

Unidad IV: Programación de microcontroladores

El microcontrolador ejecuta el programa cargado en la memoria Flash. Esto se denomina el código ejecutable y está compuesto por una serie de ceros y unos, aparentemente sin significado. Dependiendo de la arquitectura del microcontrolador, el código binario está compuesto por palabras de 12, 14 o 16 bits de anchura. Cada palabra se interpreta por la CPU como una instrucción a ser ejecutada durante el funcionamiento del microcontrolador. Todas las instrucciones que el microcontrolador puede reconocer y ejecutar se les denominan colectivamente Conjunto de instrucciones. Como es más fácil trabajar con el sistema de numeración hexadecimal, el código ejecutable se representa con frecuencia como una serie de los números hexadecimales denominada código Hex. En los microcontroladores PIC con las palabras de programa de 14 bits de anchura, el conjunto de instrucciones tiene 35 instrucciones diferentes.

4.1 Modelo de programación

- ***Funciones de un programa de control de procesos:***

- ✓ „ Adquisición y acondicionamiento de datos
- ✓ „ Control digital directo
- ✓ „ Supervisión del sistema
- ✓ „ Control de secuencias

- ***„ Modelos a estudiar***

- ✓ „ Organigramas (diagramas de flujo)
- ✓ „ Maquinas de estado finita (Finite State Machine--FSM)

„

- ***Organigramas***

- Suelen utilizarse en aplicaciones sencillas o en la planificación de rutinas en las que se divide una aplicación más compleja.

Máquina de estados finita (FSM)

- Estados: Son las diferentes situaciones en las que puede encontrarse el sistema.
- Entradas (señales de transición de estados): Constituyen los acontecimientos que producen los cambios de estado.
- Salidas: Respuesta del sistema ante los cambios en las entradas.
- Funciones de transición de estado: Descripción detallada, para cada estado, de cuál es el siguiente estado y la salida del sistema, cuando se producen las distintas entradas.

La forma en que los representan son:

- Diagrama de grafo de estado
- Tabla de transiciones

4.2 Estructura de los registros del CPU

Los programas de los microcontroladores suelen dividirse en dos grandes grupos:

- ✓ **La estructura de interrupciones**, formada por las rutinas de servicio a las mismas.
- ✓ **El programa principal, que consta de dos partes:** La rutina de inicialización del sistema y Un lazo de ejecución perpetua, en donde n perpetua, en donde se ejecutan todas se ejecutan todas aquellas tareas que no estén ligadas a la estructura de n ligadas a la estructura de interrupciones: máquina de estados principal, rutinas no quina de estados principal, rutinas no críticas, inicialización del circuito de vigilancia

4.3 Modos de direccionamiento

- Modos de direccionamiento 8051

Dir

En el direccionamiento directo, se indica la dirección a operar de forma absoluta.

Para la familia de microcontroladores 8051 se dispone de 256 direcciones directas, correspondientes a (Ram interna + Registros SFR).

El OpCode va seguido de un byte que representa la dirección.

MOV A,30H ; $A \leftarrow (30H)$, El contenido de la dirección 30H se mueve al acumulador
ADD A,31H ; $A \leftarrow (A) + (31H)$, Se suma el contenido de acumulador con el de la dirección 31H
; el resultado se deposita en el acumulador

Por registro (Rn)

Rn. Para el direccionamiento por registro el código de la instrucción (OpCode) contiene la información del registro a operar, del banco de registros seleccionado.

MOV R1,A ; $R1 \leftarrow (A)$, El contenido del acumulador se mueve al registro R1

OpCode de (MOV Rn,A) es F8H, en binario:

1	1	1	1	1	0	0	1
---	---	---	---	---	---	---	---

Los tres últimos bits identifican el registro sobre el cual se opera.

Indirecto (@)

@. Se especifica un registro que contiene la dirección del dato a operar.

R0, R1 cuando se accede a la memoria interna de direccionamiento indirecto 256 Bytes.

DPTR para el acceso a la memoria externa 64K Bytes.

MOV R0,#30H ; R0<-30H, mover al registro 0 el dato inmediato 30H (constante)
MOV A,@R0 ; A<-((R0)), mueve el contenido de la posición de memoria indicada en R0 al Acc.

Implícito

La instrucción indica implícitamente el registro sobre el que se ha de operar.

INC A ; A<-(A)+1, incrementar el acumulador

CPL A ; A<-($\tilde{A}$), complementa el acumulador

Inmediato (#)

#. El OpCode está seguido de una constante que representa el valor con el que se operará.

MOV A,#0 ; A<-0, poner cero en el Acc

MOV R1,#123 ; R1<-123, poner el valor 123 (decimal) en el registro uno

Indexado

Se utilizan dos registros para apuntar a la dirección que contiene el dato. La suma del contenido del DPTR más el acumulador determina la dirección a operar.

Este direccionamiento esta limitado a dos únicas instrucciones que mueven datos de la ROM al acumulador.

MOVC A,@A+DPTR ; A<-((DPTR+A)), el contenido de la posición que apunta el DPTR+A se deposita en el Acc

4.4 Conjunto de instrucciones

Un conjunto de instrucciones o repertorio de instrucciones, juego de instrucciones o ISA (del inglés Instruction Set Architecture, Arquitectura del Conjunto de Instrucciones) es una especificación que detalla las instrucciones que una CPU de un ordenador puede entender y ejecutar, o el conjunto de todos los comandos implementados por un diseño particular de una CPU. El término describe los aspectos del procesador generalmente visibles a un programador, incluyendo los tipos de datos nativos, las instrucciones, los registros, la arquitectura de memoria y las interrupciones, entre otros aspectos.

4.5 Lenguajes ensambladores

Ventajas:

- Permite desarrollar programas muy eficientes:
- Uso eficiente de recursos
- Código compacto
- Control total sobre el tiempo de ejecución de las instrucciones especialmente importante en aplicaciones en tiempo real

Desventajas:

- „ Programación costosa y difícil de modificar

4.6 Codificación

A menudo gastamos una gran cantidad de energía diseñando un circuito elegante, bien diseñado y robusto. Hoy día, el cerebro de muchos de estos circuitos es un microcontrolador que necesita un software para funcionar. ¿No es razonable entonces esperar un programa bien diseñado y escrito adecuadamente que se complemente con la calidad del trabajo del circuito? Aparentemente así es. Hablemos de la calidad del software. Es un hecho muy bien conocido: los errores matan a gente y los errores de programación no son una excepción. Todos los

años muere gente debido a errores de programas. Algunos accidentes fatales de avión, helicóptero y coches pueden ser atribuidos a problemas en el software. El mal funcionamiento del 'firmware' interno de los dispositivos médicos crea víctimas de manera regular. Los edificios se convierten en humo debido a errores de programa, y algunas personas son asesinadas a machetazos. El software erróneo puede incluso hundir barcos. Creedme, he visto que eso sucede.