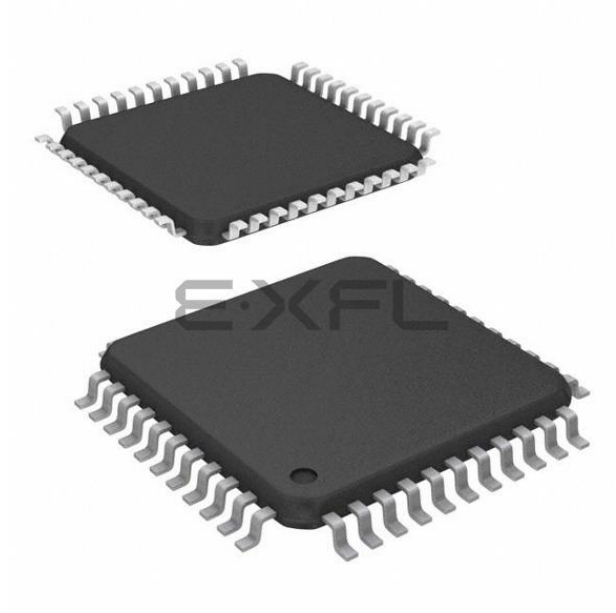
Microchip Technology - AT89LP6440-20AU Datasheet



Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	8051
Core Size	8-Bit
Speed	20MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	38
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	8K x 8
RAM Size	4.25K x 8
Voltage - Supply (Vcc/Vdd)	2.4V ~ 3.6V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-TQFP
Supplier Device Package	44-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at89lp6440-20au

There is no difference in counting rate between Timer 2's Auto-Reload/Capture and Baud Rate/Clock Out modes. All modes increment the timer once per clock cycle. Timer 2 in Auto-Reload/Capture mode increments at 12 times the rate of standard 8051s. Setting TPS_{3-0} = 1101B will force Timer 2 to count every twelve clocks. Timer 2 in Baud Rate or Clock Out mode increments at twice the rate of standard 8051s. Setting TPS_{3-0} = 0001B will force Timer 2 to count every two clocks.

2.3.6 Serial Port

The baud rate of the UART in Mode 0 defaults to 1/4 the clock frequency, compared to 1/12 the clock frequency in the standard 8051. In should also be noted that when using Timer 1 to generate the baud rate in UART Modes 1 or 3, the timer counts at the clock frequency and not at 1/12 the clock frequency. To maintain the same baud rate in the AT89LP3240/6440 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.

Timer 2 generated baud rates are twice as fast in the AT89LP3240/6440 than on standard 8051s when operating at the same frequency. The Timer Prescaler can also scale the baud rate to match an existing application.

2.3.7 SPI

The Serial Peripheral Interface (SPI) has a dedicated interrupt vector. The ESPI (IE2.2) bit replaces SPIE (SPCR.7). SPCR.7 (TSCK) now enables timer-generated baud rate.

The SPI includes Mode Fault detection. If multiple-master capabilities are not required, SSIG (SPSR.2) must be set to one for master mode to function correctly when $\overline{SS}$ (P1.4) is a general purpose I/O.

2.3.8 Watchdog Timer

The Watchdog Timer in AT89LP3240/6440 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.

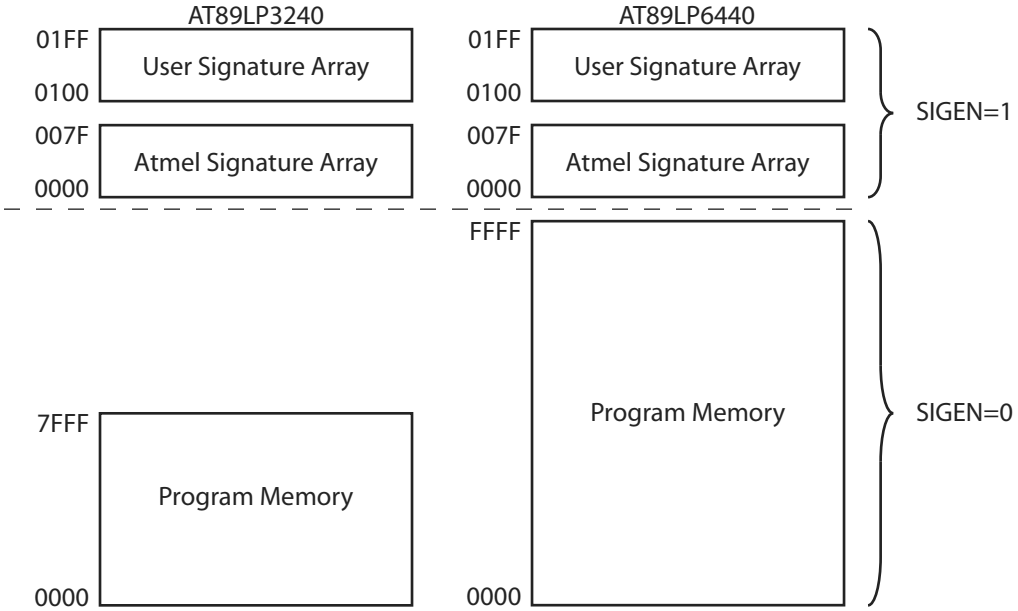
2.3.9 I/O Ports

The I/O ports of the AT89LP3240/6440 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, P2M0, P3M0 and P4M0 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, P2M0, P3M0 and P4M0 will reset to 00h instead of FFh and the ports will be weakly pulled high. Port 0 and the upper nibble of Port 2 always power up tristated regardless of the fuse setting due to their analog functions.

2.3.10 External Memory Interface

The AT89LP3240/6440 does not support external program memory. The $\overline{PSEN}$ and $\overline{EA}$ functions are not supported and those pins are replaced with general purpose I/O. The ALE strobe does not toggle continuously and cannot be used as a board-level clock.

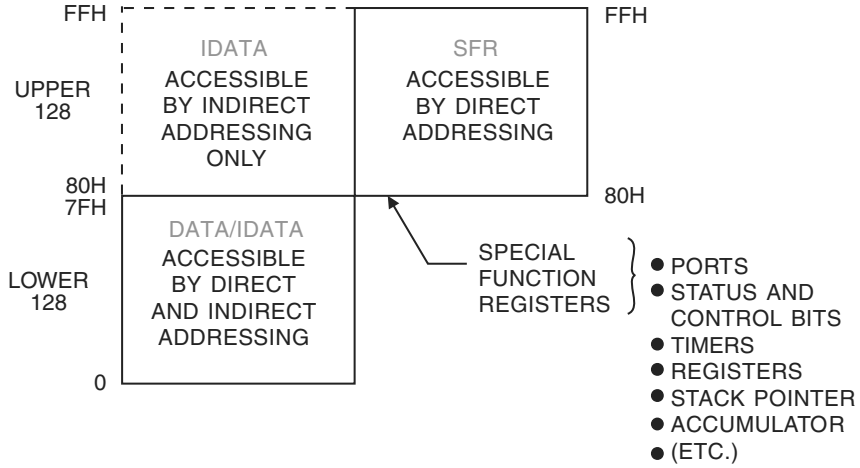
Figure 3-1. Program Memory Map



3.2 Internal Data Memory

The AT89LP3240/6440 contains 256 bytes of general SRAM data memory plus 128 bytes of I/O memory mapped into a single 8-bit address space. Access to the internal data memory does not require any configuration. The internal data memory has three address spaces: DATA, IDATA and SFR; as shown in Figure 3-2. Some portions of external data memory are also implemented internally. See “External Data Memory” below for more information.

Figure 3-2. Internal Data Memory Map



3.2.1 DATA

The first 128 bytes of RAM are directly addressable by an 8-bit address (00H–7FH) included in the instruction. 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 lower 128 bit addresses are also mapped into DATA addresses 20H—2FH.

Figure 3-5. FDATA Byte Write

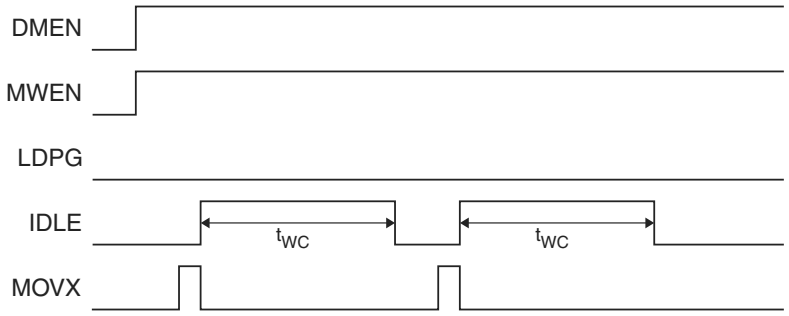
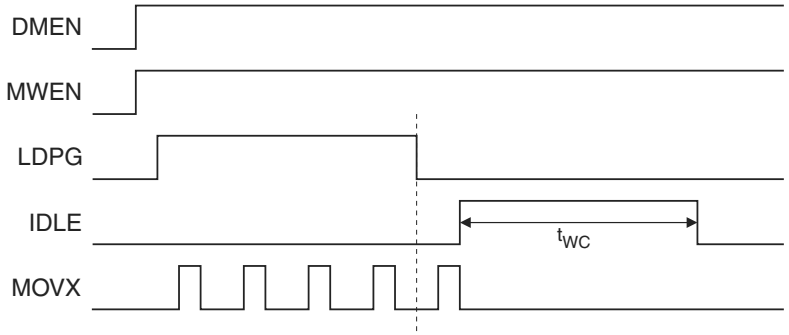


Figure 3-6. FDATA Page Write



Frequently just a few bytes within a page must be updated while maintaining the state of the other bytes. There are two options for handling this situation that allow the Flash Data memory to emulate a traditional EEPROM memory. The simplest method is to copy the entire page into a buffer allocated in RAM, modify the desired byte locations in the RAM buffer, and then load and write back first the low half page (with auto-erase) and then the high half page to the Flash memory. This option requires that at least one page size of RAM is available as a temporary buffer. The second option is to store only one half page in RAM. The unmodified bytes of the other page are loaded directly into the Flash memory's temporary load buffer before loading the updated values of the modified bytes. For example, if just the low half page needs modification, the user must first store the high half page to RAM, followed by reading and loading the unaffected bytes of the low half page into the page buffer. Then the modified bytes of the low half page are stored to the page buffer before starting the auto-erase sequence. The stored value of the high half page must be written without auto-erase after the programming of the low half page completes. This method reduces the amount of RAM required; however, more software overhead is needed because the read-and-load-back routine must skip those bytes in the page that need to be updated in order to prevent those locations in the buffer from being loaded with the previous data, as this will block the new data from being loaded correctly.

A write sequence will not occur if the Brown-out Detector is active, even if the BOD reset has been disabled. In cases where the BOD reset is disabled, the user should check the BOD status by reading the $\overline{\text{WRTINH}}$ bit in MEMCON. If a write currently in progress is interrupted by the BOD due to a low voltage condition, the ERR flag will be set. FDATA can always be read regardless of the BOD state.

For more details on using the Flash Data Memory, see the application note titled "AT89LP Flash Data Memory". FDATA may also be programmed by an external device programmer (See Section 25. on page 157).

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
ADC Interrupt	ADIF	0053H
Two-Wire Interface Interrupt	TWIF	005BH

9.1 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, IPH, IE2, IP2 or IP2H registers; the CPU is currently forced into idle by an IAP or FDATA write. Each 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, IPH, IE2, IP2 or IP2H, 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 8 cycles, since the longest instruction is 9 cycles long. If the instruction in progress is RETI with XSTK, the additional wait time cannot be more than 14 cycles (a maximum of 5 more cycles to complete the instruction in progress, plus a maximum of 9 cycles to complete the next instruction). Thus, in a single-inter-

rupt system, the response time is always more than 5 clock cycles and less than 21 clock cycles. See Figure 9-1 and Figure 9-2.

Figure 9-1. Minimum Interrupt Response Time

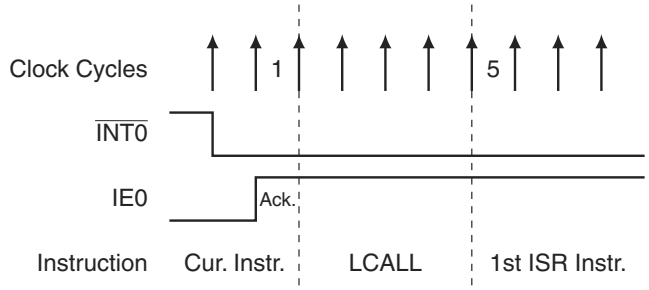


Figure 9-2. Maximum Interrupt Response Time

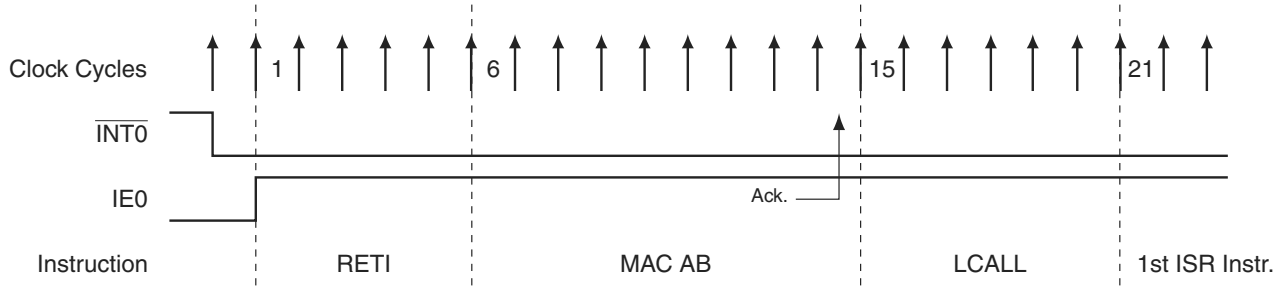


Table 9-2. IE – Interrupt Enable Register

IE = A8H — Reset Value = 0000 0000B

Bit Addressable

	EA	EC	ET2	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
ET2	Timer 2 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 9-3. IE2 – Interrupt Enable 2 Register

IE = B4H — Reset Value = xxxx x000B

Not Bit Addressable

	–	–	–	ETWI	EADC	ESPI	ECC	EGP
Bit	7	6	5	4	3	2	1	0

Symbol	Function
ETWI	Two-Wire Interface Interrupt Enable
EADC	ADC Interrupt Enable
ESPI	Serial Peripheral Interface Interrupt Enable
ECC	Compare/Capture Array Interrupt Enable
EGP	General-purpose Interrupt Enable

Table 9-4. IP – Interrupt Priority Register

IP = B8H — Reset Value = 0000 0000B

Bit Addressable

	IP0D	PC	PT2	PS	PT1	PX1	PT0	PX0
Bit	7	6	5	4	3	2	1	0

Symbol	Function
IP0D	Interrupt Priority 0 Disable. Set IP0D to 1 to disable all interrupts with priority level zero. Clear to 0 to enable all interrupts with priority level zero when EA = 1.
PC	Comparator Interrupt Priority Low
PT2	Timer 2 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 9-5. IP2 – Interrupt Priority 2 Register

IP = B5H — Reset Value = 0xxx x000B

No Bit Addressable

	IP2D	–	–	PTWI	PADC	PSP	PCC	PGP
Bit	7	6	5	4	3	2	1	0

Symbol	Function
IP2D	Interrupt Priority 2 Disable. Set IP2D to 1 to disable all interrupts with priority level two. Clear to 0 to enable all interrupts with priority level two when EA = 1.
PTWI	Two-wire Interface Interrupt Priority Low
PADC	ADC Interrupt Priority Low

12.1 Timer 2 Registers

Control and status bits for Timer 2 are contained in registers T2CON (see Table 12-3) and T2MOD (see Table 12-4). The register pair {TH2, TL2} at addresses 0CDH and 0CCH are the 16-bit timer register for Timer 2. The register pair {RCAP2H, RCAP2L} at addresses 0CBH and 0CAH are the 16-bit Capture/Reload register for Timer 2 in capture and auto-reload modes.

Table 12-3. T2CON – Timer/Counter 2 Control Register

T2CON Address = 0C8H — Reset Value = 0000 0000B

Bit Addressable

	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$
Bit	7	6	5	4	3	2	1	0

Symbol	Function
TF2	Timer 2 overflow flag set by a Timer 2 overflow and must be cleared by software. TF2 will not be set when either RCLK = 1 or TCLK = 1.
EXF2	Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2EX and EXEN2 = 1. When Timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software. EXF2 does not cause an interrupt in up/down counter mode (DCEN = 1) or dual-slope mode.
RCLK	Receive clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its receive clock in serial port Modes 1 and 3. RCLK = 0 causes Timer 1 overflows to be used for the receive clock.
TCLK	Transmit clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its transmit clock in serial port Modes 1 and 3. TCLK = 0 causes Timer 1 overflows to be used for the transmit clock.
EXEN2	Timer 2 external enable. When set, allows a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the serial port. EXEN2 = 0 causes Timer 2 to ignore events at T2EX.
TR2	Start/Stop control for Timer 2. TR2 = 1 starts the timer.
C/$\overline{T2}$	Timer or counter select for Timer 2. C/$\overline{T2}$ = 0 for timer function. C/$\overline{T2}$ = 1 for external event counter (falling edge triggered).
CP/$\overline{RL2}$	Capture/Reload select. CP/$\overline{RL2}$ = 1 causes captures to occur on negative transitions at T2EX if EXEN2 = 1. CP/$\overline{RL2}$ = 0 causes automatic reloads to occur when Timer 2 overflows or negative transitions occur at T2EX when EXEN2 = 1. When either RCLK or TCLK = 1, this bit is ignored and the timer is forced to auto-reload on Timer 2 overflow.

Table 12-4. T2MOD – Timer 2 Mode Control Register

T2MOD Address = 0C9H — Reset Value = 0000 0000B

Not Bit Addressable

	PHSD	PHS2	PHS1	PHS0	T2CM1	T2CM0	T2OE	DCEN
Bit	7	6	5	4	3	2	1	0

Symbol	Function
PHSD	CCA Phase Direction. For phase modes with 3 or 4 channels, PHSD determines the direction that the channels are cycled through. PHSD also determines the initial phase relationship for 2 phase modes. **PHSD** **Direction** 0 A →B →A →B or A →B →C →A →B →C or A →B →C →D →A →B →C →D 1 B →A →B →A or C →B →A →C →B →A or D →C →B →A →D →C →B →A

13.4.1 Asymmetrical PWM

For Asymmetrical PWM, Timer 2 should be configured for Auto-Reload mode and Count Mode 1 (CP/$\overline{RL2}$ = 0, DCEN = 0, T2CM$\overline{1\text{-}0}$ = 01B). Asymmetrical PWM uses single slope operation as shown in Figure 13-8. The timer counts up from BOTTOM to TOP and then restarts from BOTTOM. In non-inverting mode, the output CC*x* is set on the compare match between Timer 2 (TL2, TH2) and the channel data register (CC*x*L, CC*x*H), and cleared at BOTTOM. In inverting mode, the output CC*x* is cleared on the compare match between Timer 2 and the data register, and set at BOTTOM. The resulting asymmetrical output waveform is left-edge aligned.

The TOP value in RCAP2L and RCAP2H is double buffered such that the output frequency is only updated at the TOP to BOTTOM overflow. The channel data register (CC*x*L, CC*x*H) is also double-buffered such that the duty cycle is only updated at the TOP to BOTTOM overflow to prevent glitches. The output frequency and duty cycle for asymmetrical PWM are given by the following equations:

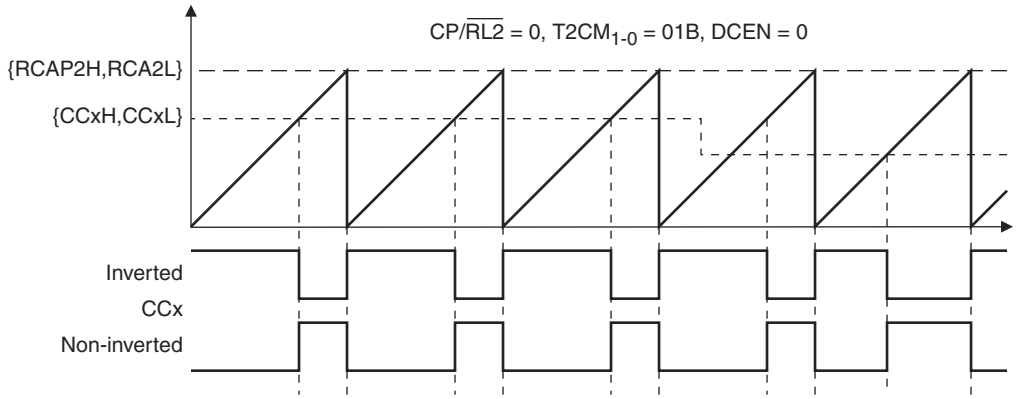
$$f_{OUT} = \frac{\text{Oscillator Frequency}}{\{\text{RCAP2H, RCAP2L}\} + 1} \times \frac{1}{\text{TPS} + 1}$$

$$\text{Inverting:} \qquad \text{Duty Cycle} = 100\% \times \frac{\{\text{CCxH, CCxL}\}}{\{\text{RCAP2H, RCAP2L}\} + 1}$$

$$\text{Non-Inverting:} \qquad \text{Duty Cycle} = 100\% \times \frac{\{\text{RCAP2H, RCAP2L}\} - \{\text{CCxH, CCxL}\} + 1}{\{\text{RCAP2H, RCAP2L}\} + 1}$$

The extreme compare values represent special cases when generating a PWM waveform. If the compare value is set equal to (or greater than) TOP, the output will remain low or high for non-inverting and inverting modes, respectively. If the compare value is set to BOTTOM (0000H), the output will remain high or low for non-inverting and inverting modes, respectively.

Figure 13-8. Asymmetrical (Edge-Aligned) PWM



13.4.2 Symmetrical PWM

For Symmetrical PWM, Timer 2 should be configured for Auto-Reload mode and Count Mode 2 or 3 (CP/$\overline{RL2}$ = 0, DCEN = 0, T2CM$\overline{1\text{-}0}$ = 1xB). Symmetrical PWM uses dual-slope operation as shown in Figure 13-9. The timer counts up from MIN to TOP and then counts down from TOP to MIN. The timer is equal to TOP for exactly one clock cycle. In non-inverting mode, the output CC*x* is cleared on the up-count compare match between Timer 2 (TL2, TH2) and the channel data register (CC*x*L, CC*x*H), and set at the down-count compare match. In inverting mode, the output CC*x* is set on the up-count compare match between Timer 2 and the data register, and cleared at the down-count compare match. The resulting symmetrical PWM output waveform is

16.3 More About Mode 0

In Mode 0, the UART is configured as a two wire half-duplex synchronous serial interface. Serial data enters and exits through RXD. TXD outputs the shift clock. Eight data bits are transmitted/received, with the LSB first. Figure 16-1 on page 90 shows a simplified functional diagram of the serial port in Mode 0 and associated timing. The baud rate is programmable to 1/2 or 1/4 the oscillator frequency by setting/clearing the SMOD1 bit. However, changing SMOD1 has an effect on the relationship between the clock and data as described below. The baud rate can also be generated by Timer 1 by setting TB8. Table 16-4 lists the baud rate options for Mode 0.

Table 16-4. Mode 0 Baud Rates

TB8	SMOD1	Baud Rate
0	0	$f_{SYS}/4$
0	1	$f_{SYS}/2$
1	0	(Timer 1 Overflow) / 4
1	1	(Timer 1 Overflow) / 2

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 bit slot may 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. 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. 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.

The relationship between the shift clock and data is determined by the combination of the SM2 and SMOD1 bits as listed in Table 16-5 and shown in Figure 16-2. The SM2 bit determines the idle state of the clock when not currently transmitting/receiving. The SMOD1 bit determines if the output data is stable for both edges of the clock, or just one.

Table 16-5. Mode 0 Clock and Data Modes

SM2	SMOD1	Clock Idle	Data Changed	Data Sampled
0	0	High	While clock is high	Positive edge of clock
0	1	High	Negative edge of clock	Positive edge of clock
1	0	Low	While clock is low	Negative edge of clock
1	1	Low	Negative edge of clock	Positive edge of clock

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.

17. Enhanced Serial Peripheral Interface

The Serial Peripheral Interface (SPI) allows high-speed synchronous data transfer between the AT89LP3240/6440 and peripheral devices or between multiple AT89LP3240/6440 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

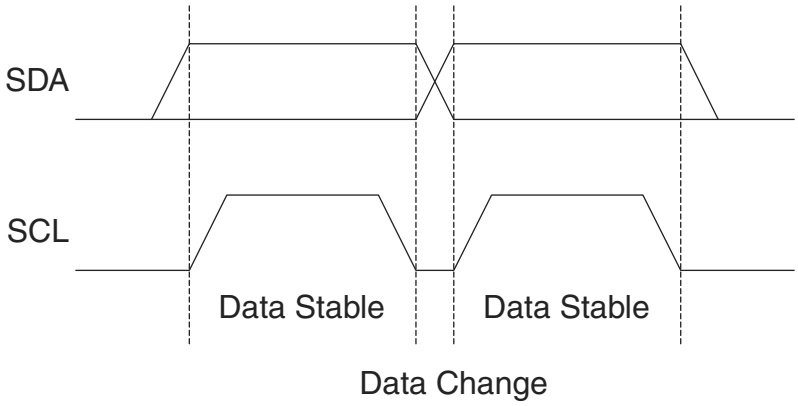
A block diagram of the SPI is shown below in Figure 17-1.

18.1 Data Transfer and Frame Format

18.1.1 Transferring Bits

Each data bit transferred on the TWI bus is accompanied by a pulse on the clock line. The level of the data line must be stable when the clock line is high. The only exception to this rule is for generating start and stop conditions.

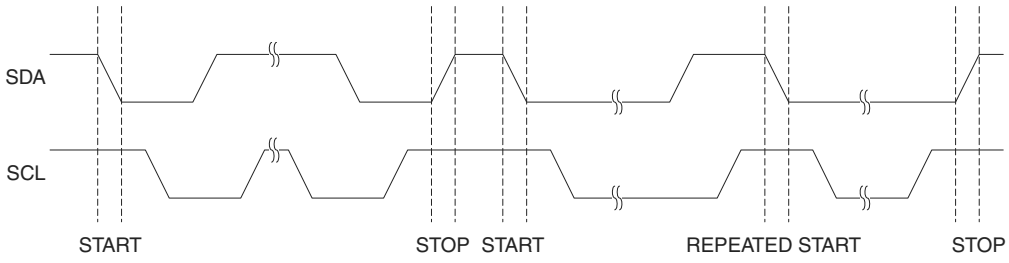
Figure 18-2. Data Validity



18.1.2 START and STOP Conditions

The Master initiates and terminates a data transmission. The transmission is initiated when the Master issues a START condition on the bus, and it is terminated when the Master issues a STOP condition. Between a START and a STOP condition, the bus is considered busy, and no other Master should try to seize control of the bus. A special case occurs when a new START condition is issued between a START and STOP condition. This is referred to as a REPEATED START condition, and is used when the Master wishes to initiate a new transfer without relinquishing control of the bus. After a REPEATED START, the bus is considered busy until the next STOP. This is identical to the START behavior, and therefore START is used to describe both START and REPEATED START for the remainder of this data sheet, unless otherwise noted. As depicted below, START and STOP conditions are signalled by changing the level of the SDA line when the SCL line is high.

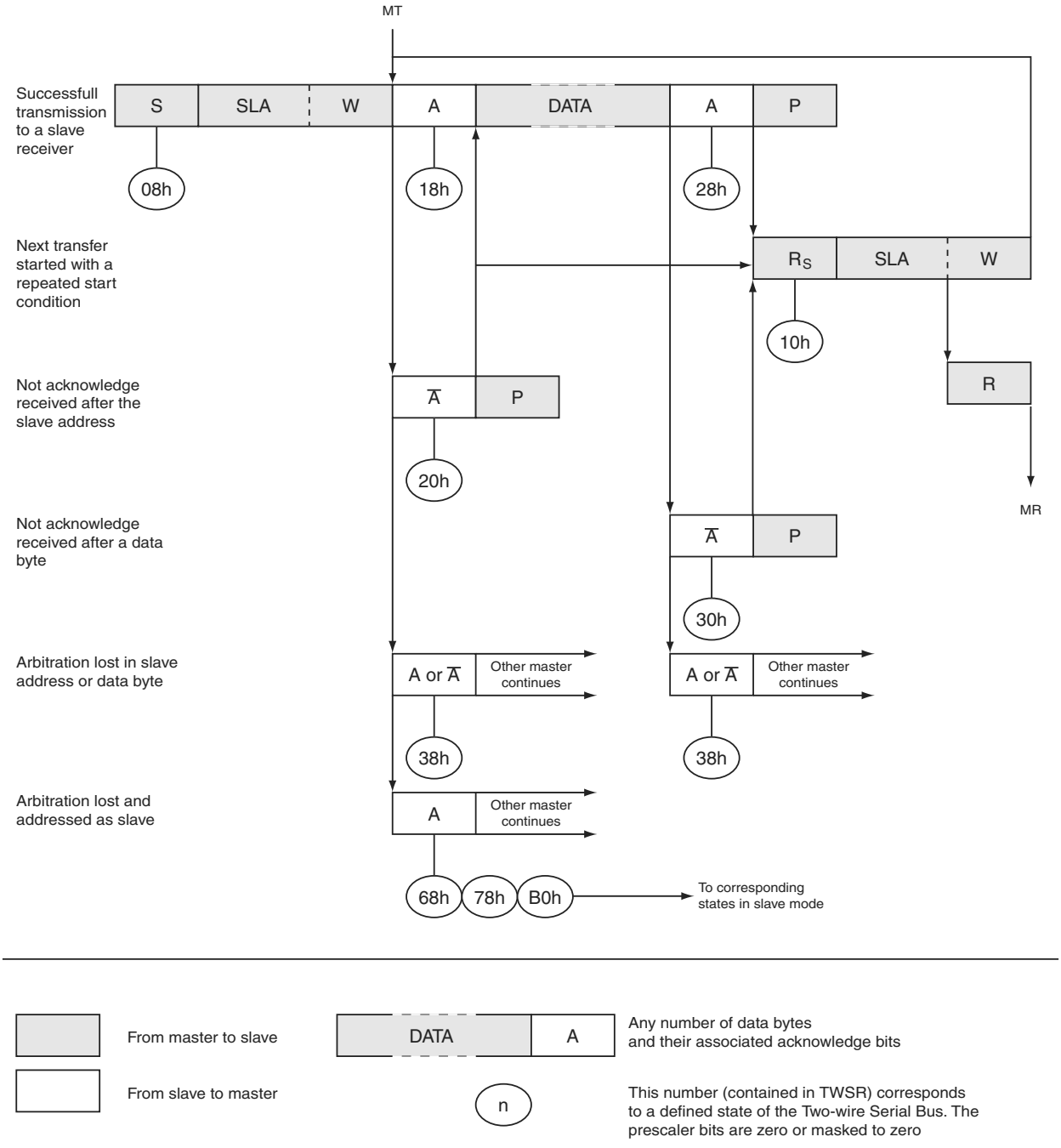
Figure 18-3. START, REPEATED START, and STOP Conditions



18.1.3 Address Packet Format

All address packets transmitted on the TWI bus are nine bits long, consisting of seven address bits, one READ/$\overline{\text{WRITE}}$ control bit and an acknowledge bit. If the READ/$\overline{\text{WRITE}}$ bit is set, a read operation is to be performed, otherwise a write operation should be performed. When a slave recognizes that it is being addressed, it should acknowledge by pulling SDA low in the ninth SCL (ACK) cycle. If the addressed Slave is busy, or for some other reason can not service the Mas-

Figure 18-11. Format and States in Master Transmitter Mode



18.6.2 Master Receiver Mode

In the Master Receiver mode, a number of data bytes are received from a slave transmitter. In order to enter a Master mode, a START condition must be transmitted. The format of the following address packet determines whether Master Transmitter or Master Receiver mode is to be entered. If SLA+W is transmitted, MT mode is entered, if SLA+R is transmitted, MR mode is entered.

18.6.5 Miscellaneous States

There are two status codes that do not correspond to a defined TWI state, see Table 18-10.

Status F8h indicates that no relevant information is available because the TWIF flag is not set. This occurs between other states, and when the TWI is not involved in a serial transfer.

Status 00h indicates that a bus error has occurred during a Two-wire Serial Bus transfer. A bus error occurs when a START or STOP condition occurs at an illegal position in the format frame. Examples of such illegal positions are during the serial transfer of an address byte, a data byte, or an acknowledge bit. When a bus error occurs, TWIF is set. To recover from a bus error, the STO flag must set and TWIF must be cleared. This causes the TWI to enter the not addressed Slave mode and to clear the STO flag (no other bits in TWCR are affected). The SDA and SCL lines are released, and no STOP condition is transmitted.

Table 18-10. Miscellaneous States

Status Code (TWSR)	Status of the Two-wire Serial Bus and Two-wire Serial Interface hardware	Application Software Response					Next Action Taken by TWI Hardware
		To/from TWDR	To TWCR				
			STA	STO	TWIF	AA	
F8h	No relevant state information available; TWIF = “0”	No action	No action				Wait or proceed current transfer
00h	Bus error due to an illegal START or STOP condition	No action	0	1	1	X	Only the internal hardware is affected, no STOP condition is sent on the bus. In all cases, the bus is released and STO is cleared.

18.6.6 Combining Several TWI Modes

In some cases, several TWI modes must be combined in order to complete the desired action. Consider for example reading data from a serial EEPROM. Typically, such a transfer involves the following steps:

1. The transfer must be initiated.
2. The EEPROM must be instructed what location should be read.
3. The reading must be performed.
4. The transfer must be finished.

Note that data is transmitted both from Master to Slave and vice versa. The Master must instruct the Slave what location it wants to read, requiring the use of the MT mode. Subsequently, data must be read from the Slave, implying the use of the MR mode. Thus, the transfer direction must be changed. The Master must keep control of the bus during all these steps, and the steps should be carried out as an atomic operation. If this principle is violated in a multi-master system, another Master can alter the data pointer in the EEPROM between steps 2 and 3, and the Master will read the wrong data location. Such a change in transfer direction is accomplished by transmitting a REPEATED START between the transmission of the address byte and reception of the data. After a REPEATED START, the Master keeps ownership of the bus. The following figure shows the flow in this transfer.

22.1 Instruction Set Extensions

The following instructions are extensions to the standard 8051 instruction set that provide enhanced capabilities not found in standard 8051 devices. All extended instructions start with an A5H escape code. For this reason random A5H reserved codes should not be placed in the instruction stream even though other devices may have treated these as NOPs.

Other AT89LP devices may not support all of these instructions.

22.1.1 ASR M

Function: Shift MAC Accumulator Right Arithmetically

Description: The forty bits in the M register are shifted one bit to the right. Bit 39 retains its value to preserve the sign of the value. No flags are affected.

Example: The M register holds the value 0C5B1A29384H . The following instruction,

```
ASR   M
```

leaves the M register holding the value 0E2D8D149C2H.

Bytes: 2

Cycles: 2

Encoding:

A5	0	0	0	0	0	0	1	1

Operation: ASR
$(M_n) \leftarrow (M_{n+1})$ n = 0 - 38
$(M_{39}) \leftarrow (M_{39})$

22.1.2 BREAK

Function: Software Breakpoint (Halt execution)

Description: BREAK transfers control from normal execution to the On-Chip Debug (OCD) handler if OCD is enabled. The PC is left pointing to the following instruction. If OCD is disabled, BREAK acts as a double NOP. No flags are affected.

Example: If On-Chip Debugging is allowed, the following instruction,

```
BREAK
```

will halt instruction execution prior to the immediately following instruction. If debugging is not allowed, the BREAK is treated as a double NOP.

Bytes: 2

Cycles: 2

Encoding:

A5	0	0	0	0	0	0	0	0

Operation: BREAK
(PC) ←(PC) + 2

22.1.7 LSL M

Function: Shift MAC Accumulator Left Logically

Description: The forty bits in the M register are shifted one bit to the left. Bit 0 is cleared. No flags are affected.

Example: The M register holds the value 0C5B1A29384H. The following instruction,

```
LSL   M
```

leaves the M register holding the value 8B63452708H.

Bytes: 2

Cycles: 2

Encoding:

A5	0	0	1	0	0	0	1	1

Operation: LSL
$(M_{n+1}) \leftarrow (M_n)$ n = 0 - 38
$(M_0) \leftarrow 0$

22.1.8 MOVC A, @A+/DPTR

Function: Move code byte relative to Alternate Data Pointer

Description: The MOVC instructions load the Accumulator with a code byte or constant from program memory. The address of the byte fetched is the sum of the original unsigned 8-bit Accumulator contents and the contents of the unselected Data Pointer. The base register is not altered. Sixteen-bit addition is performed so a carry-out from the low-order eight bits may propagate through higher-order bits. No flags are affected.

Example: A value between 0 and 3 is in the Accumulator. The following instructions will translate the value in the Accumulator to one of four values defined by the DB (define byte) directive.

```
    MOV     /DPTR, #TABLE
    MOVC    A, @A+PC
    RET

TABLE:
    DB      66H
    DB      77H
    DB      88H
    DB      99H
```

If the subroutine is called with the Accumulator equal to 01H, it returns with 77H in the Accumulator.

Bytes: 2

Cycles: 4

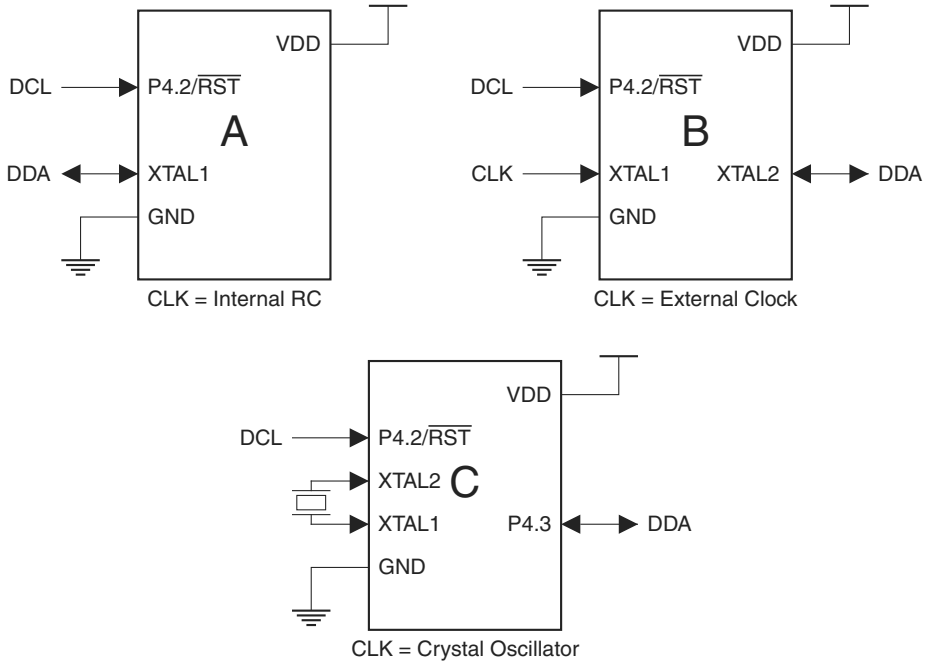
Encoding:

A5	1	0	0	1	0	0	1	1

Operation: MOVC
IF (DPS) = 0
THEN
    (A) ←((A) + (DPTR1))
ELSE
    (A) ←((A) + (DPTR0))

- P4.2/$\overline{\text{RST}}$ cannot be connected directly to V_{DD} and any external capacitors connected to $\overline{\text{RST}}$ must be removed.
- All external reset sources must be removed.
- If P4.3 needs to be debugged in systems 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.

Figure 24-1. AT89LP3240/6440 On-Chip Debug Connections



24.2 Software Breakpoints

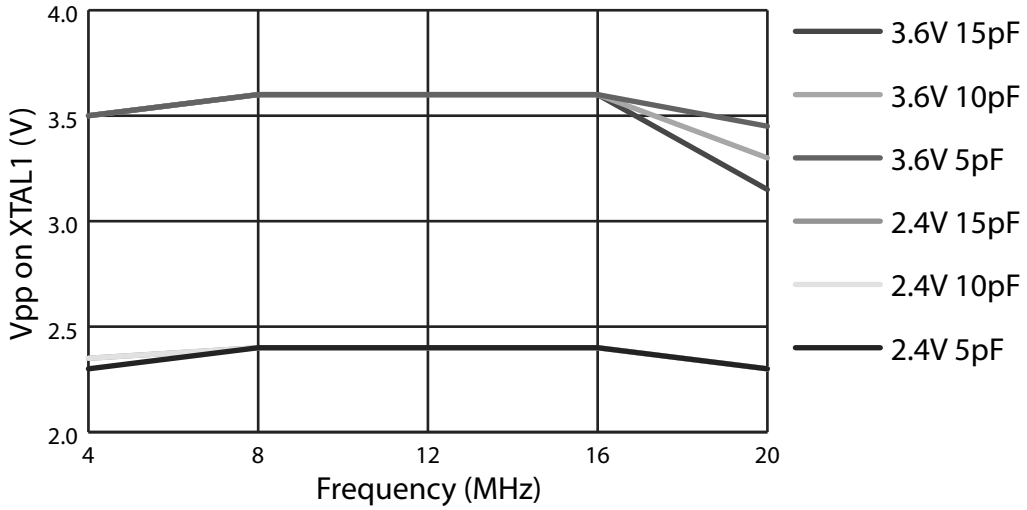
The AT89LP3240/6440 microcontroller includes a BREAK instruction for implementing program memory breakpoints in software. A software breakpoint can be inserted manually by placing the BREAK instruction in the program code. Some emulator systems may allow for automatic insertion/deletion of software breakpoints. The Flash memory must be re-programmed each time a software breakpoint is changed. Frequent insertions/deletions of software breakpoints will reduce the endurance 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.

24.3 Limitations of On-Chip Debug

The AT89LP3240/6440 is a 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 P4.2 and the External Reset ($\overline{\text{RST}}$). Therefore, neither P4.2 nor an external reset source may be emulated when OCD is enabled.

Figure 26-13. Typical Crystal Oscillator Swing with Quartz Crystal and C1=C2, T_A = 25°C



Note: 1. Replacing capacitor C1 with a resistor R1 of 4 MΩ results in similar swing levels on XTAL1.

Figure 26-14. Typical Crystal Oscillator Swing with Ceramic Resonator and C1=C2, T_A = 25°C

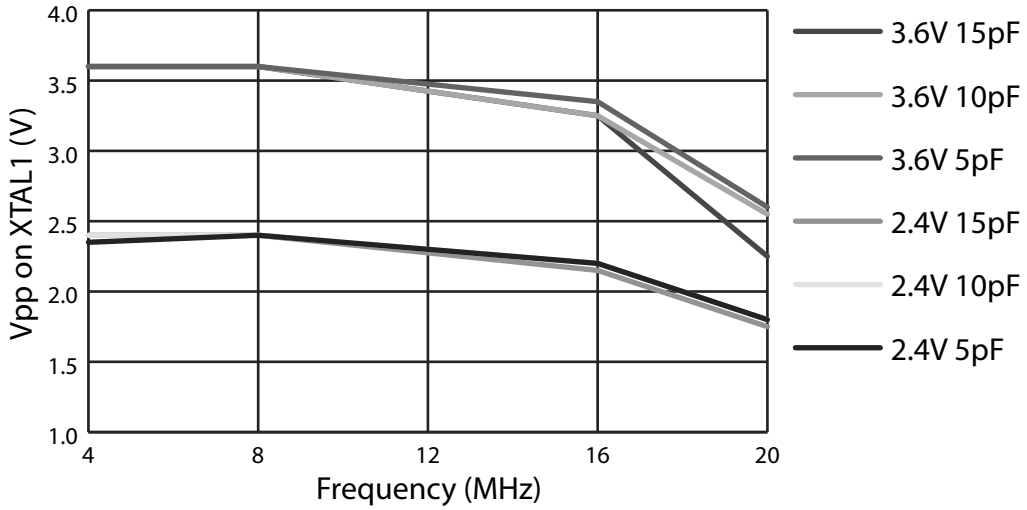


Table of Contents

Table of Contents (Continued)

Table of Contents (Continued)