

Problemas resueltos de "Diseño de transferencia de registros"

Problema 5.1

Diseñar los circuitos que hacen:

a) Si X es impar : $A \leftarrow B + C$

" " " " : $A \leftarrow B - C$

- A, B, C y X son registros de 8 bits

- Inicialmente vacíos

- se cargan desde el Bus de entrada

- El reg. A vuelca su valor en el bus de salida

1.ª - Descripción del algoritmo

1: Declaración de registros $A[8], B[8], C[8], X[8];$

2: " " buses $B\text{-entrada}[8], B\text{-salida}[8];$

3: Inicio : $X \leftarrow \text{Bus-entrada}$

4 : $B \leftarrow \text{" "}$

5 : $C \leftarrow \text{" "}$

6 : Test if $x_0=0$ then goto par;

7 : Imper $A \leftarrow B + C ;$

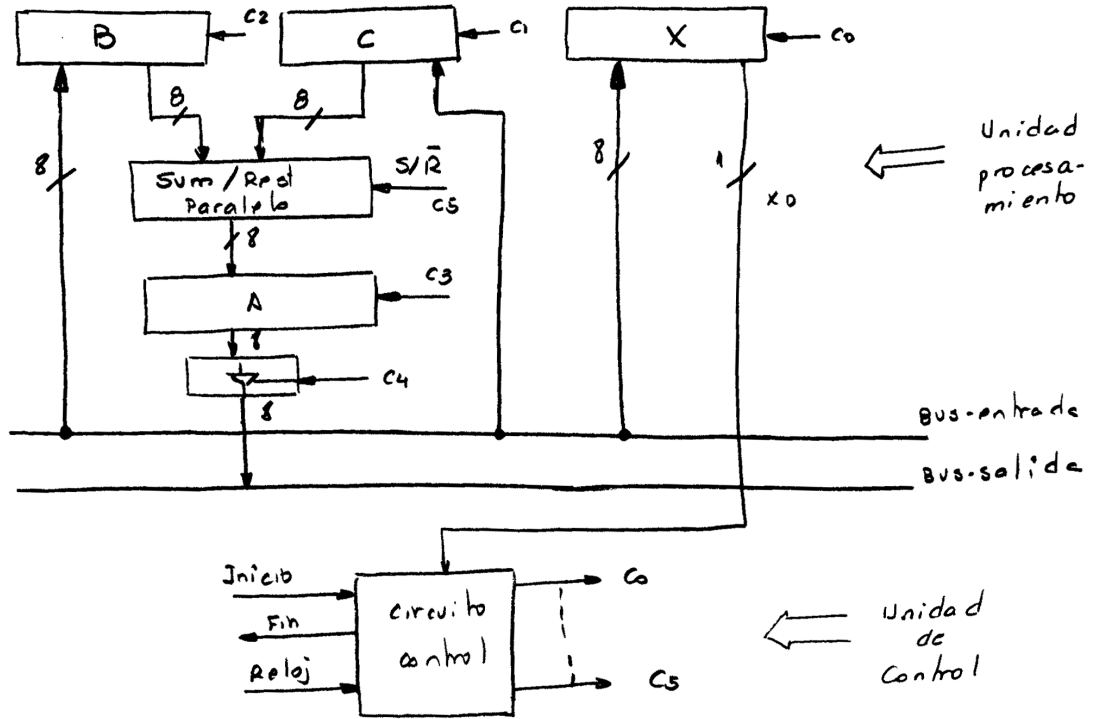
8 : goto Fin ;

9 : Par $A \leftarrow B - C ;$

10 : Fin Bus-salida $\leftarrow A$

11 : Parar

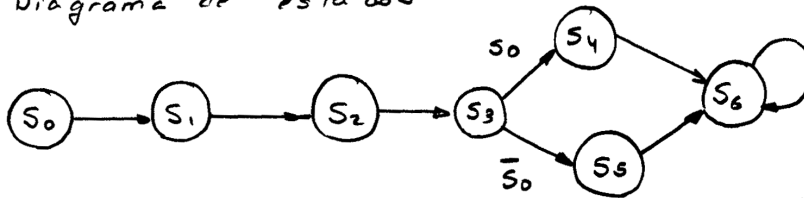
2.- Diagrama de bloques



3.- Señales control

- $c_0 \rightarrow$ carga de X desde Bus entrada
- $c_1 \rightarrow$ " " C " " "
- $c_2 \rightarrow$ " " B " " "
- $c_3 \rightarrow$ " " A " Sum/restador
- $c_4 \rightarrow$ Salida de A a Bus-salida
- $c_5 \rightarrow$ 0 \rightarrow Resta 1 \rightarrow Suma

4.- Diagrama de estados



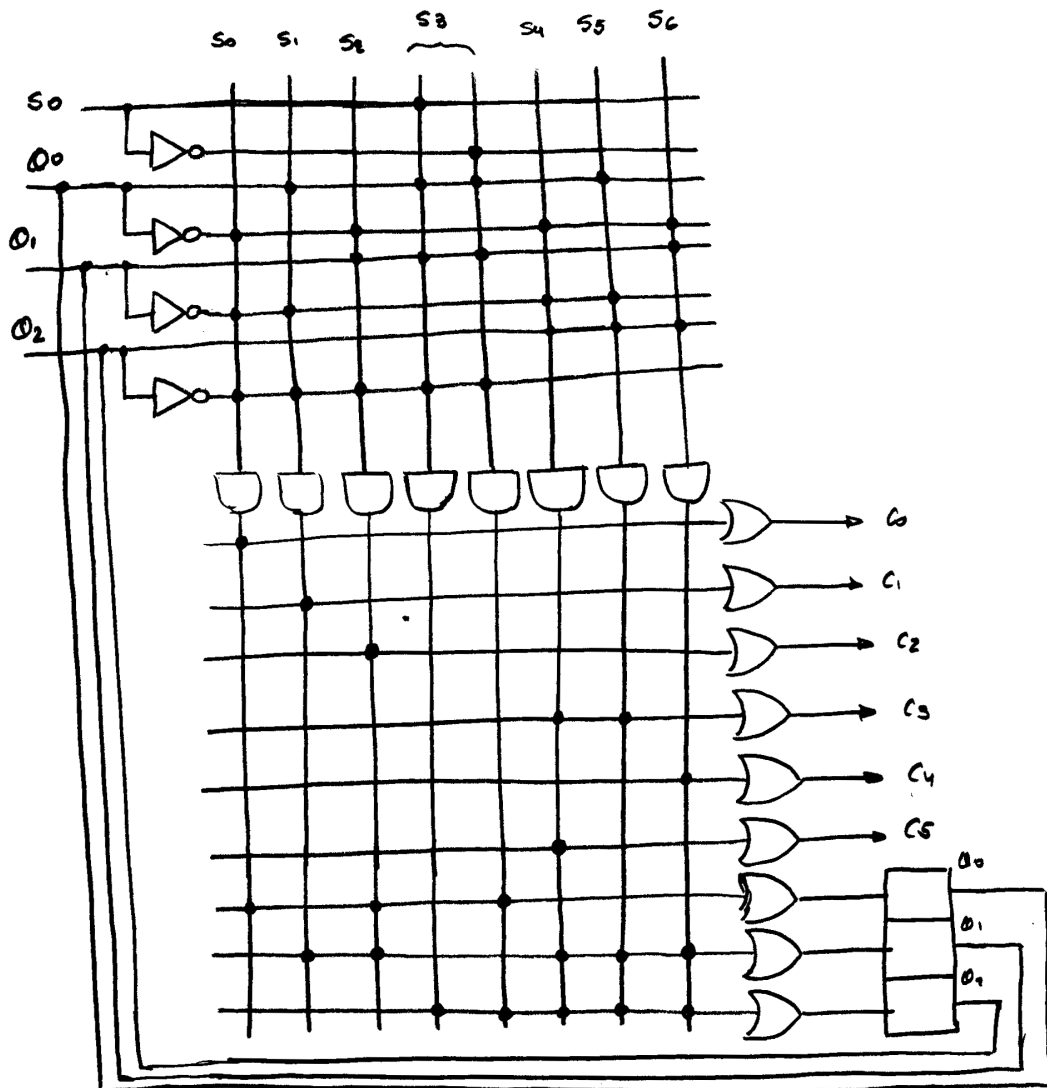
- $S_0 \rightarrow (X \leftarrow B-e) \Rightarrow c_0$
- $S_1 \rightarrow (C \leftarrow B-e) \Rightarrow c_1$
- $S_2 \rightarrow (B \leftarrow B-e) \Rightarrow c_2$
- $S_3 \rightarrow$ Nada; para testeo
- $S_4 \rightarrow (A \leftarrow B+C) \Rightarrow \begin{cases} c_3 = 1 \\ c_5 = 1 \end{cases}$
- $S_5 \rightarrow (A \leftarrow B-C) \Rightarrow \begin{cases} c_3 = 1 \\ c_5 = 0 \end{cases}$
- $S_6 \rightarrow$ Sacar A a Bus-salida $\Rightarrow c_4 = 1$

Entradas	Est. Actual			Est. Próximo			Salidas					
S_0	Q_2	Q_1	Q_0	$Q_{2,t+1}$	$Q_{1,t+1}$	$Q_{0,t+1}$	C_5	C_4	C_3	C_2	C_1	C_0
x	0	0	0	0	0	1	0	0	0	0	0	1
x	0	0	1	0	1	0	0	0	0	0	1	0
x	0	1	0	0	1	1	0	0	0	0	0	0
↓	0	1	1	1	0	0	1	0	1	0	0	0
0	0	1	1	1	0	1	0	0	1	0	0	0
x	1	0	0	1	1	0	0	1	0	0	0	0
x	1	0	1	1	1	0	0	1	0	0	0	0
x	1	1	0	1	1	0	0	0	0	0	0	0

Hecho con PLA →

Entradas: S_0, Q_0, Q_1, Q_2

Salidas: $C_0 \dots C_5$

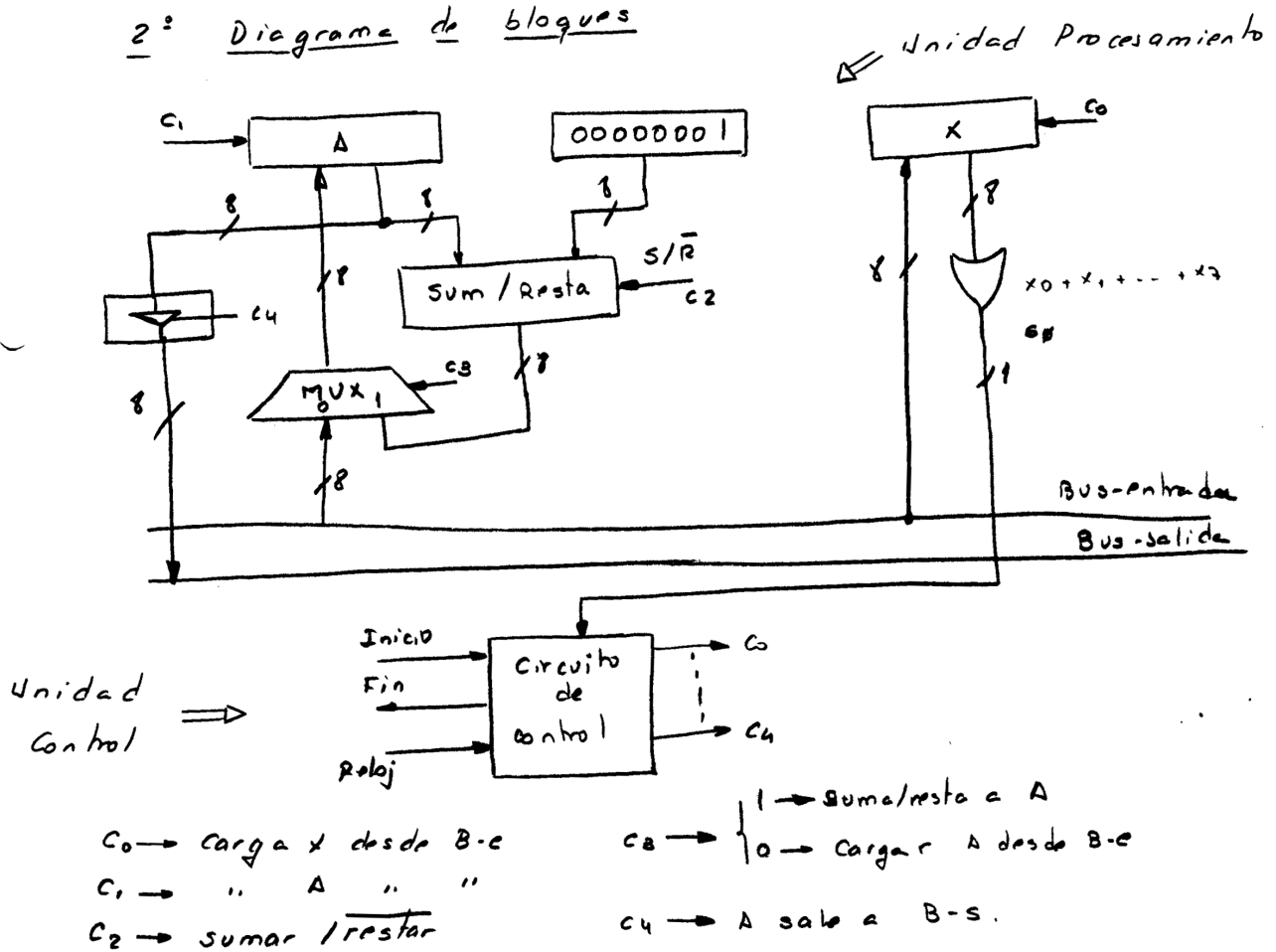


- b) Si $x \neq 0 \rightarrow A \leftarrow A - 1$
 " $x = 0 \rightarrow A \leftarrow A + 1$

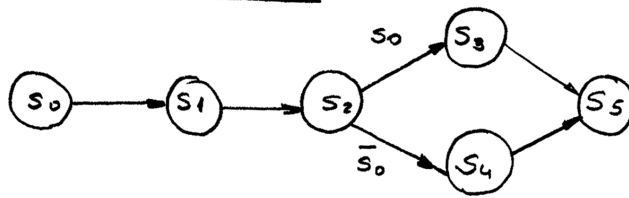
1º Descripción algoritmo

- 1: Declarar registros $A [8], x [8];$
 2: " buces $B-en [8], B-sa [8]$
 3: Inicio : $x \leftarrow B-en$
 4: $A \leftarrow B-en$
 5: Test: if $x=0$ then go to sumar
 6: Restar: $A \leftarrow A - 1$
 7: go to Fin
 8: Sumar: $A \leftarrow A + 1$
 9: Fin: Parar

2º Diagrama de bloques



4.- Diagrama de estados



$S_0 \rightarrow (X \leftarrow B-e) \Rightarrow C_0$

$S_1 \rightarrow (A \leftarrow B-e) \Rightarrow C_1, C_3 = 0$

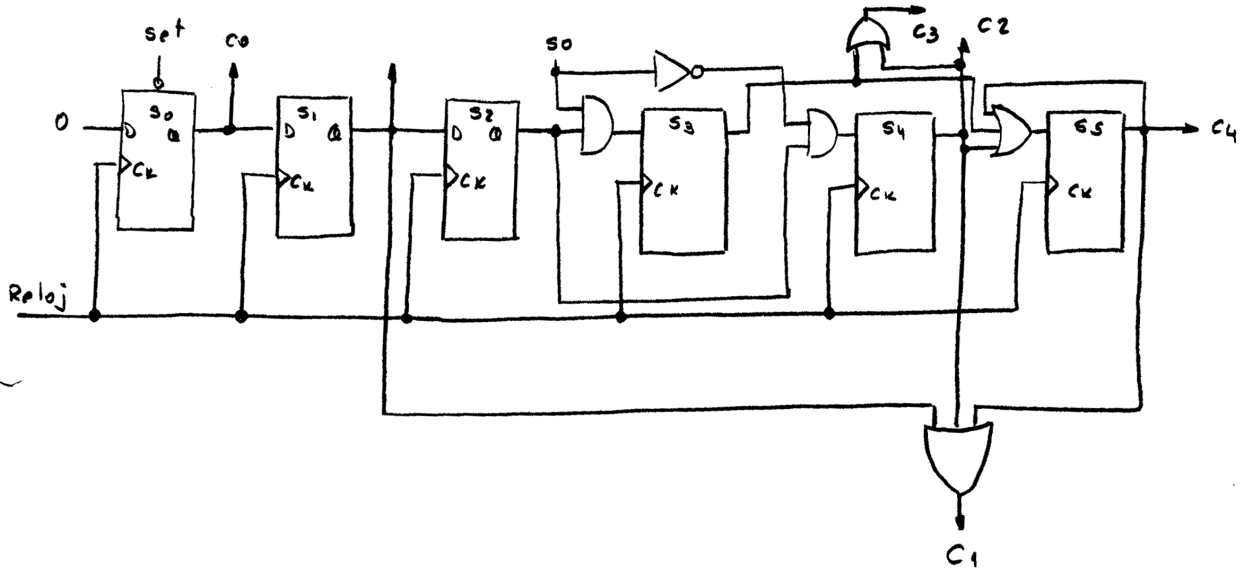
$S_2 \rightarrow \text{Nada}$

$S_3 \rightarrow (A \leftarrow (A-1)) \Rightarrow C_2 = \emptyset, C_3 = 1, C_1 = 1$

$S_4 \rightarrow (A \leftarrow A+1) \Rightarrow C_2 = 1, C_3 = 1, C_1 = 1$

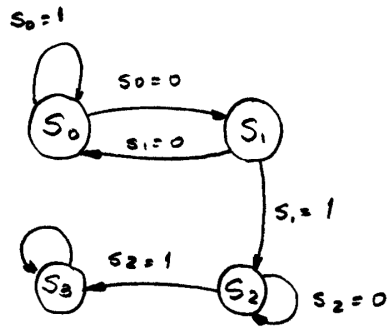
$S_5 \rightarrow (B-s \leftarrow \Delta) \Rightarrow C_4 = 1$

Unidad de control de báscula por estado



Problema 5.2

- Diseñar la unidad de control del sistema que cumpla el diagrama de estados indicado.

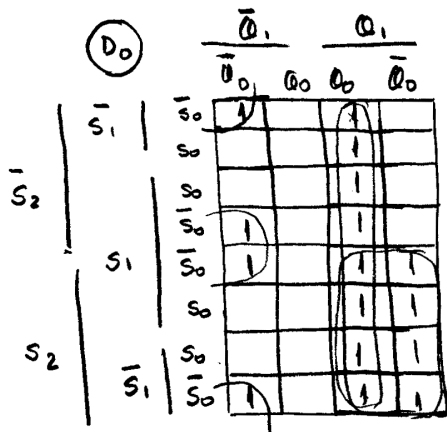


- Suponer que cada estado S_i activa únicamente la señal de control c_i .

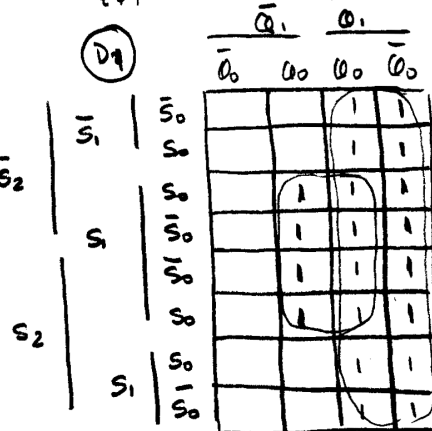
- Diseño con basculas D
- J: Tabla de estados

Entradas	Est. Actual	Est. Próximo	Salidas
s_2, s_1, s_0	Q_1, Q_0	$Q_{1,t+1}, Q_{0,t+1}$	$\forall S_i \rightarrow c_i$ En este caso las salidas son c_0, c_1, c_2 y c_3 y para codificar las se colocará un codificador 2 a 4
x x 1	0 0	0 0	
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	

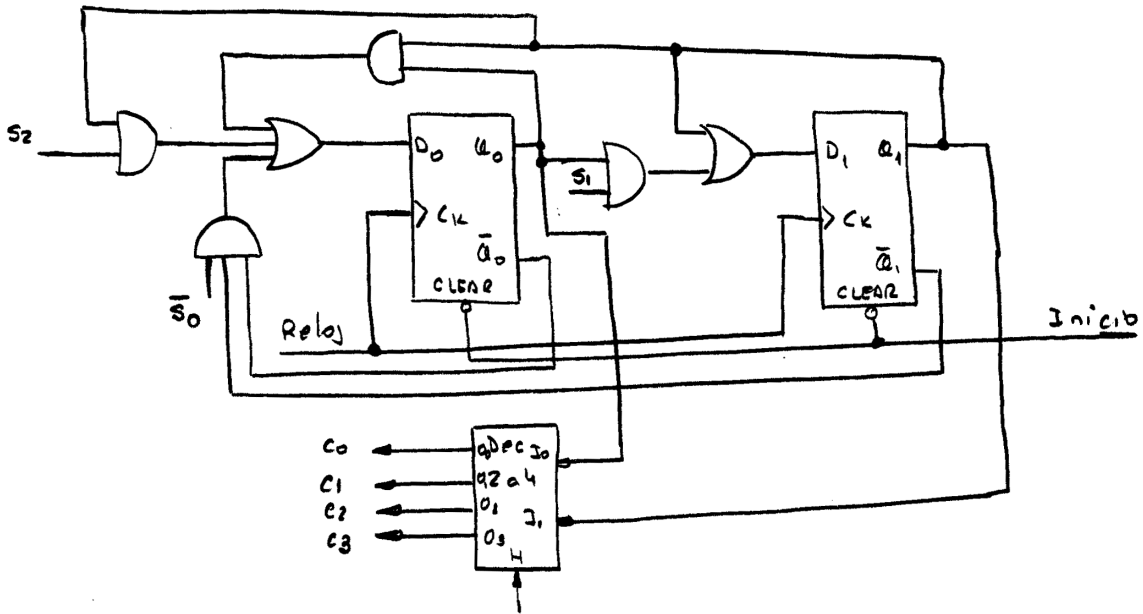
Con basculas D $\Rightarrow D_i = Q_{i,t+1} \Rightarrow Q_{1,t+1} = D_1$ y $Q_{0,t+1} = D_0$



$$D_0 = Q_1 Q_0 + s_2 Q_1 + \bar{s}_0 \bar{Q}_1 \bar{Q}_0$$



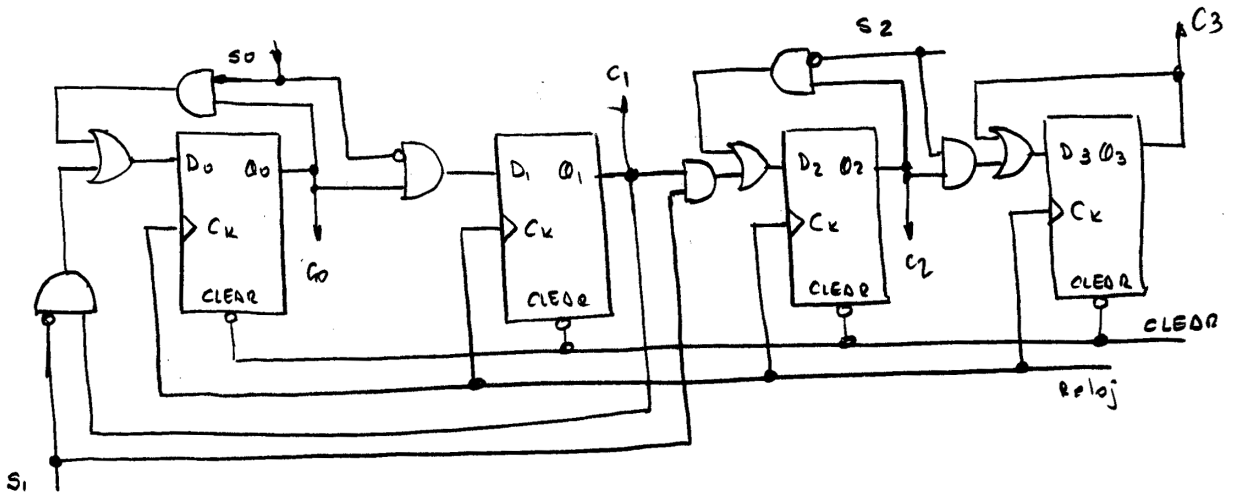
$$D_1 = Q_1 + s_1 Q_0$$



Problema 5.4

El problema 5.2 pero con una bscula D por estado.

4 estados \Rightarrow 4 bsculas
 Las salidas c_i directamente de cada estado



Problema 5.5

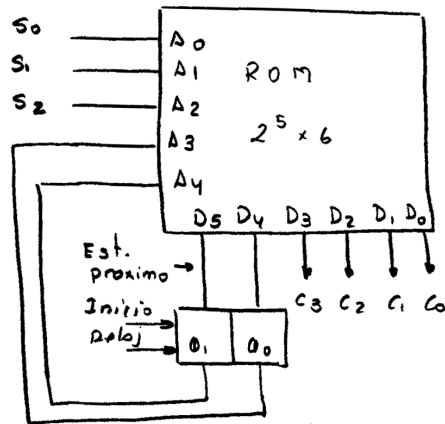
Problema 5.2 utilizando ROM y registro

a) Memoria ROM y registro

Entradas $\Rightarrow s_2, s_1, s_0$
 Estados $\Rightarrow 4 \Rightarrow 0, 0_0$ } Bus direcciones ROM $\Rightarrow 5$ bits

Salidas $\Rightarrow c_3, c_2, c_1, c_0$
 Estados $\Rightarrow 4 \Rightarrow 0, 0_0$ } Bus datos $\Rightarrow 6$ bits

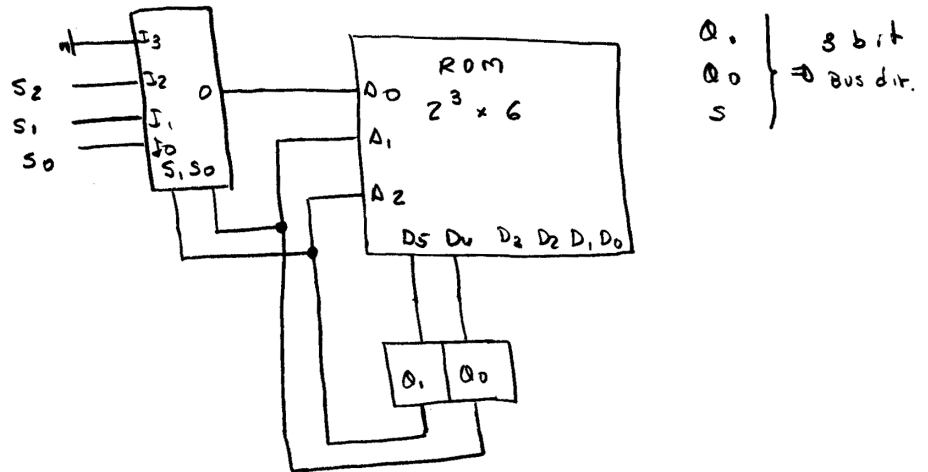
Esquema



					Estado presente																
s_2	s_1	s_0	Estados		c_3	c_2	c_1	c_0	s_2	s_1	s_0										
A_2	A_1	A_0	A_3	A_4	D_5	D_4	D_3	D_2	D_1	D_0	A_4	A_3	A_2	A_1	A_0	D_5	D_4	D_3	D_2	D_1	D_0
0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
0	0	1	1	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
0	1	1	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
0	1	1	1	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
1	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
1	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
1	0	1	1	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
1	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
1	1	0	1	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
1	1	1	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0

b) Memoria ROM y selección por estado.

El paso de estado depende siempre de una única variable \Rightarrow
 en lugar de conectar las 3 variables se coloca un multiplexor
 y en función del estado se multiplexa la variable que interviene



Contenido			ROM					
S_2	S_1	S_0	D_5	D_4	D_3	D_2	D_1	D_0
0	0	0	0	1	0	0	0	1
0	0	1	0	0	0	0	0	1
0	1	0	0	0	0	0	1	0
0	1	1	1	0	0	0	1	0
1	0	0	1	0	0	1	0	0
1	0	1	1	1	0	1	0	0
1	1	0	1	1	1	0	0	0
1	1	1	1	1	1	0	0	0

Problema 5.7

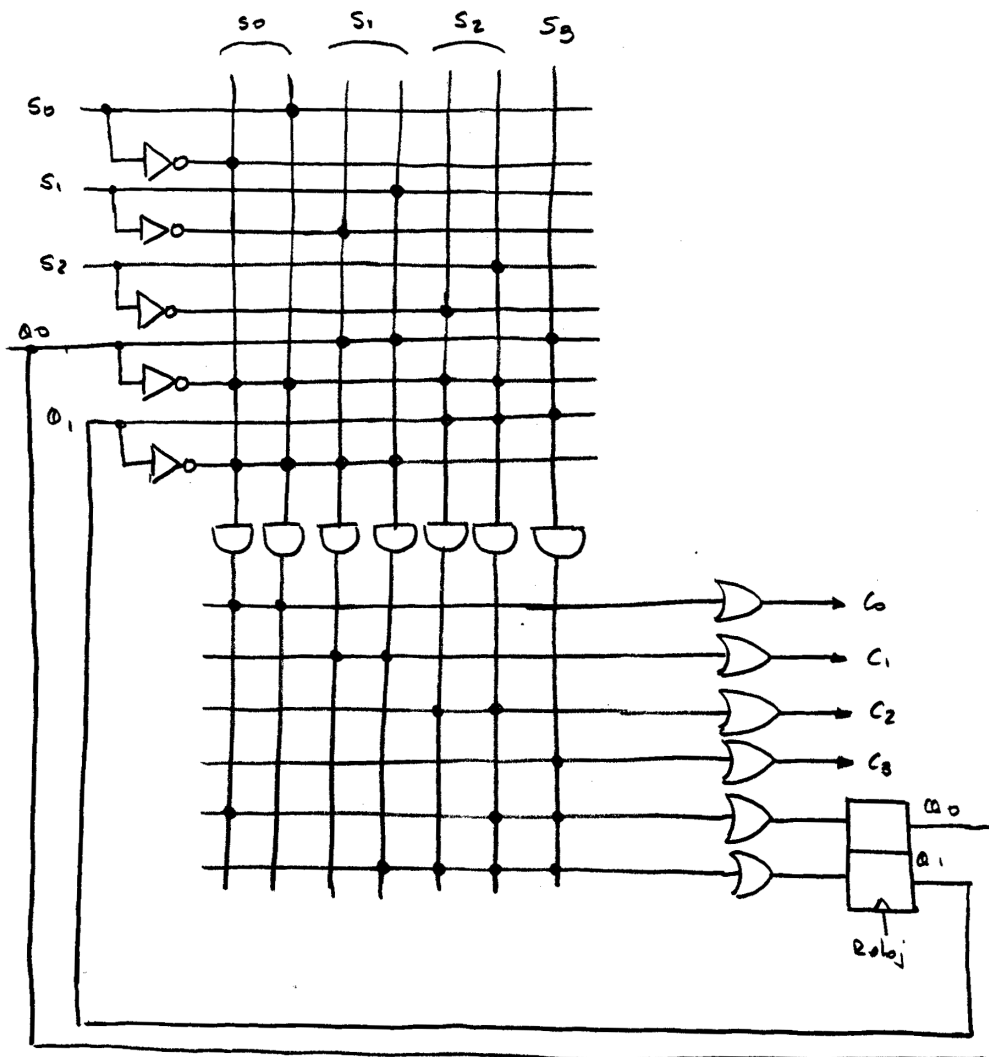
Problema 5.2 con PLA y un registro.

Entradas $\left. \begin{array}{l} s_2, s_1, s_0 \\ 0_1, 0_0 \end{array} \right\} \Rightarrow 5 \text{ entradas}$

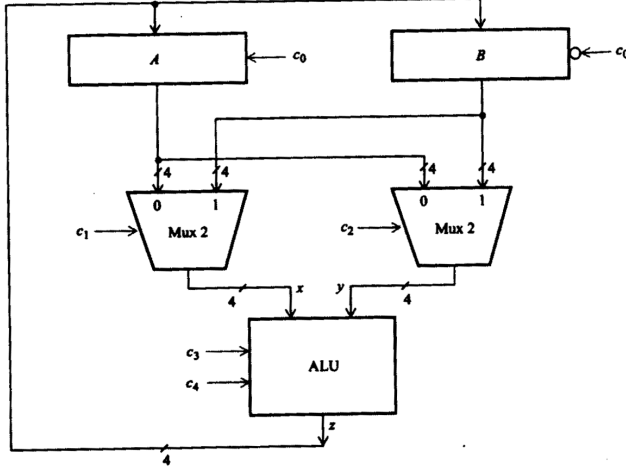
Células AND $\Rightarrow 4 \text{ estados}$

$\left. \begin{array}{l} s_0 \left\{ \begin{array}{l} s_0=0 \\ s_1=1 \end{array} \right. \textcircled{1} \\ \phantom{\left\{ \right.} \textcircled{2} \\ s_1 \left\{ \begin{array}{l} s_1=0 \\ s_2=1 \end{array} \right. \textcircled{3} \\ \phantom{\left\{ \right.} \textcircled{4} \\ s_2 \left\{ \begin{array}{l} s_2=0 \\ s_3=1 \end{array} \right. \textcircled{5} \\ \phantom{\left\{ \right.} \textcircled{6} \\ s_3 \phantom{\left\{ \right.} \textcircled{7} \end{array} \right\} \Rightarrow 7 \text{ células AND}$

Células OR $\Rightarrow \left. \begin{array}{l} 0_1 \\ 0_0 \\ c_3, c_2, c_1, c_0 \end{array} \right\} \Rightarrow 6 \text{ células OR}$



Problema 5.8



Señal de Control	Operación controlada
$c_0 = 0$	$B \leftarrow z$
$c_0 = 1$	$A \leftarrow z$
$c_1 c_2 = 00$	$x = A \quad y = A$
$c_1 c_2 = 01$	$x = A \quad y = B$
$c_1 c_2 = 10$	$x = B \quad y = A$
$c_1 c_2 = 11$	$x = B \quad y = B$
$c_3 c_4 = 00$	$z = x + y$
$c_3 c_4 = 01$	$z = x - y$
$c_3 c_4 = 10$	$z = x \text{ AND } y$
$c_3 c_4 = 11$	$z = x \oplus y$

Señales de control de la Unidad de Procesamiento

a) ¿Cómo se puede poner $B = 0$?

1.- $B \leftarrow B - B$

\Downarrow

$$\left\{ \begin{array}{l} x = B \Rightarrow c_1 = 1 \quad y = B \Rightarrow c_2 = 1 \\ z = x - y \Rightarrow c_3 c_4 = 01 \\ B \leftarrow z \Rightarrow c_0 = 0 \end{array} \right.$$

2.- $B \leftarrow B \oplus B$

\Downarrow

$$\left\{ \begin{array}{l} x = B \Rightarrow c_1 = 1 \quad y = B \Rightarrow c_2 = 1 \\ z = x \oplus y \Rightarrow c_3 c_4 = 11 \\ B \leftarrow z \Rightarrow c_0 = 0 \end{array} \right.$$

b) secuencia de control que intercambia A y B

	c_4	c_3	c_2	c_1	c_0
$A \leftarrow A - B \rightarrow$	1	0	1	0	1
$A \leftarrow A - B \Rightarrow A = A - 2B$	1	0	1	0	1
$B \leftarrow A + B \Rightarrow B = A - 2B + B = A - B$	0	0	1	0	0
$A \leftarrow B - A \Rightarrow (A - B) - (A - 2B) = B$	1	0	0	1	1
$B \leftarrow A + B \Rightarrow B + (A - B) = A$	0	0	1	0	0

Problema 5.9

según el siguiente algoritmo:

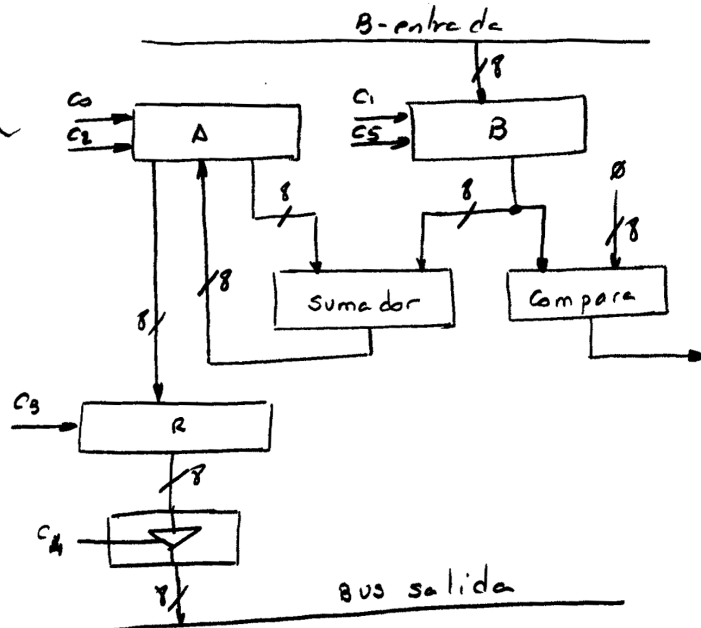
- 1: Declaración de registros: $A[8], B[8], R[8]$
 2: " " buses: $Bus-ent[8], Bus-sal[8]$
 3: Inicio: $A \leftarrow B-e$
 4: $B \leftarrow B-e$
 5: Bucle: $A \leftarrow A+B$
 6: $B \leftarrow B-1$
 7: if $B \neq 0$ then go to Bucle
 8: $R \leftarrow A$
 9: $B-5 \leftarrow A$
 10: Parar

a) ¿Qué operación se ejecuta si inicialmente se carga A con $\underbrace{00000000}_0$

y B con 00001111? $\rightarrow 15$

$\hookrightarrow \underline{0+15} + \underline{15+14} + \underline{15+14+13} + \underline{15+14+13+12} + \dots + \underline{15+14+\dots+1}$

b) Diseñar la U. Procesamiento para realizar el algoritmo



- $C_0 \rightarrow$ Poner a 0 A
- $C_1 \rightarrow$ Decrem. B
- $C_2 \rightarrow$ Cargar A con sumador
- $C_3 \rightarrow$ " R con A
- $C_4 \rightarrow$ sacar $R = B-5$
- $C_5 \rightarrow$ Cargar B con $B-e$

Problema 5.10

1: Declaración de registros $A[8], B[8], R[8], J[1]$
2: " " buses $B-e[8]; B-s[8]$
3: Inicio: if $I=0$ then go to Inicio;
4: $A \leftarrow B-e$
5: $B \leftarrow B-e$
6: $B \leftarrow B+A$
7: if $B[0]=0$ then go to Par;
8: $B \leftarrow \bar{B}$
9: $B \leftarrow B+1$
10: go to Fin
11: Par: $A \leftarrow A+1$
12: Fin: $R \leftarrow A+B, B-s \leftarrow A+B$
13: go to Inicio

$I =$ Señal de Inicio $B[0] =$ bit menos signi de B

a) Si inicialmente $A = 01101010$ y $B = 11011010$ determinar el valor de A, B y R tras la ejecución

$$6: \quad 01101010 + 11011010 = 01000100$$

$$7: \quad A = 01101010 \quad B = 01000100 \Rightarrow B[0]=0 \Rightarrow \text{Par}$$

$$11: \quad A \leftarrow A+1 \Rightarrow A = 01101011$$

$$12: \quad \begin{array}{r} 01101011 = A \\ 01000100 = B \\ \hline 10101111 = R \end{array}$$

b) Unidad de procesamiento:

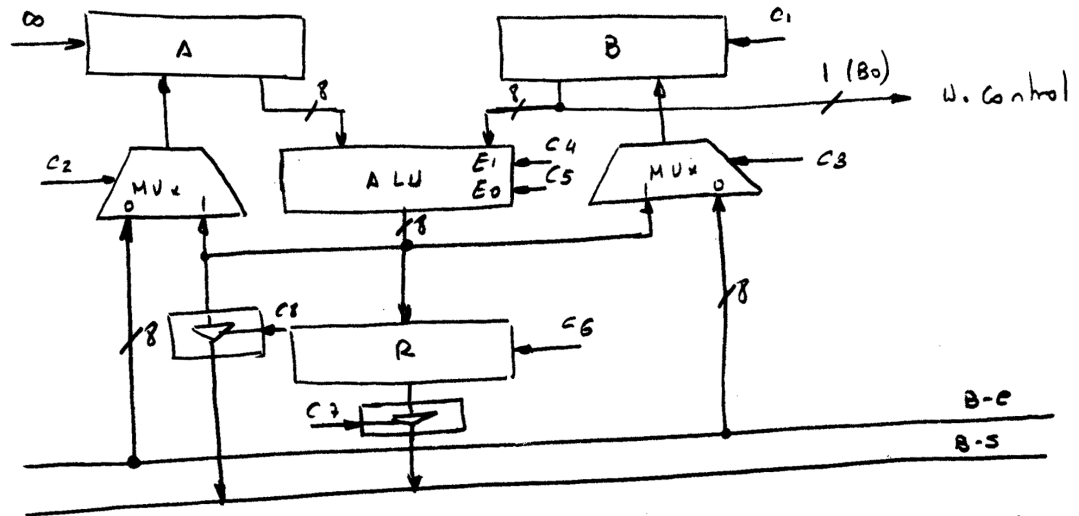
A se carga desde } Entrada
ALU — sale a ALU

B " " " } Entrada
ALU — sale a ALU

ALU \rightarrow operaciones } $\begin{array}{l} X+1 \\ Y+1 \\ X+Y \\ Y \end{array}$ — sale a } $\begin{array}{l} A \\ B \\ R \end{array}$

R \rightarrow se carga desde ALU — sale a R

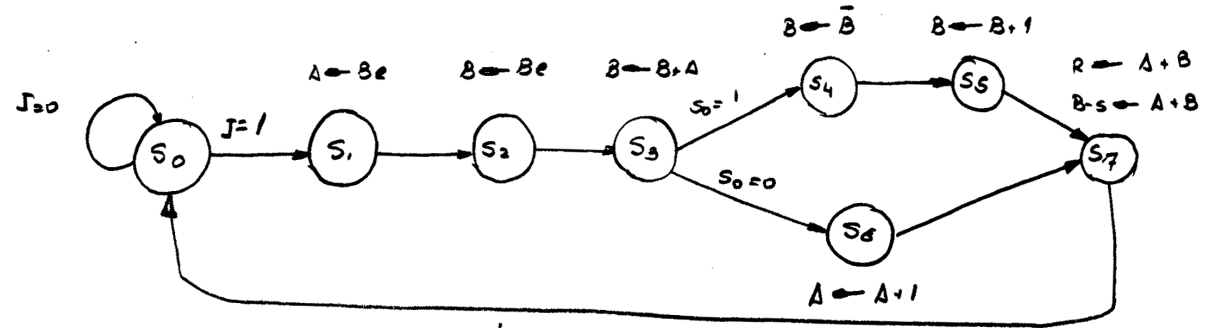
Oper. ALU	E ₁	E ₀
x+1	0	0
y+1	0	1
x+y	1	0
\bar{y}	1	1
	c ₄	c ₅



$c_0 \rightarrow$ carga A
 $c_1 \rightarrow$ " B
 $c_2 \rightarrow$ entrada a A desde B-e/ALU
 $c_3 \rightarrow$ " " B " " "

$c_4, c_5 \rightarrow$ Selección operación
 $c_6 \rightarrow$ carga de R
 $c_7 \rightarrow$ salida de R a B-s
 $c_8 \rightarrow$ " " ALU " "
 pq para 12 la salida es directa sin pasar por R.

c) Unidad control



Selección por campo

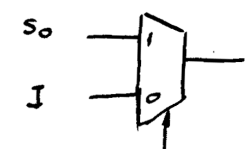
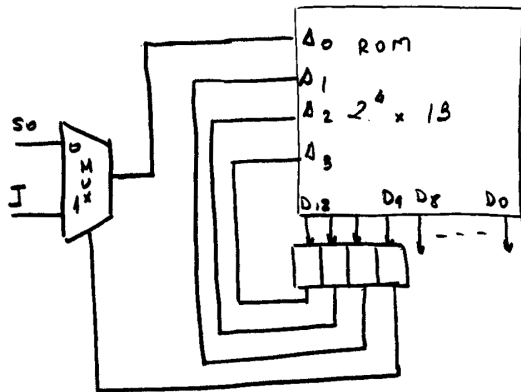


Tabla de estados

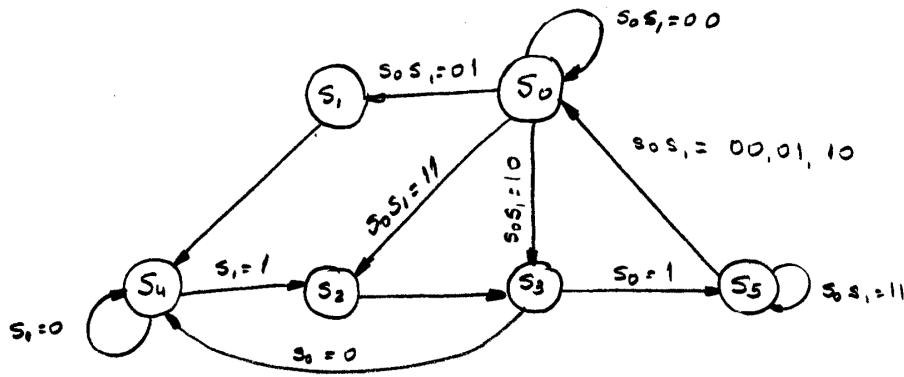
Entradas	Estado Actual			Estado Prximo			Salidas								
s ₀ /I	Q ₂	Q ₁	Q ₀	Q ₂ '	Q ₁ '	Q ₀ '	C ₈	C ₇	C ₆	C ₅	C ₄	C ₃	C ₂	C ₁	C ₀
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
x	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1
x	0	1	0	0	1	1	0	0	0	0	0	0	0	1	0
0	0	1	1	1	1	0	0	0	0	1	1	0	1	0	0
1	0	1	1	1	0	0	0	0	0	1	1	0	1	0	0
x	1	0	0	1	0	1	0	0	0	1	1	1	0	1	0
x	1	0	1	1	1	1	0	0	0	1	0	1	0	1	0
x	1	1	0	1	1	1	0	0	0	0	0	0	1	0	1
x	1	1	1	0	0	0	1	0	1	0	1	0	0	0	0

Ancho → 1
 Largo → 4



Q ₂ Q ₁ Q ₀ s ₀ /I	Q ₂ Q ₁ Q ₀ Mux	C ₈ C ₇ C ₆ C ₅ C ₄ C ₃ C ₂ C ₁ C ₀
A ₃ A ₂ A ₁ A ₀	D ₁₂ D ₁₁ D ₁₀ D ₉	D ₈
0 0 0 0	0 0 0 1	0 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 1 0	0 1 0 x	0 0 0 0 0 0 0 0 0 1
0 0 1 1	0 1 0 x	0 0 0 0 0 0 0 0 0 1
0 1 0 0	0 1 1 x	0 0 0 0 0 0 0 0 0 1 0
0 1 0 1	0 1 1 x	0 0 0 0 0 0 0 0 0 1 0
0 1 1 0	1 1 0 0	0 0 0 0 0 1 1 0 1 0
0 1 1 1	1 0 0 0	0 0 0 0 0 1 1 0 1 0
1 0 0 0	1 0 1 x	0 0 0 1 1 1 0 1 0
1 0 0 1	1 0 1 x	0 0 0 1 1 1 0 1 0
1 0 1 0	1 1 1 x	0 0 0 1 0 1 0 1 0
1 0 1 1	1 1 1 x	0 0 0 1 0 1 0 1 0
1 1 0 0	1 1 1 x	0 0 0 0 0 0 1 0 1
1 1 0 1	1 1 1 x	0 0 0 0 0 0 1 0 1
1 1 1 0	0 0 0 x	1 0 1 0 1 0 0 0 0
1 1 1 1	0 0 0 x	1 0 1 0 1 0 0 0 0

Problema 5.11



Diseñar la Unidad de Control si cada estado activa una c_i

Tabla de estados

Entradas	Est. Actual	Est. próximo	Salidas
s_0s_1	$Q_2 Q_1 Q_0$	$Q_2^{t+1} Q_1^{t+1} Q_0^{t+1}$	$c_5 c_4 c_3 c_2 c_1 c_0$
0 0	0 0 0	0 0 0	0 0 0 0 0 1
0 1	0 0 0	0 0 1	0 0 0 0 0 1
1 0	0 0 0	0 1 1	0 0 0 0 0 1
1 1	0 0 0	0 1 0	0 0 0 0 0 1
x x	0 0 1	1 0 0	0 0 0 0 1 0
x x	0 1 0	0 1 1	0 0 0 1 0 0
1 x	0 1 1	1 0 1	0 0 1 0 0 0
0 x	0 1 1	1 0 0	0 0 1 0 0 0
x 0	1 0 0	1 0 0	0 1 0 0 0 0
x 1	1 0 0	0 1 0	0 1 0 0 0 0
1 1	1 0 1	1 0 1	1 0 0 0 0 0
0 x	1 0 1	0 0 0	1 0 0 0 0 0
1 0	1 0 1	0 0 0	1 0 0 0 0 0

a) Basculas D

Q_2

	s_0	s_0	s_0	s_0
Q_1	Q_0	1	1	1
Q_1	1	1	1	1
Q_1	x	x	x	x
Q_1	x	x	x	x
Q_1	1	1	1	1

$D_2 = \bar{Q}_2 Q_0 + Q_2 \bar{Q}_0 \bar{s}_1 + Q_0 s_1 s_0$

Q_1

	s_0	s_0	s_0	s_0
Q_0	1	1	1	1
Q_0	1	1	1	1
Q_0	x	x	x	x
Q_0	x	x	x	x
Q_0	1	1	1	1

$D_1 = Q_1 \bar{Q}_0 + Q_2 \bar{Q}_0 s_1 + \bar{Q}_2 \bar{Q}_0 s_0$

Q_0

	s_0	s_0	s_0	s_0
Q_1	1	1	1	1
Q_1	1	1	1	1
Q_1	x	x	x	x
Q_1	x	x	x	x
Q_1	1	1	1	1

$D_0 = Q_1 \bar{Q}_0 + Q_1 s_0 + \bar{Q}_2 \bar{Q}_0 s_1 s_0 + \bar{Q}_1 \bar{Q}_0 s_1 s_0 + Q_2 \bar{Q}_0 s_1 s_0$

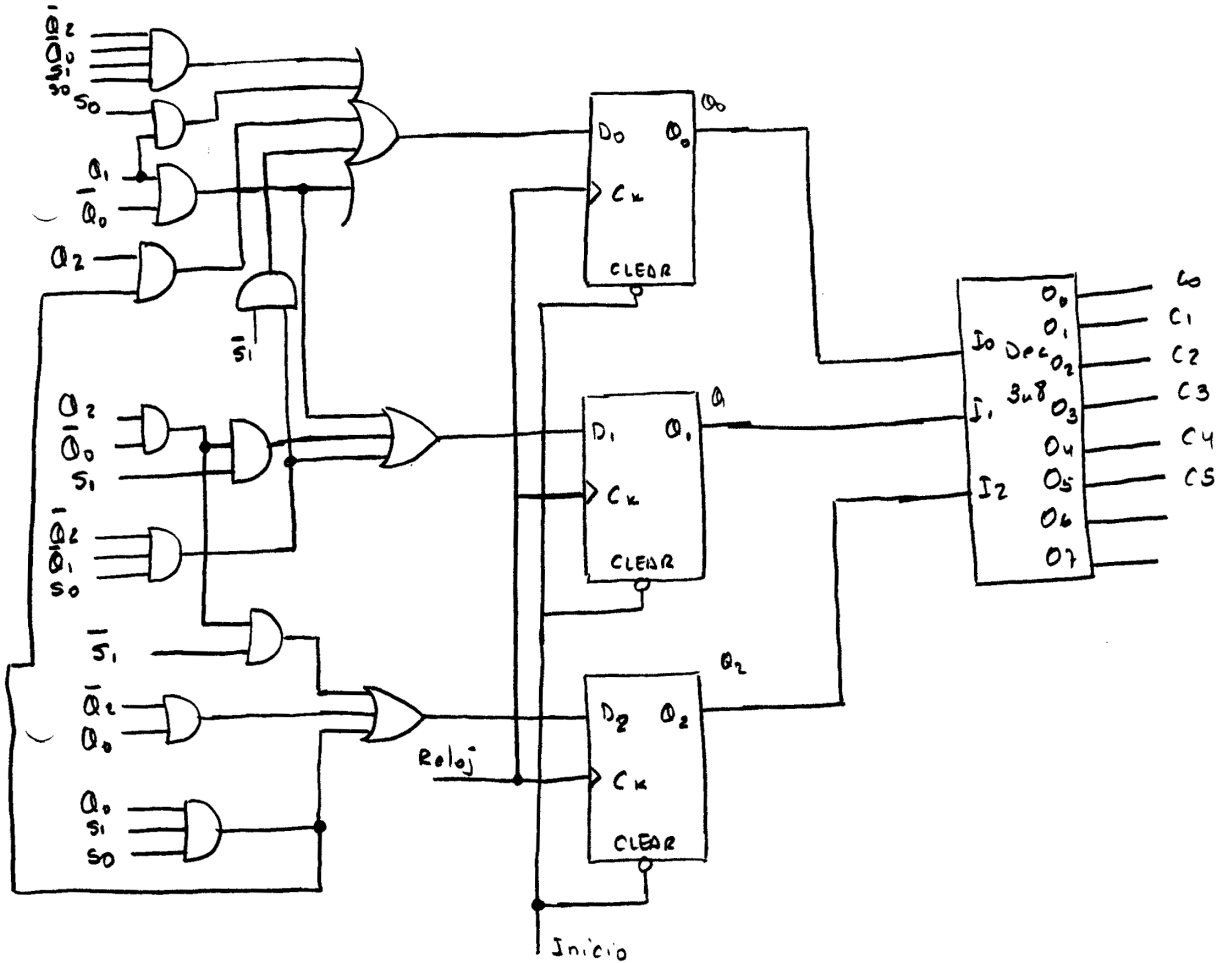
P.R. D.T.R. 16

$$D_2 = \bar{Q}_2 Q_0 + \underline{Q_2 \bar{Q}_0 \bar{S}_1} + \underline{Q_0 S_1 S_0}$$

$$D_1 = \underline{Q_1 \bar{Q}_0} + \underline{Q_2 \bar{Q}_0 S_1} + \underline{Q_2 \bar{Q}_1 S_0}$$

$$D_0 = \underline{Q_1 \bar{Q}_0} + Q_1 S_0 + \bar{Q}_2 \bar{Q}_0 S_1 \bar{S}_0 + \underline{Q_2 \bar{Q}_0 \bar{S}_1 S_0} + \underline{Q_2 Q_0 S_1 S_0}$$

- Las salidas c_i se obtienen a partir de un decodificador 3 a 8 al que entran las salidas de las básculas Q_2, Q_1, Q_0 pq hay una única salida de control y además diferente para cada estado



c) Con mem. ROM y un registro

ROM \Rightarrow $\left\{ \begin{array}{l} \text{Entradas} \\ \text{Salidas} \end{array} \right. \left\{ \begin{array}{l} \text{Estados } 0 \div 5 \Rightarrow 2^3 \Rightarrow 3 \text{ bit} \\ s_1, s_0 \Rightarrow 2 \text{ bit} \end{array} \right. \left\{ \begin{array}{l} 2^5 \text{ dir.} \\ 5 \text{ bit bus} \\ \text{dir.} \end{array} \right.$

$\left\{ \begin{array}{l} \text{Estados} \Rightarrow 3 \text{ bit} \\ \text{Salidas} \Rightarrow C_0 \div C_5 \Rightarrow 6 \end{array} \right. \left\{ \begin{array}{l} 9 \text{ bit bus} \\ \text{datos} \end{array} \right.$

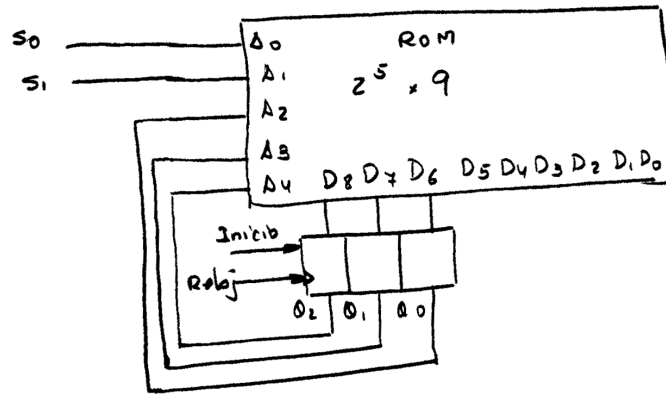


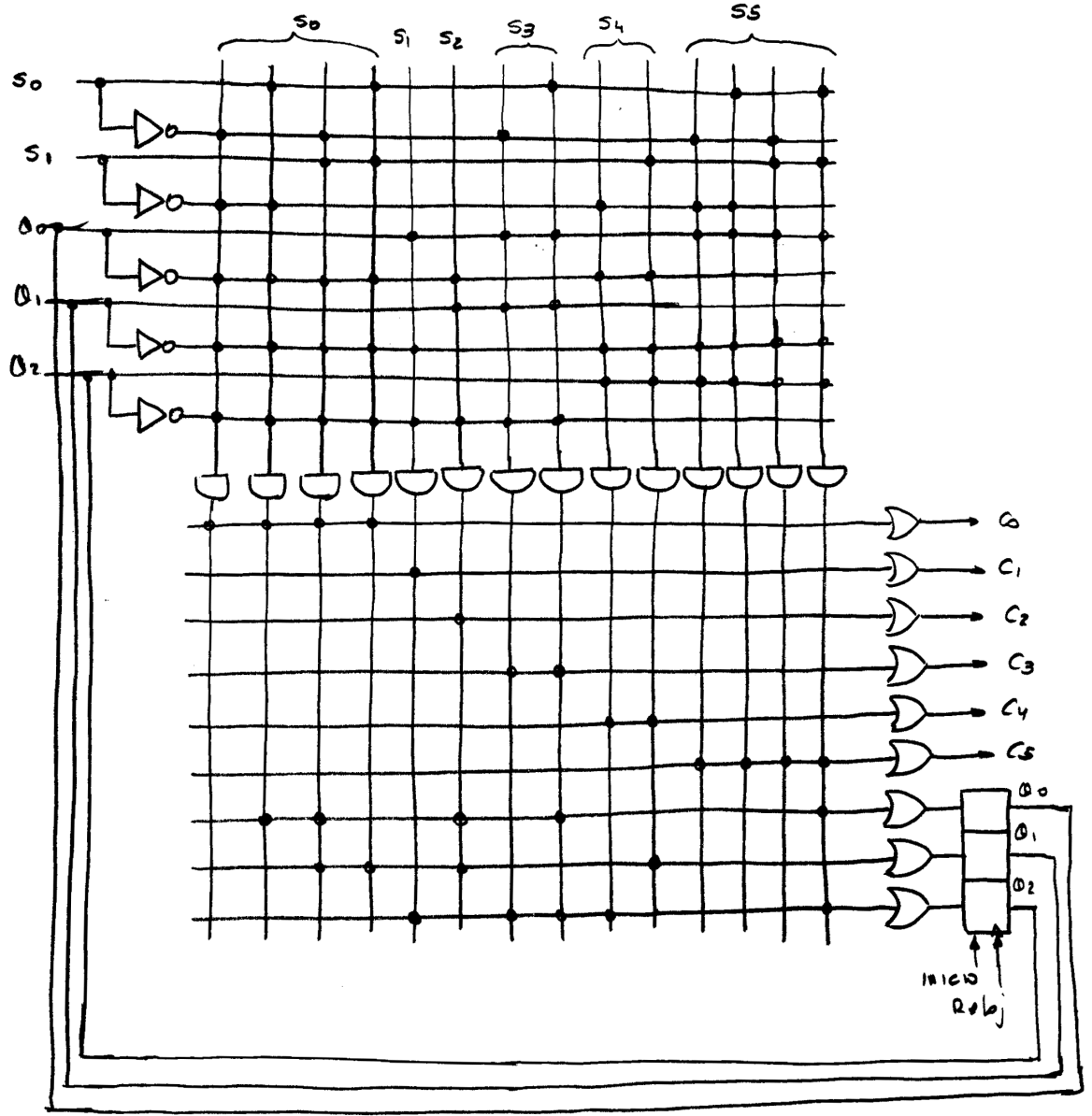
Tabla de la ROM

A_4	A_3	A_2	A_1	A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	1	0	0	0	0	0	0	0	0
0	1	0	1	0	0	0	0	0	0	0	0	0
0	1	0	1	1	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	0
1	0	1	0	1	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0	0	0
1	1	1	0	1	0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0	0

d) con PLA y un registro

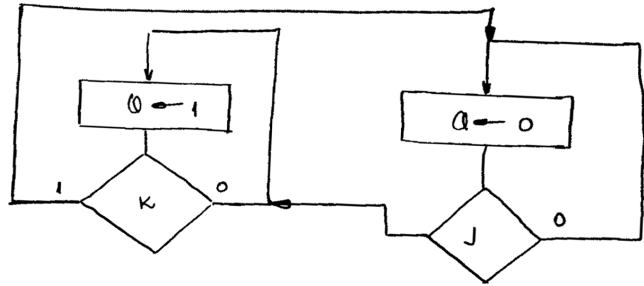
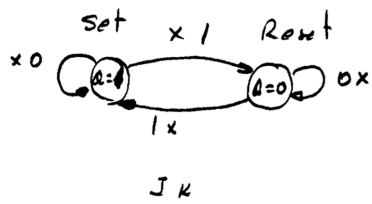
Entradas \Rightarrow Igual a ROM $\Rightarrow 5$

Salidas \Rightarrow " " " $\Rightarrow 9$



Problema 5.13

Diagrama ASM de una JK

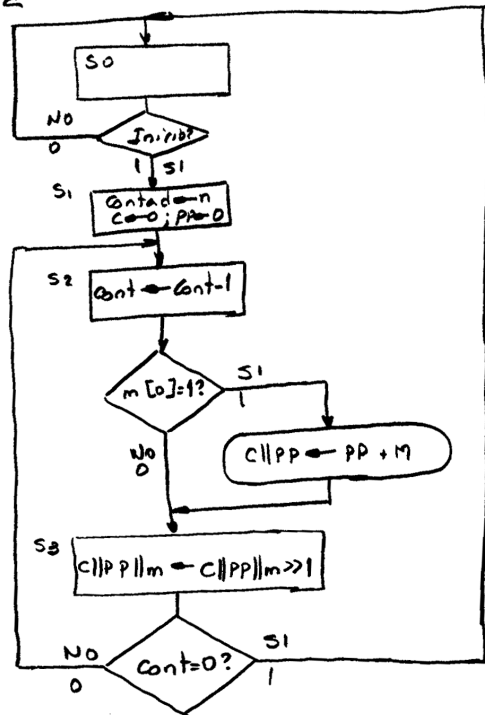


if (J=1) and (K=0) then Q := 1;
elseif (J=0) and (K=1) then Q := 0;
elseif (J=1) and (K=1) then Q := not Q;
end if

Problema 5.14

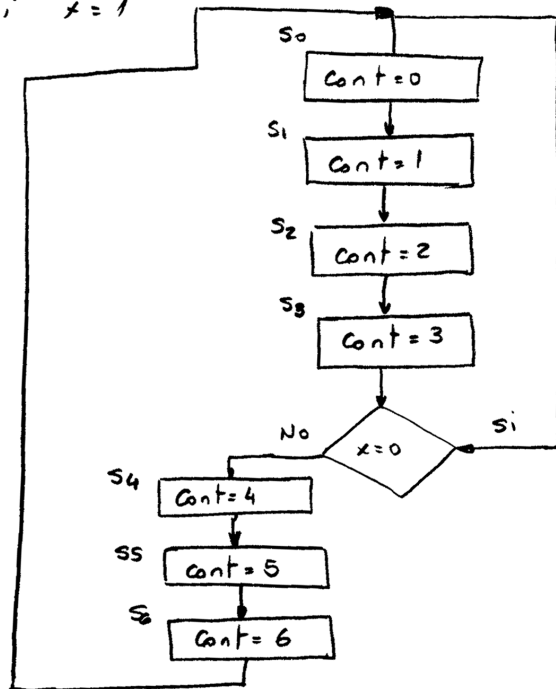
Diagrama ASM de multiplicador de n bits de desplazamiento y suma que suma el multiplicando (M) al producto parcial (PP) si el bit del multiplicador (m) es 1 y desplace el producto parcial y el multiplicador 1 bit a la derecha

- 1: Contador ← n
- 2: C ← 0
- 3: PP ← 0
- repeat
- 4: Contador ← Contador - 1
- 5: if m[0] = 1 then C || PP ← PP + M
- 6: C || PP || m ← C || PP || m >> 1
- until Contador = 0



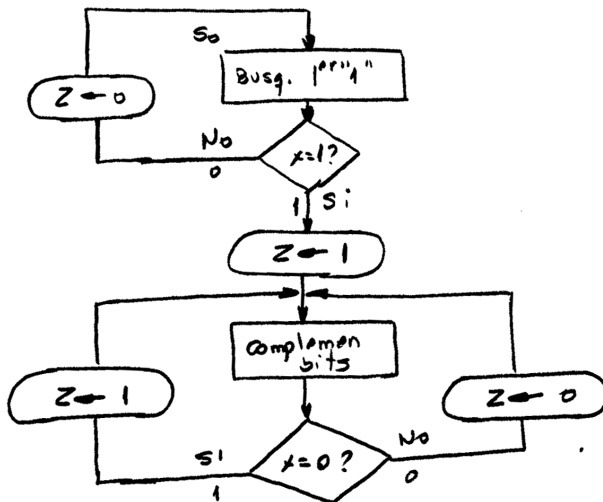
Problema 5.15

Diagrama ASM de contador en mod 4 si $x=0$ y en mod 7 si $x=1$



Problema 5.16

Diagrama ASM de un complementador a 2 serie. \Rightarrow se examinan los bits del n° a complementar de dcha a izda y se dejan inalterados hasta encontrar el primer "1", a partir de ese instante se invierten todos los bits
 $x \rightarrow n^{\circ}$ entrada y $z \rightarrow n^{\circ}$ salida

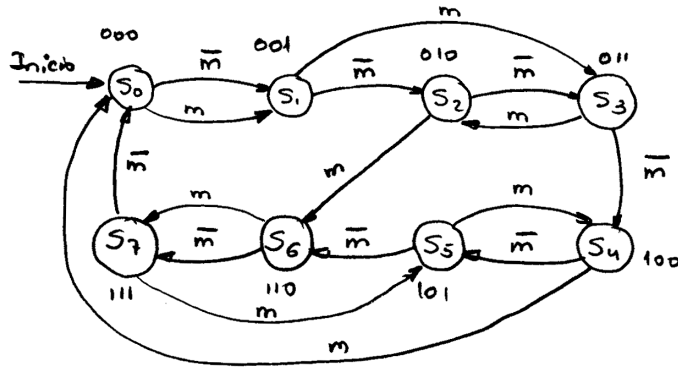


Problema 5.17

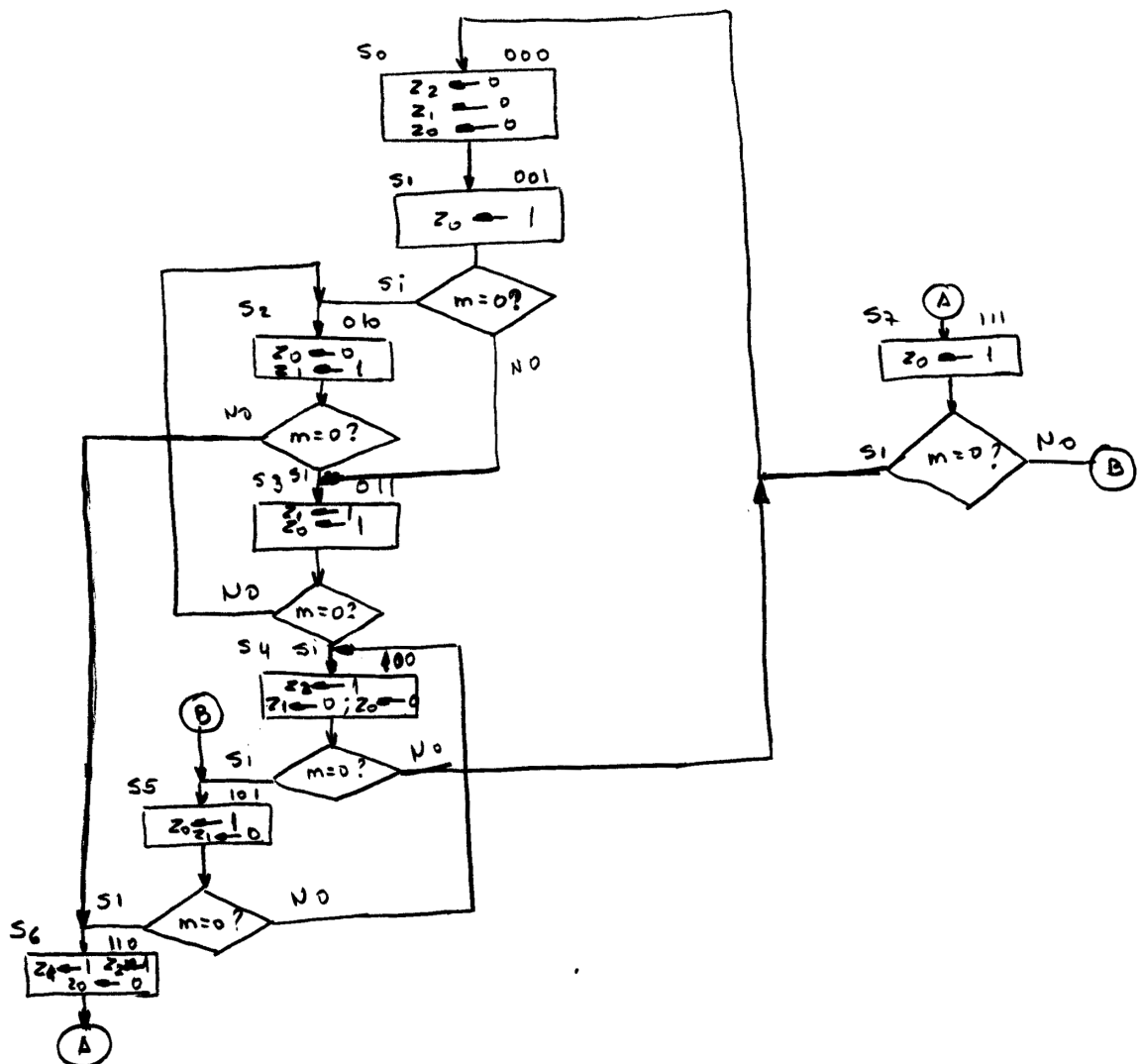
Diagrama ASM de un contador síncrono 3 bit con una entrada de control m y t_q

$m = 0 \Rightarrow 000, 001, 010, 011, 100, 101, 110, 111$

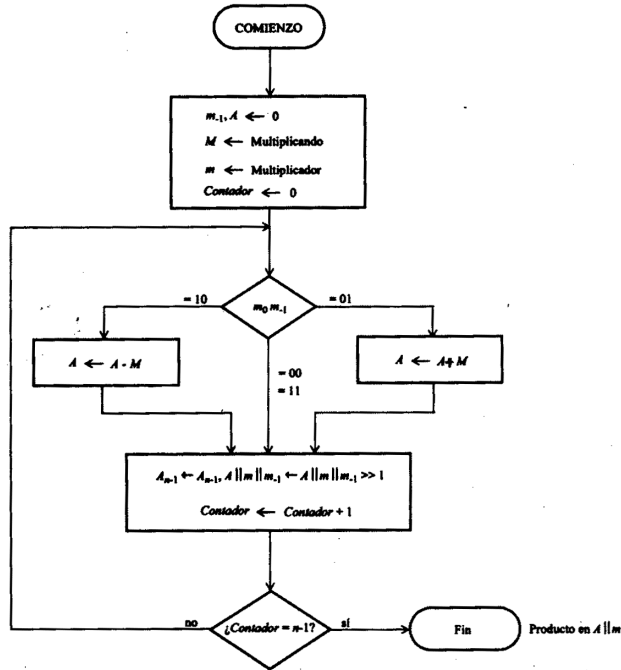
$m = 1 \Rightarrow 000, 001, 011, 010, 110, 111, 101, 100$



Salidas z_2, z_1, z_0



Problema 5.18

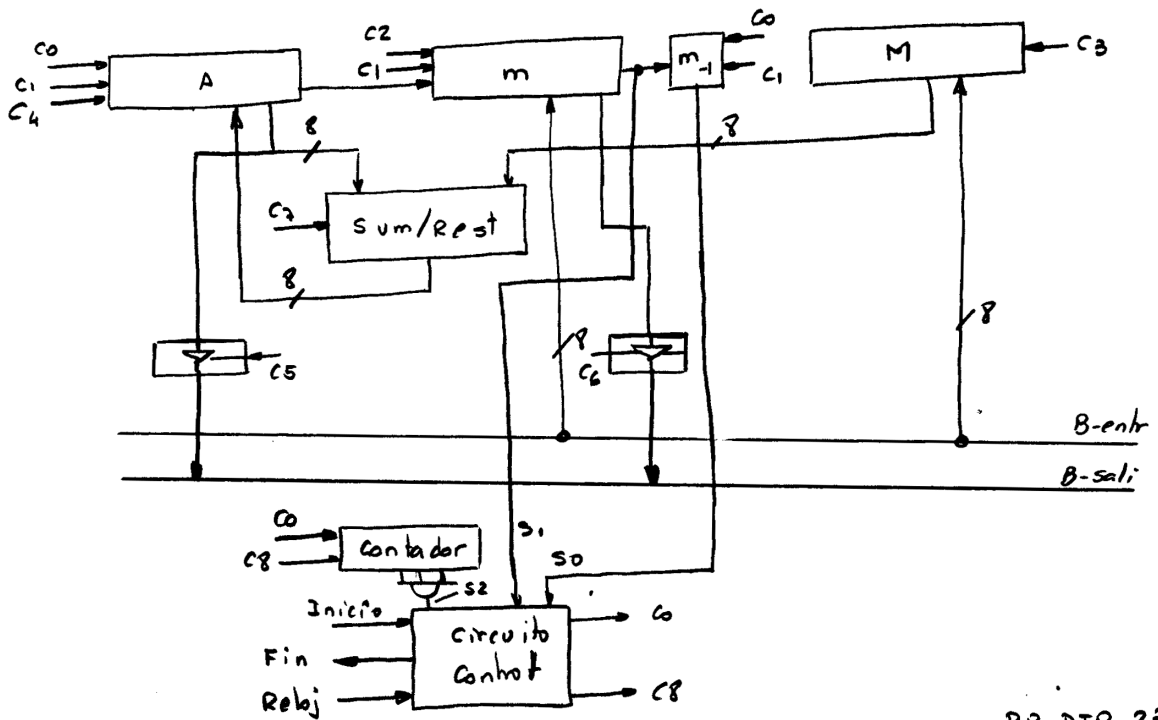


Multiplicador de 2
 n^{os} enteros representados
 en complemento a 2.
 Diseñar U.P. y U.C.
 N^{os} de 8 bits

Unidad de procesamiento

Registros { A [8]
 M [8]
 m [8]
 m_{n-1} [1]

Contador [8] Sum/Restador



- C₀ → Puesta a cero de A, m-1, Contador
- C₁ → Desplaz. dcha de A - m - m-1
- C₂ → Carga de m desde B-e
- C₃ → " " M " "
- C₄ → " " A " sum/rest
- C₅ → Salida de A a B-s
- C₆ → " " m " "
- C₇ → Suma / Resta
- C₈ → JNC contador A ← A-M

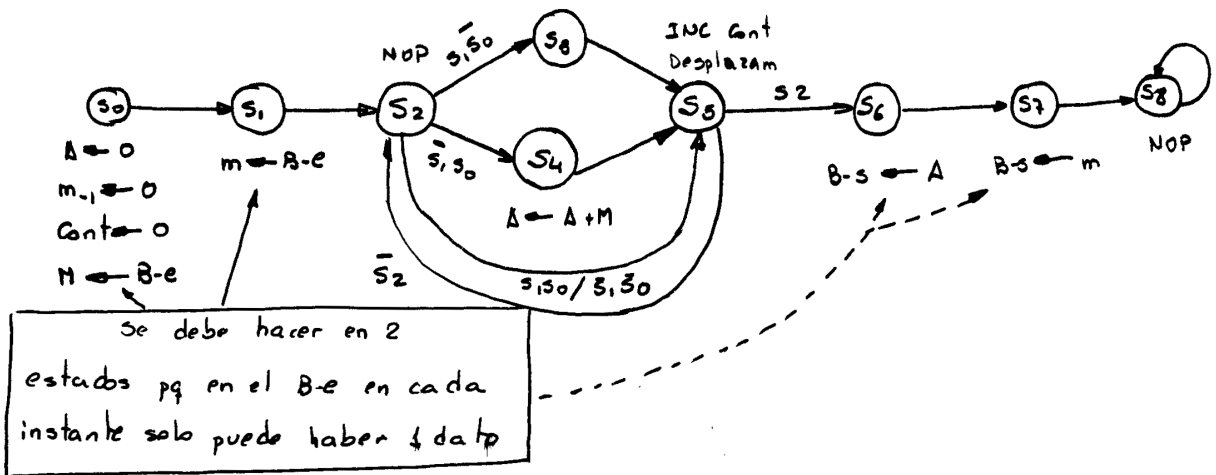


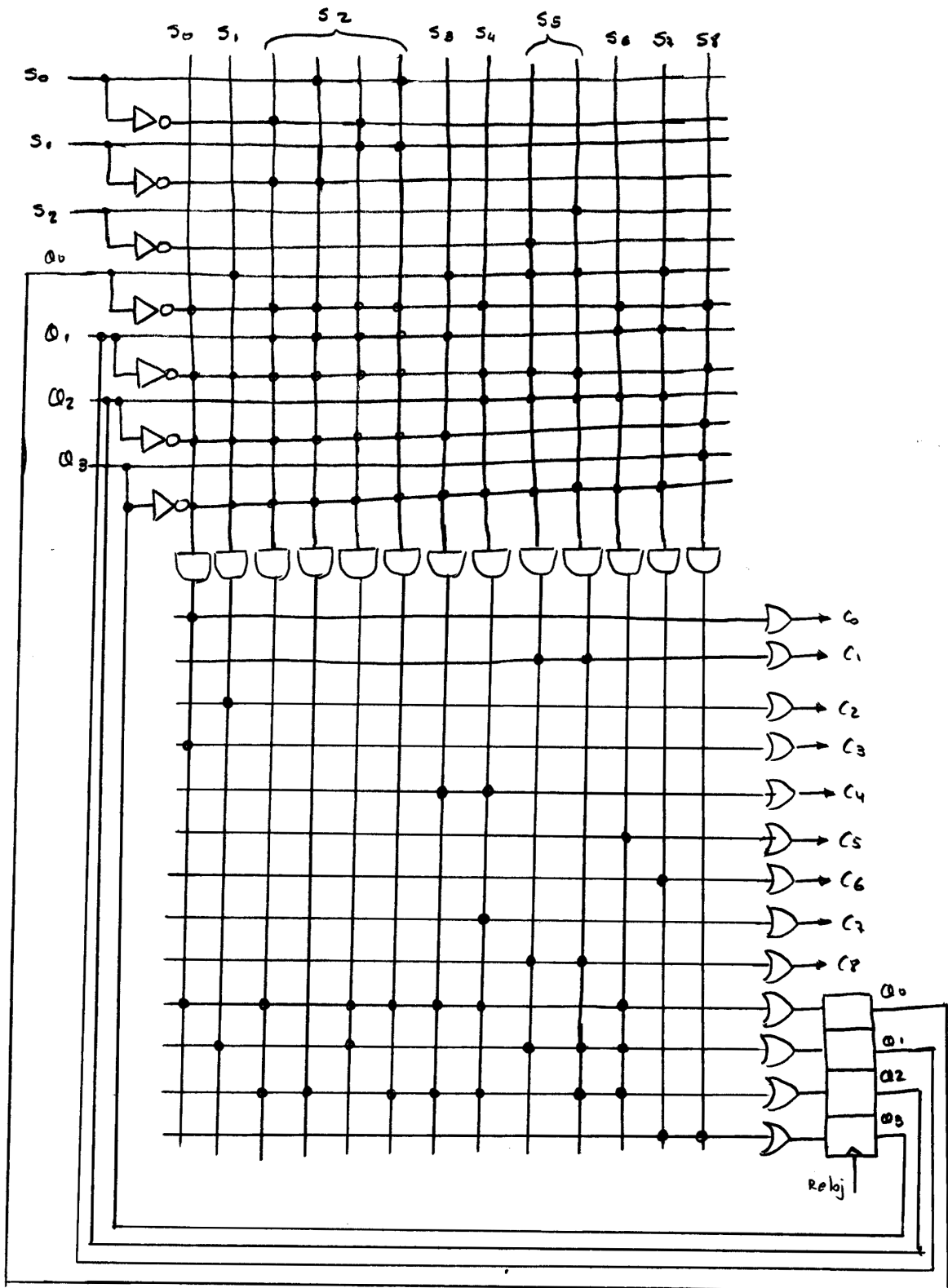
Tabla de estados

Entradas	Est. Actual	Est. Próximo	Salidas
s ₂ s ₁ s ₀	Q ₃ Q ₂ Q ₁ Q ₀	Q _{3,t+1} Q _{2,t+1} Q _{1,t+1} Q _{0,t+1}	C ₈ C ₇ C ₆ C ₅ C ₄ C ₃ C ₂ C ₁ C ₀
x x x	0 0 0 0	0 0 0 1	0 0 0 0 0 1 0 0 1
x x x	0 0 0 1	0 0 1 0	0 0 0 0 0 0 1 0 0
x ↓ 0	0 0 1 0	0 0 1 1	0 0 0 0 0 0 0 0 0
x 0 1	0 0 1 0	0 1 0 0	0 0 0 0 0 0 0 0 0
x x x	0 0 1 1	0 1 0 1	0 0 0 0 1 0 0 0 0
x x x	0 1 0 0	0 1 0 1	0 1 0 0 1 0 0 0 0
x 0 0	0 0 1 0	0 1 0 1	0 0 0 0 0 0 0 0 0
x ↓ 1	0 0 1 0	0 1 0 1	0 0 0 0 0 0 0 0 0
0 x x	0 1 0 1	0 0 1 0	1 0 0 0 0 0 0 1 0
↓ x x	0 1 0 1	0 1 1 0	1 0 0 0 0 0 0 1 0
x x x	0 1 1 0	0 1 1 1	0 0 0 1 0 0 0 0 0
x x x	0 1 1 1	1 0 0 0	0 0 1 0 0 0 0 0 0
x x x	1 0 0 0	1 0 0 0	0 0 0 0 0 0 0 0 0

Diseño con PLA y registro

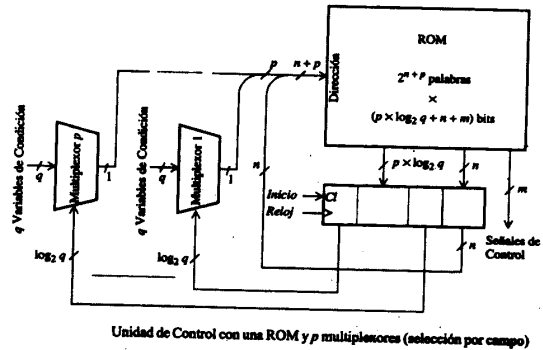
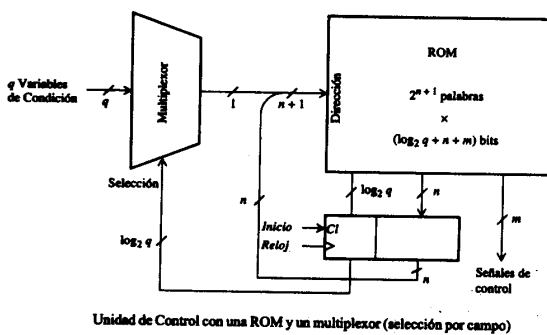
Entradas : $s_2, s_1, s_0, Q_3, Q_2, Q_1, Q_0 \Rightarrow 7$

Salidas : $c_8, c_0, c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8 \Rightarrow 13$



Problema 5.22.

La figura muestra una U.C. basada en ROM con multiplexor para seleccionar 1 de q condiciones. Generalizar el esquema cuando el próximo estado depende de p condiciones



Dentro del ancho de palabra de la ROM se distinguen 2 campos el campo p y el q .

Campo $q \Rightarrow \log_2 q$ con el que ponemos un n° codificado en binario de 0 a q con el que seleccionamos la variable que se va a tener en cuenta para el cambio de estado

Campo $p \Rightarrow$ con el que seleccionamos el n° de variables que influyen para el cambio de estado.

Ejemplo: si hay 8 variables de entrada s_0 a s_7 y para el cambio de estado influyen 3 variables de éstas.

$$q = 3 \Rightarrow 2^3 = 8 \quad p = 3 \Rightarrow \text{Campos ROM} - \left. \begin{array}{l} p = 3 \\ q = 3 \\ n \Rightarrow N^{\circ} \text{ estados} \end{array} \right\}$$



Supuesto que:

					p_0	p_1	p_2
					↓	↓	↓
Para paso de	S_0	a	S_1	influyen	s_0, s_4, s_6		
"	"	"	S_1 " S_2	"	s_3, s_5, s_6		
"	"	"	S_2 " S_3	"	s_4, s_6, s_7		

El contenido de la ROM sería:

Dirección	Campos $p+q$			Campo n	Salidas
	p_0	p_1	p_2		
000	0	0	0	s_0	100 s_4 110 s_6
001	0	1	1	s_3	101 s_5 110 s_6
010	1	0	0	s_4	110 s_6 111 s_7
⋮	⋮	⋮	⋮	⋮	⋮

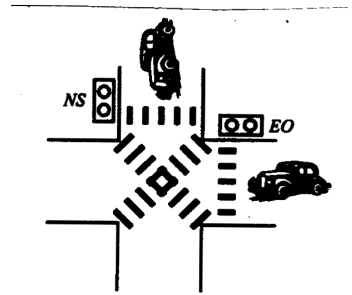
Problema 5.23

Diseñar el control de un semáforo.

- Dos señales NS y EO

- Pulsador peatones \Rightarrow

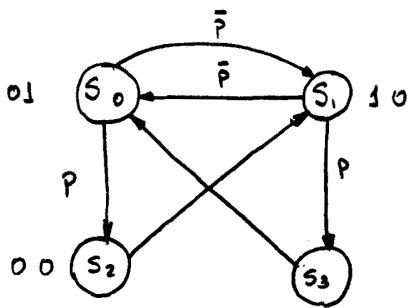
$$\begin{aligned}
 \hookrightarrow P \left\{ \begin{array}{l} p=0 \Rightarrow \text{NS y EO alternos } 1 \text{ min} \\ p=1 \Rightarrow \text{NS=EO=0 un } 1 \text{ min y luego a lterno} \end{array} \right.
 \end{aligned}$$



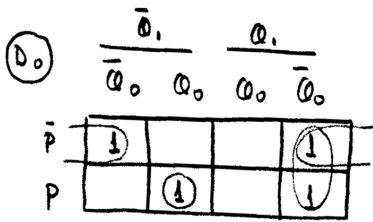
- El minuto se consigue en base a un reloj de 1 minuto.

Tabla de estados

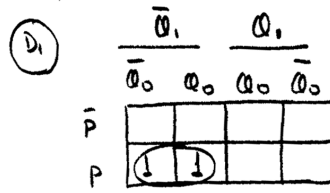
Diagrama de estados



Entra. P	Est Actv.		Est Próxi.		Salidas	
	Q_1	Q_0	Q_{1+1}	Q_{0+1}	NS	EO
0	0	0	0	↓	0	↓
↓	0	0	↓	0	0	↓
0	0	1	0	0	↓	0
↓	0	1	1	↑	↓	0
x	1	0	0	↓	0	0
x	1	1	0	0	0	0



$$D_0 = Q_1 \bar{Q}_0 + \bar{Q}_0 \bar{P} + \bar{Q}_1 Q_0 P$$



$$D_1 = \bar{Q}_1 P$$

Simplificar NS y EO directamente desde la tabla de estados, ya que directamente se ve que en cada caso solo cambia p = 0 se puede eliminar

