

5.- Diseño de transferencia entre registros

0.- Introducción

1.- Niveles de diseño de un sistema digital: diseño jerárquico

2.- Nivel de transferencia entre registros.

3.- Estructura de un sistema digital

4.- Diagramas de máquinas de estados algorítmicas

5.- Ejemplo de diseño: multiplicador binario

0.- Introducción

Niveles
procedimiento
de diseño
sistema
digital

- De puertas lógicas → Unidad información = bit
- De registros → registros de la máquina y sus interconexiones
- De procesador → CPU, procesador EIS, memorias, etc.
Unidad información = bloque de palabras

Descripción
sistema
digital
PARTES

- Unidad de procesamiento → Donde se almacenan y transforman los datos
- Unidad de control → genera las secuencias de señales de control de acuerdo al algoritmo de transferencia de registros.

Elementos de
la unidad
de procesamiento

- Reg. almacenamiento
- Operadores ALU
- Red de interconexión
- Puntos de control
- Señales de condición

- Alternativas para diseño de la unidad de control
- Elementos memoria tipo D
 - Reg. de secuencia y decodificador (NO)
 - Un elemento de memoria por estado
 - Un registro de estado y una mem. ROM
 - Contador y decodificador (NO)
 - Un registro de estado y un array lógico prog. (PLIS)

1.- Niveles de diseño de un sistema digital: diseño jerárquico

Diseño descendente o estructurado = aquel que empieza a nivel de procesador, pasa luego este nivel a nivel de registros y por último, éste a nivel de puertas lógicas.

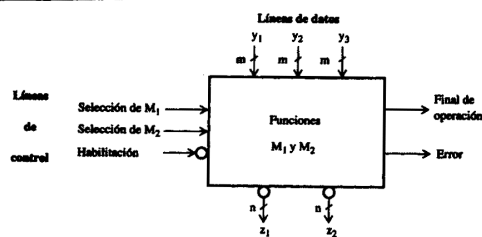
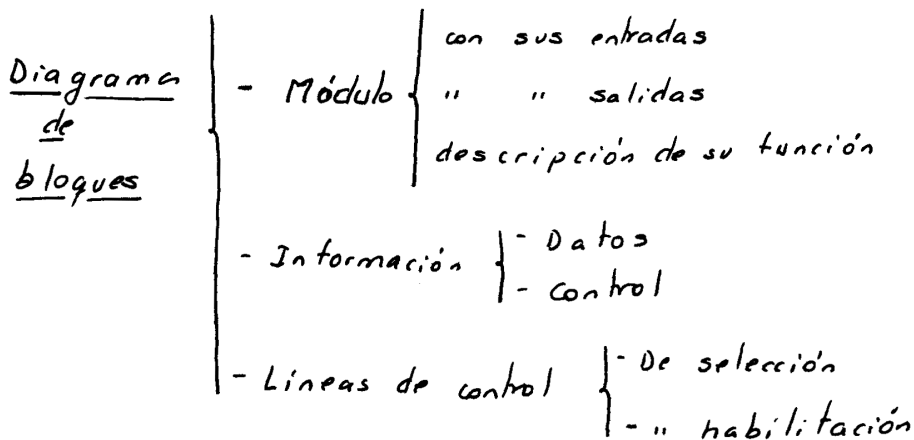
- Características de un buen diseño
- Independencia de componentes
 - Modularidad del diseño

2.- Nivel de transferencia entre registros

Unidad información básica = palabra

- Clases de palabras
- Datos
 - Control

Representación → Diagrama de bloques



Representación de un diagrama de bloques a nivel de registro

3.-Estructura de un sistema digital

Comportamiento de un sistema digital

- Conjunto de operaciones O_1, O_2, \dots, O_n realizadas sobre diferentes datos externos o internos. Cada una de estas operaciones O_i se realiza mediante una secuencia de operaciones elementales de transferencias entre registros

Problema de diseño a nivel de transferencia de registros

Dados un conjunto de algoritmos, un repertorio de instrucciones y un conjunto de restricciones, obtener el circuito que realiza las funciones deseadas a nivel de registros

Componentes
de un sistema
digital

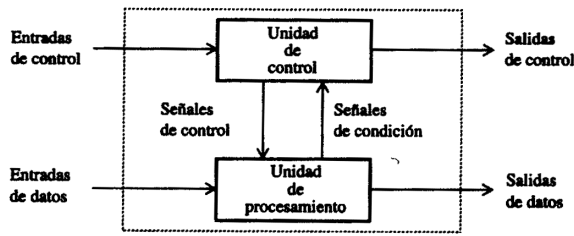
- Un conjunto de posiciones de memoria
- Interfaz con el exterior
- Conjunto dispositivos lógicos que realizan las funciones lógicas y aritméticas de transformación
- Conjunto de camino de datos
- Elemento o unidad de control.

puntos de control → Se emplean para controlar la selección de los operadores y la carga de los registros.

Modelo de Glushkov

Modelo
de
Glushkov

- Unidad de procesamiento
 - Elementos memoria
 - Interfaz con el exterior
 - Operadores aritméticos y lógicos
 - Caminos transferencia datos
- Unidad de control → Máquina secuencial → recibe de la unidad de procesamiento una indicación sobre estado de las operaciones y controla la secuencia de ejecución de las micro-operaciones



Modelo de Glushkov de un sistema digital

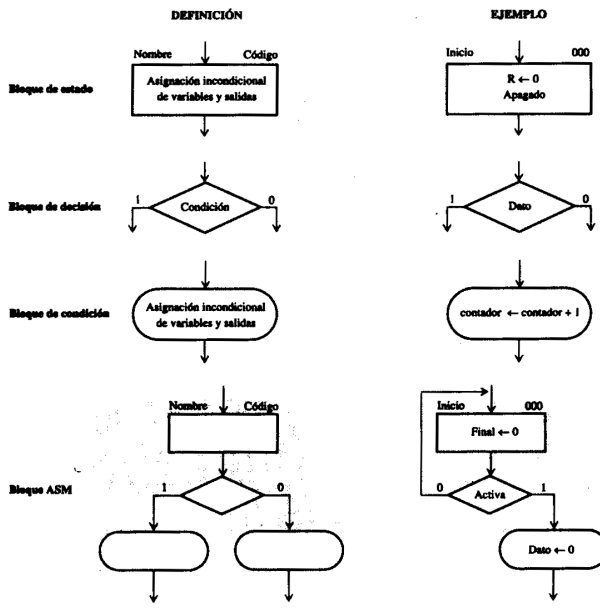
- La U.C.
y procesamiento
se define con
- Conjunto de componentes
 - Estructura interconexión
 - Función de los componentes interconectados

- Componentes
unidad procesamiento
- Conjunto de registros
 - " " primitivas o recursos de cálculo
 - Red de interconexión
 - Puntos de control
 - Señales de condición o estado.

4.- Diagramas de máquinas de estados algorítmicas (ASM)

Es una forma gráfica de especificar el modelo de un sistema digital.

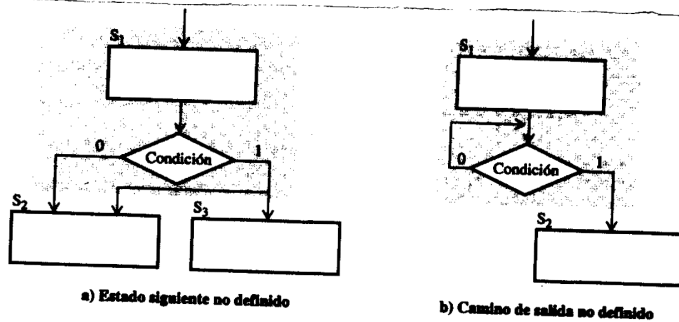
- Componentes
básicos
- Bloque de estado
 - Bloque de decisión
 - " " condición → Asignación de variables o estado
 - " ASM → Conjunto de bloques



Elementos de un diagrama ASM

Reglas de un diagrama ASM

- Un diagrama debe definir un único próximo estado para cada estado presente y conjunto de condiciones
- Todo camino definido por el conjunto de bloques de condición debe llevar necesariamente a otro estado



Diagramas ASM incorrectos

Recursos
cálculo

- Reg. n bits para multiplicando (m) ← carga paralela
- " " " " multiplicador (m) → almacen resultado bajo
- " " " " A para almacen resultado alto con carga paralelo y desplaz. dcha
- Sumador paralelo de $2n$ bits
- Contador (descontador) para n bits multiplicador
- Detector de n "0" de bits de m
- Elemento memoria 1 bit para carry

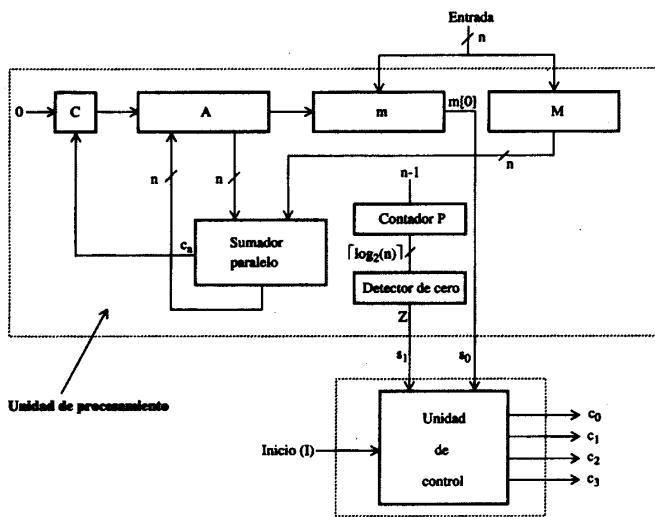
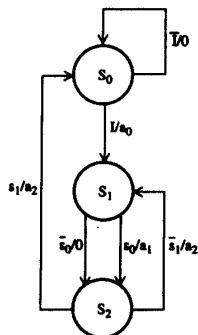
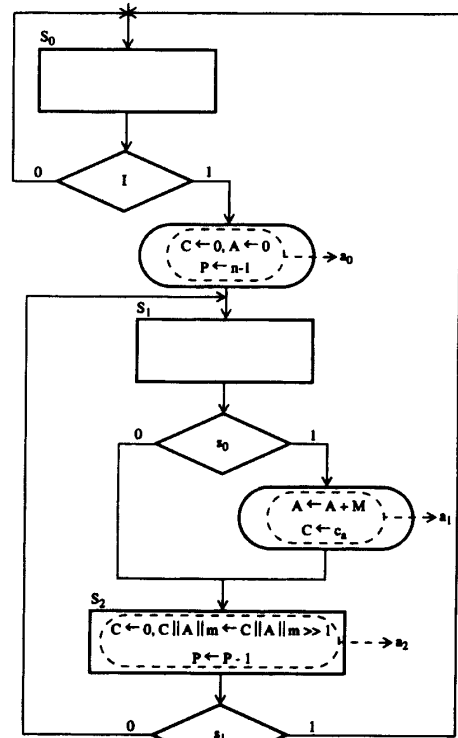


Diagrama de bloques del multiplicador binario con el algoritmo de lapiz y papel mejorado



b) Diagrama de estados de Huffman-Mealy



a) Diagrama ASM

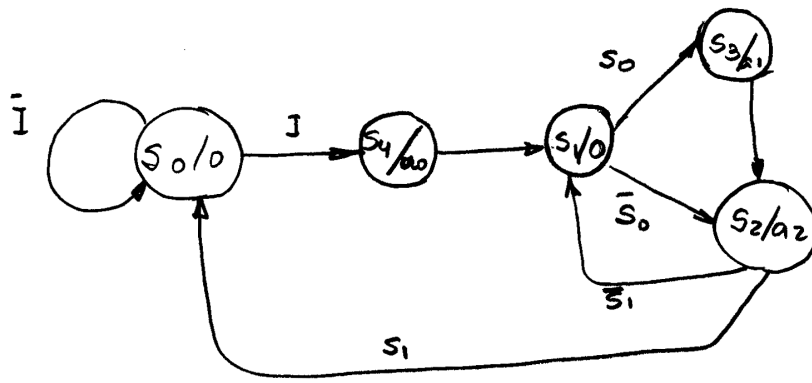
Diagrama ASM del multiplicador binario

s_0 → bit LSB de m .
multiplicando $\} 1$

s_1 → bit de valor de P
 $P \neq 0 \rightarrow s_1 = 0$
 $P = 0 \rightarrow s_1 = 1$

$a_0, a_1, a_2 \rightarrow$ Acciones de los estados
 $0, 1, 2$ respectivamente

Diagrama de estados de Moore



- $a_0 \left\{ \begin{array}{l} C \leftarrow 0 \\ A \leftarrow 0 \\ P \leftarrow n-1 \end{array} \right.$
- $a_1 \left\{ \begin{array}{l} A \leftarrow A+M \\ C \leftarrow c_a \end{array} \right.$
- $a_2 \left\{ \begin{array}{l} C \leftarrow 0 \\ C \parallel A \parallel m \leftarrow C \parallel A \parallel m \gg 1 \\ P \leftarrow P-1 \end{array} \right.$

Unidad de control con lógica cableada

Componente	Microoperación	Señal de control	Ecuación lógica
Registro A	$A \leftarrow 0$ $A \leftarrow A+M$ $C \parallel A \parallel m \leftarrow C \parallel A \parallel m \gg 1$	Inicializar Sumar_Cargar Desplazar_Dec	$c_0 = S_0 I$ $c_1 = S_1 S_0$ $c_2 = S_2$
Registro M	$M \leftarrow \text{Entrada}$	Cargar_M	
Elemento de memoria C	$C \leftarrow 0$ $C \leftarrow c_a$	Borrar_C Sumar_Cargar	$c_3 = S_0 I + S_2$ $c_1 = S_1 S_0$
Registro m	$m \leftarrow \text{Entrada}$ $C \parallel A \parallel m \leftarrow C \parallel A \parallel m \gg 1$	Cargar_m Desplazar_Dec	$c_2 = S_2$
Contador P	$P \leftarrow n-1$ $P \leftarrow P-1$	Inicializar Desplazar_Dec	$c_0 = S_0 I$ $c_2 = S_2$

Tabla de estados para la unidad de control del multiplicador binario

- $c_0 \rightarrow \text{Inicializar}$
- $c_1 \rightarrow \text{Sumar-cargar}$
- $c_2 \rightarrow \text{Desplazar}$
- $c_3 \rightarrow \text{Borrar-C}$

Unidad de control con básculas D

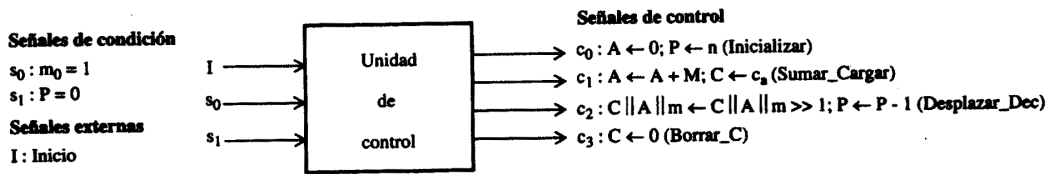


Diagrama de bloques de la unidad de control del multiplicador binario

$c_0 \rightarrow$ Inicializar \Rightarrow Estado 0 y señal de Inicialización I
 \uparrow
 a_0 $c_0 = S_0 \cdot I$

$c_1 \rightarrow$ Sumar y cargar \Rightarrow se suma y se carga en los registros cuando estamos en el estado S_1 y el bit del multiplicador es 1 ($s_0 = 1$)
 \uparrow
 a_1 $c_1 = S_1 \cdot s_0$

$c_2 \rightarrow$ Desplazar \Rightarrow se desplaza a la derecha siempre, y además en el estado S_2
 \uparrow
 a_2 $c_2 = S_2$

$c_3 \rightarrow$ Borrar carry \Rightarrow siempre al inicializar ($S_0 \cdot I$) y al desplazar (S_2)
 $c_3 = S_0 I + S_2 \Rightarrow c_3 = c_0 + c_2$

Diseño

Señales entrada

$I \rightarrow$ inicio
 $s_0 \rightarrow$ bit LSB $m = 1$
 $s_1 \rightarrow P \neq 0$

Señales salida

c_0
 c_1
 c_2
 c_3

Estados

S_0, S_1, S_2
 S_{00}, S_{10}
 \downarrow
 3 básculas
 Q_2, Q_1, Q_0

Entradas Est. Actual Est. Próximo Salidas

J	s ₁ s ₀		Q ₂ Q ₁ Q ₀			Q _{2,t+1} Q _{1,t+1} Q _{0,t+1}			c ₃ c ₂ c ₁ c ₀					
	0	x	x	0	0	0	0	0	0	0	0	0	0	0
1	x	x	0	0	0	1	0	0	0	0	0	0	1	a ₀
x	x	x	1	0	0	0	0	1	0	0	0	0	0	
x	x	0	0	0	1	0	1	0	1	1	0	0	0	a ₂
x	x	1	0	0	1	0	1	1	0	0	1	0	0	a ₁
x	x	x	0	1	1	0	1	0	1	1	0	0	0	a ₂

Para pasar una báscula del estado Q_i al Q_{i,t+1}, al ser báscula D, habrá que introducir en D_i el valor de

Q_{i,t+1} \bar{Q}_2 Q₂

		\bar{Q}_1 Q ₁ \bar{Q}_1							
		\bar{Q}_0	Q ₀	\bar{Q}_0	\bar{Q}_0	Q ₀	Q ₀	\bar{Q}_0	\bar{Q}_0
I	s ₁	s ₀	0	0	x	x	x	0	0
	s ₀	0	0	x	x	x	0	0	
I	s ₁	s ₀	0	0	x	x	x	0	0
	s ₀	0	0	x	x	x	0	0	
I	s ₁	s ₀	1	0	x	x	x	0	0
	s ₀	1	0	x	x	x	0	0	
I	s ₁	s ₀	1	0	x	x	x	0	0
	s ₀	1	0	x	x	x	0	0	

Estados 5, 6, 7 → imposibles ⇒ se puede tomar valor x
 \hookrightarrow Q₂ Q₁ Q₀ { 101, 110, 111

$$D_2 = I \bar{Q}_2 \bar{Q}_1 \bar{Q}_0$$

$$D_1 = \bar{Q}_2 Q_0$$

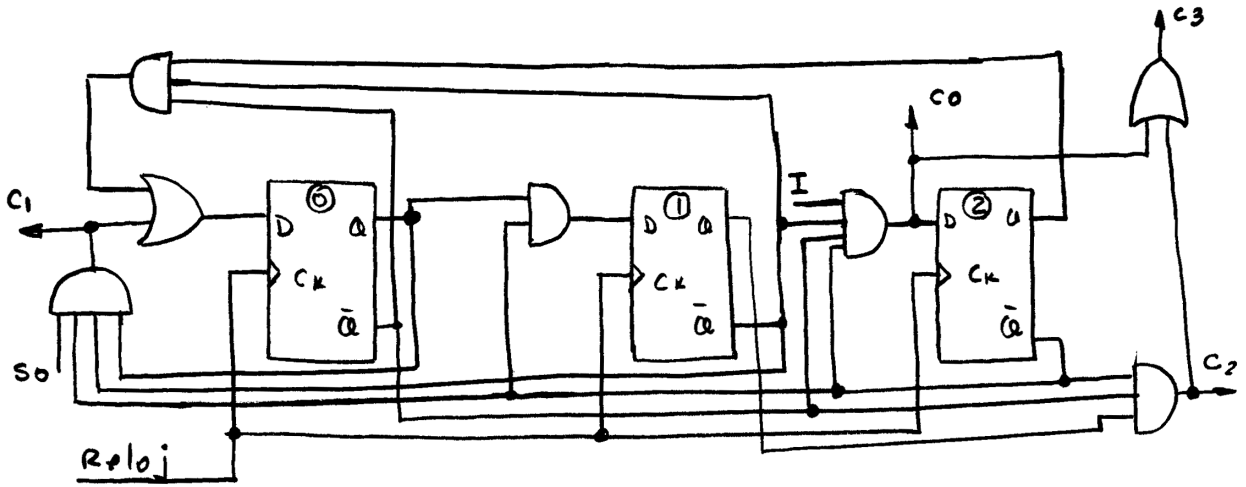
$$D_0 = Q_2 \bar{Q}_1 \bar{Q}_0 + \bar{Q}_2 \bar{Q}_1 Q_0 s_0$$

$$c_0 = s_0 \cdot I \Rightarrow I \bar{Q}_2 \bar{Q}_1 \bar{Q}_0 = D_2$$

$$c_1 = s_1 \cdot s_0 = \bar{Q}_2 \bar{Q}_1 Q_0 \cdot s_0$$

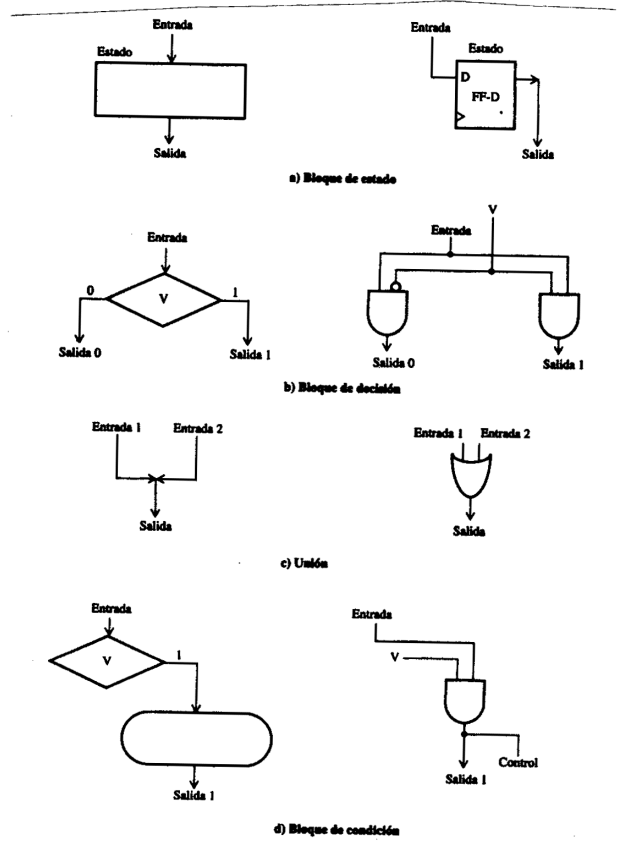
$$c_2 = s_2 = \bar{Q}_2 Q_1 \bar{Q}_0$$

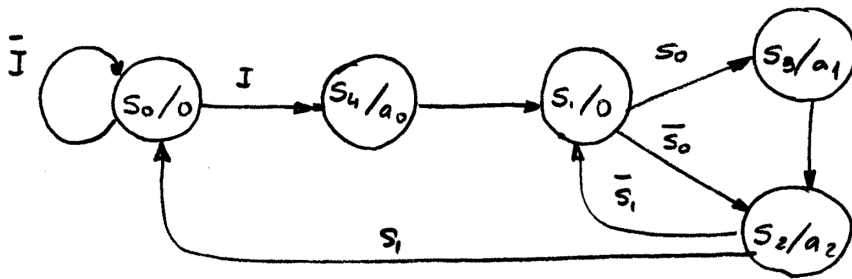
$$c_3 = c_2 + c_0$$



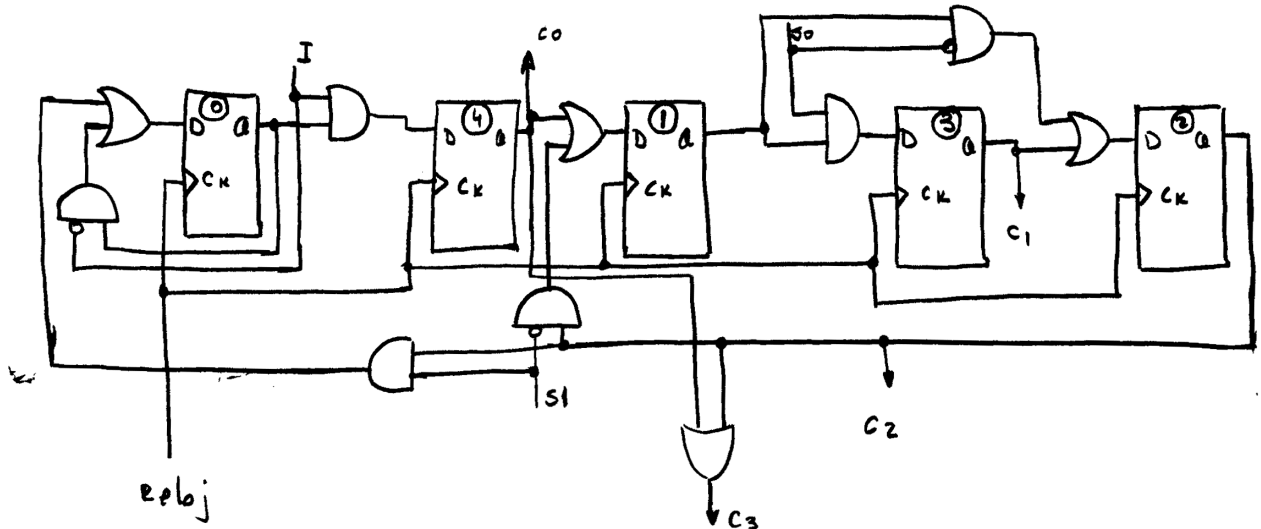
Unidad de control diseñada con una búscula por estado

Cada estado se implementa con una búscula implementando las decisiones, uniones y condiciones como se indica:





$a_0 \Rightarrow c_0$
 $a_1 \Rightarrow c_1$
 $a_2 \Rightarrow c_2, c_3$



Unidad de control diseñada con registro de estado y mem. ROM

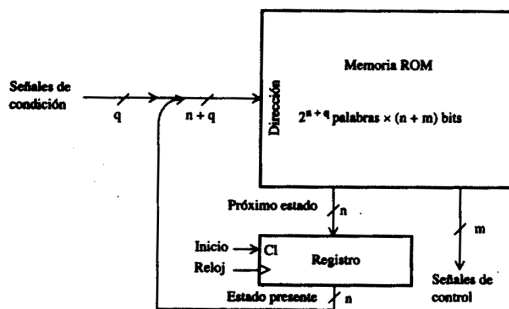


Figura 5.31: Estructura básica de un circuito secuencial con una memoria ROM

- Las var. condición, señales externas y estado presente forman parte del bus de direcciones.
- Cada palabra tiene 2 campos, uno de n bits que representa el próximo estado y otro de m bits para las señales de control

Dirección de la ROM			Contenido de la ROM				
Estado presente	i_1	i_0	Próximo estado	c_0	c_1	c_2	c_3
0	00	000	00	0000			
1	00	001	00	0000			
2	00	010	00	0000			
3	00	011	00	0000			
4	00	100	01	1001			
5	00	101	01	1001			
6	00	110	01	1001			
7	00	111	01	1001			
8	01	000	10	0000			
9	01	001	10	0100			
10	01	010	10	0000			
11	01	011	10	0100			
12	01	100	10	0000			
13	01	101	10	0100			
14	01	110	10	0000			
15	01	111	10	0100			
16	10	000	01	0011			
17	10	001	01	0011			
18	10	010	00	0011			
19	10	011	00	0011			
20	10	100	01	0011			
21	10	101	01	0011			
22	10	110	00	0011			
23	10	111	00	0011			

Contenido de la memoria ROM de la unidad de control del multiplicador

Selección por estado

Un método para reducir el tamaño de la ROM se basa en el hecho de que no se necesitan todas las variables de condición para el paso de un estado a otro.

Para seleccionar la condición de paso se utiliza un multiplexor, lo que permite que los bits destinados a variables de condición sea solo 1 en la memoria y el multiplexor sea de tantos canales como estados.

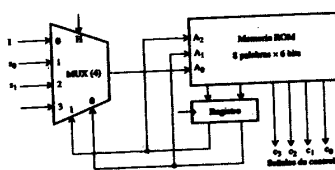
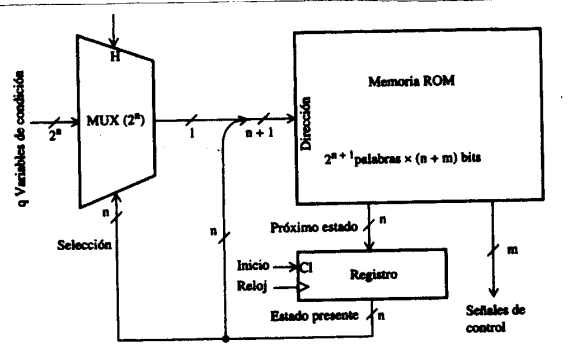


Figura 5.33: Unidad de control del multiplicador (selección por estado)



Unidad de control con una memoria ROM y un multiplexor (selección por estado)

Dirección de la ROM		Contenido de la ROM				
Estado presente	Estado de condición	Próximo estado	c_0	c_1	c_2	c_3
0	00	0(0=0)	00	0000		
1	00	1(0=1)	01	1001		
2	01	0($a_0=0$)	10	0000		
3	01	1($a_0=1$)	10	0100		
4	10	0($a_1=0$)	01	0011		
5	10	1($a_1=1$)	00	0011		

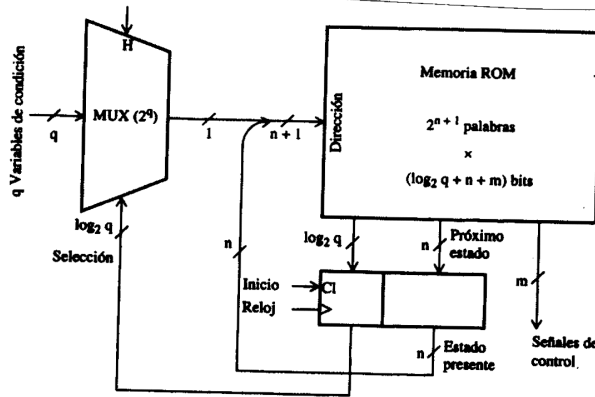
Contenido de la memoria ROM (método de selección por estado)

Selección por campo

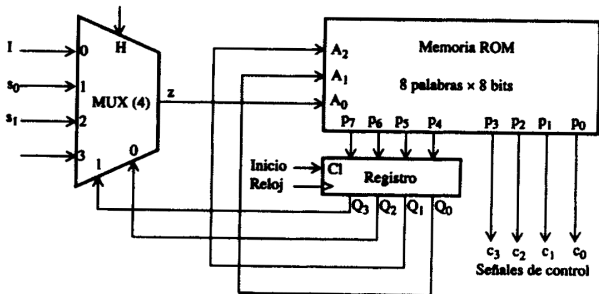
Es una variante del de selección por estado, consiste en grabar en la ROM un campo en el que se codifica el orden de la variable de condición.

Campos ROM } - Orden de var. de condición
 } - Próximo estado
 } - Señales de control

Es preferible al de selección de estado si el nº total de variables de condición es más pequeño que el nº de estados.



Unidad de control con una memoria ROM y un multiplexor (método de selección por campo)

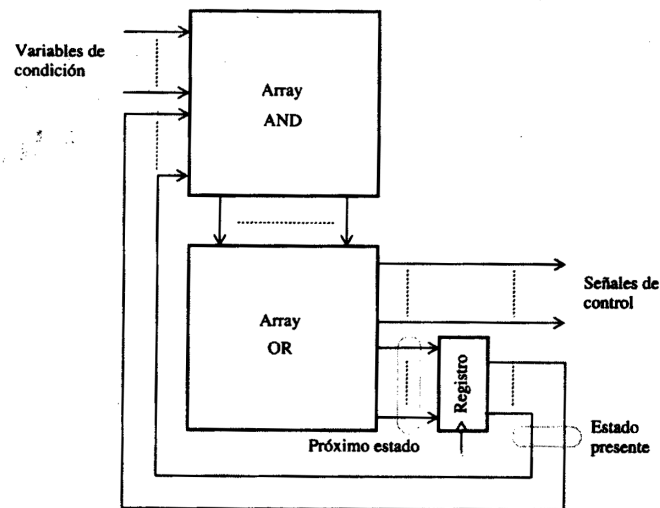


Unidad de control del multiplicador (selección por campo)

Dirección de la ROM		Contenido de la ROM							
Estado presente	Señal de condición	P7	P6	P5	P4	P3	P2	P1	P0
		c0	c1	c2	c3				
0	00	0 (I = 0)				0000			0000
1	00	1 (I = 1)				0101			1001
2	01	0 (s0 = 0)				1010			0000
3	01	1 (s0 = 1)				1010			0100
4	10	0 (s1 = 0)				0101			0011
5	10	1 (s1 = 1)				0000			0011

Contenido de la memoria ROM (método de selección por campo)

Unidad de control con un reg. de estado y un PLA



- Las variables de condición y estado presente son las entradas al Array AND
- Próximo estado y señales de control son las salidas OR

Transformación

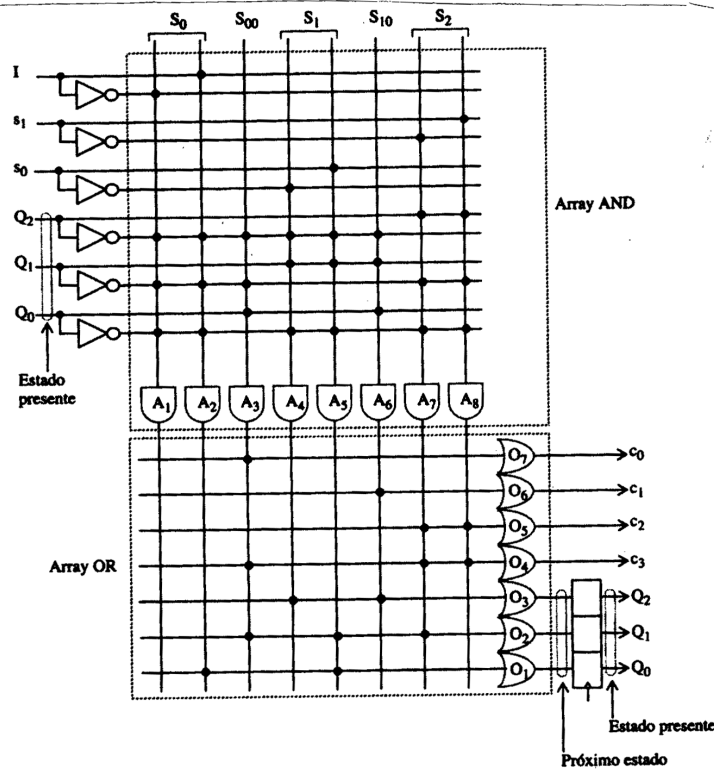
Diagrama estados

a PLA

- Una OR de salida para cada señal de control y por cada bit de estado.

- Array AND \rightarrow tantas AND como estados
Tantas entradas como variables de condición + bits de codificación de estados.

Multiplicador con PLA



Procedimiento
de diseño
a nivel de
registro

- 1^o.- Definir comportamiento del sistema digital
↳ secuencias de operaciones de transferencia entre registros.
- 2^o.- Analizar el algoritmo
- 3^o.- Construir el diagrama de bloques } Unid. procesa.
Unid. control
- 4^o.- Diseñar unidad de control
- 5^o.- Comprobar el funcionamiento del diseño resultante