**Welcome to E-XFL.COM**

## What is "Embedded - Microcontrollers"?

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

## Applications of "Embedded - Microcontrollers"

| Details | |
|---|---|
| Product Status | Active |
| Core Processor | 8051 |
| Core Size | 8-Bit |
| Speed | 40MHz |
| Connectivity | EBI/EMI, Serial Port |
| Peripherals | POR, WDT |
| Number of I/O | 36 |
| Program Memory Size | 32KB (32K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 1K x 8 |
| Voltage - Supply (Vcc/Vdd) | 4.5V ~ 5.5V |
| Data Converters | - |
| Oscillator Type | Internal |
| Operating Temperature | 0°C ~ 70°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 44-BQFP |
| Supplier Device Package | - |
| Purchase URL | https://www.e-xfl.com/product-detail/nuvoton-technology-corporation-america/w77e058a40fl |

## 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 and the program counter is within 32 KB area. Otherwise they will be present on the bus. |
| $\overline{PSEN}$ | 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. |
| $V_{SS}$ | I | **GROUND:** Ground potential |
| $V_{DD}$ | 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. |
| 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 1 RXD<br>TXD1(P1.3): Serial port 1 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. |

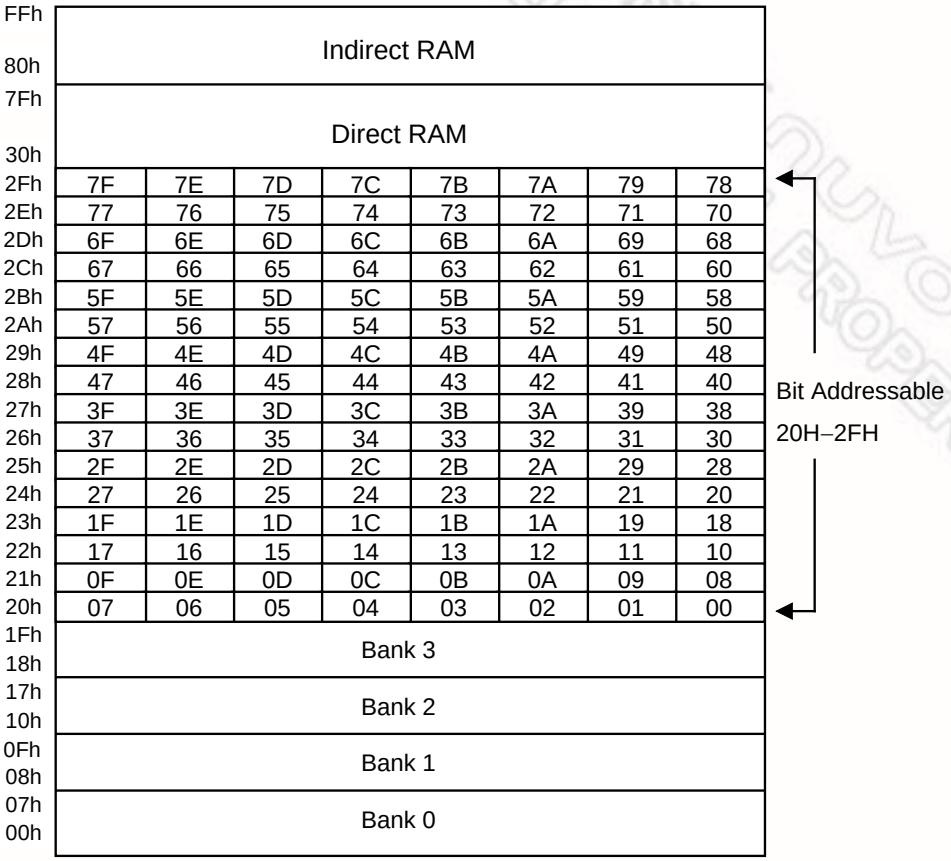



Figure 2. Scratchpad RAM/Register Addressing

## 7. SPECIAL FUNCTION REGISTERS

The W77E058 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 W77E058 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 | ROMMAP | | PMR | STATUS | | TA |
| B8 | IP | SADEN | SADEN1 | | | | | |
| B0 | P3 | | | | | | | |
| A8 | IE | SADDR | SADDR1 | | | | | |
| A0 | P2 | | | | | P4 | | |
| 98 | SCON0 | SBUF | | | | | | |
| 90 | P1 | EXIF | | | | | | |
| 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.

M1, M0: Mode Select bits:

| M1 | M0 | Mode |
|---|---|---|
| 0 | 0 | Mode 0: 8-bits with 5-bit prescale. |
| 0 | 1 | Mode 1: 18-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

**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 Iinterrupt Flag**

| Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | IE5 | IE4 | IE3 | IE2 | XT/$\overline{\text{RG}}$ | RGMD | RGSL | - |

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.

XT/$\overline{\text{RG}}$: Crystal/RC Oscillator Select. Setting this bit selects crystal or external clock as system clock source. Clearing this bit selects the on-chip RC oscillator as clock source. XTUP(STATUS.4) must be set to 1 and XTOFF (PMR.3) must be cleared before this bit can be set. Attempts to set this bit without obeying these conditions will be ignored. This bit is set to 1 after a power-on reset and unchanged by other forms of reset.

RGMD: RC Mode Status. This bit indicates the current clock source of microcontroller. When cleared, CPU is operating from the external crystal or oscillator. When set, CPU is operating from the on-chip RC oscillator. This bit is cleared to 0 after a power-on reset and unchanged by other forms of reset.

RGSL: RC Oscillator Select. This bit selects the clock source following a resume from Power Down Mode. Setting this bit allows device operating from RC oscillator when a resume from Power Down Mode. When this bit is cleared, the device will hold operation until the crystal oscillator has warmed-up following a resume from Power Down Mode. This bit is cleared to 0 after a power-on reset and unchanged by other forms of reset.

## 8. INSTRUCTION

The W77E058 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 W77E058, each machine cycle consists of 4 clock periods, while in the standard 8032 it consists of 12 clock periods. Also, in the W77E058 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 W77E058 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 W77E058 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.

**Table 3. Instruction Timing for W77E058**

| INSTRUCTION | HEX OP-CODE | BYTES | W77E058 MACHINE CYCLES | W77E058 CLOCK CYCLES | 8032 CLOCK CYCLES | W77E058 VS. 8032 SPEED RATIO |
|---|---|---|---|---|---|---|
| NOP | 00 | 1 | 1 | 4 | 12 | 3 |
| ADD A, R0 | 28 | 1 | 1 | 4 | 12 | 3 |
| ADD A, R1 | 29 | 1 | 1 | 4 | 12 | 3 |
| ADD A, R2 | 2A | 1 | 1 | 4 | 12 | 3 |
| ADD A, R3 | 2B | 1 | 1 | 4 | 12 | 3 |
| ADD A, R4 | 2C | 1 | 1 | 4 | 12 | 3 |
| ADD A, R5 | 2D | 1 | 1 | 4 | 12 | 3 |
| ADD A, R6 | 2E | 1 | 1 | 4 | 12 | 3 |
| ADD A, R7 | 2F | 1 | 1 | 4 | 12 | 3 |
| ADD A, @R0 | 26 | 1 | 1 | 4 | 12 | 3 |
| ADD A, @R1 | 27 | 1 | 1 | 4 | 12 | 3 |
| ADD A, direct | 25 | 2 | 2 | 8 | 12 | 1.5 |
| ADD A, #data | 24 | 2 | 2 | 8 | 12 | 1.5 |

Table 3. Instruction Timing for W77E058, continued

| INSTRUCTION | HEX OP-CODE | BYTES | W77E058 MACHINE CYCLES | W77E058 CLOCK CYCLES | 8032 CLOCK CYCLES | W77E058 VS. 8032 SPEED RATIO |
|---|---|---|---|---|---|---|
| MOVX A, @R0 | E2 | 1 | 2 - 9 | 8 - 36 | 24 | 3 - 0.66 |
| MOVX A, @R1 | E3 | 1 | 2 - 9 | 8 - 36 | 24 | 3 - 0.66 |
| MOVX A, @DPTR | E0 | 1 | 2 - 9 | 8 - 36 | 24 | 3 - 0.66 |
| MOVX @R0, A | F2 | 1 | 2 - 9 | 8 - 36 | 24 | 3 - 0.66 |
| MOVX @R1, A | F3 | 1 | 2 - 9 | 8 - 36 | 24 | 3 - 0.66 |
| MOVX @DPTR, A | F0 | 1 | 2 - 9 | 8 - 36 | 24 | 3 - 0.66 |
| MOV C, bit | A2 | 2 | 2 | 8 | 12 | 1.5 |
| MOV bit, C | 92 | 2 | 2 | 8 | 24 | 3 |
| ORL A, R0 | 48 | 1 | 1 | 4 | 12 | 3 |
| ORL A, R1 | 49 | 1 | 1 | 4 | 12 | 3 |
| ORL A, R2 | 4A | 1 | 1 | 4 | 12 | 3 |
| ORL A, R3 | 4B | 1 | 1 | 4 | 12 | 3 |
| ORL A, R4 | 4C | 1 | 1 | 4 | 12 | 3 |
| ORL A, R5 | 4D | 1 | 1 | 4 | 12 | 3 |
| ORL A, R6 | 4E | 1 | 1 | 4 | 12 | 3 |
| ORL A, R7 | 4F | 1 | 1 | 4 | 12 | 3 |
| ORL A, @R0 | 46 | 1 | 1 | 4 | 12 | 3 |
| ORL A, @R1 | 47 | 1 | 1 | 4 | 12 | 3 |
| ORL A, direct | 45 | 2 | 2 | 8 | 12 | 1.5 |
| ORL A, #data | 44 | 2 | 2 | 8 | 12 | 1.5 |
| ORL direct, A | 42 | 2 | 2 | 8 | 12 | 1.5 |
| ORL direct, #data | 43 | 3 | 3 | 12 | 24 | 2 |
| ORL C, bit | 72 | 2 | 2 | 8 | 24 | 3 |
| ORL C, /bit | A0 | 2 | 2 | 6 | 24 | 3 |
| PUSH direct | C0 | 2 | 2 | 8 | 24 | 3 |
| POP direct | D0 | 2 | 2 | 8 | 24 | 3 |
| RET | 22 | 1 | 2 | 8 | 24 | 3 |
| RETI | 32 | 1 | 2 | 8 | 24 | 3 |
| RL A | 23 | 1 | 1 | 4 | 12 | 3 |
| RLC A | 33 | 1 | 1 | 4 | 12 | 3 |
| RR A | 03 | 1 | 1 | 4 | 12 | 3 |
| RRC A | 13 | 1 | 1 | 4 | 12 | 3 |
| SETB C | D3 | 1 | 1 | 4 | 12 | 3 |
| SETB bit | D2 | 2 | 2 | 8 | 12 | 1.5 |
| SWAP A | C4 | 1 | 1 | 4 | 12 | 3 |
| SJMP rel | 80 | 2 | 3 | 12 | 24 | 2 |
| SUBB A, R0 | 98 | 1 | 1 | 4 | 12 | 3 |

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 |

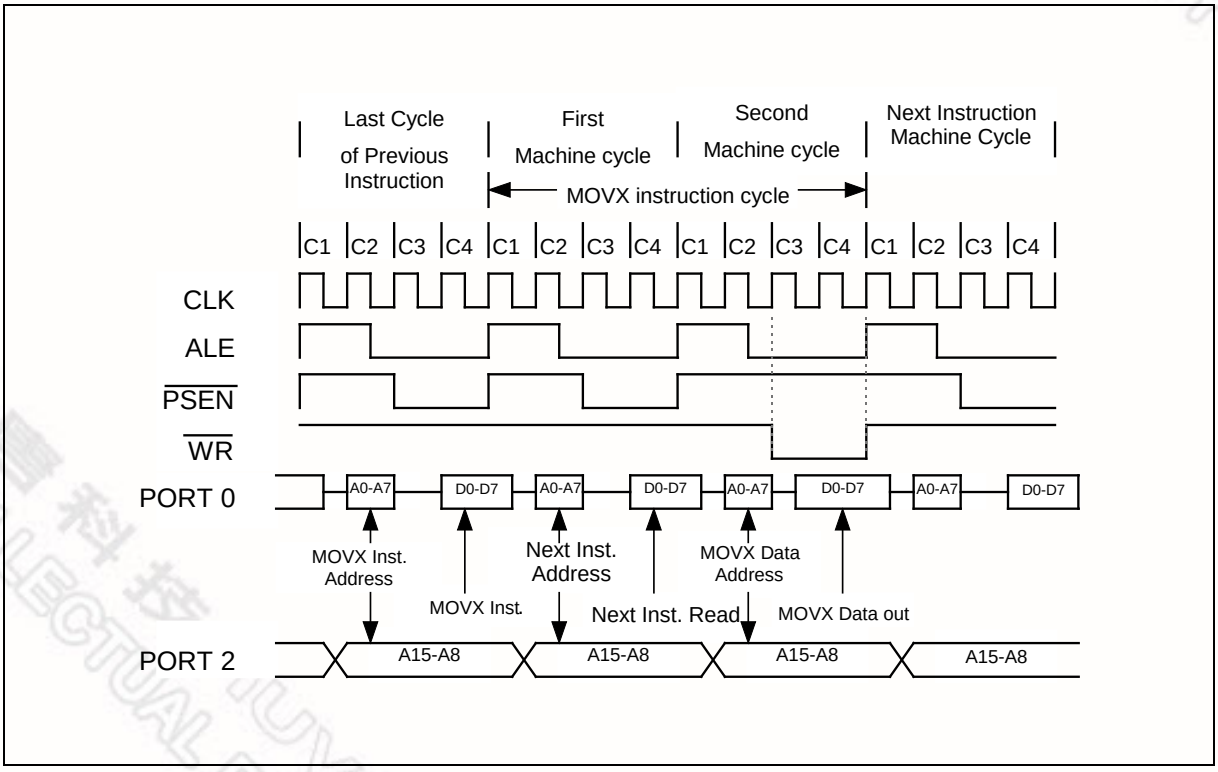



Figure 8. Data Memory Write with Stretch Value = 0

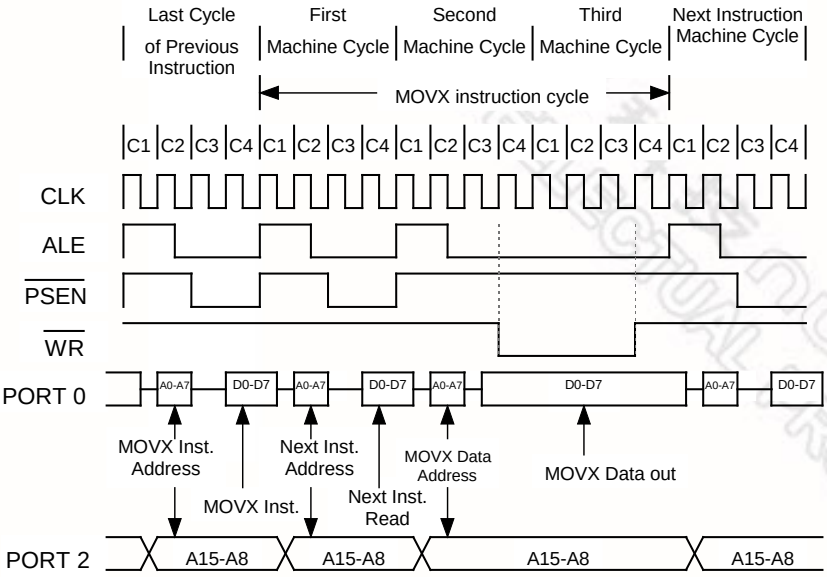



Figure 9. Data Memory Write with Stretch Value = 1

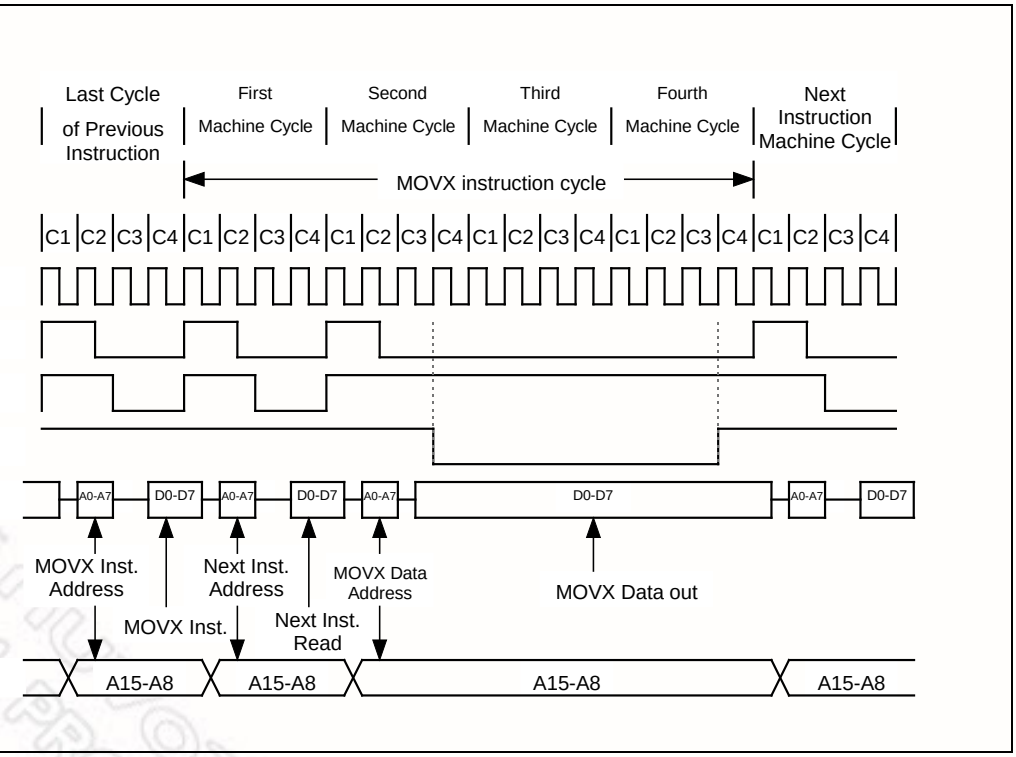



Figure 10. Data Memory Write with Stretch Value = 2

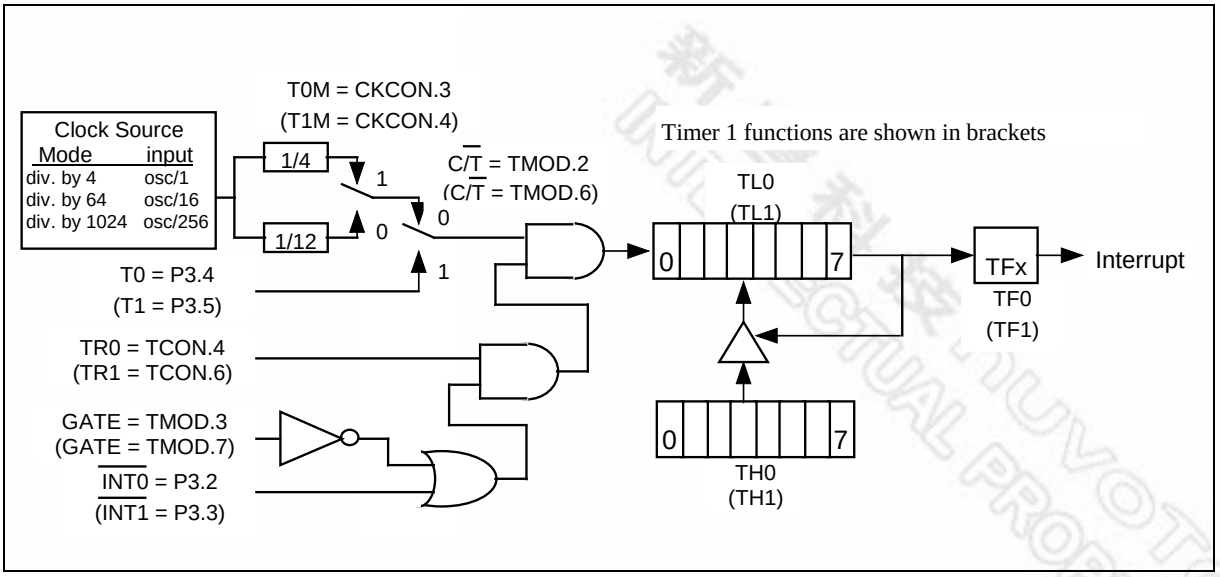



Figure 12. Timer/Counter Mode 2.

### 12.2.4 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.

### 12.3.5 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 (RCAP2H, RCAP2L) ]**

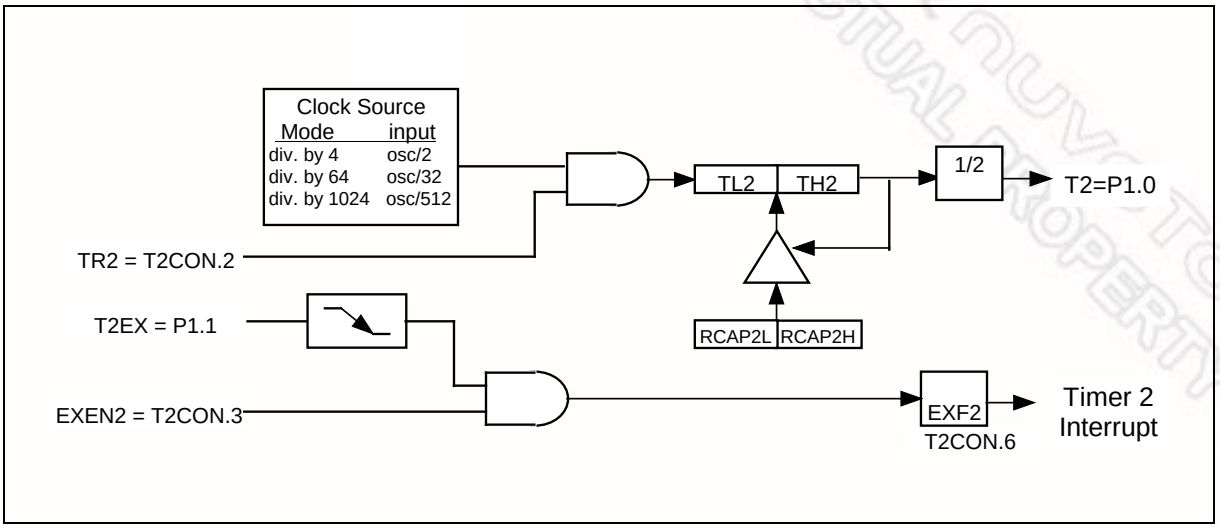



Figure 18. Programmable Clock-Out Mode

## 13. WACHDOG 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.

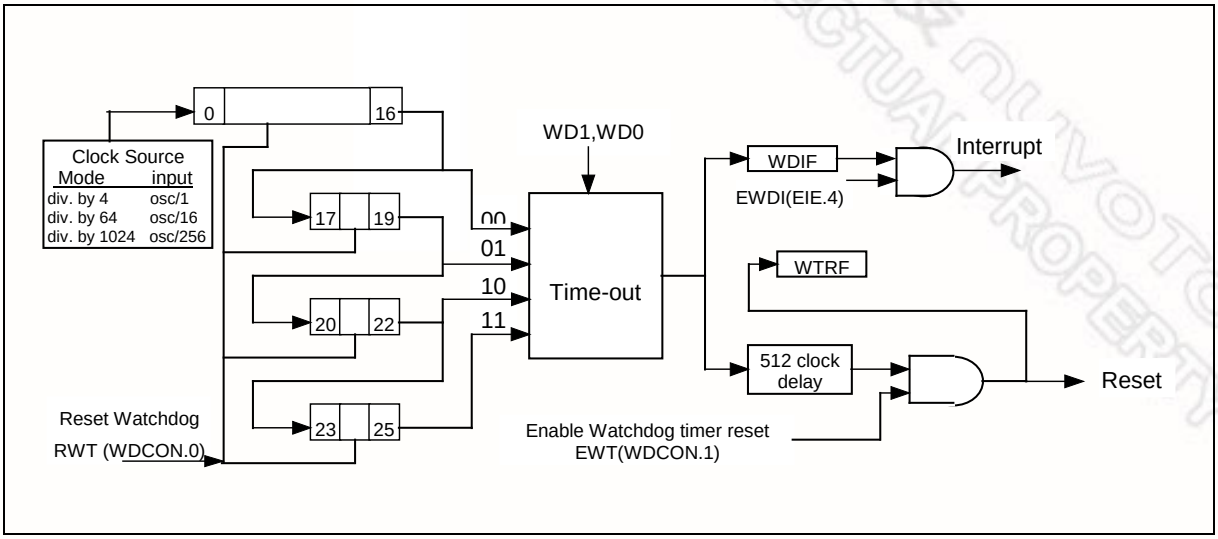



Figure 19. Watchdog Timer

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

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.

## 14.4 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.

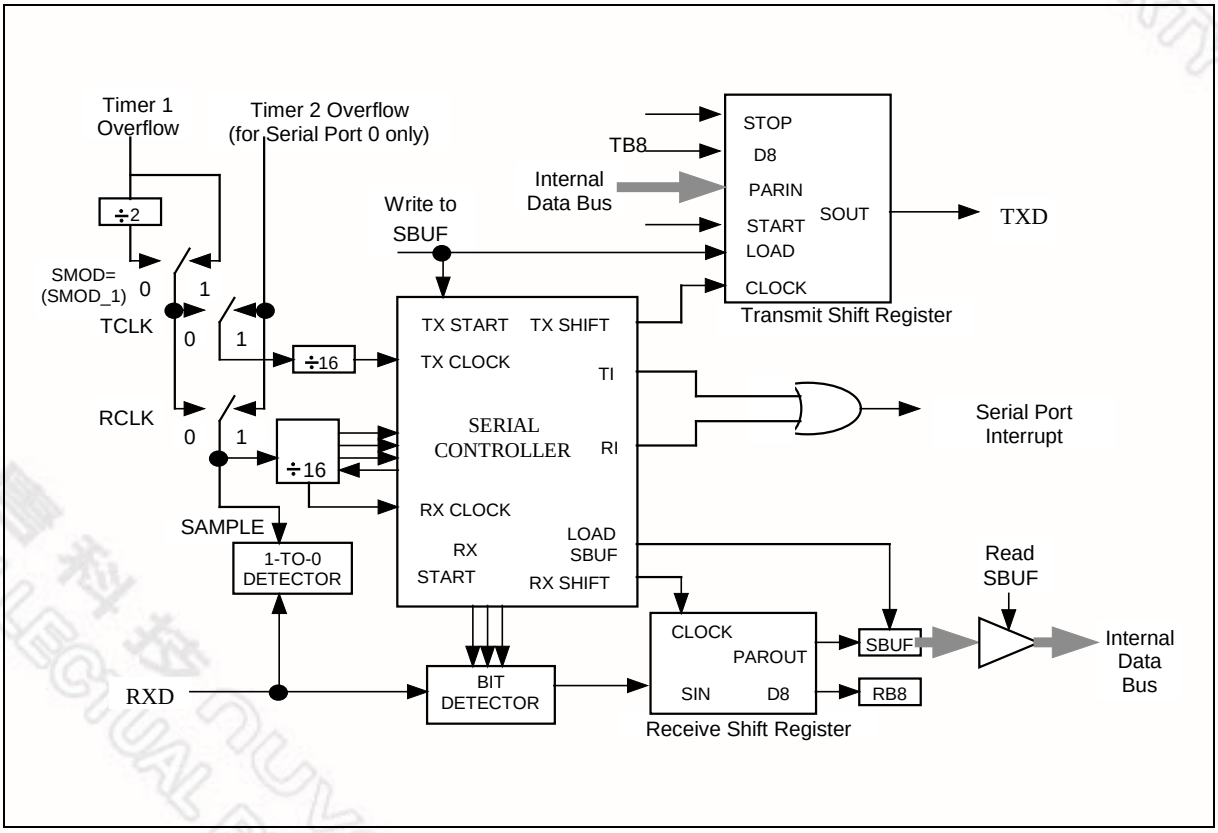



Figure 23. Serial Port Mode 3

**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 |

## 14.5 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 W77E058 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 W77E058 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.

## 14.6 Multiprocessor Communications

Multiprocessor communications makes use of the 9th data bit in modes 2 and 3. In the W77E058, 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 9th 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 9th 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.

## 15. TIMED ACCESS PROTECTION

The W77E058 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 W77E058 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
MOV     TA, #055h       3 M/C
MOV     WDCON, #00h     3 M/C
```

Note: M/C = Machine Cycles

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.

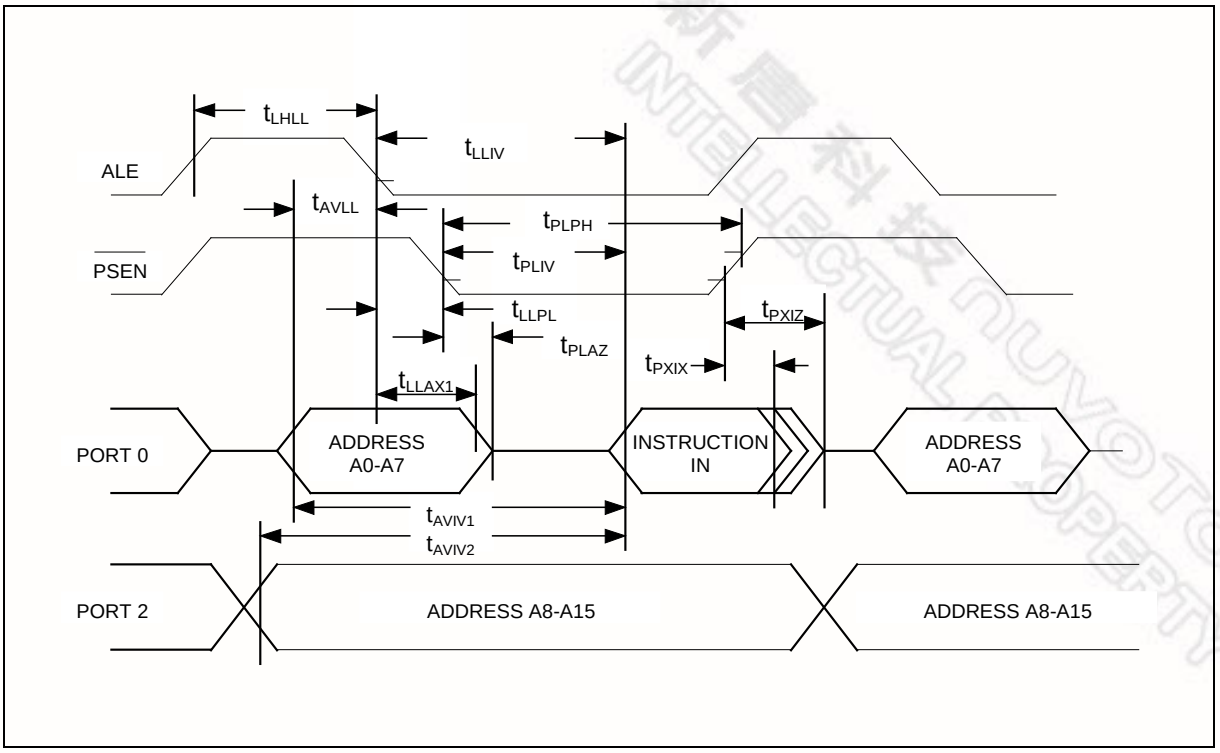
### B0: Lock bit

This bit is used to protect the customer's program code in the W77E058. 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.

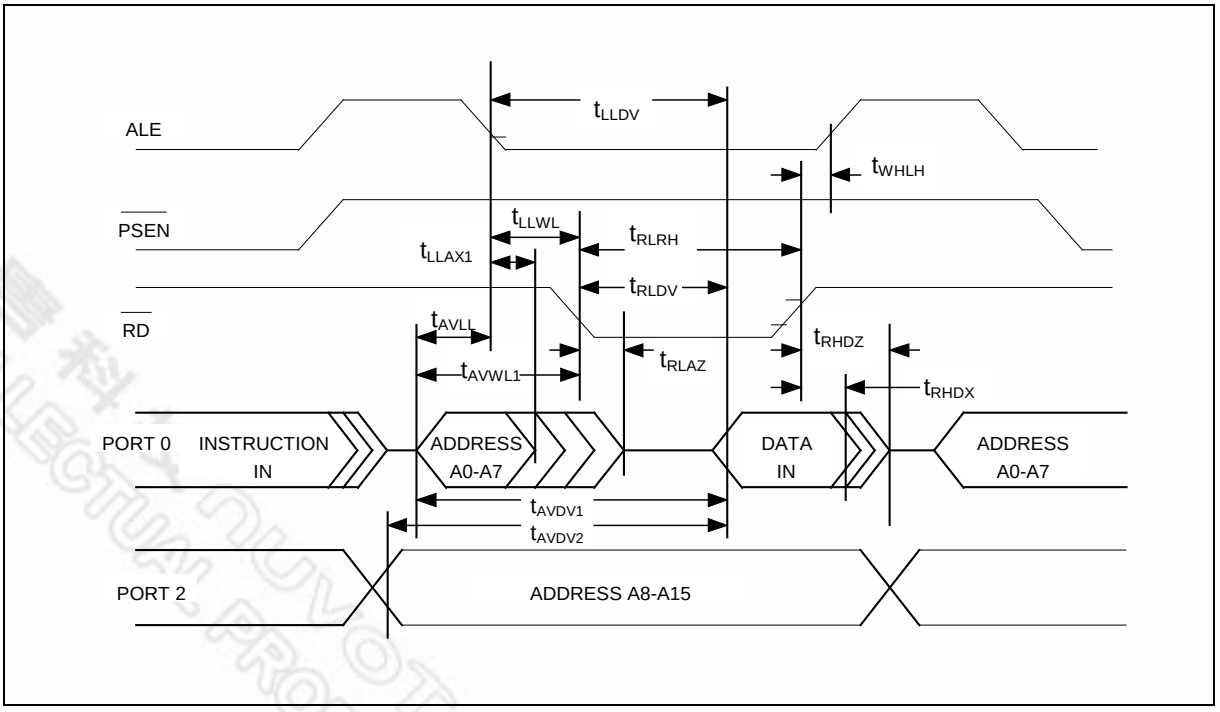
### B1: MOVC Inhibit

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.

## Program Memory Read Cycle




## Data Memory Read Cycle



## Data Memory Write Cycle