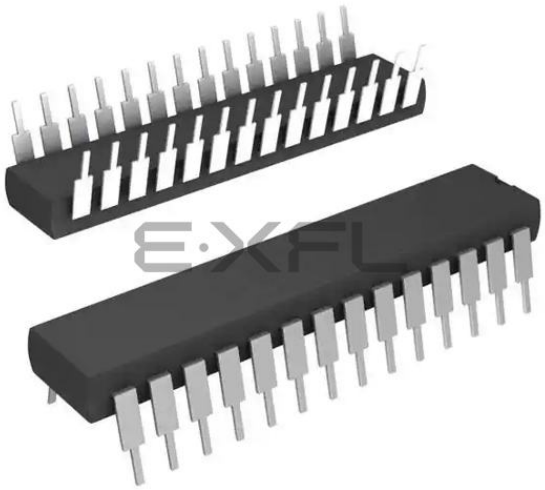


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	AVR
Core Size	8-Bit
Speed	4MHz
Connectivity	SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	20
Program Memory Size	4KB (2K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 6V
Data Converters	A/D 6x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C
Mounting Type	Through Hole
Package / Case	28-DIP (0.300", 7.62mm)
Supplier Device Package	28-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at90ls4433-4pi

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-, Y-, and Z-register.

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 5 shows the AT90S4433 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.

Figure 5. The AT90S4433 AVR RISC Architecture

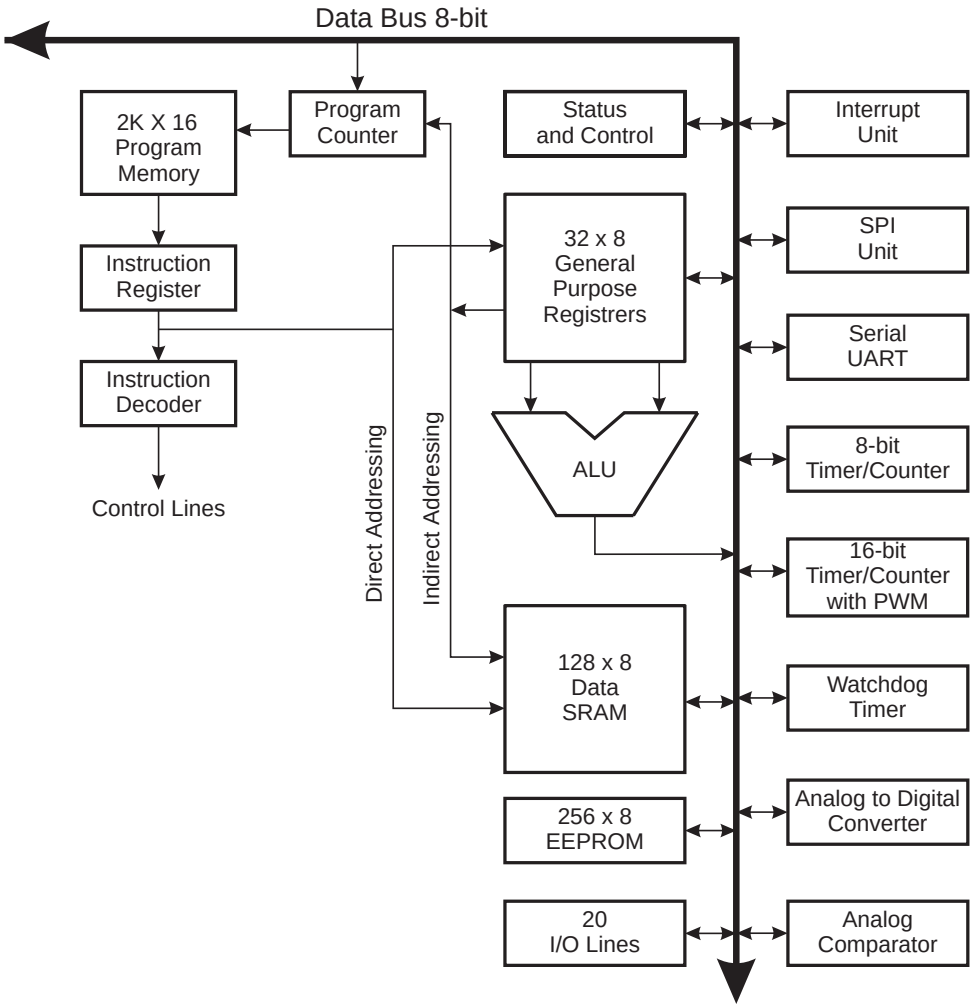


Figure 24. Reset Logic

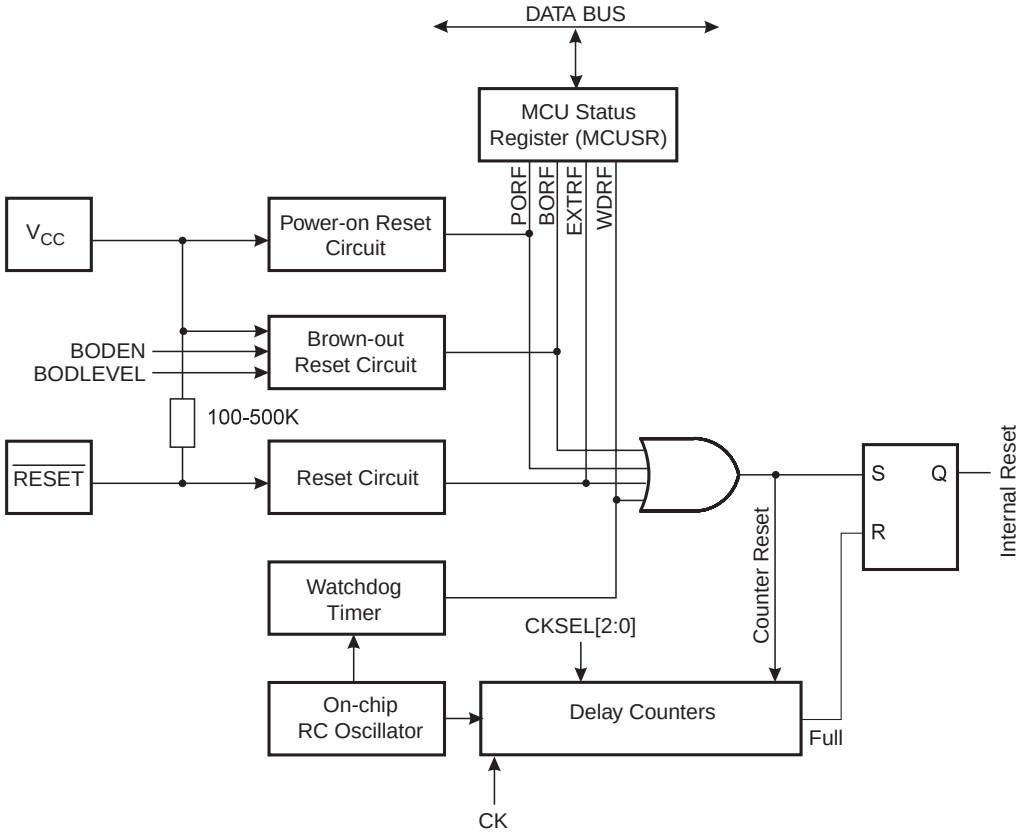


Table 4. Reset Characteristics (V_{CC} = 5.0V)

Symbol	Parameter	Min	Typ	Max	Units
V_{POT}[1]	Power-on Reset Threshold Voltage, rising	1.0	1.4	1.8	V
	Power-on Reset Threshold Voltage, falling	0.4	0.6	0.8	V
V_{RST}	$\overline{\text{RESET}}$ Pin Threshold Voltage		0.6 V_{CC}		V
V_{BOT}	Brown-out Reset Threshold Voltage	2.2 (BODLEVEL=1)	2.7 (BODLEVEL=1)	3.0 (BODLEVEL=1)	V
		3.5 (BODLEVEL=0)	4.0 (BODLEVEL=0)	4.5 (BODLEVEL=0)	

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

the TEMP Register. Consequently, the Low Byte TCNT1L must be accessed first for a full 16-bit register read operation.

The Timer/Counter1 is realized as an up or up/down (in PWM mode) counter with read and write access. If Timer/Counter1 is written to and a clock source is selected, the Timer/Counter1 continues counting in the timer clock cycle after it is preset with the written value.

Timer/Counter1 Output Compare Register – OCR1H and OCR1L

Bit	15	14	13	12	11	10	9	8	
$2B ($4B)	**MSB**								**OCR1H**
$2A ($4A)								**LSB**	**OCR1L**
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	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 Output Compare Register is a 16-bit read/write register.

The Timer/Counter1 Output Compare Register contains the data to be continuously compared with Timer/Counter1. Actions on compare matches are specified in the Timer/Counter1 Control and Status Register.

Since the Output Compare Register (OCR1) is a 16-bit register, a temporary register TEMP is used when OCR1 is written to ensure that both bytes are updated simultaneously. When the CPU writes the High Byte, OCR1H, the data is temporarily stored in the TEMP Register. When the CPU writes the Low Byte, OCR1L, the TEMP Register is simultaneously written to OCR1H. Consequently, the High Byte OCR1H must be written first for a full 16-bit register write operation.

The TEMP Register is also used when accessing TCNT1 and ICR1. If the main program and interrupt routines perform access to registers using TEMP, interrupts must be disabled during access from the main program.

UART

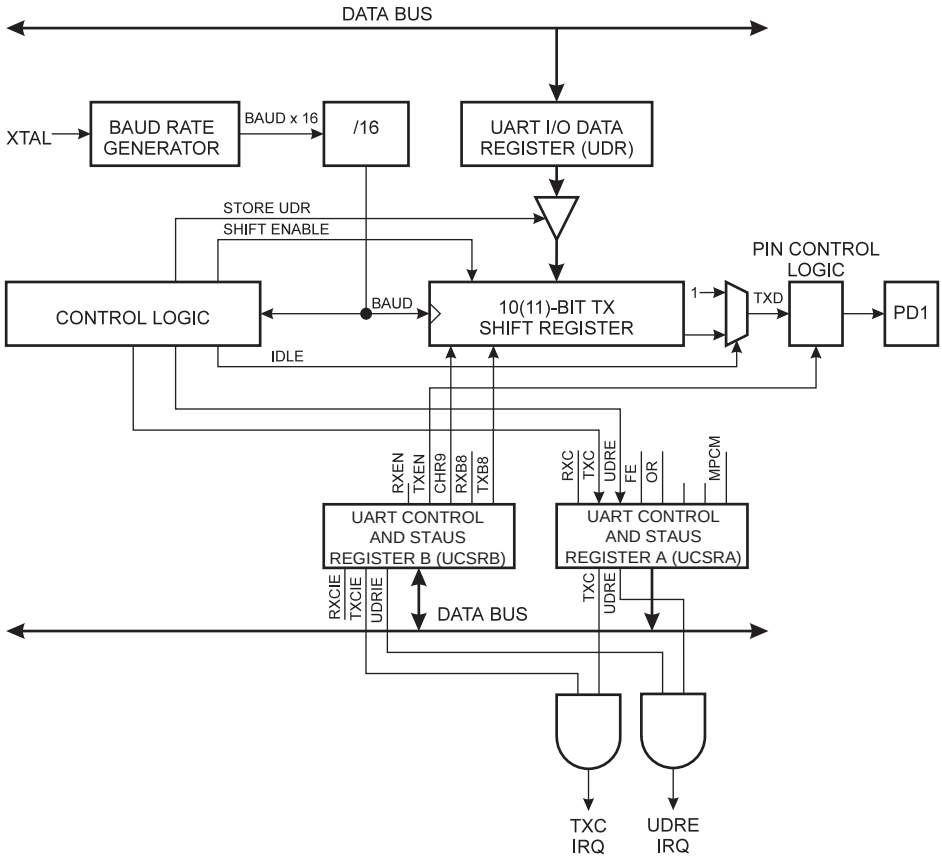
The AT90S4433 features a full duplex (separate Receive and Transmit Registers) Universal Asynchronous Receiver and Transmitter (UART). The main features are:

- **Baud Rate Generator Generates any Baud Rate**
- **High Baud Rates at Low XTAL Frequencies**
- **8 or 9 Bits Data**
- **Noise Filtering**
- **Overrun Detection**
- **Framing Error Detection**
- **False Start Bit Detection**
- **Three Separate Interrupts on TX Complete, TX Data Register Empty, and RX Complete**
- **Multi-processor Communication Mode**

Data Transmission

A block schematic of the UART Transmitter is shown in Figure 40.

Figure 40. UART Transmitter



Data transmission is initiated by writing the data to be transmitted to the UART I/O Data Register (UDR). Data is transferred from UDR to the Transmit Shift Register when:

- A new character has been written to UDR after the stop bit from the previous character has been shifted out. The Shift Register is loaded immediately.
- A new character has been written to UDR before the stop bit from the previous character has been shifted out. The Shift Register is loaded when the stop bit of the character currently being transmitted has been shifted out.

When data is transferred from UDR to the Shift Register, the UDRE (UART Data Register Empty) bit in the UART Control and Status Register A, UCSRA, is set. When this bit is set (one), the UART is ready to receive the next character. At the same time as the

data is transferred from UDR to the 10(11)-bit Shift Register, bit 0 of the Shift Register is cleared (start bit) and bit 9 or 10 is set (stop bit). If 9-bit data word is selected (the CHR9 bit in the UART Control and Status Register B, UCSRB is set), the TXB8 bit in UCSRB is transferred to bit nine in the Transmit Shift Register.

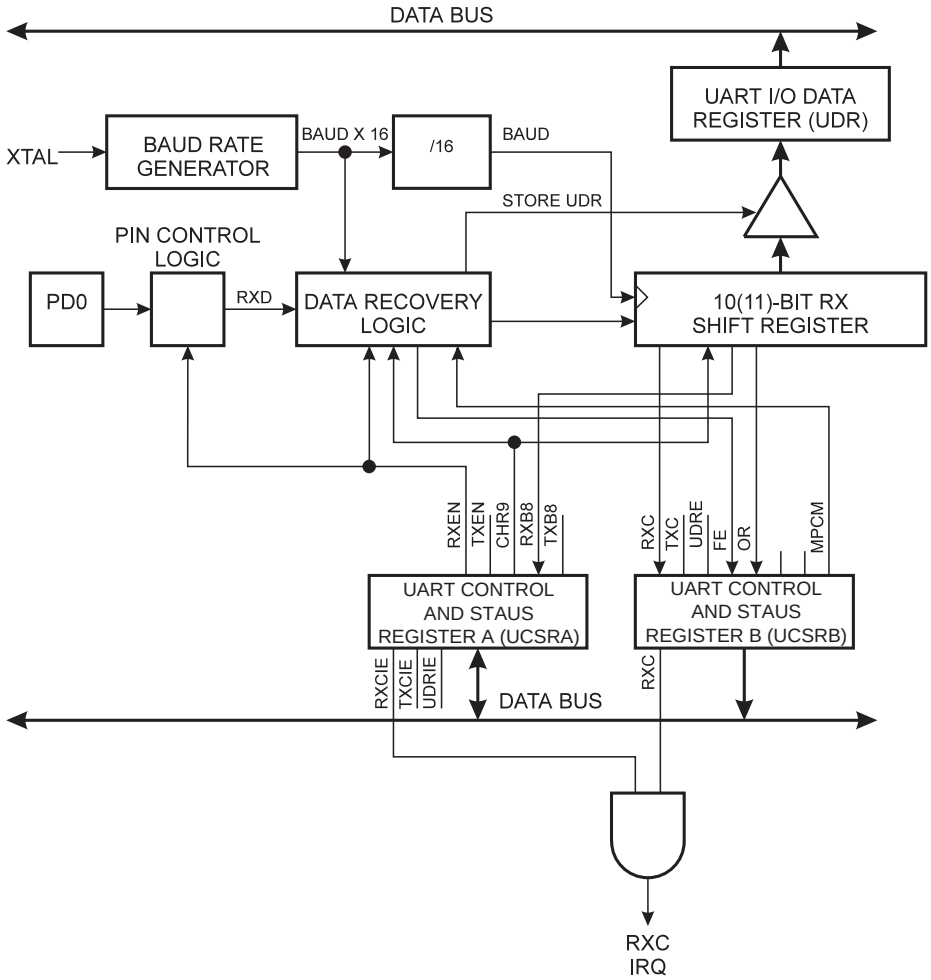
On the baud rate clock following the transfer operation to the Shift Register, the start bit is shifted out on the TXD pin. Then follows the data, LSB first. When the stop bit has been shifted out, the Shift Register is loaded if any new data has been written to the UDR during the transmission. During loading, UDRE is set. If there is no new data in the UDR Register to send when the stop bit is shifted out, the UDRE Flag will remain set until UDR is written again. When no new data has been written, and the stop bit has been present on TXD for one bit length, the TX Complete Flag, TXC, in UCSRA is set.

The TXEN bit in UCSRB enables the UART Transmitter when set (one). When this bit is cleared (zero), the PD1 pin can be used for general I/O. When TXEN is set, the UART Transmitter will be connected to PD1, which is forced to be an output pin regardless of the setting of the DDD1 bit in DDRD.

Data Reception

Figure 41 shows a block diagram of the UART Receiver.

Figure 41. UART Receiver



UART Control

UART I/O Data Register – UDR

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

The UDR Register is actually two physically separate registers sharing the same I/O address. When writing to the register, the UART Transmit Data Register is written. When reading from UDR, the UART Receive Data Register is read.

UART Control and Status Register A – UCSRA

Bit	7	6	5	4	3	2	1	0	
$0B ($2B)	RXC	TXC	UDRE	FE	OR	–	–	MPCM	UCSRA
Read/Write	R	R/W	R	R	R	R	R	R/W	
Initial Value	0	0	1	0	0	0	0	0	

- **Bit 7 – RXC: UART Receive Complete**

This bit is set (one) when a received character is transferred from the Receiver Shift Register to UDR. The bit is set regardless of any detected framing errors. When the RXCIE bit in UCSRB is set, the UART Receive Complete interrupt will be executed when RXC is set (one). RXC is cleared by reading UDR. When interrupt-driven data reception is used, the UART Receive Complete Interrupt routine must read UDR in order to clear RXC, otherwise a new interrupt will occur once the interrupt routine terminates.

- **Bit 6 – TXC: UART Transmit Complete**

This bit is set (one) when the entire character (including the stop bit) in the Transmit Shift Register has been shifted out and no new data has been written to UDR. This flag is especially useful in half-duplex communications interfaces, where a transmitting application must enter Receive mode and free the communications bus immediately after completing the transmission.

When the TXCIE bit in UCSRB is set, setting of TXC causes the UART Transmit Complete interrupt to be executed. TXC is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, the TXC bit is cleared (zero) by writing a logical “1” to the bit.

- **Bit 5 – UDRE: UART Data Register Empty**

This bit is set (one) when a character written to UDR is transferred to the Transmit Shift Register. Setting of this bit indicates that the Transmitter is ready to receive a new character for transmission.

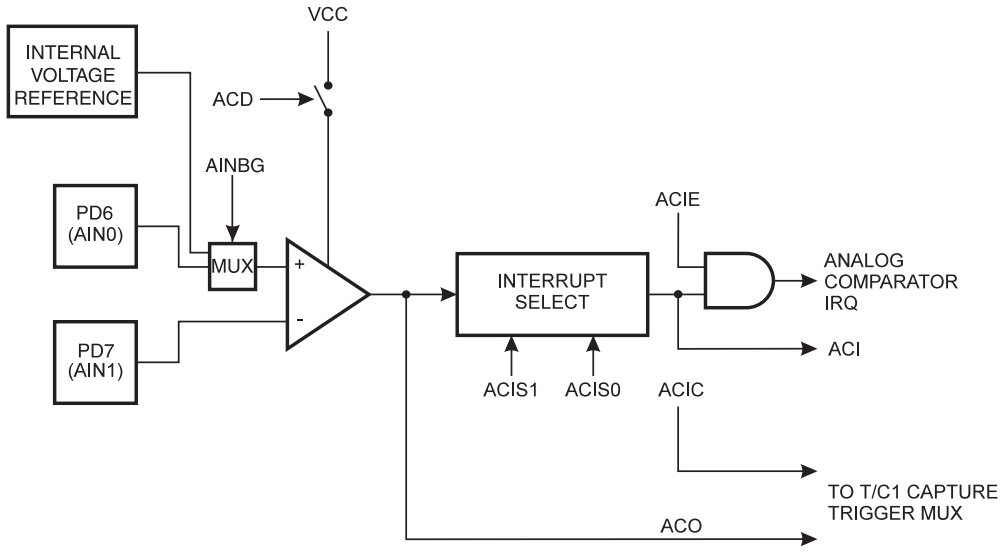
When the UDRIE bit in UCSRB is set, the UART Transmit Complete interrupt to be executed as long as UDRE is set. UDRE is cleared by writing UDR. When interrupt-driven data transmittal is used, the UART Data Register Empty Interrupt routine must write UDR in order to clear UDRE, otherwise a new interrupt will occur once the interrupt routine terminates.

UDRE is set (one) during reset to indicate that the transmitter is ready.

Analog Comparator

The Analog Comparator compares the input values on the positive input PD6 (AIN0) and negative input PD7 (AIN1). When the voltage on the positive input PD6 (AIN0) is higher than the voltage on the negative input PD7 (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 43.

Figure 43. Analog Comparator Block Diagram



Analog Comparator Control and Status Register – ACSR

Bit	7	6	5	4	3	2	1	0	
$08 ($28)	**ACD**	**AINBG**	**ACO**	**ACI**	**ACIE**	**ACIC**	**ACIS1**	**ACIS0**	**ACSR**
Read/Write	R/W	R/W	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 – AINBG: Analog Comparator Bandgap Select**

When this bit is set, BOD is enabled and the BODEN is programmed, a fixed bandgap voltage of 1.22V ± 0.1V replaces the normal input to the positive input (AIN0) of the comparator. When this bit is cleared, the normal input pin, PD6, is applied to the positive input of the comparator.

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

ACO is directly connected to the comparator output.

Analog-to-Digital Converter

Features

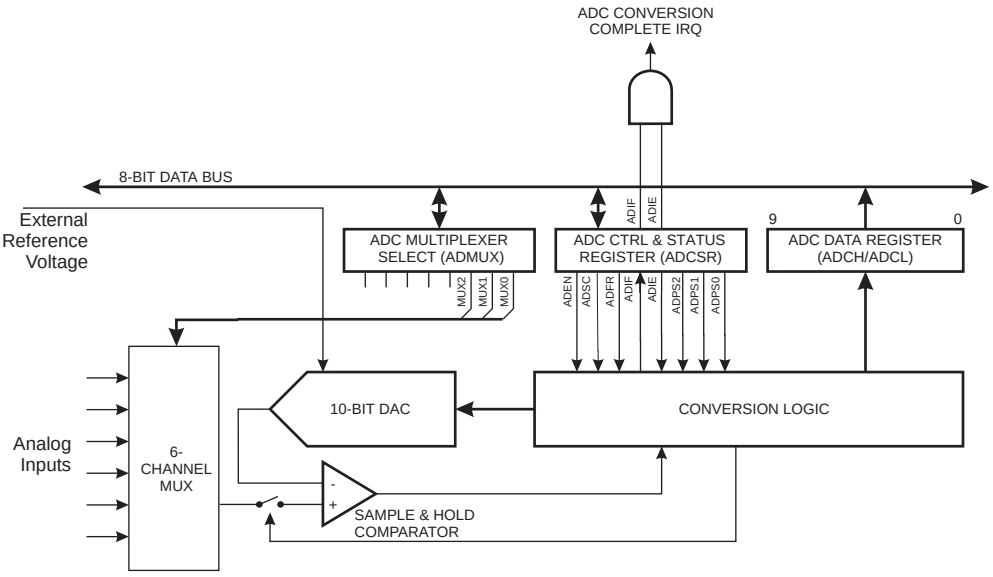
- **10-bit Resolution**
- **±2 LSB Absolute Accuracy**
- **0.5 LSB Integral Non-linearity**
- **65 - 260 µs Conversion Time**
- **Up to 15 kSPS**
- **Six Multiplexed Input Channels**
- **Rail-to-Rail Input Range**
- **Free Run or Single Conversion Mode**
- **Interrupt on ADC Conversion Complete**
- **Sleep Mode Noise Canceler**

The AT90S4433 features a 10-bit successive approximation ADC. The ADC is connected to a 6-channel Analog Multiplexer, which allows each pin of Port C to be used as an input for the ADC. The ADC contains a Sample and Hold Amplifier, which ensures that the input voltage to the ADC is held at a constant level during conversion. A block diagram of the ADC is shown in Figure 44.

The ADC has two separate analog supply voltage pins: AVCC and AGND. AGND must be connected to GND, and the voltage on AVCC must not differ from V_{CC} more than ±0.3V. See the section “ADC Noise Canceling Techniques” on page 70 for how to connect these pins.

An external reference voltage must be applied to the AREF pin. This voltage must be in the range 2.0 - AVCC.

Figure 44. Analog-to-Digital Converter Block Schematic



keeps running for as long as the ADEN bit is set and is continuously reset when ADEN is low.

When initiating a conversion by setting the ADSC bit in ADCSR, the conversion starts at the following rising edge of the ADC clock cycle. The actual sample-and-hold takes place 1.5 ADC clock cycles after the start of the conversion. The result is ready and written to the ADC Result Register after 13 cycles. In Single Conversion mode, the ADC needs one more clock cycle before a new conversion can be started (see Figure 47). If ADSC is set high in this period, the ADC will start the new conversion immediately. In Free Run mode, a new conversion will be started immediately after the result is written to the ADC Result Register. Using Free Run mode and an ADC clock frequency of 200 kHz gives the lowest conversion time, 65 µs, equivalent to 15.4 kSPS. For a summary of conversion times, see Table 21.

Figure 46. ADC Timing Diagram, First Conversion (Single Conversion Mode)

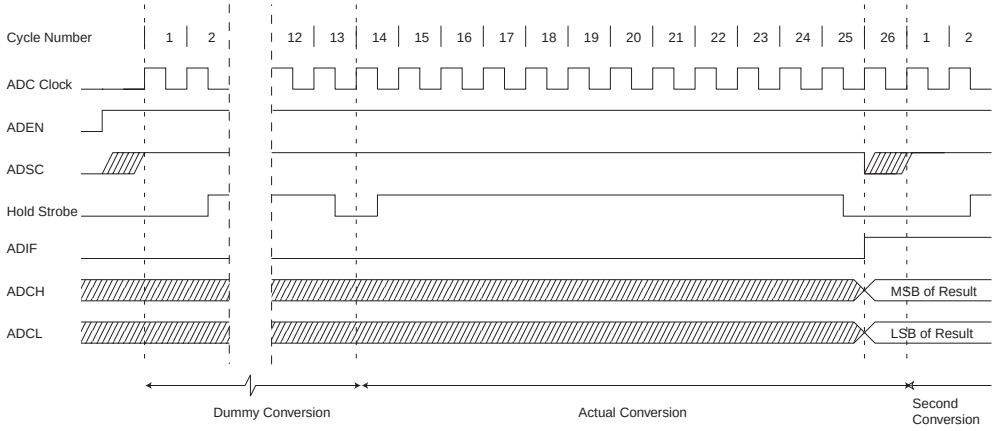


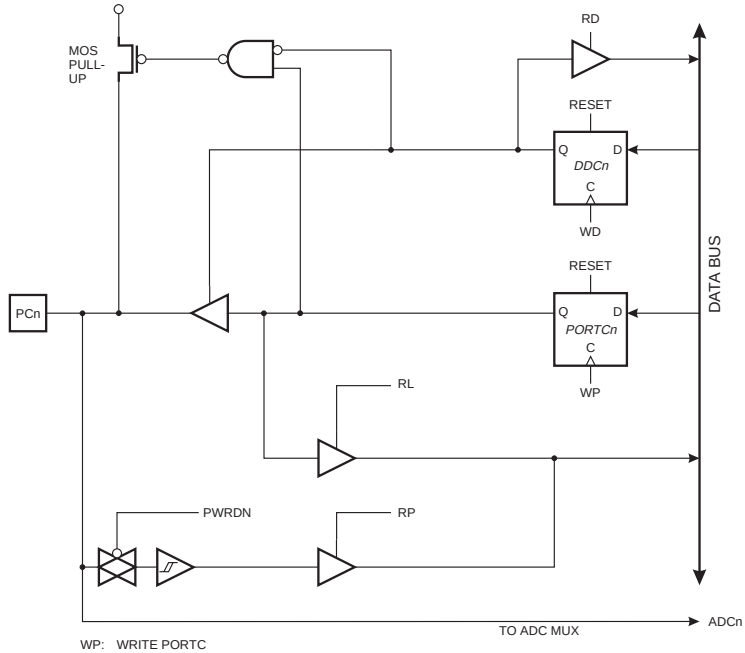
Table 21. ADC Conversion Time

Condition	Sample Cycle Number	Result Ready(Cycle Number)	Total Conversion Time (Cycles)	Total Conversion Time (µs)
1st Conversion, Free Run	13.5	25	25	125 - 500
1st Conversion, Single	13.5	25	26	130 - 520
Free Run Conversion	1.5	13	13	65 - 260
Single Conversion	1.5	13	14	70 - 280

Port C Schematics

Note that all port pins are synchronized. The synchronization latch is, however, not shown in the figure.

Figure 56. Port C Schematic Diagrams (Pins PC0 - PC5)



Port D as General Digital I/O

PDn, General I/O pin: The DDDn bit in the DDRD Register selects the direction of this pin. If DDDn is set (one), PDn is configured as an output pin. If DDDn is cleared (zero), PDn is configured as an input pin. If PDn is set (one) when configured as an input pin, the MOS pull-up resistor is activated. To switch the pull-up resistor off, the PDn 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.

Table 27. DDDn Bits on Port D Pins(1)

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

Note: 1. n: 7,6..0, pin number.

Alternate Functions of Port D

- **AIN1 – Port D, Bit 7**

AIN1, Analog Comparator Negative Input. When configured as an input (DDD7 is cleared [zero]), and with the internal MOS pull-up resistor switched off (PD7 is cleared [zero]), this pin also serves as the negative input of the On-chip Analog Comparator. During Power-down mode, the Schmitt trigger of the digital input is disconnected. This allows analog signals, which are close to $V_{CC}/2$, to be present during Power-down without causing excessive power consumption.

- **AIN0 – Port D, Bit 6**

AIN0, Analog Comparator Positive Input. When configured as an input (DDD6 is cleared [zero]), and with the internal MOS pull-up resistor switched off (PD6 is cleared [zero]), this pin also serves as the positive input of the On-chip Analog Comparator. During Power-down mode, the Schmitt trigger of the digital input is disconnected. This allows analog signals, which are close to $V_{CC}/2$, to be present during Power-down without causing excessive power consumption.

- **T1 – Port D, Bit 5**

T1, Timer/Counter1 Counter Source. See the Timer description for further details

- **T0 – Port D, Bit 4**

T0: Timer/Counter0 Counter Source. See the Timer description for further details.

- **INT1 – Port D, Bit 3**

INT1, External Interrupt Source 1: The PD3 pin can serve as an external interrupt source to the MCU. See the interrupt description for further details and how to enable the source.

- **INT0 – Port D, Bit 2**

INT0, External Interrupt Source 0: The PD2 pin can serve as an external interrupt source to the MCU. See the interrupt description for further details and how to enable the source.

Figure 58. Port D Schematic Diagram (Pin PD1)

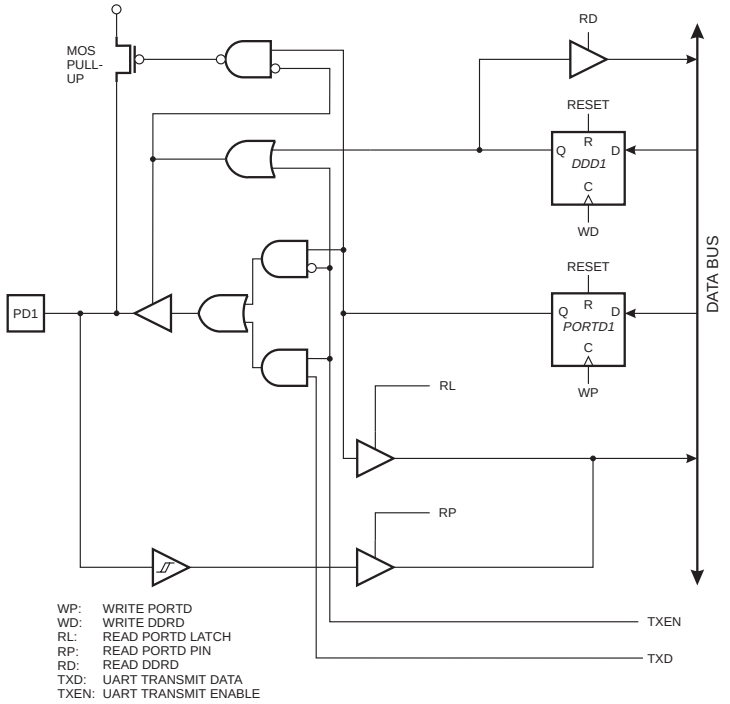
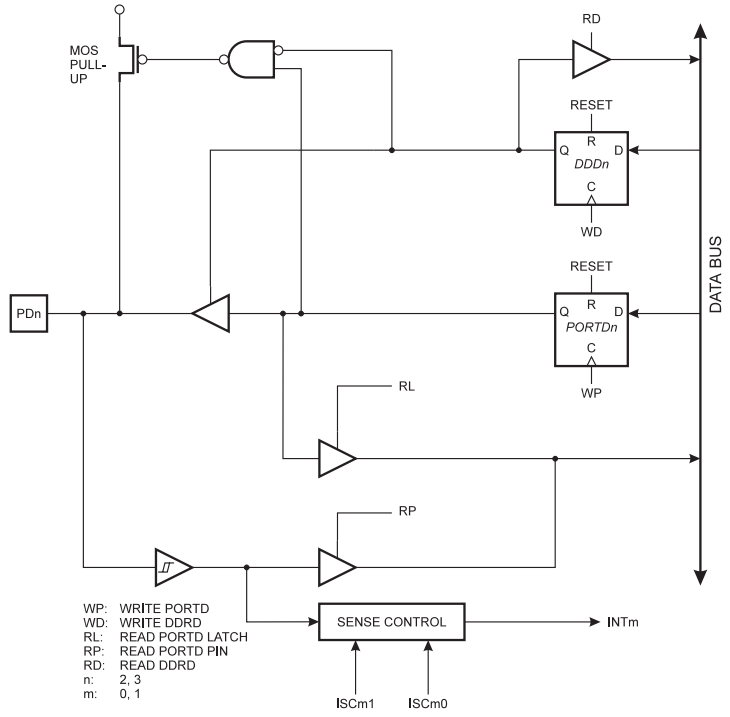


Figure 59. Port D Schematic Diagram (Pins PD2 and PD3)



G: Write Data High Byte

1. Set BS to “1”. This selects high data.
2. Give $\overline{WR}$ a negative pulse. This starts programming of the data byte. RDY/$\overline{BSY}$ goes low.
3. Wait until RDY/$\overline{BSY}$ goes high to program the next byte.

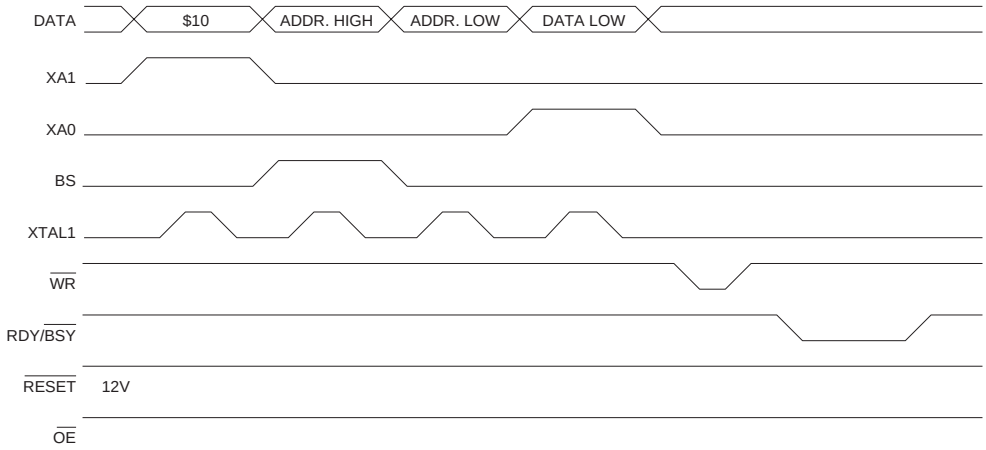
(See Figure 64 for signal waveforms.)

The loaded command and address are retained in the device during programming. For efficient programming, the following should be considered:

- The command needs to be loaded only once when writing or reading multiple memory locations.
- Address High Byte needs to be loaded only before programming a new 256-word page in the Flash.
- Skip writing the data value $FF, that is, the contents of the entire Flash and EEPROM after a Chip Erase.

These considerations also apply to EEPROM programming and Flash, EEPROM and signature bytes reading.

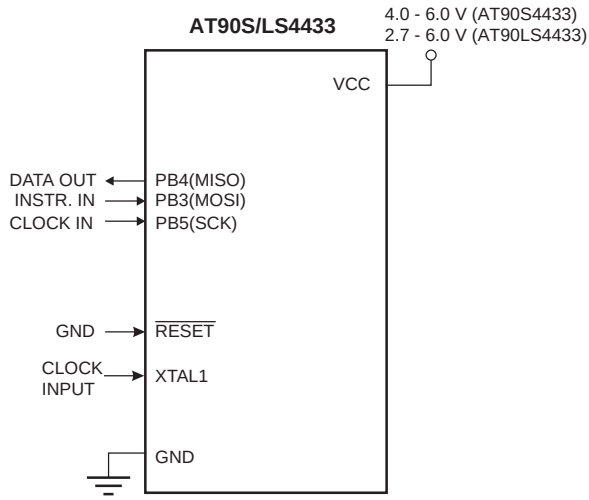
Figure 63. Programming the Flash Waveforms



Serial Downloading

Both the Program and Data memory arrays can be programmed using the SPI bus while $\overline{\text{RESET}}$ is pulled to GND. The serial interface consists of pins SCK, MOSI (input) and MISO (output) (see Figure 66). After $\overline{\text{RESET}}$ is set low, the Programming Enable instruction needs to be executed first before program/erase instructions can be executed.

Figure 66. 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 $07FF for Program memory and $0000 to $00FF for EEPROM memory.

Either an external system 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 AT90S4433, data is clocked on the rising edge of CLK.

When reading data from the AT90S4433, data is clocked on the falling edge of CLK. See Figure 67, Figure 68 and Table 36 for details.

To program and verify the AT90S4433 in the Serial Programming mode, the following sequence is recommended (see 4-byte instruction formats in Table 35):

1. Power-up sequence:
   Apply power between V_{CC} and GND while $\overline{\text{RESET}}$ 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, $\overline{\text{RESET}}$ 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 pin MOSI/PB3.
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 issu-

Table 35. Serial Programming Instruction Set

Instruction	Instruction Format				Operation
	Byte 1	Byte 2	Byte 3	Byte4	
Programming Enable	1010 1100	0101 0011	xxxx xxxx	xxxx xxxx	Enable Serial Programming while $\overline{\text{RESET}}$ is low.
Chip Erase	1010 1100	100x xxxx	xxxx xxxx	xxxx xxxx	Chip Erase Flash and EEPROM memory arrays.
Read Program Memory	0010 **H**000	xxxx x**aaa**	**bbbb bbbb**	**oooo oooo**	Read **H** (high or low) data **o** from program memory at word address **a**:**b**.
Write Program Memory	0100 **H**000	xxxx x**aaa**	**bbbb bbbb**	**iiii iiii**	Write **H** (high or low) data **i** to program memory at word address **a**:**b**.
Read EEPROM Memory	1010 0000	xxxx xxxx	**bbbb bbbb**	**oooo oooo**	Read data **o** from EEPROM memory at address **a**:**b**.
Write EEPROM Memory	1100 0000	xxxx xxxx	**bbbb bbbb**	**iiii iiii**	Write data **i** to EEPROM memory at address **a**:**b**.
Write Lock Bits	1010 1100	1111 1***21***1	xxxx xxxx	xxxx xxxx	Write Lock bits. Set bits ***1,2***= “0” to program Lock bits.
Read Lock Bits	0101 1000	xxxx xxxx	xxxx xxxx	xxxx x***21***x	Read Lock bits. “0” = programmed, “1” = unprogrammed.
Read Sigature Bytes	0011 0000	xxxx xxxx	xxxx xx**bb**	**oooo oooo**	Read signature byte **o** at address **b**.(1)
Write Fuse Bits	1010 1100	101**7 6543**	xxxx xxxx	xxxx xxxx	Set bits **7** - **3** = “0” to program, “1” to unprogram.
Read Fuse Bits	0101 0000	xxxx xxxx	xxxx xxxx	xx**87 6543**	Read Fuse bits. “0” = programmed, “1” = unprogrammed.

Note: 1. The signature bytes are not readable in lock mode 3, i.e., both Lock bits programmed.

a = address high bits
b = address low bits
H = 0 – Low Byte, 1 – High Byte
o = data out
i = data in
x = don’t care
1 = Lock bit 1
2 = Lock bit 2
3 = CKSEL0 Fuse
4 = CKSEL1 Fuse
5 = CKSEL2 Fuse
6 = BODEN Fuse
7 = BODLEVEL Fuse
8 = SPIEN Fuse

Figure 71. Active Supply Current vs. V_{CC}

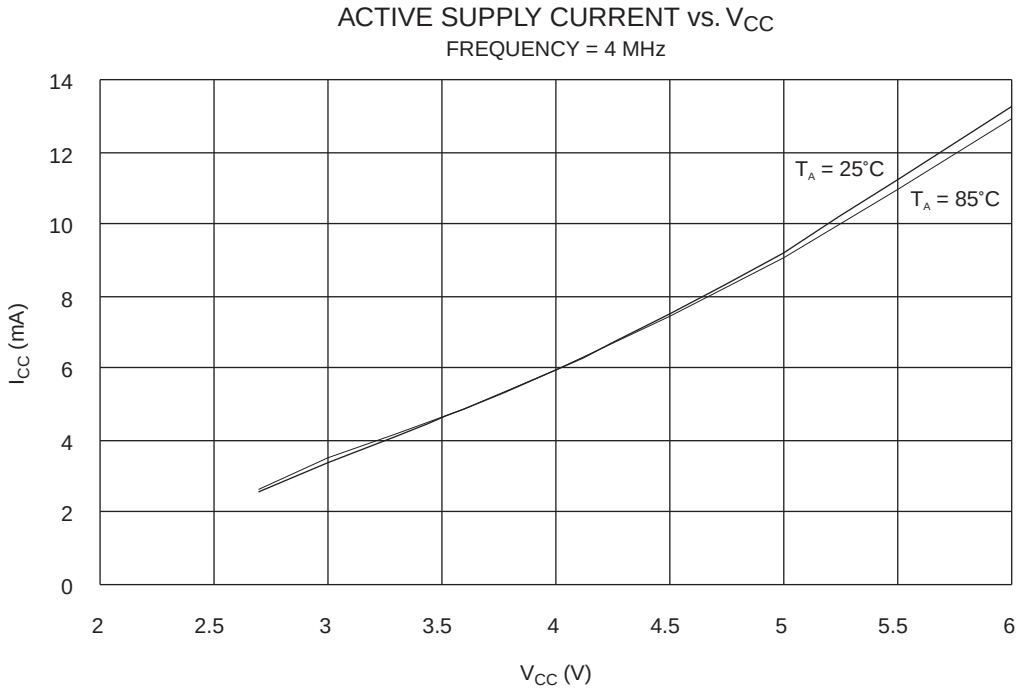


Figure 72. Idle Supply Current vs. Frequency

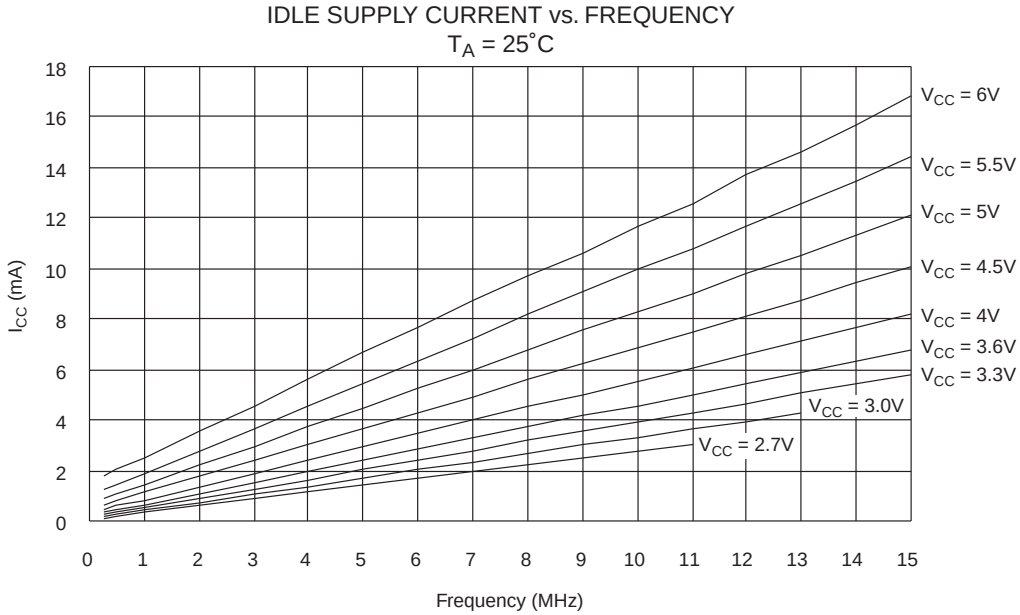


Figure 75. Power-down Supply Current vs. V_{CC}

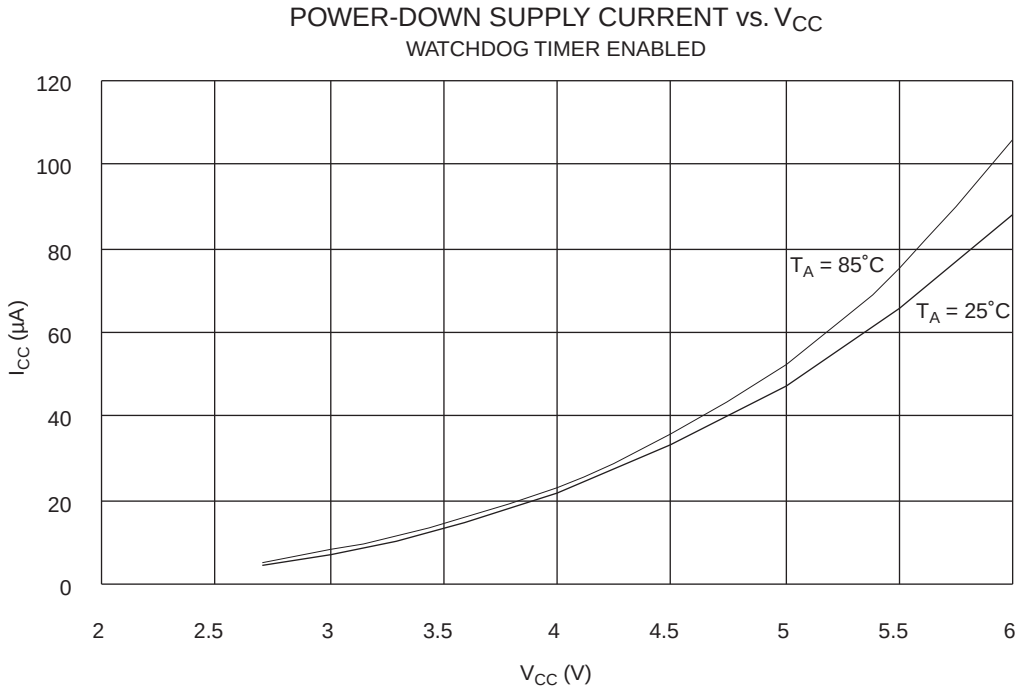


Figure 76. Power-down Supply Current vs. V_{CC}

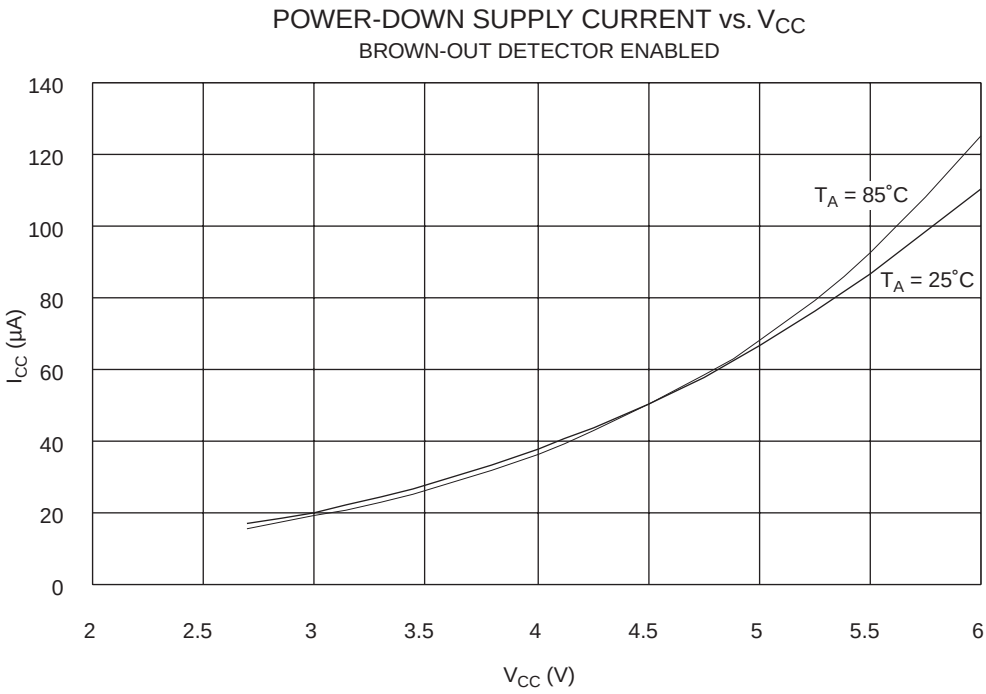


Figure 85. I/O Pin Source Current vs. Output Voltage

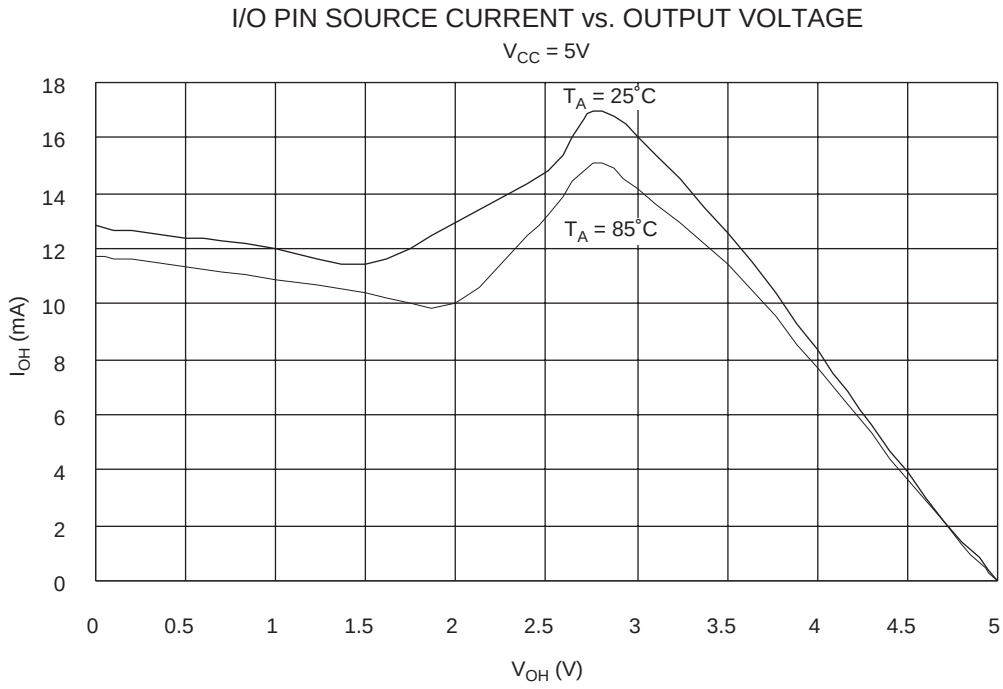
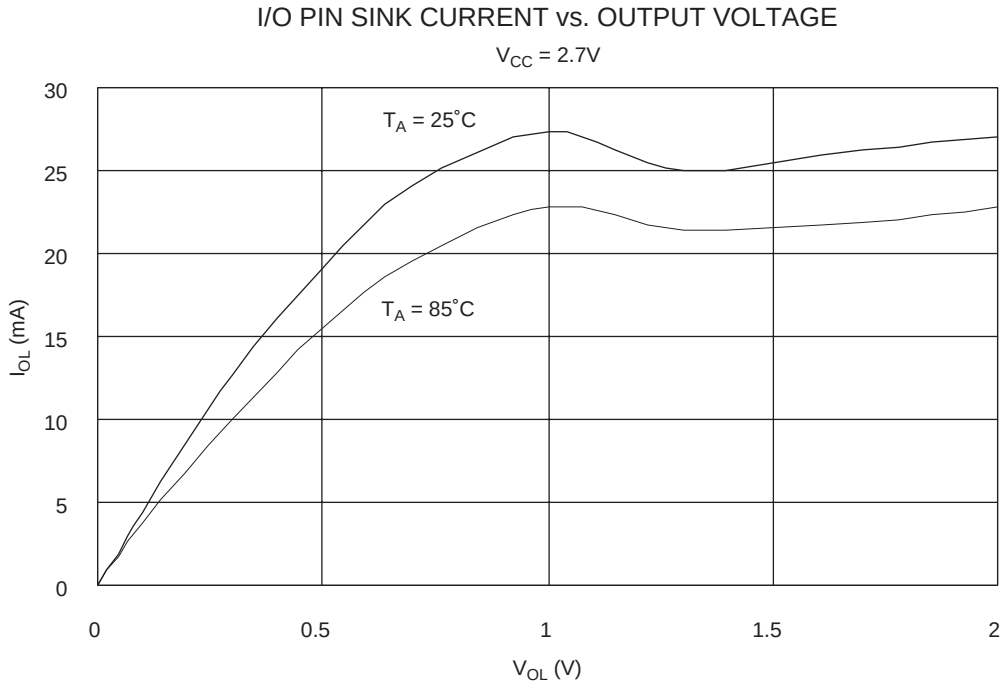


Figure 86. I/O Pin Sink Current vs. Output Voltage



Instruction Set Summary (Continued)

Mnemonic	Operands	Description	Operation	Flags	# Clocks
LD	Rd, Y	Load Indirect	Rd ← (Y)	None	2
LD	Rd, Y+	Load Indirect and Post-inc.	Rd ← (Y), Y ← Y + 1	None	2
LD	Rd, -Y	Load Indirect and Pre-dec.	Y ← Y - 1, Rd ← (Y)	None	2
LDD	Rd,Y+q	Load Indirect with Displacement	Rd ← (Y + q)	None	2
LD	Rd, Z	Load Indirect	Rd ← (Z)	None	2
LD	Rd, Z+	Load Indirect and Post-inc.	Rd ← (Z), Z ← Z + 1	None	2
LD	Rd, -Z	Load Indirect and Pre-dec.	Z ← Z - 1, Rd ← (Z)	None	2
LDD	Rd, Z+q	Load Indirect with Displacement	Rd ← (Z + q)	None	2
LDS	Rd, k	Load Direct from SRAM	Rd ← (k)	None	2
ST	X, Rr	Store Indirect	(X) ← Rr	None	2
ST	X+, Rr	Store Indirect and Post-inc.	(X) ← Rr, X ← X + 1	None	2
ST	-X, Rr	Store Indirect and Pre-dec.	X ← X - 1, (X) ← Rr	None	2
ST	Y, Rr	Store Indirect	(Y) ← Rr	None	2
ST	Y+, Rr	Store Indirect and Post-inc.	(Y) ← Rr, Y ← Y + 1	None	2
ST	-Y, Rr	Store Indirect and Pre-dec.	Y ← Y - 1, (Y) ← Rr	None	2
STD	Y+q, Rr	Store Indirect with Displacement	(Y + q) ← Rr	None	2
ST	Z, Rr	Store Indirect	(Z) ← Rr	None	2
ST	Z+, Rr	Store Indirect and Post-inc.	(Z) ← Rr, Z ← Z + 1	None	2
ST	-Z, Rr	Store Indirect and Pre-dec.	Z ← Z - 1, (Z) ← Rr	None	2
STD	Z+q, Rr	Store Indirect with Displacement	(Z + q) ← Rr	None	2
STS	k, Rr	Store Direct to SRAM	(k) ← Rr	None	2
LPM		Load Program Memory	R0 ← (Z)	None	3
IN	Rd, P	In Port	Rd ← P	None	1
OUT	P, Rr	Out Port	P ← Rr	None	1
PUSH	Rr	Push Register on Stack	STACK ← Rr	None	2
POP	Rd	Pop Register from Stack	Rd ← STACK	None	2
BIT AND BIT-TEST INSTRUCTIONS					
SBI	P, b	Set Bit in I/O Register	I/O(P,b) ← 1	None	2
CBI	P, b	Clear Bit in I/O Register	I/O(P,b) ← 0	None	2
LSL	Rd	Logical Shift Left	Rd(n+1) ← Rd(n), Rd(0) ← 0	Z,C,N,V	1
LSR	Rd	Logical Shift Right	Rd(n) ← Rd(n+1), Rd(7) ← 0	Z,C,N,V	1
ROL	Rd	Rotate Left through Carry	Rd(0) ← C, Rd(n+1) ← Rd(n), C ← Rd(7)	Z,C,N,V	1
ROR	Rd	Rotate Right through Carry	Rd(7) ← C, Rd(n) ← Rd(n+1), C ← Rd(0)	Z,C,N,V	1
ASR	Rd	Arithmetic Shift Right	Rd(n) ← Rd(n+1), n = 0..6	Z,C,N,V	1
SWAP	Rd	Swap Nibbles	Rd(3..0) ← Rd(7..4), Rd(7..4) ← Rd(3..0)	None	1
BSET	s	Flag Set	SREG(s) ← 1	SREG(s)	1
BCLR	s	Flag Clear	SREG(s) ← 0	SREG(s)	1
BST	Rr, b	Bit Store from Register to T	T ← Rr(b)	T	1
BLD	Rd, b	Bit Load from T to Register	Rd(b) ← T	None	1
SEC		Set Carry	C ← 1	C	1
CLC		Clear Carry	C ← 0	C	1
SEN		Set Negative Flag	N ← 1	N	1
CLN		Clear Negative Flag	N ← 0	N	1
SEZ		Set Zero Flag	Z ← 1	Z	1
CLZ		Clear Zero Flag	Z ← 0	Z	1
SEI		Global Interrupt Enable	I ← 1	I	1
CLI		Global Interrupt Disable	I ← 0	I	1
SES		Set Signed Test Flag	S ← 1	S	1
CLS		Clear Signed Test Flag	S ← 0	S	1
SEV		Set Two's Complement Overflow	V ← 1	V	1
CLV		Clear Two's Complement Overflow	V ← 0	V	1
SET		Set T in SREG	T ← 1	T	1
CLT		Clear T in SREG	T ← 0	T	1
SEH		Set Half-carry Flag in SREG	H ← 1	H	1
CLH		Clear Half-carry Flag in SREG	H ← 0	H	1
NOP		No Operation		None	1
SLEEP		Sleep	(see specific descr. for Sleep function)	None	1
WDR		Watchdog Reset	(see specific descr. for WDR/timer)	None	1

3. Brown-out Detection Level

The Brown-out Detection level can increase when there is heavy I/O-activity on the device. The increase can be significant when some of the I/O pins are driving heavy loads.

Problem Fix/Workaround

Select a V_{CC} well above the Brown-out Detection level.

Avoid loading I/O ports with high capacitive or resistive loads.

2. Serial Programming at Voltages below 2.9V

At voltages below 2.9V, serial programming might fail.

Problem Fix/Workaround

Keep V_{CC} at 2.9V or higher during In-System Programming.

1. UART Loses Synchronization if RXD Line is Low when UART Receive is Disabled

The UART will detect a UART start bit and start reception even if the UART is not enabled. If this occurs, the first byte after reenabling the UART will be corrupted.

Problem Fix/Workaround

Make sure that the RX line is high at start-up and when the UART is disabled. An external RS-232 level converter keeps the line high during start-up.