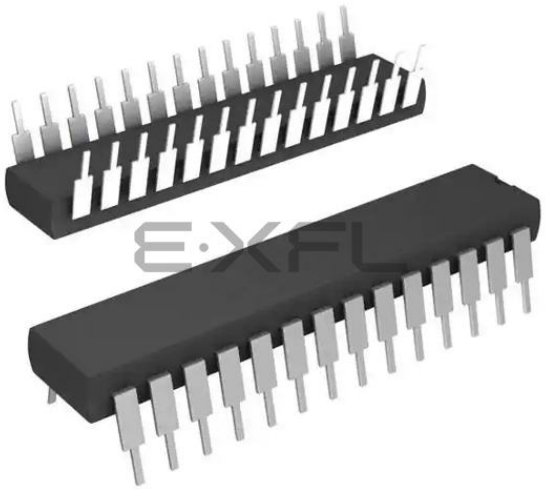


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	8051
Core Size	8-Bit
Speed	20MHz
Connectivity	SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	30
Program Memory Size	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	768 x 8
Voltage - Supply (Vcc/Vdd)	2.4V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	28-DIP (0.300", 7.62mm)
Supplier Device Package	28-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at89lp428-20pu

1.5 Pin Description

Table 1-1. AT89LP428/828 Pin Description

Pin Number			Symbol	Type	Description
TQFP /MLF	PLCC	PDIP			
1	5	5	P4.1	I/O O O I/O	P4.1: User-configurable I/O Port 4 bit 1. 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. CLKOUT: When the internal RC oscillator is selected as the clock source, may be used to output the internal clock divided by 2. DDA: Serial Data input/output for On-chip Debug Interface when OCD is enabled and the external clock is selected as the clock source.
2	6	6	P4.0	I/O I I/O	P4.0: User-configurable I/O Port 4 bit 0. 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. 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.
3	7	N/A	P4.5	I/O	P4.5: User-configurable I/O Port 4 bit 5.
4	8	7	GND	I	Ground
5	9	N/A	P4.4	I/O	P4.4: User-configurable I/O Port 4 bit 4.
6	10	8	P3.2	I/O I	P3.2: User-configurable I/O Port 3 bit 2. $\overline{\text{INT0}}$: External Interrupt 0 Input or Timer 0 Gate Input.
7	11	9	P3.3	I/O I	P3.3: User-configurable I/O Port 3 bit 3. INT1: External Interrupt 1 Input or Timer 1 Gate Input
8	12	10	P3.4	I/O I/O	P3.4: User-configurable I/O Port 3 bit 4. T1: Timer/Counter 0 External input or PWM output.
9	13	11	P3.5	I/O I/O	P3.5: User-configurable I/O Port 3 bit 5. T1: Timer/Counter 1 External input or PWM output.
10	14	12	P3.6	I/O I I	P3.6: User-configurable I/O Port 3 bit 6 (if Reset Fuse is disabled). $\overline{\text{RST}}$: External Active-low Reset input (if Reset Fuse is enabled, see “External Reset” on page 26). DCL: Serial Clock input for On-chip Debug Interface when OCD is enabled.
11	15	13	P2.3	I/O I/O	P2.3: User-configurable I/O Port 2 bit 3. CCD: Timer 2 Channel D Compare Output or Capture Input.
12	16	14	P2.1	I/O I/O	P2.2: User-configurable I/O Port 2 bit 2. CCC: Timer 2 Channel C Compare Output or Capture Input.
13	17	15	P2.1	I/O I/O	P2.1: User-configurable I/O Port 2 bit 1. CCB: Timer 2 Channel B Compare Output or Capture Input.
14	18	16	P2.0	I/O I/O	P2.0: User-configurable I/O Port 2 bit 0. CCA: Timer 2 Channel A Compare Output or Capture Input.
15	19	17	P3.7	I/O I/O	P3.7: User-configurable I/O Port 3 bit 7. DDA: Serial Data input/output for On-chip Debug Interface when OCD is enabled and the Crystal oscillator is selected as the clock source.
16	20	18	P1.0	I/O I/O I	P1.0: User-configurable I/O Port 1 bit 0. T2: Timer 2 External Input or Clock Output. GPI0: General-purpose Interrupt input 0.

Table 3-4. MEMCON – Memory Control Register

MEMCON = 96H — Reset Value = 0000 00XXB

Not Bit Addressable

	IAP	AERS	LDPG	MWEN	DMEN	ABORT	–	$\overline{\text{WRTINH}}$
Bit	7	6	5	4	3	2	1	0

Symbol	Function
IAP	In-Application Programming Enable. When IAP = 1 and the IAP Fuse is enabled, programming of the CODE/SIG space is enabled and MOVX @DPTR instructions will access CODE/SIG instead of EDATA or FDATA. Clear IAP to disable programming of CODE/SIG and allow access to EDATA and FDATA.
AERS	Auto-Erase Enable. Set to perform an auto-erase of a Flash memory page (CODE, SIG or FDATA) during the next write sequence. Clear to perform write without erase.
LDPG	Load Page Enable. Set to this bit to load multiple bytes to the temporary page buffer. Byte locations may not be loaded more than once before a write. LDPG must be cleared before writing.
MWEN	Memory Write Enable. Set to enable programming of a nonvolatile memory location (CODE, SIG or FDATA). Clear to disable programming of all nonvolatile memories.
DMEN	Data Memory Enable. Set to enable nonvolatile data memory and map it into the FDATA space. Clear to disable nonvolatile data memory.
ABORT	Abort Flag. Set by hardware if an error occurred during the last programming sequence due to a brownout condition (low voltage on V_{CC}). Must be cleared by software.
$\overline{\text{WRTINH}}$	Write Inhibit Flag. Cleared by hardware when the voltage on V_{CC} has fallen below the minimum programming voltage. Set by hardware when the voltage on V_{CC} is above the minimum programming voltage.

3.4 In-Application Programming (IAP)

The AT89LP428/828 supports In-Application Programming (IAP), allowing the program memory to be modified during execution. The IAP can be used to modify the user application on-the-fly or to use program memory for nonvolatile data storage. The same write protocol for FDATA also applies to IAP (see “Write Protocol” on page 12). The CPU is always placed in idle while modifying the program memory. When the write completes, the CPU will continue executing with the instruction after the MOVX @DPTR,A instruction that started the write.

To enable access to the program memory, the IAP bit (MEMCON.7) must be set to one and the IAP User Fuse must be enabled. The IAP User Fuse can disable all IAP operations. When this fuse is disabled, the IAP bit will be forced to 0. While IAP is enabled, all MOVX @DPTR instructions will access the CODE space instead of EDATA or FDATA. The IAP also allows reprogramming of the User Signature Array when SIGEN = 1. The IAP access settings are summarized in Table 3-5.

Table 3-5. IAP Access Settings

IAP	SIGEN	DMEN	MOVX @DPTR	MOVC @DPTR
0	0	0	EDATA	CODE
0	0	1	FDATA	CODE
0	1	0	EDATA	SIG
0	1	1	FDATA	SIG
1	0	X	CODE	CODE
1	1	X	SIG	SIG

For assemblers that do not support this notation, the 0A5H prefix must be declared in-line:

```
EX:    DB    0A5H
       INC   DPTR           ; equivalent to INC /DPTR
```

A summary of data pointer instructions with fast context switching is listed in Table 5-1.

Table 5-1. Data Pointer Instructions

Instruction	Operation	
	DPS = 0	DPS = 1
JMP @A+DPTR	JMP @A+DPTR0	JMP @A+DPTR1
MOV DPTR, #data16	MOV DPTR0, #data16	MOV DPTR1, #data16
MOV/DPTR, #data16	MOV DPTR1, #data16	MOV DPTR0, #data16
INC DPTR	INC DPTR0	INC DPTR1
INC/DPTR	INC DPTR1	INC DPTR0
MOVC A,@A+DPTR	MOVC A,@A+DPTR0	MOVC A,@A+DPTR1
MOVC A,@A+/DPTR	MOVC A,@A+DPTR1	MOVC A,@A+DPTR0
MOVX A,@DPTR	MOVX A,@DPTR0	MOVX A,@DPTR1
MOVX A,@/DPTR	MOVX A,@DPTR1	MOVX A,@DPTR0
MOVX @DPTR, A	MOVX @DPTR0, A	MOVX @DPTR1, A
MOVX @/DPTR, A	MOVX @DPTR1, A	MOVX @DPTR0, A

5.1.1 Data Pointer Update

The Dual Data Pointers on the AT89LP428/828 include two additional features that control how the data pointers are updated. The data pointer decrement bits, DPD1 and DPD0 in DPCF, configure the INC DPTR instruction to act as DEC DPTR. The resulting operation will depend on DPS as shown in Table 5-2.

Table 5-2. INC DPTR Behavior

DPD1	DPD0	Operation			
		DPS = 0		DPS = 1	
		INC DPTR	INC /DPTR	INC DPTR	INC /DPTR
0	0	INC DPTR0	INC DPTR1	INC DPTR1	INC DPTR0
0	1	DEC DPTR0	INC DPTR1	INC DPTR1	DEC DPTR0
1	0	INC DPTR0	DEC DPTR1	DEC DPTR1	INC DPTR0
1	1	DEC DPTR0	DEC DPTR1	DEC DPTR1	DEC DPTR0

The data pointer update bits, DPU1 and DPU0, allow MOVX @DPTR and MOVC @DPTR instructions to update the selected data pointer automatically in a post-increment or post-decrement fashion. The direction of update depends on the DPD1 and DPD0 bits as shown in Table 5-3.

5.2 Restrictions on Certain Instructions

The AT89LP428/828 is an economical and cost-effective member of Atmel's growing family of microcontrollers. It contains 4K/8K 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 0000H–0FFFH for the AT89LP428 and 0000H–1FFFH for the AT89LP828. This should be the responsibility of the software programmer. For example, LJMP 07E0H would be a valid instruction, whereas LJMP 9000H would not. 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.

5.2.1 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. 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.

5.2.2 MOVX-related Instructions

The AT89LP428/828 contains 512 bytes of internal Extra RAM and 512/1024 bytes of Flash data memory mapped into the XRAM address space. MOVX accesses to addresses above 03FFH/05FFH will return invalid data.

6. 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 on-chip crystal oscillator may also be configured for low and high speed operation. The clock source is selected by the Clock Source User Fuses as shown in Table 6-1. See “User Configuration Fuses” on page 121. By default, no internal clock division is used to generate the CPU clock from the system clock. However, the system clock divider may be used to prescale the system clock. The choice of clock source also affects the start-up time after a POR, BOD or Power-down event (see “Reset” on page 23 or “Power-down Mode” on page 27).

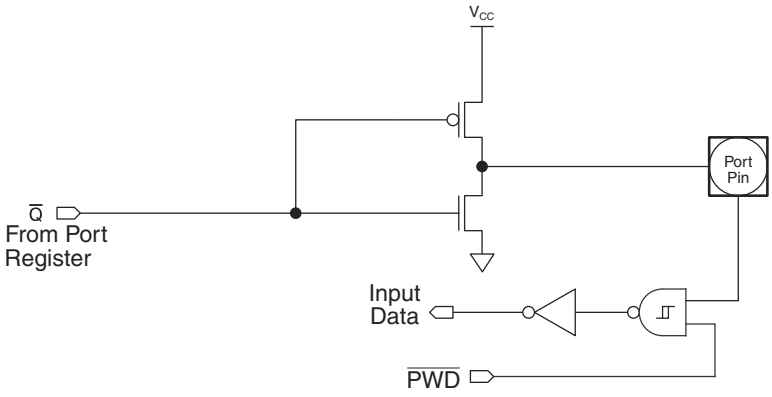
Table 6-1. Clock Source Settings

Clock Source Fuse 1	Clock Source Fuse 0	Selected Clock Source
0	0	High Speed Crystal Oscillator (f > 500 kHz)
0	1	Low Speed Crystal Oscillator (f ≤100 kHz)
1	0	External Clock on XTAL1
1	1	Internal 8 MHz RC Oscillator

10.1.4 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. The push-pull port configuration is shown in Figure 10-5.

Figure 10-5. Push-pull Output



10.2 Port 2 Analog Functions

The AT89LP428/828 incorporates two analog comparators. 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 inputs disabled. Digital outputs are disabled by putting the port pins into the input-only mode as described in “Port Configuration” on page 35. Digital inputs on P2.4, P2.5, P2.6 and P2.7 are disabled whenever an analog comparator is enabled by setting the CENA or CENB bits in ACSRA and ACSRB and that pin is configured for input-only mode. To use an analog input pin as a high-impedance digital input while a comparator is enabled, that pin should be configured in open-drain mode and the corresponding port register bit should be set to 1. The analog input pins will always default to input-only mode after reset regardless of the state of the Tristate-Port User Fuse.

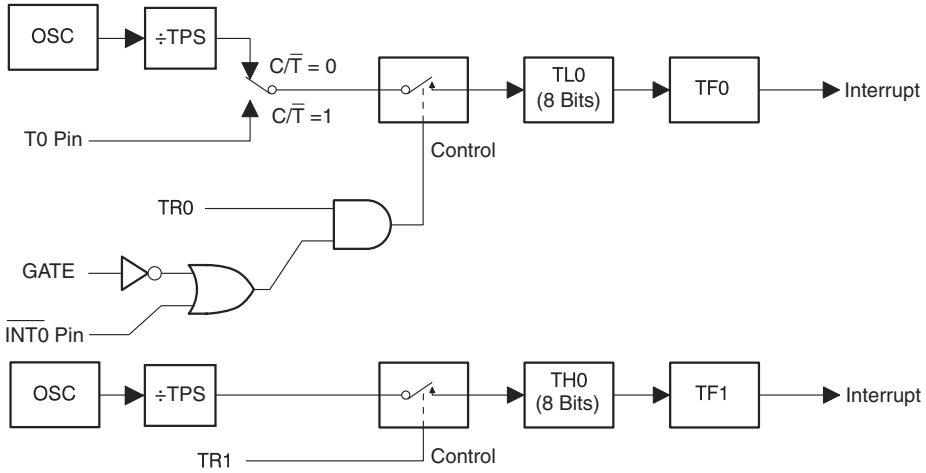
If analog noise immunity is a concern, the P2.4–7 pins should not be used as high speed digital inputs or outputs while the comparators are enabled.

11.4 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 11-4. TL0 uses the Timer 0 control bits: C/T, GATE, TR0, $\overline{INT0}$, and TF0. TH0 is locked into a timer function (counting clock 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.

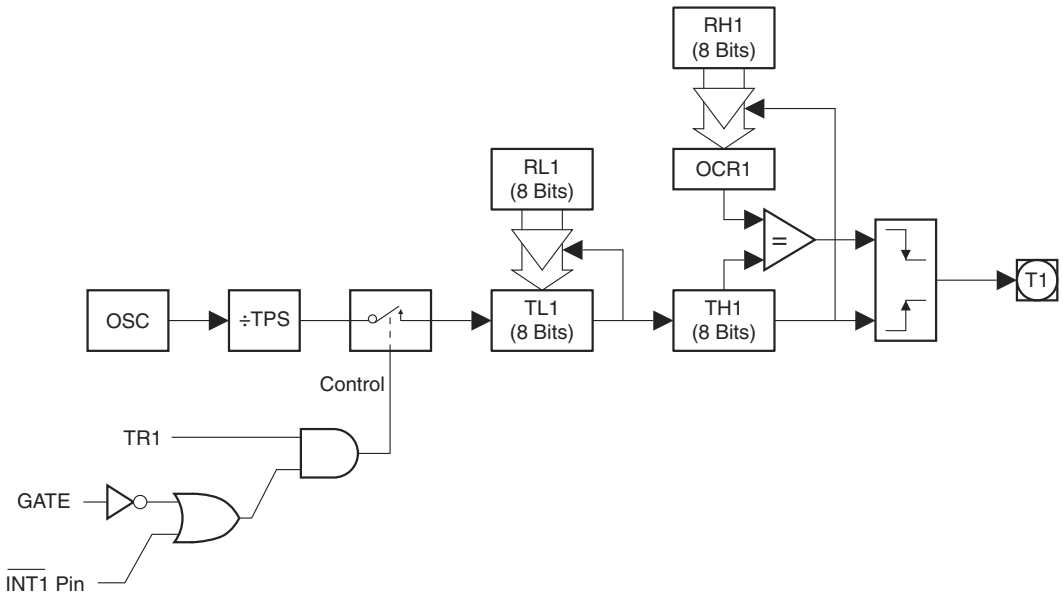
Mode 3 is for applications requiring an extra 8-bit timer or counter. With Timer 0 in Mode 3, the AT89LP428/828 can appear to have three Timer/Counters. When Timer 0 is in Mode 3, Timer 1 can be turned on and off by switching it out of and into its own Mode 3. In this case, Timer 1 can still be used by the serial port as a baud rate generator or in any application not requiring an interrupt.

Figure 11-4. 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.

Figure 11-7. Timer/Counter 1 PWM Mode 1

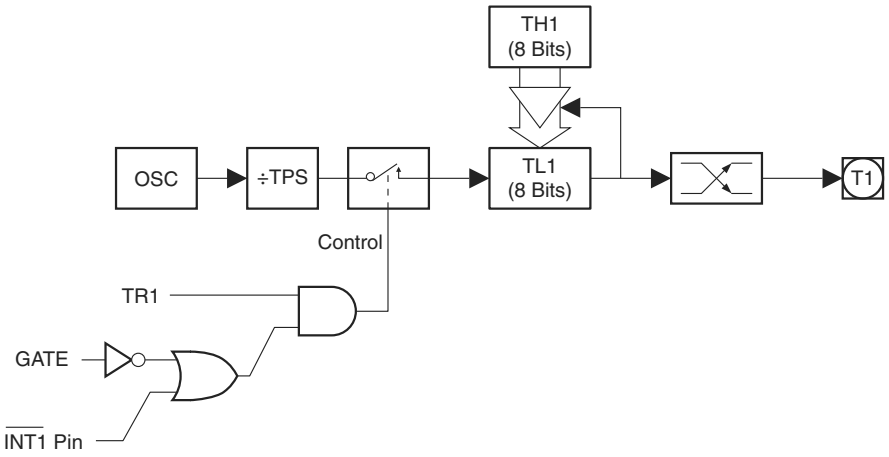


11.5.3 Mode 2 – 8-bit Frequency Generator

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 Figure 11-8 and Figure 11-9 on page 50). 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.

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

Figure 11-8. 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.

12. Enhanced Timer 2

The AT89LP428/828 includes a 16-bit Timer/Counter 2 with the following features:

- 16-bit timer/counter with one 16-bit reload/capture register
- One external reload/capture input
- Up/Down counting mode with external direction control
- UART baud rate generation
- Output-pin toggle on timer overflow
- Dual slope symmetric operating modes

Timer 2 is a 16-bit Timer/Counter that can operate as either a timer or an event counter. The type of operation is selected by bit C/$\overline{T2}$ in the SFR T2CON. Timer 2 has three operating modes: capture, auto-reload (up or down counting), and baud rate generator. The modes are selected by bits in T2CON and T2MOD, as shown in Table 12-3. Timer 2 also serves as the time base for the Compare/Capture Array (see “Compare/Capture Array” on page 61).

Timer 2 consists of two 8-bit registers, TH2 and TL2. In the Timer function, the register is incremented every clock cycle. 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 6-2 on page 23).

In the Counter function, the register is incremented in response to a 1-to-0 transition at its corresponding external input pin, T2. In this function, 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 two clock cycles are required to recognize a 1-to-0 transition, the maximum count rate is 1/2 of the oscillator frequency. To ensure that a given level is sampled at least once before it changes, the level should be held for at least one full clock cycle.

Table 12-1. Timer 2 Operating Modes

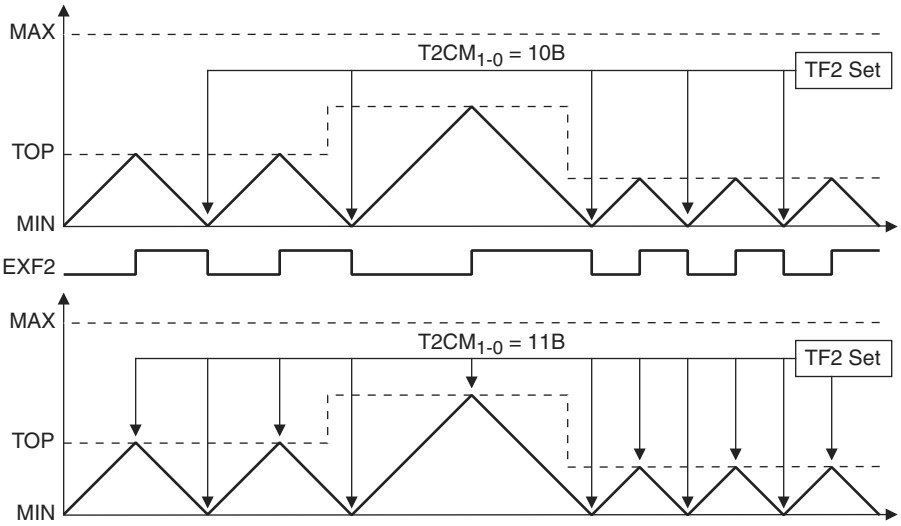
RCLK + TCLK	CP/$\overline{RL2}$	DCEN	T2OE	TR2	Mode
0	0	0	0	1	16-bit Auto-reload
0	0	1	0	1	16-bit Auto-reload Up-down
0	1	X	0	1	16-bit Capture
1	X	X	X	1	Baud Rate Generator
X	X	X	1	1	Frequency Generator
X	X	X	X	0	(Off)

The following definitions for Timer 2 are used in the subsequent paragraphs:

Table 12-2. Timer 2 Definitions

Symbol	Definition
MIN	0000H
MAX	FFFFH
BOTTOM	16-bit value of {RCAP2H,RCAP2L} (standard modes)
TOP	16-bit value of {RCAP2H,RCAP2L} (enhanced modes)

Figure 12-6. Timer 2 Waveform: Dual Slope Modes



12.4 Baud Rate Generator

Timer 2 is selected as the baud rate generator by setting TCLK and/or RCLK in T2CON (Table 12-3 on page 53). Note that the baud rates for transmit and receive can be different if Timer 2 is used for the receiver or transmitter and Timer 1 is used for the other function. Setting RCLK and/or TCLK puts Timer 2 into its baud rate generator mode, as shown in Figure 12-7.

The baud rate generator mode is similar to the auto-reload mode, in that a rollover in TH2 causes the Timer 2 registers to be reloaded with the 16-bit value in registers RCAP2H and RCAP2L, which are preset by software.

The baud rates in UART modes 1 and 3 are determined by Timer 2's overflow rate according to the following equation.

$$\text{Modes 1 and 3 Baud Rates} = \frac{\text{Timer 2 Overflow Rate}}{16}$$

The Timer can be configured for either timer or counter operation. In most applications, it is configured for timer operation ($C/\overline{T2} = 0$). The baud rate formulas are given below

.

$$\text{T2CM} = 00\text{B} \qquad \frac{\text{Modes 1, 3}}{\text{Baud Rate}} = \frac{\text{Oscillator Frequency}}{16 \times (\text{TPS} + 1) \times [65536 - (\text{RCAP2H,RCAP2L})]}$$

$$\text{T2CM} = 01\text{B} \qquad \frac{\text{Modes 1, 3}}{\text{Baud Rate}} = \frac{\text{Oscillator Frequency}}{16 \times (\text{TPS} + 1) \times [(\text{RCAP2H,RCAP2L}) + 1]}$$

where (RCAP2H, RCAP2L) is the content of RCAP2H and RCAP2L taken as a 16-bit unsigned integer.

Table 13-5. T2CCF – Timer/Counter 2 Compare/Capture Flags

T2CCF Address = 0D5H — Reset Value = XXXX 0000B

Not Bit Addressable

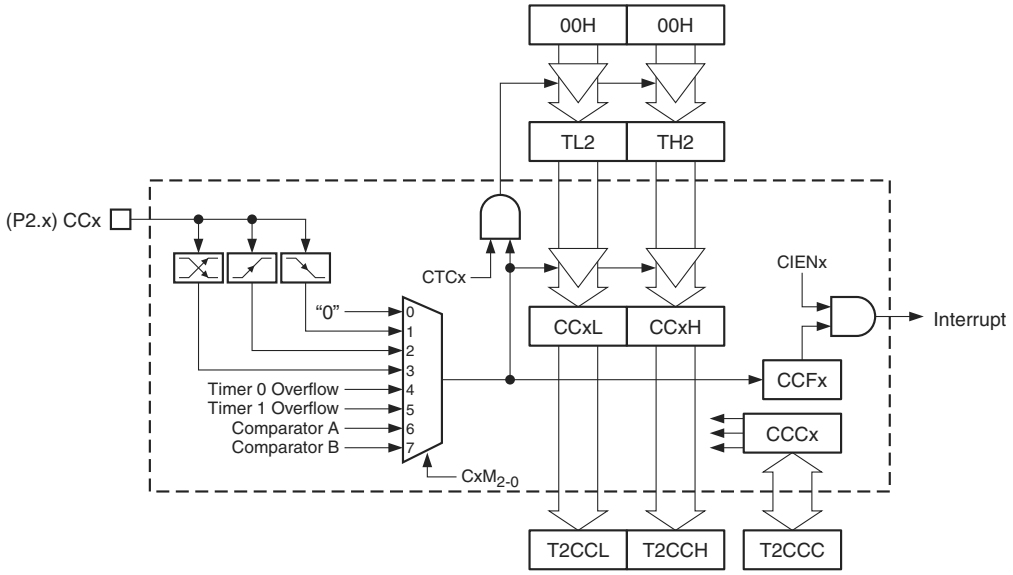
	–	–	–	–	CCFD	CCFC	CCFB	CCFA
Bit	7	6	5	4	3	2	1	0

Symbol	Function
CCFD	Channel D Compare/Capture Interrupt Flag. Set by a compare/capture event on channel D. Must be cleared by software. CCFD will generate an interrupt when CIEND = 1 and ECC = 1.
CCFC	Channel C Compare/Capture Interrupt Flag. Set by a compare/capture event on channel C. Must be cleared by software. CCFC will generate an interrupt when CIENC = 1 and ECC = 1.
CCFB	Channel B Compare/Capture Interrupt Flag. Set by a compare/capture event on channel B. Must be cleared by software. CCFB will generate an interrupt when CIENB = 1 and ECC = 1.
CCFA	Channel A Compare/Capture Interrupt Flag. Set by a compare/capture event on channel A. Must be cleared by software. CCFA will generate an interrupt when CIENA = 1 and ECC = 1.

13.2 Input Capture Mode

The Compare/Capture Array provides a variety of capture modes suitable for time-stamping events or performing measurements of pulse width, frequency, slope, etc. The CCA channels are configured for capture mode by clearing the CCM*x* bit in the associated CCC*x* register to 0. Each time a capture event occurs, the contents of Timer 2 (TH2 and TL2) are transferred to the 16-bit data register of the corresponding channel, and the channel's interrupt flag CCF*x* is set in T2CCF. Optionally, the capture event may also clear Timer 2 to 0000H by setting the CTC*x* bit in CCC*x*. The capture event is defined by the C*x*M$_{2-0}$ bits in CCC*x* and may be either externally or internally generated. A diagram of a CCA channel in capture mode is shown in Figure 13-2.

Figure 13-2. CCA Capture Mode Diagram



16. Serial Interface (UART)

The serial interface on the AT89LP428/828 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 or Timer 2
- 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 programmable to 1/2 or 1/4 the oscillator frequency, or variable based on Time 1.
- **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 or Timer 2.
- **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 or Timer 3 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.

16.1 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

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 treated 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.

17. Enhanced Serial Peripheral Interface

The Serial Peripheral Interface (SPI) allows high-speed synchronous data transfer between the AT89LP428/828 and peripheral devices or between multiple AT89LP428/828 devices, including multiple masters and slaves on a single bus. The SPI includes the following features:

- Full-duplex, 3-wire or 4-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 or Timer 1-based Baud Generation (Master Mode)
- End of Transmission Interrupt Flag
- Write Collision Flag Protection
- Double-buffered Receive and Transmit
- Transmit Buffer Empty Interrupt Flag
- Mode Fault (Master Collision) Detection and Interrupt
- Wake up from Idle Mode

A block diagram of the SPI is shown in Figure 17-1.

Table 18-3. AREF – Analog Comparator Reference Control Register

AREF = AFH — Reset Value = 0000 0000B

Not Bit Addressable

	CBC1	CBC0	RFB1	RFB0	CAC1	CAC0	RFA1	RFA0
Bit	7	6	5	4	3	2	1	0

Symbol	Function		
CSC [1 - 0]	Comparator B Clock Select		
	CBC1	**CBC0**	**Clock Source**
	0	0	System Clock
	0	0	Timer 0 Overflow
	0	1	Timer 1 Overflow
	0	1	Timer 2 Overflow
RFB [1 - 0]	Comparator B Negative Input Channel Select(1)		
	CRF1	**RFB0**	**B-channel**
	0	0	AIN2 (P2.6)
	0	0	Internal $V_{AREF-\Delta}$ (~1.2V)
	0	1	Internal V_{AREF} (~1.3V)
	0	1	Internal $V_{AREF+\Delta}$ (~1.4V)
CAC [1 - 0]	Comparator A Clock Select		
	CAC1	**CAC0**	**Clock Source**
	0	0	System Clock
	0	0	Timer 0 Overflow
	0	1	Timer 1 Overflow
	0	1	Timer 2 Overflow
RFB [1 - 0]	Comparator A Negative Input Channel Select(2)		
	RFA1	**RFA0**	**A-channel**
	0	0	AIN1 (P2.5)
	0	0	Internal $V_{AREF-\Delta}$ (~1.2V)
	0	1	Internal V_{AREF} (~1.3V)
	0	1	Internal $V_{AREF+\Delta}$ (~1.4V)

Notes: 1. CONB (ACSRB.5) must be cleared to 0 before changing RFB [1 - 0].
2. CONA (ACSRA.5) must be cleared to 0 before changing RFA [1 - 0].

19.1 Software Reset

A Software Reset of the AT89LP428/828 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 19-2. 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 PS1 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 19-3. WDTRST – Watchdog Reset Register

WDTRST 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.

20. Instruction Set Summary

The AT89LP428/828 is fully binary compatible with the MCS-51 instruction set. The difference between the AT89LP428/828 and the standard 8051 is the number of cycles required to execute an instruction. Instructions in the AT89LP428/828 may take 1, 2, 3, 4 or 5 clock cycles to complete. The execution times of most instructions may be computed using Table 20-1 on page 107.

Table 20-1. 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	2/4[2]
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
INC /DPTR[1]	2	–	3	A5 A3

Table 21-1. Special Function Register Cross Reference (Continued)

PSW	D0H	
RCAP2H	CBH	Section 12.1 on page 53
RCAP2L	CAH	Section 12.1 on page 53
RH0	94H	Table 11-1 on page 41
RH1	95H	Table 11-1 on page 41
RL0	92H	Table 11-1 on page 41
RL1	93H	Table 11-1 on page 41
SADDR	A9H	Section 16.7 on page 90
SADEN	B9H	Section 16.7 on page 90
SBUF	99H	Section 16.3 on page 81
SCON	98H	Table 16-1 on page 78
SP	81H	
SPCR	E9H	Table 17-3 on page 97
SPDR	EAH	Table 17-2 on page 97
SPSR	E8H	Table 17-4 on page 98
T2CCA	D1H	Table 13-1 on page 63
T2CCC	D4H	Table 13-4 on page 64
T2CCF	D5H	Table 13-5 on page 65
T2CCH	D3H	Table 13-2 on page 63
T2CCL	D2H	Table 13-3 on page 63
T2CON	C8H	Table 12-3 on page 53
T2MOD	C9H	Table 12-4 on page 54
TCON	88H	Table 11-2 on page 45
TCONB	91H	Table 11-4 on page 47
TH0	8CH	Table 11-1 on page 41
TH1	8DH	Table 11-1 on page 41
TH2	CDH	Section 12.1 on page 53
TL0	8AH	Table 11-1 on page 41
TL1	8BH	Table 11-1 on page 41
TL2	CCH	Section 12.1 on page 53
TMOD	89H	Table 11-3 on page 46
WDTCON	**A7H**	Table 19-2 on page 106
WDTRST	**A6H**	Table 19-3 on page 106

Table 23-3. 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	00bb bbbb	DataIn 0 ... DataIn n	
Write Code Page[2]	0101 0000	000a aaaa	aabb bbbb	DataIn 0 ... DataIn n	
Write Code Page with Auto-Erase[2]	0111 0000	000a aaaa	aabb bbbb	DataIn 0 ... DataIn n	
Read Code Page[2]	0011 0000	000a aaaa	aabb bbbb	DataOut 0 ... DataOut n	
Write Data Page[2]	1101 0000	0000 0aaa	aabb bbbb	DataIn 0 ... DataIn n	
Write Data Page with Auto-Erase[2]	1101 0010	0000 0aaa	aabb bbbb	DataIn 0 ... DataIn n	
Read Data Page[2]	1011 0000	0000 0aaa	aabb bbbb	DataOut 0 ... DataOut n	
Write User Fuses[2][3][4]	1110 0001	0000 0000	00bb bbbb	DataIn 0 ... DataIn n	
Write User Fuses with Auto-Erase[2][3][4]	1111 0001	0000 0000	00bb bbbb	DataIn 0 ... DataIn n	
Read User Fuses[2][3][4]	0110 0001	0000 0000	00bb bbbb	DataOut 0 ... DataOut n	
Write Lock Bits[2][3][5]	1110 0100	0000 0000	00bb bbbb	DataIn 0 ... DataIn n	
Read Lock Bits[2][3][5]	0110 0100	0000 0000	00bb bbbb	DataOut 0 ... DataOut n	
Write User Signature Page[2]	0101 0010	0000 0000	0abb bbbb	DataIn 0 ... DataIn n	
Write User Signature Page with Auto-Erase[2]	0111 0010	0000 0000	0abb bbbb	DataIn 0 ... DataIn n	
Read User Signature Page[2]	0011 0010	0000 0000	0abb bbbb	DataOut 0 ... DataOut n	
Read Atmel Signature Page[2][6]	0011 1000	0000 0000	00bb 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 64 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 23-6 on page 121 for Fuse definitions.
5. See Table 23-5 on page 120 for Lock Bit definitions.
6. **Atmel Signature Bytes**:

Address:	0000H	0001H	0002H
AT89LP428:	1EH	40H	FFH
AT89LP828:	1EH	42H	FFH

7. **Symbol Key**:

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

23.9.5 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. For In-System Programming, bytes are transferred MSB first as shown in Figure 23-9. 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 23-10.

Figure 23-9. ISP Byte Sequence

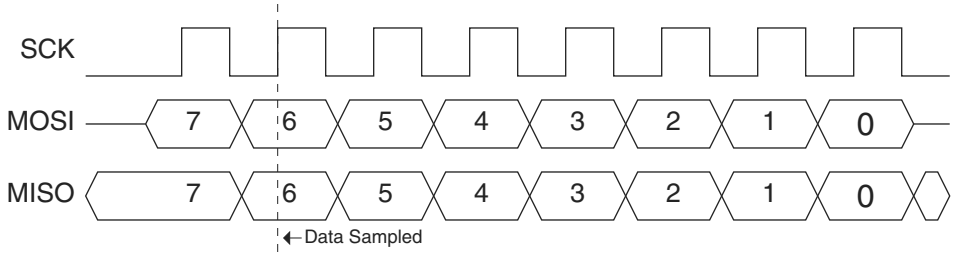


Figure 23-10. Serial Programming Interface Timing