**Welcome to E-XFL.COM**

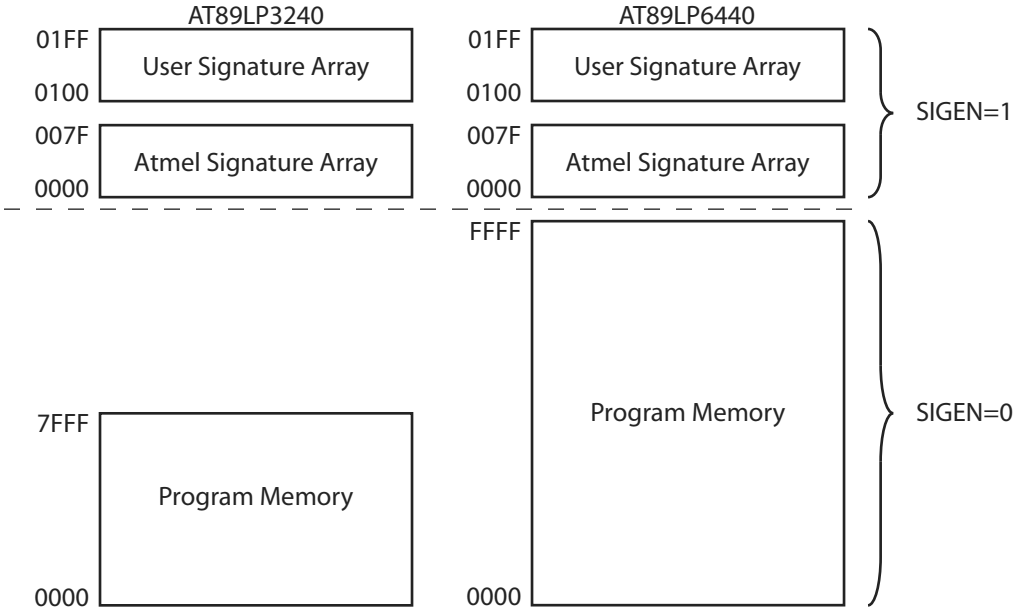
## What is "Embedded - Microcontrollers"?

"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 | 32KB (32K 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-VFQFN Exposed Pad |
| Supplier Device Package | 44-VQFN (7x7) |
| Purchase URL | https://www.e-xfl.com/product-detail/atmel/at89lp3240-20mu |

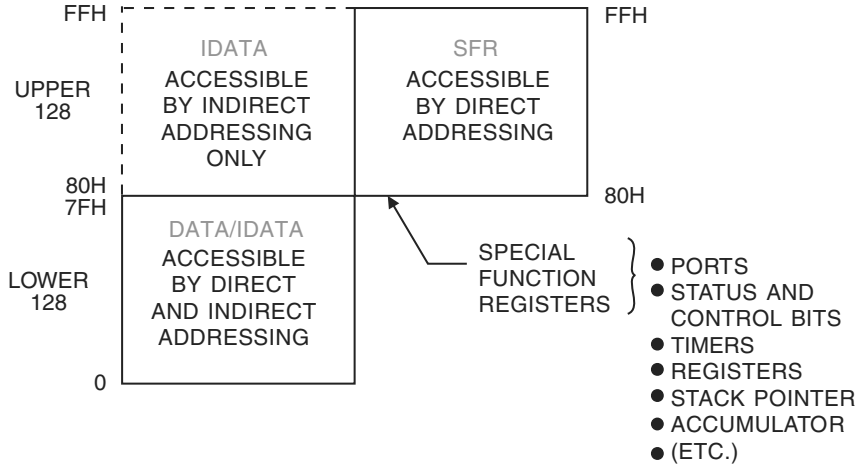
**Figure 3-1.** Program Memory Map




## 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.

**Table 3-3.** MEMCON – Memory Control Register

MEMCON = 96H — Reset Value = 0000 00XXB

Not Bit Addressable

| | IAP | AERS | LDPG | MWEN | DMEN | ERR | – | $\overline{\text{WRTINH}}$ |
|---|---|---|---|---|---|---|---|---|
| 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. |
| ERR | Error Flag. Set by hardware if an error occurred during the last programming sequence due to a brownout condition (low voltage on VDD). Must be cleared by software. |
| $\overline{\text{WRTINH}}$ | Write Inhibit Flag. Cleared by hardware when the voltage on VDD has fallen below the minimum programming voltage. Set by hardware when the voltage on VDD is above the minimum programming voltage. |

### 3.3.4 External Memory Interface

The AT89LP3240/6440 uses the standard 8051 external memory interface with the upper address on Port 2, the lower address and data in/out multiplexed on Port 0, and the ALE, $\overline{\text{RD}}$ and $\overline{\text{WR}}$ strobes. The interface may be used in two different configurations depending on which type of MOVX instruction is used to access XDATA.

Figure 3-7 shows a hardware configuration for accessing up to 64K bytes of external RAM using a 16-bit linear address. Port 0 serves as a multiplexed address/data bus to the RAM. The Address Latch Enable strobe (ALE) is used to latch the lower address byte into an external register so that Port 0 can be freed for data input/output. Port 2 provides the upper address byte throughout the operation. The MOVX @DPTR instructions use Linear Address mode

**Figure 3-7.** External Memory 16-bit Linear Address Mode

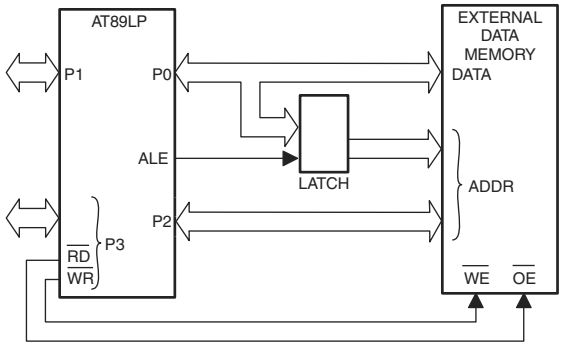


**Table 5-1.** DSPR – Digital Signal Processing Configuration Register

DSPR = E2H — Reset Value = 0000 0000B

Not Bit Addressable

| | MRW1 | MRW0 | SMLB | SMLA | CBE1 | CBE0 | MVCD | DPRB |
|---|---|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

| Symbol | Function |
|---|---|
| $MRW_{1-0}$ | M Register Window. Selects which pair of bytes from the 5-byte M register is accessible through MACH (E5H) and MACL (E4H) as shown in Figure 5-5. For example, MRW = 10B for normal 16-bit fixed-point operations where the lowest order portion of the fractional result is discarded. |
| SMLB | Signed Multiply Operand B. When SMLB = 0, the MUL AB instruction treats the contents of B as an unsigned value. When SMLB = 1, the MUL AB instruction interprets the contents of B as a signed two's complement value. SMLB does not affect the MAC operation. |
| SMLA | Signed Multiply Operand A. When SMLA = 0, the MUL AB instruction treats the contents of ACC as an unsigned value. When SMLA = 1, the MUL AB instruction interprets the contents of ACC as a signed two's complement value. SMLA does not affect the MAC operation. |
| CBE1 | DPTR1 Circular Buffer Enable. Set CBE1 = 1 to configure DPTR1 for circular addressing over the two circular buffer address ranges. Clear CBE1 for normal DPTR operation. |
| CBE0 | DPTR0 Circular Buffer Enable. Set CBE0 = 1 to configure DPTR0 for circular addressing over the two circular buffer address ranges. Clear CBE0 for normal DPTR operation. |
| MVCD | MOVC Index Disable. When MVCD = 0, the MOVC A, @A+DPTR instruction functions normally with indexed addressing. Setting MVCD = 1 disables the indexed addressing mode such that MOVC A, @A+DPTR functions as MOVC A, @DPTR. |
| DPRB | DPTR1 Redirect to B. DPRB selects the source/destination register for MOVC/MOVX instructions that reference DPTR1. When DPRB = 0, ACC is the source/destination. When DPRB = 1, B is the source/destination. DPRB does not change the index register for MOVC instructions. |

- In some cases, both data pointers must be used simultaneously. To prevent frequent toggling of DPS, the AT89LP3240/6440 supports a prefix notation for selecting the opposite data pointer per instruction. All DPTR instructions, with the exception of JMP @A+DPTR, when prefixed with an 0A5H opcode will use the inverse value of DPS ($\overline{DPS}$) to select the data pointer. Some assemblers may support this operation by using the /DPTR operand. For example, the following code performs a block copy within EDATA:

```
        MOV  DPCF, #00H      ; DPS = 0
        MOV  DPTR, #SRC      ; load source address to dptr0
        MOV  /DPTR, #DST     ; load destination address to dptr1
        MOV  R7, #BLKSIZE    ; number of bytes to copy
COPY:   MOVX A, @DPTR        ; read source (dptr0)
        INC  DPTR            ; next src (dptr0+1)
        MOVX @/DPTR, A       ; write destination (dptr1)
        INC  /DPTR           ; next dst (dptr1+1)
        DJNZ R7, COPY
```

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
```

#### 5.2.2.2 Index Disable

The MOVC Index Disable bit, MVCD (DSPR.1), disables the indexed addressing mode of the MOVC A, @A+DPTR instruction. When MVCD = 1, the MOVC instruction functions as MOVC A, @DPTR with no indexing as shown in Table 5-7. MVCD can improve the efficiency of routines that must fetch multiple operands from program memory. DPRB can change the MOVC destination register from ACC to B, but has no effect on the MOVC index register.

**Table 5-7.** MOVC @DPTR Operating Modes

| MVCD | DPRB | Equivalent Operation for MOVC A, @A+DPTR | | | |
|---|---|---|---|---|---|
| | | DPS = 0 | | DPS = 1 | |
| | | DPTR | /DPTR | DPTR | /DPTR |
| 0 | 0 | MOVC A, @A+DPTR0 | MOVC A, @A+DPTR1 | MOVC A, @A+DPTR1 | MOVC A, @A+DPTR0 |
| 0 | 1 | MOVC A, @A+DPTR0 | MOVC B, @A+DPTR1 | MOVC B, @A+DPTR1 | MOVC A, @A+DPTR0 |
| 1 | 0 | MOVC A, @DPTR0 | MOVC A, @DPTR1 | MOVC A, @DPTR1 | MOVC A, @DPTR0 |
| 1 | 1 | MOVC A, @DPTR0 | MOVC B, @DPTR1 | MOVC B, @DPTR1 | MOVC A, @DPTR0 |

#### 5.2.2.3 Circular Buffers

The CBE0 and CBE1 bits in DSPR can configure DPTR0 and DPTR1, respectively, to operate in circular buffer mode. The AT89LP3240/6440 maps circular buffers into two identically sized regions of EDATA/XDATA. These buffers can speed up convolution computations such as FIR and IAR digital filters. The length of the buffers are set by the value of the FIRD (E3H) register for up to 256 entries. Buffer A is mapped from 0000H to FIRD and Buffer B is mapped from 0100H to 100H+FIRD as shown in Figure 5-6. Both data pointers may operate in either buffer. When circular buffer mode is enabled, updates to a data pointer referencing the buffer region will follow circular addressing rules. If the data pointer is equal to FIRD or 100H+FIRD any increment will cause it to overflow to 0000H or 0100H respectively. If the data pointer is equal to 0000H or 0100H any decrement will cause it to underflow to FIRD or 100H+FIRD respectively. In this mode, updates can be either an explicit INC DPTR or an automatic update using DPU*n* where the DPD*n* bits control the direction. The data pointer will increment or decrement normally at any other addresses. Therefore, when circular addressing is in use, the data pointers can still operate as regular pointers in the FIRD+1 to 00FFH and greater than 100H+FIRD ranges.

**Figure 5-6.** Circular Buffer Mode

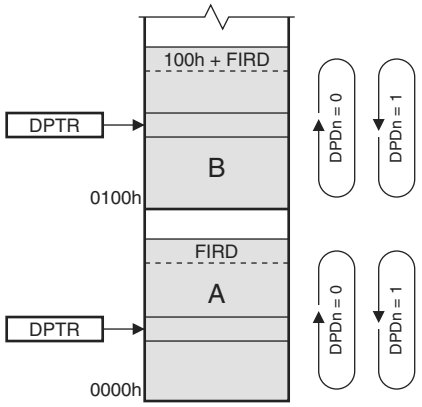
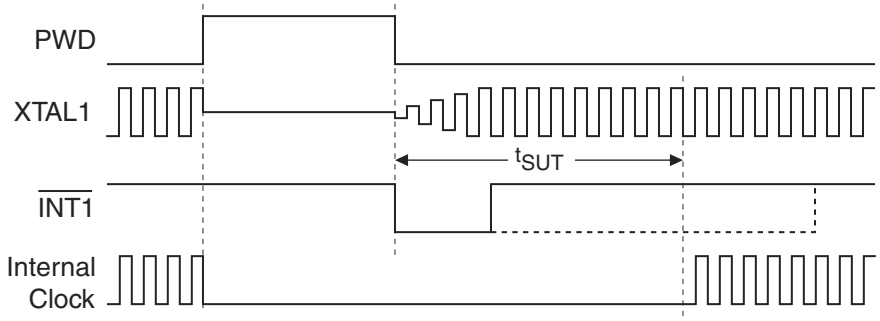
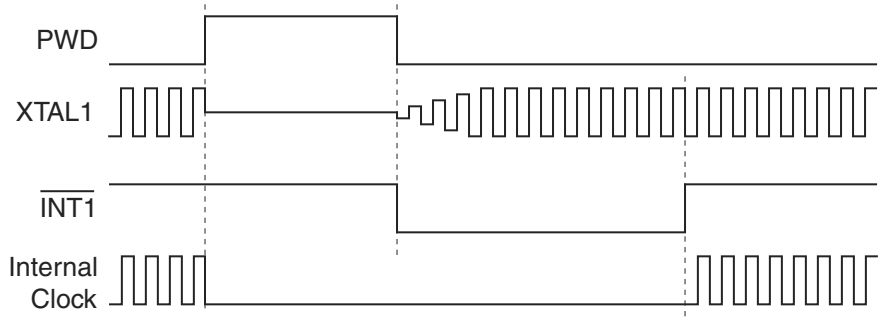


**Figure 8-1.** Interrupt Recovery from Power-down (PWDEX = 0)




When PWDEX = “1”, the wake-up period is controlled externally by the interrupt. Again, at the falling edge on the interrupt pin, power-down is exited and the oscillator is restarted. However, the internal clock will not propagate until the rising edge of the interrupt pin as shown in Figure 8-2. The interrupt pin should be held low long enough for the selected clock source to stabilize. After the rising edge on the pin the interrupt service routine will be executed.

**Figure 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 35). 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.

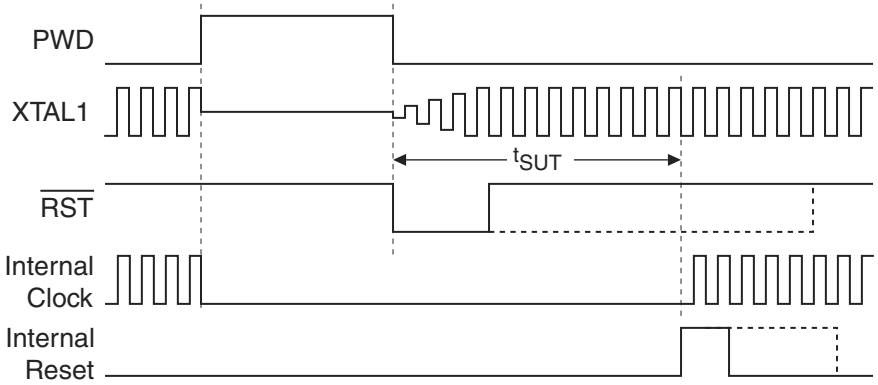
## 8.3 Reducing Power Consumption

Several possibilities need consideration when trying to reduce the power consumption in an AT89LP-based system. Generally, Idle or Power-down mode should be used as much as possible. All unneeded functions should be disabled. In particular, the following modules may need special consideration when trying to achieve the lowest possible power consumption.

### 8.3.1 Brown-out Detector

If the Brown-out Detector is not needed by the application, this module should be turned off. If the Brown-out Detector is enabled by the BOD Enable Fuse, it will be enabled in all modes except Power-down. See Section 25.7 “User Configuration Fuses” on page 164.

**Figure 8-3.** Reset Recovery from Power-down




### 8.3.2 Analog Comparators

The comparators will operate during Idle mode if enabled. To save power, the comparators should be disabled before entering Idle mode if possible. When the comparators are turned off and on again, some settling time is required for the analog circuits to stabilize. If the comparators are enabled, they will consume the least power when using an external reference, $RFA_{1-0}$ = 00B and $RFB_{1-0}$ = 00B.

### 8.3.3 Analog-to-Digital Converter

The DADC will operate during Idle mode if enabled. To save power, the DADC should be disabled before entering Idle mode if possible. When the DADC is turned off and on again, some settling time is required for the analog circuits to stabilize. If the DADC is enabled, it will consume the least power when configured to use the system clock instead of the internal RC oscillator (unless the IRC is the system clock source) and when the internal reference is disabled (IREF = 0). The DADC must always be disabled before entering power-down.

## 9. Interrupts

The AT89LP3240/6440 provides 12 interrupt sources: two external interrupts, three timer interrupts, a serial port interrupt, an analog comparator interrupt, a general-purpose interrupt, a compare/capture interrupt, a two-wire interrupt, an ADC interrupt and an SPI interrupt. These interrupts and the system reset each have a separate program vector at the start of the program memory space. Each interrupt source can be individually enabled or disabled by setting or clearing a bit in the interrupt enable registers IE and IE2. The IE register also contains a global disable bit, EA, which disables all interrupts.

Each interrupt source can be individually programmed to one of four priority levels by setting or clearing bits in the interrupt priority registers IP, IPH, IP2 and IP2H. IP and IP2 hold the low order priority bits and IPH and IP2H hold the high priority bits for each interrupt. An interrupt service routine in progress can be interrupted by a higher priority interrupt, but not by another interrupt of the same or lower priority. The highest priority interrupt cannot be interrupted by any other interrupt source. If two requests of different priority levels are pending at the end of an instruction, the request of higher priority level is serviced. If requests of the same priority level are pending at the end of an instruction, an internal polling sequence determines which request is serviced. The polling sequence is based on the vector address; an interrupt with a lower vector address has higher priority than an interrupt with a higher vector address. Note that the polling sequence is only used to resolve pending requests of the same priority level.

**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-

**Table 11-3.** TMOD – Timer/Counter Mode Control Register

TMOD Address = 089H — Reset Value = 0000 0000B

Not Bit Addressable

| | GATE1 | C/$\overline{T1}$ | T1M1 | T1M0 | GATE0 | C/$\overline{T0}$ | T0M0 | T0M1 |
|---|---|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

| Symbol | Function |
|---|---|
| GATE1 | Timer 1 Gating Control. When set, Timer/Counter 1 is enabled only while $\overline{INT1}$ pin is high and TR1 control pin is set. When cleared, Timer 1 is enabled whenever TR1 control bit is set. |
| C/$\overline{T1}$ | Timer or Counter Selector 1. Cleared for Timer operation (input from internal system clock). Set for Counter operation (input from T1 input pin). C/$\overline{T1}$ must be zero when using Timer 1 in PWM mode. |
| T1M1<br>T1M0 | Timer 1 Operating Mode<br>**Mode** / **T1M1** / **T1M0** / **Operation**<br>0 / 0 / 0 / Variable 9–16-bit Timer Mode. 8-bit Timer/Counter TH1 with TL1 as 1–8-bit prescaler.<br>1 / 0 / 1 / 16-bit Auto-Reload Mode. TH1 and TL1 are cascaded to form a 16-bit Timer/Counter that is reloaded with RH1 and RL1 each time it overflows.<br>2 / 1 / 0 / 8-bit Auto Reload Mode. TH1 holds a value which is reloaded into 8-bit Timer/Counter TL1 each time it overflows.<br>3 / 1 / 1 / Timer/Counter 1 is stopped |
| GATE0 | Timer 0 Gating Control. When set, Timer/Counter 0 is enabled only while $\overline{INT0}$ pin is high and TR0 control pin is set. When cleared, Timer 0 is enabled whenever TR0 control bit is set. |
| C/$\overline{T0}$ | Timer or Counter Selector 0. Cleared for Timer operation (input from internal system clock). Set for Counter operation (input from T0 input pin). C/$\overline{T0}$ must be zero when using Timer 0 in PWM mode. |
| T0M1<br>T0M0 | Timer 0 Operating Mode<br>**Mode** / **T0M1** / **T0M0** / **Operation**<br>0 / 0 / 0 / Variable 9–16-bit Timer Mode. 8-bit Timer/Counter TH0 with TL0 as 1–8-bit prescaler.<br>1 / 0 / 1 / 16-bit Auto-Reload Mode. TH0 and TL0 are cascaded to form a 16-bit Timer/Counter that is reloaded with RH0 and RL0 each time it overflows.<br>2 / 1 / 0 / 8-bit Auto Reload Mode. TH0 holds a value which is reloaded into 8-bit Timer/Counter TL0 each time it overflows.<br>3 / 1 / 1 / Split Timer Mode. TL0 is an 8-bit Timer/Counter controlled by the standard Timer 0 control bits. TH0 is an 8-bit timer only controlled by Timer 1 control bits. |

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

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

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)

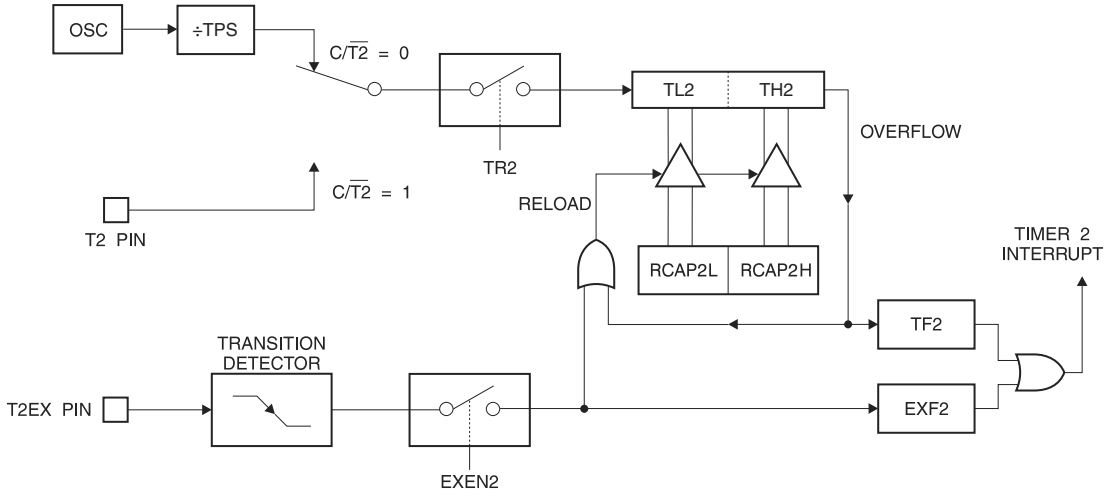
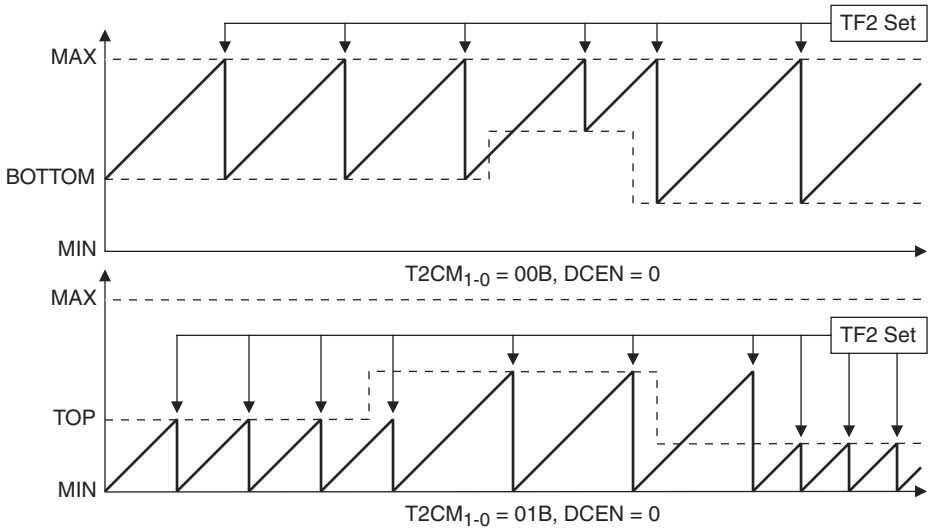



**Figure 12-3.** Timer 2 Waveform: Auto-Reload Mode (DCEN = 0)




### 12.3.2 Up or Down Counter

Setting DCEN = 1 enables Timer 2 to count up or down, as shown in Figure 12-4. In this mode, the T2EX pin controls the direction of the count (if EXEN2 = 1). A logic 1 at T2EX makes Timer 2 count up. When $T2CM_{1-0}$ = 00B, the timer will overflow at MAX and set the TF2 bit. This overflow also causes BOTTOM, the 16-bit value in RCAP2H and RCAP2L, to be reloaded into the timer

**Table 13-4.** T2CCF – Timer/Counter 2 Compare/Capture Flags

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

Not Bit Addressable

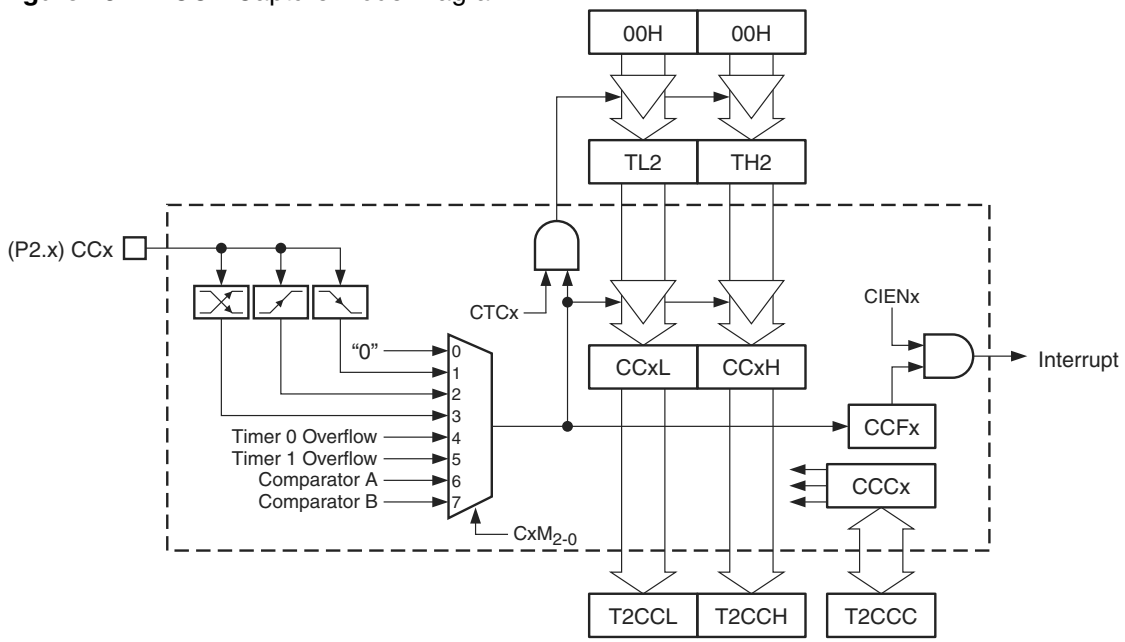
| | – | – | – | – | CCFD | CCFC | CCFB | CCFA |
|---|---|---|---|---|---|---|---|---|
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

| 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. 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




Each CCA channel has an associated external capture input pin: CCA (P2.0), CCB (P2.1), CCC (P2.2) and CCD (P2.3). External capture events are always edge-triggered and can be selected

**Table 13-6.** Summary of Multi-Phasic Modes

| $PHS_{2-0}$ | Mode | Behavior | |
|---|---|---|---|
| | | PHSD = 0 | PHSD = 1 |
| 000 | Off | Normal Operation (all channels active at all times) | |
| 001 | 1:2 | A →B →A →B | B →A →B →A |
| 010 | 1:3 | A →B →C →A →B →C | C →B →A →C →B →A |
| 011 | 1:4 | A →B →C →D →A →B →C →D | D →C →B →A →D →C →B →A |
| 100 | 2:4 | A →B →A →B<br>C →D →C →D | B →A →B →A<br>D →C →D →C |

**Figure 13-12.** Multi-Phasic PWM Output Stage

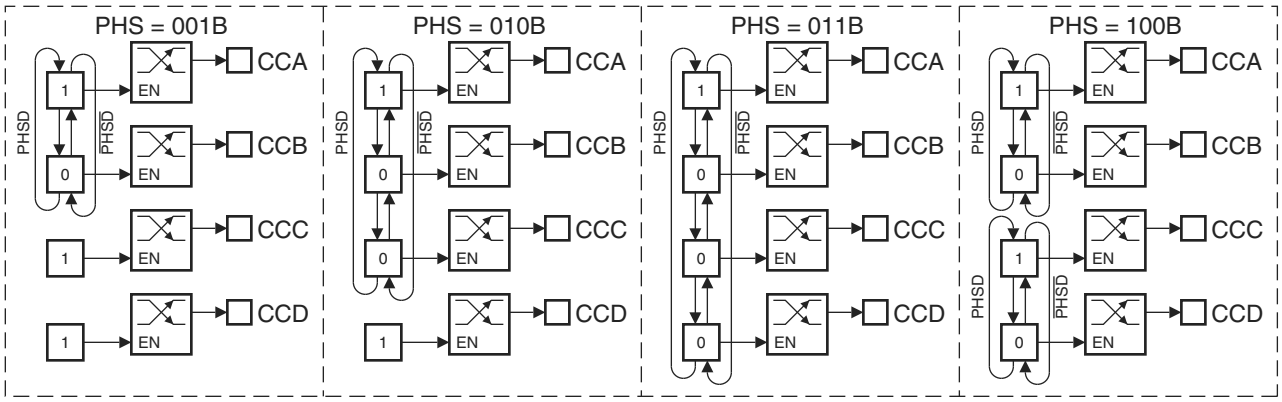



**Figure 13-13.** Three-Phase Mode with Channel B Disabled

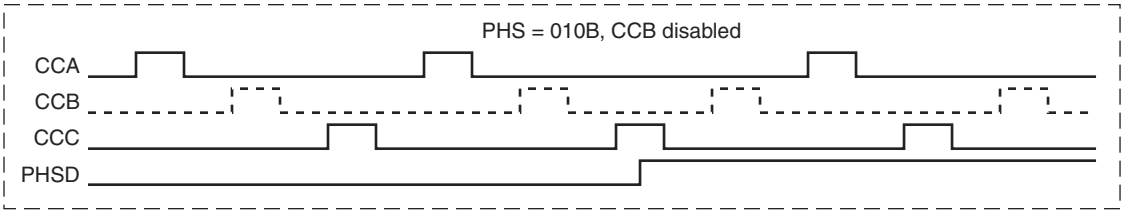
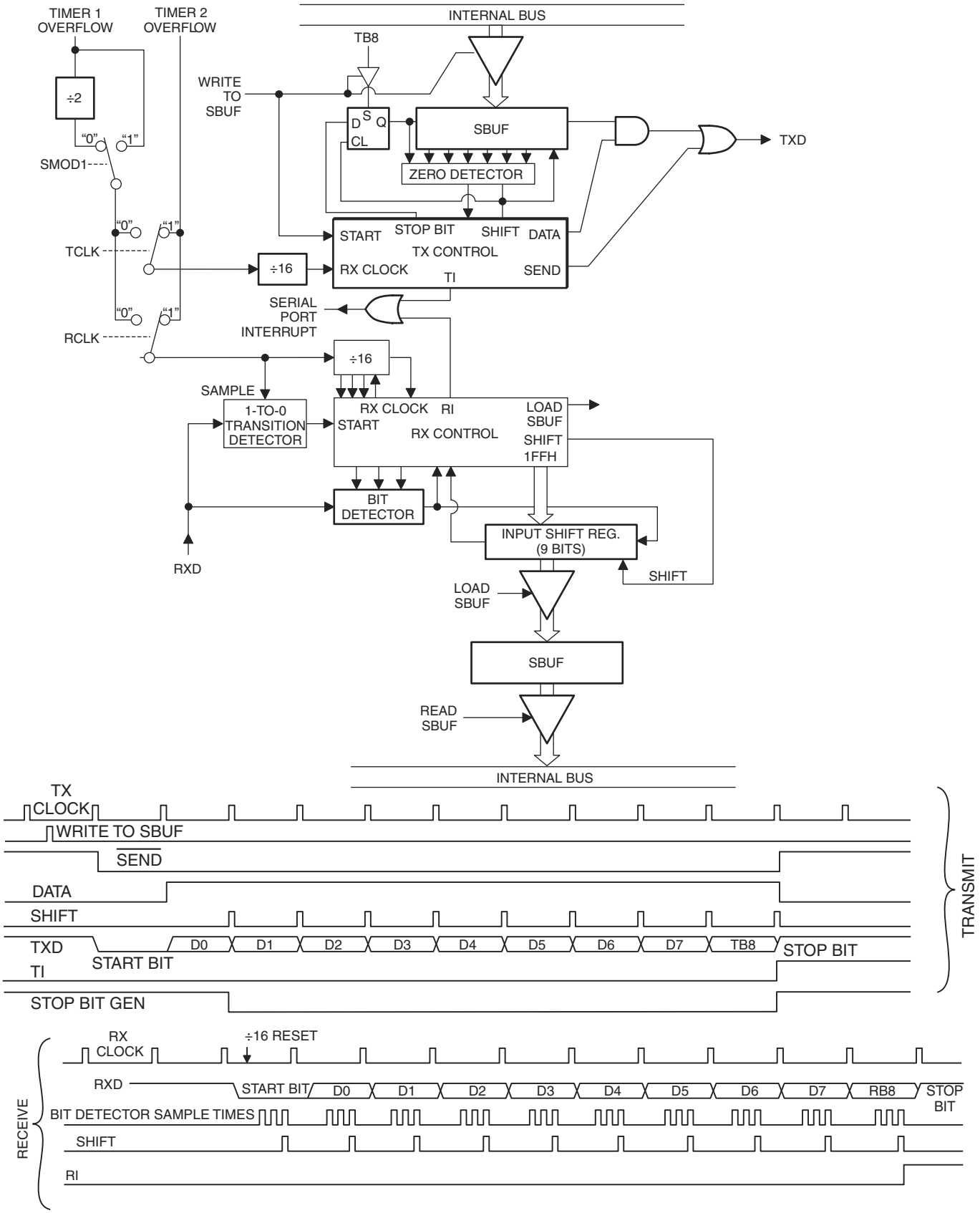


**Figure 16-6.** Serial Port Mode 3

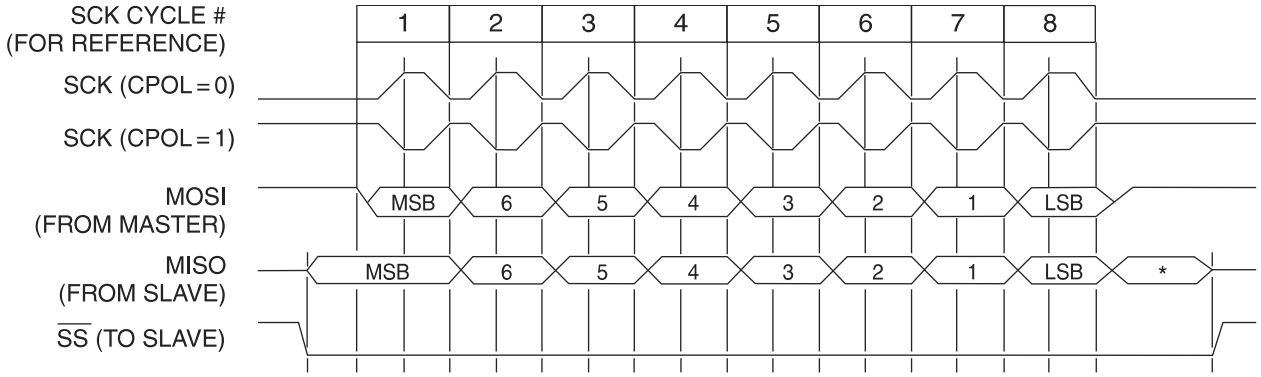


| | |
|---|---|
| SSIG | Slave Select Ignore. If SSIG = 0, the SPI will only operate in slave mode if $\overline{SS}$ (P1.4) is pulled low. When SSIG = 1, the SPI ignores $\overline{SS}$ in slave mode and is active whenever SPE (SPCR.6) is set. When MSTR = 1 and SSIG = 0, $\overline{SS}$ is monitored 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. |

## 17.4 Serial Clock Timing

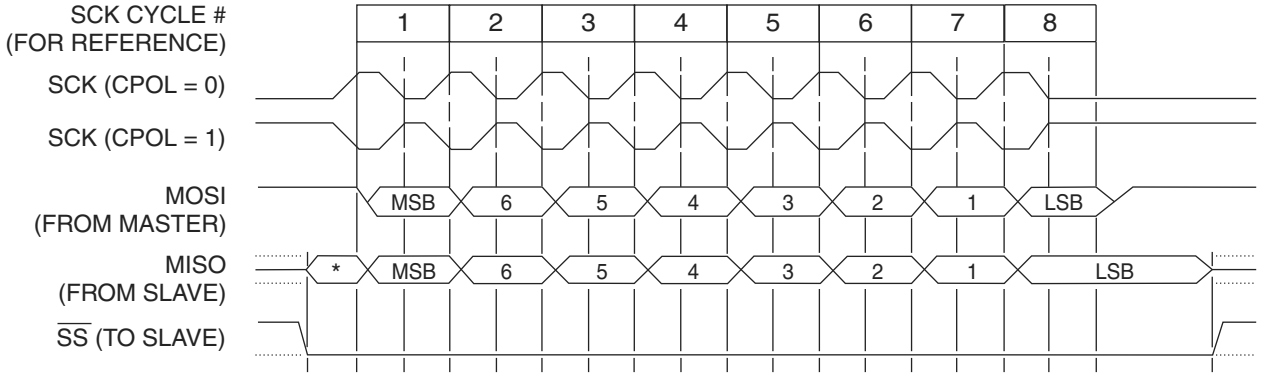
The CPHA, CPOL and SPR bits in SPCR control the shape and rate of SCK. The two SPR bits provide four possible clock rates when the SPI is in master mode. In slave mode, the SPI will operate at the rate of the incoming SCK as long as it does not exceed the maximum bit rate. There are also four possible combinations of SCK phase and polarity with respect to the serial data. CPHA and CPOL determine which format is used for transmission. The SPI data transfer formats are shown in Figures 17-3 and 17-4. To prevent glitches on SCK from disrupting the interface, CPHA, CPOL, and SPR should not be modified while the interface is enabled, and the master device should be enabled before the slave device(s).

**Figure 17-3.** SPI Transfer Format with CPHA = 0




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

**Figure 17-4.** SPI Transfer Format with CPHA = 1




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

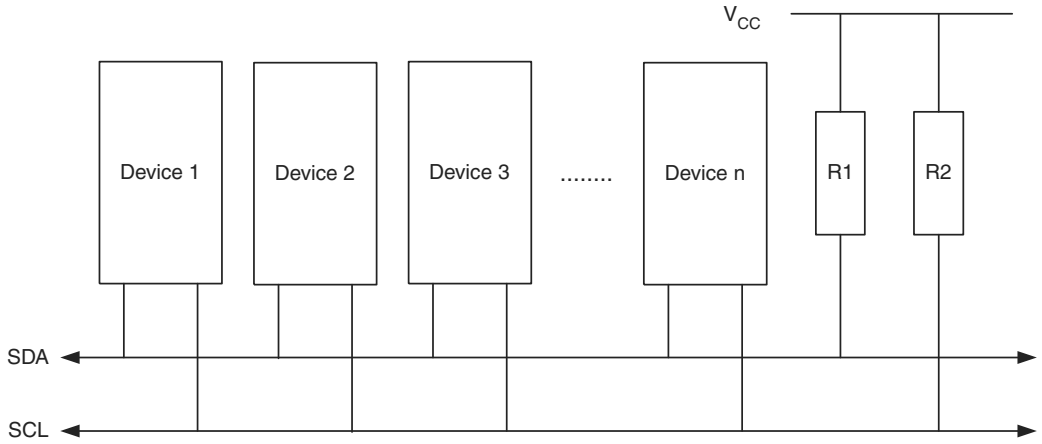
## 18. Two-Wire Serial Interface

The Two-Wire Interface (TWI) is a bi-directional 2-wire serial communication standard. It is designed primarily for simple but efficient integrated circuit (IC) control. The system is comprised of two lines, SCL (Serial Clock) and SDA (Serial Data) that carry information between the ICs connected to them. The only external hardware needed to implement the bus is a single pull-up resistor for each of the TWI bus lines. All devices connected to the bus have individual addresses, and mechanisms for resolving bus contention are inherent in the TWI protocol. The serial data transfer is limited to 400Kbit/s in standard mode. Various communication configurations can be designed using this bus. Figure 18-1 shows a typical 2-wire bus configuration. Any of the devices connected to the bus can be master or slave.

The Two-Wire Interface on the AT89LP provides the following features:

- Simple Yet Powerful and Flexible Communication Interface, only two Bus Lines Needed
- Both Master and Slave Operation Supported
- Device can Operate as Transmitter or Receiver
- 7-bit Address Space Allows up to 128 Different Slave Addresses
- Multi-master Arbitration Support
- Up to 400 kHz Data Transfer Speed
- Fully Programmable Slave Address with General Call Support

**Figure 18-1.** Two-Wire Bus Configuration




As depicted in Figure 18-1, both bus lines are connected to the positive supply voltage through pull-up resistors. The bus drivers of all TWI-compliant devices are open-drain or open-collector. This implements a wired-AND function which is essential to the operation of the interface. A low level on a TWI bus line is generated when one or more TWI devices output a zero. A high level is output when all TWI devices tristate their outputs, allowing the pull-up resistors to pull the line high. Note that all AT89LP devices connected to the TWI bus must be powered in order to allow any bus operation. The number of devices that can be connected to the bus is only limited by the bus capacitance limit of 400 pF and the 7-bit slave address space.

average TWI bus clock period. The SCL frequency is generated according to the following equation:

$$\text{SCL frequency} = \frac{\text{System Clock}}{16 \times (\text{TWBR} + 1)}$$

### 18.3.3 Bus Interface Unit

This unit contains the Data and Address Shift Register (TWDR), a START/STOP Controller and Arbitration detection hardware. The TWDR contains the address or data bytes to be transmitted, or the address or data bytes received. In addition to the 8-bit TWDR, the Bus Interface Unit also contains a register containing the (N)ACK bit to be transmitted or received. This (N)ACK Register is not directly accessible by the application software. However, when receiving, it can be set or cleared by manipulating the TWI Control Register (TWCR). When in Transmitter mode, the value of the received (N)ACK bit can be determined by the value in the TWSR. The START/STOP Controller is responsible for generation and detection of START, REPEATED START, and STOP conditions.

If the TWI has initiated a transmission as Master, the Arbitration Detection hardware continuously monitors the transmission trying to determine if arbitration is in process. If the TWI has lost an arbitration, the Control Unit is informed. Correct action can then be taken and appropriate status codes generated.

### 18.3.4 Address Match Unit

The Address Match unit checks if received address bytes match the 7-bit address in the TWI Address Register (TWAR). If the TWI General Call Recognition Enable (GC) bit in the TWAR is written to one, all incoming address bits will also be compared against the General Call address. Upon an address match, the Control unit is informed, allowing correct action to be taken. The TWI may or may not acknowledge its address, depending on settings in the TWCR.

### 18.3.5 Control Unit

The Control unit monitors the TWI bus and generates responses corresponding to settings in the TWI Control Register (TWCR). When an event requiring the attention of the application occurs on the TWI bus, the TWI Interrupt Flag (TWIF) is asserted. In the next clock cycle, the TWI Status Register (TWSR) is updated with a status code identifying the event. The TWSR only contains relevant status information when the TWI interrupt flag is asserted. At all other times, the TWSR contains a special status code indicating that no relevant status information is available. As long as the TWIF flag is set, the SCL line is held low. This allows the application software to complete its tasks before allowing the TWI transmission to continue.

The TWIF flag is set in the following situations:

- After the TWI has transmitted a START/REPEATED START condition.
- After the TWI has transmitted SLA+R/W.
- After the TWI has transmitted an address byte.
- After the TWI has lost arbitration.
- After the TWI has been addressed by own slave address or general call.
- After the TWI has received a data byte.
- After a STOP or REPEATED START has been received while still addressed as a Slave.
- When a bus error has occurred due to an illegal START or STOP condition.

# 21. Programmable Watchdog Timer

The programmable Watchdog Timer (WDT) protects the system from incorrect execution by triggering a system reset when it times out after the software has failed to feed the timer prior to the timer overflow. By Default the WDT counts CPU clock cycles. The prescaler bits, PS0, PS1 and PS2 in SFR WDTCON are used to set the period of the Watchdog Timer from 16K to 2048K clock cycles. The Timer Prescaler can also be used to lengthen the time-out period (see Table 6-2 on page 33) The WDT is disabled by Reset and during Power-down mode. When the WDT times out without being serviced, an internal RST pulse is generated to reset the CPU. See Table 21-1 for the available WDT period selections.

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

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

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

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

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

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

## 22. Instruction Set Summary

The AT89LP3240/6440 is fully binary compatible with the 8051 instruction set. The difference between the AT89LP3240/6440 and the standard 8051 is the number of cycles required to execute an instruction. Instructions in the AT89LP3240/6440 may take 1 to 9 clock cycles to complete. The execution times of most instructions may be computed using Table 22-1.

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

| Generic Instruction Types | Cycle Count Formula |
|---|---|
| Most arithmetic, logical, bit and transfer instructions | # bytes |
| Branches and Calls | # bytes + 1 |
| Single Byte Indirect (i.e. ADD A, @Ri, etc.) | 2 |
| RET, RETI | 4/5[4] |
| MOVC | 3 |
| MOVX | 2/4[2] |
| MUL | 2 |
| DIV | 4 |
| MAC | 9 |
| 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 |

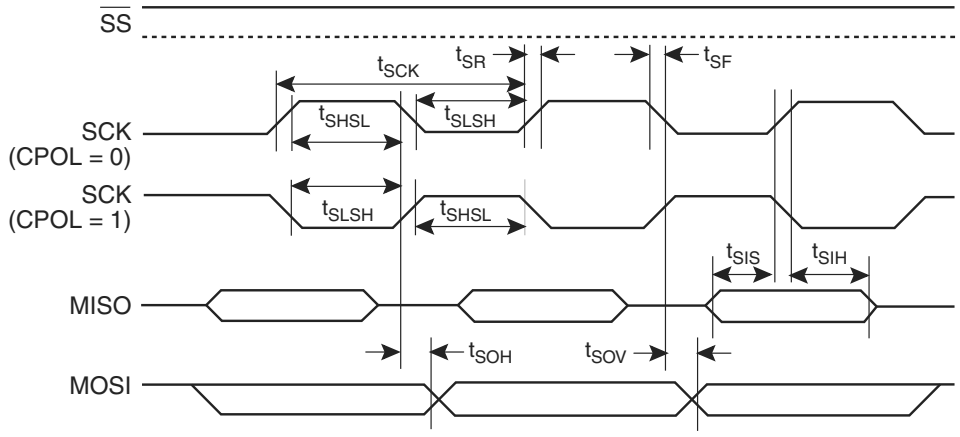
**Table 26-5.** SPI **Master** Characteristics

| Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|
| $t_{SIS}$ | Serial Input Setup Time | 10 | | ns |
| $t_{SIH}$ | Serial Input Hold Time | 10 | | ns |
| $t_{SOH}$ | Serial Output Hold Time | | 10 | ns |
| $t_{SOV}$ | Serial Output Valid Time | | 35 | ns |

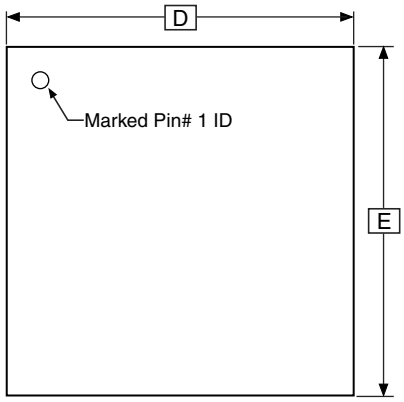
**Table 26-6.** SPI **Slave** Characteristics

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

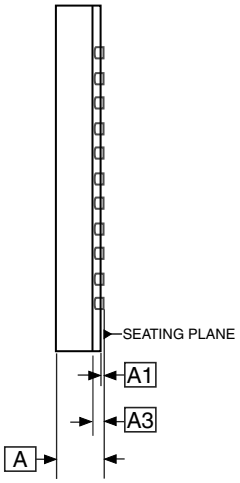
**Figure 26-17.** SPI Master Timing (CPHA = 0)



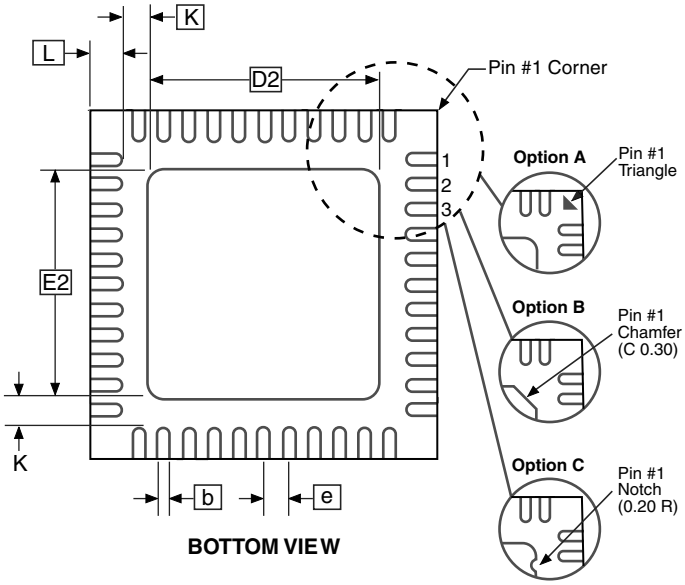
## 28.4 44M1 – VQFN/MLF




TOP VIEW




SIDE VIEW




BOTTOM VIEW

Note: JEDEC Standard MO-220, Fig. 1 (SAW Singulation) VKKD-3.

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

| SYMBOL | MIN | NOM | MAX | NOTE |
|---|---|---|---|---|
| A | 0.80 | 0.90 | 1.00 | |
| A1 | – | 0.02 | 0.05 | |
| A3 | | 0.20 REF | | |
| b | 0.18 | 0.23 | 0.30 | |
| D | 6.90 | 7.00 | 7.10 | |
| D2 | 5.00 | 5.20 | 5.40 | |
| E | 6.90 | 7.00 | 7.10 | |
| E2 | 5.00 | 5.20 | 5.40 | |
| e | | 0.50 BSC | | |
| L | 0.59 | 0.64 | 0.69 | |
| K | 0.20 | 0.26 | 0.41 | |

9/26/08

| Package Drawing Contact: packagedrawings@atmel.com | TITLE | GPC | DRAWING NO. | REV. |
|---|---|---|---|---|
| | **44M1,** 44-pad, 7 x 7 x 1.0 mm Body, Lead Pitch 0.50 mm, 5.20 mm Exposed Pad, Thermally Enhanced Plastic Very Thin Quad Flat No Lead Package (VQFN) | ZWS | 44M1 | H |

## Table of Contents (Continued)