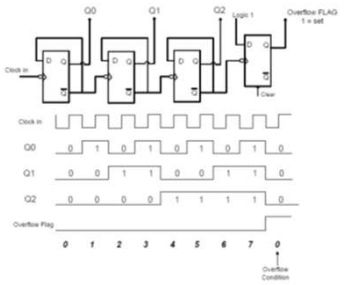
Timer

1

1

Basic Timer

- A Timer/Counter can be configured in various modes, typically based on 8-bit or 16-bit operation. It is used to counter the number of input pulse.



2

2

Introduction

Timer
•Set the initial value of registers (by software)
•Start the timer (by software) and then the timer counts up.
•Input from internal system clock (machine cycle)
•When the timer : FFFF --> 0, the time sets a bit to denote time out.

Counter
•Set the initial value of registers (by software)
•Start the timer (by software) and then the 8051 counts up.
•Input from external signal
•When the timer: FFFF --> 0, the timer sets a bit to denote time out.

3

3

Introduction

- **Time** is represented by the count in a timer.
- There are many applications that cannot be implemented without a timer:
  1. Event arrival time recording and comparison
  2. Periodic interrupt generation
  3. Pulse width and period measurement
  4. Frequency and duty cycle measurement
  5. Generation of waveforms with certain frequency and duty cycle
  6. Event counting

4

4

Introduction

- PIC18 has two to five timers
  - Depending on the family number
- These timers can be used as
  - Timers to generate a time delay
  - Counters to count events happening outside the PIC18
- Every timer needs a clock pulse to tick
- Clock source can be
  - **Internal** → 1/4th of the frequency of the crystal oscillator on OSC1 and OSC2 pins (Fosc/4) is fed into timer
  - **External:** pulses are fed through one of the PIC18's pins → Counter
- Timers are 16-bit wide (TMRxL & TMRxH)
  - Each timer has TCON (timer Control) reg.

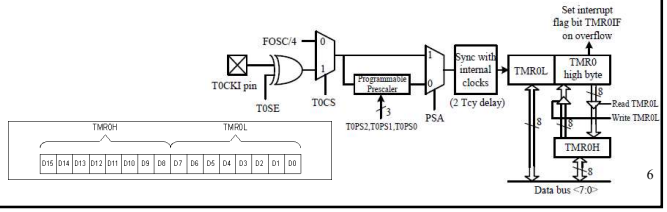
5

5

Timer0

- Can be configured as an 8-bit or 16-bit timer or counter.
- Can select the internal instruction cycle clock or the T0CKI signal as the clock signal.
- The user can choose to divide the clock signal by a prescaler before connecting it to the clock input to Timer0.
- The T0CON register controls the operation of Timer0.

7	6	5	4	3	2	1	0
TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0



6

6

Timer0

	7	6	5	4	3	2	1	0
	TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0
value after reset	1	1	1	1	1	1	1	1

TMR0ON: *Timer0 on/off control bit*
0 = stops Timer0
1 = Enables Timer0
T08BIT: *Timer0 8-bit/16-bit control bit*
0 = Timer0 is configured as a 16-bit timer
1 = Timer0 is configured as an 8-bit timer
T0CS: *Timer0 clock source select*
0 = Instruction cycle clock
1 = Transition on T0CKI pin
T0SE: *Timer0 source edge select bit*
0 = Increment on falling edge transition on T0CKI pin
1 = Increment on rising edge transition on T0CKI pin
PSA: *Timer0 prescaler assignment bit*
0 = Timer0 prescaler is assigned . Timer0 clock input comes from prescaler output.
1 = Timer0 prescaler is not assigned. Timer0 clock input bypasses prescaler.
T0PS2:T0PS0: *Timer0 prescaler select bits*
000 = 1:2 prescaler value
001 = 1:4 prescaler value
010 = 1:8 prescaler value
011 = 1:16 prescaler value
100 = 1:32 prescaler value
101 = 1:64 prescaler value
110 = 1:128 prescaler value
111 = 1:256 prescaler value

7

7

Example

Find the value for T0CON if we want to program Timer0 in 16-bit mode, no prescaler. Use PIV18 Fosc/4 crystal oscillator for the clock source, increment on positive-edge.

7	6	5	4	3	2	1	0
TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0

PSA=1,

T0CON=0000 1000

8

8

Example

Find the timer's clock frequency and its period for various PIC18-based systems, with the following frequencies. Assume that no prescaler is used.

(1)10 MHz (2) 16 MHz (3) 4 MHz

(1)¼ x 10 MHz = 2.5 MHz and T = 0.4 µs

(2)¼ x 16 MHz = 4 MHz and T = 0.25 µs

(3)¼ x 4 MHz and T = 1 µs

9

9

Characteristics and operations of 16-bit mode

1. 16-bit timer, 0000 to FFFFH.
2. After loading TMR0H and TMR0L, the timer must be started.
3. Count up, till it reaches FFFFH, then it rolls over to 0000 and activate TMR0IF bit.
4. Then TMR0H and TMR0L must be reloaded with the original value and deactivate TMR0IF bit.

10

10

Steps to program Timer0 in 16-bit mode to generate time delay

1. Load the value into the T0CON register
2. Load reg. TMR0H followed by reg. TMR0L with initial value
3. Start the timer with instruction

   BSF T0CON, TMR0ON
4. Keep monitoring the timer flag (TMR0IF) to see if it is raised.
5. Stop the timer
6. Clear the TMR0IF flag 3
7. Go Back to step 2

11

11

Example

A square wave of 50% duty cycle on the PORTB.5 is created Analyze the program.

```
        BCF     TRISB,5             ;PB5 as an output
        MOVLW   0x08                ;Timer0, 16-bit, int clk, no
prescaler
        MOVWF   T0CON               ; load T0CON reg.
HERE
        MOVLW   0xFF                ; TMR0H = FFH, the high byte
        MOVWF   TMR0H               ; load Timer0 high byte
        MOVLW   0xF2                ; TMR0L = F2H, the low byte
        MOVWF   TMR0L               ; load Timer0 low byte
        BCF     INTCON, TMR0IF ; clear timer interrupt flag bit
        BTG     PORTB,5             ; toggle PB5
        BSF     T0CON, TMR0ON       ; start Timer0
AGAIN
        BTFSS   INTCON, TMR0IF ; minitor Timer0 flag until
        BRA     AGAIN               ;  it rolls over
        BCF     T0CON, TMR0ON ; stop timer
        BRA     HERE                ; load TH, TL again
```

12

12

Example

Solution

1.T0CON is load.

2.FFF2H is loaded into TMR0H-TMR0L

3.The Timer0 interrupt flag is cleared by "BCF INTCON, TMR0IF"

4.PORTB.5 is toggled

5.Timer0 is started by "BSF T0CON, TMR0ON"

6.Timer0 counts up from FFF2H. Until it reaches FFFFH, one more clock rolls it to 0. Then Timer flag TMR0IF =1. At that point, the "BTFSS INTCON, TMR0IF" instruction bypasses the "BRA AGAIN".

7.Timer0 is stopped by the instruction "BCF T0CON, TMR0ON" and the process is repeated.

13

13

Example

In the previous example, calculate **the amount of time delay generated the timer**. Assume that the clock frequency is 10MHz.

Fosc/4 = 2.5 MHz. 1 MC =0.4µs.

FFFFH-FFF2+1=0EH=14.

14 x 0.4 = 5.6 µs.

Note: if the TMR0=XXXX, the number of MC is FFFFH+1-XXXXH

14

14

Example

Calculate the frequency of the wave generated on PIN PORTB 5.

Assume that the clock frequency is 10MHz.

```
        BCF     TRISB,5
        MOVLW   0x08
        MOVWF   T0CON
        BCF     INTCON, TMR0IF
HERE
        MOVLW   0xFF
        MOVWF   TMR0H               1
        MOVLW   -D'48'              1
        MOVWF   TMR0L               1
        CALL    DELAY               1
        BTG     PORTB,5             1
        BRA     HERE                1
DELAY
        BSF     T0CON, TMR0ON       1
AGAIN
        BTFSS   INTCON, TMR0IF      1
        BRA     AGAIN               1
        BCF     T0CON, TMR0ON       1
        BCF     INTCON, TMR0IF      1
        RETURN                      1
                                    ---
                                    13
```

T=2x(48+13)x0.4 = 48.8µs and F=20.491 MHz

15

15

Example

Calculate the frequency of the wave generated on PIN PORTB 5.

Assume that the clock frequency is 10MHz. Do not include the overhead due to instructions in the loop.

```
        BCF     TRISB,5
        MOVLW   0x08
        MOVWF   T0CON
HERE                                    FFFFH-7634H+1=89CCH=35276,
        MOVLW   0x76                    Delay 35276x0.4=14.11ms,
        MOVWF   TMR0H                   Period = 28.22 ms
        MOVLW   0x34                    Frequency = 35.434 Hz
        MOVWF   TMR0L
        BCF     INTCON, TMR0IF
        CALL    DELAY
        BTG     PORTB,5
        BRA     HERE
DELAY
        BSF     T0CON, TMR0ON
AGAIN
        BTFSS   INTCON, TMR0IF
        BRA     AGAIN
        BCF     T0CON, TMR0ON
        RETURN
```

16

16

Example

Assuming that clock = 10 MHz, write a problem to generate a square wave with period of 10ms on pin PORTB.3

T=10ms=> Delay = 5 ms. 5/0.4=12,500. 65,536-12,500=53,036=CF2CH.

TMR0H=CFH, TMR0L=2CH

```
        BCF     TRISB,3
        MOVLW   0x08
        MOVWF   T0CON
HERE
        MOVLW   0xCF
        MOVWF   TMR0H
        MOVLW   0x2C
        MOVWF   TMR0L
        BCF     INTCON, TMR0IF
        CALL    DELAY
        BTG     PORTB,3
        BRA     HERE
DELAY
        BSF     T0CON, TMR0ON
AGAIN
        BTFSS   INTCON, TMR0IF
        BRA     AGAIN
        BCF     T0CON, TMR0ON
        RETURN
```

Modify the program to get the lowest frequency.

If we set TMR0 = 0000H, the delay generated is 65536 MC,
The delay is 26.214 ms.

The smallest frequency is 1/(52.428ms) = 19.074 Hz.

17

17

Prescaler and generating larger delay

The size of delay depend on
- The Crystal frequency
- The timer's 16-bit register.

The largest timer happens when TMR0L=TMR0H=0

Prescaler option is used to duplicate the delay by dividing the clock by a factor of 2,4, 8,16, 32,64 ,128,256

If T0CON=0000 0101, then T = 4*64/f

7	6	5	4	3	2	1	0
TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0

PSA: *Timer0 prescaler assignment bit*
- 0 = Timer0 prescaler is assigned . Timer0 clock input comes from prescaler output.
- 1 = Timer0 prescaler is not assigned. Timer0 clock input bypasses prescaler.

T0PS2:T0PS0: *Timer0 prescaler select bits*
- 000 = 1:2 prescaler value
- 001 = 1:4 prescaler value
- 010 = 1:8 prescaler value
- 011 = 1:16 prescaler value
- 100 = 1:32 prescaler value
- 101 = 1:64 prescaler value
- 110 = 1:128 prescaler value
- 111 = 1:256 prescaler value

18

18

Example

Examine the following program and find the time delay in second and frequency of the wave generated. Assume that XALT = 10 MHz.

```
        BCF     TRISB,2
        MOVLW   0x05
        MOVWF   T0CON
HERE
        MOVLW   0x01
        MOVWF   TMR0H
        MOVLW   0x08
        MOVWF   TMR0L
        BCF     INTCON, TMR0IF
        CALL    DELAY
        BTG     PORTB,2
        BRA     HERE
DELAY
        BSF     T0CON, TMR0ON
AGAIN
        BTFSS   INTCON, TMR0IF
        BRA     AGAIN
        BCF     T0CON, TMR0ON
        RETURN
```

Prescaler=64.

TMR0=0108H=264, 65536-264=65272.

Now 65273x (64x0.4 µs)=1.671 s.

The frequency is 1/(2x1.671) Hz

If presclar 256 is chosen, the prescaler clock period to the timer is 0.4 µs x 256 = 1.024 ms. The largest delay is 67 seconds.

How to generate a longer delay?
Hint: call a delay routine a number of times by a loop.

19

19

Example

Write a program to generate a square wave of 50Hz on pin PORTB.7. Use Timer0, 16 bit model with prescaler = 128. Assume that XALT = 10 MHz.

```
        BCF     TRISB,7
        MOVLW   0x06
        MOVWF   T0CON
HERE
        MOVLW   0xFF
        MOVWF   TMR0H
        MOVLW   0x3D
        MOVWF   TMR0L
        BCF     INTCON, TMR0IF
        CALL    DELAY
        BTG     PORTB,7
        BRA     HERE
DELAY
        BSF     T0CON, TMR0ON
AGAIN
        BTFSS   INTCON, TMR0IF
        BRA     AGAIN
        BCF     T0CON, TMR0ON
        RETURN
```

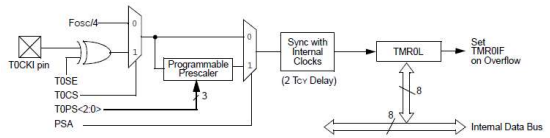
(1) F=50Hz, T=20ms
(2) Delay=10 ms
(3) 10ms/0.4 µs/128=195
(4) 65536-195=65341=FF3D
(5) TL=3D and TH =FF

20

20

Timer0 8-bit

1. Load the T0CON value
2. Load TMR0L only.
3. Start timer
4. Keep monitor the timer flag (TMR01F)



21

21

Example

Assume that XTAL=10 MHz, find (a) the frequency of the square wave generated on PORTB.0 in the following program, and (b) the smallest frequency acheivable in this program, and the TH value to do that

```
BCF     TRISB,0
        MOVLW   0x48; no prescalar
        MOVWF   T0CON
        BCF     INTCON,TMR0IF
HERE    MOVLW   0x5
        MOVWF   TMR0L
        CALL    DELAY
        BTG     PORTB,0
        BRA     HERE

DELAY   BSF     T0CON,TMR0ON
AGAIN   BTFSS   INTCON,TMR0IF
        BRA     AGAIN

        BCF     T0CON,TMR0ON
        BCF     INTCON,TMR0IF
        RETURN
```

(a) (256-5) =251x0.4 = 100.4 µs is the half period. So the period is 200.8µs, and the freq is 4.98 kHz.
(b) To get the smallest freq, we set TMR0L=00. In this case
T=256x0.4x2=204.8 µs. The freq is is 4,882.8 KHz

22

22

Assembler and negative value

(a) MOVLW –D'200'
MOVWF TMR0L
is identical to move 38H into TMR0L. Then FFH-38H + 1 = 200_{10}
Assume 10 MHz

```
        BCF     TRISB,3
        BCF     INTCON,TMR0IF
        MOVLW   0x48
        MOVWF   T0CON
HERE    BSF     PORTB,3
        CALL    DELAY
        CALL    DELAY
        BCF     PORTB,3
        CALL    DELAY
        BRA     HERE

DELAY   MOVLW   -D'150'
        MOVWF   TMR0L
        BSF     T0CON,TMR0ON
AGAIN   BTFSS   INTCON,TMR0IF
        BRA     AGAIN
        BCF     T0CON,TMR0ON
        BCF     INTCON,TMR0IF
        RETURN
```

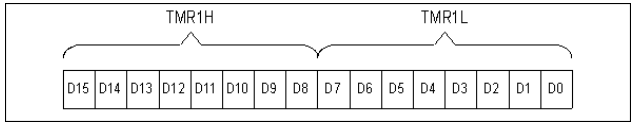
Delay == 150 MC = 150x0.4=60µs
High portion + low portion =
180 µs. Freq= 5.5555KHz.

23

23

Timer1

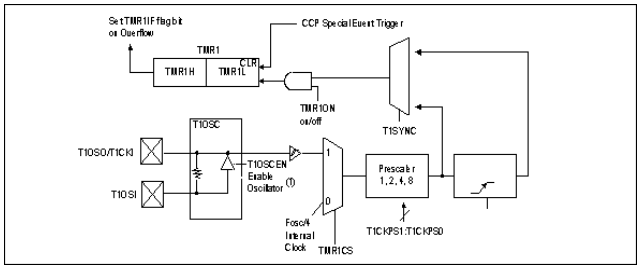
- Can be programmed in 16-bit mode only
- It has 2 bytes named as TMR1L and RMR1H
- It has also T1CON and TMR1IF
- The module incorporates its own low-power oscillator to provide an additional clocking option.
- Used as a low-power clock source for the microcontroller in power-managed operation.



24

24

Timer1 Block Diagram



25

T1CON (Timer 1 Control) Register

RD16	...	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON

RD16 D7 16-bit read/write enable bit
1 = Timer1 16-bit is accessible in one 16-bit operation.
0 = Timer1 16-bit is accessible in two 8-bit operations.

D6 Not used

T1CKPS2:T1CKPS0 D5 D4 Timer1 prescaler selector
0 0 = 1:1 Prescale value
0 1 = 1:2 Prescale value
1 0 = 1:4 Prescale value
1 1 = 1:8 Prescale value

T1OSCEN D3 Timer1 oscillator enable bit
1 = Timer1 oscillator is enabled.
0 = Timer1 oscillator is shutoff.

T1SYNC D2 Timer1 synchronization (used only when TMR1CS = 1 for counter mode to synchronize external clock input)
If TMR1CS = 0 this bit is not used.

TMR1CS D1 Timer1 clock source select bit
1 = External clock from pin RC0/T1CKI
0 = Internal clock (Fosc/4 from XTAL)

TMR1ON D0 Timer1 ON and OFF control bit
1 = Enable (start) Timer1
0 = Stop Timer1

26

PIR1 (Interrupt Control Register 1) Contains the TMR1IF Flag

							TMR1IF

TMR1IF D1 Timer1 Interrupt overflow flag bit
0 = Timer1 did not overflow
1 = Timer1 has overflowed (FFFF to 0000).

The importance of TMR1IF: When TMR1H:TMR1L overflows from FFFF to 0000, this flag is raised. We monitor this flag bit before we reload the TMR1H:TMR1L registers.

The other bits of this register are discussed in Chapter 11.

27

Example

Write a program to generate a square wave of 50Hz on PORTB.5. Use Timer 1 in 16-bit mode with the maximum prescaler. Assume that the clock frequency is 10MHz.

```
        BCF     TRISB,5
        MOVLW   0x30
        MOVWF   T1CON
HERE
        MOVLW   0xF3
        MOVWF   TMR1H
        MOVLW   0xCB
        MOVWF   TMR1L
        BCF     PIR1, TMR1IF
        CALL    DELAY
        BTG     PORTB,5
        BRA     HERE
DELAY
        BSF     T1CON, TMR1ON
AGAIN
        BTFSS   PIR1, TMR1IF
        BRA     AGAIN
        BCF     T1CON TMR1ON
        RETURN
```

(1) F=50Hz, T=20ms
(2) Delay=10 ms
(3) 10ms/0.4 µs/8=3125=C34H
(4) FFFF-C34H+1=F3CBH
(5) TL=CBH and TH =34H

28

Counter Programming

- Used to counts event outside the PIC
  - Increments the TMR0H and TMR0L registers
- T0CS in T0CON reg. determines the clock source,
  - If T0CS = 1, the timer is used as a counter. TMR0 count up (NOT count down)
  - Counts up as pulses are fed from pin RA4 (T0CKI)
  - What does T0CON=0110 1000 mean?

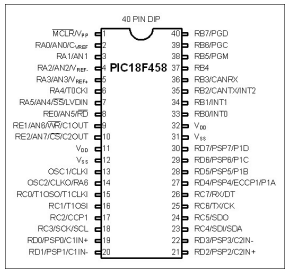
7	6	5	4	3	2	1	0
TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0

29

29

Using external Crystal for Timer clock

- Timer0 comes with two options,
  - clock fed into T0CKI
    - T0CS=0
  - Clock from a crystal
    - T0CS=1



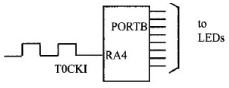
30

30

Example

Assuming that a signal is fed into pin T0CK1, write a program for counter 0 in 8-bit mode to count the pulses and display the state of the TMR0L count on PORTB.

```
      BSF TRISA,RA4          ;PORTA.4 as external input for timer0
      CLRF TRISB
      MOVLW    0x68          ; control word for timer0, 8 bit and counter model
      MOVWF    T0CON
HERE  MOVLW    0x0           ; TMR0L=0
      MOVWF    TMR0L
      BCF      INTCON,TMR0IF
      BSF       T0CON,TMR0ON
AGAIN
      MOVFF    TMR0L,PORTB
      BTFSS    INTCON,TMR0IF
      BRA      AGAIN
      BCF      T0CON,TMR0ON
      GOTO     HERE
```



31

31

Timer 1

RD16	...	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON

RD16 D7 16-bit read/write enable bit
1 = Timer1 16-bit is accessible in one 16-bit operation.
0 = Timer1 16-bit is accessible in two 8-bit operations.

D6 Not used

T1CKPS2:T1CKPS0 D5 D4 Timer1 prescaler selector
0 0 = 1:1 Prescale value
0 1 = 1:2 Prescale value
1 0 = 1:4 Prescale value
1 1 = 1:8 Prescale value

T1OSCEN D3 Timer1 oscillator enable bit
1 = Timer1 oscillator is enabled.
0 = Timer1 oscillator is shutoff

T1SYNC D2 Timer1 synchronization (used only when TMR1CS = 1 for counter mode to synchronize external clock input)
If TMR1CS = 0 this bit is not used.

TMR1CS D1 Timer1 clock source select bit
1 = External clock from pin RC0/T1CKI
0 = Internal clock (Fosc/4 from XTAL)

TMR1ON D0 Timer1 ON and OFF control bit
1 = Enable (start) Timer1
0 = Stop Timer1

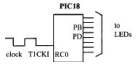
32

32

Example

Assuming that a signal is fed into pin T1CKI, write a program for counter 1 to count the pulses and display the state of the count on PORTB and C

```
      BSF TRISC,RC0          ;PORTC.0 as external input for timer0
      CLRF TRISB
      CLRF TRISD
      MOVLW     0x02         ; control word for timer1
      MOVWF     T1CON
HERE  MOVLW 0x0              ; TMR1L=0
      MOVWF     TMR1L
      MOVWF     TMR1H
      BCF       PIR1,TMR1IF
      BSF       T1CON,TMR1ON
AGAIN
      MOVFF     TMR1L,PORTB
      MOVFF     TMR1H,PORTD
      BTFSS     PIR1,TMR1IF
      BRA       AGAIN
      BCF       T1CON,TMR1ON
      GOTO      HERE
```

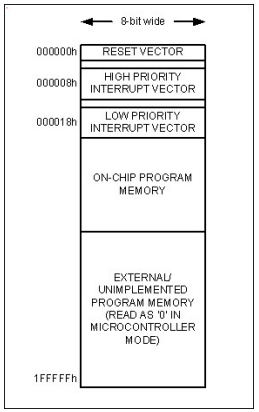


33

33

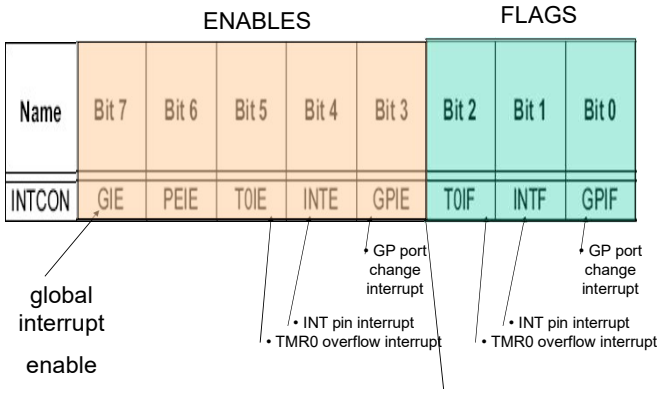
Timer Interrupt

When roll over occur,
the circuit set TMR0IF and generate a timer interrupt.



34

INTCON

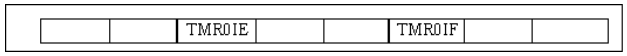


35

Timer Interrupts

Interrupt	Flag Bit	Register	Enable Bit	Register
Timer0	TMR0IF	INTCON	TMR0IE	INTCON
Timer1	TMR1IF	PIR1	TMR1IE	PIE1
Timer2	TMR2IF	PIR1	TMR3IE	PIE1
Timer3	TMR3IF	PIR3	TMR3IE	PIE2

Timer Interrupt Flag Bits and Associated Registers



INTCON Register with Timer0 Interrupt Enable and Interrupt Fla

36

Timer Interrupts

GIE/GIEH: Global Interrupt Enable bit
When IPEN = 0:
1 = Enables all unmasked interrupts
0 = Disables all interrupts
When IPEN = 1:
1 = Enables all high-priority interrupts
0 = Disables all interrupts
PEIE/GIEL: Peripheral Interrupt Enable bit
When IPEN = 0:
1 = Enables all unmasked peripheral interrupts
0 = Disables all peripheral interrupts
When IPEN = 1:
1 = Enables all low-priority peripheral interrupts
0 = Disables all low-priority peripheral interrupts
TMR0IE: TMR0 Overflow Interrupt Enable bit
1 = Enables the TMR0 overflow interrupt
0 = Disables the TMR0 overflow interrupt

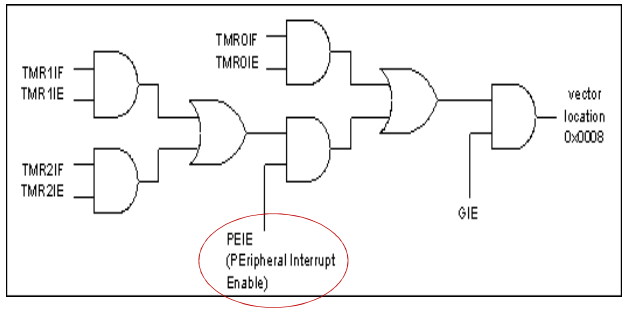
37

Timer Interrupts

INT0IE: INT0 External Interrupt Enable bit
1 = Enables the INT0 external interrupt
0 = Disables the INT0 external interrupt
RBIE: RB Port Change Interrupt Enable bit
1 = Enables the RB port change interrupt
0 = Disables the RB port change interrupt
TMR0IF: TMR0 Overflow Interrupt Flag bit
1 = TMR0 register has overflowed (must be cleared in software)
0 = TMR0 register did not overflow
INT0IF: INT0 External Interrupt Flag bit
1 = The INT0 external interrupt occurred (must be cleared in software)
0 = The INT0 external interrupt did not occur
RBIF: RB Port Change Interrupt Flag bit(1)
1 = At least one of the RB<7:4> pins changed state (must be cleared in software)
0 = None of the RB<7:4> pins have changed state

38

Timer Interrupts



39

Program

```
ORG     0000H
GOTO    MAIN

ORG     0008H
BTFSS INTCON,TMR0IF
RETFIE
GOTO  T0_ISR

ORG     00100H
MAIN    BCF  TRISB,5
        CLRF TRISD
        SETF TRISC
        MOVLW 0x08
        MOVWF T0CON
        MOVLW  0xFF
        MOVWF  TMR0H
        MOVLW  0xF2
        MOVWF TMR0L
        BCF INTCON,TMR0IF
        BSF T0CON,TMR0ON
        BSF INTCON,TMR0IE
        BSF INTCON,GIE

OVER    MOVFF PORTC,PORTD
        BRA OVER

ORG 200H
T0_ISR  MOVLW     0xFF
        MOVWF     TMR0H
        MOVLW     0xF2
        MOVWF TMR0L
        BTG       PORTB,5
        BCF INTCON,TMR0IF
        RETFIE
        END
```

Timer0 Interrupt

40

Example

Use Timer0 and Timer1 interrupts to generate square waves on pins RB1 and RB7 respectively, while data is being transferred from PORTC to PORTD.

```
        ORG     0000H
        GOTO    MAIN
        ORG     0008H
        GOTO CHK_INT
        ORG     0040H
CHK_INT BTFSC
        INTCON,TMR0IF
        CALL    T0_ISR
        BTFSC PIR1,TMR1IF
        CALL    T1_ISR
        RETFIE
        ORG     0100H
MAIN    BCF  TRISB,1
        BCF  TRISB,7
        CLRF TRISD
        SETF TRISC
        MOVLW 0x08
        MOVWF T0CON
        MOVLW   0xFF
        MOVWF   TMR0H
        MOVLW   0xF2
        MOVWF TMR0L
      BCF INTCON,TMR0IF
        MOVLW 0x0
        MOVWF T1CON
        MOVLW   0xFF
        MOVWF   TMR1H
        MOVLW   0xF2
        MOVWF TMR1L
        BCF PIR1,TMR1IF

        BSF INTCON,TMR0IE
        BSF PIE1,TMR1IE
        BSF INTCON,PEIE
        BSF INTCON,GIE
        BSF T0CON,TMR0ON
        BSF T1CON,TMR1ON

OVER    MOVFF PORTC,PORTD
        BRA OVER

T0_ISR
        ORG 200H
        MOVLW   0xFF
        MOVWF   TMR0H
        MOVLW   0xF2
        MOVWF TMR0L
        BTG     PORTB,1
        BCF INTCON,TMR0IF
        RETURN
T1_ISR
        ORG 300H
        MOVLW   0xFF
        MOVWF   TMR1H
        MOVLW   0xF2
        MOVWF TMR1L
        BTG PORTB,7
        BCF PIR1,TMR1IF
        RETURN
END
```

41

41