# THE DATASHEET OF

# AT89LP216-20PU

www.win-source.net 0086-755-83957316

## Features

- 8-bit Microcontroller Compatible with MCS®51 Products
- Enhanced 8051 Architecture
  - Single Clock Cycle per Byte Fetch
  - Up to 20 MIPS Throughput at 20 MHz Clock Frequency
  - Fully Static Operation: 0 Hz to 20 MHz
  - On-chip 2-cycle Hardware Multiplier
  - 128 x 8 Internal RAM
  - 4-level Interrupt Priority
- Nonvolatile Program Memory
  - 2K Bytes of In-System Programmable (ISP) Flash Memory
  - Endurance: Minimum 10,000 Write/Erase Cycles
  - Serial Interface for Program Downloading
  - 32-byte Fast Page Programming Mode
  - 64-byte User Signature Array
  - 2-level Program Memory Lock for Software Security
- Peripheral Features
  - Two 16-bit Enhanced Timer/Counters
  - Two 8-bit PWM Outputs
  - Enhanced UART with Automatic Address Recognition and Framing Error Detection
  - Enhanced Master/Slave SPI with Double-buffered Send/Receive
  - Programmable Watchdog Timer with Software Reset
  - Analog Comparator with Selectable Interrupt and Debouncing
  - 8 General-purpose Interrupt Pins
- Special Microcontroller Features
  - Two-wire On-chip Debug Interface
  - Brown-out Detection and Power-on Reset with Power-off Flag
  - Internal 8 MHz RC Oscillator
  - Low Power Idle and Power-down Modes
  - Interrupt Recovery from Power-down Mode
- I/O and Packages
  - Up to 14 Programmable I/O Lines
  - Configurable I/O with Quasi-bidirectional, Input, Push-pull Output, and Open-drain Modes
  - 5V Tolerant I/O
  - 16-lead TSSOP, SOIC or PDIP
- Operating Conditions
  - 2.4V to 5.5V $V_{CC}$ Voltage Range
  - -40· C to 85°C Temperature Range

# 8-bit Microcontroller with 2K Bytes Flash

# AT89LP216

## 1. Description

The AT89LP216 is a low-power, high-performance CMOS 8-bit microcontroller with 2K bytes of In-System Programmable Flash memory. The device is manufactured using Atmel®'s high-density nonvolatile memory technology and is compatible with the industry-standard MCS-51 instruction set. The AT89LP216 is built around an enhanced CPU core that can fetch a single byte from memory every clock cycle. In the classic 8051 architecture, each fetch requires 6 clock cycles, forcing instructions to execute in 12, 24 or 48 clock cycles. In the AT89LP216 CPU, instructions

3621E–MICRO–11/10

need only 1 to 4 clock cycles providing 6 to 12 times more throughput than the standard 8051. Seventy percent of instructions need only as many clock cycles as they have bytes to execute, and most of the remaining instructions require only one additional clock. The enhanced CPU core is capable of 20 MIPS throughput whereas the classic 8051 CPU can deliver only 4 MIPS at the same current consumption. Conversely, at the same throughput as the classic 8051, the new CPU core runs at a much lower speed and thereby greatly reduces power consumption.

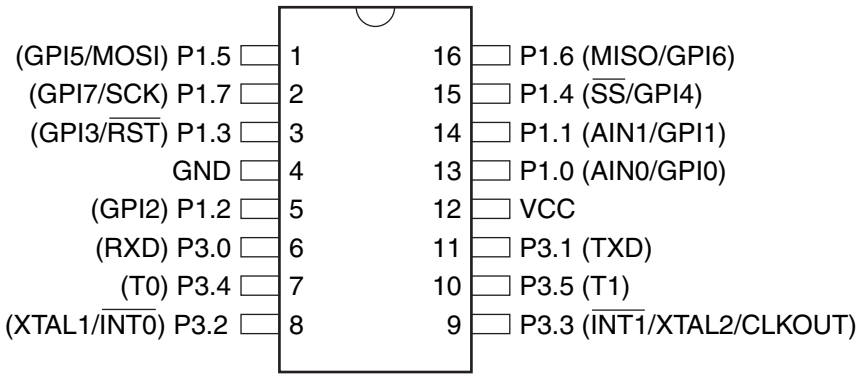
The AT89LP216 provides the following standard features: 2K bytes of In-System Programmable Flash memory, 128 bytes of RAM, up to 14 I/O lines, two 16-bit timer/counters, two PWM outputs, a programmable watchdog timer, a full duplex serial port, a serial peripheral interface, an internal 8 MHz RC oscillator, on-chip crystal oscillator, and a four-level, six-vector interrupt system.

The two timer/counters in the AT89LP216 are enhanced with two new modes. Mode 0 can be configured as a variable 9- to 16-bit timer/counter and Mode 1 can be configured as a 16-bit auto-reload timer/counter. In addition, the timer/counters may independently drive a pulse width modulation output.

The I/O ports of the AT89LP216 can be independently configured in one of four operating modes. In quasi-bidirectional mode, the ports operate as in the classic 8051. In input mode, the ports are tristated. Push-pull output mode provides full CMOS drivers and open-drain mode provides just a pull-down. In addition, all 8 pins of Port 1 can be configured to generate an interrupt using the general-purpose interrupt interface. The I/O pins of the AT89LP216 tolerate voltages higher than the device's own power supply, up to 5.5V. When the device is supplied at 2.4V and all I/O ports receive 5.5V, the total back flowing current in all I/Os is less than 100 µA.

# 2. Pin Configuration

## 2.1 16-lead PDIP/SOIC/TSSOP



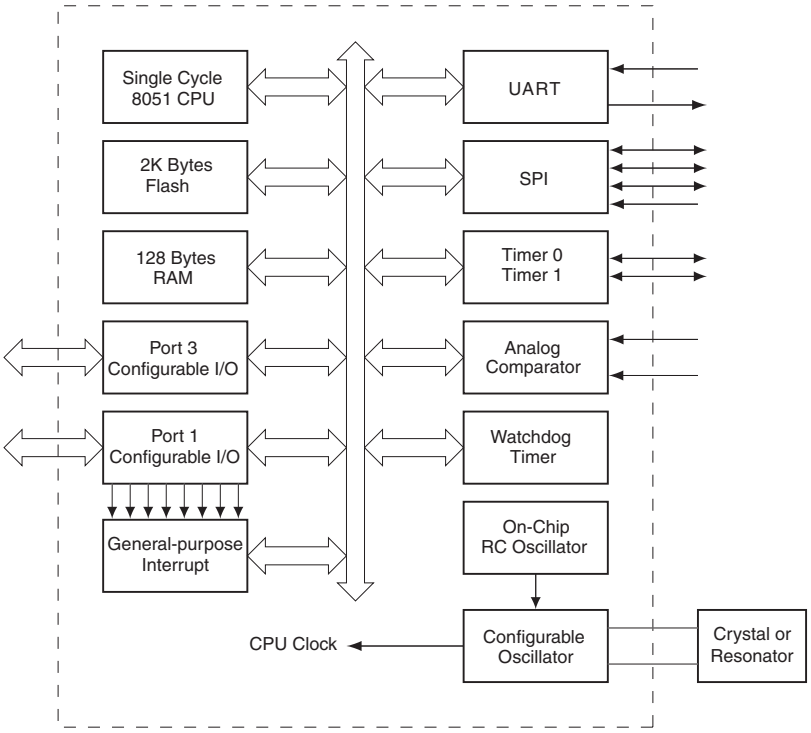
# 3. Pin Description

**Table 3-1.** Pin Description

| Pin | Symbol | Type | Description |
|---|---|---|---|
| 1 | P1.5 | I/O<br>I/O<br><br>I | **P1.5**: User-configurable I/O Port 1 bit 5.<br>**MOSI**: SPI master-out/slave-in. When configured as master, this pin is an output. When configured as slave, this pin is an input.<br>**GPI5**: General-purpose Interrupt input 5. |
| 2 | P1.7 | I/O<br>I/O<br><br>I | **P1.7**: User-configurable I/O Port 1 bit 7.<br>**SCK**: SPI Clock. When configured as master, this pin is an output. When configured as slave, this pin is an input.<br>**GPI7**: General-purpose Interrupt input 7. |
| 3 | P1.3 | I/O<br>I<br>I<br>I | **P1.3**: User-configurable I/O Port 1 bit 3 (if Reset Fuse is disabled).<br>**$\overline{\text{RST}}$**: External **Active-Low** Reset input (if Reset Fuse is enabled, see Section 10.3 “External Reset” on page 15)<br>**GPI3**: General-purpose Interrupt input 3.<br>**DCL**: Serial Clock input for On-Chip Debug Interface when OCD is enabled. |
| 4 | GND | I | Ground |
| 5 | P1.2 | I/O<br>I | **P1.2**: User-configurable I/O Port 1 bit 2.<br>**GPI2**: General-purpose Interrupt input 2. |
| 6 | P3.0 | I/O<br>I | **P3.0**: User-configurable I/O Port 3 bit 0.<br>**RXD**: Serial Port Receiver Input. |
| 7 | P3.4 | I/O<br>I/O | **P3.4**: User-configurable I/O Port 3 bit 4.<br>**T0**: Timer/Counter 0 External Input or PWM Output. |
| 8 | P3.2 | I/O<br>I<br><br>I/O | **P3.2**: User-configurable I/O Port 3 bit 2.<br>**XTAL1**: Input to the inverting oscillator amplifier and internal clock generation circuits. It may be used as a port pin if the internal RC oscillator is selected as the clock source.<br>**DDA**: Serial Data input/output for On-Chip Debug Interface when OCD is enabled and the internal RC oscillator is selected as the clock source. |
| 9 | P3.3 | I/O<br>O<br><br>O<br><br>I/O | **P3.3**: User-configurable I/O Port 3 bit 3.<br>**XTAL2**: Output from inverting oscillator amplifier. It may be used as a port pin if the internal RC oscillator is selected as the clock source.<br>**CLKOUT**: When the internal RC oscillator is selected as the clock source, may be used to output the internal clock divided by 2.<br>**DDA**: Serial Data input/output for On-Chip Debug Interface when OCD is enabled and the external clock is selected as the clock source. |
| 10 | P3.5 | I/O<br>I/O | **P3.5**: User-configurable I/O Port 3 bit 5.<br>**T1**: Timer/Counter 1 External input or PWM output. |
| 11 | P3.1 | I/O<br>O | **P3.1**: User-configurable I/O Port 3 bit 1.<br>**TXD**: Serial Port Transmitter Output. |
| 12 | VDD | I | Supply Voltage |
| 13 | P1.0 | I/O<br>I<br>I | **P1.0**: User-configurable I/O Port 1 bit 0.<br>**AIN0**: Analog Comparator Positive input.<br>**GPI0**: General-purpose Interrupt input 0. |
| 14 | P1.1 | I/O<br>I<br>I | **P1.1**: User-configurable I/O Port 1 bit 1.<br>**AIN1**: Analog Comparator Negative input.<br>**GPI1**: General-purpose Interrupt input 1 |
| 15 | P1.4 | I/O<br>I<br>I | **P1.4**: User-configurable I/O Port 1 bit 4.<br>**$\overline{\text{SS}}$**: SPI slave select input.<br>**GPI4**: General-purpose Interrupt input 4. |
| 16 | P1.6 | I/O<br>I/O<br><br>I | **P1.6**: User-configurable I/O Port 1 bit 6.<br>**MISO**: SPI master-in/slave-out. When configured as master, this pin is an input. When configured as slave, this pin is an output.<br>**GPI6**: General-purpose Interrupt input 6. |

# 4. Block Diagram

**Figure 4-1.** AT89LP216 Block Diagram




# 5. Comparison to Standard 8051

The AT89LP216 is part of a family of devices with enhanced features that are fully binary compatible with the MCS-51 instruction set. In addition, most SFR addresses, bit assignments, and pin alternate functions are identical to Atmel's existing standard 8051 products. However, due to the high performance nature of the device, some system behaviors are different from those of Atmel's standard 8051 products such as AT89S52 or AT89S2051. The differences from the standard 8051 are outlined in the following paragraphs.

## 5.1 System Clock

The CPU clock frequency equals the external XTAL1 frequency. The oscillator is no longer divided by 2 to provide the internal clock and x2 mode is not supported.

## 5.2 Instruction Execution with Single-cycle Fetch

The CPU fetches one code byte from memory every clock cycle instead of every six clock cycles. This greatly increases the throughput of the CPU. As a consequence, the CPU no longer executes instructions in 12 to 48 clock cycles. Each instruction executes in only 1 to 4 clock cycles. See "Instruction Set Summary" on page 59 for more details.

## 5.3 Interrupt Handling

The interrupt controller polls the interrupt flags during the last clock cycle of any instruction. In order for an interrupt to be serviced at the end of an instruction, its flag needs to have been latched as active during the next to last clock cycle of the instruction, or in the last clock cycle of the previous instruction if the current instruction executes in only a single clock cycle.

The external interrupt pins, $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$, are sampled at every clock cycle instead of once every 12 clock cycles. Coupled with the shorter instruction timing and faster interrupt response, this leads to a higher maximum rate of incidence for the external interrupts.

## 5.4 Timer/Counters

By default the Timer/Counters is incremented at a rate of once per clock cycle. This compares to once every 12 clocks in the standard 8051. A common prescaler is available to divide the time base for all timers and reduce the increment rate. The TPS bits in the CLKREG SFR control the prescaler (Table 9-2 on page 13). Setting TPS = 1011B will cause the timers to count once every 12 clocks.

The external Timer/Counter pins, T0 and T1, are sampled at every clock cycle instead of once every 12 clock cycles. This increases the maximum rate at which the Counter modules may function.

## 5.5 Serial Port

The baud rate of the UART in Mode 0 is 1/2 the clock frequency, compared to 1/12 the clock frequency in the standard 8051; and output data is only stable around the rising edge of the serial clock. In should also be noted that when using Timer 1 to generate the baud rate in Mode 1 or Mode 3, the timer counts at the clock frequency and not at 1/12 the clock frequency. To maintain the same baud rate in the AT89LP216 while running at the same frequency as a standard 8051, the time-out period must be 12 times longer. Mode 1 of Timer 1 supports 16-bit auto-reload to facilitate longer time-out periods for generating low baud rates.

## 5.6 Watchdog Timer

The Watchdog Timer in AT89LP216 counts at a rate of once per clock cycle. This compares to once every 12 clocks in the standard 8051. A common prescaler is available to divide the time base for all timers and reduce the counting rate.

## 5.7 I/O Ports

The I/O ports of the AT89LP216 may be configured in four different modes. By default all the I/O ports revert to input-only (tristated) mode at power-up or reset. In the standard 8051, all ports are weakly pulled high during power-up or reset. To enable 8051-like ports, the ports must be put into quasi-bidirectional mode by clearing the P1M0 and P3M0 SFRs. The user can also configure the ports to start in quasi-bidirectional mode by disabling the Tristate-Port User Fuse. When this fuse is disabled, P1M0 and P3M0 will reset to 00h instead of FFh and the ports will be weakly pulled high.

## 5.8 Reset

The $\overline{\text{RST}}$ pin of the AT89LP216 is active-low as compared with the active high reset in the standard 8051. In addition, the $\overline{\text{RST}}$ pin is sampled every clock cycle and must be held low for a minimum of two clock cycles, instead of 24 clock cycles, to be recognized as a valid reset.

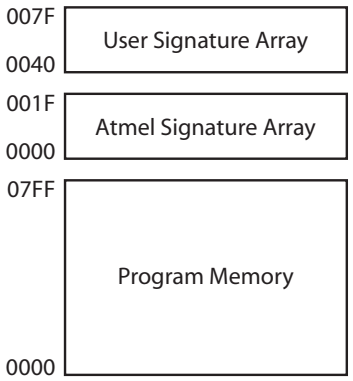
# 6. Memory Organization

The AT89LP216 uses a Harvard Architecture with separate address spaces for program and data memory. The program memory has a regular linear address space with support for up to 64K bytes of directly addressable application code. The data memory has 128 bytes of internal RAM and 128 bytes of Special Function Register I/O space. The AT89LP216 does not support external data memory or external program memory.

## 6.1 Program Memory

The AT89LP216 contains 2K bytes of on-chip In-System Programmable Flash memory for program storage. The Flash memory has an endurance of at least 10,000 write/erase cycles and a minimum data retention time of 10 years. The reset and interrupt vectors are located within the first 59 bytes of program memory (refer to Table 12-1 on page 19). Constant tables can be allocated within the entire 2K program memory address space for access by the MOVC instruction. The AT89LP216 does not support external program memory.

**Figure 6-1.** Program Memory Map




A map of the AT89LP216 program memory is shown in Figure 6-1. In addition to the 2K code space from 0000h to 07FFh, the AT89LP216 also supports a 64-byte User Signature Array and a 32-byte Atmel Signature Array that are accessible by the CPU in a read-only fashion. In order to read from the signature arrays, the SIGEN bit in AUXR1 must be set. While SIGEN is one, MOVC A,@A+DPTR will access the signature arrays. The User Signature Array is mapped to addresses 0040h to 007Fh and the Atmel Signature Array is mapped to addresses 0000h to 001Fh. SIGEN must be cleared before using MOVC to access the code memory.

The Atmel Signature Array is initialized with the Device ID in the factory. The User Signature Array is available for user identification codes or constant parameter data. Data stored in the signature array is not secure. Security bits will disable writes to the array; however, reads are always allowed.

**Table 6-1.** AUXR1 – Auxiliary Register 1

AUXR1 = A2H — Reset Value = XXXX 0XXXB

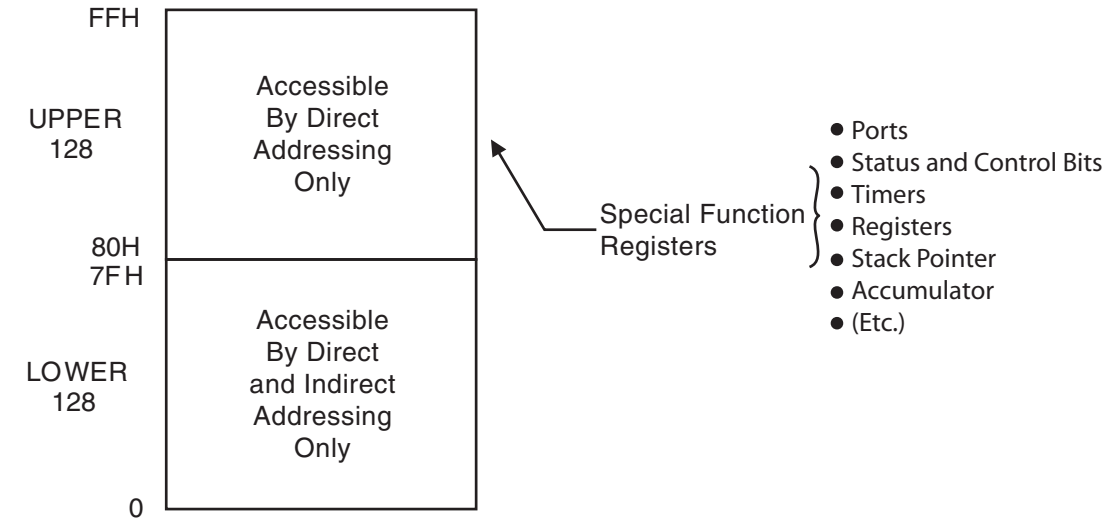
Not Bit Addressable

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | – | – | – | – | SIGEN | – | – | – |
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

## 6.2 Data Memory

The AT89LP216 contains 128 bytes of general SRAM data memory plus 128 bytes of I/O memory mapped into a single 8-bit address space. The 128 bytes of data memory may be accessed through both direct and indirect addressing of the lower 128 byte addresses. The 128 bytes of I/O memory reside in the upper 128 byte address space (Figure 6-2). The I/O memory can only be accessed through direct addressing and contains the Special Function Registers (SFRs). Indirect accesses to the upper 128 byte addresses will return invalid data. The lowest 32 bytes of data memory are grouped into 4 banks of 8 registers each. The RS0 and RS1 bits (PSW.3 and PSW.4) select which register bank is in use. Instructions using register addressing will only access the currently specified bank. The AT89LP216 does not support external data memory.

**Figure 6-2.** Data Memory Map



# 7. Special Function Registers

A map of the on-chip memory area called the Special Function Register (SFR) space is shown in Table 7-1.

Note that not all of the addresses are occupied, and unoccupied addresses may not be implemented on the chip. Read accesses to these addresses will in general return random data, and write accesses will have an indeterminate effect. User software should not write to these unlisted locations, since they may be used in future products to invoke new features.

**Table 7-1.** AT89LP216 SFR Map and Reset Values

| | 8 | 9 | A | B | C | D | E | F | |
|---|---|---|---|---|---|---|---|---|---|
| 0F8H | | | | | | | | | 0FFH |
| 0F0H | B<br>0000 0000 | | | | | | | | 0F7H |
| 0E8H | SPSR<br>000x x000 | SPCR<br>0000 0000 | SPDR<br>xxxx xxxx | | | | | | 0EFH |
| 0E0H | ACC<br>0000 0000 | | | | | | | | 0E7H |
| 0D8H | | | | | | | | | 0DFH |
| 0D0H | PSW<br>0000 0000 | | | | | | | | 0D7H |
| 0C8H | | | | | | | | | 0CFH |
| 0C0H | | | P1M0[2] | P1M1<br>xx00 0000 | | | P3M0[2] | P3M1<br>xx00 0000 | 0C7H |
| 0B8H | IP<br>x000 0000 | SADEN<br>0000 0000 | | | | | | | 0BFH |
| 0B0H | P3<br>xx11 1111 | | | | | | | IPH<br>x000 0000 | 0B7H |
| 0A8H | IE<br>0000 0000 | SADDR<br>0000 0000 | | | | | | | 0AFH |
| 0A0H | | | AUXR1<br>xxxx 0xxx | | | | WDTRST<br>(write-only) | WDTCON<br>0000 x000 | 0A7H |
| 98H | SCON<br>0000 0000 | SBUF<br>xxxx xxxx | GPMOD<br>0000 0000 | GPLS<br>0000 0000 | GPIEN<br>0000 0000 | GPIF<br>0000 0000 | | | 9FH |
| 90H | P1<br>1111 1111 | TCONB<br>0010 0100 | RL0<br>0000 0000 | RL1<br>0000 0000 | RH0<br>0000 0000 | RH1<br>0000 0000 | | ACSR<br>xx00 0000 | 97H |
| 88H | TCON<br>0000 0000 | TMOD<br>0000 0000 | TL0<br>0000 0000 | TL1<br>0000 0000 | TH0<br>0000 0000 | TH1<br>0000 0000 | | CLKREG<br>0000 x000 | 8FH |
| 80H | | SP<br>0000 0111 | DPL<br>0000 0000 | DPH<br>0000 0000 | | | | PCON<br>0000 0000 | 87H |
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | |

Notes: 1. All SFRs in the left-most column are bit-addressable.

2. Reset value is xx11 1111B when Tristate-Port Fuse is enabled and xx00 0000B when disabled.

# 8. Enhanced CPU

The AT89LP216 uses an enhanced 8051 CPU that runs at 6 to 12 times the speed of standard 8051 devices (or 3 to 6 times the speed of X2 8051 devices). The increase in performance is due to two factors. First, the CPU fetches one instruction byte from the code memory every clock cycle. Second, the CPU uses a simple two-stage pipeline to fetch and execute instructions in parallel. This basic pipelining concept allows the CPU to obtain up to 1 MIPS per MHz. A simple example is shown in Figure 8-1.

The MCS-51 instruction set allows for instructions of variable length from 1 to 3 bytes. In a single-clock-per-byte-fetch system this means each instruction takes at least as many clocks as it has bytes to execute. The majority of instructions in the AT89LP216 follow this rule: the instruction execution time in clock cycles equals the number of bytes per instruction with a few exceptions. Branches and Calls require an additional cycle to compute the target address and some other complex instructions require multiple cycles. See “Instruction Set Summary” on page 59 for more detailed information on individual instructions. Figures 8-2 and 8-3 show examples of 1- and 2-byte instructions.

**Figure 8-1.** Parallel Instruction Fetches and Executions

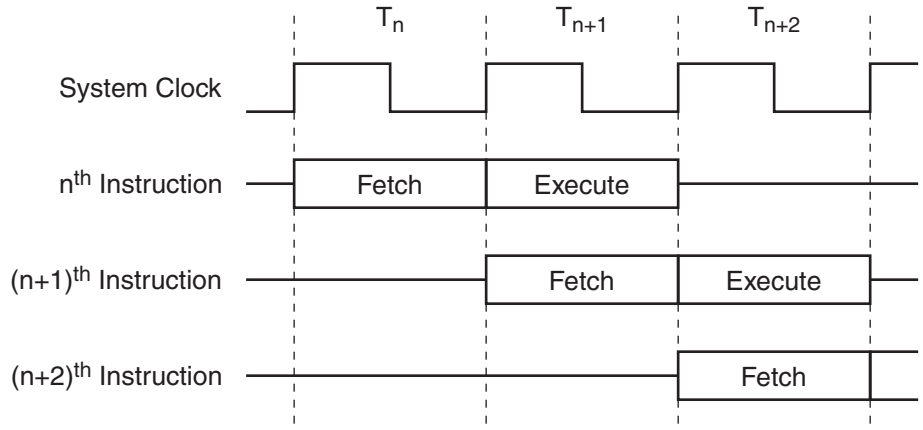



**Figure 8-2.** Single-cycle ALU Operation (Example: INC R0)

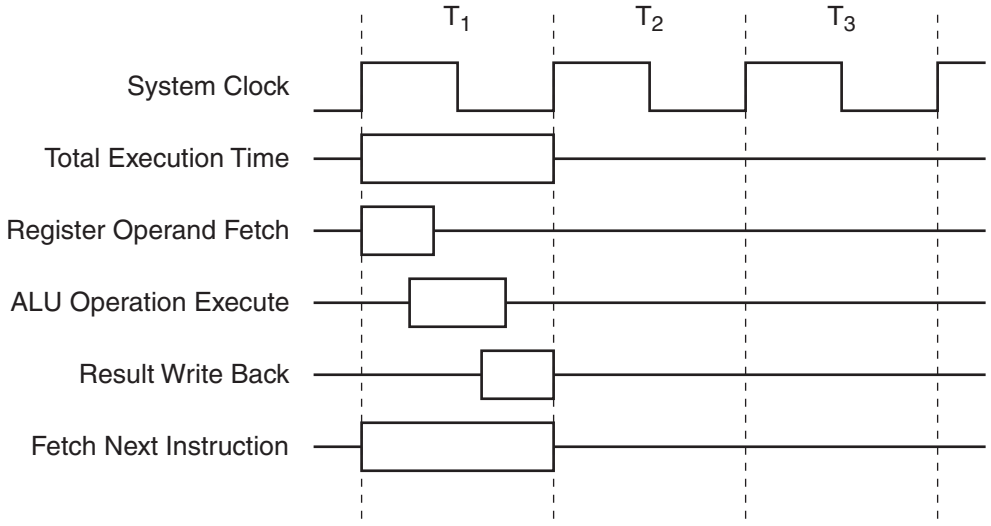
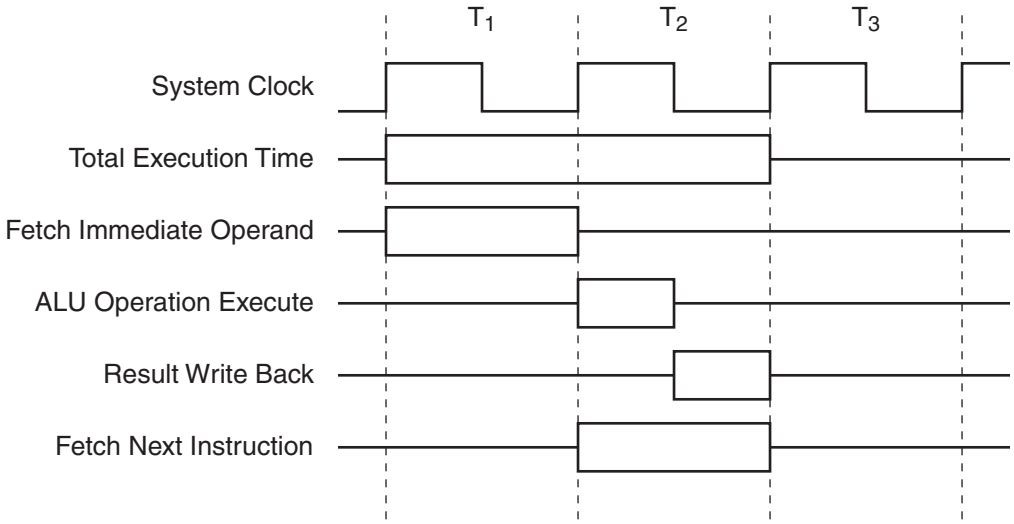


**Figure 8-3.** Two-cycle ALU Operation (Example: ADD A, #Data)




## 8.1 Restrictions on Certain Instructions

The AT89LP216 is an economical and cost-effective member of Atmel's growing family of microcontrollers. It contains 2K bytes of Flash program memory. It is fully compatible with the MCS-51 architecture, and can be programmed using the MCS-51 instruction set. However, there are a few considerations one must keep in mind when utilizing certain instructions to program this device. All the instructions related to jumping or branching should be restricted such that the destination address falls within the physical program memory space of the device, which is 2K for the AT89LP216. This should be the responsibility of the software programmer. For example, LJMP 7E0H would be a valid instruction, whereas LJMP 900H would not.

### 8.1.1 Branching Instructions

The LCALL, LJMP, ACALL, AJMP, SJMP, and JMP @A+DPTR unconditional branching instructions will execute correctly as long as the programmer keeps in mind that the destination branching address must fall within the physical boundaries of the program memory size (locations 000H to 7FFH for the AT89LP216). Violating the physical space limits may cause unknown program behavior. With the CJNE [...], DJNZ [...], JB, JNB, JC, JNC, JBC, JZ, and JNZ conditional branching instructions, the same previous rule applies. Again, violating the memory boundaries may cause erratic execution. For applications involving interrupts the normal interrupt service routine address locations of the 8051 family architecture have been preserved.

### 8.1.2 MOVX-related Instructions, Data Memory

The AT89LP216 contains 128 bytes of internal data memory. The RAM accesses to addresses above 7FH will return invalid data. Furthermore, the stack depth is limited to 128 bytes, the amount of available RAM. The Stack Pointer should not be allowed to point to locations above 7FH. External DATA memory access is not supported in this device, nor is external PROGRAM memory execution. Therefore, no MOVX [...] instructions should be included in the program.

A typical 8051 assembler will still assemble instructions, even if they are written in violation of the restrictions mentioned above. It is the responsibility of the user to know the physical features and limitations of the device being used and to adjust the instructions used accordingly.

# 9. System Clock

The system clock is generated directly from one of three selectable clock sources. The three sources are the on-chip crystal oscillator, external clock source, and internal RC oscillator. The clock source is selected by the Clock Source User Fuses as shown in Table 9-1. No internal clock division is used to generate the CPU clock from the system clock. See "User Configuration Fuses" on page 71.

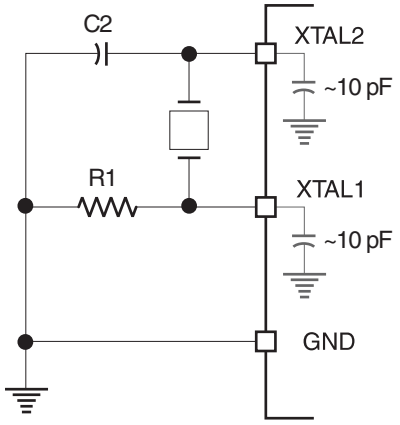
**Table 9-1.** Clock Source Settings

| Clock Source Fuse 1 | Clock Source Fuse 0 | Selected Clock Source |
|---|---|---|
| 0 | 0 | Crystal Oscillator |
| 0 | 1 | Reserved |
| 1 | 0 | External Clock on XTAL1 |
| 1 | 1 | Internal 8 MHz RC Oscillator |

## 9.1 Crystal Oscillator

When enabled, the internal inverting oscillator amplifier is connected between XTAL1 and XTAL2. Note that the internal structure of the device adds about 10 pF of capacitance to both XTAL1 and XTAL2, so that in some cases an external capacitor may **NOT** be required. It is recommended that a resistor R1 be connected to XTAL1, instead of load capacitor C1, for improved startup performance. The total capacitance on XTAL1 or XTAL2, including the external load capacitor plus internal device load, board trace and crystal loadings, should not exceed 20 pF. When using the crystal oscillator, P3.2 and P3.3 will have their inputs and outputs disabled. When using the crystal oscillator, XTAL2 should not be used to drive a board-level clock without a buffer

**Figure 9-1.** Crystal Oscillator Connections

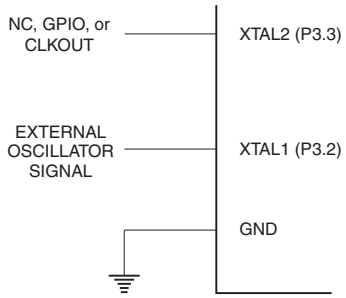



Note: 1. C2 = 0–10 pF for Crystals
= 0–10 pF for Ceramic Resonators
R1 = 4–5 MΩ

## 9.2 External Clock Source

The external clock option disables the oscillator amplifier and allows XTAL1 to be driven directly by the clock source as shown in Figure 9-2. XTAL2 may be left unconnected, used as P3.3 I/O, or configured to output a divided version of the system clock.

**Figure 9-2.** External Clock Drive Configuration




## 9.3 Internal RC Oscillator

The AT89LP216 has an internal RC oscillator tuned to 8.0 MHz ±1.0% at 5.0V and 25· C. When enabled as the clock source, XTAL1 and XTAL2 may be used as P3.2 and P3.3, respectively. The XTAL2 may also be configured to output a divided version of the system clock. The frequency of the oscillator may be adjusted by changing the RC Adjust Fuses. (See “User Configuration Fuses” on page 71.) A copy of the initial factory setting is stored at location 0007h of the Atmel Signature.

## 9.4 System Clock Out

When the AT89LP216 is configured to use either an external clock or the internal RC oscillator, a divided version of the system clock may be output on XTAL2 (P3.3). The Clock Out feature is enabled by setting the COE bit in CLKREG. The two CDV bits determine the clock divide ratio. For example, setting COE = “1” and CDIV = “00” when using the internal oscillator will result in a 4.0 MHz clock output on P3.3. P3.3 must be configured as an output in order to use the clock out feature.

**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<br>TPS2<br>TPS1<br>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<br>CDV0 | Clock Out Division. Determines the frequency of the clock output relative to the system clock.<br>CDIV1 CDIV0 Clock Out Frequency<br>0 0 f/2<br>0 1 f/4<br>1 0 f/8<br>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. |

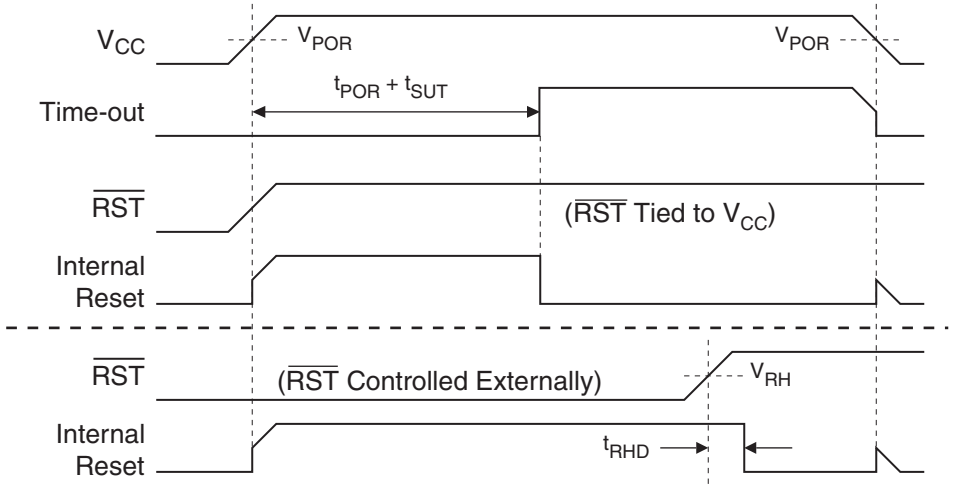
# 10. Reset

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

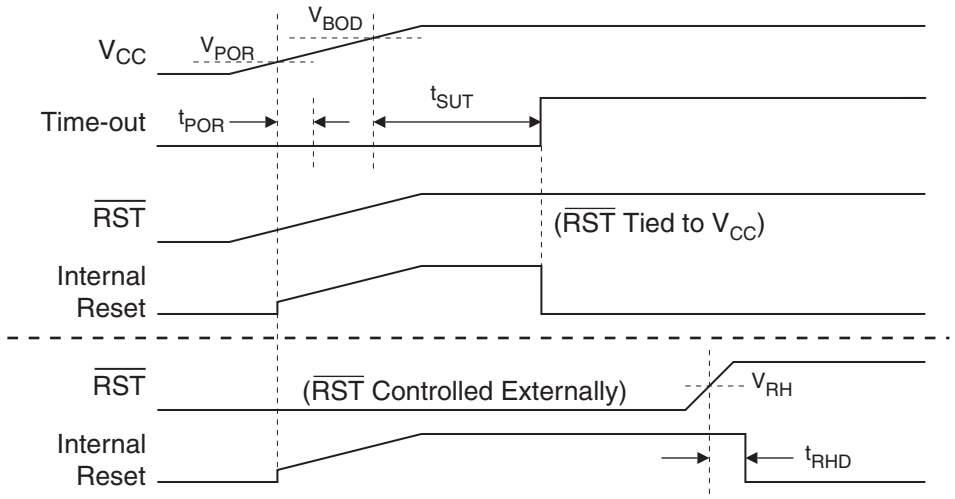
A Power-on Reset (POR) is generated by an on-chip detection circuit. The detection level is nominally 1.4V. The POR is activated whenever $V_{CC}$ is below the detection level. The POR circuit can be used to trigger the start-up reset or to detect a supply voltage failure in devices without a brown-out 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.

**Figure 10-1.** Power-on Reset Sequence (BOD Disabled)




If the Brown-out Detector (BOD) is also enabled, the start-up timer does not begin counting until after $V_{CC}$ reaches the BOD threshold voltage $V_{BOD}$ as shown in Figure 10-2. However, if this event occurs prior to the end of the initialization sequence, the timer must first wait for that sequence to complete before counting.

**Figure 10-2.** Power-on Reset Sequence (BOD Enabled)




Note: $t_{POR}$ is approximately 92 µs ± 5%.

The start-up timer delay is user configurable with the Start-up Time User Fuses and depends on the clock source (Table 10-1). The start-up delay should be selected to provide enough settling time for $V_{CC}$ and the selected clock source. The Start-Up Time fuses also control the length of the start-up time after a Brown-out Reset or when waking up from Power-down during internally timed mode.

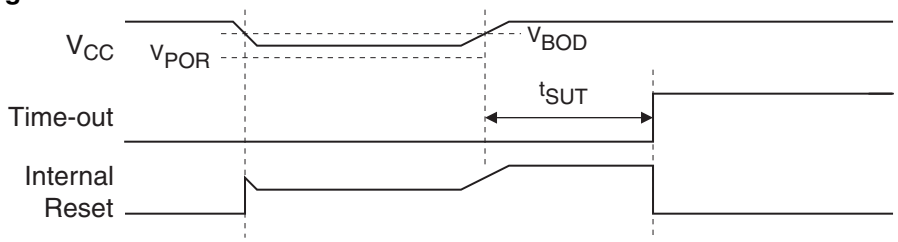
**Table 10-1.** Start-up Timer Settings

| SUT Fuse 1 | SUT Fuse 0 | Clock Source | $t_{SUT}$ (±5%) |
|---|---|---|---|
| 0 | 0 | Internal RC/External Clock | 16 µs |
| | | Crystal Oscillator | 1024 µs |
| 0 | 1 | Internal RC/External Clock | 512 µs |
| | | Crystal Oscillator | 2048 µs |
| 1 | 0 | Internal RC/External Clock | 1024 µs |
| | | Crystal Oscillator | 4096 µs |
| 1 | 1 | Internal RC/External Clock | 4096 µs |
| | | Crystal Oscillator | 16384 µs |

## 10.2 Brown-out Reset

The AT89LP216 has an on-chip Brown-out Detection (BOD) circuit for monitoring the $V_{CC}$ level during operation by comparing it to a fixed trigger level. The trigger level for the BOD is nominally 2.2V. The purpose of the BOD is to ensure that if $V_{CC}$ fails or dips while executing at speed, the system will gracefully enter reset without the possibility of errors induced by incorrect execution. A BOD sequence is shown in Figure 10-3. When $V_{CC}$ decreases to a value below the trigger level $V_{BOD}$, the internal reset is immediately activated. When $V_{CC}$ increases above the trigger level, the start-up timer releases the internal reset after the specified time-out period has expired (Table 10-1). The Brown-out Detector must be enabled by setting the BOD Enable Fuse. (See “User Configuration Fuses” on page 71.)

**Figure 10-3.** Brown-out Detector Reset




## 10.3 External Reset

The P1.3/$\overline{RST}$ pin can function as either an active-LOW reset input or as a digital general-purpose I/O, P1.3. The Reset Pin Enable Fuse, when set to “1”, enables the external reset input function on P1.3. (See “User Configuration Fuses” on page 71.) When cleared, P1.3 may be used as an input or output pin. When configured as a reset input, the pin must be held low for at least two clock cycles to trigger the internal reset.

Note: During a power-up sequence, the fuse selection is always overridden and therefore the pin will always function as a reset input. **An external circuit connected to this pin should not hold this pin LOW during a power-on sequence as this will keep the device in reset until the pin transitions high**. After the power-up delay, this input will function either as an external reset input or as a digital input as defined by the fuse bit. Only a power-up reset will temporarily override the selection defined by the reset fuse bit. Other sources of reset will not override the reset fuse bit. P1.3/$\overline{RST}$ also serves as the In-System Programming (ISP) enable. ISP is enabled when the external reset pin is held low. When the reset pin is disabled by the fuse, ISP may only be entered by pulling P1.3 low during power-up.

## 10.4 Watchdog Reset

When the Watchdog times out, it will generate an internal reset pulse lasting 16 clock cycles. Watchdog reset will also set the WDTOVF flag in WDTCON. To prevent a Watchdog reset, the watchdog reset sequence 1EH/E1H must be written to WDTRST before the Watchdog times out. See “Programmable Watchdog Timer” on page 57 for details on the operation of the Watchdog.

## 10.5 Software Reset

The CPU may generate an internal 16-clock cycle reset pulse by writing the software reset sequence 5AH/A5H to the WDRST register. A software reset will set the SWRST bit in WDTCON. See “Software Reset” on page 58 for more information on software reset.

# 11. Power Saving Modes

The AT89LP216 supports two different power-reducing modes: Idle and Power-down. These modes are accessed through the PCON register.

## 11.1 Idle Mode

Setting the IDL bit in PCON enters idle mode. Idle mode halts the internal CPU clock. The CPU state is preserved in its entirety, including the RAM, stack pointer, program counter, program status word, and accumulator. The Port pins hold the logic states they had at the time that Idle was activated. Idle mode leaves the peripherals running in order to allow them to wake up the CPU when an interrupt is generated. The timers, UART, SPI, and GPI blocks continue to function during Idle. The comparator and watchdog may be selectively enabled or disabled during Idle. Any enabled interrupt source or reset may terminate Idle mode. When exiting Idle mode with an interrupt, the interrupt will immediately be serviced, and following RETI the next instruction to be executed will be the one following the instruction that put the device into Idle.

## 11.2 Power-down Mode

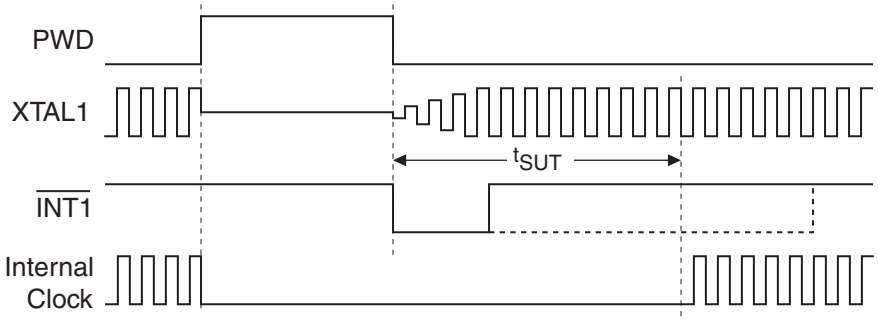
Setting the Power-down (PD) bit in PCON enters Power-down mode. Power-down mode stops the oscillator, disables the BOD and powers down the Flash memory in order to minimize power consumption. Only the power-on circuitry will continue to draw power during Power-down. During Power-down, the power supply voltage may be reduced to the RAM keep-alive voltage. The RAM contents will be retained, but the SFR contents are not guaranteed once $V_{CC}$ has been reduced. Power-down may be exited by external reset, power-on reset, or certain interrupts.

### 11.2.1 Interrupt Recovery from Power-down

Three external interrupts may be configured to terminate Power-down mode. XTAL1 or XTAL2, when not used for the crystal oscillator or external clock, may be used to exit Power-down through external interrupts $\overline{\text{INT0}}$ (P3.2) and $\overline{\text{INT1}}$ (P3.3). To wake up by external interrupt $\overline{\text{INT0}}$ or $\overline{\text{INT1}}$, that interrupt must be enabled and configured for level-sensitive operation. General purpose interrupt 3 (GPI3) can also wake up the device when the $\overline{\text{RST}}$ pin is disabled. GPI3 must be enabled and configured for low level detection in order to terminate Power-down.

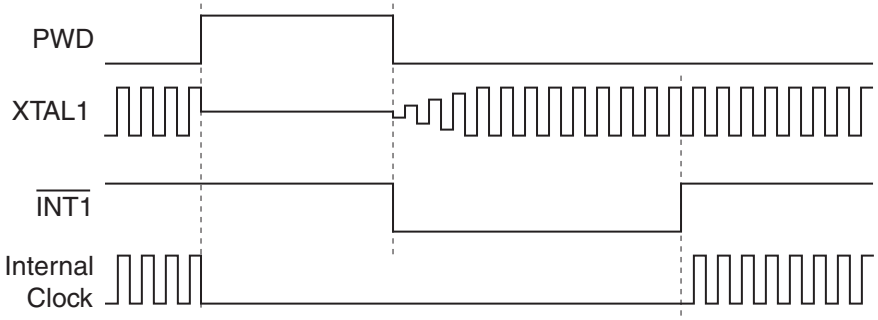
When terminating Power-down by an interrupt, two different wake-up modes are available. When PWDEX in PCON is zero, the wake-up period is internally timed as shown in Figure 11-1. At the falling edge on the interrupt pin, Power-down is exited, the oscillator is restarted, and an internal timer begins counting. The internal clock will not be allowed to propagate to the CPU until after the timer has timed out. After the time-out period the interrupt service routine will begin. The time-out period is controlled by the Start-up Timer Fuses (see Table 10-1 on page 15). The interrupt pin need not remain low for the entire time-out period.

**Figure 11-1.** Interrupt Recovery from Power-down (PWDEX = 0)




When PWDEX = “1”, the wake-up period is controlled externally by the interrupt. Again, at the falling edge on the interrupt pin, power-down is exited and the oscillator is restarted. However, the internal clock will not propagate until the rising edge of the interrupt pin as shown in Figure 11-2. The interrupt pin should be held low long enough for the selected clock source to stabilize. After the rising edge on the pin the interrupt service routine will be executed.

**Figure 11-2.** Interrupt Recovery from Power-down (PWDEX = 1)




### 11.2.2 Reset Recovery from Power-down

The wake-up from Power-down through an external reset is similar to the interrupt with PWDEX = “0”. At the falling edge of $\overline{RST}$, Power-down is exited, the oscillator is restarted, and an internal timer begins counting as shown in Figure 11-3. The internal clock will not be allowed to propagate to the CPU until after the timer has timed out. The time-out period is controlled by the Start-up Timer Fuses. (See Table 10-1 on page 15). If $\overline{RST}$ returns high before the time-out, a two clock cycle internal reset is generated when the internal clock restarts. Otherwise the device will remain in reset until $\overline{RST}$ is brought high.

**Figure 11-3.** Reset Recovery from Power-down.

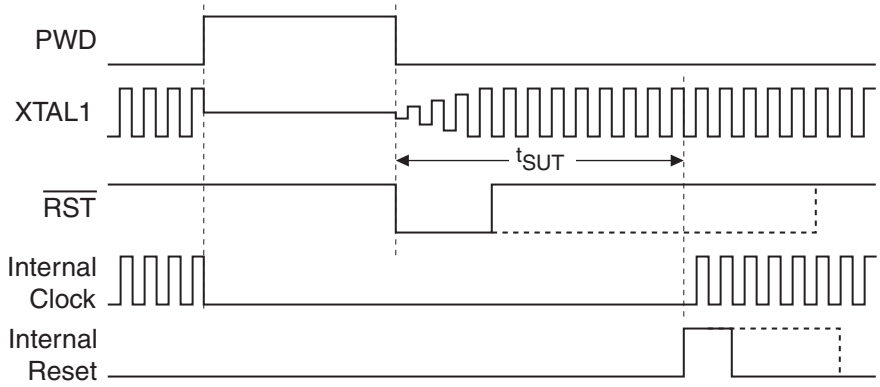


**Table 11-1.** PCON – Power Control Register

PCON = 87H — Reset Value = 000X 0000B

Not Bit Addressable

| | SMOD1 | SMOD0 | PWDEX | POF | GF1 | GF0 | PD | IDL |
|---|---|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

| Symbol | Function |
|---|---|
| SMOD1 | Double Baud Rate bit. Doubles the baud rate of the UART in Modes 1, 2, or 3. |
| SMOD0 | Frame Error Select. When SMOD0 = 1, SCON.7 is SM0. When SMOD0 = 1, SCON.7 is FE. Note that FE will be set after a frame error regardless of the state of SMOD0. |
| PWDEX | Power-down Exit Mode. When PWDEX = 1, wake up from Power-down is externally controlled. When PWDEX = 1, wake up from Power-down is internally timed. |
| POF | Power Off Flag. POF is set to “1” during power up (i.e. cold reset). It can be set or reset under software control and is not affected by RST or BOD (i.e. warm resets). |
| GF1, GF0 | General-purpose Flags |
| PD | Power-down bit. Setting this bit activates power-down operation. |
| IDL | Idle Mode bit. Setting this bit activates Idle mode operation |

# 12. Interrupts

The AT89LP216 provides 7 interrupt sources: two external interrupts, two timer interrupts, a serial port interrupt, a general-purpose interrupt, and an analog comparator interrupt. These interrupts and the system reset each have a separate program vector at the start of the program memory space. Each interrupt source can be individually enabled or disabled by setting or clearing a bit in the interrupt enable register IE. The IE register also contains a global disable bit, EA, which disables all interrupts.

Each interrupt source (**except** the analog comparator) can be individually programmed to one of four priority levels by setting or clearing bits in the interrupt priority registers IP and IPH. The analog comparator is fixed at the lowest priority level. An interrupt service routine in progress can be interrupted by a higher priority interrupt, but not by another interrupt of the same or lower priority. The highest priority interrupt cannot be interrupted by any other interrupt source. If two requests of different priority levels are pending at the end of an instruction, the request of higher priority level is serviced. If requests of the same priority level are pending at the end of an instruction, an internal polling sequence determines which request is serviced. The polling sequence is based on the vector address; an interrupt with a lower vector address has higher priority than an interrupt with a higher vector address. Note that the polling sequence is only used to resolve pending requests of the same priority level.

The External Interrupts $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ can each be either level-activated or edge-activated, depending on bits IT0 and IT1 in Register TCON. The flags that actually generate these interrupts are the IE0 and IE1 bits in TCON. When the service routine is vectored to, hardware clears the flag that generated an external interrupt only if the interrupt was edge-activated. If the interrupt was level activated, then the external requesting source (rather than the on-chip hardware) controls the request flag.

The Timer 0 and Timer 1 Interrupts are generated by TF0 and TF1, which are set by a rollover in their respective Timer/Counter registers (except for Timer 0 in Mode 3). When a timer interrupt is generated, the on-chip hardware clears the flag that generated it when the service routine is vectored to.

The Serial Port Interrupt is generated by the logic OR of RI and TI in SCON plus SPIF in SPSR. None of these flags is cleared by hardware when the service routine is vectored to. In fact, the service routine normally must determine whether RI, TI, or SPIF generated the interrupt, and the bit must be cleared by software.

A logic OR of all eight flags in the GPIF register causes the general-purpose interrupt. None of these flags is cleared by hardware when the service routine is vectored to. The service routine must determine which bit generated the interrupt, and the bit must be cleared in software. If the interrupt was level activated, then the external requesting source must de-assert the interrupt before the flag may be cleared by software.

The CF bit in ACSR generates the Comparator Interrupt. The flag is not cleared by hardware when the service routine is vectored to and must be cleared by software.

Most of the bits that generate interrupts can be set or cleared by software, with the same result as though they had been set or cleared by hardware. That is, interrupts can be generated and pending interrupts can be canceled in software. The two exceptions are the SPI interrupt flag SPIF and the general-purpose interrupt flags in GPIF. These flags are only set by hardware and may only be cleared by software.

**Table 12-1.** Interrupt Vector Addresses

| Interrupt | Source | Vector Address |
|---|---|---|
| System Reset | RST or POR or BOD | 0000H |
| External Interrupt 0 | IE0 | 0003H |
| Timer 0 Overflow | TF0 | 000BH |
| External Interrupt 1 | IE1 | 0013H |
| Timer 1 Overflow | TF1 | 001BH |
| Serial Port | RI or TI or SPIF | 0023H |
| General-purpose Interrupt | GPIF | 002BH |
| Analog Comparator | CF | 0033H |

## 12.1 Interrupt Response Time

The interrupt flags may be set by their hardware in any clock cycle. The interrupt controller polls the flags in the last clock cycle of the instruction in progress. If one of the flags was set in the preceding cycle, the polling cycle will find it and the interrupt system will generate an LCALL to the appropriate service routine as the next instruction, provided that the interrupt is not blocked by any of the following conditions: an interrupt of equal or higher priority level is already in progress; the instruction in progress is RETI or any write to the IE, IP, or IPH registers. Either of these conditions will block the generation of the LCALL to the interrupt service routine. The second condition ensures that if the instruction in progress is RETI or any access to IE, IP or IPH, then at least one more instruction will be executed before any interrupt is vectored to. The polling cycle is repeated at the last cycle of each instruction, and the values polled are the values that were present at the previous clock cycle. If an active interrupt flag is not being serviced because of one of the above conditions and is no longer active when the blocking condition is removed, the denied interrupt will not be serviced. In other words, the fact that the interrupt flag was once active but not serviced is not remembered. Every polling cycle is new.

If a request is active and conditions are met for it to be acknowledged, a hardware subroutine call to the requested service routine will be the next instruction executed. The call itself takes four cycles. Thus, a minimum of five complete clock cycles elapsed between activation of an interrupt request and the beginning of execution of the first instruction of the service routine. A longer response time results if the request is blocked by one of the previously listed conditions. If an interrupt of equal or higher priority level is already in progress, the additional wait time depends on the nature of the other interrupt's service routine. If the instruction in progress is not in its final clock cycle, the additional wait time cannot be more than 3 cycles, since the longest are only 4 cycles long. If the instruction in progress is RETI or an access to IE or IP, the additional wait time cannot be more than 7 cycles (a maximum of three more cycles to complete the instruction in progress, plus a maximum of 4 cycles to complete the next instruction). Thus, in a single-interrupt system, the response time is always more than 5 clock cycles and less than 13 clock cycles. See Figures 12-1 and 12-2.

**Figure 12-1.** Minimum Interrupt Response Time

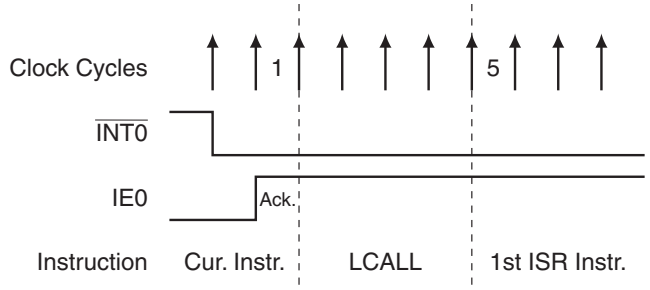



**Figure 12-2.** Maximum Interrupt Response Time

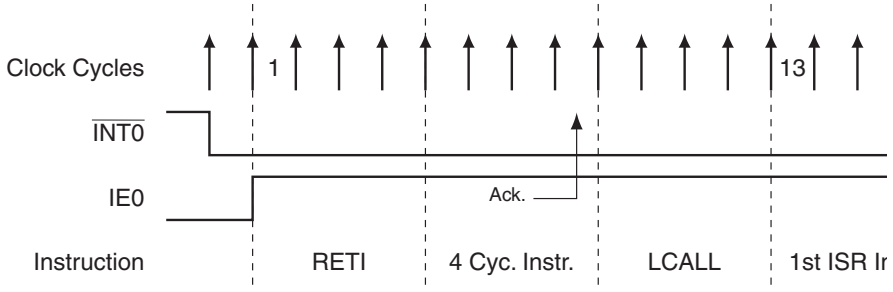


**Table 12-2.** IE – Interrupt Enable Register

IE = A8H — Reset Value = 0000 0000B

Bit Addressable

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | EA | EC | EGP | ES | ET1 | EX1 | ET0 | EX0 |
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

| Symbol | Function |
|---|---|
| EA | Global enable/disable. All interrupts are disabled when EA = 0. When EA = 1, each interrupt source is enabled/disabled by setting /clearing its own enable bit. |
| EC | Comparator Interrupt Enable |
| EGP | General-purpose Interrupt Enable |
| ES | Serial Port Interrupt Enable |
| ET1 | Timer 1 Interrupt Enable |
| EX1 | External Interrupt 1 Enable |
| ET0 | Timer 0 Interrupt Enable |
| EX0 | External Interrupt 0 Enable |

**Table 12-3.** IP – Interrupt Priority Register

IP = B8H — Reset Value = X000 0000B

Bit Addressable

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | – | – | PGP | PS | PT1 | PX1 | PT0 | PX0 |
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

| Symbol | Function |
|---|---|
| PGP | General-purpose Interrupt Priority Low |
| PS | Serial Port Interrupt Priority Low |
| PT1 | Timer 1 Interrupt Priority Low |
| PX1 | External Interrupt 1 Priority Low |
| PT0 | Timer 0 Interrupt Priority Low |
| PX0 | External Interrupt 0 Priority Low |

**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 $\overline{\text{RST}}$ Pin | 11 |
| | No external reset | 12 |
| External Clock | External $\overline{\text{RST}}$ Pin | 12 |
| | No external reset | 13 |
| Internal RC Oscillator | External $\overline{\text{RST}}$ 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 13-2.** Configuration Modes for Port x, Bit y

| PxM0.y | PxM1.y | Port Mode |
|---|---|---|
| 0 | 0 | Quasi-bidirectional |
| 0 | 1 | Push-pull Output |
| 1 | 0 | Input Only (High Impedance) |
| 1 | 1 | Open-drain Output |

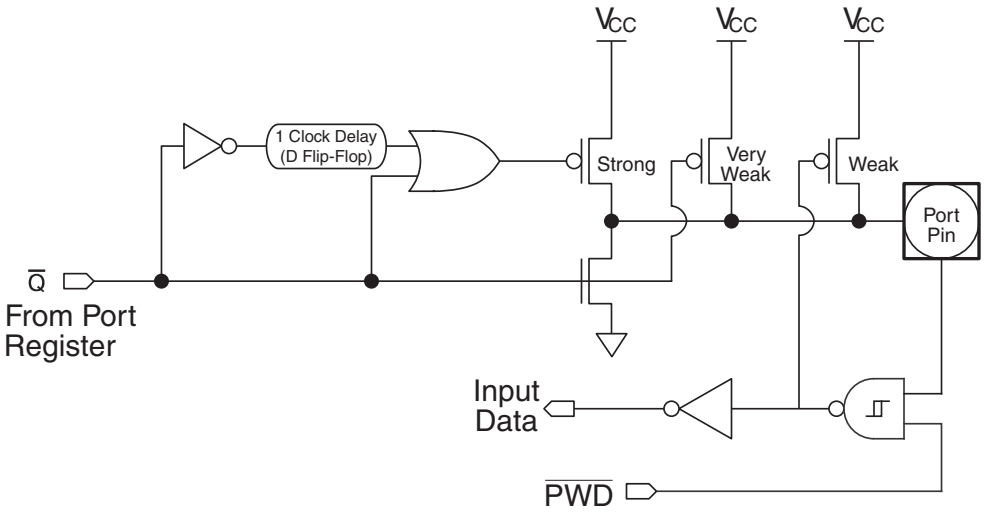
### 13.1.1 Quasi-bidirectional Output

Port pins in quasi-bidirectional output mode function similar to standard 8051 port pins. A Quasi-bidirectional port can be used both as an input and output without the need to reconfigure the port. This is possible because when the port outputs a logic high, it is weakly driven, allowing an external device to pull the pin low. When the pin is driven low, it is driven strongly and able to sink a large current. There are three pull-up transistors in the quasi-bidirectional output that serve different purposes. One of these pull-ups, called the “very weak” pull-up, is turned on whenever the port latch for the pin contains a logic “1”. This very weak pull-up sources a very small current that will pull the pin high if it is left floating.

A second pull-up, called the “weak” pull-up, is turned on when the port latch for the pin contains a logic “1” and the pin itself is also at a logic “1” level. This pull-up provides the primary source current for a quasi-bidirectional pin that is outputting a “1”. If this pin is pulled low by an external device, this weak pull-up turns off, and only the very weak pull-up remains on. In order to pull the pin low under these conditions, the external device has to sink enough current to overpower the weak pull-up and pull the port pin below its input threshold voltage.

The third pull-up is referred to as the “strong” pull-up. This pull-up is used to speed up low-to-high transitions on a quasi-bidirectional port pin when the port latch changes from a logic “0” to a logic “1”. When this occurs, the strong pull-up turns on for two CPU clocks quickly pulling the port pin high. The quasi-bidirectional port configuration is shown in Figure 13-1. The input circuitry of P1.3, P3.2 and P3.3 is not disabled during Power-down (see Figure 13-3).

**Figure 13-1.** Quasi-bidirectional Output



### 13.1.2 Input-only Mode

The input only port configuration is shown in Figure 13-2. The output drivers are tristated. The input includes a Schmitt-triggered input for improved input noise rejection. The input circuitry of P1.3, P3.2 and P3.3 is not disabled during Power-down (see Figure 13-3). Input pins can be safely driven to 5.5V even when operating at lower $V_{CC}$ levels; however, the input threshold of the Schmitt trigger will be set by the $V_{CC}$ level and must be taken into consideration.

**Figure 13-2.** Input Only

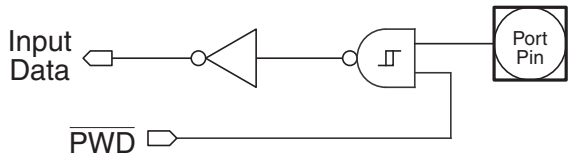
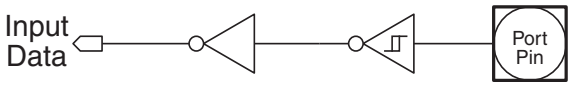


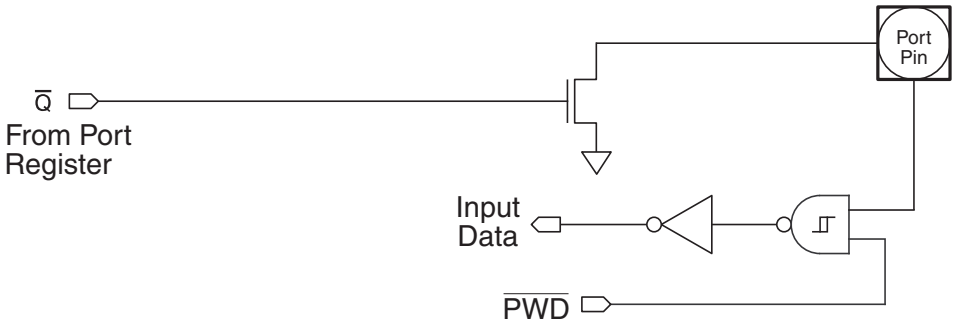

**Figure 13-3.** Input Only for P1.3, P3.2 and P3.3




### 13.1.3 Open-drain Output

The open-drain output configuration turns off all pull-ups and only drives the pull-down transistor of the port pin when the port latch contains a logic “0”. To be used as a logic output, a port configured in this manner must have an external pull-up, typically a resistor tied to $V_{CC}$. The pull-down for this mode is the same as for the quasi-bidirectional mode. The open-drain port configuration is shown in Figure 13-4.The input circuitry of P1.3, P3.2 and P3.3 is not disabled during Power-down (see Figure 13-3). Open-drain pins can be safely pulled high to 5.5V even when operating at lower $V_{CC}$ levels; however, the input threshold of the Schmitt trigger will be set by the $V_{CC}$ level and must be taken into consideration.

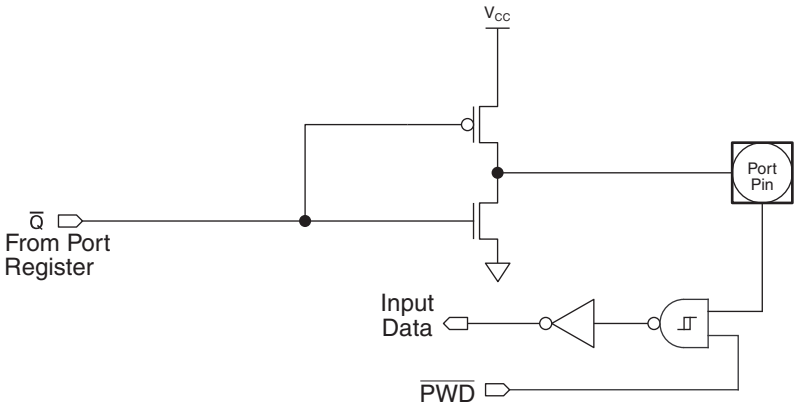
**Figure 13-4.** Open-drain Output




### 13.1.4 Push-pull Output

The push-pull output configuration has the same pull-down structure as both the open-drain and the quasi-bidirectional output modes, but provides a continuous strong pull-up when the port latch contains a logic “1”. The push-pull mode may be used when more source current is needed from a port output. Note that due to the 5V tolerant architecture, the push-pull output will have reduced output high levels at DC operation and hot temperature. Under AC operation an integrated boost circuit provides more source current. The push-pull port configuration is shown in Figure 13-5. The input circuitry of P1.3, P3.2 and P3.3 is not disabled during Power-down (see Figure 13-3).

**Figure 13-5.** Push-pull Output




## 13.2 Port 1 Analog Functions

The AT89LP216 incorporates an analog comparator. In order to give the best analog performance and minimize power consumption, pins that are being used for analog functions must have both their digital outputs and digital inputs disabled. Digital outputs are disabled by putting the port pins into the input-only mode as described in "Port Configuration" on page 22. Digital inputs on P1.0 and P1.1 are disabled whenever the Analog Comparator is enabled by setting the CEN bit in ACSR. CEN forces the $\overline{PWD}$ input on P1.0 and P1.1 low, thereby disabling the Schmitt trigger circuitry. P1.0 and P1.1 will always default to input-only mode after reset regardless of the state of the Tristate-Port Fuse.

## 13.3 Port Read-modify-write

A read from a port will read either the state of the pins or the state of the port register depending on which instruction is used. Simple read instructions will always access the port pins directly. Read-modify-write instructions, which read a value, possibly modify it, and then write it back, will always access the port register. This includes bit write instructions such as CLR or SETB as they actually read the entire port, modify a single bit, then write the data back to the entire port. See Table 13-3 for a complete list of Read-modify-write instruction which may access the ports.

**Table 13-3.** Port Read-modify-write Instructions

| Mnemonic | Instruction | Example |
|---|---|---|
| ANL | Logical AND | ANL P1, A |
| ORL | Logical OR | ORL P1, A |
| XRL | Logical EX-OR | XRL P1, A |
| JBC | Jump if bit set and clear bit | JBC P3.0, LABEL |
| CPL | Complement bit | CPL P3.1 |
| INC | Increment | INC P1 |
| DEC | Decrement | DEC P3 |
| DJNZ | Decrement and jump if not zero | DJNZ P3, LABEL |
| MOV PX.Y, C | Move carry to bit Y of Port X | MOV P1.0, C |
| CLR PX.Y | Clear bit Y of Port X | CLR P1.1 |
| SETB PX.Y | Set bit Y of Port X | SETB P3.2 |

## 13.4 Ports During Powerdown

Due to the 5V tolerant port structure, the output ports may have reduced $V_{OH}$ while the device is under DC conditions, i.e. in the powerdown state, and at high temperature. It is recommended that output ports be placed in a low state, if possible, prior to entering powerdown, or else external pull-ups can be used to maintain the output at a higher voltage.

## 13.5 Port Alternate Functions

Most general-purpose digital I/O pins of the AT89LP216 share functionality with the various I/Os needed for the peripheral units. Table 13-5 lists the alternate functions of the port pins. Alternate functions are connected to the pins in a logic AND fashion. In order to enable the alternate function on a port pin, that pin must have a “1” in its corresponding port register bit, otherwise the input/output will always be “0”. Furthermore, each pin must be configured for the correct input/output mode as required by its peripheral before it may be used as such. Table 13-4 shows how to configure a generic pin for use with an alternate function.

**Table 13-4.** Alternate Function Configurations for Pin y of Port x

| PxM0.y | PxM1.y | Px.y | I/O Mode |
|---|---|---|---|
| 0 | 0 | 1 | bidirectional (internal pull-up) |
| 0 | 1 | 1 | output |
| 1 | 0 | X | input |
| 1 | 1 | 1 | bidirectional (external pull-up) |

**Table 13-5.** Port Pin Alternate Functions

| Port Pin | Configuration Bits | | Alternate Function | Notes |
|---|---|---|---|---|
| | PxM0.y | PxM1.y | | |
| P1.0 | P1M0.0 | P1M1.0 | AIN0 | input-only |
| | | | GPI0 | |
| P1.1 | P1M0.1 | P1M1.1 | AIN1 | input-only |
| | | | GPI1 | |
| P1.2 | P1M0.2 | P1M1.2 | GPI2 | |
| P1.3 | P1M0.3 | P1M1.3 | GPI3 | $\overline{\text{RST}}$ must be disabled |
| P1.4 | P1M0.4 | P1M1.4 | SS | Resets to quasi-bidirectional |
| | | | GPI4 | |
| P1.5 | P1M0.5 | P1M1.5 | MOSI | |
| | | | GPI5 | |
| P1.6 | P1M0.6 | P1M1.6 | MISO | |
| | | | GPI6 | |
| P1.7 | P1M0.7 | P1M1.7 | SCK | |
| | | | GPI7 | |
| P3.0 | P3M0.0 | P3M1.0 | RXD | |
| P3.1 | P3M0.1 | P3M1.1 | TXD | |

**Table 13-5.** Port Pin Alternate Functions (Continued)

| Port Pin | Configuration Bits | | Alternate Function | Notes |
|---|---|---|---|---|
| | PxM0.y | PxM1.y | | |
| P3.2 | P3M0.2 | P3M1.2 | INT0 | Internal RC Oscillator Only |
| P3.3 | P3M0.3 | P3M1.3 | INT1 | Internal RC Oscillator or External Clock Source Only |
| | | | CLKOUT | |
| P3.4 | P3M0.4 | P3M1.4 | T0 | |
| P3.5 | P3M0.5 | P3M1.5 | T1 | |
| P3.6 | not configurable | | CMPOUT | Pin is tied to comparator output |

# 14. Enhanced Timer/Counters

The AT89LP216 has two 16-bit Timer/Counter registers: Timer 0 and Timer 1. As a Timer, the register increase every clock cycle by default. Thus, the register counts clock cycles. Since a clock cycle consists of one oscillator period, the count rate is equal to the oscillator frequency. The timer rate can be prescaled by a value between 1 and 16 using the Timer Prescaler (see Table 9-2 on page 13). Both Timers share the same prescaler.

As a Counter, the register is incremented in response to a l-to-0 transition at its corresponding input pin, T0 or T1. The external input is sampled every clock cycle. When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register during the cycle following the one in which the transition was detected. Since 2 clock cycles are required to recognize a l-to-0 transition, the maximum count rate is 1/2 of the oscillator frequency. There are no restrictions on the duty cycle of the input signal, but it should be held for at least one full clock cycle to ensure that a given level is sampled at least once before it changes.

Furthermore, the Timer or Counter functions for Timer 0 and Timer 1 have four operating modes: variable width timer, 16-bit auto-reload timer, 8-bit auto-reload timer, and split timer. The control bits C/T in the Special Function Register TMOD select the Timer or Counter function. The bit pairs (M1, M0) in TMOD select the operating modes.

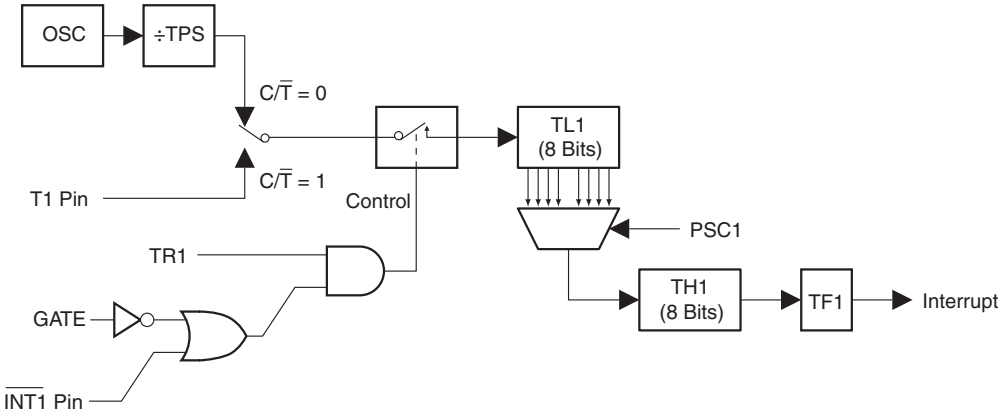
## 14.1 Mode 0 – Variable Width Timer/Counter

Both Timers in Mode 0 are 8-bit Counters with a variable prescaler. The prescaler may vary from 1 to 8 bits depending on the PSC bits in TCONB, giving the timer a range of 9 to 16 bits. By default the timer is configured as a 13-bit timer compatible to Mode 0 in the standard 8051. Figure 14-1 shows the Mode 0 operation as it applies to Timer 1 in 13-bit mode. As the count rolls over from all “1”s to all “0”s, it sets the Timer interrupt flag TF1. The counter input is enabled to the Timer when TR1 = 1 and either GATE = 0 or $\overline{INT1}$ = 1. Setting GATE = 1 allows the Timer to be controlled by external input $\overline{INT1}$, to facilitate pulse width measurements. TR1 is a control bit in the Special Function Register TCON. GATE is in TMOD. The 13-bit register consists of all 8 bits of TH1 and the lower 5 bits of TL1. The upper 3 bits of TL1 are indeterminate and should be ignored. Setting the run flag (TR1) does not clear the registers.

$$\text{Mode 0:} \qquad \text{Time-out Period} = \frac{256 \times 2^{PSC0+1}}{\text{Oscillator Frequency}} \times (TPS + 1)$$

Note: RH1/RL1 are not required by Timer 1 during Mode 0 and may be used as temporary storage registers.

**Figure 14-1.** Timer/Counter 1 Mode 0: Variable Width Counter



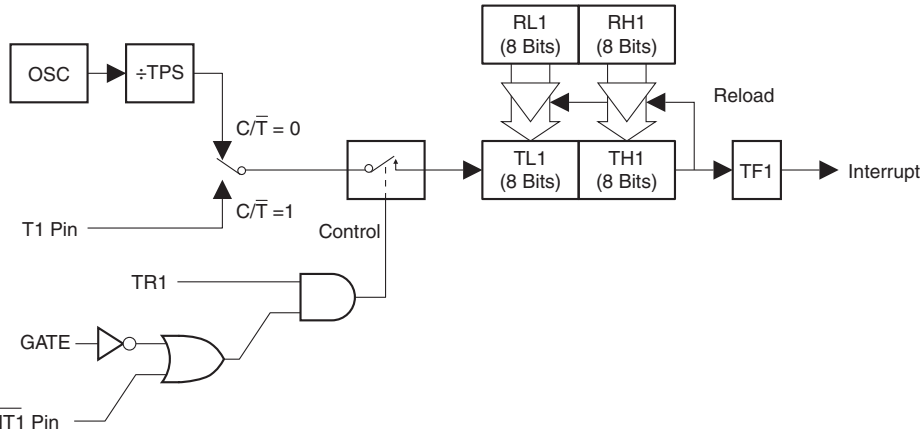

Mode 0 operation is the same for Timer 0 as for Timer 1, except that TR0, TF0 and $\overline{INT0}$ replace the corresponding Timer 1 signals in Figure 14-1. There are two different GATE bits, one for Timer 1 (TMOD.7) and one for Timer 0 (TMOD.3). The $\overline{INT0}$ and $\overline{INT1}$ pins are shared with the XTAL oscillator. They may only be used for the GATE function when using the internal RC oscillator as the system clock.

## 14.2 Mode 1 – 16-bit Auto-Reload Timer/Counter

In Mode 1 the Timers are configured for 16-bit auto-reload. The Timer register is run with all 16 bits. The 16-bit reload value is stored in the high and low reload registers (RH1/RL1). The clock is applied to the combined high and low timer registers (TH1/TL1). As clock pulses are received, the timer counts up: 0000H, 0001H, 0002H, etc. An overflow occurs on the FFFFH-to-0000H transition, upon which the timer register is reloaded with the value from RH1/RL1 and the overflow flag bit in TCON is set. See Figure 14-2. The reload registers default to 0000H, which gives the full 16-bit timer period compatible with the standard 8051. Mode 1 operation is the same for Timer/Counter 0.

$$\text{Mode 1:} \qquad \text{Time-out Period} = \frac{(65536 - \{RH0, RL0\})}{\text{Oscillator Frequency}} \times (TPS + 1)$$

**Figure 14-2.** Timer/Counter 1 Mode 1: 16-bit Auto-reload

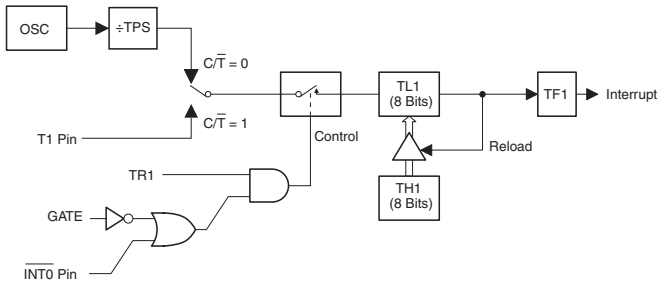


## 14.3 Mode 2 – 8-bit Auto-reload Timer/Counter

Mode 2 configures the Timer register as an 8-bit Counter (TL1) with automatic reload, as shown in Figure 14-3. Overflow from TL1 not only sets TF1, but also reloads TL1 with the contents of TH1, which is preset by software. The reload leaves TH1 unchanged. Mode 2 operation is the same for Timer/Counter 0.

$$\text{Mode 2:} \qquad \text{Time-out Period} = \frac{(256 - TH0)}{\text{Oscillator Frequency}} \times (TPS + 1)$$

**Figure 14-3.** Timer/Counter 1 Mode 2: 8-bit Auto-reload



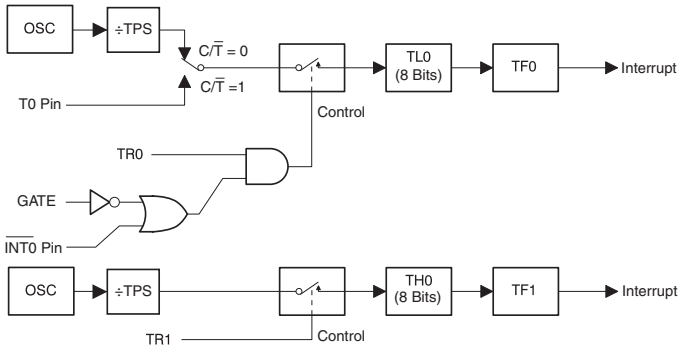

Note: RH1/RL1 are not required by Timer 1 during Mode 2 and may be used as temporary storage registers.

## 14.4 Mode 3 – 8-bit Split Timer

Timer 1 in Mode 3 simply holds its count. The effect is the same as setting TR1 = 0. Timer 0 in Mode 3 establishes TL0 and TH0 as two separate counters. The logic for Mode 3 on Timer 0 is shown in Figure 14-4. TL0 uses the Timer 0 control bits: C/T, GATE, TR0, $\overline{INT0}$, and TF0. TH0 is locked into a timer function (counting machine cycles) and takes over the use of TR1 and TF1 from Timer 1. Thus, TH0 now controls the Timer 1 interrupt. While Timer 0 is in Mode 3, Timer 1 will still obey its settings in TMOD but cannot generate an interrupt.

Mode 3 is for applications requiring an extra 8-bit timer or counter. With Timer 0 in Mode 3, the AT89LP216 can appear to have three Timer/Counters. When Timer 0 is in Mode 3, Timer 1 can be turned on and off by switching it out of and into its own Mode 3. In this case, Timer 1 can still be used by the serial port as a baud rate generator or in any application not requiring an interrupt.

**Figure 14-4.** Timer/Counter 0 Mode 3: Two 8-bit Counters




Note: RH0/RL0 are not required by Timer 0 during Mode 3 and may be used as temporary storage registers.

**Table 14-1.** TCON – Timer/Counter Control Register

TCON = 88H — Reset Value = 0000 0000B

Bit Addressable

| | TF1 | TR1 | TF0 | TR0 | IE1 | IT1 | IE0 | IT0 |
|---|---|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

| Symbol | Function |
|---|---|
| TF1 | Timer 1 overflow flag. Set by hardware on Timer/Counter overflow. Cleared by hardware when the processor vectors to interrupt routine. |
| TR1 | Timer 1 run control bit. Set/cleared by software to turn Timer/Counter on/off. |
| TF0 | Timer 0 overflow flag. Set by hardware on Timer/Counter overflow. Cleared by hardware when the processor vectors to interrupt routine. |
| TR0 | Timer 0 run control bit. Set/cleared by software to turn Timer/Counter on/off. |
| IE1 | Interrupt 1 edge flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed. |
| IT1 | Interrupt 1 type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupts. |
| IE0 | Interrupt 0 edge flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed. |
| IT0 | Interrupt 0 type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupts. |

**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.<br>8-bit Timer/Counter THx with TLx as 1 - 8-bit prescaler. |
| 0 | 1 | 1 | 16-bit auto-reload mode.<br>16-bit Timer/Counters THx and TLx are cascaded; there is no prescaler. |
| 1 | 0 | 2 | 8-bit auto reload.<br>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.<br>(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 |

**Table 14-3.** TCONB – Timer/Counter Control Register B

TCONB = 91H — Reset Value = 0010 0100B

Not Bit Addressable

| | PWM1EN | PWM0EN | PSC12 | PSC11 | PSC10 | PSC02 | PSC01 | PSC00 |
|---|---|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

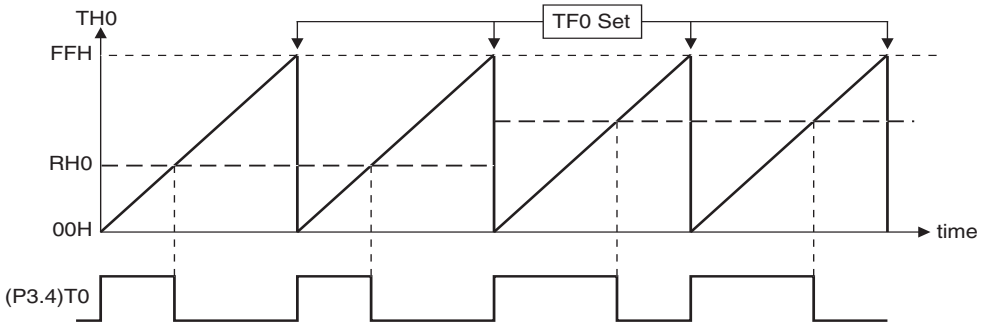
| Symbol | Function |
|---|---|
| PWM1EN | Configures Timer 1 for Pulse Width Modulation output on T1 (P3.5). |
| PWM0EN | Configures Timer 0 for Pulse Width Modulation output on T0 (P3.4). |
| PSC12<br>PSC11<br>PSC10 | Prescaler for Timer 1 Mode 0. The number of active bits in TL1 equals PSC1 + 1. After reset PSC1 = 100B which enables 5 bits of TL1 for compatibility with the 13-bit Mode 0 in AT89S2051. |
| PSC02<br>PSC01<br>PSC00 | Prescaler for Timer 0 Mode 0. The number of active bits in TL0 equals PSC0 + 1. After reset PSC0 = 100B which enables 5 bits of TL0 for compatibility with the 13-bit Mode 0 in AT89C52. |

## 14.5 Pulse Width Modulation

On the AT89LP216, Timer 0 and Timer 1 may be independently configured as 8-bit asymmetrical (edge-aligned) pulse width modulators (PWM) by setting the PWM0EN or PWM1EN bits in TCONB, respectively. In PWM Mode the generated waveform is output on the timer's input pin, T0 or T1. Therefore, C/$\overline{T}$ must be set to “0” when in PWM mode. and the T0 (P3.4) and T1 (P3.5) must be configured in an output mode. The Timer Overflow Flags and Interrupts will continue to function while in PWM Mode and Timer 1 may still generate the baud rate for the UART. Each PWM channel has four modes selected by the mode bits in TMOD.

An example waveform for Timer 0 in PWM Mode 0 is shown in Figure 14-5. TH0 acts as an 8-bit counter while RH0 stores the 8-bit compare value. When TH0 is 00H the PWM output is set high. When the TH0 count reaches the value stored in RH0 the PWM output is set low. Therefore, the pulse width is proportional to the value in RH0. To prevent glitches, writes to RH0 only take effect on the FFH to 00H overflow of TH0. Setting RH0 to 00H will keep the PWM output low.

**Figure 14-5.** Asymmetrical Pulse Width Modulation



### 14.5.1 Mode 0 – 8-bit PWM with 8-bit Logarithmic Prescaler

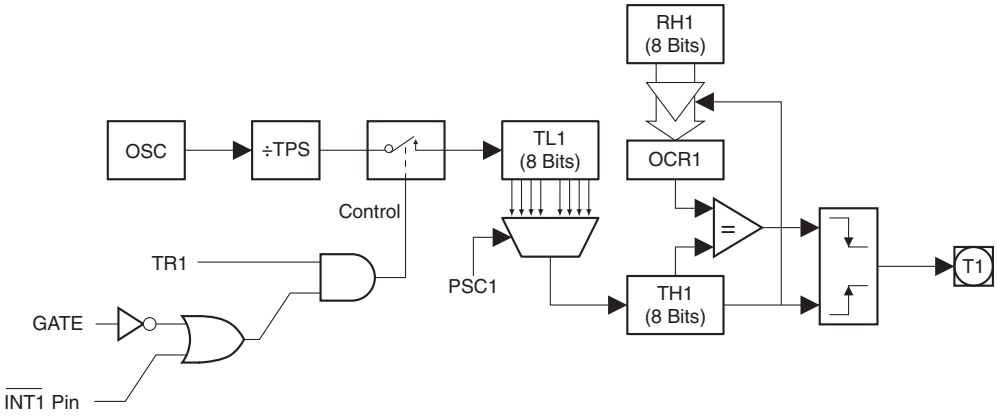
In Mode 0, TLx acts as a logarithmic prescaler driving 8-bit counter THx (see Figure 14-6). The PSCx bits in TCONB control the prescaler value. On THx overflow, the duty cycle value in RHx is transferred to OCRx and the output pin is set high. When the count in THx matches OCRx, the output pin is cleared low. The following formulas give the output frequency and duty cycle for Timer 0 in PWM Mode 0. Timer 1 in PWM Mode 0 is identical to Timer 0.

Mode 0:

$$f_{out} = \frac{\text{Oscillator Frequency}}{256 \times 2^{PSC0+1}} \times \frac{1}{TPS+1}$$

$$\text{Duty Cycle \%} = 100 \times \frac{RH0}{256}$$

**Figure 14-6.** Timer/Counter 1 PWM Mode 0




### 14.5.2 Mode 1 – 8-bit PWM with 8-bit Linear Prescaler

In Mode 1, TLx provides linear prescaling with an 8-bit auto-reload from RLx (see Figure 14-7 on page 34). On TLx overflow, TLx is loaded with the value of RLx. THx acts as an 8-bit counter. On THx overflow, the duty cycle value in RHx is transferred to OCRx and the output pin is set high. When the count in THx matches OCRx, the output pin is cleared low. The following formulas give the output frequency and duty cycle for Timer 0 in PWM Mode 1. Timer 1 in PWM Mode 1 is identical to Timer 0.

Mode 1:

$$f_{out} = \frac{\text{Oscillator Frequency}}{256 \times (256 - RL0)} \times \frac{1}{TPS+1}$$

$$\text{Duty Cycle \%} = 100 \times \frac{RH0}{256}$$

### 14.5.3 Mode 2 – 8-bit Frequency Generator

Timer 0 in PWM Mode 2 functions as an 8-bit auto-reload timer, the same as normal Mode 2, with the exception that the output pin T0 is toggled at every TL0 overflow (see Figures 14-8 and 14-9). Timer 1 in PWM Mode 2 is identical to Timer 0. PWM Mode 2 can be used to output a square wave of varying frequency. THx acts as an 8-bit counter. The following formula gives the output frequency for Timer 0 in PWM Mode 2.

Mode 2:

$$f_{out} = \frac{\text{Oscillator Frequency}}{2 \times (256 - TH0)} \times \frac{1}{TPS+1}$$

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

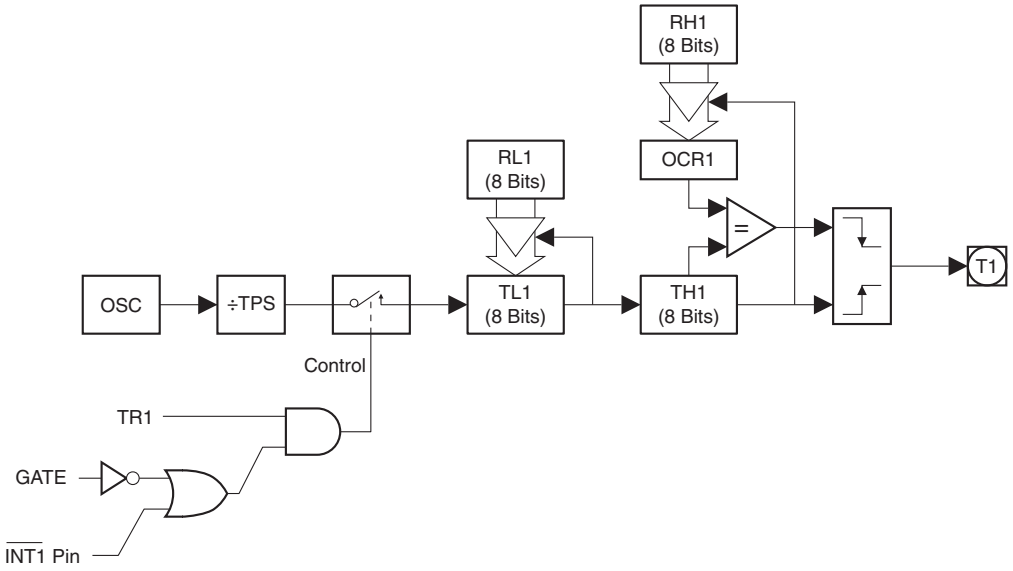
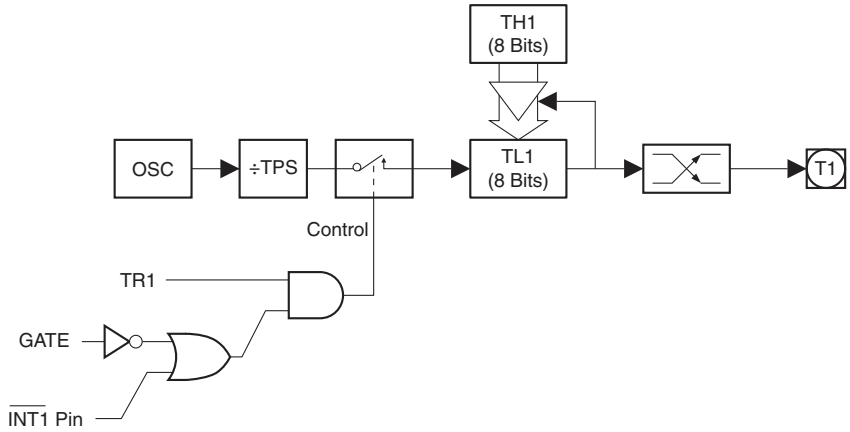
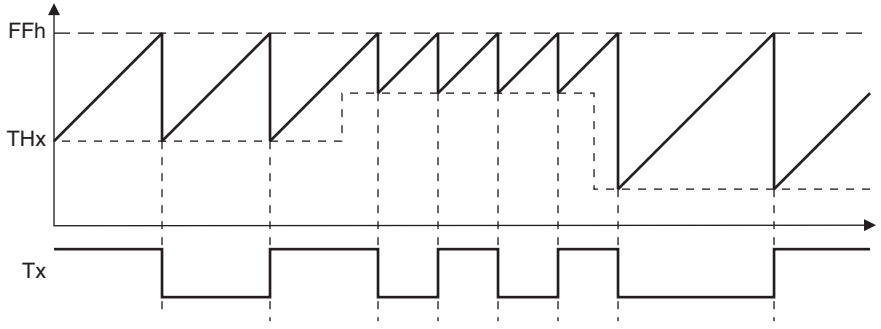



**Figure 14-8.** Timer/Counter 1 PWM Mode 2




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



### 14.5.4 Mode 3 – Split 8-bit PWM

Timer 1 in PWM Mode 3 simply holds its count. The effect is the same as setting TR1 = 0. Timer 0 in PWM Mode 3 establishes TL0 and TH0 as two separate PWM counters in a manner similar to normal Mode 3. PWM Mode 3 on Timer 0 is shown in Figure 14-10. Only the Timer Prescaler is available to change the output frequency during PWM Mode 3. TL0 can use the Timer 0 control bits: GATE, TR0, $\overline{\text{INT0}}$, PWM0EN and TF0. TH0 is locked into a timer function and uses TR1, PWM1EN and TF1. RL0 provides the duty cycle for TL0 and RH0 provides the duty cycle for TH0.

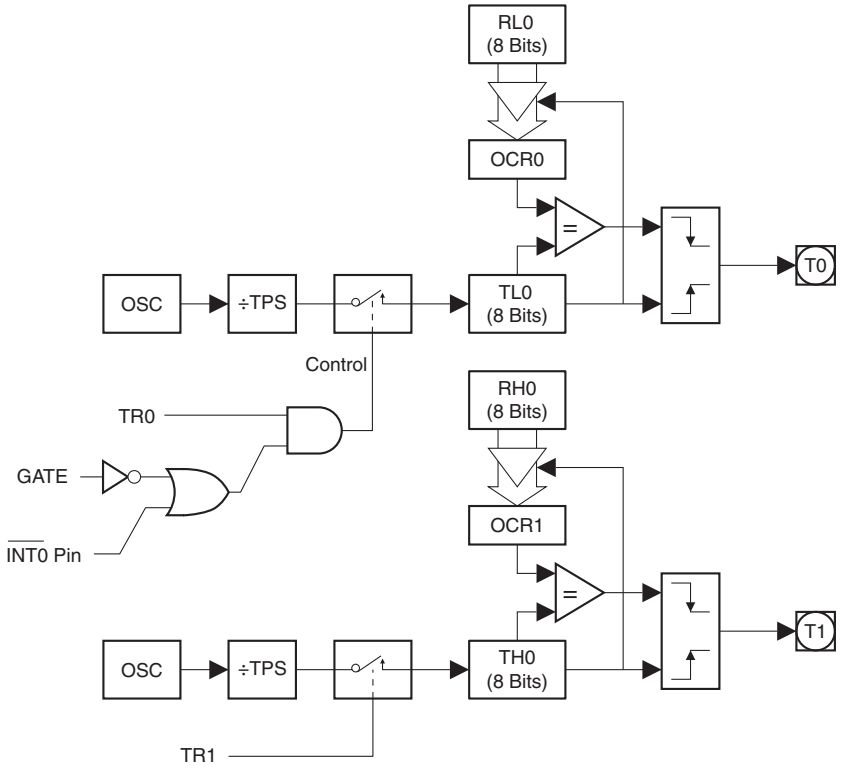
PWM Mode 3 is for applications requiring a single PWM channel and two timers, or two PWM channels and an extra timer or counter. With Timer 0 in PWM Mode 3, the AT89LP216 can appear to have three Timer/Counters. When Timer 0 is in PWM Mode 3, Timer 1 can be turned on and off by switching it out of and into its own Mode 3. In this case, Timer 1 can still be used by the serial port as a baud rate generator or in any application not requiring an interrupt. The following formulas give the output frequency and duty cycle for Timer 0 in PWM Mode 3.

$$\text{Mode 3:} \qquad f_{out} = \frac{\text{Oscillator Frequency}}{256} \times \frac{1}{\text{TPS} + 1}$$

$$\text{Mode 3, T0:} \qquad \text{Duty Cycle \%} = 100 \times \frac{\text{RL0}}{256}$$

$$\text{Mode 3, T1:} \qquad \text{Duty Cycle \%} = 100 \times \frac{\text{RH0}}{256}$$

**Figure 14-10.** Timer/Counter 0 PWM Mode 3



# 15. External Interrupts

When the AT89LP216 is configured to use the internal RC Oscillator, XTAL1 and XTAL2 may be used as the $\overline{INT0}$ and $\overline{INT1}$ external interrupt sources. When the external clock source is used, XTAL2 is available as INT1. Neither interrupt is available in crystal oscillator mode.The external interrupts can be programmed to be level-activated or transition-activated by setting or clearing bit IT1 or IT0 in Register TCON. If ITx = 0, external interrupt x is triggered by a detected low at the $\overline{INTx}$ pin. If ITx = 1, external interrupt x is edge-triggered. In this mode if successive samples of the $\overline{INTx}$ pin show a high in one cycle and a low in the next cycle, interrupt request flag IEx in TCON is set. Flag bit IEx then requests the interrupt. Since the external interrupt pins are sampled once each clock cycle, an input high or low should hold for at least 2 oscillator periods to ensure sampling. If the external interrupt is transition-activated, the external source has to hold the request pin high for at least two clock cycles, and then hold it low for at least two clock cycles to ensure that the transition is seen so that interrupt request flag IEx will be set. IEx will be automatically cleared by the CPU when the service routine is called if generated in edge-triggered mode. If the external interrupt is level-activated, the external source has to hold the request active until the requested interrupt is actually generated. Then the external source must deactivate the request before the interrupt service routine is completed, or else another interrupt will be generated.

# 16. General-purpose Interrupts

The General-purpose Interrupt (GPI) function provides 8 configurable external interrupts on Port 1. Each port pin can detect high/low levels or positive/negative edges. The GPIEN register select which bits of Port 1 are enabled to generate an interrupt. The GPMOD and GPLS registers determine the mode for each individual pin. GPMOD selects between level-sensitive and edge-triggered mode. GPLS selects between high/low in level mode and positive/negative in edge mode. The pins of Port 1 are sampled every clock cycle. In level-sensitive mode, a valid level must appear in two successive samples before generating the interrupt. In edge-triggered mode, a transition will be detected if the value changes from one sample to the next. When an interrupt condition on a pin is detected, and that pin is enabled, the appropriate flag in the GPIF register is set. The flags in GPIF must be cleared by software.

**Table 16-1.** GPMOD – General-purpose Interrupt Mode Register

GPMOD = 9AH — Reset Value = 0000 0000B

Not Bit Addressable

| | GPMOD7 | GPMOD6 | GPMOD5 | GPMOD4 | GPMOD3 | GPMOD2 | GPMOD1 | GPMOD0 |
|---|---|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

GPMOD.x 0 = level-sensitive interrupt for P1.x

1 = edge-triggered interrupt for P1.x

**Table 16-2.** GPLS – General-purpose Interrupt Level Select Register

GPLS = 9BH — Reset Value = 0000 0000B

Not Bit Addressable

| | GPLS7 | GPLS6 | GPLS5 | GPLS4 | GPLS3 | GPLS2 | GPLS1 | GPLS0 |
|---|---|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

GPMOD.x 0 = detect low level or negative edge on P1.x

1 = detect high level or positive edge on P1.x

**Table 16-3.** GPIEN – General-purpose Interrupt Enable Register

GPIEN = 9CH — Reset Value = 0000 0000B

Not Bit Addressable

| | GPIEN7 | GPIEN6 | GPIEN5 | GPIEN4 | GPIEN3 | GPIEN2 | GPIEN1 | GPIEN0 |
|---|---|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

GPIEN.x 0 = interrupt for P1.x disabled

1 = interrupt for P1.x enabled

**Table 16-4.** GPIF – General-purpose Interrupt Flag Register

GPIF = 9DH — Reset Value = 0000 0000B

Not Bit Addressable

| | GPIF7 | GPIF6 | GPIF5 | GPIF4 | GPIF3 | GPIF2 | GPIF1 | GPIF0 |
|---|---|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

GPIF.x 0 = interrupt on P1.x inactive

1 = interrupt on P1.x active. Must be cleared by software.

# 17. Serial Interface

The serial interface on the AT89LP216 implements a Universal Asynchronous Receiver/Transmitter (UART). The UART has the following features:

- Full Duplex Operation
- 8 or 9 Data Bits
- Framing Error Detection
- Multiprocessor Communication Mode with Automatic Address Recognition
- Baud Rate Generator Using Timer 1
- Interrupt on Receive Buffer Full or Transmission Complete

The serial interface is full duplex, which means it can transmit and receive simultaneously. It is also receive-buffered, which means it can begin receiving a second byte before a previously received byte has been read from the receive register. (However, if the first byte still has not been read when reception of the second byte is complete, one of the bytes will be lost.) The serial port receive and transmit registers are both accessed at the Special Function Register SBUF. Writing to SBUF loads the transmit register, and reading SBUF accesses a physically separate receive register. The serial port can operate in the following four modes.

- **Mode 0:** Serial data enters and exits through RXD. TXD outputs the shift clock. Eight data bits are transmitted/received, with the LSB first. The baud rate is fixed at 1/2 the oscillator frequency.
- **Mode 1:** 10 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 the Special Function Register SCON. The baud rate is variable based on Timer 1.
- **Mode 2:** 11 bits are transmitted (through TXD) or received (through RXD): a start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1). On transmit, the 9th data bit (TB8 in SCON) can be assigned the value of “0” or “1”. For example, the parity bit (P, in the PSW) can be moved into TB8. On receive, the 9th data bit goes into RB8 in the Special Function Register SCON, while the stop bit is ignored. The baud rate is programmable to either 1/16 or 1/32 the oscillator frequency.
- **Mode 3:** 11 bits are transmitted (through TXD) or received (through RXD): a start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1). In fact, Mode 3 is the same as Mode 2 in all respects except the baud rate, which is variable based on Timer 1 in Mode 3.

In all four modes, transmission is initiated by any instruction that uses SBUF as a destination register. Reception is initiated in Mode 0 by the condition RI = 0 and REN = 1. Reception is initiated in the other modes by the incoming start bit if REN = 1.

## 17.1 Multiprocessor Communications

Modes 2 and 3 have a special provision for multiprocessor communications. In these modes, 9 data bits are received, followed by a stop bit. The 9th bit goes into RB8. Then comes a stop bit. The port can be programmed such that when the stop bit is received, the serial port interrupt is activated only if RB8 = 1. This feature is enabled by setting bit SM2 in SCON.

The following example shows how to use the serial interrupt for multiprocessor communications. When the master processor must transmit a block of data to one of several slaves, it first sends out an address byte that identifies the target slave. An address byte differs from a data byte in that the 9th bit is “1” in an address byte and “0” in a data byte. With SM2 = 1, no slave is

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<br>**SM0** / **SM1** / **Mode** / **Description** / **Baud Rate**[2]<br>0 / 0 / 0 / shift register / $f_{osc}/2$<br>0 / 1 / 1 / 8-bit UART / variable (Timer 1)<br>1 / 0 / 2 / 9-bit UART / $f_{osc}/32$ or $f_{osc}/16$<br>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. Baud Rates

The baud rate in Mode 0 is fixed as shown in the following equation:

$$\text{Mode 0 Baud Rate} = \frac{\text{Oscillator Frequency}}{2}$$

The baud rate in Mode 2 depends on the value of the SMOD1 bit in Special Function Register PCON.7. If SMOD1 = 0 (the value on reset), the baud rate is 1/32 of the oscillator frequency. If SMOD1 = 1, the baud rate is 1/16 of the oscillator frequency, as shown in the following equation:

$$\text{Mode 2 Baud Rate} = \frac{2^{\text{SMOD1}}}{32} \times (\text{Oscillator Frequency})$$

### 18.0.1 Using Timer 1 to Generate Baud Rates

The Timer 1 overflow rate determines the baud rates in Modes 1 and 3. When Timer 1 is the baud rate generator, the baud rates are determined by the Timer 1 overflow rate and the value of SMOD1 according to the following equation:

$$\text{Modes 1, 3 Baud Rate} = \frac{2^{\text{SMOD1}}}{32} \times (\text{Timer 1 Overflow Rate})$$

The Timer 1 interrupt should be disabled in this application. The Timer itself can be configured for either timer or counter operation in any of its 3 running modes. In the most typical applications, it is configured for timer operation in auto-reload mode (high nibble of TMOD = 0010B). In this case, the baud rate is given by the following formula:

$$\text{Modes 1, 3 Baud Rate} = \frac{2^{\text{SMOD1}}}{32} \times \frac{\text{Oscillator Frequency}}{[256 - (\text{TH1})]} \times \frac{1}{\text{TPS} + 1}$$

Programmers can achieve very low baud rates with Timer 1 by configuring the Timer to run as a 16-bit auto-reload timer (high nibble of TMOD = 0001B). In this case, the baud rate is given by the following formula.

$$\text{Modes 1, 3 Baud Rate} = \frac{2^{\text{SMOD1}}}{32} \times \frac{\text{Oscillator Frequency}}{[256 - (\text{RH1,RL1})]} \times \frac{1}{\text{TPS} + 1}$$

Table 18-1 lists commonly used baud rates and how they can be obtained from Timer 1.

**Table 18-1.** Commonly Used Baud Rates Generated by Timer 1 (TPS = 0000B)

| Baud Rate | $f_{osc}$ (MHz) | SMOD1 | Timer 1 | | |
|---|---|---|---|---|---|
| | | | C/$\overline{T}$ | Mode | Reload Value |
| Mode 0: 1 MHz | 2 | X | X | X | X |
| Mode 2: 375K | 12 | 0 | X | X | X |
| 62.5K | 12 | 1 | 0 | 2 | F4H |
| 19.2K | 11.059 | 1 | 0 | 2 | DCH |
| 9.6K | 11.059 | 0 | 0 | 2 | DCH |
| 4.8K | 11.059 | 0 | 0 | 2 | B8H |
| 2.4K | 11.059 | 0 | 0 | 2 | 70H |
| 1.2K | 11.059 | 0 | 0 | 1 | FEE0H |
| 137.5 | 11.986 | 0 | 0 | 1 | F55CH |
| 110 | 6 | 0 | 0 | 1 | F958H |
| 110 | 12 | 0 | 0 | 1 | F304H |

## 18.1 More About Mode 0

Serial data enters and exits through RXD. TXD outputs the shift clock. Eight data bits are transmitted/received, with the LSB first. The baud rate is fixed at 1/2 the oscillator frequency. Figure 18-1 on page 42 shows a simplified functional diagram of the serial port in Mode 0 and associated timing.

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 position of the transmit shift register and tells the TX Control Block to begin a transmission. The internal timing is such that one full machine cycle will elapse between “write to SBUF” and activation of SEND.

SEND transfers the output of the shift register to the alternate output function line of P3.0, and also transfers Shift Clock to the alternate output function line of P3.1. At the falling edge of Shift Clock the contents of the transmit shift register are shifted one position to the right.

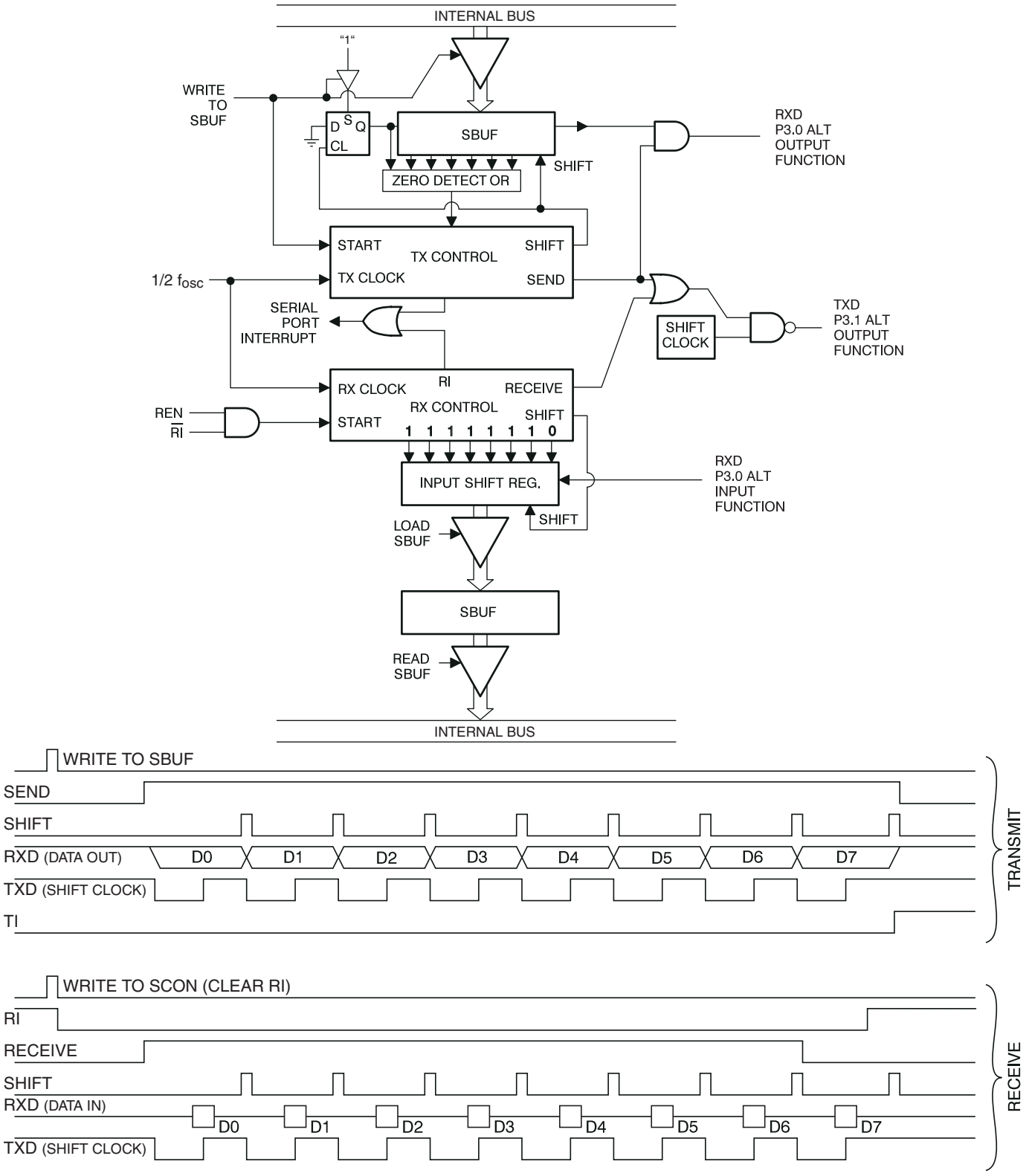
As data bits shift out to the right, “0”s come in from the left. When the MSB of the data byte is at the output position of the shift register, the “1” that was initially loaded into the 9th position is just to the left of the MSB, and all positions to the left of that contain “0”s. This condition flags the TX Control block to do one last shift, then deactivate SEND and set TI.

Reception is initiated by the condition REN = 1 and R1 = 0. At the next clock cycle, the RX Control unit writes the bits 11111110 to the receive shift register and activates RECEIVE in the next clock phase.

RECEIVE enables Shift Clock to the alternate output function line of P3.1. At the falling edge of Shift Clock the contents of the receive shift register are shifted one position to the left. The value that comes in from the right is the value that was sampled at the P3.0 pin at rising edge of Shift Clock.

As data bits come in from the right, “1”s shift out to the left. When the “0” that was initially loaded into the right-most position arrives at the left-most position in the shift register, it flags the RX Control block to do one last shift and load SBUF. Then RECEIVE is cleared and RI is set.

**Figure 18-1.** Serial Port Mode 0



## 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.

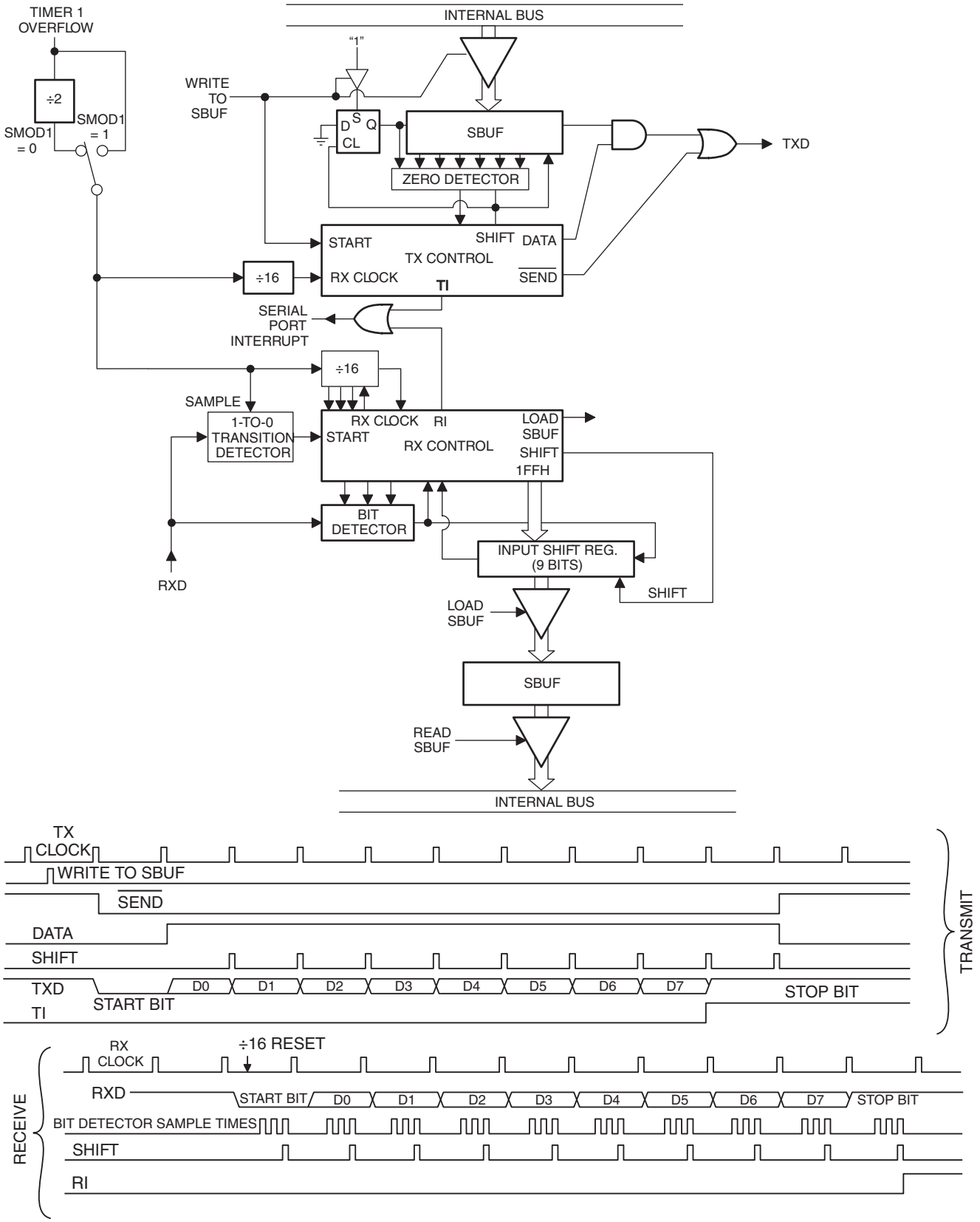
As data bits come in from the right, “1”s shift out to the left. When the start bit arrives at the left-most position in the shift register, (which is a 9-bit register in Mode 1), it flags the RX Control block to do one last shift, load SBUF and RB8, and set RI. The signal to load SBUF and RB8 and to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



## 18.3 More About Modes 2 and 3

Eleven bits are transmitted (through TXD), or received (through RXD): a start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1). On transmit, the 9th data bit (TB8) can be assigned the value of “0” or “1”. On receive, the 9th data bit goes into RB8 in SCON. The baud rate is programmable to either 1/16 or 1/32 of the oscillator frequency in Mode 2. Mode 3 may have a variable baud rate generated from Timer 1.

Figures 18-3 and 18-4 show a functional diagram of the serial port in Modes 2 and 3. The receive portion is exactly the same as in Mode 1. The transmit portion differs from Mode 1 only in the 9th bit of the transmit shift register.

Transmission is initiated by any instruction that uses SBUF as a destination register. The “write to SBUF” signal also loads TB8 into the 9th bit position of the transmit shift register and flags the TX Control unit that a transmission is requested. Transmission 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. The first shift clocks a “1” (the stop bit) into the 9th bit position of the shift register. Thereafter, only “0”s are clocked in. Thus, as data bits shift out to the right, “0”s are clocked in from the left. When TB8 is at the output position of the shift register, then the stop bit is just to the left of TB8, 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 SEND and set TI. This occurs at the 11th 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 to the input shift register.

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. 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 proves 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 leftmost position in the shift register (which in Modes 2 and 3 is a 9-bit register), 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 9th data bit = 1

If either of these conditions is not met, the received frame is irretrievably lost, and RI is not set. If both conditions are met, the received 9th data bit goes into RB8, and the first 8 data bits go into SBUF. One bit time later, whether the above conditions were met or not, the unit continues looking for a 1-to-0 transition at the RXD input.

Note that the value of the received stop bit is irrelevant to SBUF, RB8, or RI.

**Figure 18-3.** Serial Port Mode 2

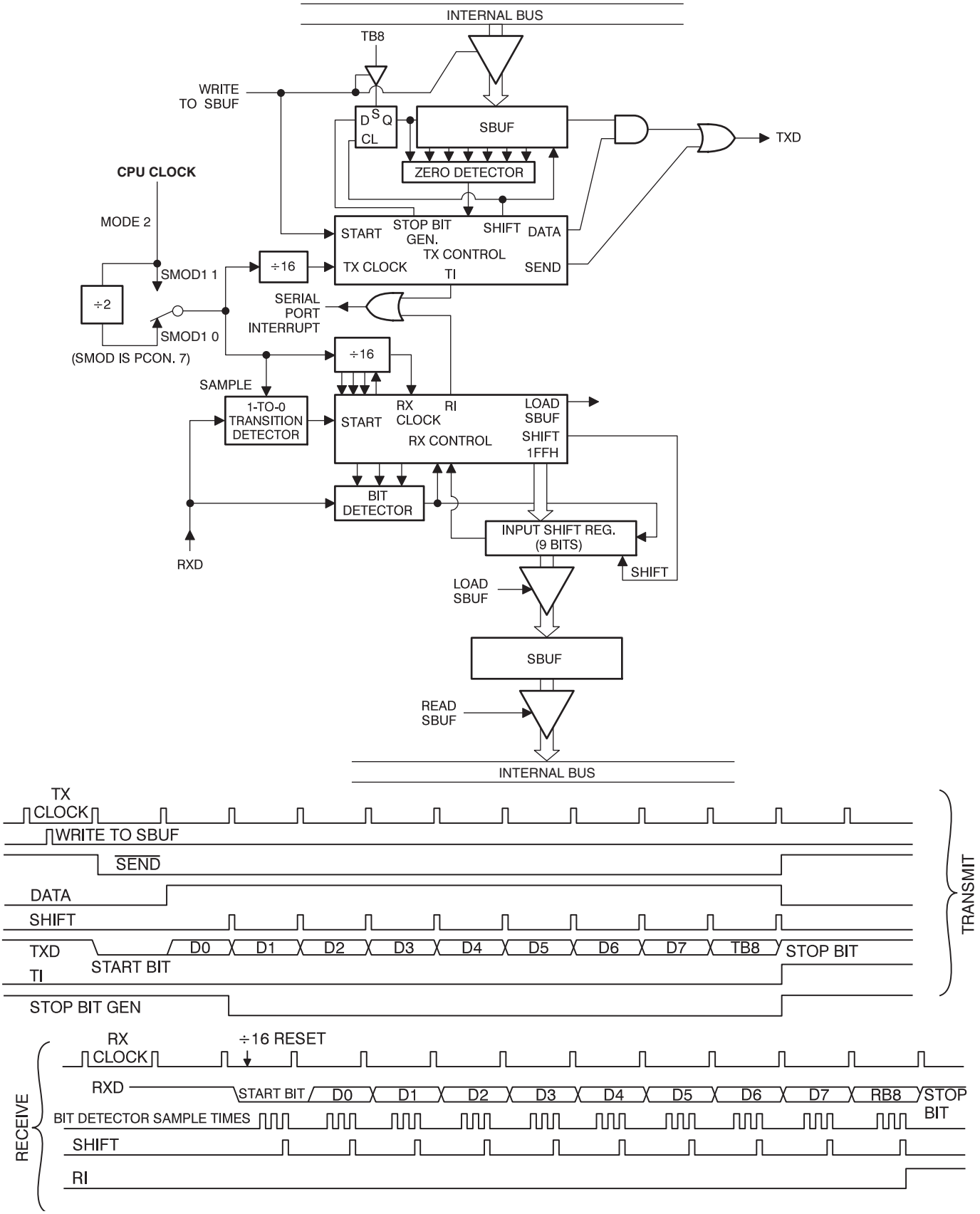
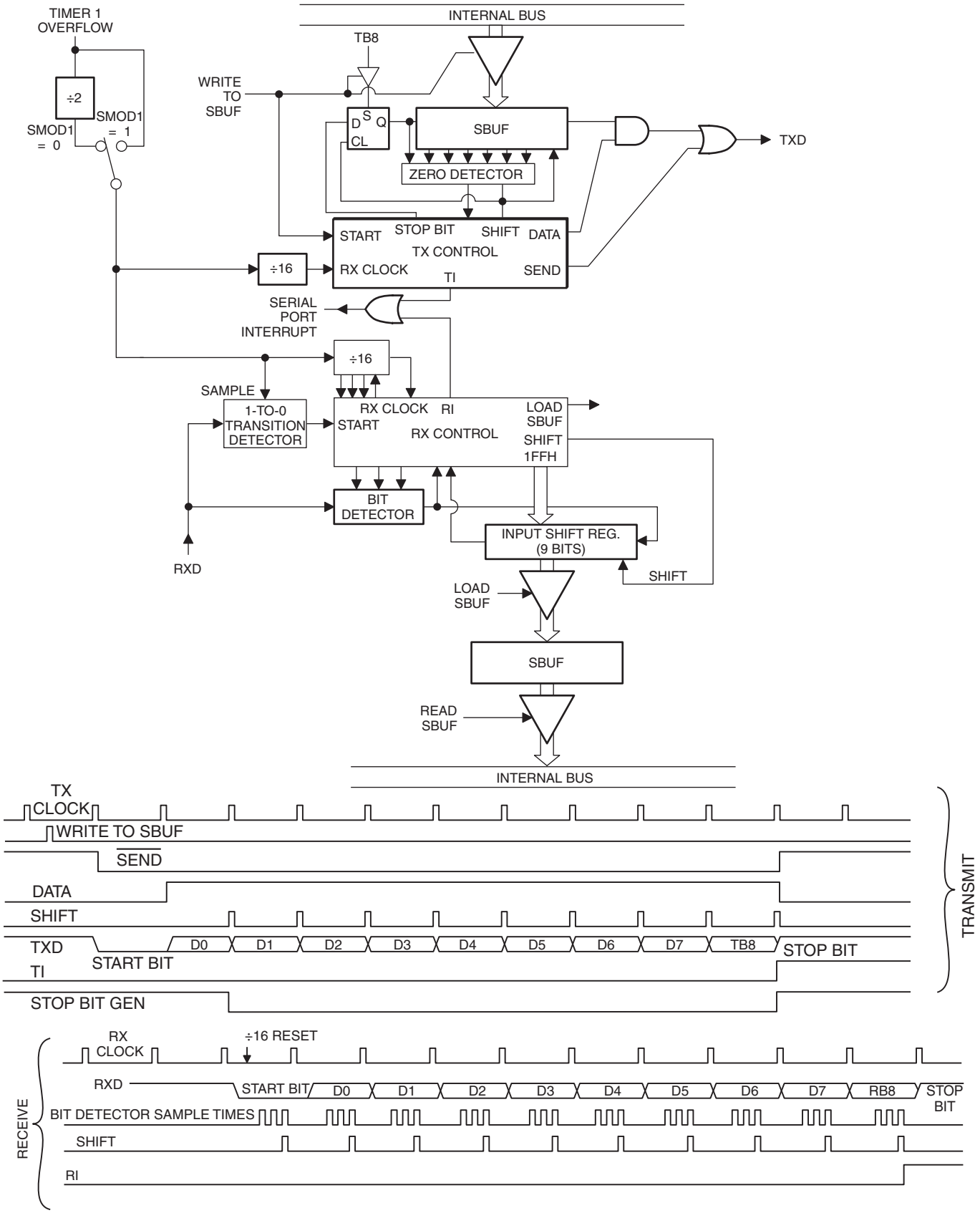


**Figure 18-4.** Serial Port Mode 3



## 18.4 Framing Error Detection

In addition to all of its usual modes, the UART can perform framing error detection by looking for missing stop bits, and automatic address recognition. When used for framing error detect, the UART looks for missing stop bits in the communication. A missing bit will set the FE bit in the SCON register. The FE bit shares the SCON.7 bit with SM0 and the function of SCON.7 is determined by PCON.6 (SMOD0). If SMOD0 is set then SCON.7 functions as FE. SCON.7 functions as SM0 when SMOD0 is cleared. When used as FE, SCON.7 can only be cleared by software. The FE bit will be set by a framing error regardless of the state of SMOD0.

## 18.5 Automatic Address Recognition

Automatic Address Recognition is a feature which allows the UART to recognize certain addresses in the serial bit stream by using hardware to make the comparisons. This feature saves a great deal of software overhead by eliminating the need for the software to examine every serial address which passes by the serial port. This feature is enabled by setting the SM2 bit in SCON. In the 9th bit UART modes, Mode 2 and Mode 3, the Receive Interrupt flag (RI) will be automatically set when the received byte contains either the “Given” address or the “Broadcast” address. The 9th bit mode requires that the 9th information bit to be a “1” to indicate that the received information is an address and not data.

The 8th bit mode is called Mode 1. In this mode the RI flag will be set if SM2 is enabled and the information received has a valid stop bit following the 8th address bits and the information is either a Given or Broadcast address.

Mode 0 is the Shift Register mode and SM2 is ignored.

Using the Automatic Address Recognition feature allows a master to selectively communicate with one or more slaves by invoking the given slave address or addresses. All of the slaves may be contacted by using the Broadcast address. Two special Function Registers are used to define the slave’s address, SADDR, and the address mask, SADEN. SADEN is used to define which bits in the SADDR are to be used and which bits are “don’t care”. The SADEN mask can be logically ANDed with the SADDR to create the “Given” address which the master will use for addressing each of the slaves. Use of the Given address allows multiple slaves to be recognized while excluding others. The following examples show the versatility of this scheme:

Slave 0

SADDR = 1100 0000

SADEN = 1111 1101

Given = 1100 00X0

Slave 1

SADDR = 1100 0000

SADEN = 1111 1110

Given = 1100 000X

In the previous example, SADDR is the same and the SADEN data is used to differentiate between the two slaves. Slave 0 requires a “0” in bit 0 and it ignores bit 1. Slave 1 requires a “0” in bit 1 and bit 0 is ignored. A unique address for slave 0 would be 1100 0010 since slave 1 requires a “0” in bit 1. A unique address for slave 1 would be 1100 0001 since a “1” in bit 0 will exclude slave 0. Both slaves can be selected at the same time by an address which has bit 0 = 0 (for slave 0) and bit 1 = 0 (for slave 1). Thus, both could be addressed with 1100 0000.

In a more complex system, the following could be used to select slaves 1 and 2 while excluding slave 0:

Slave 0 SADDR = 1100 0000

SADEN = 1111 1001

Given = 1100 0XX0

Slave 1 SADDR = 1110 0000

SADEN = 1111 1010

Given = 1110 0X0X

Slave 2 SADDR = 1110 0000

SADEN = 1111 1100

Given = 1110 00XX

In the above example, the differentiation among the 3 slaves is in the lower 3 address bits. Slave 0 requires that bit 0 = 0 and it can be uniquely addressed by 1110 0110. Slave 1 requires that bit 1 = 0 and it can be uniquely addressed by 1110 and 0101. Slave 2 requires that bit 2 = 0 and its unique address is 1110 0011. To select Slaves 0 and 1 and exclude Slave 2, use address 1110 0100, since it is necessary to make bit 2 = 1 to exclude slave 2.

The Broadcast Address for each slave is created by taking the logic OR of SADDR and SADEN. Zeros in this result are trended as don't cares. In most cases, interpreting the don't cares as ones, the broadcast address will be FF hexadecimal.

Upon reset SADDR (SFR address 0A9H) and SADEN (SFR address 0B9H) are loaded with "0"s. This produces a given address of all "don't cares" as well as a Broadcast address of all "don't cares". This effectively disables the Automatic Addressing mode and allows the microcontroller to use standard 80C51-type UART drivers which do not make use of this feature.

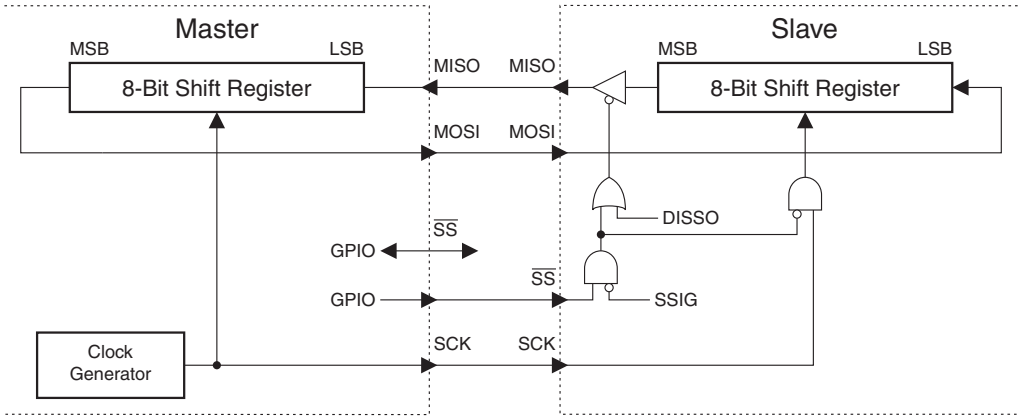
## 19. Serial Peripheral Interface

The serial peripheral interface (SPI) allows high-speed synchronous data transfer between the AT89LP216 and peripheral devices or between multiple AT89LP216 devices. The SPI features include the following:

- Full-duplex, 3-wire Synchronous Data Transfer
- Master or Slave Operation
- Maximum Bit Frequency = $f_{OSC}/4$
- LSB First or MSB First Data Transfer
- Four Programmable Bit Rates in Master Mode
- End of Transmission Interrupt Flag
- Write Collision Flag Protection
- Double-buffered Receive
- Double-buffered Transmit (Enhanced Mode Only)
- Wake up from Idle Mode (Slave Mode Only)

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.

**Table 19-1.** SPCR – SPI Control Register

SPCR Address = E9H

Reset Value = 0000 0000B

Not Bit Addressable

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | SPIE | SPE | DORD | MSTR | CPOL | CPHA | SPR1 | SPR0 |
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

| Symbol | Function |
|---|---|
| SPIE | SPI interrupt enable. This bit, in conjunction with the ES bit in the IE register, enables SPI interrupts: SPIE = 1 and ES = 1 enable SPI interrupts. SPIE = 0 disables SPI interrupts. |
| SPE | SPI enable. SPI = 1 enables the SPI channel and connects $\overline{SS}$, MOSI, MISO and SCK to pins P1.4, P1.5, P1.6, and P1.7. SPI = 0 disables the SPI channel. |
| DORD | Data order. DORD = 1 selects LSB first data transmission. DORD = 0 selects MSB first data transmission. |
| MSTR | Master/slave select. MSTR = 1 selects Master SPI mode. MSTR = 0 selects slave SPI mode. |
| CPOL | Clock polarity. When CPOL = 1, SCK is high when idle. When CPOL = 0, SCK of the master device is low when not transmitting. Please refer to figure on SPI clock phase and polarity control. |
| CPHA | Clock phase. The CPHA bit together with the CPOL bit controls the clock and data relationship between master and slave. Please refer to figure on SPI clock phase and polarity control. |
| SPR0<br>SPR1 | SPI clock rate select. These two bits control the SCK rate of the device configured as master. SPR1 and SPR0 have no effect on the slave. The relationship between SCK and the oscillator frequency, $F_{OSC.}$, is as follows:<br>SPR1 SPR0 SCK<br>0 0 $f_{osc}/4$<br>0 1 $f_{osc}/8$<br>1 0 $f_{osc}/32$<br>1 1 $f_{osc}/64$ |

Notes: 1. Set up the clock mode before enabling the SPI: set all bits needed in SPCR except the SPE bit, then set SPE.
2. Enable the master SPI prior to the slave device.
3. Slave echoes master on the next Tx if not loaded with new data.

**Table 19-2.** SPDR – SPI Data Register

SPDR Address = EAH

Not Bit Addressable

Reset Value = 00H (after cold reset)
unchanged (after warm reset)

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | SPD7 | SPD6 | SPD5 | SPD4 | SPD3 | SPD2 | SPD1 | SPD0 |
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

**Table 19-3.** SPSR – SPI Status Register

SPSR Address = E8H — Reset Value = 000X X000B

Not Bit Addressable

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SPIF | WCOL | LDEN | – | – | SSIG | DISSO | ENH |

| Symbol | Function |
|---|---|
| SPIF | SP interrupt flag. When a serial transfer is complete, the SPIF bit is set and an interrupt is generated if SPIE = 1 and ES = 1. The SPIF bit is cleared by reading the SPI status register followed by reading/writing the SPI data register. |
| WCOL | When ENH = 0: Write collision flag. The WCOL bit is set if the SPI data register is written during a data transfer. During data transfer, the result of reading the SPDR register may be incorrect, and writing to it has no effect. The WCOL bit (and the SPIF bit) are cleared by reading the SPI status register followed by reading/writing the SPI data register.<br>When ENH = 1: WCOL works in Enhanced mode as Tx Buffer Full. Writing during WCOL = 1 in enhanced mode will overwrite the waiting data already present in the Tx Buffer. In this mode, WCOL is no longer reset by the SPIF reset but is reset when the write buffer has been unloaded into the serial shift register. |
| LDEN | Load enable for the Tx buffer in enhanced SPI mode.<br>When ENH is set, it is safe to load the Tx Buffer while LDEN = 1 and WCOL = 0. LDEN is high during bits 0 - 3 and is low during bits 4 - 7 of the SPI serial byte transmission time frame. |
| SSIG | Slave Select Ignore. If SSIG = 0, the SPI will only operate in slave mode if $\overline{SS}$ (P1.4) is pulled low. When SSIG = 1, the SPI ignores $\overline{SS}$ in slave mode and is active whenever SPE (SPCR.6) is set. P1.4 may be used as a regular I/O pin when SSIG = 1. |
| DISSO | Disable slave output bit.<br>When set, this bit causes the MISO pin to be tri-stated so more than one slave device can share the same interface with a single master. Normally, the first byte in a transmission could be the slave address and only the selected slave should clear its DISSO bit. |
| ENH | Enhanced SPI mode select bit. When ENH = 0, SPI is in normal mode, i.e. without write double buffering.<br>When ENH = 1, SPI is in enhanced mode with write double buffering. The Tx buffer shares the same address with the SPDR register. |

**Figure 19-2.** SPI Shift Register Diagram

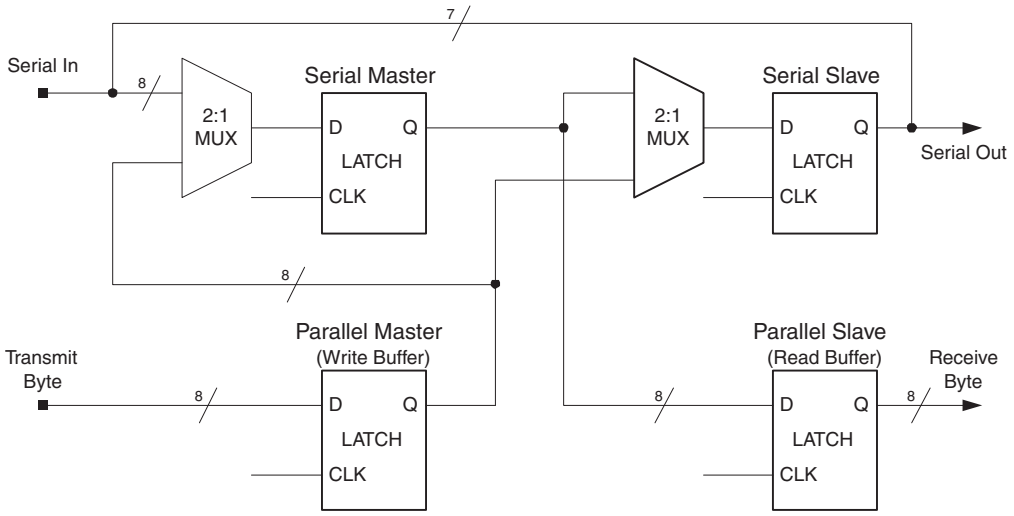
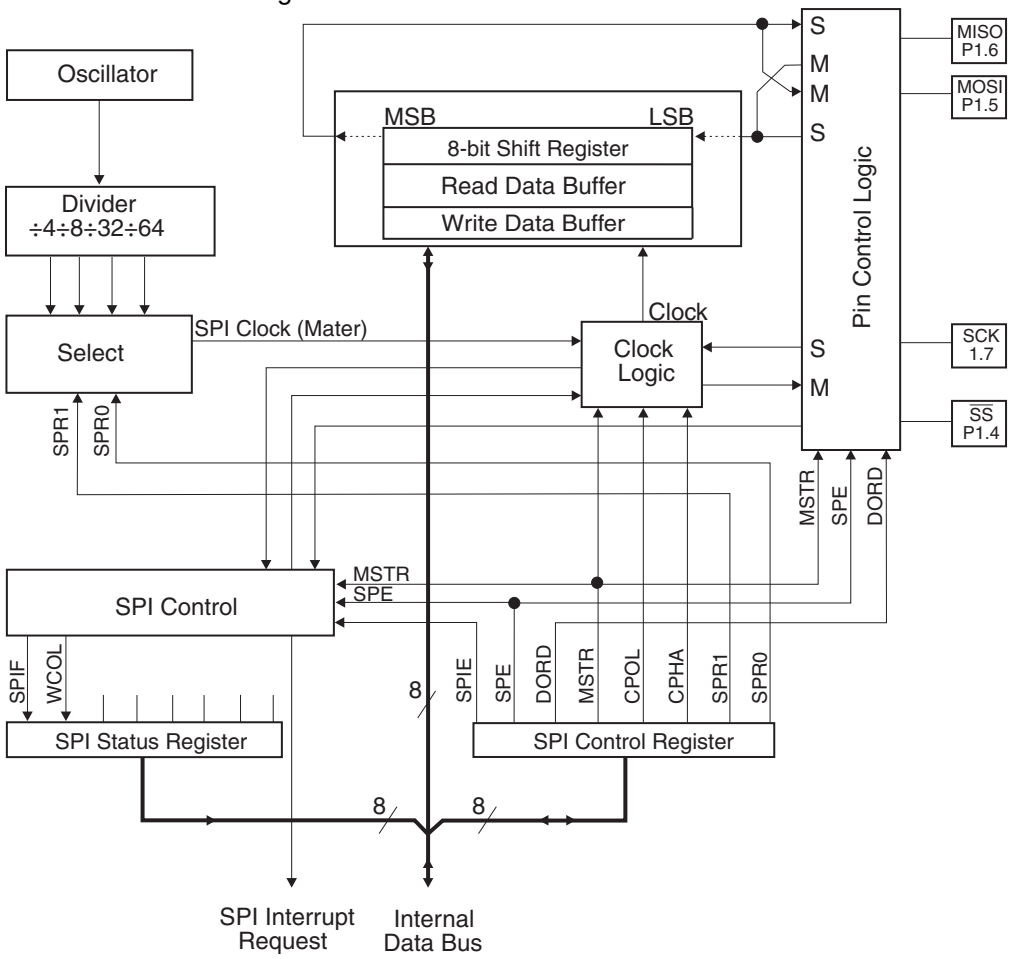
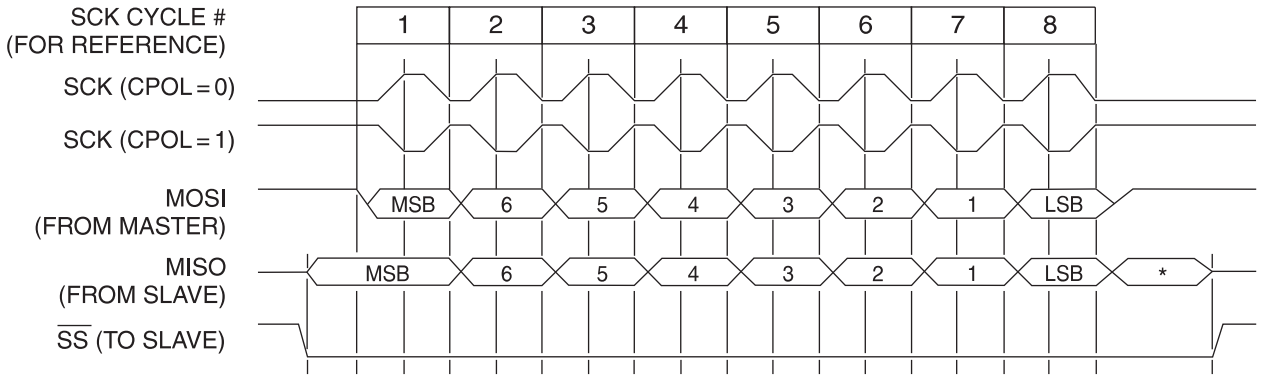



**Figure 19-3.** SPI Block Diagram



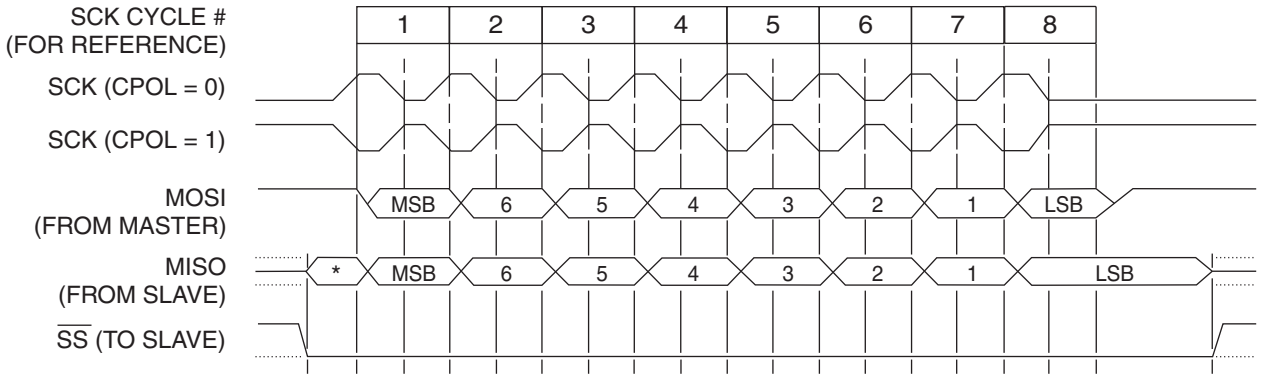
The CPHA (Clock PHAse), CPOL (Clock POLarity), and SPR (Serial Peripheral clock Rate = baud rate) bits in SPCR control the shape and rate of SCK. The two SPR bits provide four possible clock rates when the SPI is in master mode. In slave mode, the SPI will operate at the rate of the incoming SCK as long as it does not exceed the maximum bit rate. There are also four possible combinations of SCK phase and polarity with respect to the serial data. CPHA and CPOL determine which format is used for transmission. The SPI data transfer formats are shown in Figures 19-4 and 19-5. To prevent glitches on SCK from disrupting the interface, CPHA, CPOL, and SPR should not be modified while the interface is enabled, and the master device should be enabled before the slave device(s).

**Figure 19-4.** SPI Transfer Format with CPHA = 0




Note: *Not defined but normally MSB of character just received.

**Figure 19-5.** SPI Transfer Format with CPHA = 1




Note: *Not defined but normally LSB of previously transmitted character.

# 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 20-1.** ACSR – Analog Comparator Control & Status Register

ACSR = 97H — Reset Value = XXX0 0000B

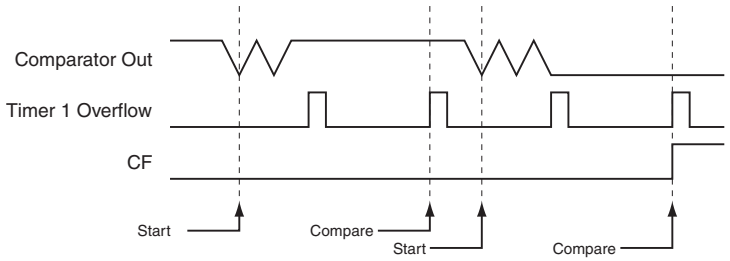
Not Bit Addressable

| | – | – | CIDL | CF | CEN | CM3 | CM1 | CM0 |
|---|---|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

| Symbol | Function |
|---|---|
| CIDL | Comparator Idle Enable. If CIDL = 1 the comparator will continue to operate during Idle mode. If CIDL = 0 the comparator is powered down during Idle mode. The comparator is always shut down during Power-down mode. |
| CF | Comparator Interrupt Flag. Set when the comparator output meets the conditions specified by the CM [2:0] bits and CEN is set. The flag must be cleared by software. The interrupt may be enabled/disabled by setting/clearing bit 6 of IE. |
| CEN | Comparator Enable. Set this bit to enable the comparator. Clearing this bit will force the comparator output low and prevent further events from setting CF. When CEN = 1 the analog input pins, P1.0 and P1.1, have their digital inputs disabled. |
| CM [2:0] | Comparator Interrupt Mode<br>CM2 CM1 CM0 Interrupt Mode<br>0 0 0 Negative (Low) level<br>0 0 1 Positive edge<br>0 1 0 Toggle with debouncing(1)<br>0 1 1 Positive edge with debouncing(1)<br>1 0 0 Negative edge<br>1 0 1 Toggle<br>1 1 0 Negative edge with debouncing(1)<br>1 1 1 Positive (High) level |

Note: 1. Debouncing modes require the use of Timer 1 to generate the sampling delay.

**Figure 20-1.** Negative Edge with Debouncing Example



# 21. Programmable Watchdog Timer

The programmable Watchdog Timer (WDT) protects the system from incorrect execution by triggering a system reset when it times out after the software has failed to feed the timer prior to the timer overflow. By Default the WDT counts CPU clock cycles. The prescaler bits, PS0, PS1 and PS2 in SFR WDTCON are used to set the period of the Watchdog Timer from 16K to 2048K clock cycles. The Timer Prescaler can also be used to lengthen the time-out period (see Table 9-2 on page 13). The WDT is disabled by Reset and during Power-down mode. When the WDT times out without being serviced, an internal RST pulse is generated to reset the CPU. See Table 21-1 for the available WDT period selections.

**Table 21-1.** Watchdog Timer Time-out Period Selection

| WDT Prescaler Bits | | | Period[1] (Clock Cycles) |
|---|---|---|---|
| PS2 | PS1 | PS0 | |
| 0 | 0 | 0 | 16K |
| 0 | 0 | 1 | 32K |
| 0 | 1 | 0 | 64K |
| 0 | 1 | 1 | 128K |
| 1 | 0 | 0 | 256K |
| 1 | 0 | 1 | 512K |
| 1 | 1 | 0 | 1024K |
| 1 | 1 | 1 | 2048K |

Note: 1. The WDT time-out period is dependent on the system clock frequency.

$$\text{Time-out Period} = \frac{2^{(PS+14)}}{\text{Oscillator Frequency}} \times (TPS+1)$$

The Watchdog Timer consists of a 14-bit timer with 7-bit programmable prescaler. Writing the sequence 1EH/E1H to the WDTRST register enables the timer. When the WDT is enabled, the WDTEN bit in WDTCON will be set to “1”. To prevent the WDT from generating a reset when if overflows, the watchdog feed sequence must be written to WDTRST before the end of the time-out period. To feed the watchdog, two write instructions must be sequentially executed successfully. Between the two write instructions, SFR reads are allowed, but writes are not allowed. The instructions should move 1EH to the WDTRST register and then 1EH to the WDTRST register. An incorrect feed or enable sequence will cause an immediate watchdog reset. The program sequence to feed or enable the watchdog timer is as follows:

```
MOV WDTRST, #01Eh
MOV WDTRST, #0E1h
```

## 21.1 Software Reset

A Software Reset of the AT89LP216 is accomplished by writing the software reset sequence 5AH/A5H to the WDTRST SFR. The WDT does not need to be enabled to generate the software reset. A normal software reset will set the SWRST flag in WDTCON. However, if at any time an incorrect sequence is written to WDTRST (i.e. anything other than 1EH/E1H or 5AH/A5H), a software reset will immediately be generated and both the SWRST and WDTOVF flags will be set. In this manner an intentional software reset may be distinguished from a software error-generated reset. The program sequence to generate a software reset is as follows:

```
MOV WDTRST, #05Ah
MOV WDTRST, #0A5h
```

**Table 21-2.** WDTCON – Watchdog Control Register

WDTCON Address = A7H — Reset Value = 0000 X000B

Not Bit Addressable

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | PS2 | PS1 | PS0 | WDIDLE | – | SWRST | WDTOVF | WDTEN |
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

| Symbol | Function |
|---|---|
| PS2<br>PS1<br>PS0 | Prescaler bits for the watchdog timer (WDT). When all three bits are cleared to 0, the watchdog timer has a nominal period of 16K clock cycles. When all three bits are set to 1, the nominal period is 2048K clock cycles. |
| WDIDLE | Disable/enable the Watchdog Timer in IDLE mode. When WDIDLE = 0, WDT continues to count in IDLE mode. When WDIDLE = 1, WDT freezes while the device is in IDLE mode. |
| SWRST | Software Reset Flag. Set when a software reset is generated by writing the sequence 5AH/A5H to WDTRST. Also set when an incorrect sequence is written to WDTRST. Must be cleared by software. |
| WDTOVF | Watchdog Overflow Flag. Set when a WDT rest is generated by the WDT timer overflow. Also set when an incorrect sequence is written to WDTRST. Must be cleared by software. |
| WDTEN | Watchdog Enable Flag. This bit is READ-ONLY and reflects the status of the WDT (whether it is running or not). The WDT is disabled after any reset and must be re-enabled by writing 1EH/E1H to WDTRST |

**Table 21-3.** WDTRST – Watchdog Reset Register

WDTCON Address = A6H — (Write-Only)

Not Bit Addressable

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | – | – | – | – | – | – | – | – |
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

The WDT is enabled by writing the sequence 1EH/E1H to the WDTRST SFR. The current status may be checked by reading the WDTEN bit in WDTCON. To prevent the WDT from resetting the device, the same sequence 1EH/E1H must be written to WDTRST before the time-out interval expires. A software reset is generated by writing the sequence 5AH/A5H to WDTRST.

## 22. Instruction Set Summary

The AT89LP216 is fully binary compatible with the MCS-51 instruction set. The difference between the AT89LP216 and the standard 8051 is the number of cycles required to execute an instruction. Instructions in the AT89LP216 may take 1, 2, 3 or 4 clock cycles to complete. The execution times of most instructions may be computed using Table 22-1.

**Table 22-1.** Instruction Execution Times and Exceptions

| Generic Instruction Types | Cycle Count Formula |
|---|---|
| Most arithmetic, logical, bit and transfer instructions | # bytes |
| Branches and Calls | # bytes + 1 |
| Single Byte Indirect (i.e. ADD A, @Ri, etc.) | 2 |
| RET, RETI | 4 |
| MOVC | 3 |
| MOVX | 4 |
| MUL | 2 |
| DIV | 4 |
| INC DPTR | 2 |

| | | Clock Cycles | | |
|---|---|---|---|---|
| **Arithmetic** | **Bytes** | **8051** | **AT89LP** | **Hex Code** |
| ADD A, Rn | 1 | 12 | 1 | 28-2F |
| ADD A, direct | 2 | 12 | 2 | 25 |
| ADD A, @Ri | 1 | 12 | 2 | 26-27 |
| ADD A, #data | 2 | 12 | 2 | 24 |
| ADDC A, Rn | 1 | 12 | 1 | 38-3F |
| ADDC A, direct | 2 | 12 | 2 | 35 |
| ADDC A, @Ri | 1 | 12 | 2 | 36-37 |
| ADDC A, #data | 2 | 12 | 2 | 34 |
| SUBB A, Rn | 1 | 12 | 1 | 98-9F |
| SUBB A, direct | 2 | 12 | 2 | 95 |
| SUBB A, @Ri | 1 | 12 | 2 | 96-97 |
| SUBB A, #data | 2 | 12 | 2 | 94 |
| INC Rn | 1 | 12 | 1 | 08-0F |
| INC direct | 2 | 12 | 2 | 05 |
| INC @Ri | 1 | 12 | 2 | 06-07 |
| INC A | 2 | 12 | 2 | 04 |
| DEC Rn | 1 | 12 | 1 | 18-1F |
| DEC direct | 2 | 12 | 2 | 15 |
| DEC @Ri | 1 | 12 | 2 | 16-17 |
| DEC A | 2 | 12 | 2 | 14 |
| INC DPTR | 1 | 24 | 2 | A3 |
| MUL AB | 1 | 48 | 2 | A4 |
| DIV AB | 1 | 48 | 4 | 84 |
| DA A | 1 | 12 | 1 | D4 |

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

| Logical | Bytes | Clock Cycles | | Hex Code |
|---|---|---|---|---|
| | | 8051 | AT89LP | |
| CLR A | 1 | 12 | 1 | E4 |
| CPL A | 1 | 12 | 1 | F4 |
| ANL A, Rn | 1 | 12 | 1 | 58-5F |
| ANL A, direct | 2 | 12 | 2 | 55 |
| ANL A, @Ri | 1 | 12 | 2 | 56-57 |
| ANL A, #data | 2 | 12 | 2 | 54 |
| ANL direct, A | 2 | 12 | 2 | 52 |
| ANL direct, #data | 3 | 24 | 3 | 53 |
| ORL A, Rn | 1 | 12 | 1 | 48-4F |
| ORL A, direct | 2 | 12 | 2 | 45 |
| ORL A, @Ri | 1 | 12 | 2 | 46-47 |
| ORL A, #data | 2 | 12 | 2 | 44 |
| ORL direct, A | 2 | 12 | 2 | 42 |
| ORL direct, #data | 3 | 24 | 3 | 43 |
| XRL A, Rn | 1 | 12 | 1 | 68-6F |
| XRL A, direct | 2 | 12 | 2 | 65 |
| XRL A, @Ri | 1 | 12 | 2 | 66-67 |
| XRL A, #data | 2 | 12 | 2 | 64 |
| XRL direct, A | 2 | 12 | 2 | 62 |
| XRL direct, #data | 3 | 24 | 3 | 63 |
| RL A | 1 | 12 | 1 | 23 |
| RLC A | 1 | 12 | 1 | 33 |
| RR A | 1 | 12 | 1 | 03 |
| RRC A | 1 | 12 | 1 | 13 |
| SWAP A | 1 | 12 | 1 | C4 |

**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 |

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

| Bit Operations | Bytes | Clock Cycles | | Hex Code |
|---|---|---|---|---|
| | | 8051 | AT89LP | |
| CLR C | 1 | 12 | 1 | C3 |
| CLR bit | 2 | 12 | 2 | C2 |
| SETB C | 1 | 12 | 1 | D3 |
| SETB bit | 2 | 12 | 2 | D2 |
| CPL C | 1 | 12 | 1 | B3 |
| CPL bit | 2 | 12 | 2 | B2 |
| ANL C, bit | 2 | 24 | 2 | 82 |
| ANL C, bit | 2 | 24 | 2 | B0 |
| ORL C, bit | 2 | 24 | 2 | 72 |
| ORL C, /bit | 2 | 24 | 2 | A0 |
| MOV C, bit | 2 | 12 | 2 | A2 |
| MOV bit, C | 2 | 24 | 2 | 92 |
| **Branching** | **Bytes** | **Clock Cycles** | | **Hex Code** |
| | | **8051** | **AT89LP** | |
| JC rel | 2 | 24 | 3 | 40 |
| JNC rel | 2 | 24 | 3 | 50 |
| JB bit, rel | 3 | 24 | 4 | 20 |
| JNB bit, rel | 3 | 24 | 4 | 30 |
| JBC bit, rel | 3 | 24 | 4 | 10 |
| JZ rel | 2 | 24 | 3 | 60 |
| JNZ rel | 2 | 24 | 3 | 70 |
| SJMP rel | 2 | 24 | 3 | 80 |
| ACALL addr11 | 2 | 24 | 3 | 11,31,51,71,91, B1,D1,F1 |
| LCALL addr16 | 3 | 24 | 4 | 12 |
| RET | 1 | 24 | 4 | 22 |
| RETI | 1 | 24 | 4 | 32 |
| AJMP addr11 | 2 | 24 | 3 | 01,21,41,61,81, A1,C1,E1 |
| LJMP addr16 | 3 | 24 | 4 | 02 |
| JMP @A+DPTR | 1 | 24 | 2 | 73 |
| JMP @A+PC[1] | 2 | – | 3 | A5 73 |
| CJNE A, direct, rel | 3 | 24 | 4 | B5 |
| CJNE A, #data, rel | 3 | 24 | 4 | B4 |
| CJNE Rn, #data, rel | 3 | 24 | 4 | B8-BF |
| CJNE @Ri, #data, rel | 3 | 24 | 4 | B6-B7 |
| DJNZ Rn, rel | 2 | 24 | 3 | D8-DF |
| DJNZ direct, rel | 3 | 24 | 4 | D5 |
| NOP | 1 | 12 | 1 | 00 |
| BREAK[1] | 2 | – | 2 | A5 00 |

Note: 1. This escaped instruction is an extension to the instruction set.

# 23. On-chip Debug System

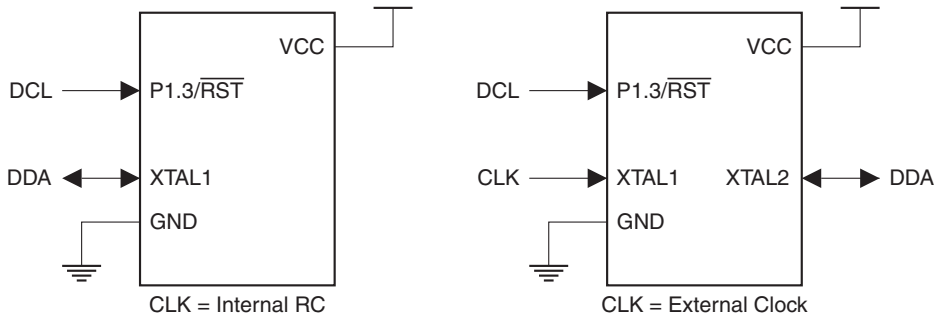
The AT89LP216 On-chip Debug (OCD) System uses a two-wire serial interface to control program flow; read, modify, and write the system state; and program the nonvolatile memory. The OCD System has the following features:

- Complete program flow control
- Read-modify-write access to all internal SFRs and data memories
- Four hardware program address breakpoints
- Unlimited program software breakpoints using BREAK instruction
- Break on stack overflow/underflow
- Break on Watchdog overflow
- Non-intrusive operation
- Programming of nonvolatile memory

## 23.1 Physical Interface

The On-chip Debug System uses a two-wire synchronous serial interface to establish communication between the target device and the controlling emulator system. The OCD interface is controlled by two User Fuses. OCD is enabled by clearing the OCD $\overline{\text{Enable}}$ Fuse. When OCD is enabled, the $\overline{\text{RST}}$ port pin is configured as an input for the Debug Clock (DCL). Either the XTAL1 or XTAL2 pin is configured as a bi-directional data line for the Debug Data (DDA) depending on the clock source selected. If the External Clock is selected, XTAL2 is configured as DDA. If the Internal RC Oscillator is selected, XTAL1 is configured as DDA. The OCD device connections are shown in Figure 23-1. The OCD Interface Select User Fuse should always be set for the fast two-wire interface (FTWI). It is the duty of the user to program these fuses to the correct settings before using the device in their debug system (see “User Configuration Fuses” on page 71).

**Figure 23-1.** AT89LP216 On-chip Debug Connections




When designing a system where On-chip Debug will be used, the following observations must be considered for correct operation:

- P1.3/$\overline{\text{RST}}$ cannot be connected directly to $V_{CC}$ and any external capacitors connect to $\overline{\text{RST}}$ must be removed.
- All external reset sources must be removed.
- The quartz crystal and any capacitors on XTAL1 or XTAL2 must be removed and an external clock signal must be driven on XTAL1 if the user does not wish to use the internal RC oscillator. Some emulator systems may provide a user-configurable clock for this purpose.

## 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

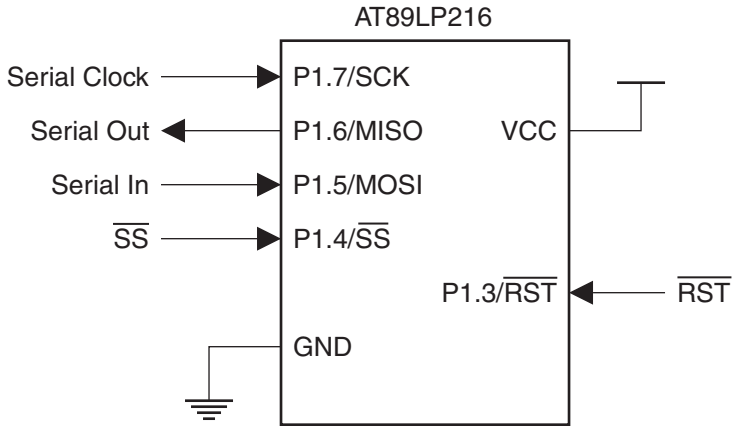
- Flexible Page Programming
- Row Erase Capability
- Page Write with Auto-Erase Commands
- Programming Status Register

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 $\overline{\text{RST}}$ input may be disabled to gain an extra I/O pin. In these cases the $\overline{\text{RST}}$ pin will always function as a reset during power up. To enter programming the $\overline{\text{RST}}$ 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 $\overline{\text{RST}}$ 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 $\overline{\text{SS}}$ 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

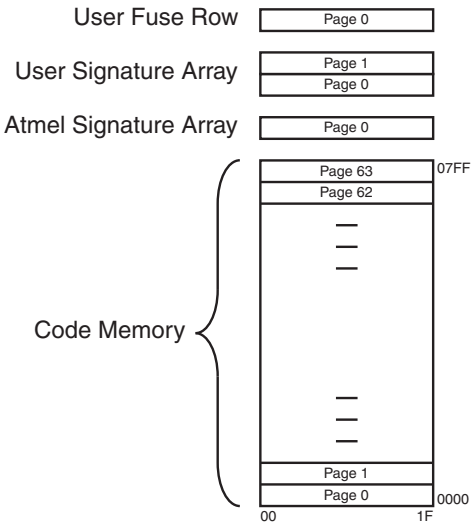
The AT89LP216 offers 2K bytes of In-System Programmable (ISP) nonvolatile Flash code memory. In addition, the device contains a 64-byte User Signature Array and a 32-byte read-only Atmel Signature Array. The memory organization is shown in Table 24-1 and Figure 24-2. The memory is divided into pages 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.

**Figure 24-3.** Command Sequence Flow Chart

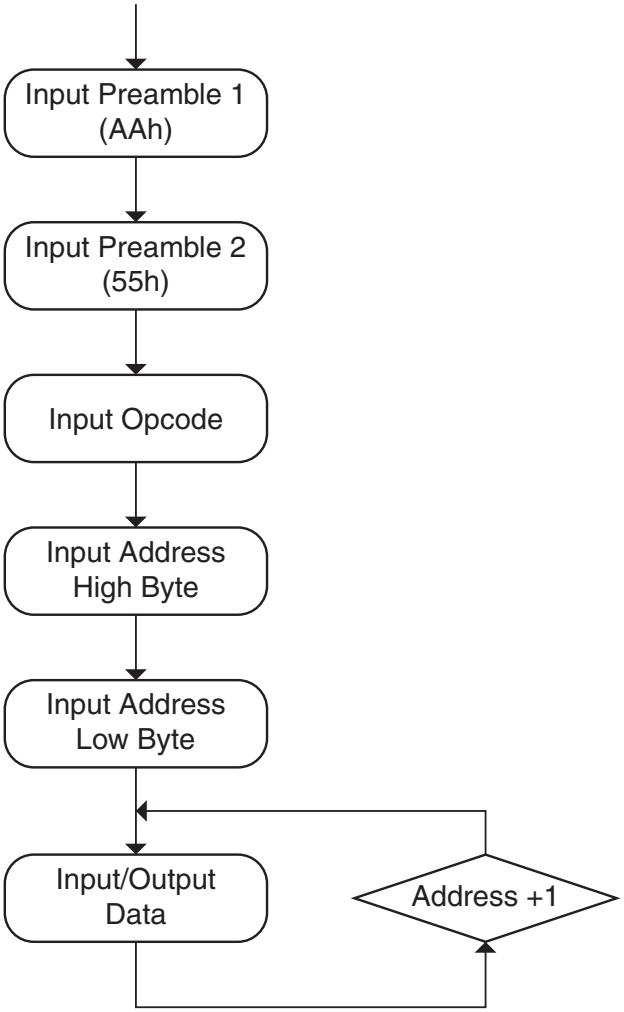



**Figure 24-4.** ISP Command Packet

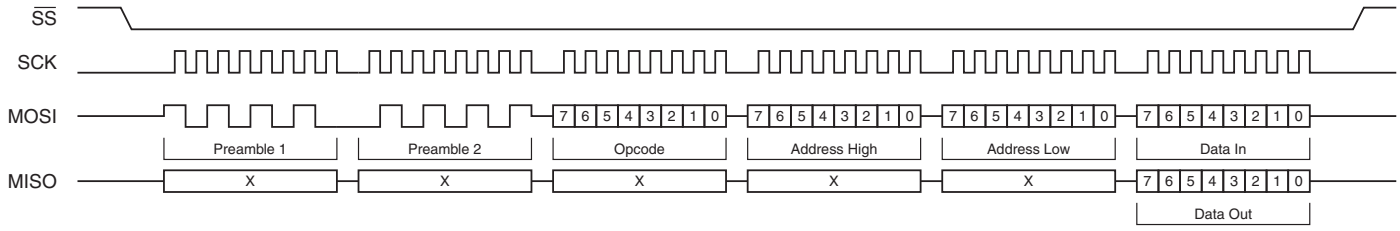


**Table 24-2.** Programming Command Summary

| Command | Opcode | Addr High | Addr Low | Data 0 | Data n |
|---|---|---|---|---|---|
| Program Enable[1] | 1010 1100 | 0101 0011 | – | – | – |
| Chip Erase | 1000 1010 | – | – | – | – |
| Read Status | 0110 0000 | xxxx xxxx | xxxx xxxx | Status Out | |
| Load Page Buffer[2] | 0101 0001 | xxxx xxxx | xxxb bbbb | DataIn 0 ... DataIn n | |
| Write Code Page[2] | 0101 0000 | xxxx xaaa | aaab bbbb | DataIn 0 ... DataIn n | |
| Write Code Page with Auto-Erase[2] | 0111 0000 | xxxx xaaa | aaab bbbb | DataIn 0 ... DataIn n | |
| Read Code Page[2] | 0011 0000 | xxxx xaaa | aaab bbbb | DataOut 0 ... DataOut n | |
| Write User Fuses[2][3][4] | 1110 0001 | 0000 0000 | 000b bbbb | DataIn 0 ... DataIn n | |
| Write User Fuses with Auto-Erase[2][3][4] | 1111 0001 | 0000 0000 | 000b bbbb | DataIn 0 ... DataIn n | |
| Read User Fuses[2][3][4] | 0110 0001 | 0000 0000 | 000b bbbb | DataOut 0 ... DataOut n | |
| Write Lock Bits[2][3][5] | 1110 0100 | 0000 0000 | 000b bbbb | DataIn 0 ... DataIn n | |
| Read Lock Bits[2][3][5] | 0110 0100 | 0000 0000 | 000b bbbb | DataOut 0 ... DataOut n | |
| Write User Signature Page[2] | 0101 0010 | xxxx xxxx | xaab bbbb | DataIn 0 ... DataIn n | |
| Write User Signature Page with Auto-Erase[2] | 0111 0010 | xxxx xxxx | xaab bbbb | DataIn 0 ... DataIn n | |
| Read User Signature Page[2] | 0011 0010 | xxxx xxxx | xaab bbbb | DataOut 0 ... DataOut n | |
| Read Atmel Signature Page[2][6] | 0011 1000 | xxxx xxxx | xxxb bbbb | DataOut 0 ... DataOut n | |

Notes:
1. Program Enable must be the **first** command issued after entering into programming mode.
2. Any number of Data bytes from 1 to 32 may be written/read. The internal address is incremented between each byte.
3. Each byte address selects one fuse or lock bit. Data bytes must be 00h or FFh.
4. See Table 24-5 on page 71 for Fuse definitions.
5. See Table 24-4 on page 70 for Lock Bit definitions.
6. **Atmel Signature Bytes**:

AT89LP216: Address 00H = 1EH
01H = 29H
02H = FFH

7. **Symbol Key**:

a: Page Address Bit
b: Byte Address Bit
x: Don't Care Bit

## 24.4 Status Register

The current state of the memory may be accessed by reading the status register. The status register is shown in Table 24-3.

**Table 24-3.** Status Register

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | – | – | – | – | LOAD | SUCCESS | WRTINH | BUSY |
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

| Symbol | Function |
|---|---|
| LOAD | Load flag. Cleared low by the load page buffer command and set high by the next memory write. This flag signals that the page buffer was previously loaded with data by the load page buffer command. |
| SUCCESS | Success flag. Cleared low at the start of a programming cycle and will only be set high if the programming cycle completes without interruption from the brownout detector. |
| WRTINH | Write Inhibit flag. Cleared low by the brownout detector (BOD) whenever programming is inhibited due to $V_{CC}$ falling below the minimum required programming voltage. If a BOD episode occurs during programming, the SUCCESS flag will remain low after the cycle is complete. $\overline{\text{WRTINH}}$ low also forces $\overline{\text{BUSY}}$ low. |
| BUSY | Busy flag. Cleared low whenever the memory is busy programming or if write is currently inhibited. |

## 24.5 $\overline{\text{DATA}}$ Polling

The AT89LP216 implements $\overline{\text{DATA}}$ polling to indicate the end of a programming cycle. While the device is busy, any attempted read of the last byte written will return the data byte with the MSB complemented. Once the programming cycle has completed, the true value will be accessible. During Erase the data is assumed to be FFH and $\overline{\text{DATA}}$ polling will return 7FH. When writing multiple bytes in a page, the $\overline{\text{DATA}}$ value will be the last data byte loaded before programming begins, not the written byte with the highest physical address within the page.

## 24.6 Flash Security

The AT89LP216 provides two Lock Bits for Flash Code Memory security. Lock bits can be left unprogrammed (FFh) or programmed (00h) to obtain the protection levels listed in Table 24-4. Lock bits can only be erased (set to FFh) by Chip Erase. Lock bit mode 2 disables programming of all memory spaces, including the User Signature Array and User Configuration Fuses. User fuses must be programmed before enabling Lock bit mode 2 or 3. Lock bit mode 3 implemented mode 2 and also blocks reads from the code memory; however, reads of the User Signature Array, Atmel Signature Array, and User Configuration Fuses are still allowed.

**Table 24-4.** Lock Bit Protection Modes

| Program Lock Bits (by address) | | | |
|---|---|---|---|
| **Mode** | **00h** | **01h** | **Protection Mode** |
| 1 | FFh | FFh | No program lock features |
| 2 | 00h | FFh | Further programming of the Flash is disabled |
| 3 | 00h | 00h | Further programming of the Flash is disabled and verify (read) is also disabled; OCD is disabled |

## 24.7 User Configuration Fuses

The AT89LP216 includes 19 user fuses for configuration of the device. Each fuse is accessed at a separate address in the User Fuse Row as listed in Table 24-5. Fuses are cleared by programming 00h to their locations. Programming FFh to fuse location will cause that fuse to maintain its previous state. To set a fuse (set to FFh) the fuse row must be erased and then reprogrammed using the Fuse Write with Auto-erase command. The default state for all fuses is FFh.

**Table 24-5.** User Configuration Fuse Definitions

| Address | Fuse Name | Description |
|---|---|---|
| 00 – 01h | Clock Source – CS[0:1](2) | Selects source for the system clock:<br>CS1 CS0 Selected Source<br>00h 00h Crystal Oscillator (XTAL)<br>00h FFh Reserved<br>FFh 00h External Clock on XTAL1 (XCLK)<br>FFh FFh Internal RC Oscillator (IRC) |
| 02 – 03h | Start-up Time – SUT[0:1] | Selects time-out delay for the POR/BOD/PWD wake-up period:<br>SUT1 SUT0 Selected Time-out<br>00h 00h 1 ms (XTAL); 16 µs (XCLK/IRC)<br>00h FFh 2 ms (XTAL); 512 µs (XCLK/IRC)<br>FFh 00h 4 ms (XTAL); 1 ms (XCLK/IRC)<br>FFh FFh 16 ms (XTAL); 4 ms (XCLK/IRC) |
| 04h | Reset Pin Enable(3) | FFh: $\overline{\text{RST}}$ pin functions as reset<br>00h: $\overline{\text{RST}}$ pin functions as general purpose I/O |
| 05h | Brown-out Detector Enable | FFh: Brown-out Detector Enabled<br>00h: Brown-out Detector Disabled |
| 06h | On-chip Debug $\overline{\text{Enable}}$ | FFh: On-chip Debug Disabled<br>00h: On-chip Debug Enabled |
| 07h | ISP Enable(3) | FFh: In-System Programming Enabled<br>00h: In-System Programming Disabled (Enabled at POR only) |
| 08 – 0FH | RC Oscillator Frequency Adjustment [0:7] | Adjusts the frequency of the internal RC oscillator. A copy of the 8MHz factory setting is stored at location 0008h of the Atmel Signature. |
| 10H | User Signature Programming | FFh: Programming of User Signature Disabled<br>00h: Programming of User Signature Enabled |
| 11H | Tristate Ports(4) | FFh: I/O Ports start in input-only mode (tristated) after reset<br>00h: I/O Ports start in quasi-bidirectional mode after reset |
| 12H | OCD Interface Select | FFh: Fast two-wire interface<br>00h: Do not use |

Notes:
1. The default state for all fuses is FFh.
2. Changes to these fuses will only take effect after a device POR.
3. Changes to these fuses will only take effect after the ISP session terminates by bringing $\overline{\text{RST}}$ high.
4. AIN0 (P1.0) and AIN1 (P1.1) always reset to input-only mode. $\overline{\text{SS}}$ (P1.4) always resets to quasi-bidirectional mode.

## 24.8 Programming Interface Timing

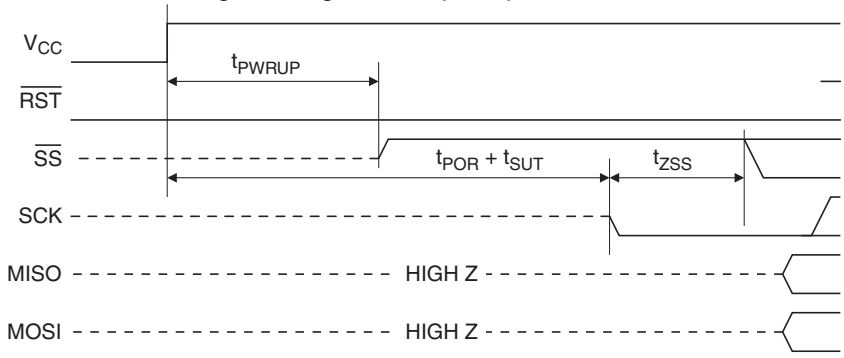
This section details general system timing sequences and constraints for entering or exiting In-System Programming as well as parameters related to the Serial Peripheral Interface during ISP. The general timing parameters for the following waveform figures are listed in Section 24.8.6 “Timing Parameters” on page 75.

### 24.8.1 Power-up Sequence

Execute this sequence to enter programming mode immediately after power-up. In the $\overline{RST}$ pin is disabled or if the ISP Fuse is disabled, this is the only method to enter programming (see Section 10.3 “External Reset” on page 15).

1. Apply power between VCC and GND pins. $\overline{RST}$ should remain low.
2. Wait at least $t_{PWRUP}$ and drive $\overline{SS}$ high.
3. Wait at least $t_{SUT}$ for the internal Power-on Reset to complete. The value of $t_{SUT}$ will depend on the current settings of the device.
4. Start programming session.

**Figure 24-5.** Serial Programming Power-up Sequence

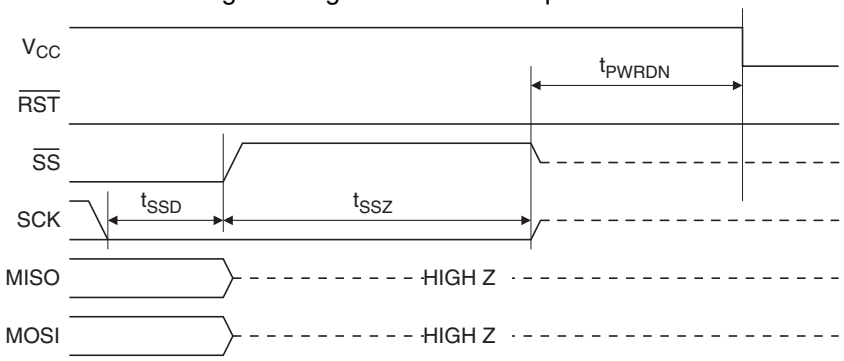



### 24.8.2 Power-down Sequence

Execute this sequence to power-down the device **after** programming.

1. Drive SCK low.
2. Wait at least $t_{SSD}$ and bring $\overline{SS}$ high.
3. Tristate MOSI.
4. Wait at least $t_{SSZ}$ and then tristate $\overline{SS}$ and SCK.
5. Wait no more than $t_{PWRDN}$ and power off VCC.

**Figure 24-6.** Serial Programming Power-down Sequence

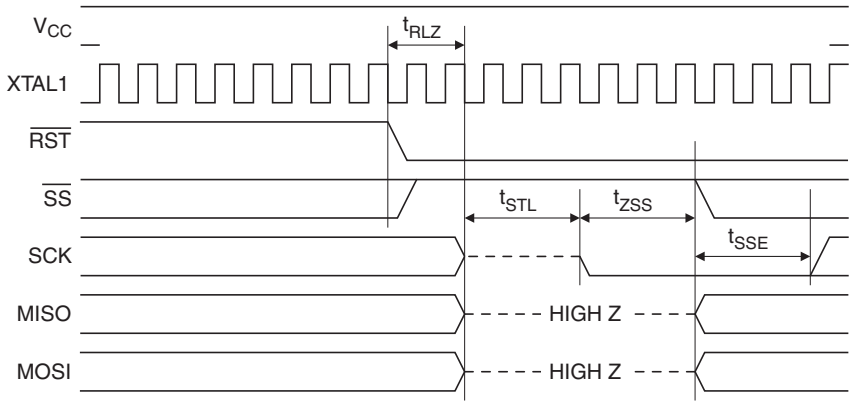


### 24.8.3 ISP Start Sequence

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

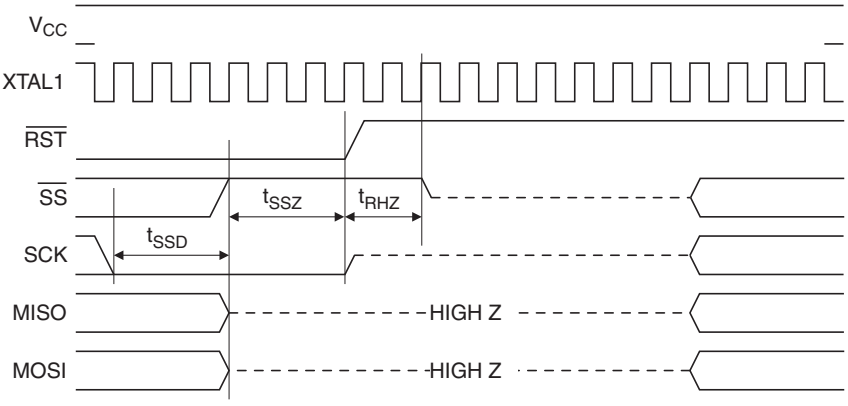



### 24.8.4 ISP Exit Sequence

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.

### 24.8.5 Serial Peripheral Interface

The Serial Peripheral Interface (SPI) is a byte-oriented full duplex synchronous serial communication channel. During In-System programming the programmer always acts as the SPI master and the target device always acts as the SPI slave. The target device receives serial data on MOSI and outputs serial data on MISO. The Programming Interface implements a standard SPI Port with a fixed data order and For In-System programming, bytes are transferred MSB first as shown in Figure 24-9. The SCK phase and polarity follow SPI clock mode 0 (CPOL = 0, CPHA = 0) where bits are sampled on the rising edge of SCK and output on the falling edge of SCK. For more detailed timing information see Figure 24-10.

**Figure 24-9.** ISP Byte Sequence

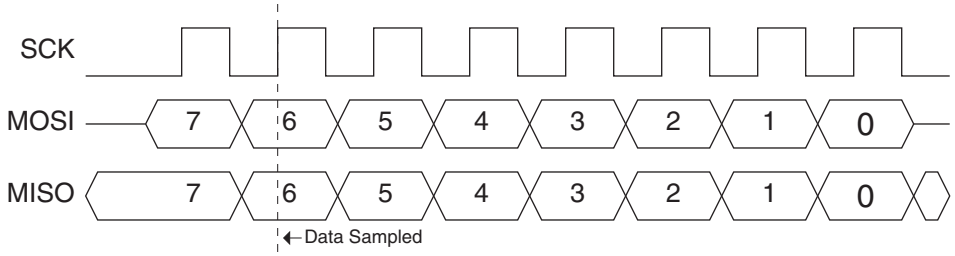
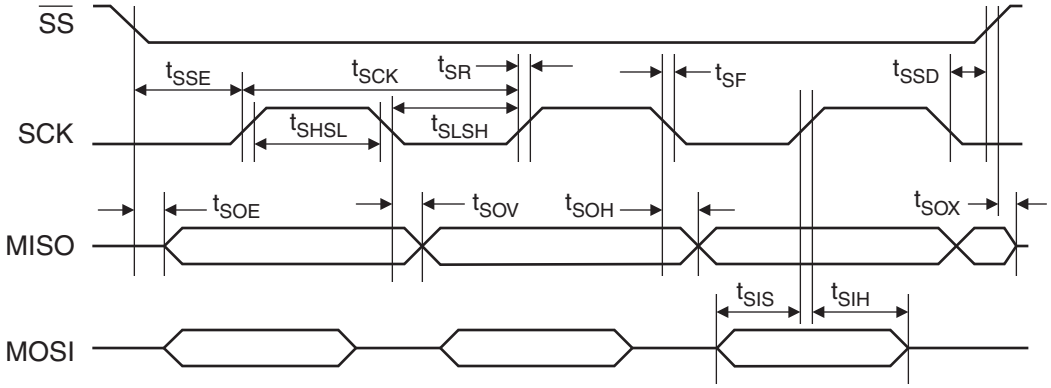



**Figure 24-10.** Serial Programming Interface Timing



### 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 |
| $t_{PWRUP}$ | Power On to $\overline{SS}$ High Time | 10 | | µs |
| $t_{POR}$ | Power-on Reset Time | | 100 | µs |
| $t_{PWRDN}$ | $\overline{SS}$ 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 |
| $t_{WR}$ | Write Cycle Time | 2.5 | | ms |
| $t_{AWR}$ | 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. Electrical Characteristics

## 25.1 Absolute Maximum Ratings*

| | |
|---|---|
| Operating Temperature | -40°C to +85°C |
| Storage Temperature | -65°C to +150°C |
| Voltage on Any Pin with Respect to Ground | -0.7V to +5.5V |
| Maximum Operating Voltage | 5.5V |
| DC Output Current | 15.0 mA |

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

## 25.2 DC Characteristics

$T_A$ = -40°C to 85°C, $V_{CC}$ = 2.4V to 5.5V (unless otherwise noted)

| 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 |
| $V_{OL}$ | Output Low-voltage[1] (Ports 1, 3) | $I_{OL}$ = 10 mA, $T_A$ = 85°C | | 0.5 | V |
| $V_{OH}$ | Output High-voltage (Ports 1, 3) With Weak Pull-ups Enabled | $I_{OH}$ = -80 µA, $V_{CC}$ = 5V ±10% | 2.4 | | V |
| | | $I_{OH}$ = -25 µA | 0.7 $V_{CC}$ | | V |
| | | $I_{OH}$ = -10 µA | 0.9 $V_{CC}$ | | V |
| $V_{OH1}$ | Output High-voltage (Ports 1, 3) With Strong Pull-ups Enabled[2] | $I_{OH}$ = -2 mA, $V_{CC}$ = 5V ±10% | 2.4 | | V |
| | | $I_{OH}$ = -500 µA, $V_{CC}$ = 5V ±10% | 0.6 $V_{CC}$ | | V |
| | | $I_{OH}$ = -100 µA | 0.6 $V_{CC}$ | | V |
| $I_{IL}$ | Logic 0 Input Current (Quasi-Bidirectional Mode) | $V_{IN}$ = 0.45V | | -50 | µA |
| $I_{TL}$ | Logic 1 to 0 Transition Current (Quasi-Bidirectional Mode) | $V_{IN}$ = 2V, $V_{CC}$ = 5V ±10% | | -300 | µA |
| $I_{LI}$ | Input Leakage Current (Input-Only Mode) | 0 < $V_{IN}$ < $V_{CC}$ | | ±10 | µA |
| $V_{OS}$ | Comparator Input Offset Voltage | $V_{CC}$ = 5V | | 20 | mV |
| $V_{CM}$ | Comparator Input Common Mode Voltage | | 0 | $V_{CC}$ | V |
| RRST | Reset Pull-up Resistor | | 50 | 150 | kΩ |
| $C_{IO}$ | Pin Capacitance | Test Freq. = 1 MHz, $T_A$ = 25°C | | 10 | pF |
| $I_{CC}$ | Power Supply Current | Active Mode, 12 MHz, $V_{CC}$ = 5V/3V | | 10/6 | mA |
| | | Idle Mode, 12 MHz, $V_{CC}$ = 5./3V | | 3/1.5 | mA |
| | Power-down Mode[3] | $V_{CC}$ = 5V, P1.0 & P1.1 = 0V or $V_{CC}$ | | 5 | µA |
| | | $V_{CC}$ = 3V, P1.0 & P1.1 = 0V or $V_{CC}$ | | 2 | µA |

Notes:
1. Under steady state (non-transient) conditions, $I_{OL}$ must be externally limited as follows:
   Maximum $I_{OL}$ per port pin: 10 mA
   Maximum total $I_{OL}$ for all output pins: 15 mA
   If $I_{OL}$ exceeds the test condition, $V_{OL}$ may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
2. $V_{OH1}$ levels are listed for DC operation. Under AC conditions a boost circuit provides additional source current.
3. Minimum $V_{CC}$ for Power-down is 2V.

## 25.3 Typical Characteristics

The following charts show typical behavior. These figures are not tested during manufacturing. All current consumption measurements are performed with all I/O pins configured as quasi-bidirectional (with internal pull-ups). A square wave generator with rail-to-rail output is used as an external clock source for consumption versus frequency measurements.

### 25.3.1 Supply Current (Internal Oscillator)

**Figure 25-1.** Active Supply Current vs. VCC ( 8MHz Internal Oscillator)

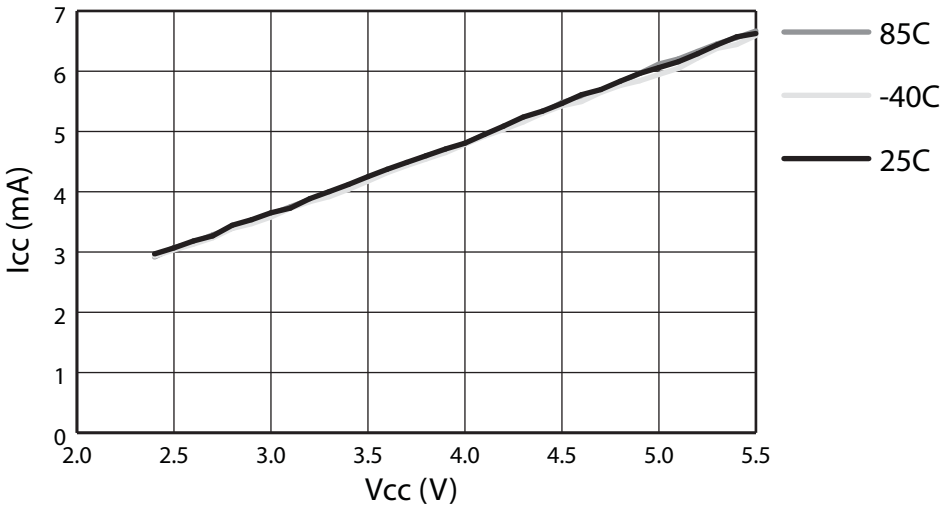
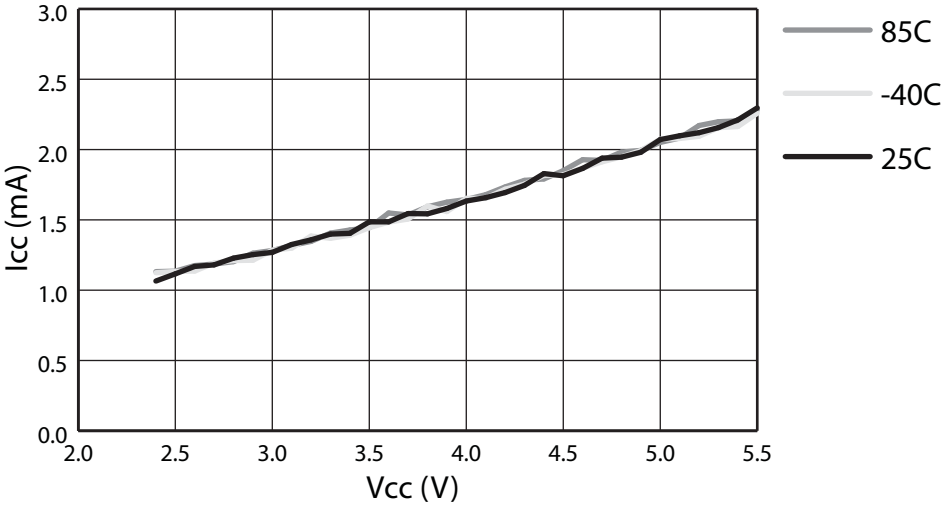



**Figure 25-2.** Idle Supply Current vs. Vcc (8 MHz Internal Oscillator)



### 25.3.2 Supply Current (External Clock)

**Figure 25-3.** Active Supply Current vs. Frequency

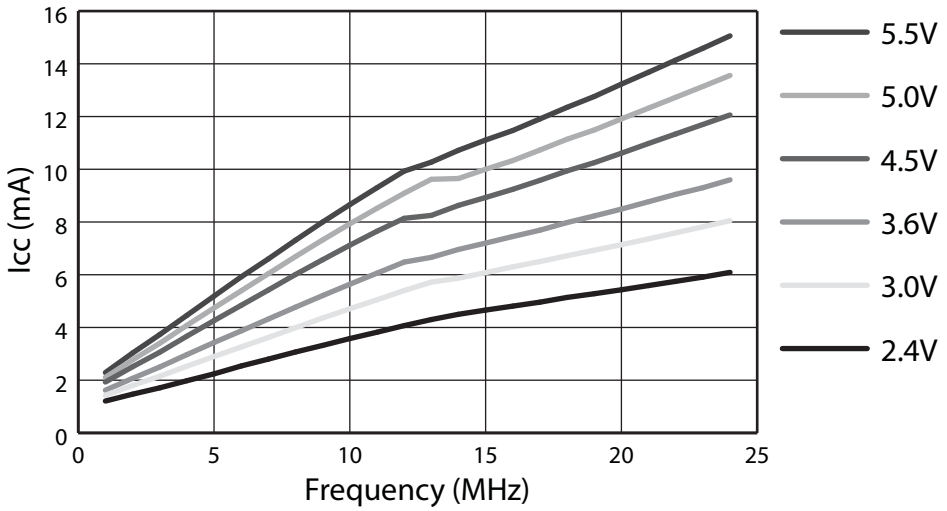
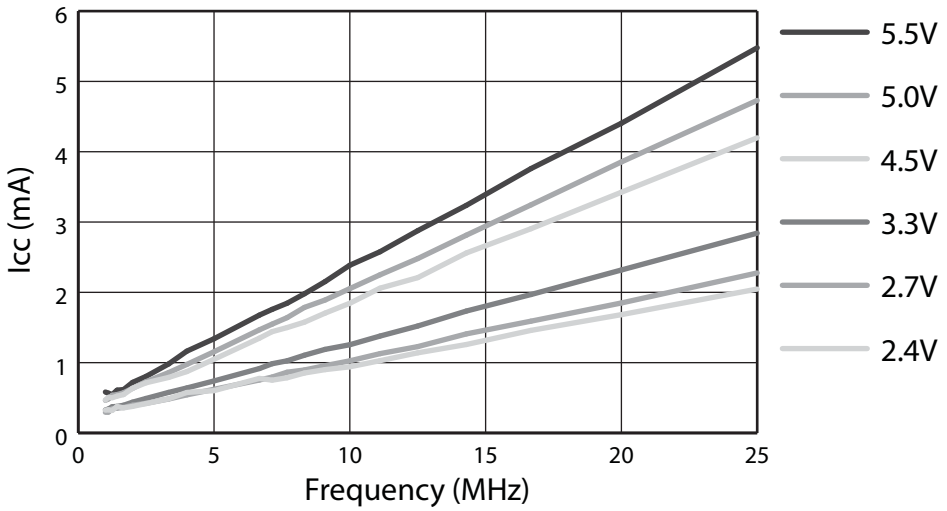



**Figure 25-4.** Idle Supply Current vs. Frequency



### 25.3.3 Internal Oscillator Frequency

**Figure 25-5.** Internal Oscillator Frequency vs. VCC

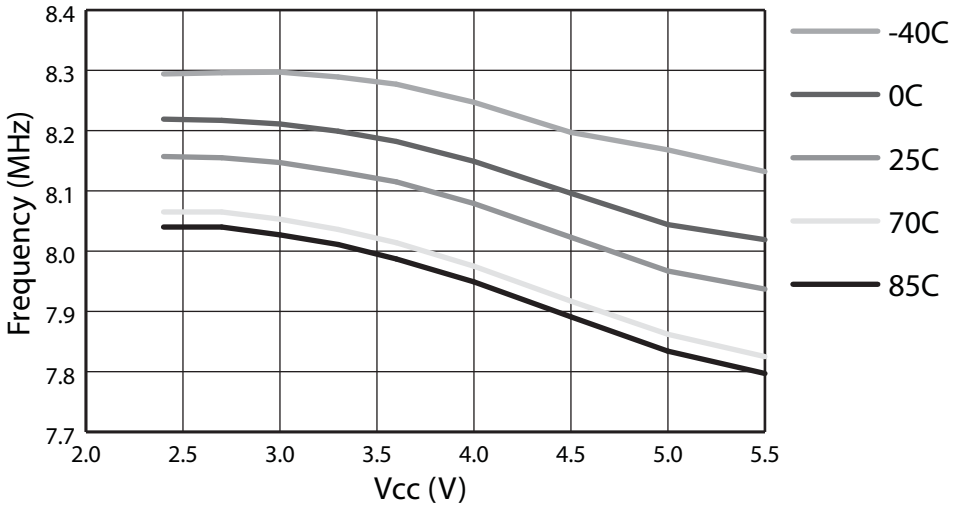



**Table 25-1.** Typical Internal Oscillator Behavior

| 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.

### 25.3.4 Quasi-Bidirectional Output

**Figure 25-6.** Quasi-Bidirectional Output I-V Characteristic at 5V

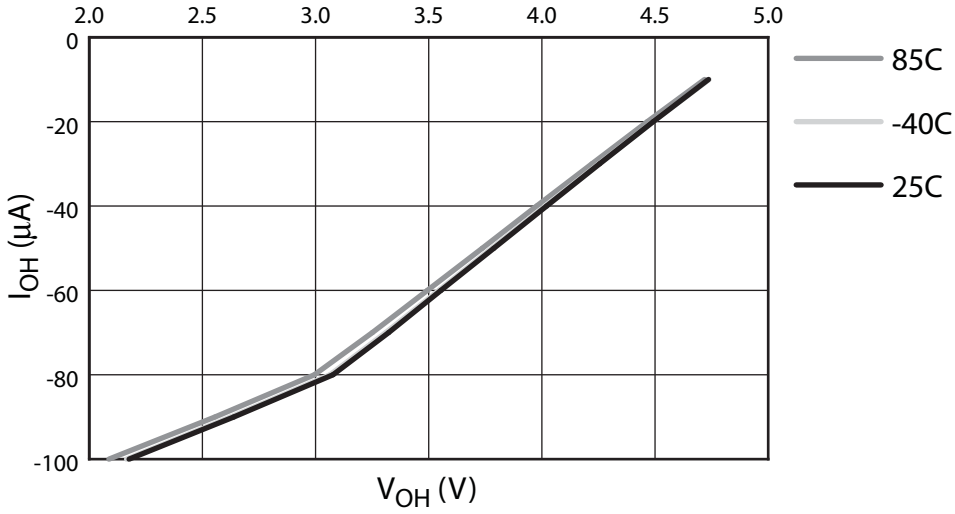
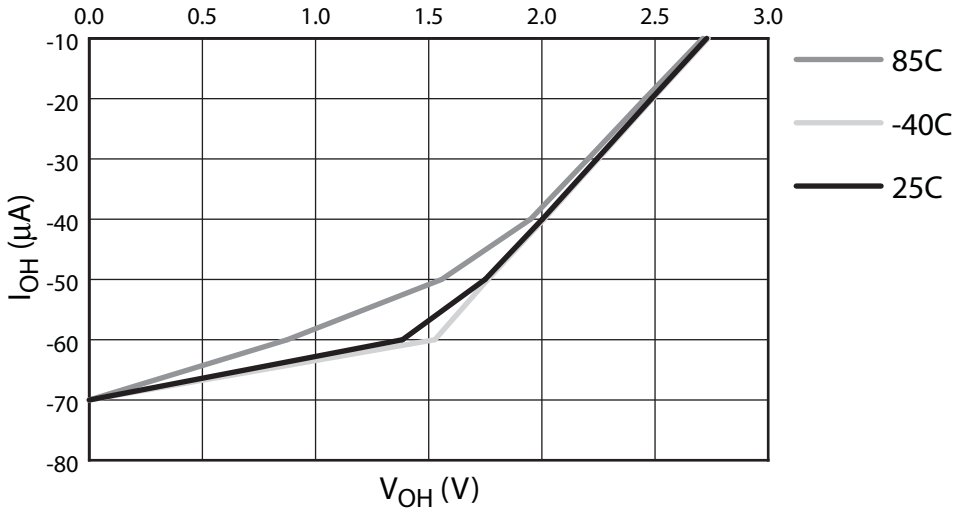



**Figure 25-7.** Quasi-Bidirectional Output I-V Characteristic at 3V



### 25.3.5 Push-Pull Output

**Figure 25-8.** Push-Pull Output I-V Characteristic at 5V

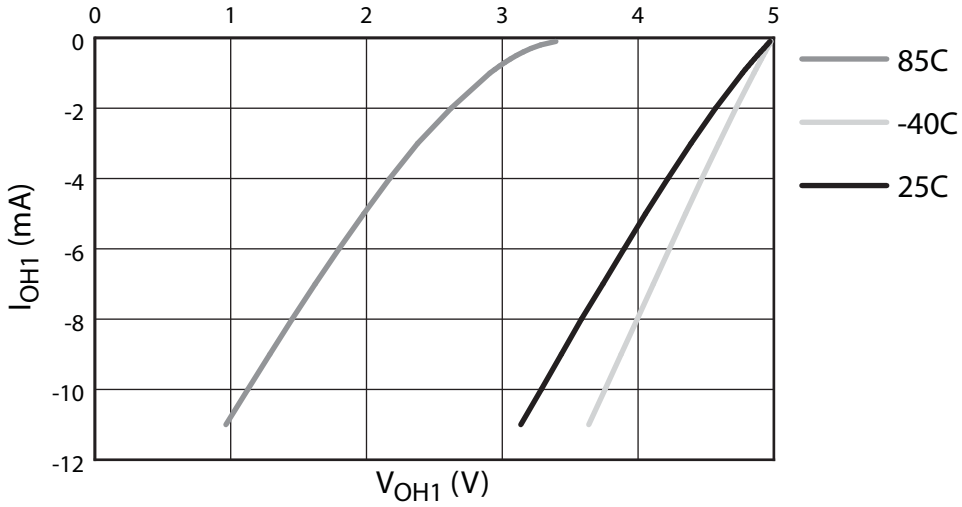
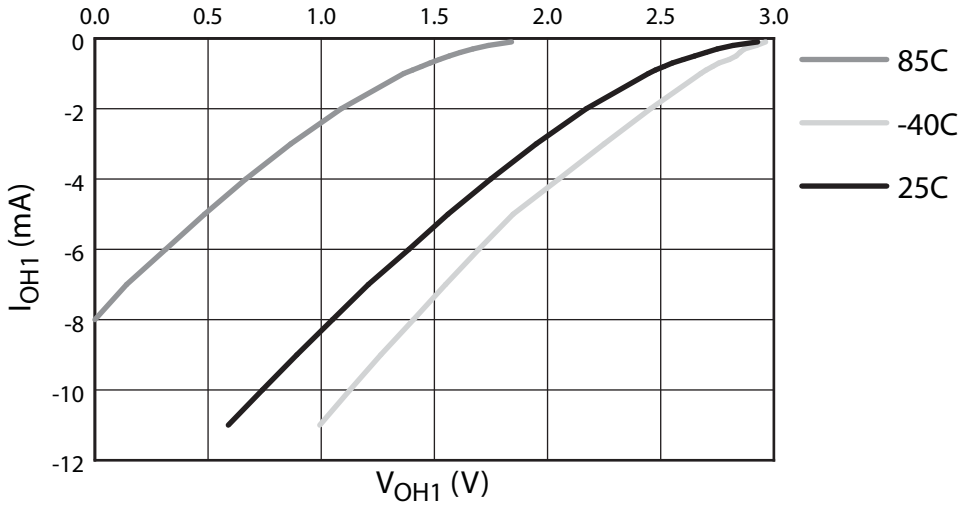



**Figure 25-9.** Push-Pull Output I-V Characteristic at 3V




Note: Under DC operating conditions the Push-Pull Outputs exhibit reduced $V_{OH}$ levels at higher temperatures due to the 5V tolerant port structure. Under AC conditions a boost circuit provides additional source current. If additional DC source current is required, or if $V_{OH}$ is too low, external pull-ups may be needed. DC conditions are most likely to exist when the device enters Power-down mode.

### 25.3.6 Crystal Oscillator

**Figure 25-10.** Quartz Crystal Input at 5V

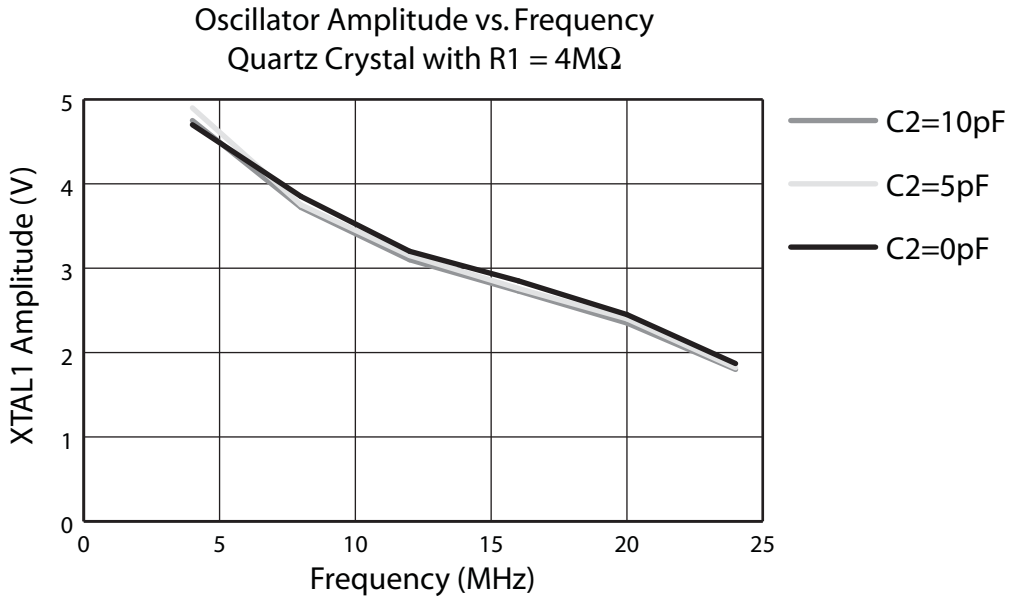
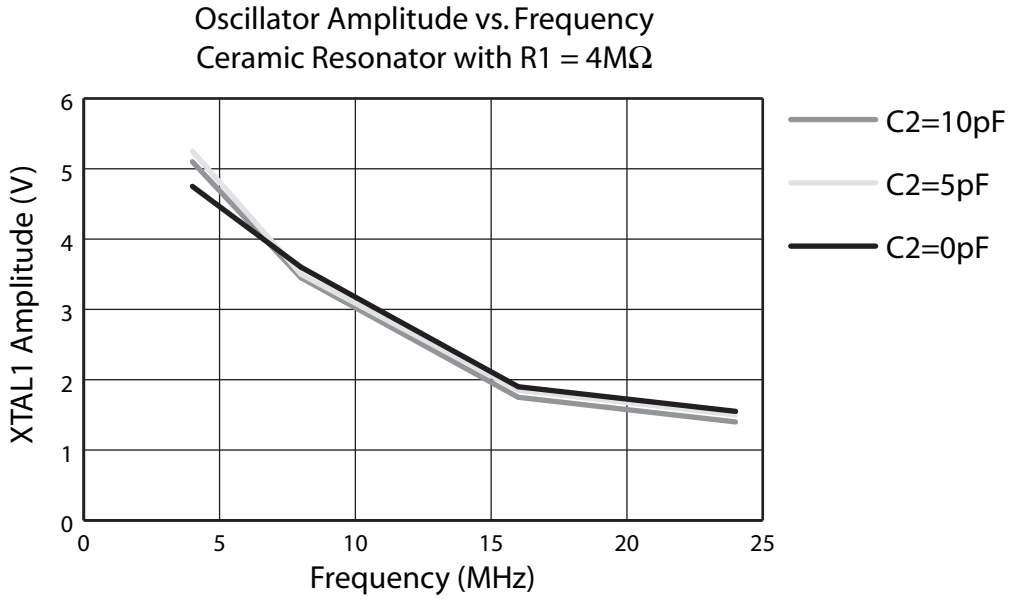



**Figure 25-11.** Ceramic Resonator Input at 5V



## 25.4 Clock Characteristics

**Figure 25-12.** External Clock Drive Waveform

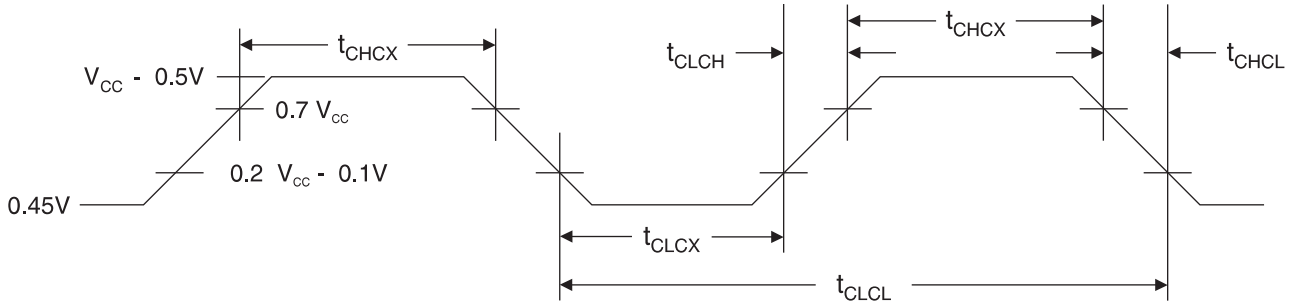



**Table 25-2.** External Clock Parameters

| Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|
| $1/t_{CLCL}$ | Oscillator Frequency | 0 | 20 | MHz |
| $t_{CLCL}$ | Clock Period | 50 | | ns |
| $t_{CHCX}$ | External Clock High Time | 12 | | ns |
| $t_{CLCX}$ | External Clock Low Time | 12 | | ns |
| $t_{CLCH}$ | External Clock Rise Time | | 5 | ns |
| $t_{CHCL}$ | External Clock Fall Time | | 5 | ns |

**Table 25-3.** Clock Characteristics

| Symbol | Parameter | Condition | Min | Max | Units |
|---|---|---|---|---|---|
| $f_{XTAL}$ | Crystal Oscillator Frequency | | 0 | 20 | MHz |
| $f_{IRC}$ | Internal Oscillator Frequency | $T_A$ = 25°C; $V_{CC}$ = 5.0V | 7.92 | 8.08 | MHz |
| | | $V_{CC}$ = 2.4V to 5.5V | 7.70 | 8.50 | MHz |

## 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 |

**Figure 25-13.** SPI Master Timing (CPHA = 0)

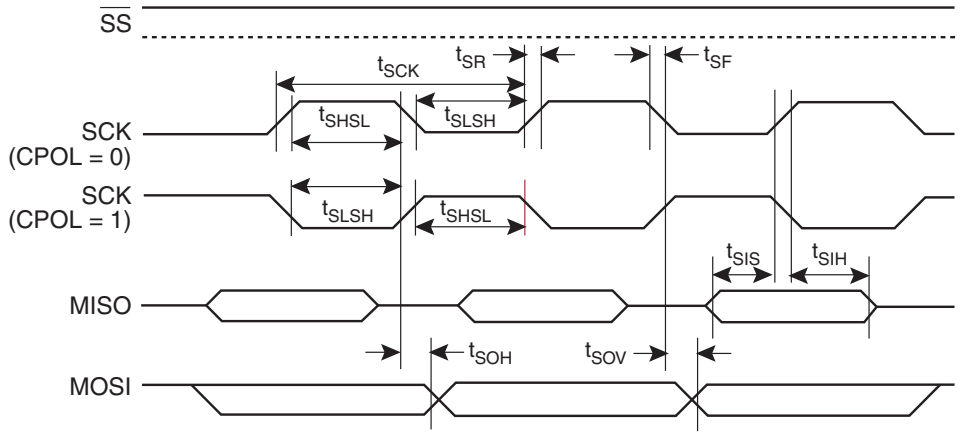



**Figure 25-14.** SPI Slave Timing (CPHA = 0)

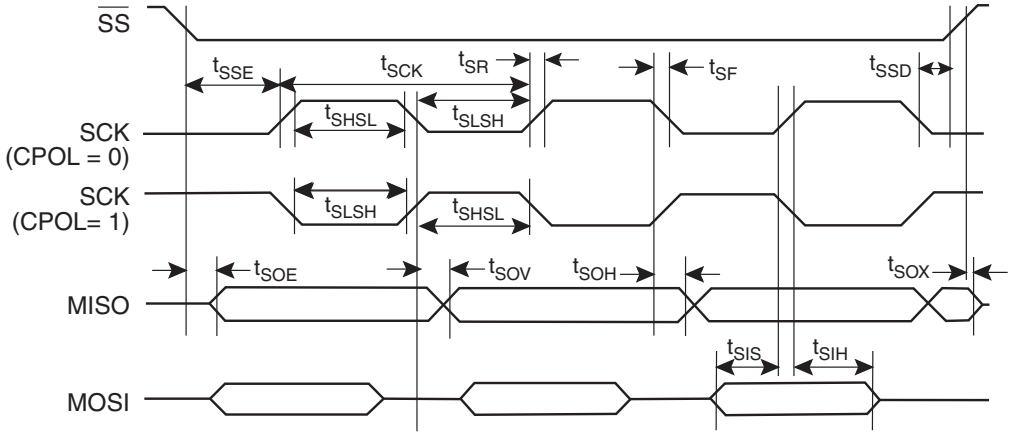


**Figure 25-15.** SPI Master Timing (CPHA = 1)

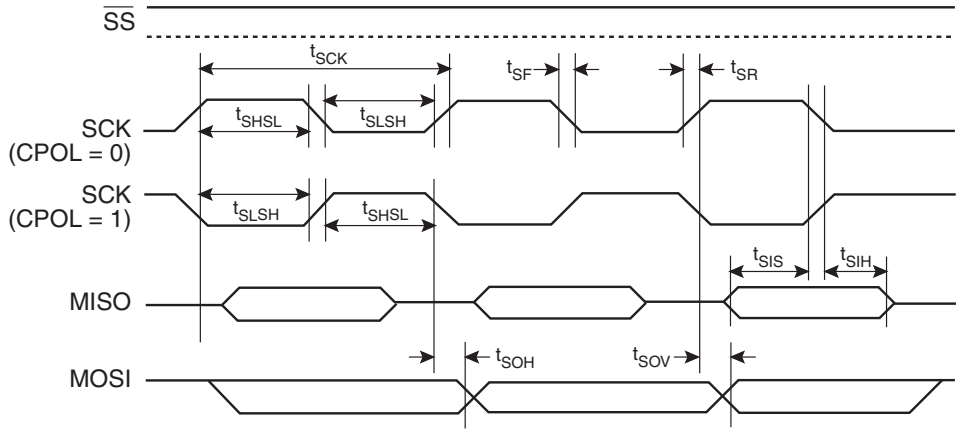
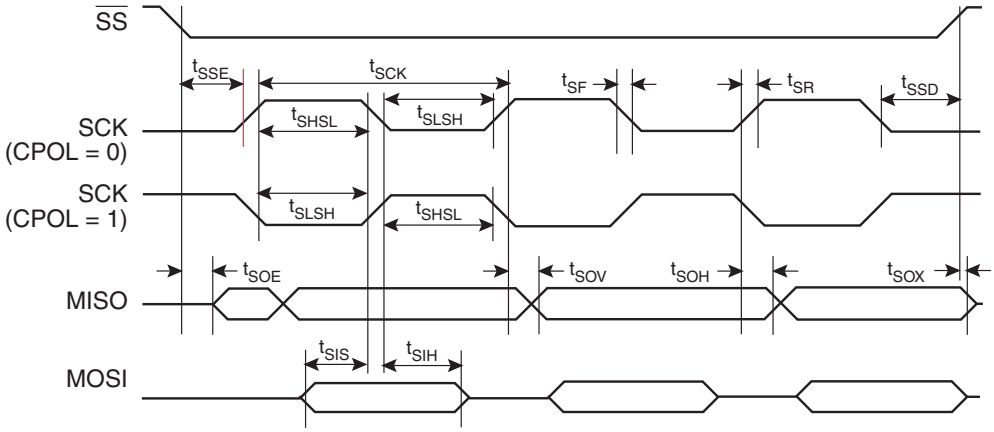



**Figure 25-16.** SPI Slave Timing (CPHA = 1)

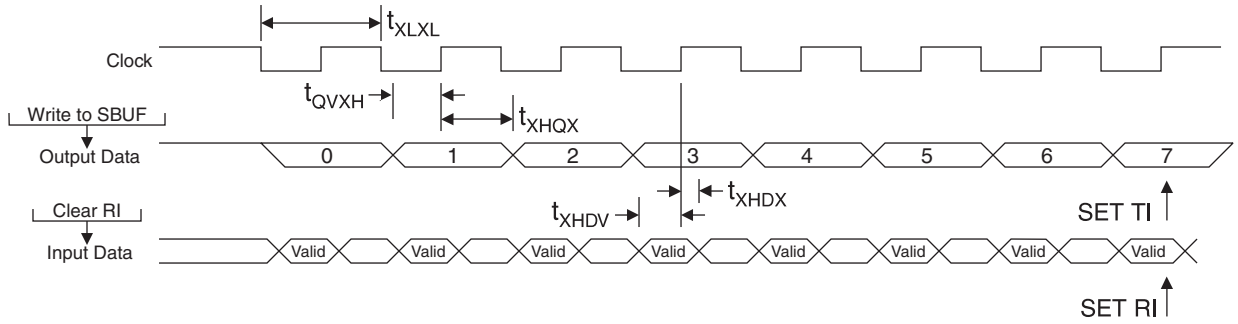


## 25.6 Serial Port Timing: Shift Register Mode Test Conditions

The values in this table are valid for $V_{CC}$ = 2.4V to 5.5V and Load Capacitance = 80 pF.

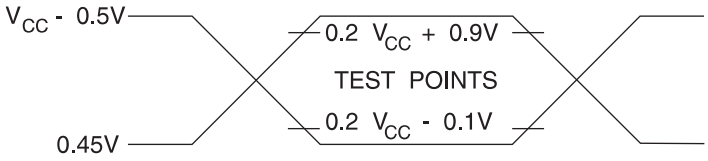
| Symbol | Parameter | Variable Oscillator | | Units |
|---|---|---|---|---|
| | | Min | Max | |
| $t_{XLXL}$ | Serial Port Clock Cycle Time | $2t_{CLCL}$ -15 | | µs |
| $t_{QVXH}$ | Output Data Setup to Clock Rising Edge | $t_{CLCL}$ -15 | | ns |
| $t_{XHQX}$ | Output Data Hold after Clock Rising Edge | $t_{CLCL}$ -15 | | ns |
| $t_{XHDX}$ | Input Data Hold after Clock Rising Edge | 0 | | ns |
| $t_{XHDV}$ | Input Data Valid to Clock Rising Edge | 15 | | ns |

**Figure 25-17.** Shift Register Mode Timing Waveform



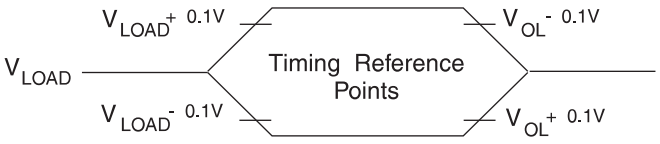

## 25.7 Test Conditions

### 25.7.1 AC Testing Input/Output Waveform[1]



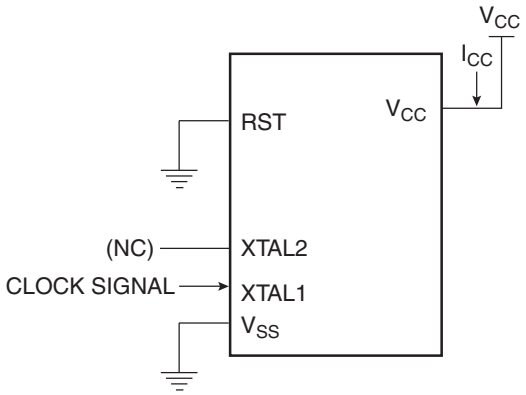

Note: 1. AC Inputs during testing are driven at $V_{CC}$ - 0.5V for a logic “1” and 0.45V for a logic “0”. Timing measurements are made at $V_{IH}$ min. for a logic “1” and $V_{IL}$ max. for a logic “0”.

### 25.7.2 Float Waveform[1]

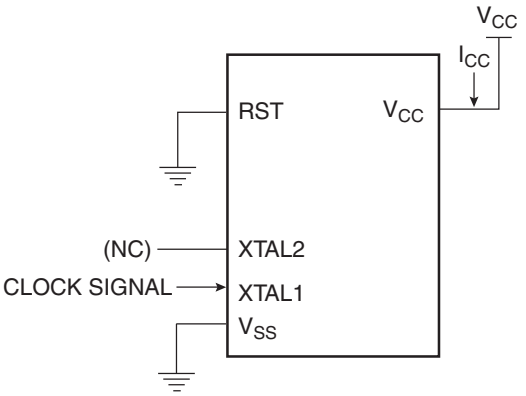



Note: 1. For timing purposes, a port pin is no longer floating when a 100 mV change from load voltage occurs. A port pin begins to float when 100 mV change from the loaded $V_{OH}/V_{OL}$ level occurs.

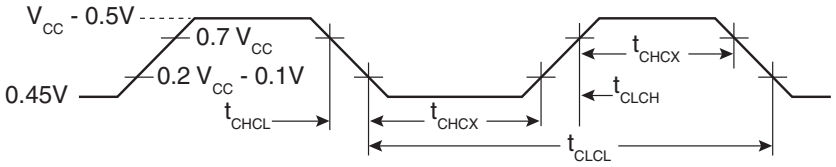
### 25.7.3 $I_{CC}$ Test Condition, Active Mode, All Other Pins are Disconnected



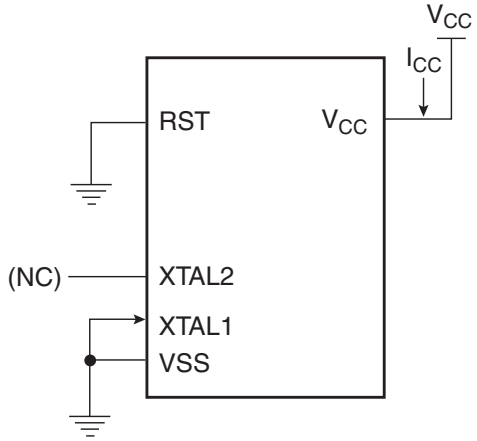

### 25.7.4 $I_{CC}$ Test Condition, Idle Mode, All Other Pins are Disconnected




### 25.7.5 Clock Signal Waveform for $I_{CC}$ Tests in Active and Idle Modes, $t_{CLCH} = t_{CHCL} = 5$ ns




### 25.7.6 $I_{CC}$ Test Condition, Power-down Mode, All Other Pins are Disconnected, $V_{CC}$ = 2V to 5.5V



# 26. Ordering Information

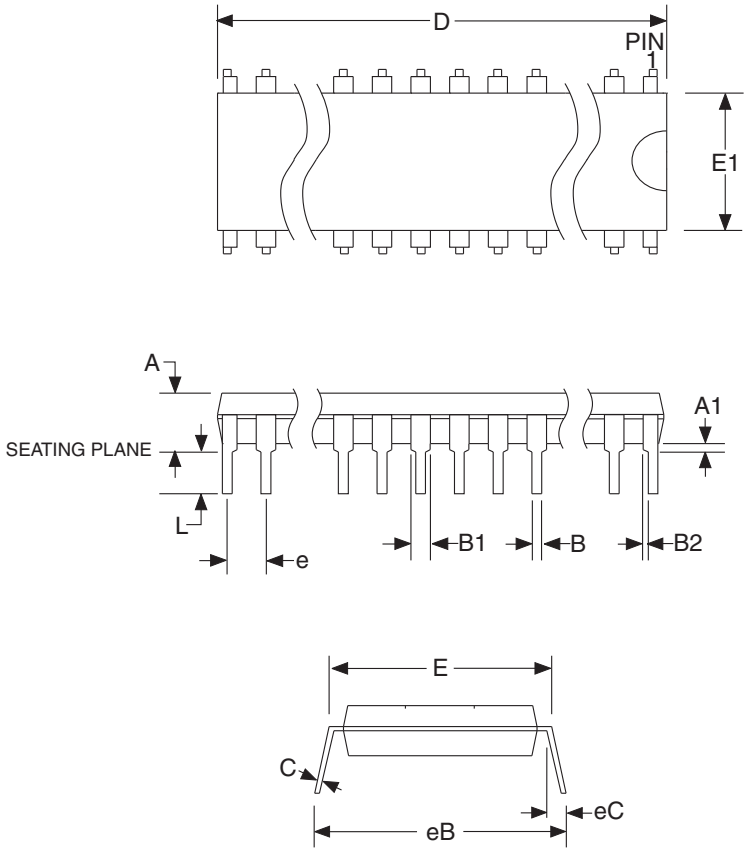
## 26.1 Green Package Option (Pb/Halide-free)

| Speed (MHz) | Power Supply | Ordering Code | Package | Operation Range |
|---|---|---|---|---|
| 20 | 2.4V to 5.5V | AT89LP216-20PU<br>AT89LP216-20SU<br>AT89LP216-20XU | 16P3<br>16S<br>16X | Industrial<br>(-40· C to 85· C) |

| Package Type | |
|---|---|
| **16P3** | 16-lead, 0.300” Wide, Plastic Dual In-line Package (PDIP) |
| **16S** | 16-lead, 0.300” Wide, Plastic Gull Wing Small Outline (SOIC) |
| **16X** | 16-lead, 0.173” Wide, Plastic Thin Shrink Small Outline Package (TSSOP) |

# 27. Packaging Information

## 27.1 16P3 – PDIP




**COMMON DIMENSIONS**
(Unit of Measure = mm)

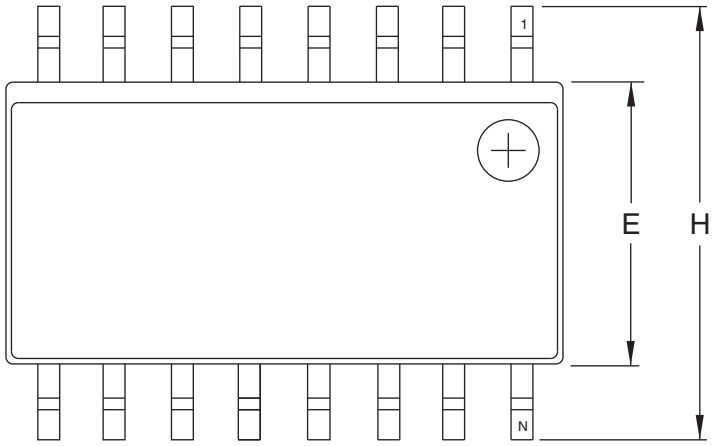
| SYMBOL | MIN | NOM | MAX | NOTE |
|---|---|---|---|---|
| A | – | – | 5.334 | |
| A1 | 0.381 | – | – | |
| D | 19.81 | – | 20.32 | Note 2 |
| E | 7.620 | – | 8.255 | |
| E1 | 6.096 | – | 7.112 | Note 2 |
| B | 0.356 | – | 0.559 | |
| B1 | 1.143 | – | 1.778 | |
| B2 | 0.762 | – | 1.143 | |
| L | 2.921 | – | 3.810 | |
| C | 0.203 | – | 0.356 | |
| eB | – | – | 10.922 | |
| eC | 0.000 | – | 1.524 | |
| e | 2.540 TYP | | | |

Notes: 1. This package conforms to JEDEC reference MS-001, Variation AB.
2. Dimensions D and E1 do not include mold Flash or Protrusion. Mold Flash or Protrusion shall not exceed 0.25 mm (0.010").

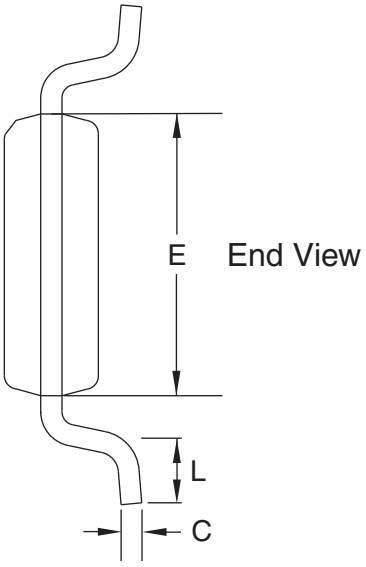
11/02/05

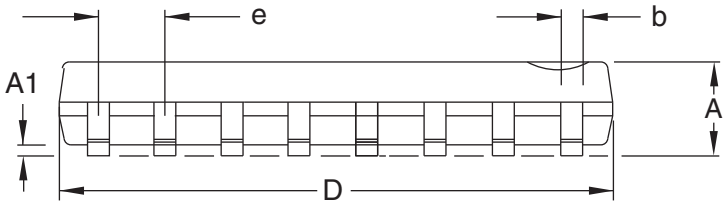
| | TITLE | DRAWING NO. | REV. |
|---|---|---|---|
| 2325 Orchard Parkway<br>San Jose, CA 95131 | **16P3**, 16-lead (0.300"/7.62 mm Wide) Plastic Dual Inline Package (PDIP) | 16P3 | A |

## 27.2 16S – SOIC




Top View




Side View

**COMMON DIMENSIONS**
(Unit of Measure = mm)

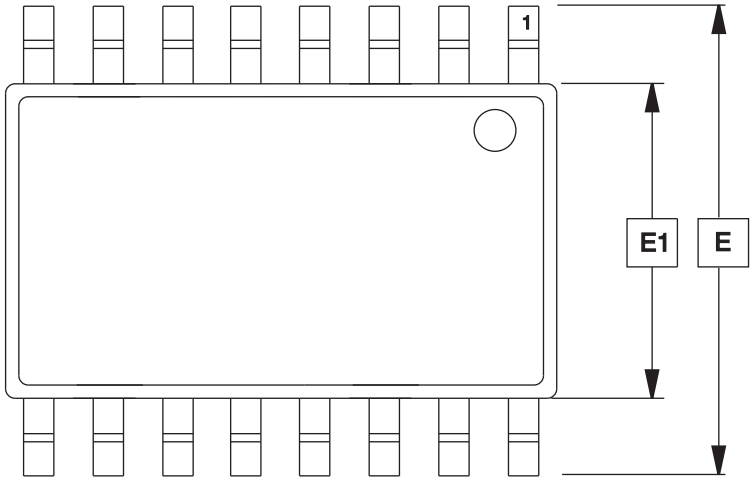
| SYMBOL | MIN | NOM | MAX | NOTE |
|---|---|---|---|---|
| A | 2.35 | – | 2.65 | |
| A1 | 0.10 | – | 0.30 | |
| b | 0.31 | – | 0.51 | |
| D | | 10.30 BSC | | 2 |
| E | | 7.50 BSC | | 3 |
| H | | 10.30 BSC | | |
| L | 0.40 | – | 1.27 | 4 |
| e | | 1.27 BSC | | |
| C | 0.20 | | 0.33 | |

Notes:
1. This drawing is for general information only; refer to JEDEC Drawing MS-013, Variation AA for additional information.
2. Dimension D does not include mold Flash, protrusions or gate burrs. Mold Flash, protrusion and gate burrs shall not exceed 0.15 mm (0.006") per side.
3. Dimension E does not include inter-lead Flash or protrusion. Inter-lead flash and protrusions shall not exceed 0.25 mm (0.010") per side.
4. L is the length of the terminal for soldering to a substrate.

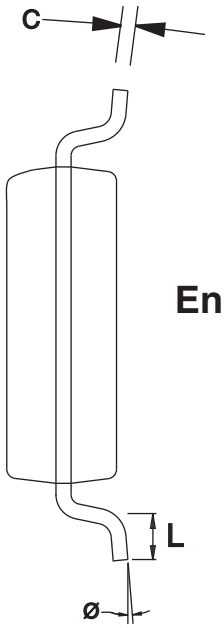
11/02/05

| | TITLE | DRAWING NO. | REV. |
|---|---|---|---|
| 2325 Orchard Parkway<br>San Jose, CA 95131 | **16S**, 16-lead, 0.300" Wide Body, Plastic Gull Wing Small Outline Package (SOIC) | 16S | A |

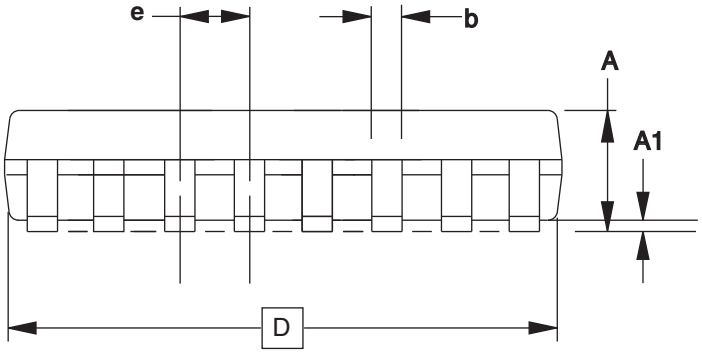
## 27.3 16X – TSSOP




Top View




End View




Side View

**COMMON DIMENSIONS**
(Unit of Measure = mm)

| SYMBOL | MIN | NOM | MAX | NOTE |
|---|---|---|---|---|
| A | – | – | 1.20 | |
| A1 | 0.05 | – | 0.15 | |
| b | 0.19 | – | 0.30 | |
| C | 0.09 | – | 0.20 | |
| D | 4.90 | 5.00 | 5.10 | |
| E1 | 4.30 | 4.40 | 4.50 | |
| E | 6.40 BSC | | | |
| e | 0.65 TYP | | | |
| L | 0.45 | 0.60 | 0.75 | |
| ø | 0° | – | 8° | |

Note: These drawings are for general information only. Refer to JEDEC Drawing MO-153-AB.

11/11/05

| | | | |
|---|---|---|---|
| 2325 Orchard Parkway<br>San Jose, CA 95131 | **TITLE**<br>**16X**, 16-lead, 4.4 mm Body Width, Plastic Thin Shrink Small Outline Package (TSSOP) | **DRAWING NO.**<br>16X | **REV.**<br>A |

## 28. Revision History

| Revision No. | History |
|---|---|
| Revision A – July 2006 | • Initial Preliminary Release |
| Revision B – Nov. 2007 | • Removed “Preliminary” status from the datasheet. |
| Revision C – June 2008 | • Added oscillator connection diagram, Figure 9-1 on page 11<br>• Added external clock connection diagram, Figure 9-2 on page 12<br>• Added PWM Mode 2 waveform, Figure 14-9 on page 34<br>• Updated SPI connection diagram, Figure 19-1 on page 50<br>• Add note for Tristate Ports Fuse on page 71<br>• Updated definition for OCD Interface Fuse on page 71<br>• Updated DC Parameters on page 76<br>• Added typical power consumption characteristics on page 77<br>• Added typical frequency characteristics on page 79<br>• Added typical I/O characteristics on page 80<br>• Updated Clock Parameters on page 83<br>• Removed Standard Packaging Offering |
| Revision D – Oct. 2009 | • Replaced C1 with R1 in oscillator diagram Figure 9-1 on page 11.<br>• Added oscillator input characteristics on page 82. |
| Revision E – Nov. 2010 | • Noted output levels as TTL. See Section 13. on page 22. |

# Table of Contents

*Features* ..... *1*

*1 Description* ..... *1*

*2 Pin Configuration* ..... *2*

2.1 16-lead PDIP/SOIC/TSSOP ..... 2

*3 Pin Description* ..... *3*

*4 Block Diagram* ..... *4*

*5 Comparison to Standard 8051* ..... *4*

5.1 System Clock ..... 4

5.2 Instruction Execution with Single-cycle Fetch ..... 4

5.3 Interrupt Handling ..... 5

5.4 Timer/Counters ..... 5

5.5 Serial Port ..... 5

5.6 Watchdog Timer ..... 5

5.7 I/O Ports ..... 5

5.8 Reset ..... 5

*6 Memory Organization* ..... *6*

6.1 Program Memory ..... 6

6.2 Data Memory ..... 7

*7 Special Function Registers* ..... *8*

*8 Enhanced CPU* ..... *9*

8.1 Restrictions on Certain Instructions ..... 10

*9 System Clock* ..... *11*

9.1 Crystal Oscillator ..... 11

9.2 External Clock Source ..... 12

9.3 Internal RC Oscillator ..... 12

9.4 System Clock Out ..... 12

*10 Reset* ..... *13*

10.1 Power-on Reset ..... 13

10.2 Brown-out Reset ..... 15

10.3 External Reset ..... 15

10.4 Watchdog Reset ..... 16

## Table of Contents (Continued)


10.5 Software Reset ..... 16

***11 Power Saving Modes ..... 16***

11.1 Idle Mode ..... 16

11.2 Power-down Mode ..... 16

***12 Interrupts ..... 18***

12.1 Interrupt Response Time ..... 20

***13 I/O Ports ..... 22***

13.1 Port Configuration ..... 22

13.2 Port 1 Analog Functions ..... 25

13.3 Port Read-modify-write ..... 25

13.4 Ports During Powerdown ..... 26

13.5 Port Alternate Functions ..... 26

***14 Enhanced Timer/Counters ..... 27***

14.1 Mode 0 – Variable Width Timer/Counter ..... 27

14.2 Mode 1 – 16-bit Auto-Reload Timer/Counter ..... 28

14.3 Mode 2 – 8-bit Auto-reload Timer/Counter ..... 29

14.4 Mode 3 – 8-bit Split Timer ..... 29

14.5 Pulse Width Modulation ..... 32

***15 External Interrupts ..... 36***

***16 General-purpose Interrupts ..... 36***

***17 Serial Interface ..... 38***

17.1 Multiprocessor Communications ..... 38

***18 Baud Rates ..... 40***

18.1 More About Mode 0 ..... 41

18.2 More About Mode 1 ..... 43

18.3 More About Modes 2 and 3 ..... 45

18.4 Framing Error Detection ..... 48

18.5 Automatic Address Recognition ..... 48

***19 Serial Peripheral Interface ..... 49***

***20 Analog Comparator ..... 55***

20.1 Comparator Interrupt with Debouncing ..... 55

## 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***

## Headquarters

***Atmel Corporation***
2325 Orchard Parkway
San Jose, CA 95131
USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

## International

***Atmel Asia***
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

***Atmel Europe***
Le Krebs
8, Rue Jean-Pierre Timbaud
BP 309
78054 Saint-Quentin-en-
Yvelines Cedex
France
Tel: (33) 1-30-60-70-00
Fax: (33) 1-30-60-71-11

***Atmel Japan***
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

## Product Contact

***Web Site***
www.atmel.com

***Technical Support***
mcu@atmel.com

***Sales Contact***
www.atmel.com/contacts

***Literature Requests***
www.atmel.com/literature

**Disclaimer:** The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. **EXCEPT AS SET FORTH IN ATMEL'S TERMS AND CONDITIONS OF SALE LOCATED ON ATMEL'S WEB SITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.** Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel's products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.

**© 2008 Atmel Corporation. All rights reserved.** Atmel®, logo and combinations thereof, and others are registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others.

## Looking for pricing, stock, or lifecycle information?

Click below to explore more details on WIN SOURCE:

- View AT89LP216-20PU on WIN SOURCE
- Microchip Technology Information

## Optimize Your Supply Chain with WIN SOURCE Solutions

- Global Sourcing Solution
- Obsolete Management
- Cost Control Management
- Shortage Management
- Alternative Solution
- Excess Inventory Management