

## **UNIDAD III**

### **ORGANIZACION DE LA MEMORIA**

El PIC16F84 tiene dos bloques de memoria separados, uno de datos y el otro de programa. La memoria EEPROM y los registros GPR (registros de propósito general) localizados en la memoria RAM constituyen el bloque de datos, mientras que la memoria FLASH es el bloque de programa.

#### **III.1. Memoria de programa**

La memoria de programa está hecha con tecnología FLASH, lo cual permite programar al microcontrolador una gran cantidad de veces antes de instalarlo en un dispositivo y aún después de haberlo instalado si ocurren cambios eventuales en el programa o parámetros de procesos. El tamaño de la memoria de programa es de 1024 localidades de 14 bits cada una, donde las localidades cero y cuatro están reservadas para el reset y el vector de interrupción, respectivamente.

#### **III.2. Memoria de datos**

La memoria de datos está compuesta de la memoria EEPROM y la memoria RAM. La memoria EEPROM consiste de 64 localidades de 8 bits cada una cuyo contenido no se pierde al apagar la fuente de alimentación. La memoria EEPROM no es direccionable directamente sino que se accesa por medio de los registros EEADR y EEDATA. Ya que la memoria EEPROM sirve para almacenar parámetros importantes (por ejemplo, la temperatura establecida por un regulador), existe un

procedimiento estricto para escribir en la memoria EEPROM, el cual debe ser seguido para evitar escrituras erróneas o accidentales.

La memoria RAM ocupa espacio dentro del mapa de memoria de datos usando de la localidad 0x0C a la 0x4F (68 localidades). Las localidades de la memoria RAM se llaman también registros GPR lo cual es una abreviación de General Purpose Registers. Los registros GPR se pueden acceder sin importar el banco seleccionado en ese momento.

### **III.3. Registros**

#### **Registros SFR**

Los registros SFR (Special Function Registers-registros de funciones especiales) ocupan las primeras 12 localidades en los bancos 0 y 1 de la memoria de datos.

### **III.4. Tipos de memoria**

En la siguiente figura se muestra a bloques el mapa de memoria y los diferentes tipos de memoria con los que cuenta el PIC16F84.

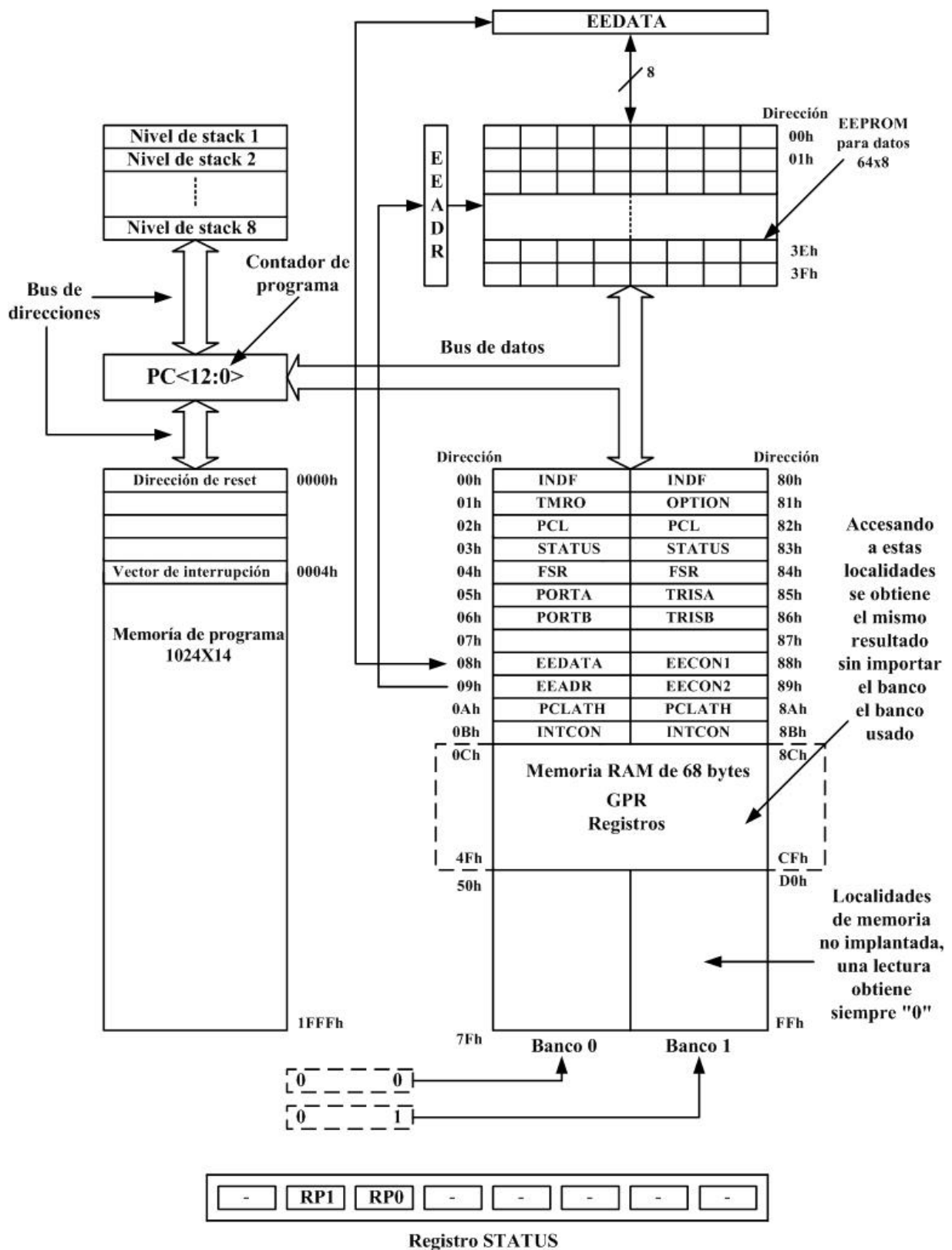


Figura III.1. Organización de memoria del PIC16F84

## **Bancos de memoria**

En el espacio de memoria se observa una línea vertical y una línea horizontal en la memoria de datos. La línea horizontal divide los registros SFR de los registros GPR. La línea vertical separa dos áreas llamadas bancos. La selección de uno de los dos bancos se hace por medio de los bits RP0 y RP1 del registro STATUS.

Ejemplo:

```
bcf STATUS,RP0
```

La instrucción bcf limpia el bit RP0 (RP0=0) del registro STATUS seleccionando el banco 0.

```
bsf STATUS,RP0
```

La instrucción bsf establece a 1 lógico el bit RP0 (RP0=1) del registro STATUS seleccionando el banco 1.

Usualmente, se agrupan las instrucciones más comúnmente usadas en unidades llamadas macros y que fácilmente se pueden llamar o invocar en un programa, donde el nombre de esos grupos tiene un significado claro y relacionado con la función del grupo. Por ejemplo, se puede escribir un macro para seleccionar uno de los dos bancos y hacer el programa más legible.

```
BANK0    macro
          bcf  STATUS, RP0    ;Selecciona el banco 0
          endm
```

```
BANK1    macro
          bsf  STATUS, RP0    ;Selecciona el banco 1
          endm
```

Como ya se dijo anteriormente, las localidades 0Ch a la 4Fh son los registros de propósito general (GPR) los cuales pueden ser usados como memoria RAM. Cuando las localidades 8Ch a la CFh del banco 1 son accesadas, se estarán accedando exactamente las mismas localidades del banco 0. En otras palabras, si se desea acceder uno de los registros GPR, no importará el banco que esté seleccionado o en el cual se esté posicionado.

## **Program counter**

El program counter (contador de programa-PC) es un registro de 13 bits que contiene la dirección de la siguiente instrucción a ejecutar. Este registro se incrementa o cambia (por ejemplo en caso de una instrucción de salto) cuando el microcontrolador ejecuta las instrucciones del programa paso-a-paso.

## **Stack**

El PIC16F84 tiene un stack de 13 bits de 8 niveles, o en otras palabras, es un grupo de 8 localidades de memoria de 13 bits cada una. El papel del stack es almacenar el valor del program counter después de un salto en el programa principal a la dirección de un subprograma (por ejemplo con una instrucción CALL) y poder retornar después al punto donde se interrumpió el programa principal (por ejemplo ejecutando las instrucciones RETURN, RETLW o RETFIE). Las operaciones de colocar y tomar del stack el program counter se llaman PUSH y POP, respectivamente y como tal no están implantadas como instrucciones en el PIC16F84.