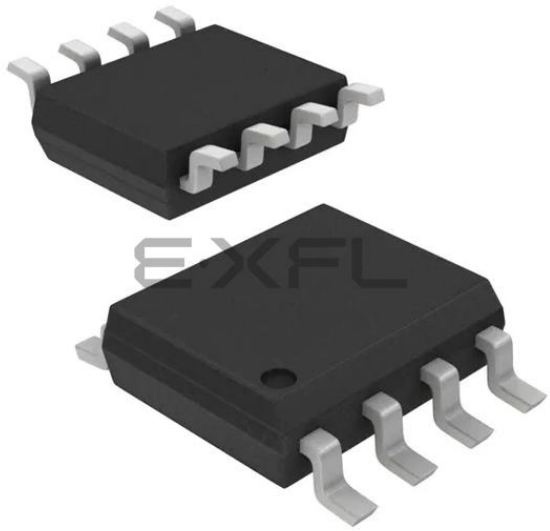


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	AVR
Core Size	8-Bit
Speed	10MHz
Connectivity	SPI
Peripherals	Brown-out Detect/Reset, POR, WDT
Number of I/O	5
Program Memory Size	2KB (1K x 16)
Program Memory Type	FLASH
EEPROM Size	128 x 8
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	4V ~ 6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C
Mounting Type	Surface Mount
Package / Case	8-SOIC (0.209", 5.30mm Width)
Supplier Device Package	8-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at90s2343-10si

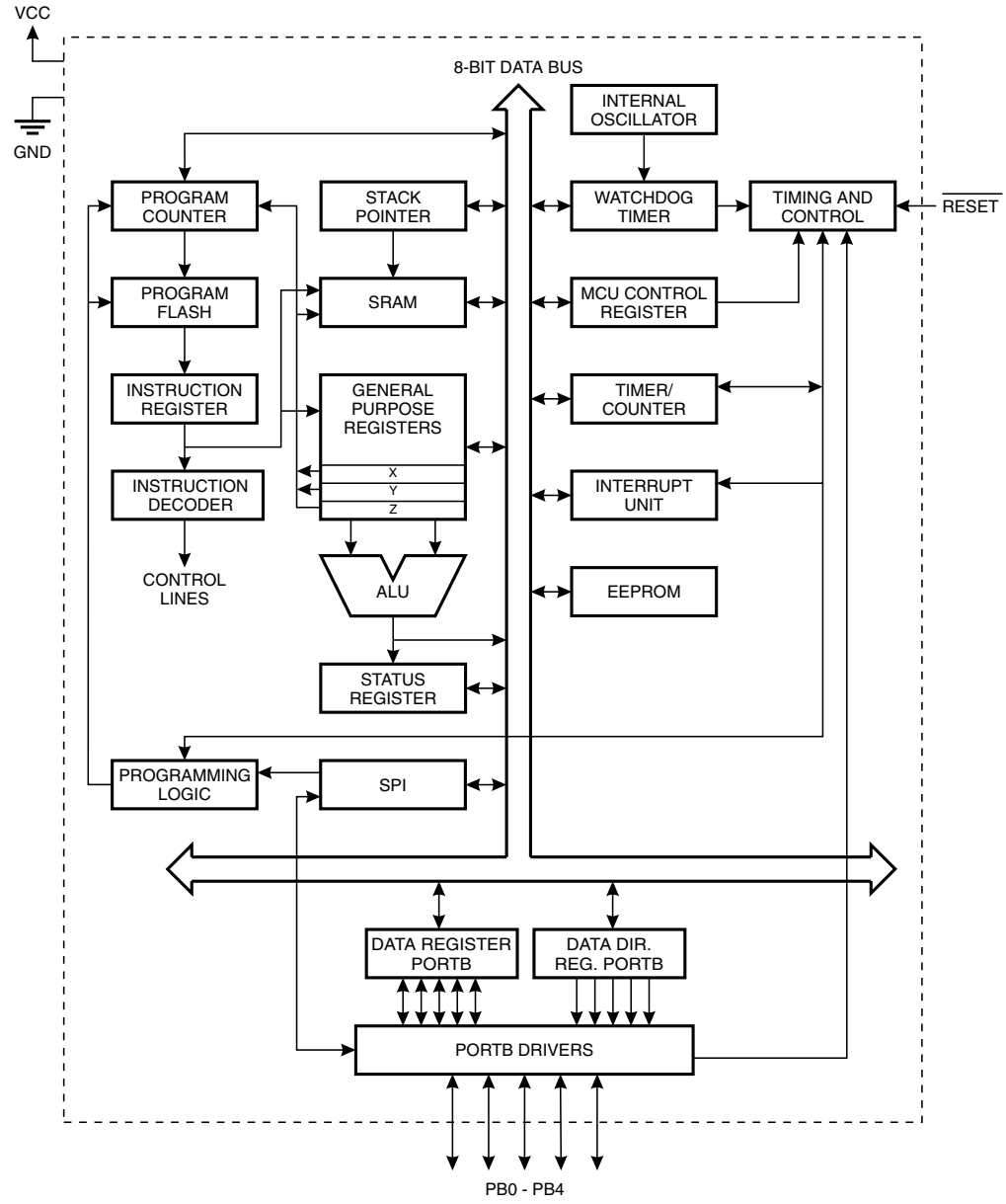
Description

The AT90S/LS2323 and AT90S/LS2343 are low-power, CMOS, 8-bit microcontrollers based on the AVR RISC architecture. By executing powerful instructions in a single clock cycle, the AT90S2323/2343 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

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

Block Diagram

Figure 1. The AT90S/LS2343 Block Diagram



Pin Descriptions
AT90S/LS2343

VCC Supply voltage pin.

GND Ground pin.

Port B (PB4..PB0) Port B is a 5-bit bi-directional I/O port with internal pull-up resistors. The Port B output buffers can sink 20 mA. As inputs, Port B pins that are externally pulled low, will source current if the pull-up resistors are activated.

Port B also serves the functions of various special features.

Port pins can provide internal pull-up resistors (selected for each bit). The Port B pins are tri-stated when a reset condition becomes active.

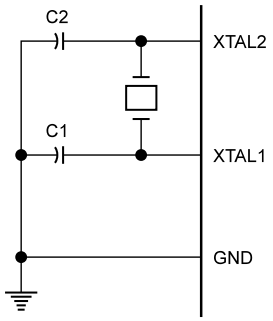
$\overline{\text{RESET}}$ Reset input. An external reset is generated by a low level on the $\overline{\text{RESET}}$ pin. Reset pulses longer than 50 ns will generate a reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a reset.

CLOCK Clock signal input in external clock mode.

Clock Options

Crystal Oscillator The AT90S/LS2323 contains an inverting amplifier that can be configured for use as an On-chip oscillator, as shown in Figure 3. XTAL1 and XTAL2 are input and output respectively. Either a quartz crystal or a ceramic resonator may be used. It is recommended that the AT90S/LS2343 be used if an external clock source is used, since this gives an extra I/O pin.

Figure 3. Oscillator Connection



External Clock The AT90S/LS2343 can be clocked by an external clock signal, as shown in Figure 4, or by the On-chip RC oscillator. This RC oscillator runs at a nominal frequency of 1 MHz (V_{CC} = 5V). A fuse bit (RCEN) in the Flash memory selects the On-chip RC oscillator as the clock source when programmed (“0”). The AT90S/LS2343 is shipped with this bit programmed. The AT90S/LS2343 is recommended if an external clock source is used, because this gives an extra I/O pin.

The AT90S/LS2323 can be clocked by an external clock as well, as shown in Figure 4. No fuse bit selects the clock source for AT90S/LS2323.

The AVR has Harvard architecture – with separate memories and buses for program and data. The program memory is accessed with a two-stage pipeline. While one instruction is being executed, the next instruction is pre-fetched from the program memory. This concept enables instructions to be executed in every clock cycle. The program memory is in-system downloadable Flash memory.

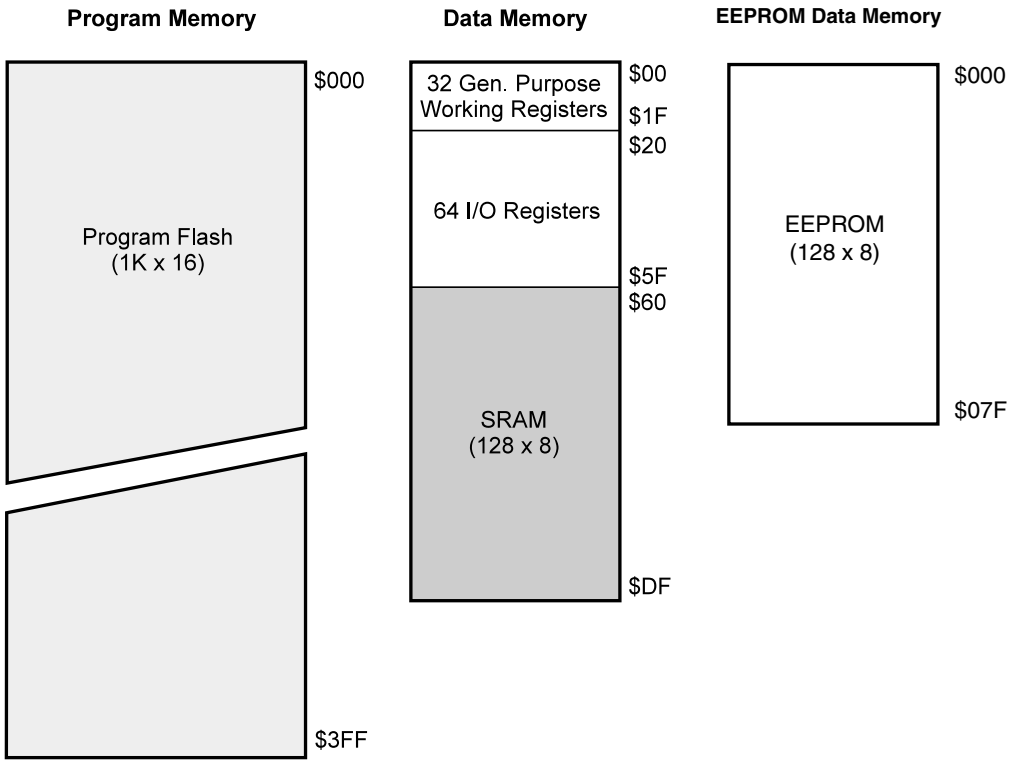
With the relative jump and call instructions, the whole 1K address space is directly accessed. Most AVR instructions have a single 16-bit word format. Every program memory address contains a 16- or 32-bit instruction.

During interrupts and subroutine calls, the return address Program Counter (PC) is stored on the stack. The stack is effectively allocated in the general data SRAM and consequently, the stack size is only limited by the total SRAM size and the usage of the SRAM. All user programs must initialize the SP in the reset routine (before subroutines or interrupts are executed). The 8-bit stack pointer (SP) is read/write-accessible in the I/O space.

The 128 bytes data SRAM + register file and I/O registers can be easily accessed through the five different addressing modes supported in the AVR architecture.

The memory spaces in the AVR architecture are all linear and regular memory maps.

Figure 6. Memory Maps

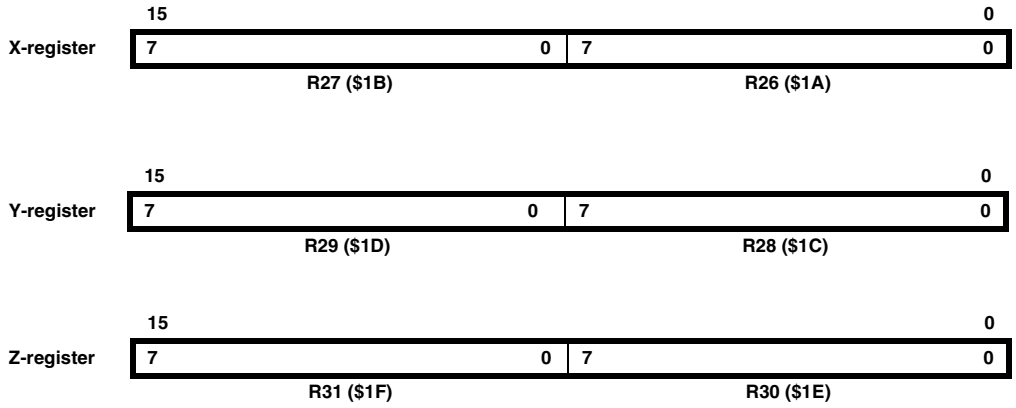


A flexible interrupt module has its control registers in the I/O space with an additional global interrupt enable bit in the status register. All the different interrupts have a separate interrupt vector in the interrupt vector table at the beginning of the program memory. The different interrupts have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

X-register, Y-register and Z-register

The registers R26..R31 have some added functions to their general-purpose usage. These registers are the address pointers for indirect addressing of the Data Space. The three indirect address registers X, Y, and Z, are defined in Figure 8.

Figure 8. The X-, Y-, and Z-registers



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

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 AT90S2323/2343 contains 2K bytes On-chip, In-System Programmable Flash memory for program storage. Since all instructions are 16- or 32-bit words, the Flash is organized as 1K x 16. The Flash memory has an endurance of at least 1000 write/erase cycles.

The AT90S2323/2343 Program Counter (PC) is 10 bits wide, hence addressing the 1024 program memory addresses. See page 42 for a detailed description on Flash data programming.

Constant tables must be allocated within the address 0 - 2K (see the LPM – Load Program Memory instruction description on page 60).

See page 12 for the different addressing modes.

EEPROM Data Memory

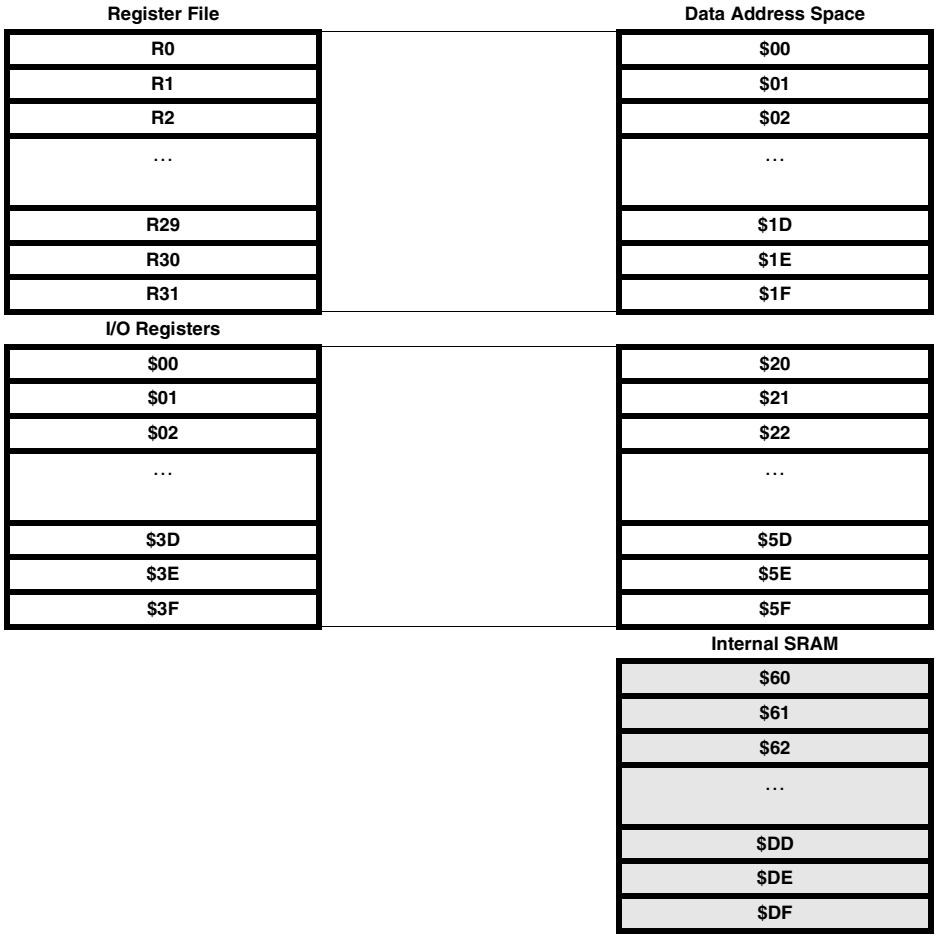
The AT90S2323/2343 contains 128 bytes of EEPROM data memory. It is organized as a separate data space, in which single bytes can be read and written. The EEPROM has an endurance of at least 100,000 write/erase cycles. The access between the EEPROM and the CPU is described on page 32, specifying the EEPROM address register, the EEPROM data register and the EEPROM control register.

For the SPI data downloading, see page 42 for a detailed description.

SRAM Data Memory

Figure 9 shows how the AT90S2323/2343 Data Memory is organized.

Figure 9. SRAM Organization



The 224 data memory locations address the Register file, I/O memory and the data SRAM. The first 96 locations address the Register file + I/O memory, and the next 128 locations address the data SRAM.

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

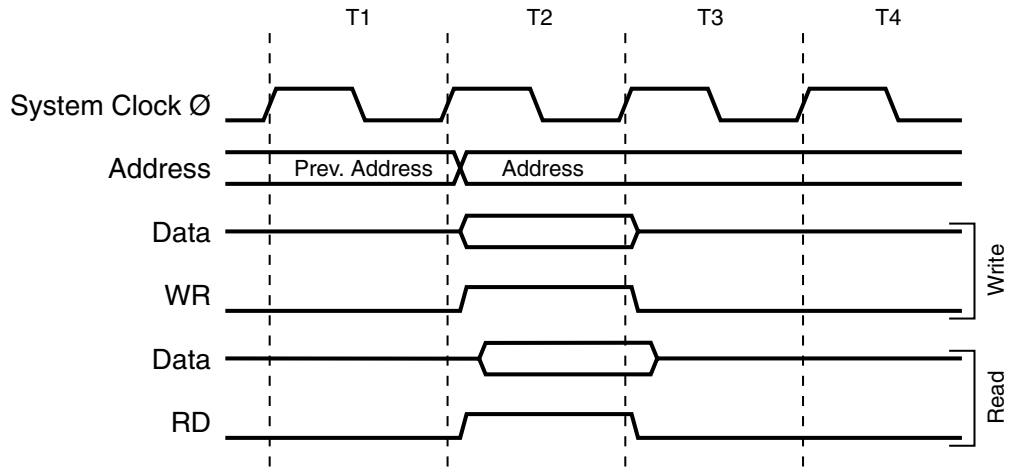
The direct addressing reaches the entire data address space.

The Indirect with Displacement mode features 63 address locations reached from the base address given by the Y- and Z-register.

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

The 32 general-purpose working registers, 64 I/O registers and the 128 bytes of data SRAM in the AT90S2323/2343 are all directly accessible through all these addressing modes.

Figure 23. On-chip Data SRAM Access Cycles



I/O Memory

The I/O space definition of the AT90S2323/2343 is shown in Table 2.

Table 2. AT90S2323/2343 I/O Space

Address Hex	Name	Function
$3F ($5F)	SREG	Status REGister
$3D ($5D)	SPL	Stack Pointer Low
$3B ($5B)	GIMSK	General Interrupt MaSK register
$3A ($5A)	GIFR	General Interrupt Flag Register
$39 ($59)	TIMSK	Timer/Counter Interrupt MaSK register
$38 ($58)	TIFR	Timer/Counter Interrupt Flag register
$35 ($55)	MCUCR	MCU Control Register
$34 ($54)	MCUSR	MCU Status Register
$33 ($53)	TCCR0	Timer/Counter0 Control Register
$32 ($52)	TCNT0	Timer/Counter0 (8-bit)
$21 ($41)	WDTCR	Watchdog Timer Control Register
$1E ($3E)	EEAR	EEPROM Address Register
$1D ($3D)	EEDR	EEPROM Data Register
$1C ($3C)	EECR	EEPROM Control Register
$18 ($38)	PORTB	Data Register, Port B
$17 ($37)	DDRB	Data Direction Register, Port B
$16 ($36)	PINB	Input Pins, Port B

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

All AT90S2323/2343 I/Os and peripherals are placed in the I/O space. The I/O locations are accessed by the IN and OUT instructions transferring data between the 32 general-purpose working registers and the I/O space. I/O registers within the address range $00 - $1F are directly bit-accessible using the SBI and CBI instructions. In these registers, the value of single bits can be checked by using the SBIS and SBIC instructions. Refer to the instruction set section for more details. When using the I/O-specific commands IN

The most typical program setup for the Reset and Interrupt vector addresses are:

```
Address     Labels      Code                    Comments
$000                    rjmp RESET              ; Reset Handler
$001                    rjmp EXT_INT0           ; IRQ0 Handler
$002                    rjmp TIM_OVF0           ; Timer0 Overflow
                                                ; Handler;
$003        MAIN:       ldi r16, low(RAMEND)    ; Main program start
                        out SPL, r16
                        <instr> xxx
...         ...         ...                     ...
```

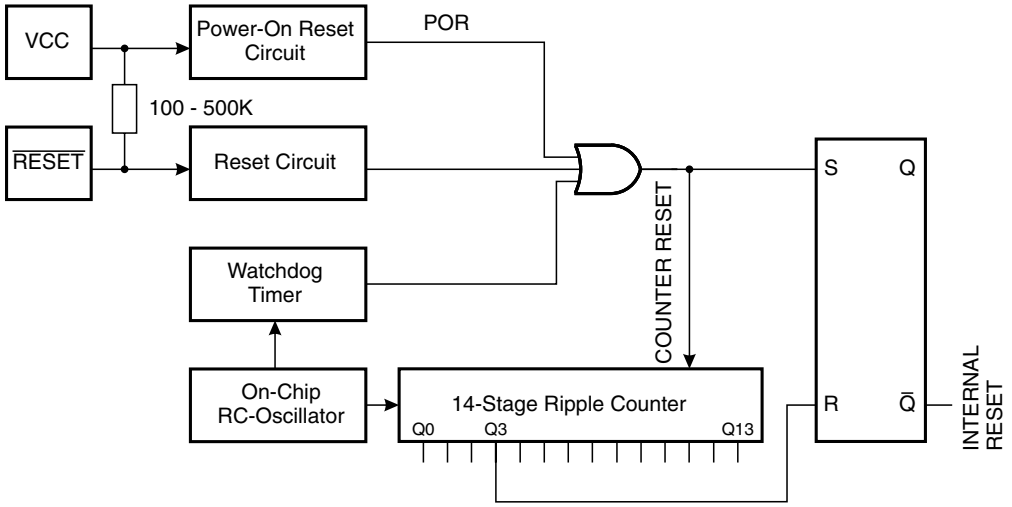
Reset Sources

The AT90S2323/2343 provides 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 set to their initial values and the program starts execution from address $000. The instruction placed in address $000 must be an RJMP (relative jump) instruction to the reset handling routine. If the program never enables an interrupt source, the interrupt vectors are not used and regular program code can be placed at these locations. The circuit diagram in Figure 24 shows the reset logic. Table 4 defines the timing and electrical parameters of the reset circuitry.

Figure 24. Reset Logic



The AT90S/LS2323 has a programmable start-up time. A fuse bit (FSTRT) in the Flash memory selects the shortest start-up time when programmed (“0”). The AT90S/LS2323 is shipped with this bit unprogrammed.

The AT90S/LS2343 has a fixed start-up time.

Figure 25. MCU Start-up, $\overline{\text{RESET}}$ Tied to V_{CC}.

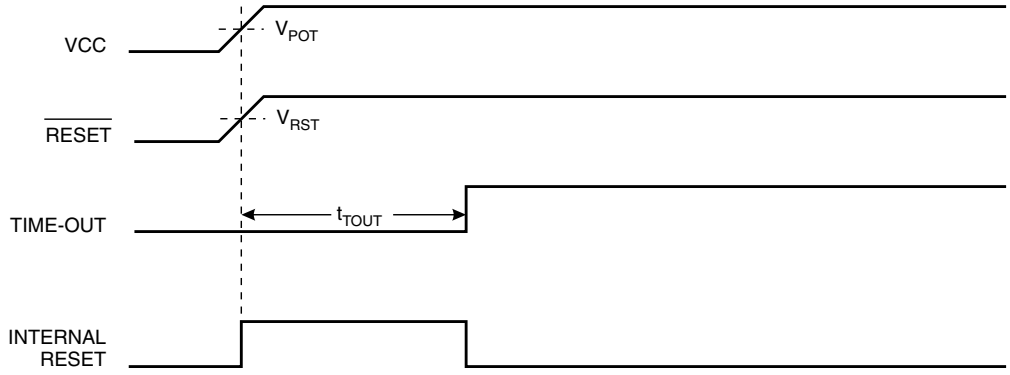
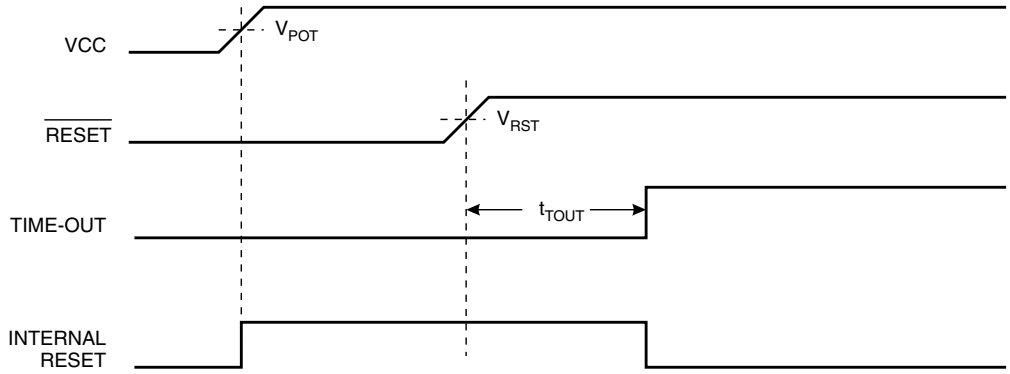


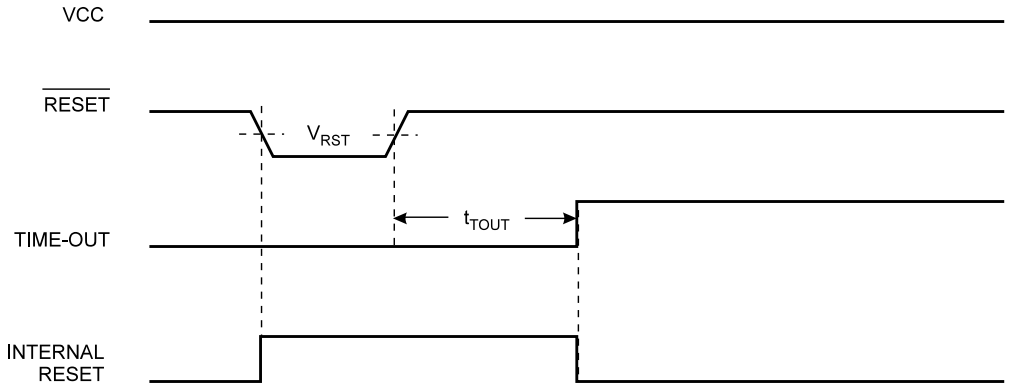
Figure 26. MCU Start-up, $\overline{\text{RESET}}$ Controlled Externally



External Reset

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

Figure 27. External Reset during Operation



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 PB2/(T0) will clock the counter even if the pin is configured as an output. This feature can give the user software control of the counting.

Timer/Counter0 – TCNT0

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

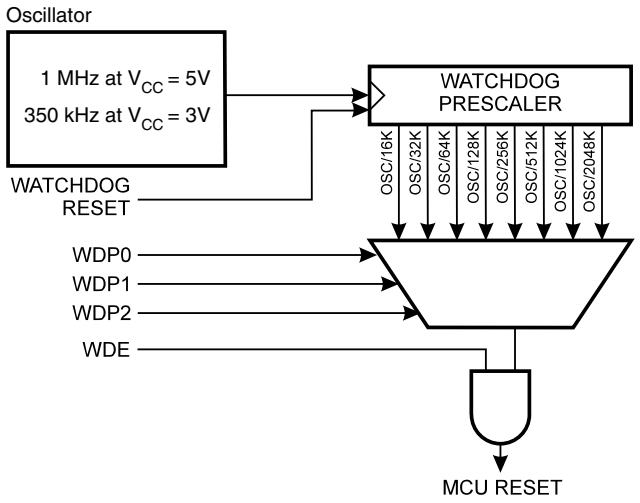
The Timer/Counter0 is realized as an up-counter with read and write access. If the Timer/Counter0 is written and a clock source is present, the Timer/Counter0 continues counting in the timer clock cycle following the write operation.

Watchdog Timer

The Watchdog Timer is clocked from a separate On-chip oscillator. By controlling the Watchdog Timer prescaler, the Watchdog reset interval can be adjusted as shown in Table 11. See characterization data for typical values at other V_{CC} levels. The WDR (Watchdog Reset) instruction resets the Watchdog Timer. Eight different clock cycle periods can be selected to determine the reset period. If the reset period expires without another Watchdog reset, the AT90S2323/2343 resets and executes from the reset vector. For timing details on the Watchdog reset, refer to page 23.

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

Figure 31. Watchdog Timer



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.

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

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

EEPROM Address Register – EEAR

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

- **Bit 7 – Res: Reserved Bit**

This bit is a reserved bit in the AT90S2323/2343 and will always read as zero.

- **Bit 6..0 – EEAR6..0: EEPROM Address**

The EEPROM Address Register (EEAR6..0) specifies the EEPROM address in the 128-byte EEPROM space. The EEPROM data bytes are addressed linearly between 0 and 127.

EEPROM Data Register – EEDR

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

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

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

EEPROM Control Register – EECR

Bit	7	6	5	4	3	2	1	0	
$1C ($3C)	–	–	–	–	–	EEMWE	EEWE	EERE	EECR
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7..3 – Res: Reserved Bits**

These bits are reserved bits in the AT90S2323/2343 and will always read as zero.

- **Bit 2 – EEMWE: EEPROM Master Write Enable**

The EEMWE bit determines whether setting EEWE to “1” causes the EEPROM to be written. When EEMWE is set (one), setting EEWE will write data to the EEPROM at the selected address. If EEMWE is zero, setting EEWE will have no effect. When EEMWE has been set (one) by software, hardware clears the bit to zero after four clock cycles. See the description of the EEWE bit for a EEPROM write procedure.

- **Bit 1 – EEWE: EEPROM Write Enable**

The EEPROM Write Enable signal (EEWE) is the write strobe to the EEPROM. When address and data are correctly set up, the EEWE bit must be set to write the value into the EEPROM. The EEMWE bit must be set when the logical “1” is written to EEWE, otherwise no EEPROM write takes place. The following procedure should be followed when writing the EEPROM (the order of steps 2 and 3 is unessential):

1. Wait until EEWE becomes zero.
2. Write new EEPROM address to EEAR (optional).
3. Write new EEPROM data to EEDR (optional).
4. Write a logical “1” to the EEMWE bit in EECR (to be able to write a logical “1” to the EEMWE bit, the EEWE bit must be written to “0” in the same cycle).
5. Within four clock cycles after setting EEMWE, write a logical “1” to EEWE.

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

When the write access time (typically 2.5 ms at V_{CC} = 5V or 4 ms at V_{CC} = 2.7V) has elapsed, the EEWE bit is cleared (zero) by hardware. The user software can poll this bit and wait for a zero before writing the next byte. When EEWE has been set, the CPU is halted for two cycles before the next instruction is executed.

- **Bit 0 – EERE: EEPROM Read Enable**

The EEPROM Read Enable signal (EERE) is the read strobe to the EEPROM. When the correct address is set up in the EEAR register, the EERE bit must be set. When the EERE bit is cleared (zero) by hardware, requested data is found in the EEDR register. The EEPROM read access takes one instruction and there is no need to poll the EERE bit. When EERE has been set, the CPU is halted for four cycles before the next instruction is executed.

The user should poll the EEWE bit before starting the read operation. If a write operation is in progress when new data or address is written to the EEPROM I/O registers, the write operation will be interrupted and the result is undefined.

I/O Port B

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).

For the AT90S/LS2323, Port B is an 3-bit bi-directional I/O port. For the AT90S/LS2343, Port B is a 5-bit bi-directional I/O port.

Please note: Bits 3 and 4 in the description of PORTB, DDRB and PINB do not apply to the AT90S/LS2323. They are read only with a value of 0.

Three I/O memory address locations are allocated for Port B, one each for the Data Register – PORTB, $18 ($38), Data Direction Register – DDRB, $17($37) and the Port B Input Pins – PINB, $16($36). The Port B Input Pins address is read-only, while the Data Register and the Data Direction Register are read/write.

All port pins have individually selectable pull-up resistors. The Port B output buffers can sink 20 mA and thus drive LED displays directly. When pins PB0 to PB4 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 12.

Table 12. Port B Pin Alternate Functions

Port Pin	Alternate Functions
PB0	MOSI (Data input line for memory downloading)
PB1	MISO (Data output line for memory uploading) INT0 (External Interrupt0 Input)
PB2	SCK (Serial clock input for serial programming) TO (Timer/Counter0 counter clock input)
PB3	CLOCK (Clock input, AT90S/LS2343 only)

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

Port B Data Register – PORTB

Bit	7	6	5	4	3	2	1	0	
$18 ($38)	–	–	–	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	PORTB
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Port B Data Direction Register – DDRB

Bit	7	6	5	4	3	2	1	0	
$17 ($37)	–	–	–	DDB4	DDB3	DDB2	DDB1	DDB0	DDRB
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

High-voltage Serial Programming Algorithm

To program and verify the AT90S/LS2323 and AT90S/LS234 in the high-voltage Serial Programming mode, the following sequence is recommended (see instruction formats in Table 16):

1. Power-up sequence: Apply 4.5 - 5.5V between V_{CC} and GND. Set $\overline{\text{RESET}}$ and PB0 to “0” and wait at least 100 ns. Then, if the RCEN Fuse is not programmed, toggle XTAL1/PB3 at least four times with minimum 100 ns pulse width. Set PB3 to “0”. Wait at least 100 ns. Or, if the RCEN Fuse is programmed, set PB3 to “0”. Wait for least 4 µs. In both cases, apply 12V to $\overline{\text{RESET}}$ and wait at least 100 ns before changing PB0. Wait 8 µs before giving any instructions.
2. The Flash array is programmed one byte at a time by supplying first the address, then the low and high data bytes. The write instruction is self-timed; wait until the PB2 (RDY/$\overline{\text{BSY}}$) pin goes high.
3. The EEPROM array is programmed one byte at a time by supplying first the address, then the data byte. The write instruction is self-timed; wait until the PB2 (RDY/$\overline{\text{BSY}}$) pin goes high.
4. Any memory location can be verified by using the Read instruction, which returns the contents at the selected address at serial output PB2.
5. Power-off sequence:Set PB3 to “0”.
   Set $\overline{\text{RESET}}$ to “0”.
   Turn V_{CC} power off.

When writing or reading serial data to the device, data is clocked on the rising edge of the serial clock. See Figure 33, Figure 34 and Table 17 for details.

Figure 33. High-voltage Serial Programming Waveforms

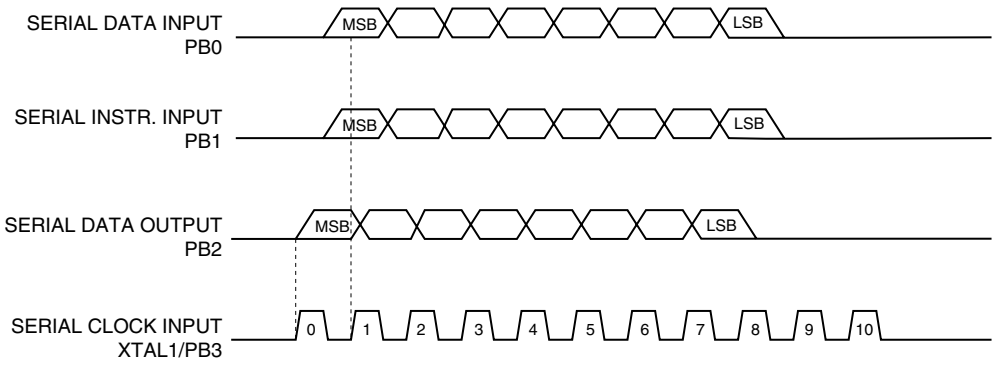


Table 16. High-voltage Serial Programming Instruction Set (Continued)

Instruction		Instruction Format				Operation Remarks
		Instr.1	Instr.2	Instr.3	Instr.4	
Read Fuse and Lock Bits (AT90S/ LS2323)	PB0 PB1 PB2	0_0000_0100_00 0_0100_1100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0111_1000_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0111_1100_00 ***1_2*****S**xx_xx**R**x_xx		Reading ***1***, ***2***, **S**, **R** = "0" means the Fuse/Lock bit is programmed.
Read Fuse and Lock Bits (AT90S/ LS2343)	PB0 PB1 PB2	0_0000_0100_00 0_0100_1100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0111_1000_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0111_1100_00 ***1_2*****S**xx_xx**R**x_xx		Reading ***1***, ***2***, **S**, **R** = "0" means the Fuse/Lock bit is programmed.
Read Signature Bytes	PB0 PB1 PB2	0_0000_1000_00 0_0100_1100_00 x_xxxx_xxxx_xx	0_0000_00**bb**_00 0_0000_1100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_1000_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_1100_00 **o_oooo_ooo**x_xx	Repeat Instr.2 - Instr.4 for each signature byte address.

Note: **a** = address high bits
b = address low bits
i = data in
o = data out
x = don't care
1 = Lock Bit1
2 = Lock Bit2
F = FSTRT Fuse
R = RCEN Fuse
S = SPIEN Fuse

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

The program and EEPROM memory arrays have separate address spaces: $0000 to $03FF for Flash program memory and $000 to $07F for EEPROM data memory.

Either an external clock is applied to the XTAL1/PB3 pin or the device must be clocked from the internal RC oscillator (AT90S/LS2343 only). The minimum low and high periods for the serial clock (SCK) input are defined as follows:

Low: > 2 MCU clock cycles

High: > 2 MCU clock cycles

Low-voltage Serial Programming Algorithm

When writing serial data to the AT90S2323/2343, data is clocked on the rising edge of SCK.

When reading data from the AT90S2323/2343, data is clocked on the falling edge of SCK. See Figure 36, Figure 37 and Table 20 for timing details.

To program and verify the AT90S2323/2343 in the low-voltage Serial Programming mode, the following sequence is recommended (see 4-byte instruction formats in Table 19):

1. Power-up sequence:

   Apply power between V_{CC} and GND while RESET and SCK are set to “0”. (If the programmer cannot guarantee that SCK is held low during power-up, RESET must be given a positive pulse after SCK has been set to “0”.) If the device is programmed for external clocking, apply a 0 - 8 MHz clock to the XTAL1/PB3 pin. If the internal RC oscillator is selected as the clock source, no external clock source needs to be applied (AT90S/LS2343 only).
2. Wait for at least 20 ms and enable serial programming by sending the Programming Enable serial instruction to the MOSI (PB0) pin. Refer to the above section for minimum low and high periods for the serial clock input, SCK.
3. The serial programming instructions will not work if the communication is out of synchronization. When in sync, the second byte ($53) will echo back when issuing the third byte of the Programming Enable instruction. Whether the echo is correct or not, all four bytes of the instruction must be transmitted. If the $53 did not echo back, give SCK a positive pulse and issue a new Programming Enable instruction. If the $53 is not seen within 32 attempts, there is no functional device connected.
4. If a Chip Erase is performed (must be done to erase the Flash), wait t_{WD_ERASE} after the instruction, give $\overline{\text{RESET}}$ a positive pulse and start over from step 2. See Table 21 on page 46 for t_{WD_ERASE} value.
5. The Flash or EEPROM array is programmed one byte at a time by supplying the address and data together with the appropriate Write instruction. An EEPROM memory location is first automatically erased before new data is written. Use Data Polling to detect when the next byte in the Flash or EEPROM can be written. If polling is not used, wait t_{WD_PROG} before transmitting the next instruction. See Table 22 on page 46 for t_{WD_PROG} value. In an erased device, no $FFs in the data file(s) need to be programmed.
6. Any memory location can be verified by using the Read instruction, which returns the content at the selected address at the serial output MISO (PB1) pin.

Table 19. Low-voltage Serial Programming Instruction Set AT90S2323/2343

Instruction	Instruction Format				Operation
	Byte 1	Byte 2	Byte 3	Byte 4	
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 **H**000	0000 00**aa**	**bbbb bbbb**	**oooo oooo**	Read **H** (high or low) data **o** from program memory at word address **a**:**b**.
Write Program Memory	0100 **H**000	0000 00**aa**	**bbbb bbbb**	**iiii iiii**	Write **H** (high or low) data **i** to program memory at word address **a**:**b**.
Read EEPROM Memory	1010 0000	0000 0000	x**bbb bbbb**	**oooo oooo**	Read data **o** from EEPROM memory at address **b**.
Write EEPROM Memory	1100 0000	0000 0000	x**bbb bbbb**	**iiii iiii**	Write data **i** to EEPROM memory at address **b**.
Read Lock and Fuse Bits (AT90S/LS2323)	0101 1000	xxxx xxxx	xxxx xxxx	***12S***x xxx**F**	Read Lock and Fuse bits. “0” = programmed, “1” = unprogrammed
Read Lock and Fuse Bits (AT90S/LS2343)	0101 1000	xxxx xxxx	xxxx xxxx	***12S***x xxx**R**	Read Lock and Fuse bits. “0” = programmed, “1” = unprogrammed
Write Lock Bits	1010 1100	1111 1***21***1	xxxx xxxx	xxxx xxxx	Write Lock bits. Set bits ***1***,***2*** = “0” to program Lock bits.
Write FSTRT Bit (AT90S/LS2323)	1010 1100	1011 111**F**	xxxx xxxx	xxxx xxxx	Write FSTRT fuse. Set bit **F** = “0” to program, “1” to unprogram.(2)
Write RCEN Bit (AT90S/LS2343)	1010 1100	1011 111**R**	xxxx xxxx	xxxx xxxx	Write RCEN Fuse. Set bit **R** = ‘0’ to program, ‘1’ to unprogram.(2)
Read Signature Bytes	0011 0000	xxxx xxxx	xxxx xx**bb**	**oooo oooo**	Read signature byte **o** from address **b**.(3)

Notes: 1. **a** = address high bits
b = address low bits
H = 0 – Low byte, 1 – High byte
o = data out
i = data in
x = don’t care
1 = lock bit 1
2 = lock bit 2
F = FSTRT Fuse
R = RCEN Fuse
S = SPIEN Fuse
2. When the state of the RCEN/FSTRT bit is changed, the device must be power cycled for the changes to have any effect.
3. The signature bytes are not readable in Lock mode 3, i.e., both Lock bits programmed.

Figure 40. Active Supply Current vs. V_{CC}

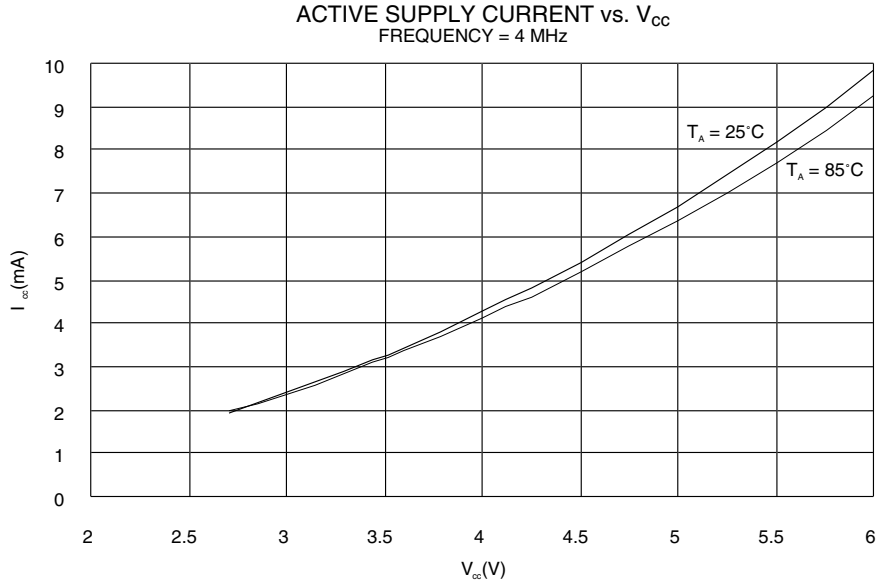


Figure 41. Active Supply Current vs. V_{CC}

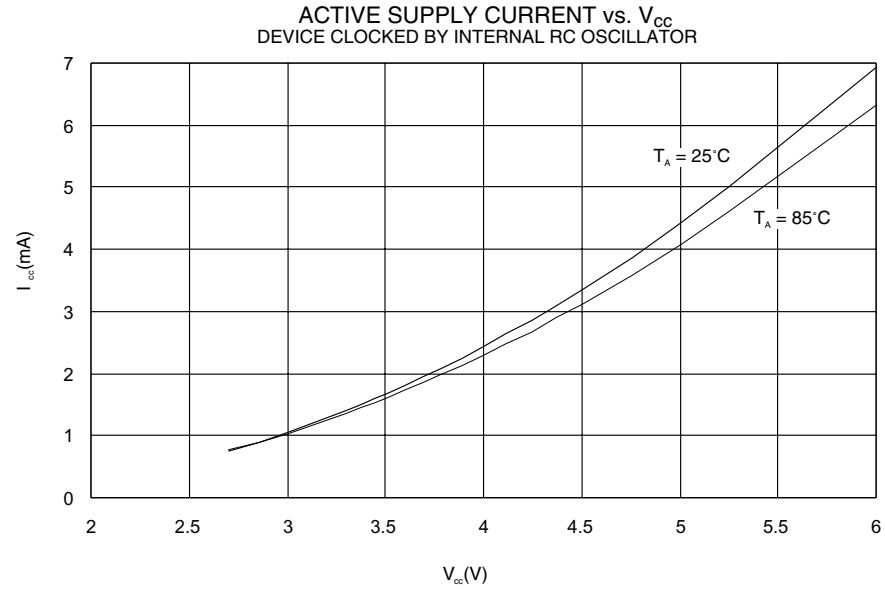


Figure 44. Idle Supply Current vs. V_{CC}

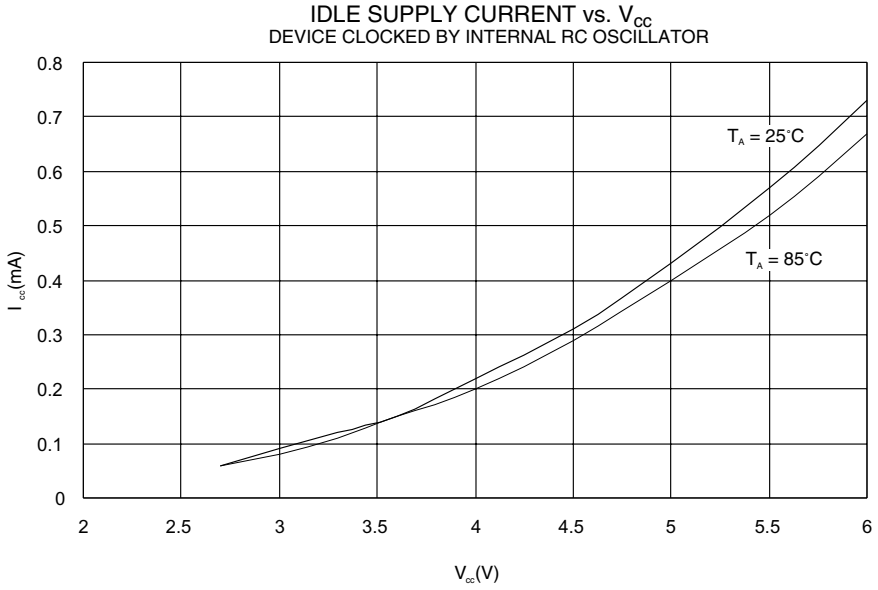


Figure 45. Power-down Supply Current vs. V_{CC}

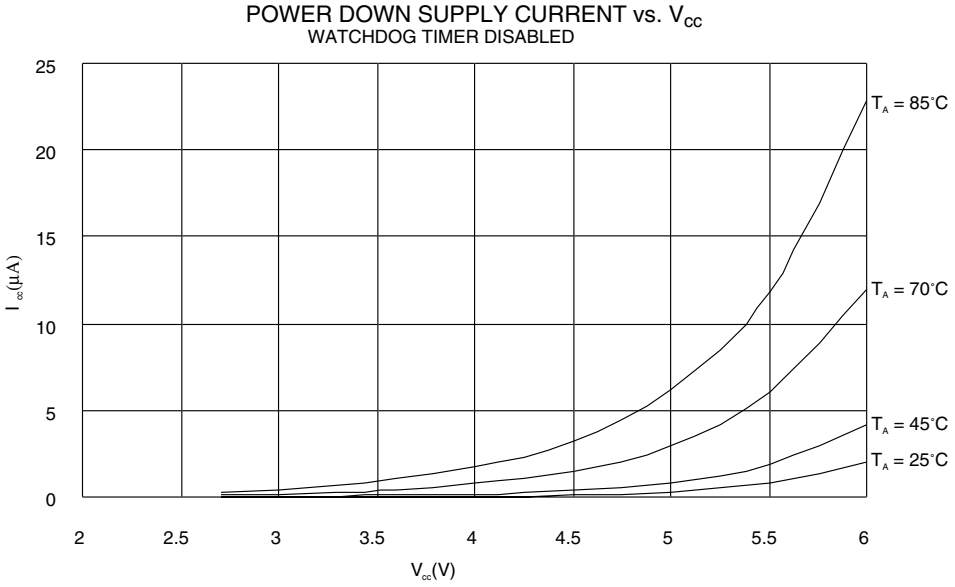


Figure 50. I/O Pin Sink Current vs. Output Voltage

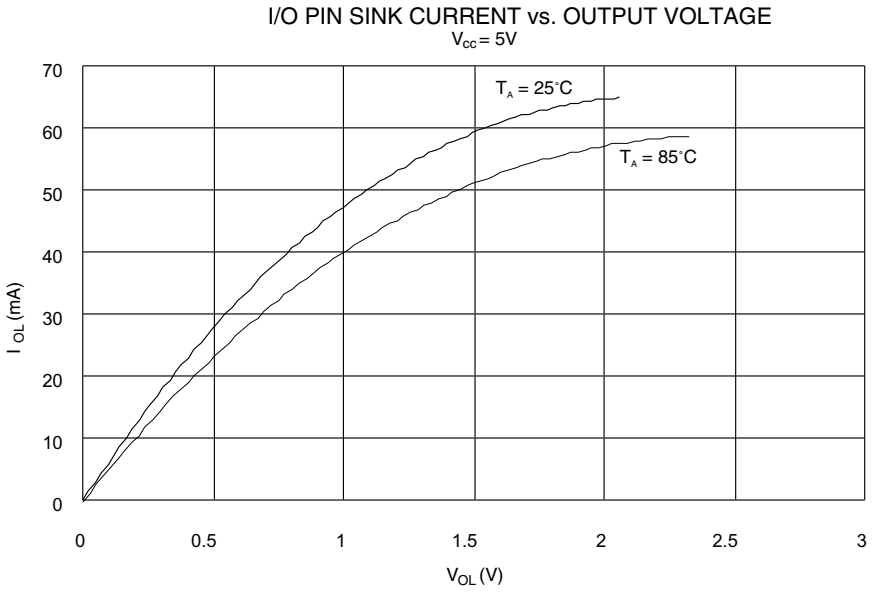
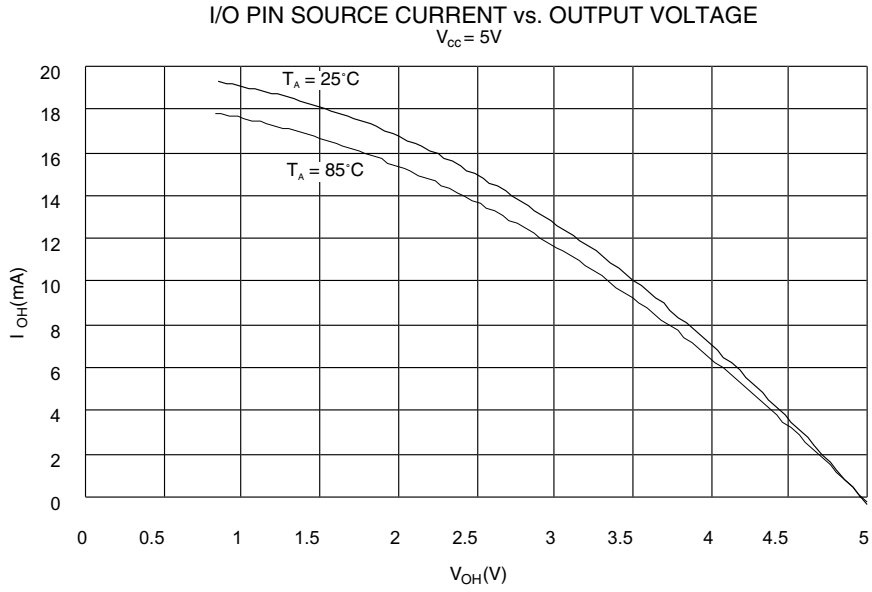


Figure 51. I/O PIn Source Current vs. Output Voltage



Instruction Set Summary

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