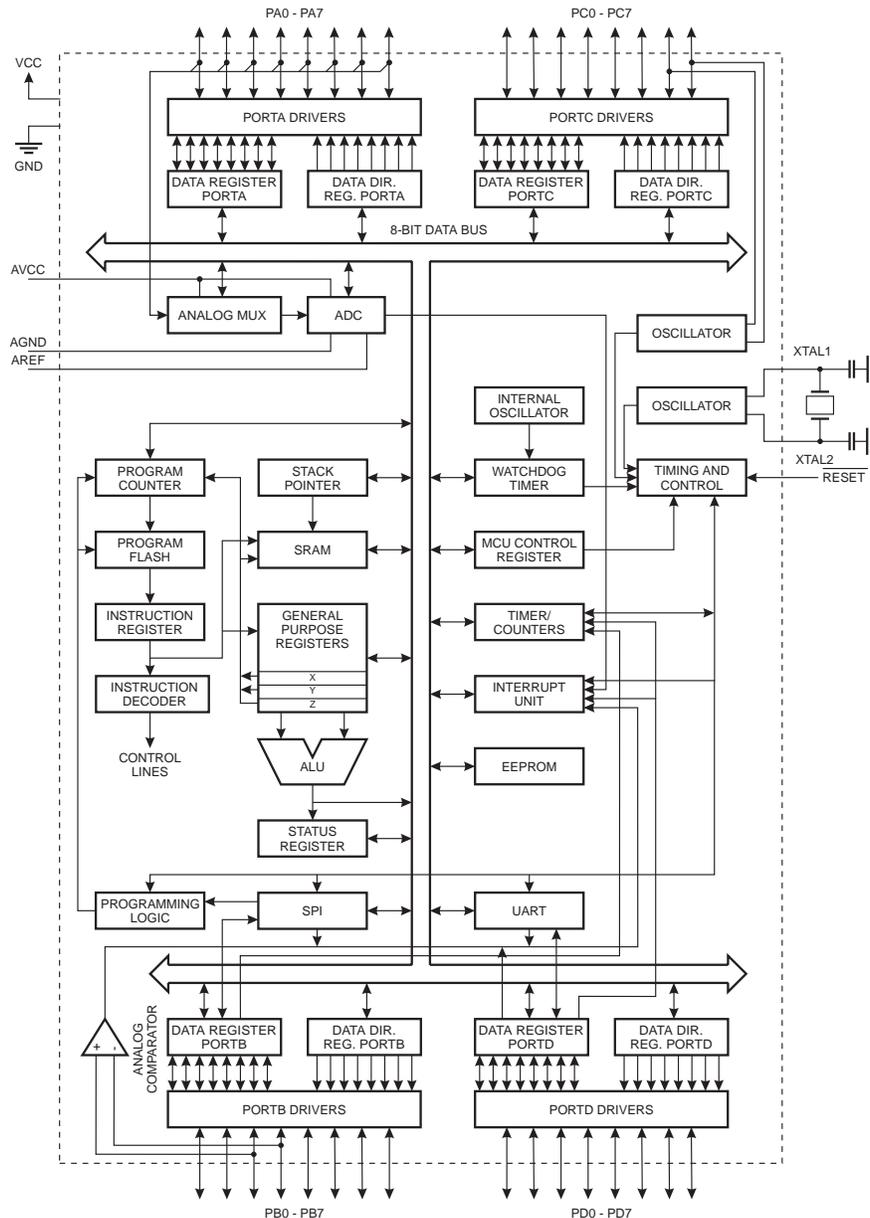




## Block Diagram



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

The AT90S4434/8535 provides the following features: 4K/8K bytes of In-System Programmable Flash, 256/512 bytes EEPROM, 256/512 bytes SRAM, 32 general purpose I/O lines, 32 general purpose working registers, RTC, three flexible timer/counters with compare modes, internal and

external interrupts, a programmable serial UART, 8-channel, 10-bit ADC, programmable Watchdog Timer with internal oscillator, an SPI serial port and three software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, timer/counters, SPI port and interrupt system to continue functioning. The Power Down mode saves the register contents but freezes the oscillator, disabling all other chip functions until the next interrupt or hardware reset. In Power Save mode, the timer oscillator continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping.

The device is manufactured using Atmel's high density non-volatile memory technology. The on-chip ISP Flash

allows the program memory to be reprogrammed in-system through an SPI serial interface or by a conventional nonvolatile memory programmer. By combining an 8-bit RISC CPU with In-System Programmable Flash on a monolithic chip, the Atmel AT90S4434/8535 is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The AT90S4434/8535 AVR is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, in-circuit emulators, and evaluation kits.

### Comparison between AT90S4434 and AT90S8535

The AT90S4434 has 4K bytes of Downloadable Flash, 256 bytes of EEPROM, and 256 bytes of internal SRAM.

The AT90S8535 has 8K bytes of Downloadable Flash, 512 bytes of EEPROM, and 512 bytes of internal SRAM.

Table 1 summarizes the different memory sizes for the two devices.

**Table 1.** Memory Size Summary

Part	Flash	EEPROM	SRAM
AT90S4434	4K bytes	256 bytes	256 bytes
AT90S8535	8K bytes	512 bytes	512 bytes

## Pin Descriptions

### VCC

Digital supply voltage

### GND

Digital ground

### Port A (PA7..PA0)

Port A is an 8-bit bi-directional I/O port. Port pins can provide internal pull-up resistors (selected for each bit). The Port A output buffers can sink 20mA and can drive LED displays directly. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated.

Port A also serves as the analog inputs to the A/D Converter.

### Port B (PB7..PB0)

Port B is an 8-bit bi-directional I/O pins with internal pull-up resistors. The Port B output buffers can sink 20 mA. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated.

Port B also serves the functions of various special features of the AT90S4434/8535 as listed on page 52.

### Port C (PC7..PC0)

Port C is an 8-bit bi-directional I/O port with internal pullup resistors. The Port C output buffers can sink 20 mA. As

inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. Two Port C pins can alternatively be used as oscillator for Timer/Counter2.

### Port D (PD7..PD0)

Port D is an 8-bit bidirectional I/O port with internal pull-up resistors. The Port D output buffers can sink 20 mA. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated.

Port D also serves the functions of various special features of the AT90S4434/8535 as listed on page 59.

### RESET

Reset input. A low on this pin for two machine cycles while the oscillator is running resets the device.

### XTAL1

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

### XTAL2

Output from the inverting oscillator amplifier

### AVCC

This is the supply voltage pin for the A/D Converter. It should be externally connected to VCC via a low-pass filter. See page 47 for details on operation of the ADC.

### AREF

This is the analog reference input for the A/D Converter. For ADC operations, a voltage in the range AGND to AV<sub>CC</sub> must be applied to this pin.

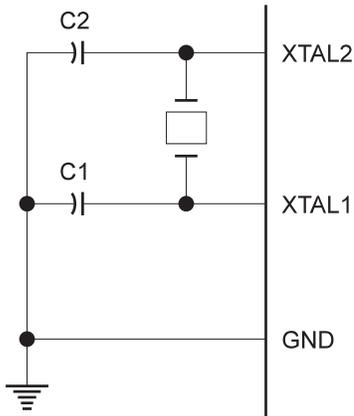
### AGND

Analog ground. If the board has a separate analog ground plane, this pin should be connected to this ground plane. Otherwise, connect to GND.

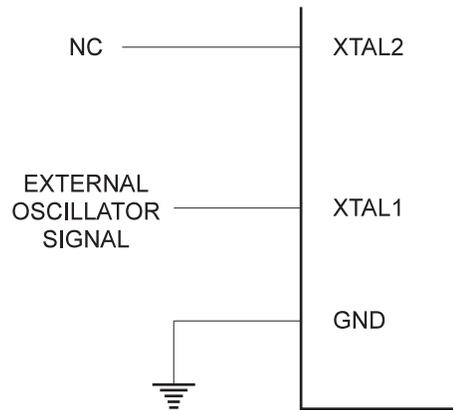
## Crystal Oscillators

XTAL1 and XTAL2 are input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or a ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 2. For the Timer Oscillator pins, PC6(OSC1) and PC7(OSC2), the crystal is connected directly between the pins. No external capacitors are needed. The oscillator is optimized for use with a 32,768 Hz watch crystal. An external clock signal applied to this pin goes through the same amplifier having a bandwidth of 256 kHz. The external clock signal should therefore be in the interval 0 Hz - 256 kHz.

**Figure 1. Oscillator Connections**



**Figure 2. External Clock Drive Configuration**

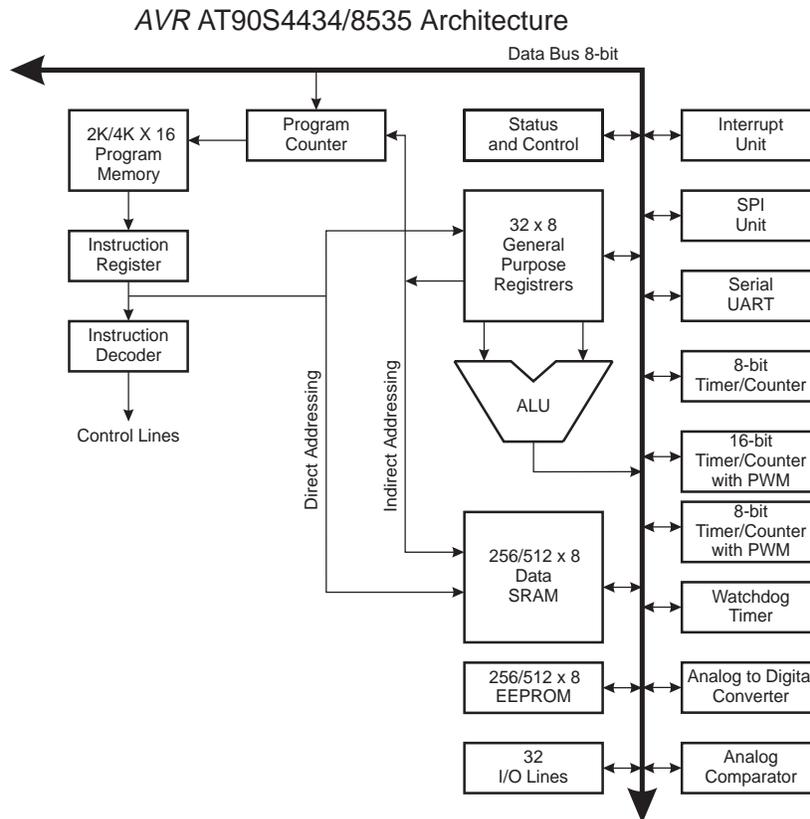


**Architectural Overview**

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-bits 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-bits X-register, Y-register and Z-register.

**Figure 3. The AT90S4434/8535 AVR Enhanced RISC Architecture**



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 3 shows the AT90S4434/8535 AVR Enhanced 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 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 single level pipelining. 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 downloadable Flash memory.

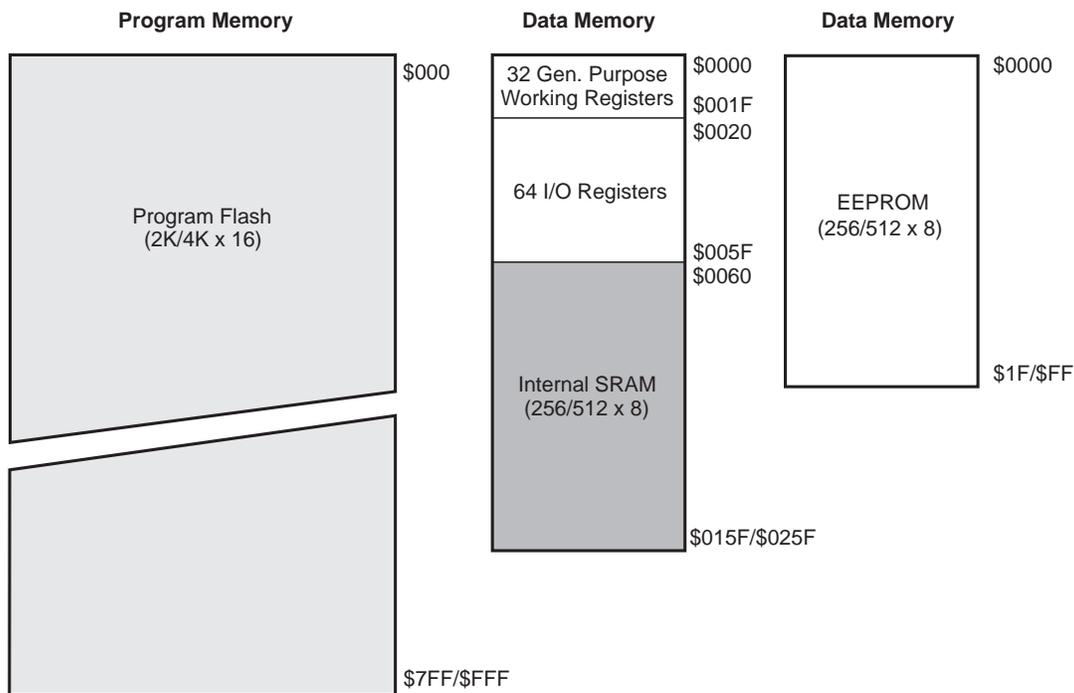
With the relative jump and call instructions, the whole 2K/4K address space is directly accessed. Most AVR instructions have a single 16-bit word format. Every program memory address contains a 16- or 32-bit instruction.

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 9-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. Memory Maps**



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.

## The General Purpose Register File

Figure 5 shows the structure of the 32 general purpose working registers in the CPU.

**Figure 5.** AVR CPU General Purpose Working Registers

	7	0	Addr.	
General Purpose Working Registers	R0		\$00	
	R1		\$01	
	R2		\$02	
	...			
	R13		\$0D	
	R14		\$0E	
	R15		\$0F	
	R16		\$10	
	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 5, 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.

### The 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 6.** The 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).

## The ALU - Arithmetic Logic Unit

The high-performance AVR ALU operates in direct connection with all the 32 general purpose working registers. Within a single clock cycle, ALU operations between registers in the register file are executed. The ALU operations are divided into three main categories - arithmetic, logical, and bit-functions.

## The In-System Programmable Flash Program Memory

The AT90S4434/8535 contains 4K/8K bytes on-chip In-System Programmable Flash memory for program storage. Since all instructions are 16-or 32-bit words, the Flash is organized as 2K/4K x 16. The Flash memory has an endurance of at least 1000 write/erase cycles. The

AT90S4434/8535 Program Counter (PC) is 11/12 bits wide, thus addressing the 2048/4096 program memory addresses.

See page 70 for a detailed description on Flash data downloading.

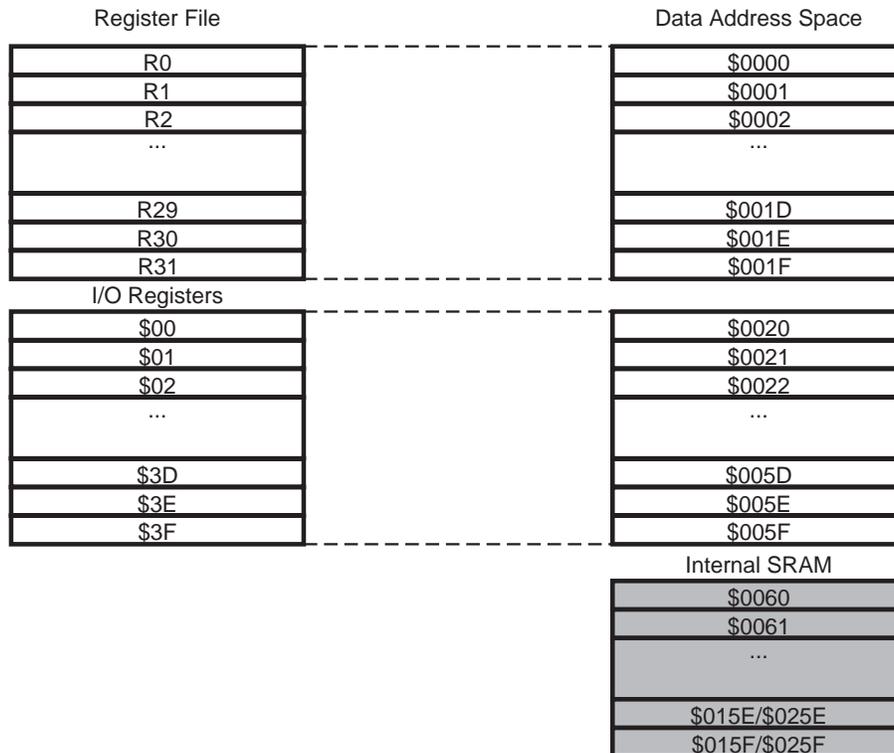
Constant tables must be allocated within the address 0-2K/4K(see the LPM - Load Program Memory instruction description).

See page 8 for the different program memory addressing modes.

## The SRAM Data Memory

The following figure shows how the AT90S4434/8535 SRAM Memory is organized:

**Figure 7.** SRAM Organization



The lower 352/608 Data Memory locations address the Register file, the I/O Memory, and the internal data SRAM. The first 96 locations address the Register File + I/O Memory, and the next 256/512 locations address the internal data SRAM.

The five different addressing modes for the data memory cover: Direct, Indirect with Displacement, Indirect, Indirect with Pre-Decrement and Indirect with Post-Increment. In the register file, registers R26 to R31 feature the indirect addressing pointer registers.

The direct addressing reaches the entire data space.

The Indirect with Displacement mode features a 63 address locations reach from the base address given by the Y or Z-register.

When using register indirect addressing modes with automatic pre-decrement and post-increment, the address registers X, Y and Z are decremented and incremented.

The 32 general purpose working registers, 64 I/O registers and the 256/512 bytes of internal data SRAM in the AT90S4434/8535 are all accessible through all these addressing modes.

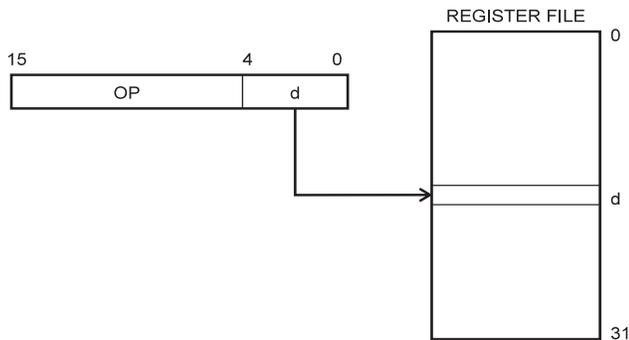
See the next section for a detailed description of the different addressing modes.

## The Program and Data Addressing Modes

The AT90S4434/8535 AVR Enhanced RISC microcontroller supports powerful and efficient addressing modes for access to the program memory (Flash) and data memory (SRAM, Register File and I/O Memory). This section describes the different addressing modes supported by the AVR architecture. In the figures, OP means the operation code part of the instruction word. To simplify, not all figures show the exact location of the addressing bits.

### Register Direct, Single Register Rd

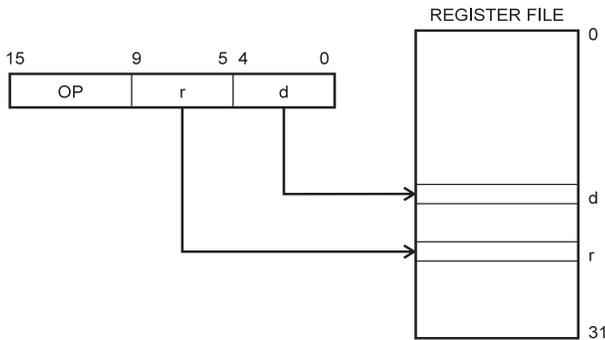
**Figure 8.** Direct Single Register Addressing



The operand is contained in register d (Rd).

### Register Direct, Two Registers Rd And Rr

**Figure 9.** Direct Register Addressing, Two Registers

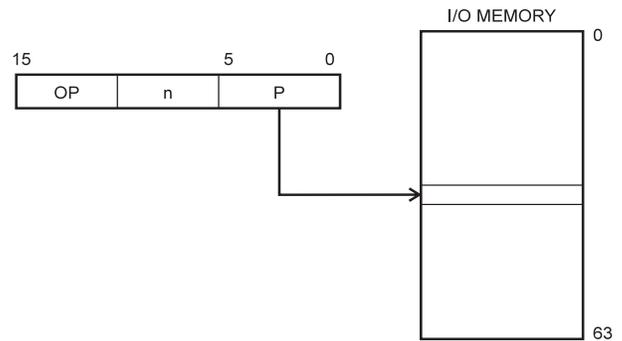


Operands are contained in register r (Rr) and d (Rd). The result is stored in register d (Rd).

### Data Indirect

### I/O Direct

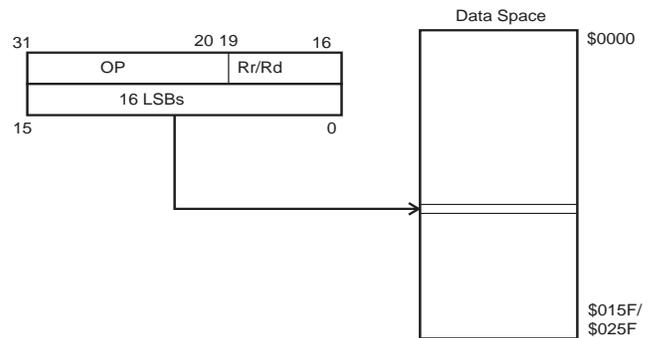
**Figure 10.** I/O Direct Addressing



Operand address is contained in 6 bits of the instruction word. n is the destination or source register address.

### Data Direct

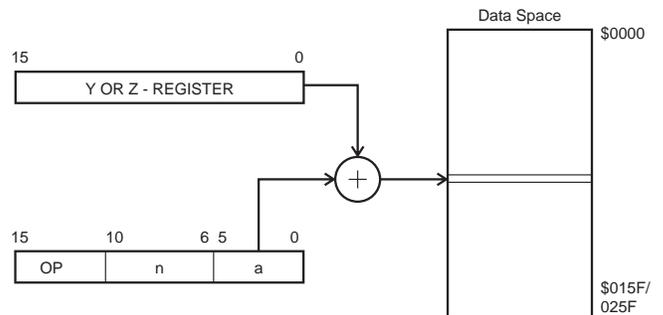
**Figure 11.** Direct Data Addressing



A 16-bit Data Address is contained in the 16 LSBs of a two-word instruction. Rd/Rr specify the destination or source register.

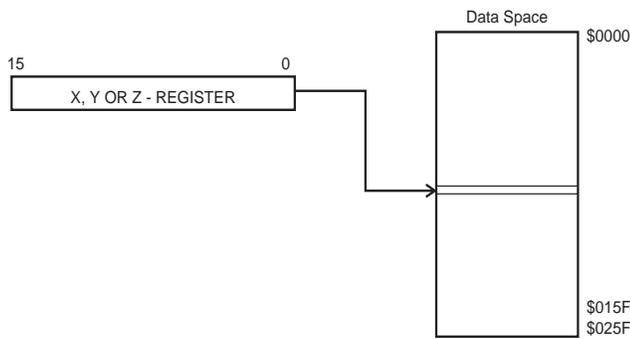
### Data Indirect With Displacement

**Figure 12.** Data Indirect with Displacement



Operand address is the result of the Y or Z-register contents added to the address contained in 6 bits of the instruction word.

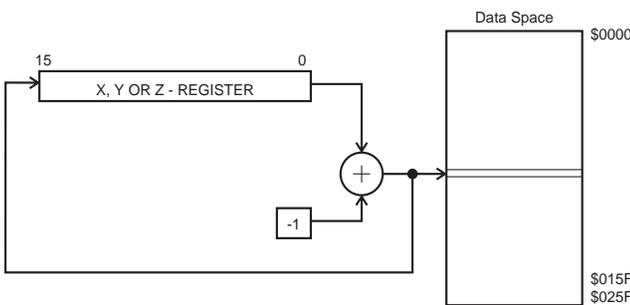
**Figure 13.** Data Indirect Addressing



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

**Data Indirect With Pre-Decrement**

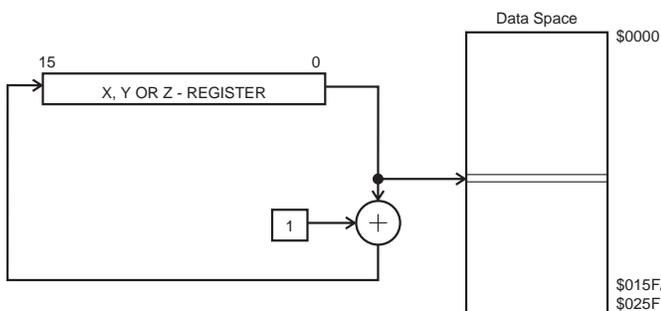
**Figure 14.** Data Indirect Addressing With Pre-Decrement



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.

**Data Indirect With Post-Increment**

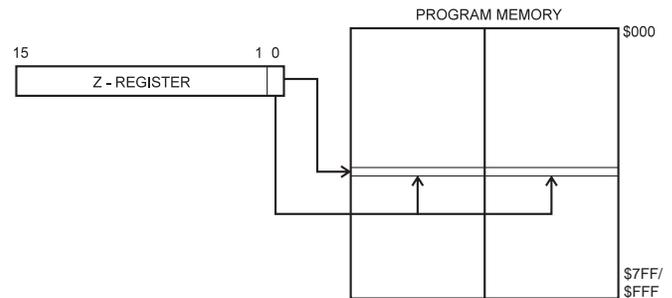
**Figure 15.** Data Indirect Addressing With Post-Increment



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**

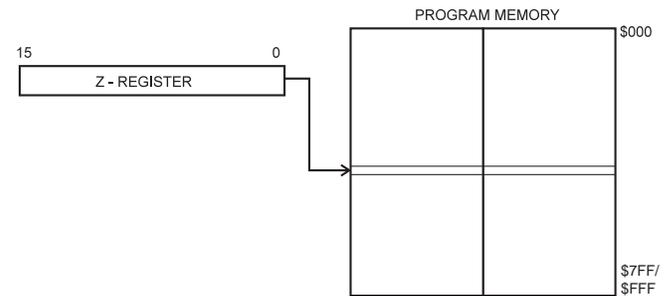
**Figure 16.** Code Memory Constant Addressing



Constant byte address is specified by the Z-register contents. The 15 MSBs select word address (0 - 2K/4K) and LSB, select low byte if cleared (LSB = 0) or high byte if set (LSB = 1).

**Indirect Program Addressing, IJMP And ICALL**

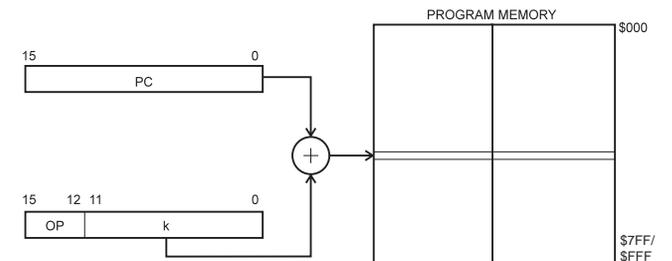
**Figure 17.** Indirect Program Memory Addressing



Program execution continues at address contained by the Z-register (i.e. the PC is loaded with the contents of the Z-register).

**Relative Program Addressing, RJMP And RCALL**

**Figure 18.** Relative Program Memory Addressing



Program execution continues at address PC + k + 1. The relative address k is from -2048 to 2047.

## The EEPROM Data Memory

The AT90S4434/8535 contains 256/512 bytes of data EEPROM memory. It is organized as a separate data space, in which single bytes can be read and written. The EEPROM has an endurance of at least 100,000 write/erase cycles. The access between the EEPROM and the CPU is described on page 36 specifying the EEPROM Address Registers, the EEPROM Data Register, and the EEPROM Control Register.

For the SPI data downloading, see page 70 for a detailed description.

## Memory Access Times and Instruction Execution Timing

This section describes the general access timing concepts for instruction execution and internal memory access.

The AVR CPU is driven by the System Clock  $\emptyset$ , directly generated from the external clock crystal for the chip. No internal clock division is used.

Figure 19 shows the parallel instruction fetches and instruction executions enabled by the Harvard architecture and the fast-access register file concept. This is the basic pipelining concept to obtain up to 1 MIPS per MHz with the corresponding unique results for functions per cost, functions per clocks, and functions per power-unit.

**Figure 19.** The Parallel Instruction Fetches and Instruction Executions

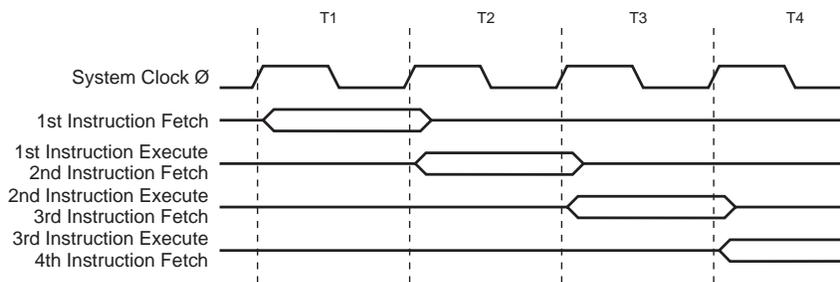
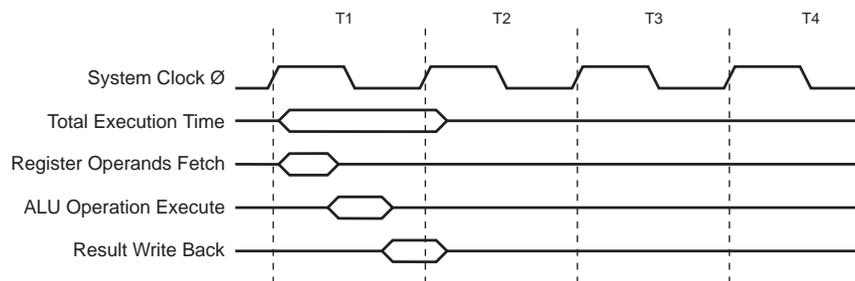


Figure 20 shows the internal timing concept for the register file. In a single clock cycle an ALU operation using two reg-

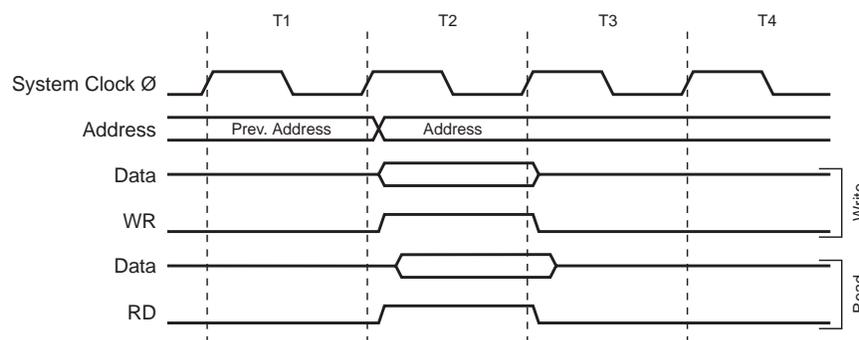
ister operands is executed, and the result is stored back to the destination register.

**Figure 20.** Single Cycle ALU Operation



The internal data SRAM access is performed in two System Clock cycles as described in Figure 21.

**Figure 21.** On-Chip Data SRAM Access Cycles



## I/O Memory

The I/O space definition of the AT90S4434/8535 is shown in the following table:

**Table 1.** AT90S4434/8535 I/O Space

I/O Address (SRAM Address)	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
\$34 (\$45)	MCUSR	MCU general Status 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
\$27 (\$47)	ICR1H	T/C 1 Input Capture Register High Byte
\$26 (\$46)	ICR1L	T/C 1 Input Capture Register Low Byte
\$25 (\$45)	TCCR2	Timer/Counter2 Control Register
\$24 (\$44)	TCNT2	Timer/Counter2 (8-bit)
\$23 (\$43)	OCR2	Timer/Counter2 Output Compare Register
\$22 (\$42)	ASSR	Asynchronous Mode Status Register
\$21 (\$41)	WDTCR	Watchdog Timer Control Register
\$1F (\$3E)	EEARH	EEPROM Address Register High Byte
\$1E (\$3E)	EEARL	EEPROM Address Register Low Byte
\$1D (\$3D)	EEDR	EEPROM Data Register
\$1C (\$3C)	EECR	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



**Table 1.** AT90S4434/8535 I/O Space (Continued)

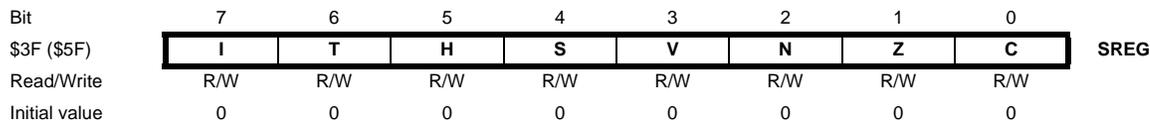
I/O Address (SRAM Address)	Name	Function
\$17 (\$37)	DDRB	Data Direction Register, Port B
\$16 (\$36)	PINB	Input Pins, Port B
\$15 (\$35)	PORTC	Data Register, Port C
\$14 (\$34)	DDRC	Data Direction Register, Port C
\$13 (\$33)	PINC	Input Pins, Port C
\$12 (\$32)	PORTD	Data Register, Port D
\$11 (\$31)	DDRD	Data Direction Register, Port D
\$10 (\$30)	PIND	Input Pins, Port D
\$0F (\$2F)	SPDR	SPI I/O Data Register
\$0E (\$2E)	SPSR	SPI Status Register
\$0D (\$2D)	SPCR	SPI Control Register
\$0C (\$2C)	UDR	UART I/O Data Register
\$0B (\$2B)	USR	UART Status Register
\$0A (\$2A)	UCR	UART Control Register
\$09 (\$29)	UBRR	UART Baud Rate Register
\$08 (\$28)	ACSR	Analog Comparator Control and Status Register
\$07 (\$27)	ADMUX	ADC Multiplexer Select Register
\$06 (\$26)	ADCSR	ADC Control and Status Register
\$05 (\$25)	ADCH	ADC Data Register High
\$04 (\$24)	ADCL	ADC Data Register Low

Note: Reserved and unused locations are not shown in the table.

All the different AT90S4434/8535 I/Os and peripherals are placed in the I/O space. The different I/O locations are accessed by the IN and OUT instructions transferring data between the 32 general purpose working registers and the I/O space. I/O registers within the address range \$00 - \$1F are directly bit-accessible using the SBI and CBI instructions. In these registers, the value of single bits can be checked by using the SBIS and SBIC instructions. Refer to the instruction set chapter for more details.

**The Status Register - SREG**

The AVR status register - SREG - at I/O space location \$3F (\$5F) is defined as:



• **Bit 7 - I: Global Interrupt Enable**

The global interrupt enable bit must be set (one) for the interrupts to be enabled. The individual interrupt enable control is then performed in the interrupt mask registers - GIMSK and TIMSK. If the global interrupt enable register is

When using the I/O specific commands, IN, OUT, SBIS and SBIC, the I/O addresses \$00 - \$3F must be used. When addressing I/O registers as SRAM, \$20 must be added to this address. All I/O register addresses throughout this document are shown with the SRAM address in parentheses.

The different I/O and peripherals control registers are explained in the following sections.

cleared (zero), none of the interrupts are enabled independent of the GIMSK and TIMSK values. The I-bit is cleared by hardware after an interrupt has occurred, and is set by the RETI instruction to enable subsequent interrupts.

- **Bit 6 - T: Bit Copy Storage**

The bit copy instructions BLD (Bit Load) and BST (Bit Store) use the T bit as source and destination for the operated bit. A bit from a register in the register file can be copied into T by the BST instruction, and a bit in T can be copied into a bit in a register in the register file by the BLD instruction.

- **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.

### The Stack Pointer - SP

The AT90S4434/8535 Stack Pointer is implemented as two 8-bit registers in the I/O space locations \$3E (\$5E) and \$3D (\$5D). As the AT90S4434/8535 data memory has \$25F locations, 10 bits are used.

Bit	15	14	13	12	11	10	9	8	
\$3E (\$5E)	-	-	-	-	-	-	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								
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 is decremented by one when data is pushed onto the Stack with the PUSH instruction, and it is decremented by two when data is pushed onto the Stack with subroutine CALL and interrupt. 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 data is popped from the Stack with return from subroutine RET or return from interrupt IRET.



## Reset and Interrupt Handling

The AT90S4434/8535 provides 16 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 2. 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 INTO - the External Interrupt Request 0, etc.

**Table 2.** Reset and Interrupt Vectors

Vector No.	Program Address	Source	Interrupt Definition
1	\$000	RESET	Hardware Pin and Watchdog Reset
2	\$001	INT0	External Interrupt Request 0
3	\$002	INT1	External Interrupt Request 1
4	\$003	TIMER2 COMP	Timer/Counter2 Compare Match
5	\$004	TIMER2 OVF	Timer/Counter2 Overflow
6	\$005	TIMER1 CAPT	Timer/Counter1 Capture Event
7	\$006	TIMER1 COMPA	Timer/Counter1 Compare Match A
8	\$007	TIMER1 COMPB	Timer/Counter1 Compare Match B
9	\$008	TIMER1 OVF	Timer/Counter1 Overflow
10	\$009	TIMER0 OVF	Timer/Counter0 Overflow
11	\$00A	SPI, STC	Serial Transfer Complete
12	\$00B	UART, RX	UART, Rx Complete
13	\$00C	UART, UDRE	UART Data Register Empty
14	\$00D	UART, TX	UART, Tx Complete
15	\$00E	ADC	ADC Conversion Complete
16	\$00F	EE_RDY	EEPROM Ready
17	\$010	ANA_COMP	Analog Comparator

The most typical and general program setup for the Reset and Interrupt Vector Addresses are:

Address	Labels	Code	Comments
\$000		rjmp RESET	; Reset Handler
\$001		rjmp EXT_INT0	; IRQ0 Handler
\$002		rjmp EXT_INT1	; IRQ1 Handler
\$003		rjmp TIM2_COMP	; Timer2 Compare Handler
\$004		rjmp TIM2_OVF	; Timer2 Overflow Handler
\$005		rjmp TIM1_CAPT	; Timer1 Capture Handler
\$006		rjmp TIM1_COMPA	; Timer1 CompareA Handler
\$007		rjmp TIM1_COMPB	; Timer1 CompareB Handler
\$008		rjmp TIM1_OVF	; Timer1 Overflow Handler
\$009		rjmp TIM0_OVF	; Timer0 Overflow Handler
\$00a		rjmp SPI_STC	; SPI Transfer Complete Handler
\$00b		rjmp UART_RXC	; UART RX Complete Handler
\$00c		rjmp UART_DRE	; UDR Empty Handler
\$00d		rjmp UART_TXC	; UART TX Complete Handler
\$00e		rjmp ADC	; ADC Conversion Complete Interrupt Handler
\$00f		rjmp EE_RDY	; EEPROM Ready Handler
\$010		rjmp ANA_COMP	; Analog Comparator Handler
;			
\$011	MAIN:	<instr> xxx	; Main program start
...	...	...	...

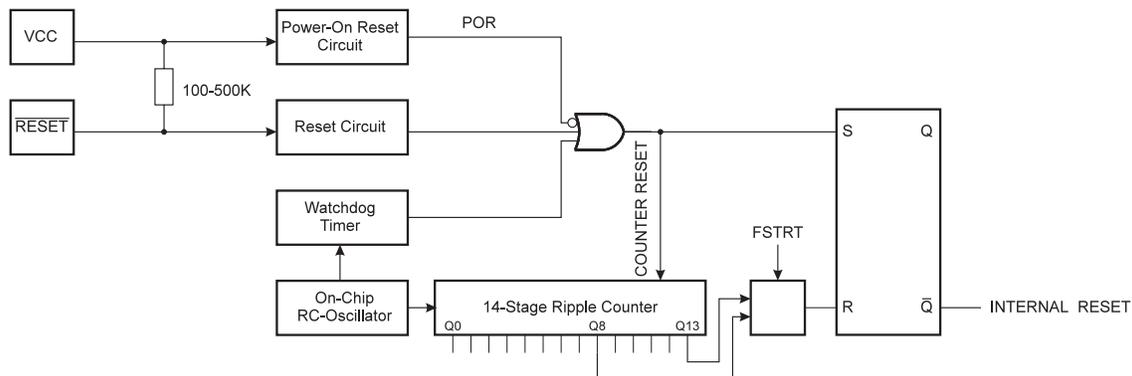
## Reset Sources

The AT90S4434/8535 has three sources of reset:

- Power-On Reset. The MCU is reset when a supply voltage is applied to the VCC and GND pins.
- External Reset. The MCU is reset when a low level is present on the RESET pin for more than two XTAL cycles
- 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 22 shows the reset logic. Table 3 defines the timing and electrical parameters of the reset circuitry.

**Figure 22.** Reset Logic



**Table 3.** Reset Characteristics ( $V_{CC} = 5.0V$ )

Symbol	Parameter	Min	Typ	Max	Units
$V_{POT}$	Power-On Reset Threshold Voltage	1.4	1.6	1.8	V
$V_{RST}$	$\overline{RESET}$ Pin Threshold Voltage		$0.6V_{CC}$		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

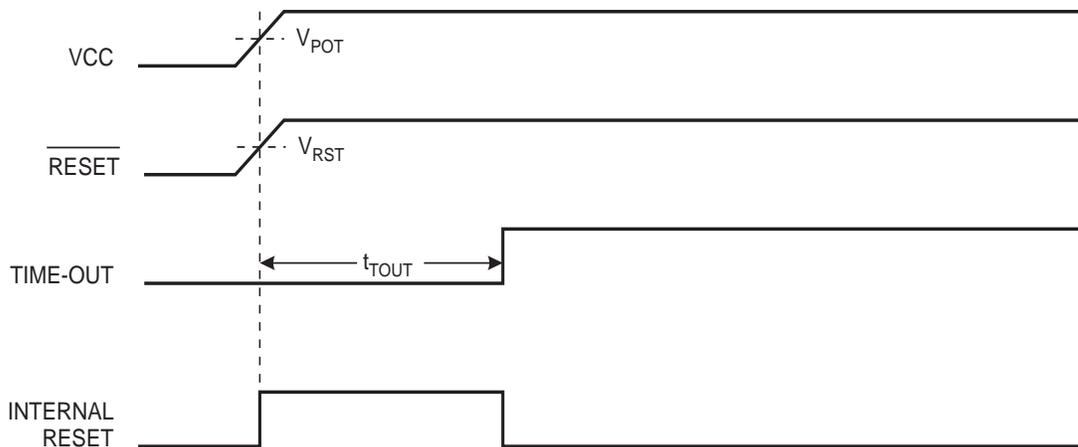
**Power-On Reset**

A Power-On Reset (POR) circuit ensures that the device is not started until  $V_{CC}$  has reached a safe level. As shown in Figure 22, 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 23 and Figure 24). The FSTRT fuse bit in the Flash can be programmed to give a shorter start-up time if a ceramic

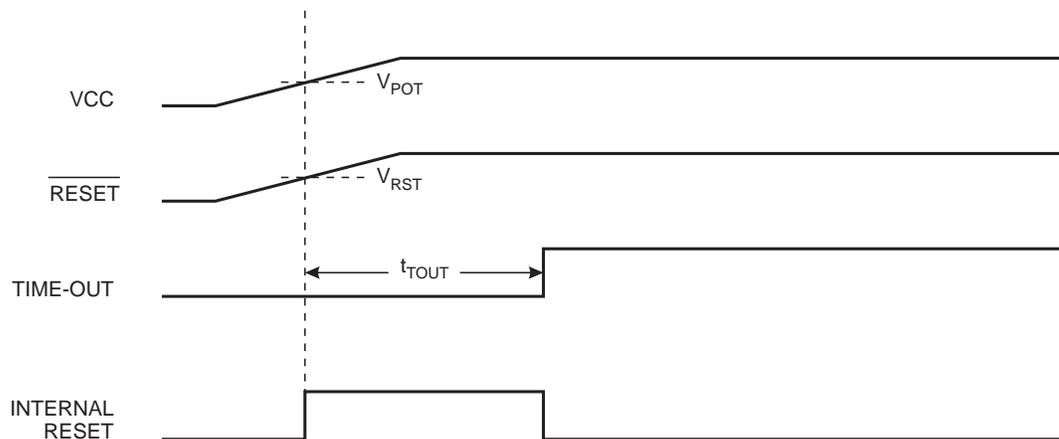
resonator or any other fast-start oscillator is used to clock the MCU.

If the built-in start-up delay is sufficient,  $\overline{RESET}$  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.

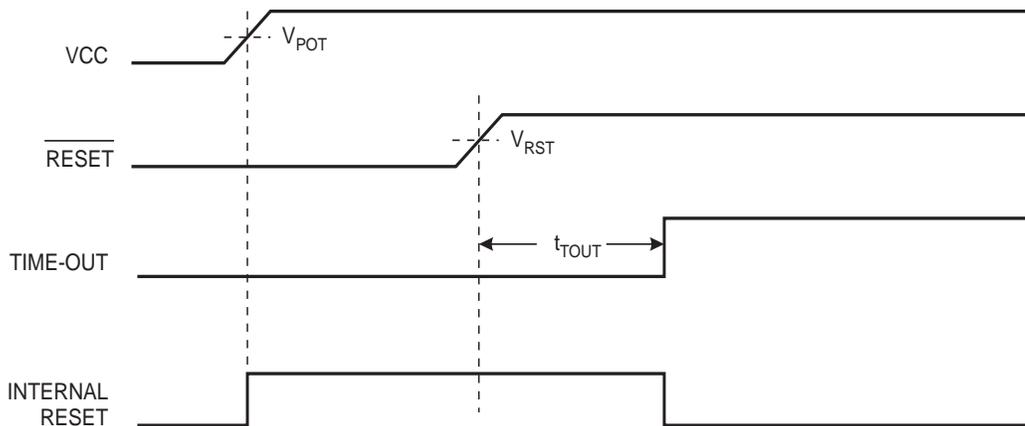
**Figure 23.** MCU Start-Up,  $\overline{RESET}$  Tied to VCC. Rapidly Rising  $V_{CC}$



**Figure 24.** MCU Start-Up,  $\overline{RESET}$  Tied to VCC. Slowly Rising  $V_{CC}$



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

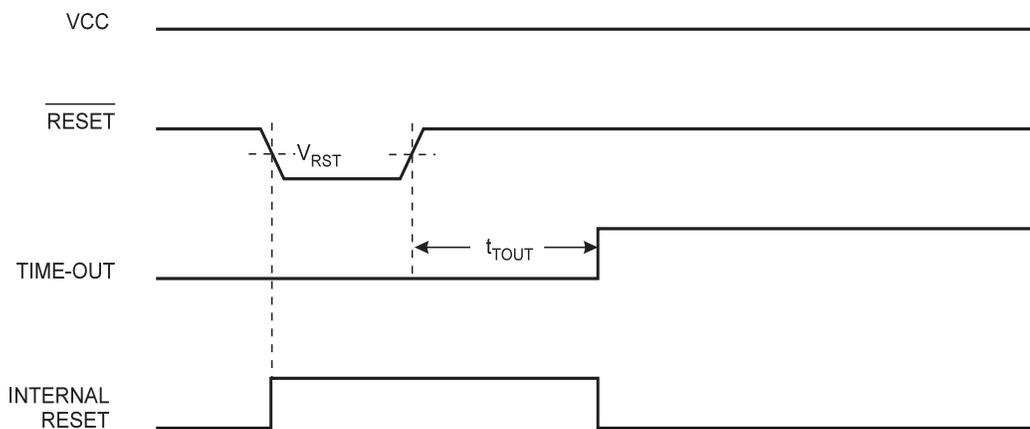


**External Reset**

An external reset is generated by a low level on the  $\overline{\text{RESET}}$  pin. The  $\overline{\text{RESET}}$  pin must be held low for at least two crystal clock cycles. 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  $t_{TOUIT}$  has expired.

**Figure 26.** External Reset During Operation

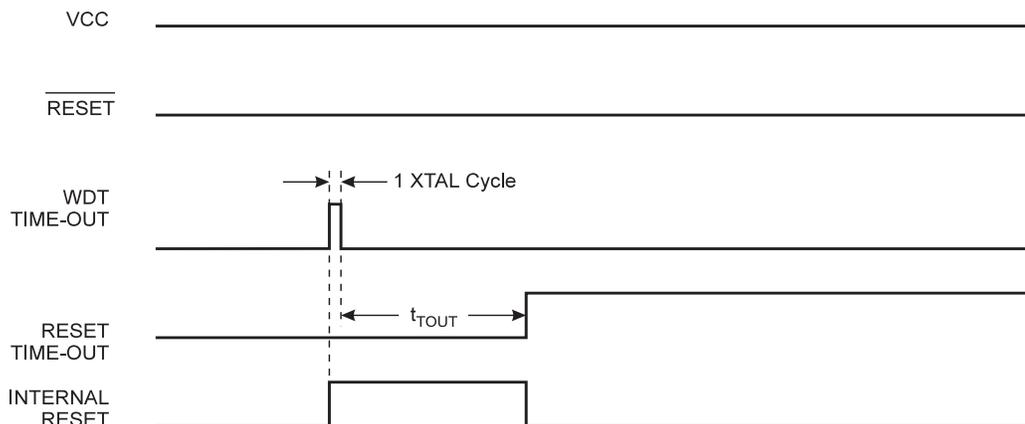


**Watchdog Reset**

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  $t_{TOUIT}$ . Refer to page 35 for details on operation of the Watchdog.

**Figure 27.** Watchdog Reset During Operation



## MCU Status Register - MCUSR

The MCU Status Register provides information on which reset source caused an MCU reset.

Bit	7	6	5	4	3	2	1	0	
\$34 (\$54)	-	-	-	-	-	-	EXTRF	PORF	MCUSR
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial value	0	0	0	0	0	0	See bit description		

- **Bits 7..2 - Res: Reserved Bits**

These bits are reserved bits in the AT90S4434/8535 and always read as zero.

- **Bit 1 - PORF: Power On Reset Flag**

This bit is only cleared by a power on reset. A watchdog reset or an external reset will leave this bit unchanged.

- **Bit 2 - EXTRF: External Reset Flag**

After a power on reset, this bit is undefined (X). It can only be cleared by an external reset. A watchdog reset will leave this bit unchanged.

To summarize, the following table shows the value of these two bits after the three modes of reset:

**Table 4.** PORF and EXTRF Values after Reset

Reset Source	PORF	EXTRF
Power On Reset	1	undefined
External Reset	unchanged	1
Watchdog Reset	unchanged	unchanged

To make use of these bits to identify a reset condition, the user software should set both the PORF and EXTRF bits as early as possible in the program. Checking the PORF and EXTRF values is done before the bits are set. If the bit becomes set before an external or watchdog reset occurs,

## The General Interrupt Mask Register - GIMSK

Bit	7	6	5	4	3	2	1	0	
\$3B (\$5B)	INT1	INT0	-	-	-	-	-	-	GIMSK
Read/Write	R/W	R/W	R	R	R	R	R	R	
Initial value	0	0	0	0	0	0	0	0	

- **Bit 7 - INT1: External Interrupt Request 1 Enable**

When the INT1 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), the external pin interrupt is activated. The Interrupt Sense Control1 bits 1/0 (ISC11 and ISC10) in the MCU general Control Register (MCUCR) define whether the external interrupt is activated on rising or falling edge of the INT1 pin or level sensed. Activity on the pin will cause an interrupt request even if INT1 is configured as an output. The corresponding interrupt of External Interrupt Request 1 is executed from program memory address \$002. See also "External Interrupts".

- **Bit 6 - INT0: External Interrupt Request 0 Enable**

When the INT0 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), the external pin interrupt is activated. The Interrupt Sense Control0 bits 1/0 (ISC01 and

the source of reset can be found by using the following truth table:

**Table 5.** Reset Source Identification

PORF	EXTRF	Reset Source
0	0	Watchdog Reset
0	1	External Reset
1	0	Power-On Reset
1	1	Power-On Reset

## Interrupt Handling

The AT90S4434/8535 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 must set (one) the I-bit to enable interrupts.

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 interrupt. Some of the interrupt flags can also be cleared by writing a logic one to the flag bit position(s) to be cleared.

ISC00) in the MCU general Control Register (MCUCR) define whether the external interrupt is activated on rising or falling edge of the INT0 pin or level sensed. Activity on the pin will cause an interrupt request even if INT0 is configured as an output. The corresponding interrupt of External Interrupt Request 0 is executed from program memory address \$001. See also "External Interrupts."

- **Bits 5.0 - Res: Reserved bits**

These bits are reserved bits in the AT90S4434/8535 and always read as zero.

## The General Interrupt Flag Register - GIFR

Bit	7	6	5	4	3	2	1	0	
\$3A (\$5A)	INTF1	INTF0	-	-	-	-	-	-	GIFR
Read/Write	R/W	R/W	R	R	R	R	R	R	
Initial value	0	0	0	0	0	0	0	0	

- **Bit 7 - INTF1: External Interrupt Flag1**

When an event on the INT1 pin triggers an interrupt request, INTF1 becomes set (one). If the I-bit in SREG and the INT1 bit in GIMSK are set (one), the MCU will jump to the interrupt vector at address \$002. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it.

- **Bit 6 - INTF0: External Interrupt Flag0**

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 AT90S4434/8535 and always read as zero.

## The Timer/Counter Interrupt Mask Register - TIMSK

Bit	7	6	5	4	3	2	1	0	
\$39 (\$59)	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	-	TOIE0	TIMSK
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Initial value	0	0	0	0	0	0	0	0	

- **Bit 7 - OCIE2: Timer/Counter2 Output Compare Match Interrupt Enable**

When the OCIE1A bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter2 Compare Match interrupt is enabled. The corresponding interrupt (at vector \$003) is executed if a compare match in Timer/Counter2 occurs. The compare flag in Timer/Counter2 is set (one) in the Timer/Counter Interrupt Flag Register - TIFR.

- **Bit 6 - TOIE2: Timer/Counter2 Overflow Interrupt Enable**

When the TOIE2 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter2 Overflow interrupt is enabled. The corresponding interrupt (at vector \$004) is executed if an overflow in Timer/Counter2 occurs. The Timer/Counter2 Overflow Flag is set (one) in the Timer/Counter Interrupt Flag Register - TIFR. When Timer/Counter1 is in PWM mode, the Timer Overflow flag is set when the counter changes counting direction at \$00.

- **Bit 5 - 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 \$005) is executed if a capture-triggering event occurs on pin 20, PD6 (ICP). The Input Capture Flag in Timer/Counter1 is set (one) in the Timer/Counter Interrupt Flag Register - TIFR.

- **Bit 4 - 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 \$006) is executed if a CompareA match in Timer/Counter1

occurs. The CompareA Flag in Timer/Counter1 is set (one) in the Timer/Counter Interrupt Flag Register - TIFR.

- **Bit 3 - 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 \$007) is executed if a CompareB match in Timer/Counter1 occurs. The CompareB Flag in Timer/Counter1 is set (one) in the Timer/Counter Interrupt Flag Register - TIFR.

- **Bit 2 - 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 \$008) is executed if an overflow in Timer/Counter1 occurs. The Overflow Flag (Timer/Counter1) is set (one) in the Timer/Counter Interrupt Flag Register - TIFR. When Timer/Counter1 is in PWM mode, the Timer Overflow flag is set when the counter changes counting direction at \$0000.

- **Bit 1 - Res: Reserved bit**

This bit is a reserved bit in the AT90S4434/8535 and always reads zero.

- **Bit 0 - 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 \$009) is executed if an overflow in Timer/Counter0 occurs. The Overflow Flag (Timer0) is set (one) in the Timer/Counter Interrupt Flag Register - TIFR.

## The Timer/Counter Interrupt Flag Register - TIFR

Bit	7	6	5	4	3	2	1	0	
\$38 (\$58)	<b>TIFR</b>								
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Initial value	0	0	0	0	0	0	0	0	

- **Bit 7 - OCF2: Output Compare Flag 2**

The OCF2 bit is set (one) when compare match occurs between the Timer/Counter2 and the data in OCR2 - Output Compare Register2. OCF2 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, OCF2 is cleared by writing a logic one to the flag. When the I-bit in SREG, and OCIE2 (Timer/Counter2 Compare match Interrupt Enable), and the OCF2 are set (one), the Timer/Counter2 Compare match Interrupt is executed.

- **Bit 6 - TOV2: Timer/Counter0 Overflow Flag**

The TOV2 bit is set (one) when an overflow occurs in Timer/Counter2. TOV2 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, TOV2 is cleared by writing a logic one to the flag. When the SREG I-bit, and TOIE2 (Timer/Counter2 Overflow Interrupt Enable), and TOV2 are set (one), the Timer/Counter2 Overflow interrupt is executed. In PWM mode, this bit is set when Timer/Counter2 changes counting direction at \$00.

- **Bit 5 - ICF1: Input Capture Flag 1**

The ICF1 bit is set (one) to flag an input capture event, indicating that the Timer/Counter1 value has been transferred to the input capture register - ICR1. ICF1 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, ICF1 is cleared by writing a logic one to the flag.

- **Bit 4 - OCF1A: Output Compare Flag 1A**

The OCF1A bit is set (one) when compare match occurs between the Timer/Counter1 and the data in OCR1A - Output Compare Register 1A. OCF1A is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, OCF1A is cleared by writing a logic one to the flag. When the I-bit in SREG, and OCIE1A (Timer/Counter1 Compare match InterruptA Enable), and the OCF1A are set (one), the Timer/Counter1A Compare match Interrupt is executed.

- **Bit 3 - OCF1B: Output Compare Flag 1B**

The OCF1B bit is set (one) when compare match occurs between the Timer/Counter1 and the data in OCR1B - Output Compare Register 1B. OCF1B is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, OCF1B is cleared by writing a logic one to the flag. When the I-bit in SREG, and OCIE1B (Timer/Counter1 Compare match InterruptB Enable), and the OCF1B are set (one), the Timer/Counter1B Compare match Interrupt is executed.

- **Bit 2 - 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.

- **Bit 1 - Res: Reserved bit**

This bit is a reserved bit in the AT90S4434/8535 and always reads zero.

- **Bit 0 - TOV0: Timer/Counter0 Overflow Flag**

The bit TOV0 is set (one) when an overflow occurs in Timer/Counter0. TOV0 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, TOV0 is cleared by writing a logic one to the flag. When the SREG I-bit, and TOIE0 (Timer/Counter0 Overflow Interrupt Enable), and TOV0 are set (one), the Timer/Counter0 Overflow interrupt is executed.

## External Interrupts

The external interrupts are triggered by the INT1 and INTO pins. Observe that, if enabled, the interrupts will trigger even if the INTO/INT1 pins are configured as outputs. This feature provides a way of generating a software interrupt. The external interrupts can be triggered by a falling or rising edge or a low level. This is set up as indicated in the specification for the MCU Control Register - MCUCR. When the external interrupt is enabled and is configured as level triggered, the interrupt will trigger as long as the pin is held low.

The external interrupts are set up as described in the specification for the MCU Control Register - MCUCR.

## Interrupt Response Time

The interrupt execution response for all the enabled AVR interrupts is 4 clock cycles minimum. 4 clock cycles after the interrupt flag has been set, the program vector address for the actual interrupt handling routine is executed. During this 4 clock cycle period, the Program Counter (2 bytes) is pushed onto the Stack, and the Stack Pointer is decremented by 2. The vector is a relative jump to the interrupt

## MCU Control Register - MCUCR

The MCU Control Register contains control bits for general MCU functions.

Bit	7	6	5	4	3	2	1	0	
\$35 (\$55)	-	SE	SM1	SM0	ISC11	ISC10	ISC01	ISC00	MCUCR
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

- Bit 7 - Res: Reserved bit**

This bit is a reserved bit in the AT90S4434/8535 and always reads zero.

- Bit 6 - SE: Sleep Enable**

The SE bit must be set (one) to make the MCU enter the sleep mode when the SLEEP instruction is executed. To avoid the MCU entering the sleep mode unless it is the programmers purpose, it is recommended to set the Sleep Enable SE bit just before the execution of the SLEEP instruction.

- Bits 5,4 - SM1/SM0: Sleep Mode Select bits 1 and 0**

This bit selects between the three available sleep modes as shown in the following table.

**Table 6.** Sleep Mode Select

SM1	SM0	Sleep Mode
0	0	Idle Mode
0	1	Reserved
1	0	Power Down
1	1	Power Save

- Bit 1, 0 - ISC01, ISC00: Interrupt Sense Control 0 bit 1 and bit 0**

routine, and this jump takes 2 clock cycles. If an interrupt occurs during execution of a multi-cycle instruction, this instruction is completed before the interrupt is served.

A return from an interrupt handling routine (same as for a subroutine call routine) takes 4 clock cycles. During these 4 clock cycles, the Program Counter (2 bytes) is popped back from the Stack, and the Stack Pointer is incremented by 2. When the AVR exits from an interrupt, it will always return to the main program and execute one more instruction before any pending interrupt is served.

Note that the Status Register - SREG - is not handled by the AVR hardware, neither for interrupts nor for subroutines. For the interrupt handling routines requiring a storage of the SREG, this must be performed by user software.

For Interrupts triggered by events that can remain static (E.g. the Output Compare Register1 A 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.

- Bits 3, 2 - ISC11, ISC10: Interrupt Sense Control 1 bit 1 and bit 0**

The External Interrupt 1 is activated by the external pin INT1 if the SREG I-flag and the corresponding interrupt mask in the GIMSK is set. The level and edges on the external INT1 pin that activate the interrupt are defined in the following table:

**Table 7.** Interrupt 1 Sense Control

ISC11	ISC10	Description
0	0	The low level of INT1 generates an interrupt request.
0	1	Reserved
1	0	The falling edge of INT1 generates an interrupt request.
1	1	The rising edge of INT1 generates an interrupt request.

**Note:** When changing the ISC11/ISC10 bits, INT1 must be disabled by clearing its Interrupt Enable bit in the GIMSK Register. Otherwise an interrupt can occur when the bits are changed.

The External Interrupt 0 is activated by the external pin INTO if the SREG I-flag and the corresponding interrupt

mask is set. The level and edges on the external INT0 pin that activate the interrupt are defined in the following table:

**Table 8.** Interrupt 0 Sense Control

ISC01	ISC00	Description
0	0	The low level of INT0 generates an interrupt request.
0	1	Reserved
1	0	The falling edge of INT0 generates an interrupt request.
1	1	The rising edge of INT0 generates an interrupt request.

Note: When changing the ISC10/ISC00 bits, INT0 must be disabled by clearing its Interrupt Enable bit in the GIMSK Register. Otherwise an interrupt can occur when the bits are changed.

### Sleep Modes

To enter any of the three sleep modes, the SE bit in MCUCR must be set (one) and a SLEEP instruction must be executed. The SM1 and SM0 bits in the MCUCR register select which sleep mode (Idle, Power Down, or Power Save) will be activated by the SLEEP instruction.

If an enabled interrupt occurs while the MCU is in a sleep mode, the MCU awakes, executes the interrupt routine, and resumes execution from the instruction following SLEEP. If a reset occurs during sleep mode, the MCU wakes up and executes from the Reset vector. The contents of the register file, SRAM, and I/O memory are unaltered when the device wakes up from sleep. See Table 6 on how to select sleep mode.

Note that if a level triggered interrupt is used for wake-up from power down or power save, the low level must be held for a time longer than the reset delay time-out period  $t_{\text{TOUR}}$ . Otherwise, the device will not wake up.

### Idle Mode

When the SM1/SM0 bits are set to 00, the SLEEP instruction forces the MCU into the Idle Mode stopping the CPU but allowing Timer/Counters, Watchdog and the interrupt system to continue operating. This enables the MCU to wake up from external triggered interrupts as well as internal ones like the Timer Overflow and UART Receive Complete interrupts. If wake-up from the Analog Comparator interrupt is not required, the analog comparator can be powered down by setting the ACD-bit in the Analog Comparator Control and Status register - ACSR. This will reduce power consumption in Idle Mode. When the MCU wakes up from Idle mode, the CPU starts program execution immediately.

### Power Down Mode

When the SM1/SM0 bits are 10, the SLEEP instruction forces the MCU into the Power Down Mode. In this mode,

the external oscillator is stopped. The user can select whether the watchdog shall be enabled during power-down mode. If the watchdog is enabled, it will wake up the MCU when the Watchdog Time-out period expires. If the watchdog is disabled, only an external reset or an external level triggered interrupt can wake up the MCU.

### Power Save Mode

When the SM1/SM0 bits are 11, the SLEEP instruction forces the MCU into the Power Save Mode. This mode is identical to Power Down, with one exception:

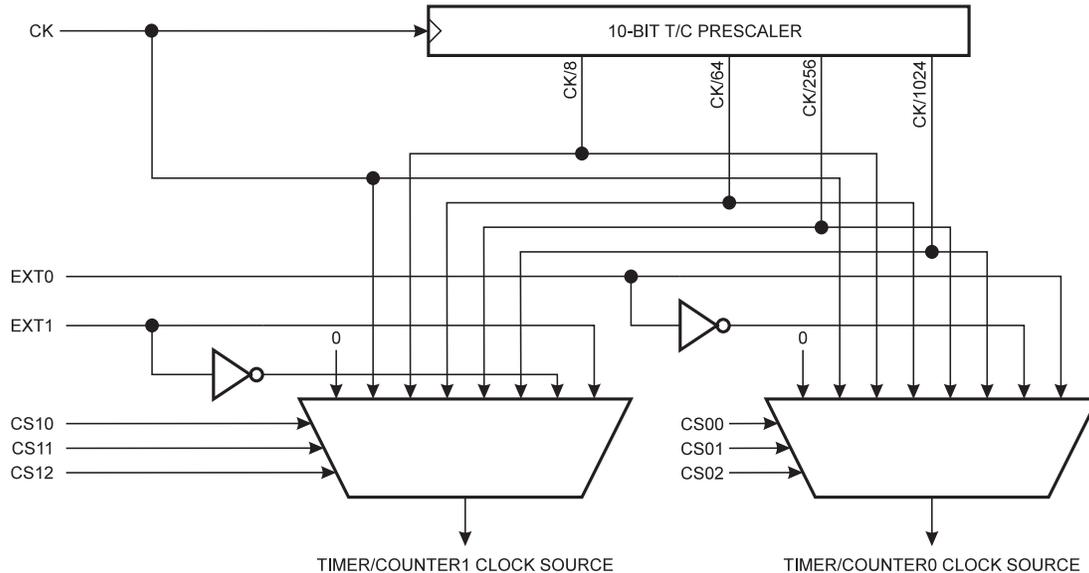
If Timer/Counter2 is clocked asynchronously, i.e. the AS2 bit in ASSR is set, Timer/Counter2 will run during sleep. The device can wake up from either Timer Overflow or Output Compare interrupt from Timer/Counter2.

## Timer / Counters

The AT90S4434/8535 provides three general purpose Timer/Counters - two 8-bit T/Cs and one 16-bit T/C. Timer/Counter2 can as an option be asynchronously clocked from an external oscillator. This oscillator is optimized for use with a 32.768 kHz watch crystal, enabling use of Timer/Counter2 as a Real Time Clock (RTC). Timer/Counters 0 and 1 have individual prescaling selection from the same 10-bit prescaling timer. Timer/Counter2 has its own prescaler. These Timer/Counters can either be used as a timer with an internal clock timebase or as a counter with an external pin connection which triggers the counting.

## The Timer/Counter Prescalers

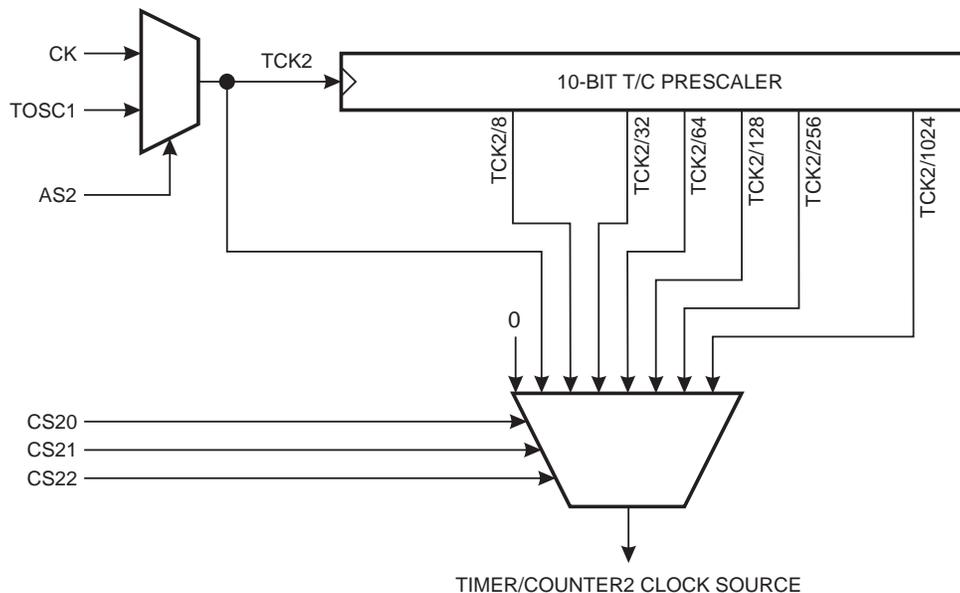
**Figure 28.** Prescaler for Timer/Counter0 and 1



For Timer/Counters 0 and 1, 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 0 and

1, CK, external source, and stop, can also be selected as clock sources.

**Figure 29.** Timer/Counter2 Prescaler



The clock source for Timer/Counter2 is named TCK2. TCK2 is by default connected to the main system clock CK. By setting the AS2 bit in ASSR, Timer/Counter2 is asynchronously clocked from the PC6(TOSC1) pin. This enables use of Timer/Counter2 as a Real Time Clock (RTC). When AS2 is set, pins PC6(TOSC1) and PC7(TOSC2) are disconnected from Port C and connected to a crystal oscillator. A crystal can then be connected

between the PC6(TOSC1) and PC7(TOSC2) pins to serve as an independent clock source for Timer/Counter2. The oscillator is optimized for use with a 32.768 kHz crystal. Alternatively, an external clock signal can be applied to PC6(TOSC1). The frequency of this clock must be lower than one fourth of the CPU clock and not higher than 256 kHz.

## The 8-Bit Timer/Counter0

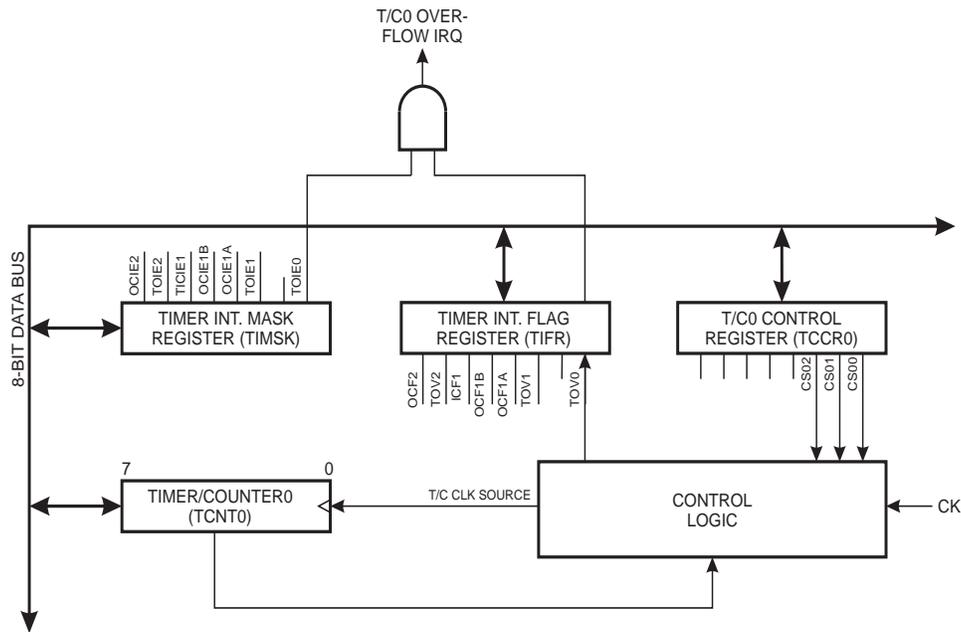
Figure 30 shows the block diagram for Timer/Counter0.

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.

**Figure 30.** Timer/Counter0 Block Diagram



### The 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 AT90S4434/8535 and always read 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 Timer0.

**Table 9.** Clock 0 Prescale Select

CS02	CS01	CS00	Description
0	0	0	Stop, 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 Stop condition provides a Timer Enable/Disable function. The prescaled CK modes are scaled directly from the CK oscillator clock. If the external pin modes are used, the

corresponding setup must be performed in the actual data direction control register (cleared to zero gives an input pin).

### The Timer Counter 0 - TCNT0

Bit	7	6	5	4	3	2	1	0	
\$32 (\$52)	<b>MSB</b>							<b>LSB</b>	TCNT0
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	

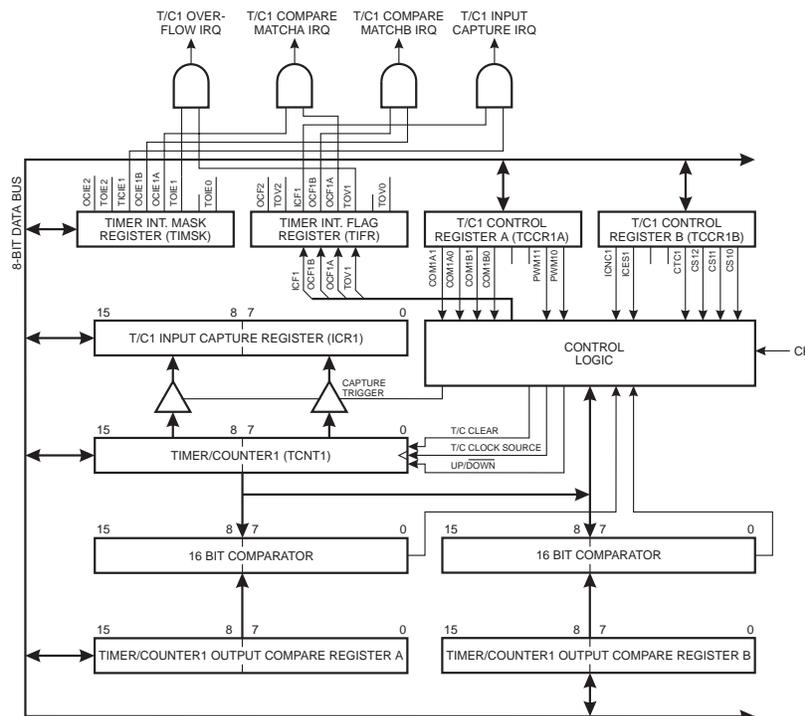
The Timer/Counter0 is realized as an up-counter with read and write access. If the Timer/Counter0 is written and a

clock source is present, the Timer/Counter0 continues counting in the clock cycle following the write operation.

### The 16-Bit Timer/Counter1

Figure 31 shows the block diagram for Timer/Counter1.

**Figure 31.** Timer/Counter1 Block Diagram



The 16-bit Timer/Counter1 can select clock source from CK, prescaled CK, or an external pin. In addition it can be stopped as described in the specification for the Timer/Counter1 Control Registers - TCCR1A and TCCR1B. The different status flags (overflow, compare match and capture event) and 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.

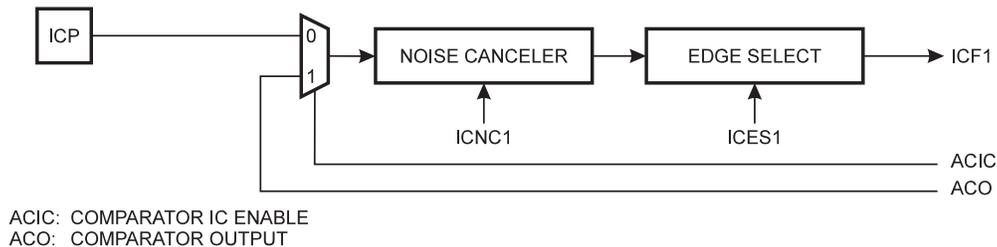
The 16-bit Timer/Counter1 features both a high resolution and a high accuracy usage with the lower prescaling opportunities. Similarly, the high prescaling opportunities makes the Timer/Counter1 useful for lower speed functions or exact timing functions with infrequent actions.

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 29 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 32.

**Figure 32.** ICP Pin Schematic Diagram



If the noise canceler function is enabled, the actual trigger condition for the capture event is monitored over 4 samples

before the capture is activated. The input pin signal is sampled at XTAL clock frequency.

**The 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. Since this is an alternative function to an I/O port, the corresponding direction control bit must be set (one) to control an output pin. The control configuration is shown in Table 10.

- **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. Since this is an alternative function to an I/O port, the corresponding direction control bit must be set (one) to control an output pin. The following control configuration is given:

**Table 10.** Compare 1 Mode Select

COM1X1	COM1X0	Description
0	0	Timer/Counter1 disconnected from output pin OC1X
0	1	Toggle the OC1X output line.
1	0	Clear the OC1X output line (to zero).
1	1	Set the OC1X output line (to one).

Note: X = A or B.

In PWM mode, these bits have a different function. Refer to Table 14 for a detailed description. When changing the COM1X1/COM1X0 bits, Output Compare Interrupts 1 must be disabled by clearing their Interrupt Enable bits in the TIMSK Register. Otherwise an interrupt can occur when the bits are changed.

### The 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 measures on the ICP - input capture pin, 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 Input Capture Register - ICR1 - on the rising edge of the input capture pin - ICP.

• **Bits 5, 4 - Res: Reserved bits**

These bits are reserved bits in the AT90S4434/8535 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:

• **Bits 3..2 - Res: Reserved bits**

These bits are reserved bits in the AT90S4434/8535 and always read zero.

• **Bits 1..0 - PWM11, PWM10: Pulse Width Modulator Select Bits**

These bits select PWM operation of Timer/Counter1 as specified in Table 11. This mode is described on page 29.

**Table 11.** PWM Mode Select

PWM11	PWM10	Description
0	0	PWM operation of Timer/Counter1 is disabled
0	1	Timer/Counter1 is an 8-bit PWM
1	0	Timer/Counter1 is a 9-bit PWM
1	1	Timer/Counter1 is a 10-bit PWM

... | C-1 | C | C+1 | 0 | 1 | ...

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

... | C-1, C-1, C-1, C-1, C-1, C-1, C-1, C-1 | C, C, C, C, C, C, C, C | C+1, 0, 0, 0, 0, 0, 0, 0, 0 | 1, 1, 1, 1, 1, 1, 1, 1 | ...

In PWM mode, this bit has no effect.

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

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

**Table 12.** 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 Stop condition provides a Timer Enable/Disable function. The CK down divided modes are scaled directly from the CK oscillator clock. If the external pin modes are used, the corresponding setup must be performed in the actual direction control register (cleared to zero gives an input pin).



## The Timer/Counter1 - TCNT1H AND TCNT1L

Bit	15	14	13	12	11	10	9	8	
\$2D (\$4D)	<b>MSB</b>								
\$2C (\$4C)								<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	

TCNT1H  
TCNT1L

This 16-bit register contains the prescaled value of the 16-bit Timer/Counter1. To ensure that both the high and low bytes are read and written simultaneously when the CPU accesses these registers, the access is performed using an 8-bit temporary register (TEMP). This temporary register is also used when accessing OCR1A, OCR1B 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.

### TCNT1 Timer/Counter1 Write:

When the CPU writes to the high byte TCNT1H, the written data is placed in the TEMP register. Next, when the CPU writes the low byte TCNT1L, this byte of data is combined with the byte data in the TEMP register, and all 16 bits are written to the TCNT1 Timer/Counter1 register simultaneously. Conse-

quently, the high byte TCNT1H must be accessed first for a full 16-bit register write operation.

### • 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>								
\$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	

OCR1AH  
OCR1AL

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

Bit	15	14	13	12	11	10	9	8	
\$29 (\$49)	<b>MSB</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	

OCR1BH  
OCR1BL

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. Writing to PORTD5 and PORTD4 sets the OC1A and OC1B values correspondingly.

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.

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

Bit	15	14	13	12	11	10	9	8		
\$27 (\$47)	<b>MSB</b>									ICR1H
\$26 (\$46)								<b>LSB</b>	ICR1L	
Read/Write	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.

### 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 PD4(OC1B) pins. Timer/Counter1 acts as an up/down counter, counting up from \$0000 to TOP (see Table 13), when 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)/PD4(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 14 for details.

**Table 13.** Timer TOP Values and PWM Frequency

PWM Resolution	Timer TOP value	Frequency
8-bit	\$00FF (255)	$f_{TC1}/510$
9-bit	\$01FF (511)	$f_{TC1}/1022$
10-bit	\$03FF(1023)	$f_{TC1}/2046$

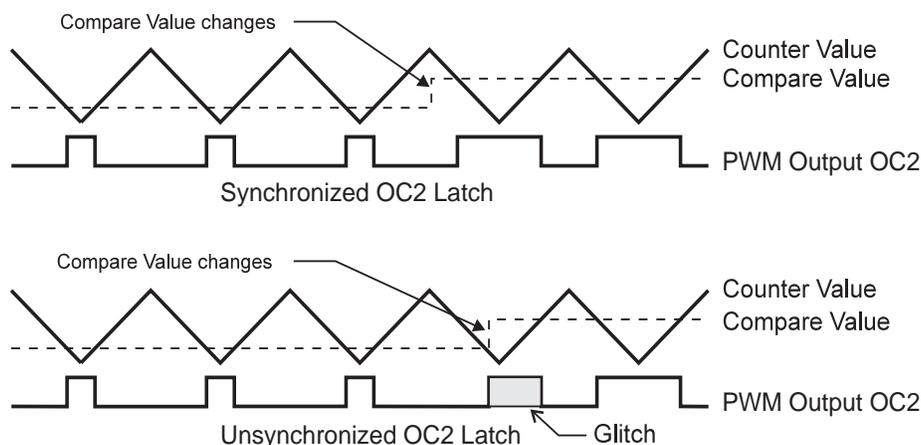
**Table 14.** Compare1 Mode Select in PWM Mode

COM1X1	COM1X0	Effect on OCN1
0	0	Not connected
0	1	Not connected
1	0	Cleared on compare match, upcounting. Set on compare match, downcounting (non-inverted PWM).
1	1	Cleared on compare match, downcounting. Set on compare match, upcounting (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 33 for an example.

**Figure 33.** Effects of Unsynchronized OCR1 Latching



During the time between the write and the latch operation, a read from OCR1A or OCR1B will read the contents of the temporary location. This means that the most recently written value always will read out of OCR1A/B.

When OCR1 contains \$0000 or TOP, the output OC1A/OC1B is held low or high according to the settings of COM1A1/COM1A0 or COM1B1/COM1B0. This is shown in Table 15.

**Table 15.** PWM Outputs OCR1X = \$0000 or TOP

COM1X1	COM1X0	OCR1X	Output OC1X
1	0	\$0000	L
1	0	TOP	H
1	1	\$0000	H
1	1	TOP	L

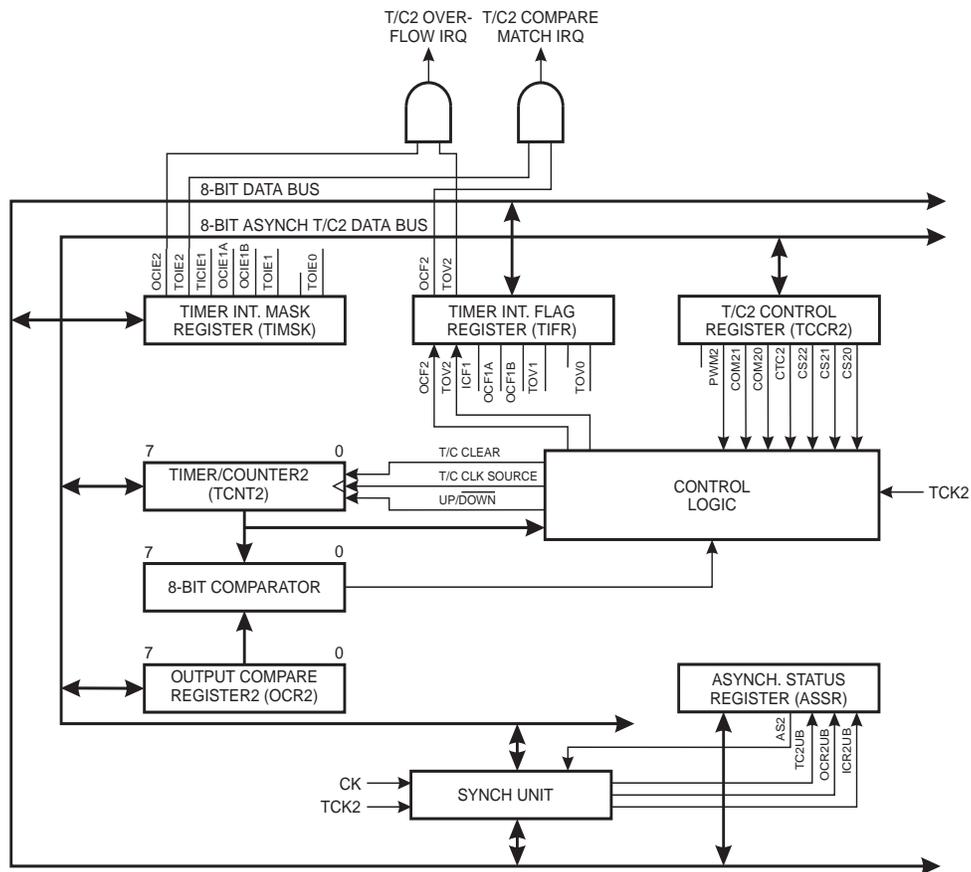
Note: X = A or B

In PWM mode, the Timer Overflow Flag1, TOV1, is set when the counter changes direction at \$0000. Timer Overflow Interrupt1 operates exactly as in normal Timer/Counter mode, i.e. it is executed when TOV1 is set provided that Timer Overflow Interrupt1 and global interrupts are enabled. This does also apply to the Timer Output Compare1 flags and interrupts.

## The 8-Bit Timer/Counter 2

Figure 34 shows the block diagram for Timer/Counter2.

**Figure 34.** Timer/Counter2 Block Diagram



The 8-bit Timer/Counter2 can select clock source from TCK2 or prescaled TCK2. It can also be stopped as described in the specification for the Timer/Counter Control Register TCCR2.

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/Counter Control Register TCCR2. The interrupt enable/disable settings are found in the Timer/Counter Interrupt Mask Register - TIMSK.

This module features a high resolution and a high accuracy usage with the lower prescaling opportunities. Similarly, the high prescaling opportunities make this unit useful for lower speed functions or exact timing functions with infrequent actions.

The Timer/Counter supports an Output Compare function using the Output Compare Register OCR2 as the data source to be compared to the Timer/Counter contents. The Output Compare function includes optional clearing of the counter on compare match, and action on the Output Com-

pare Pin - PD7(OC2) - on compare match. Writing to PORTD7 sets the OC2 value correspondingly.

Timer/Counter2 can also be used as an 8-bit Pulse Width Modulator. In this mode, Timer/Counter2 and the output compare register serve as a glitch-free, stand-alone PWM with centered pulses. Refer to page 34 for a detailed description on this function.

### The Timer/Counter2 Control Register - TCCR2

Bit	7	6	5	4	3	2	1	0	
\$25 (\$45)	-	PWM2	COM21	COM20	CTC2	CS22	CS21	CS20	TCCR2
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

• **Bit 7 - Res: Reserved Bit**

This bit is a reserved bit in the AT90S4434/8535 and always reads as zero.

• **Bit 6 - PWM2: Pulse Width Modulator Enable**

When set (one) this bit enables PWM mode for Timer/Counter2. This mode is described on page 36.

• **Bits 5,4 - COM21, COM20: Compare Output Mode, bits 1 and 0**

The COMn1 and COMn0 control bits determine any output pin action following a compare match in Timer/Counter2. Output pin actions affect pin PD7(OC2). Since this is an alternative function to an I/O port, the corresponding direction control bit must be set (one) to control an output pin. The control configuration is shown in Table 16.

**Table 16.** Compare Mode Select

COM21	COM20	Description
0	0	Timer/Counter disconnected from output pin OC2
0	1	Toggle the OC2 output line.
1	0	Clear the OC2 output line (to zero).
1	1	Set the OC2 output line (to one).

Note: In PWM mode, these bits have a different function. Refer to Table 18 for a detailed description. When changing the COM21/COM20 bits, the Output Compare 2 Interrupt must be disabled by clearing its Interrupt Enable bit in the TIMSK Register. Otherwise an interrupt can occur when the bits are changed.

• **Bit 3 - CTC2: Clear Timer/Counter on Compare Match**

When the CTC2 control bit is set (one), Timer/Counter2 is reset to \$00 in the CPU clock cycle after a compare match. If the control bit is cleared, Timer/Counter2 continues counting and is unaffected by a compare match. Since the

### The Timer/Counter2 - TCNT2

Bit	7	6	5	4	3	2	1	0	
\$24 (\$44)	MSB							LSB	TCNT2
Read/Write	R/W								
Initial value	0	0	0	0	0	0	0	0	

This 8-bit register contains the value of Timer/Counter2.

Timer/Counter2 is realized as an up or up/down (in PWM mode) counter with read and write access. If the Timer/Counter2 is written to and a clock source is selected, it continues counting in the timer clock cycle following the write operation.

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 CTC2 is set:

... | C-1 | C | C+1 | 0 | 1 | ...

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

... | C-1, C-1, C-1, C-1, C-1, C-1, C-1, C-1 | C, C, C, C, C, C, C, C | C+1, 0, 0, 0, 0, 0, 0, 0 | 1, 1, 1, ...

In PWM mode, this bit has no effect.

• **Bits 2,1,0 - CS22, CS21, CS20: Clock Select bits 2,1 and 0**

The Clock Select bits 2,1 and 0 define the prescaling source of Timer/Counter2.

**Table 17.** Timer/Counter2 Prescale Select

CS22	CS21	CS20	Description
0	0	0	Timer/Counter2 is stopped.
0	0	1	TCK2
0	1	0	TCK2 / 8
0	1	1	TCK2 / 32
1	0	0	TCK2 / 64
1	0	1	TCK2 / 128
1	1	0	TCK2 / 256
1	1	1	TCK2 / 1024

The Stop condition provides a Timer Enable/Disable function. The prescaled CK modes are scaled directly from the CK oscillator clock.

## Timer/Counter2 Output Compare Register - OCR2

Bit	7	6	5	4	3	2	1	0	
\$23 (\$43)	<b>MSB</b>							<b>LSB</b>	<b>OCR2</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 output compare register is an 8-bit read/write register. The Timer/Counter Output Compare Register contains the data to be continuously compared with Timer/Counter2. Actions on compare matches are specified in TCCR2. A compare match does only occur if Timer/Counter2 counts to the OCR2 value. A software write that sets TCNT2 and OCR2 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.

Precaution must be taken when Timer/Counter2 operates in Asynchronous mode, i.e. the AS2 bit in ASSR is set(one). When writing OCR2, the value is transferred to the register on the TCK2 clock following the write operation.

### Timer/Counter 2 in PWM mode

When the PWM mode is selected, Timer/Counter2 and the Output Compare Register - OCR2 form an 8-bit, free-running, glitch-free and phase correct PWM with outputs on the PD7(OC2) pin. Timer/Counter2 acts as an up/down counter, counting up from \$00 to \$FF, when it turns and counts down again to zero before the cycle is repeated. When the counter value matches the contents of the Out-

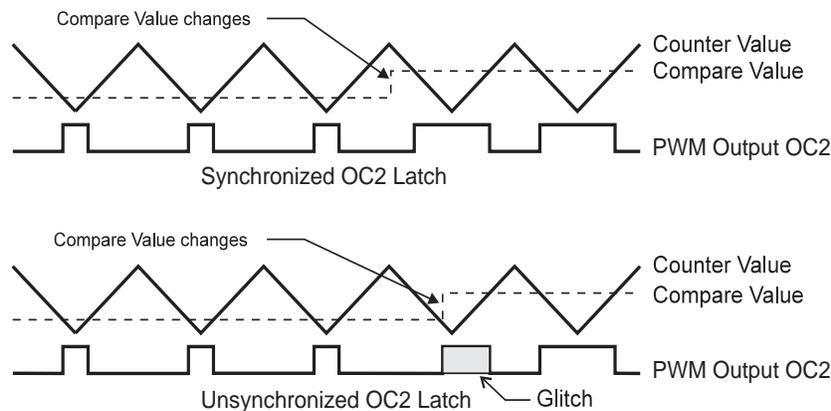
put Compare register, the PD7(OC2) pin is set or cleared according to the settings of the COM21/COM20 bits in the Timer/Counter2 Control Registers TCCR2. Refer to Table 18 for details.

**Table 18.** Compare Mode Select in PWM Mode

COMn1	COMn0	Effect on Compare Pin
0	0	Not connected
0	1	Not connected
1	0	Cleared on compare match, upcounting. Set on compare match, downcounting (non-inverted PWM).
1	1	Cleared on compare match, downcounting. Set on compare match, upcounting (inverted PWM).

Note that in PWM mode, the Output Compare register is transferred to a temporary location when written. The value is latched when the Timer/Counter reaches \$FF. This prevents the occurrence of odd-length PWM pulses (glitches) in the event of an unsynchronized OCR2 write. See Figure 35 for an example.

**Figure 35.** Effects of Unsynchronized OCR Latching



During the time between the write and the latch operation, a read from OCR2 will read the contents of the temporary location. This means that the most recently written value always will read out of OCR2.

When OCR2 contains \$00 or \$FF, the output PD7(OC2) is held low or high according to the settings of COM21/COM20. This is shown in Table 19.

**Table 19.** PWM Outputs OCR2 = \$00 or \$FF

COM21	COM20	OCR2	Output PWMn
1	0	\$00	L
1	0	\$FF	H
1	1	\$00	H
1	1	\$FF	L

In PWM mode, the Timer Overflow Flag - TOV2, is set when the counter changes direction at \$00. Timer Overflow Interrupt2 operates exactly as in normal Timer/Counter mode, i.e. it is executed when TOV2 is set provided that Timer Overflow Interrupt and global interrupts are enabled. This does also apply to the Timer Output Compare flag and interrupt.

The frequency of the PWM will be Timer Clock Frequency divided by 510.

### Asynchronous Operation of Timer/Counter2

When Timer/Counter2 operates asynchronously, some considerations must be taken.

- Warning: When switching between asynchronous and synchronous clocking of Timer/Counter2, the timer registers; TCNT2, OCR2 and TCCR2 might get corrupted. A safe procedure for switching clock source is:
  1. Disable the time 2 interrupts OCIE2 and TOIE2.
  2. Select clock source by setting AS2 as appropriate.
  3. Write new values to TCNT2, OCR2 and TCCR2.
  4. To switch to asynchronous operation: Wait for TCN2UB, OCR2UB and TCR2UB.
  5. Enable interrupts, if needed.
- The oscillator is optimized for use with a 32,768 Hz watch crystal. An external clock signal applied to this pin goes through the same amplifier having a bandwidth of 256 kHz. The external clock signal should therefore be in the interval 0 Hz - 256 kHz. The frequency of the clock signal applied to the TOSC1 pin must be lower than one fourth of the CPU main clock frequency.
- When writing to one of the registers TCNT2, OCR2, or TCCR2, the value is transferred to a temporary register, and latched after two positive edges on TOSC1. The user should not write a new value before the contents of the temporary register have been transferred to its destination. Each of the three mentioned registers have their individual temporary register, which means that e.g. writing to TCNT2 does not disturb an OCR2 write in progress. To detect that a transfer to the destination register has taken place, a Asynchronous Status Register - ASSR has been implemented.

- When entering a sleep mode after having written to TCNT2, OCR2 or TCCR2, the user must wait until the written register has been updated if Timer/Counter2 is used to wake up the device. Otherwise, the MCU will go to sleep before the changes have had any effect. This is extremely important if the Output Compare2 interrupt is used to wake up the device; Output compare is disabled during write to OCR2 or TCNT2. If the write cycle is not finished (i.e. the user goes to sleep before the OCR2UB bit returns to zero), the device will never get a compare match and the MCU will not wake up.
- If Timer/Counter2 is used to wake up the device from Power Save mode, precautions must be taken if the user wants to re-enter Power Save mode: The interrupt logic needs one TOSC1 cycle to be reset. If the time between wake up and re-entering Power Save mode is less than one TOSC1 cycle, the interrupt will not occur and the device will fail to wake up. If the user is in doubt whether the time before re-entering Power Save is sufficient, the following algorithm can be used to ensure that one TOSC1 cycle has elapsed:
  1. Write a value to TCCR2, TCNT2 or OCR2
  2. Wait until the corresponding Update Busy flag in ASSR returns to zero.
  3. Enter Power Save mode
- When asynchronous operation is selected, the 32 kHz oscillator for Timer/Counter2 is always running, except in power down mode. After a power up reset or wakeup from power down, the user should be aware of the fact that this oscillator might take as long as one second to stabilize. The user is advised to wait for at least one second before using Timer/Counter2 after power-up or wake-up from power down.
- Description of wake up from power save mode when the timer is clocked asynchronously: When the interrupt condition is met, the wake up process is started on the following cycle of the timer clock, that is, the timer is always advanced by at least one before the processor can read the counter value. The interrupt flags are updated 3 processor cycles after the processor clock has started. During these cycles, the processor executes instructions, but the interrupt condition is not readable, and the interrupt routine has not started yet.
- During asynchronous operation, the synchronization of the interrupt flags for the asynchronous timer takes 3 processor cycles plus one timer cycle. The timer is therefore advanced by at least one before the processor can read the timer value causing the setting of the interrupt flag. The output compare pin is changed on the timer clock and is not synchronized to the processor clock.

## Asynchronous Status Register - ASSR

Bit	7	6	5	4	3	2	1	0	
\$22 (\$22)	-	-	-	-	AS2	TCN2UB	OCR2UB	TCR2UB	ASSR
Read/Write	R	R	R	R	R/W	R	R	R	
Initial value	0	0	0	0	0	0	0	0	

- Bit 7..4 - Res: Reserved Bits**

These bits are reserved bits in the AT90S4434/8535 and always read as zero.

- Bit 3 - AS2: Asynchronous Timer/Counter2**

When AS2 is set(1), Timer/Counter2 is clocked from the TOSC1 pin. Pins PC6 and PC7 become connected to a crystal oscillator and cannot be used as general I/O pins. When cleared (0) Timer/Counter2 is clocked from the internal system clock, CK. When the value of this bit is changed the contents of TCNT2, OCR2 and TCCR2 might get corrupted.

- Bit 2 - TCN2UB: Timer/Counter2 Update Busy**

When Timer/Counter2 operates asynchronously and TCNT2 is written, this bit becomes set (1). When the value written to TCNT2 has been updated from the temporary storage register, this bit is cleared (0) by hardware. A logical zero in this bit indicates that TCNT2 is ready to be updated with a new value.

- Bit 1 - OCR2UB: Output Compare Register2 Update Busy**

When Timer/Counter2 operates asynchronously and OCR2 is written, this bit becomes set (1). When the value written to OCR2 has been updated from the temporary storage register, this bit is cleared (0) by hardware. A logical zero in this bit indicates that OCR2 is ready to be updated with a new value.

- Bit 0 - TCR2UB: Timer/Counter Control Register2 Update Busy**

When Timer/Counter2 operates asynchronously and TCCR2 is written, this bit becomes set (1). When the value written to TCCR2 has been updated from the temporary storage register, this bit is cleared (0) by hardware. A logical zero in this bit indicates that TCCR2 is ready to be updated with a new value.

If a write is performed to any of the three Timer/Counter2 registers while its update busy flag is set (1), the updated value might get corrupted and cause an unintentional interrupt to occur.

## The Watchdog Timer Control Register - WDTCR

Bit	7	6	5	4	3	2	1	0	
\$21 (\$41)	-	-	-	WDTOE	WDE	WDP2	WDP1	WDP0	WDTCR
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

- Bits 7..5 - Res: Reserved bits**

These bits are reserved bits in the AT90S4434/8535 and will always read as zero.

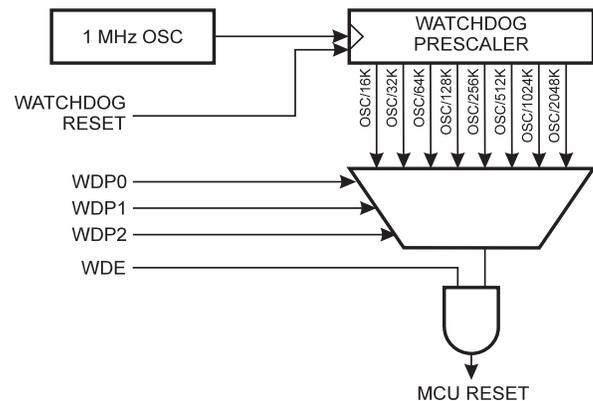
The mechanisms for reading TCNT2, OCR2, and TCCR2 are different. When reading TCNT2, the actual timer value is read. When reading OCR2 or TCCR2, the value in the temporary storage register is read.

## The Watchdog Timer

The Watchdog Timer is clocked from a separate on-chip oscillator which runs at 1MHz. This is the typical value at  $V_{CC} = 5V$ . See characterization data for typical values at other  $V_{CC}$  levels. By controlling the Watchdog Timer prescaler, the Watchdog reset interval can be adjusted from 16K to 2,048K cycles (nominally 16 - 2048 ms). The WDR - Watchdog Reset - instruction resets the Watchdog Timer. Eight different clock cycle periods can be selected to determine the reset period. If the reset period expires without another Watchdog reset, the AT90S4434/8535 resets and executes from the reset vector. For timing details on the Watchdog reset, refer to page 17.

To prevent unintentional disabling of the watchdog, a special turn-off sequence must be followed when the watchdog is disabled. Refer to the description of the Watchdog Timer Control Register for details.

**Figure 36.** Watchdog Timer



- Bit 4 - WDTOE: Watch Dog Turn-Off Enable**

This bit must be set (1) when the WDE bit is cleared. Otherwise, the watchdog will not be disabled. Once set, hardware will clear this bit to zero after four clock cycles.



Refer to the description of the WDE bit for a watchdog disable procedure.

• **Bit 3 - WDE: Watch Dog Enable**

When the WDE is set (one) the Watchdog Timer is enabled, and if the WDE is cleared (zero) the Watchdog Timer function is disabled. WDE can only be cleared if the WDTOE bit is set(one). To disable an enabled watchdog timer, the following procedure must be followed:

1. In the same operation, write a logical one to WDTOE and WDE. A logical one must be written to

WDE even though it is set to one before the disable operation starts.

2. Within the next four clock cycles, write a logical 0 to WDE. This disables the watchdog.

• **Bits 2..0 - WDP2, WDP1, WDP0: Watch Dog Timer Prescaler 2, 1 and 0**

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

**Table 20.** Watch Dog Timer Prescale Select

WDP2	WDP1	WDP0	Timeout Period
0	0	0	16K cycles
0	0	1	32K cycles
0	1	0	64K cycles
0	1	1	128K cycles
1	0	0	256K cycles
1	0	1	512K cycles
1	1	0	1,024K cycles
1	1	1	2,048K cycles

## EEPROM Read/Write Access

The EEPROM access registers are accessible in the I/O space.

The write access time is in the range of 2.5 - 4ms, depending on the  $V_{CC}$  voltages. A self-timing function lets the user software detect when the next byte can be written. A special EEPROM Ready interrupt can be set to trigger when the EEPROM is ready to accept new data.

In order to prevent unintentional EEPROM writes, a specific write procedure must be followed. Refer to the description of the EEPROM Control Register for details on this.

When the EEPROM is read or written, the CPU is halted for two clock cycles before the next instruction is executed.

### The EEPROM Address Register - EEARH and EEARL

Bit	15	14	13	12	11	10	9	8	
\$1F (\$3F)	-	-	-	-	-	-	-	EEAR9	EEARH
\$1E (\$3E)	EEAR7	EEAR6	EEAR5	EEAR4	EEAR3	EEAR2	EEAR1	EEAR0	EEARL
	7	6	5	4	3	2	1	0	
Read/Write	R/W								
	R/W								
Initial value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

The EEPROM Address Registers - EEARH and EEARL specify the EEPROM address in the 256/512 bytes

EEPROM space. The EEPROM data bytes are addressed linearly between 0 and 255/511.

### The EEPROM Data Register - EEDR

Bit	7	6	5	4	3	2	1	0	
\$1D (\$3D)	MSB							LSB	EEDR
Read/Write	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

**The EEPROM Control Register - EECR**

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

• **Bit 7..4 - Res: Reserved bits**

These bits are reserved bits in the AT90S4434/8535 and will always read as zero.

• **Bit 3 - EERIE: EEPROM Ready Interrupt Enable**

When the I bit in SREG and EERIE are set (one), the EEPROM Ready Interrupt is enabled. When cleared (zero), the interrupt is disabled. The EEPROM Ready interrupt generates a constant interrupt when EEWE is cleared (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.

• **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

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.

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.

**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 notes AVR 190 and 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.

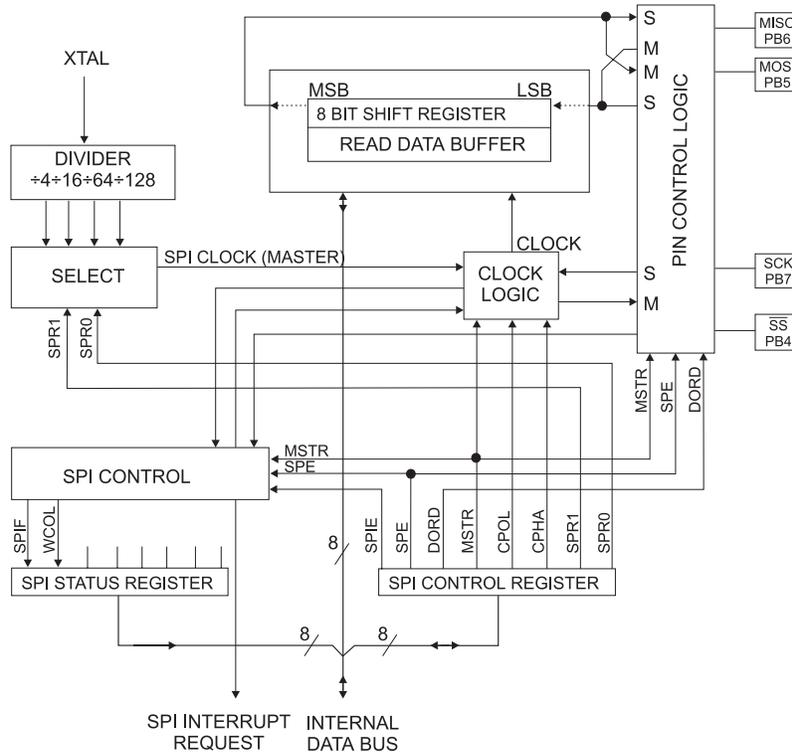
## The Serial Peripheral Interface - SPI

The Serial Peripheral Interface (SPI) allows high-speed synchronous data transfer between the AT90S4434/8535 and peripheral devices or between several AT90S4434/8535 devices. The AT90S4434/8535 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

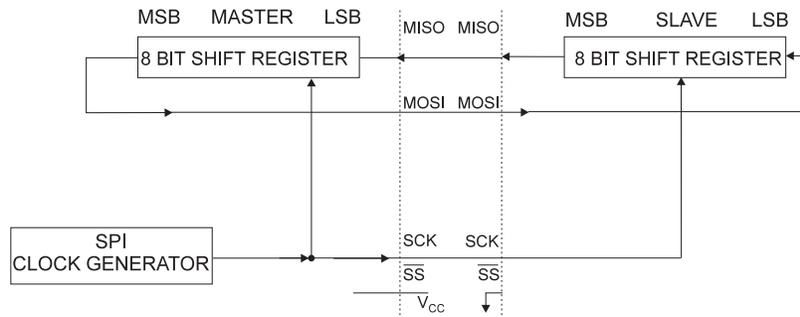
Figure 37. SPI Block Diagram



The interconnection between master and slave CPUs with SPI is shown in Figure 38. The PB7(SCK) pin is the clock output in the Master mode and is the clock input in the Slave mode. Writing to the SPI Data Register of the master CPU starts the SPI clock generator, and the data written shifts out of the PB5(MOSI) pin and into the PB5(MOSI) pin of the slave CPU. After shifting one byte, the SPI clock generator stops, setting the end of transmission flag (SPIF). If the SPI interrupt enable bit (SPIE) in the SPCR register is

set, an interrupt is requested. The Slave Select input, PB4(SS), is set low to select an individual SPI device as a slave. The two shift registers in the Master and the Slave can be considered as one distributed 16-bit circular shift register. This is shown in Figure 38. When data is shifted from the master to the slave, data is also shifted from the master to the slave, data is also shifted in the opposite direction, simultaneously. During one shift cycle, data in the master and the slave is interchanged.

**Figure 38.** SPI Master-Slave Interconnection



The system is single buffered in the transmit direction and double buffered in the receive direction. This means that characters to be transmitted cannot be written to the SPI Data Register before the entire shift cycle is completed. When receiving data, however, a received character must be read from the SPI Data Register before the next character has been completely shifted in. Otherwise, the first character is lost.

When the SPI is enabled, the data direction of the MOSI, MISO, SCK and  $\overline{SS}$  pins is overridden according to the following table:

**Table 21.** SPI Pin Overrides

Pin	Direction, Master SPI	Direction, Slave SPI
MOSI	User Defined	Input
MISO	Input	User Defined
SCK	User Defined	Input
$\overline{SS}$	User Defined	Input

### $\overline{SS}$ Pin Functionality

When the SPI is configured as a master (MSTR in SPCR is set), the user can determine the direction of the  $\overline{SS}$  pin. If  $\overline{SS}$  is configured as an output, the pin is a general output pin which does not affect the SPI system. If  $\overline{SS}$  is configured as an input, it must be held high to ensure Master SPI operation. If, in master mode, the  $\overline{SS}$  pin is input, and is

driven low by peripheral circuitry, the SPI system interprets this as another master selecting the SPI as a slave and starting to send data to it. To avoid bus contention, the SPI system takes the following actions:

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

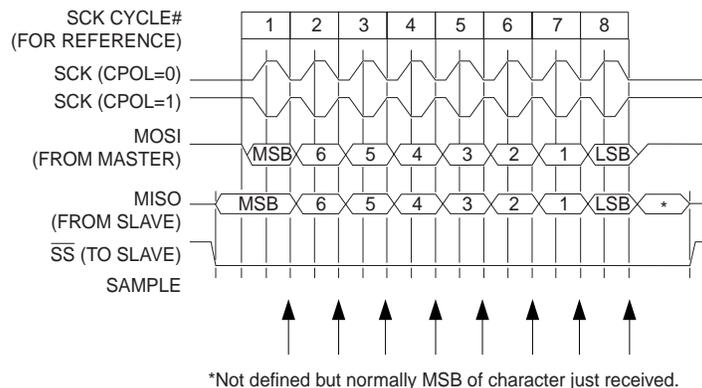
Thus, when interrupt-driven SPI transmission is used in master mode, and there exists a possibility that  $\overline{SS}$  is driven low, the interrupt should always check that the MSTR bit is still set. Once the MSTR bit has been cleared by a slave select, it must be set by the user.

When the SPI is configured as a slave, the  $\overline{SS}$  pin is always input. When  $\overline{SS}$  is held low, the SPI is activated and MISO becomes an output if configured so by the user. All other pins are inputs. When  $\overline{SS}$  is driven high, all pins are inputs, and the SPI is passive, which means that it will not receive incoming data.

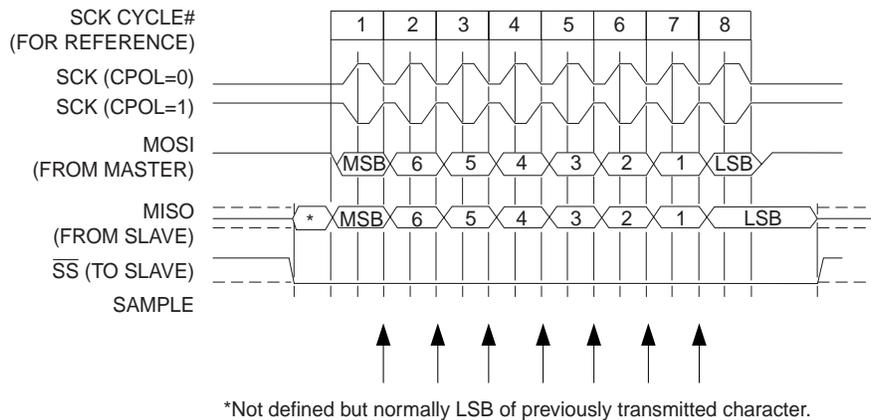
### Data Modes

There are four combinations of SCK phase and polarity with respect to serial data, which are determined by control bits CPHA and CPOL. The SPI data transfer formats are shown in Figure 39 and Figure 40.

**Figure 39.** SPI Transfer Format with CPHA = 0



**Figure 40. SPI Transfer Format with CPHA = 1**



**The SPI Control Register - SPCR**

Bit	7	6	5	4	3	2	1	0	
\$0D (\$2D)	<b>SPCR</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	

- **Bit 7 - SPIE: SPI Interrupt Enable**  
This bit causes setting of the SPIF bit in the SPSR register to execute the SPI interrupt provided that global interrupts are enabled.
- **Bit 6 - SPE: SPI Enable**  
When the SPE bit is set (one), the SPI is enabled. This bit must be set to enable any SPI operations.
- **Bit 5 - DORD: Data Order**  
When the DORD bit is set (one), the LSB of the data word is transmitted first.  
When the DORD bit is cleared (zero), the MSB of the data word is transmitted first.
- **Bit 4 - MSTR: Master/Slave Select**  
This bit selects Master SPI mode when set (one), and Slave SPI mode when cleared (zero). If  $\overline{SS}$  is configured as an input and is driven low while MSTR is set, MSTR will be cleared, and SPIF in SPSR will become set. The user will then have to set MSTR to re-enable SPI master mode.

- **Bit 3 - CPOL: Clock Polarity**  
When this bit is set (one), SCK is high when idle. When CPOL is cleared (zero), SCK is low when idle. Refer to Figure 40 and Figure 41 for additional information.
- **Bit 2 - CPHA: Clock Phase**  
Refer to Figure 40 or Figure 41 for the functionality of this bit.
- **Bits 1,0 - SPR1, SPR0: SPI Clock Rate Select 1 and 0**  
These two bits control the SCK rate of the device configured as a master. SPR1 and SPR0 have no effect on the slave. The relationship between SCK and the Oscillator Clock frequency  $f_{cl}$  is shown in the following table:

**Table 22. Relationship Between SCK and the Oscillator Frequency**

SPR1	SPR0	SCK Frequency
0	0	$f_{cl} / 4$
0	1	$f_{cl} / 16$
1	0	$f_{cl} / 64$
1	1	$f_{cl} / 128$

**The SPI Status Register - SPSR**

Bit	7	6	5	4	3	2	1	0	
\$0E (\$2E)	<b>SPSR</b>								
Read/Write	R	R	R	R	R	R	R	R	
Initial value	0	0	0	0	0	0	0	0	

- **Bit 7 - SPIF : SPI Interrupt Flag**  
When a serial transfer is complete, the SPIF bit is set (one) and an interrupt is generated if SPIE in SPCR is set (one) and global interrupts are enabled. If  $\overline{SS}$  is an input and is driven low when the SPI is in master mode, this will also set

the SPIF flag. SPIF is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, the SPIF bit is cleared by first reading the SPI status register with SPIF set (one), then accessing the SPI Data Register (SPDR).

• **Bit 6 - WCOL : Write Collision flag**

The WCOL bit is set if the SPI data register (SPDR) is written during a data transfer. During data transfer, the result of reading the SPDR register may be incorrect, and writing to it will have no effect. The WCOL bit (and the SPIF bit) are cleared (zero) by first reading the SPI Status Register with WCOL set (one), and then accessing the SPI Data Register.

• **Bit 5..0 - Res: Reserved bits**

These bits are reserved bits in the AT90S4434/8535 and will always read as zero.

The SPI interface on the AT90S4434/8535 is also used for program memory and EEPROM downloading or uploading. See page 70 for serial programming and verification.

**The SPI Data Register - SPDR**

Bit	7	6	5	4	3	2	1	0	
\$0F (\$2F)	<b>MSB</b>							<b>LSB</b>	<b>SPDR</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 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.

**The UART**

The AT90S4434/8535 features a full duplex Universal Asynchronous Receiver and Transmitter (UART). The main features are:

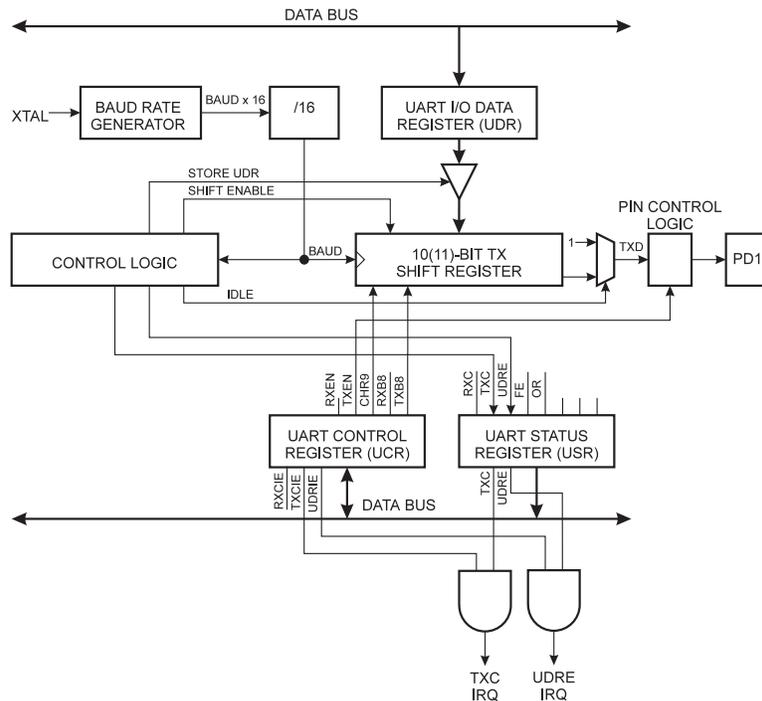
- Baud rate generator generates any baud rate
- High baud rates at low XTAL frequencies
- 8 or 9 bits data
- Noise filtering
- Overrun detection

- Framing Error detection
- False Start Bit detection
- Three separate interrupts on TX Complete, TX Data Register Empty and RX Complete
- Buffered Transmit and Receive.

**Data Transmission**

A block schematic of the UART transmitter is shown in Figure 41.

**Figure 41. UART Transmitter**



Data transmission is initiated by writing the data to be transmitted to the UART I/O Data Register, UDR. Data is transferred from UDR to the Transmit shift register when:

- A new character has been written to UDR after the stop bit from the previous character has been shifted out. The shift register is loaded immediately.
- A new character has been written to UDR before the stop bit from the previous character has been shifted out. The shift register is loaded when the stop bit of the character currently being transmitted has been shifted out.

If the 10(11)-bit Transmitter shift register is empty or when, data is transferred from UDR to the shift register. At this time the UDRE (UART Data Register Empty) bit in the UART Status Register, USR, is set. When this bit is set (one), the UART is ready to receive the next character. At the same time as the data is transferred from UDR to the 10(11)-bit shift register, bit 0 of the shift register is cleared (start bit) and bit 9 or 10 is set (stop bit). If 9 bit data word is selected (the CHR9 bit in the UART Control Register, UCR is set), the TXB8 bit in UCR is transferred to bit 9 in the Transmit shift register.

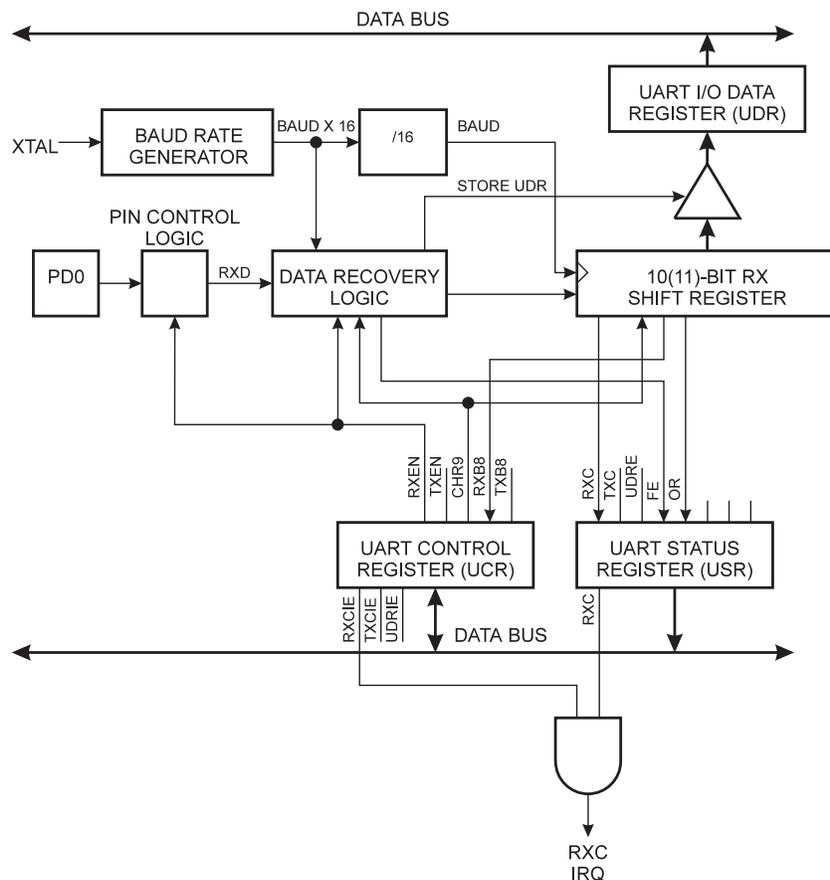
On the Baud Rate clock following the transfer operation to the shift register, the start bit is shifted out on the TXD pin. Then follows the data, LSB first. When the stop bit has been shifted out, the shift register is loaded if any new data has been written to the UDR during the transmission. During loading, UDRE is set. If there is no new data in the UDR register to send when the stop bit is shifted out, the UDRE flag will remain set until UDR is written again. When no new data has been written, and the stop bit has been present on TXD for one bit length, the TX Complete Flag, TXC, in USR is set.

The TXEN bit in UCR enables the UART transmitter when set (one). When this bit is cleared (zero), the PD1 pin can be used for general I/O. When TXEN is set, the UART Transmitter will be connected to PD1, which is forced to be an output pin regardless of the setting of the DDD1 bit in DDRD.

### Data Reception

Figure 42 shows a block diagram of the UART Receiver.

**Figure 42.** UART Receiver



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 sam-

ples 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 43.

**Figure 43. 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.

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.

## UART Control

### The UART I/O Data Register - UDR

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

### The 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 recep-

tion 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 one 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 transmission 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.

**The UART Control Register - UCR**

Bit	7	6	5	4	3	2	1	0	
\$0A (\$2A)	RXCIE	TXCIE	UDRIE	RXEN	TXEN	CHR9	RXB8	TXB8	UCR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	W	
Initial value	0	0	0	0	0	0	0	0	

• **Bit 7 - RXCIE: RX Complete Interrupt Enable**

When this bit is set (one), a setting of the RXC bit in USR will cause the Receive Complete interrupt routine to be executed provided that global interrupts are enabled.

• **Bit 6 - TXCIE: TX Complete Interrupt Enable**

When this bit is set (one), a setting of the TXC bit in USR will cause the Transmit Complete interrupt routine to be executed provided that global interrupts are enabled.

• **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,

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.

The FE bit is cleared when the stop bit of received data is one.

• **Bit 3 - OR: OverRun**

This bit is set if an Overrun condition is detected, i.e. when a character already present in the UDR register is not read before the next character has been shifted into the Receiver Shift register. The OR bit is buffered, which means that it will be set once the valid data still in UDR is read.

The OR bit is cleared (zero) when data is received and transferred to UDR.

• **Bits 2..0 - Res: Reserved bits**

These bits are reserved bits in the AT90S4434/8535 and will always read as zero.

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.

**The Baud Rate Generator**

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

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

• 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 23. UBRR values which yield an actual baud rate differing less than 2% from the target baud rate, are bold in the table.

**Table 23. UBRR Settings at Various Crystal Frequencies (Examples)**

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

Maximum Baud rate to each frequency.

### The UART Baud Rate Register - UBRR

Bit	7	6	5	4	3	2	1	0	
\$09 (\$29)	<div style="border: 1px solid black; padding: 2px; display: inline-block;"> <span style="float: left; padding-right: 5px;">MSB</span> <span style="float: right; padding-left: 5px;">LSB</span> </div>								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	

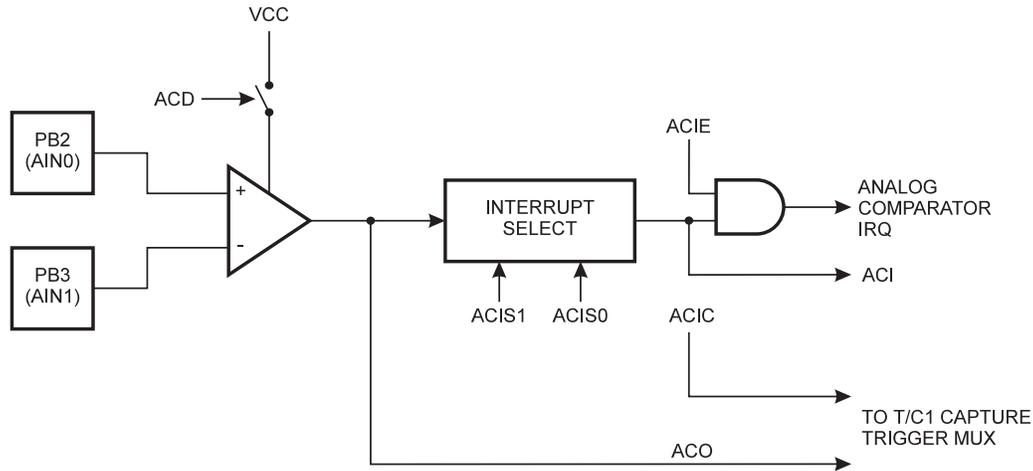
The UBRR register is an 8-bit read/write register which specifies the UART Baud Rate according to the equation on the previous page.

## The Analog Comparator

The analog comparator compares the input values on the positive pin PB2 (AIN0) and negative pin PB3 (AIN1). When the voltage on the positive pin PB2 (AIN0) is higher than the voltage on the negative pin PB3 (AIN1), the Analog Comparator Output, ACO is set (one). The comparator's output can be set to trigger the Timer/Counter1 Input

Capture function. In addition, the comparator can trigger a separate interrupt, exclusive to the Analog Comparator. The user can select Interrupt triggering on comparator output rise, fall or toggle. A block diagram of the comparator and its surrounding logic is shown in Figure 44.

**Figure 44.** Analog Comparator Block Diagram



### The Analog Comparator Control And Status Register - ACSR

Bit	7	6	5	4	3	2	1	0	
\$08 (\$28)	ACD	-	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	ACSR
Read/Write	R/W	R	R	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

- **Bit 7 - ACD: Analog Comparator Disable**

When this bit is set(one), the power to the analog comparator is switched off. This bit can be set at any time to turn off the analog comparator. When changing the ACD bit, the Analog Comparator Interrupt must be disabled by clearing the ACIE bit in ACSR. Otherwise an interrupt can occur when the bit is changed.

- **Bit 6 - Res: Reserved bit**

This bit is a reserved bit in the AT90S4434/8535 and will always read as zero.

- **Bit 5 - ACO: Analog Comparator Output**

ACO is directly connected to the comparator output.

- **Bit 4 - ACI: Analog Comparator Interrupt Flag**

This bit is set (one) when a comparator output event triggers the interrupt mode defined by ACI1 and ACI0. The Analog Comparator Interrupt routine is executed if the ACIE bit is set (one) and the I-bit in SREG is set (one). ACI is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, ACI is cleared by writing a logic one to the flag.

- **Bit 3 - ACIE: Analog Comparator Interrupt Enable**

When the ACIE bit is set (one) and the I-bit in the Status Register is set (one), the analog comparator interrupt is activated. When cleared (zero), the interrupt is disabled.

- **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 24.

**Table 24.** 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

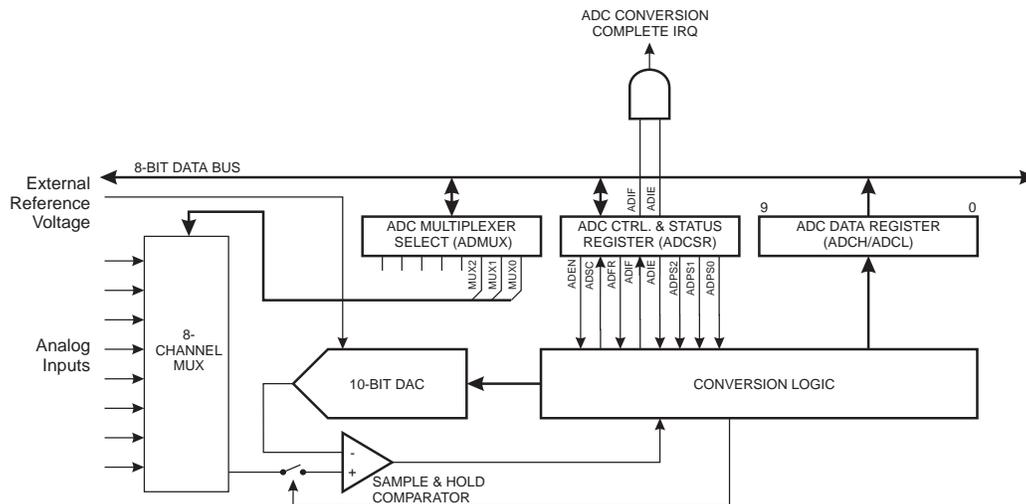
Note: When changing the ACIS1/ACIS0 bits, The Analog Comparator Interrupt must be disabled by clearing its Interrupt Enable bit in the ACSR register. Otherwise an interrupt can occur when the bits are changed.

## The Analog to Digital Converter

Feature list:

- 10-bit Resolution

**Figure 45.** Analog to Digital Converter Block Schematic



### Operation

The ADC can operate in two modes - Single Conversion and Free Running Mode. In Single Conversion Mode, each conversion will have to be initiated by the user. In Free Running Mode, the ADC is constantly sampling and updating the ADC Data Register. The ADFR bit in ADCSR selects between the two available modes.

The ADC is enabled by writing a logical one to the ADC Enable bit, ADEN in ADCSR. The first conversion that is started after enabling the ADC, will be preceded by a dummy conversion to initialize the ADC. To the user, the

- $\pm\frac{1}{2}$  LSB Accuracy
- 65 - 260  $\mu$ s Conversion Time
- 8 Multiplexed Input Channels
- Rail-to-Rail Input Range
- Free Run or Single Conversion Mode
- Interrupt on ADC Conversion Complete
- Sleep Mode Noise Canceler

The AT90S4434/8535 features a 10-bit successive approximation ADC. The ADC is connected to an 8-channel Analog Multiplexer which allows each pin of Port A to be used as an input for the ADC. The ADC contains a Sample and Hold Amplifier which ensures that the input voltage to the ADC is held at a constant level during conversion. A block diagram of the ADC is shown in Figure 45.

The ADC has two separate analog supply voltage pins, AVCC and AGND. AGND must be connected to GND, and the voltage on AVCC must not differ more than  $\pm 0.3V$  from  $V_{CC}$ . See the paragraph ADC Noise Canceling Techniques on how to connect these pins.

An external reference voltage must be applied to the AREF pin. This voltage must be in the range AGND -  $AV_{CC}$ .

only difference will be that this conversion takes 25 clock pulses instead of the normal 14.

A conversion is started by writing a logical one to the ADC Start Conversion bit, ADSC. This bit stays high as long as the conversion is in progress and will be set to zero by hardware when the conversion is completed. If a different data channel is selected while a conversion is in progress, the ADC will finish the current conversion before performing the channel change.

As the ADC generates a 10-bit result, two data registers, ADCH and ADCL, must be read to get the result when the

conversion is complete. Special data protection logic is used to ensure that the contents of the data registers belong to the same conversion when they are read. This mechanism works as follows:

When reading data, ADCL must be read first. Once ADCL is read, ADC access to data registers is blocked. This means that if ADCL has been read, and a conversion completes before ADCH is read, none of the registers are updated and the result from the conversion is lost. When ADCH is read, ADC access to the ADCH and ADCL registers is re-enabled.

The ADC has its own interrupt which can be triggered when a conversion completes. When ADC access to the data registers is prohibited between reading of ADCH and ADCL, the interrupt will trigger even if the result gets lost.

### Prescaling

The ADC accepts input clock frequencies in the range 50 - 200 kHz. In free running mode, the ADC needs 13 clock pulses to perform a conversion, which means that the conversion time range is 65 - 260  $\mu$ s. In single conversion mode, the conversion time is 14 clock cycles. The output of the ADC is not guaranteed to be correct if the input clock is out of range. The ADPS0 - ADPS2 bits are used to gener-

### The ADC Multiplexer Select Register - ADMUX

Bit	7	6	5	4	3	2	1	0	
\$07 (\$27)	-	-	-	-	-	MUX2	MUX1	MUX0	ADMUX
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 AT90S4434/8535 and always read as zero.

### The ADC Control and Status Register - ADCSR

Bit	7	6	5	4	3	2	1	0	
\$06 (\$26)	ADEN	ADSC	ADFR	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ADCSR
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	

- **Bit 7 - ADEN: ADC Enable**

Writing a logical '1' to this bit enables the ADC. By clearing this bit to zero, the ADC is turned off. Turning the ADC off while a conversion is in progress, will terminate this conversion.

- **Bit 6 - ADSC: ADC Start Conversion**

In Single Conversion Mode, a logical '1' must be written to this bit to start each conversion. In Free Run Mode, a logical '1' must be written to this bit to start the first conversion. The first time ADSC has been written after the ADC has been enabled, or if ADSC is written at the same time as the ADC is enabled, a dummy conversion will precede the initiated conversion. This dummy conversion performs initialization of the ADC.

ate a proper ADC clock input frequency from any XTAL frequency above 100 kHz.

### ADC Noise Canceler Function

The ADC features a noise canceler that enables conversion during idle mode to reduce noise induced from the CPU core. To make use of this feature, the following procedure should be used:

1. Make sure that the ADC is enabled and is not busy converting. Single Conversion Mode must be selected and the ADC conversion complete interrupt must be enabled.  
ADEN = 1  
ADSC = 0  
ADFR = 0  
ADIE = 1
2. Enter idle mode. The ADC will start a conversion once the CPU has been halted.
3. If no other interrupts occur before the ADC conversion completes, the ADC interrupt will wake up the MCU and execute the ADC conversion complete interrupt routine.

- **Bits 2..0 - MUX2..MUX0: Analog Channel Select Bits 2-0**

The value of these three bits selects which analog input 7-0 is connected to the ADC.

ADSC will read as one as long as a conversion is in progress. When the conversion is complete, it returns to zero. When a dummy conversion precedes a real conversion, ADSC will stay high until the real conversion completes.

Writing a 0 to this bit has no effect.

- **Bit 5 - ADFR: ADC Free Run Select**

When this bit is set (one) the ADC operates in Free Running mode. In this mode, the ADC samples and updates the data registers continuously. Clearing this bit (zero) will terminate Free Running mode.

- **Bit 4 - ADIF: ADC Interrupt Flag**

This bit is set (one) when an ADC conversion completes and the data registers are updated. The ADC Conversion

Complete Interrupt is executed if the ADIE bit and the I-bit in SREG are set (one). ADIF is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, ADIF is cleared by writing a logical one to the flag. Beware that if doing a read-modify-write on ADCSR, a pending interrupt can be disabled. This also applies if the SBI and CBI instructions are used.

• **Bit 3 - ADIE: ADC Interrupt Enable**

When this bit is set (one) and the I-bit in SREG is set (one), the ADC Conversion Complete Interrupt is activated.

• **Bits 2..0 - ADPS2..ADPS0: ADC Prescaler Select Bits**

These bits determine the division factor between the XTAL frequency and the input clock to the ADC.

**Table 25.** ADC Prescaler Selections

ADPS2	ADPS1	ADPS0	Division Factor
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

**The ADC Data Register - ADCL AND ADCH**

Bit	15	14	13	12	11	10	9	8	
\$05 (\$25)	-	-	-	-	-	-	ADC9	ADC8	ADCH
\$04 (\$24)	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0	ADCL
	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	8	0	0	0	0	0	0	0	
	8	0	0	0	0	0	0	0	

When an ADC conversion is complete, the result is found in these two registers. In free-run mode, it is essential that both registers are read, and that ADCL is read before ADCH.

**Scanning Multiple Channels**

Since change of analog channel always is delayed until a conversion is finished, the free running mode can be used to scan multiple channels without interrupting the converter. Typically, the ADC Conversion Complete interrupt will be used to perform the channel shift. However, the user should take the following fact into consideration:

The interrupt triggers once the result is ready to be read. In free running mode, the next conversion will start immediately when the interrupt triggers. If ADMUX is changed after the interrupt triggers, the next conversion has already started, and the old setting is used.

**ADC Noise Canceling Techniques**

Digital circuitry inside and outside the AT90S4434/8535 generates EMI which might affect the accuracy of analog measurements. If conversion accuracy is critical, the noise level can be reduced by applying the following techniques:

1. The analog part of the AT90S4434/8535 and all analog components in the application should have a separate analog ground plane on the PCB. This ground plane is con-

nected to the digital ground plane via a single point on the PCB.

2. Keep analog signal paths as short as possible. Make sure analog tracks run over the analog ground plane, and keep them well away from high-speed switching digital tracks.

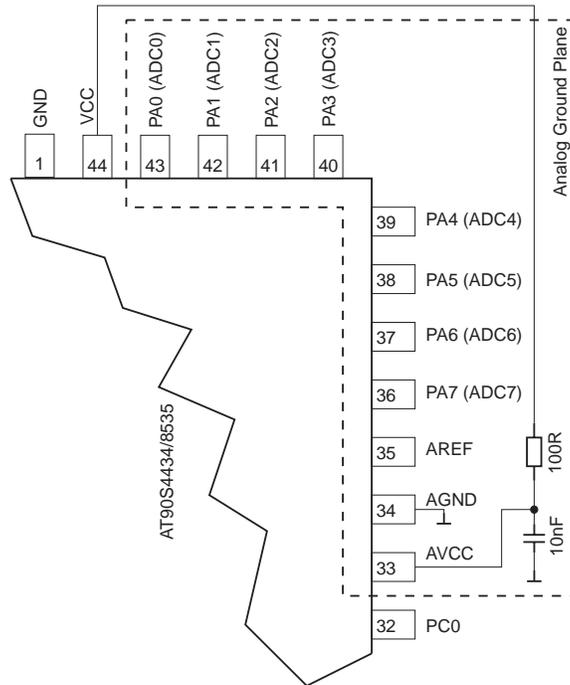
3. The AVCC pin on the AT90S4434/8535 should be connected to the digital VCC supply voltage via an RC network as shown in Figure 46.

4. Use the ADC noise canceler function to reduce induced noise from the CPU.

5. If some Port A pins are used as digital outputs, it is essential that these do not switch while a conversion is in progress.



**Figure 46. ADC Power Connections**



Note that since  $AV_{CC}$  feeds the Part A output drivers, the RC network shown should not be employed if any Port A serve as outputs.

### ADC Characteristics

$T_A = -40^{\circ}\text{C}$  to  $85^{\circ}\text{C}$

Symbol	Parameter	Condition	Min	Typ	Max	Units
	Resolution			10		Bits
	Integral Non-Linearity	$V_{REF} > 2V$		0.2	0.5	LSB
	Differential Non-Linearity	$V_{REF} > 2V$		0.2	0.5	LSB
	Zero Error (Offset)			1		LSB
	Conversion Time		65		260	ms
	Clock Frequency		50		200	KHz
$AV_{CC}$	Analog Supply Voltage		$V_{CC} - 0.3^{(1)}$		$V_{CC} + 0.3^{(2)}$	V
$V_{REF}$	Reference Voltage		AGND		$AV_{CC}$	V
$R_{REF}$	Reference Input Resistance		6	10	13	$K\Omega$
$R_{AIN}$	Analog Input Resistance			100		$M\Omega$

- Notes: 1. Minimum for  $AV_{CC}$  is 2.7V.  
2. Maximum for  $AV_{CC}$  is 6.0V.

### I/O-Ports

#### Port A

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

Three data memory address locations are allocated for Port A, one each for the Data Register - PORTA, \$1B(\$3B),

Data Direction Register - DDRA, \$1A(\$3A) and the Port A Input Pins - PINA, \$19(\$39). The Port A Input Pins address is read only, while the Data Register and the Data Direction Register are read/write.

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

Port A has an alternate function as analog inputs for the ADC. If some Port A pins are configured as outputs, it is

essential that these do not switch when a conversion is in progress. This might corrupt the result of the conversion.

During powerdown mode, the schmitt trigger of the digital input is disconnected. This allows analog signals that are close to  $V_{CC}/2$  to be present during powerdown without causing excessive power consumption.

## The Port A Data Register - PORTA

Bit	7	6	5	4	3	2	1	0	
\$1B (\$3B)	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	PORTA
Read/Write	R/W								
Initial value	0	0	0	0	0	0	0	0	

## The Port A Data Direction Register - DDRA

Bit	7	6	5	4	3	2	1	0	
\$1A (\$3A)	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	DDRA
Read/Write	R/W								
Initial value	0	0	0	0	0	0	0	0	

## The Port A Input Pins Address - PINA

Bit	7	6	5	4	3	2	1	0	
\$19 (\$39)	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	PINA
Read/Write	R	R	R	R	R	R	R	R	
Initial value	Hi-Z								

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

## PORT A as General Digital I/O

All 8 bits in PORT A are equal when used as digital I/O pins.

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

**Table 26.** DDAn Effects on PORTA Pins

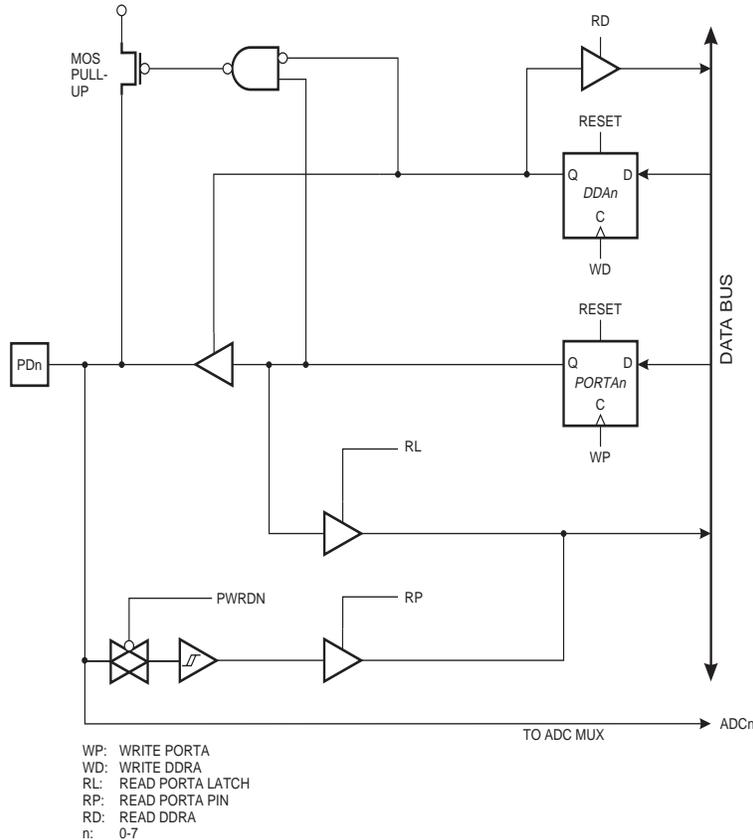
DDAn	PORTAn	I/O	Pull Up	Comment
0	0	Input	No	Tri-state (Hi-Z)
0	1	Input	Yes	PAn will source current if ext. pulled low.
1	0	Output	No	Push-Pull Zero Output
1	1	Output	No	Push-Pull One Output

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 47.** PORTA Schematic Diagrams (Pins PA0 - PA7)



## Port B

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

Three data memory address locations are allocated for the Port B, one each for the Data Register - PORTB, \$18(\$38), Data Direction Register - DDRB, \$17(\$37) and the Port B Input Pins - PINB, \$16(\$36). The Port B 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 B output buffers can sink 20mA and thus drive LED displays directly. When pins PB0 to PB7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated.

The Port B pins with alternate functions are shown in the following table:

**Table 27.** Port B Pins Alternate Functions

Port Pin	Alternate Functions
PB0	T0 (Timer/Counter 0 external counter input)
PB1	T1 (Timer/Counter 1 external counter input)
PB2	AIN0 (Analog comparator positive input)
PB3	AIN1 (Analog comparator negative input)
PB4	$\overline{SS}$ (SPI Slave Select input)
PB5	MOSI (SPI Bus Master Output/Slave Input)
PB6	MISO (SPI Bus Master Input/Slave Output)
PB7	SCK (SPI Bus Serial Clock)

When the pins are used for the alternate function, the DDRB and PORTB registers have to be set according to the alternate function description.

## The Port B Data Register - PORTB

Bit	7	6	5	4	3	2	1	0	
\$18 (\$38)	<b>PORTB7 PORTB6 PORTB5 PORTB4 PORTB3 PORTB2 PORTB1 PORTB0</b>								PORTB
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 Port B Data Direction Register - DDRB

Bit	7	6	5	4	3	2	1	0	
\$17 (\$37)	<b>DDB7 DDB6 DDB5 DDB4 DDB3 DDB2 DDB1 DDB0</b>								DDRB
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 Port B Input Pins Address - PINB

Bit	7	6	5	4	3	2	1	0	
\$16 (\$36)	<b>PINB7 PINB6 PINB5 PINB4 PINB3 PINB2 PINB1 PINB0</b>								PINB
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 B Input Pins address - PINB - is not a register, and this address enables access to the physical value on each Port B pin. When reading PORTB, the PORTB Data Latch is read, and when reading PINB, the logical values present on the pins are read.

### Port B As General Digital I/O

All 8 bits in Port B are equal when used as digital I/O pins.

PB<sub>n</sub>, General I/O pin: The DDB<sub>n</sub> bit in the DDRB register selects the direction of this pin, if DDB<sub>n</sub> is set (one), PB<sub>n</sub> is configured as an output pin. If DDB<sub>n</sub> is cleared (zero), PB<sub>n</sub> is configured as an input pin. If PORTB<sub>n</sub> 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, the PORTB<sub>n</sub> has to be cleared (zero) or the pin has to be configured as an output pin.

**Table 28.** DDB<sub>n</sub> Effects on Port B Pins

DDB <sub>n</sub>	PORTB <sub>n</sub>	I/O	Pull Up	Comment
0	0	Input	No	Tri-state (Hi-Z)
0	1	Input	Yes	PB <sub>n</sub> 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.

### Alternate Functions Of PORTB

The alternate pin configuration is as follows:

#### • SCK - PORTB, Bit 7

SCK: Master clock output, slave clock input pin for SPI channel. When the SPI is enabled as a slave, this pin is configured as an input regardless of the setting of DDB7. When the SPI is enabled as a master, the data direction of this pin is controlled by DDB7. When the pin is forced to be an input, the pull-up can still be controlled by the PORTB7 bit. See the description of the SPI port for further details.

#### • MISO - PORTB, Bit 6

MISO: Master data input, slave data output pin for SPI channel. When the SPI is enabled as a master, this pin is configured as an input regardless of the setting of DDB6. When the SPI is enabled as a slave, the data direction of this pin is controlled by DDB6. When the pin is forced to be an input, the pull-up can still be controlled by the PORTB6 bit. See the description of the SPI port for further details.

#### • MOSI - PORTB, Bit 5

MOSI: SPI Master data output, slave data input for SPI channel. When the SPI is enabled as a slave, this pin is configured as an input regardless of the setting of DDB5. When the SPI is enabled as a master, the data direction of this pin is controlled by DDB5. When the pin is forced to be an input, the pull-up can still be controlled by the PORTB5 bit. See the description of the SPI port for further details.

#### • SS - PORTB, Bit 4

SS: Slave port select input. When the SPI is enabled as a slave, this pin is configured as an input regardless of the setting of DDB4. As a slave, the SPI is activated when this pin is driven low. When the SPI is enabled as a master, the data direction of this pin is controlled by DDB4. When the pin is forced to be an input, the pull-up can still be controlled by the PORTB4 bit. See the description of the SPI port for further details.

- **AIN1 - PORTB, Bit 3**

AIN1, Analog Comparator Negative Input. When configured as an input (DDB3 is cleared (zero)) and with the internal MOS pull up resistor switched off (PB3 is cleared (zero)), this pin also serves as the negative input of the on-chip analog comparator. During power down mode, the schmitt trigger of the digital input is disconnected. This allows analog signals which are close to  $V_{CC}/2$  to be present during power down without causing excessive power consumption.

- **AIN0 - PORTB, Bit 2**

AIN0, Analog Comparator Positive Input. When configured as an input (DDB2 is cleared (zero)) and with the internal

MOS pull up resistor switched off (PB2 is cleared (zero)), this pin also serves as the positive input of the on-chip analog comparator. During power down mode, the schmitt trigger of the digital input is disconnected. This allows analog signals which are close to  $V_{CC}/2$  to be present during power down without causing excessive power consumption.

- **T1 - PORTB, Bit 1**

T1, Timer/Counter1 counter source. See the timer description for further details

- **T0 - PORTB, Bit 0**

T0: Timer/Counter0 counter source. See the timer description for further details.

### Port B Schematics

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

**Figure 48.** PORTB Schematic Diagram (Pins PB0 and PB1)

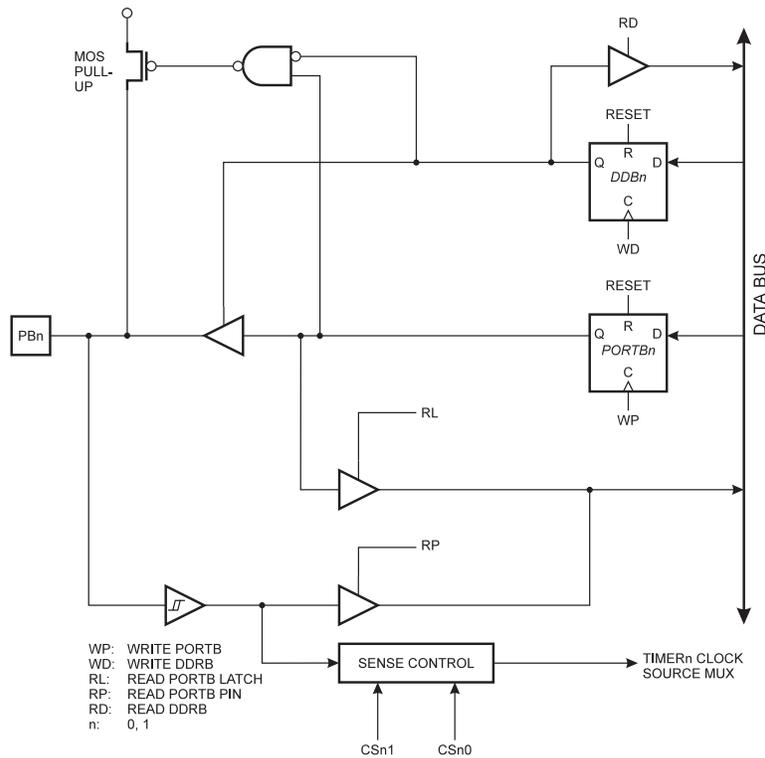


Figure 49. PORTB Schematic Diagram (Pins PB2 and PB3)

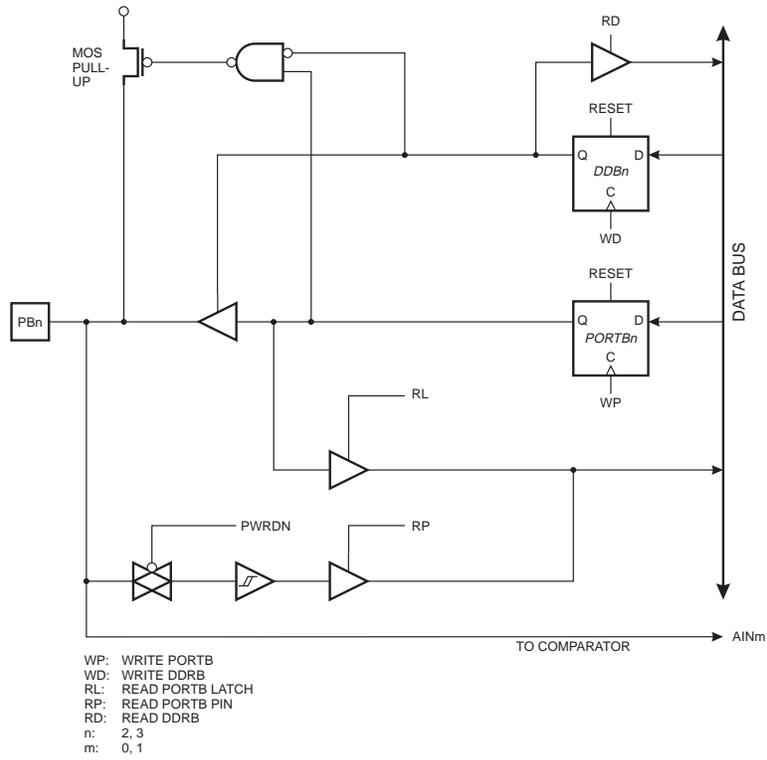
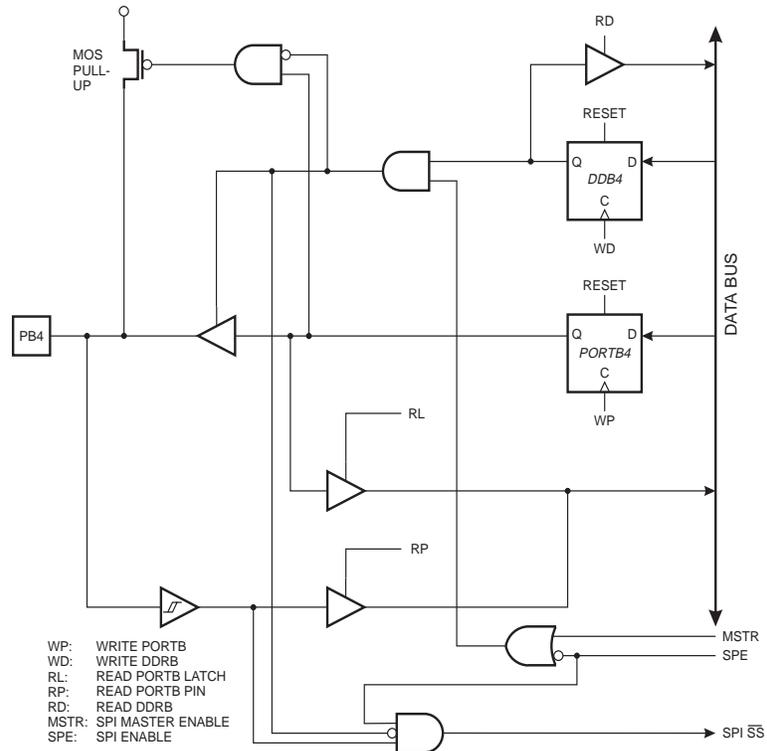
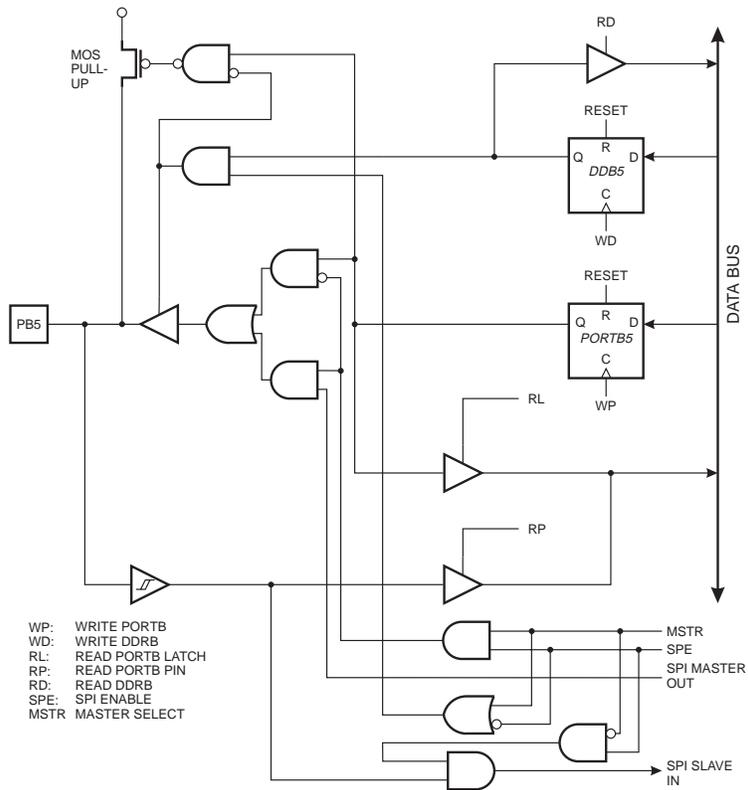


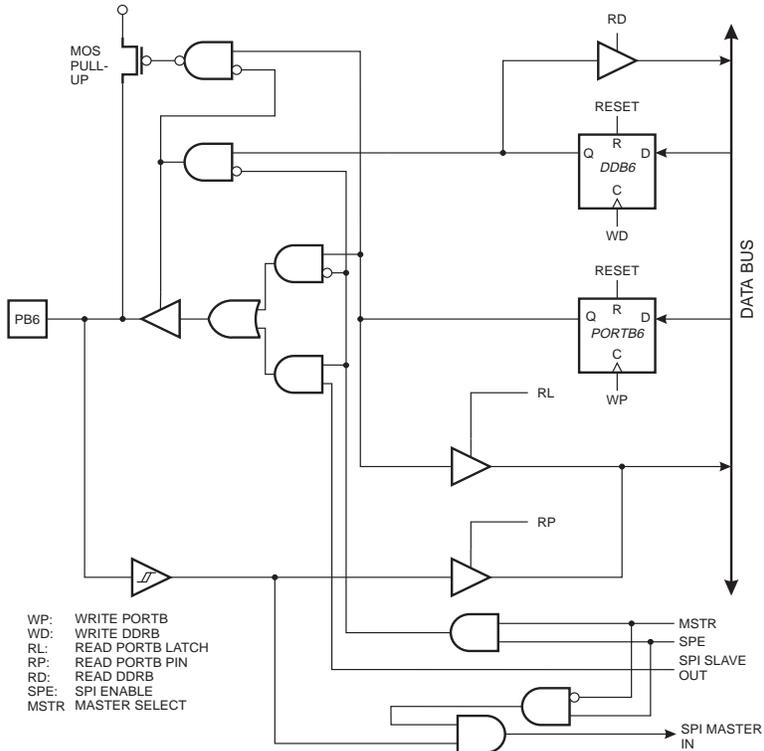
Figure 50. PORTB Schematic Diagram (Pin PB4)



**Figure 51. PORTB Schematic Diagram (Pin PB5)**



**Figure 52. PORTB Schematic Diagram (Pin PB6)**





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 PORTC Data Latch is read, and when reading PINC, the logical values present on the pins are read.

### Port C As General Digital I/O

All 8 bits in PORT C are equal when used as digital I/O pins.

**Table 29.** 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

### Alternate Functions of PORTC

When the AS2 bit in ASSR is set (one) to enable asynchronous clocking of Timer/Counter2, pins PC6 and PC7 are disconnected from the port. In this mode, a crystal oscillator

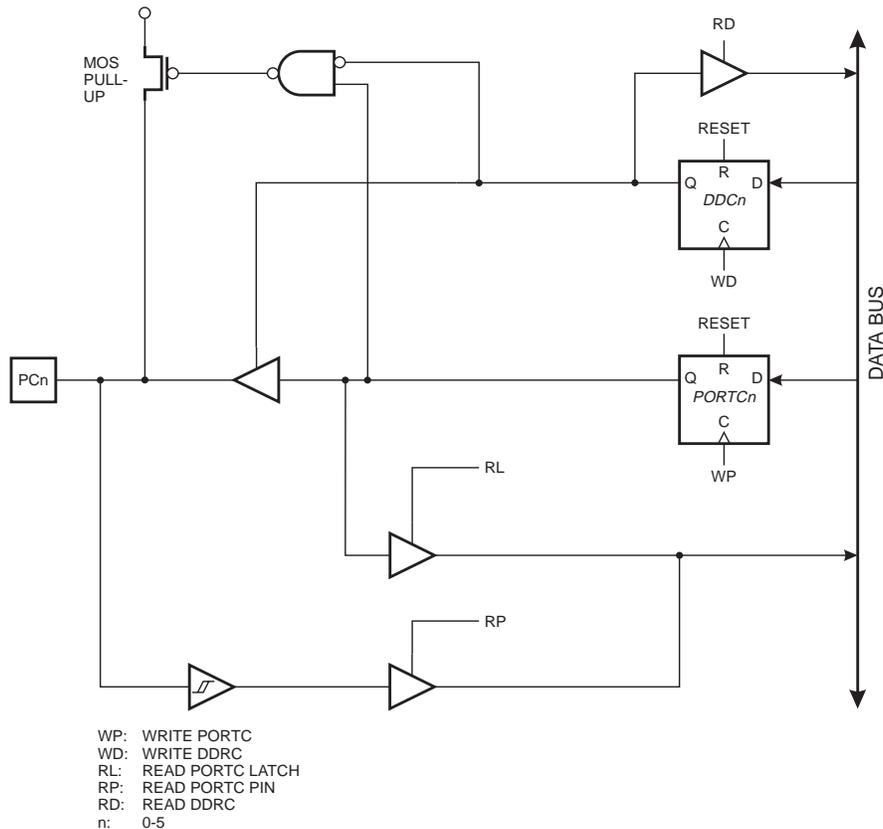
PCn, General I/O pin: The DDCn bit in the DDRC register selects the direction of this pin, if DDCn is set (one), PCn is configured as an output pin. If DDCn is cleared (zero), PCn is configured as an input pin. If PORTCn is set (one) when the pin configured as an input pin, the MOS pull up resistor is 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.

is connected to the pins, and the pins can not be used as I/O pins.

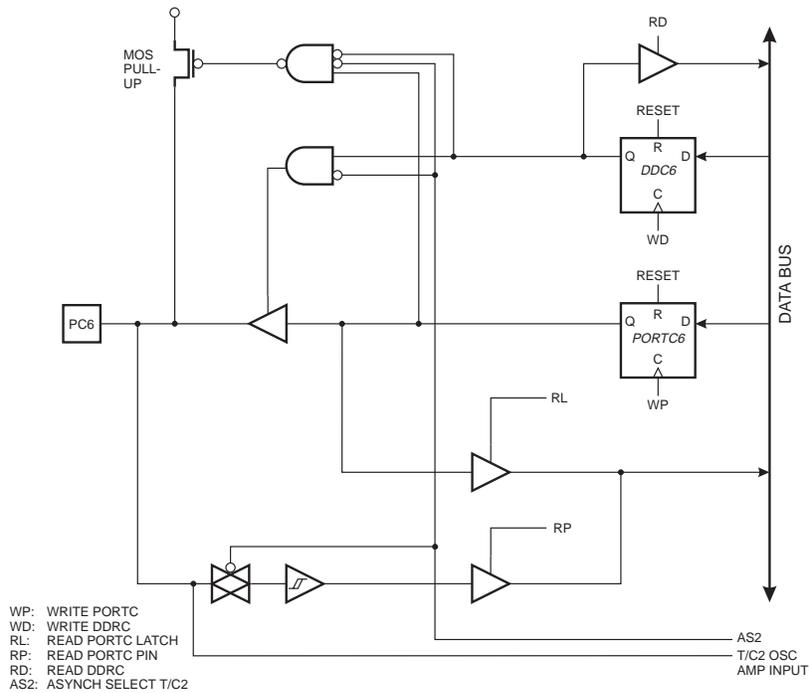
### Port C Schematics

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

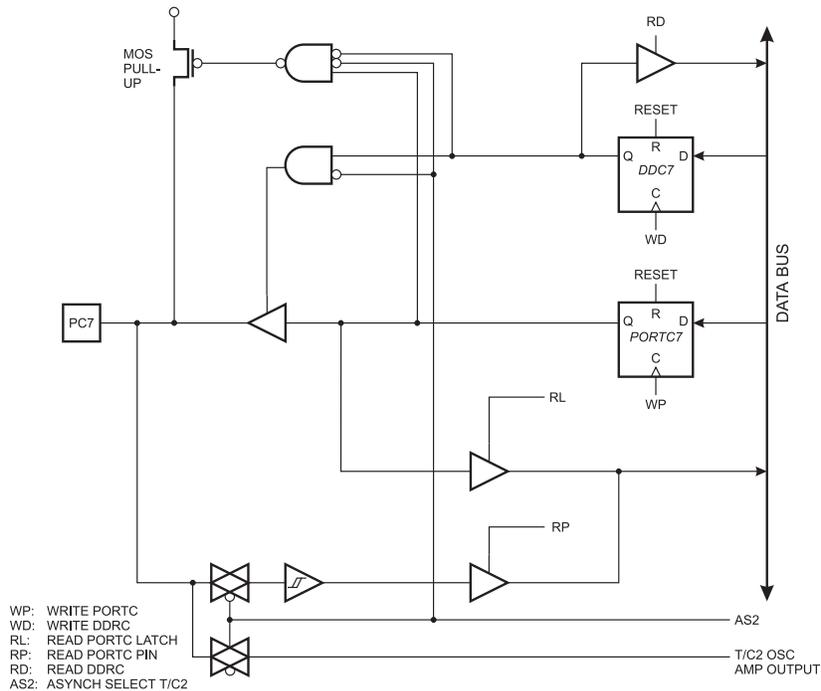
**Figure 54.** PORTC Schematic Diagram (Pins PC0 - PC5)



**Figure 55.** PORTC Schematic Diagram (Pins PC6)



**Figure 56.** PORTC Schematic Diagram (Pins PC7)



## Port D

Port D is an 8 bit bi-directional I/O port with internal pull-up resistors.

Three data memory address locations are allocated for Port D, one each for the Data Register - PORTD, \$12(\$32),

Data Direction Register - DDRD, \$11(\$31) and the Port D Input Pins - PIND, \$10(\$30). The Port D Input Pins address is read only, while the Data Register and the Data Direction Register are read/write.



The Port D output buffers can sink 20 mA. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated.

Some Port D pins have alternate functions as shown in the following table:

**Table 30.** Port D Pins Alternate Functions

Port Pin	Alternate Function
PD0	RDX (UART Input line)
PD1	TDX (UART Output line)
PD2	INT0 (External interrupt 0 input)
PD3	INT1 (External interrupt 1 input)
PD4	OC1B (Timer/Counter1 output compareB match output)
PD5	OC1A (Timer/Counter1 output compareA match output)
PD6	ICP (Timer/Counter1 input capture pin)
PD7	OC2 (Timer/Counter2 output compare match output)

**The Port D Data Register - PORTD**

Bit	7	6	5	4	3	2	1	0	
\$12 (\$32)	<b>PORTD7 PORTD6 PORTD5 PORTD4 PORTD3 PORTD2 PORTD1 PORTD0</b>								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	

**The Port D Data Direction Register - DDRD**

Bit	7	6	5	4	3	2	1	0	
\$11 (\$31)	<b>DDD7 DDD6 DDD5 DDD4 DDD3 DDD2 DDD1 DDD0</b>								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	

**The Port D Input Pins Address - PIND**

Bit	7	6	5	4	3	2	1	0	
\$10 (\$30)	<b>PIND7 PIND6 PIND5 PIND4 PIND3 PIND2 PIND1 PIND0</b>								PIND
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 D Input Pins address - PIND - is not a register, and this address enables access to the physical value on each Port D pin. When reading PORTD, the PORTD 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.

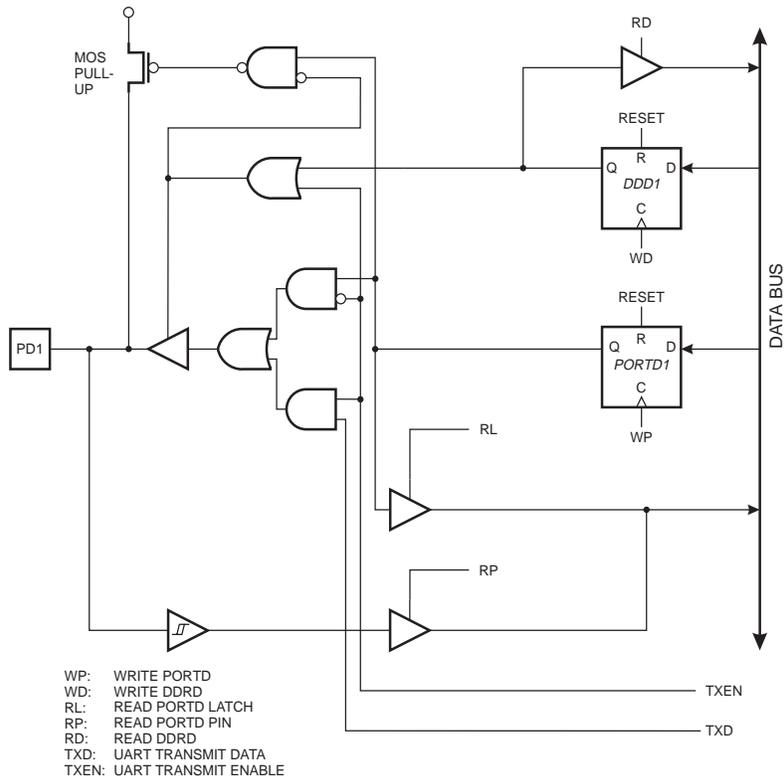
**Table 31.** DDDn Bits on Port D Pins

DDDn	PORTDn	I/O	Pull Up	Comment
0	0	Input	No	Tri-state (Hi-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

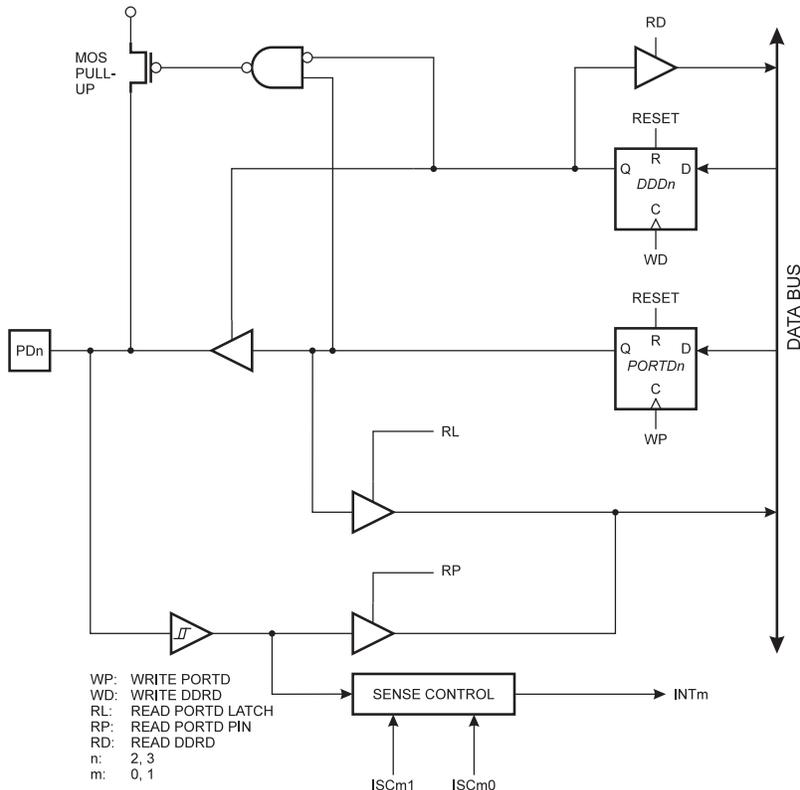
n: 7,6...0, pin number.



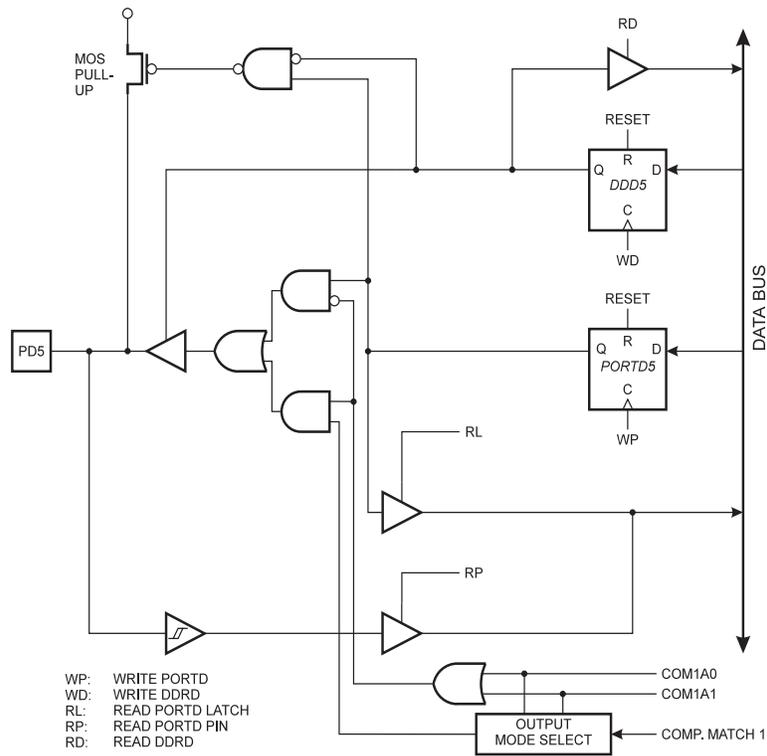
**Figure 58.** PORTD Schematic Diagram (Pin PD1)



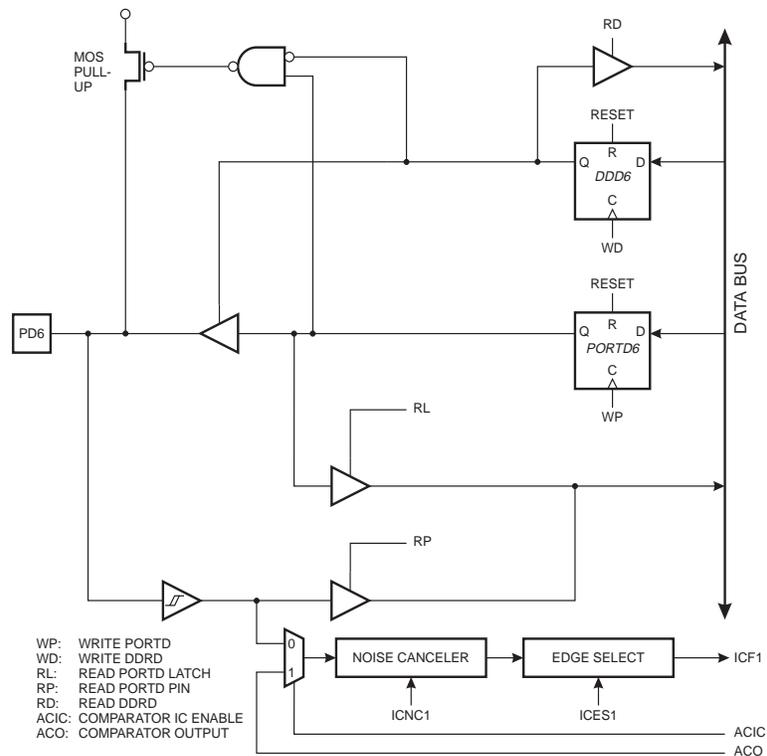
**Figure 59.** PORTD Schematic Diagram (Pins PD2 and PD3)



**Figure 60.** PORTD Schematic Diagram (Pins PD4 and PD5)



**Figure 61.** PORTD Schematic Diagram (Pin PD6)





## Memory Programming

### Program Memory Lock Bits

The AT90S4434/8535 MCU provides two lock bits which can be left unprogrammed ('1') or can be programmed ('0') to obtain the additional features listed in Table 32.

**Table 32.** Lock Bit Protection Modes

Program Lock Bits			Protection Type
Mode	LB1	LB2	
1	1	1	No program lock features
2	0	1	Further programming of the Flash and EEPROM is disabled
3	0	0	Same as mode 2, but verify is also disabled.

Note: The Lock Bits can only be erased with the Chip Erase operation.

### Fuse Bits

The AT90S4434/8535 has two fuse bits, SPIEN and FSTRT.

- When SPIEN is programmed ('0'), Serial Program Downloading is enabled. Default value is programmed ('0'). This bit is not accessible in serial programming mode.
- When FSTRT is programmed ('0'), the short start-up time is selected. This bit is accessible through serial programming.

The status of the fuse bits is not affected by a chip erase.

### Signature Bytes

All Atmel microcontrollers have a three-byte signature code which identifies the device. This code can be read in both serial<sup>(1)</sup> and parallel mode. The three bytes reside in a separate address space.

For the AT90S4434, they are:

1. \$00: \$1E (indicates manufactured by Atmel)
2. \$01: \$92 (indicates 4kB Flash memory)
3. \$02: \$03 (indicates 90S4434 device when \$001 is \$92)

For the AT90S8535, they are:

1. \$00: \$1E (indicates manufactured by Atmel)
2. \$01: \$93 (indicates 8kB Flash memory)
3. \$02: \$03 (indicates 90S8535 device when \$001 is \$93)

Note: 1. When both lock bits are programmed (lock mode 3), the signature bytes can not be read in serial mode.

### Programming the Flash and EEPROM

Atmel's AT90S4434/8535 offers 4K/8K bytes of in-system reprogrammable Flash Program memory and 256/512 bytes of EEPROM Data memory.

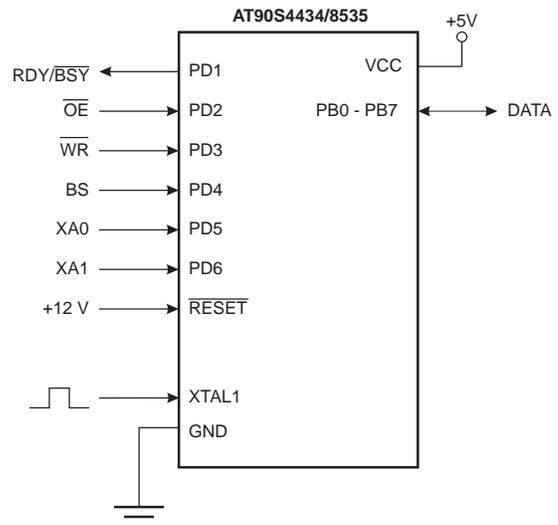
The AT90S4434/8535 is normally shipped with the on-chip Flash Program and EEPROM Data memory arrays in the erased state (i.e. contents = \$FF) and ready to be programmed. This device supports a High-Voltage (12V) Parallel programming mode and a Low-Voltage Serial programming mode. The +12V is used for programming enable only, and no current of significance is drawn by this pin. The serial programming mode provides a convenient way to download the Program and Data into the AT90S4434/8535 inside the user's system.

The Program and Data memory arrays on the AT90S4434/8535 are programmed byte-by-byte in either programming modes. For the EEPROM, an auto-erase cycle is provided with the self-timed programming operation in the serial programming mode.

### Parallel Programming

This section describes how to parallel program and verify Flash Program memory, EEPROM Data memory + Program Memory Lock bits and Fuse bits in the AT90S4434/8535.

**Figure 63.** Parallel Programming



## Signal Names

In this section, some pins of the AT908535 are referenced by signal names describing their functionality during parallel programming rather than their pin names. Pins not described in the following table are referenced by pin names.

**Table 33.** 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
XA0	PD5	I	XTAL Action Bit 0
XA1	PD6	I	XTAL Action Bit 1

The XA1/XA0 bits determine the action taken when the XTAL1 pin is given a positive pulse. The bit settings are shown in the following table:

**Table 34.** XA1 and XA0 Coding

XA1	XA0	Action when XTAL1 is Pulsed
0	0	Load Flash or EEPROM Address (High or Low address byte for Flash 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

When pulsing  $\overline{\text{WR}}$  or  $\overline{\text{OE}}$ , the command loaded determines the action on input or output. The command is a byte where the different bits are assigned functions as shown in the following table:

**Table 35.** Command Byte Bit Coding

Bit#	Meaning when Set
7	Chip Erase
6	Write Fuse Bits. Located in the data byte at the following bit positions: D5: SPIEN Fuse, D0: FSTRT Fuse (Note: Write '0' to program, '1' to erase)
5	Write Lock Bits. Located in the data byte at the following bit positions: D1: LB1, D0: LB2 (Note: write '0' to program)
4	Write Flash or EEPROM (determined by bit 0)
3	Read signature row
2	Read Lock and Fuse Bits. Located in the data byte at the following bits positions: D7: LB1, D6: LB2, D5: SPIEN Fuse, D0: FSTRT Fuse (Note: '0' means programmed)
1	Read from Flash or EEPROM (determined by bit 0)
0	0: Flash Access, 1: EEPROM Access

## Enter Programming Mode

The following algorithm puts the device in parallel programming mode:

1. Apply 4.5 - 5.5 V between VCC and GND.
2. Set  $\overline{\text{RESET}}$  and BS pins 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.

## Chip Erase

The chip erase will erase the Flash and EEPROM memories plus Lock bits. The lock bits are not reset until the program memory has been completely erased. The Fuse bits are not changed. A chip erase must be performed before the Flash is programmed.

Load Command "Chip Erase"

1. Set XA1, XA0 to '10'. This enables command loading.
2. Set BS to '0'.
3. Set PB(7:0) to '1000 0000'. This is the command for Chip erase.
4. Give XTAL1 a positive pulse. This loads the command, and starts the erase of the Flash and EEPROM arrays. After pulsing XTAL1, give  $\overline{\text{WR}}$  a negative pulse to enable lock bit erase at the end of the erase cycle, then wait for at least 10 ms. Chip erase does not generate any activity on the RDY/ $\overline{\text{BSY}}$  pin.

## Programming The Flash

Load Command "Program Flash"

1. Set XA1, XA0 to '10'. This enables command loading.
2. Set BS to '0'
3. Set PB(7:0) to '0001 0000'. This is the command for Flash programming.
4. Give XTAL1 a positive pulse. This loads the command.

Load Address Low byte

1. Set XA1, XA0 to '00'. This enables address loading.
2. Set BS to '0'. This selects Low address.
3. Set PB(7:0) = Address Low byte (\$00 - \$FF)
4. Give XTAL1 a positive pulse. This loads the Address Low byte.

Load Address High byte

1. Set XA1, XA0 to '00'. This enables address loading.
2. Set BS to '1'. This selects High address.
3. Set PB(7:0) = Address High byte (\$00 - \$07/\$0F)

4. Give XTAL1 a positive pulse. This loads the Address High byte.

Load Data byte

1. Set XA1, XA0 to '01'. This enables data loading.
2. Set PB(7:0) = Data Low byte (\$00 - \$FF)
3. Give XTAL1 a positive pulse. This loads the Data byte.

Write Data Low byte

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

Load Data byte

1. Set XA1, XA0 to '01'. This enables data loading.
2. Set PB(7:0) = Data High byte (\$00 - \$FF)
3. Give XTAL1 a positive pulse. This loads the Data byte.

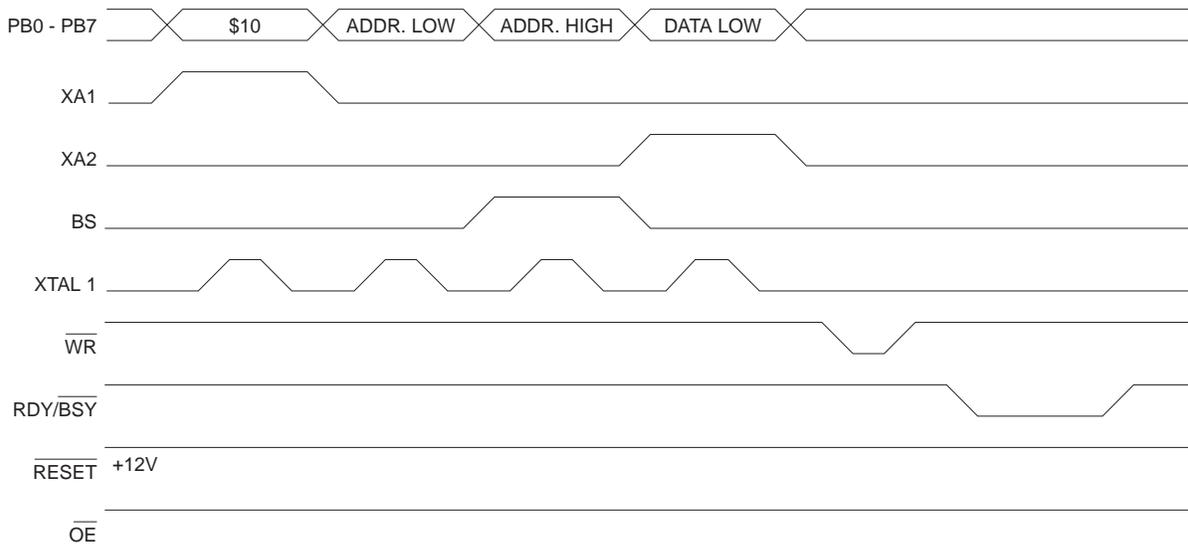
Write Data High byte

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

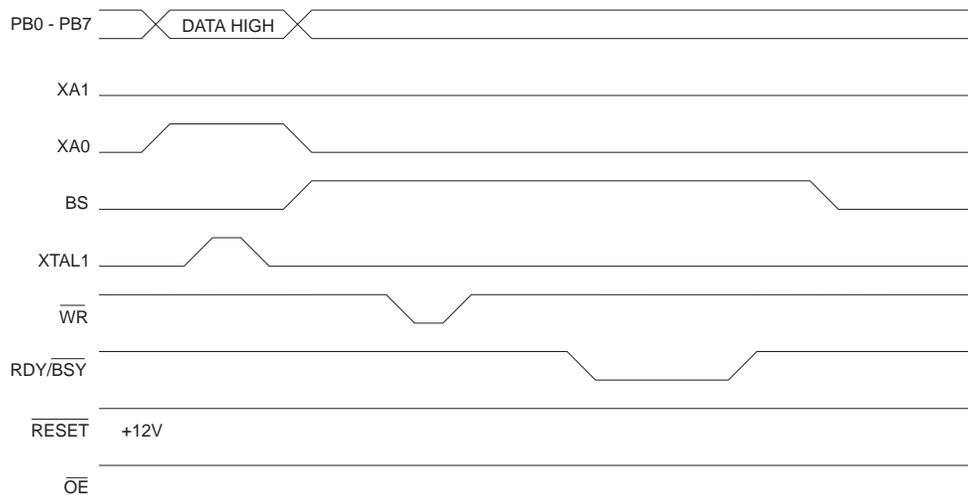
The loaded command and address are retained in the device during programming. To simplify programming, the following should be considered.

- The command for Flash programming needs only be loaded before programming of the first byte.
- Address High byte needs only be loaded before programming a new 256 word page in the Flash.

**Figure 64.** Programming Flash Low Byte



**Figure 65.** Programming Flash High Byte



### Programming The EEPROM

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

1. Load Command '0001 0001'.
2. Load Low EEPROM Address (\$00 - \$FF)
3. Load High EEPROM Address (\$00 - \$01). Only needed for AT90S8535.
4. Load Low EEPROM Data (\$00 - \$FF)
5. Give  $\overline{WR}$  a negative pulse and wait for  $\overline{RDY/BSY}$  to go high.

The Command needs only be loaded before programming the first byte.

### Reading The Flash

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

1. Load Command '0000 0010'.
2. Load Low Address (\$00 - \$FF)
3. Load High Address (\$00 - \$07/\$0F)
4. Set  $\overline{OE}$  to '0', and BS to '0'. The Low Data byte can now be read at PB(7:0)
5. Set BS to '1'. The High Data byte can now be read from PB(7:0)
6. Set  $\overline{OE}$  to '1'.

The Command needs only be loaded before reading the first byte.

## Reading The EEPROM

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

1. Load Command '0000 0011'.
2. Load Low EEPROM Address (\$00 - \$FF)
3. Load High EEPROM Address (\$00 - \$01). Only needed for AT90S8535.
4. Set  $\overline{OE}$  to '0', and BS to '0'. The EEPROM Data byte can now be read at PB(7:0)
5. Set  $\overline{OE}$  to '1'.

The Command needs only be loaded before reading the first byte.

## Programming The Fuse Bits

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

1. Load Command '0100 0000'.
2. Load Data.  
Bit 5 = '0' programs the SPIEN Fuse bit. Bit 5 = '1' erases the SPIEN Fuse bit.  
Bit 0 = '0' programs the FSTRT fuse bit. Bit 5 = '1' erases the FSTRT fuse bit.
3. Give  $\overline{WR}$  a negative pulse, 1 ms wide.

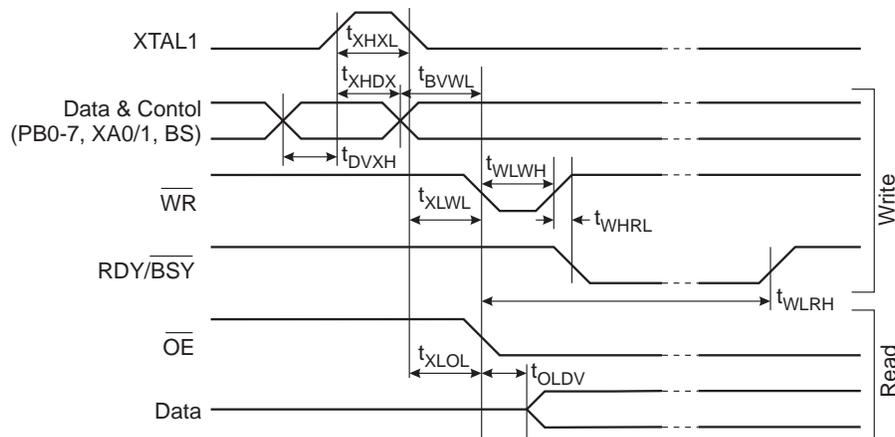
## Programming The Lock Bits

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

1. Load Command '0010 0000'.
2. Load Data.  
Bit 2 = '0' programs Lock Bit2  
Bit 1 = '0' programs Lock Bit1

## Parallel Programming Characteristics

Figure 66. Parallel Programming Timing



3. Give  $\overline{WR}$  a negative pulse and wait for RDY/BSY to go high.

The lock bits can only be cleared by executing a chip erase.

## Reading The Fuse And Lock Bits

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

1. Load Command '0000 0100'.
2. Set  $\overline{OE}$  to '0', and BS to '1'. The Status of Fuse and Lock bits can now be read at PB(7:0)  
Bit 7: Lock Bit1 ('0' means programmed)  
Bit 6: Lock Bit2 ('0' means programmed)  
Bit 5: SPIEN Fuse ('0' means programmed)  
Bit 0: FSTRT Fuse ('0' means programmed)
3. Set  $\overline{OE}$  to '1'.

Observe especially that BS needs to be set to '1'.

## Reading The Signature Bytes

The algorithm for reading the Signature Bytes bits is as follows (refer to Flash Programming for details on Command, Address and Data loading):

1. Load Command '0000 1000'.
2. Load Low address (\$00 - \$02)
3. Set  $\overline{OE}$  to '0', and BS to '0'. The Selected Signature byte can now be read at PB(7:0)
4. Set  $\overline{OE}$  to '1'.

The command needs only be programmed before reading the first byte.

**Table 36.** Parallel Programming Characteristics

$T_A = 21^\circ\text{C}$  to  $27^\circ\text{C}$ ,  $V_{CC} = 4.5 - 5.5\text{V}$

Symbol	Parameter	Min	Typ	Max	Units
$t_{DVXH}$	Data and Control Setup before XTAL1 High	67			ns
$t_{XHXL}$	XTAL1 Pulse Width High	67			ns
$t_{XLDH}$	Data and Control Hold after XTAL1 High	67			ns
$t_{BVWL}$	BS Valid to $\overline{WR}$ Low	67			ns
$t_{WLWH}$	$\overline{WR}$ Pulse Width Low	67			ns
$t_{WHRL}$	$\overline{WR}$ High to $\text{RDY}/\overline{\text{BSY}}$ Low <sup>(1)</sup>		20		ns
$t_{XL0L}$	XTAL1 Low to $\overline{OE}$ Low	67			ns
$t_{OLDV}$	$\overline{OE}$ Low to Data Valid		20		ns
$t_{WLRH}$	$\overline{WR}$ Low to $\text{RDY}/\overline{\text{BSY}}$ High <sup>(1)</sup>	0.5	0.7	0.9	ms

Note: 1. If  $t_{WPWL}$  is held longer than  $t_{WLRH}$ , no  $\text{RDY}/\overline{\text{BSY}}$  pulse will be seen.

### Serial Downloading

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

When programming the EEPROM, an auto-erase cycle is built into the self-timed programming operation (in the serial mode ONLY) and there is no need to first execute the Chip Erase instruction. The Chip Erase operation turns the 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 for Program memory and \$0000 to \$00FF/\$01FF for EEPROM memory.

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

Low: > 2 XTAL1 clock cycles

High: > 2 XTAL1 clock cycles

### Data Polling

When a new byte has been written and is being programmed into the Flash or EEPROM, reading the address location being programmed will give the value \$FF. At the time the device is ready for a new byte, the programmed value will read correctly. This is used to determine when the next byte can be written. This will not work for the value \$FF, but the user should have the following in mind: As a chip-erased device contains \$FF in all locations, programming of addresses that are meant to contain \$FF, can be

skipped. This does not apply if the EEPROM is re-programmed without chip-erasing the device. In this case, data polling cannot be used for the value \$FF, and the user will have to wait at least 4ms before programming the next byte.

### Serial Programming Algorithm

To program and verify the AT90S4434/8535 in the serial programming mode, the following sequence is recommended (See four byte instruction formats in Table 37):

1. Power-up sequence:
  - Apply power between VCC and GND while  $\overline{\text{RESET}}$  and SCK are set to '0'. If a crystal is not connected across pins XTAL1 and XTAL2, apply a clock signal to the XTAL1 pin. In some systems, the programmer can not guarantee that SCK is held low during power-up. In this case,  $\overline{\text{RESET}}$  must be given a positive pulse of at least two XTAL1 cycles duration after SCK has been set to '0'.
2. Wait for at least 20 ms and enable serial programming by sending the Programming Enable serial instruction to pin MOSI/PB5.
3. When issuing the third byte in Programming Enable, the value sent as byte number two (\$53), will echo back during transmission of byte number three. In any case, all four bytes in programming enable must be transmitted. If the \$53 did not echo back, give SCK a positive pulse and issue a new Programming Enable command. If the \$53 is not seen within 32 attempts, there is no functional device connected.
4. If a chip erase is performed (must be done to erase the Flash), wait 10 ms, give  $\overline{\text{RESET}}$  a positive pulse, and start over from Step 2.

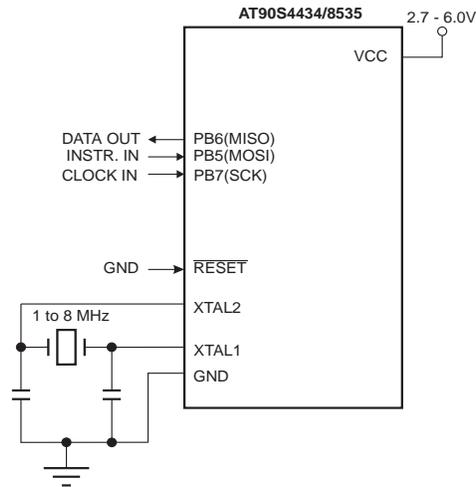
5. The Flash or EEPROM array is programmed one byte at a time by supplying the address and data together with the appropriate Write instruction. An EEPROM memory location is first automatically erased before new data is written. Use Data Polling to detect when the next byte in the Flash or EEPROM can be written. In a chip erased device, no \$FFs in the data file(s) need to be programmed.
6. Any memory location can be verified by using the Read instruction which returns the content at the selected address at serial output MISO/PB6.
7. At the end of the programming session,  $\overline{\text{RESET}}$  can be set high to commence normal operation.
8. Power-off sequence (if needed):  
Set XTAL1 to '0' (if a crystal is not used).  
Set  $\overline{\text{RESET}}$  to '1'.  
Turn  $V_{CC}$  power off

**Table 37.** Serial Programming Instruction Set

Instruction	Instruction Format				Operation
	Byte 1	Byte 2	Byte 3	Byte4	
Programming Enable	1010 1100	0101 0011	xxxx xxxx	xxxx xxxx	Enable Serial Programming after $\overline{\text{RESET}}$ goes low.
Chip Erase	1010 1100	100x xxxx	xxxx xxxx	xxxx xxxx	Chip erase EEPROM and Flash
Read Program Memory	0010 H000	xxxx aaaa	bbbb bbbb	oooo oooo	Read H(high or low) data o from Program memory at word address a:b
Write Program Memory	0100 H000	xxxx aaaa	bbbb bbbb	iiii iiiii	Write H(high or low) data i to Program memory at word address a:b
Read EEPROM Memory	1010 0000	xxxx xxxa	bbbb bbbb	oooo oooo	Read data o from EEPROM memory at address a:b
Write EEPROM Memory	1100 0000	xxxx xxxa	bbbb bbbb	iiii iiiii	Write data i to EEPROM memory at address a:b
Read Lock and Fuse Bits	0101 1000	xxxx xxxx	xxxx xxxx	12Sx xxxF	Read lock and fuse bits. '0': Programmed, '1': Unprogrammed
Write Lock Bits	1010 1100	111x x21x	xxxx xxxx	xxxx xxxx	Write lock bits. Set bits 1,2='0' to program lock bits.
Read Signature Byte	0011 0000	xxxx xxxx	xxxx xxbb	oooo oooo	Read Signature Byte o at address b
Write FSTRT Bit	1010 1100	101x xxxF	xxxx xxxx	xxxx xxxx	Write FSTRT fuse. Set bit F='0' to program fuse, '1' to unprogram

Note: a = address high bits  
b = address low bits  
H = 0 - Low byte, 1 - High Byte  
o = data out  
i = data in  
x = don't care  
1 = lock bit 1  
2 = lock bit 2  
F = FSTRT fuse  
S = SPIEN fuse

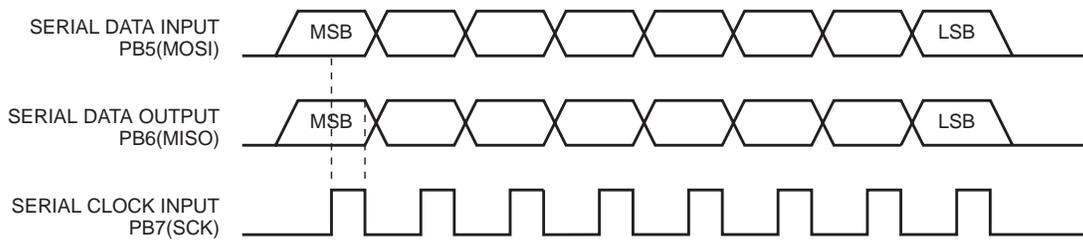
**Figure 67.** Serial Programming and Verify



When writing serial data to the AT90S4434/8535, data is clocked on the rising edge of SCK.

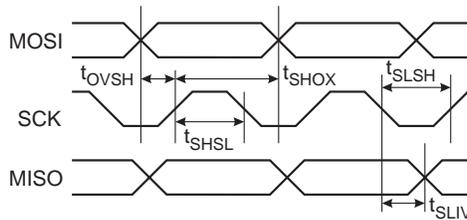
When reading data from the AT90S4434/8535, data is clocked on the falling edge of SCK. See Figure 68 for an explanation.

**Figure 68.** Serial Programming Waveforms



### Serial Programming Characteristics

**Figure 69.** Serial Programming Timing



**Table 38.** Serial Programming Characteristics

$T_A = -40^{\circ}\text{C}$  to  $85^{\circ}\text{C}$ ,  $V_{CC} = 2.7 - 6.0\text{V}$  (Unless otherwise noted)

Symbol	Parameter	Min	Typ	Max	Units
$1/t_{CLCL}$	Oscillator Frequency ( $V_{CC} = 2.7 - 4.0\text{V}$ )	0		4	MHz
$t_{CLCL}$	Oscillator Period ( $V_{CC} = 2.7 - 4.0\text{V}$ )	250			ns
$1/t_{CLCL}$	Oscillator Frequency ( $V_{CC} = 4.0 - 6.0\text{V}$ )	0		8	MHz
$t_{CLCL}$	Oscillator Period ( $V_{CC} = 4.0 - 6.0\text{V}$ )	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

## Absolute Maximum Ratings\*

Operating Temperature .....	$-40^{\circ}\text{C}$ to $+105^{\circ}\text{C}$
Storage Temperature .....	$-65^{\circ}\text{C}$ to $+150^{\circ}\text{C}$
Voltage on any Pin except $\overline{\text{RESET}}$ with respect to Ground .....	$-1.0\text{V}$ to $+7.0\text{V}$
Maximum Operating Voltage .....	6.6V
I/O Pin Maximum Current .....	40.0 mA
Maximum Current VCC and GND.....	140.0 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

$T_A = -40^{\circ}\text{C}$  to  $85^{\circ}\text{C}$ ,  $V_{CC} = 2.7\text{V}$  to  $6.0\text{V}$  (unless otherwise noted)

Symbol	Parameter	Condition	Min	Typ	Max	Units
$V_{IL}$	Input Low Voltage		-0.5		$0.3V_{CC}$	V
$V_{IL1}$	Input Low Voltage	XTAL	-0.5		$0.2 V_{CC}$	V
$V_{IH}$	Input High Voltage	Except (XTAL, RESET)	$0.6 V_{CC}$		$V_{CC} + 0.5$	V
$V_{IH1}$	Input High Voltage	XTAL	$0.8 V_{CC}$		$V_{CC} + 0.5$	V
$V_{IH2}$	Input High Voltage	RESET	$V_{CC}$		$V_{CC} + 0.5$	V
$V_{OL}$	Output Low Voltage (Ports A, B, C, D)	$I_{OL} = 20\text{ mA}$ , $V_{CC} = 5\text{V}$			0.6	V
		$I_{OL} = 10\text{ mA}$ , $V_{CC} = 3\text{V}$			0.5	V
$V_{OH}$	Output High Voltage (Ports A, B, C, D)	$I_{OH} = 3\text{ mA}$ , $V_{CC} = 5\text{V}$	4.2			V
		$I_{OH} = 1.5\text{ mA}$ , $V_{CC} = 3\text{V}$	2.3			V
$I_{IL}$	Input Leakage (Ports A, B, C, D)	$V_{CC} = 6\text{V}$ , pin low	-8.0		8.0	$\mu\text{A}$
$I_{IH}$	Input Leakage Current I/O pin	$V_{CC} = 6\text{V}$ , pin high	-8.0		8.0	$\mu\text{A}$
RRST	Reset Pull-Up		100		500	$\text{K}\Omega$
$R_{I/O}$	I/O Pin Pull-Up Resistor		35		120	$\text{K}\Omega$
$I_{CC}$	Power Supply Current	Active 4MHz, 3 $V_{CC}$			3.0	mA
		Idle 4MHz, 3 $V_{CC}$		1.0	1.2	mA
		Power Down 4MHz, 3 $V_{CC}$ WDT enabled		8.5	15.0	$\mu\text{A}$
		Power Down 4 MHz, 3 $V_{CC}$ WDT disabled		<1	2.0	$\mu\text{A}$
$V_{ACIO}$	Analog Comparator Input Offset Voltage	$V_{CC} = 5\text{V}$			40	mV
$I_{ACLK}$	Analog Comparator Input Leakage A	$V_{CC} = 5\text{V}$ $V_{in} = V_{CC}/2$	-50		50	nA
$t_{ACPD}$	Analog Comparator Propagation Delay	$V_{CC} = 2.7\text{V}$		750		ns
		$V_{CC} = 4.0\text{V}$		500		

Notes: 1. Under steady state (non-transient) conditions,  $I_{OL}$  must be externally limited as follows:

Maximum  $I_{OL}$  per port pin: 20 mA

Maximum total  $I_{OL}$  for all output pins: 80 mA

Port A: 26 mA

Ports A, B, D: 15 mA

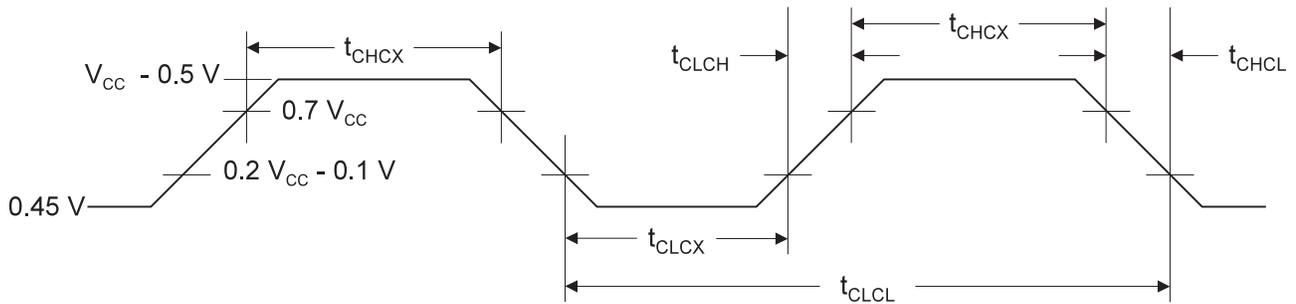
Maximum total  $I_{OL}$  for all output pins: 70 mA

If  $I_{OL}$  exceeds the test condition,  $V_{OL}$  may exceed the related specification.

Pins are not guaranteed to sink current greater than the listed test conditions.

2. Minimum  $V_{CC}$  for Power Down is 2V.

## External Clock Drive Waveforms



## External Clock Drive

Symbol	Parameter	$V_{CC} = 2.7V \text{ to } 4.0V$		$V_{CC} = 4.0V \text{ to } 6.0V$		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	115		58.3		ns
$t_{CLCX}$	Low Time	115		58.3		ns
$t_{CLCH}$	Rise Time		10		4.15	ns
$t_{CHCL}$	Fall Time		10		4.15	ns



## AT90S4434/8535 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	21
\$3E (\$5E)	SPH	-	-	-	-	-	-	SP9	SP8	22
\$3D (\$5D)	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	22
\$3C (\$5C)	Reserved									
\$3B (\$5B)	GIMSK	INT1	INT0	-	-	-	-	-	-	28
\$3A (\$5A)	GIFR	INTF1	INTF0							28
\$39 (\$59)	TIMSK	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	-	TOIE0	29
\$38 (\$58)	TIFR	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	-	TOV0	30
\$37 (\$57)	Reserved									
\$36 (\$56)	Reserved									
\$35 (\$55)	MCUCR	-	SE	SM1	SM0	ISC11	ISC10	ISC01	ISC00	31
\$34 (\$54)	MCUSR	-	-	-	-	-	-	EXTRF	PORF	27
\$33 (\$53)	TCCR0	-	-	-	-	-	CS02	CS01	CS00	35
\$32 (\$52)	TCNT0	Timer/Counter0 (8 Bits)								36
\$31 (\$51)	Reserved									
\$30 (\$50)	Reserved									
\$2F (\$4F)	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	-	-	PWM11	PWM10	38
\$2E (\$4E)	TCCR1B	ICNC1	ICES1	-	-	CTC1	CS12	CS11	CS10	39
\$2D (\$4D)	TCNT1H	Timer/Counter1 - Counter Register High Byte								40
\$2C (\$4C)	TCNT1L	Timer/Counter1 - Counter Register Low Byte								40
\$2B (\$4B)	OCR1AH	Timer/Counter1 - Output Compare Register A High Byte								41
\$2A (\$4A)	OCR1AL	Timer/Counter1 - Output Compare Register A Low Byte								41
\$29 (\$49)	OCR1BH	Timer/Counter1 - Output Compare Register B High Byte								41
\$28 (\$48)	OCR1BL	Timer/Counter1 - Output Compare Register B Low Byte								41
\$27 (\$47)	ICR1H	Timer/Counter1 - Input Capture Register High Byte								41
\$26 (\$46)	ICR1L	Timer/Counter1 - Input Capture Register Low Byte								41
\$25 (\$45)	TCCR2	-	PWM2	COM21	COM20	CTC2	CS22	CS21	CS20	45
\$24 (\$44)	TCNT2	Timer/Counter2 (8 Bits)								46
\$23 (\$43)	OCR2	Timer/Counter2 Output Compare Register								46
\$22 (\$42)	ASSR	-	-	-	-	AS2	TCN2UB	OCR2UB	TCR2UB	48
\$21 (\$41)	WDTCR	-	-	-	WDTOE	WDE	WDP2	WDP1	WDP0	50
\$20 (\$40)	Reserved									
\$1F (\$3F)	EEARH								EEAR9	
\$1E (\$3E)	EEARL	EEAR7	EEAR6	EEAR5	EEAR4	EEAR3	EEAR2	EEAR1	EEAR0	51
\$1D (\$3D)	EEDR	EEPROM Data Register								51
\$1C (\$3C)	EECR	-	-	-	-	EERIE	EEMWE	EEWE	EERE	52
\$1B (\$3B)	PORTA	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	70
\$1A (\$3A)	DDRA	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	70
\$19 (\$39)	PINA	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	70
\$18 (\$38)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	72
\$17 (\$37)	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	72
\$16 (\$36)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	72
\$15 (\$35)	PORTC	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	78
\$14 (\$34)	DDRC	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	78
\$13 (\$33)	PINC	PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	78
\$12 (\$32)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	81
\$11 (\$31)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	81
\$10 (\$30)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	81
\$0F (\$2F)	SPDR	SPI Data Register								57
\$0E (\$2E)	SPSR	SPIF	WCOL	-	-	-	-	-	-	56
\$0D (\$2D)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	56
\$0C (\$2C)	UDR	UART I/O Data Register								60
\$0B (\$2B)	USR	RXC	TXC	UDRE	FE	OR	-	-	-	60
\$0A (\$2A)	UCR	RXCIE	TXCIE	UDRIE	RXEN	TXEN	CHR9	RXB8	TXB8	61
\$09 (\$29)	UBRR	UART Baud Rate Register								
\$08 (\$28)	ACSR	ACD	-	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	67
\$07 (\$27)	ADMUX	-	-	-	-	-	MUX2	MUX1	MUX0	67
\$06 (\$26)	ADCSR	ADEN	ADSC	ADFR	ADIF	ADIE	ADPS2	ADPS1	ADPS0	67
\$05 (\$25)	ADCH	-	-	-	-	-	-	ADC9	ADC8	68
\$04 (\$24)	ADCL	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0	68
\$03 (\$20)	Reserved									
\$02 (\$22)	Reserved									
\$01 (\$21)	Reserved									
\$00 (\$20)	Reserved									

## AT90S4434/8535 Instruction Set Summary

Mnemonics	Operands	Description	Operation	Flags	#Clocks
<b>ARITHMETIC AND LOGIC INSTRUCTIONS</b>					
ADD	Rd, Rr	Add two Registers	$Rd \leftarrow Rd + Rr$	Z,C,N,V,H	1
ADC	Rd, Rr	Add with Carry two Registers	$Rd \leftarrow Rd + Rr + C$	Z,C,N,V,H	1
ADIW	Rdl,K	Add Immediate to Word	$Rdh:Rdl \leftarrow Rdh:Rdl + K$	Z,C,N,V,S	2
SUB	Rd, Rr	Subtract two Registers	$Rd \leftarrow Rd - Rr$	Z,C,N,V,H	1
SUBI	Rd, K	Subtract Constant from Register	$Rd \leftarrow Rd - K$	Z,C,N,V,H	1
SBC	Rd, Rr	Subtract with Carry two Registers	$Rd \leftarrow Rd - Rr - C$	Z,C,N,V,H	1
SBCI	Rd, K	Subtract with Carry Constant from Reg.	$Rd \leftarrow Rd - K - C$	Z,C,N,V,H	1
SBIW	Rdl,K	Subtract Immediate from Word	$Rdh:Rdl \leftarrow Rdh:Rdl - K$	Z,C,N,V,S	2
AND	Rd, Rr	Logical AND Registers	$Rd \leftarrow Rd \bullet Rr$	Z,N,V	1
ANDI	Rd, K	Logical AND Register and Constant	$Rd \leftarrow Rd \bullet K$	Z,N,V	1
OR	Rd, Rr	Logical OR Registers	$Rd \leftarrow Rd \vee Rr$	Z,N,V	1
ORI	Rd, K	Logical OR Register and Constant	$Rd \leftarrow Rd \vee K$	Z,N,V	1
EOR	Rd, Rr	Exclusive OR Registers	$Rd \leftarrow Rd \oplus Rr$	Z,N,V	1
COM	Rd	One's Complement	$Rd \leftarrow \$FF - Rd$	Z,C,N,V	1
NEG	Rd	Two's Complement	$Rd \leftarrow \$00 - Rd$	Z,C,N,V,H	1
SBR	Rd,K	Set Bit(s) in Register	$Rd \leftarrow Rd \vee K$	Z,N,V	1
CBR	Rd,K	Clear Bit(s) in Register	$Rd \leftarrow Rd \bullet (\$FF - K)$	Z,N,V	1
INC	Rd	Increment	$Rd \leftarrow Rd + 1$	Z,N,V	1
DEC	Rd	Decrement	$Rd \leftarrow Rd - 1$	Z,N,V	1
TST	Rd	Test for Zero or Minus	$Rd \leftarrow Rd \bullet Rd$	Z,N,V	1
CLR	Rd	Clear Register	$Rd \leftarrow Rd \oplus Rd$	Z,N,V	1
SER	Rd	Set Register	$Rd \leftarrow \$FF$	None	1
<b>BRANCH INSTRUCTIONS</b>					
RJMP	k	Relative Jump	$PC \leftarrow PC + k + 1$	None	2
IJMP		Indirect Jump to (Z)	$PC \leftarrow Z$	None	2
RCALL	k	Relative Subroutine Call	$PC \leftarrow PC + k + 1$	None	3
ICALL		Indirect Call to (Z)	$PC \leftarrow Z$	None	3
RET		Subroutine Return	$PC \leftarrow STACK$	None	4
RETI		Interrupt Return	$PC \leftarrow STACK$	I	4
CPSE	Rd,Rr	Compare, Skip if Equal	if (Rd = Rr) $PC \leftarrow PC + 2$ or 3	None	1 / 2
CP	Rd,Rr	Compare	$Rd - Rr$	Z, N,V,C,H	1
CPC	Rd,Rr	Compare with Carry	$Rd - Rr - C$	Z, N,V,C,H	1
CPI	Rd,K	Compare Register with Immediate	$Rd - K$	Z, N,V,C,H	1
SBRC	Rr, b	Skip if Bit in Register Cleared	if (Rr(b)=0) $PC \leftarrow PC + 2$ or 3	None	1 / 2
SBRs	Rr, b	Skip if Bit in Register is Set	if (Rr(b)=1) $PC \leftarrow PC + 2$ or 3	None	1 / 2
SBIC	P, b	Skip if Bit in I/O Register Cleared	if (P(b)=0) $PC \leftarrow PC + 2$ or 3	None	1 / 2
SBIS	P, b	Skip if Bit in I/O Register is Set	if (P(b)=1) $PC \leftarrow PC + 2$ or 3	None	1 / 2
BRBS	s, k	Branch if Status Flag Set	if (SREG(s) = 1) then $PC \leftarrow PC + k + 1$	None	1 / 2
BRBC	s, k	Branch if Status Flag Cleared	if (SREG(s) = 0) then $PC \leftarrow PC + k + 1$	None	1 / 2
BREQ	k	Branch if Equal	if (Z = 1) then $PC \leftarrow PC + k + 1$	None	1 / 2
BRNE	k	Branch if Not Equal	if (Z = 0) then $PC \leftarrow PC + k + 1$	None	1 / 2
BRCS	k	Branch if Carry Set	if (C = 1) then $PC \leftarrow PC + k + 1$	None	1 / 2
BRCC	k	Branch if Carry Cleared	if (C = 0) then $PC \leftarrow PC + k + 1$	None	1 / 2
BRSH	k	Branch if Same or Higher	if (C = 0) then $PC \leftarrow PC + k + 1$	None	1 / 2
BRLO	k	Branch if Lower	if (C = 1) then $PC \leftarrow PC + k + 1$	None	1 / 2
BRMI	k	Branch if Minus	if (N = 1) then $PC \leftarrow PC + k + 1$	None	1 / 2
BRPL	k	Branch if Plus	if (N = 0) then $PC \leftarrow PC + k + 1$	None	1 / 2
BRGE	k	Branch if Greater or Equal, Signed	if (N $\oplus$ V = 0) then $PC \leftarrow PC + k + 1$	None	1 / 2
BRLT	k	Branch if Less Than Zero, Signed	if (N $\oplus$ V = 1) then $PC \leftarrow PC + k + 1$	None	1 / 2
BRHS	k	Branch if Half Carry Flag Set	if (H = 1) then $PC \leftarrow PC + k + 1$	None	1 / 2
BRHC	k	Branch if Half Carry Flag Cleared	if (H = 0) then $PC \leftarrow PC + k + 1$	None	1 / 2
BRTS	k	Branch if T Flag Set	if (T = 1) then $PC \leftarrow PC + k + 1$	None	1 / 2
BRTC	k	Branch if T Flag Cleared	if (T = 0) then $PC \leftarrow PC + k + 1$	None	1 / 2
BRVS	k	Branch if Overflow Flag is Set	if (V = 1) then $PC \leftarrow PC + k + 1$	None	1 / 2
BRVC	k	Branch if Overflow Flag is Cleared	if (V = 0) then $PC \leftarrow PC + k + 1$	None	1 / 2
BRIE	k	Branch if Interrupt Enabled	if (I = 1) then $PC \leftarrow PC + k + 1$	None	1 / 2
BRID	k	Branch if Interrupt Disabled	if (I = 0) then $PC \leftarrow PC + k + 1$	None	1 / 2

Mnemonics	Operands	Description	Operation	Flags	#Clocks
<b>DATA TRANSFER INSTRUCTIONS</b>					
MOV	Rd, Rr	Move Between Registers	$Rd \leftarrow Rr$	None	1
LDI	Rd, K	Load Immediate	$Rd \leftarrow K$	None	1
LD	Rd, X	Load Indirect	$Rd \leftarrow (X)$	None	2
LD	Rd, X+	Load Indirect and Post-Inc.	$Rd \leftarrow (X), X \leftarrow X + 1$	None	2
LD	Rd, -X	Load Indirect and Pre-Dec.	$X \leftarrow X - 1, Rd \leftarrow (X)$	None	2
LD	Rd, Y	Load Indirect	$Rd \leftarrow (Y)$	None	2
LD	Rd, Y+	Load Indirect and Post-Inc.	$Rd \leftarrow (Y), Y \leftarrow Y + 1$	None	2
LD	Rd, -Y	Load Indirect and Pre-Dec.	$Y \leftarrow Y - 1, Rd \leftarrow (Y)$	None	2
LDD	Rd, Y+q	Load Indirect with Displacement	$Rd \leftarrow (Y + q)$	None	2
LD	Rd, Z	Load Indirect	$Rd \leftarrow (Z)$	None	2
LD	Rd, Z+	Load Indirect and Post-Inc.	$Rd \leftarrow (Z), Z \leftarrow Z + 1$	None	2
LD	Rd, -Z	Load Indirect and Pre-Dec.	$Z \leftarrow Z - 1, Rd \leftarrow (Z)$	None	2
LDD	Rd, Z+q	Load Indirect with Displacement	$Rd \leftarrow (Z + q)$	None	2
LDS	Rd, k	Load Direct from SRAM	$Rd \leftarrow (k)$	None	2
ST	X, Rr	Store Indirect	$(X) \leftarrow Rr$	None	2
ST	X+, Rr	Store Indirect and Post-Inc.	$(X) \leftarrow Rr, X \leftarrow X + 1$	None	2
ST	-X, Rr	Store Indirect and Pre-Dec.	$X \leftarrow X - 1, (X) \leftarrow Rr$	None	2
ST	Y, Rr	Store Indirect	$(Y) \leftarrow Rr$	None	2
ST	Y+, Rr	Store Indirect and Post-Inc.	$(Y) \leftarrow Rr, Y \leftarrow Y + 1$	None	2
ST	-Y, Rr	Store Indirect and Pre-Dec.	$Y \leftarrow Y - 1, (Y) \leftarrow Rr$	None	2
STD	Y+q, Rr	Store Indirect with Displacement	$(Y + q) \leftarrow Rr$	None	2
ST	Z, Rr	Store Indirect	$(Z) \leftarrow Rr$	None	2
ST	Z+, Rr	Store Indirect and Post-Inc.	$(Z) \leftarrow Rr, Z \leftarrow Z + 1$	None	2
ST	-Z, Rr	Store Indirect and Pre-Dec.	$Z \leftarrow Z - 1, (Z) \leftarrow Rr$	None	2
STD	Z+q, Rr	Store Indirect with Displacement	$(Z + q) \leftarrow Rr$	None	2
STS	k, Rr	Store Direct to SRAM	$(k) \leftarrow Rr$	None	2
LPM		Load Program Memory	$R0 \leftarrow (Z)$	None	3
IN	Rd, P	In Port	$Rd \leftarrow P$	None	1
OUT	P, Rr	Out Port	$P \leftarrow Rr$	None	1
PUSH	Rr	Push Register on Stack	$STACK \leftarrow Rr$	None	2
POP	Rd	Pop Register from Stack	$Rd \leftarrow STACK$	None	2
<b>BIT AND BIT-TEST INSTRUCTIONS</b>					
SBI	P, b	Set Bit in I/O Register	$I/O(P, b) \leftarrow 1$	None	2
CBI	P, b	Clear Bit in I/O Register	$I/O(P, b) \leftarrow 0$	None	2
LSL	Rd	Logical Shift Left	$Rd(n+1) \leftarrow Rd(n), Rd(0) \leftarrow 0$	Z, C, N, V	1
LSR	Rd	Logical Shift Right	$Rd(n) \leftarrow Rd(n+1), Rd(7) \leftarrow 0$	Z, C, N, V	1
ROL	Rd	Rotate Left Through Carry	$Rd(0) \leftarrow C, Rd(n+1) \leftarrow Rd(n), C \leftarrow Rd(7)$	Z, C, N, V	1
ROR	Rd	Rotate Right Through Carry	$Rd(7) \leftarrow C, Rd(n) \leftarrow Rd(n+1), C \leftarrow Rd(0)$	Z, C, N, V	1
ASR	Rd	Arithmetic Shift Right	$Rd(n) \leftarrow Rd(n+1), n=0..6$	Z, C, N, V	1
SWAP	Rd	Swap Nibbles	$Rd(3..0) \leftarrow Rd(7..4), Rd(7..4) \leftarrow Rd(3..0)$	None	1
BSET	s	Flag Set	$SREG(s) \leftarrow 1$	SREG(s)	1
BCLR	s	Flag Clear	$SREG(s) \leftarrow 0$	SREG(s)	1
BST	Rr, b	Bit Store from Register to T	$T \leftarrow Rr(b)$	T	1
BLD	Rd, b	Bit load from T to Register	$Rd(b) \leftarrow T$	None	1
SEC		Set Carry	$C \leftarrow 1$	C	1
CLC		Clear Carry	$C \leftarrow 0$	C	1
SEN		Set Negative Flag	$N \leftarrow 1$	N	1
CLN		Clear Negative Flag	$N \leftarrow 0$	N	1
SEZ		Set Zero Flag	$Z \leftarrow 1$	Z	1
CLZ		Clear Zero Flag	$Z \leftarrow 0$	Z	1
SEI		Global Interrupt Enable	$I \leftarrow 1$	I	1
CLI		Global Interrupt Disable	$I \leftarrow 0$	I	1
SES		Set Signed Test Flag	$S \leftarrow 1$	S	1
CLS		Clear Signed Test Flag	$S \leftarrow 0$	S	1
SEV		Set Twos Complement Overflow	$V \leftarrow 1$	V	1
CLV		Clear Twos Complement Overflow	$V \leftarrow 0$	V	1
SET		Set T in SREG	$T \leftarrow 1$	T	1
CLT		Clear T in SREG	$T \leftarrow 0$	T	1
SEH		Set Half Carry Flag in SREG	$H \leftarrow 1$	H	1
CLH		Clear Half Carry Flag in SREG	$H \leftarrow 0$	H	1
NOP		No Operation		None	1
SLEEP		Sleep	(see specific descr. for Sleep function)	None	3
WDR		Watchdog Reset	(see specific descr. for WDR/timer)	None	1

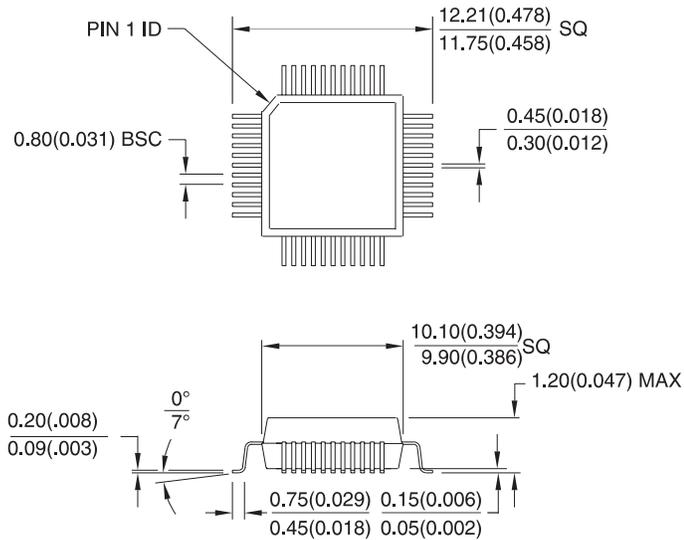
## Ordering Information

Power Supply	Speed (MHz)	Ordering Code	Package	Operation Range
2.7 - 6.0V	4	AT90LS4434-4AC	44A	Commercial (0°C to 70°C)
		AT90LS4434-4JC	44J	
		AT90LS4434-4PC	40P6	
		AT90LS4434-4AI	44A	Industrial (-40°C to 85°C)
		AT90LS4434-4JI	44J	
		AT90LS4434-4PI	40P6	
4.0 - 6.0V	8	AT90S4434-8AC	44A	Commercial (0°C to 70°C)
		AT90S4434-8JC	44J	
		AT90S4434-8JC	40P6	
		AT90S4434-8AI	44A	Industrial (-40°C to 85°C)
		AT90S4434-8JI	44J	
		AT90S4434-8PI	40P6	
2.7 - 6.0V	4	AT90LS8535-4AC	44A	Commercial (0°C to 70°C)
		AT90LS8535-4JC	44J	
		AT90LS8535-4PC	40P6	
		AT90LS8535-4AI	44A	Industrial (-40°C to 85°C)
		AT90LS8535-4JI	44J	
		AT90LS8535-4PI	40P6	
4.0 - 6.0V	8	AT90S8535-8AC	44A	Commercial (0°C to 70°C)
		AT90S8535-8JC	44J	
		AT90S8535-8JC	40P6	
		AT90S8535-8AI	44A	Industrial (-40°C to 85°C)
		AT90S8535-8JI	44J	
		AT90S8535-8PI	40P6	

Package Type	
<b>44A</b>	44 Lead, Thin (1.0mm) Plastic Gull Wing Quad Flat Package (TQFP)
<b>44J</b>	44 Lead, Plastic J-Leaded Chip Carrier (PLCC)
<b>40P6</b>	40 Lead, 0.600" Wide, Plastic Dual in Line Package (PDIP)

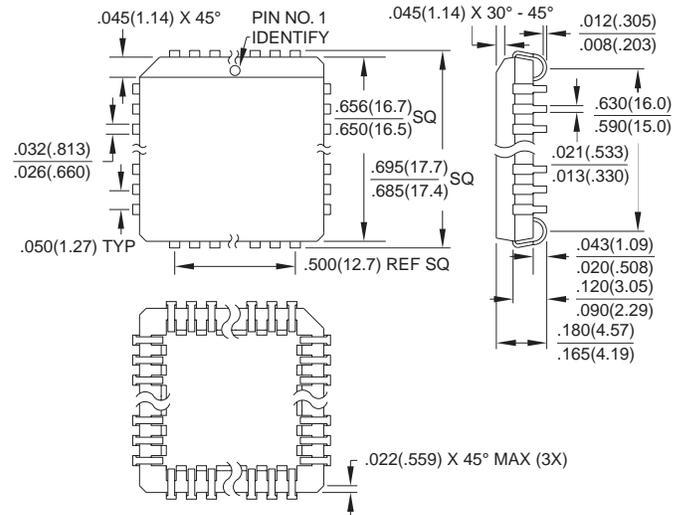
## Packaging Information

**44A, 44-Lead, Thin (1.0 mm) Plastic Gull Wing Quad Flat Package (TQFP)**  
Dimensions in Millimeters and (Inches)



\*Controlling dimension: millimeters

**44J, 44-Lead, Plastic J-Leaded Chip Carrier (PLCC)**  
Dimensions in Inches and (Millimeters)



**40P6, 40-Lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)**  
Dimensions in Inches and (Millimeters)  
JEDEC STANDARD MS-011 AC

