

# Práctica 4



## Memorias

### Objetivo

Diseñar y construir un circuito detector de secuencia usando una memoria para la función de transición y salida de la máquina de estados.

### Equipo

Computadora personal con el software Logisim.

### Fundamento teórico

#### Memorias no volátiles

Las memorias de solo lectura (Read-only memory, ROM) son un tipo de memoria semiconductora diseñadas para almacenar datos que son permanentes o que no van a cambiar frecuentemente. Durante la operación normal, no se escriben datos nuevos en la ROM, pero los datos que ya están almacenados pueden leerse de ella. En algunas ROMs los datos son ingresados cuando son manufacturadas; en otras los datos se pueden ingresar eléctricamente. El proceso de ingresar datos es llamado programar o quemar la ROM. En algunas ROMs no se pueden modificar los datos que ya fueron programados, otras pueden ser borradas y reprogramadas varias veces.

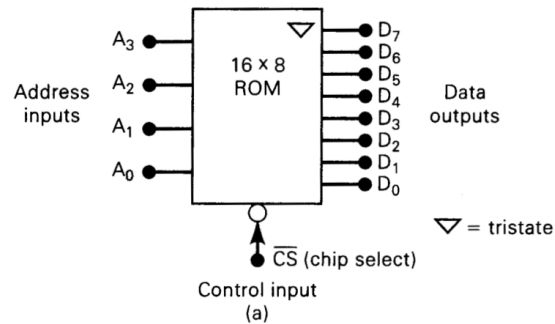
Las ROM son usadas para almacenar datos que no van a cambiar durante la operación normal del sistema. Un uso común de las ROM es para almacenar programas en las computadoras. Debido a que son no volátiles, los programas no se pierden cuando se quita la alimentación de energía a la computadora.

#### Arquitectura de una ROM

La Fig. 1 muestra el diagrama a bloques de una ROM. Se pueden distinguir tres conjuntos de señales: líneas de dirección, líneas de control y líneas de datos. Las líneas de dirección y de control son entradas y las líneas de datos son salidas en una ROM. En memorias que no son de solo lectura, las líneas de datos son bidireccionales.

- **Líneas de dirección:** Seleccionan la localidad o registro de la memoria a leer. La cantidad de localidades en una memoria está dada por la relación  $2^N$ , donde N es el número de líneas de dirección en el dispositivo. En la Fig. 1, las líneas de dirección son 4: A0 – A3, por lo tanto, la memoria tiene  $2^4 = 16$  localidades.
- **Líneas de datos:** Proveen los datos almacenados en una localidad de memoria. En la Fig. 1, las líneas de datos son D0 – D7, por lo tanto, cada localidad almacena 8 bits. La capacidad de la memoria es de  $16 \times 8$  bits.

- **Líneas de control:** Permiten habilitar la memoria, habilitar las salidas, seleccionar el modo de lectura (y/o de escritura si la memoria no es de solo lectura). En la Fig. 1, la línea de control es CS (Chip Select), la cual se activa en bajo. La memoria solo provee las salidas cuando está línea está en bajo. Cuando se encuentra en alto, las líneas de datos están en alta impedancia, Hi-Z.



Word	Address				Data							
	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
0	0	0	0	0	1	1	0	1	1	1	1	0
1	0	0	0	1	0	0	1	1	1	0	1	0
2	0	0	1	0	1	0	0	0	0	1	0	1
3	0	0	1	1	1	0	1	0	1	1	1	1
4	0	1	0	0	0	0	0	1	1	0	0	1
5	0	1	0	1	0	1	1	1	1	0	1	1
6	0	1	1	0	0	0	0	0	0	0	0	0
7	0	1	1	1	1	1	1	0	1	1	0	1
8	1	0	0	0	0	0	1	1	1	1	0	0
9	1	0	0	1	1	1	1	1	1	1	1	1
10	1	0	1	0	1	0	1	1	1	0	0	0
11	1	0	1	1	1	1	0	0	0	1	1	1
12	1	1	0	0	0	0	1	0	0	1	1	1
13	1	1	0	1	0	1	1	0	1	0	1	0
14	1	1	1	0	1	1	0	1	0	0	1	0
15	1	1	1	1	0	1	0	1	1	0	1	1

(b)

Word	Address				Data							
	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	D <sub>7</sub> -D <sub>0</sub>							
0	0				DE							
1	1				3A							
2	2				85							
3	3				AF							
4	4				19							
5	5				7B							
6	6				00							
7	7				ED							
8	8				3C							
9	9				FF							
10	A				B8							
11	B				C7							
12	C				27							
13	D				6A							
14	E				D2							
15	F				5B							

(c)

Figura 1. a) Diagrama a bloques de una ROM; b) tabla mostrando los datos en cada dirección de la ROM en formato binario; c) similar a la tabla anterior pero los datos en formato hexadecimal.

### Procedimiento de lectura:

1. Colocar la dirección a leer en las líneas de dirección A0 – A3.
2. Colocar la señal de habilitación del dispositivo, línea CS en bajo.
3. La memoria provee en las líneas de datos D0 – D7 el dato que está almacenado en la localidad seleccionada en el paso 1.

### Tiempo de respuesta:

Al hacer una lectura de una ROM, existe un retardo de propagación de señales entre la aplicación de las entradas a la ROM para la lectura y la salida de los datos. Este retardo, llamado tiempo de acceso,  $t_{ACC}$ , es una medición de la velocidad de operación de la

ROM. La Fig. 2 muestra el tiempo de acceso. Justo antes de  $t_1$  las direcciones están cambiando para una lectura en una nueva dirección. En  $t_1$  la dirección es válida, esto es, cada línea de dirección está en un nivel lógico válido. En este punto, la ROM empieza a decodificar la dirección para seleccionar la localidad que va a enviar sus datos a los buffers de salida. En  $t_2$ ,  $\overline{CS}$  es activado para habilitar los buffers de salida. Finalmente, en  $t_3$ , las salidas cambian de un estado de alta impedancia a los niveles que representan los datos almacenados en la localidad específica.

El tiempo entre  $t_1$  y  $t_3$  es el tiempo de acceso  $t_{ACC}$ . ROMs bipolares típicas tienen un  $t_{ACC}$  entre 30 a 90 ns, dispositivos NMOS de 35 a 500 ns, tecnologías CMOS tienen un rango de 20 a 60 ns.

Otro parámetro de tiempo importante es el tiempo en la habilitación de la salida,  $t_{OE}$ , el cual es el retardo entre la activación de  $\overline{CS}$  y la salida de datos válidos. Valores típicos para tecnologías bipolares son de 10 a 20 ns, 25 a 100 ns para NMOS y 12 a 50 ns para CMOS.

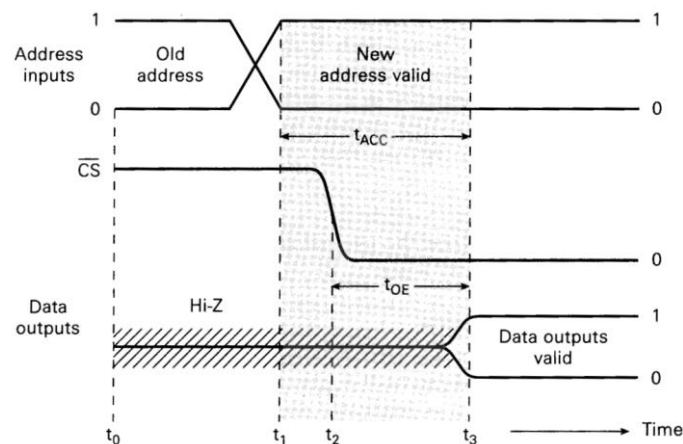


Figura 2. Tiempos típicos de una ROM en una lectura.

### Tipos de ROM:

- **Máscara ROM:** Las localidades de memoria son escritas en tiempo de manufactura. Una máscara especial es usada para controlar las interconexiones eléctricas en el chip. Sus datos no pueden ser reprogramados. La Fig. 3 muestra la estructura de una memoria Máscara ROM.

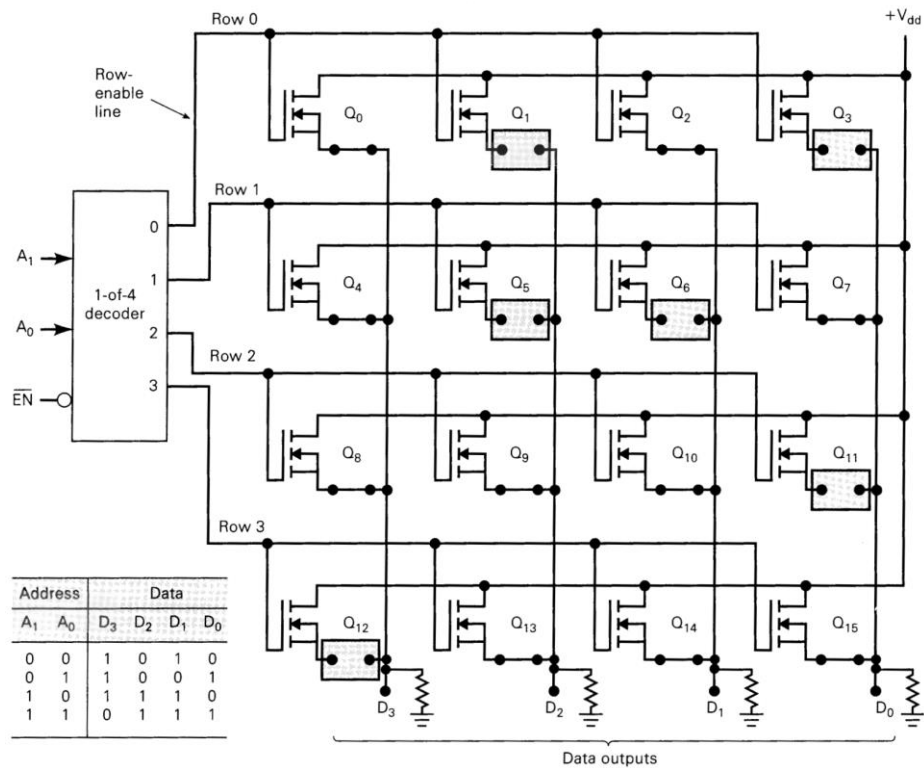


Figura 3. Estructura de una Máscara ROM. Un MOSFET es usado por cada celda de memoria. Una conexión abierta almacena un 0, una conexión cerrada almacena un 1.

- **PROM (Programmable ROM, ROM Programable):** No son programadas durante la manufactura. El usuario puede programarlas una sola vez, ya programadas no pueden ser borradas.
- **EPROM (Erasable Programmable ROM, ROM Borrable y Programable):** Puede ser programada por el usuario y también puede ser borrada y reprogramada cuantas veces sea requerido. El proceso de programado involucra aplicar niveles especiales de voltaje (típicamente en el rango de 10 a 25 V), a algunos pines del chip por cierto tiempo (típicamente 50 ms por localidad de memoria). El programado usualmente se realiza con un circuito de programación especial separado del sistema en el cual la EPROM va a operar.

En estado normal, todos los transistores en las celdas de la EPROM están apagados y almacenan un 1 lógico. Los transistores pueden ser encendidos al aplicar un alto voltaje de programación para que la celda almacene un 0 lógico. Durante el proceso de programado, las direcciones de la EPROM y las líneas de datos son usadas para seleccionar que celdas van a ser programadas con ceros y cuáles van a permanecer con unos.

Una vez que una EPROM ha sido programada, puede ser borrada al exponerla a luz ultravioleta (UV) aplicada a través de una ventana en el empaquetado del chip, por entre 15 a 20 minutos. La luz UV apaga todos los transistores, de forma que todas las celdas

almacenan 1. Una vez borrada la memoria, ésta puede ser reprogramada. La Fig. 5 muestra un ejemplo de una EPROM y la Fig. 6 de un borrador para EPROMs por luz UV.

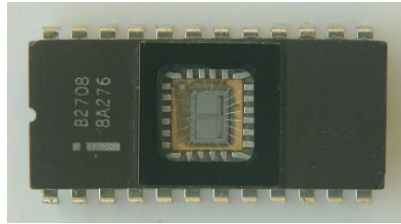


Figura 5. EPROM.



Figura 6. Luz UV para borrado de EPROMs.

- **EEPROM (Electrically Erasable Programmable ROM, ROM Eléctricamente Borrable y Programable):** Es una mejora de las EPROM ya que permite borrar el contenido de la memoria eléctricamente en lugar de con luz UV. La operación de borrar y almacenar nuevos datos usualmente toma 5 ms. Debido a la complejidad de sus celdas y circuitería de soporte, las memorias EEPROM suelen ser de menor capacidad y mayor costo que las EPROM.
- **Memoria Flash:** El reto de los ingenieros de semiconductores era fabricar una memoria con las características de borrado eléctrico de las EEPROM pero con las capacidades de memoria y costos de las EPROM, manteniendo la alta velocidad de lectura de ambas. Esto se logró con las memorias flash. El nombre de las memorias hace referencia a la rapidez con que son borradas y escritas. Muchas poseen un modo de borrado masivo donde toda la memoria es borrada, otras ofrecen un modo de borrado por sector donde solo sectores específicos de la memoria son borrados. Esto evita tener que borrar toda la memoria si solo una porción va a ser actualizada.

La Fig. 6 muestra una comparación de las memorias no volátiles respecto a cómo la flexibilidad de borrado y programado incrementan la complejidad y costo del dispositivo.

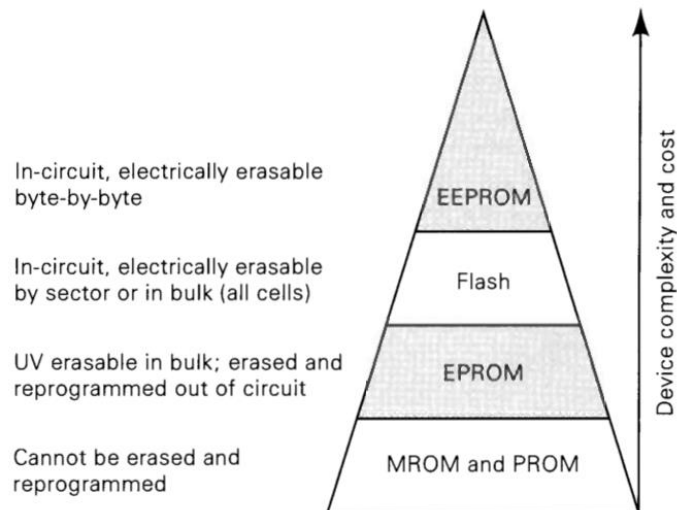
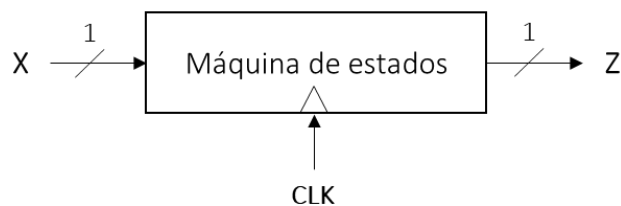


Figura 6. Comparativa de distintas memorias no volátiles respecto a su complejidad y costo.

## Desarrollo

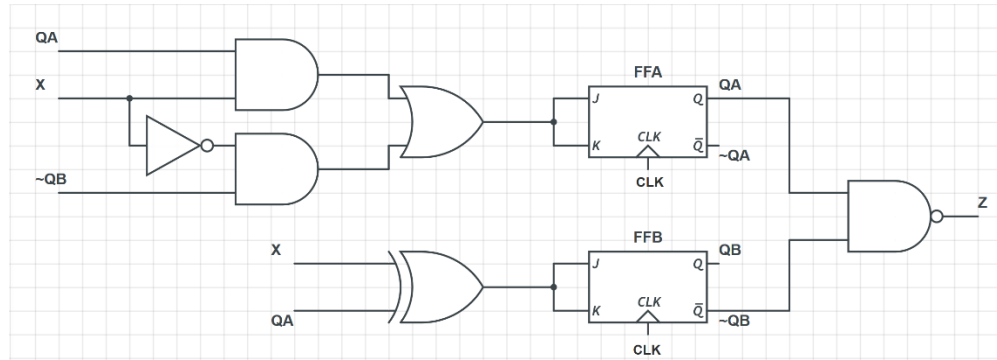
1. Diseñe y simule un circuito detector de secuencia con una entrada **X** y una salida **Z**, que detecte la aparición de la secuencia 11001 en la entrada. La salida **Z** es 1 cada vez que la secuencia es recibida. El detector debe ser modelo Moore y con traslape. Haga uso de una memoria ROM para la función de transición y la salida de la máquina de estados. Utilice flip flops D como elemento de memoria.



Procedimiento:

1. Realice el diagrama de estados y la tabla de transición de estados.
2. Si aplica, reduzca la cantidad de estados al eliminar estados redundantes.
3. Construya la tabla de excitación para las señales de entrada de los flip-flops D.
4. Almacene en la ROM los valores a alimentar a los flip flops para el cambio de estado y los estados de la salida **Z**.
5. Simule en Logisim su solución.
6. Incluya en el reporte el diagrama de tiempos del circuito secuencial que contenga la señal de reloj, señal de entrada **X**, los cambios de estado y la salida **Z**.

2. Analice el siguiente circuito y derive:
  - a) Ecuaciones lógicas para JA y KA (flip flop FFA), JB y KB (flip flop FFB) y Z.
  - b) Tabla de excitación y tabla de transición de estados.
  - c) Diagrama de estados.



3. Realice un video donde describa **detalladamente** su detector de secuencia en Logisim, muestre la ejecución del circuito y describa el valor de entrada que le esté activando, los estados que va recorriendo el circuito y la salida obtenida. Posteriormente, describa **todo** el procedimiento que siguió para obtener el diagrama de estados del ejercicio 2.
4. En su reporte incluya:
  - Diagramas de estados.
  - Tablas de transición.
  - Tablas de excitación.
  - Ecuaciones lógicas.
  - Circuito en Logisim.
5. Al entregar su práctica, adjunte el archivo de Logisim con el circuito simulado.

## Conclusiones y comentarios

## Dificultades en el desarrollo

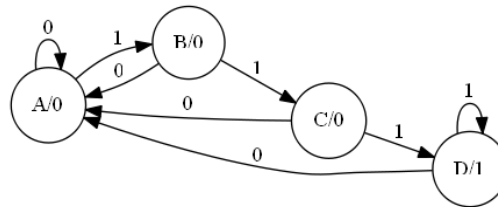
## Referencias

# Apéndice

## Ejemplo de un detector de secuencia implementado con una memoria EPROM

Diseñe un detector de secuencia con una entrada **X** y una salida **Z** que detecte una secuencia con tres o más unos consecutivos. La salida **Z** es 1 cada vez que este tipo de secuencia es encontrada. El detector debe ser con traslape, donde los últimos dos unos de una secuencia aceptada son considerados como los primeros de una secuencia posterior.

Diagrama de estados



Estado	q1q0
A	00
B	01
C	11
D	10

Tabla 1. Asignación de estados.

Estado actual		Entrada	Siguiete estado		Salida
q1	q0		q1+	q0+	
0	0	0	0	0	0
0	1	0	0	0	0
1	0	0	0	0	1
1	1	0	0	0	0
0	0	1	0	1	0
0	1	1	1	1	0
1	0	1	1	0	1
1	1	1	1	0	0

Tabla 2. Transición de estados.



Estado actual		Entrada	Siguiete estado		Entradas de flip-flops		Salida
q1	q0	X	q1+	q0+	d1	d0	Z
0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0
1	0	0	0	0	0	0	1
1	1	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	1	1	1	1	1	1	0
1	0	1	1	0	1	0	1
1	1	1	1	0	1	0	0

Tabla 3. Tabla de excitación para flip-flops D.

## Implementación usando una memoria EPROM

En la memoria EPROM se almacenan los valores a colocar en las entradas de los flip flops para el nuevo estado (d1 y d0), así como la salida **Z**. Las direcciones donde se almacenan estos valores son determinadas por el estado actual (q1 y q0) y la entrada **X**.

### - Entradas de la EPROM:

Las líneas q1, q0 y **X** se conectan a las líneas de dirección de la EPROM.

### - Salidas de la EPROM:

Dos líneas de datos de la EPROM se conectan a las entradas d1 y d0 de los flip flops. Una línea de datos de la EPROM se conecta a un led para visualizar la salida **Z**.

### - Contenido de la EPROM:

Líneas de dirección de la EPROM				Líneas de datos de la EPROM			
Ax - A3	A2	A1	A0	IO7 - IO3	IO2	IO1	IO0
	q1	q0	X		d1	d0	Z
0	0	0	0	X	0	0	0
0	0	1	0	X	0	0	0
0	1	0	0	X	0	0	1
0	1	1	0	X	0	0	0
0	0	0	1	X	0	1	0
0	0	1	1	X	1	1	0
0	1	0	1	X	1	0	1
0	1	1	1	X	1	0	0

Tabla 4. Datos almacenados en la EPROM correspondientes a la función de transición y la salida.

## Diagrama de conexión

