

CHƯƠNG 7
MODUL CCP
(CAPTURE/COMPARE/PWM)

CHƯƠNG 6: MODUL CCP

Nội dung bao gồm:

- **Tổng quan về modul CCP**
 - Capture: Bắt sự kiện
 - Compare: So sánh
 - PWM: Điều chế độ rộng xung
- **Các thanh ghi của modul CCP**
- **Giới thiệu sơ lược về chế độ Capture**
- **Giới thiệu sơ lược về chế độ Compare**
- **Giới thiệu sơ lược về chế độ PWM**
- **Các tính năng nâng cao của modul CCP1**
- **Ví dụ minh họa và bài tập ứng dụng.**

TỔNG QUAN VỀ MODUL CCP

- PIC16F887 có 2 modul CCP (CCP1 và CCP2). Cách thức cấu hình và hoạt động hoàn toàn tương tự nhau trong chế độ bình thường, ngoại trừ các tính năng nâng cao PWM chỉ có trên CCP1
- Các Timer sử dụng cho modul CCP.

Chế độ CCP	Timer sử dụng
Capture	Timer 1
Compare	Timer 1
PWM	Timer 2

TỔNG QUAN VỀ MODUL CCP

➤ Chế độ Capture:

- Xác định khoảng thời gian tồn tại của một sự kiện bên ngoài được phản ánh thông qua một ngõ vào của vi điều khiển

➤ Chế độ Compare:

- Thay đổi trạng thái của một chân ngõ ra hoặc tạo ra một ngắt sau một khoảng thời gian xác định

➤ Chế độ PWM:

- Tạo ra sóng vuông có chu kỳ nhiệm vụ thay đổi được tại một tần số xác định
- Cung cấp những tính năng nâng cao cho nhiều kết nối yêu cầu khác nhau.

THANH GHI CỦA MODUL CCP

CCP1 Control Register (CCP1CON)

P1M1 | P1M0 | CCP1X | CCP1Y | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0

BIT	Chức năng
CCP1M<3:0>	Các bit chọn chế độ CCP, cấu hình modul này như là ngõ vào Capture, ngõ ra Compare hoặc ngõ ra PWM

THANH GHI CỦA MODUL CCP

CCPx M3	CCPx M2	CCPx M1	CCPx M0	Chế độ cho modul CCPx
0	0	0	0	Tắt Capture/Compare/PWM (reset modul CCP)
0	0	0	1	Không dùng
0	0	1	0	Chế độ Compare, đảo ngõ ra khi bằng nhau
0	0	1	1	Không dùng
0	1	0	0	Chế độ Capture, tại mỗi cạnh xuống
0	1	0	1	Chế độ Capture, tại mỗi cạnh lên
0	1	1	0	Chế độ Capture, tại mỗi cạnh lên thứ 4
0	1	1	1	Chế độ Capture, tại mỗi cạnh lên thứ 16
1	0	0	0	Chế độ Compare, đặt ngõ ra khi bằng nhau
1	0	0	1	Chế độ Compare, xóa ngõ ra khi bằng nhau
1	0	1	0	Chế độ Compare, tạo ngắt phần mềm khi bằng nhau
1	0	1	1	Chế độ Compare, kích sự kiện đặc biệt
1	1	0	0	Chế độ PWM; P1A, P1C active-high; P1B, P1D active-high
1	1	0	1	Chế độ PWM; P1A, P1C active-high; P1B, P1D active-low
1	1	1	0	Chế độ PWM; P1A, P1C active-low; P1B, P1D active-high
1	1	1	1	Chế độ PWM; P1A, P1C active-low; P1B, P1D active-low

THANH GHI CỦA MODUL CCP

CCP1 Control Register (CCP1CON)

P1M1 | P1M0 | CCP1X | CCP1Y | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0

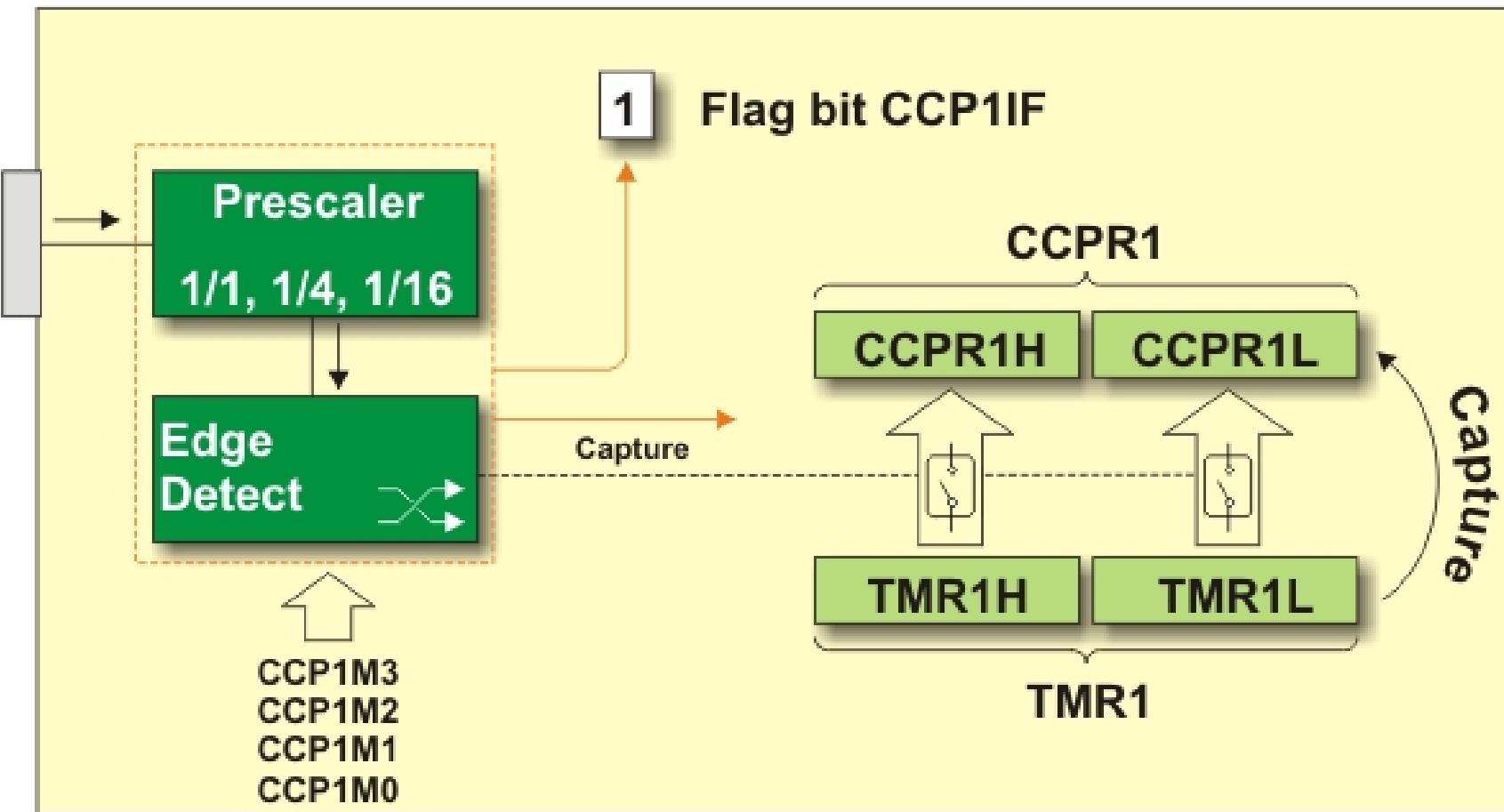
BIT	Chức năng
CCP1M<3:0>	Các bit chọn chế độ CCP, cấu hình modul này như là ngõ vào Capture, ngõ ra Compare hoặc ngõ ra PWM
CCP1<X:Y>	2 bit LSB qui định chu kỳ nhiệm vụ cho xung PWM (8 bit MSB được chứa trong thanh ghi CCPR1L)
P1M<1:0>	Các bit cấu hình ngõ ra PWM này chỉ có thể dùng cho modul ECCP (Enhanced CCP). Điều khiển lái ngõ ra toàn cầu hoặc nữa cầu.

CHẾ ĐỘ CAPTURE

- Trong chế độ này, giá trị thanh ghi TMR1 (TMR1H:TMR1L) được sao chép vào thanh ghi CCPx (CCPRxH:CCPRxL) và cờ báo CCPxFIF được đặt khi xảy ra:
 - Mỗi cạnh xuống tại chân CCPx
 - Mỗi cạnh lên tại chân CCPx
 - Mỗi cạnh lên thứ 4 tại chân CCPx
 - Mỗi cạnh lên thứ 16 tại chân CCPx
- Điều kiện cần có:
 - Chân CCPx phải được cấu hình là ngõ vào
 - Modul Timer 1 phải được cấu hình hoạt động ở chế độ định thời gian hoặc bộ đếm đồng bộ.

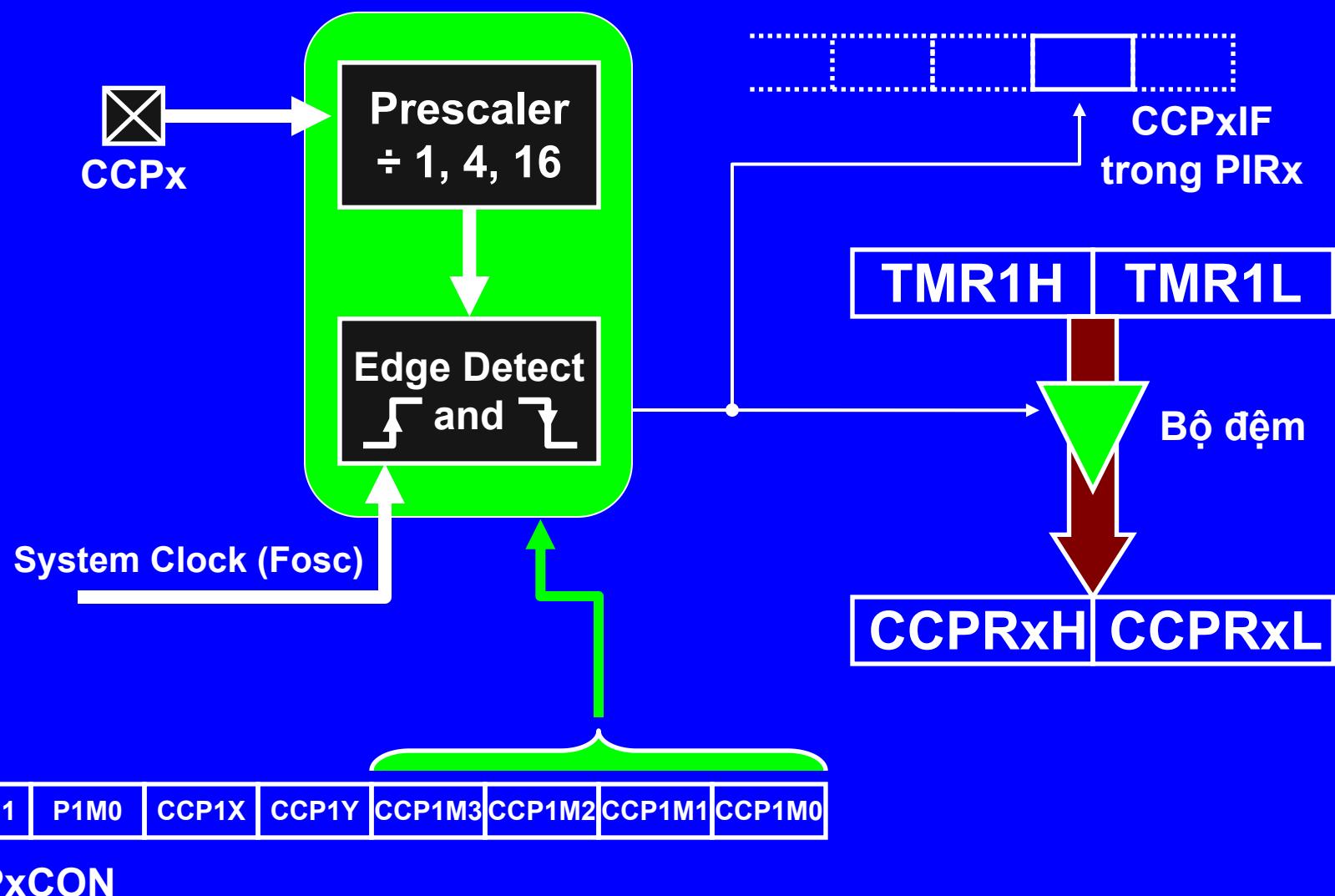
SƠ ĐỒ KHỐI CHẾ ĐỘ CAPTURE

Pin
RC2/CCP1

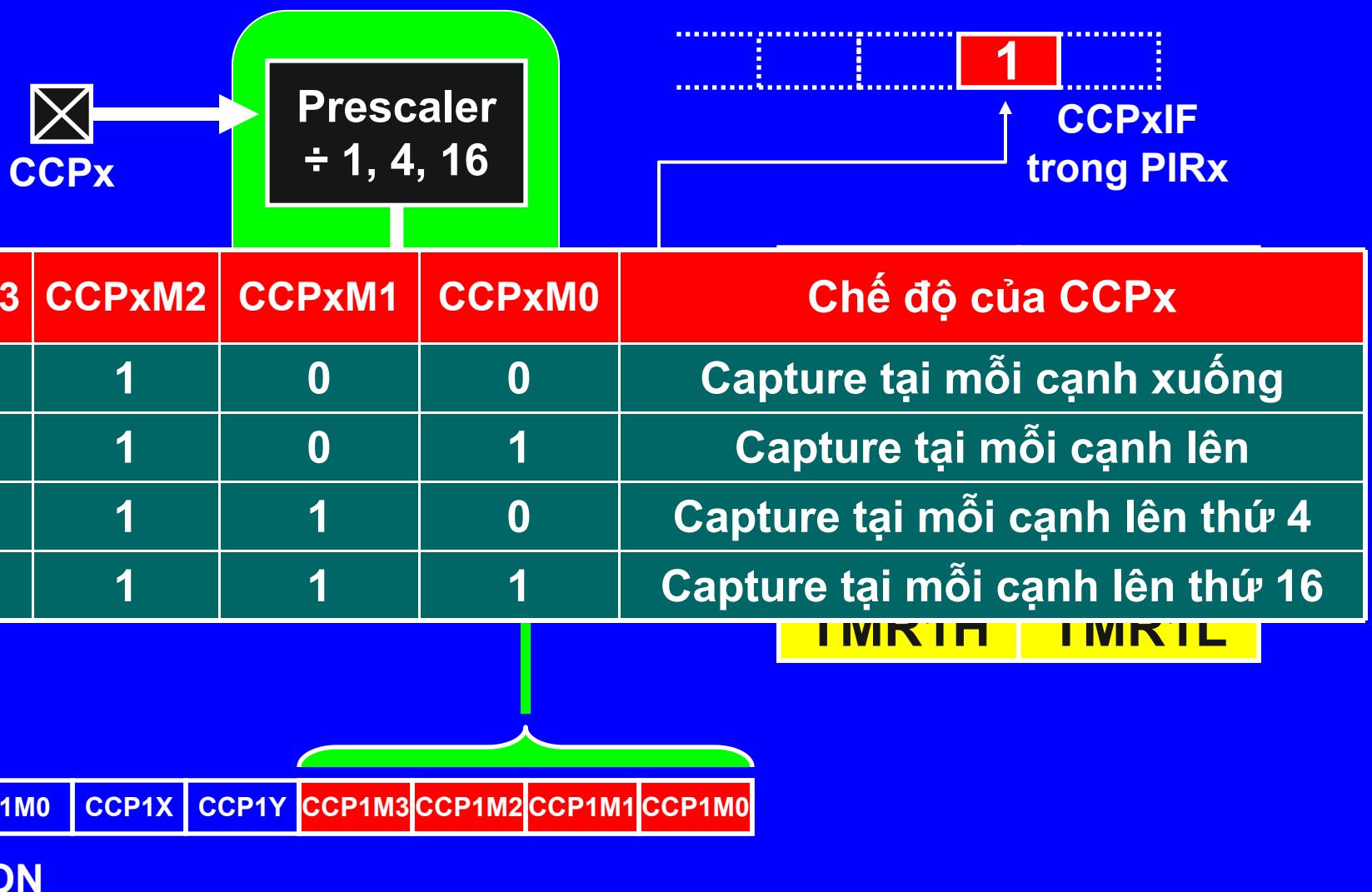


CCP1 trong chế độ CAPTURE

SƠ ĐỒ KHỐI CHẾ ĐỘ CAPTURE



SƠ ĐỒ KHỐI CHẾ ĐỘ CAPTURE



MINH HỌA KHỞI ĐỘNG CAPTURE

;Tắt modul CCP 1

banksel	CCP1CON
clrf	CCP1CON

;Tắt (dừng) Timer 1

bcf	T1CON, TMR1ON
-----	---------------

;Xóa giá trị Timer 1

clrf	TMR1H
clrf	TMR1L

;Cảm các ngắt dừng cho CCP1

bcf	PIR1, CCP1IF
banksel	PIE1
bcf	PIE1, CCP1IE

;Cấu hình chân CCP1 là ngõ vào

bsf	TRISC, 2
-----	----------

;Khởi động bắt sự kiện tại mỗi cạnh lên thứ 4

banksel	CCP1CON
movlw	b'00000110'
movwf	CCP1CON

;Bật (khởi động) Timer 1

bsf	T1CON, TMR1ON
-----	---------------

;Kiểm tra cờ CCP1IF để bắt sự kiện (Capture)

btfss	PIR1, CCP1IF
goto	\$-1

< Tiếp theo >

TMR1H

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

TMR1L

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

CCPR1H

--	--	--	--	--	--	--	--

CCPR1L

--	--	--	--	--	--	--	--

PIR1

						0	
--	--	--	--	--	--	---	--

CCP1IF

0	0	0	0	0	1	1	0
---	---	---	---	---	---	---	---

T1CON

							0
--	--	--	--	--	--	--	---

TMR1ON

MINH HỌA KHỞI ĐỘNG CAPTURE

;Tắt modul CCP 1

banksel CCP1CON

clrf CCP1CON

;Tắt (dừng) Timer 1

bcf T1CON, TMR1ON

;Xóa giá trị Timer 1

clrf TMR1H

clrf TMR1L

;Cảm các ngắt dừng cho CCP1

bcf PIR1, CCP1IF

banksel PIE1

bcf PIE1, CCP1IE

;Cấu hình chân CCP1 là ngõ vào

bsf TRISC, 2

;Khởi động bắt sự kiện tại mỗi cạnh lên thứ 4

banksel CCP1CON

movlw b'00000110'

movwf CCP1CON

;Bật (khởi động) Timer 1

bsf T1CON, TMR1ON

;Kiểm tra cờ CCP1IF để bắt sự kiện (Capture)

btfss PIR1, CCP1IF

goto \$-1

< Tiếp theo >

TMR1H

TMR1L

TIMER 1 đang tăng giá trị !!!

CCPR1H

 | | | | | | | |

CCPR1L

 | | | | | | | |

PIR1

 | | | | | | | |

CCP1IF

CCP1CON

 | | | | | | | |

T1CON

 | | | | | | | |

1

TMR1ON

;Tắt modul CCP 1

banksel CCP1CON

clrf CCP1CON

;Tắt (dừng) Timer 1

bcf T1CON, TMR1ON

;Xóa giá trị Timer 1

clrf TMR1H

clrf TMR1L

;Cảm các ngắt dừng cho CCP1

bcf PIR1, CCP1IF

banksel PIE1

bcf PIE1, CCP1IE

;Cấu hình chân CCP1 là ngõ vào

bsf TRISC, 2

;Khởi động bắt sự kiện tại mỗi cạnh lên thứ 4

banksel CCP1CON

movlw b'00000110'

movwf CCP1CON

;Bật (khởi động) Timer 1

bsf T1CON, TMR1ON

;Kiểm tra cờ CCP1IF để bắt sự kiện (Capture)

btfs S PIR1, CCP1IF

goto \$-1

< Tiếp theo >



TMR1H

TMR1L

TIMER 1 đang tăng giá trị !!!

CCPR1H

CCPR1L

PIR1

1

CCP1IF

CCP1CON

0 0 0 0 0 1 1 0

T1CON

1

TMR1ON

**Phát hiện
cạnh lên thứ 4
của sự kiện !!!**

MINH HỌA KHỞI ĐỘNG CAPTURE

;Tắt modul CCP 1

banksel CCP1CON

clrf CCP1CON

;Tắt (dừng) Timer 1

bcf T1CON, TMR1ON

;Xóa giá trị Timer 1

clrf TMR1H

clrf TMR1L

;Cảm các ngắt dừng cho CCP1

bcf PIR1, CCP1IF

banksel PIE1

bcf PIE1, CCP1IE

;Cấu hình chân CCP1 là ngõ vào

bsf TRISC, 2

;Khởi động bắt sự kiện tại mỗi cạnh lên thứ 4

banksel CCP1CON

movlw b'00000110'

movwf CCP1CON

;Bật (khởi động) Timer 1

bsf T1CON, TMR1ON

;Kiểm tra cờ CCP1IF để bắt sự kiện (Capture)

btfss PIR1, CCP1IF

goto \$-1

TMR1H

TMR1L

Giá trị hiện tại của Timer 1

CCPR1H

CCPR1L

Giá trị hiện tại của Timer 1

PIR1

1

CCP1IF

CCP1CON

0 0 0 0 0 1 1 0

T1CON

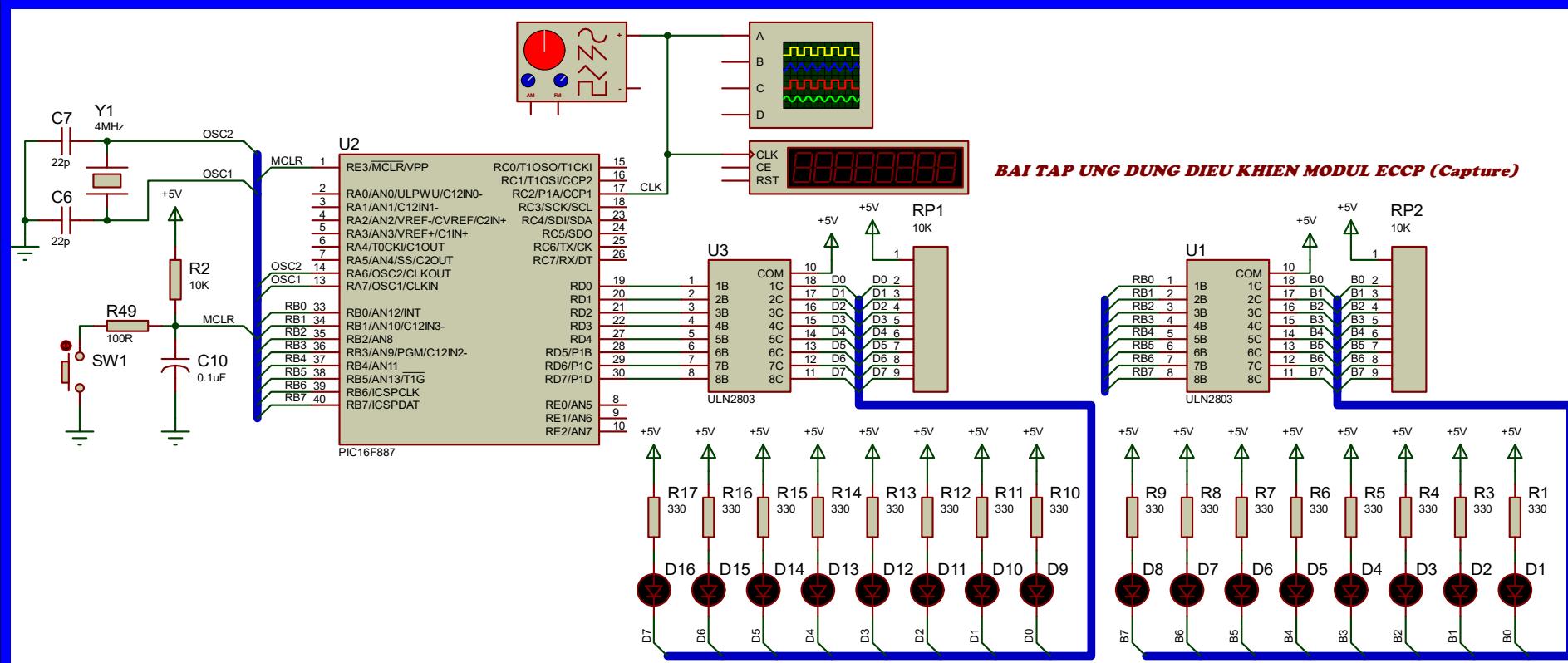
1

TMR1ON

VÍ DỤ MINH HỌA

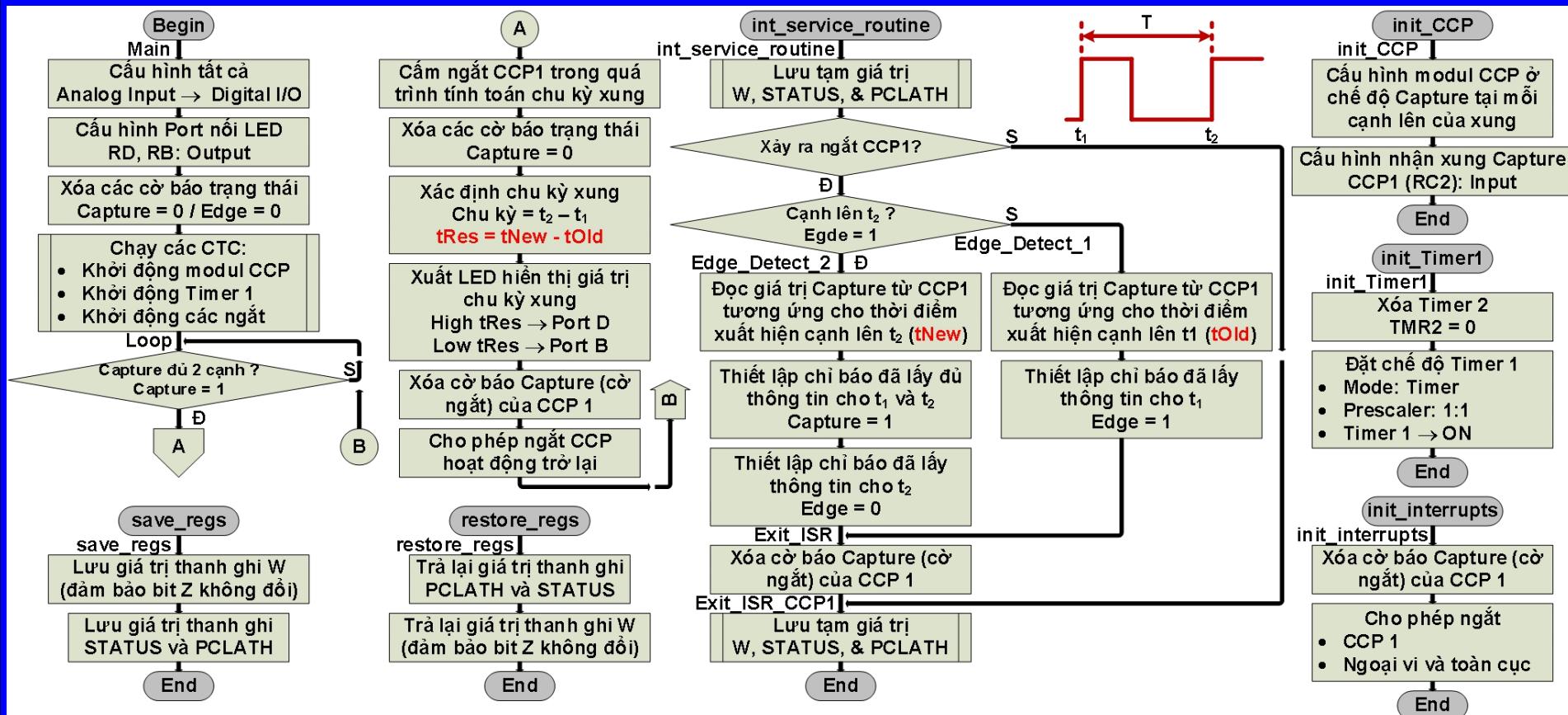
➤ **Ví dụ 1:** Dựa vào sơ đồ, viết chương trình điều khiển để đo chu kỳ của xung vuông ($f = 20\text{Hz} - 25\text{kHz}$) tại chân CCP1 (RC2). Sử dụng modul CCP ở chế độ Capture để đo chu kỳ của xung vuông.

- **Sơ đồ nguyên lý:**



VÍ DỤ MINH HỌA

• Giải thuật:



VÍ DỤ MINH HỌA

- Cấu hình (Hi-Tech C):

```
_CONFIG(FOSC_HS & WDTE_OFF & PWRTE_ON &  
MCLRE_ON & CP_OFF & CPD_OFF & BOREN_OFF &  
IESO_OFF & FCMEN_OFF & LVP_OFF & DEBUG_ON);
```

```
#define _XTAL_FREQ 4000000
```

VÍ DỤ MINH HỌA

- **Chương trình (Hi-Tech C):**

```
unsigned int tOld, tNew;  
unsigned int tword;  
bit edge = 0;  
bit capture = 0;  
void init_CCP(void)  
{  
    CCP1CON = 0X05;  
    TRISC2 = 1;  
}  
void init_Timer1(void)  
{  
    TMR1H = 0;  
    TMR1L = 0;  
    T1CON = 0X01;  
}
```

```
void init_interrupts(void)  
{  
    CCP1IF = 0;  
    CCP1IE = 1;  
    PEIE = 1;  
    GIIE = 1;  
}
```

VÍ DỤ MINH HỌA

- Chương trình (Hi-Tech C):

```
void interrupt ISR(void)
```

```
{  
if(CCP1IF)  
{  
if(!edge)  
{  
tOld = (256*CCPR1H)+CCPR1L;  
edge = 1;  
}  
else  
{  
tNew = (256*CCPR1H)+CCPR1L;  
capture = 1;  
edge = 0;  
}  
CCP1IF = 0;  
}  
}
```

VÍ DỤ MINH HỌA

- **Chương trình (Hi-Tech C):**

```
void main(void)
```

```
{
```

```
ANSEL = 0;
```

```
ANSELH = 0;
```

```
TRISB = 0x00;
```

```
TRISD = 0x00;
```

```
init_CCP();
```

```
init_Timer1();
```

```
init_interrupts();
```

```
while(1)
```

```
{
```

```
if(capture)
```

```
{
```

```
CCP1IE = 0;
```

```
capture = 0;
```

```
tword = ~tOld + tNew+1;
```

```
PORTD = tword / 256;
```

```
PORTB = tword % 256;
```

```
CCP1IF = 0;
```

```
CCP1IE = 1;
```

```
}
```

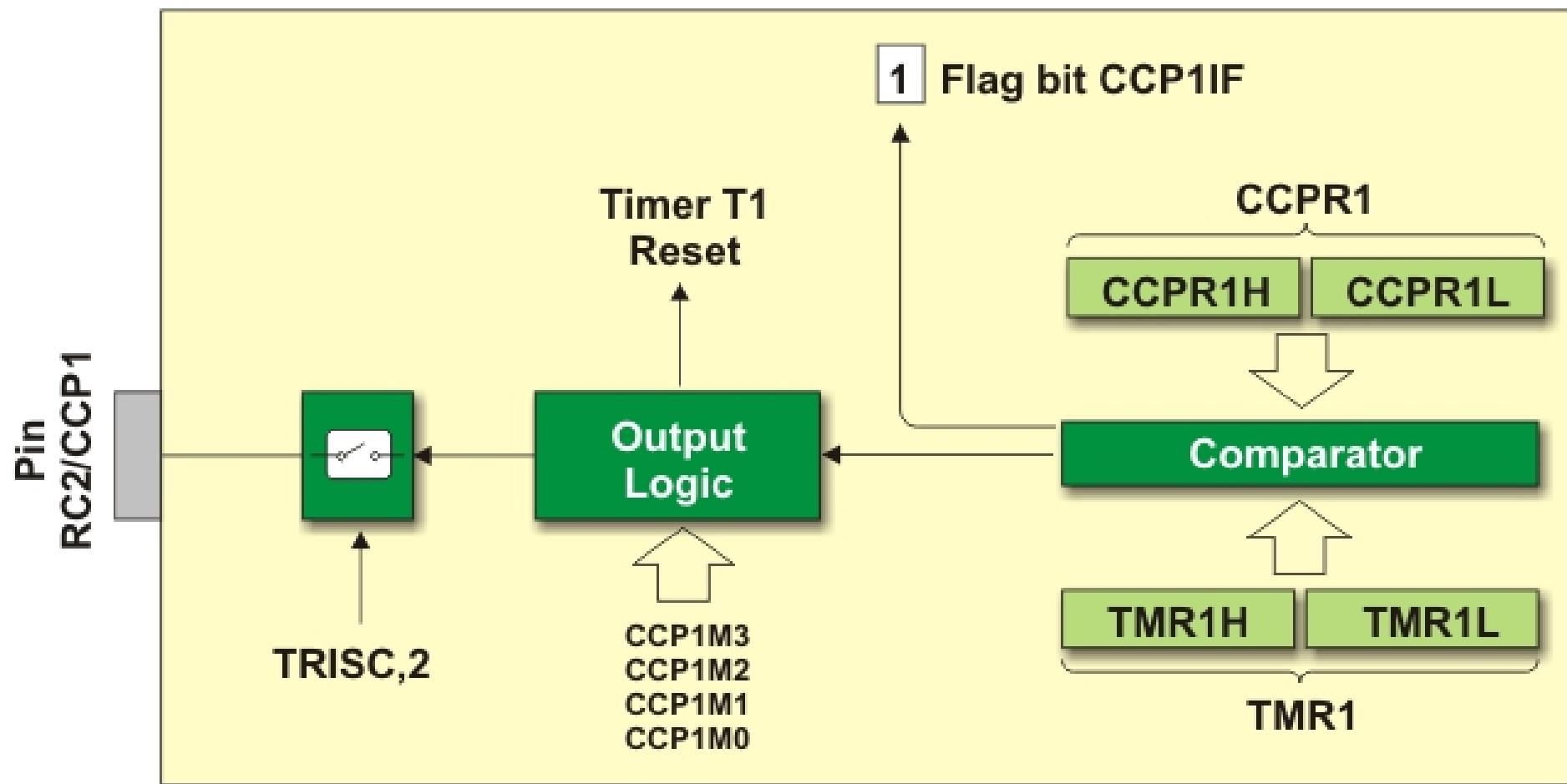
```
}
```

```
}
```

CHẾ ĐỘ COMPARE

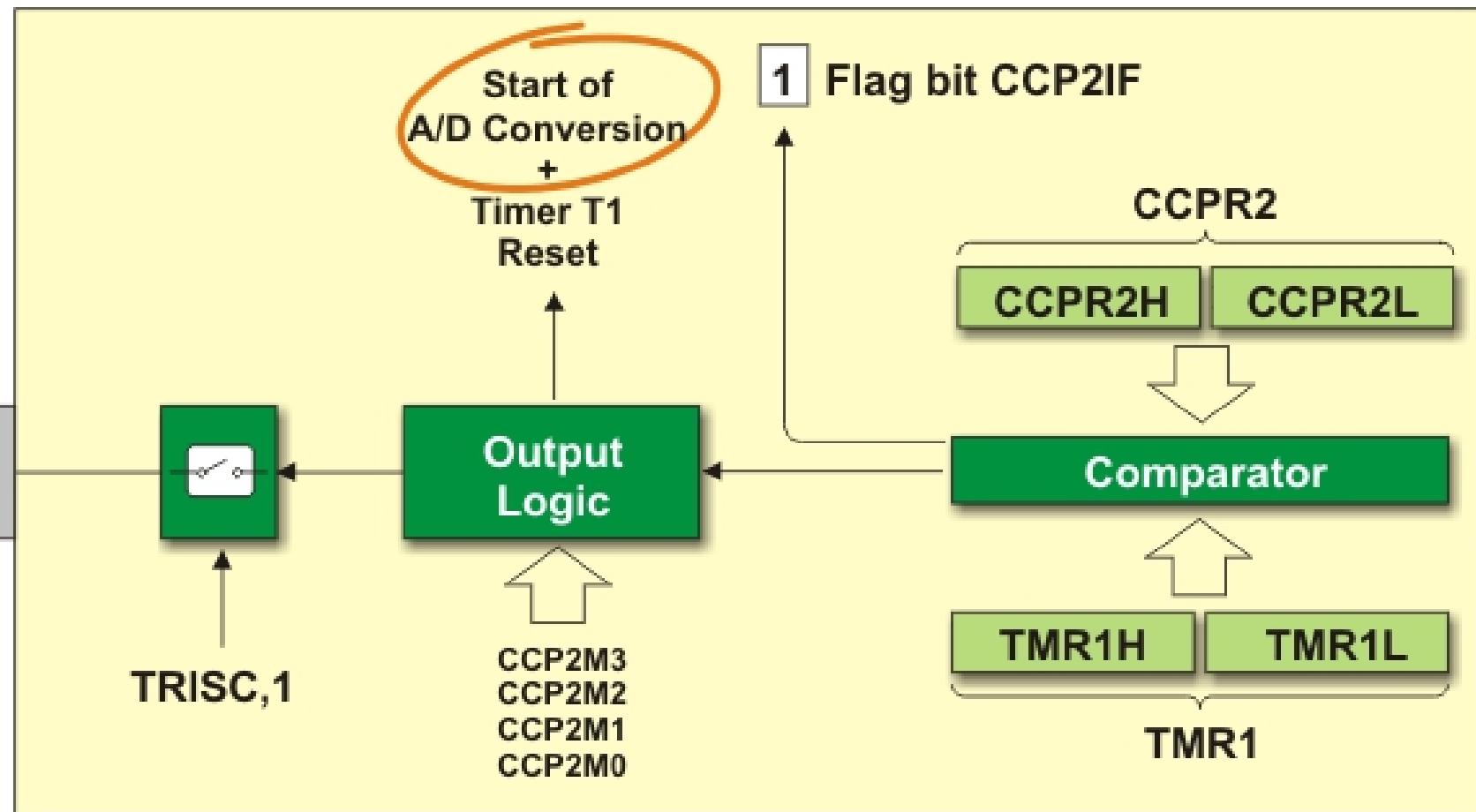
- Trong chế độ này, giá trị trong thanh ghi CCPx (không đổi) được so sánh với giá trị trong thanh ghi TMR1 và khi giá trị của chúng bằng nhau thì:
 - Cờ báo CCPxF được đặt
 - Thay đổi trạng thái logic của chân CCPx
 - Kích các sự kiện đặc biệt
- Điều kiện cần có:
 - Chân CCPx phải được cấu hình là ngõ ra
 - Modul Timer 1 phải được cấu hình hoạt động ở chế độ định thời gian hoặc bộ đếm đồng bộ.

SƠ ĐỒ KHỐI CHẾ ĐỘ COMPARE



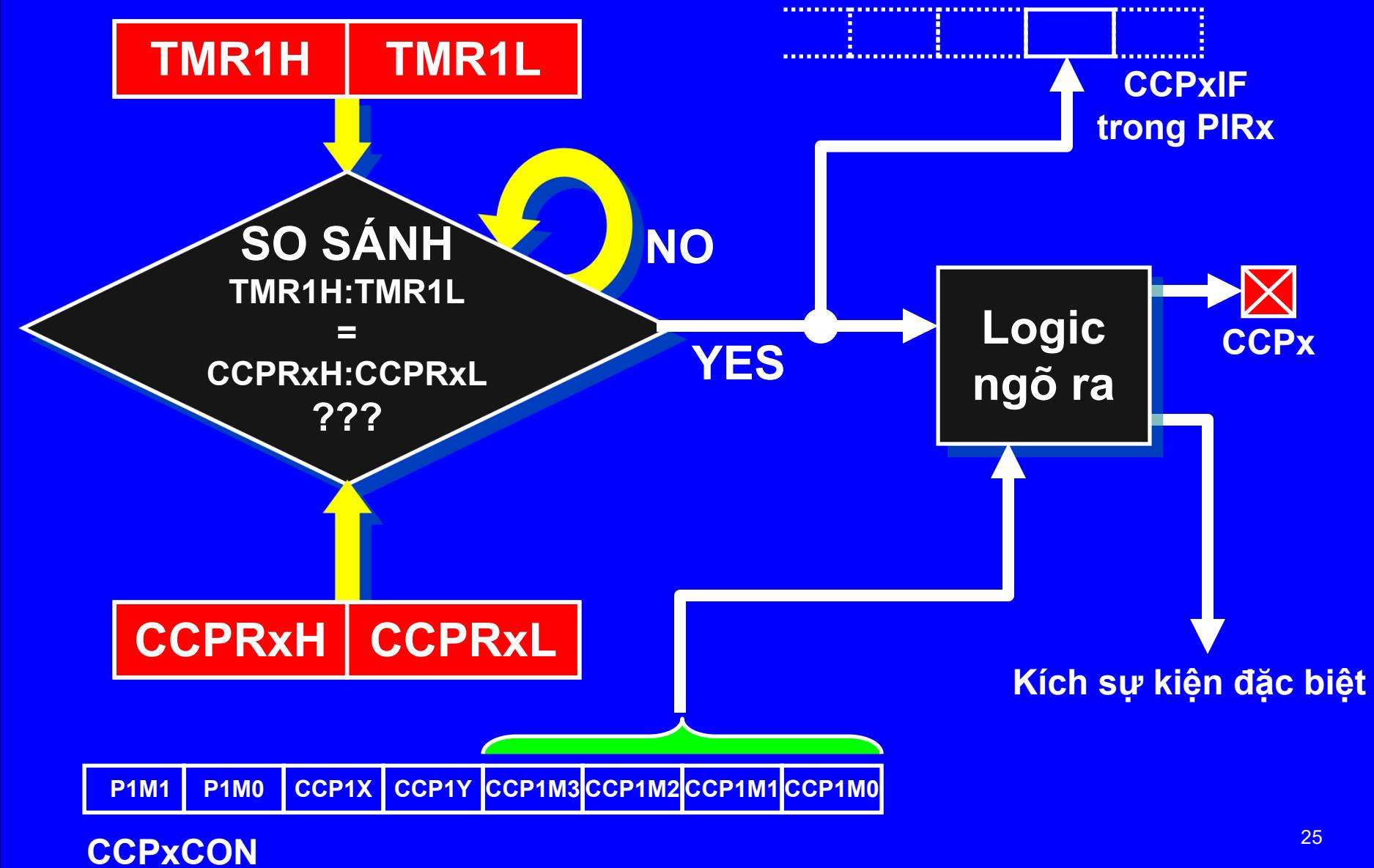
CCP1 trong chế độ COMPARE

SƠ ĐỒ KHỐI CHẾ ĐỘ COMPARE



CCP2 trong chế độ COMPARE

SƠ ĐỒ KHỐI CHẾ ĐỘ COMPARE



SƠ ĐỒ KHỐI CHẾ ĐỘ COMPARE

CCPxM3	CCPxM2	CCPxM1	CCPxM0	Chế độ của CCPx
1	0	0	0	Đặt ngõ ra khi bằng nhau (CCPxIF = 1)
1	0	0	1	Xóa ngõ ra khi bằng nhau (CCPxIF = 1)
1	0	1	0	Tạo ngắt phần mềm khi bằng nhau (CCPxIF = 1, chân CCP1 không bị tác động)
1	0	1	1	Kích các sự kiện đặc biệt (CCPxIF = 1, CCP1 reset TMR1 hoặc TMR2 và bắt đầu thực hiện chuyển đổi AD nếu được cho phép)

KÍCH SỰ KIỆN ĐẶC BIỆT



CCPxCON

MINH HỌA KHỞI ĐỘNG COMPARE

;Tắt modul CCP 1

banksel CCP1CON

clrf CCP1CON

;Tắt (dừng) Timer 1

bcf T1CON, TMR1ON

;Xóa giá trị Timer 1

clrf TMR1H

clrf TMR1L

;Cấm các ngắt dùng cho CCP1

;và đảm bảo rằng các cờ của nó được xóa

banksel PIE1

bcf PIE1, CCP1IE

banksel PIR1

bcf PIR1, CCP1IF

;Cấu hình chân CCP1 là ngõ ra

banksel TRISC

bcf TRISC, 2

;Khởi động chế độ so sánh để đặt ngõ ra khi bằng nhau

banksel CCP1CON

movlw b'00001000'

movwf CCP1CON

;Đặt giá trị cần so sánh vào thanh ghi

;CCPR1H:CCPR1L

banksel CCPR1H

movlw b'10000000'

movwf CCPR1H

clrf CCPR1L

;Bật (khởi động) Timer 1

bsf T1CON, TMR1ON

;Kiểm tra cờ CCP1IF để biết giá trị Timer1 bằng CCPRx

btfs S, CCP1IF

goto \$-1

< Tiếp theo >

TMR1H

TMR1H:TMR1L = 1000 0000 0000 0000

TMR1L

TMR1H:TMR1L = 1000 0000 0000 0000
(bằng giá trị của CCPR1H:CCPR1L)

CCPR1H

1	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---

CCPR1L

0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---

T1CON

								1
--	--	--	--	--	--	--	--	---

TMR1ON

CCP1CON

0	0	0	0	1	0	0	0
---	---	---	---	---	---	---	---

PIR1

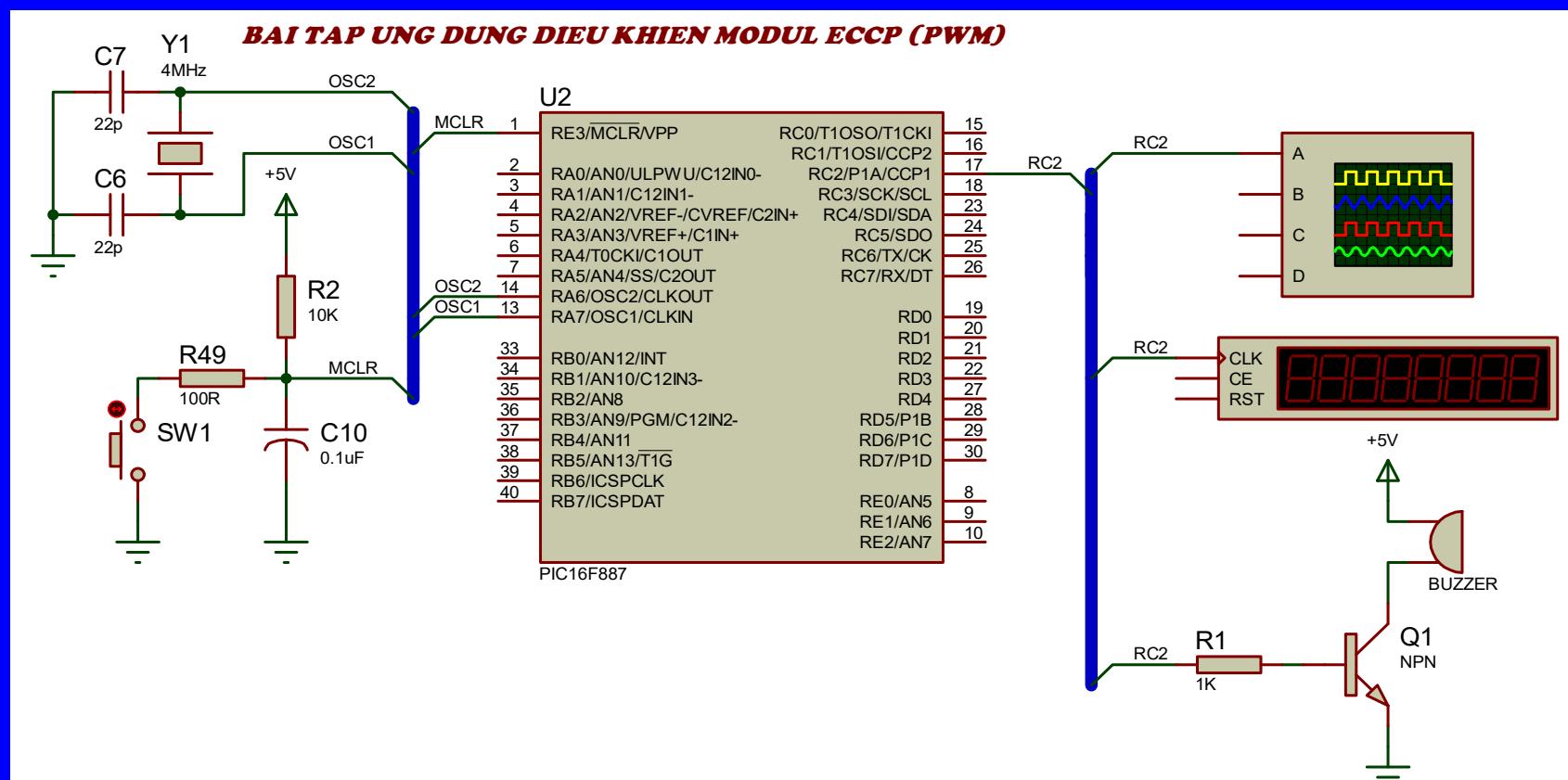
						1		
--	--	--	--	--	--	---	--	--

CCP1IF

VÍ DỤ MINH HỌA

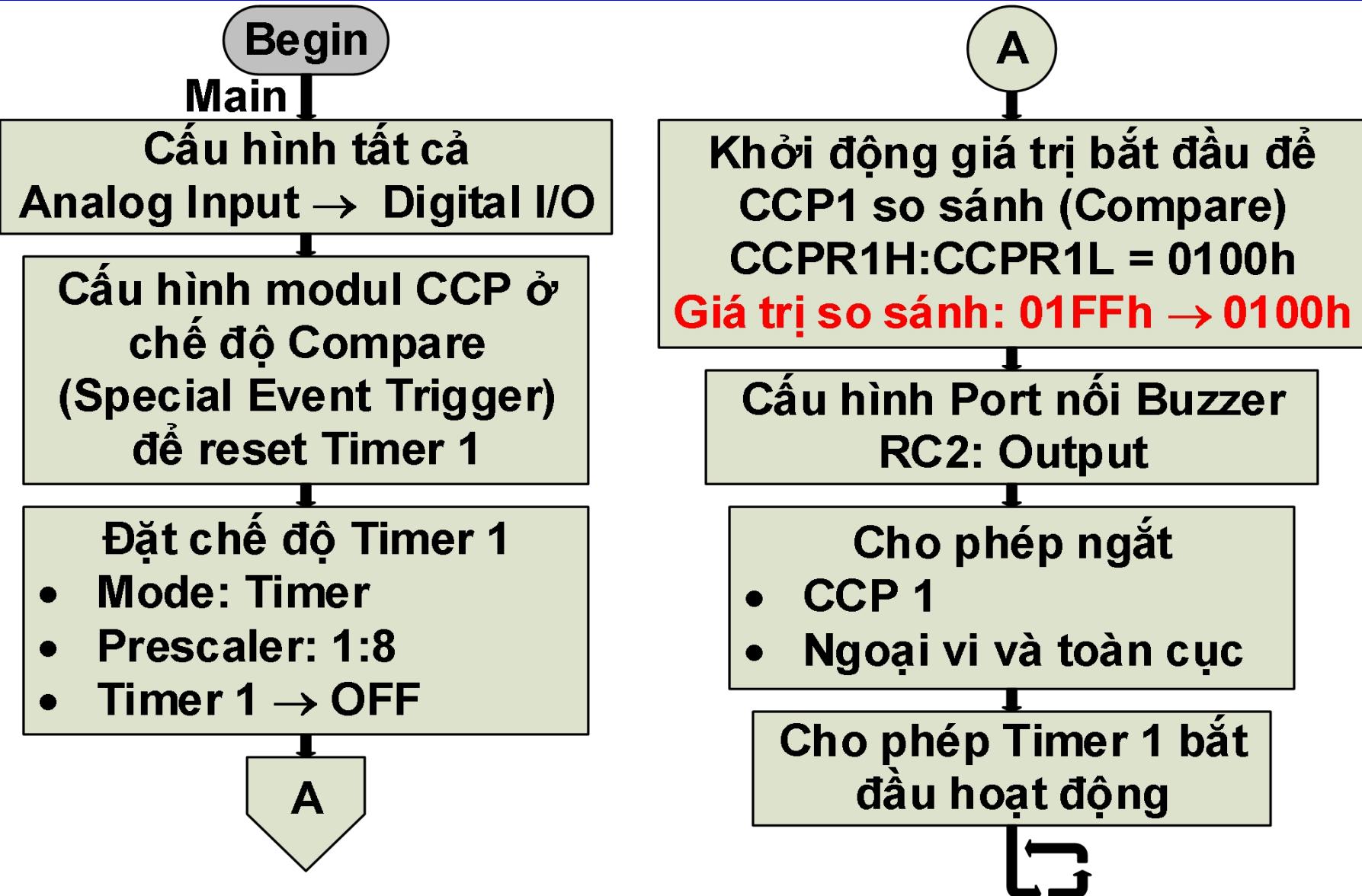
➤ **Ví dụ 2:** Dựa vào sơ đồ, viết chương trình điều khiển tạo xung có tần số thay đổi trong khoảng từ 122,3Hz - 244,1Hz tại chân CCP1 (RC2). Sử dụng modul CCP ở chế độ Compare để tạo xung.

- **Sơ đồ nguyên lý:**



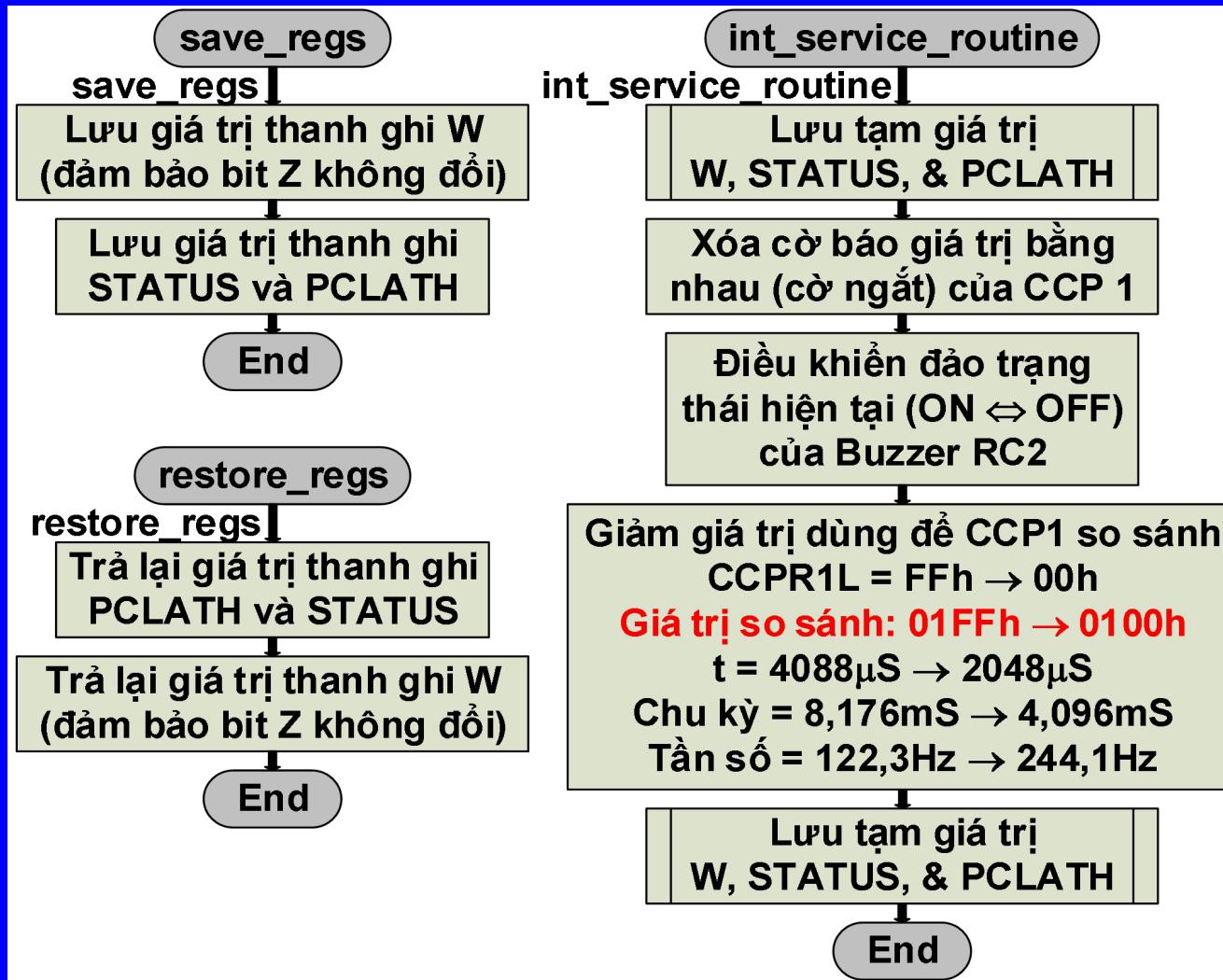
VÍ DỤ MINH HỌA

- Giải thuật:



VÍ DỤ MINH HỌA

- Giải thuật:



$$t = TMR1H:TMR1L \times \frac{4}{f_{osc}} \times Pre. = CCPR1H:CCPR1L \times \frac{4}{f_{osc}} \times Pre.$$

VÍ DỤ MINH HỌA

- Cấu hình (Hi-Tech C):

```
_CONFIG(FOSC_HS & WDTE_OFF & PWRTE_ON &  
MCLRE_ON & CP_OFF & CPD_OFF & BOREN_OFF &  
IESO_OFF & FCMEN_OFF & LVP_OFF & DEBUG_ON);
```

```
#define _XTAL_FREQ 4000000
```

VÍ DỤ MINH HỌA

- **Chương trình (Hi-Tech C):**

```
void interrupt OC_interrupt(void)
{
    CCP1IF = 0;
    PORTC ^= 0X04;
    --CCPR1L;
}
```

```
void init_CCP(void)
{
    CCP1CON = 0X0B;
    CCPR1L = 0X00;
    CCPR1H = 0X01;
    TRISC2 = 0;
}
```

```
void init_Timer1(void)
{
    T1CON = 0X30;
}

void init_interrupts(void)
{
    CCP1IE = 1;
    PEIE = 1;
    GIE = 1;
}
```

VÍ DỤ MINH HỌA

- **Chương trình (Hi-Tech C):**

```
void main(void)
```

```
{
```

```
init_CCP();
```

```
init_Timer1();
```

```
init_interrupts();
```

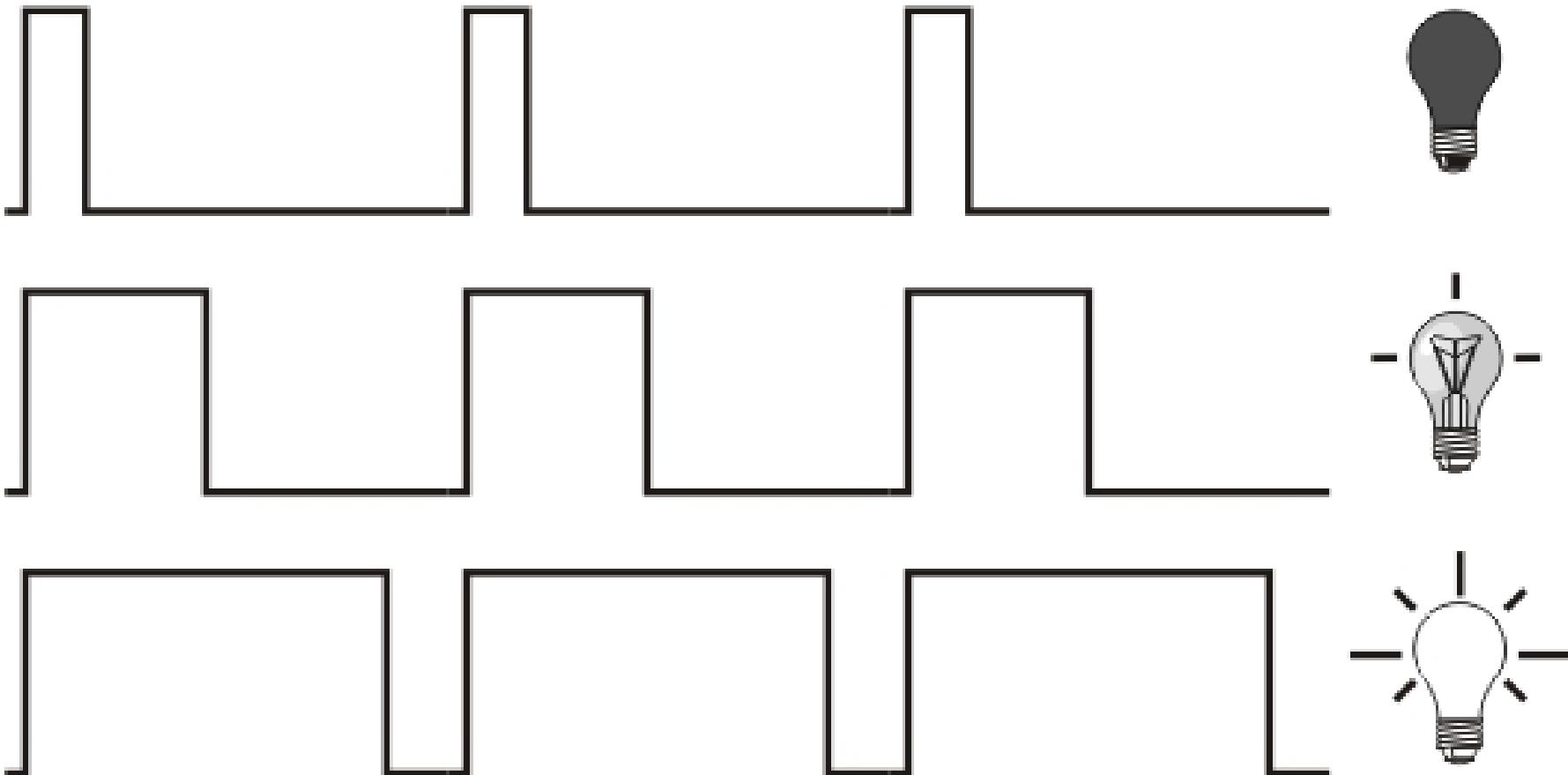
```
TMR1ON = 1;
```

```
while(1);
```

```
}
```

CHẾ ĐỘ PWM

- Trong chế độ này, tại chân CCPx sẽ xuất ra tín hiệu PWM (Pulse-Width Modulated)



CHẾ ĐỘ PWM

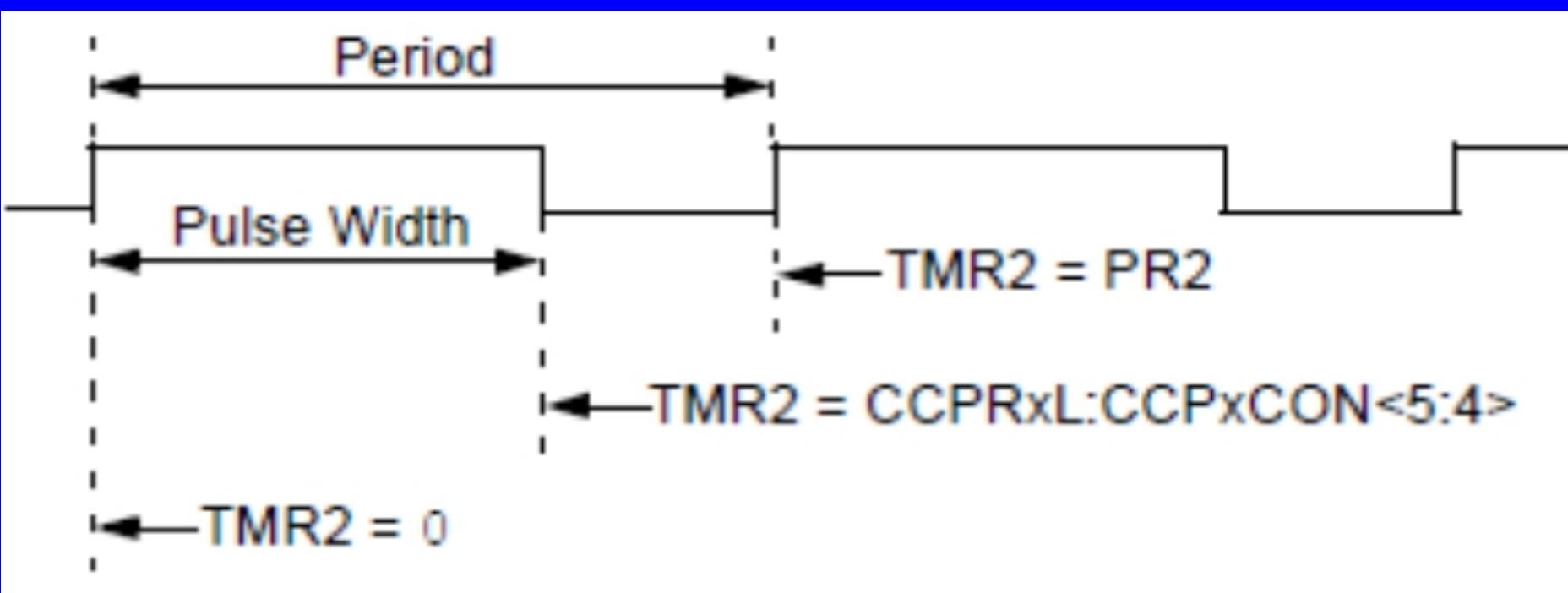
- Chu kỳ nhiệm vụ (Duty cycle), chu kỳ (Period) và độ phân giải (Resolution) có thể điều khiển được bằng các thanh ghi sau:

PR2

T2CON

CCP R_x L

CCPxCON



CHẾ ĐỘ PWM

➤ Điều kiện cần có:

- Chân CCPx phải được cấu hình là ngõ ra
- Đặc biệt lưu ý rằng độ rộng xung PWM phải luôn luôn nhỏ hơn chu kỳ PWM, nếu không sẽ làm cho ngõ ra không thể thay đổi trạng thái
- Bộ chia tỉ lệ sau của Timer 2 (Postscaler) không được dùng để tạo ra chu kỳ PWM dài hơn.

CHẾ ĐỘ PWM

➤ Các công thức tính toán của PWM:

- Chu kỳ PWM (PWM Period):

$$T_{PWM} = ([PR2] + 1) \times 4 \times \frac{1}{f_{osc}} \times Pre.$$

T_{PWM} : Chu kỳ xung PWM (μs)

f_{osc} : Tần số dao động (MHz)

Pre.: Giá trị hệ số Prescaler (Pre = 1, 4, 16)

[PR2]: Giá trị cần ghi vào thanh ghi PR2.

$$0 \leq [PR2] \leq 255$$

CHẾ ĐỘ PWM

➤ Các công thức tính toán của PWM:

- Chu kỳ PWM (PWM Period):

$$T_{PWM} = ([PR2] + 1) \times 4 \times \frac{1}{f_{osc}} \times Pre.$$

TẦN SỐ PWM [KHZ]	1.22	4.88	19.53	78.12	156.3	208.3
Prescaler của Timer 2	16	4	1	1	1	1
Thanh ghi PR2	FFh	FFh	FFh	3Fh	1Fh	17h

Ví dụ minh họa các tần số xung PWM (PIC sử dụng thạch anh 20MHz)

CHẾ ĐỘ PWM

➤ Các công thức tính toán của PWM:

- Độ rộng xung PWM (PWM Pulse Width):

$$PW = (CCPRxL: CCPxCON < 5:4 >) \times \frac{1}{f_{osc}} \times Pre$$

PW: Độ rộng xung PWM (μs)

f_{osc} : Tần số dao động (MHz)

Pre: Giá trị hệ số Prescaler (Pre = 1, 4, 16)

[CCPRxL:CCPxCON<5:4>]: Con số có giá trị 10 bit (gồm 8 bit MSB trong thanh ghi CCPRxL và 2 bit LSB trong thanh ghi CCPxCON).

$$0 \leq [CCPRxL: CCPxCON < 5:4 >] \leq 1023$$

CHẾ ĐỘ PWM

➤ Các công thức tính toán của PWM:

- Tỉ số chu kỳ nhiệm vụ (Duty Cycle Ratio):

$$DCR = \frac{[CCPRxL: CCPxCON < 5:4 >]}{4 \times ([PR2] + 1)} \times 100\%$$

DCR: Tỉ số chu kỳ nhiệm vụ (%)

[PR2]: Giá trị của thanh ghi PR2.

[CCPRxL:CCPxCON<5:4>]: Con số có giá trị 10 bit (gồm 8 bit MSB trong thanh ghi CCPRxL và 2 bit LSB trong thanh ghi CCPxCON).

THANH GHI CỦA MODUL CCP

CCP1 Control Register (CCP1CON)

P1M1 | P1M0 | CCP1X | CCP1Y | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0

BIT	Chức năng
CCP1M<3:0>	Các bit chọn chế độ CCP, cấu hình modul này như là ngõ vào Capture, ngõ ra Compare hoặc ngõ ra PWM
CCP1<X:Y>	2 bit LSB qui định chu kỳ nhiệm vụ cho xung PWM (8 bit MSB được chứa trong thanh ghi CCPR1L)
P1M<1:0>	Các bit cấu hình ngõ ra PWM này chỉ có thể dùng cho modul ECCP (Enhanced CCP). Điều khiển lái ngõ ra toàn cầu hoặc nữa cầu.

CHẾ ĐỘ PWM

➤ Các công thức tính toán của PWM:

- Độ phân giải PWM (PWM Resolution):

$$Res = \frac{\log [4 \times ([PR2] + 1)]}{\log (2)}$$

Res: Độ phân giải PWM (bit)

[PR2]: Giá trị của thanh ghi PR2.

TẦN SỐ PWM	1.22 KHZ	4.88 KHZ	19.53 KHZ	78.12 KHZ	156.3 KHZ	208.3 KHZ
Prescale Timer 2	16	4	1	1	1	1
Giá trị PR2	FFh	FFh	FFh	3Fh	1Fh	17h
Độ phân giải lớn nhất	10	10	10	8	7	6

Ví dụ minh họa quan hệ tần số xung PWM và độ phân giải
của PWM (PIC sử dụng thạch anh 20MHz)

CHẾ ĐỘ PWM

➤ Các công thức tính toán của PWM:

- Độ phân giải PWM (PWM Resolution):

$$Res = \frac{\log [4 \times ([PR2] + 1)]}{\log (2)}$$

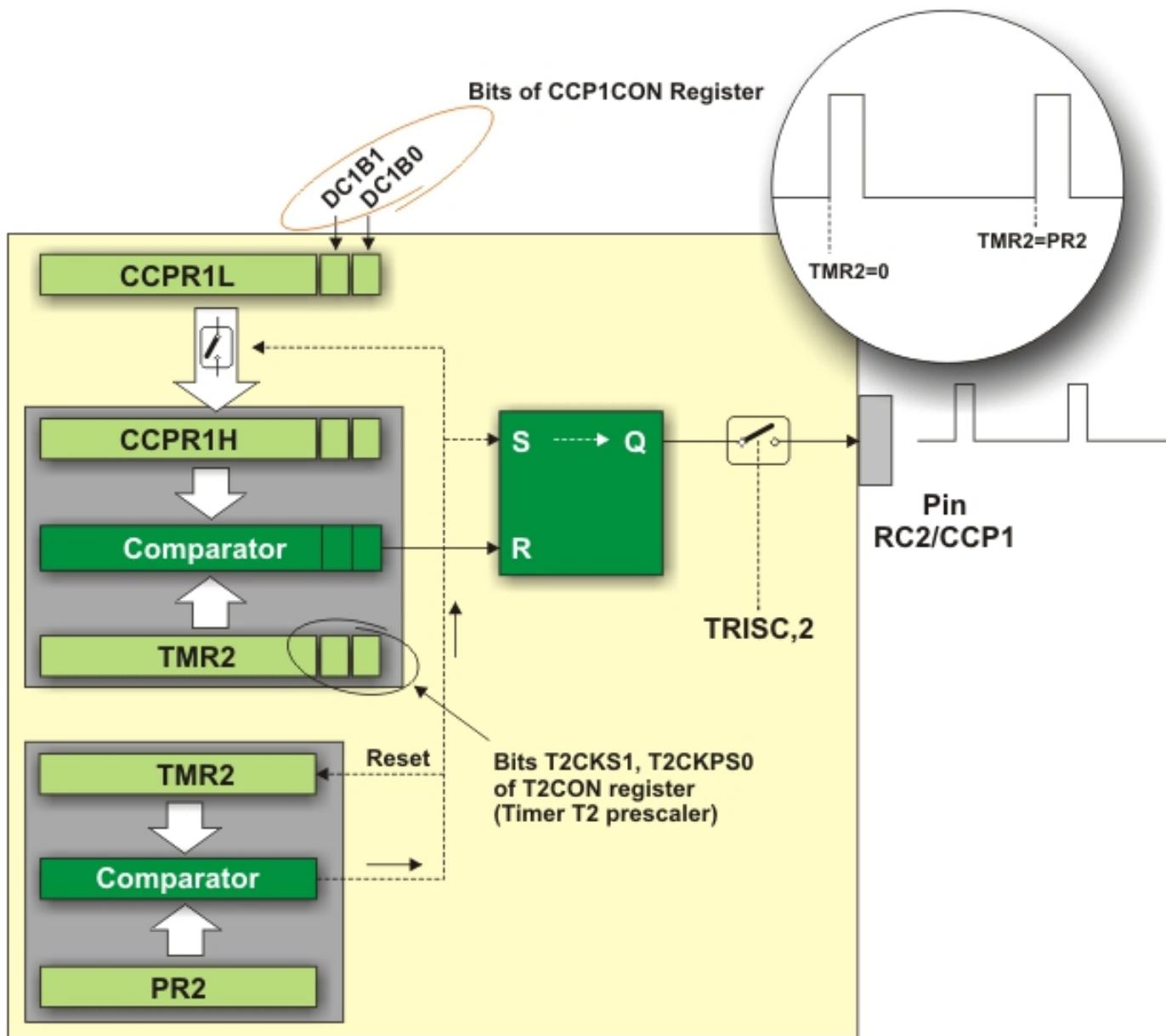
Res: Độ phân giải PWM (bit)

[PR2]: Giá trị của thanh ghi PR2.

TẦN SỐ PWM	1.22 KHZ	4.90 KHZ	19.61 KHZ	76.92 KHZ	153.85 KHZ	200.0 KHZ
Prescale Timer 2	16	4	1	1	1	1
Giá trị PR2	65h	65h	65h	19h	0Ch	09h
Độ phân giải lớn nhất	8	8	8	6	5	5

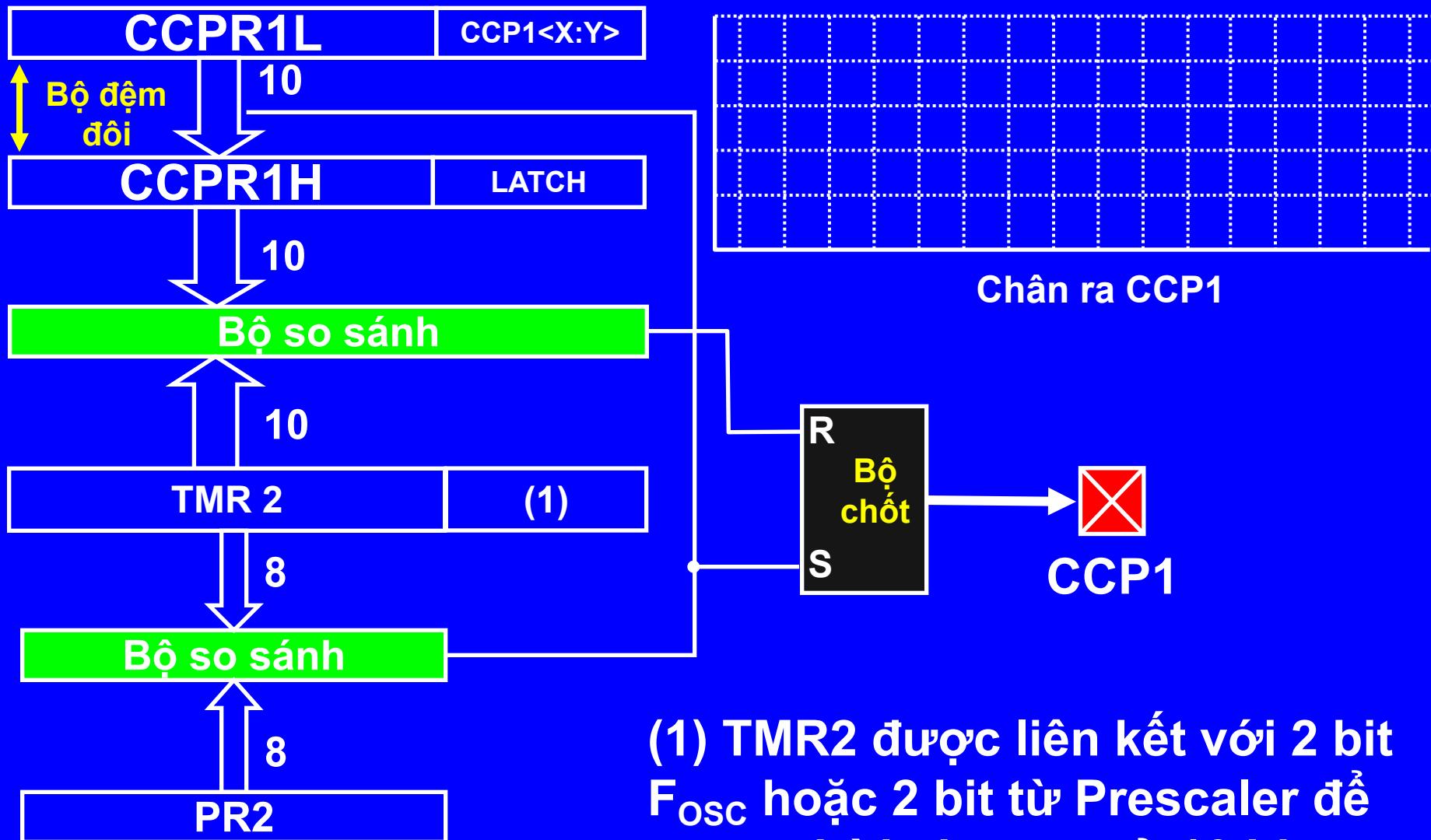
Ví dụ minh họa quan hệ tần số xung PWM và độ phân giải
của PWM (PIC sử dụng thạch anh 8MHz)

CHẾ ĐỘ PWM

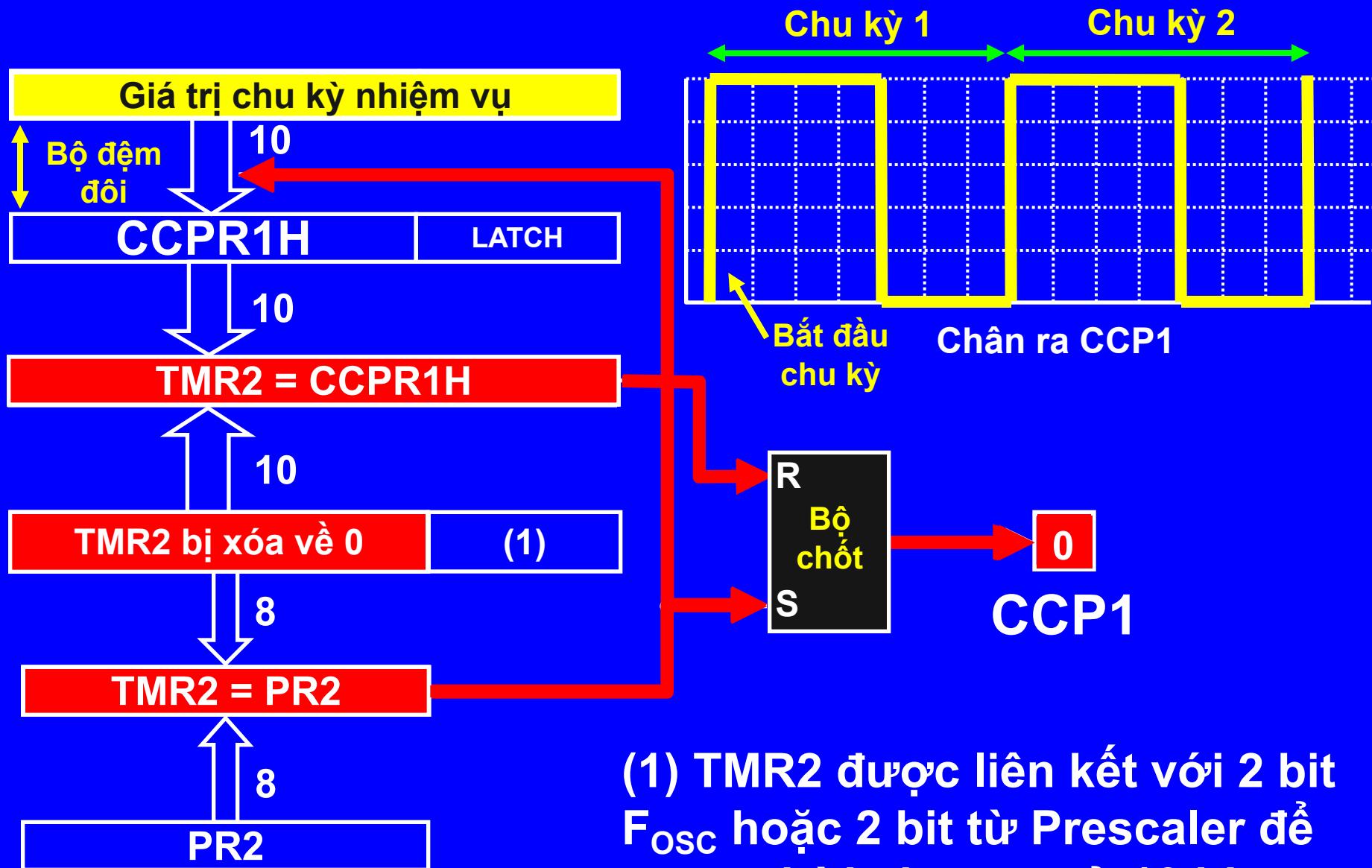


CCP1 trong chế độ PWM

SƠ ĐỒ KHỐI PWM



SƠ ĐỒ KHỐI PWM



PWM: MỘT SỐ ĐIỂM CẦN LƯU Ý

➤ Để cấu hình modul CCP ở chế độ PWM:

- **BƯỚC 1: Cắm xuất xung PWM**
 - Cắm bộ điều khiển xuất của chân CCPx (cấu hình chân này là ngõ vào)
- **BƯỚC 2: Đặt giá trị chu kỳ PWM** (đặt vào thanh ghi PR2 giá trị thích hợp)
- **BƯỚC 3: Cấu hình modul CCP ở chế độ PWM** (đặt vào thanh ghi CCPxCON giá trị thích hợp)
- **BƯỚC 4: Đặt giá trị chu kỳ nhiệm vụ PWM** (đặt vào thanh ghi CCPRxL và các bit DCxB<1:0> của thanh ghi CCPxCON

PWM: MỘT SỐ ĐIỂM CẦN LƯU Ý

- Để cấu hình modul CCP ở chế độ PWM:
 - **BƯỚC 5:** Cấu hình và kích hoạt Timer 2
 - Xóa cờ ngắt TMR2IF của thanh ghi PIR1 ($TMR2IF = 0$)
 - Đặt giá trị prescaler của Timer 2 (đặt vào các bit T2CKPS của thanh ghi T2CON giá trị thích hợp)
 - Kích hoạt Timer 2 ($TMR2ON = 1$)
 - **BƯỚC 6:** Cho phép xuất xung PWM
 - Chờ cho đến khi Timer 2 tràn ($TMR2IF = 1$)
 - Cho phép bộ điều khiển xuất của chân CCPx (cấu hình chân này là ngoẽ ra)

MINH HỌA KHỞI ĐỘNG PWM

;Tắt chân CCPx bằng cách cấu hình CCPx là ngõ vào

```
banksel TRISC
```

```
bsf TRISC, 2
```

;Xóa Timer 2

```
banksel TMR2
```

```
clrf TMR2
```

;Thiết lập chu kỳ và chu kỳ nhiệm vụ với $f_{osc} = 8MHz$

```
banksel PR2
```

```
movlw b'01111111' ;
```

```
movwf PR2 ;Nạp giá trị chu kỳ 64μS
```

```
banksel CCPR1L
```

```
movlw b'00011111' ;
```

```
movwf CCPR1L ;Nạp giá trị chu kỳ nhiệm vụ của xung PWM (25%)
```

;Cấu hình modul ECCP cho PWM đơn

với P1A kích hoạt mức cao và các bit

;LSB của chu kỳ nhiệm vụ là '10'

```
movlw b'00101100' ;
```

```
movwf CCP1CON ;Modul ECCP được cấu hình cho PWM và các bit
;LSB của chu kỳ được nạp
```

;Bật chân CCPx trở lại

```
banksel TRISC
```

```
bcf TRISC, 2 ;Làm cho CCP1 là ngõ ra
```

;Timer 2 chạy khi TMR2ON = 1

```
banksel T2CON
```

```
movlw b'00000100' ;
```

```
movwf T2CON ;Prescaler và Postscaler
;đều có giá trị 1:1
```

TMR2

0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---

PR2

0	1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---

CCPR1L

0	0	0	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---

CCP1CON

0	0	1	0	1	1	0	0
---	---	---	---	---	---	---	---

LSB của chu kỳ nhiệm vụ Chế độ PWM
DC1B<1:0> CCP1M <3:0>

T2CON

0	0	0	0	0	0	1	0	0
---	---	---	---	---	---	---	---	---

Prescaler
TOUTPS<3:0>

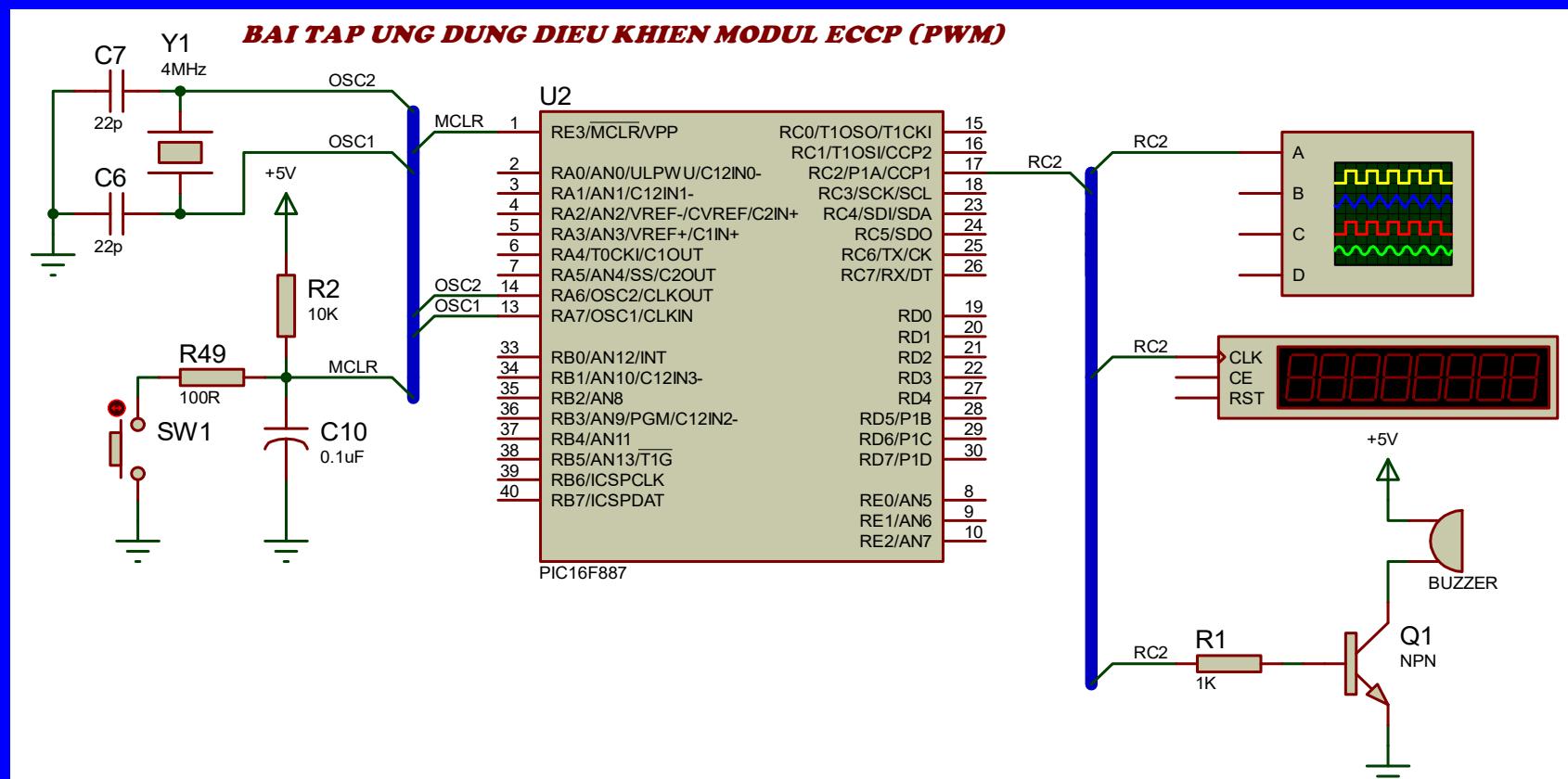
Postscaler
T2CKPS<1:0>

TMR2ON

VÍ DỤ MINH HỌA

➤ **Ví dụ 3:** Dựa vào sơ đồ, viết chương trình điều khiển tạo xung PWM có $f_{PWM} = 4\text{KHz}$, $D = 50\%$ tại chân CCP1 (RC2). Sử dụng modul CCP ở chế độ PWM để tạo xung.(fosc=4Mhz, pre=1)

- **Sơ đồ nguyên lý:**



Tính toán giá trị thanh ghi

- $f_{PWM} = 4\text{KHz}$, $f_{osc} = 4\text{MHz}$, chọn prescaler = 1

$$T_{PWM} = \frac{1}{f_{PWM}} = \frac{1}{4000} = 250\mu s$$

$$T_{PWM} = ([PR2] + 1) \times 4 \times \frac{1}{f_{osc}} \times Pre.$$

- Thay T_{PWM} , f_{osc} , pre vào ta tính được $PR2 = 249$
- Với CCP1: $D\% = 50\%$

Tính được

$$PW = \frac{D \cdot T_{PWM}}{100} = \frac{50 \cdot 250}{100} = 125\mu s$$

Tính toán giá trị thanh ghi

- Từ công thức :

$$PW = (CCPRxL: CCPxCON < 5:4 >) \times \frac{1}{fosc} \times Pre$$

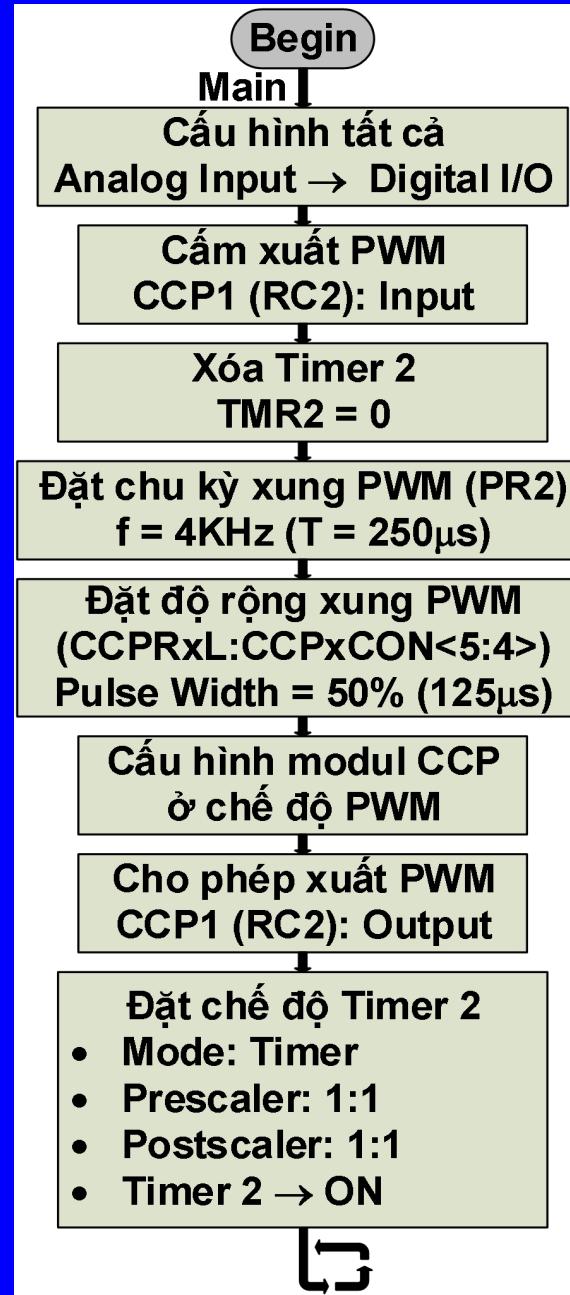
- Đặt E =(CCPR1L:CCP1CON<5:4>) suy ra E=125x4=500
- CCPR1L = E/4 = 125 ;
- CCP1CON<5:4> = E%4 = 0 suy ra giá trị từ bảng TT
ta có :

DC1B1=0 và DC1B0=0

CCPxCON<5:4>	DCxB1	DCxB0
0	0	0
1	0	1
2	1	0
3	1	1

VÍ DỤ MINH HỌA

- **Giải thuật:**



VÍ DỤ MINH HỌA

- Cấu hình (Hi-Tech C):

```
_CONFIG(FOSC_HS & WDTE_OFF & PWRTE_ON &  
MCLRE_ON & CP_OFF & CPD_OFF & BOREN_OFF &  
IESO_OFF & FCMEN_OFF & LVP_OFF & DEBUG_ON);
```

```
#define _XTAL_FREQ 4000000
```

VÍ DỤ MINH HỌA

• Chương trình (Hi-Tech C):

```
void setup_ccp(void)
{
    TRISC2 = 1;
    TMR2 = 0x00;
    PR2 = 249;
    CCPR1L = 125;
    CCP1CON = 0x0C;
    DC1B1=0;DC1B0=0;
    TRISC2 = 0;
    T2CON = 0X00;
    TMR2ON =1;
}
```

```
void main(void)
{
    ANSEL=ANSELH=0;
    setup_ccp();
    while(1);
}
```

BÀI TẬP ỨNG DỤNG

- **Bài tập 1:** Viết chương trình điều khiển tạo xung PWM có $f_{\text{PWM}} = 5\text{KHz}$, $D = 40\%$ tại chân CCP2. Sử dụng modul CCP ở chế độ PWM để tạo xung.
- **Bài tập 2:** Viết chương trình điều khiển tạo xung PWM có $f_{\text{PWM}} = 10\text{KHz}$, $D1 = 80\%$ tại chân CCP1 và $D2 = 20\%$ tại chân CCP2. Sử dụng modul CCP ở chế độ PWM để tạo xung.

Biết rằng $f_{\text{osc}} = 4\text{MHz}$

BÀI TẬP ỨNG DỤNG

- **Bài tập 3: Viết chương trình điều khiển tạo xung PWM có $f_{\text{PWM}} = 8\text{KHz}$, D1 = 60% tại chân CCP1 và D2 = 40% tại chân CCP2. Sử dụng modul CCP ở chế độ PWM để tạo xung.**

Biết rằng $f_{\text{osc}} = 4\text{MHz}$

Bài tập 4 :Thiết kế mạch điện tạo ra 2 xung PWM có

Tần số xung PWM: $f_{\text{PWM}} = 5 \text{ kHz}$

Chu kì nhiệm vụ của xung tại chân RC2 là $D_{\text{RC2}} = 30\%$

Chu kì nhiệm vụ của xung tại chân RC1 là $D_{\text{RC1}} = 70\%$

Cho biết mạch điện sử dụng vi điều khiển PIC16F887 với tần số X_TAL 4MHz.

Yêu cầu sử dụng modul CCP hoạt động ở chế độ PWM để thực hiện yêu cầu tạo xung PWM.

- a) Xác định đầu vào/ đầu ra của mạch điện. [CDRMH3]
- b) Vẽ sơ đồ mạch nguyên lý.[CDRMH4]
- c) Tính toán các giá trị thanh ghi sử dụng chức năng PWM của khối CCP. [CDRMH2]
- d) Vẽ lưu đồ giải thuật. [CDRMH6]
- e) Viết chương trình. [CDRMH6]

Bài tập 5: Thiết kế mạch điện tạo ra 2 xung PWM có :

Tần số xung PWM: $f_{\text{PWM}} = 5 \text{ kHz}$

Nhấn-nhả SW4 : tạo xung PWM có chu kì nhiệm vụ tại chân RC2 là D =40%

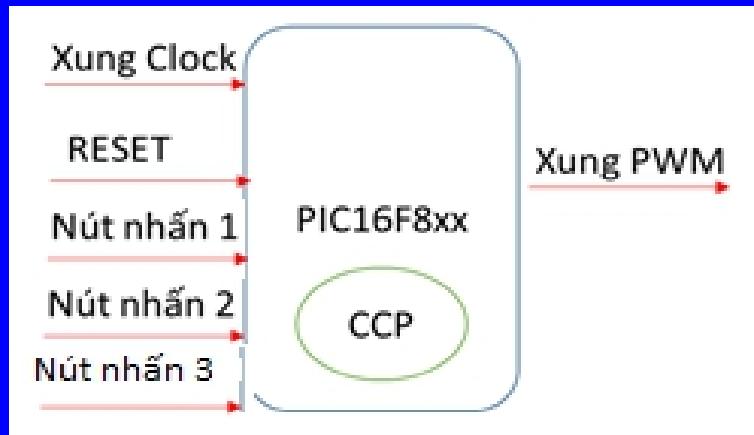
Nhấn-nhả SW5 :tạo xung PWM có chu kì nhiệm vụ tại chân RC1 là D =60%

Nhấn-nhả SW6 :dùng phát xung PWM

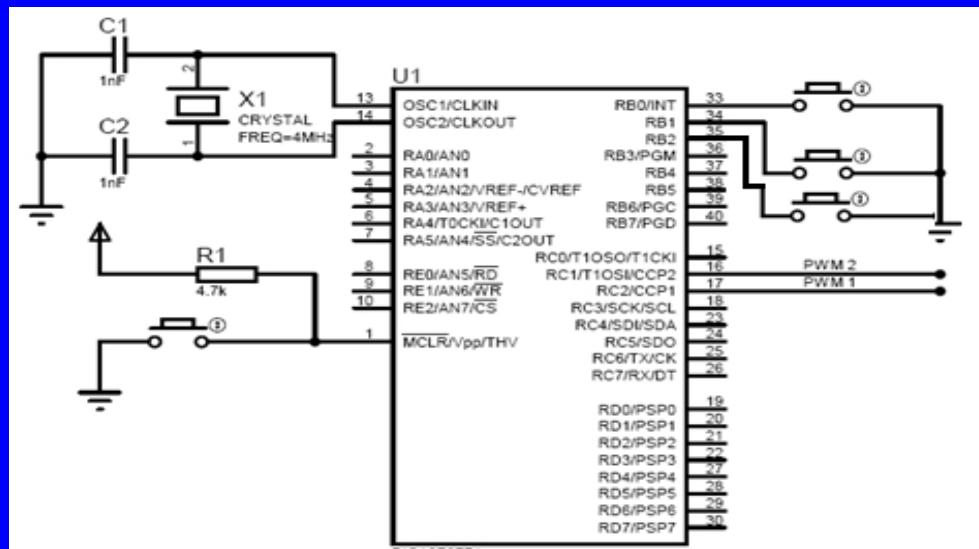
Cho biết mạch điện sử dụng vi điều khiển PIC16F8xx, 3 nút nhấn SW4 (nối chân RB0), SW5(nối chân RB1), SW6 (nối chân RB2).

- a) Xác định đầu vào/ đầu ra của mạch điện. [CDRMH3]
- b) Vẽ sơ đồ mạch nguyên lý.[CDRMH4]
- c) Tính toán các giá trị thanh ghi sử dụng chức năng PWM của khối CCP. [CDRMH2]
- d) Vẽ lưu đồ giải thuật. [CDRMH6]
- e) Viết chương trình. [CDRMH6]

a) Xác định đầu vào/ đầu ra của mạch điện.

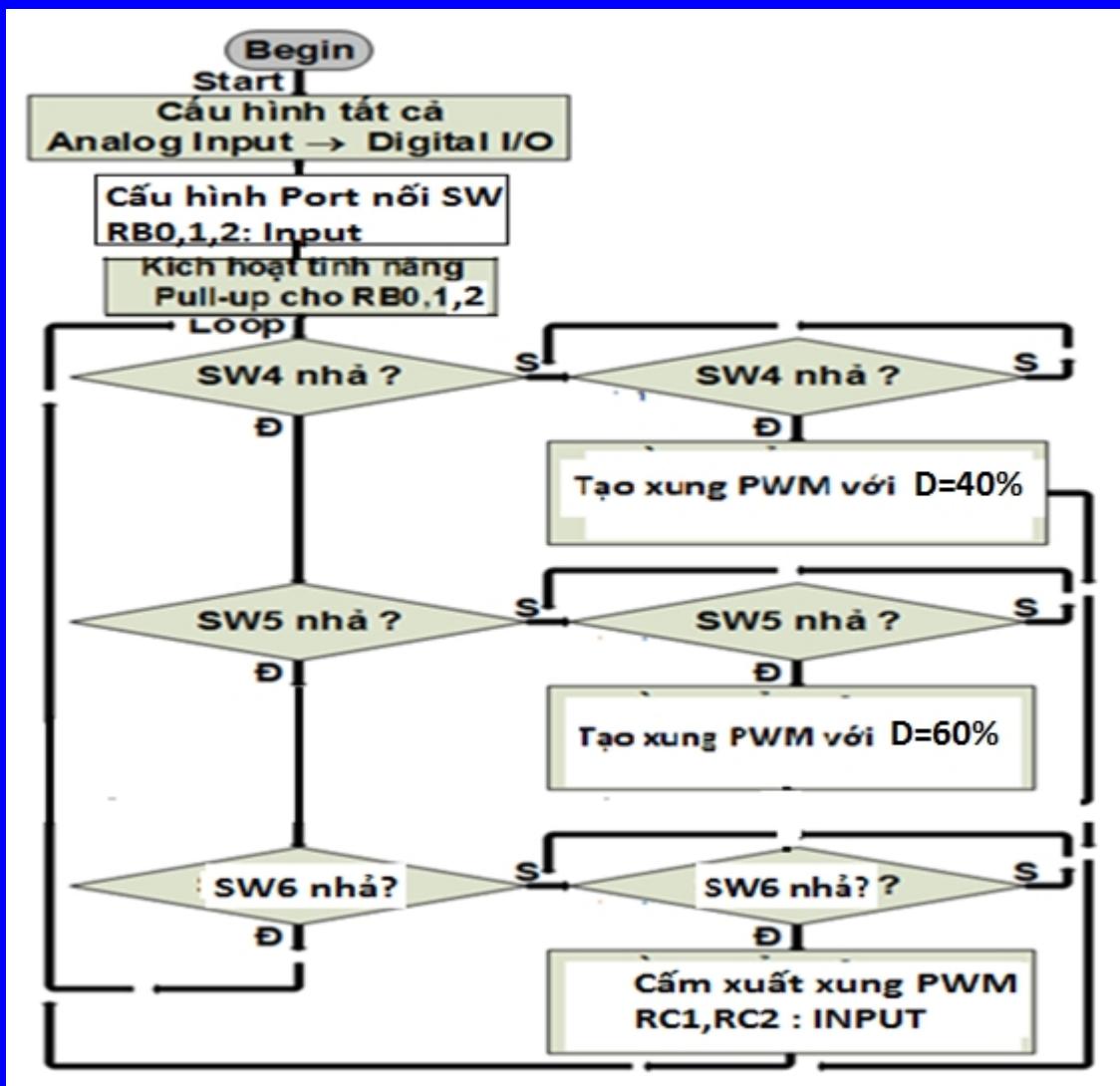


b) Vẽ sơ đồ mạch nguyên lý.



- c) Tính toán các giá trị thanh ghi sử dụng chức năng PWM của khối CCP.
- PR2=?
- Với CCP1: D%=40% >CCPR1L, CCP1CON=?
- Với CCP2: D%=60% >CCPR2L, CCP2CON=?

d) Vẽ lưu đồ giải thuật.



```
#include <xc.h>
#define _XTAL_FREQ 4000000
__CONFIG(.....);
void main (void)
{
    ANSEL=ANSEL=0;
    TRISB =0X07 ; nRBPU = 0;  WPUB = 0x07;
    while(1)
    {
        if (!RB0 )
        {
            while(!RB0);
            TRISC2 = 1;TRISC1=1;
            TMR2 = 0;
            PR2 = 199;
            CCPR1L =80; CCP1CON = 0X0C;
            TRISC2 = 0;T2CON = 0x00;TMR2ON = 1;}
```

```
else if (!RB1)
{
    while(!RB1);
        TRISC1 = 1; TRISC2=1;
        TMR2 = 0; PR2 =199;
        CCPR2L = 120;CCP2CON = 0X0C;
        TRISC1 = 0;
        T2CON = 0x00; TMR2ON = 1;
}
else if (!RB2 )
{
    while(!RB2);
        TRISC2 = 1;
        TRISC1 = 1;
}
}
```