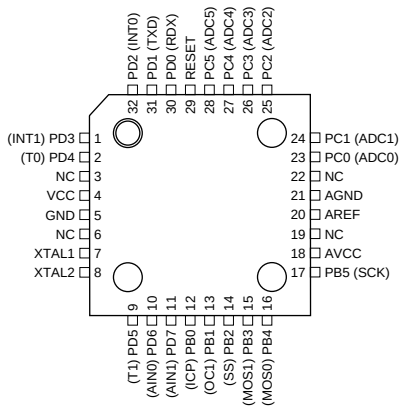
Features

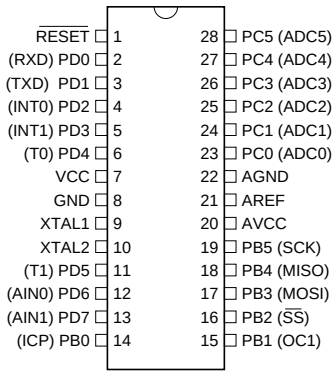
- High-performance and Low-power AVR® 8-bit RISC Architecture
  - 118 Powerful Instructions - Most Single Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Up to 8 MIPS Throughput at 8 MHz
- Data and Nonvolatile Program Memory
  - 2K/4K Bytes of In-System Programmable Flash
    Endurance 1,000 Write/Erase Cycles
  - 128 Bytes of SRAM
  - 128/256 Bytes of In-System Programmable EEPROM
    Endurance: 100,000 Write/Erase Cycles
  - Programming Lock for Flash Program and EEPROM Data Security
- Peripheral Features
  - One 8-bit Timer/Counter with Separate Prescaler
  - Expanded 16-bit Timer/Counter with Separate Prescaler, Compare, Capture Modes and 8-, 9- or 10-bit PWM
  - On-chip Analog Comparator
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - Programmable UART
  - 6-channel, 10-bit ADC
  - Master/Slave SPI Serial Interface
- Special Microcontroller Features
  - Brown-Out Reset Circuit
  - Enhanced Power-on Reset Circuit
  - Low-Power Idle and Power Down Modes
  - External and Internal Interrupt Sources
- Specifications
  - Low-power, High-speed CMOS Process Technology
  - Fully Static Operation
- Power Consumption at 4 MHz, 3V, 25°C
  - Active: 3.4 mA
  - Idle Mode: 1.4 mA
  - Power Down Mode: <1 µA
- I/O and Packages
  - 20 Programmable I/O Lines
  - 28-pin PDIP and 32-pin TQFP
- Operating Voltage
  - 2.7V - 6.0V (AT90LS2333 and AT90LS4433)
  - 4.0V - 6.0V (AT90S2333 and AT90S4433)
- Speed Grades
  - 0 - 4 MHz (AT90LS2333 and AT90LS4433)
  - 0 - 8 MHz (AT90S2333 and AT90S4433)

Pin Configurations

TQFP Top View



PDIP



8-bit AVR® Microcontroller with 2K/4K bytes In-System Programmable Flash

AT90S2333
AT90LS2333
AT90S4433
AT90LS4433

Preliminary

Rev. 1042D–04/99

Description

The AT90S2333/4433 is a low-power CMOS 8-bit microcontroller based on the AVR RISC architecture. By executing powerful instructions in a single clock cycle, the AT90S2333/4433 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

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 microcontrollers.

The AT90S2333/4433 provides the following features: 2K/4K bytes of In-System Programmable Flash, 128/256 bytes EEPROM, 128 bytes SRAM, 20 general purpose I/O lines, 32 general purpose working registers, two flexible timer/counters with compare modes, internal and external interrupts, a programmable serial UART, 6-channel, 10-bit ADC, programmable Watchdog Timer with internal oscillator, an SPI serial port and two 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.

The device is manufactured using Atmel's high-density nonvolatile memory technology. The on-chip Flash program memory can be reprogrammed in-system through an SPI serial interface or by a conventional nonvolatile memory programmer. By combining a RISC 8-bit CPU with In-System Programmable Flash on a monolithic chip, the Atmel AT90S2333/4433 is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

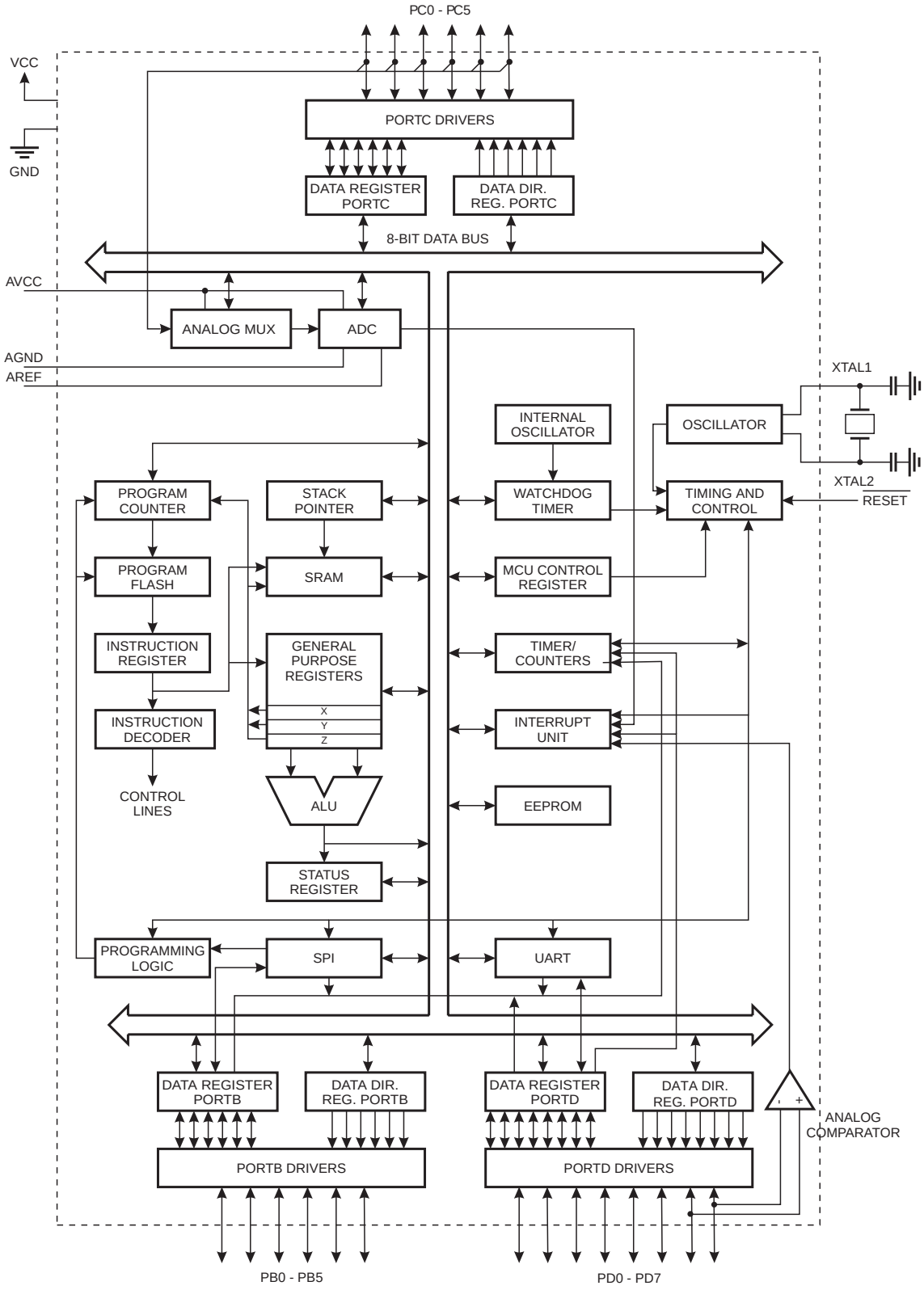
The AT90S2333/4433 AVR is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, in-circuit emulators, and evaluation kits.

Table 1. Comparison Table

Device	Flash	EEPROM	SRAM	Voltage Range	Frequency
AT90S2333	2K	128B	128B	4.0V - 6.0V	0 - 8 MHz
AT90LS2333	2K	128B	128B	2.7V - 6.0V	0 - 4 MHz
AT90S4433	4K	256B	128B	4.0V - 6.0V	0 - 8 MHz
AT90LS4433	4K	256B	128B	2.7V - 6.0V	0 - 4 MHz

Block Diagram

Figure 1. The AT90S2333/4433 Block Diagram



Pin Descriptions

VCC

Supply voltage

GND

Ground

Port B (PB5..PB0)

Port B is a 6-bit bi-directional I/O port with internal pullup 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 AT90S2333/4433 as listed on page 60.

The port B pins are tristated when a reset condition becomes active, even if the clock is not running.

Port C (PC5..PC0)

Port C is a 6-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. Port C also serves as the analog inputs to the A/D Converter.

The port C pins are tristated when a reset condition becomes active, even if the clock is not running.

Port D (PD7..PD0)

Port D is an 8-bit bi-directional 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 AT90S2333/4433 as listed on page 67.

The port D pins are tristated when a reset condition becomes active, even if the clock is not running.

$\overline{\text{RESET}}$

Reset input. An external reset is generated by a low level on the $\overline{\text{RESET}}$ pin. Reset pulses longer than 50 ns will generate a reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a reset.

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 V_{CC} via a low-pass filter. See page 52 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 2.7V to AVCC must be applied to this pin.

AGND

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

Clock Options

Crystal Oscillator

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 2 and Figure 3. Either a quartz crystal or a ceramic resonator may be used.

External Clock

If the oscillator is to be used as a clock for an external device, the clock signal from XTAL2 may be routed to one HC buffer, while reducing the load capacitor by 5 pF, as shown in Figure 3. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 4.

Figure 2. Oscillator Connections

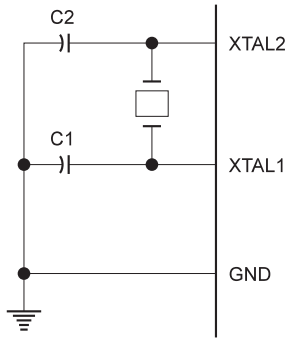


Figure 3. Using MCU Oscillator as a Clock for an External Device

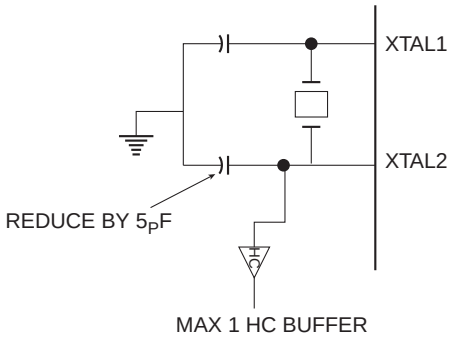
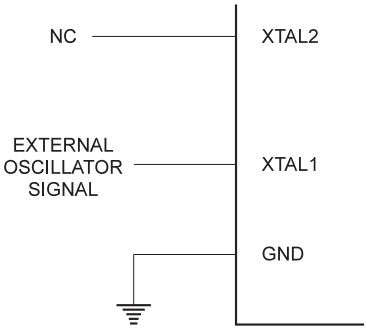


Figure 4. External Clock Drive Configuration



Architectural Overview

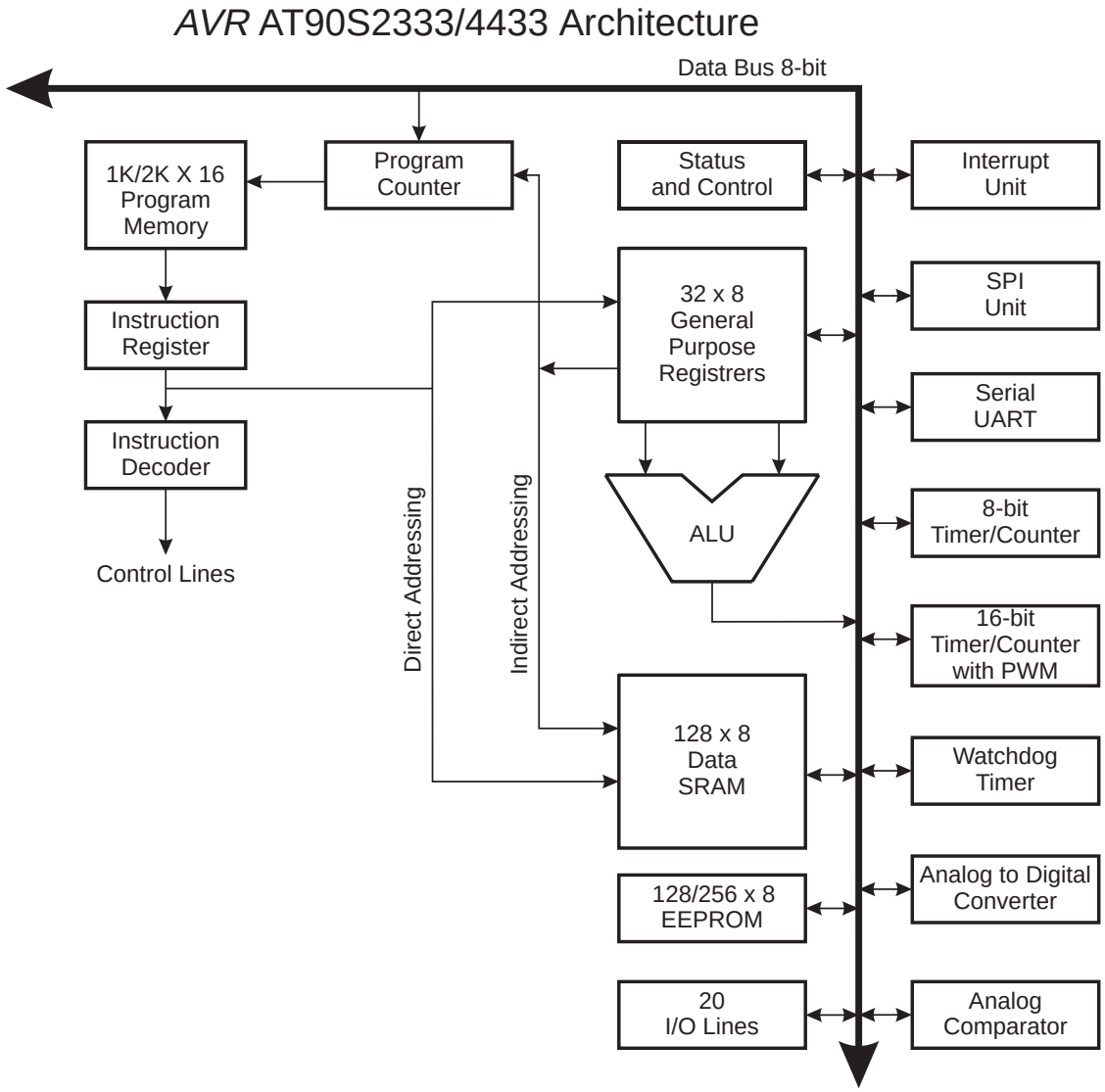
The fast-access register file concept contains 32 x 8-bit general purpose working registers with a single clock cycle access time. This means that during one single clock cycle, one Arithmetic Logic Unit (ALU) operation is executed. Two operands are output from the register file, the operation is executed,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

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 5 shows the AT90S2333/4433 AVR RISC microcontroller architecture.

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

Figure 5. The AT90S2333/4433 AVR RISC Architecture



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 two stage pipeline. While one instruction is being executed, the next instruction is pre-fetched from the program memory. This concept enables instructions to be executed in every clock cycle.

The program memory is In-System Programmable Flash memory.

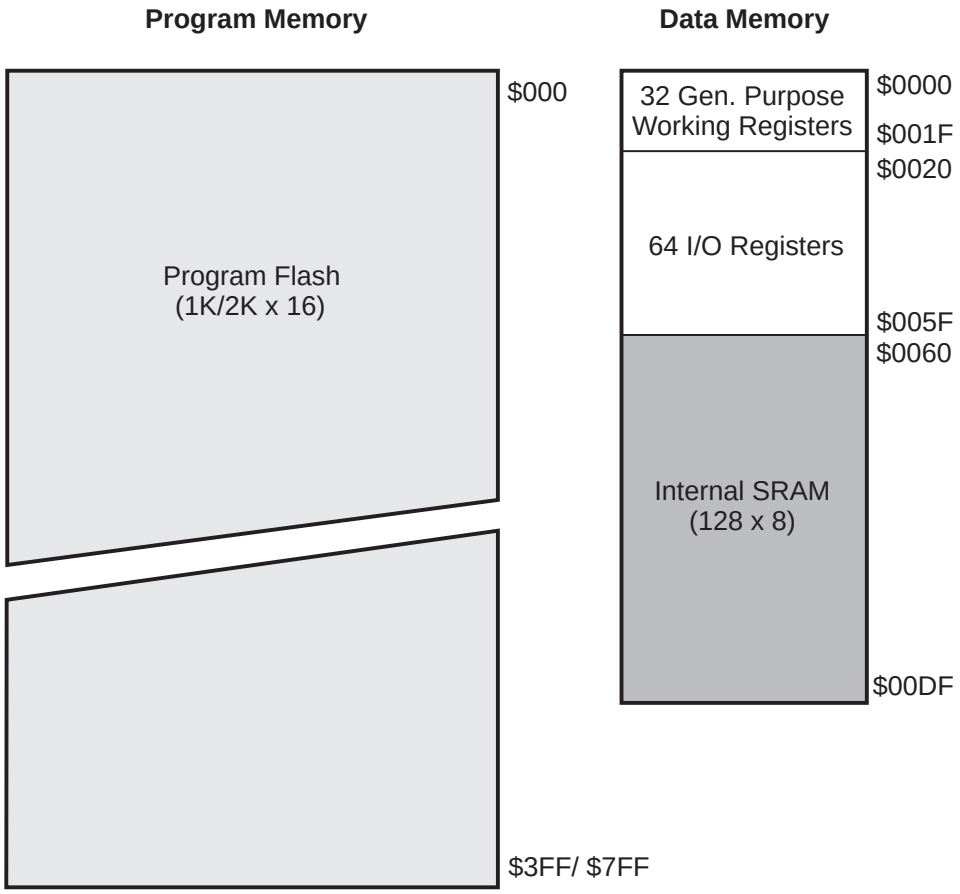
With the relative jump and call instructions, the whole 1K/2K word address space is directly accessed. Most AVR instructions have a single 16-bit word format. Every program memory address contains a 16- or 32-bit instruction.

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 8-bit stack pointer SP is read/write accessible in the I/O space.

The 128 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 6. AT90S2333/4433 Memory Maps

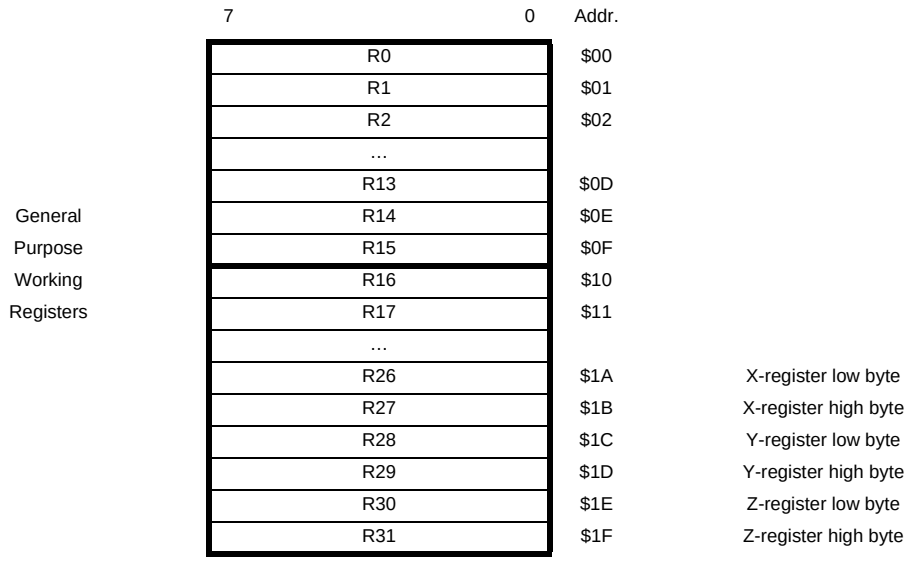


A flexible interrupt module has its control registers in the I/O space with an additional global interrupt enable bit in the status register. All the different interrupts have a separate interrupt vector in the interrupt vector table at the beginning of the program memory. The different interrupts have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

General Purpose Register File

Figure 7 shows the structure of the 32 general purpose working registers in the CPU.

Figure 7. AVR CPU General Purpose Working Registers



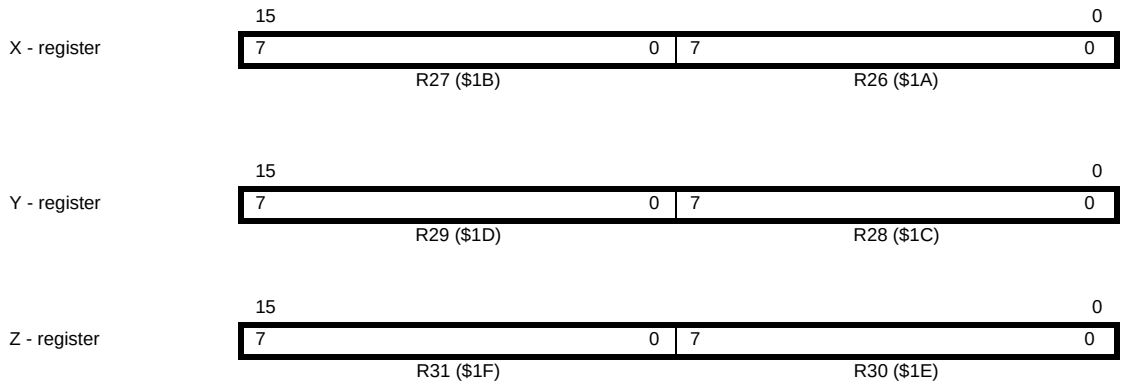
All the register operating instructions in the instruction set have direct and single cycle access to all registers. The only exceptions are the five constant arithmetic and logic instructions SBCI, SUBI, CPI, ANDI, and ORI between a constant and a register, and the LDI instruction for load immediate 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 7, each register is also assigned a data memory address, mapping them directly into the first 32 locations of the user Data Space. Although not being physically implemented as SRAM locations, this memory organization provides great flexibility in access of the registers, as the X, Y and Z registers can be set to index any register in the file.

X-register, Y-register, and Z-register

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

Figure 8. 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).

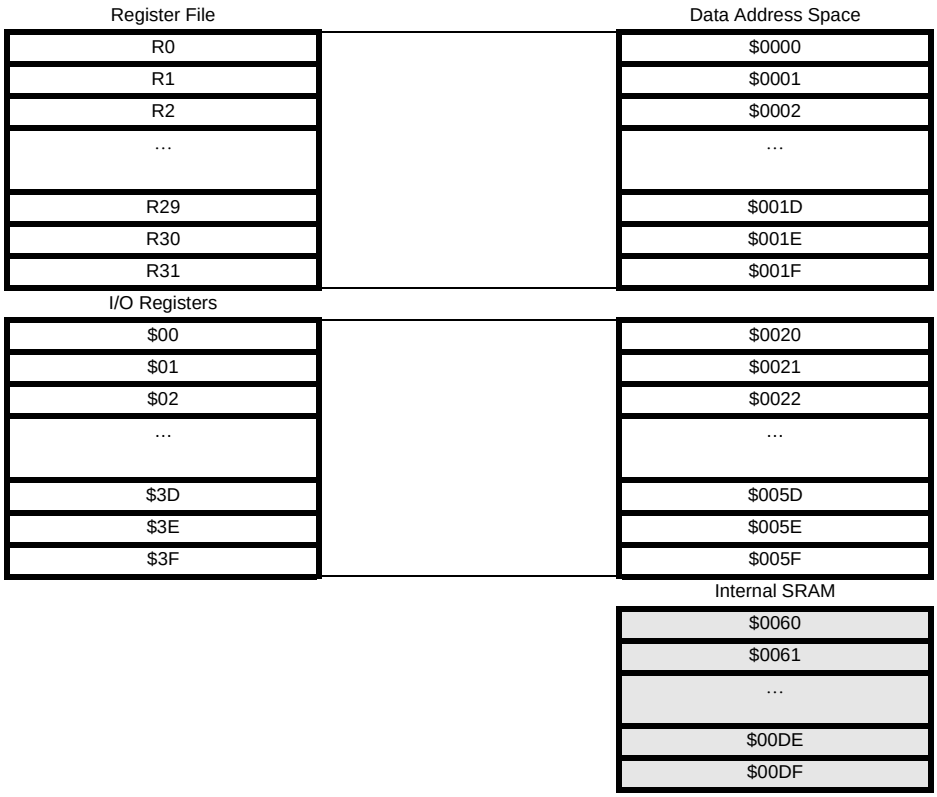
ALU - Arithmetic Logic Unit

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.

In-System Programmable Flash Program Memory

The AT90S2333/4433 contains 2K/4K 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 1K/2K x 16. The Flash memory has an endurance of at least 1000 write/erase cycles. The AT90S2333/4433 Program Counter (PC) is 10/11 bits wide, thus addressing the 1024/2048 program memory addresses. See page 78 for a detailed description on Flash data downloading. See page 10 for the different program memory addressing modes.

Figure 9. SRAM Organization



SRAM Data Memory

The figure above shows how the AT90S2333/4433 SRAM Memory is organized.

The lower 224 Data Memory locations address the Register file, the I/O Memory and the internal data SRAM. The first 96 locations address the Register File and I/O Memory, and the next 128 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 128 bytes of internal data SRAM in the AT90S2333/4433 are all accessible through all these addressing modes.

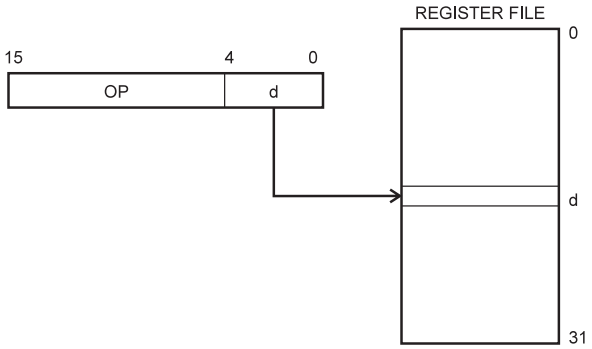
See the next section for a detailed description of the different addressing modes.

Program and Data Addressing Modes

The AT90S2333/4433 AVR RISC microcontroller supports powerful and efficient addressing modes for access to the Flash program memory, SRAM, Register File, and I/O data 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

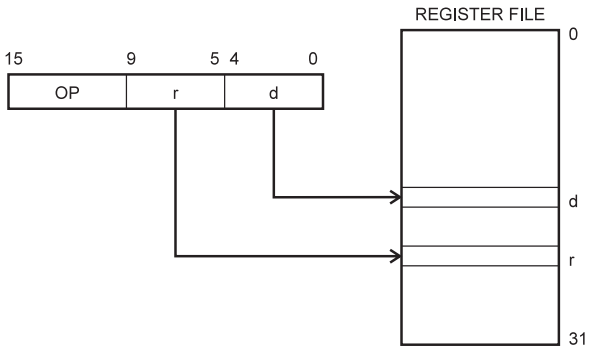
Figure 10. Direct Single Register Addressing



The operand is contained in register d (Rd).

Register Direct, Two Registers Rd and Rr

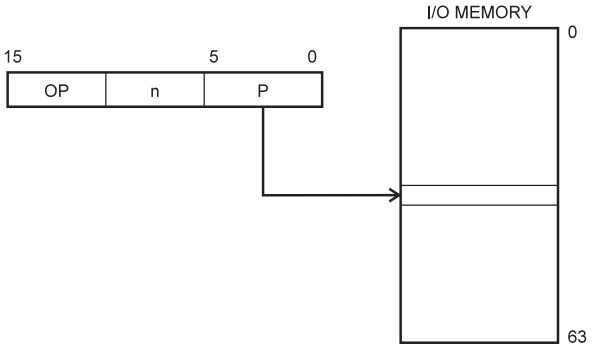
Figure 11. Direct Register Addressing, Two Registers



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

I/O Direct

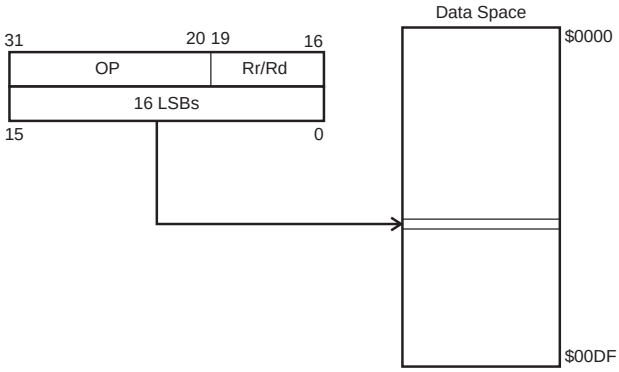
Figure 12. I/O Direct Addressing



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

Data Direct

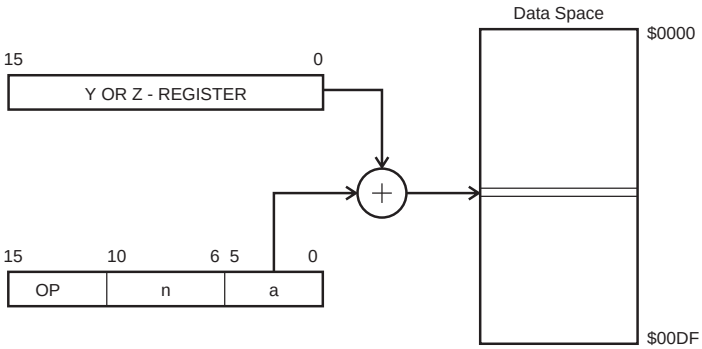
Figure 13. Direct Data Addressing



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

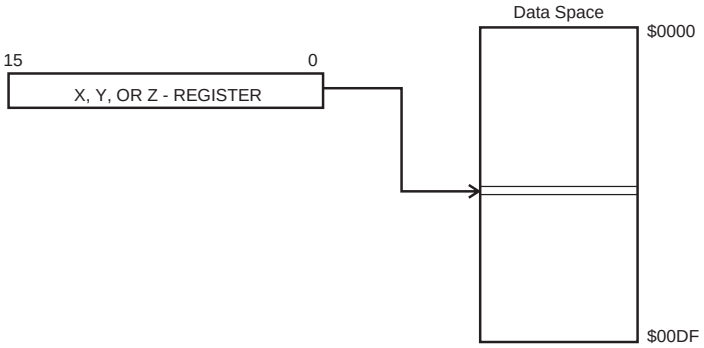
Figure 14. Data Indirect with Displacement



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

Data Indirect

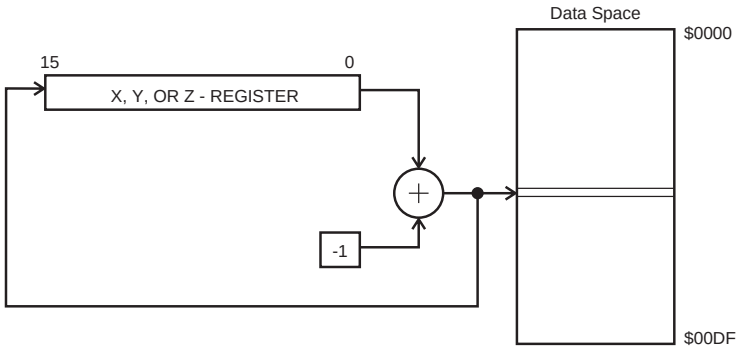
Figure 15. Data Indirect Addressing



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

Data Indirect with Pre-Decrement

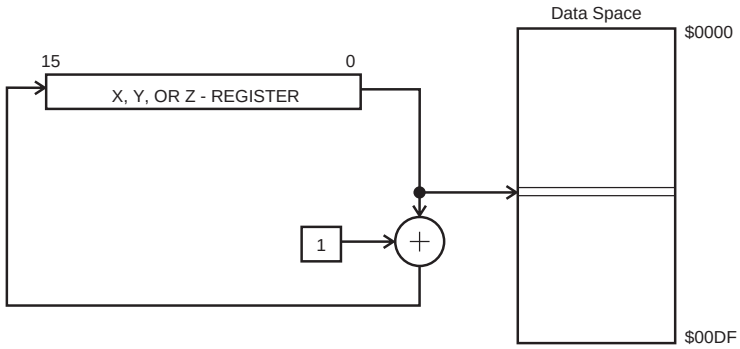
Figure 16. Data Indirect Addressing with Pre-Decrement



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

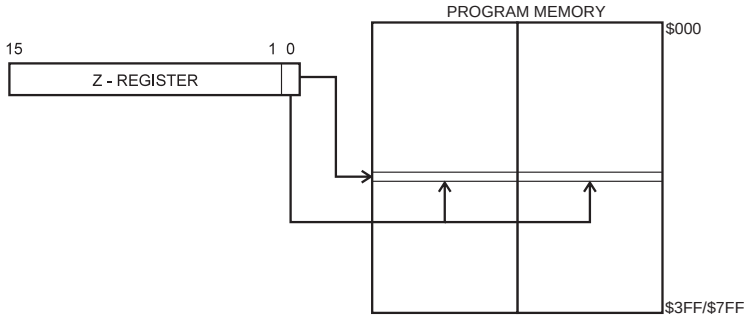
Figure 17. Data Indirect Addressing with Post-Increment



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

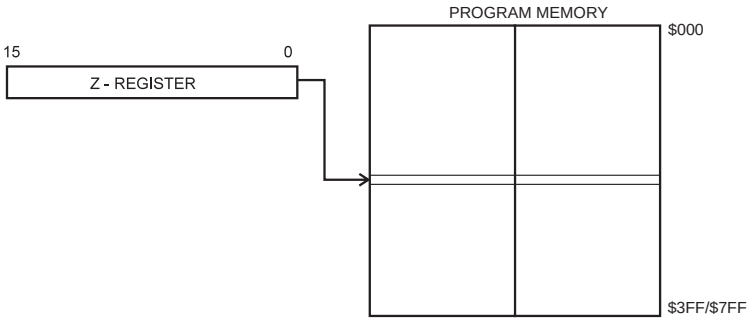
Figure 18. Code Memory Constant Addressing



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

Indirect Program Addressing, IJMP and ICALL

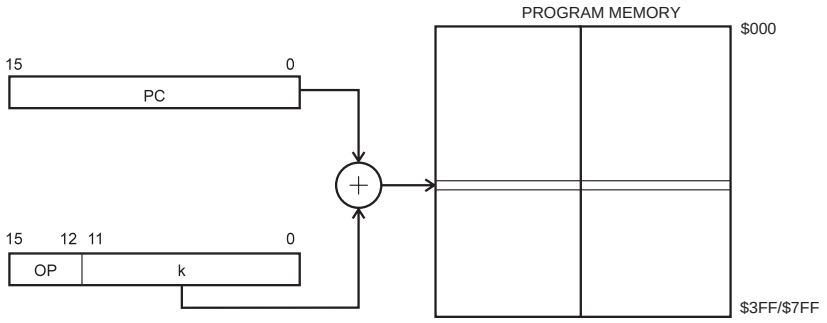
Figure 19. Indirect Program Memory Addressing



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 20. Relative Program Memory Addressing



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

EEPROM Data Memory

The AT90S2333/4433 contains 128/256 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 per location. The access between the EEPROM and the CPU is described on page 44 specifying the EEPROM address registers, the EEPROM data register, and the EEPROM control register.

For the SPI data downloading, see page 78 for a detailed description. The EEPROM data memory is In-System Programmable through the SPI port. Please refer to the “EEPROM Read/Write Access” section on page 38 for a thorough description on EEPROM access.

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 Ø, directly generated from the external clock crystal for the chip. No internal clock division is used.

Figure 21 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 21. The Parallel Instruction Fetches and Instruction Executions

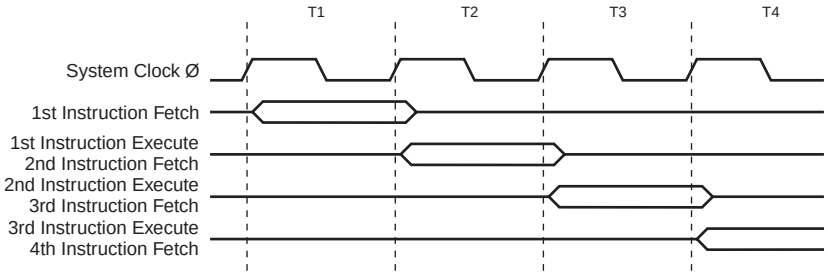
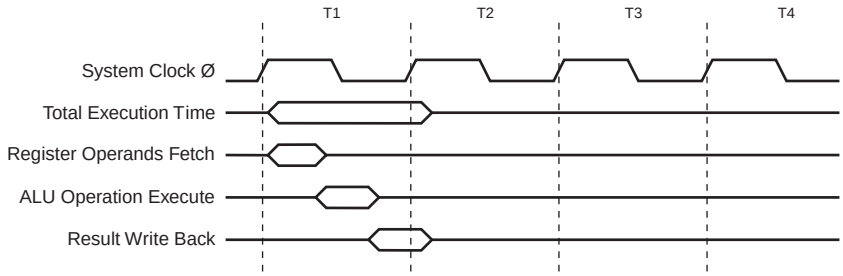


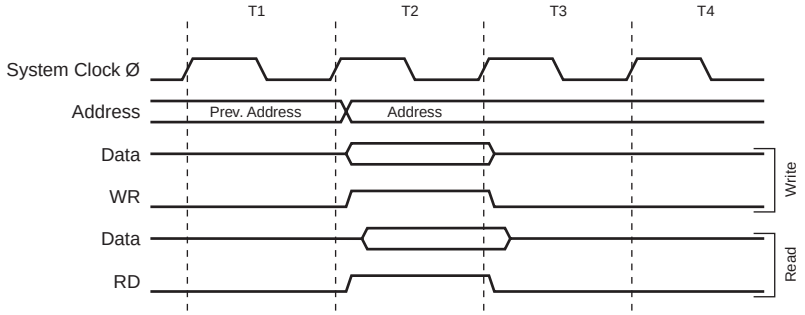
Figure 22 shows the internal timing concept for the register file. In a single clock cycle an ALU operation using two register operands is executed, and the result is stored back to the destination register.

Figure 22. Single Cycle ALU Operation



The internal data SRAM access is performed in two System Clock cycles as described in Figure 23.

Figure 23. On-Chip Data SRAM Access Cycles



I/O Memory

The I/O space definition of the AT90S2333/4433 is shown in the following table:

Table 2. AT90S2333/4433 I/O Space

I/O Address (SRAM Address)	Name	Function
$3F ($5F)	SREG	Status REGister
$3D ($5D)	SP	Stack Pointer
$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 ($54)	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)	OCR1H	Timer/Counter1 Output Compare Register High Byte
$2A ($4A)	OCR1L	Timer/Counter1 Output Compare Register Low Byte
$27 ($47)	ICR1H	Timer/Counter1 Input Capture Register High Byte
$26 ($46)	ICR1L	Timer/Counter 1 Input Capture Register Low Byte
$21 ($41)	WDTCR	Watchdog Timer Control Register
$1E ($3E)	EEAR	EEPROM Address Register
$1D ($3D)	EEDR	EEPROM Data Register
$1C ($3C)	EECR	EEPROM Control Register
$18 ($38)	PORTB	Data Register, Port B
$17 ($37)	DDRB	Data Direction Register, Port B
$16 ($36)	PINB	Input Pins, Port B

Table 2. AT90S2333/4433 I/O Space (Continued)

I/O Address (SRAM Address)	Name	Function
$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
$03 ($23)	UBRRHI	UART Baud Rate Register High

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

All AT90S2333/4433 I/Os and peripherals are placed in the I/O space. The 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. When using the I/O specific commands IN, OUT 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.

For compatibility with future devices, reserved bits should be written to zero when accessed. Reserved I/O memory addresses should never be written.

Some of the status flags are cleared by writing a logical one to them. Note that the CBI and SBI instructions will operate on all bits in the I/O register, writing a one back into any flag read as set, thus clearing the flag. The CBI and SBI instructions work with registers $00 to $1F only.

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

Status Register - SREG

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

Bit	7	6	5	4	3	2	1	0	
$3F ($5F)	I	T	H	S	V	N	Z	C	SREG
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 - 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 separate control registers. If the global interrupt enable register is cleared (zero), none of the interrupts are enabled independent of the individual interrupt enable settings. 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 arithmetical 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 from an arithmetical or logical operation. See the Instruction Set Description for detailed information.

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

The zero flag Z indicates a zero result from an arithmetical or logical operation. See the Instruction Set Description for detailed information.

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

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

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

Stack Pointer - SP

The AT90S2333/4433 Stack Pointer is implemented as an 8-bit register in the I/O space location $3D ($5D). As the AT90S2333/4433 data memory has $0DF locations, 8 bits are used.

$3D ($5D)	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	SP
	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	
Initial value	0	0	0	0	0	0	0	0	

The Stack Pointer points to the data SRAM stack area where the Subroutine and Interrupt Stacks are located. This Stack space in the data SRAM must be defined by the program before any subroutine calls are executed or interrupts are enabled. The stack pointer must be set to point above $60. The Stack Pointer is decremented by one when data is pushed onto the Stack with the PUSH instruction, and it is decremented by two when an address is pushed onto the Stack with subroutine calls and interrupts. The Stack Pointer is incremented by one when data is popped from the Stack with the POP

instruction, and it is incremented by two when an address is popped from the Stack with return from subroutine RET or return from interrupt RETI.

Reset and Interrupt Handling

The AT90S2333/4433 provides 13 different interrupt sources. These interrupts and the separate reset vector, each have a separate program vector in the program memory space. All interrupts are assigned individual enable bits which must be set (one) together with the I-bit in the status register in order to enable the interrupt.

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

Table 3. Reset and Interrupt Vectors

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

The most typical 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    TIM1_CAPT ; Timer1 Capture Handler
$004                           rjmp    TIM1_COMP ; Timer1 compare Handler
$005                           rjmp    TIM1_OVF  ; Timer1 Overflow Handler
$006                           rjmp    TIM0_OVF  ; Timer0 Overflow Handler
$007                           rjmp    SPI_STC;  ; SPI Transfer Complete Handler
$008                           rjmp    UART_RXC  ; UART RX Complete Handler
$009                           rjmp    UART_DRE  ; UDR Empty Handler
$00a                           rjmp    UART_TXC  ; UART TX Complete Handler
$00b                           rjmp    ADC       ; ADC Conversion Complete Interrupt Handler
$00c                           rjmp    EE_RDY    ; EEPROM Ready Handler
$00d                           rjmp    ANA_COMP  ; Analog Comparator Handler
;
```

```
$00e                MAIN:       ldi     r16,low(RAMEND); Main program start
$00f                            out     SP,r16;
$010                            <instr>  xxx       ;
  …                 …             …       …
```

Reset Sources

The AT90S2333/4433 has four sources of reset:

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

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 24 shows the reset logic. Table 4 and Table 5 define the timing and electrical parameters of the reset circuitry.

Figure 24. Reset Logic

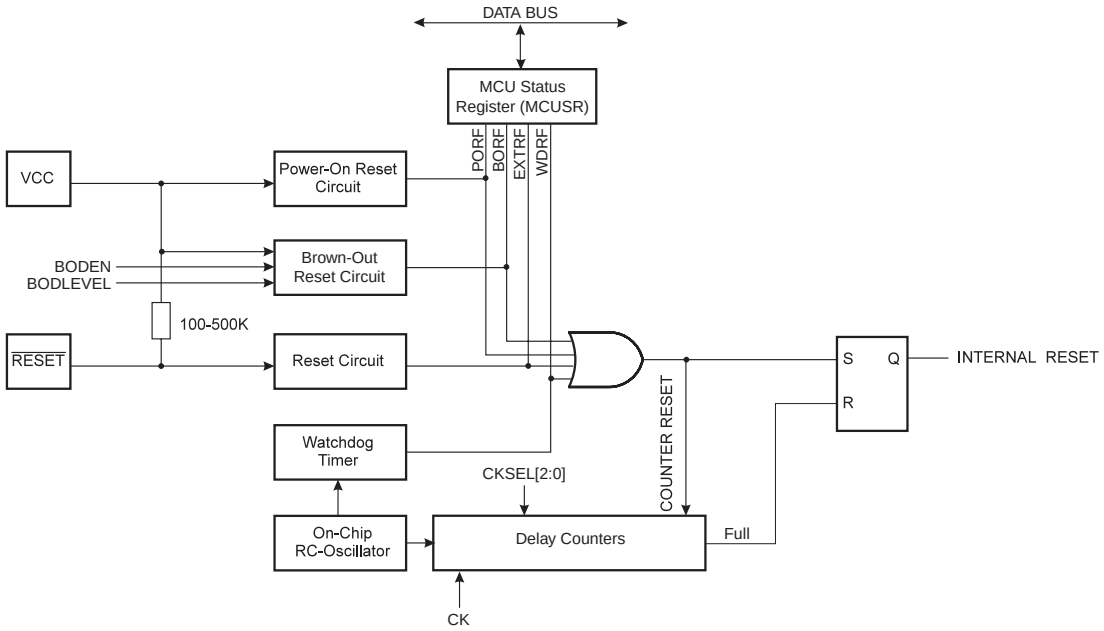


Table 4. Reset Characteristics (V_{CC} = 5.0V)

Symbol	Parameter	Min	Typ	Max	Units
V_{POT}	Power-On Reset Threshold Voltage, rising	1.0	1.4	1.8	V
	Power-On Reset Threshold Voltage, falling	0.4	0.6	0.8	V
V_{RST}	$\overline{\text{RESET}}$ Pin Threshold Voltage		$0.6V_{CC}$		V
V_{BOT}	Brown-Out Reset Threshold Voltage	2.6 (BODLEVEL = 1)	2.7 (BODLEVEL = 1)	2.8 (BODLEVEL = 1)	V
		3.8 (BODLEVEL = 0)	4.0 (BODLEVEL = 0)	4.2 (BODLEVEL = 0)	

Note: The Power-On Reset will not work unless the supply voltage has been below Vpot (falling).

Table 5. Reset Delay Selections

CKSEL [2:0]	Start-Up Time, t_{TOUT} at V_{CC} = 2.7V	Start-Up Time, t_{TOUT} at V_{CC} = 5.0V	Recommended Usage
000	16 ms + 6 CK	4 ms + 6 CK	External Clock, slowly rising power
001	6 CK	6 CK	External Clock, BOD enabled(1)
010	256 ms + 16K CK	64 ms + 16K CK	Crystal Oscillator
011	16 ms + 16K CK	4 ms + 16K CK	Crystal Oscillator, fast rising power
100	16K CK	16K CK	Crystal Oscillator, BOD enabled(1)
101	256 ms + 1K CK	64 ms + 1K CK	Ceramic Resonator
110	16 ms + 1K CK	4 ms + 1K CK	Ceramic Resonator, fast rising power
111	1K CK	1K CK	Ceramic Resonator, BOD enabled (1)

Notes: 1. Or external power-on reset.

This table shows the start-up times from reset. From sleep, only the clock counting part of the start-up time is used. The watchdog oscillator is used for timing the real-time part of the start-up time. The number WDT oscillator cycles used for each time-out is shown in Table 6.

Table 6. Number of Watchdog Oscillator Cycles

Time-out	Number of cycles
4.0 ms (at V_{cc}=5.0V)	4K
64 ms (at V_{cc}=5.0V)	64K

The frequency of the watchdog oscillator is voltage dependent as shown in the Electrical Characteristics section.

Power-On Reset

A Power-On Reset (POR) pulse is generated by an on-chip detection circuit. The detection level is nominally 2.2V. The POR is activated whenever V_{CC} is below the detection level. The POR circuit can be used to trigger the start-up reset, as well as detect a failure in supply voltage.

The Power-On Reset (POR) circuit ensures that the device is reset from power-on. Reaching the power-on reset threshold voltage invokes a delay counter, which determines the delay, for which the device is kept in RESET after V_{CC} rise. The time-out period of the delay counter is a combination of internal RC oscillator cycles and external oscillator cycles, and it can be defined by the user through the CKSEL fuses. The eight different selections for the delay period are presented in Table 5. The RESET signal is activated again, without any delay, when the V_{CC} decreases below detection level.

Figure 25. MCU Start-Up, $\overline{\text{RESET}}$ Tied to VCC.

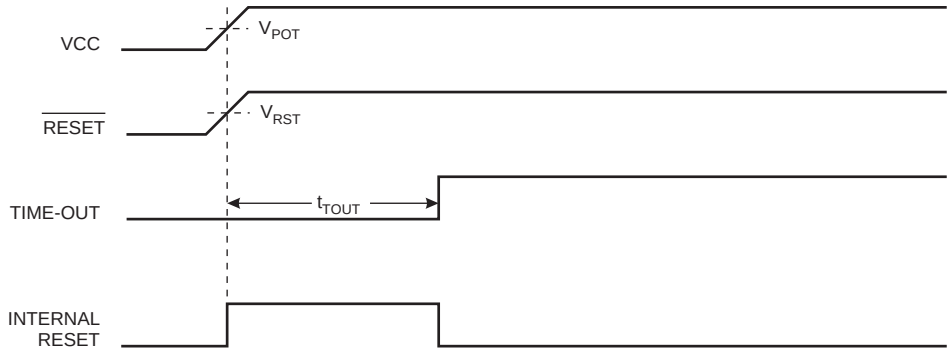
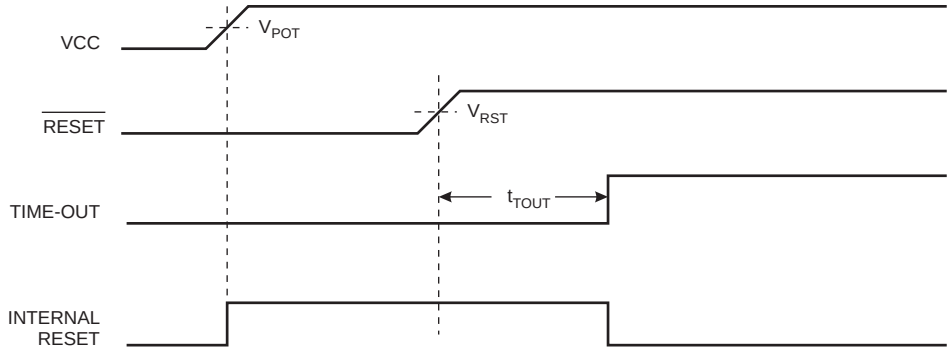


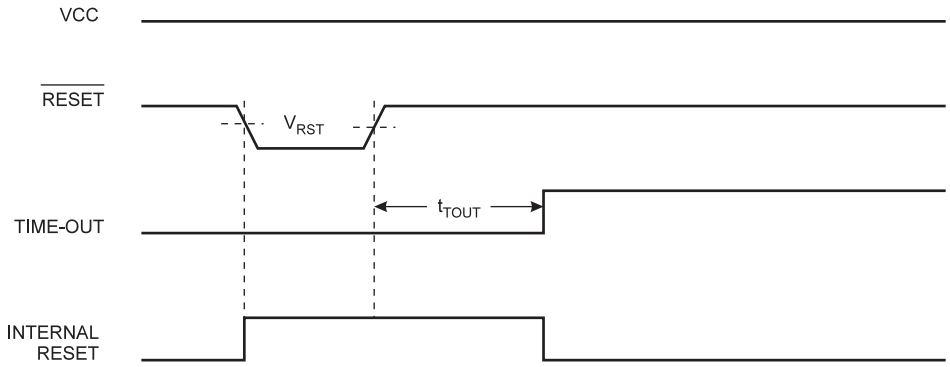
Figure 26. MCU Start-Up, $\overline{\text{RESET}}$ Controlled Externally



External Reset

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

Figure 27. External Reset During Operation

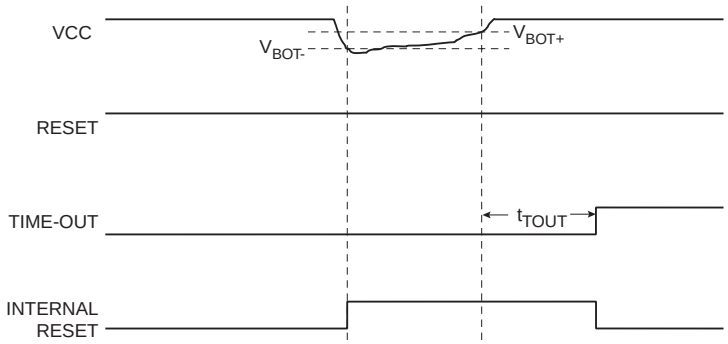


Brown-Out Detection

AT90S2333/4433 has an on-chip brown-out detection (BOD) circuit for monitoring the V_{CC} level during the operation. The power supply must be decoupled with a 47 nF to 100 nF capacitor if the BOD function is used. The BOD circuit can be enabled/disabled by the fuse BODEN. When BODEN is enabled (BODEN programmed), and V_{CC} decreases to a value below the trigger level, the brown-out reset is immediately activated. When V_{CC} increases above the trigger level, the brown-out reset is deactivated after a delay. The delay is defined by the user in the same way as the delay of POR signal, in Table 5. The trigger level for the BOD can be selected by the fuse BODLEVEL to be 2.7V (BODLEVEL unprogrammed), or 4.0V (BODLEVEL programmed). The trigger level has a hysteresis of 50 mV to ensure spike free brown-out detection.

The BOD circuit will only detect a drop in V_{CC} if the voltage stays below the trigger level for longer than 3 µs for trigger level 4.0V, 7 µs for trigger level 2.7V (typical values).

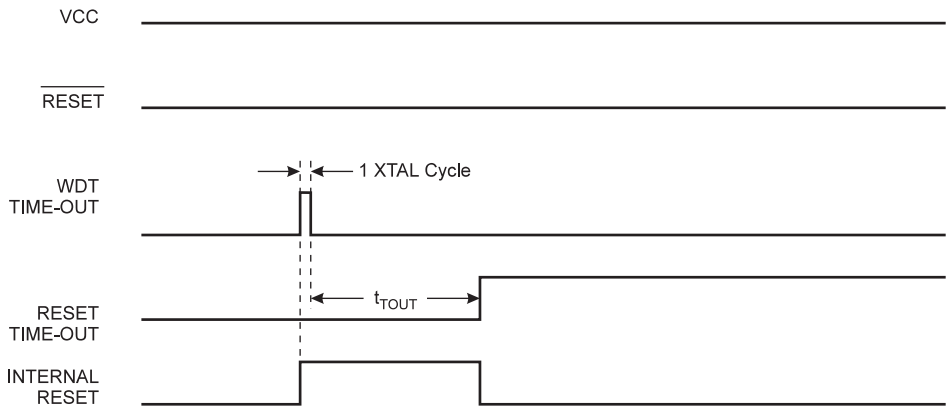
Figure 28. Brown-Out Reset During Operation



Watchdog Reset

When the Watchdog times out, it will generate a short reset pulse of 1 XTAL cycle duration. On the falling edge of this pulse, the delay timer starts counting the Time-out period t_{TOUT}. Refer to Page page 36 for details on operation of the Watchdog.

Figure 29. 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)	-	-	-	-	WDRF	BORF	EXTRF	PORF	MCUSR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	See bit description				

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

These bits are reserved bits in the AT90S2333 and always read as zero.

- **Bit 3 - WDRF: Watchdog Reset Flag**

This bit is set if a watchdog reset occurs. The bit is cleared by a power-on reset, or by writing a logic zero to the flag.

- **Bit 2 - BORF: Brown-Out Reset Flag**

This bit is set if a brown-out reset occurs. The bit is cleared by a power-on reset, or by writing a logic zero to the flag.

- **Bit 1 - EXTRF: External Reset Flag**

This bit is set if an external reset occurs. The bit is cleared by a power-on reset, or by writing a logic zero to the flag.

- **Bit 0 - PORF: Power-on Reset Flag**

This bit is set if a power-on reset occurs. The bit is cleared only by writing a logic zero to the flag.

To make use of the reset flags to identify a reset condition, the user should read and then clear the MCUSR as early as possible in the program. If the register is cleared before another reset occurs, the source of the reset can be found by examining the reset flags.

Interrupt Handling

The AT90S2333/4433 has two 8-bit Interrupt Mask control registers; GIMSK - General Interrupt Mask register and TIMSK - Timer/Counter Interrupt Mask register.

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

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.

If an interrupt condition occurs when the corresponding interrupt enable bit is cleared (zero), the interrupt flag will be set and remembered until the interrupt is enabled, or the flag is cleared by software.

If one or more interrupt conditions occur when the global interrupt enable bit is cleared (zero), the corresponding interrupt flag(s) will be set and remembered until the global interrupt enable bit is set (one), and will be executed by order of priority.

Note that external level interrupt does not have a flag, and will only be remembered for as long as the interrupt condition is active.

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

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 enabled. The Interrupt Sense Control1 bits 1/0 (ISC11 and ISC10) in the MCU general Control Register (MCUCR) defines whether the external interrupt is activated on rising or falling edge of the INT1 pin or level sensed. Please note that INTF1 flag is not set when level sensitive interrupt condition is met. However, INT1 interrupt is generated, provided that INT1 mask bit is set in GIMSK register. 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 enabled. The Interrupt Sense Control0 bits 1/0 (ISC01 and ISC00) in the MCU general Control Register (MCUCR) defines whether the external interrupt is activated on rising or falling edge of the INT0 pin or level sensed. Please note that INTF0 flag is not set when level sensitive interrupt condition is met. However, INT0 interrupt is generated, provided that INT0 mask bit is set in GIMSK register. 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 AT90S2333/4433 and always read as zero.

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 AT90S2333/4433 and always read as zero.

Timer/Counter Interrupt Mask Register - TIMSK

Bit	7	6	5	4	3	2	1	0	
$39 ($59)	TOIE1	OCIE1	-	-	TICIE1	-	TOIE0	-	TIMSK
Read/Write	R/W	R/W	R	R	R/W	R	R/W	R	
Initial value	0	0	0	0	0	0	0	0	

- **Bit 7 - TOIE1: Timer/Counter1 Overflow Interrupt Enable**

When the TOIE1 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter1 Overflow interrupt is enabled. The corresponding interrupt (at vector $005) is executed if an overflow in Timer/Counter1 occurs, i.e., when the TOV1 bit is set in the Timer/Counter Interrupt Flag Register - TIFR.

- **Bit 6 - OCIE1: Timer/Counter1 Output Compare Match Interrupt Enable**

When the OCIE1 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter1 Compare Match interrupt is enabled. The corresponding interrupt (at vector $004) is executed if a Compare match in Timer/Counter1 occurs, i.e., when the OCF1 bit is set in the Timer/Counter Interrupt Flag Register - TIFR.

- **Bit 5, 4 - Res: Reserved Bits**

These bits are reserved bits in the AT90S2333/4433 and always read as 0.

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

When the TICIE1 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter1 Input Capture Event Interrupt is enabled. The corresponding interrupt (at vector $003) is executed if a capture-triggering event occurs on pin 14, PB0 (ICP), i.e., when the ICF1 bit is set in the Timer/Counter Interrupt Flag Register - TIFR.

- **Bit 2 - Res: Reserved Bit**

This bit is a reserved bit in the AT90S2333/4433 and always reads as 0.

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

When the TOIE0 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter0 Overflow interrupt is enabled. The corresponding interrupt (at vector $006) is executed if an overflow in Timer/Counter0 occurs, i.e., when the TOV0 bit is set in the Timer/Counter Interrupt Flag Register - TIFR.

- **Bit 0 - Res: Reserved bit**

This bit is a reserved bit in the AT90S2333/4433 and always reads as zero.

Timer/Counter Interrupt Flag Register - TIFR

Bit	7	6	5	4	3	2	1	0	
$38 ($58)	TOV1	OCF1	-	-	ICF1	-	TOV0	-	TIFR
Read/Write	R/W	R/W	R	R	R/W	R	R/W	R	
Initial value	0	0	0	0	0	0	0	0	

- **Bit 7 - TOV1: Timer/Counter1 Overflow Flag**

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

- **Bit 6 - OCF1: Output Compare Flag 1**

The OCF1 bit is set (one) when compare match occurs between the Timer/Counter1 and the data in OCR1 - Output Compare Register 1. OCF1 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, OCF1 is cleared by writing a logic one to the flag. When the I-bit in SREG, and OCIE1 (Timer/Counter1 Compare match InterruptA Enable), and the OCF1 are set (one), the Timer/Counter1 Compare match Interrupt is executed.

- **Bit 5, 4 - Res: Reserved Bits**

These bits are reserved bits in the AT90S2333/4433 and always read as 0.

- **Bit 3 - 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. When the SREG I-bit, and TICIE1 (Timer/Counter1 Input Capture Interrupt Enable), and ICF1 are set (one), the Timer/Counter1 Capture Interrupt is executed.

- **Bit 2 - Res: Reserved Bit**

This bit is a reserved bit in the AT90S2333/4433 and always reads as 0.

- **Bit 1 - 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.

- **Bit 0 - Res: Reserved bit**

This bit is a reserved bit in the AT90S2333/4433 and always reads as zero.

External Interrupts

The external interrupts are triggered by the INT1 and INT0 pins. Observe that, if enabled, the interrupts will trigger even if the INT0/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 normally a relative jump to the interrupt 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, the Stack Pointer is incremented by 2, and the I flag in SREG is set. 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.

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	SM	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	

- **Bits 7, 6 - Res: Reserved bit**

These bits are reserved bits in the AT90S2333/4433 and always reads as zero.

- **Bit 5 - 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 4 - SM: Sleep Mode**

This bit selects between the two available sleep modes. When SM is cleared (zero), Idle Mode is selected as sleep mode. When SM is set (one), Power Down Mode is selected as Sleep Mode. For details, refer to the paragraph “Sleep Modes” below.

- **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	Any logical change on INT1 generates an interrupt request.
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 value on the INT1 pin is sampled before detecting edges. If edge or toggle interrupt is selected, pulses that last longer than one clock period will generate an interrupt. Shorter pulses are not guaranteed to generate an interrupt. If low level interrupt is selected, the low level must be held until the completion of the currently executing instruction to generate an interrupt

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

The External Interrupt 0 is activated by the external pin INT0 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	Any logical change on INT0 generates an interrupt request.
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 ISC01/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.

The value on the INT0 pin is sampled before detecting edges. If edge or toggle interrupt is selected, pulses that last longer than one clock period will generate an interrupt. Shorter pulses are not guaranteed to generate an interrupt. If low level interrupt is selected, the low level must be held until the completion of the currently executing instruction to generate an interrupt

Sleep Modes

To enter the sleep modes, the SE bit in MCUCR must be set (one) and a SLEEP instruction must be executed. The SM bit in the MCUCR register selects which sleep mode (Idle or Power Down) 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. On wake-up from Power Down Mode on pin change, the two instructions following SLEEP are executed before the pin change interrupt routine. The contents of the register file and I/O memory are unaltered. If a reset occurs during sleep mode, the MCU wakes up and executes from the Reset vector.

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

Idle Mode

When the SM bit is cleared (zero), 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 Timer Overflow interrupt and watchdog reset. 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.

Power Down Mode

When the SM bit is set (one), the SLEEP instruction forces the MCU into the Power Down Mode. In this mode, the external oscillator is stopped, while the external interrupts and the Watchdog (if enabled) continue operating. Only an external reset, a watchdog reset (if enabled), an external level interrupt, or a pin change interrupt can wake up the MCU.

Note that if a level triggered or pin change interrupt is used for wake-up from Power Down Mode, the changed level must be held for a time to wake up the MCU. This makes the MCU less sensitive to noise. The wake-up period is equal to the clock-counting part of the reset period (see Table 5). The MCU will wake up from power-down if the input has the required level for two watchdog oscillator cycles. If the wake up period is shorter than two watchdog oscillator cycles, the MCU will wake up if the input has the required level for the duration of the wake-up period. If the wake-up condition disappears before the wake-up period has expired, the MCU will wake up from power down without executing the corresponding interrupt.

The period of the watchdog oscillator is 2.7µs (nominal) at 3.0V and 25°C. The frequency of the watchdog oscillator is voltage dependent as shown in the Electrical Characteristics section.

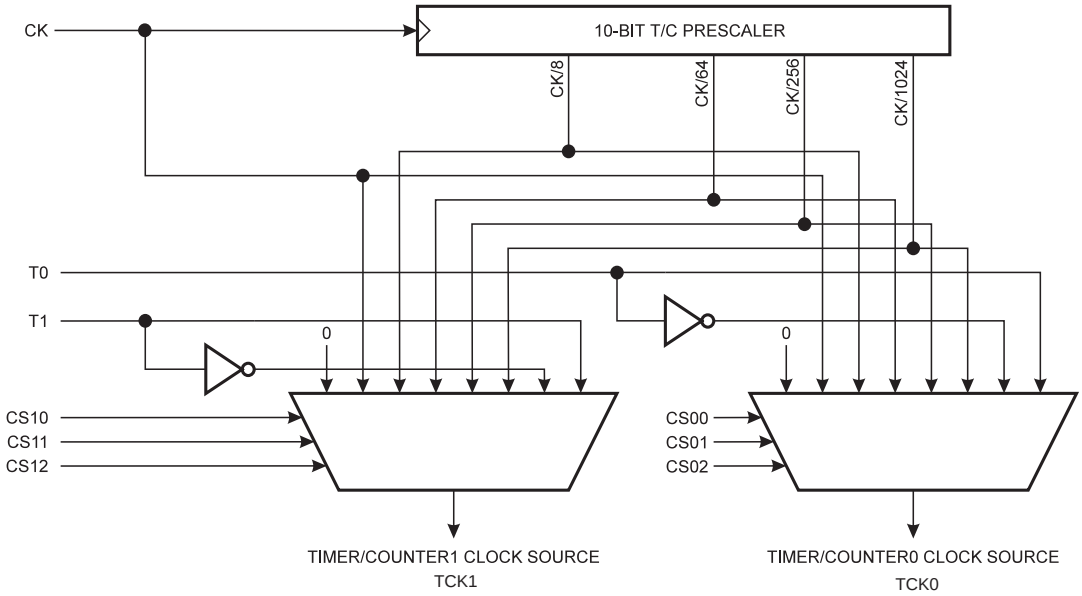
When waking up from Power Down Mode, there is a delay from the wake-up condition occurs until the wake-up becomes effective. This allows the clock to restart and become stable after having been stopped. The wake-up period is defined by the same CKSEL fuses that define the reset time-out period.

Timer / Counters

The AT90S2333/4433 provides two general purpose Timer/Counters - one 8-bit T/C and one 16-bit T/C. Timer/Counters 0 and 1 have individual prescaling selection from the same 10-bit prescaling timer. 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.

Timer/Counter Prescaler

Figure 30. 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, external source and stop can also be selected as clock sources.

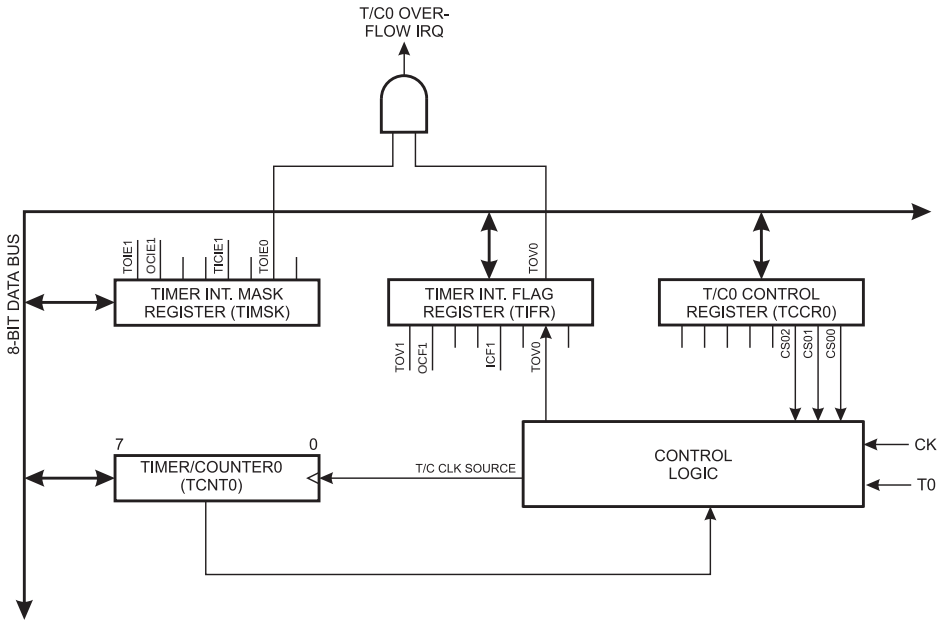
8-bit Timer/Counter0

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 31 shows the block diagram for Timer/Counter0.

Figure 31. Timer/Counter0 Block Diagram



Timer/Counter0 Control Register - TCCR0

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

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

These bits are reserved bits in the AT90S2333/4433 and always read as zero.

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

The Clock Select0 bits 2,1, and 0 define the prescaling source of 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, for Timer/Counter0, transitions on PD4/(T0) will clock the counter even if the pin is configured as an output. This feature can give the user SW control of the counting.

Timer Counter 0 - TCNT0

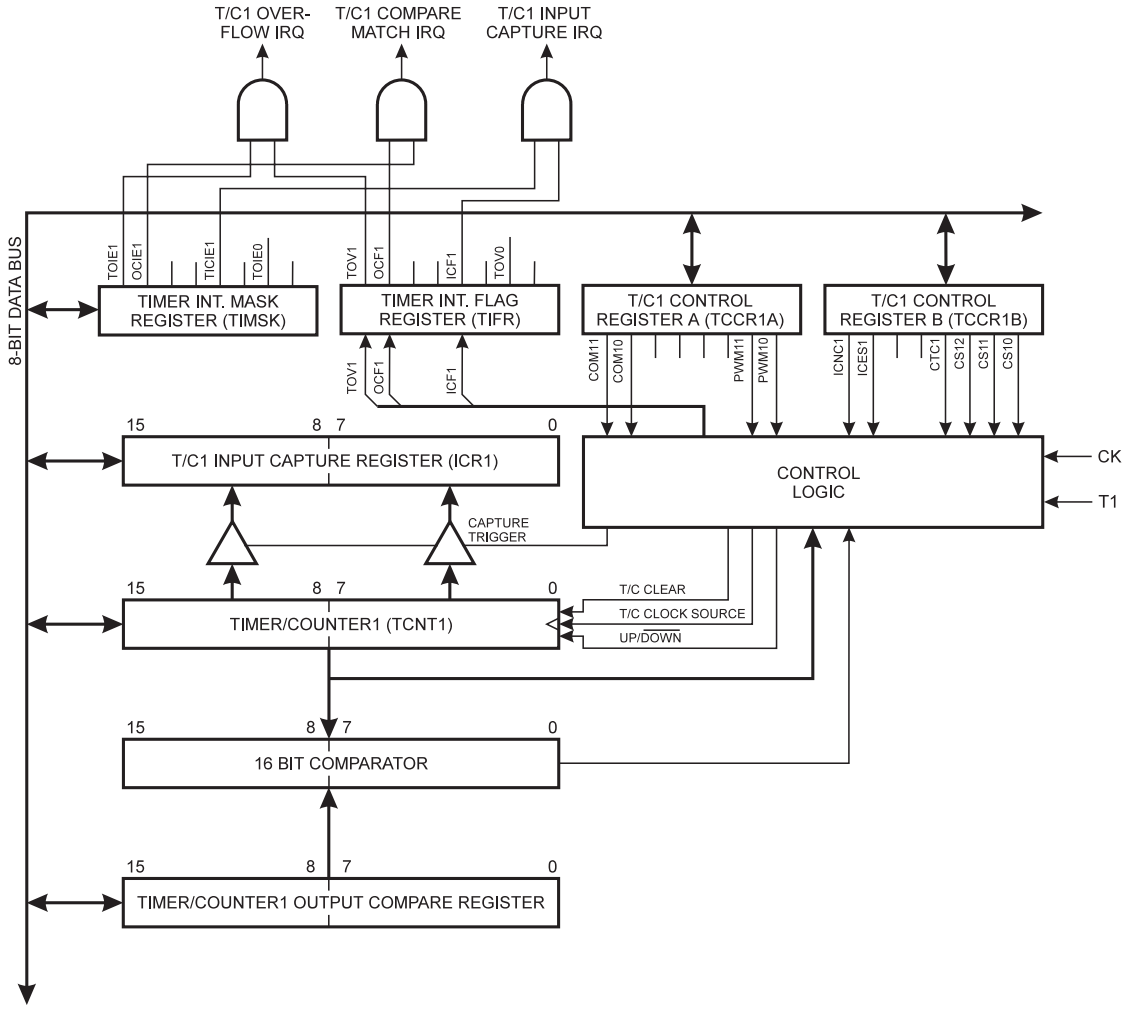
Bit	7	6	5	4	3	2	1	0	
$32 ($52)	MSB							LSB	TCNT0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

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.

16-bit Timer/Counter1

Figure 32 shows the block diagram for Timer/Counter1.

Figure 32. 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 Register - TCCR1A. The different status flags (overflow, compare match and capture event) and control signals are found in the Timer/Counter Interrupt Flag Register - TIFR. 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.

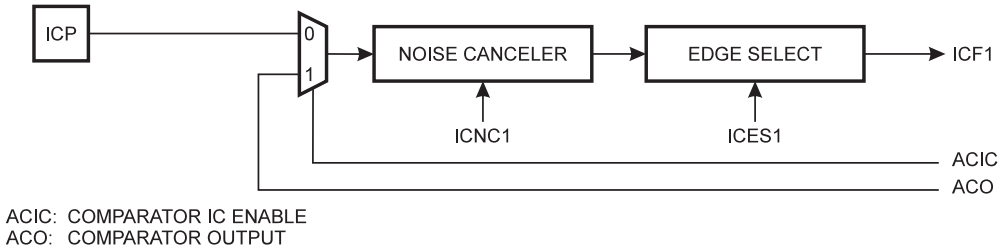
The 16-bit Timer/Counter1 features both a high resolution and a high accuracy usage with the lower prescaling opportunities. Similarly, the high prescaling opportunities makes the Timer/Counter1 useful for lower speed functions or exact timing functions with infrequent actions.

The Timer/Counter1 supports an Output Compare function using the Output Compare Register 1 - OCR1 as the data source to be compared to the Timer/Counter1 contents. The Output Compare functions include optional clearing of the counter on compare matches, and actions on the Output Compare pin 1 on 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 OCR1 register serve as a glitch-free stand-alone PWM with centered pulses. Refer to page 34 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 - TCCR1. 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 33.

Figure 33. ICP Pin Schematic Diagram



If the noise canceler function is enabled, the actual trigger condition for the capture event is monitored over 4 samples, and all 4 must be equal to activate the capture flag. The input pin signal is sampled at XTAL clock frequency.

Timer/Counter1 Control Register A - TCCR1A

Bit	7	6	5	4	3	2	1	0	
$2F ($4F)	**COM11**	**COM10**	**-**	**-**	**-**	**-**	**PWM11**	**PWM10**	**TCCR1A**
Read/Write	R/W	R/W	R	R	R	R	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

- **Bits 7,6 - COM11, COM10: Compare Output Mode1, bits 1 and 0**

The COM11 and COM10 control bits determine any output pin action following a compare match in Timer/Counter1. Any output pin actions affect pin OC1 - Output Compare pin 1. This is an alternative function to an I/O port, and the corresponding direction control bit must be set (one) to control an output pin. The control configuration is shown in Table 10.

Table 10. Compare 1 Mode Select

COM11	COM10	Description
0	0	Timer/Counter1 disconnected from output pin OC1
0	1	Toggle the OC1 output line.
1	0	Clear the OC1 output line (to zero).
1	1	Set the OC1 output line (to one).

In PWM mode, these bits have a different function. Refer to Table 11 for a detailed description.

- **Bits 5..2 - Res: Reserved bits**

These bits are reserved bits in the AT90S2333/4433 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 34.

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

Timer/Counter1 Control Register B - TCCR1B

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

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

When the ICNC1 bit is cleared (zero), the input capture trigger noise canceler function is disabled. The input capture is triggered at the first rising/falling edge sampled on the ICP - input capture pin - as specified. When the ICNC1 bit is set (one), four successive samples are measured on the ICP - 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 the 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 AT90S2333/4433 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 compare 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 compare register is set to C, the timer will count as follows if CTC1 is set:

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

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

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

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 prescaled CK modes are scaled directly from the CK oscillator clock. If the external pin modes are used for Timer/Counter0, transitions on PD5/(T1) will clock the counter even if the pin is configured as an output. This feature can give the user SW control of the counting.

Timer/Counter1 - TCNT1H and TCNT1L

Bit	15	14	13	12	11	10	9	8	
$2D ($4D)	**MSB**								TCNT1H
$2C ($4C)								**LSB**	TCNT1L
	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	

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 OCR1 and ICR1. If the main program and also interrupt routines perform access to registers using TEMP, interrupts must be disabled during access from the main program (and from interrupt routines if interrupts are allowed from within interrupt routines).

- **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. Consequently, 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 - OCR1H and OCR1L

Bit	15	14	13	12	11	10	9	8	
$2B ($4B)	**MSB**								OCR1H
$2A ($4A)								**LSB**	OCR1L
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

The output compare register is a 16-bit read/write register.

The Timer/Counter1 Output Compare Register contains the data to be continuously compared with Timer/Counter1. Actions on compare matches are specified in the Timer/Counter1 Control and Status register.

Since the Output Compare Register - OCR1 - is a 16-bit register, a temporary register TEMP is used when OCR1 is written to ensure that both bytes are updated simultaneously. When the CPU writes the high byte, OCR1H, the data is temporarily stored in the TEMP register. When the CPU writes the low byte, OCR1L, the TEMP register is simultaneously written to OCR1H. Consequently, the high byte OCR1H 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.

Timer/Counter1 Input Capture Register - ICR1H and ICR1L

Bit	15	14	13	12	11	10	9	8	
$27 ($47)	**MSB**								ICR1H
$26 ($46)								**LSB**	ICR1L
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R	
	R	R	R	R	R	R	R	R	
Initial value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

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

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

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

The TEMP register is also used when accessing TCNT1 and OCR1. 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 Register1 - OCR1, form a 8, 9, or 10-bit, free-running, glitch-free, and phase correct PWM with output on the PB1(OC1) pin. Timer/Counter1 acts as an up/down counter, counting up from $0000 to TOP (see Table 13), where it turns and counts down again to zero before the cycle is repeated. When the counter value matches the contents of the 8, 9 or 10 least significant bits of OCR1, the PB1(OC1) pin

is set or cleared according to the settings of the COM11 and COM10 bits in the Timer/Counter1 Control Register TCCR1. Refer to Table 14 for details.

Table 13. Timer TOP Values and PWM Frequency

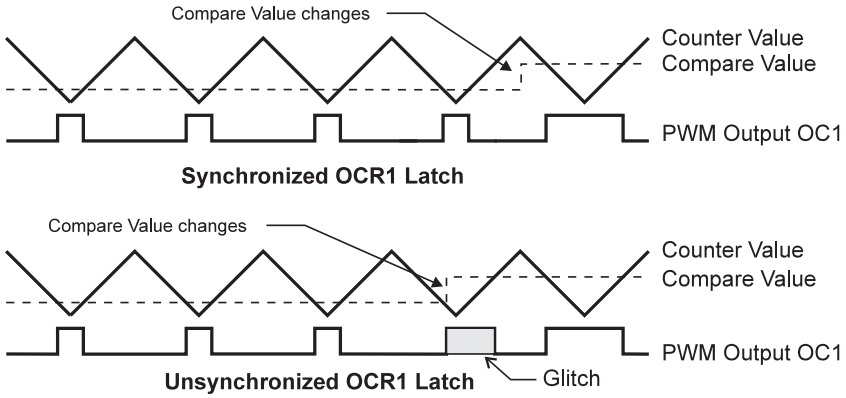
PWM Resolution	Timer TOP value	Frequency
8-bit	$00FF (255)	f_{TCK1}/510
9-bit	$01FF (511)	f_{TCK1}/1022
10-bit	$03FF(1023)	f_{TCK1}/2046

Table 14. Compare1 Mode Select in PWM Mode

COM11	COM10	Effect on OC1
0	0	Not connected
0	1	Not connected
1	0	Cleared on compare match, up-counting. Set on compare match, down-counting (non-inverted PWM).
1	1	Cleared on compare match, down-counting. Set on compare match, up-counting (inverted PWM).

Note that in the PWM mode, the 10 least significant OCR1 bits, when written, are transferred to a temporary location. They are latched when Timer/Counter1 reaches TOP. This prevents the occurrence of odd-length PWM pulses (glitches) in the event of an unsynchronized OCR1 write. See Figure 34 for an example.

Figure 34. Effects on Unsynchronized OCR1 Latching



During the time between the write and the latch operation, a read from OCR1 will read the contents of the temporary location. This means that the most recently written value always will read out of OCR1.

When OCR1 contains $0000 or TOP, the output OC1 is updated to low or high on the next compare match according to the settings of COM11 and COM10. This is shown in Table 15.

Table 15. PWM Outputs OCR = $0000 or TOP

COM11	COM10	OCR1	Output OC1
1	0	$0000	L
1	0	TOP	H
1	1	$0000	H
1	1	TOP	L

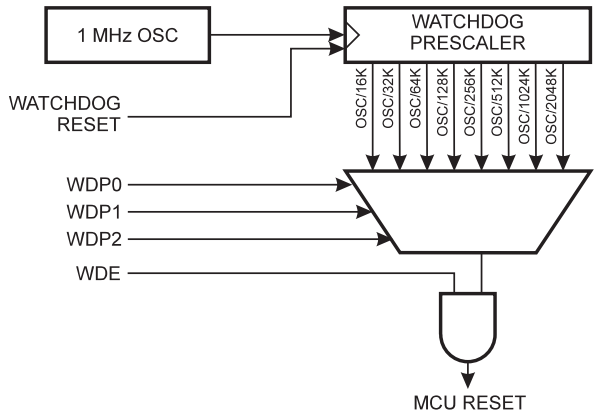
In PWM mode, the Timer Overflow Flag1, TOV1, is set when the counter changes direction at $0000. Timer Overflow Interrupt1 operates exactly as in normal Timer/Counter mode, i.e. it is executed when TOV1 is set provided that Timer Overflow Interrupt1 and global interrupts are enabled. This also applies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

Watchdog Timer

The Watchdog Timer is clocked from a separate on-chip oscillator. By controlling the Watchdog Timer prescaler, the Watchdog reset interval can be adjusted as shown in Table 6. See characterization data for typical values at other V_{CC} levels. 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 AT90S2333/4433 resets and executes from the reset vector. For timing details on the Watchdog reset, refer to page 22.

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 35. Watchdog Timer



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 AT90S2333/4433 and will always read as zero.

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

This bit must be set (one) 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 time-out Periods are shown inTable 16. .

Table 16. Watch Dog Timer Prescale Select

WDP2	WDP1	WDP0	Number of WDT Oscillator cycles	Typical time-out at V_{CC} = 3.0V	Typical time-out at V_{CC} = 5.0V
0	0	0	16K cycles	47 ms	15 ms
0	0	1	32K cycles	94 ms	30 ms
0	1	0	64K cycles	0.19 s	60 ms
0	1	1	128K cycles	0.38 s	0.12 s
1	0	0	256K cycles	0.75 s	0,24 s
1	0	1	512K cycles	1.5 s	0.49 s
1	1	0	1,024K cycles	3.0 s	0.97 s
1	1	1	2,048K cycles	6.0 s	1.9 s

Note: The frequency of the watchdog oscillator is voltage dependent as shown in the Electrical Characteristics section.
The WDR - Watchdog Reset - instruction should always be executed before the Watchdog Timer is enabled. This ensures that the reset period will be in accordance with the Watchdog Timer prescale settings. If the Watchdog Timer is enabled without reset, the watchdog timer may not start counting from zero.

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.

An ongoing EEPROM write operation will complete even if a reset condition occurs.

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

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

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

EEPROM Address Register - EEAR

Bit	7	6	5	4	3	2	1	0	
$1E ($3E)	**EEAR7**	**EEAR6**	**EEAR5**	**EEAR4**	**EEAR3**	**EEAR2**	**EEAR1**	**EEAR0**	**EEAR**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	X	X	X	X	X	X	X	X	

The EEPROM Address Register - EEAR specifies the EEPROM address in the 128/256 bytes EEPROM space. The EEPROM data bytes are addressed linearly between 0 and 127/255. The initial value of EEAR is undefined. A proper value must be written before the EEPROM may be accessed.

EEPROM Data Register - EEDR

Bit	7	6	5	4	3	2	1	0	
$1D ($3D)	**MSB**							**LSB**	**EEDR**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

- **Bits 7..0 - EEDR7.0: EEPROM Data**

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

EEPROM Control Register - EECR

Bit	7	6	5	4	3	2	1	0	
$1C ($3C)	**-**	**-**	**-**	**-**	**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 AT90S2333/4433 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 EEAR (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.

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

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 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):

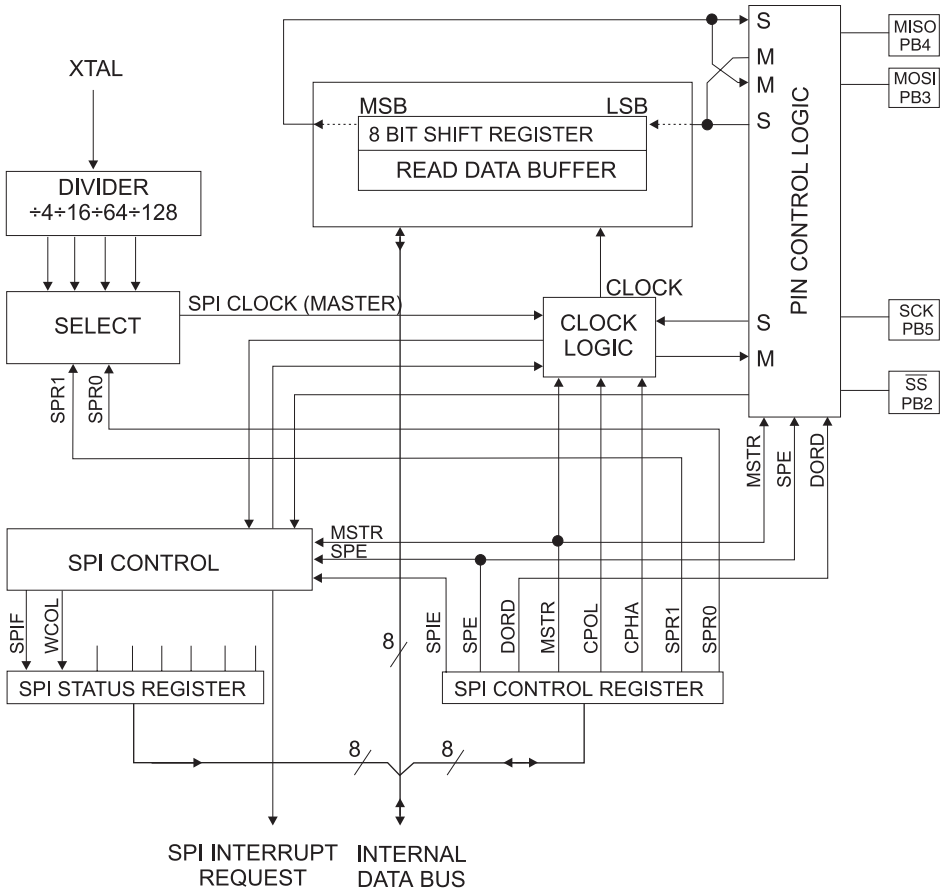
1. Keep the AVR RESET active (low) during periods of insufficient power supply voltage. This can be done by enabling the internal Brown-Out Detector (BOD) if the operating speed matches the detection level. If not, an external low V_{CC} Reset Protection circuit can be applied.
2. Keep the AVR core in Power Down Sleep Mode during periods of low V_{CC}. This will prevent the CPU from attempting to decode and execute instructions, effectively protecting the EEPROM registers from unintentional writes.
3. Store constants in Flash memory if the ability to change memory contents from software is not required. Flash memory can not be updated by the CPU, and will not be subject to corruption.

Serial Peripheral Interface - SPI

The Serial Peripheral Interface (SPI) allows high-speed synchronous data transfer between the AT90S2333/4433 and peripheral devices or between several AVR devices. The AT90S2333/4433 SPI features include the following:

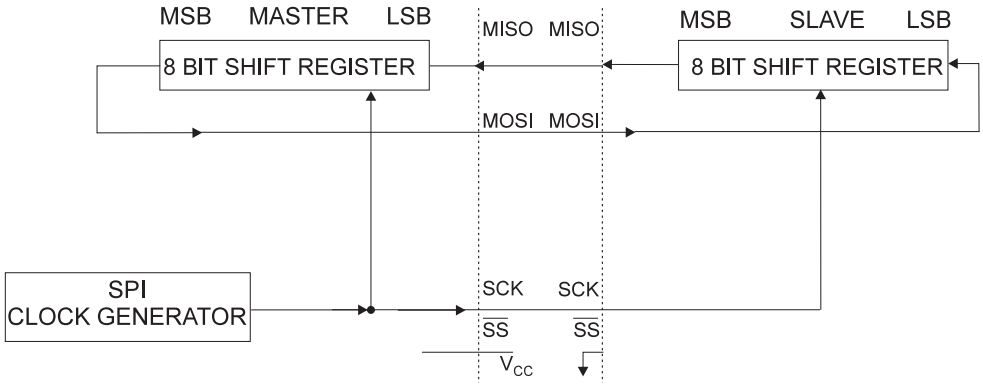
- Full-Duplex, 3-Wire Synchronous Data Transfer
- Master or Slave Operation
- LSB First or MSB First Data Transfer
- Four Programmable Bit Rates
- End of Transmission Interrupt Flag
- Write Collision Flag Protection
- Wake-up from Idle Mode

Figure 36. SPI Block Diagram



The interconnection between master and slave CPUs with SPI is shown in Figure 37. The PB5(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 PB3(MOSI) pin and into the PB3(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, PB2($\overline{SS}$), is set low to select an individual slave SPI device. 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 37. When data is shifted from the master to the slave, data is also shifted in the opposite direction, simultaneously. This means that during one shift cycle, data in the master and the slave are interchanged.

Figure 37. SPI Master-Slave Interconnection



The system is single buffered in the transmit direction and double buffered in the receive direction. This means that bytes to be transmitted cannot be written to the SPI Data Register before the entire shift cycle is completed. When receiving data, however, a received byte must be read from the SPI Data Register before the next byte has been completely shifted in. Otherwise, the first byte 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 17. SPI Pin Direction Overrides

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

Note: See “Alternate Functions Of Port B” on page 60 for a detailed description og how to define the direction of the user defined SPI pins.

$\overline{SS}$ 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 hold high to ensure Master SPI operation. If the SS pin is driven low by peripheral circuitry when the SPI is configured as master with the SS pin defined as an input, the SPI system interprets this as another master selecting the SPI as a slave and starting to send data to it. To avoid bus contention, the SPI system takes the following actions:

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

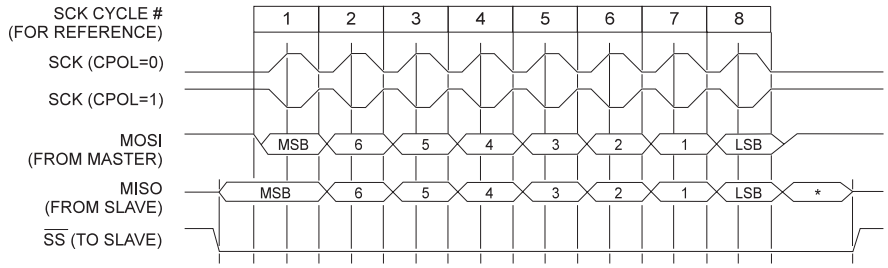
Thus, when interrupt-driven SPI transmittal is used in master mode, and there exists a possibility that $\overline{SS}$ is driven low, the interrupt should always check that the MSTR bit is still set. Once the MSTR bit has been cleared by a slave select, it must be set by the user to re-enale the SPI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

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

Data Modes

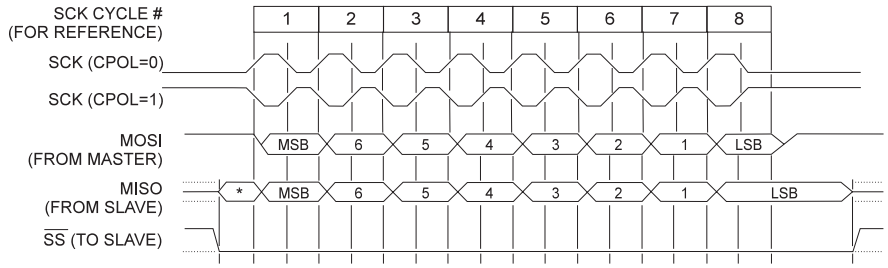
There are four combinations of SCK phase and polarity with respect to serial data, which are determined by control bits CPHA and CPOL. The SPI data transfer formats are shown in Figure 38 and Figure 39.

Figure 38. SPI Transfer Format with CPHA = 0 and DORD = 0



* Not defined but normally MSB of character just received

Figure 39. SPI Transfer Format with CPHA = 1 and DORD = 0



* Not defined but normally LSB of previously transmitted character

SPI Control Register - SPCR

Bit	7	6	5	4	3	2	1	0	
$0D ($2D)	**SPIE**	**SPE**	**DORD**	**MSTR**	**CPOL**	**CPHA**	**SPR1**	**SPR0**	**SPCR**
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 the SPI interrupt to be executed if SPIF bit in the SPSR register is set and the 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 38 and Figure 39 for additional information.

- **Bit 2 - CPHA: Clock PHAse**

Refer to Figure 38 or Figure 39 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 18. 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

SPI Status Register - SPSR

Bit	7	6	5	4	3	2	1	0	
$0E ($2E)	**SPIF**	**WCOL**	**-**	**-**	**-**	**-**	**-**	**-**	**SPSR**
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. 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 AT90S2333/4433 and will always read as zero.

The SPI interface on the AT90S2333/4433 is also used for program memory and EEPROM downloading or uploading. See page 78 for serial programming and verification.

SPI Data Register - SPDR

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

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

UART

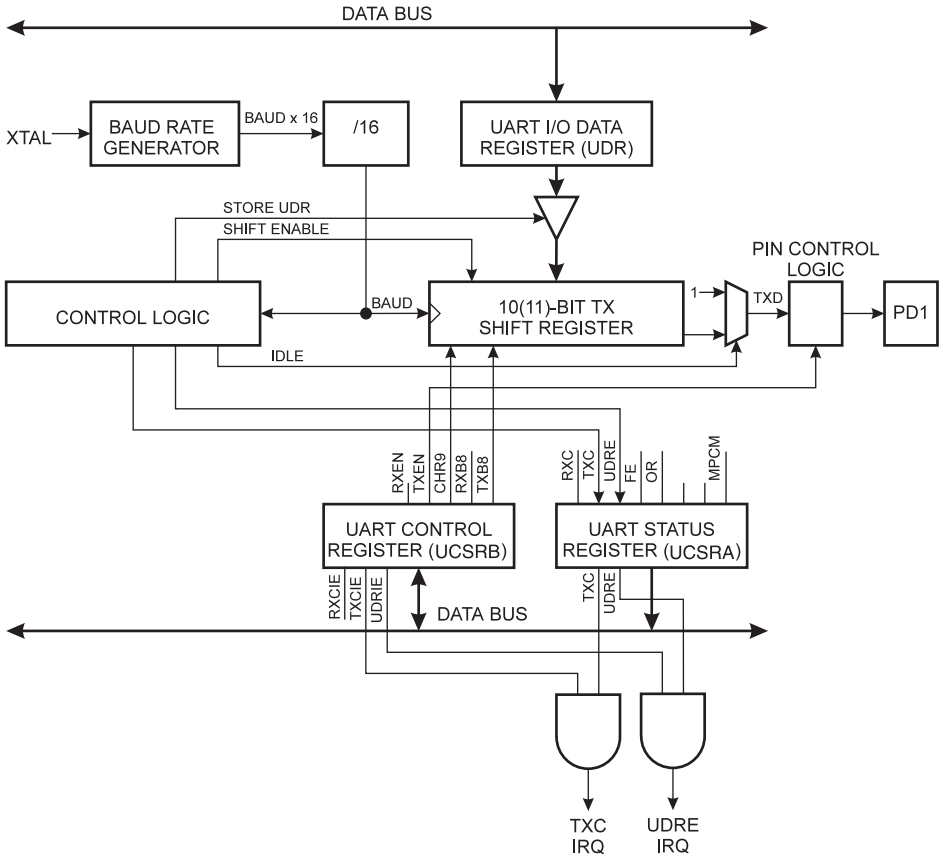
The AT90S2333/4433 features a full duplex (separate receive and transmit registers) Universal Asynchronous Receiver and Transmitter (UART). The main features are:

- 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
- Multi-Processor Communication Mode

Data Transmission

A block schematic of the UART transmitter is shown in Figure 40.

Figure 40. 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.

When data is transferred from UDR to the shift register, 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.

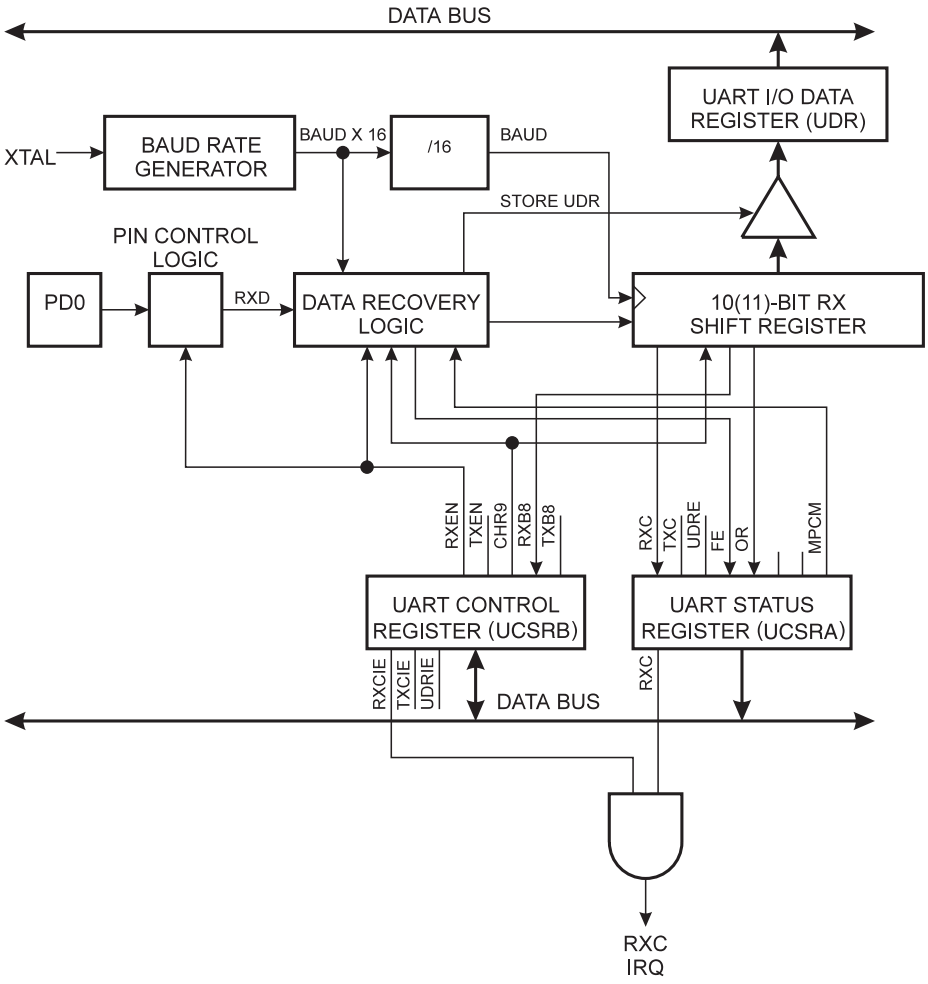
On the Baud Rate clock following the transfer operation to the shift register, the start bit is shifted out on the TXD pin. Then follows the data, LSB first. When the stop bit has been shifted out, the shift register is loaded if any new data has been written to the UDR during the transmission. During 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 41 shows a block diagram of the UART Receiver

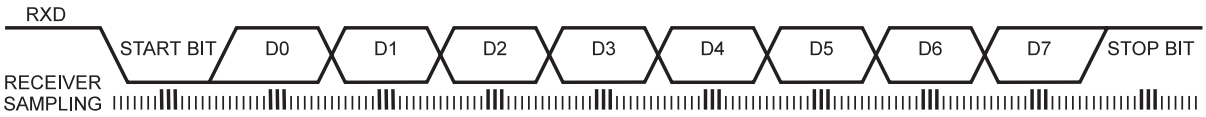
Figure 41. UART Receiver



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 samples the RXD pin at samples 8, 9 and 10. If two or more of these three samples are found to be logical ones, the start bit is rejected as a noise spike and the receiver starts looking for the next 1 to 0-transition.

If however, a valid start bit is detected, sampling of the data bits following the start bit is performed. These bits are also sampled at samples 8, 9 and 10. The logical value found in at least two of the three samples is taken as the bit value. All bits are shifted into the transmitter shift register as they are sampled. Sampling of an incoming character is shown in Figure 42.

Figure 42. Sampling Received Data



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

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

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

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

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

Multi-Processor Communication Mode

The Multi-Processor Communication Mode enables several slave MCUs to receive data from a master MCU. This is done by first decoding an address byte to find out which MCU has been addressed. If a particular slave MCU has been addressed, it will receive the following data bytes as normal, while the other slave MCUs will ignore the data bytes until another address byte is received.

For an MCU to act as a master MCU, it should enter 9-bit transmission mode (CHR9 in UCSRB set). The 9th bit must be one to indicate that an address byte is being transmitted, and zero to indicate that a data byte is being transmitted.

For the slave MCUs, the mechanism appears slightly differently for 8-bit and 9-bit reception mode. In 8-bit reception mode (CHR9 in UCSRB cleared), the stop bit is one for an address byte and zero for a data byte. In 9-bit reception mode (CHR9 in UCSRB set), the 9th bit is one for an address byte and zero for a data byte, whereas the stop bit is always high.

The following procedure should be used to exchange data in Multi-Processor Communication Mode:

1. All slave MCUs are in Multi-Processor Communication Mode (MPCM in UCSRA is set).
2. The master MCU sends an address byte, and all slaves receive and read this byte. In the slave MCUs, the RXC flag in UCSRA will be set as normal.

3. Each slave MCU reads the UDR register and determines if it has been selected. If so, it clears the MPCM bit in UCSRA, otherwise it waits for the next address byte.
4. For each received data byte, the receiving MCU will set the receive complete flag (RXC in UCSRA). In 8-bit mode, the receiving MCU will also generate a framing error (FE in UCSRA set), since the stop bit is zero. The other slave MCUs, which still have the MPCM bit set, will ignore the data byte. In this case, the UDR register and the RXC or FE flags will not be affected.
5. After the last byte has been transferred, the process repeats from step 2.

UART Control

UART I/O Data Register - UDR

Bit	7	6	5	4	3	2	1	0	
$0C ($2C)	**MSB**							**LSB**	**UDR**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

The UDR register is actually two physically separate registers sharing the same I/O address. When writing to the register, the UART Transmit Data register is written. When reading from UDR, the UART Receive Data register is read.

UART Control and Status Registers - UCSRA

Bit	7	6	5	4	3	2	1	0	
$0B ($2B)	**RXC**	**TXC**	**UDRE**	**FE**	**OR**	**-**	**-**	**MPCM**	**UCSRA**
Read/Write	R	R/W	R	R	R	R	R	R/W	
Initial value	0	0	1	0	0	0	0	0	

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

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

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

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

When the TXCIE bit in UCR is set, setting of TXC causes the UART Transmit Complete interrupt to be executed. TXC is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, the TXC bit is cleared (zero) by writing a logical 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 transmittal is used, the UART Data Register Empty Interrupt routine must write UDR in order to clear UDRE, otherwise a new interrupt will occur once the interrupt routine terminates.

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

- **Bit 4 - FE: Framing Error**

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

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 UDRE is read.

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

- **Bits 2..1 - Res: Reserved bits**

These bits are reserved bits in the AT90S2333/4433 and will always read as zero.

- **Bit 0 - MPCM: Multi-Processor Communication Mode**

This bit is used to enter Multi-Processor Communication Mode. The bit is set when the slave MCU waits for an address byte to be received. When the MCU has been addressed, the MCU switches off the MPCM bit, and starts data reception.

For a detailed description, see “Multi-Processor Communication Mode”.

UART Control and Status Registers - UCSRB

Bit	7	6	5	4	3	2	1	0	
$0A ($2A)	**RXCIE**	**TXCIE**	**UDRIE**	**RXEN**	**TXEN**	**CHR9**	**RXB8**	**TXB8**	**UCSRB**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	W	
Initial value	0	0	0	0	0	0	1	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, respectively. The 9th data bit can be used as an extra stop bit or a parity bit.

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

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

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

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

Baud Rate Generator

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

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

- BAUD = Baud-Rate
- f_{CK}= Crystal Clock frequency
- UBR = Contents of the UBRRH and UBRR registers, (0-4095)

For standard crystal frequencies, the most commonly used baud rates can be generated by using the UBR settings in Table 19. UBR values which yield an actual baud rate differing less than 2% from the target baud rate, are bold in the table. However, using baud rates that have more than 1% error is not recommended. High error ratings give less noise resistanse.

Table 19. UBR Settings at Various Crystal Frequencies

Baud Rate	1 MHz		% Error	1.8432 MHz		% Error	2 MHz		% Error
2400	UBR=	**25**	**0.2**	UBR=	**47**	**0.0**	UBR=	**51**	**0.2**
4800	UBR=	**12**	**0.2**	UBR=	**23**	**0.0**	UBR=	**25**	**0.2**
9600	UBR=	6	7.5	UBR=	**11**	**0.0**	UBR=	**12**	**0.2**
14400	UBR=	3	7.8	UBR=	**7**	**0.0**	UBR=	8	3.7
19200	UBR=	2	7.8	UBR=	**5**	**0.0**	UBR=	6	7.5
28800	UBR=	1	7.8	UBR=	**3**	**0.0**	UBR=	3	7.8
38400	UBR=	1	22.9	UBR=	**2**	**0.0**	UBR=	2	7.8
57600	UBR=	0	7.8	UBR=	**1**	**0.0**	UBR=	1	7.8
76800	UBR=	0	22.9	UBR=	1	33.3	UBR=	1	22.9
115200	UBR=	0	84.3	UBR=	**0**	**0.0**	UBR=	0	7.8

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

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

UART Baud Rate Register - UBRR

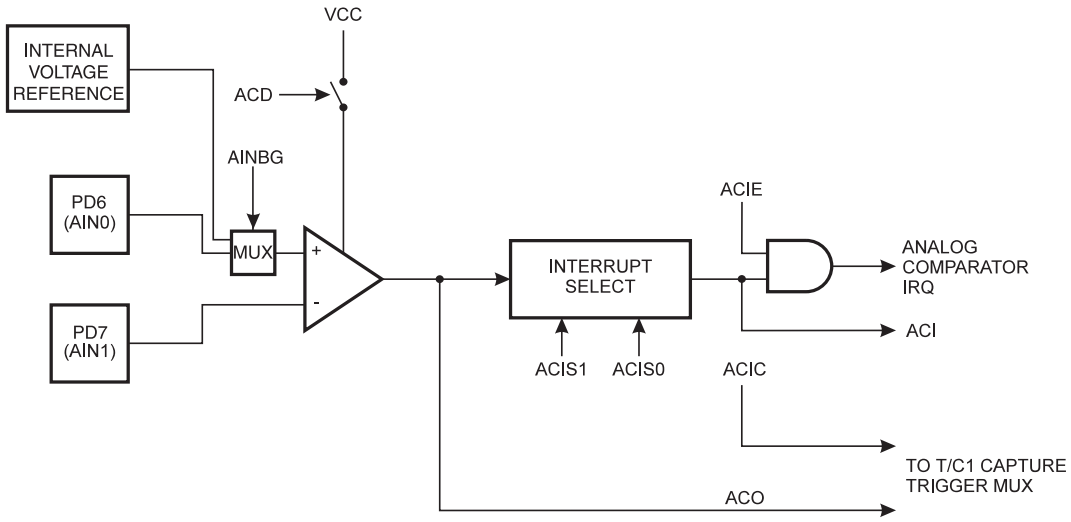
Bit	15	14	13	12	11	10	9	8	
$03 ($23)	-	-	-	-	MSB			LSB	UBRRHI
$09 ($29)	MSB							LSB	UBRR
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

This is a 12-bit register which contains the UART Baud Rate according to the equation on the previous page. The UBRRHI contains the 4 most significant bits, and the UBRR contains the 8 least significant bits of the UART Baud Rate.

Analog Comparator

The analog comparator compares the input values on the positive input PD6 (AIN0) and negative input PD7 (AIN1). When the voltage on the positive input PD6 (AIN0) is higher than the voltage on the negative input PD7 (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 43.

Figure 43. Analog Comparator Block Diagram



Analog Comparator Control And Status Register - ACSR

Bit	7	6	5	4	3	2	1	0	
$08 ($28)	ACD	AINBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	ACSR
Read/Write	R/W	R/W	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 - AINBG: Analog Comparator Bandgap Select**

When this bit is set BOD is enabled and the BODEN is programmed, a fixed bandgap voltage of 1.22 ± 0.05V replaces the normal input to the positive input (AIN0) of the comparator. When this bit is cleared, the normal input pin PD6 is applied to the positive input of the comparator.

- **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 20.

Table 20. 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.

Caution: Using the SBI or CBI intruction on other bits than ACI in this register, will write a one back into ACI if it is read as set, thus clearing the flag.

Analog to Digital Converter

Feature list:

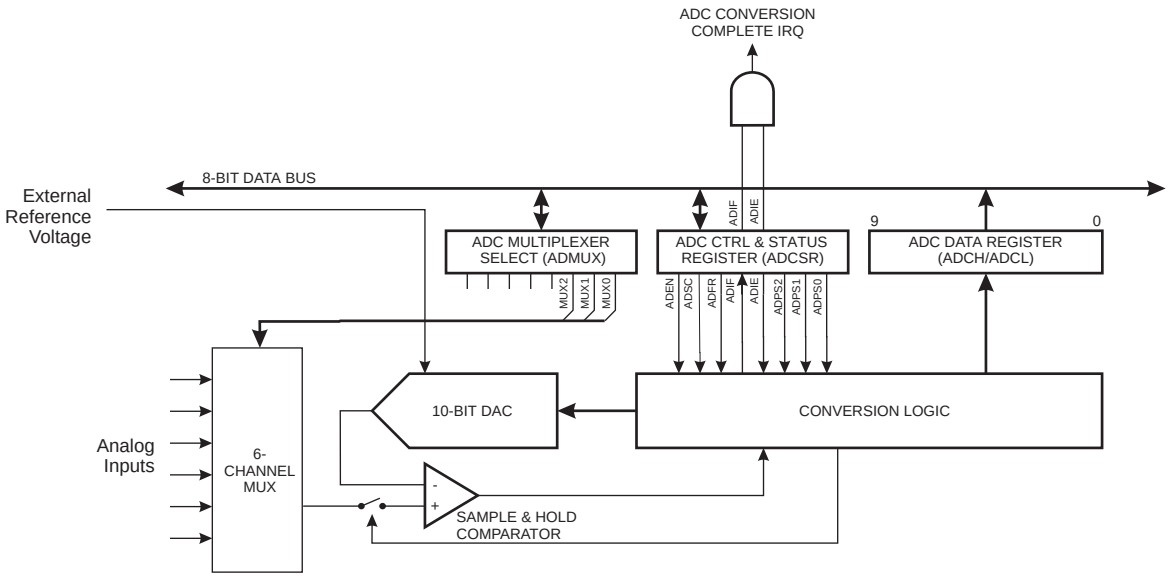
- 10-bit Resolution
- ± 2 LSB Absolute Accuracy
- 0.5 LSB Integral Non-Linearity
- 65 - 260 µs Conversion Time
- Up to 15 kSPS
- 6 Multiplexed Input Channels
- Rail-to-Rail Input Range
- Free Run or Single Conversion Mode
- Interrupt on ADC conversion complete.
- Sleep Mode Noise Canceler

The AT90S2333/4433 features a 10-bit successive approximation ADC. The ADC is connected to a 6-channel Analog Multiplexer which allows each pin of Port C 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 44.

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 ± 0.3 V 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 - AVCC.

Figure 44. Analog to Digital Converter Block Schematic



Operation

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

The ADMUX register selects which one of the six analog input channels to be used as input to the ADC.

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 only difference will be that this conversion takes 12 more clock cycles than a normal conversion.

A conversion is started by writing a logical one to the ADC Start Conversion bit, ADSC. This bit will stay high as long as the conversion is in progress and 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.

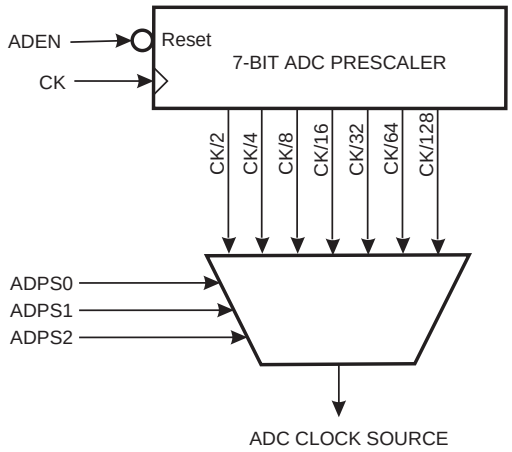
As the ADC generates a 10-bit result, two data registers, ADCH and ADCL, must be read to get the result when the conversion is complete. Special data protection logic is used to ensure that the contents of the data registers belong to the same result when they are read. This mechanism works as follows:

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, ADIF, 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

Figure 45. ADC Prescaler



The ADC contains a prescaler, which divides the system clock to an acceptable ADC clock frequency. The ADC accepts input clock frequencies in the range 50 - 200 kHz. Applying a higher input frequency will result in a poorer accuracy, see “ADC Characteristics” on page 58.

The ADPS0 - ADPS2 bits in ADCSR are used to generate a proper ADC clock input frequency from any XTAL frequency above 100 kHz. The prescaler starts counting from the moment the ADC is switched on by setting the ADEN bit in ADCSR. The prescaler keeps running for as long as the ADEN bit is set, and is continuously reset when ADEN is low.

When initiating a conversion by setting the ADSC bit in ADCSR, the conversion starts at the following rising edge of the ADC clock cycle. The actual sample-and-hold takes place 1.5 ADC clock cycles after the start of the conversion. The result is ready and written to the ADC Result Register after 13 cycles. In single conversion mode, the ADC needs one more clock cycle before a new conversion can be started, see Figure 47. If ADSC is set high in this period, the ADC will start the new conversion immediately. In Free Run Mode, a new conversion will be started immediately after the result is written to the ADC Result Register. Using Free Run Mode and an ADC clock frequency of 200 kHz gives the lowest conversion time, 65 µs, equivalent to 15.4 kSPS. For a summary of conversion times, see Table 21.

Figure 46. ADC Timing Diagram, First Conversion (Single Conversion Mode)

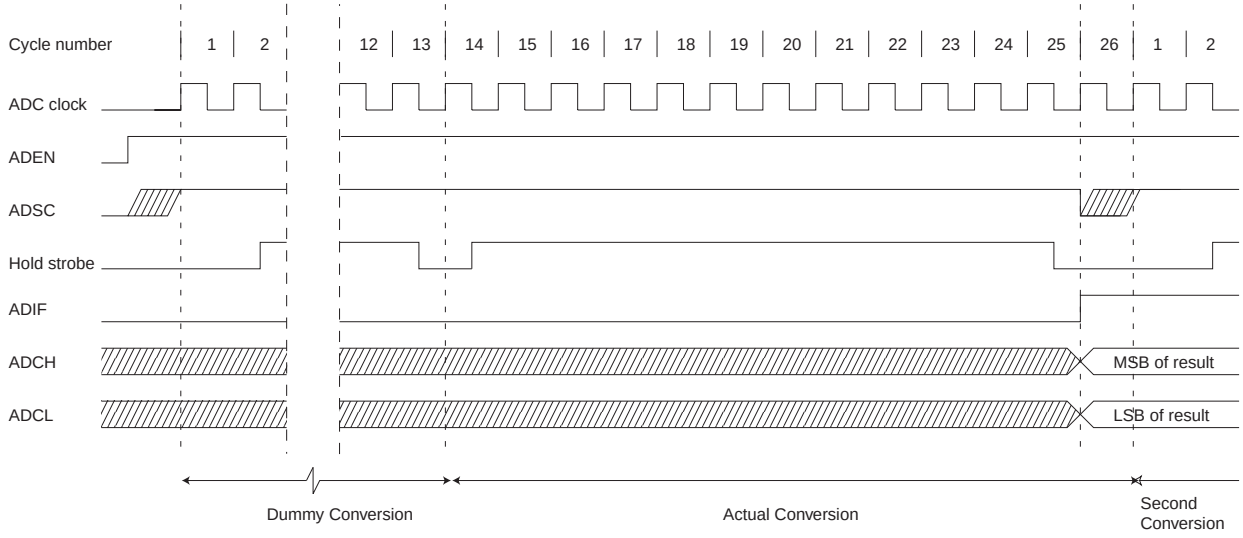


Table 21. ADC Conversion Time

Condition	Sample Cycle Number	Result Ready (cycle number)	Total Conversion Time (cycles)	Total Conversion Time (µs)
1st Conversion, Free Run	14	25	25	125 - 500
1st Conversion, Single	14	25	26	130 - 520
Free Run Conversion	2	13	13	65 - 260
Single Conversion	2	13	14	70 - 280

Figure 47. ADC Timing Diagram, Single Conversion

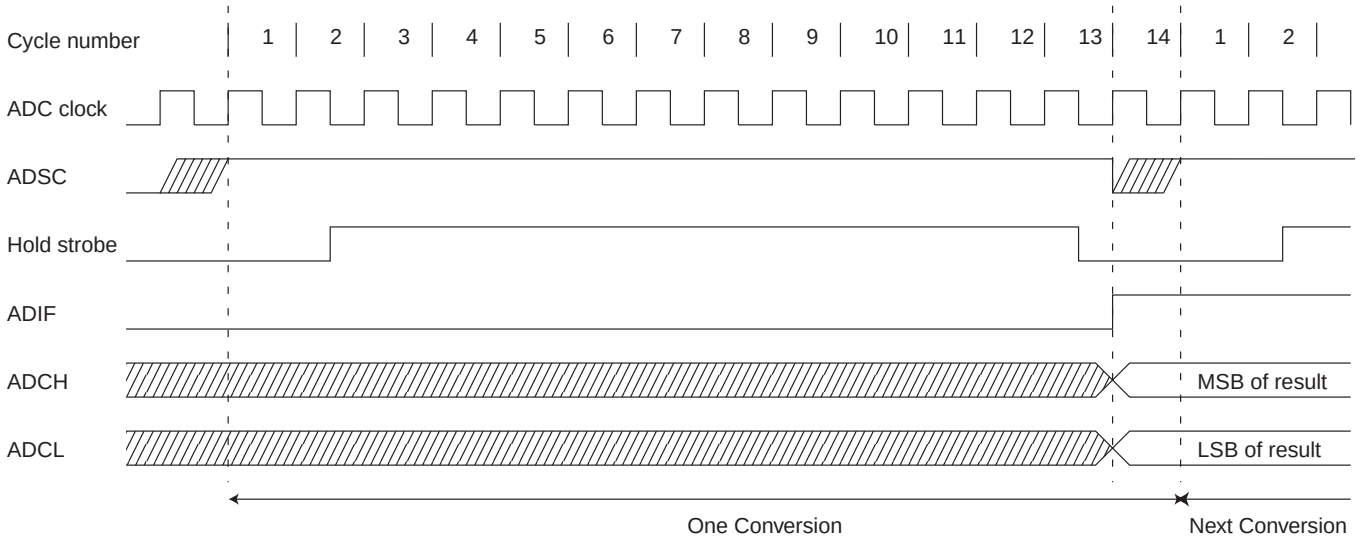
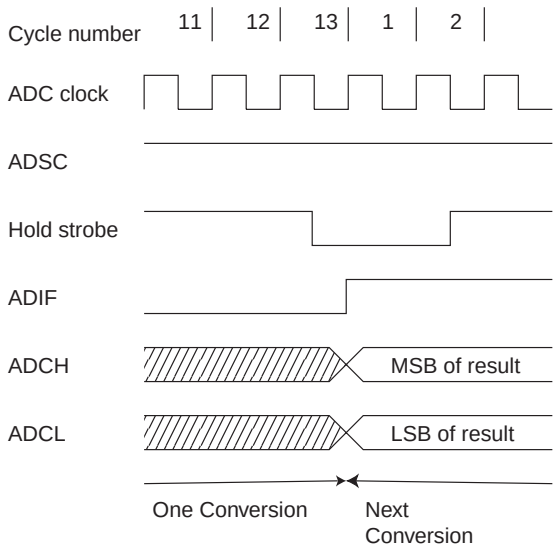


Figure 48. ADC Timing Diagram, Free Run Conversion



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. Thus:

   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.

ADC Multiplexer Select Register - ADMUX

Bit	7	6	5	4	3	2	1	0	
$07 ($27)	-	ADCBG	-	-	-	MUX2	MUX1	MUX0	ADMUX
Read/Write	R	R/W	R	R	R	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

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

These bits are reserved bits in the AT90S2333/4433, and should be written to zero if accessed.

- **Bit 6 - ADCBG: ADC Bandgap Select**

When this bit is set and the BOD is enabled (BODEN fuse is programmed), a fixed bandgap voltage of 1.22 ± 0.05V replaces the normal input to the ADC. When this bit is cleared, the normal input pin (as selected by MUX2..MUX0) is applied to the ADC.

- **Bit 5..3 - Res: Reserved Bits**

These bits are reserved bits in the AT90S2333/4433, and should be written to zero if accessed.

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

The value of these three bits selects which analog input 5-0 is connected to the ADC.

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.

ADSC remains high during the conversion. ADSC goes low after the conversion is complete, but before the result is written to the ADC Data Registers. This allows a new conversion to be initiated before the current conversion is complete. The new conversion will then start immediately after the current conversion completes. 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 Run Mode. In this mode, the ADC samples and updates the data registers continuously. Clearing this bit (zero) will terminate Free Run 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 22. 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

ADC Data Register - ADCL AND ADCH

Bit	15	14	13	12	11	10	9	8	
$05 ($25)	-	-	-	-	-	-	ADC9	ADC8	ADCH
$04 ($26)	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	0	0	0	0	0	0	0	0	
	0	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 Run 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 Run 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 AT90S2333/4433 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 AT90S2333/4433 and all analog components in the application should have a separate analog ground plane on the PCB. This ground plane is connected to the digital ground plane via a single point on the PCB.

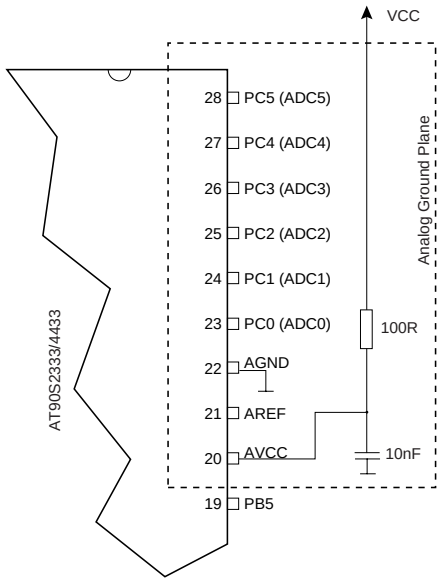
2. Keep analog signal paths as short as possible. Make sure analog tracks run over the analog ground plane, and keep them well away from high-speed switching digital tracks.

3. The AV_{CC} pin on the AT90S2333/4433 should be connected to the digital V_{CC} supply voltage via an RC network as shown in Figure 47.

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

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

Figure 49. ADC Power Connections



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

ADC Characteristics, TA = -40°C to 85°C

Symbol	Parameter	Condition	Min	Typ	Max	Units
	Resolution			10		Bits
	Absolute accuracy	VREF = 4V ADC clock = 200 kHz		1	2	LSB
	Absolute accuracy	VREF = 4V ADC clock = 1 MHz		4		LSB
	Absolute accuracy	VREF = 4V ADC clock = 2 MHz		16		LSB
	Integral Non-Linearity	V_{REF} > 2V		0.5		LSB
	Differential Non-Linearity	V_{REF} > 2V		0.5		LSB
	Zero Error (Offset)			1		LSB
	Conversion Time		65		260	µs
	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	kΩ
R_{AIN}	Analog Input Resistance			100		MΩ

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

I/O Ports

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

Port B

Port B is a 6-bit bi-directional I/O port.

Three I/O 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 23. Port B Pins Alternate Functions

Port Pin	Alternate Functions
PB0	ICP (Timer/Counter 1 input capture pin)
PB1	OC1 (Timer/Counter 1 output compare match output)
PB2	$\overline{SS}$ (SPI Slave Select input)
PB3	MOSI (SPI Bus Master Output/Slave Input)
PB4	MISO (SPI Bus Master Input/Slave Output)
PB5	SCK (SPI Bus Serial Clock)

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

Port B Data Register - PORTB

Bit	7	6	5	4	3	2	1	0	
$18 ($38)	**-**	**-**	**PORTB5**	**PORTB4**	**PORTB3**	**PORTB2**	**PORTB1**	**PORTB0**	**PORTB**
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	

Port B Data Direction Register - DDRB

Bit	7	6	5	4	3	2	1	0	
$17 ($37)	**-**	**-**	**DDB5**	**DDB4**	**DDB3**	**DDB2**	**DDB1**	**DDB0**	**DDRB**
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	

Port B Input Pins Address - PINB

Bit	7	6	5	4	3	2	1	0	
$16 ($36)	**-**	**-**	**PINB5**	**PINB4**	**PINB3**	**PINB2**	**PINB1**	**PINB0**	**PINB**
Read/Write	R	R	R	R	R	R	R	R	
Initial value	0	0	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 Port B 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 6 pins in Port B have equal functionality when used as digital I/O pins.

PBn, General I/O pin: The DDBn bit in the DDRB register selects the direction of this pin, if DDBn is set (one), PBn is configured as an output pin. If DDBn is cleared (zero), PBn is configured as an input pin. If PORTBn 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 PORTBn has to be cleared (zero) or the pin has to be configured as an output pin.The port pins are tristated when a reset condition becomes active, even if the clock is not running.

Table 24. DDBn Effects on Port B Pins

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

Note: n: 5…0, pin number.

Alternate Functions Of Port B

The alternate pin configuration is as follows:

- **SCK - Port B, Bit 5**

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 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.

- **MISO - Port B, Bit 4**

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 DDB4. When the SPI is enabled as a slave, 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.

- **MOSI - Port B, Bit 3**

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 DDB3. When the SPI is enabled as a master, the data direction of this pin is controlled by DDB3. When the pin is forced to be an input, the pull-up can still be controlled by the PORTB3 bit. See the description of the SPI port for further details.

- **$\overline{SS}$ - Port B, Bit 2**

$\overline{SS}$: Slave port select input. When the SPI is enabled as a slave, this pin is configured as an input regardless of the setting of DDB2. 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 DDB2. When the pin is forced to be an input, the pull-up can still be controlled by the PORTB2 bit. See the description of the SPI port for further details.

- **OC1 - Port B, Bit1**

OC1, Output compare match output: PB1 pin can serve as an external output for the Timer/Counter1 output compare. The pin has to be configured as an output (DDB1 set (one)) to serve this function. See the timer description on how to enable this function. The OC1 pin is also the output pin for the PWM mode timer function.

- **ICP- Port B, Bit0**

ICP, Input Capture Pin: PB0 pin can serve as an external input for the Timer/Counter1 input capture. The pin has to be configured as an input (DDB0 cleared (zero)) to serve this function. See the timer description on how to enable this function.

Figure 50. Port B Schematic Diagram (Pin PB0)

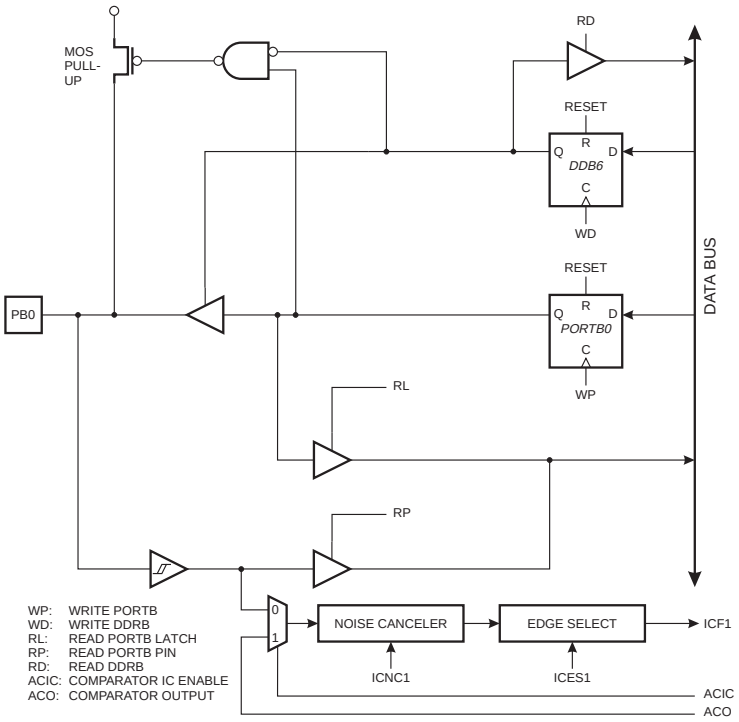


Figure 51. Port B Schematic Diagram (Pin PB1)

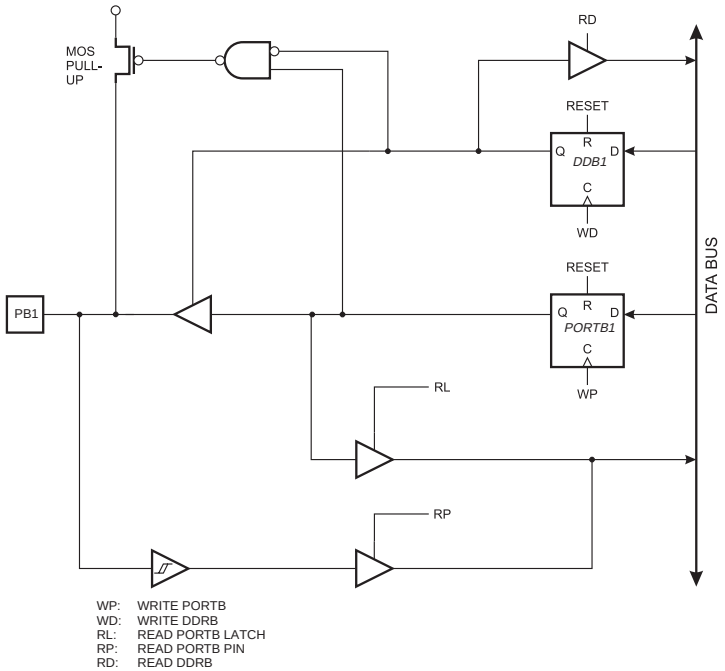


Figure 52. Port B Schematic Diagram (Pin PB2)

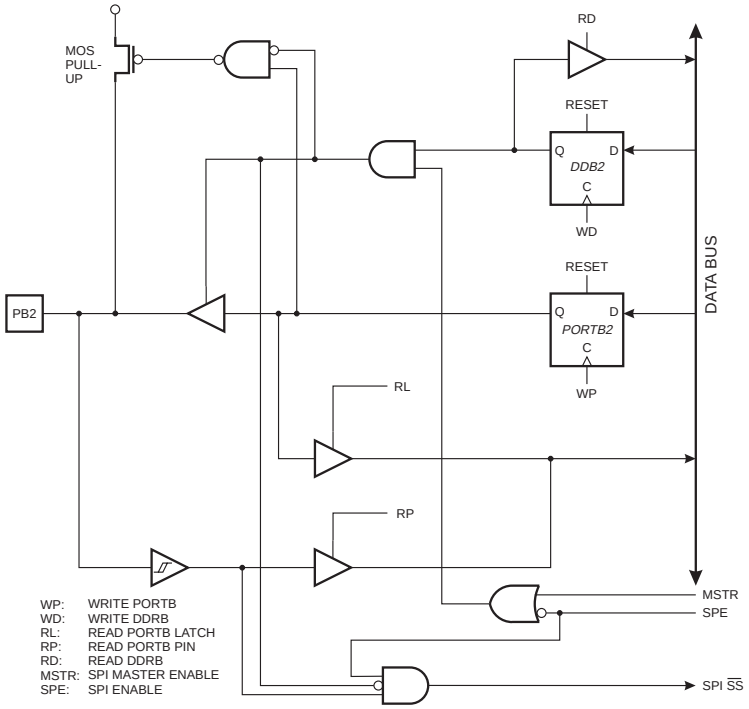


Figure 53. Port B Schematic Diagram (Pin PB3)

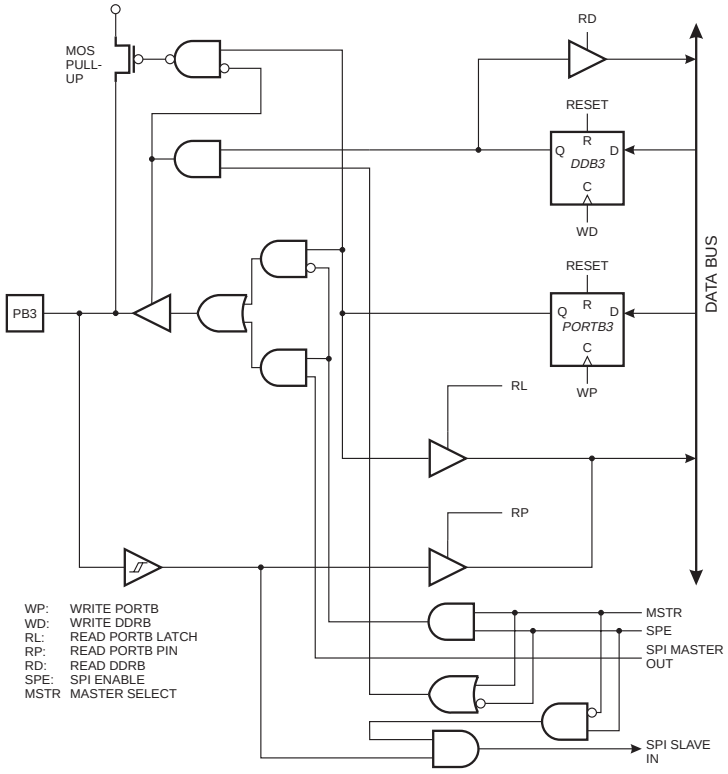


Figure 54. Port B Schematic Diagram (Pin PB4)

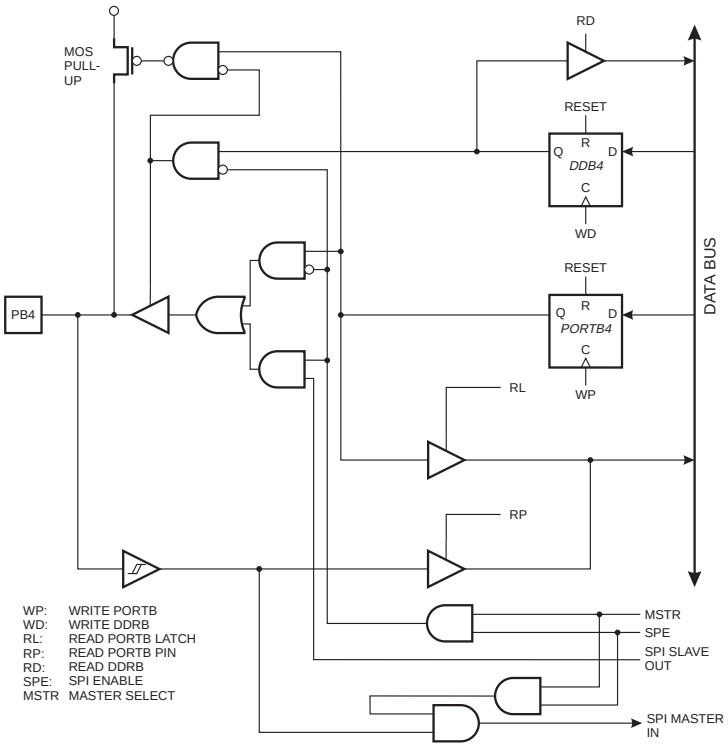
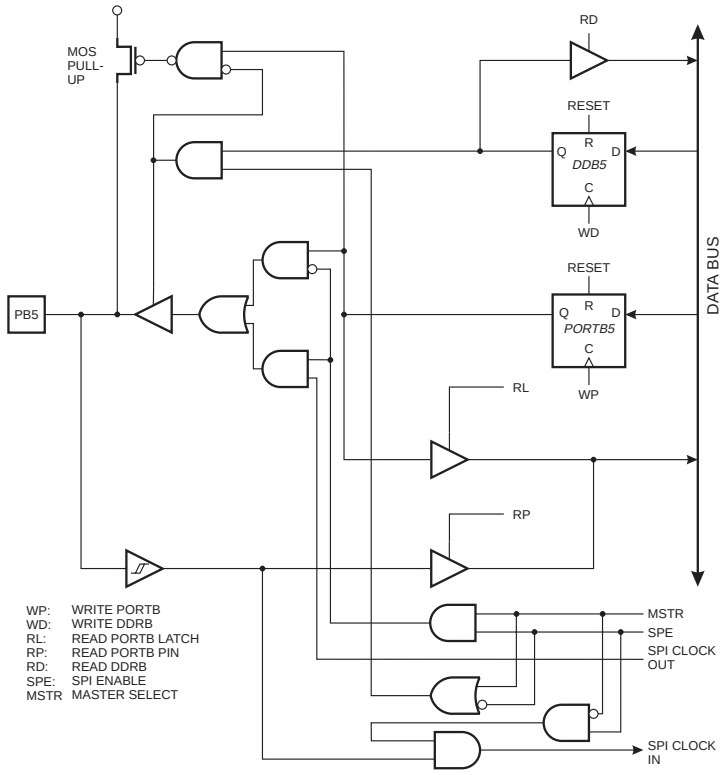


Figure 55. Port B Schematic Diagram (Pin PB5)



Port C

Port C is a 6-bit bi-directional I/O port.

Three I/O memory address locations are allocated for the Port C, one each for the Data Register - PORTC, $15($35), Data Direction Register - DDRC, $14($34) and the Port C Input Pins - PINC, $13($33). The Port C Input Pins address is read only, while the Data Register and the Data Direction Register are read/write.

All port pins have individually selectable pull-up resistors. The Port C output buffers can sink 20mA and thus drive LED displays directly. When pins PC0 to PC5 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated.

Port C has an alternate function as analog inputs for the ADC. If some Port C 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 Power Down Mode, the schmitt triggers of the digital inputs are disconnected. This allows an analog voltage close to $V_{CC}/2$ to be present during power down without causing excessive power consumption.

Port C Data Register - PORTC

Bit	7	6	5	4	3	2	1	0	
$15 ($35)	-	-	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	PORTC
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	

Port C Data Direction Register - DDRC

Bit	7	6	5	4	3	2	1	0	
$14 ($34)	-	-	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	DDRC
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	

Port C Input Pins Address - PINC

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

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

Port C As General Digital I/O

All 6 pins in Port C have equal functionality when used as digital I/O pins.

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

Table 25. DDCn Effects on Port C Pins

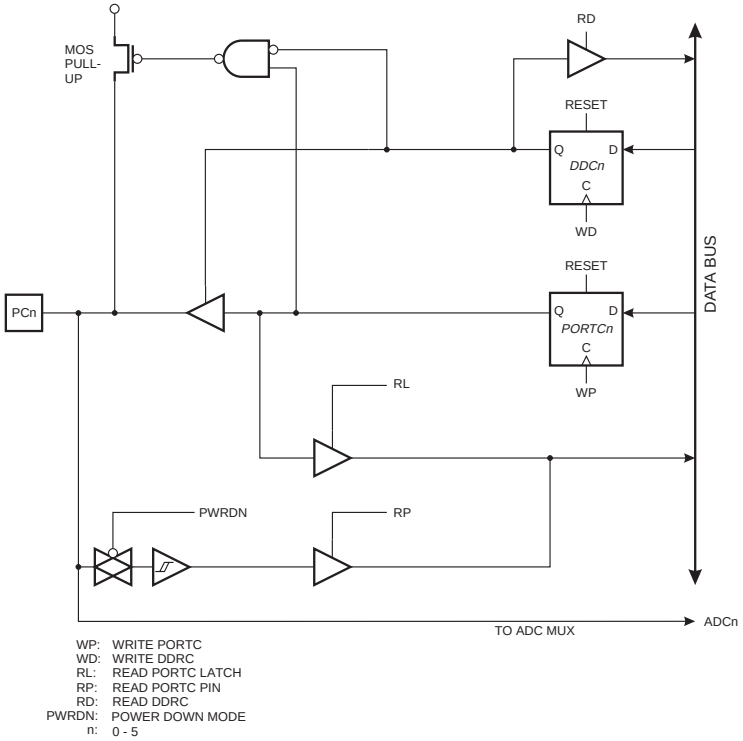
DDCn	PORTCn	I/O	Pull Up	Comment
0	0	Input	No	Tri-state (Hi-Z)
0	1	Input	Yes	PCn will source current if ext. pulled low.
1	0	Output	No	Push-Pull Zero Output
1	1	Output	No	Push-Pull One Output

Note: n: 5…0, pin number

Port C Schematics

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

Figure 56. Port C Schematic Diagrams (Pins PC0 - PC5)



Port D

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

Three I/O 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 26. Port D Pins Alternate Functions

Port Pin	Alternate Function
PD0	RXD (UART Input line)
PD1	TXD (UART Output line)
PD2	INT0 (External interrupt 0 input)
PD3	INT1 (External interrupt 1 input)
PD4	T0 (Timer/Counter 0 external counter input)
PD5	T1 (Timer/Counter 1 external counter input)
PD6	AIN0 (Analog comparator positive input)
PD7	AIN1 (Analog comparator negative input)

Port D Data Register - PORTD

Bit	7	6	5	4	3	2	1	0	
$12 ($32)	**PORTD7**	**PORTD6**	**PORTD5**	**PORTD4**	**PORTD3**	**PORTD2**	**PORTD1**	**PORTD0**	**PORTD**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Port D Data Direction Register - DDRD

Bit	7	6	5	4	3	2	1	0	
$11 ($31)	**DDD7**	**DDD6**	**DDD5**	**DDD4**	**DDD3**	**DDD2**	**DDD1**	**DDD0**	**DDRD**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Port D Input Pins Address - PIND

Bit	7	6	5	4	3	2	1	0	
$10 ($30)	**PIND7**	**PIND6**	**PIND5**	**PIND4**	**PIND3**	**PIND2**	**PIND1**	**PIND0**	**PIND**
Read/Write	R	R	R	R	R	R	R	R	
Initial value	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 Port D Data Latch is read, and when reading PIND, the logical values present on the pins are read.

Port D As General Digital I/O

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

Table 27. 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

Note: n: 7,6…0, pin number.

Alternate Functions Of Port D

- **AIN1 - Port D, Bit 7**

AIN1, Analog Comparator Negative Input. When configured as an input (DDD7 is cleared (zero)) and with the internal MOS pull up resistor switched off (PD7 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 - Port D, Bit 6**

AIN0, Analog Comparator Positive Input. When configured as an input (DDD6 is cleared (zero)) and with the internal MOS pull up resistor switched off (PD6 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 - Port D, Bit 5**

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

- **T0 - Port D, Bit 4**

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

- **INT1 - Port D, Bit 3**

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

- **INT0 - Port D, Bit 2**

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

- **TXD - Port D, Bit 1**

Transmit Data (Data output pin for the UART). When the UART transmitter is enabled, this pin is configured as an output regardless of the value of DDD1.

- **RXD - Port D, Bit 0**

Receive Data (Data input pin for the UART). When the UART receiver is enabled this pin is configured as an input regardless of the value of DDD0. When the UART forces this pin to be an input, a logical one in PORTD0 will turn on the internal pull-up.

Port D Schematics

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

Figure 57. Port D Schematic Diagram (Pin PD0)

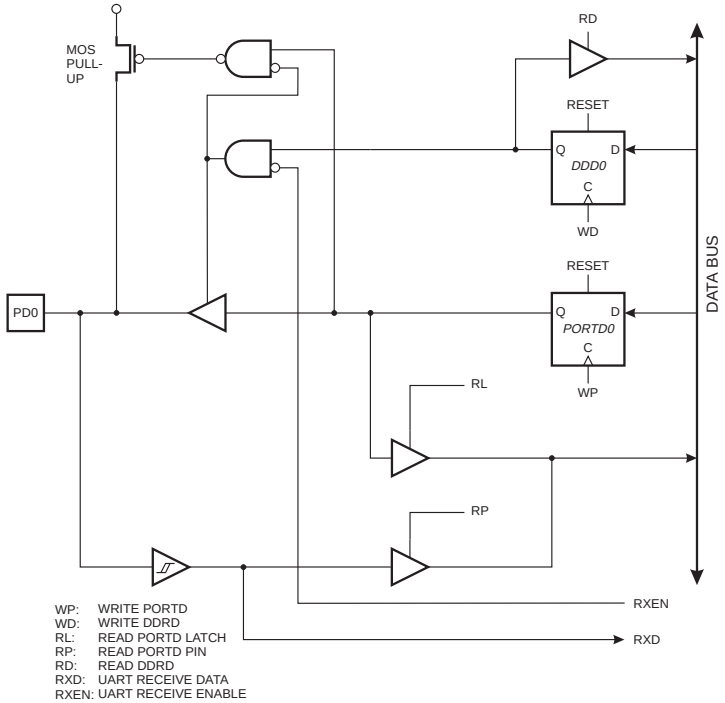


Figure 58. Port D Schematic Diagram (Pin PD1)

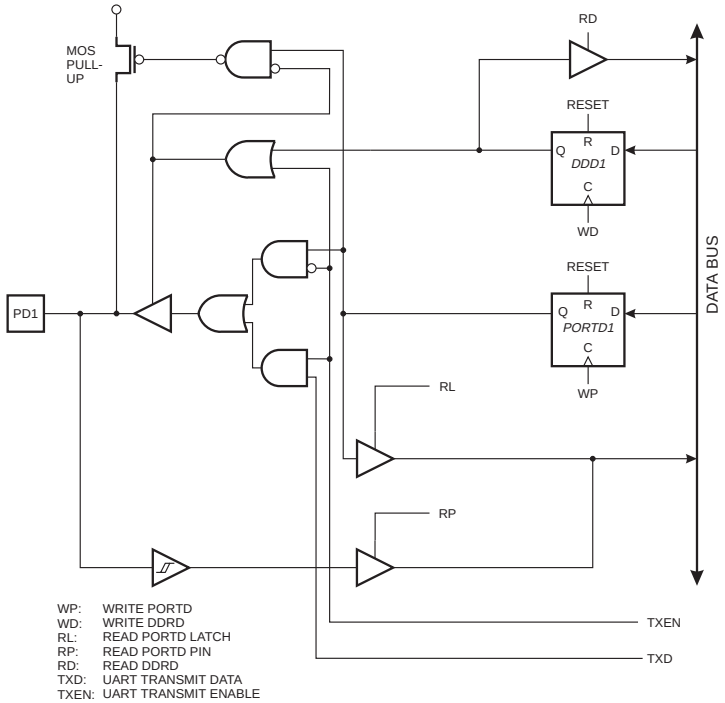


Figure 59. Port D Schematic Diagram (Pins PD2 and PD3)

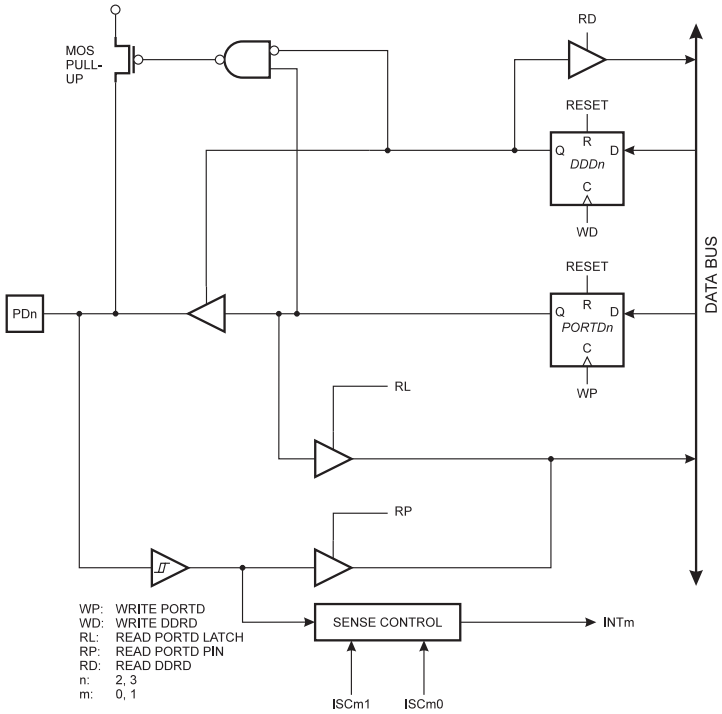


Figure 60. Port D Schematic Diagram (Pins PD4 and PD5)

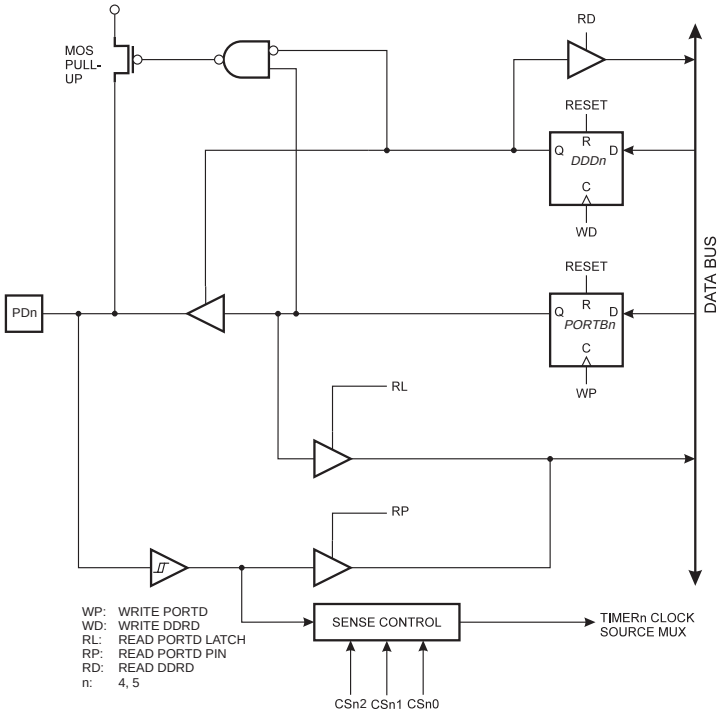
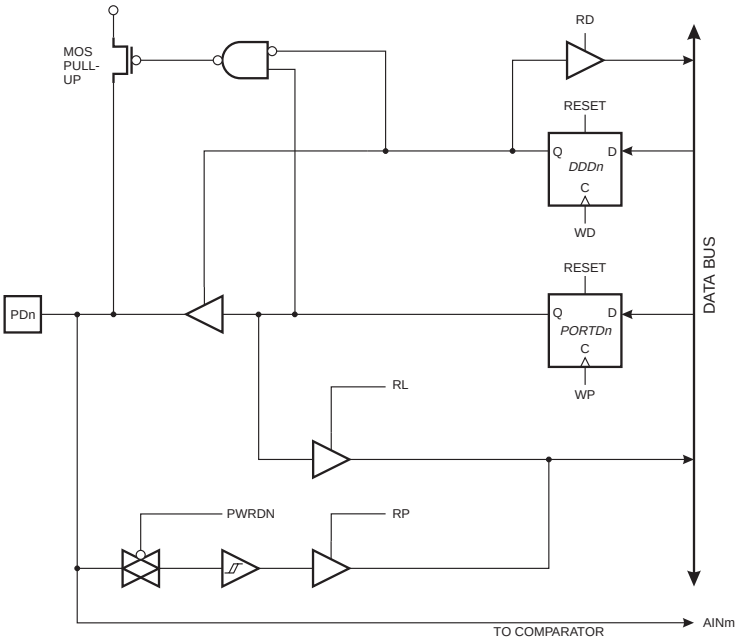


Figure 61. Port D Schematic Diagram (Pins PD6 and PD7)



WP: WRITE PORTD
WD: WRITE DDRD
RL: READ PORTD LATCH
RP: READ PORTD PIN
RD: READ DDRD
PWRDN: POWER DOWN MODE
n: 6, 7
m: 0, 1

Memory Programming

Program and Data Memory Lock Bits

The AT90S2333/4433 MCU provides two Lock bits which can be left unprogrammed (‘1’) or can be programmed (‘0’) to obtain the additional features listed in Table 28. The Lock bits can only be erased with the Chip Erase command.

Table 28. Lock Bit Protection Modes

Memory Lock Bits			Protection Type
Mode	**LB1**	**LB2**	
1	1	1	No memory lock features enabled.
2	0	1	Further programming of the Flash and EEPROM is disabled.(1)
3	0	0	Same as mode 2, and verify is also disabled.

Note: 1. In Parallel mode, programming of the Fuse bits are also disabled. Program the Fuse bits before programming the Lock bits.

Fuse Bits

The AT90S2333/4433 has six Fuse bits, SPIEN, BODLEVEL, BODEN and CKSEL2..0.

- When the SPIEN Fuse is programmed (‘0’), Serial Program and Data Downloading is enabled. Default value is programmed (‘0’). This bit is not accessible in serial programming mode.
- The BODLEVEL Fuse selects the Brown-Out Detection Level and changes the Start-up times. See “Brown-Out Detection” on page 21. Default value is unprogrammed (‘1’).
- When the BODEN Fuse is programmed (‘0’), the Brown- Out Detector is enabled. See “Brown-Out Detection” on page 21. Default value is unprogrammed (‘1’).
- CKSEL2..0: See Table 5, “Reset Delay Selections”, for which combination of CKSEL2..0 to use. Default value is ‘010’.

Signature Bytes

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

For the AT90S4433(1) they are:

1. $000: $1E (indicates manufactured by Atmel)
2. $001: $92 (indicates 4KB Flash memory)
3. $002: $03 (indicates AT90S4433 device when signature byte $001 is $92)

For AT90S2333(1) they are:

1. $000: $1E (indicates manufactured by Atmel)
2. $001: $91 (indicates 2KB Flash memory)
3. $002: $05 (indicates AT90S2333 device when signature byte $001 is $91)

Note: 1. When both Lock bits are programmed (Lock mode 3), the signature bytes can not be read in serial mode. Reading the signature bytes will return: $00, $01 and $02.

Programming the Flash and EEPROM

Atmel’s AT90S2333/4433 offers 2K/4K bytes of in-system reprogrammable Flash Program memory and 128/256 bytes of EEPROM Data memory.

The AT90S2333/4433 is 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 program and data into the AT90S2333/4433 inside the user’s system.

The Program and Data memory arrays on the AT90S2333/4433 are programmed byte-by-byte in either programming modes. For the EEPROM, an auto-erase cycle is provided within the self-timed write instruction in the serial programming mode. During programming, the supply voltage must be in accordance with Table 29.

Table 29. Supply voltage during programming

Part	Serial programming	Parallel programming
AT90LS2333	2.7 - 6.0 V	4.5 - 5.5 V
AT90S2333	4.0 - 6.0 V	4.5 - 5.5 V
AT90LS4433	2.7 - 6.0 V	4.5 - 5.5 V
AT90S4433	4.0 - 6.0 V	4.5 - 5.5 V

Parallel Programming

This section describes how to parallel program and verify Flash Program memory, EEPROM Data memory, Lock bits and Fuse bits in the AT90S2333/4433.

Signal Names

In this section, some pins of the AT90S2333/4433 are referenced by signal names describing their function during parallel programming. See Figure 62 and Table 30. Pins not described in Table 30 are referenced by pin names.

The XA1/XA0 pins determine the action executed when the XTAL1 pin is given a positive pulse. The bit coding are shown in Table 31.

When pulsing $\overline{WR}$ or $\overline{OE}$, the command loaded determines the action executed. The Command is a byte where the different bits are assigned functions as shown in Table 32.

Figure 62. Parallel Programming

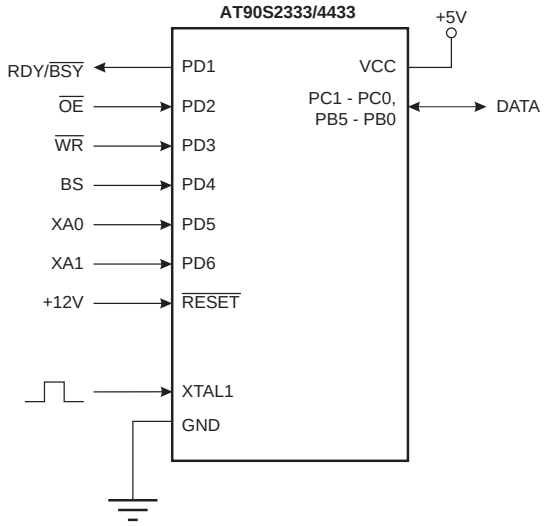


Table 30. Pin Name Mapping

Signal Name in Programming Mode	Pin Name	I/O	Function
RDY/$\overline{\text{BSY}}$	PD1	O	0: Device is busy programming, 1: Device is ready for new command
$\overline{\text{OE}}$	PD2	I	Output Enable (Active low)
$\overline{\text{WR}}$	PD3	I	Write Pulse (Active low)
BS	PD4	I	Byte Select (‘0’ selects low byte, ‘1’ selects high byte)
XA0	PD5	I	XTAL Action Bit 0
XA1	PD6	I	XTAL Action Bit 1
DATA	PC1-0, PB5-0	I/O	Bidirectional Databus (Output when $\overline{\text{OE}}$ is low)

Table 31. XA1 and XA0 Coding

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

Table 32. Command Byte Bit Coding

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

Enter Programming Mode

The following algorithm puts the device in parallel programming mode:

1. Apply supply voltage according to Table 29, between V_{CC} and GND.
2. Set the $\overline{\text{RESET}}$ and BS pin to ‘0’ and wait at least 100 ns.
3. Apply 11.5 - 12.5V to $\overline{\text{RESET}}$. Any activity on BS within 100 ns after +12V has been applied to $\overline{\text{RESET}}$, will cause the device to fail entering programming mode.

Chip Erase

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

Load Command “Chip Erase”

1. Set XA1, XA0 to ‘10’. This enables command loading.
2. Set BS to ‘0’.
3. Set DATA to ‘1000 0000’. This is the command for Chip erase.
4. Give XTAL1 a positive pulse. This loads the command.
5. Give $\overline{WR}$ a t_{WLWH_CE} wide negative pulse to execute Chip Erase. See Table 33 for t_{WLWH_CE} value. Chip Erase does not generate any activity on the RDY/$\overline{BSY}$ pin.

Programming the Flash

A: Load Command “Write Flash”

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

B: Load Address High Byte

1. Set XA1, XA0 to ‘00’. This enables address loading.
2. Set BS to ‘1’. This selects high byte.
3. Set DATA = Address high byte ($00 - $03/$07)
4. Give XTAL1 a positive pulse. This loads the address high byte.

C: Load Address Low Byte

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

D: Load Data Low Byte

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

E: Write Data Low Byte

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

(See Figure 63 for signal waveforms.)

F: Load Data High Byte

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

G: Write Data High Byte

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

(See Figure 64 for signal waveforms.)

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

- The command needs only be loaded once when writing or reading multiple memory locations.

- Address high byte needs only be loaded before programming a new 256 word page in the Flash.
- Skip writing the data value $FF, that is the contents of the entire Flash and EEPROM after a Chip Erase.

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

Figure 63. Programming the Flash Waveforms

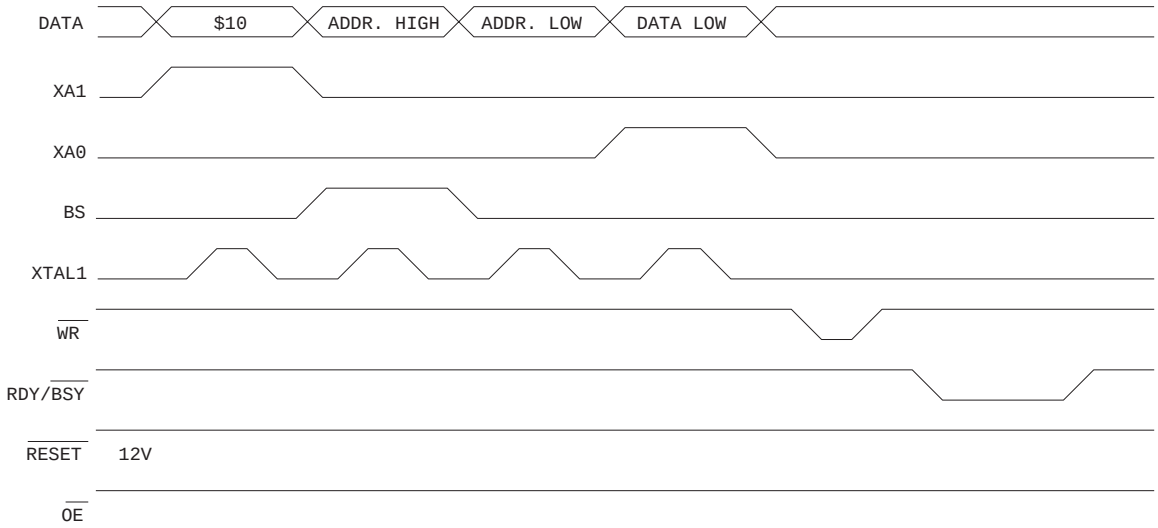
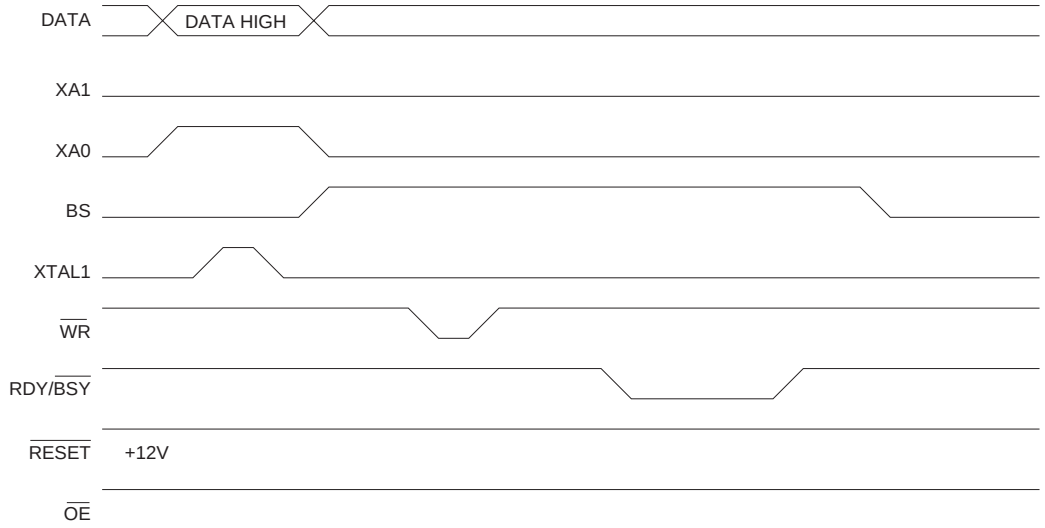


Figure 64. Programming the Flash Waveforms (continued)



Reading the Flash

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

1. A: Load Command ‘0000 0010’.
2. B: Load Address High Byte ($00 - $03/$07).
3. C: Load Address Low Byte ($00 - $FF).
4. Set $\overline{OE}$ to ‘0’, and BS to ‘0’. The Flash word low byte can now be read at DATA.
5. Set BS to ‘1’. The Flash word high byte can now be read from DATA.
6. Set $\overline{OE}$ to ‘1’.

Programming the EEPROM

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

1. A: Load Command ‘0001 0001’.
2. C: Load Address Low Byte ($00 - $7F/$FF).
3. D: Load Data Low Byte ($00 - $FF).
4. E: Write Data Low Byte.

Reading the EEPROM

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

1. A: Load Command ‘0000 0011’.
2. C: Load Address Low Byte ($00 - $7F/$FF).
3. Set $\overline{OE}$ to ‘0’, and BS to ‘0’. The EEPROM data byte can now be read at DATA.
4. Set $\overline{OE}$ to ‘1’.

Programming the Fuse Bits

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

1. A: Load Command ‘0100 0000’.
2. D: Load Data Low Byte. Bit n = ‘0’ programs and bit n = ‘1’ erases the Fuse bit.

   Bit 5 = SPIEN Fuse bit
   Bit 4 = BODLEVEL Fuse bit
   Bit 3 = BODEN Fuse bit
   Bit 2 = CKSEL2 Fuse bit
   Bit 1 = CKSEL1 Fuse bit
   Bit 0 = CKSEL0 Fuse bit

   Bit 7-6 = ‘1’. These bits are reserved and should be left unprogrammed (‘1’).
3. Give $\overline{WR}$ a t_{WLWH_PFB} wide negative pulse to execute the programming, t_{WLWH_PFB} is found in Table 33. Programming the Fuse bits does not generate any activity on the RDY/$\overline{BSY}$ pin.

Programming the Lock Bits

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

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

   Bit 2 = Lock Bit2
   Bit 1 = Lock Bit1

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

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

Reading the Fuse and Lock Bits

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

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

Bit 5 = SPIEN Fuse bit
Bit 4 = BODLEVEL Fuse bit
Bit 3 = BODEN Fuse bit
Bit 2 = CKSEL2 Fuse bit
Bit 1 = CKSEL1 Fuse bit
Bit 0 = CKSEL0 Fuse bit

3. Set BS to ‘1’. The status of the Lock bits can now be read at DATA (‘0’ means programmed).
   Bit 2 = Lock Bit2
   Bit 1= Lock Bit1
4. Set $\overline{OE}$ to ‘1’.

Reading the Signature Bytes

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

1. A: Load Command ‘0000 1000’.
2. C: Load Address Low Byte ($00 - $02).
   Set $\overline{OE}$ to ‘0’, and BS to ‘0’. The selected Signature byte can now be read at DATA.
3. Set $\overline{OE}$ to ‘1’.

Parallel Programming Characteristics

Figure 65. Parallel Programming Timing

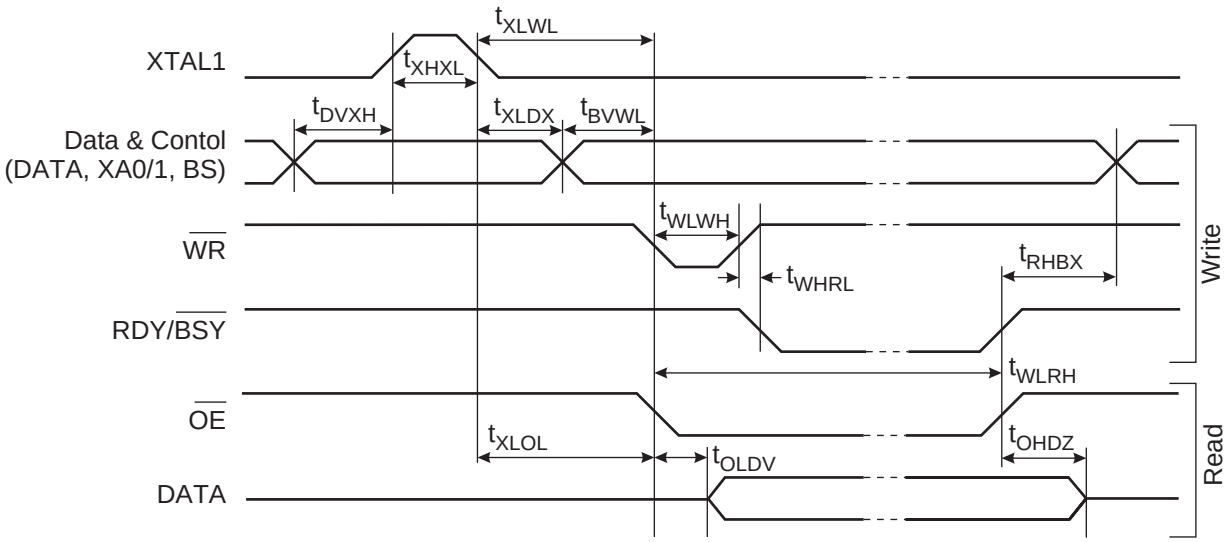


Table 33. Parallel Programming Characteristics T_A = 25°C ± 10%, V_{CC} =5V ± 10%

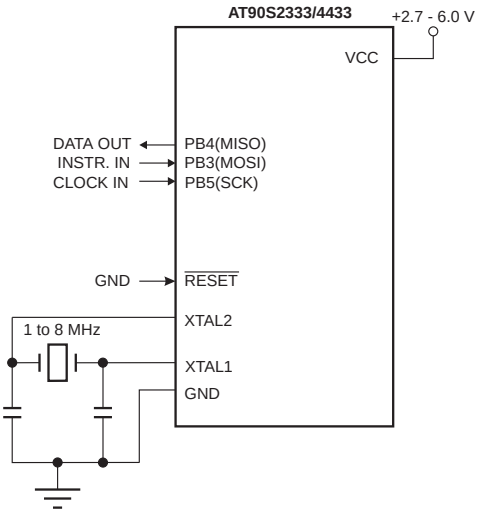
Symbol	Parameter	Min	Typ	Max	Units
V_{PP}	Programming Enable Voltage	11.5		12.5	V
I_{PP}	Programming Enable Current			250	µA
t_{DVXH}	Data and Control Setup before XTAL1 High	67			ns
t_{XHXL}	XTAL1 Pulse Width High	67			ns
t_{XLDX}	Data and Control Hold after XTAL1 Low	67			ns
t_{XLWL}	XTAL1 Low to $\overline{WR}$ Low	67			ns
t_{BVWL}	BS Valid to $\overline{WR}$ Low	67			ns
t_{RHBX}	BS Hold after RDY/$\overline{BSY}$ High	67			ns
t_{WLWH}	$\overline{WR}$ Pulse Width Low(1)	67			ns
t_{WHRL}	$\overline{WR}$ High to RDY/$\overline{BSY}$ Low(2)		20		ns
t_{WLRH}	$\overline{WR}$ Low to RDY/$\overline{BSY}$ High(2)	0.5	0.7	0.9	ms
t_{XLOL}	XTAL1 Low to $\overline{OE}$ Low	67			ns
t_{OLDV}	$\overline{OE}$ Low to DATA Valid		20		ns
t_{OHDZ}	$\overline{OE}$ High to DATA Tristated			20	ns
t_{WLWH_CE}	$\overline{WR}$ Pulse Width Low for Chip Erase	5	10	15	ms
t_{WLWH_PFB}	$\overline{WR}$ Pulse Width Low for Programming the Fuse Bits	1.0	1.5	1.8	ms

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

Serial Downloading

Both the Program and Data memory arrays can be programmed using the SPI bus while $\overline{RESET}$ is pulled to GND. The serial interface consists of pins SCK, MOSI (input) and MISO (output), see Figure 66. After $\overline{RESET}$ is set low, the Programming Enable instruction needs to be executed first before program/erase instructions can be executed.

Figure 66. Serial Programming and Verify



For the EEPROM, an auto-erase cycle is provided within the self-timed write instruction and there is no need to first execute the Chip Erase instruction. The Chip Erase instruction turns the content of every memory location in both the Program and EEPROM arrays into $FF.

The Program and EEPROM memory arrays have separate address spaces:

0000 to $03FF/$07FF (AT90S2333/AT90S4433) for Program memory and $0000 to $007F/$00FF (AT90S2333/AT90S4433) 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

Serial Programming Algorithm

When writing serial data to the AT90S2333/AT90S4433, data is clocked on the rising edge of CLK.

When reading data from the AT90S2333/AT90S4433, data is clocked on the falling edge of CLK. See Figure 67, Figure 68 and Table 36 for details.

To program and verify the AT90S2333/AT90S4433 in the serial programming mode, the following sequence is recommended (See four byte instruction formats in Table 35):

1. Power-up sequence:

   Apply power between V_{CC} 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/PB3.
3. The serial programming instructions will not work if the communication is out of syncronization. When in sync, the second byte ($53) will echo back when issuing the third byte of the Programming Enable instruction. Wheter the echo is correct or not, all 4 bytes of the instruction must be transmitted. If the $53 did not echo back, give SCK a positive pulse and issue a new Programming Enable instruction. 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 t_{WD_ERASE} after the instruction, give $\overline{\text{RESET}}$ a positive pulse, and start over from Step 2. See Table 37 on page 82 for t_{WD_ERASE} value.
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. If polling is not used, wait t_{WD_PROG} before transmitting the next instruction. In an erased device, no $FFs in the data file(s) needs to be programmed. See Table 38 on page 82 for t_{WD_PROG} value.
6. Any memory location can be verified by using the Read instruction which returns the content at the selected address at serial output MISO/PB4.
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

Data Polling EEPROM

When a byte is being programmed into the EEPROM, reading the address location being programmed will give the value P1 until the auto-erase is finished, and then the value P2. See Table 34 for P1 and P2 values.

At the time the device is ready for a new EEPROM 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 values P1 and P2, so when programming these values, the user will have to wait for at least the prescribed time t_{WD_PROG} before programming the next byte. See Table 38 for t_{WD_PROG} value. 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 reprogrammed without first chip-erasing the device.

Table 34. Read Back Value during EEPROM polling

Part	P1	P2
AT90S/LS2333	$00	$FF
AT90S/LS4433	$00	$FF

Data Polling Flash

When a byte is being programmed into the Flash, 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, so when programming this value, the user will have to wait for at least t_{WD_PROG} before programming the next byte. As a chip-erased device contains $FF in all locations, programming of addresses that are meant to contain $FF, can be skipped.

Figure 67. Serial Programming Waveforms

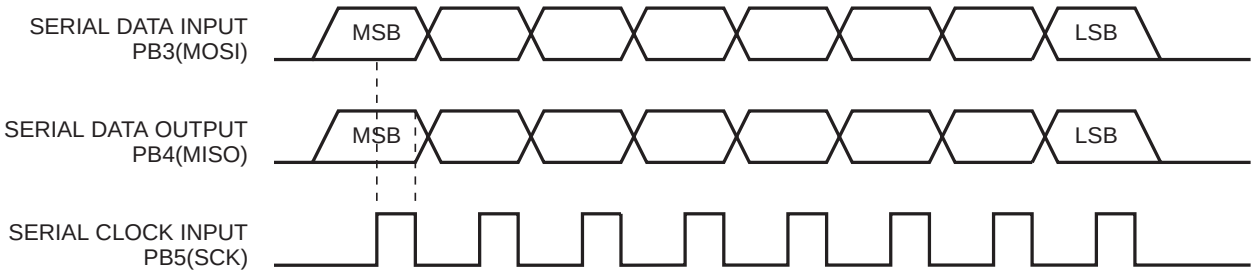


Table 35. Serial Programming Instruction Set

Instruction	Instruction Format				Operation
	Byte 1	Byte 2	Byte 3	Byte4	
Programming Enable	1010 1100	0101 0011	xxxx xxxx	xxxx xxxx	Enable Serial Programming while $\overline{\text{RESET}}$ is low.
Chip Erase	1010 1100	100x xxxx	xxxx xxxx	xxxx xxxx	Chip Erase Flash and EEPROM memory arrays.
Read Program Memory	0010 **H**000	xxxx x**aaa**	**bbbb bbbb**	**oooo oooo**	Read **H** (high or low) data **o** from Program memory at word address **a**:**b**.
Write Program Memory	0100 **H**000	xxxx x**aaa**	**bbbb bbbb**	**iiii iiii**	Write **H** (high or low) data **i** to Program memory at word address **a**:**b**.
Read EEPROM Memory	1010 0000	xxxx xxxx	**bbbb bbbb**	**oooo oooo**	Read data **o** from EEPROM memory at address **a**:**b**.
Write EEPROM Memory	1100 0000	xxxx xxxx	**bbbb bbbb**	**iiii iiii**	Write data **i** to EEPROM memory at address **a**:**b**.
Write Lock Bits	1010 1100	1111 1***21***1	xxxx xxxx	xxxx xxxx	Write Lock bits. Set bits ***1,2***='0' to program Lock bits.
Read Lock Bits	0101 1000	xxxx xxxx	xxxx xxxx	xxxx x***21***x	Rad Lock bits. '0' = programmed, '1' = unprogrammed.
Read Sigature Bytes	0011 0000	xxxx xxxx	xxxx xx**bb**	**oooo oooo**	Read Signature Byte **o** at address **b**.(1)
Write Fuse Bits	1010 1100	101**7 6543**	xxxx xxxx	xxxx xxxx	Set bits **7** - **3** = '0' to program, '1' to unprogram.
Read Fuse Bits	0101 0000	xxxx xxxx	xxxx xxxx	xx**87 6543**	Read fuse bits. '0' = programmed, '1' = unprogrammed.

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
3 = CKSEL0 Fuse
4 = CKSEL1 Fuse
5 = CKSEL2 Fuse
6 = BODEN Fuse
7 = BODLEVEL Fuse
8 = SPIEN Fuse

Note: 1. The signature bytes are not readable in Lock mode 3, i.e. both Lock bits programmed.

Serial Programming Characteristics

Figure 68. Serial Programming Timing

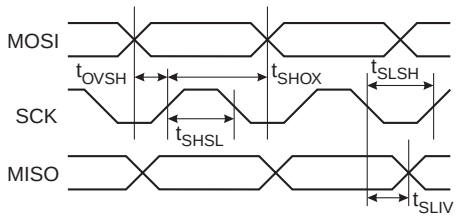


Table 36. Serial Programming Characteristics

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

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

Table 37. Minimum wait delay after the Chip Erase instruction

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

Table 38. Minimum wait delay after writing a Flash or EEPROM location

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

Electrical Characteristics

Absolute Maximum Ratings*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on any Pin except $\overline{\text{RESET}}$ with respect to Ground	-1.0V to V_{CC}+0.5V
Voltage on $\overline{\text{RESET}}$ with respect to Ground	-1.0V to +13.0V
Maximum Operating Voltage	6.6V
DC Current per I/O Pin	40.0 mA
DC Current V_{CC} and GND Pins	300.0 mA

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

DC Characteristics

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

Symbol	Parameter	Condition	Min	Typ	Max	Units
V_{IL}	Input Low Voltage	Except (XTAL, $\overline{\text{RESET}}$)	-0.5		$0.3V_{CC}$(1)	V
V_{IL1}	Input Low Voltage	XTAL	-0.5		0.1(1)	V
V_{IL1}	Input Low Voltage	$\overline{\text{RESET}}$	-0.5		$0.2V_{CC}$(1)	V
V_{IH}	Input High Voltage	Except (XTAL, $\overline{\text{RESET}}$)	$0.7\ V_{CC}$(2)		V_{CC} + 0.5	V
V_{IH1}	Input High Voltage	XTAL	$0.7\ V_{CC}$(2)		V_{CC} + 0.5	V
V_{IH2}	Input High Voltage	$\overline{\text{RESET}}$	$0.85\ V_{CC}$(2)		V_{CC}+0.5	V
V_{OL}	Output Low Voltage(3) (Ports B, C, D)	I_{OL} = 20 mA, V_{CC} = 5V I_{OL} = 10 mA, V_{CC} = 3V			0.6 0.5	V V
V_{OH}	Output High Voltage(4) (Ports B, C, D)	I_{OH} = -3 mA, V_{CC} = 5V I_{OH} = -1.5 mA, V_{CC} = 3V	4.3 2.2			V V
I_{IL}	Input Leakage Current I/O pin	V_{CC} = 6V, pin = low (Absolute value)			8.0	ua
I_{IH}	Input Leakage Current I/O pin	V_{CC} = 6V, pin = high (Absolute value)			8.0	ua
RRST	Reset Pull-Up		100		500	kΩ
$R_{I/O}$	I/O Pin Pull-Up Resistor		35		120	kΩ
I_{CC}	Power Supply Current	Active 4MHz, V_{CC} = 3V			5.0	mA
		Idle 4MHz, V_{CC} = 3V			2.0	mA
		Power Down, V_{CC} = 3V WDT enabled(5)			20.0	µA
		Power Down, V_{CC} = 3V WDT disbled(5)			10	µA

DC Characteristics

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

Symbol	Parameter	Condition	Min	Typ	Max	Units
V_{ACIO}	Analog Comparator Input Offset Voltage	V_{CC} = 5V			40	mV
I_{ACLK}	Analog Comparator Input Leakage A	V_{CC} = 5V V_{in} = V_{CC}/2	-50		50	nA
t_{ACPD}	Analog Comparator Propagation Delay	V_{CC} = 2.7V V_{CC} = 4.0V		750 500		ns

Notes:
1. “Max” means the highest value where the pin is guaranteed to be read as low (logical zero).
2. “Min” means the lowest value where the pin is guaranteed to be read as high (logical one).
3. Although each I/O port can sink more than the test conditions (20mA at Vcc = 5V, 10mA at Vcc = 3V) under steady state conditions (non-transient), the following must be observed:
   1] The sum of all IOL, for all ports, should not exceed 300 mA.
   2] The sum of all IOL, for port C0-C5 , should not exceed 100 mA.
   3] The sum of all IOL, for ports B0-B5, D0-D7 and XTAL2, should not exceed 200 mA.
   If IOL exceeds the test condition, VOL may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test condition.
4. Although each I/O port can source more than the test conditions (3mA at Vcc = 5V, 1.5mA at Vcc = 3V) under steady state conditions (non-transient), the following must be observed:
   1] The sum of all IOL, for all ports, should not exceed 300 mA.
   2] The sum of all IOL, for port C0-C5 , should not exceed 100 mA.
   3] The sum of all IOL, for ports B0-B5, D0-D7 and XTAL2, should not exceed 200 mA.
   If IOH exceeds the test condition, VOH may exceed the related specification. Pins are not guaranteed to source current greater than the listed test condition.
5. Minimum V_{CC} for Power Down is 2V.

External Clock Drive Waveforms

Figure 69. External Clock

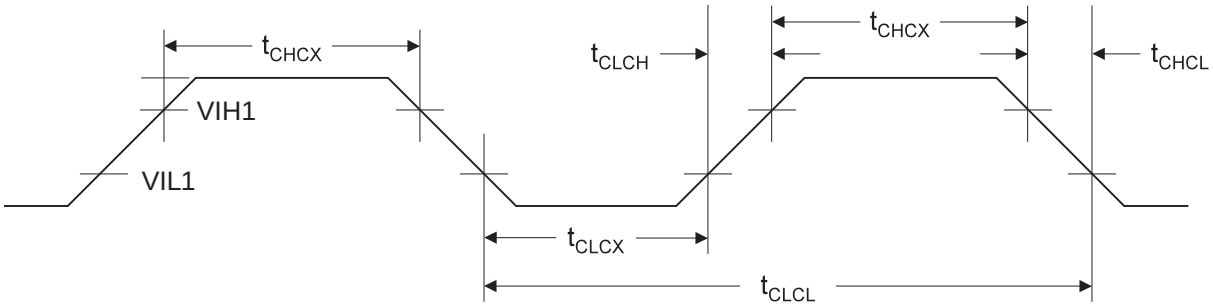


Table 39. External Clock Drive

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

Typical Characteristics

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

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

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

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

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

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

The difference between Power Down mode with Brown Out Detector enabled and Power Down mode with Watchdog timer disabled represents the differential current drawn by the brown out detector.

Figure 70. Active Supply Current vs. Frequency

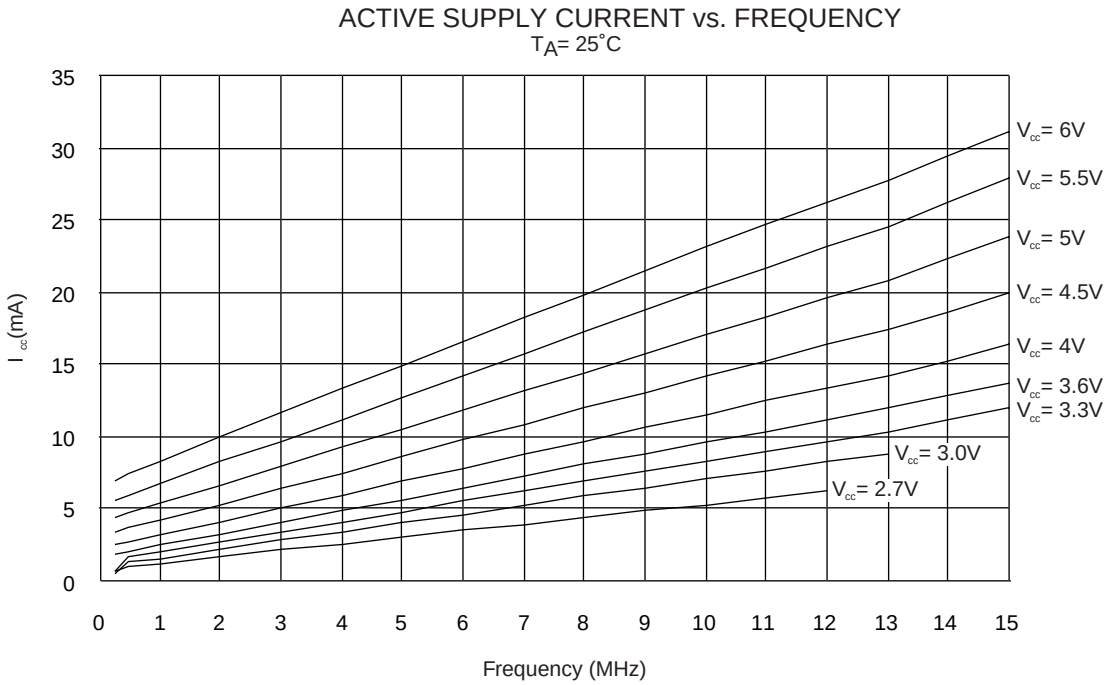


Figure 71. Active Supply Current vs. V_{CC}

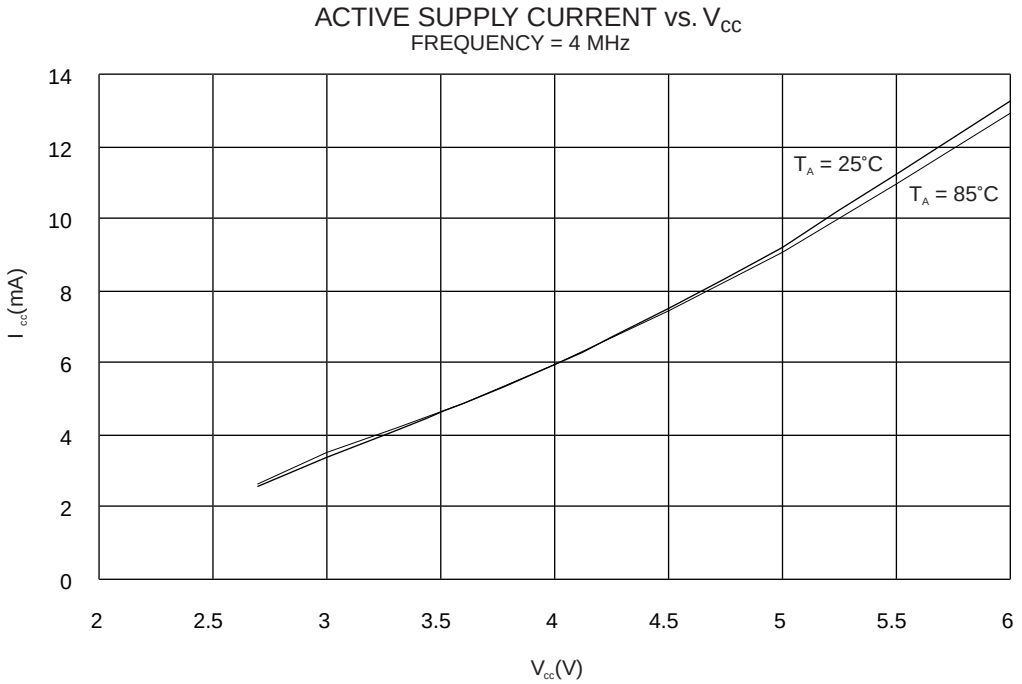


Figure 72. Idle Supply Current vs. Frequency

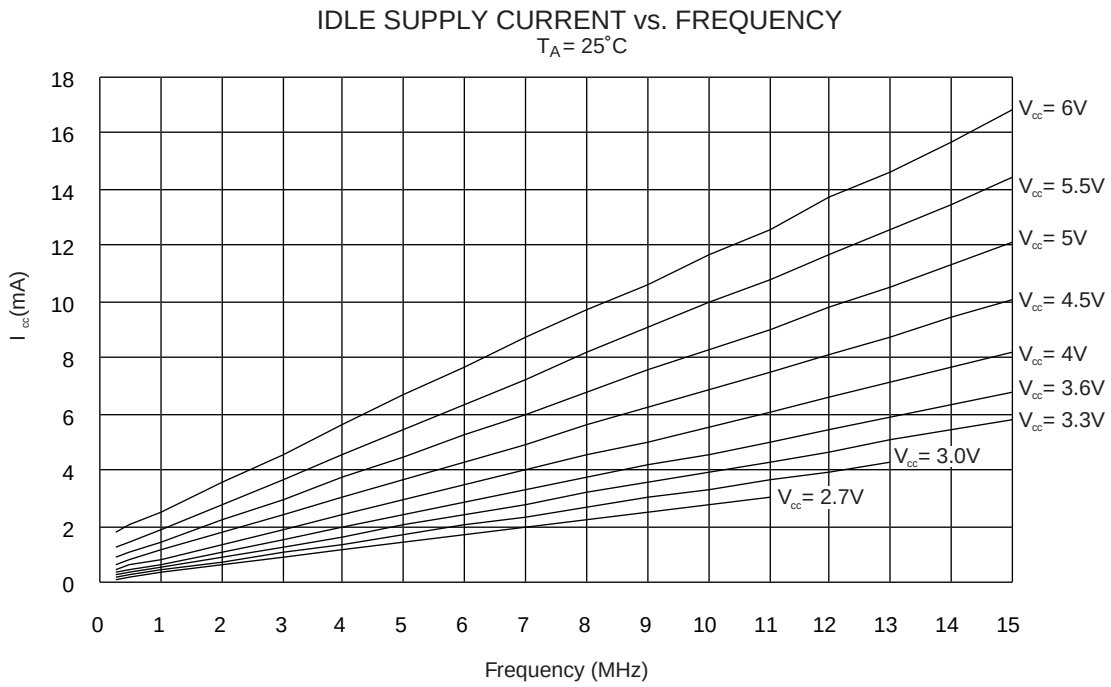


Figure 73. Idle Supply Current vs. V_{CC}

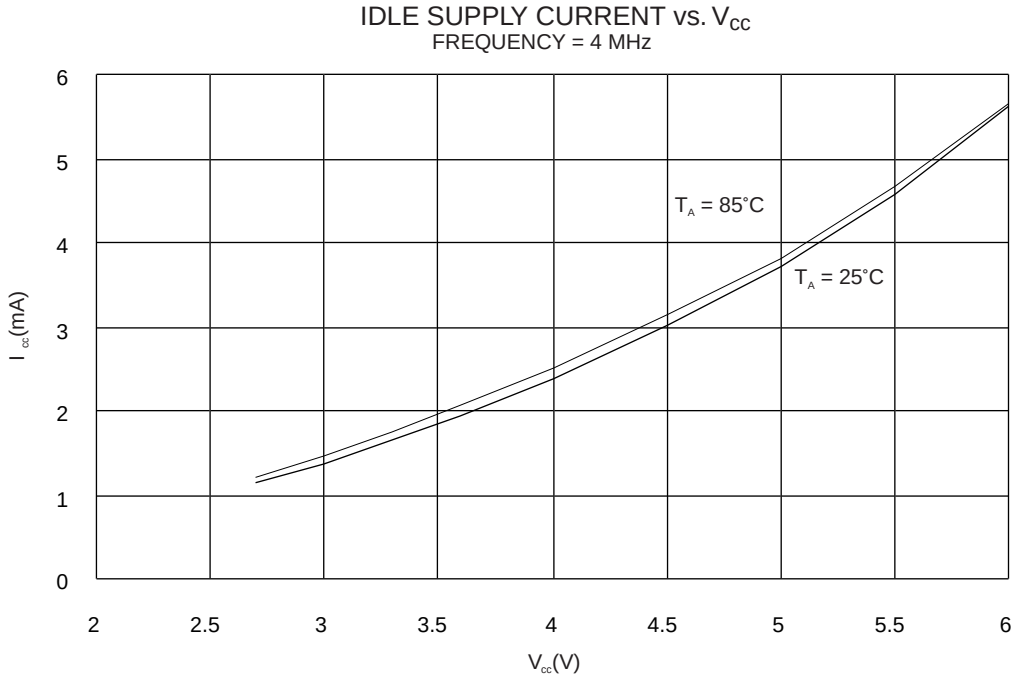


Figure 74. Power Down Supply Current vs. V_{CC}

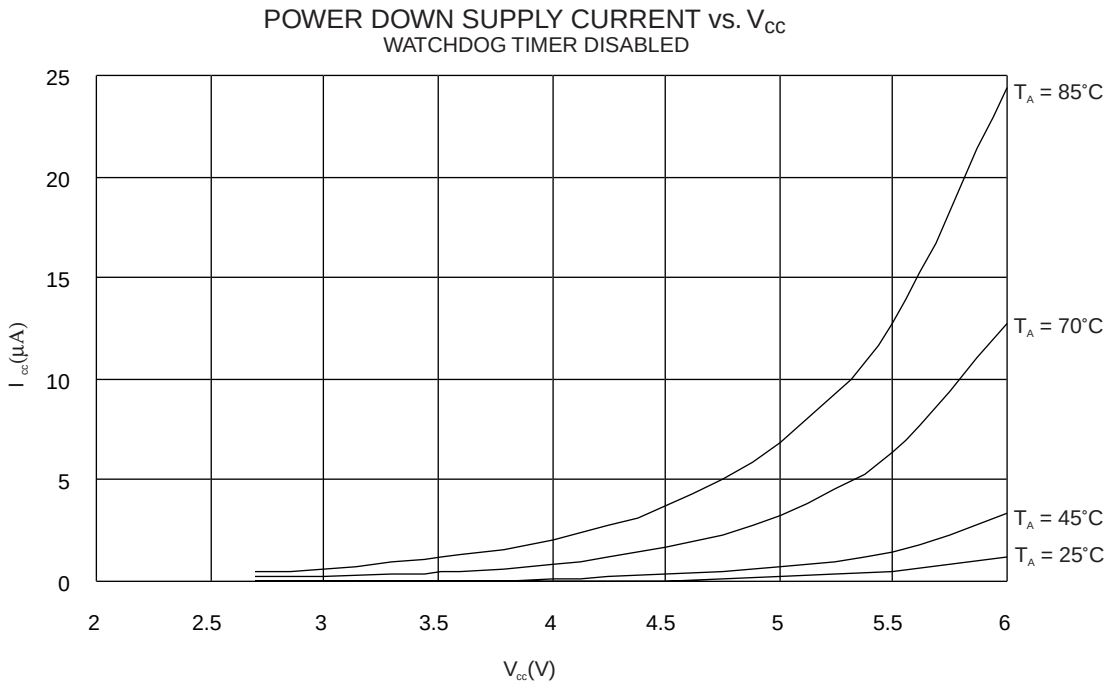


Figure 75. Power Down Supply Current vs. V_{CC}

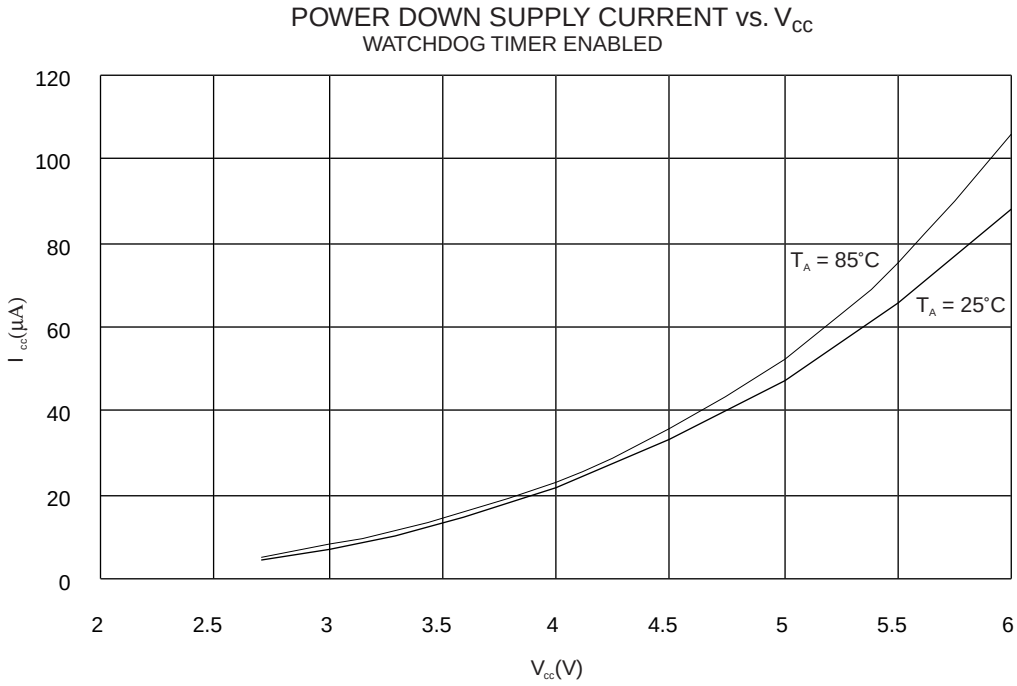


Figure 76. Power Down Supply Current vs. V_{CC}

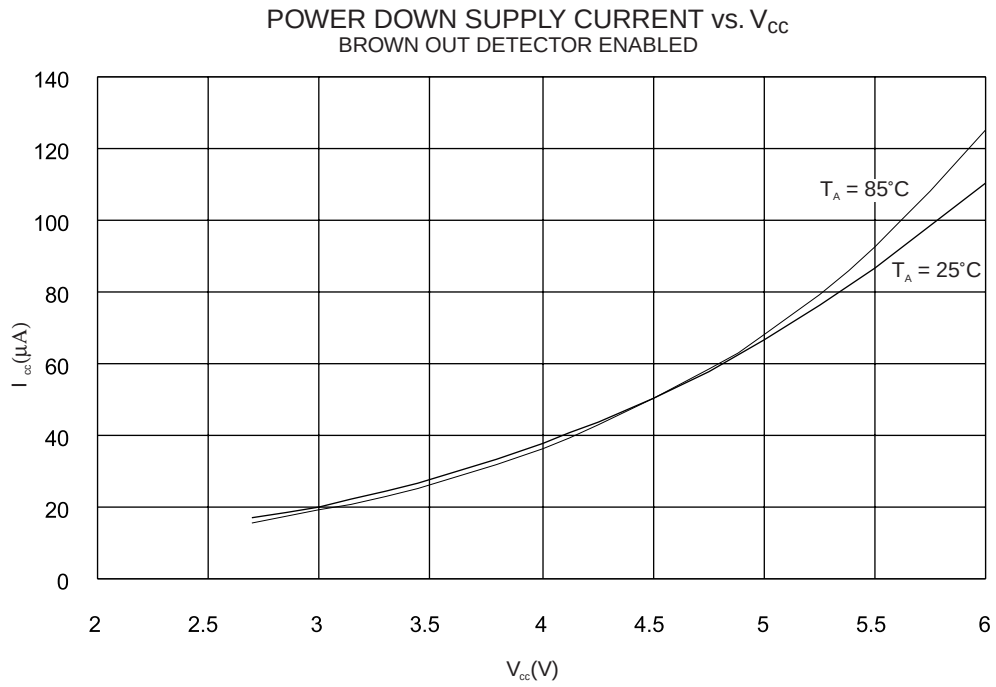
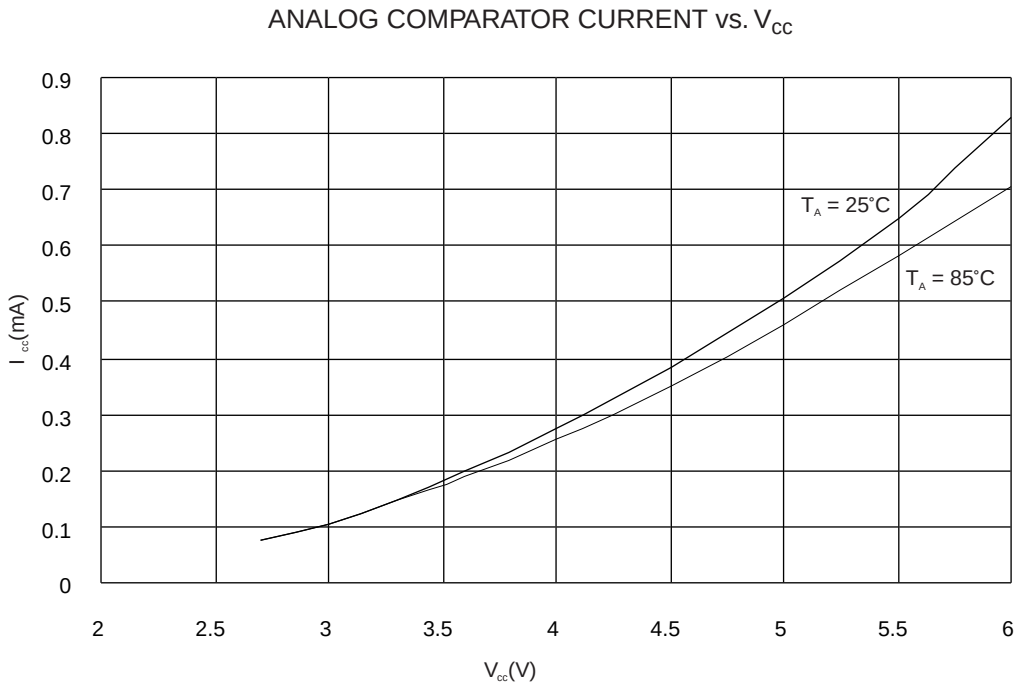


Figure 77. Analog Comparator Current vs. V_{CC}



Analog comparator offset voltage is measured as absolute offset

Figure 78. Analog Comparator Offset Voltage vs. Common Mode Voltage

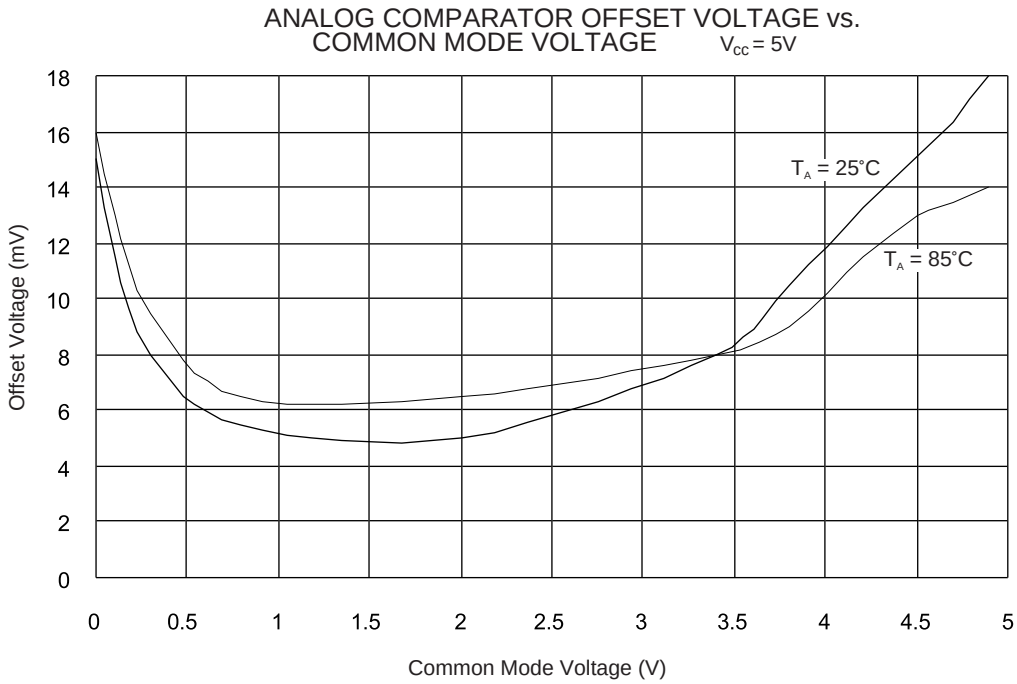


Figure 79. Analog Comparator Offset Voltage vs. Common Mode Voltage

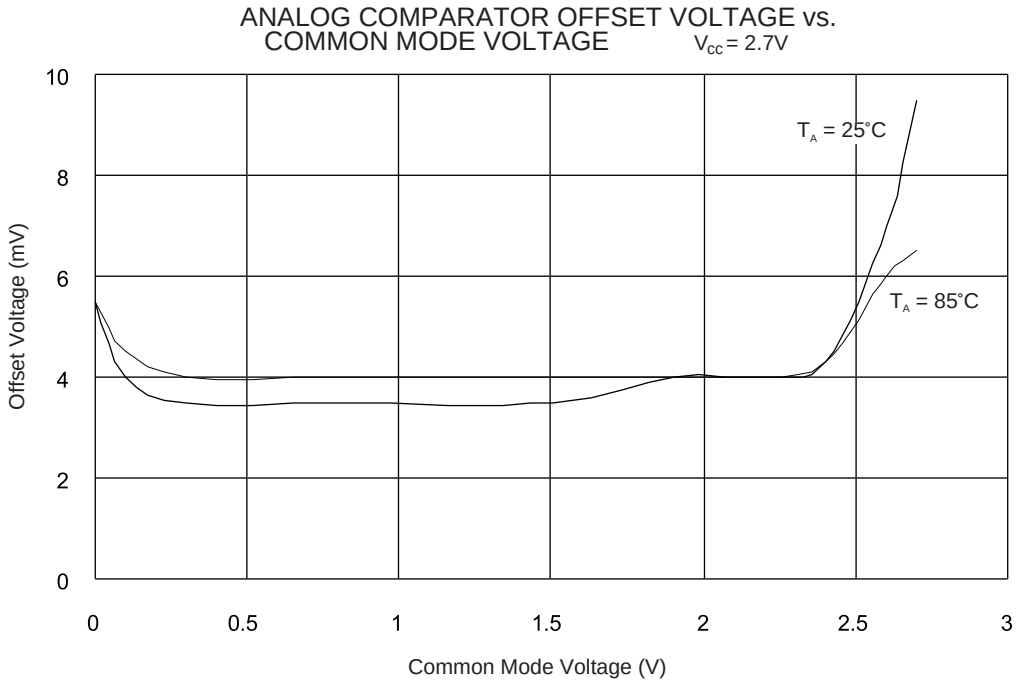


Figure 80. Analog Comparator Input Leakage Current

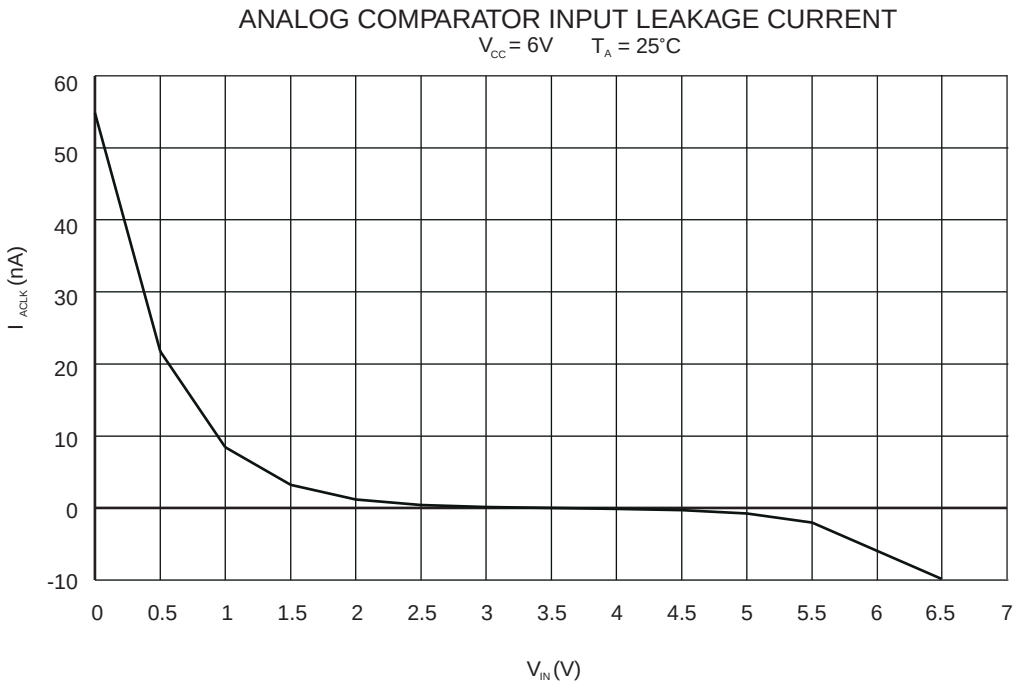
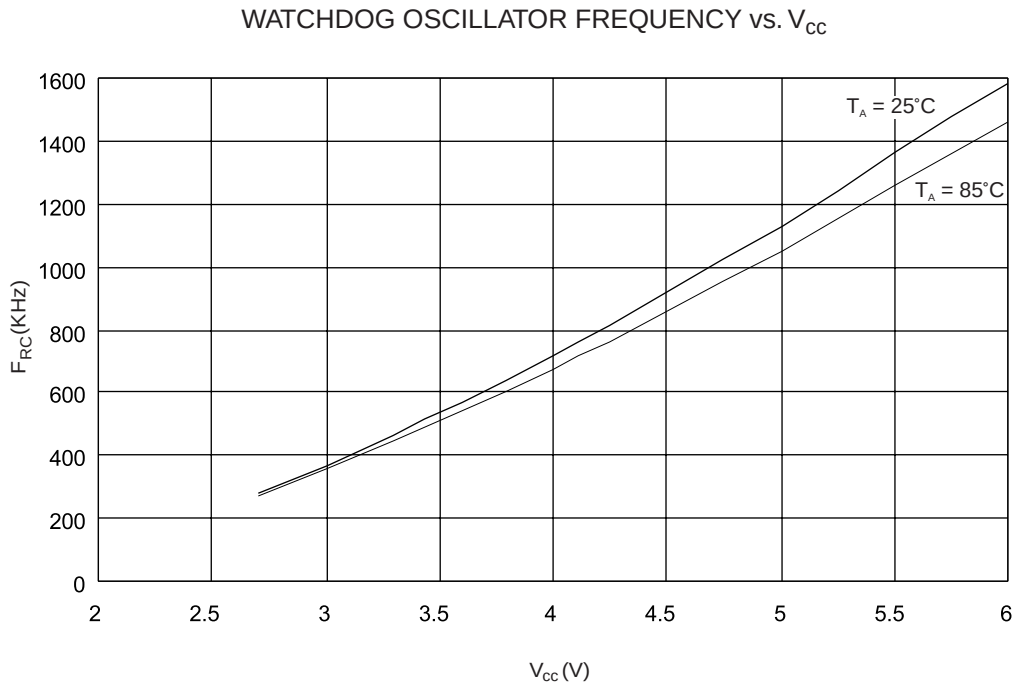


Figure 81. Watchdog Oscillator Frequency vs. V_{CC}



Sink and source capabilities of I/O ports are measured on one pin at a time.

Figure 82. Pull-Up Resistor Current vs. Input Voltage

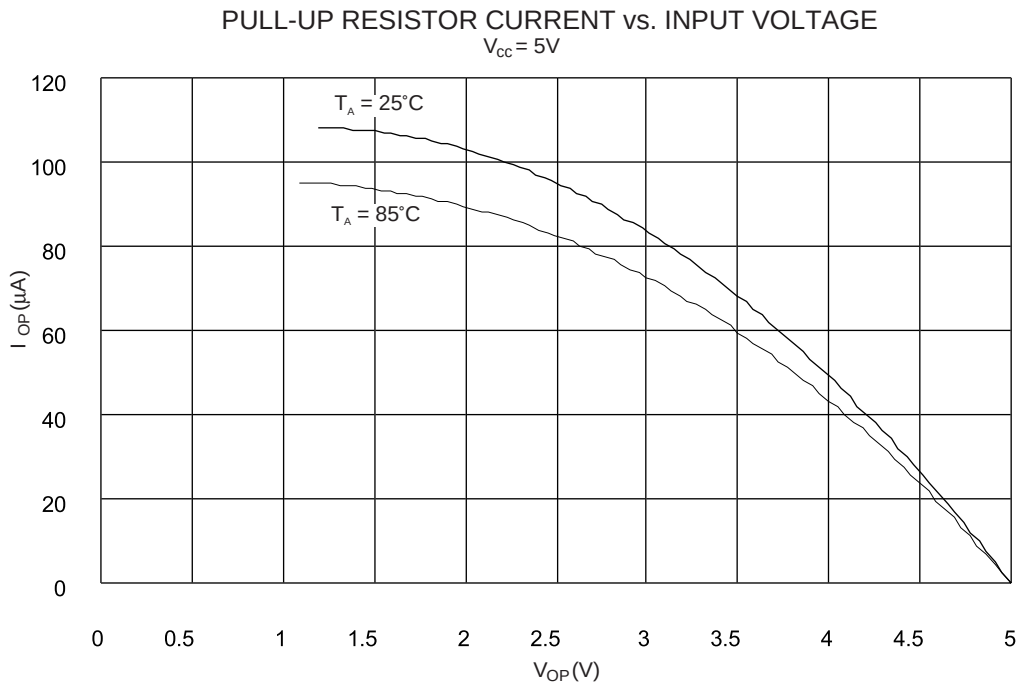


Figure 83. Pull-Up Resistor Current vs. Input Voltage

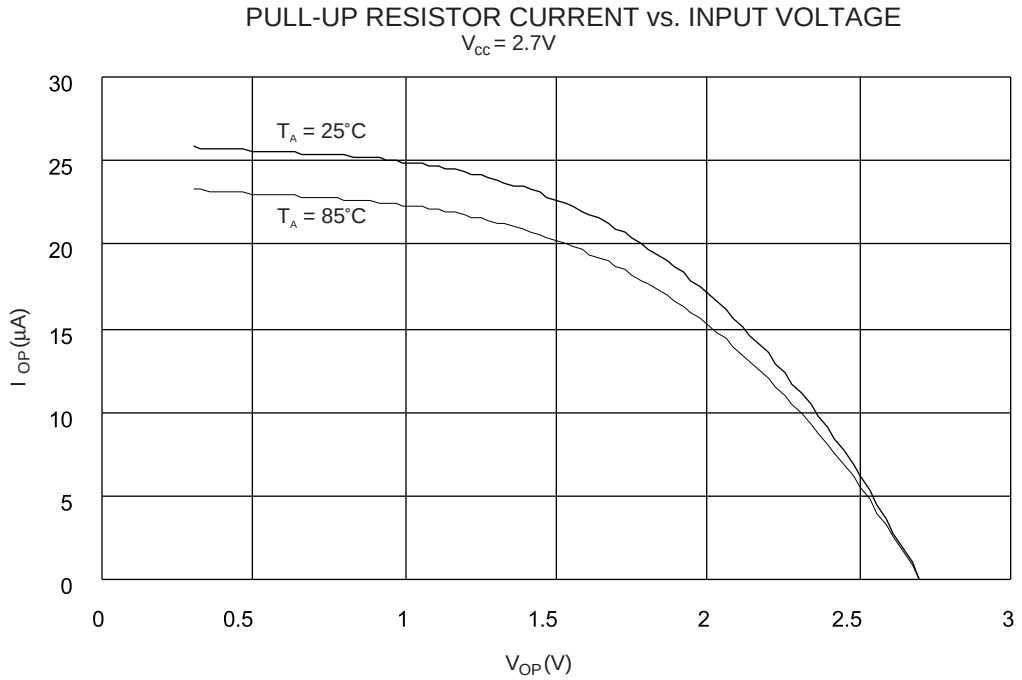


Figure 84. I/O Pin Sink Current vs. Output Voltage

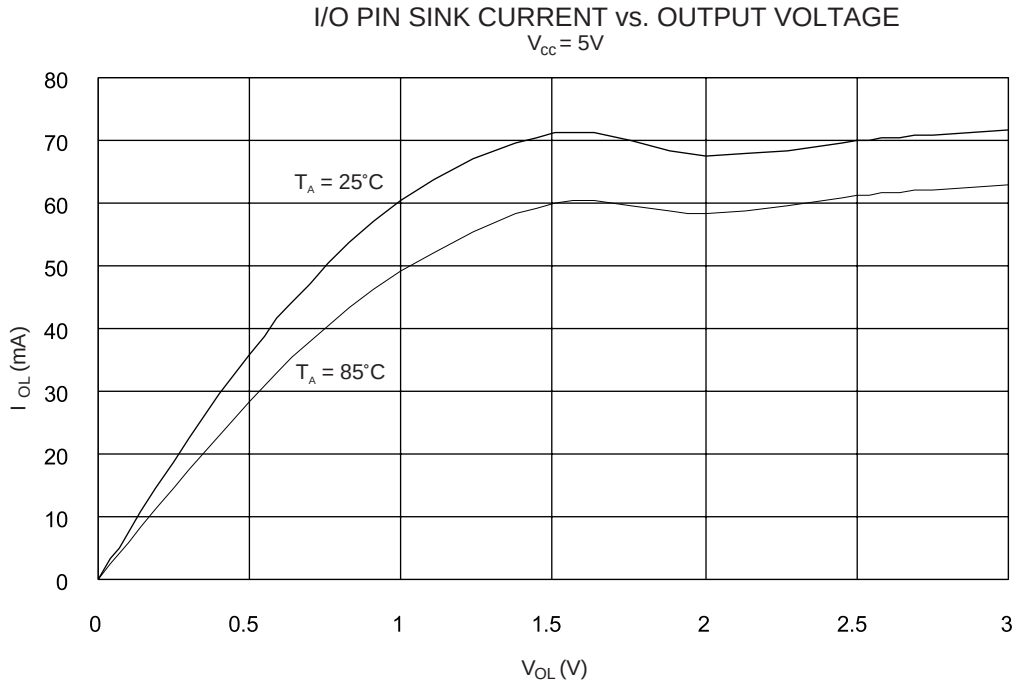


Figure 85. I/O Pin Source Current vs. Output Voltage

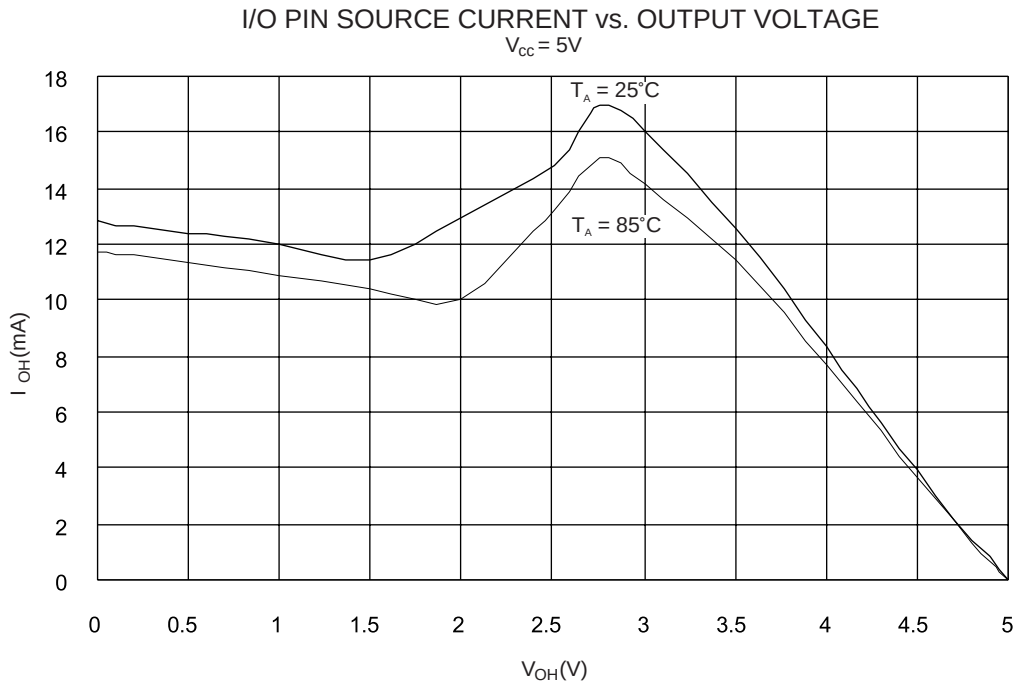


Figure 86. I/O Pin Sink Current vs. Output Voltage

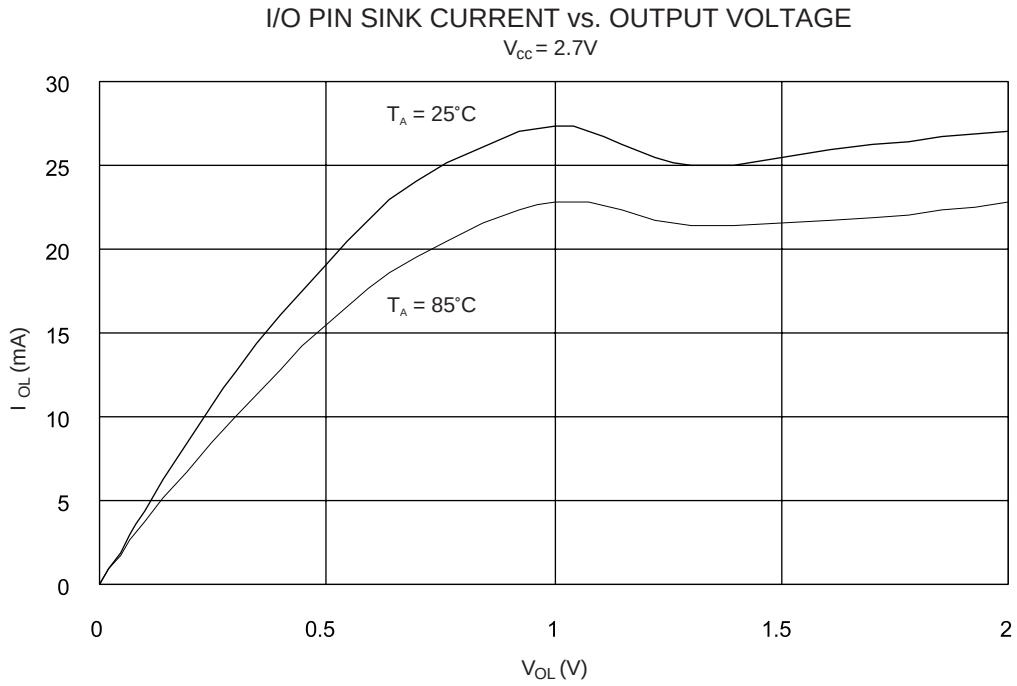


Figure 87. I/O Pin Source Current vs. Output Voltage

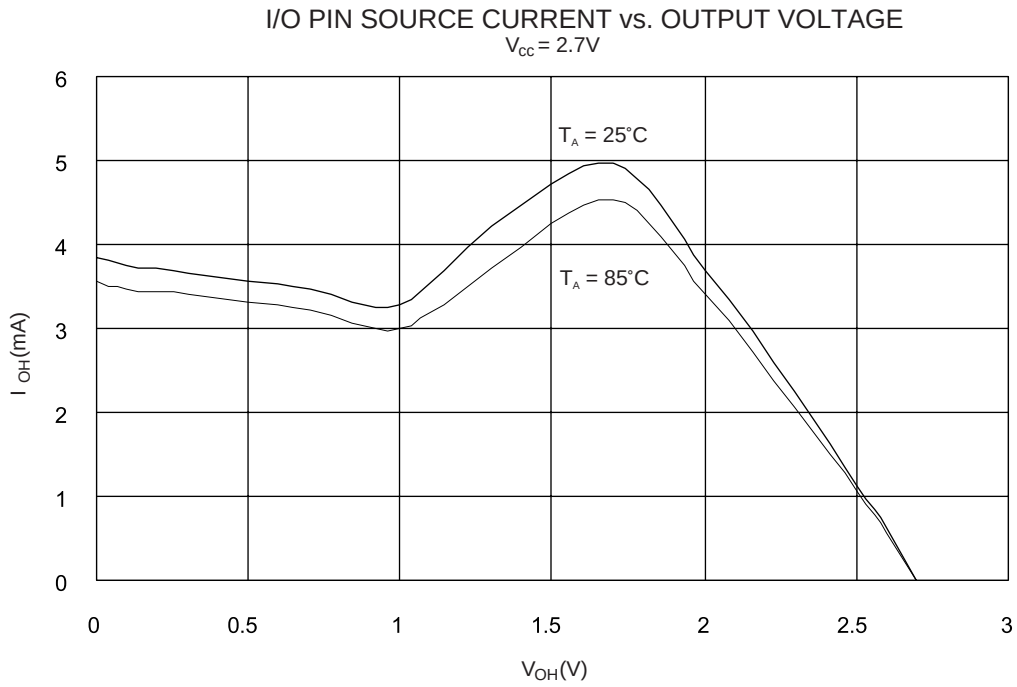


Figure 88. I/O Pin Input Threshold Voltage vs. V_{CC}

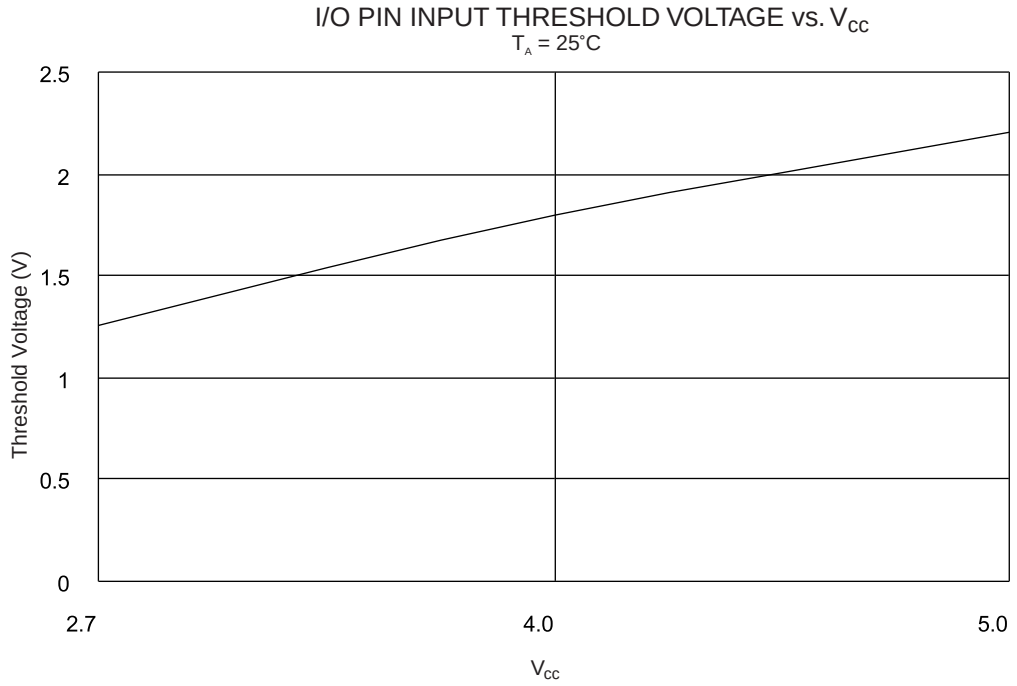
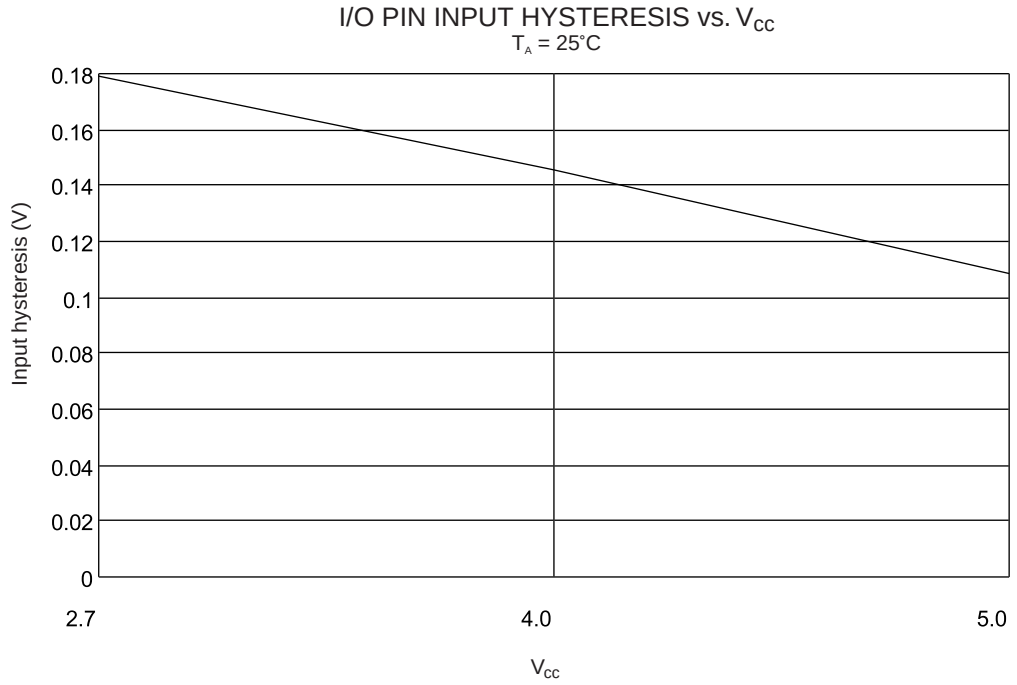


Figure 89. I/O Pin Input Hysteresis vs. V_{CC}



Register Summary

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
$3F ($5F)	SREG	I	T	H	S	V	N	Z	C	page 17
$3E ($5E)	Reserved	-	-	-	-	-	-	-	-	page 17
$3D ($5D)	SP	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	page 17
$3C ($5C)	Reserved									
$3B ($5B)	GIMSK	INT1	INT0	-	-	-	-	-	-	page 23
$3A ($5A)	GIFR	INTF1	INTF0							page 24
$39 ($59)	TIMSK	TOIE1	OCIE1	-	-	TICIE1	-	TOIE0	-	page 24
$38 ($58)	TIFR	TOV1	OCF1	-	-	ICF1	-	TOV0	-	page 25
$37 ($57)	Reserved									
$36 ($56)	Reserved									
$35 ($55)	MCUCR	-		SE	SM	ISC11	ISC10	ISC01	ISC00	page 26
$34 ($54)	MCUSR	-	-	-	-	WDRF	BORF	EXTRF	PORF	page 22
$33 ($53)	TCCR0	-	-	-	-	-	CS02	CS01	CS00	page 29
$32 ($52)	TCNT0	Timer/Counter0 (8 Bits)								page 30
$31 ($51)	Reserved									
$30 ($50)	Reserved									
$2F ($4F)	TCCR1A	COM11	COM10	-	-	-	-	PWM11	PWM10	page 31
$2E ($4E)	TCCR1B	ICNC1	ICES1	-	-	CTC1	CS12	CS11	CS10	page 32
$2D ($4D)	TCNT1H	Timer/Counter1 - Counter Register High Byte								page 33
$2C ($4C)	TCNT1L	Timer/Counter1 - Counter Register Low Byte								page 33
$2B ($4B)	OCR1H	Timer/Counter1 - Output Compare Register High Byte								page 34
$2A ($4A)	OCR1L	Timer/Counter1 - Output Compare Register Low Byte								page 34
$29 ($49)	Reserved									
$28 ($48)	Reserved									
$27 ($47)	ICR1H	Timer/Counter1 - Input Capture Register High Byte								page 34
$26 ($46)	ICR1L	Timer/Counter1 - Input Capture Register Low Byte								page 34
$25 ($45)	Reserved									
$24 ($44)	Reserved									
$23 ($43)	Reserved									
$22 ($42)	Reserved									
$21 ($41)	WDTCR	-	-	-	WDTOE	WDE	WDP2	WDP1	WDP0	page 36
$20 ($40)	Reserved									
$1F ($3F)	Reserved									
$1E ($3E)	EEAR	EEPROM Address Register								page 38
$1D ($3D)	EEDR	EEPROM Data Register								page 38
$1C ($3C)	EECR	-	-	-	-	EERIE	EEMWE	EEWE	EERE	page 38
$1B ($3B)	Reserved									
$1A ($3A)	Reserved									
$19 ($39)	Reserved									
$18 ($38)	PORTB	-	-	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	page 59
$17 ($37)	DDRB	-	-	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	page 59
$16 ($36)	PINB	-	-	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	page 59
$15 ($35)	PORTC	-	-	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	page 64
$14 ($34)	DDRC	-	-	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	page 64
$13 ($33)	PINC	-	-	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	page 64
$12 ($32)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	page 66
$11 ($31)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	page 66
$10 ($30)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	page 66
$0F ($2F)	SPDR	SPI Data Register								page 43
$0E ($2E)	SPSR	SPIF	WCOL	-	-	-	-	-	-	page 43
$0D ($2D)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	page 42
$0C ($2C)	UDR	UART I/O Data Register								page 47
$0B ($2B)	UCSRA	RXC	TXC	UDRE	FE	OR	-	-	-	page 47
$0A ($2A)	UCSRB	RXCIE	TXCIE	UDRIE	RXEN	TXEN	CHR9	RXB8	TXB8	page 48
$09 ($29)	UBRR	UART Baud Rate Register								page 50
$08 ($28)	ACSR	ACD	AINBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	page 50
$07 ($27)	ADMUX	-	ADCBG	-	-	-	MUX2	MUX1	MUX0	page 55
$06 ($26)	ADCSR	ADEN	ADSC	ADFR	ADIF	ADIE	ADPS2	ADPS1	ADPS0	page 56
$05 ($25)	ADCH	-	-	-	-	-	-	ADC9	ADC8	page 57
$04 ($24)	ADCL	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0	page 57
$03 ($23)	UBRRHI					UART Baud Rate Register High				page 50

Register Summary (Continued)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
$02 ($22)	Reserved									
$01 ($21)	Reserved									
$00 ($20)	Reserved									

Notes: 1. For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.

2. Some of the status flags are cleared by writing a logical one to them. Note that the CBI and SBI instructions will operate on all bits in the I/O register, writing a one back into any flag read as set, thus clearing the flag. The CBI and SBI instructions work with registers $00 to $1F only.

Instruction Set Summary

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

Instruction Set Summary (Continued)

Mnemonics	Operands	Description	Operation	Flags	#Clocks
DATA TRANSFER INSTRUCTIONS					
MOV	Rd, Rr	Move Between Registers	Rd ← Rr	None	1
LDI	Rd, K	Load Immediate	Rd ← K	None	1
LD	Rd, X	Load Indirect	Rd ← (X)	None	2
LD	Rd, X+	Load Indirect and Post-Inc.	Rd ← (X), X ← X + 1	None	2
LD	Rd, - X	Load Indirect and Pre-Dec.	X ← X - 1, Rd ← (X)	None	2
LD	Rd, Y	Load Indirect	Rd ← (Y)	None	2
LD	Rd, Y+	Load Indirect and Post-Inc.	Rd ← (Y), Y ← Y + 1	None	2
LD	Rd, - Y	Load Indirect and Pre-Dec.	Y ← Y - 1, Rd ← (Y)	None	2
LDD	Rd,Y+q	Load Indirect with Displacement	Rd ← (Y + q)	None	2
LD	Rd, Z	Load Indirect	Rd ← (Z)	None	2
LD	Rd, Z+	Load Indirect and Post-Inc.	Rd ← (Z), Z ← Z+1	None	2
LD	Rd, -Z	Load Indirect and Pre-Dec.	Z ← Z - 1, Rd ← (Z)	None	2
LDD	Rd, Z+q	Load Indirect with Displacement	Rd ← (Z + q)	None	2
LDS	Rd, k	Load Direct from SRAM	Rd ← (k)	None	2
ST	X, Rr	Store Indirect	(X) ← Rr	None	2
ST	X+, Rr	Store Indirect and Post-Inc.	(X) ← Rr, X ← X + 1	None	2
ST	- X, Rr	Store Indirect and Pre-Dec.	X ← X - 1, (X) ← Rr	None	2
ST	Y, Rr	Store Indirect	(Y) ← Rr	None	2
ST	Y+, Rr	Store Indirect and Post-Inc.	(Y) ← Rr, Y ← Y + 1	None	2
ST	- Y, Rr	Store Indirect and Pre-Dec.	Y ← Y - 1, (Y) ← Rr	None	2
STD	Y+q,Rr	Store Indirect with Displacement	(Y + q) ← Rr	None	2
ST	Z, Rr	Store Indirect	(Z) ← Rr	None	2
ST	Z+, Rr	Store Indirect and Post-Inc.	(Z) ← Rr, Z ← Z + 1	None	2
ST	-Z, Rr	Store Indirect and Pre-Dec.	Z ← Z - 1, (Z) ← Rr	None	2
STD	Z+q,Rr	Store Indirect with Displacement	(Z + q) ← Rr	None	2
STS	k, Rr	Store Direct to SRAM	(k) ← Rr	None	2
LPM		Load Program Memory	R0 ← (Z)	None	3
IN	Rd, P	In Port	Rd ← P	None	1
OUT	P, Rr	Out Port	P ← Rr	None	1
PUSH	Rr	Push Register on Stack	STACK ← Rr	None	2
POP	Rd	Pop Register from Stack	Rd ← STACK	None	2
BIT AND BIT-TEST INSTRUCTIONS					
SBI	P,b	Set Bit in I/O Register	I/O(P,b) ← 1	None	2
CBI	P,b	Clear Bit in I/O Register	I/O(P,b) ← 0	None	2
LSL	Rd	Logical Shift Left	Rd(n+1) ← Rd(n), Rd(0) ← 0	Z,C,N,V	1
LSR	Rd	Logical Shift Right	Rd(n) ← Rd(n+1), Rd(7) ← 0	Z,C,N,V	1
ROL	Rd	Rotate Left Through Carry	Rd(0)←C,Rd(n+1)← Rd(n),C←Rd(7)	Z,C,N,V	1
ROR	Rd	Rotate Right Through Carry	Rd(7)←C,Rd(n)← Rd(n+1),C←Rd(0)	Z,C,N,V	1
ASR	Rd	Arithmetic Shift Right	Rd(n) ← Rd(n+1), n=0..6	Z,C,N,V	1
SWAP	Rd	Swap Nibbles	Rd(3..0)←Rd(7..4),Rd(7..4)←Rd(3..0)	None	1
BSET	s	Flag Set	SREG(s) ← 1	SREG(s)	1
BCLR	s	Flag Clear	SREG(s) ← 0	SREG(s)	1
BST	Rr, b	Bit Store from Register to T	T ← Rr(b)	T	1
BLD	Rd, b	Bit load from T to Register	Rd(b) ← T	None	1
SEC		Set Carry	C ← 1	C	1
CLC		Clear Carry	C ← 0	C	1
SEN		Set Negative Flag	N ← 1	N	1
CLN		Clear Negative Flag	N ← 0	N	1
SEZ		Set Zero Flag	Z ← 1	Z	1
CLZ		Clear Zero Flag	Z ← 0	Z	1
SEI		Global Interrupt Enable	I ← 1	I	1
CLI		Global Interrupt Disable	I ← 0	I	1
SES		Set Signed Test Flag	S ← 1	S	1
CLS		Clear Signed Test Flag	S ← 0	S	1
SEV		Set Twos Complement Overflow.	V ← 1	V	1
CLV		Clear Twos Complement Overflow	V ← 0	V	1
SET		Set T in SREG	T ← 1	T	1
CLT		Clear T in SREG	T ← 0	T	1
SEH		Set Half Carry Flag in SREG	H ← 1	H	1
CLH		Clear Half Carry Flag in SREG	H ← 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

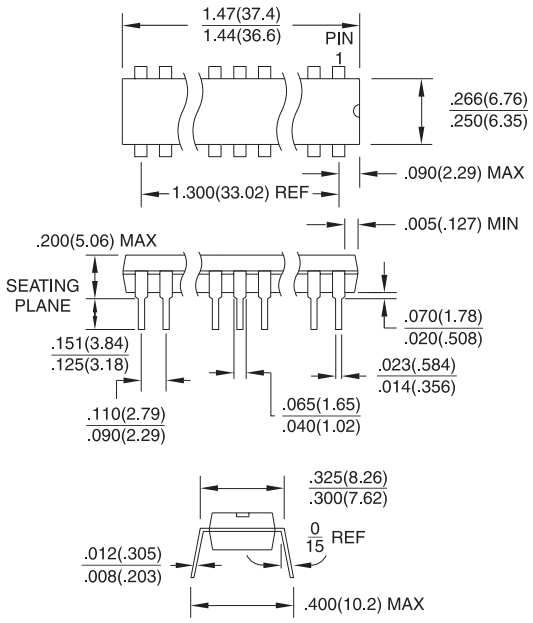
Ordering Information

Power Supply	Speed (MHz)	Ordering Code	Package	Operation Range
2.7 - 6.0V	4	AT90LS2333-4AC AT90LS2333-4PC	32A 28P3	Commercial (0°C to 70°C)
		AT90LS2333-4AI AT90LS2333-4PI	32A 28P3	Industrial (-40°C to 85°C)
4.0 - 6.0V	8	AT90S2333-8AC AT90S2333-8PC	32A 28P3	Commercial (0°C to 70°C)
		AT90S2333-8AI AT90S2333-8PI	32A 28P3	Industrial (-40°C to 85°C)
2.7 - 6.0V	4	AT90LS4433-4AC AT90LS4433-4PC	32A 28P3	Commercial (0°C to 70°C)
		AT90LS4433-4AI AT90LS4433-4PI	32A 28P3	Industrial (-40°C to 85°C)
4.0 - 6.0V	8	AT90S4433-8AC AT90S4433-8PC	32A 28P3	Commercial (0°C to 70°C)
		AT90S4433-8AI AT90S4433-8PI	32A 28P3	Industrial (-40°C to 85°C)

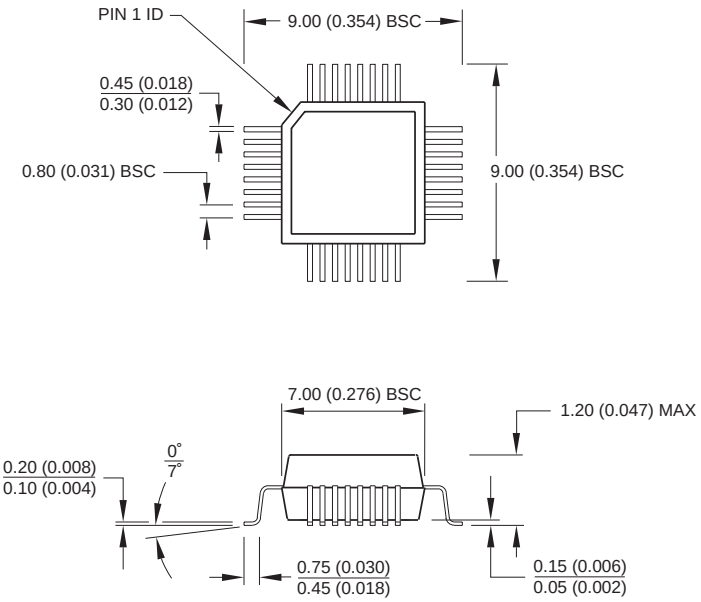
Package Type	
28P3	28-lead, 0.300” Wide, Plastic Dual in Line Package (PDIP)
32A	32-lead, Thin (1.0 mm) Plastic Gull Wing Quad Flat Package (TQFP)

Packaging Information

28P3, 28-lead, 0.300” Wide,
Plastic Dual Inline Package (PDIP)
Dimensions in Inches and (Millimeters)



32A, 32-lead, Thin (1.0 mm) Plastic Gull Wing Quad
Flat Package (TQFP)
Dimensions in Millimeters and (Inches)



Atmel Headquarters

Corporate Headquarters
2325 Orchard Parkway
San Jose, CA 95131
TEL (408) 441-0311
FAX (408) 487-2600

Europe
Atmel U.K., Ltd.
Coliseum Business Centre
Riverside Way
Camberley, Surrey GU15 3YL
England
TEL (44) 1276-686-677
FAX (44) 1276-686-697

Asia
Atmel Asia, Ltd.
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimhatsui
East Kowloon
Hong Kong
TEL (852) 2721-9778
FAX (852) 2722-1369

Japan
Atmel Japan K.K.
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
TEL (81) 3-3523-3551
FAX (81) 3-3523-7581

Atmel Operations

Atmel Colorado Springs
1150 E. Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL (719) 576-3300
FAX (719) 540-1759

Atmel Rousset
Zone Industrielle
13106 Rousset Cedex
France
TEL (33) 4-4253-6000
FAX (33) 4-4253-6001

Fax-on-Demand
North America:
1-(800) 292-8635
International:
1-(408) 441-0732

e-mail
literature@atmel.com

Web Site
http://www.atmel.com

BBS
1-(408) 436-4309

© Atmel Corporation 1999.
Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not authorized for use as critical components in life support devices or systems.

Marks bearing ® and/or ™ are registered trademarks and trademarks of Atmel Corporation.

Terms and product names in this document may be trademarks of others.

Printed on recycled paper.

1042D–04/99/xM