

## TEMA 7: SISTEMAS SECUENCIALES SÍNCRONOS.

### 7.1. Autómata de Estados Finitos.

Por definición, un Autómata de Estados Finitos es una quintupla compuesta por 5 elementos:

$$A = [E, Z, Q, \alpha, \delta] \quad \alpha: Q \times E \rightarrow Z \quad \delta: Q \times E \rightarrow Q$$

donde E es un conjunto de entradas, Z es un conjunto de salidas, Q es un conjunto de estados,  $\alpha$  es una función de salida y  $\delta$  es una función de transición de estados.

Un sistema secuencial es aquél cuyas salidas dependen de las entradas actuales (como los combinacionales), y, además, de las entradas anteriores. Las entradas anteriores crean una situación que se llama **estado**, con lo que un circuito secuencial será tal que:

- 1) Sus salidas dependen de sus entradas y sus estados.
- 2) Los estados, a su vez dependen de las entradas y de los estados anteriores.

Estado: situación en la que está una cosa, en relación con los cambios que influyen en ella.

### 7.2. Diagramas de estado.

Son unos diagramas que representan el cambio de estado en función del estado previo y de las entradas.

La Figura representa el diagrama de transición de estados y salidas, independientemente de cómo se codifiquen. En realidad, representa el comportamiento de un autómata finito.

$$E=\{0,1\} \quad Z=\{0,1\} \quad Q=\{q_0, q_1, q_2, q_3\} \quad E/S$$

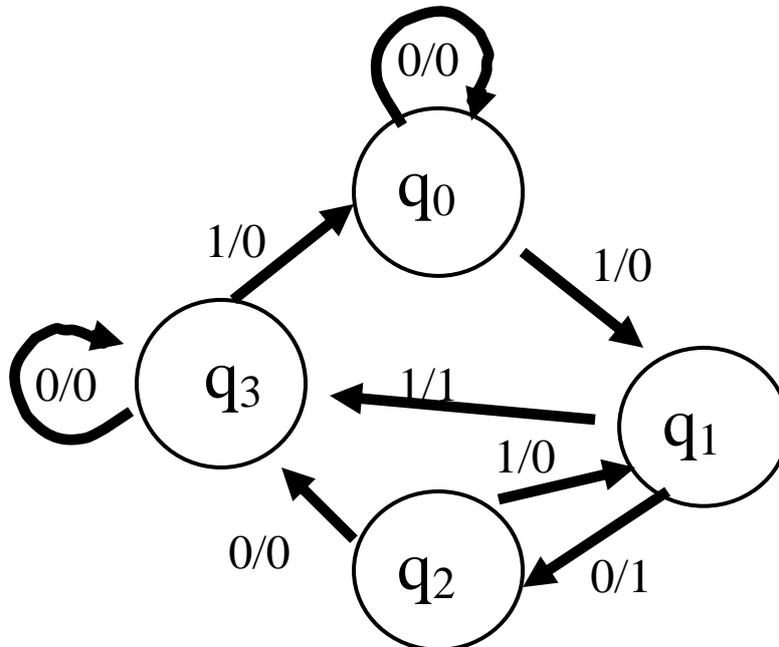
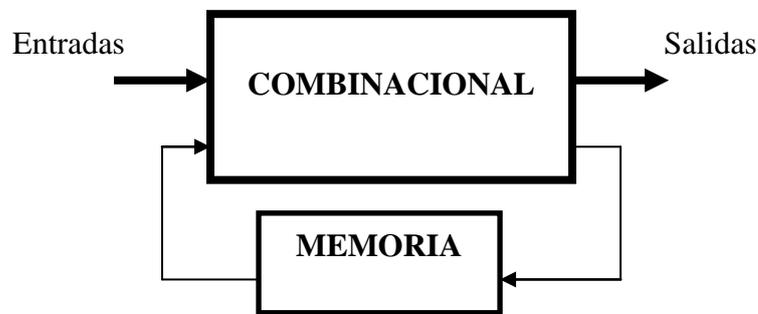


Tabla de transiciones de estados.

ESTADO ACTUAL: Q(t)	ENTRADA	PRÓXIMO ESTADO: Q(t+1)	SALIDA
q <sub>0</sub>	0	q <sub>0</sub>	0
q <sub>0</sub>	1	q <sub>1</sub>	0
q <sub>1</sub>	0	q <sub>2</sub>	1
q <sub>1</sub>	1	q <sub>3</sub>	1
q <sub>2</sub>	0	q <sub>3</sub>	0
q <sub>2</sub>	1	q <sub>1</sub>	0
q <sub>3</sub>	0	q <sub>3</sub>	0
q <sub>3</sub>	1	q <sub>0</sub>	0

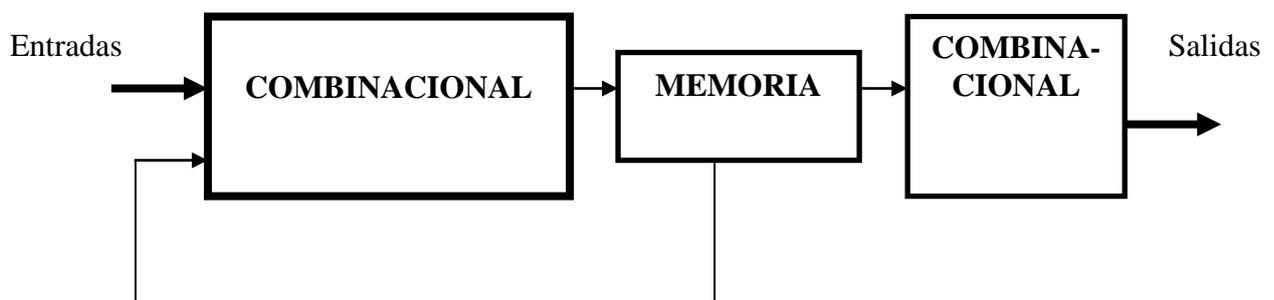
### 7.3. Autómatas de Moore y Mealy.

En el diagrama anterior, cada par (entrada, estado) puede generar cualquier salida y cualquier transición a un estado determinado. A este tipo de autómatas se le llama **Autómata de Mealy**. Las salidas están asociadas con las transiciones.



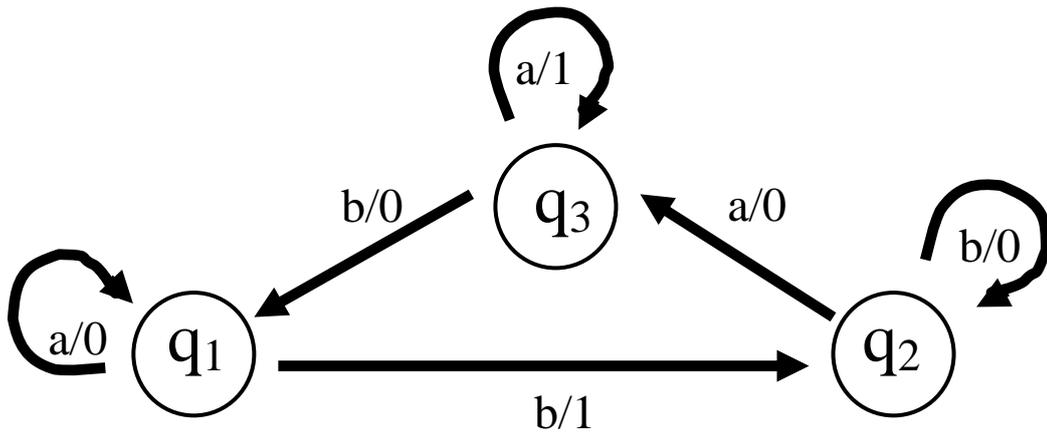
Autómata de Mealy

Por el contrario, hay autómatas en los que a cada estado le corresponde de forma inequívoca una salida determinada, es decir, estar en el **estado q<sub>i</sub>** supone que la salida ha sido una cierta **S<sub>i</sub>**. No puede ocurrir que alcanzando el **estado q<sub>i</sub>** la salida haya sido distinta de S<sub>i</sub>. A este autómata que tiene asociada una salida a un estado se le llama **Autómata de Moore**. En los autómatas de Moore no es necesario escribir la salida en la flecha de la transición, puesto que la misma será la que tenga asociada el estado al que se llega.



Autómata de Moore

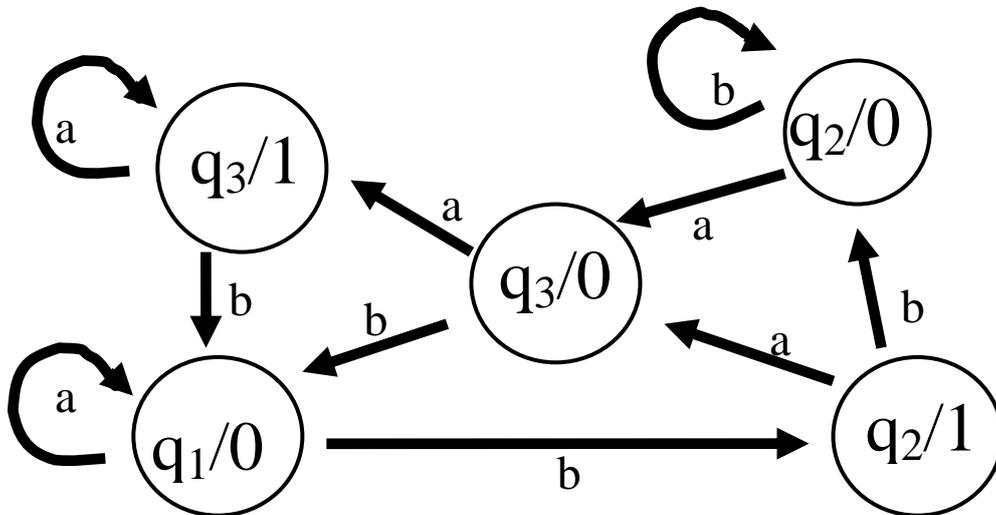
Dado un autómata de Mealy siempre se puede encontrar un autómata de Moore equivalente. Veamos un ejemplo:



Autómata de Mealy. Las salidas están asociadas a las transiciones.

Al estado  $q_1$  se llega desde el estado  $q_3$  con entrada  $b$  o desde el propio  $q_1$  con entrada  $a$ , pero siempre dando **salida 0**. Podemos asociar al estado  $q_1$  la **salida 0**, es decir, si el estado es  $q_1$  la salida ha sido **0**. No ocurre lo mismo con  $q_2$ . Al estado  $q_2$  se llega desde el estado  $q_1$  con entrada  $b$  dando **salida 1** o bien desde el propio  $q_2$  con entrada  $b$  dando **salida 0**. El hecho de estar en  $q_2$  no presupone conocer cuál ha sido la salida. Lo mismo ocurre con  $q_3$ .

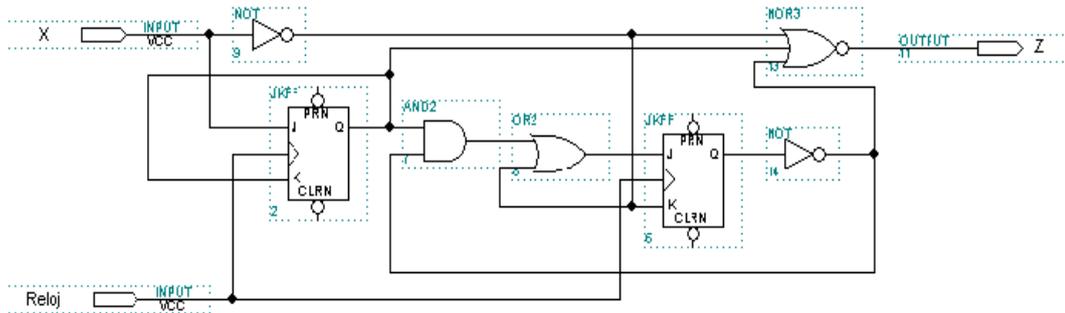
La forma de convertir un autómata de Mealy en uno de Moore es desdoblar aquellos estados que no tienen asociada una salida en tantos estados como salidas distintas tengan y de esta forma cada uno de los nuevos estados tiene asociada de forma inequívoca una salida. Para nuestro ejemplo quedaría:



Autómata de Moore. Las salidas están asociadas a los estados.

### 7.4. Análisis de Sistemas Secuenciales Síncronos.

Consiste en ver cómo responde en función de sus entradas y estados actuales. Ejemplo.



La salida  $z(t)$  se puede expresar como

$$Z(t) = \overline{\overline{X} + Q_2(t) + Q_1(t)}$$

Las entradas de los biestables se pueden poner como:

$$\begin{aligned} J_2 &= X & K_2 &= Q_2(t) \\ J_1 &= \overline{X} + Q_2(t) \cdot \overline{Q_1(t)} & K_1 &= \overline{X} \end{aligned}$$

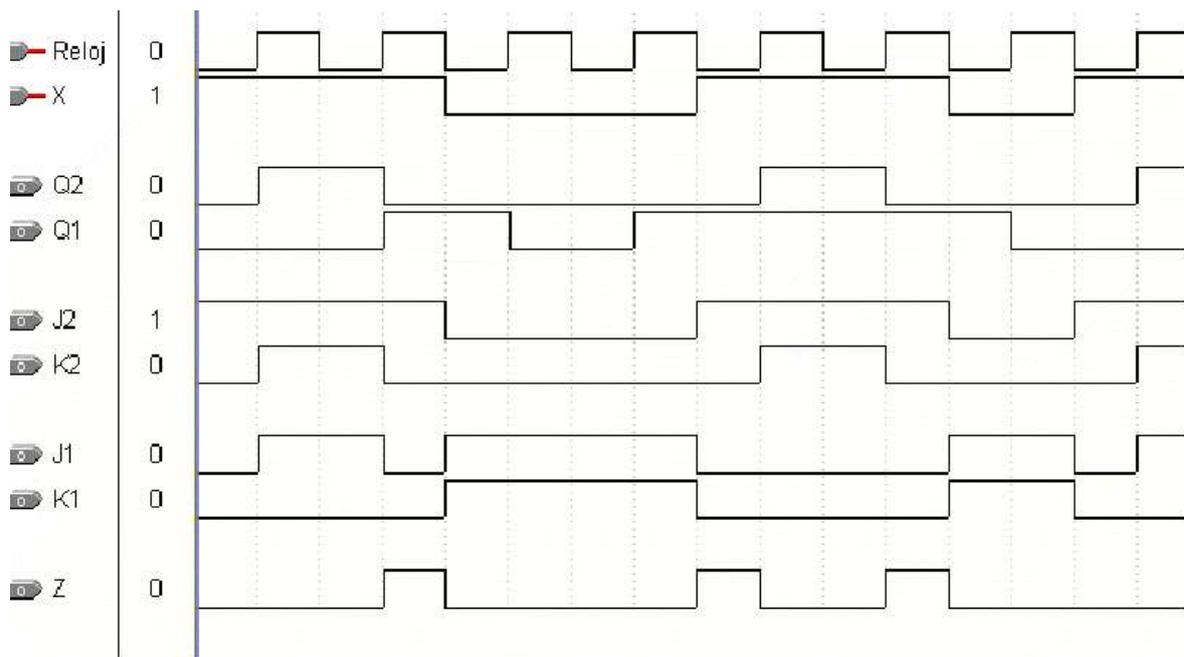
Los valores de Próximo estado y Salida se pueden obtener aplicando las ecuaciones booleanas o bien mirando para cada caso qué saldrá del biestable en función de las entradas actuales y del estado actual, para lo cual hay que conocer cómo funciona el biestable.

Entrada X	Estado actual		Biestable 2		Biestable 1		Próximo estado		Salida Z
	$Q_2(t)$	$Q_1(t)$	$J_2$	$K_2$	$J_1$	$K_1$	$Q_2(t+1)$	$Q_1(t+1)$	
0	0	0	0	0	1	1	0	1	0
0	0	1	0	0	1	1	0	0	0
0	1	0	0	1	1	1	0	1	0
0	1	1	0	1	1	1	0	0	0
1	0	0	1	0	0	0	1	0	0
1	0	1	1	0	0	0	1	1	1
1	1	0	1	1	1	0	0	1	0
1	1	1	1	1	0	0	0	1	0

Veamos un ejemplo de cronograma suponiendo que inicialmente los dos biestables están a 0 y conociendo la evolución temporal o cronograma completo de la señal de entrada X. Se ha supuesto que la entrada X está sincronizada con el reloj, pero con un desfase de medio período para que en el instante del flanco del reloj, la señal de entrada esté estabilizada. Para obtener el cronograma se procede de la siguiente manera:

1º) Conocida la entrada y el valor de los biestables (el primero de ellos debe ser conocido de antemano como dato inicial), se determinan los valores de las entradas a los biestables y la salida.

2º) Conocidas las entradas a los biestables en el momento del flanco de subida, se calculan los próximos estados de todos los biestables, que junto con la próxima entrada permite repetir de forma cíclica el punto anterior.



Como se observa en la Figura, en un autómata de Mealy las salidas pueden cambiar antes de que llegue el flanco activo del reloj si cambian las entradas, ya que la salida depende de la entrada directamente. Además, las salidas y los estados cambian o pueden cambiar en el flanco activo del reloj. Por el contrario, en un autómata de Moore, tanto los estados como las salidas sólo cambian cuando llega el flanco correspondiente del reloj.

## 7.5. Síntesis o Diseño de Sistemas Secuenciales Síncronos.

*Si no se dice lo contrario, supondremos un autómata de Mealy con biestables síncronos por flanco.*

### 1º. Paso de las especificaciones verbales a diagrama de estados.

Lo primero que hay que hacer es "traducir" el enunciado a diagrama de estados, es decir, distinguir estados, numerarlos, identificar entradas, determinar salidas, y en definitiva expresar el diagrama de estado de forma gráfica.

### 2º. Construcción de la tabla de estados y de excitación.

Se coloca en una tabla la entrada, el estado inicial, el próximo estado y la salida, y se pone la tabla de excitación del biestable que resuelva el problema.

### 3º. Minimización de las funciones de excitación.

Utilizando las técnicas habituales: mapas de Karnaugh, módulos combinacionales, etc...

### 4º. Implementación del circuito.

Se construye el circuito.

## 7.6. Ejemplos de diseño de sistemas secuenciales síncronos.

### 1º. Diseño de un contador síncrono de 2 bits, ascendente.

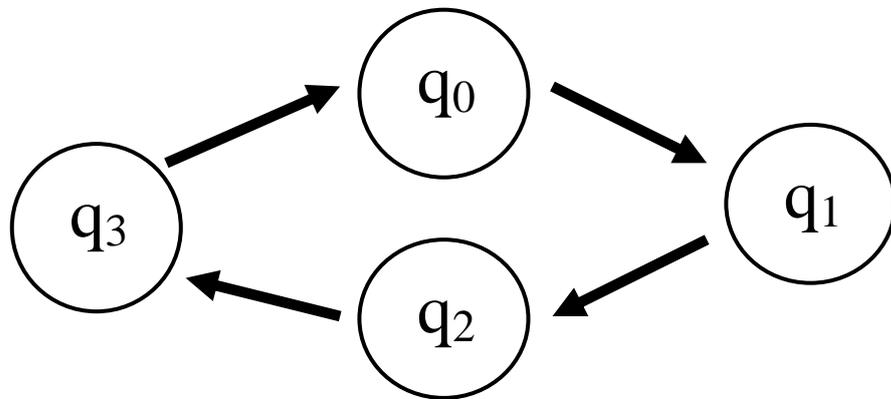
Autómata de Moore. Las salidas del circuito coinciden con las salidas de los biestables.

#### 1º. Paso de las especificaciones verbales a diagrama de estados.

Definición de Estados.

$q_0$ : Cuenta 0.    $q_1$ : Cuenta 1.    $q_2$ : Cuenta 2.    $q_3$ : Cuenta 3.

Diagrama de estados



Codificación de estados

	$Q_1$	$Q_0$
$q_0$	0	0
$q_1$	0	1
$q_2$	1	0
$q_3$	1	1

#### 2º. Construcción de la tabla de estados y de excitación.

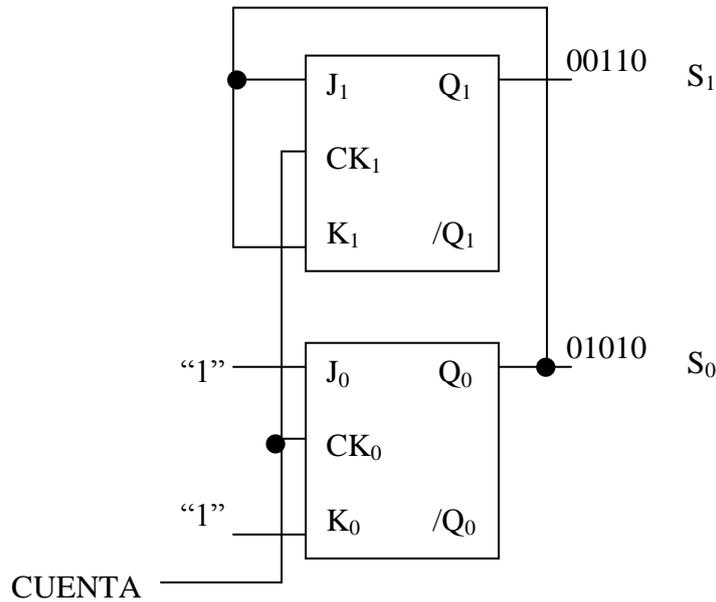
ESTADO ACTUAL		ESTADO PRÓXIMO		BIESTABLE 1		BIESTABLE 0	
$Q_1(t)$	$Q_0(t)$	$Q_1(t+1)$	$Q_0(t+1)$	$J_1$	$K_1$	$J_0$	$K_0$
0	0	0	1	0	X	1	X
0	1	1	0	1	X	X	1
1	0	1	1	X	0	1	X
0	1	0	0	X	1	X	1

#### 3º. Minimización de las funciones de excitación.

Haciendo los MK sale:

$$J_1 = Q_0 \quad J_0 = 1 \quad K_1 = Q_0 \quad K_0 = 1$$

**4º. Implementación del circuito.**



**2º. Sumador serie.**

Recibe los dos sumandos en serie, de menor a mayor peso, bit a bit en cada período de reloj. Al mismo tiempo genera las sumas parciales y los acarrees. La necesidad de memoria está en recordar si para la etapa siguiente hay o no acarreo. Lo mismo que cuando sumamos mentalmente recordamos si "nos llevamos una" o no, el estado que hay que recordar es si hay o no acarreo.

Por tanto en circuito secuencial que implementa un sumador de 4 bits debe tener una tabla de verdad del siguiente tipo:

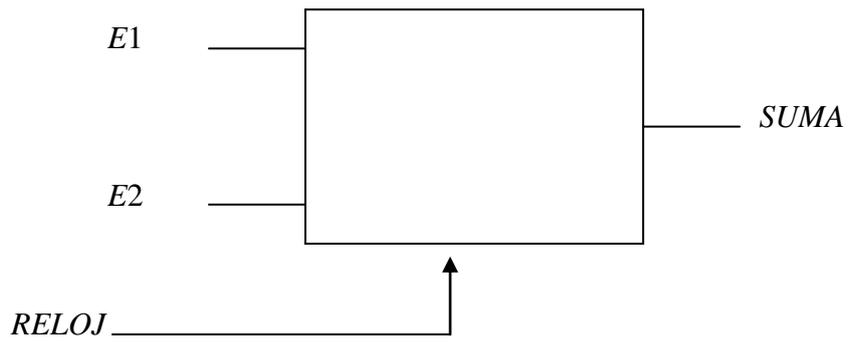
ENTRADAS Y ESTADO ACTUAL			SALIDA Y ESTADO PRÓXIMO		EXCITACIÓN BIESTABLE	
$E_1$	$E_2$	$Q_0(t)$	SUMA	$Q_0(t+1)$	$J_0$	$K_0$
0	0	0	0	0	0	X
0	0	1	1	0	X	1
0	1	0	1	0	0	X
0	1	1	0	1	X	0
1	0	0	1	0	0	X
1	0	1	0	1	X	0
1	1	0	0	1	1	X
1	1	1	1	1	X	0

El circuito que resuelve esta situación será el siguiente, cuyas ecuaciones son:

$$J_0 = E_1 \cdot E_2 \quad K_0 = \overline{E_1 \cdot E_2} = \overline{E_1 + E_2}$$

$$Q_0(t+1) = E_1 \cdot E_2 + Q_0(t) \cdot (E_1 + E_2) = J_0 + Q_0(t) \cdot \overline{K_0}$$

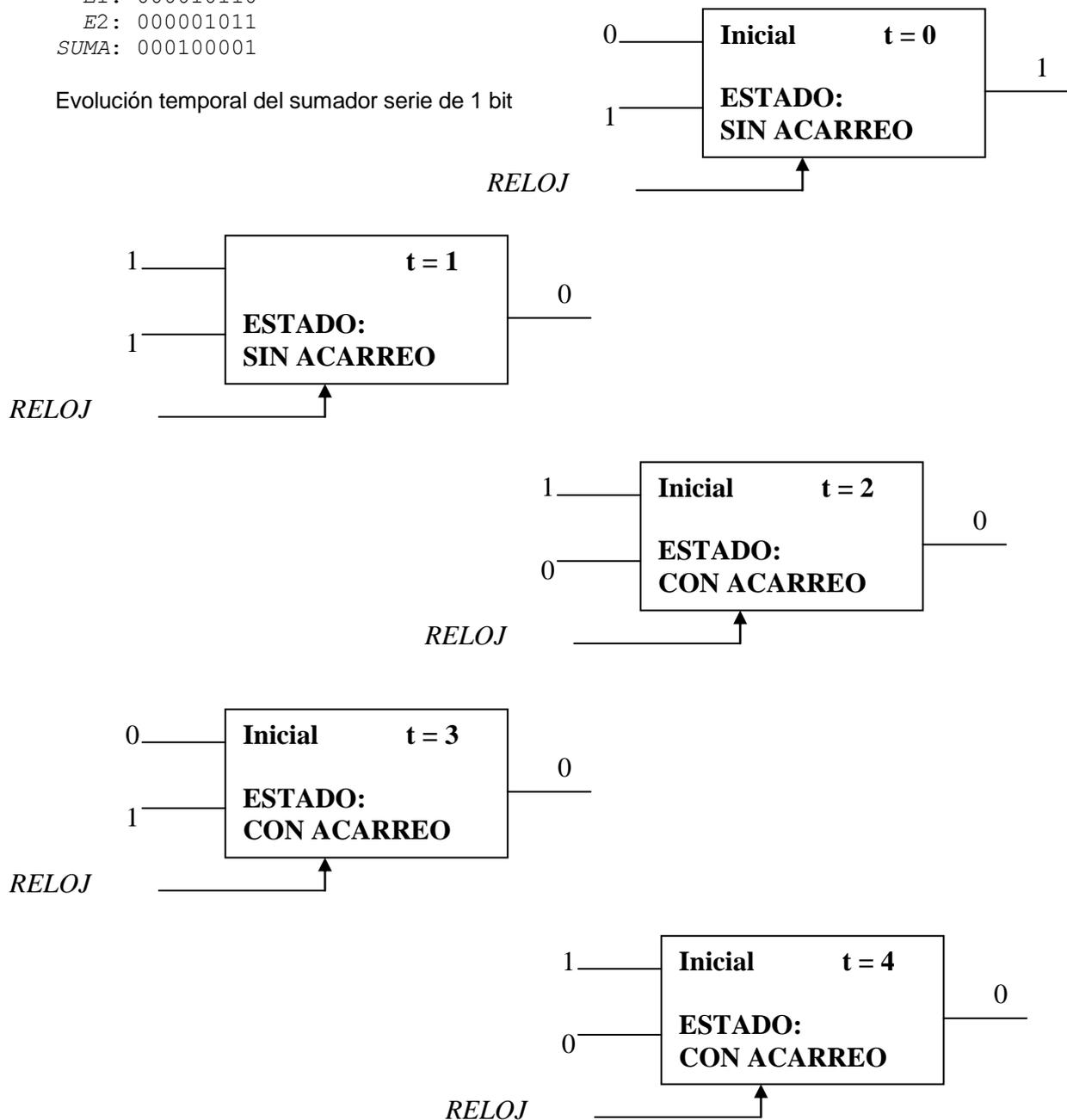
$$SUMA = Q_0(t) \oplus E_1 \oplus E_2$$



Ejemplo: sumar 22 + 11

E1: 000010110  
E2: 000001011  
SUMA: 000100001

Evolución temporal del sumador serie de 1 bit



### 3º. Reconocedores de secuencias.

Supongamos que se quiere detectar la subsecuencia 1101 dentro de una secuencia de bits de longitud indefinida.

Vamos a hacerlo usando el diseño de circuitos secuenciales.

#### Definición de estados

$q_0$ : estado inicial. Resto de estados.

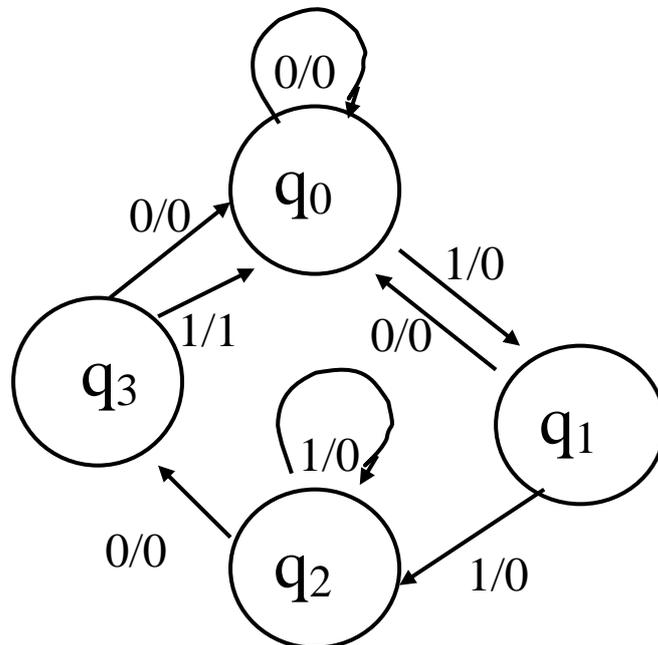
$q_1$ : se ha recibido un 1 en el último bit.

$q_2$ : se han recibidos dos 1's en los dos últimos bits.

$q_3$ : se ha recibido un 0 en el último bit después de que se hayan recibidos dos 1's en los dos anteriores bits.

#### Codificación

	$Q_1$	$Q_0$
$q_0$ :	0	0
$q_1$ :	0	1
$q_2$ :	1	0
$q_3$ :	1	1



#### Tablas de transición y excitación

$E$	$Q_1(t)$	$Q_0(t)$	$Q_1(t+1)$	$Q_0(t+1)$	$J_1$	$K_1$	$J_0$	$K_0$	$S$
0	0	0	0	0	0	X	0	X	0
0	0	1	0	0	0	X	X	1	0
0	1	0	1	1	X	0	1	X	0
0	1	1	0	0	X	1	X	1	0
1	0	0	0	1	0	X	1	X	0
1	0	1	1	0	1	X	X	1	0
1	1	0	1	0	X	0	0	X	0
1	1	1	0	0	X	1	X	1	1

#### Simplificación

$$J_1 = E \cdot Q_0$$

$$J_0 = E \oplus Q_1$$

$$K_1 = Q_0$$

$$K_0 = 1$$

$$S = E \cdot Q_1 \cdot Q_0$$

#### 4º. Generador de paridad par de 3 bits.

Se trata de diseñar un generador secuencial de paridad par de cadenas de 3 bits serie. Por ejemplo, si la secuencia de entrada es  $X = 000110101\dots$  la salida deberá ser  $Z = 0100010\dots$  Utilizar para ello biestables tipo D.

##### 1. Paso de las especificaciones verbales al diagrama de estados

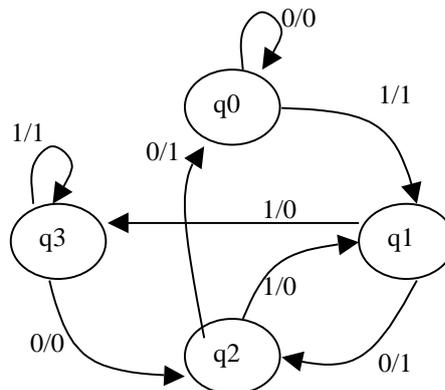
Como puede apreciarse, para poder calcular cada paridad de una serie de tres bits es necesario conservar los dos últimos bits, más el bit que se introduce en la entrada. Es necesario por lo tanto que el circuito secuencial sea capaz de recordar cuáles fueron estos dos últimos bits. De esta manera el número de estados diferentes será 4, correspondientes a las cuatro posibles combinaciones de 2 bits (00, 01, 10 y 11).

Una vez determinado el número de estados debe dibujarse el diagrama en el que se reflejan dos cosas:

1. La función de transición, que permite el cálculo del siguiente estado a partir del estado actual y la entrada, representada por las flechas que unen los diferentes estados.
2. La función de salida. Para cada transición se indicará cuál será la salida del circuito.

En nuestro caso el diagrama resultante será el de la figura adjunta, en la que los diferentes estados  $q_0$  a  $q_3$  corresponden a la descripción de la tabla adjunta.

Estado	Definición de estados
$q_0$	Dos últimos bits = 00
$q_1$	Dos últimos bits = 01 0 penúltimo 1 último
$q_2$	Dos últimos bits = 10 1 penúltimo 0 último
$q_3$	Dos últimos bits = 11



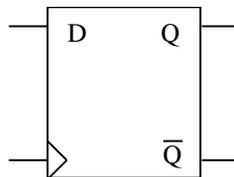
Supongamos por ejemplo que nos hallamos en el estado  $q_1$ , lo que significa que después del último cálculo de paridad tenemos almacenada la secuencia 01. Como puede apreciarse en el diagrama, un 1 en la entrada daría lugar a la combinación 1 + 10, cuya paridad es par. La salida por lo tanto será 0, y el siguiente estado  $q_3$ , puesto que ahora los dos últimos bits serán 11.

##### 2. Construcción de la tabla de estados

Tomando el diagrama de estados de la figura anterior debe generarse la tabla de estados, en la que a partir de las entradas y el estado actual se obtendrá el siguiente estado y la salida.

x	Q <sub>1</sub> (t)	Q <sub>0</sub> (t)	Q <sub>1</sub> (t+1)	Q <sub>0</sub> (t+1)	z
0	0	0	0	0	0
0	0	1	1	0	1
0	1	0	0	0	1
0	1	1	1	0	0
1	0	0	0	1	1
1	0	1	1	1	0
1	1	0	0	1	0
1	1	1	1	1	1

Como se observa hay dos columnas diferentes dentro de las funciones a implementar. La primera (Q<sub>1</sub>(t+1) Q<sub>0</sub>(t+1)) indicaría cuál debe ser la función de transición para el cálculo del nuevo estado. La segunda (es la función de salida). Sin embargo, puesto que no se pueden manipular directamente q1 y q0, sino las entradas D1 y D0 de los biestables, es necesario calcular qué valores hay que colocar en las entradas D para obtener los estados deseados. Dicho de otra manera si estamos en el estado Q<sub>1</sub>(t) y quiero pasar a Q<sub>1</sub>(t+1), qué debemos colocar como entrada D en el biestable de la figura



Para ello es necesario disponer de la tabla de excitación del biestable (del D en este caso):

Q(t)	Q(t+1)	D
0	0	0
0	1	1
1	0	0
1	1	1

A partir de esta tabla, conociendo Q(t) y Q(t+1) para ambos bits, podemos calcular los valores de entrada a los biestables D<sub>1</sub> y D<sub>0</sub>. La nueva tabla quedará por tanto de la siguiente manera.

x	Q <sub>1</sub> (t)	Q <sub>0</sub> (t)	Q <sub>1</sub> (t+1)	Q <sub>0</sub> (t+1)	z	D <sub>1</sub>	D <sub>0</sub>
0	0	0	0	0	0	0	0
0	0	1	1	0	1	1	0
0	1	0	0	0	1	0	0
0	1	1	1	0	0	1	0
1	0	0	0	1	1	0	1
1	0	1	1	1	0	1	1
1	1	0	0	1	0	0	1
1	1	1	1	1	1	1	1

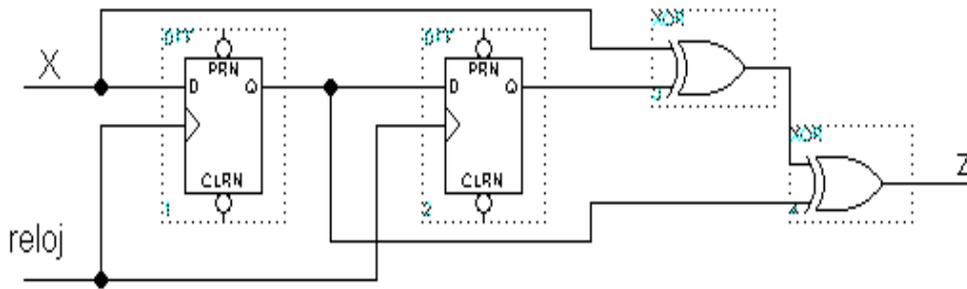
### 3. Minimización de las funciones e implementación del circuito

Simplificando por Karnaugh obtendremos las funciones de entrada a los biestables (D<sub>1</sub> y D<sub>0</sub>) para el cálculo del nuevo estado, y la función de salida Z, implementadas en la figura.

$$D_1 = Q_0$$

$$D_0 = x$$

$$Z = x \oplus Q_1 \oplus Q_0$$

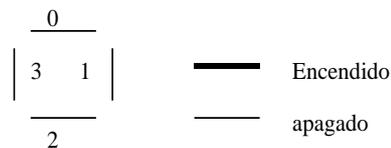


### 5º. Contador específico.

A veces el diseño de una secuencia se reduce a diseñar un contador cuya secuencia de cuenta sea una específica, que es función del propio enunciado. Veamos un ejemplo.

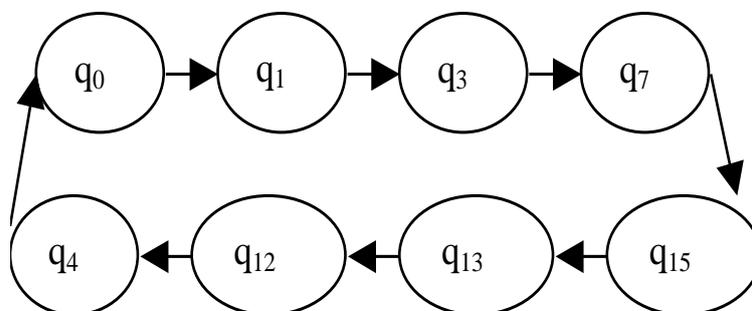
Se dispone de 4 barras luminosas dispuestas y numeradas según la figura. Se quiere diseñar una secuencia cíclica de iluminación controlada por una señal de reloj de frecuencia baja. La secuencia requerida es cíclica y es la que se representa en la figura.

Implementar un circuito digital que consiga realizar la secuencia descrita y que comience con todos los segmentos apagados, teniendo en cuenta que la iluminación de un segmento se logra proporcionando un nivel lógico 1. Disponer de biestables tipo T, decodificador/es para implementar las entradas a los biestables, y puertas OR necesarias. Respetar la numeración de los luminosos a la hora de codificar los estados.



#### 1. Paso de las especificaciones verbales al diagrama de estados

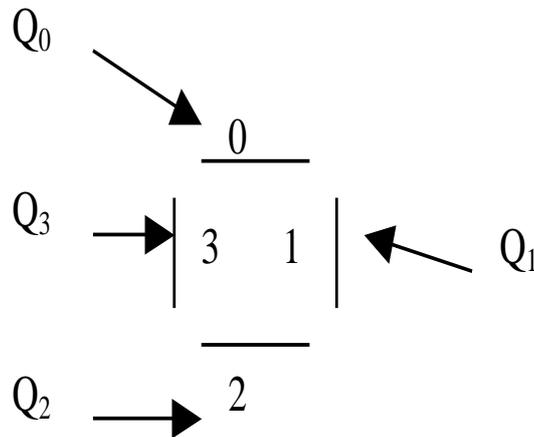
Como muestra la figura del enunciado, pueden presentarse 8 configuraciones diferentes en las barras luminosas, por lo que el número de estados necesario será también 8, una para cada configuración:



Como son 8 estados, podríamos asignar un código consecutivo del 000 al 111 y utilizar 3 bits para codificar cada uno de ellos. Sin embargo, como se verá en el circuito resultante puede resultar más sencillo utilizar un biestable para controlar el encendido o apagado de cada barra. De esta manera utilizaríamos 4 biestables. Aunque existen 16 posibles combinaciones, en realidad

sólo vamos a utilizar 8. El resto no se presentará nunca, por lo que pueden utilizarse como condiciones de “no importa” para simplificar las funciones resultantes.

Vamos a realizar la codificación de los estados, por lo tanto, de la siguiente manera. Tal y como muestra la figura, el biestable 0 controlará el encendido de la barra 0, el biestable 1 de la barra 1, y así sucesivamente.



De esta manera, por ejemplo, la primera configuración corresponderá a todos los bits a 0 (0000). La segunda, al caso en que  $Q_0(t)$  sea 1 y el resto 0 (0001), la tercera a  $Q_0(t)$  y  $Q_1(t)$  a 1 y el resto a 0 (0011), etc. Si la codificación se asigna de esta manera puede verse cómo las señales de activación de las barras coinciden exactamente con los valores almacenados en los biestables (máquina de Moore). De otra manera habríamos tenido que calcular una función de salida para cada barra, convirtiendo la codificación de los tres biestables usados en las 4 líneas de salida de activación de las luces correspondientes.

Así, la codificación de los estados será la siguiente:

Secuencia específica de cuenta	Codificación y activación de luces
0	0000
1	0001
3	0011
7	0111
15	1111
13	1101
12	1100
4	0100

## 2. Construcción de la tabla de estados

Tomando el diagrama de estados de la figura anterior debe generarse la tabla de estados, en la que a partir de las entradas y el estado actual se obtendrá el siguiente estado, y en la que a partir de ambos, y con ayuda de la tabla de excitación se calcularán las entradas a los biestables (T en este caso).

Cuenta	Q <sub>3</sub> (t)	Q <sub>2</sub> (t)	Q <sub>1</sub> (t)	Q <sub>0</sub> (t)	Q <sub>3</sub> (t+1)	Q <sub>2</sub> (t+1)	Q <sub>1</sub> (t+1)	Q <sub>0</sub> (t+1)	T <sub>3</sub>	T <sub>2</sub>	T <sub>1</sub>	T <sub>0</sub>
0	0	0	0	0	0	0	0	1	0	0	0	1
1	0	0	0	1	0	0	1	1	0	0	1	0
3	0	0	1	0	X	X	X	X	X	X	X	X
4	0	0	1	1	0	1	1	1	0	1	0	0
4	0	1	0	0	0	0	0	0	0	1	0	0
4	0	1	0	1	X	X	X	X	X	X	X	X
7	0	1	1	0	X	X	X	X	X	X	X	X
7	0	1	1	1	1	1	1	1	1	0	0	0
7	1	0	0	0	X	X	X	X	X	X	X	X
7	1	0	0	1	X	X	X	X	X	X	X	X
7	1	0	1	0	X	X	X	X	X	X	X	X
7	1	0	1	1	X	X	X	X	X	X	X	X
12	1	1	0	0	0	1	0	0	1	0	0	0
12	1	1	0	1	1	1	0	0	0	0	0	1
12	1	1	1	0	X	X	X	X	X	X	X	X
15	1	1	1	1	1	1	0	1	0	0	1	0

### 3. Minimización de las funciones e implementación del circuito

Simplificando obtendremos las funciones de entrada a los biestables (T<sub>3</sub>, T<sub>2</sub>, T<sub>1</sub> y T<sub>0</sub>) para el cálculo del nuevo estado. La función de salida, como se ha comentado, consiste simplemente en conectar cada salida de los biestables a la señal de activación de cada una de las 4 barras luminosas, tal y como se muestra en la figura.

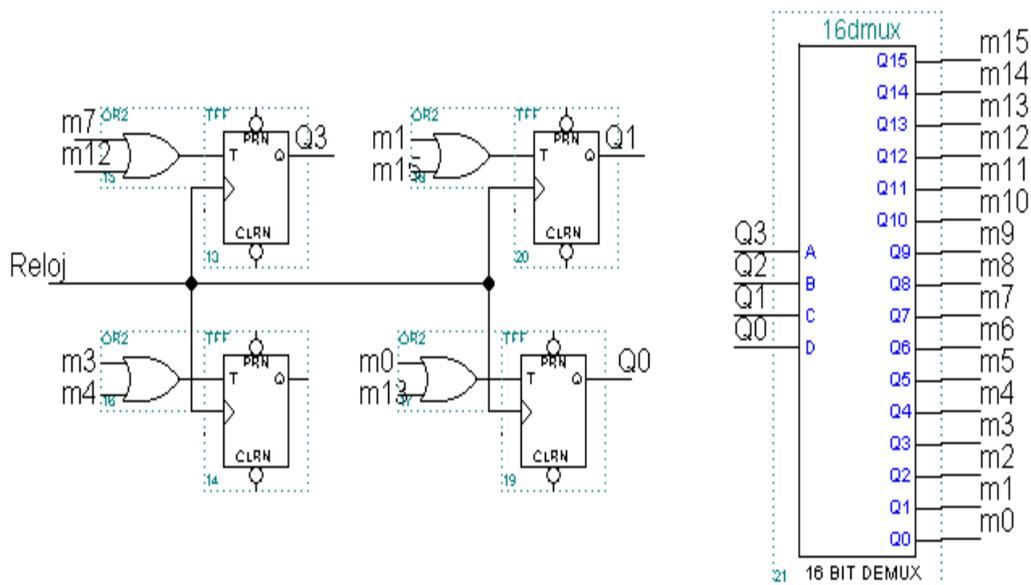
$$T_3 = m_7 + m_{12}$$

$$T_2 = m_3 + m_4$$

$$T_1 = m_1 + m_{15}$$

$$T_0 = m_0 + m_{13}$$

Puesto que el enunciado pide utilizar puertas OR y decodificadores, lo que haremos será utilizar un decodificador 4X16, al que conectaremos las salidas de los biestables (Q<sub>3</sub>, Q<sub>2</sub>, Q<sub>1</sub> y Q<sub>0</sub>), de manera que para la implementación de T<sub>3</sub>, por ejemplo, sólo tendremos que sumar las salidas 7 y 12 del decodificador.



## EJERCICIOS PROPUESTOS.

1) Diseñar un circuito secuencial con una entrada (E) y una salida (S), de manera que S tome el valor '1' cuando se hayan recibido por la entrada **al menos** dos ceros y **al menos** un uno, **independientemente del orden de aparición**. Una vez que S sea '1' se mantendrá este valor sea cual sea la entrada.

Ejemplos:

E: 011111001001                      E: 00000111  
S: 000000111111                      S: 000001...

Se pide:

1. El diagrama de estados y la codificación de los mismos (0,8 p).
2. La tabla de transición de estados, teniendo en cuenta que el bit más significativo del estado debe implementarse con un biestable tipo JK, el siguiente con un biestable tipo T y el último (si lo hubiera) con un biestable tipo D. (1,2p).

2) Una máquina de bebidas refrescantes proporciona una lata por valor de 0,80 € o 133 ptas. Sólo admite una moneda de 25, 50 y 100 ptas y de 0,20 € para adquirir la lata y no devuelve cambio. Si la cantidad introducida es mayor que la necesaria, proporciona la lata, poniendo a 1 la salida S y activa un panel luminoso que pone "Gracias por renunciar al cambio" poniendo a 1 la salida L. Si se introduce el precio exacto, sólo proporciona la lata, sin activar el panel luminoso.

No se puede cambiar el tipo de monedas con el que se empezó (si se comenzó con pesetas, se seguirá con pesetas, y si se comenzó con euros se seguirá con euros) salvo que se alcance el valor de 100 ptas ó 0,60 €, en cuyo caso se puede cambiar de moneda. Cada vez que entregue una lata, volverá al estado inicial, preparada para funcionar de nuevo.

Utilizar la siguiente codificación de monedas de entrada:

<u>AB</u>	<u>Significado</u>
00	moneda de 25 ptas
01	moneda de 50 ptas
10	moneda de 100 ptas
11	moneda de 0,20 €

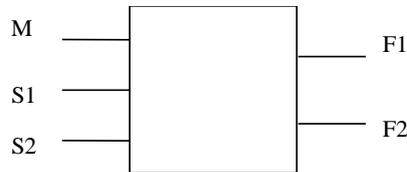
NOTA: Recordar que el valor de 1 € es de 166,386 ptas y que 100 ptas son exactamente 0,60 €.

Codificar los estados de forma que el valor numérico que indica el código generado con los biestables de mayor a menor peso sea mayor cuanto mayor sea la cantidad de dinero introducida en la máquina hasta ese momento, independientemente de la moneda introducida.

Se pide:

- 1) Definición y codificación de estados (0,6 puntos).
- 2) Diagrama de estados (0,7 puntos).
- 3) Tabla de transición y excitación usando biestables T (0,7 puntos).
- 4) Circuitaría combinacional adicional que admitiendo como entradas las salidas de los biestables ponga a 1 dos salidas, una P que activa un panel luminoso que pone "PESETAS" y otra E, que activa un panel luminoso que pone "EUROS" indicándole al usuario qué tipo de moneda debe introducir. Es posible que en algún caso puede introducir dos, y se encenderán ambos paneles poniendo a 1 sus respectivas entradas. Implementar P con un decodificador con salidas activas a nivel bajo y una puerta lógica con el mínimo número de entradas (0,3 puntos). ¿Cuál es la forma más sencilla de obtener E? (0,2 puntos).

3) El circuito de la Figura corresponde al sistema de apertura de una puerta de garaje controlada por un mando a distancia.



El sensor M detecta la orden del mando, mientras que los sensores S1 y S2 indica si la puerta está completamente cerrada o completamente abierta, respectivamente. M toma el valor lógico '1' cuando recibe la orden del mando, y cero en caso contrario. Los sensores S1 y S2 toman el valor lógico '1' cuando la puerta está completamente cerrada o completamente abierta, respectivamente, mientras que su valor es '0' durante las operaciones de apertura o cierre.

Las salidas F1 y F2 indican si está o no activado el motor de la puerta, y el sentido en caso de estar activado, respectivamente. Es decir, si F1 está activada ('1' lógico) entonces el motor se pone en marcha, mientras que al desactivarla ('0' lógico) éste se para. Mientras, F2 indica si el sentido del motor debe ser abrir ('1' lógico) o cerrar ('0' lógico). El circuito que gestiona la apertura de la puerta sigue los siguientes criterios:

- Cuando la puerta está cerrada y recibe una orden del mando (M) se pone en marcha la apertura de la puerta, que finalizará cuando el sensor S2 indique que está totalmente abierta (cuando tome el valor '1' lógico).
- Cuando la puerta está abierta y recibe una orden del mando (M) se pone en marcha el cierre de la puerta, que finalizará cuando el sensor S1 indique que está totalmente cerrada (cuando tome el valor '1' lógico).
- Si durante la apertura o cierre de la puerta, y antes de que se haya completado la operación, se recibe una nueva orden del mando, ésta será ignorada. Para ello, se pide:
  - a) Definir claramente los estados, entradas y salidas (0,5 puntos)
  - b) Dibujar el diagrama de estados (1 puntos).
  - c) Hacer la tabla de verdad del circuito (0,5 puntos).
  - d) Si se quiere hacer que la puerta se cierre sola transcurridos 32 segundos desde que se abrió totalmente y no habiendo recibido la orden del mando a distancia, ¿qué circuitería se podría añadir al circuito anterior para conseguir este efecto (0,5 puntos)?

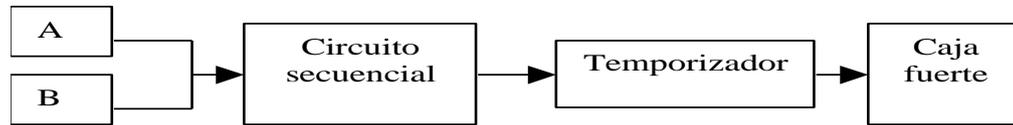
4) Diseñar un sistema secuencial capaz de reconocer el patrón 001011. La entrada al circuito se realizará a través de una señal de entrada E, de forma síncrona, y en serie (un nuevo bit cada ciclo de reloj). El circuito dispondrá de una salida S que tomará el valor '1' en el instante en el que se reconozca el patrón en la secuencia de entrada, y será '0' en todos los demás casos.

NOTA: debe tenerse en cuenta que al recibir un bit fuera de secuencia no hay que desechar todos los valores recogidos hasta ese momento. Es posible que parte de la secuencia siga siendo válida. Por ejemplo, si suponemos que se ha recibido la secuencia 00, y a continuación llega un tercer 0, se considerará que los 2 últimos ceros son parte de una secuencia correcta.

Se pide:

- a) Identificación y codificación de estados (0.75 p).
- b) Diagrama de estados (1 p).
- c) Completar la tabla de verdad teniendo en cuenta que deben utilizarse biestables tipo JK, T y D (de más a menos significativo) para la implementación física del circuito (0.75 p).

5) El sistema de apertura de una caja fuerte está compuesto por dos teclas A y B, un circuito secuencial a diseñar y un temporizador que mantiene la caja fuerte abierta durante 5 minutos cuando recibe un nivel lógico 1 desde el circuito secuencial. Este temporizador vuelve a cerrar la caja fuerte pasado dicho tiempo, independientemente del circuito secuencial. Cuando se pulsa la tecla A, se produce un nivel lógico 1 que entra al circuito secuencial, mientras que cuando se pulsa la tecla B se produce un nivel lógico 0 de entrada al circuito a diseñar. Mientras no se pulse ninguna tecla no se genera ningún nivel lógico de entrada al circuito secuencial.



Para abrir la caja fuerte, la combinación secreta es: pulsar dos veces seguidas la tecla A, a continuación pulsar una vez la tecla B, y finalmente pulsar una vez la tecla A. Si se hace de esta manera, el circuito secuencial dará una salida a nivel lógico 1, que actuará sobre el temporizador, permitiendo la apertura de la caja fuerte durante 5 minutos.

Si en cualquier momento se introdujera un error al pulsar la secuencia secreta, en el siguiente ciclo de reloj todos los biestables se pondrán a cero (el sistema pasará al estado inicial), y la secuencia debe volver a introducirse desde el principio.

- A) Dibujar el diagrama de estados, explicando claramente en qué consiste cada estado.
- B) Implementar el circuito secuencial a diseñar usando biestables JK y las puertas necesarias.

6) Se pretende diseñar un sistema secuencial síncrono con dos entradas E1 y E0, y una salida S usando biestables D, de manera que proporcione salida alta sólo cuando las dos entradas estén a nivel bajo habiendo estado también a nivel bajo ambas entradas en el ciclo de reloj anterior. Las transiciones se producen en el flanco de bajada del reloj. En los restantes ciclos de reloj, la salida debe ser baja. Las señales de entrada son periódicas, con un periodo 5 veces superior al período de reloj, y son las mostradas en la figura. NOTA: Observar que no se producen todas las transiciones posibles.

- a) Realizar el diagrama de transición de estados, definiendo y codificando los estados y las entradas.
- b) Realizar la tabla de verdad que resuelve el problema.
- c) Encontrar el circuito.

7) Una fuente de agua posee tres focos luminosos (azul, rojo, verde) y una caja musical que tiene 7 canales digitales de entrada, cada uno de ellos para activar cada una de las 7 notas musicales. Por ejemplo, si se proporciona un 1 lógico por el canal de entrada de la nota "Mi" sonará esa nota durante un ciclo de reloj. Suponer que la secuencia que se quiere seguir en la iluminación con los focos es: todos apagados, rojo, azul, rojo y azul, verde, verde y rojo, verde y azul, todos encendidos. Al mismo tiempo la secuencia de notas musicales debe ser: Re, Mi, Fa, Sol, Do, La, Si, La. Diseñar un circuito secuencial que implemente esta secuencia 5 veces y que termine con todos sus componentes apagados: luces y sonido.

1) Resolver el sistema de iluminación tomando la codificación más adecuada al enunciado del problema. Mostrar el diagrama de estados, la codificación de estados y la tabla de excitación del sistema. Implementar este subsistema de la forma más sencilla posible utilizando biestables JK. (1 punto).

2) Resolver el sistema digital, asociado al diseño anterior, que excite la caja musical según la secuencia indicada. (1 punto).

3) Resolver el sistema de apagado de los dos sistemas anteriores. (1 punto).

Utilizar en el diseño puertas lógicas (las imprescindibles), módulos combinacionales y contadores.