**Vi điều khiển PIC18F2550**

1. Cấu hình bộ dao động

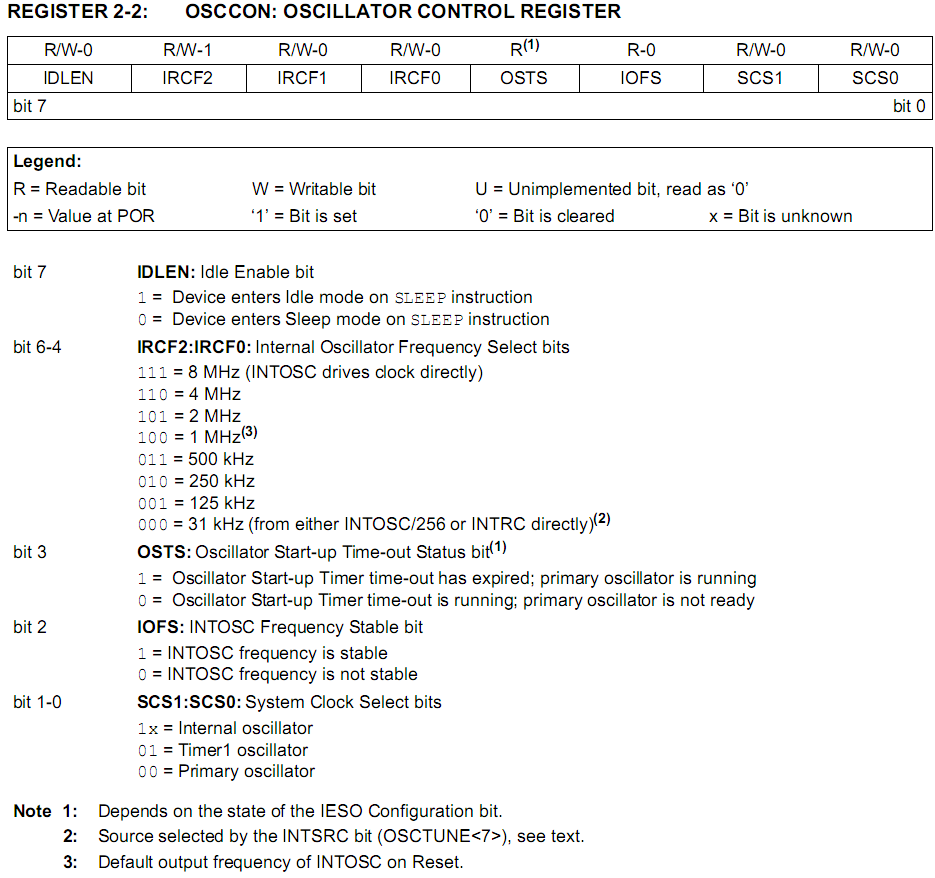
Bộ dao động nội gồm 2 nguồn dao động là 8 Mhz clock và INTRC. INTRC có tần số dao động xấp xỉ 31 kHz và nó ổn định khi nhiệt độ và VDD thay đổi.

Cấu hình bộ dao động là điều khiển 4 thanh ghi :

* 2 thanh ghi cấu hình : CONFIG1L và CONFIG1H
* 2 thanh ghi điều khiển : OSCCON và OSCTUNE

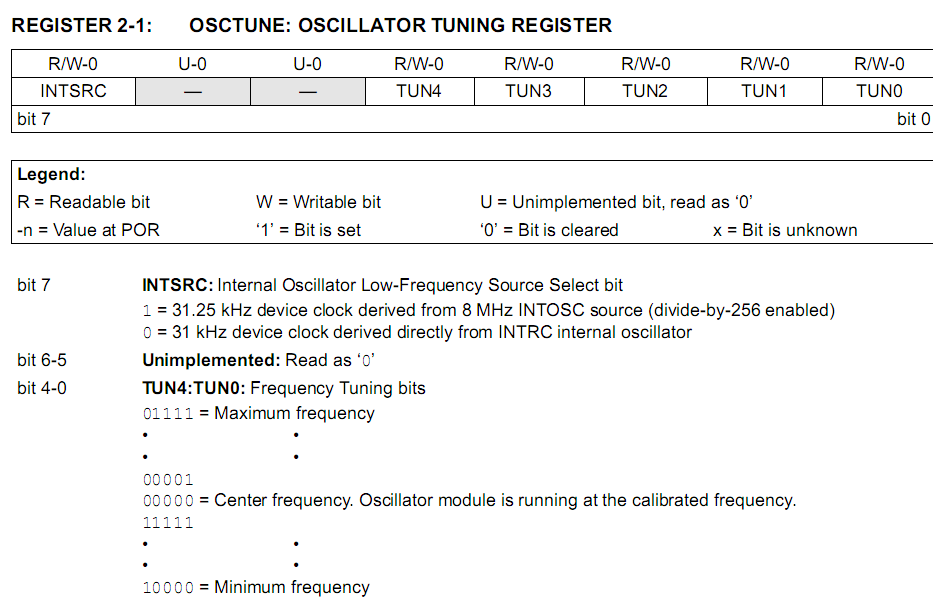
**Thanh ghi OSCCON** để chọn nguồn xung đồng hồ cho vđk hoạt động. Khi chuyển qua lại giữa các nguồn xung đồng hồ sẽ có khoảng tạm dùng dài 2 chu kỳ cũ + 3 hay 4 chu kỳ mới.

PIC18F2550 có bộ dao động Timer1 là bộ dao động phụ, nó hoạt động trong mọi chế độ quản lý năng lượng và thường là căn cứ về thời gian cho các chức năng như Real-time clock.

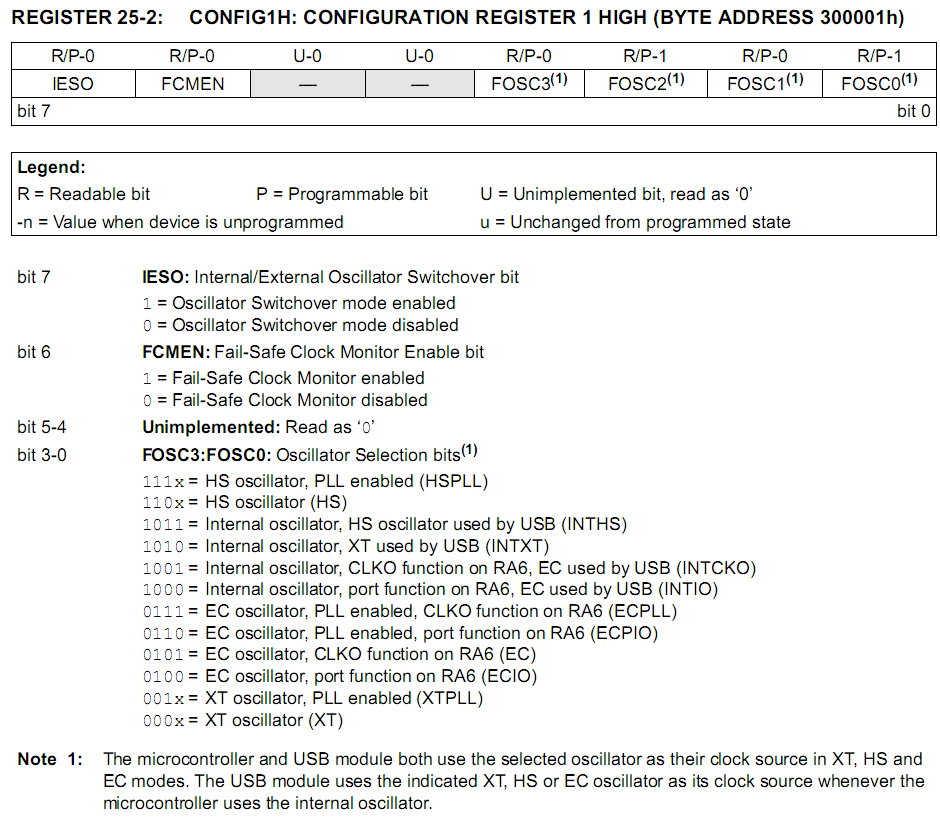


**Thanh ghi OSCTUNE** để cân chỉnh (đã được cân chỉnh ở nhà máy) và chọn tần số sử dùng từ bộ dao động trong (chọn thông qua bộ chia)

Tần số của bộ dao động trong là 8 Mhz. Bộ chia có 5 bit.



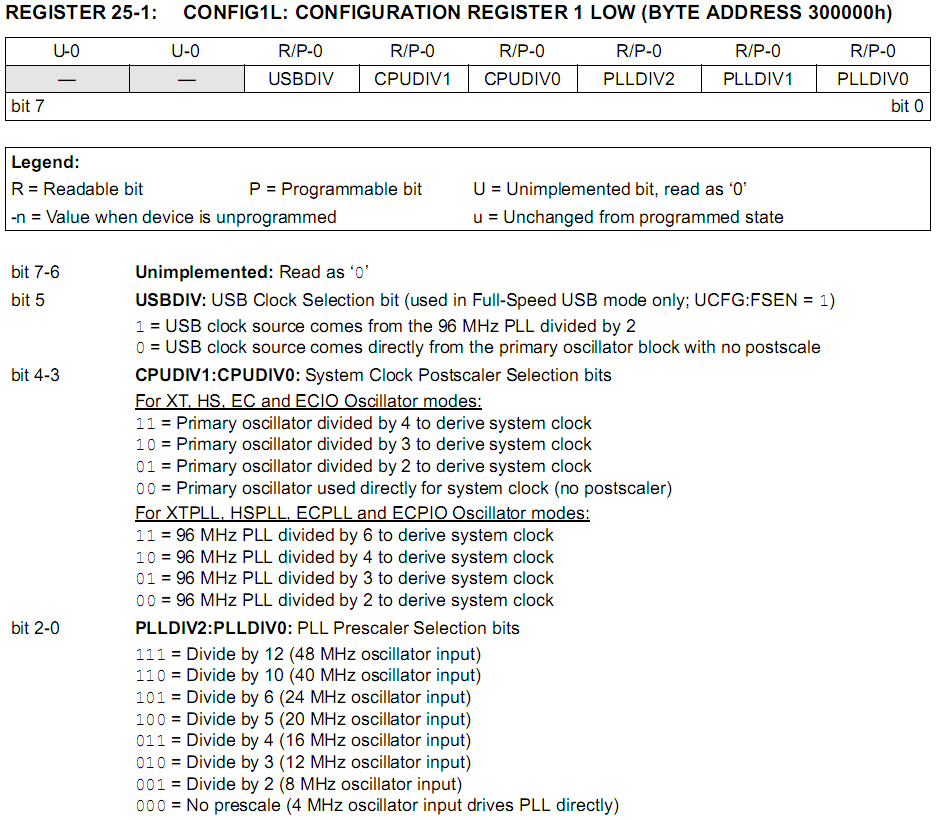
**Thanh ghi CONFIG1H**



**Thanh ghi CONFIG1L**

Dùng để chọn thiết lập xung clock cho CPU của 18F2550 và USB lấy từ nguồn xung chính (primary Oscillator).

Do ta dùng xung nội nên không quan tâm. Để mặc định.



* Vậy tóm tắt lại để thiết lập chế độ dao động nội ta làm như sau

CONFIG1H = 01--1001

CONFIG1L không set

OSCTUNE = 1--00000

OSCCON = 1100xx10

“x” là bit trạng thái không set được, chỉ đọc được

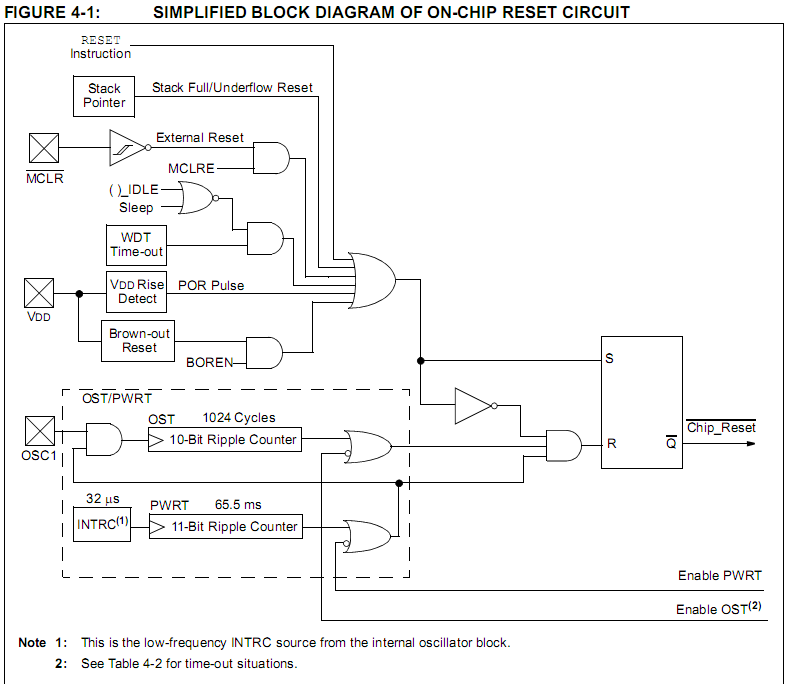
“-“ là bit không được dùng

**3.0 POWER-MANAGED MODES /37**

1. Vấn đề reset

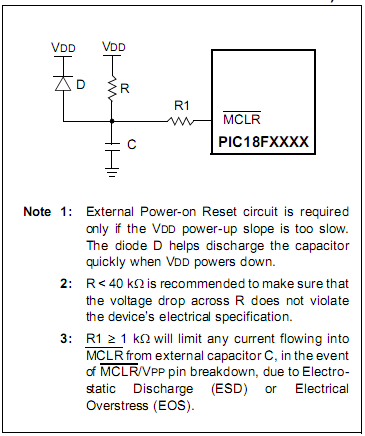
Các loại reset :

* Reset lúc bật nguồn (POR)
* MCLR
* Watchdog timer
* Reset Do yếu nguồn (BOR)
* Reset Do Stack
* Lệnh Reset



Nếu ta dùng điện áp thấp cho VDD  thì ta cần phải ráp mạch ngoài để tạo tín hiệu reset vào chân MCLR như hình

Nếu ta dùng áp nguồn đủ lớn thì không cần mạch reset ngoài như hình

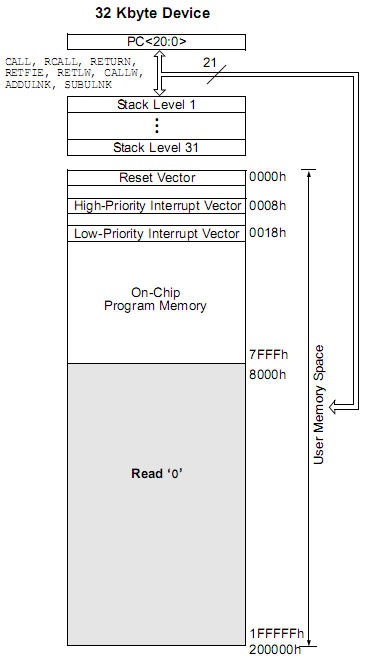


Reset do BOR không cần quan tâm

Để xung đồng hồ của thiết bị ổn định trước khi thực thi code, có 3 timer trì hoãn đảm nhận nhiệm vụ này

* Power-up Timer (PWRT) dùng nguồn xung từ INTRC để trì hoãn. Để bật tính năng trì hoãn cho bộ timer này hoạt động cần : xóa bit PWRTEN (thời gian chờ danh nghĩa là 66 ms).
* Oscillator Start-up Timer (OST) dùng khi dùng mạch dao động ngoài
* PLL Lock Time-out dùng khi dùng vòng khóa pha PLL

Vậy tóm lại dùng nguồn 5 V cho VDD thì không cần gắn thêm gì cả.

1. Tổ chức bộ nhớ

Có 3 loại bộ nhớ trong các thiết bị vi điều khiển cải tiến dòng PIC18F:

* Program Memory
* Data RAM
* Data EEPROM

Vì là kiến trúc Harvard nên bộ nhớ chương trình và dữ liệu dùng bus riêng,do đó có thể truy cập đồng thời.

A.Tổ chức bộ nhớ chương trình

1. Program counter (PC)

PC có 21 bit có thể định địa chỉ được 2 MB bộ nhớ. Tuy nhiên bộ nhớ thực tế ít hơn và là 32 KB (32768 ô, 1 byte/ô) nên nếu truy cập vùng nhớ ngoài vùng thực tế cho tới 2 MB sẽ trả về toàn giá trị 0. Tạo thành từ 3 thanh ghi rời :

|  |  |  |
| --- | --- | --- |
| PCU register (upper byte) | PCD register (high byte) | PCL register (low byte) |
| Đọc ghi thông qua thanh ghi PCLATU | Đọc ghi thông qua thanh ghi PCLATH | Đọc ghi được trực tiếp |
| 20 16 | 15 8 | 7 0 |
| **Program Counter** | | |

Địa chỉ vector reset là 0000h, vector ngắt là 0008h và 0018h.

18F2550 có 32KB bộ nhớ flash

1. Ngăn xếp chứa địa chỉ trả về (Return Address Stack)

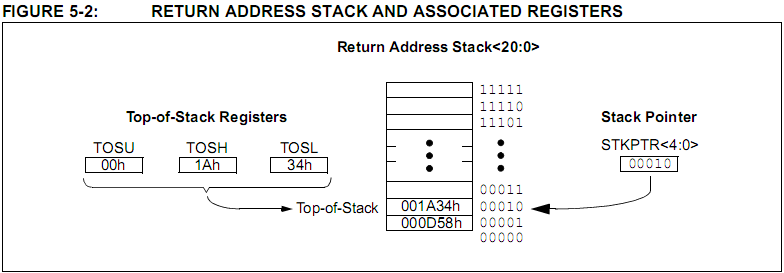
Trong PIC có ngăn xếp chứa địa chỉ trả về cho phép kết hợp tới 31 lời gọi chương trình và ngắt diễn ra. Đưa vào hay lấy ra bởi các lệnh CALL or RCALL, RETURN, RETLW or RETFIE.

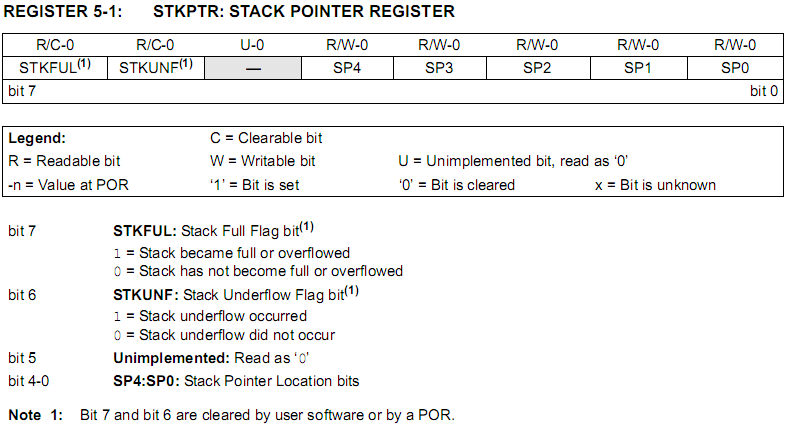
Ngăn xếp này có 31 ô, 21 bit/ô, và 1 con trỏ ngăn xếp STKPTR

5 bit trỏ tới ô trên cùng ngăn xếp (là cái mới được đưa vào lần gần nhất). STKPTR<4:0> và ô trên cùng ngăn xếp có thể đọc và ghi được. Khi truy xuất thay đổi phải tắt tất cả chế độ ngắt (dùng bit ngắt toàn cục) để tránh lỗi stack do sơ xuất.

Nội dung ô trên cùng chứa trong 3 thanh ghi TOSU:TOSH:TOSL.

Sau khi bị reset STKPTR sẽ trỏ tới địa chỉ 00000.





Bit STVREN (Stack Overflow Reset Enable) ảnh hưởng tới hoạt động của ngăn xếp

|  |  |  |
| --- | --- | --- |
|  | Stack tràn (push lần 31) | Stack rỗng (pop đủ số lần) |
| STVREN = 1 | -Lưu địa chỉ lệnh kế (PC + 2) vào ô 31  -bit STKFUL = 1  -Reset thiết bị  ->STKPTR = 00000  ->STKFUL = 1  Push nữa tiếp tục như bình thường | -PC = 0 (khác reset thiết bị là cac thanh ghi SFR không đổi)  -bit STKUNF = 1  -STKPTR = 0  Pop tiếp mọi thứ như cũ |
| STVREN = 0 | -bit STKFUL = 1  -STKPTR = 11111  Push tiếp thì STKPTR không đổi | -bit STKUNF = 1  -STKPTR = 0  Pop tiếp mọi thứ như cũ |

Vậy tóm lại không call hay return (push hay pop) liên tục nhiều vì gây reset hay lỗi.

1. Ngăn xếp lưu nhanh thanh ghi (fast register stack)

Dùng để lưu tạm 3 thanh ghi STATUS, WREG and BSR. Ngăn xếp này chỉ sâu 1 cấp và không thể đọc hay ghi được.

Khi có dùng ngắt ưu tiên thì ngăn xếp ghi tạm có thể bị ghi đè gây lỗi nên cần lưu lại các thanh ghi then chốt.

Nếu không dùng ngắt ưu tiên thì dùng bình thường. Nếu không dùng ngắt có thể dùng nó để lưu giá trị cho các thanh ghi này khi gọi chương trình con.

Để dùng ngăn xếp này dùng lệnh sau :

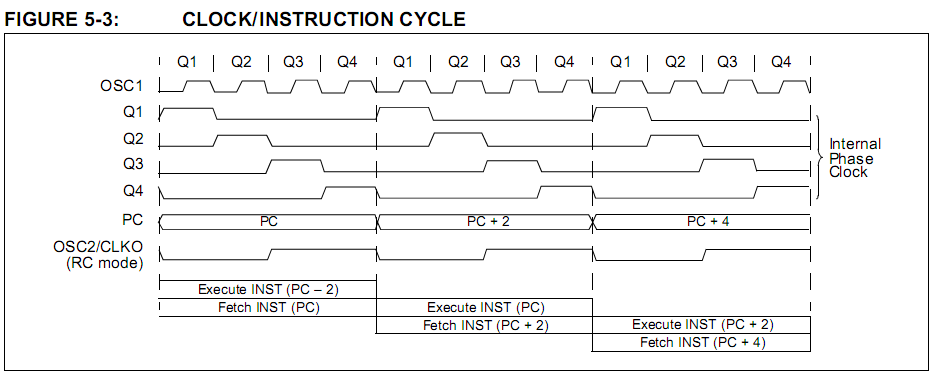
CALL label, FAST

RETURN, FAST

B.Chu kỳ lệnh của PIC18F

Chu kỳ lệnh chia làm 4 giai đoạn không trùng lắp nhau.

Một lệnh có giai đoạn Fetch ở chu kỳ n thì giai đoạn Decode và Excute trong chu kỳ n + 1. Nhưng do trong 1 chu kỳ, vxl fetch lệnh kế và thực hiện lệnh trước đó đã fetch nên hiệu quả thực sư xem như 1 lệnh chỉ thực thi trong 1 chu kỳ.

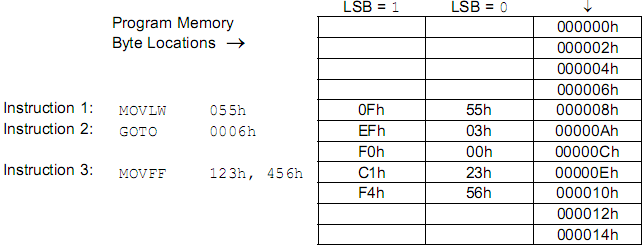


Tổ chức lệnh trong bộ nhớ chương trình

Lệnh được định địa chỉ theo byte, được lưu trữ thành 2 byte hay 4 byte trong bộ nhớ chương trình và byte thấp nhất luôn lưu trong bộ nhớ chương trình tại địa chỉ chẵn (LSB = 0) bằng cách PC tăng mỗi lần là 2 byte.

Với PIC18F các lệnh 2 word (4 byte) là CALL, MOVFF, GOTO và LSFR : trong mọi trường hợp, word thứ 2 luôn bắt đầu với 4 bit lớn nhất là 1111, 12 bit còn lại là dữ liệu 12 số, thường là địa chỉ bộ nhớ dữ liệu.

4 bit 1111 ở trên tạo thành lệnh NOP đặc biệt, nếu như word đầu tiên của lệnh 2 word bị bỏ qua do lệnh làm thay đổi PC thì khi chỉ thực thi word thứ 2 sẽ là lệnh NOP.



C.Tổ chức bộ nhớ dữ liệu

* Bộ nhớ dữ liệu trong thiết bị PIC18F là RAM tĩnh (static RAM). Mỗi thanh ghi trong bộ nhớ dữ liệu có địa chỉ 12 bit
* Vậy tức là có thế có tối đa 4096 byte bộ nhớ dữ liệu. Không gian bộ nhớ được chia thành 16 băng, 256 byte/băng.
* PIC18F2555 được chế tạo 8 băng hoàn chỉnh nên có tổng cộng bộ nhớ dữ liệu là 2048 bytes, truy cập ngoài vùng này sẽ chỉ đọc ra ‘0’
* Vùng bộ nhớ dữ liệu chứa các thanh ghi chức năng đặc biệt SFR và thanh ghi đa năng GPR. SFR dùng cho điều khiển và trạng thái của bộ điều khiển và chức năng ngoại vi, GPR dùng cho lưu trữ dữ liệu và vận hành có dùng vùng nhớ tạm (scratchpad operation) trong ứng dụng người dùng.
* Toàn bộ bộ nhớ dữ liệu có thể truy cập trực tiếp, gián tiếp hay chế độ định địa chỉ bằng chỉ số.
* Để bảo đảm truy cập thanh ghi hay dùng (gồm SFR và phần thấp của GPR bank 0) chỉ trong 1 chu kỳ, PIC18 có Access Bank – là một vùng nhớ 256 byte. (thông thường truy cập thì phải dùng BSR định địa chỉ).

1. RAM của USB

Băng 4 tới 7 của bộ nhớ dữ liệu dùng cho bộ giao tiếp nối tiếp Usb (USB serial interface engine - SIE) trao đổi dữ liệu trực tiếp với nhân bộ vi điều khiển. Khi tắt chức năng USB thì các băng này dùng bình thường

1. Thanh ghi lựa chọn băng (Bank select register – BSR)

Để truy cập nhanh chóng tới địa chỉ, PIC18 dùng các lệnh mang

* Điạ chỉ trực tiếp 12 bit của vùng cần truy cập (chỉ có lệnh MOVFF là theo cách này)
* Hay chỉ ra 8 bit thấp của địa chỉ và dùng thanh ghi chọn băng BSR.

BSR chứa 4 bit (BSR3:BSR0). 4 bit cao của BSR không dùng. BSR nạp được trực tiếp bằng lệnh MOVLB

1. Access bank

Access bank dùng để truy cập nhanh vùng bộ nhớ dữ liệu thường dùng nhiều nhất

Access bank có 96 byte (00h-5Fh) và 160 byte (60h-FFh). Hai vùng này trong Access bank có thể truy cập bằng địa chỉ 8 bit. Vùng Access Ram low là nơi rất tốt cho giá trị mà người dùng cần truy cập nhanh (kết quả hay các biến).

Vùng Access Ram high có thể giúp người dùng truy cập SFR hiệu quả hơn.

Trong 1 lệnh có truy xuất bộ nhớ dữ liệu sẽ có kèm bit Acess RAM là thông số a trong lệnh.

Khi a = 1 dùng BSR và địa chỉ 8 bit trong mã giả.

a = 0 dùng Access bank, bỏ qua BSR.

Lập sơ đồ của Access bank khác biệt nhỏ khi tập lệnh mở rộng được kích hoạt.



1. Dãy thanh ghi đa năng (General purpose register file)

PIC18 có thể có bộ nhớ dữ liệu được làm trống trong vùng GPR.

Vùng GPR là vùng RAM dữ liệu có thể dùng bởi bất kỳ lệnh nào, nó không bị thay đổi bởi bất cứ reset nào và nó không bị Power on reset gán giá trị.

Vùng GPR bắt đầu từ Bank 0 (địa chỉ 0h) tới chỗ bắt đầu SFR

1. Các thanh ghi chức năng đặc biệt (special function register)

Các SFR là các thanh ghi được dùng bởi CPU và các module ngoại vi để điều khiển hoạt động của thiết bị.

Các thanh ghi này cũng là RAM tĩnh nằm trong vùng bộ nhớ dữ liệu từ địa chỉ F60h tới FFFh.

Các SFR có thể chia làm 2 tập hợp:

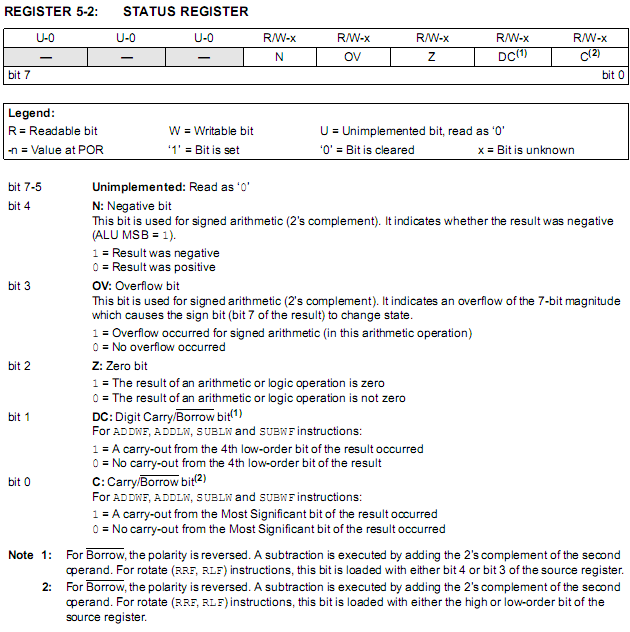
* Các thanh ghi phục vụ nhân vi điều khiển (khối ALU, reset, interrupt)
* Các thanh ghi chức năng ngoại vi.

1. Thanh ghi trạng thái (status register)

Thanh ghi trạng thái chứa trạng thái của khối số học của ALU.

Thanh ghi này cũng có thể là đích đến trong các lệnh nhưng thay vì ghi kết quả lệnh vào trong thanh ghi này thì chỉ có tác dụng của lệnh đó là làm thay đổi các bit trong thanh ghi này.

Ví dụ lệnh xóa CLRF STATUS : sẽ không làm các bit trong thanh ghi này thành 0 mà chỉ bít Z thành 1, các bit khác không ảnh hưởng.



D.Các chế độ định địa chỉ dữ liệu

Bộ nhớ chương trình chỉ có 1 cách định địa chỉ là thông qua bộ đếm chương trình (PC) nhưng bộ nhớ dữ liệu có nhiều cách tùy vào lệnh được thực thi, đó là 5 cách :

* HIển nhiên (inherent)
* Hằng số
* Trực tiếp
* Gián tiếp
* Hằng số sai lệch (indexed literal offset – khi kích hoạt tập lệnh mở rộng)

Hầu hết lệnh, chế độ địa chỉ là cố định. Các lệnh khác dùng lên tới 3 chế độ và phụ thuộc vào toán hạn được dùng hay tập lệnh mở rộng có được kích hoạt.

1. Định địa chỉ bằng hằng số hay theo cách hiển nhiên

Nhiều lệnh PIC18 không cần bất cứ đối số nào, các lệnh này có ảnh hưởng toàn bộ thiết bị hay hoạt động ngầm trên 1 thanh ghi cố định. Đây được xem là chế độ định địa chỉ hiển nhiên. VD Sleep, Reset và DAW

Các lệnh khác làm việc giống như vậy nhưng đòi hỏi thêm đối số rõ ràng trong mã giả. Đây là chế độ định địa chỉ bằng hằng số. VD ADDLW và MOVLW

1. Định địa chỉ trực tiếp

Chế độ này chỉ định tắt cả phần của địa chỉ nguồn hay đích thông qua các đối số trong lệnh.

Nguồn của dữ liệu :

* Trong tập lệnh PIC18, Các lệnh hướng bit và hướng byte dùng phiên bản định địa chỉ trực tiếp nào đó mặc định. Tất cả lệnh này bao gồm địa chỉ hằng số 8 bit là byte thấp nhất (LSByte), địa chỉ này chỉ rõ địa chỉ thanh ghi trong băng nào trong số các băng của RAM dữ liệu hay một vị trí trong Access bank là nguồn dữ liệu cho lệnh.
* bit Access RAM ‘a’ quyết định địa chỉ sẽ được dịch ra sao. Định địa chỉ dùng Access RAM còn được gọi là chế độ định địa chỉ áp đặt trực tiếp (Direct Forced Addressing mode).
* Một ít lệnh, như MOVFF, đính kèm 12 bit địa chỉ trong mã và trong trường hợp này thì BSR bị bỏ qua hoàn toàn.

Nơi đến của kết quả :

Được xác định bởi bit ‘d’.

‘d’ = 1 kết quả lưu vào thanh ghi nguồn

‘d’ = 0 kết quả lưu trong thanh ghi W.

Lệnh không có ‘d’ , có nơi đến được hiểu ngầm trong lệnh và là thanh ghi mục tiêu đang hoạt động trên hay thanh ghi W

1. Định địa chỉ gián tiếp

Định địa chỉ gián tiếp là truy cập bộ nhớ dữ liệu bằng cách dùng thanh ghi chọn dãy (File Select Register – FSR) làm con trỏ tới nơi cần.

FSR là SFR trong RAM dữ liệu nên có thể thao tác trực tiếp dưới sự điều khiển chương trình. Do đó hữu dụng trong việc tao cấu trúc dữ liệu, như bảng hay mảng trong bộ nhớ dữ liệu.

Các thanh ghi để định địa chỉ gián tiếp còn được hoạt động với toán hạng dãy gián tiếp (Indirect file operands – INDFs) cho phép thao tác tự động giá trị con trỏ : tự động tăng, tự động giảm hay độ lệch với giá trị khác. Điều nay cho phép mã lệnh hiệu suất cao, dùng vòng lặp.

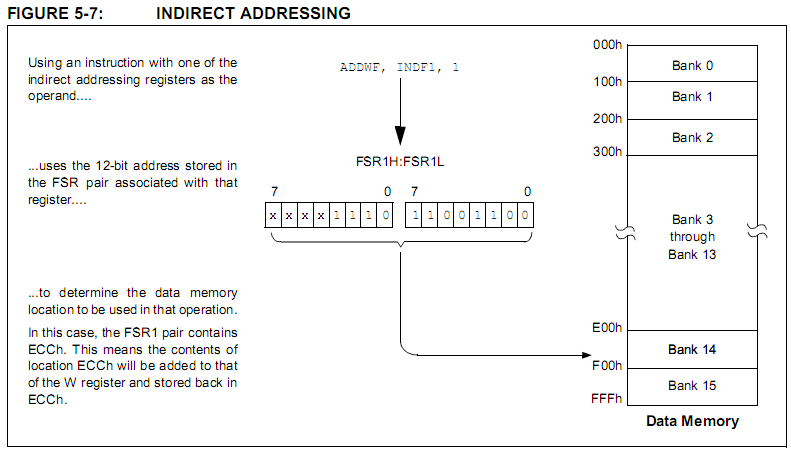
1. Các thanh ghi FSR và toán hạng INDF

Cốt Lõi của định địa chỉ gián tiếp là 3 thanh ghi FSR0, FSR1 và FSR2. Mỗi cái gồm 1 cặp thanh ghi 8 bit FSRnL và FSRnH. Bốn bit cao của thanh ghi FSRnH không được dùng như vậy mỗi thanh ghi FSRn có 12 bit đủ để định địa chỉ kiểu tuyến tính cho toàn bộ bộ nhớ dữ liệu

Định địa chỉ gián tiếp đạt được với tập các toán hạng dãy gián tiếp INDF0 tới INDF2, có thể xem như các thanh ghi ảo vì chúng được sơ đồ hóa trong không gian SFR của bộ nhớ dữ liệu nhưng không có.

Đọc hay ghi một “thanh ghi” INDF cụ thể thật sự truy cập cặp thanh ghi FSR tương ứng.

Ví dụ : Đọc từ INDF1, sẽ đọc dữ liệu tại địa chỉ được chỉ ra bởi FSR1H:FSR1L.



1. Các toán hạng khác của FSR

Ngoài INDF, mỗi FSR có thêm các toán hạng gián tiếp khác là POSTINC, POSTDEC, PREINC, PLUSW.

POSTINC truy cập giá trị trong FSR, sau đó tăng giá trị trong FSR lên 1

POSTDEC truy cập giá trị trong FSR, sau đó giảm giá trị trong FSR xuống 1

PREINC tăng giá trị trong FSR lên 1, rồi dùng giá trị trong FSR cho lệnh

PLUSW cộng giá trị FSR với giá trị thanh ghi W (có dấu -127 tới 128), rồi dùng giá trị trong FSR cho lệnh

1. Truy cập FSR bằng FSR

Dùng FSR để trỏ tới một thanh ghi ảo (INDFn) sẽ không thực hiện được.

Vd : nếu dùng FSR0H:FSR0L (chứa FE7h là địa chỉ INDF1) đọc INDF1 thì chỉ ra 0, nếu ghi thì chỉ có 1 lệnh NOP được thực thi

Dùng thanh ghi ảo để ghi vào FSR sẽ không hoàn toàn như ý muốn

Vd : nếu ghi vào FSR2 bằng cách ghi vào INDR2 hay POSTDEC2 sẽ vẫn ghi giá trị vào FSR2 nhưng trong 2 trường hợp giá trị FSR2 giống nhau.

Vì FSR là thanh ghi vật lý được sơ đồ hóa trong không gian SFR nên có thể bị thao tác trực tiếp thông qua mọi hoạt động. Vì vậy cần cẩn trọng nếu có dùng chế độ địa chỉ gián tiếp.

1. Bộ nhớ chương trình Flash

Bộ nhớ chương trình flash có thể đọc, ghi và xóa trong quá trinh vận hành bình thường với toàn bộ khoảng điện áp VDD

Đọc 1 byte 1 lần, ghi từng khối 32 byte 1 lần, xóa từng khối 64 byte một lần.

Bộ nhớ flash không thể truy cập được trong khi ghi hay xóa nên vi điều khiển cũng không thể thực thi mã lệnh.

Có một bộ định thời lập trình nội sẽ ngừng việc ghi hay xóa flash.

Giá trị được ghi vào bộ nhớ flash không cần phải là một lệnh đúng, khi thực thi vị trí nhớ không phải là lệnh đúng thì sẽ thành lệnh NOP.

1. Đọc và ghi bảng (table reads and table writes)

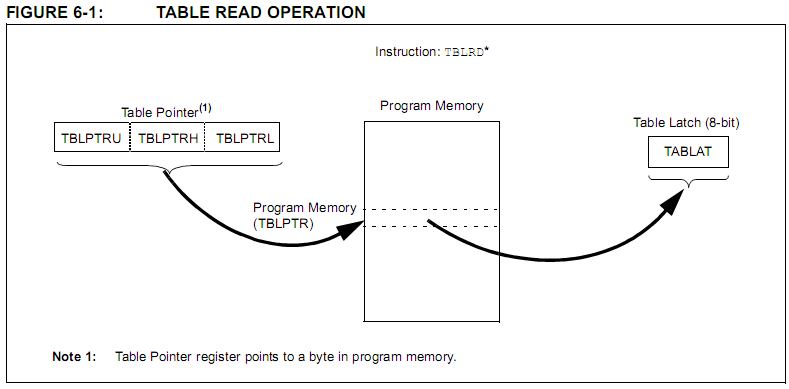
Để đọc ghi bộ nhớ flash dùng 2 lệnh sau

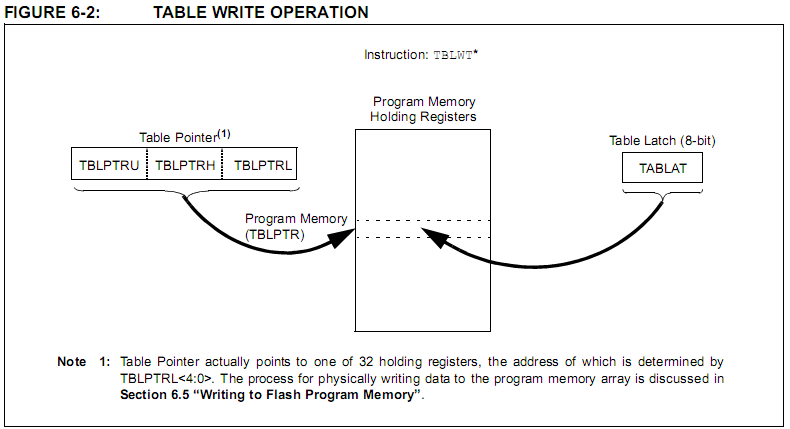
Table Read(TBLRD) lấy dữ liệu từ bộ nhớ flash để vào bộ nhớ dữ liệu

Table Write(TBLWT) lấy dữ liệu từ bộ nhớ dữ liệu lưu vào các thanh ghi giữ (holding registers)của bộ nhớ flash.

Quá trình hoạt động của bảng làm việc với thực thể byte. Dữ liệu chứa trong bảng không cần sắp xếp theo word

, nếu dùng bảng để ghi mã thực thi vào bộ nhớ flash, các lệnh của chương trình sẽ cần sắp xếp theo word.





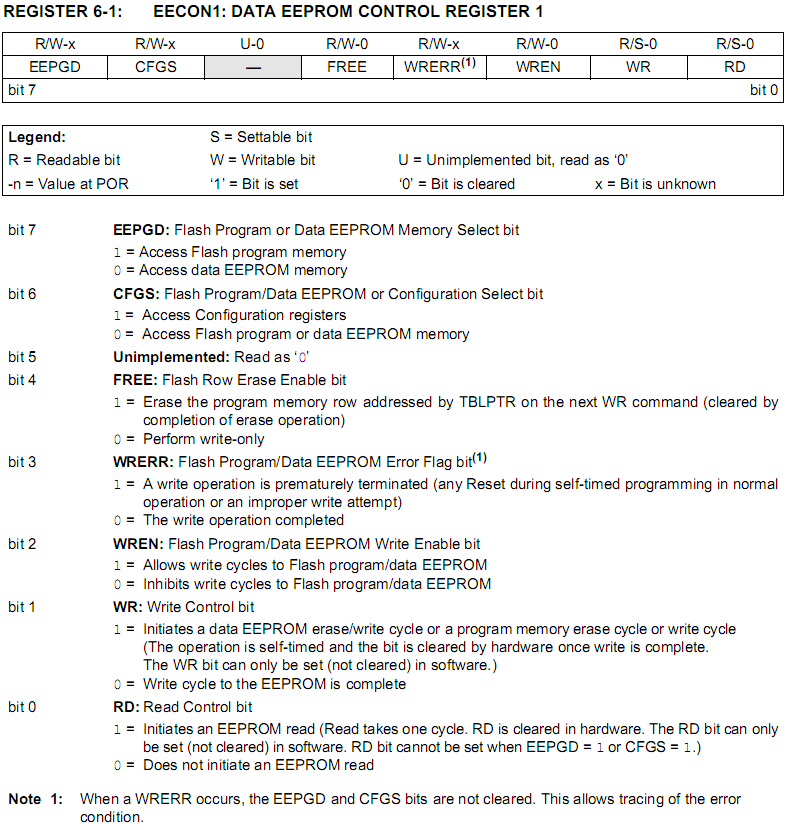
1. Thanh ghi điều khiển

Các thanh ghi được dùng trong liên kết với lệnh TBLRD và TBLWT gồm :

* EECON1
* EECON2
* TABLAT
* TBLPTR
  1. Thanh ghi EECON1 và EECON2

EECON1 là thanh ghi điều khiển các truy cập bộ nhớ.

EECON2 không phải là thanh ghi vật lý, được dùng chuyên biệt cho quá trình ghi và xóa bộ nhớ flash. Đọc thanh ghi EECON2 sẽ ra ‘0’.



* 1. Thanh ghi chốt bảng (table latch register – TABLAT)

TABLAT là thanh ghi 8 bit được sơ đồ hóa trong không gian SFR, được dùng giữ dữ liệu 8 bit trong quá trình truyền dữ liệu giữa bộ nhớ flash và Ram dữ liệu.

* 1. Thanh ghi con trỏ bảng (table pointer register – TBLPTR)

TBLPTR định địa chỉ 1 byte trong bộ nhớ flash, gồm 3 thanh ghi :

Table pointer upper byte : table pointer high byte : table pointer low byte (TBLPTRU:TBLPTRH:TBLPTRL)

Ba thanh ghi này kết hợp lại tạo thành con trỏ rộng 22 bit : 21 bit thấp cho phép truy cập 2 Mbyte bộ nhớ flash, bit thứ 22 cho phép truy cập tới Device ID, user ID và các bit cấu hình.

TBLPTR được dùng bởi lệnh TBLRD và TBLWT và 2 lệnh này chỉ gây ảnh hưởng lên 21 bit của thấp theo 4 cách:



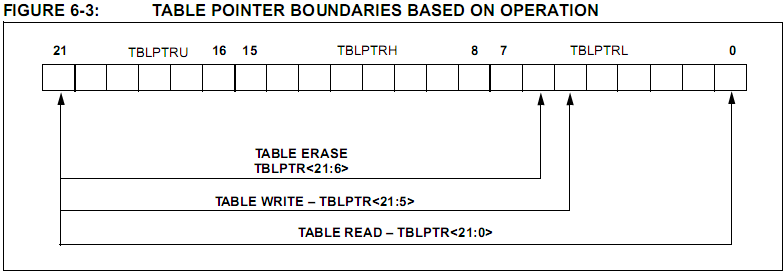
* 1. Giới hạn của con trỏ bảng (table pointer boundaries)

TBLPTR được dùng trong đọc, ghi và xóa bộ nhớ flash.

Khi TBLRD được thực thi, cà 22 bit của TBLPTR quyết định byte nào được đọc từ bộ nhớ flash vào TABLAT.

Khi TBLWT được thực thi, 5 bit thấp nhất của TBLPTR<4:0> quyết định thanh ghi giữ (holding register) trong số 32 thanh ghi sẽ được ghi vào. Khi quá trình ghi vào bộ nhớ flash bắt đầu (bằng cách set bit WR), 16 bit của TBLPTR<21:6> sẽ quyết định khối 32 byte nào của bộ nhớ flash được ghi vào.

Khi xóa , 5 bit thấp TBLPTR<5:0> bị bỏ qua, 16 bit TBLPTR<21:6> chỉ ra khối 64 byte sẽ được xóa.

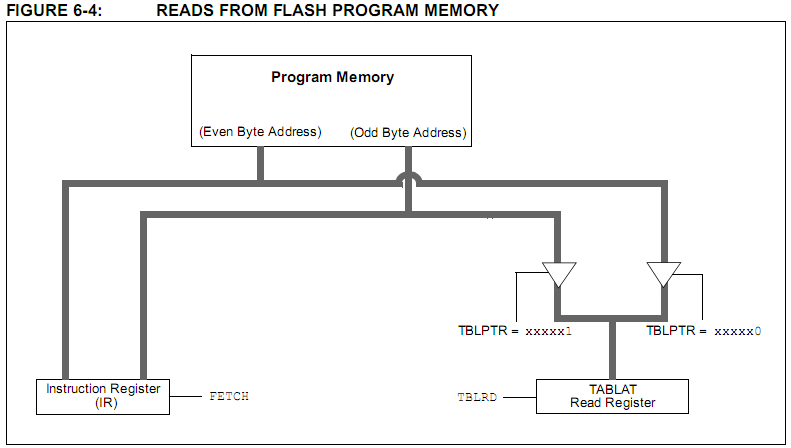


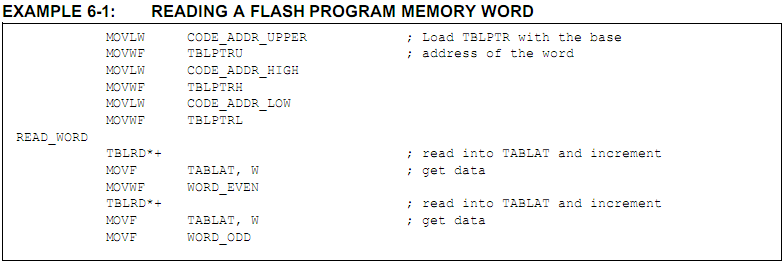
1. Đọc bộ nhớ chương trình flash

Dùng lệnh TBLRD để đọc bộ nhớ flash.

Địa chỉ của byte được đọc trong TBLPTR. Ta có thể vừa đọc vừa tăng giá trị của TBLPTR cho lần đọc sau.

Chú ý là dùng LSb của TBLPTR để đọc được byte chẵn hay lẻ mong muốn vì bộ nhớ thường được tổ chức theo word





1. Xóa bộ nhớ chương trình flash

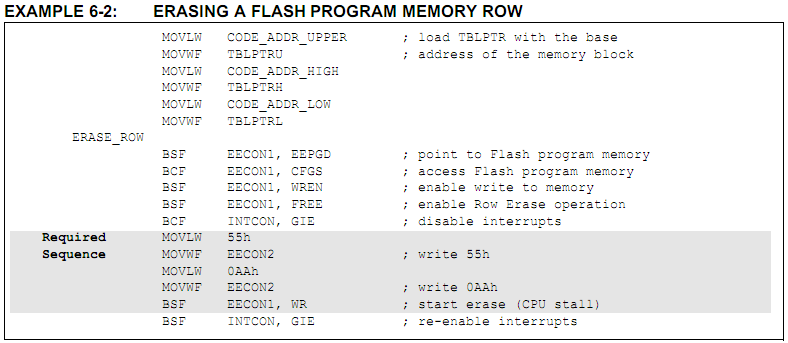
Khối xóa nhỏ nhất là 64 byte. Chỉ có dùng bộ lập trình ngoài hay thông qua điều khiển ICSP mới có thể xóa khối lớn hơn. Xóa 1 từ trong mảng flash không được hỗ trợ.

Địa chỉ của khối sẽ xóa lưu trong 16 bit TBLPTR<21:6>, 6 LSb của TBLPTR bị bỏ qua

Thanh ghi EECON1 điều khiển quá trình xóa.

Trình tự xóa như sau:

* Nạp địa chỉ hàng cần xóa vào TBLPTR<21:6>
* Thiết đặt thanh ghi EECON1 để xóa:
  + Set bit EEPGD để trỏ tới bộ nhớ flash
  + Xóa bit CFGS để truy cập bộ nhớ flash
  + Set bit WREN để cho phép ghi vào flash
  + Set bit FREE để cho phép xóa
* Tắt các ngắt
* Ghi 055h vào EECON2
* Ghi 0Aah vào EECON2
* Set bit WR để bắt đầu xóa
* CPU ngừng chạy trong khoảng 2ms
* Bặt lại các ngắt



1. Ghi bộ nhớ chương trình flash

Khối ghi nhỏ nhất là 32 byte. Lập trình 1 byte hay 1 word không được hỗ trợ.

Lệnh TBLWT sẽ nạp dữ liệu từ data RAM vào TABLAT rồi vào thanh ghi giữ. (có 32 thanh ghi giữ, 1 byte/thanh)

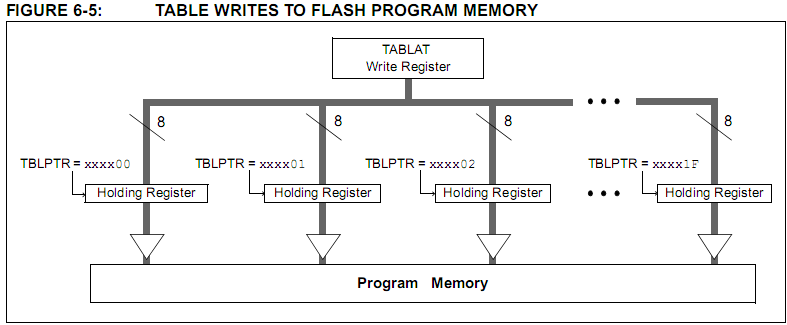
Vì TABLAT là thanh ghi 8 bit. Lệnh TBLWT cần phải thực hiện 32 lần cho mỗi lần ghi. Quá trình ghi vào thanh ghi giữa là ngắn – short writes. Sau khi ghi vào 32 thanh ghi giữ ta sẽ thiết đặt EECON1 để bắt đầu ghi từ thanh ghi giữ vào bộ nhớ flash – đây là ghi dài long writes.

Việc thực thi lệnh bị treo trong chu kỳ ghi dài (long write cycle). Quá trình ghi dài cũng bị ngắt bởi bộ đinh thời được lập trình gắn trong.

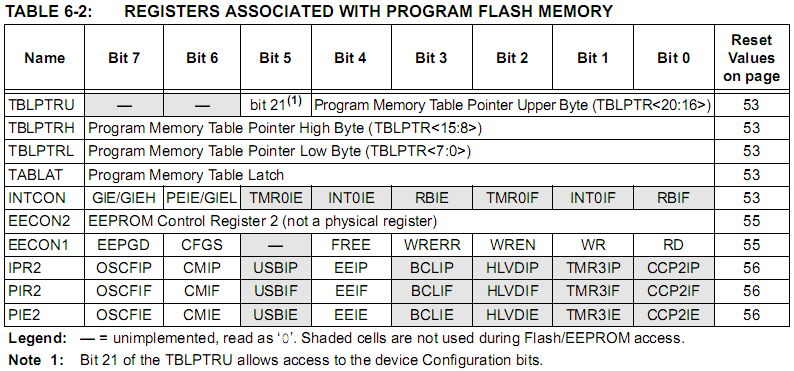
Bộ định thời trên chịp EEPROM điều khiển thời gian ghi.

Điên áp ghi xóa định mức cao hơn khoảng điện áp của thiết bị.

Giá trị mặc định của các thanh ghi giữ sau khi reset thiết bị và sau khi ghi là FFh. Ghi FFh vào thanh ghi giữ không thay đổi byte đó. Nghĩa là ta có thể thay đổi một vài byte nào đó ta cần, không nhất thiết phải nạp lại cho cả 32 thanh ghi giữ trước khi ghi vào bộ nhớ flash.



1. Trình tự ghi vào bộ nhớ flash
2. Đọc 64 byte vào RAM
3. Cập nhật giá trị trong RAM nếu cần
4. Nạp vào TBLPTR<21:6> địa chỉ khối bị xóa (để lấy chỗ ghi cái mới)
5. Thực hiện thủ tục xóa hàng (Row Erase procedure)
6. Nạp địa chỉ của thanh ghi sẽ chứa byte đầu tiên cần ghi vào TBLPTR<5:0>
7. Ghi 32 byte vào thanh ghi giữ và tự động tăng TBLPTR (TBLWT\*+)
8. Thiết đặt thanh ghi EECON1 để ghi từ thanh ghi giữ vào bộ nhớ flash
   1. Set bit EEPGD để trỏ tới bộ nhớ flash
   2. Xóa bit CFGS để truy cập bộ nhớ flash
   3. Set bit WREN để cho phép ghi byte
9. Tắt ngắt
10. Ghi 055h vào EECON2
11. Ghi 0Aah vào EECON2
12. Nạp vào TBLPTR<21:6> Địa chỉ bắt đầu khối (trong bộ nhớ flash) cần ghi vào
13. Set bit WR để bắt đầu ghi xuống bộ nhớ flash
14. CPU sẽ ngừng chạy trong khoảng 2 ms
15. Bật lại ngắt
16. Lập lại bước 6 tới 14 mộ lần nữa để ghi tiếp 32 byte nữa là được 64 byte
17. Kiểm tra bộ nhớ flash
18. Kiểm tra việc ghi

Cần thực hiện kiểm tra ghi tùy theo ứng dụng

1. Ngắt trong quá trình ghi

Nếu việc ghi bị gián đoạn do mất điện hay reset không mong muốn cần phải kiểm tra lại dùng bit WRERR

1. Bảo vệ chống ghi sai (xem 25.0)
2. Vận hành của flash trong chế độ bảo vệ mã (xem 25.5)
3. Bộ nhớ EEPROM dữ liệu

EEPROM dữ liệu là một mảng bộ nhớ không bay hơi, tách riêng với RAM dữ liệu và bộ nhớ chương trình, được dùng lưu trữ lâu dài dữ liệu chương trình.

EEPROM không được sơ đồ hóa trực tiếp trong dãy thanh ghi hay không gian bộ nhớ chương trình nhưng được định địa chỉ gián tiếp thông qua SFRs.

EEPROM có thể đọc ghi trong quá trình vận hành bình thường trên toàn khoảng VDD

4 thanh ghi SFR dùng để đọc ghi EEPROM cũng như bộ nhớ chương trình là

* EECON1
* EECON2
* EEDATA chứa dữ liệu 8 bit
* EEADR chứa địa chỉ vị trí trong EEPROM đang được truy cập

EEPROM dữ liệu cho phép đọc ghi byte, được đánh giá có chu kỳ ghi xóa cao.

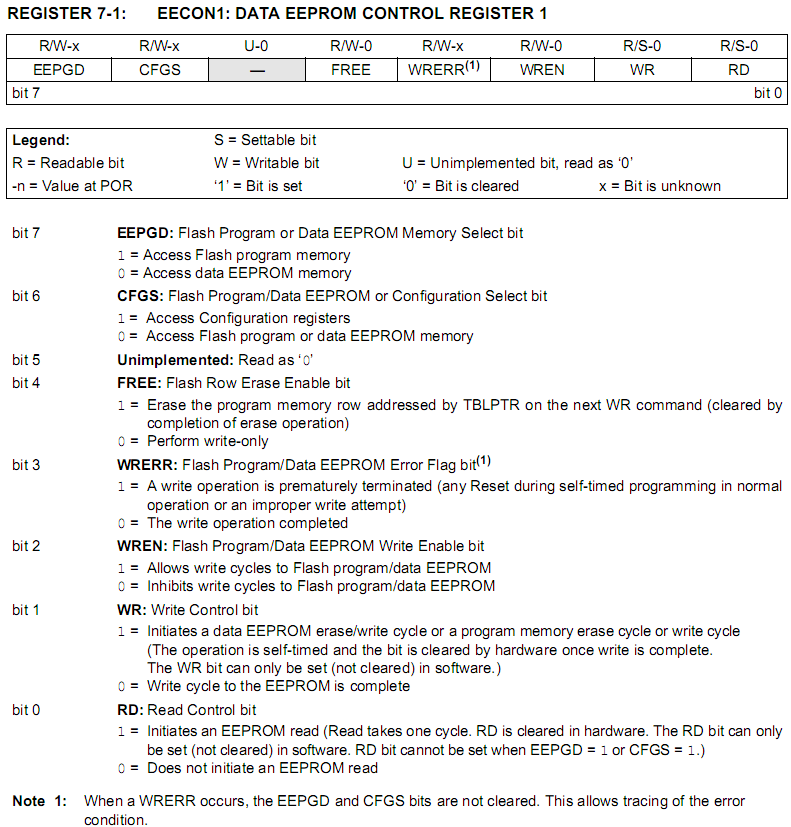
Ghi 1 byte sẽ tự động xóa vị trí sắp ghi và ghi dữ liệu mới vào (xóa - trước khi - ghi)

Thời gian ghi được kiểm soát bởi một timer trên chip (on-chip timer), khác nhau với điện thế, nhiệt độ cũng như giữa các chip với nhau

1. Thanh ghi EECON1 và EECON2

Truy cập EEPROM dữ liệu được kiểm soát bởi 2 thanh ghi : EECON1 và EECON2.

Đây cũng là cặp thanh ghi dùng truy cập bộ nhớ flash và chúng được dùng theo cách tương tự cho EEPROM dữ liệu.

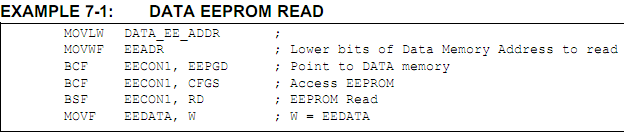


1. Đọc bộ nhớ EEPROM dữ liệu

Để đọc EEPROM :

* Nạp địa chỉ vị trí cần đọc vào thanh ghi EEADR
* Xóa bit EEPGD để chọn truy cập tới EEPROM
* Set bit RD để bắt đầu đọc
* Dữ liệu xong ở chu kỳ lệnh kế và chứa trong thanh ghi EEDATA

EEDATA sẽ giữ giá trị này cho tới khi có quá trình đọc khác được thực hiện hoặc là tới khi vị trí nhớ đó bị ghi đè vào một giá trị.



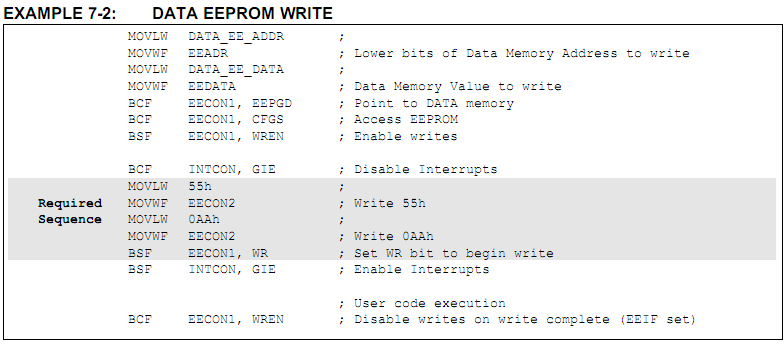
1. Ghi bộ nhớ EEPROM dữ liệu

Để ghi vào EEPROM

* Nạp địa chỉ ô cần ghi vào thanh ghi EEADR
* Nạp dữ liệu cần ghi vào thanh ghi EEDATA
* Xóa bit EEPGD để chọn truy cập tới EEPROM
* Xóa bit CFGS truy cập tới EEPROM
* Set bit WREN cho phép ghi
* Tắt ngắt
* Nạp 55h vào EECON2, nạp 0AAh vào EECON2
* Set bit WR để bắt đầu ghi
* Bật lại cách ngắt
* Xóa bit WREN để tắt chế độ ghi (tránh lỗi nhầm lẫn)

Chú ý là bit WR sẽ không thể set được nếu chưa set bit WREN, không thể set 2 bit này cùng lúc trong một lệnh.

Sau khi ghi xong bit EEIF (EEPROM interrupt flag bit) sẽ được set, có thể dùng bit này để ngắt hay thăm do và nó phải được xóa bằng phần mềm.



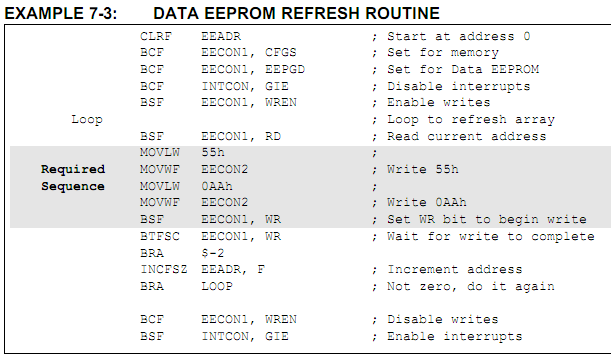
1. Hoạt động trong chế độ bảo vệ mã (code – protect)

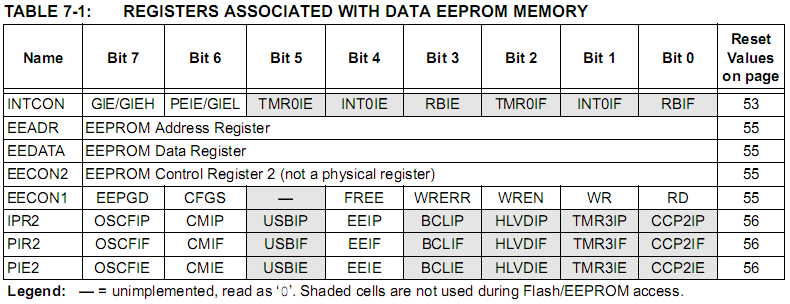
Bộ nhớ EEPROM có code-protect bit của riêng nó trong thanh ghi thiết đặt (xem 25.0), khi bật tính năng bảo vệ mã thì hoạt động đọc ghi từ bên ngoài bị tắt.

1. Sử dụng EEPROM dữ liệu

Bộ nhớ EEPROM có độ bền cao, có thể định địa chỉ theo byte và được tối ưu cho việc lưu trữ thông tin thay đổi thường xuyên (như các biến hay các dữ liệu cập nhật liên tục).

Các biến thay đổi không thường xuyên( hằng số, IDs, hiệu chình,…) nên lưu trong bộ nhớ flash.

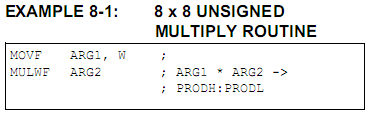
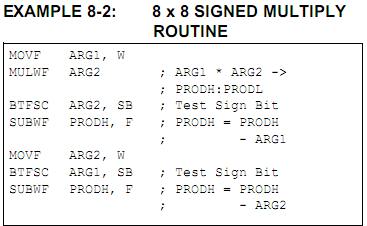


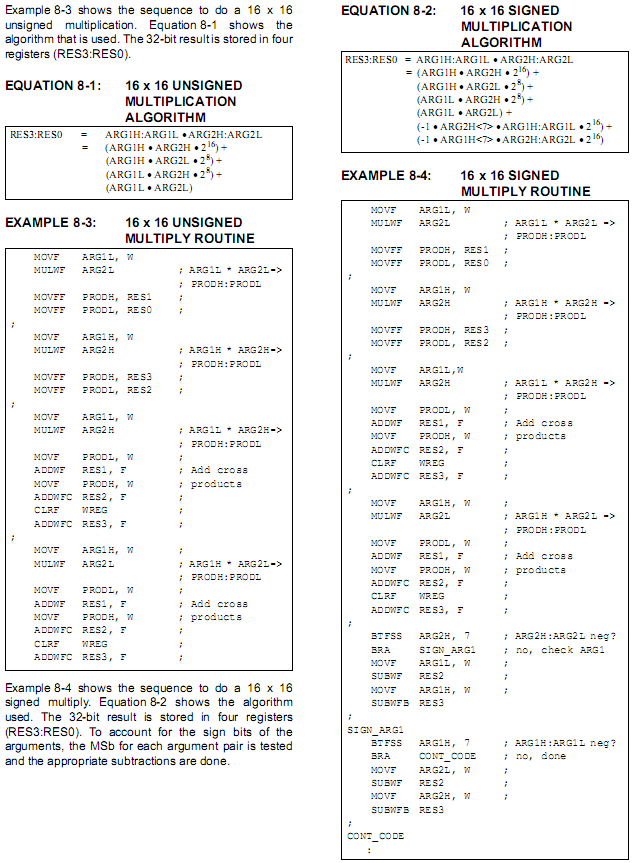


1. Bộ nhân 8 bit phần cứng

Trong tất cả PIC18 đểu có bộ nhân phần cứng 8x8 là một phần của bộ ALU. Khối này thực hiện phép nhân không dấu và tạo ra kết quả 16 bit lưu trong cặp thanh ghi tich (product register pair) PRODH:PRODL

Quá trình nhân bằng bộ này không ảnh hưởng tới thanh ghi trạng thái status register(cờ Z, OV,…)





1. Ngắt

PIC18F2550 có nhiều nguồn ngắt và tính năng ưu tiên ngắt.

Ngắt ưu tiên cao tại địa chỉ 08h, ngắt ưu tiên thấp tại địa chỉ 018h.

Có 10 thanh ghi để điều khiển ngắt :

* RCON
* INTCON, INTCON2, INTCON3
* PIR1, PIR2
* PIE1, PIE2
* IPR1, IPR2

Mỗi nguồn ngắt có 3 bit điều khiển vận hành :

* Bit cờ được set khi sự kiện ngắt đã diễn ra bất chấp bit cho phép, phải xóa bằng phần mềm trước khi cho phép ngắt và sau khi ngắt xảy ra.
* Bit cho phép ngắt
* Bit chọn mức ưu tiên ngắt

Tính năng ưu tiên ngắt được kích hoạt bằng bit IPEN (RCON<7>)

Khi bật tính năng ngắt ưu tiên, sẽ có 2 bit cho phép ngắt toàn cục :

* GIEH (INTCON<7>) cho phép các ngắt cao
* GIEL (INTCON<6>) cho phép các ngắt thấp

Khi tắt tính năng ngắt ưu tiên, các ngắt tương thích với các thiết bị PIC tầm trung, tức là chế độ tương thích, lúc này mọi nhánh ngắt đều định địa chỉ tại 08h

Bit INTCON<6> cho phép/tắt các nguồn ngắt ngoại vi

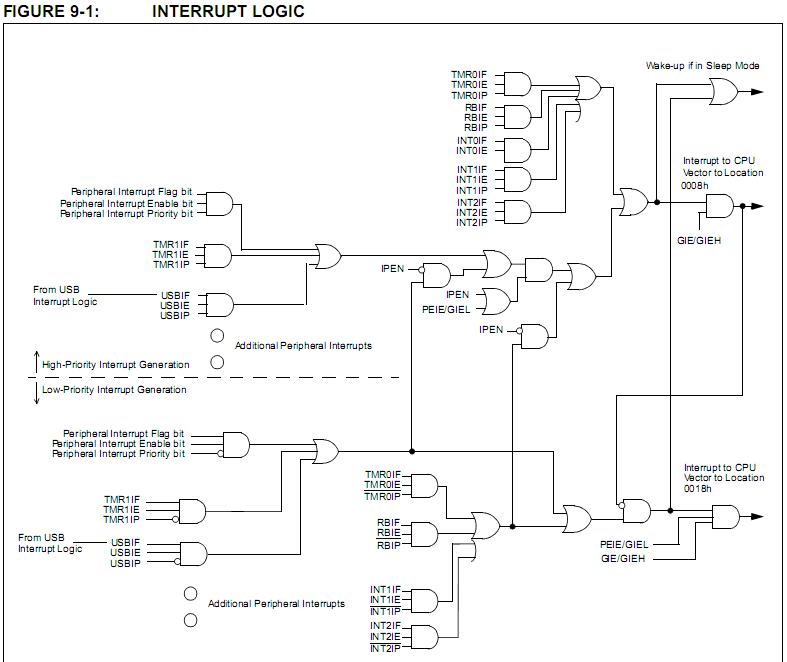
Bit INTCON<7> cho phép/tắt mọi nguồn ngắt

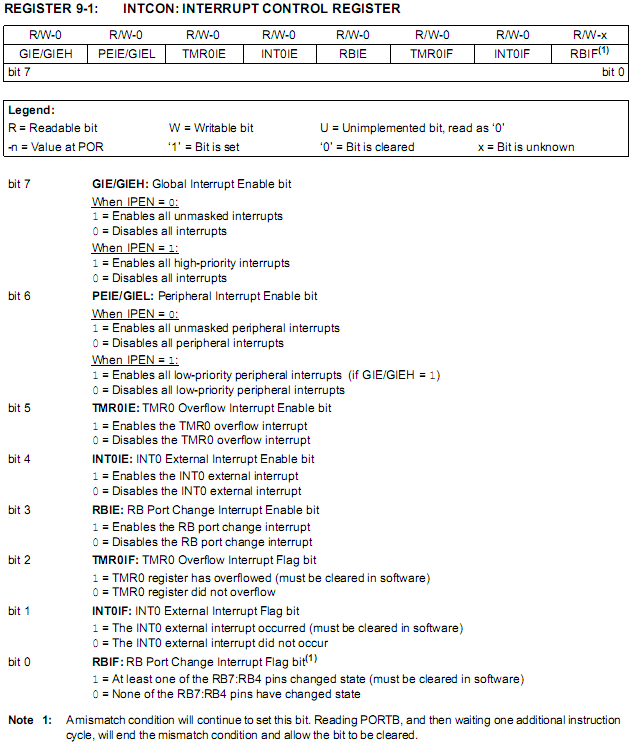
Khi đáp ứng ngắt :

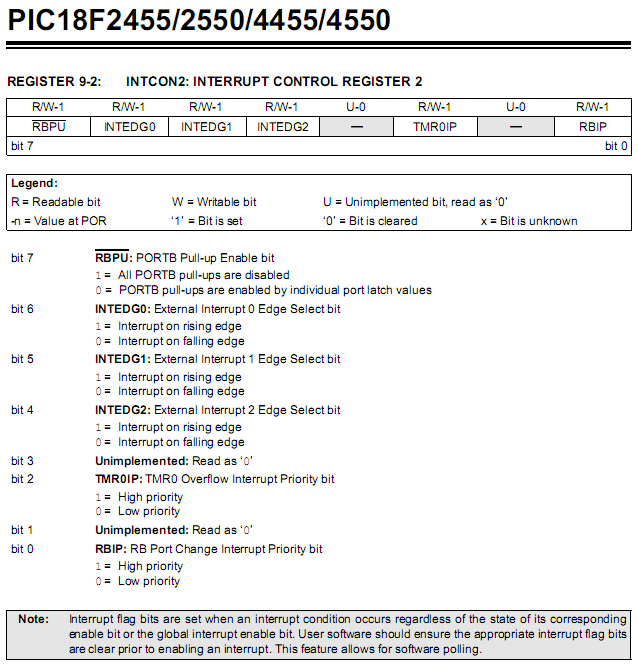
* Nếu
  + tắt tính năng ngắt ưu tiên, thì bit GIE bị xóa để cấm mọi nguồn ngắt khác.
  + bật tính năng ngắt ưu tiên, thì cả bit GIEH hay GIEL bị xóa.
* Địa chỉ trả về lưu vào stack và PC nạp địa chỉ ngắt (08h hay 018h), khi trong đoạn mã dịch vụ ngắt, nguồn ngắt có thể được thăm dò qua bit cờ ngắt, bit cờ phải được xóa bằng phần mềm để tránh ngắt đệ quy (recursive interrupts).
* Lệnh trờ về từ ngắt RETFIE sẽ thoát chương trình ngắt và set bit GIE, GIEH hay GIEL – cho phép ngắt lại.

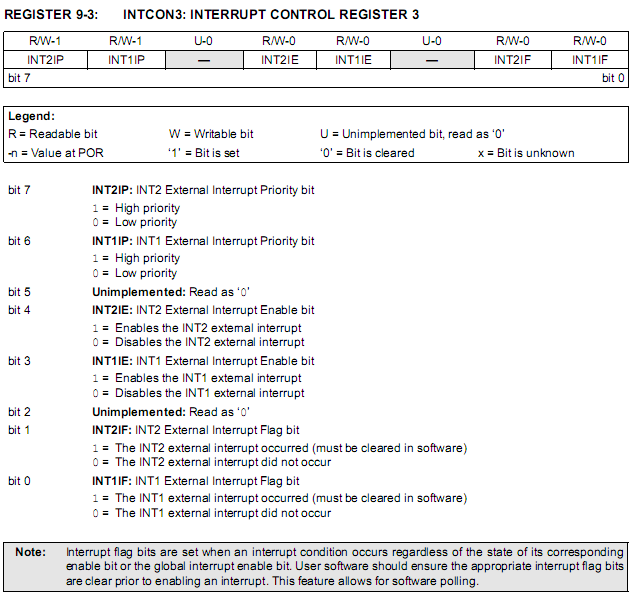
Với ngắt ngoài như chân INTx hay PORTB, sẽ có độ trển là 3 – 4 chu kỳ lệnh, bit cờ ngắt tương ứng sẽ được set bất chấp các bit cho phép hay bit GIE.

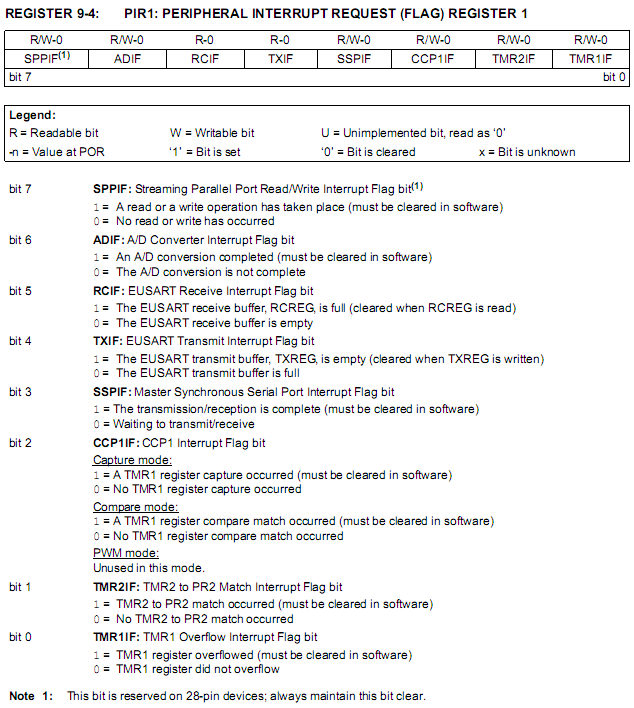
Lưu ý không dùng lệnh MOVFF để thay đổi bất kỳ thanh ghi điều khiển ngắt nào trong khi bất kỳ ngắt nào đang được cho phép vì sẽ làm vi điều khiển hoạt động sai.

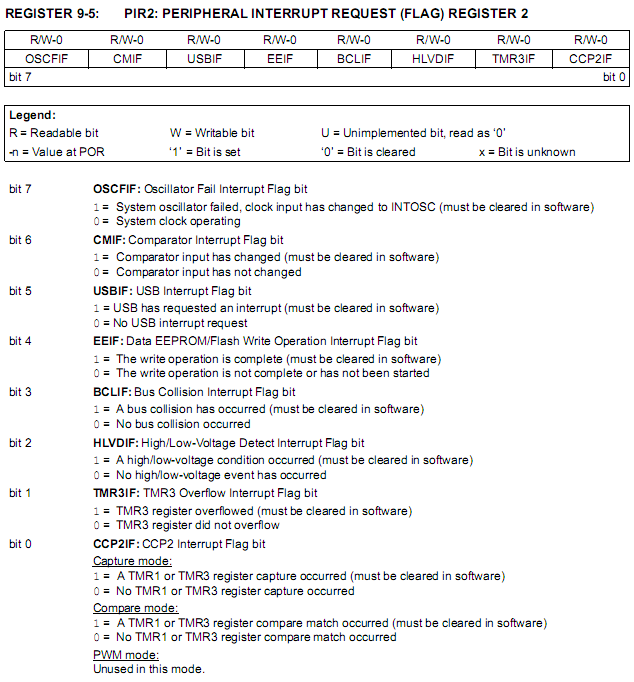


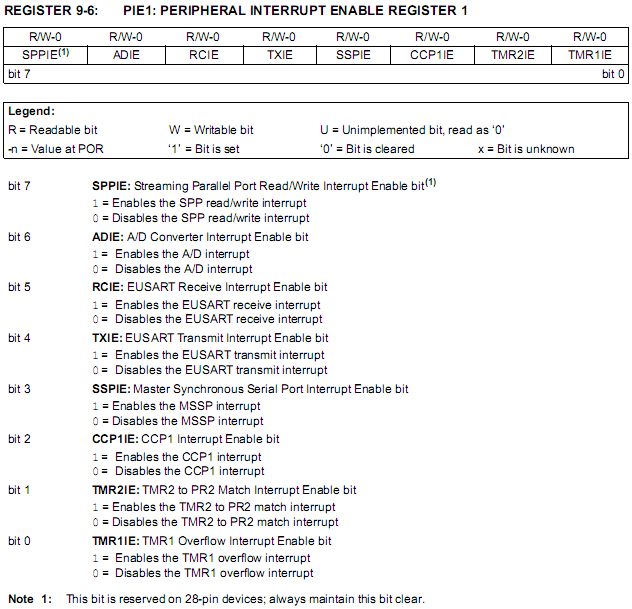


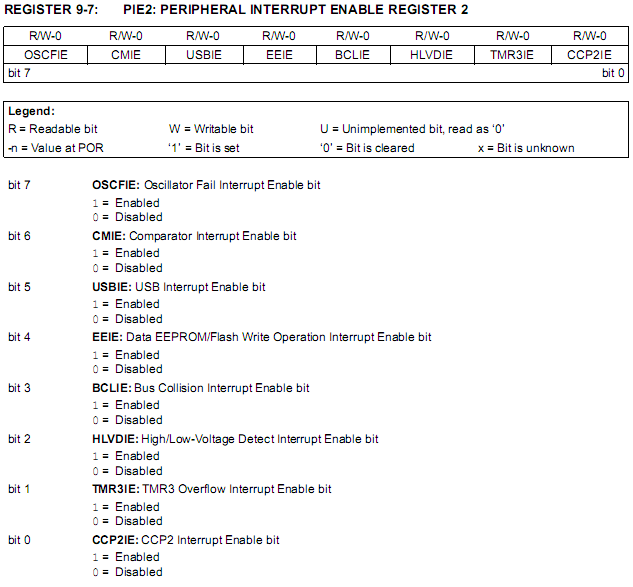


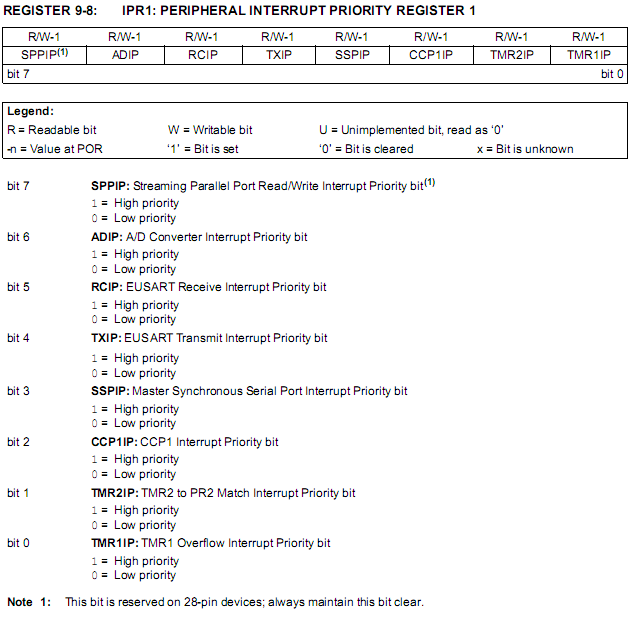


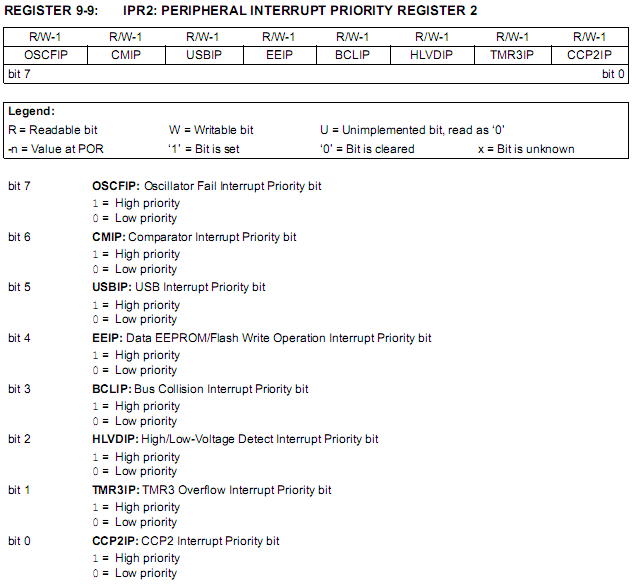


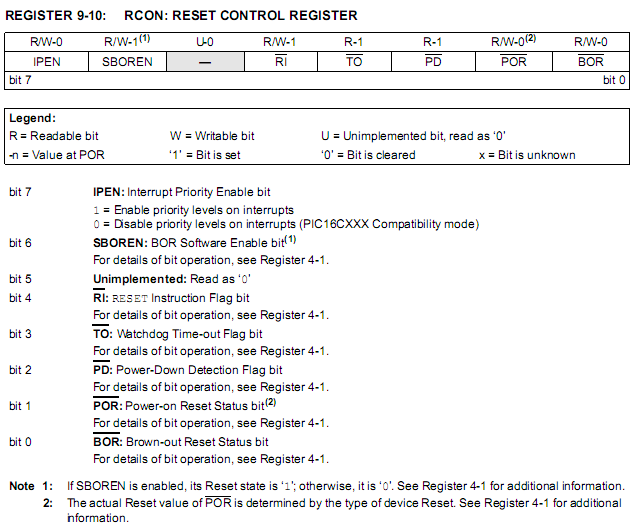












1. Các ngắt chân INTx

Các ngắt ngoài trên chân INT0, INT1, INT2 được kích bằng cạnh.

Nếu bit INTEDGx = 1 cạnh lên, INTEDGx = 0 cạnh xuống.

Bit cờ tương ứng là INTxIF phải được xóa trong đoạn chương trình con phục vụ ngắt

Bit cho phép tương ứng INTxIE.

Ngắt ngoài có thể đánh thức vi xử lý từ các chế độ quản lý năng lượng (power – managed modes) nếu bit INTxIE được set trước khi đi vào chế độ quản lý năng lượng. Nếu ngắt được cho phép thì sau khi thức dậy vi xử lý sẽ thực hiện chương trinh ngắt.

Ngắt ở INT0 luôn ưu tiên cao, không có bit ưu tiên cho INT0. INT1 và INT2 có bit ngắt ưu tiên là INT1IP và INT2IP

1. Ngắt TMR0

Ở chế độ 8 bit hay 16 bit thì khi đếm tràn (FFh->00h hay FFFFh->0000h) thì bit TMR0IF được set

Bit TMR0IE trong INTCON<5> cho phép ngắt

Bit TMR0IP trong INTCON2<2> chọn loại ưu tiên

1. Thay đổi khi ngắt ở PORTB(PORTB interrupt-on-change)

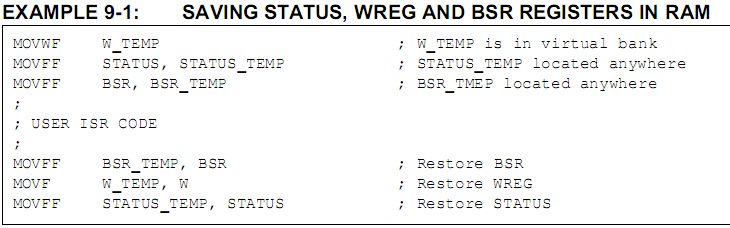
Khi ngõ vào thay đổi ở chân PORTB<7:4> sẽ set bit cờ RBIF trong <INTCON<0>)

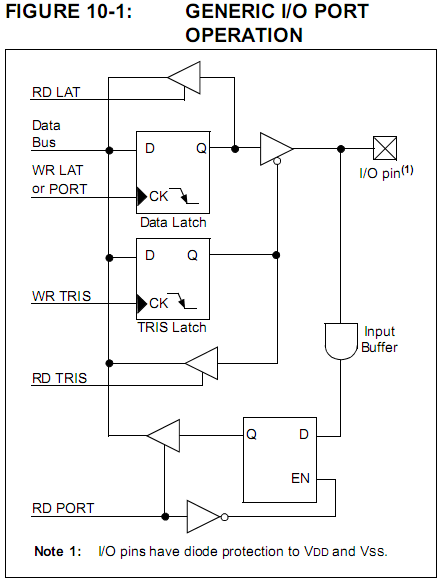
Bit RBIE trong INTCON<3> cho phép

Bit RBIP trong INTCON2<0> chọn loại ưu tiên

1. Việc lưu tạm các thanh ghi trong khi có ngắt

Trong khi ngắt diễn ra, địa chỉ PC trả về lưu trong stack và các thanh ghi WREG, STATUS, BSR được lưu vào ngăn xếp lưu nhanh Fast Return Stack. Người dùng ó thể cần lưu WREG, STATUS, BSR vào đầu chương trình con phục vụ ngắt hoặc cả những thanh ghi khác.





1. Cổng xuất nhập (I/O ports)

PIC18F2550 có 3 cổng xuất nhập có thể dùng. Một số chân được ghép kênh với các chức năng khác. Thông thường, khi ngoại vi được cho phép thì chân tương ứng sẽ không được dùng như chân xuất nhập đa năng.

Mỗi port có 3 thanh ghi để vận hành :

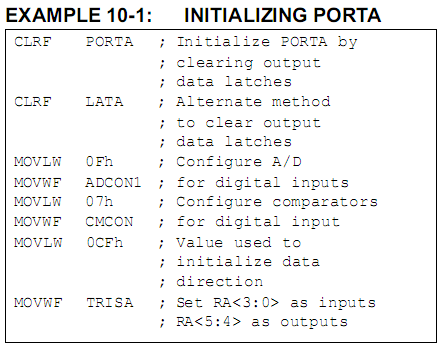
* TRIS thanh ghi chiều dữ liệu
* PORT thanh ghi đọc mức logic của chân
* LAT thanh ghi chốt ngõ ra

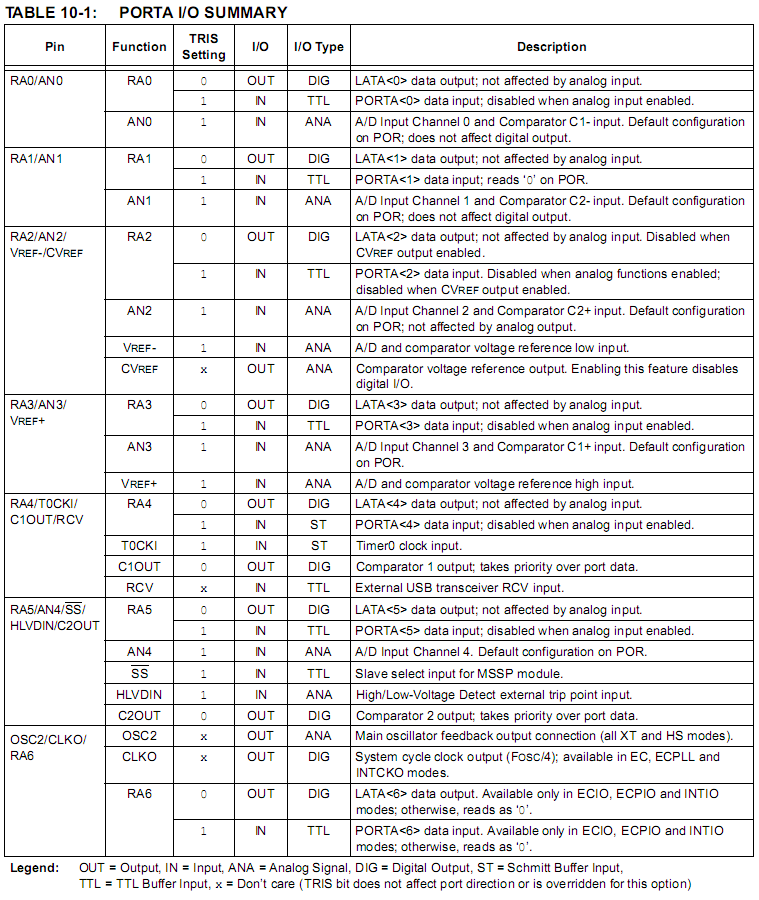
Thanh ghi chốt dữ liệu (Data latch register - LATA) hữu dụng cho quá trình đọc-sửa-ghi giá trị của chân I/O(read-modify-write operation)

Mô hình đơn giản hóa của cổng I/O chung, không có giao tiếp với các ngoại vi khác ở hình bên.

1. Cổng A

* Cổng A (PortA) là cổng 2 chiều rộng 8 bit. Thanh ghi chiều dữ liệu tương ứng là TRISA.
* Thiết đặt một bit của TRISA là 1 sẽ làm cho chân tương ưng là một ngõ vào (và làm cho bộ lái ngõ ra chân đó ở chế độ tổng trở cao).Khi xóa thì sẽ là ngõ ra (và đặt nội dung của chốt ngõ ra lên chân đó).
* Đọc thanh ghi PORTA là đọc trạng thái của các chân; ghi tới thì là ghi vào chốt của cổng (port latch).
* Thanh ghi chốt (LATA) cũng được sơ đồ hóa trong bộ nhớ. Quá trình đọc-sửa-ghi trên thanh ghi LATA sẽ đọc và ghi giá trị ngõ ra được chốt cho cổng A.
* Chân RA4 được ghép kênh với ngõ vào bộ Timer0 để trở thành chân RA4/T0CKI. RA4 cũng đồng thời được ghép kênh với bộ USB, được dùng như là ngõ vào bộ nhận từ một bộ truyền nhận USB ngoài.
* Chân RA6 được ghép kênh với chân bộ dao động chính; chân này là cổng I/O hay chân cho bộ dao động tùy vào thiết đặt bộ dao động. Khi không dùng như chân I/O thì chân RA6 và các bit liên kết TRIS và LAT đọc ra ‘0’.
* Nhiều chân cổng A được ghép kênh với ngõ vào tương tự , ngõ vào Vref+ và Vref- và ngõ ra bộ so sánh. Hoạt động của chân RA5 và RA3:RA0 như là ngõ vào bộ ADC được lựa chọn bằng cách xóa/set bit điều khiển trong thanh ghi ADCON1
* Tất cả ngõ ra cổng A có mức vào là TTL và bộ lái ngõ ra CMOS đầy đủ.
* Thanh ghi TRISA điều khiển chiểu của chân RA, thậm chí ngay cả khi dùng như ngõ vào tương tự. Do đó phải set nó cả khi dùng làm ngõ vào cho bộ ADC
* Khi reset bật nguồn, RA5 và RA3:RA0 là ngõ vào tương tự vả đọc ra ‘0’, RA4 là ngõ vào số.







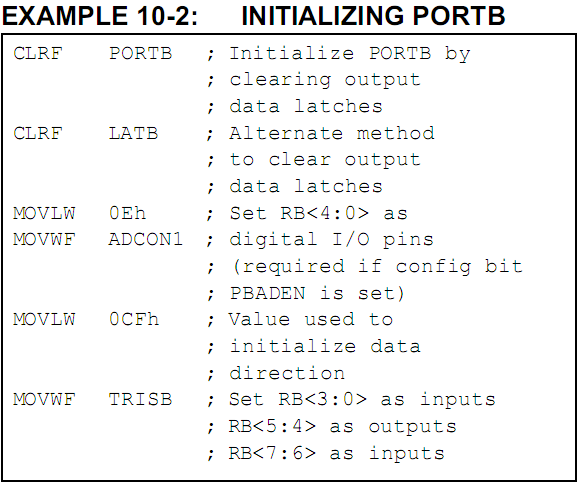
1. Cổng B

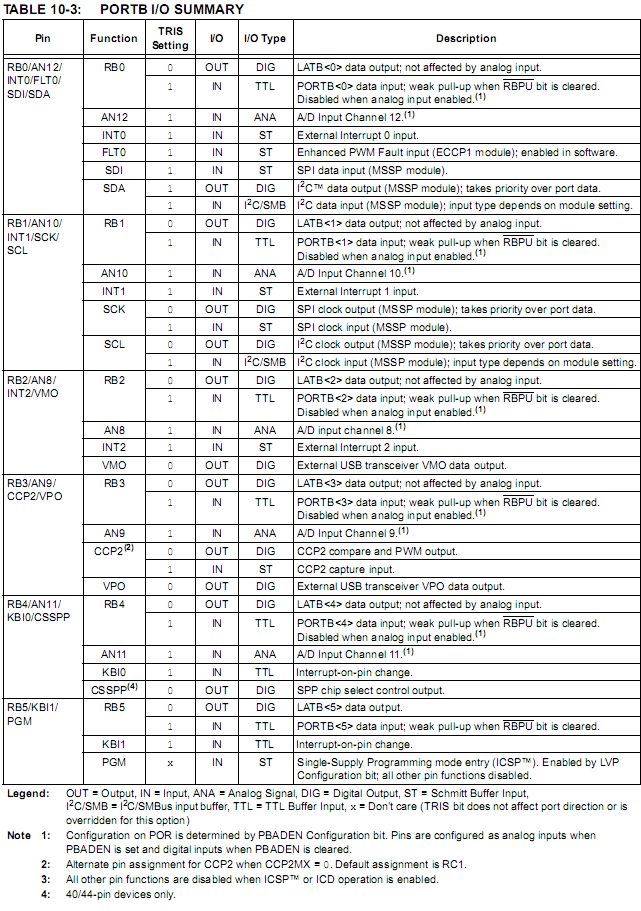
* Cổng B là cổng 2 chiều rộng 8 bit
* Giống cổng A vể các thanh ghi TRISB, LATB, PORTB.
* Mỗi chân cổng B có điện trở kéo lên ở trong yếu. Bit RBPU(INTCON2<7>) sẽ bật tất cả điện trở kéo lên và các điện trở được tắt tự động khi chân port làm ngõ ra. Điện trở kéo lên cũng được tắt khi reset bật nguồn (Power-on Reset)
* Lúc reset bật nguồn, RB4:RB0 được thiết đặt là ngõ vào tương tự và đọc ra ‘0’, RB7:RB5 được cấu hình là ngõ vào số.
* Bằng lập trình bit cấu hình PBADEN (CONFIG3H<1>), RB4:RB0 sẽ là ngõ vào số khi POR.
* Chân RB7:RB4 có tính năng ngắt khi có thay đổi (interrupt-on-change). Chỉ các chân được thiết đặt là ngõ vào có thể gây ra ngắt. Nếu chân nào trong RB7:RB4 được thiết đặt là ngõ ra thì không thể dùng tính năng này. Các chân sẽ được so sánh với giá trị cũ được chốt trong lần đọc trước đó của PORTB. Các chân sẽ được or với nhau để tạo ra ngắt thay đổi trên cổng RB với cờ hiệu là bit RBIF (INTCON<0>).
* Thay đổi khi có ngắt có thể được dùng đánh thức thiết bị từ chế độ ngủ. Người dùng, trong đoạn mã phục vụ ngắt có thể xóa ngắt theo cách sau :
  + Bất cứ sự đọc hay ghi PORTB( trừ lệnh MOVFF, PORTB) Nó sẽ chấm dứt điều kiện sai lệch.
  + Chờ một chu kỳ lệnh Tcy( ví sdụ thực thi lệnh NOP)
  + Xóa bit cờ RBIF

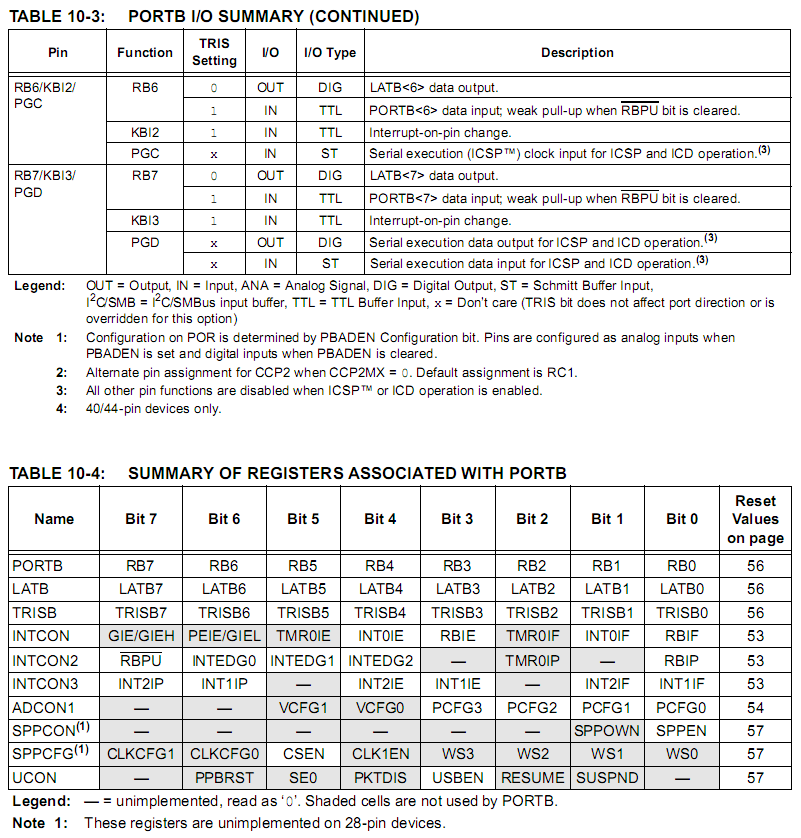
Điều kiện sai sẽ tiếp tục set bit cờ RBIF. Đọc cổng B sẽ làm kết thúc điều kiện sai và cho phép bit cờ RBIF được xóa sau một chu kỳ trì hoãn Tcy. Tính năng ngắt khi có thay đổi được khuyên cho hoạt động mà ở đó cộng B chỉ dùng tính năng thay đổi khi có ngắt.

Các chân, RB2 và RB3 được ghép kênh với ngoại vi USB và dùng như là ngõ ra tín hiệu vi sai cho bộ truyền nhận USB ngoài

RB4 được ghép kênh với CSSPP, chip chọn chức năng cho cổng song song dòng (streaming parallel port – SPP) – thiết đặt TRIS.

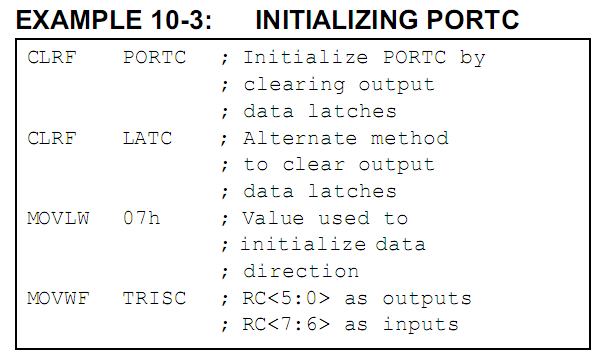


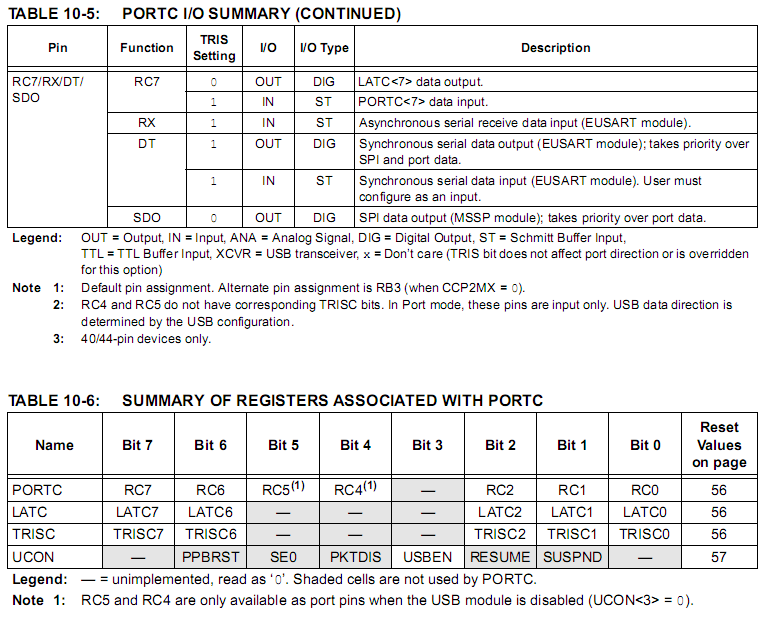


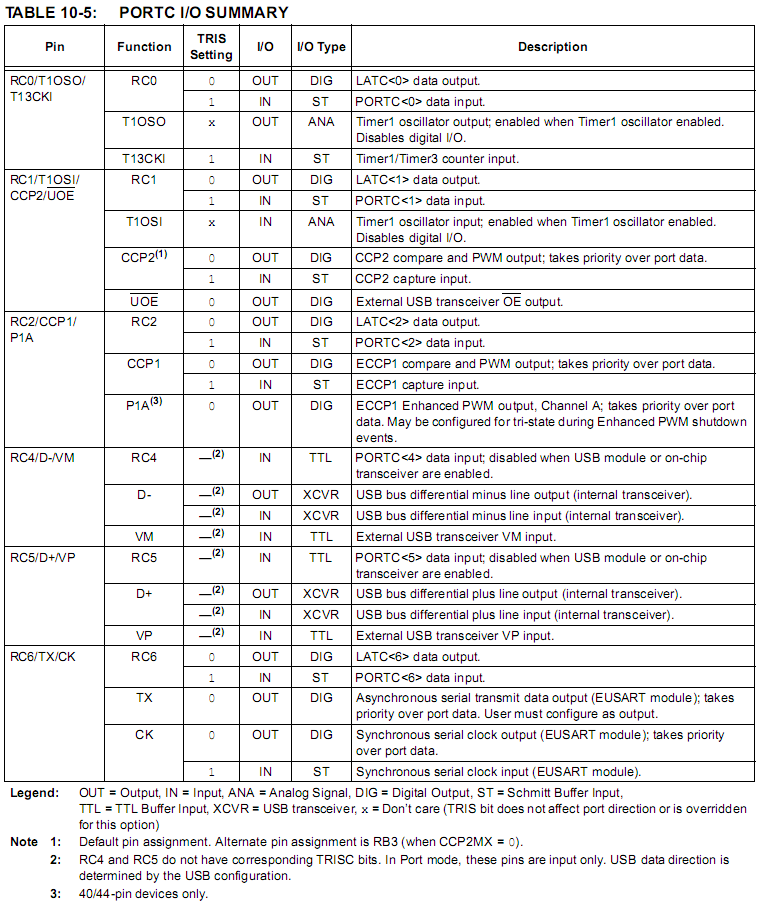


1. Cổng C

* Cổng C là cổng 2 chiểu rộng 7 bit.
* Giống cổng A vể các thanh ghi TRISB, LATB, PORTB.
* RC3 không được chế tạo.
* Cổng C được ghép kênh chính với bộ giao tiếp nối tiếp gồm EUSART, MSSP và USB. Trừ RC4 và RC5, cổng C dùng bộ đệm ngõ vào Schmitt trigger.
* Chân RC4 và RC5 được ghép kênh với bộ USB. Tùy vào cấu hình có thể là đường vào dữ liệu vi sai cho bộ truyền nhận USB trên chip hay ngõ vào dữ liệu từ bộ truyền nhận USB ngoài. Cả RC4 và RC5 có bộ đệm ngõ vào TTL thay cho bộ đệm Schmitt trigger trên các chân khác.
* Chân RC4 và RC5 không có bit TRISC. Khi là cổng số, chúng chỉ có thể là ngõ vào số. Khi cấu hình để hoạt động USB, chiều dữ liệu được quyết định bởi cấu hình và trạng thái của bộ USB tại thời điểm cho trước. Nếu bộ truyền nhận ngoài được dùng, RC4 và RC5 luôn là ngõ vào nhận từ bộ truyền nhận. Nếu bộ truyền nhận trên chip được dùng, chiều dữ liệu được quyết định ởi hoạt động đang được thực hiện bởi bộ truyền nhận tại thời điểm đó.
* Khi bộ truyền nhận ngoài được cho phép, RC2 cũng được dùng như ngõ ra cho phép điều khiển bộ truyền nhận.
* Khi cho phép các chức năng ngoại vi trên các chân cổng C khác chân RC4 và RC5, cần thận trọng xác định bit TRIS. Một vài ngoại vi sẽ bỏ qua bit TRIS và làm cho 1 chân là ngõ ra, trong khi các ngoại vi khác lại làm cho chân là ngõ vào. Người dùng nên tra cứu phần ngoại vi tương ứng để thiết đặt bit TRIS đúng.
* Khi có reset bật nguồn, các chân này trừ RC4 và RC5 được thiết đặt là ngõ vào số. Để dùng chân RC4 và RC5 là ngõ vào số luôn thì bộ USB phải bị tắt (UCON<3> = 0) và bộ truyền nhận USB trên chip phải bị tắt (UCFG<3> = 1)
* Nội dung của thanh ghi TRISC bị ảnh hưởng bởi sự lấn át của ngoại vi. Đọc TRISC luôn trả về nội dung hiện tại, mặc dù 1 thiết bị ngoại vi có thể đang chiếm lấy một hay nhiều chân.

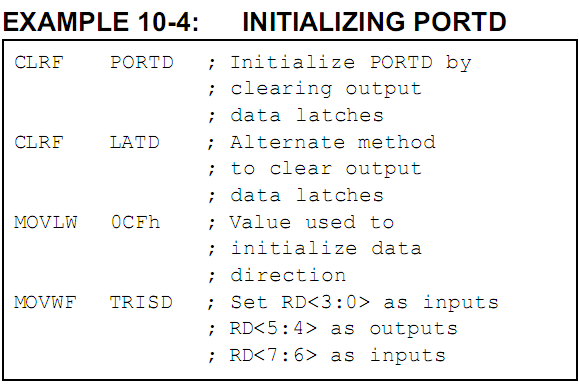




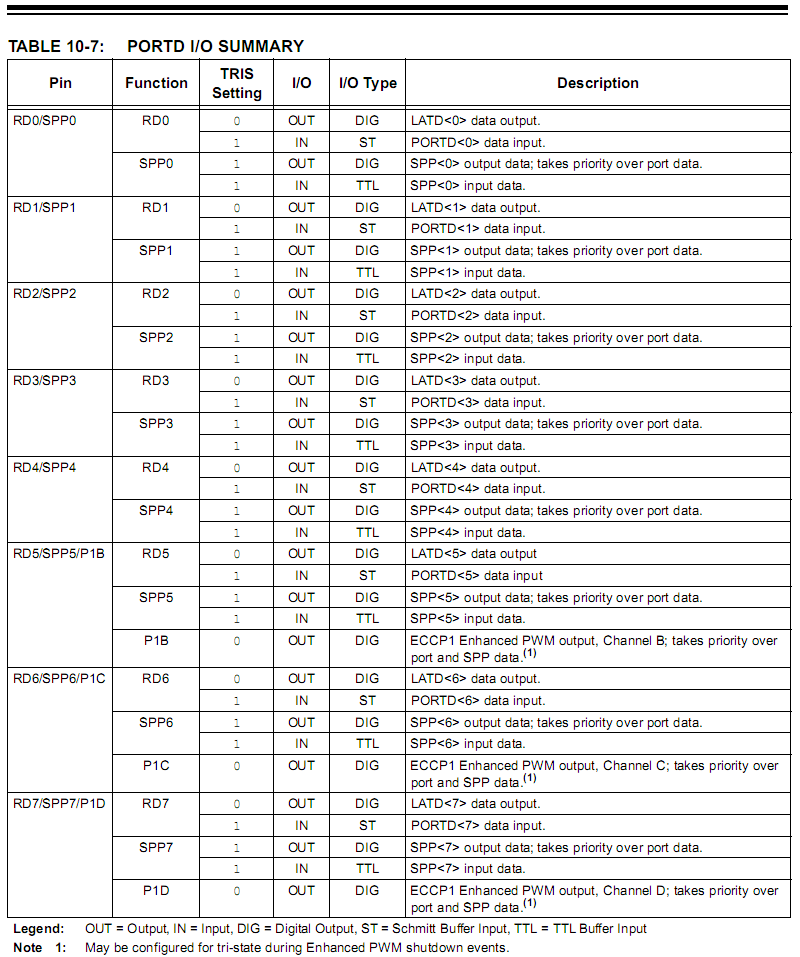


1. Cổng D

* Cổng D chỉ có trên con 18F4455/4550
* Giống cổng B
* Tất cả chân cổng D được chế tạo với bộ đệm ngõ vào Schmitt trigger. Mỗi chân được thiết đặt là ngõ vào hay ra riêng biệt.
* 3 chân của cổng D được ghép kênh với ngõ ra P1B, P1C và P1D của bộ CCP cài tiến (Enhanced CCP)
* Cổng D cũng có thể được thiết đặt là cổng song song dòng rộng 8 bit (SPP). Trong chế độ này, bộ đệm ngõ ra là TTL.
* Khi chế độ PWM cải tiến được dùng với 2 hay 4 ngõ ra, chức năng MSSP của cổng D tự động bị tắt.
* Khi reset bật nguồn, các chân cổng D được thiết đặt là ngõ vào số.







1. Cổng E

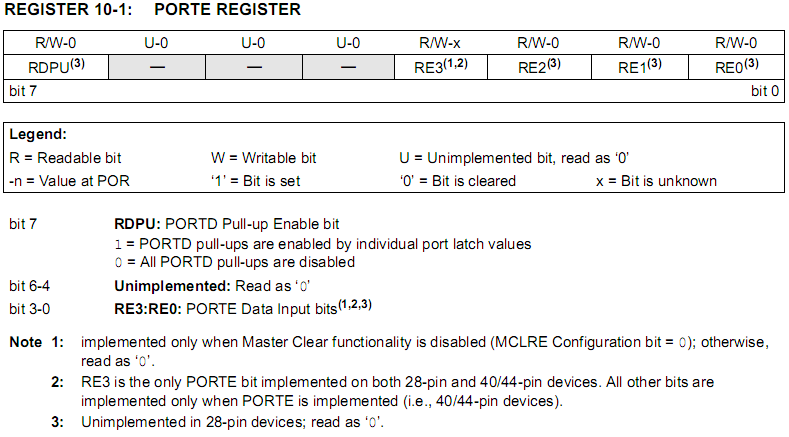
Tùy vào thiết bị cụ thể PIC18F2455/2550/4455/4550 được chọn, cổng E được chế tạo theo 2 cách.

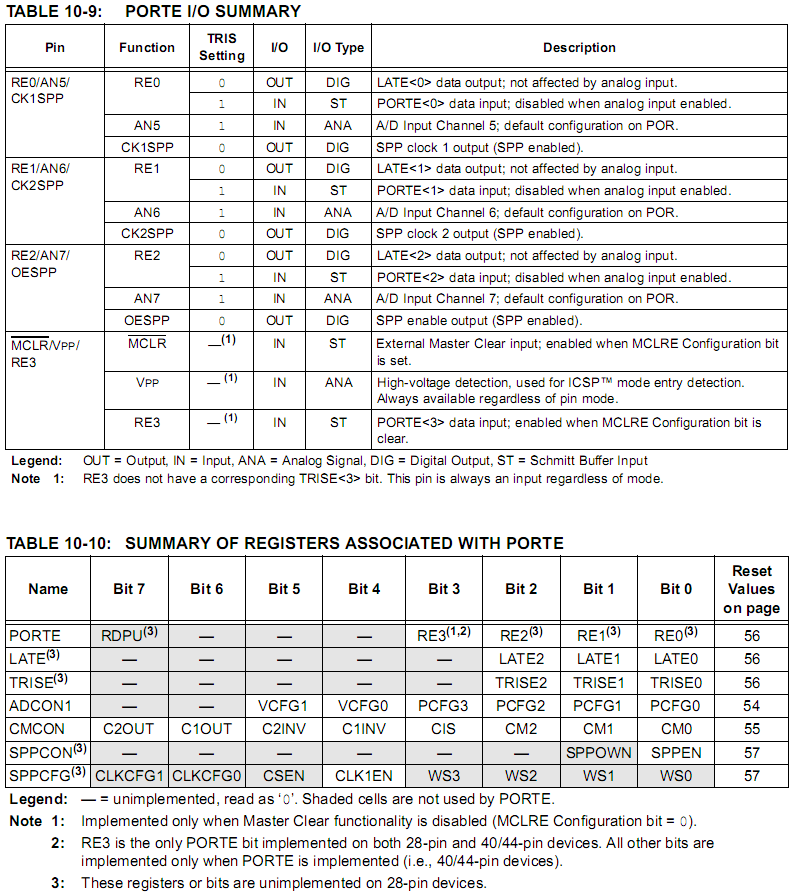
1. Với thiết bị 40/44 chân :

* cổng E là cổng rộng 4 bit : 3 chân RE0, RE1, RE2 được thiết lập riêng rẻ là ngõ vào hay ngõ ra. Các chân này có bộ đệm ngõ vào Schmitt trigger. Khi được chọn là ngõ vào tương tự, các chân sẽ đọc ra ‘0’.
* Các thanh ghi TRISE, LATE, PORTE giống với cổng A.
* Cùng với dữ liệu cổng, thanh ghi PORTE cũng chứa bit điều khiển RDPU (PORTE<7>) cho phép hay tắt điện trở kéo lên yếu của cổng E
* Khi reset bật nguồn, chân RE2:RE0 được thiết đặt là ngõ vào tương tự.
* Chân thứ 4 của cổng E (MCLR/VPP/RE3) chỉ là ngõ vào. Nó không có bit TRIS hay LAT tương ứng. khi bit cấu hình MCLRE = 0 nó là ngõ vào số, nó là ngõ vào Master Clear của thiết bị.
* Trong cả 2 cấu hình, RE3 cũng đồng thời làm chức năng là ngõ vào điện áp lập trình trong khi lập trình.
* Khi reset bật nguồn, RE3 là ngõ vào số chỉ khi chức năng Master Clear bị tắt.

1. Với thiết bị 28 chân :

Cổng E chỉ có thể dùng khi chức năng Master Clear bị tắt (MCLRE = 0). Trong trường hợp đó, PORTE là bit đơn, một cổng chỉ có ngõ vào là RE3 mà thôi. Chân hoạt động như đã nói trước đó.



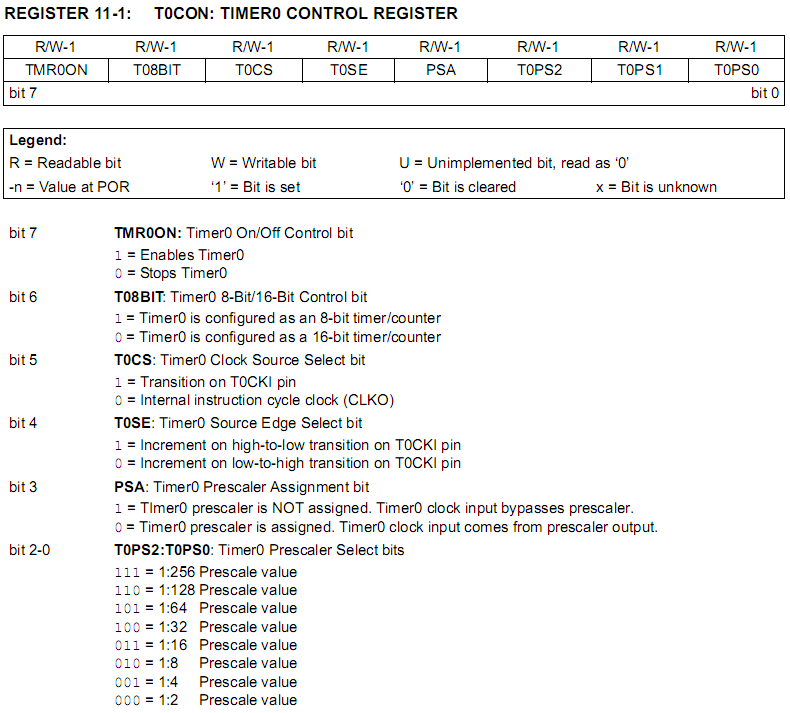


1. Timer0

Timer0 kết hợp chặt chẽ các tính năng sau:

* Vận hành lựa chọn bằng phần mềm như là bộ định thời hay bộ đếm cả chế độ 8 bit và 16 bit
* Các thanh ghi đọc/ghi được
* Bộ tiền tỉ (prescaler) lệ riêng 8 bit lập trình được bằng phần mềm.
* Nguồn xung chọn lựa được (trong hay ngoài)
* Chọn cạnh tác động cho xung ngoài
* Ngắt khi tràn.

Thanh ghi T0CON điều khiển mọi khía cạnh hoạt động của timer0, gồm cả chọn tỉ lệ và có thể đọc ghi được.



1. Hoạt động

Timer0 có thể vận hành như bộ định thời hay bộ đếm, chọn lựa bằng bit T0CS = 0(T0CON<5>)

Trong chế độ timer, việc tăng xảy ra mỗi xung là mặc định trừ khi một giá trị khác của bộ tiền tỉ lệ được chọn. Nếu ghi vào thanh ghi TMR0 thì việc tăng bị cấm trong 2 chu kỳ tiếp theo. Người dùng có thể bù lại sai lệch này bằng cách hiệu chỉnh giá trị nạp vào.

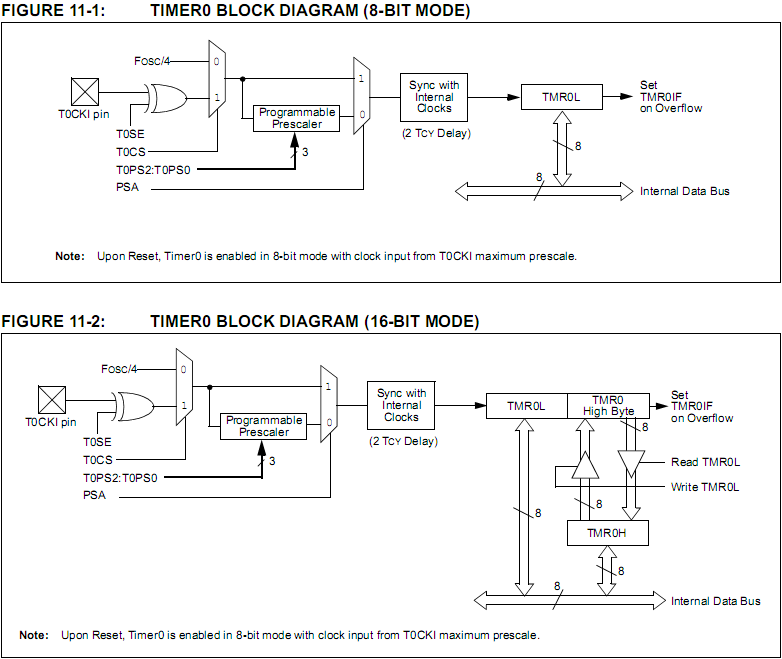
Trong chế độ counter (T0CS = 1), việc tăng khi có cạnh lên hay cạnh xuống tuy theo bit T0SE (T0CON<4>) của xung ở chân RA4/T0CKI. Khi dùng nguồn xung ngoài phải bảo đảm đạt các yêu cầu nhất định để có thể đồng bộ với pha xung nội (Tosc). Có sự trì hoãn giữa đồng bộ và bắt đầu đếm tăng lên của timer/counter.

1. Đọc và ghi trong chế độ 16 bit

TMR0H không phải là byte cao thật của timer0 trong chế độ 16 bit mà là phiên bản đệm của byte cao thật sự.

TMR0H được cập nhật cùng với nội dung byte cao của timer0 trong quá trình đọc TMR0L. điều nay cho phép không phải kiểm tra thứ tự đọc byte cao và thấp là đúng.

Tương tự, việc ghi byte cao của Timer0 phải diễn ra thông qua thanh ghi đệm TMR0H. Byte cao được cập nhật với nội dung của TMR0H khi việc ghi vào TMR0L diễn ra. Điều này cho phép Timer0 cập nhật ngay lập tức.



1. Bộ tiền tỉ lệ (Prescaler)

Bộ tiền tỉ lệ là một counter 8 bit nhưng không ghi đọc trực tiếp được. Giá trị của nó được set bằng bit PSA và các bit T0PS2:T0PS0 (T0CON<3:0>) xác định nhiệm vụ của bộ tiền tỉ lệ và tỉ lệ chia trước.

Khi bit PSA = 0, timer0 dùng bộ tiền tỉ lệ, giá trị từ 1:2 tới 1:256, chọn lựa tăng theo mũ 2.

Khi dùng bộ tiền tỉ lệ, mọi lệnh ghi vào thanh ghi TMR0 sẽ cũng xóa số đếm trong bộ tiền tỉ lệ nhưng không thay đổi thiết đặt tỉ lệ.

Định tỉ lệ của bộ tiền tỉ lệ hoàn toàn điều khiển bằng phần mềm và có thể thay đổi trong khi đang thực thi chương trình

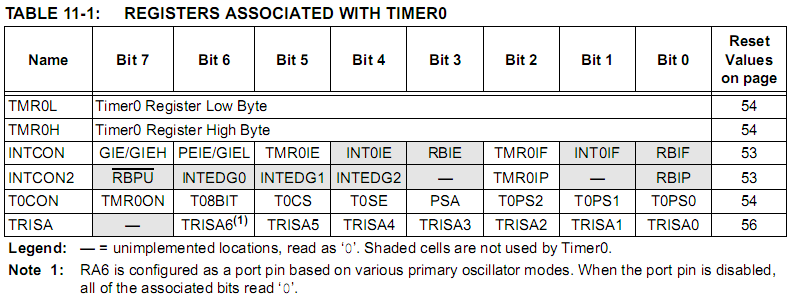
1. Ngắt timer0

Ngắt xảy ra khi tràn thanh ghi TMR0 (FFh sang 00h : 8 bit , FFFFh sang 0000h : 16 bit)

Ngắt sẽ gây set bit cờ TMR0IF. Có thể che ngắt bằng cách xáo bit TMR0IE (INTCON<5>)

Trước khi cho phép ngắt tiếp, cần xóa bit cờ TMR0IF bằng phần mềm trong đoạn mã phục vụ ngắt.

Vì Timer0 bị tắt trong chế độ ngủ, ngắt của TMR0 không thể đánh thức bộ vi xử lý.



1. Timer1

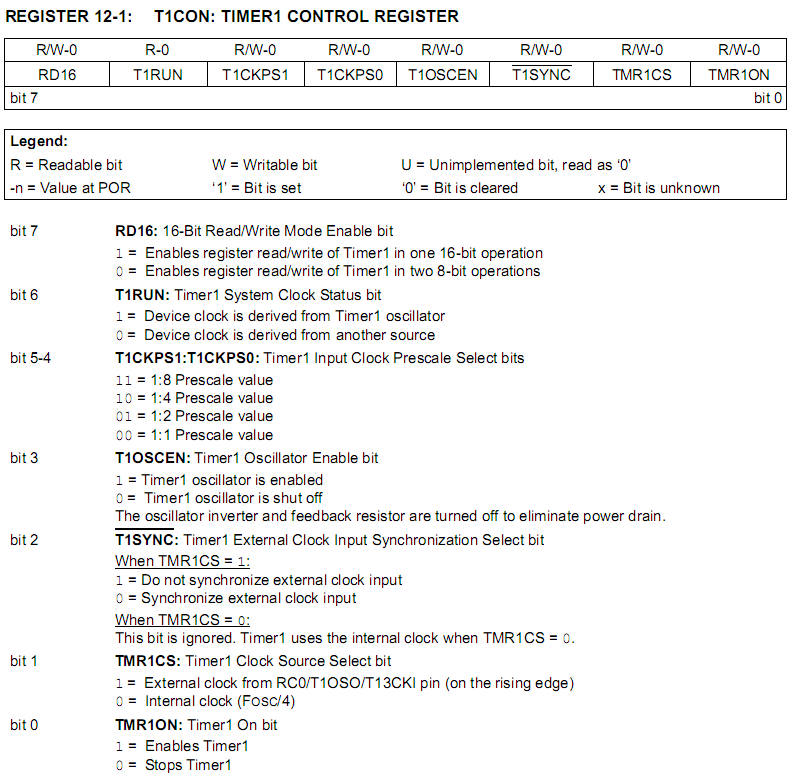
Bộ đếm/bộ định thời Timer1 kết hợp chặt chẽ các tính năng sau:

* Vận hành lựa chọn bằng phần mềm như là bộ định thời hay bộ đếm 16 bit
* Bộ tiền tỉ (prescaler) lệ riêng 8 bit lập trình được bằng phần mềm.
* Nguồn xung chọn lựa được (trong hay ngoài) với xung hệ thống hay các tùy chọn bộ dao động nội timer1
* Ngắt khi tràn.
* Bộ reset trên tín hiệu kích khởi sự kiện đặc biệt CCP (CCP special Even trigger)
* Cờ trạng thái đồng hồ thiết bị(T1RUN)

Bộ timer1 kêt hợp chặt chẽ với bộ dao động công suất thấp của nó để cung cấp tùy chọn xung thêm vào. Bộ dao động timer1 cũng có thể dùng làm nguồn xung công suất thấp cho bộ vi điều khiển trong vận hành bị quản lý năng lượng.

Timer1 cũng có thể dùng để cung cấp chức năng Real-timer clock cho các ứng dụng với chỉ một sư thêm vào tối thiểu các thành phần ngoài và mã trên đầu chương trình.

Timer1 được điều khiển thông qua thanh ghi T1CON. Có thể bật tắt timer1 và bộ dao động của nó nhờ thanh ghi T1CON



1. Vận hành timer1

Timer 1 có thể vận hành trong các mode sau :

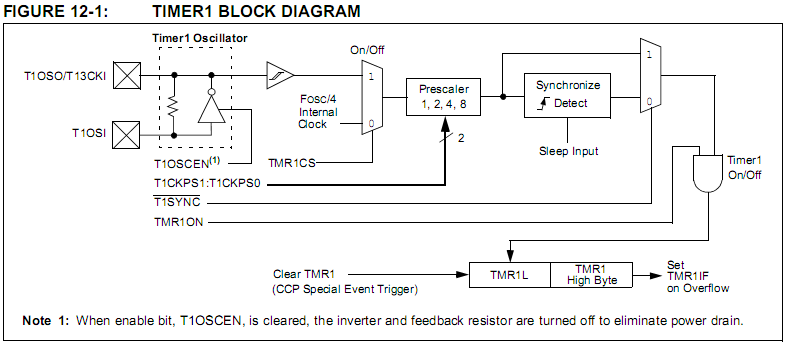
* Timer
* Counter đồng độ
* Counter bất đồng bộ.

Chế độ vận hành được quyết định bởi bit lựa chọn xung TMR1CS (T1CON<1>)

TMR1CS = 0, Timer1 tăng mỗi chu kỳ lệnh nội (Fosc/4)

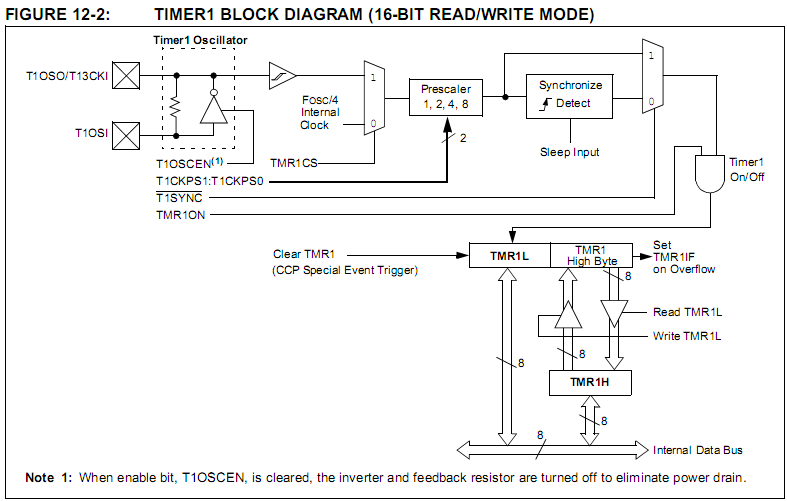
TMR1CS = 1, Timer1 tăng ở cạnh lên của xung đưa vào ngõ vào xung ngoài của timer1 hay bộ dao động của timer1(nếu được bật).

Khi timer1 được bật, chân RC1 và RC0 là ngõ vào. Điều này nghĩa là các giá trị của TRISC<1:0> bị bỏ qua và các chân đọc ra ‘0’



1. Chế độ đọc/ghi 16 bit timer1

Giống timer0

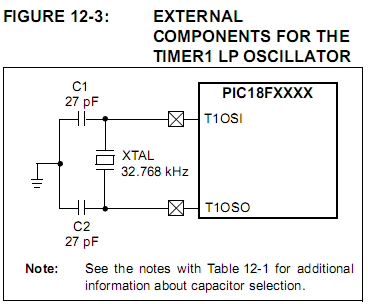


1. Bộ dao động timer1

Một mạch dao động tinh thể trên chip được kết hợp chặt chẽ giữa chân T1OSI (ngõ vào) và T1OSO (ngõ ra bộ khuếch đại). Bộ dao động được thiết kế cho thạch anh 32kHz. Nó tiếp tục chạy trong mọi chế độ quản lý năng lượng. Mạch cho bộ dao động LP điển hình

Kích hoạt bộ dao động bằng bit T1OSCEN (T1CON<3>)

Người dùng phải cung cấp một khoảng trễ trong phần mềm để bảo đảm bộ dao động timer1 khởi động hoàn tất.

* 1. Dùng timer1 là một nguồn xung

Bộ dao động timer1 cũng có thể là một nguồn xung trong chế độ quản lý năng lượng.

Bằng cách thiết lập OSCCON<1:0> = ‘01’, thiết bị chuyển sang chế độ SEC\_RUN – cả CPU và thiết bị ngoại vi được cấp xung từ bộ dao động timer1. Nếu IDLEN = 0 và lệnh Sleep được thực thi, thiết bị sẽ tiến vào chế độ SEC\_IDLE.

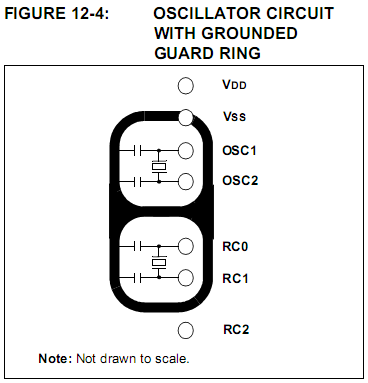
Khi bộ dao động timer1 đang cung cấp xung cho hệ thống, bit cờ trạng thái xung hệ thống timer1 T1RUN (T1CON<6>) được set. Từ đó xác định được nguồn xung hiện tại cho bộ điều khiển. Nó cũng chỉ ra nguồn xung đang dùng hiện thời cho bộ giám sát xung an toàn tránh hỏng (Fail Safe clock monitor) . Nếu bộ giám sát xung bật và bộ dao động timer1 hỏng trong khi đang cấp xung, việc thăm dò bit T1RUN sẽ chỉ ra xung đang được cung cấp bởi bộ dao động timer1 hay nguồn khác.

* 1. Tùy chọn timer1 công suất thấp

Bộ dao động timer1 có thể vận hành ở 2 mức năng lượng khác nhau tùy vào thiết đặt thiết bị.

Khi bit thiết đặt LPT1OSC = 1, bộ dao động Timer1 vận hành trong chế độ năng lượng thấp, khi = 0 thì mức năng lượng cao hơn.

Năng lượng tiêu thụ cho môt chế độ cụ thể khá là cố định, bất chất chế độ vận hành của thiêt bị. Thiết đặt timer1 mặc định là chế độ năng lượng cao hơn.

Vì chế độ công suất thấp timer1 có xu hướng nhạy với nhiễu cảm ứng, môi trường nhiễu cao có thể gây vài sự không ổn định cho bộ dao động. Tùy chọn năng lượng thấp do đó phù hợp nhất cho ứng dụng nhiễu thấp mà ở đó tiết kiệm năng lượng là cân nhắc thiết kế quan trọng.

* 1. Cân nhắc bộ dao động timer1

Mach bộ dao động timer1 lấy rất ít năng lượng trong khi vận hành. Vì bản chất tiêu công suất thấp của bộ dao động, nó có thể cũng nhạy với tín hiệu thay đổi nhanh ở gần.

Mach dao động nên gần vi điều khiển nhất có thể và không mạch nào trong phạm vi đường bao quanh ngoài VSS hay VDD

Nếu mạch tốc độ cao phải đặt gần bộ dao động (như chân CCP1 trong ngõ ra bộ so sánh hay chế độ PWM, hay bộ dao động chính dùng chân OSC2), một vòng nối đất bảo vệ xung quanh mạch dao động như trong hình 12-4 có thể giúp ích khi dùng trên bo một mặt hay thêm vào một vùng nối đất.

1. Ngắt timer1

Khi cặp thanh ghi TMR1H:TMR1L từ FFFFh trở về 0000h sẽ xảy ra ngắt.

Bit cờ ngắt TMR1IF

Bit cho bật tắt ngắt TMR1IE.

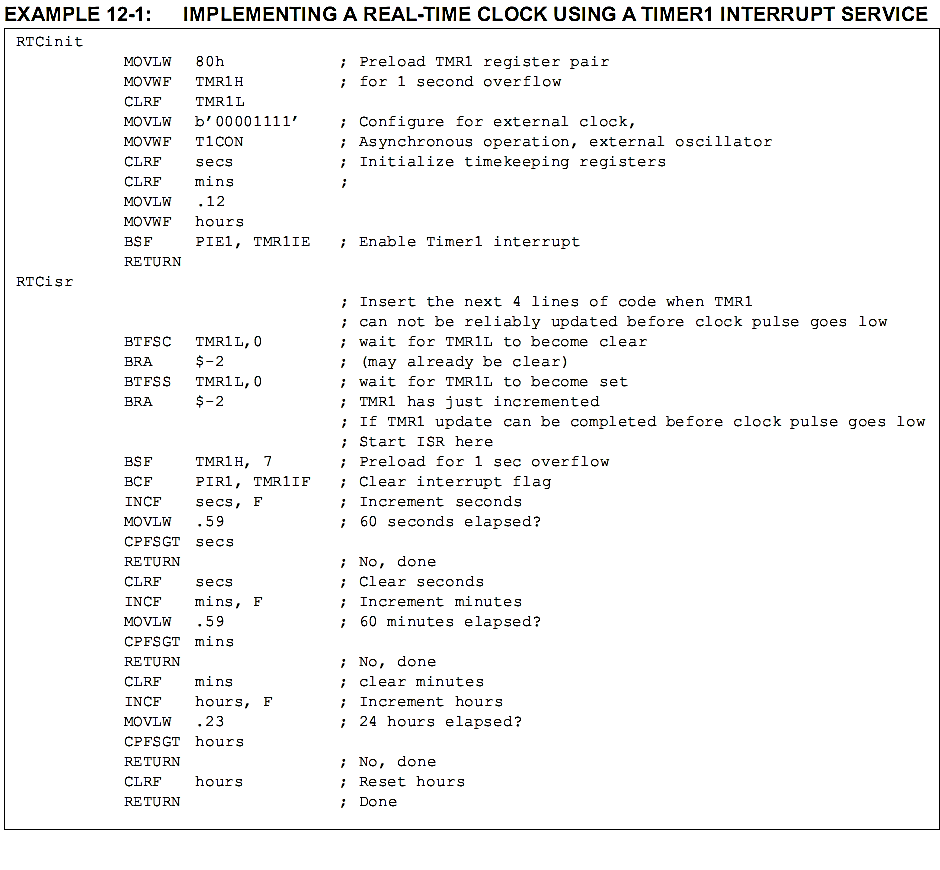
1. Reset timer1 dùng tín hiệu kích khởi sự kiện đặc biệt CCP

* Nếu một trong 2 bộ CCP được thiết đặt trong chế độ so sánh tạo ra tín hiệu kích khởi sự kiện đặc biệt(CCP1M3:CCP1M0 hay CCP2M3:CCP2M0 = 1011), tín hiệu này sẽ reset timer1.
* Bộ timer1 phải được cấu hình là timer hay counter đồng bộ để tận dụng tính năng này. Khi dùng theo cách này, cặp thanh ghi CCPRH:CCPRL trở thành thanh ghi thời khoảng cho timer1.
* Nếu timer1 trong chế độ counter bất đồng bộ, quá trình reset này có thể không làm việc.
* Trong sự kiện ghi vào timer1 trùng với tín hiệu kích khởi sự kiện đặc biệt, quá trình ghi được ưu tiên.
* Sự kiện kích khởi đặc biệt từ CCP2 sẽ không set bit cờ ngắt của TMR1IF

1. Dùng timer1 làm đồng hồ thời gian thực

Thêm vào bộ dao động năng lượng thấp ngoài cho timer1 cho người dùng tùy chọn để đính kèm chức năng đồng hồ thời gian thực RTC cho ứng dụng người dùng. Điều này đạt được bằng cách dùng một tinh thể cung cấp thời gian chuẩn chính xác và nhiều dòng lệnh để tính toán thời gian.

Khi vận hành trong chế độ Sleep và dùng pin hay siêu tụ điện làm nguồn năng lượng, nó có thể loại bỏ hoàn toàn như cầu có thiết bị RTC riêng và pin dự phòng.

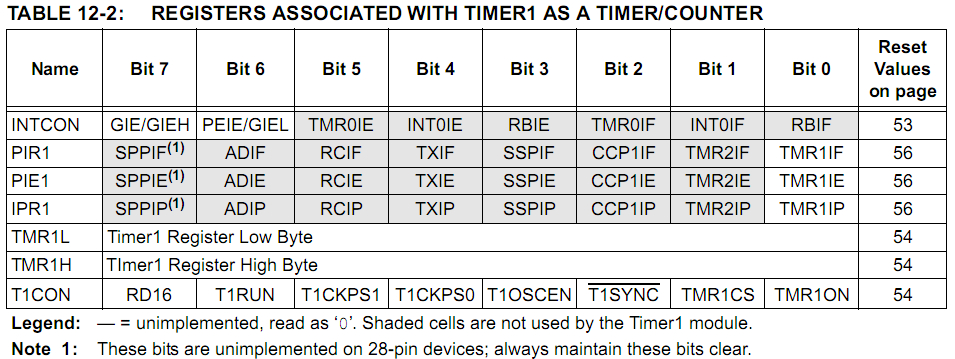


1. Điều lưu tâm về chế độ bộ đếm bất đồng bộ (considerations in Asynchronous counter mode)

Theo sau một ngắt timer1 và việc cập nhật thanh ghi TMR1, bộ Timer1dùng cạnh xuống của nguồn xung của nó để kích khởi việc cập nhật thanh ghi kế tiếp ở cạnh lên. Nếu cập nhật hoàn tất sau khi xung ngõ vào đã xuống, cảnh lên kế tiếp sẽ không được đếm.

Nếu ứng dụng có thể cập nhật tin cậy TMR1 trước khi ngõ vào timer1 xuống mức thấp, không phải làm gì thêm. Ngược lại, một cập nhật điều chỉnh có thể được thực hiện sau khi timer1 tăng. Điều này có thể được làm bằng cách giám sát TMR1L trong đoạn mã ngắt cho tới khi nó tăng, và sau đó cập nhật cặp thanh ghi TMR1H:TMR1L trong khi xung là thấp hay một nữa thời gian của nguồn xung. Giả sử là timer1 đang được dùng làm RTC, nguồn xung là bộ dao động tinh thể 32.768 kHz; trong trường hợp này, một nữa thời khoảng của xung là 15.25us

Mã ứng dụng RTC trong ví dụ 12-1 thể hiện một đoạn chương trình ngắt điển hình cho timer1, cũng như là mã tùy chọn được yêu cầu nếu việc cập nhất không được thực hiện tin cậy với khoảng giữa được yêu cầu.



1. Timer2

Timer2 bộ định thời kết hợp chặt chẽ với các tính năng sau :

* Thanh ghi timer 8 bit TMR2 và thanh ghi thời khoảng PR2
* Cả 2 thanh ghi có thể đọc ghi
* Bộ tiền tỉ lệ lập trình được bằng phần mềm (1:1, 1:4 và 1:16)
* Bộ hậu tỉ lệ lập trình được bằng phần mềm (1:1 tới 1:16)
* Ngắt khi TMR2 bằng PR2
* Dùng tùy chọn là xung dịch cho bộ MSSP

Bộ timer2 được điều khiển th6ng qua thanh ghi T2CON : tắt mở timer2, bộ tiền tỉ lệ, bộ hậu tỉ lệ.

1. Hoạt động timer2

Trong hoạt động bình thường TMR2 được tăng từ 0 khi có xung Fosc/4.

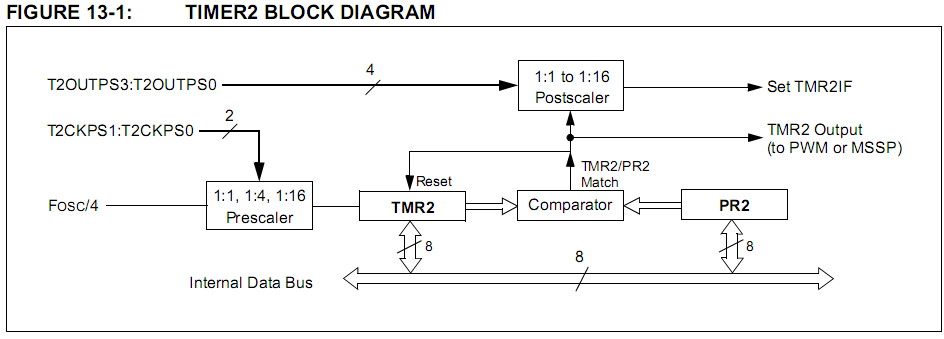
Một bộ đếm hay bộ tiền tỉ lệ 2 bit ở ngõ vào xung cho các tùy chọn tiền tỉ lệ của xung là chia 1, chia 4 và chia 16. Được điều khiển bởi bit T2CKPS1:T2CKPS0 (T2CON<1:0>).

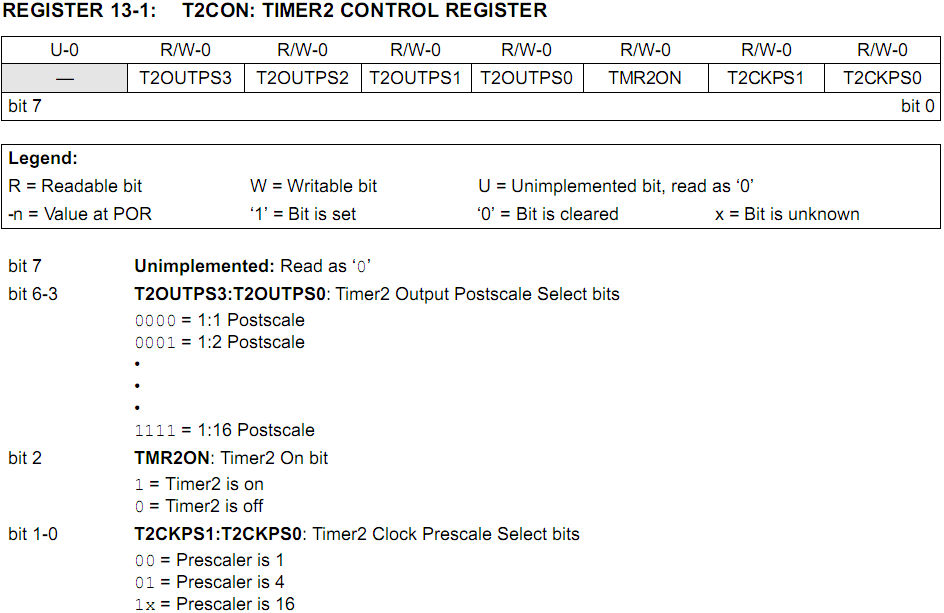
Giá trị của TMR2 được so sánh với PR2 ở mỗi chu kỳ xung, khi 2 giá trị giống nhau, bộ so sánh tạo ra tín hiệu giống nhau tới ngõ ra timer2. Tín hiệu này cũng reset giá trị TMR2 về 0h ở chu kỷ kế và lái bộ đếm hay bộ hậu tỉ lệ ngõ ra.

Các thanh ghi TMR2 và PR2 đọc ghi trực tiếp được. Thanh ghi TMR2 được xóa khi có bất kỳ ngắt nào của thiết bị, trong khi PR2 khởi tạo tại FFh. Cả bộ đếm tiền tỉ lệ và hậu tỉ lệ được xóa khi có các sự kiện sau :

* Ghi vào TMR2
* Ghi vào T2CON
* Bất kỳ reset nào (POR, MCLR reset, BOR hay WDT reset)

TMR2 không bị xóa khi T2CON bị ghi.





1. Ngắt timer2

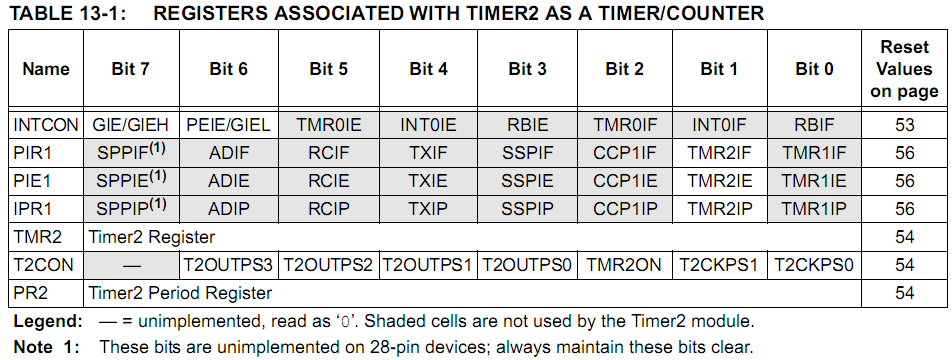
Timer2 cũng tạo ra một ngắt thiết bị tùy chọn. Tín hiệu ngõ ra timer2 (khi TMR2 bằng PR2) cung cấp ngõ vào cho bộ đếm hay bộ tiền tỉ lệ 4 bit. Bộ đếm này tạo ra cờ ngắt trùng TMR2 được chốt trong TMR2IF(PIR<1>)

Ngắt được cho phép bằng bit TMR2IE (PIE1<1>)

Một khoảng 16 tùy chọn hậu tỉ lệ (từ 1:1 tới 1:16) có thể được chọn với bit điều khiển bộ hậu tỉ lệ T2OUTPS3:T2OUTPS0 (T2CON<6:3>).

1. Ngõ ra timer2

Ngõ ra không chia tỉ lệ của TMR2 chủ yếu để đưa tới bộ CCP, nơi nó được dùng làm thời gian nền cho hoạt động chế độ PWM. Timer2 có thể được dùng tùy chọn làm nguồn xung dịch cho bộ MSSP trong chế độ SPI



1. Timer3

Timer3 bộ đếm/bộ định thời kết hợp chặt chẽ các tính năng sau :

* Hoat động như là bộ đếm hay bộ định thời 16 bit.
* Các thanh ghi 8 bit đọc ghi được TMR3H:TMR3L
* Nguồn xung chọn được (trong hay ngoài) vớii xung thiết bị hay tùy chọn bộ dao động timer1
* Ngắt khi tràn
* Bộ timer3 sẽ reset dựa trên tín hiệu kích khởi sự kiện đặc biệt CCP.

Timer3 được điều khiển thông qua thanh ghi T3CON. Nó cũng là tùy chọn nguồn xung cho bộ CCP.



1. Hoạt động bộ Timer3

Timer 3 có thể vận hành trong các mode sau :

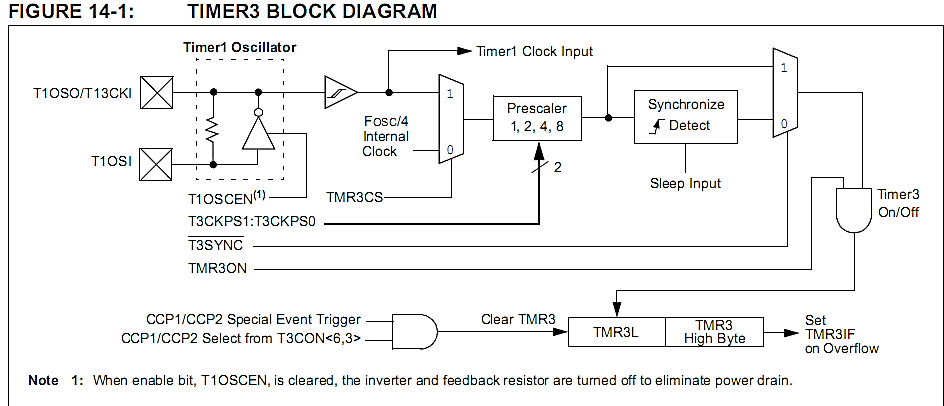
* Timer
* Counter đồng độ
* Counter bất đồng bộ.

Chế độ vận hành được quyết định bởi bit lựa chọn xung TMR3CS (T3CON<1>)

TMR3CS = 0, Timer3 tăng mỗi chu kỳ lệnh nội (Fosc/4)

TMR3CS = 1, Timer3 tăng ở cạnh lên của xung đưa vào ngõ vào xung ngoài của timer1 hay bộ dao động của timer1(nếu được bật).

Khi bộ dao động timer1 được bật, chân RC1 và RC0 là ngõ vào. Điều này nghĩa là các giá trị của TRISC<1:0> bị bỏ qua và các chân đọc ra ‘0’



1. Chế độ đọc ghi 16 bit của timer3

* Timer3 có thể được thiết đặt đọc ghi 16 bit.
* Khi bit RD16 = 1, địa chi TMR3H được sơ đồ hóa thành thanh ghi đệm cho byte cao của timer3
* Việc đọc TMR3L sẽ nạp nội dụng byte cao timer3 vào bộ đệm TMR3H.
* Việc ghi vào byte cao của Timer3 phải diễn ra thông qua thanh ghi đệm TMR3H. Nội dung TMR3H tự động được ghi cập nhật vào byte cao của timer3 khi việc ghi vào TMR3L diễn ra.
* Việc ghi vào TMR3H không xóa bộ tiền tỉ lệ timer3, nó chỉ bị xóa khi ghi vào TMR3L.

1. Ngắt timer1

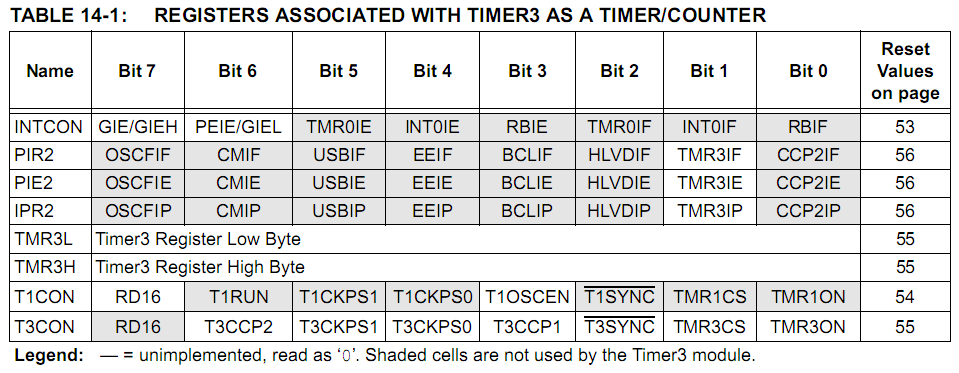
Khi cặp thanh ghi TMR3H:TMR3L từ FFFFh trở về 0000h sẽ xảy ra ngắt.

Bit cờ ngắt TMR3IF (PIR<1>)

Bit cho bật tắt ngắt TMR3IE (PIE<1>)

1. Reset timer3 dùng tín hiệu kích khởi sự kiện đặc biệt CCP

* Nếu bộ CCP2 được cấu hình để tạo ra một tín hiệu kích khởi sự kiện đặc biệt trong chế độ so sánh (CCP2M3:CCP2M0 = 1011), tín hiệu này sẽ reset timer3. Nó cũng sẽ bắt đầu quá trình chuyển đổi AD nếu bộ AD được kích hoạt.
* Bộ timer3 phải được cấu hình là timer hay counter đồng bộ để tận dụng tính năng này. Khi dùng theo cách này, cặp thanh ghi CCPR2H:CCPR2L trở thành thanh ghi thời khoảng cho timer3.
* Nếu timer3 trong chế độ counter bất đồng bộ, quá trình reset này có thể không làm việc.
* Khi việc ghi vào timer3 trùng với tín hiệu kích khởi sự kiện đặc biệt, quá trình ghi được ưu tiên.
* Sự kiện kích khởi đặc biệt từ CCP2 sẽ không set bit cờ ngắt của TMR3IF (PIR2<1>).



1. Bộ Capture/Compare/PWM (CCP)

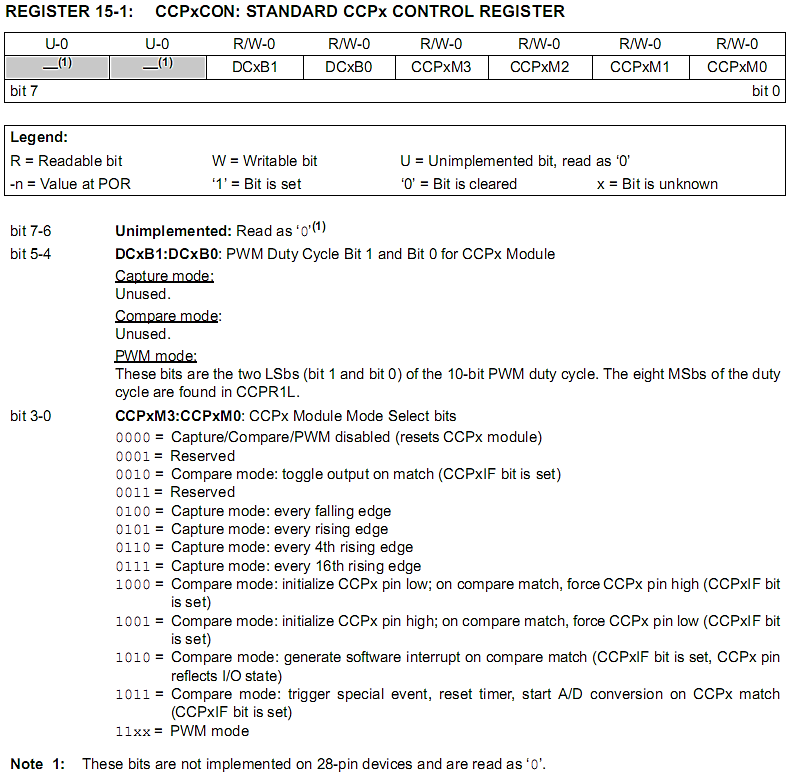
PIC18F2445/2550/4455/4550 đều có 2 bộ CCP (Capture/compare/PWM).

MỖi bộ có 1 thanh ghi 16 bit có thể hoạt động như là thanh ghi lưu giữ 16 bit (capture register), thanh ghi so sánh 16 bit (compare register), thanh ghi chu kỳ hoạt động chính/phụ độ rộng xung (PWM master/slave duty cycle register).

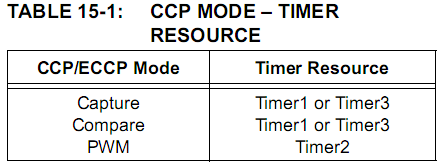
Trong thiết bị 28 chân (dòng 2445 và 2550) bộ CCP 1 và CCP2 vận hành như trong chương này.

Trong thiết bị 40/44 chân (dòng 4455 và 4550), CCP1 được chế tạo là bộ CCP cài tiến với chế độ capture, compare tiêu chuẩn vả chế độ PWM cải tiến.

Do vậy sự vận hành mô tả trong chương này đúng cho các chế độ compare và capture của bộ CCP tiêu chuẩn và cải tiến.



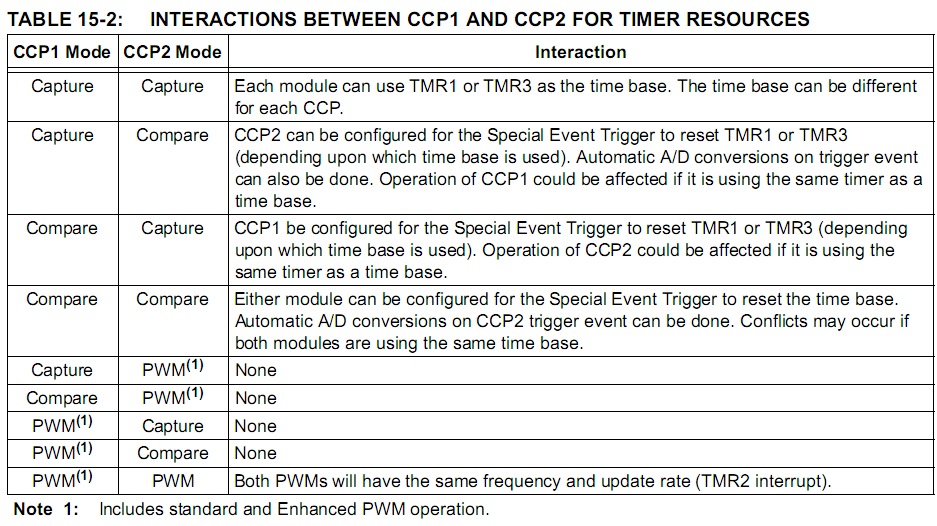
1. Cấu hình bộ CCP

Mỗi bộ CCP được liên kết với một thanh ghi điều khiển thường là CCPxCON 8 bit và một thanh ghi dữ liệu CCPRx 16 bit CCPRxH:CCPRxL . Cả 2 thanh ghi đều đọc ghi được.

1. Tài nguyên timer và bộ CCP

Bộ CCP dùng timer 1,2 hay 3 tùy vào chế độ lựa chọn.

Sư phân định này dựa vào các bit trong thanh ghi T3CON. Cả 2 bộ CCP có thể được kích hoạt bất kỳ lúc nào và có thể chia sẽ cùng tài nguyên timer nếu chúng được thiết đặt để vận hành trong cùng chế độ cùng lúc. Tương tác của chúng tóm tắt trong bảng sau



1. Quy ươc chân CCP2

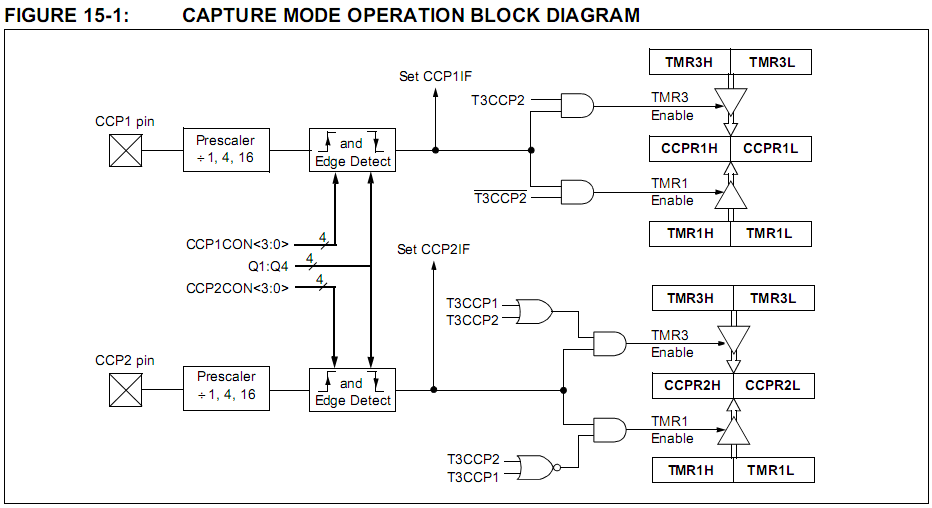
Quy ước chân cho CCP2 (ngõ vào capture, ngõ ra của PWM và compare) có thể thay đổi dựa trên cấu hình thiết bị.

Bit cấu hình CCP2MX quyết định chân nào của CCP2 được ghép kênh tới.

Mặc định CCP2MX = 1, CCP2 ghép kênh với RC1

Nếu CCP2MX = 0 CCP2 ghép kênh với với RB3

Thay đổi quy ước chân của CCP2 không tự động thay đổi yêu cầu nào cho việc cấu hình cân công. Người dùng phải luôn kiểm tra thanh ghi TRIS thích hợp được cấu hình đúng cho việc vận hành CCP2 bất chấp nó ở chân nào



1. Chế độ capture

Trong chế độ capture, cặp thanh ghi CCPRxH:CCPRxL giữ giá trị 16 bit của thanh ghi TMR1 hay TMR3 khi một sự kiện diễn ra ở chân CCPx tương ứng. Một sự kiện được định nghĩa là một trong số như sau:

* Mỗi cạnh xuống
* Mỗi cạnh lên
* Mỗi 4 cạnh lên
* Mỗi 16 cạnh lên

Sự kiện được chọn bằng bit chọn chế độ CCPxM3:CCPxM0 (CCPxCONN<3:0>). Khi một việc bắt giữ được thực hiện, bit cờ yêu cầu ngắt CCPxIF được set, nó phải được xóa bằng phần mềm. Nếu một việc bắt giữ khác diễn ra trước khi giá trị thanh ghi CCPRx được đọc thi giá trị bắt giữ trước đó sẽ bị ghi đè bởi giá trị mới bắt giữ.

1. Thiết đặt chân CCP

Trong chế độ capture, chân CCPx thích hợp phải được cấu hình là ngõ vào thông qua bit tương ứng của thanh ghi TRIS.

Lưu ý nếu chân RB3/CCP2 hay RC1/CCP2 được cấu hình là ngõ ra, việc ghi vào cổng có thể gây ra điều kiện bắt giữ.

1. Lựa chọn chế độ timer1 / timer3

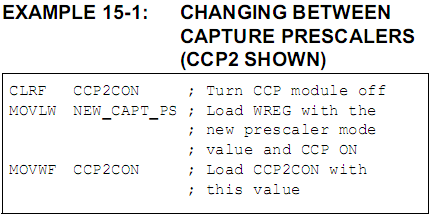
Các timer mà để dùng với tính năng capture phải được chạy trong chế độ timer hay chế độ counter đồng bộ. Trong chế dộ counter bất đồng bộ, capture sẽ không hoạt động. Timer được dùng được chọn bằng thanh ghi T3CON.

1. Ngắt phần mềm

Khi chế độ capture được thay đổi, một ngắt sai có thể được tao ra. Người dùng nên giữ cho bit cho phép ngắt cảu capture bị xóa để tránh ngắt sai. Bit cờ ngắt CCPxIF nên được xóa sau những thay đổi như vậy trong vận hành.

1. Bộ tiền tỉ lệ CCP

Có 4 thiết đặt tiền tỉ lệ trong chế độ capture. Chúng được định nghĩa là một phần của chế độ vận hành được lựa chọn thông qua bit chọn chế độ CCPxM3:CCPxM0). Khi bộ CCP bị tắt hay chế độ capture bị tắt, bộ đếm tiền tỉ lệ bị xóa. Nghĩa là bất kỳ reset nào sẽ xóa bộ đếm tiền tị lệ.



1. Chế độ compare

Trong chế độ compare, giá trị thanh ghi CCPRx được so sánh liên tục với giá trị cặp thanh ghi TMR1 hay TMR3. Khi giống nhau, chân CCPx có thể   
lái lên mức cao

Lái xuống mức thấp

Đảo (cao => thấp, thấp => cao)

Vẫn không đổi (đó là , phản ánh trạng thái của chốt I/O)

1. Thiết đặt chân cho CCP

Người dùng phải thiết đặt chân CCPx là ngõ ra bằng thanh ghi TRIS

Xóa thanh ghi CCP2CON sẽ buộc RB3 hay RC1 là ngõ ra bộ so sánh chốt xuống mức thấp mặc định. Đây không phải là chốt dữ liệu cổng c hay b

1. Lựa chọn chế độ timer1/timer3

Timer 1 hay/và timer3 phải được chạy trong chế độ timer hay chế độ counter đồng bộ để dùng tính năng so sánh. Chế độ counter bất đồng bộ, hoạt động so sánh có thể không làm việc.

1. Chế độ ngắt phần mềm

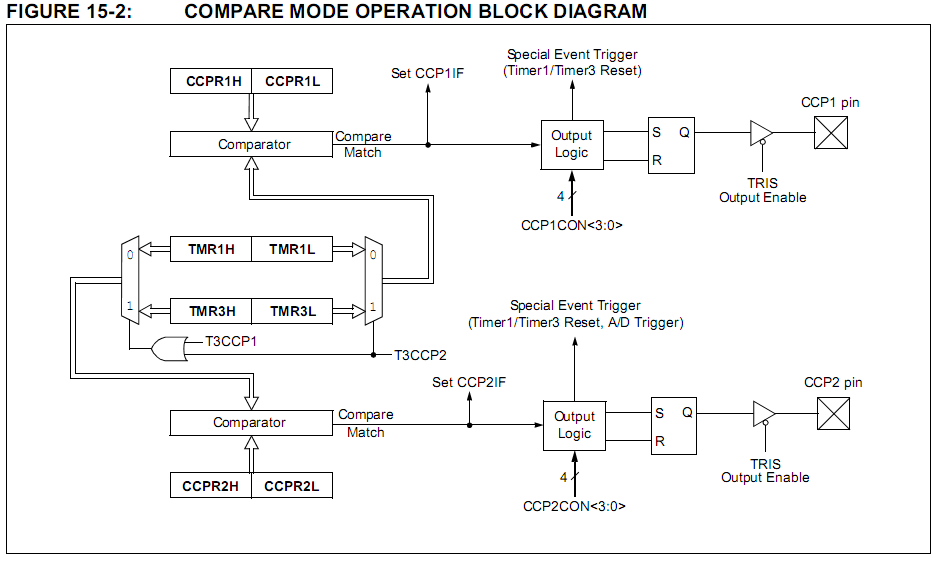
Khi chế độ ngắt phần mềm tạo ra được chọn (CCPxM3:CCPxM0 = 1010), chân CCPx tương ứng không bị ảnh hưởng. Chỉ một ngắt CCP được tạo ra nếu được chon phép sẽ set bit CCPxIE.

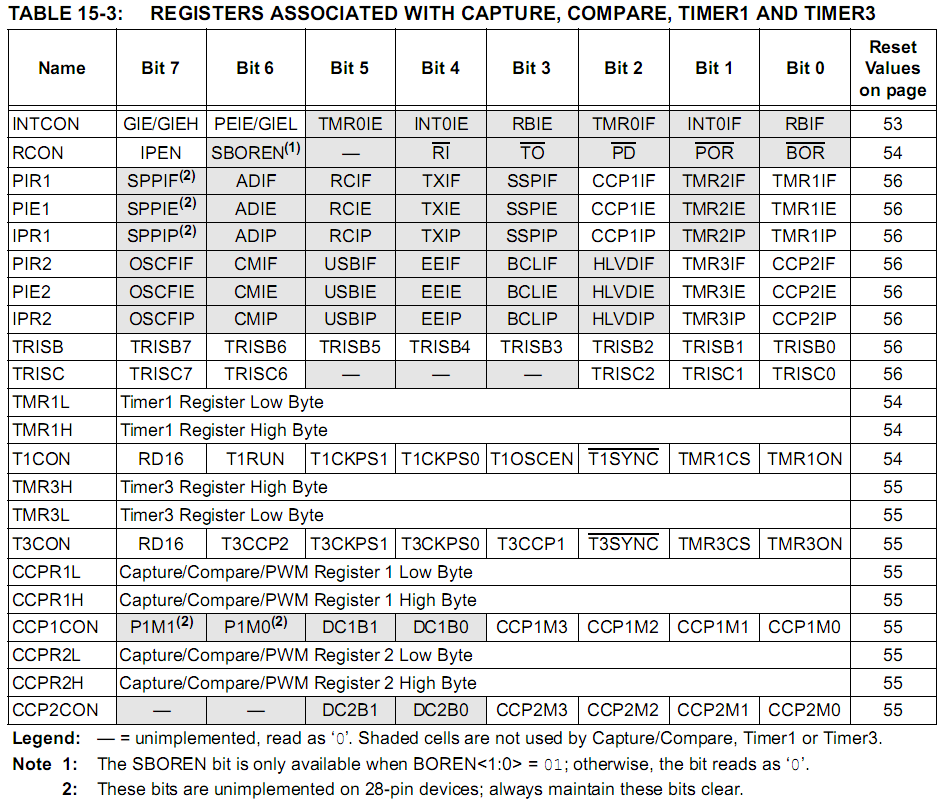
1. Tín hiệu kích khởi sự kiện đặc biệt

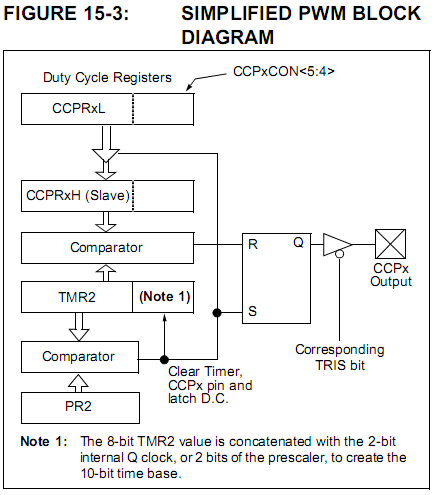
Cả hai bộ CCP được trang bị tín hiệu kích khởi sự kiện đặc biệt. Đây là một tín hiệu phần cứng bên trong được tạo ra trong chế độ so sánh để kích khởi hành động bỏi bộ khác. Tín hiệu kích khởi sự kiện đặc biệt được cho phép bằng cách chọn chế độ so sánh tín hiệu kích khởi sự kiện đặc biệt (compare special event trigger mode) CCPxM3:CCPxM0 = 1011

Với cả 2 bộ CCP, tín hiệu kích khởi sự kiện đặc biệt reset cặp thanh ghi timer cho bất kỳ tài nguyên timer nào đang được phân làm bộ cấp thời gian chuẩn. Việc này giúp cho phép các thanh ghi CCPRx dùng như là thanh ghi thời khoảng cho cả 2 timer.

Tín hiệu kích khởi sự kiện đặc biệt cho CCP2 cũng có thể bắt đầu sự chuyển đổi AD. Để như vậy bộ A/D phải được kích hoạt sẵn.



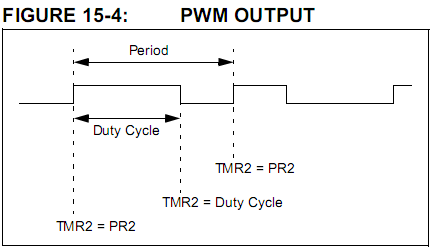


1. Chế độ PWM

Chế độ điều chế độ rộng xung, chân CCPx tạo ra ngõ ra PWM độ phân giải tới 10 bit. Vì chân CCP2 được ghép kênh với cổng B hay cổng C chốt dữ liệu, bit TRIS thích hợp phải được xóa để làm chân CCP2 là ngõ ra.

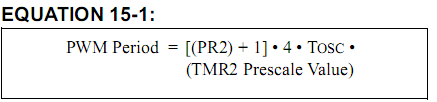
Xóa thanh ghi CCP2CON sẽ buộc chân ngõ ra RB3 hay RC1 chốt tới mức thấp mặc định. Đây không phải là chốt dữ liệu cảu cổng c hay b

Ngõ ra PWM có thời gian chuẩn (chu kỳ period) và một thời gian ngõ ra ở mức cao (chu kỳ làm việc – duty cycle). Tần số của PWM là đảo của thời khoảng (1/period)



1. Chu kỳ PWM (PWM period)

Chu kỳ PWM được chỉ định bằng cách ghi vào thanh ghi PR2. Chu kỳ PWM có thể được tính toán dùng công thức dưới đây :



Tần số PWM được định nghĩa là 1/[ chu kỳ PWM ]

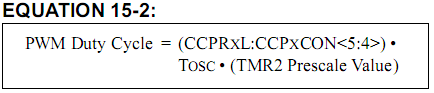
Kh TMR2 tương đương với PR2, 3 sự kiện sau sẽ diễn ra ở chu kỳ tăng lên kế tiếp :

* TMR2 bị xóa
* Chân CCPx được set (trừ khi chu kỳ làm việc PWM = 0 %, chân CCPx sẽ không được set)
* Chu kỳ công tác PWM được chốt từ CCPRxL vào CCPRxH

Bộ hậu tỉ lệ timer2 không được dùng trong quyết định tần số PWM. Bộ hậu tỉ lệ có thể được dùng để có tỉ lệ cập nhật cơ cấu tự động ở tần số khác ngõ ra PWM

1. Chu kỳ làm việc PWM

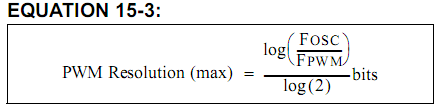
Chu kỳ làm việc PWM được chỉ định bằng cách ghi vào thanh ghi CCPRxL (8 bit thấp) và 2 bit CCPxCON<5:4> (2 bit cao)  
Độ phân giải có thể lên tới 10 bit. Giá trị 10 bit này được đại diện bởi CCPRxL:CCPxCON<5:4>. Biểu thức sau dùng tính chu kỳ làm việc trong thời gian ;

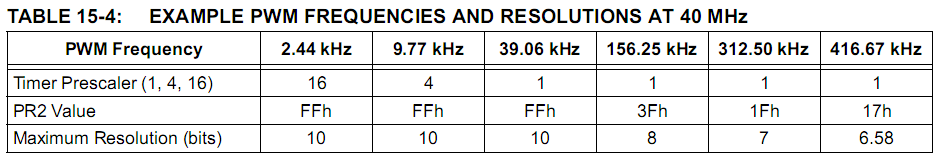


Chú ý là nếu chu kỳ làm việc dài hơn chu kỳ PWM thì chân CCPx sẽ không bị xóa

CCPRxL và CCPxCON<5:4> có thể được ghi vào bất cứ lúc nào, nhưng giá trị chu kỳ làm việc không được chốt vào thanh ghi CCPRxH cho tới hi sau khi giá trị trong PR2 và TMR2 giống nhau diễn ra. Trong chế độ PWM, thanh ghi CCPRxH là thanh ghi chỉ đọc.

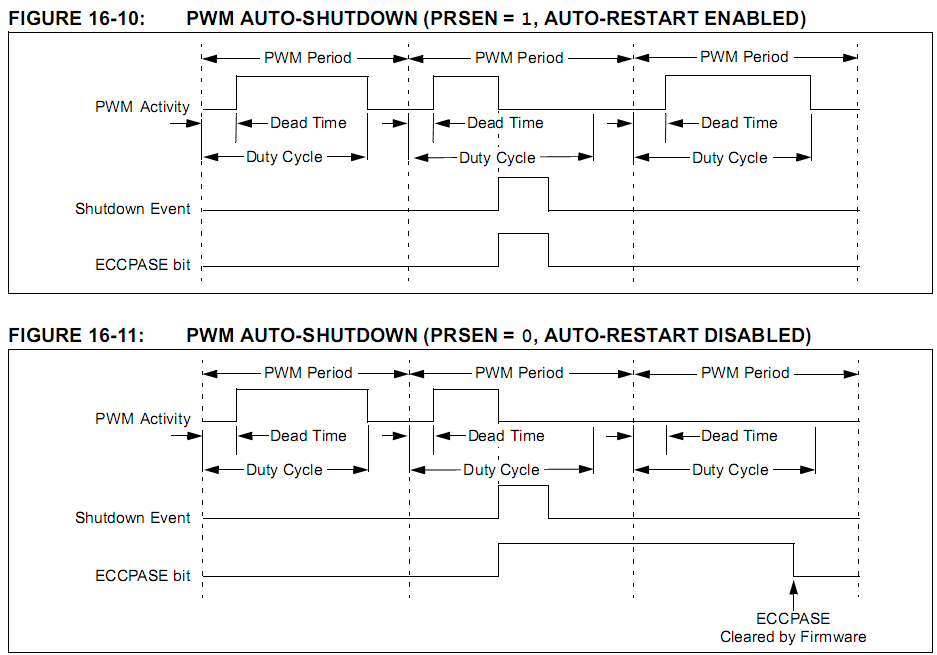
Thanh ghi CCPRxH và 2 bit chốt ngoài được dùng để đệm kép cho chu kỳ làm việc PWM. Việc đệm kép là cần thiết để hoạt động PWM ổn định.





1. Tự tắt PWM (cho bộ CCP1 thôi)

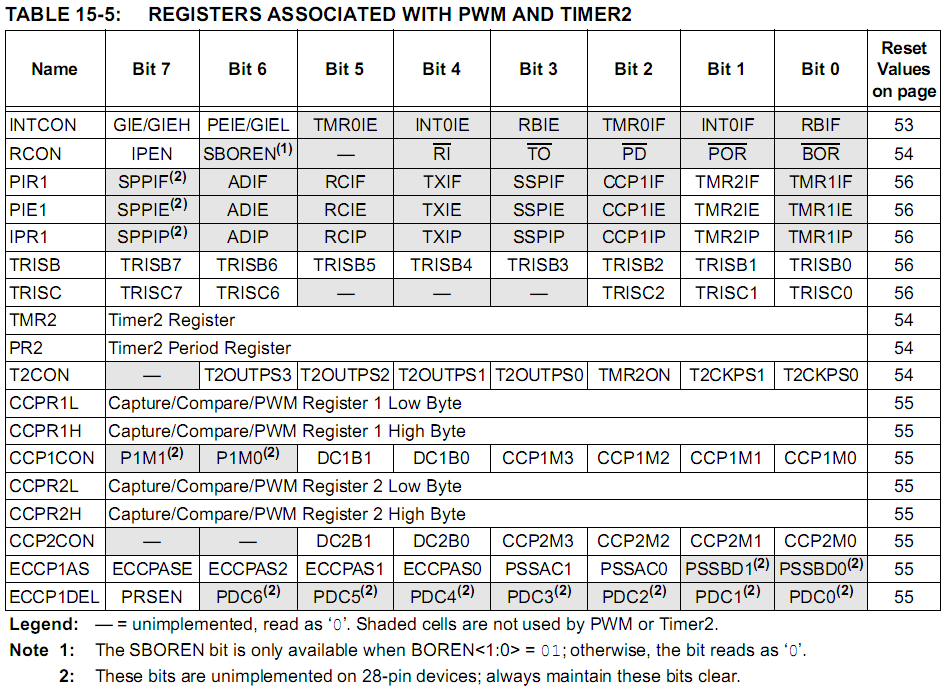
* Tính năng tự động tắt này của bộ CCP cải tiến cũng có trong bộ CCP1 của thiết bị 28 chân(phần 16.4.7.1)
* Tính năng tự tắt có thể được thiết đặt để cho phép tự động khởi động lại bộ CCP sau sự kiện tắt. Bật tính năng này bằng bit PRSEN (ECCP1DEL <7>)
* Trong chế độ shutdown với :
* Nếu PRSEN = 1 : bit ECCPASE sẽ vẫn còn set miễn là nguyên nhân gây tắt tiếp tục. Khi điều kiện shutdown bị xóa, bit ECCP1ASE được xóa.
* Nếu PRSEN = 0, khi điều kiện xóa diễn ra bit ECCPASE sẽ vần còn được set cho tới khi được xóa bằng phần sụn. Một khi ECCPASE được xóa, PWM cải tiến sẽ tiếp tục lại từ đầu cuả chu kỳ PWM kế.
* Không phụ thuộc vàO thiết đặt bit PRSEN, nếu nguồn tự động tắt là comparator, điều kiện tắt là một mức. Bit ECCPASE không thể xóa khi nguyên nhân shutdown vẫn còn.
* Chế độ tự tắt có thể bị buộc bằng cách ghi ‘1’ vào bit ECCPASE.



1. Cài đặt đê vận hành PWM

Các bước phải thiết đặt CCPx để vận hành PWM :

* Thiết đặt chu kỳ PWM bằng cách ghi thanh ghi CCPRxL 8 bit thấp và CCPxCON<5:4> 2 bit cao.
* Thiết đặt chu kỳ làm việc trong thanh ghi PR2
* Xóa bit TRIS tương ứng để chân CCPx là ngõ ra
* Thiết lập giá trị tiền tỉ lệ TMR2, sau đó bật timer2 bằng cách ghi vào thanh ghi T2CON
* Thiết đặt bộ CCPx hoạt động chế độ PWM

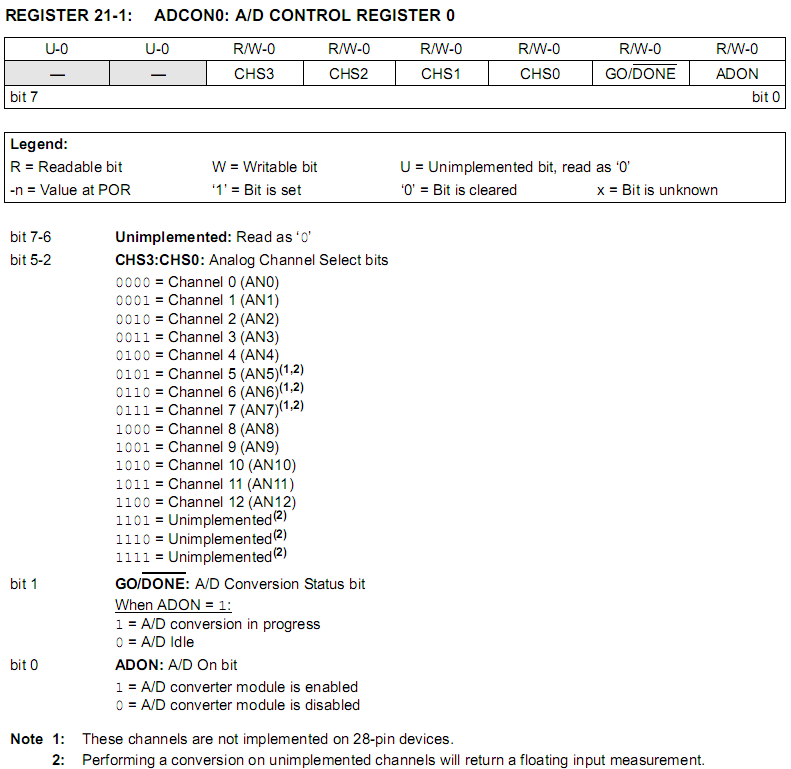


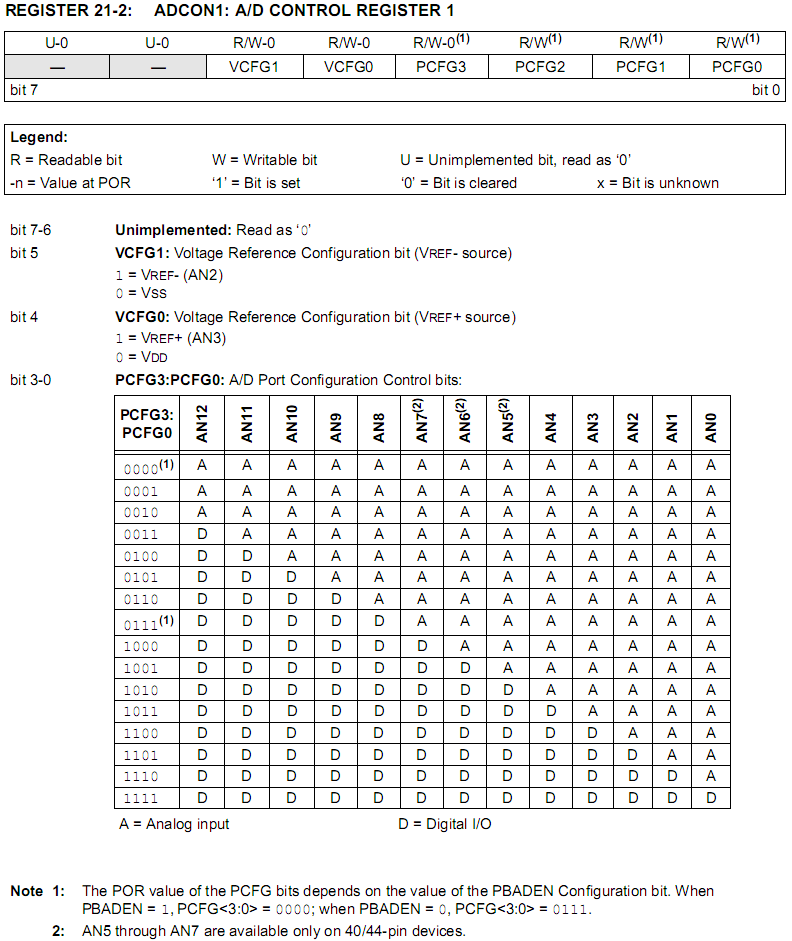
1. Bộ PWM cài tiến (Enhanced PWM module)
2. Bộ ADC 10 bit

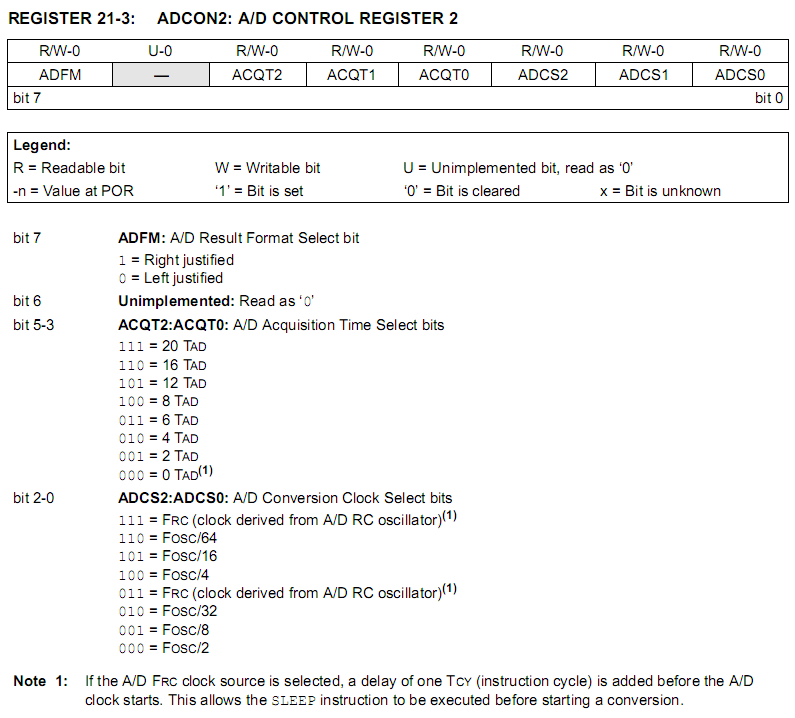
Bộ ADC có 10 ngõ vào cho thiết bị 28 chân và 13 cho 40/44 chân. ADC cho phép chuyển tín hiệu ngõ vào tương tự sang số 10 bit tương ứng.

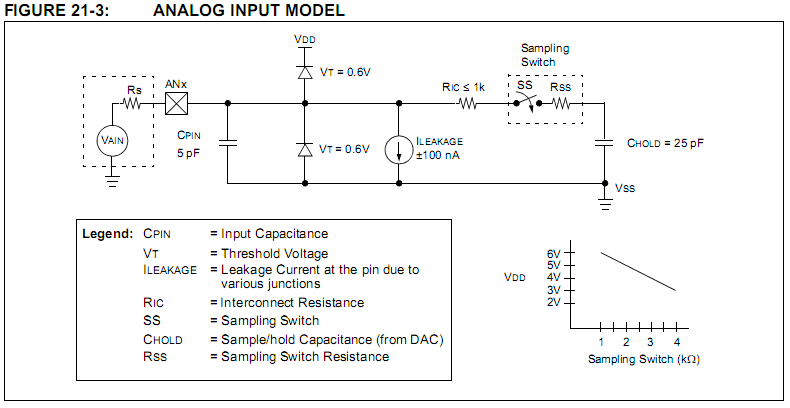
Bộ ADC có 5 thanh ghi :

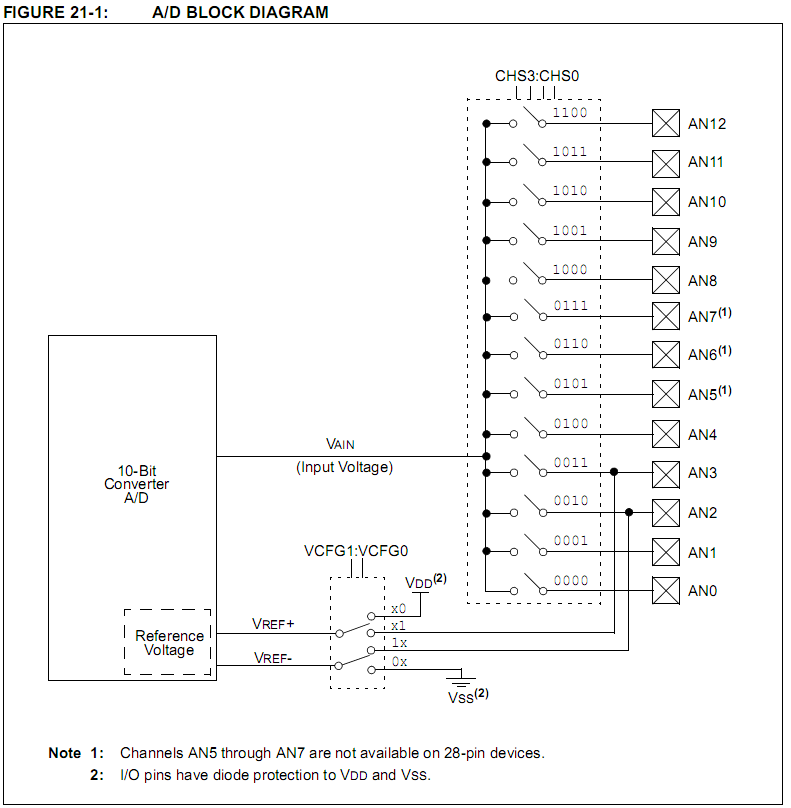
* A/D result high register ADRESH chứa kết quả chuyển đổi A/D
* A/D result low register ADRESL chứa kết quả chuyển đổi A/D
* A/D control register 0 ADCON0 điều khiển quá trình hoạt động bộ A/D
* A/D control register 1 ADCON1 cấu hình chức năng của chân cổng
* A/D control register 2 ADCON2 thiết đặt nguồn xung A/D, sự hiệu chỉnh và thời gian thu thập được lập trình.











Điện áp tham chiếu tương tự có thể chọn bằng phần mềm là bằng điện áp nguồn cung cấp (VDD và VSS) hay mức áp trên chân RA3/VREF+ vàRA2/VREF-

Bộ A/D có tính năng độc nhất là có khả năng vận hành trong khi thiết bị trong chế độ ngủ(Sleep mode). Để vận hành trong chế độ ngủ, xung định thời chuyển đổi A/D phải được rút ra từ bộ dao động RC bên trong bộ A/D.

Ngõ ra của mẫu và giữ là ngõ vào trong bộ chuyển đổi, cái tạo ra kết quả thông qua sự xấp xỉ liên tiếp.

Reset thiết bị buộc mọi thanh ghi trở về trạng thái reset. Việc này buộc bộ A/D bị tắt và quá trình chuyển đổi bị hủy bỏ.

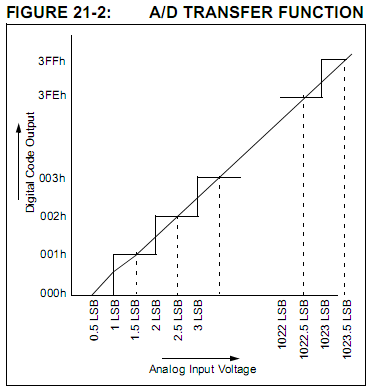
Mỗi chân cổng nối với bộ chuyển đổi A/D có thể được thiết đặt là ngõ vào tương tự hay ngõ xuất nhập số. Thanh ghi ADRESH và ADRESL chứa kết quả chuyển đổi A/D.

Khi chuyển đổi A/D hoàn tất, kết quả được nạp vào trong cặp thanh ghi ADRESH:ADRESL, bit GO/DONE (thanh ghi ADCON0) được xóa và bit cờ ngắt A/D ADIF được set. Sơ đổ khối của bộ A/D trong hình ngay trên.

Giá trị trong các thanh ghi ADRESH:ADRESL không biết được sau POR và BOR và không bị ảnh hưởng bởi các reset khác.

Sau khi đã thiết đặt bộ A/D như mong muốn, kênh được chọn phải nhận được tín hiệu trước khi bắt đầu chuyển đổi.

Các kênh ngõ vào tương tự phải có bit TRIS tương ứng được chọn là ngõ vào. Sau thời gian nhận tín hiệu trôi qua, chuyển đổi A/D có thể bắt đầu. Thời gian nhận có thể lập trình để diễn ra giữa thiết đặt bit GO/DONE và bắt đầu thực hiện chuyển đổi.

Các bước thực hiện để chuyển đổi A/D:

1. Cấu hình bộ A/D:
   1. Cấu hình chân analog, áp tham chiếu và cổng xuất nhập số (ADCON1)
   2. Chọn kênh ngõ vào A/D (ADCON0)
   3. Chọn thời gian nhận (ADCON2)
   4. Chọn xung định thời cho chuyển đổi A/D (ADCON2)
   5. Mở bộ A/D (ADCON0)
2. Cấu hình ngắt cho bộ A/D (nếu muốn)
   1. Xóa bit cờ ADIF
   2. Set bit ADIE
   3. Set bi GIE
3. Chờ thời gian nhận được yêu cầu (nếu yêu cầu)
4. Bắt đầu chuyển đổi:
   1. Set bit GO/DONE (ADCON0)
5. Chờ cho chuyển đổi A/D xong bằng cách :
   1. Thăm dò bit GO/DONE được xóa hay chưa
   2. Hay chờ ngắt A/D
6. Đọc thanh ghi kết quả A/D (ADRESH:ADRESL), xóa bit cờ ADIF nếu có yêu cầu
7. Để chuyển đổi tiếp, tới bước 1 hay 2 tùy yêu cầu. Thời gian chuyển đổi A/D cho 1 bit được xác định bởi TAD. Chờ đọi ít nhất là 3 TAD được đòi hỏi trước khi bắt đầu nhận tín hiệu tiếp.
8. Yêu cầu việc nhận A/D

Để bộ chuyển đổi A/D đạt độ chính xác của nó, việc xạc tụ giữ (CHOLD) phải được cho phép xạc đầy tới mức áp kênh vào. Tổng trở nguồn (Rs) và nội trở công tắc lấy mẫu (Rss) ảnh hưởng trực tiếp tới thời gian cần để xạc tụ. Tổng trở công tắc lấy mẫu thay đổi theo điện áp thiết bị (VDD). Tổng trở nguồn ảnh hưởng tới điện áp offset tại ngõ vào tương tự vì dòng rõ chân.

**Tổng trở tối đa đề nghị cho nguồn analog là 2.5 kΩ**

Sau khi kênh ngõ vào tương tự được chọn hay thay đổi, kênh phải được lấy mẫu trong thời gian tối thiểu bằng thời gian nhận nhỏ nhất trước khi bắt đầu chuyển đổi. Khi chuyển đổi bắt đầu, tụ giữ được ngắt kết nối với chân ngõ vào.

Đế tính toán thời gian nhận tối thiểu dùng biểu thức 21-1. Biểu thức này giả sử rằng lỗi ½ LSb được dùng (1024 bước cho A/D). Lỗi ½ LSb là lỗi lớn nhất được cho phép để A/D đạt tới độ phân giải đã nói.

Ví dụ 21-3 giả sử là :

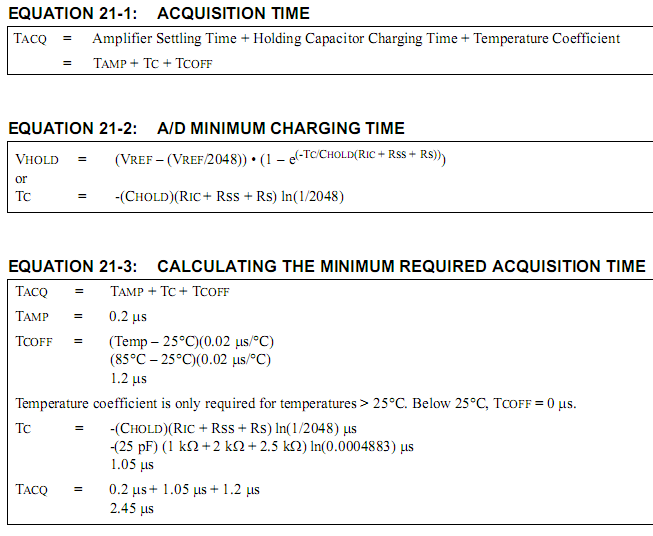
CHOLD = 25 pF

Rs = 2.5 kΩ

Conversion Error ≤ 1/2 LSb

VDD =5V → Rss= 2 kΩ

Temperature = 85°C (system max.)



1. Chọn và thiết lập thời gian nhận

Thanh ghi ADCON2 cho phép người dùng chọn thời gian nhận diễn ra mỗi lần bit GO/DONE được set. Nó cũng cho người dùng tùy chọn để quyết định thời gian nhận.

Thời gian nhận có thể được set bởi bit ACQT2:ACQT0 (ADCON2<5:3>) trong khoảng từ 2 tới 20 TAD.

Khi bit GO/DONE được set, bộ A/D tiếp tục lấy mẫu ngõ vào trong thời gian nhận đã chọn, sau đó tự động bắt đầu chuyển đổi. Vì thời gian nhận được lập trình, có thể không cần chờ thời gian nhận giữa việc chọn kênh và set bit GO/DONE.

Việc nhận bằng tay được chọn bằng cách cho ACQT2:ACQT0 = 000. Khi bit GO/DONE được set, việc lấy mẫu bị dừng và việc chuyển đổi bắt đầu. Người dùng chịu trách nhiệm bảo đảm thời gian nhận yêu cầu đã trôi qua giữa lúc chọn kênh ngõ vào mong muốn và set bit GO/DONE. Tùy chọn này là trạng thái reset mặc định của các bit ACQT2:ACQT0 và tương thích với các thiết bị không cho lập trình thời gian nhận.

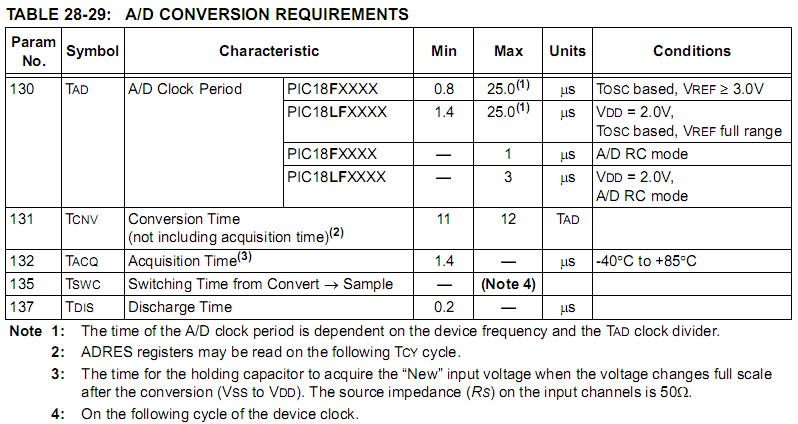
Trong cả hai trường hợp, khi mà chuyển đổi hoàn tất, bit GO/DONE bị xóa, cờ ADIF được set và A/D bắt đầu lấy mãu kênh hiện đang chọn tiếp. Nếu thời gian nhận được lập trình, không có gì chỉ ra là thời gian nhận đã hết hay chuyển đổi đã bắt đầu.

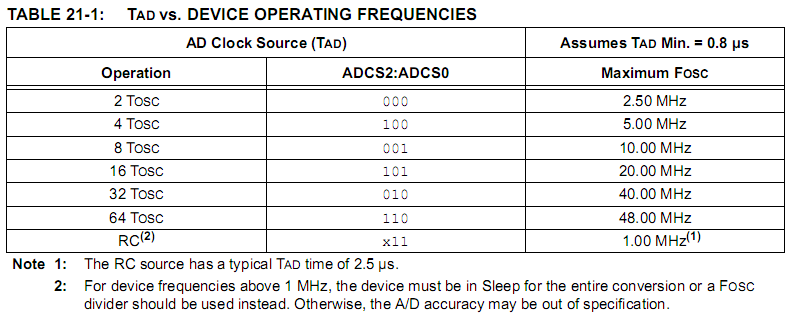
1. Chọn xung định thời chuyển đổi A/D

Thời gian chuyển đổi 1 bit được xác định là TAD. Chuyển đổi A/D đòi hỏi 11 TAD để chuyển đổi 10 bit. Nguồn của xung định thời chuyển đổi A/D có thể chọn bằng phần mềm. Có 7 tùy chọn có thể cho TAD :

* 2 Tosc
* 4 Tosc
* 8 Tosc
* 16 Tosc
* 32 Tosc
* 64 Tosc
* Bộ dao động RC trong

Để chuyển đổi A/D chính xác, xung định thời chuyển đổi A/D phải ngắn nhất có thể nhưng phải lớn hơn TAD tối thiểu. Xem bảng dưới





1. Vận hành trong chế dộ quản lý năng lượng

Sự lựa chọn thời gian nhận tự động và xung định thời chuyển đổi A/D được quyết định là một phần cảu nguồn xung và tần số trong khi trong chế độ quản lý năng lượng.

Nếu A/D được mong vận hành trong khi thiết bị trong chế độ quản lý năng lượng, ACQT2:ACQT0 và ADCS2:ADCS0 trong thanh ghi ADCON2 cần được cập nhật theo nguồn xung dùng trong chế độ đó. Sau khi vào chế độ, việc nhận hay chuyển A/D có thể bắt đầu. Một khi bắt đầu, thiết bị nên tiếp tục được cấp xung bởi cùng nguồn xung cho tới khi chuyển đổi hoàn tất.

Nếu muốn, thiết bị có thể được đặt vào chế độ rỗi (idle mode) trong khi chuyển đổi. Nếu tần số thiết bị ít hơn 1MHz, nguồn xung RC nên được chọn.

Vận hành trong chế độ ngủ yêu cầu xung FRC A/D được chọn. Nếu bit ACQT2:ACQT0 = ‘000’ và chuyển đổi bắt đầu, chuyển đổi sẽ trì hoạn 1 chu kỳ lệnh để cho phép thực thi lệnh ngủ và đi vào chế độ ngủ. Bit IDLEN (OSCCON<7>) phải vừa được xóa trước khi bắt đầu chuyển đổi.

1. Thiết đặt chân cổng tương tự

Các thanh ghi ADCÓN, TRISA, TRISB và TRISE tất cả được thiết đặt chân cổng A/D. Các chân cổng cần làm ngõ vào tương tự phải có bit TRIS tương ứng được set , nếu bị xóa, mức áp ra số (VOH hay VOL) sẽ được chuyển đổi.

Hoạt động A/D độc lập với trạng thái của bit CHS3:CHS0 và bit TRIS.

Khi đọc thanh ghi PORT, tất cả chân được thiết lập là kênh ngõ vào tương tự sẽ đọc ra mức thấp. Các chân thiết lập là ngõ vào số sẽ chuyển đổi là ngõ vào tương tự. Mức tương tự trên ngõ vào được thiết lập số sẽ được chuyển đổi chính xác.

Mức tương tự trên chân nào được xác định là ngõ vào số sẽ gây bộ đệm ngõ vào số tiêu thụ dòng vượt giới hạn đặc tính của thiết bị.

Bit PBADEN trong thanh ghi cấu hình 3H thiết đặt chân cổng B để khi reset là chân số hay tương tự bằng cách điều khiển bit PCFG0 như thế nào khi ADCON1 được reset.

1. Chuyển đổi A/D

Xóa bit GO/DONE trong quá trình chuyển đổi sẽ hủy bỏ việc chuyển dổi hiện thời. Căp thanh ghi ADRESH:ADRESL sẽ không được cập nhật với một phần giá trị chuyển đổi xong tức là vẫn chứa giá trị của lần chuyển đổi hoàn chỉnh trước đó (hay giá trị cuối được ghi vào)

Sau khi chuyển đổi A/D hoàn tất hay bị hủy bỏ, 2 TCY được yêu cầu trước khi việc nhận kế tiếp có thể được bắt đầu. Sau khi việc chờ này, việc nhận trên kệnh đã chọn có thể đươc bắt đầu tự động.

Chú ý : bit GO/DONE **không** nên được set trong cùng lệnh bật A/D. Mã lệnh nên chờ ít nhất 2 us sau khi kích hoạt bộ A/D trước khi bắt đầu chu kỳ nhận và chuyển đổi.



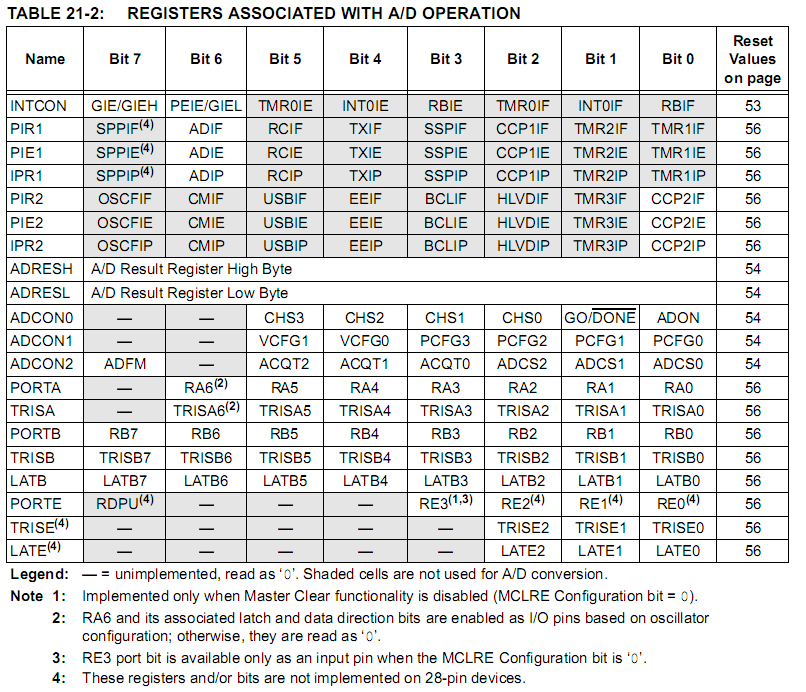
1. Xả tụ

Giai đoạn xả đươc dùng để bắt đầu giá trị của mảng tụ. Mảng được xả trước mỗi mẫu. Tính năng này giúp tối ưu bộ khuếch đại độ lợi đơn vị vì mạch thường cần nạp mảng tụ, hơn là nạp/xả dựa trên giá trị đo trước đó.

1. Dùng bộ kích khởi CCP2

Một chuyển đổi A/D có thể bắt đầu bởi tín hiệu kích khởi sự kiện đặc biệt của bộ CCP2. Điều này đỏi hỏi bit CCP2M3:CCP2M0 (CCP2CON<3:0>) = ‘1011’ và bộ A/D được kích hoạt (bit ADON = 1). Khi tín hiệu kích khởi diễn ra, bit GO/DONE sẽ được set, bắt đầu nhận và chuyển đổi và bộ đếm timer1 hay timer3 sẽ bị reset về 0. Timer1 hay timer3 bị reset để tự động lập lại thời gian nhận A/D với overhead phần mềm tối thiểu (di chuyển ADRESH:ADRESL tới nơi mong muốn). Kênh ngõ vào tương tự thích hợp phải được chọn và thời gian nhận tối thiểu được định thời bởi người dùng hay thời gian TACQ thích hợp được chọn trước khi tín hiệu kích khởi sự kiện đặc biệt set bit GO/DONE (bắt đầu chuyển đổi).

Nếu bộ A/D không được kích hoạt (ADON = 0), tín hiệu kích khởi sự kiện đặc biệt sẽ bị bỏ qua bởi bộ A/D nhưng vẫn sẽ reset bộ đếm timer1 hay timer3.



1. Bộ so sánh
2. Bộ tham chiếu điện áp so sánh
3. Tính năng đặc biệt CPU
4. Tập lệnh

Vấn đề refresh EEPROM : khi nào refresh , sau bao nhiêu lần ghi/xóa phải refresh ?

Scratchpad memory và scratchpad operation

Look up table