

EXAMEN RESUELTO DE IC

8 de noviembre de 2000

Temps: 2h 30min.

PROBLEMA 1 (1.5 punts)

a) Escriu una fórmula que expressi el valor d'un nombre natural representat amb quatre dígitos en base 5, en funció dels seus dígitos x_3 , x_2 , x_1 i x_0 . Quins valors pot agafar cada dígit? Quin és el rang d'aquesta representació? (Objectiu 4.1: 0.75 punts).

b) Representa en complement a 2 (Ca2) els nombres implicats en les següents operacions (justificant com ho has fet). Realitza les operacions treballant amb les representacions en Ca2, amb el nombre de bits que s'indica. En cada cas, digues si el resultat és representable o no (si es produeix sobreiximent/overflow), sense passar-lo a decimal. Assenyala d'on treus aquesta informació. (Objectiu 5.5: 0.75 punts).

b1) $(-2)+(-3)$, en 3 bits

b2) $5+6$, en 4 bits

b3) $11+(-11)$, en 5 bits

Solució

a)

Fórmula que expressa el valor d'un nombre natural representat en quatre xifres en base 5 ($x_3x_2x_1x_0$):

$$V = \sum_{i=0}^3 x_i \cdot 5^i = x_3 \cdot 5^3 + x_2 \cdot 5^2 + x_1 \cdot 5^1 + x_0 \cdot 5^0$$

Cada xifra en base 5 pot prendre els valors 0, 1, 2, 3 o 4.

El rang de números representables amb quatre xifres en base 5 és del 0 al $5^4-1 = 624$.

b1)

La representació en Ca2, 3 bits, de -2 és 110 (010 \rightarrow 101+1 = 110).

La representació en Ca2, 3 bits, de -3 és 101 (011 \rightarrow 100+1 = 101).

$$\begin{array}{r}
 \textcircled{1} \textcircled{0} \ 0 \ 0 \\
 \textcircled{1} \ 1 \ 0 \\
 + \ 1 \ 0 \ 1 \\
 \hline
 \textcircled{0} \ 1 \ 1
 \end{array}$$

Per determinar si el resultat de la operació és correcte, a partir de la representació (no reinterpretant els números en decimal) hi ha dues formes:

regla dels signes: si els dos operands són del mateix signe i el resultat és de signe contrari, la operació és incorrecta (els tres bits marcats a la darrera columna, la de més a l'esquerra).

regla dels *carrys*: si la XOR dels dos últims *carrys* dóna 1, la operació és incorrecta (els dos últims *carrys* són el que entra i el que surt de la darrera columna, marcats a la operació).

Quan la operació és incorrecta es diu que s'ha produït un sobreiximent o *overflow*. Amb una petita taula es pot comprovar que les dues regles són equivalents. És bo observar també que el *carry* de sortida es calcula només per poder determinar la regla dels *carrys*. En Ca2, el *carry* de sortida no té cap sentit.

En aquest cas, qualsevol de les dues regles revela que el resultat és incorrecte. Se sumen dos negatius i el resultat és positiu i la XOR dels dos *carrys* indicats dóna 1.

b2)

La representació en Ca2, 4 bits, de 5 és 0101.

La representació en Ca2, 4 bits, de 6 és 0110.

$$\begin{array}{r}
 \textcircled{0} \textcircled{1} \ 0 \ 0 \ 0 \\
 \textcircled{0} \ 1 \ 0 \ 1 \\
 + \ 0 \ 1 \ 1 \ 0 \\
 \hline
 \textcircled{1} \ 0 \ 1 \ 1
 \end{array}$$

Altres cop, qualsevol de les dues formes de determinar la correcció del resultat ens revela que no és correcte. Se sumen dos positius i el resultat és negatiu i la XOR dels *carrys* dóna 1. Hi ha *overflow*.

b3)

La representació en Ca2, 5 bits, d'11 és 01011.

La representació en Ca2, 5 bits, de -11 és 10101 (01011 -> 10100+1 = 10101).

$$\begin{array}{r}
 \textcircled{1} \textcircled{1} \ 1 \ 1 \ 1 \ 0 \\
 \textcircled{0} \ 1 \ 0 \ 1 \ 1 \\
 + \ 1 \ 0 \ 1 \ 0 \ 1 \\
 \hline
 \textcircled{0} \ 0 \ 0 \ 0 \ 0
 \end{array}$$

En aquesta ocasió les dues regles confirmen que el resultat és correcte. Els signes dels operands són diferents i la XOR dels dos *carrys* dóna 0.

Errors habituals

La mecànica del càlcul no ha significat un problema per a quasi ningú. Els dos errors més habituals han estat:

Aplicar la operació de canvi de signe en Ca2 (intercanviar 0's per 1's i 1's per 0's i sumar 1) als nombres positius. Només recordar que en el sistema de representació del Ca2 els positius es representen igual que en binari natural. Són els negatius els que es representa en binari natural el seu valor absolut i se li canvia el signe aplicant l'esmentada operació de canvi de signe en Ca2.

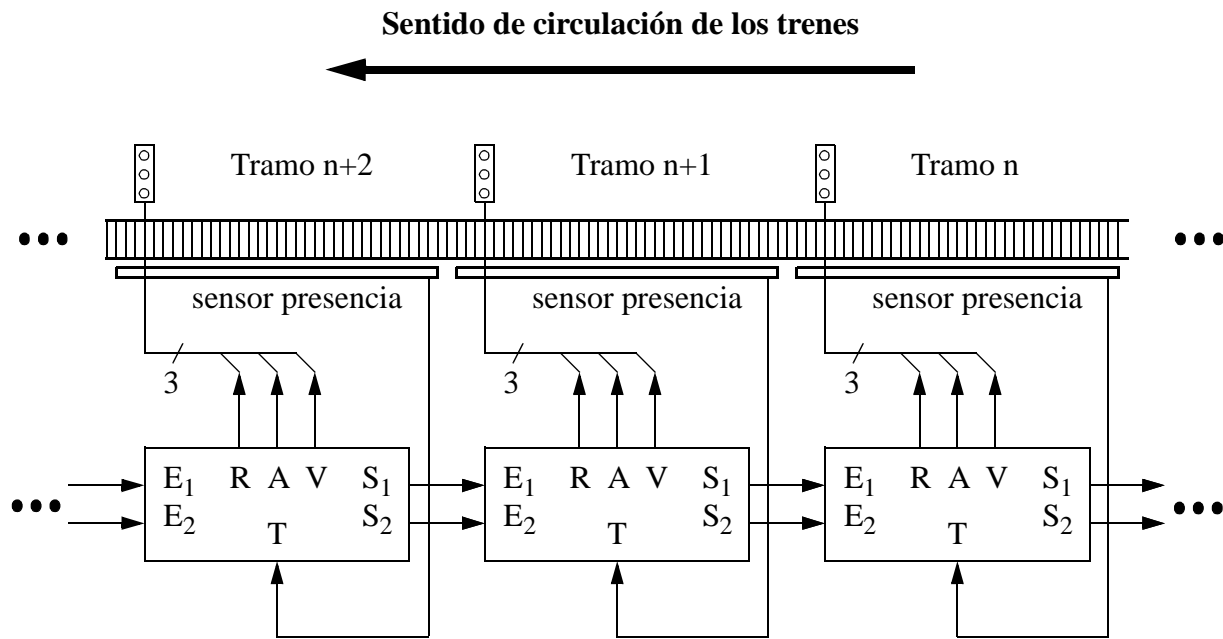
Determinar si el resultat de la operació és correcte o no en funció del *carry*. Insistir un cop més que en Ca2, el *carry* no té cap sentit i s'ha de despreciar. De fet, no cal ni calcular-lo; només si es pensa usar per a determinar la correcció del resultat aplicant la regla dels *carrys*.

PROBLEMA 2 (3 punts)

Se desea diseñar un sistema de señalización para controlar los trenes que circulan por una vía de un solo sentido. La vía se descompone en tramos de unos 5 kilómetros numerados de forma creciente en el sentido de circulación de los trenes. Cada tramo tiene un sensor de presencia que determina si hay un tren en ese tramo y un semáforo al final del tramo que puede producir uno de los siguientes colores: rojo, amarillo o verde.

Para evitar la colisión que podría producirse cuando un tren este parado o circule despacio y llegue otro por detrás a más velocidad, se desea que los semáforos funcionen de la siguiente forma: el semáforo del tramo n se debe poner rojo si hay un tren en el tramo $n+1$, amarillo si no hay tren en el tramo $n+1$ pero si lo hay en el $n+2$ y verde en cualquier otro caso.

El sistema lógico combinacional que controla toda la vía se descompone en tantos bloques combinatoriales iguales como tramos, conectados entre sí como muestra la siguiente figura.



Para el bloque del tramo n:

- La entrada T vale 1 cuando hay un tren en el tramo n.
- Las entradas E₁ y E₂ que vienen del bloque del tramo n+1 tienen la siguiente información: E₁ vale 1 cuando hay un tren en el tramo n+1 y E₂ vale 1 cuando hay un tren en el tramo n+2.
- Las salidas R, A y V se deben poner a 1 para encender las luces roja, amarilla y verde del semáforo, respectivamente.
- Las salidas S₁ y S₂ deben ser las adecuadas para que el sistema funcione tal como se ha descrito.

Se pide:

a) Completar la tabla de verdad del bloque que controla un tramo de vía. (Objetivo 6.1: 2 puntos).

T	E ₁	E ₂	R	A	V	S ₁	S ₂

b) Implementar el bloque que controla cada tramo con el mínimo número de puertas a 2 niveles usando para la minimización tablas (mapas) de Karnaugh. (Objetivo 6.4.a.3: 1 punto).

Solución

a)

T	E ₁	E ₂	R	A	V	S ₁	S ₂
0	0	0	0	0	1	0	0
0	0	1	0	1	0	0	0
0	1	0	1	0	0	0	1
0	1	1	1	0	0	0	1
1	0	0	0	0	1	1	0
1	0	1	0	1	0	1	0
1	1	0	1	0	0	1	1
1	1	1	1	0	0	1	1

También hemos considerado correctas otras soluciones, como por ejemplo que cuando no hay tren en el tramo el semáforo no importa que color tenga (aunque las señales S₁ y S₂ sí que tienen que valer lo que indica la anterior tabla).

b)

R

		E ₂		E ₁	
		00	01	11	10
T	0	0	0	1	1
	1	0	0	1	1

$R = E_1$

A

		E ₂		E ₁	
		00	01	11	10
T	0	0	1	0	0
	1	0	1	0	0

$A = \bar{E}_1 E_2$

V

		E ₂		E ₁	
		00	01	11	10
T	0	1	0	0	0
	1	1	0	0	0

$V = \bar{E}_1 \bar{E}_2$

S₁

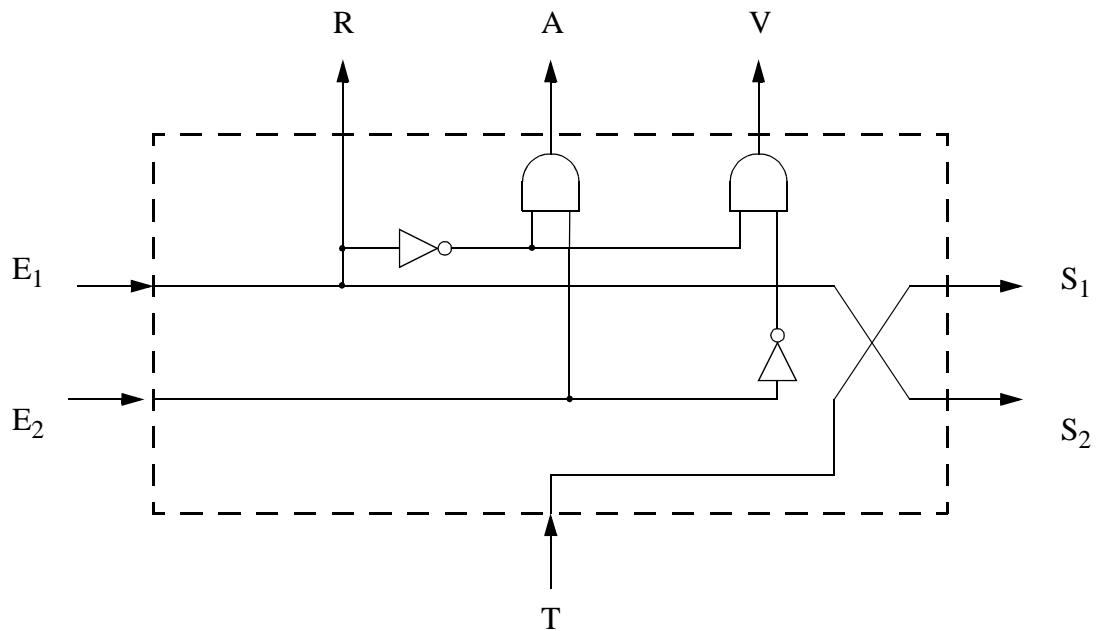
		E ₂		E ₁	
		00	01	11	10
T	0	0	0	0	0
	1	1	1	1	1

$S_1 = T$

E₂

		E ₂		E ₁	
		00	01	11	10
T	0	0	0	1	1
	1	0	0	1	1

$S_2 = E_1$



Errores habituales

En muchos exámenes la minimización de las funciones de salida era correcta, pero a la hora de implementar las expresiones a 2 niveles se han cometido uno de los dos errores siguientes, tal vez motivados porque las expresiones de las funciones eran muy sencillas (no requerían ninguna puerta o solo una AND):

- 1) Después de implementar las funciones correctamente unir todas (o al menos R, V y A) las salidas con una puerta OR, obteniendo una única salida. Esto no tiene ningún sentido y es un error grave.
- 2) Para implementar, por ejemplo $S_1 = T$, poner S1 a una AND con la otra entrada a 1 y su salida a una OR con la otra entrada a 0. Esto no es grave pues es funcionalmente correcto, pero se usan puertas innecesariamente. Diseño a 2 niveles NOT-AND-OR quiere decir que la implementación no debe tener más de 2 niveles de puertas (AND y OR), pero puede tener menos si la función es muy simple.

PROBLEMA 3 (2.5 puntos)

Diseña los siguientes circuitos:

a) Implementa un sumador de dos números naturales de 16 bits utilizando **únicamente** sumadores de dos números naturales de 4 bits. (Objectiu 6.3.c: 0.75 punts).

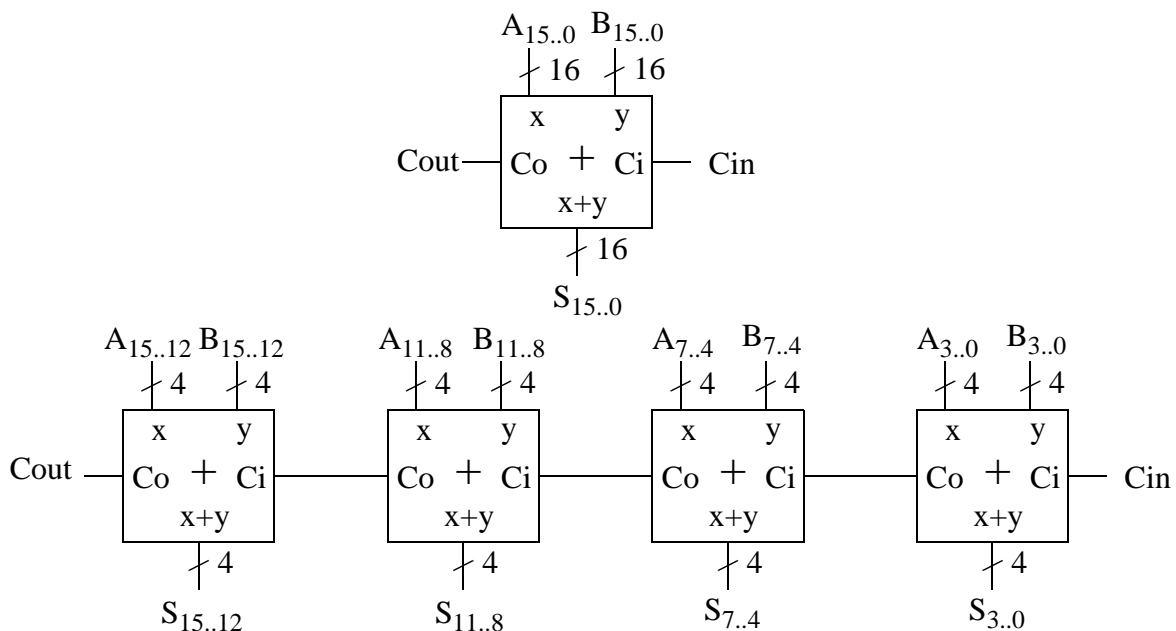
b) Implementa, a nivel de bloques, un circuito que calcule la función Y descrita a continuación:

$$Y = \begin{cases} B + C & \text{si } A = B \\ 2 \times C & \text{si } A \neq B \end{cases}$$

Y, A, B y C son números naturales codificados en binario con 8 bits (No hace falta indicar mediante ninguna señal los casos en los que el resultado no es representable con 8 bits). Para diseñar el circuito puedes usar **únicamente** puertas y bloques del chuletario del número de bits que creas conveniente. (Objectiu 6.4.b: 1.75 punts).

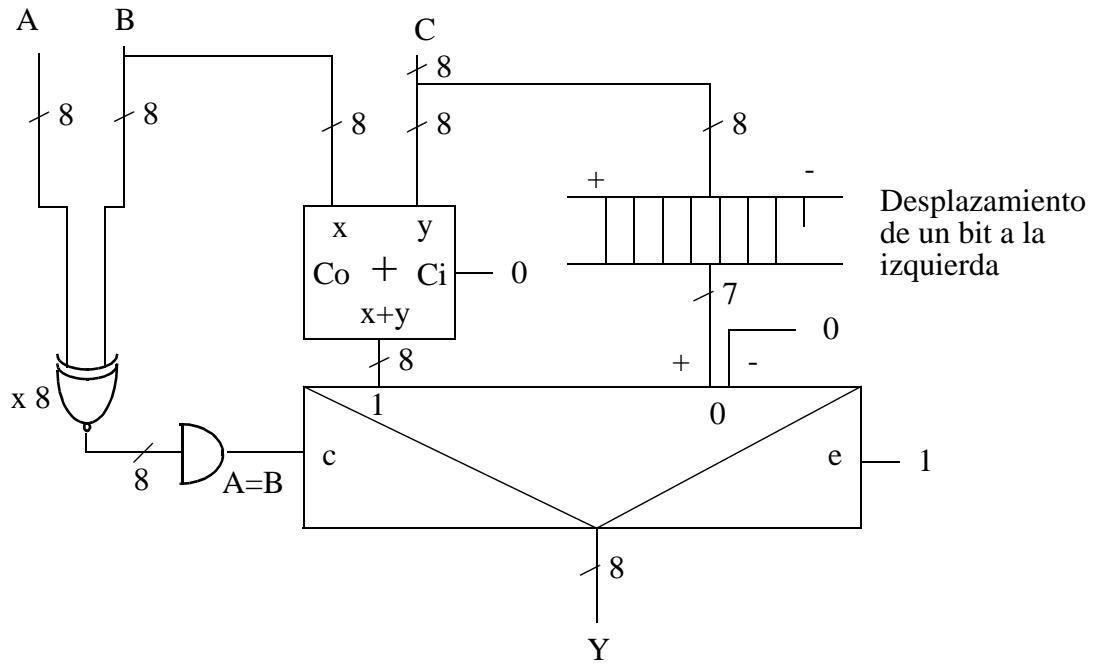
Solución

Apartado a) (0,75 puntos)

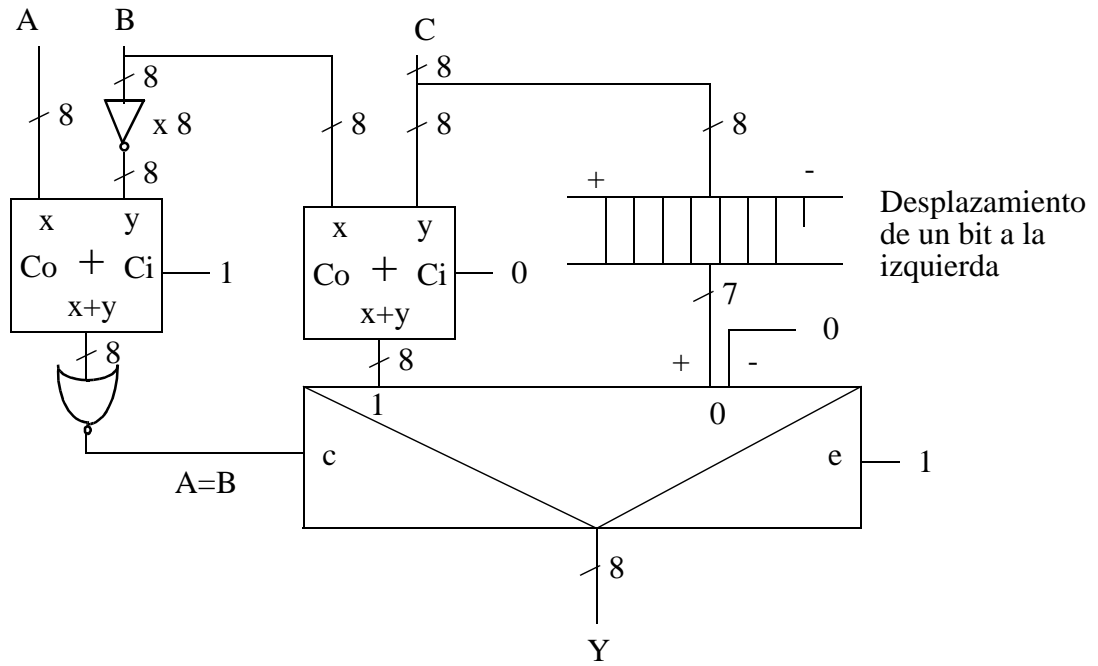


Apartado b) (1.75 puntos)

Sol. 1



Sol. 2



Errores habituales

Cuando hay que usar un sumador de n bits, muchos alumnos han usado:

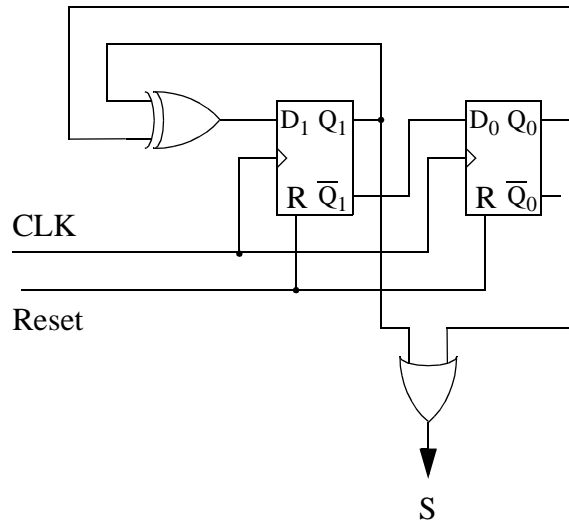
a) un Full adder de dos entradas de n bits (???). Algunos incluso ponen half adders. Está claro que estos tienen una confusión en el nombre del bloque, pero no en su funcionamiento.

b) una puerta or de dos entradas de n bits. Esto es un error grave. No es lo mismo la suma lógica que la suma aritmética.

También hay muchos alumnos que han usado (para hacer el comparador de igualdad) una puerta ex-or de dos entradas de n bits ¡con un solo bit de salida! Se podría intuir que en realidad han puesto 8 ex-or (aunque no esté indicado), pero lo que no se entiende es lo del bit único en la salida.

PROBLEMA 4 (3 punts)

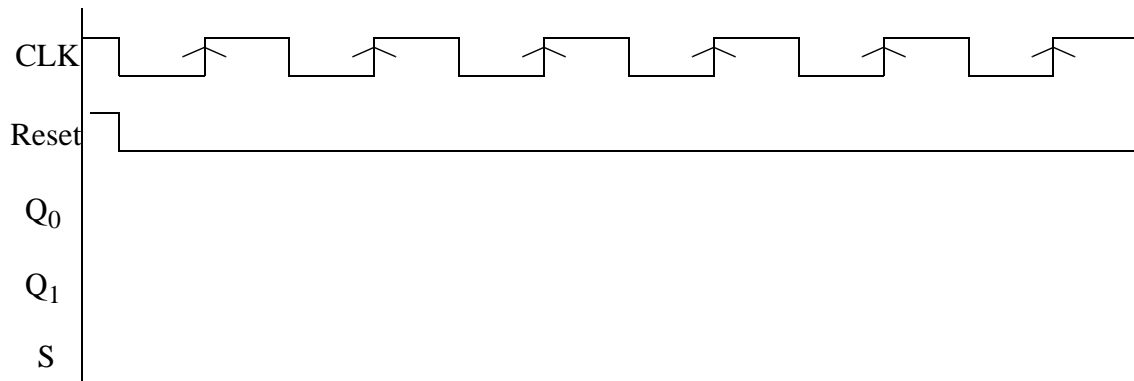
Donat el següent sistema seqüencial:



Es demana:

a) Obtenir la taula de transició d'estats, la taula de sortides i el graf d'estats de Moore. (Objectiu 6.6.a.1: 2 punts).

b) Completar la següent cronograma (Objectiu 6.6.a.2: 1 punt).



Solució

a)

Les expressions lògiques per la sortida i el següent estat son:

$$S = Q_1 + Q_0$$

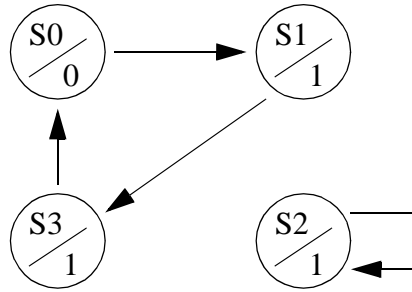
$$D_1 = Q_1 \oplus Q_0$$

$$D_0 = \overline{Q_1}$$

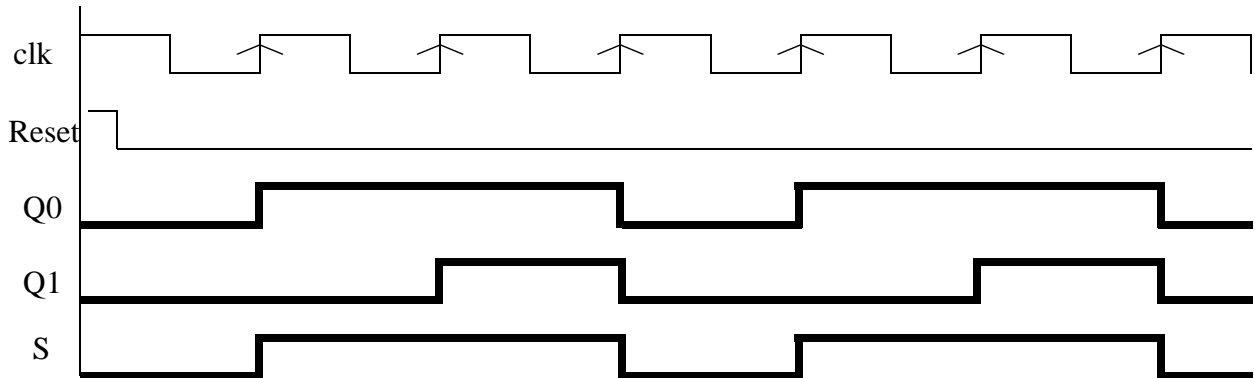
Les taules de transició d'estats i la taula de sortides s'obtenen aplicant les expressions anteriors.

Q ₁	Q ₀	D ₁	D ₀	Q ₁	Q ₀	S
0	0	0	1	S ₀ → 0	0	0
0	1	1	1	S ₁ → 0	1	1
1	0	1	0	S ₂ → 1	0	1
1	1	0	0	S ₃ → 1	1	1

El graf d'estats de Moore s'obté a partir de les taules anteriors



b) Completar la següent cronograma (Objectiu 6.6.a.2: 1 punt)



Errors habituals

a) El SLS de l'exàmen no tenia entrades síncrones externes. Un error comú ha estat considerar el RESET com una entrada síncrona, o inventarse entrades al SLS. Al disseny que es volia analitzar es podia observar clarament com el RESET entra a la senyal asíncrona dels biestables, i com a conseqüència, la taula de veritat del bloc G no es podia passar en funció del RESET.

b) Un segon error comú ha estat posar com a sortida del SLS la codificació dels estats (Q1 i Q0). El SLS que es demanava analitzar només tenia un bit de sortida.

c) Finalment, un error comú ha estat l'obtenció del cronograma. El cronograma es podia utilitzar com una forma de verificar que el graf d'estats s'havia obtingut en el primer apartat era correcte. Es a dir, fents els dos apartats per separat havia de donar el mateix resultat. Llavors, un error comú ha estat fer el primer apartat be i el segon malament, o a l'inrevés.