

Exámenes de "Diseño de transferencia de registros"

Problemas

Septiembre 2002

Diseñar la U.P. y la U.C. ^{Una "D" por estado} para cumplir el algoritmo:

1: Declaración: A[8], B[8], Cont[3]

2: A ← Bus

3: B ← 1

4: for Cont=0 to 3 do

5: if A, A₀ ≠ 0 then

6: B ← 0

7: end if

8: Desplaz. cerrado Dcha(A)

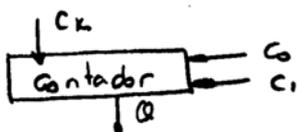
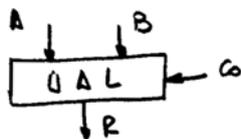
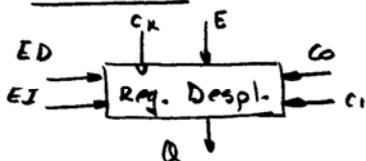
9: " " " (A)

10: end for

11: Bus ← B

12: Parar

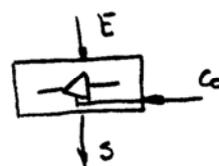
Elementos:



C ₀ , C ₁	Q(+1)
0 0	Q(+1)
0 1	Desp. Dcha
1 0	" Izda
1 1	Carga

C ₀	R
0	A+B
1	A-B

C ₀ , C ₁	Q(+1)
0 0	Q(+1)
0 1	0
1 0	Q(+1)+1 mod B
1 1	Q



C ₀	S
0	-
1	E

Unidad de procesamiento

- Elementos
- Reg A →
 - Analizar A₁ y A₀
 - Cargar desde BUS
 - Desplaz. cerrado a Dcha ⇒ E₀ unida a A₀
 - Reg B →
 - Ponerlo a 0
 - Sacarlo al bus
 - Contador →
 - puesta a 0
 - contar
 - Parar al llegar a 4

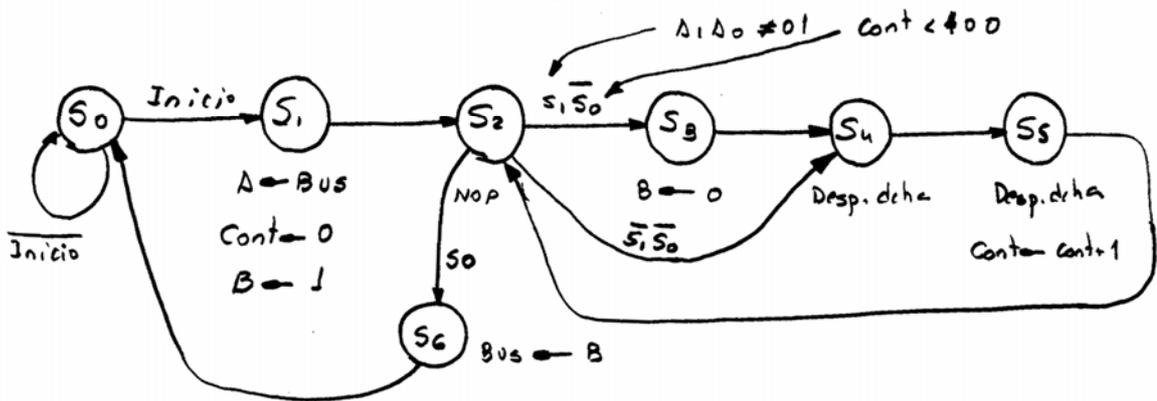
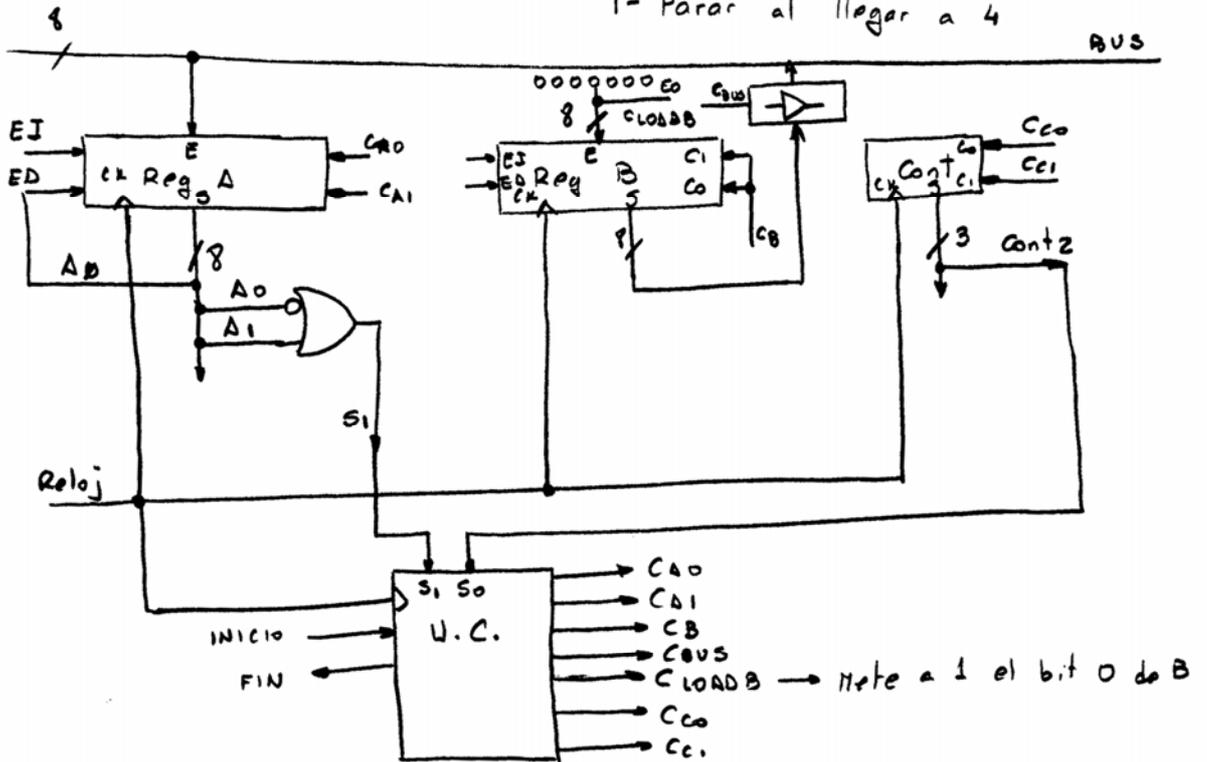
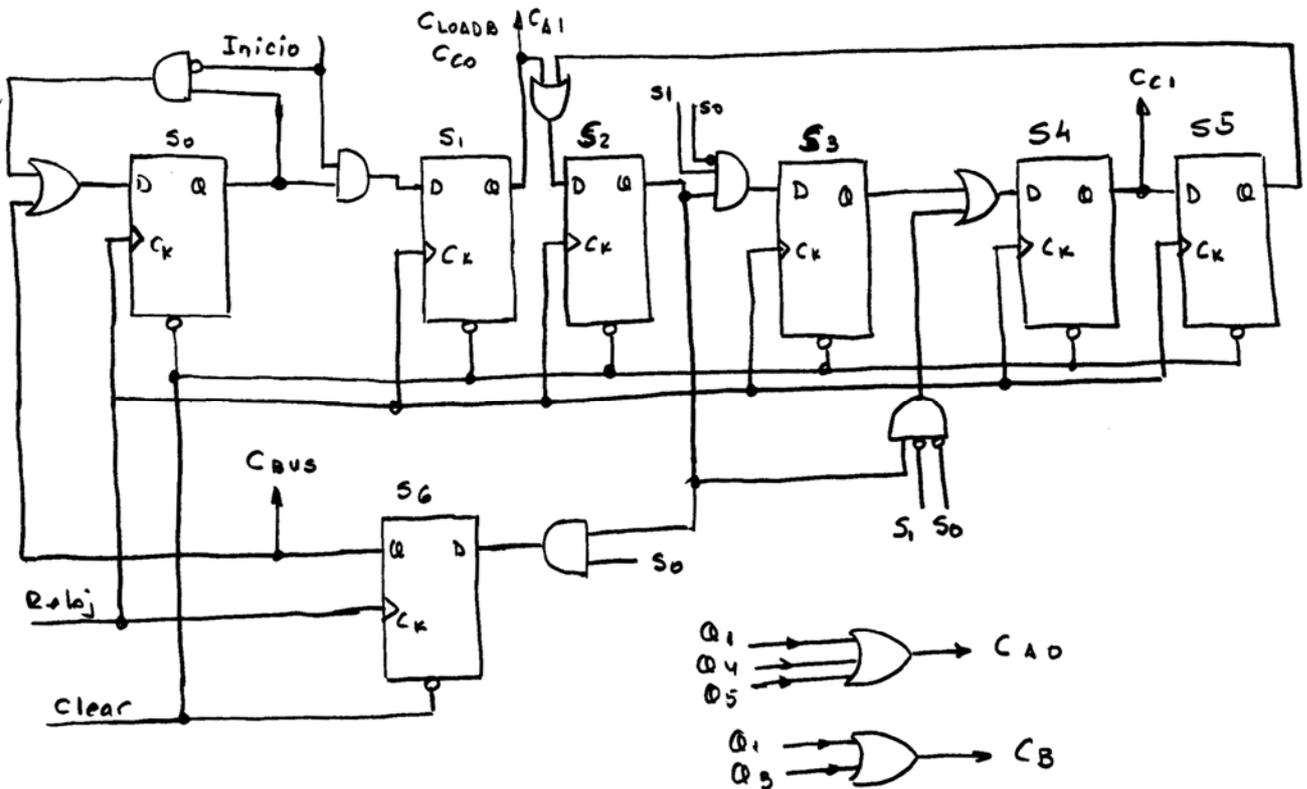


Tabla de estados

Entradas			Est. Act.	Est. Próximo			Salidas				
S ₁	S ₀	I	Q ₂ Q ₁ Q ₀	Q _{2,t+1} Q _{1,t+1} Q _{0,t+1}	C _{A1}	C _{A0}	C _B	C _{bus}	C _{LOADB}	C _{cc}	C _{co}
Al ser una D por estado sobre la codificación binaria											
S ₁	S ₀	I	Est. Actual	Est. Próximo	C _A	C _{A0}	C _B	C _{bus}	C _{LOADB}	C _c	C _{co}
x	x	0	0	0	0	0	0	0	0	0	0
x	x	1	0	1	0	0	0	0	0	0	0
x	x	x	1	2	1	1	1	0	1	0	0
x	1	x	2	6	0	0	0	0	0	0	0
1	0	x	2	3	0	0	0	0	0	0	0
0	0	x	2	4	0	0	0	0	0	0	0
x	x	x	3	4	0	0	1	0	0	0	0
x	x	x	4	5	0	1	0	0	0	1	0
x	x	x	5	2	0	1	0	0	0	0	0
x	x	x	6	0	0	0	0	1	0	0	0



Junio 2002 - 2^º S

Diseñar la U.P. y la U.C. (una D por estado) para el sistema digital que cumple el algoritmo:

1: Declaración: $A [8], B [8], Cont [4], Bus [8]$

2: $A \leftarrow Bus$

3: $B \leftarrow Bus, Cont = 0$

4: while $Cont \neq 14$

5: if A es múltiplo de 4 then

6: $A \leftarrow A - B, Cont = (Cont + 2) \bmod 16$

7: else

8: $B \leftarrow B + A;$

9: endif

10: end while

11: $Bus \leftarrow B$

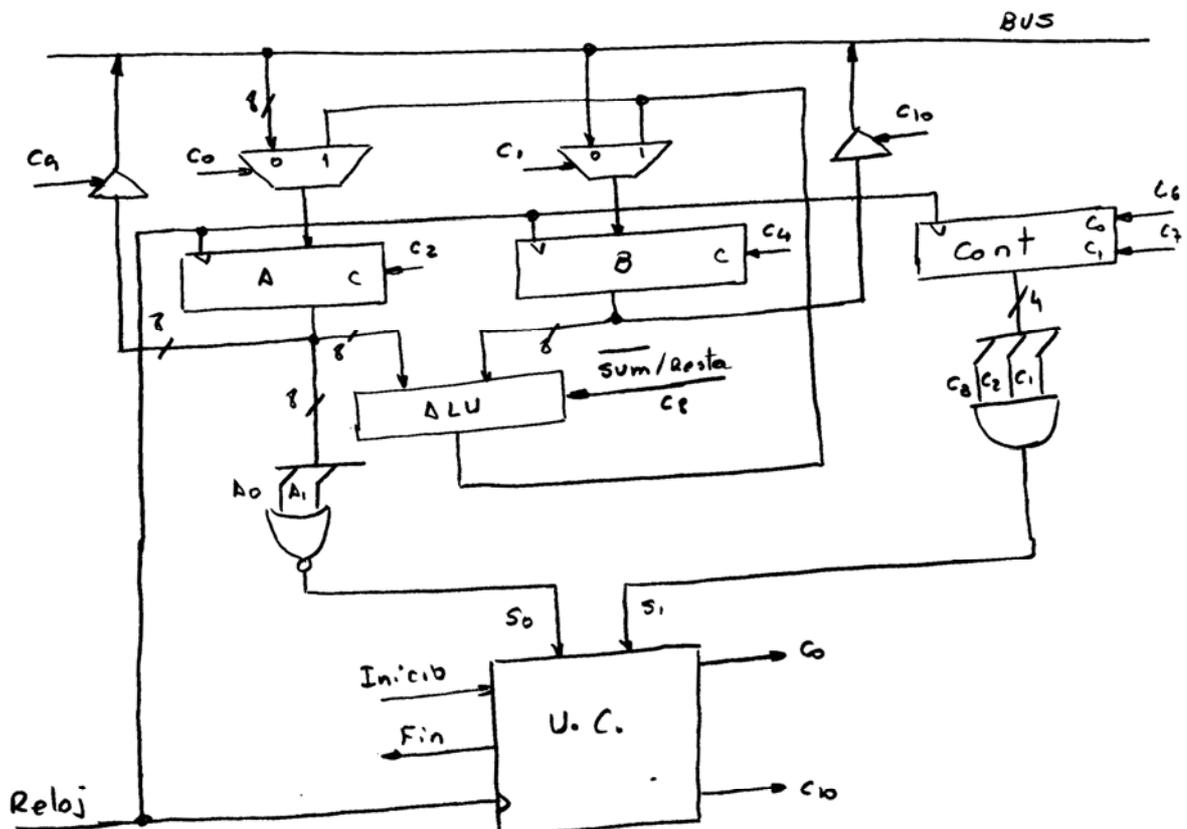
12: $Bus \leftarrow A$

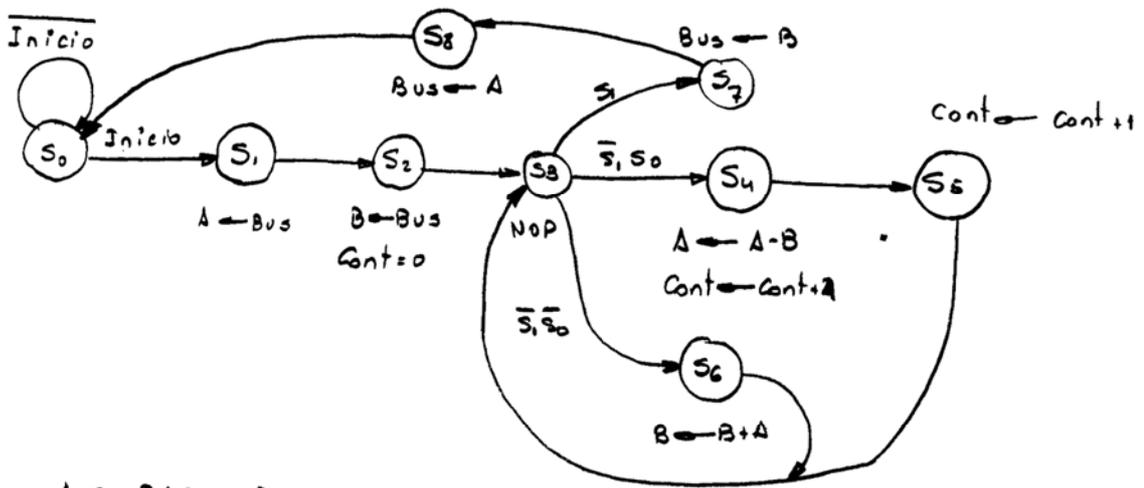
13: Parar

Múltiplo de 4 $\Rightarrow A_1, A_0 = 00$

$14 = 1110 \Rightarrow C_3 C_2 C_1 = 111$

A, B se pueden cargar desde ALU y BUS

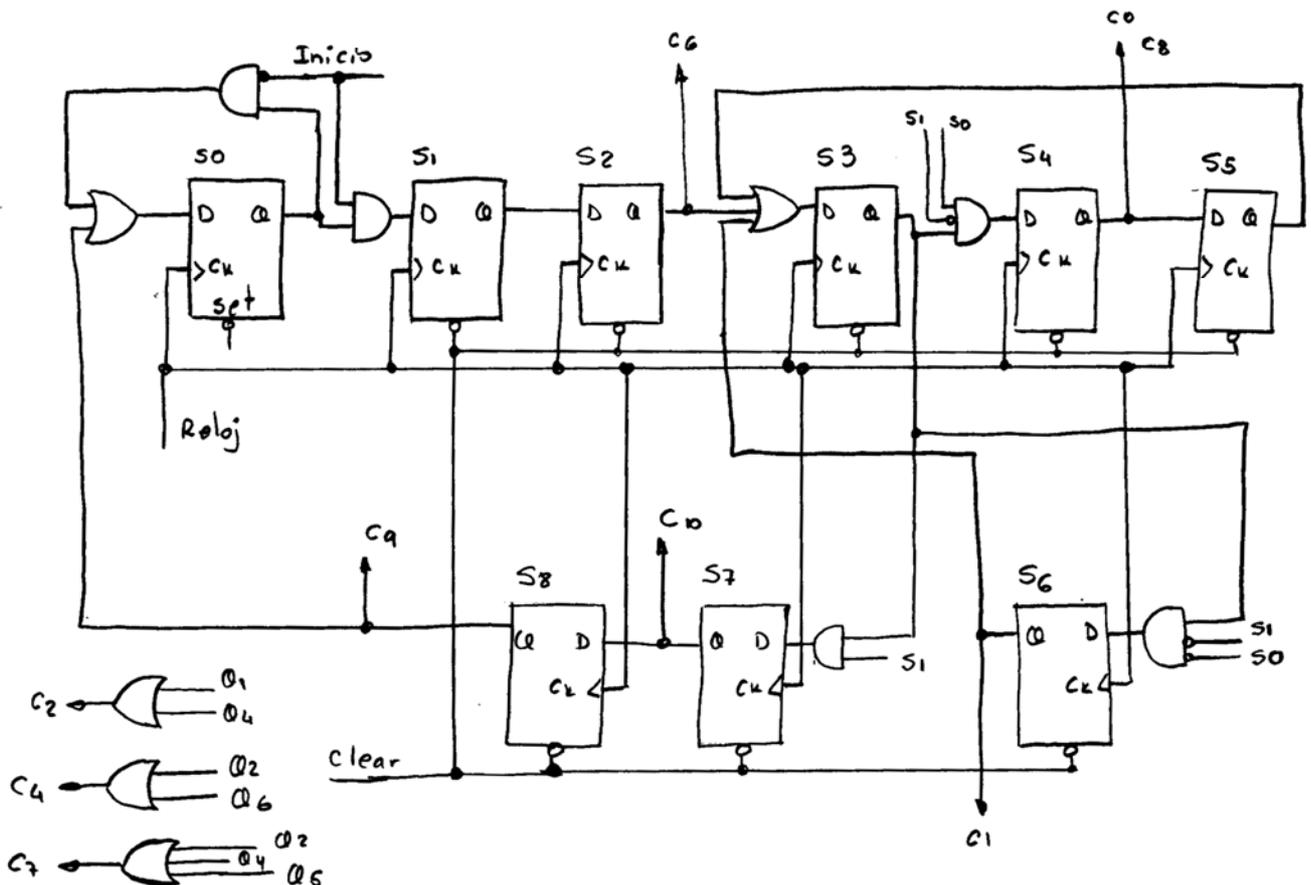




- $A \leftarrow \text{Bus} \Rightarrow c_0 = 0, c_2$
- $B \leftarrow \text{Bus} \Rightarrow c_1 = 0, c_4$
- $\text{Cont} = 0 \Rightarrow c_6, c_7$
- $A \leftarrow A - B \Rightarrow c_8, c_0, c_2$
- $\text{Cont} \leftarrow \text{Cont} + 1 \Rightarrow c_7 = 1, c_6 = 0$
- $B \leftarrow B + A \Rightarrow c_8 = 0, c_1, c_4$
- $\text{Bus} \leftarrow A \Rightarrow c_9$
- $\text{Bus} \leftarrow B \Rightarrow c_{10}$

Esto es así, porque así se indica en los elementos del enunciado (pg 6)

- No es necesario hacer la tabla de estados pq es una base de D por estado y la traducción es inmediata



Septiembre 2001

Problema similar a los del 2002

Preguntas de test

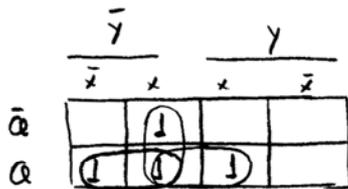
Septiembre 2002 - 1ª

Est. pres Q	Entradas xy			
	00	01	10	11
0	0	0	1	0
1	1	0	1	1

Tabla de estados

Entr.		Est. Actual	Est próximo
x	y	Q	Q _{entr}
0	0	0	0
0	1	0	0
1	0	0	1
1	1	0	0
0	0	1	1
0	1	1	0
1	0	1	1
1	1	1	1

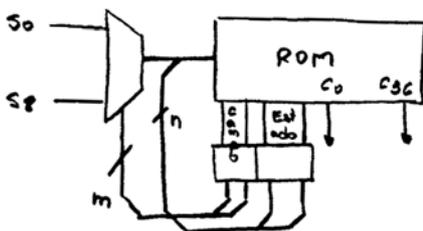
Con básculas D



$$D = x\bar{y} + 0\bar{y} + 0x = x\bar{y} + Q(x+y) \Rightarrow 10$$

Junio 2001 - 1ª S-4ª

- Unidad de control con 200 estados.
- 37 señales de control totalmente independientes
- 9 " " condición pero en cada estado solo una consultada
- Selección por campo



m = n° bits necesarios para codificar hasta 9 "señales de control"

$$2^4 = 16 \Rightarrow 4 \text{ pautillas selección}$$

$$\text{el mux} \Rightarrow 16 \text{ ent. datos}$$

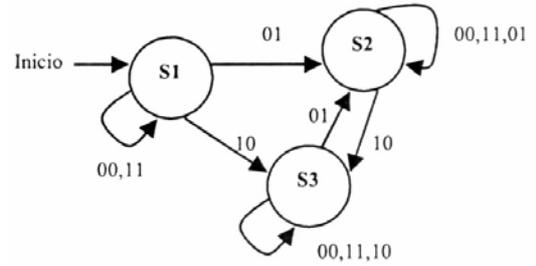
C

E.D.T.R. 6

Junio 2003

1.- Dada la siguiente implementación secuencial de un comparador de dos números de n bits sin signo X e Y , donde éste recibe los bits x_i e y_i de forma serie, comenzando por los bits menos significativos, los estados $S1$, $S2$ y $S3$ se corresponden con:

- A) $S1: x=y$, $S2: x<y$, $S3: x>y$.
- B) $S1: x<y$, $S2: x>y$, $S3: x=y$.
- C) $S1: x>y$, $S2: x<y$, $S3: x=y$.
- D) Ninguno de los anteriores.



Solución:

$S1 \Rightarrow$ No sale de $S1$ mientras 00, 11 ; esto es mientras $x = y$.

$S2 \Rightarrow$ A $S2$ se entra cuando $x = 0$ e $Y = 1 \Rightarrow x < y$.

Nos quedamos en $S2$ mientras $(x = y)$ o $(x < y)$.

Salimos a $S3$ cuando $(x > y)$.

Van entrando bits más significativos y nos quedamos en $S2$ hasta que un bit $(x > y)$.

Por lo tanto $S2 \Rightarrow (x < y)$.

$S3 \Rightarrow$ A $S3$ se entra cuando $(x>y)$ desde el inicio (bits iguales).

A $S3$ se entra cuando $(x > y)$ desde un $(x < y)$.

En $S3$ se mantiene mientras $(x = y)$ o $(x > y)$.

Por lo tanto $S3 \Rightarrow (x > y)$.

Ejemplo:

								Bit menos significativo	
X	\Rightarrow	1	0	0	0	0	0	1	$\Rightarrow x = y$
Y	\Rightarrow	1	0	0	0	0	0	1	
		↓	↓	↓	↓	↓	↓	↓	
Secuencia		7°	6°	5°	4°	3°	2°	1°	
		S1							

								Bit menos significativo	
X	\Rightarrow	1	0	0	0	1	0	1	$\Rightarrow x < y$
Y	\Rightarrow	1	0	0	1	0	0	0	
		↓	↓	↓	↓	↓	↓	↓	
Secuencia		7°	6°	5°	4°	3°	2°	1°	
		S2	S2	S2	S2	S3	S3	S3	

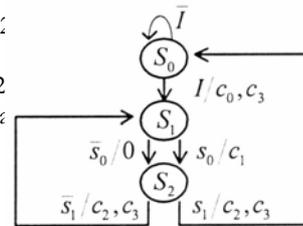
								Bit menos significativo	$\Rightarrow x > y$
X	\Rightarrow	1	0	0	0	1	0	1	$\Rightarrow x > y$
Y	\Rightarrow	0	1	1	1	1	1	1	
		↓	↓	↓	↓	↓	↓	↓	
Secuencia		7°	6°	5°	4°	3°	2°	1°	
		S3	S2	S2	S2	S2	S2	S1	

Solución la A

Junio 2003

3.- Considérese el diagrama de estados, correspondiente a una unidad de control, mostrado en la figura adjunta. Señale cual de las siguientes afirmaciones es correcta:

- A) Puede realizarse la síntesis de la unidad de control empleando un registro de 2 y una memoria ROM de 32 palabras, con 8 bits por palabra.
- B) Puede realizarse la síntesis de la unidad de control empleando un registro de 2 un multiplexor de 4 a 1, y una memoria ROM de 8 palabras, con 8 bits por palabra.
- C) Las dos anteriores son correctas.
- D) Todas las anteriores son falsas.

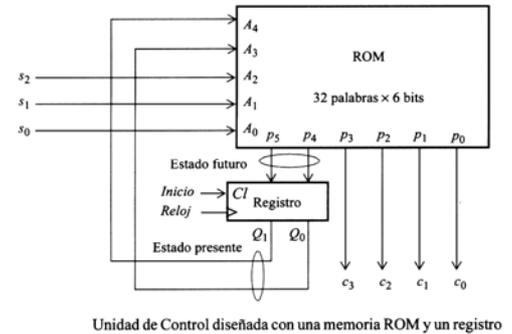


Solución:

- Utilizando una ROM con selección por estado:

Entradas $\rightarrow I, s_0, s_1$	Bus de direcciones ROM $\rightarrow 5$ bits
Estados $\rightarrow 4 \rightarrow Q_1, Q_0$	

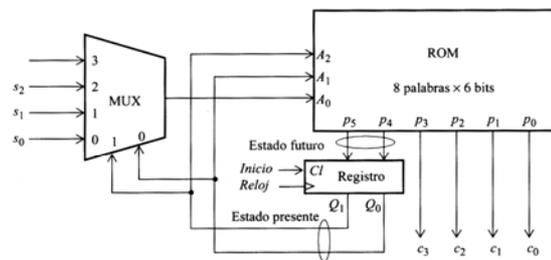
Salidas de control $\rightarrow c_3, c_2, c_1, c_0$	Bus de datos ROM $\rightarrow 6$ bits
Estados $\rightarrow 4 \rightarrow Q_{1t+1}, Q_{0t+1}$	



Unidad de Control diseñada con una memoria ROM y un registro

ROM $\Rightarrow 2^5 \times 6 \Rightarrow$ se puede implementar con una ROM de 32 palabras y 8 bits por palabra, aún incluso si se considerase I (inicio) como una señal de control al registro, ya que en este caso la ROM sería más pequeña todavía ($2^4 \times 6$). Esto implica que la opción A es correcta.

- Utilizando una ROM con registro y multiplexor:



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

Entradas $\rightarrow I, s_1, s_0 \rightarrow$ Multiplexor de 4 canales $\rightarrow 1$ bit (s_i) en el bus de direcciones de la ROM	Bus direcciones de la ROM $\rightarrow 3$ bits
Estados $\rightarrow 4 \rightarrow Q_1, Q_0 \rightarrow 2$ bits	

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

ROM $\Rightarrow 2^3 \times 6 \Rightarrow$ se puede implementar con un multiplexor de 4 a 1 y una ROM de 8 palabras y 8 bits por palabra, aún incluso si se considerase I (inicio) como una señal de control al registro, ya que en este caso el multiplexor necesario podría ser de 2 a 1. Esto implica que la opción B es correcta.

Solución la C) Las dos anteriores son correctas.