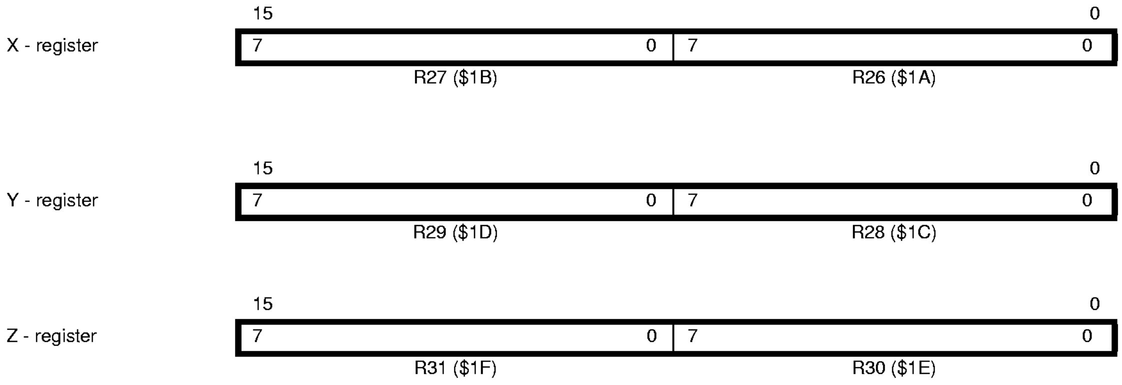
All the register operating instructions in the instruction set have direct and single cycle access to all registers. The only exception is the five constant arithmetic and logic instructions SBCI, SUBI, CPI, ANDI and ORI between a constant and a register and the LDI instruction for load immediate constant data. These instructions apply to the second half of the registers in the register file - R16..R31. The general SBC, SUB, CP, AND and OR and all other operations between two registers or on a single register apply to the entire register file.

As shown in Figure 6, each register is also assigned a data memory address, mapping them directly into the first 32 locations of the user Data Space. Although not being physically implemented as SRAM locations, this memory organization provides great flexibility in access of the registers, as the X,Y and Z registers can be set to index any register in the file.

X-Register, Y-Register And Z-Register

The registers R26..R31 have some added functions to their general purpose usage. These registers are address pointers for indirect addressing of the Data Space. The three indirect address registers X, Y and Z are defined as:

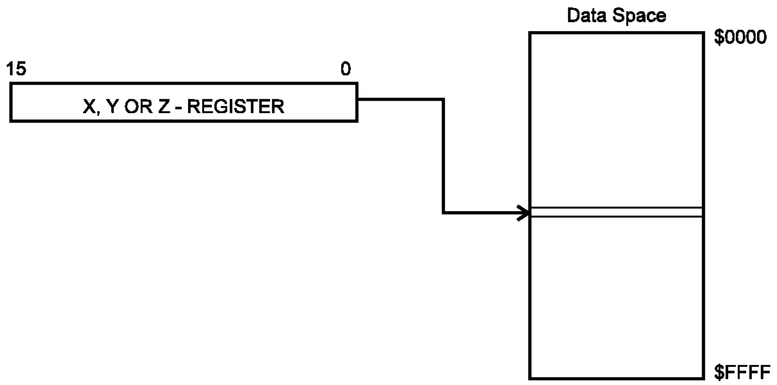
Figure 7. X, Y and Z Registers



In the different addressing modes these address registers have functions as fixed displacement, automatic increment and decrement (see the descriptions for the different instructions).

Data Indirect

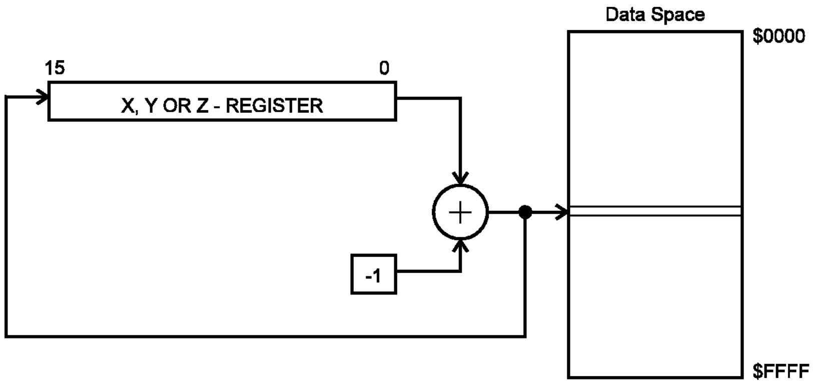
Figure 14. Data Indirect Addressing



Operand address is the contents of the X, Y or the Z-register.

Data Indirect with Pre-decrement

Figure 15. Data Indirect Addressing with Pre-decrement



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.

- **Bit 5 - H: Half Carry Flag**

The half carry flag H indicates a half carry in some arithmetic operations. See the Instruction Set Description for detailed information.

- **Bit 4 - S: Sign Bit, $S = N \oplus V$**

The S-bit is always an exclusive or between the negative flag N and the two's complement overflow flag V. See the Instruction Set Description for detailed information.

- **Bit 3 - V: Two's Complement Overflow Flag**

The two's complement overflow flag V supports two's complement arithmetics. See the Instruction Set Description for detailed information.

- **Bit 2 - N: Negative Flag**

The negative flag N indicates a negative result after the different arithmetic and logic operations. See the Instruction Set Description for detailed information.

- **Bit 1 - Z: Zero Flag**

The zero flag Z indicates a zero result after the different arithmetic and logic operations. See the Instruction Set Description for detailed information.

- **Bit 0 - C: Carry Flag**

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

Note that the status register is not automatically stored when entering an interrupt routine and restored when returning from an interrupt routine. This must be handled by software.

Stack Pointer - SP

The general AVR 16-bit Stack Pointer is effectively built up of two 8-bit registers in the I/O space locations $3E ($5E) and $3D ($5D). As the AT90S4414/8515 supports up to 64 kB external SRAM, all 16-bits are used.

Bit	15	14	13	12	11	10	9	8	
$3E ($5E)	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SPH
$3D ($5D)	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	SPL
	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 Stack Pointer points to the data SRAM stack area where the Subroutine and Interrupt Stacks are located. This Stack space in the data SRAM must be defined by the program before any subroutine calls are executed or interrupts are enabled. The stack pointer must be set to point above $60. The Stack Pointer is decremented by one when data is pushed onto the Stack with the PUSH instruction, and it is decremented by two when an address is pushed onto the Stack with subroutine calls and interrupts. The Stack Pointer is incremented by one when data is popped from the Stack with the POP instruction, and it is incremented by two when an address is popped from the Stack with return from subroutine RET or return from interrupt RETI.

Reset and Interrupt Handling

The AT90S4414/8515 provides 12 different interrupt sources. These interrupts and the separate reset vector, each have a separate program vector in the program memory space. All interrupts are assigned individual enable bits which must be set (one) together with the I-bit in the status register in order to enable the interrupt.

The lowest addresses in the program memory space are automatically defined as the Reset and Interrupt vectors. The complete list of vectors is shown in Table 3. The list also determines the priority levels of the different interrupts. The lower the address the higher is the priority level. RESET has the highest priority, and next is INT0 - the External Interrupt Request 0 etc.

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 $\overline{\text{RESET}}$ 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

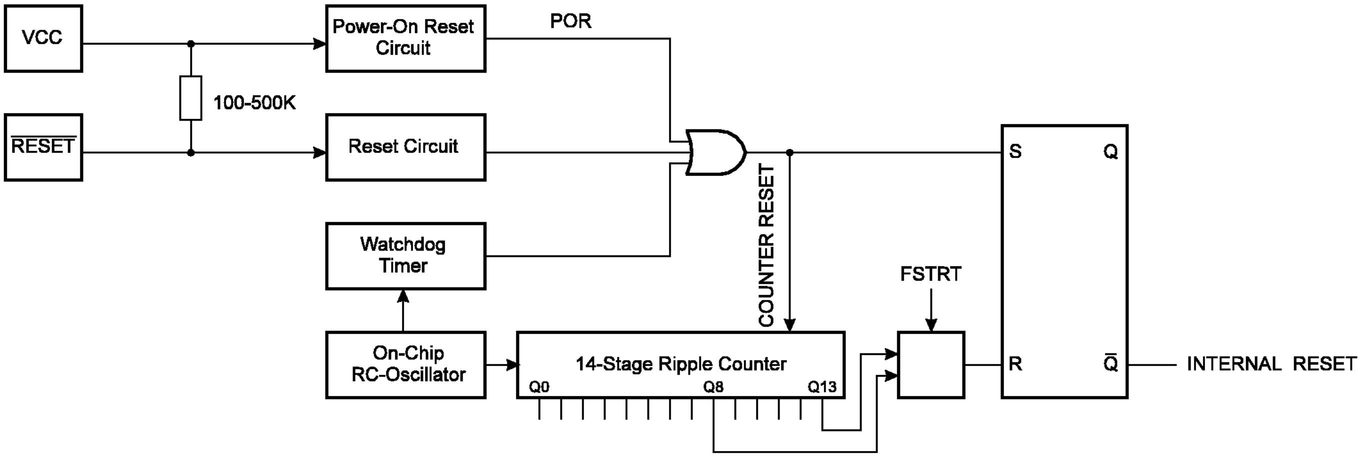


Table 4. Reset Characteristics

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}	$\overline{\text{RESET}}$ 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

Power-on Reset

A Power-on Reset (POR) circuit ensures that the device is reset from power-on. As shown in Figure 23, an internal timer clocked from the Watchdog timer oscillator prevents the MCU from starting until after a certain period after V_{CC} has reached the Power-on Threshold voltage - V_{POT}, regardless of the V_{CC} rise time (see Figure 24). The FSTRT Fuse bit in the Flash can be programmed to give a shorter start-up time if a ceramic resonator or any other fast-start oscillator is used to clock the MCU.

If the build-in start-up delay is sufficient, $\overline{\text{RESET}}$ can be connected to V_{CC} directly or via an external pull-up resistor. By holding the pin low for a period after V_{CC} has been applied, the Power-on Reset period can be extended. Refer to Figure 25 for a timing example on this.

Figure 24. MCU Start-up, $\overline{\text{RESET}}$ Tied to V_{CC}.

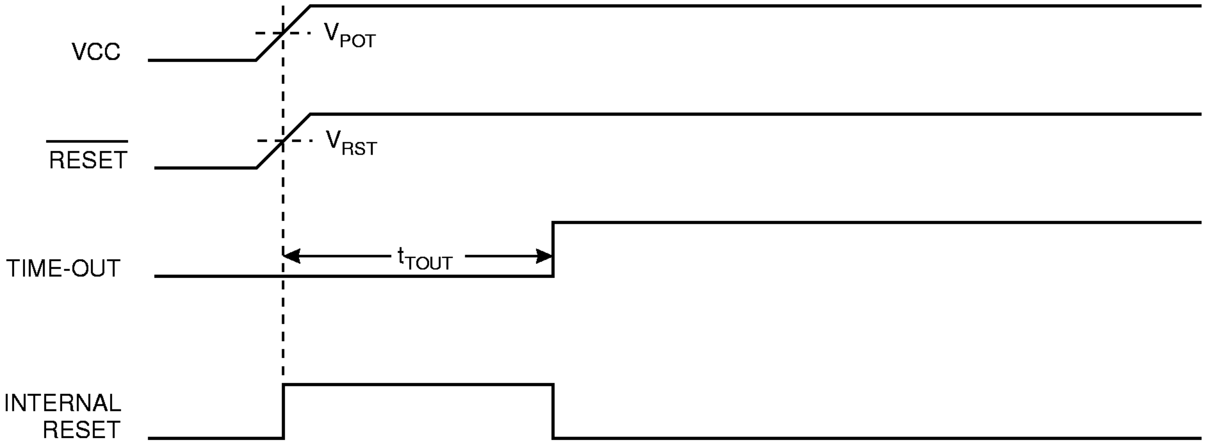


Figure 25. MCU Start-up, $\overline{\text{RESET}}$ Controlled Externally

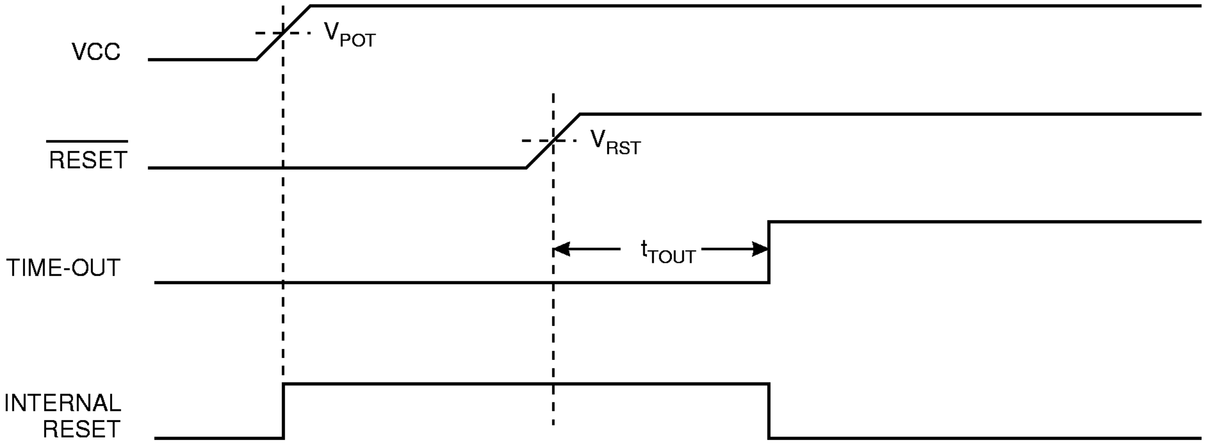
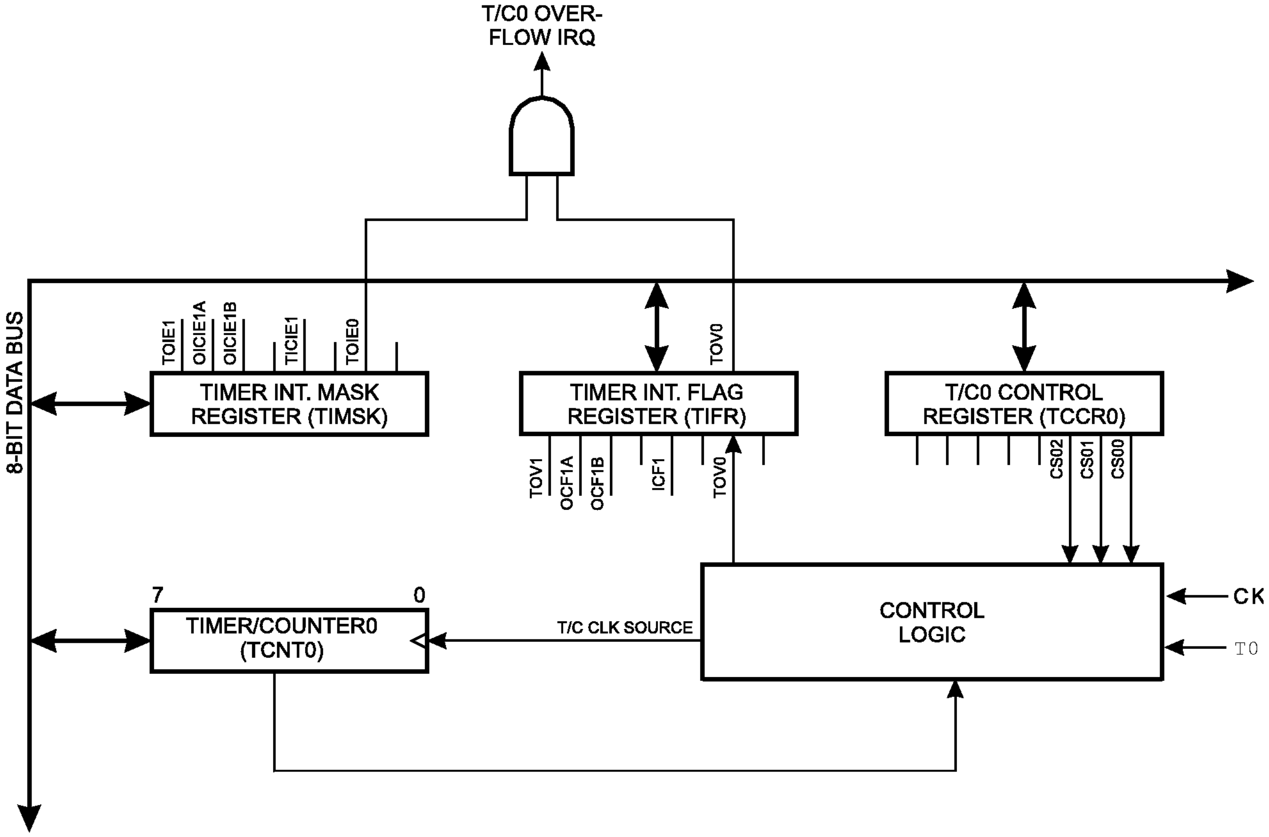


Figure 29. Timer/Counter0 Block Diagram



Timer/Counter0 Control Register - TCCR0

Bit	7	6	5	4	3	2	1	0	
$33 ($53)	-	-	-	-	-	CS02	CS01	CS00	TCCR0
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

- **Bits 7..3 - Res: Reserved bits**

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

- **Bits 2,1,0 - CS02, CS01, CS00: Clock Select0, bit 2,1 and 0**

The Clock Select0 bits 2,1 and 0 define the prescaling source of Timer/Counter0.

Table 8. Clock 0 Prescale Select

CS02	CS01	CS00	Description
0	0	0	Stop, the Timer/Counter0 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 T0, falling edge
1	1	1	External Pin T0, rising edge

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 the specification for the Timer/Counter1 Control Registers - TCCR1A and 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.

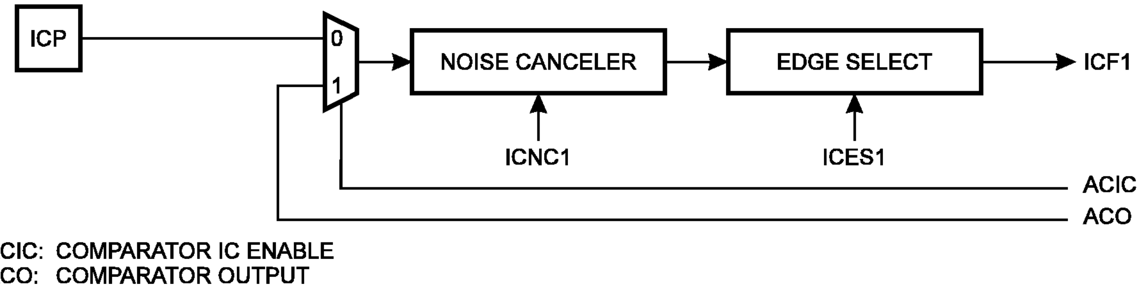
The 16-bit Timer/Counter1 features both a high resolution and a high accuracy usage with the lower prescaling opportunities. Similarly, the high prescaling opportunities makes the Timer/Counter1 useful for lower speed functions or exact timing functions with infrequent actions.

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 a 8, 9 or 10-bit Pulse With Modulator. In this mode the counter and the OCR1A/OCR1B registers serve as a dual glitch-free stand-alone PWM with centered pulses. Refer to page 41 for a detailed description on 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 31.

Figure 31. ICP Pin Schematic Diagram



If the noise canceler function is enabled, the actual trigger condition for the capture event is monitored over 4 samples, and all 4 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	-	-	PWM11	PWM10	TCCR1A
Read/Write	R/W	R/W	R/W	R/W	R	R	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 the output pin. The control configuration is shown in Table 9.

- **Bits 5,4 - COM1B1, COM1B0: Compare Output Mode1B, bits 1 and 0**

The COM1B1 and COM1B0 control bits determine any output pin action following a compare match in Timer/Counter1. Any output pin actions affect pin OC1B - Output CompareB. The following control configuration is given:

- **Bit 5 - UDRIE: UART Data Register Empty Interrupt Enable**

When this bit is set (one), a setting of the UDRE bit in USR will cause the UART Data Register Empty interrupt routine to be executed provided that global interrupts are enabled.

- **Bit 4 - RXEN: Receiver Enable**

This bit enables the UART receiver when set (one). When the receiver is disabled, the TXC, OR and FE status flags cannot become set. If these flags are set, turning off RXEN does not cause them to be cleared.

- **Bit 3 - TXEN: Transmitter Enable**

This bit enables the UART transmitter when set (one). When disabling the transmitter while transmitting a character, the transmitter is not disabled before the character in the shift register plus any following character in UDR has been completely transmitted.

- **Bit 2 - CHR9: 9-bit Characters**

When this bit is set (one) transmitted and received characters are 9 bit long plus start and stop bits. The 9th bit is read and written by using the RXB8 and TXB8 bits in UCR, respectively. The 9th data bit can be used as an extra stop bit or a parity bit.

- **Bit 1 - RXB8: Receive Data Bit 8**

When CHR9 is set (one), RXB8 is the 9th data bit of the received character.

- **Bit 0 - TXB8: Transmit Data Bit 8**

When CHR9 is set (one), TXB8 is the 9th data bit in the character to be transmitted.

BAUD Rate Generator

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

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

- BAUD = Baud-Rate
- fck= 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 18. UBRR values which yield an actual baud rate differing less than 2% from the target baud rate, are bolded in the table. However, using baud rates that have more than 1% error is not recommended. High error ratings give less noise immunity.

Table 18. UBRR Settings at Various Crystal Frequencies

Baud Rate	**1 MHz**	%Error	**1.8432 MHz**	%Error	**2 MHz**	%Error	**2.4576 MHz**	%Error
2400	UBRR= **25**	**0.2**	UBRR= **47**	**0.0**	UBRR= **51**	**0.2**	UBRR= **63**	**0.0**
4800	UBRR= **12**	**0.2**	UBRR= **23**	**0.0**	UBRR= **25**	**0.2**	UBRR= **31**	**0.0**
9600	UBRR= 6	7.5	UBRR= **11**	**0.0**	UBRR= **12**	**0.2**	UBRR= **15**	**0.0**
14400	UBRR= 3	7.8	UBRR= **7**	**0.0**	UBRR= 8	3.7	UBRR= 10	3.1
19200	UBRR= 2	7.8	UBRR= **5**	**0.0**	UBRR= 6	7.5	UBRR= **7**	**0.0**
28800	UBRR= 1	7.8	UBRR= **3**	**0.0**	UBRR= 3	7.8	UBRR= 4	6.3
38400	UBRR= 1	22.9	UBRR= **2**	**0.0**	UBRR= 2	7.8	UBRR= **3**	**0.0**
57600	UBRR= 0	7.8	UBRR= **1**	**0.0**	UBRR= 1	7.8	UBRR= 2	12.5
76800	UBRR= 0	22.9	UBRR= 1	33.3	UBRR= 1	22.9	UBRR= **1**	**0.0**
115200	UBRR= 0	84.3	UBRR= **0**	**0.0**	UBRR= 0	7.8	UBRR= 0	25.0

Baud Rate	**3.2768 MHz**	%Error	**3.6864 MHz**	%Error	**4 MHz**	%Error	**4.608 MHz**	%Error
2400	UBRR= **84**	**0.4**	UBRR= **95**	**0.0**	UBRR= **103**	**0.2**	UBRR= **119**	**0.0**
4800	UBRR= **42**	**0.8**	UBRR= **47**	**0.0**	UBRR= **51**	**0.2**	UBRR= **59**	**0.0**
9600	UBRR= **20**	**1.6**	UBRR= **23**	**0.0**	UBRR= **25**	**0.2**	UBRR= **29**	**0.0**
14400	UBRR= **13**	**1.6**	UBRR= **15**	**0.0**	UBRR= 16	2.1	UBRR= **19**	**0.0**
19200	UBRR= 10	3.1	UBRR= **11**	**0.0**	UBRR= **12**	**0.2**	UBRR= **14**	**0.0**
28800	UBRR= **6**	**1.6**	UBRR= **7**	**0.0**	UBRR= 8	3.7	UBRR= **9**	**0.0**
38400	UBRR= 4	6.3	UBRR= **5**	**0.0**	UBRR= 6	7.5	UBRR= 7	6.7
57600	UBRR= 3	12.5	UBRR= **3**	**0.0**	UBRR= 3	7.8	UBRR= **4**	**0.0**
76800	UBRR= 2	12.5	UBRR= **2**	**0.0**	UBRR= 2	7.8	UBRR= 3	6.7
115200	UBRR= 1	12.5	UBRR= **1**	**0.0**	UBRR= 1	7.8	UBRR= 2	20.0

Baud Rate	**7.3728 MHz**	%Error	**8 MHz**	%Error	**9.216 MHz**	%Error	**11.059 MHz**	%Error
2400	UBRR= **191**	**0.0**	UBRR= **207**	**0.2**	UBRR= **239**	**0.0**	UBRR= 287	-
4800	UBRR= **95**	**0.0**	UBRR= **103**	**0.2**	UBRR= **119**	**0.0**	UBRR= **143**	**0.0**
9600	UBRR= **47**	**0.0**	UBRR= **51**	**0.2**	UBRR= **59**	**0.0**	UBRR= **71**	**0.0**
14400	UBRR= **31**	**0.0**	UBRR= **34**	**0.8**	UBRR= **39**	**0.0**	UBRR= **47**	**0.0**
19200	UBRR= **23**	**0.0**	UBRR= **25**	**0.2**	UBRR= **29**	**0.0**	UBRR= **35**	**0.0**
28800	UBRR= **15**	**0.0**	UBRR= 16	2.1	UBRR= **19**	**0.0**	UBRR= **23**	**0.0**
38400	UBRR= **11**	**0.0**	UBRR= **12**	**0.2**	UBRR= **14**	**0.0**	UBRR= **17**	**0.0**
57600	UBRR= **7**	**0.0**	UBRR= 8	3.7	UBRR= **9**	**0.0**	UBRR= **11**	**0.0**
76800	UBRR= **5**	**0.0**	UBRR= 6	7.5	UBRR= 7	6.7	UBRR= **8**	**0.0**
115200	UBRR= **3**	**0.0**	UBRR= 3	7.8	UBRR= **4**	**0.0**	UBRR= **5**	**0.0**

UART BAUD Rate Register - UBRR

Bit	7	6	5	4	3	2	1	0	
$09 ($29)	**MSB**							**LSB**	**UBRR**
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 UBRR register is an 8-bit read/write register which specifies the UART Baud Rate according to the equation on the previous page.

Table 20. DDAn Effects on Port A Pins

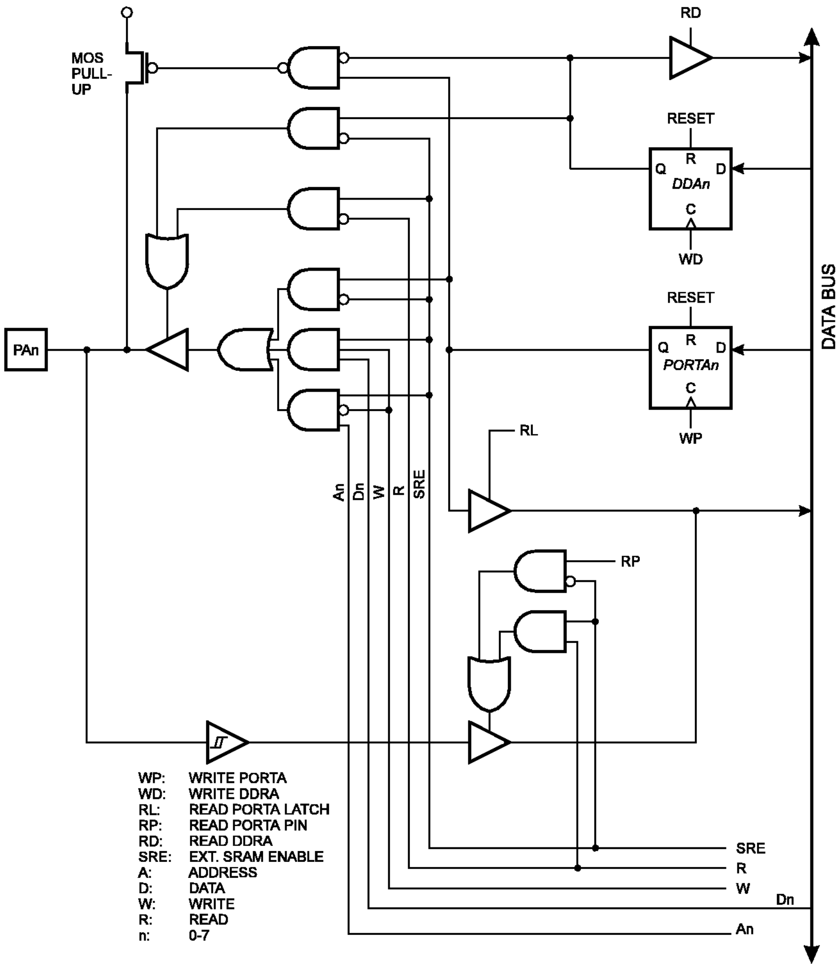
DDAn	PORTAn	I/O	Pull up	Comment
0	0	Input	No	Tri-state (Hi-Z)
0	1	Input	Yes	PAn will source current if ext. pulled low.
1	0	Output	No	Push-Pull Zero Output
1	1	Output	No	Push-Pull One Output

n: 7,6…0, pin number.

Port A Schematics

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

Figure 45. Port A Schematic Diagrams (Pins PA0 - PA7)



Port C

Port C is an 8-bit bi-directional I/O port. Three I/O memory address locations are allocated for the Port C, one each for the Data Register - PORTC, $15($35), Data Direction Register - DDRC, $14($34) and the Port C Input Pins - PINC, $13($33). The Port C 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 C output buffers can sink 20mA and thus drive LED displays directly. When pins PC0 to PC7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated.

The Port C pins have alternate functions related to the optional external data SRAM. Port C can be configured to be the high-order address byte during accesses to external data memory. When Port C 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 C Data Register - PORTC

Bit	7	6	5	4	3	2	1	0	
$15 ($35)	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	PORTC
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 C Data Direction Register - DDRC

Bit	7	6	5	4	3	2	1	0	
$14 ($34)	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	DDRC
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 C Input Pins Address - PINC

Bit	7	6	5	4	3	2	1	0	
$13 ($33)	PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	PINC
Read/Write	R	R	R	R	R	R	R	R	
Initial value	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	

The Port C Input Pins address - PINC - is not a register, and this address enables access to the physical value on each Port C pin. When reading PORTC, the Port C Data Latch is read, and when reading PINC, the logical values present on the pins are read.

PortC as General Digital I/O

All 8 pins in Port C have equal functionality when used as digital I/O pins.

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

Table 23. DDCn Effects on Port C Pins

DDCn	PORTCn	I/O	Pull up	Comment
0	0	Input	No	Tri-state (Hi-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

n: 7…0, pin number

Table 28. Pin Name Mapping

Signal Name in Programming Mode	Pin Name	I/O	Function
RDY/$\overline{\text{BSY}}$	PD1	O	0: Device is busy programming, 1: Device is ready for new command
$\overline{\text{OE}}$	PD2	I	Output Enable (Active low)
$\overline{\text{WR}}$	PD3	I	Write Pulse (Active low)
BS	PD4	I	Byte Select (“0” selects low byte, “1” selects high byte)
XA0	PD5	I	XTAL Action Bit 0
XA1	PD6	I	XTAL Action Bit 1
DATA	PB7-0	I/O	Bidirectional Databus (Output when $\overline{\text{OE}}$ is low)

Table 29. XA1 and XA0 Coding

XA1	XA0	Action when XTAL1 is Pulsed
0	0	Load Flash or EEPROM Address (High or low address byte determined by BS)
0	1	Load Data (High or Low data byte for Flash determined by BS)
1	0	Load Command
1	1	No Action, Idle

Table 30. Command Byte Bit Coding

Command Byte	Command Executed
1000 0000	Chip Erase
0100 0000	Write Fuse Bits
0010 0000	Write Lock Bits
0001 0000	Write Flash
0001 0001	Write EEPROM
0000 1000	Read Signature Bytes
0000 0100	Read Lock and Fuse Bits
0000 0010	Read Flash
0000 0011	Read EEPROM

Enter Programming Mode

The following algorithm puts the device in parallel programming mode:

1. Apply supply voltage according to Table 27, between V_{CC} and GND.
2. Set the $\overline{\text{RESET}}$ and BS pin to “0” and wait at least 100 ns.
3. Apply 11.5 - 12.5V to $\overline{\text{RESET}}$. Any activity on BS within 100 ns after +12V has been applied to $\overline{\text{RESET}}$, will cause the device to fail entering programming mode.

Reading the Flash

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

1. A: Load Command “0000 0010”.
2. B: Load Address High Byte ($00 - $07/$0F).
3. C: Load Address Low Byte ($00 - $FF).
4. Set $\overline{\text{OE}}$ to “0”, and BS to “0”. The Flash word low byte can now be read at DATA.
5. Set BS to “1”. The Flash word high byte can now be read from DATA.
6. Set $\overline{\text{OE}}$ to “1”.

Programming the EEPROM

The programming algorithm for the EEPROM data memory is as follows (refer to Programming the Flash for details on Command, Address and Data loading):

1. A: Load Command “0001 0001”.
2. (AT90S8515 only) B: Load Address High Byte ($00 - $01)
3. C: Load Address Low Byte ($00 - $FF).
4. D: Load Data Low Byte ($00 - $FF).
5. E: Write Data Low Byte.

Reading the EEPROM

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

1. A: Load Command “0000 0011”.
2. (AT90S8515 only) B: Load Address High Byte ($00 - $01)
3. C: Load Address Low Byte ($00 - $FF).
4. Set $\overline{\text{OE}}$ to “0”, and BS to “0”. The EEPROM data byte can now be read at DATA.
5. Set $\overline{\text{OE}}$ to “1”.

Programming the Fuse Bits

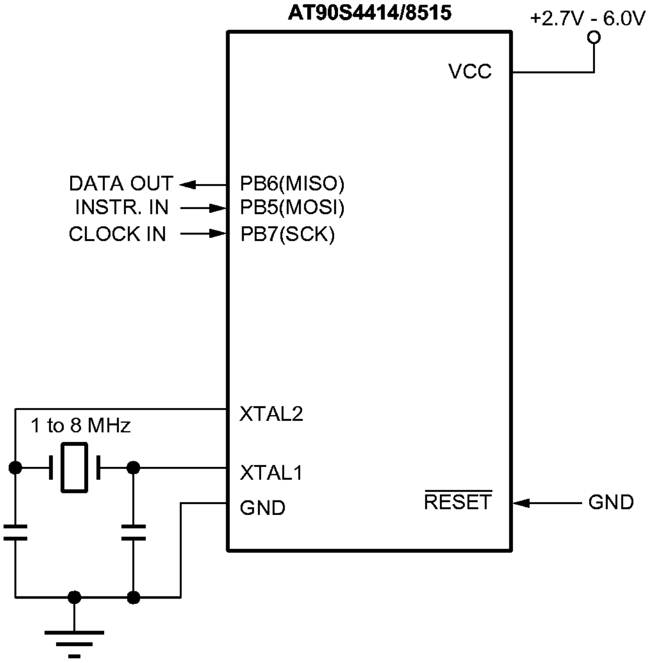
The algorithm for programming the Fuse bits is as follows (refer to Programming the Flash for details on Command and Data loading):

1. A: Load Command “0100 0000”.
2. D: Load Data Low Byte. Bit n = ‘0’ programs and bit n = “1” erases the Fuse bit.
   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 $\overline{\text{WR}}$ a t_{WLWH_PFB} wide negative pulse to execute the programming, t_{WLWH_PFB} is found in Table 31. Programming the Fuse bits does not generate any activity on the RDY/$\overline{\text{BSY}}$ pin.

Serial Downloading

Both the Program and Data memory arrays can be programmed using the SPI bus while $\overline{\text{RESET}}$ is pulled to GND. The serial interface consists of pins SCK, MOSI (input) and MISO (output), see Figure 64. After $\overline{\text{RESET}}$ is set low, the Programming Enable instruction needs to be executed first before program/erase instructions can be executed.

Figure 64. Serial Programming and Verify



For the EEPROM, an auto-erase cycle is provided within the self-timed write instruction and there is no need to first execute the Chip Erase instruction. The Chip Erase instruction turns the content 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 $07FF/$0FFF (AT90S4414/8515) for Program memory and $0000 to $00FF/$01FF (AT90S4414/8515) for EEPROM memory.

Either an external 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

Serial Programming Characteristics

Figure 66. Serial Programming Timing

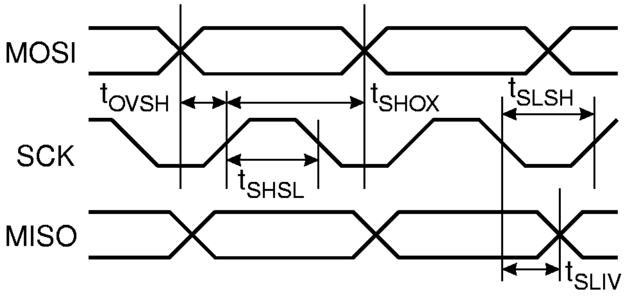


Table 34. Serial Programming Characteristics, T_A = -40°C to 85°C, V_{CC} = 2.7 - 6.0V (Unless otherwise noted)

Symbol	Parameter	Min	Typ	Max	Units
$1/t_{CLCL}$	Oscillator Frequency (V_{CC} = 2.7 - 4.0V)	0		4	MHz
t_{CLCL}	Oscillator Period (V_{CC} = 2.7 - 4.0V)	250			ns
$1/t_{CLCL}$	Oscillator Frequency (V_{CC} = 4.0 - 6.0V)	0		8	MHz
t_{CLCL}	Oscillator Period (V_{CC} = 4.0 - 6.0V)	125			ns
t_{SHSL}	SCK Pulse Width High	2 t_{CLCL}			ns
t_{SLSH}	SCK Pulse Width Low	2 t_{CLCL}			ns
t_{OVSH}	MOSI Setup to SCK High	t_{CLCL}			ns
t_{SHOX}	MOSI Hold after SCK High	2 t_{CLCL}			ns
t_{SLIV}	SCK Low to MISO Valid	10	16	32	ns

Table 35. Minimum Wait Delay after the Chip Erase Instruction

Symbol	3.2V	3.6V	4.0V	5.0V
t_{WD_ERASE}	18 ms	14 ms	12 ms	8 ms

Table 36. Minimum Wait Delay after Writing a Flash or EEPROM Location

Symbol	3.2V	3.6V	4.0V	5.0V
t_{WD_PROG}	9 ms	7 ms	6 ms	4 ms

External Clock Drive Waveforms

Figure 67. External Clock

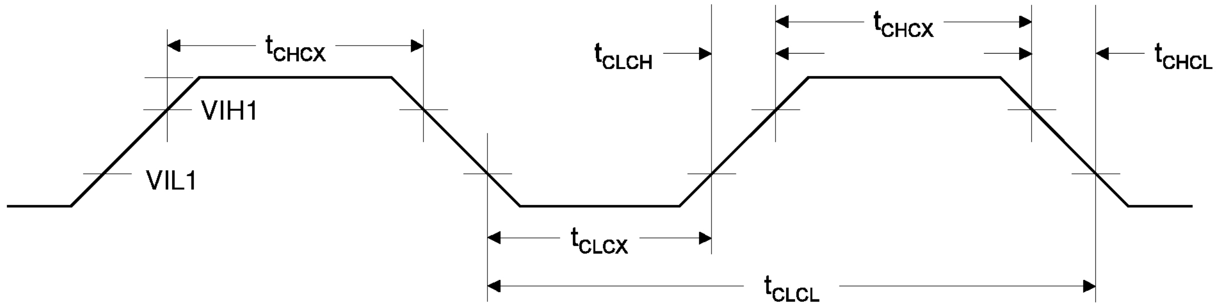
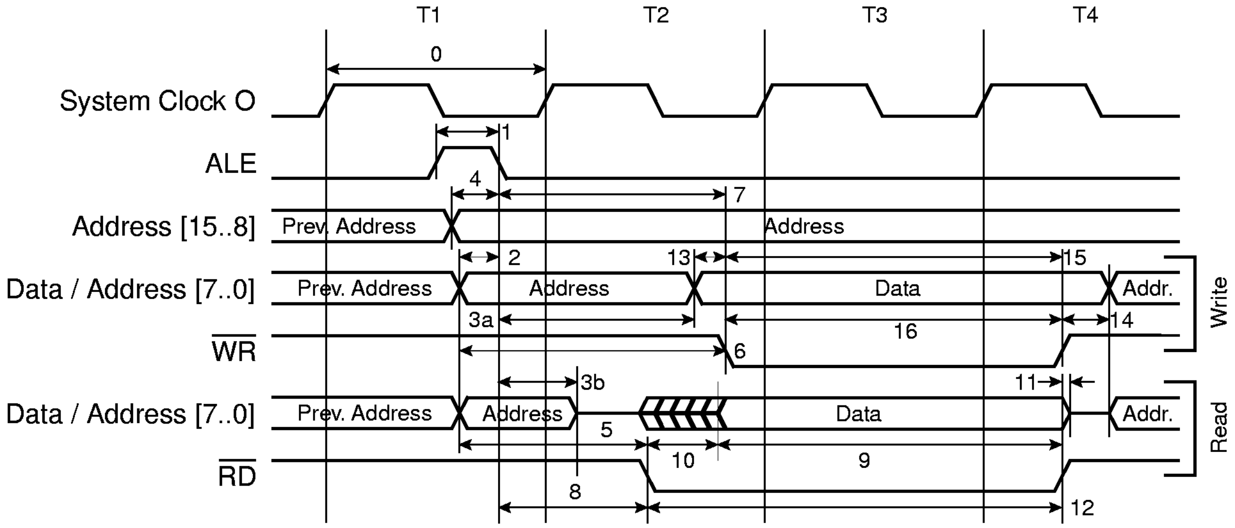


Table 37. External Clock Drive

Symbol	Parameter	V_{CC} = 2.7V to 4.0V		V_{CC} = 4.0V to 6.0V		Units
		Min	**Max**	**Min**	**Max**	
$1/t_{CLCL}$	Oscillator Frequency	0	4	0	8	MHz
t_{CLCL}	Clock Period	250		125		ns
t_{CHCX}	High Time	100		50		ns
t_{CLCX}	Low Time	100		50		ns
t_{CLCH}	Rise Time		1.6		0.5	µs
t_{CHCL}	Fall Time		1.6		0.5	µs

Note: See “External Data Memory Timing” on page 84. for a description of how the duty cycle influences the timing for the External Data Memory

Figure 68. External RAM Timing



Note: Clock cycle T3 is only present when external SRAM waitstate is enabled

T3 is only present when wait-state is enabled.

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

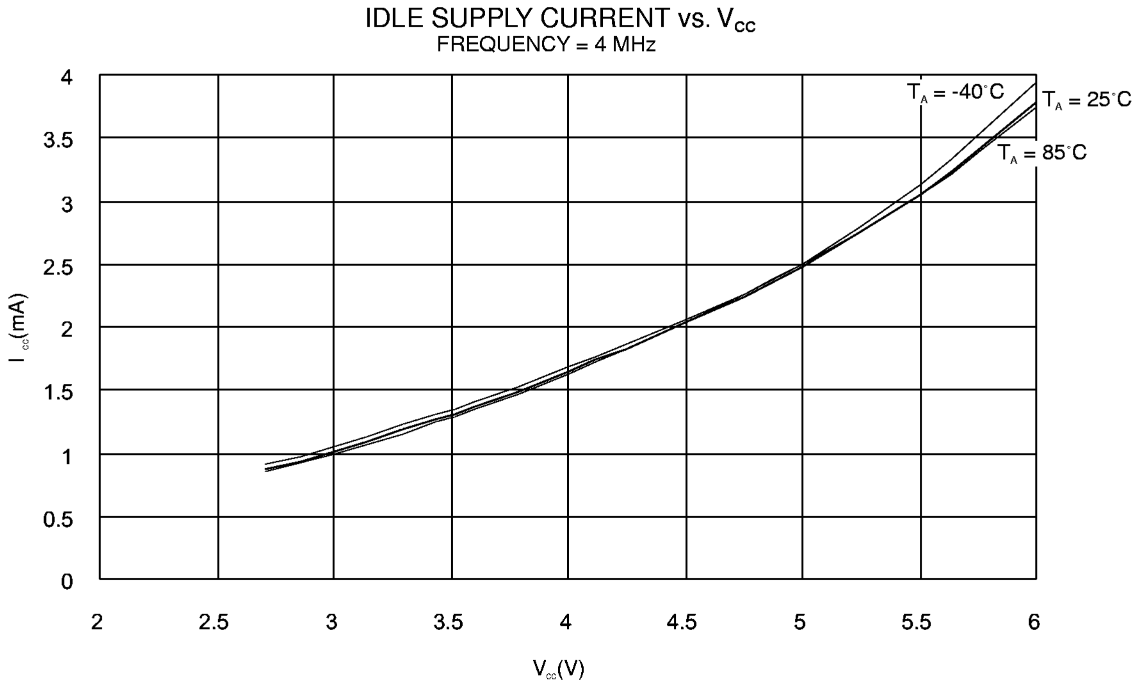
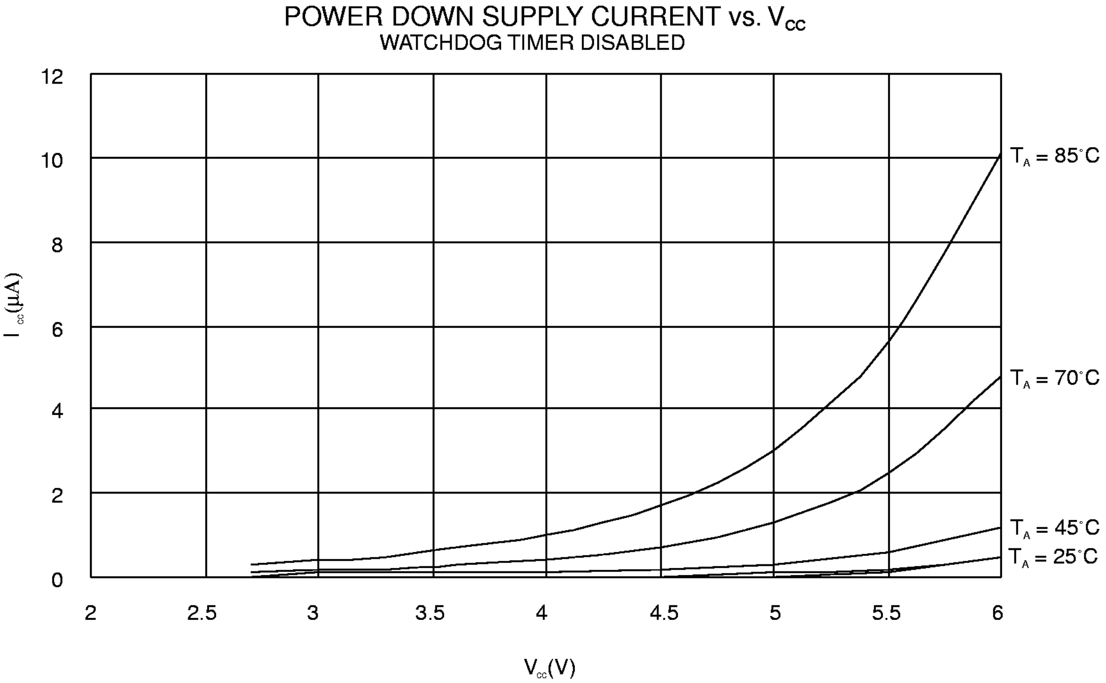


Figure 73. Power Down Supply Current vs. V_{CC}



Register Summary

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
$3F ($5F)	SREG	I	T	H	S	V	N	Z	C	18
$3E ($5E)	SPH	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	19
$3D ($5D)	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	19
$3C ($5C)	Reserved									
$3B ($5B)	GIMSK	INT1	INT0	-	-	-	-	-	-	24
$3A ($5A)	GIFR	INTF1	INTF0							24
$39 ($59)	TIMSK	TOIE1	OCIE1A	OCIE1B	-	TICIE1	-	TOIE0	-	25
$38 ($58)	TIFR	TOV1	OCF1A	OCF1B	-	ICF1	-	TOV0	-	25
$37 ($57)	Reserved									
$36 ($56)	Reserved									
$35 ($55)	MCUCR	SRE	SRW	SE	SM	ISC11	ISC10	ISC01	ISC00	27
$34 ($54)	Reserved									
$33 ($53)	TCCR0	-	-	-	-	-	CS02	CS01	CS00	30
$32 ($52)	TCNT0	Timer/Counter0 (8 Bit)								31
...	Reserved									
$2F ($4F)	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	-	-	PWM11	PWM10	32
$2E ($4E)	TCCR1B	ICNC1	ICES1	-	-	CTC1	CS12	CS11	CS10	33
$2D ($4D)	TCNT1H	Timer/Counter1 - Counter Register High Byte								34
$2C ($4C)	TCNT1L	Timer/Counter1 - Counter Register Low Byte								34
$2B ($4B)	OCR1AH	Timer/Counter1 - Output Compare Register A High Byte								35
$2A ($4A)	OCR1AL	Timer/Counter1 - Output Compare Register A Low Byte								35
$29 ($49)	OCR1BH	Timer/Counter1 - Output Compare Register B High Byte								35
$28 ($48)	OCR1BL	Timer/Counter1 - Output Compare Register B Low Byte								35
...	Reserved									
$25 ($45)	ICR1H	Timer/Counter1 - Input Capture Register High Byte								36
$24 ($44)	ICR1L	Timer/Counter1 - Input Capture Register Low Byte								36
...	Reserved									
$21 ($41)	WDTCR	-	-	-	WDTOE	WDE	WDP2	WDP1	WDP0	38
$20 ($40)	Reserved									
$1F ($3F)	EEARH[1]	-	-	-	-	-	-	-	EEAR8	39
$1E ($3E)	EEARL	EEPROM Address Register Low Byte								39
$1D ($3D)	EEDR	EEPROM Data Register								40
$1C ($3C)	EECR	-	-	-	-	-	EEMWE	EEWE	EERE	40
$1B ($3B)	PORTA	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	55
$1A ($3A)	DDRA	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	55
$19 ($39)	PINA	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	55
$18 ($38)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	57
$17 ($37)	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	57
$16 ($36)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	57
$15 ($35)	PORTC	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	62
$14 ($34)	DDRC	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	62
$13 ($33)	PINC	PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	62
$12 ($32)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	64
$11 ($31)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	64
$10 ($30)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	64
$0F ($2F)	SPDR	SPI Data Register								45
$0E ($2E)	SPSR	SPIF	WCOL	-	-	-	-	-	-	45
$0D ($2D)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	44
$0C ($2C)	UDR	UART I/O Data Register								48
$0B ($2B)	USR	RXC	TXC	UDRE	FE	OR	-	-	-	49
$0A ($2A)	UCR	RXCIE	TXCIE	UDRIE	RXEN	TXEN	CHR9	RXB8	TXB8	49
$09 ($29)	UBRR	UART Baud Rate Register								51
$08 ($28)	ACSR	ACD	-	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	52
...	Reserved									
$00 ($20)	Reserved									

Notes:
1. EEARH only present for AT90S8515
2. For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.
3. Some of the status flags are cleared by writing a logical one to them. Note that the CBI and SBI instructions will operate on all bits in the I/O register, writing a one back into any flag read as set, thus clearing the flag. The CBI and SBI instructions work with registers $00 to $1F only.

Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
4	2.7 - 6.0V	AT90S4414-4AC AT90S4414-4JC AT90S4414-4PC	44A 44J 40P6	Commercial (0°C to 70°C)
		AT90S4414-4AI AT90S4414-4JI AT90S4414-4PI	44A 44J 40P6	Industrial (-40°C to 85°C)
8	4.0 - 6.0V	AT90S4414-8AC AT90S4414-8JC AT90S4414-8PC	44A 44J 40P6	Commercial (0°C to 70°C)
		AT90S4414-8AI AT90S4414-8JI AT90S4414-8PI	44A 44J 40P6	Industrial (-40°C to 85°C)

Note: Order AT904414A-XXX for devices with the FSTRT Fuse programmed.

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
4	2.7 - 6.0V	AT90S8515-4AC AT90S8515-4JC AT90S8515-4PC	44A 44J 40P6	Commercial (0°C to 70°C)
		AT90S8515-4AI AT90S8515-4JI AT90S8515-4PI	44A 44J 40P6	Industrial (-40°C to 85°C)
8	4.0 - 6.0V	AT90S8515-8AC AT90S8515-8JC AT90S8515-8PC	44A 44J 40P6	Commercial (0°C to 70°C)
		AT90S8515-8AI AT90S8515-8JI AT90S8515-8PI	44A 44J 40P6	Industrial (-40°C to 85°C)

Note: Order AT90S8515A-XXX for devices with the FSTRT Fuse programmed.

Package Type	
44A	44-lead, Thin (1.0 mm) Plastic Gull Wing Quad Flat Package (TQFP)
44J	44-lead, Plastic J-leaded Chip Carrier (PLCC)
40P6	40-lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)