

# **CAPÍTULO IV**

## **CONSTRUCCIÓN DE MÁQUINAS DE ESTADOS USANDO SECUENCIADORES**

---

## 4.1 EL SECUENCIADOR BÁSICO

En el capítulo anterior se vio el diseño de máquinas de estados usando memorias y dispositivos tales como contadores. Para el diseño de los módulos de control de una computadora se requieren máquinas de estados que sean capaces de ejecutar algoritmos más complejos. Haciendo modificaciones y agregando componentes a la variante del direccionamiento implícito se pueden crear máquinas de estados que efectúen cartas ASM con llamadas a subrutinas, estructuras DO WHILE, iteraciones tipo FOR, entre otras. Los dispositivos que son capaces de efectuar este tipo de operaciones son llamados **secuenciadores**.

La figura 4.1 muestra el diagrama de bloques de un secuenciador básico. Como puede observar en el diagrama, la dirección del estado siguiente, dada por el bus Y, puede venir de dos lugares posibles: 1) del registro  $\mu$ PC, ó 2) de la entrada D.

1. El registro de micro-programa ( $\mu$ PC) contiene la dirección del estado presente más uno, es decir, la dirección que se encuentra a la salida del multiplexor es incrementada en una unidad y cargada en este registro en el siguiente ciclo de reloj.
2. En la entrada D se introduce una dirección de salto. Esta dirección puede venir de tres lugares diferentes: del campo de liga, del registro de transformación o del registro de interrupciones.

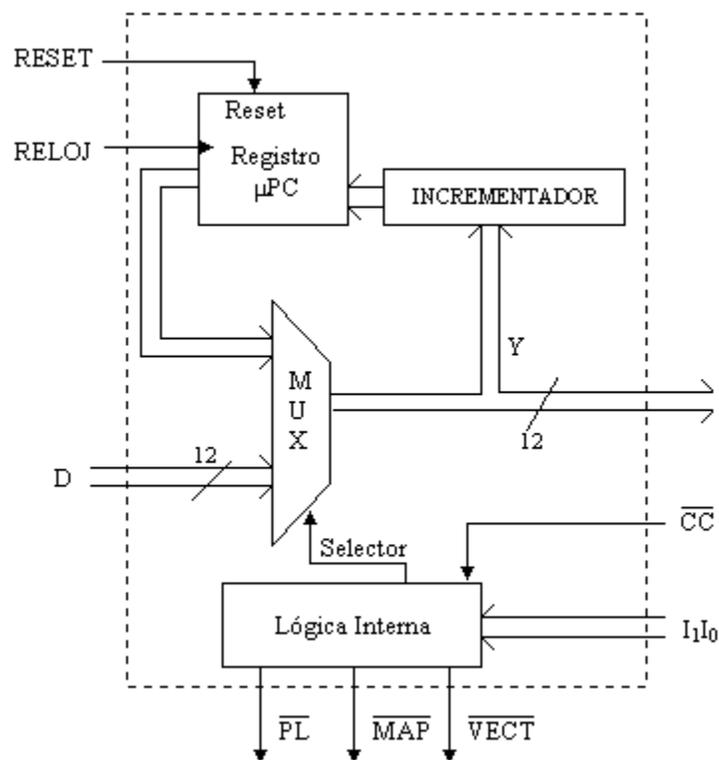


Figura 4.1. Diagrama de bloques del secuenciador básico.

El secuenciador cuenta con una lógica interna que se encarga de generar las señales que controlan al multiplexor. Dependiendo de la instrucción dada por las líneas  $I_1$  e  $I_0$  y de la línea  $CC$ , la lógica es capaz de seleccionar entre la salida del registro  $\mu PC$  o la entrada D. Dicha salida direcciona una memoria que contiene el estado siguiente del algoritmo de la máquina de estados.

La lógica interna también genera las líneas  $PL$ ,  $MAP$  y  $VECT$ , las cuales seleccionan unos registros cuyas salidas están conectadas a la entrada D del secuenciador. De esta forma la dirección de salto puede venir de tres lugares distintos. Esta característica se utilizará cuando se diseñe la unidad central de procesos (UCP), como se verá más adelante.

La figura 4.2 muestra las señales de entrada y salida del secuenciador.

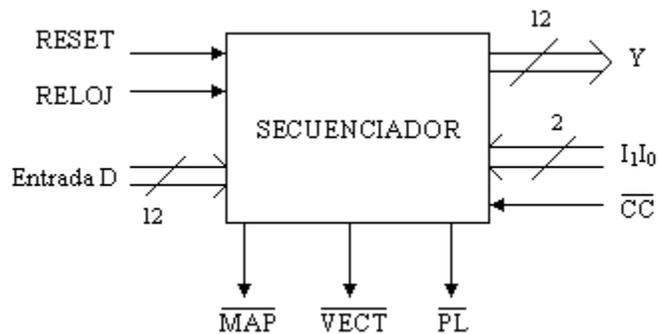


Figura 4.2. Secuenciador básico.

A continuación se muestran las instrucciones que el secuenciador puede ejecutar y su representación en carta ASM.

## 4.2 INSTRUCCIONES PARA EL SECUENCIADOR

### 4.2.1 CONTINÚA (C)

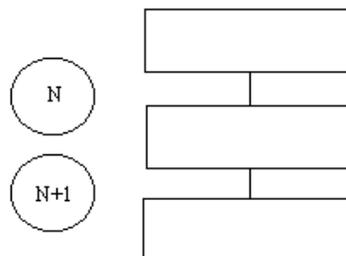


Figura 4.3. Representación en notación ASM de la instrucción continua.

En la instrucción continúa la dirección del estado siguiente la proporciona el registro  $\mu\text{PC}$ .

#### 4.2.2 SALTO CONDICIONAL (SCO)

En esta instrucción se revisa el valor de la línea  $\overline{CC}$ , si es igual a uno, la dirección del estado siguiente la proporciona el registro  $\mu\text{PC}$ ; si es igual a cero, la dirección del estado siguiente, contenida en el registro seleccionado por  $\text{PL}$ , ingresa a través de la entrada  $D$ .

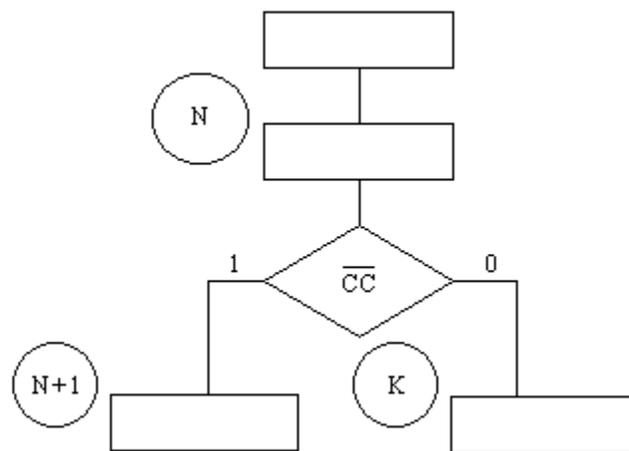


Figura 4.4. Representación en notación ASM de la instrucción SCO.

#### 4.2.3 SALTO DE TRANSFORMACIÓN (ST)

La dirección del estado siguiente se obtiene del registro seleccionado por la línea de  $\text{MAP}$ . Este registro también está conectado a la entrada  $D$ . Aquí se introduce una nueva notación de carta ASM: un rombo con varias bifurcaciones. La bifurcación que se elija dependerá del contenido del registro seleccionado por  $\text{MAP}$ .

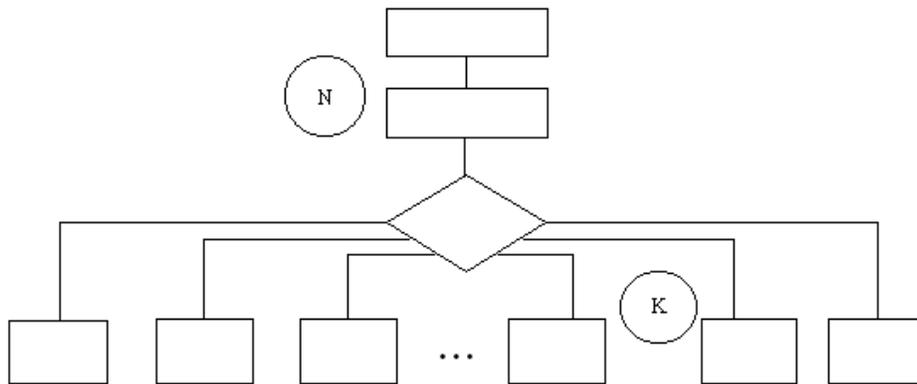


Figura 4.5. Representación en notación ASM de la instrucción ST.

#### 4.2.4 SALTO CONDICIONAL USANDO LA DIRECCIÓN DE LAS INTERRUPTACIONES (SCI)

En esta instrucción se revisa el valor de  $CC$ , si es igual a uno, la dirección del estado siguiente proviene del registro  $\mu PC$ ; si es igual a cero, la dirección del estado siguiente, contenida en el registro seleccionado por  $VECT$ , ingresa a través de la entrada D.

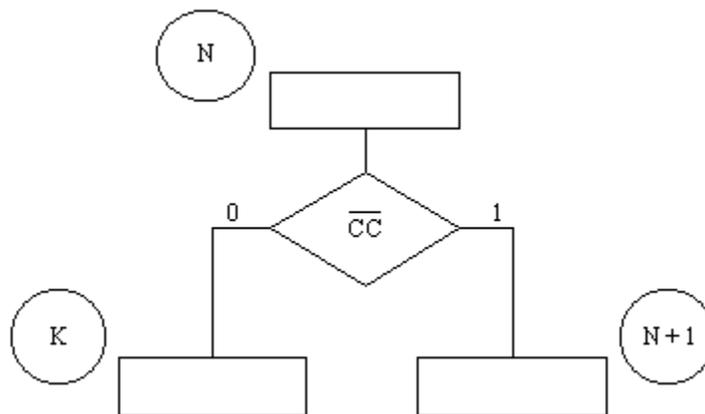


Figura 4.6. Representación en notación ASM de la Instrucción SCI.

La lógica interna del secuenciador se construye a partir de la siguiente tabla.

Entradas			Salidas				
I1	I0	CC	Selector	PL	MAP	VECT	Y
0	0	0	0	1	1	1	$\mu PC$
0	0	1	0	1	1	1	$\mu PC$

0	1	0	1	0	1	1	Entrada D
0	1	1	0	0	1	1	$\mu$ PC
1	0	0	1	1	0	1	Entrada D
1	0	1	1	1	0	1	Entrada D
1	1	0	1	1	1	0	Entrada D
1	1	1	0	1	1	0	$\mu$ PC

Tabla. 4.1. Entradas y salidas de la lógica interna del secuenciador.

### 4.3 SECUENCIADORES Y MEMORIAS

La figura 4.7 muestra el diagrama de bloques de un secuenciador conectado a una memoria.

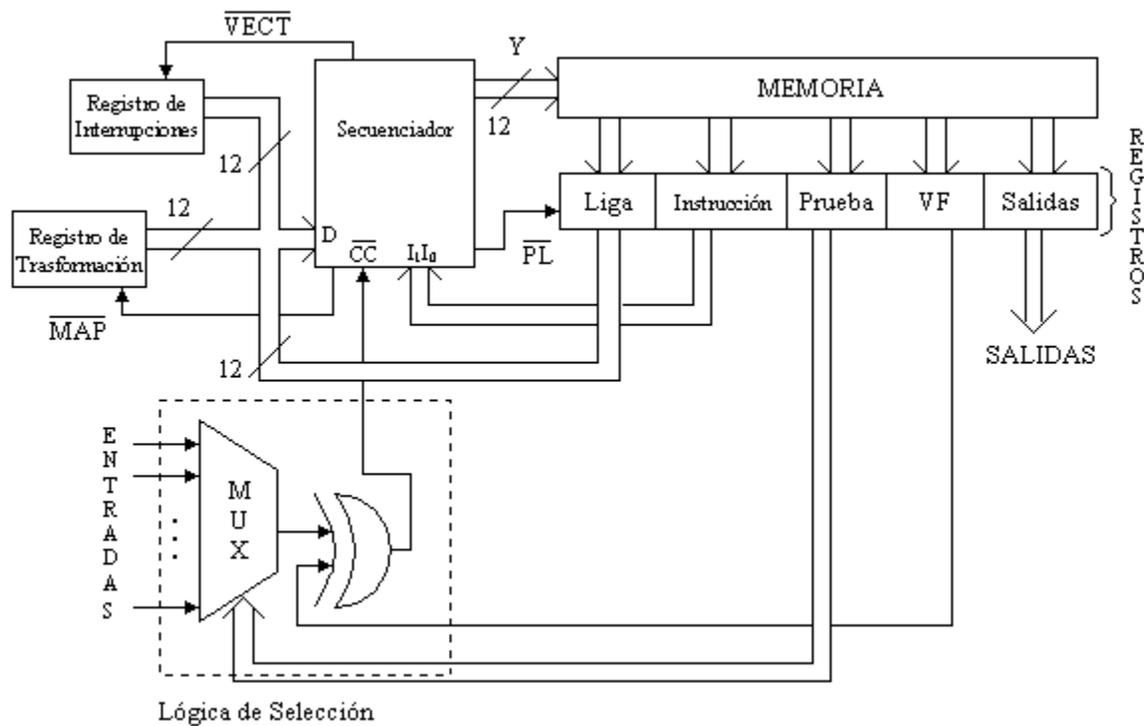


Figura 4.7. Construcción de cartas ASM usando un secuenciador básico y memorias.

Usando este secuenciador, el contenido de la memoria para la carta ASM de la figura 3.11 queda como se muestra en la tabla 4.2.

<b>Dirección de la Memoria</b>			<b>Contenido de Memoria</b>												
Estado Presente			Liga			Micro Instrucción		Prueba		VF	Salidas				
S1	S2	S3	S1	S2	S3	M1	M2	P1	P2	VF	S1	S2	S3	S4	S5
0	0	0	0	1	0	0	1	1	0	1	1	1	0	0	0
0	0	1	0	1	1	0	1	0	0	1	0	1	0	0	1
0	1	0	1	0	0	0	1	0	1	1	0	0	1	0	0
0	1	1	0	0	1	0	1	1	1	0	0	0	1	1	0
1	0	0	0	0	0	0	1	0	0	1	0	1	0	1	0

Tabla 4.2. Contenido de la memoria para la figura 3.11 usando el secuenciador.

## 4.4 IMPLANTACIÓN DE CARTAS ASM USANDO SECUENCIADORES

En la siguiente figura se presenta una carta ASM en donde se hace uso de todas las instrucciones que este secuenciador básico puede ejecutar.

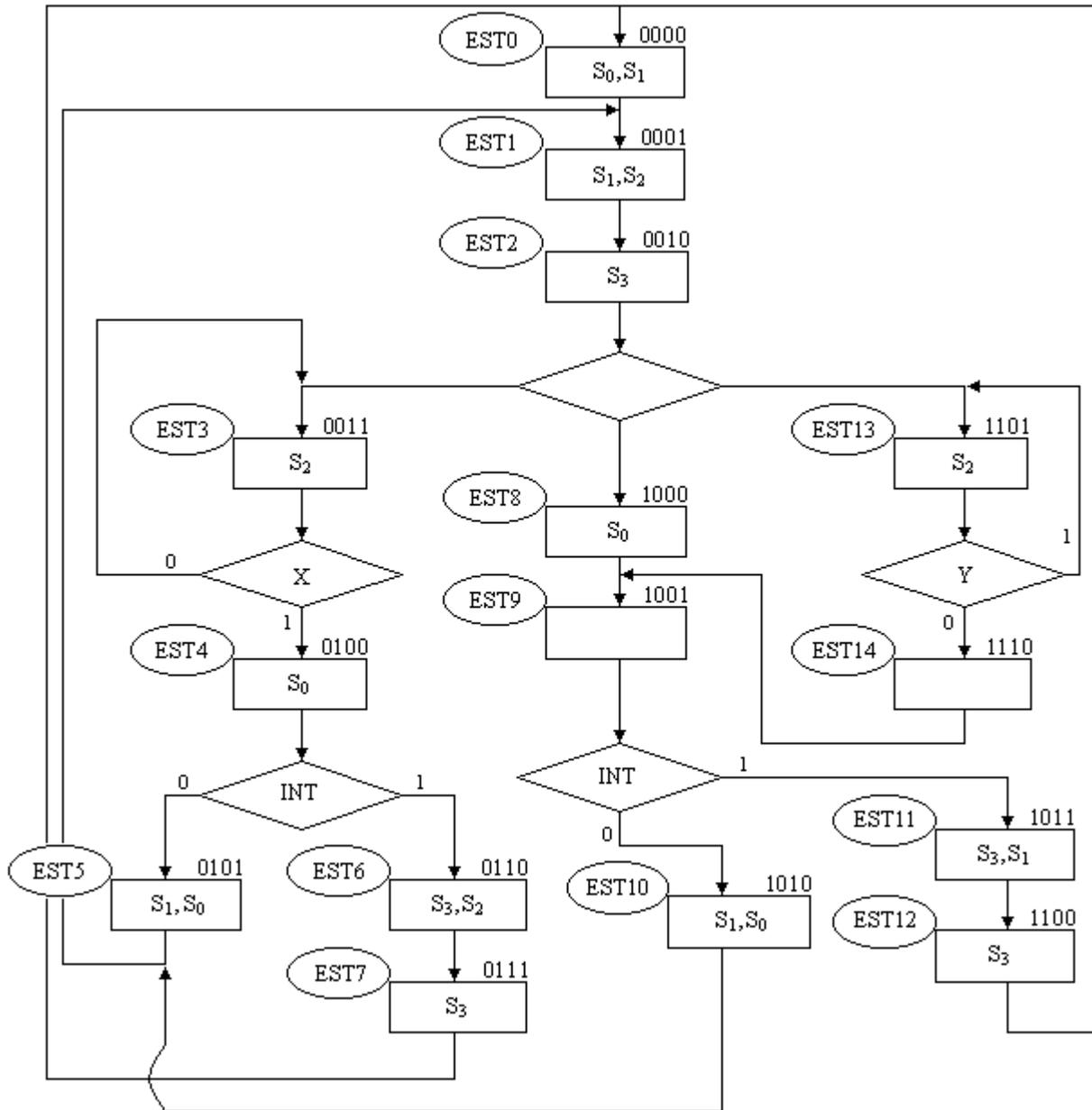


Figura 4.8. Carta ASM.

En el estado EST2 la dirección del estado siguiente está determinada por el contenido del registro de transformación, seleccionado cuando el secuenciador ejecuta la instrucción ST. En el estado

EST4, la dirección del estado siguiente la proporciona el registro de interrupciones o el registro  $\mu$ PC dependiendo del valor de la entrada INT.

La asignación binaria de las variables de entrada para la carta ASM es la siguiente:

$Q_x = 00$   
 $X = 01$   
 $Y = 10$   
 $INT = 11$

Con  $Q_x$  igual a '0' lógico, es decir,  $Q_x$  presenta un nivel lógico bajo.

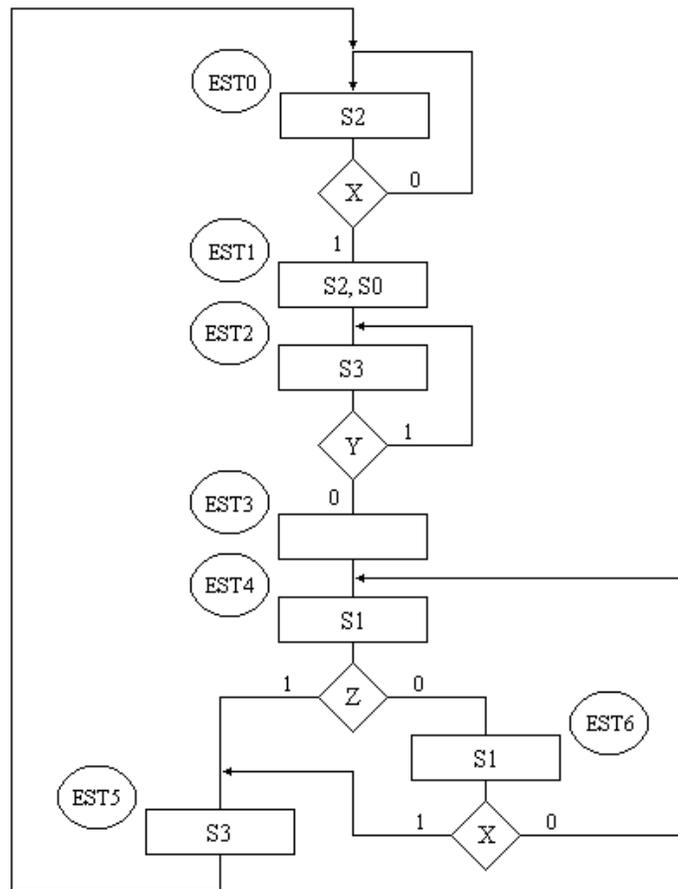
La tabla 4.3 muestra el contenido de la memoria para la carta ASM de la figura 4.8.

Dirección de la Memoria	Contenido de la Memoria					Mnemónico de la Instrucción
	Estado Presente	Liga	Micro Instrucción	Prueba	VF	
0000	0000	00	00	0	1 1 0 0	C
0001	0000	00	00	0	0 1 1 0	C
0010	0000	10	00	0	0 0 0 1	ST
0011	0011	01	01	0	0 0 1 0	SCC
0100	0000	11	11	1	1 0 0 0	SCI
0101	0001	01	00	0	1 1 0 0	SCC
0110	0000	00	00	0	0 0 1 1	C
0111	0000	01	00	0	0 0 0 1	SCC
1000	0000	00	00	0	1 0 0 0	C
1001	0000	11	11	1	0 0 0 0	SCI
1010	0001	01	00	0	1 1 0 0	SCC
1011	0000	00	00	0	0 1 0 1	C
1100	0000	01	00	0	0 0 0 1	SCC
1101	1101	01	10	1	0 0 1 0	SCC
1110	1001	01	00	0	0 0 0 0	SCC

Tabla 4.3. Contenido de la memoria para la carta ASM de la figura 4.8.

## PROBLEMAS

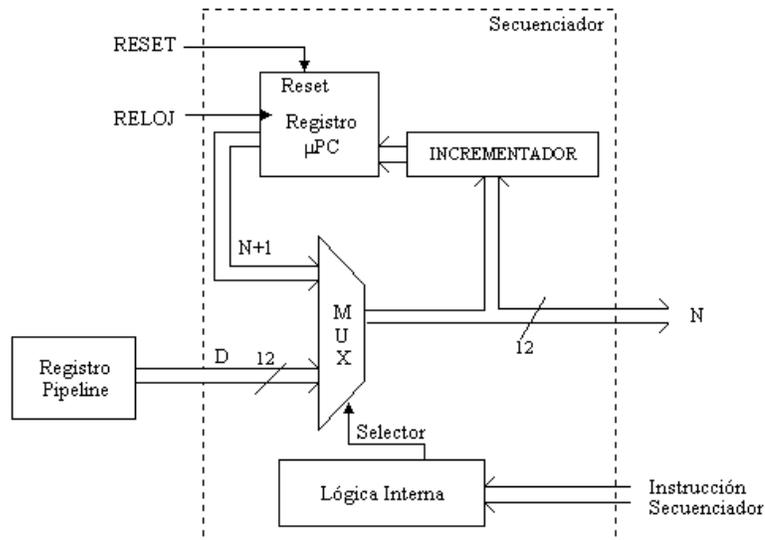
1. Diseñe la lógica interna del secuenciador mostrado en la figura 4.1.
2. Encuentre el contenido de la memoria para instrumentar la siguiente carta ASM. Utilice las instrucciones y el secuenciador de la figura 4.1.



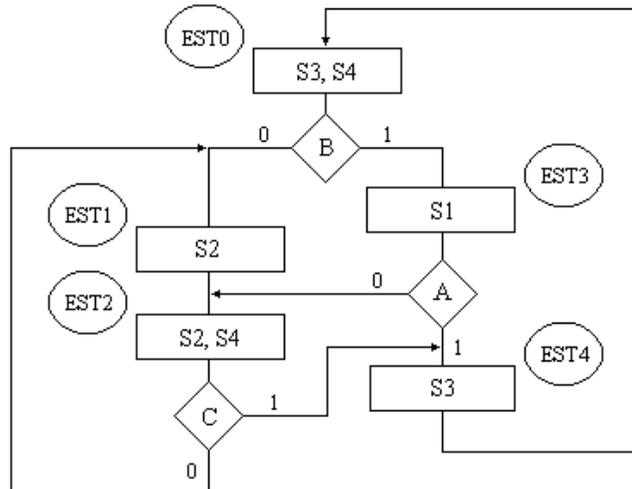
3. En la implantación de máquinas de estados usando memorias se utiliza una variante del direccionamiento implícito denominado secuenciador. La siguiente figura muestra un secuenciador capaz de ejecutar dos instrucciones: continúa y salto. Cuando se ejecuta la instrucción de continúa, el multiplexor selecciona el contenido del registro  $\mu$ PC con la

dirección del estado siguiente ( $N+1$ , donde  $N$  es el estado presente). Cuando se ejecuta la instrucción de salto, el multiplexor selecciona la dirección proveniente del registro Pipeline.

Nota: La señal 'Instrucción Secuenciador' le indica a la lógica interna qué instrucción estamos ejecutando, de esta manera, la lógica interna puede generar las señales de control adecuadas para el multiplexor.

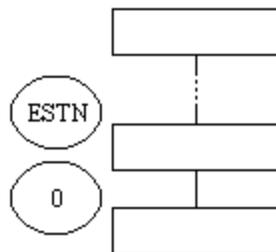


- a) Proponga una nueva arquitectura de secuenciador, con base en la figura anterior, que permita ejecutar, adicionalmente, las siguientes dos instrucciones: 1) una instrucción que guarde el contenido del registro  $\mu PC$  y haga un salto a la dirección contenida en el registro Pipeline (análogamente a un llamado a subrutina); y 2) una instrucción que recupere la dirección guardada por la instrucción anterior, la cual servirá como la dirección del estado siguiente (análogamente a un retorno de subrutina). Especifique el funcionamiento de la lógica interna para las dos nuevas instrucciones utilizando una tabla de verdad.
  - b) ¿Qué dispositivos se tendrían que utilizar para poder hacer este tipo de llamadas de forma anidada, es decir, dentro de una subrutina poder llamar a otra subrutina?
4. Codifique el secuenciador de la figura 4.1 utilizando el lenguaje Verilog HDL.
  5. Encuentre el contenido de la memoria para instrumentar la siguiente carta ASM. Utilice las instrucciones y el secuenciador de la figura 4.1.



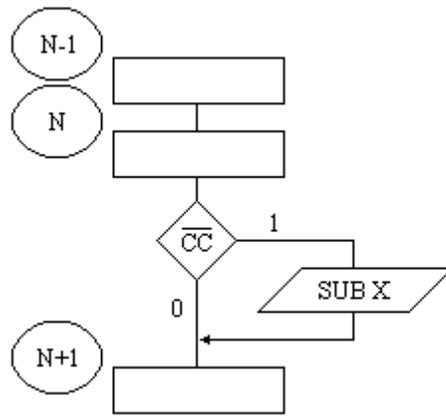
6. Modifique la estructura interna del secuenciador mostrado en la figura 4.1 para que pueda ejecutar, adicionalmente, las siguientes instrucciones.

a) Salto a cero (SC). En esta instrucción se hace un salto del estado N al estado cero, el cual tiene una representación binaria de ceros.

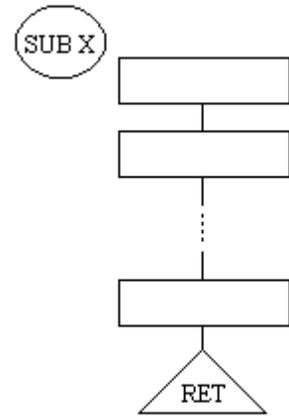


b) Salto condicional a subrutina (SCS). En esta instrucción, estando en el estado N, se pregunta por la variable de entrada CC. Si CC es igual a uno, la dirección del estado siguiente, procedente del registro de liga, ingresa a través de la entrada D del secuenciador. Esta dirección representa la dirección de inicio de una subrutina. La dirección de regreso de la subrutina, es decir, el estado N+1, debe ser guardado en una pila. Se podrán tener hasta 16 subrutinas anidadas. En caso de que CC valga cero, la dirección del estado siguiente estará dada por el  $\mu$ PC.

c) Regreso de subrutina (RS). La dirección de regreso de la subrutina, el estado N+1, es obtenido de la pila.



Salto condicional a subrutina



Regreso de subrutina