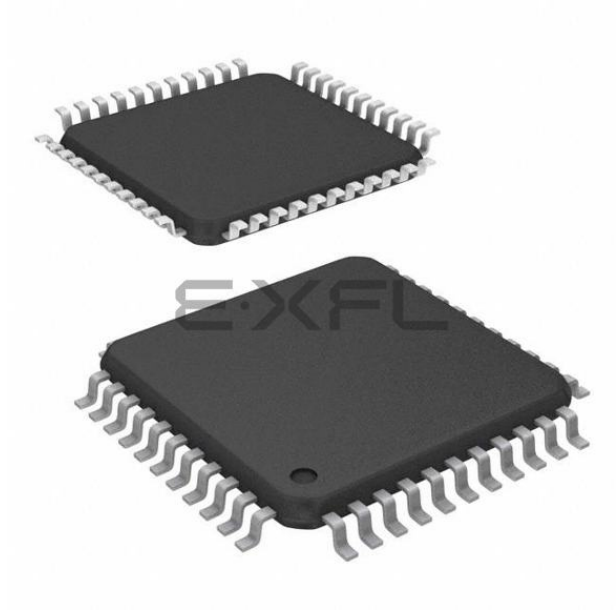
Microchip Technology - ATMEGA644PV-10AQR Datasheet



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	Active
Core Processor	AVR
Core Size	8-Bit
Speed	10MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	32
Program Memory Size	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	2K x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	44-TQFP
Supplier Device Package	44-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atmega644pv-10aqr

1. Description

The Atmel® ATmega644P is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega644P achieves throughputs close to 1MIPS per MHz. This empowers system designer to optimize the device for power consumption versus processing speed.

The Atmel AVR® core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in a single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

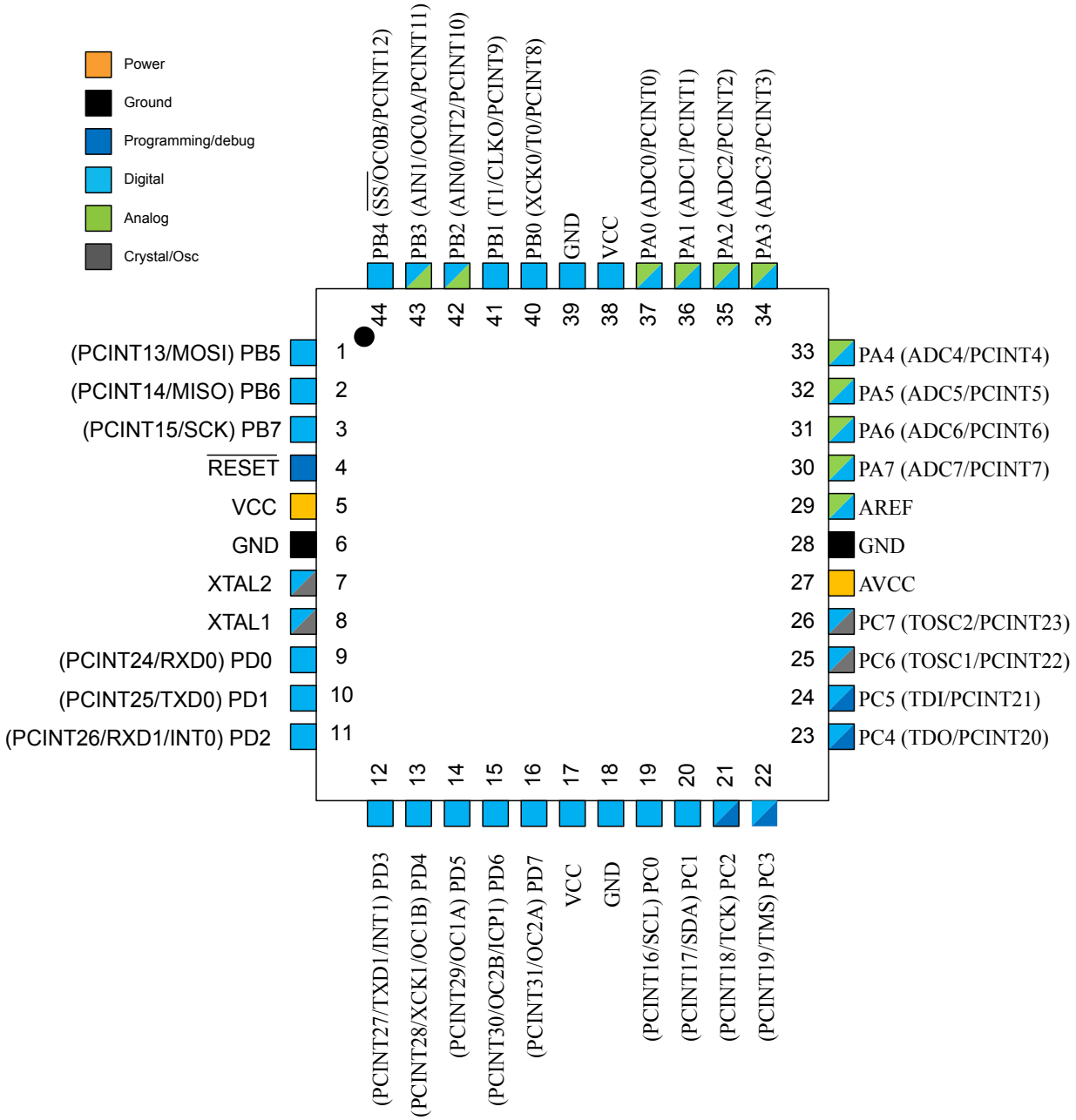
The ATmega644P provides the following features: 64Kbytes of In-System Programmable Flash with Read-While-Write capabilities, 2Kbytes EEPROM, 4Kbytes SRAM, 32 general purpose I/O lines, 32 general purpose working registers, Real Time Counter (RTC), three flexible Timer/Counters with compare modes and PWM, two serial programmable USARTs , one byte-oriented 2-wire Serial Interface (I2C), a 8-channel 10-bit ADC with optional differential input stage with programmable gain, a programmable Watchdog Timer with internal Oscillator, an SPI serial port, IEEE std. 1149.1 compliant JTAG test interface, also used for accessing the On-chip Debug system and programming and six software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next interrupt or hardware reset. In Power-save mode, the asynchronous timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except asynchronous timer and ADC to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low power consumption. In Extended Standby mode, both the main oscillator and the asynchronous timer continue to run.

Atmel offers the QTouch® library for embedding capacitive touch buttons, sliders and wheels functionality into AVR microcontrollers. The patented charge-transfer signal acquisition offers robust sensing and includes fully debounced reporting of touch keys and includes Adjacent Key Suppression® (AKS™) technology for unambiguous detection of key events. The easy-to-use QTouch Suite toolchain allows you to explore, develop and debug your own touch applications.

The device is manufactured using Atmel's high density non-volatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed In-System through an SPI serial interface, by a conventional nonvolatile memory programmer, or by an On-chip Boot program running on the AVR core. The Boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega644P is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The ATmega644P is supported with a full suite of program and system development tools including: C Compilers, Macro Assemblers, Program Debugger/Simulators, In-Circuit Emulators, and Evaluation kits.

5.1.2. TQFN and QFN



5.2. Pin Descriptions

5.2.1. VCC

Digital supply voltage.

5.2.2. GND

Ground.

5.2.3. Port A (PA[7:0])

This port serves as analog inputs to the Analog-to-digital Converter.

This is an 8-bit, bi-directional I/O port with internal pull-up resistors, individually selectable for each bit. The output buffers have symmetrical drive characteristics, with both high sink and source capability. As inputs, the port pins that are externally pulled low will source current if pull-up resistors are activated. Port pins are tri-stated when a reset condition becomes active, even if the clock is not running.

5.2.4. Port B (PB[7:0])

This is an 8-bit, bi-directional I/O port with internal pull-up resistors, individually selectable for each bit. The output buffers have symmetrical drive characteristics, with both high sink and source capability. As inputs, the port pins that are externally pulled low will source current if pull-up resistors are activated. Port pins are tri-stated when a reset condition becomes active, even if the clock is not running.

This port also serves the functions of various special features.

5.2.5. Port C (PC[7:0])

This is an 8-bit, bi-directional I/O port with internal pull-up resistors, individually selectable for each bit. The output buffers have symmetrical drive characteristics, with both high sink and source capability. As inputs, the port pins that are externally pulled low will source current if pull-up resistors are activated. Port pins are tri-stated when a reset condition becomes active, even if the clock is not running.

This port also serves the functions of the JTAG interface, along with special features.

5.2.6. Port D (PD[7:0])

This is an 8-bit, bi-directional I/O port with internal pull-up resistors, individually selectable for each bit. The output buffers have symmetrical drive characteristics, with both high sink and source capability. As inputs, the port pins that are externally pulled low will source current if pull-up resistors are activated. Port pins are tri-stated when a reset condition becomes active, even if the clock is not running.

This port also serves the functions of various special features.

5.2.7. $\overline{\text{RESET}}$

Reset input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a reset.

5.2.8. XTAL1

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

5.2.9. XTAL2

Output from the inverting Oscillator amplifier.

5.2.10. AVCC

AVCC is the supply voltage pin for Port A and the Analog-to-digital Converter. It should be externally connected to V_{CC}, even if the ADC is not used. If the ADC is used, it should be connected to V_{CC} through a low-pass filter.

5.2.11. AREF

This is the analog reference pin for the Analog-to-digital Converter.

6. I/O Multiplexing

Each pin is by default controlled by the PORT as a general purpose I/O and alternatively it can be assigned to one of the peripheral functions.

The following table describes the peripheral signals multiplexed to the PORT I/O pins.

Table 6-1. PORT Function Multiplexing

32-pin TQFP/ QFN/ MLF Pin #	40-pin PDIP Pin #	PAD	EXTINT	PCINT	ADC/AC	OSC	T/C # 0	T/C # 1	USART	I2C	SPI	JTAG
1	6	PB[5]		PCINT13							MOSI	
2	7	PB[6]		PCINT14							MISO	
3	8	PB[7]		PCINT15							SCK	
4	9	$\overline{\text{RESET}}$										
5	10	VCC										
6	11	GND										
7	12	XTAL2										
8	13	XTAL1										
9	14	PD[0]		PCINT24					RxD0			
10	15	PD[1]		PCINT25					TxD0			
11	16	PD[2]	INT0	PCINT26					RxD1			
12	17	PD[3]	INT1	PCINT27					TXD1			
13	18	PD[4]		PCINT28				OC1B	XCK1			
14	19	PD[5]		PCINT29				OC1A				
15	20	PD[6]		PCINT30			OC2B	ICP1				
16	21	PD[7]		PCINT31			OC2A					
17	-	VCC							RxD2		MISO1	
18	-	GND							TxD2		MOSI1	
19	22	PC[0]		PCINT16						SCL		
20	23	PC[1]		PCINT17						SDA		
21	24	PC[2]		PCINT18								TCK
22	25	PC[3]		PCINT19								TMS
23	26	PC[4]		PCINT20								TDO
24	27	PC[5]		PCINT21								TDI
25	28	PC[6]		PCINT22		TOSC1						
26	29	PC[7]		PCINT23		TOSC2						
27	30	AVCC										
28	31	GND										
29	32	AREF			AREF							
30	33	PA[7]		PCINT7	ADC7							
31	34	PA[6]		PCINT6	ADC6							
32	35	PA[5]		PCINT5	ADC5							
33	36	PA[4]		PCINT4	ADC4							
34	37	PA[3]		PCINT3	ADC3							

9. AVR Memories

9.1. Overview

This section describes the different memory types in the device. The AVR architecture has two main memory spaces, the Data Memory and the Program Memory space. In addition, the device features an EEPROM Memory for data storage. All memory spaces are linear and regular.

9.2. In-System Reprogrammable Flash Program Memory

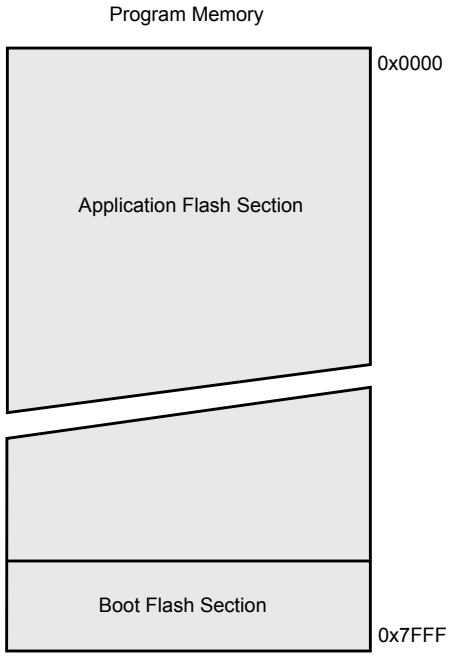
The ATmega644P contains 64Kbytes On-chip In-System Reprogrammable Flash memory for program storage. Since all AVR instructions are 16 or 32 bits wide, the Flash is organized as 64 x 16.

The Flash memory has an endurance of at least 10,000 write/erase cycles. The ATmega644P Program Counter (PC) is 15 bits wide, thus addressing the 64 program memory locations. The operation of Boot Program section and associated Boot Lock bits for software protection are described in detail in *Boot Loader Support – Read-While-Write Self-Programming*. Refer to *Memory Programming* for the description on Flash data serial downloading using the SPI pins or the JTAG interface.

Constant tables can be allocated within the entire program memory address space, using the Load Program Memory (LPM) instruction.

Timing diagrams for instruction fetch and execution are presented in *Instruction Exectution Timing*.

Figure 9-1. Program Memory Map ATmega644P



Related Links

BTLDR - Boot Loader Support – Read-While-Write Self-Programming on page 347

MEMPROG- Memory Programming on page 364

Instruction Execution Timing on page 28

10.12.2. Clock Prescaler Register

Name: CLKPR
Offset: 0x61
Reset: Refer to the bit description
Property: -

Bit	7	6	5	4	3	2	1	0
	CLKPCE				CLKPS3	CLKPS2	CLKPS1	CLKPS0
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				x	x	x	x

Bit 7 – CLKPCE: Clock Prescaler Change Enable
The CLKPCE bit must be written to logic one to enable change of the CLKPS bits. The CLKPCE bit is only updated when the other bits in CLKPR are simultaneously written to zero. CLKPCE is cleared by hardware four cycles after it is written or when CLKPS bits are written. Rewriting the CLKPCE bit within this time-out period does neither extend the time-out period, nor clear the CLKPCE bit.

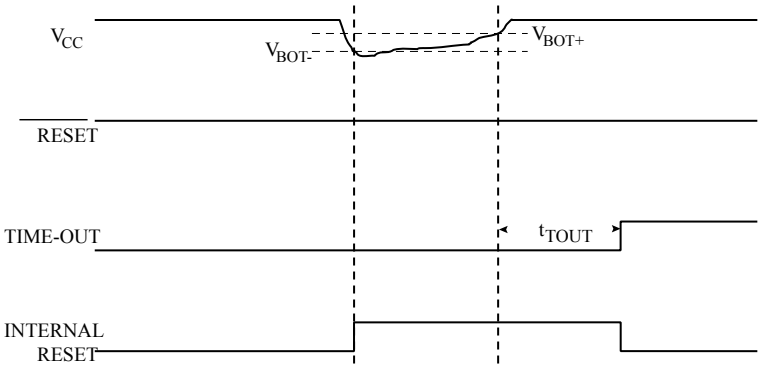
Bits 3:0 – CLKPSn: Clock Prescaler Select n [n = 3:0]
These bits define the division factor between the selected clock source and the internal system clock. These bits can be written run-time to vary the clock frequency to suit the application requirements. As the divider divides the master clock input to the MCU, the speed of all synchronous peripherals is reduced when a division factor is used. The division factors are given in the table below.

The CKDIV8 Fuse determines the initial value of the CLKPS bits. If CKDIV8 is unprogrammed, the CLKPS bits will be reset to “0000”. If CKDIV8 is programmed, CLKPS bits are reset to “0011”, giving a division factor of 8 at start up. This feature should be used if the selected clock source has a higher frequency than the maximum frequency of the device at the present operating conditions. Note that any value can be written to the CLKPS bits regardless of the CKDIV8 Fuse setting. The Application software must ensure that a sufficient division factor is chosen if the selected clock source has a higher frequency than the maximum frequency of the device at the present operating conditions. The device is shipped with the CKDIV8 Fuse programmed.

Table 10-16. Clock Prescaler Select

CLKPS[3:0]	Clock Division Factor
0000	1
0001	2
0010	4
0011	8
0100	16
0101	32
0110	64
0111	128
1000	256
1001	Reserved

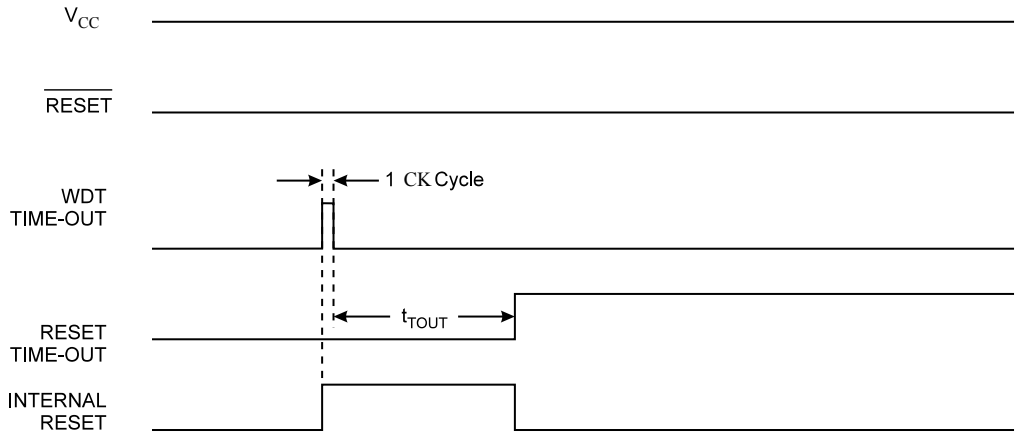
Figure 12-5. Brown-out Reset During Operation



12.6. Watchdog System Reset

When the Watchdog times out, it will generate a short reset pulse of one CK cycle duration. On the falling edge of this pulse, the delay timer starts counting the Time-out period t_{TOUT}.

Figure 12-6. Watchdog System Reset During Operation



12.7. Internal Voltage Reference

The device features an internal bandgap reference. This reference is used for Brown-out Detection, and it can be used as an input to the Analog Comparator or the ADC.

12.7.1. Voltage Reference Enable Signals and Start-up Time

The voltage reference has a start-up time that may influence the way it should be used. To save power, the reference is not always turned on. The reference is on during the following situations:

1. When the BOD is enabled (by programming the BODLEVEL [2:0] Fuses).
2. When the bandgap reference is connected to the Analog Comparator (by setting the ACBG bit in ACSR (ACSR.ACBG)).
3. When the ADC is enabled.

Thus, when the BOD is not enabled, after setting ACSR.ACBG or enabling the ADC, the user must always allow the reference to start up before the output from the Analog Comparator or ADC is used. To reduce power consumption in Power-Down mode, the user can avoid the three conditions above to ensure that the reference is turned off before entering Power-Down mode.

14.1.2.4. Pin Change Interrupt Control Register

Name: PCICR
Offset: 0x68
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
					PCIE3	PCIE2	PCIE1	PCIE0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

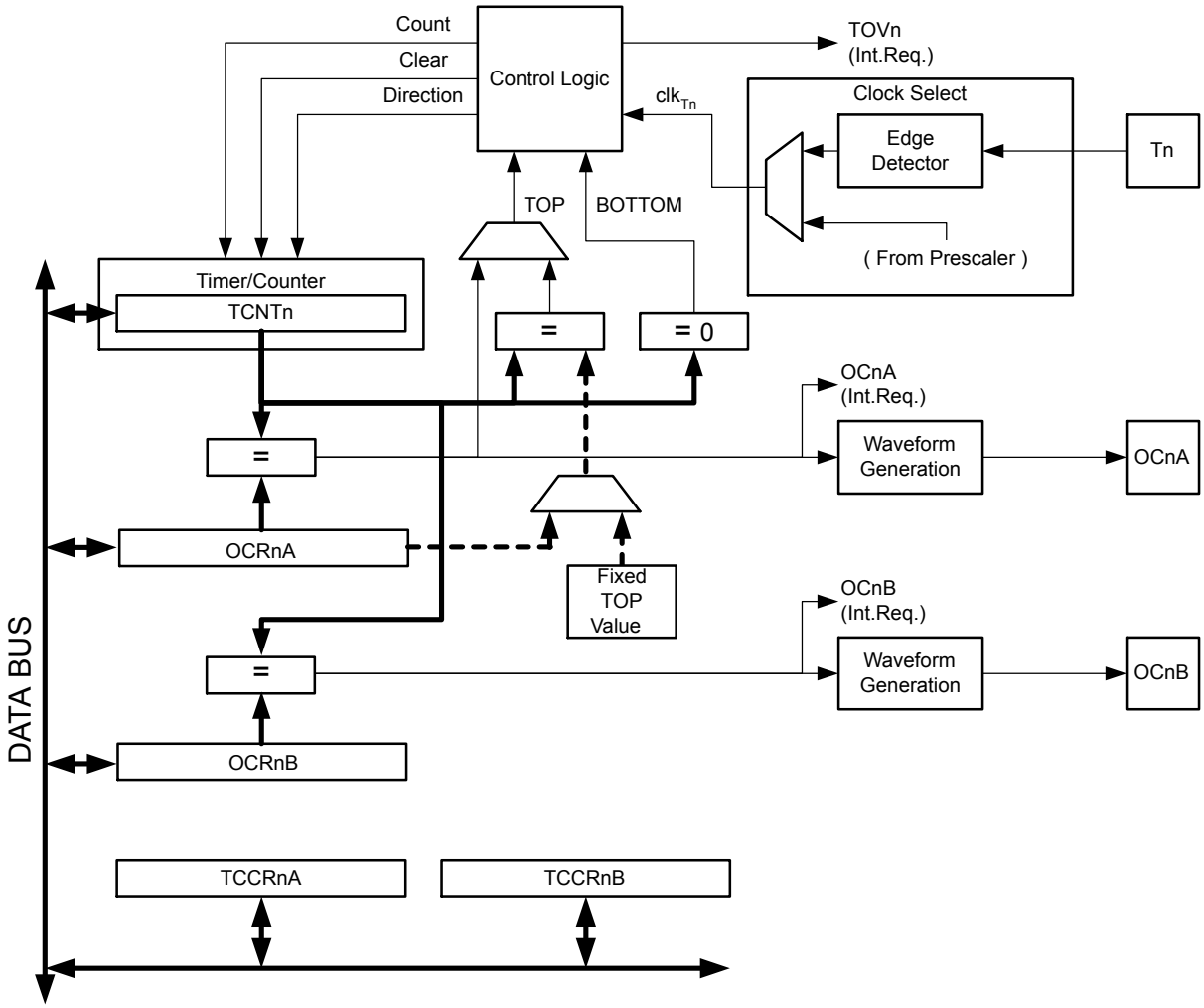
Bit 3 – PCIE3: Pin Change Interrupt Enable 3
When the PCIE3 bit is set and the I-bit in the Status Register (SREG) is set, pin change interrupt 3 is enabled. Any change on any enabled PCINT[31:24] pin will cause an interrupt. The corresponding interrupt of Pin Change Interrupt Request is executed from the PCI3 Interrupt Vector. PCINT[31:24] pins are enabled individually by the PCMSK3 Register.

Bit 2 – PCIE2: Pin Change Interrupt Enable 2
When the PCIE2 bit is set and the I-bit in the Status Register (SREG) is set, pin change interrupt 2 is enabled. Any change on any enabled PCINT[23:16] pin will cause an interrupt. The corresponding interrupt of Pin Change Interrupt Request is executed from the PCI2 Interrupt Vector. PCINT[23:16] pins are enabled individually by the PCMSK2 Register.

Bit 1 – PCIE1: Pin Change Interrupt Enable 1
When the PCIE1 bit is set and the I-bit in the Status Register (SREG) is set, pin change interrupt 1 is enabled. Any change on any enabled PCINT[14:8] pin will cause an interrupt. The corresponding interrupt of Pin Change Interrupt Request is executed from the PCI1 Interrupt Vector. PCINT[14:8] pins are enabled individually by the PCMSK1 Register.

Bit 0 – PCIE0: Pin Change Interrupt Enable 0
When the PCIE0 bit is set and the I-bit in the Status Register (SREG) is set, pin change interrupt 0 is enabled. Any change on any enabled PCINT[7:0] pin will cause an interrupt. The corresponding interrupt of Pin Change Interrupt Request is executed from the PCI0 Interrupt Vector. PCINT[7:0] pins are enabled individually by the PCMSK0 Register.

Figure 16-1. 8-bit Timer/Counter Block Diagram



16.2.1. Definitions

Many register and bit references in this section are written in general form:

- n=0 represents the Timer/Counter number
- x=A,B represents the Output Compare Unit A or B

However, when using the register or bit definitions in a program, the precise form must be used, i.e., TCNT0 for accessing Timer/Counter0 counter value.

The following definitions are used throughout the section:

Table 16-1. Definitions

Constant	Description
BOTTOM	The counter reaches the BOTTOM when it becomes zero (0x00 for 8-bit counters, or 0x0000 for 16-bit counters).

16.9.4. General Timer/Counter Control Register

When addressing I/O Registers as data space using LD and ST instructions, the provided offset must be used. When using the I/O specific commands IN and OUT, the offset is reduced by 0x20, resulting in an I/O address offset within 0x00 - 0x3F.

The device is a complex microcontroller with more peripheral units than can be supported within the 64 locations reserved in Opcode for the IN and OUT instructions. For the Extended I/O space from 0x60 in SRAM, only the ST/STS/STD and LD/LDS/LDD instructions can be used.

Name: GTCCR
Offset: 0x43
Reset: 0x00
Property: When addressing as I/O Register: address offset is 0x23

Bit	7	6	5	4	3	2	1	0
	TSM						PSRASY	PSRSYNC
Access	R/W						R/W	R/W
Reset	0						0	0

Bit 7 – TSM: Timer/Counter Synchronization Mode

Writing the TSM bit to one activates the Timer/Counter Synchronization mode. In this mode, the value that is written to the PSRASY and PSRSYNC bits is kept, hence keeping the corresponding prescaler reset signals asserted. This ensures that the corresponding Timer/Counters are halted and can be configured to the same value without the risk of one of them advancing during configuration. When the TSM bit is written to zero, the PSRASY and PSRSYNC bits are cleared by hardware, and the Timer/Counters start counting simultaneously.

Bit 1 – PSRASY: Prescaler Reset Timer/Counter2

When this bit is one, the Timer/Counter2 prescaler will be reset. This bit is normally cleared immediately by hardware. If the bit is written when Timer/Counter2 is operating in asynchronous mode, the bit will remain one until the prescaler has been reset. The bit will not be cleared by hardware if the TSM bit is set.

Bit 0 – PSRSYNC: Prescaler Reset

When this bit is one, Timer/Counter1 and Timer/Counter0 prescaler will be Reset. This bit is normally cleared immediately by hardware, except if the TSM bit is set. Note that Timer/Counter1 and Timer/Counter0 share the same prescaler and a reset of this prescaler will affect both timers.

SPI Mode	Conditions	Leading Edge	Trailing Edge
2	CPOL=1, CPHA=0	Sample (Falling)	Setup (Rising)
3	CPOL=1, CPHA=1	Setup (Falling)	Sample (Rising)

The SPI data transfer formats are shown in the following figure.

Figure 20-3. SPI Transfer Format with CPHA = 0

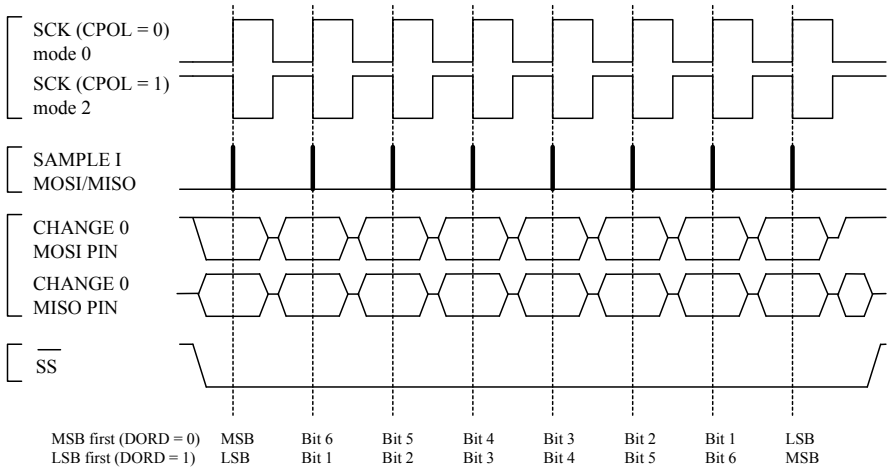
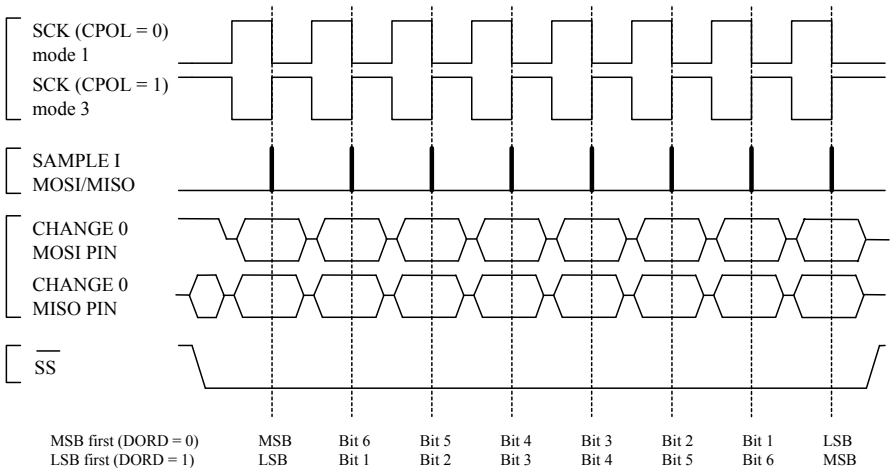


Figure 20-4. SPI Transfer Format with CPHA = 1



20.5. Register Description

TWINT clears the flag. The TWI n will then commence executing whatever operation was specified by the TWCRn setting.

The following table lists assembly and C implementation examples for TWI0. Note that the code below assumes that several definitions have been made, e.g. by using include-files.

Table 23-2. Assembly and C Code Example

	Assembly Code Example	C Example	Comments

1	`ldi r16, (1<<TWINT)|`
	`(1<<TWSTA)|(1<<TWEN)`
	`out TWCR0, r16`
		`TWCR0 = (1<<TWINT)|`
		`(1<<TWSTA)|(1<<TWEN)`
			Send START condition

2	`wait1:`
	`in r16,TWCR0`
	`sbrs r16,TWINT`
	`rjmp wait1`
		`while (!(TWCR0 &`
		`(1<<TWINT)));`
			Wait for TWINT Flag set. This indicates that the START condition has been transmitted.

3	`in r16,TWSR0`
	`andi r16, 0xF8`
	`cpi r16, START`
	`brne ERROR`
		`if ((TWSR0 & 0xF8) !=`
		`START)`
		`ERROR();`
			Check value of TWI Status Register. Mask prescaler bits. If status different from START go to ERROR.

	`ldi r16, SLA_W`
	`out TWDR0, r16`
	`ldi r16, (1<<TWINT) |`
	`(1<<TWEN)`
	`out TWCR0, r16`
		`TWDR0 = SLA_W;`
		`TWCR0 = (1<<TWINT) |`
		`(1<<TWEN);`
			Load SLA_W into TWDR Register. Clear TWINT bit in TWCR to start transmission of address.

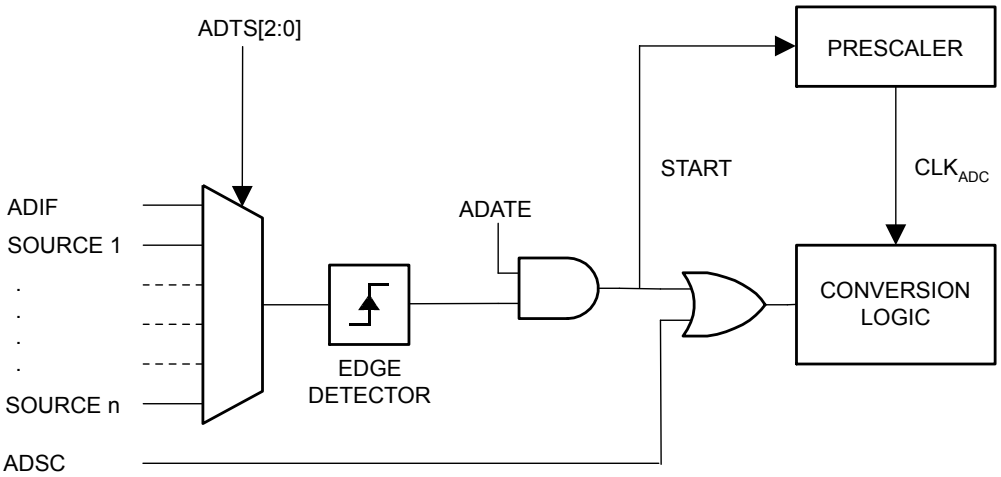
4	`wait2:`
	`in r16,TWCR0`
	`sbrs r16,TWINT`
	`rjmp wait2`
		`while (!(TWCR0 &`
		`(1<<TWINT)));`
			Wait for TWINT Flag set. This indicates that the SLA+W has been transmitted, and ACK/NACK has been received.

5	`in r16,TWSR0`
	`andi r16, 0xF8`
	`cpi r16, MT_SLA_ACK`
	`brne ERROR`
		`if ((TWSR0 & 0xF8) !=`
		`MT_SLA_ACK) ERROR();`
			Check value of TWI Status Register. Mask prescaler bits. If status different from MT_SLA_ACK go to ERROR.

	`ldi r16, DATA`
	`out TWDR0, r16`
	`ldi r16, (1<<TWINT) |`
	`(1<<TWEN)`
	`out TWCR, r16`
		`TWDR0 = DATA;`
		`TWCR0 = (1<<TWINT) |`
		`(1<<TWEN);`
			Load DATA into TWDR Register. Clear TWINT bit in TWCR to start transmission of data.

6	`wait3:`
	`in r16,TWCR0`
	`sbrs r16,TWINT`
	`rjmp wait3`
		`while (!(TWCR0 &`
		`(1<<TWINT)));`
			Wait for TWINT Flag set. This indicates that the DATA has been transmitted, and ACK/NACK has been received.

Figure 25-2. ADC Auto Trigger Logic

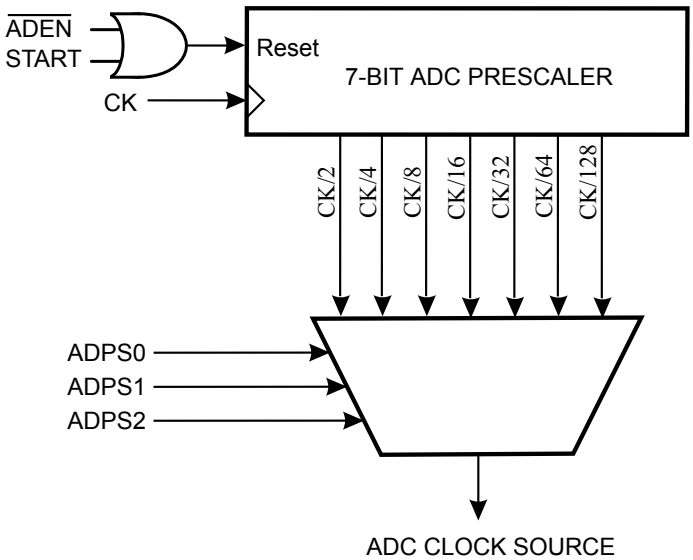


Using the ADC Interrupt Flag as a trigger source makes the ADC start a new conversion as soon as the ongoing conversion has finished. The ADC then operates in Free Running mode, constantly sampling and updating the ADC Data Register. The first conversion must be started by writing a '1' to ADCSRA.ADSC. In this mode the ADC will perform successive conversions independently of whether the ADC Interrupt Flag (ADIF) is cleared or not.

If Auto Triggering is enabled, single conversions can be started by writing ADCSRA.ADSC to '1'. ADSC can also be used to determine if a conversion is in progress. The ADSC bit will be read as '1' during a conversion, independently of how the conversion was started.

25.4. Prescaling and Conversion Timing

Figure 25-3. ADC Prescaler



By default, the successive approximation circuitry requires an input clock frequency between 50kHz and 200kHz to get maximum resolution. If a lower resolution than 10 bits is needed, the input clock frequency to the ADC can be higher than 200kHz to get a higher sample rate.

The ADC module contains a prescaler, which generates an acceptable ADC clock frequency from any CPU frequency above 100kHz. The prescaling is selected by the ADC Prescaler Select bits in the ADC Control and Status Register A (ADCSRA.ADPS). The prescaler starts counting from the moment the ADC

software may be in an undetermined state when exiting the test mode. Entering Reset, the outputs of any Port Pin will instantly enter the high impedance state, making the HIGHZ instruction redundant. If needed, the BYPASS instruction can be issued to make the shortest possible scan chain through the device. The device can be set in the Reset state either by pulling the external $\overline{\text{RESET}}$ pin low, or issuing the AVR_RESET instruction with appropriate setting of the Reset Data Register.

The EXTEST instruction is used for sampling external pins and loading output pins with data. The data from the output latch will be driven out on the pins as soon as the EXTEST instruction is loaded into the JTAG IR-register. Therefore, the SAMPLE/PRELOAD should also be used for setting initial values to the scan ring, to avoid damaging the board when issuing the EXTEST instruction for the first time. SAMPLE/PRELOAD can also be used for taking a snapshot of the external pins during normal operation of the part.

The JTAGEN fuse must be programmed and the JTD bit in the I/O register MCUCSR must be cleared to enable the JTAG Test Access Port.

When using the JTAG interface for Boundary-scan, using a JTAG TCK clock frequency higher than the internal chip frequency is possible. The chip clock is not required to run.

26.11. Data Registers

The data registers relevant for Boundary-scan operations are:

- Bypass Register
- Device Identification Register
- Reset Register
- Boundary-scan Chain

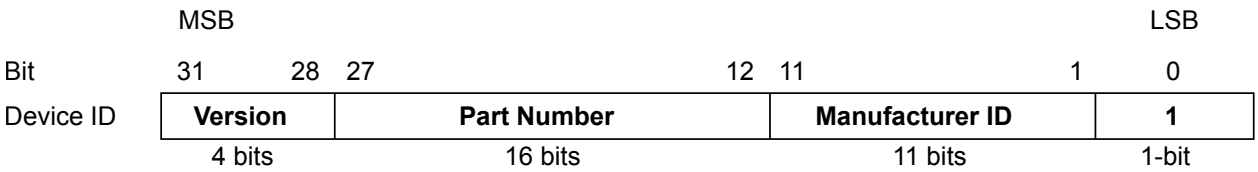
26.11.1. Bypass Register

The Bypass Register consists of a single Shift Register stage. When the Bypass Register is selected as path between TDI and TDO, the register is reset to 0 when leaving the Capture-DR controller state. The Bypass Register can be used to shorten the scan chain on a system when the other devices are to be tested.

26.11.2. Device Identification Register

The figure below shows the structure of the Device Identification Register.

Figure 26-3. The format of the Device Identification Register



26.11.2.1. Version

Version is a 4-bit number identifying the revision of the component. The JTAG version number follows the revision of the device, and wraps around at revision P (0xF). Revision A and Q is 0x0, revision B and R is 0x1 and so on.

26.11.2.2. Part Number

The part number is a 16-bit code identifying the component. The JTAG Part Number for ATmega644P is listed in the table below.

Table 26-1. AVR JTAG Part Number

Part Number	JTAG Part Number
ATmega644P	960A

26.11.2.3. Manufacturer ID

The Manufacturer ID is a 11-bit code identifying the manufacturer. The JTAG manufacturer ID for ATMEL is listed in the table below.

Table 26-2. Manufacturer ID

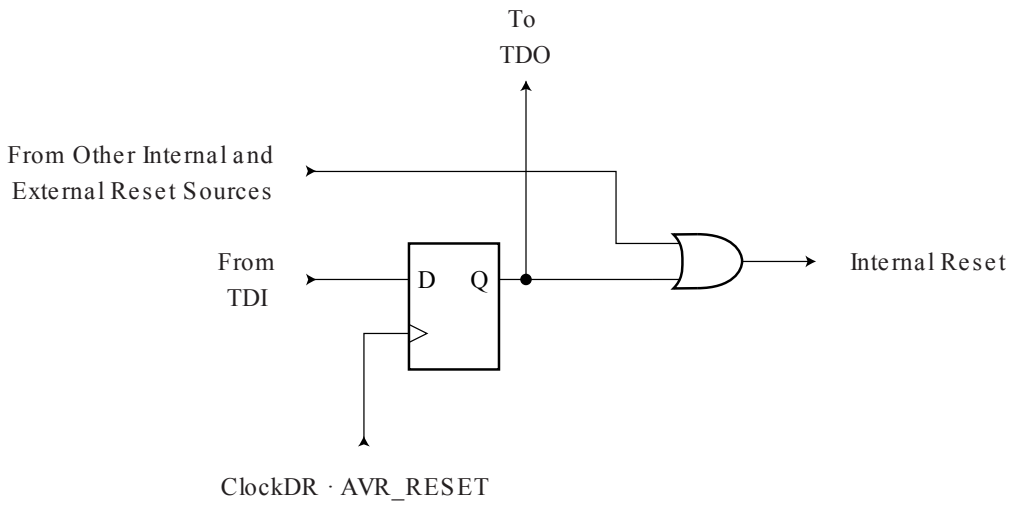
Manufacturer	JTAG Manufacturer ID (Hex)
ATMEL	0x01F

26.11.3. Reset Register

The Reset Register is a Test Data Register used to reset the part. Since the AVR tri-states Port Pins when reset, the Reset Register can also replace the function of the unimplemented optional JTAG instruction HIGHZ.

A high value in the Reset Register corresponds to pulling the External Reset low. The part is reset as long as there is a high value present in the Reset Register. Depending on the Fuse settings for the clock options, the part will remain reset for a Reset Time-Out Period (refer to *Clock Sources*) after releasing the Reset Register. The output from this Data Register is not latched, so the Reset will take place immediately, as shown in the figure below.

Figure 26-4. Reset Register



26.11.4. Boundary-scan Chain

The Boundary-scan Chain has the capability of driving and observing the logic levels on the digital I/O pins, as well as the boundary between digital and analog logic for analog circuitry having off-chip connections. Refer to Boundary-scan Chain for a complete description.

26.12. Boundry-scan Specific JTAG Instructions

The Instruction Register is 4-bit wide, supporting up to 16 instructions. Listed below are the JTAG instructions useful for Boundary-scan operation. Note that the optional HIGHZ instruction is not

Instruction	TDI sequence	TDO sequence	Notes
6c. Write Fuse Extended byte	0111011_00000000 0111001_00000000 0111011_00000000 0111011_00000000	xxxxxxx_xxxxxxxx xxxxxxx_xxxxxxxx xxxxxxx_xxxxxxxx xxxxxxx_xxxxxxxx	(1)
6d. Poll for Fuse Write complete	0110111_00000000	xxxxxox_xxxxxxxx	(2)
6e. Load Data Low Byte[7]	0010011_iiiiiiii	xxxxxxx_xxxxxxxx	(3)
6f. Write Fuse High byte	0110111_00000000 0110101_00000000 0110111_00000000 0110111_00000000	xxxxxxx_xxxxxxxx xxxxxxx_xxxxxxxx xxxxxxx_xxxxxxxx xxxxxxx_xxxxxxxx	(1)
6g. Poll for Fuse Write complete	0110111_00000000	xxxxxox_xxxxxxxx	(2)
6h. Load Data Low Byte[7]	0010011_iiiiiiii	xxxxxxx_xxxxxxxx	(3)
6i. Write Fuse Low byte	0110011_00000000 0110001_00000000 0110011_00000000 0110011_00000000	xxxxxxx_xxxxxxxx xxxxxxx_xxxxxxxx xxxxxxx_xxxxxxxx xxxxxxx_xxxxxxxx	(1)
6j. Poll for Fuse Write complete	0110011_00000000	xxxxxox_xxxxxxxx	(2)
7a. Enter Lock bit Write	0100011_00100000	xxxxxxx_xxxxxxxx	
7b. Load Data Byte[9]	0010011_11iiiiii	xxxxxxx_xxxxxxxx	(4)
7c. Write Lock bits	0110011_00000000 0110001_00000000 0110011_00000000 0110011_00000000	xxxxxxx_xxxxxxxx xxxxxxx_xxxxxxxx xxxxxxx_xxxxxxxx xxxxxxx_xxxxxxxx	(1)
7d. Poll for Lock bit Write complete	0110011_00000000	xxxxxox_xxxxxxxx	(2)
8a. Enter Fuse/Lock bit Read	0100011_00000100	xxxxxxx_xxxxxxxx	
8b. Read Extended Fuse Byte[6]	0111010_00000000 0111011_00000000	xxxxxxx_xxxxxxxx xxxxxxx_oooooooo	
8c. Read Fuse High Byte[7]	0111110_00000000 0111111_00000000	xxxxxxx_xxxxxxxx xxxxxxx_oooooooo	
8d. Read Fuse Low Byte[8]	0110010_00000000 0110011_00000000	xxxxxxx_xxxxxxxx xxxxxxx_oooooooo	
8e. Read Lock bits[9]	0110110_00000000 0110111_00000000	xxxxxxx_xxxxxxxx xxxxxxx_xxoooooo	(5)

Figure 29-4. SPI Interface Timing Requirements (Master Mode)

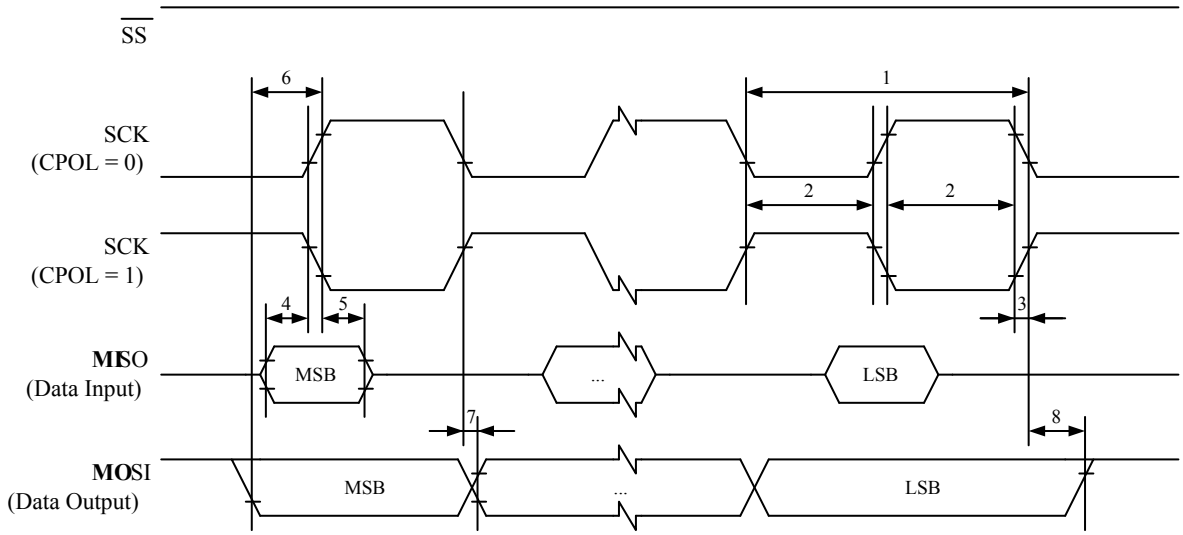
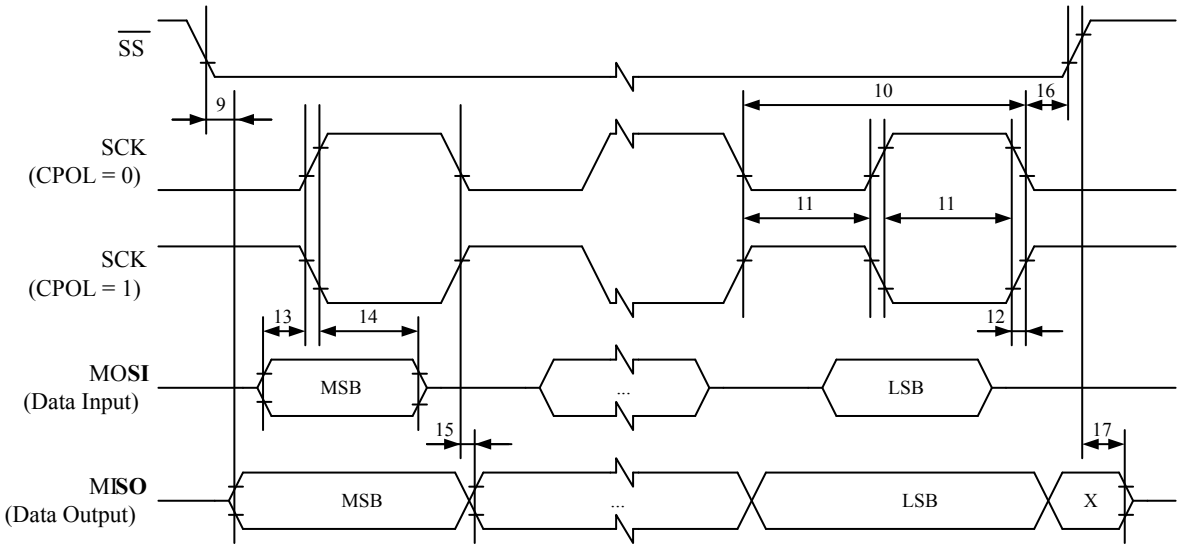


Figure 29-5. SPI Interface Timing Requirements (Slave Mode)



29.8. Two-wire Serial Interface Characteristics

The table in this section describes the requirements for devices connected to the 2-wire Serial Bus. The 2-wire Serial Interface meets or exceeds these requirements under the noted conditions.

The timing symbols refers to Figure 29-6.

Table 29-11. Two-wire Serial Bus Requirements

Symbol	Parameter	Condition	Min.	Max	Units
V_{IL}	Input Low-voltage		-0.5	0.3 V_{CC}	V
V_{IH}	Input High-voltage		0.7 V_{CC}	V_{CC} + 0.5	V
V_{hys}(1)	Hysteresis of Schmitt Trigger Inputs		0.05 V_{CC}(2)	–	V

Figure 30-7. Idle Supply Current vs. Frequency (1 - 20MHz)

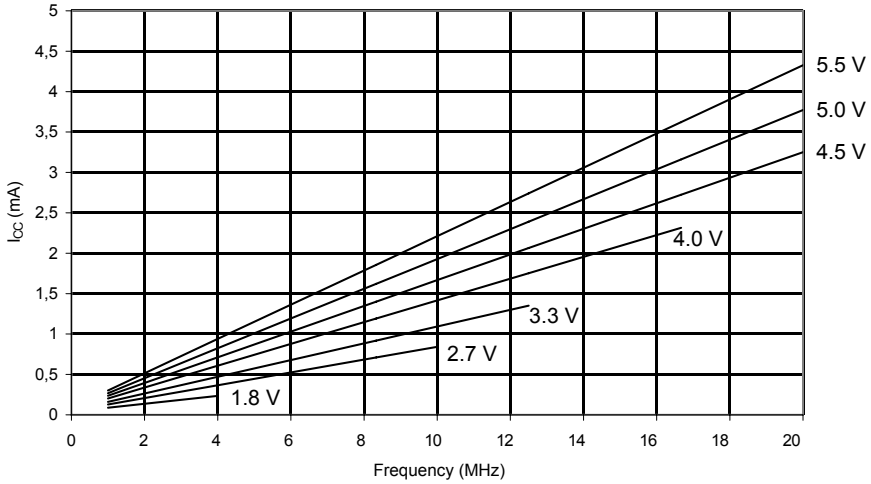


Figure 30-8. Idle Supply Current vs. V_{CC} (Internal RC Oscillator, 8 MHz)

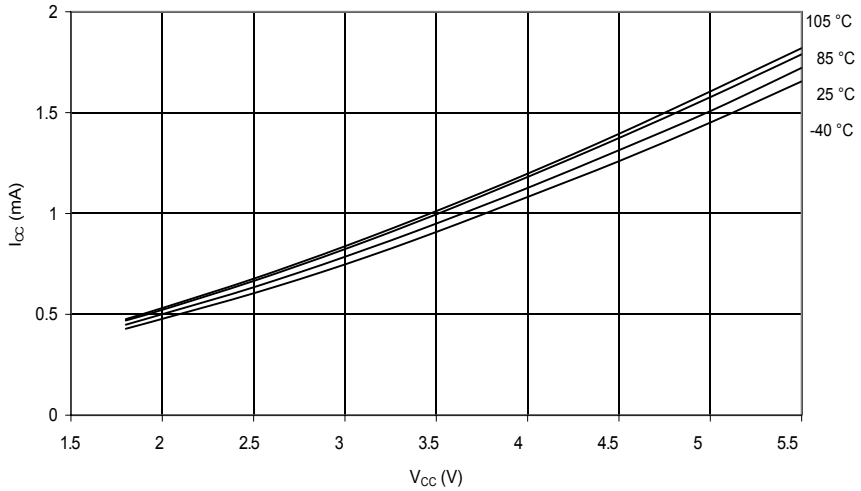


Figure 30-9. Idle Supply Current vs. V_{CC} (Internal RC Oscillator, 1 MHz)

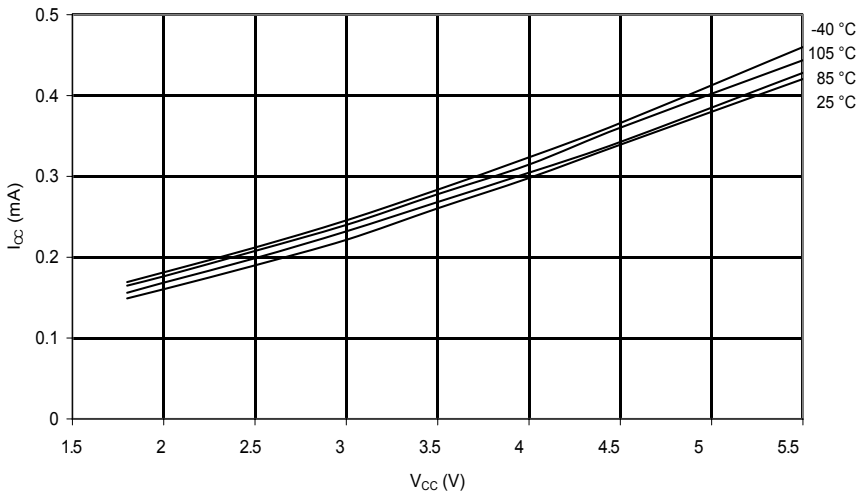


Figure 30-23. I/O Pin Output Voltage vs. Source Current (V_{CC} = 3V)

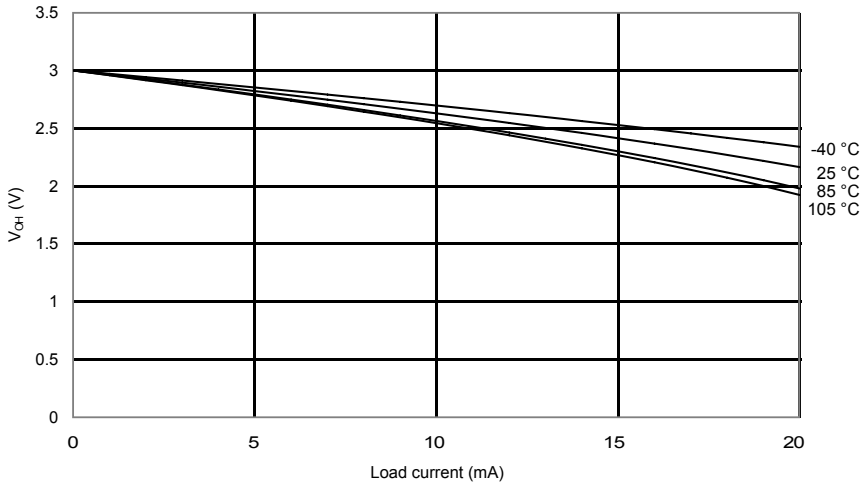


Figure 30-24. I/O Pin Output Voltage vs. Source Current (V_{CC} = 5V)

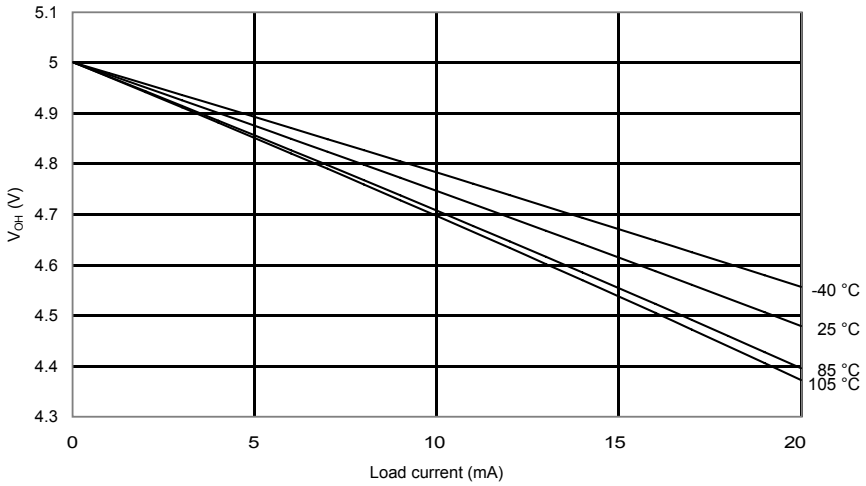


Figure 30-41. AREF External Reference Current vs. V_{CC}

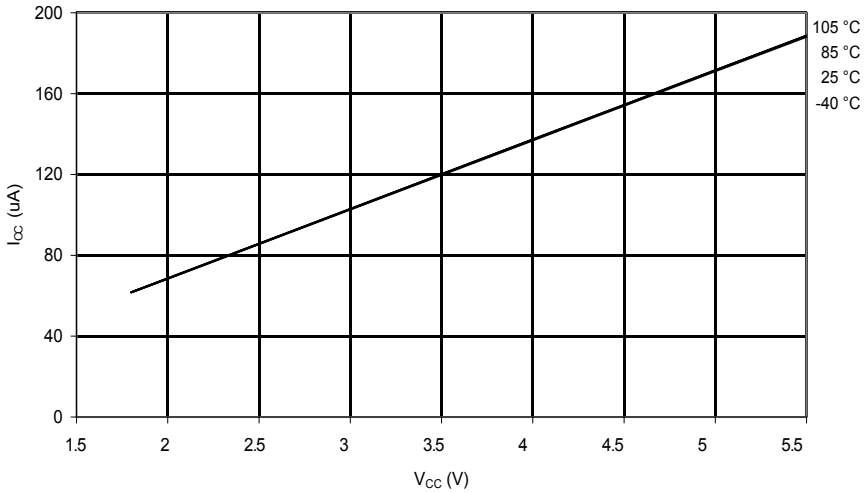


Figure 30-42. Brownout Detector Current vs. V_{CC}

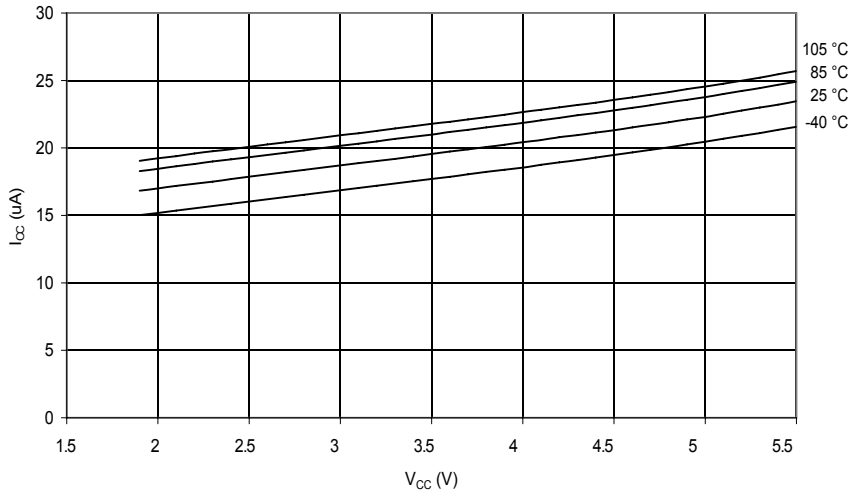


Figure 30-43. Programming Current vs. V_{CC}