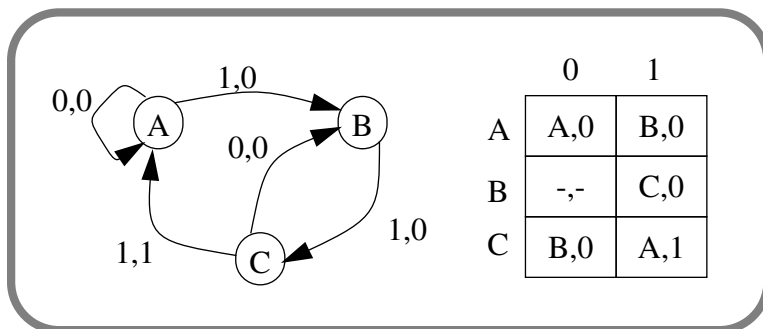
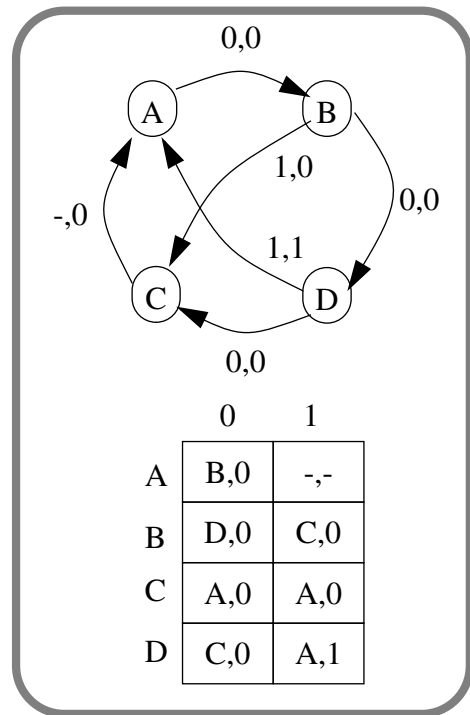
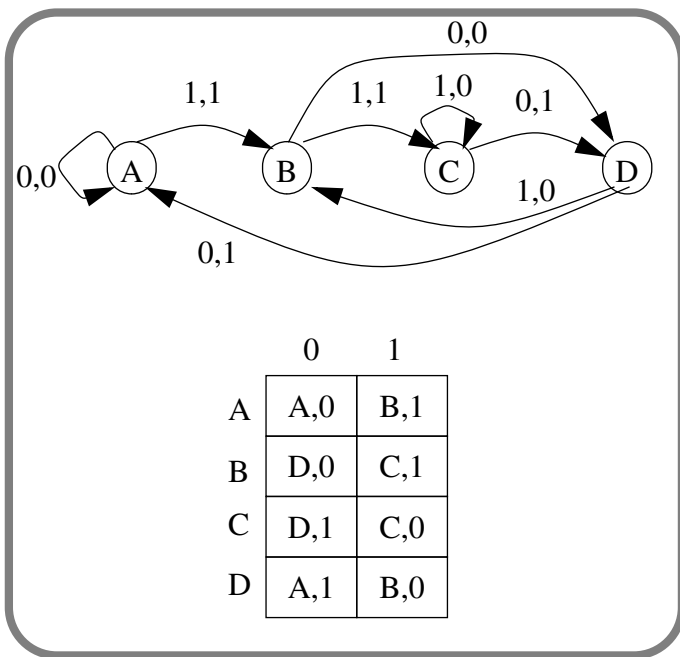
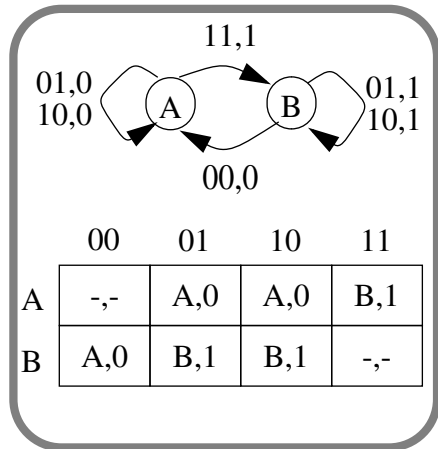
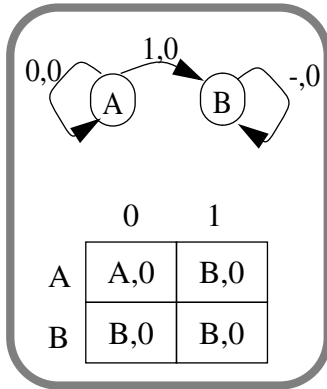
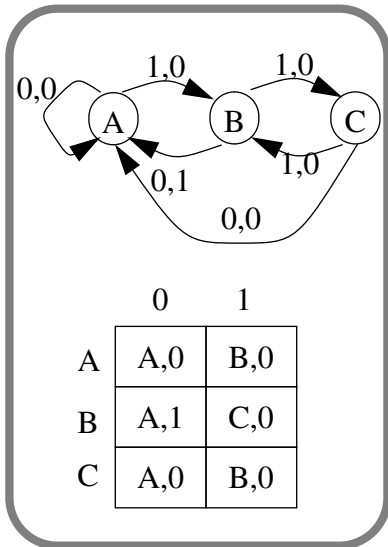
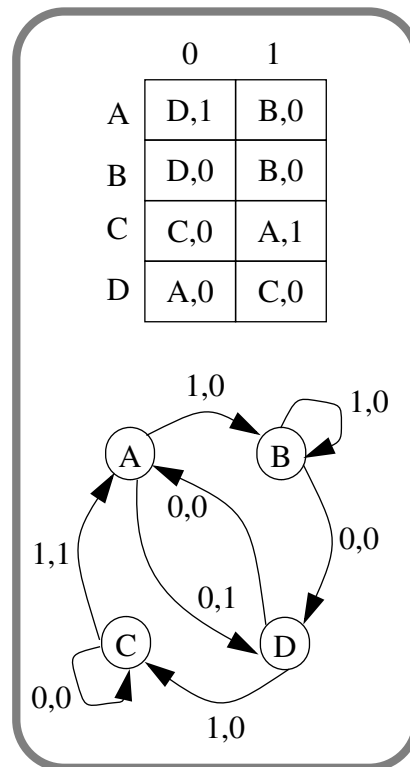
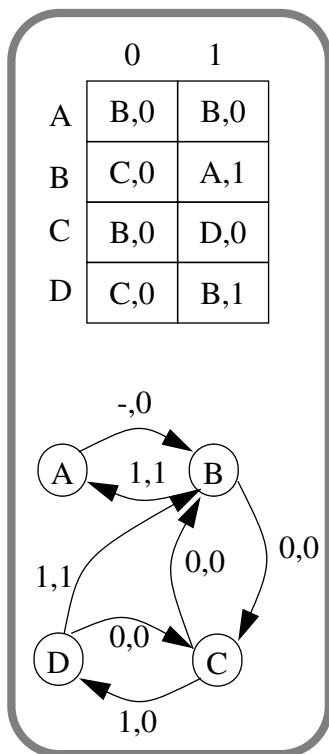
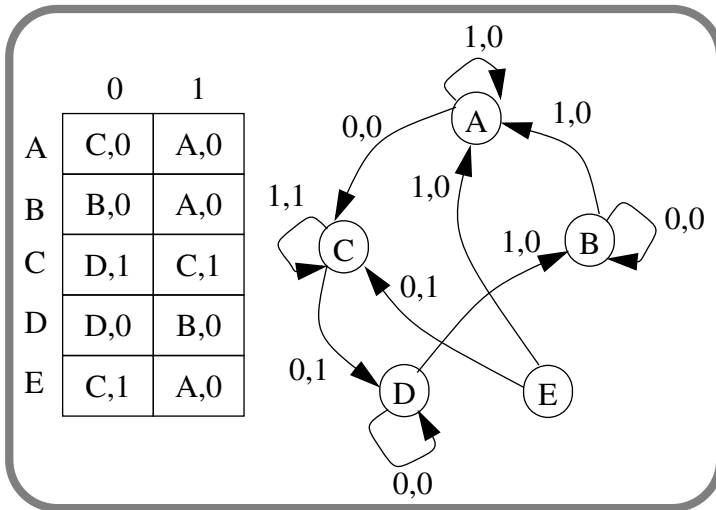
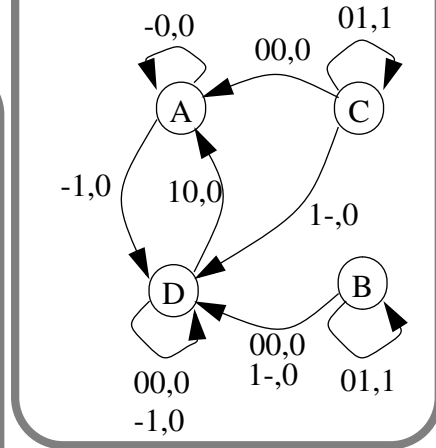
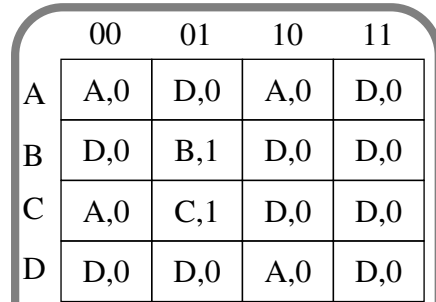
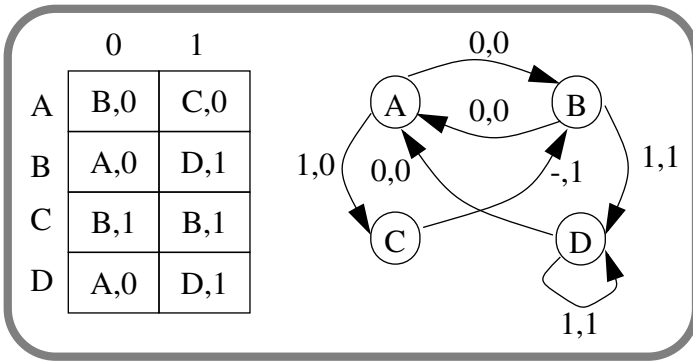


### Problemas del Tema 1

1.1. Obtener las tablas de estado correspondientes a los siguientes diagramas:



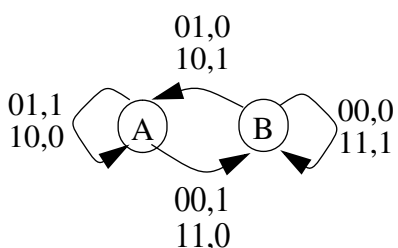
1.2. Obtener los diagramas de estado correspondientes a las siguientes tablas de estado:



1.3. Determinar si las siguientes máquinas de estado se corresponden con máquinas de Moore. En caso negativo, obtener la descripción de dicha máquina como una máquina de Moore.

Se define una máquina de Moore como aquella en la que la salida sólo depende del estado. Por lo tanto, en una máquina de Moore siempre se llegará a un estado con el mismo valor de salida, independiente de las señales de entradas. En caso contrario, estamos ante una máquina de Mealy.

Para pasar de una máquina de Mealy a una de Moore, la nueva máquina deberá tener tantos estados tal que contemos como estados diferentes, los estados a los que se llega con diferente valor de salida.



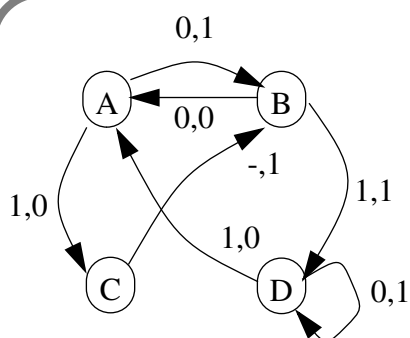
Máquina de Mealy, pues a los estados A y B se llegan con dos valores de salida diferentes

	00	01	10	11
A	B,1	A,1	A,0	B,0
B	B,0	A,0	A,1	B,1

La nueva máquina tendrá cuatro estados:

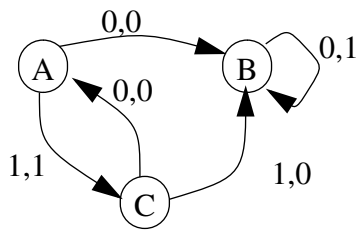
- A0, A con F=0
- A1, A con F=1
- B0, B con F=0
- B1, B con F=1

	00	01	10	11	S
A0	B1	A1	A0	B0	0
A1	B1	A1	A0	B0	1
B0	B0	A0	A1	B1	0
B1	B0	A0	A1	B1	1



Máquina de Moore, ya que al estado:

- A se llega con F=0
- B se llega con F=1
- C se llega con F=0
- D se llega con F=1



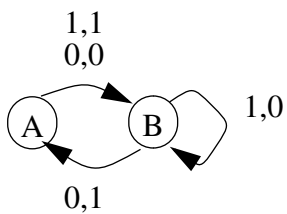
Máquina de Mealy, puesto que al estado B se llega con dos valores de salida diferentes

La nueva máquina tendrá cuatro estados:

	0	1
A	B,0	C,1
B	B,1	-, -
C	A,0	B,0

- A, A con F=0
- B0, B con F=0
- B1, B con F=1
- C, C con F=1

	0	1	S
A	B0	C	0
B0	B1	-	0
B1	B1	-	1
C	A	B0	1



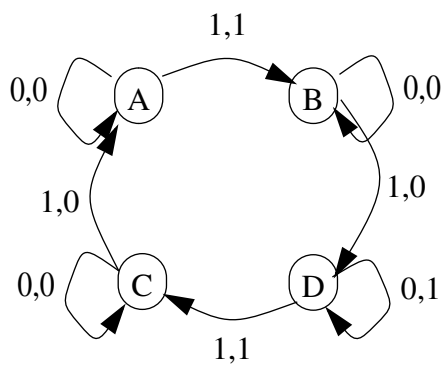
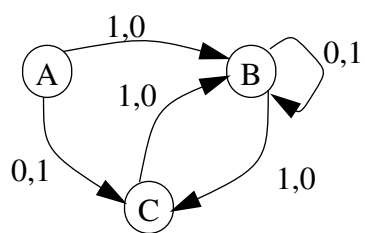
Máquina de Mealy, puesto que al estado B se llega con dos valores de salida diferentes

	0	1
A	B,0	B,1
B	A,1	B,0

La nueva máquina tendrá cuatro estados:

- A, A con F=0
- B0, B con F=0
- B1, B con F=1

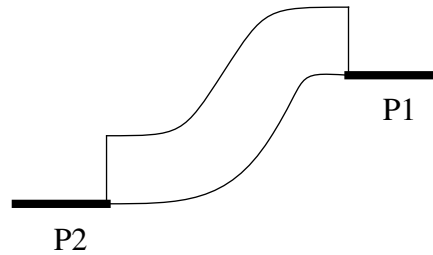
	0	1	S
A	B0	B1	1
B0	A	B0	0
B1	A	B0	1



## Problemas del Tema 2

2.1. Encontrar una máquina de estados para los siguientes sistemas:

- Sistema de control de una escalera mecánica bidireccional. Para ello disponemos de dos sensores de presión, P1 y P2, como se muestra en la figura adjunta. Cuando se activa un sensor, la escalera empezará a andar en dirección al sensor restante y no parará hasta que se active el otro sensor. La máquina deberá mostrar el comportamiento de la dirección de los motores, así como su estado (encendido o apagado).



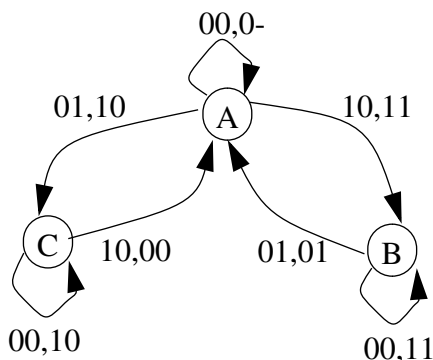
Las situaciones que nos podemos encontrar son las siguientes:

- La escalera está parada
- La escalera está subiendo
- La escalera está bajando

Como señales de entrada tendremos los dos sensores de presión, y como salidas tendremos la acción (A) y el sentido (S) del motor. Luego, las transiciones estarán etiquetadas por P1P2,AS.

Las posibles secuencias que hay que describir son las siguientes:

- Una persona se coloca sobre el sensor P1, luego la escalera debe empezar a bajar, hasta que dicha persona se coloque sobre el sensor P2
- Cuando no se active ninguno de los sensores la escalera debe permanecer parada.



A: escalera parada  
B: escalera bajando  
C: escalera subiendo

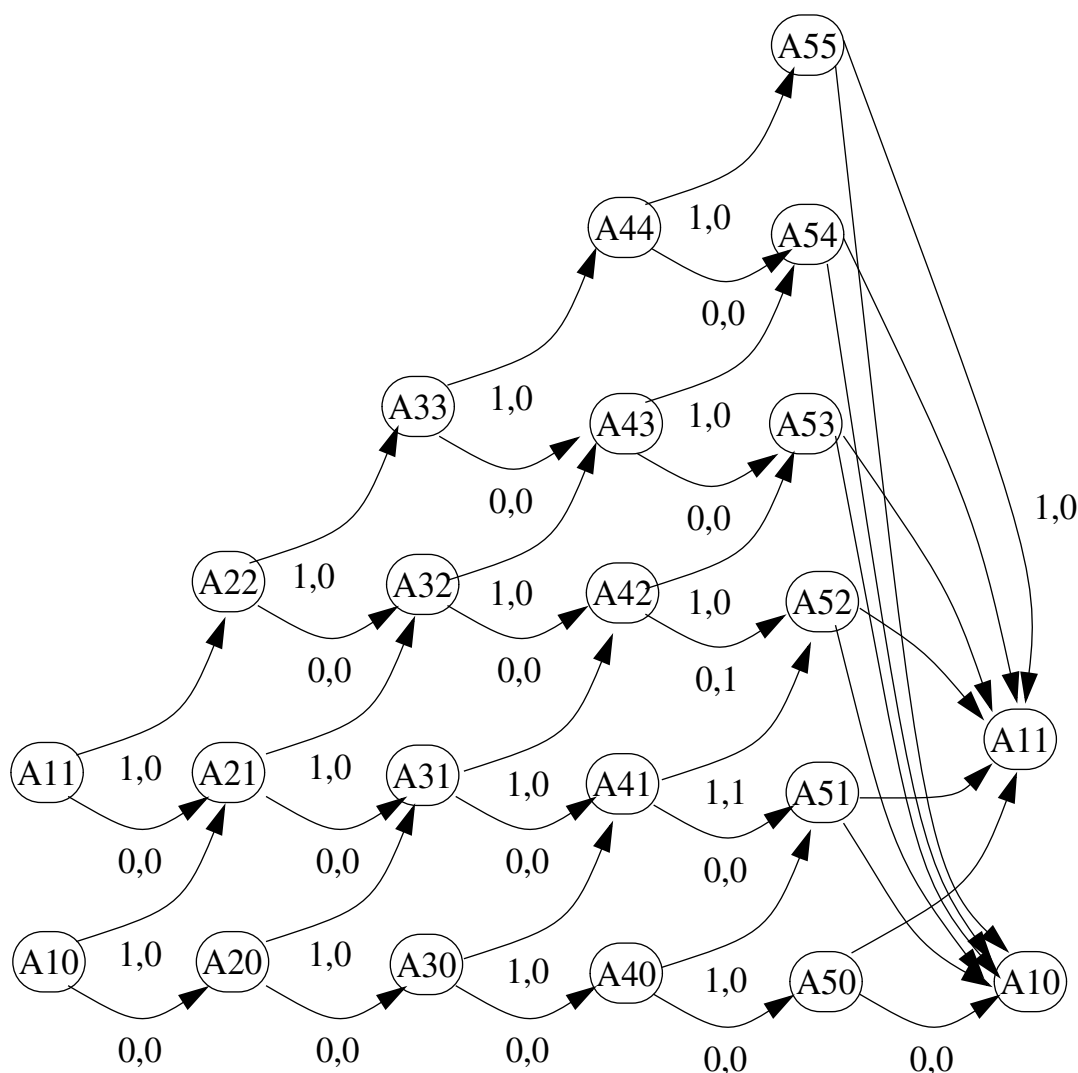
señales  
de salida

A=0, motor parado  
A=1, motor moviéndose  
S=0, sentido hacia arriba  
S=1, sentido hacia abajo

No se ha considerado la situación de que se coloquen de forma simultánea dos personas sobre P1 y P2 porque no ha sido contemplada en las especificaciones de diseño.

- Sistema de detección de error de un código "2-out-of-5", con un solo canal de entrada por el que recibimos los bits de la palabra secuencialmente (uno detrás de otro). La determinación de la palabra correcta o no deberá coincidir con la llegada del último (quinto) bit.

Como el código "2-out-of-5" es aquel en el que las palabras de código son las que tienen dos 1's en cinco bits, este sistema tendrá que contar el número de 1's según el número de bits que nos lleguen. Por lo tanto, tendremos tantos estados como combinaciones del número de 1's con el número de bits que han llegado.



Los estados han sido nombrados como AIJ, donde I es el número de bit de la palabra de código que ha llegado, y J es el número de 1's que tiene la palabra hasta dicho momento. Luego, las palabras correctas serán aquellas que llegan al estado A52. Para clarificar el diagrama, se ha duplicado los estados A11 y A10 (los estados repetidos son los mismos).

- Sistema encargado por la siguiente carta:
 

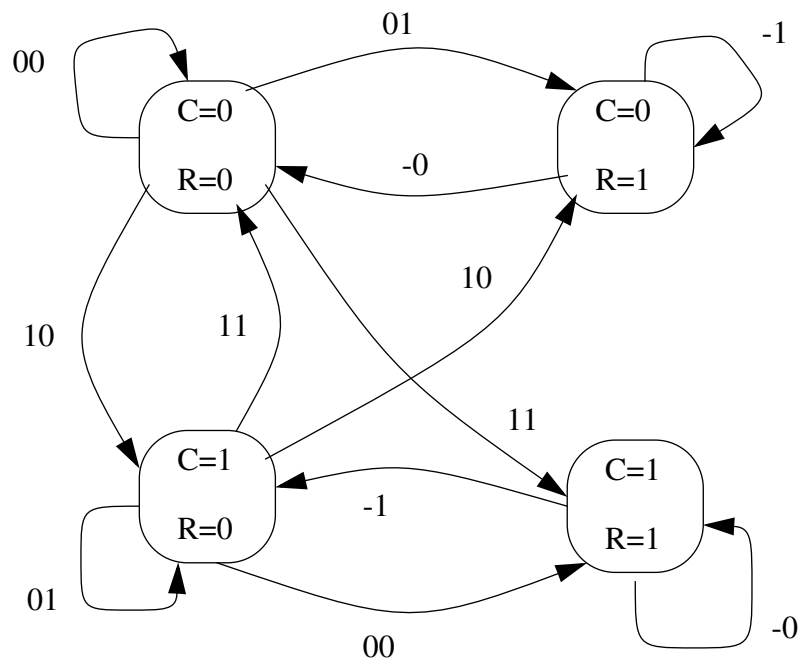
“Querido amigo: Al poco tiempo de comprar esta vieja mansión, tuve la desagradable sorpresa de comprobar que está hechizada con dos sonidos de ultratumba: un canto picaresco y una sonrisa sardónica. Estos sonidos obedecen a ciertas leyes en función del sonido de un órgano o al olor de incienso quemado; de tal forma que en cada minuto el sonido está presente o ausente. El estado de cada sonido depende del siguiente comportamiento:

  - El canto conservará su estado (presente o ausente) salvo si durante el minuto actual no se oye la risa y toco el órgano, en cuyo caso el canto tomará el estado opuesto.
  - En cuanto a la risa, si no quemo incienso, se oirá o no según el canto esté presente o ausente (de modo que la risa imita al canto con un minuto de retardo). Pero si quemo incienso, la risa hará lo contrario que hacía el canto.”

En este sistema tenemos dos variables de entrada, sonido del órgano (O) y el olor a incienso (I), y dos variables de salida, el canto picaresco (C) y la risa sardónica (R). Debido a las dos variables de salida podemos tener cuatro situaciones diferentes. Las secuencias serán las siguientes:

- $R = 0$  y  $O = 1 \rightarrow \bar{C}$ , en caso contrario,  $C$
- $I = 0 \rightarrow R = C_{-1}$
- $I = 1 \rightarrow R = \bar{C}_{-1}$

Si las etiquetas de entrada son OI, la máquina de estado que describiría el sistema sería el mostrado a continuación





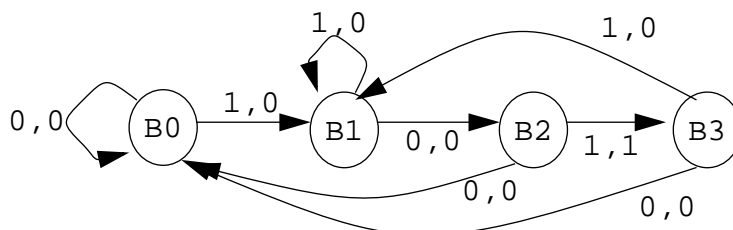
- En el caso anterior, ¿cuál será la secuencia de hechos para que no se escuche ni el canto ni la risa, si inicialmente se escuchaban los dos?, ¿cuánto tiempo pasará hasta que la casa se quede en silencio?

Si inicialmente se escuchaban los dos sonidos (nos encontramos en el estado  $R=C=1$ ), bastaría con quemar incienso durante dos minutos y tocar el órgano durante el último minuto. Después no se debería ni quemar incienso ni tocar el órgano, para mantener en silencio los dos sonidos de ultratumba.

El tiempo que sería necesario para acallar la casa serían los dos minutos que tardaría en llegar al estado de silencio ( $R=C=0$ ).

- Sistema de detección de fin de cadena en un equipo de telecomunicaciones. El sistema deberá detectar la cadena "101".

Este sistema debe detectar una secuencia de números, por lo tanto deberemos tener tantos estados como números tenga la secuencia más un estado inicial (necesario por si la secuencia no comienza por el valor correcto). Si en algún momento, la secuencia no es correcta, deberemos iniciar la detección porque en ningún momento se nos dice que debemos mantener la detección cada tres bits.



En el caso de que la secuencia se rompa por la llegada de un '1', dicho bit puede ser el primero de la secuencia, y por eso el próximo estado debe ser el B1 y no el B0.

También si lo pensamos bien, los estados B0 y B3 son los mismos, por lo que bastaría con poner uno solo de ellos.

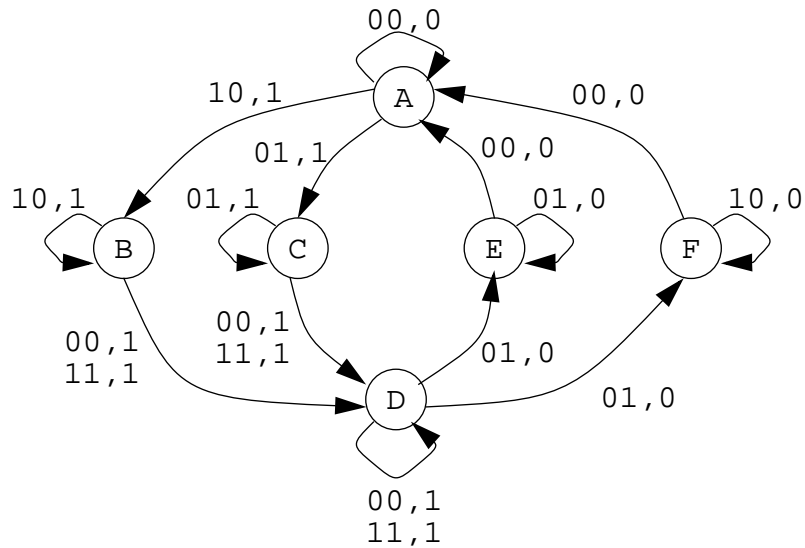
- Sistema de control de un paso a nivel con barreras. El sistema contará con dos sensores de posición, P1 y P2, a una distancia igual a 200 m. del paso a nivel.

En este sistema nos podemos encontrar con seis posibles situaciones:

- No esté ningún tren dentro (A)
- El tren esté entrando por la derecha (B)
- El tren esté entrando por la izquierda (C)
- El tren esté dentro de los límites del paso a nivel, es decir, dentro de los 400m. que separan a los dos sensores (D)

- El tren esté saliendo por la izquierda (E)
- El tren esté saliendo por la derecha (F)

El diagrama de estado de esta máquina se muestra a continuación.

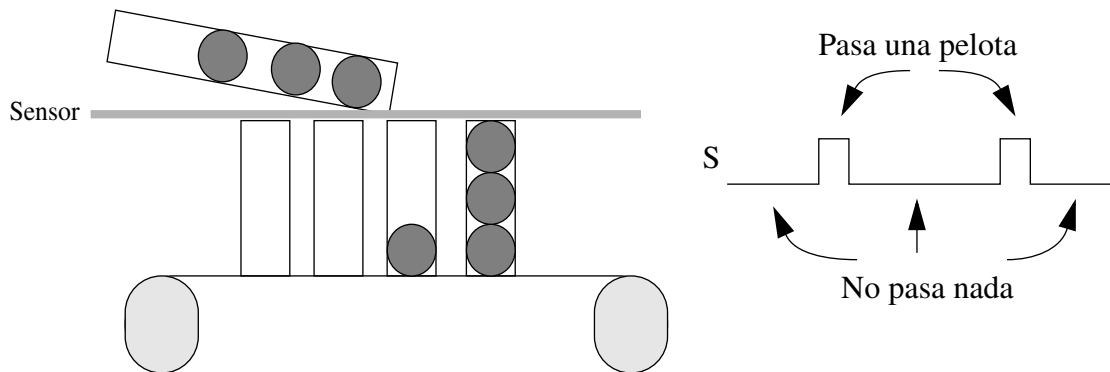


Si lo pensamos bien, los estado B y C son los mismos, igual que los estados E y F. Por lo tanto, con dejar uno de cada pareja de dichos estados bastaría.

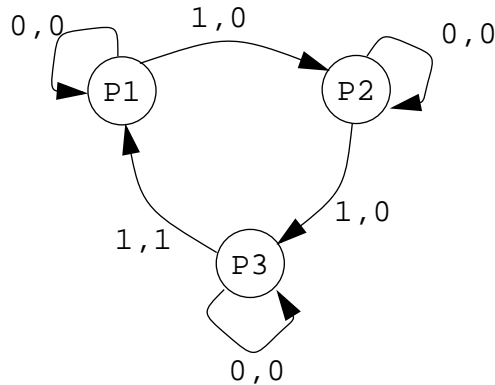
- ¿Cómo cambiará el apartado anterior si sabemos que la longitud máxima de los trenes que utilizarán dicha vía no superará los 300 m.?

La única diferencia con el caso anterior es que la secuencia de entradas "11" no se puede dar nunca, y por lo tanto no aparecería en ninguna transición.

- En el departamento de logística de una fabrica de pelotas de tenis, éstas ruedan por una cinta hasta caer en una caja que está sobre una cinta transportadora. La cinta transportadora debe dar un paso cuando una caja esté llena (con un total de tres pelotas), para colocar debajo una nueva caja. Se desea diseñar el sistema de control de la cinta transportadora, sabiendo que cuando cae una pelota en la caja activa un sensor de movimiento colocado encima de la entrada a la caja, como se muestra en la siguiente figura. Las pelotas caen con el suficiente espacio temporal para que no se solapen los pulsos del sensor.



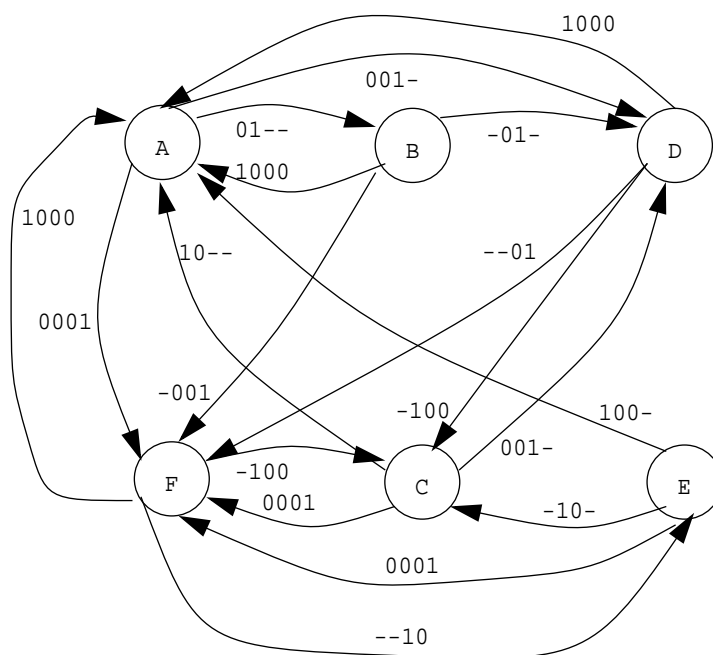
Este nuevo sistema debe contar tres pulsos del sensor. Por lo tanto, la máquina de estado tendrá tres estados (uno por cada pulso).



- Sistema de control de un ascensor para un edificio de cuatro plantas, de tal forma que las salidas deben ser la acción del motor y el sentido. Para evitar peticiones simultáneas, el sistema debe incluir las siguientes prioridades en el mismo orden:
  - Mantener en el caso que sea posible el mismo sentido (ascendente o descendente)
  - Pasar primero por el piso más próximo al actual

Las posibles situaciones que nos podemos encontrar son que el ascensor esté en:

- la planta baja (A)
- la primera planta y subiendo (B)
- la primera planta y bajando (C)
- la segunda planta y subiendo (D)
- la segunda planta y bajando (E)
- la tercera planta (F)

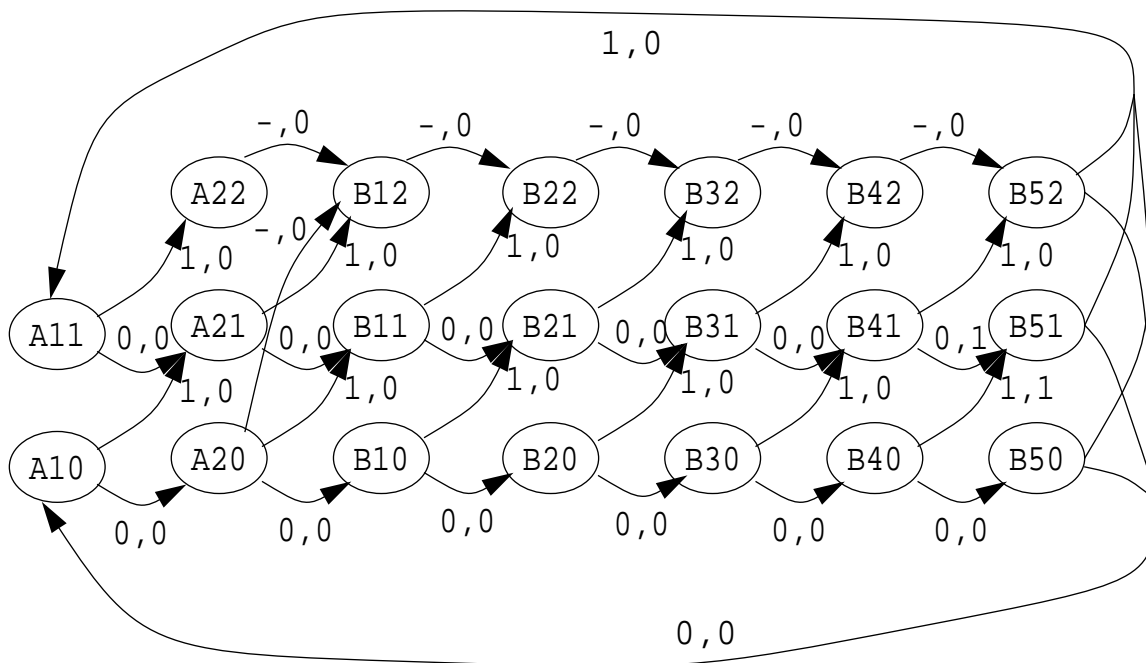


Señales de entrada  
I0 I1 I2 I3

Por claridad del diagrama no se han puesto las transiciones que no deben cambiar de estado, que se darían cuando no se pulsa ningún botón, o el botón que se ha pulsado es el de la planta en la que está (situación más prioritaria).

- Sistema de detección de error de un código biquinario, con un solo canal de entrada por el que recibimos los bits de la palabra secuencialmente (uno detrás de otro). La determinación de la palabra correcta o no deberá coincidir con la llegada del último bit.

Este código tiene un total de siete bits, de tal forma que en los dos primeros debe haber un solo '1', y en los otros cinco también debe haber un solo '1'. Por lo tanto, además de los estados necesarios para contar los 1's también debemos tener estados de error, por si en los dos primeros bits, el número de 1's es diferente de uno.



El estado de error es aquel que tiene un número mayor o igual a dos. Debemos mantener este estado para cada bit ya que debemos contar los siete bits para saber cuando empieza una palabra.

- Sistema codificador de números binarios naturales a código Gray. Los datos tendrán una longitud de cuatro bits, los cuales irán entrando de forma secuencial (uno detrás de otro). Además de generar los datos en el nuevo código, también deberá detectar el final de un dato y el comienzo del siguiente.
- Sistema generador de paridad para bits que van llegando de forma secuencial (uno detrás de otro). El sistema deberá tener un estado de inicialización para separar una generación de otra.

- Sistema de alarma de una planta industrial. En la planta hay elementos inflamables según las siguientes condiciones (existen sensores que nos indican cuando la presión, densidad y la exposición al sol han superado su límite):
  - Si la presión del contenedor es igual o superior a 4 unidades, después de haber estado expuesto al sol
  - Si la densidad del contenedor es igual o superior a 5 unidades, después de estar sometido a una presión mayor o igual a 4 unidades.

2.2. Minimizar las siguientes máquinas de estado.

La reducción de las máquinas de estado completamente especificadas, se reduce a determinar cuáles son los máximos compatibles. Siendo estos máximos, los estados de la máquina reducida. Por lo tanto, debemos realizar los siguientes pasos:

- Determinar los pares compatibles
- Determinar los máximos compatibles
- Generar la nueva tabla o diagrama de estados.

	0	1
A	D,0	F,0
B	D,0	E,0
C	G,0	E,1
D	A,0	F,0
E	E,1	E,1
F	A,0	B,0
G	C,0	G,1

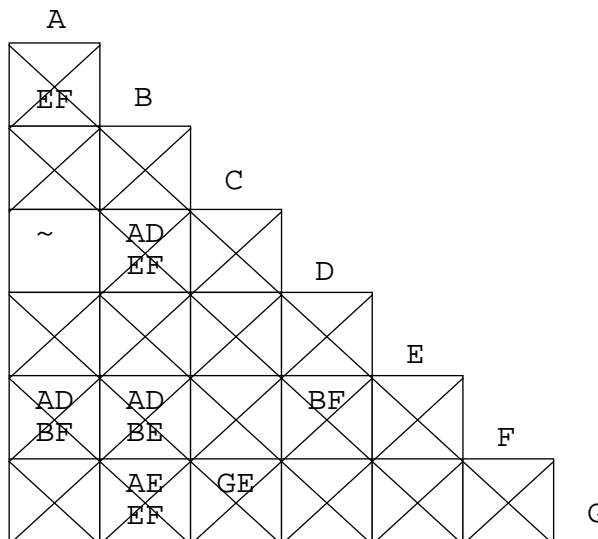


Tabla de pares compatibles

	0	1
AD	AD,0	F,0
B	AD,0	E,0
C	G,0	E,1
E	E,1	E,1
F	AD,0	B,0
G	C,0	G,1

Estados C.E.C. Máximos compatibles

G	-	G
F	-	G, F
E	-	G, F, E
D	-	G, F, E, D
C	-	G, F, E, D, C
B	-	G, F, E, D, C, B
A	D	<b>G, F, E, AD, C, B</b>

Lista de máximos compatibles

Máquina mínima

	00	01	10	11
A	E,1	A,1	A,0	D,1
B	B,0	C,0	C,0	D,1
C	F,1	C,1	C,0	D,1
D	G,1	D,1	D,0	D,1
E	E,0	E,0	A,0	D,1
F	F,0	F,0	A,0	D,1
G	G,0	A,1	B,1	G,0
H	B,0	C,1	H,1	H,0

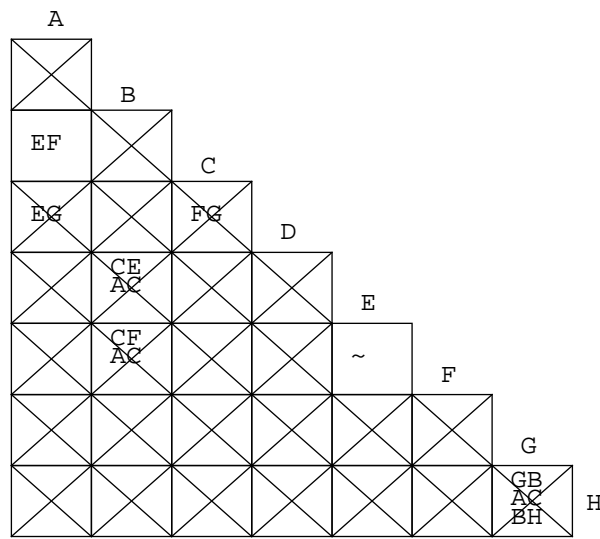


Tabla de pares compatibles

Estados C.E.C. Máximos compatibles

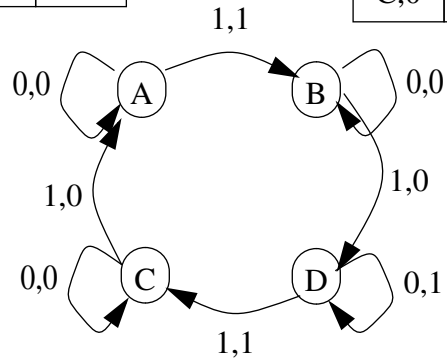
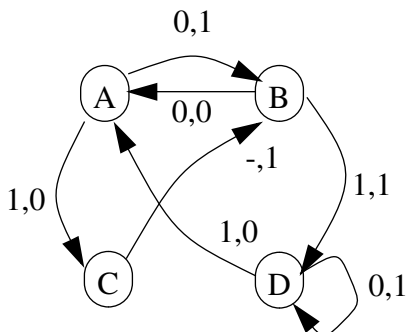
H	-	H
G	-	G, H
F	-	F, G, H
E	F	EF, G, H
D	-	D, EF, G, H
C	-	C, D, EF, G, H
B	-	B, C, D, EF, G, H
A	C	<b>B, AC, D, EF, G, H</b>

	00	01	10	11
AC	EF,1	AC,1	AC,0	D,1
B	B,0	AC,0	AC,0	D,1
D	G,1	D,1	D,0	D,1
EF	EF,0	EF,0	AC,0	D,1
G	G,0	AC,1	B,1	G,0
H	B,0	AC,1	H,1	H,0

	0	1
A	B,0	C,0
B	A,0	D,1
C	B,1	B,1
D	A,0	D,1

	0	1
A	D,1	B,0
B	D,0	B,0
C	C,0	A,1
D	A,0	C,0

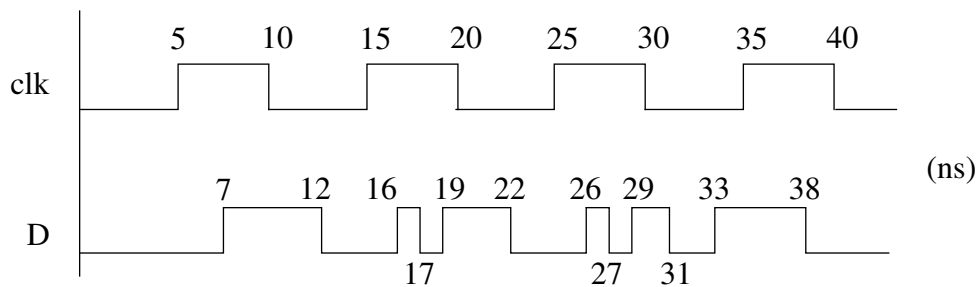
	0	1
A	B,0	B,0
B	C,0	A,1
C	B,0	D,0
D	C,0	B,1



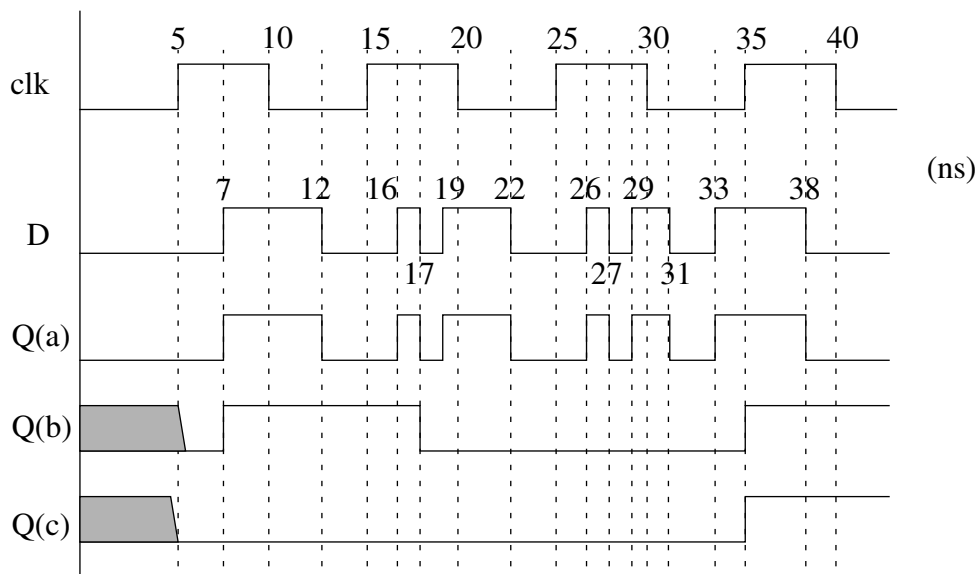
### Problemas del Tema 3

3.1. Completar las formas de onda para un elemento de memoria tipo D (a) transparente, (b) latch sensible al nivel alto y (c) flip-flop disparado por la transición de subida. Los biestables tendrán las siguientes características temporales, según corresponda:

- $t_{\text{setup}} = 1 \text{ ns}$ .
- $t_{\text{hold}} = 2 \text{ ns}$ .
- $t_w = 3 \text{ ns}$ .



Los elementos de memoria realizan la operación correcta siempre y cuando se cumplan una serie de restricciones temporales. Estas restricciones son básicamente: el tiempo que deben permanecer los datos estables antes de poder almacenarse (setup) ;el tiempo que deben permanecer los datos estables después de almacenarse (hold); yel tiempo que debe permanecer estable la señal de reloj para poder almacenarse (width); donde los dos primeros se deben cumplir para todos los elementos no transparente, mientras que el último sólo se deberá cumplir para los latches (o elementos sensibles al nivel). Así la salida de los biestables sería las siguientes:



En el caso de elemento transparente (a), la señal de reloj no influye en la operación y al tratarse de un elemento tipo D, las señales de dato y de salida son idénticas.

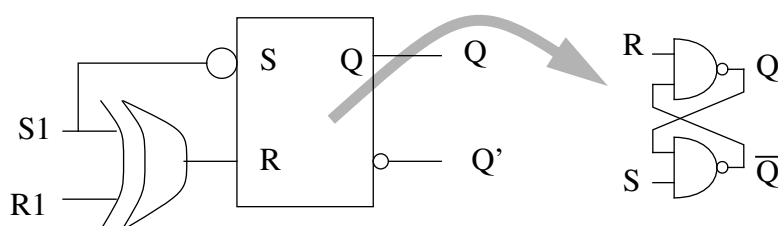
En el caso del latch (b) sólo se almacenarán los cambios durante el nivel activo (alto) de la señal de reloj. Se deben cumplir todas las restricciones temporales. La anchura de pulso siempre se cumple, ya que los pulsos de la señal de reloj tienen una anchura de 5ns, que es mayor que la mínima (3ns). La primera subida de la señal de reloj cumple las restricciones de setup y hold, ya que el almacenamiento se puede producir en el intervalo 7ns-10ns, que cumple que es mayor que 1ns (setup) y que 2ns (hold). El pulso del dato correspondiente a 16ns-17ns no cumple las restricciones, y por lo tanto no es almacenado. Y los únicos pulsos de dato que cumplen las restricciones, y por tanto son almacenados, son 17ns-19ns y 35ns-38ns.

En el caso del flip-flop (c) sólo se almacenarán los valores que tenga la señal de dato en la transición activa (de subida) de la señal de reloj. En este caso, la anchura de pulso es reducida a una transición, y por lo tanto esta restricción no debe considerarse. En este caso, las únicas transiciones que cumplen las restricciones de setup y hold son la primera y la última, siendo éstas las únicas que se almacenen.

Es interesante notar que en el que caso de los latches y flip-flops, durante el inicio de la operación (que no se debe almacenar ningún dato), la salida no está especificada pudiendo ser cualquiera. Para ello se añaden a estos elementos señales de control asíncronas (como un reset) para que la salida siempre esté completamente especificada. Además, notar que por el "simple" cambio de temporización, el comportamiento es totalmente diferente.

3.2. Repetir el ejercicio anterior utilizando biestables tipo T.

3.3. Realizar un estudio del siguiente elemento de memoria.





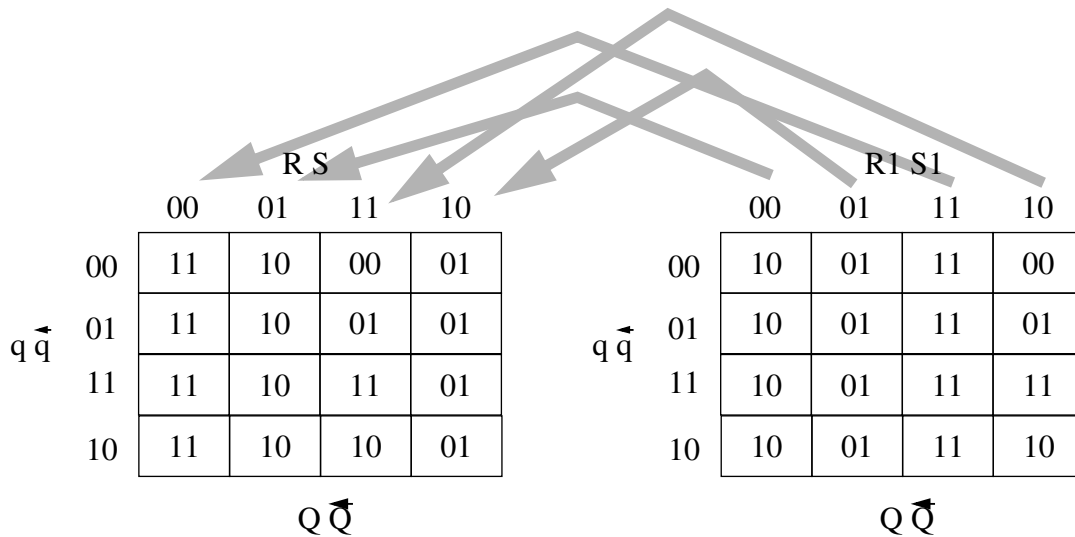
En primer lugar hay que deducir cuáles son las funciones a las que están conectadas las señales de entrada del latch RS, es decir, S y R:

- $S = \overline{S1}$
- $R = S1 \text{ xor } R1$

El siguiente paso es obtener la tabla de combinaciones de las salidas a partir de las entradas S1 y R1. Para esto vamos también a mostrar la tabla de combinaciones de un biestable RS.

		S1	
		0	1
R1	0	1	0
	1	1	0
		S	

		S1	
		0	1
R1	0	0	1
	1	1	0
		R	



Podemos apreciar que las tablas del nuevo biestable y la del RS coinciden variando las etiquetas de las columnas, es decir, todo lo estudiado para el biestable RS es válido cambiando las combinaciones de entrada:

- R1S1 = "00", Combinación de set. Existe una carrera no crítica para el estado presente "01"
- R1S1 = "01", Combinación de reset. Existe una carrera no crítica para el estado presente "10"
- R1S1 = "11", Combinación prohibida, para que Q y Q̄ sean complementarios
- R1S1 = "10", Combinación de almacenamiento. Para esta combinación, el circuito no cambia de estado.

3.4. Obtenga:

- Biestables JK, T y D a partir de biestables RS

Para realizar esta transformación, lo que debemos hacer es obtener los valores R y S del biestable RS, para que a partir de las nuevas entradas tengamos el comportamiento de los nuevos biestables. Para ello, en primer lugar vamos a ver cuanto deben valer las entradas de los diferentes biestables para cada cambio de estado.

D

		0	1
q	0	0	1
	1	0	1

Q

RS

		00	01	11	10
q	0	0	1	-	0
	1	1	1	-	0

Q

T

		0	1
q	0	0	1
	1	1	0

Q

JK

		00	01	11	10
q	0	0	0	1	1
	1	1	0	0	1

Q

q -> Q ..... D ..... T ..... RS ..... JK  
 0 -> 0 ..... 0 ..... 0 ..... - 0 ..... 0 -  
 0 -> 1 ..... 1 ..... 1 ..... 0 1 ..... 1 -  
 1 -> 0 ..... 0 ..... 1 ..... 1 0 ..... - 1  
 1 -> 1 ..... 1 ..... 0 ..... 0 - ..... - 0

Luego, las entradas RS estarán determinadas por los siguientes mapas:

D

		0	1
q	0	-0	01
	1	10	0-

RS  
 $R = \bar{D}$   
 $S = D$

T

		0	1
q	0	-0	01
	1	0-	10

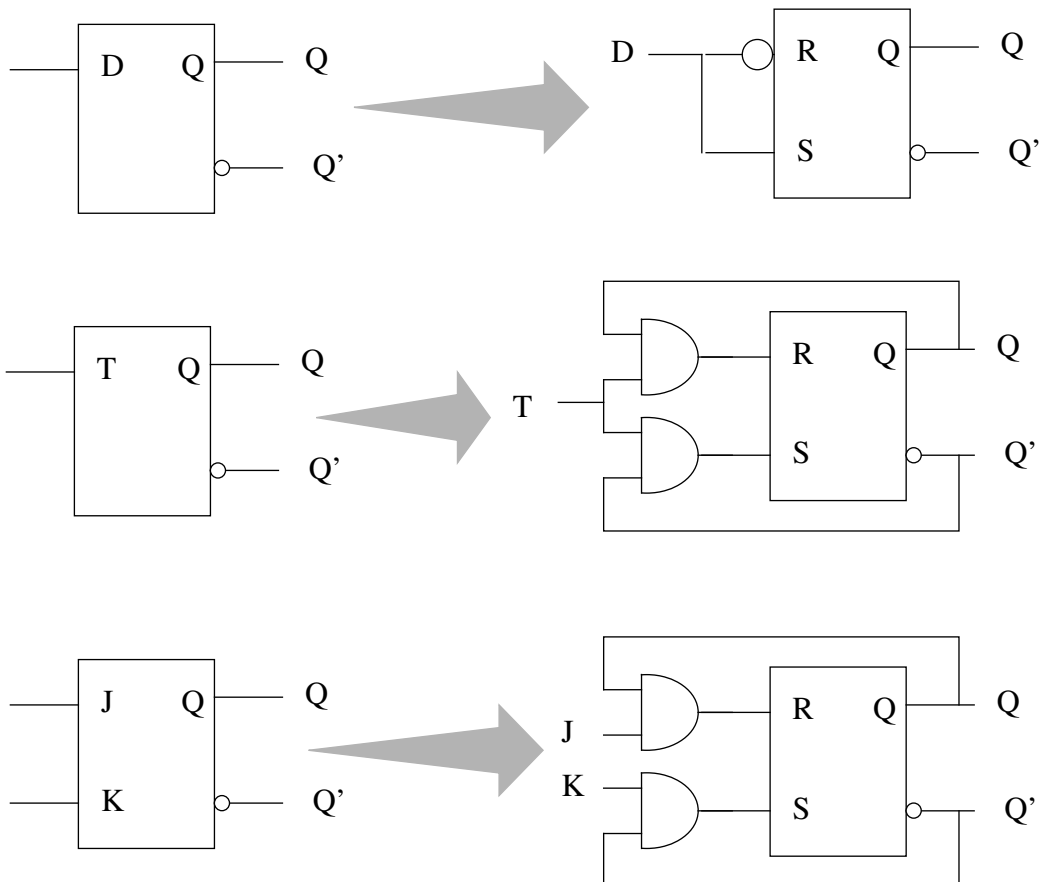
RS  
 $R = T \cdot q$   
 $S = T \cdot \bar{q}$

JK

		00	01	11	10
q	0	-0	-0	01	01
	1	0-	10	10	0-

RS  
 $R = K \cdot q$   
 $S = J \cdot \bar{q}$

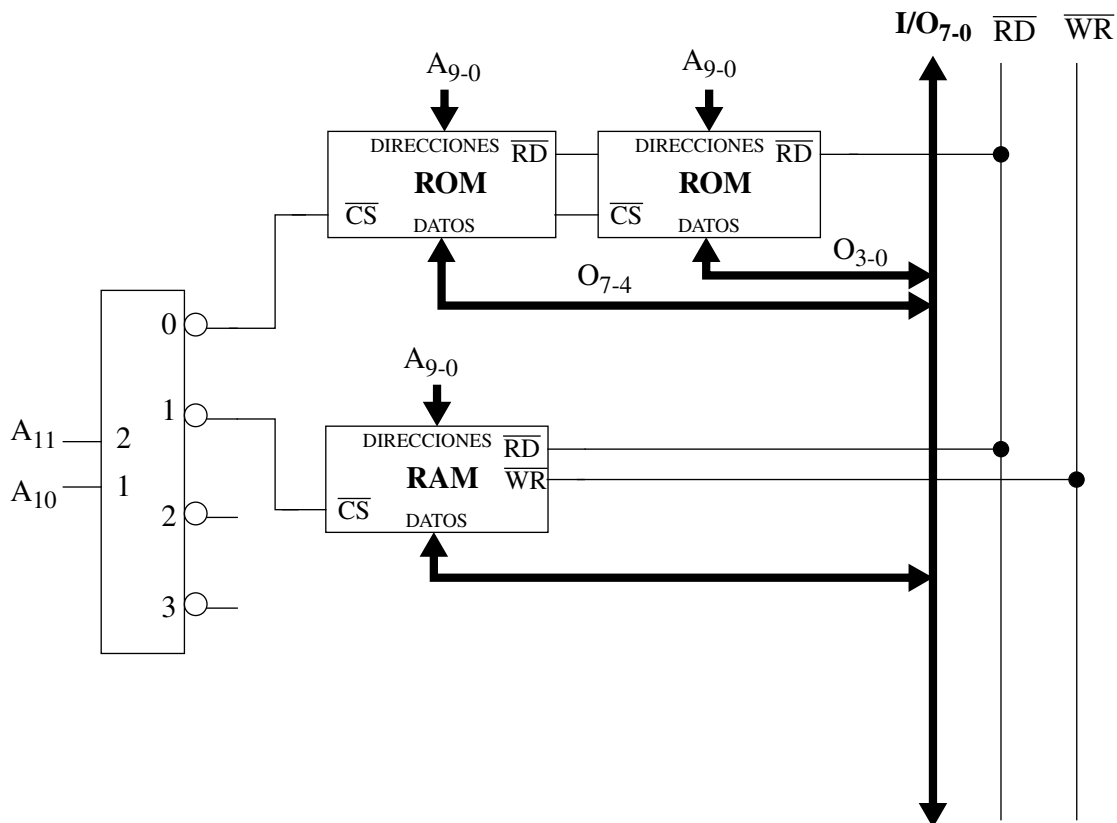
Luego:



- Biestables JK y T a partir de biestables D
- Biestables JK y D a partir de biestables T
- Biestables D y T a partir de biestables JK

3.5. Analizar el siguiente mapa de memoria. El análisis deberá indicar las siguientes características:

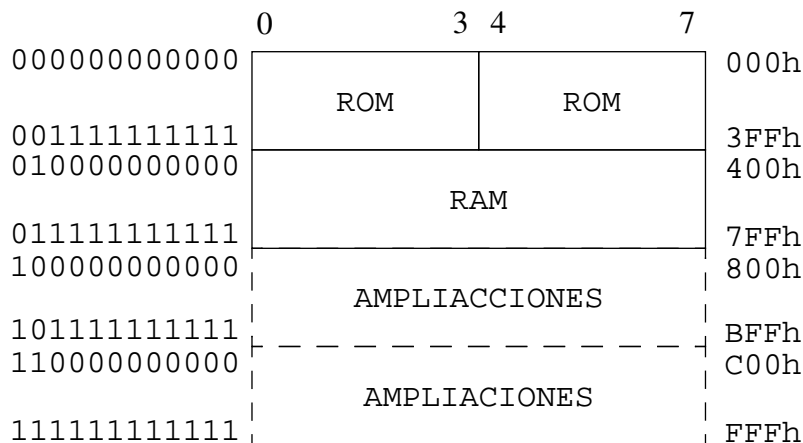
- Anchura del bus de datos
- Anchura del bus de direcciones
- Capacidad total del mapa de memoria
- Espacio de memoria destinado para los programas de inicialización
- Espacio de memoria destinado para datos y aplicaciones
- Espacio de memoria destinado para ampliaciones
- Tamaño de los diferentes módulos utilizados.
- Dibujar la configuración del mapa de memoria correspondiente.



- Anchura del bus de datos: Es el número de bits de datos que están conectados en paralelo, es decir, 8 bits
- Anchura del bus de direcciones: Es el número de bits de dirección diferentes, independientemente a donde estén conectados, es decir, 12 bits (de 0 a 9, 10 y 11)
- Capacidad total del mapa de memoria: Es el número total de palabras que podemos codificar con todos los bits de direcciones, es decir,  $2^{12} = 4$  Kpalabras
- Espacio de memoria destinado para los programas de inicialización: Es el número total de palabras almacenadas en memorias de solo lectura (módulos ROM). Como sólo tenemos una fila de módulos con 10 bits de direcciones, este espacio será:  $1 \times 2^{10} = 1$  Kpalabras
- Espacio de memoria destinado para datos y aplicaciones: Es el número total de palabras almacenadas en memorias de lectura-escritura (módulos RAM), es decir,  $1 \times 2^{10} = 1$  Kpalabras
- Espacio de memoria destinado para ampliaciones: Es la cantidad de memoria que no es utilizada, es decir,  $4K - 1K(ROM) - 1K(RAM) = 2K$
- Tamaño de los diferentes módulos utilizados: Es

el número de palabras que tiene cada módulo por el número de bits de cada palabra, es decir,

- Módulos ROM: 1K x 4
- Módulos RAM: 1K x 8
- Dibujar la configuración del mapa de memoria correspondiente: Como tenemos 12 bits de direcciones, nuestras palabras irán desde 000000000000 (000h) hasta 111111111111 (FFFh). Luego el mapa será el siguiente:



3.6. Diseñar un mapa de memoria con las siguientes características:

- Bus de datos de 8 bits
- Bus de direcciones de 14 bits
- Almacenar programas de inicialización de un tamaño máximo de 3 kp
- Almacenar aplicaciones y sistema operativo de un tamaño máximo de 10kp
- Se dispone de módulos de cualquier tipo de un tamaño igual a 2kx4.

En primer lugar vamos a realizar un esquema del mapa de memoria solicitado, para lo cual debemos obtener los siguientes datos:

- N° de módulos en paralelo: Como el bus de datos es de 8 bits y nuestros módulos tienen una palabra de 4 bits, el número de éstos conectados en paralelo deben ser de 2 (=8/4)
- N° de módulos en serie y tipos: El número de bits de direcciones es de 14, luego tendremos un total de 16 Kpalabras, dispuestas módulos de 2Kpalabras.

Además nos hacen falta 3Kpalabras para inicialización (módulos ROM), luego necesitaremos 2(=3/2), de los cuales la mitad de uno no será utilizada.

También nos hacen falta 10Kpalabras para aplicaciones (módulos RAM), luego necesitaremos 5 (=10/2).

El resto del espacio será destinado a posibles aplicaciones.

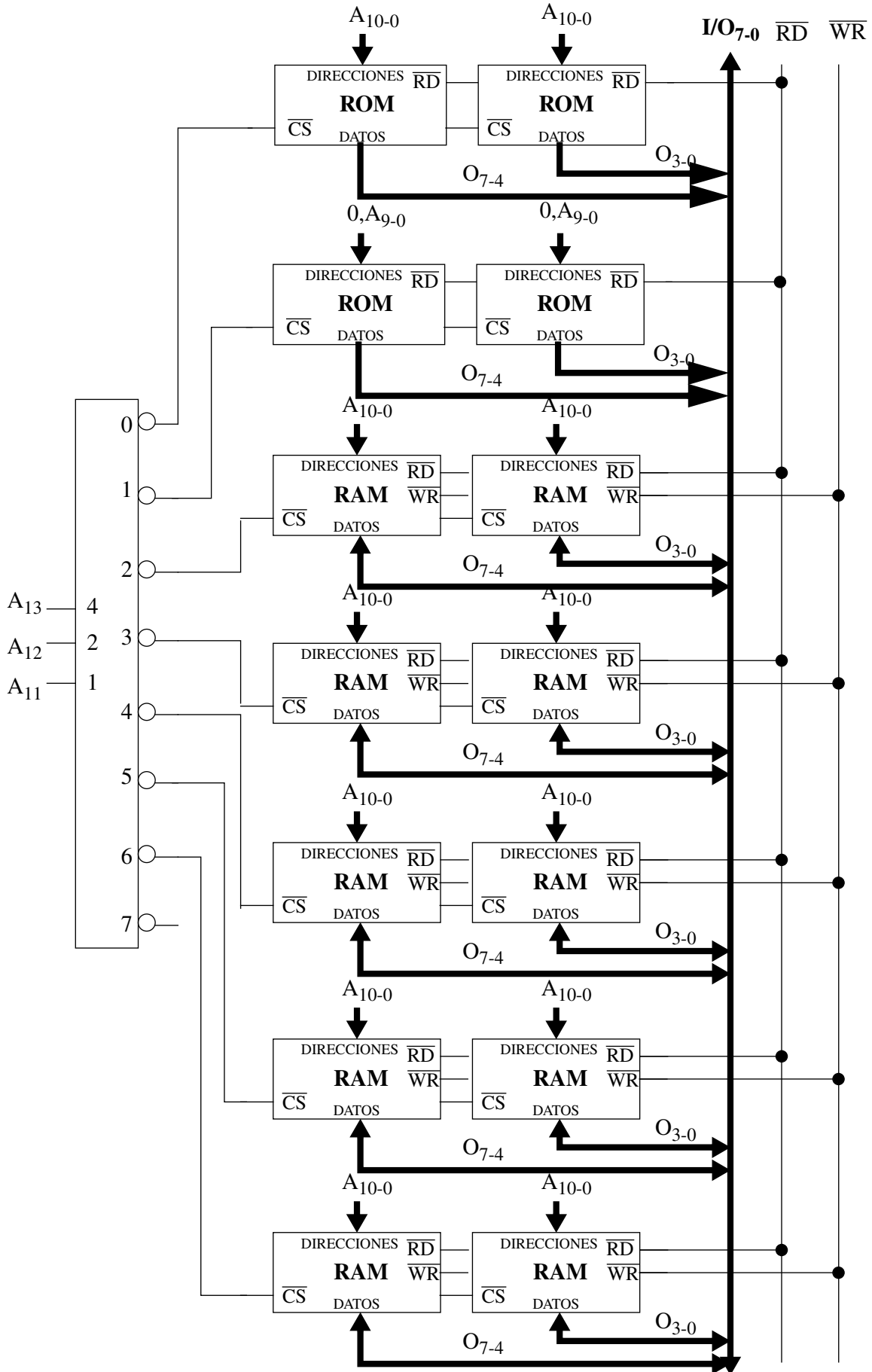
- N° de palabras de cada módulo: Cada módulo tiene 2 Kpalabras, luego irá de la dirección 0000h hasta la dirección 07FFh (11 bits de dirección)

Luego la configuración del mapa de memoria será el siguiente:

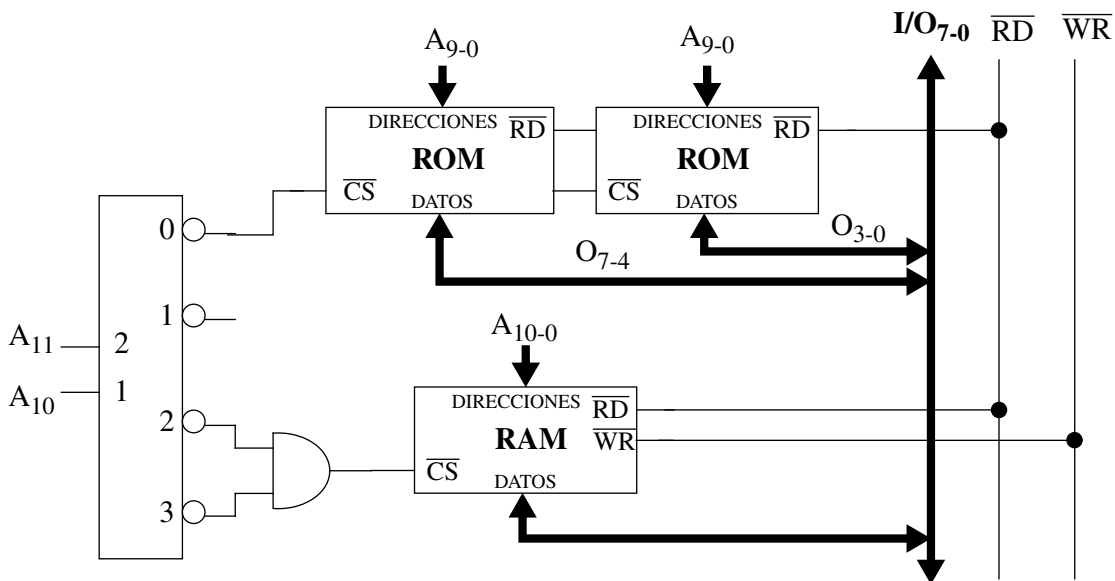
0	3 4	7	
ROM	ROM		0000h
ROM	ROM		07FFh
AMPLIACIONES ROM			0800h
RAM	RAM		0FFFh
RAM	RAM		1000h
RAM	RAM		17FFh
RAM	RAM		1800h
RAM	RAM		1FFFh
RAM	RAM		2000h
RAM	RAM		27FFh
RAM	RAM		2800h
RAM	RAM		2FFFh
RAM	RAM		3000h
AMPLIACIONES			37FFh
-			3800h
-			3FFFh

A la hora de implementar el mapa, además debemos identificar la codificación interna a los módulos y externa a través de un decodificador. Cada módulo tiene 11 bits de dirección (como ya hemos dicho); entonces para que la implementación se ajuste al mapa anterior debemos escoger los 11 bits menos significativos para la codificación interna, mientras que el resto, 3 bits, serán introducidos en un decodificador (3:8) para seleccionar los diferentes módulos.

Tendremos un total de cuatro módulos ROM (dispuestos en dos filas de dos módulos) y diez módulos RAM (dispuestos en cinco filas de dos módulos). Para dejar inaccesible, temporalmente la ampliación de memoria ROM conectaremos el bit de dirección más significativo de dicho módulo a '0'. Luego la implementación quedaría de la siguiente forma:



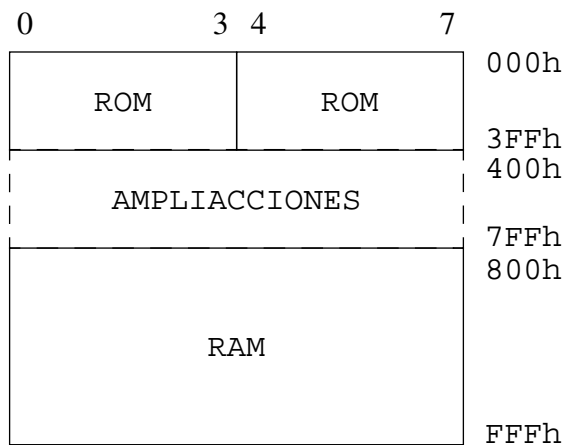
3.7. Determinar la configuración del mapa de memoria correspondiente a la siguiente implementación.



En la configuración del mapa vamos a realizar las siguientes consideraciones:

- Los módulos ROM tienen un tamaño de 1 Kpalabras, dispuestos en dos módulos en paralelo.
- El módulo RAM tiene un tamaño de 2 Kpalabras. Para que no exista ninguna incongruencia, debemos asegurarnos que el bit A<sub>10</sub> (que se encuentra en la decodificación interna y externa) sólo actúa en una de ellas (la interna). La puerta AND va a seleccionar el módulo RAM (CS̄ = '0'), cuando se seleccione las salidas 2 y 3 del decodificador, es decir, cuando A<sub>11</sub>A<sub>10</sub> = '10' ó '11', luego A<sub>11</sub>A<sub>10</sub> = '1X', por lo que en la decodificación externa el bit A<sub>10</sub> no realiza ninguna función al no importar su valor.

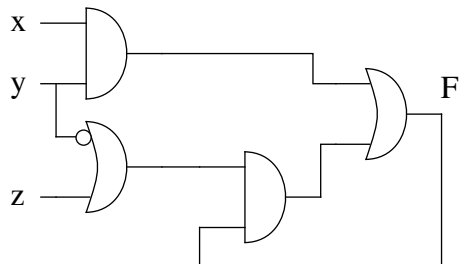
Luego la configuración será la siguiente:



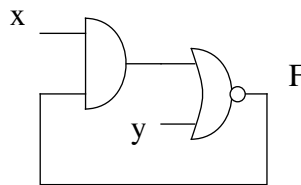


### Problemas del Tema 4

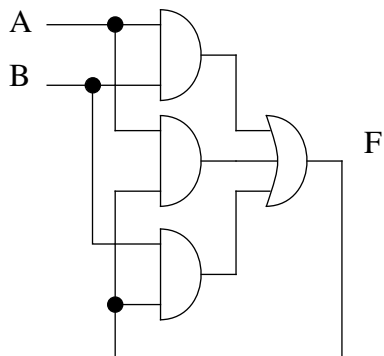
4.1. Analiza los siguientes circuitos secuenciales. Además del análisis, también se deberá completar el cronograma adjunto suponiendo que todos los elementos son ideales (en el sentido de que todas las características temporales son nulas).



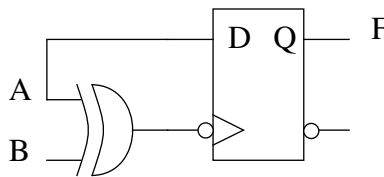
(a)



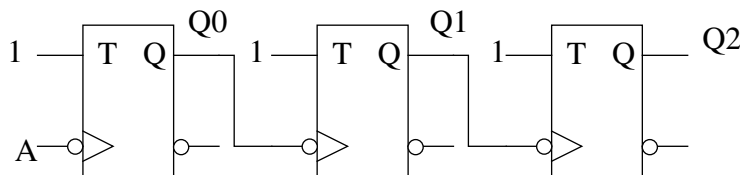
(b)



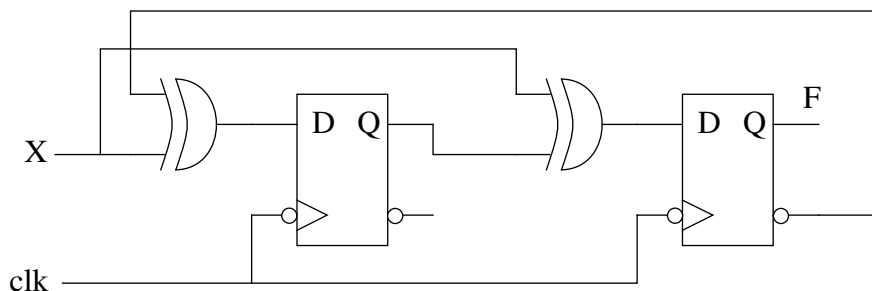
(c)



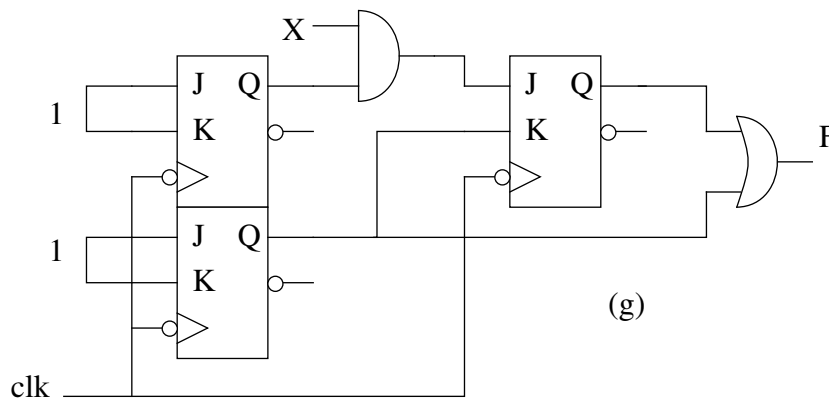
(d)



(e)

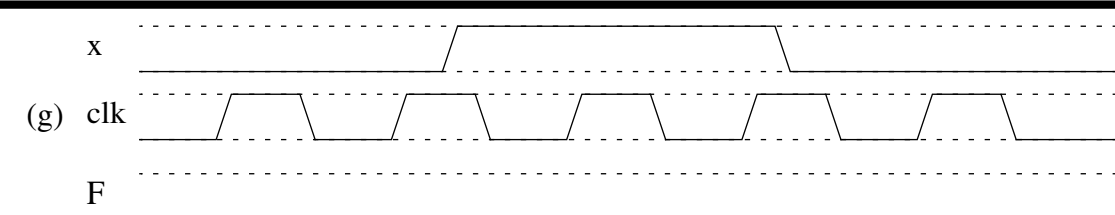
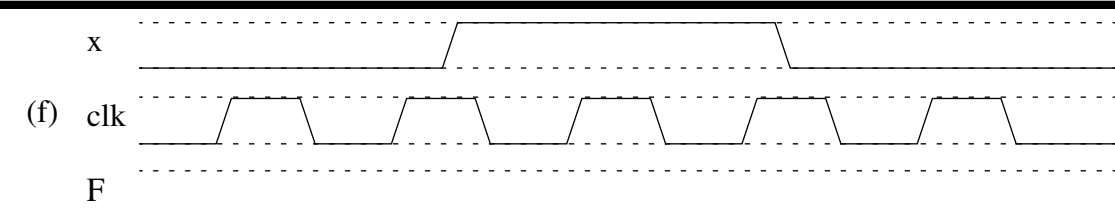
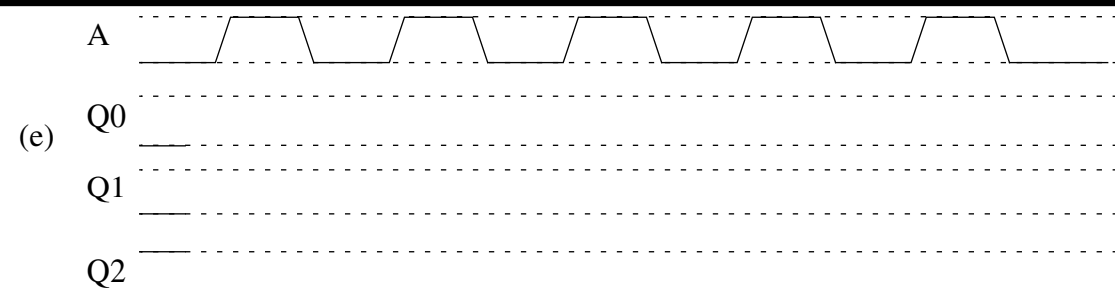
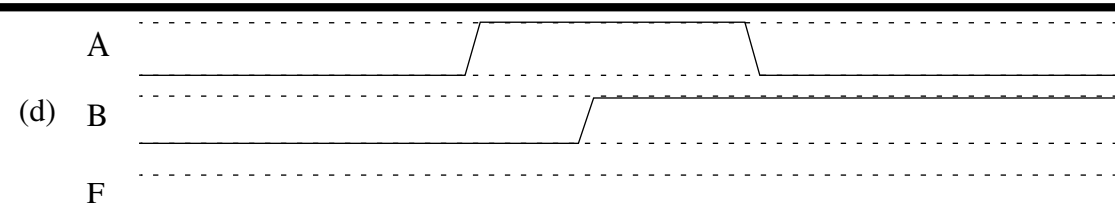
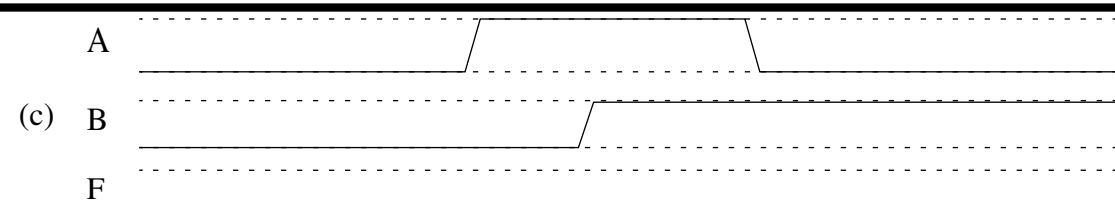
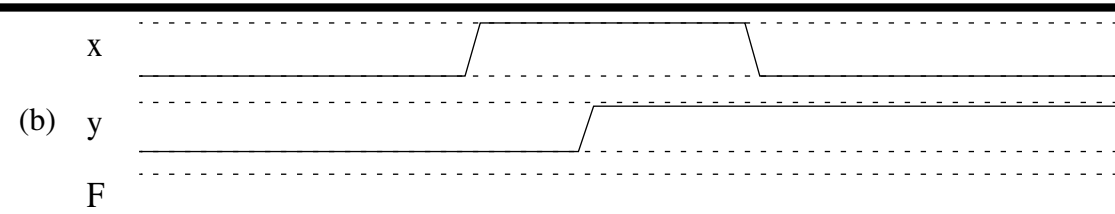
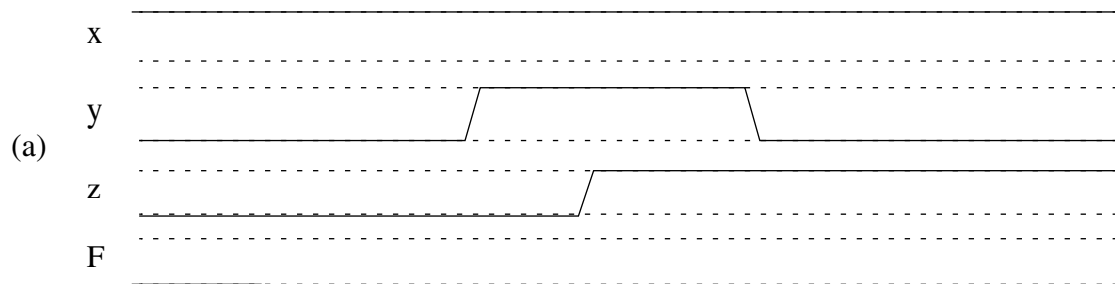


(f)

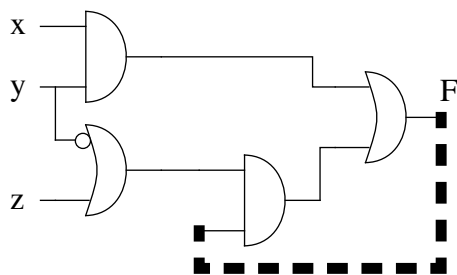


(g)

• Cronogramas



Circuito A



En primer lugar hay que identificar a qué categoría de circuitos secuenciales pertenece: síncronos o asíncronos. Como no existe ningún elemento de memoria, este circuito es un circuito asíncrono en modo fundamental.

Seguidamente hay que localizar todos los caminos de realimentación, ya que éstos nos indicarán el número de variables de estado que existen. En este caso particular, sólo existe un camino de realimentación por lo que sólo hay una variable de estado, que coincide con la señal de salida:

$$F = x \cdot y + f(z + \bar{y})$$

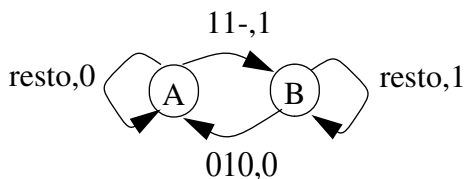
Una vez que conocemos las ecuaciones de excitación, que en estos casos coinciden con las de transición (salidas y próximos estados), generamos la tabla correspondiente:

		x y z							
		000	001	010	011	100	101	110	111
f	0	0	0	0	0	0	0	1	1
	1	1	1	0	1	1	1	1	1

F

A continuación hay que darles nombres a cada una de las combinaciones de las variables de estado. Seguidamente podemos observar la tabla y el diagrama de estados para la denominación adjunta.

		x y z							
		000	001	010	011	100	101	110	111
f = 0 -->	A	A,0*	A,0*	A,0*	A,0*	A,0*	A,0*	B,1	B,1
f = 1 -->	B	B,1*	B,1*	A,0	B,1*	B,1*	B,1*	B,1*	B,1*



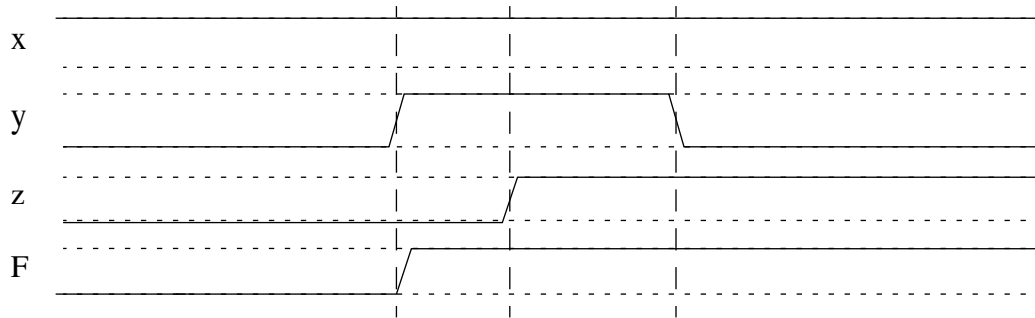
En cuanto a los problemas transitorios, vamos a estudiar ciclos entre estados, transiciones consecutivas y carreras.

Los ciclos entre estados se producen cuando existe alguna combinación de entrada para la cual no existe ningún estado estable. Por lo tanto, cuando se produzca un cambio en las señales de entrada, no se sabe desde qué estado se producirá dicho cambio. Para el caso particular que nos ocupa, podemos apreciar de la tabla de estado que todas las combinaciones muestran un estado estable, luego no existe ningún ciclo de estados.

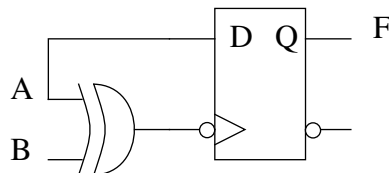
Las transiciones consecutivas son problemáticas cuando no se les da el tiempo necesario para estabilizar el nuevo estado. Como el número máximo de cambios de estado para un solo cambio de entrada es uno (de A hasta B), el tiempo que debemos dar para la estabilización es el retraso máximo del circuito, es decir:

$$\Delta = T_{OR} + T_{AND} + T_{OR}$$

Las carreras son producidas por el cambio simultáneo de dos o más variables de estado. Éstas pueden ser críticas, cuando podemos llegar a estados diferentes cuando variamos la variables que cambia antes, o no críticas, cuando el próximo estado es siempre el mismo. En este caso particular, sólo existe una variable de estado, luego no se pueden producir carreras. En cuanto al cronograma, sería el siguiente:



Circuito D



Este circuito pertenece a la categoría de circuitos asíncronos, ya que, aunque la secuencialidad sea proporcionada por un elemento de memoria, éste no es controlado por una señal externa a nuestro circuito.

En los circuitos que tienen elementos de memoria, el próximo estado no pasa a ser estado presente hasta que el elemento de memoria en cuestión sea activado por su señal de control. Por lo tanto, las señales de próximo estado son aquellas señales que se encuentran en las entradas (o una función lógica combinacional) de los elementos de memoria, mientras que las señales de estado presente son las salidas de los elementos de memoria.

Vamos a analizar con mayor detenimiento el elemento de memoria. Estamos frente a un flip-flop tipo D disparado por el flanco de bajada. Luego no se producirá ningún cambio de estado hasta que se produzca una bajada en la señal " $A \oplus B$ ". Luego en estos circuitos (en los que existen flip-flops disparados por flancos) no tiene sentido hablar de estados estables e inestables, ya que sólo se producirá una transición, como máximo, por cada cambio de entradas.

Para determinar el número de variables de estados, debemos contar el número de elementos de memoria utilizados, ya que cada elemento se corresponderá con una variable. Luego, estamos ante un sistema con una sola variable de estado.

Seguidamente obtendremos las ecuaciones de excitación, es decir, las salidas y las entradas de los elementos de memorias:

$$F = q ; D = A$$

A la hora de contruir la tabla de excitación, tenemos que tener en cuenta que estamos ante un circuito asíncrono, y por lo tanto no se van a permitir cambios múltiples de señales de entrada; luego siempre que lleguemos a las combinaciones de entrada "00" y "11" se producirá un cambio de estado. Las únicas situaciones en las que esto no sería verdad son cuando forzamos transiciones entre ambas combinaciones ( " $A \oplus B$ " permanecería a '0' y no se produciría la transición), pero se tratan de transiciones múltiples que hemos prohibido. Por lo tanto, la tabla de excitación sería:

		A B			
		00	01	11	10
q	0	0, 0	0, 0	1, 0	1, 0
	1	0, 1	0, 1	1, 1	1, 1

D, F

Como estamos ante un flip-flop tipo D, en el que la ecuación es  $Q = D$ , la tabla de transición es:

		A B			
		00	01	11	10
q	0	0, 0	0, 0	1, 0	0, 0
	1	0, 1	1, 1	1, 1	1, 1

Q, F

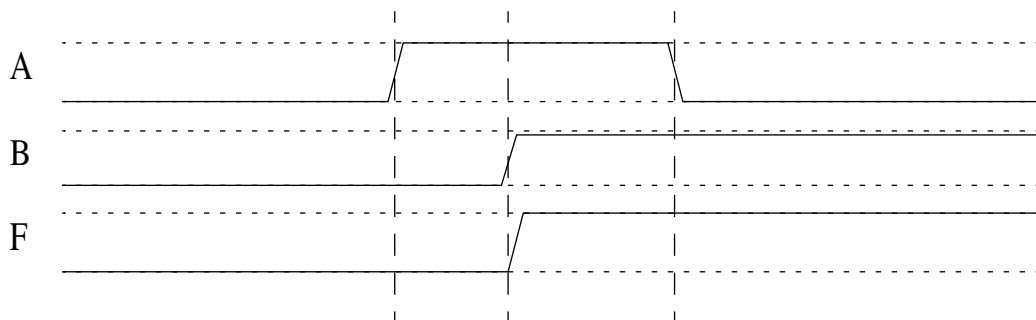
Luego las tablas de estados son (puestas como máquina de Mealy y de Moore):

		A B				
		00	01	11	10	
q = 0 -->	A	A, 0	A, 0	B, 1	A, 0	Mealy
q = 1 -->	B	A, 0	B, 1	B, 1	B, 1	

		A B				F	
		00	01	11	10		
q = 0 -->	A	A	A	B	A	0	Moore
q = 1 -->	B	A	B	B	B	1	

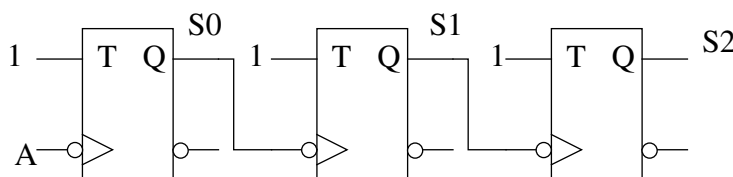
En estos circuitos, en los que los cambios de estado se producen durante un instante (la transición de alguna señal) no existen los problemas de sincronización estudiados (ciclos de estados, transiciones consecutivas y carreras) si el control de los flip-flops no depende de las señales de estado presente, como es este caso.

Por último el cronograma será el siguiente:



Este cronograma es correcto en el caso supuesto, es decir, considerando las restricciones temporales nulas. Pero en el caso de que el tiempo de setup fuese mayor que el retraso de la puerta XOR, el flip-flop podría no almacenar el valor correcto de la variable A.

Circuito E



En este caso nos encontramos ante un circuito asíncrono ya que esta formado por elementos de memoria, pero se encuentran controlados con señales diferentes y además son internas al circuito.

Los elementos de memoria, flip-flops, son controlados por la bajada de la señal de salida del flip-flop que le precede, excepto el primero que es controlado por una señal externa. Así, el flip-flop 1 cambiará cuando se produzca la transición:

$$q_0 \rightarrow Q_0$$

$$'1' \rightarrow '0'$$

Como tenemos tres flip-flops, estamos ante un sistema en el que hay tres variables de estado.

Las funciones de excitación serán:

$$T_0 = '1'$$

$$T_1 = '1'$$

$$T_2 = '1'$$

Por lo tanto, las ecuaciones de transición serán:

$$Q_0 = T_0 \oplus q_0 = \bar{q}_0, \text{ cuando } A : '1' \rightarrow '0'$$

$$Q_1 = T_1 \oplus q_1 = \bar{q}_1, \text{ cuando } q_0 : '1' \rightarrow '0'$$

$$Q_2 = T_2 \oplus q_2 = \bar{q}_2, \text{ cuando } q_1 : '1' \rightarrow '0'$$

Luego si suponemos que el valor inicial de la señal A es '1', y los cambios sólo se permiten en las transiciones, la tabla de transición y de estado serán:

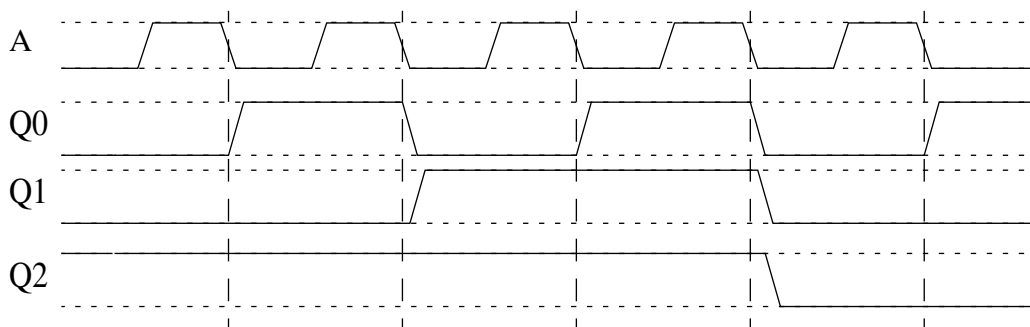
		A				A	
		0	1			0	1
$q_2q_1q_0$	000	001	000	→	A	B	A
	001	010	001	→	B	C	B
	010	011	010	→	C	D	C
	011	100	011	→	D	E	D
	100	101	100	→	E	F	E
	101	110	101	→	F	G	F
	110	111	110	→	G	H	G
	111	000	111	→	H	A	H

Con respecto a los problemas transitorios podemos observar.

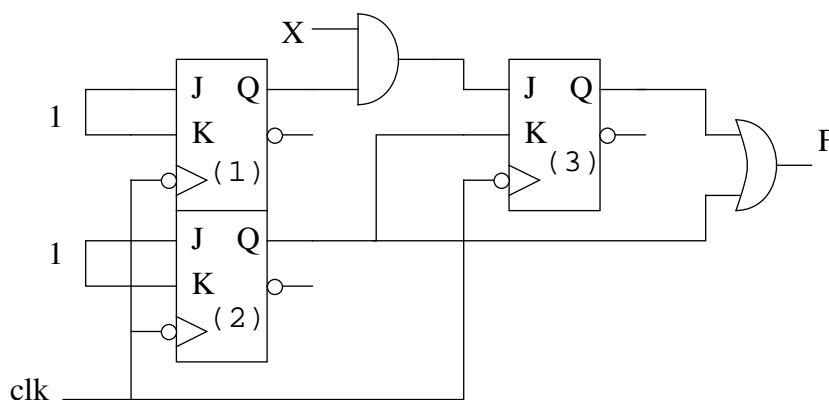
No existen ciclos de estados ya que estamos utilizando flip-flops, en el que el cambio se producirá en una transición, y por lo tanto sólo habrá un cambio para cada transición de entradas.

En este circuito existen carreras ya que existen cambios de estado en los que cambian más de una de estas señales; por ejemplo '111' --> '000'. Además son carreras críticas ya que al haber un solo cambio, el estado al que se llega es diferente según las señales que cambien antes.

El cronograma será el siguiente:



Circuito G



En este caso particular estamos ante un circuito compuesto por elementos de memoria controlados todos por la misma señal externa, luego estamos ante un circuito síncrono.

Los elementos de memoria son flip-flops disparados por el flanco de bajada, por lo tanto los cambios de estado sólo se producirán cuando la señal de control (clk) sufra una transición de bajada.

El circuito tiene tres flip-flops, por lo tanto, existirán tres variables de estado.

Las ecuaciones de excitación serán:



$$\begin{aligned}
 J_1 &= '1' & \dots & \dots & k_1 &= '1' \\
 J_2 &= '1' & \dots & \dots & k_2 &= '1' \\
 J_3 &= x \cdot q_1 & \dots & \dots & k_3 &= q_2 \\
 & & & & & F = q_3 + q_2
 \end{aligned}$$

Luego las ecuaciones de transición serán:

$$\begin{aligned}
 Q_1 &= J_1 \cdot \bar{q}_1 + \bar{k}_1 \cdot q_1 = \bar{q}_1 \\
 Q_2 &= J_2 \cdot \bar{q}_2 + \bar{k}_2 \cdot q_2 = \bar{q}_2
 \end{aligned}$$

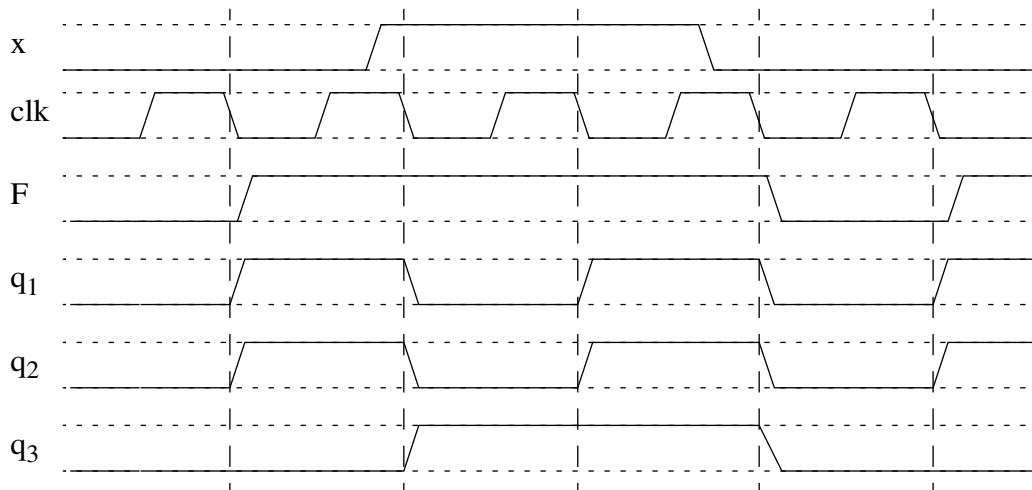
$$Q_3 = J_3 \cdot \bar{q}_3 + \bar{k}_3 \cdot q_3 = x \cdot q_1 \cdot \bar{q}_3 + \bar{q}_2 \cdot q_3$$

Por lo tanto, las tablas de transición y de estado serán:

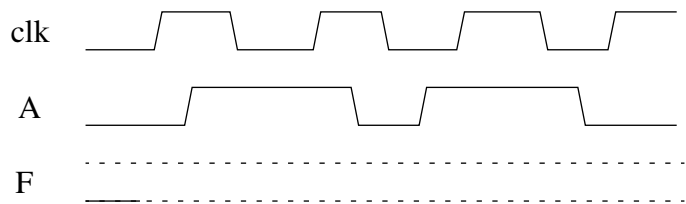
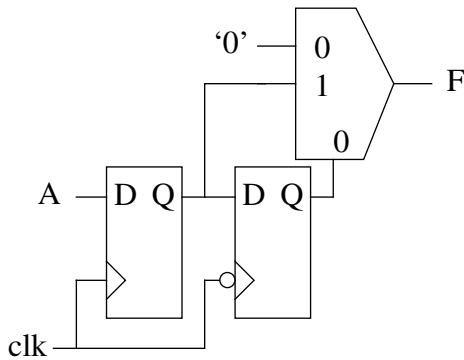
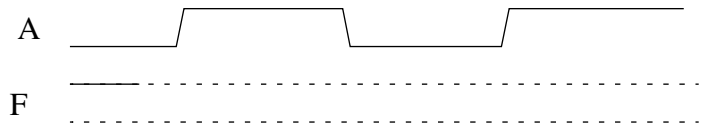
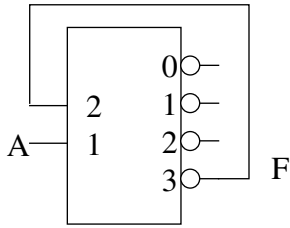
		x			x	
		0	1		0	1
$q_2 q_1 q_0$	000	011,0	011,0	→	A	D,0
	001	010,0	110,0	→	B	C,0
	010	001,1	001,1	→	C	B,1
	011	000,1	100,1	→	D	A,1
	100	111,1	111,1	→	E	H,1
	101	110,1	110,1	→	F	G,1
	110	001,1	001,1	→	G	B,1
	111	000,1	000,1	→	H	A,1

Con respecto a los problemas transitorios, debido a que el circuito es síncrono y sensible a la transición, no se pueden producir ni carreras ni ciclos de estados.

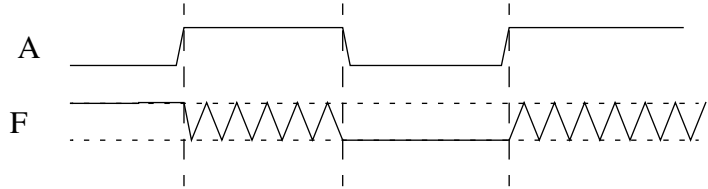
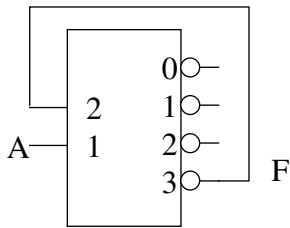
El cronograma será:



4.2. Analizar los siguientes circuitos y completa los cornogramas adjuntos (en el caso de que algunas señales internas no estén inicializadas en el cornograma, tomar el valor bajo como valor inicial).



Circuito 1



En este caso no existe ningún elemento de memoria, por lo que el circuito pertenece a la categoría de circuitos asíncronos del modo fundamental.

La ecuación lógica de transición es:

$$F = \overline{f} \cdot A$$

ya que la ecuación correspondiente a la salida 3 del decodificador es:  $S_3 = \overline{I_1} \cdot \overline{I_2}$ .

Las ecuaciones de excitación coinciden con la de transición. Por lo tanto, las tablas de transición y de estado serán:

	A			A	
	0	1		0	1
0	1	1	→	S1	S2,1
1	1	0	→	S2	S1,0

En este caso, al ser un circuito asíncrono en modo fundamental, hay que estudiar con detenimiento los problemas transitorios.

Podemos comprobar que existe un ciclo de estados cuando la entrada A vale '1'.

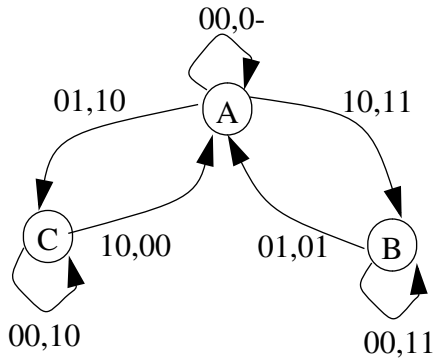
Por tener una sola variable de estado (debido a un solo camino de realimentación), no pueden existir carreras de ninguna clase.

### Problemas del Tema 5

5.1. Diseñe los circuitos cuyas especificaciones se dan en el problema 1 del tema 2, utilizando circuitos en modo fundamental.

Punto nº 1

De las soluciones del tema 2, el diagrama de estados es:



A: escalera parada  
 B: escalera bajando  
 C: escalera subiendo

señales de salida  
 A=0, motor parado  
 A=1, motor moviéndose  
 S=0, sentido hacia arriba  
 S=1, sentido hacia abajo

El siguiente paso consiste en determinar la máquina mínima que realiza dicho comportamiento. Para ello, hallamos la tabla de estados y la tabla de pares compatibles:

	P <sub>1</sub> P <sub>2</sub>			
	00	01	11	10
A	A, 0-	C, 10	-, --	B, 11
B	B, 11	A, 01	-, --	-, --
C	C, 10	-, --	-, --	A, 00

	A		
	X		B
	X	X	
			C

A pesar de que estamos ante una máquina incompletamente especificada, podemos observar que ninguno de los estados son compatibles por lo que estamos ante la máquina mínima.

Este sistema no muestra una sincronización sencilla, ya que no existe ninguna relación para los cambios de las señales de entrada. Por lo tanto, vamos a implementarlo con un sistema asíncrono en modo fundamental; luego tenemos que procurar que el asignamiento de estados produzcan el menor número de problemas transitorios, es decir, se deben cumplir la mayoría de las reglas de adyacencia.

Las reglas de adyacencia serán:

- A debe ser adyacente a B y C
- B debe ser adyacente a A
- C debe ser adyacente a A

Al tener tres estados, necesitaremos dos variables de estados ya que  $(2^1 < 3 < 2^2)$ . Por lo tanto, una posi-

ble codificación que garantiza todas las reglas de adyacencia sería la siguiente:

		$q_1$	
		0	1
$q_0$	0	A	B
	1	C	-

Por lo tanto, la tabla de transición será:

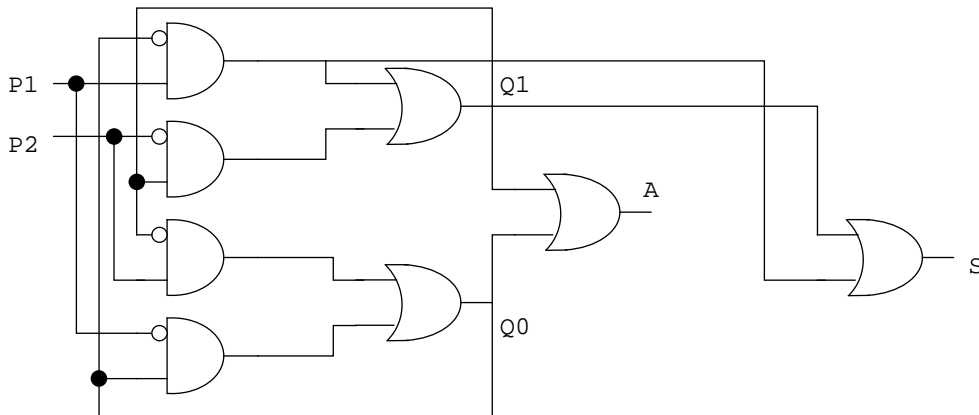
		$P_1 P_2$				
		00	01	11	10	
$q_1 q_0$	00	00, 0-	01, 10	--, --	10, 11	$Q_1 Q_0, AS$
	01	01, 10	--, --	--, --	00, 00	
	11	--, --	--, --	--, --	--, --	
	10	10, 11	00, 01	--, --	--, --	

En el caso de los circuitos asíncronos en modo fundamental, la tabla de transición y de excitación coinciden. Por lo tanto, vamos a obtener las ecuaciones lógicas de las salidas de la función:

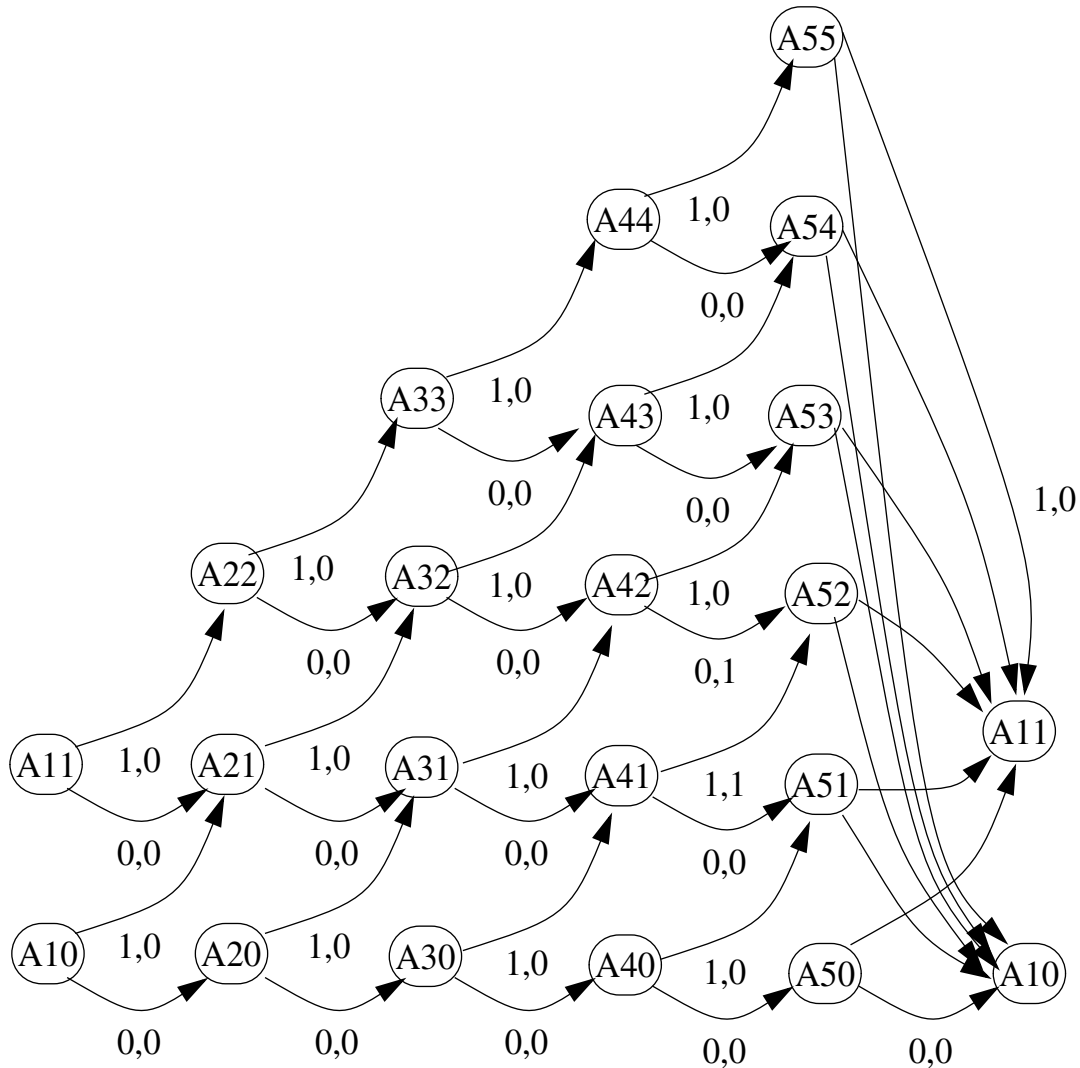
- $Q_1 = \sum m(2,8) + \phi (3,5,7,10,11,12,13,14,15)$
- $Q_0 = \sum m(1,4) + \phi (3,5,7,10,11,12,13,14,15)$
- $A = \sum m(1,2,4,8) + \phi (3,5,7,10,11,12,13,14,15)$
- $S = \sum m(2,6,8) + \phi (0,3,5,7,10,11,12,13,14,15)$

En principio, al tratarse de un función multisalida, la implementación lógica habría que realizarla utilizando el método de McCluskey; no obstante, y para reducir el tiempo, vamos a tratarlo como funciones independientes, y utilizaremos el método del mapa de Karnaugh.

- $Q_1 = \bar{P}_2 \cdot q_1 + P_1 \cdot \bar{q}_0$
- $Q_0 = \bar{P}_1 \cdot q_0 + P_2 \cdot \bar{q}_1$
- $A = \bar{P}_1 \cdot q_0 + \bar{P}_2 \cdot q_1 + P_2 \cdot \bar{q}_1 + P_1 \cdot \bar{q}_0$
- $S = P_1 \cdot \bar{q}_0 + q_1 \dots \text{ ó } \dots S = \bar{P}_2 \cdot \bar{q}_0 + q_1$



## Punto nº 2



Segun el problema debemos indicar cuando una palabra que nos llega pertenece al código "2-out-of-5" y cuando no pertenece. Por lo tanto, tenemos que separar los bits que nos llegan en grupos de 5. Luego, como ya comentamos en el tema II, la máquina secuencial que modela este comportamiento debe contar el número de 1's que han llegado según el número de bits que nos han llegado. Así cuando llegue el quinto bit, sabremos si han llegado dos 1's o un número diferente de 1's, que es lo que determina si la palabra es correcta o no.

Una vez que hemos obtenido el diagrama de estados que modela la máquina secuencial, debemos minimizarla. Para determinar la máquina mínima vamos a obtener la tabla de estados, y de ella la tabla de pares compatibles, y por último obtener los máximos compatibles.

		X																				
		0	1																			
A10	A20,0	A21,0	A20-A21 A21-A22	A10																		
A11	A21,0	A22,0	A20-A30 A21-A31	A11	A20																	
A20	A30,0	A31,0	A20-A31 A21-A32	A20	A21																	
A21	A31,0	A32,0	A20-A32 A21-A33	A20	A21	A22																
A22	A32,0	A33,0	A20-A40 A21-A41	A20	A21	A22	A23															
A30	A40,0	A41,0	A20-A41 A21-A42	A20	A21	A22	A23	A24														
A31	A41,0	A42,0	A20-A42 A21-A43	A20	A21	A22	A23	A24	A25													
A32	A42,0	A43,0	A20-A43 A21-A44	A20	A21	A22	A23	A24	A25	A26												
A33	A43,0	A44,0	A20-A50 A21-A51	A20	A21	A22	A23	A24	A25	A26	A27											
A40	A50,0	A51,0	X X	X	X	X	X	X	X	X	X	X	A41									
A41	A51,0	A52,1	X X	X	X	X	X	X	X	X	X	X	X	A42								
A42	A52,1	A53,0	A20-A53 A20-A54	A20	A21	A22	A23	A24	A25	A26	A27	A28	A29	A43								
A43	A53,0	A54,0	A20-A54 A21-A55	A20	A21	A22	A23	A24	A25	A26	A27	A28	A29	A30	A44							
A44	A54,0	A55,0	A20-A10 A21-A11	A20	A21	A22	A23	A24	A25	A26	A27	A28	A29	A30	A31	A45						
A50	A10,0	A11,0	A20-A10 A21-A11	A20	A21	A22	A23	A24	A25	A26	A27	A28	A29	A30	A31	A32	A46					
A51	A10,0	A11,0	A20-A10 A21-A11	A20	A21	A22	A23	A24	A25	A26	A27	A28	A29	A30	A31	A32	A33	A47				
A52	A10,0	A11,0	A20-A10 A21-A11	A20	A21	A22	A23	A24	A25	A26	A27	A28	A29	A30	A31	A32	A33	A34	A48			
A53	A10,0	A11,0	A20-A10 A21-A11	A20	A21	A22	A23	A24	A25	A26	A27	A28	A29	A30	A31	A32	A33	A34	A35	A49		
A54	A10,0	A11,0	A20-A10 A21-A11	A20	A21	A22	A23	A24	A25	A26	A27	A28	A29	A30	A31	A32	A33	A34	A35	A36	A50	
A55	A10,0	A11,0	A20-A10 A20-A11	A20	A21	A22	A23	A24	A25	A26	A27	A28	A29	A30	A31	A32	A33	A34	A35	A36	A37	A51

Estados ..... Clases de estados compatibles .....Lista de máximos compatibles

- A55 ..... - ..... A55
  - A54 ..... A55 ..... A55A54
  - A53 ..... A54, A55 ..... A55A54
  - A52 ..... A53, A54, A55 ..... A55A54A53A52
  - A51 ..... A52, A53, A54, A55 ..... A55A54A53A52A51
  - A50 ..... A51, A52, A53, A54, A55 ..... A55A54A53A52A51A50
  - A44 ..... - ..... A5, A44
  - A43 ..... A44 ..... A5, A44A43
  - A42 ..... - ..... A5, A44A43, A42
  - A41 ..... - ..... A5, A44A43, A42, A41
  - A40 ..... A43, A44 ..... A5, A4, A42, A41
  - A33 ..... - ..... A5, A4, A42, A41, A33
  - A32 ..... - ..... A5, A4, A42, A41, A33, A32
  - A31 ..... - ..... A5, A4, A42, A41, A33, A32, A31
  - A30 ..... - ..... A5, A4, A42, A41, A33, A32, A31, A30
  - A22 ..... - ..... A5, A5, A42, A41, A33, A32, A31, A30, A22
  - A21 ..... - ..... A5, A4, A42, A41, A33, A32, A31, A30, A22, A21
  - A20 ..... - ..... A5, A4, A42, A41, A33, A32, A31, A30, A22, A21
- 
- A11 ..... - ..... A5, A4, A42, A41, A33, A32, A31, A30, A22, A21
  - ..... A20, A11
- 
- A10 ..... - ..... A5, A4, A42, A41, A33, A32, A31, A30, A22, A21
  - ..... A20,A11, A10

Por lo tanto los máximos compatibles son: A5, A4, A41, A42, A33, A32, A31, A30, A22, A21, A20, A11 y A10, donde:

- A5 = A55A54A53A52A51A50
- A4 = A44A43A40

Seguidamente podemos observar que nos hacen falta un total de 4 variables de estados ( $2^3 < 13 \leq 2^4$ ). Así, previo al asignamiento de estados, las reglas de adyacencia que se deben cumplir son:

- A10 debe ser adyacente a A20 y A21
- A11 debe ser adyacente a A21 y A22
- A20 debe ser adyacente a A30 y A31
- A21 debe ser adyacente a A31 y A32
- A22 debe ser adyacente a A32 y A33
- A30 debe ser adyacente a A4 y A41
- A31 debe ser adyacente a A41 y A42
- A32 debe ser adyacente a A42 y A4
- A33 debe ser adyacente a A4
- A4 debe ser adyacente a A5
- A41 debe ser adyacente a A5
- A42 debe ser adyacente a A5
- A5 debe ser adyacente a A10 y A11

En este caso particular se comprueba que no se puede hallar una codificación que garantice todas las reglas de adyacencia. Esto es debido a que con cuatro bits, un estado sólo puede ser adyacente a otros cuatro; no obstante, el estado A5 debe ser adyacente a 5 (A10, A11, A4, A41 y A42). Así una posible codificación se muestra a continuación:

		$q_3 q_2$			
		00	01	11	10
$q_1 q_0$	00	A42	A31	A20	A30
	01	A32	A21	A10	-
	11	A22	A11	A5	A41
	10	A33	-	A4	-

A continuación, con esta codificación pasamos a la tabla de transición:

		$X q_3 q_2$								
		000	001	011	010	110	111	101	100	
$q_1 q_0$	00	1111,1	1011,0	1000,0	1110,0	1011,0	0100,0	0000,0	1111,0	
	01	0000,0	0100,0	1100,0	-	-	0101,0	0001,0	1110,0	$Q_3 Q_2 Q_1 Q_0, F$
	11	0001,0	0101,0	1101,0	1111,0	1111,1	0111,0	0111,0	0010,0	
	10	1110,0	-	1111,0	-	-	1111,0	-	1110,0	



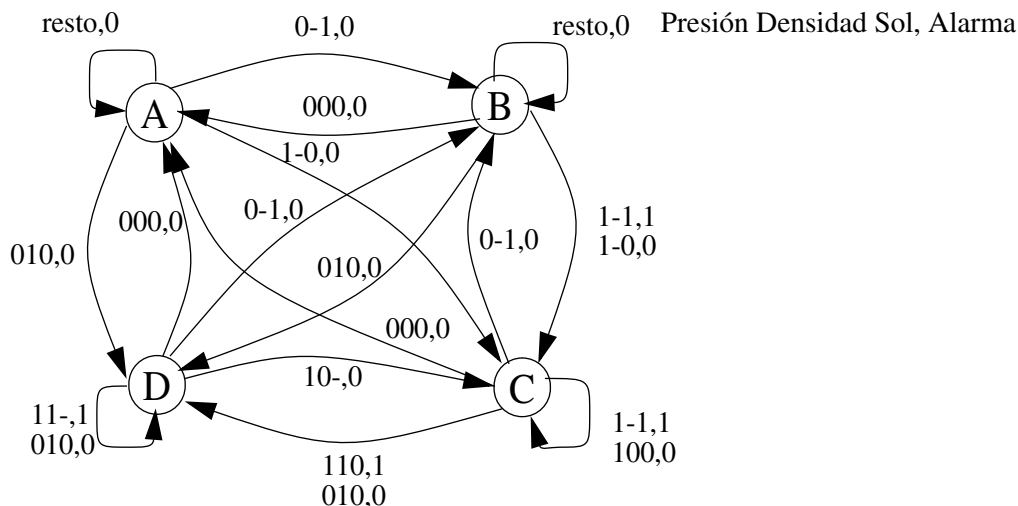
Al tratarse de una máquina en modo fundamental, las tablas de transición y de excitación coinciden, luego la función que hay que implementar es:

- $Q_3 = \sum m(0, 2, 4, 8, 11, 12, 13, 14, 15, 16, 17, 18, 24, 27, 30) + \phi (6, 9, 10, 22, 25, 26)$
  - $Q_2 = \sum m(0, 2, 5, 7, 8, 11, 13, 14, 15, 16, 17, 18, 23, 27, 28, 29, 30, 31) + \phi (6, 9, 10, 22, 25, 26)$
  - $Q_1 = \sum m(0, 2, 4, 8, 11, 14, 16, 17, 18, 19, 23, 24, 27, 30, 31) + \phi (6, 9, 10, 22, 25, 26)$
  - $Q_0 = \sum m(0, 3, 4, 7, 11, 14, 15, 16, 21, 23, 24, 27, 29, 30, 31) + \phi (6, 9, 10, 22, 25, 26)$
  - $F = \sum m(0, 27) + \phi (6, 9, 10, 22, 25, 26)$
- Sistema de alarma de una planta industrial. En la planta hay elementos inflamables según las siguientes condiciones (existen sensores que nos indican cuando la presión, densidad y la exposición al sol han superado su límite):
    - Si la presión del contenedor es igual o superior a 4 unidades, después de haber estado expuesto al sol
    - Si la densidad del contenedor es igual o superior a 5 unidades, después de estar sometido a una presión mayor o igual a 4 unidades.

En esta máquina existirán tres señales de entrada, correspondientes a los sensores de densidad, presión y exposición al sol (P, D, E) y una alarma (A='1', cuando exista peligro).

Los estados por los que puede pasar nuestro sistema son:

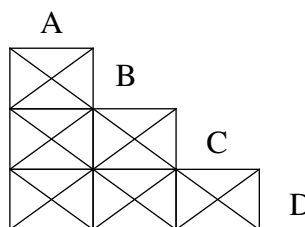
- Ni estar expuesto al sol, ni la presión es mayor o igual que 4 unidades (A)
- Estar expuesto al sol con una presión menor a 4 unidades (B)
- Tener una presión mayor o igual que 4 unidades (C)
- Tener una densidad mayor o igual a 5 unidades (D)



Al realizar este diagrama hemos tenido que tomar una serie de decisiones, ya que las especificaciones de partida no determinaban completamente el comportamiento. En primer lugar hemos supuesto que una vez cumplida la primera condición (estar expuesto al sol o tener una presión con una presión mayor o igual a 4 unidades), ésta se debe mantener para la siguiente condición (tener una presión mayor o igual a 4 unidades, o tener una densidad mayor o igual a 5 unidades, respectivamente). Además también se ha considerado que no se pueden producir cambios múltiples de señales de entradas.

El siguiente paso será reducir la máquina de estados. Para lo cual la tabla de estados y de pares compatibles son:

	000	001	010	011	100	101	110	111
A	A,0	B,0	D,0	B,0	C,0	A,0	C,0	A,0
B	A,0	B,0	D,0	B,0	C,0	C,1	C,0	C,1
C	A,0	B,0	D,0	B,0	C,0	C,1	D,1	C,1
D	A,0	B,0	D,0	B,0	C,0	C,0	D,1	D,1



Luego la máquina de partida ya es mínima.

El siguiente paso es el asignamiento de estado, para lo cual, las reglas de adyacencia que se deben cumplir son:

- A debe ser adyacente a B, D y C
- B debe ser adyacente a A, D y C
- C debe ser adyacente a A, B y D
- D debe ser adyacente a A, B y C

Para cumplir todas las reglas de adyacencia necesitaríamos un mínimo de tres señales de estado (ya que un estado debe ser adyacente a tres). No obstante, el número mínimo de estas señales es 2 ( $2^2$ ). Así que, de nuevo, no se pueden cumplir todas las reglas. Por lo tanto una posible codificación es:

		$q_1$	
		0	1
$q_0$	0	A	C
	1	B	D

Por lo tanto, la tabla de transición será:

PDS

		000	001	011	010	110	111	101	100
$q_1q_0$	00	00,0	01,0	01,0	11,0	10,0	00,0	00,0	10,0
	01	00,0	01,0	01,0	11,0	10,0	10,1	10,1	10,0
	11	00,0	01,0	01,0	11,0	11,1	11,1	10,0	10,0
	10	00,0	01,0	01,0	11,0	11,1	10,1	10,1	10,0

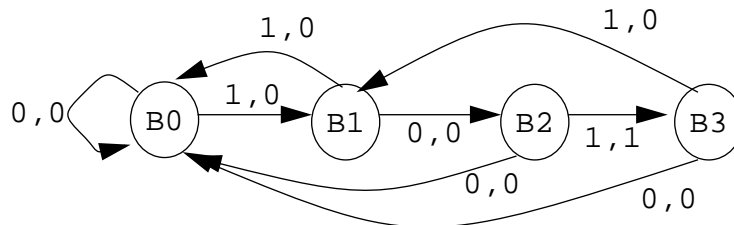
$Q_1 Q_0, F$

Luego la función que hay que implementar es la que tiene como salidas:

- $Q_1 = \sum m(8, 9, 10, 11, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 27, 29, 30, 31)$
- $Q_0 = \sum m(4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 26, 27, 31)$
- $F = \sum m(21, 22, 26, 27, 29, 30, 31)$

5.2. Diseñe los mismos circuitos de forma síncrona, utilizando biestables tipo T.

- Sistema de detección de fin de cadena en un equipo de telecomunicaciones. El sistema deberá detectar la cadena "101".



En este caso tendremos dos señales de entrada: una señal de entrada de datos, por el que nos llegarán los diferentes bits de la cadena; y una señal de reloj, que determinará la llegada de los diferentes bits de la cadena. Para una mayor simplicidad en el diseño, vamos a sincronizar la llegada de los bits con el flanco de subida del reloj, ya que de esta forma evitaremos la mayoría de problemas de sincronización. Como el valor de la señal de reloj no influye en el valor de la salida, esta señal no aparecerá en el proceso del diseño hasta que no dibujemos el circuito, y por tanto, la conectemos a las señales de control de los flip-flops.

El siguiente paso consiste en reducir la máquina secuencial. Así que las tablas de estado, pares compatibles y máximos compatibles son:

	0	1				
B0	B0,0	B1,0	<del>B0-B2</del> <del>B1-B0</del>	B1		
B1	B2,0	B0,0	X	X	B2	
B2	B0,0	B3,1				
B3	B0,0	B1,0	~	<del>B0-B2</del> <del>B1-B0</del>	X	B3

E ..... C.E.C. .... M.C.  
 B3 ..... - ..... B3  
 B2 ..... - ..... B3, B2  
 B1 ..... - ..... B3, B2, B1  
 B0 .....B3 ..... B, B2, B1

	0	1
B	B,0	B1,0
B1	B2,0	B,0
B2	B,0	B,1

Una vez que tenemos la tabla mínima, el siguiente paso es el asignamiento de estados. Para ello, las reglas de adyacencia que se deben cumplir son:

- B debe ser adyacente con B1
- B1 debe ser adyacente con B2 y B
- B2 debe ser adyacente con B

Así, una posible codificación es:

		$q_1$	
		0	1
$q_0$	0	B	B1
	1	-	B2

En este caso no es crítico que no se cumplan las reglas de adyacencia, ya que al tratarse de un circuito síncrono, los cambios de estado están gobernados por la señal de reloj y no por los cambios de las señales internas, por lo tanto, sólo habrá un cambio de estado, y problemas como las carreras y ciclos de estados son eliminados.

La tabla de transición es:

		0	1
$q_1q_0$	00	00,0	10,0
	01	-, -	-, -
	11	00,0	00,1
	10	11,0	00,0

$Q_1Q_0, F$

No obstante, las señales  $Q_1$  y  $Q_0$  no nos sirven porque son señales internas de los flip-flops. Las señales que tenemos que hallar son las entradas de los flip-flops, en este caso particular la entrada T. Así, en función de la tabla de excitación del biestable tipo T, la tabla de excitación del circuito es:

$q$	$\rightarrow$	$Q$	.....	$T$
0	$\rightarrow$	0	.....	0
0	$\rightarrow$	1	.....	1
1	$\rightarrow$	0	.....	1
1	$\rightarrow$	1	.....	0

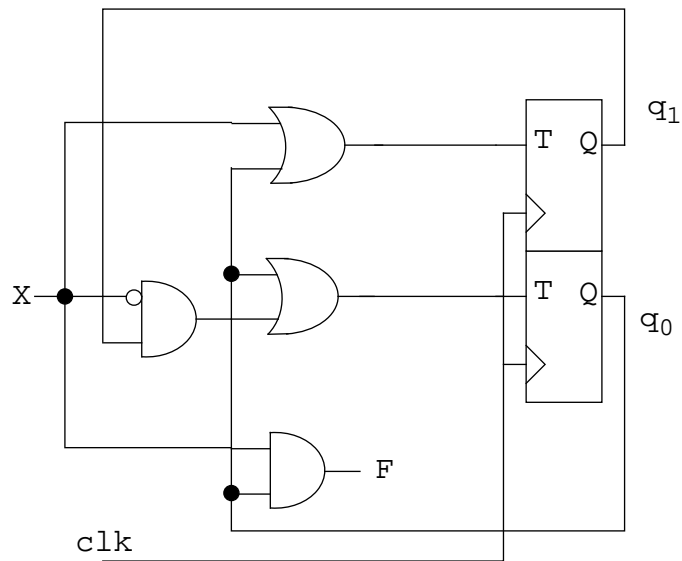
	$q_1q_0$	
	00	01
	11	10

	0	1
00	00,0	10,0
01	-, -	-, -
11	11,0	11,1
10	01,0	10,0

$T_1T_0, F$

Por lo tanto:

- $T_1 = \sum m(3, 4, 6, 7) + \phi(1, 5) = X + q_0$
- $T_0 = \sum m(2, 3, 7) + \phi(1, 5) = q_0 + \bar{X} \cdot q_1$
- $F = \sum m(7) + \phi(1, 5) = X \cdot q_0$



5.3. Diseñe los mismos circuitos de forma síncrona, utilizando biestables tipo JK.

## Problemas del Tema 6

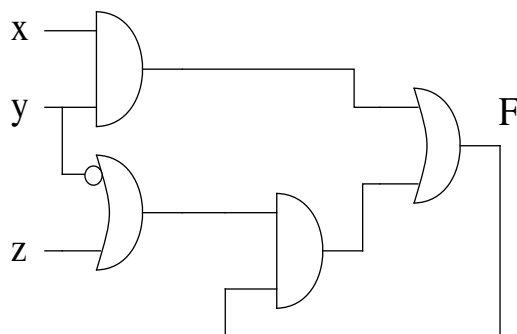
6.1. Indicar las especificaciones mínimas que deben tener los siguientes dispositivos programables, para poder implementar los circuitos mostrados en el problema 1 del tema 4:

- PLS
- PAL
- GAL
- FPGA (nº IOB, nº CLB, nº entradas y flip-flops del CLB)

De forma previa a la solución del problema vamos a recordar resumidamente como estaban formados cada uno de estos dispositivos, y por tanto, que hay que indicar:

- PLS (Secuenciador lógico programable), se trata de un PLA cuyas salidas estaban registradas (conectadas a flip-flops, por lo general tipo D). Luego hay que indicar:
  - Número de entradas
  - Número de líneas AND
  - Número de flip-flops
- PAL, se trata de una PAL, tal que tiene una serie de salidas como salidas combinacionales, y otro grupo de ellas como salidas registradas. Luego hay que indicar:
  - Número de entradas
  - Número de líneas AND
  - Número de salidas sin registrar
  - Número de salidas registradas
- GAL, se trata de una PAL cuyas salidas entran en un bloque que puede ser configurado de diferentes formas (salidas combinacionales/registradas, salidas realimentadas/sin realimentar). Luego hay que indicar:
  - Número de líneas AND
  - Número de salidas, así como su configuración
- FPGA, se trata de una matriz de bloques IOB (entrada/salida) y CLB (procesado, que son básicamente GAL sustituyendo la PAL por una memoria y en las que las salidas no se realimentan de forma interna, sino que se desaprovecha una entrada en caso necesario). Por lo tanto, hay que indicar:
  - Número de IOB, así como su configuración
  - Número de CLB, así como su configuración (es decir, toda la configuración de salida de la GAL equivalente)

## Circuito A



En el caso de los PLS, este circuito no puede ser implementado ya que todas sus salidas están registradas, luego no podemos implementar un circuito en modo fundamental.

En el caso de las PAL, tendremos que expresar la función como suma de productos, luego:

$$F = x \cdot y + (z + \bar{y}) \cdot f = x \cdot y + z \cdot f + \bar{y} \cdot f$$

Por lo tanto, el dispositivo PAL necesitará:

- 4 entradas (tres de dato y otra para la realimentación)
- 3 líneas AND
- 1 salida combinacional
- 0 salidas registradas

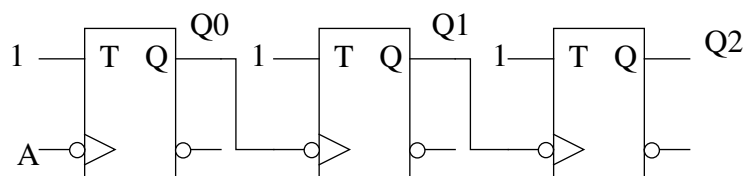
En el caso de la GAL, trabajaremos con la función para la PAL. Así que la GAL necesitará:

- 3 entradas (tres de dato)
- 3 líneas AND
- 1 salida configurada como salida combinacional realimentada

En el caso de las FPGA tenemos varias soluciones, en función de como realicemos el particionado del circuito. No obstante, siempre necesitaremos 4 IOB (3 configurados como entradas sin registrar y 1 configurado como salida sin registrar)

- 1º particionado:  $F = x \cdot y + (z + \bar{y}) \cdot f$ . Luego necesitaremos 1 CLB con cuatro entradas (tres de dato y la salida realimentada) y 1 salida sin registrar.
- 2º particionado:  $F = x \cdot y + G$ ;  $G = (z + \bar{y}) \cdot f$ . Luego necesitaremos 2 CLB con tres entradas y una salida sin registrar.
- 3º particionado:  $G1 = x \cdot y$ ;  $G2 = z + \bar{y}$ ;  $G3 = f \cdot G2$ ;  $F = G1 + G3$ . Luego necesitaremos 4 CLB con dos entradas y una salida sin registrar.

## Circuito E



Vamos a considerar que los flip-flops que disponen todos los dispositivos programables son de tipo D, luego hemos de pasar los biestables de tipo diferente a tipo D. Así, en este caso en que  $T_i = '1'$ , pasamos a que  $D_i = \bar{q}_i$ .

En el caso del PLS, no puede ser implementado en uno solo ya que todas las entradas de reloj de todos los flip-flops son comunes, situación que no se verifica en el circuito que nos ocupa.

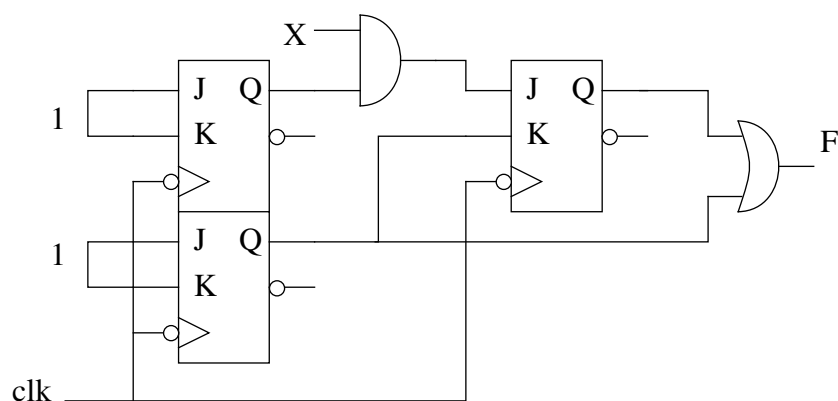
En el caso de la PAL, nos encontramos en una situación similar al del PLS

En el caso de la GAL, no encontramos en una situación similar a las anteriores.

En el caso de las FPGA, los relojes comunes pertenecen a aquellos flip-flops que se encuentran en el mismo CLB. Por lo tanto, en este caso, el número de CLB vendrá determinado por el número de flip-flops, en lugar del particionado que podamos realizar del circuito. Por lo tanto, necesitaremos:

- 4 IOB: 1 configurado como entrada no registrada y 3 configuradas como salidas no registradas
- 3 CLB con 1 entrada y una salida registrada.

## Circuito G



Al igual que pasaba en el caso anterior, debemos transformar los flip-flops JK en flip-flops tipo D. Como  $J_{1,2} = K_{1,2} = '1'$ , obtenemos que  $D_{1,2} = \bar{q}_{1,2}$ , y  $J_3 = X \cdot q_1$ ,  $K_3 = q_2$ ,  $D_3 = Q_3 = X \cdot q_1 \cdot \bar{q}_3 + \bar{q}_2 \cdot q_3$ .

En el caso del PLS, este circuito no es posible imple-



mentarlo ya que la salida es combinacional, que no e disponen en este tipo de dispositivos.

En el caso de la PAL, necesitaremos

- 4 entradas
- 8 líneas AND
- 4 salidas, tres de las cuales registradas, y una sin registrar

En el caso de la GAL, serán necesarios

- 1 entrada
- 8 líneas AND
- 4 salidas, tres de las cuales registradas y realimentadas, y una sin registrar ni realimentar

En el caso de la FPGA necesitaremos 2 IOB (uno configurado como entrada sin registrar y otro como salida sin registrar). Una posible opción de particionado, nos lleva a utilizar dos CLB con cuatro entradas y dos etapas de salida (en uno de ellos ambas salidas están registradas, mientras que en el otro existe una registrada y otra sin registrar).

6.2. Realizar los diseños del problema 2 del tema 2 utilizando un dispositivo programable secuencial con las siguientes características:

- 2 entradas
- 1 salida combinacional
- 1 salida registrada realimentada sin conexión al exterior
- 1 salida registrada realimentada con conexión al exterior.

	00	01	10	11
A	E,1	A,1	A,0	D,1
B	B,0	C,0	C,0	D,1
C	F,1	C,1	C,0	D,1
D	G,1	D,1	D,0	D,1
E	E,0	E,0	A,0	D,1
F	F,0	F,0	A,0	D,1
G	G,0	A,1	B,1	G,0
H	B,0	C,1	H,1	H,0

De la solución del tema 2, obtuvimos que los máximos compatibles eran:  $MC = \{AC, B, D, EF, G, H\}$ , es decir, necesitaremos tres señales de estado. Pero como el dispositivo que tenemos sólo puede tener dos, esta máquina no es implementable en este dispositivo.

Máquina

	0	1
A	D,1	B,0
B	D,0	B,0
C	C,0	A,1
D	A,0	C,0

Podemos comprobar que esta máquina es la máquina mínima.

Las reglas de adyacencia deberán ser:

- A debe ser adyacente a D y B
- B debe ser adyacente a D
- C debe ser adyacente a A
- D debe ser adyacente a A y C

Luego una posible codificación es:

		q <sub>1</sub>	
		0	1
q <sub>0</sub>	0	A	D
	1	B	C

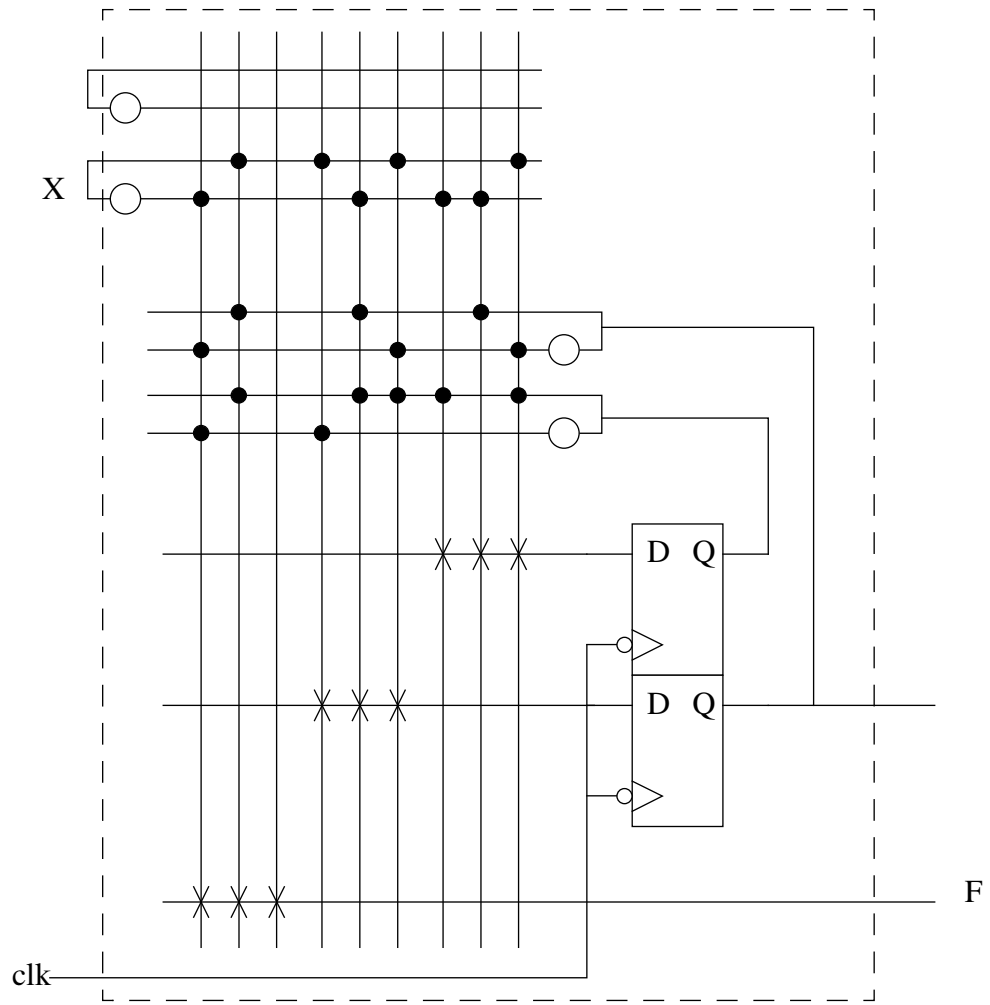
Por lo tanto, la tabla de transición será:

		0	1
q <sub>1</sub> q <sub>0</sub>	00	10,1	01,0
	01	10,0	01,0
	11	11,0	00,1
	10	00,0	11,0

Q<sub>1</sub>Q<sub>0</sub>, F

Al tratarse de flip-flops tipo D, Q=D, las tablas de transición y excitación coinciden. Luego la función que hay que implementar es:

- $Q_1 = \sum m(0,1,2,6) = \bar{X} \cdot q_1 + \bar{X} \cdot q_0 + X \cdot q_1 \cdot \bar{q}_0$
- $Q_0 = \sum m(3,4,5,6) = X \cdot \bar{q}_1 + \bar{X} \cdot q_1 \cdot q_0 + X \cdot q_1 \cdot \bar{q}_0$
- $F = \sum m(0,7) = \bar{X} \cdot \bar{q}_1 \cdot \bar{q}_0 + X \cdot q_1 \cdot q_0$



## Problemas del Tema 7

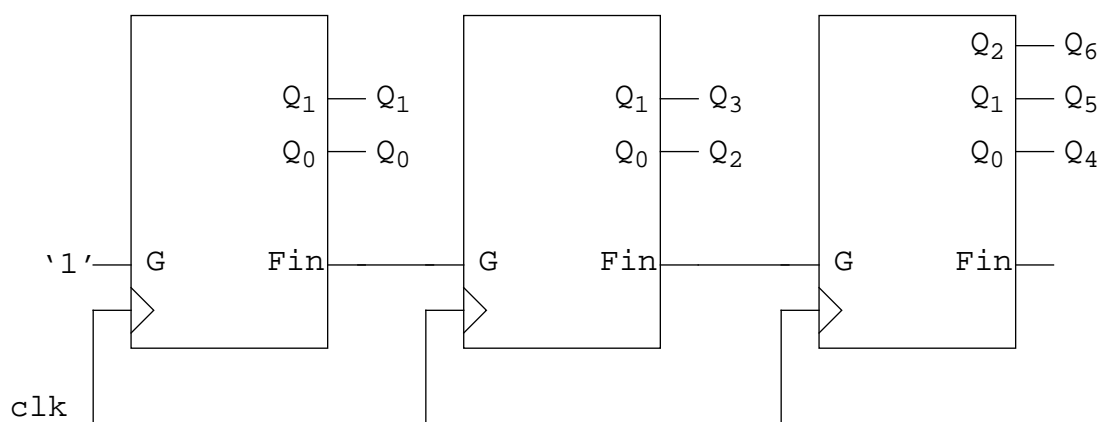
7.1. Diseñe un contador de módulo 4 con las siguientes características:

- Síncrono disparado por el flanco de subida
- Carga paralela
- Inhabilitación de la cuenta
- Cuenta ascendente y descendente
- utilice puertas lógicas y biestables tipo T

7.2. Diseñar un contador serie de módulo 96 utilizando:

- contadores de módulo 4 y 6

En primer lugar, debemos obtener el número de contadores que necesitamos para obtener el contador de módulo 96. Para ello debemos ir dividiendo 96 entre los módulos que tenemos hasta que el resultado sea la unidad. Es decir  $((96:4):4):6=1$ ; luego para este caso particular necesitaremos dos contadores de módulo 4 y uno de módulo 6. Por lo tanto, una posible implementación será:



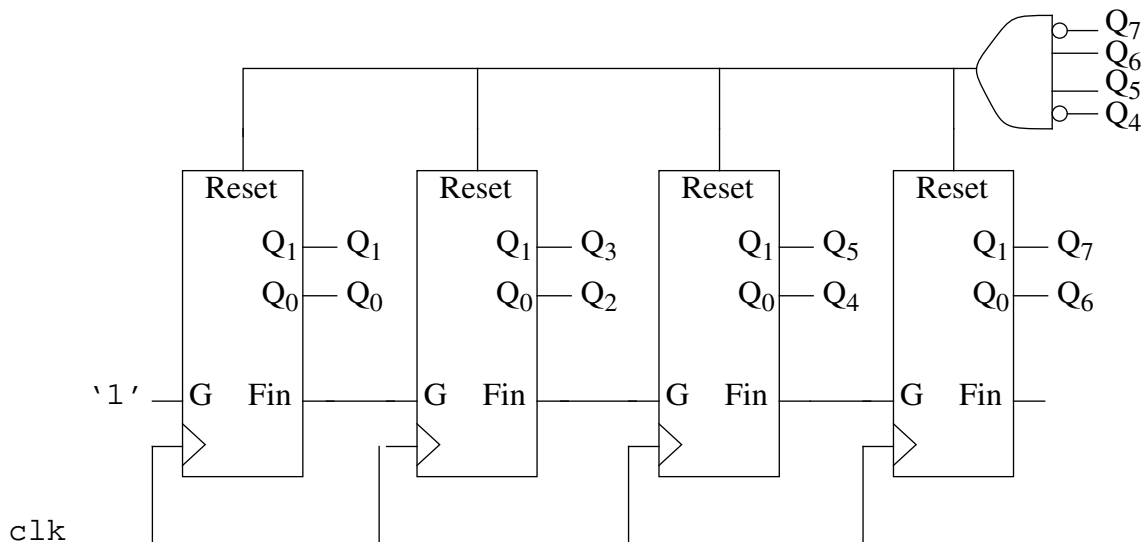
En el caso en que las divisiones no sean exactas, es decir, exista un resto distinto de cero, nos indica que habrá que añadir una circuitería para resetear los contadores cuando sea necesario. Además, cuando el resto es diferente de cero, nos está indicando que las salidas de dicho contador estarán en la circuitería de reset.

Por ejemplo si queremos realizar el contador utilizando únicamente contadores de módulo 4, necesitaremos:

- $96 : 4 = 24$ , resto= 0
- $24 : 4 = 6$ , resto= 0

- $6 : 4 = 1$ , resto=2
- $1 : 4 = 0$ , resto=1

Así el valor máximo al que debe llegar los dos últimos contadores (de forma simultánea) es '2' y '1'. Cuando el resto vale '0', esos contadores realizarán su cuenta completa, luego no influye en el reset. Por lo tanto, el diseño será:



No obstante, la circuitería de reset se puede mejorar ya que el bit  $Q_7$  nunca puede llegar a '1', luego esa entrada de la puerta AND se puede eliminar.

- contadores de módulo 8 y 12

7.3. Se desea construir un reloj digital, para lo cual se dispone de circuitos secuenciales y combinacionales MSI. La señal periódica de partida se repite cada tercio de segundo.

Un reloj digital suele tener 6 dígitos: dos para las horas, dos para los minutos y dos para los segundos. Así, necesitaremos:

- $S_0$  --> contador de módulo 10
- $S_1$  --> contador de módulo 6
- $M_0$  --> contador de módulo 10
- $M_1$  --> contador de módulo 6
- $H_0$  --> contador de módulo 10
- $H_1$  --> contador de módulo 3

También necesitaremos seis displays BCD para mostrar los dígitos.

Cuando se llegue a 24h, debemos resetear el reloj completo, para empezar un nuevo día, luego también hace falta una circuitetría de reset.

Todas estas consideraciones han sido realizadas, tomando como señal de reloj, una señal de 1s., no obstante tenemos una señal de 1/3 de segundo. Por lo

tanto, necesitaremos un nuevo contador de módulo 3 para generar el reloj de 1s.  
Luego el circuito será:

