Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	8051
Core Size	8-Bit
Speed	20MHz
Connectivity	SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	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	Surface Mount
Package / Case	32-LCC (J-Lead)
Supplier Device Package	32-PLCC (13.97x11.43)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at89lp428-20ju

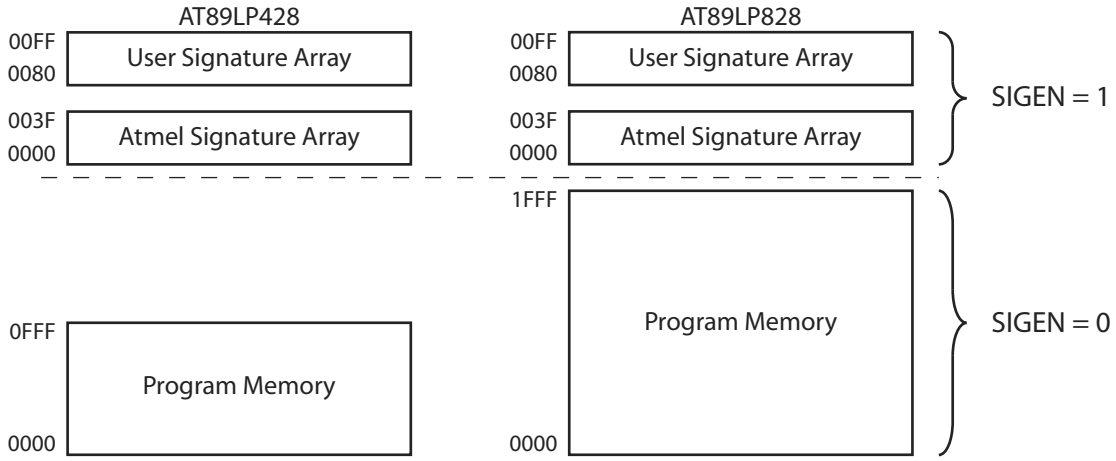
Table 3-2. AT89LP828 Memory Address Spaces

Name	Description	Range
DATA	Directly addressable internal RAM	00H - 7FH
IDATA	Indirectly addressable internal RAM and stack space	00H - FFH
SFR	Directly addressable I/O register space	80H - FFH
EDATA	On-chip Extra RAM	0000H - 01FFH
FDATA	On-chip nonvolatile Flash data memory	0200H - 05FFH
CODE	On-chip nonvolatile Flash program memory	0000H - 1FFFH
SIG	On-chip nonvolatile Flash signature array	0000H - 00FFH

3.1 Program Memory

The AT89LP428/828 contains 4K/8K bytes of on-chip In-System Programmable Flash memory for program storage. The Flash memory has an endurance of at least 100,000 write/erase cycles and a minimum data retention time of 10 years. The reset and interrupt vectors are located within the first 83 bytes of program memory (refer to Table 9-1 on page 30). Constant tables can be allocated within the entire 4K/8K program memory address space for access by the MOVC instruction. The AT89LP428/828 does not support external program memory. A map of the AT89LP428/828 program memory is shown in Figure 3-1.

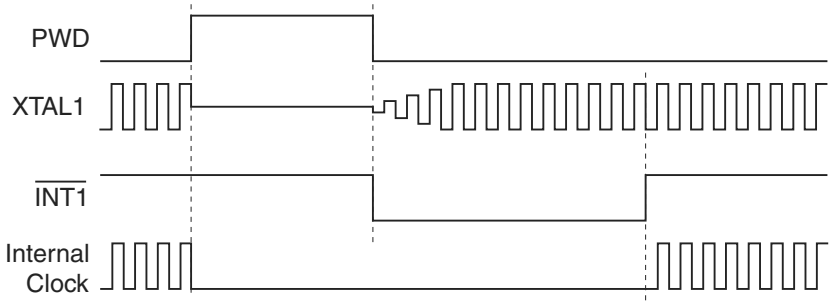
Figure 3-1. Program Memory Map



3.1.1 SIG

In addition to the 4K/8K code space, the AT89LP428/828 also supports a 128-byte User Signature Array and a 64-byte Atmel Signature Array that are accessible by the CPU. The Atmel Signature Array is initialized with the Device ID in the factory. The second page of the User Signature Array (00C0H - 00FFH) is initialized with analog configuration data including the Internal RC Oscillator calibration byte. The User Signature Array is also 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 by an external device programmer are always allowed.

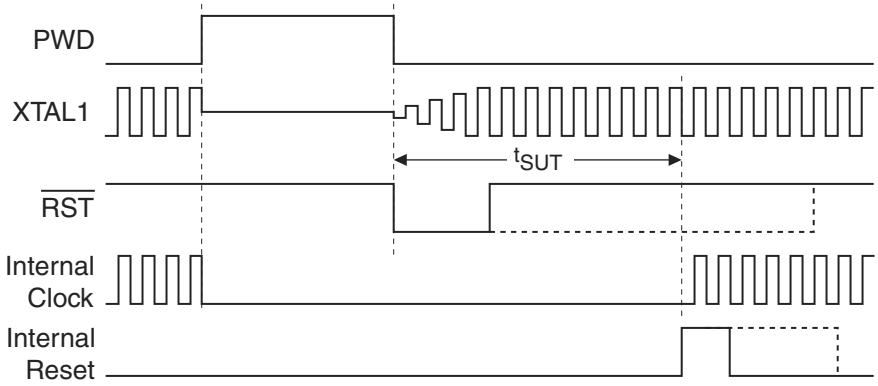
Figure 8-2. Interrupt Recovery from Power-down (PWDEX = 1)



8.2.2 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 8-3. 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 7-1 on page 25). 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 8-3. Reset Recovery from Power-down



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 Timer 2 Interrupt is generated by a logic OR of bits TF2 and EXF2 in register T2CON. Neither of these flags is cleared by hardware when the CPU vectors to the service routine. The service routine normally must determine whether TF2 or EXF2 generated the interrupt and that bit must be cleared by software.

The Serial Port Interrupt is generated by the logic OR of RI and TI in SCON. Neither of these flags is cleared by hardware when the CPU vectors to the service routine. The service routine normally must determine whether RI or TI generated the interrupt and that bit must be cleared by software. The Serial Peripheral Interface Interrupt is generated by the logic OR of SPIF, MODF and TXE in SPSR. None of these flags is cleared by hardware when the CPU vectors to the service routine. The service routine normally must determine which bit generated the interrupt and that bit must be cleared by software.

A logic OR of all eight flags in the GPIF register causes the GPI. 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 that 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 CFA and CFB bits in ACSRA and ACSRB respectively generate the Comparator Interrupt. The service routine must normally determine whether CFA or CFB generated the interrupt, and the bit must be cleared by software.

A logic OR of the four least significant bits in the T2CCF register causes the Compare/Capture Array 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 that bit must be cleared in software.

All 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.

Table 9-1. 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 Interrupt	RI or TI	0023H
Timer 2 Interrupt	TF2 or EXF2	002BH
Analog Comparator Interrupt	CFA or CFB	0033H
General-purpose Interrupt	$GPIF_{7-0}$	003BH
Compare/Capture Array Interrupt	$T2CCF_{3-0}$	0043H
Serial Peripheral Interface Interrupt	SPIF or MODF or TXE	004BH

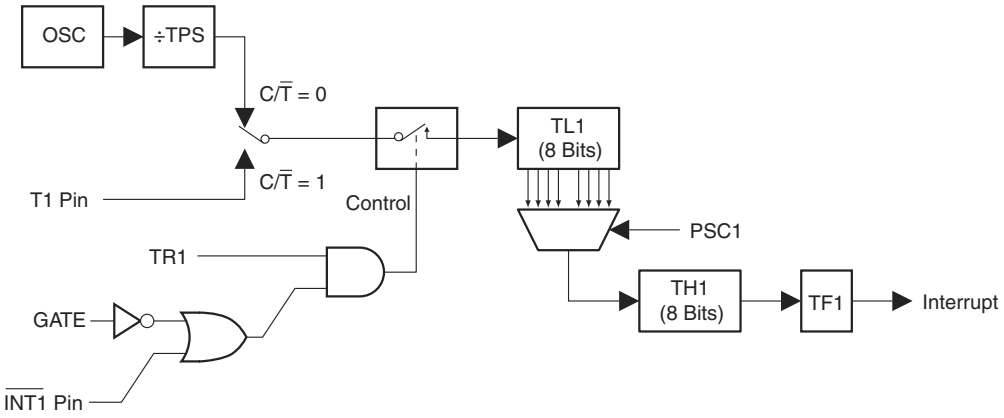
11.1 Mode 0 – Variable Width Timer/Counter

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 11-1 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 11-1. 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 11-1. There are two different GATE bits, one for Timer 1 (TMOD.7) and one for Timer 0 (TMOD.3).

11.2 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 11-2. 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.

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

Table 11-4. 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

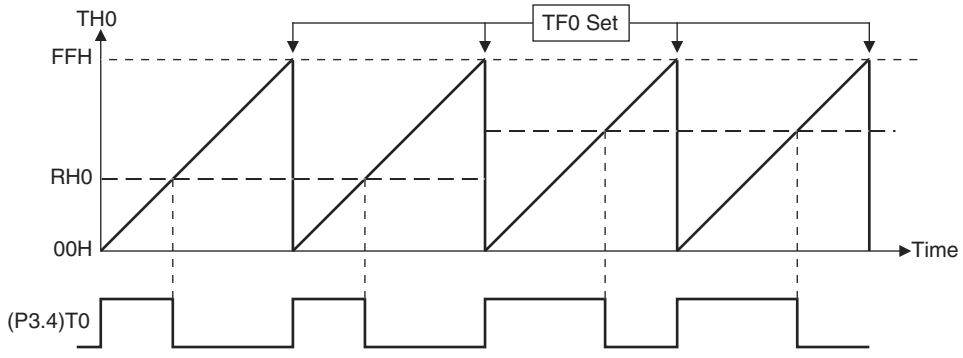
Symbol	Function
PWM1EN	Configures Timer 1 for Pulse Width Modulation output on T1 (P3.5).
PWM0EN	Configures Timer 0 for Pulse Width Modulation output on T0 (P3.4).
$PSC1_{2-0}$	Prescaler for Timer 1 Mode 0. The number of active bits in TL1 equals PSC1 + 1. After reset PSC1 = 100B which enables 5 bits of TL1 for compatibility with the 13-bit Mode 0 in AT89S2051.
$PSC0_{2-0}$	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.

11.5 Pulse Width Modulation

On the AT89LP428/828, 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/$\overline{Tx}$ 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. The timer GATE function also works in PWM mode, allowing the output to be halted by an external input. 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 11-5. 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 11-5. 8-bit Asymmetrical Pulse Width Modulation

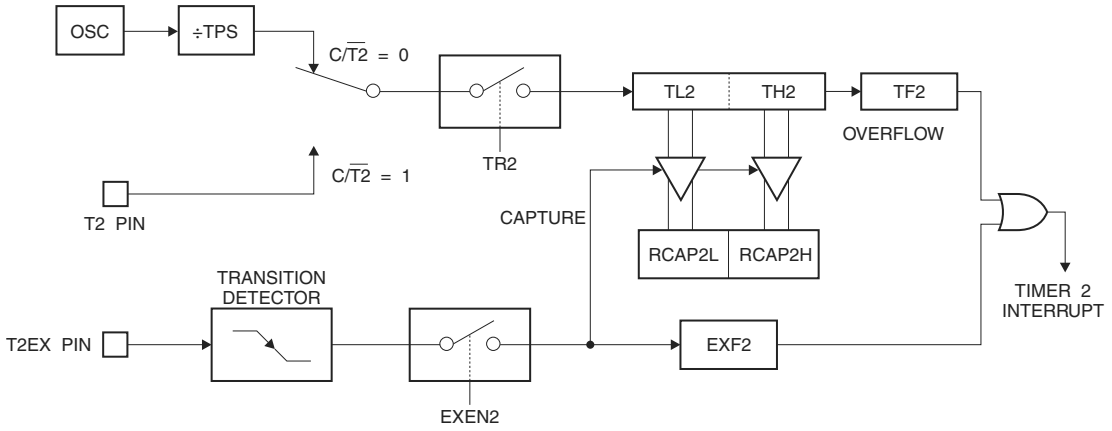


12.2 Capture Mode

In the Capture mode, Timer 2 is a fixed 16-bit timer or counter that counts up from MIN to MAX. An overflow from MAX to MIN sets bit TF2 in T2CON. If EXEN2 = 1, a 1-to-0 transition at external input T2EX also causes the current value in TH2 and TL2 to be captured into RCAP2H and RCAP2L, respectively. In addition, the transition at T2EX causes bit EXF2 in T2CON to be set. The EXF2 and TF2 bits can generate an interrupt. Capture mode is illustrated in Figure 12-1. The Timer 2 overflow rate in Capture mode is given by the following equation:

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

Figure 12-1. Timer 2 Diagram: Capture Mode



12.3 Auto-Reload Mode

Timer 2 can be programmed to count up or down when configured in its 16-bit auto-reload mode. This feature is invoked by the DCEN (Down Counter Enable) bit located in the SFR T2MOD (see Table 12-4). Upon reset, the DCEN bit is set to 0 so that timer 2 will default to count up. When DCEN is set, Timer 2 can count up or down, depending on the value of the T2EX pin. The overflow and reload values depend on the Timer 2 Count Mode bits, $T2CM_{1\text{-}0}$ in T2MOD. A summary of the Auto-reload behaviors is listed in Table 12-5.

Table 12-5. Summary of Auto-reload Modes

$T2CM_{1\text{-}0}$	DCEN	T2EX	Direction	Behavior
00	0	X	Up	BOTTOM →MAX reload to BOTTOM
00	1	0	Down	MAX →BOTTOM underflow to MAX
00	1	1	Up	BOTTOM →MAX overflow to BOTTOM
01	0	X	Up	MIN →TOP reload to MIN
01	1	0	Down	TOP →MIN underflow to TOP
01	1	1	Up	MIN →TOP overflow to MIN
10	X	X	Up-down	MIN →TOP →MIN and repeat
11	X	X	Up-down	MIN →TOP →MIN and repeat

12.3.1 Up Counter

Figure 12-2 shows Timer 2 automatically counting up when DCEN = 0 and $T2CM_{1-0}$ = 00B. In this mode Timer 2 counts up to MAX and then sets the TF2 bit upon overflow. The overflow also causes the timer registers to be reloaded with BOTTOM, the 16-bit value in RCAP2H and RCAP2L. If EXEN2 = 1, a 16-bit reload can be triggered either by an overflow or by a 1-to-0 transition at external input T2EX. This transition also sets the EXF2 bit. Both the TF2 and EXF2 bits can generate an interrupt. The Timer 2 overflow rate for this mode is given in the following equation:

Auto-Reload Mode:
DCEN = 0, T2CM = 00B

$$\text{Time-out Period} = \frac{65536 - \{RCAP2H, RCAP2L\}}{\text{Oscillator Frequency}} \times (TPS + 1)$$

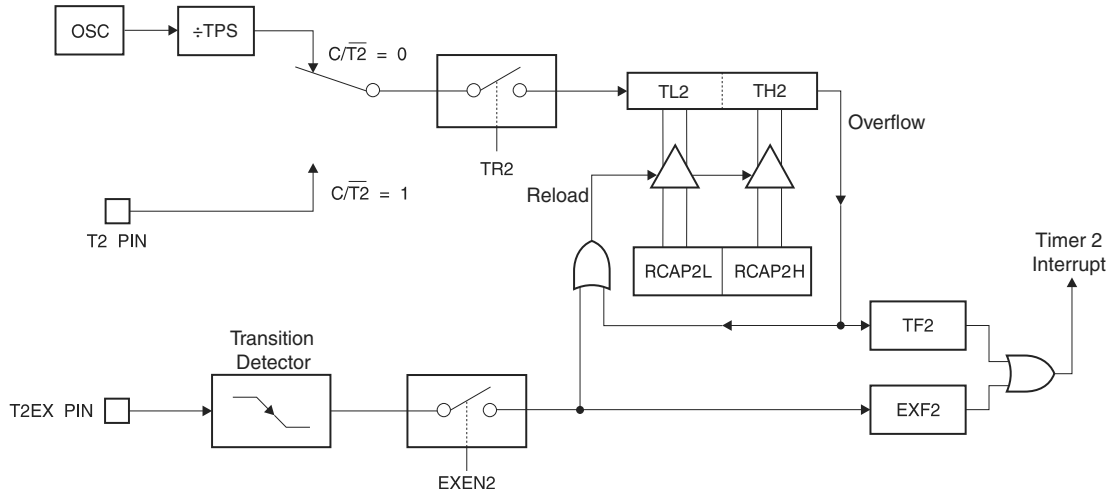
Timer 2 may also be configured to count from MIN to TOP instead of BOTTOM to MAX by setting $T2CM_{1-0}$ = 01B. In this mode Timer 2 counts up to TOP, the 16-bit value in RCAP2H and RCAP2L and then overflows. The overflow sets TF2 and causes the timer registers to be reloaded with MIN. If EXEN2 = 1, a 1-to-0 transition on T2EX will clear the timer and set EXF2. The Timer 2 overflow rate for this mode is given in the following equation:

Auto-Reload Mode:
DCEN = 0, T2CM = 01B

$$\text{Time-out Period} = \frac{\{RCAP2H, RCAP2L\} + 1}{\text{Oscillator Frequency}} \times (TPS + 1)$$

Timer 2 Count Mode 1 is provided to support variable precision asymmetrical PWM in the CCA. The value of TOP stored in RCAP2H and RCAP2L is double-buffered such that a new TOP value takes affect only after an overflow. The behavior of Count Mode 0 versus Count Mode 1 is shown in Figure 12-3.

Figure 12-2. Timer 2 Diagram: Auto-reload Mode (DCEN = 0)



16.2 Baud Rates

The baud rate in Mode 0 depends on the value of the SMOD1 bit in Special Function Register PCON.7. If SMOD1 = 0 (the value on reset) and TB8 = 0, the baud rate is 1/4 of the oscillator frequency. If SMOD1 = 1 and TB8 = 0, the baud rate is 1/2 of the oscillator frequency, as shown in the following equation:

$$\underset{TB8 = 0}{\text{Mode 0 Baud Rate}} = \frac{2^{SMOD1}}{4} \times \text{Oscillator Frequency}$$

The baud rate in Mode 2 also depends on the value of the SMOD1 bit. If SMOD1 = 0, 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:

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

16.2.1 Using Timer 1 to Generate Baud Rates

Setting TB8 = 1 in Mode 0 enables Timer 1 as the baud rate generator. When Timer 1 is the baud rate generator for Mode 0, the baud rates are determined by the Timer 1 overflow rate and the value of SMOD1 according to the following equation:

$$\underset{TB8 = 1}{\text{Mode 0 Baud Rate}} = \frac{2^{SMOD1}}{4} \times \text{(Timer 1 Overflow Rate)}$$

The Timer 1 overflow rate normally 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:

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

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:

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

While the TXE flag is set, the transmit buffer is empty. TXE can be cleared by software or by writing to SPDR. Writing to SPDR will clear TXE and load the transmit buffer. The user may load the buffer while the shift register is busy, i.e. before the current transfer completes. When the current transfer completes, the queued byte in the transmit buffer is moved to the shift register and the next transfer commences. TXE will generate an interrupt if the SPI interrupt is enabled and if the ENH bit in SPSR is set. For multi-byte transfers, TXE may be used to remove any dead time between byte transmissions.

The SPI master can operate in two modes: multi-master mode and single-master mode. By default, multi-master mode is active when SSIG = 0. In this mode, the $\overline{SS}$ input is used to disable a master device when another master is accessing the bus. When $\overline{SS}$ is driven low, the master device becomes a slave by clearing its MSTR bit and a Mode Fault is generated by setting the MODF bit in SPSR. MODF will generate an interrupt if enabled. The MSTR bit must be set in software before the device may become a master again. Single-master mode is enabled by setting SSIG = 1. In this mode $\overline{SS}$ is ignored and the master is always active. $\overline{SS}$ may be used as a general-purpose I/O in this mode.

17.2 Slave Operation

When the AT89LP428/828 is not configured for master operation, MSTR = 0, it will operate as an SPI slave. In slave mode, bytes are shifted in through MOSI and out through MISO by a master device controlling the serial clock on SCK. When a byte has been transferred, the SPIF flag is set to “1” and an interrupt request is generated, if enabled. The data received from the addressed master device is also transferred from the shift register to the receive buffer. The received data is accessed by reading SPDR. A slave device cannot initiate transfers. Data to be transferred to the master device must be preloaded by writing to SPDR. Writes to SPDR are double-buffered. The transmit buffer is loaded first and if the shift register is empty, the contents of the buffer will be transferred to the shift register.

While the TXE flag is set, the transmit buffer is empty. TXE can be cleared by software or by writing to SPDR. Writing to SPDR will clear TXE and load the transmit buffer. The user may load the buffer while the shift register is busy, i.e. before the current transfer completes. When the current transfer completes, the queued byte in the transmit buffer is moved to the shift register and waits for the master to initiate another transfer. TXE will generate an interrupt if the SPI interrupt is enabled and if the ENH bit in SPSR is set.

The SPI slave can operate in two modes: 4-wire mode and 3-wire mode. By default, 4-wire mode is active when SSIG = 0. In this mode, the $\overline{SS}$ input is used to enable/disable the slave device when addressed by a master. When $\overline{SS}$ is driven low, the slave device is enabled and will shift out data on MISO in response to the serial clock on SCK. While $\overline{SS}$ is high, the SPI slave will remain sleeping with MISO inactive. 3-wire mode is enabled by setting SSIG = 1. In this mode $\overline{SS}$ is ignored and the slave is always active. $\overline{SS}$ may be used as a general-purpose I/O in this mode.

The Disable Slave Output bit, DISSO in SPSR, may be used to disable the MISO line of a slave device. DISSO can allow several slave devices to share MISO while operating in 3-wire mode. In this case some protocol other than $\overline{SS}$ may be used to determine which slave is enabled.

17.5 SPI Registers

Table 17-2. 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

Writes to SPDR load the transmit buffer. In Master mode, a write also starts a transfer if the master is currently idle. In Slave mode, if data is not loaded to SPDR the SPI will echo the last byte received on the next transfer. Reads from SPDR return the value of the receive buffer, which is the last byte received. If SPDR is not read before completion of the next transfer, the old value will be lost.

Table 17-3. SPCR – SPI Control Register

SPCR Address = E9H

Not Bit Addressable

Reset Value = 0000 0000B

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

Symbol	Function
TSCK	SCK Clock Mode. When TSCK = 0, the SCK baud rate is based on the system clock, divided by the SPR_{1-0} ratio.When TSCK = 1, the SCK baud rate is based on the Timer 1 overflow rate, divided by the SPR_{1-0} ratio.
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 23-9 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 23-9 on SPI clock phase and polarity control.
SPR0 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: **SPR1 SPR0 SCK (TSCK = 0) SCK (TSCK = 1)** 0 0 $f_{OSC}/4$ $f_{T1OVF}/4$ 0 1 $f_{OSC}/8$ $f_{T1OVF}/8$ 1 0 $f_{OSC}/32$ $f_{T1OVF}/32$ 1 1 $f_{OSC}/64$ $f_{T1OVF}/64$

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 selecting the slave device ($\overline{SS}$ low).

Table 17-4. SPSR – SPI Status Register

SPSR Address = E8H — Reset Value = 0000 X000B

Not Bit Addressable

	SPIF	WCOL	MODF	TXE	–	SSIG	DISSO	ENH
Bit	7	6	5	4	3	2	1	0

Symbol	Function
SPIF	SPI Transfer Complete Interrupt Flag. When a serial transfer is complete, the SPIF bit is set by hardware and an interrupt is generated if ESP = 1. The SPIF bit may be cleared by software or by reading the SPI status register followed by reading/writing the SPI data register.
WCOL	Write Collision Flag. The WCOL bit is set by hardware if SPDR is written while the transmit buffer is full. The ongoing transfer is not affected. WCOL may be cleared by software or by reading the SPI status register followed by reading/writing the SPI data register.
MODF	Mode Fault Flag. MODF is set by hardware when a master mode collision is detected (MSTR = 1, SSIG = 0 and $\overline{SS}$ = 0) and an interrupt is generated if ESP = 1. MODF must be cleared by software.
TXE	Transmit Buffer Empty Flag. Set by hardware when the transmit buffer is loaded into the shift register, allowing a new byte to be loaded. TXE must be cleared by software. When ENH = 1 and ESP = 1, TXE will generate an interrupt.
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. When MSTR = 1 and SSIG = 0, $\overline{SS}$ is monitored for master mode collisions. Setting SSIG = 1 will ignore collisions on $\overline{SS}$. P1.4 may be used as a regular I/O pin when SSIG = 1.
DISSO	Disable slave output bit. When set, this bit causes the MISO pin to be tristated so that more than one slave device can share the same interface without multiple $\overline{SS}$ lines. Normally, the first byte in a transmission could be the slave address and only the selected slave should clear its DISSO bit.
ENH	TX Buffer Interrupt Enable. When ENH = 1, TXE will generate an SPI interrupt if ESP = 1. When ENH = 0, TXE does not generate an interrupt.

18. Dual Analog Comparators

The AT89LP428/828 provides two analog comparators. The analog comparators have the following features:

- Internal 3-level Voltage Reference
- Multiple Shared Analog Input Channels
- Selectable Interrupt Conditions
  - High- or Low-level
  - Rising- or Falling-edge
  - Output Toggle
- Hardware Debouncing Modes

A block diagram of the dual analog comparators with relevant connections is shown in Figure 18-1. Input options allow the comparators to function in a number of different configurations as shown in Figure 18-3. Comparator operation is such that the output is a logic “1” when the positive input is greater than the negative input. Otherwise, the output is a zero. Setting the CENA (ACSRA.3) and CENB (ACSRB.3) bits enable Comparator A and B, respectively. The user must also set the CONA (ACSRA.5) or CONB (ACSRB.5) bits to connect the comparator inputs before using a comparator. When a comparator is first enabled, the comparator output and

18.1 Analog Input Muxes

The positive input terminal of each comparator may be connected to any of the four analog input pins by changing the CSA_{1-0} or CSB_{1-0} bits in ACSRA and ACSRB. When changing the analog input pins, the comparator must be disconnected from its inputs by clearing the CONA or CONB bits. The connection is restored by setting the bits again after the muxes have been modified.

```
CLR  EC           ; Disable comparator interrupts
ANL  ACSRA, #0DFh ; Clear CONA to disconnect COMP A
...               ; Modify CSA or RFA bits
ORL  ACSRA, #020h ; Set CONA to connect COMP A
ANL  ACSRA, #0EFh ; Clear any spurious interrupt
SETB EC           ; Re-enable comparator interrupts
```

The corresponding comparator interrupt should not be enabled while the inputs are being changed, and the comparator interrupt flag must be cleared before the interrupt is re-enabled in order to prevent an unintentional interrupt request.

18.2 Internal Reference Voltage

The negative input terminal of each comparator may be connected to an internal voltage reference by changing the RFB_{1-0} or RFA_{1-0} bits in AREF. The internal reference voltage, V_{AREF}, is set to 1.25V ±5%. The voltage reference also provides two additional voltage levels approximately 125 mV above and below V_{AREF}. These levels may be used to configure the comparators as an internally referenced window comparator with up to four input channels. Changing the reference input must follow the same routine used for changing the positive input as described in the “Analog Input Muxes” section.

18.3 Comparator Interrupt Debouncing

The comparator output is normally 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 for edge-triggered interrupts. In debouncing mode, the comparator uses Timer 1 to modulate its sampling time when $Cx C_{1-0}$ = 00B. 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*x* 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 running free, 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 18-2. When the comparator clock is provided by one of the timer overflows, i.e. $Cx C_{1-0}$ = 00B, any change in the comparator output must be valid after 4 samples to be accepted as an edge event.

Figure 18-2. Negative Edge with Debouncing Example

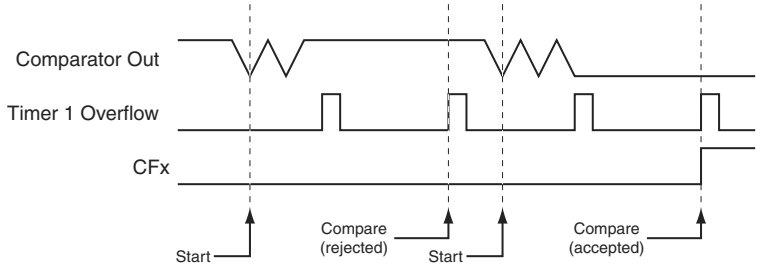


Table 20-1. Instruction Execution Times and Exceptions (Continued)

MUL AB	1	48	2	A4
DIV AB	1	48	4	84
DA A	1	12	1	D4
		Clock Cycles		
Logical	**Bytes**	**8051**	**AT89LP**	**Hex Code**
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
		Clock Cycles		
Data Transfer	**Bytes**	**8051**	**AT89LP**	**Hex Code**
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

22. On-chip Debug System

The AT89LP428/828 On-chip Debug (OCD) System uses a 2-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 change in program memory flow
- Break on stack overflow/underflow
- Break on Watchdog overflow
- Break on reset
- Non-intrusive operation
- Programming of nonvolatile memory

22.1 Physical Interface

The On-chip Debug System uses a 2-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. The OCD device connections are shown in Figure 22-1. When OCD is enabled, the $\overline{\text{RST}}$ port pin is configured as an input for the Debug Clock (DCL). Either the XTAL1, XTAL2 or P3.7 pin is configured as a bi-directional data line for the Debug Data (DDA) depending on the clock source selected. If the Internal RC Oscillator is selected, XTAL1 is configured as DDA (A). If the External Clock is selected, XTAL2 is configured as DDA (B). If the Crystal Oscillator is selected, P3.7 is configured as DDA (C). When OCD is enabled, the type of interface used depends on the OCD Interface Select User Fuse. This fuse selects between a normal Two-wire Interface (TWI) and a fast Two-wire Interface (FTWI). It is the duty of the user to program this fuse to the correct setting for their debug system at the same time they enable OCD (see “User Configuration Fuses” on page 121).

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

- P3.6/$\overline{\text{RST}}$ cannot be connected directly to V_{CC} and any external capacitors connected to $\overline{\text{RST}}$ must be removed.
- All external reset sources must be removed.
- If P3.7 needs to be debugged in-system using the crystal oscillator, the external clock option should be selected. The quartz crystal and any capacitors on XTAL1 or XTAL2 must be removed and an external clock signal must be driven on XTAL1. Some emulator systems may provide a user-configurable clock for this purpose.

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 23-3 on page 119.

Figure 23-3. Command Sequence Flow Chart

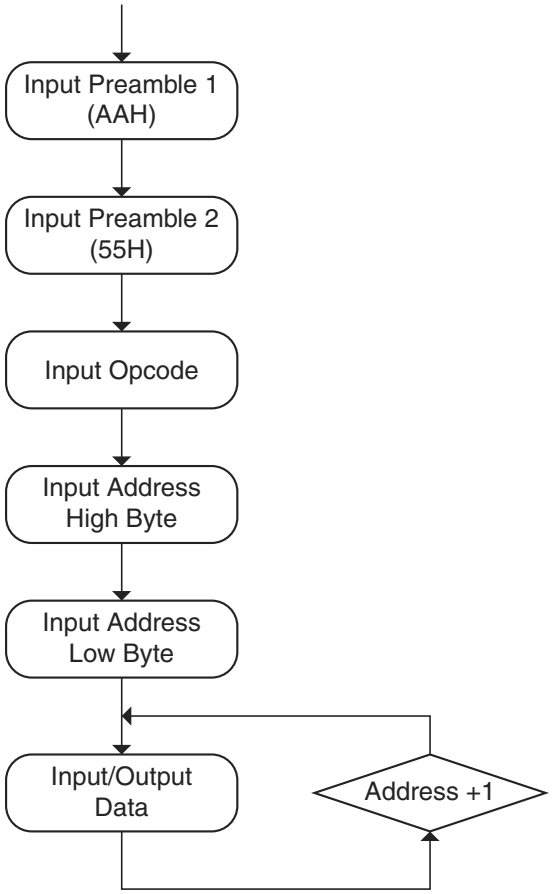
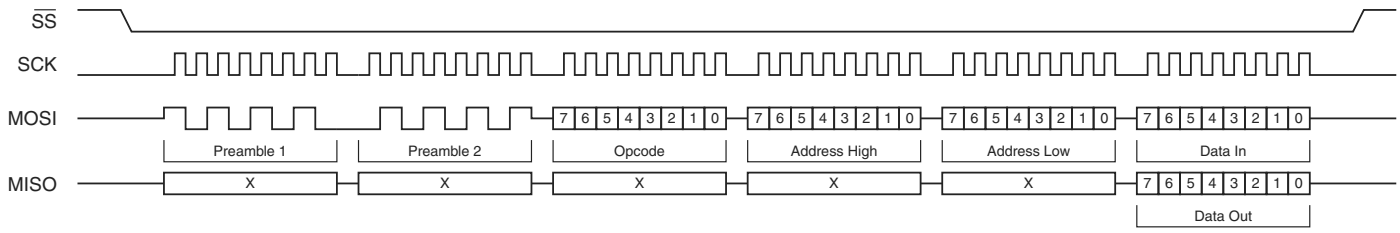


Figure 23-4. ISP Command Packet



23.4 Status Register

The current state of the memory may be accessed by reading the status register. The status register is shown in Table 23-4.

Table 23-4. Status Register

	–	–	–	–	$\overline{\text{LOAD}}$	SUCCESS	$\overline{\text{WRTINH}}$	$\overline{\text{BUSY}}$
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 (BOD).
WRTINH	Write Inhibit Flag. Cleared low by the 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.

23.5 $\overline{\text{DATA}}$ Polling

The AT89LP428/828 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.

23.6 Flash Security

The AT89LP428/828 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 23-5. 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 implements 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 23-5. Lock Bit Protection Modes

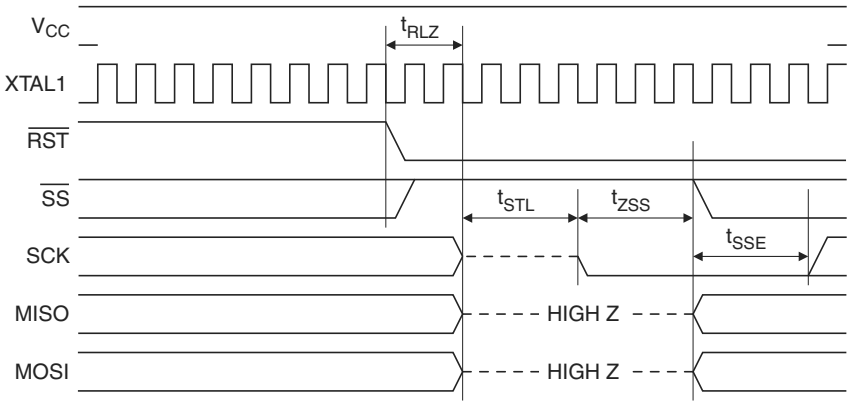
Program Lock Bits (by address)			
Mode	**00H**	**01H**	**Protection Mode**
1	FFH	FFH	No program lock features
2	00H	FFH	Further programming of the Flash is disabled
3	00H	00H	Further programming of the Flash is disabled and verify (read) is also disabled; OCD is disabled

23.9.3 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 23-7. In-System Programming (ISP) Start Sequence

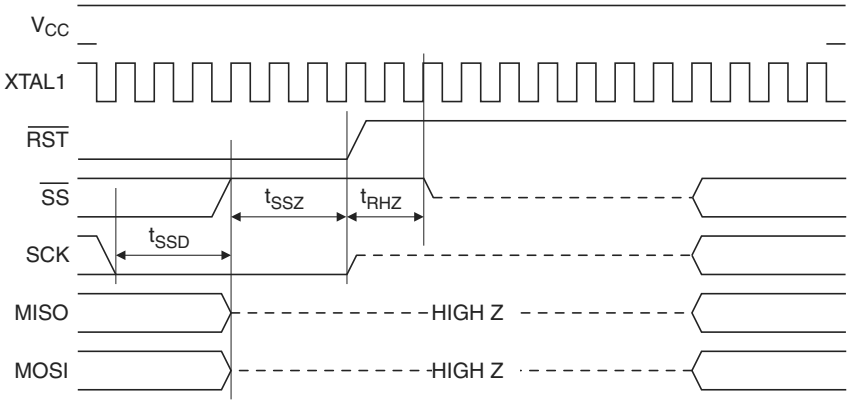


23.9.4 ISP Exit Sequence

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 23-8. In-System Programming (ISP) Exit Sequence



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

24.3.4 Quasi-Bidirectional Input

Figure 24-7. Quasi-bidirectional Input Transition Current at 5V

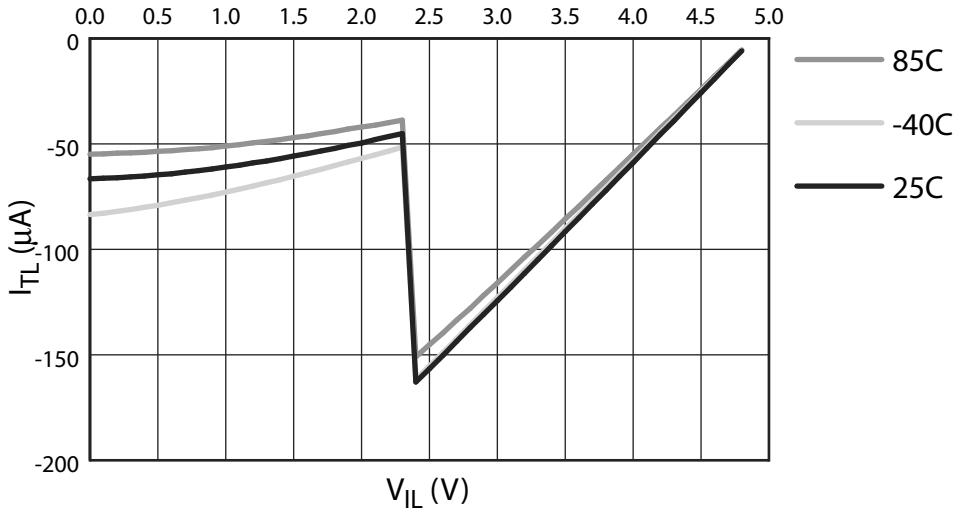
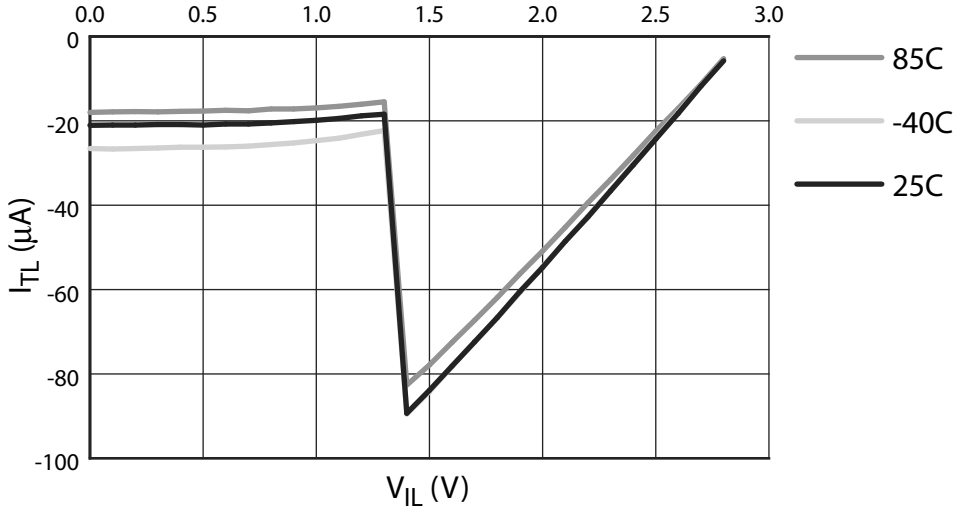


Figure 24-8. Quasi-bidirectional Input Transition Current at 3V



24.4 Clock Characteristics

The values shown in this table are valid for T_A = -40°C to 85°C and V_{CC} = 2.4 to 5.5V, unless otherwise noted.

Figure 24-9. External Clock Drive Waveform

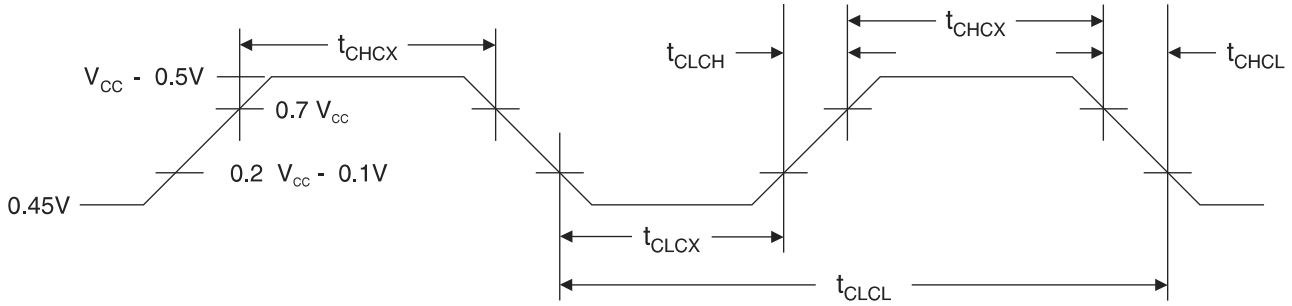


Table 24-1. External Clock Parameters

Symbol	Parameter	V_{CC} = 2.4V to 5.5V		V_{CC} = 4.0V to 5.5V		Units
		Min	Max	Min	Max	
$1/t_{CLCL}$	Oscillator Frequency	0	20	0	25	MHz
t_{CLCL}	Clock Period	50		40		ns
t_{CHCX}	External Clock High Time	12		12		ns
t_{CLCX}	External Clock Low Time	12		12		ns
t_{CLCH}	External Clock Rise Time		5		4	ns
t_{CHCL}	External Clock Fall Time		5		4	ns

Table 24-2. Clock Characteristics

Symbol	Parameter	Condition	Min	Max	Units
f_{XTAL}	Crystal Oscillator Frequency	Low Speed Oscillator	10	500	kHz
		High Speed Oscillator	0.5	25	MHz
f_{RC}	Internal Oscillator Frequency	T_A = 25°C; V_{CC} = 5.0V	7.92	8.08	MHz
		V_{DD} = 2.4 to 5.5V	7.60	8.40	MHz

24.5 Reset Characteristics

The values shown in this table are valid for T_A = -40°C to 85°C and V_{CC} = 2.4 to 5.5V, unless otherwise noted.

Table 24-3. Reset Characteristics

Symbol	Parameter	Condition	Min	Max	Units
$R_{\overline{RST}}$	Reset Pull-up Resistor		50	250	kΩ
V_{POR}	Power-On Reset Threshold		1.3	1.6	V
V_{BOD}	Brown-Out Detector Threshold		1.9	2.2	V
V_{BH}	Brown-Out Detector Hysteresis		200	300	mV
t_{POR}	Power-On Reset Delay		135	150	µs
t_{WDTRST}	Watchdog Reset Pulse Width		$16t_{CLCL}$		ns