TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI
KHOA ĐIỆN TỬ

Tài liệu

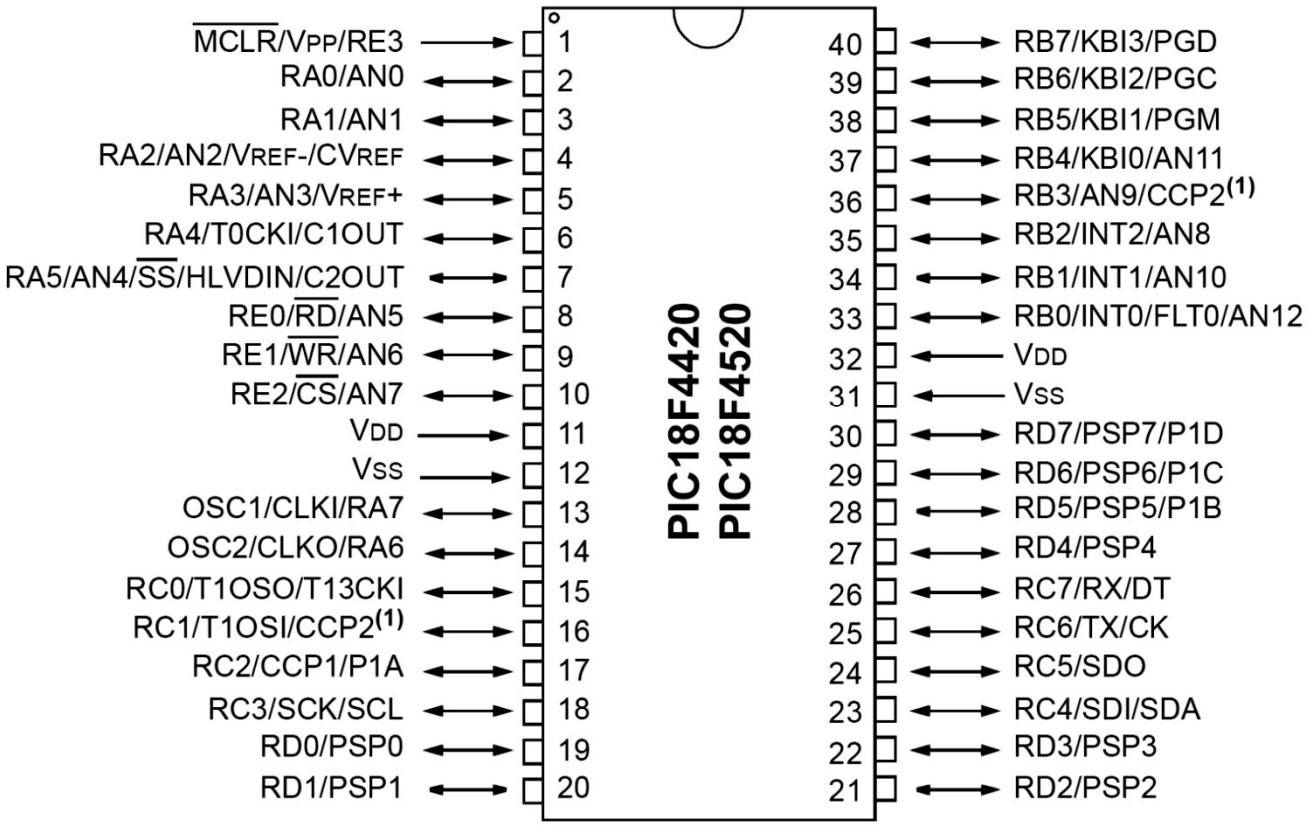
VI ĐIỀU KHIỂN PIC

(Dùng cho HS-SV tra cứu trong trong quá trình thực hành, thi, kiểm tra môn học Vi điều khiển)

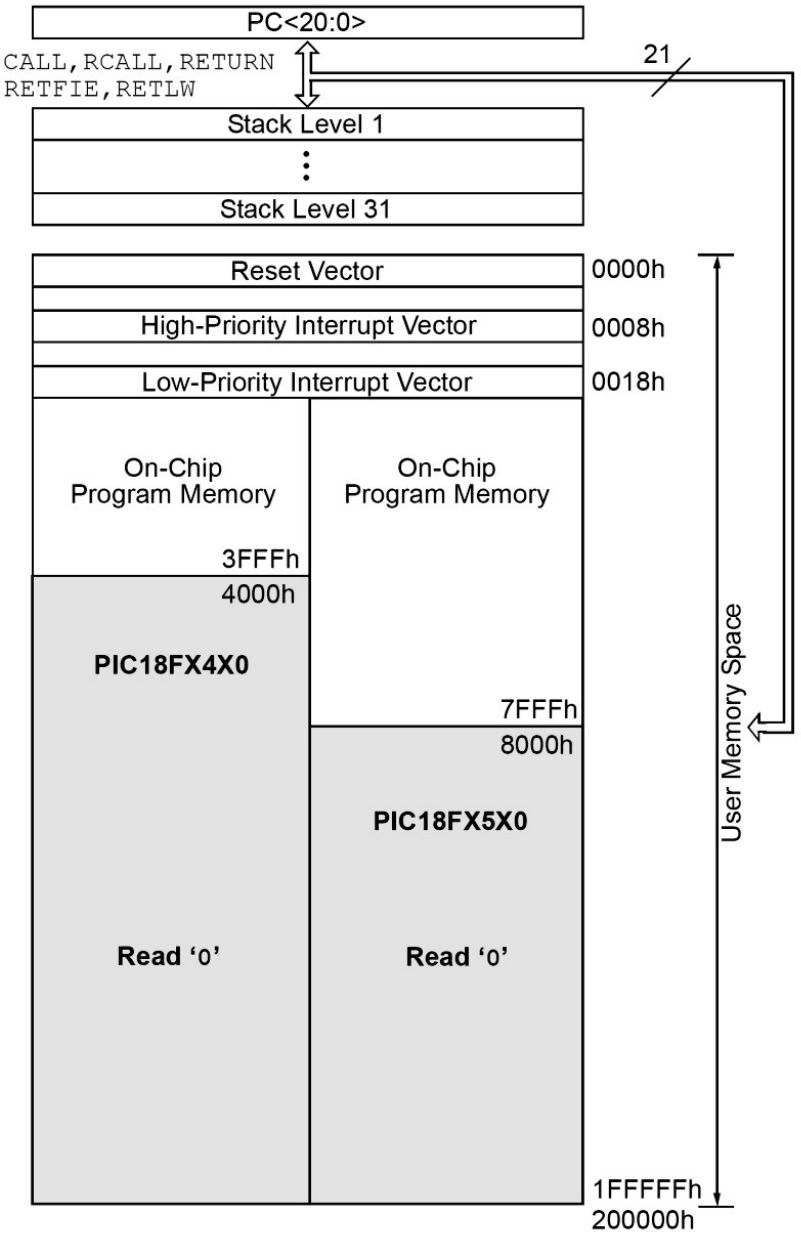
Mục lục

1. PHẦN CỨNG CỦA PIC 18F4520

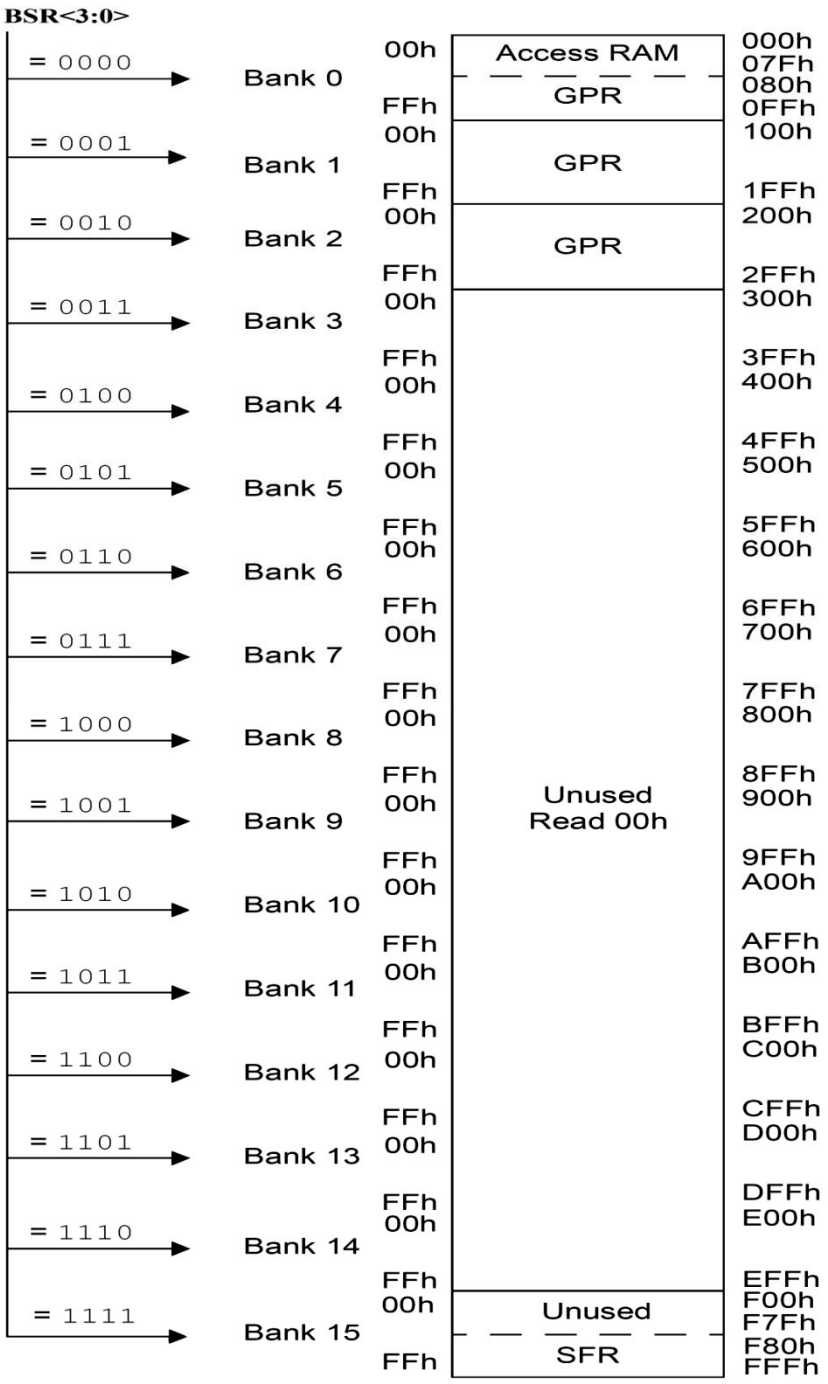
1.1. Sơ đồ chân



1.2. Tổ chức bộ nhớ



Sơ đồ tổ chức bộ nhớ chương trình và ngăn xếp



Sơ đồ tổ chức bộ nhớ dữ liệu RAM

Address	Name	Address	Name	Address	Name	Address	Name
FFFh	TOSU	FDFh	INDF2(1)	FBFh	CCPR1H	F9Fh	IPR1
FFEh	TOSH	FDEh	POSTINC2(1)	FBEh	CCPR1L	F9Eh	PIR1
FFDh	TOSL	FDDh	POSTDEC2(1)	FBDh	CCP1CON	F9Dh	PIE1
FFCh	STKPTR	FDCh	PREINC2(1)	FBCh	CCPR2H	F9Ch	—(2)
FFBh	PCLATU	FDBh	PLUSW2(1)	FBBh	CCPR2L	F9Bh	OSCTUNE
FFAh	PCLATH	FDAh	FSR2H	FBAh	CCP2CON	F9Ah	—(2)
FF9h	PCL	FD9h	FSR2L	FB9h	—(2)	F99h	—(2)
FF8h	TBLPTRU	FD8h	STATUS	FB8h	BAUDCON	F98h	—(2)
FF7h	TBLPTRH	FD7h	TMR0H	FB7h	PWM1CON(3)	F97h	—(2)
FF6h	TBLPTRL	FD6h	TMR0L	FB6h	ECCP1AS(3)	F96h	TRISE(3)
FF5h	TABLAT	FD5h	T0CON	FB5h	CVRCON	F95h	TRISD(3)
FF4h	PRODH	FD4h	—(2)	FB4h	CMCON	F94h	TRISC
FF3h	PRODL	FD3h	OSCCON	FB3h	TMR3H	F93h	TRISB
FF2h	INTCON	FD2h	HLVDCON	FB2h	TMR3L	F92h	TRISA
FF1h	INTCON2	FD1h	WDTCON	FB1h	T3CON	F91h	—(2)
FF0h	INTCON3	FD0h	RCON	FB0h	SPBRGH	F90h	—(2)
FEFh	INDF0(1)	FCFh	TMR1H	FAFh	SPBRG	F8Fh	—(2)
FEEh	POSTINC0(1)	FCEh	TMR1L	FAEh	RCREG	F8Eh	—(2)
FEDh	POSTDEC0(1)	FCDh	T1CON	FADh	TXREG	F8Dh	LATE(3)
FECh	PREINC0(1)	FCCh	TMR2	FACh	TXSTA	F8Ch	LATD(3)
FEBh	PLUSW0(1)	FCBh	PR2	FABh	RCSTA	F8Bh	LATC
FEAh	FSR0H	FCAh	T2CON	FAAh	—(2)	F8Ah	LATB
FE9h	FSR0L	FC9h	SSPBUF	FA9h	EEADR	F89h	LATA
FE8h	WREG	FC8h	SSPADD	FA8h	EEDATA	F88h	—(2)
FE7h	INDF1(1)	FC7h	SSPSTAT	FA7h	EECON2(1)	F87h	—(2)
FE6h	POSTINC1(1)	FC6h	SSPCON1	FA6h	EECON1	F86h	—(2)
FE5h	POSTDEC1(1)	FC5h	SSPCON2	FA5h	—(2)	F85h	—(2)
FE4h	PREINC1(1)	FC4h	ADRESH	FA4h	—(2)	F84h	PORTE(3)
FE3h	PLUSW1(1)	FC3h	ADRESL	FA3h	—(2)	F83h	PORTD(3)
FE2h	FSR1H	FC2h	ADCON0	FA2h	IPR2	F82h	PORTC
FE1h	FSR1L	FC1h	ADCON1	FA1h	PIR2	F81h	PORTB
FE0h	BSR	FC0h	ADCON2	FA0h	PIE2	F80h	PORTA

Phân bổ địa chỉ của các thanh ghi chức năng đặc biệt SFR

1.3. Các thanh ghi của EEPROM

- Thanh ghi điều khiển EEPROM 1 : EECON1

R/W-x	R/W-x	U-0	R/W-0	R/W-x	R/W-0	R/S-0	R/S-0
EEPGD	CFGS	—	FREE	WRERR(1)	WREN	WR	RD
bit 7							bit 0

Ghi chú: S = Bit chỉ được thiết lập (không xóa được bằng phần mềm)
R = Cho phép đọc W = Cho phép ghi U = Không sử dụng, đọc bằng ‘0’
-n = Reset - POR ‘1’ = Được thiết lập ‘0’ = Được xóa -x = Reset không xác định

bit 7 **EEPGD**: Bit lựa chọn bộ nhớ dữ liệu EEPROM hay bộ nhớ chương trình Flash

1 = Truy cập bộ nhớ chương trình Flash

0 = Truy cập bộ nhớ dữ liệu EEPROM

bit 6 **CFGS**: Bit lựa chọn cấu hình hoặc bộ nhớ chương trình Flash/dữ liệu EEPROM

1 = Truy cập thanh ghi cấu hình

0 = Truy cập bộ nhớ chương trình Flash hoặc dữ liệu EEPROM

bit 5 Không sử dụng, đọc trả về giá trị '0'

bit 4 **FREE**: Bit cho phép xóa hàng bộ nhớ Flash

1 = Xóa hàng bộ nhớ chương trình được thiết lập, địa chỉ chứa trong thanh ghi TBLPTR, được xóa từ lệnh WR kê tiếp.

0 = Chỉ thực hiện ghi

bit 3 **WRERR**: Bit cờ lỗi bộ nhớ chương trình Flash/ bộ nhớ dữ liệu EEPROM

1 = Lỗi hoạt động ghi (hoạt động ghi bị kết thúc trước)

0 = Ghi hoàn thành

bit 2 **WREN**: Bit cho phép ghi bộ nhớ chương trình Flash / Dữ liệu EEPROM

1 = Cho phép ghi vào bộ nhớ chương trình Flash /dữ liệu EEPROM

0 = Không cho phép ghi

bit 1 **WR**: Bit điều khiển ghi

1 = Khởi tạo quá trình xóa/ghi bộ nhớ dữ liệu EEPROM hoặc xóa bộ nhớ chương trình hoặc ghi bộ nhớ chương trình. (Được xóa bằng phần cứng khi việc ghi hoàn thành. Thiết lập được bằng phần mềm nhưng không được xóa).

0 = Quá trình ghi hoàn thanh.

bit 0 **RD**: Bit điều khiển đọc

1 = Khởi tạo quá trình đọc bộ nhớ EEPROM (Đọc mất một chu kỳ máy. Bit RD được xóa bằng phần cứng. Thiết lập được bằng phần mềm nhưng không được xóa. Bit RD không được thiết lập khi EEPGD = 1 hoặc CFGS = 1.)

0 = Không khởi tạo quá trình đọc EEPROM

- Thanh ghi điều khiển EEPROM 2 : EECON2

Thanh ghi EECON2 không phải là thanh vật lý, nó được dành riêng cho việc ghi và xóa bộ nhớ. Đọc EECON2 sẽ được '0'.

- Thanh ghi dữ liệu EEPROM: EEDATA

Thanh ghi EEDATA có 8 bit, là thanh ghi đệm dữ liệu cho bộ nhớ dữ liệu EEPROM, được sử dụng để truy cập vào dữ liệu của bộ nhớ (Cho phép đọc ghi bằng phần mềm, mỗi ô nhớ của EEPROM có 8 bit).

- Thanh ghi địa chỉ EEPROM: EEADR

Thanh ghi EEADR có 8 bit, là thanh ghi địa chỉ của bộ nhớ dữ liệu EEPROM. 8 bit của thanh ghi EEADR được sử dụng để địa chỉ hóa 256 ô nhớ của EEPROM từ 00h đến FFh.

1.4. Các thanh ghi của bộ phát xung

- Thanh ghi chuyển chế độ bộ phát xung : OSCTUNE

R/W-0	R/W-0(1)	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INTSRC	PLLEN(1)	—	TUN4	TUN3	TUN2	TUN1	TUN0
bit 7							bit 0

Ghi chú:
R = Cho phép đọc W = Cho phép ghi U = Không sử dụng, đọc bằng '0'
-n = Reset - POR '1' = Được thiết lập '0' = Được xóa -x = Reset không xác định

bit 7 **INTSRC**: Bit lựa chọn nguồn xung nội tần số thấp

1 = Chọn tần số 31.25 kHz từ bộ chia tần Postscaler (8 MHz INTOSC chia 256)

0 = Chọn tần số 31 kHz từ bộ dao động nội INTRC

bit 6 **PLLEN**: Bit lựa chọn bộ nhân PLL cho chế độ INTOSC

1 = Cho phép xung từ INTOSC qua bộ nhân tần số PLL (chỉ sử dụng với tần số 4 MHz và 8 MHz)

0 = Không cho phép PLL

bit 5 **Không sử dụng**: Đọc được '0'

bit 4-0 **TUN4:TUN0**: Bit chuyển chế độ tần số

01111 = Tần số Max

...

00001

00000 = Tần số trung bình. Bộ phát xung hoạt động ở tần số đã hiệu chuẩn.

11111

...

10000 = Tần số Min

- Thanh ghi điều khiển bộ phát xung OSCCON

R/W-0	R/W-1	R/W-0	R/W-0	R(1)	R-0	R/W-0	R/W-0
IDLEN	IRCF2	IRCF1	IRCF0	OSTS	IOFS	SCS1	SCS0
bit 7							bit 0

bit 7 **IDLEN**: Bit cho phép chế độ **Idle**

1 = Chuyển sang chế độ Idle bằng lệnh SLEEP

0 = Chuyển sang chế độ Sleep bằng lệnh SLEEP

bit 6-4 **IRCF2**:IRCF0: Các bit lựa chọn hệ số chia bộ phát cung nội INTOSC

111 = 8 MHz (xung trực tiếp từ INTOSC)

110 = 4 MHz

101 = 2 MHz

100 = 1 MHz (tần số mặc định khi Reset)

011 = 500 kHz

010 = 250 kHz

001 = 125 kHz

000 = 31 kHz (xung từ INTOSC/256 hoặc trực tiếp từ INTRC)

bit 3 **OSTS**: Bit trạng thái bộ bộ định thời khởi động (Oscillator Start-up Timer)

1 = Kết thúc thời gian chờ khởi động từ bộ OST; bộ phát xung chính hoạt động

0 = Đang đếm thời gian khởi động; bộ phát xung chính chưa hoạt động

bit 2 **IOFS**: Bit báo sự ổn định tín hiệu bộ phát xung nội INTOSC

1 = Bộ phát xung nội INTOSC ở trạng thái ổn định

0 = Bộ phát xung nội INTOSC chưa ổn định

bit 1-0 **SCS1:SCS0:** Bit lựa chọn nguồn xung cho hệ thống

1x = Nguồn hệ thống từ bộ dao động nội

01 = Nguồn xung phụ, nối qua các chân của Timer1(Secondary oscillator)

00 = Nguồn xung chính qua các chân OSC1, OSC2 (Primary oscillator)

- Thanh ghi cấu hình 1 byte cao: CONFIG1H

R/P-0	R/P-0	U-0	U-0	R/P-0	R/P-1	R/P-1	R/P-1
IESO	FCMEN	—	—	FOSC3	FOSC2	FOSC1	FOSC0
bit 7							bit 0

bit 7 **IESO**: Bit cho phép luân phiên bộ phát xung nội/ngoại

1 = Cho phép

0 = Không cho phép

bit 6 **FCMEN**: Bit cho phép chế độ quản lý an toàn bộ phát xung (Fail-Safe Clock Monitor)

1 = Cho phép

0 = Không cho phép

bit 5-4 **không sử dụng**: đọc sẽ được ‘0’

bit 3-0 **FOSC3:FOSC0:** Bit lựa chọn bộ phát xung

11xx = Chế độ phát xung RC ngoài, chức năng phát xung CLKO trên chân RA6

101x = Chế độ phát xung RC ngoài, chức năng phát xung CLKO trên chân RA6

1001 = Chế độ phát xung nội, chức năng phát xung CLKO trên chân RA6, vào/ra trên chân RA7

1000 = Chế độ phát xung nội, vào/ra trên chân RA6 và RA7

0111 = Chế độ dao động RC ngoài, vào/ra trên chân RA6

0110 = Chế độ HS, cho phép PLL (xung hệ thống được nhân 4)

0101 = Chế độ EC, vào/ra trên chân RA6

0100 = Chế độ EC, phát xung trên chân RA6

0011 = Chế độ phát xung RC ngoài, chức năng phát xung CLKO trên chân RA6

0010 = Chế độ HS

0001 = Chế độ XT

0000 = Chế độ LP

1.5. Các thanh ghi của hoạt động Reset

- Thanh ghi điều khiển Reset: RCON

R/W-0	R/W-1(1)	U-0	R/W-1	R-1	R-1	R/W-0(2)	R/W-0
IPEN	SBOREN	—	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$
bit 7							bit 0

bit 7 **IPEN:** Bit cho phép ưu tiên ngắt

1 = Cho phép ưu tiên ngắt.

0 = Không cho phép ưu tiên ngắt.

bit 6 SBOREN: Bit cho phép reset BOR bằng phần mềm.

Nếu BOREN1:BOREN0 = 01:

1 = Cho phép reset BOR

0 = Không cho phép reset BOR

Nếu BOREN1:BOREN0 = 00, 10 or 11:

Không được sử dụng, đọc sẽ được ‘0’

bit 5 **Không được sử dụng**: Đọc sẽ được ‘0’

bit 4 **RI**: Bit cờ lệnh RESET

1 = Lệnh RESET không được thực hiện

0 = Lệnh RESET được thực hiện (phải được thiết lập sau khi xảy ra ngắt BOR)

bit 3 **TO**: Bit cờ báo Watchdog Time-out (thời gian đặt cho bộ WDT)

1 = Thiết lập khi bật nguồn (power-up), lệnh CLRWDT hoặc lệnh SLEEP

0 = Xảy ra sự kiện WDT (yêu cầu reset hệ thống bằng WDT)

bit 2 **PD**: Bit cờ phát hiện ngắt nguồn

1 = Thiết lập khi bật nguồn (power-up) hoặc lệnh CLRWDT

0 = Khi thực hiện lệnh SLEEP

bit 1 **POR**: Bit trạng thái reset bật nguồn POR (Power-on Reset)

1 = Không xảy ra hiện tượng bật nguồn

0 = Xảy ra hiện tượng bật nguồn (phải được đặt bằng ‘1’ sau ngắt khi xảy ra reset POR)

bit 0 **BOR**: Bit trạng thái reset sụt nguồn BOR (Brown-out Reset)

1 = Không xảy ra hiện tượng sụt nguồn

0 = Xảy ra hiện tượng sụt nguồn (phải được đặt bằng ‘1’ sau ngắt khi xảy ra reset BOR)

- Thanh ghi cầu hình 3 byte cao: CONFIG3H

R/P-1	U-0	U-0	U-0	U-0	R/P-0	R/P-1	R/P-1
MCLRE	—	—	—	—	LPT1OSC	PBADEN	CCP2MX
bit 7							bit 0

bit 7 **MCLRE**: Bit cho phép reset trên chân MCLR/RE3

1 = Là chân reset MCLR

0 = Là chân vào/ra RE3

bit 6-3 **Không được sử dụng**: Đọc sẽ được ‘0’

bit 2 **LPT1OSC**: Bit cho phép bộ phát xung LPT1(Low-Power Timer1)

1 = Cấu hình Timer1 hoạt động ở điện áp thấp (low-Power)

0 = Cấu hình Timer1 hoạt động ở điện áp cao (higher power)

bit 1 **PBADEN**: Bit cho phép A/D PORTB

(Ảnh hưởng đến thanh ghi ADCON1 khi Reset)

1 = Các chân PORTB<4:0> được cấu hình là vào/ra tương tự khi Reset

0 = Các chân PORTB<4:0> được cấu hình là vào/ra số khi Reset

bit 0 **CCP2MX**: Bit MUX CCP2

1 = CCP2 nối với RC1

0 = CCP2 nối với RB3

- Thanh ghi cầu hình 2 byte thấp: CONFIG2L

U-0	U-0	U-0	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
—	—	—	BORV1(1)	BORV0(1)	BOREN1(2)	BOREN0(2)	$\overline{\text{PWRTEN}}$(2)
bit 7							bit 0

bit 7-5 **Unimplemented**: Đọc được ‘0’

bit 4-3 **BORV1:BORV0**: Bit chọn điện áp Reset BOR (Brown-out Reset)

11 = Max

...

00 = Min

bit 2-1 **BOREN1:BOREN0**: Bit cho phép Reset BOR (Brown-out Reset)

11 = BOR hoạt động ở chế độ phần cứng.

10 = BOR hoạt động ở chế độ phần cứng và chạy ở chế độ Run và Idle, không sử dụng ở chế độ Sleep.

01 = Cho phép điều khiển BOR bằng phần mềm; bit SBOREN điều khiển cho phép BOR.

00 = Cấm Reset BOR

bit 0 **PWRTEN**: Power-up Timer Enable bit(2)

1 = PWRT disabled

0 = PWRT enabled

1.6. Hoạt động vào/ra

Tên	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PORTA	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0
LATA	LATA7	LATA6	Thanh ghi chốt dữ liệu PORTA					
TRISA	TRISA7	TRISA6	Thanh ghi hướng dữ liệu PORTA					
ADCON1	—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0
CMCON	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0
CVRCON	CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0

Các thanh ghi liên quan đến PORTA.

Tên	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
LATB	Thanh ghi chốt dữ liệu PORTB							
TRISB	Thanh ghi hướng dữ liệu PORTB							
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
INTCON2	RBPU	INTEDG0	INTEDG1	INTEDG2	—	TMR0IP	—	RBIP
INTCON3	INT2IP	INT1IP	—	INT2IE	INT1IE	—	INT2IF	INT1IF
ADCON1	—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0

Các thanh ghi liên quan đến PORTB.

Tên	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0
LATC	Thanh ghi chốt dữ liệu của PORTC (Chốt dữ liệu đọc ghi)							
TRISC	Thanh ghi chọn hướng dữ liệu của PORTC							

Các thanh ghi liên quan đến PORTC.

Tên	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PORTD	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0
LATD	Thanh ghi chốt dữ liệu của PORTD (Chốt dữ liệu đọc ghi)							
TRISD	Thanh ghi chọn hướng dữ liệu của PORTD							
TRISE(1)	IBF	OBF	IBOV	PSPMODE	—	TRISE2	TRISE1	TRISE0
CCP1CON	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0

Các thanh ghi liên quan đến PORTD.

Tên	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PORTE	—	—	—	—	RE3	RE2	RE1	RE0
LATE(2)	—	—	—	—	—	Thanh ghi xuất dữ liệu LATE		
TRISE	IBF	OBF	IBOV	PSPMODE	—	TRISE2	TRISE1	TRISE0
ADCON1	—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0

Bảng 2.15. Các thanh ghi liên quan đến PORTE.

1.7. Lệnh điều khiển vào/ra theo byte hoặc theo bit

- Lệnh điều khiển đọc dữ liệu theo byte

```
x = PORTD;   // x là một biến kiểu byte
```

- Lệnh điều khiển ghi dữ theo byte

```
PORTD = x;   // x là một biến kiểu byte
```

- Lệnh điều khiển đọc dữ liệu theo bit

```
x = PORTDbits.RD0;   // x là một biến kiểu bit
```

Hoặc:

```
#define SW PORTBbits.RB2 // định nghĩa RB2 = SW
x = SW;  // lệnh đọc
```

- Lệnh điều khiển ghi dữ theo bit

```
PORTDbits.RD0 = x;   // x là một biến kiểu bit
```

Hoặc:

```
#define LED PORTCbits.RC2  // định nghĩa RC2 = LED
LED = x;  // lệnh xuất
```

2. NGÔN NGỮ LẬP TRÌNH VÀ TRÌNH DỊCH

2.1. Khung một chương trình viết cho vi điều khiển.

```
//khai báo các thư viện, ví dụ:
#include <P18f4520.h>
#include <delays.h>
//Cấu hình cho vi điều khiển, ví dụ:
#pragma config OSC = HS          //chế độ dao động HS
#pragma config MCLRE = ON        //sử dụng RE3 làm chân reset (OFF: không sử dụng)
#pragma config WDT = OFF         //không dùng Watchdog timer
#pragma config PBADEN = OFF      //PORTB<4:0> được cấu hình thành các chân vào/ra số
                                 //ON: PORTB<4:0> được cấu hình thành các chân AN8-AN12
#pragma config PWRT=ON           // sử dụng Power-up timer
#pragma config BOREN=OFF         // không sử dụng chức năng Bown-out reset
                                 //(reset khi Vdd xuống thấp dưới 1 ngưỡng)
#pragma config LVP=OFF           //không dụng chế độ cấp nguồn chỉ từ mạch nạp
                                 //(Single-Supply ICSP Programming)

/*khai báo biến số,hằng số,cấu trúc,chương trình con, ví dụ:*/
int x;
char m[10];
//Khai báo tên các chương trình con, ví dụ:
void high_isr (void);
//viết các chương trình con, ví dụ:
void high_isr (void)
{
//các câu lệnh
}
//chương trình chính
void main (void)
{
//các câu lệnh
}
```

2.1.1. Hằng số.

Một hằng số thông thường được định nghĩa bởi từ khoá const:

Ví dụ:

```
const  unsigned int c = 100;
const  unsigned char tens[] = { 1, 10, 100, 1000 };
```

Hằng số trong ROM được định nghĩa bởi từ khoá **rom**:

```
unsigned char rom coolant_temp = 0x02 ;
```

Một mảng các giá trị nằm trong ROM có thể được định nghĩa như sau:

```
unsigned char rom anh_so[] =
{
0x08,0x08,0x00,0x00,0x00,0x09,0x41,0x80,0xC0,0xFF,0x00,0x00,0x13,0x1A,0x26,0
x33,0x80,0xFF,0x00,0x00,0x00,0x09,0x41,0x80,0x66,0x66,0x00,0x00,0x00,0x05,0x4
A,0x46,0x40,0x40,0x00,0x00,0x00,0x08,0x43,0x43,0x3D,0x3A,0x00,0x00,0x00,0x0
0,0x2D,0x4D,0x56,0x4D,0x00,0x00,0x00,0x00,0x21,0x56,0x6C,0x6F
} ;
```

2.1.2. Biến.

Một số kiểu biến được dùng trong MCC18 như sau:

Biến nguyên:

Loại biến	Độ dài	Khoảng giá trị
char	8 bits	-128 đến 127
signed char	8 bits	-128 đến 127
unsigned char	8 bits	0 đến 255
int	16 bits	-32,768 đến 32,767
unsigned int	16 bits	0 đến 65,535
short	16 bits	-32,768 đến 32,767
unsigned short	16 bits	0 đến 65,535
short long	24 bits	-8,388,608 đến 8,388,607
unsigned short long	24 bits	0 đến 16,777,215
long	32 bits	-2,147,483,648 đến 2,147,483,647
unsigned long	32 bits	0 đến 4,294,967,295

Biến thực:

Loại biến	Độ dài	Khoảng giá trị
float	32 bits	$2^{-126} \approx 1.17549435e - 38$ đến $2^{128} * (2-2^{-15}) \approx 6.80564693e + 38$

2.1.3. Lưu trữ các đối tượng trong bộ nhớ.

MCC18 cung cấp bốn từ khóa dùng để lưu trữ các đối tượng (ví dụ: biến số, hằng số) trong bộ nhớ chương trình, dữ liệu bao gồm: far, near, rom và ram, cụ thể như sau:

far rom: Lưu các đối tượng ở mọi vị trí trong bộ nhớ chương trình

near rom: Lưu các đối tượng ở các vị trí trong bộ nhớ chương trình địa chỉ nhỏ hơn 64Kb

far ram: Lưu các đối tượng ở mọi vị trí trong bộ nhớ dữ liệu

near ram: Lưu các đối tượng trong vùng Ram dành cho người sử dụng (Access RAM)

2.1.4. Con trỏ

MCC18 cung cấp ba loại biến con trỏ như sau:

Kiểu	độ dài	ví dụ
Data memory pointer	16 bits	char * dmp;
Near program memory pointer rom near	16 bits	char * npmp;
Far program memory pointer rom	24 bits	far char * fpmp;

2.1.5. Các định danh phần cứng.

Các định danh phần cứng là tên của các thanh ghi, các bit chức năng có trên phần cứng của vi điều khiển PIC đã được định nghĩa trong tập tin Pxxx.h. Định danh phần cứng của PIC18F4520 chứa trong tệp tin P18f4520.h. Một phần của tệp tin này dùng để định nghĩa các bit trong thanh ghi PORTA như sau:

```
extern volatile near unsigned char      PORTA;
extern volatile near union {
  struct {
    unsigned RA0:1;
    unsigned RA1:1;
    unsigned RA2:1;
    unsigned RA3:1;
    unsigned RA4:1;
    unsigned RA5:1;
    unsigned RA6:1;
    unsigned RA7:1;
  };
```

Ngoài ra người lập trình cũng có thể định nghĩa các định danh phần cứng bằng cú pháp **#define.** Ví dụ:

```
#define contact       PORTAbits.RA5
//Chân RA5 của PORTA được gán tên contact.
#define LED PORTAbits.RA4
//Chân RA4 của PORTA được gán tên LED
….
```

```
if(!contact)  LED=1;
//đọc vào, nếu contact (RA5)=0 xuất ra LED (RA4)=1
else LED=0            //và ngược lại
```

2.1.6. Viết các lệnh hợp ngữ trong chương trình.

Các lệnh hợp ngữ được viết trong các dòng bắt đầu bằng từ khóa **_asm** và kết thúc bằng từ khóa **_endasm.** Ví dụ:

```
_asm
MOVLW 5
_endasm
```

2.2. Các hàm

Trong trình dịch C, hàm có thể do người lập trình xây dựng và cũng có thể là các hàm đã có sẵn trong thư viện của trình dịch, chúng ta sẽ xem xét cả hai loại này.

2.2.1. Xây dựng hàm.

2.2.1.1. Khai báo và sử dụng hàm.

Cách khai báo một hàm:

[giá trị trả về] [tên hàm(các đối số)];

Ví dụ:

int cong(int a,int b); //khai báo một hàm tên "cong" với 2 đối số là a,b

//hàm trả về kiểu int

void delay(void); //hàm không đối cũng không trả về giá trị

Ví dụ: Chương trình có dùng hàm:

```
#include<P18f4520.h>
#include<delays.h>
void delay_second (int d);
void delay_second(int d)
{
int i;
for(i=0;i<d;i++)
Delay10KTCYx(200); //trễ 1 giây (thạch anh 8Hhz)
}
void main()
{
while(1)
{
PORTB=0x00;
```

```
delay_second (6);              //gọi hàm với đối số d=6, trễ 6 giây
PORTB=0xFF;
delay_second (6);
}
}
```

2.2.1.2. Hàm ngắt.

Hàm ngắt là các hàm được dùng với hoạt động ngắt của PIC, các hàm này sẽ được bộ đếm chương trình PC trỏ tới khi xảy ra ngắt.

Cách khai báo :

(1) Khai báo địa chỉ ngắt :

#pragma code [tên địa chỉ ngắt]=[địa chỉ ngắt]

void [tên chương trình ngắt tương ứng địa chỉ ngắt] (void)

{//các câu lệnh}

Ví dụ :

#pragma code high_vector = 0x08 // vector ngắt cao

void interrupt_hight_vector (void)

{_asm GOTO high_isr   _endasm}

(2) Khai báo và viết chương trình con phục vụ ngắt :

#pragma code

#pragma interrupt [tên hàm phục vụ ngắt]

[tên hàm phục vụ ngắt] (void)

{ //các câu lệnh}

Ví dụ:

```
#pragma code
#pragma interrupt high_isr
void high_isr (void)
{
unsigned int k;
if (INTCONbits.INT0IF==1)
        INTCONbits.INT0IF=0;
for(k=0;k<10;k++)
        {
        PORTBbits.RB5 = 1;
        Delay10KTCYx(100);
        PORTBbits.RB5 = 0;
        Delay10KTCYx(50);
```

```
        }
}
```

2.2.2. Hàm trong thư viện của trình dịch.

MCC18 cung cấp một số lượng lớn các hàm giúp người lập trình có thể phát triển các ứng dụng một các nhanh nhất. Các hàm của MCC18 chia thành ba nhóm

- Nhóm các hàm sử dụng cho các tài nguyên ngoại vi của PIC18Fxxx (ví dụ: các hàm phục vụ giao tiếp ADC).

- Nhóm các hàm sử dụng cho các các phần cứng thông dụng có thể kết nối với PIC18Fxxx (ví dụ LCD).

- Nhóm các hàm công dụng chung (ví dụ: các hàm tạo trễ)

- Nhóm các hàm toán học.

Các hàm được viết trong các tệp tin *.h tương ứng, người lập trình có thể xem trực tiếp trên các tập tin này để sử dụng các hàm.

Ví dụ: Trong tập tin delay.h chứa 05 hàm tạo trễ, bao gồm:

TT	Tên hàm	Mô tả
1	Delay1TCY	Trễ 1 chu kỳ lệnh
2	Delay10TCYx	Trễ 10 chu kỳ lệnh
3	Delay100TCYx	Trễ 100 chu kỳ lệnh
4	Delay1KTCYx	Trễ 1000 chu kỳ lệnh
5	Delay10KTCYx	Trễ 10.000 chu kỳ lệnh

Các hàm trong thư viện delay.h

Để sử dụng các hàm trong trình dịch, người lập trình cần khai báo thư viện chứa hàm. Ví dụ: Để tạo trễ 1 giây (giả thiết bộ tạo dao động hoạt động ở chế độ HS, thạch anh sử dụng có tần số 8Mhz) cần khai báo và viết các lệnh sau:

```
#include <delay.h>
...
Delay10KTCYx(200);
//thời gian trễ: 200 * 10.000 chu kỳ lệnh * 0.5 uS = 1 s
```

2.3. Các lệnh xử lý bit.

Lệnh "And" :

Cú pháp: &

Ví dụ:

```
unsigned char x,y,z;
x=0b11110000;
y=0b11101111;
y=y&x; //y=0b11100000
z&=0b1111110; //z=*******0 (xóa bit có giá trị thấp nhất của biến z)
```

Lệnh "Or" :

Cú pháp: |

Ví dụ:

```
unsigned char x,y,z;
x=0b11110000;
y=0b11101111;
y=y|x; //y=0b11111111
z|=0b1000000; //z=1******* (thiết lập bit có giá trị cao nhất của biến z)
```

Lệnh "Xor" :

Cú pháp: ^

Ví dụ:

```
unsigned char x,y;
x=0b11110000;
y=0b11101111;
y=y^x; //y=0b00011111
```

Lệnh "Not" :

Cú pháp: ~

Ví dụ:

```
#define xung PORTBbits.RB0
xung=1;
~xung; //xung=0
```

Lệnh dịch trái :

Cú pháp: <<

Ví dụ:

```
unsigned char x;
x=0b11110000;
x=x<<2; //x=0b11000000
```

Lệnh dịch phải :

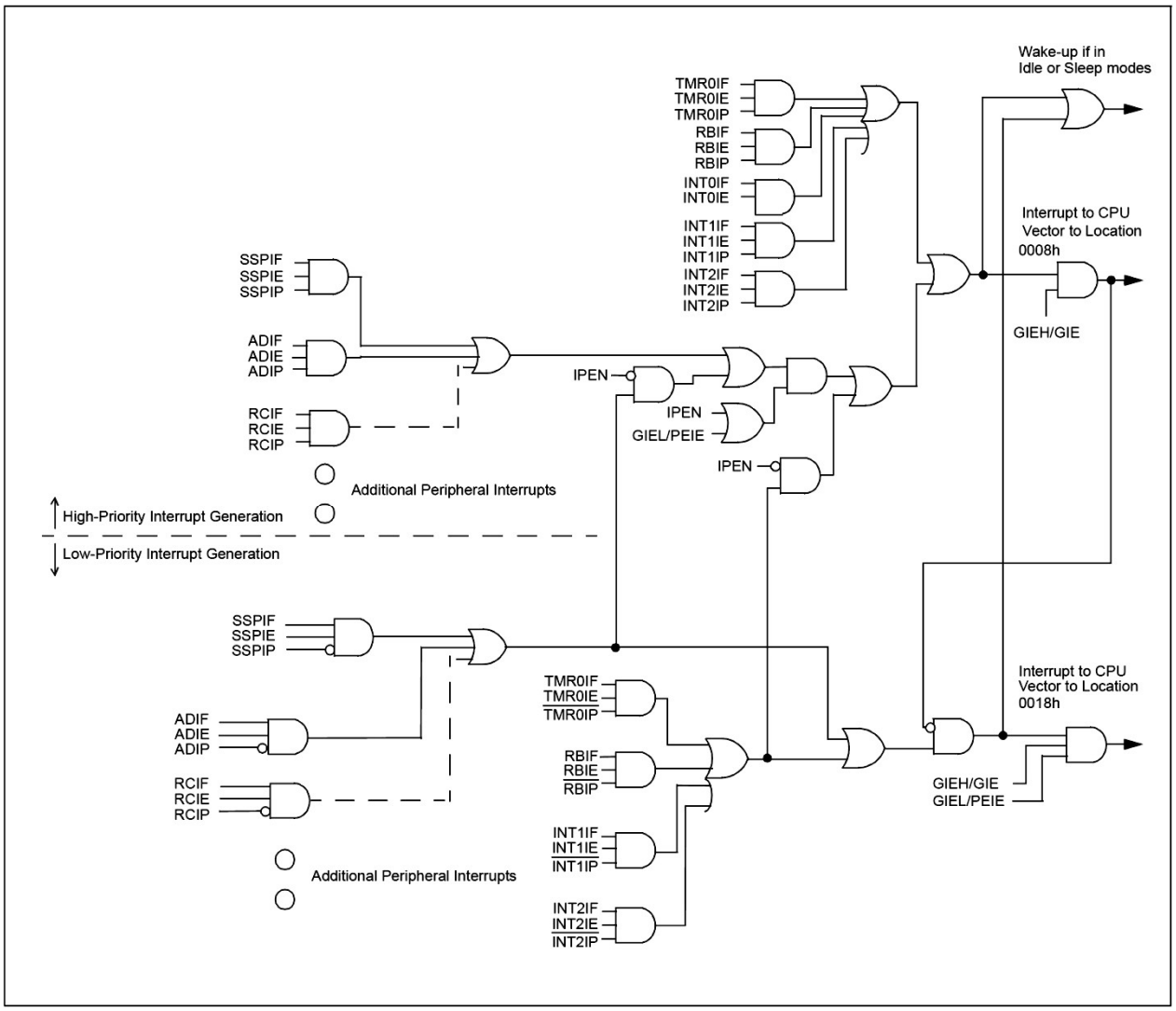
Cú pháp: >>

Ví dụ:

```
unsigned char x;
x=0b11110000;
x=x>>2; //x=0b00110000
```

3. HOẠT ĐỘNG NGẮT

3.1. Tổ chức ngắt của PIC 18F4520



Logic ngắt của PIC 18F4520

3.2. Các thanh ghi liên quan

Vi điều khiển PIC 18F4520 có 10 thanh ghi điều khiển hoạt động ngắt:

• RCON – thanh ghi điều khiển Reset.
• INTCON – thanh ghi điều khiển ngắt.
• INTCON2 – thanh ghi điều khiển ngắt 2.
• INTCON3 – thanh ghi điều khiển ngắt 3.
• PIR1, PIR2 – thanh ghi yêu cầu ngắt ngoại vi 1, thanh ghi yêu cầu ngắt ngoại vi 2.
• PIE1, PIE2 – thanh ghi cho phép ngắt ngoại vi 1, thanh ghi cho phép ngắt ngoại vi 2.
• IPR1, IPR2 – thanh ghi ưu tiên ngắt ngoại vi 1, thanh ghi ưu tiên ngắt ngoại vi 2.

3.2.1. Thanh ghi RCON

R/W-0	R/W-1(1)	U-0	R/W-1	R-1	R-1	R/W-0(2)	R/W-0
IPEN	SBOREN	—	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$
bit 7							bit 0

bit 7 **IPEN:** Bit cho phép ưu tiên ngắt

1 = Cho phép ưu tiên ngắt.

0 = Không cho phép ưu tiên ngắt.

Các bít khác: Xem phần 2.5

3.2.2. Các thanh ghi điều khiển ngắt INTCON

Các thanh ghi INTCON (Interrupt Control) có thể ghi/đọc theo cả byte hoặc từng bit. Các thanh ghi này chứa các bit cho phép/cấm các nguồn ngắt, các bit đặt mức ưu tiên, các bit cờ ngắt.

- Thanh ghi điều khiển ngắt: INTCON

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/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF(1)
bit 7							bit 0

Ghi chú:

R = Cho phép đọc W = Cho phép ghi U = Không sử dụng, đọc bằng '0'

-n = Reset - POR '1' = Được thiết lập '0' = Được xóa -x = Reset không xác định

bit 7 **GIE/GIEH:** Bit cho phép ngắt toàn cục

Khi bit **IPEN** = 0:

1 = Cho phép tất cả các ngắt không sử dụng mặt nạ (không ưu tiên ngắt)

0 = Cấm tất cả các ngắt

Khi bit **IPEN** = 1:

1 = Cho phép tất cả các ngắt ưu tiên cao

0 = Cấm tất cả các ngắt

bit 6 **PEIE/GIEL:** Bit cho phép ngắt ngoại vi

Khi bit **IPEN** = 0:

1 = Cho phép tất cả các ngắt ngoại vi không sử dụng mặt nạ

0 = Cấm tất cả các ngắt ngoại vi

Khi bit **IPEN** = 1:

1 = Cho phép tất cả các ngắt ngoại vi ưu tiên thấp

0 = Cấm tất cả các ngắt ngoại vi được ưu tiên thấp

bit 5 **TMR0IE:** Bit cho phép ngắt tràn Timer0 (TMR0)

1 = Cho phép ngắt tràn Timer0(TMR0)

0 = Cấm ngắt tràn Timer0(TMR0)

bit 4 **INT0IE:** Bit cho phép ngắt ngoài INT0

1 = Cho phép ngắt ngoài INT0

0 = Cấm ngắt ngoài INT0

bit 3 **RBIE:** Bit cho phép ngắt do thay đổi mức trên PortB

1 = Cho phép Ngắt khi có thay đổi mức trên PortB
0 = Cấm Ngắt khi có thay đổi mức trên PortB

bit 2 **TMR0IF**: Bit cờ báo ngắt tràn Timer0 (TMR0)
1 = Tràn thanh ghi TMR0 của Time0 (phải được xóa bằng phần mềm)
0 = không xảy ra tràn thanh ghi TMR0

bit 1 **INT0IF**: Bit cờ báo ngắt ngoài INT0
1 = Có ngắt ngoài INT0 (phải được xóa bằng phần mềm)
0 = Chưa phát hiện ngắt ở chân INT0

bit 0 **RBIF**: Bit cờ ngắt báo đã thay đổi mức trên PortB
1 = Ít nhất một bit từ RB7:RB4 có sự thay đổi trạng thái (bit này phải được xóa bằng phần mềm)
0 = Không có sự thay đổi trạng thái trên các chân RB7:RB4

- Thanh ghi điều khiển ngắt 2: INTCON2

R/W-1	R/W-1	R/W-1	R/W-1	U-0	R/W-1	U-0	R/W-1
RBPU	INTEDG0	INTEDG1	INTEDG2	—	TMR0IP	—	RBIP
bit 7							bit 0

bit 7 **RBPU**: Bit cho phép treo các chân PORTB (Pull-up)
1 = Cấm treo PortB
0 = Cho phép treo PortB

bit 6 **INTEDG0**: Bit lựa chon sườn xung cho ngắt ngoài INT0
1 = Ngắt bằng sườn dương
0 = Ngắt bằng sườn âm

bit 5 **INTEDG1**: Bit lựa chon sườn xung cho ngắt ngoài INT1
1 = Ngắt bằng sườn dương
0 = Ngắt bằng sườn âm

bit 4 **INTEDG2**: Bit lựa chon sườn xung cho ngắt ngoài INT2
1 = Ngắt bằng sườn dương
0 = Ngắt bằng sườn âm

bit 3 Bit không được định nghĩa

bit 2 **TMR0IP**: Bit lựu chọn mức ưu tiên ngắt cho ngắt tràn Timer0 (TMR0)
1 = Ưu tiên cao
0 = Ưu tiên thấp

bit 1 Bit không được định nghĩa

bit 0 **RBIP**: Bit lựa chọn mức ưu tiên ngắt do thay đổi PortB
1 = Ưu tiên cao
0 = Ưu tiên thấp

- Thanh ghi điều khiển ngắt 3: INTCON3

R/W-1	R/W-1	U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
INT2IP	INT1IP	—	INT2IE	INT1IE	—	INT2IF	INT1IF
bit 7							bit 0

bit 7 **INT2IP:** Bit lựu chọn mức ưu tiên ngắt cho ngắt ngoài INT2
1 = Ưu tiên cao

0 = Ưu tiên thấp

bit 6 **INT1IP:** Bit lựu chọn mức ưu tiên ngắt cho ngắt ngoài INT1
1 = Ưu tiên cao
0 = Ưu tiên thấp

bit 5 Không được định nghĩa

bit 4 **INT2IE:** Bit cho phép ngắt ngoài INT2
1 = Cho phép ngắt ngoài INT2
0 = Cấm ngắt ngoài INT2

bit 3 **INT1IE:** Bit cho phép ngắt ngoài INT1
1 = Cho phép ngắt ngoài INT1
0 = Cấm ngắt ngoài INT1

bit 2 Không được định nghĩa

bit 1 **INT2IF:** Cờ báo ngắt ngoài INT2
1= Xuất hiện ngắt ngoài tren chân INT2 (phải được xóa băng phần mềm)
0 = Không xảy ra ngắt trên chân INT2

bit 0 **INT1IF**: Cờ báo ngắt ngoài INT1
1= Xuất hiện ngắt ngoài tren chân INT1 (phải được xóa băng phần mềm)
0 = Không xảy ra ngắt trên chân INT1

3.2.3. Thanh ghi yêu cầu ngắt PIR

Các thanh ghi yêu cầu ngắt ngoại vi PIR (Peripheral Interrupt Request) chứa các bit cờ phản ánh trạng thái có/không có các yêu cầu ngắt ngoại vi.

- Thanh ghi yêu cầu ngắt ngoại vi 1: PIR1

R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
bit 7							bit 0

bit 7 **PSPIF:** Bit cờ ngắt ghi/đọc Port song song tớ (Parallel Slave Port)
1 = Đọc hoặc ghi thành công ở Port song song(Phải xóa bằng phần mềm)
0 = đã không xảy ra đọc hoặc ghi

bit 6 **ADIF:** Bit cờ ngắt của bộ biến đổi A/D
1 = Biến đổi A/D đã hoàn thành (phải xóa bằng phần mềm)
0 = Biến đổi A/D chưa hoàn thành

bit 5 **RCIF:** Bit cờ ngắt nhận USART EUSART
1 = khi bộ đệm nhân RCREG nhận đủ dữ liệu (được xóa khi đọc RCREG)
0 = bộ đệm nhân rỗng

bit 4 **TXIF:** Bit cờ ngắt truyền USART EUSART
1 = khi bộ đệm truyền TXREG rỗng (xóa khi TXREG được ghi)
0 = khi bộ đệm truyền được ghi EUSART

bit 3 **SSPIF:** Bit cờ ngắt USART đồng bộ chủ (Master Synchronous Serial Port)
1 = khi việc truyền và nhân ở USART hoàn thành (xóa bằng phần mềm)

0 = chờ truyền/nhận

bit 2 **CCP1IF:** Bit cờ ngắt module CCP1 (CAPTURE/COMPARE/PWM)

Chế độ chụp (Capture):
1 = Xảy ra hiên tượng chụp(capture) ở thanh ghi TMR1 (xóa bằng phần mềm)
0 = Không xảy ra hiện tượng chụp ở thanh ghi TMR1

Chế độ so sánh (Compare):
1 = Giá trị thanh ghi đếm TMR1 đếm bằng thanh ghi so sánh (xóa bằng phần mềm)
0 = Giá trị TMR1 chứa đếm bằng thanh ghi so sánh

Chế độ PWM:
Không sử dụng ở chế độ này

bit 1 **TMR2IF:** Bit cờ ngắt khi bộ đếm TMR2 so sánh giá trị ở PR2 của Timer2.
1 = Khi giá trị TMR2 bằng giá trị PR2 (được xóa bằng phần mềm)
0 = TMR2 không phù hợp PR2

bit 0 **TMR1IF:** Bit cờ ngắt tràn Timer1 (TMR1)
1 = Khi tràn thanh ghi TMR1 (Phải được xóa bằng phần mềm)
0 = Chưa phát hiện tràn trên thanh ghi TMR1

- Thanh ghi yêu cầu ngắt ngoại vi 2: PIR2

R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
OSCFIF	CMIF	—	EEIF	BCLIF	HLVDIF	TMR3IF	CCP2IF
bit 7							bit 0

bit 7 **OSCFIF**: Bit cờ ngắt lỗi bộ phát xung (Oscillator Fail)
1 = Bộ phát xung đã vị lỗi, đầu vào tần số INTOSC đã bị thay đổi (phải được xóa bằng phần mềm)
0 = Thiết bị phát xung hoạt động bình thường

bit 6 **CMIF**: Bit cờ ngắt từ bộ so sánh (Comparator)
1 = Tín hiệu đầu vào bộ so sánh đã thay đổi làm thay đổi đầu ra bộ ss (xóa bằng phần mềm)
0 = Chua có sự thay đổi tín hiệu vào bộ so sánh

bit 5 Không được định nghĩa

bit 4 **EEIF**: Bit cờ ngắt do hoạt động ghi dữ liệu vào EEPROM/Flash
1 = Hoàn thành việc ghi dữ liệu (xóa bằng phần mềm)
0 = Hoạt động chưa hoàn thành hoặc chưa được bắt đầu

bit 3 **BCLIF**: Bit cờ ngắt xung đột Bus (Bus Collision)
1 = Xuất hiện sự xung đột bus (xóa bằng phần mềm)
0 = Không có xung đột bus

bit 2 **HLVDIF**: Bit cờ ngắt phát hiện điện áp Cao/thấp(High/Low-Voltage Detect)
1 = Đã phát hiện trạng thái diện áp cao/thấp
0 = Chưa phát hiện thấy điện áp cao/thấp (A high/low-voltage)

bit 1 **TMR3IF**: Bit cờ ngắt tràn Timer3 (tràn thanh ghi TMR3)

1 = Tràn thanh ghi TMR3(xóa bằng phần mềm)
0 = Chưa tràn ở thanh ghi TMR3

bit 0 **CCP2IF**: Bit cờ ngắt của module CCP2 (Capture/Compare/PWM 2)

Chế độ chụp (Capture):

1 = Sau khi chụp thanh ghi TMR1 (xóa bằng phần mềm)
0 = Không xuất hiện chụp TMR1

Chế độ so sánh (Compare):

1 = Khi giá trị trong thanh ghi TMR1 bằng (match) giá trị trong thanh ghi CCPR2 (xóa bằng phần mềm)
0 = Khi giá trị trong thanh ghi TMR1 chưa bằng giá trị trong thanh ghi CCPR2

3.2.4. Thanh ghi cho phép ngắt ngoại vi PIE

Các thanh ghi cho phép ngắt ngoại vi PIE (Peripheral Interrupt Enable) chứa các bit cho phép các nguồn ngắt ngoại vi. Có 2 thanh ghi cho phép ngắt ngoại vi là PIE1 và PIE2. Khi bit IPEN (bit 7 thanh ghi RCON) được đặt bằng 0, muốn cho phép các nguồn ngắt ngoại vi bit PEIE (bit 6 trong thanh ghi INTCON) phải được đặt bằng 1.

- Thanh ghi cho phép ngắt ngoại vi 1: PIE1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
bit 7							bit 0

bit 7 **PSPIE:** Bit cho phép ngắt ghi/đọc PORT song song tớ -PSP (Parallel Slave Port)

1 = Cho phép ngắt ghi/đọc PSP
0 = Cấm ngắt ghi/đọc PSP

bit 6 **ADIE:** Bit cho phép ngắt do biến đổi A/D

1 = Cho phép ngắt biến đổi A/D
0 = Cấm ngắt biến đổi A/D

bit 5 **RCIE:** Bit cho phép ngắt do nhận ở USART (EUSART)

1 = Cho phép ngắt
0 = Cấm ngắt

bit 4 **TXIE:** Bit cho phép ngắt do truyền ở USART (EUSART)

1 = Cho phép ngắt
0 = Cấm ngắt

bit 3 **SSPIE**: Bit cho phép ngắt USART đồng bộ chủ - MSSP (Master Synchronous Serial Port)

1 = Cho phép ngắt (MSSP)
0 = Cấm ngắt

bit 2 **CCP1IE:** Bit cho phép ngắt của module CCP1 (Capture/Compare/PWM 1)

1 = Cho phép ngắt CCP1
0 = Cấm

bit 1 **TMR2IE:** Bit cho phép ngắt do so sánh TMR2 và PR2 của Timer2

1 = Cho phép ngắt
0 = Cấm

bit 0 **TMR1IE:** Bit cho phép ngắt tràn của Timer1
1 = Cho phép ngắt do tràn TMR1
0 = Cấm ngắt

- Thanh ghi cho phép ngắt ngoại vi 2: PIE2

R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
OSCFIE	CMIE	—	EEIE	BCLIE	HLVDIE	TMR3IE	CCP2IE
bit 7							bit 0

bit 7 **OSCFIE:** Bit cho phép ngắt do lỗi bộ phát xung (Oscillator Fail)
1 = Cho phép
0 = Cấm

bit 6 **CMIE:** Bit cho phép ngắt từ bộ so sánh (Comparator)
1 = Cho phép
0 = Cấm

bit 5 Không được định nghĩa

bit 4 **EEIE:** Bit cho phép ngắt ghi dữ liệu vào EEPROM/Flash
1 = Cho phép
0 = Cấm

bit 3 **BCLIE:** Bit cho phép ngắt do xung đột Bus (Bus Collision)
1 = Cho phép
0 = Cấm

bit 2 **HLVDIE:** Bit cho phép ngắt do module HLVD (High/Low-Voltage Detect)
1 = Cho phép
0 = Cấm

bit 1 **TMR3IE**: Bit cho phép ngắt tràn Timer3
1 = Cho phép
0 = Cấm

bit 0 **CCP2IE:** Bit cho phép ngắt từ module CCP2 (Capture/Compare/PWM 2)
1 = Cho phép
0 = Cấm

3.2.5. Thanh ghi ưu tiên ngắt IPR

- Thanh ghi ưu tiên ngắt 1: IPR1

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
PSPIP(1)	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP
bit 7							bit 0

bit 7 **PSPIP**: Bit ưu tiên ngắt ghi/đọc PORT song song tớ (Parallel Slave Port)
1 = Ưu tiên cao
0 = Ưu tiên thấp

bit 6 **ADIP**: Bit ưu tiên ngắt biến đổi A/D (Analog Digital Converter)
1 = Ưu tiên cao
0 = Ưu tiên thấp

bit 5 **RCIP**: Bit ưu tiên ngắt nhận USART (EUSART Receive)
1 = Ưu tiên cao
0 = Ưu tiên thấp

bit 4 **TXIP**: Bit ưu tiên ngắt truyền USART (EUSART Transmit)
1 = Ưu tiên cao
0 = Ưu tiên thấp

bit 3 **SSPIP**: Bit ưu tiên ngắt USART đồng bộ chủ MSSP
1 = Ưu tiên cao
0 = Ưu tiên thấp

bit 2 **CCP1IP**: Bit ưu tiên ngắt của module CCP1
1 = Ưu tiên cao
0 = Ưu tiên thấp

bit 1 **TMR2IP**: Bit ưu tiên ngắt so sánh giữa TMR2 và PR2
1 = Ưu tiên cao
0 = Ưu tiên thấp

bit 0 **TMR1IP**: Bit ưu tiên ngắt tràn do Timer1
1 = Ưu tiên cao
0 = Ưu tiên thấp

- Thanh ghi ưu tiên ngắt 2: IPR2

R/W-1	R/W-1	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
OSCFIP	CMIP	—	EEIP	BCLIP	HLVDIP	TMR3IP	CCP2IP
bit 7							bit 0

bit 7 **OSCFIP:** Bit ưu tiên ngắt do lỗi bộ phát xung (Oscillator Fail)
1 = Ưu tiên cao
0 = Ưu tiên thấp

bit 6 **CMIP:** Bit ưu tiên ngắt từ bộ so sánh (Comparator)
1 = Ưu tiên cao
0 = Ưu tiên thấp

bit 5 Không được định nghĩa

bit 4 **EEIP:** Bit ưu tiên ngắt đọc bộ nhớ EEPROM/Flash
1 = Ưu tiên cao
0 = Ưu tiên thấp

bit 3 **BCLIP:** Bit ưu tiên ngắt do xung đột Bus (Bus Collision)
1 = Ưu tiên cao
0 = Ưu tiên thấp

bit 2 **HLVDIP:** Bit ưu tiên ngắt phát hiện điện áp cao/thấp (High/Low-Voltage Detect)
1 = Ưu tiên cao
0 = Ưu tiên thấp

bit 1 **TMR3IP:** Bit ưu tiên ngắt tràn Timer3 (tràn thanh ghi TMR3)
1 = Ưu tiên cao
0 = Ưu tiên thấp

bit 0 **CCP2IP:** Bit ưu tiên ngắt CCP2
1 = Ưu tiên cao
0 = Ưu tiên thấp

3.3. Khung chương trình sử dụng ngắt.

- Chương trình không sử dụng ưu tiên ngắt:

```
//khai báo thư viện
#include<p18f4520.h>
#include<stdio.h>
#include<delays.h>

//cấu hình
#pragma config      OSC=HS
#pragma config      WDT=OFF
#pragma config      MCLRE=ON
#pragma config      PBADEN=OFF
//khai báo tên chương trình con phục vụ ngắt (CTCPVN)
void ngat_ngoai(void);
//Điểm đặt chương trình phục vụ ngắt có mức ưu tiên cao (vector 0008H)
#pragma       code uu_tien_cao = 0x08
// Nếu là chương trình phục vụ ngắt có mức ưu tiên thấp (vector 0018H):
// #pragma     code uu_tien_thap = 0x18
//Viết 1 chương trình con (có tên tùy chọn) để gọi tới CTCPVN
void ngat_cao(void)
{
 ngat_ngoai(); //gọi đến CTCPVN
}
//Điểm viết CTCPVN
#pragma code
#pragma interrupt ngat_ngoai
void ngat_ngoai(void)
//Noi dung cua CTCPVN
{
```

```
        //Kiểm tra bit cờ của nguồn ngắt thứ nhất, nếu = 1:
        //Xóa cờ của nguồn ngắt thứ nhất
        //Các lệnh xử lý khi có ngắt từ nguồn thứ nhất

        //Kiểm tra bit cờ của nguồn ngắt thứ hai, nếu = 1:
        //Xóa cờ của nguồn ngắt thứ hai
        //Các lệnh xử lý khi có ngắt từ nguồn thứ hai

        //Kiểm tra bit cờ của nguồn ngắt thứ n, nếu = 1:
        //Xóa cờ của nguồn ngắt thứ n
        //Các lệnh xử lý khi có ngắt từ nguồn thứ n
}
//Chương trình chính
void main()
{
//Khởi tạo các thanh ghi liên quan
//Các lệnh khác
}
```

- Chương trình có sử dụng ưu tiên ngắt:

```
//khai báo thư viện
//cấu hình
//khai báo tên CTCPVN
void ngat_uu_tien_cao(void);
void ngat_uu_tien_thap(void);
//Điểm đặt CTCPVN có mức ưu tiên cao UTC, vector 0008H
#pragma       code uu_tien_cao=0x08
void ngat_cao(void)
{
ngat_uu_tien_cao();
}
#pragma code
#pragma       interrupt ngat_uu_tien_cao
void ngat_uu_tien_cao(void)
{
//Kiểm tra bit cờ của nguồn ngắt UTC thứ nhất, nếu = 1.
        {
        //Xóa cờ của nguồn ngắt UTC thứ nhất
        //Các lệnh xử lý khi có ngắt từ nguồn thứ nhất
        }
//Kiểm tra bit cờ của nguồn ngắt UTC thứ hai, nếu = 1.
```

```
        {
        //Xóa cờ của nguồn ngắt UTC thứ hai
        //Các lệnh xử lý khi có ngắt từ nguồn thứ hai
        }
…
//Kiểm tra bit cờ của nguồn ngắt UTC thứ n, nếu = 1.
        {
        //Xóa cờ của nguồn ngắt UTC thứ n
        //Các lệnh xử lý khi có ngắt từ nguồn thứ n
        }
}
```

//Điểm đặt CTCPVN có mức ưu tiên thấp (UTT), vector 0018H

```
#pragma     code uu_tien_thap = 0x18
void ngat_thap(void)
{
ngat_uu_tien_thap();
}
#pragma code
#pragma     interrupt ngat_uu_tien_thap

void ngat_uu_tien_thap(void)
{
//Kiểm tra bit cờ của nguồn ngắt UTT thứ nhất, nếu = 1.
        {
        //Xóa cờ của nguồn ngắt UTT thứ nhất
        //Các lệnh xử lý khi có ngắt từ nguồn thứ nhất
        }
//Kiểm tra bit cờ của nguồn ngắt UTT thứ hai, nếu = 1.
        {
        //Xóa cờ của nguồn ngắt UTT thứ hai
        //Các lệnh xử lý khi có ngắt từ nguồn thứ hai
        }
…
//Kiểm tra bit cờ của nguồn ngắt UTT thứ n, nếu = 1.
        {
        //Xóa cờ của nguồn ngắt UTT thứ n
        //Các lệnh xử lý khi có ngắt từ nguồn thứ n
        }
}
```

//Chuong trinh chinh

```
void main()
{
…
}
```

4. HOẠT ĐỘNG ĐỊNH THỜI

4.1. Timer 0

4.1.1. Các thanh ghi của Timer0

- Thanh ghi điều khiển Timer0: T0CON

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0
bit 7							bit 0

Ghi chú:

R = Cho phép đọc W = Cho phép ghi U = Không sử dụng, đọc bằng '0'

-n = Reset - POR '1' = Được thiết lập '0' = Được xóa -x = Reset không xác định

Bit 7 TMR0ON: Bit điều khiển Bật/Tắt Timer.

1 = Cho phép Timer0

0 = Dừng Timer0

Bit 6 T08BIT: Bit lựa chọn 8-bit /16-bit của Timer0

1 = Timer0 được cấu hình là bộ đếm 8-bit

0 = Timer0 được cấu hình là bộ đếm 16-bit

Bit 5 T0CS: Bit lựa chọn nguồn xung cấp cho Timer0

1 = Nguồn xung từ chân T0CKI

0 = Nguồn xung hệ thống (CLKO)

Bit 4 T0SE: Bit lựa chọn sườn xung đếm cho Timer0

1 = Lựa chọn sườn âm trên chân T0CKI

0 = Lựa chọn sườn dương trên chân T0CKI

Bit 3 PSA: Bit thiết lập bộ chia tần đầu vào

1 = Xung cấp vào Timer0 không qua bộ chia tần.

0 = Xung cấp vào Timer0 qua bộ chia tần(Prescaler).

Bit 2 T0PS<2 :0>: Bit lựa chọn hệ số chia tần

111 = 1:256

110 = 1:128

101 = 1:64

100 = 1:32

011 = 1:16

010 = 1:8

001 = 1:4

000 = 1:2

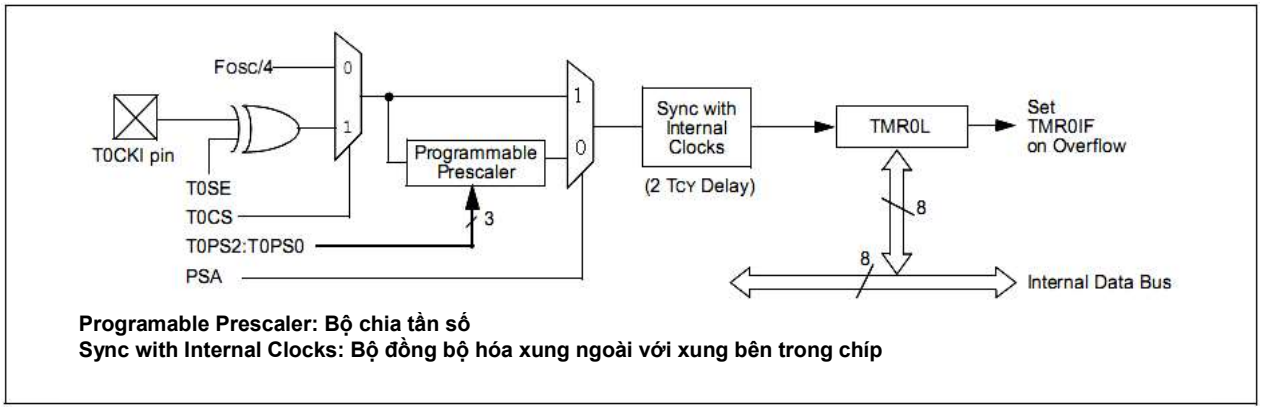
- Thanh ghi chứa byte thấp của Timer0: TMR0L (8 bit, không định địa chỉ bit)
- Thanh ghi chứa byte cao của Timer0: TMR0H (8 bit, không định địa chỉ bit)
- Thanh ghi điều khiển ngắt : INTCON (xem phần ngắt và xử lý ngắt)

Các thanh ghi liên quan tới Timer0 :

Tên	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
TMR0L	Thanh ghi chứa giá trị đếm byte thấp của Timer0							
TMR0H	Thanh ghi chứa giá trị đếm byte cao của Timer0							
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
T0CON	TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0
TRISA	RA7(1)	RA6(1)	RA5	RA4	RA3	RA2	RA1	RA0

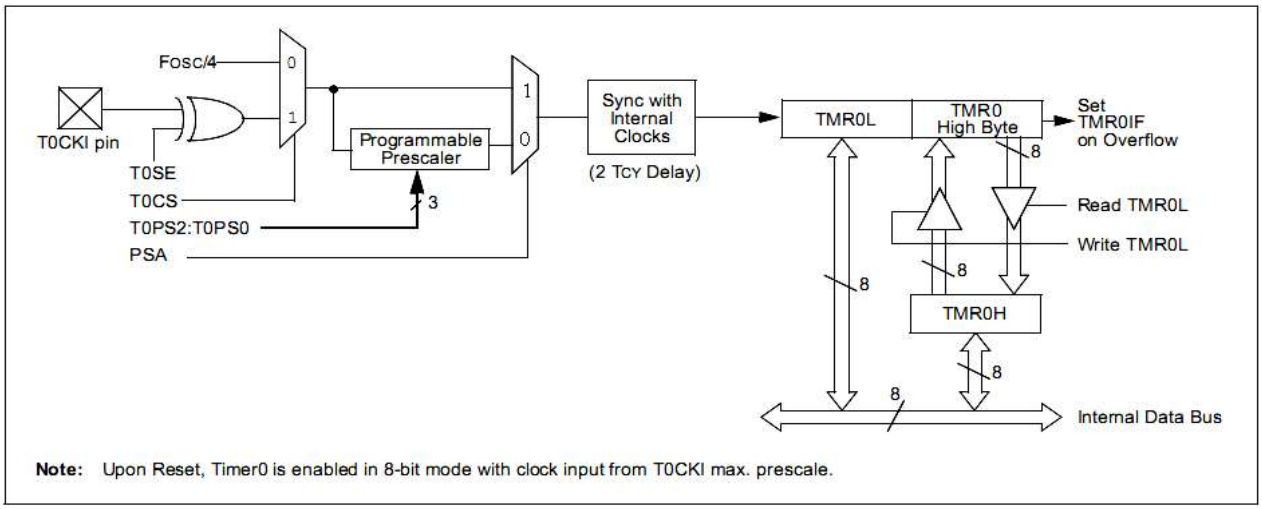
4.1.2. Chế độ hoạt động của Timer0

Chế độ 8 bit



Programable Prescaler: Bộ chia tần số
Sync with Internal Clocks: Bộ đồng bộ hóa xung ngoài với xung bên trong chíp

Chế độ 16 bit



4.1.3. Ngắt Timer0.

Ngắt Timer0 xảy ra khi Timer0 tràn. Ở chế độ đếm 8 bit sự kiện tràn xảy ra khi có sự chuyển số đếm từ FFH sang 00H và FFFFH sang 0000H ở chế độ 16 bit. Khi tràn cờ ngắt tương ứng của Timer0 (TMR0IF- bit 2 thanh ghi INTCON) được thiết lập.

Ngắt Timer0 được cho phép ngắt bởi bit TMR0IE (bit 5 thanh ghi INTCON).

Mức ưu tiên ngắt Timer0 được đặt bởi bit TMR0IP (bit 2, thanh ghi INTCON2).

4.2. Timer1

4.2.1. Các thanh ghi của Timer1

- Thanh ghi điều khiển Timer1: T1CON

R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON
bit 7							bit 0

bit 7 **RD16**: Bit lựa chọn chế độ ghi/đọc Timer1
1 = Ghi/đọc 1 lần 16 bit.
0 = Ghi/đọc 2 lần mỗi lần 8 bit.

bit 6 **T1RUN**: Bit cho phép hệ thống lấy xung từ Timer1
1 = Hệ thống hoạt động bằng nguồn xung cấp từ Timer1
0 = Hệ thống hoạt động bằng nguồn xung khác

bit 5-4 **T1CKPS1:T1CKPS0**: Các bit đặt hệ số chia tần số của xung cấp cho Timer1
11 = Hệ số chia là 1:8
10 = Hệ số chia là 1:4
01 = Hệ số chia là 1:2
00 = Hệ số chia là 1:1

bit 3 **T1OSCEN**: Bit cho phép/cấm chức năng phát xung cho hệ thống
1 = Cho phép
0 = Cấm

bit 2 **T1SYNC:** Bit lựa chọn sự đồng bộ giữa xung ngoài cấp cho Timer1 và xung trên chip.

Khi bit TMR1CS = 1:

1 = Không đồng bộ

0 = Đồng bộ xung ngoài với xung trên chip

Khi bit TMR1CS = 0:

Bit T1SYNC không có giá trị. (khi đó Timer1 sử dụng nguồn xung nội).

bit 1 **TMR1CS:** Bit lựa chọn nguồn xung cấp cho Timer1

1 = Timer1 được cấp xung từ ngoài qua chân RC0/T1OSO/T13CKI

0 = Timer1 được cấp xung nội (tần số bằng FOSC/4)

bit 0 **TMR1ON**: Bit điều khiển hoạt động của Timer1

1 = Timer1 hoạt động

0 = Dừng Timer1

- Thanh ghi chứa giá tri đếm byte thấp của Timer1: TMR1L

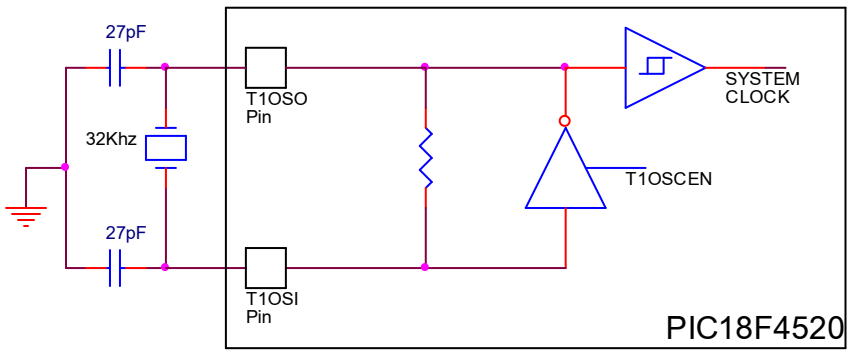
- Thanh ghi chứa giá tri đếm byte cao của Timer1: TMR1H

Các thanh ghi liên quan đến Timer1:

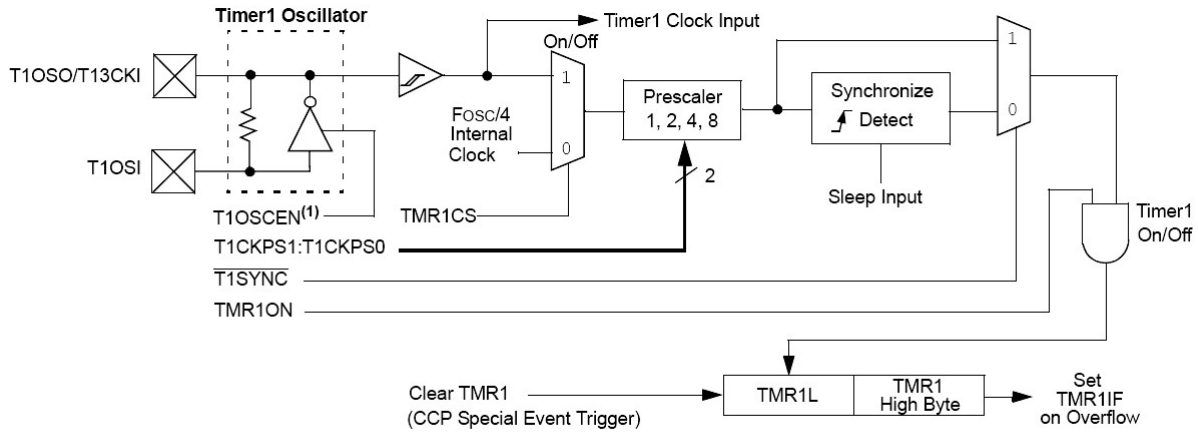
Tên	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
PIR1	PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
PIE1	PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
IPR1	PSPIP(1)	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP
TMR1L	Thanh ghi chứa giá tri đếm byte thấp của Timer1							
TMR1H	Thanh ghi chứa giá tri đếm byte cao của Timer1							
T1CON	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON

4.2.2. Chế độ hoạt động của Timer1

Chế độ phát xung cho toàn hệ thống.

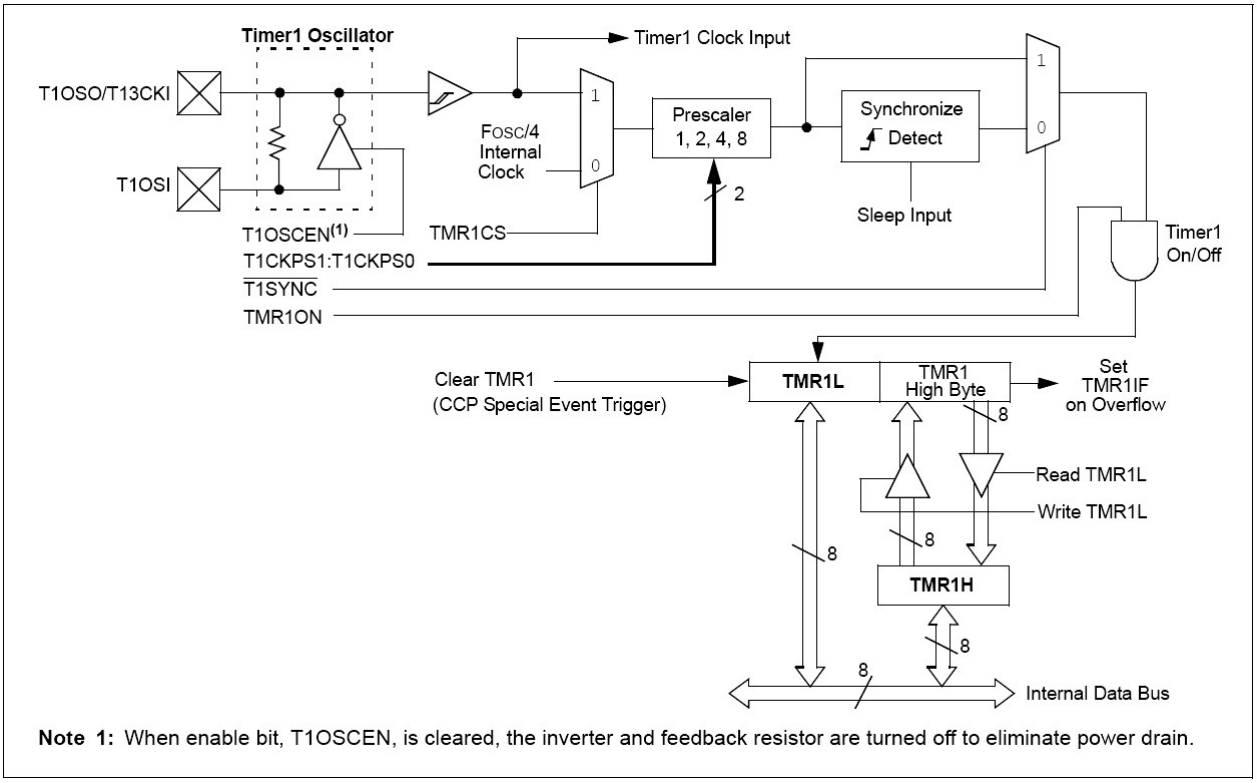


Chế độ ghi/đọc 2 lần 8bit



Note 1: When enable bit, T1OSCEN, is cleared, the inverter and feedback resistor are turned off to eliminate power drain.

Chế độ ghi/đọc 1 lần 16bit



4.2.3. Ngắt Timer1.

Ngắt Timer1 được cho phép bởi bit TMR1IE (thanh ghi PIE1), được đặt mức ưu tiên cao/thấp bởi bit TMR1IP (thanh ghi IPR1). Ngắt Timer1 xảy ra khi Timer1 tràn, khi đó cờ ngắt tràn TMR1IF (thanh ghi PIR1) được đặt bằng 1.

4.3. Timer2

4.3.1. Các thanh ghi của Timer2.

- Thanh ghi điều khiển Timer 2: T2CON

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit 7							bit 0

bit 7 Không sử dụng

bit 6-3 **T2OUTPS3:T2OUTPS0**: Các bit đặt hệ số chia của bộ Postscaler
0000 = Hệ số chia tần là: 1
0001 = Hệ số chia tần là: 2
•
•
•
1111 = Hệ số chia tần là: 16

bit 2 **TMR2ON**: Bit điều khiển Timer2
1 = Timer2 hoạt động
0 = Timer2 dừng

bit 1-0 **T2CKPS1:T2CKPS0**: Các bit đặt hệ số chia của bộ Prescaler
00 = Hệ số chia tần là: 1
01 = Hệ số chia tần là: 4
1x = Hệ số chia tần là: 16

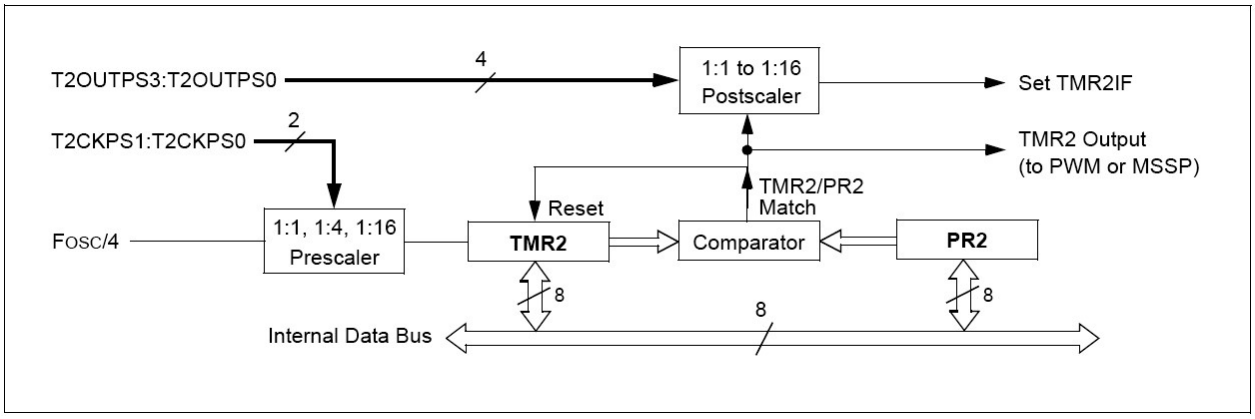
- Thanh ghi chứa giá tri đếm của Timer2: TMR2.

- Thanh ghi chu kỳ của Timer2(Period register): PR2.

Các thanh ghi liên quan đến Timer2:

Tên	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
PIR1	PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
PIE1	PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
IPR1	PSPIP(1)	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP
TMR2	Thanh ghi chứa giá trị đếm của Timer2							
T2CON	—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0
PR2	Thanh ghi chu kỳ của Timer2							

4.3.2. Chế độ hoạt động của Timer2.



4.3.3. Ngắt Timer2.

Ngắt Timer2 được cho phép bởi bit TMR2IE (thanh ghi PIE1), được đặt mức ưu tiên cao/thấp bởi bit TMR2IP (thanh ghi IPR1). Ngắt Timer2 xảy ra khi số lần "so sánh bằng" giữa 2 thanh ghi TMR2 và thanh ghi PR2 bằng với hệ số chia đặt trước của bộ chia tần liền sau, khi đó cờ ngắt tràn TMR2IF (thanh ghi PIR1) được đặt bằng 1.

4.4. Timer3

4.4.1. Các thanh ghi của Timer3

- Thanh ghi điều khiển Timer3: T3CON

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	T3SYNC	TMR3CS	TMR3ON

bit 7	bit 0

bit 7 **RD16**: Bit lựa chọn chế độ ghi/đọc Timer3
1 = Ghi/đọc 1 lần 16 bit.
0 = Ghi/đọc 2 lần mỗi lần 8 bit.

bit 6,3 **T3CCP2:T3CCP1:** Bit lựa chọn Timer1, Timer3 làm nguồn xung cho khối CCP
1x = Timer3 làm nguồn xung cho module CCP1 và CCP2
01 = Timer3 làm nguồn xung cho module CCP2,
Timer1 làm nguồn xung cho module CCP1
00 = Timer1 làm nguồn xung module CCP1 và CCP2

bit 5,4 **T3CKPS1:T3CKPS0:** Bit lựa chọn hệ số cho bộ chia tần số (Prescaler)
11 = Hệ số chia là 1:8
10 = Hệ số chia là 1:4
01 = Hệ số chia là 1:2
00 = Hệ số chia là 1:1

bit 2 **T3SYNC:** Bit lựa chọn sự đồng bộ giữa xung ngoài cấp cho Timer3 và xung trên chip.
Khi bit TMR3CS = 1:
1 = Không đồng bộ
0 = Đồng bộ xung ngoài với xung trên chip
Khi bit TMR3CS = 0: Bit T3SYNC không có giá trị. (khi đó Timer1 sử dụng nguồn xung nội).

bit 1 **TMR3CS:** Bit lựa chọn nguồn xung cấp cho Timer3
1 = Timer3 được cấp xung từ ngoài qua chân RC0/T1OSO/T13CKI
0 = Timer3 được cấp xung nội (tần số bằng FOSC/4)

bit 0 **TMR3ON**: Bit điều khiển hoạt động của Timer3
1 = Timer3 hoạt động
0 = Dừng Timer3

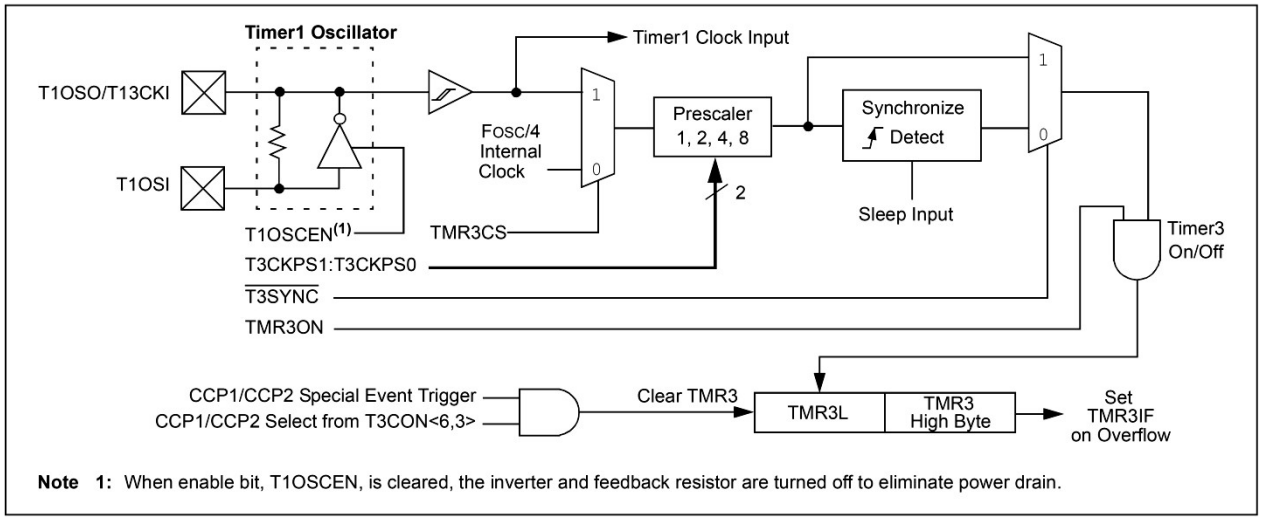
- Thanh ghi chứa giá tri đếm byte thấp của Timer3: TMR3L

- Thanh ghi chứa giá tri đếm byte cao của Timer3: TMR3H

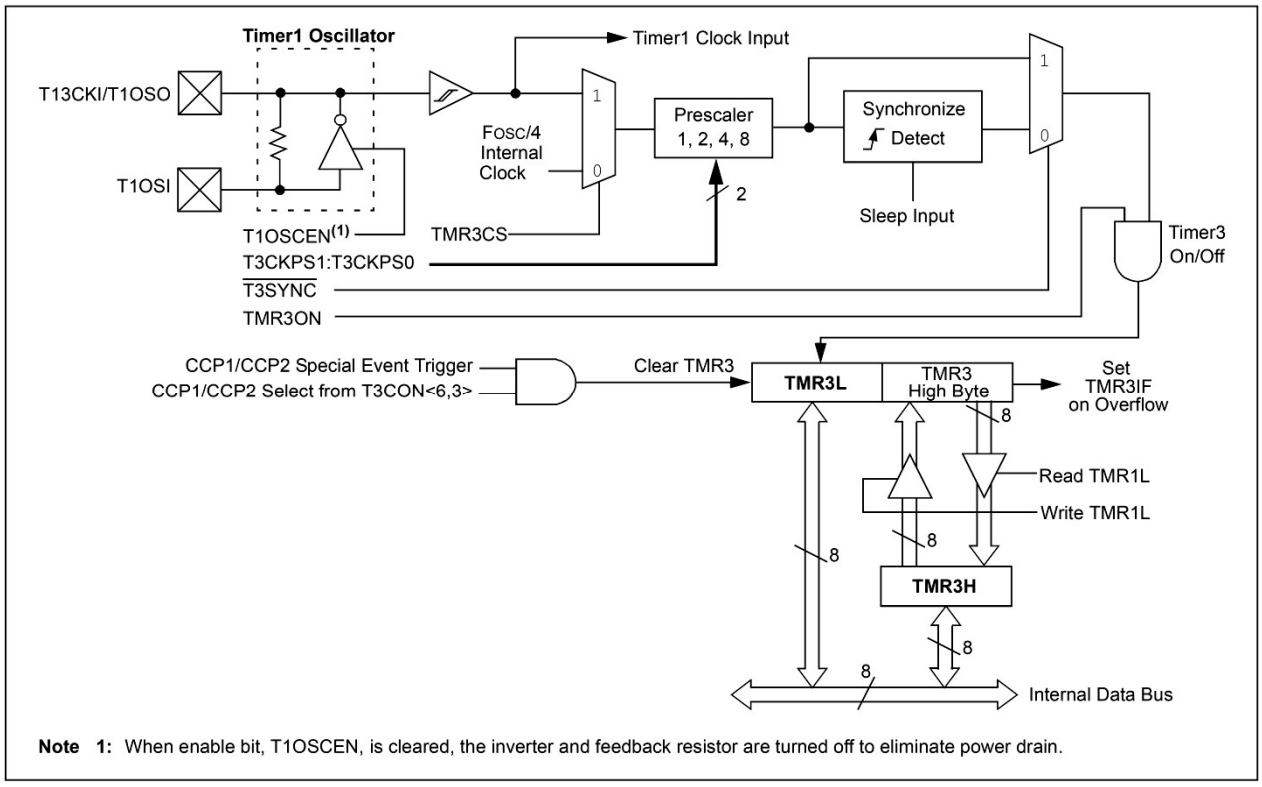
Các thanh ghi liên quan đến Timer3:

Tên	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
PIR2	OSCFIF	CMIF	—	EEIF	BCLIF	HLVDIF	TMR3IF	CCP2IF
PIE2	OSCFIE	CMIE	—	EEIE	BCLIE	HLVDIE	TMR3IE	CCP2IE
IPR2	OSCFIP	CMIP	—	EEIP	BCLIP	HLVDIP	TMR3IP	CCP2IP
TMR3L	Thanh ghi chứa giá tri đếm byte thấp của Timer3							
TMR3H	Thanh ghi chứa giá tri đếm byte cao của Timer3							
T1CON	RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON
T3CON	RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	T3SYNC	TMR3CS	TMR3ON

4.4.2. Chế độ hoạt động của Timer3



Hoạt động của Timer3 ở chế độ ghi/đọc 2 lần 8bit



Hoạt động của Timer3 ở chế độ ghi/đọc 1 lần 16bit

4.4.3. Ngắt Timer3

Cặp thanh ghi chứa giá trị đếm của Timer3 (TMR3H:TMR3L) đếm tăng từ 0000h đến FFFFh, đếm tiếp một xung nó sẽ tràn và quay trở về giá trị 0000h. Khi tràn cờ ngắt TMR3IF(PIR2<1>) sẽ được thiếp lập. Ngắt Timer3 được cho phép khi thiết lập bit TMR3IE (PIE2<1>), cấm khi xóa bit TMR3IE.

4.4. Các hàm trong thư viện timers.h

4.4.1. Các hàm của timer0:

Hàm CloseTimer0

Nguyên mẫu : void CloseTimer0(void);

Chức năng : Hàm được sử dụng để cấm hoạt động của timer0

Hàm OpenTimer0

Nguyên mẫu : void OpenTimer0(unsigned char ***config***);

Chức năng : Hàm được sử dụng để thiết lập các tham số cho timer0

Các đối số bao gồm:

TIMER_INT_ON TIMER_INT_OFF	Cho phép ngắt (Interrupt enabled) Không cho phép ngắt (Interrupt disabled)
T0_SOURCE_EXT T0_SOURCE_INT	Dùng nguồn xung từ ngoài (I/O pin) Dùng nguồn dao động nội (T_{OSC})
T0_EDGE_FALL T0_EDGE_RISE	Đếm khi có sườn xuống (External clock on falling edge) Đếm khi có sườn lên (External clock on rising edge)
T0_8BIT T0_16BIT	Chế độ 8bit Chế độ 16bit
T0_PS_1_1 T0_PS_1_2 T0_PS_1_4 T0_PS_1_8 T0_PS_1_16 T0_PS_1_32 T0_PS_1_64 T0_PS_1_128 T0_PS_1_256	Hệ số chia tần trước 1:1 Hệ số chia tần trước 1:2 Hệ số chia tần trước 1:4 Hệ số chia tần trước 1:8 Hệ số chia tần trước 1:16 Hệ số chia tần trước 1:32 Hệ số chia tần trước 1:64 Hệ số chia tần trước 1:128 Hệ số chia tần trước 1:256

Các đối số được sử dụng kết hợp với nhau bằng toán từ "&", ví dụ:

```
OpenTimer0( TIMER_INT_OFF &
            T0_8BIT &
            T0_SOURCE_INT &
            T0_PS_1_32 );
```

Hàm ReadTimer0

Nguyên mẫu : unsigned int ReadTimer0(void);

Chức năng : Hàm được sử dụng để đọc giá trị hiện thời của timer0

Hàm WriteTimer0

Nguyên mẫu : void WriteTimer0(unsigned int ***timer_value***)

Chức năng : Hàm được sử dụng để đọc giá trị hiện thời của timer0

Tham số ***timer_value*** là giá trị cần ghi cho timer, ví dụ: **WriteTimer0(10)**;

4.4.2. Các hàm của timer1:

Hàm CloseTimer1

Nguyên mẫu : void CloseTimer1(void);

Chức năng : Hàm được sử dụng để cấm hoạt động của timer1

Hàm OpenTimer1

Nguyên mẫu : void OpenTimer1(unsigned char ***config***);

Chức năng : Hàm được sử dụng để thiết lập các tham số cho Timer1

Một số đối số:

TIMER_INT_ON TIMER_INT_OFF	Cho phép ngắt (Interrupt enabled) Không cho phép ngắt (Interrupt disabled)
T1_8BIT_RW T1_16BIT_RW	Chế độ ghi/đọc 8bit Chế độ ghi/đọc 16bit
T1_SOURCE_EXT T1_SOURCE_INT	Dùng nguồn xung từ ngoài (I/O pin) Dùng nguồn dao động nội (T_{OSC})
T1_PS_1_1 T1_PS_1_2 T1_PS_1_4 T1_PS_1_8	Hệ số chia tần trước 1:1 Hệ số chia tần trước 1:2 Hệ số chia tần trước 1:4 Hệ số chia tần trước 1:8
T1_OSC1EN_ON T1_OSC1EN_OFF	Sử dụng timer1 như một bộ phát xung cho hệ thống Không sử dụng timer1 như một bộ phát xung cho hệ thống

Hàm ReadTimer1

Nguyên mẫu : unsigned int ReadTimer1(void);

Chức năng : Hàm được sử dụng để đọc giá trị hiện thời của Timer1

Hàm WriteTimer1

Nguyên mẫu : void WriteTimer1(unsigned int ***timer_value***)

Chức năng : Hàm được sử dụng để đọc giá trị hiện thời của Timer1

4.4.3. Các hàm của Timer2:

Hàm CloseTimer2

Nguyên mẫu : void CloseTimer2(void);

Chức năng : Hàm được sử dụng để cấm hoạt động của Timer2

Hàm OpenTimer2

Nguyên mẫu : void OpenTimer2(unsigned char ***config***);

Chức năng : Hàm được sử dụng để thiết lập các tham số cho Timer2

Một số đối số:

TIMER_INT_ON TIMER_INT_OFF	Cho phép ngắt (Interrupt enabled) Không cho phép ngắt (Interrupt disabled)
T2_PS_1_1 T2_PS_1_4 T2_PS_1_16	Hệ số chia tần trước 1:1 Hệ số chia tần trước 1:4 Hệ số chia tần trước 1:16
T2_POST_1_1 T2_POST_1_2 … T2_POST_1_16	Hệ số chia tần sau 1:1 Hệ số chia tần sau 1:2 … Hệ số chia tần sau 1:16

Hàm ReadTimer2

Nguyên mẫu : unsigned int ReadTimer2(void);

Chức năng : Hàm được sử dụng để đọc giá trị hiện thời của Timer2

Hàm WriteTimer2

Nguyên mẫu : void WriteTimer2(unsigned int ***timer_value***)

Chức năng : Hàm được sử dụng để đọc giá trị hiện thời của Timer2

4.4.4. Các hàm của Timer3:

Hàm CloseTimer3

Nguyên mẫu : void CloseTimer3(void);

Chức năng : Hàm được sử dụng để cấm hoạt động của Timer3

Hàm OpenTimer3

Nguyên mẫu : void OpenTimer3(unsigned char ***config***);

Chức năng : Hàm được sử dụng để thiết lập các tham số cho Timer3

Một số đối số:

TIMER_INT_ON TIMER_INT_OFF	Cho phép ngắt (Interrupt enabled) Không cho phép ngắt (Interrupt disabled)
T3_SOURCE_EXT T3_SOURCE_INT	Dùng nguồn xung từ ngoài (I/O pin) Dùng nguồn dao động nội (T_{OSC})
T3_8BIT_RW T3_16BIT_RW	Chế độ ghi/đọc 8bit Chế độ ghi/đọc 16bit
T3_PS_1_1 T3_PS_1_4 T3_PS_1_8 T3_PS_1_16	Hệ số chia tần trước 1:1 Hệ số chia tần trước 1:4 Hệ số chia tần trước 1:8 Hệ số chia tần trước 1:16

Hàm ReadTimer3

Nguyên mẫu : unsigned int ReadTimer3(void);

Chức năng : Hàm được sử dụng để đọc giá trị hiện thời của Timer3

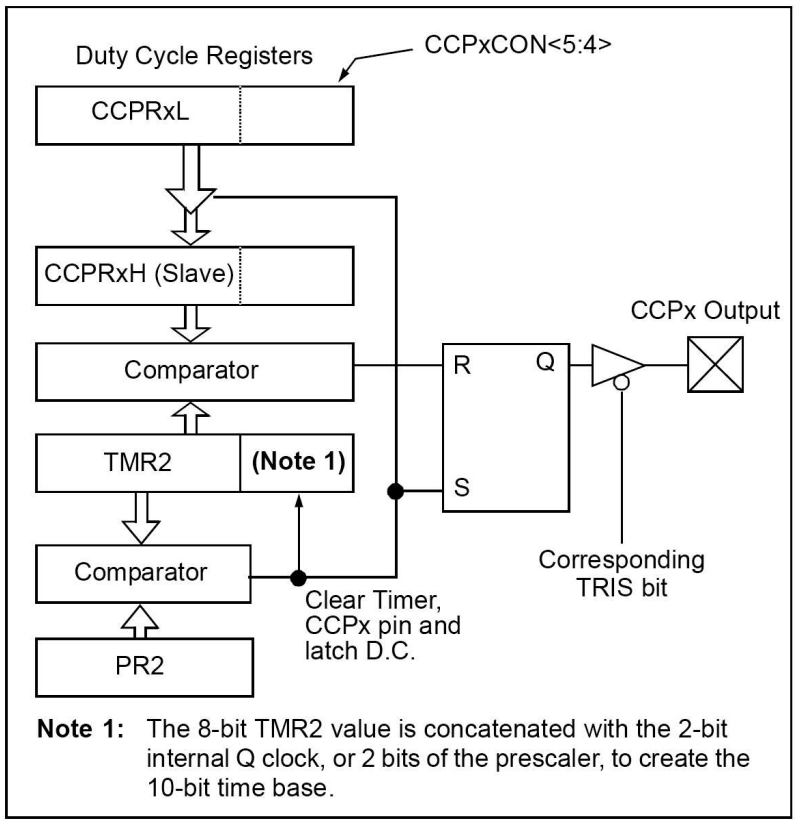
Hàm WriteTimer3

Nguyên mẫu : void WriteTimer3(unsigned int ***timer_value***)

Chức năng : Hàm được sử dụng để đọc giá trị hiện thời của Timer3

5. ĐIỀU CHẾ ĐỘ RỘNG XUNG – PWM

5.1. Sơ đồ khối bộ PWM



Sơ đồ khối của CCP ở chế độ PWM.

5.2. Các thanh ghi liên quan

- Thanh ghi điều khiển CCPx: CCPxCON

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	DCxB1	DCxB0	CCPxM3	CCPxM2	CCPxM1	CCPxM0
bit 7							bit 0

bit 7-6 **Không được sử dụng**: đọc sẽ được ‘0’

bit 5-4 **DCxB1:DCxB0:** Bit lựa chọn độ rộng xung 1:0 của chế độ PWM.

Capture mode:
Không sử dụng.
Compare mode:
Không sử dụng.
PWM mode:
Là 2 bit thấp DCxB1: DCxB0 của thanh ghi lựa chọn độ rộng xung cho PWM.

bit 3-0 **CCPxM3:CCPxM0:** Bit lựa chọn chế độ hoạt động cho bộ CCPx

0000 = Cấm CCPx hoạt động.
0001: 1011 = Không được sử dụng ở chế độ này.

11xx = Chế độ PWM.

- Thanh ghi chu kỳ (Period) : PR2

PR2 là thanh ghi 8 bit của bộ Timer2, thanh ghi này được sử dụng để tạo chu kỳ cho xung PWM.

- Thanh ghi độ rộng xung (Dyty Cycle) : CCPRx

CCPRx là thanh ghi thanh ghi 8 bit của CCPx, thanh ghi này chứa 8 bit cao DCxB9:DCxB2 sử dụng để tạo độ rộng xung(2 bit thấp trong thanh CCPxCON).

Các thanh ghi liên quan đến chế độ PWM của bộ CCPx:

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
RCON	IPEN	SBOREN	—	RI	TO	PD	POR	BOR
PIR1	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
PIE1	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
IPR1	PSPIP	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP
TRISB	Thanh ghi hướng dữ liệu PORTB							
TRISC	Thanh ghi hướng dữ liệu PORTC							
TMR2	Thanh ghi đếm của Timer2							
PR2	Thanh ghi chu kỳ của Timer2							
T2CON	—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0
CCPR1L	Thanh ghi Capture/Compare/PWM 1 byte thấp							
CCPR1H	Thanh ghi Capture/Compare/PWM 1 byte cao							
CCP1CON	P1M1(1)	P1M0(1)	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0
CCPR2L	Thanh ghi Capture/Compare/PWM 2 byte thấp							
CCPR2H	Thanh ghi Capture/Compare/PWM 2 byte cao							
CCP2CON	—	—	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0

5.3. Sử dụng các bộ PWM

5.3.1. Thiết lập chu kỳ

Công thức tính chu kỳ:

$PWM\_Period = (PR2 + 1) * 4 * T_{OSC} * prescaler)$

Trong đó:

PWM_Period: Chu kỳ xung PWM cần tạo

T_{OSC}: là chu kỳ của xung cấp cho hệ thống (thường là xung tạo bởi bộ dao động thạch anh)

prescaler: Hệ số chia tần của Timer 2.

5.3.2. Thiết lập độ rộng nửa chu kỳ dương

Công thức tính độ rộng nửa chu kỳ dương:

PWM_Duty_Cycle= (CCPRxL:CCPxCON<5:4>) * T_{OSC} * prescaler

Trong đó:

PWM_Duty_Cycle: Độ rộng của nửa chu kỳ dương.

CCPRxL:CCPxCON<5:4> : 10 bit chứa giá trị của bộ đếm (CCPRxL chứa 8 bit cao và CCPxCON <5:4> chứa 2 bit thấp).

5.4. Các hàm trong thư viện pwm.h

• Hàm : ClosePWM1, ClosePWM2

Nguyên mẫu : void ClosePWM1(void);
void ClosePWM2(void);

Chức năng : Hàm được sử dụng để cấm hoạt động của kênh PWM tương ứng (kênh 1 hoặc kênh 2).

• Hàm : OpenPWM1, OpenPWM2

Nguyên mẫu : void OpenPWM1(char period);
void OpenPWM2(char period);

Chức năng: Hàm được sử dụng để khởi tạo các bộ PWM và thiết lập chu kỳ của xung cần tạo

Đối số : period
Giá trị period nằm trong khoảng từ 0 đến 255. period được tính theo công thức sau:
period = PWM_period/(4 *TOSC*prescaler)-1
Trong đó, PWM period là chu kỳ của xung cần tạo; T_{OSC} là chu kỳ của xung cấp cho hệ thống (thường là xung tạo bởi bộ dao động thạch anh); prescaler là hệ số chia tần trước của timer2.

• Hàm : SetDCPWM1, SetDCPWM2

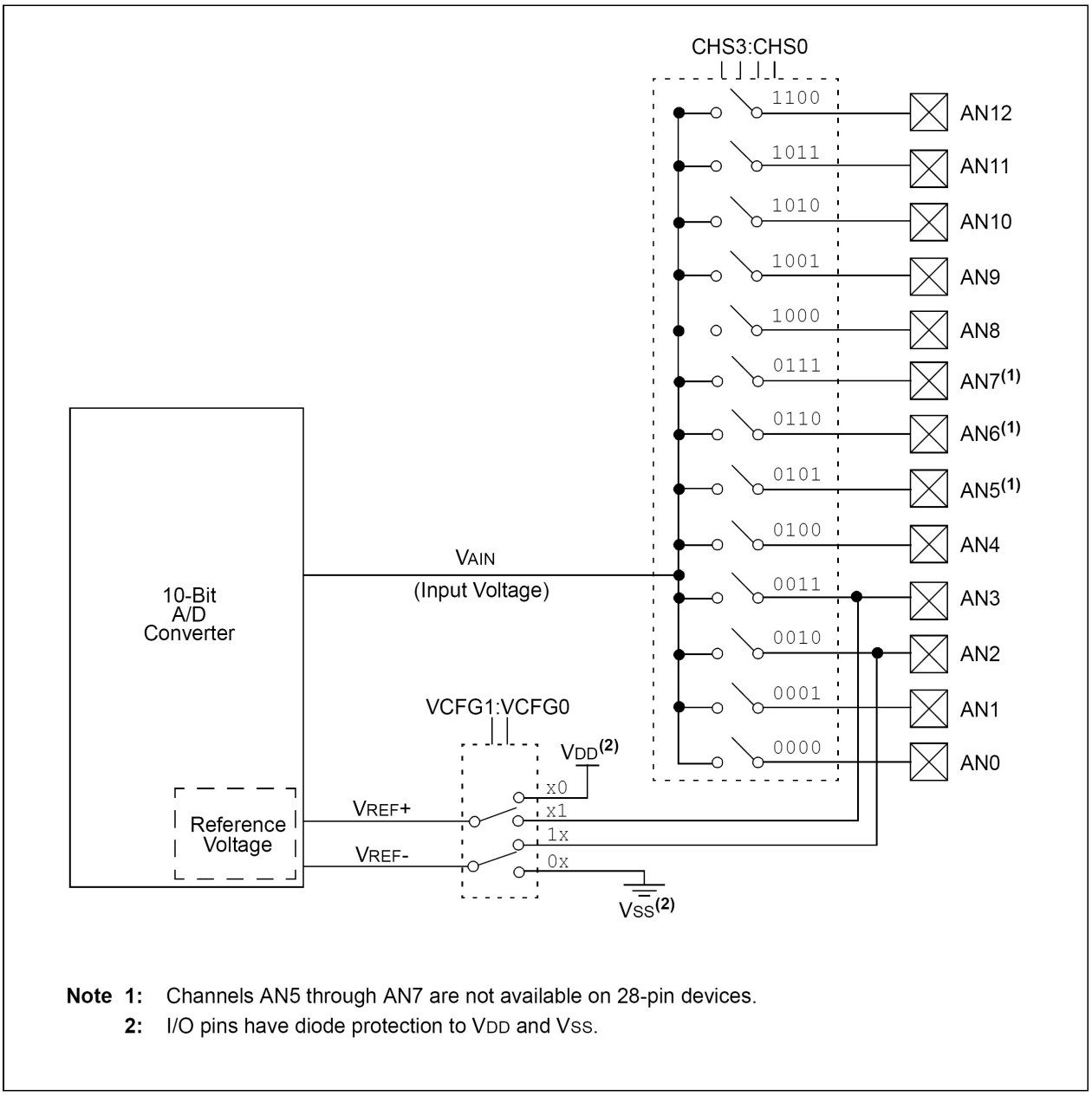
Nguyên mẫu : void SetDCPWM1(unsigned int dutycycle);
void SetDCPWM2(unsigned int dutycycle);

Chức năng : Thiết lập độ rộng của nửa chu kỳ dương cho kênh PWM tương ứng.

Đối số : dutycycle là giá trị được nạp vào 10 bit CCPRxL:CCPxCON<5:4> để thiết lập độ rộng xung PWM (xem phần 5.3). dutycycle có giá trị nằm trong khoảng từ 0 đến 1023 và được tính như sau:
dutycycle = PWM_Duty_Cycle/(T_{OSC}*prescaler)
Trong đó, PWM_Duty_Cycle là độ rộng của nửa chu kỳ dương của xung cần tạo

6. BỘ CHUYỂN ĐỔI TƯƠNG TỰ - SỐ (ADC)

6.1. Sơ đồ khối ADC trên PIC18F4520



Note 1: Channels AN5 through AN7 are not available on 28-pin devices.
2: I/O pins have diode protection to VDD and VSS.

6.2. Các thanh ghi liên quan

- Thanh ghi điều khiển A/D 0: ADCON0

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON
bit 7							bit 0

bit 7-6 **Không được định nghĩa:** Đọc sẽ được "0"

bit 5-2 **CHS<3:0>:** Bit lựa chọn kênh đầu vào tương tự

0000 = Kênh 0 (AN0)
0001 = Kênh 1 (AN1)
0010 = Kênh 2 (AN2)
0011 = Kênh 3 (AN3)
0100 = Kênh 4 (AN4)
0101 = Kênh 5 (AN5)
0110 = Kênh 6 (AN6)
0111 = Kênh 7 (AN7)
1000 = Kênh 8 (AN8)
1001 = Kênh 9 (AN9)
1010 = Kênh 10 (AN10)
1011 = Kênh 11 (AN11)
1100 = Kênh 12 (AN12)
1101 = Không được sử dụng
1110 = Không được sử dụng
1111 = Không được sử dụng

bit 1 **GO/DONE:** Bit trạng thái biến đổi A/D

Khi ADON = 1:

1 = Bắt đầu chuyển đổi A/D
0 = Bộ A/D ở trạng thái chờ

Bit 0 **ADON:** Bit cho phép chuyển đổi A/D

1 = Cho phép chuyển đổi A/D
0 = Không cho phép

- Thanh ghi điều khiển A/D 1: ADCON1

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-q(1)	R/W-q(1)	R/W-q(1)
—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

Bit 7-6 **Không được định nghĩa:** Đọc sẽ được "0"

Bit 5 **VCFG1:** Bit cấu hình điện áp tham chiếu mức cao (V_{REF-})

1 = V_{REF-} (AN2)
0 = V_{SS}

Bit 4 **VCFG0:** Bit cấu hình điện áp tham chiếu mức thấp (V_{REF+})

1 = V_{REF+} (AN3)
0 = V_{DD}

Bit 3-0 **PCFG<3:0>:** Bit cấu hình PORT vào/ra, các bit này được sử dụng để lựa chọn các chân từ AN0 đến AN12 là đầu vào/ra số (D) và đầu vào tương tự (A).

PCFG3: PCFG0	AN12	AN11	AN10	AN9	AN8	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0
0000	A	A	A	A	A	A	A	A	A	A	A	A	A
0001	A	A	A	A	A	A	A	A	A	A	A	A	A
0010	A	A	A	A	A	A	A	A	A	A	A	A	A
0011	D	A	A	A	A	A	A	A	A	A	A	A	A
0100	D	D	A	A	A	A	A	A	A	A	A	A	A
0101	D	D	D	A	A	A	A	A	A	A	A	A	A
0110	D	D	D	D	A	A	A	A	A	A	A	A	A
0111	D	D	D	D	D	A	A	A	A	A	A	A	A
1000	D	D	D	D	D	D	A	A	A	A	A	A	A
1001	D	D	D	D	D	D	D	A	A	A	A	A	A
1010	D	D	D	D	D	D	D	D	A	A	A	A	A
1011	D	D	D	D	D	D	D	D	D	A	A	A	A
1100	D	D	D	D	D	D	D	D	D	D	A	A	A
1101	D	D	D	D	D	D	D	D	D	D	D	A	A
1110	D	D	D	D	D	D	D	D	D	D	D	D	A
1111	D	D	D	D	D	D	D	D	D	D	D	D	D

A = đầu vào tương tự D = vào/ra số

- Thanh ghi điều khiển A/D 2: ADCON2

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0
bit 7							bit 0

bit 7 **ADFM:** Bit lựa chọn định dạng nhập kết quả vào thanh ghi chứa

1 = Chứa trong 10 bit thấp của cặp thanh ghi ADRESH và ADRESL.
0 = Chứa trong 10 cao.

bit 6 **Không được định nghĩa:** Khi đọc sẽ được "0"

bit 5-3 **ACQT<2:0>:** Bit lựa chọn thời gian thu nhận A/D

111 = 20 TAD
110 = 16 TAD
101 = 12 TAD
100 = 8 TAD
011 = 6 TAD
010 = 4 TAD
001 = 2 TAD
000 = 0 TAD (Khi sử dụng bộ dao động RC nội)

bit 2-0 **ADCS<2:0>:** Bit lựa chọn tần số xung cấp cho bộ biến đổi A/D

111 = F_{RC} (nguồn xung từ bộ dao động RC nội)
110 = $F_{OSC}/64$
101 = $F_{OSC}/16$
100 = $F_{OSC}/4$
011 = F_{RC} (nguồn xung từ bộ dao động RC nội)
010 = $F_{OSC}/32$
001 = $F_{OSC}/8$
000 = $F_{OSC}/2$

- Tổng hợp các thanh ghi liên quan đến hoạt động chuyển đổi A/D

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
PIR1	PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
PIE1	PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
IPR1	PSPIP(1)	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP
PIR2	OSCFIF	CMIF	—	EEIF	BCLIF	HLVDIF	TMR3IF	CCP2IF
PIE2	OSCFIE	CMIE	—	EEIE	BCLIE	HLVDIE	TMR3IE	CCP2IE
IPR2	OSCFIP	CMIP	—	EEIP	BCLIP	HLVDIP	TMR3IP	CCP2IP
ADRESH	A/D Result Register High Byte							
ADRESL	A/D Result Register Low Byte							
ADCON0	—	—	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON
ADCON1	—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0
ADCON2	ADFM	—	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0
PORTA	RA7(2)	RA6(2)	RA5	RA4	RA3	RA2	RA1	RA0
TRISA	TRISA7(2)	TRISA6(2)	PORTA Data Direction Register					
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
TRISB	PORTB Data Direction Register							
LATB	PORTB Data Latch Register (Read and Write to Data Latch)							
PORTE(4)	—	—	—	—	RE3(3)	RE2	RE1	RE0
TRISE(4)	IBF	OBF	IBOV	PSPMODE	—	TRISE2	TRISE1	TRISE0
LATE(4)	—	—	—	—	—	PORTE Data Latch Register		

Ghi chú:

— Bit không sử dụng, đọc sẽ được ‘0’ Ô tô đậm không được sử dụng ở A/D

6.4. Các bước lập trình chuyển đổi A/D

Bước 1. Cấu hình cho bộ chuyển đổi ADC:

• Cấu hình chân đầu vào tương tự cho Module ADC, lựa chọn chân điện áp tham chiếu và dùng thanh ghi TRIS để cấu hình các chân được chọn có chiều vào.

• Lựa chọn 1 trong các kênh đầu vào A/D (sử dụng thanh ghi ADCON0).

• Lựa chọn thời gian thu nhận (ADCON2).

• Lựa chọn nguồn xung cấp cho bộ chuyển đổi A/D (ADCON2).

• Cho phép bộ chuyển đổi A/D (ADON=1).

Bước 2. Cấu hình ngắt cho bộ chuyển đổi A/D (nếu sử dụng):

• Xóa bit ADIF về “0”

• Thiết lập bit ADIE bằng “1”

• Thiết lập bit GIE bằng “1”

Bước 3. Đợi cho quá thu nhận hoàn tất (nếu yêu cầu)

Bước 4. Bắt đầu chuyển đổi A/D:

• Đặt bit GO/DONE bằng “1” (ADCON0<1>).

Bước 5. Chờ cho tới khi bộ chuyển đổi A/D biến đổi xong, bằng cách:

• Chờ tới khi bit GO/DONE được xóa về “0” hoặc

• Chờ tới khi xảy ra ngắt ở bộ biến đổi A/D.

Bước 6. Đọc kết quả từ thanh ghi (ADRESH:ADRESL) và xóa cờ Ngắt ADIF nếu sử dụng ngắt ADC.

Lặp lại từ bước 1 nếu muốn quá trình biến đổi AD diễn ra liên tục.

Chú ý: Thời gian chuyển đổi cho mỗi bit được định nghĩa là TAD. Cần phải chờ thời gian tối thiểu là 2 TAD trước khi bắt đầu thực hiện việc chuyển đổi tiếp theo.

6.5. Các hàm trong thư viện adc.h

Hàm	Mô tả
BusyADC	Hàm báo bận.
CloseADC	Cấm hoạt động chuyển động A/D.
ConvertADC	Bắt đầu quá trình chuyển đổi A/D.
OpenADC	Cấu hình cho bộ chuyển đổi A/D.
ReadADC	Đọc giá trị trả về của chuyển đổi A/D.
SetChanADC	Chọn kênh đầu vào cho bộ A/D.

• Hàm : **BusyADC**

Chức năng : Hàm báo bận của bộ chuyển đổi A/D.

Nguyên mẫu : char BusyADC(void);

Chú thích : Hàm này được sử dụng để báo bộ chuyển đổi A/D đang trong quá trình chuyển đổi hay đã chuyển đổi xong.

Giá trị trả về : Bằng 1 nếu bộ ADC đang thực hiện chuyển đổi.
Bằng 0 nếu bộ ADC không thực hiện chuyển đổi.

• Hàm : **CloseADC**

Chức năng : Cấm hoạt động chuyển đổi A/D.

Nguyên mẫu : void CloseADC(void);

Chú thích : Hàm này được sử dụng để cấm hoạt động chuyển đổi A/D và cấm ngắt A/D.

• Hàm : ConvertADC

Chức năng : Bắt đầu quá trình chuyển đổi A/D

Nguyên mẫu : void ConvertADC(void);

Chú thích : Hàm này có chức năng ra lệnh bắt đầu quá trình chuyển đổi A/D. Sau khi bắt đầu chuyển đổi thì có thể sử dụng hàm BusyADC() để phát hiện hoàn thành quá trình chuyển đổi.

• Hàm : **OpenADC**

Chức năng : Cấu hình cho bộ chuyển đổi A/D.

Nguyên mẫu : void OpenADC(unsigned char **config**,
unsigned char **config2** ,
unsigned char **portconfig**);

Đối số: ***config***

Cấu hình cho **config** được thực hiện bằng phép toán AND ('&') với mỗi giá trị của mỗi loại được liệt kê bên dưới. Các giá trị này được định nghĩa trong file adc.h.

Chọn nguồn xung cho bộ A/D (A/D clock source):

ADC_FOSC_2	FOSC / 2
ADC_FOSC_4	FOSC / 4
ADC_FOSC_8	FOSC / 8
ADC_FOSC_16	FOSC / 16
ADC_FOSC_32	FOSC / 32
ADC_FOSC_64	FOSC / 64
ADC_FOSC_RC	Bộ dao động RC nội

Cách ghi kết quả:

ADC_RIGHT_JUST	10 bit thấp
ADC_LEFT_JUST	10 bit cao

Lựa chọn T_{ACQ}:

ADC_0_TAD	0 Tad
ADC_2_TAD	2 Tad
ADC_4_TAD	4 Tad
ADC_6_TAD	6 Tad
ADC_8_TAD	8 Tad
ADC_12_TAD	12 Tad
ADC_16_TAD	16 Tad
ADC_20_TAD	20 Tad

config2

Cấu hình cho **config2** được thực hiện bằng phép toán AND ('&') với mỗi giá trị của mỗi loại được liệt kê bên dưới. Các giá trị này được định nghĩa trong file adc.h.

Chọn kênh đầu vào tương tự:

ADC_CH0 Kênh 0
ADC_CH1 Kênh 1
ADC_CH2 Kênh 2
ADC_CH3 Kênh 3
ADC_CH4 Kênh 4
ADC_CH5 Kênh 5
ADC_CH6 Kênh 6
ADC_CH7 Kênh 7
ADC_CH8 Kênh 8
ADC_CH9 Kênh 9
ADC_CH10 Kênh 10
ADC_CH11 Kênh 11
ADC_CH12 Kênh 12

Ngắt A/D:

ADC_INT_ON Cho phép ngắt
ADC_INT_OFF Cấm ngắt

Cấu hình điện áp tham chiếu V_{REF+}:

ADC_VREFPLUS_VDD $V_{REF+} = V_{DD}$ (5V)
ADC_VREFPLUS_EXT V_{REF+} = Điện áp tạo từ mạch ngoài

Cấu hình điện áp tham chiếu V_{REF-}:

ADC_VREFMINUS_VSS $V_{REF-} = V_{SS}$ (0V)
ADC_VREFMINUS_EXT V_{REF-} = Điện áp tạo từ mạch ngoài

portconfig

portconfig bao gồm các giá trị từ 0 đến 15, đây cũng chính là giá trị của các bit từ 0 đến 3 của thanh ghi ADCON1, nó chính là các bit cấu hình đầu vào/ra số hay vào tương tự của các PORT.

Ví dụ:

```
OpenADC( ADC_FOSC_32    &
         ADC_RIGHT_JUST &
         ADC_12_TAD,
         ADC_CH0        &
         ADC_INT_OFF, 15 );
```

• Hàm : **ReadADC**

Chức năng : Đọc giá trị trả về của chuyển đổi A/D.

Nguyên mẫu : int ReadADC(void);

Chú thích : Hàm này đọc 16 bit kết quả của hoạt động chuyển đổi A/D.

Giá trị trả về : Hàm này trả về 16 bit kết quả có dấu của hoạt động chuyển đổi A/D.

• Hàm : SetChanADC

Chức năng : Chọn kênh đầu vào cho bộ chuyển đổi A/D.

Nguyên mẫu : void SetChanADC(unsigned char **channel**);

Đối số : ***channel***

Là một trong các giá trị dưới:

ADC_CH0 Kênh 0
ADC_CH1 Kênh 1
ADC_CH2 Kênh 2
ADC_CH3 Kênh 3
ADC_CH4 Kênh 4
ADC_CH5 Kênh 5
ADC_CH6 Kênh 6
ADC_CH7 Kênh 7
ADC_CH8 Kênh 8
ADC_CH9 Kênh 9
ADC_CH10 Kênh 10
ADC_CH11 Kênh 11
ADC_CH12 Kênh 12

Chú thích : Lựa chọn kênh đầu vào tương tự cho bộ chuyển đổi A/D.

Ví dụ : SetChanADC(ADC_CH0);

7. TRUYỀN THÔNG NỐI TIẾP QUA USART

7.1. Các thanh ghi liên quan

- Thanh ghi trạng thái truyền và điều khiển (TXSTA)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN(1)	SYNC	SENDB	BRGH	TRMT	TX9D
bit 7							bit 0

bit 7 **CSRC**: bit lựa chọn nguồn xung

Chế độ không đồng bộ:

Không hỗ trợ.

Chế độ đồng bộ:

1 = Chế độ chủ (xung được tạo trên chip từ BRG)
0 = Chế độ tớ (nguồn xung từ bên ngoài)

bit 6 **TX9:** Bit cho phép truyền 9 bit

1 = Lựa chọn chế độ truyền 9 bit
0 = Lựa chọn chế độ truyền 8 bit

bit 5 **TXEN:** Bit cho phép truyền

1 = Cho phép truyền
0 = Không cho phép truyền

bit 4 **SYNC:** Bit lựa chọn chế độ EUSART

1 = Chế độ đồng bộ
0 = Chế độ không đồng bộ

bit 3 **SENDB:** Bit gửi ký tự kết thúc (Break Character)

Chế độ không đồng bộ:

1 = Gửi đồng bộ kết thúc (được xóa bằng phần cứng lúc hoàn thành)
0 = Hoàn thành truyền đồng bộ kết thúc

Chế độ đồng bộ:

Không hỗ trợ.

bit 2 **BRGH:** Bit lựa chọn baud tốc độ cao

Chế độ không đồng bộ:

1 = Tốc độ cao
0 = Tốc độ thấp

Chế độ đồng bộ:

Không được sử dụng.

bit 1 **TRMT:** Bit báo trạng thái thanh ghi dịch truyền dữ liệu

1 = TSR rỗng
0 = TSR đầy

bit 0 **TX9D:** Bit truyền dữ liệu thứ 9

Có thể sử dụng chứa địa chỉ/dữ liệu hoặc bit chẵn lẻ.

- Thanh ghi điều khiển và trạng thái nhận (RCSTA)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
bit 7							bit 0

bit 7 **SPEN**: Bit cho phép USART

1 = Cho phép USART (các chân RX/DT và TX/CK sẽ được cấu hình là các chân của USART)

0 = Không cho phép

bit 6 **RX9:** Bit cho phép USART nhận 9 bit

1 = Lựa chọn nhận 9 bit

0 = Lựa chọn nhận 8 bit

bit 5 **SREN:** Bit cho phép nhận đơn

Chế độ không đồng bộ:

Không sử dụng.

Chế độ đồng bộ – Chủ:

1 = Cho phép nhận đơn

0 = Không cho phép nhận

Bit này sẽ được xóa sau khi quá trình nhận hoàn thành.

Chế độ đồng bộ – Tớ:

Không sử dụng.

bit 4 **CREN:** Bit cho phép nhận liên lục

Chế độ không đồng bộ:

1 = Cho phép nhận liên tục

0 = Không cho phép nhận liên tục

Chế độ đồng bộ:

1 =Cho phép nhân liên tục, CREN sẽ bị xóa khi SREN(bit cho phép nhận đơn) được thiết lập

0 = Không cho phép nhận liên tục

bit 3 **ADDEN:** Bit cho phép phát hiện địa chỉ

Chế độ đồng bô 9-Bit (RX9 = 1):

1 = Cho phép phát hiện địa chỉ, cho phép ngắt và tải dữ liệu từ bộ đệm nhận khi RSR<8> được thiết lập (=1).

0 = Không cho phép phát hiện địa chỉ, tất cả các byte được nhận và bit thứ 9 có thể được sử dụng như là bit kiểm tra chẵn lẻ.

Chế độ không đồng bộ 8-Bit (RX9 = 0):

Không được sử dụng.

bit 2 **FERR**: Bit báo lỗi khung truyền/nhận

1 = Khung bị lỗi (có thể được xóa khi đọc thanh ghi RCREG và nhận byte hợp lệ kế tiếp).

0 = Không xảy ra lỗi khung

bit 1 **OERR:** Bit lỗi do tràn

1 = Lỗi tràn (có thể được xóa bằng khi xóa bit CREN)

0 = Không xảy ra lỗi tràn

bit 0 **RX9D:** Bit nhận dữ liệu thứ 9

Có thể chứa bit địa chỉ/dữ liệu hoặc bit chẵn lẻ và được tính toán và xử lý theo chương trình của người sử dụng.

- Thanh ghi điều khiển tốc độ baud (BAUDCON)

R/W-0	R-1	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE	ABDEN
bit 7							bit 0

bit 7 **ABDOVF:** Bit trạng thái tự động điều chỉnh tốc độ baud

1 = Tốc độ baud từ BRG đã được điều chỉnh ở chế độ phát hiện tốc độ baud tự động (bit này phải được xóa bằng phần mềm).

0 = Không phát hiện điều chỉnh tốc độ ở BRG

bit 6 **RCIDL:** Bit trạng thái nghỉ (Idle) của hoạt động nhận

1 = Hoạt động nhận ở trạng thái nghỉ (Idle)

0 = Hoạt động nhận ở trạnh thái hoạt động (Active)

bit 5 **RXDTP:** Bit lưa chọn phân cực Dữ liệu/Nhận

Chế độ không đồng bộ:

1 = Nhận dữ liệu (RX) được đảo ngược (tích cực thấp)

0 = Nhận dữ liệu (RX) không được đảo ngược (tích cực cao)

Chế độ đồng bộ:

1 = Dữ liệu (DT) được đảo ngược (tích cực thấp)

0 = Dữ liệu (DT) không được đảo ngược (tích cực cao)

bit 4 **TXCKP:** Bit lựa chọn phân cực xung clock và dữ liệu

Chế độ không đồng bộ:

1 = Trạng thái nghỉ (Idle) của hoạt động truyền (TX) được thiết lập ở mức thấp

0 = Trạng thái nghỉ (Idle) của hoạt động truyền (TX) được thiết lập ở mức cao

Chế độ đồng bộ:

1 = Trạng thái nghỉ (Idle) của hoạt động phát xung clock (CK) được thiết lập ở mức cao.

0 = Trạng thái nghỉ (Idle) của hoạt động phát xung clock (CK) được thiết lập ở mức thấp.

bit 3 **BRG16:** Bit cho phép thanh ghi tốc độ baud 16-Bit

1 = Bộ phát tốc độ baud 16-bit, gồm hai thanh ghi SPBRGH và SPBRG

0 = Bộ phát tốc độ baud 8-bit, chỉ sử dụng thanh ghi SPBRG, bỏ qua thanh ghi SPBRGH.

bit 2 **Không được sử dụng:** Đọc sẽ được '0'

bit 1 **WUE:** Bit cho phep đánh thức (Wake-up)

Chế độ không đồng bộ:

1 = EUSART tiếp tục lấy mẫu trên chân RX – ngắt sẽ phát sinh ở sườn âm; bit này sẽ được xóa bằng phần cứng sau khi có sườn dương.

0 = Chân RX không được giám sát hoặc phát hiện sườn

Chế độ đồng bộ:

Không sử dụng ở chế độ này.

bit 0 **ABDEN:** Bit cho phép phát hiện tốc độ baud tự động

Chế độ không đồng bộ:

1 = Cho phép đo tốc độ baud ở ký tự tiếp theo. Bit này được xóa bằng phần cứng lúc hoàn thành.

0 = Không cho phép hoạt động đo tốc độ baud hoặc hoàn thanh.

Chế độ đồng bộ:

Không sử dụng ở chế độ này.

7.2. Tốc độ baud

Bộ tạo tốc độ baud BRG (Baud Rate Generator) có thể hoạt động ở chế độ 8 bit hoặc 16 bit, hỗ trợ cả chế độ đồng bộ và không đồng bộ của EUSART. Ở chế độ mặc định, BGR hoạt động ở chế độ 8 bit. Chế độ BGR 16 bit được lựa chọn khi bit BAUDCON<3> được thiết lập.

Hai thanh ghi SPBRGH:SPBRG được sử dụng để điều khiển chu kỳ xung tốc độ baud. Trong chế độ không đồng bộ, cả hai bit BRGH(TXSTA<2>) và BRG16(BAUDCON<3>) đều được sử dụng để điều khiển tốc độ baud. Trong chế độ đồng bộ, bit BRGH không được sử dụng.

- Lựa chọn chế độ và công công thức tính tốc độ baud:

Cấu hình các bit			**Chế độ BRG/EUSART**	**Công thức tốc độ baud**
SYNC	**BRG16**	**BRGH**		
0	0	0	8-Bit/Không đồng bộ	FOSC/[64 (n + 1)]
0	0	1	8-Bit/Không đồng bộ	FOSC/[16 (n + 1)]
0	1	0	16-Bit/Không đồng bộ	
0	1	1	16-Bit/Không đồng bộ	FOSC/[4 (n + 1)]
1	0	x	8-Bit/Đồng bộ	
1	1	x	16-Bit/Đồng bộ	

Ghi chú:

x = Giá trị bất kỳ n = Giá trị của cặp thanh ghi SPBRGH:SPBRG

Các thanh ghi liên quan đến bộ điều chỉnh tốc độ baud (BRG):

Tên	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
BAUDCON	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE	ABDEN
SPBRGH	EUSART Thanh ghi tạo tốc độ baud byte cao							
SPBRG	EUSART Thanh ghi tạo tốc độ baud byte cao							

Ghi chú: — Không được sử dụng, đọc sẽ được '0'. Các ô tô màu không được sử dụng ở BRG.

7.3. Chế độ không đồng bộ

Truyền của EUSART ở chế độ không đồng bộ

Thanh ghi TSR (Transmit (Serial) Shift Register) được sử dụng để dịch lần lượt các bit dữ liệu nối tiếp từ bit trọng số thấp nhấp LSb đến bit có trọng số cao MSb ra chân TX. Thanh ghi TSR không cho phép đọc/ghi bằng phần mềm. Thanh ghi TXREG được sử dụng để đệm dữ liệu cho thanh ghi TSR. Dữ liệu cần truyền được nạp vào thanh ghi TXREG, sau đó dữ liệu sẽ được nạp tự động từ TXREG sang TSR. Thanh ghi TSR chưa được nạp dữ liệu khi bit Dừng (Stop) trước đó chưa được truyền đi. Ngay sau khi bit Dừng được truyền đi thì dữ liệu sẽ được nạp vào TRS (nếu có dữ liệu trong TXREG).

Ngay sau khi dữ liệu được nạp từ TXREG sang TSR (trong một chu kỳ máy), thanh ghi TXREG sẽ rỗng và cờ ngắt truyền TXIF (PIR1<4>) sẽ được thiết lập (=1).

Bit TXIF được sử dụng để biết trạng thái của thanh ghi TXREG, còn bit TRMT (TXSTA<1>) được sử dụng để biết trạng thái của thanh ghi TSR. Bit TRMT chỉ được phép đọc, nó thiết lập khi TSR rỗng. Hoạt động ngắt không được gắn liền với bit này, nó chỉ sử dụng để báo trạng thái rỗng của thanh ghi TSR.

Các bước để truyền dữ liệu ở chế độ không đồng bộ:

Bước 1. Khởi tạo giá trị cho cặp thanh ghi SPBRGH:SPBRG, thiết lập hoặc xóa bit BRGH và BRG16 để đạt được tốc độ truyền mong muốn (theo bảng chế độ và công thức tính tốc độ baud).

Bước 2. Xóa bit SYNC (TXSTA<4>) để cho phép chế độ không đồng bộ và thiết lập bit SPEN (RCSTA<7>) để cho phép USART.

Bước 3. Nếu muốn sử dụng ngắt thì cần phải thiết lập bit TXIE.

Bước 4. Để thiết lập khung truyền là 9-bit cần thiết lập bit TX9 (TXSTA<6>). Khi đó bit-9 sẽ có thể được sử dụng để chứa địa chỉ/dữ liệu hoặc bit kiểm tra chẵn lẻ.

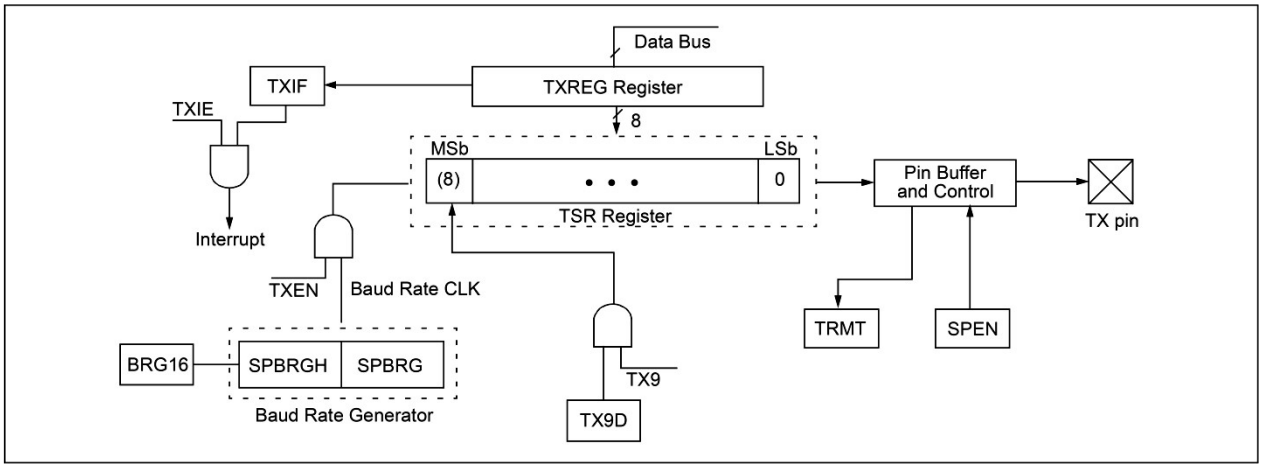
Bước 5. Cho phép truyền dữ liệu bằng bit TXEN.

Bước 6. Nếu khung truyền 9 bit được lựa chọn, bit thứ 9 cần được nạp vào TX9D.

Bước 7. Nạp dữ liệu cần truyền vào thanh ghi TXREG (quá trình truyền dữ liệu sẽ được bắt đầu).

Bước 8. Nếu sử dụng ngắt, cần chắc chắn rằng bit GIE và PEIE của thanh ghi INTCON (INTCON<7:6>) đã được thiết lập.

Sơ đồ khối hoạt động truyền của EUSART ở chế độ không đồng bộ:

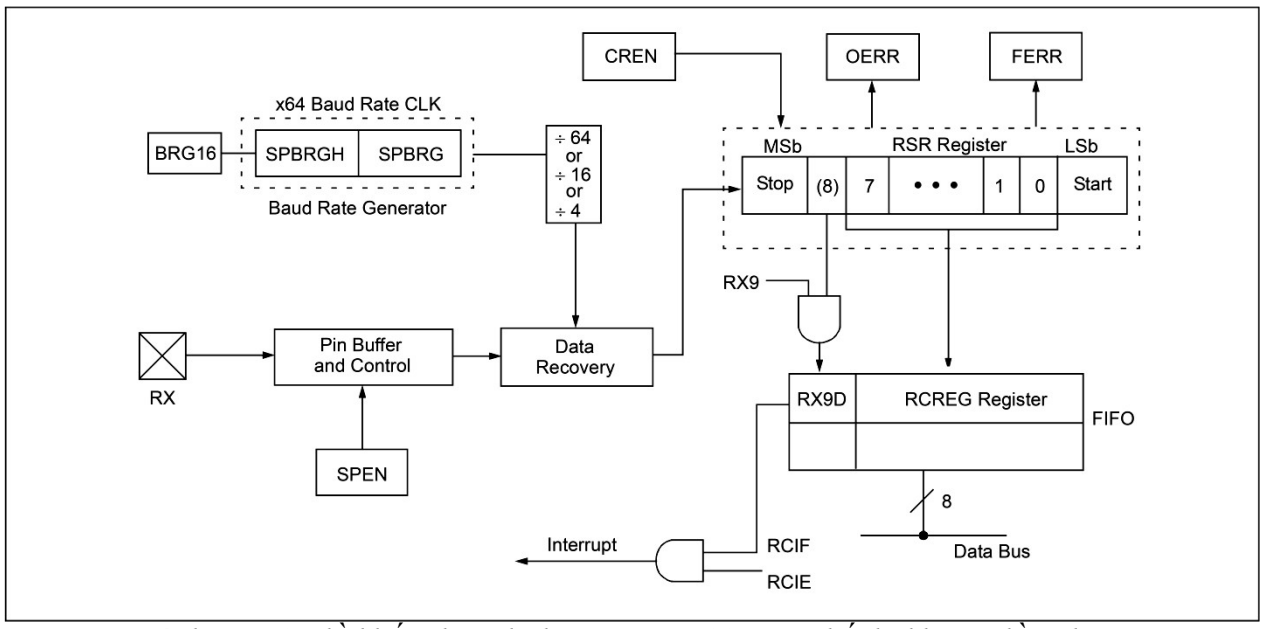


Các thanh ghi liên quan đến hoạt động truyền không đồng bộ:

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
PIR1	PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
PIE1	PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
IPR1	PSPIP(1)	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
TXREG	Thanh ghi truyền dữ liệu của EUSART							
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D
BAUDCON	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE	ABDEN
SPBRGH	Thanh ghi tạo tốc độ baud byte cao của EUSART							
SPBRG	Thanh ghi tạo tốc độ baud thấp cao của EUSART							

Nhận của EUSART ở chế độ không đồng bộ

Sơ đồ khối của hoạt động nhận ở chế độ bất động bộ được thể hiện ở hình 8.2 dưới. Dữ liệu được nhận về qua chân RX và khối phục hồi dữ liệu. Chế độ này thường được sử trong hệ thống truyền thông RS-232.



Hình 8.4. Sơ đồ khối nhận dữ liệu của EUSART ở chế độ không đồng bộ

Các bước để truyền dữ liệu ở chế độ không đồng bộ:

Bước 1. Khởi tạo giá trị cho cặp thanh ghi SPBRGH:SPBRG, thiết lập hoặc xóa bit BRGH và BRG16 để đạt được tốc độ truyền mong muốn (theo bảng chế độ và công thức tính tốc độ baud).

Bước 2. Xóa bit SYNC (TXSTA<4>) để cho phép chế độ không đồng bộ và thiết lập bit SPEN (RCSTA<7>) để cho phép USART.

Bước 3. Nếu sử dụng ngắt thì cần phải thiết lập bit RCIE.

Bước 4. Để cho phép nhận bit thứ 9 cần phải thiết lập bit RX9(RCSTA<6>).

Bước 5. Thiết lập bit CREN để cho phép hoạt động nhận.

Bước 6. Bit cờ ngắt RCIF sẽ được thiết lập khi hoạt động nhận hoàn thành, ngắt sẽ xảy ra khi bit cho phép ngắt RCIE đã được thiết lập trước đó.

Bước 7. Đọc bit RX9D(RCSTA<0>) để có được bit thứ 9 (nếu khung truyền 9 bit được cho phép), căn cứ vào bit thứ 9 để phát hiện lỗi khung truyền.

Bước 8. Đọc 8 bit dữ liệu nhận về trong thanh ghi RCREG.

Bước 9. Nếu phát hiện dữ liệu nhận bị lỗi, xóa lỗi bằng cách xóa bit cho phép nhận CREN.

Bước 10. Nếu sử dụng ngắt, cần chắc chắn rằng bit GIE và PEIE của thanh ghi INTCON (INTCON<7:6>) đã được thiết lập.

Các thanh ghi liên quan đến chế độ nhận không đồng bộ:

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
PIR1	PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
PIE1	PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE

IPR1	PSPIP(1)	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
RCREG	Thanh ghi nhận dữ liệu của EUSART.							
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D
BAUDCON	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE	ABDEN
SPBRGH	Thanh ghi tạo tốc độ baud byte cao của EUSART.							
SPBRG	Thanh ghi tạo tốc độ baud byte thấp của EUSART.							

7.5. Một số hàm thông dụng trong thư viện usart.h

Hàm	Mô tả
BusyUSART	Hàm báo bận của hoạt động truyền nối tiếp.
CloseUSART	Cấm USART.
OpenUSART	Cấu hình USART.
putsUSART	Truyền một mảng ra PORT USART.
ReadUSART	Đọc 1 byte từ PORT USART.
WriteUSART	Truyền 1 byte ra PORT USART.

• Hàm : **BusyUSART**

Chức năng : Hàm báo bận của hoạt động truyền nối tiếp
Nguyên mẫu : char BusyUSART(void);
Chú thích : Sử dụng để báo trạng thái truyền của USART
Giá trị trả về : Bằng 0 nếu USART không bận.
Bằng 1 nếu USART không bận (đang truyền).
Ví dụ: while (BusyUSART());

• Hàm : OpenUSART

Chức năng : Cấu hình cho USART
Chú thích : Sử dụng để đặt các thông số của USART.
Nguyên mẫu : void OpenUSART(unsigned char config,
unsigned int spbrg);

Các đối số:

"config": dùng thiết lập các thông số sau cho USART (các thông số có thể được thiết lập đồng thời bằng toán tử "&"):

Thông số 1:

USART_TX_INT_ON: cho phép ngắt truyền

USART_TX_INT_OFF: cấm ngắt truyền

Thông số 2:

USART_RX_INT_ON: cho phép ngắt nhận

USART_RX_INT_OFF: cấm ngắt nhận

Thông số 3:

USART_ASYNCH_MODE: Chọn chế độ cận đồng bộ (Asynchronous Mode)

USART_SYNCH_MODE: Chọn chế độ đồng bộ (Synchronous Mode)

Thông số 4:

USART_EIGHT_BIT: Chọn chế độ truyền/nhận 8 bit (8-bit transmit/receive)

USART_NINE_BIT: Chọn chế độ truyền/nhận 9 bit (9-bit transmit/receive)

Thông số 5:

USART_SYNC_SLAVE: Chọn chế độ tớ (Synchronous Slave mode)

USART_SYNC_MASTER: Chọn chế độ chủ (Synchronous Master mode)

Thông số 6:

USART_SINGLE_RX: Chọn chế độ nhận từng byte (Single reception)

USART_CONT_RX: Chọn chế độ nhận liên tục các byte (Continuous reception)

Thông số 7:

USART_BRGH_HIGH: Chọn tốc độ baud cao (High baud rate)

USART_BRGH_LOW: Chọn tốc độ baud thấp (Low baud rate)

"spbrg": dùng để đặt tốc độ baud, tốc độ baud được tính như sau:

- Với chế độ cận đồng bộ, tốc độ baud cao (Asynchronous mode, high speed):

Tốc độ baud=Fosc/ (16 * (spbrg+ 1))

- Với chế độ cận đồng bộ, tốc độ baud thấp (Asynchronous mode, low speed):

Tốc độ baud=Fosc/ (64 * (spbrg+ 1))

Với chế độ đồng bộ (Synchronous mode):

Tốc độ baud=Fosc/ (4 * (spbrg+ 1))

• Hàm : CloseUSART

Chức năng : Đóng (cấm) USART
Nguyên mẫu : void CloseUSART(void);
Chú thích : Hàm được gọi khi không sử dụng USART
Ví dụ: CloseUSART();

• Hàm : WriteUSART

Chức năng : Ghi một byte vào bộ đệm truyền của USART
Nguyên mẫu : void WriteUSART(char data);
Chú thích : Hàm được sử dụng để truyền đi một byte
Ví dụ: WriteUSART(0x41); //truyền ký tự A

• Hàm : **ReadUSART**

Chức năng : Nhận một byte từ bộ đệm nhận của USART
Nguyên mẫu : char ReadUSART(void);
Chú thích : Hàm được sử dụng để nhận một byte

Ví dụ: char x;

x= ReadUSART();

• Hàm : **putsUSART**

Chức năng : Nhận một byte từ bộ đệm nhận của USART
Nguyên mẫu : char ReadUSART(void);
Chú thích : Hàm được sử dụng để nhận một byte

Ví dụ: char x;

x= ReadUSART();

Phụ lục 1. Màn hình tinh thể lỏng –LCD

Các mô-đun LCD hiển thị ký tự được thiết kế dựa trên bộ điều khiển HD44780 của Hitachi, loại LCD 16x2 ký tự gồm 14 chân, ý nghĩa các chân như sau:

Chân số	Ký hiệu	Mức logic	I/O	Chức năng
1	Vss	-	-	Nguồn cung cấp (GND)
2	Vcc	-	-	Nguồn cung cấp (+5V)
3	Vee	-	I	Điện áp vào để điều chỉnh độ tương phản
4	RS	0/1	I	Lựa chọn thanh ghi 0 = Thanh ghi lệnh 1 = Thanh ghi dữ liệu
5	R/W	0/1	I	0 = Ghi vào LCD module 1 = Đọc từ LCD module
6	E	1, 1->0	I	Tín hiệu cho phép
7	DB0	0/1	I/O	Bus dữ liệu, bit 0 (LSB)
8	DB1	0/1	I/O	Bus dữ liệu, bit 1
9	DB2	0/1	I/O	Bus dữ liệu, bit 2
10	DB3	0/1	I/O	Bus dữ liệu, bit 3
11	DB4	0/1	I/O	Bus dữ liệu, bit 4
12	DB5	0/1	I/O	Bus dữ liệu, bit 5
13	DB6	0/1	I/O	Bus dữ liệu, bit 6
14	DB7	0/1	I/O	Bus dữ liệu, bit 7 (MSB)

Điều khiển hiển thị các ký tự trên LCD thực chất là gửi các mã lệnh tới bộ điều khiển HD44780, các mã lệnh cơ bản được liệt kê như sau:

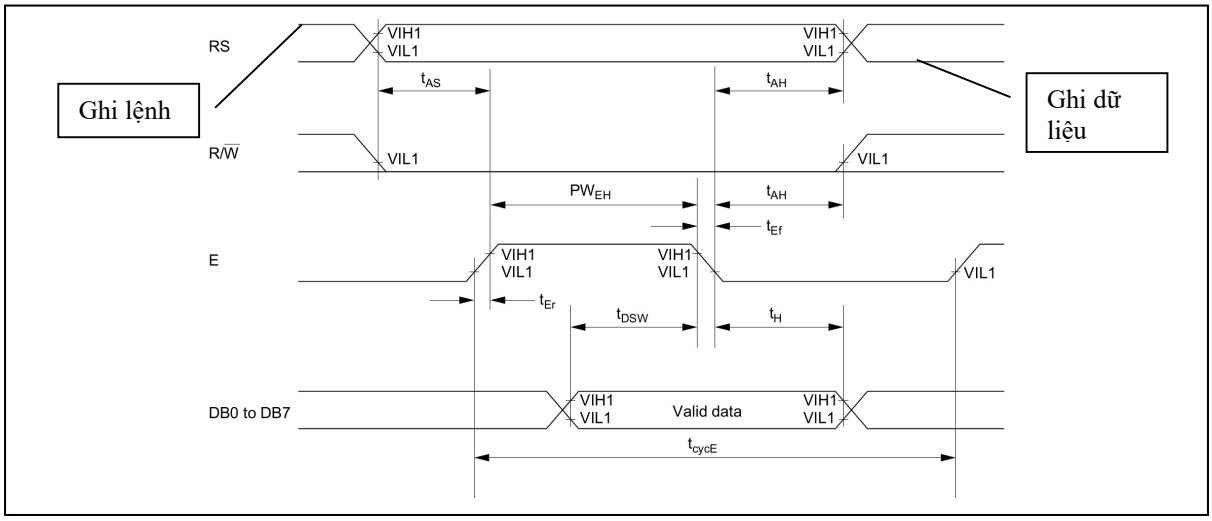
TT	Lệnh	Mã lệnh										Mô tả	Thời gian thi hành
		RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
1	Start	0	0	0	0	0	0	0	0	1	1	Đưa LCD về chế độ "bắt đầu"	1.64 mS
2	Cấu hình LCD	0	0	0	0	1	DL	N	F	0	0	Thiết lập độ dài của bus dữ liệu, số dòng và kích thước font chữ.	40µS
3	Thiết lập chế độ hiển thị	0	0	0	0	0	0	0	1	I/D	S	Thiết lập hướng dịch con trỏ(I/D), hướng hiển thị (S).	40µS
4	Bật/Tắt hiển thị	0	0	0	0	0	0	1	D	C	B	-Bật/Tắt các ký tự đang hiển thị (D); -Bật/Tắt con trỏ (C); -Bật/Tắt chế độ nhấp nháy của con trỏ (B).	40µS
5	Dịch con trỏ/hiển thị	0	0	0	0	0	1	S/C	R/L	*	*	Thiết lập chiều dịch chuyển của con trỏ và hiển thị.	40µS
6	Xóa màn hình	0	0	0	0	0	0	0	0	0	1	Xóa màn hình, đưa con trỏ về vị trí đầu (address 0).	1.64 mS
7	Thiết lập vị trí con trỏ	0	0	address									40µS
8	Hiển thị ký tự	1	0	Mã của ký tự cần hiển thị								Ghi dữ liệu vào DDRAM.	40µS

Ghi chú:

- DDRAM (Display Data RAM) : Chứa dữ liệu cần hiển thị (mã ASCII của các ký tự); địa chỉ của DDRAM tương ứng với vị trí con trỏ trên màn hình.
- Các bit/byte viết tắt trong mã lệnh:

TT	Tên	Mô tả	
1	I/D	0: Giảm vị trí con trỏ (lùi)	1: Tăng vị trí con trỏ (tiến)
2	S	0: Không dịch chuyển hiển thị	1: Dịch chuyển hiển thị
3	D	0: Tắt hiển thị	1: Bật hiển thị
4	C	0: Tắt con trỏ	1: Bật con trỏ
5	B	0: Con trỏ không nhấp nháy	1: Con trỏ nhấp nháy
6	S/C	0: Di chuyển con trỏ	1: Dịch chuyển hiển thị
7	R/L	0: Dịch trái	1: Dịch phải
8	DL	0: Chế độ 4-bit dữ liệu	1: Chế độ 8-bit dữ liệu
9	N	0= 1 dòng	1= 2 dòng
10	F	0: Font chữ 5x7	1: Font chữ 5x10
11	address	80(H)÷8F(H): đầu dòng thứ nhất đến cuối dòng thứ nhất	C0(H)÷CF(H): đầu dòng thứ hai đến cuối dòng thứ hai

Giản đồ thời gian mô tả quá trình ghi lệnh và ghi dữ liệu vào DDRAM (hiển thị ký tự) như sau:



Căn cứu giản đồ thời gian mô tả quá trình ghi lệnh và ghi dữ liệu, quá trình ghi một mã lệnh bao gồm các bước sau:

1. Đặt các chân RS, R/W ở mức thấp; chân E ở mức cao

2. Xuất mã lệnh cần ghi lên bus DB0÷DB7.

3. Đặt chân E ở mức thấp.

Quá trình ghi dữ liệu vào DDRAM (hiển thị ký tự) bao gồm các bước sau:

1. Đặt các chân R/W ở mức thấp; chân RS, chân E ở mức cao

2. Xuất mã ký tự cần hiển thị lên bus DB0÷DB7.

3. Đặt chân E ở mức thấp.

Nguyên tắc hiển thị ký tự trên LCD:

Một chương trình hiển thị ký tự trên LCD sẽ đi theo bốn bước sau:

1. Xoá toàn bộ màn hình.

2. Đặt chế độ hiển thị.

3. Đặt vị trí con trỏ (Nơi bắt đầu của ký tự hiển thị).

4. Hiển thị ký tự.

Ví dụ:

```
// Buoc 1: xoa man hinh
        Lcd_Write_Command(0x01);       // Lcd_Write_Command:Hàm
                                              //ghi lệnh cho LCD
// Buoc 2: Dat che do hien thi
//(1)dua LCD ve che do "bat dau"
        Lcd_Write_Command(0x03);
// (2)Cau hinh LCD:
// - DL=1: 8-bit du lieu
// - N=1: LCD hien thi tren 2 dong
// - F=0: Font chữ gom 5x7 diem (dot)
        Lcd_Write_Command(0x38);
// (3)Thiet lap che do hien thi:
// - I/D=1: Vi tri con tro tang (hien thi tu trai qua phai)
// - S=0: Khong hien thi dich (cac ky tu khong dich chuyen)
        Lcd_Write_Command(0x06);
// (4)Bat/Tat hien thi:
// - D=1: Bat hien thi (hien thi lien tuc-khong nhap nhay)
// - B=0: Con tro khong nhap nhay
// - C=0: Tat con tro (khong hien thi con tro tren LCD)
        Lcd_Write_Command(0x0c);
//Buoc 3: Dat vi tri hien thi
        Lcd_Write_Command(0XC0); // dau dong thu 2
//Buoc 4: Hien thi ky tu
        Lcd_Write_Data(0x41);     // hien thi A
                                        // Lcd_Write_Data: Hàm ghi dữ // liệu vào
                                        DDRAM
```

Chú ý:

- Các bước 3, 4 có thể lặp lại nhiều lần nếu cần hiển thị nhiều ký tự.

- Khi thực hiện ghi lệnh hoặc ghi dữ liệu hiển thị lên LCD cần lưu ý đến thời gian thi hành lệnh của bộ điều khiển HD44780. Ví dụ sau khi xoá màn hình thì tối thiểu 1.64 mS sau mới ra lệnh khác.

- Chế độ hiển thị mặc định sẽ là hiển thị dịch, vị trí con trỏ mặc định sẽ là đầu dòng thứ nhất.