



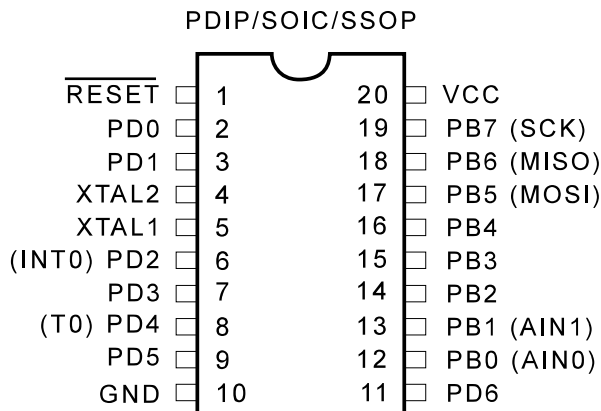
**THE DATASHEET OF
AT90S1200-4SI**



Features

- Utilizes the AVR[®] RISC Architecture
- AVR – High-performance and Low-power RISC Architecture
 - 89 Powerful Instructions – Most Single Clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Up to 12 MIPS Throughput at 12 MHz
- Data and Non-volatile Program Memory
 - 1K Byte of In-System Programmable Flash
Endurance: 1,000 Write/Erase Cycles
 - 64 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
 - On-chip Analog Comparator
 - Programmable Watchdog Timer with On-chip Oscillator
 - SPI Serial Interface for In-System Programming
- Special Microcontroller Features
 - Low-power Idle and Power-down Modes
 - External and Internal Interrupt Sources
 - Selectable On-chip RC Oscillator for Zero External Components
- Specifications
 - Low-power, High-speed CMOS Process Technology
 - Fully Static Operation
- Power Consumption at 4 MHz, 3V, 25°C
 - Active: 2.0 mA
 - Idle Mode: 0.4 mA
 - Power-down Mode: <1 µA
- I/O and Packages
 - 15 Programmable I/O Lines
 - 20-pin PDIP, SOIC and SSOP
- Operating Voltages
 - 2.7 - 6.0V (AT90S1200-4)
 - 4.0 - 6.0V (AT90S1200-12)
- Speed Grades
 - 0 - 4 MHz, (AT90S1200-4)
 - 0 - 12 MHz, (AT90S1200-12)

Pin Configuration



8-bit AVR[®]
Microcontroller
with 1K Byte
of In-System
Programmable
Flash

AT90S1200



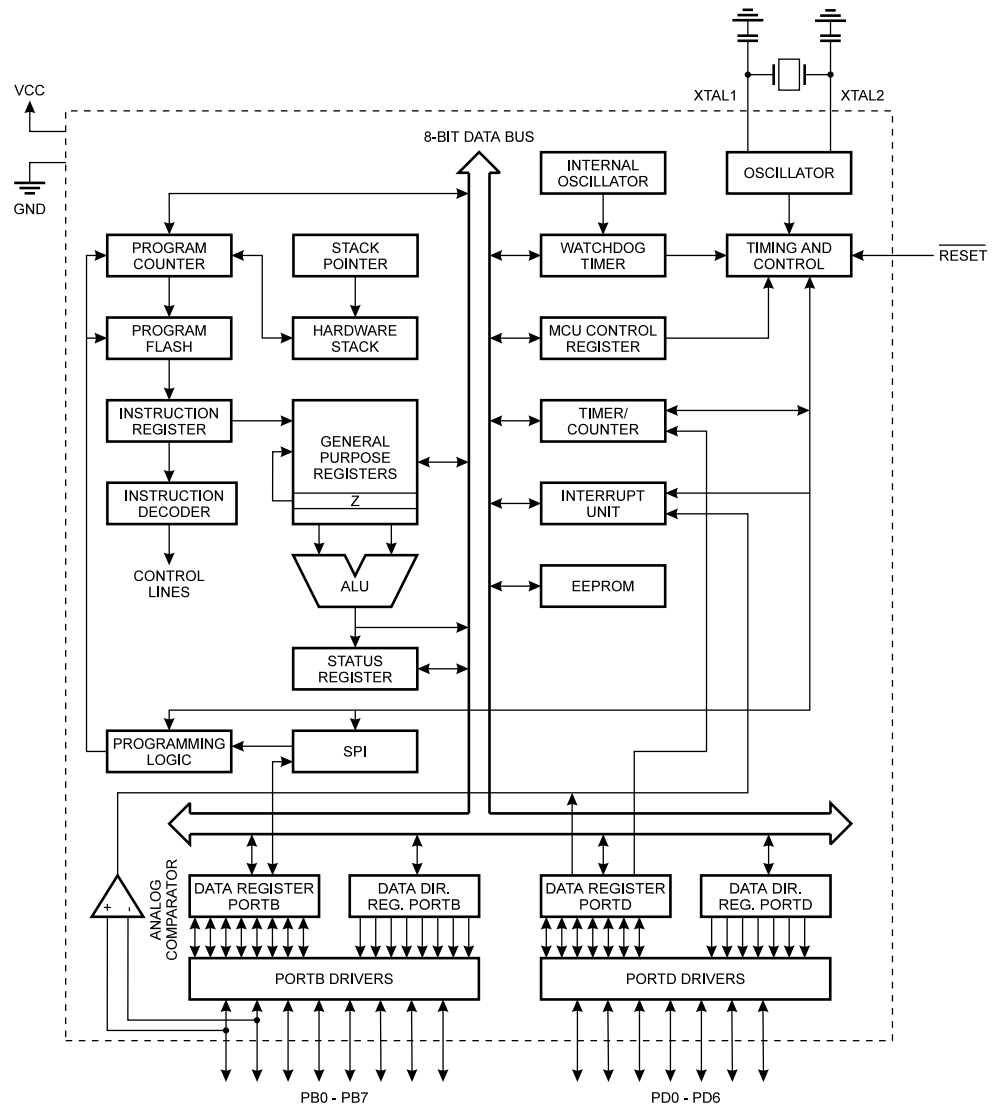
Description

The AT90S1200 is a low-power CMOS 8-bit microcontroller based on the AVR RISC architecture. By executing powerful instructions in a single clock cycle, the AT90S1200 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 the 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.

Block Diagram

Figure 1. The AT90S1200 Block Diagram



The architecture supports high-level languages efficiently as well as extremely dense assembler code programs. The AT90S1200 provides the following features: 1K byte of In-System Programmable Flash, 64 bytes EEPROM, 15 general purpose I/O lines, 32 general purpose working registers, internal and external interrupts, programmable watchdog timer with internal oscillator, an SPI serial port for program downloading and two software selectable power-saving modes. The Idle Mode stops the CPU while allow-

ing the Registers, Timer/Counter, Watchdog 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 External Interrupt or hardware Reset.

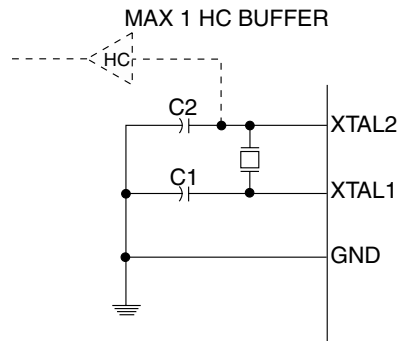
The device is manufactured using Atmel's high-density nonvolatile memory technology. The On-chip In-System Programmable Flash allows the program memory to be reprogrammed in-system through an SPI serial interface or by a conventional nonvolatile memory programmer. By combining an enhanced RISC 8-bit CPU with In-System Programmable Flash on a monolithic chip, the Atmel AT90S1200 is a powerful microcontroller that provides a highly flexible and cost-effective solution to many embedded control applications.

The AT90S1200 AVR is supported with a full suite of program and system development tools including: macro assemblers, program debugger/simulators, in-circuit emulators, and evaluation kits.

Pin Descriptions

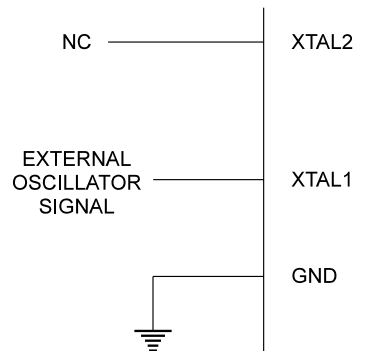
| | |
|---------------------------|---|
| VCC | Supply voltage pin. |
| GND | Ground pin. |
| Port B (PB7..PB0) | <p>Port B is an 8-bit bi-directional I/O port. Port pins can provide internal pull-up resistors (selected for each bit). PB0 and PB1 also serve as the positive input (AIN0) and the negative input (AIN1), respectively, of the On-chip Analog Comparator. The Port B output buffers can sink 20 mA 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 are tri-stated when a reset condition becomes active, even if the clock is not active.</p> <p>Port B also serves the functions of various special features of the AT90S1200 as listed on page 30.</p> |
| Port D (PD6..PD0) | <p>Port D has seven bi-directional I/O pins with internal pull-up resistors, PD6..PD0. 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. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not active.</p> <p>Port D also serves the functions of various special features of the AT90S1200 as listed on page 34.</p> |
| RESET | Reset input. A low level on this pin for more 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. |
| Crystal Oscillator | <p>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. Either a quartz crystal or a ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 3.</p> |

Figure 2. Oscillator Connections



Note: When using the MCU Oscillator as a clock for an external device, an HC buffer should be connected as indicated in the figure.

Figure 3. External Clock Drive Configuration



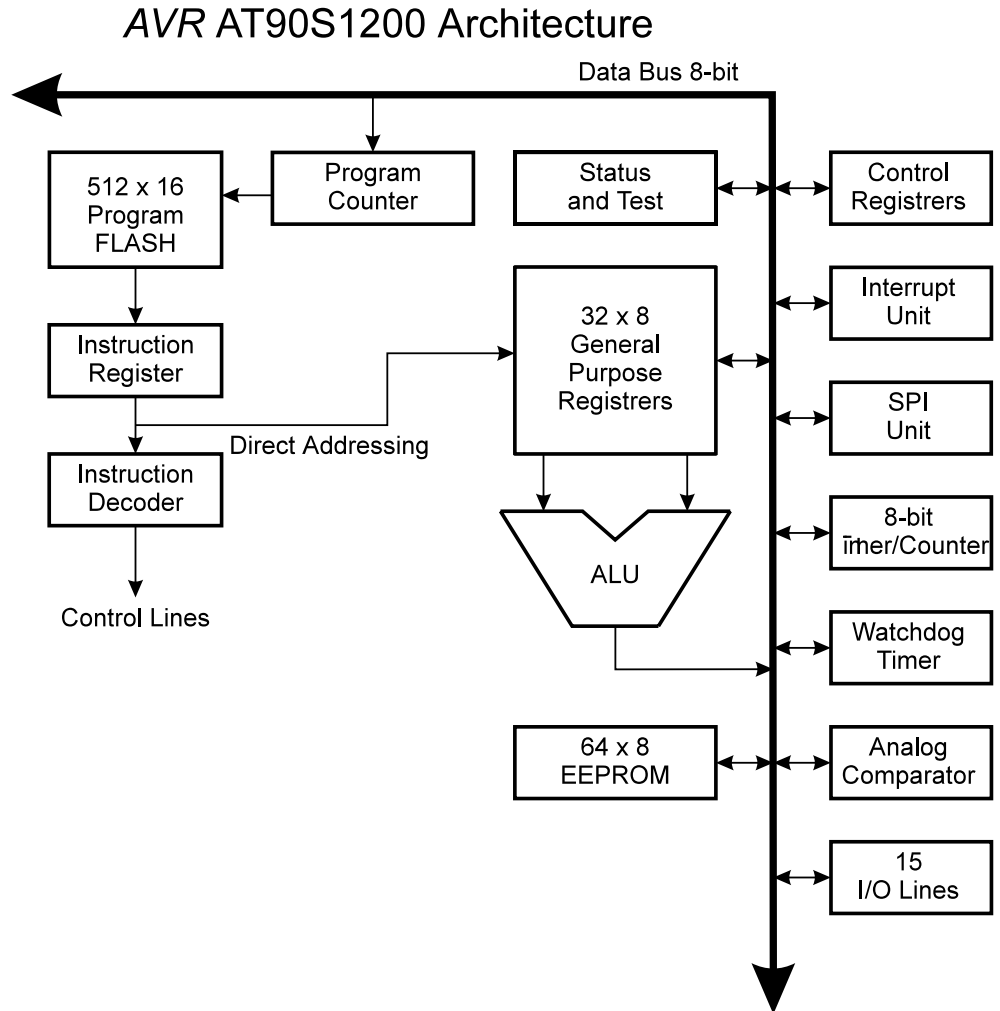
On-chip RC Oscillator

An On-chip RC Oscillator running at a fixed frequency of 1 MHz can be selected as the MCU clock source. If enabled, the AT90S1200 can operate with no external components. A control bit (RCEN) in the Flash Memory selects the On-chip RC Oscillator as the clock source when programmed (“0”). The AT90S1200 is normally shipped with this bit unprogrammed (“1”). Parts with this bit programmed can be ordered as AT90S1200A. The RCEN-bit can be changed by parallel programming only. When using the On-chip RC Oscillator for Serial Program downloading, the RCEN bit must be programmed in Parallel Programming mode first.

Architectural Overview

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

Figure 4. The AT90S1200 AVR RISC Architecture



The ALU supports arithmetic and logic functions between registers or between a constant and a register. Single register operations are also executed in the ALU. Figure 4 shows the AT90S1200 AVR RISC microcontroller architecture. The AVR uses a Harvard architecture concept – with separate memories and buses for program and data memories. The program memory is accessed with a 2-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 relative call instructions, the whole 512 address space is directly accessed. All AVR instructions have a single 16-bit word format, meaning that every program memory address contains a single 16-bit instruction.

During interrupts and subroutine calls, the return address Program Counter (PC) is stored on the stack. The stack is a 3-level-deep hardware stack dedicated for subroutines and interrupts.

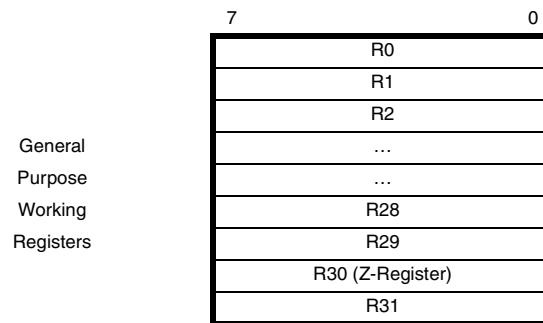
The I/O memory space contains 64 addresses for CPU peripheral functions such as Control Registers, Timer/Counters, A/D Converters and other I/O functions. The memory spaces in the AVR architecture are all linear and regular memory maps.

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

Figure 5. 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 exception is the five constant arithmetic and logic instructions SBCI, SUBI, CPI, ANDI, 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, OR and all other operations between two registers or on a single register apply to the entire register file.

Register 30 also serves as an 8-bit pointer for indirect address of the register file.

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, logic and bit-functions.

In-System Programmable Flash Program Memory

The AT90S1200 contains 1K bytes On-chip In-System Programmable Flash memory for program storage. Since all instructions are single 16-bit words, the Flash is organized as 512 x 16. The Flash memory has an endurance of at least 1000 write/erase cycles.

The AT90S1200 Program Counter is 9 bits wide, thus addressing the 512 words Flash program memory.

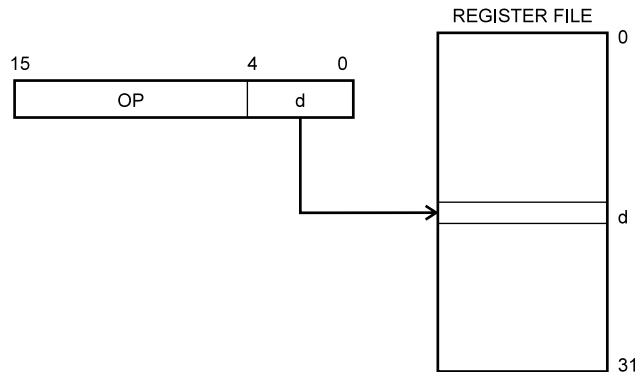
See page 37 for a detailed description on Flash data downloading.

Program and Data Addressing Modes

The AT90S1200 AVR RISC Microcontroller supports powerful and efficient addressing modes. This section describes the different addressing modes supported in the AT90S1200. 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

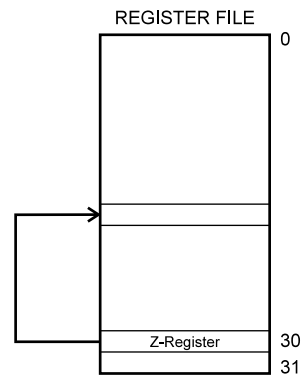
Figure 6. Direct Single Register Addressing



The operand is contained in register d (Rd).

Register Indirect

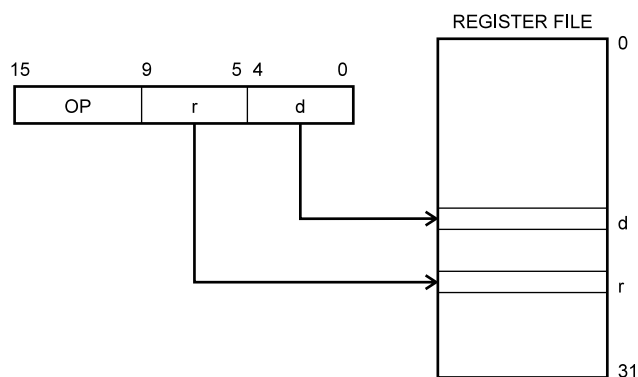
Figure 7. Indirect Register Addressing



The register accessed is the one pointed to by the Z-register (R30).

Register Direct, Two Registers Rd and Rr

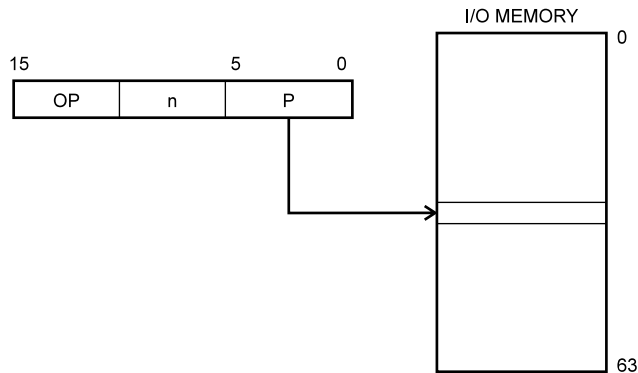
Figure 8. Direct Register Addressing, Two Registers



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

I/O Direct

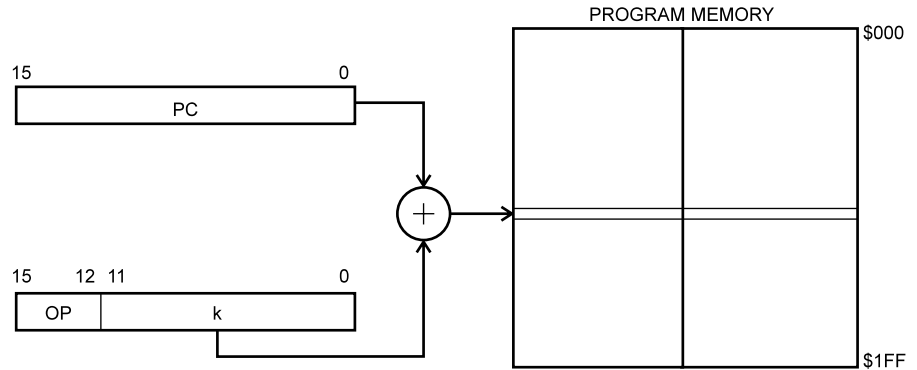
Figure 9. I/O Direct Addressing



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

Relative Program Addressing, RJMP and RCALL

Figure 10. Relative Program Memory Addressing



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

Subroutine and Interrupt Hardware Stack

The AT90S1200 uses a 3 level deep hardware stack for subroutines and interrupts. The hardware stack is 9 bits wide and stores the Program Counter (PC) return address while subroutines and interrupts are executed.

RCALL instructions and interrupts push the PC return address onto stack level 0, and the data in the other stack levels 1 - 2 are pushed one level deeper in the stack. When a RET or RETI instruction is executed the returning PC is fetched from stack level 0, and the data in the other stack levels 1 - 2 are popped one level in the stack.

If more than three subsequent subroutine calls or interrupts are executed, the first values written to the stack are overwritten.

EEPROM Data Memory

The AT90S1200 contains 64 bytes of data EEPROM memory. It is organized as a separate data space, in which single bytes can be read and written. The EEPROM has an endurance of at least 100,000 write/erase cycles. The access between the EEPROM and the CPU is described on page 25 specifying the EEPROM address register, the EEPROM data register, and the EEPROM control register. For the SPI data downloading, see page 44 for a detailed description.

Instruction Execution Timing

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

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

Figure 11 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 11. The Parallel Instruction Fetches and Instruction Executions

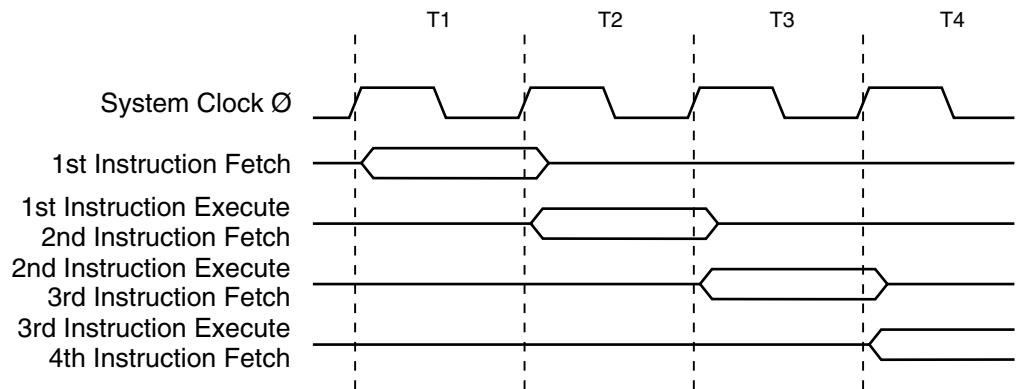
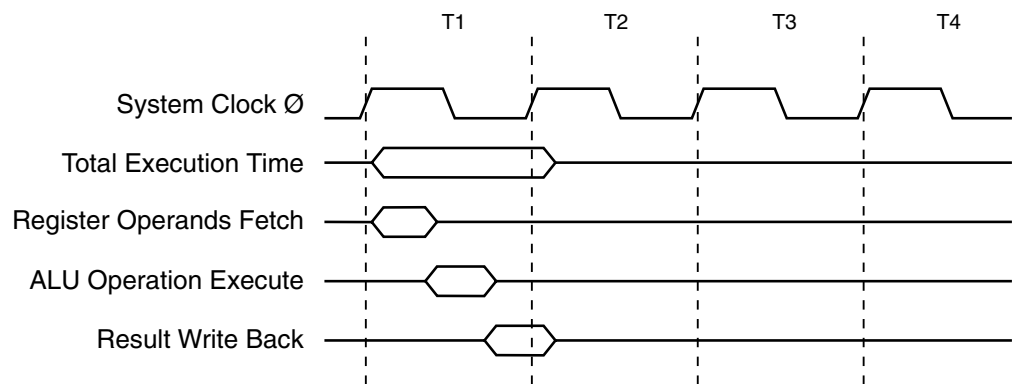


Figure 12 shows the internal timing concept for the register file. In a single clock cycle an ALU operation using two register operands is executed, and the result is stored back to the destination register.

Figure 12. Single-cycle ALU Operation



I/O Memory

The I/O space definition of the AT90S1200 is shown in the following table.

Table 1. The AT90S1200 I/O Space

| Address Hex | Name | Function |
|-------------|-------|---|
| \$3F | SREG | Status REGISTER |
| \$3B | GIMSK | General Interrupt MaSK register |
| \$39 | TIMSK | Timer/Counter Interrupt MaSK register |
| \$38 | TIFR | Timer/Counter Interrupt Flag register |
| \$35 | MCUCR | MCU general Control Register |
| \$33 | TCCR0 | Timer/Counter0 Control Register |
| \$32 | TCNT0 | Timer/Counter0 (8-bit) |
| \$21 | WDTCR | Watchdog Timer Control Register |
| \$1E | EEAR | EEPROM Address Register |
| \$1D | EEDR | EEPROM Data Register |
| \$1C | EECR | EEPROM Control Register |
| \$18 | PORTB | Data Register, Port B |
| \$17 | DDRB | Data Direction Register, Port B |
| \$16 | PINB | Input Pins, Port B |
| \$12 | PORTD | Data Register, Port D |
| \$11 | DDRD | Data Direction Register, Port D |
| \$10 | PIND | Input Pins, Port D |
| \$08 | ACSR | Analog Comparator Control and Status Register |

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

All AT90S1200 I/Os and peripherals are placed in the I/O space. The different I/O locations are accessed by the IN and OUT instructions transferring data between the 32 general purpose working registers and the I/O space. I/O registers within the address range \$00 - \$1F are directly bit-accessible using the SBI and CBI instructions. In these registers, the value of single bits can be checked by using the SBIS and SBIC instructions. Refer to the instruction set chapter for more details.

For compatibility with future devices, reserved bits should be written to zero if 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 different 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 is defined as:

| | | | | | | | | | |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| \$3F | 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 bit 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 arithmetic operations. See the Instruction Set description for detailed information.

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

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

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

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

- **Bit 2 – N: Negative Flag**

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

- **Bit 1 – Z: Zero Flag**

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

- **Bit 0 – C: Carry Flag**

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

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

Reset and Interrupt Handling

The AT90S1200 provides three different interrupt sources. These interrupts and the separate reset vector, each have a separate program vector in the program memory space. All the interrupts are assigned individual enable bits that must be set (one) together with the I-bit in the Status Register in order to enable the interrupt.

The lowest addresses in the program memory space are automatically defined as the Reset and Interrupt vectors. The complete list of vectors is shown in Table 2. The list also determines the priority levels of the different interrupts. The lower the address the higher is the priority level. RESET has the highest priority, and next is INTO (the External Interrupt Request 0), etc.

Table 2. Reset and Interrupt Vectors

| Vector No. | Program Address | Source | Interrupt Definition |
|------------|-----------------|--------------|---|
| 1 | \$000 | RESET | Hardware Pin, Power-on Reset and Watchdog Reset |
| 2 | \$001 | INT0 | External Interrupt Request 0 |
| 4 | \$002 | TIMER0, OVFO | Timer/Counter0 Overflow |
| 5 | \$003 | ANA_COMP | Analog Comparator |

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

```

Address  Labels  Code           Comments
$000           rjmp    RESET      ; Reset Handler
$001           rjmp    EXT_INT0   ; IRQ0 Handler
$002           rjmp    TIM0_OVF   ; Timer0 Overflow Handler
$003           rjmp    ANA_COMP   ; Analog Comparator Handler
;
$004  MAIN:    <instr> xxx      ; Main program start
...           ...           ...           ...

```

Reset Sources

The AT90S1200 has three sources of reset:

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

During Reset, all I/O registers are then set to their initial values, and the program starts execution from address \$000. The instruction placed in address \$000 must be an RJMP (relative jump) instruction to the reset handling routine. If the program never enables an interrupt source, the interrupt vectors are not used, and regular program code can be placed at these locations. The circuit diagram in Figure 13 shows the reset logic. Table 3 defines the timing and electrical parameters of the reset circuitry. Note that Power-on Reset timing is clocked by the internal RC Oscillator. Refer to characterization data for RC Oscillator frequency at other V_{CC} voltages.

Figure 13. Reset Logic

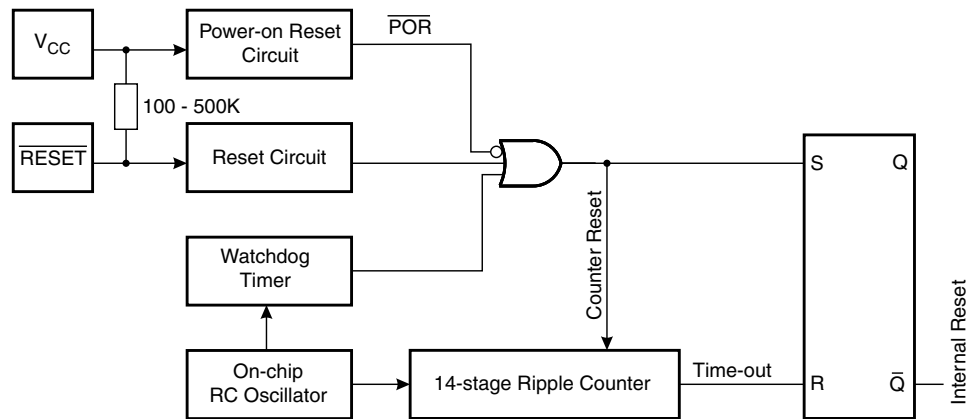


Table 3. Reset Characteristics (V_{CC} = 5.0V)

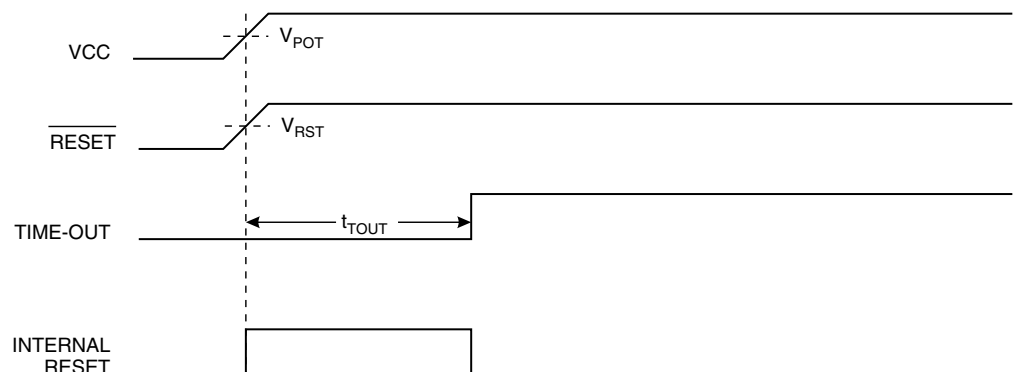
| Symbol | Parameter | Min | Typ | Max | Units |
|---------------------------------|---|------|------|----------------------|-------|
| V _{POT} ⁽¹⁾ | Power-on Reset Threshold Voltage (rising) | 0.8 | 1.2 | 1.6 | V |
| | Power-on Reset Threshold Voltage (falling) | 0.2 | 0.4 | 0.6 | V |
| V _{RST} | Pin Threshold Voltage | – | – | 0.85 V _{CC} | V |
| t _{POR} | Power-on Reset Period | 2.0 | 3.0 | 4.0 | ms |
| t _{TOUT} | Reset Delay Time-out Period (The Time-out period equals 16K WDT cycles. See “Typical Characteristics” on page 51. for typical WDT frequency at different voltages). | 11.0 | 16.0 | 21.0 | ms |

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

Power-on Reset

A Power-on Reset (POR) circuit ensures that the device is reset from power-on. As shown in Figure 13, an internal timer clocked from the Watchdog timer oscillator prevents the MCU from starting until after a certain period after V_{CC} has reached the Power-on Threshold voltage (V_{POT}), regardless of the V_{CC} rise time (see Figure 14).

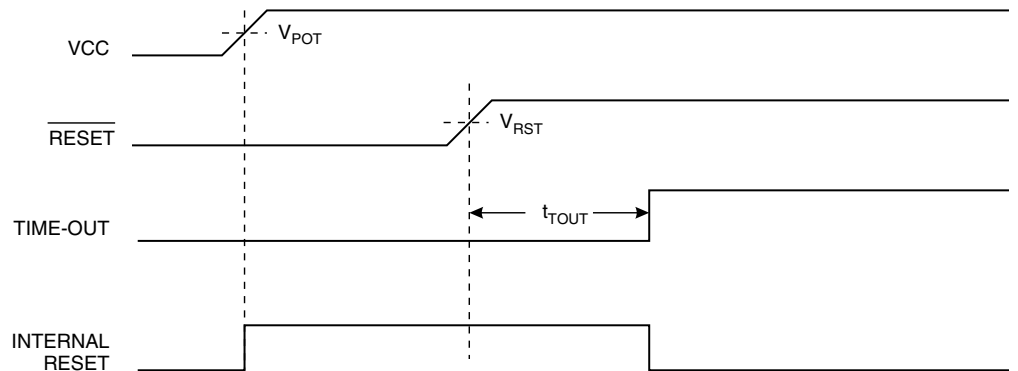
Figure 14. MCU Start-up, $\overline{\text{RESET}}$ Tied to V_{CC}.



If the built-in start-up delay is sufficient, $\overline{\text{RESET}}$ can be connected to V_{CC} directly or via an external pull-up resistor. By holding the $\overline{\text{RESET}}$ pin low for a period after V_{CC} has

been applied, the Power-on Reset period can be extended. Refer to Figure 15 for a timing example on this.

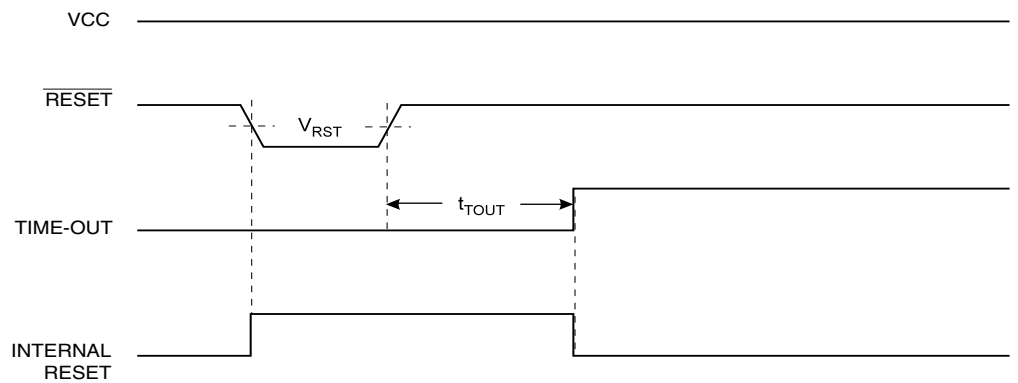
Figure 15. MCU Start-up, $\overline{\text{RESET}}$ Controlled Externally



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.

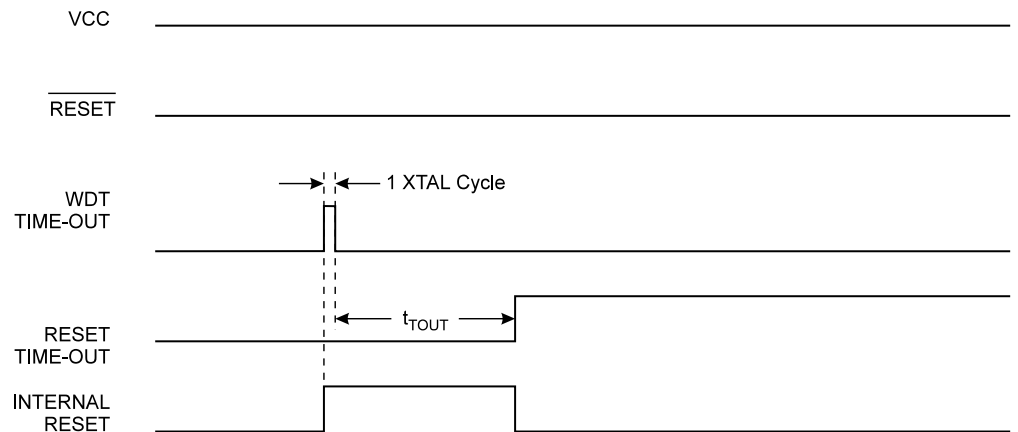
Figure 16. External Reset during Operation



Watchdog Reset

When the Watchdog times out, it will generate a short reset pulse of 1 XTAL cycle duration. On the falling edge of this pulse, the delay timer starts counting the Time-out period t_{TOUT} . Refer to page 23 for details on operation of the Watchdog.

Figure 17. Watchdog Reset during Operation



Interrupt Handling

The AT90S1200 has two Interrupt Mask Control Registers: the GIMSK (General Interrupt Mask Register) at I/O space address \$3B and the TIMSK (Timer/Counter Interrupt Mask Register) at I/O address \$39.

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 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 | - | INT0 | - | - | - | - | - | - | GIMSK |
| Read/Write | R | R/W | R | R | R | R | R | R | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

- Bit 7 – Res: Reserved Bit**

This bit is a reserved bit in the AT90S1200 and always reads as zero.

- **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 bit 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 low level sensed. INT0 can be activated even if the pin is configured as an output. See also page 17.

- **Bits 5..0 – Res: Reserved Bits**

These bits are reserved bits in the AT90S1200 and always read as zero.

Timer/Counter Interrupt Mask Register – TIMSK

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---------------|---|---|---|---|---|---|-------|---|-------|
| \$39 | - | - | - | - | - | - | TOIE0 | - | TIMSK |
| Read/Write | R | R | R | R | R | R | R/W | R | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

- **Bits 7..2 – Res: Reserved Bits**

These bits are reserved bits in the AT90S1200 and always read as zero.

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

When the TOIE0 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter0 Overflow interrupt is enabled. The corresponding interrupt (at vector \$002) 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 AT90S1200 and always reads as zero.

Timer/Counter Interrupt FLAG Register – TIFR

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---------------|---|---|---|---|---|---|------|---|------|
| \$38 | - | - | - | - | - | - | TOV0 | - | TIFR |
| Read/Write | R | R | R | R | R | R | R/W | R | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

- **Bits 7..2 – Res: Reserved Bits**

These bits are reserved bits in the AT90S1200 and always read as zero.

- **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 AT90S1200 and always reads as zero.

External Interrupts

The External Interrupt is triggered by the INT0 pin. The interrupt can trigger on rising edge, falling edge or low level. This is set up as described in the specification for the MCU Control Register (MCUCR). When INT0 is level triggered, the interrupt is pending as long as INT0 is held low.

The interrupt is triggered even if INT0 is configured as an output. This provides a way to generate a software interrupt.

The interrupt flag can not be directly accessed by the user. If an external edge-triggered interrupt is suspected to be pending, the flag can be cleared as follows.

1. Disable the External Interrupt by clearing the INT0 flag in GIMSK.
2. Select level triggered interrupt.
3. Select desired interrupt edge.
4. Re-enable the external interrupt by setting INT0 in GIMSK.

Interrupt Response Time

The interrupt execution response for all the enabled AVR interrupts is four clock cycles minimum. Four 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 (9 bits) is pushed onto the Stack. The vector is normally a relative jump to the interrupt routine, and this jump takes two 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 takes four clock cycles. During these four clock cycles, the Program Counter (9 bits) is popped back from the Stack 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.

Note that the Subroutine and Interrupt Stack is a 3-level true hardware stack, and if more than three nested subroutines and interrupts are executed, only the most recent three return addresses are stored.

MCU Control Register – MCUCR

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

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---------------|---|---|-----|-----|---|---|-------|-------|-------|
| \$35 | – | – | SE | SM | – | – | ISC01 | ISC00 | MCUCR |
| Read/Write | R | R | R/W | R/W | R | R | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

- **Bits 7, 6 – Res: Reserved Bits**

These bits are reserved bits in the AT90S1200 and always read 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.

- **Bit 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” on the following page.

- **Bits 3, 2 – Res: Reserved Bits**

These bits are reserved bits in the AT90S1200 and always read as zero.

- **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 in the GIMSK register is set. The level and edges on the external INT0 pin that activate the interrupt are defined in Table 4.

Table 4. Interrupt 0 Sense Control

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

The value on the INT0 pin is sampled before detecting edges. If edge interrupt is selected, pulses with a duration longer than one CPU 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. If enabled, a level triggered interrupt will generate an interrupt request as long as the pin is held low.

Sleep Modes

To enter the sleep modes, the SE bit in MCUCR must be set (one) and a SLEEP instruction must be executed. 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. The contents of the register file and the I/O memory are unaltered. If a Reset occurs during sleep mode, the MCU wakes up and executes from the Reset Vector.

Idle Mode

When the SM bit is cleared (zero), the SLEEP instruction makes the MCU enter 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 wakeup from the Analog Comparator interrupt is not required, the Analog Comparator can be powered down by setting the ACD-bit in the Analog Comparator Control and Status Register (ACSR). This will reduce power consumption in Idle mode. When the MCU wakes up from Idle mode, the CPU starts program execution immediately.

Power-down Mode

When the SM bit is set (one), the SLEEP instruction makes the MCU enter 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 on INT0 can wake up the MCU.

Note that when 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.

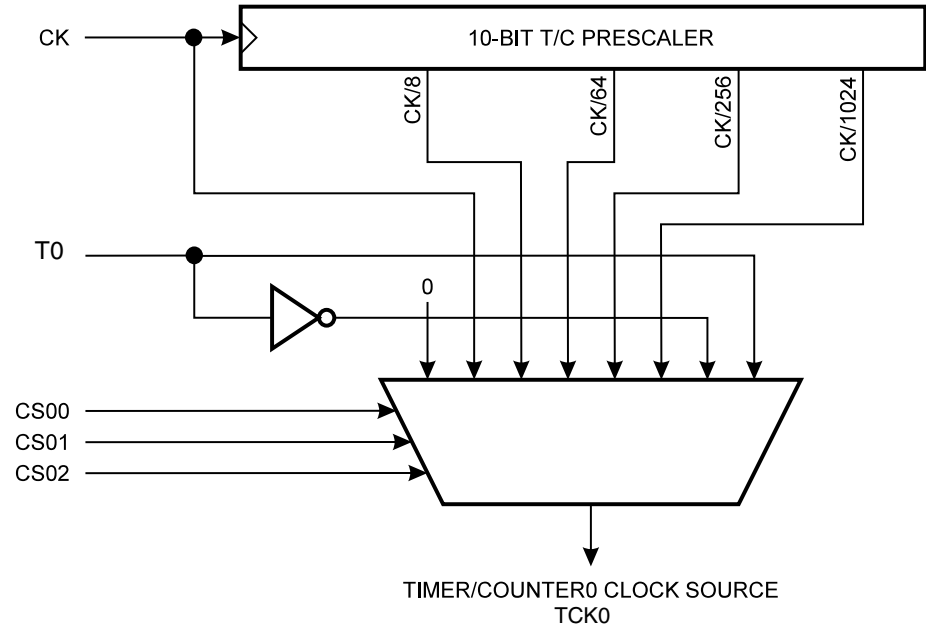
Timer/Counter0

The AT90S1200 provides one general purpose 8-bit Timer/Counter. The Timer/Counter0 gets the prescaled clock from the 10-bit prescaling timer. The Timer/Counter0 can either be used as a Timer with an internal clock time base or as a Counter with an external pin connection, which triggers the counting.

Timer/Counter0 Prescaler

Figure 18 shows the general Timer/Counter0 prescaler.

Figure 18. Timer/Counter0 Prescaler



The four different prescaled selections are: CK/8, CK/64, CK/256, and CK/1024 where CK is the Oscillator Clock. For the Timer/Counter0, added selections as CK, external clock source and stop, can be selected as clock sources. Figure 19 shows the block diagram for Timer/Counter0.

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

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

Table 5. Clock 0 Prescale Select

| CS02 | CS01 | CS00 | Description |
|------|------|------|--------------------------------------|
| 0 | 0 | 0 | Stop, the Timer/Counter0 is stopped. |
| 0 | 0 | 1 | CK |
| 0 | 1 | 0 | CK/8 |
| 0 | 1 | 1 | CK/64 |
| 1 | 0 | 0 | CK/256 |
| 1 | 0 | 1 | CK/1024 |
| 1 | 1 | 0 | External Pin T0, falling edge |
| 1 | 1 | 1 | External Pin T0, rising edge |

The Stop condition provides a Timer Enable/Disable function. The CK down divided modes are scaled directly from the CK Oscillator clock. If the external pin modes are used 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/Counter0 – TCNT0

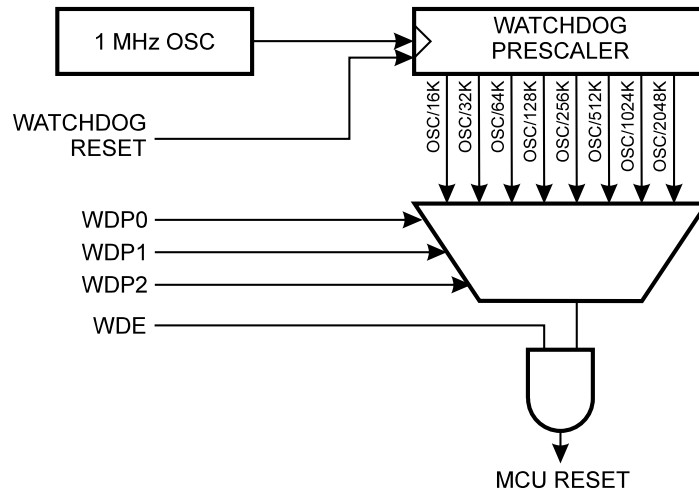
| | | | | | | | | | |
|---------------|--|-----|-----|-----|-----|-----|-----|-----|-------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| \$32 | <div style="display: flex; justify-content: space-between; align-items: center;"> MSB LSB </div> | | | | | | | | TCNT0 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

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 timer clock cycle following the write operation.

Watchdog Timer

The Watchdog Timer is clocked from a separate On-chip Oscillator that runs at 1 MHz. This is the typical value at $V_{CC} = 5V$. See characterization data for typical values at other V_{CC} levels. By controlling the Watchdog Timer prescaler, the Watchdog Reset interval can be adjusted, see Table 6 for a detailed description. The WDR (Watchdog Reset) instruction resets the Watchdog Timer. Eight different clock cycle periods can be selected to determine the maximum period between two WDR instructions to prevent the Watchdog Timer from resetting the MCU. If the reset period expires without another WDR instruction, the AT90S1200 resets and executes from the Reset Vector. For timing details on the Watchdog Reset, refer to page 14.

Figure 20. Watchdog Timer



Watchdog Timer Control Register – WDTCR

| | | | | | | | | | |
|---------------|---|---|---|---|-----|------|------|------|-------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| \$21 | – | – | – | – | WDE | WDP2 | WDP1 | WDP0 | WDTCR |
| Read/Write | R | R | R | R | R/W | R/W | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

- **Bits 7..4 – Res: Reserved Bits**

These bits are reserved bits in the AT90S1200 and will always read as zero.

- **Bit 3 – WDE: Watchdog 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.

- **Bits 2..0 – WDP2..0: Watchdog Timer Prescaler 2, 1 and 0**

The WDP2..0 determine the Watchdog Timer prescaling when the Watchdog Timer is enabled. The different prescaling values and their corresponding timeout periods are shown in Table 6.

Table 6. Watchdog 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 “Typical Characteristics” on page 51.

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 to count from zero.

To avoid unintentional MCU resets, the Watchdog Timer should be disabled or reset before changing the Watchdog Timer Prescale Select.

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 - 4 ms, depending on the V_{CC} voltages. A self-timing function, however, lets the user software detect when the next byte can be written. If the user code contains code that writes the EEPROM, some precaution must be taken. In heavily filtered power supplies, V_{CC} is likely to rise or fall slowly on Power-up/down. This causes the device for some period of time to run at a voltage lower than specified as minimum for the clock frequency used. CPU operation under these conditions is likely cause the program counter to perform unintentional jumps and eventually execute the EEPROM write code. To secure EEPROM integrity, the user is advised to use an external under-voltage reset circuit in this case.

In order to prevent unintentional EEPROM writes, a specific write procedure must be followed. Refer to “EEPROM Control Register – EECR” on page 25 for details on this.

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

EEPROM Address Register – EEAR

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---------------|---|---|-------|-------|-------|-------|-------|-------|------|
| \$1E | – | – | EEAR5 | EEAR4 | EEAR3 | EEAR2 | EEAR1 | EEAR0 | EEAR |
| Read/Write | R | R | R/W | R/W | R/W | R/W | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

- **Bit 7, 6 – Res: Reserved Bits**

These bits are reserved bits in the AT90S1200 and will always read as zero.

- **Bits 5..0 – EEAR5..0: EEPROM Address**

The EEPROM Address Register (EEAR5..0) specifies the EEPROM address in the 64-byte EEPROM space. The EEPROM data bytes are addressed linearly between 0 and 63.

EEPROM Data Register – EEDR

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|------|
| \$1D | 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 | – | – | – | – | – | – | EEWE | EERE | EECR |
| Read/Write | R | R | R | R | R | R | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

- **Bits 7..2 – Res: Reserved Bits**

These bits are reserved bits in the AT90S1200 and will always be read as zero.

- **Bit 1 – EEW: 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. When the write access time (typically 2.5 ms at $V_{CC} = 5V$ and 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.

Caution: If an interrupt routine accessing the EEPROM is interrupting another EEPROM access, the EEAR or EEDR register will be modified, causing the interrupted EEPROM access to fail. It is recommended to have the global interrupt flag cleared during EEPROM write operation to avoid these problems.

Prevent EEPROM Corruption

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

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

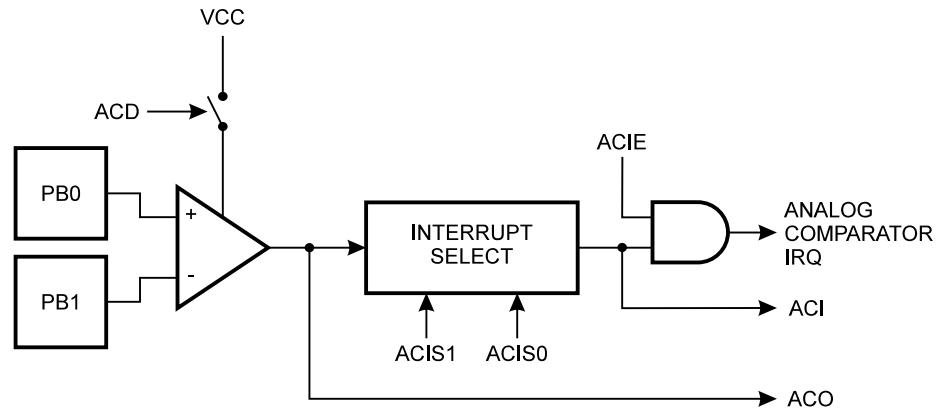
EEPROM data corruption can easily be avoided by following these design recommendations (one is sufficient):

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

Analog Comparator

The Analog Comparator compares the input values on the positive input PB0 (AIN0) and the negative input PB1 (AIN1). When the voltage on the positive input PB0 (AIN0) is higher than the voltage on the negative input PB1 (AIN1), the Analog Comparator Output (ACO) is set (one). The comparator's output can be set to trigger the Analog Comparator interrupt. 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 21.

Figure 21. Analog Comparator Block Diagram



Analog Comparator Control and Status Register – ACSR

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---------------|-----|---|-----|-----|------|---|-------|-------|------|
| \$08 | ACD | – | ACO | ACI | ACIE | – | ACIS1 | ACIS0 | ACSR |
| Read/Write | R/W | R | R | R/W | R/W | R | R/W | R/W | |
| Initial Value | 0 | 0 | N/A | 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. This will reduce power consumption in Active and Idle modes. When changing the ACD bit, the Analog Comparator Interrupt must be disabled by clearing the ACIE bit in ACSR. Otherwise, an interrupt can occur when the bit is changed.

- **Bit 6 – Res: Reserved Bit**

This bit is a reserved bit in the AT90S1200 and will always read as zero.

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

ACO is directly connected to the comparator output.

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

This bit is set (one) when a comparator output event triggers the interrupt mode defined by ACIS1 and ACIS0. 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. Observe however, that if another bit in this register is modified using the SBI or CBI instruction, ACI will be cleared if it has become set before the operation.

- **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 – Res: Reserved Bit**

This bit is a reserved bit in the AT90S1200 and will always read as zero.

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

These bits determine which comparator events trigger the Analog Comparator Interrupt. The different settings are shown in Table 7.

Table 7. 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.

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 an 8-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), Data Direction Register – DDRB (\$17), and the Port B Input Pins – PINB (\$16). 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 20 mA 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 Table 8.

Table 8. Port B Pin Alternate Functions

| Port Pin | Alternate Functions |
|----------|---|
| PB0 | AIN0 (Analog Comparator positive input) |
| PB1 | AIN1 (Analog Comparator negative input) |
| PB5 | MOSI (Data Input line for memory downloading) |
| PB6 | MISO (Data Output line for memory uploading) |
| PB7 | SCK (Serial Clock input) |

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 | PORTB7 PORTB6 PORTB5 PORTB4 PORTB3 PORTB2 PORTB1 PORTB0 | | | | | | | | PORTB |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

Port B Data Direction Register – DDRB

| | | | | | | | | | |
|---------------|--|-----|-----|-----|-----|-----|-----|-----|-------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| \$17 | DDB7 DDB6 DDB5 DDB4 DDB3 DDB2 DDB1 DDB0 | | | | | | | | DDR B |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

Port B Input Pin Address – PINB

| | | | | | | | | | |
|---------------|--|-----|-----|-----|-----|-----|-----|-----|------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| \$16 | PINB7 PINB6 PINB5 PINB4 PINB3 PINB2 PINB1 PINB0 | | | | | | | | PINB |
| Read/Write | R | R | R | R | R | R | R | R | |
| Initial Value | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | |

The Port 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 eight 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) and the pin is configured as an input pin, the MOS pull-up resistor is activated. To switch the pull-up resistor off, PORTBn has to be cleared (zero) or the pin has to be configured as an output pin. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not active.

Table 9. DDBn Effect on Port B Pins

| DDBn | PORTBn | I/O | Pull-up | Comment |
|------|--------|--------|---------|---|
| 0 | 0 | Input | No | Tri-state (High-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: 7,6...0, pin number.

Alternate Functions of Port B

The alternate pin functions of Port B are:

- **SCK – Port B, Bit 7**

SCK, Clock Input pin for memory up/downloading.

- **MISO – Port B, Bit 6**

MISO, Data Output pin for memory uploading.

- **MOSI – Port B, Bit 5**

MOSI, Data Input pin for memory downloading.

- **AIN1 – Port B, Bit 1**

AIN1, Analog Comparator Negative Input. When configured as an input (DDB1 is cleared [zero]) and with the internal MOS pull-up resistor switched off (PB1 is cleared [zero]), this pin also serves as the negative input of the On-chip Analog Comparator.

- **AIN0 – Port B, Bit 0**

AIN0, Analog Comparator Positive Input. When configured as an input (DDB0 is cleared [zero]) and with the internal MOS pull-up resistor switched off (PB0 is cleared [zero]), this pin also serves as the positive input of the On-chip Analog Comparator.

Port B Schematics

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

Figure 22. Port B Schematic Diagram (Pins PB0 and PB1)

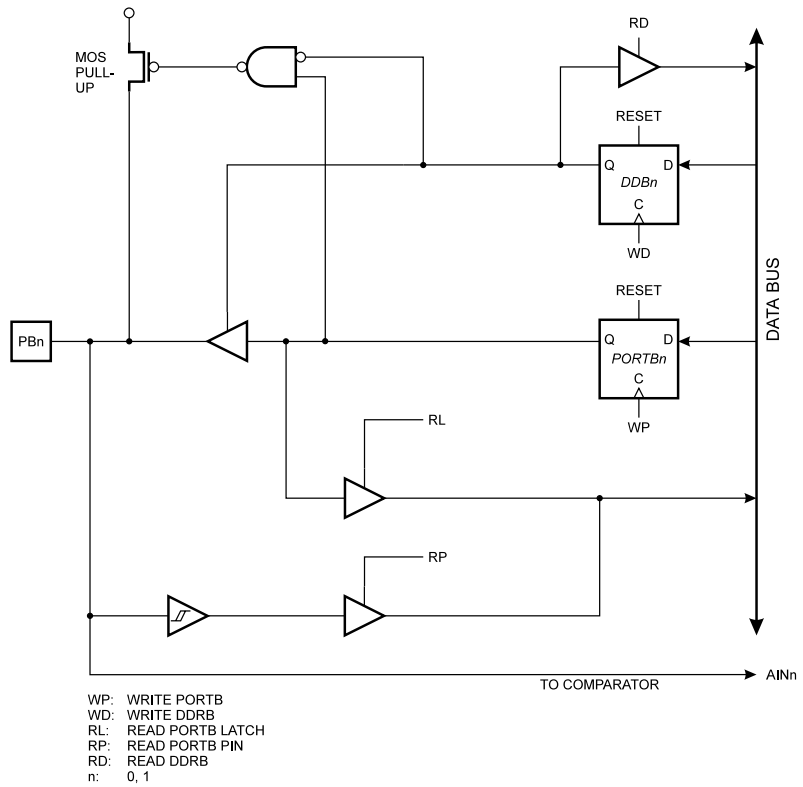


Figure 23. Port B Schematic Diagram (Pins PB2, PB3, and PB4)

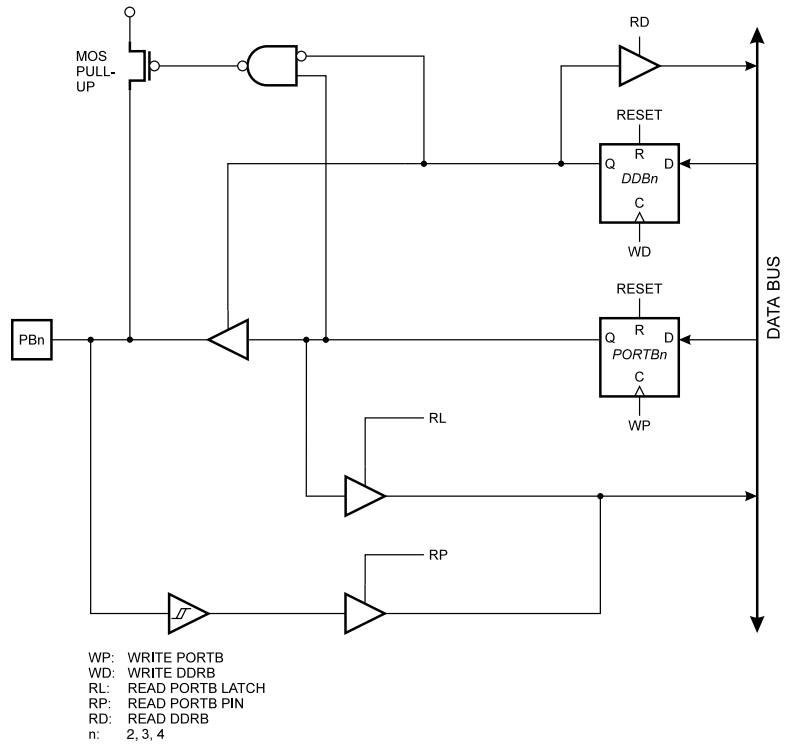


Figure 24. Port B Schematic Diagram (Pin PB5)

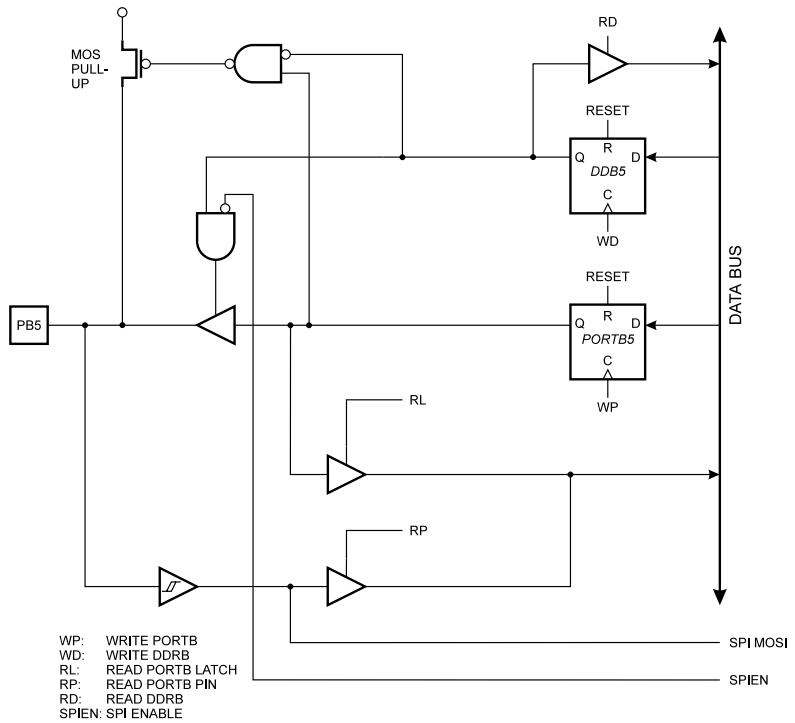


Figure 25. Port B Schematic Diagram (Pin PB6)

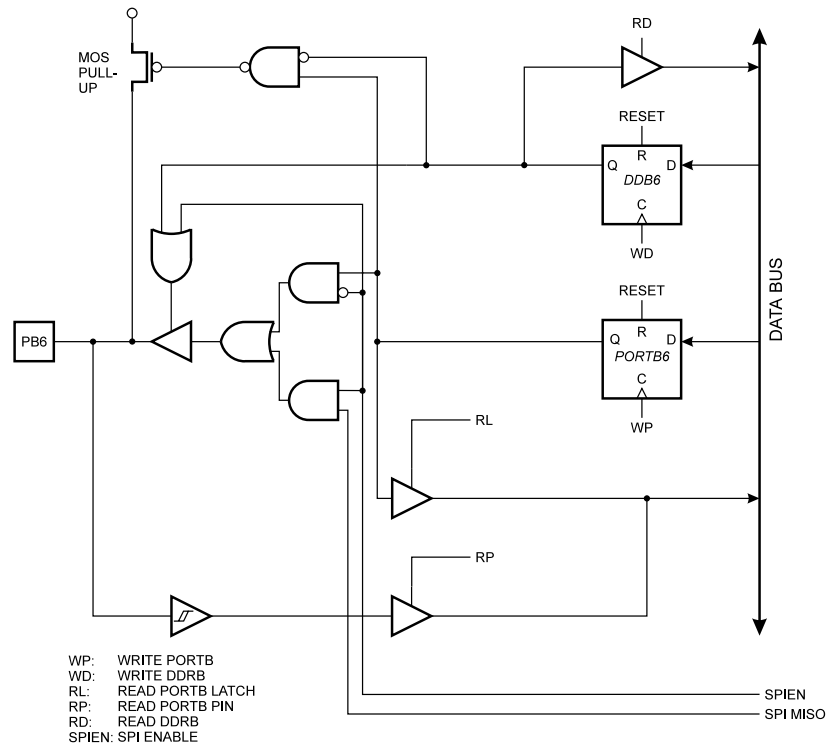
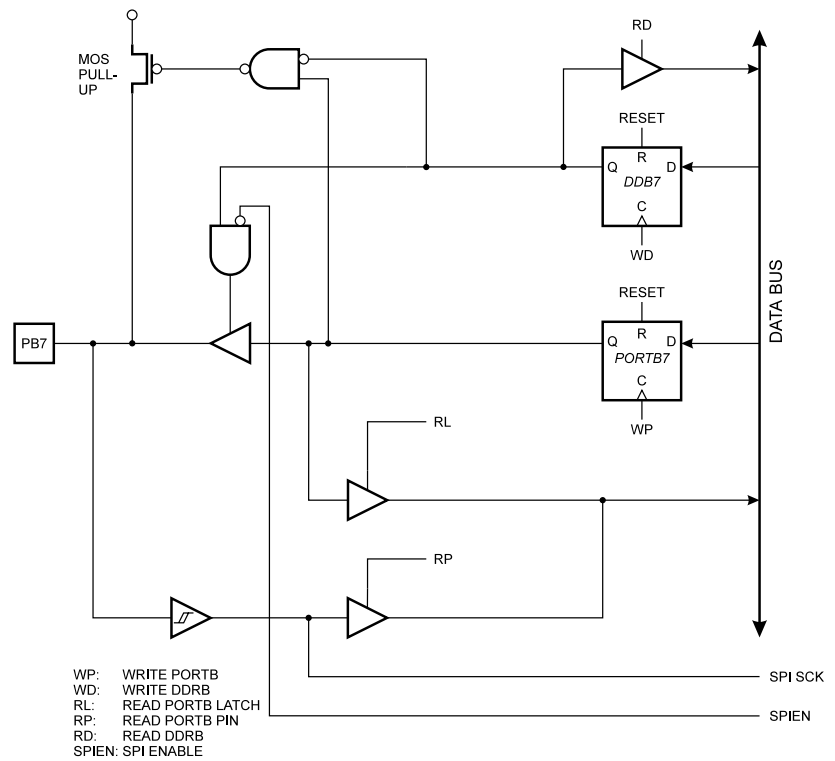


Figure 26. Port B Schematic Diagram (Pin PB7)



Port D

Three I/O memory address locations are allocated for Port D, one each for the Data Register – PORTD (\$12), Data Direction Register – DDRD (\$11), and the Port D Input Pins – PIND (\$10). The Port D Input Pins address is read-only, while the Data Register and the Data Direction Register are read/write.

Port D has seven bi-directional I/O pins with internal pull-up resistors, PD6..PD0. 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 Table 10.

Table 10. Port D Pin Alternate Functions

| Port Pin | Alternate Function |
|----------|-------------------------------------|
| PD2 | INT0 (External Interrupt 0 input) |
| PD4 | T0 (Timer/Counter 0 external input) |

Port D Data Register – PORTD

| | | | | | | | | | |
|---------------|---|--------|--------|--------|--------|--------|--------|--------|-------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| \$12 | – | PORTD6 | PORTD5 | PORTD4 | PORTD3 | PORTD2 | PORTD1 | PORTD0 | PORTD |
| Read/Write | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

Port D Data Direction Register – DDRD

| | | | | | | | | | |
|---------------|---|------|------|------|------|------|------|------|------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| \$11 | – | DDD6 | DDD5 | DDD4 | DDD3 | DDD2 | DDD1 | DDD0 | DDRD |
| Read/Write | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

Port D Input Pins Address – PIND

| | | | | | | | | | |
|---------------|---|-------|-------|-------|-------|-------|-------|-------|------|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| \$10 | – | PIND6 | PIND5 | PIND4 | PIND3 | PIND2 | PIND1 | PIND0 | PIND |
| Read/Write | R | R | R | R | R | R | R | R | |
| Initial Value | 0 | N/A | N/A | N/A | N/A | N/A | N/A | N/A | |

The Port D Input Pins address (PIND) is not a register, 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

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

Figure 28. Port D Schematic Diagram (Pin PD2)

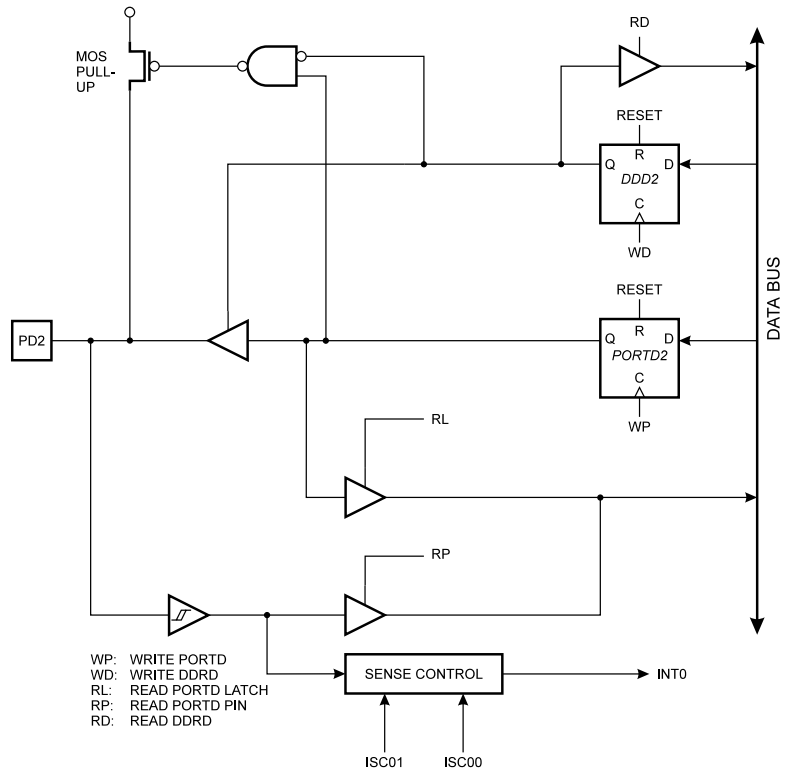
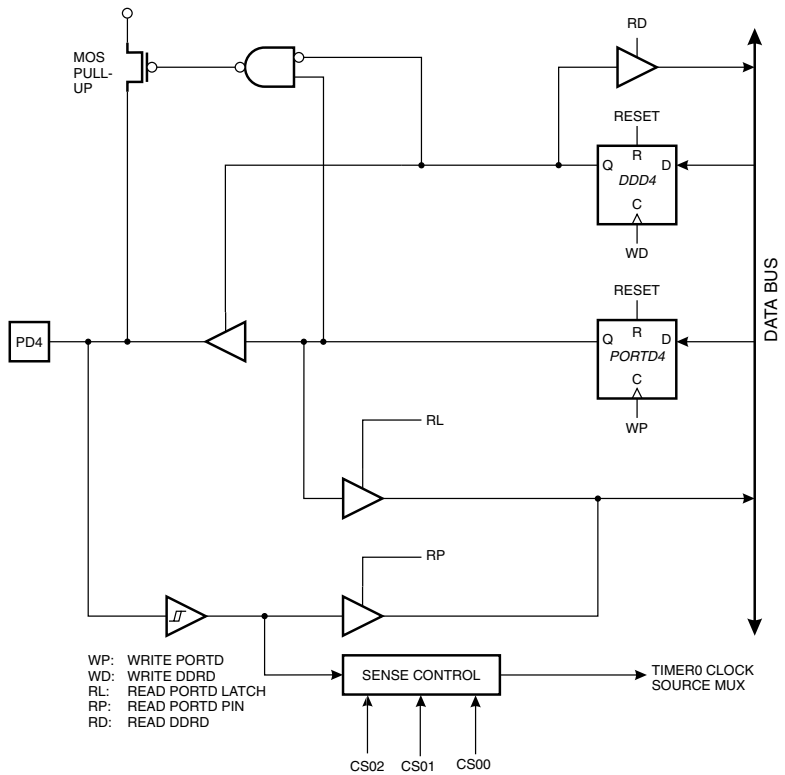


Figure 29. Port D Schematic Diagram (Pin PD4)



Memory Programming

Program and Data Memory Lock Bits

The AT90S1200 MCU provides two Lock bits that can be left unprogrammed (“1”) or can be programmed (“0”) to obtain the additional features listed in Table 12. The Lock bits can only be erased with the Chip Erase command.

Table 12. 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, further programming of the Fuse bits are also disabled. Program the Fuse bits before programming the Lock bits.

Fuse Bits

The AT90S1200 has two Fuse bits: SPIEN and RCEN.

- When the SPIEN Fuse bit is programmed (“0”), Serial Program Downloading is enabled. Default value is programmed (“0”).
- When the RCEN Fuse bit is programmed (“0”), MCU clocking from the Internal RC Oscillator is selected. Default value is erased (“1”). Parts with this bit pre-programmed (“0”) can be delivered on demand.
- The Fuse bits are not accessible in Serial Programming mode. The status of the Fuse bits is not affected by Chip Erase.

Signature Bytes

All Atmel microcontrollers have a 3-byte signature code that identifies the device. This code can be read in both Serial and Parallel modes. The three bytes reside in a separate address space.

For the AT90S1200 they are:

1. \$00: \$1E (indicates manufactured by Atmel)
2. \$01: \$90 (indicates 1 Kb Flash memory)
3. \$02: \$01 (indicates AT90S1200 device when \$01 is \$90)

Note: When both Lock bits are programmed (lock mode 3), the signature bytes cannot be read in Serial mode. Reading the signature bytes will return: \$00, \$01 and \$02.

Programming the Flash and EEPROM

Atmel’s AT90S1200 offers 1K byte of in-System Reprogrammable Flash program memory and 64 bytes of EEPROM data memory.

The AT90S1200 is normally shipped with the On-chip Flash program memory 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 AT90S1200 inside the user’s system.

The program and data memory arrays on the AT90S1200 are programmed byte-by-byte in either programming mode. 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 13.

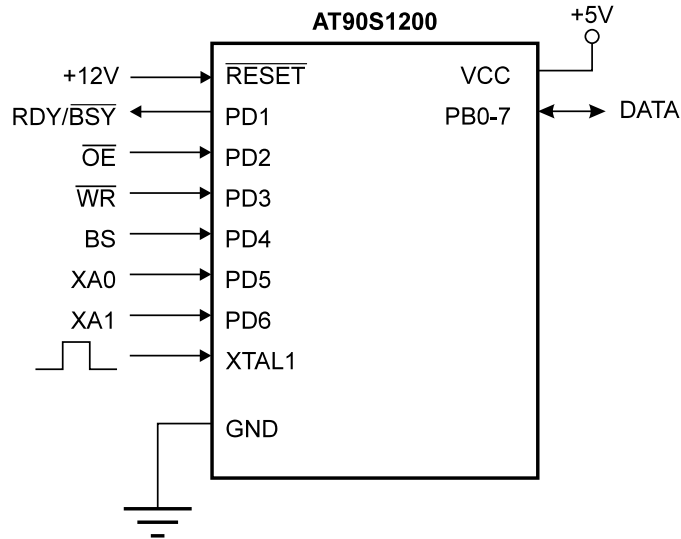
Table 13. Supply Voltage during Programming

| Part | Serial Programming | Parallel Programming |
|-----------|--------------------|----------------------|
| AT90S1200 | 2.7 - 6.0V | 4.5 - 5.5V |

Parallel Programming

This section describes how to parallel program and verify Flash program memory, EEPROM data memory, Lock bits and Fuse bits in the AT90S1200.

Figure 30. Parallel Programming



Signal Names

In this section, some pins of the AT90S1200 are referenced by signal names describing their function during parallel programming rather than their pin names, see Figure 30 and Table 14. Pins not described in Table 14 are referenced by pin names.

The XA1/XA0 pins determines the action executed when the XTAL1 pin is given a positive pulse. The coding is shown in Table 15.

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 16.

Table 14. Pin Name Mapping

| Signal Name in Programming Mode | Pin Name | I/O | Function |
|---------------------------------|----------|-----|---|
| RDY/BSY | PD1 | O | 0: Device is busy programming, 1: Device is ready for new command |
| \overline{OE} | PD2 | I | Output Enable (Active low) |
| \overline{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 | PB0-7 | I/O | Bi-directional Data Bus (Output when \overline{OE} is low) |

Table 15. XA1 and XA0 Coding

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

Table 16. Command Byte 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 13, between V_{CC} and GND.
2. Set the \overline{RESET} and BS pin to “0” and wait at least 100 ns.
3. Apply 11.5 - 12.5V to \overline{RESET} . Any activity on BS within 100 ns after +12V has been applied to \overline{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, t_{WLWH_CE} is found in Table 17. Chip Erase does not generate any activity on the RDY/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 - \$01).
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 31 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 32 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 apply to EEPROM programming and Flash, EEPROM and signature byte reading.

Figure 31. Programming the Flash Waveforms

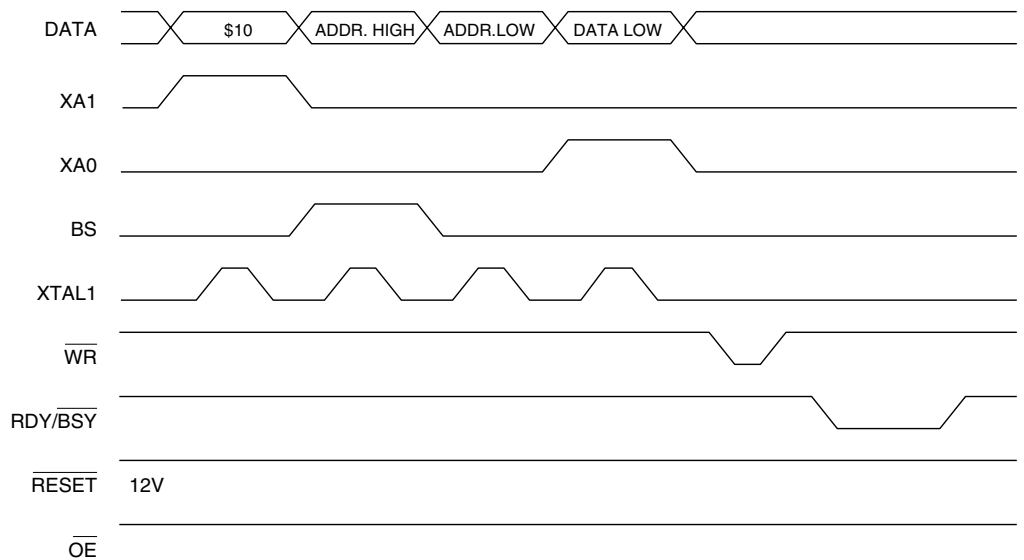
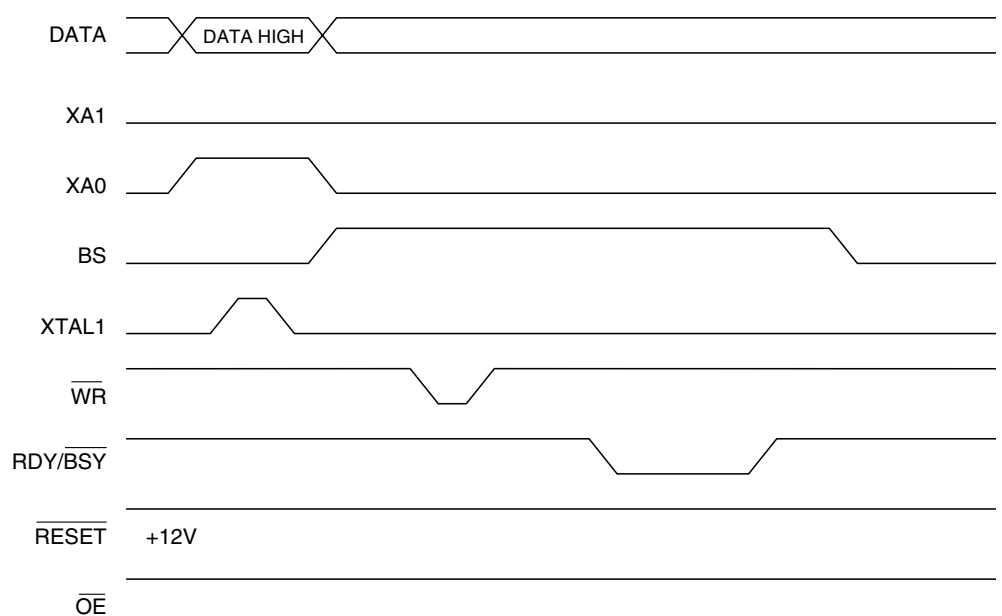


Figure 32. Programming the Flash Waveforms (Continued)



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 - \$01).
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 - \$3F).
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 - \$3F).
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 0 = RCEN Fuse
 Bit 7 - 6, 4 - 1 = “1”. These bits are reserved and should be left unprogrammed (“1”).
3. Give \overline{WR} a t_{WLWH_PFB} wide negative pulse to execute the programming; t_{WLWH_PFB} is found in Table 17. 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” on page 39 for details on command loading):

1. A: Load Command “0000 0100”.
2. Set \overline{OE} to “0”, and BS to “1”. The status of Fuse and Lock bits can now be read at DATA (“0” means programmed).
 Bit 7 = Lock Bit1
 Bit 6 = Lock Bit2
 Bit 5 = SPIEN Fuse
 Bit 0 = RCEN Fuse
3. Set \overline{OE} to “1”.

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

Reading the Signature Bytes

The algorithm for reading the signature bytes is as follows (refer to “Programming the Flash” on page 39 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.
Set \overline{OE} to “1”.

Parallel Programming Characteristics

Figure 33. Parallel Programming Timing

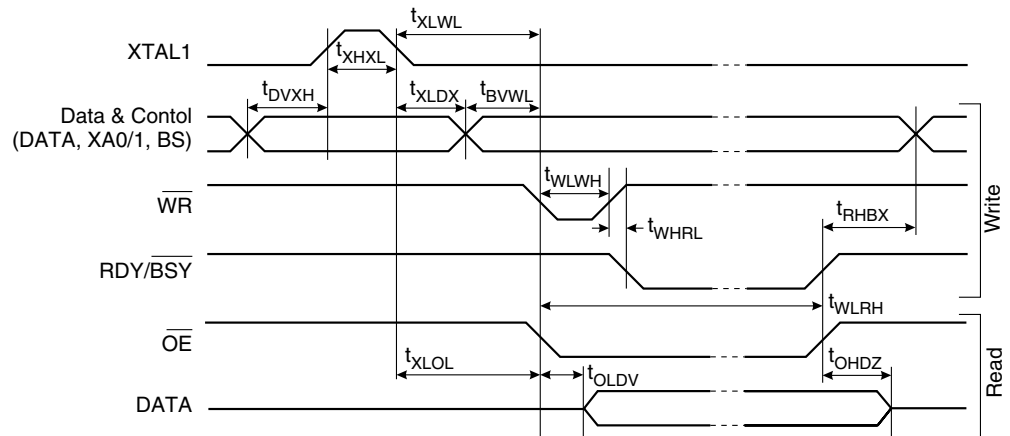


Table 17. Parallel Programming Characteristics, $T_A = 25^\circ\text{C} \pm 10\%$, $V_{CC} = 5\text{V} \pm 10\%$

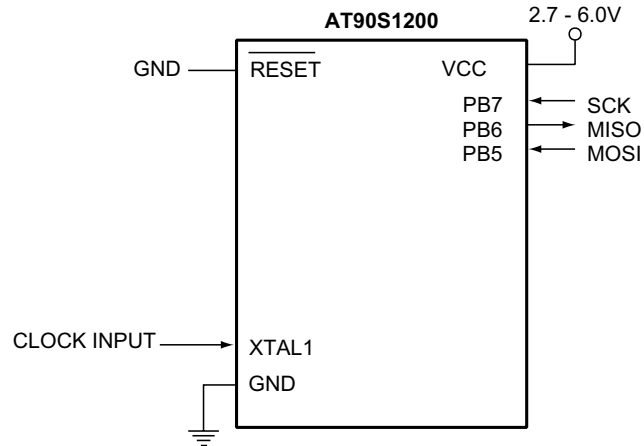
| Symbol | Parameter | Min | Typ | Max | Units |
|-----------------|--|------|------|-------|---------------|
| V_{PP} | Programming Enable Voltage | 11.5 | | 12.5 | V |
| I_{PP} | Programming Enable Current | | | 250.0 | μA |
| t_{DVXH} | Data and Control Setup before XTAL1 High | 67.0 | | | ns |
|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

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

Serial Downloading

Both the program and data memory arrays can be programmed using the SPI bus while $\overline{\text{RESET}}$ is pulled to GND. The serial interface consists of pins SCK, MOSI (input) and MISO (output) (see Figure 34). After $\overline{\text{RESET}}$ is set low, the Programming Enable instruction needs to be executed first before program/erase instructions can be executed.

Figure 34. Serial Programming and Verify



Note: If the device is clocked by the Internal Oscillator, it is no need to connect a clock source to the XTAL1 pin

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 \$01FF for Flash program memory and \$000 to \$03F for EEPROM data 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: > 1 XTAL1 clock cycle

High: > 4 XTAL1 clock cycles

Serial Programming Algorithm

When writing serial data to the AT90S1200, data is clocked on the rising edge of SCK.

When reading data from the AT90S1200, data is clocked on the falling edge of SCK. See Figure 35 and Table 20 for timing details.

To program and verify the AT90S1200 in the Serial Programming mode, the following sequence is recommended (See 4-byte instruction formats in Table 17):

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 or the device is not running from the Internal RC Oscillator, apply a clock signal to the XTAL1 pin. If the programmer can not guarantee that SCK is held low during power-up, $\overline{\text{RESET}}$ must be given a positive pulse 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 the MOSI (PB5) pin.

3. If a Chip Erase is performed (must be done to erase the Flash), wait t_{WD_ERASE} after the instruction, give \overline{RESET} a positive pulse, and start over from step 2. See Table 21 on page 47 for t_{WD_ERASE} value.
4. 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. Wait t_{WD_PROG} after transmitting the instruction. In an erased device, no \$FFs in the data file(s) needs to be programmed. See Table 22 on page 47 for t_{WD_PROG} value.
5. Any memory location can be verified by using the Read instruction which returns the content at the selected address at the serial output MISO (PB6) pin.
At the end of the programming session, \overline{RESET} can be set high to commence normal operation.
6. Power-off sequence (if needed):
Set XTAL1 to "0" (if a crystal is not used or the device is running from the Internal RC Oscillator).
Set \overline{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 18 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 22 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 18. Read Back Value during EEPROM Polling

| Part | P1 | P2 |
|-----------|------|------|
| AT90S1200 | \$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 35. Serial Programming Waveforms

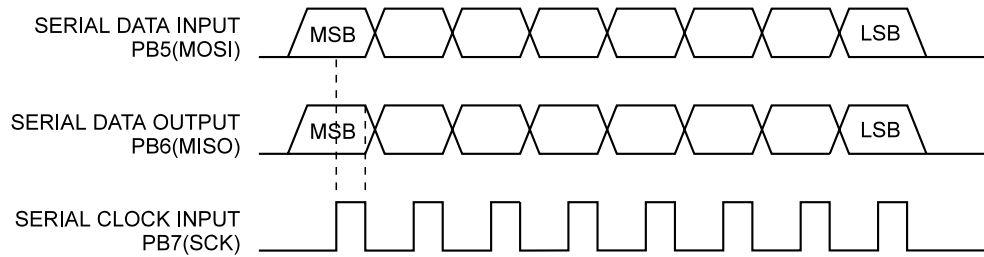


Table 19. Serial Programming Instruction Set for AT90S1200

| 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 both Flash and EEPROM memory arrays. |
| Read Program Memory | 0010 H000 | 0000 000a | bbbb bbbb | oooo oooo | Read H (high or low) byte o from program memory at word address a:b. |
| Write Program Memory | 0100 H000 | 0000 000a | bbbb bbbb | iiii iiii | Write H (high or low) byte i to program memory at word address a:b. |
| Read EEPROM Memory | 1010 0000 | 0000 0000 | 00bb bbbb | oooo oooo | Read data o from EEPROM memory at address b. |
| Write EEPROM Memory | 1100 0000 | 0000 0000 | 00bb bbbb | iiii iiii | Write data i to EEPROM memory at address b. |
| Write Lock Bits | 1010 1100 | 1111 1211 | xxxx xxxx | xxxx xxxx | Write Lock bits. Set bits 1,2 = "0" to program Lock bits. |
| Read Signature Byte | 0011 0000 | xxxx xxxx | xxxx xxbb | oooo oooo | Read signature byte o from address b. ⁽¹⁾ |

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

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

Serial Programming Characteristics

Figure 36. Serial Programming Timing

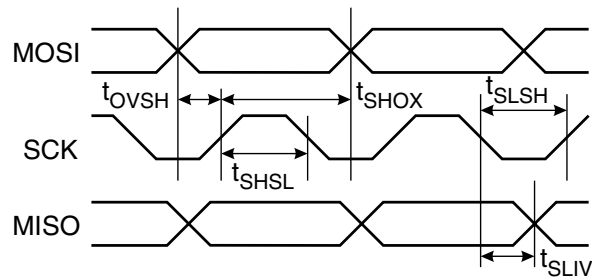


Table 20. Serial Programming Characteristics, $T_A = -40^{\circ}\text{C}$ to 85°C , $V_{CC} = 2.7 - 6.0\text{V}$ (unless otherwise noted)

| Symbol | Parameter | Min | Typ | Max | Units |
|--------------|---|-----------------|------|------|-------|
| $1/t_{CLCL}$ | Oscillator Frequency ($V_{CC} = 2.7 - 4.0\text{V}$) | 0 | | 4.0 | MHz |
| t_{CLCL} | Oscillator Period ($V_{CC} = 2.7 - 4.0\text{V}$) | 250.0 | | | ns |
| $1/t_{CLCL}$ | Oscillator Frequency ($V_{CC} = 4.0 - 6.0\text{V}$) | 0 | | 12.0 | MHz |
| t_{CLCL} | Oscillator Period ($V_{CC} = 4.0 - 6.0\text{V}$) | 83.3 | | | ns |
| t_{SHSL} | SCK Pulse Width High | $4.0 t_{CLCL}$ | | | ns |
| t_{SLSH} | SCK Pulse Width Low | t_{CLCL} | | | ns |
| t_{OVSH} | MOSI Setup to SCK High | $1.25 t_{CLCL}$ | | | ns |
| t_{SHOX} | MOSI Hold after SCK High | $2.5 t_{CLCL}$ | | | ns |
| t_{SLIV} | SCK Low to MISO Valid | 10.0 | 16.0 | 32.0 | ns |

Table 21. Minimum Wait Delay after the Chip Erase Instruction

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

Table 22. Minimum Wait Delay after Writing a Flash or EEPROM Location

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

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 | 200.0 mA |

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

DC Characteristics

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

| Symbol | Parameter | Condition | Min | Typ | Max | Units |
|-----------|--|--|---------------------|------|--------------------|---------------|
| V_{IL} | Input Low Voltage | (Except XTAL1) | -0.5 | | $0.3 V_{CC}^{(1)}$ | V |
| V_{IL1} | Input Low Voltage | (XTAL1) | -0.5 | | $0.3 V_{CC}^{(1)}$ | V |
| V_{IH} | Input High Voltage | (Except XTAL1, $\overline{\text{RESET}}$) | $0.6 V_{CC}^{(2)}$ | | $V_{CC} + 0.5$ | V |
| V_{IH1} | Input High Voltage | (XTAL1) | $0.7 V_{CC}^{(2)}$ | | $V_{CC} + 0.5$ | 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, D) | $I_{OL} = 20 \text{ mA}$, $V_{CC} = 5V$ $I_{OL} = 10 \text{ mA}$, $V_{CC} = 3V$ | | | 0.6 0.5 | V V |
| V_{OH} | Output High Voltage ⁽⁴⁾ (Ports B, D) | $I_{OH} = -3 \text{ mA}$, $V_{CC} = 5V$ $I_{OH} = -1.5 \text{ mA}$, $V_{CC} = 3V$ | 4.3 2.3 | | | V V |
| I_{IL} | Input Leakage Current I/O pin | $V_{CC} = 6V$, pin low (absolute value) | | | 8.0 | μA |
| I_{IH} | Input Leakage Current I/O pin | $V_{CC} = 6V$, pin high (absolute value) | | | 980.0 | nA |
| RRST | Reset Pull-up Resistor | | 100.0 | | 500.0 | $k\Omega$ |
| $R_{I/O}$ | I/O Pin Pull-up Resistor | | 35.0 | | 120.0 | $k\Omega$ |
| I_{CC} | Power Supply Current | Active Mode, $V_{CC} = 3V$, 4 MHz | | | 3.0 | mA |
| | | Idle Mode $V_{CC} = 3V$, 4 MHz | | | 1.0 | mA |
| I_{CC} | Power-down mode ⁽⁵⁾ | WDT enabled, $V_{CC} = 3V$ | | 9.0 | 15.0 | μA |
| | | WDT disabled, $V_{CC} = 3V$ | | <1.0 | 2.0 | μA |

DC Characteristics

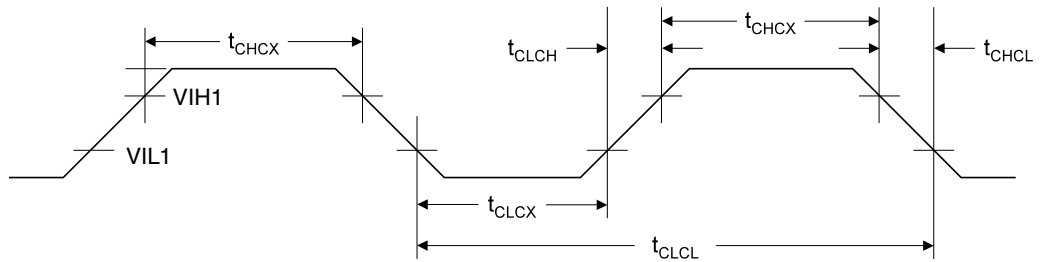
$T_A = -40^{\circ}\text{C}$ to 85°C , $V_{CC} = 2.7\text{V}$ to 6.0V (unless otherwise noted) (Continued)

| Symbol | Parameter | Condition | Min | Typ | Max | Units |
|------------|---|--|-------|----------------|------|-------|
| V_{ACIO} | Analog Comparator Input Offset Voltage | $V_{CC} = 5\text{V}$ $V_{in} = V_{CC}/2$ | | | 40.0 | mV |
| I_{ACLK} | Analog Comparator Input Leakage Current | $V_{CC} = 5\text{V}$ $V_{in} = V_{CC}/2$ | -50.0 | | 50.0 | nA |
| t_{ACPD} | Analog Comparator Propagation Delay | $V_{CC} = 2.7\text{V}$ $V_{CC} = 4.0\text{V}$ | | 750.0 500.0 | | ns |

- Notes:
1. "Max" means the highest value where the pin is guaranteed to be read as low.
 2. "Min" means the lowest value where the pin is guaranteed to be read as high.
 3. Although each I/O port can sink more than the test conditions (20 mA at $V_{CC} = 5\text{V}$, 10 mA at $V_{CC} = 3\text{V}$) under steady state conditions (non-transient), the following must be observed:
 - 1] The sum of all I_{OL} , for all ports, should not exceed 200 mA.
 - 2] The sum of all I_{OL} , for port D0 - D5 and XTAL2, should not exceed 100 mA.
 - 3] The sum of all I_{OL} , for ports B0 - B7 and D6, should not exceed 100 mA.
 If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test condition.
 4. Although each I/O port can source more than the test conditions (3 mA at $V_{CC} = 5\text{V}$, 1.5 mA at $V_{CC} = 3\text{V}$) under steady state conditions (non-transient), the following must be observed:
 - 1] The sum of all I_{OH} , for all ports, should not exceed 200 mA.
 - 2] The sum of all I_{OH} , for port D0 - D5 and XTAL2, should not exceed 100 mA.
 - 3] The sum of all I_{OH} , for ports B0 - B7 and D6, should not exceed 100 mA.
 If I_{OH} exceeds the test condition, V_{OH} 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 37. External Clock Drive



External Clock Drive

Table 23. External Clock Drive

| Symbol | Parameter | $V_{CC} = 2.7V \text{ to } 4.0V$ | | $V_{CC} = 4.0V \text{ to } 6.0V$ | | Units |
|--------------|----------------------|----------------------------------|-----|----------------------------------|------|---------|
| | | Min | Max | Min | Max | |
| $1/t_{CLCL}$ | Oscillator Frequency | 0 | 4.0 | 0 | 12.0 | MHz |
| t_{CLCL} | Clock Period | 250.0 | | 83.3 | | ns |
| t_{CHCX} | High Time | 100.0 | | 33.3 | | ns |
| t_{CLCX} | Low Time | 100.0 | | 33.3 | | 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 figures are not tested during manufacturing. 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 $C_L \cdot V_{CC} \cdot f$ where C_L = load capacitance, V_{CC} = operating voltage and f = average switching frequency of I/O pin.

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

The difference between current consumption in Power-down mode with Watchdog Timer enabled and Power-down mode with Watchdog Timer disabled represents the differential current drawn by the Watchdog Timer.

Figure 38. Active Supply Current vs. Frequency

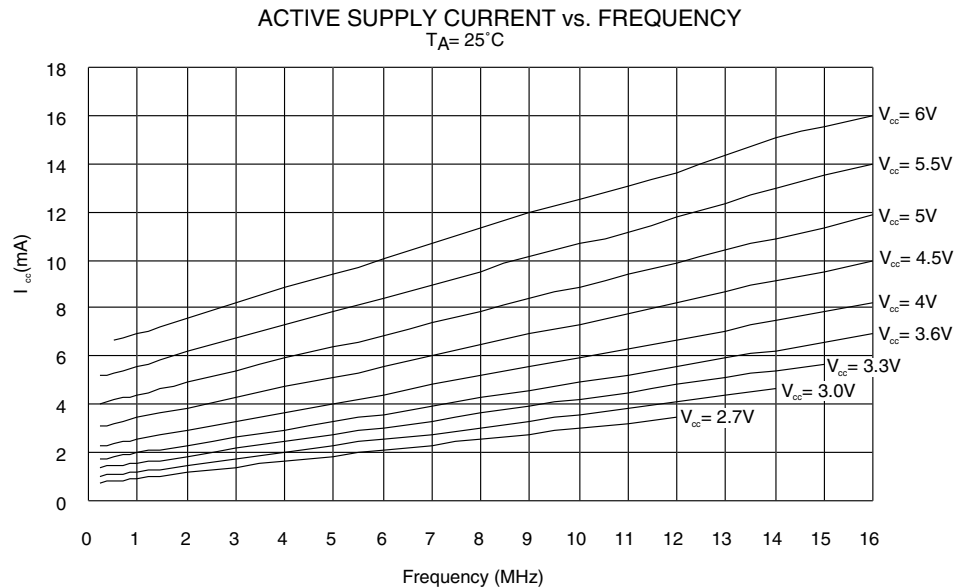


Figure 39. Active Supply Current vs. V_{CC}

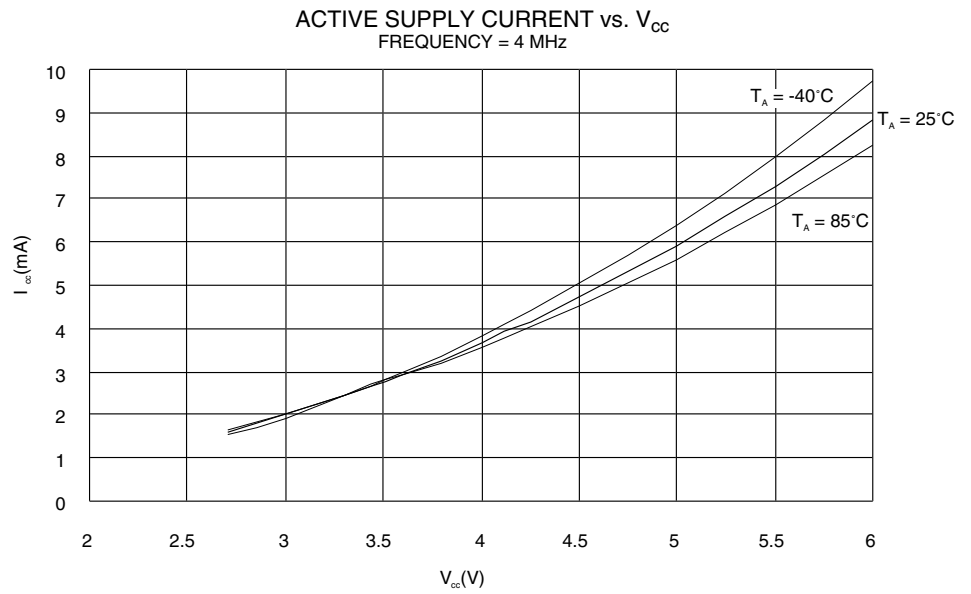


Figure 40. Active Supply Current vs. V_{CC} , Device Clocked by Internal Oscillator

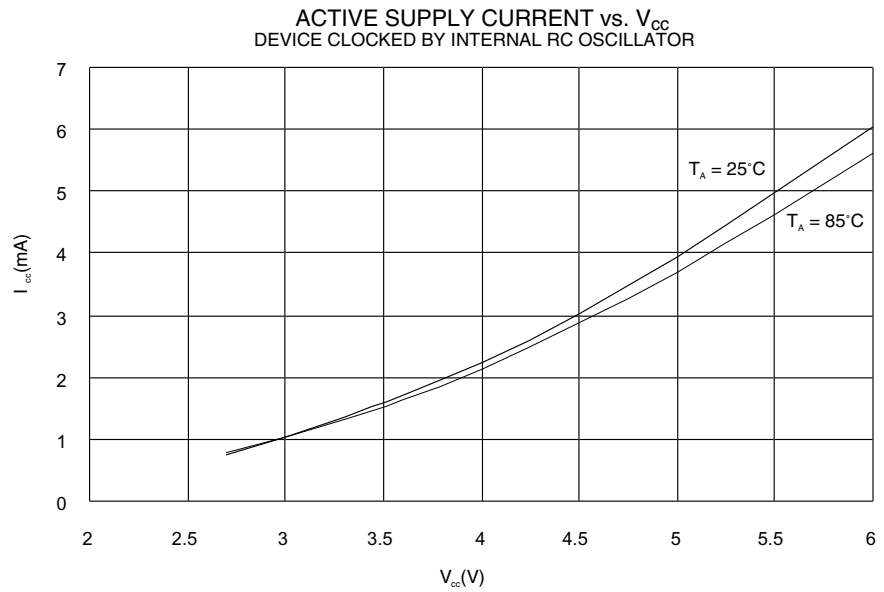


Figure 41. Idle Supply Current vs. Frequency

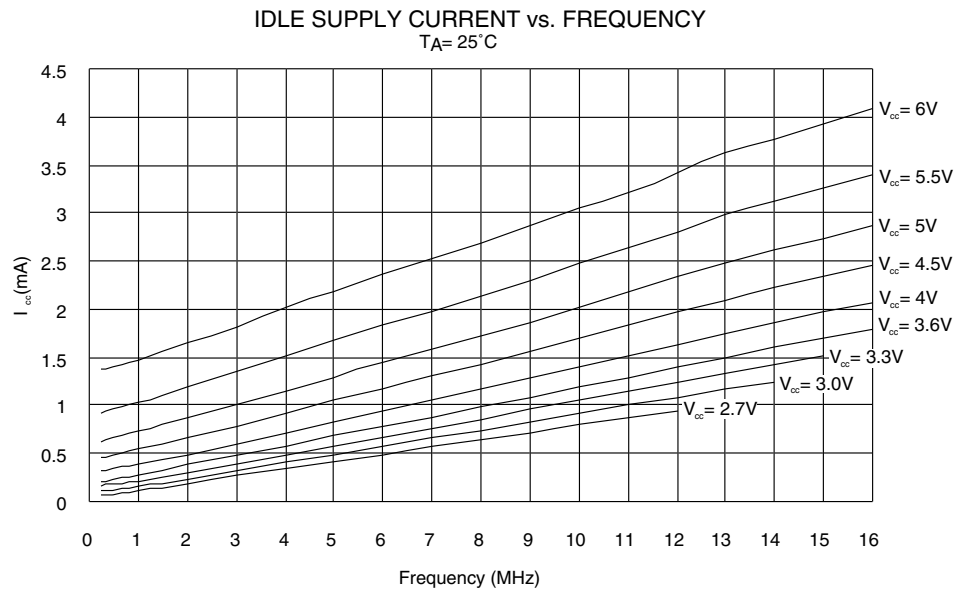


Figure 42. Idle Supply Current vs. V_{CC}

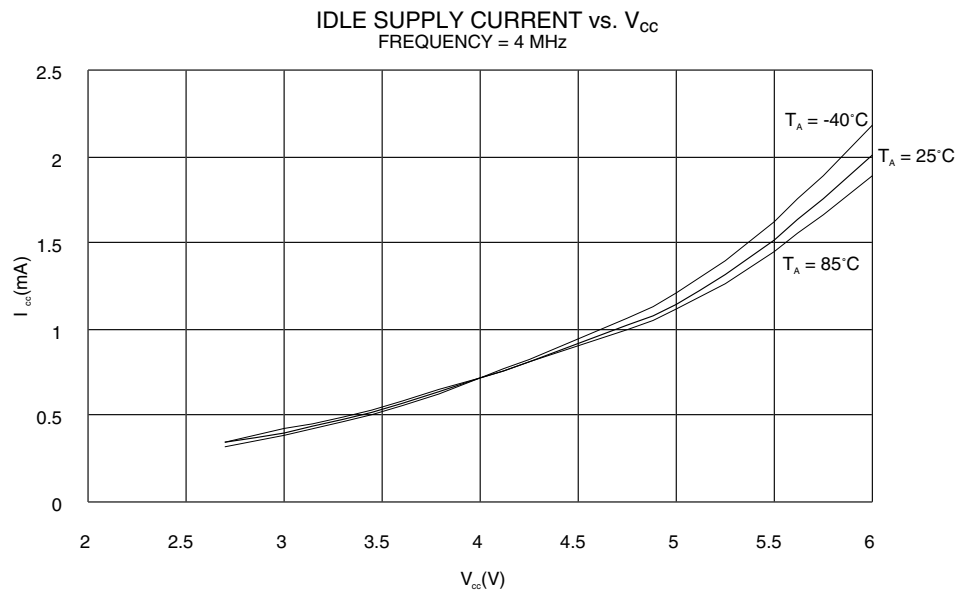


Figure 43. Idle Supply Current vs. V_{CC} , Device Clocked by Internal Oscillator

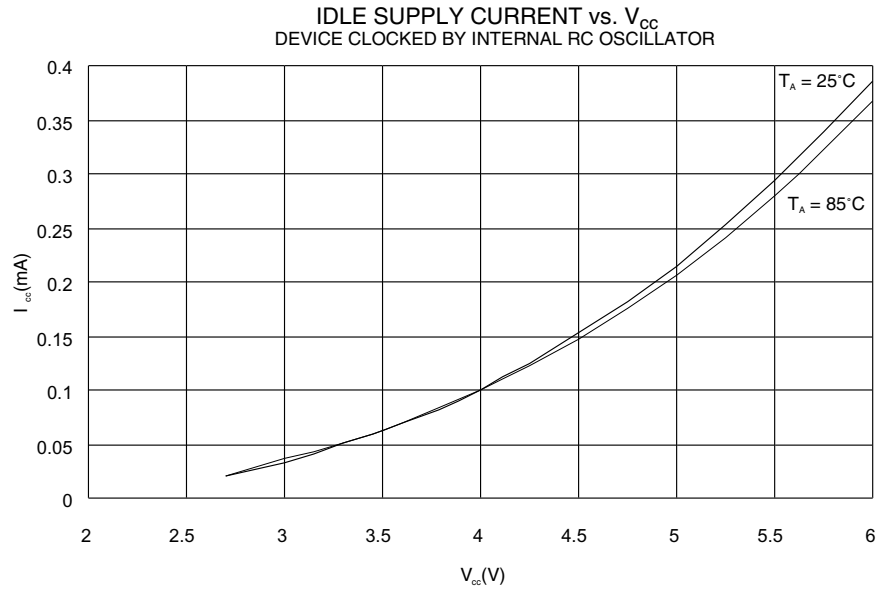


Figure 44. Power-down Supply Current vs. V_{CC} , Watchdog Timer Disabled

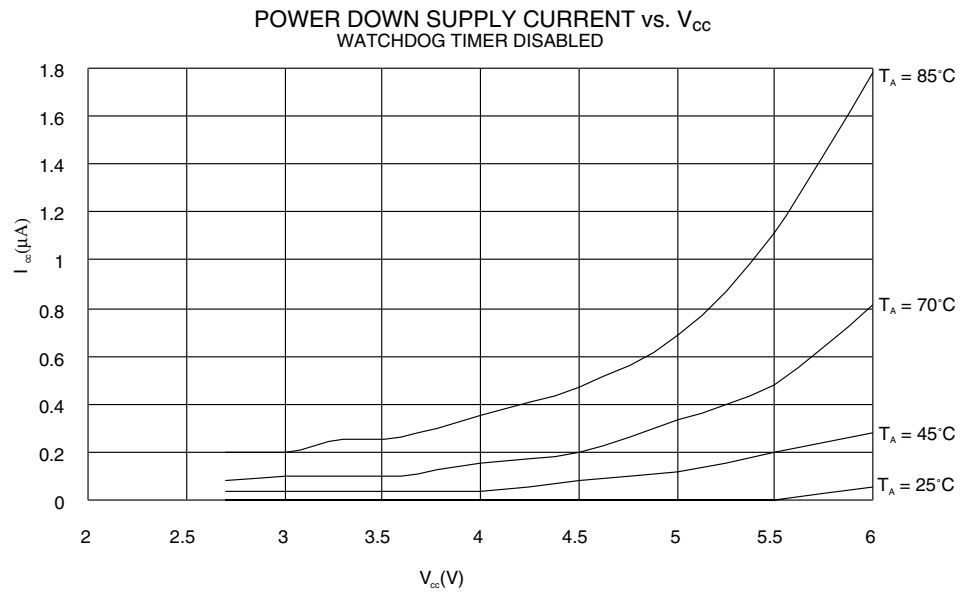


Figure 45. Power-down Supply Current vs. V_{CC} , Watchdog Timer Enabled

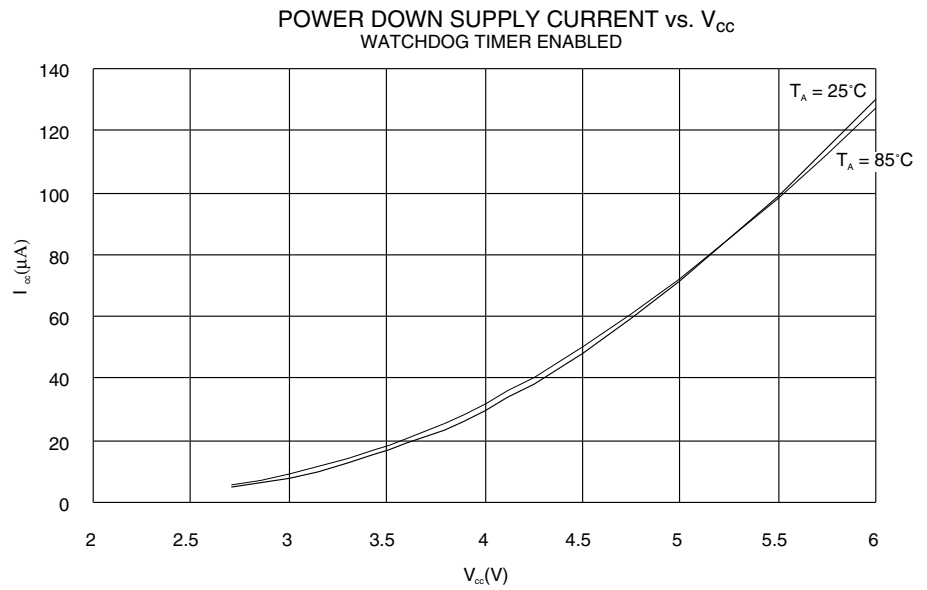


Figure 46. Internal RC Oscillator Frequency vs. V_{CC}

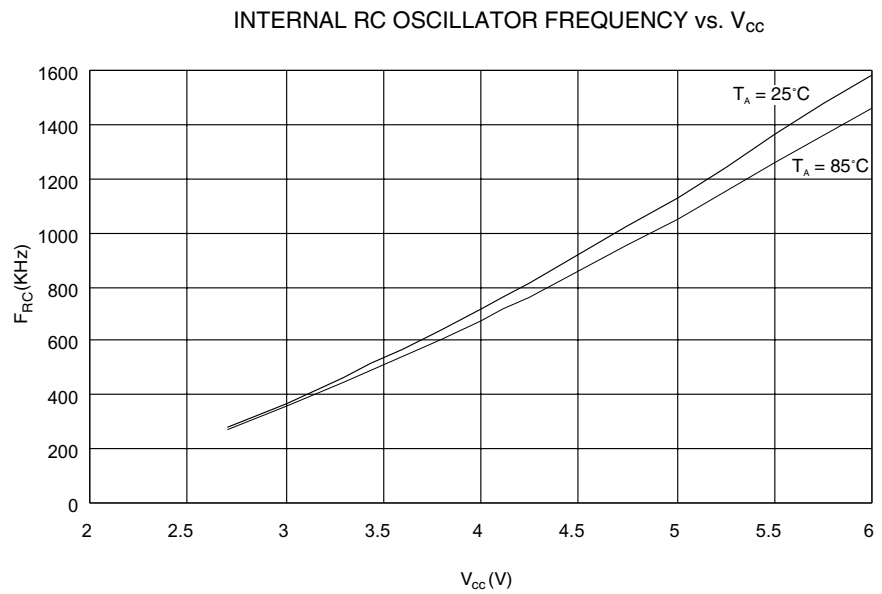
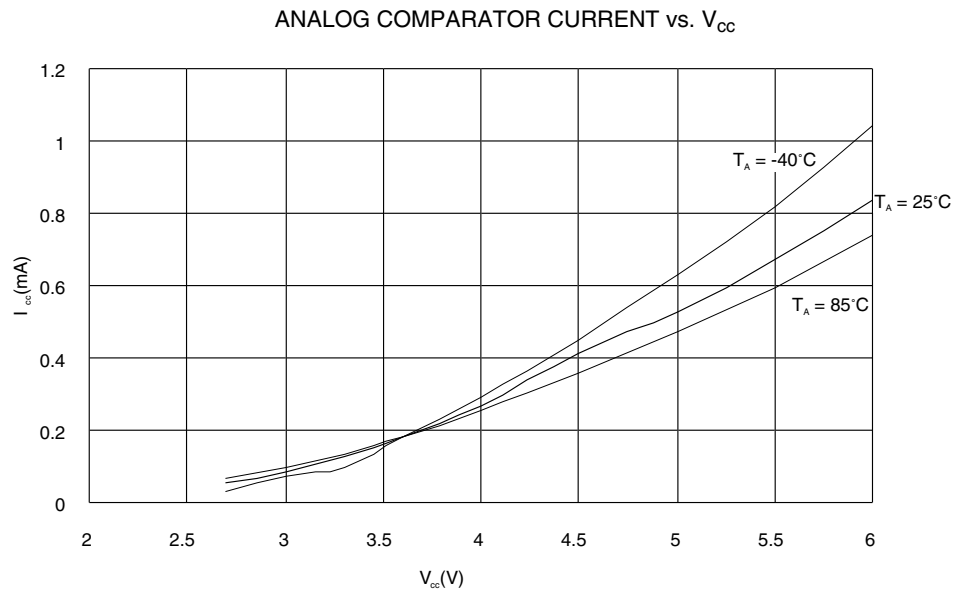


Figure 47. Analog Comparator Current vs. V_{CC}



Note: Analog comparator offset voltage is measured as absolute offset.

Figure 48. Analog Comparator Offset Voltage vs. Common Mode Voltage

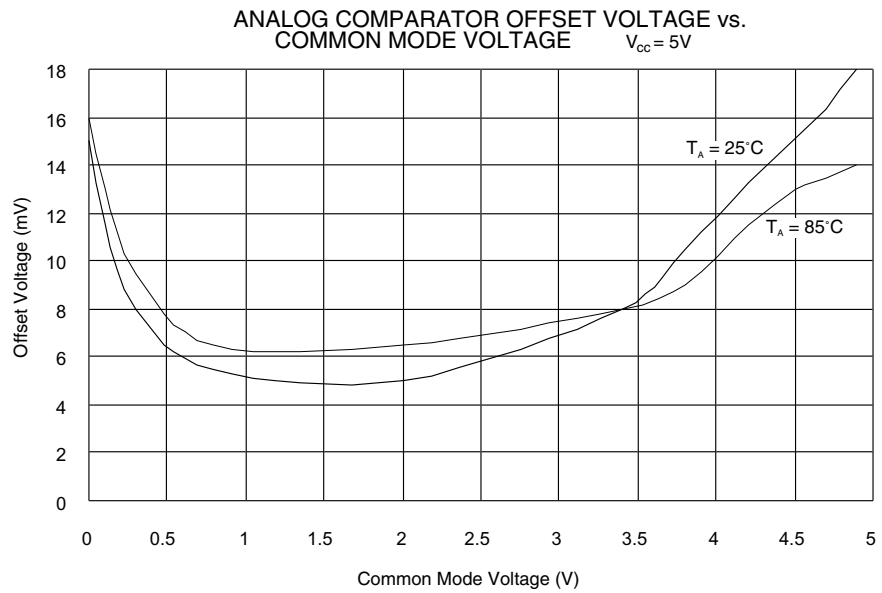


Figure 49. Analog Comparator Offset Voltage vs. Common Mode Voltage

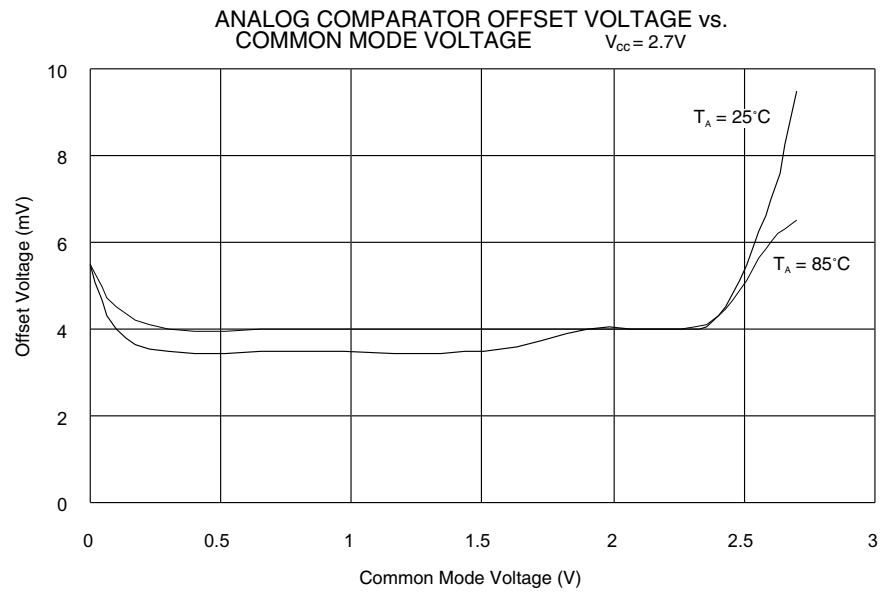
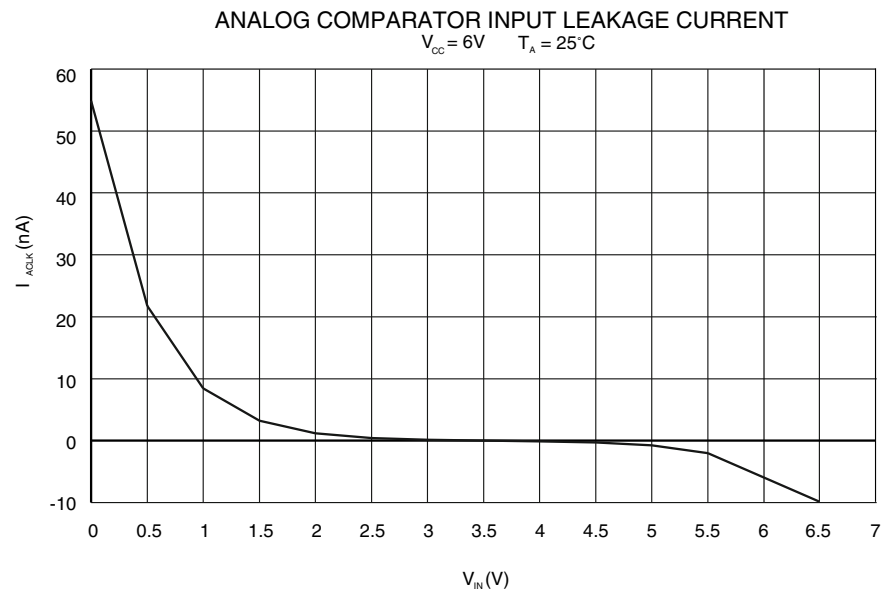


Figure 50. Analog Comparator Input Leakage Current



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

Figure 51. Pull-up Resistor Current vs. Input Voltage

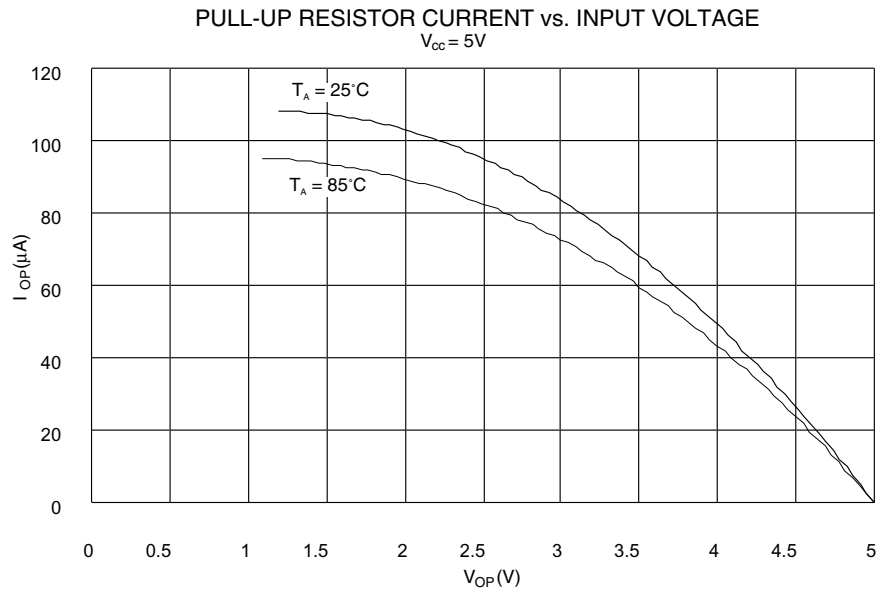


Figure 52. Pull-up Resistor Current vs. Input Voltage

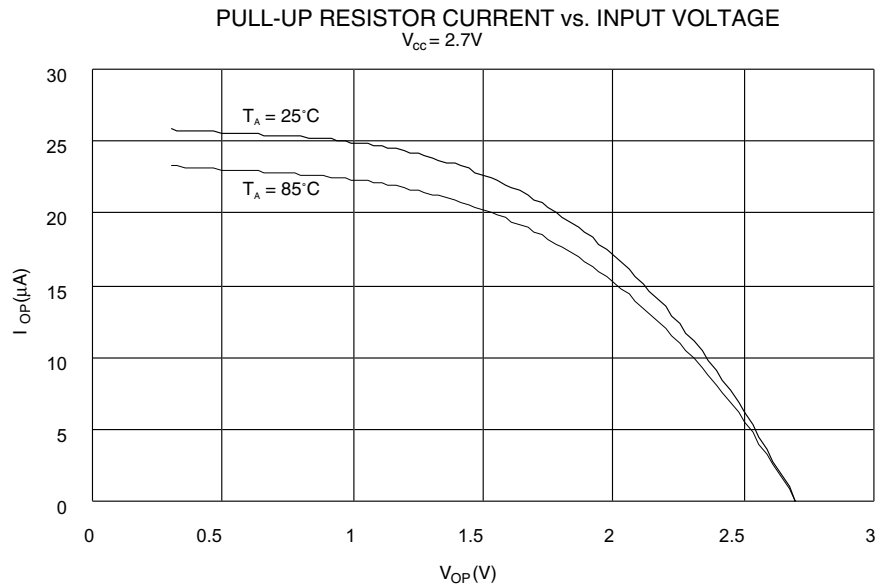


Figure 53. I/O Pin Sink Current vs. Output Voltage

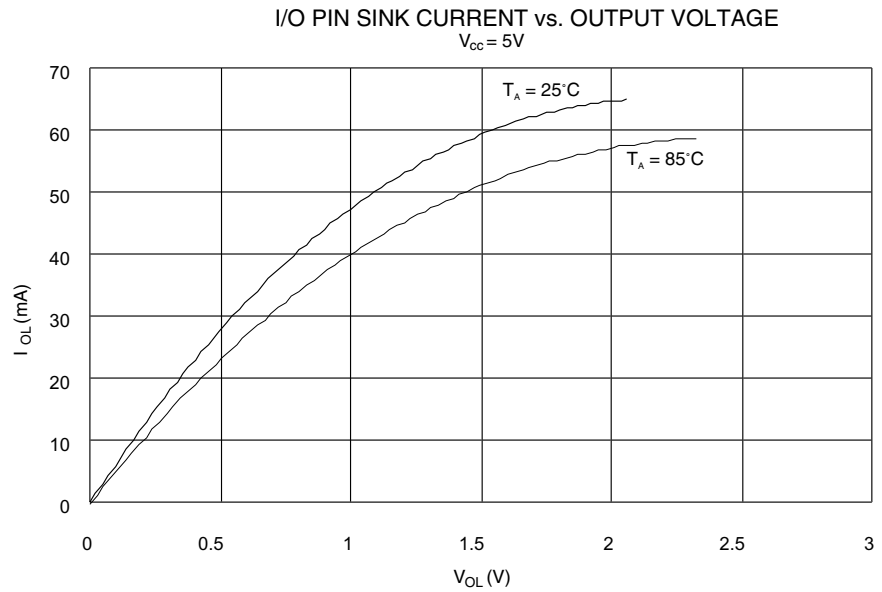


Figure 54. I/O Pin Source Current vs. Output Voltage

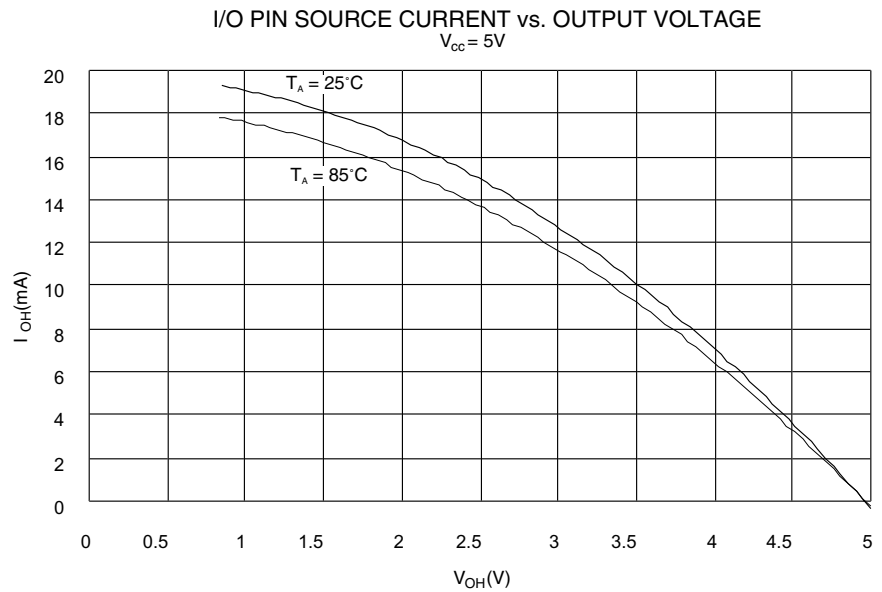


Figure 55. I/O Pin Sink Current vs. Output Voltage

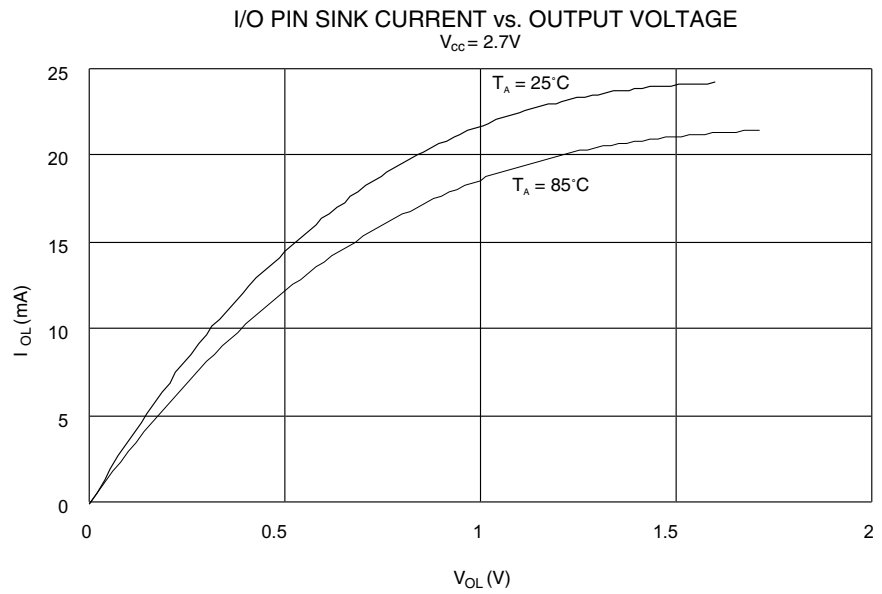
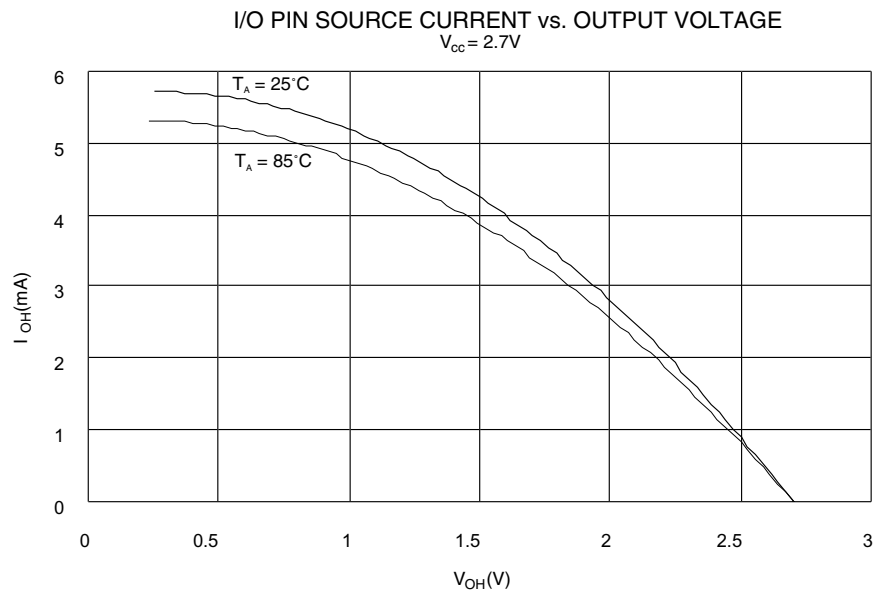


Figure 56. I/O Pin Source Current vs. Output Voltage



Note: Input threshold is measured at the center point of the hysteresis.

Figure 57. I/O Pin Input Threshold Voltage vs. V_{CC}

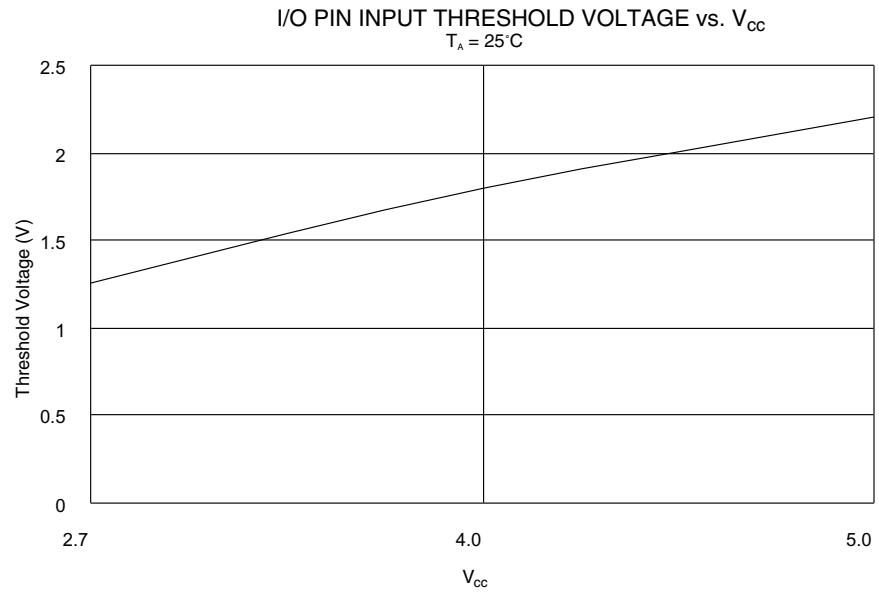
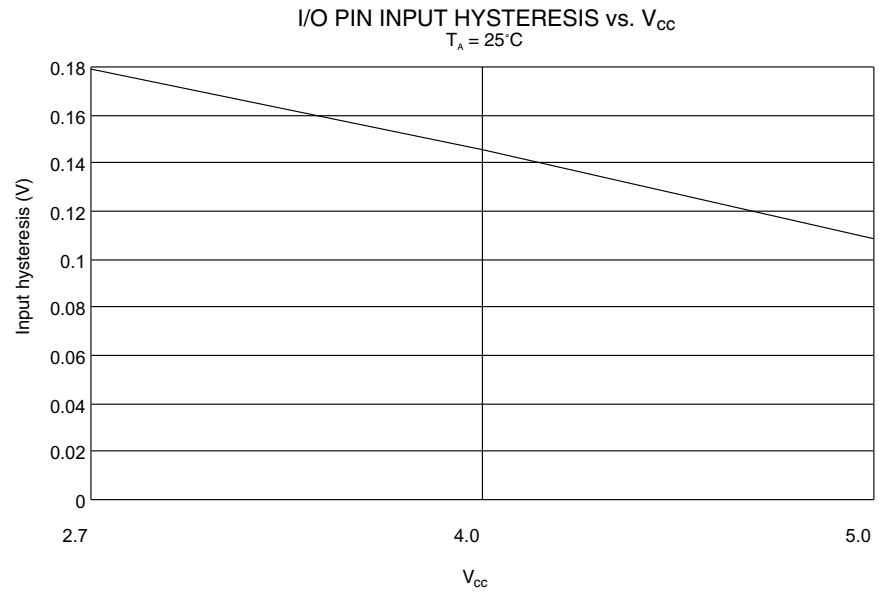


Figure 58. I/O Pin Input Hysteresis vs. V_{CC}





AT90S1200 Register Summary

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Page |
|---------|----------|-------------------------|-------------------------|--------|--------|--------|--------|--------|--------|---------|
| \$3F | SREG | I | T | H | S | V | N | Z | C | page 11 |
| \$3E | Reserved | | | | | | | | | |
| \$3D | Reserved | | | | | | | | | |
| \$3C | Reserved | | | | | | | | | |
| \$3B | GIMSK | - | INT0 | - | - | - | - | - | - | page 15 |
| \$3A | Reserved | | | | | | | | | |
| \$39 | TIMSK | - | - | - | - | - | - | TOIE0 | - | page 16 |
| \$38 | TIFR | - | - | - | - | - | - | TOV0 | - | page 16 |
| \$37 | Reserved | | | | | | | | | |
| \$36 | Reserved | | | | | | | | | |
| \$35 | MCUCR | - | - | SE | SM | - | - | ISC01 | ISC00 | page 18 |
| \$34 | Reserved | | | | | | | | | |
| \$33 | TCCR0 | - | - | - | - | - | CS02 | CS01 | CS00 | page 21 |
| \$32 | TCNT0 | Timer/Counter0 (8 Bits) | | | | | | | | page 22 |
| \$31 | Reserved | | | | | | | | | |
| \$30 | Reserved | | | | | | | | | |
| \$2F | Reserved | | | | | | | | | |
| \$2E | Reserved | | | | | | | | | |
| \$2D | Reserved | | | | | | | | | |
| \$2C | Reserved | | | | | | | | | |
| \$2B | Reserved | | | | | | | | | |
| \$2A | Reserved | | | | | | | | | |
| \$29 | Reserved | | | | | | | | | |
| \$28 | Reserved | | | | | | | | | |
| \$27 | Reserved | | | | | | | | | |
| \$26 | Reserved | | | | | | | | | |
| \$25 | Reserved | | | | | | | | | |
| \$24 | Reserved | | | | | | | | | |
| \$23 | Reserved | | | | | | | | | |
| \$22 | Reserved | | | | | | | | | |
| \$21 | WDTCR | - | - | - | - | WDE | WDP2 | WDP1 | WDP0 | page 23 |
| \$20 | Reserved | | | | | | | | | |
| \$1F | Reserved | | | | | | | | | |
| \$1E | EEAR | - | EEPROM Address Register | | | | | | | page 25 |
| \$1D | EEDR | EEPROM Data Register | | | | | | | | page 25 |
| \$1C | EEDR | - | - | - | - | - | - | EERE | EERE | page 25 |
| \$1B | Reserved | | | | | | | | | |
| \$1A | Reserved | | | | | | | | | |
| \$19 | Reserved | | | | | | | | | |
| \$18 | PORTB | PORTB7 | PORTB6 | PORTB5 | PORTB4 | PORTB3 | PORTB2 | PORTB1 | PORTB0 | page 29 |
| \$17 | DDRB | DDB7 | DDB6 | DDB5 | DDB4 | DDB3 | DDB2 | DDB1 | DDB0 | page 29 |
| \$16 | PINB | PINB7 | PINB6 | PINB5 | PINB4 | PINB3 | PINB2 | PINB1 | PINB0 | page 29 |
| \$15 | Reserved | | | | | | | | | |
| \$14 | Reserved | | | | | | | | | |
| \$13 | Reserved | | | | | | | | | |
| \$12 | PORTD | - | PORTD6 | PORTD5 | PORTD4 | PORTD3 | PORTD2 | PORTD1 | PORTD0 | page 34 |
| \$11 | DDRD | - | DDD6 | DDD5 | DDD4 | DDD3 | DDD2 | DDD1 | DDD0 | page 34 |
| \$10 | PIND | - | PIND6 | PIND5 | PIND4 | PIND3 | PIND2 | PIND1 | PIND0 | page 34 |
| \$0F | Reserved | | | | | | | | | |
| ... | Reserved | | | | | | | | | |
| \$09 | Reserved | | | | | | | | | |
| \$08 | ACSR | ACD | - | ACO | ACI | ACIE | - | ACIS1 | ACIS0 | page 27 |
| ... | Reserved | | | | | | | | | |
| \$00 | 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 "1" to them. Note that the CBI and SBI instructions will operate on all bits in the I/O register, writing a "1" 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

| Mnemonic | Operands | Description | Operation | Flags | # Clocks |
|--|----------|--|---|-----------|----------|
| ARITHMETIC AND LOGIC INSTRUCTIONS | | | | | |
| ADD | Rd, Rr | Add Two Registers | $Rd \leftarrow Rd + Rr$ | Z,C,N,V,H | 1 |
| ADC | Rd, Rr | Add with Carry Two Registers | $Rd \leftarrow Rd + Rr + C$ | Z,C,N,V,H | 1 |
| SUB | Rd, Rr | Subtract Two Registers | $Rd \leftarrow Rd - Rr$ | Z,C,N,V,H | 1 |
| SUBI | Rd, K | Subtract Constant from Register | $Rd \leftarrow Rd - K$ | Z,C,N,V,H | 1 |
| SBC | Rd, Rr | Subtract with Carry Two Registers | $Rd \leftarrow Rd - Rr - C$ | Z,C,N,V,H | 1 |
| SBCI | Rd, K | Subtract with Carry Constant from Reg. | $Rd \leftarrow Rd - K - C$ | Z,C,N,V,H | 1 |
| AND | Rd, Rr | Logical AND Registers | $Rd \leftarrow Rd \cdot Rr$ | Z,N,V | 1 |
| ANDI | Rd, K | Logical AND Register and Constant | $Rd \leftarrow Rd \cdot K$ | Z,N,V | 1 |
| OR | Rd, Rr | Logical OR Registers | $Rd \leftarrow Rd \vee Rr$ | Z,N,V | 1 |
| ORI | Rd, K | Logical OR Register and Constant | $Rd \leftarrow Rd \vee K$ | Z,N,V | 1 |
| EOR | Rd, Rr | Exclusive OR Registers | $Rd \leftarrow Rd \oplus Rr$ | Z,N,V | 1 |
| COM | Rd | One's Complement | $Rd \leftarrow \$FF - Rd$ | Z,C,N,V | 1 |
| NEG | Rd | Two's Complement | $Rd \leftarrow \$00 - Rd$ | Z,C,N,V,H | 1 |
| SBR | Rd, K | Set Bit(s) in Register | $Rd \leftarrow Rd \vee K$ | Z,N,V | 1 |
| CBR | Rd, K | Clear Bit(s) in Register | $Rd \leftarrow Rd \cdot (\text{FFh} - K)$ | Z,N,V | 1 |
| INC | Rd | Increment | $Rd \leftarrow Rd + 1$ | Z,N,V | 1 |
| DEC | Rd | Decrement | $Rd \leftarrow Rd - 1$ | Z,N,V | 1 |
| TST | Rd | Test for Zero or Minus | $Rd \leftarrow Rd \cdot Rd$ | Z,N,V | 1 |
| CLR | Rd | Clear Register | $Rd \leftarrow Rd \oplus Rd$ | Z,N,V | 1 |
| SER | Rd | Set Register | $Rd \leftarrow \$FF$ | None | 1 |
| BRANCH INSTRUCTIONS | | | | | |
| RJMP | k | Relative Jump | $PC \leftarrow PC + k + 1$ | None | 2 |
| RCALL | k | Relative Subroutine Call | $PC \leftarrow PC + k + 1$ | None | 3 |
| RET | | Subroutine Return | $PC \leftarrow \text{STACK}$ | None | 4 |
| RETI | | Interrupt Return | $PC \leftarrow \text{STACK}$ | I | 4 |
| CPSE | Rd, Rr | Compare, Skip if Equal | if (Rd = Rr) $PC \leftarrow PC + 2$ or 3 | None | 1/2 |
| CP | Rd, Rr | Compare | $Rd - Rr$ | Z,N,V,C,H | 1 |
| CPC | Rd, Rr | Compare with Carry | $Rd - Rr - C$ | Z,N,V,C,H | 1 |
| CPI | Rd, K | Compare Register with Immediate | $Rd - K$ | Z,N,V,C,H | 1 |
| SBRC | Rr, b | Skip if Bit in Register Cleared | if (Rr(b) = 0) $PC \leftarrow PC + 2$ or 3 | None | 1/2 |
| SBRS | Rr, b | Skip if Bit in Register is Set | if (Rr(b) = 1) $PC \leftarrow PC + 2$ or 3 | None | 1/2 |
| SBIC | P, b | Skip if Bit in I/O Register Cleared | if (P(b) = 0) $PC \leftarrow PC + 2$ or 3 | None | 1/2 |
| SBIS | P, b | Skip if Bit in I/O Register is Set | if (P(b) = 1) $PC \leftarrow PC + 2$ or 3 | None | 1/2 |
| BRBS | s, k | Branch if Status Flag Set | if (SREG(s) = 1) then $PC \leftarrow PC + k + 1$ | None | 1/2 |
| BRBC | s, k | Branch if Status Flag Cleared | if (SREG(s) = 0) then $PC \leftarrow PC + k + 1$ | None | 1/2 |
| BREQ | k | Branch if Equal | if (Z = 1) then $PC \leftarrow PC + k + 1$ | None | 1/2 |
| BRNE | k | Branch if Not Equal | if (Z = 0) then $PC \leftarrow PC + k + 1$ | None | 1/2 |
| BRCS | k | Branch if Carry Set | if (C = 1) then $PC \leftarrow PC + k + 1$ | None | 1/2 |
| BRCC | k | Branch if Carry Cleared | if (C = 0) then $PC \leftarrow PC + k + 1$ | None | 1/2 |
| BRSH | k | Branch if Same or Higher | if (C = 0) then $PC \leftarrow PC + k + 1$ | None | 1/2 |
| BRLO | k | Branch if Lower | if (C = 1) then $PC \leftarrow PC + k + 1$ | None | 1/2 |
| BRMI | k | Branch if Minus | if (N = 1) then $PC \leftarrow PC + k + 1$ | None | 1/2 |
| BRPL | k | Branch if Plus | if (N = 0) then $PC \leftarrow PC + k + 1$ | None | 1/2 |
| BRGE | k | Branch if Greater or Equal, Signed | if (N \oplus V = 0) then $PC \leftarrow PC + k + 1$ | None | 1/2 |
| BRLT | k | Branch if Less than Zero, Signed | if (N \oplus V = 1) then $PC \leftarrow PC + k + 1$ | None | 1/2 |
| BRHS | k | Branch if Half-carry Flag Set | if (H = 1) then $PC \leftarrow PC + k + 1$ | None | 1/2 |
| BRHC | k | Branch if Half-carry Flag Cleared | if (H = 0) then $PC \leftarrow PC + k + 1$ | None | 1/2 |
| BRTS | k | Branch if T-Flag Set | if (T = 1) then $PC \leftarrow PC + k + 1$ | None | 1/2 |
| BRTC | k | Branch if T-Flag Cleared | if (T = 0) then $PC \leftarrow PC + k + 1$ | None | 1/2 |
| BRVS | k | Branch if Overflow Flag is Set | if (V = 1) then $PC \leftarrow PC + k + 1$ | None | 1/2 |
| BRVC | k | Branch if Overflow Flag is Cleared | if (V = 0) then $PC \leftarrow PC + k + 1$ | None | 1/2 |
| BRIE | k | Branch if Interrupt Enabled | if (I = 1) then $PC \leftarrow PC + k + 1$ | None | 1/2 |
| BRID | k | Branch if Interrupt Disabled | if (I = 0) then $PC \leftarrow PC + k + 1$ | None | 1/2 |
| DATA TRANSFER INSTRUCTIONS | | | | | |
| LD | Rd, Z | Load Register Indirect | $Rd \leftarrow (Z)$ | None | 2 |
| ST | Z, Rr | Store Register Indirect | $(Z) \leftarrow Rr$ | None | 2 |
| MOV | Rd, Rr | Move between Registers | $Rd \leftarrow Rr$ | None | 1 |
| LDI | Rd, K | Load Immediate | $Rd \leftarrow K$ | None | 1 |
| IN | Rd, P | In Port | $Rd \leftarrow P$ | None | 1 |
| OUT | P, Rr | Out Port | $P \leftarrow Rr$ | None | 1 |



Instruction Set Summary (Continued)

| Mnemonic | Operands | Description | Operation | Flags | # Clocks |
|--------------------------------------|----------|---------------------------------|--|---------|----------|
| BIT AND BIT-TEST INSTRUCTIONS | | | | | |
| SBI | P, b | Set Bit in I/O Register | $I/O(P,b) \leftarrow 1$ | None | 2 |
| CBI | P, b | Clear Bit in I/O Register | $I/O(P,b) \leftarrow 0$ | None | 2 |
| LSL | Rd | Logical Shift Left | $Rd(n+1) \leftarrow Rd(n), Rd(0) \leftarrow 0$ | Z,C,N,V | 1 |
| LSR | Rd | Logical Shift Right | $Rd(n) \leftarrow Rd(n+1), Rd(7) \leftarrow 0$ | Z,C,N,V | 1 |
| ROL | Rd | Rotate Left through Carry | $Rd(0) \leftarrow C, Rd(n+1) \leftarrow Rd(n), C \leftarrow Rd(7)$ | Z,C,N,V | 1 |
| ROR | Rd | Rotate Right through Carry | $Rd(7) \leftarrow C, Rd(n) \leftarrow Rd(n+1), C \leftarrow Rd(0)$ | Z,C,N,V | 1 |
| ASR | Rd | Arithmetic Shift Right | $Rd(n) \leftarrow Rd(n+1), n = 0..6$ | Z,C,N,V | 1 |
| SWAP | Rd | Swap Nibbles | $Rd(3..0) \leftarrow Rd(7..4), Rd(7..4) \leftarrow Rd(3..0)$ | None | 1 |
| BSET | s | Flag Set | $SREG(s) \leftarrow 1$ | SREG(s) | 1 |
| BCLR | s | Flag Clear | $SREG(s) \leftarrow 0$ | SREG(s) | 1 |
| BST | Rr, b | Bit Store from Register to T | $T \leftarrow Rr(b)$ | T | 1 |
| BLD | Rd, b | Bit Load from T to Register | $Rd(b) \leftarrow T$ | None | 1 |
| SEC | | Set Carry | $C \leftarrow 1$ | C | 1 |
| CLC | | Clear Carry | $C \leftarrow 0$ | C | 1 |
| SEN | | Set Negative Flag | $N \leftarrow 1$ | N | 1 |
| CLN | | Clear Negative Flag | $N \leftarrow 0$ | N | 1 |
| SEZ | | Set Zero Flag | $Z \leftarrow 1$ | Z | 1 |
| CLZ | | Clear Zero Flag | $Z \leftarrow 0$ | Z | 1 |
| SEI | | Global Interrupt Enable | $I \leftarrow 1$ | I | 1 |
| CLI | | Global Interrupt Disable | $I \leftarrow 0$ | I | 1 |
| SES | | Set Signed Test Flag | $S \leftarrow 1$ | S | 1 |
| CLS | | Clear Signed Test Flag | $S \leftarrow 0$ | S | 1 |
| SEV | | Set Two's Complement Overflow | $V \leftarrow 1$ | V | 1 |
| CLV | | Clear Two's Complement Overflow | $V \leftarrow 0$ | V | 1 |
| SET | | Set T in SREG | $T \leftarrow 1$ | T | 1 |
| CLT | | Clear T in SREG | $T \leftarrow 0$ | T | 1 |
| SEH | | Set Half-carry Flag in SREG | $H \leftarrow 1$ | H | 1 |
| CLH | | Clear Half-carry Flag in SREG | $H \leftarrow 0$ | H | 1 |
| NOP | | No Operation | | None | 1 |
| SLEEP | | Sleep | (see specific descr. for Sleep function) | None | 1 |
| WDR | | Watchdog Reset | (see specific descr. for WDR/timer) | None | 1 |

Ordering Information⁽¹⁾

| Speed (MHz) | Power Supply | Ordering Code | Package | Operation Range |
|-------------|--------------|----------------|---------|-------------------------------|
| 4 | 2.7 - 6.0V | AT90S1200-4PC | 20P3 | Commercial (0°C to 70°C) |
| | | AT90S1200-4SC | 20S | |
| | | AT90S1200-4YC | 20Y | |
| | | AT90S1200-4PI | 20P3 | Industrial (-40°C to 85°C) |
| | | AT90S1200-4SI | 20S | |
| | | AT90S1200-4YI | 20Y | |
| 12 | 4.0 - 6.0V | AT90S1200-12PC | 20P3 | Commercial (0°C to 70°C) |
| | | AT90S1200-12SC | 20S | |
| | | AT90S1200-12YC | 20Y | |
| | | AT90S1200-12PI | 20P3 | Industrial (-40°C to 85°C) |
| | | AT90S1200-12SI | 20S | |
| | | AT90S1200-12YI | 20Y | |

Note: 1. Order AT90S1200A-XXX for devices with the RCEN Fuse programmed.

| Package Type | |
|--------------|---|
| 20P3 | 20-lead, 0.300" Wide, Plastic Dual Inline Package (PDIP) |
| 20S | 20-lead, 0.300" Wide, Plastic Gull Wing Small Outline (SOIC) |
| 20Y | 20-lead, 5.3 mm Wide, Plastic Shrink Small Outline Package (SSOP) |

Packaging Information

20P3

COMMON DIMENSIONS
(Unit of Measure = mm)

| SYMBOL | MIN | NOM | MAX | NOTE |
|--------|-----------|-----|--------|--------|
| A | - | - | 5.334 | |
| A1 | 0.381 | - | - | |
| D | 25.984 | - | 25.493 | Note 2 |
| E | 7.620 | - | 8.255 | |
| E1 | 6.096 | - | 7.112 | Note 2 |
| B | 0.356 | - | 0.559 | |
| B1 | 1.270 | - | 1.551 | |
| L | 2.921 | - | 3.810 | |
| C | 0.203 | - | 0.356 | |
| eB | - | - | 10.922 | |
| eC | 0.000 | - | 1.524 | |
| e | 2.540 TYP | | | |

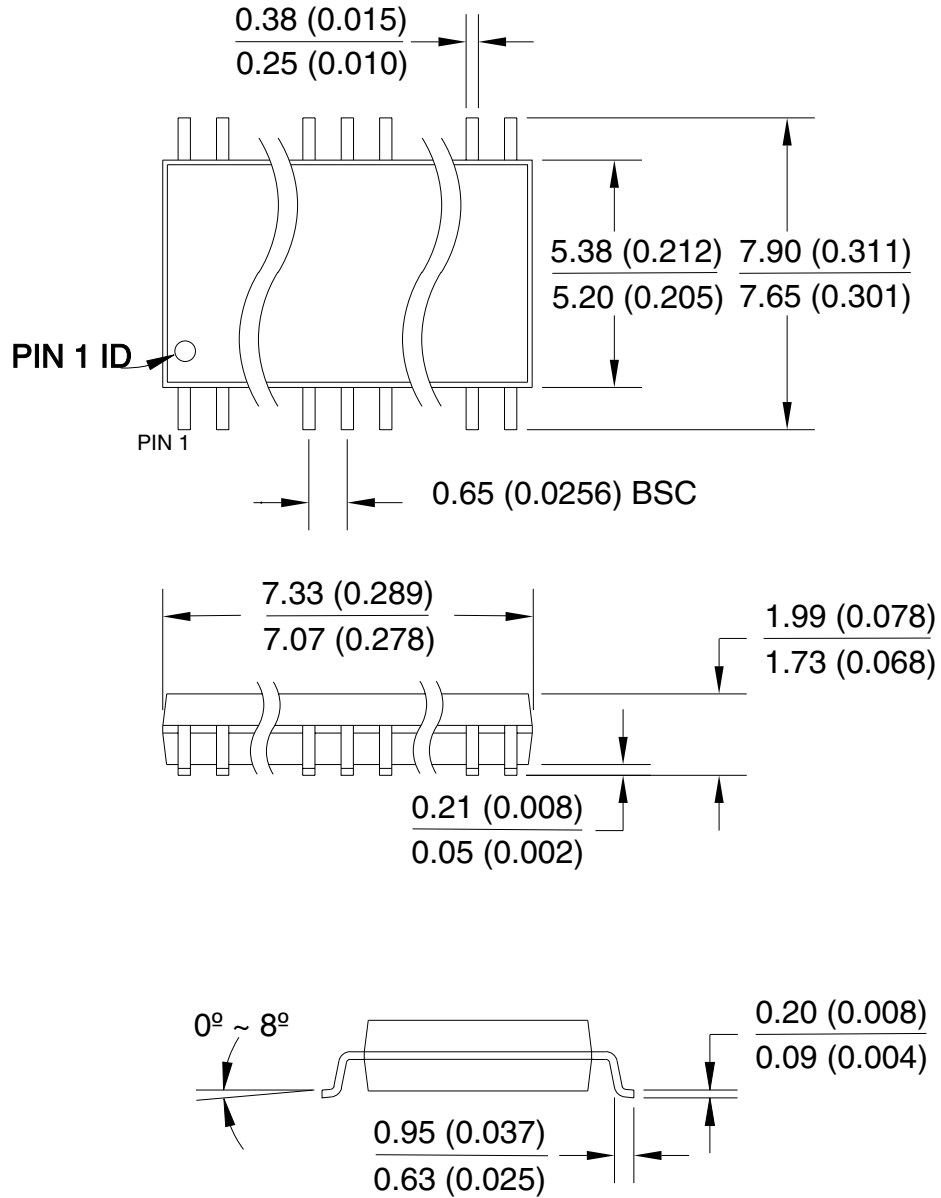
Notes: 1. This package conforms to JEDEC reference MS-001, Variation AD.
2. Dimensions D and E1 do not include mold Flash or Protrusion.
Mold Flash or Protrusion shall not exceed 0.25 mm (0.010").

09/28/01

| | | | | |
|--|--|---|--------------------|-------------|
| | 2325 Orchard Parkway San Jose, CA 95131 | TITLE | DRAWING NO. | REV. |
| | | 20P3, 20-lead (0.300"/7.62 mm Wide) Plastic Dual Inline Package (PDIP) | 20P3 | B |

20Y

20Y, 20-lead Plastic Shrink Small Outline (SSOP), 5.3mm body Width.
 Dimensions in Millimeters and (inches)*



*Controlling dimension: millimeters

REV. A 04/11/2001

| | | |
|--------------------------|---|-----------|
| Table of Contents | Features..... | 1 |
| | Pin Configuration..... | 1 |
| | Description..... | 2 |
| | Block Diagram | 2 |
| | Pin Descriptions..... | 3 |
| | Crystal Oscillator..... | 3 |
| | On-chip RC Oscillator | 4 |
| | Architectural Overview..... | 5 |
| | General Purpose Register File | 6 |
| | ALU – Arithmetic Logic Unit..... | 6 |
| | In-System Programmable Flash Program Memory | 6 |
| | Program and Data Addressing Modes..... | 7 |
| | Subroutine and Interrupt Hardware Stack | 8 |
| | EEPROM Data Memory..... | 9 |
| | Instruction Execution Timing..... | 9 |
| | I/O Memory | 10 |
| | Reset and Interrupt Handling..... | 12 |
| | Sleep Modes..... | 19 |
| | Timer/Counter0..... | 20 |
| | Timer/Counter0 Prescaler..... | 20 |
| | Watchdog Timer..... | 23 |
| | EEPROM Read/Write Access..... | 25 |
| | Prevent EEPROM Corruption | 26 |
| | Analog Comparator | 27 |
| | I/O Ports..... | 29 |
| | Port B..... | 29 |
| | Port D..... | 34 |
| | Memory Programming..... | 37 |
| | Program and Data Memory Lock Bits..... | 37 |
| | Fuse Bits..... | 37 |
| | Signature Bytes | 37 |
| | Programming the Flash and EEPROM..... | 37 |
| | Parallel Programming | 38 |
| | Parallel Programming Characteristics | 43 |
| | Serial Downloading..... | 44 |
| | Serial Programming Characteristics | 47 |



| | |
|---|-----------|
| Electrical Characteristics | 48 |
| Absolute Maximum Ratings* | 48 |
| DC Characteristics | 48 |
| External Clock Drive Waveforms | 50 |
| External Clock Drive | 50 |
| Typical Characteristics | 51 |
| AT90S1200 Register Summary | 62 |
| Instruction Set Summary | 63 |
| Ordering Information⁽¹⁾ | 65 |
| Packaging Information | 66 |
| 20P3 | 66 |
| 20S | 67 |
| 20Y | 68 |
| Table of Contents | i |



Atmel Headquarters

Corporate Headquarters

2325 Orchard Parkway
San Jose, CA 95131
TEL 1(408) 441-0311
FAX 1(408) 487-2600

Europe

Atmel SarL
Route des Arsenaux 41
Casa Postale 80
CH-1705 Fribourg
Switzerland
TEL (41) 26-426-5555
FAX (41) 26-426-5500

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

Memory

Atmel Corporate
2325 Orchard Parkway
San Jose, CA 95131
TEL 1(408) 436-4270
FAX 1(408) 436-4314

Microcontrollers

Atmel Corporate
2325 Orchard Parkway
San Jose, CA 95131
TEL 1(408) 436-4270
FAX 1(408) 436-4314

Atmel Nantes
La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
TEL (33) 2-40-18-18-18
FAX (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards

Atmel Rousset
Zone Industrielle
13106 Rousset Cedex, France
TEL (33) 4-42-53-60-00
FAX (33) 4-42-53-60-01

Atmel Colorado Springs
1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL 1(719) 576-3300
FAX 1(719) 540-1759

Atmel Smart Card ICs
Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR, Scotland
TEL (44) 1355-803-000
FAX (44) 1355-242-743

RF/Automotive

Atmel Heilbronn
Theresienstrasse 2
Postfach 3535
74025 Heilbronn, Germany
TEL (49) 71-31-67-0
FAX (49) 71-31-67-2340

Atmel Colorado Springs
1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL 1(719) 576-3300
FAX 1(719) 540-1759

Biometrics/Imaging/Hi-Rel MPU/ High Speed Converters/RF Datacom

Atmel Grenoble
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
TEL (33) 4-76-58-30-00
FAX (33) 4-76-58-34-80

e-mail

literature@atmel.com

Web Site

<http://www.atmel.com>

© Atmel Corporation 2002.

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.

ATMEL® and AVR® are the registered trademarks of Atmel.

Other terms and product names may be the trademarks of others.



Printed on recycled paper.

Looking for pricing, stock, or lifecycle information?

Click below to explore more details on WIN SOURCE:

- ⊖ [View AT90S1200-4SI on WIN SOURCE](#)
- ⊖ [Microchip Technology Information](#)

Optimize Your Supply Chain with WIN SOURCE Solutions

- ✓ Global Sourcing Solution
- ✓ Obsolete Management
- ✓ Cost Control Management
- ✓ Shortage Management
- ✓ Alternative Solution
- ✓ Excess Inventory Management