

ESTRUCTURA Y TECNOLOGIA DE COMPUTADORES II

TEMA 5: *Diseño de transferencia entre registros*

**SOLUCION A LOS PROBLEMAS
PROPUESTOS EN EXAMEN**

**Curso 2007-2008
Jose Manuel Díaz Martínez
Tutor de la asignatura ETC-II**

CONTENIDO

SOLUCION PROBLEMA 5.1	3
SOLUCION PROBLEMA 5.2	3
SOLUCION PROBLEMA 5.3	4
SOLUCION PROBLEMA 5.4	13
SOLUCION PROBLEMA 5.5	13
SOLUCION PROBLEMA 5.6	13
SOLUCION PROBLEMA 5.7	14
SOLUCION PROBLEMA 5.8	14
SOLUCION PROBLEMA 5.9	15
SOLUCION PROBLEMA 5.10	15
SOLUCION PROBLEMA 5.11	16
SOLUCION PROBLEMA 5.12	16
SOLUCION PROBLEMA 5.13	16
SOLUCION PROBLEMA 5.14	16
SOLUCION PROBLEMA 5.15	17
SOLUCION PROBLEMA 5.16	17
SOLUCION PROBLEMA 5.17	17
SOLUCION PROBLEMA 5.18	17
SOLUCION PROBLEMA 5.19	19
SOLUCION PROBLEMA 5.20	19
SOLUCION PROBLEMA 5.21	20
SOLUCION PROBLEMA 5.22	22
SOLUCION PROBLEMA 5.23	23
SOLUCION PROBLEMA 5.24	24
SOLUCION PROBLEMA 5.25	25
SOLUCION PROBLEMA 5.26	26
SOLUCION PROBLEMA 5.27	26
SOLUCION PROBLEMA 5.28	27
SOLUCION PROBLEMA 5.29	28
SOLUCION PROBLEMA 5.30	28
SOLUCION PROBLEMA 5.31	30
SOLUCION PROBLEMA 5.32	31
SOLUCION PROBLEMA 5.33	32

SOLUCION PROBLEMA 5.1

- Diseñar una UC con memoria ROM con:
 - 200 estados
 - Genere 37 señales de control independientes entre si.
 - 9 señales de condición.
- En cada estado se consulta una señal de condición como máximo.
- Interesa sobre todo minimizar el tamaño del multiplexor a utilizar.

Puesto que en cada estado se consulta una señal de condición como máximo, es posible utilizar para el diseño de la Unidad de Control las siguientes configuraciones:

- a) Memoria ROM + registro + multiplexor de selección por estado.
- b) Memoria ROM + registro + multiplexor de selección por campo.

De los datos del enunciado se deduce que:

variables de estado $n=8$ ya que $2^8=256>200$ estados

señales de control : $m=37$

señales de condición : $q=9 \leq (2^h=16) \rightarrow h=4$

En la configuración a) según lo estudiado en teoría, el multiplexor utilizado tiene $2^n=2^8=256$ entradas y $n=8$ señales de selección.

En la configuración b) según lo estudiado en teoría, el multiplexor utilizado tiene $2^h=2^4=16$ entradas y $h=4$ señales de control.

Luego es en la configuración b) donde se minimiza el tamaño del multiplexor .

Afirmación I: **Falsa**

Afirmación II: **Verdadera**

SOLUCION PROBLEMA 5.2**DATOS**

- Diseñar una UC con :
 - 18 estados
 - Genere $m=30$ señales de control independientes entre si.
 - $q=4$ señales de condición.
- Utilizar memoria ROM, pero minimizando la capacidad de la misma.
- En cada estado se utiliza como máximo una condición.
- Utilizar un *multiplexor por selección por campo*

Se tienen 18 estados luego es necesario $n=5$ variables de estado para codificarlos.

La capacidad de la ROM necesario es:

$$C = 2^{n+1} \times (n + m + \log_2 q) = 2^6 \times (5 + 30 + 2) = 2^6 \text{ pal } \times 37 \text{ bit / pal}$$

El registro tiene la siguiente estructura:

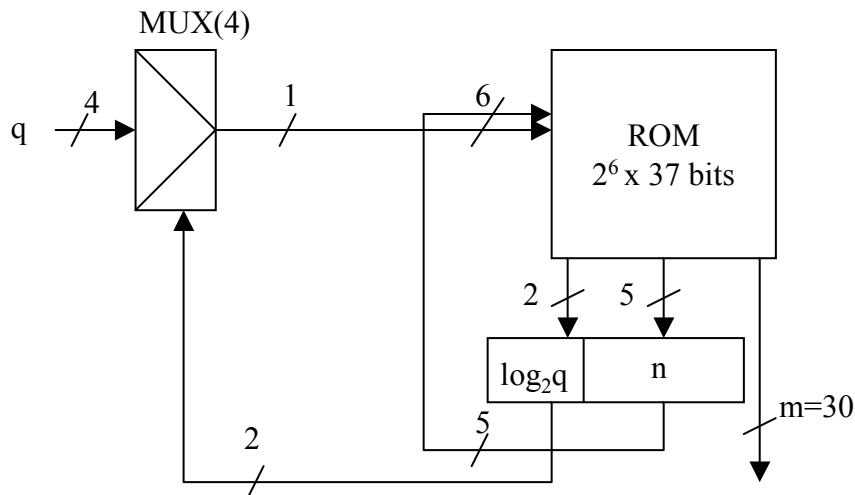
$\log_2 q$ bits	n bits
-----------------	----------

Luego debe de tener una capacidad de

$$C_R = \log_2 q + n = 2 + 5 = 7 \text{ bits}$$

El multiplexor al ser del tipo de selección por campo, tiene $q=4$ entradas y requiere de $\log_2 q=2$ señales de selección.

El esquema de la Unidad de Control implementada con los anteriores elementos sería el que se muestra en la siguiente figura:



SOLUCION PROBLEMA 5.3

a) DISEÑO DE LA UNIDAD DE PROCESAMIENTO

```

1: Declaración: A[8], B[8], Cont[3]; Bus[8]
2:   A ← Bus;
3:   B ← Bus, Cont=0;
4:   while Cont ≠ 7
5:     if A es par then
6:       A ← A - B, Cont = (Cont + 1) mod 8;
7:     else
8:       B ← B + A, Cont = (Cont - 1) mod 8;
9:     endif;
10:  endwhile;
11:  Bus ← B;
12:  Bus ← A;
13:  Parar;

```

1) **El primer paso** es conocer el tipo y número de componentes necesarios para diseñar la Unidad de Procesamiento. De la observación del algoritmo se deduce que es necesario:

- Un registro de 8 bits para el almacenamiento de A.
- Un registro de 8 bits para el almacenamiento de B.
- Un contador módulo-8.
- Un elemento de cálculo (UAL) que realice la suma y la resta de dos números binarios de 8 bits.
- Puesto que al registro A se le pueden cargar datos desde el bus ($A \leftarrow \text{Bus}$) o desde la UAL ($A \leftarrow A - B$) es necesario un multiplexor de dos salidas MUX(2).
- Puesto que al registro B se le pueden cargar datos desde el bus ($B \leftarrow \text{Bus}$) o desde la UAL ($B \leftarrow B + A$) es

necesario un multiplexor de dos salidas MUX(2).

- Como tanto el registro A como el registro B deben volcar su contenido al Bus ($\text{Bus} \leftarrow A$, $\text{Bus} \leftarrow B$), es necesaria la utilización de 2 puertas triestado de conexión unidireccional con control de 8 bits.
- Se necesita una puerta AND, que reciba como entradas las 3 salidas del contador, de tal forma que la salida de la puerta valdrá 1 cuando el contador alcance el valor 7(111).
- Finalmente se comprueba, mediante la lectura atenta del enunciado del problema, que todos los componentes que se necesitan son elementos disponibles.

2) **El segundo paso** es definir las señales de condición que serán salidas de la Unidad de Procesamiento. De la observación del algoritmo se deduce que hay dos señales de condición:

s_0 que valdrá 1 cuando el contador alcance el valor 7.

s_1 que corresponde al bit menos significativo del número A. Si $s_1=0$ el número es par sino es impar

3) **El tercer paso** es numerar *los puntos o señales de control* y especificar la función que realiza cada una de estas señales.

-Señales de control

Luego la Unidad de Procesamiento diseñada requiere de 11 señales de control (c_{10}, \dots, c_0).

c_0 selecciona Bus si vale 0, selecciona UAL si vale 1.

c_1 selecciona Bus si vale 0, selecciona UAL si vale 1.

c_2 cuando vale 1, se realiza la microoperación $\text{Bus} \leftarrow A$
 c_3 cuando vale 1, se realiza la microoperación $\text{Bus} \leftarrow B$
 c_5c_4 control del registro de desplazamiento A. (Ver enunciado)
 c_7c_6 control del registro de desplazamiento B. (Ver enunciado)
 c_8 control de la UAL (si vale 0 :suma, si vale 1 :resta)
 $c_{10}c_9$ control del contador. (Ver enunciado)

-Comentario

Las entradas E_D y E_I de los registros de desplazamiento no se utilizan y pueden conectarse a 0 o a 1 indistintamente.

- 4) El cuarto paso es dibujar la Unidad de Procesamiento. En la Figura 1 se muestra la Unidad de Procesamiento resultante:

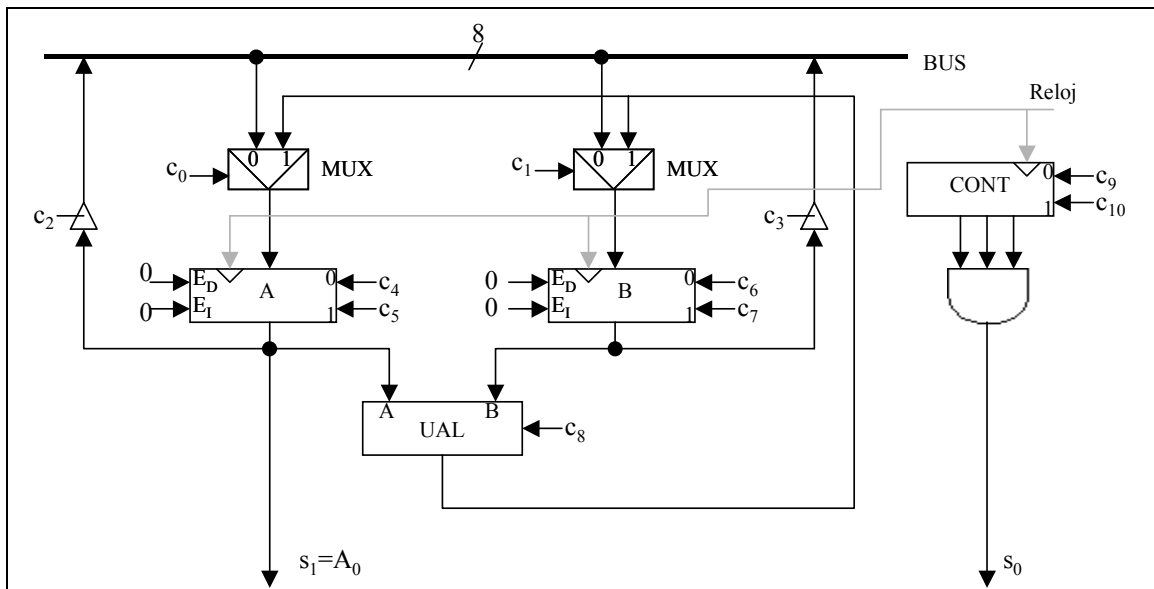


Figura 1 : Diseño de la Unidad de Procesamiento

b)DISEÑO DE LA UNIDAD DE CONTROL.

Del algoritmo se deduce que un posible diagrama de estados (existen varias posibilidades) para la Unidad de Control es el que se muestra en la Figura 2.

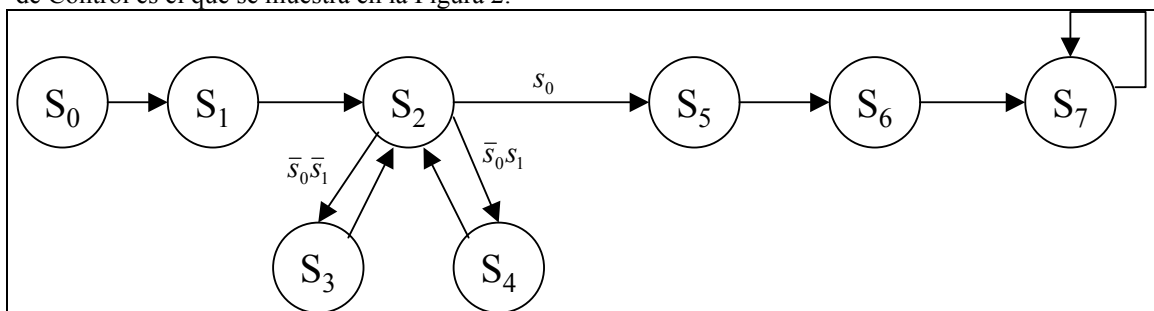


Figura 2 : Diagrama de estados de la Unidad de Control.

Se comienza en el estado S_0 , se pasa al estado S_1 y de este se pasa al estado S_2 . Este estado correspondería a comprobar si el contador ha llegado al valor 7. Ahora dependiendo de los valores de las señales de condición s_1 y s_0 , se puede pasar a S_3 ($s_1=0$ y $s_0=0$, A es par y el contador no ha llegado a 7) o a S_4 ($s_1=1$ y $s_0=0$, A es impar y el contador no ha llegado a 7) o bien a S_5 ($s_0=1$ el contador ha llegado a 7). Desde S_3 o S_4 se vuelve a S_2 , así hasta que el contador llegue a 7, en cuyo caso salta al valor S_5 . Desde S_5 pasa a S_6 , y desde S_6 pasa a S_7 donde se detiene ya que es el estado de final del algoritmo. En este diagrama se tienen 8 estados

En la Tabla 1 se muestran las microoperaciones y las señales de control asociadas a cada estado.

Estado	Microoperaciones	Señales de Control
S_0	$A \leftarrow \text{Bus}$	$\bar{c}_0 c_4 c_5$
S_1	$B \leftarrow \text{Bus},$ $\text{Cont}=0$	$\bar{c}_1 c_6 c_7$ $c_{10} c_9$
S_2	-----	-----
S_3	$A \leftarrow A-B$ $\text{cont}=(\text{cont}+1)\text{mod}8$	$c_8 c_0 c_4 c_5$ $c_{10} \bar{c}_9$
S_4	$B \leftarrow A+B$ $\text{cont}=(\text{cont}-1)\text{mod}8$	$\bar{c}_8 c_1 c_6 c_7$ $\bar{c}_{10} c_9$
S_5	$\text{Bus} \leftarrow B$	c_3
S_6	$\text{Bus} \leftarrow A$	c_2
S_7	Parar	-----

Tabla 1 : Microoperaciones y señales de control asociadas a cada estado

Una vez definido el diagrama de estados y las microoperaciones y señales de control asociadas con cada estado estamos en condiciones de diseñar la Unidad de Control mediante la técnica de *los elementos de retardo*, esta técnica también es conocida como *diseño con un elemento de memoria por estado*. Utilizando esta técnica la implementación de la Unidad de Control es inmediata. Se debe observar atentamente el diagrama de estado de la Figura 2. Los pasos a seguir son :

1) Asignar un elemento de memoria (biestable) tipo D a cada estado. Como tenemos 8 estados se utilizarán 8 elementos de memoria tipo D

2) Si un estado recibe entradas de más de un estado se debe utilizar una puerta OR. Así el estado S_2 , recibe entradas del estado S_1 , del estado S_3 y del estado S_4 por lo que debe utilizarse una puerta OR de tres entradas. Por otro lado el estado S_7 , recibe entrada del estado S_6 y del propio estado S_7 , por lo que se debe utilizar una puerta OR de 2 entradas.

3) Si un estado tiene salidas hacia más de un estado, se debe utilizar un elemento demultiplexor que tenga como señales de selección las señales de condición que regulan la transición desde ese estado hacia los otros estados. En nuestro problema se observa que en el estado S_2 , dependiendo del valor de las señales de condición s_0 y s_1 , se pasa al estado S_3 (si $s_0=0$ y $s_1=0$), al estado S_4 (si $s_0=0$ y $s_1=1$) o al estado S_5 (si $s_0=1$ independientemente del valor de s_1), por lo que se utilizará un demultiplexor con 2 señales de selección (s_0 y s_1) y 4 salidas (salida 0 hacia S_3 , salida 1 hacia S_4 y salida 2 y 3 hacia S_5).

4) La salida de cada elemento de memoria, debe llevar rotuladas las señales de control que va activar y que serán las correspondientes al estado al que está asignado el elemento de memoria. En nuestro problema las señales de control asignadas en cada estado son las que se muestran en la Tabla 1.

5) El elemento de memoria asignado al estado S_0 , debe poseer una entrada asincrónica (SET) de puesta a 1, mientras que los restantes estados deben poseer una entrada asincrónica (CLEAR) de puesta a 0. Todos los elementos de memoria tienen una entrada síncrona (CK) para la señal de reloj.

La Unidad de Control diseñada se muestra en la Figura 3. Explicaremos el funcionamiento de esta Unidad de Control. Inicialmente se activa la señal asincrónica de SET del elemento de memoria asignado al estado S_0 y la señal asincrónica de CLEAR de los restantes elementos de memoria, así el elemento de memoria asignado a S_0 contendrá un 1, mientras que los restantes elementos de memoria contendrán un 0. Este 1 se va a ir propagando por los diferentes elementos de memoria de acuerdo al diagrama de la Figura 2, en cada ciclo de reloj, de tal forma que en un instante de tiempo determinado solamente habrá un estado activo, es decir un elemento de memoria conteniendo un 1 los restantes elementos de memoria contendrán el valor 0

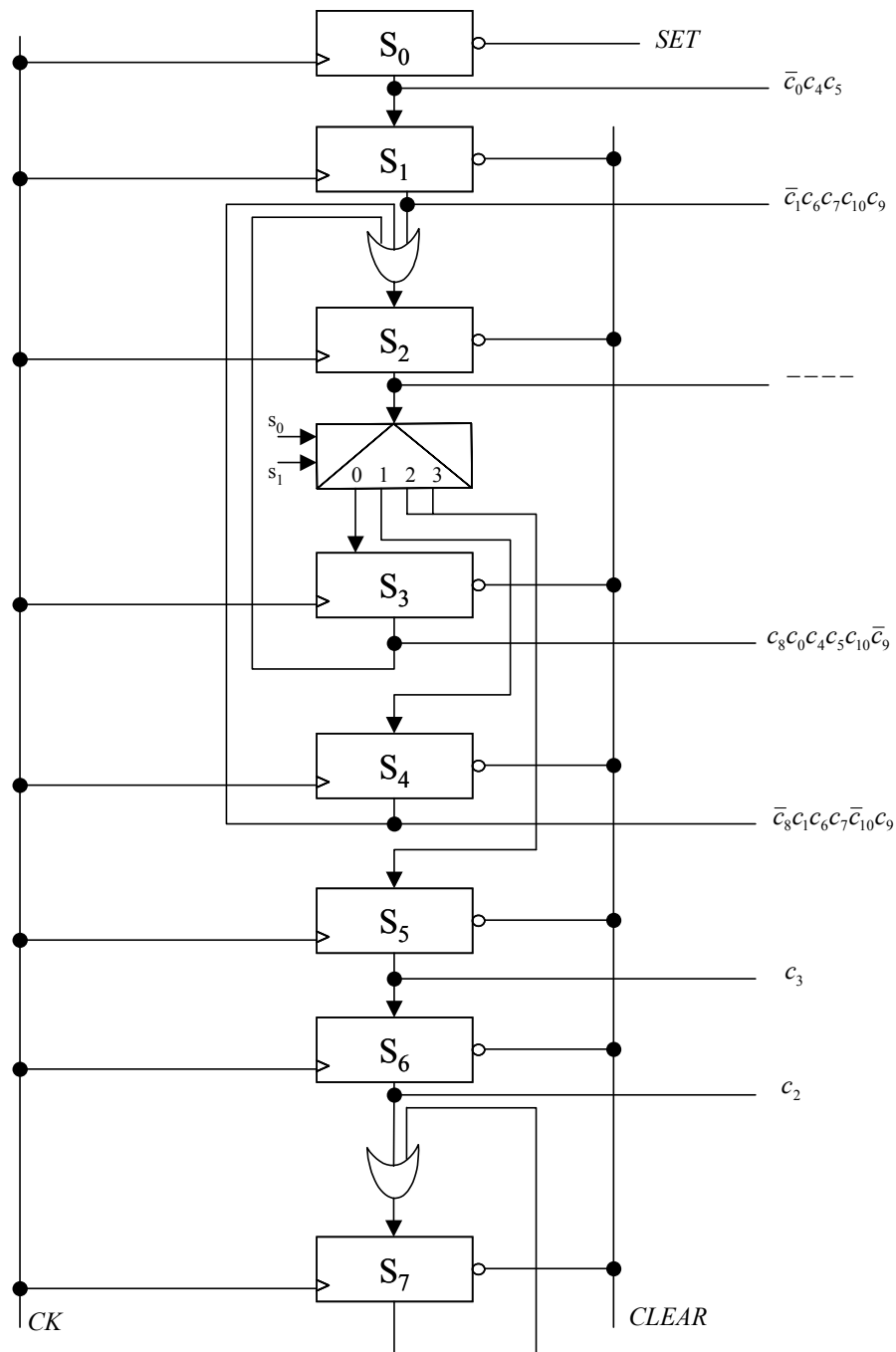


Figura 3 :Unidad de Control diseñada con un elemento de memoria por estado

OTRAS POSIBILIDADES DE DISEÑAR LA UNIDAD DE CONTROL DEL problema 5.3**1) IMPLEMENTACION DE LA UNIDAD DE CONTROL CON UNA MEMORIA ROM Y UN REGISTRO**

- Con lo mostrado en las páginas 3-6 se habría resuelto el problema de la forma que se indica en el enunciado. Ahora vamos a diseñar la Unidad de Control utilizando una memoria ROM y un registro.

- Puesto que se tienen 8 estados se necesitarían $n=3$ variables de estado (Q_2, Q_1, Q_0) para codificarlos. Por otro parte se tienen $q=2$ (s_0 y s_1) variables de condición y $m=11$ señales de control. Entonces la capacidad de la memoria ROM necesaria es :

$$C = 2^{(3+2)} \text{ palabras} \times (3+11) \text{ bits/palabra} = 2^5 \text{ palabras} \times 14 \text{ bits/palabra.}$$

- En la Tabla 2 se muestra el contenido de la memoria ROM y en la Figura 4 el esquema de la Unidad de Control

Dirección de la ROM A ₄ A ₃ A ₂ A ₁ A ₀					Contenido de la memoria ROM P ₁₃ P ₁₂ P ₁₁ P ₁₀ P ₉ P ₈ P ₇ P ₆ P ₅ P ₄ P ₃ P ₂ P ₁ P ₀													
Estado actual			Cond		Próximo Estado			Señales de control										
Q ₂	Q ₁	Q ₀	S ₀	S ₁	Q ₂	Q ₁	Q ₀	c ₁₀	c ₉	c ₈	c ₇	c ₆	c ₅	c ₄	c ₃	c ₂	c ₁	c ₀
0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0
0	0	0	0	1	0	0	1	0	0	0	0	0	1	1	0	0	0	0
0	0	0	1	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0
0	0	0	1	1	0	0	1	0	0	0	0	0	1	1	0	0	0	0
0	0	1	0	0	0	1	0	1	1	0	1	1	0	0	0	0	0	0
0	0	1	0	1	0	1	0	1	1	0	1	1	0	0	0	0	0	0
0	0	1	1	0	0	1	0	1	1	0	1	1	0	0	0	0	0	0
0	0	1	1	1	0	1	0	1	1	0	1	1	0	0	0	0	0	0
0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0
0	1	0	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	1	0	1	0	1	0	0	1	1	0	0	0	1
0	1	1	0	1	0	1	0	1	0	1	0	0	1	1	0	0	0	1
0	1	1	1	0	0	1	0	1	0	1	0	0	1	1	0	0	0	1
0	1	1	1	1	0	1	0	1	0	1	0	0	1	1	0	0	0	1
1	0	0	0	0	0	1	0	0	1	0	1	1	0	0	0	0	1	0
1	0	0	0	1	0	1	0	0	1	0	1	1	0	0	0	0	1	0
1	0	0	1	0	0	1	0	0	1	0	1	1	0	0	0	0	1	0
1	0	0	1	1	0	1	0	0	1	0	1	1	0	0	0	0	1	0
1	0	1	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0
1	0	1	0	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0
1	0	1	1	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0
1	0	1	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0
1	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	1	0	0
1	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0
1	1	0	1	0	1	1	1	0	0	0	0	0	0	0	0	1	0	0
1	1	0	1	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0
1	1	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0

Tabla 2 : Contenido de la memoria ROM

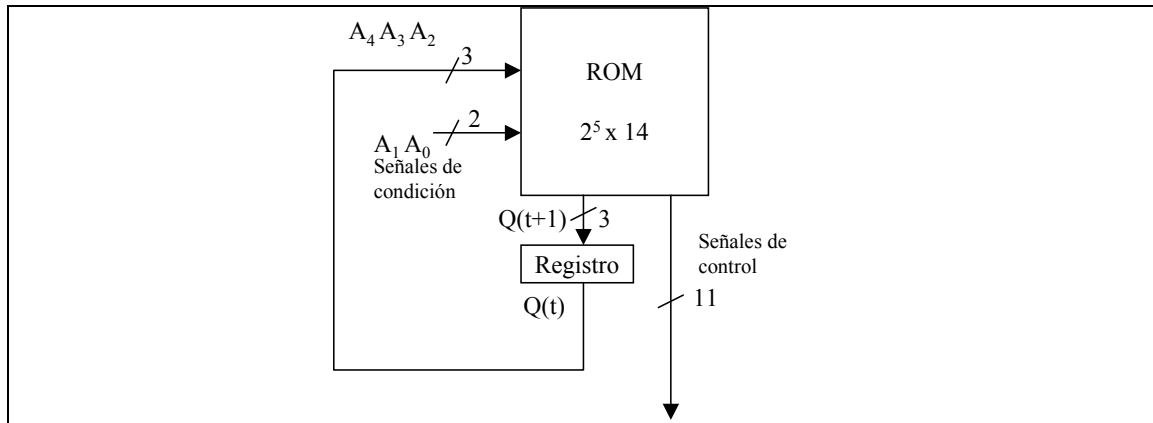


Figura 4 : Diseño de la Unidad de Control con una memoria ROM y un registro

2) IMPLEMENTACION DE LA UNIDAD DE CONTROL CON UNA MEMORIA ROM, UN REGISTRO Y UN MULTIPLEXOR DE SELECCION POR ESTADO.

Si el diagrama de estados de la Unidad de Control de la Figura 2, se modifica añadiendo un estado donde se pregunte sobre el valor de s_1 , entonces para pasar de un estado a otro solamente se necesita conocer el valor de una única señal de condición. Este nuevo diagrama de estado se muestra en la Figura 5.

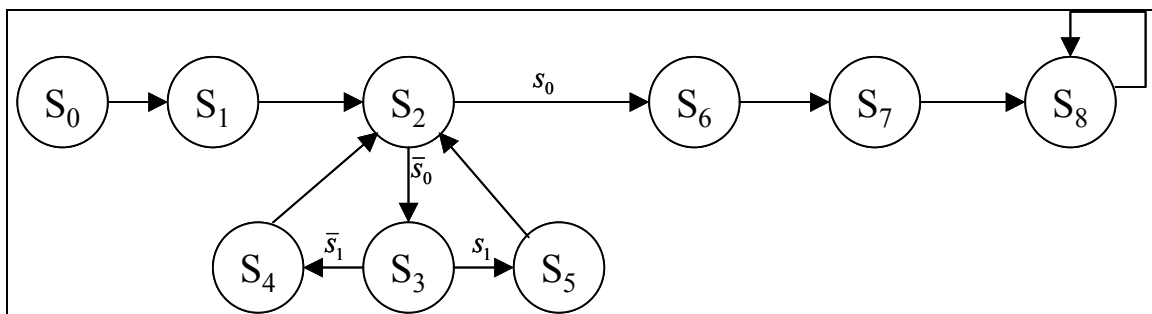


Figura 5 : Diagrama de estados de la Unidad de Control.

Se comienza en el estado S_0 , se pasa al estado S_1 y de éste se pasa al estado S_2 . Este estado correspondería a comprobar si el contador ha llegado al valor 7. Ahora dependiendo de los valores de las señales de condición s_0 , se puede pasar a S_3 ($s_0=0$) o a S_6 ($s_0=1$). En S_3 puede pasar a S_4 ($s_1=0$) o a S_5 ($s_1=1$). Cuando se encuentra en S_4 o en S_5 , vuelve a S_2 hasta que el contador alcance el valor 7 ($s_0=1$) que es cuando pasa a S_6 . Desde S_6 pasa a S_7 , y desde S_7 pasa a S_8 donde se detiene ya que es el estado de final del algoritmo.

En este diagrama se tienen 9 estados, por lo que se necesitarían $n=4$ variables de estado (Q_3, Q_2, Q_1, Q_0) para codificarlos. Por otro parte se tienen 1 (s_0 en el estado S_2 y s_1 en el estado S_3) variable de condición por estado y $m=11$ señales de control. Entonces la capacidad de la memoria ROM necesaria es :

$$C = 2^{(4+1)} \text{ palabras} \times (4+11) \text{ bits/palabra} = 2^5 \text{ palabras} \times 15 \text{ bits/palabra}.$$

- En la Tabla 3 se muestra el contenido de la memoria ROM y en la Figura 6 el esquema de la Unidad de Control

Dirección de la ROM A ₄ A ₃ A ₂ A ₁ A ₀					Contenido de la memoria ROM P ₁₄ P ₁₃ P ₁₂ P ₁₁ P ₁₀ P ₉ P ₈ P ₇ P ₆ P ₅ P ₄ P ₃ P ₂ P ₁ P ₀														
Estado actual				Cond	Próximo Estado				Señales de control										
Q ₃	Q ₂	Q ₁	Q ₀	s	Q ₃	Q ₂	Q ₁	Q ₀	c ₁₀	c ₉	c ₈	c ₇	c ₆	c ₅	c ₄	c ₃	c ₂	c ₁	c ₀
0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0
0	0	0	0	1	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0
0	0	0	1	0	0	0	1	0	1	1	0	1	1	0	0	0	0	0	0
0	0	0	1	1	0	0	1	0	1	1	0	1	1	0	0	0	0	0	0
0	0	1	0	0(s ₀)	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	1(s ₀)	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0(s ₁)	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1(s ₁)	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	1	0	1	0	1	0	0	1	1	0	0	0	1
0	1	0	0	1	0	0	1	0	1	0	1	0	0	1	1	0	0	0	1
0	1	0	1	0	0	0	1	0	0	1	0	1	1	0	0	0	0	1	0
0	1	0	1	1	0	0	1	0	0	1	0	1	1	0	0	0	0	1	0
0	1	1	0	0	0	1	1	1	0	0	0	0	0	0	0	1	0	0	0
0	1	1	0	1	0	1	1	1	0	0	0	0	0	0	0	1	0	0	0
0	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Tabla 3 : Contenido de la memoria ROM

Comentario : Puesto que se tienen 9 estados se han utilizado 4 bits para codificarlos, lo que deja el rango de direcciones 10010 hasta 11111 sin ser utilizado

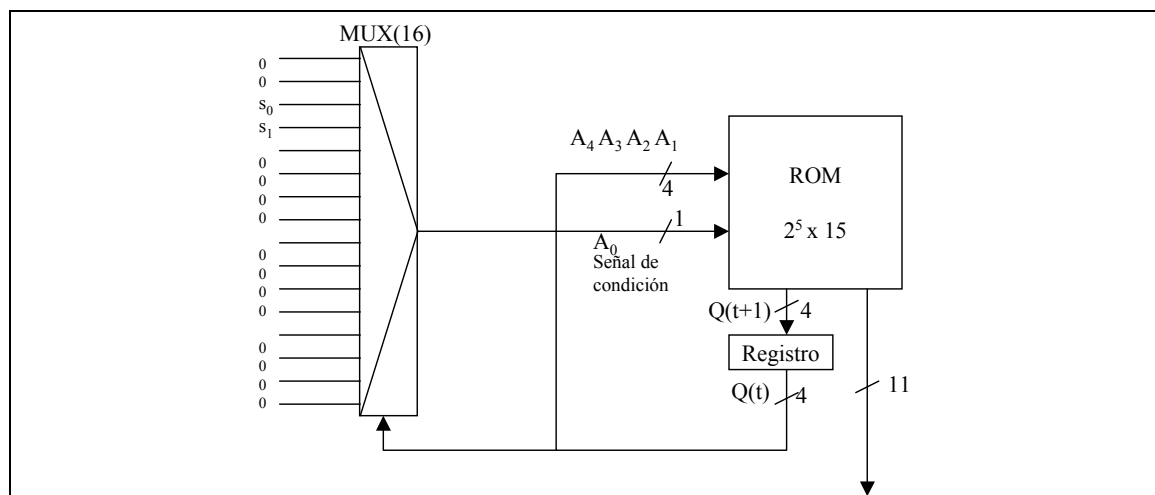


Figura 6 : Diseño de la Unidad de Control con una memoria ROM, un registro y un multiplexor por selección de estado

Comentario : El multiplexor al tener 4 señales de selección (Q_3, Q_2, Q_1, Q_0), tiene 16 entradas disponibles MUX(16). La entrada 2 se debe conectar a la señal de condición s_0 y la entrada 3 a la señal de condición s_1 . EL resto de entradas se conectan al valor 0.

3) IMPLEMENTACION DE LA UNIDAD DE CONTROL CON UNA MEMORIA ROM, UN REGISTRO Y UN MULTIPLEXOR DE SELECCION POR CAMPO.

En el diagrama de estados de la Figura 5 se tienen 9 estados, por lo que se necesitarían $n=4$ variables de estado (Q_3, Q_2, Q_1, Q_0) para codificarlos. Por otro parte se tienen $q=2$ (s_0 en el estado S_2 y s_1 en el estado S_3) variables de condición y $m=11$ señales de control. Entonces la capacidad de la memoria ROM necesaria si se desea utilizar un multiplexor de selección por campo es:

$$C = 2^{(4+1)} \text{ palabras} \times (\log_2 2 + 4 + 11) \text{ bits/palabra} = 2^5 \text{ palabras} \times 16 \text{ bits/palabra}.$$

- En la Tabla 4 se muestra el contenido de la memoria ROM y en la Figura 7 el esquema de la Unidad de Control

Dirección de la ROM $A_4 A_3 A_2 A_1 A_0$					Contenido de la memoria ROM $P_{15} P_{14} P_{13} P_{12} P_{11} P_{10} P_9 P_8 P_7 P_6 P_5 P_4 P_3 P_2 P_1 P_0$															
Estado actual				Cond	Sel MUX	Próximo Estado				Señales de control										
Q_3	Q_2	Q_1	Q_0	s	p_{15}	Q_3	Q_2	Q_1	Q_0	c_{10}	c_9	c_8	c_7	c_6	c_5	c_4	c_3	c_2	c_1	c_0
0	0	0	0	0	-	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0
0	0	0	0	1	-	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0
0	0	0	1	0	$0(s_0)$	0	0	1	0	1	1	0	1	1	0	0	0	0	0	0
0	0	0	1	1	$0(s_0)$	0	0	1	0	1	1	0	1	1	0	0	0	0	0	0
0	0	1	0	$0(s_0)$	$1(s_1)$	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	$1(s_0)$	-	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	$0(s_1)$	-	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	$1(s_1)$	-	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	$0(s_0)$	0	0	1	0	1	0	1	0	0	1	1	0	0	0	1
0	1	0	0	1	$0(s_0)$	0	0	1	0	1	0	1	0	0	1	1	0	0	0	1
0	1	0	1	0	$0(s_0)$	0	0	1	0	0	1	0	1	1	0	0	0	0	1	0
0	1	0	1	1	$0(s_0)$	0	0	1	0	0	1	0	1	1	0	0	0	0	1	0
0	1	1	0	0	-	0	1	1	1	0	0	0	0	0	0	0	1	0	0	0
0	1	1	0	1	-	0	1	1	1	0	0	0	0	0	0	0	1	0	0	0
0	1	1	1	0	-	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	1	1	1	1	-	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0
1	0	0	0	0	-	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	-	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Tabla 4 : Contenido de la memoria ROM (Cond : señal de condición. sel MUX : selección multiplexor)

Comentario : Puesto que se tienen 9 estados se han utilizado 4 bits para codificarlos, lo que deja el rango de direcciones 10010 hasta 11111 sin ser utilizado.

Comentario : El bit p_{15} va a ser la señal de selección del multiplexor, se define conjuntamente con el próximo estado. Así si nos fijamos en la Tabla 4, se puede ver que $p_{15}=0$ cuando el próximo estado es S_2 , de tal forma que la señal de condición que utiliza es la s_0 , que ha sido conectada a la entrada 0 del multiplexor. Por otro lado, $p_{15}=1$ cuando el próximo estado es S_3 , de tal forma que la señal de condición que utiliza es la s_1 , que ha sido conectada a la entrada 1 del multiplexor

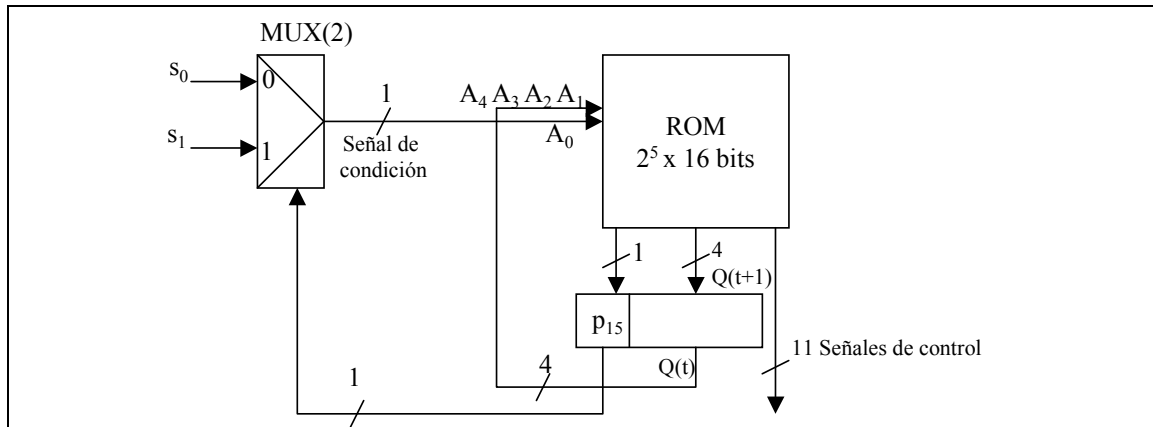


Figura 7 : Diseño de la Unidad de Control con una memoria ROM, un registro y un multiplexor de selección por campo

Comentario : El multiplexor tiene $\log_2 2 = 1$ señal de selección y 2 entradas disponibles MUX(2). La entrada 0 se debe conectar a la señal de condición s_0 y la entrada 1 a la señal de condición s_1 .

Comentario : Indicar que para diseñar la Unidad de Control, con el diagrama de estados de la Figura 5, utilizando únicamente una ROM y un registro. La capacidad de la ROM necesaria sería 2^{4+2} palabras + (11+4) bits/palabras. Que es claramente mayor a las dos propuestas anteriores..

SOLUCION PROBLEMA 5.4**DATOS**

- Diseñar una UC con memoria ROM con:
 - 200 estados
 - Genere 37 señales de control independientes entre si.
 - 9 señales de condición.
- En cada estado se consulta una señal de condición como máximo.
- Utilizar un multiplexor con selección por campo.

Del enunciado se deduce que se debe diseñar una Unidad de Control con los siguientes componentes:
Memoria ROM + registro + multiplexor de selección por campo.

Además de los datos del enunciado se deduce que:
variables de estado $n=8$ ya que $2^8=256>200$ estados
señales de control : $m=37$
señales de condición : $q=9 \leq 2^h \rightarrow h=4$

El número de entradas totales del multiplexor es $2^h=2^4=16$, de las cuales sólo se utilizan 9 (entradas 0 a 8), una para cada señal de condición, el resto (desde la entrada 9 hasta la 15) se colocan a 0.

La capacidad de la ROM sería :

$$C = 2^{n+1} \times (\log_2 q + n + m) = 2^9 \times (4+8+37)$$

$$C = 2^9 \text{ palabras} \times 49 \text{ (bits/palabra)}$$

La respuesta correcta es la **C**

SOLUCION PROBLEMA 5.5

La solución de este problema se encuentra en el fichero `probT5_05.pdf`.

SOLUCION PROBLEMA 5.6**DATOS**

- Diseñar una UC con memoria ROM con:
 - 18 estados
 - Genere 30 señales de control independientes entre si.
 - 4 señales de condición.
- En cada estado se consulta una señal de condición como máximo.
- Utilizar un multiplexor con selección por estado

Del enunciado se deduce que se debe diseñar una Unidad de Control con los siguientes componentes:
Memoria ROM + registro + multiplexor de selección por estado.

Además de los datos del enunciado se deduce que:
variables de estado $n=5$ ya que $2^5=32>18$ estados
señales de control : $m=30$
señales de condición : $q=4$

- La capacidad de la ROM necesario es:

$$C = 2^{n+1} \times (n + m) = 2^6 \times (5 + 30) = 2^6 \text{ pal} \times 35 \text{ bits / pal}$$

- El registro tiene que almacenar las variables de estado luego debe de tener una capacidad de $n=5$ bits.
- El multiplexor al ser del tipo de selección por estado, tiene 2^5 entradas y requiere de 5 señales de selección. Es decir es un MUX 32 :1.

La respuesta correcta es la **A**.

SOLUCION PROBLEMA 5.7

DATOS

- Diseñar una UC con:
 - 32 estados
 - Genere 4 señales de control independientes entre si.
 - 2 señales de condición.
- En cada estado se pueden consultar las dos señales de condición simultáneamente.

De los datos del enunciado se deduce lo siguiente ;
 variables de estado $n=5$ ya que $2^5=32$ estados
 señales de control : $m=4$
 señales de condición : $q=2$

Vamos a ir analizando cada una de las afirmaciones:

- A) Si se utiliza la técnica de los elementos de retardo sería necesario utilizar 32 elementos de retardo, uno por cada estado y no 5 como dice el enunciado. Luego la afirmación es **falsa**.
- B) Si se utiliza una memoria ROM se requiere que tenga una capacidad de $C=2^{n+q} \times (n+m)=2^7$ palabras x 9 bits/palabra. Luego la afirmación es **falsa**.
- C) Si se utiliza un contador, este debe ser de módulo- 2^5 . Luego la afirmación es **falsa**.

La respuesta correcta es la **D**.

SOLUCION PROBLEMA 5.8

DATOS

- Diseñar una UC con:
 - 64 estados
 - Genere 30 señales de control independientes entre si.
 - 5 señales de condición.
- En cada estado se pueden consultar diferentes señales de condición simultáneamente.

De los datos del enunciado se deduce lo siguiente ;
 variables de estado $64=2^6 \Rightarrow n=6$
 señales de control : $m=30$
 señales de condición : $q=5$

Vamos a ir analizando cada una de las afirmaciones:

- A) Si se utiliza la técnica de los elementos de retardo sería necesario utilizar 64 elementos de retardo, uno por cada estado y no 6 como dice el enunciado. Luego la afirmación es **falsa**.
- B) Si se utiliza una memoria ROM se requiere que tenga una capacidad mínima de $C=2^{n+q} \times (n+m)=2^{11}$ palabras x 36 bits/palabra. La afirmación es **falsa**.
- C) Si se utiliza un contador, este debe ser de módulo- 2^6 . Luego la afirmación es **Verdadera**.

La respuesta correcta es la **C**.

SOLUCION PROBLEMA 5.9

La solución de este problema se encuentra en el fichero `probT5_09.pdf`.

Explicación previa a la resolución del problema

En el algoritmo que hay que implementar se plantea la multiplicación y la división de un número binario por 2. Para realizar estas operaciones es necesario recordar como se pasa un número binario a decimal:

$$[x_{n-1} x_{n-2} \dots x_0] \text{ binario a decimal } x_{n-1}2^{n-1} + x_{n-2}2^{n-2} + \dots + x_02^0$$

- La multiplicación de un número binario por 2 es:

$$[x_{n-1} x_{n-2} \dots x_0] \times 2 \text{ binario a decimal } x_{n-1}2^n + x_{n-2}2^{n-1} + \dots + x_02^1$$

Lo que es equivalente a desplazar el número binario un bit hacia la izquierda e introducir un 0 en la posición menos significativa.

Ejemplos:

El número $4_{10} = (0100)_2$ al multiplicarlo por 2 se debe desplazar un bit hacia la izquierda e introducir un 0 en la posición menos significativa se obtiene $(1000)_2 = 8_2 = 2 \times 4_{10}$.

El número $3_{10} = (0011)_2$ al multiplicarlo por 2 se debe desplazar un bit hacia la izquierda e introducir un 0 en la posición menos significativa se obtiene $(0110)_2 = 6_{10} = 2 \times 3_{10}$.

- La división entera de un número binario por 2 es:

$$[x_{n-1} x_{n-2} \dots x_0] / 2 \text{ binario a decimal } x_{n-1}2^{n-2} + x_{n-2}2^{n-3} + \dots + x_12^0$$

Lo que es equivalente a desplazar el número binario un bit hacia la derecha perdiéndose el contenido del bit menos significativo.

Ejemplos:

El número $4_{10} = (0100)_2$ al dividirlo por 2 se debe desplazar un bit hacia la derecha se obtiene $(0010)_2 = 2_{10} = 4_{10} / 2_{10}$.

El número $3_{10} = (0011)_2$ al dividirlo por 2 se debe desplazar un bit hacia la derecha se obtiene $(0001)_2 = 1_{10} = 4_{10} / 3_{10}$. Es una división entera.

Sobre el bucle la señal de condición s_0

Hay que darse cuenta que aunque no lo pone en el algoritmo la realización de este bucle requiere inicializar el contador a 0, luego habrá que introducir la microoperación *reset cont* en el estado adecuado.

Por otra parte, la señal de condición s_0 , es la salida de una puerta OR que tiene como entradas las salidas de un contador módulo 7, es decir, cuenta desde 0 hasta 7, de tal forma que al llegar a 8 vuelve a empezar desde 0, etc. Por lo tanto:

$$\begin{aligned} s_0 &= 1, \text{ si } cont \text{ es distinto de } 0. \\ s_0 &= 0, \text{ si el } cont = 0. \end{aligned}$$

$s_0 = 0$ inicialmente (*reset Cont*) pero también cuando llegue al valor 8, es decir, cuando se haya ejecutado el bucle *for* 8 veces tal y como se indica en la sentencia 4 del algoritmo.

SOLUCION PROBLEMA 5.10

La solución de este problema se encuentra en el fichero `probT5_10.pdf`.

SOLUCION PROBLEMA 5.11

La solución de este problema se encuentra en el fichero `probT5_11.pdf`.

SOLUCION PROBLEMA 5.12

La solución de este problema se encuentra en el fichero `probT5_12.pdf`.

SOLUCION PROBLEMA 5.13

- Diseñar una UC con memoria ROM con:
 - 200 estados
 - Genere 37 señales de control independientes entre si.
 - 9 señales de condición.
- En cada estado se consulta una señal de condición como máximo.
- Se utiliza el diseño con un multiplexor con selección por campo

Se tiene una Unidad de Control diseñada con Memoria ROM + registro + multiplexor de selección por campo.

De los datos del enunciado se deduce que:
 variables de estado $n=8$ ya que $2^8=256>200$ estados
 señales de control : $m=37$
 señales de condición : $q=9 \leq 2^h \rightarrow h=4$

En este diseño según lo estudiado en teoría, se necesitará un multiplexor con $2^h = 2^4 = 16$ entradas de las cuales sólo se utilizan 9 una para cada señal de condición. el resto se conectan a 0.

Luego la respuesta es **16 entradas de datos**.

SOLUCION PROBLEMA 5.14

- Diseñar una UC con memoria ROM con:
 - 128 estados
 - Genere 244 señales de control independientes entre si.
 - 20 señales de condición.
- La memoria ROM utilizada tiene un coste de 10 ptas/Kbit

De los datos del enunciado se deduce que:
 variables de estado $n=7$ ya que $2^7=127$ estados
 señales de control : $m=244$
 señales de condición : $q=20 \leq 2^h \rightarrow h=5$

Afirmación I:

Se tiene una Unidad de Control diseñada con Memoria ROM + registro + multiplexor de selección por estado. La Capacidad de la ROM es:

$$C = 2^{n+1} \text{ palabras} \times (n+m) \text{ bits/palabras}$$

Luego substituyendo valores se tiene:
 $C = 2^8 \cdot 251 \text{ bits} = (2^8 \cdot 251 \cdot 2^{10}) / 2^{10} = (251/4) \text{ Kbits}$

Luego el coste es $\$ = (215/4) \text{ Kbits} \cdot 10 \text{ ptas/Kbits} = 2150/4 = 627.5 \text{ ptas.}$

Luego la afirmación es **FALSA**.

Afirmación II:

Se tiene una Unidad de Control diseñada con Memoria ROM + registro + multiplexor de selección por campo. La Capacidad de la ROM es:

$$C=2^{n+1} \text{ palabras} \times (h+ n+m) \text{ bits/palabras}$$

Luego sustituyendo valores se tiene:

$$C=2^8 \cdot (5+7+244) = 2^8 \cdot 256 \text{ bits} = 2^8 \cdot 2^8 = 2^{16} = 2^6 \cdot 2^{10} = 64 \text{ Kbits}$$

Luego el coste es $\$=64 \text{ Kbits} \cdot 10 \text{ ptas/Kbits} = 640 \text{ ptas.}$

Luego la afirmación es **VERDADERA**.

SOLUCION PROBLEMA 5.15

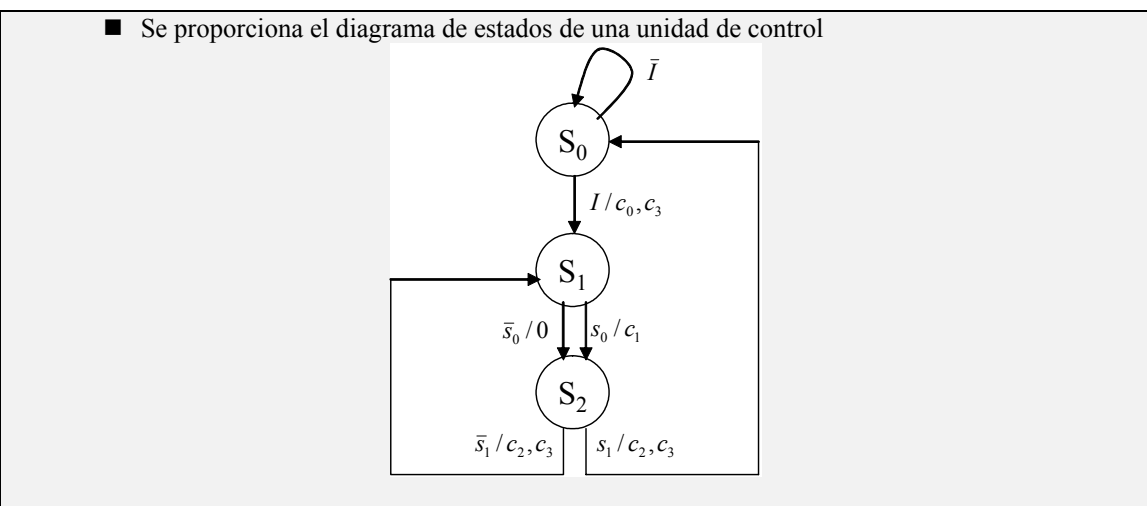
La solución de este problema se encuentra en el fichero probT5_15.pdf.

SOLUCION PROBLEMA 5.16

La solución de este problema se encuentra en el fichero probT5_16.pdf.

SOLUCION PROBLEMA 5.17

La solución de este problema se encuentra en el fichero probT5_17.pdf.

SOLUCION PROBLEMA 5.18

De la observación de este diagrama de estados se obtienen los siguientes datos del sistemas:

- Posee 3 estados $\{S_0, S_1, S_2\}$ luego son necesarios dos variables de estado (Q_1, Q_0) para codificarlos. Es decir, $n=2$.
- Posee 4 señales de control $\{c_0, c_1, c_2, c_3\}$. Luego $m=4$.
- Posee 2 señales de condición $\{I, s_0, s_1\}$. Luego $q=3$.

Pasemos a analizar las afirmaciones:

A) Afirmar que se puede implementar esta unidad de control usando un registro de 2 bits y una ROM de capacidad 32 palabras x 8 bits/palabra.

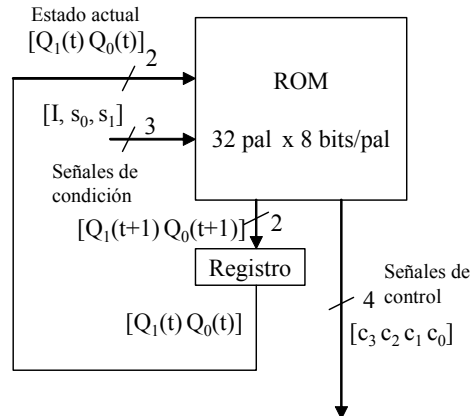
De forma general, cuando se pretende implementar una unidad de control con una ROM y un registro, estos deben tener las siguientes capacidades:

- ROM $C=2^{n+q}$ palabras x (n+m) (bits/palabra)
- Registro de n bits.

Sustituyendo los valores de n, m y q de nuestro problema se obtiene:

- ROM $C=2^{2+3}$ palabras x (2+4) (bits/palabra)= 32 palabras x (6) (bits/palabra)
- Registro de 2 bits.

El esquema del diseño se muestra en la figura siguiente:



Luego la afirmación A es **VERDADERA**. Comentar que la ROM que necesitamos requiere solo 6 bits/pal como la que nos dan es de 8 bits /pal, eso implica que se va a desaprovechar memoria.

B) Afirma que se puede implementar esta unidad de control usando un registro de 2 bits, un multiplexor de 4 a 1 y una ROM de capacidad 8 palabras x 8 bits/palabra. Puesto que $n=2$, y el registro es de 2 bits de capacidad. Eso implica que las señales de selección del multiplexor serán las variables de estado.

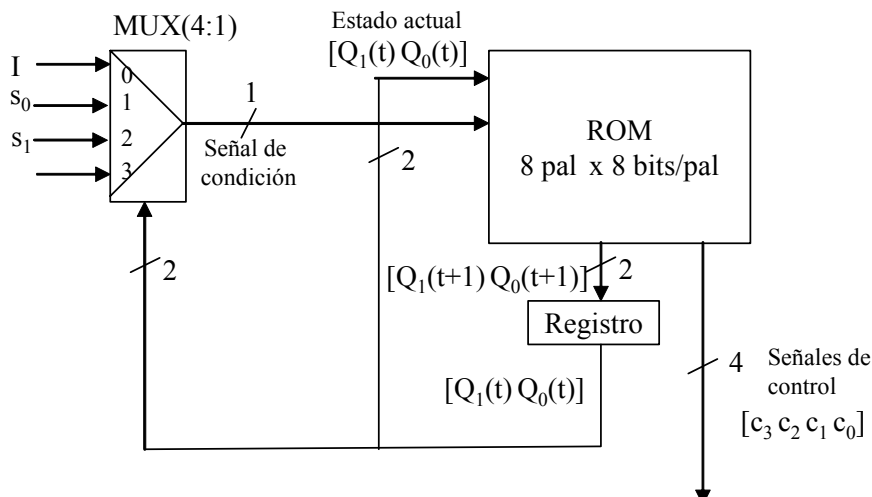
De forma general, cuando se pretende implementar una unidad de control con una ROM, un registro y un multiplexor de selección por estado estos deben tener las siguientes capacidades:

- ROM $C=2^{n+1}$ palabras x (n+m) (bits/palabra)
- Registro de n bits.
- Multiplexor de 2^n a 1

Sustituyendo los valores de n, m y q de nuestro problema se obtiene:

- ROM $C=2^{2+1}$ palabras x (2+4) (bits/palabra)= 8 palabras x (6) (bits/palabra)
- Registro de 2 bits.
- Multiplexor de 4 a 1

El esquema del diseño se muestra en la figura siguiente:

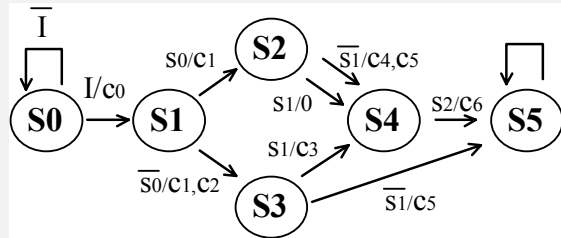


Luego la afirmación B es **VERDADERA**. Al igual que en el caso anterior comentar que la ROM que necesitamos requiere solo 6 bits/pal como la que nos dan es de 8 bits /pal, eso implica que se va a desaprovechar memoria.

En conclusión, la respuesta correcta es la C

SOLUCION PROBLEMA 5.19

- Se proporciona el diagrama de estados de una unidad de control, para calcular las expresiones de las señales de control c_1 y c_5



Las señales de control c_i $i=0, \dots, 6$ se expresarán como funciones del estado actual S_j $j=0, \dots, 5$ y de las señales de condición s_k $k=0, 1, 2$

Por simple inspección visual se observa que la señal de control c_1 se activa en los siguientes casos:

- Si el estado actual es S_1 y la señal de condición es s_0 , es decir, $s_0 \cdot S_1$
- Si el estado actual es S_1 y la señal de condición es \bar{s}_0 , es decir, $\bar{s}_0 \cdot S_1$

Luego la expresión lógica de c_1 es:

$$c_1 = s_0 \cdot S_1 + \bar{s}_0 \cdot S_1 = (s_0 + \bar{s}_0) \cdot S_1 = S_1$$

Por otra parte por simple inspección visual se observa que la señal de control c_5 se activa en los siguientes casos:

- Si el estado actual es S_2 y la señal de condición es \bar{s}_1 , es decir, $\bar{s}_1 \cdot S_2$
- Si el estado actual es S_3 y la señal de condición es \bar{s}_1 , es decir, $\bar{s}_1 \cdot S_3$

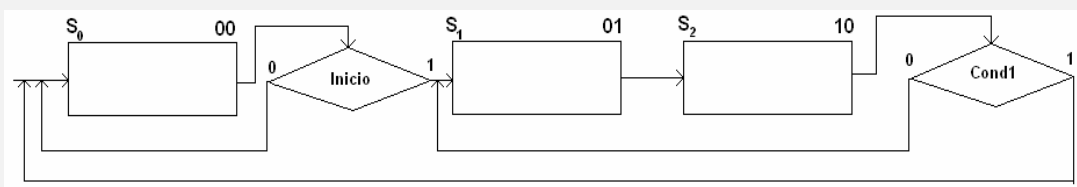
Luego la expresión lógica de c_5 es:

$$c_5 = \bar{s}_1 \cdot S_2 + \bar{s}_1 \cdot S_3 = \bar{s}_1 \cdot (S_2 + S_3)$$

Luego la afirmación correcta es la A.

SOLUCION PROBLEMA 5.20

- Diagrama ASM



- En cada estado se debe generar una sola señal de control (c_0 en S_0 , c_1 en S_1 , c_2 en S_2)

De acuerdo con el diagrama ASM adjunto esta unidad de control tiene:

- $N=3$ estados (S_0, S_1, S_2) $\rightarrow 3 \leq 2^n \rightarrow n=2$ variables de estado (Q_1, Q_0).
- $q=2$ señales de condición (Inicio, Cond1).

Además según el enunciado

- $m=3$ señales de control (c_0, c_1, c_2)

Analicemos la veracidad de las afirmaciones:

I) Para implementar una unidad de control empleando una ROM y un registro, es necesario que éstos elementos tengan las siguientes características:

Capacidad de la ROM:

$$C \geq 2^{n+q} \text{ palabras} \times (n+m) \text{ (bits/palabra)} \geq 2^4 \text{ palabras} \times 5 \text{ (bits/pal)}$$

Registro de $n \geq 2$ bits

Puesto que la ROM de la afirmación posee un ancho de palabra de 4 bits/palabra menor del ancho necesario (5 bits/palabra) esta afirmación es **FALSA**.

II) Si se usa una ROM, un MUX de selección por estado y un registro es necesario que éstos elementos tengan las siguientes características:

Capacidad de la ROM:

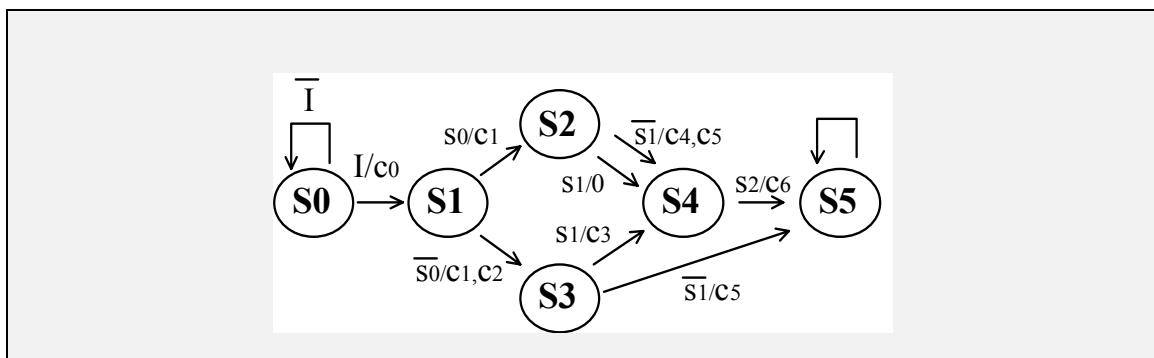
$$C \geq 2^{n+1} \text{ palabras} \times (n+m) \text{ (bits/palabra)} \geq 2^3 \text{ palabras} \times 5 \text{ (bits/pal)}$$

Registro de $n \geq 2$ bits

$$\text{MUX} (2^n : 1) \rightarrow \text{MUX} (2^2 : 1) \rightarrow \text{MUX} (4 : 1)$$

De acuerdo con el diagrama ASM las transiciones entre estados están regidas por una única señal de condición, luego es posible implementar la unidad de control usando una ROM, un registro y un MUX. Además puesto que el MUX del enunciado es del tipo 4:1 y la capacidad de la ROM del enunciado es de 8 palabras \times (5 bits/palabra). Se observa que todos los elementos proporcionados poseen las características adecuadas para poder implementar la unidad de control. En conclusión la afirmación es **VERDADERA**.

SOLUCION PROBLEMA 5.21



De acuerdo con el diagrama de estados adjunto esta unidad de control tiene:

- $N=6$ estados (S_0, S_1, \dots, S_5) $\rightarrow 6 \leq 2^n \rightarrow n=3$ variables de estado (Q_2, Q_1, Q_0).
- $q=4$ señales de condición (I, S_0, S_1, S_2).
- $m=7$ señales de control (c_0, c_1, \dots, c_6)

Analicemos la veracidad de las afirmaciones propuestas:

A) Para implementar una unidad de control empleando una ROM y un registro, es necesario que éstos elementos tengan las siguientes características:

Capacidad de la ROM:

$$C \geq 2^{n+q} \text{ palabras} \times (n+m) \text{ (bits/palabra)} \geq 2^7 \text{ palabras} \times 10 \text{ (bits/pal)}$$

Registro de $n \geq 3$ bits

Puesto que la capacidad de la ROM de la afirmación posee solo 27 palabras es menor de la necesaria (128 palabras) luego esta afirmación es **FALSA**.

B) Si se usa una ROM, un MUX y un registro es necesario que éstos elementos tengan las siguientes características:

Caso 1: *Multiplexor de selección por estado*

- Capacidad de la ROM:

$$C \geq 2^{n+1} \text{ palabras} \times (n+m) \text{ (bits/palabra)} \geq 2^4 \text{ palabras} \times 10 \text{ (bits/pal)}$$

- Registro de $n \geq 3$ bits

- MUX ($2^n:1$) \rightarrow MUX($2^3:1$) \rightarrow MUX(8:1)

Caso 2: *Multiplexor de selección por campo*

- Capacidad de la ROM:

$$C \geq 2^{n+1} \text{ palabras} \times (\log_2 q + n + m) \text{ (bits/palabra)} \geq 2^4 \text{ palabras} \times 12 \text{ (bits/pal)}$$

- Registro de $(\log_2 q + n) \geq 5$ bits

- MUX ($q:1$) \rightarrow MUX(4:1)

Puesto que de acuerdo con el diagrama de estados las transiciones entre estados están regidas por una única señal de condición es posible implementar la unidad de control usando una ROM, un registro y un MUX. El registro de la afirmación posee 3 bits, el MUX es del tipo 8:1 y la capacidad de la ROM del enunciado es de 16 palabras x (16 bits/palabra). Se observa que todos los elementos proporcionados poseen las características adecuadas para poder implementar una unidad de control según el caso 1. En conclusión la afirmación es **VERDADERA**.

Luego la respuesta correcta es la **B**.

SOLUCION PROBLEMA 5.22

- Unidad de control implementada con dos elementos de memoria tipo D, uno por cada variable de estado.
- Hay dos variables de estado (Q_1, Q_0).
- La codificación de los estados es $S_0: (Q_1=0, Q_0=0)$, $S_1: (Q_1=0, Q_0=1)$, $S_2: (Q_1=1, Q_0=0)$.

Estado presente	Próximo estado			
	$\bar{I} \cdot \bar{s}_0$	$\bar{I} \cdot s_0$	$I \cdot \bar{s}_0$	$I \cdot s_0$
S_0	S_0	S_0	S_1	S_1
S_1	S_3	S_2	S_3	S_2
S_2	S_0	S_3	S_0	S_3
S_3	S_3	S_3	S_3	S_3

Como se conoce la codificación de los estados es posible construir la siguiente tabla equivalente a la dada en el enunciado:

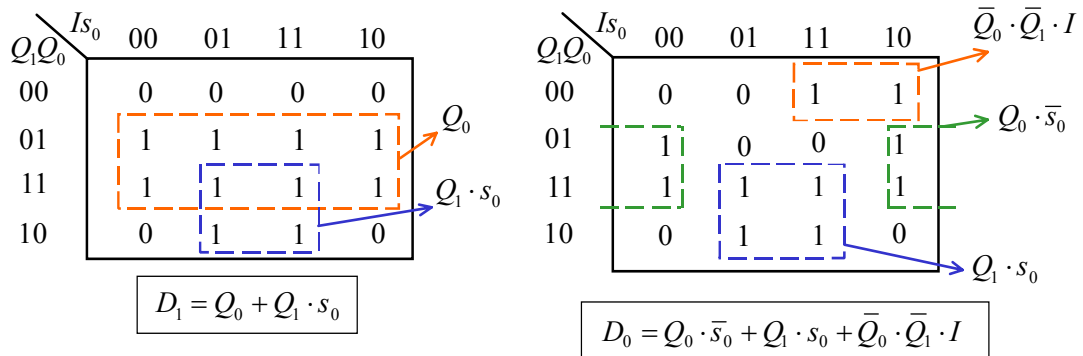
Estado actual		Señales de condición		Próximo estado	
$Q_1(t)$	$Q_0(t)$	I	s_0	$Q'_1(t+1)$	$Q'_0(t+1)$
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	1
0	0	1	1	0	1
0	1	0	0	1	1
0	1	0	1	1	0
0	1	1	0	1	1
0	1	1	1	1	0
1	0	0	0	0	0
1	0	0	1	1	1
1	0	1	0	0	0
1	0	1	1	1	1
1	1	0	0	1	1
1	1	0	1	1	1
1	1	1	0	1	1
1	1	1	1	1	1

En la implementación de una UC usando elementos de memoria tipo D, la entrada de cada uno de estos elementos es:

$$D_0 = Q'_0$$

$$D_1 = Q'_1$$

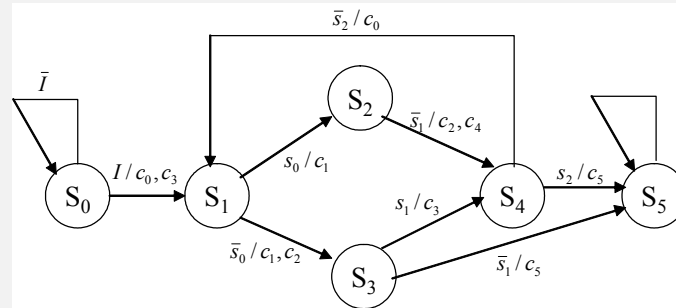
Usando la técnica de los mapas de Karnaugh es posible obtener las expresiones lógicas de D_0 y D_1 .



Luego la respuesta correcta es la C.

SOLUCION PROBLEMA 5.23

- Se proporciona el diagrama de estados de una unidad de control, para calcular las expresiones de las señales de control c_3 y c_5



Las señales de control c_i $i=0, \dots, 5$ se expresarán como funciones del estado actual S_j $j=0, \dots, 5$ y de las señales de condición s_k $k=0, 1, 2$

Por simple inspección visual se observa que la señal de control c_3 se activa en los siguientes casos:

- 1) Si el estado actual es S_0 y la señal de condición es I , es decir, $I \cdot S_0$
- 2) Si el estado actual es S_3 y la señal de condición es s_1 , es decir, $s_1 \cdot S_3$

Luego la expresión lógica de c_3 es:

$$c_3 = I \cdot S_0 + s_1 \cdot S_3$$

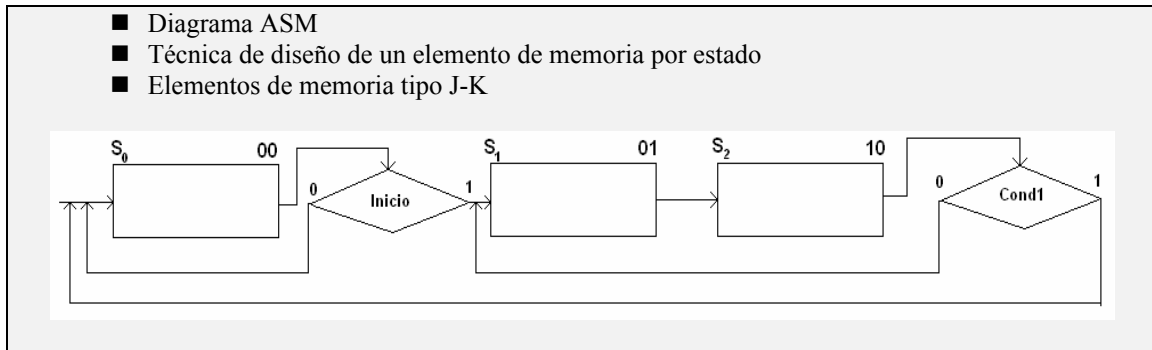
Por otra parte por simple inspección visual se observa que la señal de control c_5 se activa en los siguientes casos:

- 1) Si el estado actual es S_3 y la señal de condición es \bar{s}_1 , es decir, $\bar{s}_1 \cdot S_3$
- 2) Si el estado actual es S_4 y la señal de condición es s_2 , es decir, $s_2 \cdot S_4$

Luego la expresión lógica de c_5 es:

$$c_5 = \bar{s}_1 \cdot S_3 + s_2 \cdot S_4$$

Luego la afirmación correcta es la D.

SOLUCION PROBLEMA 5.24

En el libro de teoría esta técnica de diseño de una UC se ha descrito supuesto que se utilizan elementos de memoria o biestables tipo D, y se necesitaban tantos biestables tipo D como estados tuviese la UC. En este caso el diagrama ASM indica que la UC posee 3 estados $\{S_0, S_1, S_2\}$ luego serán necesarios tres biestables tipo D para implementar la UC por la técnica de un elemento de memoria por estado. Es posible implementar esta técnica con otros tipos de biestables pero entonces es necesario utilizar las tablas de transición de estados y calcular las ecuaciones de entrada a los elementos de memoria.

En el enunciado se desea implementar esta técnica con biestables tipo J-K. Por lo tanto se debe comprobar si con un biestable tipo J-K se puede emular un biestable tipo D.

La tabla de verdad o funcionamiento de un biestable tipo D es:

Entrada D	Estado actual Q(t)	Próximo estado Q'(t+1)
0	0	0
0	1	0
1	0	1
1	1	1

Se observa que $D=Q(t+1)$.

Mientras que la tabla de funcionamiento de un biestable tipo J-K es:

Entrada J	Entrada K	Estado actual Q(t)	Próximo estado Q'(t+1)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Se observa que las filas sombreadas de esta tabla son precisamente la tabla de funcionamiento de un biestable tipo D si se hace la entrada $J = D$ y la entrada $K = \overline{D}$.

Luego se ha demostrado que un biestable tipo J-K puede ser equivalente a un biestable tipo D. En conclusión puesto que se necesitan 3 biestables tipo D para implementar la UC con la técnica de “un elemento de memoria por estado”, de forma equivalente se podría implementar con **3 elementos biestables tipo J-K**.

SOLUCION PROBLEMA 5.25

- Unidad de control implementada con dos elementos de memoria tipo D, uno por cada variable de estado.
- Hay dos variables de estado (Q_1, Q_0).
- La codificación de los estados es $S_0: (Q_1=0, Q_0=0)$, $S_1: (Q_1=0, Q_0=1)$, $S_2: (Q_1=1, Q_0=0)$.

Estado presente	Próximo estado			
	$\bar{I} \cdot \bar{s}_0$	$\bar{I} \cdot s_0$	$I \cdot \bar{s}_0$	$I \cdot s_0$
S_0	S_0	S_0	S_1	S_1
S_1	S_3	S_2	S_3	S_2
S_2	S_3	S_1	S_3	S_1
S_3	S_3	S_0	S_3	S_0

Como se conoce la codificación de los estados es posible construir la siguiente tabla equivalente a la dada en el enunciado:

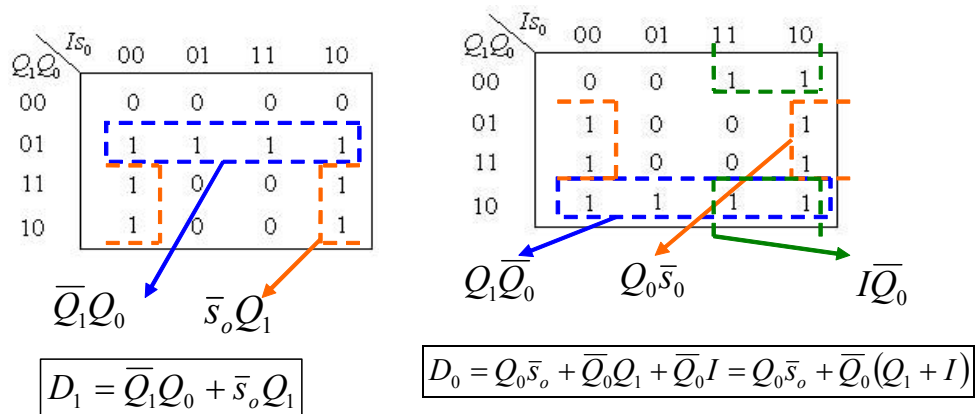
Estado actual		Señales de condición		Próximo estado	
$Q_1(t)$	$Q_0(t)$	I	s_0	$Q'_1(t+1)$	$Q'_0(t+1)$
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	1
0	0	1	1	0	1
0	1	0	0	1	1
0	1	0	1	1	0
0	1	1	0	1	1
0	1	1	1	1	0
1	0	0	0	1	1
1	0	0	1	0	1
1	0	1	0	1	1
1	0	1	1	0	1
1	1	0	0	1	1
1	1	0	1	0	0
1	1	1	0	1	1
1	1	1	1	0	0

En la implementación de una UC usando elementos de memoria tipo D, la entrada de cada uno de estos elementos es:

$$D_0 = Q'_0$$

$$D_1 = Q'_1$$

Usando la técnica de los mapas de Karnaugh es posible obtener las expresiones lógicas de D_0 y D_1 .



Luego la respuesta correcta es la A).

SOLUCION PROBLEMA 5.26

- UC de 28 estados.
- Implementarla mediante ROM + registro + MUX de selección por campo

De acuerdo con lo indicado en el libro de teoría, para implementar la UC con estos elementos es necesario conocer el número de señales de condición que rigen la transición entre los estados y si cada estado requiere como máximo de una única señal de condición para pasar al próximo estado. Esta información no se da en el enunciado. Luego la solución correcta es la C).

SOLUCION PROBLEMA 5.27

- UC de 32 estados y 10 señales de condición. Sólo es necesario consultar en cada estado una única señal de condición
- Implementarla mediante ROM + registro + MUX de selección por campo

De los datos del enunciado se deduce que:

$$N=32 \text{ estados} \leq 2^n \Rightarrow n=5 \text{ variables de estado}$$

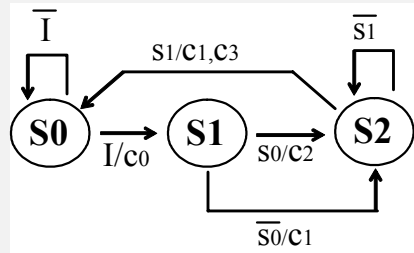
$$q=10 \text{ señales de condición} \leq 2^h \Rightarrow h=4 \text{ señales de control del MUX.}$$

Además puesto que en cada estado solamente es necesario consultar una señal de condición, es posible implementar la UC usando los elementos y la técnica propuestos en el enunciado.

Del libro de teoría se sabe que para esta metodología de diseño el registro debe poseer dos campos, uno para almacenar las h señales de control del MUX y otro para almacenar las n variables de estado. Luego el tamaño del registro debe ser de $h+n$ bits. En este caso el registro necesario debe tener un tamaño de

$$h+n=4+5=9 \text{ bits}$$

Luego la respuesta correcta es la D).

SOLUCION PROBLEMA 5.28

De acuerdo con el diagrama de estados adjunto esta unidad de control tiene:

- $N=3$ estados (S_0, S_1, S_2) $\rightarrow 3 \leq 2^n \rightarrow n=2$ variables de estado (Q_1, Q_0).
- $q=3$ señales de condición (I, S_0, S_1).
- $m=4$ señales de control (c_0, c_1, c_2, c_3)

Analicemos la veracidad de las afirmaciones propuestas:

I) Para implementar una unidad de control empleando una ROM y un registro, es necesario que éstos elementos tengan las siguientes características:

- Capacidad de la ROM:

$$C \geq 2^{n+q} \text{ palabras} \times (n+m) \text{ (bits/palabra)} \geq 2^5 \text{ palabras} \times 6 \text{ (bits/pal)}$$

- Registro de $n \geq 2$ bits

Luego esta afirmación I es **VERDADERA**.

II) Si se usa una ROM, un MUX y un registro es necesario que éstos elementos tengan las siguientes características:

Caso 1: *Multiplexor de selección por estado*

- Capacidad de la ROM:

$$C \geq 2^{n+1} \text{ palabras} \times (n+m) \text{ (bits/palabra)} \geq 2^3 \text{ palabras} \times 6 \text{ (bits/pal)}$$

- Registro de $n \geq 2$ bits

- MUX ($2^n:1$) \rightarrow MUX($2^2:1$) \rightarrow MUX(4:1)

Caso 2: *Multiplexor de selección por campo*

- Capacidad de la ROM:

$$C \geq 2^{n+1} \text{ palabras} \times (\log_2 q + n + m) \text{ (bits/palabra)} \geq 2^3 \text{ palabras} \times 8 \text{ (bits/palabra)}$$

- Registro de $(\log_2 q + n) \geq 4$ bits

- MUX ($q:1$) \rightarrow MUX(4:1)

Puesto que de acuerdo con el diagrama de estados las transiciones entre estados están regidas por una única señal de condición es posible implementar la unidad de control usando una ROM, un registro y un MUX. El registro de la afirmación posee 2 bits, el MUX es del tipo 4:1 y la capacidad de la ROM del enunciado es de 8 palabras \times 6 (bits/palabra). Se observa que todos los elementos proporcionados poseen las características adecuadas para poder implementar una unidad de control según el caso 1. En conclusión la afirmación II es **VERDADERA**.

SOLUCION PROBLEMA 5.29

- Unidad de control con cuatro estados: $S_0: (Q_1=0, Q_0=0)$, $S_1: (Q_1=0, Q_0=1)$, $S_2: (Q_1=1, Q_0=0)$.
- Hay dos variables de estado (Q_1, Q_0).
- Unidad de control implementada con dos elementos de memoria tipo D, uno por cada variable de estado.
- $D_0 = Q_1 + \overline{Q_0} \cdot I + Q_0 \cdot \overline{s_0}$
- $D_1 = s_0 \cdot Q_1 + \overline{Q_1} \cdot Q_0 + Q_1 \cdot \overline{Q_0}$

En primer lugar debe recordarse que en un elemento de memoria tipo D cuyo estado actual es $Q(t)$ la función lógica del próximo estado $Q(t+1)$ es $Q(t+1)=Q'=D(t)$.

Luego en la unidad de control propuesta se tiene que $Q_1'=D_1$ y $Q_0'=D_0$. En el enunciado dan las funciones lógicas de D_1 y D_0 . Para averiguar si las afirmaciones son verdaderas o falsas basta con evaluar estas funciones en los estados indicados en cada afirmación.

Afirmación I: Hay que evaluar las expresiones de D_1 y de D_0 en el estado S_2 ($Q_1=1, Q_0=0$):

$$D_0 = Q_1 + \overline{Q_0} \cdot I + Q_0 \cdot \overline{s_0} = 1 + I + 0 \cdot \overline{s_0} = 1$$

$$D_1 = s_0 \cdot Q_1 + \overline{Q_1} \cdot Q_0 + Q_1 \cdot \overline{Q_0} = s_0 \cdot 1 + 0 \cdot 0 + 1 \cdot 1 = 1$$

Luego el próximo estado es $(D_1, D_0) = (Q_1', Q_0') = (1, 1) = S_3$. Por lo tanto la afirmación es **VERDADERA**.

Afirmación II: Hay que evaluar las expresiones de D_1 y de D_0 en el estado S_1 ($Q_1=0, Q_0=1$) y además considerar que $s_0=1$:

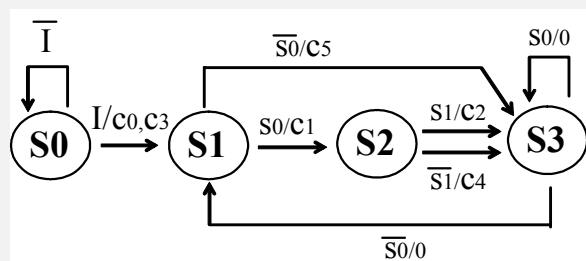
$$D_0 = Q_1 + \overline{Q_0} \cdot I + Q_0 \cdot \overline{s_0} = 0 + 0 \cdot I + 1 \cdot 0 = 0$$

$$D_1 = s_0 \cdot Q_1 + \overline{Q_1} \cdot Q_0 + Q_1 \cdot \overline{Q_0} = 1 \cdot 0 + 1 \cdot 1 + 0 \cdot 1 = 1$$

Luego el próximo estado es $(D_1, D_0) = (Q_1', Q_0') = (0, 1) = S_2$. Por lo tanto la afirmación es **VERDADERA**.

SOLUCION PROBLEMA 5.30

- Unidad de control implementada con dos elementos de memoria tipo D, uno por cada variable de estado.
- Hay dos variables de estado (Q_1, Q_0).
- La codificación de los estados es $S_0: (Q_1=0, Q_0=0)$, $S_1: (Q_1=0, Q_0=1)$, $S_2: (Q_1=1, Q_0=0)$.



Como se conoce el diagrama de transición de estados es posible construir la siguiente tabla de transición de estados

Estado actual		Señales de condición		Próximo estado	
$Q_1(t)$	$Q_0(t)$	I	s_0	$Q'_1(t+1)$	$Q'_0(t+1)$
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	1
0	0	1	1	0	1
0	1	0	0	1	1
0	1	0	1	1	0
0	1	1	0	1	1
0	1	1	1	1	0
1	0	0	0	1	1
1	0	0	1	1	1
1	0	1	0	1	1
1	0	1	1	1	1
1	1	0	0	0	1
1	1	0	1	1	1
1	1	1	0	0	1
1	1	1	1	1	1

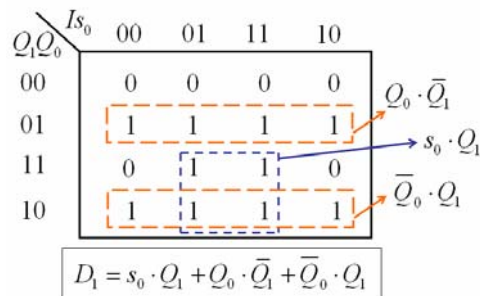
Nótese que la señal de condición s_1 que aparece en el diagrama no se ha utilizado en la construcción de esta tabla ya que realmente dicha señal no hace falta, ya que independientemente de su valor del estado S_2 siempre se pasa al estado S_3 .

En la implementación de una UC usando elementos de memoria tipo D, la entrada de cada uno de estos elementos es la variable asociada al próximo estado:

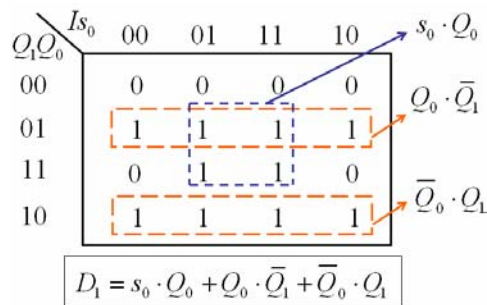
$$D_0 = Q'_0$$

$$D_1 = Q'_1$$

Usando la técnica de los mapas de Karnaugh es posible obtener la expresión lógica de D_1 .



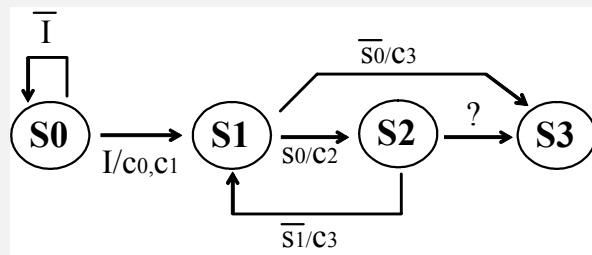
Aunque también sería válida la siguiente expresión:



Luego la afirmación correcta es la C.

SOLUCION PROBLEMA 5.31

- Unidad de control implementada con: MUX (2^2), registro de 2 bits, ROM de 2^3 palabras x 6 (bits/palabra)



Para resolver este problema conviene recordar que si se usa una ROM, un MUX y un registro para implementar una Unidad de Control es necesario que éstos elementos tengan las siguientes características (n nº de variables de estado, m nº de señales de control, y q número de señales de condición):

Caso 1: Multiplexor de selección por estado

- Capacidad de la ROM: $C \geq 2^{n+1}$ palabras x $(n+m)$ (bits/palabra)
- Registro de $n \geq 2$ bits
- MUX ($2^n:1$)

Caso 2: Multiplexor de selección por campo

- Capacidad de la ROM: $C \geq 2^{n+1}$ palabras x $(p+n+m)$ (bits/palabra)
- Registro de $(p+n)$ bits
- MUX ($q:1$)

Siendo p el número de bits necesarios para codificar las q señales de condición.

Además tanto en el caso 1 como en el caso 2 para poder implementar una unidad de control con estos elementos es necesario que las transiciones entre estados están regidas por una única señal de condición.

De la observación del diagrama de estados se deduce que la unidad de control posee cuatro estados, luego se requieren de dos variables de estado para codificarlos con lo que $n=2$.

Si se analizan las características de los dispositivos de que se disponen en el enunciado se observa que el registro es de 2 bits lo que descarta la implementación de la unidad de control según el caso 2. Por otra parte la ROM del enunciado dispone de 6 bits/palabra. Como la implementación se debe hacer según el caso 1, eso implica que

$$n+m=6$$

Como $n=2$, entonces $m=4$, es decir, como máximo sólo pueden existir cuatro señales de control (c_0, c_1, c_2, c_3). Luego la respuesta B no es correcta ya que considera la existencia de una quinta señal de control c_4 .

La respuesta B considera que el paso del estado S_2 a S_3 se regula con la señal de condición s_2 , pero del diagrama se sabe que el paso del estado S_2 a S_1 se regula con la señal de condición s_1 . Por lo tanto se tendrían dos señales de condición (s_2 y s_1) asociadas a un estado (S_2) con lo que se incumpliría la condición de que para poder implementar una unidad de control usando una ROM, registro y MUX es necesario que las transiciones entre estados están regidas por una única señal de condición.. Luego la respuesta B no es correcta. En conclusión la respuesta correcta es la **D**.

SOLUCION PROBLEMA 5.32

- Unidad de control implementada con dos elementos de memoria tipo D, uno por cada variable de estado.
- Hay dos variables de estado (Q_1, Q_0).
- La codificación de los estados es $S_0: (Q_1=0, Q_0=0)$, $S_1: (Q_1=0, Q_0=1)$, $S_2: (Q_1=1, Q_0=0)$.

Estado presente $Q_1 \ Q_0$	Próximo estado			
	$\bar{I} \ \bar{s}_0$	$\bar{I} \ s_0$	$I \ \bar{s}_0$	$I \ s_0$
00	00	00	01	01
01	11	10	11	10
10	11	01	11	01
11	11	00	11	00

Este problema puede resolverse de dos formas:

Forma 1: Usando la técnica de los mapas de Karnaugh para obtener la expresión lógica de D_0 y de D_1 .

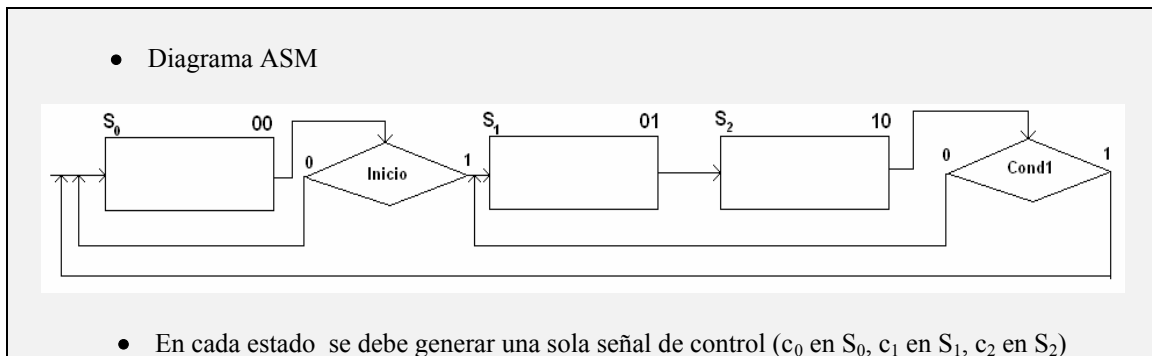
Forma 2: Evaluar las funciones propuestas para D_1 y D_0 en las afirmaciones y comprobar que efectivamente el próximo estado para cada caso coincide con el indicado en la tabla del enunciado. Con que existe un caso en el que sea distinto la expresión será falsa.

Se va resolver de la forma 2:

Estado actual		Señales de condición		Próximo estado		Evaluación de las funciones propuestas	
$Q_1(t)$	$Q_0(t)$	I	s_0	Q_1'	Q_0'	$D_1 = \bar{Q}_1 \cdot Q_0 + Q_1 \cdot \bar{s}_0$	$D_0 = Q_1 + I \cdot (\bar{s}_0 + \bar{Q}_0)$
0	0	0	0	0	0	$1 \cdot 0 + 0 \cdot 1 = 0$	$0 + 0 \cdot (1 + 1) = 0$
0	0	0	1	0	0	$1 \cdot 0 + 0 \cdot 0 = 0$	$0 + 0 \cdot (0 + 1) = 0$
0	0	1	0	0	1	$1 \cdot 0 + 0 \cdot 1 = 0$	$0 + 1 \cdot (1 + 1) = 1$
0	0	1	1	0	1	$1 \cdot 0 + 0 \cdot 0 = 0$	$0 + 1 \cdot (0 + 1) = 1$
0	1	0	0	1	1	$1 \cdot 1 + 0 \cdot 1 = 1$	$0 + 0 \cdot (1 + 0) = 0$
0	1	0	1	1	0	$1 \cdot 1 + 0 \cdot 0 = 1$	
0	1	1	0	1	1	$1 \cdot 1 + 0 \cdot 1 = 1$	
0	1	1	1	1	0	$1 \cdot 1 + 0 \cdot 0 = 1$	
1	0	0	0	1	1	$0 \cdot 0 + 1 \cdot 1 = 1$	
1	0	0	1	0	1	$0 \cdot 0 + 1 \cdot 0 = 0$	
1	0	1	0	1	1	$0 \cdot 0 + 1 \cdot 1 = 1$	
1	0	1	1	0	1	$0 \cdot 0 + 1 \cdot 0 = 0$	
1	1	0	0	1	1	$0 \cdot 1 + 1 \cdot 1 = 1$	
1	1	0	1	0	0	$0 \cdot 1 + 1 \cdot 0 = 0$	
1	1	1	0	1	1	$0 \cdot 1 + 1 \cdot 1 = 1$	
1	1	1	1	0	0	$0 \cdot 1 + 1 \cdot 0 = 0$	

De acuerdo con la tabla construida existe al menos un caso donde D_0 difiere de Q_0' , por eso no ha sido necesario seguir evaluando la función en los demás casos. Luego la afirmación I es **FALSA**.

Por otra parte D_1 es igual a Q_1' en todos los posibles casos, en conclusión la afirmación II es **VERDADERA**.

SOLUCION PROBLEMA 5.33

De acuerdo con el diagrama ASM adjunto esta unidad de control tiene:

- $N=3$ estados (S_0, S_1, S_2) $\rightarrow 3 \leq 2^n \rightarrow n=2$ variables de estado (Q_1, Q_0).
- $q=2$ señales de condición (Inicio, Cond1).

Además según el enunciado

- $m=3$ señales de control (c_0, c_1, c_2)

Analicemos la veracidad de las afirmaciones:

I) Para implementar una unidad de control empleando una ROM y un registro, es necesario que éstos elementos tengan las siguientes características:

Capacidad de la ROM:

$$C \geq 2^{n+q} \text{ palabras} \times (n+m) \text{ (bits/palabra)} \geq 2^4 \text{ palabras} \times 5 \text{ (bits/pal)}$$

Registro de $n \geq 2$ bits

Puesto que la ROM de la afirmación posee únicamente 8 palabras, ocho menos de las necesarias esta afirmación es **FALSA**.

II) Si se usa una ROM, un MUX de selección por estado y un registro es necesario que éstos elementos tengan las siguientes características:

Capacidad de la ROM:

$$C \geq 2^{n+1} \text{ palabras} \times (n+m) \text{ (bits/palabra)} \geq 2^3 \text{ palabras} \times 5 \text{ (bits/pal)}$$

Registro de $n \geq 2$ bits

$$\text{MUX } (2^n:1) \rightarrow \text{MUX}(2^2:1) \rightarrow \text{MUX}(4:1)$$

De acuerdo con el diagrama ASM las transiciones entre estados están regidas por una única señal de condición, luego es posible implementar la unidad de control usando una ROM, un registro y un MUX. Además puesto que el MUX del enunciado es del tipo 4:1 y la capacidad de la ROM del enunciado es de 8 palabras \times (8 bits/palabra). Se observa que todos los elementos proporcionados poseen las características adecuadas para poder implementar la unidad de control. En conclusión la afirmación es **VERDADERA**.