



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

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

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

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

#### Details

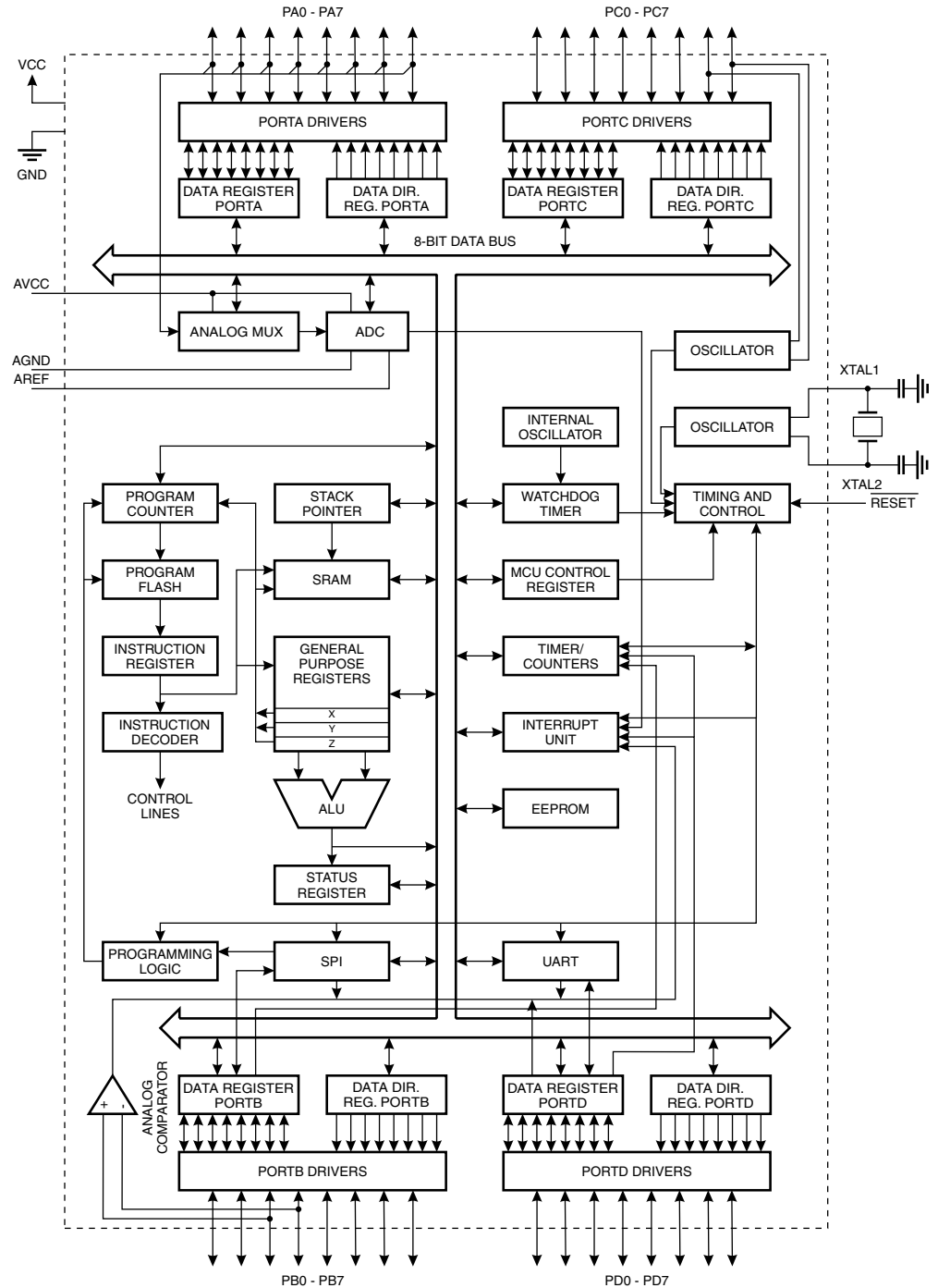
Product Status	Obsolete
Core Processor	AVR
Core Size	8-Bit
Speed	4MHz
Connectivity	SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	32
Program Memory Size	8KB (4K x 16)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 6V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-TQFP
Supplier Device Package	44-TQFP (10x10)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/at90ls8535-4ai">https://www.e-xfl.com/product-detail/microchip-technology/at90ls8535-4ai</a>

## 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

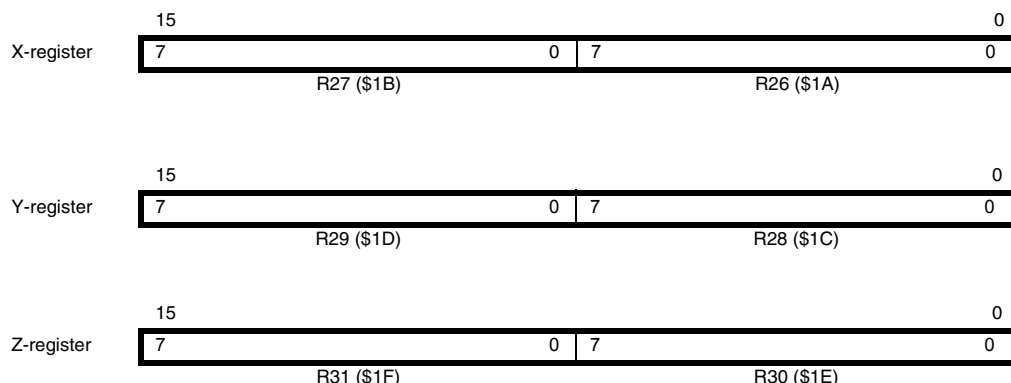
Figure 1. The AT90S8535 Block Diagram



## 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

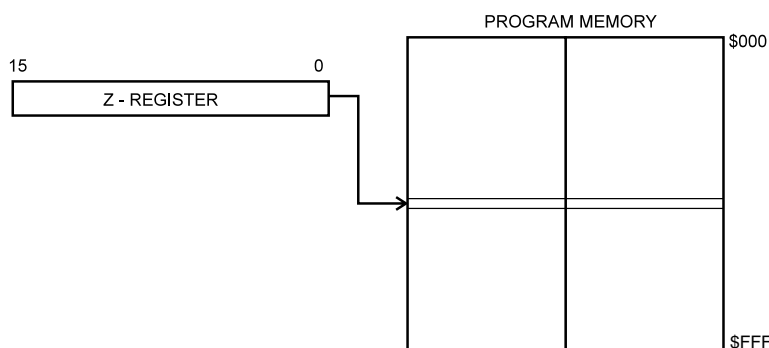
The AT90S8535 contains 8K bytes On-chip, In-System Programmable Flash memory for program storage. Since all instructions are 16- or 32-bit words, the Flash is organized as 4K x 16. The Flash memory has an endurance of at least 1000 write/erase cycles. The AT90S8535 Program Counter (PC) is 12 bits wide, 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.

## Indirect Program Addressing, IJMP and ICALL

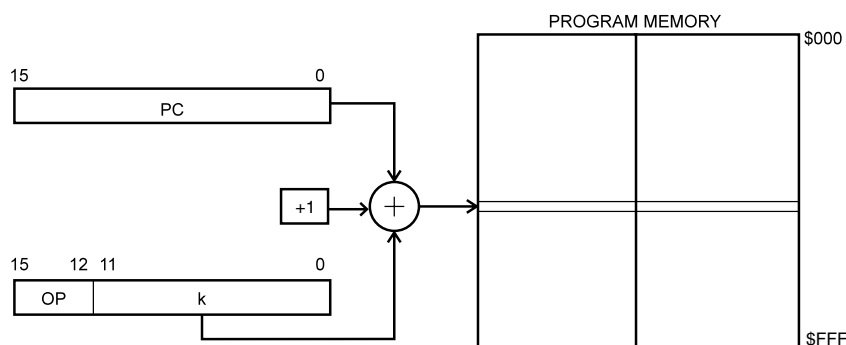
**Figure 18.** Indirect Program Memory Addressing



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  $\phi$ , 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.

**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.

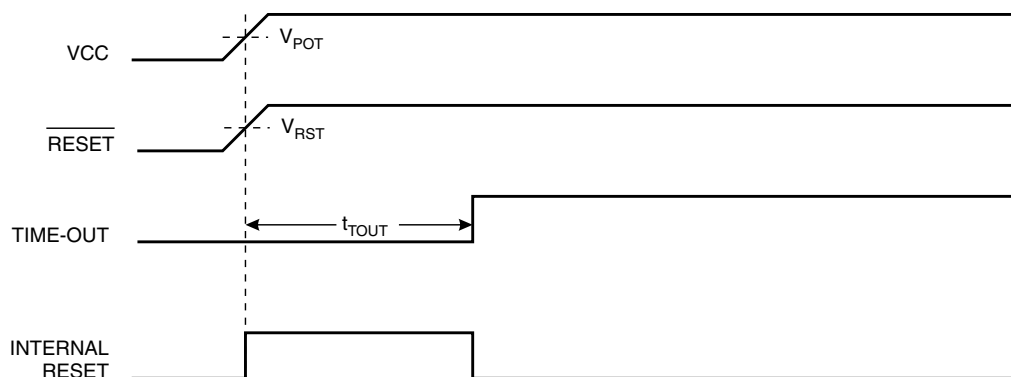
All AT90S8535 I/Os and peripherals are placed in the I/O space. The I/O locations are accessed by the IN and OUT instructions transferring data between the 32 general-purpose working registers and the I/O space. I/O registers within the address range \$00 - \$1F are directly bit-accessible using the 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.

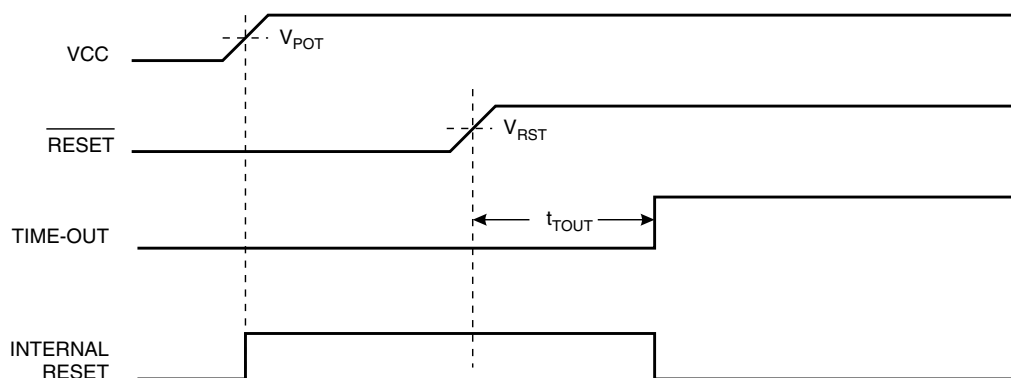
Some of the status flags are cleared by writing a logical “1” to them. Note that the CBI and SBI instructions will operate on all bits in the I/O register, writing a “1” back into any flag read as set, thus clearing the flag. The CBI and SBI instructions work with registers \$00 to \$1F only.

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

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



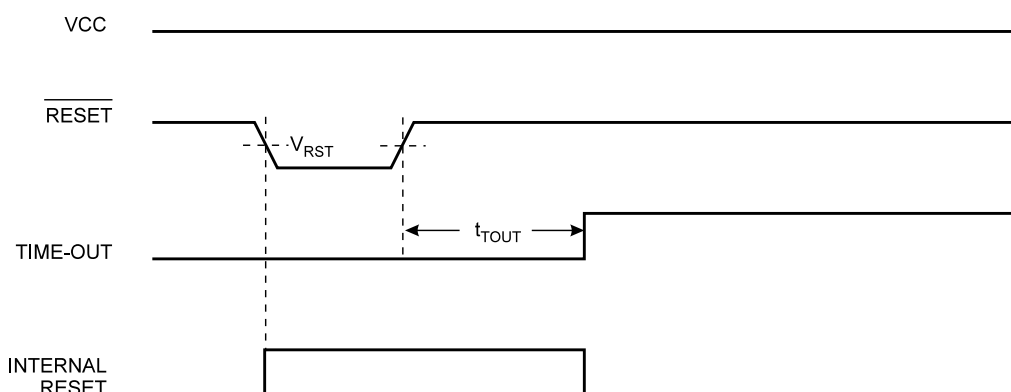
**Figure 25.** MCU Start-up,  $\overline{\text{RESET}}$  Controlled Externally



## 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



**Table 11.** Compare 1 Mode Select

COM1X1	COM1X0	Description
0	0	Timer/Counter1 disconnected from output pin OC1X
0	1	Toggle the OC1X output line.
1	0	Clear the OC1X output line (to zero).
1	1	Set the OC1X output line (to one).

Note: X = A or B.

In PWM mode, these bits have a different function. Refer to Table 15 for a detailed description. When changing the COM1X1/COM1X0 bits, Output Compare Interrupt 1 must be disabled by clearing their Interrupt Enable bits in the TIMSK Register. Otherwise an interrupt can occur when the bits are changed.

- Bits 3..2 – Res: Reserved Bits**

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

- Bits 1..0 – PWM11, PWM10: Pulse Width Modulator Select Bits**

These bits select PWM operation of Timer/Counter1 as specified in Table 12. This mode is described on page 40.

**Table 12.** PWM Mode Select

PWM11	PWM10	Description
0	0	PWM operation of Timer/Counter1 is disabled
0	1	Timer/Counter1 is an 8-bit PWM
1	0	Timer/Counter1 is a 9-bit PWM
1	1	Timer/Counter1 is a 10-bit PWM

## Timer/Counter1 Control Register B – TCCR1B

Bit	7	6	5	4	3	2	1	0	
\$2E (\$4E)	ICNC1	ICES1	–	–	CTC1	CS12	CS11	CS10	TCCR1B
Read/Write	R/W	R/W	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- Bit 7 – ICNC1: Input Capture1 Noise Canceler (4 CKs)**

When the ICNC1 bit is cleared (zero), the Input Capture Trigger Noise Canceler function is disabled. The input capture is triggered at the first rising/falling edge sampled on the ICP (input capture pin) as specified. When the ICNC1 bit is set (one), four successive samples are measured on the ICP (input capture pin), and all samples must be high/low according to the input capture trigger specification in the ICES1 bit. The actual sampling frequency is XTAL clock frequency.

- Bit 6 – ICES1: Input Capture1 Edge Select**

While the ICES1 bit is cleared (zero), the Timer/Counter1 contents are transferred to the Input Capture Register (ICR1) on the falling edge of the input capture pin (ICP). While the ICES1 bit is set (one), the Timer/Counter1 contents are transferred to the Input Capture Register (ICR1) on the rising edge of the input capture pin (ICP).

- Bits 5, 4 – Res: Reserved Bits**

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

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

The EEMWE bit determines whether setting EEW to “1” causes the EEPROM to be written. When EEMWE is set (one), setting EEW will write data to the EEPROM at the selected address. If EEMWE is zero, setting EEW 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 EEW bit for a EEPROM write procedure.

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

The EEPROM Write Enable signal (EEW) is the write strobe to the EEPROM. When address and data are correctly set up, the EEW bit must be set to write the value into the EEPROM. The EEMWE bit must be set when the logical “1” is written to EEW, 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 EEW becomes zero.
2. Write new EEPROM address to EEARL and EEARH (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 EEW bit must be written to “0” in the same cycle).
5. Within four clock cycles after setting EEMWE, write a logical “1” to EEW.

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 EEW bit is cleared (zero) by hardware. The user software can poll this bit and wait for a zero before writing the next byte. When EEW has been set, the CPU is halted for two clock cycles before the next instruction is executed.

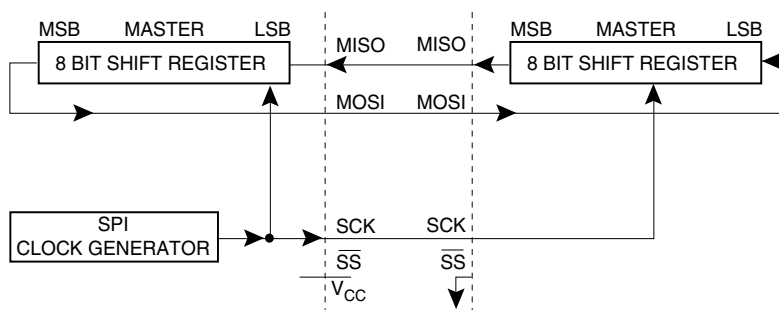
- **Bit 0 – EERE: EEPROM Read Enable**

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 clock cycles before the next instruction is executed.

The user should poll the EEW 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.



**Figure 38.** SPI Master-slave Interconnection



The system is single-buffered in the transmit direction and double-buffered in the receive direction. This means that bytes to be transmitted cannot be written to the SPI Data Register before the entire shift cycle is completed. When receiving data, however, a received byte must be read from the SPI Data Register before the next byte has been completely shifted in. Otherwise, the first byte is lost.

When the SPI is enabled, the data direction of the MOSI, MISO, SCK and  $\overline{SS}$  pins is overridden according to Table 22.

**Table 22.** SPI Pin Overrides

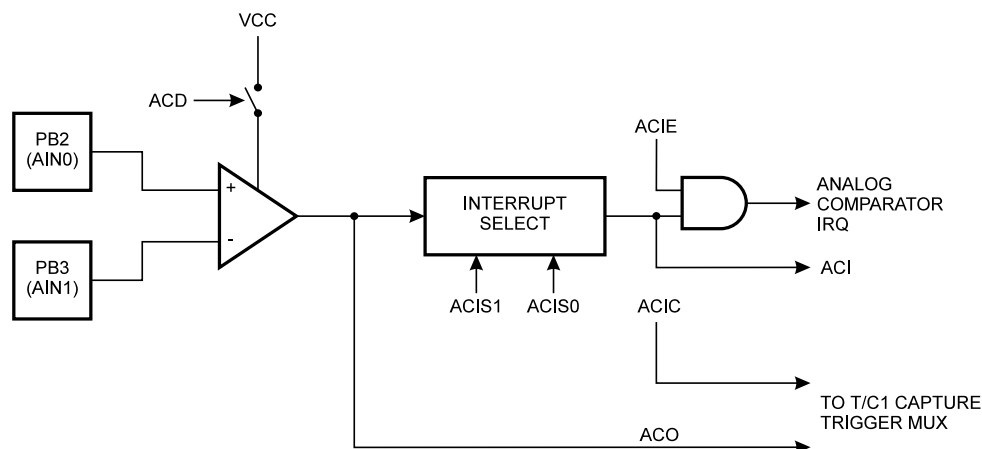
Pin	Direction, Master SPI	Direction, Slave SPI
MOSI	User Defined	Input
MISO	Input	User Defined
SCK	User Defined	Input
$\overline{SS}$	User Defined	Input

Note: See "Alternate Functions of Port B" on page 79 for a detailed description of how to define the direction of the user-defined SPI pins.

## Analog Comparator

The Analog Comparator compares the input values on the positive input PB2 (AIN0) and negative input PB3 (AIN1). When the voltage on the positive input PB2 (AIN0) is higher than the voltage on the negative input PB3 (AIN1), the Analog Comparator Output (ACO) is set (one). The comparator's output can be set to trigger the Timer/Counter1 Input Capture function. In addition, the comparator can trigger a separate interrupt, exclusive to the Analog Comparator. The user can select Interrupt triggering on comparator output rise, fall or toggle. A block diagram of the comparator and its surrounding logic is shown in Figure 44.

**Figure 44.** Analog Comparator Block Diagram



### Analog Comparator Control and Status Register – ACSR

Bit	7	6	5	4	3	2	1	0	
\$08 (\$28)	ACD	–	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	ACSR
Read/Write	R/W	R	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	N/A	0	0	0	0	0	

- **Bit 7 – ACD: Analog Comparator Disable**

When this bit is set (one), the power to the Analog Comparator is switched off. This bit can be set at any time to turn off the Analog Comparator. When changing the ACD bit, the Analog Comparator interrupt must be disabled by clearing the ACIE bit in ACSR. Otherwise an interrupt can occur when the bit is changed.

- **Bit 6 – Res: Reserved Bit**

This bit is a reserved bit in the AT90S8535 and will always read as zero.

- **Bit 5 – ACO: Analog Comparator Output**

ACO is directly connected to the comparator output.

- **Bit 4 – ACI: Analog Comparator Interrupt Flag**

This bit is set (one) when a comparator output event triggers the interrupt mode defined by ACI1 and ACI0. The Analog Comparator Interrupt routine is executed if the ACIE bit is set (one) and the I-bit in SREG is set (one). ACI is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, ACI is cleared by writing a logical “1” to the flag.

- **Bit 3 – ACIE: Analog Comparator Interrupt Enable**

When the ACIE bit is set (one) and the I-bit in the Status Register is set (one), the Analog Comparator interrupt is activated. When cleared (zero), the interrupt is disabled.

2. Enter Idle Mode. The ADC will start a conversion once the CPU has been halted.
3. If no other interrupts occur before the ADC conversion completes, the ADC interrupt will wake up the MCU and execute the ADC Conversion Complete Interrupt routine.

### ADC Multiplexer Select Register – ADMUX

Bit	7	6	5	4	3	2	1	0	
\$07 (\$27)	–	–	–	–	–	MUX2	MUX1	MUX0	ADMUX
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 AT90S8535 and always read as zero.

#### • Bits 2..0 – MUX2..MUX0: Analog Channel Select Bits 2-0

The value of these three bits selects which analog input ADC7..0 is connected to the ADC. See Table 27 for details.

If these bits are changed during a conversion, the change will not go into effect until this conversion is complete (ADIF in ADCSR is set).

**Table 27.** Input Channel Selections

MUX2..0	Single-ended Input
000	ADC0
001	ADC1
010	ADC2
011	ADC3
100	ADC4
101	ADC5
110	ADC6
111	ADC7

### ADC Control and Status Register – ADCSR

Bit	7	6	5	4	3	2	1	0	
\$06 (\$26)	ADEN	ADSC	ADFR	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ADCSR
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 – ADEN: ADC Enable

Writing a logical “1” to this bit enables the ADC. By clearing this bit to zero, the ADC is turned off. Turning the ADC off while a conversion is in progress will terminate this conversion.

#### • Bit 6 – ADSC: ADC Start Conversion

In Single Conversion Mode, a logical “1” must be written to this bit to start each conversion. In Free Running Mode, a logical “1” must be written to this bit to start the first conversion. The first time ADSC has been written after the ADC has been enabled or if ADSC is written at the same time as the ADC is enabled, an extended conversion will precede the initiated conversion. This extended conversion performs initialization of the ADC.

### • ADC9..0: ADC Conversion result

These bits represent the result from the conversion. \$000 represents analog ground and \$3FF represents the selected reference voltage minus one LSB.

## Scanning Multiple Channels

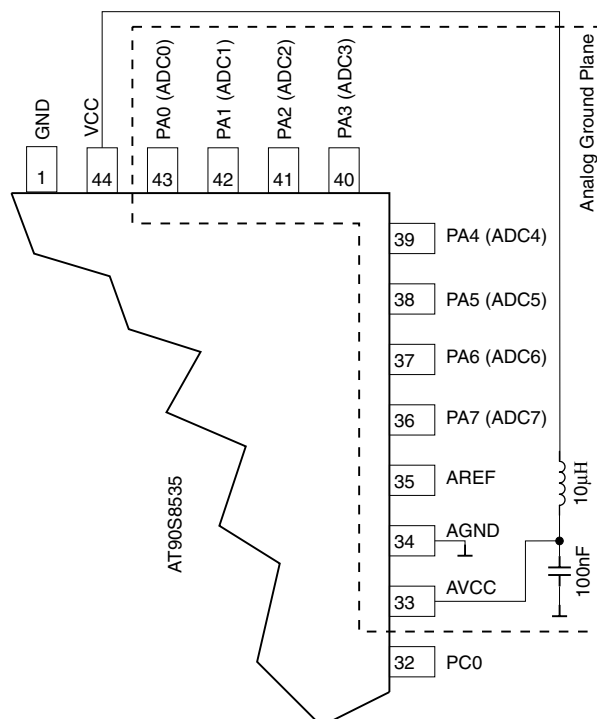
Since change of analog channel always is delayed until a conversion is finished, the Free Running Mode can be used to scan multiple channels without interrupting the converter. Typically, the ADC Conversion Complete interrupt will be used to perform the channel shift. However, the user should take the following fact into consideration: The interrupt triggers once the result is ready to be read. In Free Running Mode, the next conversion will start immediately when the interrupt triggers. If ADMUX is changed after the interrupt triggers, the next conversion has already started and the old setting is used.

## ADC Noise Canceling Techniques

Digital circuitry inside and outside the AT90S8535 generates EMI that might affect the accuracy of analog measurements. If conversion accuracy is critical, the noise level can be reduced by applying the following techniques:

1. The analog part of the AT90S8535 and all analog components in the application should have a separate analog ground plane on the PCB. This ground plane is connected to the digital ground plane via a single point on the PCB.
2. Keep analog signal paths as short as possible. Make sure analog tracks run over the analog ground plane and keep them well away from high-speed switching digital tracks.
3. The  $AV_{CC}$  pin on the AT90S8535 should be connected to the digital  $V_{CC}$  supply voltage via an LC network as shown in Figure 50.
4. Use the ADC noise canceler function to reduce induced noise from the CPU.
5. If some Port A pins are used as digital outputs, it is essential that these do not switch while a conversion is in progress.

**Figure 50.** ADC Power Connections

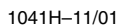


PAn, general I/O pin: The DDAn bit in the DDRA register selects the direction of this pin. If DDAn is set (one), PAn is configured as an output pin. If DDAn is cleared (zero), PAn is configured as an input pin. If PORTAn is set (one) when the pin is configured as an input pin, the MOS pull-up resistor is activated. To switch the pull-up resistor off, the PORTAn has to be cleared (zero) or the pin has to be configured as an output pin. The port pins are tri-stated when a reset condition becomes active, even if the clock is not running.

DDAn	PORTAn	I/O	Pull-up	Comment
0	0	Input	No	Tri-state (high-Z)
0	1	Input	Yes	PAn will source current if ext. pulled low
1	0	Output	No	Push-pull Zero Output
1	1	Output	No	Push-pull One Output

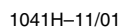
## Port A Schematics

**Figure 51. Port A Schematic Diagrams (Pins PA0 - PA7)**

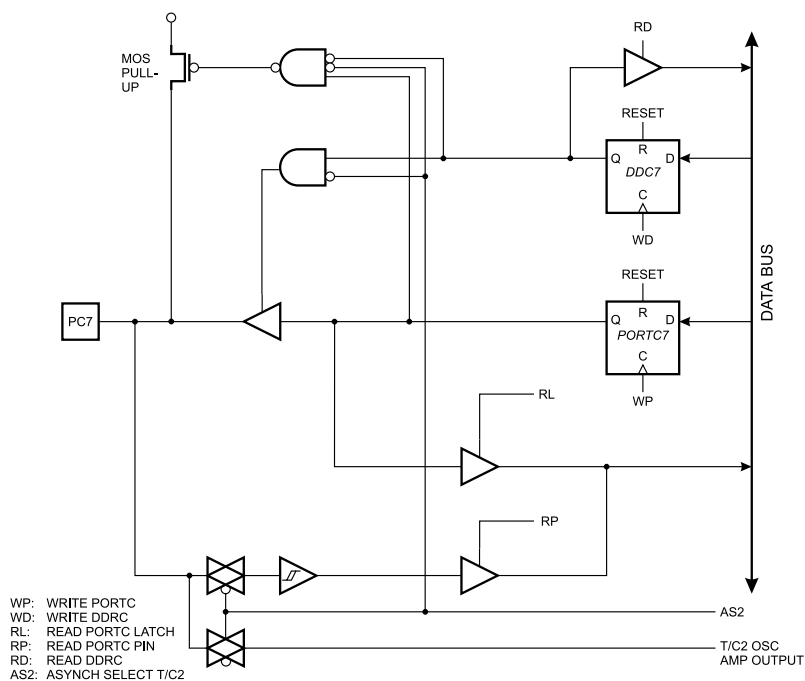


## Port C Schematics

**Figure 58.** Port C Schematic Diagram (Pins PC0 - PC5)



**Figure 60.** Port C Schematic Diagram (Pins PC7)



## Port D

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors.

Three I/O memory address locations are allocated for Port D, one each for the Data Register – PORTD, \$12(\$32), Data Direction Register – DDRD, \$11(\$31) and the Port D Input Pins – PIND, \$10(\$30). The Port D Input Pins address is read-only, while the Data Register and the Data Direction Register are read/write.

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. Some Port D pins have alternate functions as shown in Table 33.

### Table 33. Port D Pin Alternate Functions

Port Pin	Alternate Function
PD0	RXD (UART Input line)
PD1	TXD (UART Output line)
PD2	INT0 (External interrupt 0 input)
PD3	INT1 (External interrupt 1 input)
PD4	OC1B (Timer/Counter1 output compareB match output)
PD5	OC1A (Timer/Counter1 output compareA match output)
PD6	ICP (Timer/Counter1 input capture pin)
PD7	OC2 (Timer/Counter2 output compare match output)

Figure 65. Port D Schematic Diagram (Pin PD6)

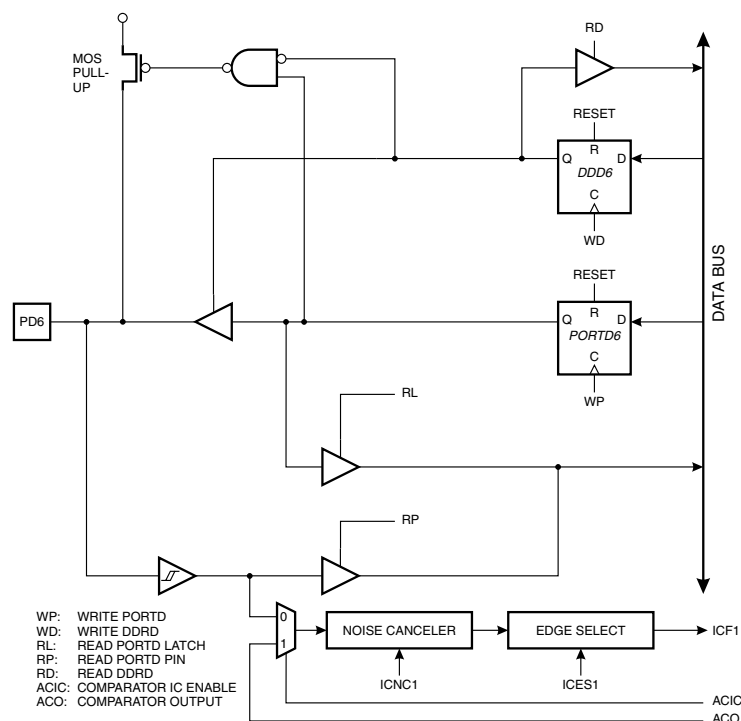
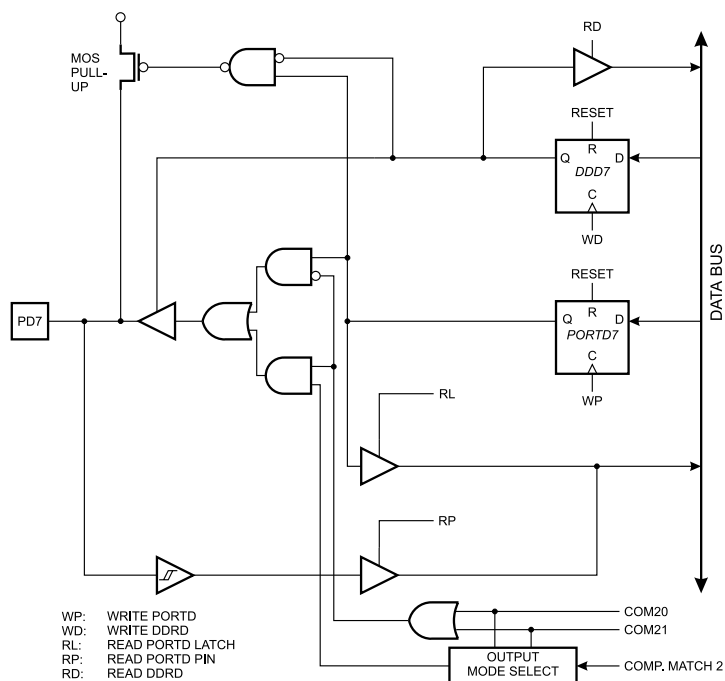


Figure 66. Port D Schematic Diagram (Pin PD7)





**Table 37.** Pin Name Mapping

Signal Name in Programming Mode	Pin Name	I/O	Function
RDY/BSY	PD1	O	0: Device is busy programming, 1: Device is ready for new command
$\overline{OE}$	PD2	I	Output Enable (Active low)
$\overline{WR}$	PD3	I	Write Pulse (Active low)
BS	PD4	I	Byte Select ("0" selects low byte, "1" selects high byte)
XA0	PD5	I	XTAL Action Bit 0
XA1	PD6	I	XTAL Action Bit 1
DATA	PB7 - 0	I/O	Bi-directional Data Bus (Output when $\overline{OE}$ is low)

**Table 38.** XA1 and XA0 Coding

XA1	XA0	Action when XTAL1 is Pulsed
0	0	Load Flash or EEPROM Address (high or low address byte determined by BS)
0	1	Load Data (High or low data byte for Flash determined by BS)
1	0	Load Command
1	1	No Action, Idle

**Table 39.** Command Byte Bit Coding

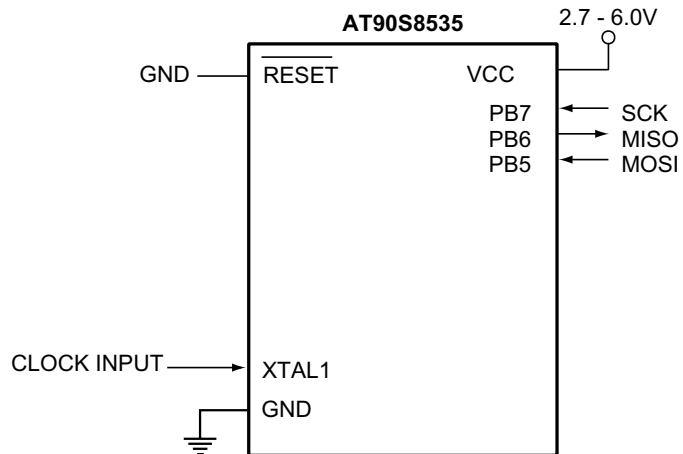
Command Byte	Command Executed
1000 0000	Chip Erase
0100 0000	Write Fuse Bits
0010 0000	Write Lock Bits
0001 0000	Write Flash
0001 0001	Write EEPROM
0000 1000	Read Signature Bytes
0000 0100	Read Lock and Fuse Bits
0000 0010	Read Flash
0000 0011	Read EEPROM

### Enter Programming Mode

The following algorithm puts the device in Parallel Programming Mode:

1. Apply supply voltage according to Table 36, between  $V_{CC}$  and GND.
2. Set the  $\overline{RESET}$  and BS pin to "0" and wait at least 100 ns.
3. Apply 11.5 - 12.5V to  $\overline{RESET}$ . Any activity on BS within 100 ns after +12V has been applied to  $\overline{RESET}$ , will cause the device to fail entering programming mode.

**Figure 71.** 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 \$0FFF for program memory and \$0000 to \$01FF for EEPROM memory.

Either an external clock is supplied at pin XTAL1 or a crystal needs to be connected across pins XTAL1 and XTAL2. The minimum low and high periods for the serial clock (SCK) input are defined as follows:

Low: > 2 XTAL1 clock cycles

High: > 2 XTAL1 clock cycles

## Serial Programming Algorithm

When writing serial data to the AT90S8535, data is clocked on the rising edge of SCK.

When reading data from the AT90S8535, data is clocked on the falling edge of SCK. See Figure 72, Figure 73 and Table 43 for timing details.

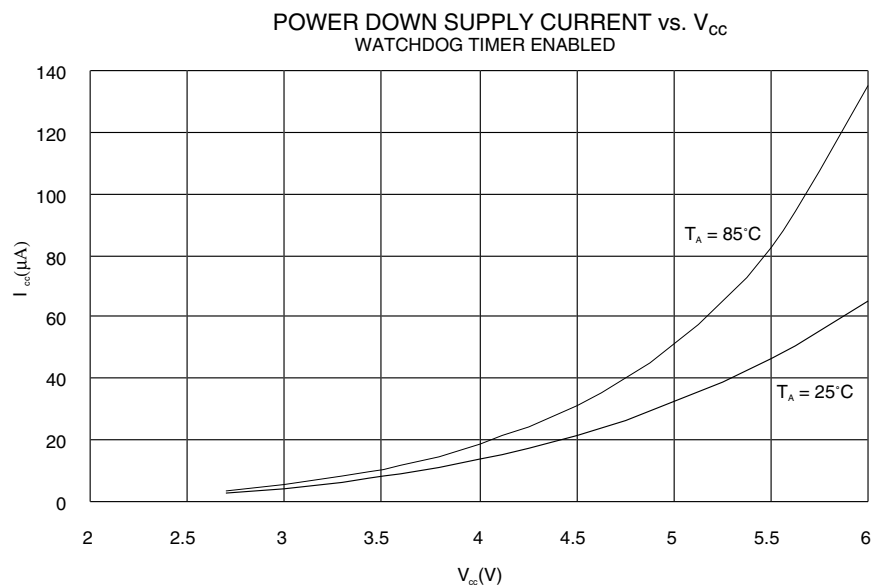
To program and verify the AT90SS8535 in the Serial Programming Mode, the following sequence is recommended (see 4-byte instruction formats in Table 42):

1. Power-up sequence:

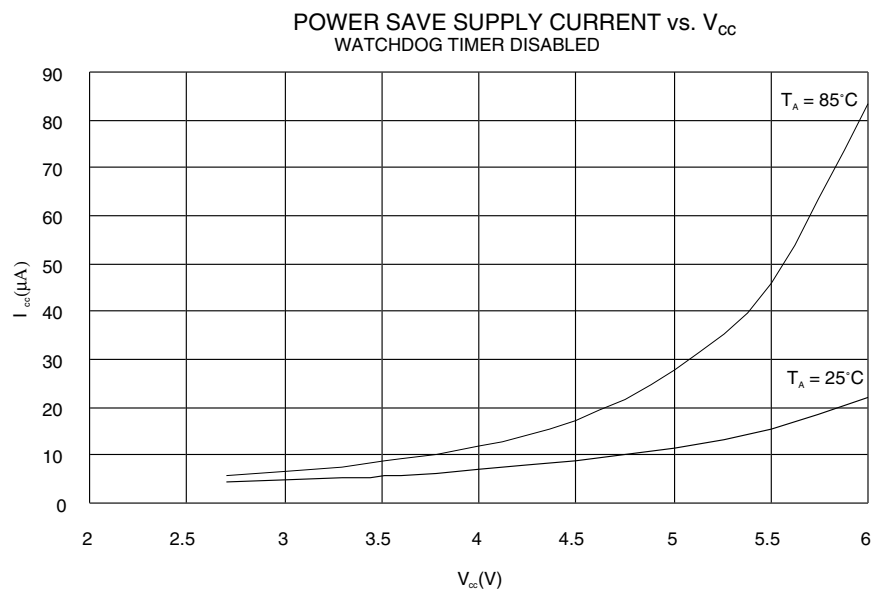
Apply power between  $V_{CC}$  and GND while  $\overline{RESET}$  and SCK are set to "0". If a crystal is not connected across pins XTAL1 and XTAL2, apply a clock signal to the XTAL1 pin. In some systems, the programmer cannot guarantee that SCK is held low during power-up. In this case,  $\overline{RESET}$  must be given a positive pulse of at least two XTAL1 cycles duration after SCK has been set to "0".

2. Wait for at least 20 ms and enable serial programming by sending the Programming Enable serial instruction to the MOSI (PB5) pin.
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.

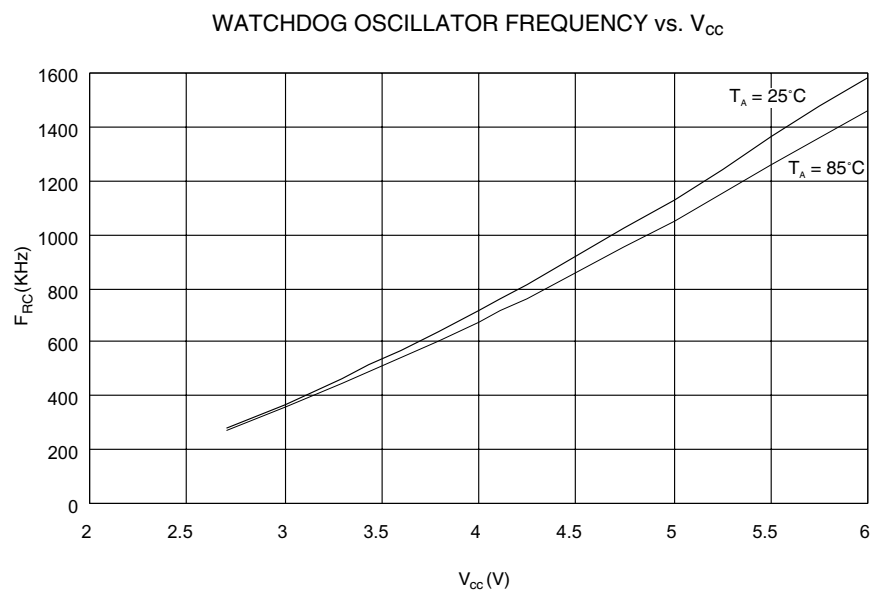
**Figure 80.** Power-down Supply Current vs.  $V_{CC}$



**Figure 81.** Power Save Supply Current vs.  $V_{CC}$

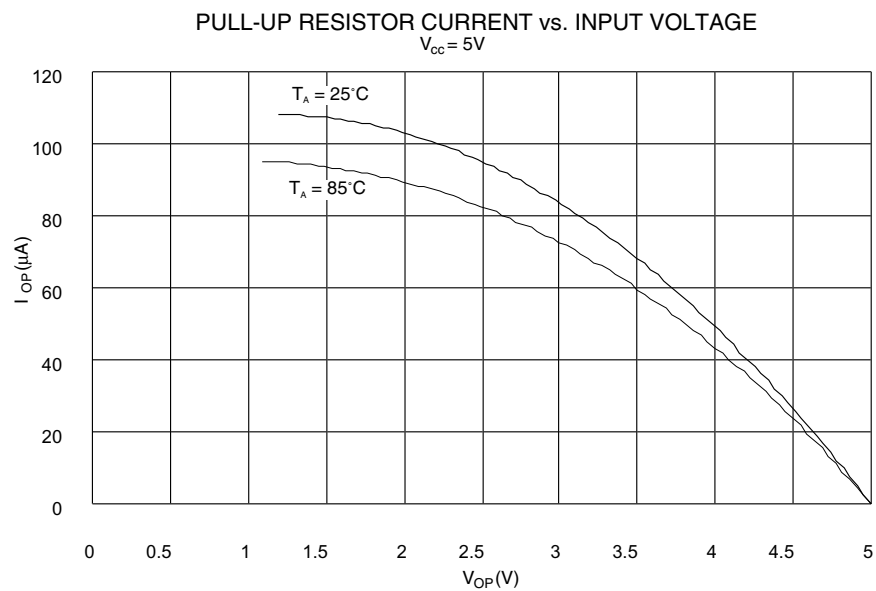


**Figure 86.** Watchdog Oscillator Frequency vs.  $V_{CC}$



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

**Figure 87.** Pull-up Resistor Current vs. Input Voltage





## Register Summary

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
\$3F (\$5F)	SREG	I	T	H	S	V	N	Z	C	page 19
\$3E (\$5E)	SPH	-	-	-	-	-	-	SP9	SP8	page 20
\$3D (\$5D)	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	page 20
\$3C (\$5C)	Reserved									
\$3B (\$5B)	GIMSK	INT1	INT0	-	-	-	-	-	-	page 25
\$3A (\$5A)	GIFR	INTF1	INTF0							page 26
\$39 (\$59)	TIMSK	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	-	TOIE0	page 26
\$38 (\$58)	TIFR	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	-	TOV0	page 27
\$37 (\$57)	Reserved									
\$36 (\$56)	Reserved									
\$35 (\$55)	MCUCR	-	SE	SM1	SM0	ISC11	ISC10	ISC01	ISC00	page 29
\$34 (\$54)	MCUSR	-	-	-	-	-	-	EXTRF	PORF	page 24
\$33 (\$53)	TCCR0	-	-	-	-	-	CS02	CS01	CS00	page 34
\$32 (\$52)	TCNT0	Timer/Counter0 (8 Bits)								page 34
\$31 (\$51)	Reserved									
\$30 (\$50)	Reserved									
\$2F (\$4F)	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	-	-	PWM11	PWM10	page 36
\$2E (\$4E)	TCCR1B	ICNC1	ICES1	-	-	CTC1	CS12	CS11	CS10	page 37
\$2D (\$4D)	TCNT1H	Timer/Counter1 – Counter Register High Byte								page 38
\$2C (\$4C)	TCNT1L	Timer/Counter1 – Counter Register Low Byte								page 38
\$2B (\$4B)	OCR1AH	Timer/Counter1 – Output Compare Register A High Byte								page 39
\$2A (\$4A)	OCR1AL	Timer/Counter1 – Output Compare Register A Low Byte								page 39
\$29 (\$49)	OCR1BH	Timer/Counter1 – Output Compare Register B High Byte								page 39
\$28 (\$48)	OCR1BL	Timer/Counter1 – Output Compare Register B Low Byte								page 39
\$27 (\$47)	ICR1H	Timer/Counter1 – Input Capture Register High Byte								page 40
\$26 (\$46)	ICR1L	Timer/Counter1 – Input Capture Register Low Byte								page 40
\$25 (\$45)	TCCR2	-	PWM2	COM21	COM20	CTC2	CS22	CS21	CS20	page 43
\$24 (\$44)	TCNT2	Timer/Counter2 (8 Bits)								page 44
\$23 (\$43)	OCR2	Timer/Counter2 Output Compare Register								page 44
\$22 (\$42)	ASSR	-	-	-	-	AS2	TCN2UB	OCR2UB	TCR2UB	page 46
\$21 (\$41)	WDTCR	-	-	-	WDT0E	WDE	WDP2	WDP1	WDP0	page 49
\$20 (\$40)	Reserved									
\$1F (\$3F)	EEARH									page 51
\$1E (\$3E)	EEARL	EEAR7	EEAR6	EEAR5	EEAR4	EEAR3	EEAR2	EEAR1	EEAR0	page 51
\$1D (\$3D)	EEDR	EEPROM Data Register								page 51
\$1C (\$3C)	EEDR	-	-	-	-	EERIE	EEMWE	EEWE	EERE	page 51
\$1B (\$3B)	PORTA	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	page 76
\$1A (\$3A)	DDRA	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	page 76
\$19 (\$39)	PINA	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	page 76
\$18 (\$38)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	page 78
\$17 (\$37)	DDRB	ddb7	ddb6	ddb5	ddb4	ddb3	ddb2	ddb1	ddb0	page 78
\$16 (\$36)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	page 78
\$15 (\$35)	PORTC	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	page 84
\$14 (\$34)	DDRC	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	page 84
\$13 (\$33)	PINC	PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	page 84
\$12 (\$32)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	page 87
\$11 (\$31)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	page 87
\$10 (\$30)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	page 87
\$0F (\$2F)	SPDR	SPI Data Register								page 58
\$0E (\$2E)	SPSR	SPIF	WCOL	-	-	-	-	-	-	page 58
\$0D (\$2D)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	page 57
\$0C (\$2C)	UDR	UART I/O Data Register								page 62
\$0B (\$2B)	USR	RXC	TXC	UDRE	FE	OR	-	-	-	page 62
\$0A (\$2A)	UCR	RXCIE	TXCIE	UDRIE	RXEN	TXEN	CHR9	RXB8	TXB8	page 63
\$09 (\$29)	UBRR	UART Baud Rate Register								page 65
\$08 (\$28)	ACSR	ACD	-	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	page 66
\$07 (\$27)	ADMUX	-	-	-	-	-	MUX2	MUX1	MUX0	page 72
\$06 (\$26)	ADCSR	ADEN	ADSC	ADFR	ADIF	ADIE	ADPS2	ADPS1	ADPS0	page 72
\$05 (\$25)	ADCH	-	-	-	-	-	-	ADC9	ADC8	page 73
\$04 (\$24)	ADCL	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0	page 73
\$03 (\$20)	Reserved									
\$02 (\$22)	Reserved									
\$01 (\$21)	Reserved									
\$00 (\$20)	Reserved									