# W77E532/W77E532A DATA SHEET

winbond

# 8-BIT MICROCONTROLLER

***Table of Contents-***


1. GENERAL DESCRIPTION ..... 2
2. FEATURES ..... 2
3. PIN CONFIGURATIONS ..... 3
4. PIN DESCRIPTION ..... 4
5. FUNCTIONAL DESCRIPTION ..... 6
6. MEMORY ORGANIZATION ..... 8
7. INSTRUCTION ..... 32
8. INSTRUCTION TIMING ..... 33
9. POWER MANAGEMENT ..... 41
10. INTERRUPTS ..... 46
11. PROGRAMMABLE TIMERS/COUNTERS ..... 49
12. SERIAL PORT ..... 58
13. TIMED ACCESS PROTECTION ..... 65
14. H/W REBOOT MODE (BOOT FROM 4K BYTES OF LDFLASH ) ..... 67
15. IN-SYSTEM PROGRAMMING ..... 68
    - 15.1 The Loader Program Locates at LDFLASH Memory ..... 68
    - 15.2 The Loader Program Locates at APFLASH Memory ..... 68
16. H/W WRITER MODE ..... 68
17. SECURITY BITS ..... 69
18. ELECTRICAL CHARACTERISTICS ..... 70
    - 18.1 Absolute Maximum Ratings ..... 70
    - 18.2 DC Characteristics ..... 70
    - 18.3 AC Characteristics ..... 71
        - 18.3.1 AC Specification ..... 72
        - 18.3.2 MOVX Characteristics Using Strech Memory Cycle ..... 73
        - 18.3.3 Program Memory Read Cycle ..... 74
        - 18.3.4 Data Memory Read Cycle ..... 75
        - 18.3.5 Data Memory Write Cycle ..... 75
19. TYPICAL APPLICATION CIRCUITS ..... 76
    - 19.1 Crystal connections ..... 76
    - 19.2 Expanded External Data Memory and Oscillator ..... 77
20. PACKAGE DIMENSIONS ..... 78
    - 20.1 40-pin DIP ..... 78
    - 20.2 44-pin PLCC ..... 78
    - 20.3 44-pin QFP ..... 79
21. APPLICATION NOTE ..... 80
22. REVISION HISTORY ..... 85

## 1. GENERAL DESCRIPTION

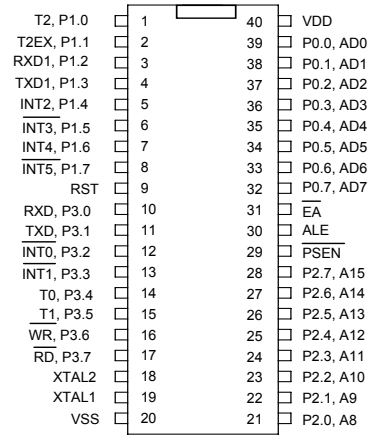
The W77E532A is a fast 8051 compatible microcontroller with a redesigned processor core without wasted clock and memory cycles. As a result, it executes every 8051 instruction faster than the original 8051 for the same crystal speed. Typically, the instruction executing time of W77E532A is 1.5 to 3 times faster then that of traditional 8051, depending on the type of instruction. In general, the overall performance is about 2.5 times better than the original for the same crystal speed. Giving the same throughput with lower clock speed, power consumption has been improved. Consequently, the W77E532A is a fully static CMOS design; it can also be operated at a lower crystal clock. The W77E532A contains In-System Programmable(ISP) 128 KB bank-addressed Flash EPROM; 4KB auxiliary Flash EPROM for loader program; operating voltage from 4.5V to 5.5V; on-chip 1 KB MOVX SRAM; three power saving modes.

## 2. FEATURES

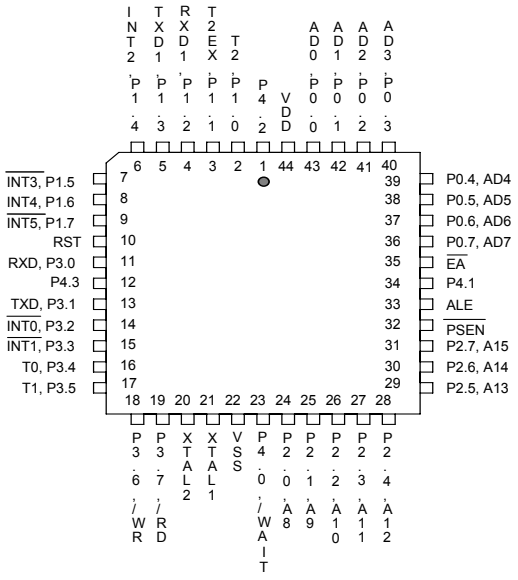
- 8-bit CMOS microcontroller
- High speed architecture of 4 clocks/machine cycle runs up to 40 MHz
- Pin compatible with standard 80C52
- Instruction-set compatible with MCS-51
- Four 8-bit I/O Ports; Port 0 has internal pull-up resisters enabled by software
- One extra 4-bit I/O port, chip select and Wait State control signal (available on 44-pin PLCC/QFP package)
- Three 16-bit Timers
- 12 interrupt sources with two levels of priority
- On-chip oscillator and clock circuitry
- Two enhanced full duplex serial ports
- Dual 64KB In-System Programmable Flash EPROM banks (APFLASH0 and APFLASH1)
- 4KB Auxiliary Flash EPROM for loader program (LDFLASH)
- 64 Bytes Non-Volatile Memory for critical date storage.
- 256 bytes scratch-pad RAM
- 1 KB on-chip SRAM for MOVX instruction
- Programmable Watchdog Timer
- Software Reset
- Dual 16-bit Data Pointers
- Software programmable access cycle to external RAM/peripherals
- Packages:
  - Lead Free(RoHS) DIP 40: W77E532A40DL
  - Lead Free(RoHS) PLCC 44: W77E532A40PL
  - Lead Free(RoHS) QFP 44: W77E532A40FL

## 3. PIN CONFIGURATIONS

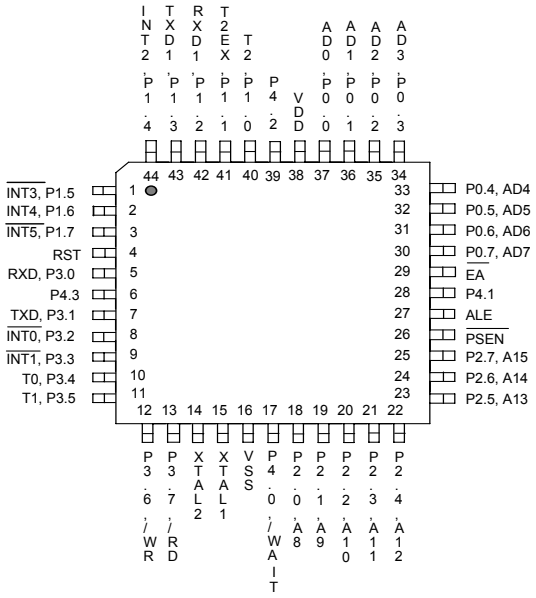
40-Pin DIP




44-Pin PLCC




44-Pin QFP



## 4. PIN DESCRIPTION

| SYMBOL | TYPE | DESCRIPTIONS |
|---|---|---|
| $\overline{EA}$ | I | **EXTERNAL ACCESS ENABLE:** This pin forces the processor to execute out of external ROM. It should be kept high to access internal ROM. The ROM address and data will not be present on the bus if $\overline{EA}$ pin is high. |
| $\overline{PSEN}$ | O | **PROGRAM STORE ENABLE:** $\overline{PSEN}$ enables the external ROM data onto the Port 0 address/data bus during fetch and MOVC operations. When internal ROM access is performed, no $\overline{PSEN}$ strobe signal outputs from this pin. |
| ALE | O | **ADDRESS LATCH ENABLE:** ALE is used to enable the address latch that separates the address from the data on Port 0. |
| RST | I | **RESET:** A high on this pin for two machine cycles while the oscillator is running resets the device. |
| XTAL1 | I | **CRYSTAL1:** This is the crystal oscillator input. This pin may be driven by an external clock. |
| XTAL2 | O | **CRYSTAL2:** This is the crystal oscillator output. It is the inversion of XTAL1. |
| VSS | I | **GROUND:** Ground potential |
| VDD | I | **POWER SUPPLY:** Supply voltage for operation. |
| P0.0 – P0.7 | I/O | **PORT 0:** Port 0 is an open-drain bi-directional I/O port. This port also provides a multiplexed low order address/data bus during accesses to external memory.<br>Port 0 has internal pull-up resisters enabled by software. |
| P1.0 – P1.7 | I/O | **PORT 1:** Port 1 is a bi-directional I/O port with internal pull-ups. The bits have alternate functions which are described below:<br>T2(P1.0): Timer/Counter 2 external count input<br>T2EX(P1.1): Timer/Counter 2 Reload/Capture/Direction control<br>RXD1(P1.2): Serial port 2 RXD<br>TXD1(P1.3): Serial port 2 TXD<br>INT2(P1.4): External Interrupt 2<br>$\overline{INT3}$ (P1.5): External Interrupt 3<br>INT4(P1.6): External Interrupt 4<br>$\overline{INT5}$ (P1.7): External Interrupt 5 |
| P2.0 – P2.7 | I/O | **PORT 2:** Port 2 is a bi-directional I/O port with internal pull-ups. This port also provides the upper address bits for accesses to external memory. |

Pin Description, continued

| SYMBOL | TYPE | DESCRIPTIONS |
|---|---|---|
| P3.0 – P3.7 | I/O | **PORT 3:** Port 3 is a bi-directional I/O port with internal pull-ups. All bits have alternate functions, which are described below:<br>RXD(P3.0) : Serial Port 0 input<br>TXD(P3.1) : Serial Port 0 output<br>$\overline{INT0}$ (P3.2) : External Interrupt 0<br>$\overline{INT1}$ (P3.3) : External Interrupt 1<br>T0(P3.4) : Timer 0 External Input<br>T1(P3.5) : Timer 1 External Input<br>$\overline{WR}$ (P3.6) : External Data Memory Write Strobe<br>$\overline{RD}$ (P3.7) : External Data Memory Read Strobe |
| P4.0 – P4.3 | I/O | **PORT 4:** Port 4 is a 4-bit bi-directional I/O port. The P4.0 also provides the alternate function $\overline{WAIT}$ which is the wait state control signal. The P4.3 also provides the alternate function $\overline{REBOOT}$ which is H/W reboot from LD flash. |

*** Note: TYPE** I : input, O: output, I/O: bi-directional.

## 5. FUNCTIONAL DESCRIPTION

The W77E532A is 8052 pin compatible and instruction set compatible. It includes the resources of the standard 8052 such as four 8-bit I/O Ports, three 16-bit timer/counters, full duplex serial port and interrupt sources.

The W77E532A features a faster running and better performance 8-bit CPU with a redesigned core processor without wasted clock and memory cycles. it improves the performance not just by running at high frequency but also by reducing the machine cycle duration from the standard 8052 period of twelve clocks to four clock cycles for the majority of instructions. This improves performance by an average of 1.5 to 3 times. The W77E532A also provides dual Data Pointers (DPTRs) to speed up block data memory transfers. It can also adjust the duration of the MOVX instruction (access to off-chip data memory) between two machine cycles and nine machine cycles. This flexibility allows the W77E532A to work efficiently with both fast and slow RAMs and peripheral devices. In addition, the W77E532A contains on-chip 1KB MOVX SRAM, the address of which is between 0000H and 03FFH. It only can be accessed by MOVX instruction; this on-chip SRAM is optional under software control.

The W77E532A is an 8052 compatible device that gives the user the features of the original 8052 device, but with improved speed and power consumption characteristics. It has the same instruction set as the 8051 family, with one addition: DEC DPTR (op-code A5H, the DPTR is decreased by 1). While the original 8051 family was designed to operate at 12 clock periods per machine cycle, the W77E532A operates at a much reduced clock rate of only 4 clock periods per machine cycle. This naturally speeds up the execution of instructions. Consequently, the W77E532A can run at a higher speed as compared to the original 8052, even if the same crystal is used. Since the W77E532A is a fully static CMOS design, it can also be operated at a lower crystal clock, giving the same throughput in terms of instruction execution, yet reducing the power consumption.

The 4 clocks per machine cycle feature in the W77E532A is responsible for a three-fold increase in execution speed. The W77E532A has all the standard features of the 8052, and has a few extra peripherals and features as well.

### I/O Ports

The W77E532A has four 8-bit ports and one extra 4-bit port. Port 0 can be used as an Address/Data bus when external program is running or external memory/device is accessed by MOVC or MOVX instruction. In these cases, it has strong pull-ups and pull-downs, and does not need any external pull-ups. Otherwise it can be used as a general I/O port with open-drain circuit. Port 2 is used chiefly as the upper 8-bits of the Address bus when port 0 is used as an address/data bus. It also has strong pull-ups and pull-downs when it serves as an address bus. Port 1 and 3 act as I/O ports with alternate functions. Port 4 is only available on 44-pin PLCC/QFP package type. It serves as a general purpose I/O port as Port 1 and Port 3. The P4.0 has an alternate function $\overline{WR}$/$\overline{RL2}$ which is the wait state control signal. When wait state control signal is enabled, P4.0 is input only.

### Serial I/O

The W77E532A has two enhanced serial ports that are functionally similar to the serial port of the original 8052 family. However the serial ports on the W77E532A can operate in different modes in order to obtain timing similarity as well. **Note that the serial port 0 can use Timer 1 or 2 as baud rate generator, but the serial port 1 can only use Timer 1 as baud rate generator**. The serial ports have the enhanced features of Automatic Address recognition and Frame Error detection.

### Timers

The W77E532A has three 16-bit timers that are functionally similar to the timers of the 8052 family. When used as timers, they can be set to run at either 4 clocks or 12 clocks per count, thus providing the user with the option of operating in a mode that emulates the timing of the original 8052. The W77E532A has an additional feature, the watchdog timer. This timer is used as a System Monitor or as a very long time period timer.

### Interrupts

The Interrupt structure in the W77E532A is slightly different from that of the standard 8052. Due to the presence of additional features and peripherals, the number of interrupt sources and vectors has been increased. The W77E532A provides 12 interrupt resources with two priority level, including six external interrupt sources, timer interrupts, serial I/O interrupts.

### Data Pointers

The original 8052 had only one 16-bit Data Pointer (DPL, DPH). In the W77E532A, there is an additional 16-bit Data Pointer (DPL1, DPH1). This new Data Pointer uses two SFR locations which were unused in the original 8052. In addition there is an added instruction, DEC DPTR (op-code A5H), which helps in improving programming flexibility for the user.

### Power Management

Like the standard 80C52, the W77E532A also has IDLE and POWER DOWN modes of operation. The W77E532A provides a new Economy mode which allow user to switch the internal clock rate divided by either 4, 64 or 1024. In the IDLE mode, the clock to the CPU core is stopped while the timers, serial ports and interrupts clock continue to operate. In the POWER DOWN mode, all the clock are stopped and the chip operation is completely stopped. This is the lowest power consumption state.

### On-chip Data SRAM

The W77E532A has 1K Bytes of data space SRAM which is read/write accessible and is memory mapped. This on-chip MOVX SRAM is reached by the MOVX instruction. It is not used for executable program memory. There is no conflict or overlap among the 256 bytes Scratchpad RAM and the 1K Bytes MOVX SRAM as they use different addressing modes and separate instructions. The on-chip MOVX SRAM is enabled by setting the DME0 bit in the PMR register. After a reset, the DME0 bit is cleared such that the on-chip MOVX SRAM is disabled, and all data memory spaces 0000H – FFFFH access to the external memory.

## 6. MEMORY ORGANIZATION

The W77E532A separates the memory into two separate sections, the Program Memory and the Data Memory. The Program Memory is used to store the instruction op-codes, while the Data Memory is used to store data or for memory mapped devices.

### Program Memory

The Program Memory on the standard 8052 can only be addressed to 64 Kbytes long. By invoking the banking methodology, W77E532A can extend to two 64KB flash EPROM banks, APFLASH0 and APFLASH1. There are on-chip ROM banks which can be used similarly to that of the 8052. All instructions are fetched for execution from this memory area. The MOVC instruction can also access this memory region. There is an auxiliary 4KB Flash EPROM bank (LDFLASH) resided user loader program for In-System Programming (ISP). Both APFLASHs allow serial or parallel download according to user loader program in LDFLASH.

### Data Memory

The W77E532A can access up to 64Kbytes of external Data Memory. This memory region is accessed by the MOVX instructions. Unlike the 8051 derivatives, the W77E532A contains on-chip 1K bytes MOVX SRAM of Data Memory, which can only be accessed by MOVX instructions. These 1K bytes of SRAM are between address 0000H and 03FFH. Access to the on-chip MOVX SRAM is optional under software control. When enabled by software, any MOVX instruction that uses this area will go to the on-chip RAM. MOVX addresses greater than 03FFH automatically go to external memory through Port 0 and 2. When disabled, the 1KB memory area is transparent to the system memory map. Any MOVX directed to the space between 0000H and FFFFH goes to the expanded bus on Port 0 and 2. This is the default condition. In addition, the W77E532A has the standard 256 bytes of on-chip Scratchpad RAM. This can be accessed either by direct addressing or by indirect addressing. There are also some Special Function Registers (SFRs), which can only be accessed by direct addressing. Since the Scratchpad RAM is only 256 bytes, it can be used only when data contents are small. In the event that larger data contents are present, two selections can be used. One is on-chip MOVX SRAM , the other is the external Data Memory. The on-chip MOVX SRAM can only be accessed by a MOVX instruction, the same as that for external Data Memory. However, the on-chip RAM has the fastest access times.

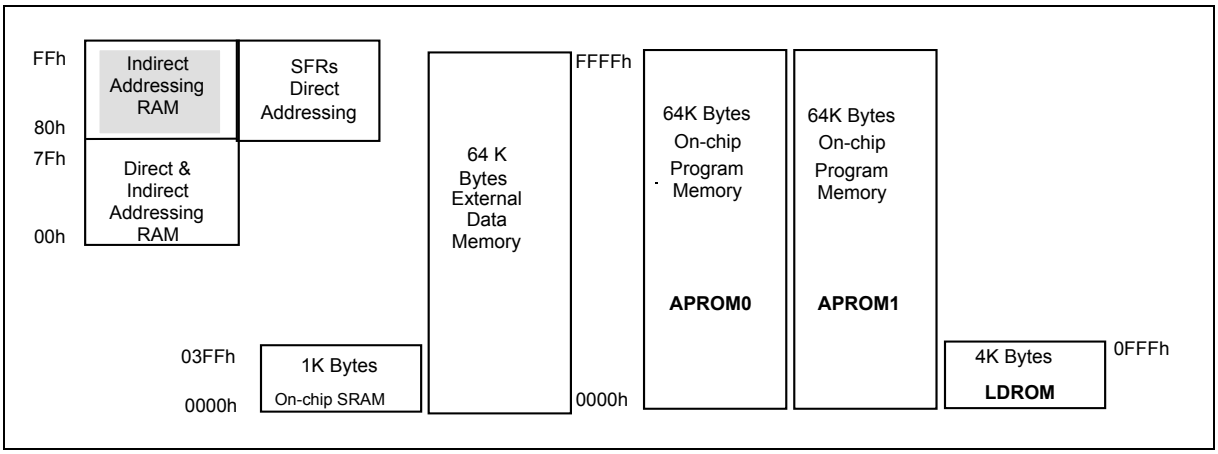



Figure 1. Memory Map

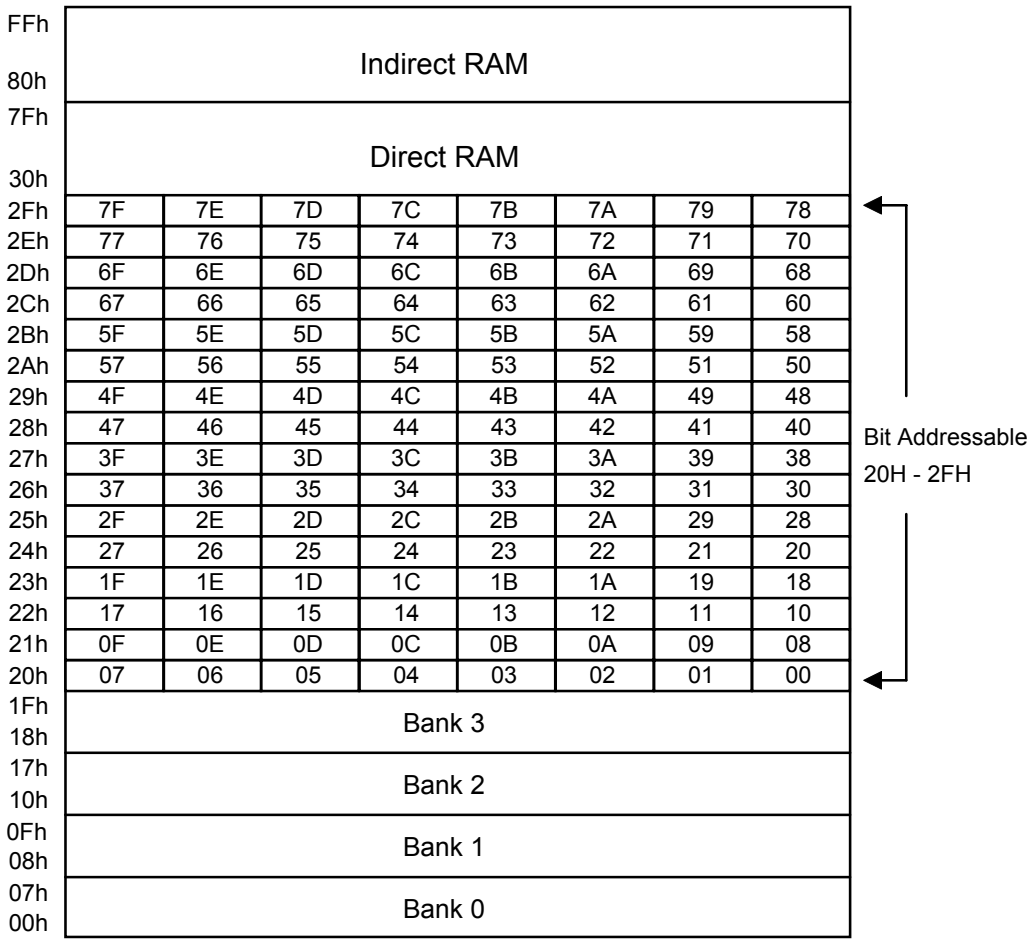



Figure 2. Scratchpad RAM/Register Addressing

### Special Function Registers

The W77E532A uses Special Function Registers (SFRs) to control and monitor peripherals and their Modes.

The SFRs reside in the register locations 80-FFh and are accessed by direct addressing only. Some of the SFRs are bit addressable. This is very useful in cases where one wishes to modify a particular bit without changing the others. The SFRs that are bit addressable are those whose addresses end in 0 or 8. The W77E532A contains all the SFRs present in the standard 8052. However, some additional SFRs have been added. In some cases unused bits in the original 8052 have been given new functions. The list of SFRs is as follows. The table is condensed with eight locations per row. Empty locations indicate that there are no registers at these addresses. When a bit or register is not implemented, it will read high.

**Table 1. Special Function Register Location Table**

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| F8 | EIP | | | | | | | |
| F0 | B | | | | | | | |
| E8 | EIE | | | | | | | |
| E0 | ACC | | | | | | | |
| D8 | WDCON | | | | | | | |
| D0 | PSW | | | | | | | |
| C8 | T2CON | T2MOD | RCAP2L | RCAP2H | TL2 | TH2 | | |
| C0 | SCON1 | SBUF1 | WSCON | | PMR | STATUS | | TA |
| B8 | IP | SADEN | SADEN1 | | | | | |
| B0 | P3 | | | | | | | |
| A8 | IE | SADDR | SADDR1 | ROMCON | SFRAL | SFRAH | SFDFD | SFRCN |
| A0 | P2 | | P4CSIN | | | P4 | | |
| 98 | SCON0 | SBUF | P42AL | P42AH | P43AL | P43AH | | CHPCON |
| 90 | P1 | EXIF | P4CONA | P4CONB | P40AL | P40AH | P41AL | P41AH |
| 88 | TCON | TMOD | TL0 | TL1 | TH0 | TH1 | CKCON | |
| 80 | P0 | SP | DPL | DPH | DPL1 | DPH1 | DPS | PCON |

**Note:** The SFRs in the column with dark borders are bit-addressable.

A brief description of the SFRs now follows.

**Port 0**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | P0.7 | P0.6 | P0.5 | P0.4 | P0.3 | P0.2 | P0.1 | P0.0 |

Mnemonic: P0 Address: 80h

Port 0 is an open-drain bi-directional I/O port. This port also provides a multiplexed low order address/data bus during accesses to external memory. Besides, it has internal pull-up resisters enabled by setting P0UP of P4CSIN (A2H) to high.

**Stack Pointer**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SP.7 | SP.6 | SP.5 | SP.4 | SP.3 | SP.2 | SP.1 | SP.0 |

Mnemonic: SP Address: 81h

The Stack Pointer stores the Scratchpad RAM address where the stack begins. In other words, it always points to the top of the stack.

**Data Pointer Low**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DPL.7 | DPL.6 | DPL.5 | DPL.4 | DPL.3 | DPL.2 | DPL.1 | DPL.0 |

Mnemonic: DPL Address: 82h

This is the low byte of the standard 8052 16-bit data pointer.

**Data Pointer High**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DPH.7 | DPH.6 | DPH.5 | DPH.4 | DPH.3 | DPH.2 | DPH.1 | DPH.0 |

Mnemonic: DPH Address: 83h

This is the high byte of the standard 8052 16-bit data pointer.

**Data Pointer Low1**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DPL1.7 | DPL1.6 | DPL1.5 | DPL1.4 | DPL1.3 | DPL1.2 | DPL1.1 | DPL1.0 |

Mnemonic: DPL1 Address: 84h

This is the low byte of the new additional 16-bit data pointer that has been added to the W77E532A. The user can switch between DPL, DPH and DPL1, DPH1 simply by setting register DPS = 1. The instructions that use DPTR will now access DPL1 and DPH1 in place of DPL and DPH. If they are not required they can be used as conventional register locations by the user.

**Data Pointer High1**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DPH1.7 | DPH1.6 | DPH1.5 | DPH1.4 | DPH1.3 | DPH1.2 | DPH1.1 | DPH1.0 |

Mnemonic: DPH1 Address: 85h

This is the high byte of the new additional 16-bit data pointer that has been added to the W77E532A. The user can switch between DPL, DPH and DPL1, DPH1 simply by setting register DPS = 1. The instructions that use DPTR will now access DPL1 and DPH1 in place of DPL and DPH. If they are not required they can be used as conventional register locations by the user.

**Data Pointer Select**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | - | - | - | DPS.0 |

Mnemonic: DPS Address: 86h

DPS.0: This bit is used to select either the DPL, DPH pair or the DPL1, DPH1 pair as the active Data Pointer. When set to 1, DPL1, DPH1 will be selected, otherwise DPL, DPH will be selected.

DPS.1-7: These bits are reserved, but will read 0.

**Power Control**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SM0D | SMOD0 | - | - | GF1 | GF0 | PD | IDL |

Mnemonic: PCON Address: 87h

SMOD : This bit doubles the serial port baud rate in mode 1, 2, and 3 when set to 1.

SMOD0: Framing Error Detection Enable: When SMOD0 is set to 1, then SCON.7(SCON1.7) indicates a Frame Error and acts as the FE(FE_1) flag. When SMOD0 is 0, then SCON.7(SCON1.7) acts as per the standard 8052 function.

GF1-0: These two bits are general purpose user flags.

PD: Setting this bit causes the W77E532A to go into the POWER DOWN mode. In this mode all the
clocks are stopped and program execution is frozen.

IDL: Setting this bit causes the W77E532A to go into the IDLE mode. In this mode the clocks to the CPU are stopped, so program execution is frozen. But the clock to the serial, timer and interrupt blocks is not stopped, and these blocks continue operating.

**Timer Control**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TF1 | TR1 | TF0 | TR0 | IE1 | IT1 | IE0 | IT0 |

Mnemonic: TCON Address: 88h

TF1: Timer 1 overflow flag: This bit is set when Timer 1 overflows. It is cleared automatically when the program does a timer 1 interrupt service routine. Software can also set or clear this bit.

TR1: Timer 1 run control: This bit is set or cleared by software to turn timer/counter on or off.

TF0: Timer 0 overflow flag: This bit is set when Timer 0 overflows. It is cleared automatically when the program does a timer 0 interrupt service routine. Software can also set or clear this bit.

TR0: Timer 0 run control: This bit is set or cleared by software to turn timer/counter on or off.

IE1: Interrupt 1 edge detect: Set by hardware when an edge/level is detected on $\overline{INT1}$. This bit is cleared by hardware when the service routine is vectored to only if the interrupt was edge triggered. Otherwise it follows the pin.

IT1: Interrupt 1 type control: Set/cleared by software to specify falling edge/ low level triggered external inputs.

IE0: Interrupt 0 edge detect: Set by hardware when an edge/level is detected on $\overline{INT0}$. This bit is cleared by hardware when the service routine is vectored to only if the interrupt was edge triggered. Otherwise it follows the pin.

IT0: Interrupt 0 type control: Set/cleared by software to specify falling edge/ low level triggered external inputs.

**Timer Mode Control**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | GATE | C / $\overline{T}$ | M1 | M0 | GATE | C / $\overline{T}$ | M1 | M0 |
| | | | TIMER1 | | | | TIMER0 | |

Mnemonic: TMOD Address: 89h

GATE: Gating control: When this bit is set, Timer/counter x is enabled only while $\overline{INTx}$ pin is high and TRx control bit is set. When cleared, Timer x is enabled whenever TRx control bit is set.

C / $\overline{T}$: Timer or Counter Select: When cleared, the timer is incremented by internal clocks. When set , the timer counts high-to-low edges of the Tx pin.

**M1, M0: Mode Select bits:**

| M1 | M0 | MODE |
|---|---|---|
| 0 | 0 | Mode 0: 8-bits with 5-bit prescale. |
| 0 | 1 | Mode 1: 16-bits, no prescale. |
| 1 | 0 | Mode 2: 8-bits with auto-reload from THx |
| 1 | 1 | Mode 3: (Timer 0) TL0 is an 8-bit timer/counter controlled by the standard Timer 0 control bits. TH0 is a 8-bit timer only controlled by Timer 1 control bits. (Timer 1) Timer/counter is stopped. |

**Timer 0 LSB**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TL0.7 | TL0.6 | TL0.5 | TL0.4 | TL0.3 | TL0.2 | TL0.1 | TL0.0 |

Mnemonic: TL0 Address: 8Ah

TL0.7–0: Timer 0 LSB

**Timer 1 LSB**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TL1.7 | TL1.6 | TL1.5 | TL1.4 | TL1.3 | TL1.2 | TL1.1 | TL1.0 |

Mnemonic: TL1 Address: 8Bh

TL1.7–0: Timer 1 LSB

**Timer 0 MSB**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TH0.7 | TH0.6 | TH0.5 | TH0.4 | TH0.3 | TH0.2 | TH0.1 | TH0.0 |

Mnemonic: TH0 Address: 8Ch

TH0.7–0: Timer 0 MSB

**Timer 1 MSB**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TH1.7 | TH1.6 | TH1.5 | TH1.4 | TH1.3 | TH1.2 | TH1.1 | TH1.0 |

Mnemonic: TH1 Address: 8Dh

TH1.7–0: Timer 1 MSB

**Clock Control**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | WD1 | WD0 | T2M | T1M | T0M | MD2 | MD1 | MD0 |

Mnemonic: CKCON Address: 8Eh

WD1–0: Watchdog timer mode select bits: These bits determine the time-out period for the watchdog timer. In all four time-out options the reset time-out is 512 clocks more than the interrupt time-out period.

| WD1 | WD0 | INTERRUPT TIME-OUT | RESET TIME-OUT |
|---|---|---|---|
| 0 | 0 | $2^{17}$ | $2^{17}$ + 512 |
| 0 | 1 | $2^{20}$ | $2^{20}$ + 512 |
| 1 | 0 | $2^{23}$ | $2^{23}$ + 512 |
| 1 | 1 | $2^{26}$ | $2^{26}$ + 512 |

T2M: Timer 2 clock select: When T2M is set to 1, timer 2 uses a divide by 4 clock, and when set to 0 it uses a divide by 12 clock.

T1M: Timer 1 clock select: When T1M is set to 1, timer 1 uses a divide by 4 clock, and when set to 0 it uses a divide by 12 clock.

T0M: Timer 0 clock select: When T0M is set to 1, timer 0 uses a divide by 4 clock, and when set to 0 it uses a divide by 12 clock.

MD2–0: Stretch MOVX select bits: These three bits are used to select the stretch value for the MOVX instruction. Using a variable MOVX length enables the user to access slower external memory devices or peripherals without the need for external circuits. The $\overline{RD}$ or $\overline{WR}$ strobe will be stretched by the selected interval. When accessing the on-chip SRAM, the MOVX instruction is always in 2 machine cycles regardless of the stretch setting. By default, the stretch has value of 1. If the user needs faster accessing, then a stretch value of 0 should be selected.

| MD2 | MD1 | MD0 | Stretch value | MOVX duration |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 2 machine cycles |
| 0 | 0 | 1 | 1 | 3 machine cycles *(Default)* |
| 0 | 1 | 0 | 2 | 4 machine cycles |
| 0 | 1 | 1 | 3 | 5 machine cycles |
| 1 | 0 | 0 | 4 | 6 machine cycles |
| 1 | 0 | 1 | 5 | 7 machine cycles |
| 1 | 1 | 0 | 6 | 8 machine cycles |
| 1 | 1 | 1 | 7 | 9 machine cycles |

**Port 1**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | P1.7 | P1.6 | P1.5 | P1.4 | P1.3 | P1.2 | P1.1 | P1.0 |

Mnemonic: P1 Address: 90h

P1.7–0: General purpose I/O port. Most instructions will read the port pins in case of a port read access, however in case of read-modify-write instructions, the port latch is read. Some pins also have alternate input or output functions. This alternate functions are described below:

P1.0 : T2 External I/O for Timer/Counter 2
P1.1 : T2EX Timer/Counter 2 Capture/Reload Trigger
P1.2 : RXD1 Serial Port 1 Receive
P1.3 : TXD1 Serial Port 1 Transmit
P1.4 : INT2 External Interrupt 2
P1.5 : $\overline{\text{INT3}}$ External Interrupt 3
P1.6 : INT4 External Interrupt 4
P1.7 : $\overline{\text{INT5}}$ External Interrupt 5

**External Interrupt Flag**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | IE5 | IE4 | IE3 | IE2 | - | - | - | - |

Mnemonic: EXIF Address: 91h

IE5: External Interrupt 5 flag. Set by hardware when a falling edge is detected on $\overline{\text{INT5}}$.
IE4: External Interrupt 4 flag. Set by hardware when a rising edge is detected on INT4.
IE3: External Interrupt 3 flag. Set by hardware when a falling edge is detected on $\overline{\text{INT3}}$.
IE2: External Interrupt 2 flag. Set by hardware when a rising edge is detected on INT2.

**Port 4 Control Register A**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | P41M1 | P41M0 | P41C1 | P41C0 | P40M1 | P40M0 | P40C1 | P40C0 |

Mnemonic: P4CONA Address: 92h

**Port 4 Control Register B**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | P43M1 | P43M0 | P43C1 | P43C0 | P42M1 | P42M0 | P42C1 | P42C0 |

Mnemonic: P4CONB Address: 93h

| BIT NAME | FUNCTION |
|---|---|
| P4xM1, P4xM0 | Port 4 alternate modes.<br>=00: Mode 0. P4.x is a general purpose I/O port which is the same as Port 1.<br>=01: Mode 1. P4.x is a Read Strobe signal for chip select purpose. The address range depends on the SFR P4xAH, P4xAL and bits P4xC1, P4xC0.<br>=10: Mode 2. P4.x is a Write Strobe signal for chip select purpose. The address range depends on the SFR P4xAH, P4xAL and bits P4xC1, P4xC0.<br>=11: Mode 3. P4.x is a Read/Write Strobe signal for chip select purpose. The address range depends on the SFR P4xAH, P4xAL and bits P4xC1, P4xC0 |
| P4xC1, P4xC0 | Port 4 Chip-select Mode address comparison:<br>=00: Compare the full address (16 bits length) with the base address registers P4xAH and P4xAL.<br>=01: Compare the 15 high bits (A15-A1) of address bus with the base address registers P4xAH and P4xAL.<br>=10: Compare the 14 high bits (A15-A2) of address bus with the base address registers P4xAH and P4xAL.<br>=11: Compare the 8 high bits (A15-A8) of address bus with the base address registers P4xAH and P4xAL. |

**P4.0 Base Address Low Byte Register**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |

Mnemonic: P40AL Address: 94h

**P4.0 Base Address High Byte Register**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 |

Mnemonic: P40AH Address: 95h

**P4.1 Base Address Low Byte Register**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |

Mnemonic: P41AL Address: 96h

**P4.1 Base Address High Byte Register**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 |

Mnemonic: P41AH Address: 97h

**Serial Port Control**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SM0/FE | SM1 | SM2 | REN | TB8 | RB8 | TI | RI |

Mnemonic: SCON Address: 98h

SM0/FE: Serial port 0, Mode 0 bit or Framing Error Flag: The SMOD0 bit in PCON SFR determines whether this bit acts as SM0 or as FE. The operation of SM0 is described below. When used as FE, this bit will be set to indicate an invalid stop bit. This bit must be manually cleared in software to clear the FE condition.

SM1: Serial port Mode bit 1:

| SM0 | SM1 | Mode | Description | Length | Baud rate |
|---|---|---|---|---|---|
| 0 | 0 | 0 | Synchronous | 8 | 4/12 Tclk |
| 0 | 1 | 1 | Asynchronous | 10 | Variable |
| 1 | 0 | 2 | Asynchronous | 11 | 64/32 Tclk |
| 1 | 1 | 3 | Asynchronous | 11 | Variable |

SM2: Multiple processors communication. Setting this bit to 1 enables the multiprocessor communication feature in mode 2 and 3. In mode 2 or 3, if SM2 is set to 1, then RI will not be activated if the received 9$^{th}$ data bit (RB8) is 0. In mode 1, if SM2 = 1, then RI will not be activated if a valid stop bit was not received. In mode 0, the SM2 bit controls the serial port clock. If set to 0, then the serial port runs at a divide by 12 clock of the oscillator. This gives compatibility with the standard 8052. When set to 1, the serial clock become divide by 4 of the oscillator clock. This results in faster synchronous serial communication.

REN: Receive enable: When set to 1 serial reception is enabled, otherwise reception is disabled.

TB8: This is the 9$^{th}$ bit to be transmitted in modes 2 and 3. This bit is set and cleared by software as desired.

RB8: In modes 2 and 3 this is the received 9$^{th}$ data bit. In mode 1, if SM2 = 0, RB8 is the stop bit that was received. In mode 0 it has no function.

TI: Transmit interrupt flag: This flag is set by hardware at the end of the 8$^{th}$ bit time in mode 0, or at the beginning of the stop bit in all other modes during serial transmission. This bit must be cleared by software.

RI: Receive interrupt flag: This flag is set by hardware at the end of the 8$^{th}$ bit time in mode 0, or halfway through the stop bits time in the other modes during serial reception. However the restrictions of SM2 apply to this bit. This bit can be cleared only by software.

**Serial Data Buffer**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SBUF.7 | SBUF.6 | SBUF.5 | SBUF.4 | SBUF.3 | SBUF.2 | SBUF.1 | SBUF.0 |

Mnemonic: SBUF Address: 99h

SBUF.7-0: Serial data on the serial port 0 is read from or written to this location. It actually consists of two separate internal 8-bit registers. One is the receive resister, and the other is the transmit buffer. Any read access gets data from the receive data buffer, while write access is to the transmit data buffer.

**P4.2 Base Address Low Byte Register**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |

Mnemonic: P42AL Address: 9Ah

**P4.2 Base Address High Byte Register**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 |

Mnemonic: P42AH Address: 9Bh

**P4.3 Base Address Low Byte Register**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |

Mnemonic: P43AL Address: 9Ch

**P4.3 Base Address High Byte Register**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 |

Mnemonic: P43AH Address: 9Dh

**ISP Control Register**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SWRST/HWB | - | LDAP | - | - | - | LDSEL | ENP |

Mnemonic: CHPCON Address: 9Fh

SWRST/HWB: Set this bit to launch a whole device reset that is same as asserting high to RST pin, micro controller will be back to initial state and clear this bit automatically. To read this bit, its alternate function to indicate the ISP hardware reboot mode is invoking when read it in high.

LDAP: This bit is Read Only. High: device is executing the program in LDFLASH. Low: device is executing the program in APFLASHs.

LDSEL: Loader program residence selection. Set to high to route the device fetching code from LDFLASH.

ENP: In System Programming Mode Enable. Set this be to launch the ISP mode. Device will operate ISP procedures, such as Erase, Program and Read operations, according to correlative SFRs settings. During ISP mode, device achieves ISP operations by the way of IDLE state. In the other words, device is not indeed in IDLE mode is set bit PCON.1 while ISP is enabled. Clear this bit to disable ISP mode, device get back to normal operation including IDLE state.

**Software Reset**

Set CHPCON = 0X83, timer and enter IDLE mode. CPU will reset and restart from APFLASH after time out.

**Port 2**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | P2.7 | P2.6 | P2.5 | P2.4 | P2.3 | P2.2 | P2.1 | P2.0 |

Mnemonic: P2 Address: A0h

P2.7-0: Port 2 is a bi-directional I/O port with internal pull-ups. This port also provides the upper address bits for accesses to external memory.

**Port 4 Chip-select Polarity**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | P43INV | P42INV | P42INV | P40INV | - | - | - | P0UP |

Mnemonic: P4CSIN Address: A2h

P4xINV: The active polarity of P4.x when set it as chip-select signal. High = Active High. Low = Active Low.

P0UP: Enable Port 0 weak pull up.

**Port 4**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | P4.3 | P4.2 | P4.1 | P4.0 |

Mnemonic: P4 Address: A5h

P4.3-0: Port 4 is a bi-directional I/O port with internal pull-ups. Port 4 can not use bit-addressable instruction (SETB or CLR).

**Interrupt Enable**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | EA | ES1 | ET2 | ES | ET1 | EX1 | ET0 | EX0 |

Mnemonic: IE Address: A8h

EA: Global enable. Enable/disable all interrupts except for PFI.
ES1: Enable Serial Port 1 interrupt.
ET2: Enable Timer 2 interrupt.
ES: Enable Serial Port 0 interrupt.
ET1: Enable Timer 1 interrupt
EX1: Enable external interrupt 1
ET0: Enable Timer 0 interrupt
EX0: Enable external interrupt 0

**Slave Address**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |

Mnemonic: SADDR Address: A9h

SADDR: The SADDR should be programmed to the given or broadcast address for serial port 0 to which the slave processor is designated.

**Slave Address 1**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |

Mnemonic: SADDR1 Address: Aah

SADDR1: The SADDR1 should be programmed to the given or broadcast address for serial port 1 to which the slave processor is designated.

**ROM Banking Control**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | - | - | EN128K | DCP12 | DCP11 | DCP10 |

Mnemonic: ROMCON Address: Abh

EN128K: On-chip ROM banking enable. Set this bit to enable APFLASH0 and APFLASH1 by banking mechanism. The P1.x is selected to be the auxiliary highest address line A16.

DCP1x: A16 selection. By default, P1.7 is defined as A16.

| A16 | P1.0 | P1.1 | P1.2 | P1.3 | P1.4 | P1.5 | P1.6 | P1.7 |
|---|---|---|---|---|---|---|---|---|
| DCP12 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| DCP11 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| DCP10 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

**ISP Address Low Byte**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |

Mnemonic: SFRAL Address: Ach

Low byte destination address for In System Programming operations. SFRAH and SFRAL address a specific ROM bytes for erasure, escription or read.

**ISP Address High Byte**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 |

Mnemonic: SFRAH Address: Adh

High byte destination address for In System Programming operations. SFRAH and SFRAL address a specific ROM bytes for erasure, escription or read.

**ISP Data Buffer**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |

Mnemonic: SFRFD Address: Aeh

In ISP mode, read/write a specific byte ROM content must go through SFRFD register.

**ISP Operation Modes**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | BANK | WFWIN | NOE | NCE | CTRL3 | CTRL2 | CTRL1 | CTRL0 |

Mnemonic: SFRCN Address: Afh

BANK: Select APFLASH banks for ISP. Set it 1 access to APFLASH1, clear it to APFLASH0.

WFWIN: Destenation ROM bank for programming, erasure and read. 0 = APFLASHx, 1 = LDFLASH.

NOE: Flash EPROM output enable.

NCE: Flash EPROM chip enable.

CTRL[3:0]: Mode Selection.

| ISP MODE | BANK | WFWIN | NOE | NCE | CTRL<3:0> | SFRAH, SFRAL | SFRFD |
|---|---|---|---|---|---|---|---|
| Erase 4KB LDFLASH | 0 | 1 | 1 | 0 | 0010 | X | X |
| Erase 64K APFLASH0 | 0 | 0 | 1 | 0 | 0010 | X | X |
| Erase 64K APFLASH1 | 1 | 0 | 1 | 0 | 0010 | X | X |
| Program 4KB LDFLASH | 0 | 1 | 1 | 0 | 0001 | Address in | Data in |
| Program 64KB APFLASH0 | 0 | 0 | 1 | 0 | 0001 | Address in | Data in |
| Program 64KB APFLASH1 | 1 | 0 | 1 | 0 | 0001 | Address in | Data in |
| Read 4KB LDFLASH | 0 | 1 | 0 | 0 | 0000 | Address in | Data out |
| Read 64KB APFLASH0 | 0 | 0 | 0 | 0 | 0000 | Address in | Data out |
| Read 64KB APFLASH1 | 1 | 0 | 0 | 0 | 0000 | Address in | Data out |

**Port 3**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | P3.7 | P3.6 | P3.5 | P3.4 | P3.3 | P3.2 | P3.1 | P3.0 |

Mnemonic: P3 Address: B0h

P3.7-0: General purpose I/O port. Each pin also has an alternate input or output function. The alternate functions are described below.

| | | |
|---|---|---|
| P3.7 | $\overline{RD}$ | Strobe for read from external RAM |
| P3.6 | $\overline{WR}$ | Strobe for write to external RAM |
| P3.5 | T1 | Timer/counter 1 external count input |
| P3.4 | T0 | Timer/counter 0 external count input |
| P3.3 | $\overline{INT1}$ | External interrupt 1 |
| P3.2 | $\overline{INT0}$ | External interrupt 0 |
| P3.1 | TxD | Serial port 0 output |
| P3.0 | RxD | Serial port 0 input |

**Interrupt Priority**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | PS1 | PT2 | PS | PT1 | PX1 | PT0 | PX0 |

Mnemonic: IP Address: B8h

IP.7: This bit is un-implemented and will read high.
PS1: This bit defines the Serial port 1 interrupt priority. PS = 1 sets it to higher priority level.
PT2: This bit defines the Timer 2 interrupt priority. PT2 = 1 sets it to higher priority level.
PS: This bit defines the Serial port 0 interrupt priority. PS = 1 sets it to higher priority level.
PT1: This bit defines the Timer 1 interrupt priority. PT1 = 1 sets it to higher priority level.
PX1: This bit defines the External interrupt 1 priority. PX1 = 1 sets it to higher priority level.
PT0: This bit defines the Timer 0 interrupt priority. PT0 = 1 sets it to higher priority level.
PX0: This bit defines the External interrupt 0 priority. PX0 = 1 sets it to higher priority level.

**Slave Address Mask Enable**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |

Mnemonic: SADEN Address: B9h

SADEN: This register enables the Automatic Address Recognition feature of the Serial port 0. When a bit in the SADEN is set to 1, the same bit location in SADDR will be compared with the incoming serial data. When SADEN.n is 0, then the bit becomes a “don’t care” in the comparison. This register enables the Automatic Address Recognition feature of the Serial port 0. When all the bits of SADEN are 0, interrupt will occur for any incoming address.

**Slave Address Mask Enable 1**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |

Mnemonic: SADEN1 Address: Bah

SADEN1: This register enables the Automatic Address Recognition feature of the Serial port 1. When a bit in the SADEN1 is set to 1, the same bit location in SADDR1 will be compared with the incoming serial data. When SADEN1.n is 0, then the bit becomes a “don’t care” in the comparison. This register enables the Automatic Address Recognition feature of the Serial port 1. When all the bits of SADEN1 are 0, interrupt will occur for any incoming address.

**Serial Port Control 1**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SM0_1/FE_1 | SM1_1 | SM2_1 | REN_1 | TB8_1 | RB8_1 | TI_1 | RI_1 |

Mnemonic: SCON1 Address: C0h

SM0_1/FE_1: Serial port 1, Mode 0 bit or Framing Error Flag 1: The SMOD0 bit in PCON SFR determines whether this bit acts as SM0_1 or as FE_1. the operation of SM0_1 is described below. When used as FE_1, this bit will be set to indicate an invalid stop bit. This bit must be manually cleared in software to clear the FE_1 condition.

SM1_1: Serial port 1 Mode bit 1:

| SM0_1 | SM1_1 | Mode | Description | Length | Baud rate |
|---|---|---|---|---|---|
| 0 | 0 | 0 | Synchronous | 8 | 4/12 Tclk |
| 0 | 1 | 1 | Asynchronous | 10 | variable |
| 1 | 0 | 2 | Asynchronous | 11 | 64/32 Tclk |
| 1 | 1 | 3 | Asynchronous | 11 | variable |

SM2_1: Multiple processors communication. Setting this bit to 1 enables the multiprocessor communication feature in mode 2 and 3. In mode 2 or 3, if SM2_1 is set to 1, then RI_1 will not be activated if the received 9$^{th}$ data bit (RB8_1) is 0. In mode 1, if SM2_1 = 1, then RI_1 will not be activated if a valid stop bit was not received. In mode 0, the SM2_1 bit controls the serial port 1 clock. If set to 0, then the serial port 1 runs at a divide by 12 clock of the oscillator. This gives compatibility with the standard 8052. When set to 1, the serial clock become divide by 4 of the oscillator clock. This results in faster synchronous serial communication.

REN_1: Receive enable: When set to 1 serial reception is enabled, otherwise reception is disabled.

TB8_1: This is the 9$^{th}$ bit to be transmitted in modes 2 and 3. This bit is set and cleared by software as desired.

RB8_1: In modes 2 and 3 this is the received 9$^{th}$ data bit. In mode 1, if SM2_1 = 0, RB8_1 is the stop bit that was received. In mode 0 it has no function.

TI_1: Transmit interrupt flag: This flag is set by hardware at the end of the 8$^{th}$ bit time in mode 0, or at the beginning of the stop bit in all other modes during serial transmission. This bit must be cleared by software.

RI_1: Receive interrupt flag: This flag is set by hardware at the end of the 8th bit time in mode 0, or halfway through the stop bits time in the other modes during serial reception. However the restrictions of SM2_1 apply to this bit. This bit can be cleared only by software.

**Serial Data Buffer 1**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SBUF1.7 | SBUF1.6 | SBUF1.5 | SBUF1.4 | SBUF1.3 | SBUF1.2 | SBUF1.1 | SBUF1.0 |

Mnemonic: SBUF1 Address: C1h

SBUF1.7-0: Serial data of the serial port 1 is read from or written to this location. It actually consists of two separate 8-bit registers. One is the receive resister, and the other is the transmit buffer. Any read access gets data from the receive data buffer, while write accesses are to the transmit data buffer.

**WSCON**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | WS | - | - | - | - | - | - | - |

Mnemonic: WSCON Address: C2h

WS: Wait State Signal Enable. Setting this bit enables the $\overline{WAIT}$ signal on P4.0. The device will sample the wait state control signal $\overline{WAIT}$ via P4.0 during MOVX instruction. This bit is time access protected.

```
TA REG  C7H
WSCON   REG  C2H
CKCON   REG  8EH
MOV          TA, #AAH
MOV          TA, #55H
ORL          WSCON, #10000000B ; Set  WS bit  and stretch value = 0 to enable wait
              signal.
```

**Power Management Register**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CD1 | CD0 | SWB | - | - | ALE-OFF | - | DME0 |

Mnemonic: PMR Address: C4h

CD1, CD0: Clock Divide Control. These bit selects the number of clocks required to generate one machine cycle. There are three modes including divide by 4, 64 or 1024. Switching between modes must first go back divide by 4 mode. For instance, to go from 64 to 1024 clocks/machine cycle the device must first go from 64 to 4 clocks/machine cycle, and then
from 4 to 1024 clocks/machine cycle.

| CD1, | CD0 | CLOCKS/MACHINE CYCLE |
|---|---|---|
| 0 | 0 | Reserved |
| 0 | 1 | 4 |
| 1 | 0 | 64 |
| 1 | 1 | 1024 |

SWB: Switchback Enable. Setting this bit allows an enabled external interrupt or serial port activity to force the CD1, CD0 to divide by 4 state (0, 1). The device will switch modes at the start of the jump to interrupt service routine while a external interrupt is enabled and actually recongnized by microcontroller. While a serial port reception, the switchback occurs at the start of the instruction following the falling edge of the start bit.

ALE0FF: This bit disables the expression of the ALE signal on the device pin during all on-board program and data memory accesses. External memory accesses will automatically enable ALE independent of ALEOFF.

0 = ALE expression is enable; 1 = ALE expression is disable

DME0: This bit determines the on-chip MOVX SRAM to be enabled or disabled. Set this bit to 1 will enable the on-chip 1KB MOVX SRAM.

**Status Register**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | HIP | LIP | XTUP | SPTA1 | SPRA1 | SPTA0 | SPRA0 |

Mnemonic: STATUS Address: C5h

HIP: High Priority Interrupt Status. When set, it indicates that software is servicing a high priority interrupt. This bit will be cleared when the program executes the corresponding RETI instruction.

LIP: Low Priority Interrupt Status. When set, it indicates that software is servicing a low priority interrupt. This bit will be cleared when the program executes the corresponding RETI instruction.

XTUP: Crystal Oscillator Warm-up Status. When set, this bit indicates CPU has detected clock to be ready. Each time the crystal oscillator is restarted by exit from power down mode, hardware will clear this bit. This bit is set to 1 after a power-on reset.

SPTA1: Serial Port 1 Transmit Activity. This bit is set during serial port 1 is currently transmitting data. It is cleared when TI_1 bit is set by hardware. Changing the Clock Divide Control bits CD0, CD1 will be ignored when this bit is set to 1 and SWB = 1.

SPRA1: Serial Port 1 Receive Activity. This bit is set during serial port 1 is currently receiving a data. It is cleared when RI_1 bit is set by hardware. Changing the Clock Divide Control bits CD0, CD1
will be ignored when this bit is set to 1 and SWB = 1.

SPTA0: Serial Port 0 Transmit Activity. This bit is set during serial port 0 is currently transmitting data. It is cleared when TI bit is set by hardware. Changing the Clock Divide Control bits CD0, CD1 will be ignored when this bit is set to 1 and SWB = 1.

SPRA0: Serial Port 0 Receive Activity. This bit is set during serial port 0 is currently receiving a data. It is cleared when RI bit is set by hardware. Changing the Clock Divide Control bits CD0, CD1 will be ignored when this bit is set to 1 and SWB = 1.

**Timed Access**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TA.7 | TA.6 | TA.5 | TA.4 | TA.3 | TA.2 | TA.1 | TfA.0 |

Mnemonic: TA Address: C7h

TA: The Timed Access register controls the access to protected bits. To access protected bits, the user must first write AAH to the TA. This must be immediately followed by a write of 55H to TA. Now a window is opened in the protected bits for three machine cycles, during which the user can write to these bits.

**Timer 2 Control**

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

Mnemonic: T2CON Address: C8h

TF2: Timer 2 overflow flag: This bit is set when Timer 2 overflows. It is also set when the count is equal to the capture register in down count mode. It can be set only if RCLK and TCLK are both 0. It is cleared only by software. Software can also set or clear this bit.

EXF2: Timer 2 External Flag: A negative transition on the T2EX pin (P1.1) or timer 2 overflow will cause this flag to set based on the CP / $\overline{RL2}$, EXEN2 and DCEN bits. If set by a negative transition, this flag must be cleared by software. Setting this bit in software or detection of a negative transition on T2EX pin will force a timer interrupt if enabled.

RCLK: Receive Clock Flag: This bit determines the serial port 0 time-base when receiving data in serial modes 1 or 3. If it is 0, then timer 1 overflow is used for baud rate generation, otherwise timer 2 overflow is used. Setting this bit forces timer 2 in baud rate generator mode.

TCLK: Transmit Clock Flag: This bit determines the serial port 0 time-base when transmitting data in modes 1 and 3. If it is set to 0, the timer 1 overflow is used to generate the baud rate clock otherwise timer 2 overflow is used. Setting this bit forces timer 2 in baud rate generator mode.

EXEN2: Timer 2 External Enable. This bit enables the capture/reload function on the T2EX pin if Timer 2 is not generating baud clocks for the serial port. If this bit is 0, then the T2EX pin will be ignored, otherwise a negative transition detected on the T2EX pin will result in capture or reload.

TR2: Timer 2 Run Control. This bit enables/disables the operation of timer 2. Clearing this bit will halt the timer 2 and preserve the current count in TH2, TL2.

C / $\overline{T2}$ : Counter/Timer Select. This bit determines whether timer 2 will function as a timer or a counter. Independent of this bit, the timer will run at 2 clocks per tick when used in baud rate generator mode. If it is set to 0, then timer 2 operates as a timer at a speed depending on T2M bit (CKCON.5), otherwise it will count negative edges on T2 pin.

CP / $\overline{RL2}$ : Capture/Reload Select. This bit determines whether the capture or reload function will be used for timer 2. If either RCLK or TCLK is set, this bit will be ignored and the timer will function in an auto-reload mode following each overflow. If the bit is 0 then auto-reload will occur when timer 2 overflows or a falling edge is detected on T2EX pin if EXEN2 = 1. If this bit is 1, then timer 2 captures will occur when a falling edge is detected on T2EX pin if EXEN2 = 1.

**Timed 2 Mode Control**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | HC5 | HC4 | HC3 | HC2 | T2CR | - | T2OE | DCEN |

Mnemonic: T2MOD Address: C9h

HC5: Hardware Clear $\overline{INT5}$ flag. Setting this bit allows the flag of external interrupt 5 to be automatically cleared by hardware while entering the interrupt service routine.

HC4: Hardware Clear INT4 flag. Setting this bit allows the flag of external interrupt 4 to be automatically cleared by hardware while entering the interrupt service routine.

HC3: Hardware Clear $\overline{INT3}$ flag. Setting this bit allows the flag of external interrupt 3 to be automatically cleared by hardware while entering the interrupt service routine.

HC3: Hardware Clear INT2 flag. Setting this bit allows the flag of external interrupt 3 to be automatically cleared by hardware while entering the interrupt service routine.

T2CR: Timer 2 Capture Reset. In the Timer 2 Capture Mode this bit enables/disables hardware automatically reset Timer 2 while the value in TL2 and TH2 have been transferred into the capture register.

T2OE: Timer 2 Output Enable. This bit enables/disables the Timer 2 clock out function.

DCEN: Down Count Enable: This bit, in conjunction with the T2EX pin, controls the direction that timer 2 counts in 16-bit auto-reload mode.

**Timer 2 Capture LSB**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RCAP2L.7 | RCAP2L.6 | RCAP2L.5 | RCAP2L.4 | RCAP2L.3 | RCAP2L.2 | RCAP2L.1 | RCAP2L.0 |

Mnemonic: RCAP2L Address: Cah

RCAP2L: This register is used to capture the TL2 value when a timer 2 is configured in capture mode. RCAP2L is also used as the LSB of a 16-bit reload value when timer 2 is configured in auto-reload mode.

**Timer 2 Capture MSB**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RCAP2H.7 | RCAP2H.6 | RCAP2H.5 | RCAP2H.4 | RCAP2H.3 | RCAP2H.2 | RCAP2H.1 | RCAP2H.0 |

Mnemonic: RCAP2H Address: CBh

RCAP2H: This register is used to capture the TH2 value when a timer 2 is configured in capture mode.
RCAP2H is also used as the MSB of a 16-bit reload value when timer 2 is configured in auto-reload mode.

**Timer 2 LSB**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TL2.7 | TL2.6 | TL2.5 | TL2.4 | TL2.3 | TL2.2 | TL2.1 | TL2.0 |

Mnemonic: TL2 Address: CCh

TL2: Timer 2 LSB

**Timer 2 MSB**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TH2.7 | TH2.6 | TH2.5 | TH2.4 | TH2.3 | TH2.2 | TH2.1 | TH2.0 |

Mnemonic: TH2 Address: CDh

TH2: Timer 2 MSB

**Program Status Word**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CY | AC | F0 | RS1 | RS0 | OV | F1 | P |

Mnemonic: PSW Address: D0h

CY: Carry flag: Set for an arithmetic operation which results in a carry being generated from the ALU. It is also used as the accumulator for the bit operations.

AC: Auxiliary carry: Set when the previous operation resulted in a carry from the high order nibble.

F0: User flag 0: General purpose flag that can be set or cleared by the user.

RS.1-0: Register bank select bits:

| RS1 | RS0 | Register bank | Address |
|---|---|---|---|
| 0 | 0 | 0 | 00-07h |
| 0 | 1 | 1 | 08-0Fh |
| 1 | 0 | 2 | 10-17h |
| 1 | 1 | 3 | 18-1Fh |

OV: Overflow flag: Set when a carry was generated from the seventh bit but not from the 8$^{th}$ bit as a result of the previous operation, or vice-versa.

F1: User Flag 1: General purpose flag that can be set or cleared by the user by software.

P: Parity flag: Set/cleared by hardware to indicate odd/even number of 1's in the accumulator.

**Watchdog Control**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SMOD_1 | POR | - | - | WDIF | WTRF | EWT | RWT |

Mnemonic: WDCON Address: D8h

SMOD_1:This bit doubles the Serial Port 1 baud rate in mode 1, 2, and 3 when set to 1.

POR: Power-on reset flag. Hardware will set this flag on a power up condition. This flag can be read or written by software. A write by software is the only way to clear this bit once it is set.

WDIF: Watchdog Timer Interrupt Flag. If the watchdog interrupt is enabled, hardware will set this bit to indicate that the watchdog interrupt has occurred. If the interrupt is not enabled, then this bit indicates that the time-out period has elapsed. This bit must be cleared by software.

WTRF: Watchdog Timer Reset Flag. Hardware will set this bit when the watchdog timer causes a reset. Software can read it but must clear it manually. A power-fail reset will also clear the bit. This bit helps software in determining the cause of a reset. If EWT = 0, the watchdog timer will have no affect on this bit.

EWT: Enable Watchdog timer Reset. Setting this bit will enable the Watchdog timer Reset function.

RWT: Reset Watchdog Timer. This bit helps in putting the watchdog timer into a know state. It also helps in resetting the watchdog timer before a time-out occurs. Failing to set the EWT before time-out will cause an interrupt, if EWDI (EIE.4) is set, and 512 clocks after that a watchdog timer reset will be generated if EWT is set. This bit is self-clearing by hardware.

The WDCON SFR is set to a 0x0x0xx0b on an external reset. WTRF is set to a 1 on a Watchdog timer reset, but to a 0 on power on/down resets. WTRF is not altered by an external reset. POR is set to 1 by a power-on reset. EWT is set to 0 on a Power-on reset and unaffected by other resets.

All the bits in this SFR have unrestricted read access. POR, EWT, WDIF and RWT require Timed Access procedure to write. The remaining bits have unrestricted write accesses. Please refer TA register escription.

```
TA          EG    C7H
WDCON       REG   D8H
CKCON       REG   8EH
        MOV  TA, #AAH
        MOV  TA, #55H
        SETB  WDCON.0                ; Reset watchdog timer
        ORL  CKCON, #11000000B       ; Select 26 bits watchdog timer
        MOV  TA, #AAH
        MOV  TA, #55H
        ORL  WDCON, #00000010B       ; Enable watchdog
```

**Accumulator**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ACC.7 | ACC.6 | ACC.5 | ACC.4 | ACC.3 | ACC.2 | ACC.1 | ACC.0 |

Mnemonic: ACC Address: E0h

ACC.7-0: The A (or ACC) register is the standard 8052 accumulator.

**Extended Interrupt Enable**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | - | EWDI | EX5 | EX4 | EX3 | EX2 |

Mnemonic: EIE Address: E8h

EIE.7-5: Reserved bits, will read high

EWDI: Enable Watchdog timer interrupt

EX5: External Interrupt 5 Enable.

EX4: External Interrupt 4 Enable.

EX3: External Interrupt 3 Enable.

EX2: External Interrupt 2 Enable.

**B Register**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | B.7 | B.6 | B.5 | B.4 | B.3 | B.2 | B.1 | B.0 |

Mnemonic: B Address: F0h

B.7-0: The B register is the standard 8052 register that serves as a second accumulator.

**Extended Interrupt Priority**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | - | - | - | PWDI | PX5 | PX4 | PX3 | PX2 |

Mnemonic: EIP Address: F8h

EIP.7-5: Reserved bits.

PWDI: Watchdog timer interrupt priority.

PX5: External Interrupt 5 Priority. 0 = Low priority, 1 = High priority.

PX4: External Interrupt 4 Priority. 0 = Low priority, 1 = High priority.

PX3: External Interrupt 3 Priority. 0 = Low priority, 1 = High priority.

PX2: External Interrupt 2 Priority. 0 = Low priority, 1 = High priority.

## 7. INSTRUCTION

The W77E532A executes all the instructions of the standard 8032 family. The operation of these instructions, their effect on the flag bits and the status bits is exactly the same. However, timing of these instructions is different. The reason for this is two fold. Firstly, in the W77E532A, each machine cycle consists of 4 clock periods, while in the standard 8032 it consists of 12 clock periods. Also, in the W77E532A there is only one fetch per machine cycle i.e. 4 clocks per fetch, while in the standard 8032 there can be two fetches per machine cycle, which works out to 6 clocks per fetch.

The advantage the W77E532A has is that since there is only one fetch per machine cycle, the number of machine cycles in most cases is equal to the number of operands that the instruction has. In case of jumps and calls there will be an additional cycle that will be needed to calculate the new address. But overall the W77E532A reduces the number of dummy fetches and wasted cycles, thereby improving efficiency as compared to the standard 8032.

**Table 2. Instructions that affect Flag settings**

| INSTRUCTION | CARRY | OVERFLOW | AUXILIARY CARRY | INSTRUCTION | CARRY | OVERFLOW | AUXILIARY CARRY |
|---|---|---|---|---|---|---|---|
| ADD | X | X | X | CLR C | 0 | | |
| ADDC | X | X | X | CPL C | X | | |
| SUBB | X | X | X | ANL C, bit | X | | |
| MUL | 0 | X | | ANL C, bit | X | | |
| DIV | 0 | X | | ORL C, bit | X | | |
| DA A | X | | | ORL C, bit | X | | |
| RRC A | X | | | MOV C, bit | X | | |
| RLC A | X | | | CJNE | X | | |
| SETB C | 1 | | | | | | |

A “X” indicates that the modification is as per the result of instruction.

## 8. INSTRUCTION TIMING

The instruction timing for the W77E532A is an important aspect, especially for those users who wish to use software instructions to generate timing delays. Also, it provides the user with an insight into the timing differences between the W77E532A and the standard 8032. In the W77E532A each machine cycle is four clock periods long. Each clock period is designated a state. Thus each machine cycle is made up of four states, C1, C2 C3 and C4, in that order. Due to the reduced time for each instruction execution, both the clock edges are used for internal timing. Hence it is important that the duty cycle of the clock be as close to 50% as possible to avoid timing conflicts. As mentioned earlier, the W77E532A does one op-code fetch per machine cycle. Therefore, in most of the instructions, the number of machine cycles needed to execute the instruction is equal to the number of bytes in the instruction. Of the 256 available op-codes, 128 of them are single cycle instructions. Thus more than half of all op-codes in the W77E532A are executed in just four clock periods. Most of the two-cycle instructions are those that have two byte instruction codes. However there are some instructions that have only one byte instructions, yet they are two cycle instructions. One instruction which is of importance is the MOVX instruction. In the standard 8032, the MOVX instruction is always two machine cycles long. However in the W77E532A, the user has a facility to stretch the duration of this instruction from 2 machine cycles to 9 machine cycles. The $\overline{RD}$ and $\overline{WR}$ strobe lines are also proportionately elongated. This gives the user flexibility in accessing both fast and slow peripherals without the use of external circuitry and with minimum software overhead. The rest of the instructions are either three, four or five machine cycle instructions. Note that in the W77E532A, based on the number of machine cycles, there are five different types, while in the standard 8032 there are only three. However, in the W77E532A each machine cycle is made of only 4 clock periods compared to the 12 clock periods for the standard 8032. Therefore, even though the number of categories has increased, each instruction is at least 1.5 to 3 times faster than the standard 8032 in terms of clock periods.

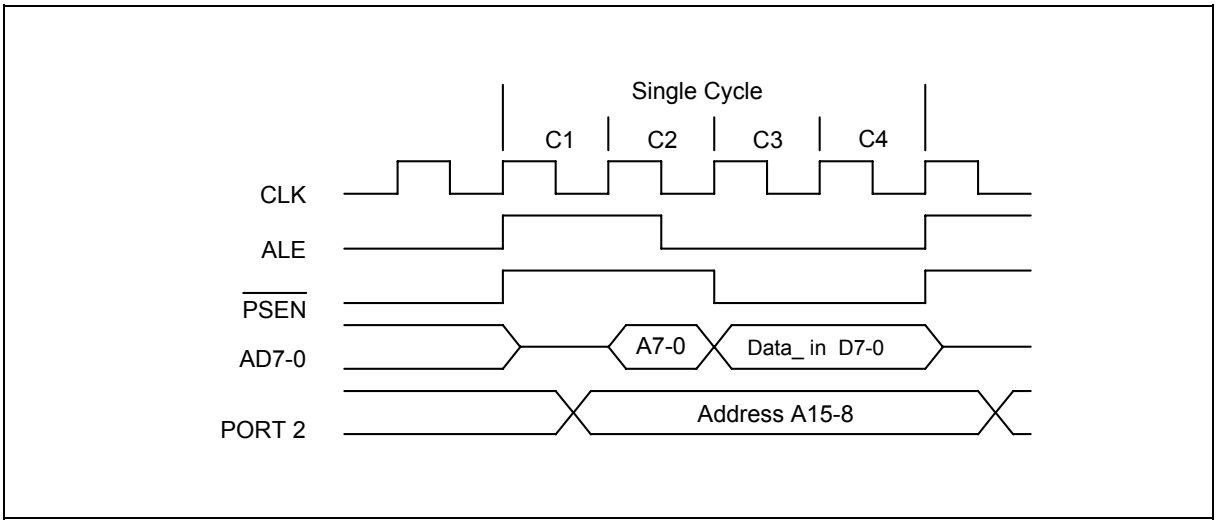



Figure 3. Single Cycle Instruction Timing

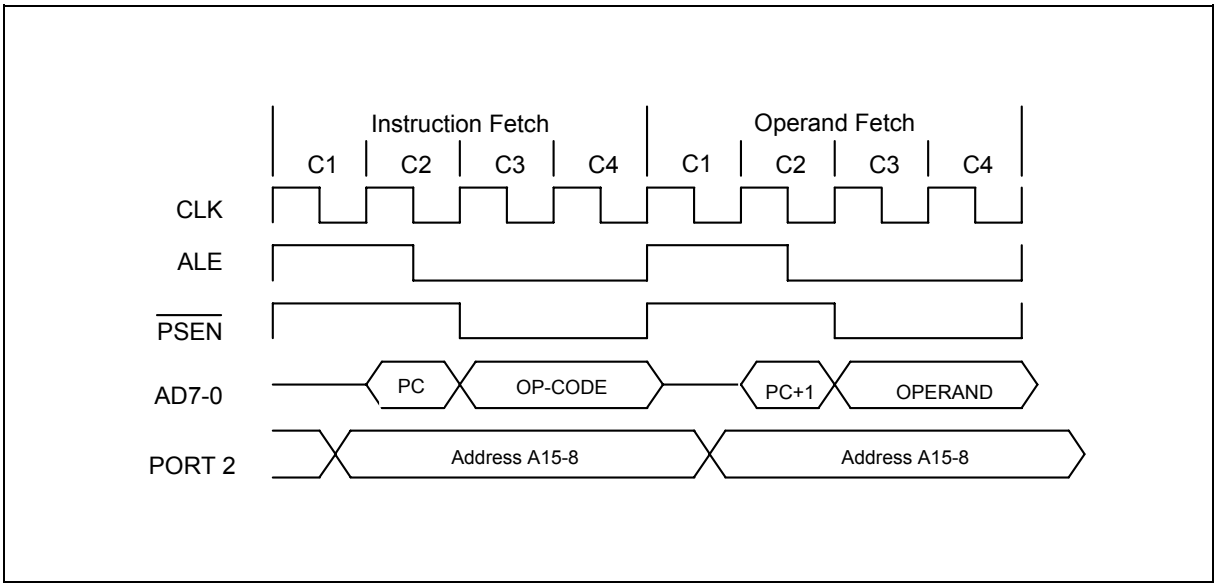



Figure 4. Two Cycle Instruction Timing

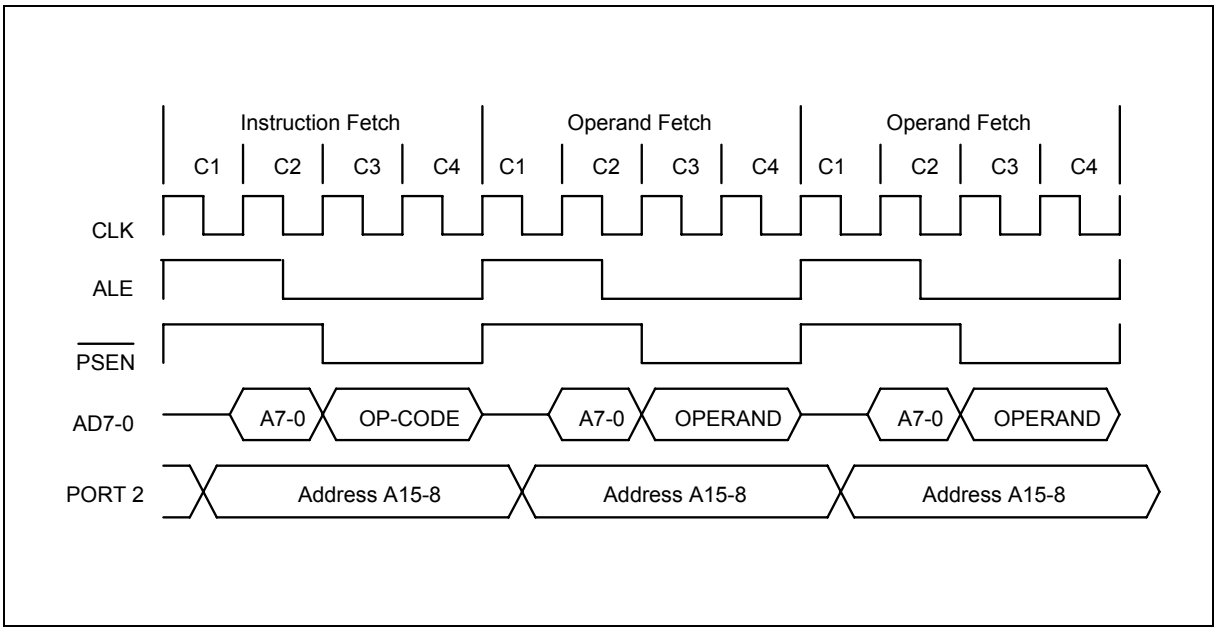



Figure 5. Three Cycle Instruction Timing

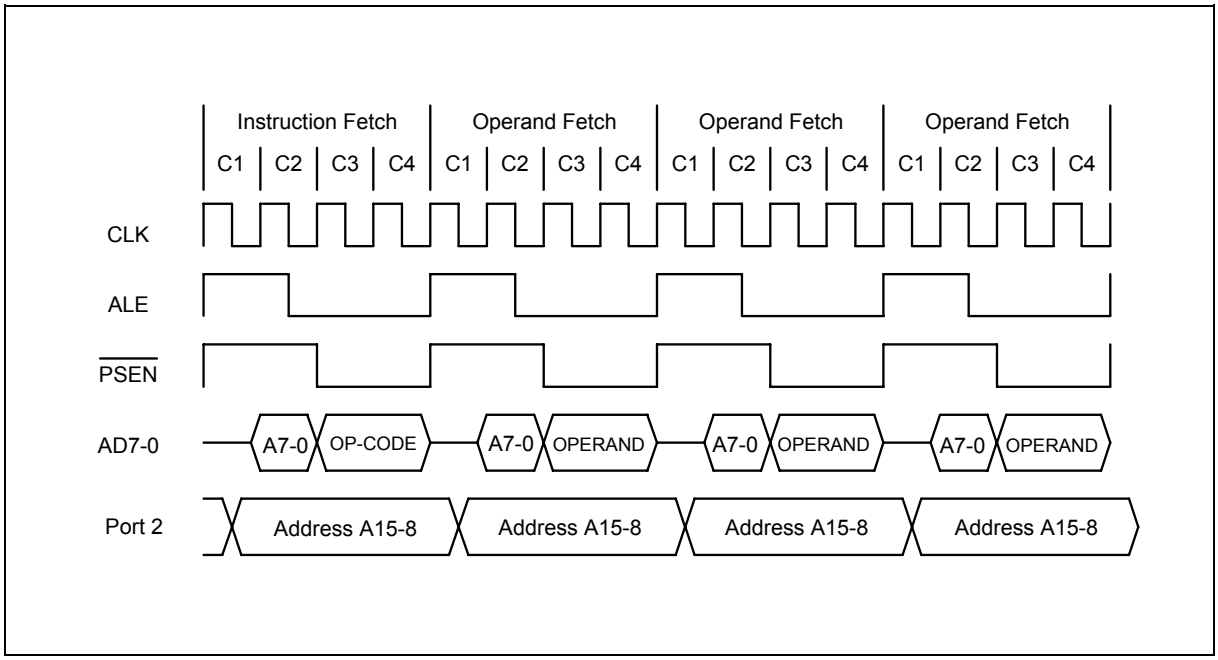



Figure 6. Four Cycle Instruction Timing

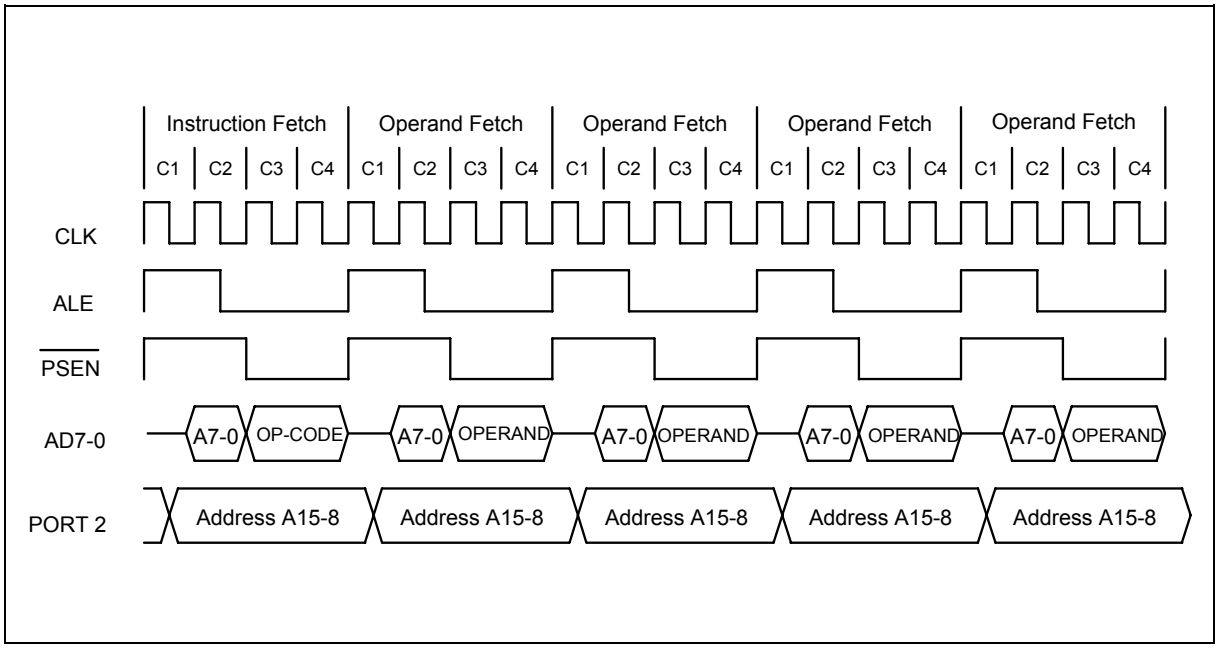



Figure 7. Five Cycle Instruction Timing

**MOVX Instruction**

The W77E532A, like the standard 8032, uses the MOVX instruction to access external Data Memory. This Data Memory includes both off-chip memory as well as memory mapped peripherals. While the results of the MOVX instruction are the same as in the standard 8032, the operation and the timing of the strobe signals have been modified in order to give the user much greater flexibility.

The MOVX instruction is of two types, the MOVX @Ri and MOVX @DPTR. In the MOVX @Ri, the address of the external data comes from two sources. The lower 8-bits of the address are stored in the Ri register of the selected working register bank. The upper 8-bits of the address come from the port 2 SFR. In the MOVX @DPTR type, the full 16-bit address is supplied by the Data Pointer.

Since the W77E532A has two Data Pointers, DPTR and DPTR1, the user has to select between the two by setting or clearing the DPS bit. The Data Pointer Select bit (DPS) is the LSB of the DPS SFR, which exists at location 86h. No other bits in this SFR have any effect, and they are set to 0. When DPS is 0, then DPTR is selected, and when set to 1, DPTR1 is selected. The user can switch between DPTR and DPTR1 by toggling the DPS bit. The quickest way to do this is by the INC instruction. The advantage of having two Data Pointers is most obvious while performing block move operations. The accompanying code shows how the use of two separate Data Pointers speeds up the execution time for code performing the same task.

Block Move with single Data Pointer:

```
; SH and SL are the high and low bytes of Source Address
; DH and DL are the high and low bytes of Destination Address
; CNT is the number of bytes to be moved
                                                    Machine cycles of W77E532A
                                                                          #
MOV   R2, #CNT          ; Load R2 with the count value                    2
MOV   R3, #SL           ; Save low byte of Source Address in R3           2
MOV   R4, #SH           ; Save high byte of Source address in R4          2
MOV   R5, #DL           ; Save low byte of Destination Address in R5      2
MOV   R6, #DH           ; Save high byte of Destination address in R6     2
LOOP:
MOV   DPL, R3           ; Load DPL with low byte of Source address        2
MOV   DPH, R4           ; Load DPH with high byte of Source address       2
MOVXA, @DPTR            ; Get byte from Source to Accumulator             2
INC   DPTR              ; Increment Source Address to next byte           2
MOV   R3, DPL           ; Save low byte of Source address in R3           2
MOV   R4, DPH           ; Save high byte of Source Address in R4          2
MOV   DPL, R5           ; Load low byte of Destination Address in DPL     2
MOV   DPH, R6           ; Load high byte of Destination Address in DPH    2
MOVX@DPTR, A            ; Write data to destination                       2
INC   DPTR              ; Increment Destination Address                   2
MOV   DPL, R5           ; Save low byte of new destination address in R5  2
MOV   DPH, R6           ; Save high byte of new destination address in R6 2
DJNZ  R2, LOOP          ; Decrement count and do LOOP again if count <> 0 2
```

Machine cycles in standard 8032 = 10 + (26 * CNT)
Machine cycles in W77E532A = 10 + (26 * CNT)
If CNT = 50
Clock cycles in standard 8032= ((10 + (26 *50)) * 12 = (10 + 1300) * 12 = 15720
Clock cycles in W77E532A = ((10 + (26 * 50)) * 4 = (10 + 1300) * 4 = 5240

Block Move with Two Data Pointers in W77E532A:

```
; SH and SL are the high and low bytes of Source Address
; DH and DL are the high and low bytes of Destination Address
; CNT is the number of bytes to be moved
                                                    Machine cycles of W77E532A
                                                                        #
MOV   R2, #CNT      ; Load R2 with the count value                      2
MOV   DPS, #00h     ; Clear DPS to point to DPTR                        2
MOV   DPTR, #DHDL         ; Load DPTR with Destination address          3
INC   DPS           ; Set DPS to point to DPTR1                         2
MOV   DPTR, #SHSL         ; Load DPTR1 with Source address              3

LOOP:
MOVXA, @DPTR        ; Get data from Source block                        2
INC   DPTR          ; Increment source address                          2
DEC   DPS           ; Clear DPS to point to DPTR                        2
MOVX@DPTR, A        ; Write data to Destination                         2
INC   DPTR          ; Increment destination address                     2
INC   DPS           ; Set DPS to point to DPTR1                         2
DJNZ  R2, LOOP      ; Check if all done                                 3
```

Machine cycles in W77E532A = 12 + (15 * CNT)

If CNT = 50

Clock cycles in W77E532A = (12 + (15 * 50)) * 4 = (12 + 750) * 4 = 3048

We can see that in the first program the standard 8032 takes 15720 cycles, while the W77E532A takes only 5240 cycles for the same code. In the second program, written for the W77E532A, program execution requires only 3048 clock cycles. If the size of the block is increased then the saving is even greater.

**External Data Memory Access Timing:**

The timing for the MOVX instruction is another feature of the W77E532A. In the standard 8032, the MOVX instruction has a fixed execution time of 2 machine cycles. However in the W77E532A, the duration of the access can be varied by the user.

The instruction starts off as a normal op-code fetch of 4 clocks. In the next machine cycle, the W77E532A puts out the address of the external Data Memory and the actual access occurs here. The user can change the duration of this access time by setting the STRETCH value. The Clock Control SFR (CKCON) has three bits that control the stretch value. These three bits are M2-0 (bits 2-0 of CKCON). These three bits give the user 8 different access time options. The stretch can be varied from 0 to 7, resulting in MOVX instructions that last from 2 to 9 machine cycles in length. Note that the stretching of the instruction only results in the elongation of the MOVX instruction, as if the state of the

CPU was held for the desired period. There is no effect on any other instruction or its timing. By default, the Stretch value is set at 1, giving a MOVX instruction of 3 machine cycles. If desired by the user the stretch value can be set to 0 to give the fastest MOVX instruction of only 2 machine cycles.

**Table 4. Data Memory Cycle Stretch Values**

| M2 | M1 | M0 | MACHINE CYCLES | $\overline{RD}$ OR $\overline{WR}$ STROBE WIDTH IN CLOCKS | $\overline{RD}$ OR $\overline{WR}$ STROBE WIDTH @ 25 MHZ | $\overline{RD}$ OR $\overline{WR}$ STROBE WIDTH @ 40 MHZ |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 2 | 2 | 80 nS | 50 nS |
| 0 | 0 | 1 | 3 (default) | 4 | 160 nS | 100 nS |
| 0 | 1 | 0 | 4 | 8 | 320 nS | 200 nS |
| 0 | 1 | 1 | 5 | 12 | 480 nS | 300 nS |
| 1 | 0 | 0 | 6 | 16 | 640 nS | 400 nS |
| 1 | 0 | 1 | 7 | 20 | 800 nS | 500 nS |
| 1 | 1 | 0 | 8 | 24 | 960 nS | 600 nS |
| 1 | 1 | 1 | 9 | 28 | 1120 nS | 700 nS |

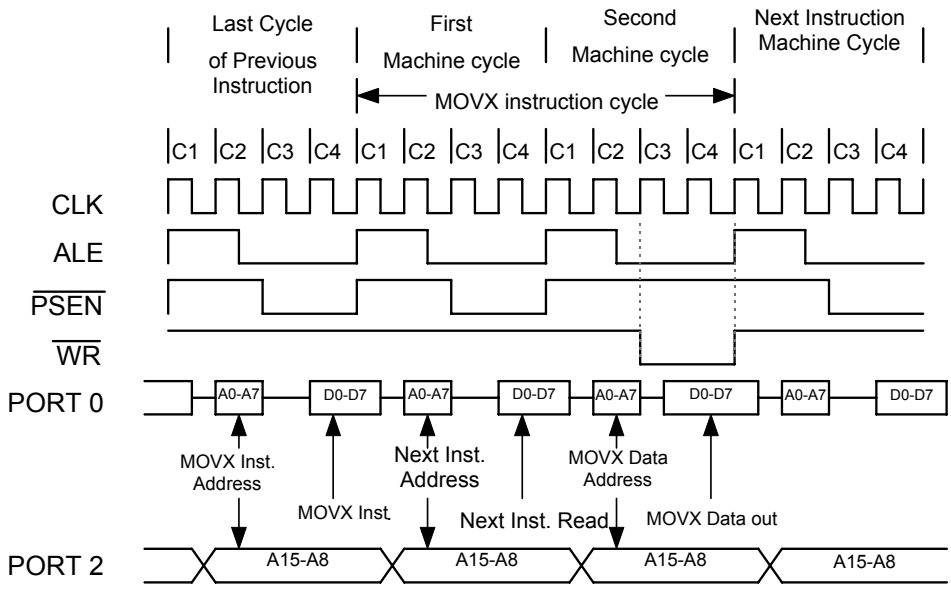



Figure 8. Data Memory Write with Stretch Value = 0

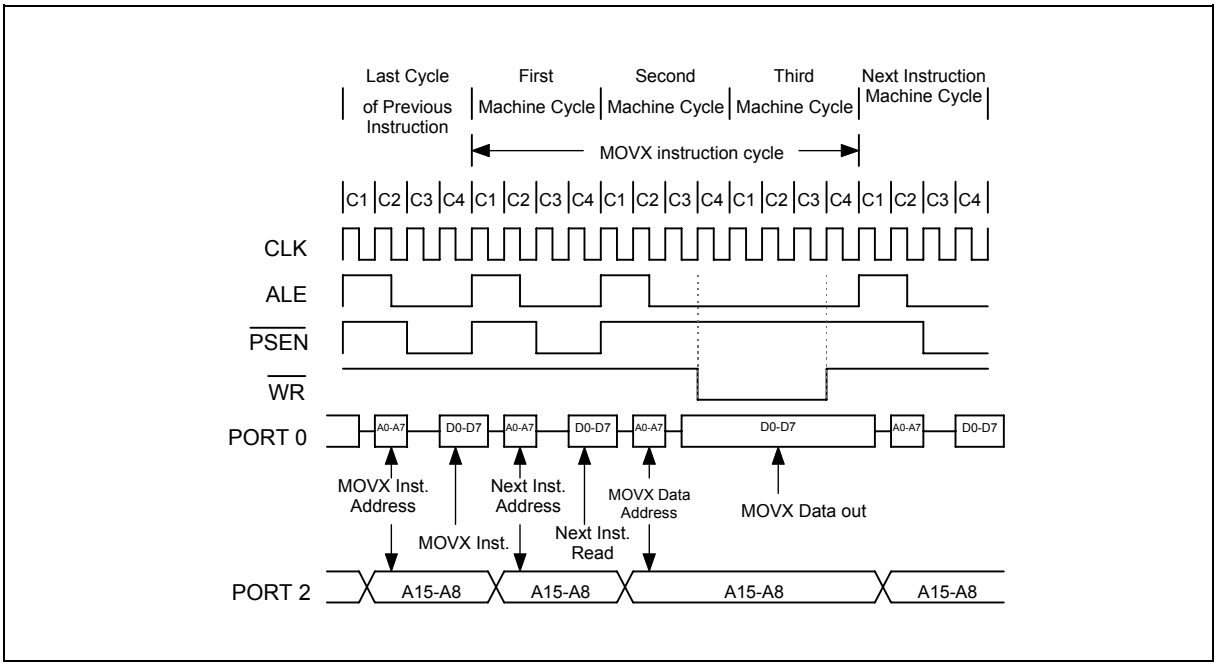



Figure 9. Data Memory Write with Stretch Value = 1

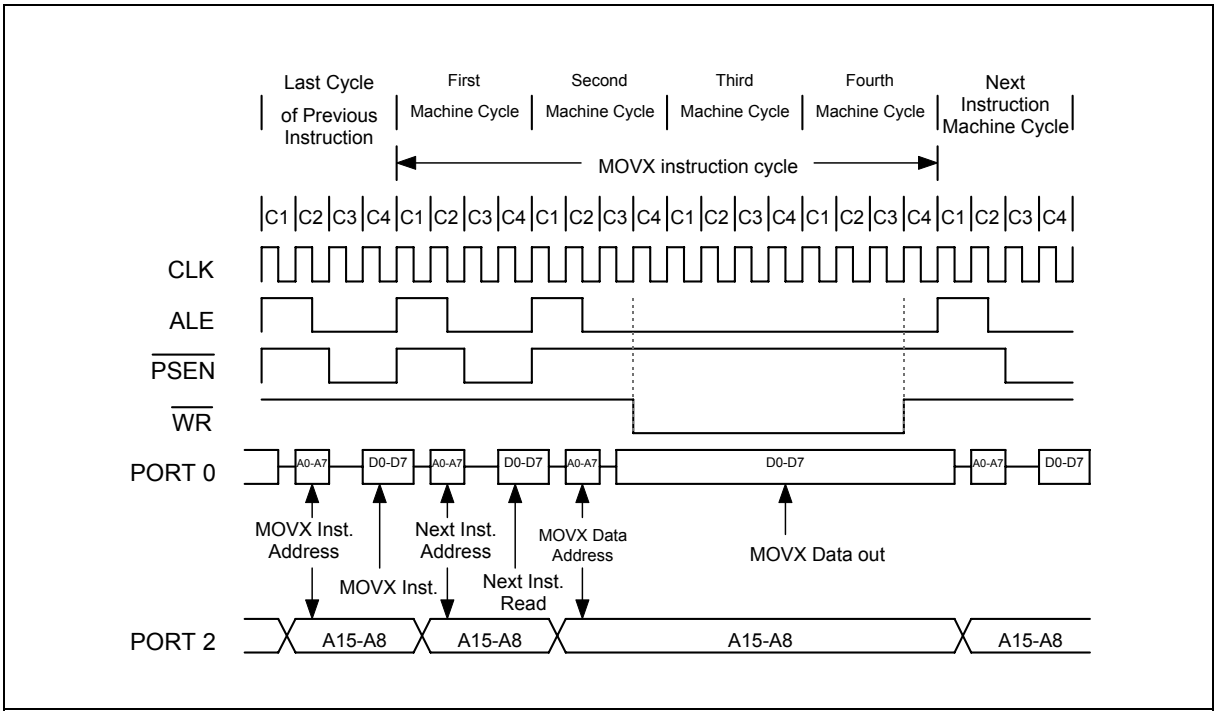



Figure 10. Data Memory Write with Stretch Value = 2

**Wait State Control Signal**

Either with the software using stretch value to change the required machine cycle of MOVX instruction, the W77E532A provides another hardware signal $\overline{\text{WAIT}}$ to implement the wider duration of external data access timing. This wait state control signal is the alternate function of P4.0 such that it can only be invoked to 44-pin PLCC/QFP package type. The wait state control signal can be enabled by setting WS (WSCON.7) bit. When enabled, the setting of stretch value decides the minimum length of MOVX instruction cycle and the device will sample the $\overline{\text{WAIT}}$ pin at each C3 state before the rising edge of read/write strobe signal during MOVX instruction. Once this signal being recognized, one more machine cycle (wait state cycle) will be inserted into next cycle. The inserted wait state cycles are unlimited, so the MOVX instruction cycle will end in which the wait state control signal is deactivated. Using wait state control signal allows a dynamically access timing to a selected external peripheral. The WS bit is accessed by the Timed Access Protection procedure.

Set WS bit and stretch value = 0 to enable wait signal.

## 9. POWER MANAGEMENT

The W77E532A has several features that help the user to control the power consumption of the device. The power saving features are basically the POWER DOWN mode, ECONOMY mode and the IDLE mode of operation.

### Idle Mode

The user can put the device into idle mode by writing 1 to the bit PCON.0. The instruction that sets the idle bit is the last instruction that will be executed before the device goes into Idle Mode. In the Idle mode, the clock to the CPU is halted, but not to the Interrupt, Timer, Watchdog timer and Serial port blocks. This forces the CPU state to be frozen; the Program counter, the Stack Pointer, the Program Status Word, the Accumulator and the other registers hold their contents. The ALE and PSEN pins are held high during the Idle state. The port pins hold the logical states they had at the time Idle was activated. The Idle mode can be terminated in two ways. Since the interrupt controller is still active, the activation of any enabled interrupt can wake up the processor. This will automatically clear the Idle bit, terminate the Idle mode, and the Interrupt Service Routine(ISR) will be executed. After the ISR, execution of the program will continue from the instruction which put the device into Idle mode.

The Idle mode can also be exited by activating the reset. The device can be put into reset either by applying a high on the external RST pin, a Power on reset condition or a Watchdog timer reset. The external reset pin has to be held high for at least two machine cycles i.e. 8 clock periods to be recognized as a valid reset. In the reset condition the program counter is reset to 0000h and all the SFRs are set to the reset condition. Since the clock is already running there is no delay and execution starts immediately. In the Idle mode, the Watchdog timer continues to run, and if enabled, a time-out will cause a watchdog timer interrupt which will wake up the device. The software must reset the Watchdog timer in order to preempt the reset which will occur after 512 clock periods of the time-out. When the W77E532A is exiting from an Idle mode with a reset, the instruction following the one which put the device into Idle mode is not executed. So there is no danger of unexpected writes.

### Economy Mode

The power consumption of microcontroller relates to operating frequency. The W77E532A offers a Economy mode to reduce the internal clock rate dynamically without external components. By default, one machine cycle needs 4 clocks. In Economy mode, software can select 4, 64 or 1024 clocks per machine cycle. It keeps the CPU operating at a acceptable speed but eliminates the power consumption. In the Idle mode, the clock of the core logic is stopped, but all clocked peripherals such as watchdog timer are still running at a rate of clock/4. In the Economy mode, all clocked peripherals run at the same reduced clocks rate as in core logic. So the Economy mode may provide a lower power consumption than idle mode.

Software invokes the Economy mode by setting the appropriate bits in the SFRs. Setting the bits CD0(PMR.6), CD1(PMR.7) decides the instruction cycle rate as below:

| CD1 | CD0 | clocks/machine cycle |
|---|---|---|
| 0 | 0 | Reserved |
| 0 | 1 | 4 (default) |
| 1 | 0 | 64 |
| 1 | 1 | 1024 |

The selection of instruction rate is going to take effect after a delay of one instruction cycle. Switching to divide by 64 or 1024 mode must first go from divide by 4 mode. This means software can not switch directly between clock/64 and clock/1024 mode. The CPU has to return clock/4 mode first, then go to clock/64 or clock/1024 mode.

In Economy mode, the serial port can not receive/transmit data correctly because the baud rate is changed. In some systems, the external interrupts may require the fastest process such that the reducing of operating speed is restricted. In order to solve these dilemmas, the W77E532A offers a switchback feature which allows the CPU back to clock/4 mode immediately when triggered by serial operation or external interrupts. The switchback feature is enabled by setting the SWB bit (PMR.5). A serial port reception/transmission or qualified external interrupt which is enabled and acknowledged without block conditions will cause CPU to return to divide by 4 mode. For the serial port reception, a switchback is generated by a falling edge associated with start bit if the serial port reception is enabled. When a serial port transmission, an instruction which writes a byte of data to serial port buffer will cause a switchback to ensure the correct transmission. The switchback feature is unaffected by serial port interrupt flags. After a switchback is generated, the software can manually return the CPU to Economy mode. Note that the modification of clock control bits CD0 and CD1 will be ignored during serial port transmit/receive when switchback is enabled. The Watchdog timer reset, power-on/fail reset or external reset will force the CPU to return to divide by 4 mode.

**Power Down Mode**

The device can be put into Power Down mode by writing 1 to bit PCON.1. The instruction that does this will be the last instruction to be executed before the device goes into Power Down mode. In the Power Down mode, all the clocks are stopped and the device comes to a halt. All activity is completely stopped and the power consumption is reduced to the lowest possible value. In this state the ALE and $\overline{\text{PSEN}}$ pins are pulled low. The port pins output the values held by their respective SFRs.

The W77E532A will exit the Power Down mode with a reset or by an external interrupt pin enabled as either level or edge detect. An external reset can be used to exit the Power down state. The high on RST pin terminates the Power Down mode, and restarts the clock. The program execution will restart from 0000h. In the Power down mode, the clock is stopped, so the Watchdog timer cannot be used to provide the reset to exit Power down mode.

The W77E532A can be woken from the Power Down mode by forcing an external interrupt pin activated, provided the corresponding interrupt is enabled, while the global enable(EA) bit is set. If these conditions are met, then the low level on the external pin re-starts the oscillator. Then device executes the interrupt service routine for the corresponding external interrupt. After the interrupt service routine is completed, the program execution returns to the instruction after the one which put the device into Power Down mode and continues from there.

**Table 5. Status of external pins during Idle and Power Down**

| MODE | PROGRAM MEMORY | ALE | $\overline{\text{PSEN}}$ | PORT0 | PORT1 | PORT2 | PORT3 |
|---|---|---|---|---|---|---|---|
| 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 |

## Reset Conditions

The user has several hardware related options for placing the W77E532A into reset condition. In general, most register bits go to their reset value irrespective of the reset condition, but there are a few flags whose state depends on the source of reset. The user can use these flags to determine the cause of reset using software. There are two ways of putting the device into reset state. They are External reset and Watchdog reset.

### External Reset

The device continuously samples the RST pin at state C4 of every machine cycle. Therefore the RST pin must be held for at least 2 machine cycles to ensure detection of a valid RST high. The reset circuitry then synchronously applies the internal reset signal. Thus the reset is a synchronous operation and requires the clock to be running to cause an external reset.

Once the device is in reset condition, it will remain so as long as RST is 1. Even after RST is deactivated, the device will continue to be in reset state for up to two machine cycles, and then begin program execution from 0000h. There is no flag associated with the external reset condition. However since the other two reset sources have flags, the external reset can be considered as the default reset if those two flags are cleared.

The software must clear the POR flag after reading it, otherwise it will not be possible to correctly determine future reset sources. If the power fails, i.e. falls below Vrst, then the device will once again go into reset state. When the power returns to the proper operating levels, the device will again perform a power on reset delay and set the POR flag.

### Watchdog Timer Reset

The Watchdog timer is a free running timer with programmable time-out intervals. The user can clear the watchdog timer at any time, causing it to restart the count. When the time-out interval is reached an interrupt flag is set. If the Watchdog reset is enabled and the watchdog timer is not cleared, then 512 clocks from the flag being set, the watchdog timer will generate a reset . This places the device into the reset condition. The reset condition is maintained by hardware for two machine cycles. Once the reset is removed the device will begin execution from 0000h.

## Reset State

Most of the SFRs and registers on the device will go to the same condition in the reset state. The Program Counter is forced to 0000h and is held there as long as the reset condition is applied. However, the reset state does not affect the on-chip RAM. The data in the RAM will be preserved during the reset. However, the stack pointer is reset to 07h, and therefore the stack contents will be lost. The RAM contents will be lost if the $V_{DD}$ falls below approximately 2V, as this is the minimum voltage level required for the RAM to operate normally. Therefore after a first time power on reset the RAM contents will be indeterminate. During a power fail condition, if the power falls below 2V, the RAM contents are lost.

After a reset most SFRs are cleared. Interrupts and Timers are disabled. The Watchdog timer is disabled if the reset source was a POR. The port SFRs have FFh written into them which puts the port pins in a high state. Port 0 floats as it does not have on-chip pull-ups.

**Table 6. SFR Reset Value**

| SFR NAME | RESET VALUE | SFR NAME | RESET VALUE |
|---|---|---|---|
| P0 | 11111111b | IE | 00000000b |
| SP | 00000111b | SADDR | 00000000b |
| DPL | 00000000b | P3 | 11111111b |
| DPH | 00000000b | IP | x0000000b |
| DPL1 | 00000000b | SADEN | 00000000b |
| DPH1 | 00000000b | T2CON | 00000000b |
| DPS | 00000000b | T2MOD | 00000x00b |
| PCON | 00xx0000b | RCAP2L | 00000000b |
| TCON | 00000000b | RCAP2H | 00000000b |
| TMOD | 00000000b | TL2 | 00000000b |
| TL0 | 00000000b | TH2 | 00000000b |
| TL1 | 00000000b | TA | 11111111b |
| TH0 | 00000000b | PSW | 00000000b |
| TH1 | 00000000b | WDCON | 0x0x0xx0b |
| CKCON | 00000001b | ACC | 00000000b |
| P1 | 11111111b | EIE | xxx00000b |
| P4CONA | 00000000b | P4CONB | 00000000b |
| P40AL | 00000000b | P40AH | 00000000b |
| P41AL | 00000000b | P41AH | 00000000b |
| P42AL | 00000000b | P42AH | 00000000b |
| P43Al | 00000000b | P43AH | 00000000b |
| CHPCON | 00000000b | P4CSIN | 00000000b |
| ROMCON | 00000111b | SFRAL | 00000000b |
| SFRAH | 00000000b | SFRFD | 00000000b |
| SFRCN | 00111111b | B | 00000000b |
| SCON | 00000000b | EIP | xxx00000b |
| SBUF | xxxxxxxxb | PC | 00000000b |
| P2 | 11111111b | SADEN1 | 00000000b |
| SADDR1 | 00000000b | SBUF1 | xxxxxxxxb |
| SCON1 | 00000000b | PMR | 010xx0x0b |
| WSCON | 00000000b | STATUS | 000x0000b |
| EXIF | 0000xxx0b | | |
| P4 | xxxx1111b | | |
| | | | |

The WDCON SFR bits are set/cleared in reset condition depending on the source of the reset.

| | External reset | Watchdog reset | Power on reset |
|---|---|---|---|
| WDCON | 0x0x0xx0b | 0x0x01x0b | 01000000b |

The POR bit WDCON.6 is set only by the power on reset. The PFI bit WDCON.4 is set when the power fail condition occurs. However, a power-on reset will clear this bit. The WTRF bit WDCON.2 is set when the Watchdog timer causes a reset. A power on reset will also clear this bit. The EWT bit WDCON.1 is cleared by power on resets. This disables the Watchdog timer resets. A watchdog or external reset does not affect the EWT bit.

## 10. INTERRUPTS

The W77E532A has a two priority level interrupt structure with 12 interrupt sources. Each of the interrupt sources has an individual priority bit, flag, interrupt vector and enable bit. In addition, the interrupts can be globally enabled or disabled.

### Interrupt Sources

The External Interrupts $\overline{INT0}$ and $\overline{INT1}$ can be either edge triggered or level triggered, depending on bits IT0 and IT1. The bits IE0 and IE1 in the TCON register are the flags which are checked to generate the interrupt. In the edge triggered mode, the INTx inputs are sampled in every machine cycle. If the sample is high in one cycle and low in the next, then a high to low transition is detected and the interrupts request flag Iex in TCON or EXIF is set. The flag bit requests the interrupt. Since the external interrupts are sampled every machine cycle, they have to be held high or low for at least one complete machine cycle. The Iex flag is automatically cleared when the service routine is called. If the level triggered mode is selected, then the requesting source has to hold the pin low till the interrupt is serviced. The Iex flag will not be cleared by the hardware on entering the service routine. If the interrupt continues to be held low even after the service routine is completed, then the processor may acknowledge another interrupt request from the same source. Note that the external interrupts INT2 to $\overline{INT5}$ are edge triggered only. By default, the individual interrupt flag corresponding to external interrupt 2 to 5 must be cleared manually by software. It can be configured with hardware cleared by setting the corresponding bit HCx in the T2MOD register. For instance, if HC2 is set hardware will clear IE2 flag after program enters the interrupt 2 service routine.

The Timer 0 and 1 Interrupts are generated by the TF0 and TF1 flags. These flags are set by the overflow in the Timer 0 and Timer 1. The TF0 and TF1 flags are automatically cleared by the hardware when the timer interrupt is serviced. The Timer 2 interrupt is generated by a logical OR of the TF2 and the EXF2 flags. These flags are set by overflow or capture/reload events in the timer 2 operation. The hardware does not clear these flags when a timer 2 interrupt is executed. Software has to resolve the cause of the interrupt between TF2 and EXF2 and clear the appropriate flag.

The Watchdog timer can be used as a system monitor or a simple timer. In either case, when the time-out count is reached, the Watchdog timer interrupt flag WDIF (WDCON.3) is set. If the interrupt is enabled by the enable bit EIE.4, then an interrupt will occur.

The Serial block can generate interrupts on reception or transmission. There are two interrupt sources from the Serial block, which are obtained by the RI and TI bits in the SCON SFR and RI_1 and TI_1 in the SCON1 SFR. These bits are not automatically cleared by the hardware, and the user will have to clear these bits using software.

All the bits that generate interrupts can be set or reset by hardware, and thereby software initiated interrupts can be generated. Each of the individual interrupts can be enabled or disabled by setting or clearing a bit in the IE SFR. IE also has a global enable/disable bit EA, which can be cleared to disable all the interrupts, except PFI, at once.

### Priority Level Structure

There are three priority levels for the interrupts, highest, high and low. The interrupt sources can be individually set to either high or low levels. Naturally, a higher priority interrupt cannot be interrupted by a lower priority interrupt. However there exists a pre-defined hierarchy amongst the interrupts themselves. This hierarchy comes into play when the interrupt controller has to resolve simultaneous requests having the same priority level. This hierarchy is defined as shown below; the interrupts are numbered starting from the highest priority to the lowest.

**Table 7. Priority structure of interrupts**

| SOURCE | FLAG | PRIORITY LEVEL |
|---|---|---|
| External Interrupt 0 | IE0 | 1(highest) |
| Timer 0 Overflow | TF0 | 2 |
| External Interrupt 1 | IE1 | 3 |
| Timer 1 Overflow | TF1 | 4 |
| Serial Port | RI + TI | 5 |
| Timer 2 Overflow | TF2 + EXF2 | 6 |
| Serial Port 1 | RI_1 + TI_1 | 7 |
| External Interrupt 2 | IE2 | 8 |
| External Interrupt 3 | IE3 | 9 |
| External Interrupt 4 | IE4 | 10 |
| External Interrupt 5 | IE5 | 11 |
| Watchdog Timer | WDIF | 12 (lowest) |

The interrupt flags are sampled every machine cycle. In the same machine cycle, the sampled interrupts are polled and their priority is resolved. If certain conditions are met then the hardware will execute an internally generated LCALL instruction which will vector the process to the appropriate interrupt vector address. The conditions for generating the LCALL are

1. An interrupt of equal or higher priority is not currently being serviced.
2. The current polling cycle is the last machine cycle of the instruction currently being executed.
3. The current instruction does not involve a write to IP, IE, EIP or EIE registers and is not a RETI.

If any of these conditions are not met, then the LCALL will not be generated. The polling cycle is repeated every machine cycle, with the interrupts sampled in the same machine cycle. If an interrupt flag is active in one cycle but not responded to, and is not active when the above conditions are met, the denied interrupt will not be serviced. This means that active interrupts are not remembered; every polling cycle is new.

The processor responds to a valid interrupt by executing an LCALL instruction to the appropriate service routine. This may or may not clear the flag which caused the interrupt. In case of Timer interrupts, the TF0 or TF1 flags are cleared by hardware whenever the processor vectors to the appropriate timer service routine. In case of external interrupt, INT0 and INT1, the flags are cleared only if they are edge triggered. In case of Serial interrupts, the flags are not cleared by hardware. In the case of Timer 2 interrupt, the flags are not cleared by hardware. The Watchdog timer interrupt flag WDIF has to be cleared by software. The hardware LCALL behaves exactly like the software LCALL instruction. This instruction saves the Program Counter contents onto the Stack, but does not save the Program Status Word PSW. The PC is reloaded with the vector address of that interrupt which caused the LCALL. These vector address for the different sources are as follows

**Table 8. Vector locations for interrupt sources**

| SOURCE | VECTOR ADDRESS | SOURCE | VECTOR ADDRESS |
|---|---|---|---|
| Timer 0 Overflow | 000Bh | External Interrupt 0 | 0003h |
| Timer 1 Overflow | 001Bh | External Interrupt 1 | 0013h |
| Timer 2 Interrupt | 002Bh | Serial Port | 0023h |
| External Interrupt 2 | 0043h | Serial Port 1 | 003Bh |
| External Interrupt 4 | 0053h | External Interrupt 3 | 004Bh |
| Watchdog Timer | 0063h | External Interrupt 5 | 005Bh |

The vector table is not evenly spaced; this is to accommodate future expansions to the device family.

Execution continues from the vectored address till an RETI instruction is executed. On execution of the RETI instruction the processor pops the Stack and loads the PC with the contents at the top of the stack. The user must take care that the status of the stack is restored to what is was after the hardware LCALL, if the execution is to return to the interrupted program. The processor does not notice anything if the stack contents are modified and will proceed with execution from the address put back into PC. Note that a RET instruction would perform exactly the same process as a RETI instruction, but it would not inform the Interrupt Controller that the interrupt service routine is completed, and would leave the controller still thinking that the service routine is underway.

**Interrupt Response Time**

The response time for each interrupt source depends on several factors, such as the nature of the interrupt and the instruction underway. In the case of external interrupts $\overline{INT0}$ to $\overline{INT5}$, they are sampled at C3 of every machine cycle and then their corresponding interrupt flags Iex will be set or reset. The Timer 0 and 1 overflow flags are set at C3 of the machine cycle in which overflow has occurred. These flag values are polled only in the next machine cycle. If a request is active and all three conditions are met, then the hardware generated LCALL is executed. This LCALL itself takes four machine cycles to be completed. Thus there is a minimum time of five machine cycles between the interrupt flag being set and the interrupt service routine being executed.

A longer response time should be anticipated if any of the three conditions are not met. If a higher or equal priority is being serviced, then the interrupt latency time obviously depends on the nature of the service routine currently being executed. If the polling cycle is not the last machine cycle of the instruction being executed, then an additional delay is introduced. The maximum response time (if no other interrupt is in service) occurs if the W77E532A is performing a write to IE, IP, EIE or EIP and then executes a MUL or DIV instruction. From the time an interrupt source is activated, the longest reaction time is 12 machine cycles. This includes 1 machine cycle to detect the interrupt, 2 machine cycles to complete the IE, IP, EIE or EIP access, 5 machine cycles to complete the MUL or DIV instruction and 4 machine cycles to complete the hardware LCALL to the interrupt vector location.

Thus in a single-interrupt system the interrupt response time will always be more than 5 machine cycles and not more than 12 machine cycles. The maximum latency of 12 machine cycle is 48 clock cycles. Note that in the standard 8051 the maximum latency is 8 machine cycles which equals 96 machine cycles. This is a 50% reduction in terms of clock periods.

## 11. PROGRAMMABLE TIMERS/COUNTERS

The W77E532A has three 16-bit programmable timer/counters and one programmable Watchdog timer. The Watchdog timer is operationally quite different from the other two timers.

**Timer/Counters 0 & 1**

The W77E532A has two 16-bit Timer/Counters. Each of these Timer/Counters has two 8 bit registers which form the 16 bit counting register. For Timer/Counter 0 they are TH0, the upper 8 bits register, and TL0, the lower 8 bit register. Similarly Timer/Counter 1 has two 8 bit registers, TH1 and TL1. The two can be configured to operate either as timers, counting machine cycles or as counters counting external inputs.

When configured as a “Timer”, the timer counts clock cycles. The timer clock can be programmed to be thought of as 1/12 of the system clock or 1/4 of the system clock. In the “Counter” mode, the register is incremented on the falling edge of the external input pin, T0 in case of Timer 0, and T1 for Timer 1. The T0 and T1 inputs are sampled in every machine cycle at C4. If the sampled value is high in one machine cycle and low in the next, then a valid high to low transition on the pin is recognized and the count register is incremented. Since it takes two machine cycles to recognize a negative transition on the pin, the maximum rate at which counting will take place is 1/24 of the master clock frequency. In either the “Timer” or “Counter” mode, the count register will be updated at C3. Therefore, in the “Timer” mode, the recognized negative transition on pin T0 and T1 can cause the count register value to be updated only in the machine cycle following the one in which the negative edge was detected.

The “Timer” or “Counter” function is selected by the “ $C/\overline{T}$ ” bit in the TMOD Special Function Register. Each Timer/Counter has one selection bit for its own; bit 2 of TMOD selects the function for Timer/Counter 0 and bit 6 of TMOD selects the function for Timer/Counter 1. In addition each Timer/Counter can be set to operate in any one of four possible modes. The mode selection is done by bits M0 and M1 in the TMOD SFR.

**Time-Base Selection**

The W77E532A gives the user two modes of operation for the timer. The timers can be programmed to operate like the standard 8051 family, counting at the rate of 1/12 of the clock speed. This will ensure that timing loops on the W77E532A and the standard 8051 can be matched. This is the default mode of operation of the W77E532A timers. The user also has the option to count in the turbo mode, where the timers will increment at the rate of 1/4 clock speed. This will straight-away increase the counting speed three times. This selection is done by the T0M and T1M bits in CKCON SFR. A reset sets these bits to 0, and the timers then operate in the standard 8051 mode. The user should set these bits to 1 if the timers are to operate in turbo mode.

**Mode 0**

In Mode 0, the timer/counters act as a 8 bit counter with a 5 bit, divide by 32 pre-scale. In this mode we have a 13 bit timer/counter. The 13 bit counter consists of 8 bits of THx and 5 lower bits of TLx. The upper 3 bits of TLx are ignored.

The negative edge of the clock increments the count in the TLx register. When the fifth bit in TLx moves from 1 to 0, then the count in the THx register is incremented. When the count in THx moves from FFh to 00h, then the overflow flag TFx in TCON SFR is set. The counted input is enabled only if TRx is set and either GATE = 0 or $\overline{INTx}$ = 1. When $C/\overline{T}$ is set to 0, then it will count clock cycles, and if $C/\overline{T}$ is set to 1, then it will count 1 to 0 transitions on T0 (P3.4) for timer 0 and T1 (P3.5) for timer 1. When the 13 bit count reaches 1FFFh the next count will cause it to roll-over to 0000h. The timer overflow flag TFx of the relevant timer is set and if enabled an interrupts will occur. Note that

when used as a timer, the time-base may be either clock cycles/12 or clock cycles/4 as selected by the bits TxM of the CKCON SFR.

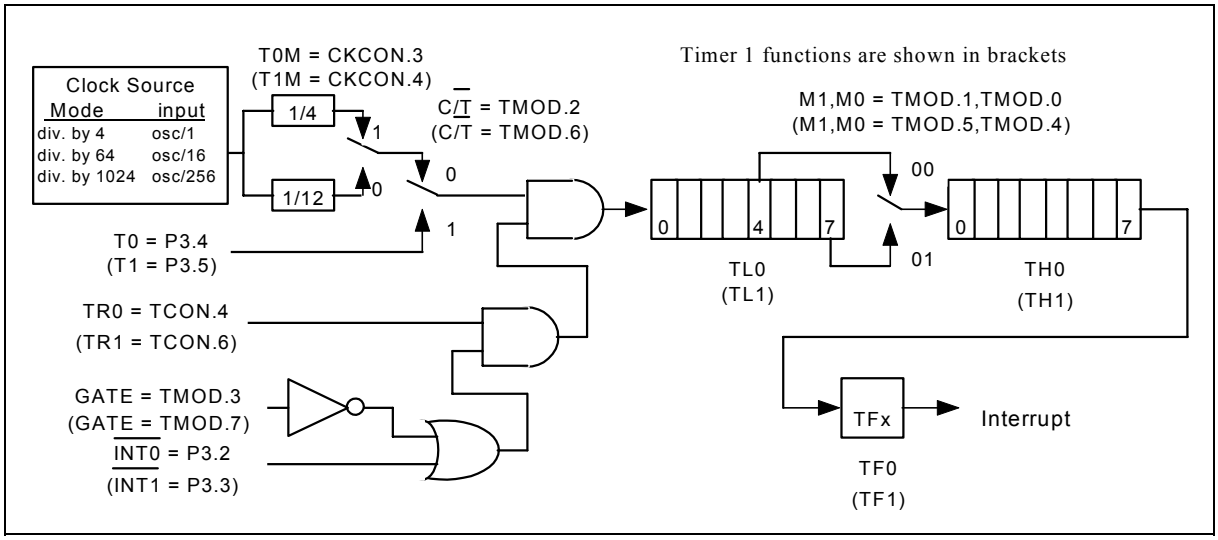



Figure 11. Timer/Counter Mode 0 & Mode 1

### Mode 1

Mode 1 is similar to Mode 0 except that the counting register forms a 16 bit counter, rather than a 13 bit counter. This means that all the bits of THx and TLx are used. Roll-over occurs when the timer moves from a count of FFFFh to 0000h. The timer overflow flag TFx of the relevant timer is set and if enabled an interrupt will occur. The selection of the time-base in the timer mode is similar to that in Mode 0. The gate function operates similarly to that in Mode 0.

### Mode 2

In Mode 2, the timer/counter is in the Auto Reload Mode. In this mode, TLx acts as a 8 bit count register, while THx holds the reload value. When the TLx register overflows from FFh to 00h, the TFx bit in TCON is set and TLx is reloaded with the contents of THx, and the counting process continues from here. The reload operation leaves the contents of the THx register unchanged. Counting is enabled by the TRx bit and proper setting of GATE and $\overline{INTx}$ pins. As in the other two modes 0 and 1 mode 2 allows counting of either clock cycles (clock/12 or clock/4) or pulses on pin Tn.

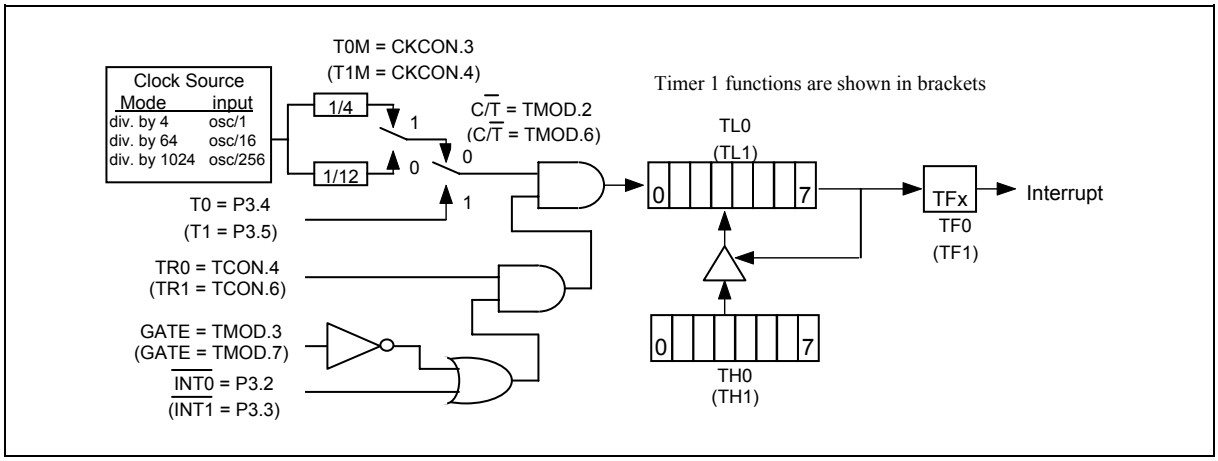



Figure 12. Timer/Counter Mode 2.

### Mode 3

Mode 3 has different operating methods for the two timer/counters. For timer/counter 1, mode 3 simply freezes the counter. Timer/Counter 0, however, configures TL0 and TH0 as two separate 8 bit count registers in this mode. The logic for this mode is shown in the figure. TL0 uses the Timer/Counter 0 control bits C / $\overline{T}$, GATE, TR0, $\overline{INT0}$ and TF0. The TL0 can be used to count clock cycles (clock/12 or clock/4) or 1-to-0 transitions on pin T0 as determined by C/T (TMOD.2). TH0 is forced as a clock cycle counter (clock/12 or clock/4) and takes over the use of TR1 and TF1 from Timer/Counter 1. Mode 3 is used in cases where an extra 8 bit timer is needed. With Timer 0 in Mode 3, Timer 1 can still be used in Modes 0, 1 and 2., but its flexibility is somewhat limited. While its basic functionality is maintained, it no longer has control over its overflow flag TF1 and the enable bit TR1. Timer 1 can still be used as a timer/counter and retains the use of GATE and INT1 pin. In this condition it can be turned on and off by switching it out of and into its own Mode 3. It can also be used as a baud rate generator for the serial port.

### Timer/Counter 2

Timer/Counter 2 is a 16 bit up/down counter which is configured by the T2MOD register and controlled by the T2CON register. Timer/Counter 2 is equipped with a capture/reload capability. As with the Timer 0 and Timer 1 counters, there exists considerable flexibility in selecting and controlling the clock, and in defining the operating mode. The clock source for Timer/Counter 2 may be selected for either the external T2 pin (C/T2 = 1) or the crystal oscillator, which is divided by 12 or 4 (C/T2 = 0). The clock is then enabled when TR2 is a 1, and disabled when TR2 is a 0.

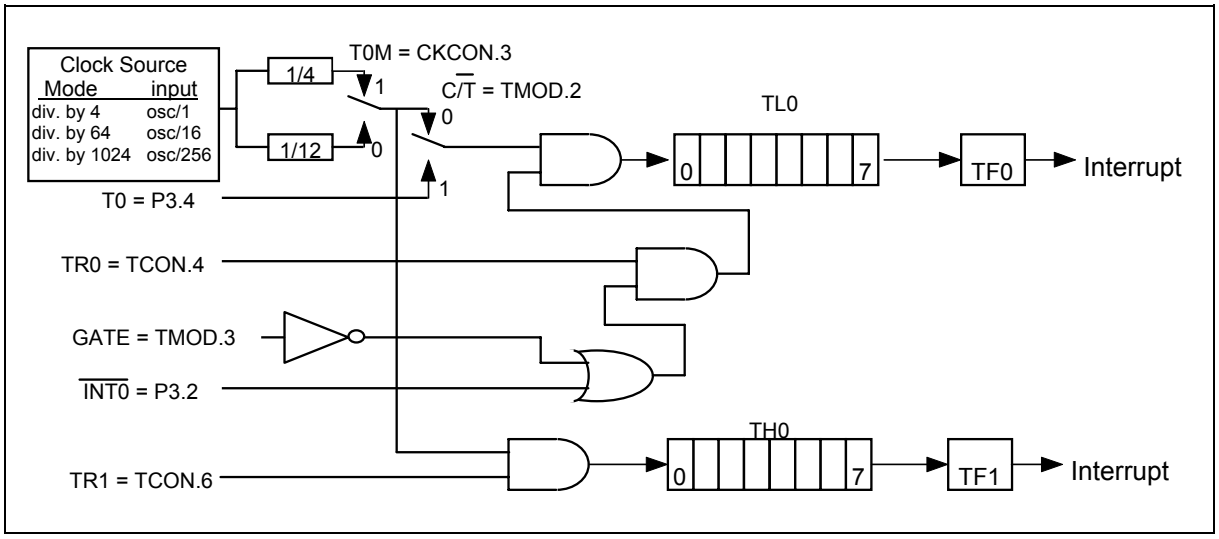



Figure 13. Timer/Counter 0 Mode 3

### Capture Mode

The capture mode is enabled by setting the CP / $\overline{RL2}$ bit in the T2CON register to a 1. In the capture mode, Timer/Counter 2 serves as a 16 bit up counter. When the counter rolls over from FFFFh to 0000h, the TF2 bit is set, which will generate an interrupt request. If the EXEN2 bit is set, then a negative transition of T2EX pin will cause the value in the TL2 and TH2 register to be captured by the RCAP2L and RCAP2H registers. This action also causes the EXF2 bit in T2CON to be set, which will also generate an interrupt. Setting the T2CR bit (T2MOD.3), the W77E532A allows hardware to reset timer 2 automatically after the value of TL2 and TH2 have been captured.

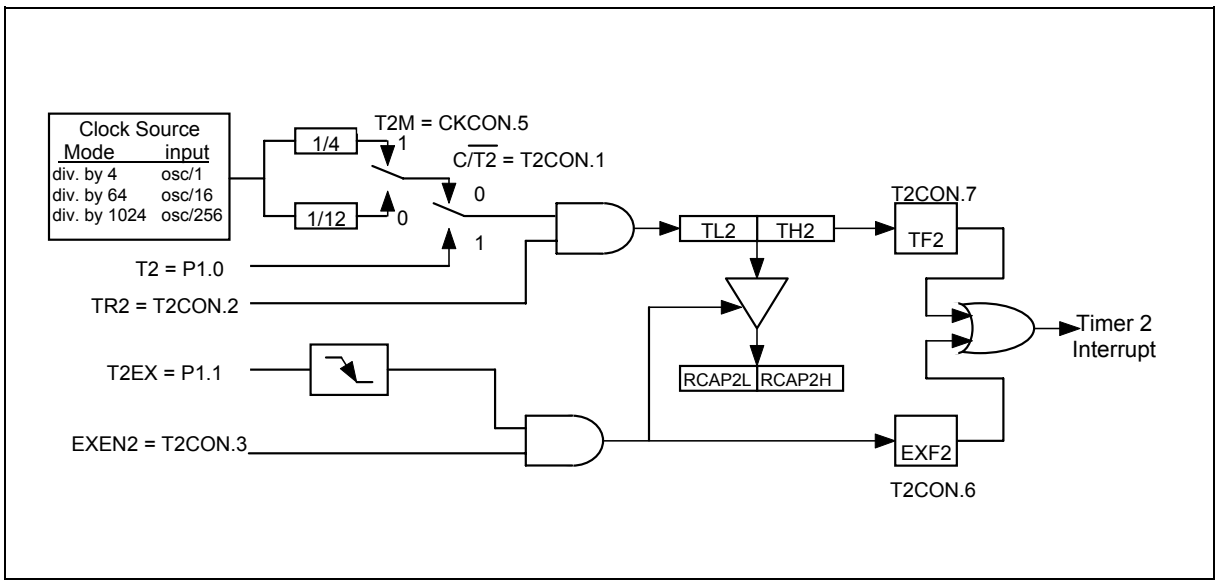



Figure 14. 16-Bit Capture Mode

### Auto-reload Mode, Counting up

The auto-reload mode as an up counter is enabled by clearing the CP / $\overline{RL2}$ bit in the T2CON register and clearing the DCEN bit in T2MOD register. In this mode, Timer/Counter 2 is a 16 bit up counter. When the counter rolls over from FFFFh, a reload is generated that causes the contents of the RCAP2L and RCAP2H registers to be reloaded into the TL2 and TH2 registers. The reload action also sets the TF2 bit. If the EXEN2 bit is set, then a negative transition of T2EX pin will also cause a reload. This action also sets the EXF2 bit in T2CON.

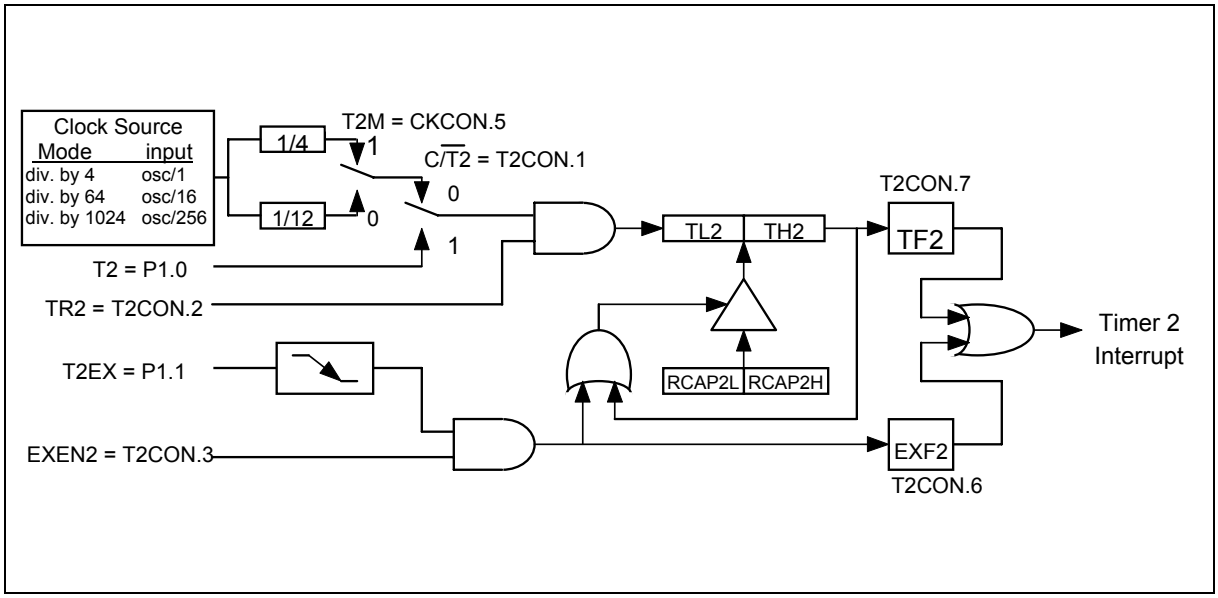



Figure 15. 16-Bit Auto-reload Mode, Counting Up

### Auto-reload Mode, Counting Up/Down

Timer/Counter 2 will be in auto-reload mode as an up/down counter if CP / $\overline{RL2}$ bit in T2CON is cleared and the DCEN bit in T2MOD is set. In this mode, Timer/Counter 2 is an up/down counter whose direction is controlled by the T2EX pin. A 1 on this pin cause the counter to count up. An overflow while counting up will cause the counter to be reloaded with the contents of the capture registers. The next down count following the case where the contents of Timer/Counter equal the capture registers will load an FFFFh into Timer/Counter 2. In either event a reload will set the TF2 bit. A reload will also toggle the EXF2 bit. However, the EXF2 bit can not generate an interrupt while in this mode.

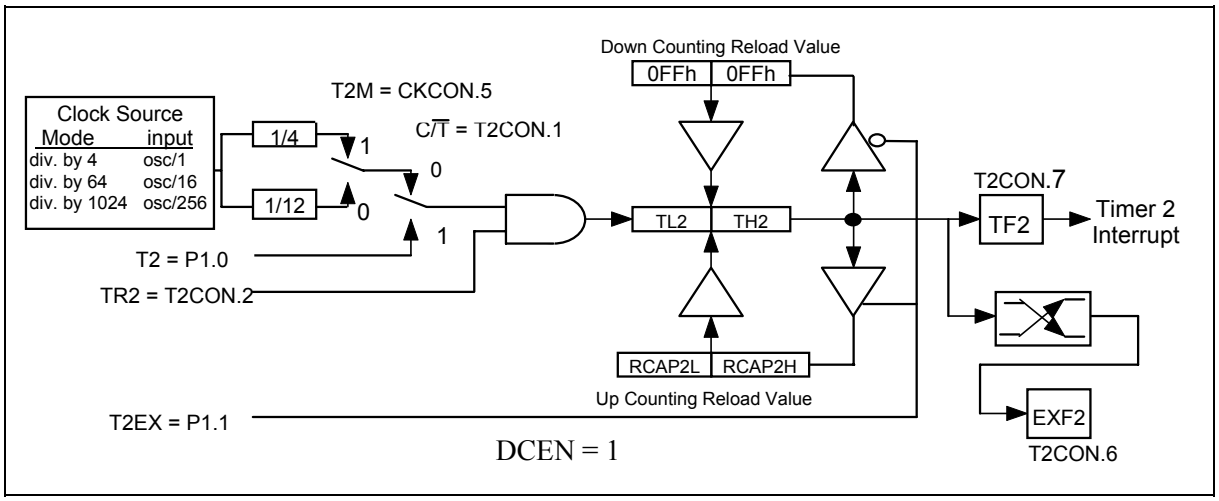



Figure 16. 16-Bit Auto-reload Up/Down Counter

### Baud Rate Generator Mode

The baud rate generator mode is enabled by setting either the RCLK or TCLK bits in T2CON register. While in the baud rate generator mode, Timer/Counter 2 is a 16 bit counter with auto reload when the count rolls over from FFFFh. However, rolling over does not set the TF2 bit. If EXEN2 bit is set, then a negative transition of the T2EX pin will set EXF2 bit in the T2CON register and cause an interrupt request.

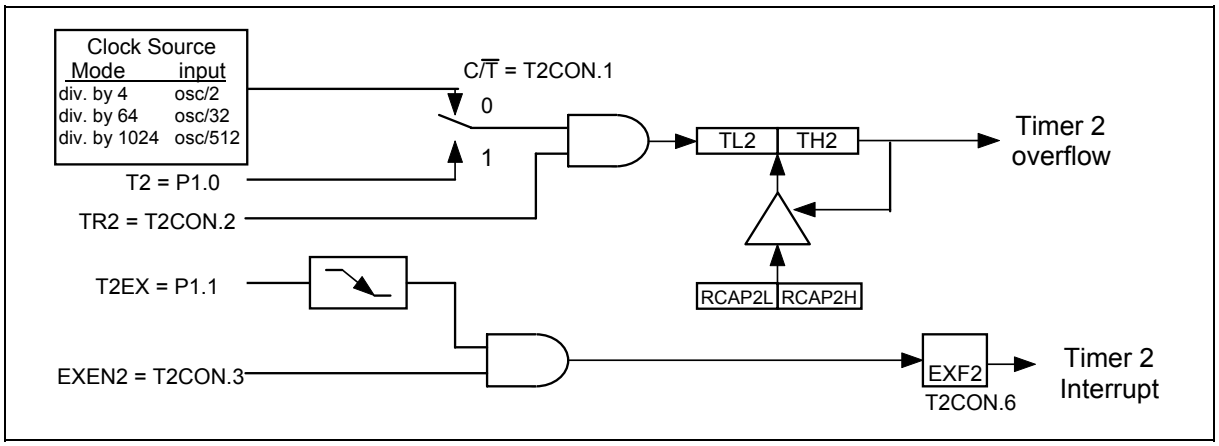



Figure 17. Baud Rate Generator Mode

### Programmable Clock-out

Timer 2 is equipped with a new clock-out feature which outputs a 50% duty cycle clock on P1.0. It can be invoked as a programmable clock generator. To configure Timer 2 with clock-out mode, software must initiate it by setting bit T2OE = 1, C/T2 = 0 and CP/RL = 0. Setting bit TR2 will start the timer. This mode is similar to the baud rate generator mode, it will not generate an interrupt while Timer 2 overflow. So it is possible to use Timer 2 as a baud rate generator and a clock generator at the same time. The clock-out frequency is determined by the following equation:

**The Clock-Out Frequency = Oscillator Frequency / [4 x (65536-RCAP2H, RCAP2L)]**

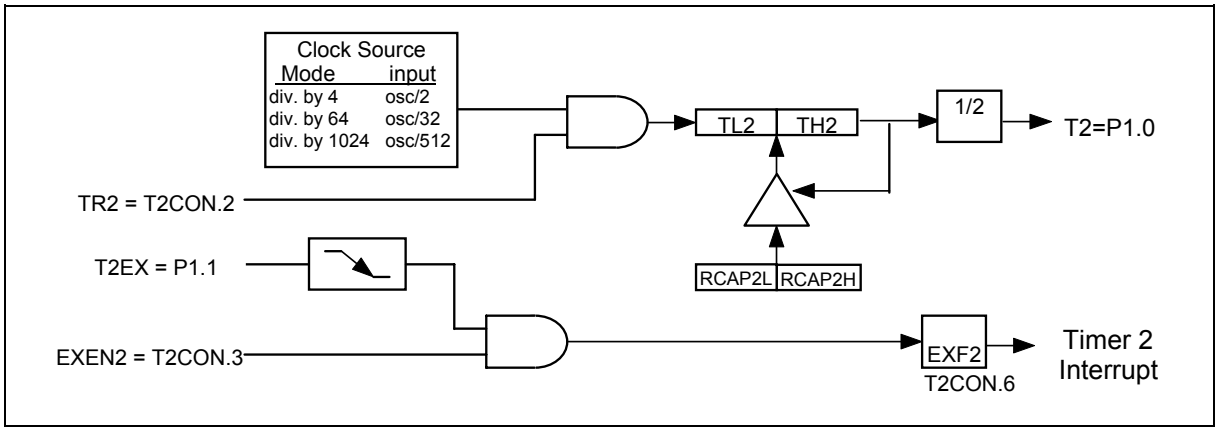



Figure 18. Programmable Clock-Out Mode

### Watchdog Timer

The Watchdog timer is a free-running timer which can be programmed by the user to serve as a system monitor, a time-base generator or an event timer. It is basically a set of dividers that divide the system clock. The divider output is selectable and determines the time-out interval. When the time-out occurs a flag is set, which can cause an interrupt if enabled, and a system reset can also be caused if it is enabled. The interrupt will occur if the individual interrupt enable and the global enable are set. The interrupt and reset functions are independent of each other and may be used separately or together depending on the users software.

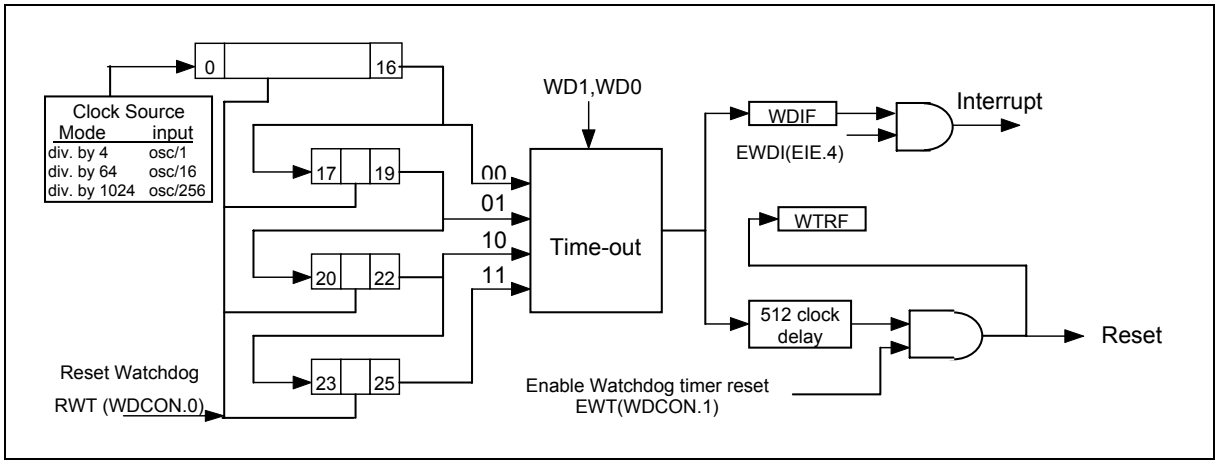



Figure 19. Watchdog Timer

The Watchdog timer should first be restarted by using RWT. This ensures that the timer starts from a known state. The RWT bit is used to restart the watchdog timer. This bit is self clearing, i.e. after writing a 1 to this bit the software will automatically clear it. The watchdog timer will now count clock cycles. The time-out interval is selected by the two bits WD1 and WD0 (CKCON.7 and CKCON.6). When the selected time-out occurs, the Watchdog interrupt flag WDIF (WDCON.3) is set. After the time-out has occurred, the watchdog timer waits for an additional 512 clock cycles. If the Watchdog Reset EWT (WDCON.1) is enabled, then 512 clocks after the time-out, if there is no RWT, a system reset due to Watchdog timer will occur. This will last for two machine cycles, and the Watchdog timer reset flag WTRF (WDCON.2) will be set. This indicates to the software that the watchdog was the cause of the reset.

When used as a simple timer, the reset and interrupt functions are disabled. The timer will set the WDIF flag each time the timer completes the selected time interval. The WDIF flag is polled to detect a time-out and the RWT allows software to restart the timer. The Watchdog timer can also be used as a very long timer. The interrupt feature is enabled in this case. Every time the time-out occurs an interrupt will occur if the global interrupt enable EA is set.

The main use of the Watchdog timer is as a system monitor. This is important in real-time control applications. In case of some power glitches or electro-magnetic interference, the processor may begin to execute errant code. If this is left unchecked the entire system may crash. Using the watchdog timer interrupt during software development will allow the user to select ideal watchdog reset locations. The code is first written without the watchdog interrupt or reset. Then the watchdog interrupt is enabled to identify code locations where interrupt occurs. The user can now insert instructions to reset the watchdog timer which will allow the code to run without any watchdog timer interrupts. Now the watchdog timer reset is enabled and the watchdog interrupt may be disabled. If any errant code is executed now, then the reset watchdog timer instructions will not be executed at the required instants and watchdog reset will occur.

The watchdog time-out selection will result in different time-out values depending on the clock speed. The reset, when enabled, will occur 512 clocks after the time-out has occurred.

**Table 9. Time-out values for the Watchdog timer**

| WD1 | WD0 | WATCHDOG INTERVAL | NUMBER OF CLOCKS | TIME @ 1.8432 MHZ | TIME @ 10 MHZ | TIME @ 25 MHZ |
|---|---|---|---|---|---|---|
| 0 | 0 | $2^{17}$ | 131072 | 71.11 mS | 13.11 mS | 5.24 mS |
| 0 | 1 | $2^{20}$ | 1048576 | 568.89 mS | 104.86 mS | 41.94 mS |
| 1 | 0 | $2^{23}$ | 8388608 | 4551.11 mS | 838.86 mS | 335.54 mS |
| 1 | 1 | $2^{26}$ | 67108864 | 36408.88 mS | 6710.89 mS | 2684.35 mS |

The Watchdog timer will de disabled by a power-on/fail reset. The Watchdog timer reset does not disable the watchdog timer, but will restart it. In general, software should restart the timer to put it into a known state.

The control bits that support the Watchdog timer are discussed below.

**WATCHDOG CONTROL**

WDIF: WDCON.3 – Watchdog Timer Interrupt flag. This bit is set whenever the time-out occurs in the watchdog timer. If the Watchdog interrupt is enabled (EIE.4), then an interrupt will occur (if the global interrupt enable is set and other interrupt requirements are met). Software or any reset can clear this bit.

WTRF: WDCON.2 – Watchdog Timer Reset flag. This bit is set whenever a watchdog reset occurs. This bit is useful for determined the cause of a reset. Software must read it, and clear it manually. A Power-fail reset will clear this bit. If EWT = 0, then this bit will not be affected by the watchdog timer.

EWT: WDCON.1 – Enable Watchdog timer Reset. This bit when set to 1 will enable the Watchdog timer reset function. Setting this bit to 0 will disable the Watchdog timer reset function, but will leave the timer running.

RWT: WDCON.0 – Reset Watchdog Timer. This bit is used to clear the Watchdog timer and to restart it. This bit is self-clearing, so after the software writes 1 to it the hardware will automatically clear it. If the Watchdog timer reset is enabled, then the RWT has to be set by the user within 512 clocks of the time-out. If this is not done then a Watchdog timer reset will occur.

**Clock Control**

WD1, WD0: CKCON.7, CKCON.6 – Watchdog Timer Mode select bits. These two bits select the time-out interval for the watchdog timer. The reset time is 512 clock longer than the interrupt time-out value.

The default Watchdog time-out is $2^{17}$ clocks, which is the shortest time-out period. The EWT, WDIF and RWT bits are protected by the Timed Access procedure. This prevents software from accidentally enabling or disabling the watchdog timer. More importantly, it makes it highly improbable that errant code can enable or disable the watchdog timer.

## 12. SERIAL PORT

Serial port in the W77E532A is a full duplex port. The W77E532A provides the user with additional features such as the Frame Error Detection and the Automatic Address Recognition. The serial ports are capable of synchronous as well as asynchronous communication. In Synchronous mode the W77E532A generates the clock and operates in a half duplex mode. In the asynchronous mode, full duplex operation is available. This means that it can simultaneously transmit and receive data. The transmit register and the receive buffer are both addressed as SBUF Special Function Register. However any write to SBUF will be to the transmit register, while a read from SBUF will be from the receive buffer register. The serial port can operate in four different modes as described below.

### Mode 0

This mode provides synchronous communication with external devices. In this mode serial data is transmitted and received on the RXD line. TXD is used to transmit the shift clock. The TxD clock is provided by the W77E532A whether the device is transmitting or receiving. This mode is therefore a half duplex mode of serial communication. In this mode, 8 bits are transmitted or received per frame. The LSB is transmitted/received first. The baud rate is fixed at 1/12 or 1/4 of the oscillator frequency. This baud rate is determined by the SM2 bit (SCON.5). When this bit is set to 0, then the serial port runs at 1/12 of the clock. When set to 1, the serial port runs at 1/4 of the clock. This additional facility of programmable baud rate in mode 0 is the only difference between the standard 8051 and the W77E532A.

The functional block diagram is shown below. Data enters and leaves the Serial port on the RxD line. The TxD line is used to output the shift clock. The shift clock is used to shift data into and out of the W77E532A and the device at the other end of the line. Any instruction that causes a write to SBUF will start the transmission. The shift clock will be activated and data will be shifted out on the RxD pin till all 8 bits are transmitted. If SM2 = 1, then the data on RxD will appear 1 clock period before the falling edge of shift clock on TxD. The clock on TxD then remains low for 2 clock periods, and then goes high again. If SM2 = 0, the data on RxD will appear 3 clock periods before the falling edge of shift clock on TxD. The clock on TxD then remains low for 6 clock periods, and then goes high again. This ensures that at the receiving end the data on RxD line can either be clocked on the rising edge of the shift clock on TxD or latched when the TxD clock is low.

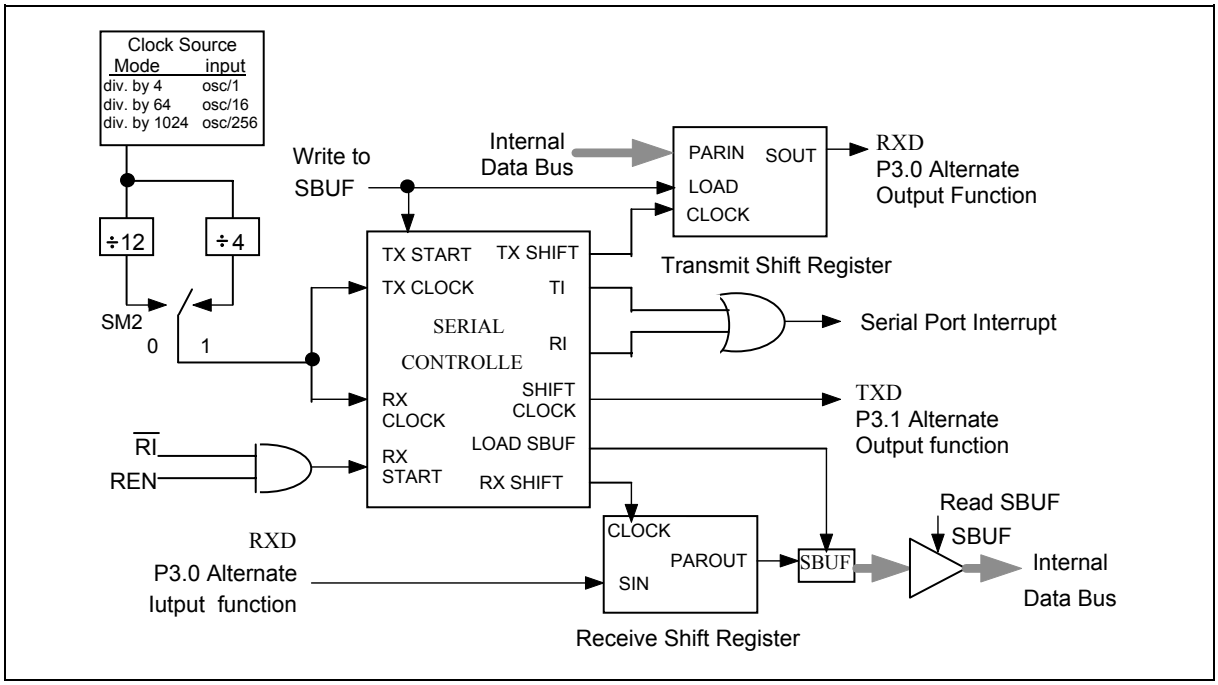



Figure 20. Serial Port Mode 0

The TI flag is set high in C1 following the end of transmission of the last bit. The serial port will receive data when REN is 1 and RI is zero. The shift clock (TxD) will be activated and the serial port will latch data on the rising edge of shift clock. The external device should therefore present data on the falling edge on the shift clock. This process continues till all the 8 bits have been received. The RI flag is set in C1 following the last rising edge of the shift clock on TxD. This will stop reception, till the RI is cleared by software.

**Mode 1**

In Mode 1, the full duplex asynchronous mode is used. Serial communication frames are made up of 10 bits transmitted on TXD and received on RXD. The 10 bits consist of a start bit (0), 8 data bits (LSB first), and a stop bit (1). On receive, the stop bit goes into RB8 in the SFR SCON. The baud rate in this mode is variable. The serial baud can be programmed to be 1/16 or 1/32 of the Timer 1 overflow. Since the Timer 1 can be set to different reload values, a wide variation in baud rates is possible.

Transmission begins with a write to SBUF. The serial data is brought out on to TxD pin at C1 following the first roll-over of divide by 16 counter. The next bit is placed on TxD pin at C1 following the next rollover of the divide by 16 counter. Thus the transmission is synchronized to the divide by 16 counter and not directly to the write to SBUF signal. After all 8 bits of data are transmitted, the stop bit is transmitted. The TI flag is set in the C1 state after the stop bit has been put out on TxD pin. This will be at the $10^{th}$ rollover of the divide by 16 counter after a write to SBUF.

Reception is enabled only if REN is high. The serial port actually starts the receiving of serial data, with the detection of a falling edge on the RxD pin. The 1-to-0 detector continuously monitors the RxD line, sampling it at the rate of 16 times the selected baud rate. When a falling edge is detected, the divide by 16 counter is immediately reset. This helps to align the bit boundaries with the rollovers of the divide by 16 counter.

The 16 states of the counter effectively divide the bit time into 16 slices. The bit detection is done on a best of three basis. The bit detector samples the RxD pin, at the 8th, 9th and 10th counter states. By using a majority 2 of 3 voting system, the bit value is selected. This is done to improve the noise rejection feature of the serial port. If the first bit detected after the falling edge of RxD pin is not 0, then this indicates an invalid start bit, and the reception is immediately aborted. The serial port again looks for a falling edge in the RxD line. If a valid start bit is detected, then the rest of the bits are also detected and shifted into the SBUF.

After shifting in 8 data bits, there is one more shift to do, after which the SBUF and RB8 are loaded and RI is set. However certain conditions must be met before the loading and setting of RI can be done.

1. RI must be 0 and
2. Either SM2 = 0, or the received stop bit = 1.

If these conditions are met, then the stop bit goes to RB8, the 8 data bits go into SBUF and RI is set. Otherwise the received frame may be lost. After the middle of the stop bit, the receiver goes back to looking for a 1-to-0 transition on the RxD pin.

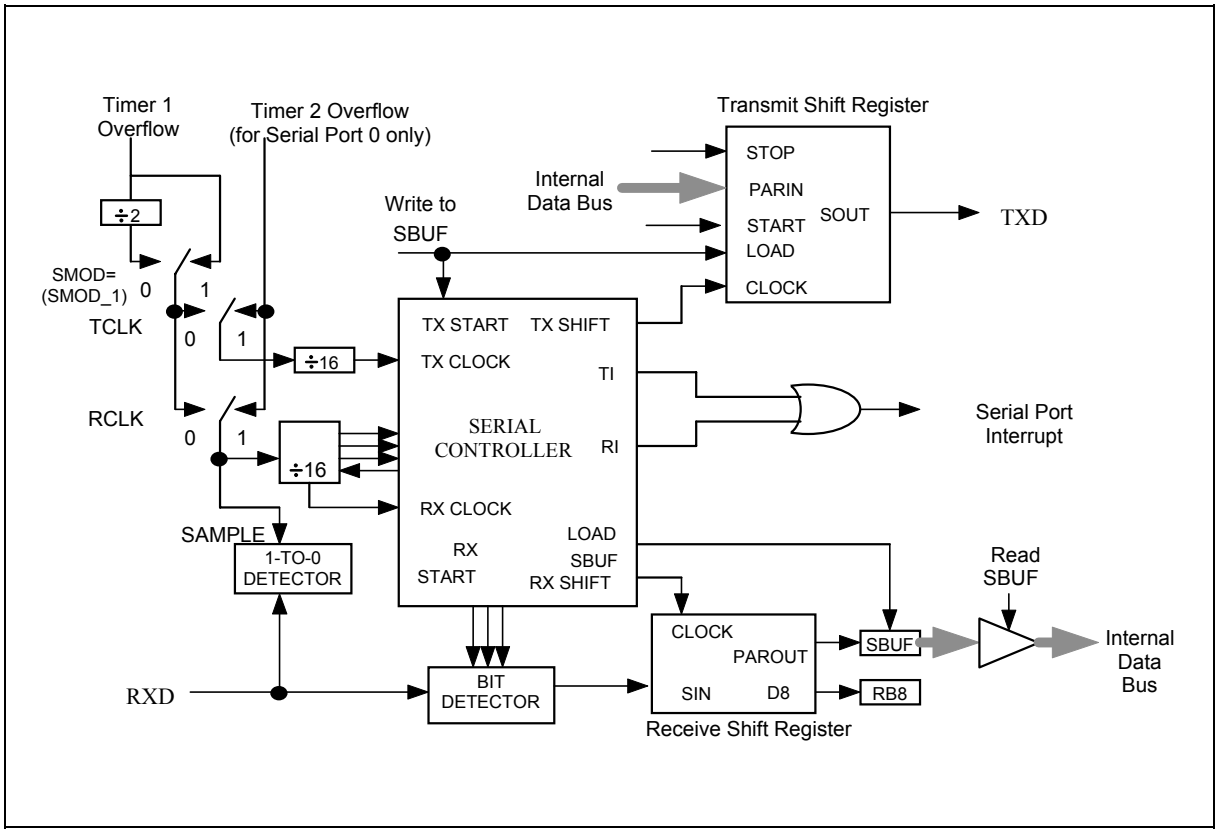



Figure 21. Serial Port Mode 1

### Mode 2

This mode uses a total of 11 bits in asynchronous full-duplex communication. The functional description is shown in the figure below. The frame consists of one start bit (0), 8 data bits (LSB first), a programmable 9th bit (TB8) and a stop bit (0). The 9th bit received is put into RB8. The baud rate is programmable to 1/32 or 1/64 of the oscillator frequency, which is determined by the SMOD bit in PCON SFR. Transmission begins with a write to SBUF. The serial data is brought out on to TxD pin at C1 following the first roll-over of the divide by 16 counter. The next bit is placed on TxD pin at C1 following the next rollover of the divide by 16 counter. Thus the transmission is synchronized to the divide by 16 counter, and not directly to the write to SBUF signal. After all 9 bits of data are transmitted, the stop bit is transmitted. The TI flag is set in the C1 state after the stop bit has been put out on TxD pin. This will be at the 11th rollover of the divide by 16 counter after a write to SBUF. Reception is enabled only if REN is high. The serial port actually starts the receiving of serial data, with the detection of a falling edge on the RxD pin. The 1-to-0 detector continuously monitors the RxD line, sampling it at the rate of 16 times the selected baud rate. When a falling edge is detected, the divide by 16 counter is immediately reset. This helps to align the bit boundaries with the rollovers of the divide by 16 counter. The 16 states of the counter effectively divide the bit time into 16 slices. The bit detection is done on a best of three basis. The bit detector samples the RxD pin, at the 8th, 9th and 10th counter states. By using a majority 2 of 3 voting system, the bit value is selected. This is done to improve the noise rejection feature of the serial port.

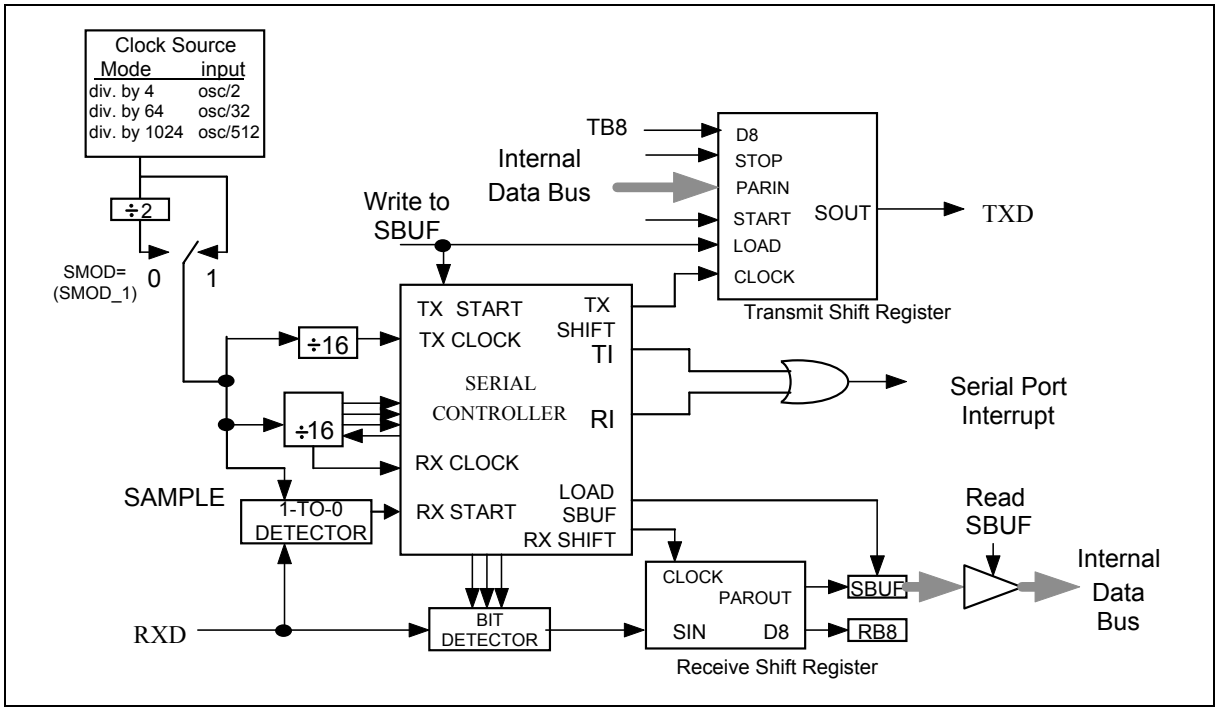



Figure 22. Serial Port Mode 2

If the first bit detected after the falling edge of RxD pin, is not 0, then this indicates an invalid start bit, and the reception is immediately aborted. The serial port again looks for a falling edge in the RxD line. If a valid start bit is detected, then the rest of the bits are also detected and shifted into the SBUF. After shifting in 9 data bits, there is one more shift to do, after which the SBUF and RB8 are loaded and RI is set. However certain conditions must be met before the loading and setting of RI can be done.

1. RI must be 0 and
2. Either SM2 = 0, or the received stop bit = 1.

If these conditions are met, then the stop bit goes to RB8, the 8 data bits go into SBUF and RI is set. Otherwise the received frame may be lost. After the middle of the stop bit, the receiver goes back to looking for a 1-to-0 transition on the RxD pin.

**Mode 3**

This mode is similar to Mode 2 in all respects, except that the baud rate is programmable. The user must first initialize the Serial related SFR SCON before any communication can take place. This involves selection of the Mode and baud rate. The Timer 1 should also be initialized if modes 1 and 3 are used. In all four modes, transmission is started by any instruction that uses SBUF as a destination register. Reception is initiated in Mode 0 by the condition RI = 0 and REN = 1. This will generate a clock on the TxD pin and shift in 8 bits on the RxD pin. Reception is initiated in the other modes by the incoming start bit if REN = 1. The external device will start the communication by transmitting the start bit.

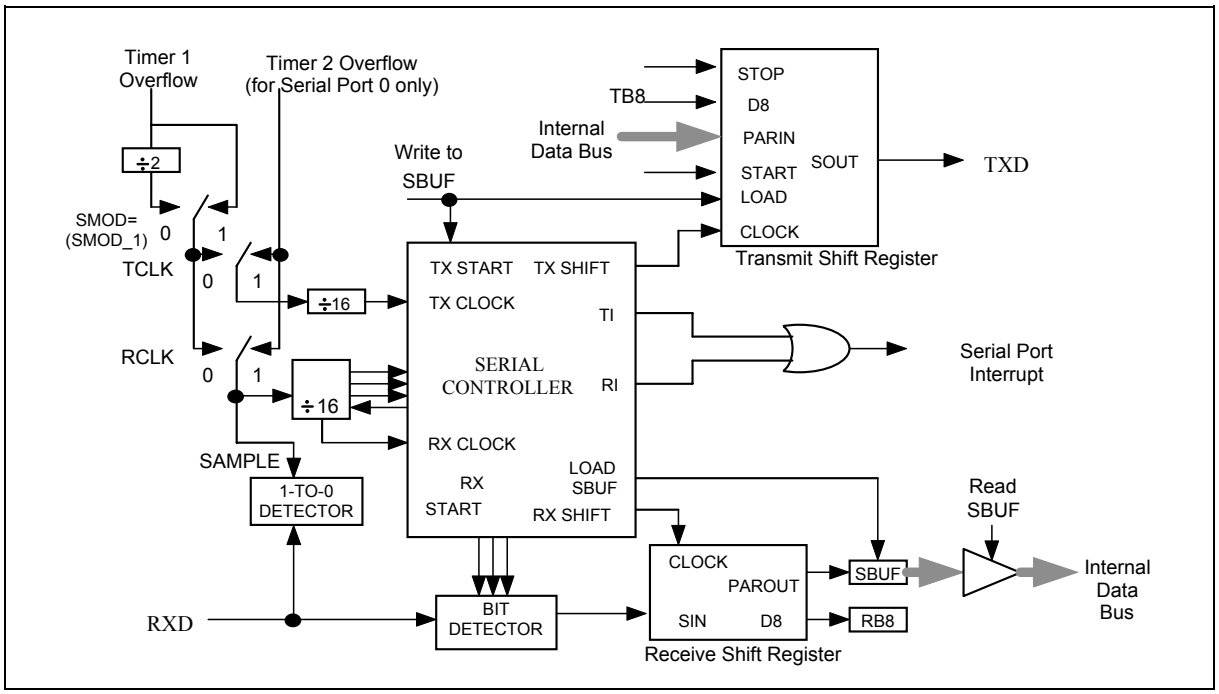



Figure 23. Serial Port Mode 3

**Table 10. Serial Ports Modes**

| SM1 | SM0 | MODE | TYPE | BAUD CLOCK | FRAME SIZE | START BIT | STOP BIT | 9TH BIT FUNCTION |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | Synch. | 4 or 12 TCLKS | 8 bits | No | No | None |
| 0 | 1 | 1 | Asynch. | Timer 1 or 2 | 10 bits | 1 | 1 | None |
| 1 | 0 | 2 | Asynch. | 32 or 64 TCLKS | 11 bits | 1 | 1 | 0, 1 |
| 1 | 1 | 3 | Asynch. | Timer 1 or 2 | 11 bits | 1 | 1 | 0, 1 |

### Framing Error Detection

A Frame Error occurs when a valid stop bit is not detected. This could indicate incorrect serial data communication. Typically the frame error is due to noise and contention on the serial communication line. The W77E532A has the facility to detect such framing errors and set a flag which can be checked by software.

The Frame Error FE(FE_1) bit is located in SCON.7(SCON1.7). This bit is normally used as SM0 in the standard 8051 family. However, in the W77E532A it serves a dual function and is called SM0/FE (SM0_1/FE_1). There are actually two separate flags, one for SM0 and the other for FE. The flag that is actually accessed as SCON.7(SCON1.7) is determined by SMOD0 (PCON.6) bit. When SMOD0 is set to 1, then the FE flag is indicated in SM0/FE. When SMOD0 is set to 0, then the SM0 flag is indicated in SM0/FE.

The FE bit is set to 1 by hardware but must be cleared by software. Note that SMOD0 must be 1 while reading or writing to FE or FE_1. If FE is set, then any following frames received without any error will not clear the FE flag. The clearing has to be done by software.

### Multiprocessor Communications

Multiprocessor communications makes use of the 9$^{th}$ data bit in modes 2 and 3. In the W77E532A, the RI flag is set only if the received byte corresponds to the Given or Broadcast address. This hardware feature eliminates the software overhead required in checking every received address, and greatly simplifies the software programmer task.

In the multiprocessor communication mode, the address bytes are distinguished from the data bytes by transmitting the address with the 9$^{th}$ bit set high. When the master processor wants to transmit a block of data to one of the slaves, it first sends out the address of the targeted slave (or slaves). All the slave processors should have their SM2 bit set high when waiting for an address byte. This ensures that they will be interrupted only by the reception of a address byte. The Automatic address recognition feature ensures that only the addressed slave will be interrupted. The address comparison is done in hardware not software.

The addressed slave clears the SM2 bit, thereby clearing the way to receive data bytes. With SM2 = 0, the slave will be interrupted on the reception of every single complete frame of data. The unaddressed slaves will be unaffected, as they will be still waiting for their address. In Mode 1, the 9$^{th}$ bit is the stop bit, which is 1 in case of a valid frame. If SM2 is 1, then RI is set only if a valid frame is received and the received byte matches the Given or Broadcast address.

The Master processor can selectively communicate with groups of slaves by using the Given Address. All the slaves can be addressed together using the Broadcast Address. The addresses for each slave are defined by the SADDR and SADEN SFRs. The slave address is an 8-bit value specified in the SADDR SFR. The SADEN SFR is actually a mask for the byte value in SADDR. If a bit position in SADEN is 0, then the corresponding bit position in SADDR is don't care. Only those bit positions in SADDR whose corresponding bits in SADEN are 1 are used to obtain the Given Address. This gives the user flexibility to address multiple slaves without changing the slave address in SADDR.

The following example shows how the user can define the Given Address to address different slaves.

Slave 1:

SADDR 1010 0100
SADEN 1111 1010
Given 1010 0x0x

Slave 2:

SADDR 1010 0111
SADEN 1111 1001
Given 1010 0xx1

The Given address for slave 1 and 2 differ in the LSB. For slave 1, it is a don't care, while for slave 2 it is 1. Thus to communicate only with slave 1, the master must send an address with LSB = 0 (1010 0000). Similarly the bit 1 position is 0 for slave 1 and don't care for slave 2. Hence to communicate only with slave 2 the master has to transmit an address with bit 1 = 1 (1010 0011). If the master wishes to communicate with both slaves simultaneously, then the address must have bit 0 = 1 and bit 1 = 0. The bit 3 position is don't care for both the slaves. This allows two different addresses to select both slaves (1010 0001 and 1010 0101).

The master can communicate with all the slaves simultaneously with the Broadcast Address. This address is formed from the logical Oring of the SADDR and SADEN SFRs. The zeros in the result are defined as don't cares In most cases the Broadcast Address is FFh. In the previous case, the Broadcast Address is (1111111X) for slave 1 and (11111111) for slave 2.

The SADDR and SADEN SFRs are located at address A9h and B9h respectively. On reset, these two SFRs are initialized to 00h. This results in Given Address and Broadcast Address being set as XXXX XXXX(i.e. all bits don't care). This effectively removes the multiprocessor communications feature, since any selectivity is disabled.

## 13. TIMED ACCESS PROTECTION

The W77E532A has several new features, like the Watchdog timer, on-chip ROM size adjustment, wait state control signal and Power on/fail reset flag, which are crucial to proper operation of the system. If left unprotected, errant code may write to the Watchdog control bits resulting in incorrect operation and loss of control. In order to prevent this, the W77E532A has a protection scheme which controls the write access to critical bits. This protection scheme is done using a timed access.

In this method, the bits which are to be protected have a timed write enable window. A write is successful only if this window is active, otherwise the write will be discarded. This write enable window is open for 3 machine cycles if certain conditions are met. After 3 machine cycles, this window automatically closes. The window is opened by writing Aah and immediately 55h to the Timed Access(TA) SFR. This SFR is located at address C7h. The suggested code for opening the timed access window is

```
TA      REG   0C7h          ;define new register TA, located at 0C7h
MOV     TA, #0Aah
MOV     TA, #055h
```

When the software writes Aah to the TA SFR, a counter is started. This counter waits for 3 machine cycles looking for a write of 55h to TA. If the second write (55h) occurs within 3 machine cycles of the first write (Aah), then the timed access window is opened. It remains open for 3 machine cycles, during which the user may write to the protected bits. Once the window closes the procedure must be repeated to access the other protected bits.

**Examples of Timed Assessing are shown below.**

Example 1: Valid access

```
MOV  TA, #0Aah    3 M/C          Note: M/C = Machine Cycles
MOV  TA, #055h    3 M/C
MOV  WDCON, #00h      3 M/C
```

Example 2: Valid access

```
MOV  TA, #0Aah    3 M/C
MOV  TA, #055h    3 M/C
NOP               1 M/C
SETB EWT          2 M/C
```

Example 3: Valid access

```
MOV TA, #0Aah     3 M/C
MOV TA, #055h     3 M/C
ORL  WDCON, #00000010B     3M/C
```

Example 4: Invalid access

```
MOV  TA, #0Aah    3 M/C
MOV  TA, #055h    3 M/C
NOP               1 M/C
NOP               1 M/C
CLR  POR          2 M/C
```

Example 5: Invalid Access

```
MOV  TA, #0Aah    3 M/C
NOP               1 M/C
MOV  TA, #055h    3 M/C
SETB EWT          2 M/C
```

In the first two examples, the writing to the protected bits is done before the 3 machine cycle window closes. In Example 3, however, the writing to the protected bit occurs after the window has closed, and so there is effectively no change in the status of the protected bit. In Example 4, the second write to TA occurs 4 machine cycles after the first write, therefore the timed access window in not opened at all, and the write to the protected bit fails.

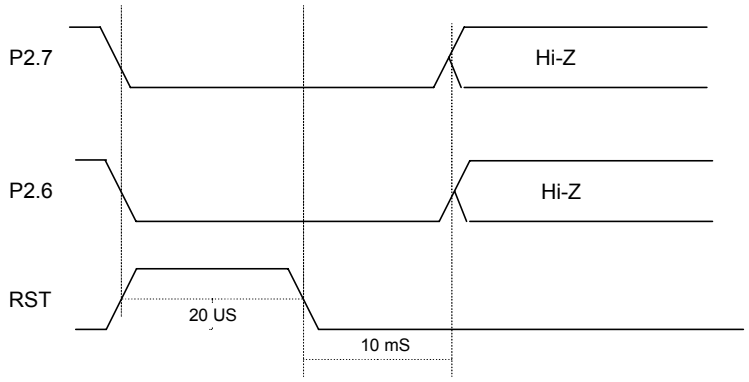
## 14. H/W REBOOT MODE (BOOT FROM 4K BYTES OF LDFLASH )

The W77E532A boots from APFLASH program (64K bytes) by default at the external reset. On some occasions, user can force W77E532A to boot from the LDFLASH program (4K bytes) at the external reset. The settings for this special mode is as follow. It is necessary to add 10K resistor on these P2.6, P2.7 and P4.3 pins.

**Reboot Mode**

| OPTION BITS | RST | P4.3 | P2.7 | P2.6 | MODE |
|---|---|---|---|---|---|
| Bit4 L | H | X | L | L | REBOOT |
| Bit5 L | H | L | X | X | REBOOT |

**The Reset Timing For Entering REBOOT Mode**




**Notes:**

1: The possible situation that you need to enter REBOOT mode is when the APFLASH program can not run normally and W77E532A can not jump to LDFLASH to execute on chip programming function. Then you can use this REBOOT mode to force the CPU jump to LDFLASH and run on chip programming procedure. When you design your system, you can connect the pins P26, P27 to switches or jumpers. For example in a CD ROM system, you can connect the P26 and P27 to PLAY and EJECT buttons on the panel. When the APFLASH program is fail to execute the normal application program. User can press both two buttons at the same time and then switch on the power of the personal computer to force the W77E532A to enter the REBOOT mode. After power on of personal computer, you can release both PLAY and EJECT button. And re-run the on chip programming procedure to let the APFLASH have the normal program code. Then you can back to normal condition of CD ROM.

2: In application system design, user must take care the P4.3, P2, P3, ALE, /EA and /PSEN pin value at reset to avoid W77E532A entering the programming mode or REBOOT mode in normal operation.

## 15. IN-SYSTEM PROGRAMMING

### 15.1 The Loader Program Locates at LDFLASH Memory

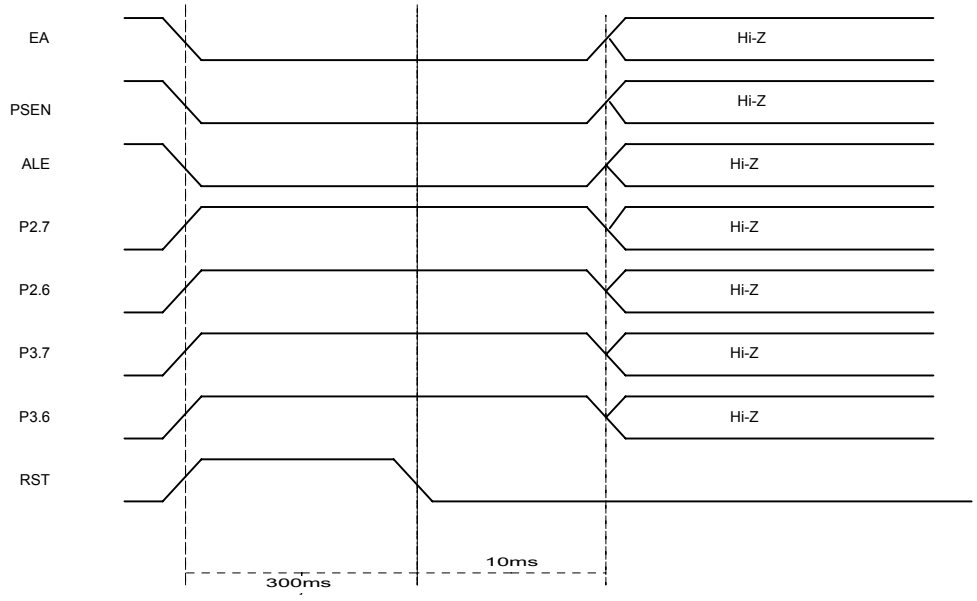
CPU is Free Run at APFLASH memory. CHPCON register had been set #03H value before CPU has entered idle state. CPU will switch to LDFLASH memory and execute a reset action. H/W reboot mode will switch to LDFLASH memory, too. Set SFRCN register where it locates at user's loader program to update APFLASH bank 0 or bank 1 memory. Set a SWRESET ( CHPCON=#83H) to switch back APFLASH after CPU has updated APFLASH program. CPU will restart to run program from reset state.

### 15.2 The Loader Program Locates at APFLASH Memory

CPU is Free Run at APFLASH memory. CHPCON register had been set #01H value before CPU has entered idle state. Set SFRCN register to update LDFLASH or another bank of APFLASH program. CPU will continue to run user's APFLASH program after CPU has updated program. Please refer demonstrative code to understand other detail description.

## 16. H/W WRITER MODE

**The Timing For Entering Flash EPROM Mode on the Programmer**



## 17. SECURITY BITS

Using device programmer, the Flash EPROM can be programmed and verified repeatedly. Until the code inside the Flash EPROM is confirmed OK, the code can be protected. The protection of Flash EPROM and those operations on it are described below. The W77E532A has Special Setting Register which can be accessed by device programmer. The register can only be accessed from the Flash EPROM operation mode. Those bits of the Security Registers can not be changed once they have been programmed from high to low. They can only be reset through erase-all operation.

| B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | **Security Bits** |
|---|---|---|---|---|---|---|---|---|

B5 : 0 -> Eable H/W reboot with P4.3
B4 : 0 -> Enable H/W reboot with P2.6, P2.7
B1 : 0 -> MOVC Inhibited
B0 : 0-> Data out lock
Default 1 for each bit.

**Special Setting Registers**

**B0: Lock bit**

This bit is used to protect the customer's program code in the W77E532A. It may be set after the programmer finishes the programming and verifies sequence. Once this bit is set to logic 0, both the Flash EPROM data and Special Setting Registers can not be accessed again.

**B1: MOVC Inhibit**

This bit is used to restrict the accessible region of the MOVC instruction. It can prevent the MOVC instruction in external program memory from reading the internal program code. When this bit is set to logic 0, a MOVC instruction in external program memory space will be able to access code only in the external memory, not in the internal memory. A MOVC instruction in internal program memory space will always be able to access the ROM data in both internal and external memory. If this bit is logic 1, there are no restrictions on the MOVC instruction.

**B4: H/W Reboot with P2.6 and P2.7**

If this bit is set to logic 0, enable to reboot 4k LDFLASH mode while RST =H, P2.6 = L and P2.7 = L state. CPU will start from LDFLASH to update the user's program.

**B5: H/W Reboot with P4.3**

If this bit is set to logic 0, enable to reboot 4k LDFLASH mode while RST =H and P4.3 = L state. CPU will start from LDFLASH to update the user's program.

# 18. ELECTRICAL CHARACTERISTICS

## 18.1 Absolute Maximum Ratings

| SYMBOL | PARAMETER | CONDITION | RATING | UNIT |
|---|---|---|---|---|
| DC Power Supply | $V_{DD} - V_{SS}$ | -0.3 | +7.0 | V |
| Input Voltage | $V_{IN}$ | $V_{SS}$ -0.3 | $V_{DD}$ +0.3 | V |
| Operating Temperature | $T_A$ | 0 | +70 | °C |
| Storage Temperatute | Tst | -55 | +150 | °C |

**Note:** Exposure to conditions beyond those listed under Absolute Maximum Ratings may adversely affect the life and reliability of the device.

## 18.2 DC Characteristics

($V_{DD} - V_{SS}$ = 5V ±10%, $T_A$ = 25°C, Fosc = 20 MHz, unless otherwise specified.)

| PARAMETER | SYMBOL | SPECIFICATION | | | TEST CONDITIONS |
|---|---|---|---|---|---|
| | | MIN. | MAX. | UNIT | |
| Operating Voltage | $V_{DD}$ | 4.5 | 5.5 | V | |
| Operating Current | $I_{DD}$ | - | 30 | mA | No load<br>$V_{DD}$ = RST = 5.5V |
| Idle Current | $I_{IDLE}$ | - | 24 | mA | Idle mode<br>$V_{DD}$ = 5.5V |
| Power Down Current | $I_{PWDN}$ | - | 10 | µA | Power-down mode<br>$V_{DD}$ = 5.5V |
| Input Current P1, P2, P3 | $I_{IN1}$ | -50 | +10 | µA | $V_{DD}$ = 5.5V<br>$V_{IN}$ = 0V or $V_{DD}$ |
| Input Current RST[*1] | $I_{IN2}$ | -10 | +120 | µA | $V_{DD}$ = 5.5V<br>0<$V_{IN}$<$V_{DD}$ |
| Input Leakage Current P0, $\overline{EA}$ | $I_{LK}$ | -10 | +10 | µA | $V_{DD}$ = 5.5V<br>0V<$V_{IN}$<$V_{DD}$ |
| Logic 1 to 0 Transition Current P1, P2, P3 | $I_{TL}$[*4] | -500 | -200 | µA | $V_{DD}$ = 5.5V<br>$V_{IN}$ = 2.0V |
| Input Low Voltage P0, P1, P2, P3, $\overline{EA}$ | $V_{IL1}$ | 0 | 0.8 | V | $V_{DD}$ = 4.5V |
| Input Low Voltage RST[*1] | $V_{IL2}$ | 0 | 0.8 | V | $V_{DD}$ = 4.5V |
| Input Low Voltage XTAL1[*3] | $V_{IL3}$ | 0 | 0.8 | V | $V_{DD}$ = 4.5V |
| Input High Voltage P0, P1, P2, P3, $\overline{EA}$ | $V_{IH1}$ | 2.4 | $V_{DD}$ +0.2 | V | $V_{DD}$ = 5.5V |
| Input High Voltage RST | $V_{IH2}$ | 3.5 | $V_{DD}$ +0.2 | V | $V_{DD}$ = 5.5V |
| Input High Voltage XTAL1[*3] | $V_{IH3}$ | 3.5 | $V_{DD}$ +0.2 | V | $V_{DD}$ = 5.5V |

DC Characteristics, continued

| PARAMETER | SYMBOL | SPECIFICATION | | | TEST CONDITIONS |
|---|---|---|---|---|---|
| | | MIN. | MAX. | UNIT | |
| Sink current<br>P1, P3 | Isk1 | 6 | 9 | mA | $V_{DD}$ = 4.5V<br>Vs = 0.45V |
| Sink current<br>P0, P2, ALE, $\overline{PSEN}$ | Isk2 | 10 | 14 | mA | $V_{DD}$ = 4.5V<br>$V_{OL}$ = 0.45V |
| Source current<br>P1, P3 | Isr1 | -180 | -360 | uA | $V_{DD}$ = 4.5V<br>$V_{OL}$ = 2.4V |
| Source current<br>P0, P2, ALE, $\overline{PSEN}$ | Isr2 | -10 | -18 | mA | $V_{DD}$ = 4.5V<br>$V_{OL}$ = 2.4V |
| Output Low Voltage<br>P1, P3 | $V_{OL1}$ | - | 0.45 | V | $V_{DD}$ = 4.5V<br>$I_{OL}$ = +6 mA |
| Output Low Voltage<br>P0, P2, ALE, $\overline{PSEN}$ [*2] | $V_{OL2}$ | - | 0.45 | V | $V_{DD}$ = 4.5V<br>$I_{OL}$ = +10 mA |
| Output High Voltage<br>P1, P3 | $V_{OH1}$ | 2.4 | - | V | $V_{DD}$ = 4.5V<br>$I_{OH}$ = -180 μA |
| Output High Voltage<br>P0, P2, ALE, $\overline{PSEN}$ [*2] | $V_{OH2}$ | 2.4 | - | V | $V_{DD}$ = 4.5V<br>$I_{OH}$ = -10mA |

**Notes:**

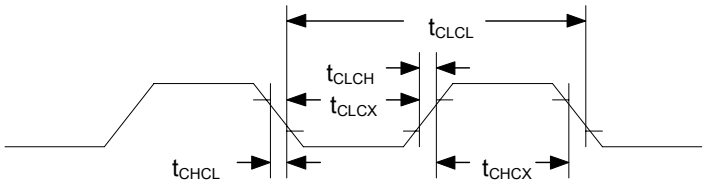
*1. RST pin is a Schmitt trigger input.

*2. P0, ALE and $\overline{PSEN}$ are tested in the external access mode.

*3. XTAL1 is a CMOS input.

*4. Pins of P1, P2, P3 can source a transition current when they are being externally driven from 1 to 0. The transition current reaches its maximum value when VIN approximates to 2V.

## 18.3 AC Characteristics




**Note:** Duty cycle is 50%.

**External Clock Characteristics**

| PARAMETER | SYMBOL | MIN. | TYP. | MAX. | UNITS | NOTES |
|---|---|---|---|---|---|---|
| Clock High Time | $t_{CHCX}$ | 12 | - | - | nS | |
| Clock Low Time | $t_{CLCX}$ | 12 | - | - | nS | |
| Clock Rise Time | $t_{CLCH}$ | - | - | 10 | nS | |
| Clock Fall Time | $t_{CHCL}$ | - | - | 10 | nS | |

### 18.3.1 AC Specification

| PARAMETER | SYMBOL | VARIABLE CLOCK MIN. | VARIABLE CLOCK MAX. | UNITS |
|---|---|---|---|---|
| Oscillator Frequency | $1/t_{CLCL}$ | 0 | 40 | MHz |
| ALE Pulse Width | $t_{LHLL}$ | $1.5t_{CLCL} - 5$ | | nS |
| Address Valid to ALE Low | $t_{AVLL}$ | $0.5t_{CLCL} - 5$ | | nS |
| Address Hold After ALE Low | $t_{LLAX1}$ | $0.5t_{CLCL} - 5$ | | nS |
| Address Hold After ALE Low for MOVX Write | $t_{LLAX2}$ | $0.5t_{CLCL} - 5$ | | nS |
| ALE Low to Valid Instruction In | $t_{LLIV}$ | | $2.5t_{CLCL} - 20$ | nS |
| ALE Low to PSEN Low | $t_{LLPL}$ | $0.5t_{CLCL} - 5$ | | nS |
| $\overline{\text{PSEN}}$ Pulse Width | $t_{PLPH}$ | $2.0t_{CLCL} - 5$ | | nS |
| $\overline{\text{PSEN}}$ Low to Valid Instruction In | $t_{PLIV}$ | | $2.0t_{CLCL} - 20$ | nS |
| Input Instruction Hold After PSEN | $t_{PXIX}$ | 0 | | nS |
| Input Instruction Float After PSEN | $t_{PXIZ}$ | | $t_{CLCL} - 5$ | nS |
| Port 0 Address to Valid Instr. In | $t_{AVIV1}$ | | $3.0t_{CLCL} - 20$ | nS |
| Port 2 Address to Valid Instr. In | $t_{AVIV2}$ | | $3.5t_{CLCL} - 20$ | nS |
| $\overline{\text{PSEN}}$ Low to Address Float | $t_{PLAZ}$ | 0 | | nS |
| Data Hold After Read | $t_{RHDX}$ | 0 | | nS |
| Data Float After Read | $t_{RHDZ}$ | | $t_{CLCL} - 5$ | nS |
| $\overline{\text{RD}}$ Low to Address Float | $t_{RLAZ}$ | | $0.5t_{CLCL} - 5$ | nS |

### 18.3.2 MOVX Characteristics Using Strech Memory Cycle

| PARAMETER | SYMBOL | VARIABLE CLOCK MIN. | VARIABLE CLOCK MAX. | UNITS | STRECH |
|---|---|---|---|---|---|
| Data Access ALE Pulse Width | $t_{LLHL2}$ | $1.5t_{CLCL} - 5$<br>$2.0t_{CLCL} - 5$ | | nS | $t_{MCS} = 0$<br>$t_{MCS}>0$ |
| Address Hold After ALE Low for MOVX write | $t_{LLAX2}$ | $0.5t_{CLCL} - 5$ | | nS | |
| $\overline{RD}$ Pulse Width | $t_{RLRH}$ | $2.0t_{CLCL} - 5$<br>$t_{MCS} - 10$ | | nS | $t_{MCS} = 0$<br>$t_{MCS}>0$ |
| $\overline{WR}$ Pulse Width | $t_{WLWH}$ | $2.0t_{CLCL} - 5$<br>$t_{MCS} - 10$ | | nS | $t_{MCS} = 0$<br>$t_{MCS}>0$ |
| $\overline{RD}$ Low to Valid Data In | $t_{RLDV}$ | | $2.0t_{CLCL} - 20$<br>$t_{MCS} - 20$ | nS | $t_{MCS} = 0$<br>$t_{MCS}>0$ |
| Data Hold after Read | $t_{RHDX}$ | 0 | | nS | |
| Data Float after Read | $t_{RHDZ}$ | | $t_{CLCL} - 5$<br>$2.0t_{CLCL} - 5$ | nS | $t_{MCS} = 0$<br>$t_{MCS}>0$ |
| ALE Low to Valid Data In | $t_{LLDV}$ | | $2.5t_{CLCL} - 5$<br>$t_{MCS} + 2t_{CLCL} - 40$ | nS | $t_{MCS} = 0$<br>$t_{MCS}>0$ |
| Port 0 Address to Valid Data In | $t_{AVDV1}$ | | $3.0t_{CLCL} - 20$<br>$2.0t_{CLCL} - 5$ | nS | $t_{MCS} = 0$<br>$t_{MCS}>0$ |
| ALE Low to $\overline{RD}$ or $\overline{WR}$ Low | $t_{LLWL}$ | $0.5t_{CLCL} - 5$<br>$1.5t_{CLCL} - 5$ | $0.5t_{CLCL} + 5$<br>$1.5t_{CLCL} + 5$ | nS | $t_{MCS} = 0$<br>$t_{MCS}>0$ |
| Port 0 Address to $\overline{RD}$ or $\overline{WR}$ Low | $t_{AVWL}$ | $t_{CLCL} - 5$<br>$2.0t_{CLCL} - 5$ | | nS | $t_{MCS} = 0$<br>$t_{MCS}>0$ |
| Port 2 Address to $\overline{RD}$ or $\overline{WR}$ Low | $t_{AVWL2}$ | $1.5t_{CLCL} - 5$<br>$2.5t_{CLCL} - 5$ | | nS | $t_{MCS} = 0$<br>$t_{MCS}>0$ |
| Data Valid to WR Transition | $t_{QVWX}$ | -5<br>$1.0t_{CLCL} - 5$ | | nS | $t_{MCS} = 0$<br>$t_{MCS}>0$ |
| Data Hold after Write | $t_{WHQX}$ | $t_{CLCL} - 5$<br>$2.0t_{CLCL} - 5$ | | nS | $t_{MCS} = 0$<br>$t_{MCS}>0$ |
| $\overline{RD}$ Low to Address Float | $t_{RLAZ}$ | | $0.5t_{CLCL} - 5$ | nS | |
| $\overline{RD}$ or WR high to ALE high | $t_{WHLH}$ | 0<br>$1.0t_{CLCL} - 5$ | 10<br>$1.0t_{CLCL} + 5$ | nS | $t_{MCS} = 0$<br>$t_{MCS}>0$ |

**Note:** $t_{MCS}$ is a time period related to the Stretch memory cycle selection. The following table shows the time period of $t_{MCS}$ for each selection of the Stretch value.

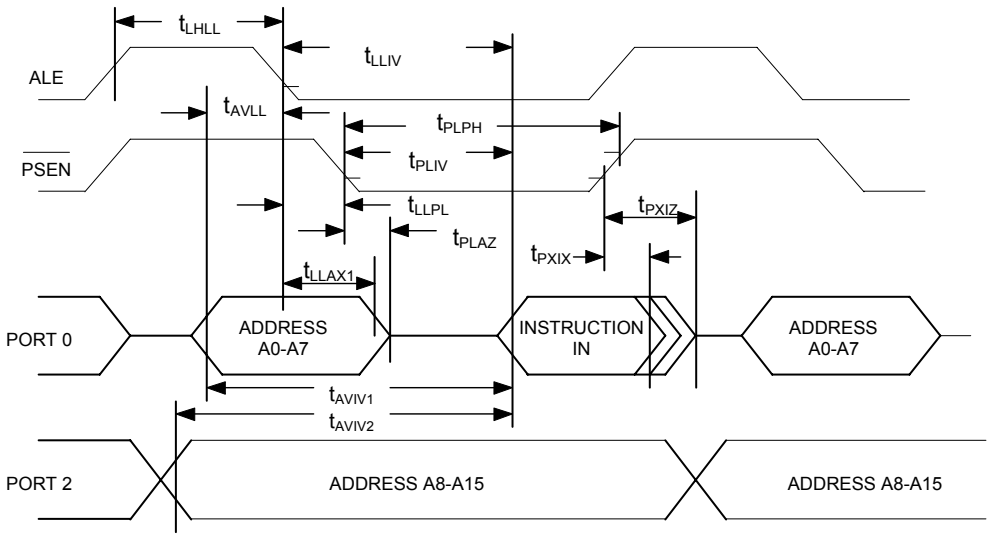
| M2 | M1 | M0 | MOVX CYCLES | $T_{MCS}$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 2 machine cycles | 0 |
| 0 | 0 | 1 | 3 machine cycles | 4 $t_{CLCL}$ |
| 0 | 1 | 0 | 4 machine cycles | 8 $t_{CLCL}$ |
| 0 | 1 | 1 | 5 machine cycles | 12 $t_{CLCL}$ |
| 1 | 0 | 0 | 6 machine cycles | 16 $t_{CLCL}$ |
| 1 | 0 | 1 | 7 machine cycles | 20 $t_{CLCL}$ |
| 1 | 1 | 0 | 8 machine cycles | 24 $t_{CLCL}$ |
| 1 | 1 | 1 | 9 machine cycles | 28 $t_{CLCL}$ |

**Explanation of Logics Symbols**

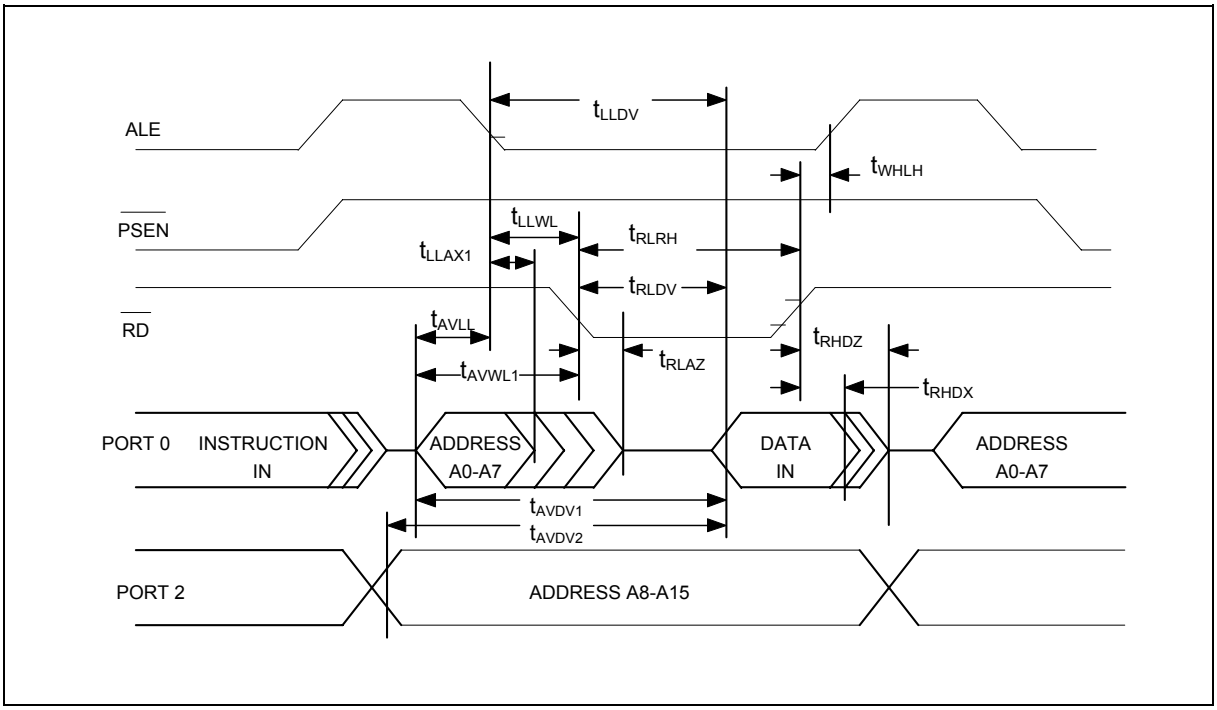
In order to maintain compatibility with the original 8051 family, this device specifies the same parameter for each device, using the same symbols. The explanation of the symbols is as follows.

| | | | |
|---|---|---|---|
| T | Time | A | Address |
| C | Clock | D | Input Data |
| H | Logic level high | L | Logic level low |
| I | Instruction | P | $\overline{PSEN}$ |
| Q | Output Data | R | $\overline{RD}$ signal |
| V | Valid | W | $\overline{WR}$ signal |
| X | No longer a valid state | Z | Tri-state |

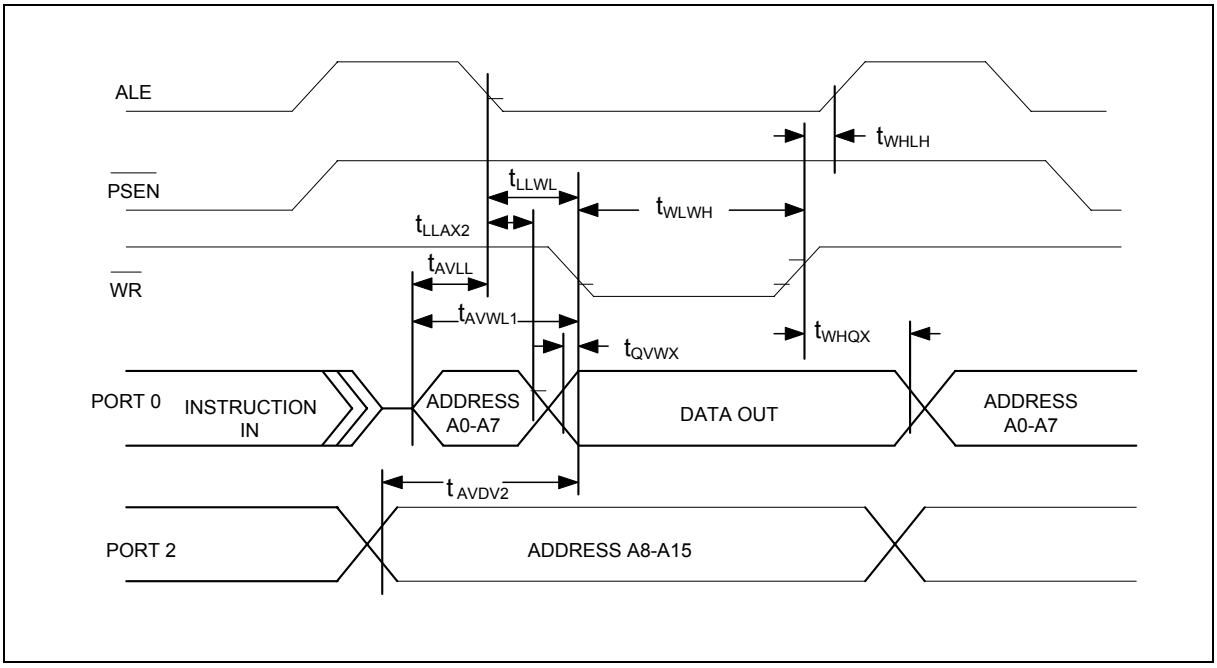
### 18.3.3 Program Memory Read Cycle



### 18.3.4 Data Memory Read Cycle




### 18.3.5 Data Memory Write Cycle



## 19. TYPICAL APPLICATION CIRCUITS

### 19.1 Crystal connections

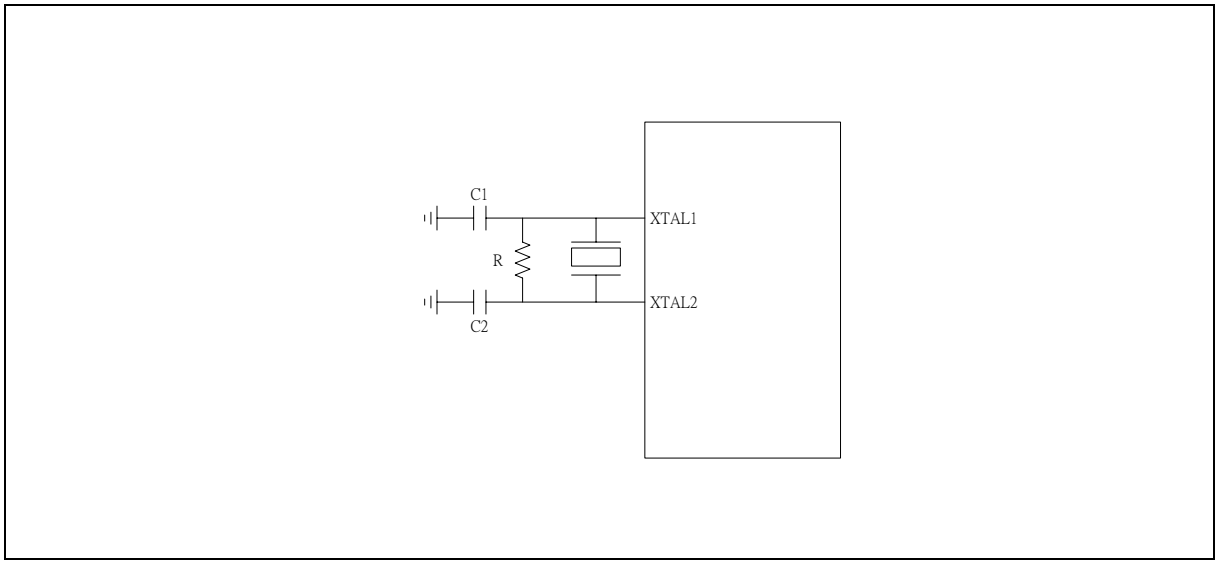



Figure A

| CRYSTAL | C1 | C2 | R |
|---|---|---|---|
| 16 MHz | 20P | 20P | - |
| 24 MHz | 12P | 12P | - |
| 33 MHz | 10P | 10P | 3K |
| 40 MHz | 1P | 1P | 3K |

The above table shows the reference values for crystal applications.

**Note:** C1, C2, R components refer to Figure A.

## 19.2 Expanded External Data Memory and Oscillator

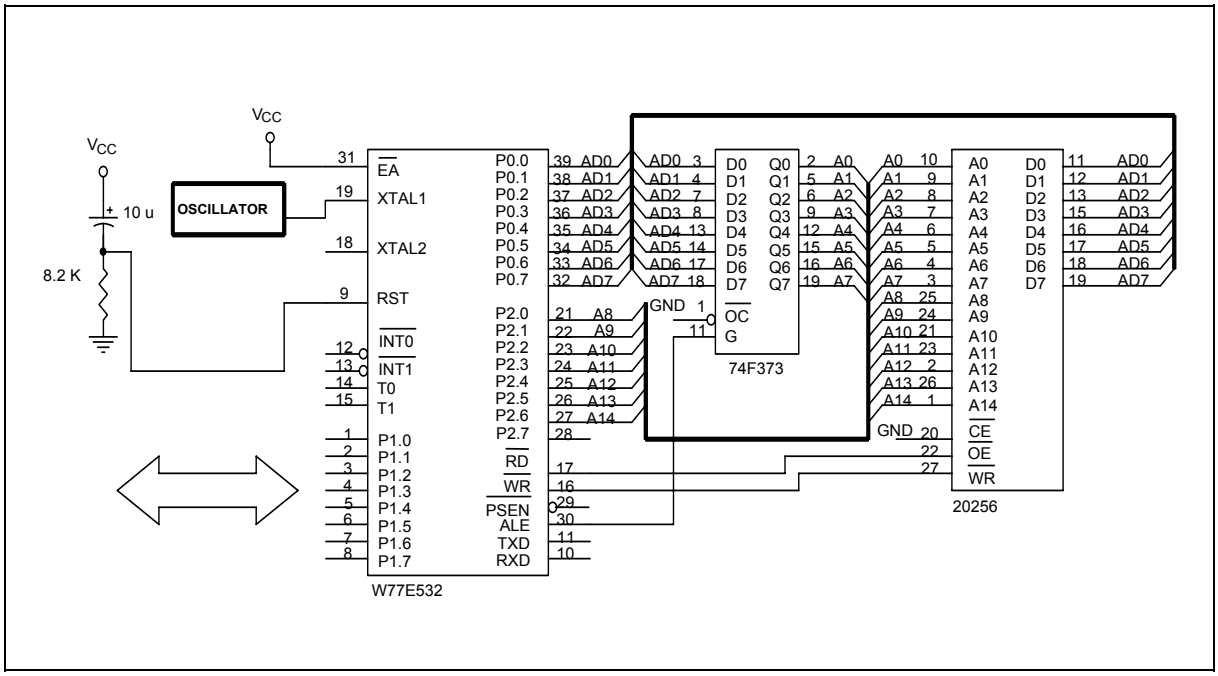
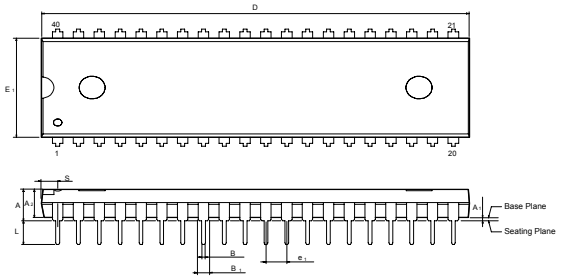



Figure B

# 20. PACKAGE DIMENSIONS

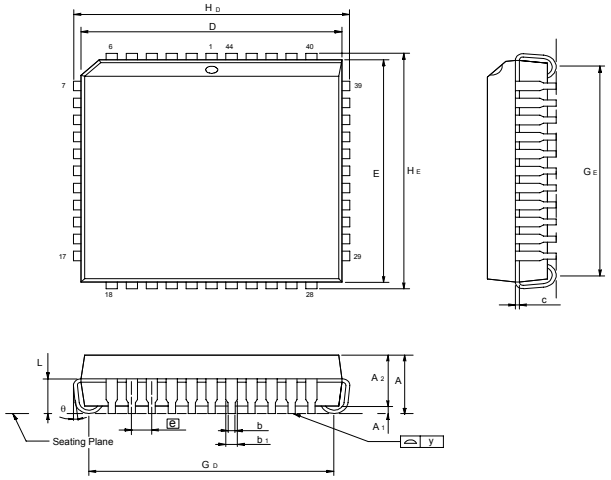
## 20.1 40-pin DIP




| Symbol | Dimension in inches | | | Dimension in mm | | |
|---|---|---|---|---|---|---|
| | Min. | Nom. | Max. | Min. | Nom. | Max. |
| A | — | — | 0.210 | — | — | 5.334 |
| $A_1$ | 0.010 | — | — | 0.254 | — | — |
| $A_2$ | 0.150 | 0.155 | 0.160 | 3.81 | 3.937 | 4.064 |
| B | 0.016 | 0.018 | 0.022 | 0.406 | 0.457 | 0.559 |
| $B_1$ | 0.048 | 0.050 | 0.054 | 1.219 | 1.27 | 1.372 |
| c | 0.008 | 0.010 | 0.014 | 0.203 | 0.254 | 0.356 |
| D | — | 2.055 | 2.070 | — | 52.20 | 52.58 |
| E | 0.590 | 0.600 | 0.610 | 14.986 | 15.24 | 15.494 |
| $E_1$ | 0.540 | 0.545 | 0.550 | 13.72 | 13.84 | 13.97 |
| $e_1$ | 0.090 | 0.100 | 0.110 | 2.286 | 2.54 | 2.794 |
| L | 0.120 | 0.130 | 0.140 | 3.048 | 3.302 | 3.556 |
| a | 0 | — | 15 | 0 | — | 15 |
| $e_A$ | 0.630 | 0.650 | 0.670 | 16.00 | 16.51 | 17.01 |
| S | — | — | 0.090 | — | — | 2.286 |

Notes:
1. Dimension D Max. & S include mold flash or tie bar burrs.
2. Dimension E1 does not include interlead flash.
3. Dimension D & E1 include mold mismatch and are determined at the mold parting line.
4. Dimension B1 does not include dambar protrusion/intrusion.
5. Controlling dimension: Inches.
6. General appearance spec. should be based on final visual inspection spec.

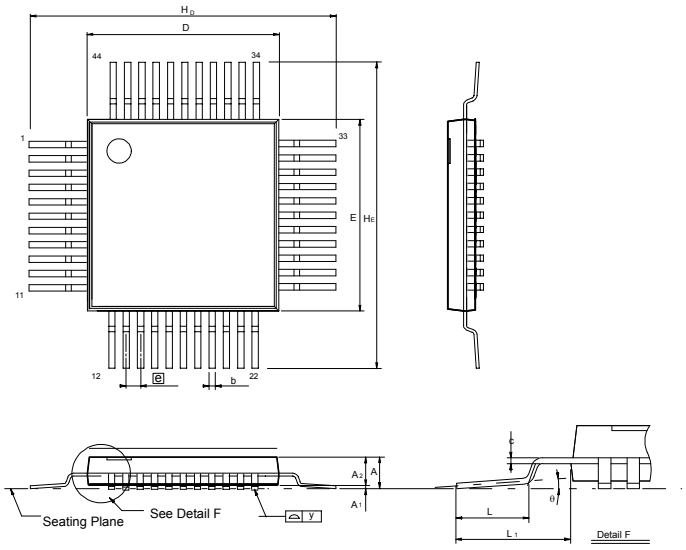
## 20.2 44-pin PLCC




| Symbol | Dimension in inches | | | Dimension in mm | | |
|---|---|---|---|---|---|---|
| | Min. | Nom. | Max. | Min. | Nom. | Max. |
| A | — | — | 0.185 | — | — | 4.699 |
| $A_1$ | 0.020 | — | — | 0.508 | — | — |
| $A_2$ | 0.145 | 0.150 | 0.155 | 3.683 | 3.81 | 3.937 |
| $b_1$ | 0.026 | 0.028 | 0.032 | 0.66 | 0.711 | 0.813 |
| b | 0.016 | 0.018 | 0.022 | 0.406 | 0.457 | 0.559 |
| c | 0.008 | 0.010 | 0.014 | 0.203 | 0.254 | 0.356 |
| D | 0.648 | 0.653 | 0.658 | 16.46 | 16.59 | 16.71 |
| E | 0.648 | 0.653 | 0.658 | 16.46 | 16.59 | 16.71 |
| e | 0.050 BSC | | | 1.27 BSC | | |
| $G_D$ | 0.590 | 0.610 | 0.630 | 14.99 | 15.49 | 16.00 |
| $G_E$ | 0.590 | 0.610 | 0.630 | 14.99 | 15.49 | 16.00 |
| $H_D$ | 0.680 | 0.690 | 0.700 | 17.27 | 17.53 | 17.78 |
| $H_E$ | 0.680 | 0.690 | 0.700 | 17.27 | 17.53 | 17.78 |
| L | 0.090 | 0.100 | 0.110 | 2.296 | 2.54 | 2.794 |
| y | — | — | 0.004 | — | — | 0.10 |

Notes:
1. Dimension D & E do not include interlead flash.
2. Dimension b1 does not include dambar protrusion/intrusion.
3. Controlling dimension: Inches
4. General appearance spec. should be based on final visual inspection spec.

## 20.3 44-pin QFP




| Symbol | Dimension in inch | | | Dimension in mm | | |
|---|---|---|---|---|---|---|
| | Min. | Nom. | Max. | Min. | Nom. | Max. |
| A | --- | --- | --- | --- | --- | --- |
| $A_1$ | 0.002 | 0.01 | 0.02 | 0.05 | 0.25 | 0.5 |
| $A_2$ | 0.075 | 0.081 | 0.087 | 1.90 | 2.05 | 2.20 |
| b | 0.01 | 0.014 | 0.018 | 0.25 | 0.35 | 0.45 |
| c | 0.004 | 0.006 | 0.010 | 0.101 | 0.152 | 0.254 |
| D | 0.390 | 0.394 | 0.398 | 9.9 | 10.00 | 10.1 |
| E | 0.390 | 0.394 | 0.398 | 9.9 | 10.00 | 10.1 |
| e | 0.025 | 0.031 | 0.036 | 0.635 | 0.80 | 0.952 |
| $H_D$ | 0.510 | 0.520 | 0.530 | 12.95 | 13.2 | 13.45 |
| $H_E$ | 0.510 | 0.520 | 0.530 | 12.95 | 13.2 | 13.45 |
| L | 0.025 | 0.031 | 0.037 | 0.65 | 0.8 | 0.95 |
| $L_1$ | 0.051 | 0.063 | 0.075 | 1.295 | 1.6 | 1.905 |
| y | — | — | 0.003 | — | — | 0.08 |
| θ | 0° | — | 7° | 0° | — | 7° |

Notes:

1. Dimension D & E do not include interlead flash.
2. Dimension b does not include dambar protrusion/intrusion.
3. Controlling dimension: Millimeter
4. General appearance spec. should be based on final visual inspection spec.

# 21. APPLICATION NOTE

## In-system Programming Software Examples

This application note illustrates the in-system programmability of the Winbond W77E532A Flash EPROM microcontroller. In this example, microcontroller will boot from 64 KB APFLASH bank and waiting for a key to enter in-system programming mode for re-programming the contents of 64 KB APFLASH. While entering in-system programming mode, microcontroller executes the loader program in 4KB LDFLASH bank. The loader program erases the 64 KB APFLASH then reads the new code data from external SRAM buffer (or through other interfaces) to update the 64KB APFLASH.

If the customer uses the reboot mode to update his program, please enable this b3 or b4 of security bits from the writer. Please refer security bits for detail descrption

**EXAMPLE 1:**

```
;*****************************************************************************************************************
;* Example of 64K APFLASH program: Program will scan the P1.0. if P1.0 = 0, enters in-system
;* programming mode for updating the content of APFLASH code else executes the current ROM
code.
;* XTAL = 24 MHz
;*****************************************************************************************************************
        .chip 8052
        .RAMCHK OFF
        .symbols

  CHPCON    EQU         9FH
  TA        EQU         C7H
  SFRAL     EQU         ACH
  SFRAH     EQU         ADH
  SFRFD     EQU         AEH
  SFRCN     EQU         AFH

        ORG   0H
          LJMP        100H                ; JUMP TO MAIN PROGRAM
    ;*********************************************************************
    ;*  TIMER0 SERVICE VECTOR ORG = 000BH
    ;*********************************************************************
        ORG   00BH
          CLR  TR0                        ; TR0 = 0, STOP TIMER0
          MOV  TL0,R6
          MOV  TH0,R7
          RETI
    ;*********************************************************************
    ;*  64K APFLASH MAIN PROGRAM
    ;*********************************************************************
        ORG 100H

  MAIN_64K:
        MOV  A,P1                         ; SCAN P1.0
        ANL  A,#01H
        CJNE  A,#01H,PROGRAM_64K          ; IF P1.0 = 0, ENTER IN-SYSTEM PROGRAMMING MODE
        JMP  NORMAL_MODE
```

```
PROGRAM_64:
      MOV  TA, #AAH              ; CHPCON register is written protect by TA register.
      MOV  TA, #55H
      MOV  CHPCON, #03H          ; CHPCON = 03H, ENTER IN-SYSTEM PROGRAMMING MODE
      MOV  SFRCN, #0H
        MOV  TCON, #00H          ; TR = 0 TIMER0 STOP
      MOV  IP, #00H              ; IP = 00H
      MOV  IE, #82H              ; TIMER0 INTERRUPT ENABLE FOR WAKE-UP FROM IDLE MODE
      MOV  R6, #F0H              ; TL0 = F0H
        MOV  R7, #FFH            ; TH0 = FFH
      MOV  TL0, R6
      MOV  TH0, R7
      MOV  TMOD, #01H            ; TMOD = 01H, SET TIMER0 A 16-BIT TIMER
      MOV  TCON, #10H            ; TCON = 10H, TR0 = 1,GO
      MOV  PCON, #01H            ; ENTER IDLE MODE FOR LAUNCHING THE IN-SYSTEM
PROGRAMMING
;************* ********************************************************************
;* Normal mode 64KB APFLASH program: depending user's application
;*****************************************************************************
NORMAL_MODE:

                                 .
      ; User's application program
                                 .
                                 .
                                 .
```

**EXAMPLE 2:**

```
;*************************************************************************************************************************** ;*
Example of 4KB LDFLASH program: This loader program will erase the 64KB APFLASH first, then reads the new ;* code from external SRAM and program them into 64KB APFLASH bank. XTAL = 24 MHz
;***************************************************************************************************************************
.chip 8052
.RAMCHK OFF
.symbols

CHPCON    EQU        9FH
TA        EQU        C7H
SFRAL     EQU        ACH
SFRAH     EQU        ADH
SFRFD     EQU        AEH
SFRCN     EQU        AFH

          ORG 000H
          LJMP 100H              ; JUMP TO MAIN PROGRAM
;*************************************************************************
;* 1. TIMER0 SERVICE VECTOR ORG = 0BH
;*************************************************************************
          ORG 000BH
          CLR   TR0              ; TR0 = 0, STOP TIMER0
          MOV  TL0, R6
          MOV  TH0, R7
```

```
            RETI
;*********************************************************************
;*  4KB LDFLASH MAIN PROGRAM
;*********************************************************************
            ORG 100H
MAIN_4K:
        MOV  TA,#AAH
        MOV  TA,#55H
        MOV  CHPCON,#03H        ; CHPCON = 03H, ENABLE IN-SYSTEM PROGRAMMING.
        MOV  SFRCN,#0H
        MOV  TCON,#00H          ; TCON = 00H, TR = 0 TIMER0 STOP
        MOV  TMOD,#01H          ; TMOD = 01H, SET TIMER0 A 16BIT TIMER
        MOV  IP,#00H            ; IP = 00H
        MOV  IE,#82H            ; IE = 82H, TIMER0 INTERRUPT ENABLED
        MOV  R6,#F0H
             MOV  R7,#FFH
        MOV  TL0,R6
        MOV  TH0,R7
        MOV  TCON,#10H          ; TCON = 10H, TR0 = 1, GO
        MOV  PCON,#01H          ; ENTER IDLE MODE

 UPDATE_64K:
        MOV  TCON,#00H          ; TCON = 00H , TR = 0 TIM0 STOP
        MOV  IP,#00H            ; IP = 00H
        MOV  IE,#82H            ; IE = 82H, TIMER0 INTERRUPT ENABLED
        MOV  TMOD,#01H          ; TMOD = 01H, MODE1
        MOV  R6,#D0H            ; SET WAKE-UP TIME FOR ERASE OPERATION, ABOUT 15 ms
                                  DEPENDING ON USER'S SYSTEM CLOCK RATE.
        MOV  R7,#8AH
        MOV  TL0,R6
        MOV  TH0,R7

ERASE_P_4K:
        MOV  SFRCN,#22H         ; SFRCN = 22H, ERASE 64K  APFLASH0
                                ; SFRCN = A2H, ERASE 64K APFLASH1
        MOV  TCON,#10H          ; TCON = 10H, TR0 = 1,GO
        MOV  PCON,#01H          ; ENTER IDLE MODE (FOR ERASE OPERATION)

;*********************************************************************
;* BLANK CHECK
;*********************************************************************
        MOV  SFRCN,#0H          ; SFRCN = 00H, READ 64KB APFLASH0
                                ; SFRCN = 80H, READ 64KB APFLASH1
        MOV  SFRAH,#0H          ; START ADDRESS = 0H
        MOV  SFRAL,#0H
        MOV  R6,#FDH            ; SET TIMER FOR READ OPERATION, ABOUT 1.5 μS.
        MOV  R7,#FFH
        MOV  TL0,R6
        MOV  TH0,R7

BLANK_CHECK_LOOP:
        SETB  TR0               ; ENABLE TIMER 0
        MOV  PCON,#01H          ; ENTER IDLE MODE
        MOV  A,SFRFD            ; READ ONE BYTE
        CJNE  A,#FFH,BLANK_CHECK_ERROR
```

```
        INC  SFRAL                  ; NEXT ADDRESS
        MOV  A,SFRAL
        JNZ   BLANK_CHECK_LOOP
        INC  SFRAH
        MOV  A,SFRAH
        CJNE  A,#0H,BLANK_CHECK_LOOP   ; END ADDRESS = FFFFH
        JMP  PROGRAM_64KROM

BLANK_CHECK_ERROR:
         JMP  $

;*****************************************************************************
;* RE-PROGRAMMING 64KB APFLASH BANK
;*****************************************************************************
PROGRAM_64KROM:
         MOV  R2,#00H               ; TARGET LOW BYTE ADDRESS
         MOV  R1,#00H               ; TARGET HIGH BYTE ADDRESS
         MOV  DPTR,#0H
         MOV  SFRAH,R1              ; SFRAH, TARGET  HIGH ADDRESS
         MOV  SFRCN,#21H            ; SFRCN = 21H, PROGRAM 64K APFLASH0
                                    ; SFRCN = A1H, PROGRAM 64K APFLASH1
         MOV  R6,#9CH               ; SET TIMER FOR PROGRAMMING, ABOUT 50 μS.
         MOV  R7,#FFH
         MOV  TL0,R6
        MOV  TH0,R7

PROG_D_64K:
        MOV  SFRAL,R2               ; SFRAL = LOW BYTE ADDRESS
        CALL  GET_BYTE_FROM_PC_TO_ACC     ; THIS PROGRAM IS BASED ON USER'S CIRCUIT.
        MOV  @DPTR,A                ; SAVE DATA  INTO SRAM TO VERIFY CODE.
        MOV  SFRFD,A                ; SFRFD = DATA IN
        MOV  TCON,#10H              ; TCON = 10H, TR0 = 1,GO
        MOV  PCON,#01H              ; ENTER IDLE MODE (PRORGAMMING)
        INC  DPTR
        INC  R2
        CJNE  R2,#0H,PROG_D_64K
        INC  R1
        MOV  SFRAH,R1
        CJNE  R1,#0H,PROG_D_64K

;*****************************************************************************
; * VERIFY 64KB APFLASH BANK
;*****************************************************************************
        MOV  R4,#03H                ; ERROR COUNTER
        MOV  R6,#FDH                ; SET TIMER FOR READ VERIFY, ABOUT 1.5 μS.
        MOV  R7,#FFH
        MOV  TL0,R6
        MOV  TH0,R7
        MOV  DPTR,#0H               ; The start address of sample code
        MOV  R2,#0H                 ; Target  low byte address
        MOV  R1,#0H                 ; Target  high byte address
        MOV  SFRAH,R1               ; SFRAH, Target high address
        MOV  SFRCN,#00H             ; SFRCN = 00H, Read APFLASH0
                                    ;SFRCN = 80H , Read APFLASH1
```

```
READ_VERIFY_64K:
        MOV  SFRAL,R2               ; SFRAL = LOW ADDRESS
        MOV  TCON,#10H              ; TCON = 10H, TR0 = 1,GO
        MOV  PCON,#01H
        INC  R2
        MOVX  A,@DPTR
        INC  DPTR
        CJNE  A,SFRFD,ERROR_64K
        CJNE  R2,#0H,READ_VERIFY_64K
        INC  R1
        MOV  SFRAH,R1
        CJNE  R1,#0H,READ_VERIFY_64K

;*******************************************************************************
;* PROGRAMMING COMPLETLY, SOFTWARE RESET CPU
;*******************************************************************************
        MOV  TA,#AAH
        MOV  TA,#55H
        MOV  CHPCON,#83H            ; SOFTWARE RESET. CPU will restart from APFLASH0

ERROR_64K:
        DJNZ  R4,UPDATE_64K         ; IF ERROR OCCURS, REPEAT 3 TIMES.
.                                   ; IN-SYST PROGRAMMING FAIL, USER'S PROCESS TO DEAL WITH IT.
                        .
                        .
                        .
```

## 22. REVISION HISTORY

| VERSION | DATE | PAGE | DESCRIPTION |
|---|---|---|---|
| A1 | May 30, 2003 | - | Initial Issued |
| A2 | March 2, 2005 | 2 | Add lead free package part number |
| A3 | April 18, 2005 | 83 | Add Important Notice |
| A4 | Aug. 23, 2005 | 2, 4, 12<br>67 | Add Port 0 pull-up resisters information<br>Remove encrypt function of Security bits B2 description |
| A5 | Sep. 29, 2006 | | Remove block diagram |
| A6 | Dec. 4, 2006 | 2 | Remove all Leaded package parts |
| A7 | February 1, 2007 | 14 | Revise the Timer Mode Setting to “Mode 1: 16-bits, no prescale”. |
| A8 | April 17, 2007 | 45 | Revise that Power Down Mode is released by external interrupt configured as either level or edge detect. |
| A9 | November 19, 2007 | -<br>76 | Remove NVM description<br>Change chapter 19.1 Figure A to crystal connections |

## Important Notice

**Winbond products are not designed, intended, authorized or warranted for use as components in systems or equipment intended for surgical implantation, atomic energy control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, combustion control instruments, or for other applications intended to support or sustain life. Further more, Winbond products are not intended for applications wherein failure of Winbond products could result or lead to a situation wherein personal injury, death or severe property or environmental damage could occur.**

**Winbond customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Winbond for any damages resulting from such improper use or sales.**

*Please note that all data and specifications are subject to change without notice.*
*All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.*