



Welcome to [E-XFL.COM](https://www.e-xfl.com)

### What is "[Embedded - Microcontrollers](#)"?

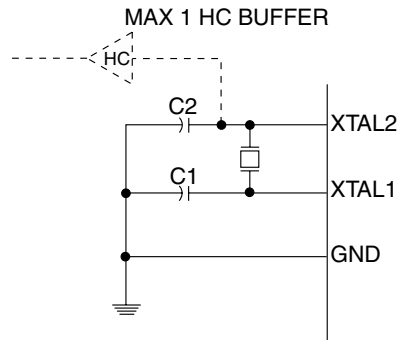
"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

### Applications of "[Embedded - Microcontrollers](#)"

#### Details

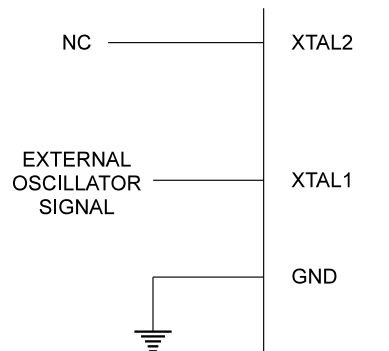
Product Status	Obsolete
Core Processor	AVR
Core Size	8-Bit
Speed	12MHz
Connectivity	SPI
Peripherals	POR, WDT
Number of I/O	15
Program Memory Size	1KB (512 x 16)
Program Memory Type	FLASH
EEPROM Size	64 x 8
RAM Size	-
Voltage - Supply (Vcc/Vdd)	4V ~ 6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C
Mounting Type	Through Hole
Package / Case	20-DIP (0.300", 7.62mm)
Supplier Device Package	20-PDIP
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/at90s1200a-12pc">https://www.e-xfl.com/product-detail/microchip-technology/at90s1200a-12pc</a>

**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



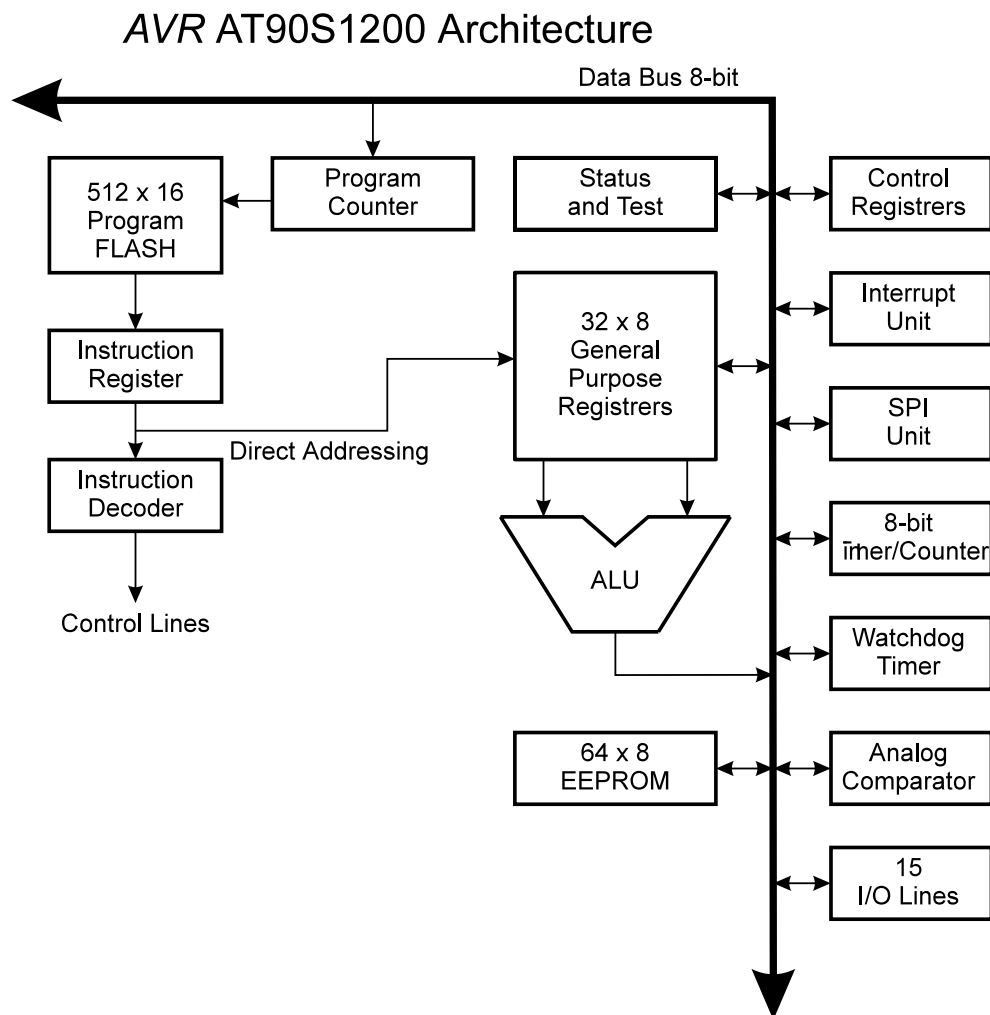
## On-chip RC Oscillator

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



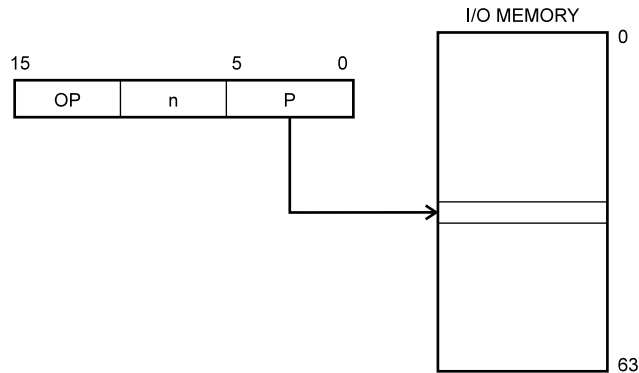
The ALU supports arithmetic and logic functions between registers or between a constant and a register. Single register operations are also executed in the ALU. Figure 4 shows the AT90S1200 AVR RISC microcontroller architecture. The AVR uses a Harvard architecture concept – with separate memories and 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.

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

## I/O Direct

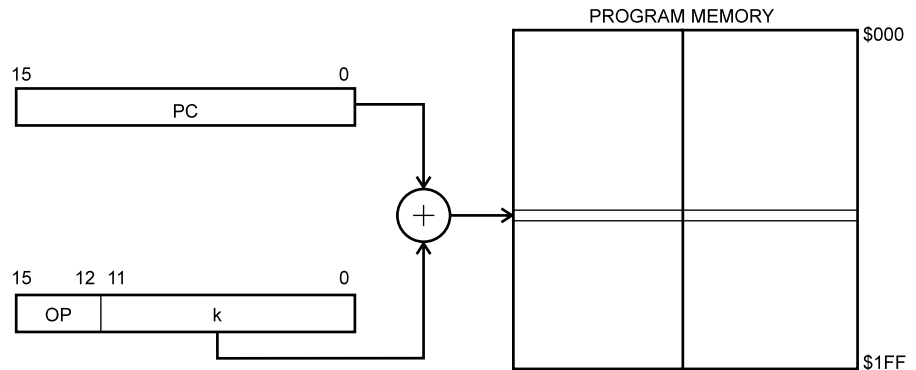
**Figure 9.** I/O Direct Addressing



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  $\phi$ , 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 pipeline 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

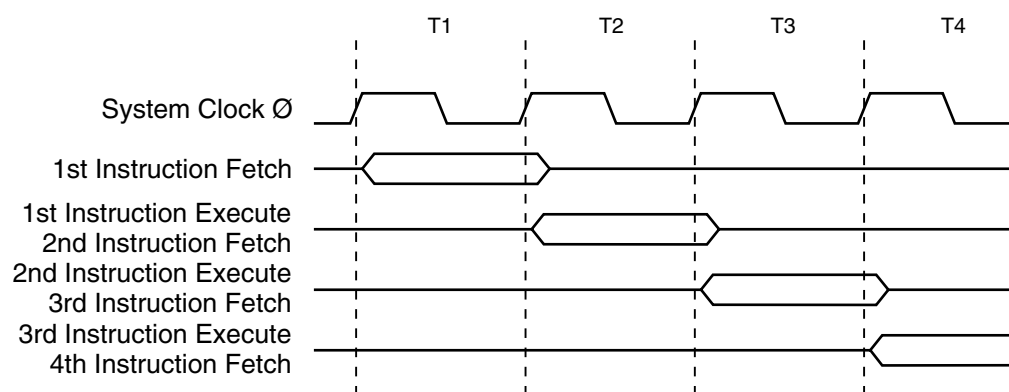
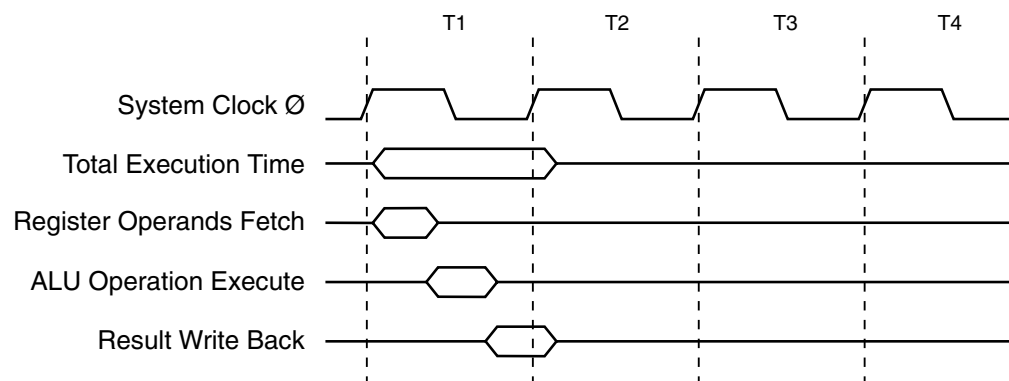


Figure 12 shows the internal timing concept for the register file. In a single clock cycle an ALU operation using two register operands is executed, and the result is stored back to the destination register.

**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**

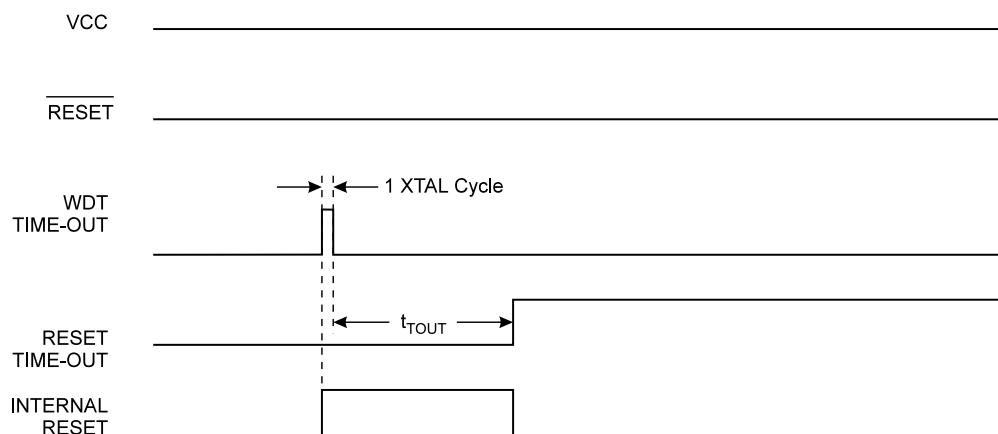
The zero flag Z indicates a zero result after the different arithmetic and logic operations. See the Instruction Set description for detailed information.

- **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.

**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.



## **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.

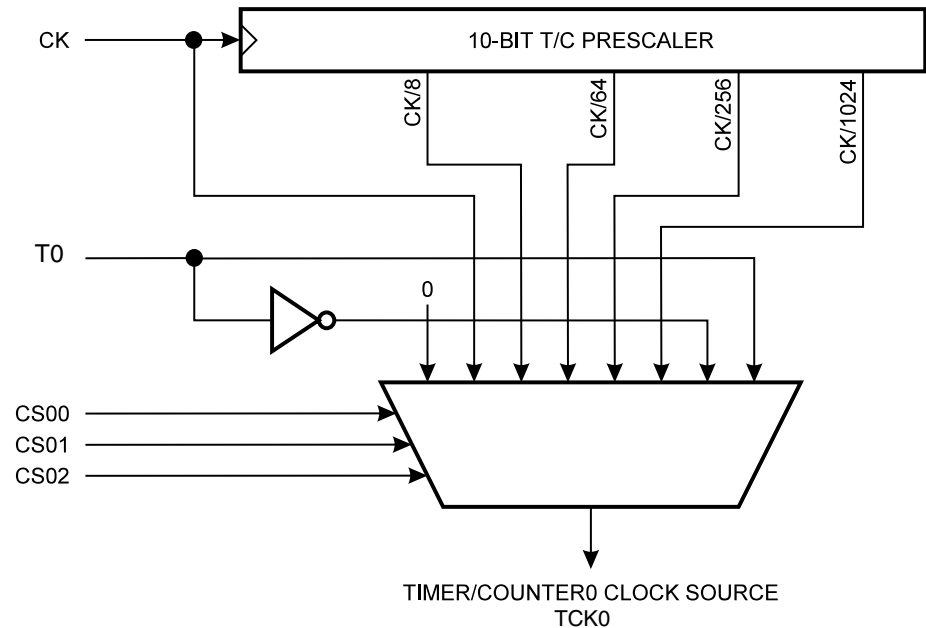
## Timer/Counter0

### Timer/Counter0 Prescaler

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.

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.

- **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	DDRB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

### 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.

the self-timed write instruction in the Serial Programming mode. During programming, the supply voltage must be in accordance with Table 13.

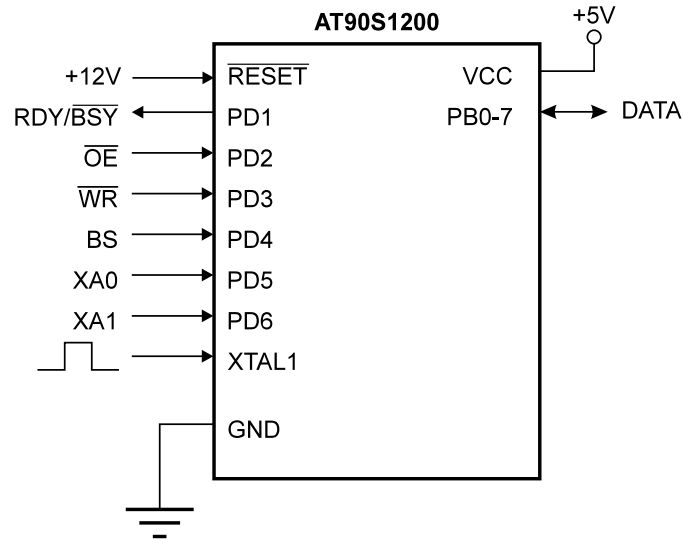
**Table 13.** Supply Voltage during Programming

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”

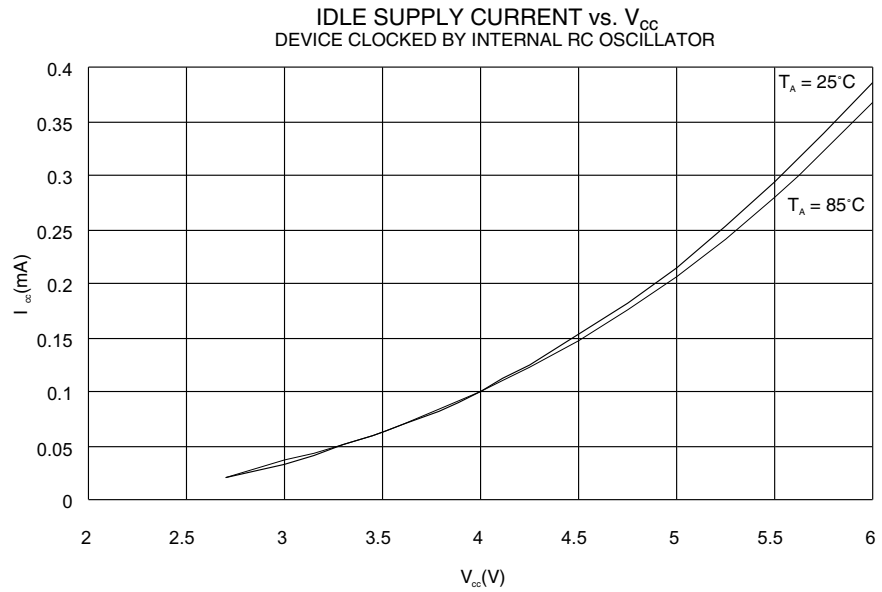
1. Set XA1, XA0 to “10”. This enables command loading.
2. Set BS to “0”.
3. Set DATA to “1000 0000”. This is the command for Chip Erase.
4. Give XTAL1 a positive pulse. This loads the command.
5. Give  $\overline{WR}$  a  $t_{WLWH\_CE}$  wide negative pulse to execute Chip Erase,  $t_{WLWH\_CE}$  is found in Table 17. Chip Erase does not generate any activity on the RDY/BSY pin.

## Programming the Flash

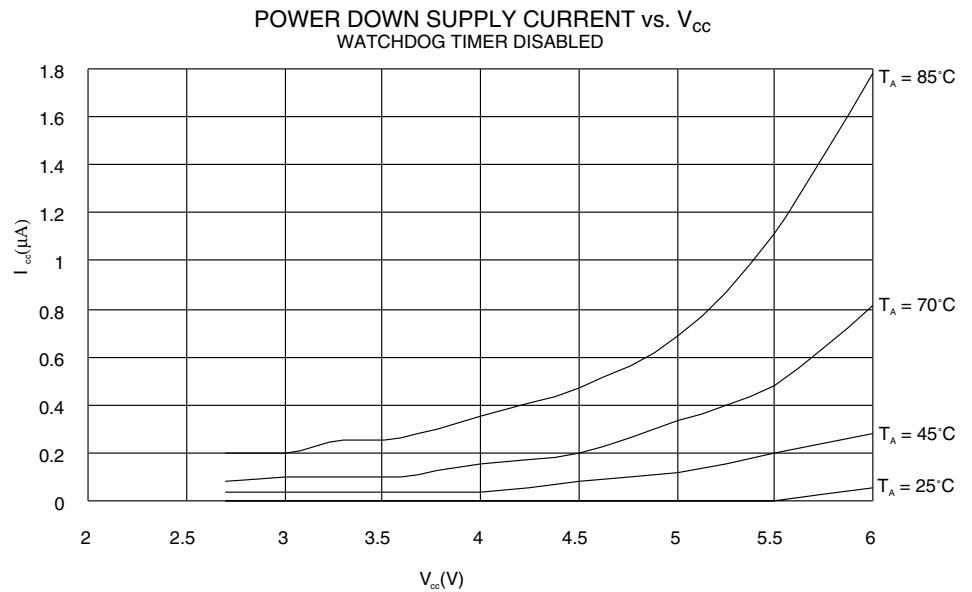
A: Load Command “Write Flash”

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.

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

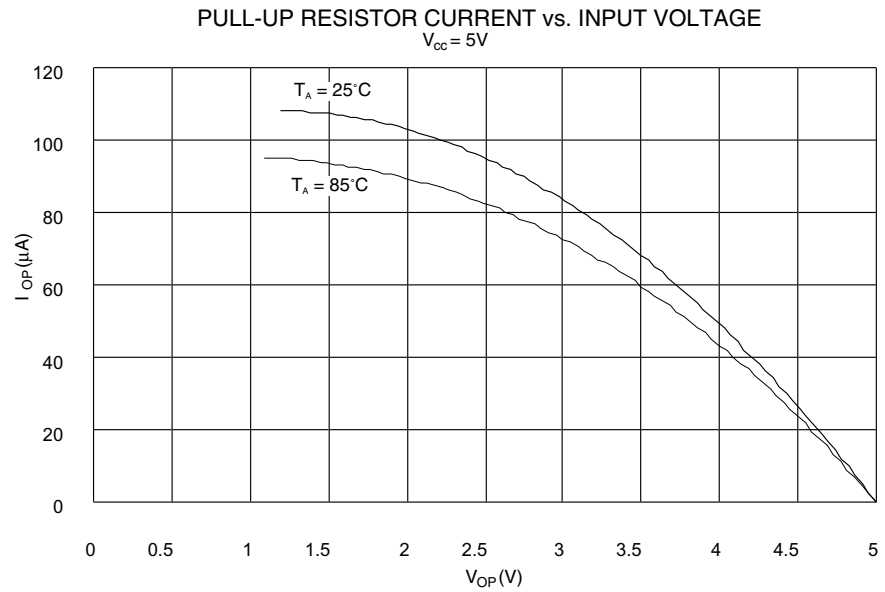


**Figure 44.** Power-down Supply Current vs.  $V_{CC}$ , Watchdog Timer Disabled

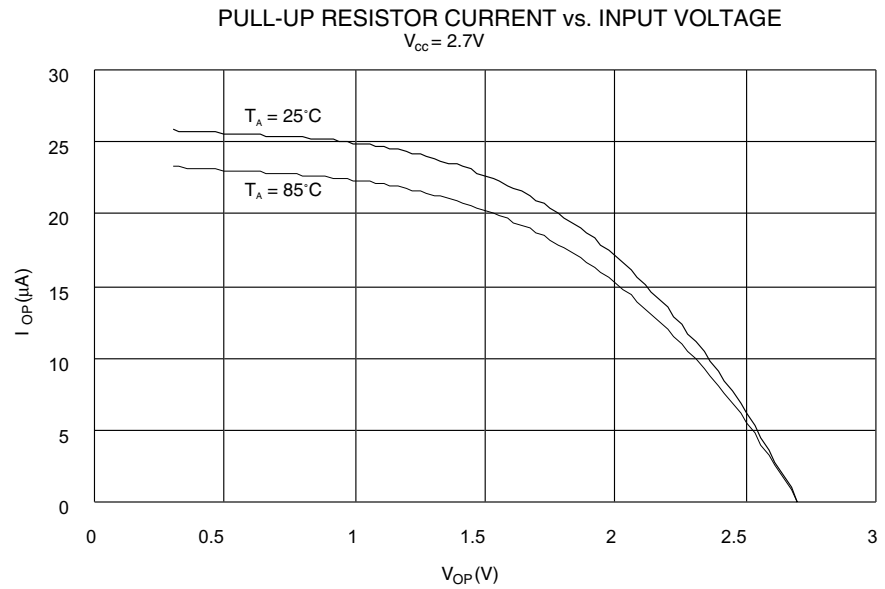


Note: Sink and source capabilities of I/O ports are measured on one pin at a time.

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

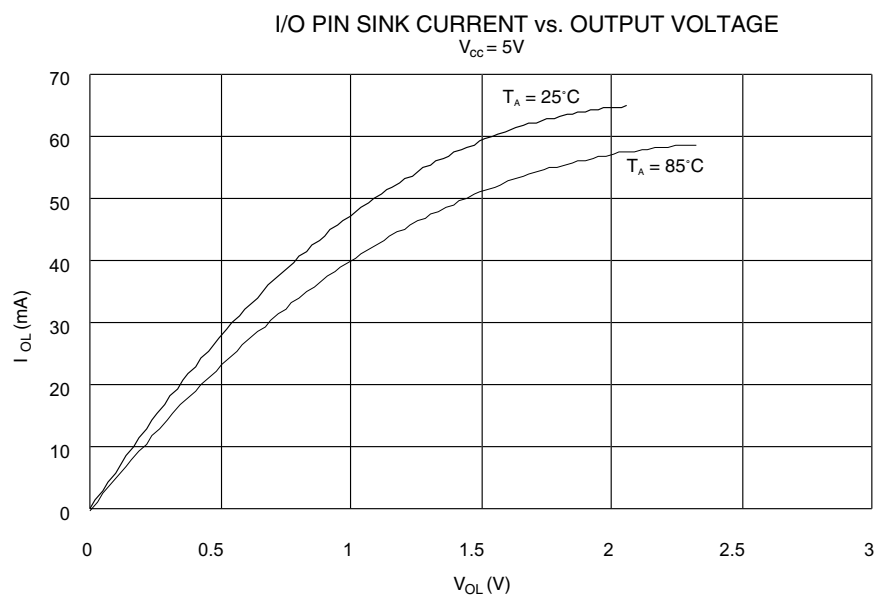


**Figure 52.** Pull-up Resistor Current vs. Input Voltage

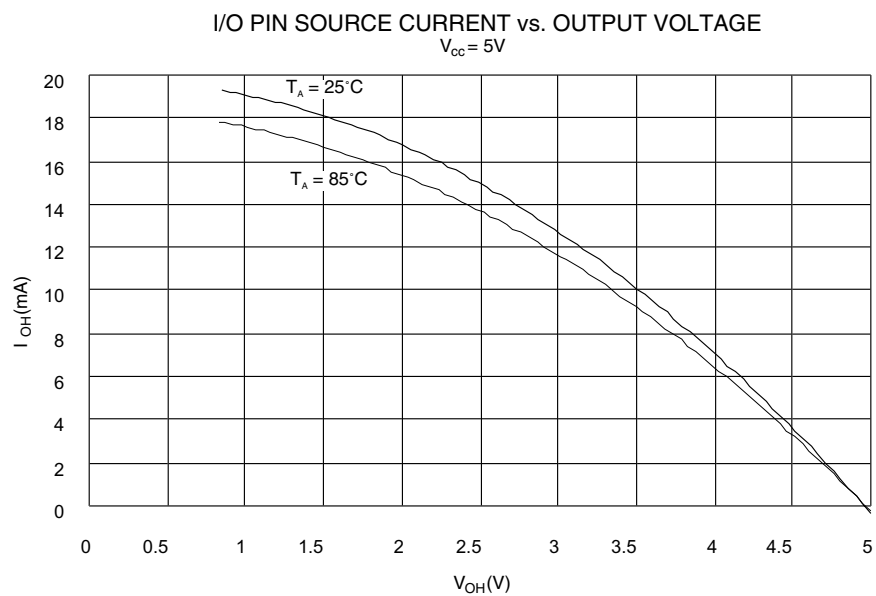




**Figure 53.** I/O Pin Sink Current vs. Output Voltage



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

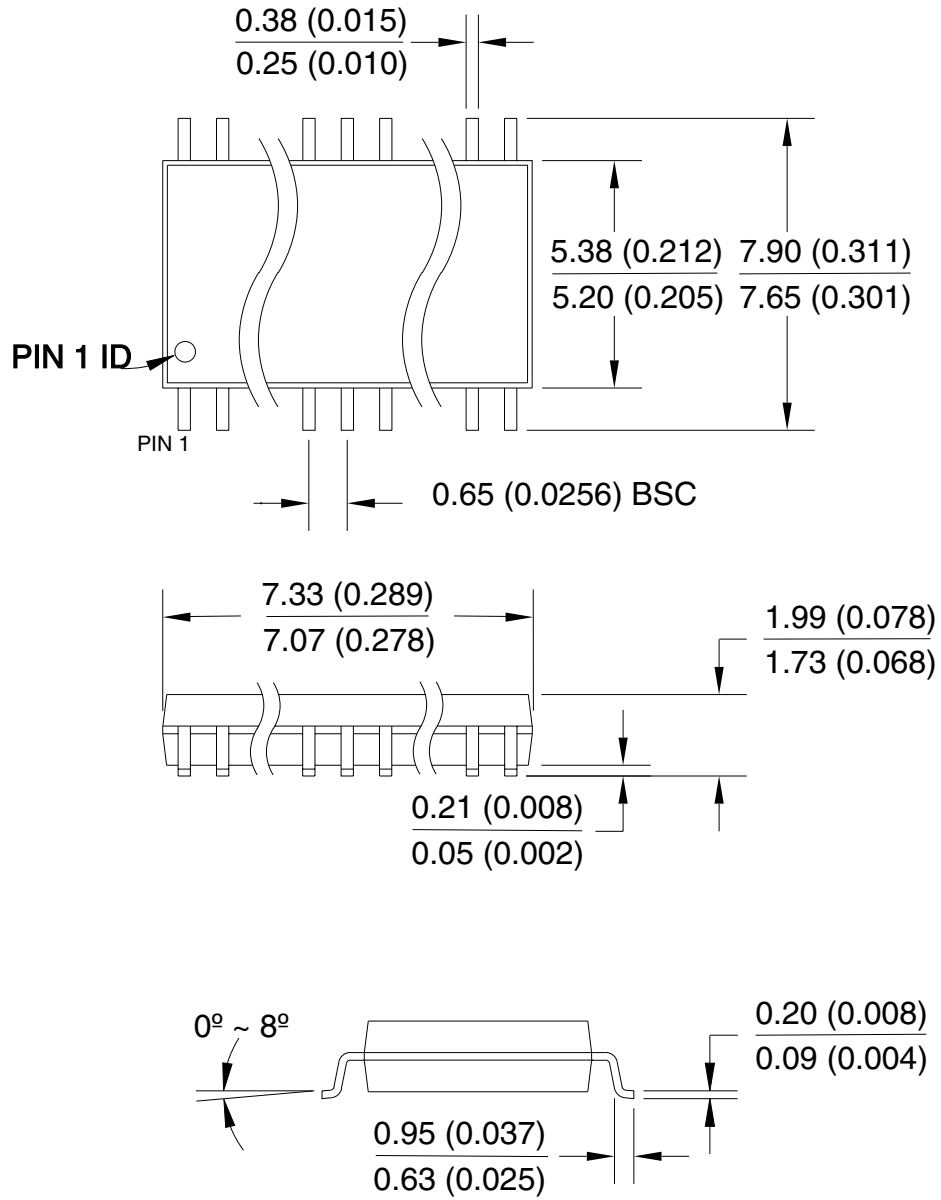


## Instruction Set Summary (Continued)

Mnemonic	Operands	Description	Operation	Flags	# Clocks
<b>BIT AND BIT-TEST INSTRUCTIONS</b>					
SBI	P, b	Set Bit in I/O Register	$I/O(P,b) \leftarrow 1$	None	2
CBI	P, b	Clear Bit in I/O Register	$I/O(P,b) \leftarrow 0$	None	2
LSL	Rd	Logical Shift Left	$Rd(n+1) \leftarrow Rd(n), Rd(0) \leftarrow 0$	Z,C,N,V	1
LSR	Rd	Logical Shift Right	$Rd(n) \leftarrow Rd(n+1), Rd(7) \leftarrow 0$	Z,C,N,V	1
ROL	Rd	Rotate Left through Carry	$Rd(0) \leftarrow C, Rd(n+1) \leftarrow Rd(n), C \leftarrow Rd(7)$	Z,C,N,V	1
ROR	Rd	Rotate Right through Carry	$Rd(7) \leftarrow C, Rd(n) \leftarrow Rd(n+1), C \leftarrow Rd(0)$	Z,C,N,V	1
ASR	Rd	Arithmetic Shift Right	$Rd(n) \leftarrow Rd(n+1), n = 0..6$	Z,C,N,V	1
SWAP	Rd	Swap Nibbles	$Rd(3..0) \leftarrow Rd(7..4), Rd(7..4) \leftarrow Rd(3..0)$	None	1
BSET	s	Flag Set	$SREG(s) \leftarrow 1$	SREG(s)	1
BCLR	s	Flag Clear	$SREG(s) \leftarrow 0$	SREG(s)	1
BST	Rr, b	Bit Store from Register to T	$T \leftarrow Rr(b)$	T	1
BLD	Rd, b	Bit Load from T to Register	$Rd(b) \leftarrow T$	None	1
SEC		Set Carry	$C \leftarrow 1$	C	1
CLC		Clear Carry	$C \leftarrow 0$	C	1
SEN		Set Negative Flag	$N \leftarrow 1$	N	1
CLN		Clear Negative Flag	$N \leftarrow 0$	N	1
SEZ		Set Zero Flag	$Z \leftarrow 1$	Z	1
CLZ		Clear Zero Flag	$Z \leftarrow 0$	Z	1
SEI		Global Interrupt Enable	$I \leftarrow 1$	I	1
CLI		Global Interrupt Disable	$I \leftarrow 0$	I	1
SES		Set Signed Test Flag	$S \leftarrow 1$	S	1
CLS		Clear Signed Test Flag	$S \leftarrow 0$	S	1
SEV		Set 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

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

<b>Features.....</b>	<b>1</b>
<b>Pin Configuration.....</b>	<b>1</b>
<b>Description.....</b>	<b>2</b>
Block Diagram .....	2
Pin Descriptions.....	3
Crystal Oscillator.....	3
On-chip RC Oscillator .....	4
<b>Architectural Overview.....</b>	<b>5</b>
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
<b>Timer/Counter0 .....</b>	<b>20</b>
Timer/Counter0 Prescaler.....	20
<b>Watchdog Timer.....</b>	<b>23</b>
<b>EEPROM Read/Write Access.....</b>	<b>25</b>
Prevent EEPROM Corruption .....	26
<b>Analog Comparator .....</b>	<b>27</b>
<b>I/O Ports.....</b>	<b>29</b>
Port B.....	29
Port D.....	34
<b>Memory Programming.....</b>	<b>37</b>
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



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