DISEÑO DE TRANSFERENCIA DE REGISTROS

Partes de un sistema digital

- Unidad de procesamiento: Se almacenan y transforman los datos
- Unidad de control: Genera las secuencias se señales de control de acuerdo al algoritmo de transferencia de registros.

Elementos de la Unidad de Procesamiento

- Registros de almacenamiento
- Operadores ALU
- Red de interconexión
- Puntos de control
- Señales de condición

Alternativas para el diseño de la Unidad de Control

- Elementos memoria tipo D
- *Registros de secuencia y decodificador. (No hay que estudiar)*
- Un elemento de memoria por estado
- Un registro de estado y una memoria ROM
- *Contador y decodificador. (No hay que estudiar)*
- Un registro de estado y un Array Lógico Programable (PLA)

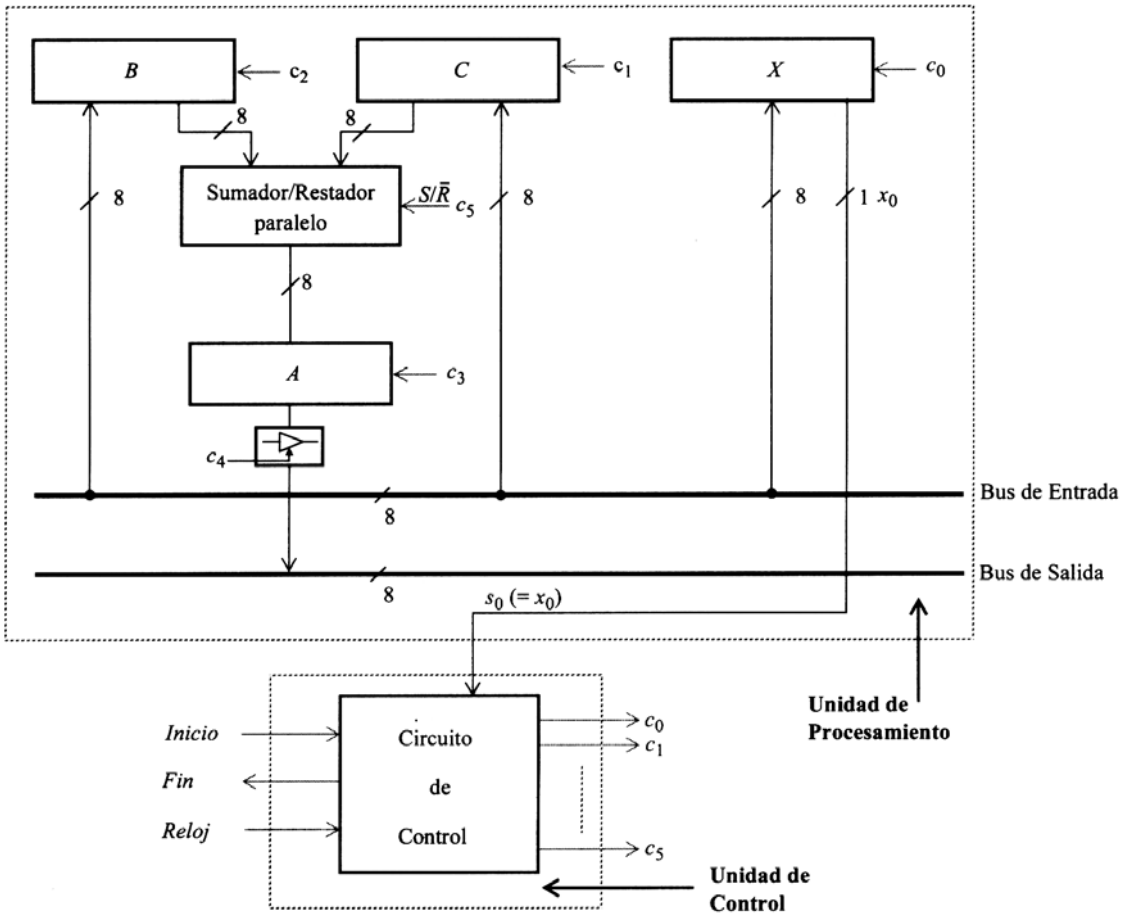
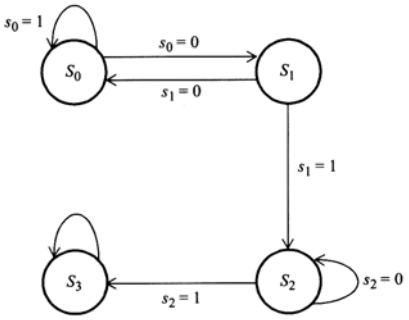


Diagrama de bloques y puntos de control de la Unidad de Procesamiento A

Ejemplo:

Diseñar la Unidad de control de un sistema digital que dispone de tres entradas s_0,s_1 y s_2 y su diagrama de estados es el siguiente y que en cada estado s_i este sistema activa únicamente la señal de control c_i



Diseño de la unidad de control utilizando elementos de memoria tipo D

1. Obtención del diagrama de estados.
2. Obtención de la tabla de estados.
3. Síntesis de las funciones de conmutación de las entradas de cada elemento memoria tipo D en función de sus salidas y de las señales de condición
4. Síntesis del circuito lógico

Tabla de estados:

El número de básculas será el necesario para que elevando 2 a dicho número dé como resultado el valor igual o inmediatamente superior al número de estados. Ebn el ejemplo expuesto como el número de estados es 4, el número de básculas será de 2

Entradas			Est. actual		Est. próximo		Salidas			
s_2	s_1	s_0	Q_1	Q_0	Q_{1t+1}	Q_{0t+1}	c_3	c_2	c_1	c_0
x	x	1	0	0	0	0	Para todo $s_i \rightarrow c_i$ En este caso las salidas son c_3, c_2, c_1 c_0 y para codificarlas se colocará un codificador de 2 a 4			
x	x	0	0	0	0	1				
x	0	x	0	1	0	0				
x	1	x	0	1	1	0				
0	x	x	1	0	1	0				
1	x	x	1	0	1	1				
x	x	x	1	1	1	1				

Funciones para las entradas de las básculas D:

Al utilizar básculas D, el valor de la entrada de cada báscula será el mismo valor que la correspondiente salida Q_{it+1}
El siguiente paso es simplificar mediante tablas de Karnaugh las correspondientes funciones canónicas:

D_0

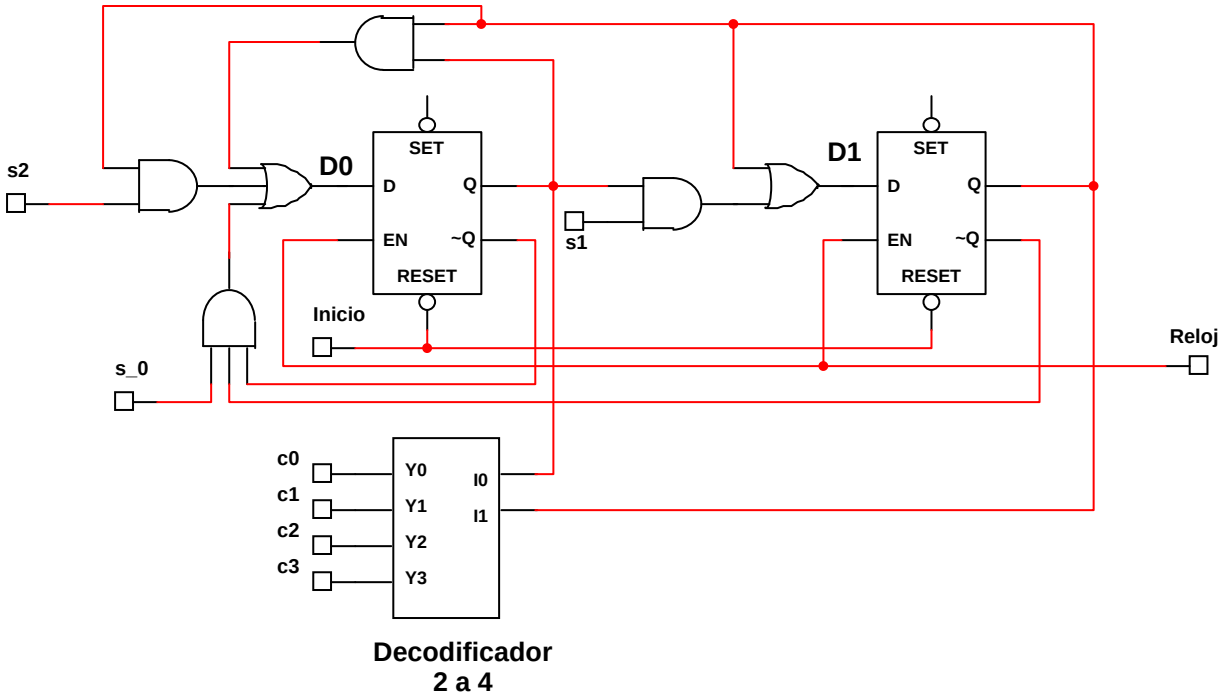
			$\overline{Q_1}$		Q_1	
			$\overline{Q_0}$	Q_0	Q_0	$\overline{Q_0}$
$\overline{s_2}$	$\overline{s_1}$	$\overline{s_0}$	1		1	
		s_0			1	
	s_1				1	
		$\overline{s_0}$	1		1	
s_2			1		1	1
		s_0			1	1
	$\overline{s_1}$				1	1
		$\overline{s_0}$	1		1	1

$$D_0 = Q_1Q_0 + s_2Q_1 + \overline{s_0}\,\overline{Q_1}\,\overline{Q_0}$$

D_1

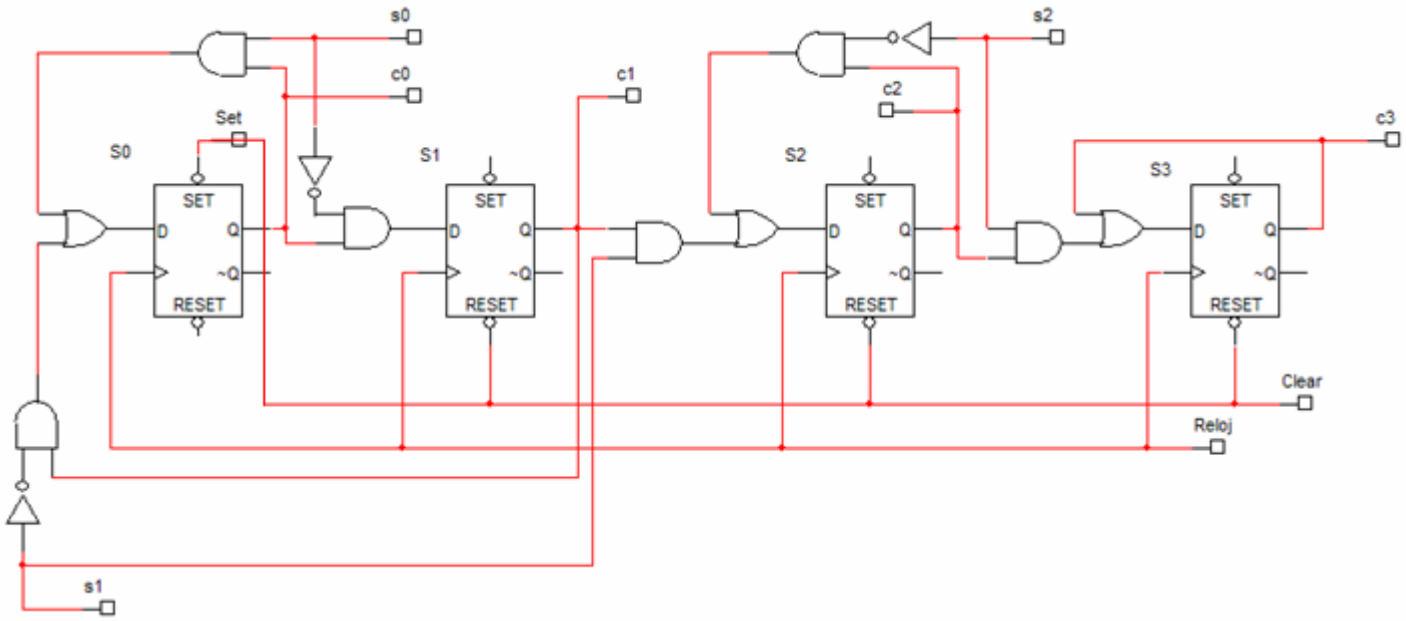
			$\overline{Q_1}$		Q_1	
			$\overline{Q_0}$	Q_0	Q_0	$\overline{Q_0}$
$\overline{s_2}$	$\overline{s_1}$	$\overline{s_0}$			1	1
		s_0			1	1
	s_1			1	1	1
		$\overline{s_0}$		1	1	1
s_2				1	1	1
		s_0		1	1	1
	$\overline{s_1}$				1	1
		$\overline{s_0}$			1	1

$$D_1 = Q_1 + s_1Q_0$$



Diseño de la unidad de control utilizando una báscula D por cada estado

El esquema se obtiene directamente del diagrama de estados, colocando una báscula D por cada uno de los estados y en la entrada de cada estado un multiplexor o una puerta “OR” de forma que la entrada viene de cada uno de los estado origen. Las señales de control se obtienen en base a colocar puertas “OR” desde cada uno de los estados que activan a cada una de ellas.



Diseño de la unidad de control utilizando una ROM y un registro:

Se utiliza una memoria ROM en la que en el bus de direcciones se conectan las entradas y las salidas del registro que indican el estado actual.
En cada una de las posiciones de memoria de la ROM se escribe la información correspondiente a varios campos, por un lado el campo correspondiente al próximo estado y por otro el correspondiente a las señales de control.

Opciones de la implementación por ROM y registro

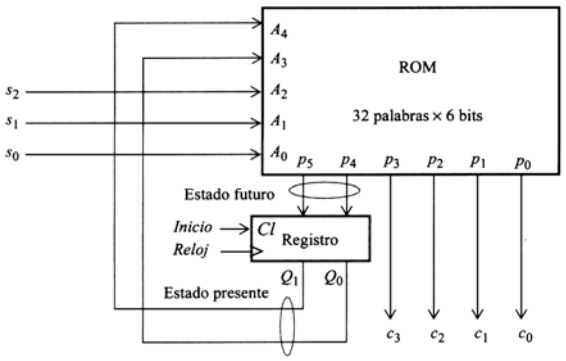
- Memoria ROM y registro
- Memoria ROM, registro y multiplexor.
  - Selección por estado.
  - Selección por campo.

Memoria ROM y registro:

Entradas → S_0,S_1 *y* S_2	Bus de direcciones ROM → 5 bits
Estados → 4 → Q_1 Q_0	

Salidas de control → C_3,C_2, C_1, C_0	Bus de datos ROM → 6 bits
Estados → 4 → $\mathbf{Q_{1t+1}}$ $\mathbf{Q_{0t+1}}$	

Circuito:



Unidad de Control diseñada con una memoria ROM y un registro

Tabla de contenido de la ROM:

Dirección de la ROM $A_4\,A_3\,A_2\,A_1\,A_0$	**Contenido de la ROM** $p_6\,p_5\,p_4\,p_3\,p_2\,p_1\,p_0$	
Estado presente + Cond. $Q_1\,Q_0\,s_2\,s_1\,s_0$	Próximo Estado $Q_1\,Q_0$	Control $S_3\,S_2\,S_1\,S_0$
0 0 0 0 0	0 1	0 0 0 1
0 0 0 0 1	0 0	0 0 0 1
0 0 0 1 0	0 1	0 0 0 1
0 0 0 1 1	0 0	0 0 0 1
0 0 1 0 0	0 1	0 0 0 1
0 0 1 0 1	0 0	0 0 0 1
0 0 1 1 0	0 1	0 0 0 1
0 0 1 1 1	0 0	0 0 0 1
0 1 0 0 0	0 0	0 0 1 0
0 1 0 0 1	0 0	0 0 1 0
0 1 0 1 0	1 0	0 0 1 0
0 1 0 1 1	1 0	0 0 1 0
0 1 1 0 0	0 0	0 0 1 0
0 1 1 0 1	0 0	0 0 1 0
0 1 1 1 0	1 0	0 0 1 0
0 1 1 1 1	1 0	0 0 1 0
1 0 0 0 0	1 0	0 1 0 0
1 0 0 0 1	1 0	0 1 0 0
1 0 0 1 0	1 0	0 1 0 0
1 0 0 1 1	1 0	0 1 0 0
1 0 1 0 0	1 1	0 1 0 0
1 0 1 0 1	1 1	0 1 0 0
1 0 1 1 0	1 1	0 1 0 0
1 0 1 1 1	1 1	0 1 0 0
1 1 0 0 0	1 1	1 0 0 0
1 1 0 0 1	1 1	1 0 0 0
1 1 0 1 0	1 1	1 0 0 0
1 1 0 1 1	1 1	1 0 0 0
1 1 1 0 0	1 1	1 0 0 0
1 1 1 0 1	1 1	1 0 0 0
1 1 1 1 0	1 1	1 0 0 0
1 1 1 1 1	1 1	1 0 0 0

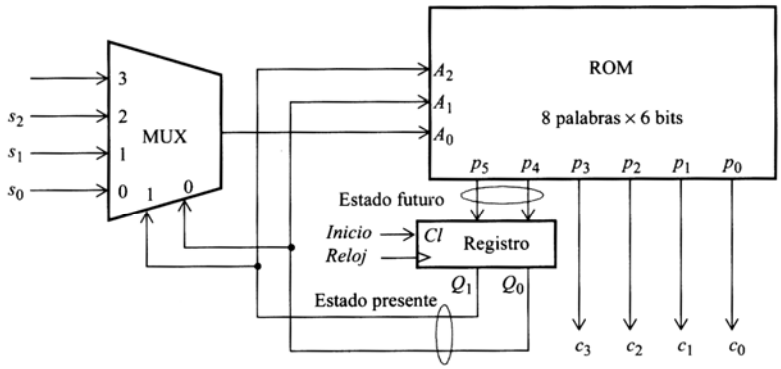
Contenido de la memoria ROM de la Unidad de Control

Memoria ROM, registro y un multiplexor. Selección por estado.

Cuando el paso de un estado a otro depende únicamente de una sola variable en cada caso , se puede reducir el número de bits del bus de direcciones en base a colocar un multiplexor en las entradas de condiciones de paso y elegir con la configuración de los estados la entrada del multiplexor que provocará el cambio de estado.

Entradas → s_2, s_1, s_0 → Multiplexor de 4 canales → 1 bit (s_i) en el bus de direcciones de la ROM	Bus direcciones de la ROM → 3 bits
Estados → 4 → Q1 Q0 → 2 bits	

Salidas → $c_3, c_2, c_1 c_0$ → 4 bit en el bus de direcciones de la ROM	Bus datos de la ROM → 6 bits
Estados → 4 → $Q_{1t+1} Q_{0t+1}$ → 2 bits	



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

Estado Presente	Dirección de la ROM $A_2 A_1 A_0$		Contenido de la ROM $P_5 P_4 P_3 P_2 P_1 P_0$	
	Estado presente + Cond. $Q_1 Q_0 s_i$		Próximo Estado $Q_1 Q_0$	Control $S_3 S_2 S_1 S_0$
S_0	0 0 0	(si $s_0 = 0$)	0 1	0 0 0 1
	0 0 1	(si $s_0 = 1$)	0 0	0 0 0 1
S_1	0 1 0	(si $s_1 = 0$)	0 0	0 0 1 0
	0 1 1	(si $s_1 = 1$)	1 0	0 0 1 0
S_2	1 0 1	(si $s_2 = 1$)	1 1	0 1 0 0
	1 0 0	(si $s_2 = 0$)	1 0	0 1 0 0
S_3	1 1 0		1 1	1 0 0 0
	1 1 1		1 1	1 0 0 0

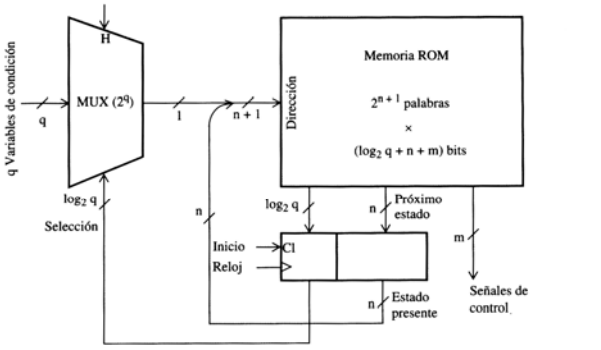
Contenido de la memoria ROM (método de selección por estado)

Memoria ROM, registro y un multiplexor. Selección por campo.

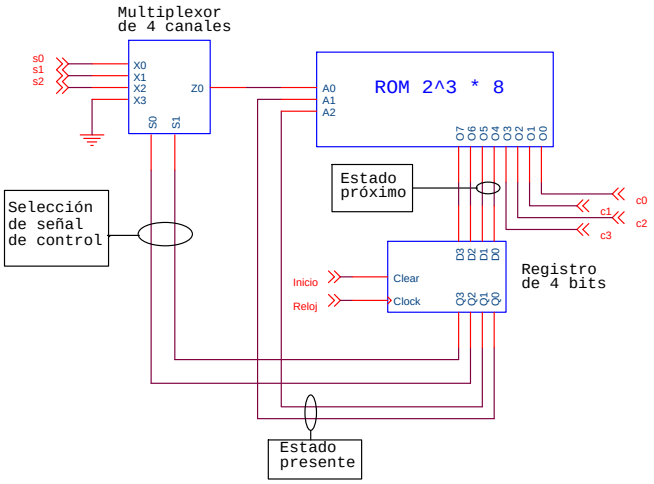
En el caso anterior el multiplexor ha de tener tantos canales (entradas) como estados tenga el sistema, resultando esto a veces muy engorroso. Una forma de solucionar este problema consiste en reservar en la memoria ROM un tercer campo en el que se escribirá el número de la condición de entrada que provoca el cambio de estado. Este nuevo campo actúa al selector del multiplexor y de esta manera hace falta solamente un multiplexor con tantos canales como señales de condición, en contrapartida la ROM se verá ampliada en su anchura de palabra.

Entradas → s_2, s_1, s_0 → Multiplexor de 4 canales → 2 bits en el campo de datos de la ROM Entradas → s_2, s_1, s_0 → Multiplexor de 4 canales → 1 bit (s_i) en el bus de direcciones de la ROM Estados → 4 → Q1 Q0 → 2 bits	Bus direcciones de la ROM → 3 bits

Salidas → $c_3, c_2, c_1 c_0$ → 4 bit en el bus de direcciones de la ROM Estados → 4 → $Q_{1t+1} Q_{0t+1}$ → 2 bits Campo de selección señal control → 2 bits	Bus datos de la ROM → 8 bits



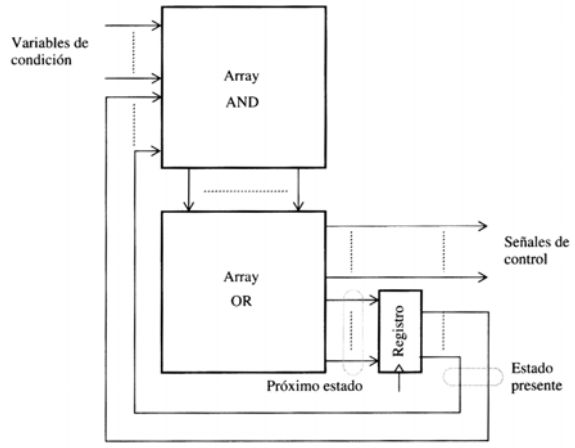
Unidad de control con una memoria ROM y un multiplexor (método de selección por campo)



A_2	A_1	A_0		D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Q_1	Q_0	s_i	s_i	Cam_1	Cam_0	Q_{1t+1}	Q_{0t+1}	c_3	c_2	c_1	c_0
0	0	0	$s_0=0$	0	0	0	1	0	0	0	1
0	0	1	$s_0=1$	0	0	0	0	0	0	0	1
0	1	0	$s_1=0$	0	1	0	0	0	0	1	0
0	1	1	$s_1=1$	0	1	1	0	0	0	1	0
1	0	0	$s_2=0$	1	0	1	0	0	1	0	0
1	0	1	$s_2=1$	1	0	1	1	0	1	0	0
1	1	0	1	1	1	1	1	1	0	0	0
1	1	1	1	1	1	1	1	1	0	0	0

En estos dos bits se codifica el número que corresponda a la señal de condición de paso

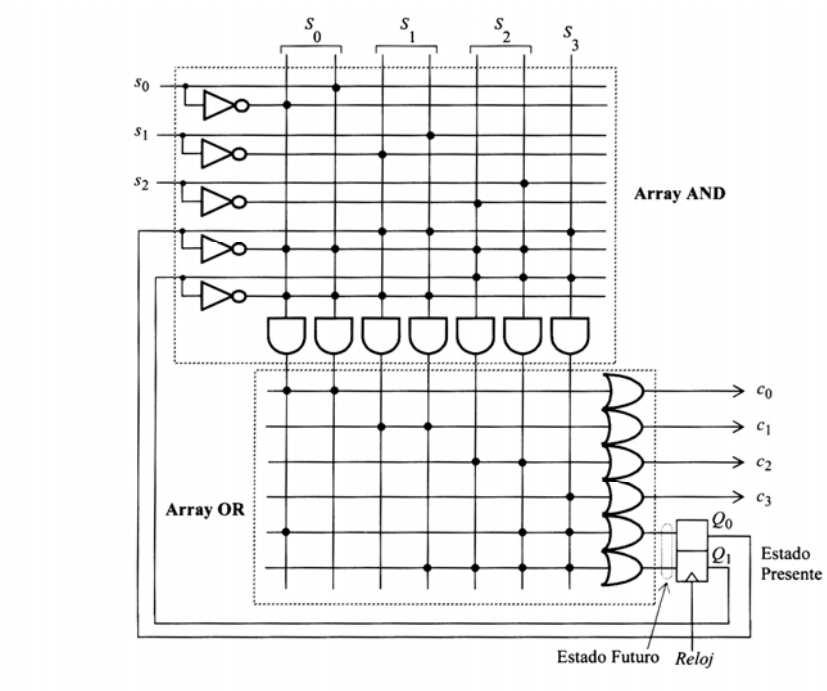
Diseño de la unidad de control utilizando un registro de estado y un Array Lógico Programable (PLA)



Entradas → ***S2, S1, S0***	Entradas a la PLA → 3 bits
Estados → 4 → Q1 Q0 → 2 bits	

		Células AND
Entradas → ***S2, S1, S0***	Estado 0 / condiciones de paso ***s0=0 y s0=1***	2
Estados → 4 → Q1 Q0 → 2 bits	Estado 1 / condiciones de paso ***s1=0 y s1=1***	2
	Estado 2 / condiciones de paso ***s2=0 y s2=1***	2
	Estado 3 / condiciones de paso ***ninguna***	1
	Total	7

	Células OR
Salidas → ***c3, c2, c1 c0***	4
Estados → 4 → Q1 Q0 → 2 bits	2
Total	6



Unidad de Control diseñada con una PLA

2001 Junio 1ª Semana

4.- Se desea diseñar con memoria ROM una Unidad de Control con 200 estados, que genere 37 señales de control totalmente independientes, y que reciba 9 señales de condición pero en cada estado va a ser consultada como máximo una de ellas. Utilizando en el diseño un multiplexor con *selección por campo* haría falta un multiplexor con:

A) 3 entradas de selección. B) 4 entradas de datos. C) 16 entradas de datos. D) Ninguna de la anteriores.

2002 Septiembre

Problema –

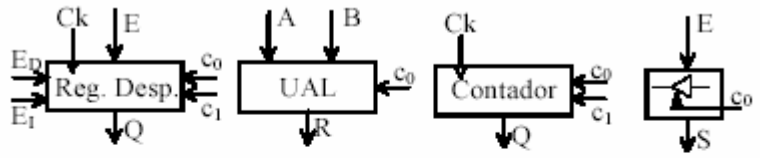
El siguiente algoritmo describe una determinada operación de un sistema digital.

A) (2 puntos) Diseñar la Unidad de Procesamiento que permita realizar este algoritmo utilizando, si son necesarios, los módulos dibujados abajo: registros de desplazamiento de 8 bits, una UAL con dos entradas de 8 bits cada una, un contador módulo-8 y circuitos triestado de conexión unidireccional con control de 8 bits; además de puertas lógicas y los módulos combinacionales (MUX, DMUX, codificadores y decodificadores) que considere necesarios. Debe tener en cuenta que al bus vuelcan datos múltiples dispositivos.

B) (2 puntos) Diseñar la Unidad de Control que ejecute este algoritmo con la Unidad de Procesamiento diseñada en el apartado A) empleando un elemento de memoria tipo D por estado. **Detalle y explique claramente** todos y cada uno de los pasos seguidos hasta obtener la solución.

Nota: A_0 es el bit menos significativo de A.

1: Declaración: A[8], B[8], Cont[3];
2: A ← Bus;
3: B ←1;
4: for Cont=0 to 3 do
5: if $A_1A_0 \neq 01$ then
6: B ←0;
7: endif;
8: Despl.CerradoDcha(A);
9: Despl.CerradoDcha(A);
10: endfor;
11: Bus ←B;
12: Parar;



c_1c_0	Q(t+1)
00	Q(t)
01	Des. Dcha.
10	Des. Izq.
11	Carga

c_0	R
0	A+B
1	A-B

c_1c_0	Q(t+1)
00	Q(t)
01	0
10	Q(t)+1 mod 8
11	Q(t)

c_0	S
0	-
1	E

Figura 1: Módulos secuenciales del problema con sus tablas de funcionamiento.

Elementos:

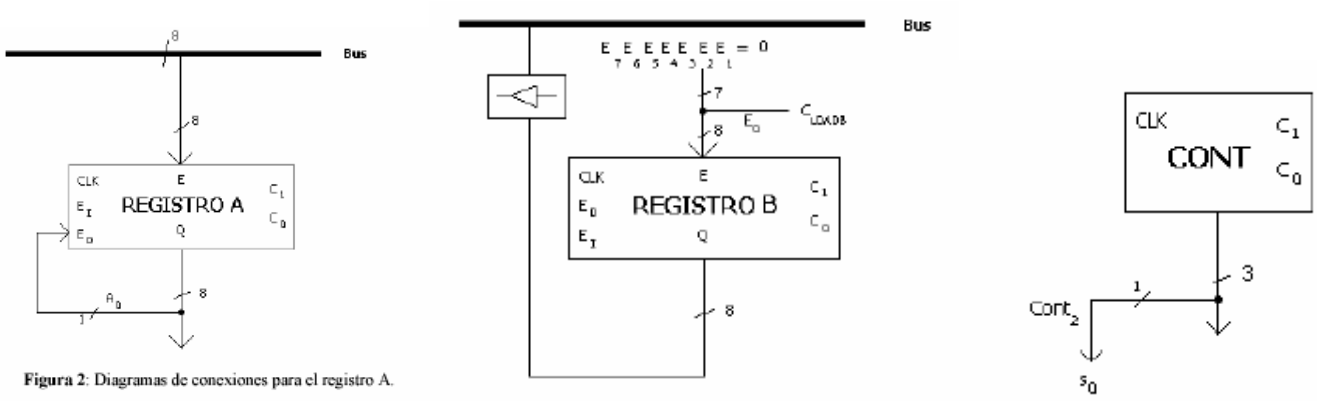


Figura 2: Diagramas de conexiones para el registro A.

Señal de control	Operación a realizar	Valor
C_{A0}, C_{A1}	Cargar registro A	11
C_{A0}, C_{A1}	Desplazamiento cerrado a la derecha de A	10
C_B,	Cargar registro B con 0	1
C_B, C_{LOADB}	Cargar registro B con 1	11
C_{C0}, C_{C1}	Resetear contador	10
C_{C0}, C_{C1}	Incrementar módulo 8 una unidad el contador.	01
C_{BUS}	Volcar el contenido del registro B en el bus	1

Tabla 1: Señales de control que debe generar la Unidad de Control para cada operación.

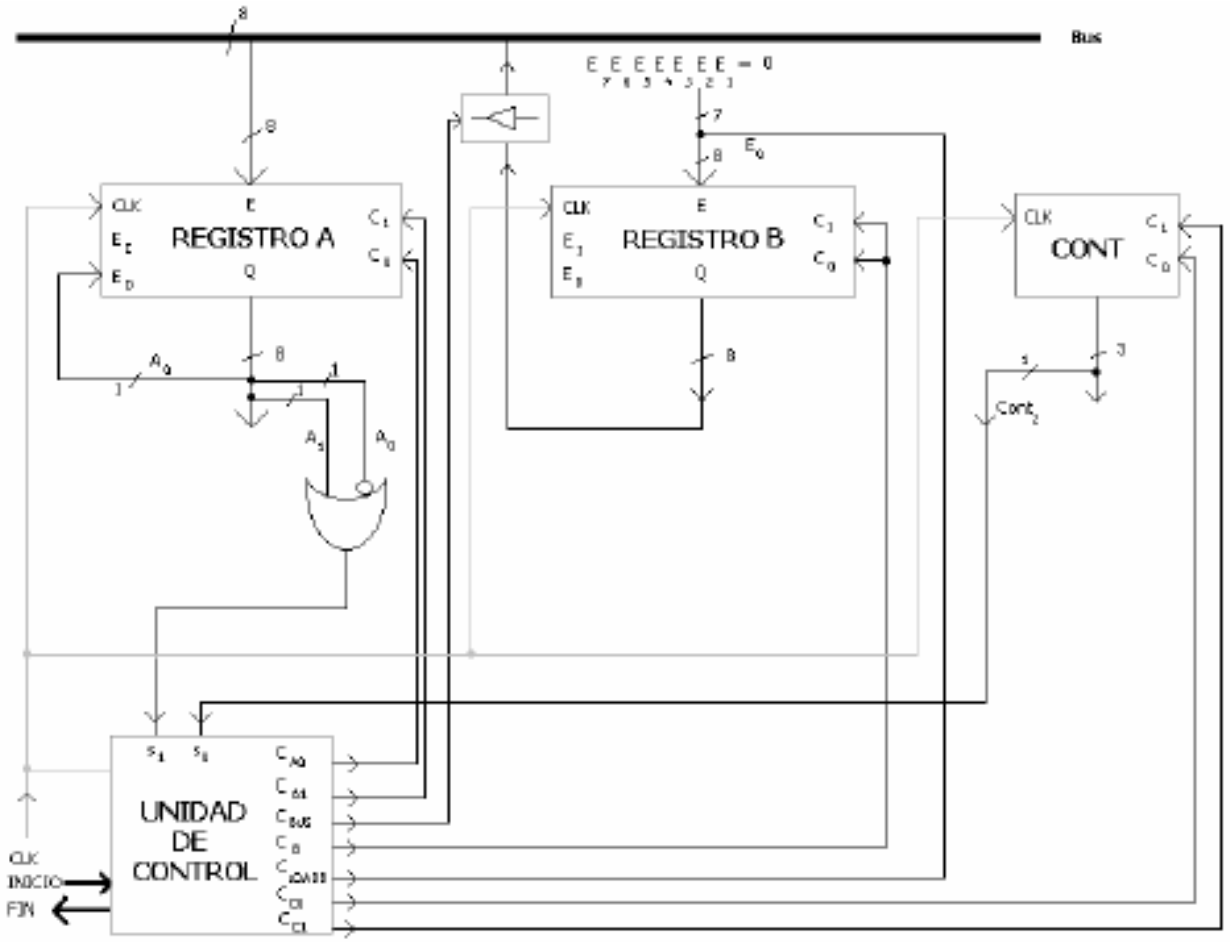


Figura 6: Diseño de la Unidad de Procesamiento.

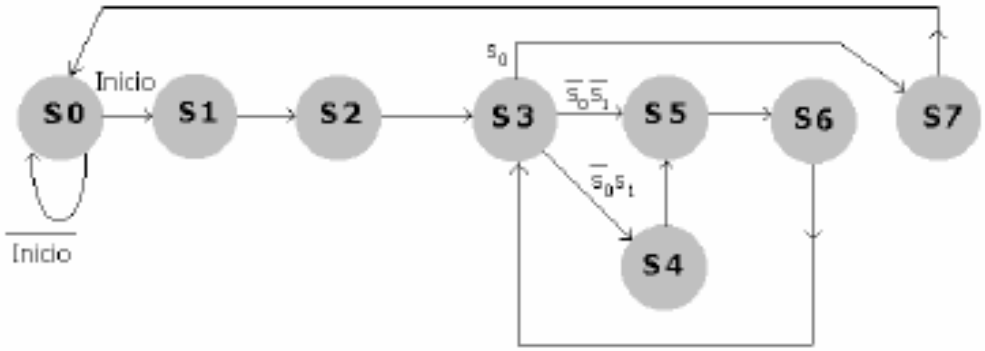


Figura 7: Diagrama de estados de la unidad de control.

Estado de la unidad de control	Microoperaciones efectuadas	Señales de control a activar
S_0	*Ninguna (Estado inicial)*	*Ninguna*
S_1	A ←Bus, Cont ←0	C_{A0}, C_{A1}, C_{C0}
S_2	B ←1	C_B, C_{LOADB}
S_3	*Ninguna (Comprobamos condiciones s_1 y s_0)*	*Ninguna*
S_4	B←0	C_B
S_5	Despl.CerradoDcha(A)	C_{A0}
S_6	Despl.CerradoDcha(A), Cont ←Cont+1 (mod 8)	C_{A0}, C_{C1}
S_7	Bus ← B	C_{BUS}

Tabla 2: Acciones tomadas por la Unidad de Control en cada estado.

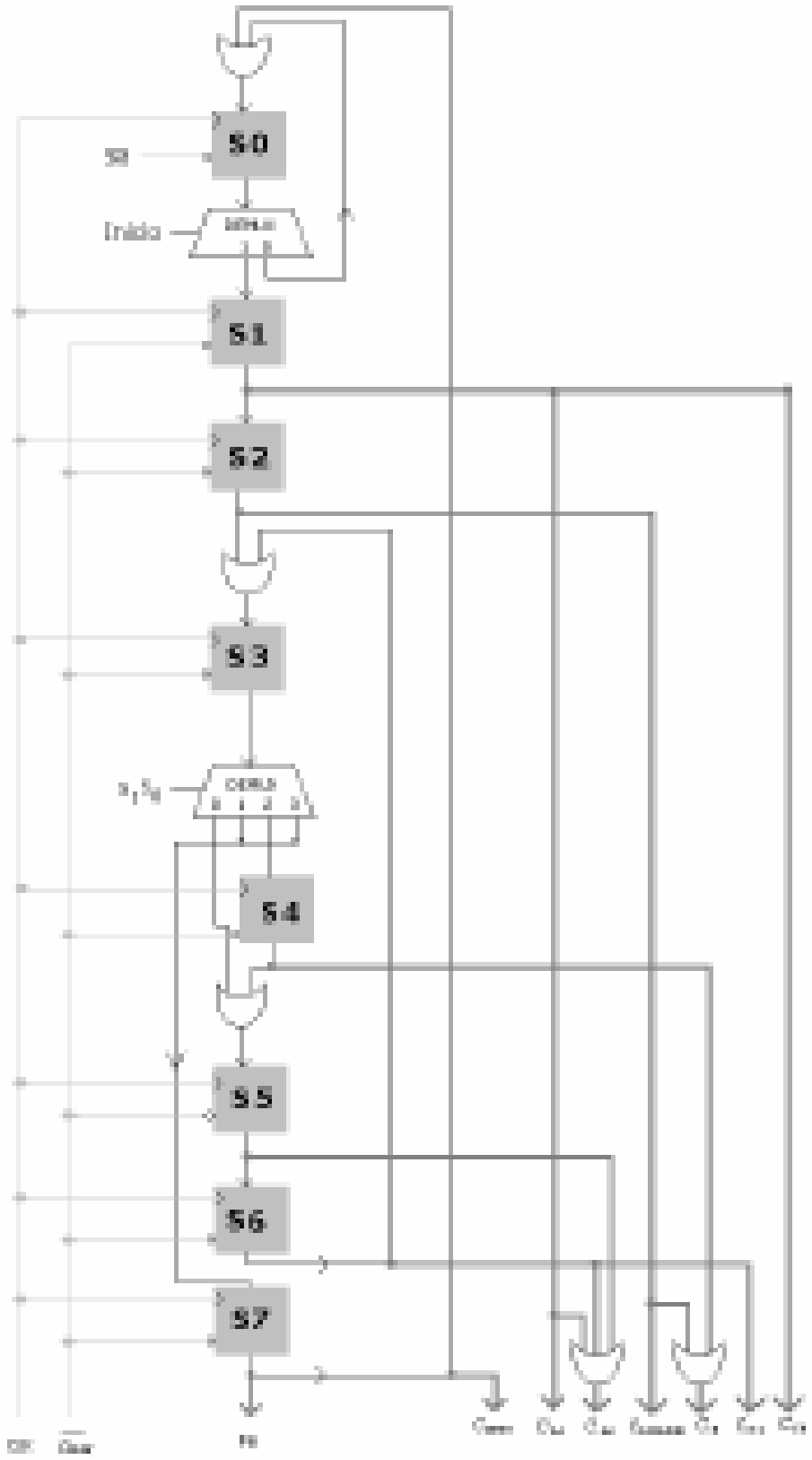
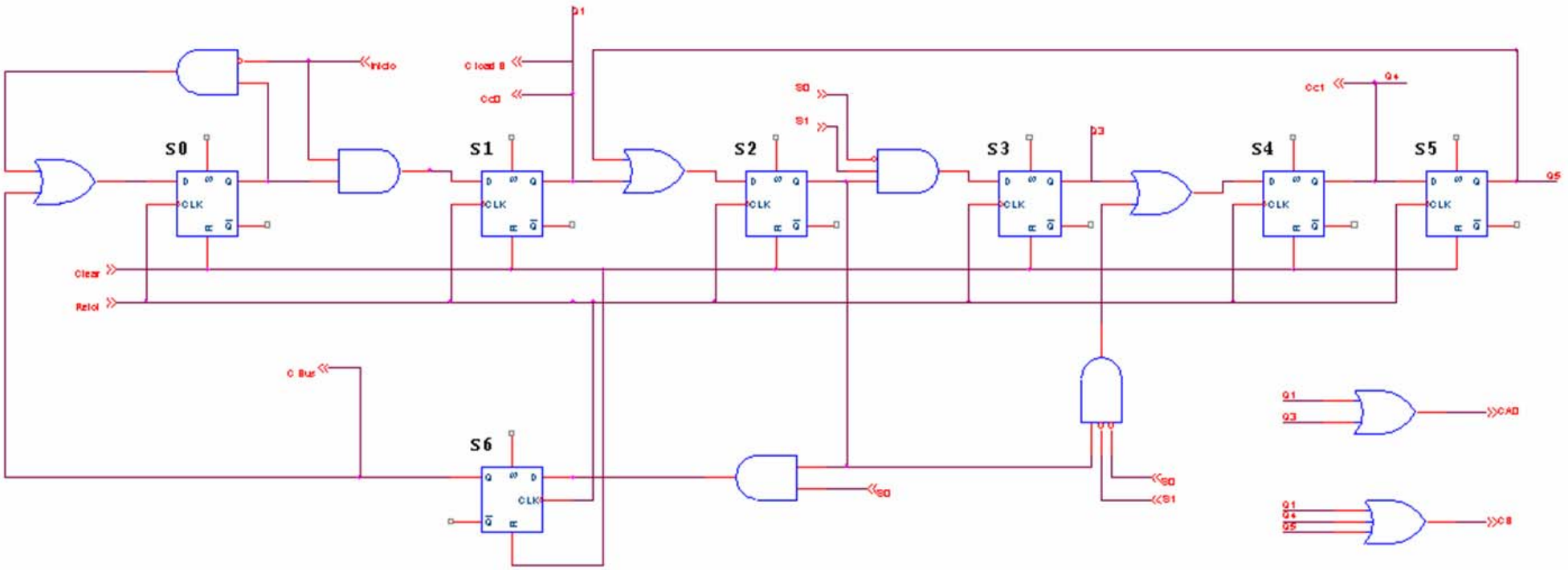


Figura 5: Unidad de Control mediante elementos de memoria.


S0
S1
S2
S3
S4
S5
S6
CLK
Clear
Reloj
Inicio
Q1
Q3
Q4
Q5
S0
S1
CAD
CB

SEPTIEMBRE 2004 RESERVA

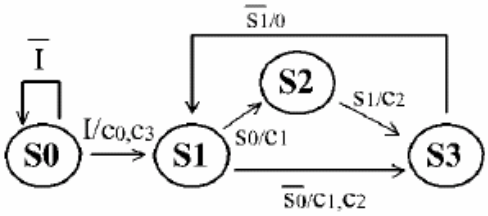
1.- En la figura adjunta se muestra el diagrama de estados de una unidad de control. Indique cuál de las siguientes afirmaciones es correcta:

A) Puede realizarse la síntesis de la unidad de control empleando un registro de 2 bits y una memoria ROM de 2^5 palabras, con 8 bits por palabra.

B) Puede realizarse la síntesis de la unidad de control empleando un registro de 2 bits, un multiplexor de 4 a 1, y una memoria ROM de 8 palabras, con 16 bits por palabra.

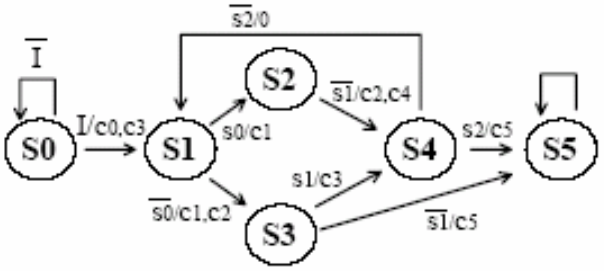
C) Las dos afirmaciones anteriores son correctas.

D) Todas las afirmaciones anteriores son falsas.



JUNIO 2004 2ª SEMANA

1.- En la figura adjunta se muestra el diagrama de estados de una unidad de control. Indique cuál de las siguientes afirmaciones, acerca de las señales de control (c_i), es correcta. .



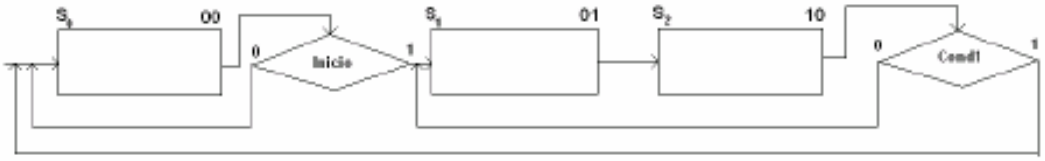
A) $c_3 = S_3 \cdot s_1$

B) $c_5 = S_4 \cdot \overline{s_2} + S_3 \cdot \overline{s_1}$

C) Las dos afirmaciones anteriores son correctas.

D) Todas las afirmaciones anteriores son falsas

6.- Dado el diagrama ASM de la Figura, indicar cuántos elementos de memoria tipo J-K son necesarios para implementarlo si se usa la técnica de diseño con un elemento de memoria por estado.



A) 3 B) 2 C) No se puede deducir del diagrama D) Ninguna de las anteriores