Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	8051
Core Size	8-Bit
Speed	25MHz
Connectivity	EBI/EMI, Serial Port
Peripherals	POR, WDT
Number of I/O	32
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Through Hole
Package / Case	40-DIP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/nuvoton-technology-corporation-america/w77l516a25dl

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

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.

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.

TIMER 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.

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 discription.

```
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
```

8. INSTRUCTION TIMING

The instruction timing for the W77L516A 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 W77L516A and the standard 8032. In the W77L516A 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 W77L516A 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 W77L516A 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 W77L516A, 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 W77L516A, based on the number of machine cycles, there are five different types, while in the standard 8032 there are only three. However, in the W77L516A 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.

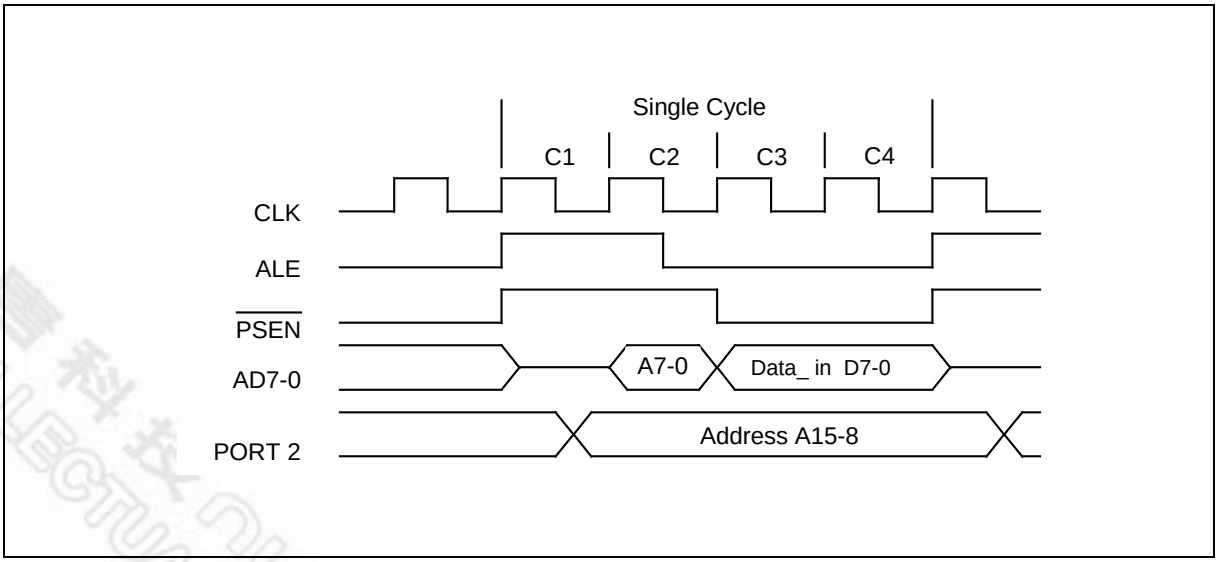


Figure 3. Single Cycle Instruction Timing

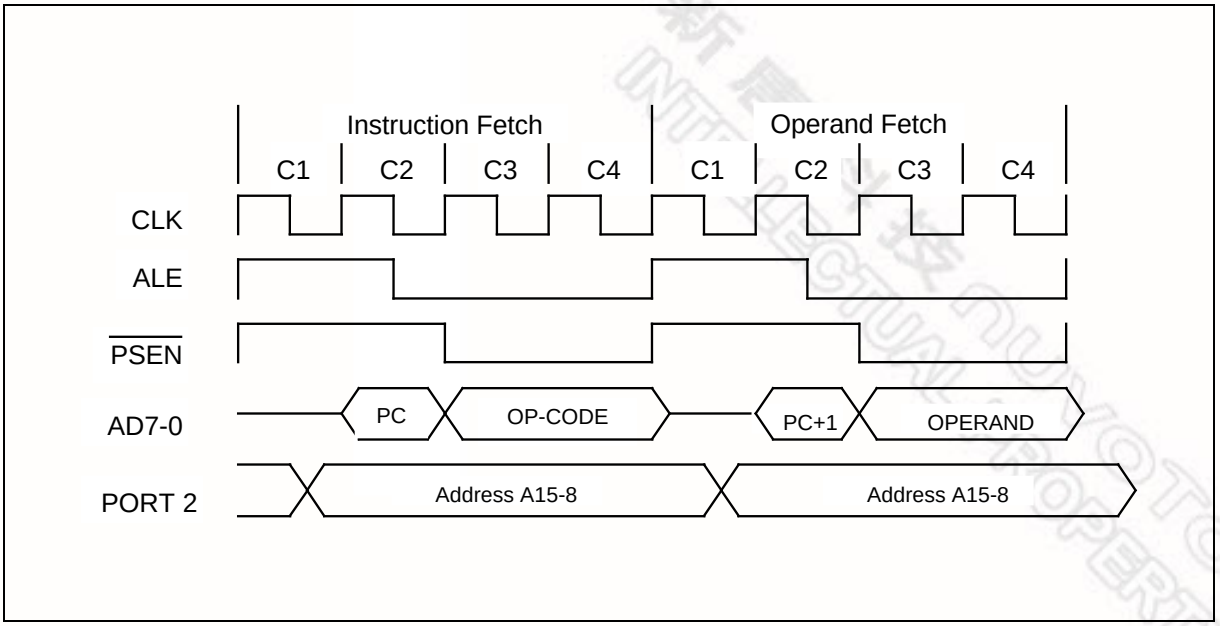


Figure 4. Two Cycle Instruction Timing

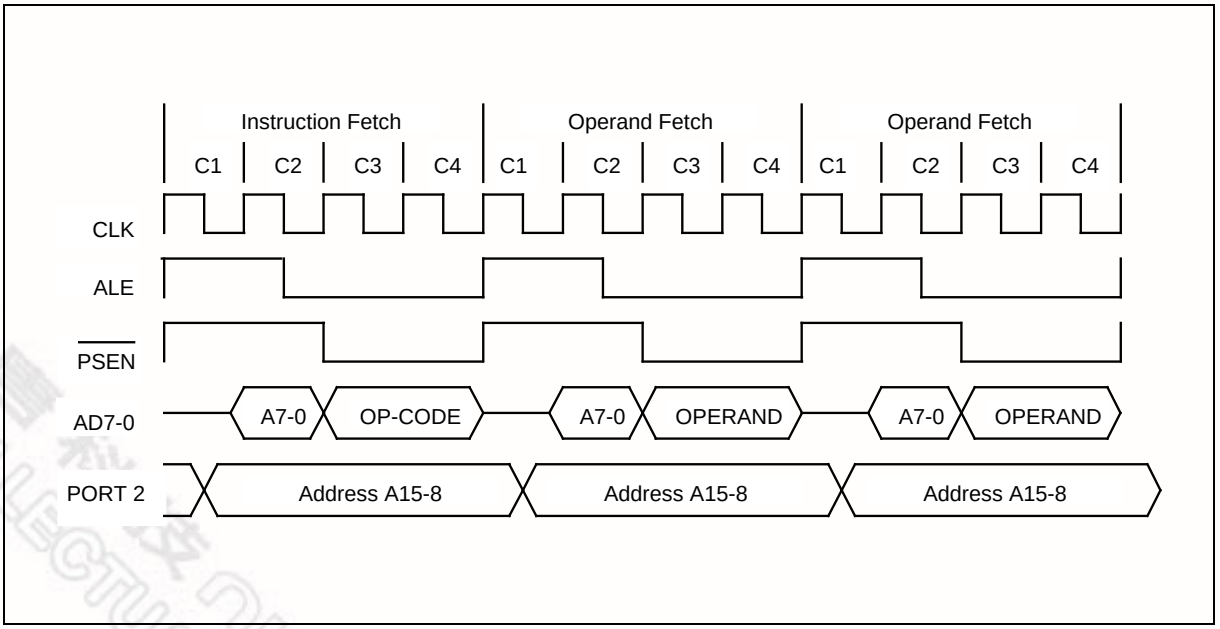


Figure 5. Three Cycle Instruction Timing

Machine cycles in standard 8032 = 10 + (26 * CNT)
Machine cycles in W77L516A = 10 + (26 * CNT)
If CNT = 50
Clock cycles in standard 8032= ((10 + (26 *50)) * 12 = (10 + 1300) * 12 = 15720
Clock cycles in W77L516A = ((10 + (26 * 50)) * 4 = (10 + 1300) * 4 = 5240

Block Move with Two Data Pointers in W77L516A:

; 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 W77L516A #
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:			
MOVX	A, @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 W77L516A = 12 + (15 * CNT)
If CNT = 50
Clock cycles in W77L516A = (12 + (15 * 50)) * 4 = (12 + 750) * 4 = 3048

We can see that in the first program the standard 8032 takes 15720 cycles, while the W77L516A takes only 5240 cycles for the same code. In the second program, written for the W77L516A, program execution requires only 3048 clock cycles. If the size of the block is increased then the saving is even greater.

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 W77L516A 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.

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.

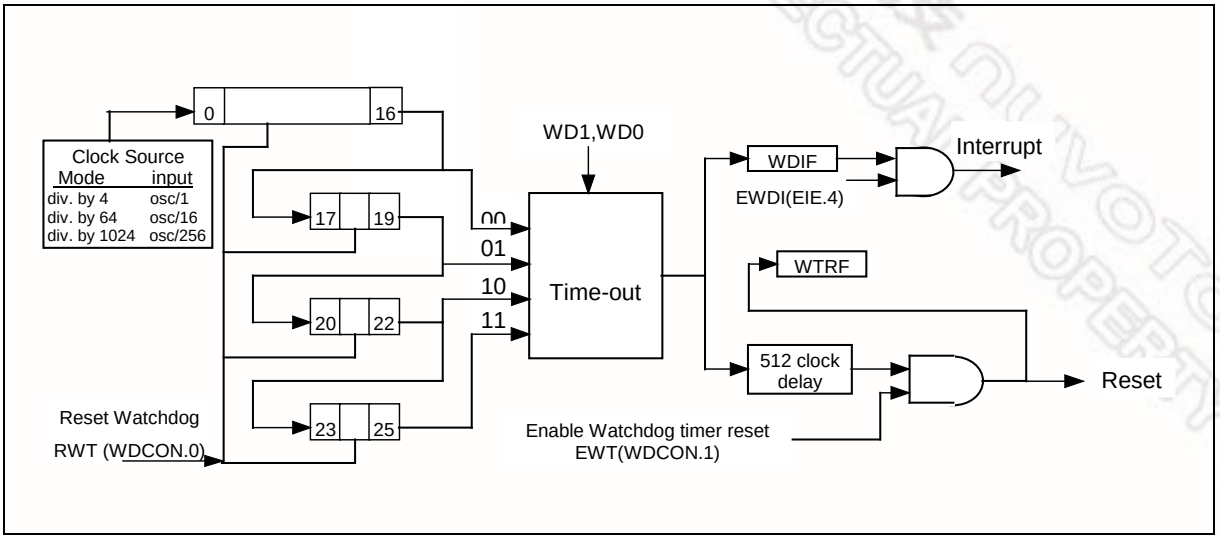


Figure 19. Watchdog Timer

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

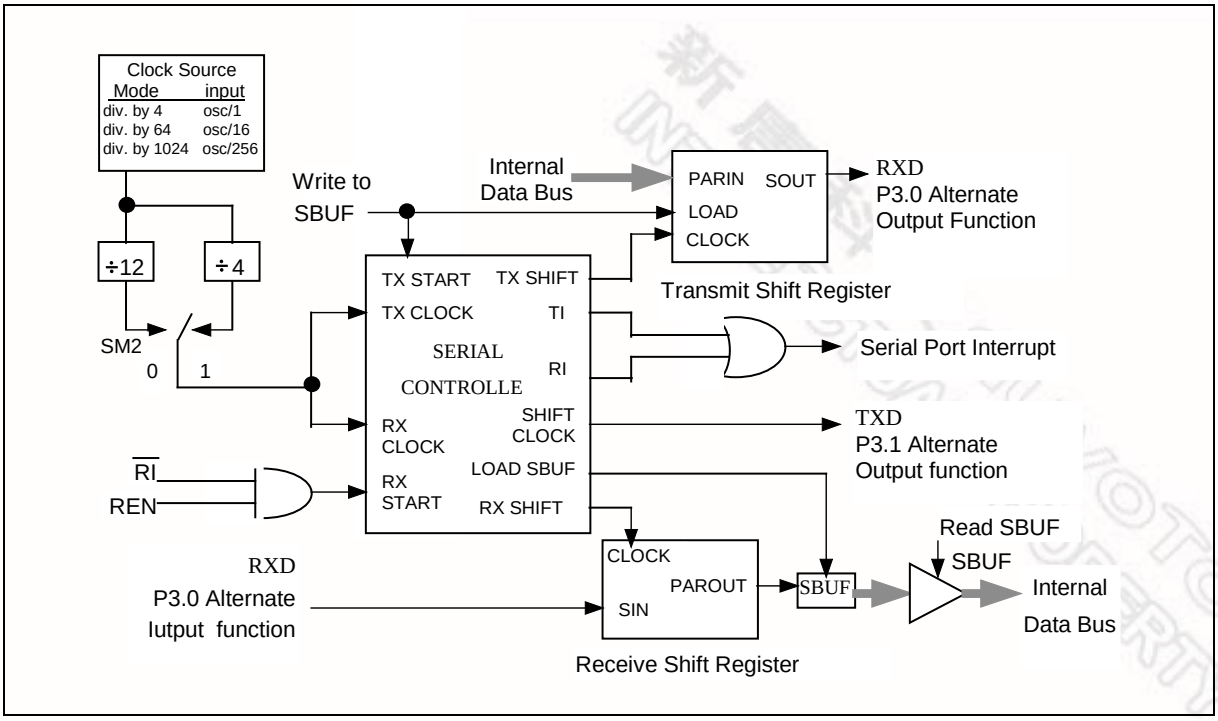


Figure 20. Serial Port Mode 1

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 10th 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.

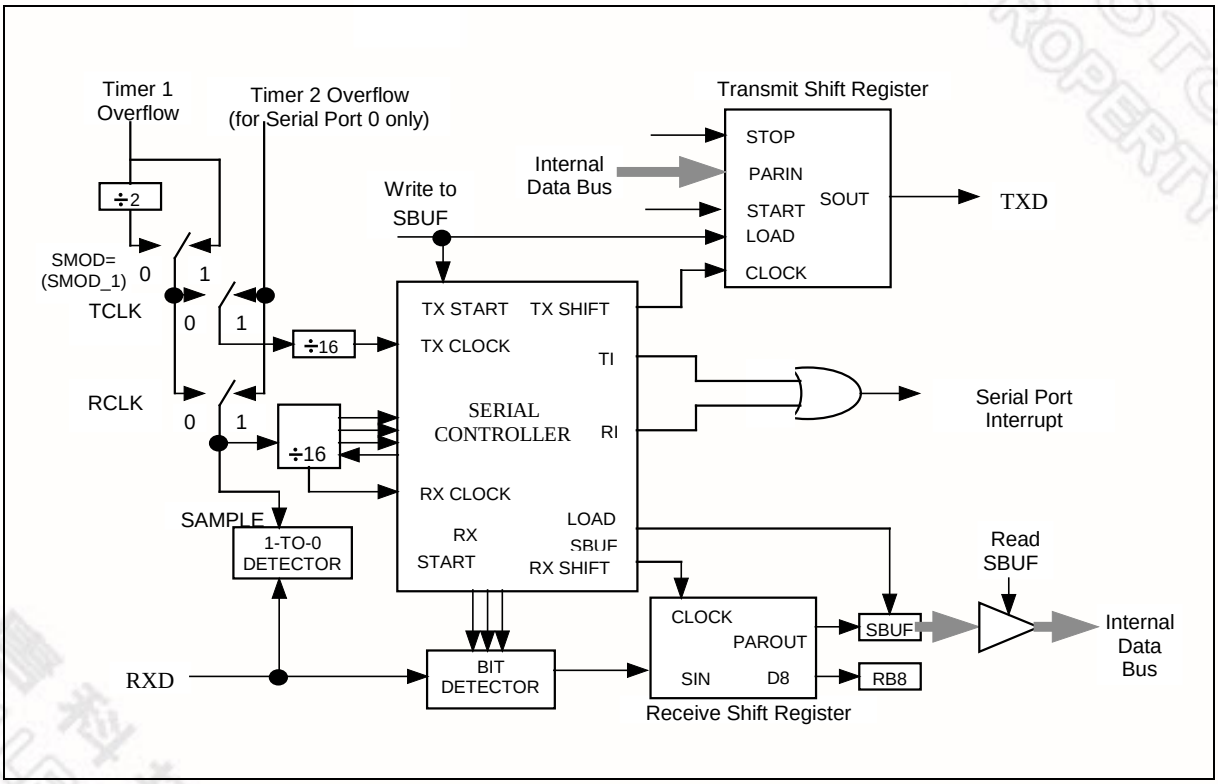


Figure 21. Serial Port Mode 1

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

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 W77L516A 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 W77L516A 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

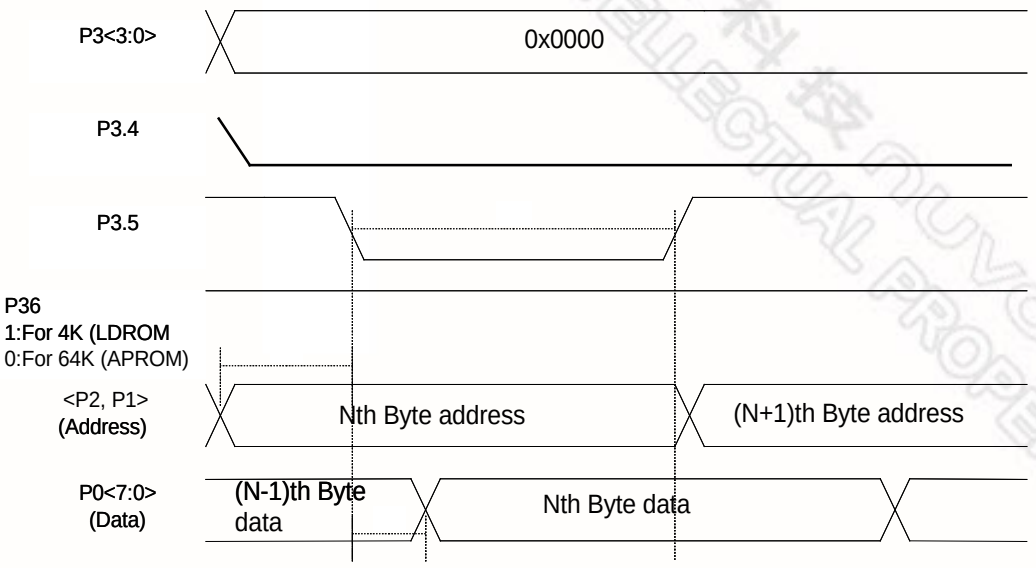
Multiprocessor communications makes use of the 9th data bit in modes 2 and 3. In the W77L516A, the RI flag is set only if the received byte corresponds to the Given or Broadcast address. This hardware feature eliminates the software overhead required in checking every received address, and greatly simplifies 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 Read Timing



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 W77L516A 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 W77L516A. 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.

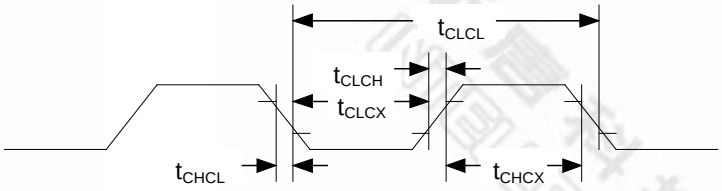
B4: H/W Reboot with P2.6 and P2.7

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.

18.3 AC Characteristics



Note: Duty cycle is 50%.

External Clock Characteristics

PARAMETER	SYMBOL	MIN.	TYP.	MAX.	UNITS	NOTES
Clock High Time	t_{CHCX}	25	-	-	nS	
Clock Low Time	t_{CLCX}	25	-	-	nS	
Clock Rise Time	t_{CLCH}	-	-	10	nS	
Clock Fall Time	t_{CHCL}	-	-	10	nS	

AC Specification

PARAMETER	SYMBOL	VARIABLE CLOCK MIN.	VARIABLE CLOCK MAX.	UNITS
Oscillator Frequency	$1/t_{CLCL}$	0	20	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
$\overline{PSEN}$ Pulse Width	t_{PLPH}	$2.0t_{CLCL}$ - 5		nS
$\overline{PSEN}$ 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
$\overline{PSEN}$ 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
$\overline{RD}$ Low to Address Float	t_{RLAZ}		$0.5t_{CLCL}$ - 5	nS

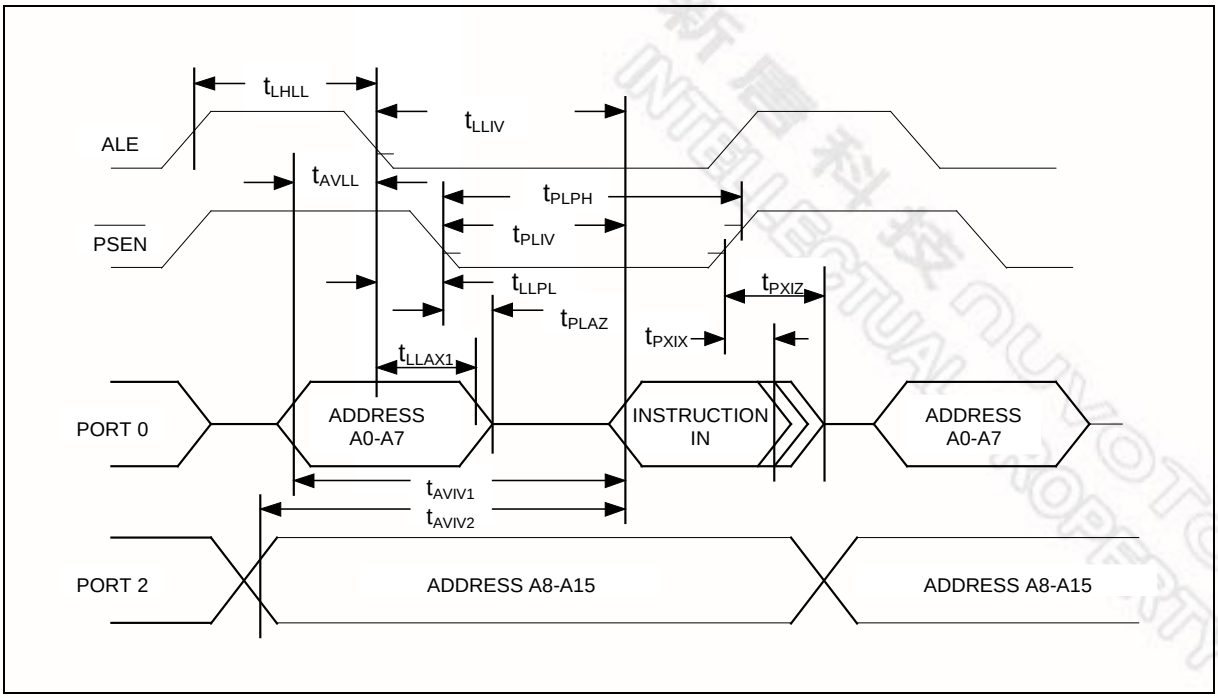
M2	M1	M0	MOVX CYCLES	T_{MCS}
0	0	0	2 machine cycles	0
0	0	1	3 machine cycles	4 t_{CLCL}
0	1	0	4 machine cycles	8 t_{CLCL}
0	1	1	5 machine cycles	12 t_{CLCL}
1	0	0	6 machine cycles	16 t_{CLCL}
1	0	1	7 machine cycles	20 t_{CLCL}
1	1	0	8 machine cycles	24 t_{CLCL}
1	1	1	9 machine cycles	28 t_{CLCL}

EXPLANATION OF LOGIC SYMBOLS

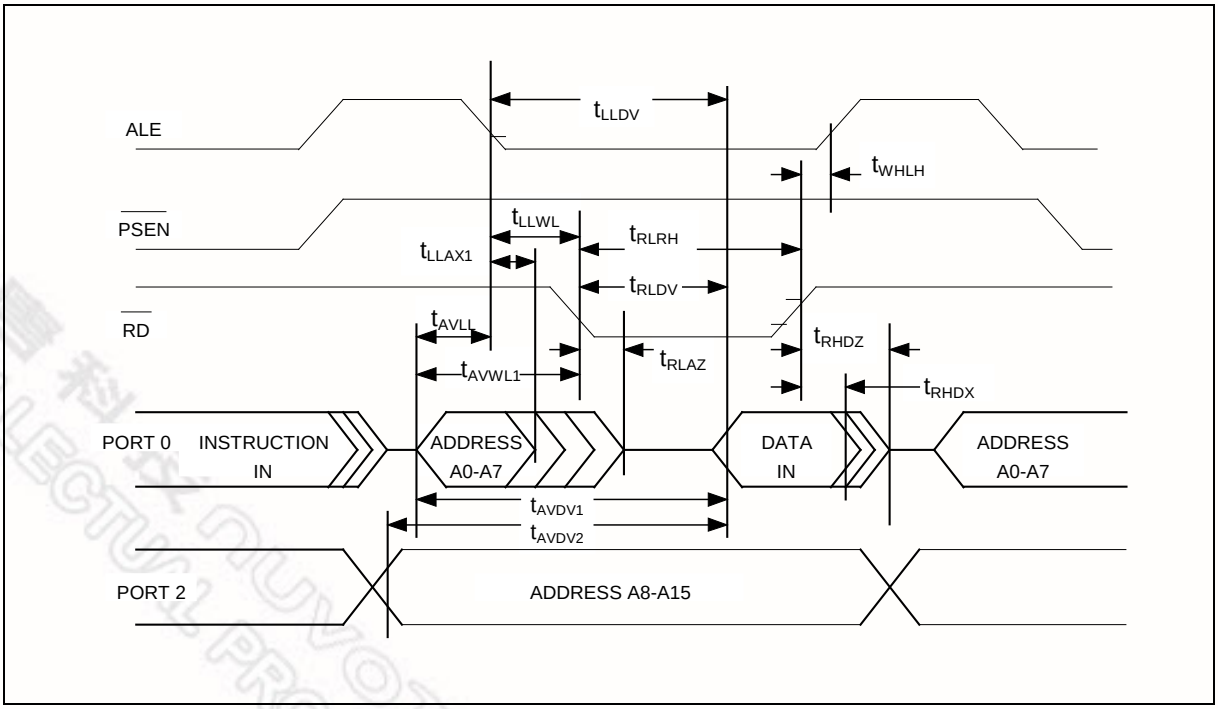
In order to maintain compatibility with the original 8051 family, this device specifies the same parameter for each device, using the same symbols. The explanation of the symbols is as follows.

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

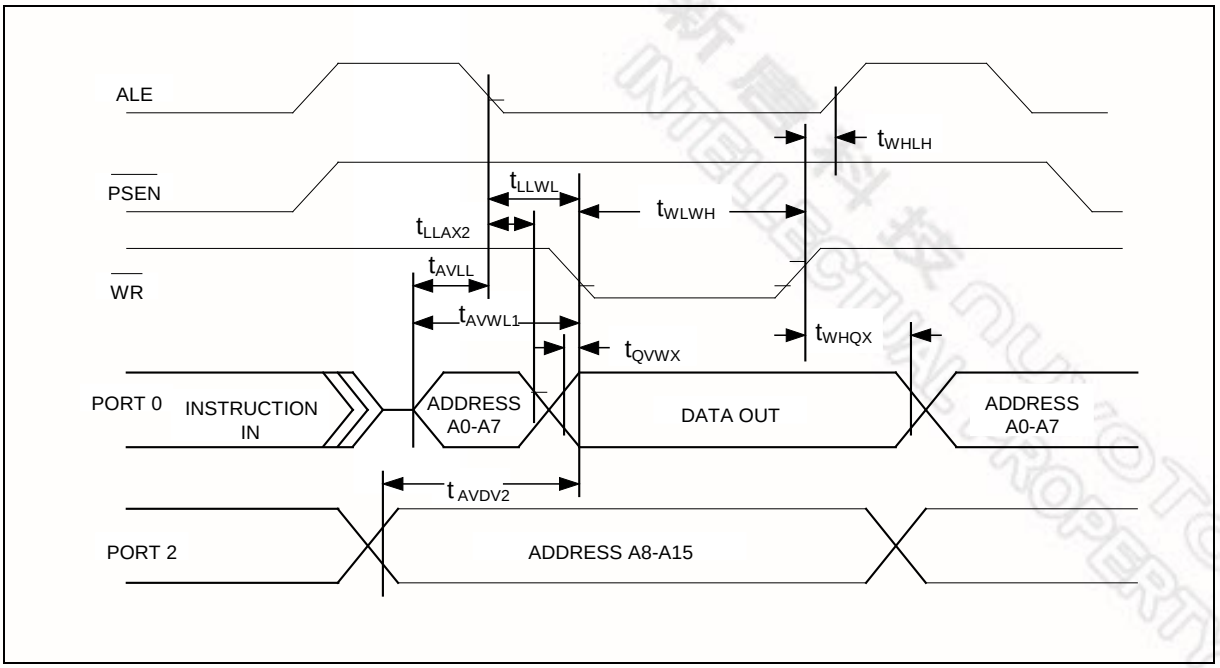
PROGRAM MEMORY READ CYCLE



DATA MEMORY READ CYCLE



DATA MEMORY WRITE CYCLE



```
        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:
```