

INTRODUCCIÓN A LOS MICRO- CONTROLADORES RISC. -PIC DE MICROCHIPS-

*Tito Flórez C.
Departamento de Ingeniería de Sistemas.
Universidad Nacional de Colombia.*

Resumen

Los microcontroladores han prestado una gran ayuda en muchos campos, de los cuales uno de los más conocidos es el control. Iniciarse en el campo de los microcontroladores requiere normalmente dedicarle una enorme cantidad de tiempo, debido, entre otros, a la facilidad de perderse en el mar de información contenida en sus manuales. Debido a la gran similitud que poseen los PIC con respecto a su arquitectura, conjunto de instrucciones y programación, se toma el PIC16C84 como un buen prototipo de microcontrolador, y se da la información más importante (con sus respectivos ejemplos), para poder ubicarse correctamente en el manejo de éstos.

INTRODUCCIÓN

Hace pocos años se popularizaron en el mercado los circuitos integrados conocidos con el nombre de microcontroladores, los cuales vinieron a facilitar enormemente los diseños de muchos circuitos que antes se desarrollaban principalmente con *hardware*, reduciendo en forma significativa su costo y tamaño, aumentando su versatilidad y potencialidad, ya que son unos computadores en un solo *chip*.

Estos elementos cada día se conocen más, debido entre otras cosas a su bajo costo, relativa sencillez para programarlos, tamaño reducido y gran potencialidad para un gran número de aplicaciones

entre la que sobresale los sistemas de control. Además, los microcontroladores son excelentes para iniciarse en la programación del lenguaje “*assembler*”.

A muchos profesionales y estudiantes de áreas como sistemas, electrónica, eléctrica, de automatización, comunicaciones, etc., se les hace necesario aprender a manejar los microcontroladores, ya que éstos ofrecen una gran ayuda y un gran potencial de aplicación en su campo. Pero ese aprendizaje se hace muy complicado cuando no se tiene una buena guía inicial, y se tiene la creencia de que para aprender a manejarlos es necesario tener una gran cantidad de conocimientos en el área de sistemas y de electrónica, lo mismo que la de poseer un buen dominio en el idioma inglés (casi toda la información viene en inglés). Poseer los conocimientos anteriores, sin duda facilita su aprendizaje, mas algunos pocos conocimientos de programación y una buena guía son suficientes para lograr muy buen manejo de éstos.

Dentro de los microcontroladores hay un grupo en auge conocido con el nombre de PIC, que poseen características RISC (Computadores con una Serie Reducida de Instrucciones). Esta serie de microcontroladores son similares en muchos

aspectos: en sus instrucciones, programación y arquitectura. Este hecho facilita enormemente poder llegar a manejar cualquiera de estos PIC, conociendo cómo funciona uno de ellos. Por tal razón, particularizaremos nuestro estudio en uno de ellos (PIC16C84), y ello se hace extensivo casi en su totalidad a los demás.

I. EL PIC 16C84

Se escoge el PIC 16C84 para trabajar, ya que es de fácil consecución, costo reducido (cerca de US\$4), tamaño cómodo (2,3 cm de largo por 0,8 cm de ancho), es un buen prototipo dentro de los PIC, y posee memoria EEPROM (Memoria ROM Borrable Eléctricamente) que lo hace apto para programarlo directamente una y otra vez, sin tener que estar sometiéndolo a la acción del borrador de luz ultravioleta, lo cual exige tener el borrador y poseer la paciencia y el tiempo que esto conlleva.

Aun cuando se tratará solamente el PIC 16C84, todos los tópicos serán aplicables al PIC 16F84. Este último posee más registros de propósito general (68), y otras mejoras respecto al primero. La configuración de los pines del 16C84 aparece en la figura 1.

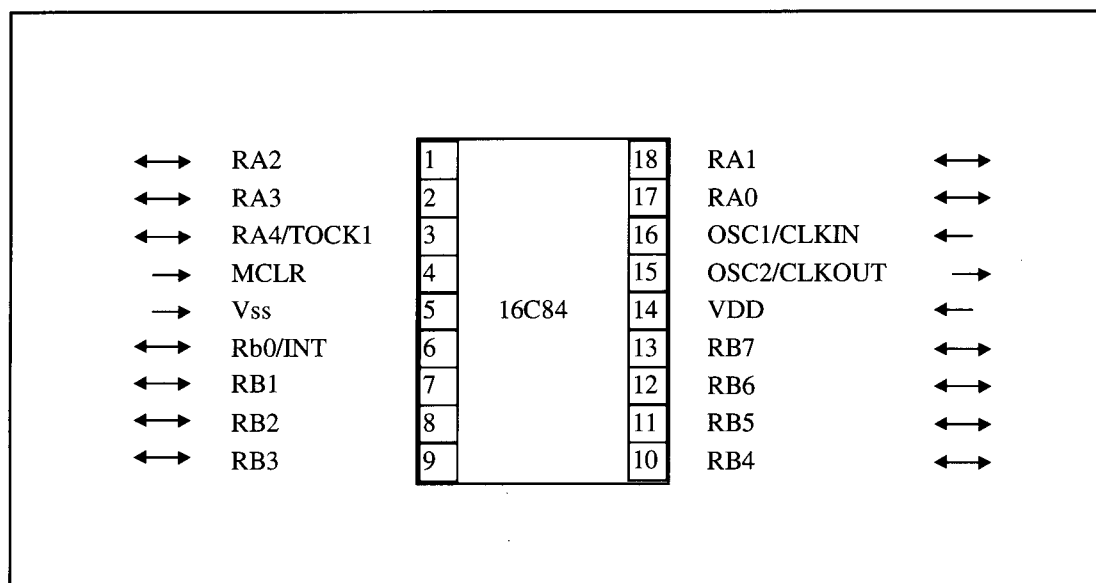


Figura 1. Configuración de los pines del 16C84.

II. SISTEMA DE DESARROLLO

Para programar el PIC, deben poseerse un computador y el sistema de desarrollo (development system) para los microcontroladores PIC16/17. Dicho sistema cuesta cerca de 200 dólares; es relativamente fácil de conseguir, y viene con:

1. Un programador (también conocido como *quemador*). Actualmente existe un programador universal, el cual es muy versátil. También puede conseguirse versiones anteriores no universales, en tal caso asegurarse que sea el PICSTART-16B1, ya que existen otras referencias como el PICSTART-16C que sirven solamente para otros PIC, tales como el PIC 16C64 y el PIC 16C74.
2. Un cable para conectar entre el programador y el puerto serial.
3. Un adaptador de potencia para conectar al programador.
4. Su respectivo *software*.

La instalación del *software* en el computador no presenta ningún problema, y se hace de forma similar a la instalación de cualquier programa. Dicho *software* posee generalmente tres módulos principales:

- a. Un ensamblador como el MPASM que tiene entre otras, la función de pasar del lenguaje *assembler* al lenguaje de máquina.
- b. Un programa como el MPS16B, que tiene por función coger el programa que se le indique (nombre.HEX) y controla al programador, para que éste programe el respectivo PIC.
- c. Un programa como el MPSIM, que sirve solamente cuando se desea hacer la simulación del programa realizado. Es decir, sirve cuando queremos ver en el monitor del computador cómo irá a funcionar el PIC. una vez esté

“quemado” con nuestro programa. En él puede verse paso a paso cómo van cambiando todos los registros y si lo que va ocurriendo corresponde a lo deseado. Razón por la cual es una herramienta que puede llegar a servir mucho para encontrar errores en la programación del PIC. Mas lo anterior no indica que sea una herramienta obligatoria, y se puede trabajar perfectamente sin necesidad de usar este tipo de programa.

5. Los manuales.

Dentro de estos pueden venir los siguientes:

- a. Microchip Data Book.
- b. MPASM User's Guide.
- c. PICSTART-16B1 User's Guide.
- d. Embedded Control Handbook.
- e. MPSIM User's Guide.

Hace relativamente poco tiempo salió al mercado una versión mejorada conocida como la “PICSTART PLUS”, cuyo software es más amigable, el programador sirve para todos los PIC, y casi todos los manuales vienen en un CD.

III. INFORMACIÓN CLAVE

Leer cuidadosamente los manuales anteriores requiere mucho tiempo, ya que existe en ellos una gran cantidad de información. Para alguien que esté iniciándose en este tema, no es muy aconsejable lo anterior, ya que muy seguramente quedará desorientado dentro de un mar de información. Por tal razón, es mejor aprender primero la cosas fundamentales para su correcto manejo, y después, si se desea, profundizar dentro de los manuales todo lo que desee.

Por tal razón, para empezar, tomar el *Microchip Data Book*. Una vez en éste, ubicarse en la sección 8-Bit Microcontroller Product Specifications, y dentro de ésta ubicarse en las páginas correspondientes al PIC 16C84 (si se posee el CD, ubicarse en el PIC 16C84).

Dentro de estas páginas existe una información clave que se debe poseer para llegar a manejar los PIC de forma sencilla y confiable. Dicha información es la siguiente:

1. Características eléctricas.

Esta información se debe tener en cuenta para no llegar a dañar el PIC y asegurarse de su correcto funcionamiento. Dentro de éstas las más importantes son:

- a. Voltaje de funcionamiento (Vdd) va de 4 a 6 voltios (usar 5 voltios es muy atractivo, ya que es un voltaje fácil de obtener, y es compatible con los circuitos TTL).
- b. Máxima corriente que puede recibir cualquier pin es de 25 miliamperios.
- c. Máxima corriente que puede recibir en total el puerto A es de 80 miliamperios.
- d. Máxima corriente que puede recibir en total el puerto B es de 150 miliamperios.
- e. Máxima corriente que puede dar cualquier pin es de 20 miliamperios.
- f. Máxima corriente que puede dar en total el puerto A es de 50 miliamperios.
- g. Máxima corriente que puede dar en total el puerto B es de 100 miliamperios.

Lo anterior significa que siempre que un pin sea escogido para salida, se debe estar atentos a no manejar corrientes que excedan las especificadas, y un método, para empezar, es colocando una resistencia a la salida de dicho pin (un buen valor puede ser usar una resistencia de 1000 ohmios a ¼ de Watt en cada pin de salida).

2. Registros acumuladores del microcontrolador.

Para efectos prácticos se puede decir que el microcontrolador posee un único acumulador llamado W, el cual es de 8 *bits* y es usado por la ALU (unidad aritmética y lógica) y sobre él actúa casi la mitad de las instrucciones.

3. Registros de propósito especial.

Es importante manejar adecuadamente los siguientes registros, los cuales aparecen en una de las figuras del manual, bajo el nombre de *Register File Map*, en donde aparece también sus direcciones con sus respectivas páginas.

- a. STATUS (Status Register : registro de estado), y los *bits* más importantes son estos tres:

A. *Bit* # 0 (C) se pone automáticamente en “1” cuando en una operación de suma o resta, ocurre un acarreo hacia el noveno *bit* (los registros son de 8 *bits*). Es muy útil cuando se desea saber si un número es mayor o menor que otro, o si se produce *overflow* en operaciones con números sin signo.

B. *Bit* # 2 (Z) se pone en “1” cuando el resultado de una operación aritmética o lógica es cero. Es muy útil cuando se manejan ciclos o se desea saber si un número es igual a otro.

C. *Bit* # 5 (RP0) se usa para poder cambiar de página. Por ejemplo, para poder llegar a los registros TRISA y TRISB que es en donde el programador define si los pines serán de entrada o de salida, es necesario pasar a la página # 1. Lo anterior se consigue colocando un “1” en el *bit* 5 del STATUS, lo cual se puede realizar con la instrucción “bsf” (*Bit Set f*) de la siguiente manera:

```
bsf STATUS,5
```

lo cual traduce: coloque un “1” en el *bit* #5 del registro STATUS.

Es muy importante regresar a la página “0” después de haber definido los registros de interés en la página “1”, y se puede realizar con la instrucción “bcf” (*Bit Clear f*):

```
bcf STATUS,5
```

- b. TRISA. Se usa para definir los pines que serán de entrada y los pines que serán de salida, obviamente con relación al puerto A. Si un *bit* lo ponemos a "1", el pin asociado con dicho *bit* será de entrada, y si el *bit* lo ponemos a "0", dicho pin asociado será de salida.
- c. TRISB. Se usa para definir los pines que serán de entrada y los pines que serán de salida, obviamente en relación con el puerto B. Si un *bit* lo ponemos a "1", el pin asociado con dicho *bit* será de entrada, y si el *bit* lo ponemos a "0", dicho pin asociado será de salida.
- d. PORTA (puerto A). En él están los datos que están entrando por los pines del puerto A, y en él se colocan los datos que se desea salgan por los pines del puerto A.
- e. PORTB (puerto B). En él están los datos que están entrando por los pines del puerto B, y en él se colocan los datos que se desea salgan por los pines del puerto B.
- f. INTCON (registro de interrupciones) es útil cuando van a manejarse interrupciones.

4. Registros de propósito general, SRAM.

Empiezan en la dirección 0C (hexadecimal). El PIC 16C84 posee un total de 36, mientras el PIC 16F84 posee un total de 68.

5. Instrucciones del 16C84.

El 16C84 posee 35 instrucciones muy sencillas, las cuales en general son compatibles con los demás PIC. Es a través de estas instrucciones que se hace la programación del micro-controlador. Esas instrucciones pueden encontrarse fácilmente en la sección Instruction Set Summary del manual Microchip Data Book.

Conclusiones

El manejo de los microcontroladores es relativamente sencillo cuando se posee la información apropiada.

El manejo de los microcontroladores no es exclusivo del personal con gran conocimiento en sistemas y electrónica.

Un individuo con algunos conocimientos en programación puede perfectamente empezar a manejar los microcontroladores.

BIBLIOGRAFÍA

1. *EMBEDDED Control Handbook*. 1993. Microchip Technology Inc.,USA.
2. *Microchip Data Book*. 1994. Microchip Technology Inc.,USA.
3. *Mpasm User's Guide*. 1994. Microchip Technology Inc.,USA.
4. *MPSIM User's Guide*. 1994. Microchip Technology Inc.,USA.
5. *PICSTART Plus, Development System User's Guide*. 1996. Microchip Technology Inc.,USA.
6. *PICSTART-16B1 User's Guide*. 1994. Microchip Technology Inc.,USA.