



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 "<u>Embedded -</u> <u>Microcontrollers</u>"

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 | Through Hole |
| Package / Case | 8-DIP (0.300", 7.62mm) |
| Supplier Device Package | 8-PDIP |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/at90s2323-10pc |

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong



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.



Figure 1. The AT90S/LS2343 Block Diagram





Figure 2. The AT90S/LS2323 Block Diagram

The AT90S2323/2343 provides the following features: 2K bytes of In-System Programmable Flash, 128 bytes EEPROM, 128 bytes SRAM, 3 (AT90S/LS2323)/5 (AT90S/LS2343) general-purpose I/O lines, 32 general-purpose working registers, an 8bit timer/counter, internal and external interrupts, programmable Watchdog Timer with internal oscillator, an SPI serial port for Flash Memory downloading and two softwareselectable power-saving modes. The Idle mode stops the CPU while allowing the SRAM, timer/counters, SPI port and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the oscillator, disabling all other chip functions until the next interrupt or hardware reset.

The device is manufactured using Atmel's high-density nonvolatile memory technology. The On-chip Flash allows the program memory to be reprogrammed in-system through an SPI serial interface. By combining an 8-bit RISC CPU with ISP Flash on a monolithic











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

| | 7 | 0 | Addr. |
|-----------|-----|---|-------|
| | RC |) | \$00 |
| | R1 | | \$01 |
| | R2 |) | \$02 |
| | | | |
| | R1 | 3 | \$0D |
| General | R1 | 4 | \$0E |
| Purpose | R1 | 5 | \$0F |
| Working | R1 | ô | \$10 |
| Registers | R1 | 7 | \$11 |
| | | | |
| | R20 | ô | \$1A |
| | R2 | 7 | \$1B |
| | R23 | В | \$1C |
| | R2 | 9 | \$1D |
| | R3 | 0 | \$1E |
| | R3 | 1 | \$1F |

X-register low byte X-register high byte Y-register low byte Y-register high byte Z-register low byte Z-register high byte

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.





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.





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



Data Indirect with Displacement



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.

| | • | |
|-------------|-------|---------------------------------------|
| 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 |

Table 2. AT90S2323/2343 I/O Space

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 generalpurpose 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



General Interrupt Flag Register – GIFR



• 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 INTO pin triggers an interrupt request, the corresponding interrupt flag, INTFO becomes set (one). If the I-bit in SREG and the corresponding interrupt enable bit, INTO 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 INTO 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

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | _ |
|---|------------------|--------------------------|-------------------------------|---------------------------------------|---|---|---|---|
| - | - | - | - | - | - | TOIE0 | - | TIMSK |
| R | R | R | R | R | R | R/W | R | _ |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | 7 - R 0 | 7 6 – – R R 0 0 | 7 6 5 - - - R R R 0 0 0 | 7 6 5 4 - - - - R R R R 0 0 0 0 | 7 6 5 4 3 - - - - - R R R R R 0 0 0 0 0 | 7 6 5 4 3 2 - - - - - - R R R R R R 0 0 0 0 0 0 | 7 6 5 4 3 2 1 - - - - - TOIE0 R R R R R R/W 0 0 0 0 0 0 | 7 6 5 4 3 2 1 0 - - - - - TOIE0 - R R R R R R/W R 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



• 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.



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.

| Гal | ble | 9. | Interrupt | 0 | Sense | Contro | |
|-----|-----|----|-----------|---|-------|--------|--|
|-----|-----|----|-----------|---|-------|--------|--|

| 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 Powerdown 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 INTO 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.







• 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 | СК |
| 0 | 1 | 0 | СК/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 |



Timer/Counter0 Control Register – TCCR0

EEPROM Control Register – EECR



• 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.



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 SerialWhen writing serial data to the AT90S2323/2343, data is clocked on the rising edge ofProgramming AlgorithmSCK.

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.
- If a Chip Erase is performed (must be done to erase the Flash), wait t_{WD_ERASE} after the instruction, give 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.





- 7. At the end of the programming session, **RESET** can be set high to commence normal operation.
- Power-off sequence (if needed): Set CLOCK/XTAL1 to "0".
 Set RESET to "1".
 Turn V_{CC} power off.

Data Polling EEPROM When a byte is being programmed into the EEPROM, reading the address location being programmed will give the value P1 until the auto-erase is finished, and then the value P2 will be given. See Table 18 for P1 and P2 values.

At the time the device is ready for a new EEPROM byte, the programmed value will read correctly. This is used to determine when the next byte can be written. This will not work for the values P1 and P2, so when programming these values, the user will have to wait for at least the prescribed time t_{WD_PROG} before programming the next byte. See Table 22 for t_{WD_PROG} value. As a chip-erased device contains \$FF in all locations, programming of addresses that are meant to contain \$FF can be skipped. This does not apply if the EEPROM is reprogrammed without first chip-erasing the device.

| Table 18. | Read Back | Value during | EEPROM Polling |
|-----------|-----------|--------------|-----------------------|
|-----------|-----------|--------------|-----------------------|

| Part | P1 | P2 |
|-----------|------|------|
| AT90S2323 | \$00 | \$FF |
| AT90S2343 | \$00 | \$FF |

Data Polling Flash

When a byte is being programmed into the Flash, reading the address location being programmed will give the value FF. At the time the device is ready for a new byte, the programmed value will read correctly. This is used to determine when the next byte can be written. This will not work for the value FF, so when programming this value, the user will have to wait for at least t_{WD_PROG} before programming the next byte. As a chiperased device contains FF in all locations, programming of addresses that are meant to contain FF can be skipped.





Typical Characteristics

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

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

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

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

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









Figure 42. Idle Supply Current vs. Frequency

















Instruction Set Summary

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



Ordering Information

| Power Supply | Speed (MHz) | Ordering Code | Package | Operation Range |
|--------------|-------------|----------------|---------|-----------------|
| 2.7 - 6.0V | 4 | AT90LS2323-4PC | 8P3 | Commercial |
| | | AT90LS2323-4SC | 8S2 | (0°C to 70°C) |
| | | AT90LS2323-4PI | 8P3 | Industrial |
| | | AT90LS2323-4SI | 8S2 | (-40°C to 85°C) |
| 4.0 - 6.0V | 10 | AT90S2323-10PC | 8P3 | Commercial |
| | | AT90S2323-10SC | 8S2 | (0°C to 70°C) |
| | | AT90S2323-10PI | 8P3 | Industrial |
| | | AT90S2323-10SI | 8S2 | (-40°C to 85°C) |
| 2.7 - 6.0V | 1 | AT90LS2343-1PC | 8P3 | Commercial |
| | | AT90LS2343-1SC | 8S2 | (0°C to 70°C) |
| | | AT90LS2343-1PI | 8P3 | Industrial |
| | | AT90LS2343-1SI | 8S2 | (-40°C to 85°C) |
| 2.7 - 6.0V | 4 | AT90LS2343-4PC | 8P3 | Commercial |
| | | AT90LS2343-4SC | 8S2 | (0°C to 70°C) |
| | | AT90LS2343-4PI | 8P3 | Industrial |
| | | AT90LS2343-4SI | 8S2 | (-40°C to 85°C) |
| 4.0 - 6.0V | 10 | AT90S2343-10PC | 8P3 | Commercial |
| | | AT90S2343-10SC | 8S2 | (0°C to 70°C) |
| | | AT90S2343-10PI | 8P3 | Industrial |
| | | AT90S2343-10SI | 8S2 | (-40°C to 85°C) |

Notes: 1. The speed grade refers to maximum clock rate when using an external crystal or external clock drive. The internal RC oscillator has the same nominal clock frequency for all speed grades.

2. In AT90LS2343-1xx, the internal RC oscillator is selected as default MCU clock source (RCEN fuse is programmed) when the device is shipped from Atmel. In AT90LS2343-4xx and AT90S2343-10xx, the default MCU clock source is the clock input pin (RCEN fuse is unprogrammed). The fuse settings can be changed by high voltage serial programming.

| | Package Type |
|-----|--|
| 8P3 | 8-lead, 0.300" Wide, Plastic Dual Inline Package (PDIP) |
| 8S2 | 8-lead, 0.200" Wide, Plastic Gull Wing Small Outline Package (EIAJ SOIC) |



8S2



