Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

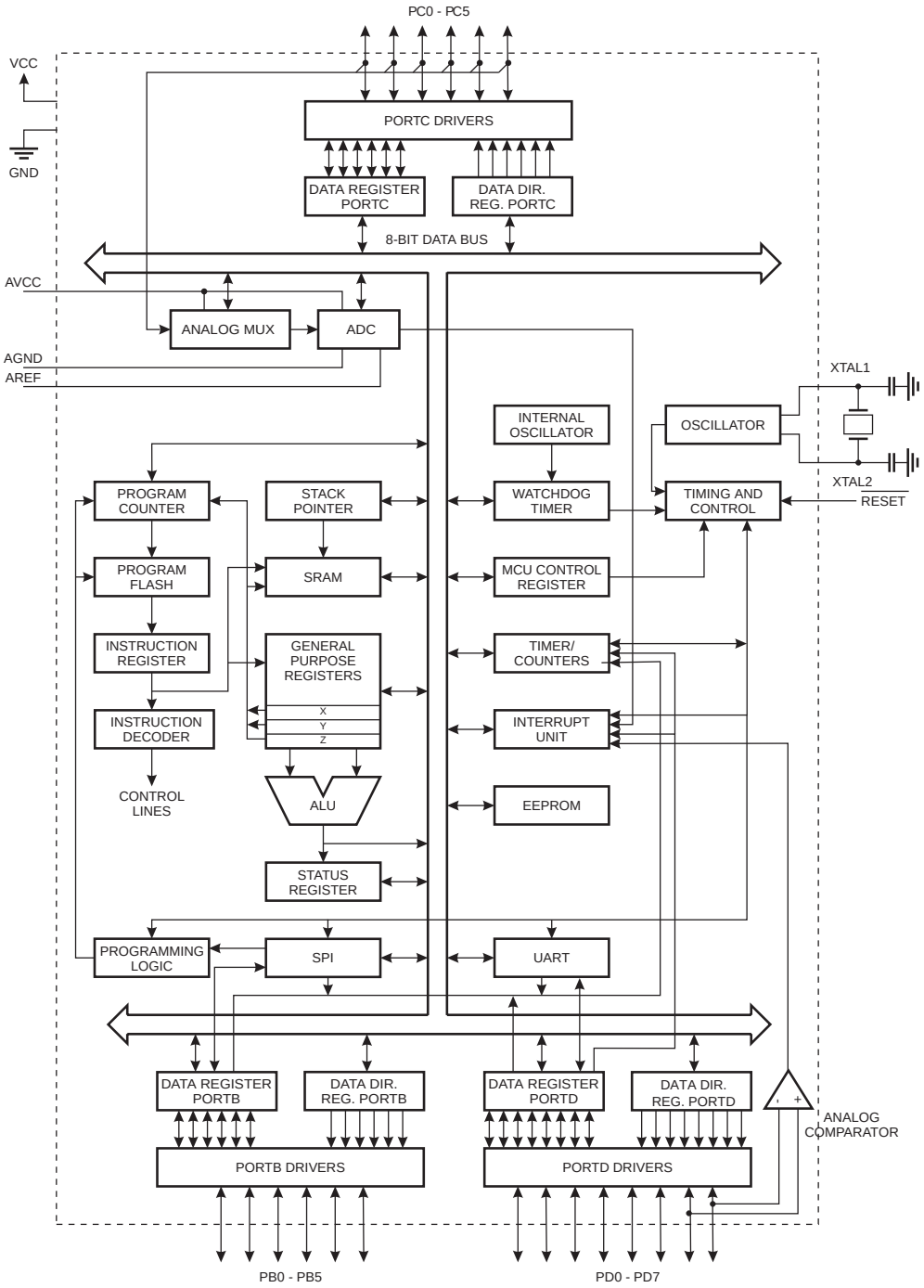
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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	8MHz
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)	4V ~ 6V
Data Converters	A/D 6x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C
Mounting Type	Surface Mount
Package / Case	32-TQFP
Supplier Device Package	32-TQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at90s4433-8ai

Block Diagram

Figure 1. The AT90S4433 Block Diagram



The I/O memory space contains 64 addresses for CPU peripheral functions such 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 Programmable Flash memory.

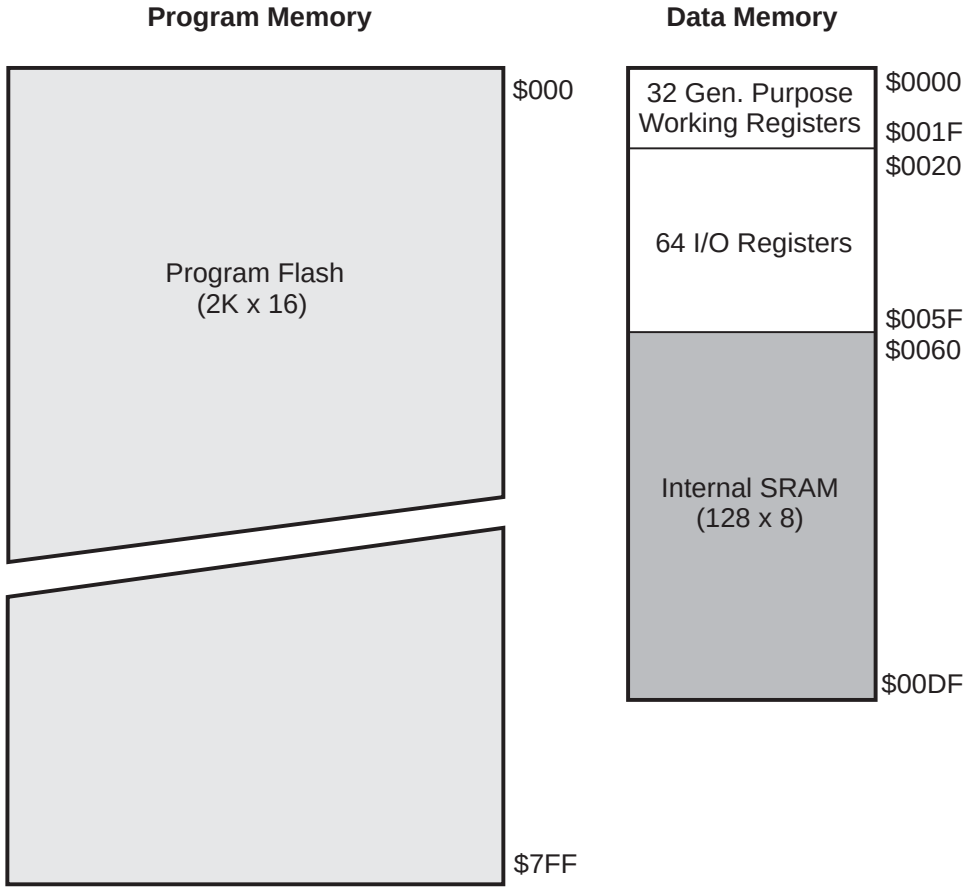
With the relative jump and call instructions, the whole 2K word address space is directly accessed. Most AVR instructions have a single 16-bit word format. Every program memory address contains a 16- or 32-bit instruction.

During interrupts and subroutine calls, the return address Program Counter (PC) is stored on the Stack. The Stack is effectively allocated in the general data SRAM and, consequently, the Stack size is only limited by the total SRAM size and the usage of the SRAM. All user programs must initialize the SP in the reset routine (before subroutines or interrupts are executed). The 8-bit Stack Pointer (SP) is read/write accessible in the I/O space.

The 128 bytes of 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 6. AT90S4433 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.

In the different addressing modes, these address registers have functions as fixed displacement, automatic increment and decrement (see the descriptions for the different instructions).

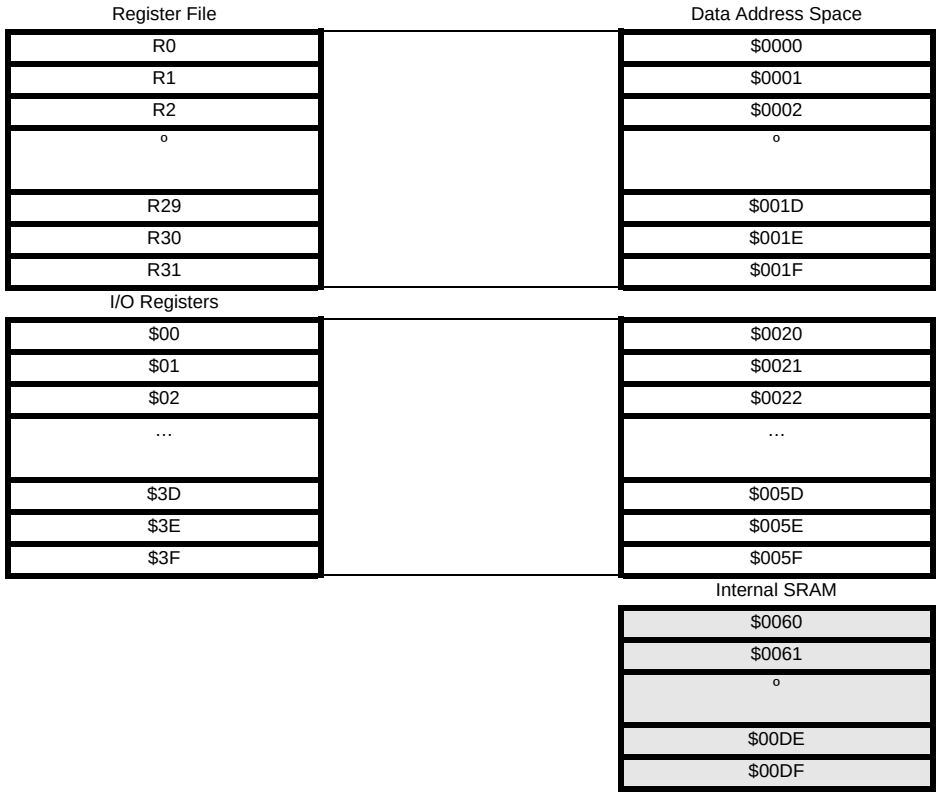
ALU – Arithmetic Logic Unit

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 AT90S4433 contains 4K bytes of On-chip, In-System Programmable Flash memory for program storage. Since all instructions are 16- or 32-bit words, the Flash is organized as 2K x 16. The Flash memory has an endurance of at least 1,000 write/erase cycles. The AT90S4433 Program Counter (PC) is 11 bits wide, thus addressing the 2,048 program memory addresses. See page 93 for a detailed description of Flash data downloading. See page 12 for the different program memory addressing modes.

Figure 9. SRAM Organization



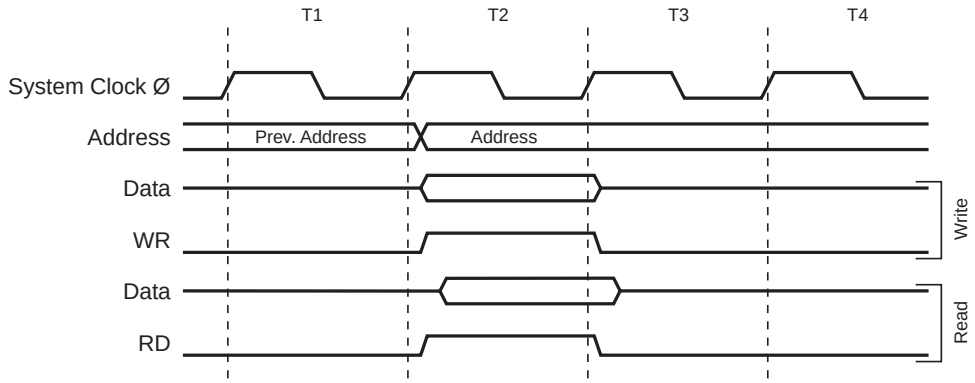
SRAM Data Memory

Figure 9 shows how the AT90S4433 SRAM memory is organized.

The lower 224 data memory locations address the Register File, the I/O memory and the internal data SRAM. The first 96 locations address the Register File and I/O memory, and the next 128 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.

Figure 23. On-chip Data SRAM Access Cycles



I/O Memory

The I/O space definition of the AT90S4433 is shown in Table 2.

Table 2. AT90S4433 I/O Space[1]

I/O Address (SRAM Address)	Name	Function
$3F ($5F)	SREG	Status Register
$3D ($5D)	SP	Stack Pointer
$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 ($54)	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)	OCR1H	Timer/Counter1 Output Compare Register High Byte
$2A ($4A)	OCR1L	Timer/Counter1 Output Compare Register Low Byte
$27 ($47)	ICR1H	Timer/Counter1 Input Capture Register High Byte
$26 ($46)	ICR1L	Timer/Counter 1 Input Capture Register Low Byte
$21 ($41)	WDTCR	Watchdog Timer Control Register
$1E ($3E)	EEAR	EEPROM Address Register
$1D ($3D)	EEDR	EEPROM Data Register
$1C ($3C)	EECR	EEPROM Control Register
$18 ($38)	PORTB	Data Register, Port B

Stack Pointer – SP

The AT90S4433 Stack Pointer is implemented as an 8-bit register in the I/O space location $3D ($5D). As the AT90S4433 data memory has $0DF locations, eight bits are used.

	7	6	5	4	3	2	1	0	
$3D ($5D)	**SP7**	**SP6**	**SP5**	**SP4**	**SP3**	**SP2**	**SP1**	**SP0**	**SP**
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 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 one when data is pushed onto the stack with the PUSH instruction, and it is decremented by two when an address is pushed onto the Stack with subroutine calls and interrupts. The Stack Pointer is incremented by one when data is popped from the stack with the POP instruction, and it is incremented by two when an address is popped from the Stack with return from subroutine RET or return from interrupt RETI.

Reset and Interrupt Handling

The AT90S4433 provides 13 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, which 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 3. 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 3. Reset and Interrupt Vectors

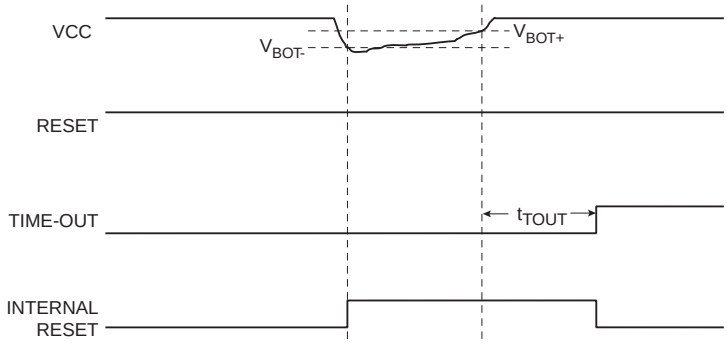
Vector No.	Program Address	Source	Interrupt Definition
1	$000	RESET	External Pin, Power-on Reset, Brown-out Reset and Watchdog Reset
2	$001	INT0	External Interrupt Request 0
3	$002	INT1	External Interrupt Request 1
4	$003	TIMER1 CAPT	Timer/Counter1 Capture Event
5	$004	TIMER1 COMP	Timer/Counter1 Compare Match
6	$005	TIMER1 OVF	Timer/Counter1 Overflow
7	$006	TIMER0 OVF	Timer/Counter0 Overflow
8	$007	SPI, STC	Serial Transfer Complete
9	$008	UART, RX	UART, Rx Complete
10	$009	UART, UDRE	UART Data Register Empty
11	$00A	UART, TX	UART, Tx Complete
12	$00B	ADC	ADC Conversion Complete
13	$00C	EE_RDY	EEPROM Ready
14	$00D	ANA_COMP	Analog Comparator

Brown-out Detection

AT90S4433 has an On-chip Brown-out Detection (BOD) circuit for monitoring the V_{CC} level during the operation. The power supply must be decoupled with a 47 nF to 100 nF capacitor if the BOD function is used. The BOD circuit can be enabled/disabled by the fuse BODEN. When BODEN is enabled (BODEN programmed), and V_{CC} decreases to a value below the trigger level, the Brown-out Reset is immediately activated. When V_{CC} increases above the trigger level, the Brown-out Reset is deactivated after a delay. The delay is defined by the user in the same way as the delay of POR signal (see Table 5). The trigger level for the BOD can be selected by the fuse BODLEVEL to be 2.7V (BODLEVEL unprogrammed), or 4.0V (BODLEVEL programmed). The trigger level has a hysteresis of 50 mV to ensure spike-free Brown-out Detection.

The BOD circuit will only detect a drop in V_{CC} if the voltage stays below the trigger level for longer than 3 µs for trigger level 4.0V, 7 µs for trigger level 2.7V (typical values).

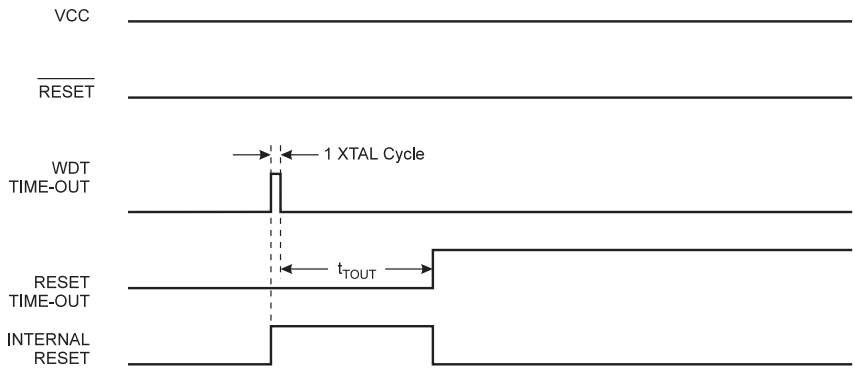
Figure 28. Brown-out Reset during Operation



Watchdog Reset

When the Watchdog times out, it will generate a short reset pulse of one XTAL cycle duration. On the falling edge of this pulse, the delay timer starts counting the Time-out period (t_{TOUT}). See page 43 for details on operation of the Watchdog.

Figure 29. Watchdog Reset during Operation



External Interrupts

The External Interrupts are triggered by the INT1 and INT0 pins. Observe that, if enabled, the interrupts will trigger even if the INT0/INT1 pins are configured as outputs. This feature provides a way of generating a software interrupt. The External Interrupts can be triggered by a falling or rising edge or a low level. This is set up as indicated in the specification for the MCU Control Register (MCUCR). When the External Interrupt is enabled and is configured as level triggered, the interrupt will trigger as long as the pin is held low.

The External Interrupts are set up as described in the specification for the MCU Control Register (MCUCR).

Interrupt Response Time

The interrupt execution response for all the enabled AVR interrupts is four clock cycles minimum. Four clock cycles after the Interrupt Flag has been set, the Program Vector address for the actual interrupt handling routine is executed. During this four clock cycle period, the Program Counter (two bytes) is pushed onto the Stack, and the Stack Pointer is decremented by two. The vector is normally a relative jump to the interrupt routine, and this jump takes two clock cycles. If an interrupt occurs during execution of a multi-cycle instruction, this instruction is completed before the interrupt is served.

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

MCU Control Register – MCUCR

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

Bit	7	6	5	4	3	2	1	0	
$35 ($55)	–	–	SE	SM	ISC11	ISC10	ISC01	ISC00	MCUCR
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7, 6 – Res: Reserved Bits**

These bits are reserved bits in the AT90S4433 and always read as zero.

- **Bit 5 – SE: Sleep Enable**

The SE bit must be set (one) to make the MCU enter the sleep mode when the SLEEP instruction is executed. To avoid having the MCU entering the sleep mode unless it is the programmer's purpose, it is recommended that the Sleep Enable SE bit be set just before the execution of the SLEEP instruction.

- **Bit 4 – SM: Sleep Mode**

This bit selects between the two available sleep modes. When SM is cleared (zero), Idle mode is selected as sleep mode. When SM is set (one), Power-down mode is selected as sleep mode. For details, refer to the paragraph "Sleep Modes" below.

- **Bits 3, 2 – ISC11, ISC10: Interrupt Sense Control 1 Bit 1 and Bit 0**

The External Interrupt 1 is activated by the external pin INT1 if the SREG I-flag and the corresponding interrupt mask in the GIMSK are set. The level and edges on the external INT1 pin that activate the interrupt are defined in Table 7.

Table 7. Interrupt 1 Sense Control

ISC11	ISC10	Description
0	0	The low level of INT1 generates an interrupt request.
0	1	Any logical change on INT1 generates an interrupt request.
1	0	The falling edge of INT1 generates an interrupt request.
1	1	The rising edge of INT1 generates an interrupt request.

The value on the INT1 pin is sampled before detecting edges. If edge or toggle interrupt is selected, pulses that last longer than one clock period will generate an interrupt. Shorter pulses are not guaranteed to generate an interrupt. If low-level interrupt is selected, the low level must be held until the completion of the currently executing instruction to generate an interrupt.

- **Bits 1, 0 – ISC01, ISC00: Interrupt Sense Control 0 Bit 1 and Bit 0**

The External Interrupt 0 is activated by the external pin INT0 if the SREG I-flag and the corresponding interrupt mask are set. The level and edges on the external INT0 pin that activate the interrupt are defined in Table 8.

Table 8. Interrupt 0 Sense Control

ISC01	ISC00	Description
0	0	The low level of INT0 generates an interrupt request.
0	1	Any logical change on INT0 generates an interrupt request.
1	0	The falling edge of INT0 generates an interrupt request.
1	1	The rising edge of INT0 generates an interrupt request.

The value on the INT0 pin is sampled before detecting edges. If edge or toggle interrupt is selected, pulses that last longer than one clock period will generate an interrupt. Shorter pulses are not guaranteed to generate an interrupt. If low-level interrupt is selected, the low level must be held until the completion of the currently executing instruction to generate an interrupt.

Sleep Modes

To enter the sleep modes, the SE bit in MCUCR must be set (one) and a SLEEP instruction must be executed. The SM bit in the MCUCR Register selects which sleep mode (Idle or Power-down) will be activated by the SLEEP instruction. If an enabled interrupt occurs while the MCU is in a sleep mode, the MCU wakes up, executes the interrupt routine, and resumes execution from the instruction following SLEEP. The contents of the Register File and I/O memory are unaltered. If a reset occurs during sleep mode, the MCU wakes up and executes from the Reset Vector.

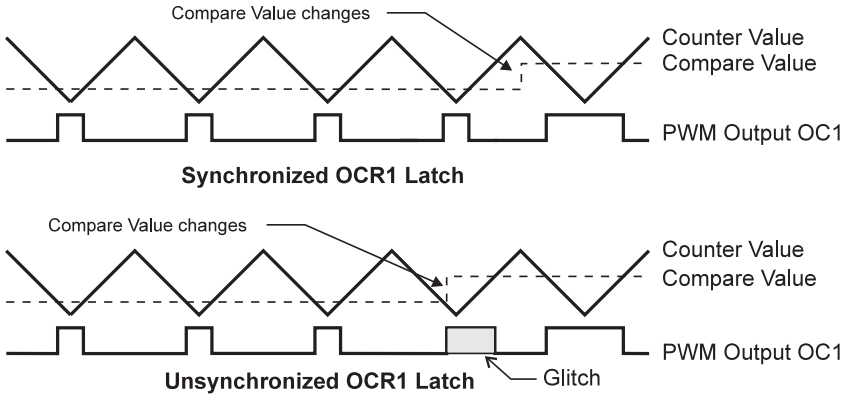
Note that if a level-triggered interrupt is used for wake-up from Power-down, the low level must be held for a time longer than the reset delay Time-out period (t_{TOUT}). Otherwise, the device will not wake up.

Table 14. Compare1 Mode Select in PWM Mode

COM11	COM10	Effect on OC1
0	0	Not connected
0	1	Not connected
1	0	Cleared on compare match, up-counting. Set on compare match, down-counting (non-inverted PWM).
1	1	Cleared on compare match, down-counting. Set on compare match, up-counting (inverted PWM).

Note that in the PWM mode, the ten least significant OCR1 bits, when written, are transferred to a temporary location. They are latched when Timer/Counter1 reaches TOP. This prevents the occurrence of odd-length PWM pulses (glitches) in the event of an unsynchronized OCR1 write. See Figure 34 for an example.

Figure 34. Effects on Unsynchronized OCR1 Latching



During the time between the write and the latch operation, a read from OCR1 will read the contents of the temporary location. This means that the most recently written value always will read out of OCR1.

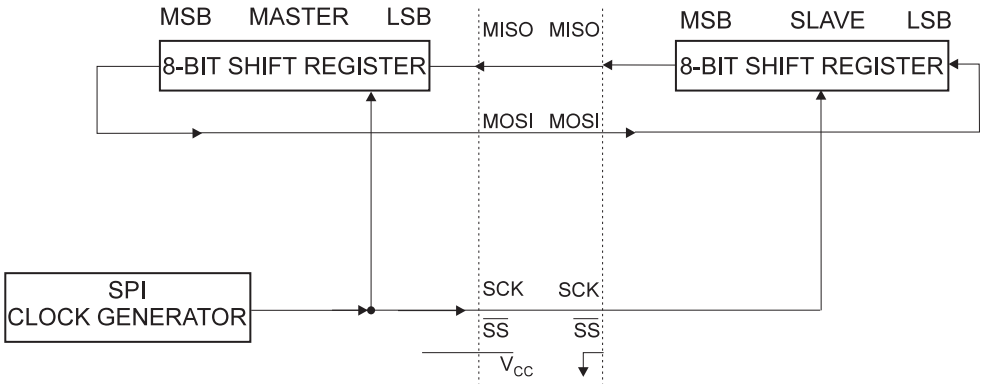
When OCR1 contains $0000 or TOP, the output OC1 is updated to low or high on the next compare match according to the settings of COM11 and COM10. This is shown in Table 15.

Table 15. PWM Outputs OCR = $0000 or TOP

COM11	COM10	OCR1	Output OC1
1	0	$0000	L
1	0	TOP	H
1	1	$0000	H
1	1	TOP	L

In PWM mode, the Timer Overflow Flag1 (TOV1) is set when the counter changes direction at $0000. Timer Overflow Interrupt1 operates exactly as in normal Timer/Counter mode, i.e., it is executed when TOV1 is set, provided that Timer Overflow Interrupt1 and global interrupts are enabled. This also applies to the Timer Output Compare1 Flag and interrupt.

Figure 37. 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 17.

Table 17. SPI Pin Direction Overrides[1]

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

Note: 1. See “Alternate Functions of Port B” on page 73 for a detailed description of how to define the direction of the user-defined SPI pins.

$\overline{SS}$ Pin Functionality

When the SPI is configured as a Master (MSTR in SPCR is set), the user can determine the direction of the $\overline{SS}$ pin. If $\overline{SS}$ is configured as an output, the pin is a general output pin, which does not affect the SPI system. If $\overline{SS}$ is configured as an input, it must be held high to ensure Master SPI operation. If the $\overline{SS}$ pin is driven low by peripheral circuitry when the SPI is configured as master with the $\overline{SS}$ pin defined as an input, the SPI system interprets this as another master selecting the SPI as a slave and starts to send data to it. To avoid bus contention, the SPI system takes the following actions:

1. The MSTR bit in SPCR is cleared and the SPI system becomes a Slave. As a result of the SPI becoming a Slave, the MOSI and SCK pins become inputs.
2. The SPIF Flag in SPSR is set, and if the SPI interrupt is enabled and the I-bit in SREG is set, the interrupt routine will be executed.

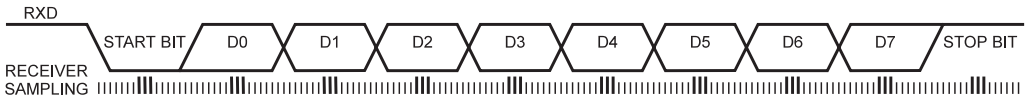
Thus, when interrupt-driven SPI transmittal is used in Master mode, and there exists a possibility that $\overline{SS}$ is driven low, the interrupt should always check that the MSTR bit is still set. Once the MSTR bit has been cleared by a slave select, it must be set by the user to re-enable the SPI Master mode.

When the SPI is configured as a slave, the $\overline{SS}$ pin is always input. When $\overline{SS}$ is held low, the SPI is activated and MISO becomes an output if configured so by the user. All other

The Receiver front-end logic samples the signal on the RXD pin at a frequency 16 times the baud rate. While the line is idle, one single sample of logical “0” will be interpreted as the falling edge of a start bit, and the start bit detection sequence is initiated. Let sample 1 denote the first zero-sample. Following the 1-to-0 transition, the Receiver samples the RXD pin at samples 8, 9, and 10. If two or more of these three samples are found to be logical “1”s, the start bit is rejected as a noise spike and the receiver starts looking for the next 1-to-0 transition.

If, however, a valid start bit is detected, sampling of the data bits following the start bit is performed. These bits are also sampled at samples 8, 9, and 10. The logical value found in at least two of the three samples is taken as the bit value. All bits are shifted into the transmitter Shift Register as they are sampled. Sampling of an incoming character is shown in Figure 42.

Figure 42. Sampling Received Data



When the stop bit enters the receiver, the majority of the three samples must be one to accept the stop bit. If two or more samples are logical “0”s, the Framing Error (FE) Flag in the UART Control and Status Register A (UCSRA) is set. Before reading the UDR Register, the user should always check the FE bit to detect Framing Errors.

Whether or not a valid stop bit is detected at the end of a character reception cycle, the data is transferred to UDR and the RXC Flag in UCSRA is set. UDR is, in fact, two physically separate registers: one for Transmitted Data and one for Received Data. When UDR is read, the Receive Data Register is accessed, and when UDR is written, the Transmit Data Register is accessed. If 9-bit data word is selected (the CHR9 bit in the UART Control and Status Register B, UCSRB is set), the RXB8 bit in UCSRB is loaded with bit nine in the Transmit Shift Register when data is transferred to UDR.

If, after having received a character, the UDR Register has not been read since the last receive, the OverRun (OR) Flag in UCSRB is set. This means that the last data byte shifted into the Shift Register could not be transferred to UDR and has been lost. The OR bit is buffered and is updated when the valid data byte in UDR is read. Thus, the user should always check the OR bit after reading the UDR Register in order to detect any overruns if the baud rate is high or CPU load is high.

When the RXEN bit in the UCSRB Register is cleared (zero), the receiver is disabled. This means that the PD0 pin can be used as a general I/O pin. When RXEN is set, the UART receiver will be connected to PD0, which is forced to be an input pin regardless of the setting of the DDD0 bit in DDRD. When PD0 is forced to input by the UART, the PORTD0 bit can still be used to control the pull-up resistor on the pin.

When the CHR9 bit in the UCSRB Register is set, transmitted and received characters are nine bits long, plus start and stop bits. The ninth data bit to be transmitted is the TXB8 bit in UCSRB Register. This bit must be set to the wanted value before a transmission is initiated by writing to the UDR Register. The ninth data bit received is the RXB8 bit in the UCSRB Register.

Table 19. UBR Settings at Various Crystal Frequencies

Baud Rate	1 MHz		%Error	1.8432 MHz		%Error	2 MHz		%Error	2.4576 MHz		%Error
2400	UBR=	**25**	**0.2**	UBR=	**47**	**0.0**	UBR=	**51**	**0.2**	UBR=	**63**	**0.0**
4800	UBR=	**12**	**0.2**	UBR=	**23**	**0.0**	UBR=	**25**	**0.2**	UBR=	**31**	**0.0**
9600	UBR=	6	7.5	UBR=	**11**	**0.0**	UBR=	**12**	**0.2**	UBR=	**15**	**0.0**
14400	UBR=	3	7.8	UBR=	**7**	**0.0**	UBR=	8	3.7	UBR=	10	3.1
19200	UBR=	2	7.8	UBR=	**5**	**0.0**	UBR=	6	7.5	UBR=	**7**	**0.0**
28800	UBR=	1	7.8	UBR=	**3**	**0.0**	UBR=	3	7.8	UBR=	4	6.3
38400	UBR=	1	22.9	UBR=	**2**	**0.0**	UBR=	2	7.8	UBR=	**3**	**0.0**
57600	UBR=	0	7.8	UBR=	**1**	**0.0**	UBR=	1	7.8	UBR=	2	12.5
76800	UBR=	0	22.9	UBR=	1	33.3	UBR=	1	22.9	UBR=	**1**	**0.0**
115200	UBR=	0	84.3	UBR=	**0**	**0.0**	UBR=	0	7.8	UBR=	0	25.0

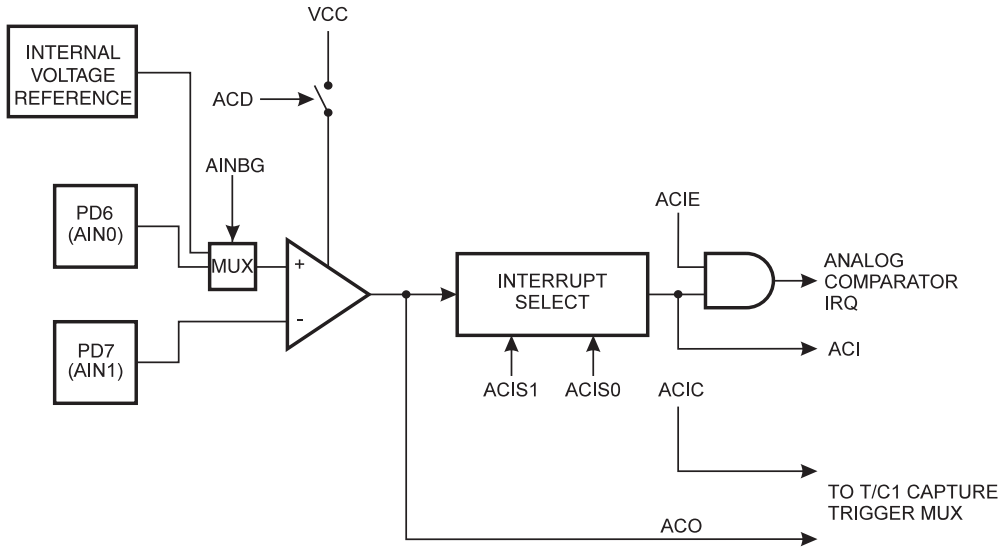
Baud Rate	3.2768 MHz		%Error	3.6864 MHz		%Error	4 MHz		%Error	4.608 MHz		%Error
2400	UBR=	**84**	**0.4**	UBR=	**95**	**0.0**	UBR=	**103**	**0.2**	UBR=	**119**	**0.0**
4800	UBR=	**42**	**0.8**	UBR=	**47**	**0.0**	UBR=	**51**	**0.2**	UBR=	**59**	**0.0**
9600	UBR=	**20**	**1.6**	UBR=	**23**	**0.0**	UBR=	**25**	**0.2**	UBR=	**29**	**0.0**
14400	UBR=	**13**	**1.6**	UBR=	**15**	**0.0**	UBR=	16	2.1	UBR=	**19**	**0.0**
19200	UBR=	10	3.1	UBR=	**11**	**0.0**	UBR=	**12**	**0.2**	UBR=	**14**	**0.0**
28800	UBR=	**6**	**1.6**	UBR=	**7**	**0.0**	UBR=	8	3.7	UBR=	**9**	**0.0**
38400	UBR=	4	6.3	UBR=	**5**	**0.0**	UBR=	6	7.5	UBR=	7	6.7
57600	UBR=	3	12.5	UBR=	**3**	**0.0**	UBR=	3	7.8	UBR=	**4**	**0.0**
76800	UBR=	2	12.5	UBR=	**2**	**0.0**	UBR=	2	7.8	UBR=	3	6.7
115200	UBR=	1	12.5	UBR=	**1**	**0.0**	UBR=	1	7.8	UBR=	2	20.0

Baud Rate	7.3728 MHz		%Error	8 MHz		%Error	9.216 MHz		%Error	11.059 MHz		%Error
2400	UBR=	**191**	**0.0**	UBR=	**207**	**0.2**	UBR=	**239**	**0.0**	UBR=	287	-
4800	UBR=	**95**	**0.0**	UBR=	**103**	**0.2**	UBR=	**119**	**0.0**	UBR=	**143**	**0.0**
9600	UBR=	**47**	**0.0**	UBR=	**51**	**0.2**	UBR=	**59**	**0.0**	UBR=	**71**	**0.0**
14400	UBR=	**31**	**0.0**	UBR=	**34**	**0.8**	UBR=	**39**	**0.0**	UBR=	**47**	**0.0**
19200	UBR=	**23**	**0.0**	UBR=	**25**	**0.2**	UBR=	**29**	**0.0**	UBR=	**35**	**0.0**
28800	UBR=	**15**	**0.0**	UBR=	16	2.1	UBR=	**19**	**0.0**	UBR=	**23**	**0.0**
38400	UBR=	**11**	**0.0**	UBR=	**12**	**0.2**	UBR=	**14**	**0.0**	UBR=	**17**	**0.0**
57600	UBR=	**7**	**0.0**	UBR=	8	3.7	UBR=	**9**	**0.0**	UBR=	**11**	**0.0**
76800	UBR=	**5**	**0.0**	UBR=	6	7.5	UBR=	7	6.7	UBR=	**8**	**0.0**
115200	UBR=	**3**	**0.0**	UBR=	3	7.8	UBR=	**4**	**0.0**	UBR=	**5**	**0.0**

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.

Figure 47. ADC Timing Diagram, Single Conversion

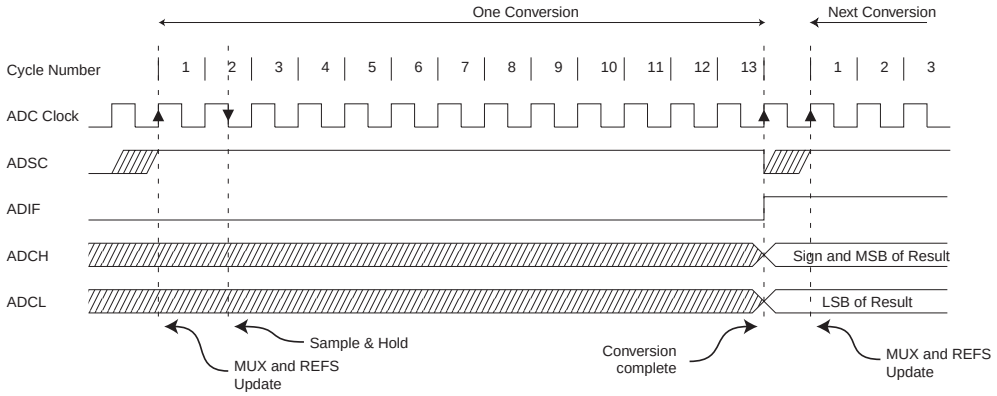
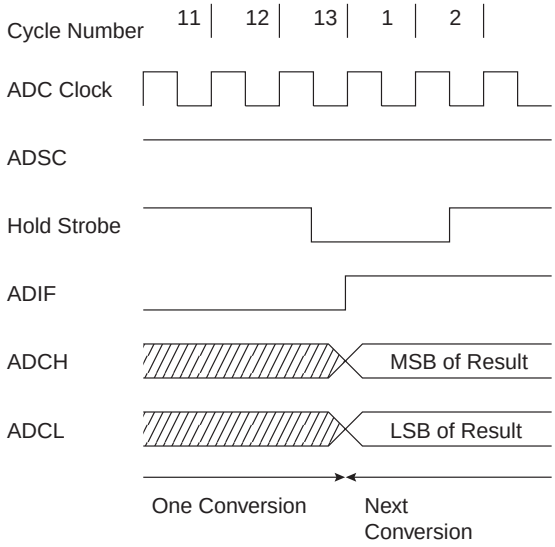


Figure 48. ADC Timing Diagram, Free Run Conversion



ADC Noise Canceler Function

The ADC features a Noise Canceler that enables conversion during Idle mode to reduce noise induced from the CPU core. To make use of this feature, the following procedure should be used:

1. Make sure that the ADC is enabled and is not busy converting. Single Conversion mode must be selected and the ADC conversion complete interrupt must be enabled. Thus:
   ADEN = 1
   ADSC = 0
   ADFR = 0
   ADIE = 1
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.

The Port B Input Pins address (PINB) is not a register; this address enables access to the physical value on each Port B pin. When reading PORTB, the Port B Data Latch is read, and when reading PINB, the logical values present on the pins are read.

Port B as General Digital I/O

All six pins in Port B have equal functionality when used as digital I/O pins.

PBn, general I/O pin: The DDBn bit in the DDRB Register selects the direction of this pin. If DDBn is set (one), PBn is configured as an output pin. If DDBn is cleared (zero), PBn is configured as an input pin. If PORTBn 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 PORTBn 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 24. DDBn Effects on Port B Pins[1]

DDBn	PORTBn	I/O	Pull-up	Comment
0	0	Input	No	Tri-state (high-Z)
0	1	Input	Yes	PBn 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: 5..0, pin number.

Alternate Functions of Port B

The alternate pin configuration is as follows:

- **SCK – Port B, Bit 5**

SCK: Master Clock output, Slave Clock input pin for SPI channel. When the SPI is enabled as a Slave, this pin is configured as an input, regardless of the setting of DDB5. When the SPI is enabled as a Master, the data direction of this pin is controlled by DDB5. When the pin is forced to be an input, the pull-up can still be controlled by the PORTB5 bit. See the description of the SPI port for further details.

- **MISO – Port B, Bit 4**

MISO: Master Data input, Slave Data output pin for SPI channel. When the SPI is enabled as a Master, this pin is configured as an input, regardless of the setting of DDB4. When the SPI is enabled as a Slave, the data direction of this pin is controlled by DDB4. When the pin is forced to be an input, the pull-up can still be controlled by the PORTB4 bit. See the description of the SPI port for further details.

- **MOSI – Port B, Bit 3**

MOSI: SPI Master Data output, Slave Data input for SPI channel. When the SPI is enabled as a Slave, this pin is configured as an input, regardless of the setting of DDB3. When the SPI is enabled as a Master, the data direction of this pin is controlled by DDB3. When the pin is forced to be an input, the pull-up can still be controlled by the PORTB3 bit. See the description of the SPI port for further details.

- **$\overline{SS}$ – Port B, Bit 2**

$\overline{SS}$: Slave Port Select input. When the SPI is enabled as a Slave, this pin is configured as an input, regardless of the setting of DDB2. As a Slave, the SPI is activated when this pin is driven low. When the SPI is enabled as a Master, the data direction of this pin is

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.

- **TXD – Port D, Bit 1**

Transmit Data (Data Output pin for the UART). When the UART Transmitter is enabled, this pin is configured as an output, regardless of the value of DDD1.

- **RXD – Port D, Bit 0**

Receive Data (Data Input pin for the UART). When the UART Receiver is enabled, this pin is configured as an input, regardless of the value of DDD0. When the UART forces this pin to be an input, a logical “1” in PORTD0 will turn on the internal pull-up.

Port D Schematics

Note that all port pins are synchronized. The synchronization latches are, however, not shown in the figures.

Figure 57. Port D Schematic Diagram (Pin PD0)

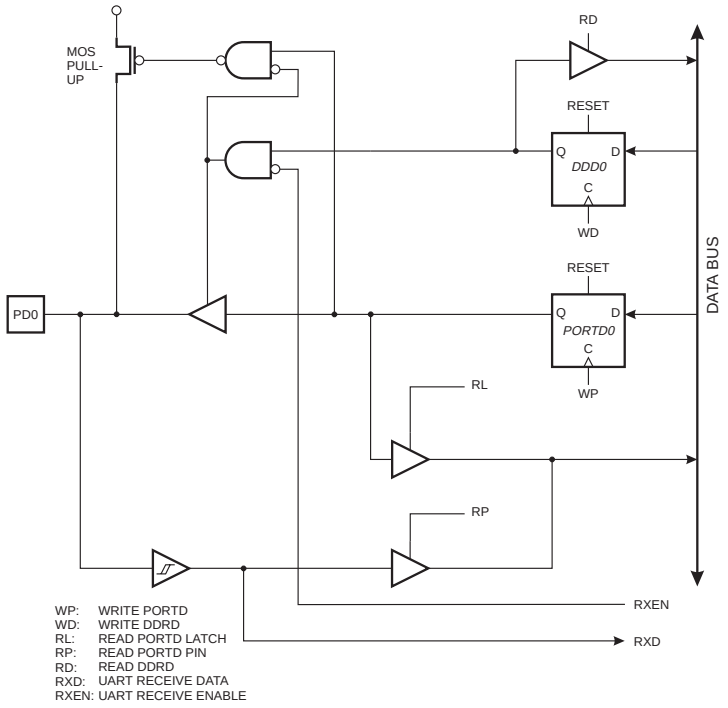


Figure 75. Power-down Supply Current vs. V_{CC}

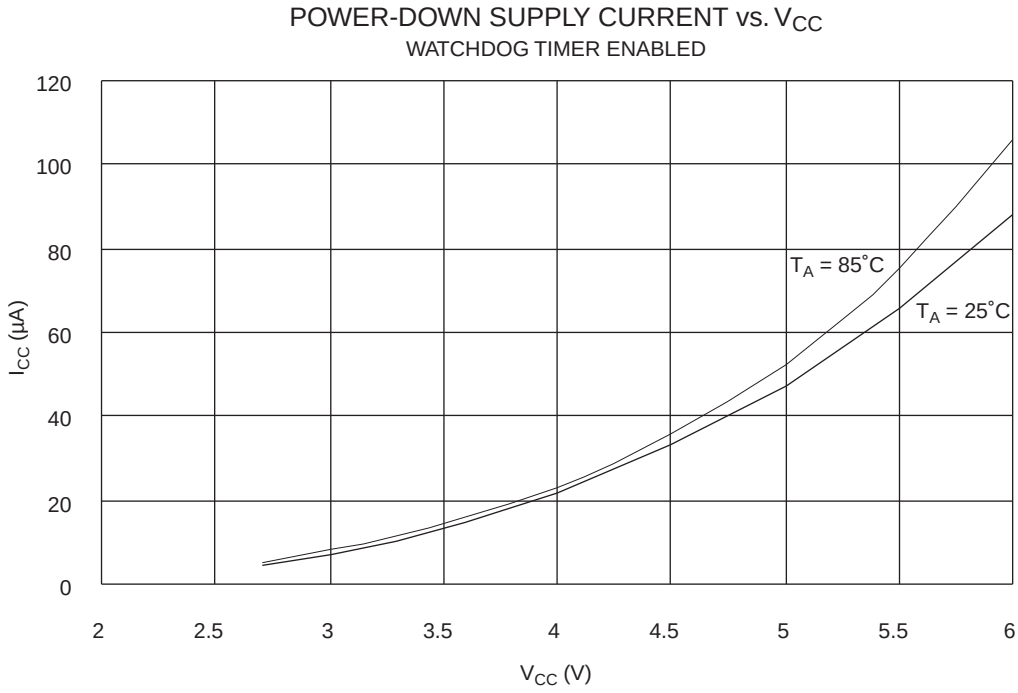


Figure 76. Power-down Supply Current vs. V_{CC}

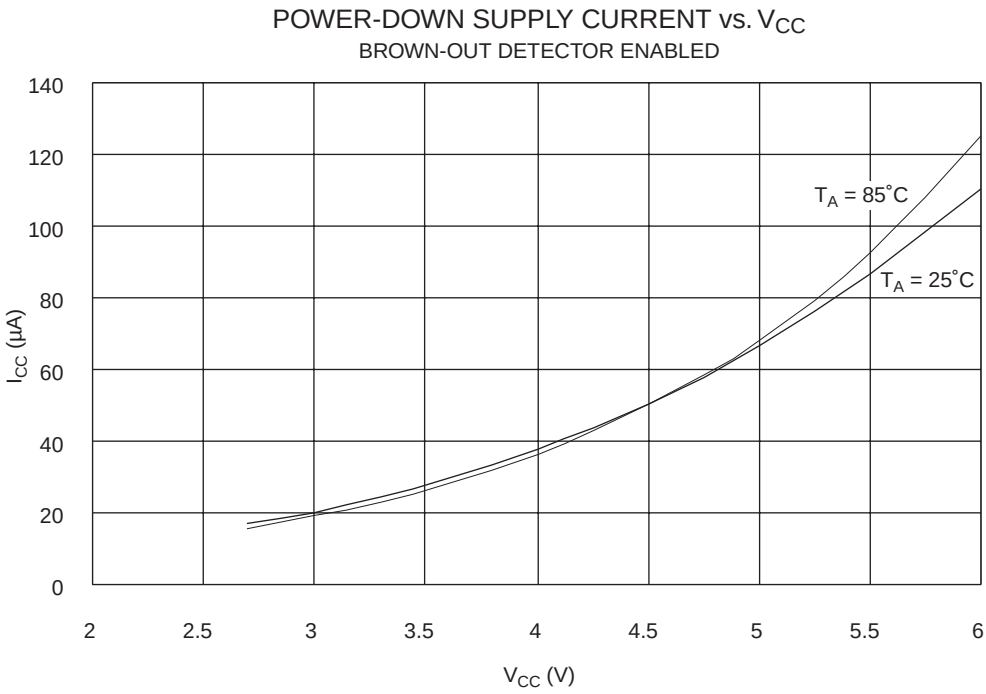


Figure 85. I/O Pin Source Current vs. Output Voltage

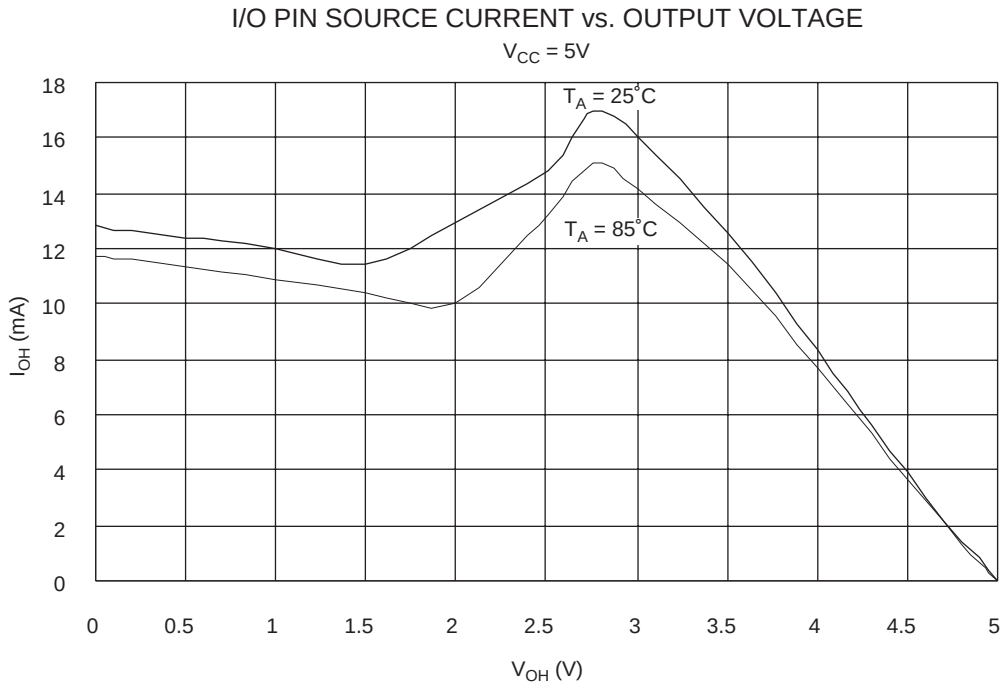


Figure 86. I/O Pin Sink Current vs. Output Voltage