

UNIDAD DE CONTROL MICROPROGRAMADA.

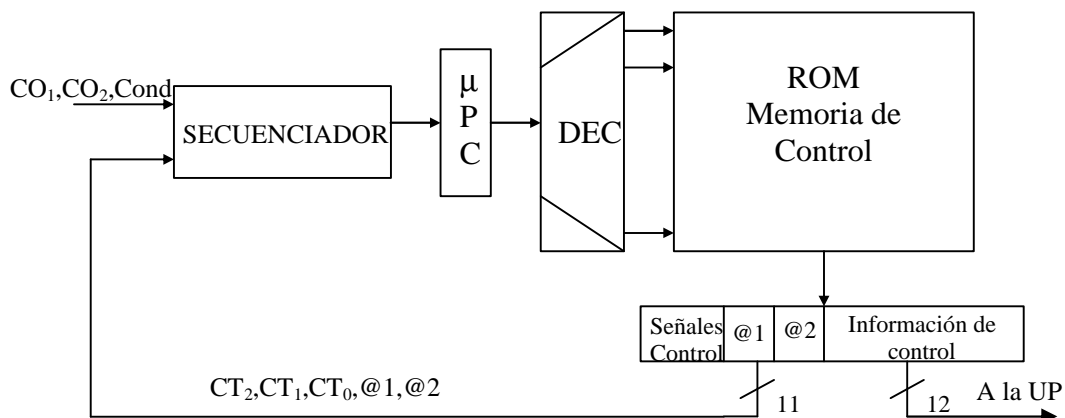
Como se ha visto en el capítulo anterior, la UC puede ser diseñada a partir de técnicas de síntesis de sistemas secuenciales y la utilización de una ROM para guardar las señales de control que se enviarán en cada estado a la UP. Sin embargo, cuando el repertorio de instrucciones es muy elevado, el número de estados aumenta considerablemente y el diseño de la unidad de control cableada puede ser muy complejo. Además, una vez diseñada la UC cableada si se quiere añadir una nueva instrucción se debe volver a diseñar de nuevo toda la UC.

Para solucionar este problema, M.V. Wilkes propuso en 1951 un método de diseño de la UC ordenado y sistemático, que conseguía reducir la complejidad del diseño cableado. Este hecho dio origen, en años sucesivos, a la aparición de la UC microprogramada.

La UC microprogramada es un circuito lógico sencillo que permite realizar la ejecución en secuencia de un conjunto de instrucciones muy elementales llamadas microinstrucciones (no confundir con las instrucciones propias del lenguaje máquina). Cada microinstrucción contiene las señales de control que se enviarán a la UP, junto a una serie de bits que permitirán seguir el secuenciamiento.

La UC estará formada por:

- Memoria de Control: contiene las microinstrucciones
- Secuenciador: es el encargado de mantener el orden correcto en la ejecución de las microinstrucciones. Genera la dirección de la siguiente microinstrucción a ejecutar y la guarda en μPC
- μPC (microcontador de programa): registro que almacena la dirección de la siguiente microinstrucción que se va a ejecutar. Esta dirección se decodificará en DEC y seleccionará una posición de la memoria de control



FORMATO DE LAS MICROINSTRUCCIONES.

A una secuencia de microinstrucciones se la conoce como microprograma y será traducido a lógica de control para dar las órdenes oportunas a la UP. Así, se puede escoger el número de campos que debe tener una microinstrucción, junto con la interpretación que tendrá cada uno de los bits que componen cada campo. Una condición esencial que se debe tener en cuenta a la hora de diseñar el formato es que éste debe hacer sencilla la interpretación del microprograma

y permitir una fácil programación del mismo. Por tanto, el formato de las microinstrucciones que forma la MR estará compuesto por:

- Campo de información de control: contendrá las señales de control que se enviarán a la UP.
- Campo de dirección: puede contener una o dos direcciones, según sea el diseño, que indicará donde se encuentra la próxima microinstrucción a ejecutar dentro de la memoria de control.
- Campo de señales de control: entran al secuenciador, junto a las señales que vienen de la UP (CO, Cond), para generar las señales de selección de dirección.

Señales Control	Dirección1	Dirección2	Información de control
-----------------	------------	------------	------------------------

SECUENCIA DE MICROINSTRUCCIONES DE LA MR.

A partir del grafo de estados y de la tabla de salidas de la figura 4.11. podemos establecer la siguiente secuencia:

- 1- FETCH: será la primera en ejecutarse y siempre pasará a la fase de decodificación.
- 2- DECO: una vez conocido el código de operación de la instrucción en curso, se tomará la decisión de saltar hacia LOAD, STORE, BRANCH o ARIT. En el caso de que el código de operación sea 10 y el bit Cond sea 0, se pasará a la fase de FETCH.
- 3- LOAD: enviará a la UP las señales necesarias para ejecutar la instrucción LOAD y saltará a FETCH.
- 4- STORE: enviará a la UP las señales necesarias para ejecutar la instrucción STORE y saltará a FETCH.
- 5- BRANCH: enviará a la UP las señales necesarias para ejecutar la instrucción BRANCH y saltará a DECO.
- 6- ARIT: enviará a la UP las señales necesarias para ejecutar la instrucción A-L correspondiente y saltará a DECO.

En la figura 4.12 podemos ver el contenido de la memoria de control.

@ μ i	Control	@1	@2	Ld_IR	Ld_PC	Ld_R@	Ld_RA	Ld_RZ	Ld_RN	ERd	L/E	PC/@	CRf1	CRf0	OPERAR	Comentarios
0	001	0001	0001	1	1	0	0	0	0	0	0	0	X	X	X	; fetch. Ir a @2=1
1	101	0010	0000	0	0	1	1	0	0	0	0	X	0	1	X	; si CO=10 y Cond=0 ir a @2=0
2	100	0011	0110	0	0	1	1	0	0	0	0	X	0	1	X	; si CO=1X ir a @2=6
3	010	0100	0101	0	0	1	1	0	0	0	0	X	0	1	X	; si CO=01 ir a @2=5
4	001	0000	0000	0	0	0	0	1	1	1	0	1	X	X	0	;LOAD. Ir a @2=0
5	001	0000	0000	0	0	0	0	0	0	0	1	1	0	0	X	; STORE. Ir a @2=0
6	110	0111	1000	0	0	1	1	0	0	0	0	X	0	1	X	; si CO=11 ir a @2=8
7	001	0001	0001	1	1	0	0	0	0	0	0	1	X	X	X	; BRANCH. Ir a @2=1
8	001	0001	0001	1	1	0	0	1	1	1	0	0	1	0	1	;A-L. Ir a @2=1

Figura 4.12

@ μ i: Dirección de las microinstrucciones en la memoria de Control

Control: campo de señales de control.

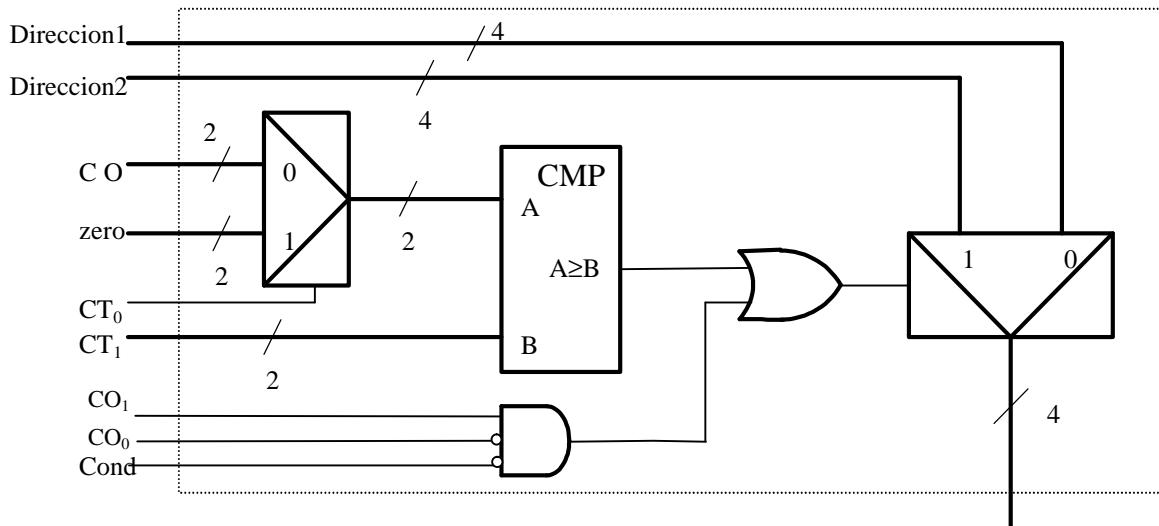
@1 y @2: campos de dirección 1 y 2.

Ld_IR...OPERAR: campo información.

Comentarios: no pertenecen a la microinstrucción.

EL SECUENCIADOR.

El secuenciador es un circuito combinacional que permite generar la dirección de la siguiente microinstrucción que se va a ejecutar, a partir de la microinstrucción en curso, el indicador de condición y el Código de operación del Registro de Instrucciones. La lógica de control es la siguiente:



El circuito está formado por:

Un multiplexor que permite seleccionar entre el código de operación (CO) de la instrucción (IR_{15} , IR_{14}) y un valor zero (2 bits siempre a cero), utilizando como entrada de selección el bit de menor peso del campo señales de control (CT_0). El valor seleccionado entra a un comparador (CMP) que pone un 1 en su salida si la comparación entre la salida del multiplexor (CO o zero) es mayor o igual que los bits de mayor peso del campo señales de control (CT_2, CT_1). Las señales de control se han definido de tal forma que una vez realizada la comparación, se puede seleccionar una de las dos direcciones donde se encuentra la siguiente microinstrucción a ejecutar. Dicha selección la realiza un multiplexor que tiene como entrada de selección la salida del comparador. El bit Cond, resultado de la evaluación de salto, sólo se tendrá en cuenta cuando la instrucción a ejecutar sea un salto ($CT_{2,1} = CO = 10$). Por este motivo se ha añadido un circuito lógico (puertas NOT, AND y OR) que permite contemplar este caso particular.

OPTIMIZACIÓN DE LA UC MICROPROGRAMADA.

En la tabla 4.12 se puede observar que el campo dirección1 siempre contiene la dirección de la microinstrucción siguiente, excepto en el caso de final de instrucción. Por ello, podemos suprimir este campo y realizar un mecanismo que nos permita incrementar en 1 la dirección actual, de tal forma que el secuenciador escoja entre la dirección proporcionada por el campo dirección2 o la dirección consecutiva a la de la microinstrucción que se está ejecutando. La modificación será la siguiente:

De esta forma, la memoria de control pasará de tener 23 bits a tener 19 por posición y su contenido es el siguiente:

@µi	Control	@2	Ld_IR	Ld_PC	Ld_R@	Ld_RA	Ld_RZ	Ld_RN	Erd	L/E	PC/@	CRf1	CRf0	OPERAR
0	001	0001	1	1	0	0	0	0	0	0	0	X	X	X
1	101	0000	0	0	1	1	0	0	0	0	X	0	1	X
2	100	0110	0	0	1	1	0	0	0	0	X	0	1	X
3	010	0101	0	0	1	1	0	0	0	0	X	0	1	X
4	001	0000	0	0	0	0	1	1	1	0	1	X	X	0
5	001	0000	0	0	0	0	0	0	0	1	1	0	0	X
6	110	1000	0	0	1	1	0	0	0	0	X	0	1	X
7	001	0001	1	1	0	0	0	0	0	0	1	X	X	X
8	001	0001	1	1	0	0	1	1	1	0	0	1	0	1