Control Microprogramado

Introducción

El término *microprograma* lo acuñó M. V. Wilkes al principio de los años 50, propuso una aproximación al diseño de la unidad de control que era ordenada y sistemática y evitaba la complejidad de la implementación cableada.

Al anunciarse en abril de 1960 el sistema/360 de IBM todos los modelos excepto los más grandes eran microprogramados, las ventajas de la microprogramación fueron lo bastante convincentes para que IBM dejara este cambio.

Conceptos básicos

El diseño de la unidad de control debe incluir lógica para realizar el secuenciamiento de las microoperaciones, la ejecución de las microoperaciones, la interpretación de los códigos de operación y la toma de decisiones basdas en los indicadores de la ALU. Lo anterior es difícil de diseñar, de verificar y de modificar el diseño original.

Una alternativa es realizar una unidad de control microprogramada. Si se revisa la siguiente tabla, se tienen las señales de control y cada microoperación está descrita en notación simbólica, es decir, en un lenguaje conocido como *lenguaje de microprogramación*. Cada línea describe un conjunto de microoperaciones que suceden y se conocen como microinstrucciones. Una secuencia de microinstrucciones se conoce como *microprograma o firmware*. El último término indica que el microprograma está entre el hardware y el software. El diseño en firmware es más fácil que en hardware, pero es más dfícil escribir un programa en firmware que en software.

Para implementar una unidad de control usando microprogramación se puede construir una *palabra de control* en la que cada bit represente una línea de control, con lo que cada microoperación representa un patrón diferente de unos y ceros en la palabra de control.

Microoperaciones y señales de control

Captación	t_1:MAR ← (PC)	C_2
	t_2:MBR ← Memoria	C_5, C_R
	PC ← (PC)+1	
	t_3:IR ← (MBR)	C_4
Indirecto	t_1:MAR ← (IR(Dirección))	C_8
	t_2:MBR ← Memoria	C_5, C_R
	t_3:IR(Dirección) ← (MBR(Dirección))	C_4
Interrupción	t_1:MBR ← (PC)	C_1
	t_2:MAR ← Dirección de salvaguarda	
	PC ← Dirección de la rutina	
	t_3:Memoria ← (MBR)	C_{12}, C_W

C_R =Señal de control de lectura (*“read”*) hacia e l bus del sistema

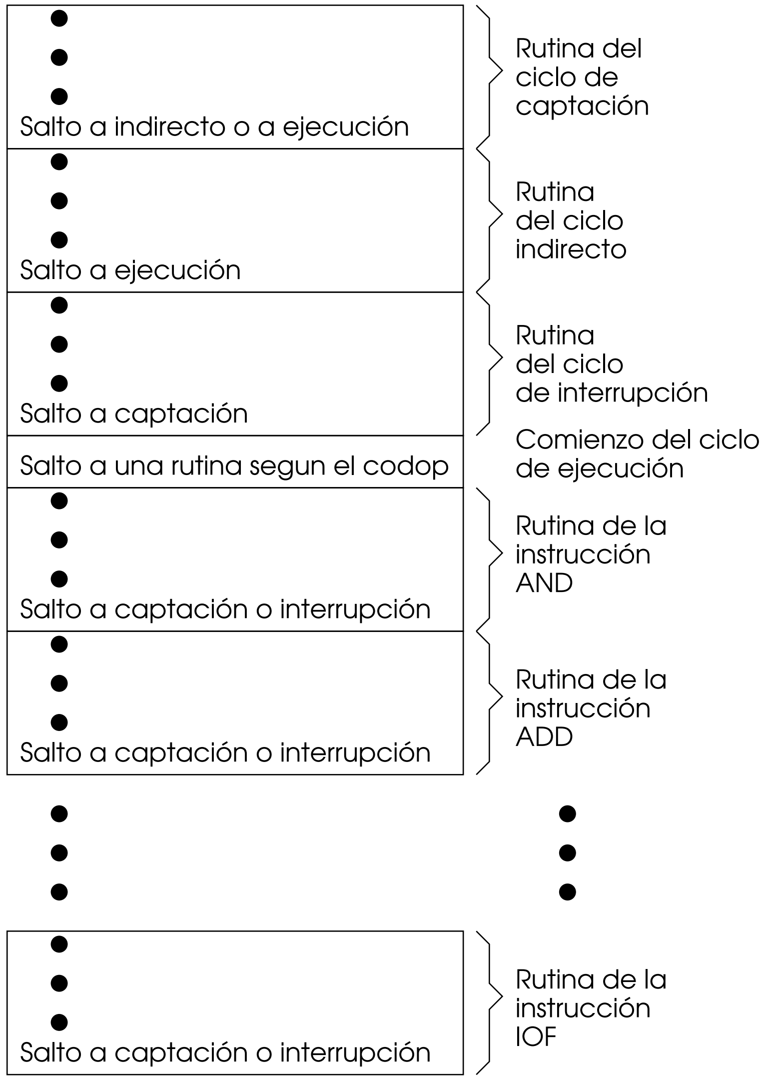
C_W =Señal de control de escritura (*“write”*) haci a el bus del sistema

Unidad de control microprogramada

La unidad de control se puede implementar suministrando los mecanismos necesarios para la ejecución del programa que describa el funcionamiento de la misma. Los elementos clave de esta implementación son:

- *Memoria de control* que almacena el conjunto de microinstrucciones.
- *Registro de dirección de control ("control address register", CAR)* que contiene la dirección de la siguiente microinstrucción a leer.
- *Registro intermedio de control ("control buffer register", CBR)* para almacenar las microinstrucciones leídas de la memoria de control. La parte izquierda del registro se conecta a las líneas de control que salen de la unidad de control.

Organización de la memoria de control



Ejecución de una instrucción

1. La unidad lógica de secuenciamiento emite una orden de lectura a la memoria de control.
2. La palabra cuya dirección se especifica en el registro de dirección de control se lee en el registro intermedio de control.
3. El registro intermedio de control genera las señales de control y contiene además la información de la siguiente dirección para la unidad lógica de secuenciamiento.
4. La unidad lógica de secuenciamiento carga en el registro de dirección de control una nueva dirección, basada en la información de la siguiente dirección del registro intermedio de control y en los indicadores de la ALU.

Todo lo anterior sucede durante un pulso de reloj.

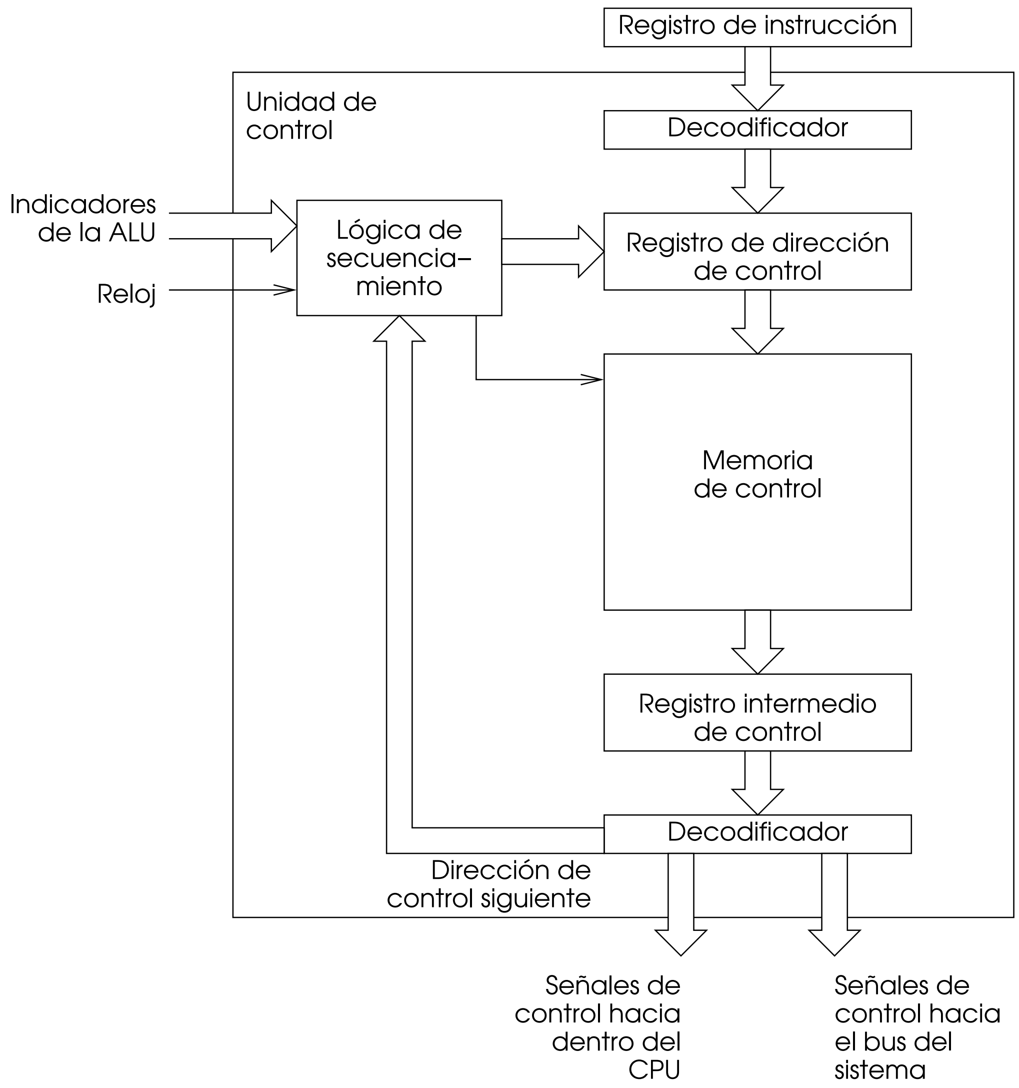
Decisiones de la unidad lógica de secuenciamiento

La carga de una nueva dirección en el registro de dirección de control, depende del valor de los indicadores y del registro intermedio de control, pudiéndose tomar alguna de las siguientes acciones:

- *Captar la microinstrucción siguiente*. Se suma uno al registro de dirección de control.
- *Saltar a una nueva rutina según indica una microinstrucción de salto*. El campo de dirección del registro intermedio de control se carga en el registro de dirección control.
- *Saltar a la rutina de una instrucción máquina*. Se carga el registro de dirección de control en función del código de operación almacenado en IR.

El decodificador superior que se muestra en la siguiente figura traduce el código de operación de IR en una dirección de memoria de control.

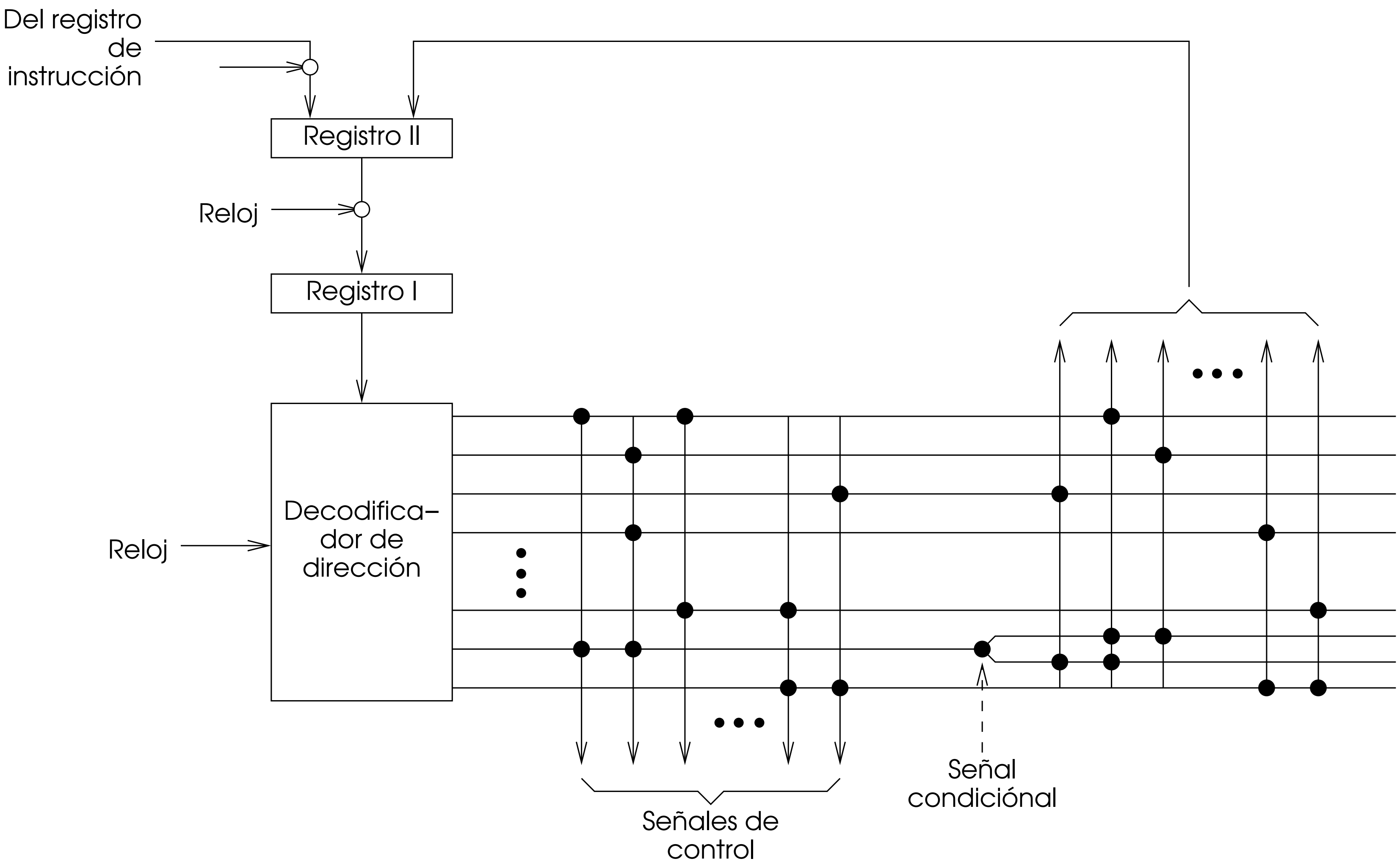
Funcionamiento de una unidad de control microprogramada



Control de Wilkes

Wilkes estaba interesado en el desarrollo de una aproximación sistemática al diseño de una unidad de control, la configuración que propuso se muestra en la siguiente figura. El núcleo del sistema es una matriz parcialmente llena de diodos. Durante un ciclo máquina, se activa una fila de la matriz mediante un pulso, que produce señales en aquellos puntos en los que un diodo está presente. La primera parte de la fila genera las señales de control que gobiernan el funcionamiento del CPU. La segunda parte genera la dirección de la fila que será seleccionada mediante un pulso en el siguiente ciclo máquina, con lo que cada fila de la matriz es una microinstrucción, y el trazado de la matriz es la memoria de control.

Unidad de control microprogramada de Wilkes



Primer diseño de una CPU microprogramada

Después de proponer su esquema Wilkes da un ejemplo de su utilización. La CPU de la máquina hipotética incluye los siguientes registros:

- A Multiplicando.
- B Acumulador (mitad menos significativa).
- C Acumulador (mitad más signficativa).
- D Registro de desplazamiento.

Hay tres registros y dos indicadores de 1 bit accesibles sólo a la unidad de control, y son:

- E Registro de dirección de memoria (MAR) o almacenamiento temporal.
- F Contador de programa.
- G Como registro temporal, usado en cálculos.

Secuenciamiento de microinstrucciones

Las dos tareas básicas realizadas por una unidad de control microprogramada son:

- *El secuenciamiento de microinstrucciones*. Obtener la siguiente microinstrucción de la memoria de control.
- *La ejecución de microinstrucciones*. Generar las señales de control necesarias para ejecutar la microinstrucción.

Las dos cuestiones involucradas en el diseño de una técnica de secuenciamiento de microinstrucciones son: el tamaño de la microinstrucción y el tiempo de generación de la dirección. Al minimizar el tamaño de la memoria de control se reduce su costo. Se desea ejecutar las microinstrucciones tan rápido como sea posible, la dirección de la siguiente microinstrucción a ejecutar esta en una de las siguientes situaciones:

- Determinada por el registro de instrucción.
- Siguiente dirección secuencial.
- Salto.

La primera situación tiene lugar sólo una vez por ciclo de instrucción, después de la captación de la instrución. La segunda es la más común, pero no se puede optimizar. Los saltos son un una parte necesaria del microprograma.

Técnicas de secuenciamiento

A partir de la microinstrucción en curso, los indicadores de condición y el contenido del registro de instrucción, se tiene que generar una dirección de la memoria de control para la siguiente microinstrucción. Para el formato de la información de dirección de la microinstrucción se han usado una gran varidedad de técnicas, que se pueden agrupar en las siguientes categorías:

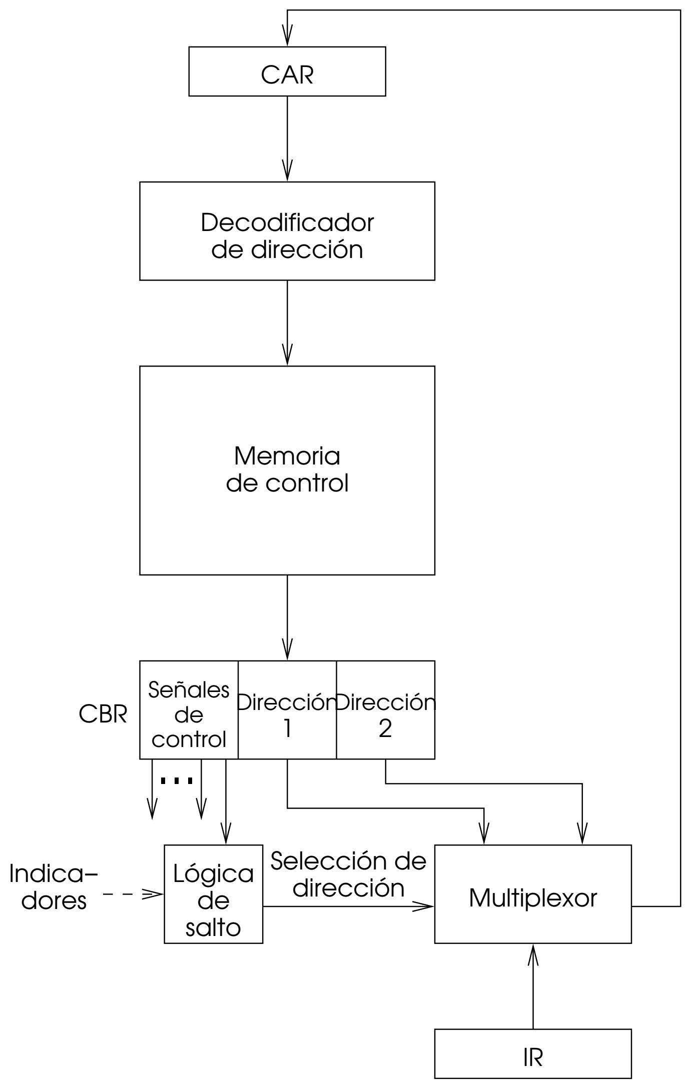
- Dos campos de dirección.
- Un único campo de dirección.
- Formato variable.

Técnicas de secuenciamiento – Dos campos de dirección

La aproximación más simple es tener dos campos de dirección en cada microinstrucción. En la siguiente figura se indica como se usa esta información. El multiplexor sirve de destino de los dos campos de dirección y del registro de instrucción. Usando la entrada de selección de dirección, el multiplexor transmite el códgo de operación o una de las dos direcciones al registro de dirección de control (CAR). El CAR después se decodifica para producir la dirección de la siguiente microinstrucción

Aunque la aproximación es sencilla necesita más bits pro microinstrucción que las otras técnicas.

Lógica de control de salto, dos campos de dirección



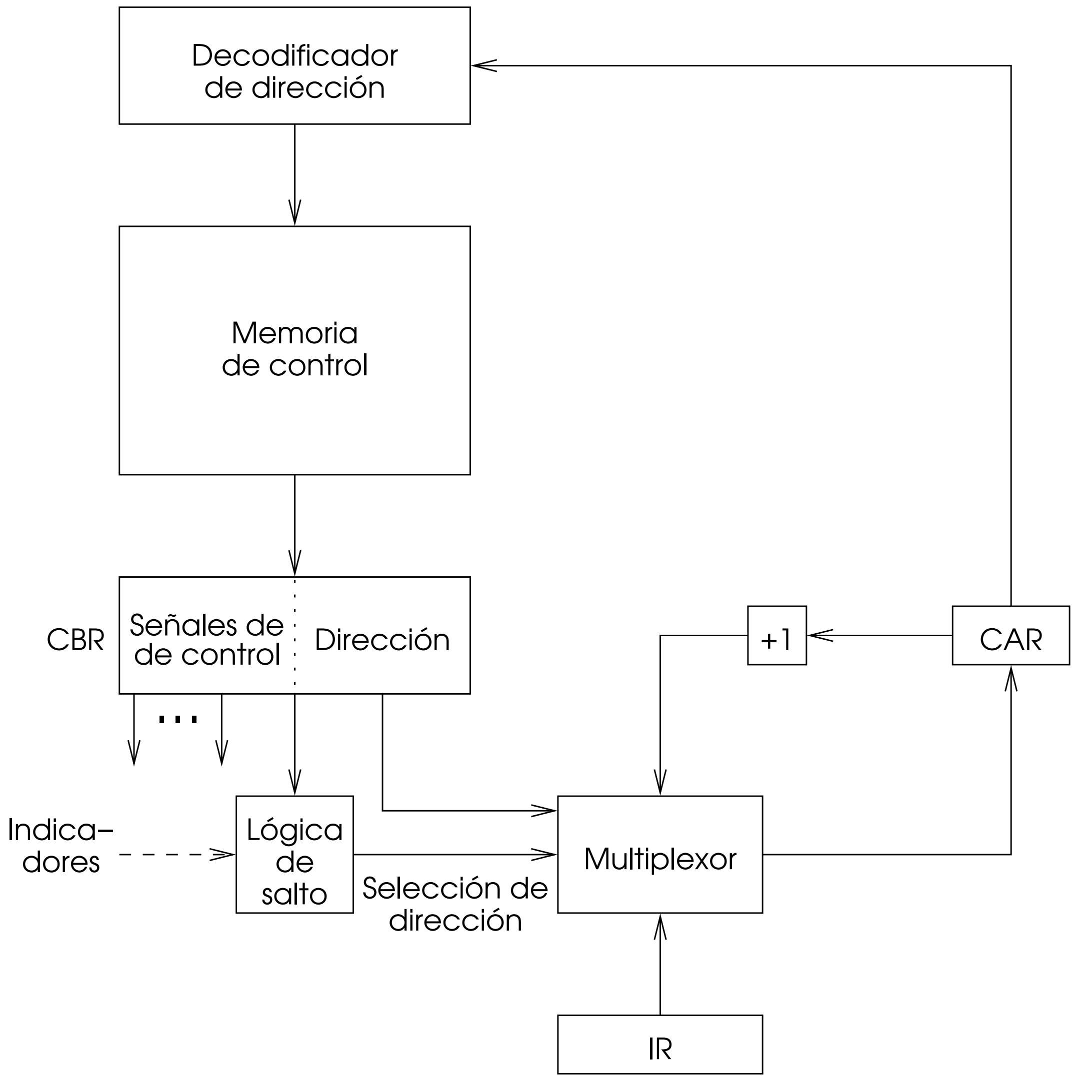
Técnicas de secuenciamiento – Un único campo de dirección

Utiliza lógica adicional para conseguir algún ahorro en los bits por microinstrucción, usando un único campo de dirección, ver siguiente figura. Las opciones para la dirección siguiente son:

- Campo de dirección.
- Código del registro de instrucción.
- Siguiente dirección secuencial.

Las señales de selección de dirección determinan que opción se escoge, con lo que se reduce el nmero de campos de dirección a uno, aunque a menudo no se usa.

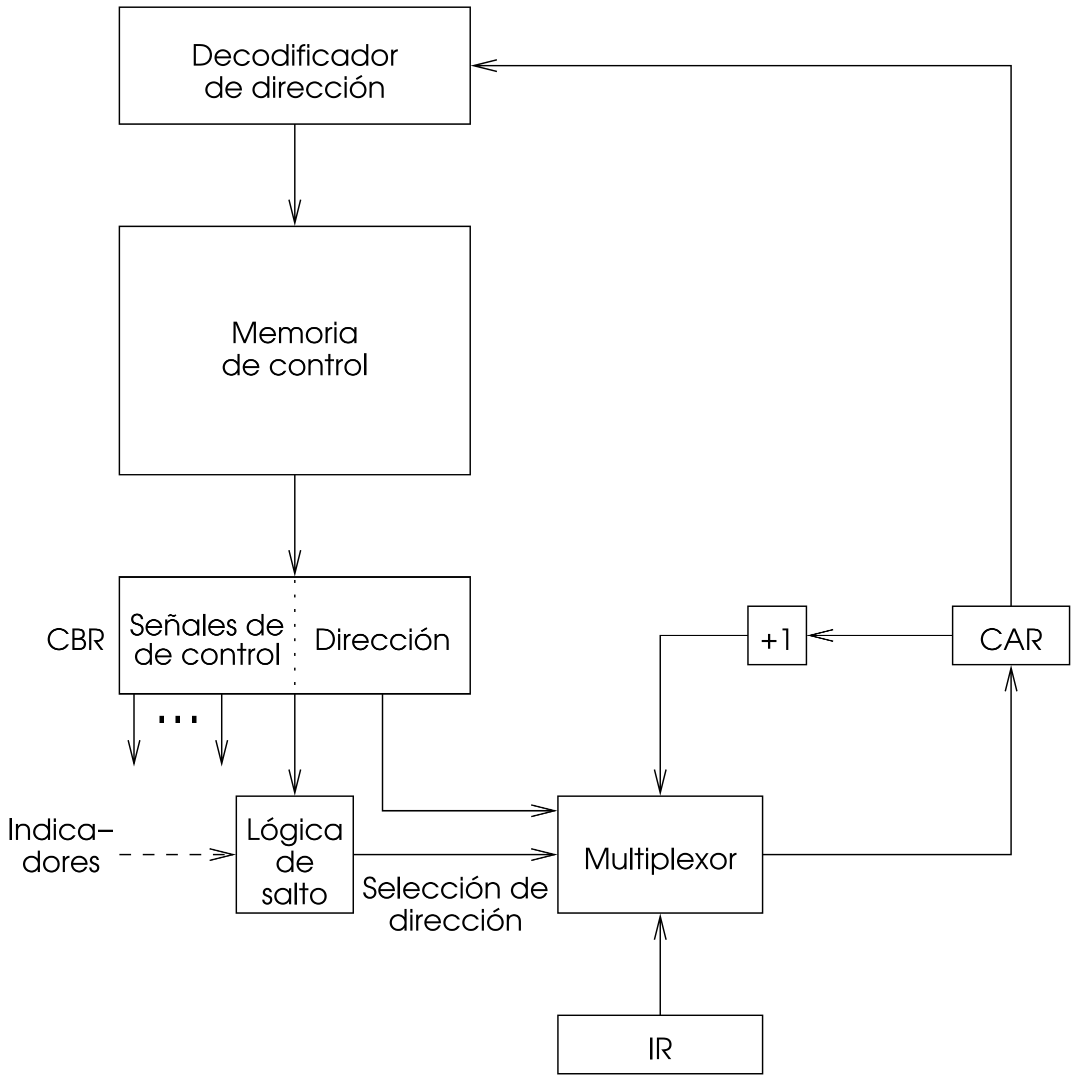
Lógica de control de salto, un único campo de dirección



Técnicas de secuenciamiento – Formato variable

Se usan dos formatos de microinstrucción, como se ve en la siguiente figura. Un bit designa el formato que se utilizará, los demás bits se usan para activar señales de control. En el otro formato, algunos bits controlan el módulo de lógica de salto, y los bits restantes suministran la dirección. En el primer formato, la dirección siguiente es la siguiente dirección secuencial o una dirección derivada del registro de instrucción. En el segundo formato, se indica un salto condicional o incondicional.

Lógica de control de salto, formato variable



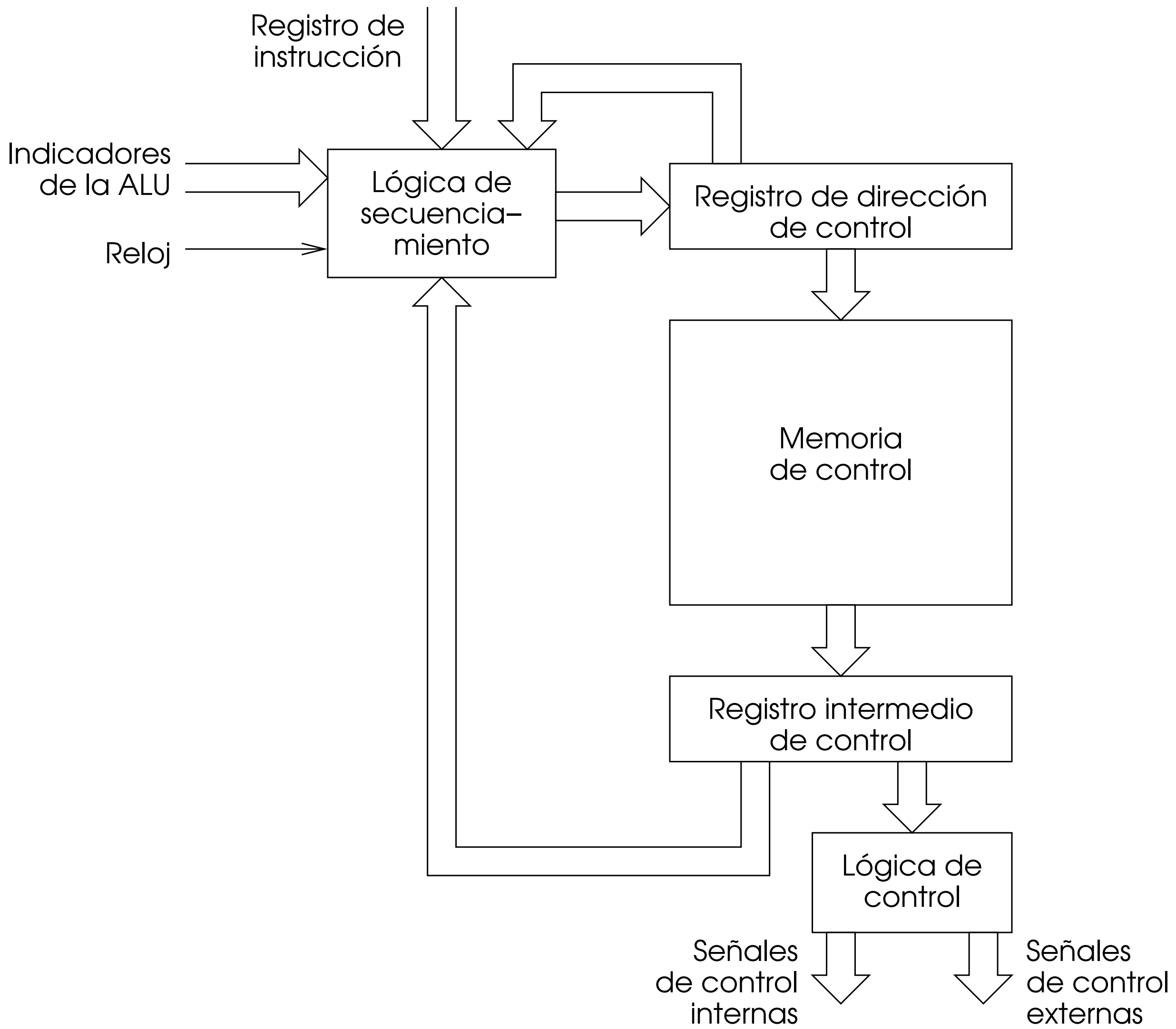
Ejecución de microinstrucciones

El ciclo de microinstrucción es el evento básico de una CPU microprogramada. Cada ciclo se compone de dos partes: captación y ejecución.

El resultado de la ejecución es la generación de señales de control para puntos internos del CPU, o al bus de control externo o a otras interfaces externas.

Con la descripción precedente se tiene la organización de la unidad de control mostrada en la siguiente figura.

Organización de la unidad de control



El espectro de las microinstrucciones

Características	
Microinstrucción no codificada	Microinstrucción muy codificada
Muchos bits	Pocos bits
Visión detallada del hardware	Visión global del hardware
Difícil de programar	Fácil de programar
Concurrencia explotada completamente	Concurrencia no explotada completamente
Poca o ninguna lógica de control	Lógica de control compleja
Ejecución rápida	Ejecución lenta
Optimización de las prestaciones	Optimización de la programación
Terminología	
Desempaquetada	Empaquetada
Horizontal	Vertical
“Hard”	“Soft”