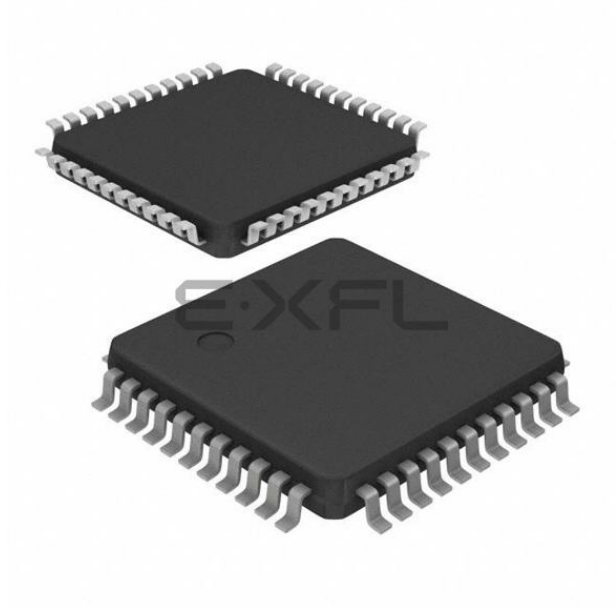


**Welcome to E-XFL.COM**

## 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 | 33MHz |
| Connectivity | EBI/EMI, UART/USART |
| Peripherals | POR |
| Number of I/O | 32 |
| Program Memory Size | - |
| Program Memory Type | ROMless |
| EEPROM Size | - |
| RAM Size | 256 x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.7V ~ 5.5V |
| Data Converters | - |
| Oscillator Type | Internal |
| Operating Temperature | 0°C ~ 70°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 44-LQFP |
| Supplier Device Package | 44-LQFP (10x10) |
| Purchase URL | https://www.e-xfl.com/product-detail/nxp-semiconductors/p80c32x2bbd-157 |

## DESCRIPTION

The Philips microcontrollers described in this data sheet are high-performance static 80C51 designs incorporating Philips' high-density CMOS technology with operation from 2.7 V to 5.5 V. They support both 6-clock and 12-clock operation.

The P8xC31X2/51X2 and P8xC32X2/52X2/54X2/58X2 contain 128 byte RAM and 256 byte RAM respectively, 32 I/O lines, three 16-bit counter/timers, a six-source, four-priority level nested interrupt structure, a serial I/O port for either multi-processor communications, I/O expansion or full duplex UART, and on-chip oscillator and clock circuits.

In addition, the devices are low power static designs which offer a wide range of operating frequencies down to zero. Two software selectable modes of power reduction — idle mode and power-down mode — are available. The idle mode freezes the CPU while allowing the RAM, timers, serial port, and interrupt system to continue functioning. The power-down mode saves the RAM contents but freezes the oscillator, causing all other chip functions to be inoperative. Since the design is static, the clock can be stopped without loss of user data. Then the execution can be resumed from the point the clock was stopped.

## SELECTION TABLE

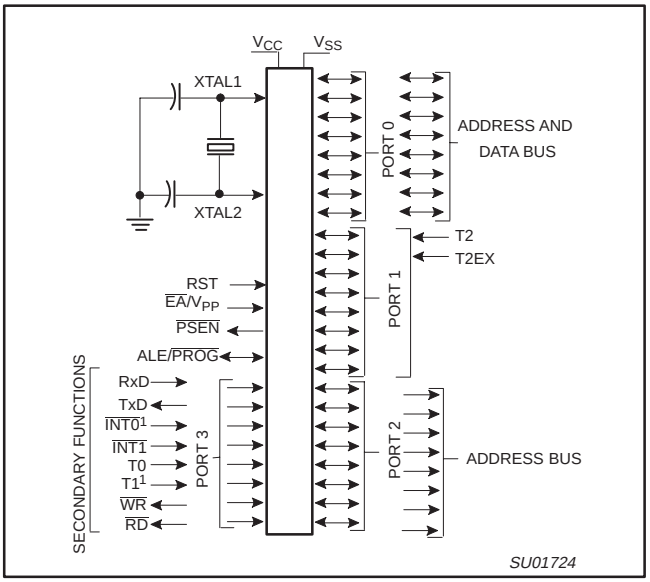
For applications requiring more ROM and RAM, as well as more on-chip peripherals, see the P89C66x and P89C51Rx2 data sheets.

| Type | Memory | | | | Timers | | | | Serial Interfaces | | | | ADC bits/ch. | I/O Pins | Interrupts (External) | Program Security | Default Clock Rate | Optional Clock Rate | Max. Freq. at 6-clk / 12-clk (MHz) | Freq. Range at 3V (MHz) | Freq. Range at 5V (MHz) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RAM | ROM | OTP | Flash | # of Timers | PWM | PCA | WD | UART | $I^2C$ | CAN | SPI | | | | | | | | | |
| P87C58X2 | 256B | – | 32K | – | 3 | – | – | – | ✓ | – | – | – | – | 32 | 6 (2) | ✓ | 12–clk | 6-clk | 30/33 | 0–16 | 0–30/33 |
| P80C58X2 | 256B | 32K | – | – | 3 | – | – | – | ✓ | – | – | – | – | 32 | 6 (2) | ✓ | 12–clk | 6-clk | 30/33 | 0–16 | 0–30/33 |
| P87C54X2 | 256B | – | 16K | – | 3 | – | – | – | ✓ | – | – | – | – | 32 | 6 (2) | ✓ | 12–clk | 6-clk | 30/33 | 0–16 | 0–30/33 |
| P80C54X2 | 256B | 16K | – | – | 3 | – | – | – | ✓ | – | – | – | – | 32 | 6 (2) | ✓ | 12–clk | 6-clk | 30/33 | 0–16 | 0–30/33 |
| P87C52X2 | 256B | – | 8K | – | 3 | – | – | – | ✓ | – | – | – | – | 32 | 6 (2) | ✓ | 12–clk | 6-clk | 30/33 | 0–16 | 0–30/33 |
| P80C52X2 | 256B | 8K | – | – | 3 | – | – | – | ✓ | – | – | – | – | 32 | 6 (2) | ✓ | 12–clk | 6-clk | 30/33 | 0–16 | 0–30/33 |
| P87C51X2 | 128B | – | 4K | – | 3 | – | – | – | ✓ | – | – | – | – | 32 | 6 (2) | ✓ | 12–clk | 6-clk | 30/33 | 0–16 | 0–30/33 |
| P80C51X2 | 128B | 4K | – | – | 3 | – | – | – | ✓ | – | – | – | – | 32 | 6 (2) | ✓ | 12–clk | 6-clk | 30/33 | 0–16 | 0–30/33 |
| P80C32X2 | 256B | – | – | – | 3 | – | – | – | ✓ | – | – | – | – | 32 | 6 (2) | – | 12–clk | 6-clk | 30/33 | 0–16 | 0–30/33 |
| P80C31X2 | 128B | – | – | – | 3 | – | – | – | ✓ | – | – | – | – | 32 | 6 (2) | – | 12–clk | 6-clk | 30/33 | 0–16 | 0–30/33 |

**NOTE:**

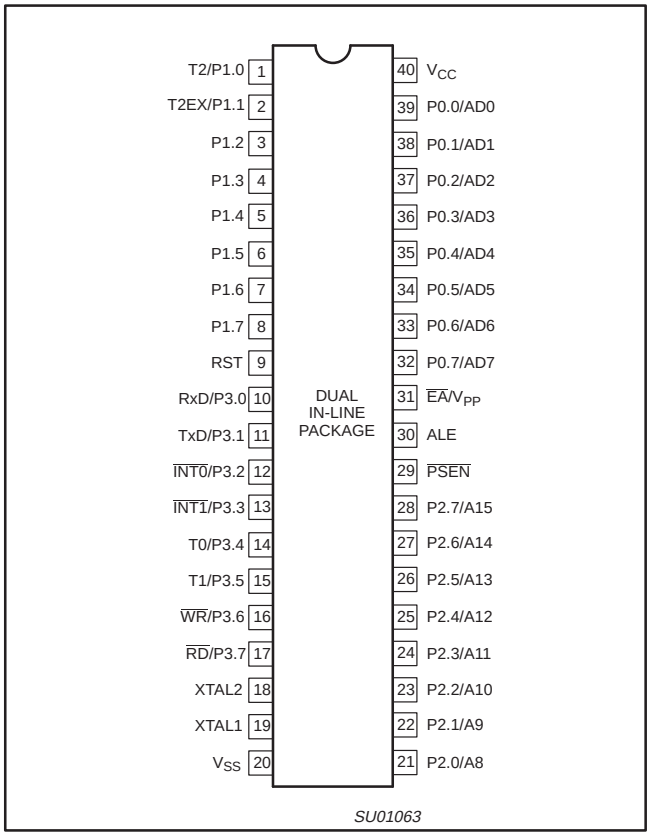
1. $I^2C$ = Inter-Integrated Circuit Bus; CAN = Controller Area Network; SPI = Serial Peripheral Interface; PCA = Programmable Counter Array; ADC = Analog-to-Digital Converter; PWM = Pulse Width Modulation

## LOGIC SYMBOL

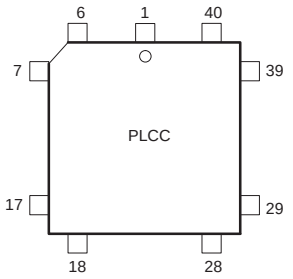



**NOTE:**
1. $\overline{INT0}$/P3.2 and T1/P3.5 are absent in the TSSOP38 package.

## PLASTIC DUAL IN-LINE PACKAGE PIN CONFIGURATIONS



## PLASTIC LEADED CHIP CARRIER PIN FUNCTIONS

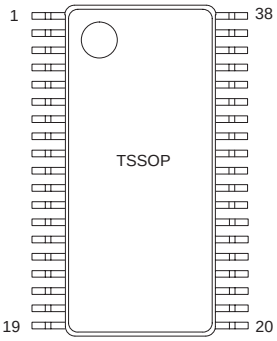



| Pin | Function | Pin | Function | Pin | Function |
|---|---|---|---|---|---|
| 1 | NIC* | 16 | P3.4/T0 | 31 | P2.7/A15 |
| 2 | P1.0/T2 | 17 | P3.5/T1 | 32 | $\overline{PSEN}$ |
| 3 | P1.1/T2EX | 18 | P3.6/$\overline{WR}$ | 33 | ALE |
| 4 | P1.2 | 19 | P3.7/$\overline{RD}$ | 34 | NIC* |
| 5 | P1.3 | 20 | XTAL2 | 35 | $\overline{EA}$/$V_{PP}$ |
| 6 | P1.4 | 21 | XTAL1 | 36 | P0.7/AD7 |
| 7 | P1.5 | 22 | $V_{SS}$ | 37 | P0.6/AD6 |
| 8 | P1.6 | 23 | NIC* | 38 | P0.5/AD5 |
| 9 | P1.7 | 24 | P2.0/A8 | 39 | P0.4/AD4 |
| 10 | RST | 25 | P2.1/A9 | 40 | P0.3/AD3 |
| 11 | P3.0/RxD | 26 | P2.2/A10 | 41 | P0.2/AD2 |
| 12 | NIC* | 27 | P2.3/A11 | 42 | P0.1/AD1 |
| 13 | P3.1/TxD | 28 | P2.4/A12 | 43 | P0.0/AD0 |
| 14 | P3.2/$\overline{INT0}$ | 29 | P2.5/A13 | 44 | $V_{CC}$ |
| 15 | P3.3/$\overline{INT1}$ | 30 | P2.6/A14 | | |

* NO INTERNAL CONNECTION

*SU01062*

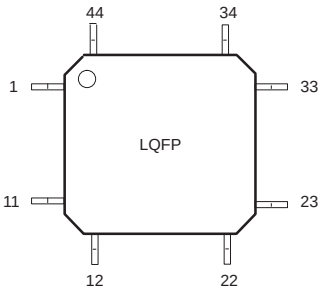
## PLASTIC THIN SHRINK SMALL OUTLINE PACK PIN FUNCTIONS




| Pin | Function | Pin | Function | Pin | Function |
|---|---|---|---|---|---|
| 1 | P3.0/RxD | 14 | P2.4/A12 | 27 | P0.1/AD1 |
| 2 | P3.1/TxD | 15 | P2.5/A13 | 28 | P0.0/AD0 |
| 3 | P3.3/$\overline{INT1}$ | 16 | P2.6/A14 | 29 | $V_{DD}$ |
| 4 | P3.4/T0 | 17 | P2.7/A15 | 30 | P1.0/T2 |
| 5 | P3.6/$\overline{WR}$ | 18 | $\overline{PSEN}$ | 31 | P1.1/T2EX |
| 6 | P3.7/$\overline{RD}$ | 19 | ALE/$\overline{PROG}$ | 32 | P1.2 |
| 7 | XTAL2 | 20 | $\overline{EA}$/$V_{PP}$ | 33 | P1.3 |
| 8 | XTAL1 | 21 | P0.7/AD7 | 34 | P1.4 |
| 9 | $V_{SS}$ | 22 | P0.6/AD6 | 35 | P1.5 |
| 10 | P2.0/A8 | 23 | P0.5/AD5 | 36 | P1.6 |
| 11 | P2.1/A9 | 24 | P0.4/AD4 | 37 | P1.7 |
| 12 | P2.2/A10 | 25 | P0.3/AD3 | 38 | RST |
| 13 | P2.3/A11 | 26 | P0.2/AD2 | | |

*su01725*

## LOW PROFILE QUAD FLAT PACK PIN FUNCTIONS




| Pin | Function | Pin | Function | Pin | Function |
|---|---|---|---|---|---|
| 1 | P1.5 | 16 | $V_{SS}$ | 31 | P0.6/AD6 |
| 2 | P1.6 | 17 | NIC* | 32 | P0.5/AD5 |
| 3 | P1.7 | 18 | P2.0/A8 | 33 | P0.4/AD4 |
| 4 | RST | 19 | P2.1/A9 | 34 | P0.3/AD3 |
| 5 | P3.0/RxD | 20 | P2.2/A10 | 35 | P0.2/AD2 |
| 6 | NIC* | 21 | P2.3/A11 | 36 | P0.1/AD1 |
| 7 | P3.1/TxD | 22 | P2.4/A12 | 37 | P0.0/AD0 |
| 8 | P3.2/$\overline{INT0}$ | 23 | P2.5/A13 | 38 | $V_{CC}$ |
| 9 | P3.3/$\overline{INT1}$ | 24 | P2.6/A14 | 39 | NIC* |
| 10 | P3.4/T0 | 25 | P2.7/A15 | 40 | P1.0/T2 |
| 11 | P3.5/T1 | 26 | $\overline{PSEN}$ | 41 | P1.1/T2EX |
| 12 | P3.6/$\overline{WR}$ | 27 | ALE | 42 | P1.2 |
| 13 | P3.7/$\overline{RD}$ | 28 | NIC* | 43 | P1.3 |
| 14 | XTAL2 | 29 | $\overline{EA}$/$V_{PP}$ | 44 | P1.4 |
| 15 | XTAL1 | 30 | P0.7/AD7 | | |

* NO INTERNAL CONNECTION

*SU01487*

## PIN DESCRIPTIONS

| MNEMONIC | PIN NUMBER DIP | PLCC | LQFP | TSSOP | TYPE | NAME AND FUNCTION |
|---|---|---|---|---|---|---|
| $V_{SS}$ | 20 | 22 | 16 | 9 | I | **Ground:** 0 V reference. |
| $V_{CC}$ | 40 | 44 | 38 | 29 | I | **Power Supply:** This is the power supply voltage for normal, idle, and power-down operation. |
| P0.0-0.7 | 39–32 | 43–36 | 37–30 | 28–21 | I/O | **Port 0:** Port 0 is an open-drain, bidirectional I/O port. Port 0 pins that have 1s written to them float and can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program and data memory. In this application, it uses strong internal pull-ups when emitting 1s. Port 0 also outputs the code bytes during program verification and received code bytes during EPROM programming. External pull-ups are required during program verification. |
| P1.0–P1.7 | 1–8 | 2–9 | 40–44, 1–3 | 30–37 | I/O | **Port 1:** Port 1 is an 8-bit bidirectional I/O port with internal pull-ups. Port 1 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 1 pins that are externally pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: $I_{IL}$). Port 1 also receives the low-order address byte during program memory verification. Alternate functions for Port 1 include: |
| | 1 | 2 | 40 | 30 | I/O | **T2 (P1.0):** Timer/Counter 2 external count input/clockout (see Programmable Clock-Out) |
| | 2 | 3 | 41 | 31 | I | **T2EX (P1.1):** Timer/Counter 2 Reload/Capture/Direction control |
| P2.0–P2.7 | 21–28 | 24–31 | 18–25 | 10–17 | I/O | **Port 2:** Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. Port 2 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 2 pins that are externally being pulled low will source current because of the internal pull-ups. (See DC Electrical Characteristics: $I_{IL}$). Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @DPTR). In this application, it uses strong internal pull-ups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOV @Ri), port 2 emits the contents of the P2 special function register. Some Port 2 pins receive the high order address bits during EPROM programming and verification. |
| P3.0–P3.7 | 10–17 | 11, 13–19 | 5, 7–13 | 1–6 | I/O | **Port 3:** Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. Port 3 pins that have 1s written to them are pulled high by the internal pull-ups and can be used as inputs. As inputs, port 3 pins that are externally being pulled low will source current because of the pull-ups. (See DC Electrical Characteristics: $I_{IL}$). Port 3 also serves the special features of the 80C51 family, as listed below: |
| | 10 | 11 | 5 | 1 | I | **RxD (P3.0):** Serial input port |
| | 11 | 13 | 7 | 2 | O | **TxD (P3.1):** Serial output port |
| | 12 | 14 | 8 | | I | **$\overline{\text{INT0}}$ (P3.2):** External interrupt[1] |
| | 13 | 15 | 9 | 3 | I | **$\overline{\text{INT1}}$ (P3.3):** External interrupt |
| | 14 | 16 | 10 | 4 | I | **T0 (P3.4):** Timer 0 external input |
| | 15 | 17 | 11 | | I | **T1 (P3.5):** Timer 1 external input[1] |
| | 16 | 18 | 12 | 5 | O | **$\overline{\text{WR}}$ (P3.6):** External data memory write strobe |
| | 17 | 19 | 13 | 6 | O | **$\overline{\text{RD}}$ (P3.7):** External data memory read strobe |
| RST | 9 | 10 | 4 | 38 | I | **Reset:** A high on this pin for two machine cycles while the oscillator is running, resets the device. An internal diffused resistor to $V_{SS}$ permits a power-on reset using only an external capacitor to $V_{CC}$. |
| ALE/$\overline{\text{PROG}}$ | 30 | 33 | 27 | 19 | O | **Address Latch Enable/Program Pulse:** Output pulse for latching the low byte of the address during an access to external memory. In normal operation, ALE is emitted at a constant rate of 1/6 (12-clock Mode) or 1/3 (6-clock Mode) the oscillator frequency, and can be used for external timing or clocking. Note that one ALE pulse is skipped during each access to external data memory. This pin is also the program pulse input ($\overline{\text{PROG}}$) during EPROM programming. ALE can be disabled by setting SFR auxiliary.0. With this bit set, ALE will be active only during a MOVX instruction. |

**Table 1. Special Function Registers**

| SYMBOL | DESCRIPTION | DIRECT ADDRESS | BIT ADDRESS, SYMBOL, OR ALTERNATIVE PORT FUNCTION MSB | | | | | | | LSB | RESET VALUE |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ACC* | Accumulator | E0H | E7 | E6 | E5 | E4 | E3 | E2 | E1 | E0 | 00H |
| AUXR# | Auxiliary | 8EH | – | – | – | – | – | – | – | AO | xxxxxxx0B |
| AUXR1# | Auxiliary 1 | A2H | – | – | – | LPEP[2] | WUPD | 0 | – | DPS | xxx000x0B |
| B* | B register | F0H | F7 | F6 | F5 | F4 | F3 | F2 | F1 | F0 | 00H |
| CKCON | Clock Control Register | 8FH | – | – | – | – | – | – | – | X2 | xxx00000B |
| DPTR: | Data Pointer (2 bytes) | | | | | | | | | | |
| DPH | Data Pointer High | 83H | | | | | | | | | 00H |
| DPL | Data Pointer Low | 82H | | | | | | | | | 00H |
| | | | AF | AE | AD | AC | AB | AA | A9 | A8 | |
| IE* | Interrupt Enable | A8H | $\overline{EA}$ | – | ET2 | ES | ET1 | EX1 | ET0 | EX0 | 0x000000B |
| | | | BF | BE | BD | BC | BB | BA | B9 | B8 | |
| IP* | Interrupt Priority | B8H | – | – | PT2 | PS | PT1 | PX1 | PT0 | PX0 | xx000000B |
| IPH# | Interrupt Priority High | B7H | – | – | PT2H | PSH | PT1H | PX1H | PT0H | PX0H | xx000000B |
| | | | 87 | 86 | 85 | 84 | 83 | 82 | 81 | 80 | |
| P0* | Port 0 | 80H | AD7 | AD6 | AD5 | AD4 | AD3 | AD2 | AD1 | AD0 | FFH |
| | | | 97 | 96 | 95 | 94 | 93 | 92 | 91 | 90 | |
| P1* | Port 1 | 90H | – | – | – | – | – | – | T2EX | T2 | FFH |
| | | | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 | |
| P2* | Port 2 | A0H | AD15 | AD14 | AD13 | AD12 | AD11 | AD10 | AD9 | AD8 | FFH |
| | | | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | |
| P3* | Port 3 | B0H | $\overline{RD}$ | $\overline{WR}$ | T1 | T0 | $\overline{INT1}$ | $\overline{INT0}$ | TxD | RxD | FFH |
| | | | | | | | | | | | |
| PCON#[1] | Power Control | 87H | SMOD1 | SMOD0 | – | POF | GF1 | GF0 | PD | IDL | 00xx0000B |
| | | | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | |
| PSW* | Program Status Word | D0H | CY | AC | F0 | RS1 | RS0 | OV | – | P | 000000x0B |
| RACAP2H# | Timer 2 Capture High | CBH | | | | | | | | | 00H |
| RACAP2L# | Timer 2 Capture Low | CAH | | | | | | | | | 00H |
| SADDR# | Slave Address | A9H | | | | | | | | | 00H |
| SADEN# | Slave Address Mask | B9H | | | | | | | | | 00H |
| SBUF | Serial Data Buffer | 99H | | | | | | | | | xxxxxxxxB |
| | | | 9F | 9E | 9D | 9C | 9B | 9A | 99 | 98 | |
| SCON* | Serial Control | 98H | SM0/FE | SM1 | SM2 | REN | TB8 | RB8 | TI | RI | 00H |
| SP | Stack Pointer | 81H | | | | | | | | | 07H |
| | | | 8F | 8E | 8D | 8C | 8B | 8A | 89 | 88 | |
| TCON* | Timer Control | 88H | TF1 | TR1 | TF0 | TR0 | IE1 | IT1 | IE0 | IT0 | 00H |
| | | | CF | CE | CD | CC | CB | CA | C9 | C8 | |
| T2CON* | Timer 2 Control | C8H | TF2 | EXF2 | RCLK | TCLK | EXEN2 | TR2 | C/$\overline{T}$2 | CP/$\overline{RL}$2 | 00H |
| T2MOD# | Timer 2 Mode Control | C9H | – | – | – | – | – | – | T2OE | DCEN | xxxxxx00B |
| TH0 | Timer High 0 | 8CH | | | | | | | | | 00H |
| TH1 | Timer High 1 | 8DH | | | | | | | | | 00H |
| TH2# | Timer High 2 | CDH | | | | | | | | | 00H |
| TL0 | Timer Low 0 | 8AH | | | | | | | | | 00H |
| TL1 | Timer Low 1 | 8BH | | | | | | | | | 00H |
| TL2# | Timer Low 2 | CCH | | | | | | | | | 00H |
| TMOD | Timer Mode | 89H | GATE | C/$\overline{T}$ | M1 | M0 | GATE | C/$\overline{T}$ | M1 | M0 | 00H |

**NOTE:**
Unused register bits that are not defined should not be set by the user's program. If violated, the device could function incorrectly.
* SFRs are bit addressable.
\# SFRs are modified from or added to the 80C51 SFRs.
– Reserved bits.
1. Reset value depends on reset source.
2. LPEP – Low Power EPROM operation (OTP only)

### Low-Power EPROM operation (LPEP)

The EPROM array contains some analog circuits that are not required when $V_{CC}$ is less than 4 V, but are required for a $V_{CC}$ greater than 4 V. The LPEP bit (AUXR.4), when set, will powerdown these analog circuits resulting in a reduced supply current. This bit should be set ONLY for applications that operate at a $V_{CC}$ less than 4 V.

### Design Consideration

When the idle mode is terminated by a hardware reset, the device normally resumes program execution from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

### ONCE™ Mode

The ONCE ("On-Circuit Emulation") Mode facilitates testing and debugging of systems without the device having to be removed from the circuit. The ONCE Mode is invoked in the following way:

1. Pull ALE low while the device is in reset and $\overline{PSEN}$ is high;
2. Hold ALE low as RST is deactivated.

While the device is in ONCE Mode, the Port 0 pins go into a float state, and the other port pins and ALE and $\overline{PSEN}$ are weakly pulled high. The oscillator circuit remains active. While the device is in this mode, an emulator or test CPU can be used to drive the circuit. Normal operation is restored when a normal reset is applied.

**Table 3. External Pin Status During Idle and Power-Down Modes**

| MODE | PROGRAM MEMORY | ALE | $\overline{PSEN}$ | PORT 0 | PORT 1 | PORT 2 | PORT 3 |
|---|---|---|---|---|---|---|---|
| Idle | Internal | 1 | 1 | Data | Data | Data | Data |
| Idle | External | 1 | 1 | Float | Data | Address | Data |
| Power-down | Internal | 0 | 0 | Data | Data | Data | Data |
| Power-down | External | 0 | 0 | Float | Data | Data | Data |

## TIMER 0 AND TIMER 1 OPERATION

### Timer 0 and Timer 1

The "Timer" or "Counter" function is selected by control bits C/$\overline{T}$ in the Special Function Register TMOD. These two Timer/Counters have four operating modes, which are selected by bit-pairs (M1, M0) in TMOD. Modes 0, 1, and 2 are the same for both Timers/Counters. Mode 3 is different. The four operating modes are described in the following text.

**Mode 0**

Putting either Timer into Mode 0 makes it look like an 8048 Timer, which is an 8-bit Counter with a divide-by-32 prescaler. Figure 2 shows the Mode 0 operation.

In this mode, the Timer register is configured as a 13-bit register. As the count rolls over from all 1s to all 0s, it sets the Timer interrupt flag TFn. The counted input is enabled to the Timer when TRn = 1 and either GATE = 0 or $\overline{INTn}$ = 1. (Setting GATE = 1 allows the Timer to be controlled by external input $\overline{INTn}$, to facilitate pulse width measurements). TRn is a control bit in the Special Function Register TCON (Figure 3).

The 13-bit register consists of all 8 bits of THn and the lower 5 bits of TLn. The upper 3 bits of TLn are indeterminate and should be ignored. Setting the run flag (TRn) does not clear the registers.

Mode 0 operation is the same for Timer 0 as for Timer 1. There are two different GATE bits, one for Timer 1 (TMOD.7) and one for Timer 0 (TMOD.3).

**Mode 1**

Mode 1 is the same as Mode 0, except that the Timer register is being run with all 16 bits.

**Mode 2**

Mode 2 configures the Timer register as an 8-bit Counter (TLn) with automatic reload, as shown in Figure 4. Overflow from TLn not only sets TFn, but also reloads TLn with the contents of THn, which is preset by software. The reload leaves THn unchanged.

Mode 2 operation is the same for Timer 0 as for Timer 1.

**Mode 3**

Timer 1 in Mode 3 simply holds its count. The effect is the same as setting TR1 = 0.

Timer 0 in Mode 3 establishes TL0 and TH0 as two separate counters. The logic for Mode 3 on Timer 0 is shown in Figure 5. TL0 uses the Timer 0 control bits: C/$\overline{T}$, GATE, TR0, and TF0 as well as pin $\overline{INT0}$. TH0 is locked into a timer function (counting machine cycles) and takes over the use of TR1 and TF1 from Timer 1. Thus, TH0 now controls the "Timer 1" interrupt.

Mode 3 is provided for applications requiring an extra 8-bit timer on the counter. With Timer 0 in Mode 3, an 80C51 can look like it has three Timer/Counters. When Timer 0 is in Mode 3, Timer 1 can be turned on and off by switching it out of and into its own Mode 3, or can still be used by the serial port as a baud rate generator, or in fact, in any application not requiring an interrupt.

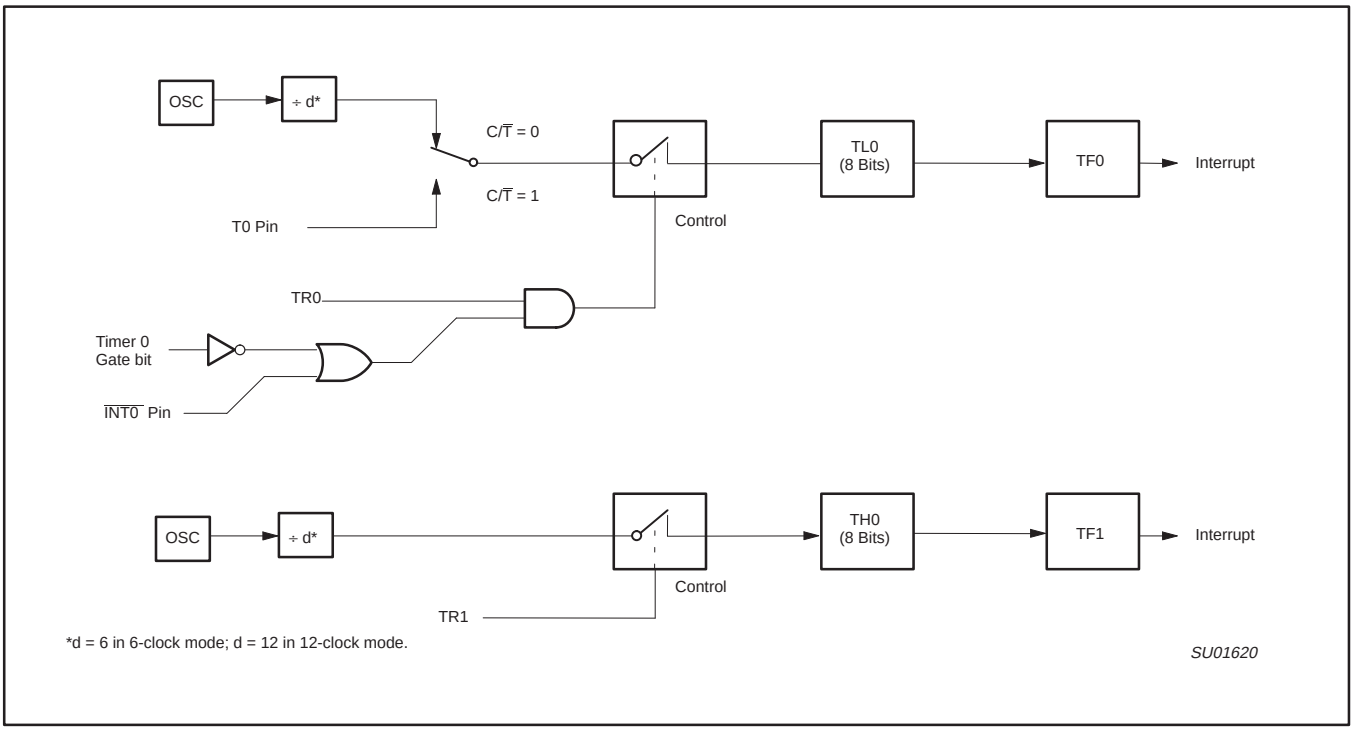



**Figure 5. Timer/Counter 0 Mode 3: Two 8-Bit Counters**

## TIMER 2 OPERATION

### Timer 2

Timer 2 is a 16-bit Timer/Counter which can operate as either an event timer or an event counter, as selected by C/$\overline{T}$2 in the special function register T2CON (see Figure 6). Timer 2 has three operating modes: Capture, Auto-reload (up or down counting), and Baud Rate Generator, which are selected by bits in the T2CON as shown in Table 4.

### Capture Mode

In the capture mode there are two options which are selected by bit EXEN2 in T2CON. If EXEN2=0, then timer 2 is a 16-bit timer or counter (as selected by C/$\overline{T}$2 in T2CON) which, upon overflowing, sets bit TF2, the timer 2 overflow bit. This bit can be used to generate an interrupt (by enabling the Timer 2 interrupt bit in the IE register). If EXEN2=1, Timer 2 operates as described above, but with the added feature that a 1-to-0 transition at external input T2EX causes the current value in the Timer 2 registers, TL2 and TH2, to be captured into registers RCAP2L and RCAP2H, respectively. In addition, the transition at T2EX causes bit EXF2 in T2CON to be set, and EXF2 (like TF2) can generate an interrupt (which vectors to the same location as Timer 2 overflow interrupt. The Timer 2 interrupt service routine can interrogate TF2 and EXF2 to determine which event caused the interrupt). The capture mode is illustrated in Figure 7 (There is no reload value for TL2 and TH2 in this mode. Even when a capture event occurs from T2EX, the counter keeps on counting T2EX pin transitions or osc/12 (12-clock Mode) or osc/6 (6-clock Mode) pulses).

### Auto-Reload Mode (Up or Down Counter)

In the 16-bit auto-reload mode, Timer 2 can be configured as either a timer or counter (C/$\overline{T}$2 in T2CON), then programmed to count up or down. The counting direction is determined by bit DCEN (Down Counter Enable) which is located in the T2MOD register (see Figure 8). After reset, DCEN=0 which means Timer 2 will default to counting up. If DCEN is set, Timer 2 can count up or down depending on the value of the T2EX pin.

Figure 9 shows Timer 2 which will count up automatically since DCEN=0. In this mode there are two options selected by bit EXEN2 in T2CON register. If EXEN2=0, then Timer 2 counts up to 0FFFFH and sets the TF2 (Overflow Flag) bit upon overflow. This causes the Timer 2 registers to be reloaded with the 16-bit value in RCAP2L and RCAP2H. The values in RCAP2L and RCAP2H are preset by software.

If EXEN2=1, then a 16-bit reload can be triggered either by an overflow or by a 1-to-0 transition at input T2EX. This transition also sets the EXF2 bit. The Timer 2 interrupt, if enabled, can be generated when either TF2 or EXF2 are 1.

In Figure 10 DCEN=1 which enables Timer 2 to count up or down. This mode allows pin T2EX to control the direction of count. When a logic 1 is applied at pin T2EX, Timer 2 will count up. Timer 2 will overflow at 0FFFFH and set the TF2 flag, which can then generate an interrupt, if the interrupt is enabled. This timer overflow also causes the 16-bit value in RCAP2L and RCAP2H to be reloaded into the timer registers TL2 and TH2.

A logic 0 applied to pin T2EX causes Timer 2 to count down. The timer will underflow when TL2 and TH2 become equal to the value stored in RCAP2L and RCAP2H. A Timer 2 underflow sets the TF2 flag and causes 0FFFFH to be reloaded into the timer registers TL2 and TH2.

The external flag EXF2 toggles when Timer 2 underflows or overflows. This EXF2 bit can be used as a 17th bit of resolution if needed. The EXF2 flag does not generate an interrupt in this mode of operation.

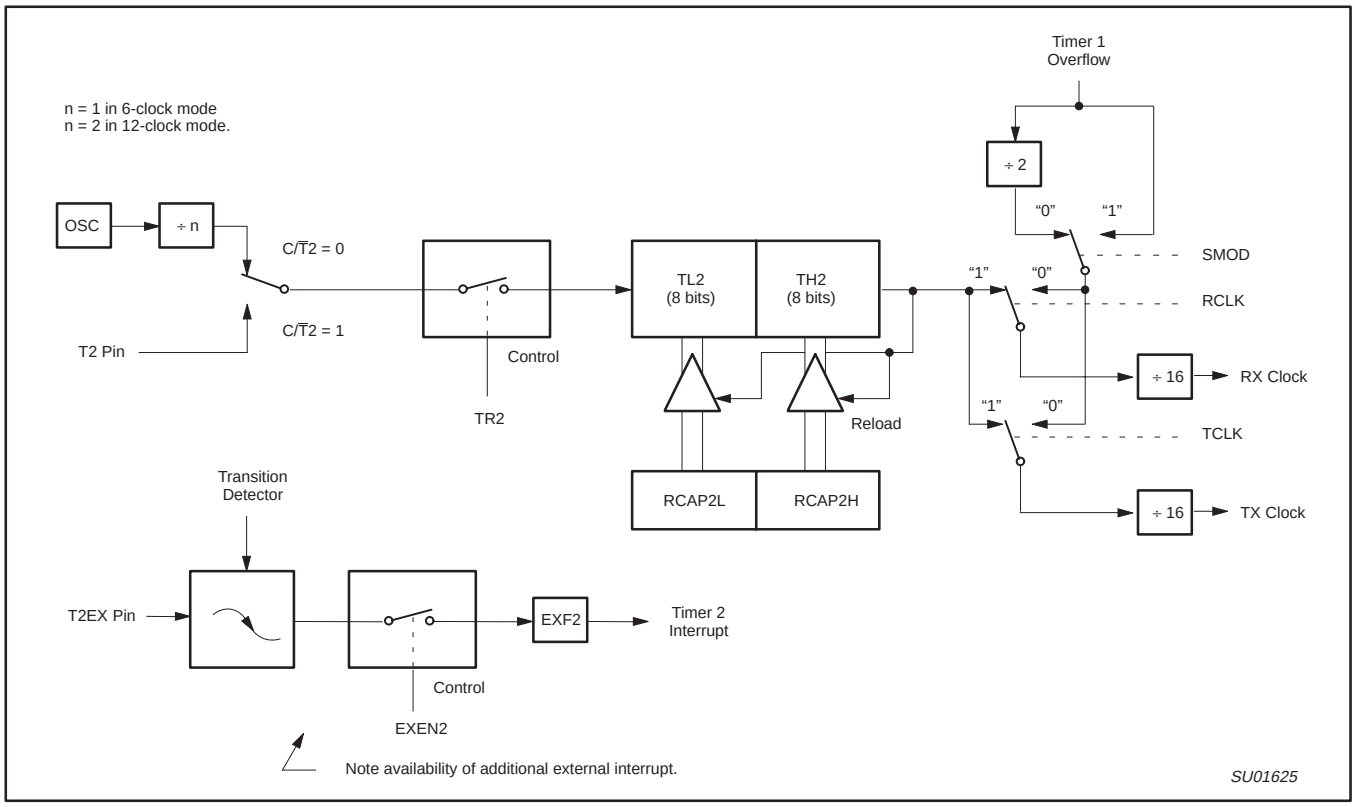



**Figure 11. Timer 2 in Baud Rate Generator Mode**

## Baud Rate Generator Mode

Bits TCLK and/or RCLK in T2CON (Table 4) allow the serial port transmit and receive baud rates to be derived from either Timer 1 or Timer 2. When TCLK= 0, Timer 1 is used as the serial port transmit baud rate generator. When TCLK= 1, Timer 2 is used as the serial port transmit baud rate generator. RCLK has the same effect for the serial port receive baud rate. With these two bits, the serial port can have different receive and transmit baud rates – one generated by Timer 1, the other by Timer 2.

Figure 11 shows the Timer 2 in baud rate generation mode. The baud rate generation mode is like the auto-reload mode, in that a rollover in TH2 causes the Timer 2 registers to be reloaded with the 16-bit value in registers RCAP2H and RCAP2L, which are preset by software.

The baud rates in modes 1 and 3 are determined by Timer 2's overflow rate given below:

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

The timer can be configured for either "timer" or "counter" operation. In many applications, it is configured for "timer" operation (C/$\overline{T}$2=0). Timer operation is different for Timer 2 when it is being used as a baud rate generator.

Usually, as a timer it would increment every machine cycle (i.e., 1/6 the oscillator frequency in 6-clock mode or 1/12 the oscillator frequency in 12-clock mode). As a baud rate generator, it increments at the oscillator frequency in 6-clock mode or at 1/2 the oscillator frequency in 12-clock mode. Thus the baud rate formula is as follows:

$$\text{Modes 1 and 3 Baud Rates} = \frac{\text{Oscillator Frequency}}{[n \times [65536 - (\text{RCAP2H, RCAP2L})]]}$$

Where:

n = 16 in 6-clock mode, 32 in 12-clock mode.

(RCAP2H, RCAP2L)= The content of RCAP2H and RCAP2L taken as a 16-bit unsigned integer.

The Timer 2 as a baud rate generator mode shown in Figure 11 is valid only if RCLK and/or TCLK = 1 in T2CON register. Note that a rollover in TH2 does not set TF2, and will not generate an interrupt. Thus, the Timer 2 interrupt does not have to be disabled when Timer 2 is in the baud rate generator mode. Also if the EXEN2 (T2 external enable flag) is set, a 1-to-0 transition in T2EX (Timer/counter 2 trigger input) will set EXF2 (T2 external flag) but will not cause a reload from (RCAP2H, RCAP2L) to (TH2,TL2). Therefore when Timer 2 is in use as a baud rate generator, T2EX can be used as an additional external interrupt, if needed.

When Timer 2 is in the baud rate generator mode, one should not try to read or write TH2 and TL2. As a baud rate generator, Timer 2 is incremented every state time (osc/2) or asynchronously from pin T2; under these conditions, a read or write of TH2 or TL2 may not be accurate. The RCAP2 registers may be read, but should not be written to, because a write might overlap a reload and cause write and/or reload errors. The timer should be turned off (clear TR2) before accessing the Timer 2 or RCAP2 registers.

Table 5 shows commonly used baud rates and how they can be obtained from Timer 2.

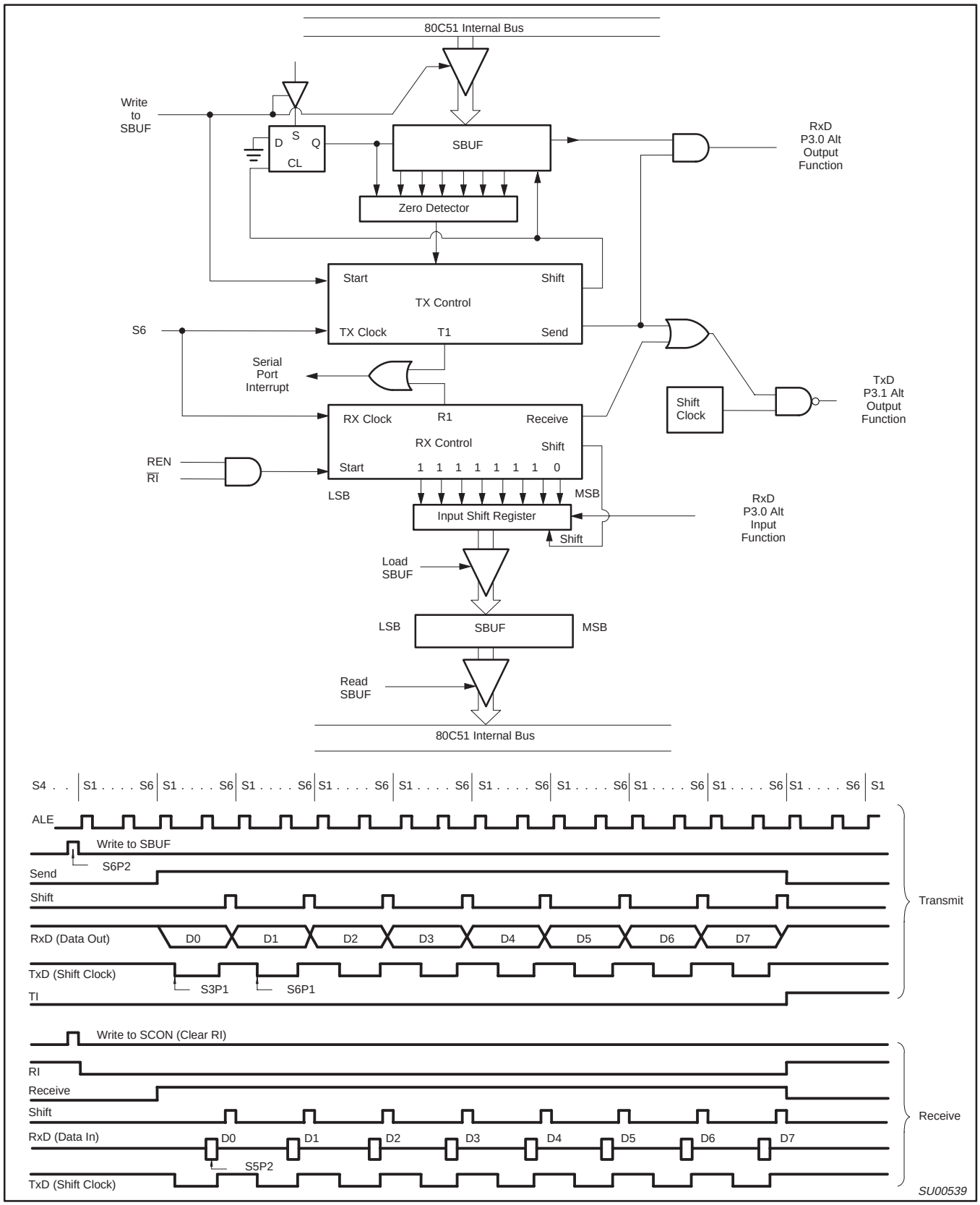



**Figure 14. Serial Port Mode 0**

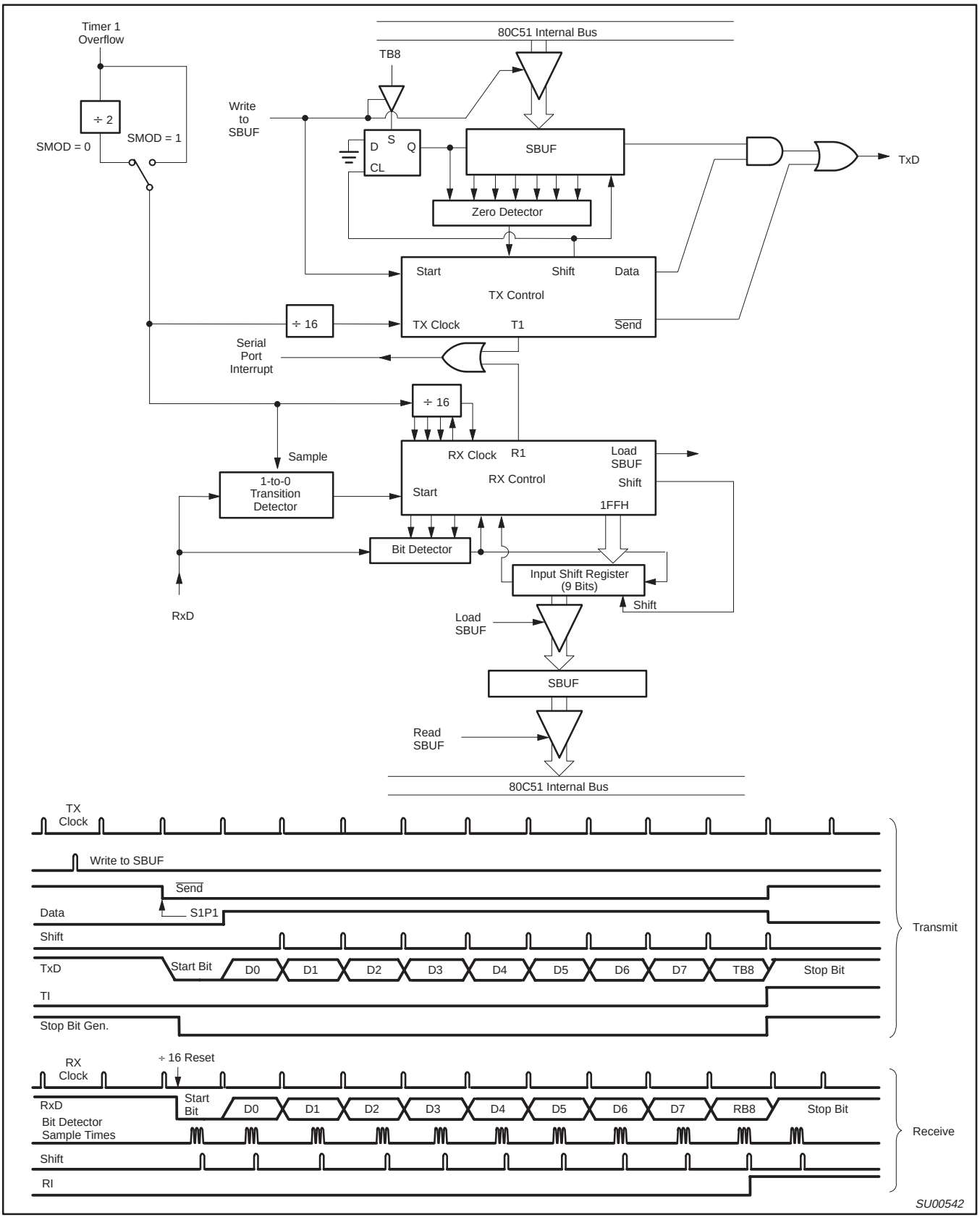



**Figure 17. Serial Port Mode 3**

## Interrupt Priority Structure

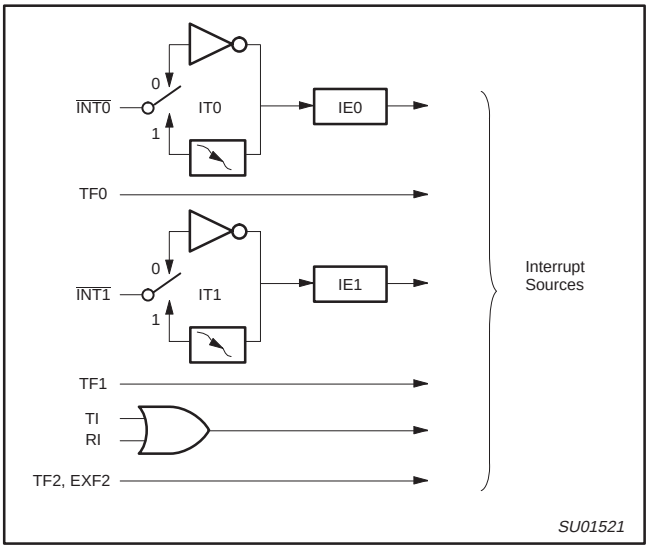



**Figure 21. Interrupt Sources**

## Interrupts

The devices described in this data sheet provide six interrupt sources. These are shown in Figure 21. The External Interrupts $\overline{INT0}$ and $\overline{INT1}$ can each be either level-activated or transition-activated, depending on bits IT0 and IT1 in Register TCON. The flags that actually generate these interrupts are bits IE0 and IE1 in TCON. When an external interrupt is generated, the flag that generated it is cleared by the hardware when the service routine is vectored to only if the interrupt was transition-activated. If the interrupt was level-activated, then the external requesting source is what controls the request flag, rather than the on-chip hardware.

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 see Timer 0 in Mode 3). When a timer interrupt is generated, the flag that generated it is cleared by the on-chip hardware when the service routine is vectored to.

The Serial Port Interrupt is generated by the logical OR of RI and TI. Neither of these flags is cleared by hardware when the service routine is vectored to. In fact, the service routine will normally have to determine whether it was RI or TI that generated the interrupt, and the bit will have to be cleared in software.

All of the bits that generate interrupts can be set or cleared by software, with the same result as though it had been set or cleared by hardware. That is, interrupts can be generated or pending interrupts can be canceled in software.

Each of these interrupt sources can be individually enabled or disabled by setting or clearing a bit in Special Function Register IE (Figure 22). IE also contains a global disable bit, $\overline{EA}$, which disables all interrupts at once.

**Priority Level Structure**

Each interrupt source can also be individually programmed to one of four priority levels by setting or clearing bits in Special Function Registers IP (Figure 23) and IPH (Figure 24). A lower-priority interrupt can itself be interrupted by a higher-priority interrupt, but not by another interrupt of the same level. A high-priority level 3 interrupt can't be interrupted by any other interrupt source.

If two request of different priority levels are received simultaneously, the request of higher priority level is serviced. If requests of the same priority level are received simultaneously, an internal polling sequence determines which request is serviced. Thus within each priority level there is a second priority structure determined by the polling sequence as follows:

| Source | Priority Within Level |
|---|---|
| 1. IE0 (External Int 0) | (highest) |
| 2. TF0 (Timer 0) | |
| 3. IE1 (External Int 1) | |
| 4. TF1 (Timer 1) | |
| 5. RI+TI (UART) | |
| 6. TF2, EXF2 (Timer 2) | (lowest) |

Note that the "priority within level" structure is only used to resolve simultaneous requests of the same priority level.

The IP and IPH registers contain a number of unimplemented bits. User software should not write 1s to these positions, since they may be used in other 80C51 Family products.

**How Interrupts Are Handled**

The interrupt flags are sampled at S5P2 of every machine cycle. The samples are polled during the following machine cycle. If one of the flags was in a set condition at S5P2 of the preceding cycle, the polling cycle will find it and the interrupt system will generate an LCALL to the appropriate service routine, provided this hardware-generated LCALL is not blocked by any of the following conditions:

1. An interrupt of equal or higher priority level is already in progress.
2. The current (polling) cycle is not the final cycle in the execution of the instruction in progress.
3. The instruction in progress is RETI or any write to the IE or IP registers.

Any of these three conditions will block the generation of the LCALL to the interrupt service routine. Condition 2 ensures that the instruction in progress will be completed before vectoring to any service routine. Condition 3 ensures that if the instruction in progress is RETI or any access to IE or IP, then at least one more instruction will be executed before any interrupt is vectored to.

The polling cycle is repeated with each machine cycle, and the values polled are the values that were present at S5P2 of the previous machine cycle. Note that if an interrupt flag is active but not being responded to for one of the above conditions, if the flag is not still 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.

**IE** Address = 0A8H Reset Value = 0X000000B

Bit Addressable

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| EA | — | ET2 | ES | ET1 | EX1 | ET0 | EX0 |

Enable Bit = 1 enables the interrupt.
Enable Bit = 0 disables it.

| BIT | SYMBOL | FUNCTION |
|---|---|---|
| IE.7 | EA | Global disable bit. If EA = 0, all interrupts are disabled. If EA = 1, each interrupt can be individually enabled or disabled by setting or clearing its enable bit. |
| IE.6 | — | Not implemented. Reserved for future use. |
| IE.5 | ET2 | Timer 2 interrupt enable bit. |
| IE.4 | ES | Serial Port interrupt enable bit. |
| IE.3 | ET1 | Timer 1 interrupt enable bit. |
| IE.2 | EX1 | External interrupt 1 enable bit. |
| IE.1 | ET0 | Timer 0 interrupt enable bit. |
| IE.0 | EX0 | External interrupt 0 enable bit. |

SU01522

**Figure 22. Interrupt Enable (IE) Register**

**IP** Address = 0B8H Reset Value = xx000000B

Bit Addressable

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| — | — | PT2 | PS | PT1 | PX1 | PT0 | PX0 |

Priority Bit = 1 assigns higher priority
Priority Bit = 0 assigns lower priority

| BIT | SYMBOL | FUNCTION |
|---|---|---|
| IP.7 | — | Not implemented, reserved for future use. |
| IP.6 | — | Not implemented, reserved for future use. |
| IP.5 | PT2 | Timer 2 interrupt priority bit. |
| IP.4 | PS | Serial Port interrupt priority bit. |
| IP.3 | PT1 | Timer 1 interrupt priority bit. |
| IP.2 | PX1 | External interrupt 1 priority bit. |
| IP.1 | PT0 | Timer 0 interrupt priority bit. |
| IP.0 | PX0 | External interrupt 0 priority bit. |

SU01523

**Figure 23. Interrupt Priority (IP) Register**

**IPH** Address = B7H Reset Value = xx000000B

Bit Addressable

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| — | — | PT2H | PSH | PT1H | PX1H | PT0H | PX0H |

Priority Bit = 1 assigns higher priority
Priority Bit = 0 assigns lower priority

| BIT | SYMBOL | FUNCTION |
|---|---|---|
| IPH.7 | — | Not implemented, reserved for future use. |
| IPH.6 | — | Not implemented, reserved for future use. |
| IPH.5 | PT2H | Timer 2 interrupt priority bit high. |
| IPH.4 | PSH | Serial Port interrupt priority bit high. |
| IPH.3 | PT1H | Timer 1 interrupt priority bit high. |
| IPH.2 | PX1H | External interrupt 1 priority bit high. |
| IPH.1 | PT0H | Timer 0 interrupt priority bit high. |
| IPH.0 | PX0H | External interrupt 0 priority bit high. |

SU01524

**Figure 24. Interrupt Priority HIGH (IPH) Register**

An interrupt will be serviced as long as an interrupt of equal or higher priority is not already being serviced. If an interrupt of equal or higher level priority is being serviced, the new interrupt will wait until it is finished before being serviced. If a lower priority level interrupt is being serviced, it will be stopped and the new interrupt serviced. When the new interrupt is finished, the lower priority level interrupt that was stopped will be completed.

**Table 8. Interrupt Table**

| SOURCE | POLLING PRIORITY | REQUEST BITS | HARDWARE CLEAR? | VECTOR ADDRESS |
|---|---|---|---|---|
| External interrupt 0 | 1 | IE0 | N (L)[1] Y (T)[2] | 03H |
| Timer 0 | 2 | TF0 | Y | 0BH |
| External interrupt 1 | 3 | IE1 | N (L) Y (T) | 13H |
| Timer 1 | 4 | TF1 | Y | 1BH |
| UART | 5 | RI, TI | N | 23H |
| Timer 2 | 6 | TF2, EXF2 | N | 2BH |

**NOTES:**
1. L = Level activated
2. T = Transition activated

## Reduced EMI

All port pins have slew rate controlled outputs. This is to limit noise generated by quickly switching output signals. The slew rate is factory set to approximately 10 ns rise and fall times.

## Reduced EMI Mode

The AO bit (AUXR.0) in the AUXR register when set disables the ALE output.

**AUXR (8EH)**

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

AUXR.0 AO Turns off ALE output.

## Dual DPTR

The dual DPTR structure (see Figure 26) enables a way to specify the address of an external data memory location. There are two 16-bit DPTR registers that address the external memory, and a single bit called DPS = AUXR1/bit0 that allows the program code to switch between them.

- New Register Name: AUXR1#
- SFR Address: A2H
- Reset Value: xxx000x0B

**AUXR1 (A2H)**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| – | – | – | LPEP | WUPD | 0 | – | DPS |

Where:
DPS = AUXR1/bit0 = Switches between DPTR0 and DPTR1.

| Select Reg | DPS |
|---|---|
| DPTR0 | 0 |
| DPTR1 | 1 |

The DPS bit status should be saved by software when switching between DPTR0 and DPTR1.

Note that bit 2 is not writable and is always read as a zero. This allows the DPS bit to be quickly toggled simply by executing an INC DPTR instruction without affecting the WUPD or LPEP bits.

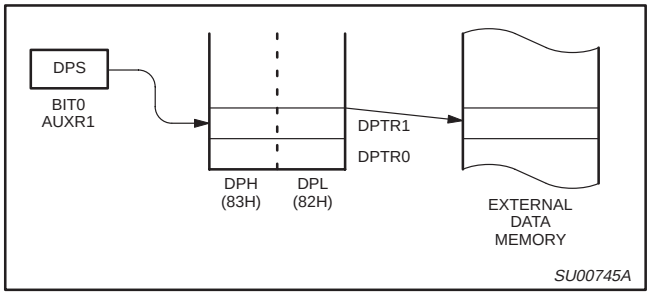



**Figure 26.**

**DPTR Instructions**

The instructions that refer to DPTR refer to the data pointer that is currently selected using the AUXR1/bit 0 register. The six instructions that use the DPTR are as follows:

| | |
|---|---|
| INC DPTR | Increments the data pointer by 1 |
| MOV DPTR, #data16 | Loads the DPTR with a 16-bit constant |
| MOV A, @ A+DPTR | Move code byte relative to DPTR to ACC |
| MOVX A, @ DPTR | Move external RAM (16-bit address) to ACC |
| MOVX @ DPTR , A | Move ACC to external RAM (16-bit address) |
| JMP @ A + DPTR | Jump indirect relative to DPTR |

The data pointer can be accessed on a byte-by-byte basis by specifying the low or high byte in an instruction which accesses the SFRs. See application note AN458 for more details.

## DC ELECTRICAL CHARACTERISTICS

$T_{amb}$ = 0 °C to +70 °C or –40 °C to +85 °C; **$V_{CC}$ = 5 V ±10%**; $V_{SS}$ = 0 V (30/33 MHz max. CPU clock)

| SYMBOL | PARAMETER | TEST CONDITIONS | LIMITS | | | UNIT |
|---|---|---|---|---|---|---|
| | | | MIN | TYP[1] | MAX | |
| $V_{IL}$ | Input low voltage[11] | 4.5 V < $V_{CC}$ < 5.5 V | –0.5 | | 0.2 $V_{CC}$–0.1 | V |
| $V_{IH}$ | Input high voltage (ports 0, 1, 2, 3, $\overline{EA}$) | – | 0.2 $V_{CC}$+0.9 | | $V_{CC}$+0.5 | V |
| $V_{IH1}$ | Input high voltage, XTAL1, RST[11] | – | 0.7 $V_{CC}$ | | $V_{CC}$+0.5 | V |
| $V_{OL}$ | Output low voltage, ports 1, 2, 3 [8] | $V_{CC}$ = 4.5 V; $I_{OL}$ = 1.6 mA[2] | – | | 0.4 | V |
| $V_{OL1}$ | Output low voltage, port 0, ALE, $\overline{PSEN}$ [7, 8] | $V_{CC}$ = 4.5 V; $I_{OL}$ = 3.2 mA[2] | – | | 0.4 | V |
| $V_{OH}$ | Output high voltage, ports 1, 2, 3 [3] | $V_{CC}$ = 4.5 V; $I_{OH}$ = –30 µA | $V_{CC}$ – 0.7 | | – | V |
| $V_{OH1}$ | Output high voltage (port 0 in external bus mode), ALE[9], $\overline{PSEN}$[3] | $V_{CC}$ = 4.5 V; $I_{OH}$ = –3.2 mA | $V_{CC}$ – 0.7 | | – | V |
| $I_{IL}$ | Logical 0 input current, ports 1, 2, 3 | $V_{IN}$ = 0.4 V | –1 | | –50 | µA |
| $I_{TL}$ | Logical 1-to-0 transition current, ports 1, 2, 3[6] | $V_{IN}$ = 2.0 V; See note 4 | – | | –650 | µA |
| $I_{LI}$ | Input leakage current, port 0 | 0.45 < $V_{IN}$ < $V_{CC}$ – 0.3 | – | | ±10 | µA |
| $I_{CC}$ | Power supply current (see Figure 34): | | | | | |
| | Active mode (see Note 5) | | | | | |
| | Idle mode (see Note 5) | | | | | |
| | Power-down mode or clock stopped (see Figure 39 for conditions) | $T_{amb}$ = 0 °C to 70 °C | | 2 | 30 | µA |
| | | $T_{amb}$ = –40 °C to +85 °C | | 3 | 50 | µA |
| $V_{RAM}$ | RAM keep-alive voltage | – | 1.2 | | | V |
| $R_{RST}$ | Internal reset pull-down resistor | – | 40 | | 225 | kΩ |
| $C_{IO}$ | Pin capacitance[10] (except $\overline{EA}$) | – | – | | 15 | pF |

**NOTES:**

1. Typical ratings are not guaranteed. The values listed are at room temperature, 5 V.
2. Capacitive loading on ports 0 and 2 may cause spurious noise to be superimposed on the $V_{OL}$s of ALE and ports 1 and 3. The noise is due to external bus capacitance discharging into the port 0 and port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading > 100 pF), the noise pulse on the ALE pin may exceed 0.8 V. In such cases, it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input. $I_{OL}$ can exceed these conditions provided that no single output sinks more than 5 mA and no more than two outputs exceed the test conditions.
3. Capacitive loading on ports 0 and 2 may cause the $V_{OH}$ on ALE and $\overline{PSEN}$ to momentarily fall below the $V_{CC}$–0.7 specification when the address bits are stabilizing.
4. Pins of ports 1, 2 and 3 source a transition current when they are being externally driven from 1 to 0. The transition current reaches its maximum value when $V_{IN}$ is approximately 2 V.
5. See Figures 36 through 39 for $I_{CC}$ test conditions and Figure 34 for $I_{CC}$ vs. Frequency.
   12-clock mode characteristics:
   Active mode (operating): $I_{CC(MAX)}$ = 1.0 mA + 0.9 mA × FREQ.[MHz]
   Active mode (reset): $I_{CC(MAX)}$ = 7.0 mA + 0.5 mA x FREQ.[MHz]
   Idle mode: $I_{CC(MAX)}$ = 1.0 mA + 0.18 mA × FREQ.[MHz]
6. This value applies to $T_{amb}$ = 0°C to +70°C. For $T_{amb}$ = –40°C to +85°C, $I_{TL}$ = –750 µA.
7. Load capacitance for port 0, ALE, and $\overline{PSEN}$ = 100 pF, load capacitance for all other outputs = 80 pF.
8. Under steady state (non-transient) conditions, $I_{OL}$ must be externally limited as follows:
   Maximum $I_{OL}$ per port pin: 15 mA (*NOTE: This is 85 °C specification.)
   Maximum $I_{OL}$ per 8-bit port: 26 mA
   Maximum total $I_{OL}$ for all outputs: 71 mA
   If $I_{OL}$ exceeds the test condition, $V_{OL}$ may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
9. ALE is tested to $V_{OH1}$, except when ALE is off then $V_{OH}$ is the voltage specification.
10. Pin capacitance is characterized but not tested. Pin capacitance is less than 25 pF. Pin capacitance of ceramic package is less than 15 pF (except $\overline{EA}$ is 25 pF).
11. To improve noise rejection a nominal 100 ns glitch rejection circuitry has been added to the RST pin, and a nominal 15 ns glitch rejection circuitry has been added to the $\overline{INT0}$ and $\overline{INT1}$ pins. Previous devices provided only an inherent 5 ns of glitch rejection.

## AC ELECTRICAL CHARACTERISTICS (12-CLOCK MODE, 2.7 V TO 5.5 V OPERATION)

$T_{amb}$ = 0 °C to +70 °C or –40 °C to +85 °C ; $V_{CC}$ = 2.7 V to 5.5 V, $V_{SS}$ = 0 V[1,2,3,4]

| Symbol | Figure | Parameter | Limits | | 16 MHz Clock | | Unit |
|---|---|---|---|---|---|---|---|
| | | | MIN | MAX | MIN | MAX | |
| $1/t_{CLCL}$ | 31 | Oscillator frequency | 0 | 16 | – | – | MHz |
| $t_{LHLL}$ | 27 | ALE pulse width | $2t_{CLCL}$–10 | – | 115 | – | ns |
| $t_{AVLL}$ | 27 | Address valid to ALE low | $t_{CLCL}$ –15 | – | 47.5 | – | ns |
| $t_{LLAX}$ | 27 | Address hold after ALE low | $t_{CLCL}$ –25 | – | 37.5 | – | ns |
| $t_{LLIV}$ | 27 | ALE low to valid instruction in | – | 4 $t_{CLCL}$ –55 | – | 195 | ns |
| $t_{LLPL}$ | 27 | ALE low to $\overline{PSEN}$ low | $t_{CLCL}$ –15 | – | 47.5 | – | ns |
| $t_{PLPH}$ | 27 | $\overline{PSEN}$ pulse width | 3 $t_{CLCL}$ –15 | – | 172.5 | – | ns |
| $t_{PLIV}$ | 27 | $\overline{PSEN}$ low to valid instruction in | – | 3 $t_{CLCL}$ –55 | – | 132.5 | ns |
| $t_{PXIX}$ | 27 | Input instruction hold after $\overline{PSEN}$ | 0 | – | 0 | – | ns |
| $t_{PXIZ}$ | 27 | Input instruction float after $\overline{PSEN}$ | – | $t_{CLCL}$ –10 | – | 52.5 | ns |
| $t_{AVIV}$ | 27 | Address to valid instruction in | – | 5 $t_{CLCL}$ –50 | – | 262.5 | ns |
| $t_{PLAZ}$ | 27 | $\overline{PSEN}$ low to address float | – | 10 | – | 10 | ns |
| **Data Memory** | | | | | | | |
| $t_{RLRH}$ | 28 | $\overline{RD}$ pulse width | 6 $t_{CLCL}$ –25 | – | 350 | – | ns |
| $t_{WLWH}$ | 29 | $\overline{WR}$ pulse width | 6 $t_{CLCL}$ –25 | – | 350 | – | ns |
| $t_{RLDV}$ | 28 | $\overline{RD}$ low to valid data in | – | 5 $t_{CLCL}$ –50 | – | 262.5 | ns |
| $t_{RHDX}$ | 28 | Data hold after $\overline{RD}$ | 0 | – | 0 | – | ns |
| $t_{RHDZ}$ | 28 | Data float after $\overline{RD}$ | – | 2 $t_{CLCL}$ –20 | – | 105 | ns |
| $t_{LLDV}$ | 28 | ALE low to valid data in | – | 8 $t_{CLCL}$ –55 | – | 445 | ns |
| $t_{AVDV}$ | 28 | Address to valid data in | – | 9 $t_{CLCL}$ –50 | – | 512.5 | ns |
| $t_{LLWL}$ | 28, 29 | ALE low to $\overline{RD}$ or $\overline{WR}$ low | 3 $t_{CLCL}$ –20 | 3 $t_{CLCL}$ +20 | 167.5 | 207.5 | ns |
| $t_{AVWL}$ | 28, 29 | Address valid to $\overline{WR}$ low or $\overline{RD}$ low | 4 $t_{CLCL}$ –20 | – | 230 | – | ns |
| $t_{QVWX}$ | 29 | Data valid to $\overline{WR}$ transition | $t_{CLCL}$ –30 | – | 32.5 | – | ns |
| $t_{WHQX}$ | 29 | Data hold after $\overline{WR}$ | $t_{CLCL}$ –20 | – | 42.5 | – | ns |
| $t_{QVWH}$ | 29 | Data valid to $\overline{WR}$ high | 7 $t_{CLCL}$ –10 | – | 427.5 | – | ns |
| $t_{RLAZ}$ | 28 | $\overline{RD}$ low to address float | – | 0 | – | 0 | ns |
| $t_{WHLH}$ | 28, 29 | $\overline{RD}$ or $\overline{WR}$ high to ALE high | $t_{CLCL}$ –15 | $t_{CLCL}$ +15 | 47.5 | 77.5 | ns |
| **External Clock** | | | | | | | |
| $t_{CHCX}$ | 31 | High time | 0.32 $t_{CLCL}$ | $t_{CLCL} - t_{CLCX}$ | – | – | ns |
| $t_{CLCX}$ | 31 | Low time | 0.32 $t_{CLCL}$ | $t_{CLCL} - t_{CHCX}$ | – | – | ns |
| $t_{CLCH}$ | 31 | Rise time | – | 5 | – | – | ns |
| $t_{CHCL}$ | 31 | Fall time | – | 5 | – | – | ns |
| **Shift register** | | | | | | | |
| $t_{XLXL}$ | 30 | Serial port clock cycle time | 12 $t_{CLCL}$ | – | 750 | – | ns |
| $t_{QVXH}$ | 30 | Output data setup to clock rising edge | 10 $t_{CLCL}$ –25 | – | 600 | – | ns |
| $t_{XHQX}$ | 30 | Output data hold after clock rising edge | 2 $t_{CLCL}$ –15 | – | 110 | – | ns |
| $t_{XHDX}$ | 30 | Input data hold after clock rising edge | 0 | – | 0 | – | ns |
| $t_{XHDV}$ | 30 | Clock rising edge to input data valid | – | 10 $t_{CLCL}$ –133 | – | 492 | ns |

**NOTES:**
1. Parameters are valid over operating temperature range unless otherwise specified.
2. Load capacitance for port 0, ALE, and $\overline{PSEN}$ = 100 pF, load capacitance for all outputs = 80 pF
3. Interfacing the microcontroller to devices with float time up to 45 ns is permitted. This limited bus contention will not cause damage to port 0 drivers.
4. Parts are guaranteed by design to operate down to 0 Hz.

## AC ELECTRICAL CHARACTERISTICS (6-CLOCK MODE, 2.7 V TO 5.5 V OPERATION)

$T_{amb}$ = 0 °C to +70 °C or –40 °C to +85 °C ; $V_{CC}$=2.7 V to 5.5 V, $V_{SS}$ = 0 V[1,2,3,4,5]

| Symbol | Figure | Parameter | Limits | | 16 MHz Clock | | Unit |
|---|---|---|---|---|---|---|---|
| | | | MIN | MAX | MIN | MAX | |
| $1/t_{CLCL}$ | 31 | Oscillator frequency | 0 | 16 | – | – | MHz |
| $t_{LHLL}$ | 27 | ALE pulse width | $t_{CLCL}$–10 | – | 52.5 | – | ns |
| $t_{AVLL}$ | 27 | Address valid to ALE low | 0.5 $t_{CLCL}$ –15 | – | 16.25 | – | ns |
| $t_{LLAX}$ | 27 | Address hold after ALE low | 0.5 $t_{CLCL}$ –25 | – | 6.25 | – | ns |
| $t_{LLIV}$ | 27 | ALE low to valid instruction in | – | 2 $t_{CLCL}$ –55 | – | 70 | ns |
| $t_{LLPL}$ | 27 | ALE low to $\overline{PSEN}$ low | 0.5 $t_{CLCL}$ –15 | – | 16.25 | – | ns |
| $t_{PLPH}$ | 27 | $\overline{PSEN}$ pulse width | 1.5 $t_{CLCL}$ –15 | – | 78.75 | – | ns |
| $t_{PLIV}$ | 27 | $\overline{PSEN}$ low to valid instruction in | – | 1.5 $t_{CLCL}$ –55 | – | 38.75 | ns |
| $t_{PXIX}$ | 27 | Input instruction hold after $\overline{PSEN}$ | 0 | – | 0 | – | ns |
| $t_{PXIZ}$ | 27 | Input instruction float after $\overline{PSEN}$ | – | 0.5 $t_{CLCL}$ –10 | – | 21.25 | ns |
| $t_{AVIV}$ | 27 | Address to valid instruction in | – | 2.5 $t_{CLCL}$ –50 | – | 101.25 | ns |
| $t_{PLAZ}$ | 27 | $\overline{PSEN}$ low to address float | – | 10 | – | 10 | ns |
| **Data Memory** | | | | | | | |
| $t_{RLRH}$ | 28 | $\overline{RD}$ pulse width | 3 $t_{CLCL}$ –25 | – | 162.5 | – | ns |
| $t_{WLWH}$ | 29 | $\overline{WR}$ pulse width | 3 $t_{CLCL}$ –25 | – | 162.5 | – | ns |
| $t_{RLDV}$ | 28 | $\overline{RD}$ low to valid data in | – | 2.5 $t_{CLCL}$ –50 | – | 106.25 | ns |
| $t_{RHDX}$ | 28 | Data hold after $\overline{RD}$ | 0 | – | 0 | – | ns |
| $t_{RHDZ}$ | 28 | Data float after $\overline{RD}$ | – | $t_{CLCL}$ –20 | – | 42.5 | ns |
| $t_{LLDV}$ | 28 | ALE low to valid data in | – | 4 $t_{CLCL}$ –55 | – | 195 | ns |
| $t_{AVDV}$ | 28 | Address to valid data in | – | 4.5 $t_{CLCL}$ –50 | – | 231.25 | ns |
| $t_{LLWL}$ | 28, 29 | ALE low to $\overline{RD}$ or $\overline{WR}$ low | 1.5 $t_{CLCL}$ –20 | 1.5 $t_{CLCL}$ +20 | 73.75 | 113.75 | ns |
| $t_{AVWL}$ | 28, 29 | Address valid to $\overline{WR}$ low or $\overline{RD}$ low | 2 $t_{CLCL}$ –20 | – | 105 | – | ns |
| $t_{QVWX}$ | 29 | Data valid to $\overline{WR}$ transition | 0.5 $t_{CLCL}$ –30 | – | 1.25 | – | ns |
| $t_{WHQX}$ | 29 | Data hold after $\overline{WR}$ | 0.5 $t_{CLCL}$ –20 | – | 11.25 | – | ns |
| $t_{QVWH}$ | 29 | Data valid to $\overline{WR}$ high | 3.5 $t_{CLCL}$ –10 | – | 208.75 | – | ns |
| $t_{RLAZ}$ | 28 | $\overline{RD}$ low to address float | – | 0 | – | 0 | ns |
| $t_{WHLH}$ | 28, 29 | $\overline{RD}$ or $\overline{WR}$ high to ALE high | 0.5 $t_{CLCL}$ –15 | 0.5 $t_{CLCL}$ +15 | 16.25 | 46.25 | ns |
| **External Clock** | | | | | | | |
| $t_{CHCX}$ | 31 | High time | 0.4 $t_{CLCL}$ | $t_{CLCL} - t_{CLCX}$ | – | – | ns |
| $t_{CLCX}$ | 31 | Low time | 0.4 $t_{CLCL}$ | $t_{CLCL} - t_{CHCX}$ | – | – | ns |
| $t_{CLCH}$ | 31 | Rise time | – | 5 | – | – | ns |
| $t_{CHCL}$ | 31 | Fall time | – | 5 | – | – | ns |
| **Shift register** | | | | | | | |
| $t_{XLXL}$ | 30 | Serial port clock cycle time | 6 $t_{CLCL}$ | – | 375 | – | ns |
| $t_{QVXH}$ | 30 | Output data setup to clock rising edge | 5 $t_{CLCL}$ –25 | – | 287.5 | – | ns |
| $t_{XHQX}$ | 30 | Output data hold after clock rising edge | $t_{CLCL}$ –15 | – | 47.5 | – | ns |
| $t_{XHDX}$ | 30 | Input data hold after clock rising edge | 0 | – | 0 | – | ns |
| $t_{XHDV}$ | 30 | Clock rising edge to input data valid | – | 5 $t_{CLCL}$ –133 | – | 179.5 | ns |

**NOTES:**
1. Parameters are valid over operating temperature range unless otherwise specified.
2. Load capacitance for port 0, ALE, and $\overline{PSEN}$=100 pF, load capacitance for all outputs = 80 pF
3. Interfacing the microcontroller to devices with float time up to 45ns is permitted. This limited bus contention will not cause damage to port 0 drivers.
4. Parts are guaranteed by design to operate down to 0 Hz.
5. Data shown in the table are the best mathematical models for the set of measured values obtained in tests. If a particular parameter calculated at a customer specified frequency has a negative value, it should be considered equal to zero.

## EXPLANATION OF THE AC SYMBOLS

Each timing symbol has five characters. The first character is always 't' (= time). The other characters, depending on their positions, indicate the name of a signal or the logical status of that signal. The designations are:

A – Address
C – Clock
D – Input data
H – Logic level high
I – Instruction (program memory contents)
L – Logic level low, or ALE
P – $\overline{\text{PSEN}}$
Q – Output data
R – $\overline{\text{RD}}$ signal
t – Time
V – Valid
W – $\overline{\text{WR}}$ signal
X – No longer a valid logic level
Z – Float

**Examples:** $t_{AVLL}$ = Time for address valid to ALE low.
$t_{LLPL}$ =Time for ALE low to $\overline{\text{PSEN}}$ low.

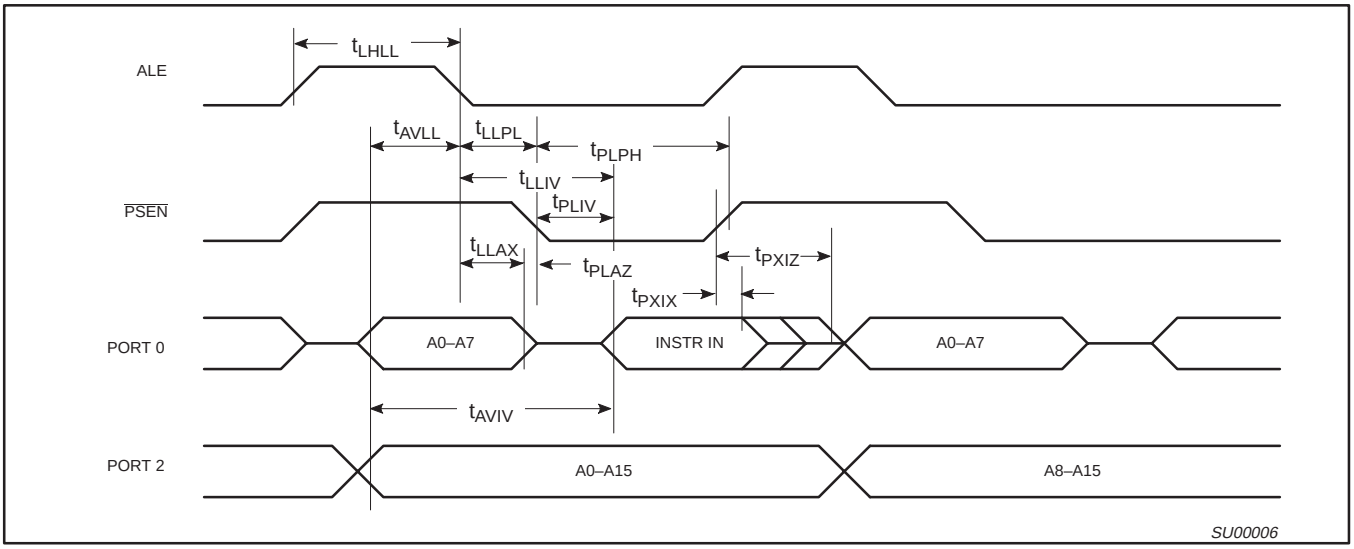



**Figure 27. External Program Memory Read Cycle**

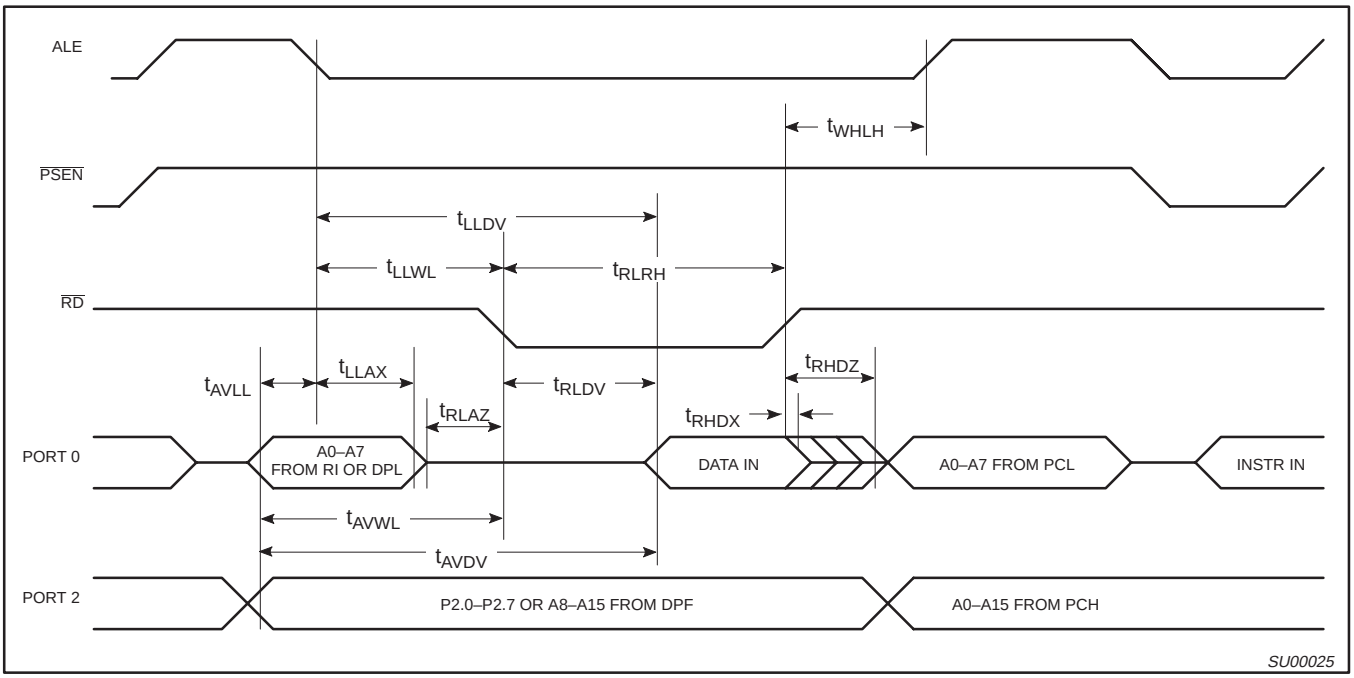



**Figure 28. External Data Memory Read Cycle**

| ADDRESS | CONTENT | BIT(S) | COMMENT |
|---|---|---|---|
| 0000H to 1FFFH | DATA | 7:0 | User ROM Data |
| 2000H to 203FH | KEY | 7:0 | ROM Encryption Key |
| 2040H | SEC | 0 | ROM Security Bit 1 |
| 2040H | SEC | 1 | ROM Security Bit 2 |

**Security Bit 1:** When programmed, this bit has two effects on masked ROM parts:
1. External MOVC is disabled, and
2. $\overline{EA}$ is latched on Reset.

**Security Bit 2:** When programmed, this bit inhibits Verify User ROM.

**NOTE:** Security Bit 2 cannot be enabled unless Security Bit 1 is enabled.

If the ROM Code file does not include the options, the following information must be included with the ROM code.

For each of the following, check the appropriate box, and send to Philips along with the code:

Security Bit #1: ☐ Enabled ☐ Disabled

Security Bit #2: ☐ Enabled ☐ Disabled

Encryption: ☐ No ☐ Yes If Yes, must send key file.

## 80C58X2 ROM CODE SUBMISSION

When submitting a ROM code for the 80C58X2, the following must be specified:

1. 32 kbyte user ROM data
2. 64 byte ROM encryption key
3. ROM security bits.

| ADDRESS | CONTENT | BIT(S) | COMMENT |
|---|---|---|---|
| 0000H to 7FFFH | DATA | 7:0 | User ROM Data |
| 8000H to 803FH | KEY | 7:0 | ROM Encryption Key<br>FFH = no encryption |
| 8040H | SEC | 0 | ROM Security Bit 1<br>0 = enable security<br>1 = disable security |
| 8040H | SEC | 1 | ROM Security Bit 2<br>0 = enable security<br>1 = disable security |

**Security Bit 1:** When programmed, this bit has two effects on masked ROM parts:

1. External MOVC is disabled, and
2. $\overline{EA}$ is latched on Reset.

**Security Bit 2:** When programmed, this bit inhibits Verify User ROM.

**NOTE:** Security Bit 2 cannot be enabled unless Security Bit 1 is enabled.

If the ROM Code file does not include the options, the following information must be included with the ROM code.

For each of the following, check the appropriate box, and send to Philips along with the code:

Security Bit #1: ☐ Enabled ☐ Disabled

Security Bit #2: ☐ Enabled ☐ Disabled

Encryption: ☐ No ☐ Yes If Yes, must send key file.

## REVISION HISTORY

| Rev | Date | Description |
|---|---|---|
| _6 | 20030124 | **Product data (9397 750 10995); ECN 853-2337 29260 of 06 December 2002**<br>Modifications:<br>• Added TSSOP38 package details |
| _5 | 20020912 | **Product data (9397 750 10361); ECN 853-2337 28906 of 12 September 2002** |
| _4 | 20020612 | **Product data (9397 750 09969); ECN 853-2337 28427 of 12 June 2002** |
| _3 | 20020422 | **Product data (9397 750 09779); ECN 853-2337 28059 of 22 April 2002** |
| _2 | 20020219 | **Preliminary data (9397 750 09467)** |
| _1 | 20010924 | **Preliminary data (9397 750 08895); initial release** |