PIC18 Timer Programming

Hsiao-Lung Chan
Dept Electrical Engineering
Chang Gung University, Taiwan
chanhl@mail.cgu.edu.tw
Functions of PIC18 timer

- Functions of the timer
  - Generate a time delay
  - As a counter to count events happening outside the microcontroller
- Clock sources of the timer
  - Internal clock pulse
    - $1/4^{th}$ of the frequency of the crystal oscillator on the OSC1 and OSC2 pins ($F_{osc}/4$)
    - Usually used to generate a time delay
  - External clock pulse
    - Usually used as a counter
# Registers of the timer

<table>
<thead>
<tr>
<th>SFR Name</th>
<th>Function</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>PORTA</td>
<td></td>
<td>F0h</td>
</tr>
<tr>
<td>PORTB</td>
<td></td>
<td>F1h</td>
</tr>
<tr>
<td>PORTC</td>
<td></td>
<td>F2h</td>
</tr>
<tr>
<td>PORTD</td>
<td></td>
<td>F3h</td>
</tr>
<tr>
<td>PORTE</td>
<td></td>
<td>F4h</td>
</tr>
<tr>
<td>PORTF</td>
<td></td>
<td>F5h</td>
</tr>
<tr>
<td>LATB</td>
<td></td>
<td>F6h</td>
</tr>
<tr>
<td>LATC</td>
<td></td>
<td>F7h</td>
</tr>
<tr>
<td>LATD</td>
<td></td>
<td>F8h</td>
</tr>
<tr>
<td>LATE</td>
<td></td>
<td>F9h</td>
</tr>
<tr>
<td>LATF</td>
<td></td>
<td>FAh</td>
</tr>
</tbody>
</table>

## Special function registers (SFRs)

- **Timer1**
  - TCON
  - F0h
- **Timer0**
  - TMR0H
  - F7h

* - These are not physical registers.
Timer0 registers

- Timer0 can be used as an 8-bit or 16-bit timer
- Can be assessed like any SFRs
  - MOVWF   TMR0L
  - MOVFF   TMR0H, PORTC
T0CON (Timer0 control) register

- Set various timer operation mode

<table>
<thead>
<tr>
<th>TMR0ON</th>
<th>T08BIT</th>
<th>T0CS</th>
<th>TOSE</th>
<th>PSA</th>
<th>TOPS2</th>
<th>TOPS1</th>
<th>TOPS0</th>
</tr>
</thead>
<tbody>
<tr>
<td>TMR0ON</td>
<td>D7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Timer0 ON and OFF control bit</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 = Enable (start) Timer0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 = Stop Timer0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T08BIT</td>
<td>D6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Timer0 8-bit/16-bit selector bit</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 = Timer0 is configured as an 8-bit timer/counter</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 = Timer0 is configured as a 16-bit timer/counter</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T0CS</td>
<td>D5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Timer0 clock source select bit</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 = External clock from RA4/T0CKI pin</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 = Internal clock (Fosc/4 from XTAL oscillator)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOSE</td>
<td>D4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Timer0 source edge select bit</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 = Increment on H-to-L transition on T0CKI pin</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 = Increment on L-to-H transition on T0CKI pin</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PSA</td>
<td>D3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Timer0 prescaler assignment bit</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 = Timer0 clock input bypasses prescaler</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 = Timer0 clock input comes from prescaler output</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOPS2: TOPS0</td>
<td>D2 D1 D0</td>
<td>Timer0 prescaler selector</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 0 0 = 1:2</td>
<td>Prescale value (Fosc / 4 / 2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 0 1 = 1:4</td>
<td>Prescale value (Fosc / 4 / 4)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 1 0 = 1:8</td>
<td>Prescale value (Fosc / 4 / 8)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 1 1 = 1:16</td>
<td>Prescale value (Fosc / 4 / 16)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 0 0 = 1:32</td>
<td>Prescale value (Fosc / 4 / 32)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 0 1 = 1:64</td>
<td>Prescale value (Fosc / 4 / 64)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 1 0 = 1:128</td>
<td>Prescale value (Fosc / 4 / 128)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 1 1 = 1:256</td>
<td>Prescale value (Fosc / 4 / 256)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
TMR0IF flag bit in INTCON (interrupt control register)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>TMR0IF</th>
</tr>
</thead>
<tbody>
<tr>
<td>TMROIF</td>
<td>D2</td>
<td>Timer0 interrupt overflow flag bit</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0 = Timer0 did not overflow</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1 = Timer0 has overflowed (FFFF to 0000, or FF to 00 in 8-bit mode).</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**The importance of TMROIF:** In 16-bit mode, when TMROH:TMROL overflows from FFFF to 0000 this flag is raised. In 8-bit, it is raised when the timer goes from FF to 00. We monitor this flag bit before we reload the TMROH:TMROL registers.

[Diagram of oscillator and timer circuits]
Timer0 16-bit block diagram

- Should load TMR0H first, and then TMR0L
Find the frequency of the square wave on the PORTB.5 bit if XTAL = 10 MHz

BCF TRISB, 5
MOVLW 0x08 ; Timer0, 16-bit, internal clock, no prescale
MOVWF T0CON
MOVLW 0x76
MOVWF T0CON
Clock period = 1 / (1/4 x 10 MHz) = 0.4 μs
Counting clocks FFFFH – 7634H + 1 = 35,276
MOVLW 0x34
MOVWF TMR0H
MOVWF TMR0L
⇒ time delay = 35,276 x 0.4 μs = 14.11 ms
frequency = 1/(14.11 ms x 2) = 35.434 Hz
BCF INTCON, TMR0IF ; clear Timer0 interrupt flag
CALL DELAY
BTG PORTB, 5
BRA HERE

DELAY BSF T0CON, TMR0ON ; start Timer0
AGAIN BTFSS INTCON, TMR0IF ; monitor Timer0 interrupt flag
BRA AGAIN
BCF T0CON, TMR0ON ; stop Timer0
RETURN
Assume that XTAL = 10 MHz, modify previous program for a period of 10 ms

- A period of 10 ms → a time delay of 5 ms
- XTAL = 10 MHz → \( \frac{1}{4} \text{ fosc} = 2.5 \) MHz
- Counter counts up every 0.4 \( \mu s \)
- Need 5 ms / 0.4 \( \mu s = 12,500 \) clocks
- Initial value for TMR0 = 65,535 – 12,500 + 1 = CF2CH
  - TMR0H = CF
  - TMR0L = 2C
Use prescaler to generate a large time delay (assume that XTAL = 10 MHz)

```
BCF      TRISB, 2
MOVLW    0x05        ; Timer0, 16-bit, internal clock, **prescaler 64**
MOVWF    T0CON
HERE
MOVLW    0x01        FFFFH - 0108H + 1 = 65,272
MOVWF    TMR0H
MOVLW    0x08
MOVWF    TMR0L
BCF      INTCON, TMR0IF ; clear Timer0 interrupt flag
CALL     DELAY
BTG      PORTB, 2
BRA      HERE

DELAY
BSF      T0CON, TMR0ON ; start Timer0
AGAIN
BTFSS    INTCON, TMR0IF ; monitor Timer0 interrupt flag
BRA      AGAIN
BCF      T0CON, TMR0ON ; stop Timer0
RETURN
```
Largest time delay that can be achieved

- Set TMR0 to 0000 → count from 0000 to FFFF
  → 65,536 clocks
- Set prescaler 256
- Time delay = 4 μs x 256 x 65,536 = 67,108,864 μs
8-bit mode programming of Timer0
Find the frequency of the square wave if XTAL = 10 MHz

BCF  TRI SB, 3
BCF  INTCON, TMR0IF ; clear Timer0 interrupt flag
MOVLW 0x48          ; Timer0, 8-bit, internal clock, no prescale
MOVWF T0CON
HERE MOVLW -D’150’
MOVWF TMR0L
BSF PORTB, 3
CALL DELAY
MOVWF TMR0L
CALL DELAY
BCF PORTB, 3
MOVWF TMR0L
CALL DELAY
BRA HERE

DELAY BSF T0CON, TMR0ON ; start Timer0
AGAIN BTFSS INTCON, TMR0IF ; monitor Timer0 interrupt flag
BRA AGAIN
BCF T0CON, TMR0ON ; stop Timer0

Delay subroutine = 150 x 0.4 µs = 60 µs

\[ T = \text{high portion} + \text{low portion} \]
\[ = 2 \times 60 \, \mu s + 60 \, \mu s = 180 \, \mu s \]
Counter programming using Timer0

- Set T0CS (Timer0 clock source) in T0CON to 1
  - Get pulses from T0CKI (Timer0 clock input): RA4 (PORTA.4)
Clock pulses are fed into T0CKI and a buzzer is connected to PORTB.1. Sound the buzzer after counting 100 pulses

```
BCF    TRISB, 1      ; RB1 as an output to a buzzer
BSF    TRISA, 4     ; RA4 as an input for clock-in
MOVWF  T0CON
MOVLW  0x68         ; Timer0, 8-bit, external clock, no prescale
MOVWF  TMR0L
BSF    T0CON, TMR0ON ; start Timer0
AGAIN  BTFSS INTCON, TMR0IF ; monitor Timer0 interrupt flag
    BRA AGAIN
BCF    T0CON, TMR0ON ; stop Timer0
OVER   BTG PORTB, 1
CALL   DELAY
GOTO   OVER
```
Timer1

- Only support 16-bit mode
# T1CON (Timer1 control) register

<table>
<thead>
<tr>
<th>RD16</th>
<th>...</th>
<th>T1CKPS1</th>
<th>T1CKPS0</th>
<th>T1OSCEN</th>
<th>T1SYNC</th>
<th>TMR1CS</th>
<th>TMR1ON</th>
</tr>
</thead>
<tbody>
<tr>
<td>D7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**RD16**  
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  
0 1 = 1:2  
1 0 = 1:4  
1 1 = 1:8  
Prescale value

**T1OSCEN**  
D3  
Timer1 oscillator enable bit  
1 = Timer1 oscillator is enabled.  
0 = Timer1 oscillator is shut off

**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
PIR1 (interrupt control register 1)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>TMR1IF</th>
</tr>
</thead>
</table>

- **TMR1IF**  
  - **D1**  
  - Timer 1 Interrupt overflow flag bit  
  - **0** = Timer 1 did not overflow  
  - **1** = Timer 1 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.
Assume XTAL = 10 MHz, use Timer1 maximum prescaler to generate a square wave of 50 Hz

BCF TRISB, 5
MOVLW 0x30 ; Timer1, 16-bit, internal clock, prescaler 1:8
MOVWF T1CON
HERE
MOVLW 0xF3
MOVWF TMR1H
MOVLW 0xCB
MOVWF TMR1L
BCF PIR1, TMR1IF ; clear Timer1 interrupt flag
CALL DELAY
BTG PORTB, RB5
BRA HERE

DELAY BSF T1CON, TMR1ON ; start Timer1
AGAIN BTFSS PIR1, TMR1IF ; monitor Timer1 interrupt flag
BRA AGAIN
BCF PIR1, TMR1ON ; stop Timer1
RETURN
Counter programming using Timer1

- Two options for the external clock source
  - The clock fed into T1CKI pin
  - The clock from a crystal connected to T1OSI and T1OSO pins where a 32-kHz crystal is connected to
    - External clock source option (TMR1CS) is set to 1
    - Timer1 oscillator enable bit (T1OSCEN) is set to 1

- SLEEP mode
  - SLEEP instruction shut down main crystal to save power
  - Timer1 is still enable for on-chip RTC (real-time clock)
Assume that a 1-Hz pulse is fed into T1CKI, display the counter values on ports B and D.

```assembly
BSF    TRISC, RC0 ; RC0 as an input for clock-in
CLRF   TRISB
CLRF   TRISD
MOVLW  0x02      ; Timer1, 16-bit, external clock, no prescale
MOVWF  T1CON
MOVLW  0x0
MOVF   TMR1H
MOVLW  0x0
MOVWF  TMR1L
BCF    PIR1, TMR1IF
BSF    T1CON, TMR1ON ; start Timer1
AGAIN  MOVFF  TMR1H, PORTD
        MOVFF  TMR1L, PORTB
BTFSS  PIR1, TMR1IF ; monitor Timer1 flag
BRA    AGAIN
BCF    PIR1, TMR1ON ; stop Timer1
GOTO   HERE
```
Timer2 block diagram

- 8-bit timer
### T2CON (Timer2 control) register

<table>
<thead>
<tr>
<th>TOUTPS3</th>
<th>TOUTPS2</th>
<th>TOUTPS1</th>
<th>TOUTPS0</th>
<th>TMR2ON</th>
<th>T2CKPS1</th>
<th>T2CKPS0</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Value</th>
<th>Prescale value</th>
</tr>
</thead>
<tbody>
<tr>
<td>D7</td>
<td>Not used</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOUTPS3=1, TOUTPS0=0</td>
<td>Timer2 Output Postscale Select bits</td>
<td>00 0 0 = 1:1</td>
<td>Postscale value</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00 0 1 = 1:2</td>
<td>Postscale value</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00 1 0 = 1:3</td>
<td>Postscale value</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00 1 1 = 1:4</td>
<td>Postscale value</td>
</tr>
<tr>
<td></td>
<td></td>
<td>11 1 0 = 1:15</td>
<td>Postscale value</td>
</tr>
<tr>
<td></td>
<td></td>
<td>11 1 1 = 1:16</td>
<td>Postscale value</td>
</tr>
<tr>
<td>TMR2ON</td>
<td>Timer2 ON and OFF Control bit</td>
<td>D2</td>
<td>1 = Enable (Start) Timer2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 = Stop Timer2</td>
</tr>
<tr>
<td>T2CKPS1=1, T2CKPS0=0</td>
<td>Timer2 Clock Prescale Select bits</td>
<td>0 0 = Prescale is 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 1 = Prescale is 4</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 x = Prescale is 16</td>
<td></td>
</tr>
</tbody>
</table>
PIR1 (interrupt control register 1)

TMR2IF  Timer2 Interrupt overflow flag bit
0 = TMR2 value is not equal to PR2 register.
1 = TMR2 value is equal to PR2 register.
Assume that XTAL = 10 MHz, turn on PORTB4 when TMR2 reach 100

```
BCF TRISB, 4
BCF PORTB, 4
MOVLW 0x0 ; Timer2, no prescale or postscale
MOVWF T2CON
MOVLW 0x0
MOVLW 0
MOVWF TMR2
MOVLW D'100'
MOVWF PR2
BCF PI F1, TMR2IF ; clear Timer2 interrupt flag
BSF T2CON, TMR2ON ; start Timer2
AGAIN BTFSS INTCON, TMR0IF ; monitor Timer0 interrupt flag
BRA AGAIN
BSF PORTB, 4
BCF T2CON, TMR2ON ; stop Timer2
HERE BRA HERE
```
Timer3 block diagram

- 16-bit timer or counter

Set interrupt flag
TMR3IF on overflow

TMR3
TMR3H  TMR3L
CLR
CCP Special Trigger
T3CCPx
TMR3ON
on/off
T3SYNC
TT1P
Fosc/4
Internal Clock

Pre scaler
1, 2, 4, 8

T3CKPS1:T3CKPS0

TMR3CS
T3CON (Timer3 control) register

<table>
<thead>
<tr>
<th>RD16</th>
<th>T3CCP2</th>
<th>T3CKPS1</th>
<th>T3CKPS0</th>
<th>T3CCP1</th>
<th>T3SYNC</th>
<th>TMR3CS</th>
<th>TMR3ON</th>
</tr>
</thead>
</table>

**RD16**
D7 16-bit read/write enable bit
1 = Timer3 16-bit is accessible in one 16-bit operation.
0 = Timer3 16-bit is accessible in two 8-bit operations.

**T3CCP2: T3CCP1**
D6 D3 Timer3 and Timer1 to CCPx Enable bits
0 0 = Timer1 is the clock source for compare/capture of the CCP module.
0 1 = Timer3 is the clock source for compare/capture of the CCP2.
1 0 = Timer1 is the clock source for compare/capture of the CCP1.
1 x = Timer3 is the clock source for compare/capture of the CCP module.

**T3CKPS1: T3CKPS0**
D5 D4 Timer3 Input Clock 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

**T3SYNC**
D2 Timer3 external clock input synchronization control bit
Used only when TMR3CS = 1 and clock comes from an external source. If TMR3CS = 0, this bit is not used.
1 = Do not synchronize external clock input
0 = Synchronize external clock input

**TMR3CS**
D1 Timer3 clock source select bit
1 = External clock from pin T1OSI or T1CKI
0 = Internal clock (Fosc/4)

**TMR3ON**
D0 Timer3 On and Off control bit
1 = Enable (start) Timer1
0 = Stop Timer1
PIR2 (interrupt control register 2)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th>TMR3IF</th>
</tr>
</thead>
<tbody>
<tr>
<td>TMR3IF</td>
<td>Timer3 interrupt overflow flag bit</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>Timer3 did not overflow.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Timer3 has overflowed (FFFF to 0000).</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The importance of TMR3IF: In 16-bit mode, when TMR3H:TMR3L overflows from FFFF to 0000, this flag is raised.
Generate a square wave of 50 Hz on the PORTB.5 bit if XTAL = 10 MHz

BCF TRISB, 5
MOVLW 0x0 ; Timer3, 16-bit, internal clock, no prescale
MOVWF T3CON
HERE
MOVLW 0x9E
MOVWF TMR3H
MOVLW 0x58
MOVWF TMR3L
BCF PIR2, TMR3IF ; clear Timer3 interrupt flag
CALL DELAY
BTG PORTB, RB5
BRA HERE

DELAY BSF T3CON, TMR3ON ; start Timer3
AGAIN BTFSS PIR2, TMR3IF ; monitor Timer3 interrupt flag
BRA AGAIN
BCF T3CON, TMR3ON ; stop Timer3
RETURN
Reference
