

Capítulo 3. Microcontroladores

En este capítulo se define el microcontrolador, con una breve semblanza histórica sobre procesadores. Se habla más detenidamente sobre los PICs y se muestran características del PIC seleccionado para el proyecto.

3.1 Definiciones

3.1.1 Un sistema computacional incluye una Unidad Central de Procesamiento (CPU), una unidad de memoria que contiene el programa y datos y una interfaz de entrada/ salida con dispositivos de entrada y salida asociados.

Antes de que Intel presentara el primer microprocesador en 1971, el CPU era constituido por varios componentes. De hecho, en 1958 la computadora SAGE de la fuerza aérea norteamericana requería 40,000 pies cuadrados, 3 MW de potencia y tenía una unidad de memoria magnética de 30,000 tubos con 4 mil palabras de 32 bits cada uno. El término microprocesador se llegó a usar por primera vez en Intel en 1972 y se refiere generalmente a la implementación de de las funciones del CPU en un solo circuito LSI (*Large Scale of Integration*). Una microcomputadora, por lo tanto, es una computadora construida con un microprocesador como CPU y otros pocos componentes para memoria y dispositivos de entrada/salida. La microcomputadora que contenía el Intel 4004 de 1972 consistía de 4 circuitos integrados: el CPU, una memoria ROM para el programa, memoria RAM para datos y un chip de corrimiento de registros como expansión de salida.

3.1.2 Un microcontrolador es una microcomputadora que, en el mismo encapsulado, incluye el CPU, la memoria y las unidades de entrada/salida. Se usan para aplicaciones específicas de control que se pueden encontrar en productos tan variados como hornos microondas, tostadores, reproductores de mp3 y automóviles.

3.1.3 PICs

Los PICs (*Peripheral Interface Controller*) son microcontroladores fabricados por la compañía Microchip Technologies. Los PICs son microcontroladores de tipo RISC (*Reduced Instruction Set Computer*) con arquitectura Harvard.

Hoy en día los PICs vienen con varios periféricos integrados, como módulos de comunicación serie, convertidores analógico-digitales, *timers*, módulos de captura y comparación, etc.

Algunas de las características más relevantes de los PICs se muestran a continuación:

- Área de código y de datos separadas (Arquitectura Harvard).
- Un reducido número de instrucciones de largo fijo.
- La mayoría de las instrucciones se ejecutan en un solo ciclo de ejecución (4 ciclos de reloj).
- Un solo acumulador (W), cuyo uso es implícito (no está especificado en la instrucción).
- Todas las posiciones de la RAM funcionan como registros de origen y/o de destino de operaciones matemáticas y otras funciones.
- Una pila de hardware para almacenar instrucciones de regreso de funciones.
- Una relativamente pequeña cantidad de espacio de datos direccionable (típicamente, 256 bytes), extensible a través de manipulación de bancos de memoria.
- El espacio de datos está relacionado con el CPU, puertos, y los registros de los periféricos.
- El contador de programa esta también relacionado dentro del espacio de datos, y es posible escribir en él (permitiendo saltos indirectos).

Los PICs se pueden clasificar de acuerdo al tamaño de palabra que emplean. Siendo así, las familias básicas tienen un tamaño de palabra de 8 bits, la gama media tiene un tamaño de palabra de 16 bits, mientras que la gama más alta trabaja con palabras de 32 bits.

3.2 Elección del microcontrolador

Se escogió la serie PIC18FXX2 ya que, dentro de las gamas bajas de los PICs (con datos de 8 bits y sin llegar a los dsPICs) esa serie logra hacer una multiplicación por hardware en un ciclo de ejecución, lo cual, empleando la frecuencia más alta de reloj que permiten, implica obtener el resultado de una multiplicación en 100 nanosegundos, por lo que imponía el reto a vencer en cuanto a rapidez del coprocesador.

Se escogió el PIC18F452, cuyas características son las siguientes:

- Memoria de programa de 32 kbytes, que equivale a 16384 instrucciones de una palabra.
- Memoria RAM de 1536 bytes.
- Memoria EEPROM para datos de 256 bytes.
- 16 bits de tamaño de palabra para instrucciones y 8 bits para datos
- Frecuencia de operación de 40 MHz
- Multiplicador hardware de un ciclo de 8x8 bits.
- Set de 75 instrucciones.
- Módulo de comunicación serie, módulo de PWM y captura y comparación, módulo de conversión analogico-digital de 10 bits y módulo de temporizador como periféricos
- 5 puertos de entrada/salida.
-

El diagrama de bloques del PIC18F452 se muestra a continuación:

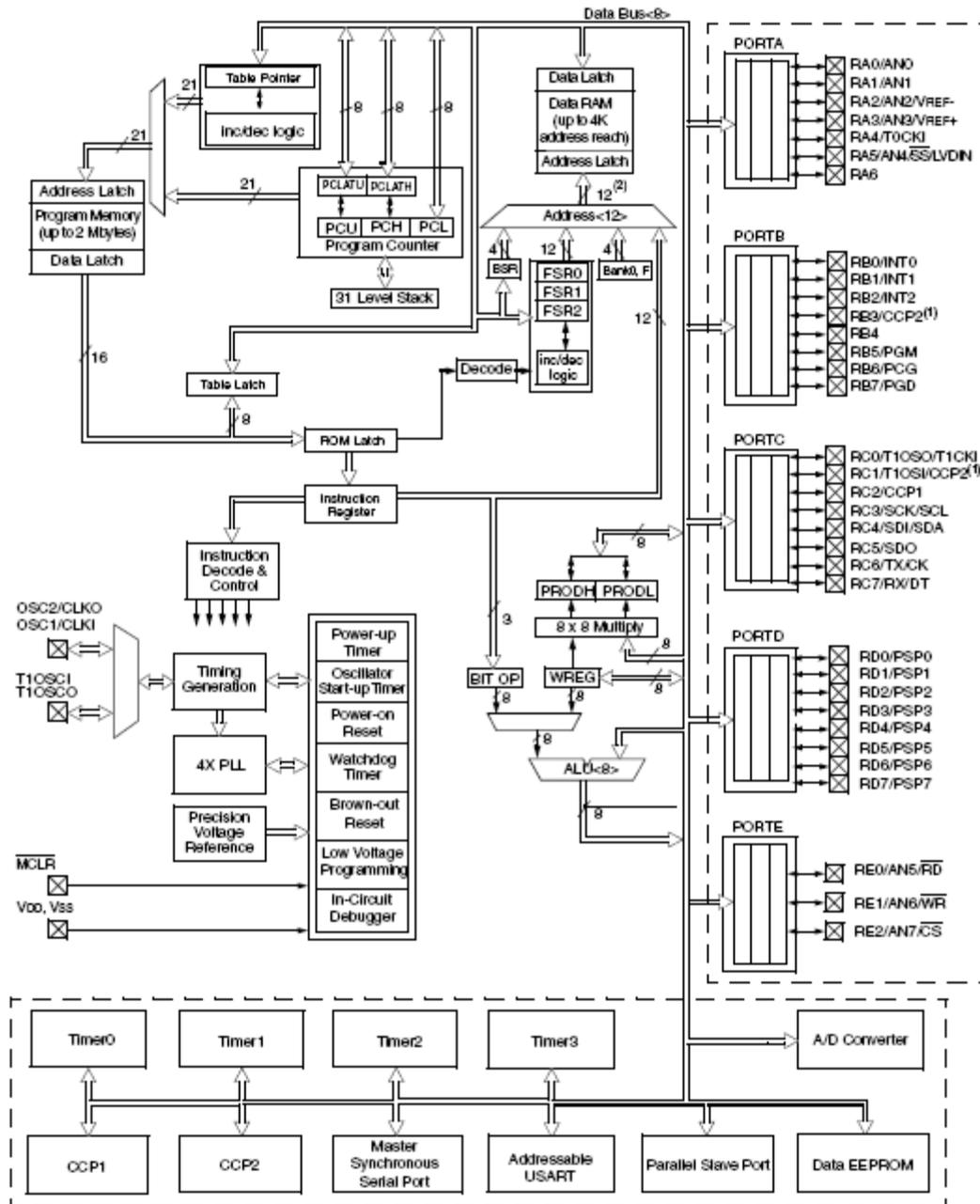


Figura 3.1 Diagrama de bloques del PIC18F45.

Organización de memoria en el PIC18F452

Existen tres bloques de memoria dentro del microcontrolador:

- Memoria de Programa
- Memoria RAM de datos
- Memoria EEPROM de datos

El contador de programa (*PC*, *program counter*) de 21 bits es capaz de direccionar todas las localidades de memoria de programa. La dirección del vector de RESET se encuentra en la localidad 0000h.

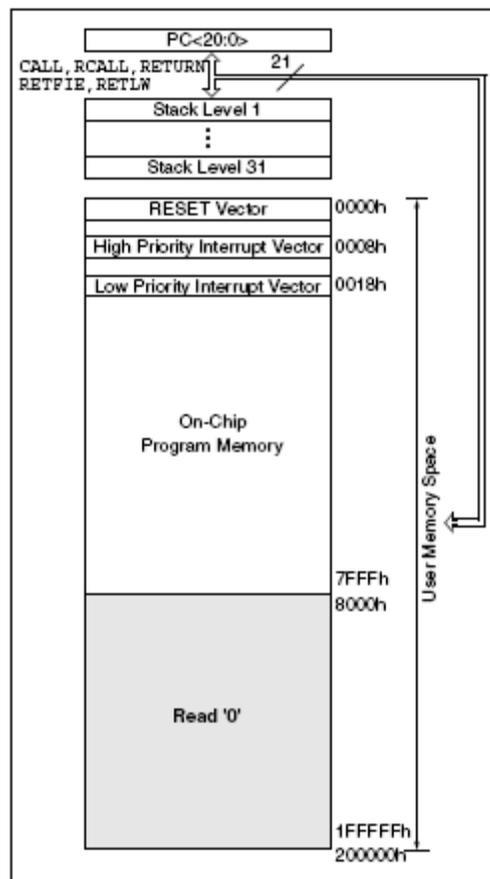


Figura 3.2 Mapa de memoria de programa del PIC18F452.

El *stack* de dirección de regreso permite que pueda ocurrir una combinación de hasta 31 llamadas de programa e interrupciones. El PC se empuja al *stack* cada que se ejecuta una instrucción CALL o RCALL, o cuando se reconoce una interrupción. El valor de PC se recupera del *stack* con instrucciones como RETURN, RETLW o RETFIE.

El *stack* opera como RAM de 31 palabras de 21 bits, con un apuntador de *stack* (*stack pointer*, SP) de 5 bits inicializado en 00000b después de todos los RESETs. Durante una instrucción de tipo CALL el *stack pointer* se incrementa y en la localidad a la que apunta se escribe el valor del PC. Por otro lado, al ejecutarse una instrucción de tipo RETURN, los contenidos de la localidad de la RAM apuntada por el SP se transfieren al *program counter* y el *stack pointer* se decrementa.

Ciclo de reloj y ciclo de instrucción

La entrada de reloj se divide internamente en cuatro pulsos, Q1, Q2, Q3 y Q4. Internamente el PC se incrementa cada Q1, se busca la instrucción de la memoria de programa y se transfiere al registro de instrucción en Q4. La instrucción se decodifica y se ejecuta durante los siguientes Q1 a Q4. La siguiente figura muestra las señales previamente explicadas.

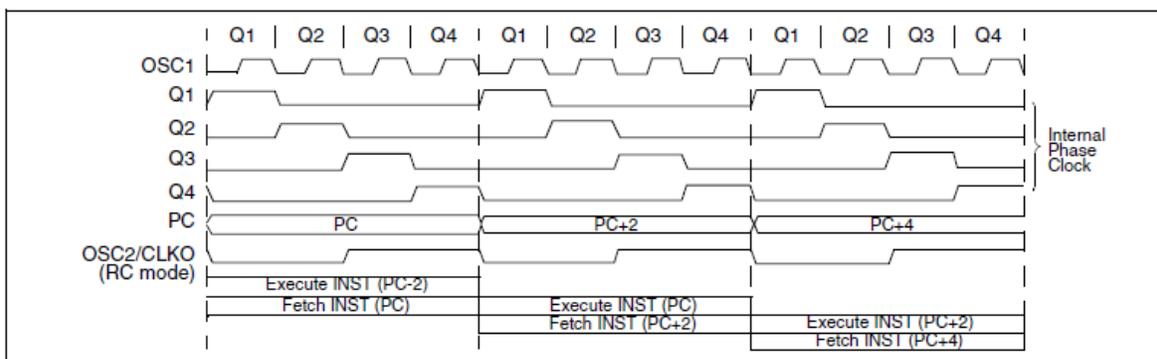


Figura 3.3 Ciclo de reloj e instrucción.

El ciclo de instrucción toma cuatro ciclos Q. La búsqueda de la instrucción (*fetch*), la decodificación y la ejecución tardan un ciclo de instrucción en realizarse, sin embargo por el método *pipeline*, cada instrucción se ejecuta efectivamente en un ciclo.

Instrucciones en la memoria de programa

La memoria de programa se direcciona en bytes. Las instrucciones se almacenan en dos o cuatro bytes dentro de la memoria de programa por lo que el PC se incrementa de dos en dos y así mantener alineada la lectura de las instrucciones.

Las instrucciones CALL y GOTO tienen una dirección de memoria de programa absoluta incrustada dentro de la misma instrucción. Los datos almacenados en dichas instrucciones son direcciones que escriben sobre el PC, el cual accesa directamente el byte de dirección requerido de la memoria de programa.

Organización de la memoria de datos

La memoria de datos está implementada como RAM estática. Cada registro tiene una dirección de 12 bits permitiendo los 4096 bytes de la memoria de datos. El mapa de memoria de datos se divide en 16 bancos que contienen 256 bytes cada uno. Los cuatro bits bajos del registro de selección de banco (*bank select register*, BSR) seleccionan el banco que será accedido, los bits altos no se implementan.

La memoria de datos puede ser accesada directa o indirectamente. El direccionamiento directo puede requerir el uso del registro BSR. El direccionamiento indirecto requiere el *File Select Register* (FSRn) y el correspondiente *Indirect File Operand* (INDFn). Cada FSR mantiene el valor de 12 bits de la dirección que puede ser usada para acceder cualquier localidad de la memoria de datos.

El set de instrucciones y la arquitectura del PIC permiten operaciones entre todos los bancos. Esto puede ser logrado mediante direccionamiento indirecto, o mediante la instrucción MOVFF, que mueve un valor de un registro a otro.