Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

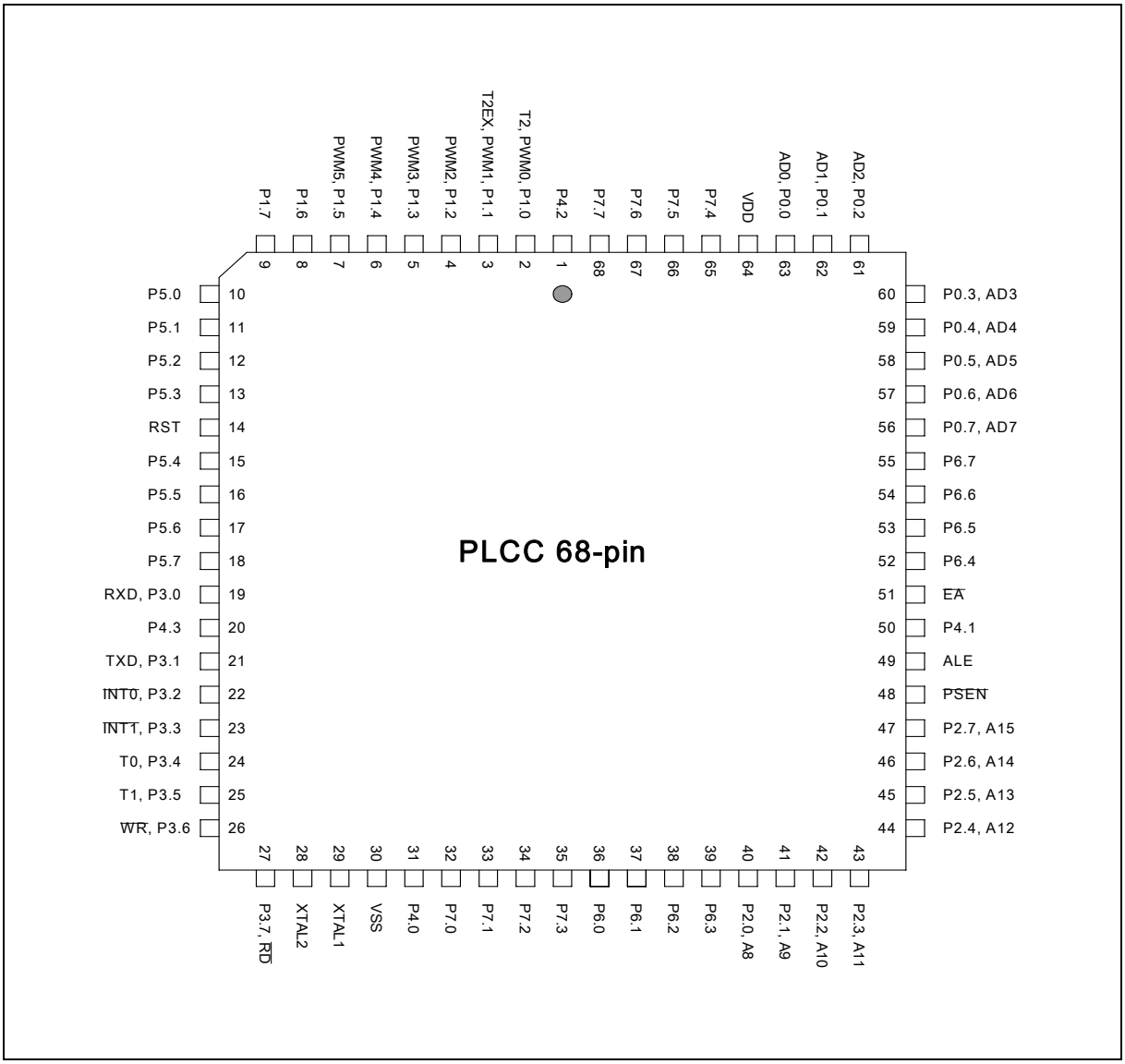
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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"

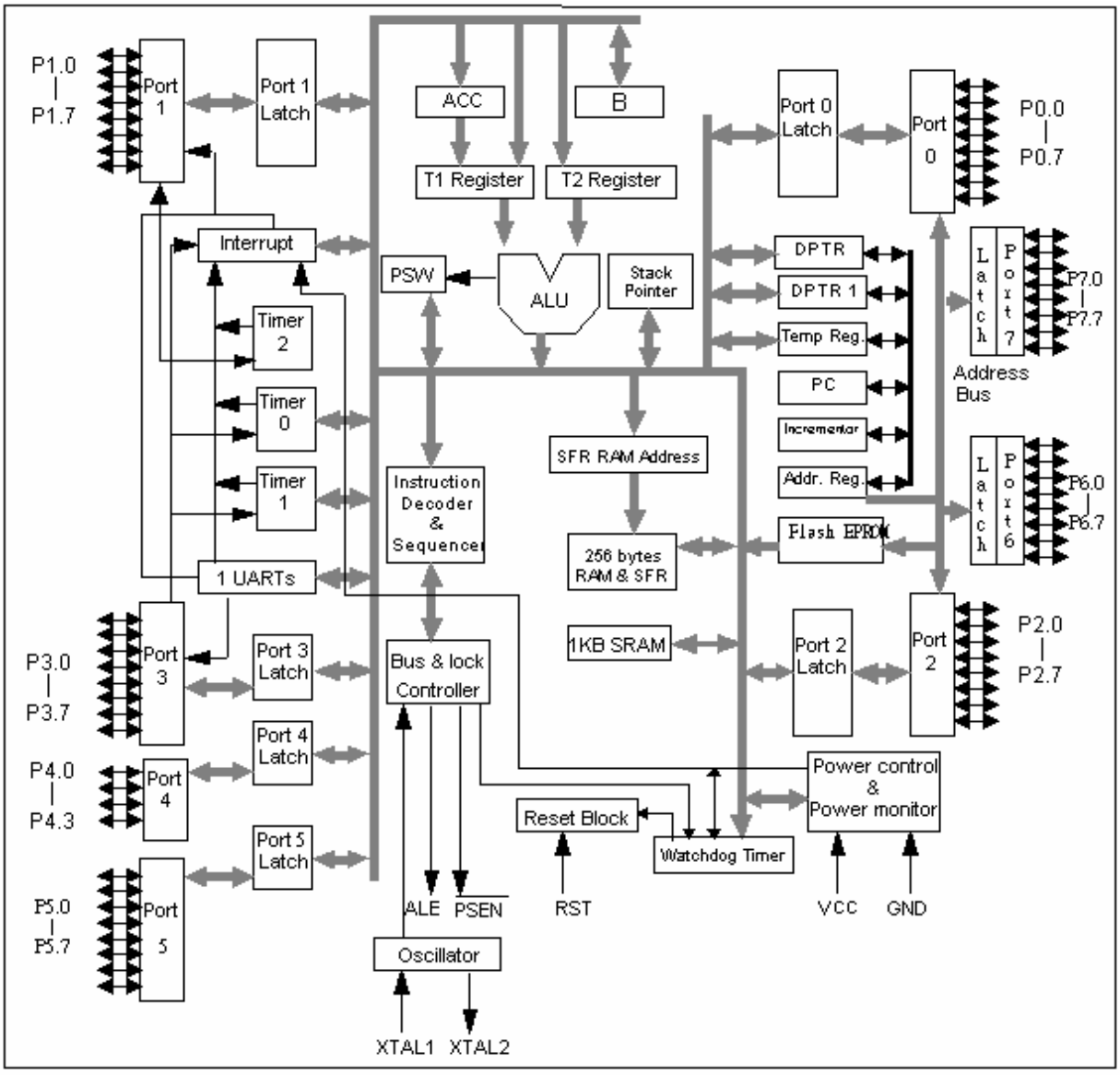
Details	
Product Status	Obsolete
Core Processor	8052
Core Size	8-Bit
Speed	40MHz
Connectivity	EBI/EMI, Serial Port
Peripherals	POR, PWM, WDT
Number of I/O	60
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1.25K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	68-PLCC
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/nuvoton-technology-corporation-america/w79e548a40pl

DEVICE	OPERATING FREQUENCY	OPERATING VOLTAGE	PACKAGE	
			NOMAL	LEAD FREE(ROHS)
W79E548	up to 40MHz	4.5V ~ 5.5V	PLCC68	PLCC68
W79L548	up to 25MHz	3.0V ~ 5.5V	PLCC68	PLCC68

3. PIN CONFIGURATIONS



5. BLOCK DIAGRAM



PD: Setting this bit causes the W79E(L)548 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 W79E(L)548 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: 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

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.

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	-	-	-	-

Mnemonic: EIE Address: E8h

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

EWDI: Enable Watchdog timer interrupt

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	-	-	-	-

Mnemonic: EIP Address: F8h

EIP.7-5: Reserved bits.

PWDI: Watchdog timer interrupt priority.

8. INSTRUCTION

The W79E(L)548 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 W79E(L)548, each machine cycle consists of 4 clock periods, while in the standard 8032 it consists of 12 clock periods. Also, in the W79E(L)548 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 W79E(L)548 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 W79E(L)548 reduces the number of dummy fetches and wasted cycles, thereby improving efficiency as compared to the standard 8032.

8.1 Instruction Timing

The instruction timing for the W79E(L)548 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 W79E(L)548 and the standard 8032. In the W79E(L)548 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 W79E(L)548 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 W79E(L)548 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 W79E(L)548, 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 W79E(L)548, based on the number of machine cycles, there are five different types, while in the standard 8032 there are only three. However, in the W79E(L)548 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.

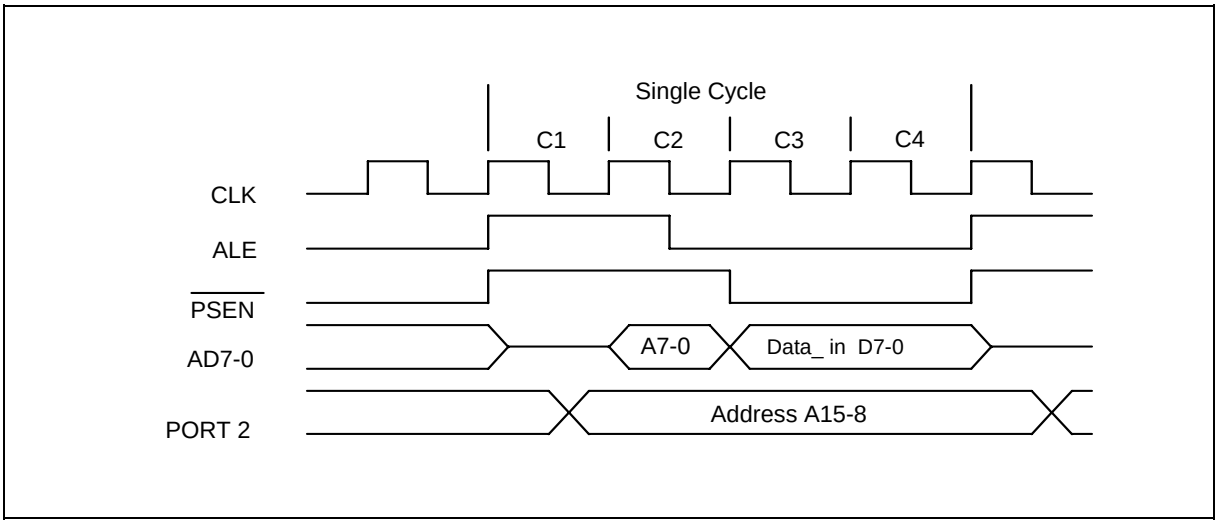


Figure 3. Single Cycle Instruction Timing

Watchdog timer in order to preempt the reset which will occur after 512 clock periods of the time-out. When the W79E(L)548 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.

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 W79E(L)548 will exit the Power Down mode with a reset or by an external interrupt pin enabled as level 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 W79E(L)548 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 and the external input has been set to a level detect mode. 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 W79E(L)548 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.

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	VECTOR ADDRESS	PRIORITY LEVEL
External Interrupt 0	IE0	0003h	1(highest)
Timer 0 Overflow	TF0	000Bh	2
External Interrupt 1	IE1	0013h	3
Timer 1 Overflow	TF1	001Bh	4
Serial Port	RI + TI	0023h	5
Timer 2 Overflow	TF2 + EXF2	002Bh	6
Watchdog Timer	WDIF	0063h	7 (lowest)

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.

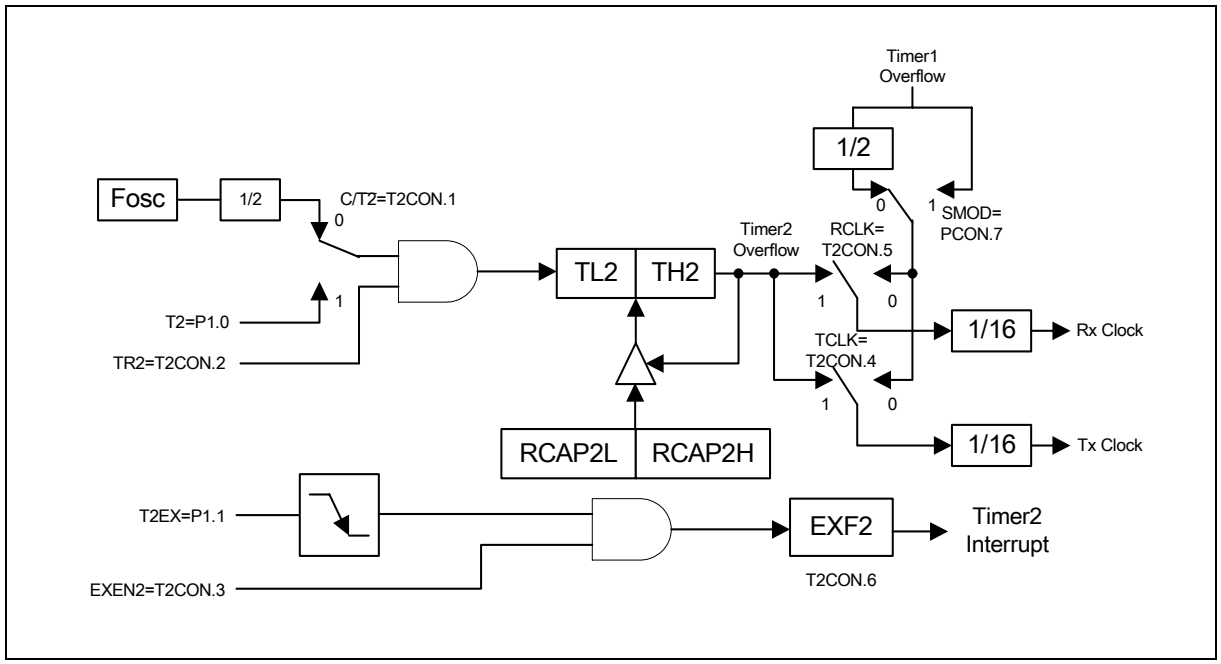


Figure 17. Baud Rate Generator Mode

PWM3 Register

Bit:	7	6	5	4	3	2	1	0

Mnemonic: PWM3 Address: DEH

PWM2 Register

Bit:	7	6	5	4	3	2	1	0

Mnemonic: PWM2 Address: DDH

PWM Control 1 Register

Bit:	7	6	5	4	3	2	1	0
	PWM3OE	PWM2OE	ENPWM3	ENPWM2	PWM1OE	PWM0OE	ENPWM1	ENWPM0

Mnemonic: PWMCON1 Address: DCH

PWM3OE: Output enable for PWM3
PWM2OE: Output enable for PWM2
ENPWM3: Enable PWM3
ENPWM2: Enable PWM2
PWM1OE: Output enable for PWM1
PWM0OE: Output enable for PWM0
ENPWM1: Enable PWM1
ENPWM0: Enable PWM0

PWM1 Register

Bit:	7	6	5	4	3	2	1	0

Mnemonic: PWM1 Address: DBH

PWM0 Register

Bit:	7	6	5	4	3	2	1	0

Mnemonic: PWM0 Address: DAH

PWMP Register

Bit:	7	6	5	4	3	2	1	0

Mnemonic: PWMP Address: D9H

```
        org     63h
        mov     TA,#AAH
        mov     TA,#55H
        clr     WDIF
        jnb     execute_reset_flag,bypass_reset     ; Test if CPU need to reset.
        jmp     $                                   ; Wait to reset
bypass_reset:
        mov     TA,#AAH
        mov     TA,#55H
        setb    RWT
        reti

        org     300h
start:
        mov     ckcon,#01h      ; select 2 ^ 17 timer
;       mov     ckcon,#61h      ; select 2 ^ 20 timer
;       mov     ckcon,#81h      ; select 2 ^ 23 timer
;       mov     ckcon,#c1h      ; select 2 ^ 26 timer
        mov     TA,#aah
        mov     TA,#55h
        mov     WDCON,#00000011B
        setb    EWDI
        setb    ea
        jmp     $               ; wait time out
```

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.

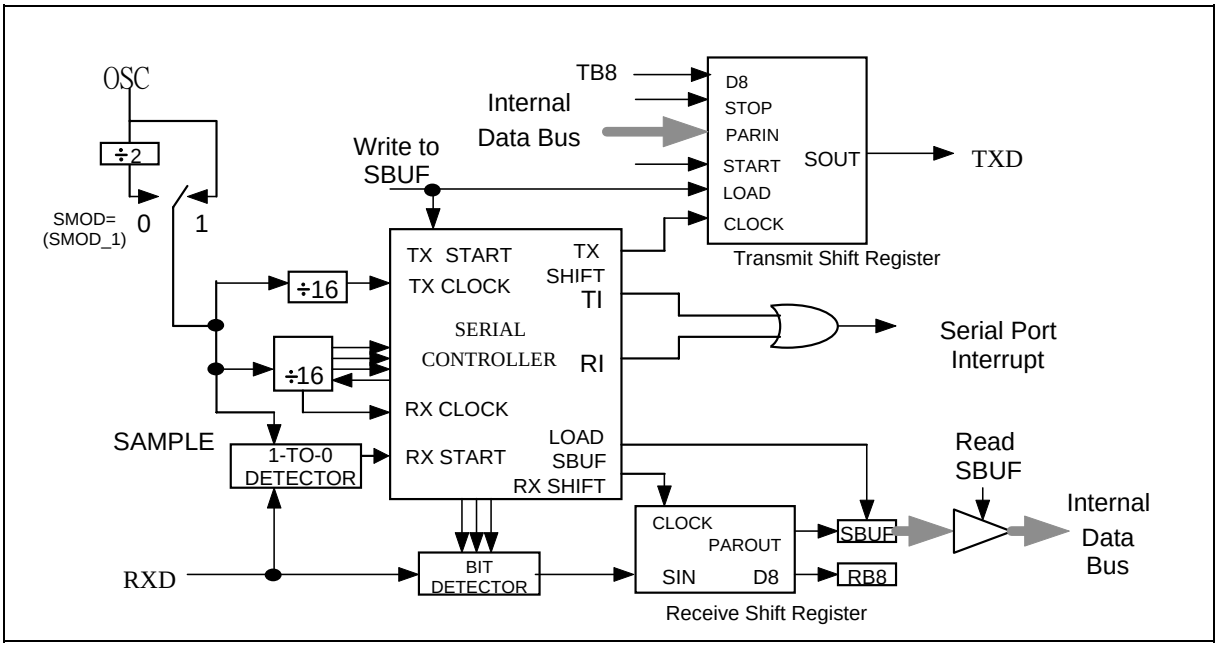


Figure 22. Serial Port Mode 2

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.

12.1 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 W79E(L)548 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. This bit is normally used as SM0 in the standard 8051 family. However, in the W79E(L)548 it serves a dual function and is called SM0/FE. There are actually two separate flags, one for SM0 and the other for FE. The flag that is actually accessed as SCON.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. 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.

12.2 Multiprocessor Communications

Multiprocessor communications makes use of the 9th data bit in modes 2 and 3. In the W79E(L)548, 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

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.7) to switch back APFlash after CPU has updated APFlash program. CPU will restart to run program from reset state.

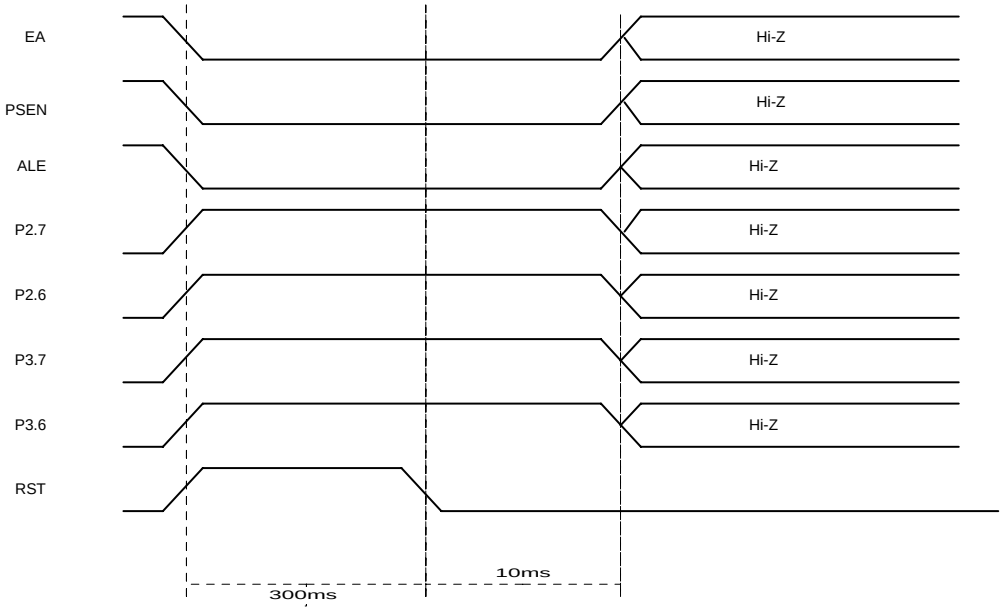
15.2 The Loader Program Locates at APFlash Memory

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.

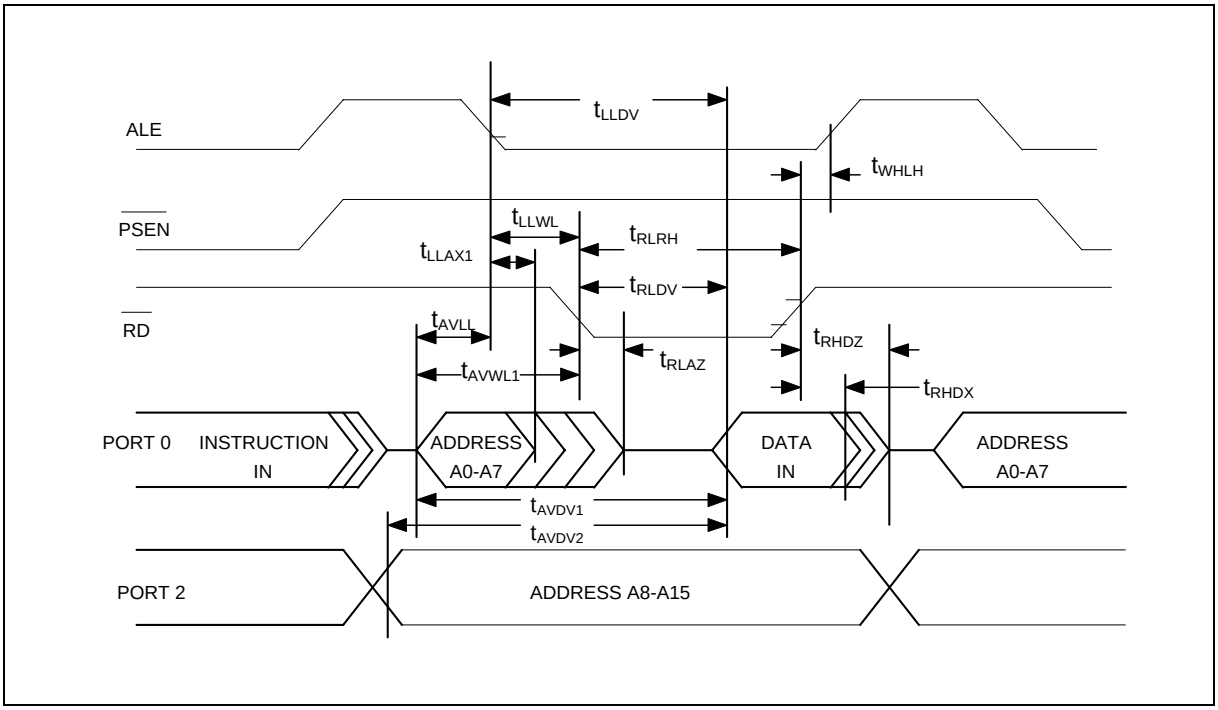
16. H/W WRITER MODE

This mode is for the writer to write / read Flash EPROM operation. A general user may not enter this mode.

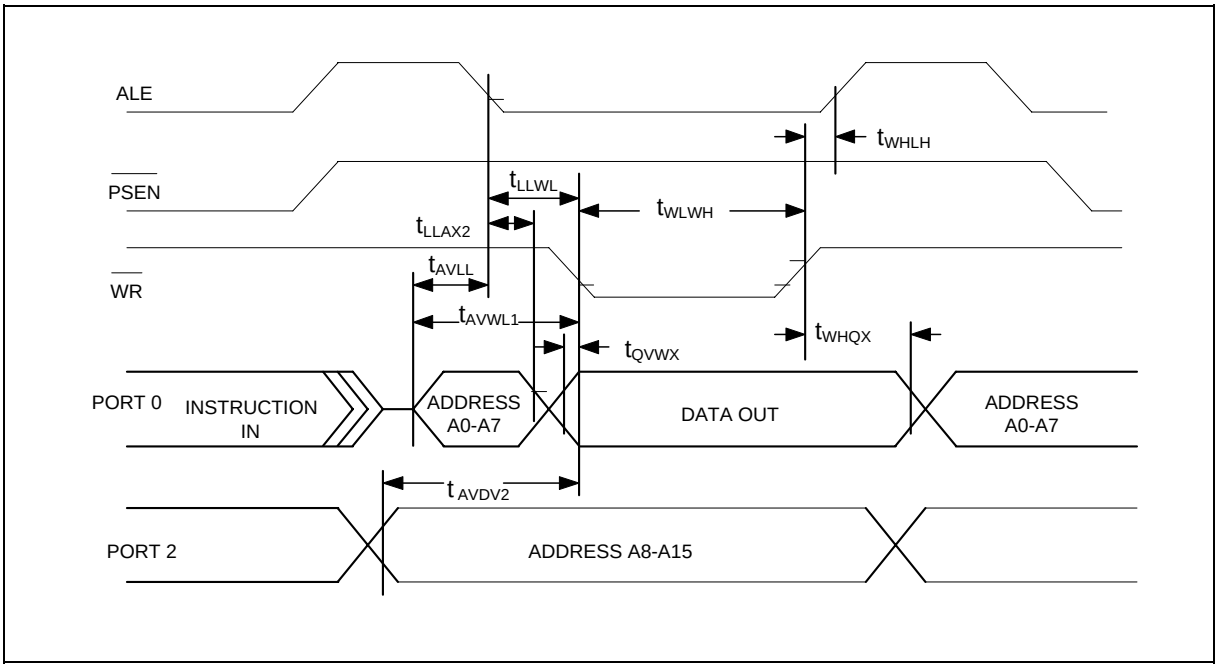
The Timing For Entering Flash EPROM Mode on the Programmer



18.3.4 Data Memory Read Cycle



18.3.5 Data Memory Write Cycle



Typical Application Circuits, continued

Expanded External Data Memory and Oscillator

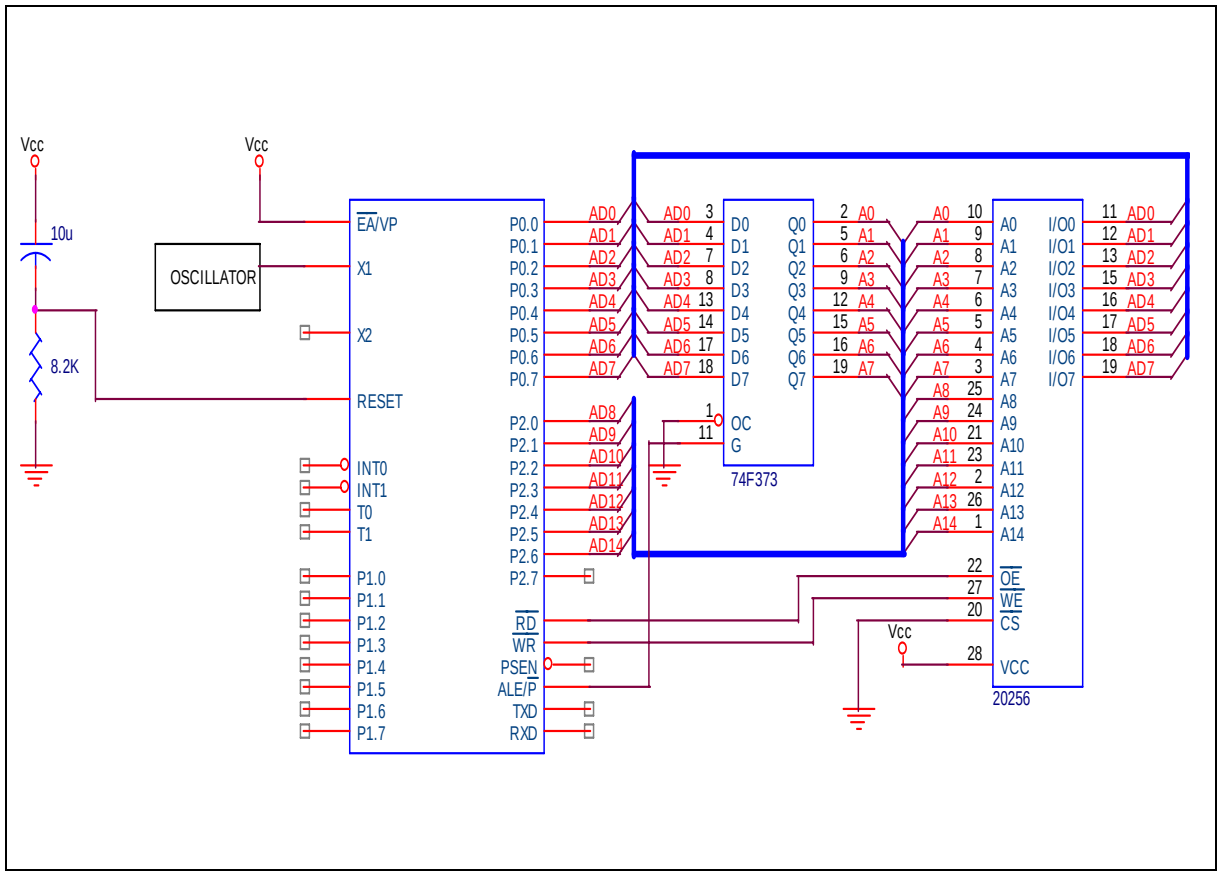


Figure B

21. APPLICATION NOTE

In-system Programming Software Examples

This application note illustrates the in-system programmability of the Winbond W79E(L)548 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
```

```
        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.
                        .
                        .
                        .
```