

# Capítulo 1. Introducción a los PIC de 8 bits

## 1.1 Memorias y Registros

Entre los componentes básicos de un microcontrolador podemos definir el contador, sus registros, la memoria, el watchdog timer, el reset y los pines I/O. En los microcontroladores Microchip podemos encontrar que existen cuatro memorias diferentes, la memoria del programa, memoria SRAM, la memoria ROM y la memoria EEPROM. Como se puede deducir estos emplean una arquitectura Harvard. A continuación, se lista las memorias empleadas en estos uC y sus usos.

**Memoria de ROM:** se encarga de decodificar las instrucciones y generar las señales de control de manera que las instrucciones que se encuentran en la memoria del programa se ejecuten de manera correcta.

**Memoria del programa:** Es donde se almacenan las instrucciones a realizar por lo que debe ser de un tamaño lo suficientemente grande para escribir las rutinas deseadas. Actualmente esta memoria es implementada con memoria tipo flash. Esta memoria viene paginada, por lo tanto, hay que ubicarse en la página correspondiente para realizar saltos, el ancho de palabra de esta memoria depende de los bits necesarios para realizar las instrucciones requeridas, usualmente es de 12,14,16 bits.

**Memoria SRAM:** Este tipo de memoria se utiliza para almacenar algunas variables que se

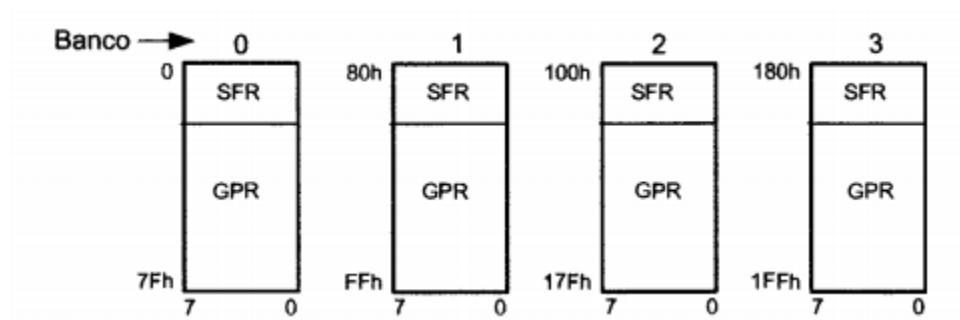
usan en el programa y es donde se encuentran los registros.

**Memoria EEPROM:** Este tipo de memoria es utilizada más que nada para almacenar datos que son requeridos para la inicialización de una rutina, valores críticos o constantes requeridas en el diseño del código.

En cuanto a registros los uC de Microchip cuentan con dos tipos diferentes, GPR (General Purpose Registers) y SFR (Special Function Registers) y estos son generalmente agrupados por medio de bancos, por lo tanto, si se quiere escribir en un registro uno tiene que ubicarse en el banco correcto y utilizar el método de direccionamiento apropiado. Estos microcontroladores cuentan con un registro W, el cual es utilizado como registro intermedio de trabajo, con él se pueden extraer datos de un registro y moverlos a otros, es el único registro al que se puede acceder de manera directa.

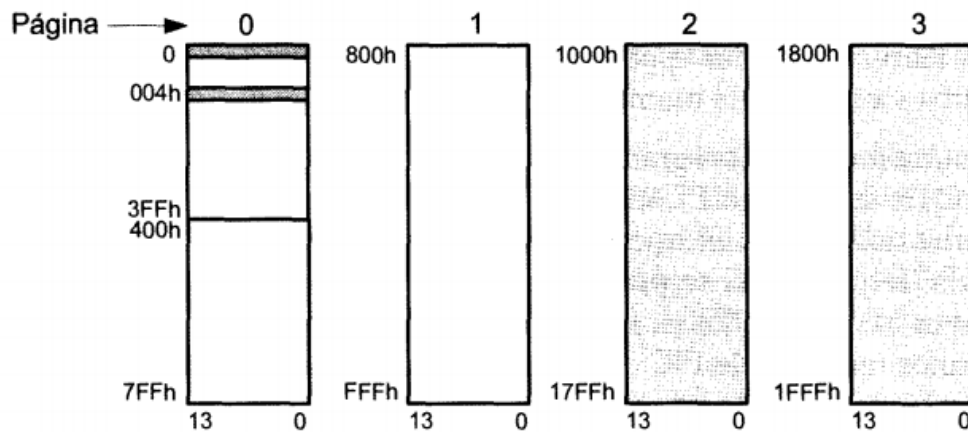
Los SFR son utilizados generalmente para observar el estado del integrado y para la configuración del microcontrolador, en especial de los módulos con los que cuenta, mientras que los GPR sirven para almacenar las variables utilizadas en el programa.

A continuación, se muestra la siguiente figura que muestra la disposición general de los registros en los PIC.



**Figura 1.3** Distribución de registros [5]

Mientras que la distribución de la memoria viene dada de la siguiente manera:



**Figura 1.4** Distribución de memoria del programa [5]

Estos microcontroladores cuentan con registros que son accesibles desde cualquier banco, registros de uso muy particular como STATUS o registros GPR que son utilizados para el resguardo de valores en el caso de saltar a una ISR o función.

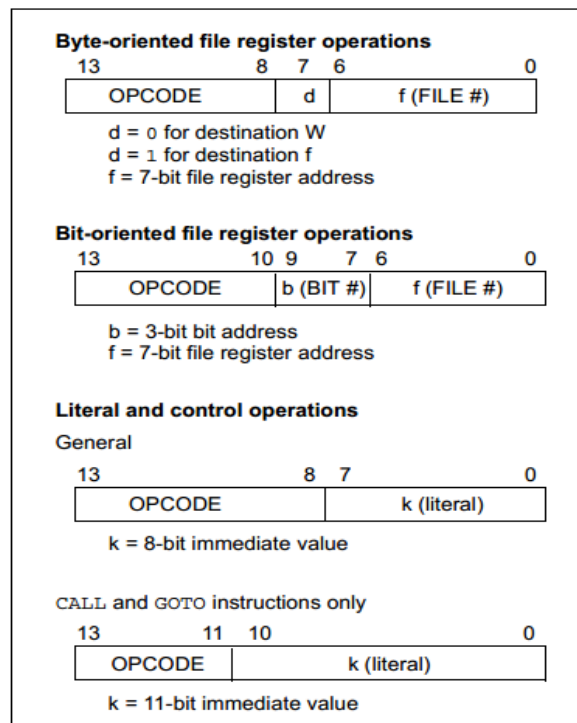
En cuanto a las interrupciones estos uC cuentan con dos vectores de interrupción, el de reset y el vector de interrupción, esto es malo de alguna manera, pues hay que deshabilitar las interrupciones una vez que se atiende una ellas, además de sobrecargar la rutina de

atención a interrupción cuando se tienen configuradas varias fuentes de interrupción.

## 1.2 Manejo de pila y formato de instrucciones

Con respecto a la pila o stack podemos decir que es una estructura en hardware que almacena las direcciones de retorno cada vez que se llama a una función o entra una interrupción. Esta pila tiene una profundidad de 8 niveles, por lo que hay que tomar cuidado cuando se programa en lenguajes de alto nivel, pues uno no sabe cómo el compilador implementa las funciones recursivas.

Los PIC de 8 bits de rango medio cuentan con un conjunto de 35 instrucciones, dichas instrucciones se encuentran en el Apéndice A. A continuación, se presenta el formato que siguen las instrucciones.



**Figura 1.5** Formato de instrucciones PIC 8 bits [6]

Como se puede observar, en las instrucciones con manejo de literales se manejan datos de 8 bits. Por otro lado, podemos ver por qué los registros están agrupados en bancos, esto es por el hecho de contar con siete bits para elegir el registro en las instrucciones. Asimismo, se puede apreciar el manejo de las instrucciones call y goto que manejan una literal de 11 bits para realizar saltos de memoria, por lo tanto, los bits restantes del registro PC para el salto se modifican a través de PCLATH. Para retirar un dato de la pila basta con utilizar las instrucciones RETURN, RETFIE y RETLW.

### **1.3 Ciclos de instrucción y diagrama a bloques**

Los microcontroladores Microchip cuentan con un pipeline de 2 niveles, una para fetching y otra de ejecución, esta es una manera de obtener un mayor *throughput*, es decir, incrementar el número de instrucciones por ciclo de reloj, esto se debe a que en un nivel buscan la instrucción y en el otro realizan su ejecución.

Hay que destacar que se realiza una instrucción por cada cuatro ciclos de reloj, esto es debido a que necesitan 4 ciclos de reloj para generar las señalizaciones de control requeridas. A continuación, se presenta una imagen extraída del datasheet del PIC16FXK50 que muestra lo ya explicado.

FIGURE 5-3: CLOCK/INSTRUCTION CYCLE

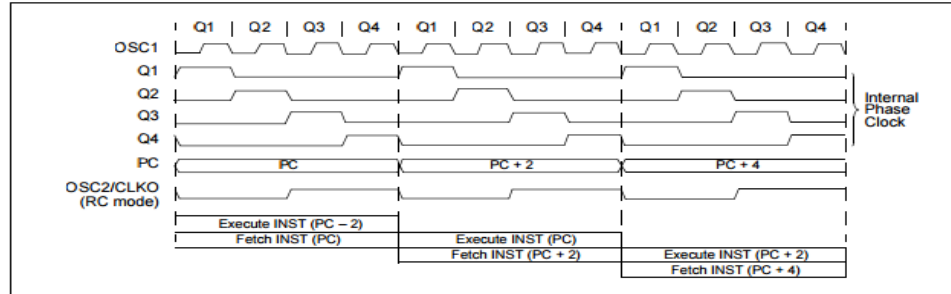


Figura 1.6 Esquema de tiempos [7]

Como se puede observar en la figura anterior, el registro PC apunta a la instrucción que se intenta conseguir, no a la instrucción que está ejecutándose. En la siguiente figura se puede observar el diagrama a bloques de un PIC de gama media.

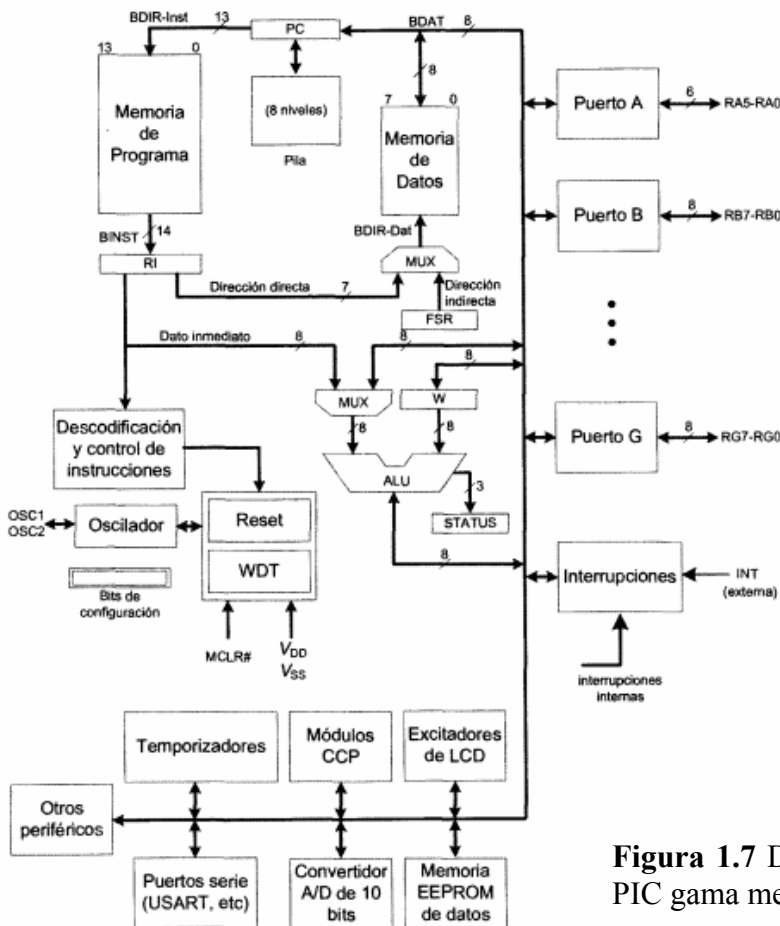


Figura 1.7 Diagrama a bloques de un PIC gama media [5]

Como se puede observar con lo anteriormente explicado, la programación de estos microcontroladores resulta un poco difícil si se requiere hacer programas extensos, esto es debido a la paginación de la memoria, a los bancos de registros e implementación de las rutinas de interrupción, además uno debe de entender los diferentes modos de direccionamiento y entender en general la organización del uC. Por otro lado, esto ya no es un problema debido al uso de lenguajes de alto nivel como C o C++, pues el compilador es el que se encarga de convertir todo el código escrito en C a ensamblador y de ahí a lenguaje máquina específico para cada microcontrolador.

El uso del lenguaje ensamblador da al programador la absoluta libertad de elegir como programar las rutinas a utilizar, sin embargo, uno debe conocer cómo se maneja el microcontrolador de manera minuciosa, lo cual lleva tiempo, por otro lado, los lenguajes de alto nivel hacen que el código sea “portable” y que el compilador se encargue de traducir el código realizado en instrucciones que el uC pueda llegar a entender.

## **1.4 Conociendo el PIC16F877A**

Habiendo dicho todo lo anterior en cuanto a cómo están constituidos internamente estos microcontroladores lo único que queda es fijarnos en los módulos o periféricos que trae consigo el integrado de nuestro interés. En nuestro caso pondremos de referencia el PIC16F877A, pues es el que se emplea actualmente en algunas escuelas.

El PIC16F877A es un integrado de 40 pines que cuenta con las siguientes características según [6].

- Frecuencia de oscilador máxima de 20 MHz
- 8K de Memoria de Programa
- EEPROM 256 Bytes
- SRAM 368 Bytes
- Watch Dog Timer
- POR, BOR (Power On Reset Timer, Brown on Reset)
- Timer0 de 8 bits con pre escalador de 8 bits
- Timer1 de 16 bits con pre escalador
- Timer2 de 8bits con pre escalador y post-escalador
- 5 Puertos I/O. PORTA, PORTB, PORTC, PORTD, PORTE.
- PSP Puerto Paralelo-Esclavo
- 2 Módulos CCP - Capture/Compare/PWM
- 1 Modulo USART (Universal Synchronous Asynchronous Receiver Transmitter)
- Modulo Comparador con dos comparadores multiplexados
- 1 Modulo A/D de 7 canales de 10 bits

Como se puede observar, el microcontrolador es muy completo en general. Sin embargo, puede quedar corto para ciertas aplicaciones, por ejemplo, el manejo de un quadricoptero, pues se requiere de 4 o 5 señales PWM. El PIC16F877A es un microcontrolador “clásico” por decirlo de alguna manera, pues ya lleva un tiempo en el mercado. Por otro lado, podemos enlistar el PIC16LF1789, este microcontrolador de 40 pines es más reciente, según su datasheet [8] salió a la venta en el 2013.



Entre las características más importantes de este microcontrolador podemos citar las siguientes:

- Oscilador interno de 32 MHz
- Juego de 49 instrucciones
- Stack de 16 niveles
- 5 Puertos I/O
- 8 kWords de memoria flash
- 4 PSMC (Controlador programable en modo conmutable), PWM dedicado de 16 bits
- 3 Módulos CCP
- ADC de 12 bits con 14 canales simples y 7 canales diferenciales
- 3 OPAM
- 4 Comparadores
- 1 DAC de 8bits y 3 DAC de 5 bits

Como se puede observar este integrado viene muy completo, mejor que el anterior, por ejemplo, podría utilizarse para generar hasta 7 señales PWM. Sin embargo, puede que quede corto para ciertas aplicaciones donde se necesitan realizar algunos cálculos para la modificación del PWM, esto es debido a que no implementa los multiplicadores en hardware y los números mayores a 8 bits pueden ralentizar el proceso de cálculo. Por lo tanto, se procederá a introducir los microcontroladores ARM.