

1.-Introducción a la unidad de control

La unidad de control es el elemento que se encarga de sincronizar las acciones que realiza cada una de las unidades funcionales de un computador. Las funciones de la unidad de control son básicamente dos.

Interpretación de las instrucciones: La unidad de control debe ser capaz de decodificar los códigos de operación y los modos de direccionamiento de las instrucciones y actuar de forma diferente para cada uno de ellos.

Secuenciamiento de las operaciones: La unidad de control se encarga de la temporización de las distintas operaciones necesarias para la ejecución de cada instrucción. También debe controlar el secuenciamiento de las instrucciones en función de la evolución del registro contador de programa.

Se llaman **señales de control** a las variables binarias que controlan las entradas y salidas de información de los registros y el funcionamiento de las unidades funcionales.

La ejecución de una instrucción se divide en varias etapas que deben realizarse según una secuencia muy precisa de señales de control que establece la señal de control. Hay dos formas básicas para implementar la unidad de control:

- Unidad de control **cableada:** Realiza sus funciones mediante elementos hardware. No la analizaremos en profundidad aquí.

- Unidad de control **microprogramada:** Es más lenta que la anterior, pero permite implementar instrucciones más potentes y flexibles. Será analizada en profundidad en los siguientes apartados.

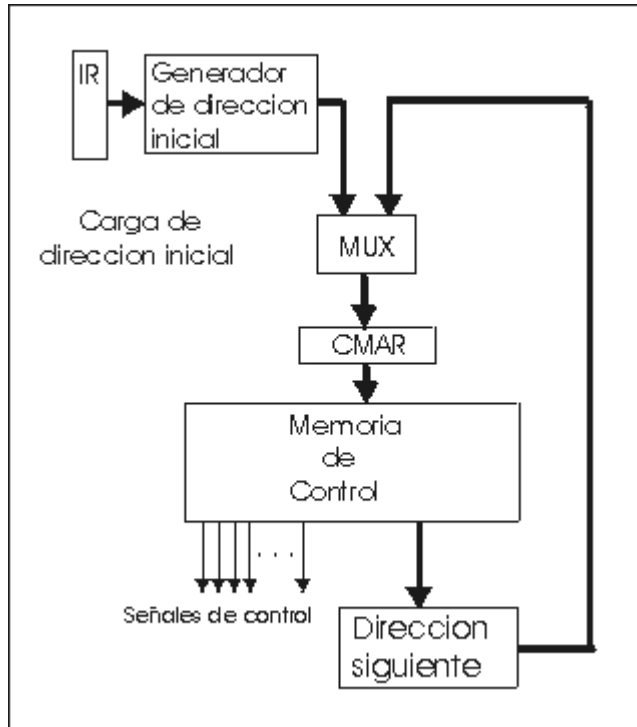
2.-Introducción a la unidad de control microprogramada

Llamaremos palabra de control a una palabra binaria cuyos bits individuales representan las diferentes señales de control de una máquina. Se denomina microprograma de una instrucción a la secuencia de valores de la palabra de control precisa para la ejecución de esa instrucción. Se llama microinstrucción a cada valor de la palabra de control de los que componen un microprograma.

En una unidad de control microprogramada las microinstrucciones se almacenan en una memoria llamada memoria de control.

3.-Modelo de Wilkes

El concepto de microprogramación fue propuesto por Maurice Wilkes (1951) como un procedimiento sistemático para diseñar unidades de control de computadores. En el modelo de Wilkes, las palabras de la memoria de control tenían dos campos: un campo con las señales de control para gobernar la maquina y otro campo con la dirección de la siguiente microinstrucción. Este último campo podía verse afectado en algunos casos por algunas señales externas, como los flags.



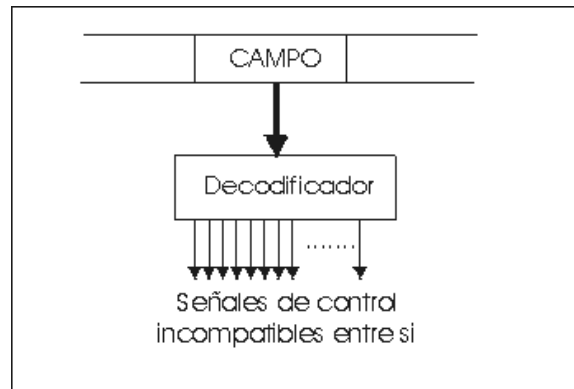
En este esquema se puede ver que en función de la señal "carga de dirección inicial" se selecciona entre una dirección dependiente del código de operación de la instrucción u otra procedente del campo de dirección siguiente. El resultado de esta selección entra en el **registro de dirección de la memoria de control** (*Control Memory Address*) o CMAR con lo que se selecciona la siguiente palabra de control.

4.-Codificación de las microinstrucciones

Las microinstrucciones almacenadas en la memoria de control en un principio tienen la misma longitud que la palabra de control, ya que se asigna un bit individual a cada una de las señales de control. Esto es un grave inconveniente, ya que hace que el microprograma ocupe mucho, y la memoria de control es un recurso escaso, ya que reside en el circuito integrado del procesador, donde la superficie debe repartirse entre todas las funciones. Existen algunas técnicas para solucionar este problema que veremos en los apartados siguientes.

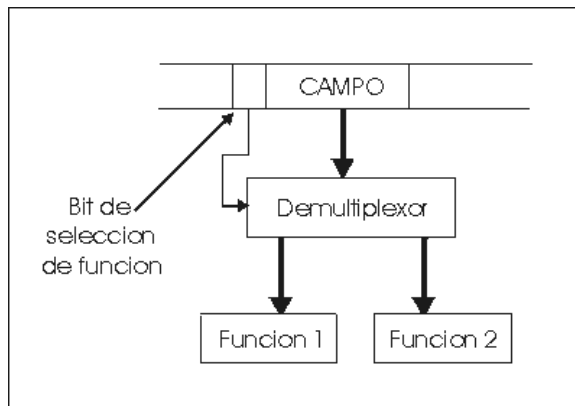
Codificación por campos

Muchas de las señales de control no son nunca simultáneas. Por ejemplo, si solo disponemos de un bus, nunca habrá dos registros que escriban información al mismo tiempo en el. Por esta razón podemos agrupar todas las señales incompatibles entre si en un único campo, que luego, dependiendo de su valor, será decodificado. Esto se muestra en la siguiente figura.



Microinstrucciones con formato múltiple

Otra técnica para acortar la longitud de la palabra de control consiste en hacer que un mismo campo, tenga dos funciones diferentes, que no se puedan usar simultáneamente. Un bit adicional indicara el significado del campo en cada caso. Este bit será la entrada de un demultiplexor que servirá para elegir el significado adecuado en cada caso.



Por cualquiera de estos dos métodos de codificación puede comprimirse la microinstrucción fuertemente, aunque a costa de alargar el microprograma. En este caso hablaremos de **microprogramación vertical**. Por el contrario, si no se comprime la microinstrucción hablaremos de **microprogramación horizontal**. En la práctica hay que llegar a una solución de compromiso intermedia en las que se compriman las instrucciones en lo posible pero alargando poco el microprograma ya

que un microprograma largo no solo ocupara mas memoria de control sino que también hará mas lenta la ejecución de las instrucciones.

5.-Secuenciamiento en los microprogramas

Aquí estudiaremos las maneras de hallar la dirección del siguiente microinstrucción a ejecutar. También veremos como se implementan en los microprogramas las instrucciones de bifurcación condicional. Estas microinstrucciones son necesarias para dar posibilidad al microprograma de avanzar por caminos diferentes en función de diferentes factores como son los modos de direccionamiento utilizados para los operandos o el estado de las banderas de estado.

Secuenciamiento implícito

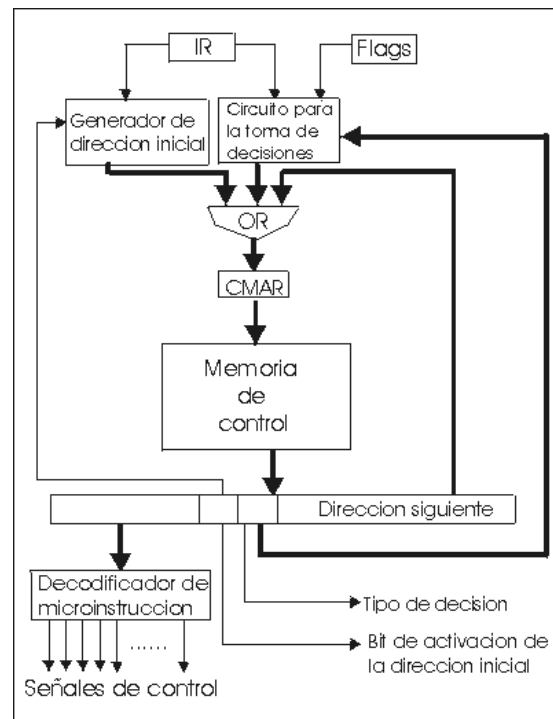
Este método sigue la misma pauta que las macroinstrucciones, ello significa que, en general, la microinstrucción siguiente se encontrará situada a continuación de la actual en la memoria de control. Esta forma de actuar exige la existencia de un registro **contador de microprograma**, que se vaya incrementando con la ejecución de cada una de las microinstrucciones. Como es lógico, las microinstrucciones no siempre se ejecutaran consecutivas, por lo que deben existir microinstrucciones específicas para efectuar bifurcaciones. Para codificar la dirección en las bifurcaciones suele emplearse la técnica de codificación por formato múltiple, en la que existe un bit que indica si la microinstrucción es una bifurcación y, si es así, algunos de los campos cambian de significado para contener la dirección de la microinstrucción siguiente. El empleo de secuenciamiento implícito ralentiza la ejecución de instrucciones ya que las microinstrucciones específicas de bifurcación utilizan ciclos adicionales de reloj.

Secuenciamiento explícito

Este método consiste en especificar en todas las microinstrucciones la dirección de la microinstrucción siguiente. Este método puede parecer poco rentable en cuanto al numero de bits ocupados porque hay que reservar un campo de dirección siguiente en todas las microinstrucciones, sean o no de bifurcación. En la práctica esto no es así, ya que en los microprogramas suele haber muchas bifurcaciones lo que hace que el campo de dirección siguiente sea útil en muchas microinstrucciones. Este método es más rápido que el anterior ya que no hay que poner microinstrucciones específicas de bifurcación que exigirían la ejecución de microinstrucciones adicionales.

6.-Hardware de la unidad de control microprogramada

El hardware de la unidad de control microprogramada depende mucho de los métodos empleados para realizar las distintas funciones: codificación de los campos, secuenciamiento, etc. En general, el hardware necesario para gobernar una unidad de control microprogramada será el compendio de todo lo dicho con anterioridad, como lo muestra la figura adjunta:



El esquema de la página siguiente muestra un microprocesador completo, junto con su memoria ROM y RAM conectadas. Adicionalmente están presentes diferentes componentes de su arquitectura básica, y se detalla el interior de la Unidad de Control.

La siguiente tabla muestra la estructura del microprograma presente en la ROM de control que debe controlar todo el hardware del sistema entero, es decir:

[illegible]

