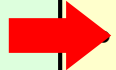


## ¿Qué veremos del PIC?

### Hardware



Organización de la memoria

- Puertos de Entrada/Salida
- Temporizadores/contadores (TMR0, TMR1, TMR2)
- ...

### Software

- Juego de instrucciones
- Ejemplos de programas
- Entorno de desarrollo (Prácticas)
- ...

# TECNOLOGÍA ELECTRÓNICA DE COMPUTADORES

2º Curso – GRADO EN INGENIERÍA INFORMÁTICA  
EN TECNOLOGÍAS DE LA INFORMACIÓN

*Tema 9: Circuitos integrados: microcontroladores*

*Lección 18. Microcontroladores PIC 16xxx:  
organización de la memoria*

## Lección 18. Microcontroladores PIC 16xxx: organización de la memoria

### 18.1. Organización de la memoria

### 18.2. Memoria de datos

RAM de datos: bancos de memoria

Formas de direccionamiento

Registros especiales

### 18.3. Memoria de programa

Memoria de programa, contador de programa (PC) y stack (pila)

Vectores de RESET e Interrupción. Información de calibración

Instrucciones que actúan sobre el PC (formas de salto)

## **18.1. Organización de la memoria**

Dentro del PIC16F877 se distinguen tres bloques de memoria principales:

### **Memoria de programa (8Kwords)**

En sus 8192 posiciones de 14 bits (8Kwords) contiene el programa con las instrucciones que gobiernan la aplicación. Es del tipo no volátil.

### **Memoria de datos RAM**

Guarda las variables y datos. Son registros de 8 bits. Es volátil.

### **Memoria EEPROM de datos**

Es una pequeña área de memoria de datos de lectura y escritura no volátil que permite garantizar que determinada información estará siempre disponible al reiniciarse el microcontrolador tras un “apagado” del mismo. Se gestiona de manera distinta a la memoria de datos RAM.

## Memoria de programa y de datos en Microcontrolador PIC16F877A (familia media)

$$2^{13}=8K$$

$$2^9=512$$

$$2^8=256$$

**EEPROM  
PROGRAMA**

**8K x 14**

Flash  
Program  
Memory

13

Program Counter

13

8 Level Stack  
(13-bit)

**8 x 13 LIFO**

**STACK (PILA)**

Data Bus

8

RAM  
File  
Registers

RAM Addr(1)

9

**368 x 8**

**RAM DATOS**

Program  
Bus

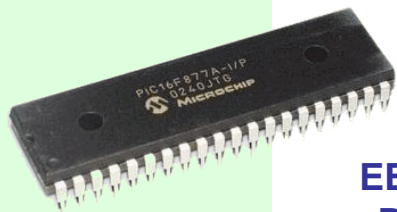
14

**EEPROM  
DATOS**

**256 x 8**

Data EEPROM

8



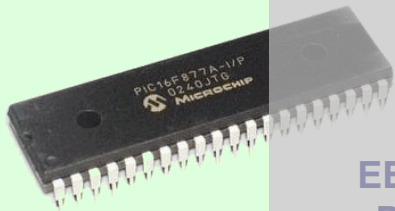
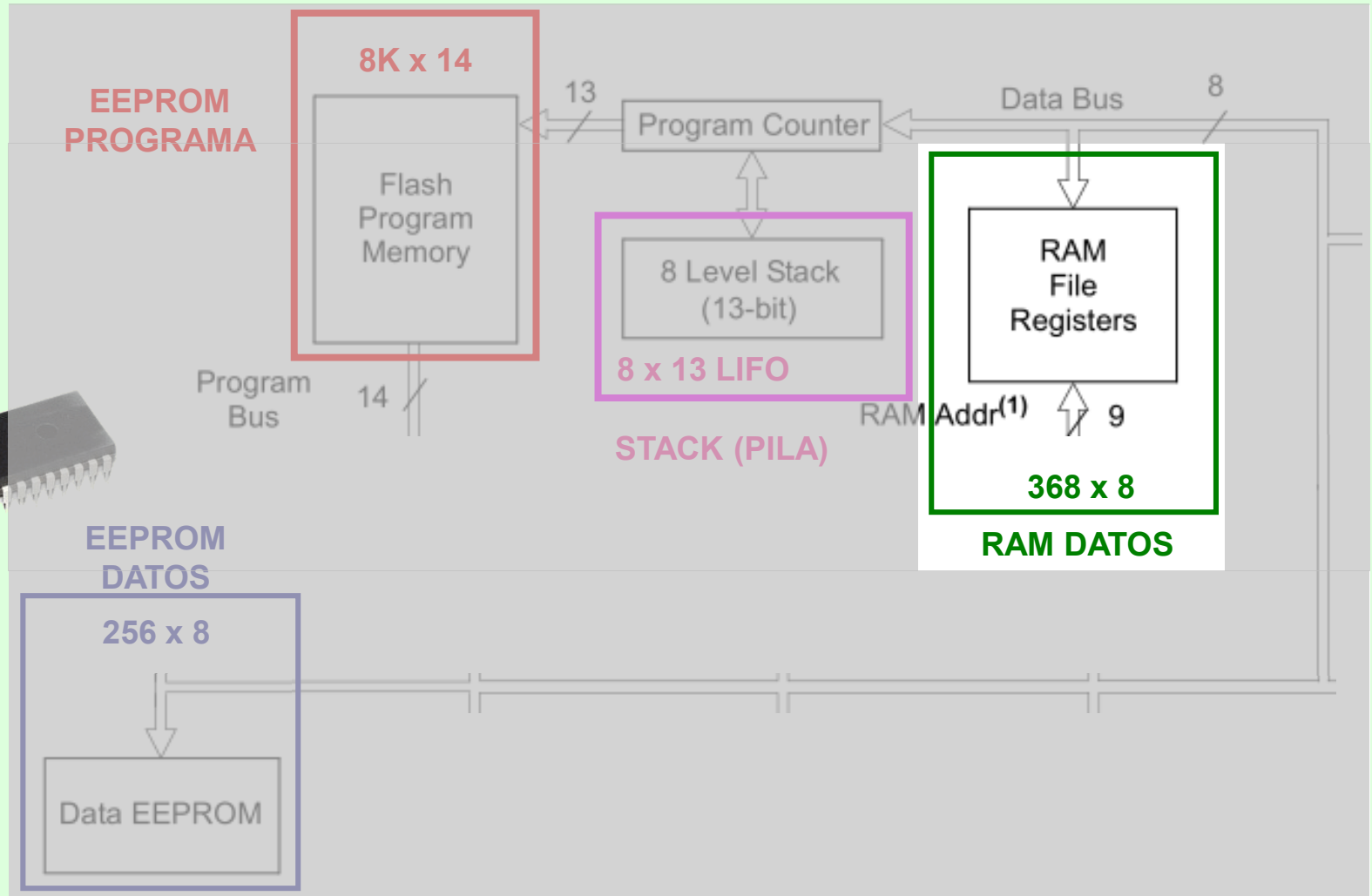
## 18.2. Memoria de datos

### Memoria RAM de datos ( $2^9 \times 8$ )

$$2^{13}=8K$$

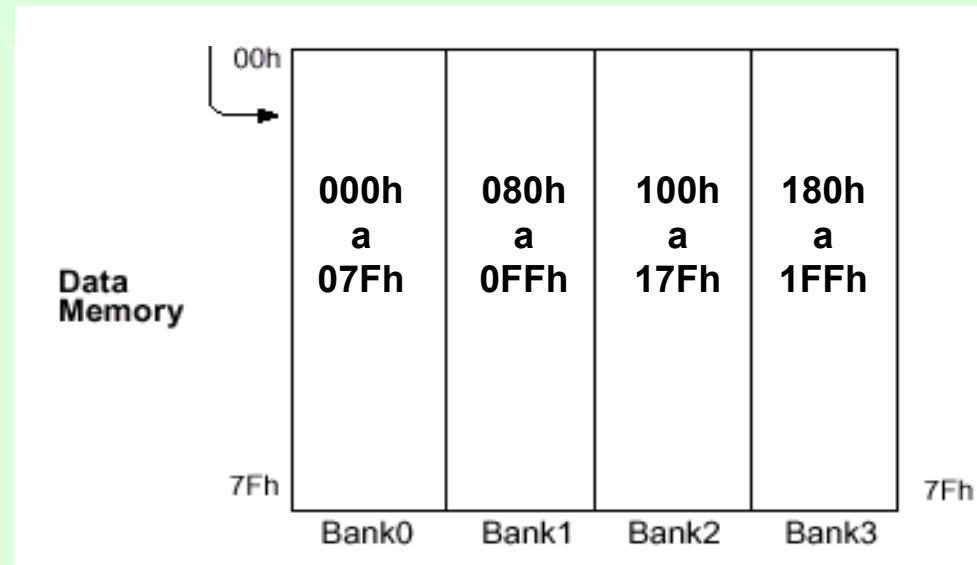
$$2^9=512$$

$$2^8=256$$



## Memoria RAM de datos ( $2^9 \times 8 = 512 \times 8$ )

- **Capacidad.** La máxima cantidad de memoria disponible en los microcontroladores PIC16 es de 512 bytes, pero **no están implementadas todas las posiciones de memoria**. Ejemplo: un PIC16F876 solo tiene implementadas 368 posiciones de memoria de datos.
- **Registros.** A cada posición de la memoria de datos se le denomina registro (File).
- **Bancos.** La memoria de datos está **distribuida en 4 posibles bancos de 128 bytes** cada uno, porque cuando se indica una dirección de operando fuente, sólo se pueden incluir 7 bits en la codificación de operación ( $2^7=128$  bytes). Los bits que faltan se toman de otro registro (status).



## Registro STATUS: bits para selección de banco: RP1:RP0 e IRP

### STATUS REGISTER (ADDRESS 03h, 83h, 103h, 183h)

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	TO	PD	Z	DC	C
bit 7							bit 0

#### Legend:

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

bit 7 **IRP**: Register Bank Select bit (used for indirect addressing)  
 1 = Bank 2, 3 (100h - 1FFh)  
 0 = Bank 0, 1 (00h - FFh)

bit 6-5 **RP1:RP0**: Register Bank Select bits (used for direct addressing)  
 11 = Bank 3 (180h - 1FFh)  
 10 = Bank 2 (100h - 17Fh)  
 01 = Bank 1 (80h - FFh)  
 00 = Bank 0 (00h - 7Fh)  
 Each bank is 128 bytes

bit 4 **TO**: Time-out bit  
 1 = After power-up, CLRWDI instruction, or SLEEP instruction  
 0 = A WDT time-out occurred

bit 3 **PD**: Power-down bit  
 1 = After power-up or by the CLRWDI instruction  
 0 = By execution of the SLEEP instruction

bit 2 **Z**: Zero bit

1 = The result of an arithmetic or logic operation is zero  
 0 = The result of an arithmetic or logic operation is not zero

bit 1 **DC**: Digit carry/borrow bit (ADDWF, ADDLW, SUBLW, SUBWF instructions)

(for borrow, the polarity is reversed)

1 = A carry-out from the 4th low order bit of the result occurred  
 0 = No carry-out from the 4th low order bit of the result

bit 0 **C**: Carry/borrow bit (ADDWF, ADDLW, SUBLW, SUBWF instructions)

1 = A carry-out from the Most Significant bit of the result occurred  
 0 = No carry-out from the Most Significant bit of the result occurred

Note: For borrow, the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand. For rotate (RRF, RLF) instructions, this bit is loaded with either the high, or low order bit of the source register.



## Memoria RAM de datos

- **Función:** Almacena todas los **datos que se manejan** en un programa.
- Se distinguen dos tipos de registro:
  - **Registros de funciones especiales SFR.** Son los primeros registros. Cada uno de ellos cumple un propósito especial en el control del PIC.
  - **Registros de propósito general GPR.** Se pueden usar para guardar datos temporales. El PIC16F877 dispone de 368.
- Cuenta con cuatro bancos de memoria: Bancos 0, 1, 2 y 3.
  - Los SFR aparecen de la dirección 00h a 1Fh del Banco 0, de 80h a 9Fh del Banco 1, de 100h a 10F en el Banco 2 y de 180h a 18Fh del Banco 3. Algunos son accesibles desde dos o más bancos.
  - Los GPR ocupan 368 posiciones de memoria. Algunas posiciones de los Bancos 1 a 3 se mapean sobre el Banco 0.
  - Existen zonas de memoria no empleadas que devuelven '0' en caso de lectura.

**GPR**  
**Registros de propósito**  
**general**

[illegible]

## Modos de direccionamiento de la memoria de datos

- Existen **2 modos de direccionamiento** para acceder a cualquiera de las posiciones de la memoria de datos:

**Direccionamiento directo**

**Direccionamiento indirecto**

- Direccionamiento directo: La posición de memoria con la que se trabaja viene **directamente definida en el código de la instrucción** (7bit + 2 bit RP1:RP0 de STATUS)

- Direccionamiento indirecto: La posición de memoria con la que se trabaja viene **definida por el contenido del registro FSR** (Posición 04h, 84h, 104h ó 184h), es decir, el registro FSR actúa como puntero de la posición de memoria con la que se pretende operar (8bit + 1 bit IRP de STATUS)

## Modos de direccionamiento

- Existen 2 modos de direccionamiento de las posiciones de la memoria

Direcciónamiento directo

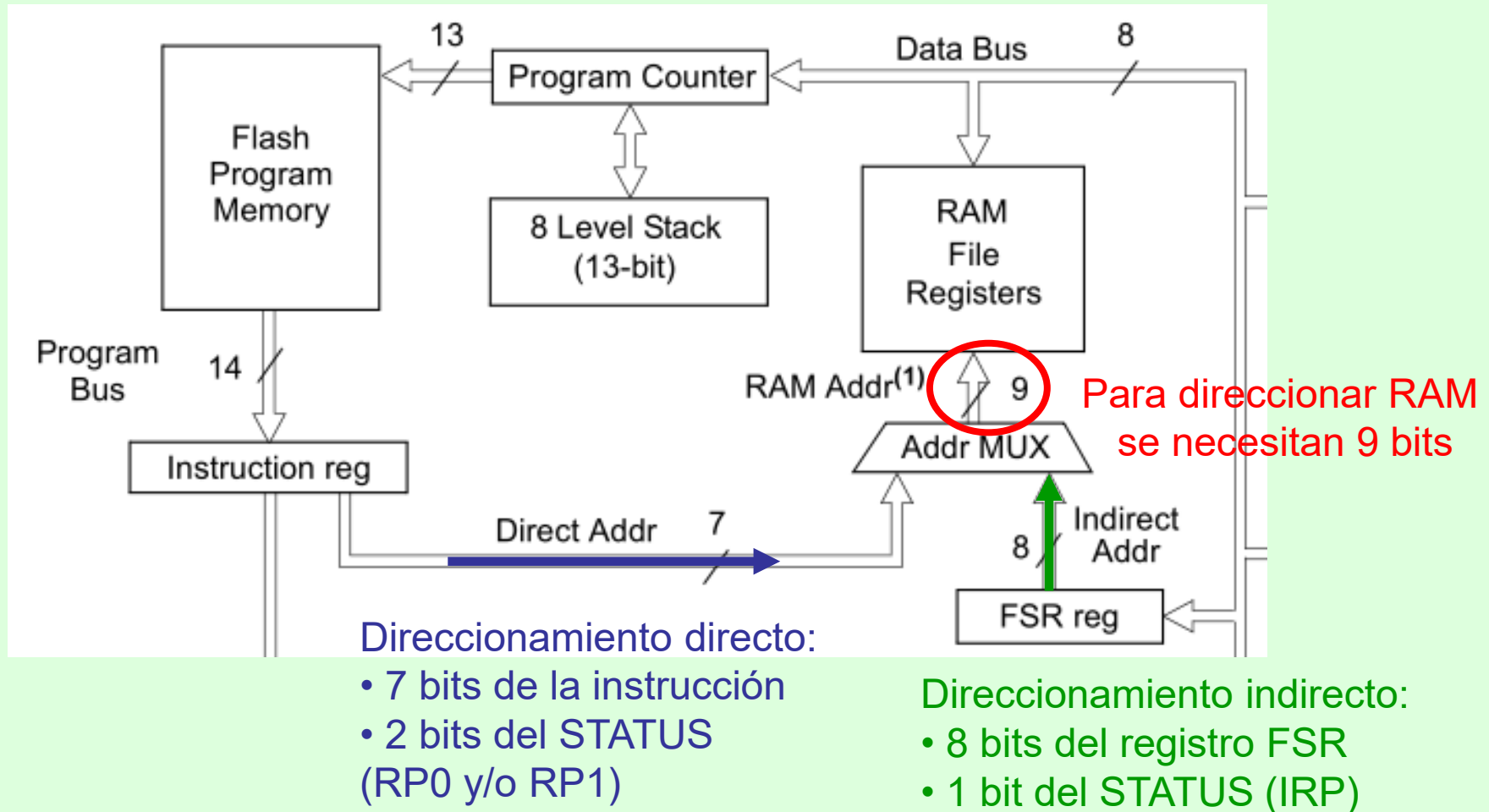
Direcciónamiento indirecto

- Direcciónamiento directo (la dirección de memoria está directamente definida en el registro STATUS)

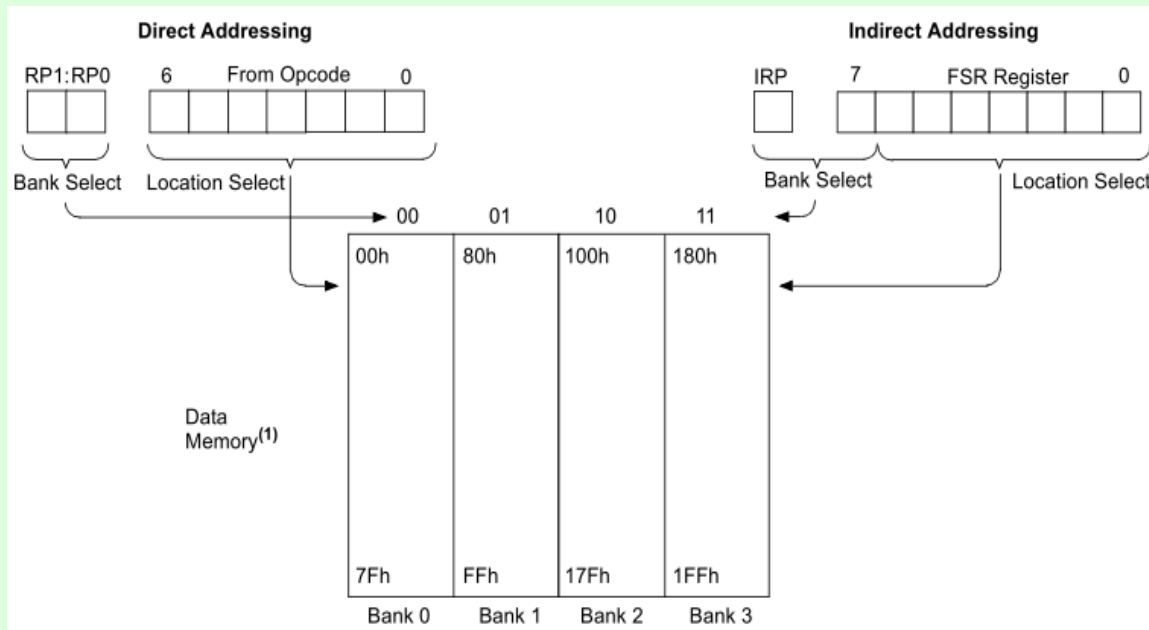
- Direcciónamiento indirecto (la dirección de memoria está definida por el contenido del registro FSR, es decir, el registro FSR actúa como puntero a la memoria que se pretende operar (8bit + 8bit = 16bit))

File Address	File Address	File Address	File Address
Indirect addr. <sup>(1)</sup> 00h	Indirect addr. <sup>(1)</sup> 80h	Indirect addr. <sup>(1)</sup> 100h	Indirect addr. <sup>(1)</sup> 180h
TMR0 01h	OPTION_REG 81h	TMR0 101h	OPTION_REG 181h
PCL 02h	PCL 82h	PCL 102h	PCL 182h
STATUS 03h	STATUS 83h	STATUS 103h	STATUS 183h
<b>FSR 04h</b>	<b>FSR 84h</b>	<b>FSR 104h</b>	<b>FSR 184h</b>
PORTA 05h	TRISA 85h	PORTB 105h	TRISB 185h
PORTB 06h	TRISB 86h	PORTB 106h	TRISB 186h
PORTC 07h	TRISC 87h		
PORTD <sup>(1)</sup> 08h	TRISD <sup>(1)</sup> 88h		
PORTE <sup>(1)</sup> 09h	TRISE <sup>(1)</sup> 89h		
PCLATH 0Ah	PCLATH 8Ah	PCLATH 10Ah	PCLATH 18Ah
INTCON 0Bh	INTCON 8Bh	INTCON 10Bh	INTCON 18Bh
PIR1 0Ch	PIE1 8Ch	EEDATA 10Ch	EECON1 18Ch
PIR2 0Dh	PIE2 8Dh	EEADR 10Dh	EECON2 18Dh
TMR1L 0Eh	PCON 8Eh	EEDATH 10Eh	Reserved <sup>(2)</sup> 18Eh
TMR1H 0Fh		EEADRH 10Fh	Reserved <sup>(2)</sup> 18Fh
T1CON 10h			
TMR2 11h	SSPCON2 91h		
T2CON 12h	PR2 92h		
SSPBUF 13h	SSPAD 93h		
SSPCON 14h	SSPSTAT 94h		
CCPR1L 15h			
CCPR1H 16h			
CCP1CON 17h			
RCSTA 18h	TXSTA 98h		
TXREG 19h	SPBRG 99h		
RCREG 1Ah			
CCPR2L 1Bh			
CCPR2H 1Ch	CMCON 9Ch		
CCP2CON 1Dh	CVRCON 9Dh		
ADRESH 1Eh	ADRESL 9Eh		
ADCON0 1Fh	ADCON1 9Fh		
General Purpose Register 96 Bytes	General Purpose Register 80 Bytes	General Purpose Register 80 Bytes	General Purpose Register 80 Bytes
	accesses 70h-7Fh	accesses 70h-7Fh	accesses 70h-7Fh
Bank 0 7Fh	Bank 1 FFh	Bank 2 17Fh	Bank 3 1FFh

## Modos de direccionamiento de la memoria de datos



## Modos de direccionamiento de la memoria de datos



BANCO ACCEDIDO	DIRECTO (RP1:RP0)	INDIRECTO (IRP)
0	0 0	0
1	0 1	
2	1 0	1
3	1 1	

Tanto con direccionamiento directo como indirecto **la dirección completa (9 bits)** no se puede obtener del código de la operación (7 bits) o del registro FSR (8 bits).

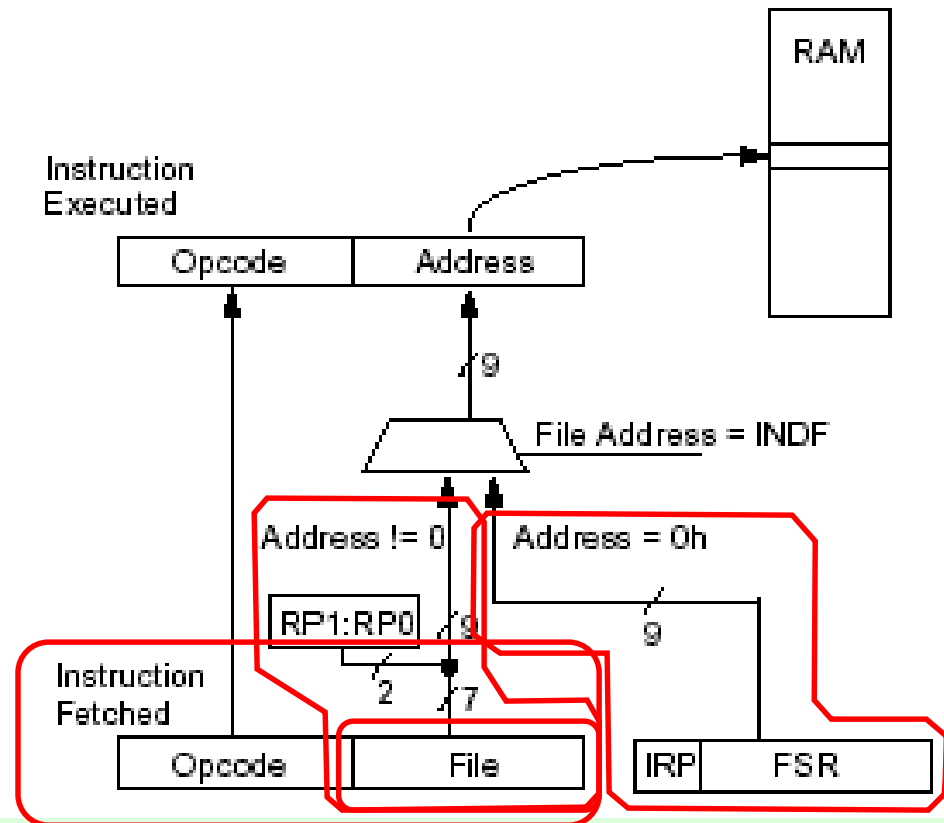
- Direccionamiento **directo**, la dirección a operar se obtiene **completando la dirección incluida en el código de la instrucción con los bits RP1:RP0 del registro STATUS**. (Ver tabla)
- Direccionamiento **indirecto**, la dirección se obtiene **completando el contenido del registro FSR con el bit IRP del registro STATUS**. (Ver tabla).

## Resumen de cómo se obtiene la dirección en el direccionamiento directo e indirecto

1) Se lee el registro (7 bits) contenido en el código de instrucción.

2) Direccionamiento directo:  
Si el registro es diferente de INDF (00h, 80h, 100h ó 180h) la instrucción se ejecuta sobre el registro indicado (7 bits), complementado con RP1:RP0 del registro STATUS (2 bits, que indican el banco)

3) Direccionamiento indirecto:  
Si el registro es INDF (00h, 80h, 100h ó 180h), la instrucción se ejecuta sobre el registro indicado por FSR (8 bits) complementado con el bit IRP del registro STATUS

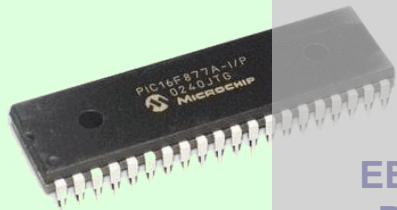
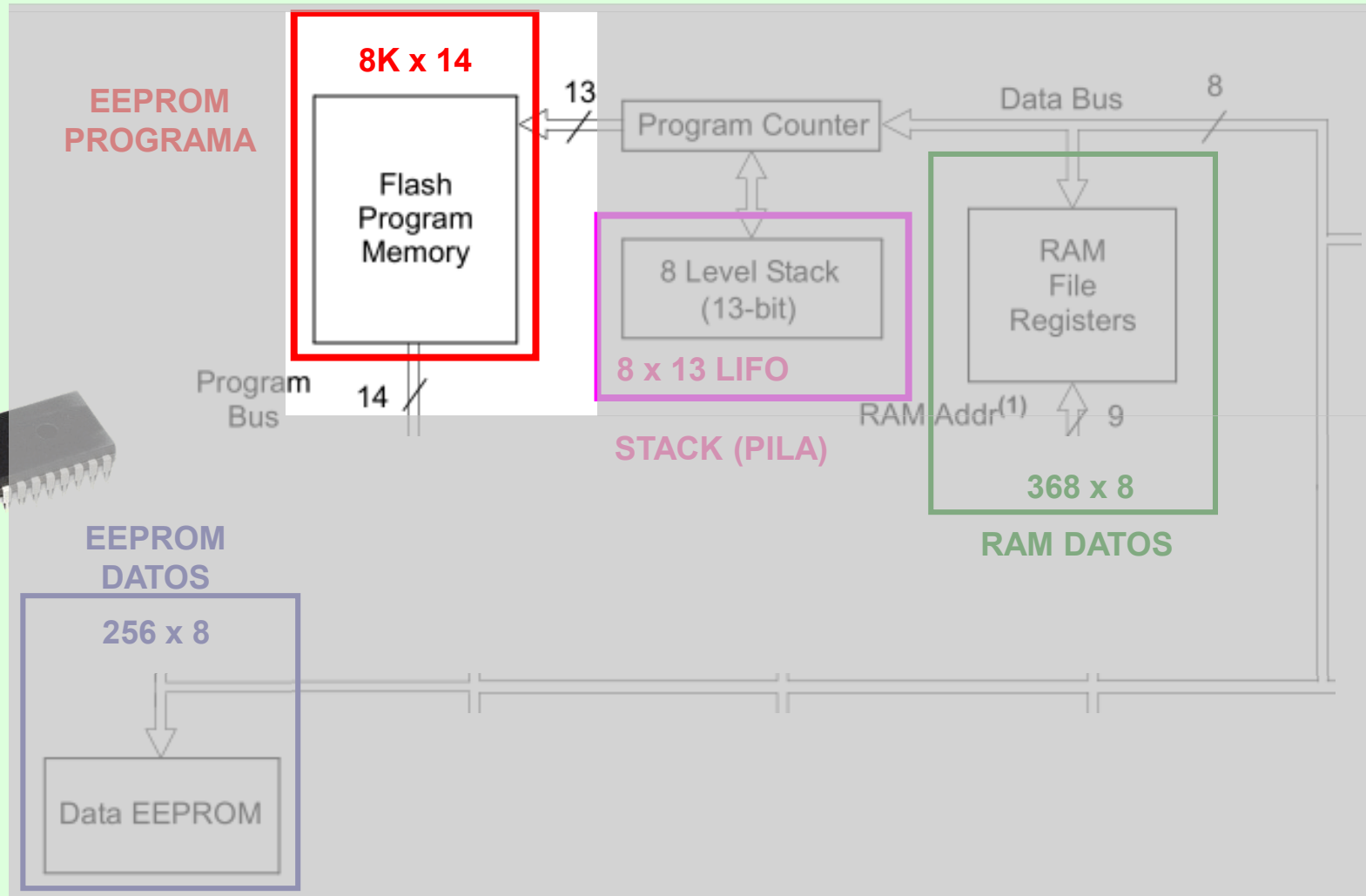


## 18.3. Memoria de programa

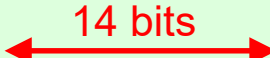
$$2^{13}=8K$$

$$2^9=512$$

$$2^8=256$$







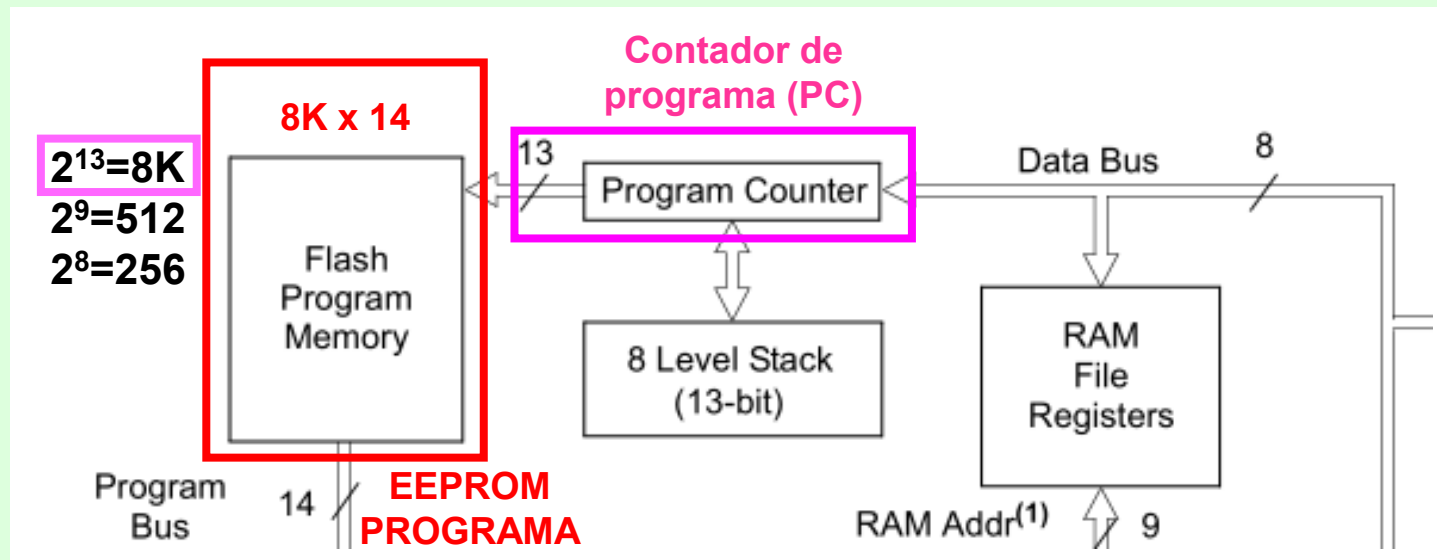
0000h	RESET
0001h	
0002h	
0003h	
0004h	INT
0005h	
0006h	
0007h	
0008h	
0009h	
000Ah	
000Bh	
000Ch	
000Dh	
000Eh	
000Fh	
0010h	
0011h	
0012h	
0013h	
0014h	
0015h	
0016h	
0017h	
0018h	
0019h	
001Ah	
001Bh	
001Ch	
001Dh	
001Eh	
001Fh	
0020h	
0021h	
0022h	
0023h	
0024h	
0025h	
0026h	
0027h	
0028h	
0029h	
002Ah	
002Bh	
002Ch	
002Dh	
002Eh	
002Fh	
0030h	
0031h	
0032h	
0033h	
0034h	
0035h	
0036h	
0037h	
0038h	
0039h	
003Ah	
003Bh	
003Ch	
003Dh	
003Eh	
003Fh	
0040h	
0041h	
0042h	
0043h	
0044h	
0045h	
0046h	
0047h	
0048h	
0049h	
004Ah	
004Bh	
004Ch	
004Dh	
004Eh	
004Fh	
0050h	
0051h	
0052h	
0053h	
0054h	
0055h	
0056h	
0057h	
0058h	
0059h	
005Ah	
005Bh	
005Ch	
005Dh	
005Eh	
005Fh	
0060h	
0061h	
0062h	
0063h	
0064h	
0065h	
0066h	
0067h	
0068h	
0069h	
006Ah	
006Bh	
006Ch	
006Dh	
006Eh	
006Fh	
0070h	
0071h	
0072h	
0073h	
0074h	
0075h	
0076h	
0077h	
0078h	
0079h	
007Ah	
007Bh	
007Ch	
007Dh	
007Eh	
007Fh	
0080h	
0081h	
0082h	
0083h	
0084h	
0085h	
0086h	
0087h	
0088h	
0089h	
008Ah	
008Bh	
008Ch	
008Dh	
008Eh	
008Fh	
0090h	
0091h	
0092h	
0093h	
0094h	
0095h	
0096h	
0097h	
0098h	
0099h	
009Ah	
009Bh	
009Ch	
009Dh	
009Eh	
009Fh	
00A0h	
00A1h	
00A2h	
00A3h	
00A4h	
00A5h	
00A6h	
00A7h	
00A8h	
00A9h	
00AAh	
00ABh	
00ACh	
00ADh	
00AEh	
00AFh	
00B0h	
00B1h	
00B2h	
00B3h	
00B4h	
00B5h	
00B6h	
00B7h	
00B8h	
00B9h	
00BAh	
00BCh	
00BDh	
00BEh	
00BFh	
00C0h	
00C1h	
00C2h	
00C3h	
00C4h	
00C5h	
00C6h	
00C7h	
00C8h	
00C9h	
00CAh	
00CBh	
00CCh	
00CDh	
00CEh	
00CFh	
00D0h	
00D1h	
00D2h	
00D3h	
00D4h	
00D5h	
00D6h	
00D7h	
00D8h	
00D9h	
00DAh	
00DBh	
00DCh	
00DDh	
00DEh	
00DFh	
00E0h	
00E1h	
00E2h	
00E3h	
00E4h	
00E5h	
00E6h	
00E7h	
00E8h	
00E9h	
00EAh	
00EBh	
00ECh	
00EDh	
00EEh	
00EFh	
00F0h	
00F1h	
00F2h	
00F3h	
00F4h	
00F5h	
00F6h	
00F7h	
00F8h	
00F9h	
00FAh	
00FBh	
00FCh	
00FDh	
00FEh	
00FFh	
0100h	
0101h	
0102h	
0103h	
0104h	
0105h	
0106h	
0107h	
0108h	
0109h	
010Ah	
010Bh	
010Ch	
010Dh	
010Eh	
010Fh	
0110h	
0111h	
0112h	
0113h	
0114h	
0115h	
0116h	
0117h	
0118h	
0119h	
011Ah	
011Bh	
011Ch	
011Dh	
011Eh	
011Fh	
0120h	
0121h	
0122h	
0123h	
0124h	
0125h	
0126h	
0127h	
0128h	
0129h	
012Ah	
012Bh	
012Ch	
012Dh	
012Eh	
012Fh	
0130h	
0131h	
0132h	
0133h	
0134h	
0135h	
0136h	
0137h	
0138h	
0139h	
013Ah	
013Bh	
013Ch	
013Dh	
013Eh	
013Fh	
0140h	
0141h	
0142h	
0143h	
0144h	
0145h	
0146h	
0147h	
0148h	
0149h	
014Ah	
014Bh	
014Ch	
014Dh	
014Eh	
014Fh	
0150h	
0151h	
0152h	
0153h	
0154h	
0155h	
0156h	
0157h	
0158h	
0159h	
015Ah	
015Bh	
015Ch	
015Dh	
015Eh	
015Fh	
0160h	
0161h	
0162h	
0163h	
0164h	
0165h	
0166h	
0167h	
0168h	
0169h	
016Ah	
016Bh	
016Ch	
016Dh	
016Eh	
016Fh	
0170h	
0171h	
0172h	
0173h	
0174h	
0175h	
0176h	
0177h	
0178h	
0179h	
017Ah	
017Bh	
017Ch	
017Dh	
017Eh	
017Fh	
0180h	
0181h	
0182h	
0183h	
0184h	
0185h	
0186h	
0187h	
0188h	
0189h	
018Ah	
018Bh	
018Ch	
018Dh	
018Eh	
018Fh	
0190h	
0191h	
0192h	
0193h	
0194h	
0195h	
0196h	
0197h	
0198h	
0199h	
019Ah	
019Bh	
019Ch	
019Dh	
019Eh	
019Fh	
01A0h	
01A1h	
01A2h	
01A3h	
01A4h	
01A5h	
01A6h	
01A7h	
01A8h	
01A9h	
01AAh	
01ABh	
01ACh	
01ADh	
01AEh	
01AFh	
01B0h	
01B1h	
01B2h	
01B3h	
01B4h	
01B5h	
01B6h	
01B7h	
01B8h	
01B9h	
01BAh	
01BCh	
01BDh	
01BEh	
01BFh	
01C0h	
01C1h	
01C2h	
01C3h	
01C4h	
01C5h	
01C6h	
01C7h	
01C8h	
01C9h	
01CAh	
01CBh	
01CCh	
01CDh	
01CEh	
01CFh	
01D0h	
01D1h	
01D2h	
01D3h	
01D4h	
01D5h	
01D6h	
01D7h	
01D8h	
01D9h	
01DAh	
01DBh	
01DCh	
01DDh	
01DEh	
01DFh	
01E0h	
01E1h	
01E2h	
01E3h	
01E4h	
01E5h	
01E6h	
01E7h	
01E8h	
01E9h	
01EAh	
01EBh	
01ECh	
01EDh	
01EEh	
01EFh	
01F0h	
01F1h	
01F2h	
01F3h	
01F4h	
01F5h	
01F6h	
01F7h	
01F8h	
01F9h	
01FAh	
01FBh	
01FCh	
01FDh	
01FEh	
01FFh	

## La memoria de programa (8K x 14)

- Almacena todas las instrucciones del **programa en ejecución**, que estará grabado de forma permanente.
- La información contenida en esta memoria se graba previamente mediante un equipo físico denominado **programador o grabador**.
- El PIC16F877 tiene una memoria de programa no volátil denominada **ROM Flash** que admite unas 1000 grabaciones.
- La memoria de programa está organizada en **palabras de 14 bits** cada una.
- Todas las instrucciones **ocupan una posición de memoria** de programa
- A la **dirección 0** se accederá tras un RESET y a la **posición 4** tras una interrupción cualquiera

## Memoria de programa – El Contador de Programa (PC)

- A la memoria de programa de los PIC16 (entre ellos los PIC16F87X) se accede mediante **un registro de 13 bits** que actúa como puntero de la instrucción que se debe ejecutar en el siguiente ciclo de instrucción.
- Este registro de 13 bits se denomina **Contador de Programa** (Program Counter) y su acrónimo es **PC**.



memoria.  
ts y para  
esas 8K  
contador de  
ones.

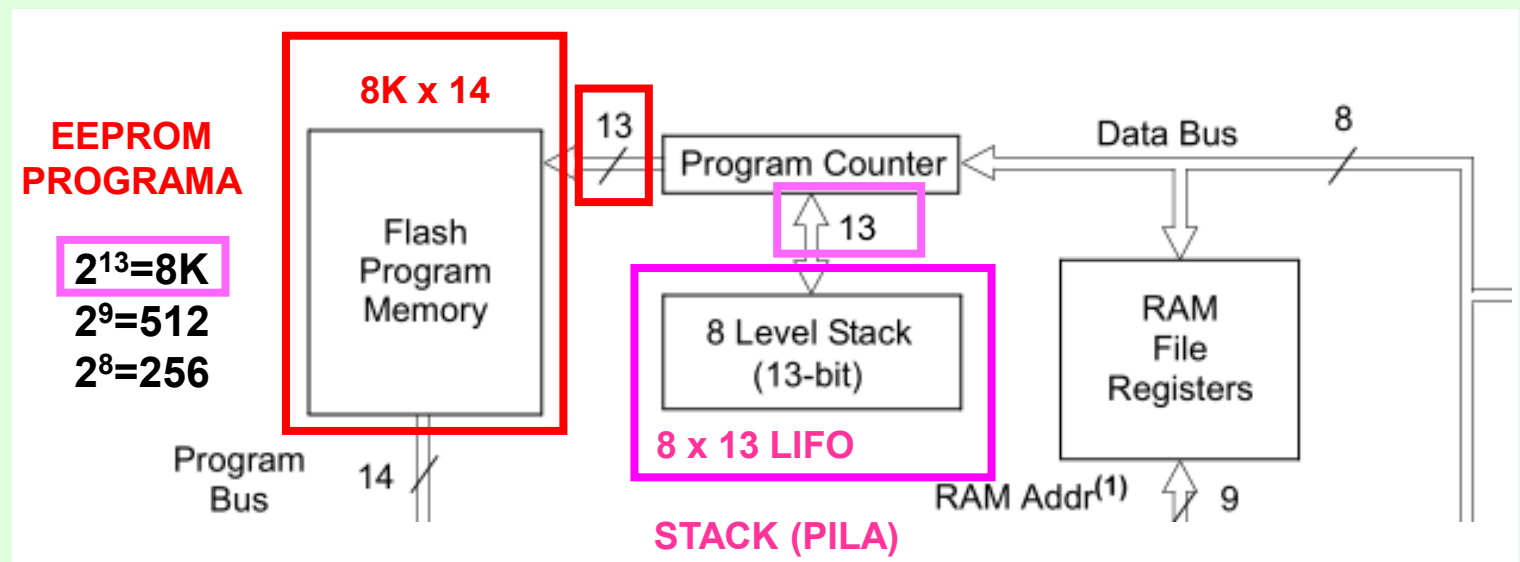
a mucho  
ente para

## Memoria de programa – El Contador de Programa (PC)

- A la memoria de programa de los PIC16 (entre ellos los PIC16F87X) se accede mediante **un registro de 13 bits** que actúa como puntero de la instrucción que se debe ejecutar en el siguiente ciclo de instrucción.
- Este registro de 13 bits se denomina **Contador de Programa** (Program Counter) y su acrónimo es **PC**.
- Con un PC de 13 bits **se pueden direccionar 8K** posiciones de memoria. Debido a que la codificación de **las instrucciones son de 14 bits** y para aprovechar las ventajas de la arquitectura Harvard, cada una de esas 8K posiciones corresponde a una instrucción y por tanto, el contador de programa es capaz de direccionar **8K x 14 posiciones ó 8K instrucciones**.
- Como cada instrucción ocupa una posición de memoria resulta mucho más fácil saber si un dispositivo tiene memoria de programa suficiente para una aplicación.

## La Pila (Stack) para almacenar el PC (Contador de Programa)

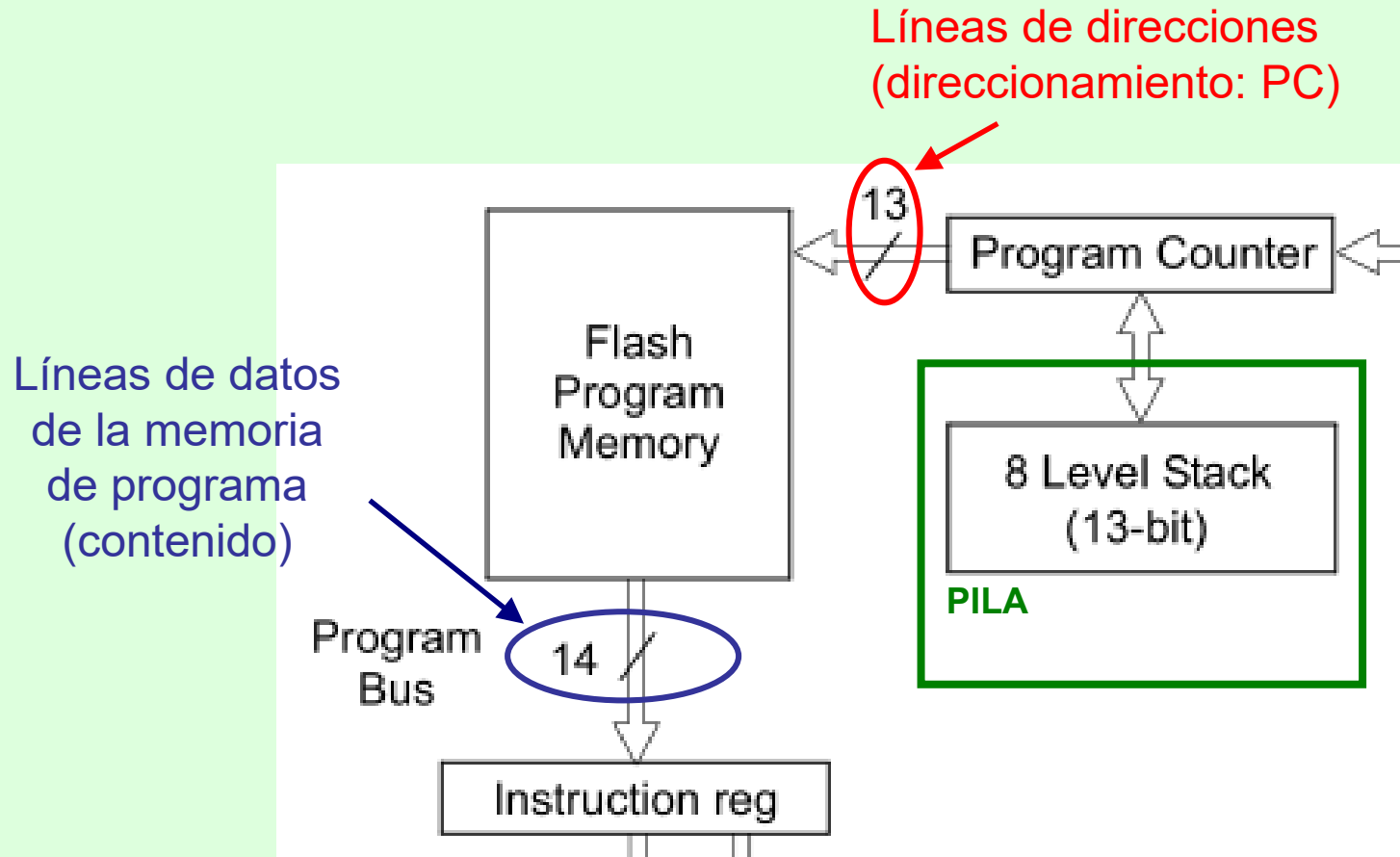
- La pila **permite almacenar las direcciones (PCs) a donde debe retornar** el programa cuando se finaliza una llamada a una subrutina o cuando se finaliza la ejecución de una rutina de interrupción.
- Los microcontroladores de la familia PIC16 tienen una **pila de 8 niveles x 13 bits**, que por tanto permite concatenar como máximo 8 saltos a subrutinas (CALLs) o ejecuciones de rutinas de interrupción (salto a 0004h).



## La Pila (Stack) para almacenar el PC (Contador de Programa)

- La pila **permite almacenar las direcciones (PCs) a donde debe retornar** el programa cuando se finaliza una llamada a una subrutina o cuando se finaliza la ejecución de una rutina de interrupción.
- Los microcontroladores de la familia PIC16 tienen una **pila de 8 niveles x 13 bits, que por tanto permite concatenar como máximo 8 saltos a subrutinas (CALLs) o ejecuciones de rutinas de interrupción (salto a 0004h).**
- El espacio de memoria para la pila no forma parte de la memoria de programa ni de la memoria de datos que tiene el microcontrolador. **Es un espacio de memoria totalmente independiente.** El puntero de pila no se puede leer ni escribir. La pila es gestionada por el hardware.
- Al contrario que en otros micros, no se dispone de una pila en RAM que se pueda gestionar por software (no hay inst.“PUSH” ni “POP”)

## Acceso a la memoria de programa: paginado



- Ninguna instrucción permite proporcionar los 13 bits completos al direccionar: La memoria se dividirá en “páginas” más pequeñas para direccionar

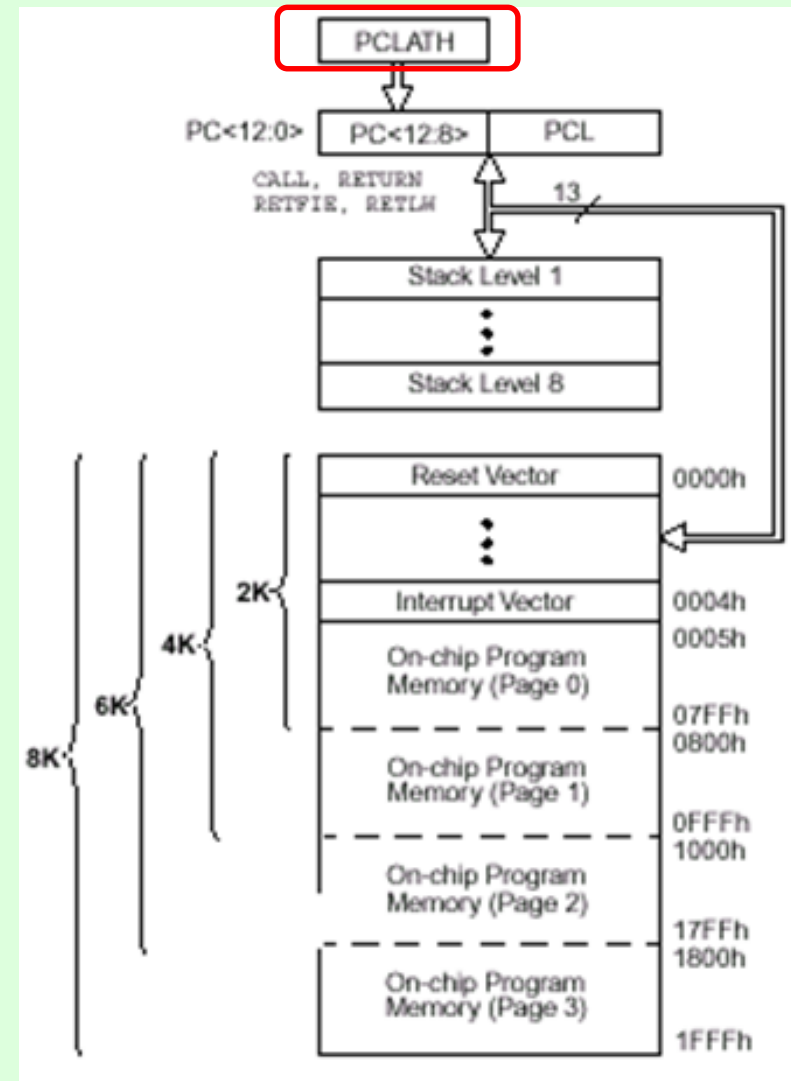
## Memoria de programa: paginado

- Los 8K de memoria de programa disponible están **divididos en 4 páginas de 2K cada una** (0h–7FFh, 800h–FFFh, 1000h–17FFh y 1800h–1FFFh). Motivo:

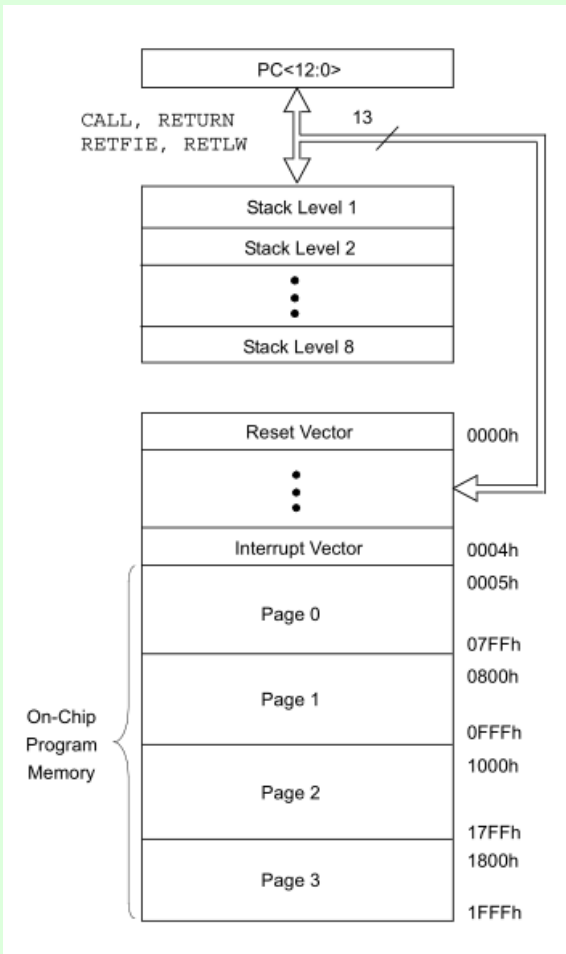
**Las instrucciones de salto y llamada a subprograma permiten cargar sólo 11 bits en el PC (desplazamiento en  $2^{11} = 2K$ )**

- Si se están ejecutando instrucciones secuencialmente, el contador de programa pasa de una página a otra sin necesidad de intervención por parte del usuario o programador.

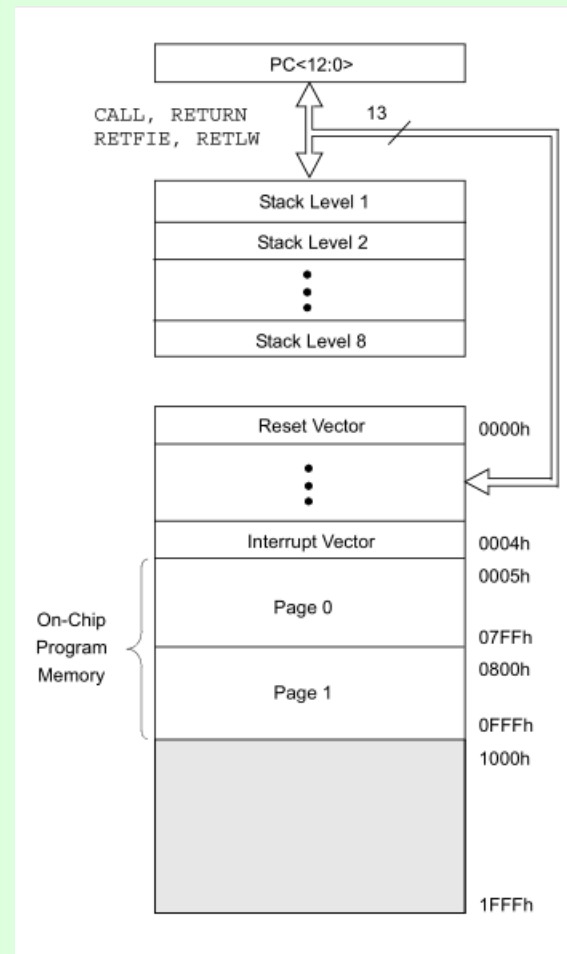
- Para saltar entre páginas de la memoria de programa los 2 bits más altos del PC deben modificarse. Esto se realiza escribiendo en el registro PCLATH (registro situado en la memoria de datos).



## Memoria de programa: paginado



MAPA DE MEMORIA DE  
PROGRAMA EN UN  
PIC16F876A/877A



MAPA DE MEMORIA DE  
PROGRAMA EN UN  
PIC16F873A/874A

- En algunos dispositivos solo una parte del mapa de memoria total posible está implementado.
- En la figura se muestra un dispositivo con 4K de memoria de programa y otro con 8K de memoria de programa.
- Los dispositivos con solo 2K de memoria de programa no necesitan paginado.



## Memoria de programa: vector de RESET

- El **vector de RESET** ocupa la **posición 0000h** de la memoria de programa.
- Cualquier reset (interno o externo) que se genere en un microcontrolador hará que su contador de programa pase a tener el valor 0000h y que por tanto el microcontrolador pase a ejecutar la instrucción situada en dicha posición.
- El **RESET también limpia** el contenido del registro **PCLATH**.

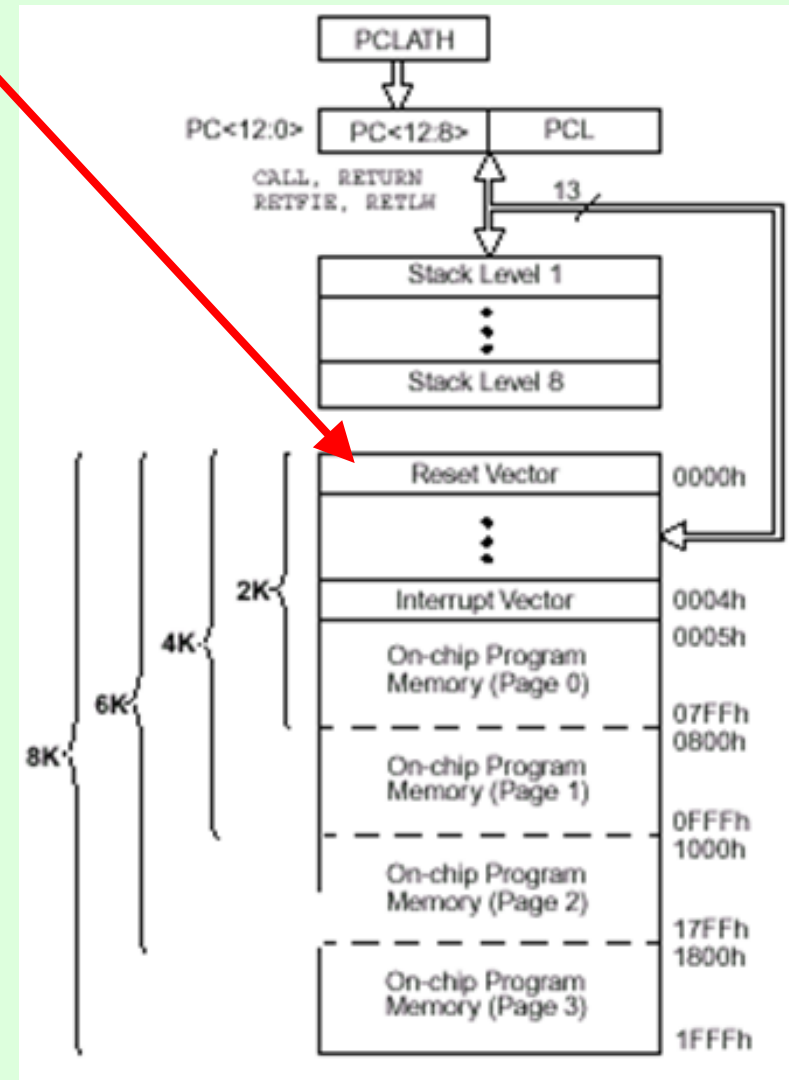
Ejemplo de programa:

```
ORG 0X00
GOTO INICIO
```

Coloca el código que sigue en 0X00 (Vector de RESET). La primera instrucción es un salto al inicio del programa

```
INICIO    ORG 0X05
          BSF STATUS,RP0
          MOVLW 0X00
          MOVWF TRISB
          ....
```

Aquí colocamos el programa que se ejecutará tras el RESET ....



## Memoria de programa: vector de interrupción

- El **vector de INTERRUPCION** ocupa la **posición 0004h** de la memoria de programa.
- Cualquier interrupción que se fuerce a un microcontrolador hará que su contador de programa pase a tener el valor 0004h y que por tanto el microcontrolador pase a ejecutar la instrucción situada en dicha posición.
- El **salto a la rutina de interrupción no modifica el contenido del PCLATH**, por lo que cualquier modificación que se realice del PC en la rutina de interrupción debe realizarse con cuidado de lo que se tiene cargado en el PCLATH

Ejemplo de programa:

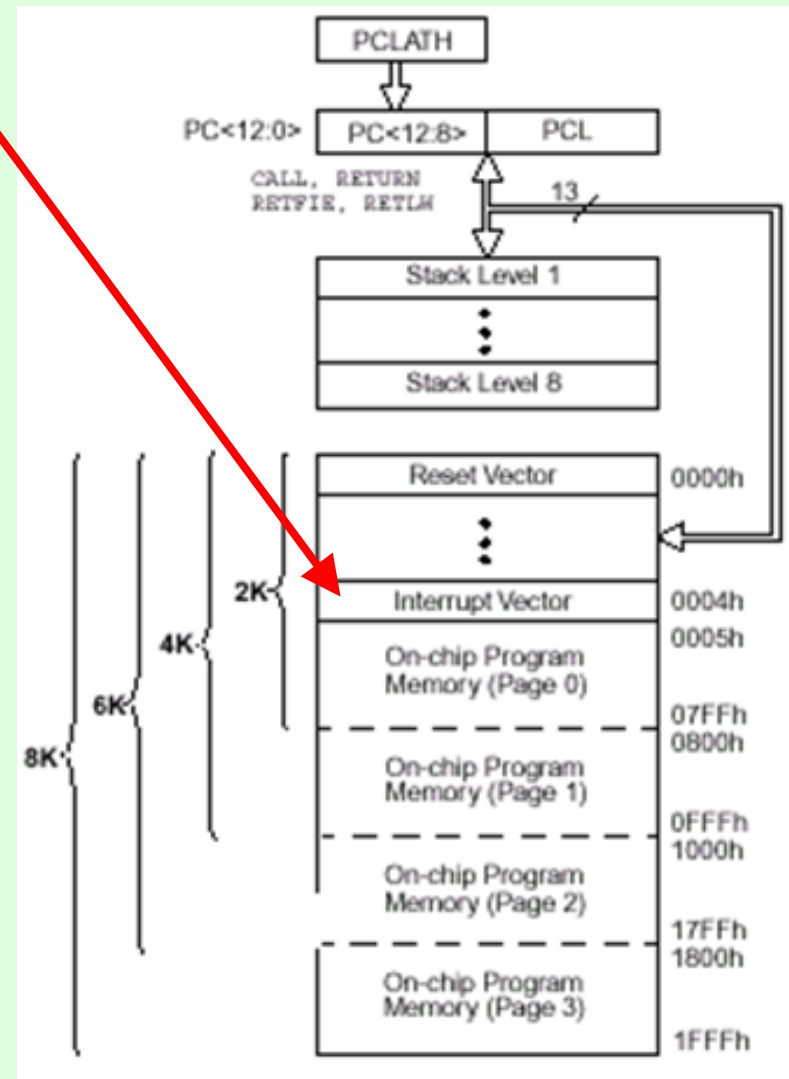
```
ORG 0X04
GOTO RSI
```

Coloca el código que sigue en 0X04 (Vector de Interrupción). Es un salto al inicio de la rutina de interrupción

```
RSI    ORG 0XF0
      BTFSC INTCON,RBIF
      GOTO RSI_PORTB
```

Aquí colocamos el programa de gestión de interrupción

....



## Memoria de programa: Información de calibración

- En algunos **dispositivos**, especialmente en aquellos que tienen la opción de utilizar como **oscilador una red RC interna**, viene grabada en la memoria de programa una **información de calibración**.
- Esta información es programada por *Microchip* cuando el dispositivo está en la fase final de test.
- Dicha información de calibración **está programada** habitualmente **al final de la memoria de programa** y está programada **como una instrucción RETLW**, donde el valor literal es la información de calibración.
- La utilización de este valor permite a la aplicación mejores resultados de precisión.

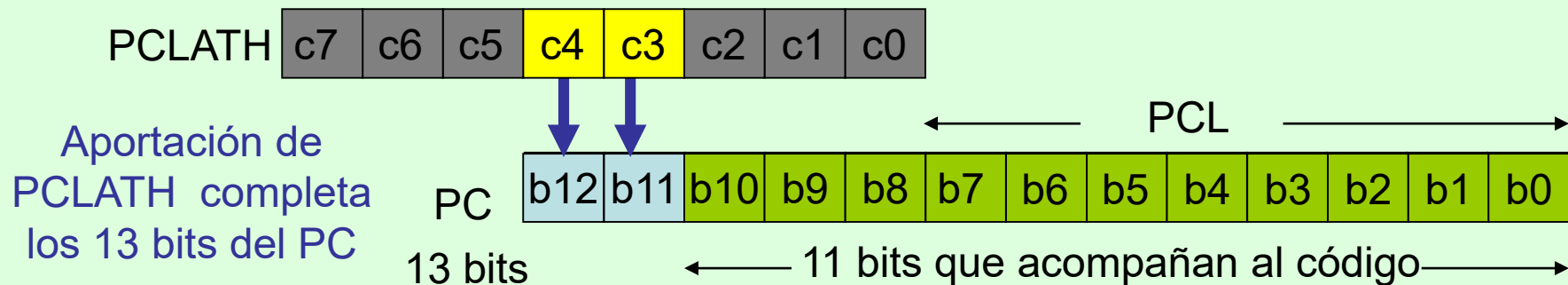
## Contador de programa (PC)

El **contador de programa (PC)** es un registro de 13 bits que se **descompone en 2 registros**:

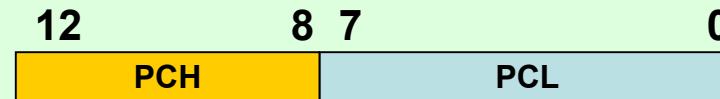


- El byte bajo **PC<0:7>** se denomina **PCL** y está disponible en la memoria de datos. Se puede leer y escribir directamente desde programa.
- El “cuasibyte” alto **PC<12:8>** se denomina **PCH** y no está disponible en la memoria de datos: no se puede leer ni escribir directamente desde programa, pero **se modifica indirectamente** (escritura) **mediante** un registro intermedio: **el registro PCLATH** que sí es un registro de 8 bits accesible en la memoria de datos del microcontrolador.
- El contenido del registro PCLATH permite completar el PC según la instrucción:
  - Se transfiere a la parte alta del PC (5 bits) al escribir en el registro PCL
  - También aporta dos bits al PC en los saltos o llamadas a subprogramas.

### Ejemplo: salto absoluto



## Modificación del Contador de programa (saltos en la ejecución)



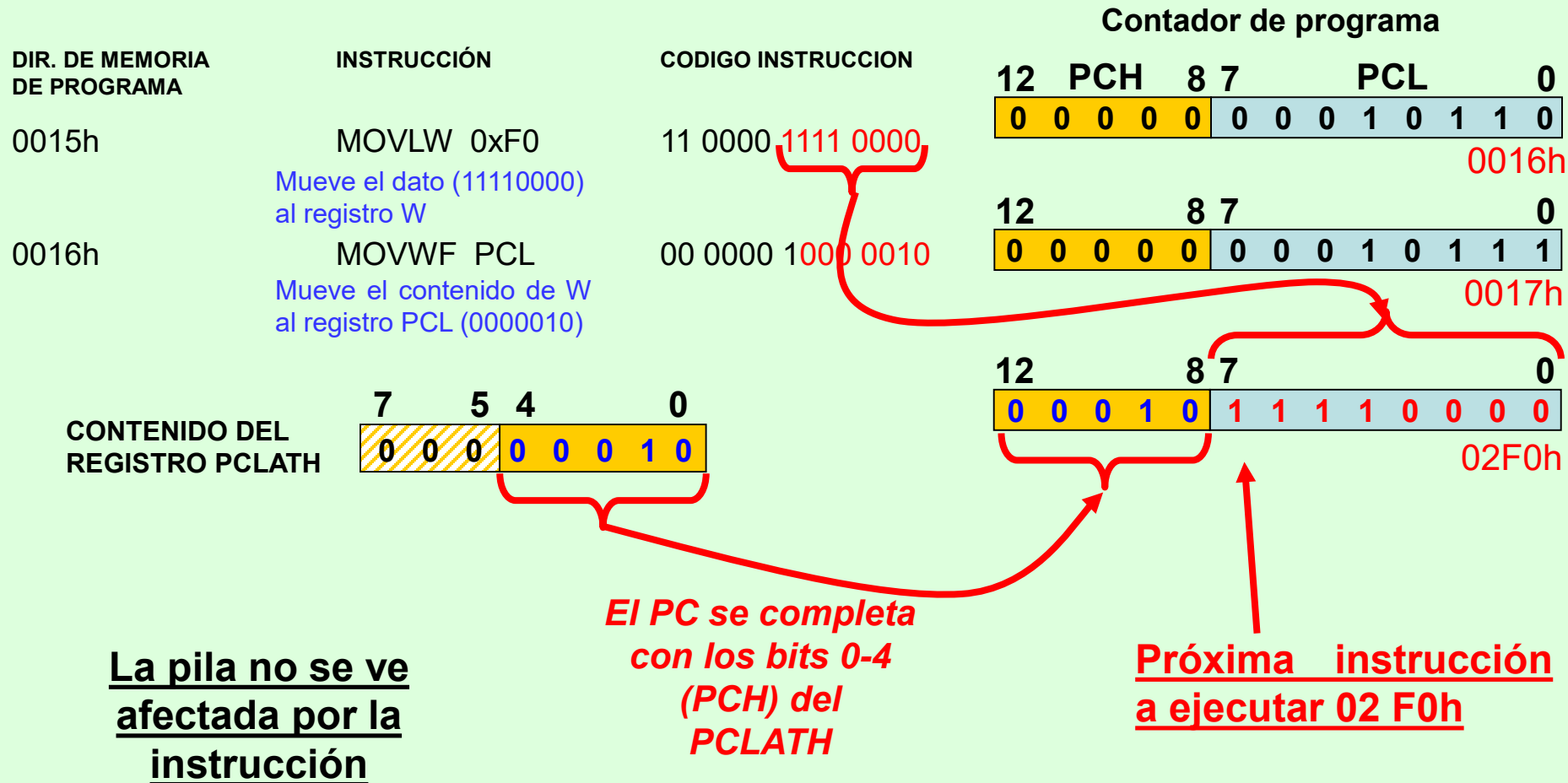
Las **instrucciones** que modifican el PCL y que por tanto pueden modificar el **PCH** son las siguientes:

- Instrucciones que tengan el PCL como destino. Ej. MOVWF PCL
- Instrucciones GOTO (Salto absoluto)
- Instrucciones CALL (Llamada de subrutina)

Dependiendo de la instrucción, la forma en que se carga el contador de programa (PC) varía

Al ejecutarse una instrucción, el PC contiene la siguiente dirección a ejecutar: si se modifica, se produce el salto

## Modificación del Contador de programa: Instrucciones que tengan el PCL como destino



## Modificación Instrucciones

DIR. DE MEMORIA  
DE PROGRAMA

0015h

0016h

INSTRUCCIÓN

MOVLW 0xF0

Mueve el dato (11110000)  
al registro W

MOVWF PCL

Mueve el contenido de W  
al registro PCL (00000000)

CONTENIDO DEL  
REGISTRO PCLATH

7	6	5	4
0	0	0	0

La pila no se ve  
afectada por la  
instrucción

File Address	File Address	File Address	File Address
Indirect addr. <sup>(1)</sup> 00h	Indirect addr. <sup>(1)</sup> 80h	Indirect addr. <sup>(1)</sup> 100h	Indirect addr. <sup>(1)</sup> 180h
TMR0 01h	OPTION_REG 81h	TMR0 101h	OPTION_REG 181h
PCL 02h	PCL 82h	PCL 102h	PCL 182h
STATUS 03h	STATUS 83h	STATUS 103h	STATUS 183h
FSR 04h	FSR 84h	FSR 104h	FSR 184h
PORTA 05h	TRISA 85h	PORTB 105h	TRISA 185h
PORTB 06h	TRISB 86h	PORTB 106h	TRISB 186h
PORTC 07h	TRISC 87h	PORTC 107h	TRISC 187h
PORTD <sup>(1)</sup> 08h	TRISD <sup>(1)</sup> 88h	PORTD 108h	TRISD 188h
PORTE <sup>(1)</sup> 09h	TRISE <sup>(1)</sup> 89h	PORTD 109h	TRISE 189h
PCLATH 0Ah	PCLATH 8Ah	PCLATH 10Ah	PCLATH 18Ah
INTCON 0Bh	INTCON 8Bh	INTCON 10Bh	INTCON 18Bh
PIR1 0Ch	PIE1 8Ch	EEDATA 10Ch	EECON1 18Ch
PIR2 0Dh	PIE2 8Dh	EEADR 10Dh	EECON2 18Dh
TMR1L 0Eh	PCON 8Eh	EEDATH 10Eh	Reserved <sup>(2)</sup> 18Eh
TMR1H 0Fh	8Fh	EEADRH 10Fh	Reserved <sup>(2)</sup> 18Fh
T1CON 10h	90h	110h	190h
TMR2 11h	SSPCON2 91h	111h	191h
T2CON 12h	PR2 92h	112h	192h
SSPBUF 13h	SSPAD 93h	113h	193h
SSPCON 14h	SSPSTAT 94h	114h	194h
CCPR1L 15h	95h	115h	195h
CCPR1H 16h	96h	116h	196h
CCP1CON 17h	97h	117h	197h
RCSTA 18h	TXSTA 98h	118h	198h
TXREG 19h	SPBRG 99h	119h	199h
RCREG 1Ah	9Ah	11Ah	19Ah
CCPR2L 1Bh	9Bh	11Bh	19Bh
CCPR2H 1Ch	CMCON 9Ch	11Ch	19Ch
CCP2CON 1Dh	CVRCON 9Dh	11Dh	19Dh
ADRESH 1Eh	ADRESL 9Eh	11Eh	19Eh
ADCON0 1Fh	ADCON1 9Fh	11Fh	19Fh
20h	A0h	120h	1A0h
General Purpose Register 96 Bytes	General Purpose Register 80 Bytes	General Purpose Register 80 Bytes	General Purpose Register 80 Bytes
7Fh	EFh	16Fh	1EFh
Bank 0	accesses 70h-7Fh F0h	accesses 70h-7Fh 170h	accesses 70h-7Fh 1F0h
	FFh	17Fh	1FFh
Bank 1		Bank 2	Bank 3

0

1 0

0016h

0

1 1

0017h

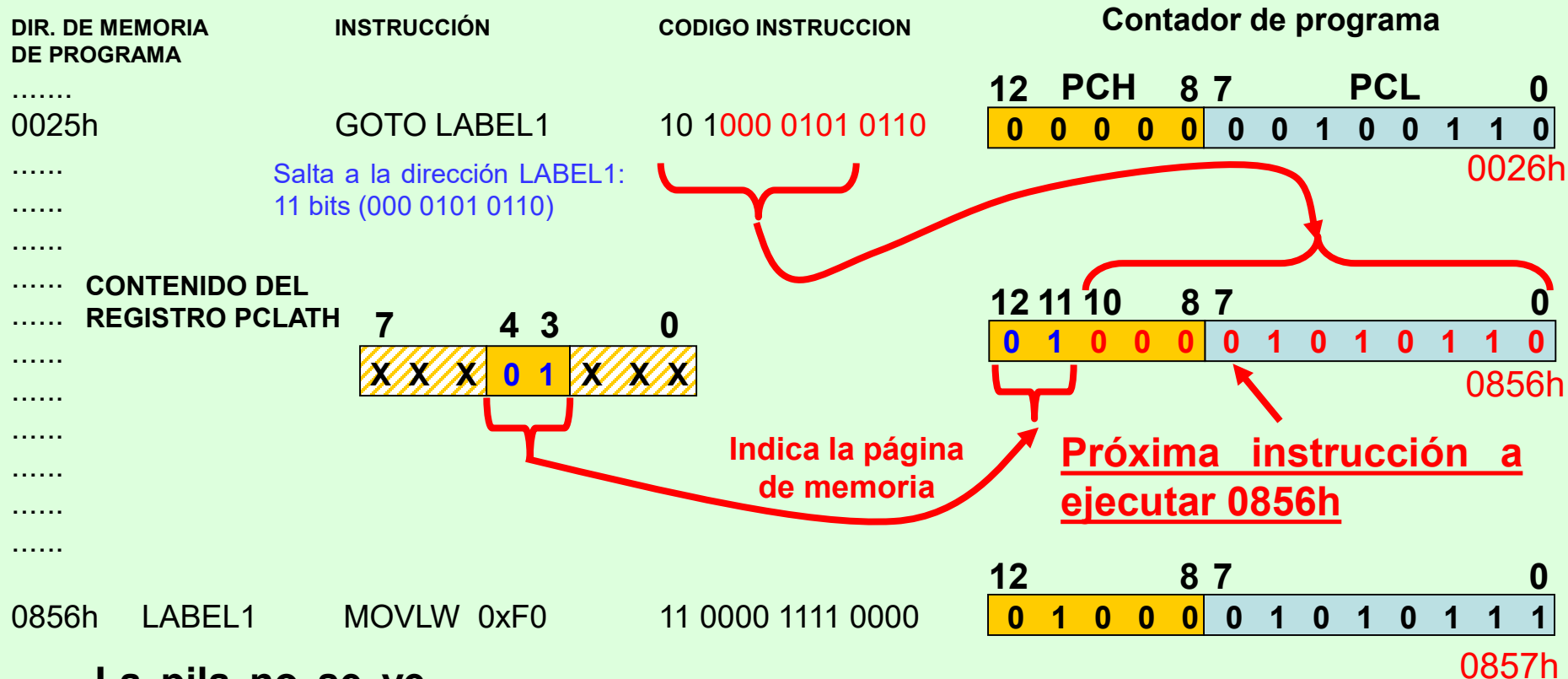
0

0 0

02F0h

cción

## Modificación del Contador de programa: Instrucción GOTO



La pila no se ve  
afectada por la  
instrucción



**Pila: se guarda el PC**

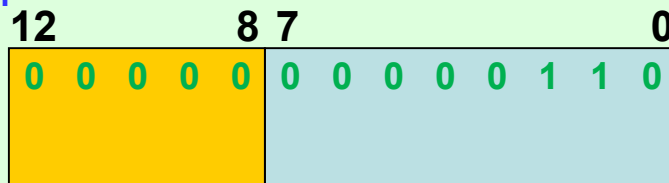
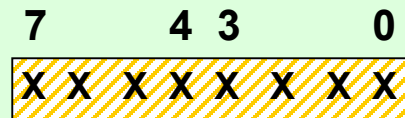


## Modificación del Contador de programa: Instrucciones de retorno: RETURN, RETFIE ó RETLW

DIR. DE MEMORIA DE PROGRAMA		INSTRUCCIÓN	CODIGO INSTRUCCION
0004h		.....	
0005h		CALL SUBRUT1	10 0000 0000 0000
.....			
.....			
1000h	SUBRUT1	MOVLW 0xF0	11 0000 1111 0000
.....			
1100h		RETURN	00 0000 0000 1000

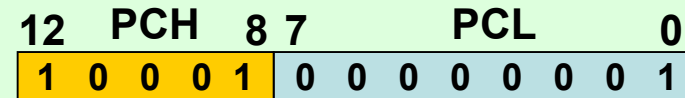
Retorno de subrutina:  
se carga PC de la pila

LA DIRECCIÓN DE  
RETORNO DE SUBROUTINA  
NO DEPENDE DEL  
CONTENIDO DEL REGISTRO  
PCLATH



La pila tiene guardada la posición  
para retornar tras la subrutina

Contador de programa



1101h



0006h

Próxima instrucción a  
ejecutar 0006h

## Registros de funciones especiales asociados al núcleo del microcontrolador: Registro STATUS

### STATUS REGISTER (ADDRESS 03h, 83h, 103h, 183h)

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C
bit 7			bit 0				

#### Legend:

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

bit 7 **IRP**: Register Bank Select bit (used for indirect addressing)

1 = Bank 2, 3 (100h - 1FFh)

0 = Bank 0, 1 (00h - FFh)

bit 6-5 **RP1:RP0**: Register Bank Select bits (used for direct addressing)

11 = Bank 3 (180h - 1FFh)

10 = Bank 2 (100h - 17Fh)

01 = Bank 1 (80h - FFh)

00 = Bank 0 (00h - 7Fh)

Each bank is 128 bytes

bit 4 **TO**: Time-out bit

1 = After power-up, CLRWDI instruction, or SLEEP instruction

0 = A WDT time-out occurred

bit 3 **PD**: Power-down bit

1 = After power-up or by the CLRWDI instruction

0 = By execution of the SLEEP instruction

bit 2 **Z**: Zero bit

1 = The result of an arithmetic or logic operation is zero

0 = The result of an arithmetic or logic operation is not zero

bit 1 **DC**: Digit carry/borrow bit (ADDWF, ADDLW, SUBLW, SUBWF instructions)

(for borrow, the polarity is reversed)

1 = A carry-out from the 4th low order bit of the result occurred

0 = No carry-out from the 4th low order bit of the result

bit 0 **C**: Carry/borrow bit (ADDWF, ADDLW, SUBLW, SUBWF instructions)

1 = A carry-out from the Most Significant bit of the result occurred

0 = No carry-out from the Most Significant bit of the result occurred

Note: For borrow, the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand. For rotate (RRF, RLF) instructions, this bit is loaded with either the high, or low order bit of the source register.

## Registros de funciones especiales asociados al nucleo del microcontrolador: Registro OPTION (OPTION REG)

### OPTION\_REG REGISTER (ADDRESS 81h, 181h)

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

#### Legend:

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

bit 7 **RBP $\overline{\text{U}}$** : PORTB Pull-up Enable bit

1 = **PORTB** pull-ups are disabled

0 = **PORTB** pull-ups are enabled by individual port latch values

bit 6 **INTEDG**: Interrupt Edge Select bit

1 = Interrupt on rising edge of RB0/INT pin

0 = Interrupt on falling edge of RB0/INT pin

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

1 = Transition on RA4/T0CKI pin

0 = Internal instruction cycle clock (CLKOUT)

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

1 = Increment on high-to-low transition on RA4/T0CKI pin

0 = Increment on low-to-high transition on RA4/T0CKI pin

bit 3 **PSA**: Prescaler Assignment bit

1 = Prescaler is assigned to the WDT

0 = Prescaler is assigned to the Timer0 module

bit 2-0 **PS2:PS0**: Prescaler Rate Select bits

Bit Value	TMR0 Rate	WDT Rate
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

## Registros de funciones especiales asociados al nucleo del microcontrolador: Registro INTCON

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

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

#### Legend:

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

bit 7 **GIE**: Global Interrupt Enable bit

1 = Enables all unmasked interrupts

0 = Disables all interrupts

bit 6 **PEIE**: Peripheral Interrupt Enable bit

1 = Enables all unmasked peripheral interrupts

0 = Disables all peripheral interrupts

bit 5 **T0IE**: TMR0 Overflow Interrupt Enable bit

1 = Enables the TMR0 interrupt

0 = Disables the TMR0 interrupt

bit 4 **INTE**: RB0/INT External Interrupt Enable bit

1 = Enables the RB0/INT external interrupt

0 = Disables the RB0/INT external interrupt

bit 3 **RBIE**: RB Port Change Interrupt Enable bit

1 = Enables the RB port change interrupt

0 = Disables the RB port change interrupt

bit 2 **T0IF**: TMR0 Overflow Interrupt Flag bit

1 = TMR0 register has overflowed (must be cleared in software)

0 = TMR0 register did not overflow

bit 1 **INTF**: RB0/INT External Interrupt Flag bit

1 = The RB0/INT external interrupt occurred (must be cleared in software)

0 = The RB0/INT external interrupt did not occur

bit 0 **RBIF**: RB Port Change Interrupt Flag bit

1 = At least one of the RB7:RB4 pins changed state; a mismatch condition will continue to set the bit. Reading PORTB will end the mismatch condition and allow the bit to be cleared (must be cleared in software).

0 = None of the RB7:RB4 pins have changed state

## Registros de funciones especiales asociados al nucleo del microcontrolador: Registro PIE1

**PIE1 REGISTER (ADDRESS 8Ch)**

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 <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
bit 7				bit 0			

**Legend:**

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

bit 7 **PSPIE(1)**: Parallel Slave Port Read/Write Interrupt Enable bit

1 = Enables the PSP read/write interrupt  
0 = Disables the PSP read/write interrupt

bit 6 **ADIE**: A/D Converter Interrupt Enable bit

1 = Enables the A/D converter interrupt  
0 = Disables the A/D converter interrupt

bit 5 **RCIE**: USART Receive Interrupt Enable bit

1 = Enables the USART receive interrupt  
0 = Disables the USART receive interrupt

bit 4 **TXIE**: USART Transmit Interrupt Enable bit

1 = Enables the USART transmit interrupt  
0 = Disables the USART transmit interrupt

bit 3 **SSPIE**: Synchronous Serial Port Interrupt Enable bit

1 = Enables the SSP interrupt  
0 = Disables the SSP interrupt

bit 2 **CCP1IE**: CCP1 Interrupt Enable bit

1 = Enables the CCP1 interrupt  
0 = Disables the CCP1 interrupt

bit 1 **TMR2IE**: TMR2 to PR2 Match Interrupt Enable bit

1 = Enables the TMR2 to PR2 match interrupt  
0 = Disables the TMR2 to PR2 match interrupt

bit 0 **TMR1IE**: TMR1 Overflow Interrupt Enable bit

1 = Enables the TMR1 overflow interrupt  
0 = Disables the TMR1 overflow interrupt

**Note 1:** PSPIE is reserved on PIC16F873/876 devices; always maintain this bit clear.

## Registros de funciones especiales asociados al nucleo del microcontrolador: Registro PIR1

### PIR1 REGISTER (ADDRESS 0Ch)

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

#### Legend:

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

bit 7 **PSPIF(1)**: Parallel Slave Port Read/Write

Interrupt Flag bit

1 = A read or a write operation has taken place (must be cleared in software)

0 = No read or write has occurred

bit 6 **ADIF**: A/D Converter Interrupt Flag bit

1 = An A/D conversion completed

0 = The A/D conversion is not complete

bit 5 **RCIF**: USART Receive Interrupt Flag bit

1 = The USART receive buffer is full

0 = The USART receive buffer is empty

bit 4 **TXIF**: USART Transmit Interrupt Flag bit

1 = The USART transmit buffer is empty

0 = The USART transmit buffer is full

bit 3 **SSPIF**: Synchronous Serial Port (SSP) Interrupt Flag

1 = The SSP interrupt condition has occurred, and must be cleared in software before returning from the Interrupt Service Routine. The conditions that will set this bit are:

- SPI - A transmission/reception has taken place.
- I2C Slave - A transmission/reception has taken place.
- I2C Master
  - A transmission/reception has taken place.
  - The initiated START condition was completed by the SSP module.
  - The initiated STOP condition was completed by the SSP module.
  - The initiated Restart condition was completed by the SSP module.
  - The initiated Acknowledge condition was completed by the SSP module.
  - A START condition occurred while the SSP module was idle (Multi-Master system).
  - A STOP condition occurred while the SSP module was idle (Multi-Master system).
- 0 = No SSP interrupt condition has occurred.

## Registros de funciones especiales asociados al nucleo del microcontrolador: Registro PIR1 (continuación)

### PIR1 REGISTER (ADDRESS 0Ch)

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

#### Legend:

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

bit 2 **CCP1IF**: CCP1 Interrupt Flag bit

- Capture mode:

1 = A TMR1 register capture occurred (must be cleared in software)

0 = No TMR1 register capture occurred

- Compare mode:

1 = A TMR1 register compare match occurred (must be cleared in software)

0 = No TMR1 register compare match occurred

- PWM mode:

Unused in this mode

bit 1 **TMR2IF**: TMR2 to PR2 Match Interrupt Flag bit

1 = TMR2 to PR2 match occurred (must be cleared in software)

0 = No TMR2 to PR2 match occurred

bit 0 **TMR1IF**: TMR1 Overflow Interrupt Flag bit

1 = TMR1 register overflowed (must be cleared in software)

0 = TMR1 register did not overflow

**Note 1:** PSPIF is reserved on PIC16F873/876 devices; always maintain this bit clear.



## Registros de funciones especiales asociados al nucleo del microcontrolador: Registro PIE2

**PIE2 REGISTER (ADDRESS 8Dh)**

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

**Legend:**

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

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

bit 6 **CMIE:** Comparator Interrupt Enable bit

1 = Enables the Comparator interrupt

0 = Disable the Comparator interrupt

bit 5 **Unimplemented:** Read as '0'

bit 4 **EEIE:** EEPROM Write Operation Interrupt Enable

1 = Enable EE Write Interrupt

0 = Disable EE Write Interrupt

bit 3 **BCLIE:** Bus Collision Interrupt Enable

1 = Enable Bus Collision Interrupt

0 = Disable Bus Collision Interrupt

bit 2-1 **Unimplemented:** Read as '0'

bit 0 **CCP2IE:** CCP2 Interrupt Enable bit

1 = Enables the CCP2 interrupt

0 = Disables the CCP2 interrupt

## Registros de funciones especiales asociados al nucleo del microcontrolador: Registro PIR2

PIR2 REGISTER (ADDRESS 0Dh)

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

**Legend:**

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

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

bit 6 **CMIF:** Comparator Interrupt Flag bit

1 = The Comparator input has changed (must be cleared in software)

0 = The Comparator input has not changed

bit 5 **Unimplemented:** Read as '0'

bit 4 **EEIF:** EEPROM Write Operation Interrupt Flag bit

1 = The write operation completed (must be cleared in software)

0 = The write operation is not complete or has not been started

bit 3 **BCLIF:** Bus Collision Interrupt Flag bit

1 = A bus collision has occurred in the SSP, when configured for I2C

Master mode

0 = No bus collision has occurred

bit 2-1 **Unimplemented:** Read as '0'

bit 0 **CCP2IF:** CCP2 Interrupt Flag bit

Capture mode:

1 = A TMR1 register capture occurred (must be cleared in software)

0 = No TMR1 register capture occurred

Compare mode:

1 = A TMR1 register compare match occurred (must be cleared in software)

0 = No TMR1 register compare match occurred

PWM mode:

Unused

## Registros de funciones especiales asociados al nucleo del microcontrolador: Registro PCON

### PCON REGISTER (ADDRESS 8Eh)

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-1
—	—	—	—	—	—	POR	BOR
bit 7						bit 0	

#### Legend:

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

bit 7-2 **Unimplemented:** Read as '0'

bit 1 **POR:** Power-on Reset Status bit

1 = No Power-on Reset occurred

0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)

bit 0 **BOR:** Brown-out Reset Status bit

1 = No Brown-out Reset occurred

0 = A Brown-out Reset occurred (must be set in software after a Brown-out Reset occurs)