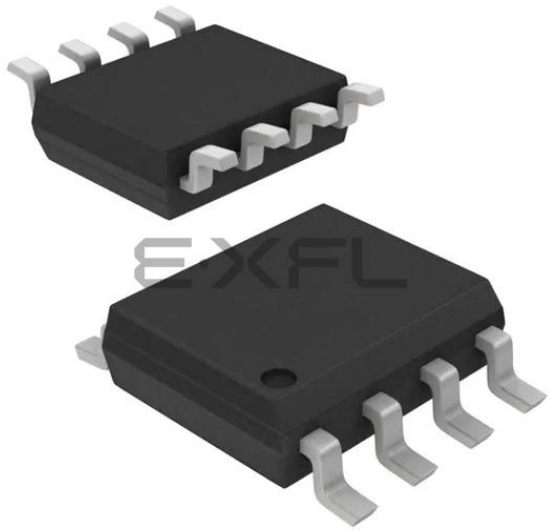


Welcome to E-XFL.COM

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	3
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	0°C ~ 70°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/at90s2323-10sc

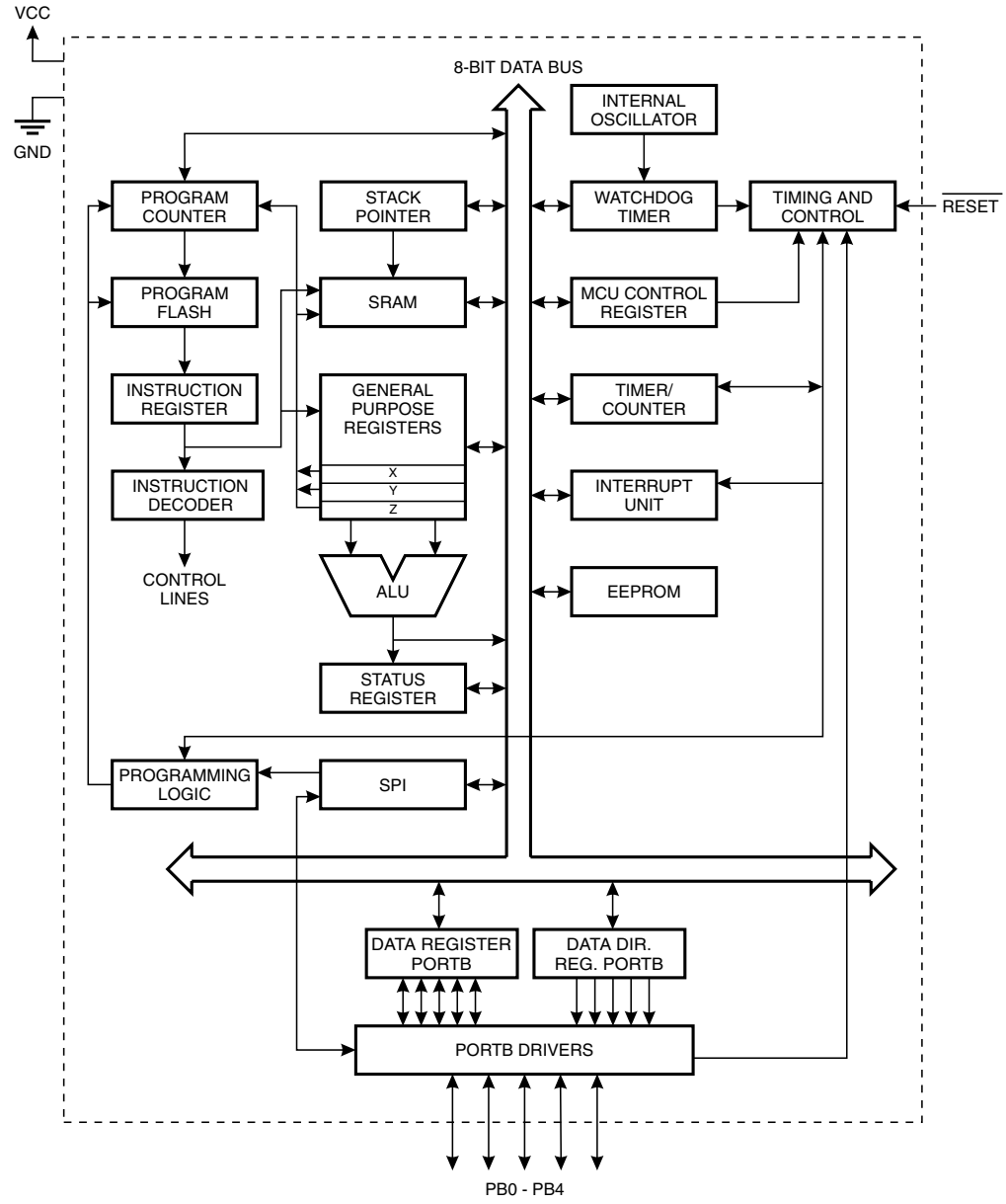
Description

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



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.

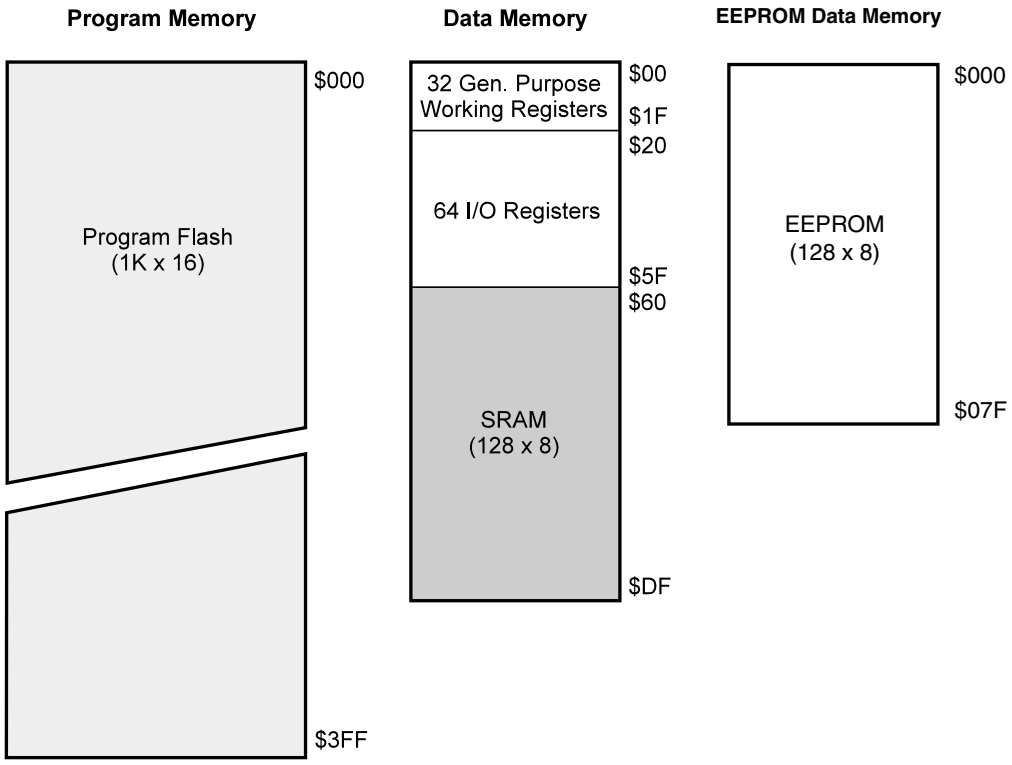
With the relative jump and call instructions, the whole 1K address space is directly accessed. Most AVR instructions have a single 16-bit word format. Every program memory address contains a 16- or 32-bit instruction.

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

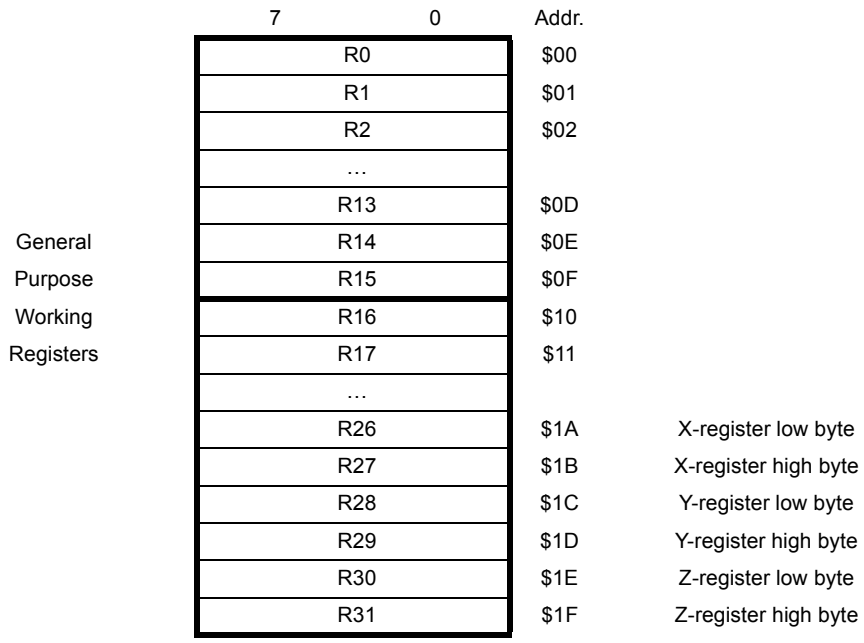


A flexible interrupt module has its control registers in the I/O space with an additional global interrupt enable bit in the status register. All the different interrupts have a separate interrupt vector in the interrupt vector table at the beginning of the program memory. The different interrupts have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

General-purpose Register File

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

Figure 7. AVR CPU General-purpose Working Registers



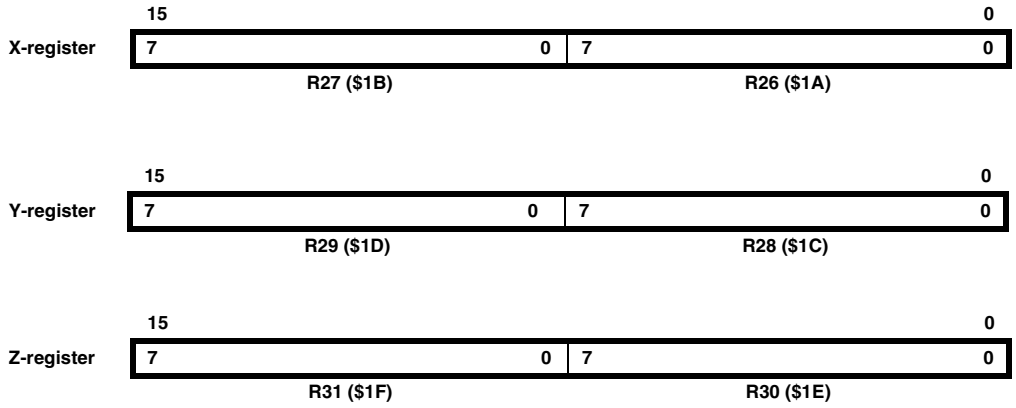
All the register operating instructions in the instruction set have direct and single-cycle access to all registers. The only exception is the five constant arithmetic and logic instructions SBCI, SUBI, CPI, ANDI and ORI between a constant and a register and the LDI instruction for load immediate constant data. These instructions apply to the second half of the registers in the register file (R16..R31). The general SBC, SUB, CP, AND and OR and all other operations between two registers or on a single register apply to the entire register file.

As shown in Figure 7, each register is also assigned a data memory address, mapping them directly into the first 32 locations of the user Data Space. Although the register file is not physically implemented as SRAM locations, this memory organization provides great flexibility in access of the registers, as the X-, Y-, and Z-registers can be set to index any register in the file.

X-register, Y-register and Z-register

The registers R26..R31 have some added functions to their general-purpose usage. These registers are 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.

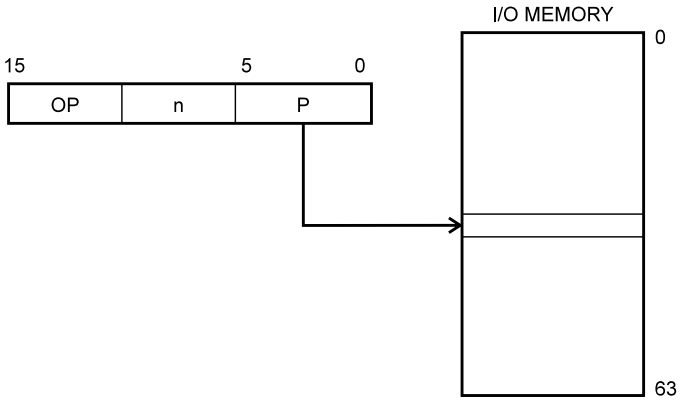
EEPROM Data Memory

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.

I/O Direct

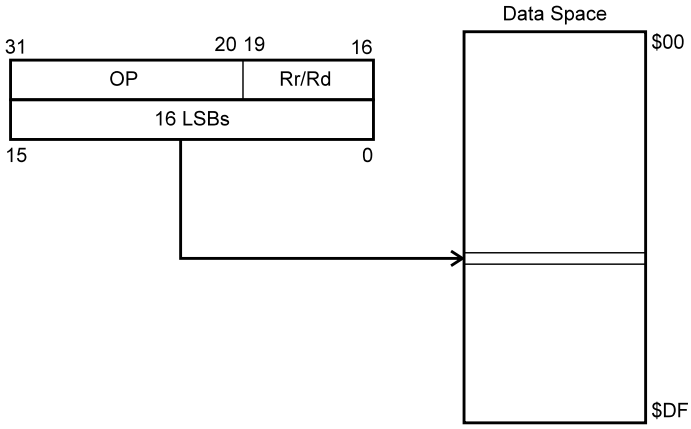
Figure 12. I/O Direct Addressing



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

Data Direct

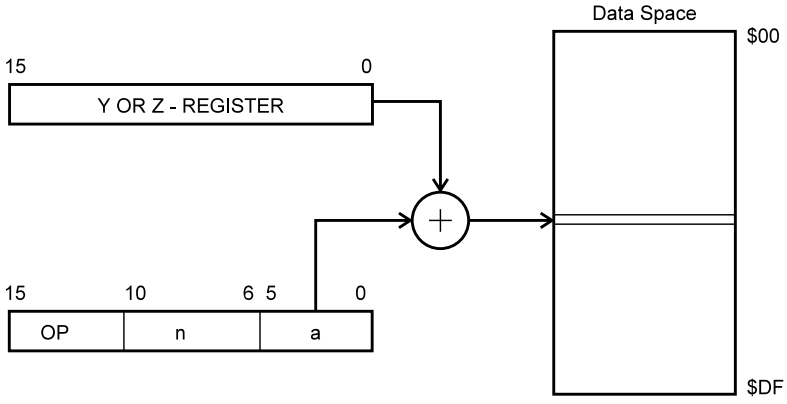
Figure 13. Direct Data Addressing



A 16-bit data address is contained in the 16 LSBs of a 2-word instruction. Rd/Rr specify the destination or source register.

Data Indirect with Displacement

Figure 14. Data Indirect with Displacement



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

Memory Access and Instruction Execution Timing

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

The AVR CPU is driven by the System Clock Ø, directly generated from the external clock signal applied to the CLOCK pin. No internal clock division is used.

Figure 21. shows the parallel instruction fetches and instruction executions enabled by the Harvard architecture and the fast-access register file concept. This is the basic pipelining concept to obtain up to 1 MIPS per MHz with the corresponding unique results for functions per cost, functions per clocks and functions per power unit.

Figure 21. The Parallel Instruction Fetches and Instruction Executions

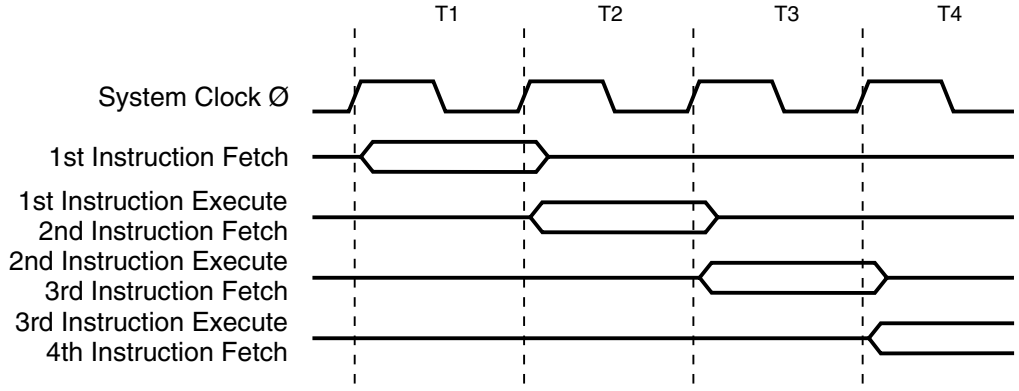
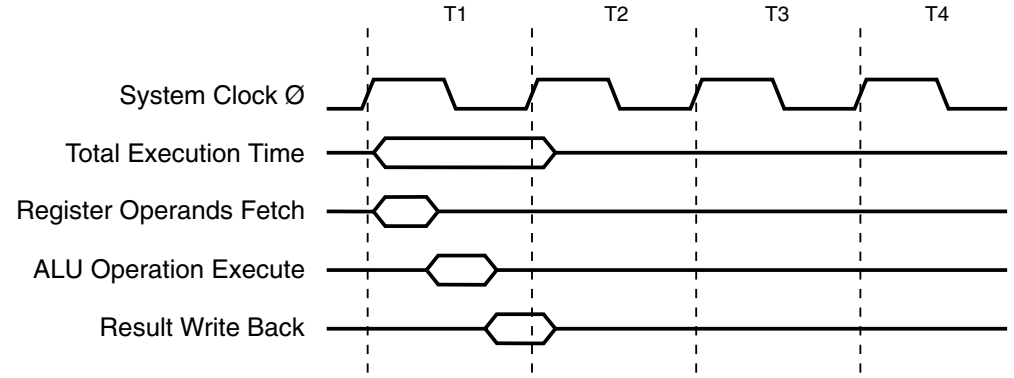


Figure 22. shows the internal timing concept for the register file. In a single clock cycle an ALU operation using two register operands is executed and the result is stored back to the destination register.

Figure 22. Single Cycle ALU Operation



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

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

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

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.

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

Status Register – SREG

The AVR Status Register (SREG) at I/O space location $3F ($5F) is defined as:

Bit	7	6	5	4	3	2	1	0	
$3F ($5F)	**I**	**T**	**H**	**S**	**V**	**N**	**Z**	**C**	**SREG**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

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

The global interrupt enable bit must be set (one) for the interrupts to be enabled. The individual interrupt enable control is then performed in separate control registers. If the global interrupt enable register is cleared (zero), none of the interrupts are enabled independent of the individual interrupt enable settings. The I-bit is cleared by hardware after an interrupt has occurred and is set by the RETI instruction to enable subsequent interrupts.

- **Bit 6 – T: Bit Copy Storage**

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

- **Bit 5 – H: Half-carry Flag**

The half-carry flag H indicates a half-carry in some 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 from an arithmetical or logical operation. See the Instruction Set description for detailed information.

- **Bit 1 – Z: Zero Flag**

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

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

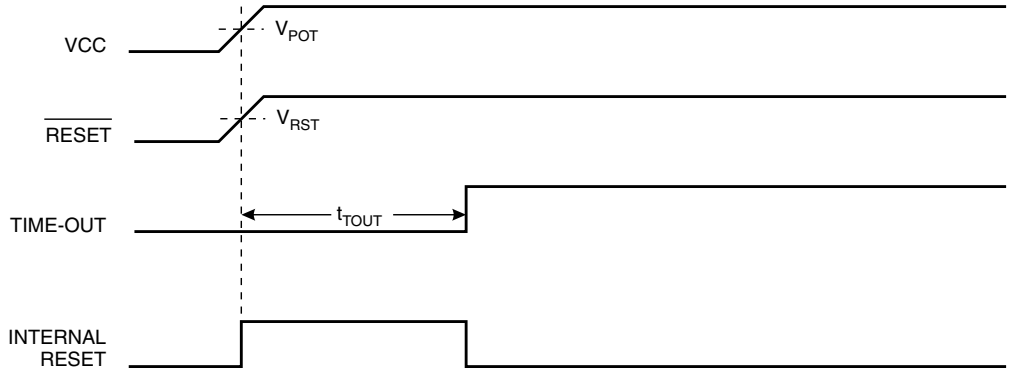
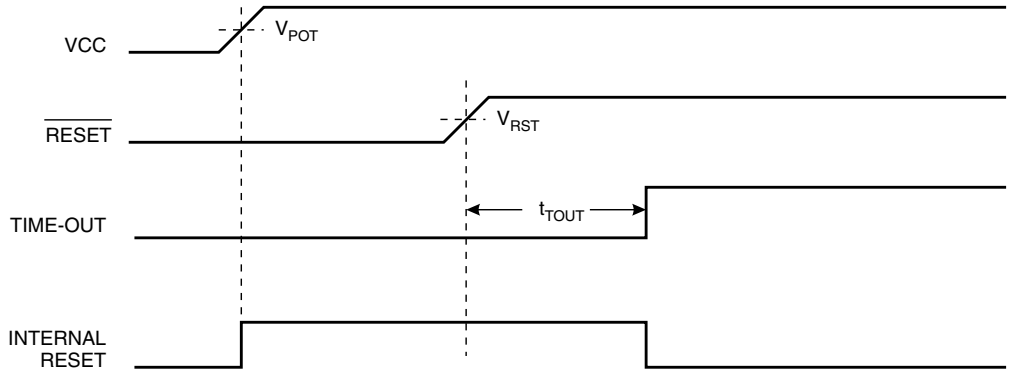


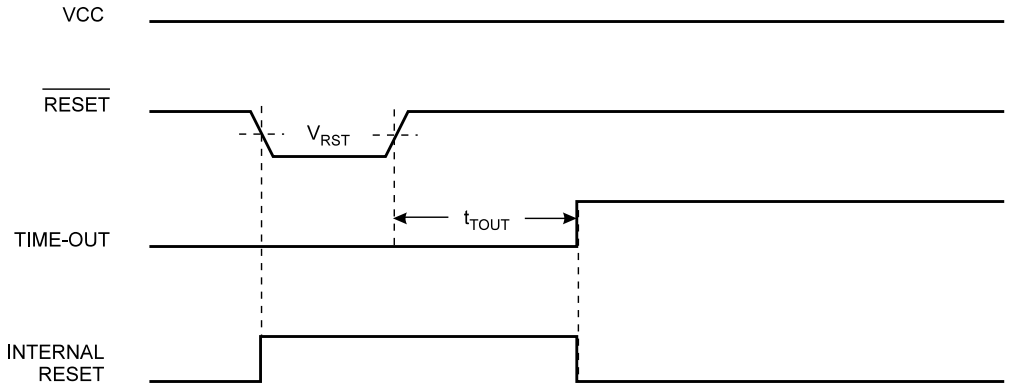
Figure 26. MCU Start-up, $\overline{\text{RESET}}$ Controlled Externally



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



General Interrupt Flag Register – GIFR

Bit	7	6	5	4	3	2	1	0	
$3A ($5A)	–	INTF0	–	–	–	–	–	–	GIFR
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 AT90S2323/2343 and always reads as zero.

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

When an edge on the INT0 pin triggers an interrupt request, the corresponding interrupt flag, INTF0 becomes set (one). If the I-bit in SREG and the corresponding interrupt enable bit, INT0 in GIMSK, is set (one), the MCU will jump to the interrupt vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag is cleared by writing a logical “1” to it. This flag is always cleared when INT0 is configured as level interrupt.

- **Bits 5..0 – Res: Reserved Bits**

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

Timer/Counter Interrupt Mask Register – TIMSK

Bit	7	6	5	4	3	2	1	0	
$39 ($59)	–	–	–	–	–	–	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 AT90S2323/2343 and always read 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 Overflow Flag (Timer/Counter0) is set (one) in the Timer/Counter Interrupt Flag Register (TIFR).

- **Bit 0 – Res: Reserved Bit**

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

Timer/Counter Interrupt FLAG Register – TIFR

Bit	7	6	5	4	3	2	1	0	
$38 ($58)	–	–	–	–	–	–	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 AT90S2323/2343 and always read 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 logical “1” 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 AT90S2323/2343 and always reads zero.

External Interrupt

The external interrupt is triggered by the INT0 pin. Observe that, if enabled, the interrupt will trigger even if the INT0 pin is configured as an output. This feature provides a way of generating a software interrupt. The external interrupt can be triggered by a falling or rising edge or a low level. This is set up as indicated in the specification for the MCU Control Register (MCUCR). When the external interrupt is enabled and is configured as level-triggered, the interrupt will trigger as long as the pin is held low.

The external interrupt is set up as described in the specification for the MCU Control Register (MCUCR).

Interrupt Response Time

The interrupt execution response for all the enabled AVR interrupts is 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 these four clock cycles, the Program Counter (2 bytes) is popped back from the stack, the Stack Pointer is incremented by 2 and the I-flag in SREG is set. The vector is 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 (same as for a subroutine call routine) takes four clock cycles. During these four clock cycles, the Program Counter (2 bytes) is popped back from the stack and the Stack Pointer is incremented by 2. When the AVR exits from an interrupt, it will always return to the main program and execute one more instruction before any pending interrupt is served.

MCU Control Register – MCUCR

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

Bit	7	6	5	4	3	2	1	0	
$35 ($55)	–	–	SE	SM	–	–	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 AT90S2323/2343 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 programmer's 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 section "Sleep Modes".

- **Bits 3, 2 – Res: Reserved Bits**

These bits are reserved bits in the AT90S2323/2343 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 are set. The level and edges on the external INT0 pin that

activate the interrupt are defined in Table 9. The value on the INT01 pin is sampled before detecting edges. If edge or toggle interrupt is selected, pulses that last longer than one clock period will generate an interrupt. Shorter pulses are not guaranteed to generate an interrupt. If low-level interrupt is selected, the low level must be held until the completion of the currently executing instruction to generate an interrupt.

Table 9. 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.

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, SRAM and 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 forces the MCU into the Idle mode, stopping the CPU but allowing Timer/Counters, Watchdog and the interrupt system to continue operating. This enables the MCU to wake up from external triggered interrupts as well as internal ones like Timer Overflow interrupt and Watchdog reset.

Power-down Mode

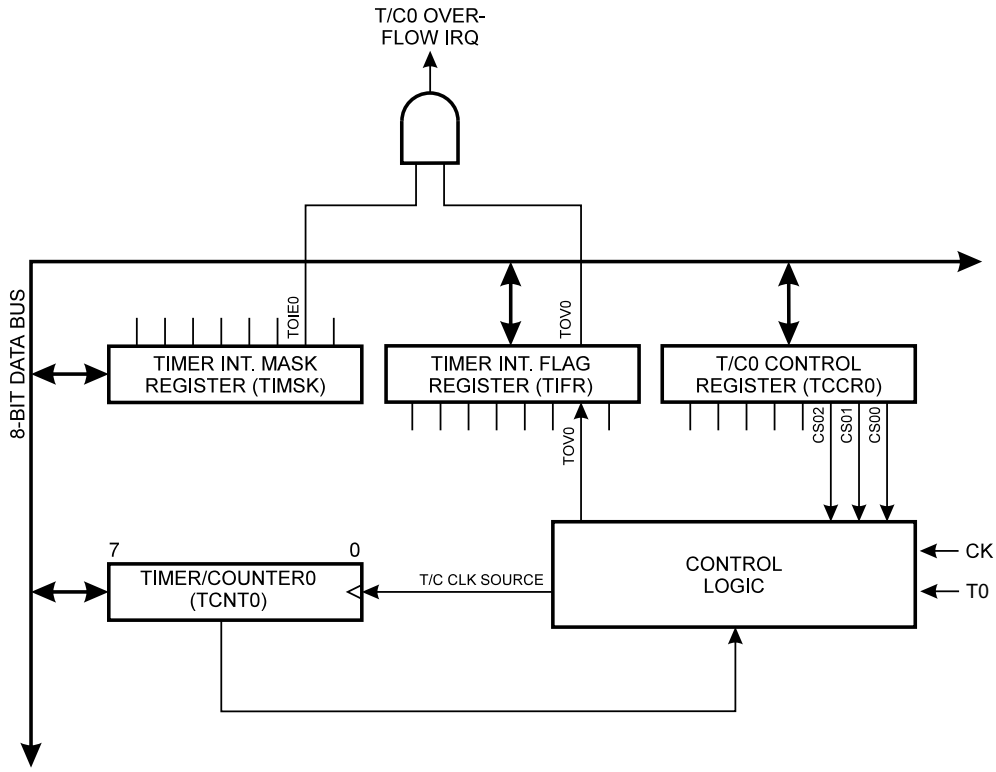
When the SM bit is set (one), the SLEEP instruction forces the MCU into the Power-down mode. In this mode, the external oscillator is stopped while the external interrupts and the Watchdog (if enabled) continue operating. Only an external reset, a Watchdog reset (if enabled), or an external level interrupt on INT0 can wake up the MCU.

Note that if a level-triggered interrupt is used for wake-up from Power-down mode, the changed level must be held for some time to wake up the MCU. This makes the MCU less sensitive to noise. The changed level is sampled twice by the Watchdog oscillator clock and if the input has the required level during this time, the MCU will wake up. The period of the Watchdog oscillator is 1 µs (nominal) at 5.0V and 25°C. The frequency of the Watchdog oscillator is voltage-dependent as shown in section “Typical Characteristics” on page 49.

When waking up from Power-down mode, a delay from the wake-up condition occurs until the wake-up becomes effective. This allows the clock to restart and become stable after having been stopped. The wake-up period is equal to the clock reset period, as shown in Table 4 and Table 5 on page 21.

If the wake-up condition disappears before the MCU wakes up and starts to execute, e.g., a low-level on is not held long enough, the interrupt causing the wake-up will not be executed.

Figure 30. Timer/Counter 0 Block Diagram



Timer/Counter0 Control Register – TCCR0

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

- **Bits 7..3 – Res: Reserved Bits**

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

- **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 10. 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

Memory Programming

Program and Data Memory Lock Bits

The AT90S2323/2343 MCU provides two Lock bits that can be left unprogrammed (“1”) or can be programmed (“0”) to obtain the additional features listed in Table 14. The Lock bits can only be erased with the Chip Erase operation.

Table 14. Lock Bit Protection Modes

Memory Lock Bits			
Mode	**LB1**	**LB2**	**Protection Type**
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 the high-voltage Serial Programming mode, further programming of the Fuse bits are also disabled. Program the Fuse bits before programming the Lock bits.

Fuse Bits in AT90S/LS2323

The AT90S/LS2323 has two Fuse bits, SPIEN and FSTRT.

- When the SPIEN Fuse is programmed (“0”), Serial Program and Data Downloading are enabled. Default value is programmed (“0”). This bit is not accessible in the low-voltage Serial Programming mode.
- When the FSTRT Fuse is programmed (“0”), the shortest start-up time is selected as indicated in Table 6 on page 21. Default value is programmed (“0”). Changing the FSTRT Fuse does not take effect until the next Power-on Reset. In AT90S/LS2343 the start-up time is fixed.

The status of the Fuse bits is not affected by Chip Erase.

Fuse Bits in AT90S/LS2343

The AT90S/LS2343 has two Fuse bits, SPIEN and RCEN.

- When the SPIEN Fuse is programmed (“0”), Serial Program and Data Downloading are enabled. Default value is programmed (“0”). This bit is not accessible in the low-voltage Serial Programming mode.
- When the RCEN Fuse is programmed (“0”), the internal RC oscillator is selected as the MCU clock source. Default value is programmed ("0") in AT90LS2343-1. Default value is un-programmed ("1") in AT90LS2343-4 and AT90S2343-10. Changing the RCEN Fuse does not take effect until the next Power-on Reset. AT90S/LS2323 cannot select the internal RC oscillator as the MCU source.

The status of the Fuse bits is not affected by Chip Erase.

Signature Bytes

All Atmel microcontrollers have a three-byte signature code that identifies the device. The three bytes reside in a separate address space.

For the AT90S/LS2323(Note:), they are:

1. $000: $1E (indicates manufactured by Atmel)
2. $001: $91 (indicates 2K bytes Flash memory)
3. $002: $02 (indicates AT90S/LS2323 when signature byte $001 is $91)

For AT90S/LS2343(Note:), they are:

1. $000: $1E (indicates manufactured by Atmel)
2. $001: $91 (indicates 2K bytes Flash memory)

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

High-voltage Serial Programming Characteristics

Figure 34. High-voltage Serial Programming Timing

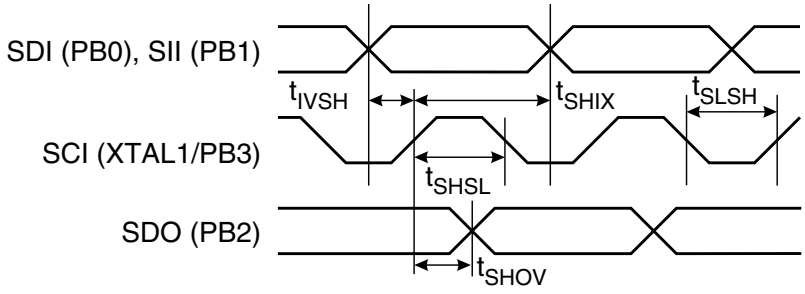


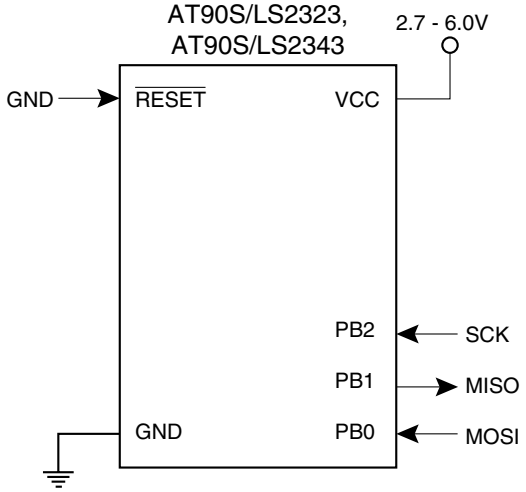
Table 17. High-voltage Serial Programming Characteristics, T_A = 25°C ± 10%, V_{CC} = 5.0V ± 10% (unless otherwise noted)

Symbol	Parameter	Min	Typ	Max	Units
t_{SHSL}	SCI (XTAL1/PB3) Pulse Width High	100.0			ns
t_{SLSH}	SCI (XTAL1/PB3) Pulse Width Low	100.0			ns
t_{IVSH}	SDI (PB0), SII (PB1) Valid to SCI (XTAL1/PB3) High	50.0			ns
t_{SHIX}	SDI (PB0), SII (PB1) Hold after SCI (XTAL1/PB3) High	50.0			ns
t_{SHOV}	SCI (XTAL1/PB3) High to SDO (PB2) Valid	10.0	16.0	32.0	ns
t_{WLWH_CE}	Wait after Instr.3 for Chip Erase	5.0	10.0	15.0	ms
t_{WLWH_PFB}	Wait after Instr.3 for Write Fuse Bits	1.0	1.5	1.8	ms

Low-voltage Serial Downloading

Both the program and data memory arrays can be programmed using the serial SPI bus while $\overline{\text{RESET}}$ is pulled to GND. The serial interface consists of pins SCK, MOSI (input) and MISO (output) (see Figure 35). After $\overline{\text{RESET}}$ is set low, the Programming Enable instruction needs to be executed first before program/erase instructions can be executed.

Figure 35. Low-voltage Serial Programming and Verify



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

The program and EEPROM memory arrays have separate address spaces: $0000 to $03FF 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.

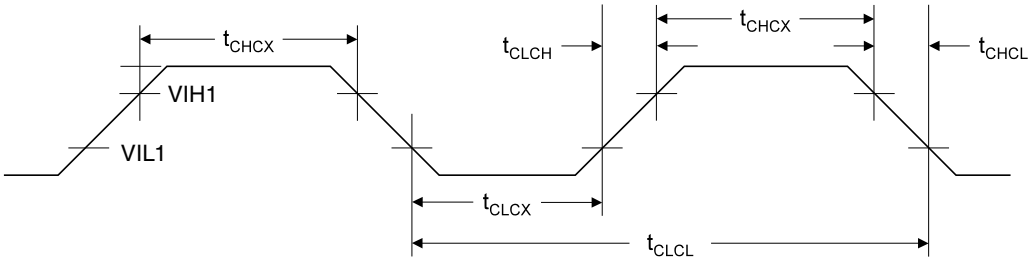
To program and verify the AT90S2323/2343 in the low-voltage Serial Programming mode, the following sequence is recommended (see 4-byte instruction formats in Table 19):

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.

External Clock Drive Waveforms

Figure 38. Waveforms



External Clock Drive

T_A = -40°C to 85°C

Symbol	Parameter	V_{CC}: 2.7V to 4.0V		V_{CC}: 4.0V to 6.0V		Units
		Min	Max	Min	Max	
$1/t_{CLCL}$	Oscillator Frequency	0	4.0	0	10.0	MHz
t_{CLCL}	Clock Period	250.0		100.0		ns
t_{CHCX}	High Time	100.0		40.0		ns
t_{CLCX}	Low Time	100.0		40.0		ns
t_{CLCH}	Rise Time		1.6		0.5	µs
t_{CHCL}	Fall Time		1.6		0.5	µs

Figure 42. Idle Supply Current vs. Frequency

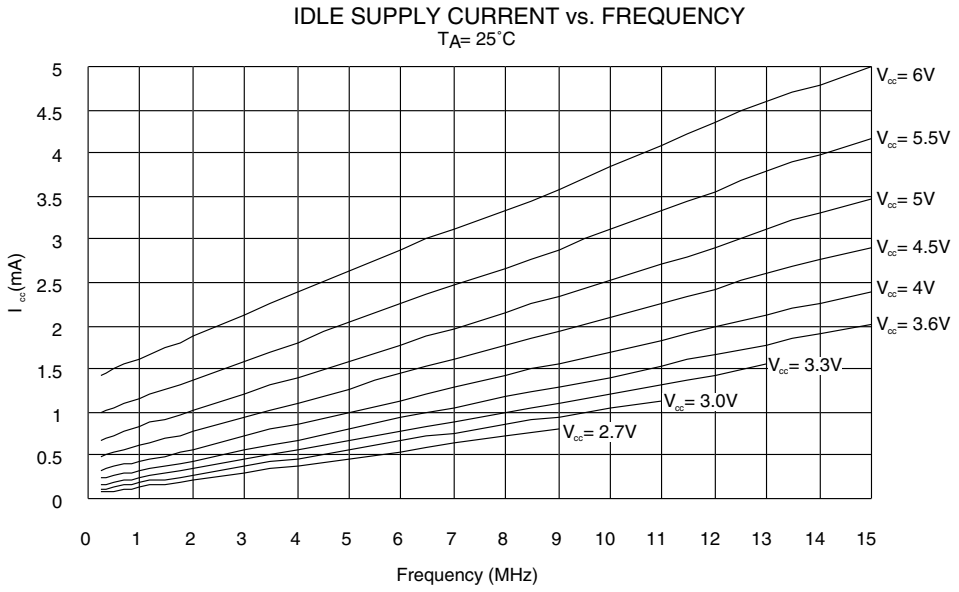


Figure 43. Idle Supply Current vs. V_{CC}

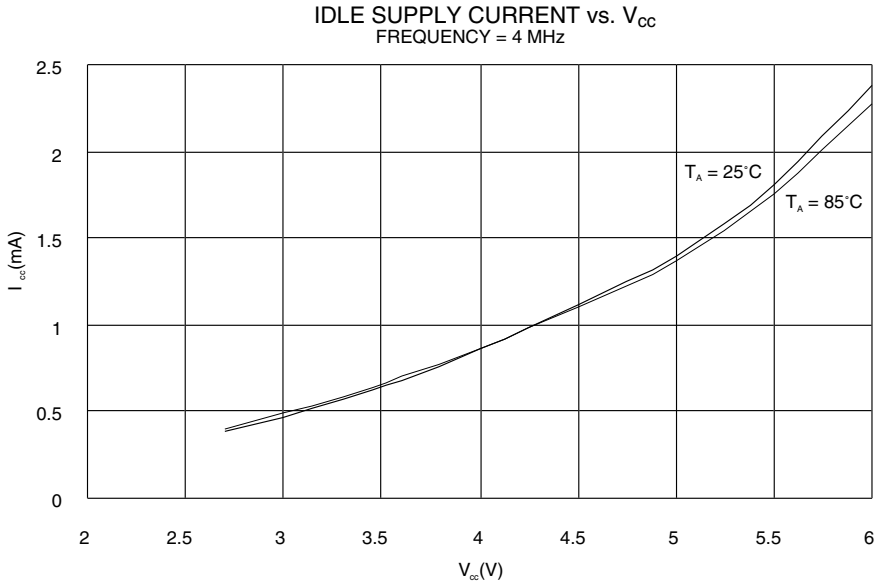


Figure 44. Idle Supply Current vs. V_{CC}

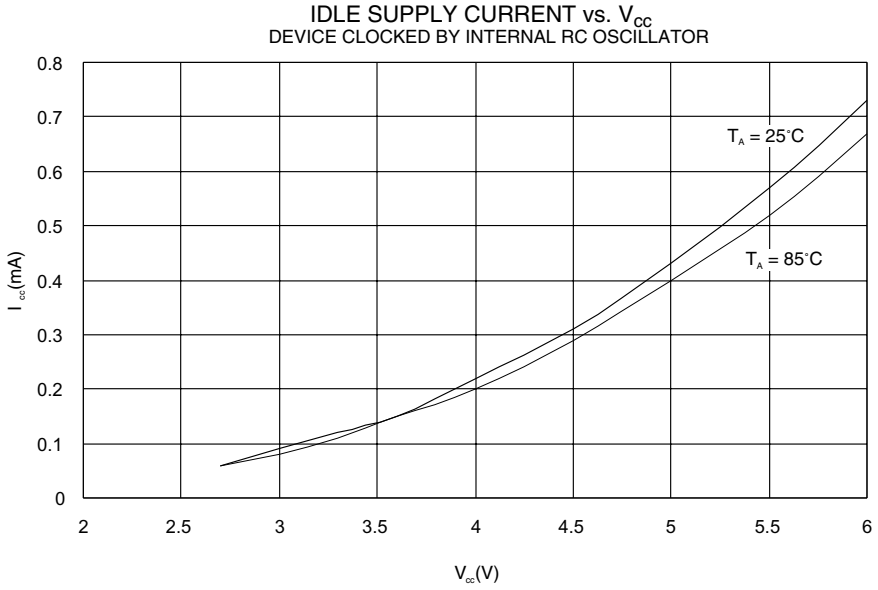


Figure 45. Power-down Supply Current vs. V_{CC}

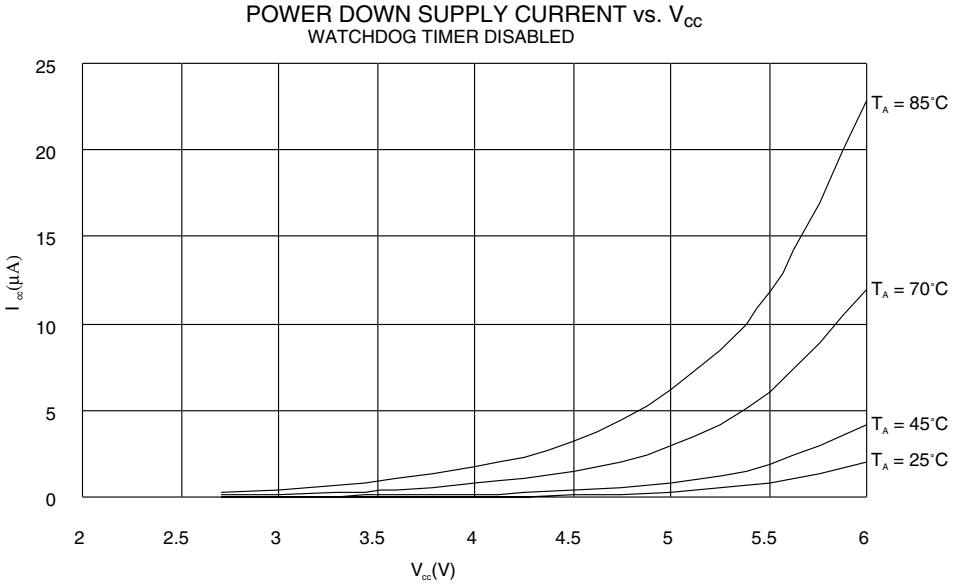


Figure 50. I/O Pin Sink Current vs. Output Voltage

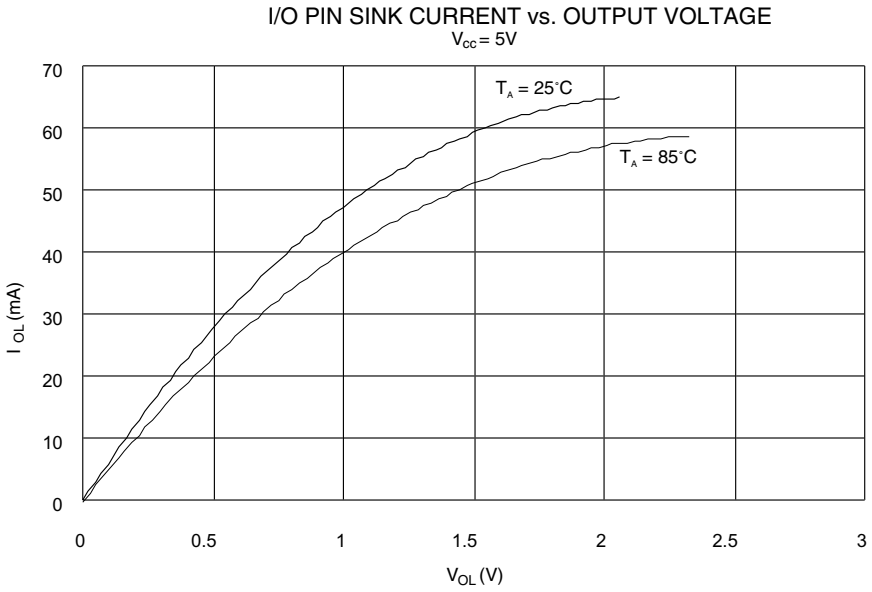


Figure 51. I/O PIn Source Current vs. Output Voltage