05. Mô đun thời gian (Timers)

NỘI DUNG

- Nguyên lý hoạt động của Timer
- Ứng dụng của Timer
- Các module Timer trong PIC16F887
- Hệ số pre-scaler
- Cách sử dụng Timer

BỘ ĐỊNH THỜI – TIMER (1)

- Một bộ định thời là một chuỗi các flipflop với mỗi flipflop là một mạch chia 2, chuỗi này nhận một tín hiệu ngõ vào làm nguồn xung clock

- Xung clock đặt vào flipflop thứ nhất, flipflop này chia đôi tần số xung clock. Ngõ ra của flipflop thứ nhất trở thành nguồn xung clock cho flipflop thứ hai, nguồn xung clock này cũng được chia cho 2, …

- Vì mỗi một tầng kế tiếp nhau đều chia cho 2 nên một bộ định thời có n tầng sẽ chia tần số xung clock ở ngõ vào của bộ này cho 2^n

BỘ ĐỊNH THỜI – TIMER (2)

- Ngõ ra của tầng cuối cùng làm xung clock cho một flipflop báo tràn bộ định thời hay còn gọi là cờ tràn (overflow flag), cờ tràn này được kiểm tra bởi phần mềm hoặc tạo ra một ngắt

- Giá trị nhị phân trong các flipflop của bộ định thời là số đếm của các xung clock từ khi bộ định thời bắt đầu đếm

Ví dụ: một bộ định thời **16-bit** sẽ đếm từ **0000**h đến **FFFF**h. Cờ tràn được set bằng 1 khi xảy ra tràn số đếm từ **FFFF**h xuống **0000**h

HOẠT ĐỘNG CỦA TIMER (1)

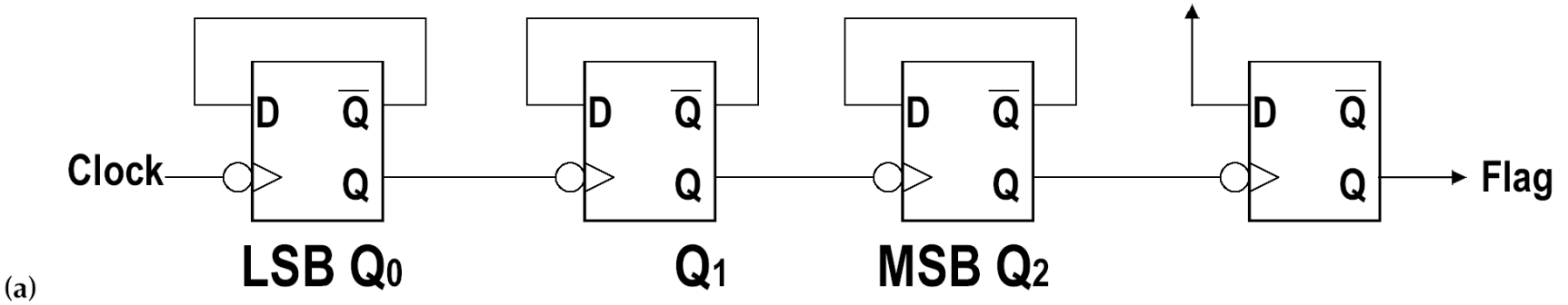
- Hoạt động của một bộ định thời **3-bit** được minh họa trong hình

- Mỗi một tầng là một flipflop kích khởi cạnh âm hoạt động như một mạch chia cho 2 do ta nối ngõ ra $\overline{Q}$ với ngõ vào **D**

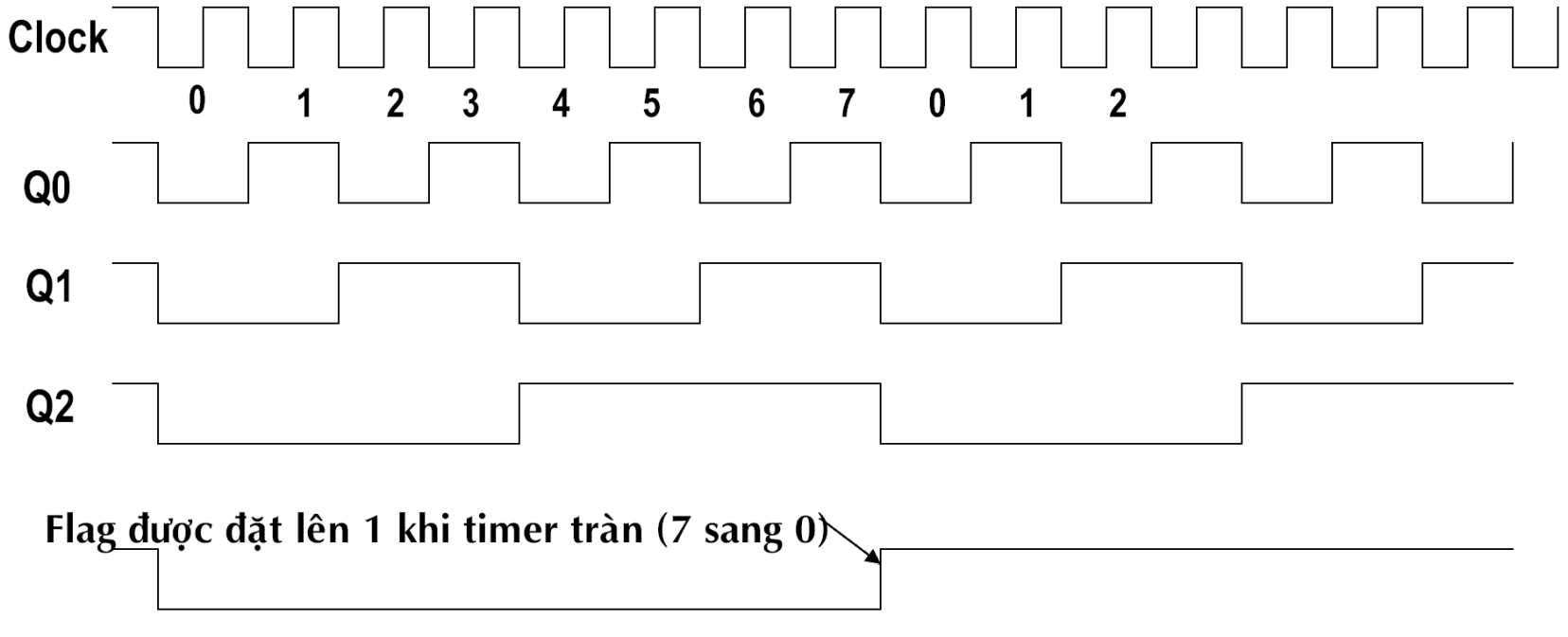
- Flipflop cờ đơn giản là một mạch chốt **D** được set bằng **1** bởi tầng cuối của bộ định thời

- Giản đồ thời gian ở hình cho thấy tầng thứ nhất ($\mathbf{Q_0}$) chia 2 tần số xung clock, tầng thứ hai chia 4 tần số xung clock và …

HOẠT ĐỘNG CỦA TIMER (2)



(a)



(b)

HOẠT ĐỘNG CỦA TIMER (3)

• Số đếm (count) được ghi ở dạng thập phân và được kiểm tra dễ dàng bằng cách khảo sát trạng thái của 3 flipflop

Ví dụ: số đếm là **4** xuất hiện khi $\mathbf{Q_2=1}$, $\mathbf{Q_1=0}$, $\mathbf{Q_0=0}$

• Các flipflop ở hình là các flipflop tác động cạnh âm (nghĩa là ngõ ra **Q** của các flipflop đổi trạng thái theo cạnh âm của xung clock)

• Khi số đếm tràn từ $\mathbf{111}_2$ xuống $\mathbf{000}_2$, ngõ ra $\mathbf{Q_2}$ có cạnh âm làm cho trạng thái của flipflop cờ đổi từ 0 lên 1 (ngõ vào **D** của flipflop này luôn ở logic 1)

HOẠT ĐỘNG CỦA TIMER (4)

- Timer **8-bits** có thể đếm từ 0 đến 2^8 = **255**
  Timer **16-bits** có thể đếm từ 0 đến 2^{16} = **65.535**
  Timer **32-bits** có thể đếm từ 0 đến 2^{32} = **4.294.967.296**

- Để có thể hoat động ta cần cấp xung clock cho timer

Ví dụ: Một xung tần số **10 kHz** là đầu vào của 1 timer thì thời gian cho mỗi lần đếm 1 đơn vị sẽ là **100 micro** giây

ỨNG DỤNG CỦA TIMER (1)

- **Định thời trong một khoảng thời gian**
- **Đếm sự kiện**
- **Tạo tốc độ baud cho port nối tiếp (8051)**

ỨNG DỤNG CỦA TIMER (2)

- Trong các ứng dụng định thời trong một khoảng thời gian, bộ định thời được lập trình sao cho sẽ tràn sau một khoảng thời gian qui định và set cờ tràn của bộ định thời bằng 1

- Cờ tràn được sử dụng để đồng bộ chương trình nhằm thực hiện một công việc như là kiểm tra trạng thái của các ngõ nhập hoặc gởi dữ liệu đến các ngõ xuất

- Các ứng dụng khác có thể sử dụng xung clock qui định của bộ định thời để đo khoảng thời gian giữa 2 sự kiện (ví dụ đo độ rộng xung)

ỨNG DỤNG CỦA TIMER (3)

- Việc đếm sự kiện được dùng để xác định số lần xuất hiện của một sự kiện hơn là đo thời gian giữa các sự kiện

- Từ "sự kiện" là một kích thích bên ngoài cung cấp một chuyển trạng thái từ 1 xuống 0 tới một chân của chip

TIMER 0 (1)

Timer 0 là bộ định thời 8-bit với các tính chất sau

- Thanh ghi TMR0 điều khiển hoạt động của timer/counter 8-bit
- Hệ số pre-scaler 8-bit
- Có thể sử dụng xung clock nội và xung clock ngoại
- Lựa chọn cạnh xung clock ngoại
- Báo ngắt cờ tràn

TIMER 0 (2)

8-bit Timer Mode

- Khi sử dụng như là timer, Timer0 module sẽ tăng mỗi chu kỳ lệnh
- Timer mode được dùng khi gán bit **T0CS** của thanh ghi **OPTION** to **0**

8-bit Counter Mode

- Khi sử dụng như là counter, Timer0 module sẽ tăng ở mỗi cạnh lên hoặc xuống của chân **T0CKI**
- Counter mode được dùng khi gán bit **T0CS** của thanh ghi **OPTION** to **1**

TIMER 0 (3)

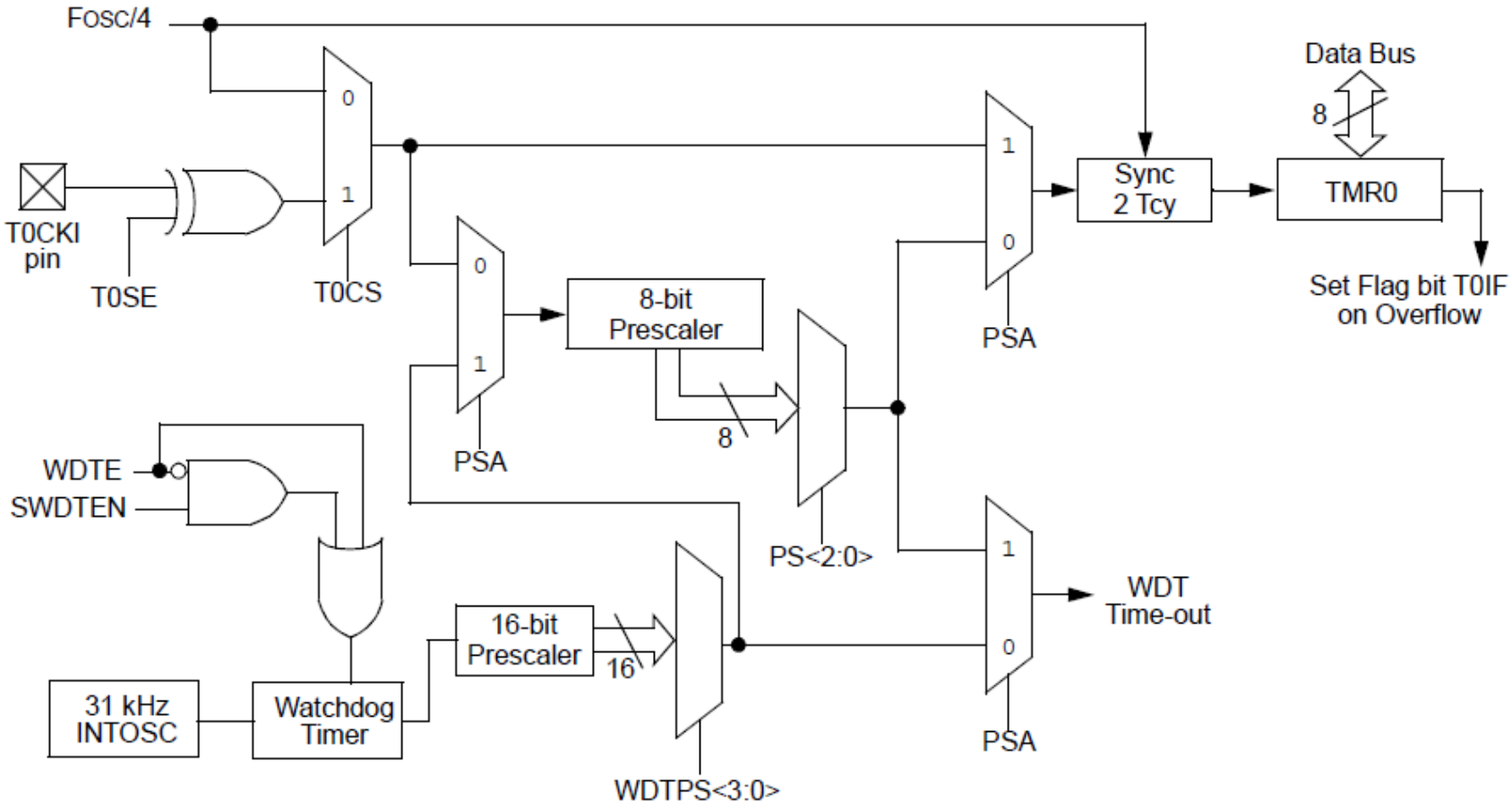
Software programmable pre-scaler

- Để sử dụng chức năng này cho Timer0, ta phải xóa bit **PSA** của thanh ghi **OPTION** xuống **0**

- Có 8 tỉ lệ có thể sử dụng từ 1:2 đến 1:256 bằng cách chọn các bit **PS**<2:0> của thanh ghi **OPTION**

TIMER 0 (4)

Sơ đồ khối nhân tỉ lệ pre-scale Timer0/WDT



Note 1: T0SE, T0CS, PSA, PS<2:0> are bits in the OPTION register.
2: SWDTEN and WDTPS<3:0> are bits in the WDTCON register.
3: WDTE bit is in the Configuration Word Register1.

TIMER 0 (5)

- Timer 0 sẽ tạo ra một ngắt khi thanh ghi **TMR0** tràn từ giá trị **FF**h sang giá trị **00**h

- Bit cờ ngắt **T0IF** của thanh ghi **INTCON** sẽ được set lên **1** cứ mỗi lần thanh ghi **TMR0** tràn

- Bit **T0IF** phải được xóa bằng phần mềm

- Bit **T0IE** của thanh ghi **INTCON** điều khiển việc sử dụng ngắt của Timer 0

TIMER 0 (6)

Thanh ghi OPTION

REGISTER 5-1: OPTION_REG: OPTION REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **$\overline{\text{RBPU}}$:** PORTB Pull-up Enable bit
1 = PORTB pull-ups are disabled
0 = PORTB pull-ups are enabled by individual PORT latch values

bit 6 **INTEDG:** Interrupt Edge Select bit
1 = Interrupt on rising edge of INT pin
0 = Interrupt on falling edge of INT pin

TIMER 0 (7)

Thanh ghi OPTION

bit 5 **T0CS:** TMR0 Clock Source Select bit

1 = Transition on T0CKI pin
0 = Internal instruction cycle clock (FOSC/4)

bit 4 **T0SE:** TMR0 Source Edge Select bit

1 = Increment on high-to-low transition on T0CKI pin
0 = Increment on low-to-high transition on T0CKI pin

bit 3 **PSA:** Prescaler Assignment bit

1 = Prescaler is assigned to the WDT
0 = Prescaler is assigned to the Timer0 module

bit 2-0 **PS<2:0>:** Prescaler Rate Select bits

BIT VALUE	TMR0 RATE	WDT RATE
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

TIMER 0 (8)

TABLE 5-1: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER0

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
TMR0	Timer0 Module Register								xxxx xxxx	uuuu uuuu
INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000x
OPTION_REG	$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	1111 1111	1111 1111

Legend: – = Unimplemented locations, read as '0', u = unchanged, x = unknown. Shaded cells are not used by the Timer0 module.

TIMER 1 (1)

Timer 1 là một module 16-bits có các tính năng sau:

- Sử dụng cặp thanh ghi timer/counter **16-bits** (**TMR1H**:**TMR1L**)
- Sử dụng xung clock ngoài hay xung clock nội
- Có **3-bits** điều khiển pre-scaler
- Có thể sử dụng dao động nội **LP**
- Hoạt động đồng bộ hoặc không đồng bộ
- Ngắt khi xảy ra tràn
- Hỗ trợ các chức năng Capture/Compare

TIMER 1 (2)

- Module Timer 1 là một bộ đếm **16-bits** thông qua cặp thanh ghi **TMR1H:TMR1L**

- Khi sử dụng xung clock nội, module là một Timer

- Khi sử dụng với nguồn xung clock ngoại, module có thể là Timer hay Counter

TIMER 1 (3)

- Bit **TMR1CS** của thanh ghi **T1CON** được dùng để lựa chọn nguồn xung clock
- Khi **TMR1CS = 0**, nguồn xung clock là **Fosc/4**
- Khi **TMR1CS = 1**, sử dụng xung clock ngoại

Clock Source	TMR1CS
Fosc/4	0
T1CKI pin	1

TIMER 1 (4)

Timer 1 có 4 hệ số Pre-scaler là: 1,2 ,4 và 8

Timer 1 có thể sử dụng tần số thấp built-in trong chip là **32.768kHz**

Cờ ngắt của Timer 1 set lên **1** khi xảy ra tràn số đếm từ **FFFF**h xuống **0000**h. Để sử dụng ngắt này chúng ta phải set:

- Bit cho phép ngắt của Timer 1 trên thanh ghi **PIE1**
- Bit **PEIE** của thanh ghi **INTCON**
- Bit **GIE** của thanh ghi **INTCON**

TIMER 1 (5)

Thanh ghi điều khiển Timer 1

REGISTER 6-1: T1CON: TIMER1 CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
T1GINV(1)	TMR1GE(2)	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **T1GINV:** Timer1 Gate Invert bit(1)

1 = Timer1 gate is active-high (Timer1 counts when gate is high)
0 = Timer1 gate is active-low (Timer1 counts when gate is low)

bit 6 **TMR1GE:** Timer1 Gate Enable bit(2)

If TMR1ON = 0:
This bit is ignored
If TMR1ON = 1:
1 = Timer1 is on if Timer1 gate is not active
0 = Timer1 is on

TIMER 1 (6)

Thanh ghi điều khiển Timer 1

bit 5-4 **T1CKPS<1:0>:** Timer1 Input Clock Prescale Select bits
11 = 1:8 Prescale Value
10 = 1:4 Prescale Value
01 = 1:2 Prescale Value
00 = 1:1 Prescale Value

bit 3 **T1OSCEN:** LP Oscillator Enable Control bit
1 = LP oscillator is enabled for Timer1 clock
0 = LP oscillator is off

bit 2 **T1SYNC:** Timer1 External Clock Input Synchronization Control bit
TMR1CS = 1:
1 = Do not synchronize external clock input
0 = Synchronize external clock input
TMR1CS = 0:
This bit is ignored. Timer1 uses the internal clock

bit 1 **TMR1CS:** Timer1 Clock Source Select bit
1 = External clock from T1CKI pin (on the rising edge)
0 = Internal clock (FOSC/4)

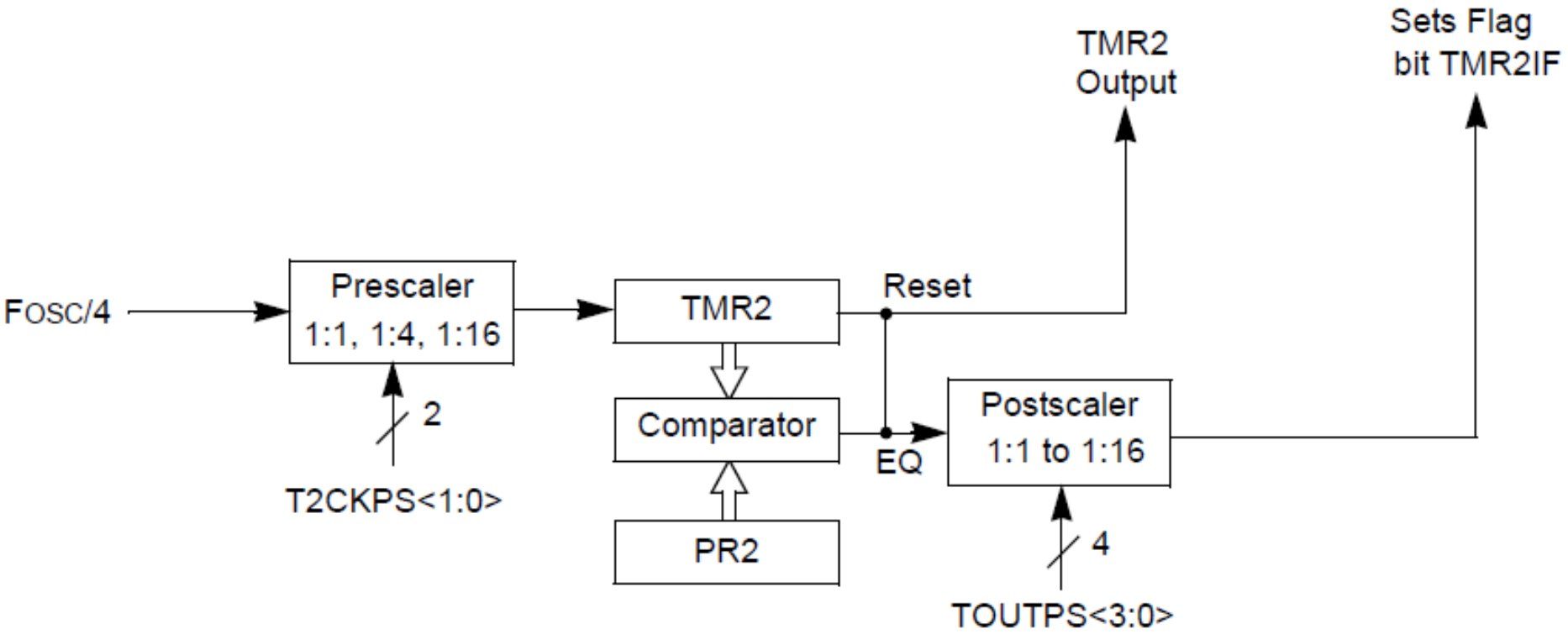
bit 0 **TMR1ON:** Timer1 On bit
1 = Enables Timer1
0 = Stops Timer1

TIMER 2 (1)

Timer 2 là 1 module 8-bits có các tính năng sau:

- Thanh ghi định thời **8-bit** (**TMR2**)
- Thanh ghi chu kỳ **8-bit** (**PR2**)
- Ngắt xảy ra khi **TMR2** tương ứng với **PR2**
- Các hệ số pre-scaler: **1:1**, **1:4**, và **1:16**
- Các hệ số post-scaler: **1:1** đến **1:16**

TIMER 2 (2)


FOSC/4
Prescaler
1:1, 1:4, 1:16
2
T2CKPS<1:0>
TMR2
Comparator
PR2
Reset
EQ
TMR2
Output
Postscaler
1:1 to 1:16
4
TOUTPS<3:0>
Sets Flag
bit TMR2IF

TIMER 2 (3)

- Nguồn xung clock vào Timer 2 là nguồn xung clock của hệ thống **Fosc/4**

- Xung clock đưa vào khối prescaler của Timer 2 (**1**,**4**,**16**)

- Ngõ ra của khối prescaler dùng để tăng số đếm trong thanh ghi **TMR2**

- Số đếm trong thanh ghi **TMR2** sẽ đếm từ **00**h cho đến khi nó bằng với số cài đặt trong thanh ghi **PR2**, lúc này có 2 trường hợp xảy ra
  + Thanh ghi **TMR2** được reset về **0** cho chu kỳ đếm mới
  + Khối postscaler của Timer 2 tăng **1** đơn vị

- Ngõ ra của khối post-scaler dùng để set cờ tràn **TMR2IF** trên thanh ghi **PIR1**

TIMER 2 (4)

- Khi được reset, thanh ghi **TMR2** được cài đặt về **00**h, còn thanh ghi **PR2** cài đặt về **FF**h

- Timer 2 được bật lên nhờ set bit **TMR2ON** trong thanh ghi **T2CON** lên **1**, tắt đi bằng cách xóa bit **TMR2ON** xuống **0**

- Hệ số pre-scaler của Timer 2 được điều khiển bởi bit **T2CKPS**, hệ số post-scaler được điều khiển bởi bit **TOUTPS** trong thanh ghi **T2CON**

TIMER 2 (5)

REGISTER 7-1: T2CON: TIMER2 CONTROL REGISTER

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **Unimplemented:** Read as '0'

bit 6-3 **TOUTPS<3:0>:** Timer2 Output Postscaler Select bits

0000 = 1:1 Postscaler
0001 = 1:2 Postscaler
0010 = 1:3 Postscaler
0011 = 1:4 Postscaler
0100 = 1:5 Postscaler
0101 = 1:6 Postscaler
0110 = 1:7 Postscaler
0111 = 1:8 Postscaler

TIMER 2 (6)

1000 = 1:9 Postscaler
1001 = 1:10 Postscaler
1010 = 1:11 Postscaler
1011 = 1:12 Postscaler
1100 = 1:13 Postscaler
1101 = 1:14 Postscaler
1110 = 1:15 Postscaler
1111 = 1:16 Postscaler

bit 2 **TMR2ON:** Timer2 On bit

1 = Timer2 is on
0 = Timer2 is off

bit 1-0 **T2CKPS<1:0>:** Timer2 Clock Prescale Select bits

00 = Prescaler is 1
01 = Prescaler is 4
1x = Prescaler is 16

CÁC BƯỚC LẬP TRÌNH TIMER

1. Chọn chế độ làm việc

2. Nạp trị số cho thanh ghi đếm

3. Cho Timer chạy

4. Kiểm tra theo dõi cờ báo tràn thanh ghi

5. Dừng Timer

6. Xóa cờ tràn

7. Quay về bước 2

VÍ DỤ 1: ĐIỀU KHIỂN LED - TIMER 0

Viết chương trình điều khiển 1 đèn LED sáng tắt 1 giây sử dụng Timer 0?

```
#include <16F887.h>
#fuses NOWDT,PUT,XT,NOPROTECT
#use delay(clock=4000000)
#bit D0=0x08.0
int16 count;
int1 b;
#int_timer0
void interrupt_timer0()
{
   set_timer0(6);
   ++count;
   if(count == 2000)
   {
      count=0;
      b=~b;
   }
}
void main(void)
{
   set_tris_d(0);
   b=0;
   count = 0;
   enable_interrupts(global);
   enable_interrupts(int_timer0);
   setup_timer_0 (RTCC_INTERNAL |
RTCC_DIV_2);
   set_timer0(6);
   while(true)
   {
      D0=b;
   }
}
```

VÍ DỤ 2: ĐIỀU KHIỂN LED - TIMER 1

Viết chương trình điều khiển 1 đèn LED sáng tắt 1 giây sử dụng Timer 1?

```
#include <16F887.h>
#fuses NOWDT,PUT,XT,NOPROTECT
#use delay(clock=4000000)
#bit D0=0x08.0
int16 count;
int1 b;
#int_timer1
void interrupt_timer1()
{
   set_timer1(64536);
   ++count;
   if(count == 500)
   {
      count=0;
      b=~b;
   }
}
void main(void)
{
   set_tris_d(0);
   b=0;
   count = 0;
   enable_interrupts(global);
   enable_interrupts(int_timer1);
   setup_timer_1 ( T1_INTERNAL |
T1_DIV_BY_2);
   set_timer1(64536);
   while(true)
   {
      D0=b;
   }
}
```

VÍ DỤ 3: SỬ DỤNG 2 TIMER

Viết chương trình sử dụng 2 Timers điều khiển 2 LED sao cho: LED 1 sáng tắt mỗi 1 giây, LED 2 sáng tắt mỗi 2 giây? Nhận xét kết quả?

VÍ DỤ 4: TẠO XUNG VUÔNG

Viết chương trình tạo ra xung vuông có tần số 0.5 Hz, chu kỳ nhiệm vụ là 75% sử dụng Timer?

VÍ DỤ 5: MẠCH ĐUỔI LED

Viết chương trình điều khiển 8 LED sao cho: LED 1 sáng 1 giây, rồi đến LED 2 sáng 1 giây, … sử dụng Timer?

VÍ DỤ 6: SỬ DỤNG COUNTER

Viết chương trình sử dụng chức năng Counter của Timer bằng cách đếm số lần nhấn nút và xuất kết quả ra LED 7 đoạn?