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	8051
Core Size	8-Bit
Speed	20MHz
Connectivity	SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	14
Program Memory Size	2KB (2K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	2.4V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	16-TSSOP (0.173", 4.40mm Width)
Supplier Device Package	16-TSSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at89lp216-20xu

Table 9-2. CLKREG – Clock Control Register

CLKREG = 8FH — Reset Value = 0000 0000B

Not Bit Addressable

	TPS3	TPS2	TPS1	TPS0	–	CDV1	CDV0	COE
Bit	7	6	5	4	3	2	1	0

Symbol	Function
TPS3 TPS2 TPS1 TPS0	Timer Prescaler. The Timer Prescaler selects the time base for Timer 0, Timer 1 and the Watchdog Timer. The prescaler is implemented as a 4-bit binary down counter. When the counter reaches zero it is reloaded with the value stored in the TPS bits to give a division ratio between 1 and 16. By default the timers will count every clock cycles (TPS = 0000B). To configure the timers to count at a standard 8051 rate of once every 12 clock cycles, TPS should be set to 1011B.
CDV1 CDV0	Clock Out Division. Determines the frequency of the clock output relative to the system clock. CDIV1 CDIV0 Clock Out Frequency 0 0 f/2 0 1 f/4 1 0 f/8 1 1 f/16
COE	Clock Out Enable. Set COE to output a divided version of the system clock on XTAL2 (P3.3). The internal RC oscillator or external clock source must be selected in order to use this feature.

10. Reset

During reset, all I/O Registers are set to their initial values, the port pins are tristated, and the program starts execution from the Reset Vector, 0000H. The AT89LP216 has five sources of reset: power-on reset, brown-out reset, external reset, watchdog reset, and software reset.

10.1 Power-on Reset

A Power-on Reset (POR) is generated by an on-chip detection circuit. The detection level is nominally 1.4V. The POR is activated whenever V_{CC} is below the detection level. The POR circuit can be used to trigger the start-up reset or to detect a supply voltage failure in devices without a brown-out detector. The POR circuit ensures that the device is reset from power-on. A power-on sequence is shown in Figure 10-1. When V_{CC} reaches the Power-on Reset threshold voltage V_{POR}, an initialization sequence lasting t_{POR} is started. When the initialization sequence completes, the start-up timer determines how long the device is kept in POR after V_{CC} rise. The POR signal is activated again, without any delay, when V_{CC} falls below the POR threshold level. A Power-on Reset (i.e. a cold reset) will set the POF flag in PCON. The internally generated reset can be extended beyond the power-on period by holding the $\overline{RST}$ pin low longer than the time-out.

Table 12-4. IPH – Interrupt Priority High Register

IPH = B7H — Reset Value = X000 0000B

Not Bit Addressable

	–	–	PGH	PSH	PT1H	PX1H	PT0H	PX0H
Bit	7	6	5	4	3	2	1	0

Symbol	Function
PGH	General-purpose Interrupt Priority High
PSH	Serial Port Interrupt Priority High
PT1H	Timer 1 Interrupt Priority High
PX1H	External Interrupt 1 Priority High
PT0H	Timer 0 Interrupt Priority High
PX0H	External Interrupt 0 Priority High

13. I/O Ports

The AT89LP216 can be configured for between 11 and 14 I/O pins. The exact number of I/O pins available depends on the clock and reset options as shown in Table 13-1. All port pins are 5V tolerant as inputs, that is they can be pulled up or driven to 5.5V even when operating at a lower V_{CC} such as 3V. **Inputs use CMOS** levels while **outputs use TTL** levels. An external pull-up is required to convert outputs to CMOS levels.

Table 13-1. I/O Pin Configurations

Clock Source	Reset Option	Number of I/O Pins
External Crystal or Resonator	External $\overline{RST}$ Pin	11
	No external reset	12
External Clock	External $\overline{RST}$ Pin	12
	No external reset	13
Internal RC Oscillator	External $\overline{RST}$ Pin	13
	No external reset	14

13.1 Port Configuration

All port pins on the AT89LP216 may be configured to one of four modes: quasi-bidirectional (standard 8051 port outputs), push-pull output, open-drain output, or input-only. Port modes may be assigned in software on a pin-by-pin basis as shown in Table 13-2. The Tristate-Port User Fuse determines the default state of the port pins. When the fuse is enabled, all port pins default to input-only mode after reset, with the exception of P1.4 which starts in quasi-bidirectional mode. When the fuse is disabled, all port pins, with the exception of P1.0 and P1.1, default to quasi-bidirectional mode after reset and are weakly pulled high. Each port pin also has a Schmitt-triggered input for improved input noise rejection. During Power-down all the Schmitt-triggered inputs are disabled with the exception of P1.3, P3.2 and P3.3, which may be used to wake up the device. Therefore P1.3, P3.2 and P3.3 should not be left floating during Power-down.

Table 14-2. TMOD: Timer/Counter Mode Control Register

TMOD = 89H — Reset Value = 0000 0000B

Not Bit Addressable

GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0
7	$\overline{6}$	5	4	3	$\overline{2}$	1	0
Timer1				**Timer0**			

Gate	Gating control when set. Timer/Counter x is enabled only while $\overline{INTx}$ pin is high and TRx control pin is set. When cleared, Timer x is enabled whenever TRx control bit is set.	Timer 0 gate bit
C/$\overline{T}$	Timer or Counter Selector cleared for Timer operation (input from internal system clock). Set for Counter operation (input from Tx input pin).	Timer 0 counter/timer select bit
M1	Timer 1 Mode bit 1	Timer 0 M1 bit
M0	Timer 1 Mode bit 0	Timer 0 M0 bit

M1	M0	Mode	Operating Mode
0	0	0	Variable 9 - 16-bit Timer mode. 8-bit Timer/Counter THx with TLx as 1 - 8-bit prescaler.
0	1	1	16-bit auto-reload mode. 16-bit Timer/Counters THx and TLx are cascaded; there is no prescaler.
1	0	2	8-bit auto reload. 8-bit auto-reload Timer/Counter THx holds a value which is to be reloaded into TLx each time it overflows.
1	1	3	Split Timer mode. (Timer 0) TL0 is an 8-bit Timer/Counter controlled by the standard Timer 0 control bits. TH0 is an 8-bit timer only controlled by Timer 1 control bits.
1	1	3	(Timer 1) Timer/Counter 1 stopped.

Timer SFR	Purpose	Address	Bit-Addressable
TCON	Control	88H	Yes
TMOD	Mode	89H	No
TL0	Timer 0 low-byte	8AH	No
TL1	Timer 1 low-byte	8BH	No
TH0	Timer 0 high-byte	8CH	No
TH1	Timer 1 high-byte	8DH	No
TCONB	Mode	91H	No
RL0	Timer 0 reload low-byte	92H	No
RL1	Timer 1 reload low-byte	93H	No
RH0	Timer 0 reload high-byte	94H	No
RH1	Timer 1 reload high-byte	95H	No

Figure 14-7. Timer/Counter 1 PWM Mode 1

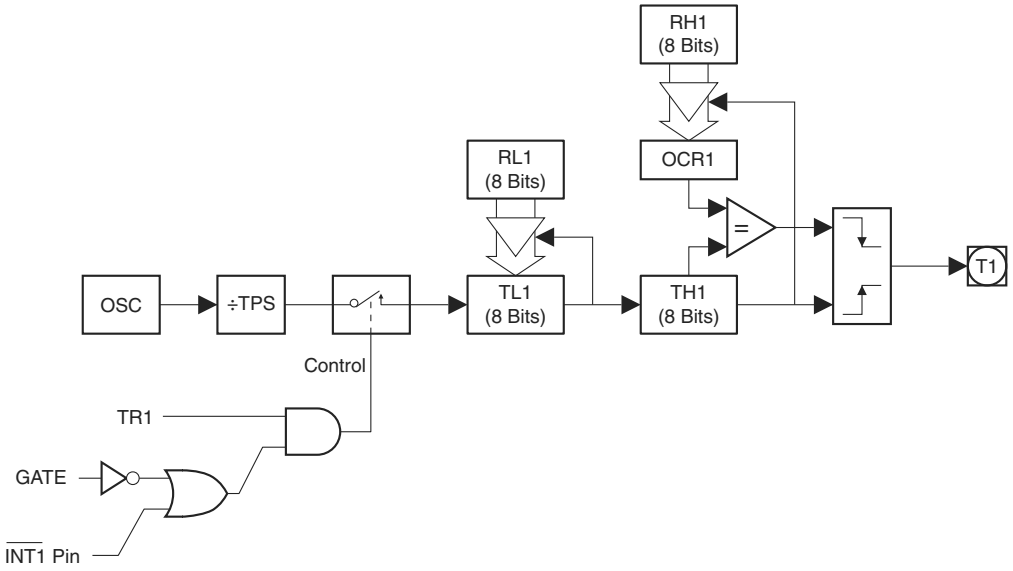
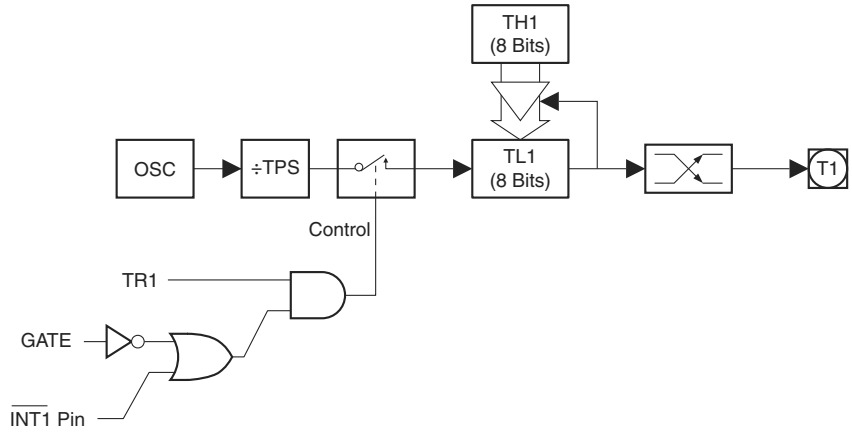
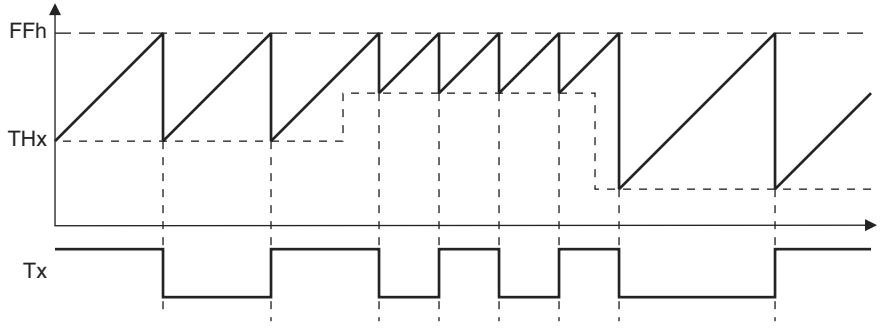


Figure 14-8. Timer/Counter 1 PWM Mode 2



Note: {RH0 & RL0}/{RH1 & RL1} are not required by Timer 0/Timer 1 during PWM Mode 2 and may be used as temporary storage registers.

Figure 14-9. PWM Mode 2 Waveform



interrupted by a data byte. An address byte, however, interrupts all slaves. Each slave can examine the received byte and see if it is being addressed. The addressed slave clears its SM2 bit and prepares to receive the data bytes that follows. The slaves that are not addressed set their SM2 bits and ignore the data bytes. See “Automatic Address Recognition” on page 48.

The SM2 bit has no effect in Mode 0 but can be used to check the validity of the stop bit in Mode 1. In a Mode 1 reception, if SM2 = 1, the receive interrupt is not activated unless a valid stop bit is received.

Table 17-1. SCON – Serial Port Control Register

SCON Address = 98H — Reset Value = 0000 0000B

Bit Addressable

Bit	7	6	5	4	3	2	1	0
	SM0/FE	SM1	SM2	REN	TB8	RB8	T1	RI

(SMOD0 = 0/1)[1]

Symbol	Function
FE	Framing error bit. This bit is set by the receiver when an invalid stop bit is detected. The FE bit is not cleared by valid frames and must be cleared by software. The SMOD0 bit must be set to enable access to the FE bit. FE will be set regardless of the state of SMOD0.
SM0	Serial Port Mode Bit 0, (SMOD0 must = 0 to access bit SM0)
SM1	Serial Port Mode Bit 1 **SM0** / **SM1** / **Mode** / **Description** / **Baud Rate**[2] 0 / 0 / 0 / shift register / $f_{osc}/2$ 0 / 1 / 1 / 8-bit UART / variable (Timer 1) 1 / 0 / 2 / 9-bit UART / $f_{osc}/32$ or $f_{osc}/16$ 1 / 1 / 3 / 9-bit UART / variable (Timer 1)
SM2	Enables the Automatic Address Recognition feature in Modes 2 or 3. If SM2 = 1 then RI will not be set unless the received 9th data bit (RB8) is 1, indicating an address, and the received byte is a Given or Broadcast Address. In Mode 1, if SM2 = 1 then RI will not be activated unless a valid stop bit was received, and the received byte is a Given or Broadcast Address. In Mode 0, SM2 should be 0.
REN	Enables serial reception. Set by software to enable reception. Clear by software to disable reception.
TB8	The 9th data bit that will be transmitted in Modes 2 and 3. Set or clear by software as desired.
RB8	In Modes 2 and 3, the 9th data bit that was received. In Mode 1, if SM2 = 0, RB8 is the stop bit that was received. In Mode 0, RB8 is not used.
TI	Transmit interrupt flag. Set by hardware at the end of the 8th bit time in Mode 0, or at the beginning of the stop bit in the other modes, in any serial transmission. Must be cleared by software.
RI	Receive interrupt flag. Set by hardware at the end of the 8th bit time in Mode 0, or halfway through the stop bit time in the other modes, in any serial reception (except see SM2). Must be cleared by software.

Notes: 1. SMOD0 is located at PCON.6.

2. f_{osc} = oscillator frequency.

18.2 More About Mode 1

Ten bits are transmitted (through TXD), or received (through RXD): a start bit (0), 8 data bits (LSB first), and a stop bit (1). On receive, the stop bit goes into RB8 in SCON. In the AT89LP216, the baud rate is determined by the Timer 1 overflow rate. Figure 18-2 shows a simplified functional diagram of the serial port in Mode 1 and associated timings for transmit and receive.

Transmission is initiated by any instruction that uses SBUF as a destination register. The “write to SBUF” signal also loads a “1” into the 9th bit position of the transmit shift register and flags the TX Control unit that a transmission is requested. Transmission actually commences at S1P1 of the machine cycle following the next rollover in the divide-by-16 counter. Thus, the bit times are synchronized to the divide-by-16 counter, not to the “write to SBUF” signal.

The transmission begins when $\overline{\text{SEND}}$ is activated, which puts the start bit at TXD. One bit time later, DATA is activated, which enables the output bit of the transmit shift register to TXD. The first shift pulse occurs one bit time after that.

As data bits shift out to the right, “0”s are clocked in from the left. When the MSB of the data byte is at the output position of the shift register, the “1” that was initially loaded into the 9th position is just to the left of the MSB, and all positions to the left of that contain “0”s. This condition flags the TX Control unit to do one last shift, then deactivate $\overline{\text{SEND}}$ and set TI. This occurs at the tenth divide-by-16 rollover after “write to SBUF.”

Reception is initiated by a 1-to-0 transition detected at RXD. For this purpose, RXD is sampled at a rate of 16 times the established baud rate. When a transition is detected, the divide-by-16 counter is immediately reset, and 1FFH is written into the input shift register. Resetting the divide-by-16 counter aligns its roll-overs with the boundaries of the incoming bit times.

The 16 states of the counter divide each bit time into 16ths. At the 7th, 8th, and 9th counter states of each bit time, the bit detector samples the value of RXD. The value accepted is the value that was seen in at least 2 of the 3 samples. This is done to reject noise. In order to reject false bits, if the value accepted during the first bit time is not 0, the receive circuits are reset and the unit continues looking for another l-to-0 transition. If the start bit is valid, it is shifted into the input shift register, and reception of the rest of the frame proceeds.

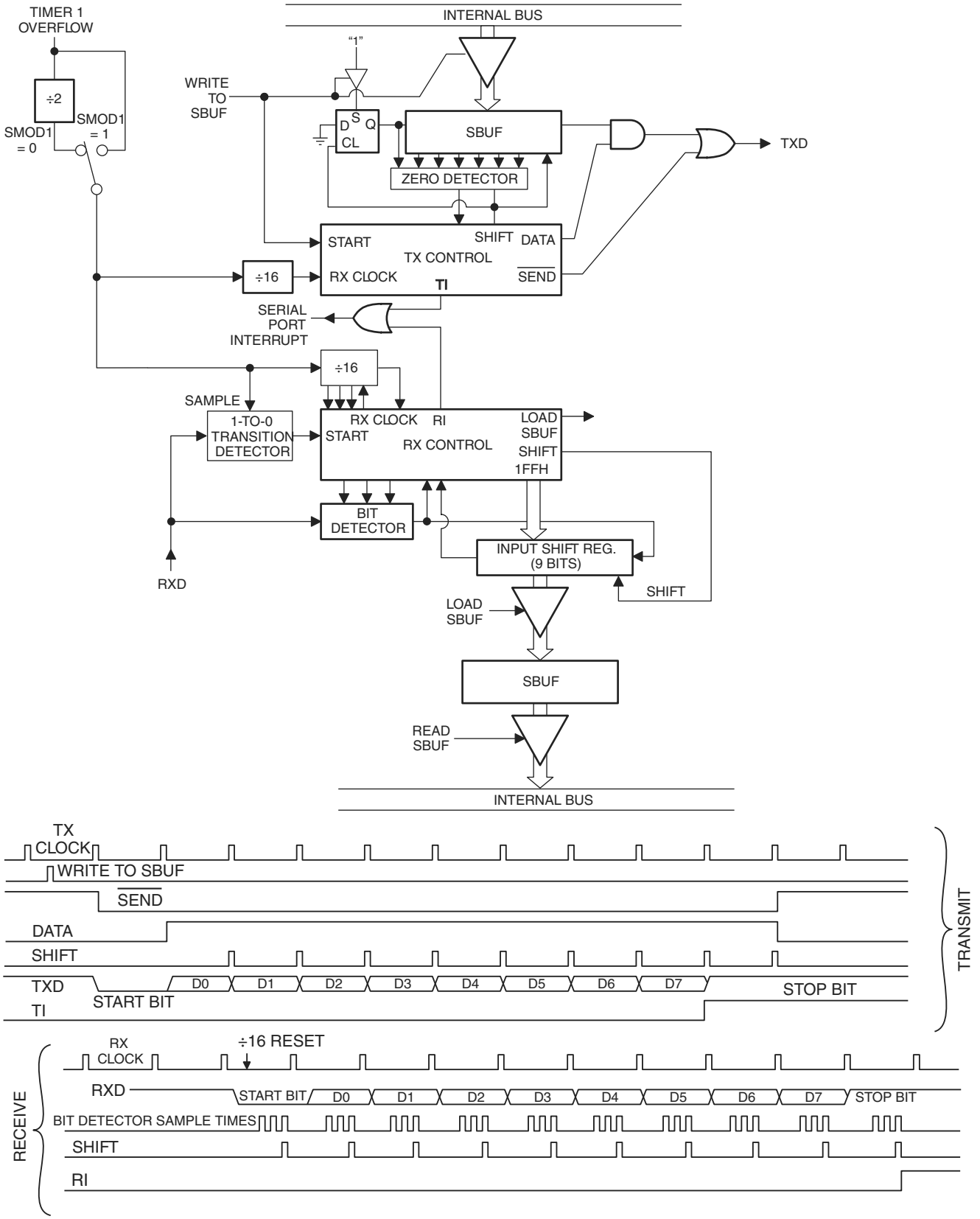
As data bits come in from the right, “1”s shift out to the left. When the start bit arrives at the left-most position in the shift register, (which is a 9-bit register in Mode 1), it flags the RX Control block to do one last shift, load SBUF and RB8, and set RI. The signal to load SBUF and RB8 and to set RI is generated if, and only if, the following conditions are met at the time the final shift pulse is generated.

RI = 0 and

Either SM2 = 0, or the received stop bit = 1

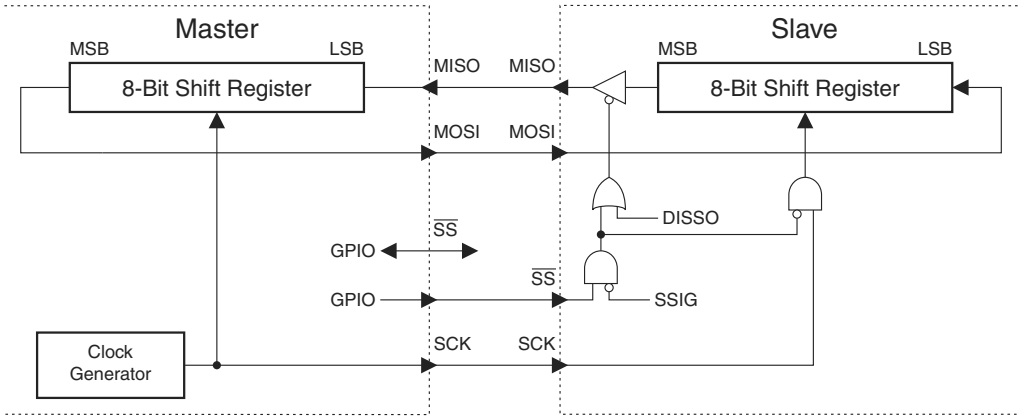
If either of these two conditions is not met, the received frame is irretrievably lost. If both conditions are met, the stop bit goes into RB8, the 8 data bits go into SBUF, and RI is activated. At this time, whether or not the above conditions are met, the unit continues looking for a 1-to-0 transition in RXD.

Figure 18-2. Serial Port Mode 1



The interconnection between master and slave CPUs with SPI is shown in Figure 19-1. The four pins in the interface are Master-In/Slave-Out (MISO), Master-Out/Slave-In (MOSI), Shift Clock (SCK), and Slave Select ($\overline{SS}$). The SCK pin is the clock output in master mode, but is the clock input in slave mode. The MSTR bit in SPCR determines the directions of MISO and MOSI. Also notice that MOSI connects to MOSI and MISO to MISO. In master mode, $\overline{SS}$/P1.4 is ignored and may be used as a general-purpose input or output. In slave mode, $\overline{SS}$ must be driven low to select an individual device as a slave. When $\overline{SS}$ is driven high, the slave's SPI port is deactivated and the MOSI/P1.5 pin can be used as a general-purpose input.

Figure 19-1. SPI Master-slave Interconnection



The SPI has two modes of operation: normal (non-buffered write) and enhanced (buffered write). In normal mode, writing to the SPI data register (SPDR) of the master CPU starts the SPI clock generator and the data written shifts out of the MOSI pin and into the MOSI pin of the slave CPU. Transmission may start after an initial delay while the clock generator waits for the next full bit slot of the specified baud rate. After shifting one byte, the SPI clock generator stops, setting the end of transmission flag (SPIF) and transferring the received byte to the read buffer (SPDR). If both the SPI interrupt enable bit (SPIE) and the serial port interrupt enable bit (ES) are set, an interrupt is requested. Note that SPDR refers to either the write data buffer or the read data buffer, depending on whether the access is a write or read. In normal mode, because the write buffer is transparent (and a write access to SPDR will be directed to the shift buffer), any attempt to write to SPDR while a transmission is in progress will result in a write collision with WCOL set. However, the transmission will still complete normally, but the new byte will be ignored and a new write access to SPDR will be necessary.

Enhanced mode is similar to normal mode except that the write buffer holds the next byte to be transmitted. Writing to SPDR loads the write buffer and sets WCOL to signify that the buffer is full and any further writes will overwrite the buffer. WCOL is cleared by hardware when the buffered byte is loaded into the shift register and transmission begins. If the master SPI is currently idle, i.e. if this is the first byte, then after loading SPDR, transmission of the byte starts and WCOL is cleared immediately. While this byte is transmitting, the next byte may be written to SPDR. The Load Enable flag (LDEN) in SPSR can be used to determine when transmission has started. LDEN is asserted during the first four bit slots of a SPI transfer. The master CPU should first check that LDEN is set and that WCOL is cleared before loading the next byte. In enhanced mode, if WCOL is set when a transfer completes, i.e. the next byte is available, then the SPI immediately loads the buffered byte into the shift register, resets WCOL, and continues transmission without stopping and restarting the clock generator. As long as the CPU can keep the write buffer full in this manner, multiple bytes may be transferred with minimal latency between bytes.

Figure 19-2. SPI Shift Register Diagram

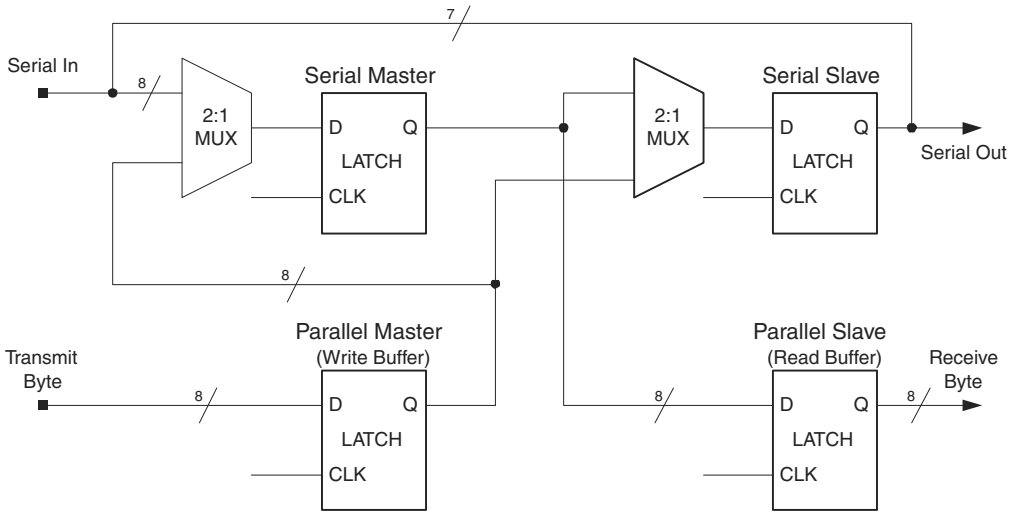
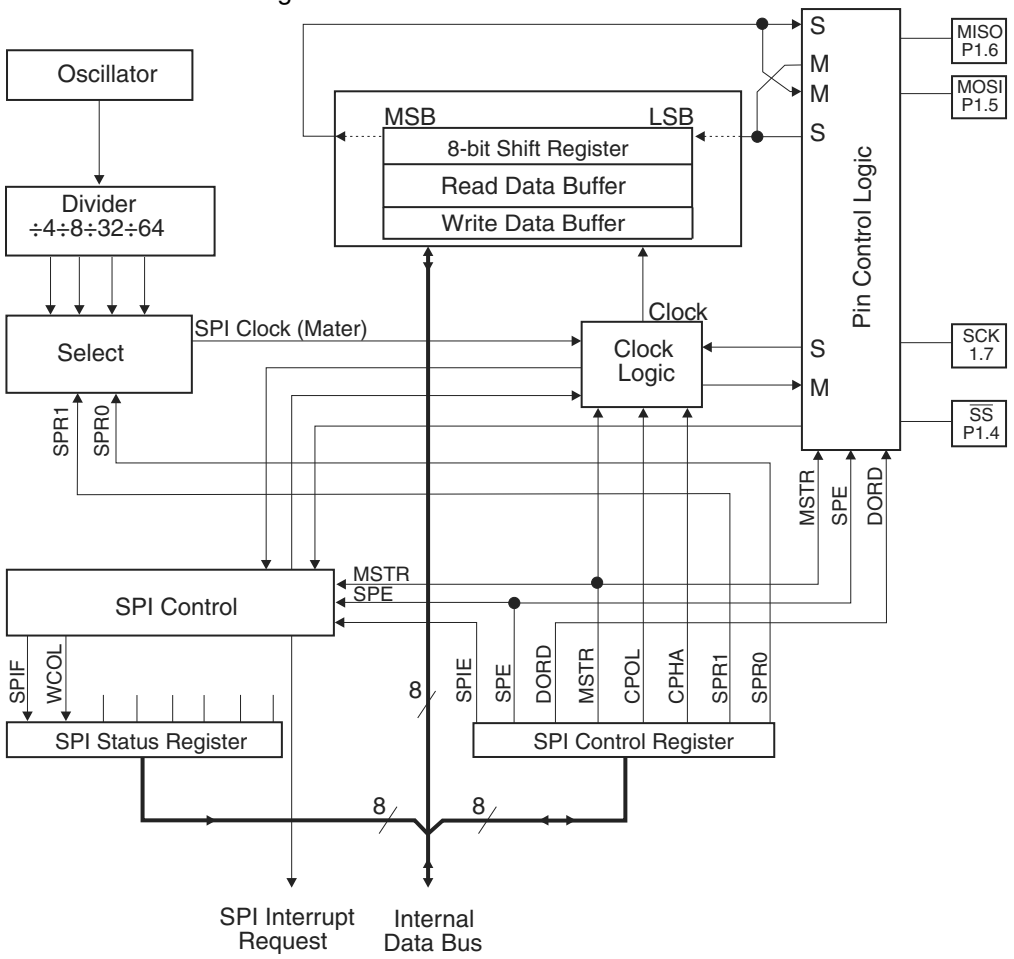


Figure 19-3. SPI Block Diagram



20. Analog Comparator

A single analog comparator is provided on the AT89LP216. The analog comparator has the following features:

- Comparator Output Flag and Interrupt
- Selectable Interrupt Condition
  - High- or Low-level
  - Rising- or Falling-edge
  - Output Toggle
- Hardware Debouncing Modes

Comparator operation is such that the output is a logic “1” when the positive input AIN0 (P1.0]) is greater than the negative input AIN1 (P1.1). Otherwise the output is a zero. Setting the CEN bit in ACSR enables the comparator. When the comparator is first enabled, the comparator output and interrupt flag are not guaranteed to be stable for 10 µs. The corresponding comparator interrupt should not be enabled during that time, and the comparator interrupt flag must be cleared before the interrupt is enabled in order to prevent an immediate interrupt service. Before enabling the comparator the analog inputs should be tristated by putting P1.0 and P1.1 into input-only mode. See “Port 1 Analog Functions” on page 25.

The comparator may be configured to cause an interrupt under a variety of output value conditions by setting the CM bits in ACSR. The comparator interrupt flag CF in ACSR is set whenever the comparator output matches the condition specified by CM. The flag may be polled by software or may be used to generate an interrupt and must be cleared by software.

20.1 Comparator Interrupt with Debouncing

The comparator output is sampled every clock cycle. The conditions on the analog inputs may be such that the comparator output will toggle excessively. This is especially true if applying slow moving analog inputs. Three debouncing modes are provided to filter out this noise. In debouncing mode, the comparator uses Timer 1 to modulate its sampling time. When a relevant transition occurs, the comparator waits until two Timer 1 overflows have occurred before resampling the output. If the new sample agrees with the expected value, CF is set. Otherwise, the event is ignored. The filter may be tuned by adjusting the time-out period of Timer 1. Because Timer 1 is free running, the debouncer must wait for two overflows to guarantee that the sampling delay is at least 1 time-out period. Therefore, after the initial edge event, the interrupt may occur between 1 and 2 time-out periods later. See Figure 20-1 on page 56.

By default the comparator is disabled during Idle mode. To allow the comparator to function during Idle, the CIDL bit is ACSR must be set. When CIDL is set, the comparator can be used to wake-up the CPU from Idle if the comparator interrupt is enabled. The comparator is always disabled during Power-down mode.

Table 22-1. Instruction Execution Times and Exceptions (Continued)

Data Transfer	Bytes	Clock Cycles		Hex Code
		8051	AT89LP	
MOV A, Rn	1	12	1	E8-EF
MOV A, direct	2	12	2	E5
MOV A, @Ri	1	12	2	E6-E7
MOV A, #data	2	12	2	74
MOV Rn, A	1	12	1	F8-FF
MOV Rn, direct	2	24	2	A8-AF
MOV Rn, #data	2	12	2	78-7F
MOV direct, A	2	12	2	F5
MOV direct, Rn	2	24	2	88-8F
MOV direct, direct	3	24	3	85
MOV direct, @Ri	2	24	2	86-87
MOV direct, #data	3	24	3	75
MOV @Ri, A	1	12	1	F6-F7
MOV @Ri, direct	2	24	2	A6-A7
MOV @Ri, #data	2	12	2	76-77
MOV DPTR, #data16	3	24	3	90
MOVC A, @A+DPTR	1	24	3	93
MOVC A, @A+PC	1	24	3	83
MOVX A, @Ri	1	24	4	E2-E3
MOVX A, @DPTR	1	24	4	E0
MOVX @Ri, A	1	24	4	F2-F3
MOVX @DPTR, A	1	24	4	F0
PUSH direct	2	24	2	C0
POP direct	2	24	2	D0
XCH A, Rn	1	12	1	C8-CF
XCH A, direct	2	12	2	C5
XCH A, @Ri	1	12	2	C6-C7
XCHD A, @Ri	1	12	2	D6-D7

23.2 Software Breakpoints

The AT89LP216 microcontroller includes a BREAK instruction for implementing program memory breakpoints in software. A software breakpoint can be inserted manually by placing the BREAK instruction in the program code. Some emulator systems may allow for automatic insertion/deletion of software breakpoints. The Flash memory must be re-programmed each time a software breakpoint is changed. Frequent insertions/deletions of software breakpoints will reduce the data retention of the nonvolatile memory. Devices used for debugging purposes should not be shipped to end customers. The BREAK instruction is treated as a two-cycle NOP when OCD is disabled.

23.3 Limitations of On-chip Debug

The AT89LP216 is a low-cost, low-pincount yet fully-featured microcontroller that multiplexes several functions on its limited I/O pins. Some device functionality must be sacrificed to provide resources for On-chip Debugging. The On-chip Debug System has the following limitations:

- The Debug Clock pin (DCL) is physically located on that same pin as Port Pin P1.3 and the External Reset ($\overline{RST}$). Therefore, neither P1.3 nor an external reset source may be emulated when OCD is enabled.
- The Debug Data pin (DDA) is physically located on either the XTAL1/P3.2 or XTAL2/P3.3 pin. The crystal oscillator is therefore not supported during debug. The user must select either the Internal RC Oscillator or the External Clock source to provide the system clock. Devices fused for the crystal oscillator will default to external clock mode when OCD is enabled.
- When using the Internal RC Oscillator during debug, DDA is located on the XTAL1/P3.2 pin. The $\overline{INT0}$ function cannot be emulated in this mode.
- When using the External Clock during debug, DDA is located on the XTAL2/P3.3 pin and the system clock drives XTAL1/P3.2. The $\overline{INT0}$, $\overline{INT1}$ and CLKOUT functions cannot be emulated in this mode.
- The AT89LP216 does not support In-Application Programming and therefore the device must be reset before changing the program code during debugging. This includes the insertion/deletion of software breakpoints.
- When using the watchdog to generate a break, the state of the watchdog will not be reset. An OCD reset command should be sent to the device prior to resuming normal execution to ensure correct watchdog behavior.

24. Programming the Flash Memory

The Atmel AT89LP216 microcontroller features 2KB of on-chip In-System Programmable Flash program memory. In-System Programming (ISP) allows programming and reprogramming of the microcontroller positioned inside the end system. Using a simple 4-wire SPI interface, the In-System programmer communicates serially with the AT89LP216 microcontroller, reprogramming all nonvolatile memories on the chip. In-System programming eliminates the need for physical removal of the chips from the system. This will save time and money, both during development in the lab, and when updating the software or parameters in the field. The ISP interface of the AT89LP216 includes the following features:

- Four Wire SPI Programming Interface
- Active-low Reset Entry into Programming
- Slave Select allows multiple devices on same interface
- User Signature Array

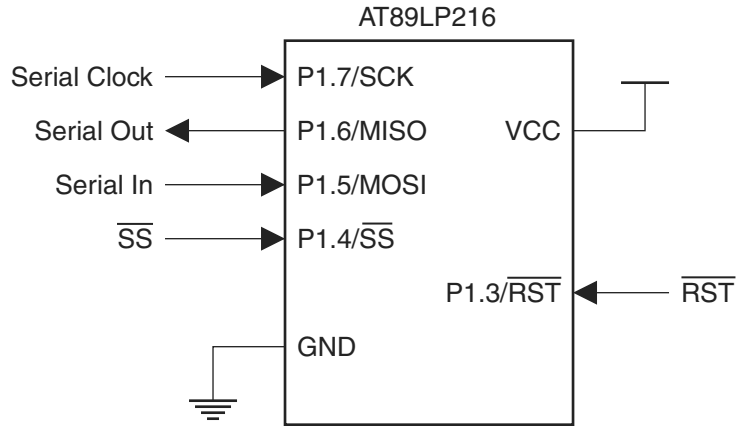
- Flexible Page Programming
- Row Erase Capability
- Page Write with Auto-Erase Commands
- Programming Status Register

For more detailed information on In-System Programming, refer to the Application Note entitled “AT89LP In-System Programming Specification”.

24.1 Physical Interface

In-System Programming utilizes the Serial Peripheral Interface (SPI) pins of an AT89LP216 microcontroller. The SPI is a full duplex synchronous serial interface consisting of four wires: Serial Clock (SCK), Master-In/Slave-out (MISO), Master-out/Slave-in (MOSI), and an active-low Slave Select ($\overline{SS}$). When programming an AT89LP216 device, the programmer always operates as the SPI master, and the target system always operates as the SPI slave. To enter or remain in In-System Programming mode the device’s reset line ($\overline{RST}$) must be held active (low). With the addition of VCC and GND, an AT89LP216 microcontroller can be programmed with a minimum of seven connections as shown in Figure 24-1.

Figure 24-1. In-System Programming Device Connections



The In-System Programming Interface is the only means of externally programming the AT89LP216 microcontroller. The ISP Interface can be used to program the device both in-system and in a stand-alone serial programmer. The ISP Interface does not require any clock other than SCK and is not limited by the system clock frequency. During In-System programming the system clock source of the target device can operate normally.

When designing a system where In-System Programming will be used, the following observations must be considered for correct operation:

- The ISP interface uses the SPI clock mode 0 (CPOL = 0,CPHA = 0) exclusively with a maximum frequency of 5 MHz.
- The AT89LP216 will enter programming mode only when its reset line ($\overline{RST}$) is active (low). To simplify this operation, it is recommended that the target reset can be controlled by the In-System programmer. To avoid problems, the In-System programmer should be able to keep the entire target system reset for the duration of the programming cycle. The target system should never attempt to drive the four SPI lines while reset is active.

- The $\overline{RST}$ input may be disabled to gain an extra I/O pin. In these cases the $\overline{RST}$ pin will always function as a reset during power up. To enter programming the $\overline{RST}$ pin must be driven low prior to the end of Power-On Reset (POR). After POR has completed the device will remain in ISP mode until $\overline{RST}$ is brought high. Once the initial ISP session has ended, the power to the target device must be cycled OFF and ON to enter another session.
- The $\overline{SS}$ pin should not be left floating during reset if ISP is enabled.
- The ISP Enable Fuse must be set to allow programming during any reset period. If the ISP Fuse is disabled, ISP may only be entered at POR.

24.2 Memory Organization

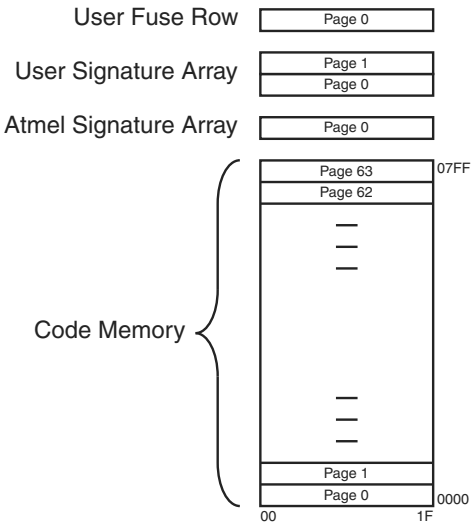
The AT89LP216 offers 2K bytes of In-System Programmable (ISP) nonvolatile Flash code memory. In addition, the device contains a 64-byte User Signature Array and a 32-byte read-only Atmel Signature Array. The memory organization is shown in Table 24-1 and Figure 24-2. The memory is divided into pages of 32 bytes each. A single read or write command may only access a single page in the memory. Each memory type resides in its own address space and is accessed by commands specific to that memory. However, all memory types share the same page size.

User configuration fuses are mapped as a row in the memory, with each byte representing one fuse. From a programming standpoint, fuses are treated the same as normal code bytes except they are not affected by Chip Erase. Fuses can be enabled at any time by writing 00h to the appropriate locations in the fuse row. However, to disable a fuse, i.e. set it to FFh, the **entire** fuse row must be erased and then reprogrammed. The programmer should read the state of all the fuses into a temporary location, modify those fuses which need to be disabled, then issue a Fuse Write with Auto-Erase command using the temporary data. Lock bits are treated in a similar manner to fuses except they may only be erased (unlocked) by Chip Erase.

Table 24-1. Code Memory Sizes

Device #	Code Size	Page Size	# Pages	Address Range
AT89LP216	2K bytes	32 bytes	64	0000H - 07FFH

Figure 24-2. AT89LP216 Memory Organization



24.3 Command Format

Programming commands consist of an opcode byte, two address bytes, and zero or more data bytes. In addition, all command packets must start with a two-byte preamble of AAH and 55H. The preamble increases the noise immunity of the programming interface by making it more difficult to issue unintentional commands. Figure 24-3 on page 68 shows a simplified flow chart of a command sequence.

A sample command packet is shown in Figure 24-4 on page 68. The $\overline{SS}$ pin defines the packet frame. $\overline{SS}$ must be brought low before the first byte in a command is sent and brought back high after the final byte in the command has been sent. The command is not complete until $\overline{SS}$ returns high. Command bytes are issued serially on MOSI. Data output bytes are received serially on MISO. Packets of variable length are supported by returning $\overline{SS}$ high when the final required byte has been transmitted. In some cases command bytes have a don't care value. Don't care bytes in the middle of a packet must be transmitted. Don't care bytes at the end of a packet may be ignored.

Page oriented instructions always include a full 16-bit address. The higher order bits select the page and the lower order bits select the byte within that page. The AT89LP216 allocates 5 bits for byte address and 6 bits for page address. The page to be accessed is always fixed by the page address as transmitted. The byte address specifies the starting address for the first data byte. After each data byte has been transmitted, the byte address is incremented to point to the next data byte. This allows a page command to linearly sweep the bytes within a page. If the byte address is incremented past the last byte in the page, the byte address will roll over to the first byte in the same page. While loading bytes into the page buffer, overwriting previously loaded bytes will result in data corruption.

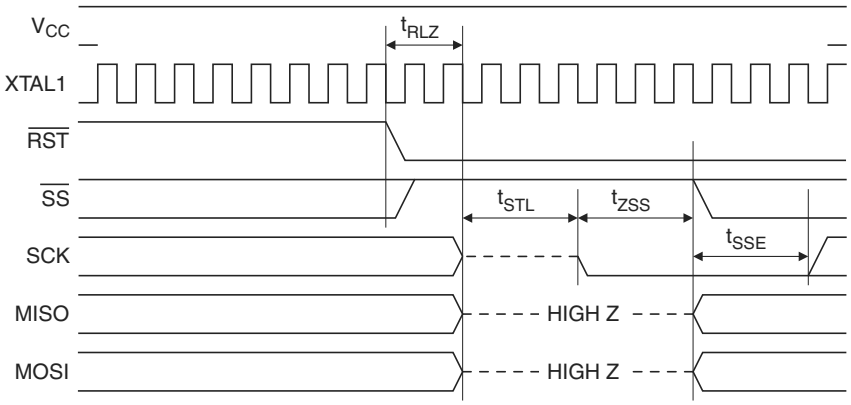
For a summary of available commands, see Table 24-2 on page 69.

24.8.3 ISP Start Sequence

Execute this sequence to exit CPU execution mode and enter ISP mode when the device has passed Power-on Reset and is already operational.

1. Drive $\overline{RST}$ low.
2. Drive $\overline{SS}$ high.
3. Wait $t_{RLZ} + t_{STL}$.
4. Start programming session.

Figure 24-7. In-System Programming (ISP) Start Sequence

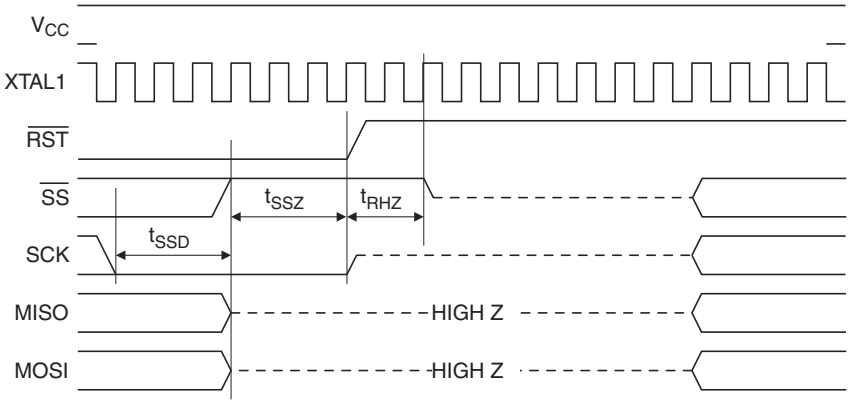


24.8.4 ISP Exit Sequence

Execute this sequence to exit ISP mode and resume CPU execution mode.

1. Drive SCK low.
1. Wait at least t_{SSD} and drive $\overline{SS}$ high.
2. Tristate MOSI.
3. Wait at least t_{SSZ} and bring $\overline{RST}$ high.
4. Tristate SCK.
5. Wait t_{RHZ} and tristate $\overline{SS}$.

Figure 24-8. In-System Programming (ISP) Exit Sequence



Note: The waveforms on this page are not to scale.

24.8.6 Timing Parameters

The timing parameters for Figure 24-5, Figure 24-6, Figure 24-7, Figure 24-8, and Figure 24-10 are shown in Table .

Table 24-6. Programming Interface Timing Parameters

Symbol	Parameter	Min	Max	Units
t_{CLCL}	System Clock Cycle Time	0	60	ns
t_{PWUP}	Power On to $\overline{SS}$ High Time	10		µs
t_{POR}	Power-on Reset Time		100	µs
t_{PWDN}	$\overline{SS}$ Tristate to Power Off		1	µs
t_{RLZ}	$\overline{RST}$ Low to I/O Tristate	t_{CLCL}	2 t_{CLCL}	ns
t_{STL}	$\overline{RST}$ Low Settling Time	100		ns
t_{RHZ}	$\overline{RST}$ High to $\overline{SS}$ Tristate	0	2 t_{CLCL}	ns
t_{SCK}	Serial Clock Cycle Time	200[1]		ns
t_{SHSL}	Clock High Time	75		ns
t_{SLSH}	Clock Low Time	50		ns
t_{SR}	Rise Time		25	ns
t_{SF}	Fall Time		25	ns
t_{SIS}	Serial Input Setup Time	10		ns
t_{SIH}	Serial Input Hold Time	10		ns
t_{SOH}	Serial Output Hold Time		10	ns
t_{SOV}	Serial Output Valid Time		35	ns
t_{SOE}	Output Enable Time		10	ns
t_{SOX}	Output Disable Time		25	ns
t_{SSE}	$\overline{SS}$ Enable Lead Time	t_{SLSH}		ns
t_{SSD}	$\overline{SS}$ Disable Lag Time	t_{SLSH}		ns
t_{ZSS}	SCK Setup to $\overline{SS}$ Low	25		ns
t_{SSZ}	SCK Hold after $\overline{SS}$ High	25		ns
t_{WW}	Write Cycle Time	2.5		ms
t_{AWW}	Write Cycle with Auto-Erase Time	5		ms
t_{ERS}	Chip Erase Cycle Time	7.5		ms

Note: 1. t_{SCK} is independent of t_{CLCL}.

25.3.3 Internal Oscillator Frequency

Figure 25-5. Internal Oscillator Frequency vs. VCC

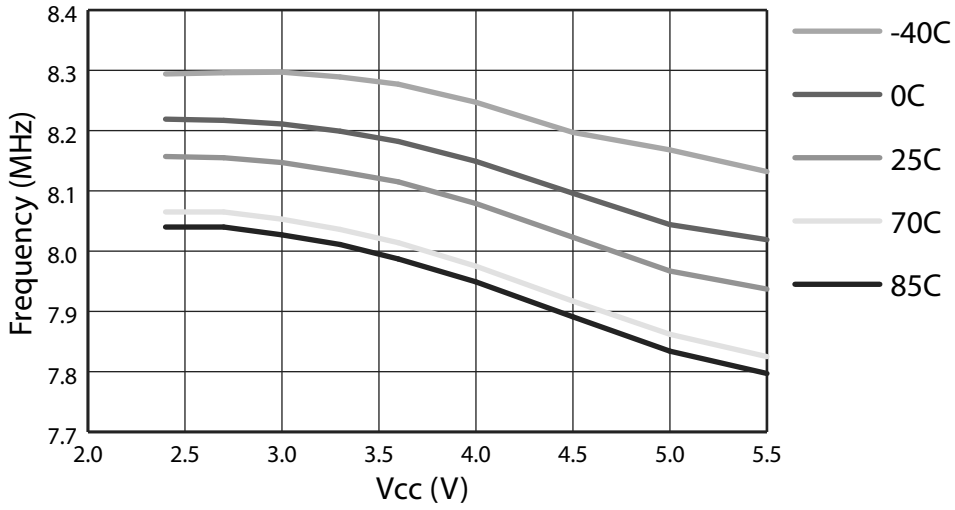


Table 25-1. Typical Internal Oscillator Behavior

Symbol	Parameter	Condition	Min	Max	Units
Df_{IRC}	Relative Frequency Error (MAX–MIN) / (MAX+MIN)	T_A = -40–85°C; V_{CC} = 2.4–5.5V		±4	%
		T_A = -40–85°C; V_{CC} = 4.5–5.5V		±3	%
		T_A = -40–85°C; V_{CC} = 2.4–3.6V		±3	%
		T_A = 0–70°C; V_{CC} = 2.4–5.5V		±3	%
		T_A = 0–70°C; V_{CC} = 4.5–5.5V		±2	%
		T_A = 0–70°C; V_{CC} = 2.4–3.6V		±2	%
f_{IRC}	Frequency (Calibrated at 25°C; 5V)	T_A = -40–85°C; V_{CC} = 2.4–5.5V	7.8	8.4	MHz
		T_A = -40–85°C; V_{CC} = 4.5–5.5V	7.8	8.2	MHz
		T_A = -40–85°C; V_{CC} = 2.4–3.6V	8.0	8.4	MHz
		T_A = 0–70°C; V_{CC} = 2.4–5.5V	7.8	8.3	MHz
		T_A = 0–70°C; V_{CC} = 4.5–5.5V	7.8	8.1	MHz
		T_A = 0–70°C; V_{CC} = 2.4–3.6V	8.0	8.3	MHz

Note: The data in this table was characterized on factory calibrated devices. It is not tested during manufacturing and is provided for reference only. Devices may need to be recalibrated to target different operating conditions than the single-point factory calibration.

25.5 Serial Peripheral Interface Timing

Table 25-4. SPI **Master** Characteristics

Symbol	Parameter	Min	Max	Units
t_{CLCL}	Oscillator Period	41.6		ns
t_{SCK}	Serial Clock Cycle Time	$4t_{CLCL}$		ns
t_{SHSL}	Clock High Time	$t_{SCK}/2 - 25$		ns
t_{SLSH}	Clock Low Time	$t_{SCK}/2 - 25$		ns
t_{SR}	Rise Time		25	ns
t_{SF}	Fall Time		25	ns
t_{SIS}	Serial Input Setup Time	10		ns
t_{SIH}	Serial Input Hold Time	10		ns
t_{SOH}	Serial Output Hold Time		10	ns
t_{SOV}	Serial Output Valid Time		35	ns

Table 25-5. SPI **Slave** Characteristics

Symbol	Parameter	Min	Max	Units
t_{CLCL}	Oscillator Period	41.6		ns
t_{SCK}	Serial Clock Cycle Time	$4t_{CLCL}$		ns
t_{SHSL}	Clock High Time	$1.5\ t_{CLCL} - 25$		ns
t_{SLSH}	Clock Low Time	$1.5\ t_{CLCL} - 25$		ns
t_{SR}	Rise Time		25	ns
t_{SF}	Fall Time		25	ns
t_{SIS}	Serial Input Setup Time	10		ns
t_{SIH}	Serial Input Hold Time	10		ns
t_{SOH}	Serial Output Hold Time		10	ns
t_{SOV}	Serial Output Valid Time		35	ns
t_{SOE}	Output Enable Time		10	ns
t_{SOX}	Output Disable Time		25	ns
t_{SSE}	Slave Enable Lead Time	10		ns
t_{SSD}	Slave Disable Lag Time	0		ns

Table of Contents (Continued)

21 Programmable Watchdog Timer 57

21.1 Software Reset 58

22 Instruction Set Summary 59

23 On-chip Debug System 63

23.1 Physical Interface 63

23.2 Software Breakpoints 64

23.3 Limitations of On-chip Debug 64

24 Programming the Flash Memory 64

24.1 Physical Interface 65

24.2 Memory Organization 66

24.3 Command Format 67

24.4 Status Register 70

24.5 DATA Polling 70

24.6 Flash Security 70

24.7 User Configuration Fuses 71

24.8 Programming Interface Timing 72

25 Electrical Characteristics 76

25.1 Absolute Maximum Ratings* 76

25.2 DC Characteristics 76

25.3 Typical Characteristics 77

25.4 Clock Characteristics 83

25.5 Serial Peripheral Interface Timing 84

25.6 Serial Port Timing: Shift Register Mode Test Conditions 87

25.7 Test Conditions 87

26 Ordering Information 89

26.1 Green Package Option (Pb/Halide-free) 89

27 Packaging Information 90

27.1 16P3 – PDIP 90

27.2 16S – SOIC 91

27.3 16X – TSSOP 92

28 Revision History 93