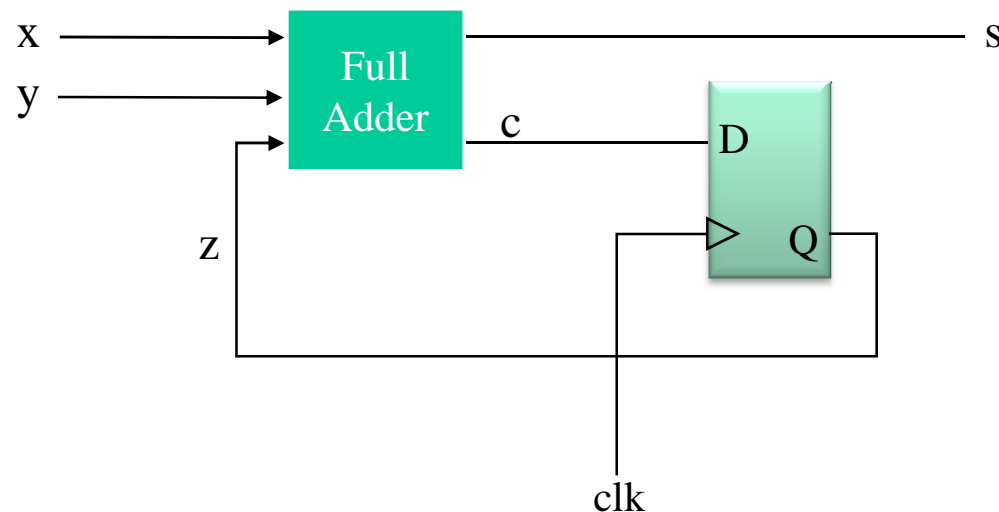


Digital System Design Course

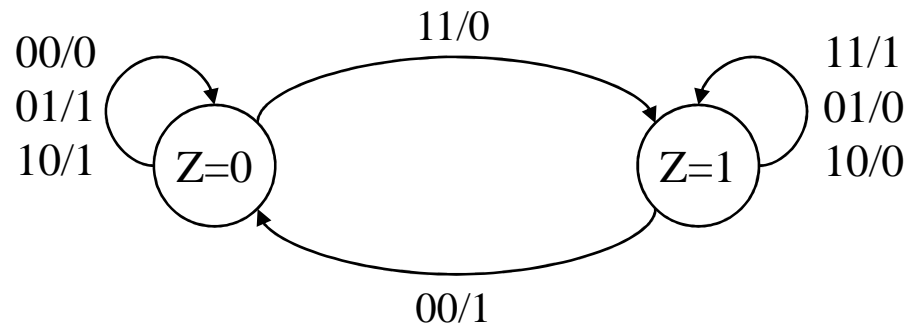
Ejercicio 1

□ El sumador completo de la figura recibe dos entradas externas X y Y ; la tercera entrada Z viene de la salida del Flip-flop D. El carry se transfiere al flip-flop en cada pulso de reloj. La salida externa S da la suma de X , Y , y Z . Obtenga la tabla de estados y el diagrama de estados del circuito secuencial.



Ejercicio 1

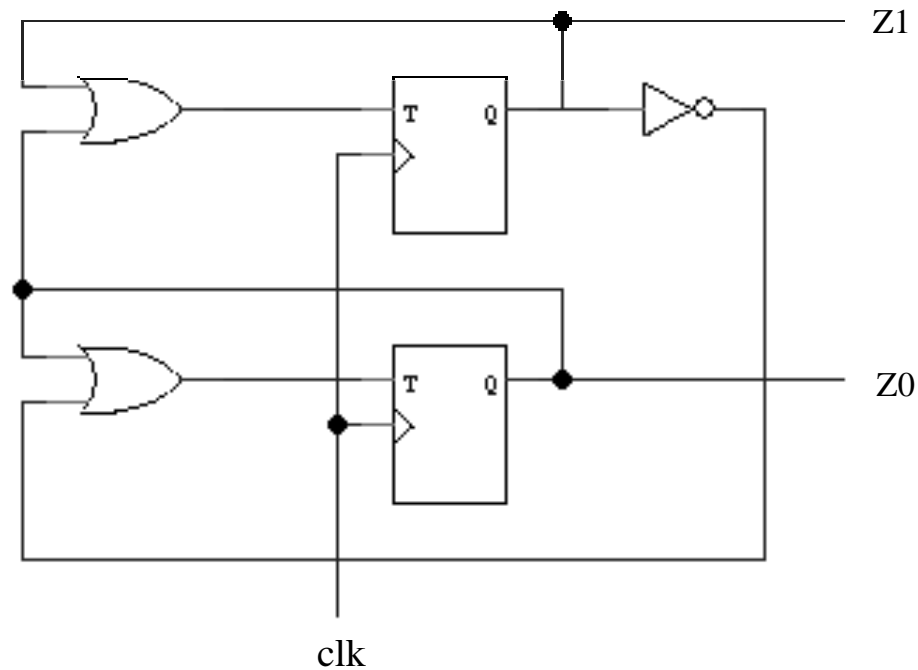
X	Y	Z	Z+	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



Entradas: xy
Salida: s

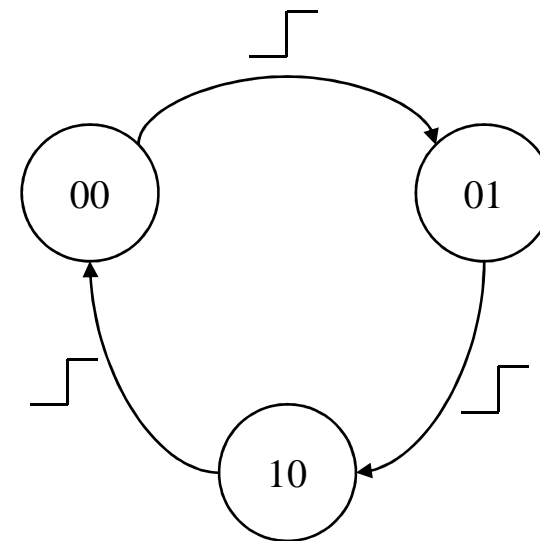
Ejercicio 2

- Deduzca la tabla de estado y diagrama de estado del circuito secuencial de la figura. Cual es la función del circuito?



Ejercicio 2

Estado Presente		Próximo Estado		Salida	
Q1	Q0	Q1+	Q0+	Z1	Z0
0	0	0	1	0	0
0	1	1	0	0	1
1	0	0	0	1	0
1	1	0	0	1	1



El circuito realiza la cuenta 00 – 01 – 10

Ejercicio 3

- Un circuito secuencial tiene dos flip-flops (A y B), dos entradas X y Y, y una salida Z. Las funciones de entrada de los flip-flops y la función de salida del circuito son las siguientes:

$$J_A = xB + y'B'$$

$$K_A = xy'B'$$

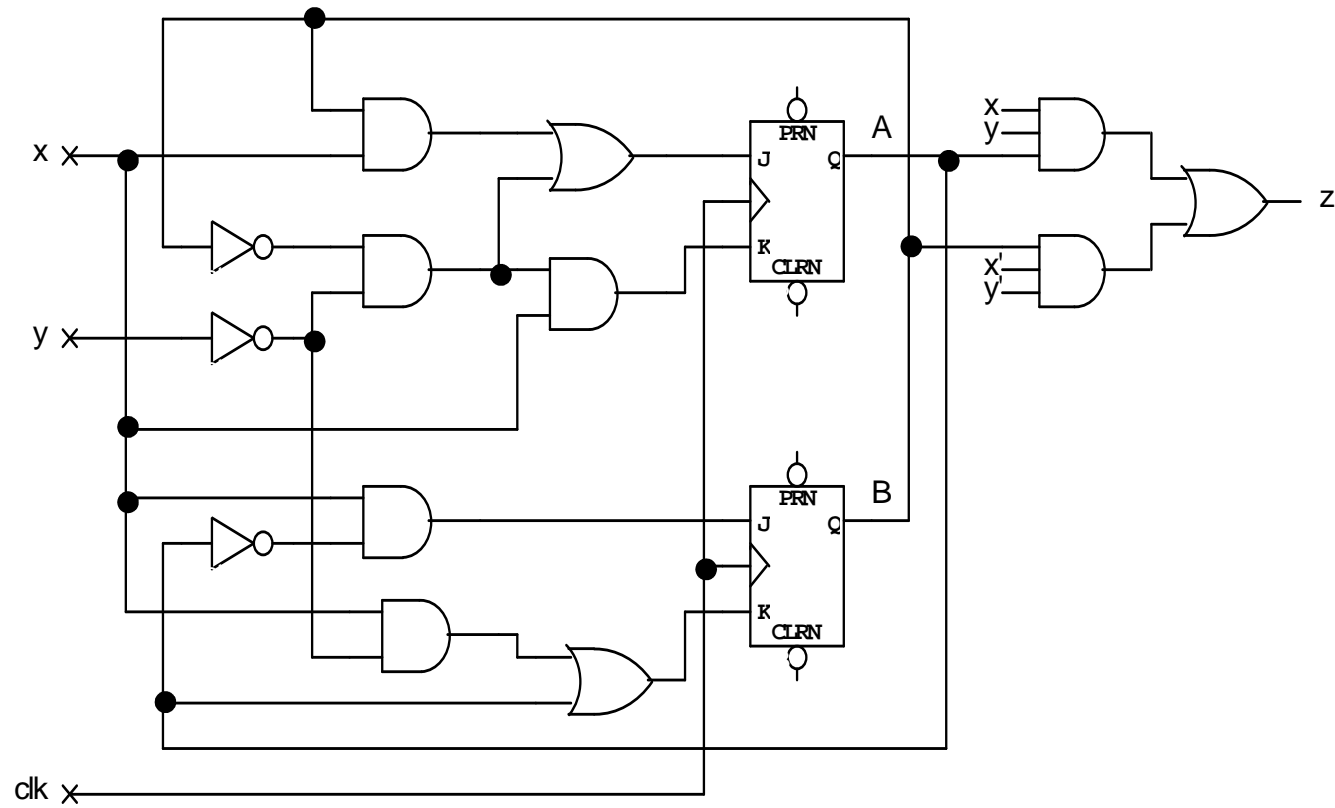
$$J_B = xA'$$

$$K_B = xy' + A$$

$$Z = xyA + x'y'B$$

Obtenga el diagrama lógico, la tabla de estado y el diagrama de estados.

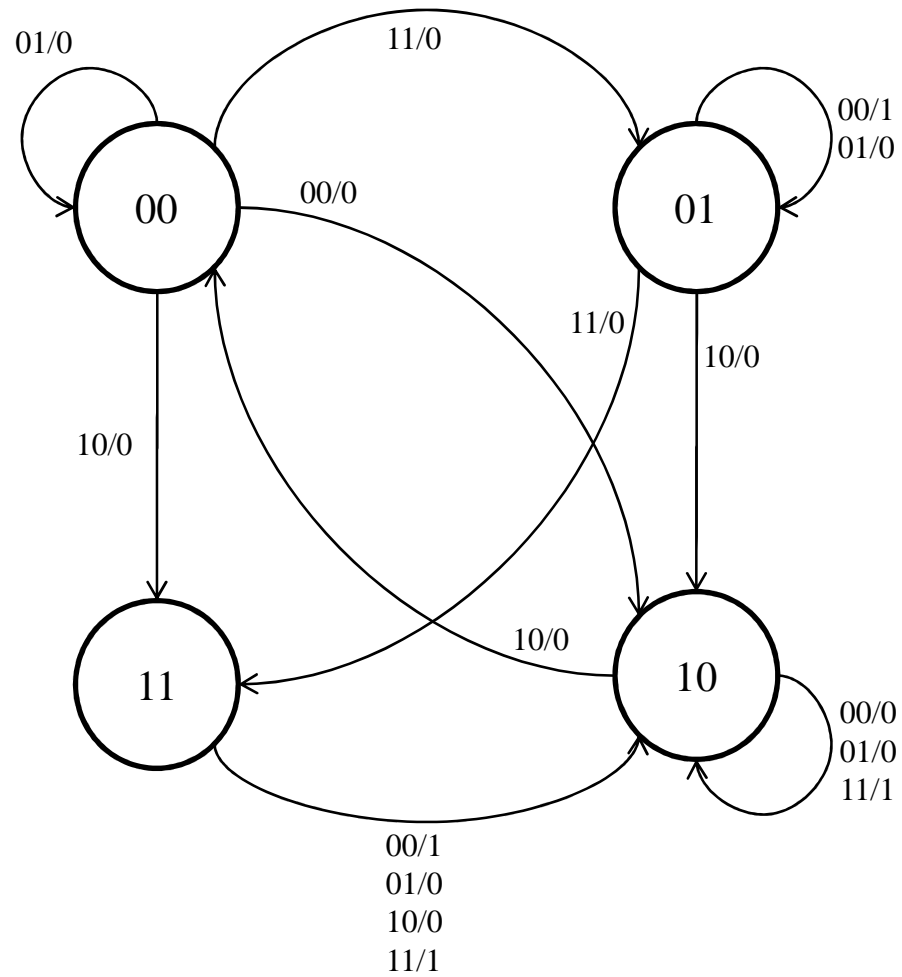
Ejercicio 3



Ejercicio 3

Estado Presente		Entradas		Entradas Flip-flops				Próximo Estado		Salida
A	B	x	y	JA	KA	JB	KB	A+	B+	z
0	0	0	0	1	0	0	0	1	0	0
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	1	1	1	1	1	0
0	0	1	1	0	0	1	0	0	1	0
0	1	0	0	0	0	0	0	0	1	1
0	1	0	1	0	0	0	0	0	1	0
0	1	1	0	1	0	1	1	1	0	0
0	1	1	1	1	0	1	0	1	1	0
1	0	0	0	1	0	0	1	1	0	0
1	0	0	1	0	0	0	1	1	0	0
1	0	1	0	1	1	0	1	0	0	0
1	0	1	1	0	0	0	1	1	0	1
1	1	0	0	0	0	0	1	1	0	1
1	1	0	1	0	0	0	1	1	0	0
1	1	1	0	1	0	0	1	1	0	0
1	1	1	1	1	0	0	1	1	0	1

Ejercicio 3

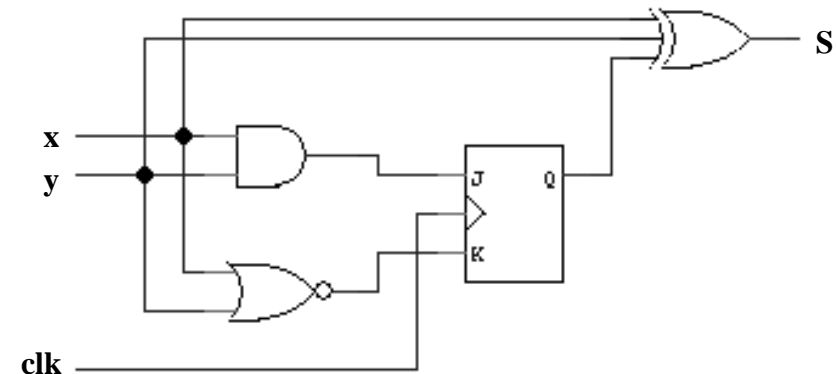


Ejercicio 4

- Diseñar un sumador en serie usando el procedimiento de lógica secuencial (tabla de estados). Utilizar flip-flops JK.

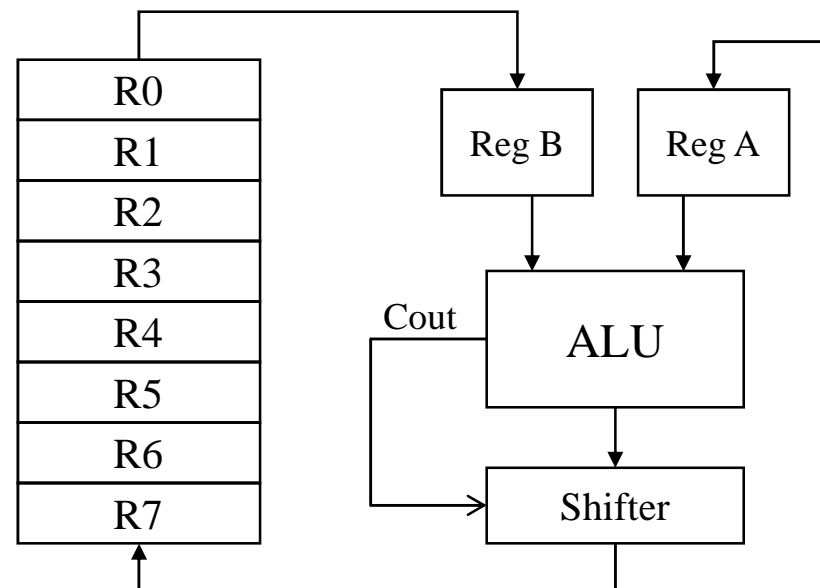
Estado Presente	Entradas		Proximo Estado	Salida	Flip-Flops	
Q	x	y	Q+	S	JQ	KQ
0	0	0	0	0	0	X
0	0	1	0	1	0	X
0	1	0	0	1	0	X
0	1	1	1	0	1	X
1	0	0	0	1	X	1
1	0	1	1	0	X	0
1	1	0	1	0	X	0
1	1	1	1	1	X	0

El estado presente Q es el valor presente del carry.



Ejercicio 5

- Es necesario calcular el valor promedio de cuatro números binarios sin signo almacenados en los registros R0, R1, R2 y R3 del procesador definido en la figura. El valor promedio se debe almacenar en el registro R4. Se debe tener cuidado de no causar sobrecapacidad. Utilizar el menor numero de estados posibles.
- ❖ Dar la lista de la secuencia de operaciones en RTL.
- ❖ Listar las señales de control binarias correspondientes



Ejercicio 5

Para evitar la sobrecapacidad en la ALU, primero se debe realizar el promedio entre R0 y R1, y almacenar el resultado temporal en el registro R4. Lo mismo se hace con los registros R2 y R3 almacenando el resultado en RA. Finalmente se calcula el promedio entre R4 y RA obteniendo el promedio total.

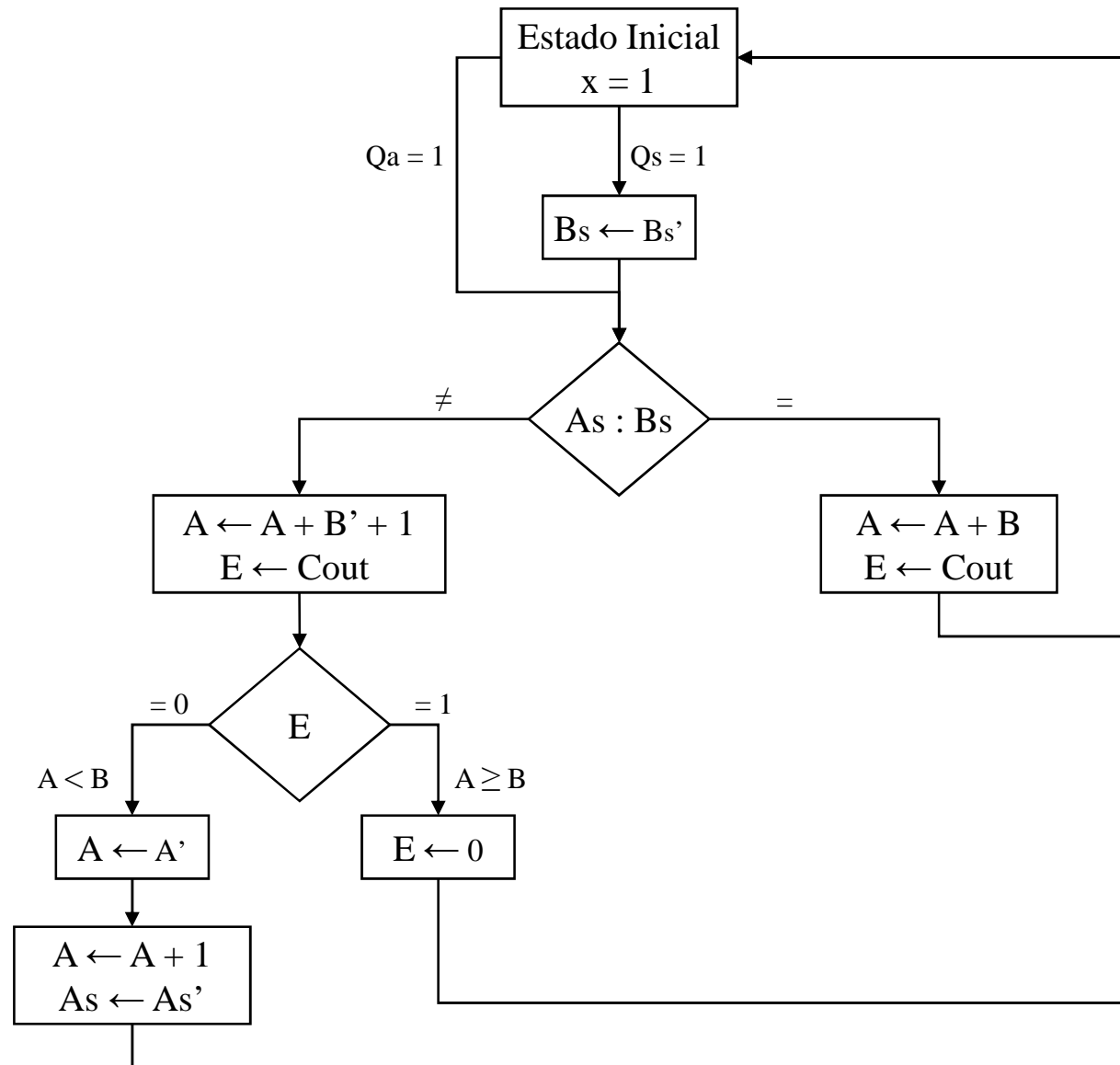
S0: $RB \leftarrow R0$	OC_0
S1: $RA \leftarrow RB, RB \leftarrow R1$	ALU = 0, OC_1
S2: $R4 \leftarrow \text{shr}(RA + RB), RB \leftarrow R2$	EN_4, ALU = 1, SHR, OC_2
S3: $RA \leftarrow RB, RB \leftarrow R3$	ALU = 0, OC_3
S4: $RA \leftarrow \text{shr}(RA + RB), RB \leftarrow R4$	ALU = 1, SHR, OC_4
S5: $R4 \leftarrow \text{shr}(RA + RB)$	ALU = 1, SHR, EN_4

ALU:	0: Transferir B (no suma)
	1: Sumar A + B

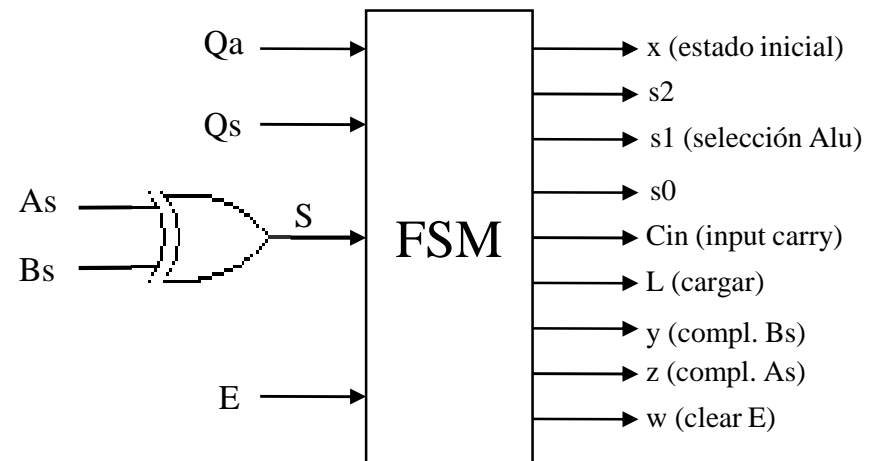
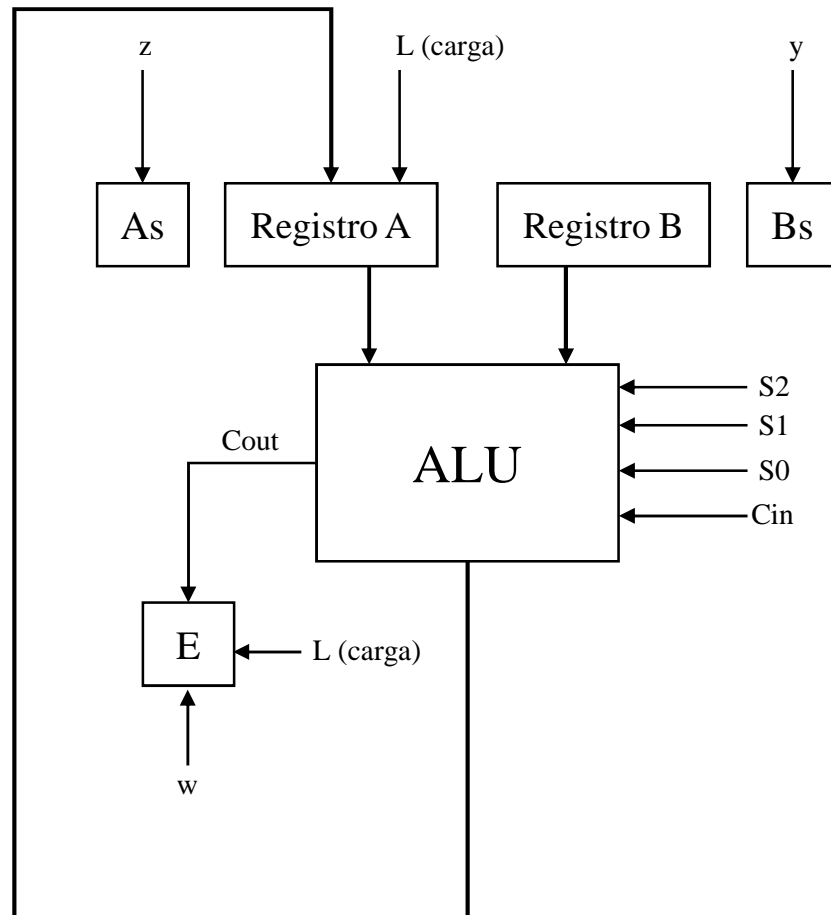
Ejercicio 6

- Diseñar un circuito que realice la adición y sustracción de dos números binarios de punto fijo representado en forma de signo-magnitud. Se puede usar aritmética complementada siempre y cuando el resultado final esté en la forma de signo-magnitud. El circuito debe tener un flip-flop para almacenar el bit de desbordamiento por sobrecapacidad.

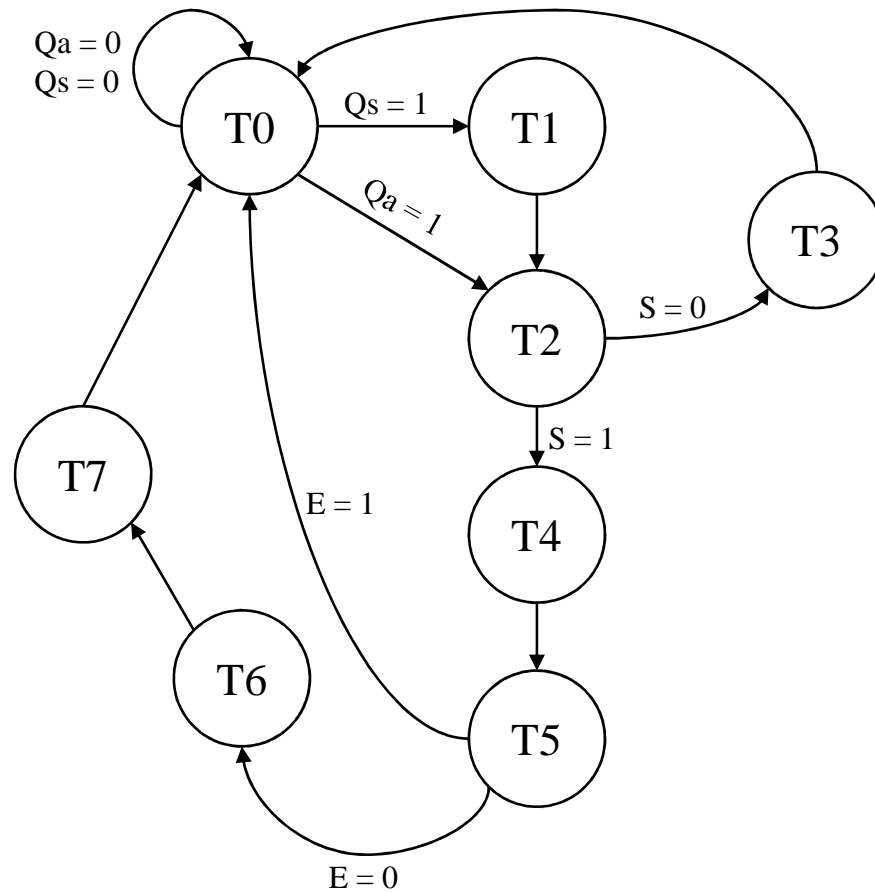
Ejercicio 6



Ejercicio 6



Ejercicio 6



$Qa = 1$	Sumar
$Qs = 1$	Restar
$S = 0$	Signos iguales
$S = 1$	Signos diferentes
E	Output Carry

Ejercicio 6

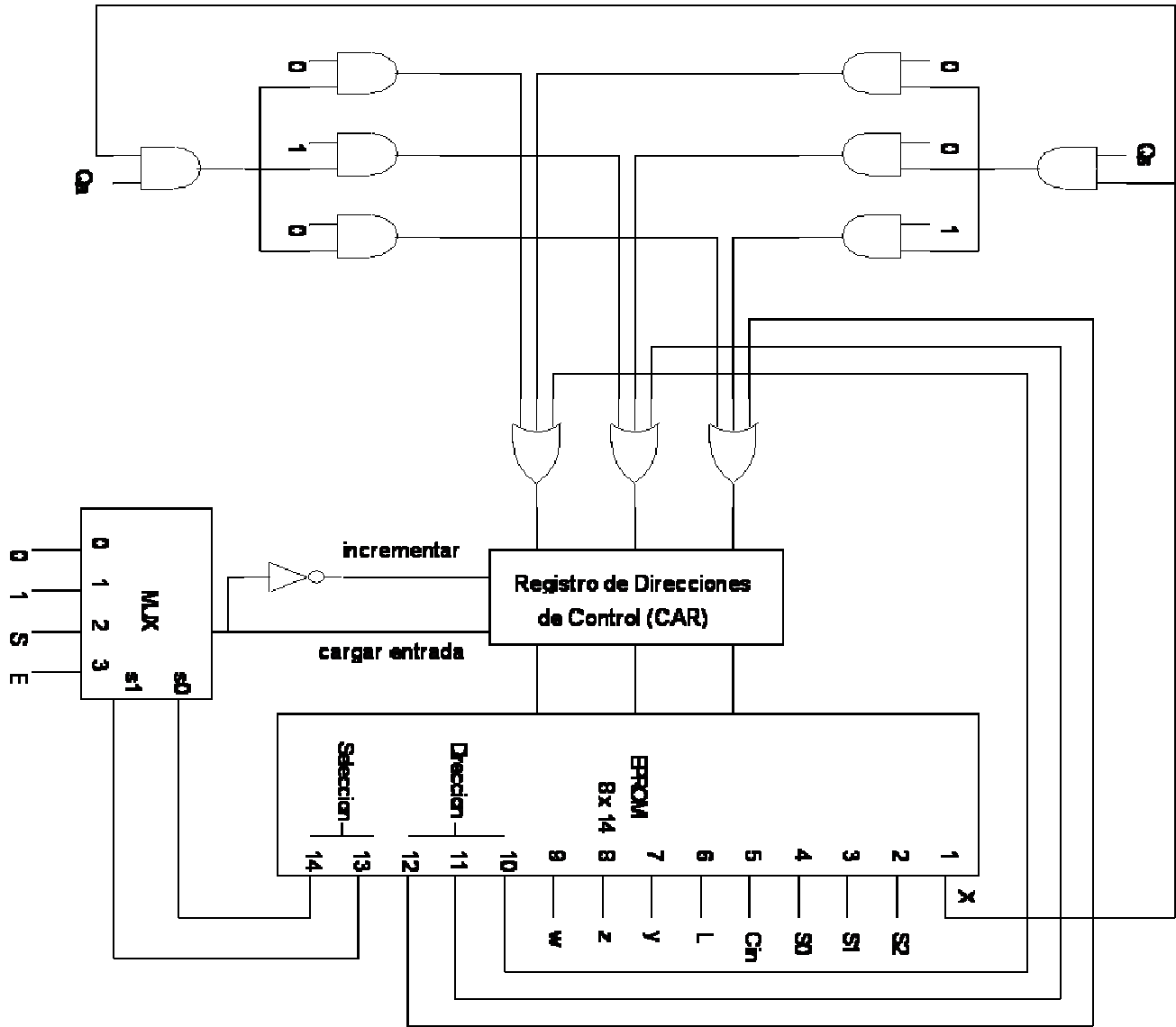
	x	s2	s1	s0	Cin	L	Y	z	w
T0: Estado Inicial $x = 1$	1	0	0	0	0	0	0	0	0
T1: $Bs \leftarrow Bs'$	0	0	0	0	0	0	1	0	0
T2: nada	0	0	0	0	0	0	0	0	0
T3: $A \leftarrow A + B$, $E \leftarrow Cout$	0	0	0	1	0	1	0	0	0
T4: $A \leftarrow A + B' + 1$, $E \leftarrow Cout$	0	0	1	0	1	1	0	0	0
T5: $E \leftarrow 0$	0	0	0	0	0	0	0	0	1
T6: $A \leftarrow A'$	0	1	1	1	0	1	0	0	0
T7: $A \leftarrow A + 1$, $As \leftarrow As'$	0	0	0	0	1	1	0	1	0

Salidas de la FSM por estado

Ejercicio 7

- Implementar el diseño del ejercicio anterior utilizando un registro de desplazamiento para la secuencia del algoritmo y una memoria ROM para la lógica combinatorial de las señales de control necesarias. El registro tiene capacidad de carga en paralelo con el fin de poder hacer “saltos” en la secuencia del programa. Se puede utilizar otros componentes si es necesario.

Ejercicio 7



Ejercicio 7

Bits de ROM		Función de selección del MUX
13	14	
0	0	Incrementar el CAR
0	1	Cargar la entrada al CAR
1	0	Cargar las entradas al CAR si S=1, incrementar el CAR si S=0
1	1	Cargar las entradas al CAR si E=1, incrementar el CAR si E=0

Dirección De ROM	Microinstrucción	Comentarios
0	$x=1$, si ($Q_s=1$) entonces (va a 1), si ($Q_a=1$) entonces (va a 2), si ($Q_s \text{ OR } Q_a = 0$) entonces (va a 0)	Cargar 0 o dirección externa
1	$B_s \leftarrow B_s'$	$Q_s = 1$, comenzar sustracción
2	If ($S = 1$) entonces (va a 4)	$Q_a = 1$, comenzar suma
3	$A \leftarrow A + B$, $E \leftarrow \text{Cout}$, va a 0	Sumar magnitudes y regreso
4	$A \leftarrow A + B' + 1$, $E \leftarrow \text{Cout}$	Sustraer Magnitudes
5	Si ($E = 1$) entonces (va a 0), $E \leftarrow 0$	Operación Finalizada si $E = 1$
6	$A \leftarrow A'$	$E = 0$, complementar A
7	$A \leftarrow A + 1$, $A_s \leftarrow A_s'$, va a 0	Terminado, regresar a la dirección 0

Microprograma simbólico para la memoria de control

Ejercicio 7

Dirección EPROM	Salidas de EPROM													
	x	s2	s1	s0	Cin	L	y	z	w	Dirección			Selección	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
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	1	0	0	0	1	0	0	1
0 1 0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
0 1 1	0	0	0	1	0	1	0	0	0	0	0	0	0	1
1 0 0	0	0	1	0	1	1	0	0	0	1	0	1	0	1
1 0 1	0	0	0	0	0	0	0	0	1	0	0	0	1	1
1 1 0	0	1	1	1	0	0	0	0	0	1	1	1	0	1
1 1 1	0	0	0	0	1	1	0	1	0	0	0	0	0	1

Microprograma binario para la memoria de control

Ejercicio 8

- Diseñe un sistema digital que sume y reste dos números binarios de punto fijo representados en la forma de signo-complemento de 2. Incluya una indicación de sobrecapacidad.

Ejercicio 9

- Diseñe un circuito que compare dos números binarios sin signo almacenados en R1 y R2. El registro que contiene el número menor se borra entonces. Si los dos números son iguales, se borran ambos registros. Use la configuración del Ejercicio 5.

Ejercicio 10

- Diseñe un circuito que ejecute la multiplicación de dos números binarios de 8 bits por el método de sumas sucesivas.

