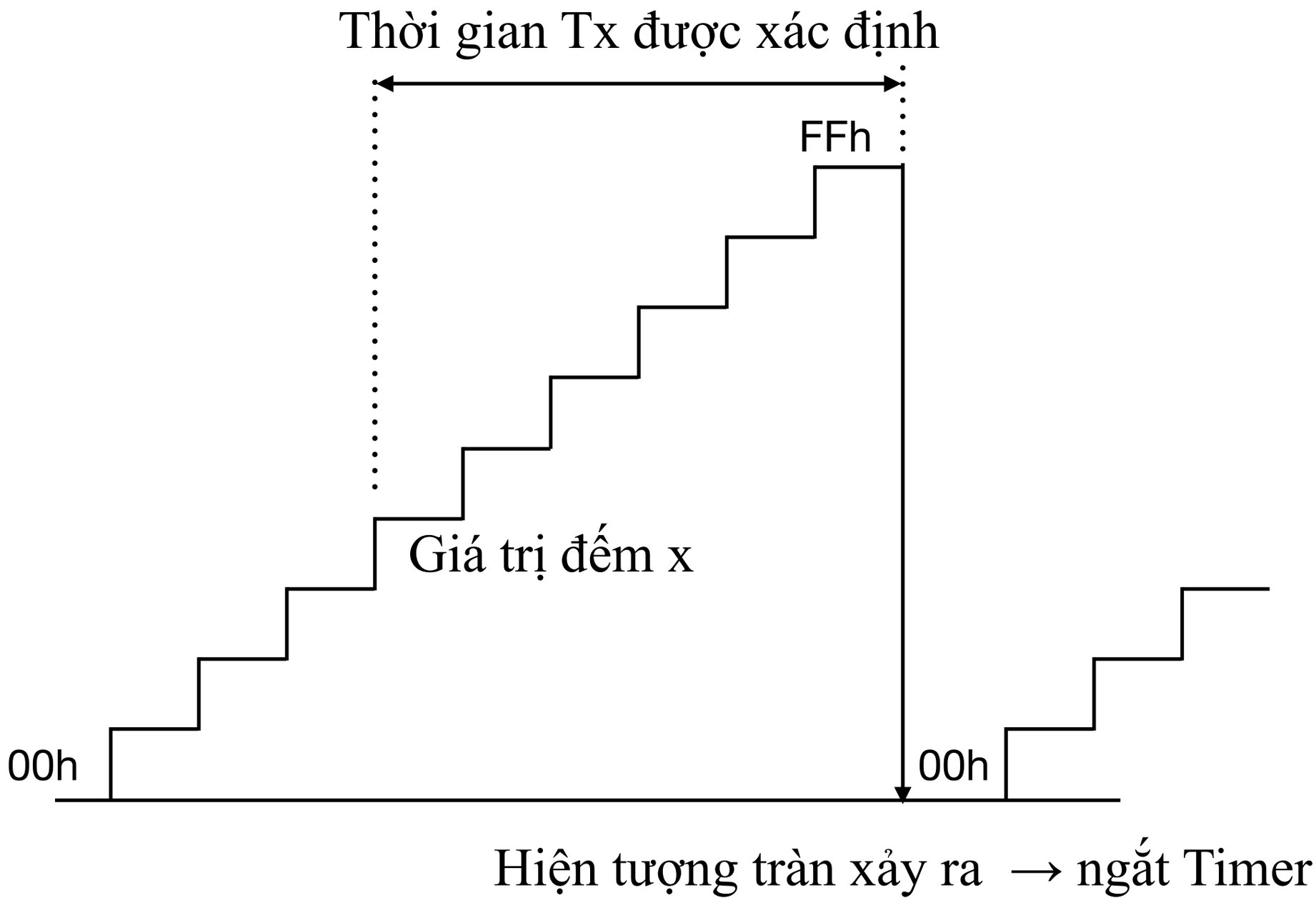
Phần Lý thuyết

1. Giới thiệu chung - Mô hình hệ VXL - Nguyên tắc hoạt động
2. Cấu trúc và hoạt động của vi xử lý 8085
3. Quá trình thực hiện 1 lệnh trong VXL 8085
4. Giới thiệu về vi điều khiển PIC
5. Bộ công cụ nạp chương trình, công cụ mô phỏng vi điều khiển
6. Bộ định thời Timer
7. Ghép nối với bộ hiển thị
8. ADC
9. Giao tiếp truyền dữ liệu
10. Ngắt
11. PWM

Timer trong PIC16F877A

- Có 3 timer được đánh số Timer0, Timer1 và Timer2
- là timer/counter 8 bit (Timer1 là 16 bit)
- có thể đọc và ghi
- có bộ chia trước 8 bit có thể lập trình bằng phần mềm
- Cho phép lựa chọn xung nguồn clock bên trong hoặc bên ngoài
- Phát sinh ngắt khi bị tràn từ FFh xuống 00h
- Cho phép lựa chọn các tác động theo sườn xung clock bên ngoài

Hoạt động của Timer

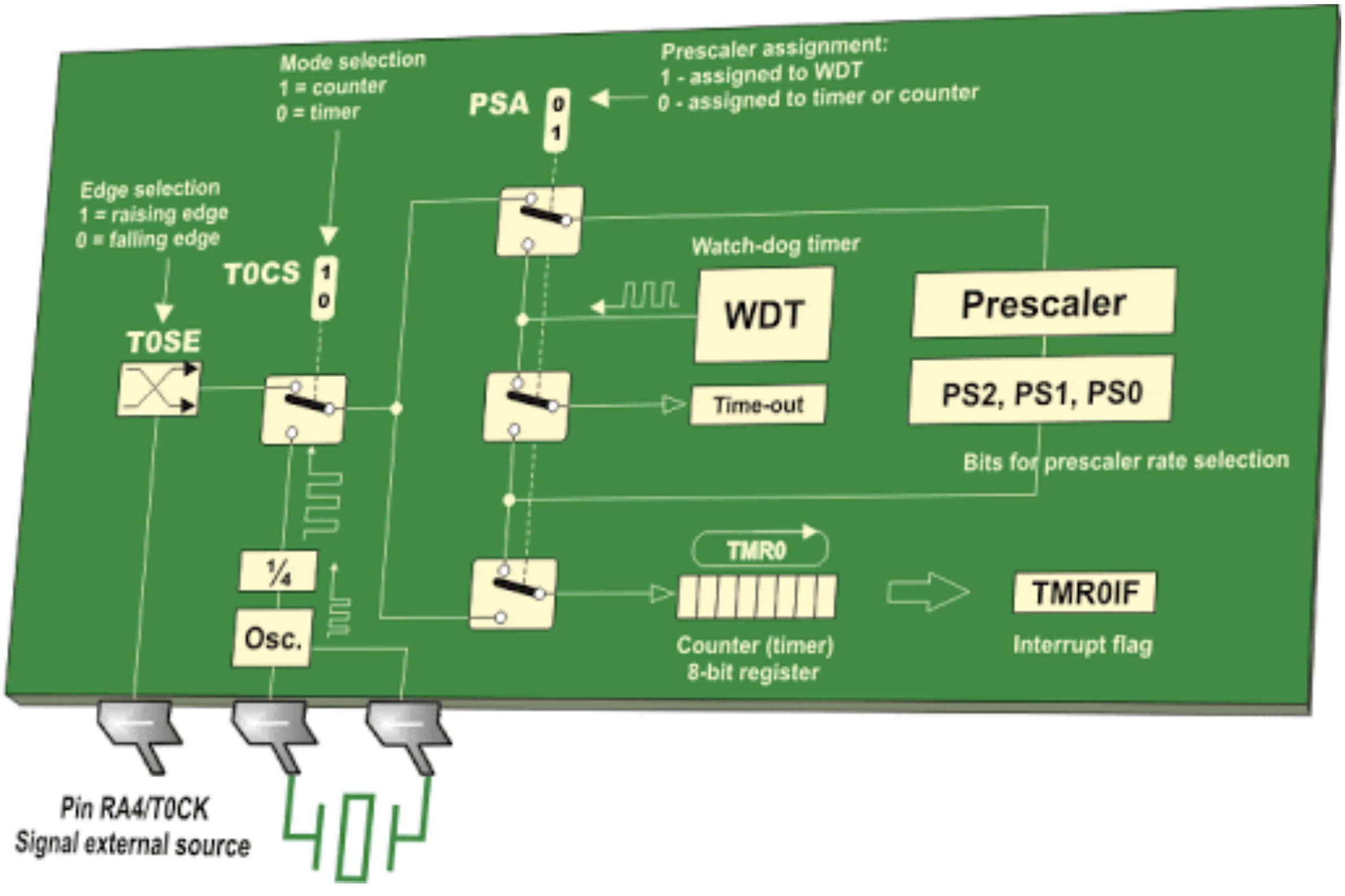


Hiện tượng tràn xảy ra → ngắt Timer

Câu hỏi đặt ra

- Giá trị đếm nằm ở đâu ?
- Khi nào thì giá trị đếm tăng ?
- Thời gian Tx được xác định như thế nào ?
- Khi tràn thì điều gì xảy ra ?

Timer0


Mode selection
1 = counter
0 = timer
Prescaler assignment:
1 - assigned to WDT
0 - assigned to timer or counter
PSA
0
1
Edge selection
1 = raising edge
0 = falling edge
T0CS
1
0
T0SE
Watch-dog timer
WDT
Prescaler
Time-out
PS2, PS1, PS0
Bits for prescaler rate selection
TMR0
1/4
Osc.
Counter (timer)
8-bit register
TMR0IF
Interrupt flag
Pin RA4/T0CK
Signal external source

Timer0

- Giá trị đếm nằm ở đâu ? ⟶ Thanh ghi TMR0
- Khi nào thì giá trị đếm tăng ?

Hoạt động chế độ Timer

- Clear bit TOSC tức là bit thứ 5 của POTION_REG)
- TMR0 sẽ tăng ứng với mỗi chu kỳ máy (tức là gấp 4 lần chu kỳ dao động)

Hoạt động chế độ Counter

- Set bit TOSC tức là bit thứ 5 của OPTION_REG)
- TMR0 sẽ tăng ứng với mỗi xung tác động vào chân RA4/TOCK1

  TOSE=0 sườn lên
  TOSE=1 sườn xuống

TOSE là bit 4 của OPTION_REG

Timer0

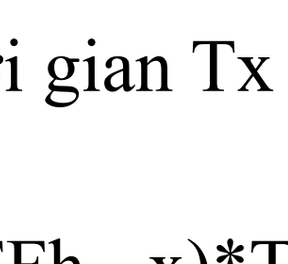
- Thời gian Tx được xác định như thế nào ?

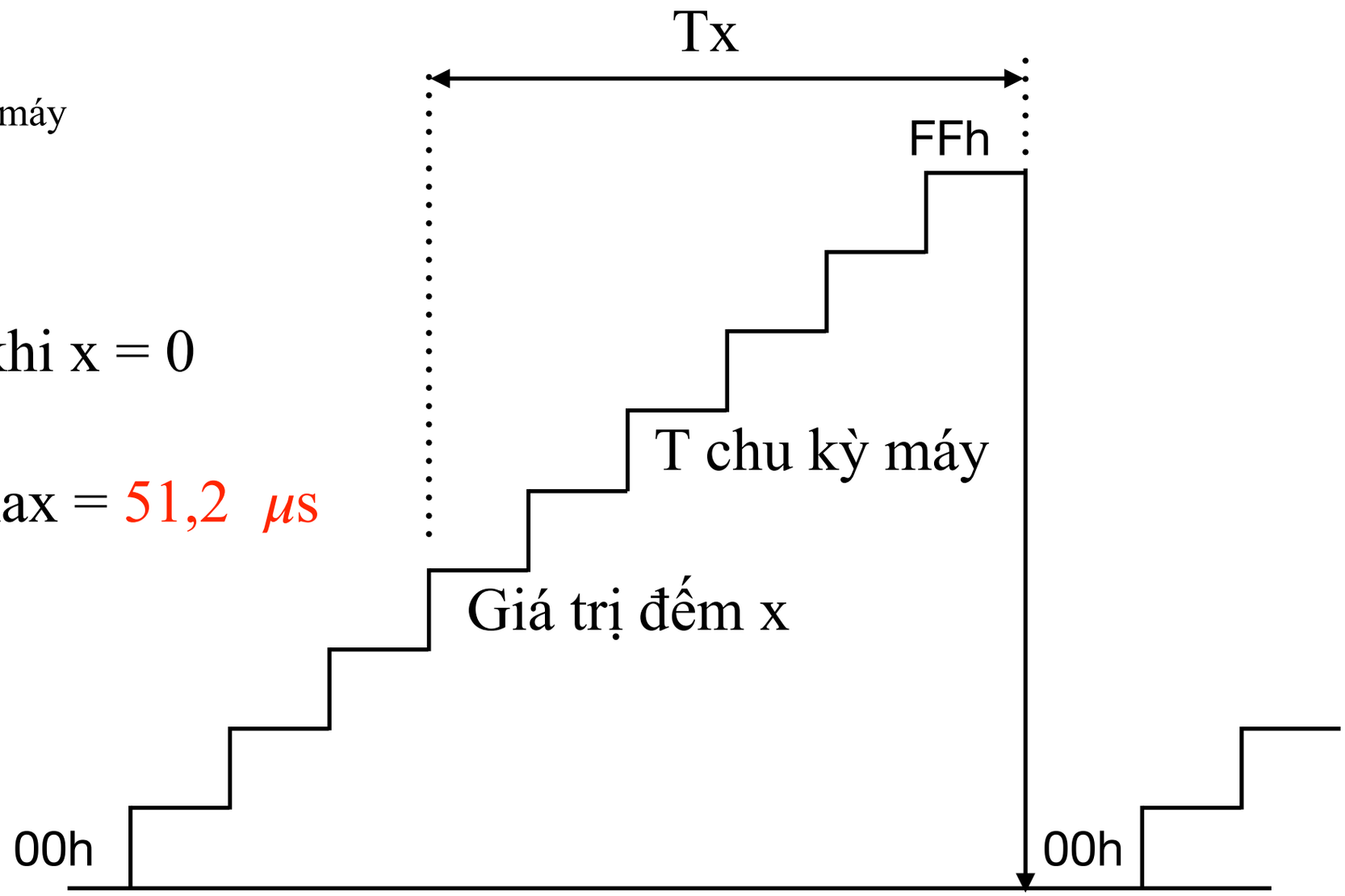
$Tx = (FFh - x)*T_{chu\ kỳ\ máy}$

$T_{chu\ kỳ\ máy} = 4* T_{osc}$

Giá trị lớn nhất của Tx là khi x = 0

Nếu f_{osc} = 20Mhz thì Tx max = 51,2 μs





Làm thế nào để tăng thời gian Tx lên ?

Timer0

Sử dụng bộ chia trước prescale

Khi gán bộ chia trước prescale cho Timer0 thì không thể sử dụng nó cho WDT và ngược lại

OPTION_REG REGISTER (ADDRESS 81h, 181h)

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

3 bit xác định prescale

PS2:PS0: 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

$Tx = Prescale*(FFh - x)*T_{chu\ kỳ\ máy}$

PSA: Prescaler Assignment bit

1 = Prescaler is assigned to the WDT

0 = Prescaler is assigned to the Timer0 module

Lặp đi lặp lại hiện tượng tràn Timer cho đến khi cần

Sử dụng timer có khả năng đếm lớn hơn

Timer0

- Khi tràn thì điều gì xảy ra ?

  Bit TMR0IF (là bit số 2 của thanh ghi INTCON) sẽ được set lên 1

  Bit này phải được xoá bằng phần mềm

INTCON REGISTER (ADDRESS 0Bh, 8Bh, 10Bh, 18Bh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF
bit 7							bit 0

Bit TMR0IE (là bit số 5 của thanh ghi INTCON) là bit ngắt/cấm ngắt Timer 0

TMR0IE =0. Tắt ngắt Timer0
TMR0IE=1. Cho phép ngắt Timer0

Những thanh ghi liên quan Timer0

Address	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
01h,101h	TMR0	Timer0 Module Register								xxxx xxxx	uuuu uuuu
0Bh,8Bh, 10Bh,18Bh	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	0000 000u
81h,181h	OPTION_REG	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111

Legend: x = unknown, u = unchanged, - = unimplemented locations read as '0'. Shaded cells are not used by Timer0.

GIE: Global Interrupt Enable bit

1 = Enables all unmasked interrupts

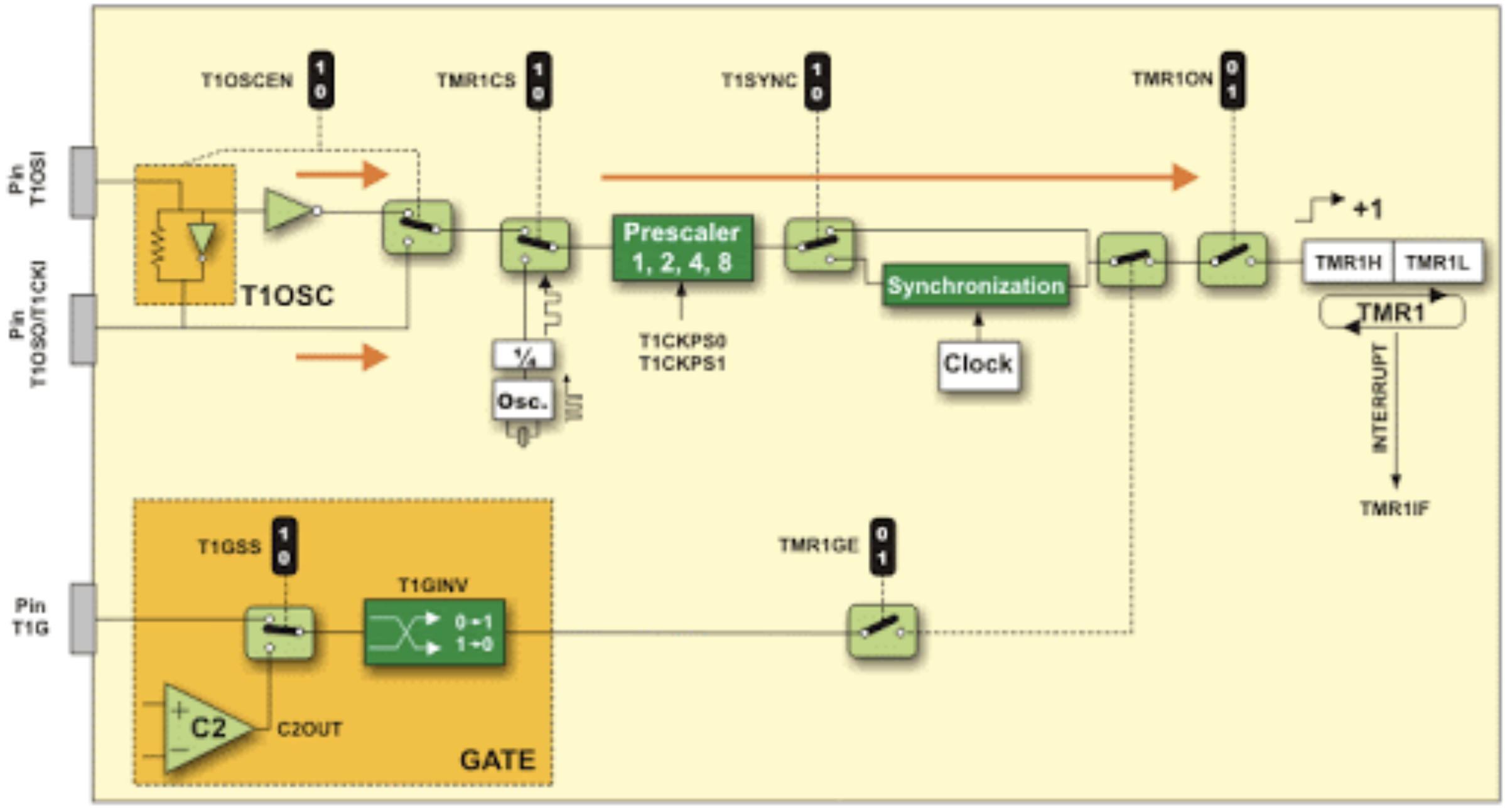
0 = Disables all interrupts

PEIE: Peripheral Interrupt Enable bit

1 = Enables all unmasked peripheral interrupts

0 = Disables all peripheral interrupts

Timer1


T1OSCEN
TMR1CS
T1SYNC
TMR1ON
Pin T1OSI
Pin T1OSO/T1CKI
T1OSC
Prescaler 1, 2, 4, 8
T1CKPS0
T1CKPS1
Synchronization
Clock
1/4
Osc.
+1
TMR1H
TMR1L
TMR1
INTERRUPT
TMR1IF
T1GSS
T1GINV
0→1
1→0
Pin T1G
TMR1GE
C2
C2OUT
GATE

Timer1

- Timer0 vs Timer1

	Timer0	Timer1
Độ lớn	8 bit	16 bit
Nơi lưu giá trị đếm	TMR0	TMR1H và TMR1L
Cờ ngắt	TMR0IF	TMR1IF là bit 0 của PIR1
Bit cấm ngắt	TMR0IE	TMR1IE là bit 0 của PIE1
Tính thời gian	$Tx = Prescale*(FFh - x)*T_{chu\ kỳ\ máy}$ $T_{chu\ kỳ\ máy} = 4* T_{osc}$	$Tx = Prescale*(FFFFh - x)*T_{chu\ kỳ\ máy}$ $T_{chu\ kỳ\ máy} = 4* T_{osc}$

Timer1

- Thanh ghi điều khiển Timer1

T1CON: TIMER1 CONTROL REGISTER (ADDRESS 10h)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON
bit 7							bit 0

- Chế độ Timer

TMR1CS = 0

Giá trị 16 bit trong 2 thanh ghi TMR1H và TMR1L tăng theo mỗi chu kỳ lệnh

Quá trình đếm có thể dừng lại bởi bit TMR1ON

TMR1ON=1. Cho phép đếm
TMR1ON=0. Ngừng đếm

Timer1

- Thanh ghi điều khiển Timer1

T1CON: TIMER1 CONTROL REGISTER (ADDRESS 10h)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON
bit 7							bit 0

- Chế độ Counter

TMR1CS = 1

Giá trị 16 bit trong 2 thanh ghi TMR1H và TMR1L tăng theo sườn lên của tín hiệu tác động vào chân RC0/T1OSO/T1CKI

Ở chế độ này, Counter phải nhận 1 xung xuống trước khi bắt đầu đếm

Quá trình đếm có thể dừng lại bởi bit TMR1ON

TMR1ON=1. Cho phép đếm
TMR1ON=0. Ngừng đếm

Timer1

- Thanh ghi điều khiển Timer1

T1CON: TIMER1 CONTROL REGISTER (ADDRESS 10h)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON
bit 7							bit 0

- Bộ đếm đồng bộ

  TMR1CS = 1

  $\overline{\text{T1SYNC}}$ = 0

  Bộ đếm tăng khi gặp sườn lên ở

  RC1/T1OSI/CCP2 khi T1OSCEN=1

  hoặc RC0/T1OSO/T1CKI khi T1OSCEN=0

  Bộ đếm không tăng ở chế độ sleep

- Bộ đếm không đồng bộ

  TMR1CS = 1

  $\overline{\text{T1SYNC}}$ = 1

  Bộ đếm tăng khi gặp sườn lên ở

  RC0/T1OSO/T1CKI

  Bộ đếm tăng ở chế độ sleep

Bộ dao động công suất thấp Timer1

- Thiết kế và tích hợp bên trong giữa các chân T1OSI (input) và T1OSO (putput)
- Bộ dao động hoạt động khi bit T1OSEN =1 (bit thứ 3 của T1CON)
- Công suất thấp với tần số max là 200kHz
- Hoạt động ngay cả trong chế độ sleep

Osc Type	Freq.	C1	C2
LP	32 kHz	33 pF	33 pF
	100 kHz	15 pF	15 pF
	200 kHz	15 pF	15 pF

Bộ chia trước của Timer1

- Thanh ghi điều khiển Timer1

T1CON: TIMER1 CONTROL REGISTER (ADDRESS 10h)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON
bit 7							bit 0

T1CKPS1:T1CKPS0: 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

Các thanh ghi tác động tới Timer1

Address	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
0Bh,8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
0Eh	TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
0Fh	TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON	--00 0000	--uu uuuu

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by the Timer1 module.

Note 1: Bits PSPIE and PSPIF are reserved on the 28-pin devices; always maintain these bits clear.

GIE: Global Interrupt Enable bit

1 = Enables all unmasked interrupts

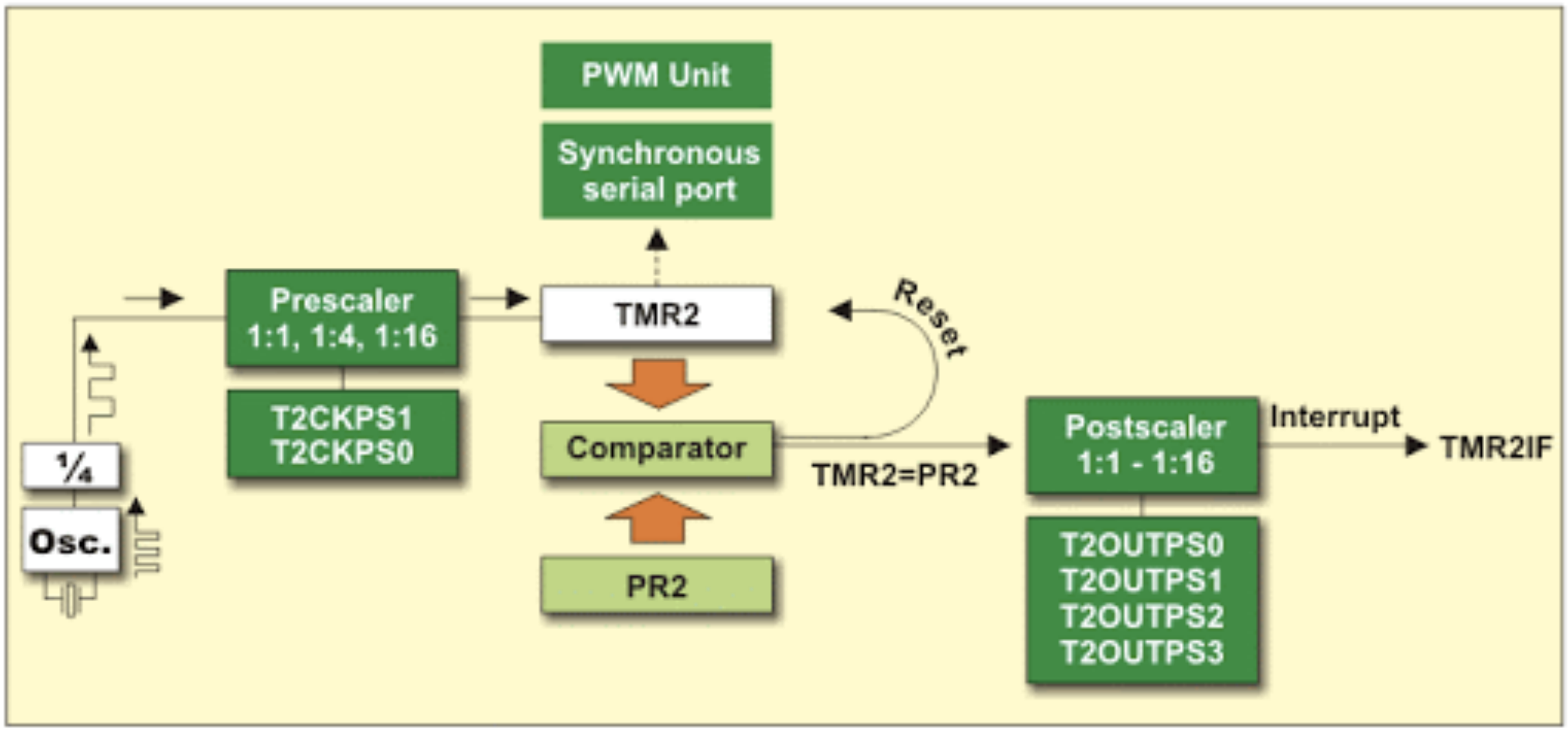
0 = Disables all interrupts

PEIE: Peripheral Interrupt Enable bit

1 = Enables all unmasked peripheral interrupts

0 = Disables all peripheral interrupts

Timer2


PWM Unit
Synchronous serial port
Prescaler 1:1, 1:4, 1:16
T2CKPS1 T2CKPS0
TMR2
Reset
Comparator
PR2
TMR2=PR2
Postscaler 1:1 - 1:16
T2OUTPS0 T2OUTPS1 T2OUTPS2 T2OUTPS3
Interrupt
TMR2IF
¼
Osc.

Timer2

- Timer0, Timer1 vs Timer2

	Timer0	Timer1	Timer2
Độ lớn	8 bit	16 bit	8 bit
Nơi lưu giá trị đếm	TMR0	TMR1H và TMR1L	TMR2
Cờ ngắt	TMR0IF	TMR1IF	TMR2IF là bit 1 của PIR1
Bit cấm ngắt	TMR0IE	TMR1IE	TMR2IE là bit 2 của PIE1
Tính thời gian	$Tx = Prescale*(FFh - x)*T_{chu\ kỳ\ máy}$ $T_{chu\ kỳ\ máy} = 4* T_{osc}$	$Tx = Prescale*(FFFFh - x)*T_{chu\ kỳ\ máy}$ $T_{chu\ kỳ\ máy} = 4* T_{osc}$	$Tx = Prescale*(PR2 - x)*postscale*T_{chu\ kỳ\ máy}$ $T_{chu\ kỳ\ máy} = 4* T_{osc}$

Timer2

- Thanh ghi điều khiển Timer2

T2CON: TIMER2 CONTROL REGISTER (ADDRESS 12h)

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

TOUTPS3:TOUTPS0: Timer2 Output Postscale Select bits
0000 = 1:1 postscale
0001 = 1:2 postscale
0010 = 1:3 postscale
•
•
•
1111 = 1:16 postscale

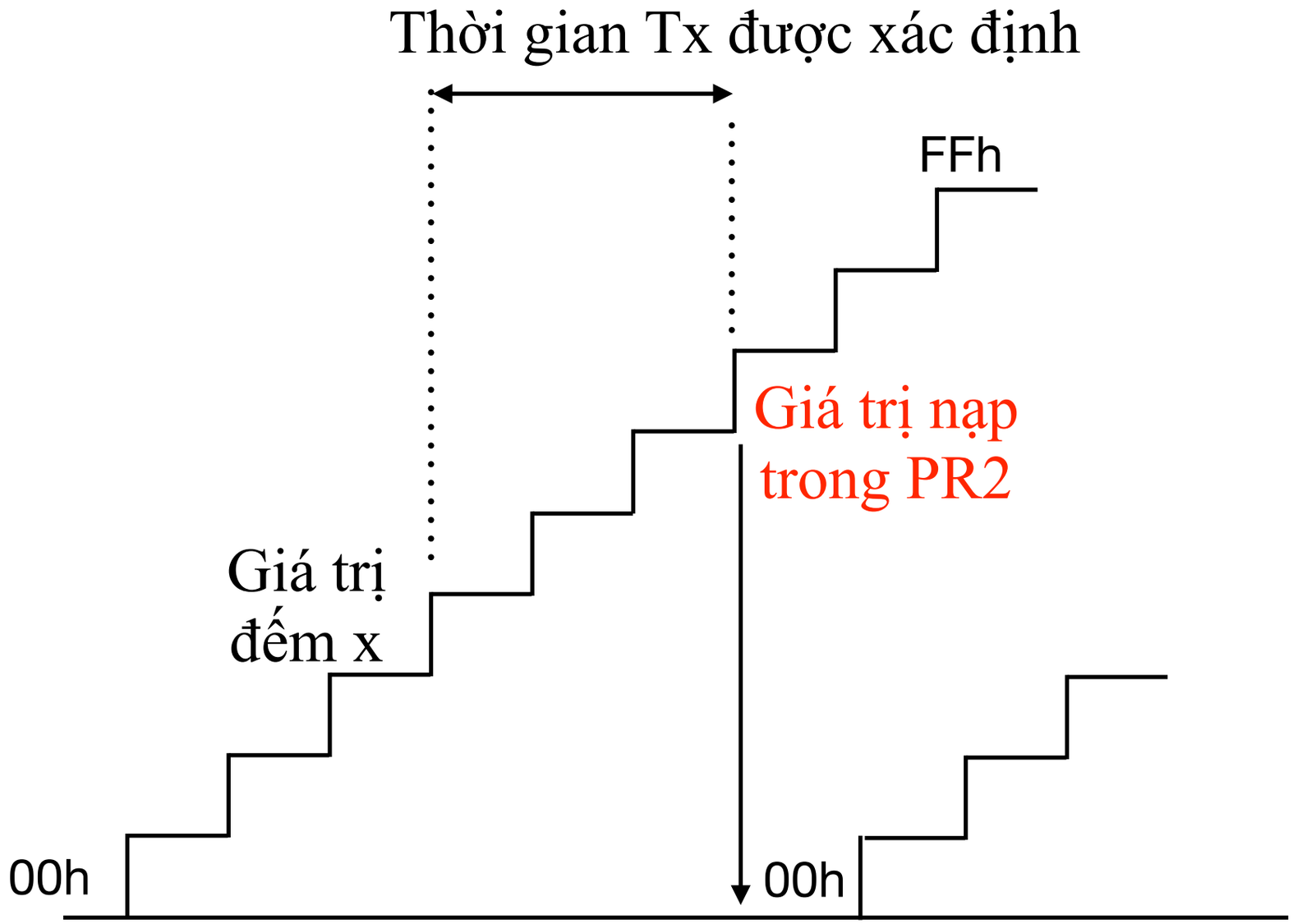
T2CKPS1:T2CKPS0: Timer2 Clock Prescale Select bits
00 = Prescaler is 1
01 = Prescaler is 4
1x = Prescaler is 16

TMR2ON được sử dụng để tắt/bật Timer2 nhằm giảm công suất tiêu thụ

TMR2ON =1 Timer2 ON
TMR2ON =0 Timer2 OFF

Timer2 khác Timer0 và Timer1 ở PR2

- Thanh ghi điều khiển Timer2



Hiện tượng tràn xảy ra khi giá trị trong TMR2 bằng PR2

Các thanh ghi tác động tới Timer2

Address	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
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
11h	TMR2	Timer2 Module's Register								0000 0000	0000 0000
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
92h	PR2	Timer2 Period Register								1111 1111	1111 1111

GIE: Global Interrupt Enable bit

1 = Enables all unmasked interrupts

0 = Disables all interrupts

PEIE: Peripheral Interrupt Enable bit

1 = Enables all unmasked peripheral interrupts

0 = Disables all peripheral interrupts

Sử dụng Timer0 trong CCS

setup_timer_0(mode) được định nghĩa trong 16F877A.h

mode là các phần tử sau, có thể kết hợp bằng dấu “|”

RTCC_INTERNAL	chọn xung clock nội	RTCC_DIV_16	Prescale 1:16
RTCC_EXT_L_TO_H	Sườn lên trên chân RA4	RTCC_DIV_32	Prescale 1:32
RTCC_EXT_H_TO_L	Sườn xuống trên chân RA4	RTCC_DIV_64	Prescale 1:64
RTCC_DIV_2	Prescale 1:2	RTCC_DIV_128	Prescale 1:128
RTCC_DIV_4	Prescale 1:4	RTCC_DIV_256	Prescale 1:256
RTCC_DIV_8	Prescale 1:8		

set_timer0(value) Nạp giá trị 8 bit ban đầu cho Timer0

get_timer0() Trả về giá trị 8 bit hiện có trong Timer0

Sử dụng Timer0 trong CCS

setup_counters(rtcc_state,ps_state) được định nghĩa trong 16F877A.h

rtcc_state là một trong những constant sau

RTCC_INTERNAL	chọn xung clock nội
RTCC_EXT_L_TO_H	Sườn lên trên chân RA4
RTCC_EXT_H_TO_L	Sườn xuống trên chân RA4

ps_state là một trong những constant sau

RTCC_DIV_2	Prescale 1:2	WDT_18ms	
RTCC_DIV_4	Prescale 1:4	WDT_36ms	
RTCC_DIV_8	Prescale 1:8	WDT_72ms	
RTCC_DIV_16	Prescale 1:16	WDT_144ms	
RTCC_DIV_32	Prescale 1:32	WDT_288ms	
RTCC_DIV_64	Prescale 1:64	WDT_576ms	
RTCC_DIV_128	Prescale 1:128	WDT_1152ms	
RTCC_DIV_256	Prescale 1:256	WDT_2304ms	

Sử dụng Timer1 trong CCS

setup_timer_1(mode) được định nghĩa trong 16F877A.h

mode là các phần tử sau, có thể kết hợp bằng dấu “|”

T1_DISABLED	tắt Timer1	T1_DIV_BY_1	Prescale 1:1
T1_INTERNAL	Xung clock nội (fosc/4)	T1_DIV_BY_2	Prescale 1:2
T1_EXTERNAL	Xung clock ngoài trên chân RC0	T1_DIV_BY_4	Prescale 1:4
T1_EXTERNAL_SYNC	Xung clock ngoài đồng bộ	T1_DIV_BY_8	Prescale 1:8
T1_CLK_OUT	Tạo xung dao động timer1		

set_timer1(value) Nạp giá trị 16 bit ban đầu cho Timer1

get_timer1() Trả về giá trị 16 bit hiện có trong Timer1

Sử dụng Timer2 trong CCS

setup_timer_2(mode, period, postscale)

mode là các phần tử sau, có thể kết hợp bằng dấu “|”

T2_DISABLED	tắt Timer2	T2_DIV_BY_4	Prescale 1:4
T2_DIV_BY_1	Prescale 1:1	T2_DIV_BY_16	Prescale 1:16

period: số nguyên từ 0~255, xác định giá trị xung reset

postscale: số nguyên 1~16, xác định reset bao nhiêu lần trước khi ngắt

set_timer2(value) Nạp giá trị 8 bit ban đầu cho Timer2

get_timer2() Trả về giá trị 8 bit hiện có trong Timer2

Chương trình CCS sử dụng Timer

```
enable_interrupts(int_timerX);
enable_interrupts(global);
```

X là chỉ số Timer. Khi tràn (với timer 0,1) hoặc sau một số lần tràn(timer2) thì chương trình nhảy tới hàm ngắt

```
#int_timerX
```

Khai báo chương trình ngắt timerX

```
Void Ngat_timerX(void)
{
```

Chương trình ngắt

```
}
```

Chương trình CCS sử dụng Timer

```
#include <16F877a.h>

#use Delay(Clock=4000000)

#define led RC0

#int_TIMER0
void TIMER0_isr(void)
{
set_timer0(206);
led=!led;
}

void main()
{
set_tris_c(0x00);
setup_timer_0(RTCC_INTERNAL|RTCC_DIV_4);//0-255 tran 51.2us
set_timer0(206);
enable_interrupts(INT_TIMER0);
enable_interrupts(GlOBAL);

while(true)
{

}
}
```

Sinh viên tính toán thời gian ngắt là bao nhiêu ?