



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China



Features

- AVR[®] – High-performance and Low-power RISC Architecture
 - 118 Powerful Instructions – Most Single Clock Cycle Execution
 - 32 x 8 General-purpose Working Registers
 - Up to 8 MIPS Throughput at 8 MHz
- Data and Nonvolatile Program Memories
 - 8K Bytes of In-System Programmable Flash
SPI Serial Interface for In-System Programming
Endurance: 1,000 Write/Erase Cycles
 - 512 Bytes EEPROM
Endurance: 100,000 Write/Erase Cycles
 - 512 Bytes Internal SRAM
 - Programming Lock for Software Security
- Peripheral Features
 - 8-channel, 10-bit ADC
 - Programmable UART
 - Master/Slave SPI Serial Interface
 - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
 - One 16-bit Timer/Counter with Separate Prescaler, Compare and Capture Modes and Dual 8-, 9-, or 10-bit PWM
 - Programmable Watchdog Timer with On-chip Oscillator
 - On-chip Analog Comparator
- Special Microcontroller Features
 - Power-on Reset Circuit
 - Real-time Clock (RTC) with Separate Oscillator and Counter Mode
 - External and Internal Interrupt Sources
 - Three Sleep Modes: Idle, Power Save and Power-down
- Power Consumption at 4 MHz, 3V, 20°C
 - Active: 6.4 mA
 - Idle Mode: 1.9 mA
 - Power-down Mode: <1 μA
- I/O and Packages
 - 32 Programmable I/O Lines
 - 40-lead PDIP, 44-lead PLCC, 44-lead TQFP, and 44-pad MLF
- Operating Voltages
 - V_{CC}: 4.0 - 6.0V AT90S8535
 - V_{CC}: 2.7 - 6.0V AT90LS8535
- Speed Grades:
 - 0 - 8 MHz for the AT90S8535
 - 0 - 4 MHz for the AT90LS8535



8-bit AVR[®]
Microcontroller
with 8K Bytes
In-System
Programmable
Flash

AT90S8535

AT90LS8535

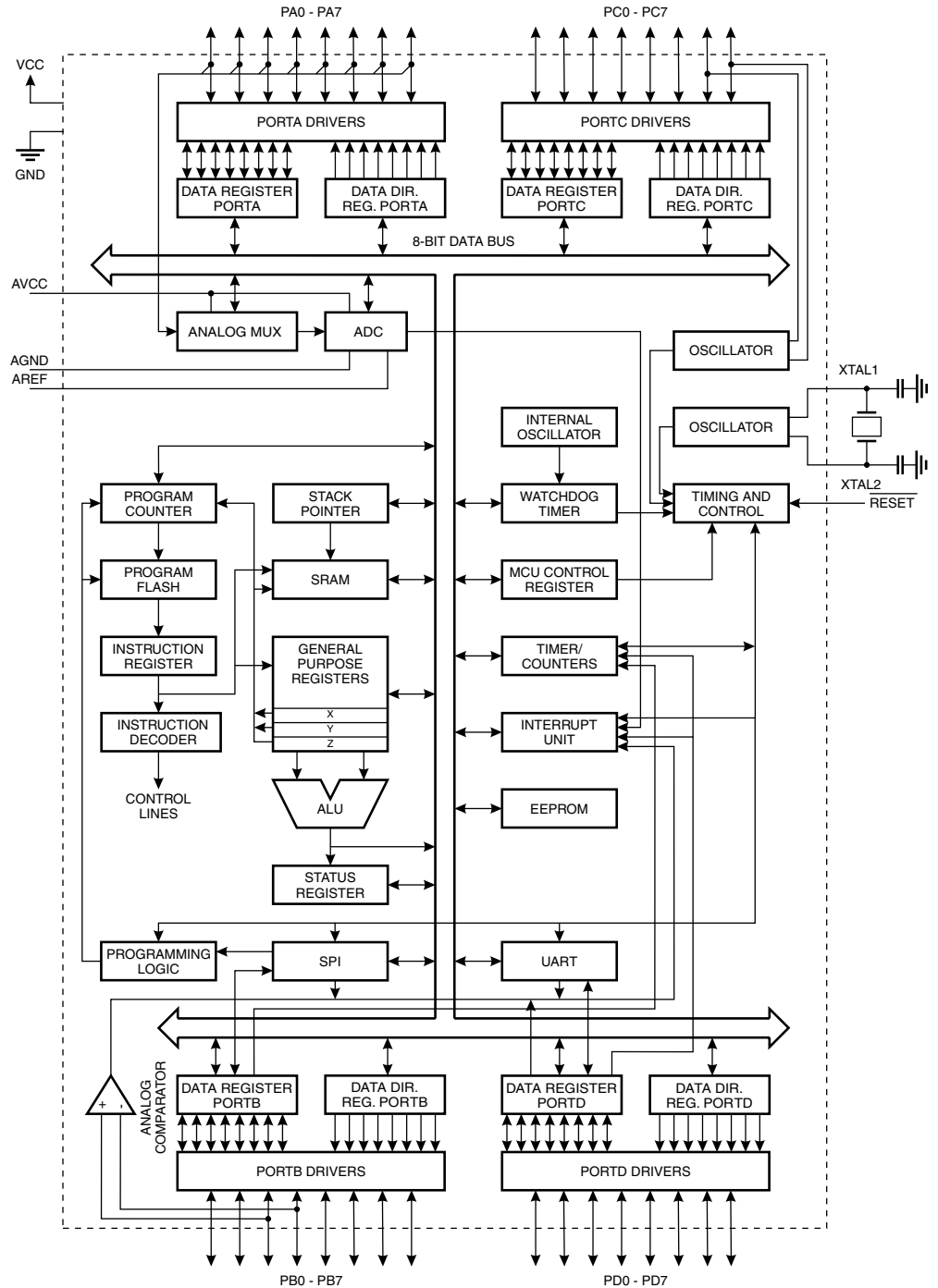


Description

The AT90S8535 is a low-power CMOS 8-bit microcontroller based on the AVR RISC architecture. By executing powerful instructions in a single clock cycle, the AT90S8535 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

Block Diagram

Figure 1. The AT90S8535 Block Diagram





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.

The AT90S8535 provides the following features: 8K bytes of In-System Programmable Flash, 512 bytes EEPROM, 512 bytes SRAM, 32 general-purpose I/O lines, 32 general-purpose working registers, Real-time Clock (RTC), three flexible timer/counters with compare modes, internal and external interrupts, a programmable serial UART, 8-channel, 10-bit ADC, programmable Watchdog Timer with internal oscillator, an SPI serial port and three software-selectable 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. In Power Save Mode, the timer oscillator continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping.

The device is manufactured using Atmel's high-density nonvolatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed in-system through an SPI serial interface or by a conventional nonvolatile memory programmer. By combining an 8-bit RISC CPU with In-System Programmable Flash on a monolithic chip, the Atmel AT90S8535 is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The AT90S8535 AVR is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, in-circuit emulators and evaluation kits.

Pin Descriptions

VCC	Digital supply voltage.
GND	Digital ground.
Port A (PA7..PA0)	<p>Port A is an 8-bit bi-directional I/O port. Port pins can provide internal pull-up resistors (selected for each bit). The Port A output buffers can sink 20 mA and can drive LED displays directly. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated.</p> <p>Port A also serves as the analog inputs to the A/D Converter.</p> <p>The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p>
Port B (PB7..PB0)	<p>Port B is an 8-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 of the AT90S8535 as listed on page 78.</p> <p>The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p>
Port C (PC7..PC0)	<p>Port C is an 8-bit bi-directional I/O port with internal pull-up resistors. The Port C output buffers can sink 20 mA. As inputs, Port C pins that are externally pulled low will source</p>

current if the pull-up resistors are activated. Two Port C pins can alternatively be used as oscillator for Timer/Counter2.

The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port D (PD7..PD0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors. The Port D output buffers can sink 20 mA. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated.

Port D also serves the functions of various special features of the AT90S8535 as listed on page 86.

The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

 $\overline{\text{RESET}}$

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.

XTAL1

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

XTAL2

Output from the inverting oscillator amplifier.

AVCC

AVCC is the supply voltage pin for Port A and the A/D Converter. If the ADC is not used, this pin must be connected to VCC. If the ADC is used, this pin must be connected to VCC via a low-pass filter. See page 68 for details on operation of the ADC.

AREF

AREF is the analog reference input for the A/D Converter. For ADC operations, a voltage in the range 2V to AV_{CC} must be applied to this pin.

AGND

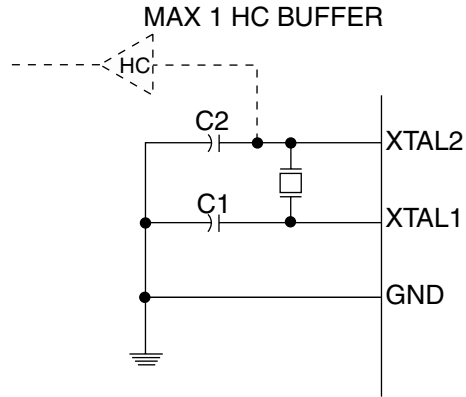
Analog ground. If the board has a separate analog ground plane, this pin should be connected to this ground plane. Otherwise, connect to GND.

Clock Options

Crystal Oscillator

XTAL1 and XTAL2 are input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 2. Either a quartz crystal or a ceramic resonator may be used.

Figure 2. Oscillator Connections

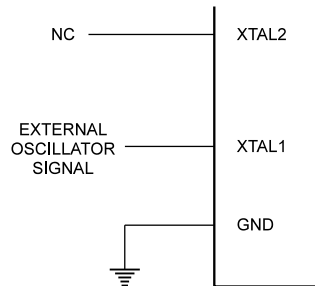


Note: When using the MCU Oscillator as a clock for an external device, an HC buffer should be connected as indicated in the figure.

External Clock

To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 3.

Figure 3. External Clock Drive Configuration



Timer Oscillator

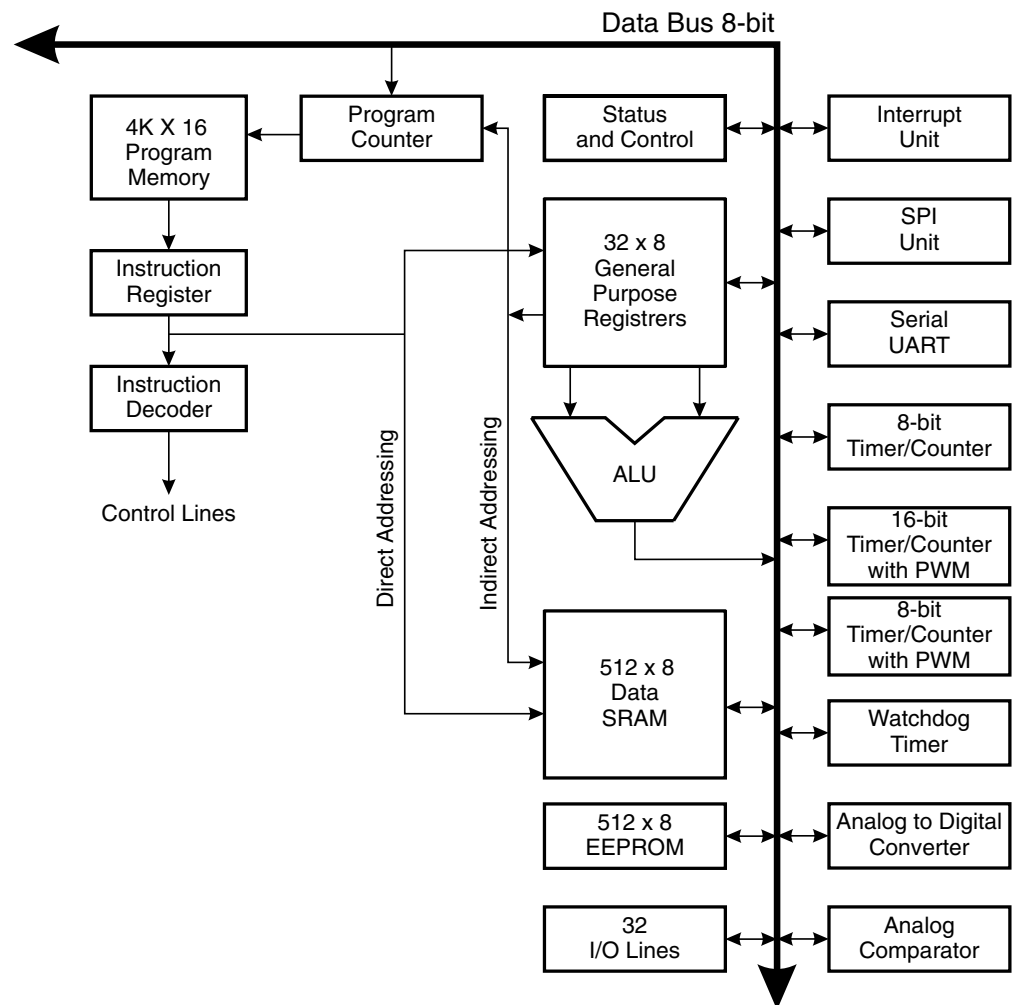
For the Timer Oscillator pins, TOSC1 and TOSC2, the crystal is connected directly between the pins. No external capacitors are needed. The oscillator is optimized for use with a 32,768 Hz watch crystal. Applying an external clock source to TOSC1 is not recommended.

Architectural Overview

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

Six of the 32 registers can be used as three 16-bit indirect address register pointers for Data Space addressing, enabling efficient address calculations. One of the three address pointers is also used as the address pointer for the constant table look-up function. These added function registers are the 16-bit X-register, Y-register, and Z-register.

Figure 4. The AT90S8535 AVR RISC Architecture



The ALU supports arithmetic and logic functions between registers or between a constant and a register. Single register operations are also executed in the ALU. Figure 4 shows the AT90S8535 AVR RISC microcontroller architecture.

In addition to the register operation, the conventional memory addressing modes can be used on the register file as well. This is enabled by the fact that the register file is

assigned the 32 lowermost Data Space addresses (\$00 - \$1F), allowing them to be accessed as though they were ordinary memory locations.

The I/O memory space contains 64 addresses for CPU peripheral functions as Control Registers, Timer/Counters, A/D converters and other I/O functions. The I/O memory can be accessed directly or as the Data Space locations following those of the register file, \$20 - \$5F.

The AVR uses a Harvard architecture concept – with separate memories and buses for program and data. The program memory is executed 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.

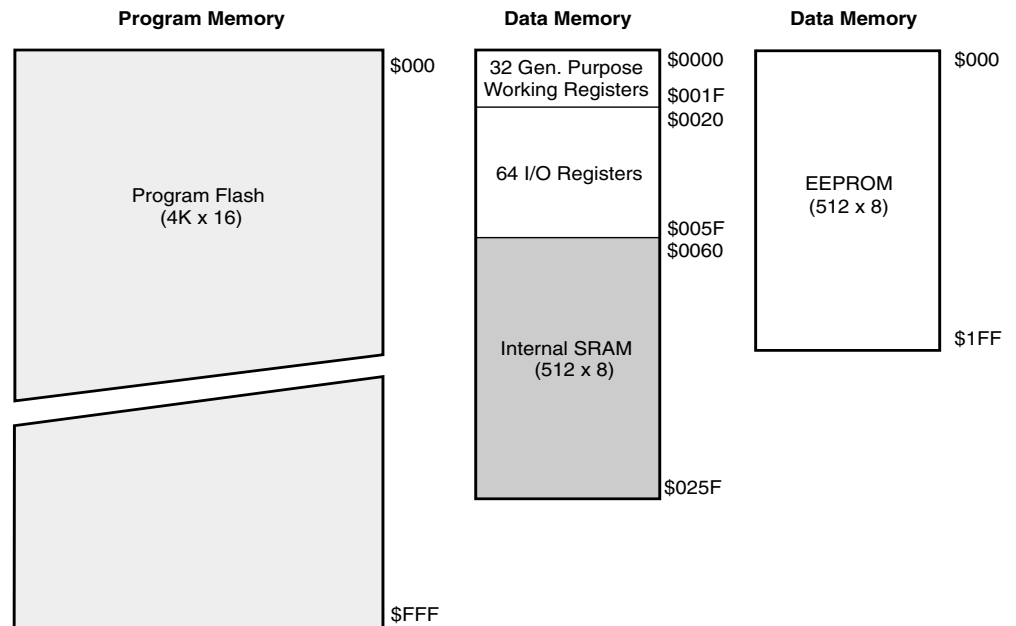
With the relative jump and call instructions, the whole 4K address space is directly accessed. Most AVR instructions have a single 16-bit word format. Every program memory address contains a 16- or 32-bit instruction.

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 10-bit stack pointer (SP) is read/write-accessible in the I/O space.

The 512 bytes data SRAM 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 5. Memory Maps



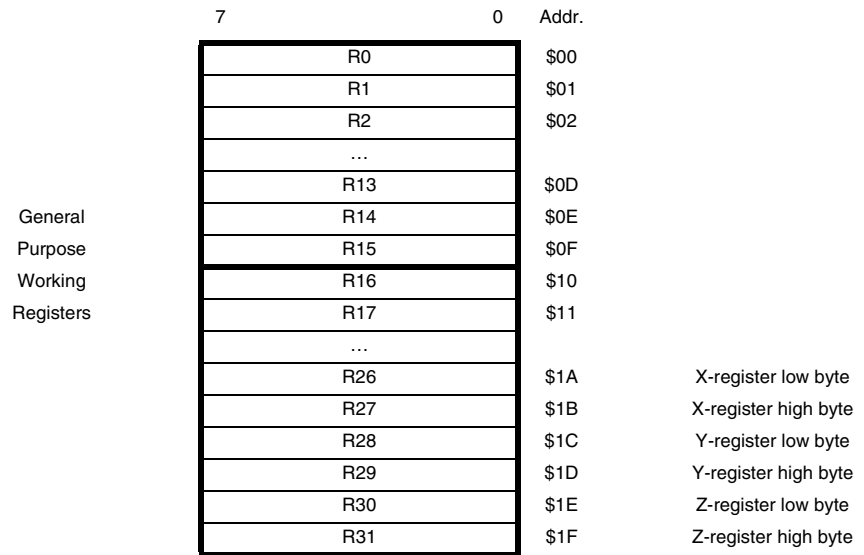
A flexible interrupt module has its control registers in the I/O space with an additional global interrupt enable bit in the status register. All the different interrupts have a separate interrupt vector in the interrupt vector table at the beginning of the program

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 6 shows the structure of the 32 general-purpose working registers in the CPU.

Figure 6. AVR CPU General-purpose Working Registers



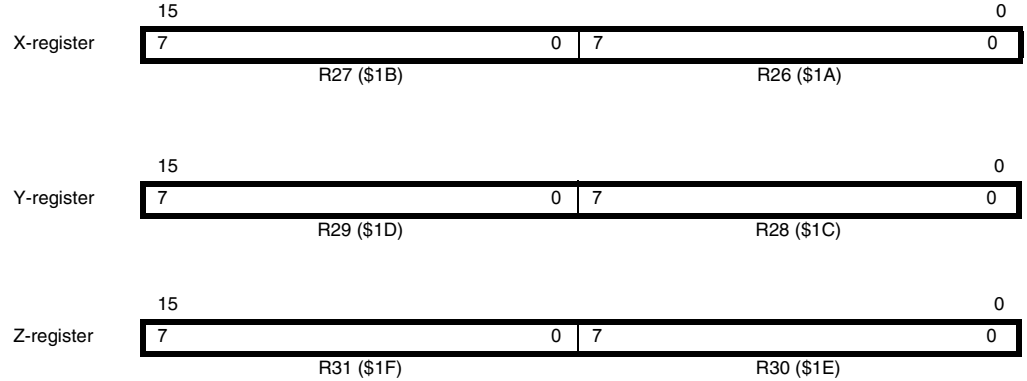
All the register operating instructions in the instruction set have direct and single-cycle access to all registers. The only exception is the five constant arithmetic and logic instructions SBCI, SUBI, CPI, ANDI and ORI between a constant and a register and the LDI instruction for load immediate constant 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 6, each register is also assigned a data memory address, mapping them directly into the first 32 locations of the user Data Space. Although not being 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 address pointers for indirect addressing of the Data Space. The three indirect address registers, X, Y, and Z, are defined in Figure 7.

Figure 7. X-, Y-, and Z-register



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, logical and bit functions.

In-System Programmable Flash Program Memory

The AT90S8535 contains 8K 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 4K x 16. The Flash memory has an endurance of at least 1000 write/erase cycles. The AT90S8535 Program Counter (PC) is 12 bits wide, thus addressing the 4096 program memory addresses.

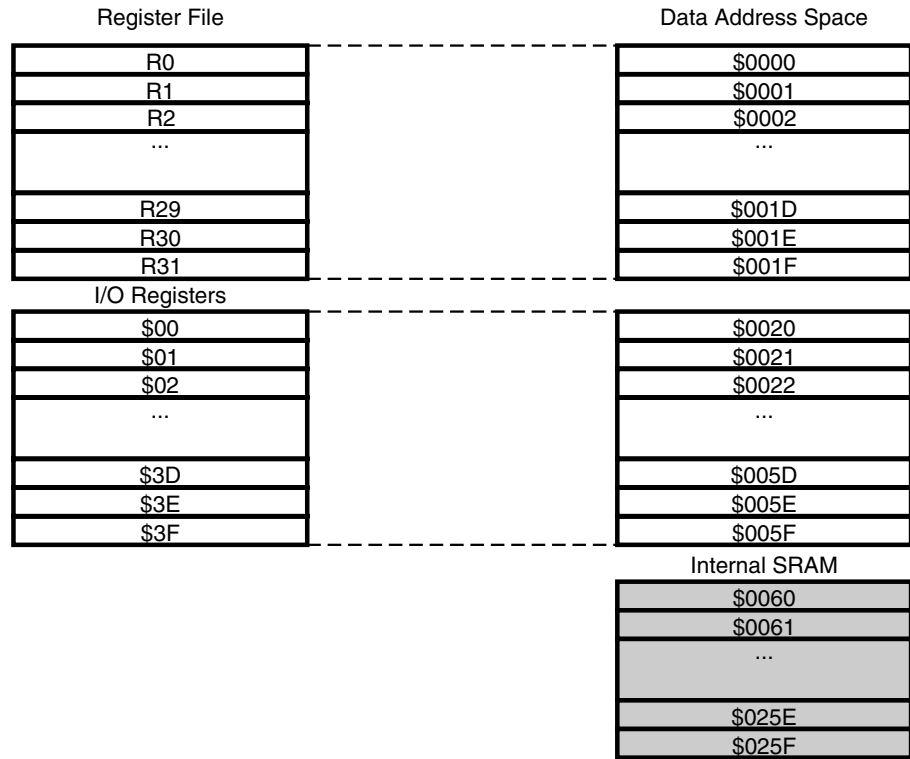
See page 99 for a detailed description on Flash data downloading.

See page 12 for the different program memory addressing modes.

SRAM Data Memory

Figure 8 shows how the AT90S8535 SRAM memory is organized.

Figure 8. SRAM Organization



The lower 608 data memory locations address the Register file, the I/O memory and the internal data SRAM. The first 96 locations address the Register file + I/O memory, and the next 512 locations address the internal 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 space.

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

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

The 32 general-purpose working registers, 64 I/O registers and the 512 bytes of internal data SRAM in the AT90S8535 are all accessible through all these addressing modes.

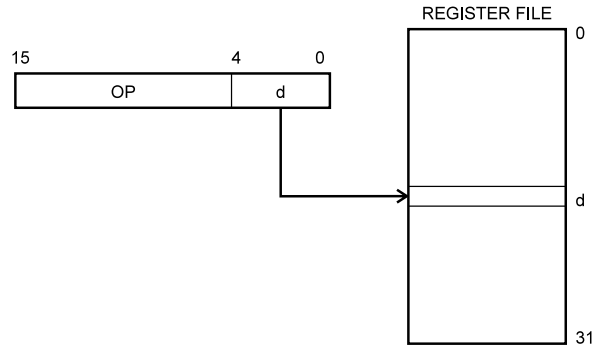
See the next section for a detailed description of the different addressing modes.

Program and Data Addressing Modes

The AT90S8535 AVR RISC microcontroller supports powerful and efficient addressing modes for access to the program memory (Flash) and data memory (SRAM, register file and I/O memory). This section describes the different addressing modes supported by the AVR architecture. In the figures, OP means the operation code part of the instruction word. To simplify, not all figures show the exact location of the addressing bits.

Register Direct, Single Register Rd

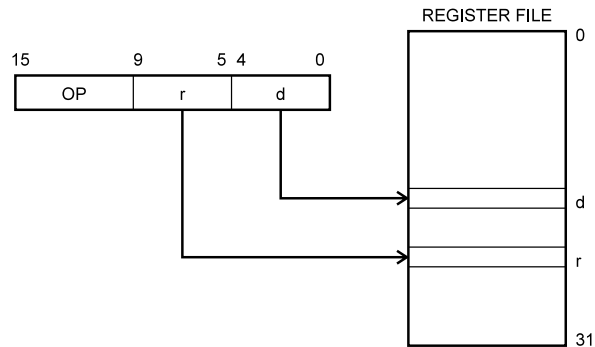
Figure 9. Direct Single Register Addressing



The operand is contained in register d (Rd).

Register Direct, Two Registers Rd And Rr

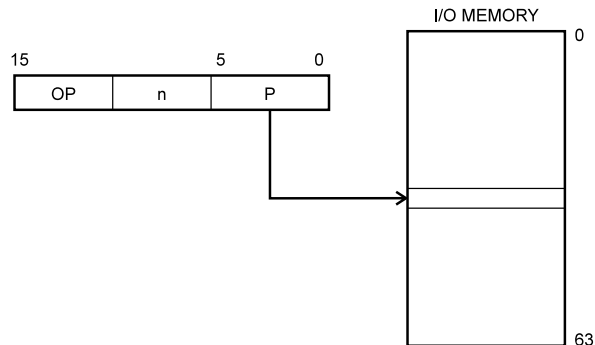
Figure 10. Direct Register Addressing, Two Registers



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

I/O Direct

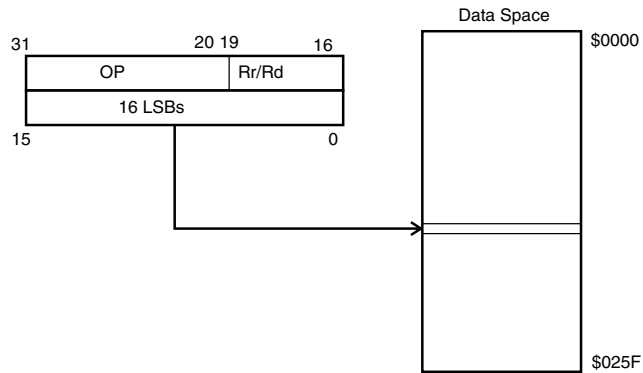
Figure 11. I/O Direct Addressing



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

Data Direct

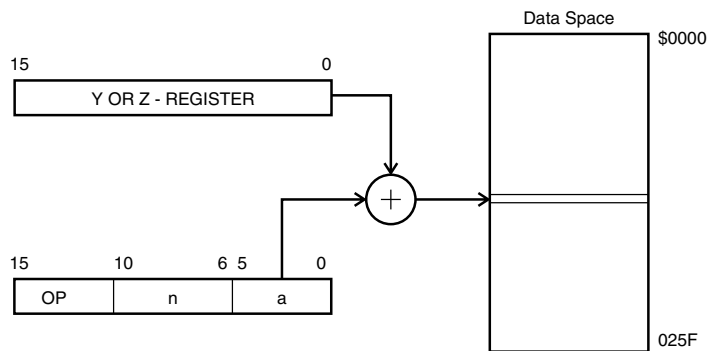
Figure 12. Direct Data Addressing



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

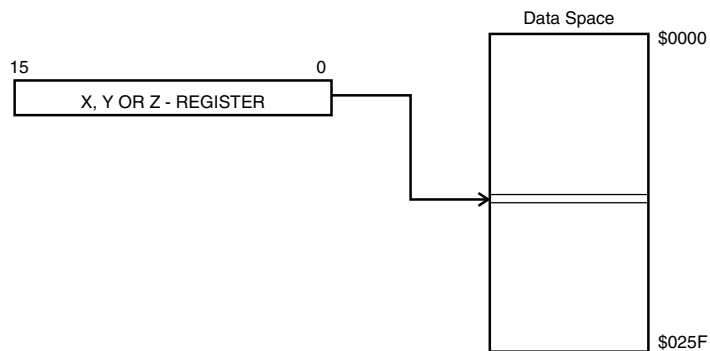
Figure 13. Data Indirect with Displacement



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

Data Indirect

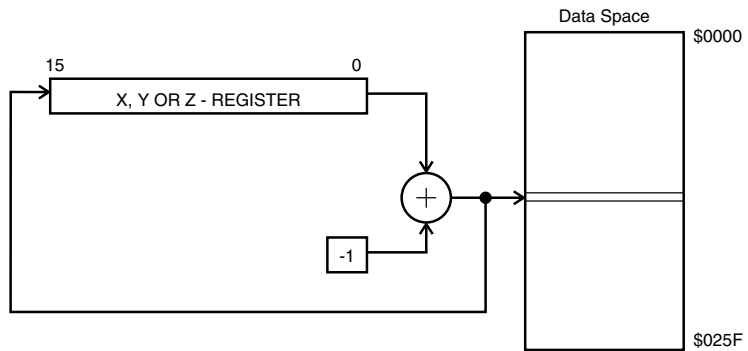
Figure 14. Data Indirect Addressing



Operand address is the contents of the X-, Y-, or the Z-register.

Data Indirect with Pre-decrement

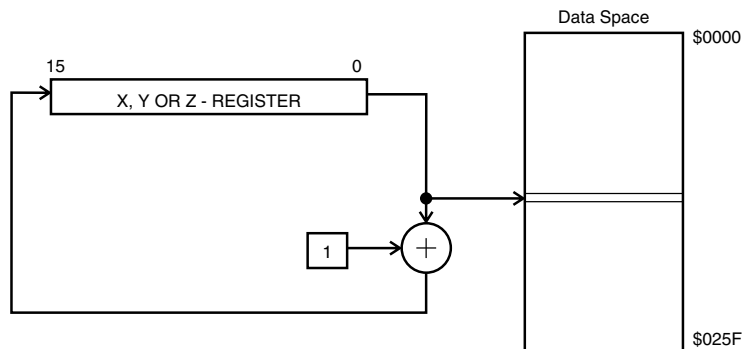
Figure 15. Data Indirect Addressing with Pre-decrement



The X-, Y-, or the Z-register is decremented before the operation. Operand address is the decremented contents of the X-, Y-, or the Z-register.

Data Indirect with Post-increment

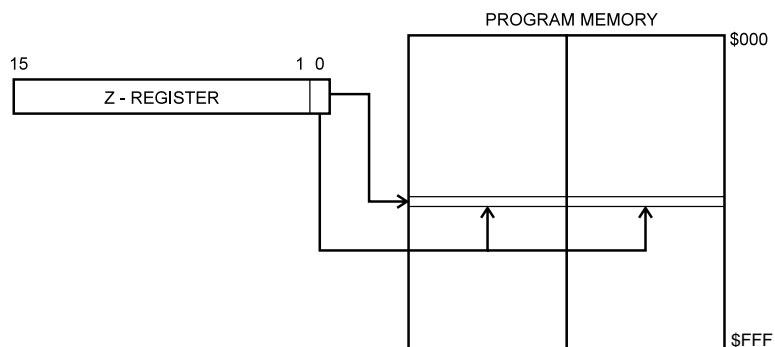
Figure 16. Data Indirect Addressing with Post-increment



The X-, Y-, or the Z-register is incremented after the operation. Operand address is the content of the X-, Y-, or the Z-register prior to incrementing.

Constant Addressing Using the LPM Instruction

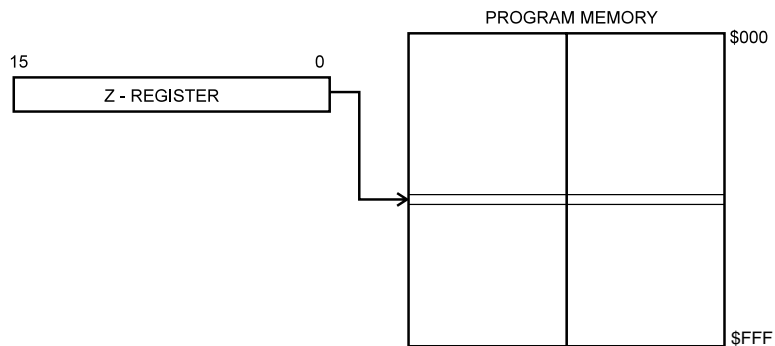
Figure 17. Code Memory Constant Addressing



Constant byte address is specified by the Z-register contents. The 15 MSBs select word address (0 - 4K), the LSB selects low byte if cleared (LSB = 0) or high byte if set (LSB = 1).

Indirect Program Addressing, IJMP and ICALL

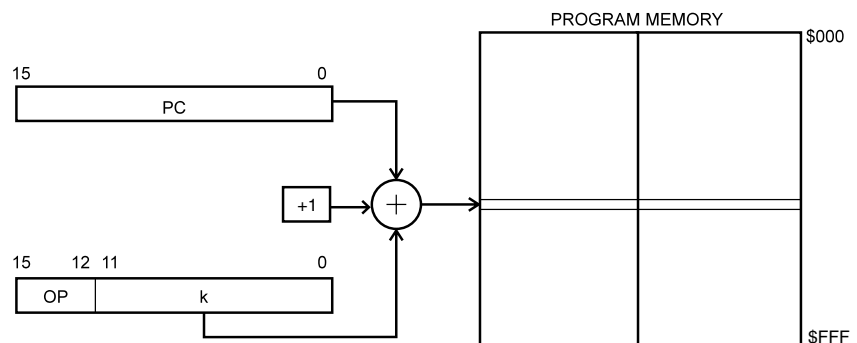
Figure 18. Indirect Program Memory Addressing



Program execution continues at address contained by the Z-register (i.e., the PC is loaded with the contents of the Z-register).

Relative Program Addressing, RJMP and RCALL

Figure 19. Relative Program Memory Addressing



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

EEPROM Data Memory

The AT90S8535 contains 512 bytes of data EEPROM memory. It is organized as a separate data space, in which single bytes can be read and written. The EEPROM has an endurance of at least 100,000 write/erase cycles. The access between the EEPROM and the CPU is described on page 51 specifying the EEPROM address registers, the EEPROM data register and the EEPROM control register.

For the SPI data downloading, see page 99 for a detailed description.

Memory Access Times 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 \emptyset , directly generated from the external clock crystal for the chip. No internal clock division is used.

Figure 20 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 20. The Parallel Instruction Fetches and Instruction Executions

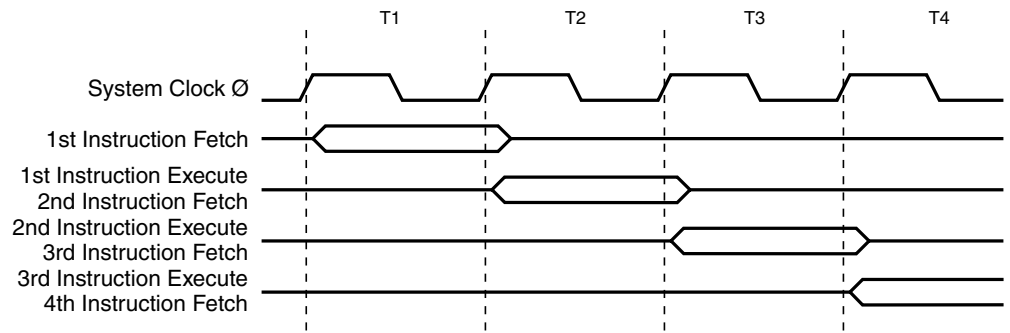
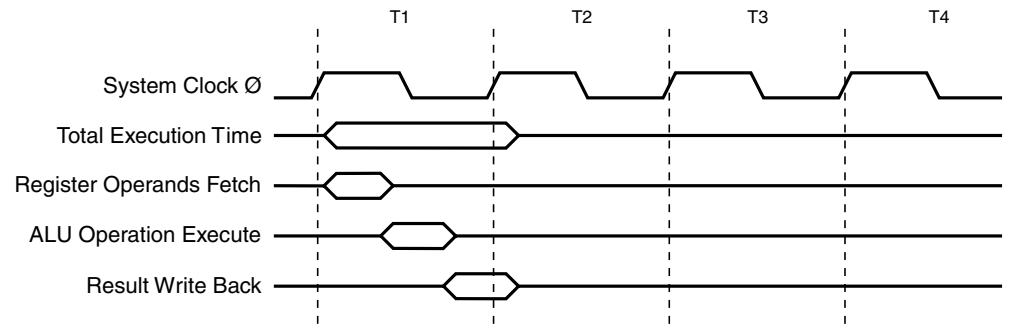


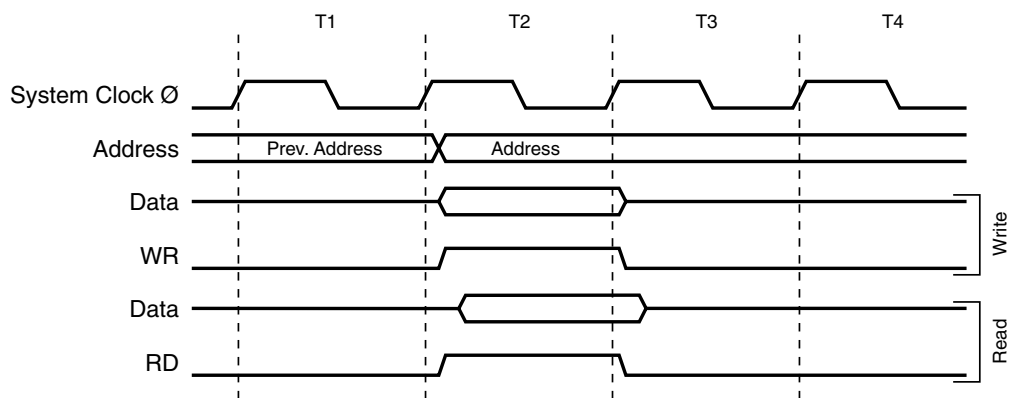
Figure 21 shows the internal timing concept for the register file. In a single clock cycle an ALU operation using two register operands is executed and the result is stored back to the destination register.

Figure 21. Single Cycle ALU Operation



The internal data SRAM access is performed in two System Clock cycles as described in Figure 22.

Figure 22. On-chip Data SRAM Access Cycles



I/O Memory

The I/O space definition of the AT90S8535 is shown in Table 1.

Table 1. AT90S8535 I/O Space

I/O Address (SRAM Address)	Name	Function
\$3F (\$5F)	SREG	Status REGISTER
\$3E (\$5E)	SPH	Stack Pointer High
\$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 general Control Register
\$34 (\$45)	MCUSR	MCU general Status Register
\$33 (\$53)	TCCR0	Timer/Counter0 Control Register
\$32 (\$52)	TCNT0	Timer/Counter0 (8-bit)
\$2F (\$4F)	TCCR1A	Timer/Counter1 Control Register A
\$2E (\$4E)	TCCR1B	Timer/Counter1 Control Register B
\$2D (\$4D)	TCNT1H	Timer/Counter1 High Byte
\$2C (\$4C)	TCNT1L	Timer/Counter1 Low Byte
\$2B (\$4B)	OCR1AH	Timer/Counter1 Output Compare Register A High Byte
\$2A (\$4A)	OCR1AL	Timer/Counter1 Output Compare Register A Low Byte
\$29 (\$49)	OCR1BH	Timer/Counter1 Output Compare Register B High Byte
\$28 (\$48)	OCR1BL	Timer/Counter1 Output Compare Register B Low Byte
\$27 (\$47)	ICR1H	T/C 1 Input Capture Register High Byte
\$26 (\$46)	ICR1L	T/C 1 Input Capture Register Low Byte
\$25 (\$45)	TCCR2	Timer/Counter2 Control Register
\$24 (\$44)	TCNT2	Timer/Counter2 (8-bit)
\$23 (\$43)	OCR2	Timer/Counter2 Output Compare Register
\$22 (\$42)	ASSR	Asynchronous Mode Status Register
\$21 (\$41)	WDTCR	Watchdog Timer Control Register
\$1F (\$3E)	EEARH	EEPROM Address Register High Byte
\$1E (\$3E)	EEARL	EEPROM Address Register Low Byte
\$1D (\$3D)	EEDR	EEPROM Data Register
\$1C (\$3C)	EECR	EEPROM Control Register
\$1B (\$3B)	PORTA	Data Register, Port A
\$1A (\$3A)	DDRA	Data Direction Register, Port A
\$19 (\$39)	PINA	Input Pins, Port A
\$18 (\$38)	PORTB	Data Register, Port B



Table 1. AT90S8535 I/O Space (Continued)

I/O Address (SRAM Address)	Name	Function
\$17 (\$37)	DDRB	Data Direction Register, Port B
\$16 (\$36)	PINB	Input Pins, Port B
\$15 (\$35)	PORTC	Data Register, Port C
\$14 (\$34)	DDRC	Data Direction Register, Port C
\$13 (\$33)	PINC	Input Pins, Port C
\$12 (\$32)	PORTD	Data Register, Port D
\$11 (\$31)	DDRD	Data Direction Register, Port D
\$10 (\$30)	PIND	Input Pins, Port D
\$0F (\$2F)	SPDR	SPI I/O Data Register
\$0E (\$2E)	SPSR	SPI Status Register
\$0D (\$2D)	SPCR	SPI Control Register
\$0C (\$2C)	UDR	UART I/O Data Register
\$0B (\$2B)	USR	UART Status Register
\$0A (\$2A)	UCR	UART Control Register
\$09 (\$29)	UBRR	UART Baud Rate Register
\$08 (\$28)	ACSR	Analog Comparator Control and Status Register
\$07 (\$27)	ADMUX	ADC Multiplexer Select Register
\$06 (\$26)	ADCSR	ADC Control and Status Register
\$05 (\$25)	ADCH	ADC Data Register High
\$04 (\$24)	ADCL	ADC Data Register Low

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

All AT90S8535 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 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 logic operation. See the Instruction Set description for detailed information.

• **Bit 0 – C: Carry Flag**

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

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



Stack Pointer – SP

The AT90S8535 Stack Pointer is implemented as two 8-bit registers in the I/O space locations \$3E (\$5E) and \$3D (\$5D). As the AT90S8535 data memory has \$25F locations, 10 bits are used.

Bit	15	14	13	12	11	10	9	8	
\$3E (\$5E)	–	–	–	–	–	–	SP9	SP8	SPH
\$3D (\$5D)	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	SPL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

The Stack Pointer points to the data SRAM stack area where the Subroutine and Interrupt stacks are located. This stack space in the data SRAM must be defined by the program before any subroutine calls are executed or interrupts are enabled. The Stack Pointer must be set to point above \$60. The Stack Pointer is decremented by 1 when data is pushed onto the stack with the PUSH instruction and it is decremented by 2 when an address is pushed onto the stack with subroutine calls and interrupts. The Stack Pointer is incremented by 1 when data is popped from the stack with the POP instruction and it is incremented by 2 when an address is popped from the stack with return from subroutine RET or return from interrupt RETI.

Reset and Interrupt Handling

The AT90S8535 provides 16 different interrupt sources. These interrupts and the separate reset vector each have a separate program vector in the program memory space. All interrupts are assigned individual enable bits that must be set (one) together with the I-bit in the Status Register in order to enable the interrupt.

The lowest addresses in the program memory space are automatically defined as the Reset and Interrupt vectors. The complete list of vectors is shown in Table 2. The list also determines the priority levels of the different interrupts. The lower the address, the higher the priority level. RESET has the highest priority, and next is INT0 (the External Interrupt Request 0), etc.

Table 2. Reset and Interrupt Vectors

Vector No.	Program Address	Source	Interrupt Definition
1	\$000	RESET	Hardware Pin, Power-on Reset and Watchdog Reset
2	\$001	INT0	External Interrupt Request 0
3	\$002	INT1	External Interrupt Request 1
4	\$003	TIMER2 COMP	Timer/Counter2 Compare Match
5	\$004	TIMER2 OVF	Timer/Counter2 Overflow
6	\$005	TIMER1 CAPT	Timer/Counter1 Capture Event
7	\$006	TIMER1 COMPA	Timer/Counter1 Compare Match A
8	\$007	TIMER1 COMPB	Timer/Counter1 Compare Match B
9	\$008	TIMER1 OVF	Timer/Counter1 Overflow
10	\$009	TIMER0 OVF	Timer/Counter0 Overflow
11	\$00A	SPI, STC	SPI Serial Transfer Complete
12	\$00B	UART, RX	UART, Rx Complete

Table 2. Reset and Interrupt Vectors (Continued)

Vector No.	Program Address	Source	Interrupt Definition
13	\$00C	UART, UDRE	UART Data Register Empty
14	\$00D	UART, TX	UART, Tx Complete
15	\$00E	ADC	ADC Conversion Complete
16	\$00F	EE_RDY	EEPROM Ready
17	\$010	ANA_COMP	Analog Comparator

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

Address Labels	Code	Comments
\$000	rjmp RESET	; Reset Handler
\$001	rjmp EXT_INT0	; IRQ0 Handler
\$002	rjmp EXT_INT1	; IRQ1 Handler
\$003	rjmp TIM2_COMP	; Timer2 Compare Handler
\$004	rjmp TIM2_OVF	; Timer2 Overflow Handler
\$005	rjmp TIM1_CAPT	; Timer1 Capture Handler
\$006	rjmp TIM1_COMP_A	; Timer1 CompareA Handler
\$007	rjmp TIM1_COMP_B	; Timer1 CompareB Handler
\$008	rjmp TIM1_OVF	; Timer1 Overflow Handler
\$009	rjmp TIM0_OVF	; Timer0 Overflow Handler
\$00a	rjmp SPI_STC;	; SPI Transfer Complete Handler
\$00b	rjmp UART_RXC	; UART RX Complete Handler
\$00c	rjmp UART_DRE	; UDR Empty Handler
\$00d	rjmp UART_TXC	; UART TX Complete Handler
\$00e	rjmp ADC	; ADC Conversion Complete Interrupt Handler
\$00f	rjmp EE_RDY	; EEPROM Ready Handler
\$010	rjmp ANA_COMP	; Analog Comparator Handler
\$011	MAIN: ldi r16, high(RAMEND);	Main program start
\$012	out SPH, r16	
\$013	ldi r16, low(RAMEND);	
\$014	out SPL, r16	
\$015	<instr> xxx	
...

Reset Sources

The AT90S8535 has three sources of reset:

- Power-on Reset. The MCU is reset when the supply voltage is below the Power-on Reset threshold (V_{POT}).
- External Reset. The MCU is reset when a low level is present on the 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 23 shows the reset logic. Table 3 defines the timing and electrical parameters of the reset circuitry.

Figure 23. Reset Logic

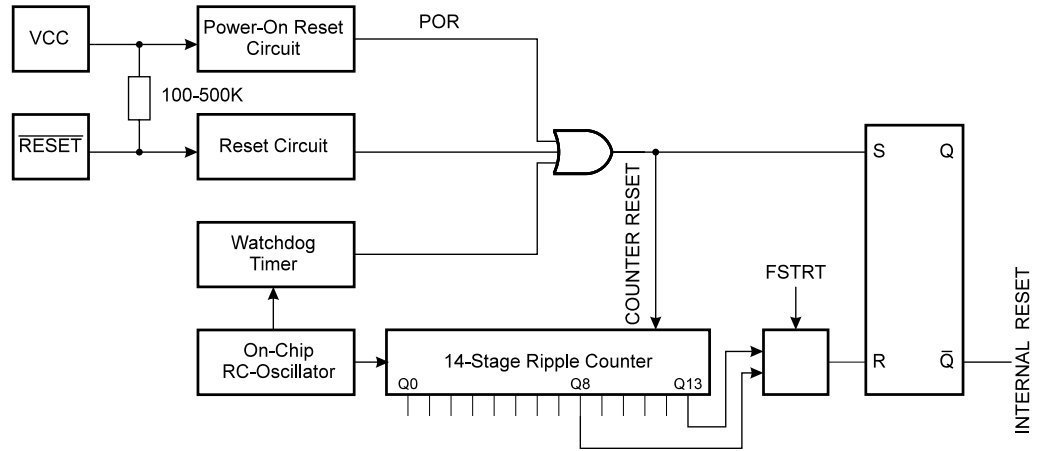


Table 3. Reset Characteristics ($V_{CC} = 5.0V$)

Symbol	Parameter	Min	Typ	Max	Units
$V_{POT}^{(1)}$	Power-on Reset Threshold (rising)	1.0	1.4	1.8	V
	Power-on Reset Threshold (falling)	0.4	0.6	0.8	V
V_{RST}	\overline{RESET} Pin Threshold Voltage		$0.6 V_{CC}$		V
t_{TOUT}	Reset Delay Time-out Period FSTRT Unprogrammed	11.0	16.0	21.0	ms
t_{TOUT}	Reset Delay Time-out Period FSTRT Programmed	1.0	1.1	1.2	ms

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

Table 4. Number of Watchdog Oscillator Cycles

FSTRT	Time-out at $V_{CC} = 5V$	Number of WDT Cycles
Programmed	1.1 ms	1K
Unprogrammed	16.0 ms	16K

Power-on Reset

A Power-on Reset (POR) circuit ensures that the device is reset from power-on. As shown in Figure 23, an internal timer clocked from the Watchdog Timer oscillator prevents the MCU from starting until after a certain period after V_{CC} has reached the Power-on Threshold voltage (V_{POT}), regardless of the V_{CC} rise time (see Figure 24).

The user can select the start-up time according to typical oscillator start-up time. The number of WDT oscillator cycles is shown in Table 4. The frequency of the Watchdog oscillator is voltage-dependent as shown in “Typical Characteristics” on page 107.

If the built-in start-up delay is sufficient, \overline{RESET} can be connected to V_{CC} directly or via an external pull-up resistor. By holding the pin low for a period after V_{CC} has been applied, the Power-on Reset period can be extended. Refer to Figure 25 for a timing example of this.

Figure 24. MCU Start-up, $\overline{\text{RESET}}$ Tied to V_{CC} .

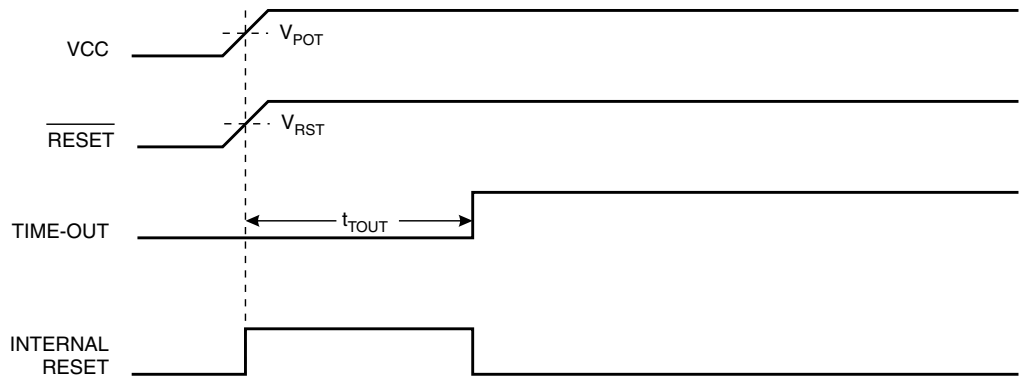
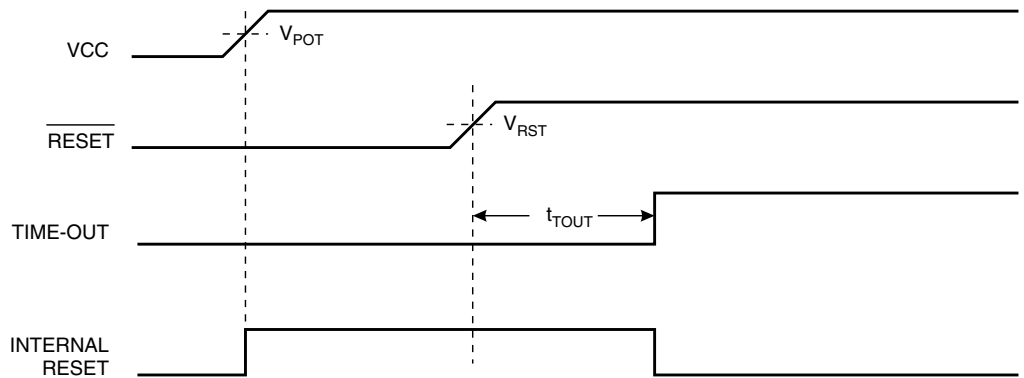


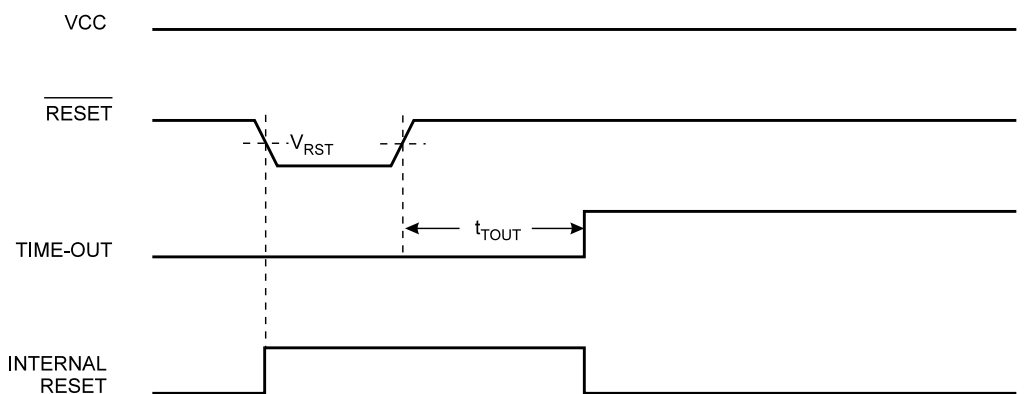
Figure 25. MCU Start-up, $\overline{\text{RESET}}$ Controlled Externally



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.

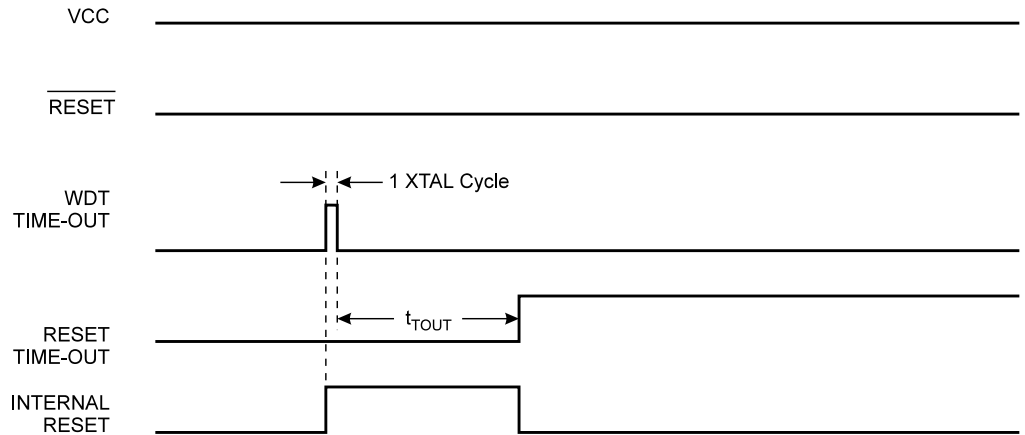
Figure 26. External Reset during Operation



Watchdog Reset

When the Watchdog times out, it will generate a short reset pulse of 1 XTAL cycle duration. On the falling edge of this pulse, the delay timer starts counting the Time-out period t_{TOUT} . Refer to page 49 for details on operation of the Watchdog.

Figure 27. Watchdog Reset during Operation



MCU Status Register – MCUSR

The MCU Status Register provides information on which reset source caused an MCU reset.

Bit	7	6	5	4	3	2	1	0	
\$34 (\$54)	–	–	–	–	–	–	EXTRF	PORF	MCUSR
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	See Bit Description		

- **Bits 7..2 – Res: Reserved Bits**

These bits are reserved bits in the AT90S8535 and always read as zero.

- **Bit 1 – EXTRF: External Reset Flag**

After a power-on reset, this bit is undefined (X). It can only be set by an External Reset. A Watchdog Reset will leave this bit unchanged. The bit is cleared by writing a logical zero to the bit.

- **Bit 0 – PORF: Power-on Reset Flag**

This bit is only set by a Power-on Reset. A Watchdog Reset or an External Reset will leave this bit unchanged. The bit is cleared by writing a logical zero to the bit.

To summarize, Table 5 shows the value of these two bits after the three modes of reset.

Table 5. PORF and EXTRF Values after Reset

Reset Source	EXTRF	PORF
Power-on Reset	Undefined	1
External Reset	1	Unchanged
Watchdog Reset	Unchanged	Unchanged

To make use of these bits to identify a reset condition, the user software should clear both the PORF and EXTRF bits as early as possible in the program. Checking the PORF and EXTRF values is done before the bits are cleared. If the bit is cleared before an External or Watchdog Reset occurs, the source of reset can be found by using Table 6.

Table 6. Reset Source Identification

EXTRF	PORF	Reset Source
0	0	Watchdog Reset
0	1	Power-on Reset
1	0	External Reset
1	1	Power-on Reset

Interrupt Handling

The AT90S8535 has two 8-bit interrupt mask control registers: GIMSK (General Interrupt Mask register) and TIMSK (Timer/Counter Interrupt Mask register).

When an interrupt occurs, the Global Interrupt Enable I-bit is cleared (zero) and all interrupts are disabled. The user software can set (one) the I-bit to enable nested interrupts. The I-bit is set (one) when a Return from Interrupt instruction (RETI) is executed.

When the Program Counter is vectored to the actual interrupt vector in order to execute the interrupt handling routine, hardware clears the corresponding flag that generated the interrupt. Some of the interrupt flags can also be cleared by writing a logical “1” to the flag bit position(s) to be cleared. If an interrupt condition occurs when the corresponding interrupt enable bit is cleared (zero), the interrupt flag will be set and remembered until the interrupt is enabled or the flag is cleared by software.

If one or more interrupt conditions occur when the global interrupt enable bit is cleared (zero), the corresponding interrupt flag(s) will be set and remembered until the global interrupt enable bit is set (one) and will be executed by order of priority.

Note that external level interrupt does not have a flag and will only be remembered for as long as the interrupt condition is active.

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

General Interrupt Mask Register – GIMSK

Bit	7	6	5	4	3	2	1	0	
\$3B (\$5B)	INT1	INT0	–	–	–	–	–	–	GIMSK
Read/Write	R/W	R/W	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

• **Bit 7 – INT1: External Interrupt Request 1 Enable**

When the INT1 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), the external pin interrupt is enabled. The Interrupt Sense Control1 bits 1/0 (ISC11 and ISC10) in the MCU general Control Register (MCUCR) define whether the external interrupt is activated on rising or falling edge of the INT1 pin or level sensed. Activity on the pin will cause an interrupt request even if INT1 is configured as an output. The corresponding interrupt of External Interrupt Request 1 is executed from program memory address \$002. See also “External Interrupts.”

• **Bit 6 – INT0: External Interrupt Request 0 Enable**

When the INT0 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), the external pin interrupt is enabled. The Interrupt Sense Control0 bits 1/0 (ISC01 and ISC00) in the MCU general Control Register (MCUCR) define whether the external interrupt is activated on rising or falling edge of the INT0 pin or level sensed. Activity on the pin will cause an interrupt request even if INT0 is configured as an output. The corre-

