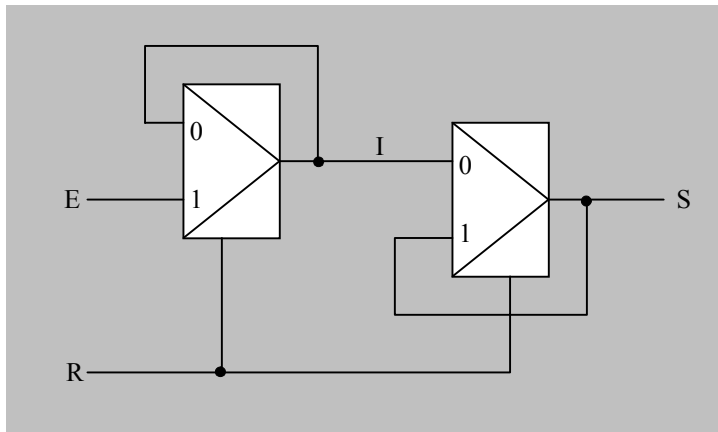


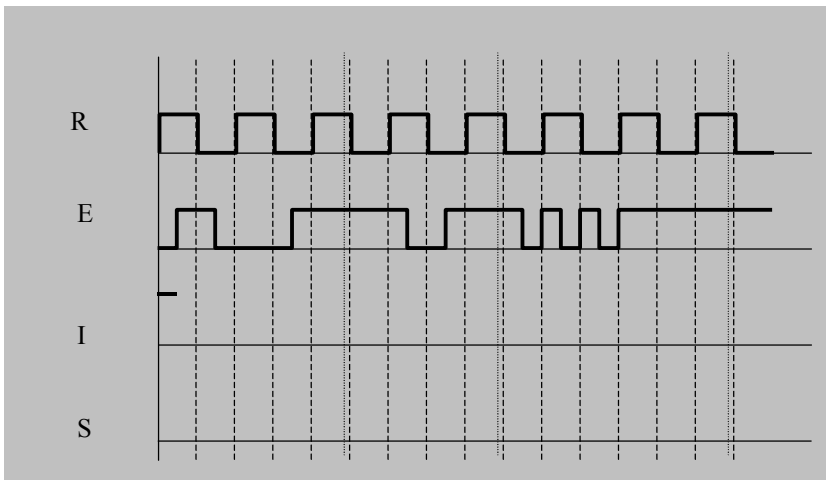
Problemes de circuitis lògics seqüencials

Fermín Sánchez Carracedo
Montse Peiron Guàrdia

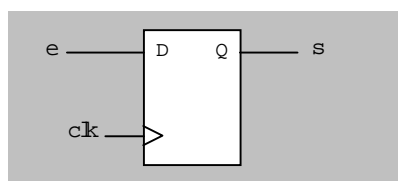
- Es desitja dissenyar un sistema que reconegui el patró "1010" en una entrada de 4 bits. Indiqueu si el sistema és seqüencial o combinacional, raonant la resposta.
- Es desitja dissenyar un sistema que reconegui una seqüència de 4 dígits decimals per identificar el número secret d'una targeta de crèdit. El sistema té una única entrada de dades de 4 bits, que codifiquen cada dígit. Indiqueu si el sistema és seqüencial o combinacional, raonant la resposta.
- Quin és el període del rellotge d'un Pentium III que funciona a 650 MHz?
- Donat el circuit següent:

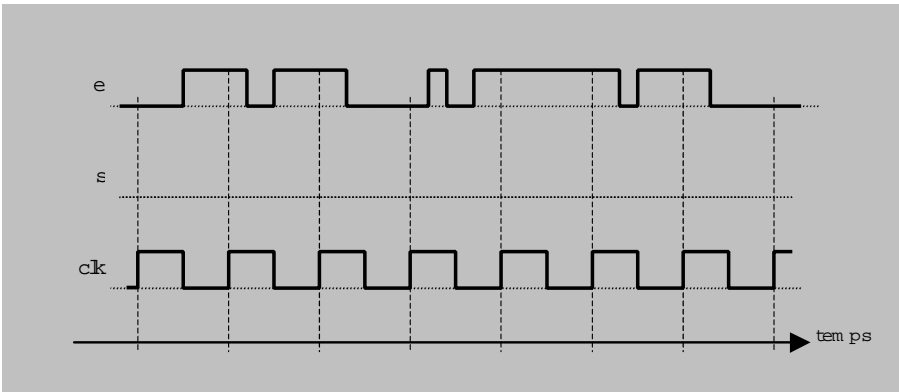


Dibuixa el valor dels senyals I i S al cronograma, suposant que el senyal I val inicialment 1.

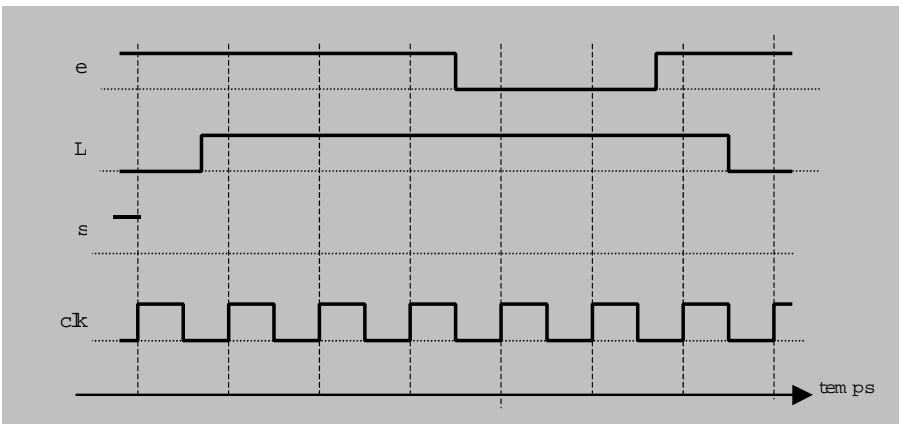
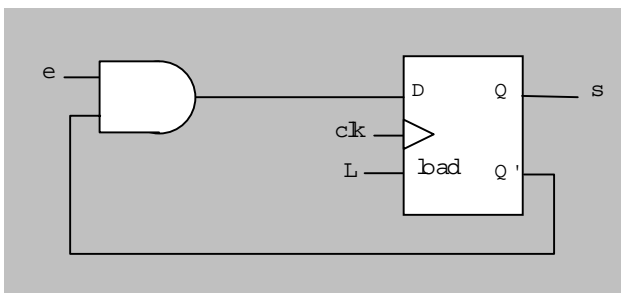


- Completeu el cronograma que correspon al circuit de la figura.

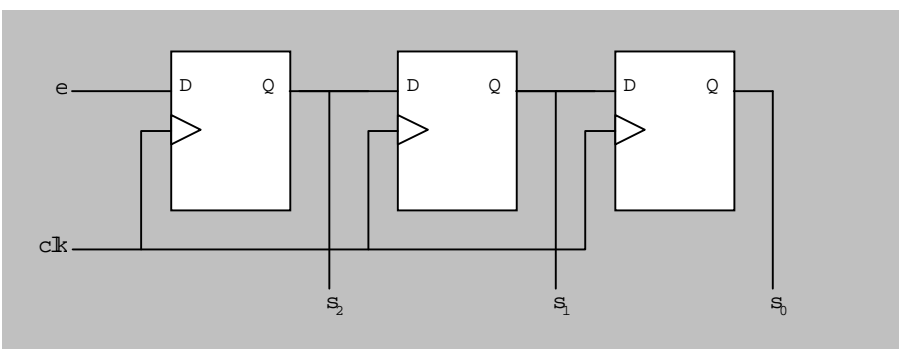


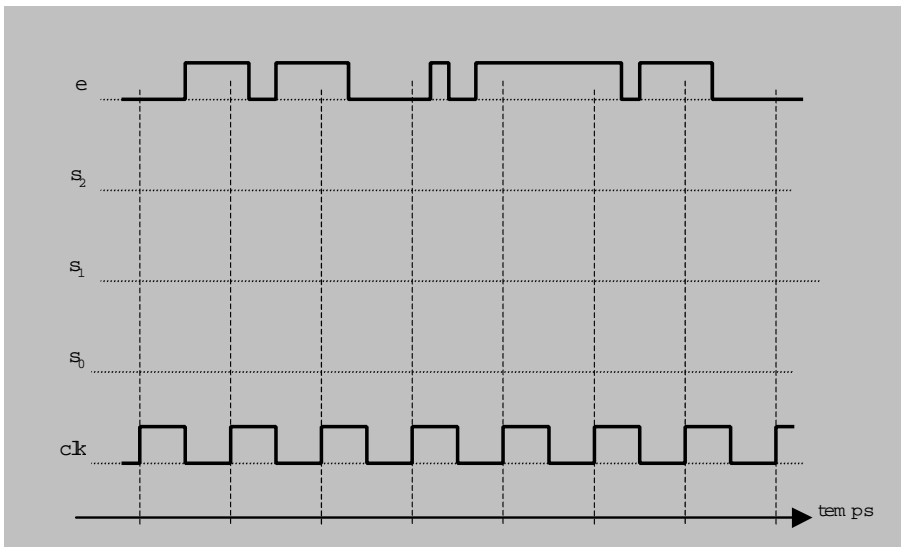


6. Completeu el cronograma que correspon al circuit de la figura, suposant que inicialment la sortida Q val 1. Quin és el paper del senyal e en el circuit?

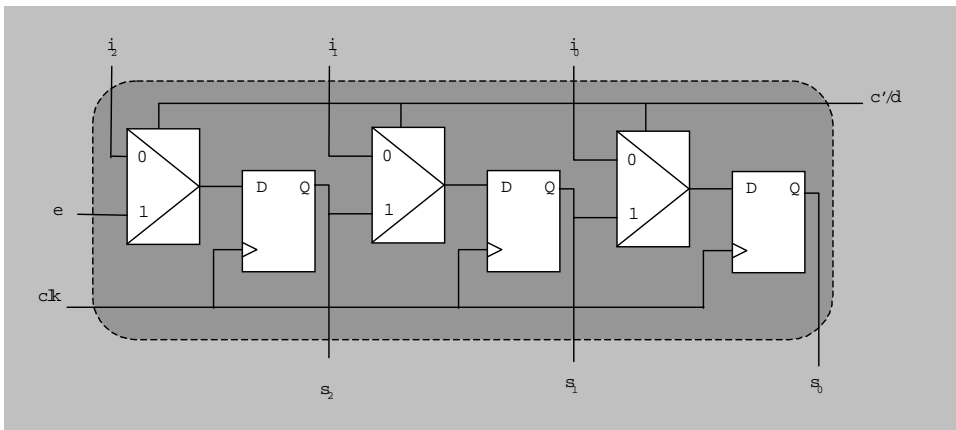


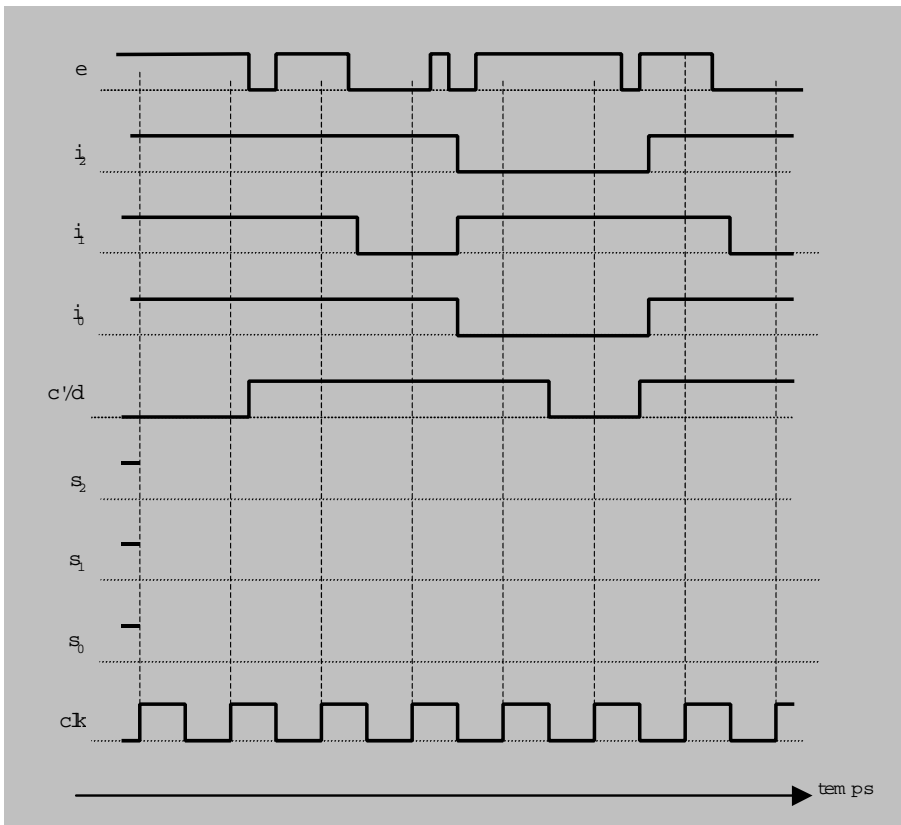
7. Completeu el cronograma que correspon al circuit de la figura.



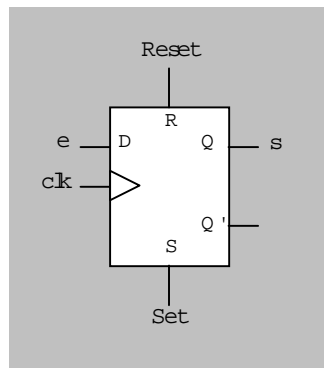


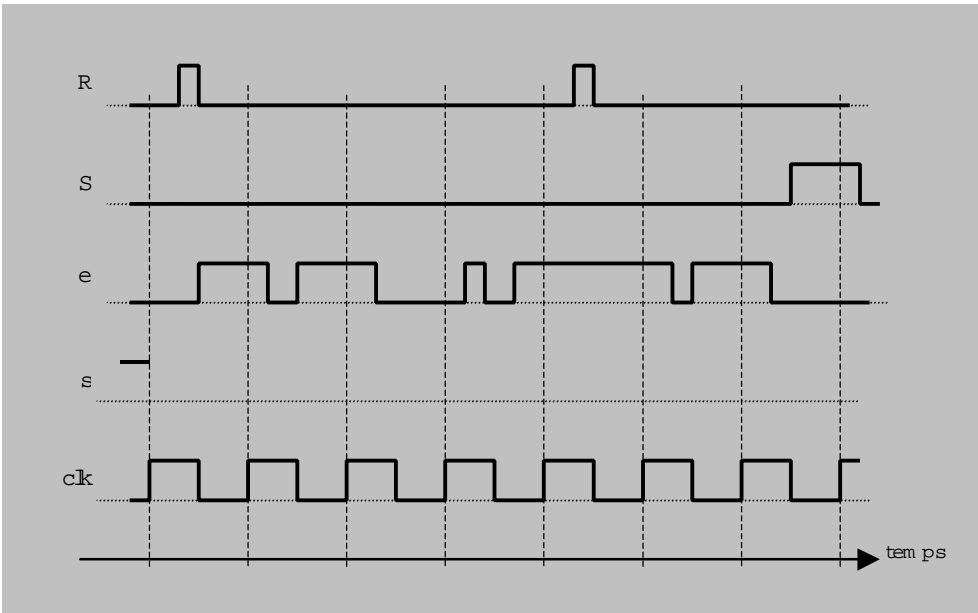
8. Completeu el cronograma que correspon al circuit de la figura, suposant que inicialment les sortides Q de tots els biestables valen 0. Descriviu en poques paraules què fa el circuit en funció del senyal c/d .



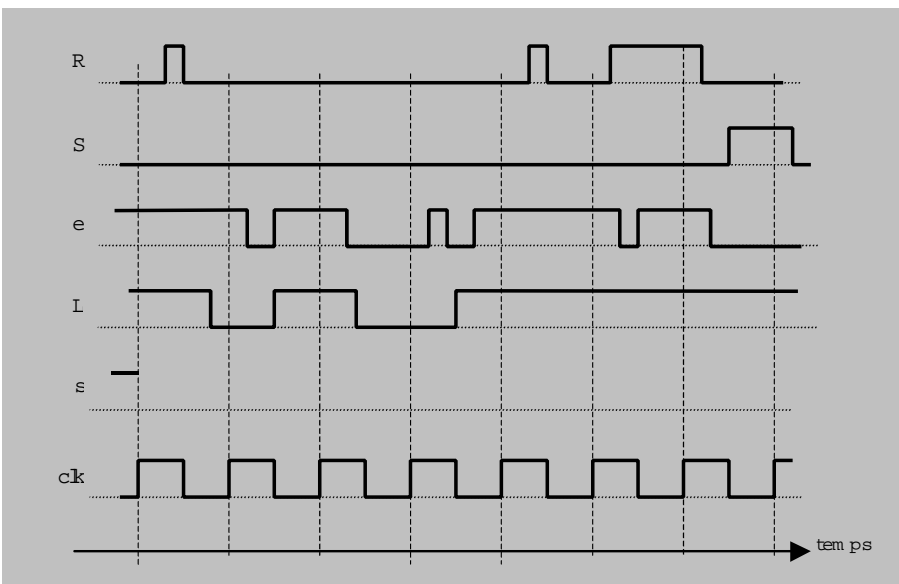
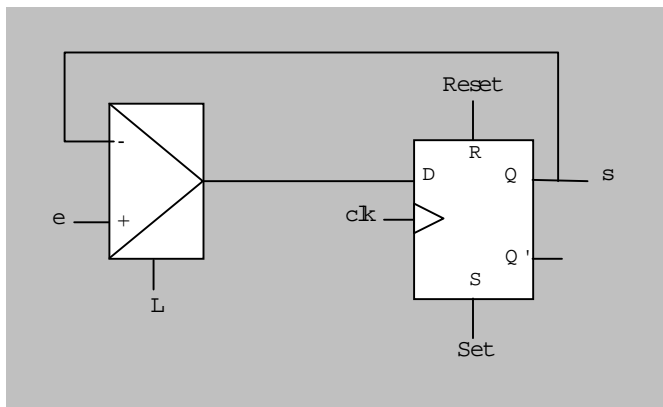


9. Completeu el cronograma que correspon al circuit de la figura, suposant que inicialment la sortida Q val 1.

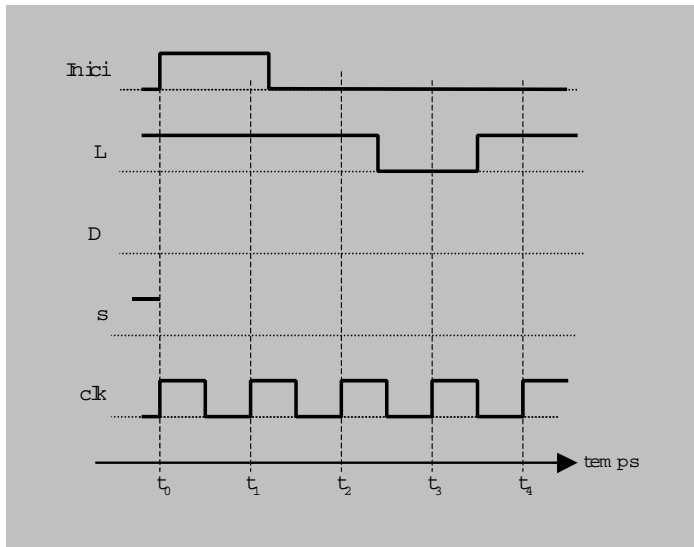
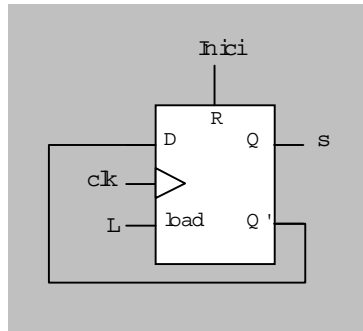




10. Completeu el cronograma que correspon al circuit de la figura, suposant que inicialment la sortida Q val 1. Quin és el paper del senyal L en el circuit?



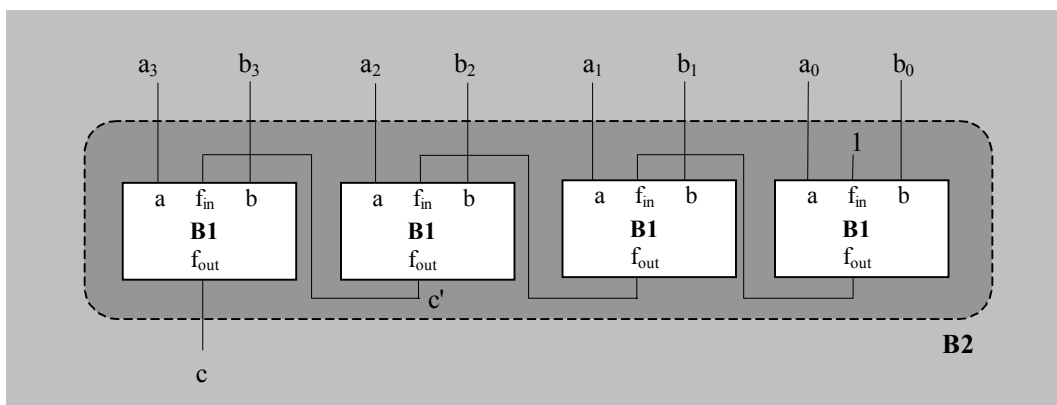
11. Completeu el cronograma que correspon al circuit de la figura.



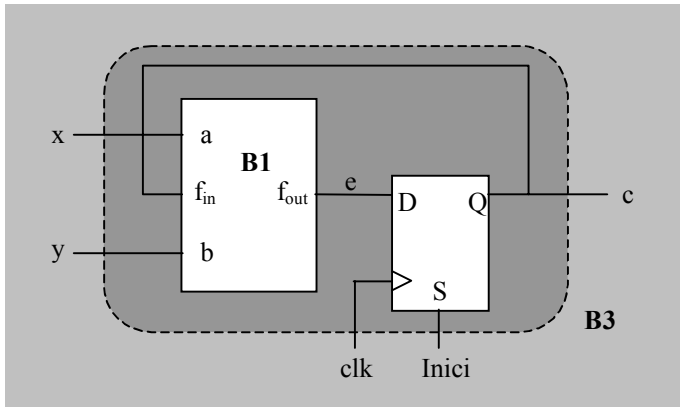
12. El circuit combinacional B2 és un comparador de dos números naturals representats en binari A i B. El seu funcionament és el següent:

$c = 0$ si A és menor que B

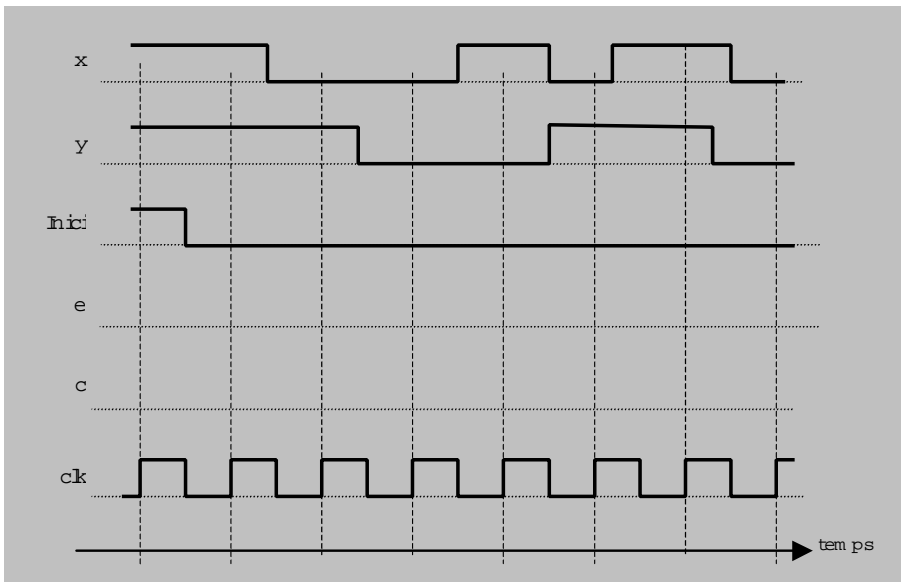
$c = 1$ si A és més gran o igual que B



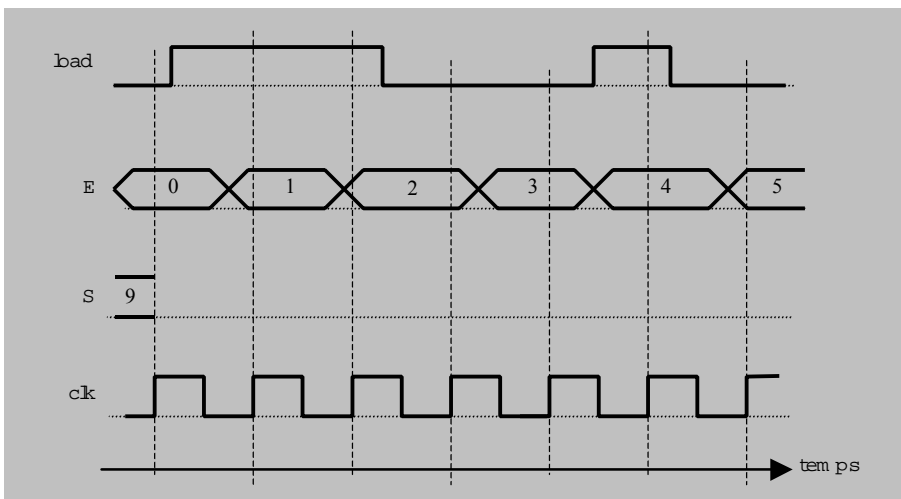
- Veient com s'ha construït el circuit B2 i la seva funcionalitat, deduïu la taula de veritat del bloc B1.
- Utilitzant el mateix bloc B1 s'ha construït aquest altre circuit, B3. Descriu quina funció realitza aquest circuit, i compareu-lo amb el circuit B2.



c) Completeu el cronograma següent, que correspon al circuit B3. Si interpretem les entrades x i y en cada cicle de rellotge com els diferents bits d'una parella de número A i B, quins són aquests números A i B?



13. La figura següent mostra els valors dels senyals E i $load$ d'un registre de 8 bits durant un cert interval de temps. Indiqueu en la línia etiquetada com a *temps* els instants en què el registre es carrega amb l'entrada E i la seqüència de valors que prendrà la sortida S del circuit. Observeu que inicialment el valor d' S és 9.

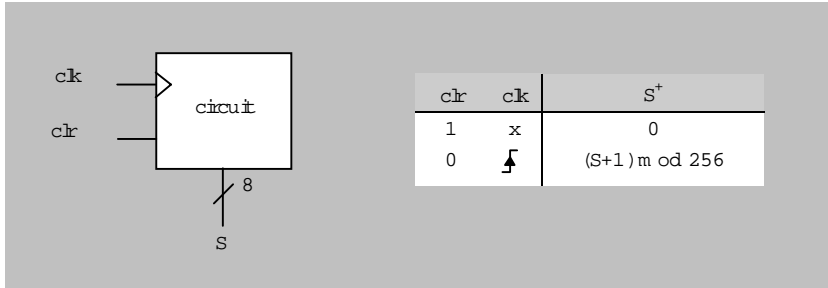


En un cronograma, els valors de mots d'n bits es dibuixen mitjançant un hexàgon, dins del qual s'escriu el valor del senyal en decimal. Els punts de contacte entre els extrems dels hexàgons indiquen el moment en què el mot canvia de valor.

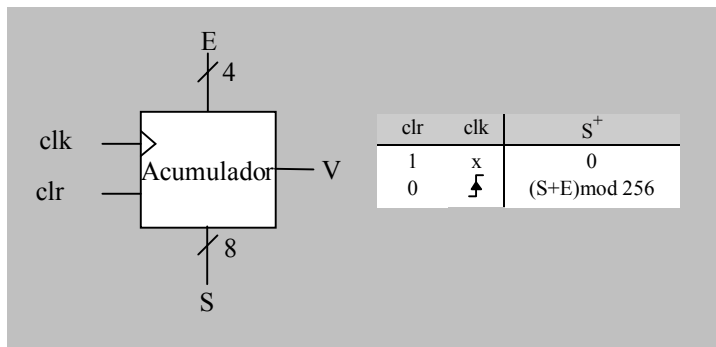
14. Utilitzant només un sumador de 8 bits i un registre de 8 bits, dissenyeu un circuit seqüencial que tingui una sortida S de 8 bits. Aquesta sortida és un número natural (codificat en binari) que s'ha d'incrementar a cada flanc del rellotge segons aquesta expressió:

$$S^+ = (S+1) \bmod 256$$

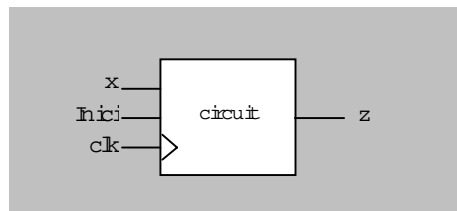
(per tant, després de valdre 255 passa a valdre 0). El circuit disposa, a més, d'un senyal asíncron clr que quan val 1 posa la sortida a 0. El funcionament del circuit es mostra a la figura següent.



15. Utilitzant només un sumador de 8 bits i un registre de 8 bits, dissenyeu un circuit seqüencial que faci d'acumulador de números naturals. Aquest circuit té una entrada E de 4 bits i una sortida S de 8 bits. El valor d'aquesta sortida ha de canviar amb cada flanc del rellotge, segons aquesta expressió: $S^+ = (S+E) \bmod 256$. L'acumulador té una sortida d'un bit, V , que indica si s'ha produït sobreiximent en la suma. El circuit disposa, a més, d'un senyal asíncron clr que quan val 1 posa la sortida S a 0. El funcionament del circuit es mostra a la figura següent.



16. Utilitzant un registre de 4 bits, blocs combinacionals i portes, dissenyeu un circuit seqüencial que reconegui si s'ha produït la seqüència de valors "1010" en una entrada x d'un bit (el valor d' x es llegeix a cada flanc de rellotge, com és habitual). En reconèixer la seqüència, el circuit ha de posar el senyal de sortida z a 1. El circuit disposa d'un altre senyal d'entrada, *Inici*, que en començar a funcionar el circuit fa un pols a 1 durant un cicle de rellotge, i després es manté a 0 fins al final.



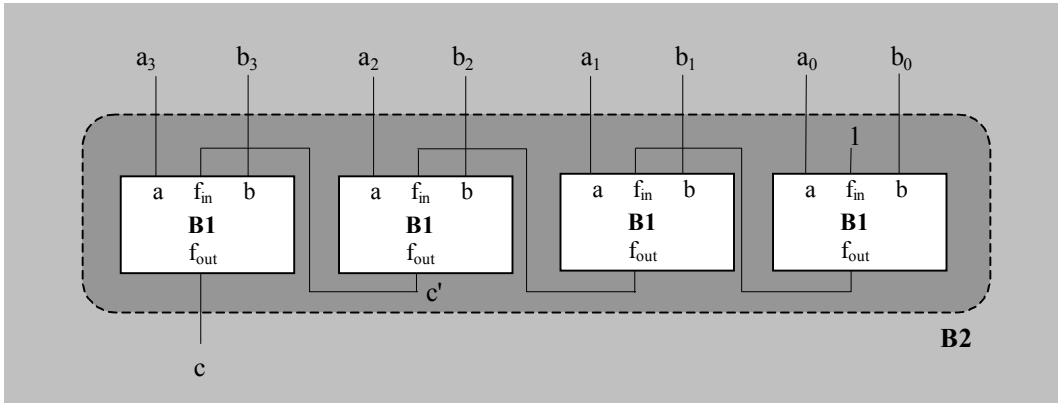
17. Analitzeu què fa el circuit de la figura, tenint en compte que:

- X i S són números enters representats en complement a 2
- L'entrada X té un nou valor a cada cicle de rellotge
- R_{n-1} es refereix al bit de més pes del registre REG1
- El senyal *Inici* funciona com és habitual.

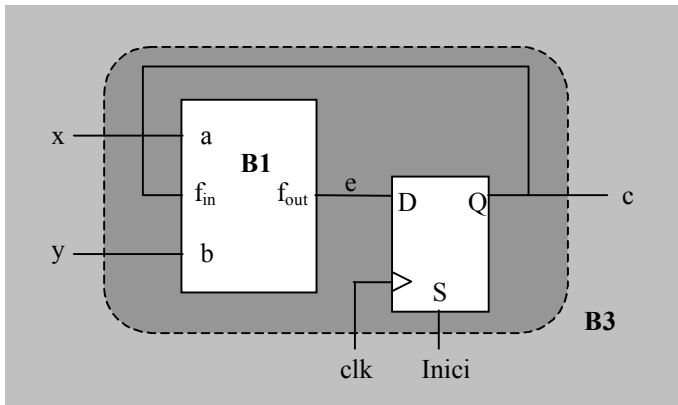
18. El circuit combinacional B2 és un comparador de dos números naturals representats en binari A i B. El seu funcionament és el següent:

$c = 0$ si A és menor que B

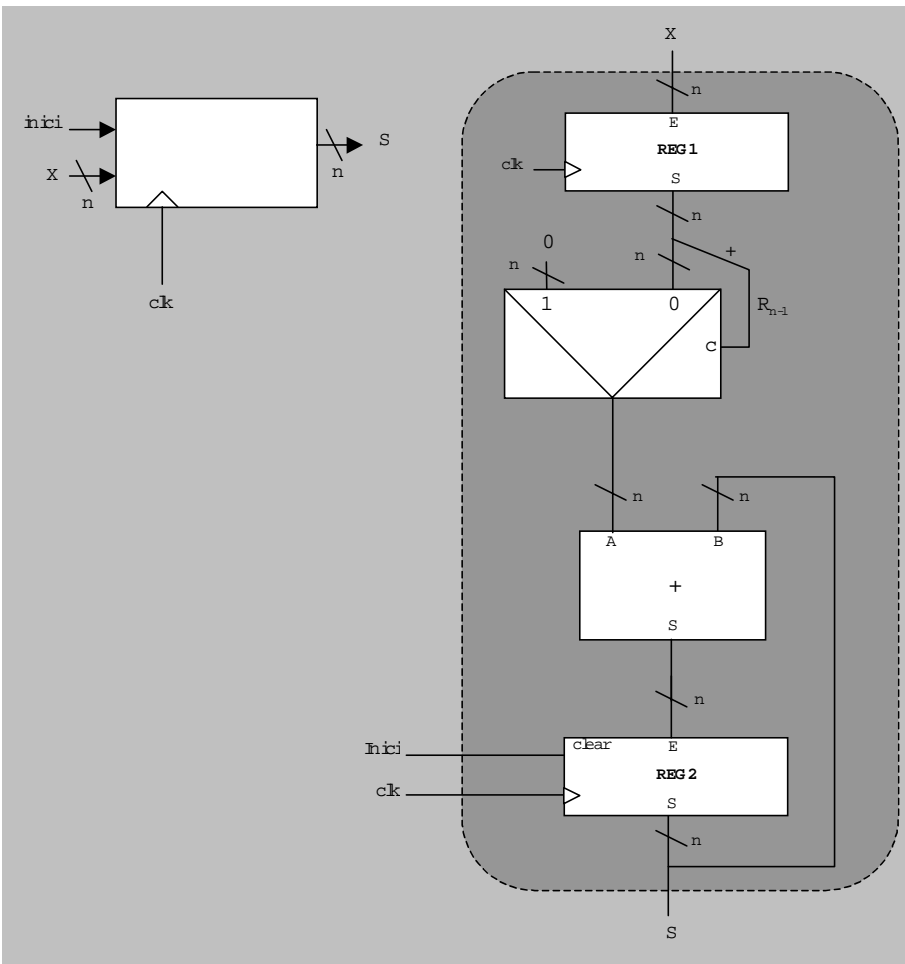
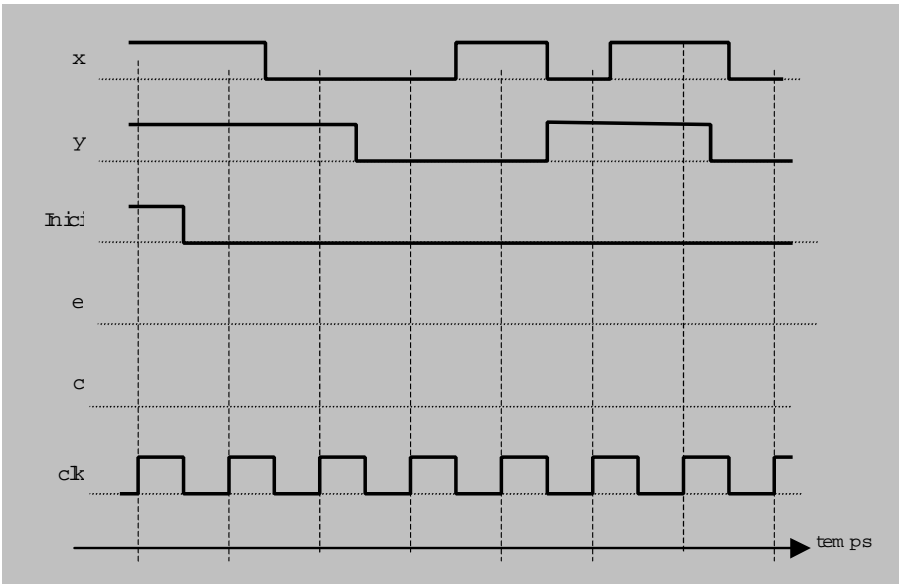
$c = 1$ si A és més gran o igual que B



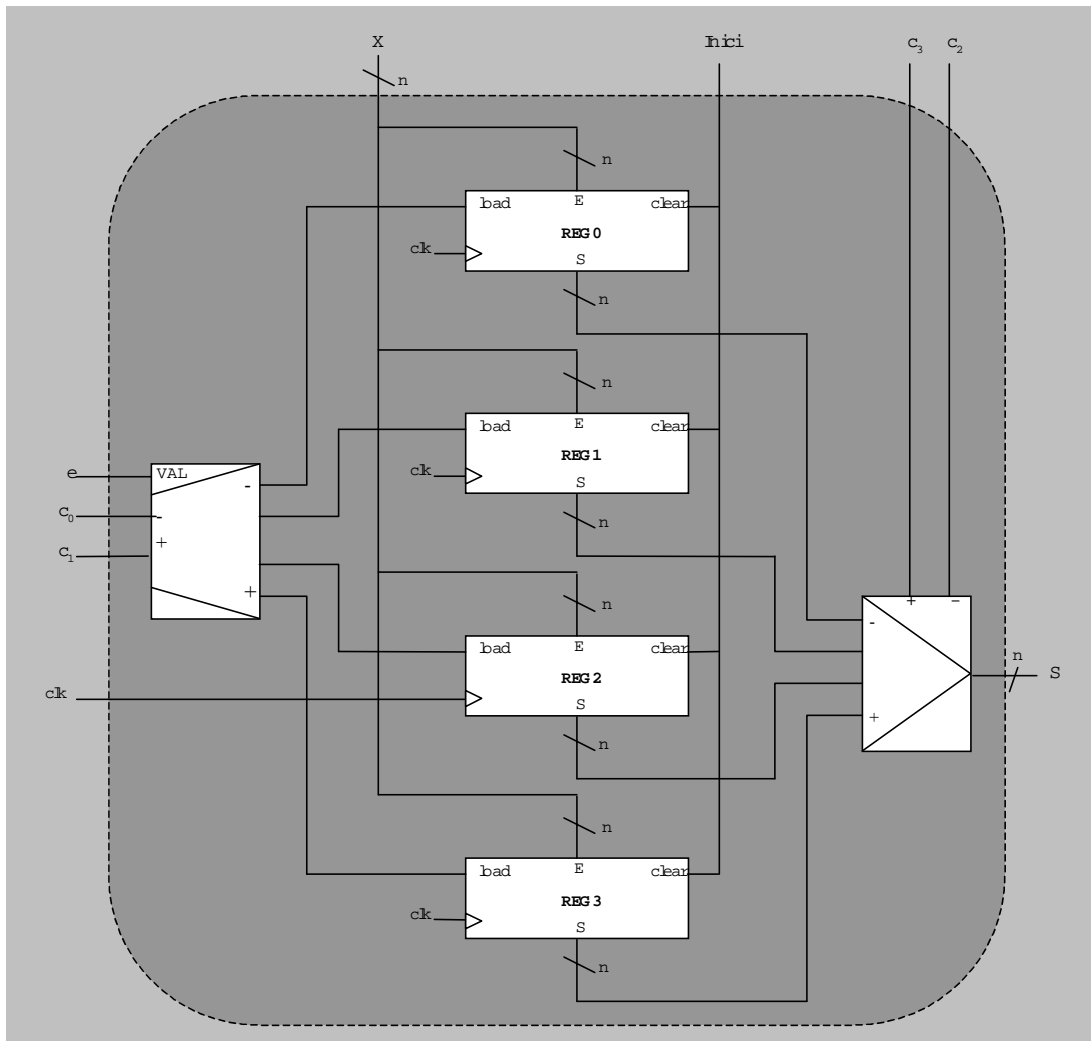
- d) Veient com s'ha construït el circuit B2 i la seva funcionalitat, dedueu la taula de veritat del bloc B1.
- e) Utilitzant el mateix bloc B1 s'ha construït aquest altre circuit, B3. Descriviu quina funció realitza aquest circuit, i compareu-lo amb el circuit B2.



- f) Completeu el cronograma següent, que correspon al circuit B3. Si interpretem les entrades x i y en cada cicle de relotge com els diferents bits d'una parella de número A i B, quins són aquests números A i B?

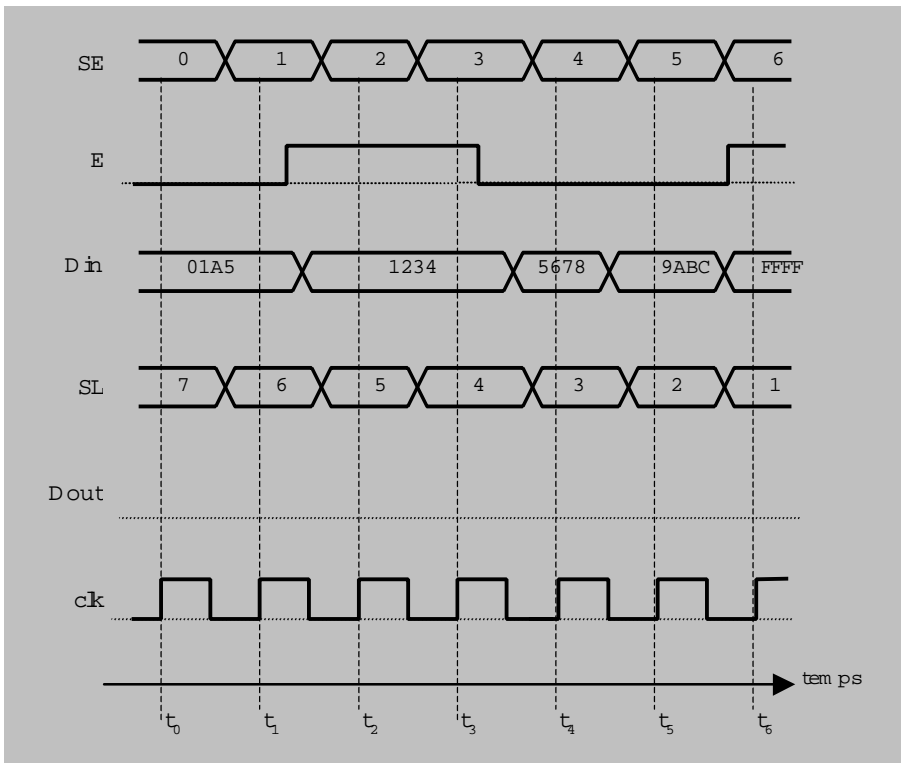


19. Analitzeu què fa el circuit de la figura següent:



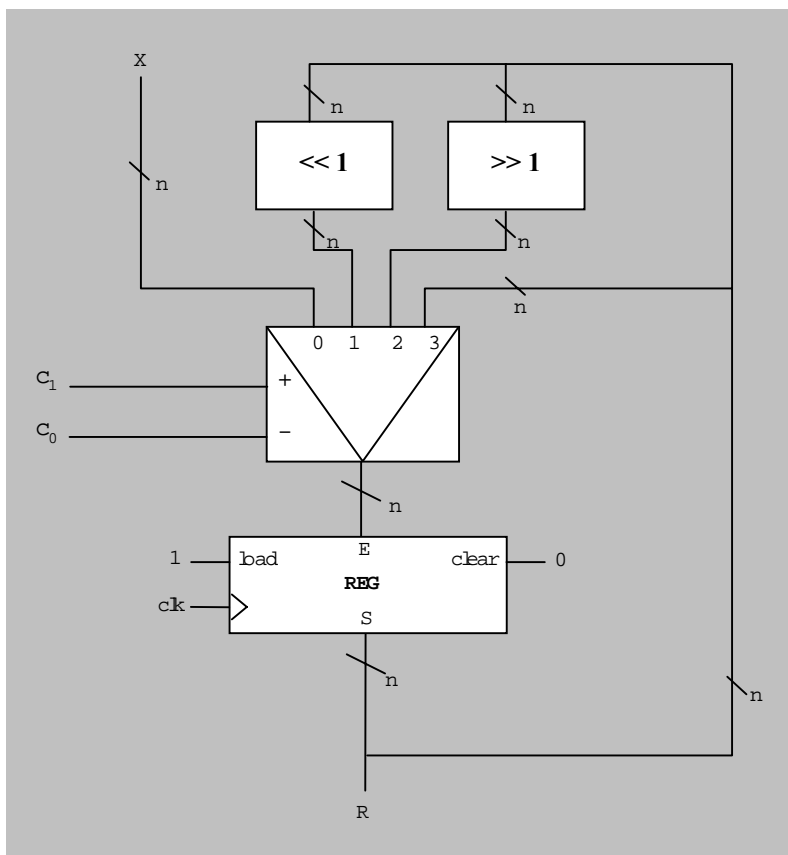
20. Es disposa d'un banc de registres de 8 registres de 16 bits. Donat el cronograma que es mostra a continuació,
- Indiqueu quins registres s'escriuen, en quin moment i amb quin valor.
 - Indiqueu en el cronograma el valor de la sortida *Dout* suposant que el valor inicial dels registres, expressat en hexadecimal, és el següent:

R0=0000
 R1=1111
 R2=2222
 R3=3333
 R4=4444
 R5=5555
 R6=6666
 R7=7777



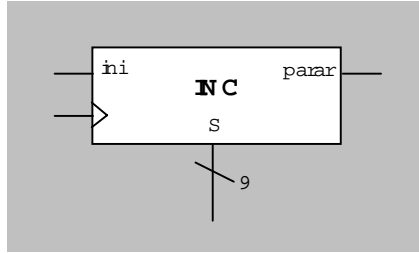
c) Indiqueu el valor de tots els registres del banc després de l'instant t_6 .

21. Implementeu un circuit amb la mateixa funcionalitat que el del problema 19, però sense el senyal *Inici*, usant només un banc de 4 registres.
22. Suposant que els decaladors son aritmètics, indiqueu quina funció fa el següent circuit en funció dels bits c_1 i c_0 .



23.

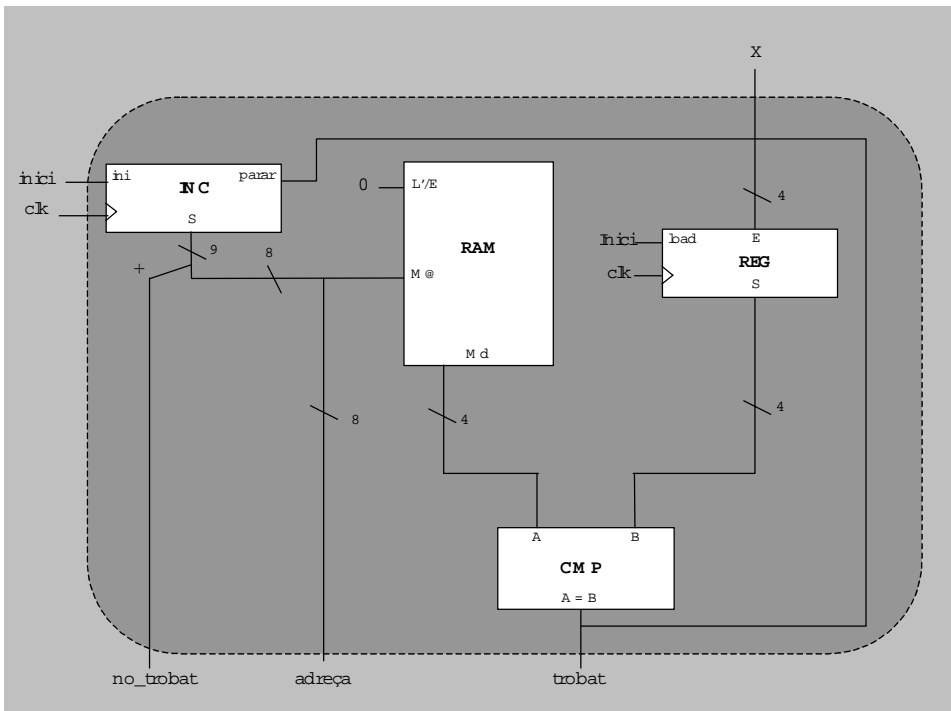
- a) Dissenyu un bloc incrementador, INC, amb una sortida S de 9 bits (semblant al que s'ha dissenyat al problema 15, que funcioni de la manera següent:



- quan el senyal *ini* està a 1 es posa S a 0
- a cada flanc de rellotge, S s'incrementa en 1
- si en algun moment el senyal *parar* val 1, llavors S deixa d'incrementar-se i manté el seu valor a la sortida
- quan S arriba a 256 també deixa d'incrementar-se, i la seva sortida es manté a 256 fins que el senyal *ini* es torni a posar a 1

A partir del circuit de la figura següent (el bloc INC és el dissenyat a l'apartat a)), contesteu aquestes preguntes:

- b) Indiqueu quines són les entrades i sortides del circuit, i quants bits té cadascuna.
 c) Quins blocs del circuit son combinacionals i quins seqüencials?
 d) Quina és la grandària de la memòria RAM del circuit?
 e) Indiqueu quina funció fa aquest circuit i raoneu la resposta (recordeu que *Inici* fa un pols a 1 quan el circuit es posa en marxa).



PROBLEMES DEL MODEL DE MOORE

Especificació de Sistemes Lògics Seqüencials mitjançant grafs d'estats.

24. Especifica (fes el graf d'estats) d'un sistema lògic seqüencial que generi cíclicament la seqüència de números 0,1,3,4,7, codificats en binari. Quants bits d'entrada i sortida tindrà el circuit?

25. Especifica un sistema secuencial que, en funció de una entrada x de un bit, genere per la sortida las siguientes secuencias periódicas de números (codificados en binario):

$ent = 0 \Rightarrow Salida = 0,5,7,11,14,\dots$ (la secuencia se repite a partir de aquí)

$ent = 1 \Rightarrow Salida = 7,11,14,\dots$ (la secuencia se repite a partir de aquí)

La entrada ent puede cambiar de valor en cualquier momento.

26. Se quiere diseñar un *reconocedor de secuencia* con salida $s=1$ durante un ciclo de reloj al detectar la secuencia '0110' por la línea de entrada z . Haz el grafo de estados del sistema suponiendo que:

a) El circuito no detecta solapamiento entre dos secuencias consecutivas. Es decir, si entra la secuencia 0110110, la segunda secuencia no debe poner la salida s a 1 porque el primer bit está solapado con el último bit de la secuencia anterior.

b) El circuito detecta solapamiento entre dos secuencias consecutivas. Es decir, si entra la secuencia 0110110, la segunda secuencia debe poner la salida s a 1 porque el primer bit está solapado con el último bit de la secuencia anterior.

c) Realiza un cronograma para los casos a) y b) si la secuencia de entrada es 001011011011 (los bits de esta secuencia entran en el circuito de izquierda a derecha)

27. Especifica un sistema lògic seqüencial que posi la sortida a 1 durant un cicle quan hagin arribat per l'entrada dos o més 0s consecutius, o bé quatre o més 1s consecutius. Inicialment la sortida ha de valdre 0. Abans de fer el disseny, fes un cronograma amb l'exemple i comprova que la sortida Y esta un cicle desplaçada con respecte a la entrada X . ¿Quin és el motiu?

Per exemple:

X 10001011111011...

Y 00011000001100...

(s'entén que els bits entren d'esquerra a dreta)

28. Un sistema lògic seqüencial rep a l'entrada dígitos decimales ([0..9]) codificats en BCD (un per cicle). Quan ha arribat la seqüència 121, la sortida es posa a 1 durant un cicle. Dibuixa el graf d'estats del sistema.

29. Contesta las siguientes preguntas:

a) Especifica un sistema secuencial que reconozca cualquiera de las dos secuencias de tres bits '101' y '010' y detecte el posible solapamiento entre ellas. El sistema tiene una entrada x de un bit. Cuando se detecta una de las secuencias especificadas se pone el bit de salida s a 1.

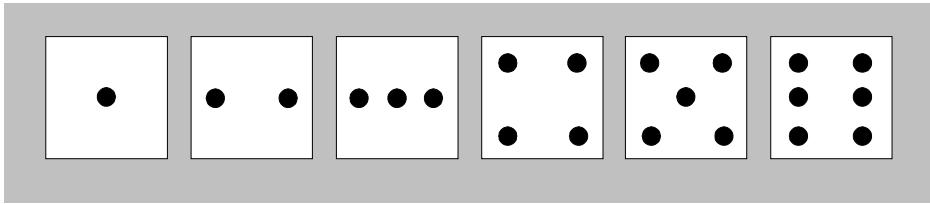
b) Realiza un cronograma en el que se especifique el reloj, x y s para la secuencia de entrada 101101001011

30. Especifica un sistema lògic seqüencial en què la seqüència 101 a l'entrada (d'un bit) provoqui que la sortida es posi a 1, i la seqüència 001 a l'entrada provoqui que la sortida es posi a 0. Inicialment la sortida ha de valdre 1.

31. Un sistema tiene una entrada x de un bit. Dibuja el grafo de estados del sistema sabiendo es capaz de detectar, poniendo un '1' en la salida s , que han entrado un número par de unos o un número impar de ceros.

32. Es vol dissenyar un sistema lògic seqüencial per a tirar un dau. Les sortides del sistema estaran connectades als punts d'un "dau electrònic" de la manera següent:

Les combinacions possibles del dau són les següents:



Mentre el senyal d'entrada jugar estigui activat, el sistema generarà cíclicament les 6 combinacions del dau. Quan el senyal jugar estigui desactivat, no es produirà cap transició. Així, quan un jugador deixi de polsar el senyal jugar veurà al dau la combinació que li ha sortit (se suposa que la freqüència del rellotge és molt alta i no es poden arribar a distingir les combinacions intermitges).

Dibuixa el graf d'estats del sistema.

33. Es vol dissenyar el comandament del control remot d'un cotxe de juguina. El comandament té dos botons: E i D.

Si el cotxe està parat, en polsar-se qualsevol botó es posa en moviment: gira a l'esquerra si es polsa E, gira a la dreta si es polsa D, i tira endavant si es polsen tots dos alhora.

Mentre el cotxe està en moviment, si es polsa E

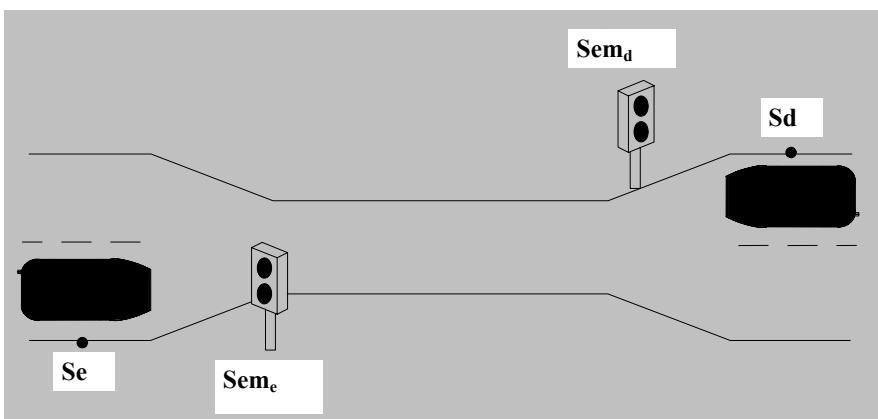
- girarà a l'esquerra si anava recte,
- anirà recte si estava girant a la dreta,
- seguirà girant a l'esquerra si ja ho feia

i similarmet quan es polsi D. Quan es polsin tots dos botons alhora es parerà.

Especifica un sistema lògic seqüencial que rebí com a entrada 2 senyals connectats als botons E i D (1: polsat; 0: no polsat) i generi dos senyals Z1 i Z0 que governaran el cotxe, segons la taula següent:

Acció del cotxe	Z1	Z0
Girar a la dreta	0	0
Girar a l'esquerra	0	1
Parar	1	0
Moure's endavant	1	1

34. Dibuixa el graf d'estats d'un sistema lògic seqüencial que controli els semàfors d'un pont pel qual només poden passar cotxes en un sentit.



A cada extrem del pont hi ha:

- un sensor que posa el senyal Sd (o Se) a 1 quan hi ha un cotxe.

- un semàfor que es posa vermell quan li arriba un 0 pel senyal sem_d (o sem_e) i es posa verd quan li arriba un 1.

El sistema ha de funcionar de la manera següent:

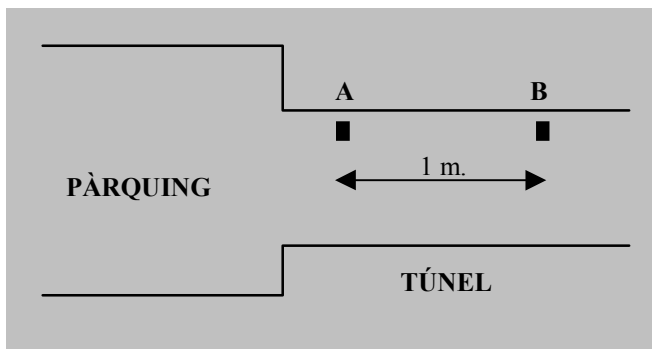
- Els semàfors es posen verds de forma alternada durant 2 cicles de rellotge com a mínim
- Després del segon cicle de rellotge, el semàfor que està verd s'hi queda en verd fins que no arribi un cotxe per l'altre extrem.

35. Un pàrquing necessita comptar el nombre de cotxes que hi ha en cada moment.

Els cotxes entren i surten del pàrquing pel mateix túnel, en el qual només hi cap un cotxe. Al túnel hi ha dos sensors, A i B, separats un metre, de manera que es pot saber si un cotxe està entrant o sortint segons l'ordre en què s'activen els sensors (se suposa que tots els cotxes medeixen més d'un metre i que entre un cotxe i el següent hi ha més d'un metre).

Quan un cotxe ha entrat totalment al pàrquing s'incrementa el nombre de cotxes aparcats, i quan n'ha sortit totalment es decrementa. Un cotxe pot fer marxa enrera en qualsevol moment.

Dos cotxes no es trobaran mai de cara al túnel. Tampoc no hi haurà mai peatons.



Dissenya un sistema lògic seqüencial que generi els senyals COMPTAR i DESCOMPTAR a partir dels senyals A i B que envien els sensors.

36. Se desea diseñar un sistema secuencial para controlar un paso a nivel con barreras como el que se muestra en la figura 1.

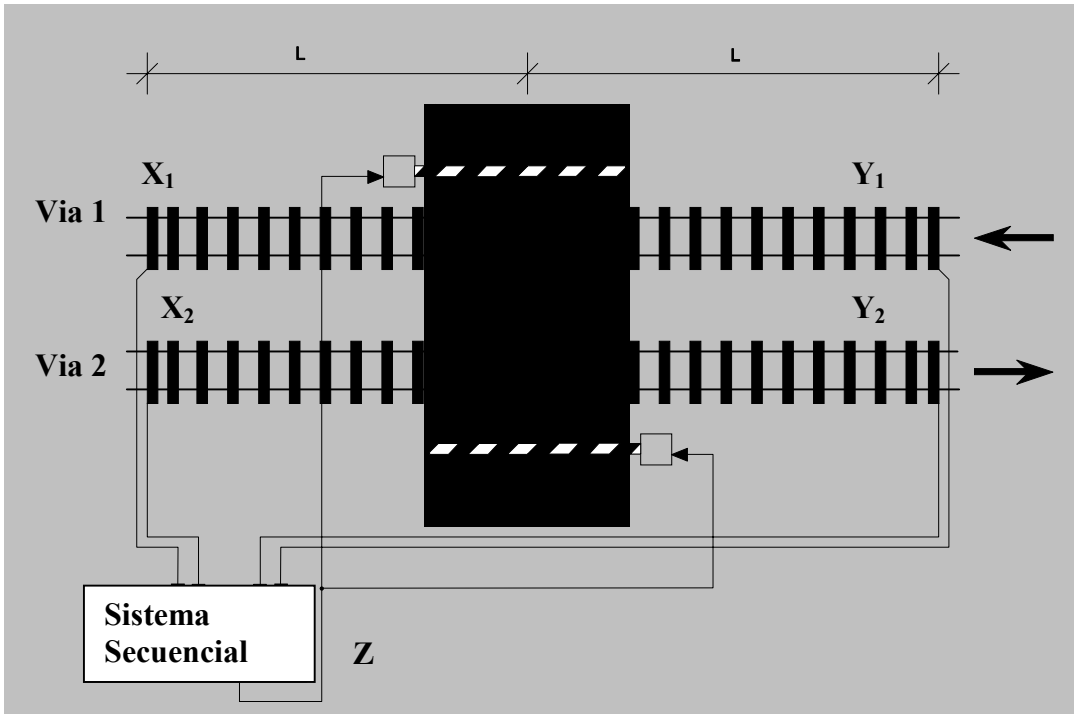


Figura 1

Los trenes siempre circulan en los sentidos indicados en la figura 1. Se han colocado sobre las vías cuatro sensores (X_1, X_2, Y_1, Y_2) que envían una señal de valor '1' mientras un tren pasa por encima suyo.

El sistema tiene como salida la señal Z , que toma el valor '1' cuando las barreras están cerradas y '0' cuando están abiertas. Las barreras se han de cerrar ($Z=1$) cuando un tren encuentre el primer sensor (Y_1 para la vía 1 y X_2 para la vía 2), y se han de abrir ($Z=0$) cuando encuentre el segundo sensor. En la figura 2 puede verse un ejemplo.

Nota: la longitud de los trenes es menor que L (un mismo tren no puede estar activando simultáneamente Y_1 y X_1 , $i=1,2$) y los trenes que circulan por la misma vía están separados por una distancia mayor que $2L$, por lo que las combinaciones '1x1x' y 'x1x1' están prohibidas.

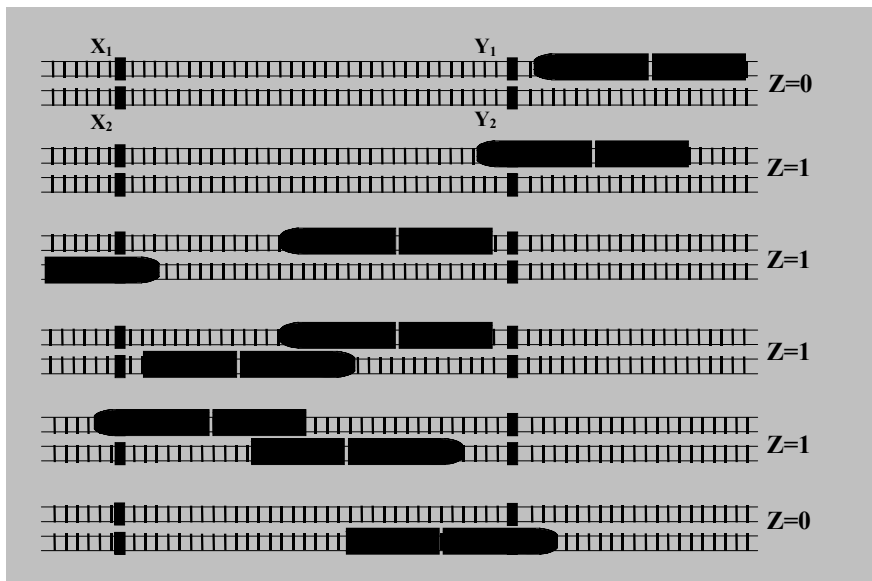
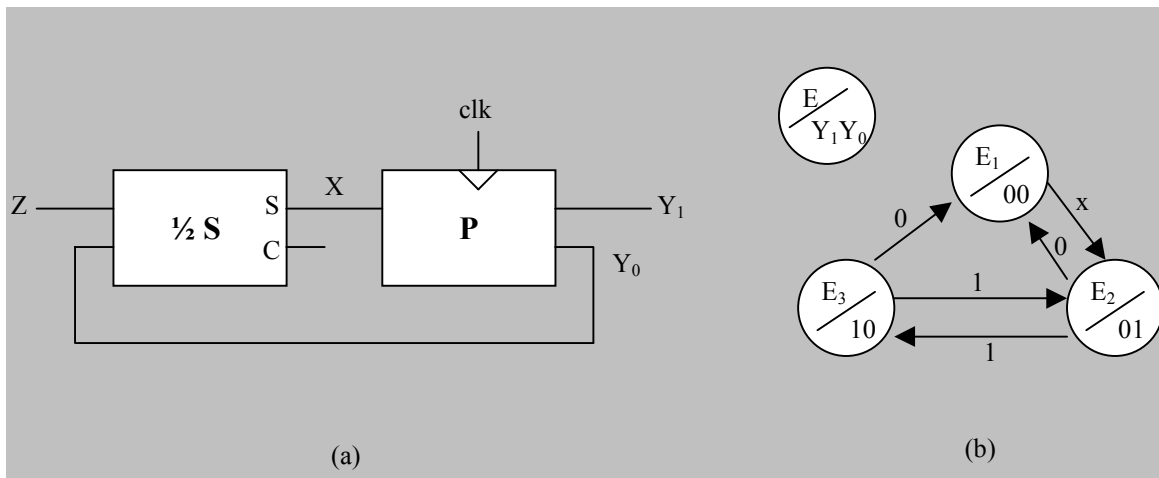


Figura 2

- Realiza el grafo de estados para la vía 1.
- Realiza el grafo de estados para las dos vías.

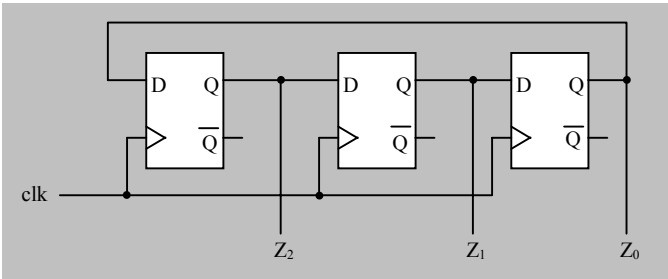
c) ¿Cómo se podría resolver, de forma fácilmente escalable, el problema para n vías, con $n > 5$?

37. Dibuixa el graf d'estats del sistema de la figura (a), sabent que P és un sistema lògic seqüencial que es comporta segons el graf de la figura (b).



Anàlisi de Sistemes Lògics Seqüencials

38. Fes l'anàlisi (dibuixa el graf d'estats) que descriu el funcionament del sistema lògic seqüencial següent.

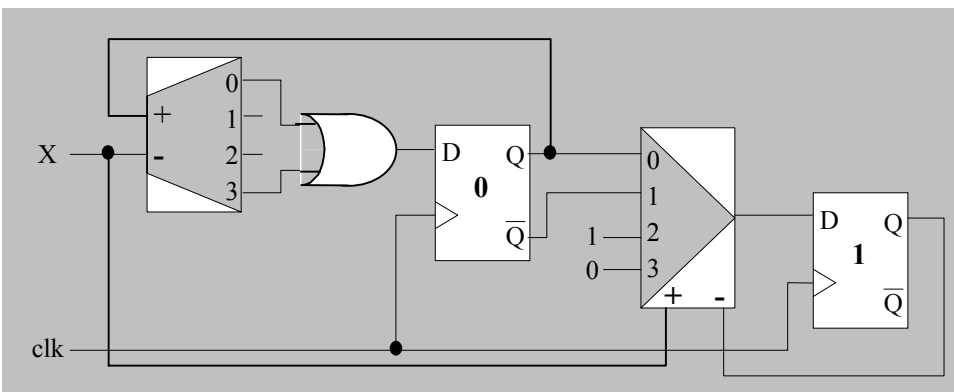


39. Suposant que la memòria ROM de la figura (a) té el contingut que es mostra a la figura (b), dibuixa el graf d'estats del sistema lògic seqüencial de la figura (a).

@	M[@]		
	D ₂	D ₁	D ₀
0	0	0	1
1	0	1	0
2	0	1	1
3	1	0	0
4	1	0	1
5	0	0	0
6	0	0	0
7	0	0	0
8	1	0	1
9	0	0	0
10	0	0	1
11	0	1	0
12	0	1	1
13	1	0	0
14	0	0	0
15	0	0	0

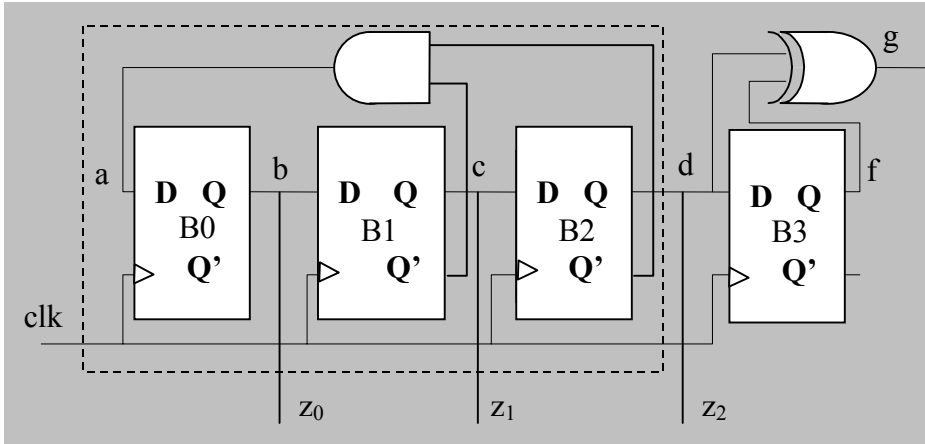
(a)
(b)

40. Donat el circuit següent:



- a) Encuentra las expresiones algebraicas de D_0 y D_1 .
- b) Escribe la tabla de transiciones y la tabla de salidas del circuito.
- c) Dibuja el grafo de estados asociado al circuito.

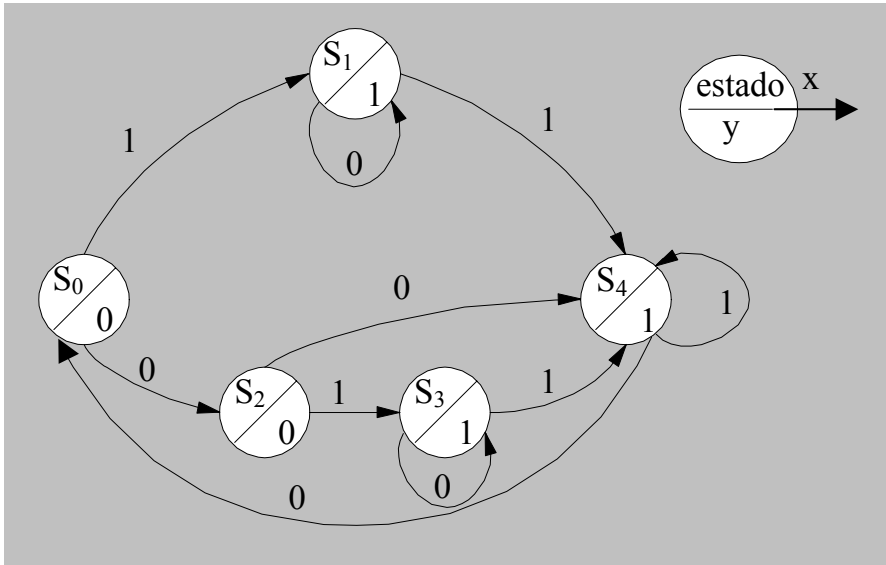
41. Dado el circuito de la figura, contesta las siguientes preguntas:



- a) Suponiendo que inicialmente las salidas de los biestables B0, B1 y B2 valen '0', y que la salida del biestable B3 vale '1', dibuja un cronograma que exprese el funcionamiento del sistema formado por los biestables de la zona punteada (dibuja en el cronograma las señales en los puntos a, b, c, d, f, y g).
- b) ¿Cuántos estados tiene el sistema enmarcado por la línea de puntos? Dibuja el grafo de estados correspondiente.

Implementació de Sistemes Lògics Seqüencials

42. Dado el siguiente grafo de estados, contesta a las siguientes preguntas:



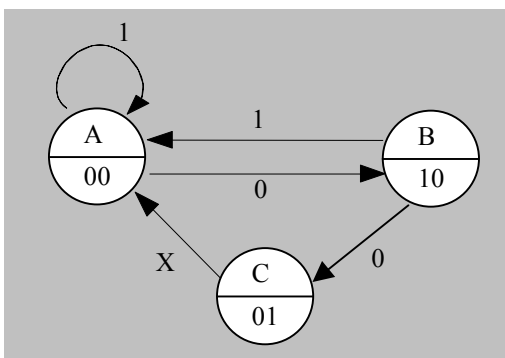
- Escribe la tabla de transiciones de estados.
- Escribe la tabla de salidas.
- Implementa todo el sistema con biestables de tipo D y el mínimo número de puertas posible.

43. Considera el grafo de estados del problema 26b.

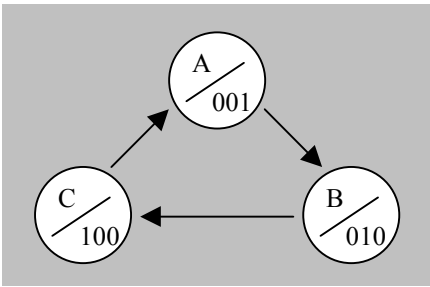
- Implementa el sistema con el menor número posible de biestables tipo D y las puertas lógicas necesarias.
- Implementa el sistema con un registro de longitud adecuada y las puertas lógicas y bloques combinatoriales que consideres necesarios.
- Compara las dos soluciones, considerando:
 - la facilidad en el diseño,
 - el coste,
 - la facilidad de modificar cualquier parámetro del diseño.

44. Implementa els sistema lògic seqüencials que descriuen cadascun dels grafos d'estats següents. En cada cas, fes-ho codificant els estats per tal de minimitzar el nombre de biestables. Utilitza una memòria ROM per dissenyar la part combinatorial del circuit, indicant el seu contingut i la seva mida.

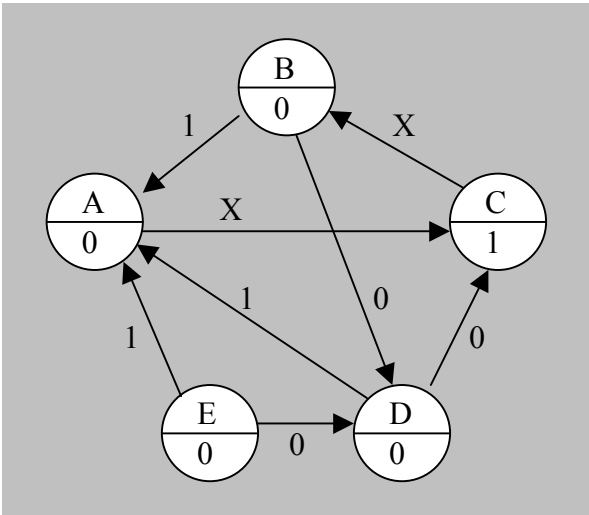
a)



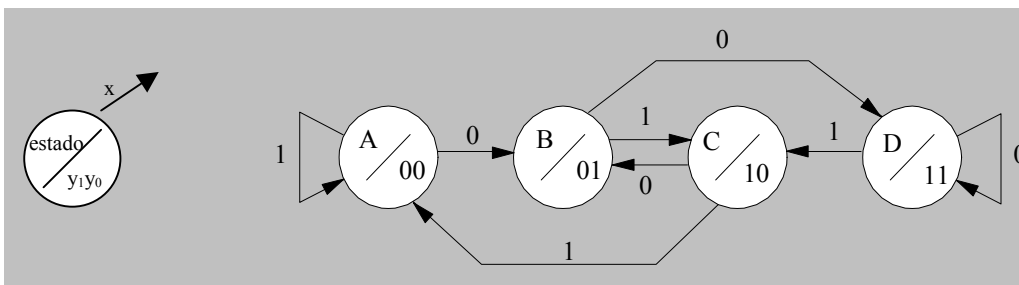
b)



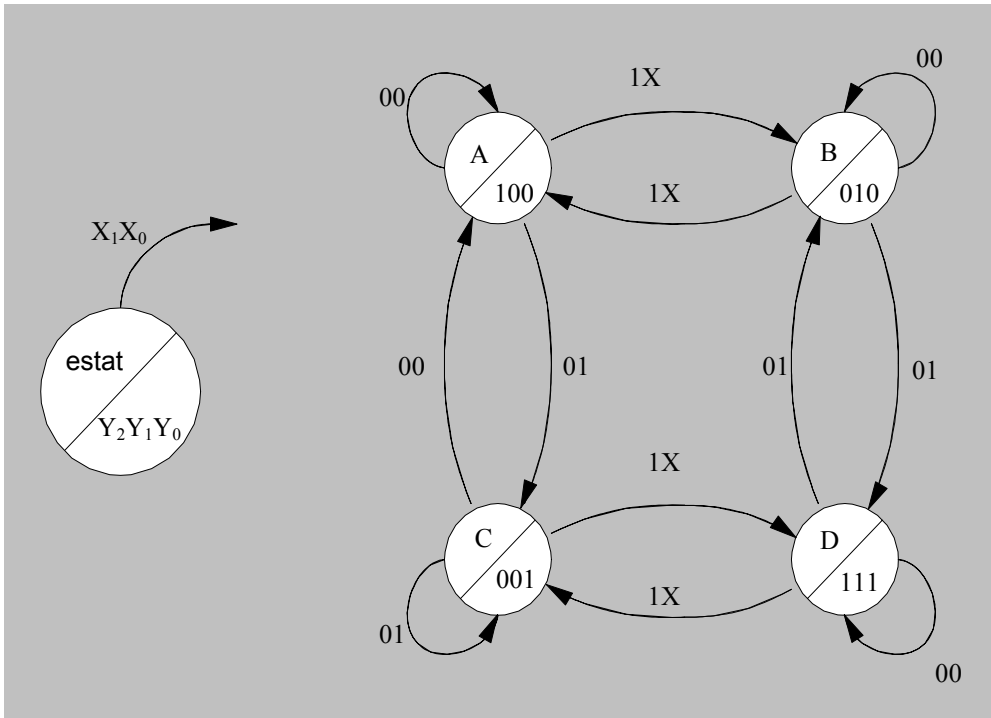
c)



45. Implementa el sistema secuencial siguiente usando el menor número posible de biestables D y una memoria ROM del tamaño necesario. Indica el tamaño de la memoria.



46. Fes la síntesi del sistema lògic seqüencial següent, usant 2 biestables i una memòria ROM. Indica el contingut i la mida de la memòria.



47. Fes el disseny del graf d'estats del problema 24 utilitzant portes lògiques i una memòria ROM de la mida que calgui. Indica clarament la grandària i contingut de la ROM. Introdueix un senyal asíncron INICIO que faci que el sistema comenci la seqüència generant un 1 per la seva sortida.
48. Dissenya i implementa amb el menor nombre possible de biestables un sistema lògic seqüencial que funcioni com un comptador reversible mòdul 8. Comptarà endavant o endarrera en funció d'un senyal d'entrada.

$X = 0$: comptar endavant

$X = 1$: comptar endarrera

La sortida es codificarà segons aquesta taula:

	Sortida
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Utilitza una memòria ROM per a implementar les funcions de transició, i dos blocs combinacionals (diferents d'una ROM) per a implementar les sortides.

Solucionari

1.

Es tracta de reconèixer si els 4 bits que arriben per l'entrada del sistema valen "1010" o no. Per a saber-ho n'hi ha prou amb examinar el valor del mot d'entrada en el moment actual. Per tant, el sistema és de tipus combinacional.

En canvi, si l'entrada del circuit fos d'un bit el circuit hauria de ser de tipus seqüencial, perquè en cada moment hauria de recordar els 3 últims valors que han arribat per l'entrada, a més de l'actual.

2.

Un dígit decimal (rang del 0 al 9) requereix 4 bits per ser codificat. L'entrada del sistema a dissenyar, per tant, pot llegir un dígit en cada moment.

Com que s'ha de detectar una seqüència de quatre dígits, aquests han d'entrar un darrera l'altre per l'entrada del sistema, i el circuit ha de recordar dels dígits que han entrat amb anterioritat per tal de reconèixer la seqüència.

Es tracta doncs, clarament, d'un sistema seqüencial.

3.

El període T és l'invers de la freqüència: $T = 1/F$

Com que la freqüència és de $650 \cdot 10^6$ Hz, el període és

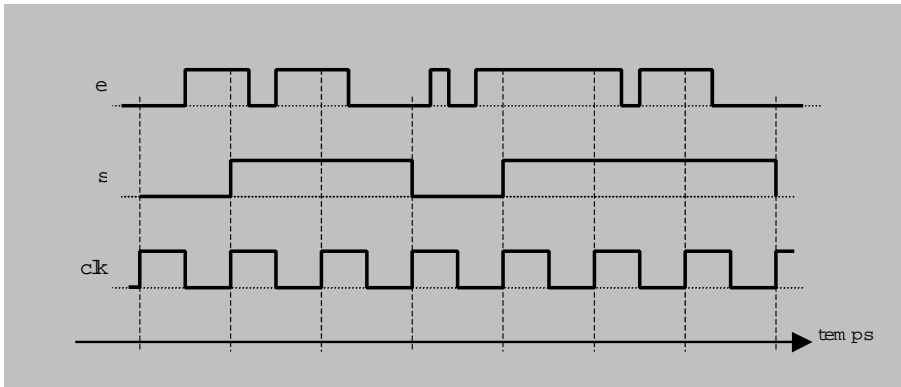
$$T = 1/(650 \cdot 10^6) \text{ s} = 0'00154 \cdot 10^{-6} \text{ s} = 1'54 \cdot 10^{-9} \text{ s} = 1'54 \text{ ns (nanosegons)}$$

5.

Com es pot veure al cronograma, a la sortida s del biestable hi ha el valor llegit de l'entrada e en cada flanc ascendent del rellotge.

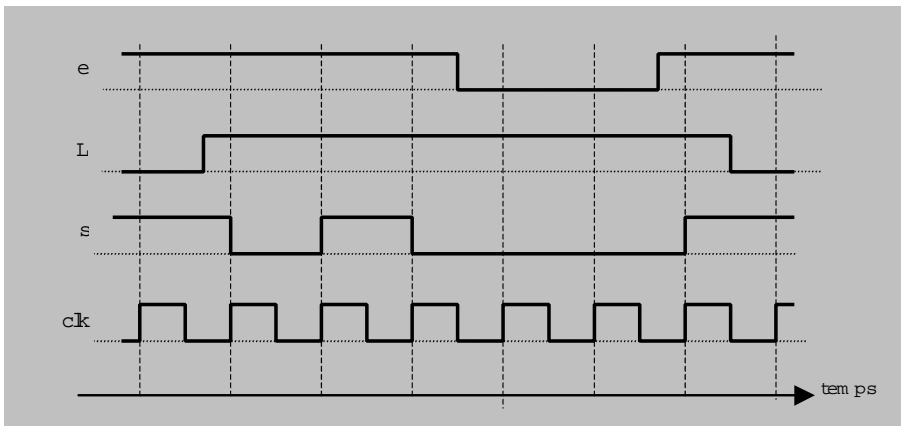
Fixeu-vos que la sortida del biestable només canvia en els flancs ascendents del rellotge (és a dir, canvia de forma *síncrona*), mentre que l'entrada pot canviar en qualsevol moment (és a dir, canvia de forma *asíncrona*).

El valor d' s no és conegut abans del primer flanc de rellotge, perquè no es coneix el valor que hi havia a l'entrada D del biestable en l'instant del flanc anterior.



6.

El senyal e està connectat a una porta AND amb la sortida connectada a l'entrada D del biestable. Per tant, sempre que e valgui 0 arribarà un 0 a aquesta entrada. Podríem dir, doncs, que fa el paper d'un senyal de *reset* síncron. Mentre e val 1, el biestable inverteix el seu valor a cada cicle. El senyal L , connectat a l'entrada *load*, inhibeix la càrrega del biestable quan es posa a 0.

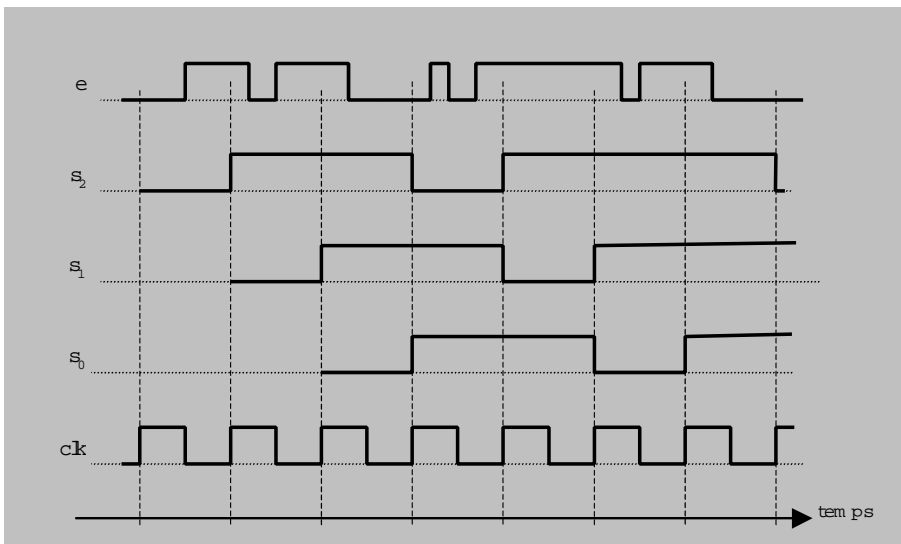


7.

En aquest cas es pot veure que l'entrada e es va "desplaçant" successivament per les sortides s_2 , s_1 i s_0 dels biestables.

La millor manera de fer aquest tipus de cronogrames és dibuixar primer la línia corresponent al senyal s_2 , que només depèn d' e , sencera. Una vegada aquesta està dibuixada, es dibuixa la del senyal s_1 , que depèn només del senyal s_2 . Finalment es dibuixa la línia corresponent al senyal s_0 , que depèn només del senyal s_1 .

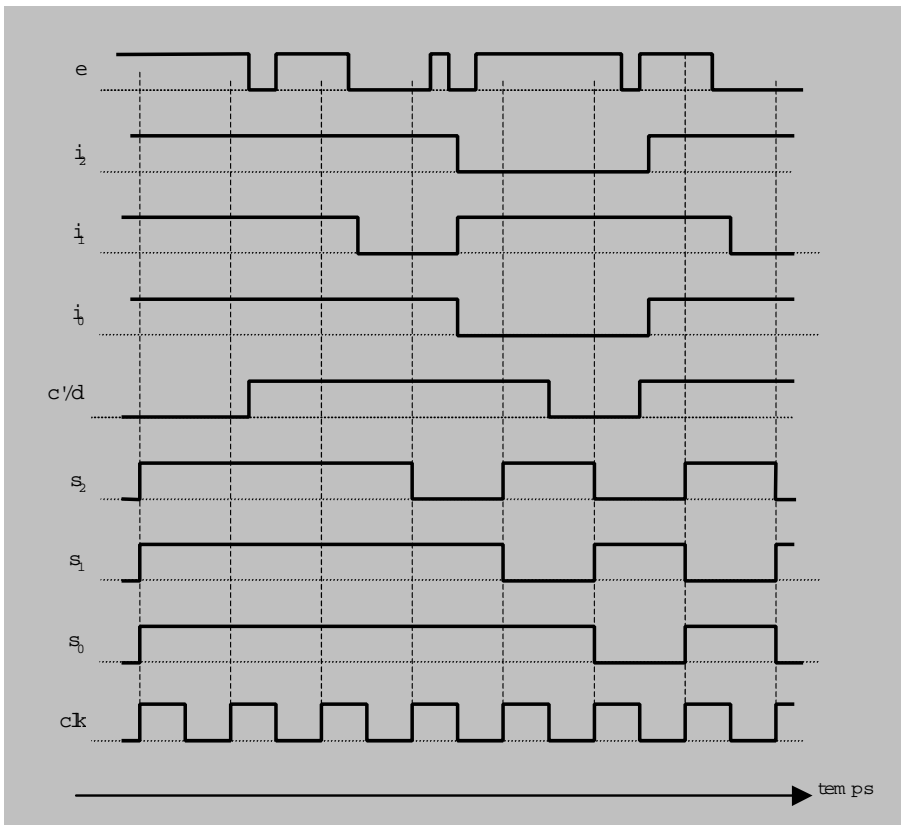
Notem que el valor d' s_1 no es pot determinar fins al segon flanc de rellotge, i el d' s_0 fins al tercer.



8.

Com es pot veure al cronograma següent, quan l'entrada $c'd$ val 0 les sortides del circuit (s_2 , s_1 i s_0 , que corresponen al valor guardat a cadascun dels biestables) prenen el valor de les entrades i_2 , i_1 i i_0 respectivament.

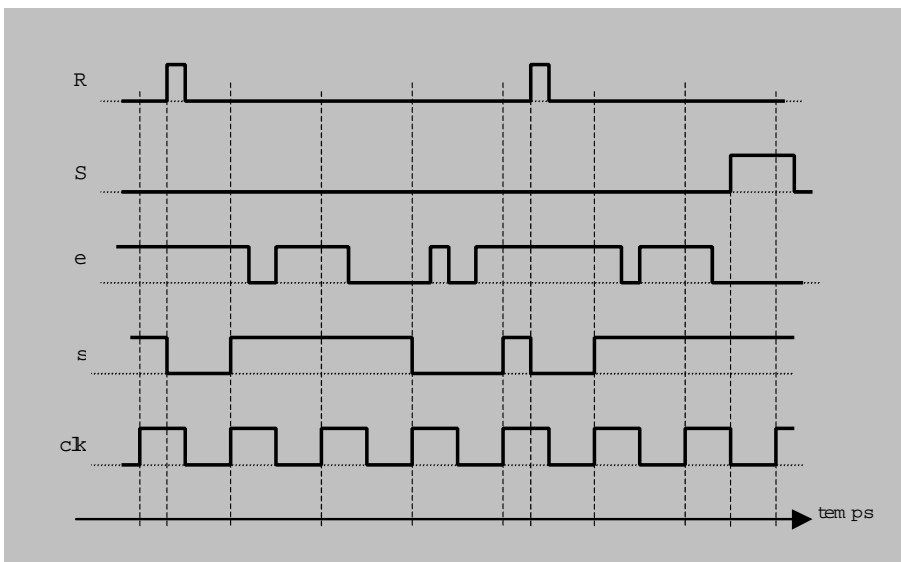
D'altra banda, quan l'entrada $c'd$ val 1 les sortides del circuit prenen el valor del biestable de l'esquerra. Per tant, es produeix un desplaçament dels bits emmagatzemats una posició cap a la dreta. El bit de més a l'esquerra, s_2 , es carrega amb el valor que hi hagi a l'entrada e .



9.

Per resoldre aquest exercici hem de tenir en compte que les entrades asíncrones tenen prioritats sobre les síncrones, és a dir, primer s'avaluen els valors dels senyals R i S del biestable, i només quan ambdues estan a 0 s'avalua el valor present a l'entrada D en cada flanc de rellotge (com que el biestable no té senyal de càrrega, assumim que està a 1).

Per aquesta raó, al cronograma estan marcats amb línies verticals els moments en què canvien les entrades asíncrones, a més dels flancs ascendents del rellotge.



10.

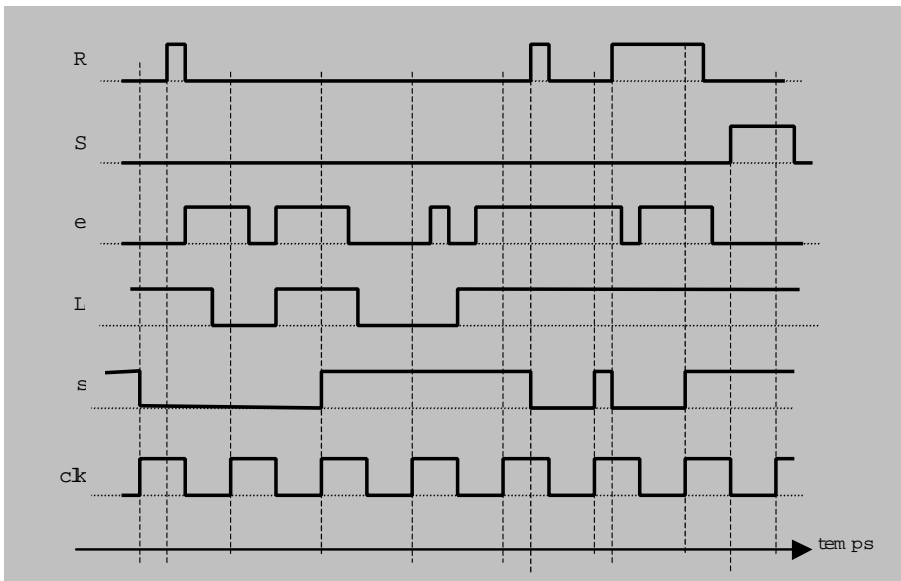
A l'entrada D del biestable hi arriba o bé la sortida del biestable (quan L val 0) o bé el senyal e (quan L val 1).

Com que el biestable es carrega a cada flanc de rellotge, tenim que quan $L = 0$ s'hi carrega el mateix que hi havia. L'efecte és que la seva sortida Q no canvia.

Pel contrari, quan $L = 1$ s'emmagatzema al biestable el valor del senyal e .

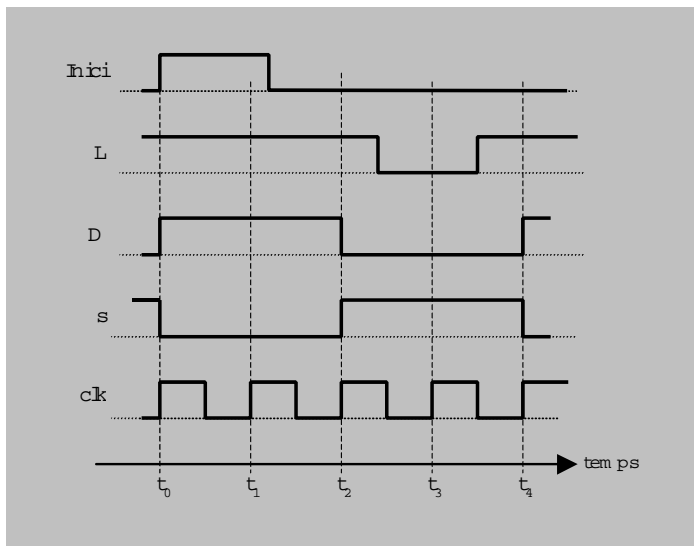
Per tant, podem dir que aquest circuit es comporta igual que un biestable amb senyal de càrrega, en el qual el senyal L fa el paper de l'entrada *load*.

Fixem-nos que després de l'últim flanc el biestable es manté a 1, encara que $L = 1$ i $e = 0$. Això és degut a que S està a 1. Al cronograma estan marcats els moments en què canvien les entrades asíncrones a més dels flancs de pujada del rellotge.



11.

Inicialment, $D = 0$, perquè s està a 1. En l'instant t_0 *Inici* es posa a 1, i per tant el biestable es posa a 0. S'hi quedarà almenys fins a t_2 , perquè en t_1 *Inici* segueix valent 1. En t_2 es carrega amb el que hi ha a l'entrada D , és a dir un 1. En t_3 , el biestable no es carrega, perquè $L = 0$. En t_4 es carrega amb el que hi ha a l'entrada D , és a dir un 0.



12.

a) El bit c ha de valdre 0 si $A < B$ i 1 si $A \geq B$. Aquest bit c està connectat a la sortida f_{out} del bloc B1 de més a l'esquerra. Vegem quant ha de valdre aquesta sortida f_{out} :

- si $a_3 = 0$ i $b_3 = 1$ llavors $A < B$, de manera que c (i per tant f_{out}) ha de valdre 0
- si $a_3 = 1$ i $b_3 = 0$, llavors $A > B$, i per tant c (i per tant f_{out}) ha de valdre 1.
- si $a_3 = b_3$, llavors cal saber quant valen els bits de menor pes d'A i B per decidir el valor que haurà de prendre c .

Fixem-nos en el valor del punt c' del circuit. Aquesta seria la sortida del circuit B2 en el cas que els números fossin de 3 bits en lloc de 4. Per tant, $c' = 0$ si $[a_2 a_1 a_0] < [b_2 b_1 b_0]$ i

1 en cas contrari. Aquesta és la informació que fa falta per decidir el valor de c en cas que $a_3 = b_3$. Per tant, com que c' està connectat a l'entrada f_{in} de l'últim bloc B1, obtenim que la taula de veritat del bloc B1 de més a l'esquerra és la següent (en general, la taula de veritat d'un bloc B1 qualsevol serà la mateixa, canviant només a_3 i b_3 per a_i i b_i):

a_3	b_3	f_{in}	f_{out}	
0	0	0	0	$A < B$, ja que $a_3 = b_3$ i $[a_2 a_1 a_0] < [b_2 b_1 b_0]$
0	0	1	1	$A \geq B$, ja que $a_3 = b_3$ i $[a_2 a_1 a_0] \geq [b_2 b_1 b_0]$
0	1	0	0	$A < B$, ja que $a_3 < b_3$
0	1	1	0	"
1	0	0	1	$A \geq B$, ja que $a_3 > b_3$
1	0	1	1	"
1	1	0	0	$A < B$, ja que $a_3 = b_3$ i $[a_2 a_1 a_0] < [b_2 b_1 b_0]$
1	1	1	1	$A \geq B$, ja que $a_3 = b_3$ i $[a_2 a_1 a_0] \geq [b_2 b_1 b_0]$

Donat que els bits de pes -1 no existeixen, el bloc B1 de més a la dreta s'ha de comportar "com si fossin iguals", és a dir, com si " $a_{-1} = b_{-1}$ ". Això s'aconsegueix connectant un 1 a la seva entrada f_{in} .

b)

Hem vist a l'apartat a) que cada bloc B1 fa la comparació de dos bits de A i B . Aquesta comparació consulta, quan no pot decidir només amb els bits a_i i b_i , el resultat de la comparació dels bits de menys pes.

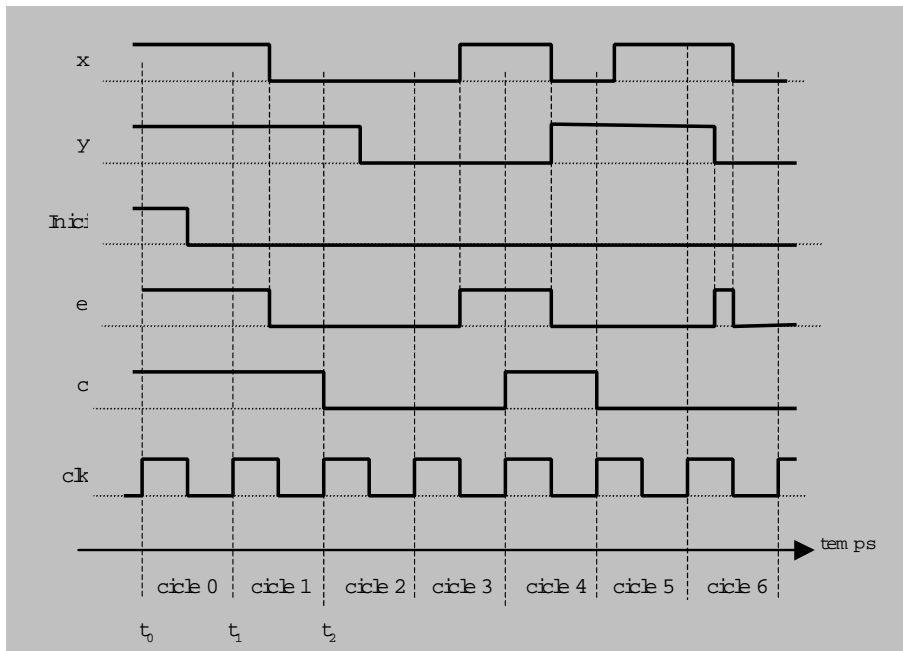
El circuit B3 compara dos bits d'entrada, x i y , i emmagatzema el resultat de la comparació en un biestable. Aquest resultat serà consultat pel bloc B1 quan compari els bits x i y en el següent flanc de rellotge, ja que la sortida del biestable està connectada a l'entrada f_{in} .

Si comparem aquest circuit amb el de l'apartat a) podem veure que poden complir la mateixa funció, si en el primer cicle de rellotge es connecten $[a_0 b_0]$ als punts $[x y]$, en el segon cicle s'hi connecten $[a_1 b_1]$ i així *successivament*. El senyal *Inici* es pot usar per carregar un 1 al biestable en començar a funcionar el circuit, és a dir, pot fer el paper de l'1 que es connecta a l'entrada f_{in} del bloc B1 de més a la dreta en el circuit B2.

La diferència entre els circuits B2 i B3 és que el primer és combinacional mentre que el segon és seqüencial. En el circuit B2, tots els bits dels números A i B es comparen alhora, i el resultat de la comparació estarà disponible immediatament (per ser més precisos, caldrà esperar només el temps de retard de les portes que formen els blocs B1). En canvi, en el circuit B3 els bits d' A i B es comparen parella per parella, de forma seqüencial en el temps. El circuit ha de disposar d'un biestable per poder recordar en cada moment el resultat de la comparació de la parella anterior. El resultat final de la comparació estarà disponible 4 cicles després de començar a funcionar el circuit.

c)

El cronograma es mostra a continuació. Observem que el punt e canvia de valor de forma asíncrona, d'acord amb les variacions en x i y .



Durant el cicle 0, $x = y = 0$. Per tant, $a_0 = b_0 = 1$. Durant el cicle 1, x canvia de valor. El valor que es guardi en el biestable depèn del valor d' x , y i e al final de cicle 1 (en l'instant t_2), i per tant direm que $a_1 = 0$. Seguint el mateix raonament en tots els casos, obtenim que els números que s'han comparat són

$$A = 1001010, B = 1100110$$

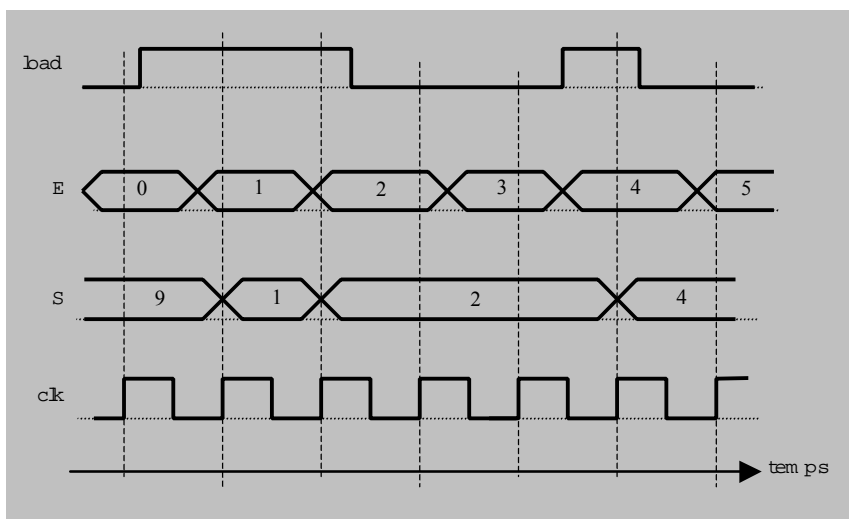
Es compleix que $A < B$. Podem comprovar que la sortida c del circuit val 0 indicant-ho.

13.

Podem observar en el cronograma que el registre només es carrega en els flancs en els quals $\text{load} = 1$. En aquests instants, es carrega amb el valor que hi ha a l'entrada E .

La seqüència de valors que pren la sortida del registre, tal com es mostra al cronograma, és la següent:

9 1 2 4

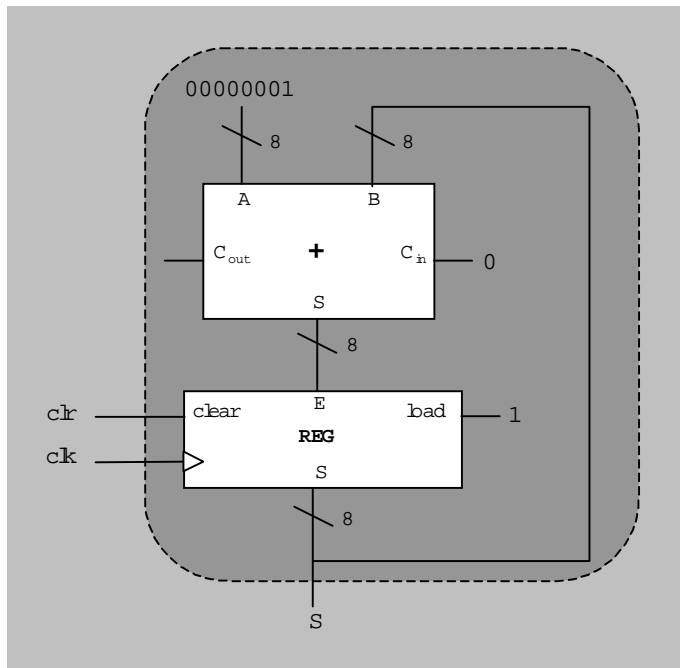


14.

El contingut del registre serà la sortida del circuit. Aquest contingut s'ha d'incrementar en una unitat a cada flanc del rellotge, per la qual cosa connectarem la sortida del registre a l'entrada d'un sumador. A l'altra entrada del sumador hi connectarem un 1. La sortida del sumador està connectada a l'entrada del registre per tal que s'hi guardi al proper flanc de rellotge, tal com es veu a la figura.

Una altra possibilitat és posar un 0 a l'altra entrada del sumador i sumar l'1 a través de l'entrada de transport del sumador, que al disseny mostrat té un 0.

El registre s'ha de carregar sempre que $clr = 0$, i per tant el seu senyal de *load* ha de valdre sempre 1. D'altra banda, el senyal *clr* està connectat a l'entrada *clear* del registre per tal de posar-lo a 0 de forma asíncrona.

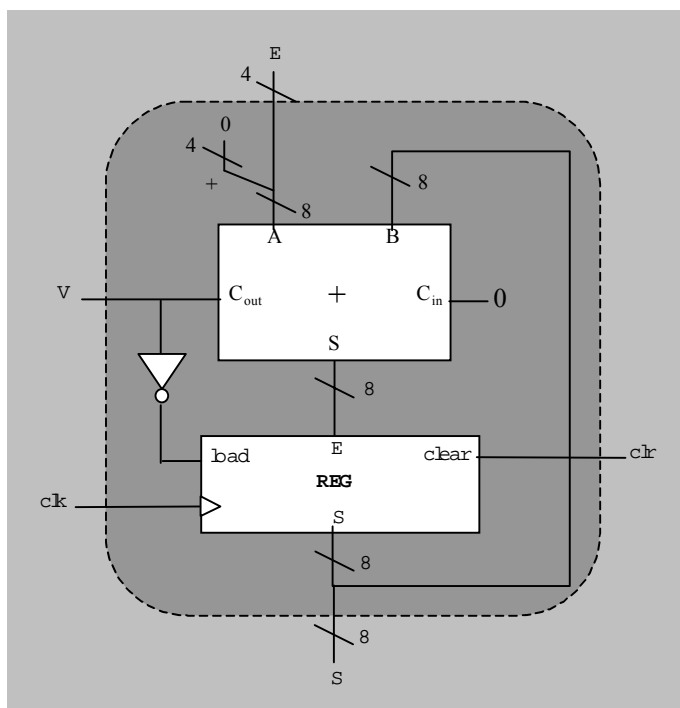


15.

Al registre hi guardarem a cada cicle el resultat de sumar el seu propi contingut més el valor de l'entrada E. Com que aquesta entrada és de 4 bits, li afegirem quatre 0s a l'esquerra abans que arribi a l'entrada del sumador.

El senyal *clr* inicialitza el registre a 0. La sortida *V* es posarà a 1 quan el resultat no pugui ser representat en 8 bits, és a dir, quan el resultat d'alguna suma hagi produït transport.

En el moment que es produeixi sobreiximent, posem l'entrada *load* del registre a 0. Així evitem que se segueixi carregant amb nous valors, la qual cosa podria provocar que el bit *V* tornés a 0, indicant erròniament que el resultat és correcte.



16.

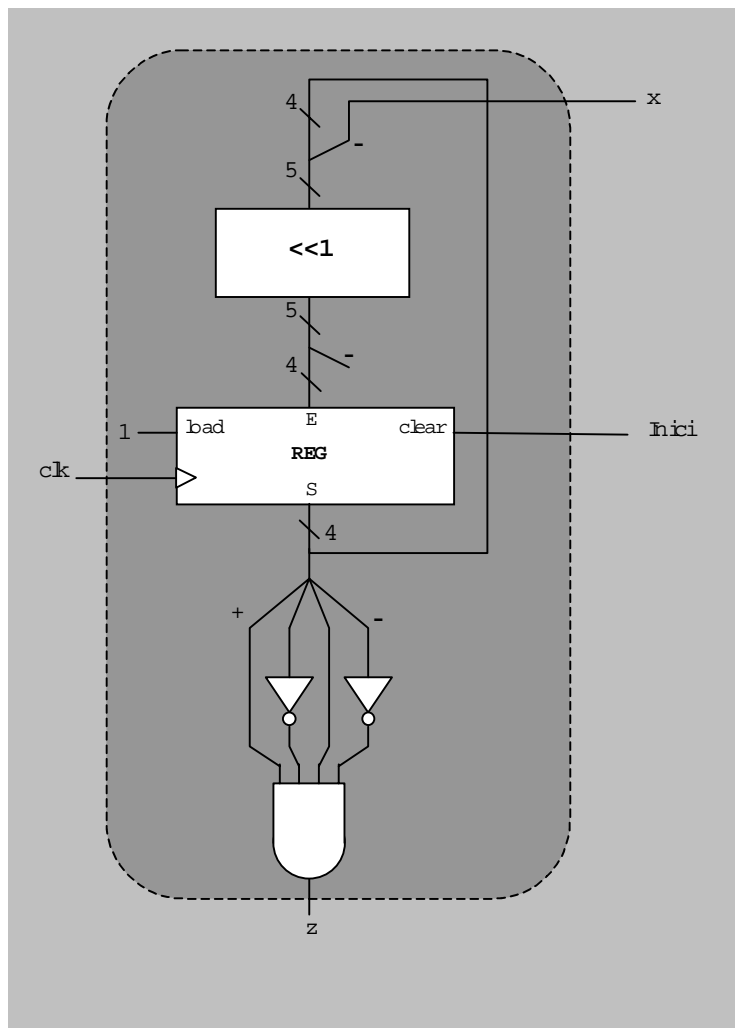
Volem reconèixer si a l'entrada x d'un bit es produeix la seqüència de valors "1010", i disposem d'un registre de 4 bits. Per tant, hem de fer que els valors que vagin arribant per l'entrada x es vagin desplaçant pels biestables del registre, per exemple de dreta a esquerra. El biestable de més a la dreta es carregarà a cada flanc amb el valor d' x . La resta de biestables es carregaran amb el valor que contenia el seu "veí" de la dreta.

Per a aconseguir-ho, farem que el registre es carregui a cada flanc amb el valor que contenia fins ara decalat un bit a l'esquerra, excepte pel bit de més a la dreta, que es carregarà amb el nou valor d' x .

D'aquesta manera, obtenim que la sortida z ha de valdre 1 quan el contingut del registre sigui "1010". Si anomenem $[s_3 s_2 s_1 s_0]$ els bits de sortida del registre, tenim que

$$z = s_3 \cdot s_2' \cdot s_1 \cdot s_0'$$

Inicialment, el registre ha de contenir un 0, altrament es podria reconèixer la seqüència sense que s'hagués produït. Per exemple, si el contingut inicial del registre fos 0101 i el primer valor d' x fos 0, el circuit reconeixeria la seqüència "1010" després del primer flanc de rellotge, de forma errònia. Per això connectem el senyal *Inici* a l'entrada *clear* del registre.



17.

Analitzarem punt per punt el circuit.

- L'entrada X es carrega al registre REG1 a cada flanc de rellotge (com que no s'hi han dibuixat les entrades *load* i *clear*, s'assumeix que valen 1 i 0 respectivament).
- El bit de més pes del registre REG1, R_{n-1} , controla el multiplexor.
- Com que els números estan representats en complement a 2, el bit R_{n-1} és el bit de signe de l'últim número que s'ha emmagatzemat al registre REG1. Aquest bit és 1 si el número és negatiu i 0 si és positiu.
- Per tant, el multiplexor deixa passar el número que hi ha a REG1 si aquest és positiu, i deixa passar un 0 si és negatiu.
- El sumador suma aquest número amb el contingut del registre REG2, que inicialment està a 0, i el resultat es guarda de nou a REG2.
- La sortida S del circuit està connectada a la sortida de REG2.

Podem concloure, doncs, que la sortida S del circuit és la suma de tots els números positius que entren per X .

19.

Veiem que el senyal *Inici* està connectat a totes les entrades *clear* dels registres. Per tant, tots es posen a 0 en començar a funcionar el circuit.

Les entrades *load* de cada registre estan connectades a les sortides d'un descodificador. Per tant, només una d'elles està a 1 en cada moment, si $e = 1$ (el senyal e està connectat a l'entrada de validació del descodificador). Si $e = 0$, llavors no es carrega cap registre.

A les entrades del descodificador hi ha els senyals c_1 i c_0 . Per tant, deduïm que aquests dos senyals controlen quin dels registres es carrega en cada moment amb el valor de l'entrada X , que està connectada a l'entrada de dades de tots els registres.

La sortida S del circuit està connectada a un multiplexor de busos controlat per c_3 i c_2 . Cada entrada de dades del multiplexor està connectada a la sortida de dades d'un dels 4 registres. Per tant, deduïm que c_3 i c_2 controlen quin dels continguts dels 4 registres surt en cada moment per la sortida.

20.

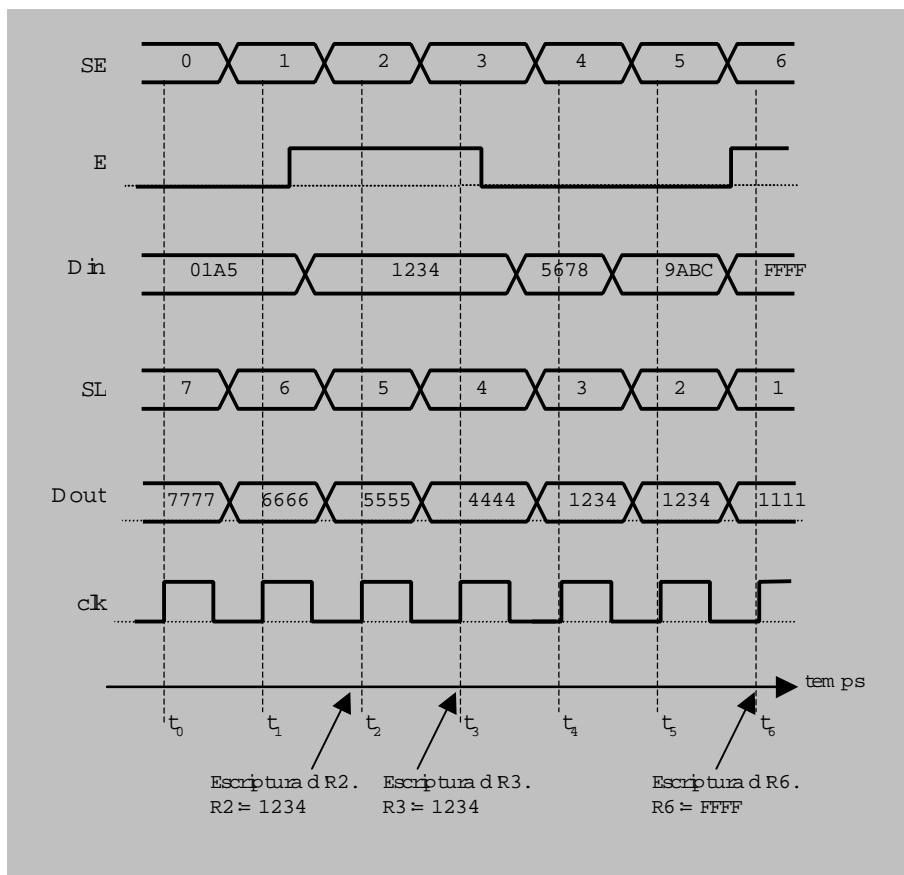
- a) Les escriptures al banc de registres tenen lloc en els flancs ascendents si $E = 1$. Per tant, els instants en què s'escriurà algun registre són t_2 , t_3 i t_6 . Per saber quin registre s'escriu analitzem el valor d' SE en aquests instants; per saber quin valor s'hi carrega, analitzem Din . Obtenim el següent:

en l'instant t_2 , $R2 := 1234$

en l'instant t_3 , $R3 := 1234$

en l'instant t_6 , $R6 := FFFF$

- b) Al cronograma es pot veure la seqüència de valors en $Dout$. Per a deduir-lo, cal mirar en cada moment quin registre s'està llegint (SL) i quin és el seu contingut (recordem que les lectures es fan de forma asíncrona).



- c) El valor final dels registres és el següent:

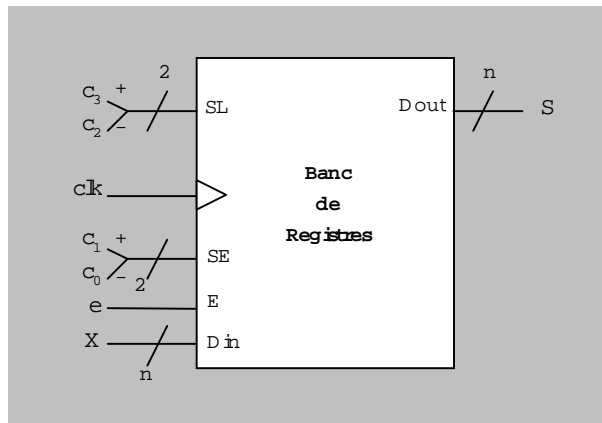
$R0 = 0000$

$R1 = 1111$

R2=1234
 R3=1234
 R4=4444
 R5=5555
 R6=FFFF
 R7=7777

21.

La funcionalitat del circuit del problema 19 (sense l'entrada *Inic*) l'aconguim amb un banc de registres connectant els senyals a les entrades i sortides tal com es mostra a la figura.



Deduïm, doncs, que el circuit del problema 19 correspon a una possible implementació d'un banc de 4 registres, al qual se li ha afegit la funcionalitat d'inicialitzar tots els registres a 0.

22.

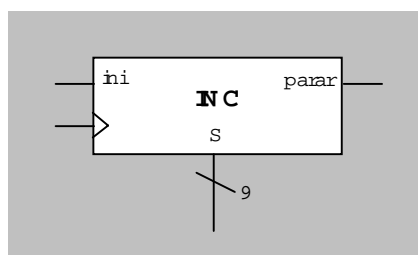
El registre es carrega a cada flanc amb la sortida del multiplexor. El valor d'aquest senyal es pot descriure amb la taula següent:

c_1	c_0	E
0	0	X
0	1	$S \ll 1$
1	0	$S \gg 1$
1	1	S

Per tant, aquest circuit permet emmagatzemar al registre el valor d'entrada X ($c_1=0$, $c_0=0$), multiplicar per 2 el seu valor ($c_1=0$, $c_0=1$), dividir-lo per 2 ($c_1=1$, $c_0=0$) o bé mantenir-lo sense variació ($c_1=1$, $c_0=1$).

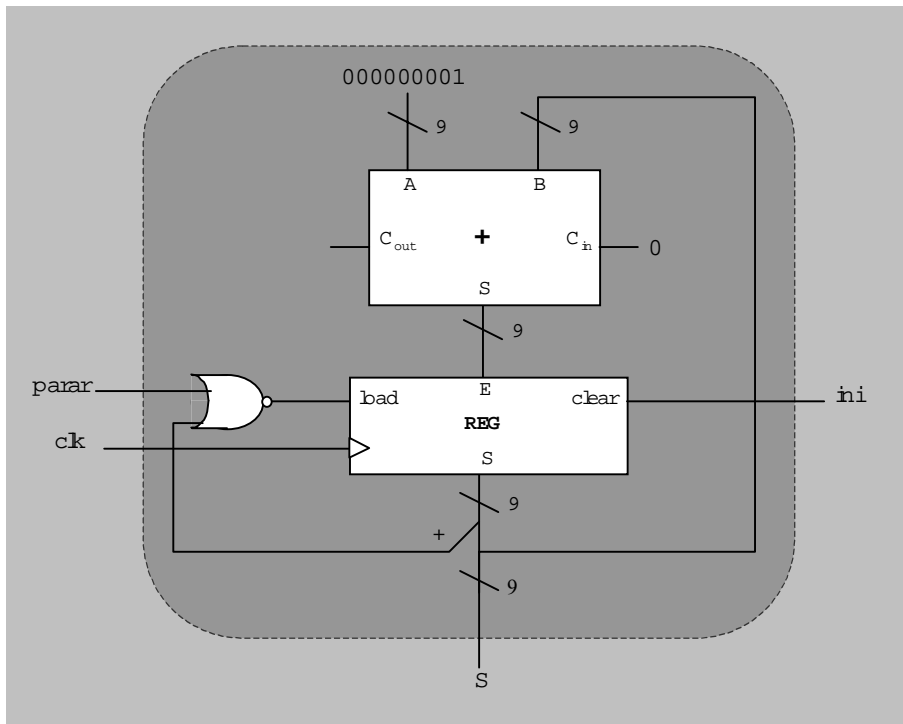
23.

a) El circuit a dissenyar és el següent:



- Per fer aquest bloc utilitzarem, bàsicament, un registre i un sumador. El registre emmagatzemarà la sortida S i el sumador permetrà incrementar-la. Per fer això, a una entrada del sumador hi arribarà el valor d' S i a l'altra un 1.
- El senyal *ini* l'implementarem amb el *clear* del registre

- El senyal *parar* l'implementarem amb el senyal de *load* del registre. Quan *parar* sigui 1 o el bit de més pes d'*S* sigui 1 posarem un 0 a *load*. En qualsevol altre cas, el senyal de *load* serà 1. Per tant, $load = (parar + S_8)'$



b) El circuit té tres entrades i tres sortides.

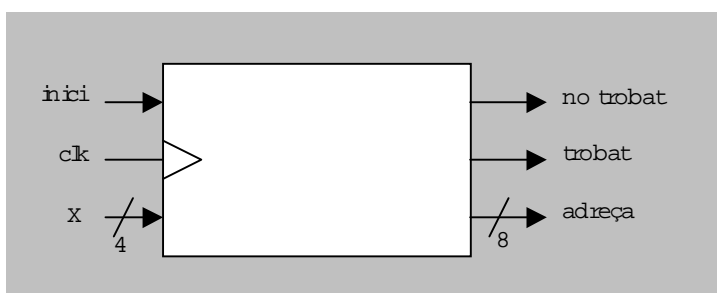
Entrades:

- Una entrada *Inici*, d'1 bit, que indica quan s'ha de començar
- Una entrada de rellotge, *clk*
- Una entrada de dades de 4 bits, *X*

Sortides:

- Una sortida de 8 bits, *adreça*
- Dues sortides d'un bit, *trobat* i *no_trobat*

A continuació es mostra una figura amb aquesta descripció.



c) El bloc CMP és combinacional, i la resta són seqüencials.

d) La grandària de la RAM és de $2^8 \times 4$ bits.

e) Per veure què fa el circuit l'analitzarem per parts:

- En començar a funcionar el circuit, l'entrada *X* s'emmagatzema al registre (perquè *Inici* està connectat a l'entrada *load*)
- El bloc *INC* s'incrementa a cada flanc de rellotge. Comença des de 0 gràcies al senyal *Inici*.
- La sortida del bloc *INC* està connectada a l'entrada d'adreces de la memòria. Com que $L/E = 0$, a cada cicle de rellotge es llegeix un mot de la memòria, des de l'adreça 0 fins a la 255. Si el bloc *INC* arriba a 256 (10000000) parerà d'incrementar-se, i s'haurà recorregut tota la memòria. En aquest moment, el bit de més pes del bloc *INC* (*no_trobat*) valdrà 1, i a *M@*hi haurà un 0 (els 8 bits de menys pes de *INC*).
- El bloc *INC* també es pot parar abans si l'entrada *parar* es posa a 1. Això passa quan el senyal *trobat* val 1.

- La sortida de dades de la RAM es compara en cada instant amb el contingut del registre (l'entrada X). Si són iguals, el senyal *trobat* es posa a 1.
- Com que *trobat* para l'incrementador, aquest es para quan s'ha trobat un mot a la memòria igual al mot d'entrada X . En aquest moment, la sortida *adreça* conté la posició de memòria on s'ha trobat el mot X .

Podem concloure que el què fa aquest circuit és buscar a quina posició de memòria es troba un mot X . Si aquest mot és a la memòria, el senyal *trobat* es posa a 1 i per la sortida *adreça* surt la posició de memòria on s'ha trobat. Si el mot no és a la memòria, el senyal *no_trobat* es posa a 1 i el contingut de la sortida *adreça* no té cap significat.

Soluciones als problemes del Model de Moore

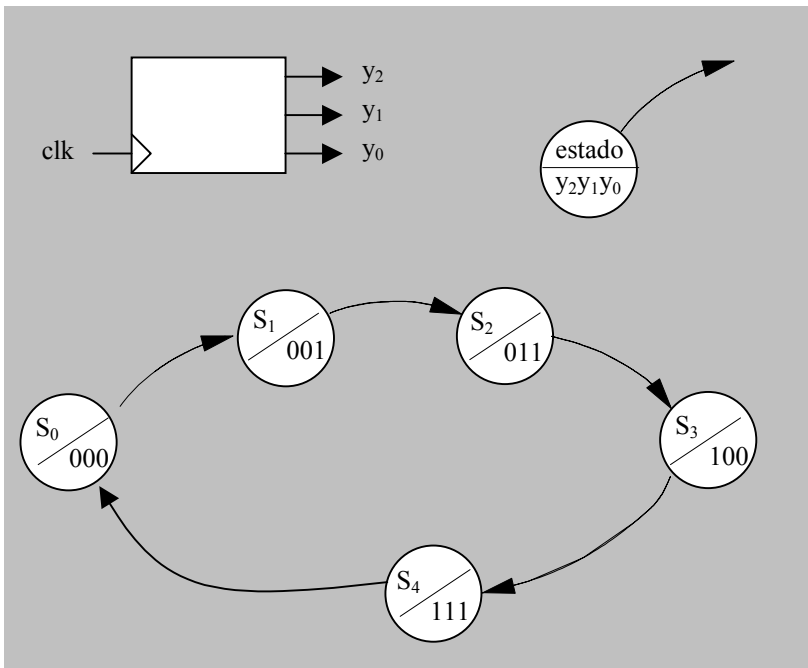
24.

El circuito no tiene ningún bit de entrada, tal como se indica en el enunciado. Por tanto, la secuencia de salida debe producirse siempre en el mismo orden.

Tendrá 3 bits de salida, suficientes para codificar en binario el 7, que es el número más grande de la secuencia.

Dado que en el enunciado no se especifica cuál es el primer número que debe aparecer a la salida, cualquiera de los estados puede servir como estado inicial.

A continuación se presenta un dibujo del sistema desde el punto de vista de entradas/salidas y su grafo de estados.



25.

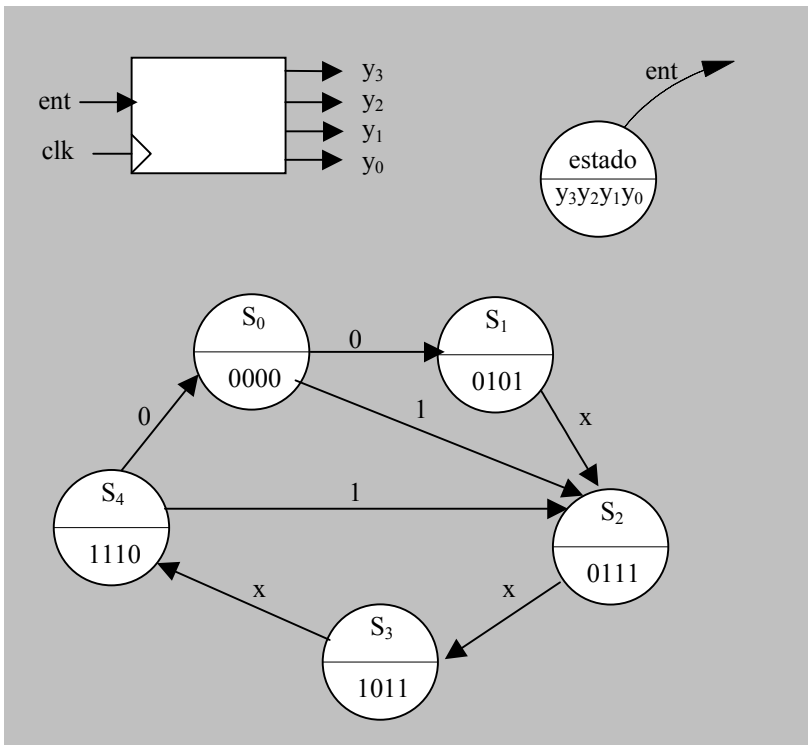
El circuito tiene un bit *ent* de entrada, tal como se describe en el enunciado. La secuencia de salida debe variar en función del valor de este bit.

Tendrá 4 bits de salida, suficientes para codificar en binario el 14, que es el número más grande de la secuencia.

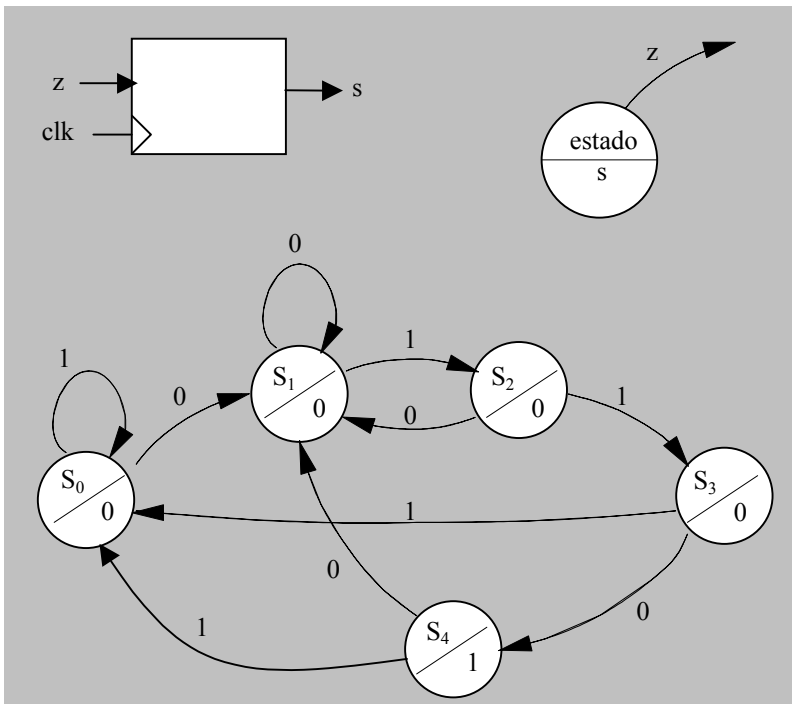
A continuación se presenta un dibujo del sistema desde el punto de vista de entradas/salidas y su grafo de estados.

En este grafo hay transiciones etiquetadas con *x*, señalando que la transición se produce independientemente del valor de la entrada *ent*.

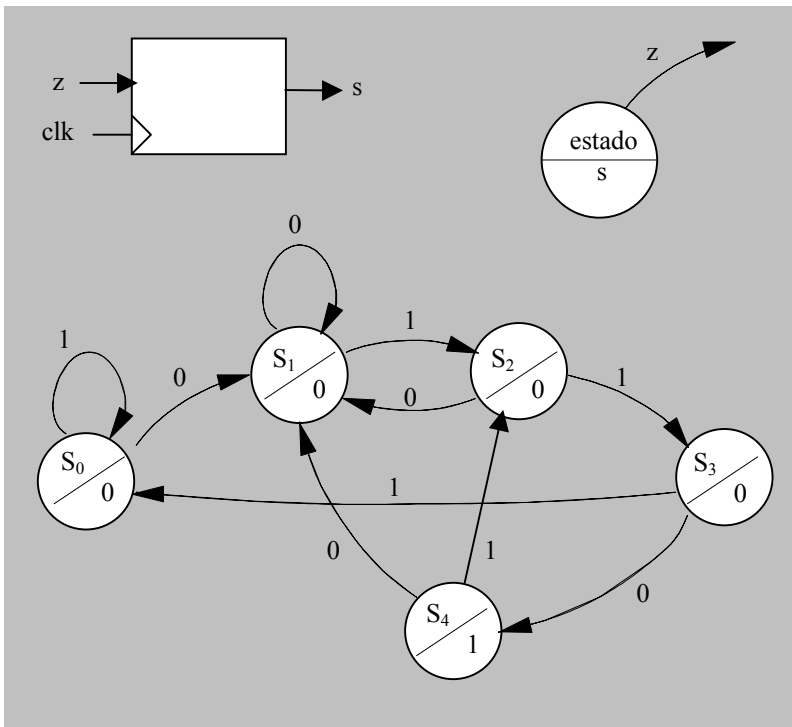
Por otra parte, desde los estados S_0 y S_1 se va hasta el estado S_2 en caso de que en la entrada *ent* aparezca un 1. Este caso puede producirse cuando la salida tiene un valor que corresponde a la secuencia que se genera cuando *ent*=0, y sin embargo no existe en la secuencia para *ent*=1. En realidad, podría haberse realizado una transición tanto al S_2 como al S_3 o al S_4 , ya que en el enunciado no se especifica que la secuencia deba comenzar por un número en particular.



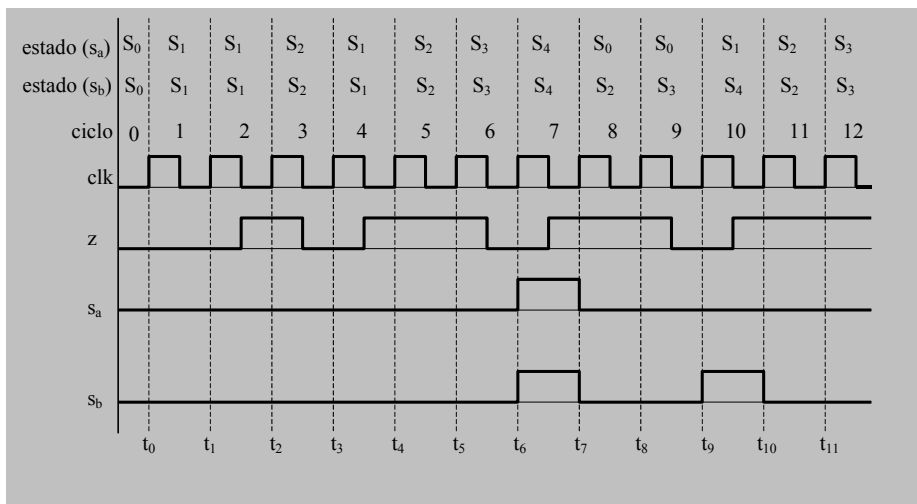
26.
a)



b)
La diferencia está en la transición que se produce desde *S₄* cuando en la entrada *z* hay un 1.



c)
Las salidas s_a y s_b representan la salida de los apartados a y b respectivamente.



En el cronograma se indican los valores t_i en los que el circuito "ve" de la señal z en cada ciclo. Recordemos que el circuito "ve" los valores de las señales en los instantes de los flancos ascendentes de reloj. En este cronograma se ha considerado que esos instantes son al final del ciclo.

Podemos observar que en el ciclo 6 se ha producido la secuencia 0110. En consecuencia, a final del ciclo (con el flanco del reloj) se pasará al estado S_4 . Este cambio de estado se produce en el instante t_6 . En este estado, la salida del sistema vale 1, como se puede observar en el cronograma (tanto s_a como s_b valen 1 durante el ciclo 7. Siguiendo el mismo razonamiento, obtenemos que el sistema considerado en el apartado b) estará en el estado s_4 durante el ciclo 10, y por eso s_b vale 1 durante ese ciclo.

Podemos escribir de esta forma la secuencia de valores que se producen en la entrada y la salida del sistema:

z: 001011011011
 sa: 000000010000
 sb: 000000010010

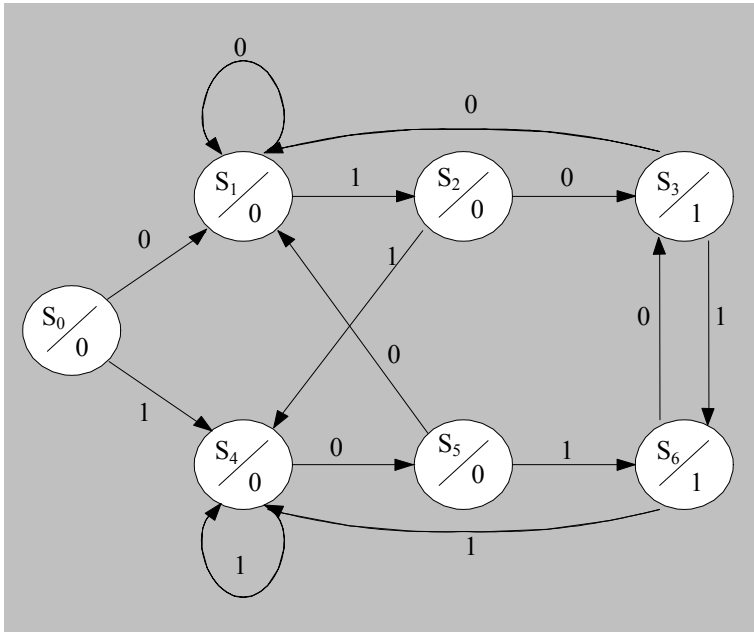
Fijaos que la salida se activa un ciclo "retrasada" con respecto de la entrada. Eso es debido a que, como se ha visto en el cronograma, los valores de z se leen al final del ciclo. Así, si

leemos $z=0$ suponemos que ese 0 ha estado presente durante todo el ciclo (solamente a efectos de escribir la secuencia en forma lineal), ya que no tenemos manera de saber qué valores ha tomado z a excepción de los leídos en los flancos del reloj.

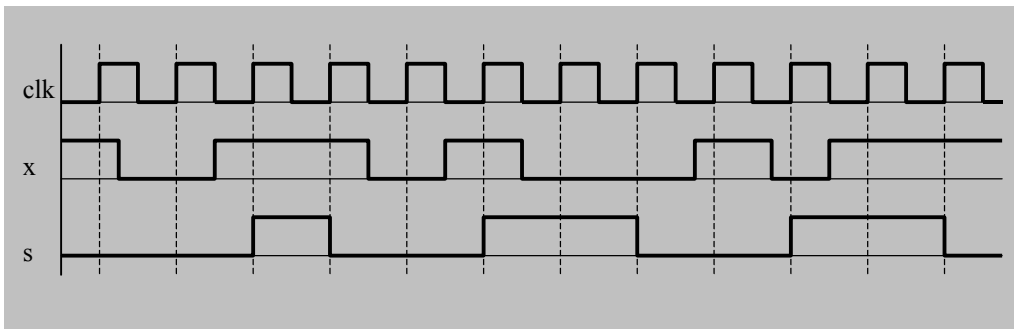
Como puede verse, esta forma de representar los valores de z contiene menos información que la presentada en un cronograma, pero esta información es suficiente para decidir cómo se comporta el sistema.

29.

a)



b)



30.

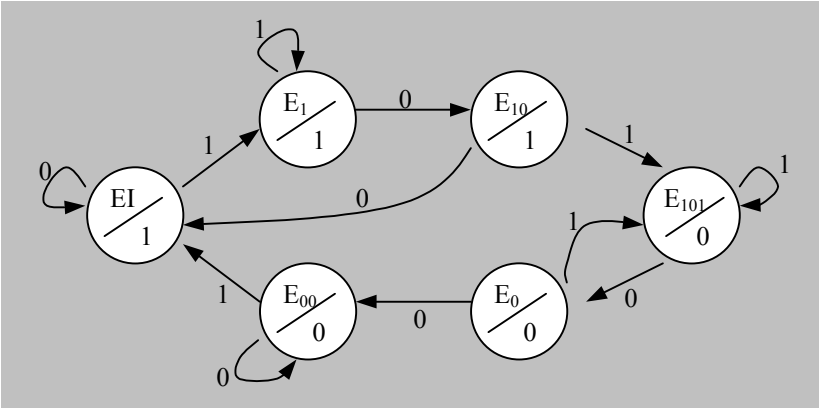
Com que ens diuen que la sortida pot valdre 0 o 1, deduïm que és d'un bit.

Partim d'un estat inicial EI en què la sortida val 1. El primer objectiu és saber reconèixer que ha arribat la seqüència 101 per l'entrada. Per tant,

- quan arribi un 1 per l'entrada, el sistema haurà de recordar-ho. Aquesta és la funció de l'estat E_1 (el missatge que recorda l'estat E_1 és "l'últim valor que ha arribat ha estat un 1").
- Si arriba un 0 immediatament després d'un 1, s'ha de recordar també. Aquesta és la funció de l'estat E_{10} ("els dos últims valors que han arribat han estat 10, en aquest ordre")
- Estant en l'estat E_{10} , si arriba un 0 haurem de tornar a esperar que arribi el proper 1 per reconèixer la seqüència. Per tant, tornem a l'estat inicial (en el qual "no es recorda res"). En canvi, si arriba un 1 sabem que ha arribat la seqüència 101 sencera. Creem l'estat E_{101} per recordar-ho, i posem la sortida a 0.

- Quan el sistema hagi arribat a l'estat E_{101} , l'objectiu serà reconèixer la seqüència 001 a l'entrada. Seguint un raonament anàleg al que s'acaba d'exposar, deduíem que cal crear els estats E_0 i E_{00} .

El graf d'estats complet és aquest:



31.

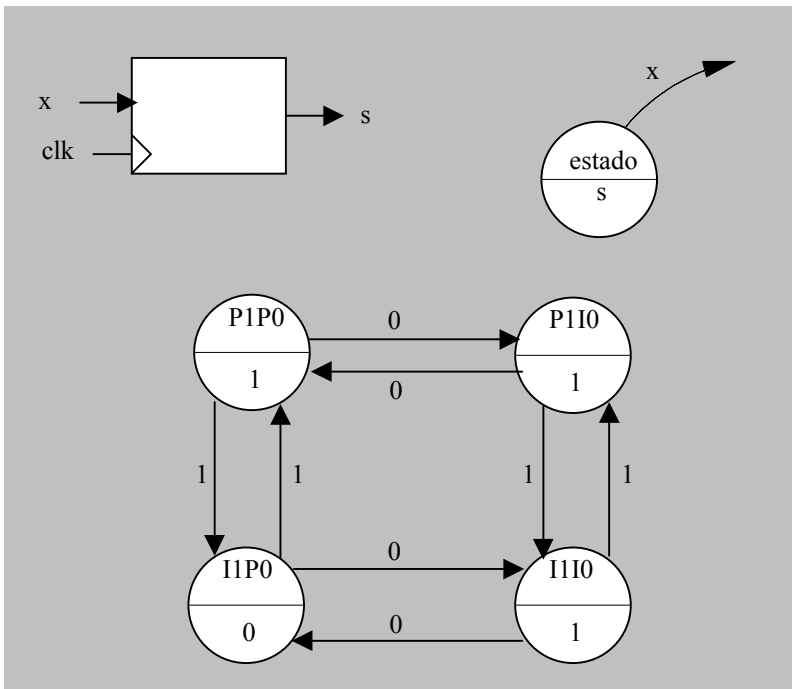
El sistema puede encontrarse en uno de los siguientes 4 estados:

- I1I0: Han entrado un número impar de unos y un número impar de ceros
- I1P0: Han entrado un número impar de unos y un número par de ceros
- P1I0: Han entrado un número par de unos y un número impar de ceros
- P1P0: Han entrado un número par de unos y un número par de ceros

La salida del sistema será un 0 para el estado I1P0 y un 1 en cualquiera de los otros tres estados.

Para comenzar a diseñar asumiremos que el 0 es un número par, y que por tanto el sistema comienza en el estado P1P0.

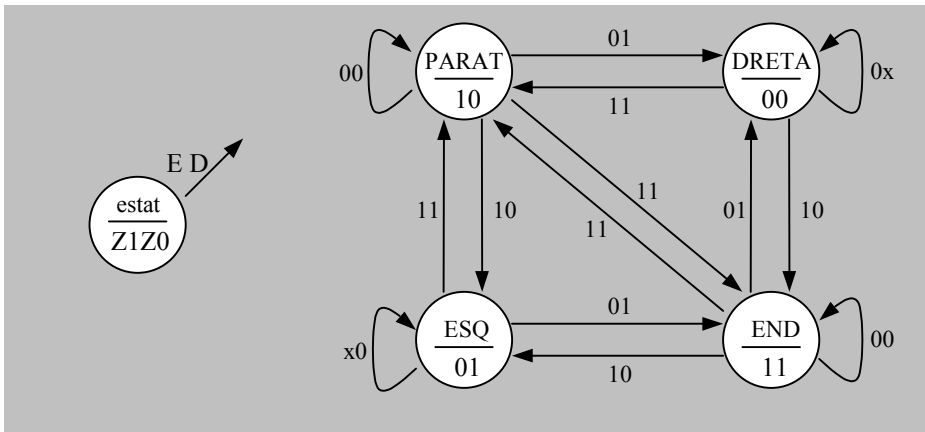
A continuación se muestra el grafo de estados del sistema.



33.

Al cotxe li poden passar 4 coses: estar parat, anar cap a l'esquerra, anar cap a la dreta i anar endavant. El sistema tindrà doncs 4 estats, que seran respectivament PARAT, ESQ, DRETA i END.

De l'enunciat s'obtenen directament totes les transicions:



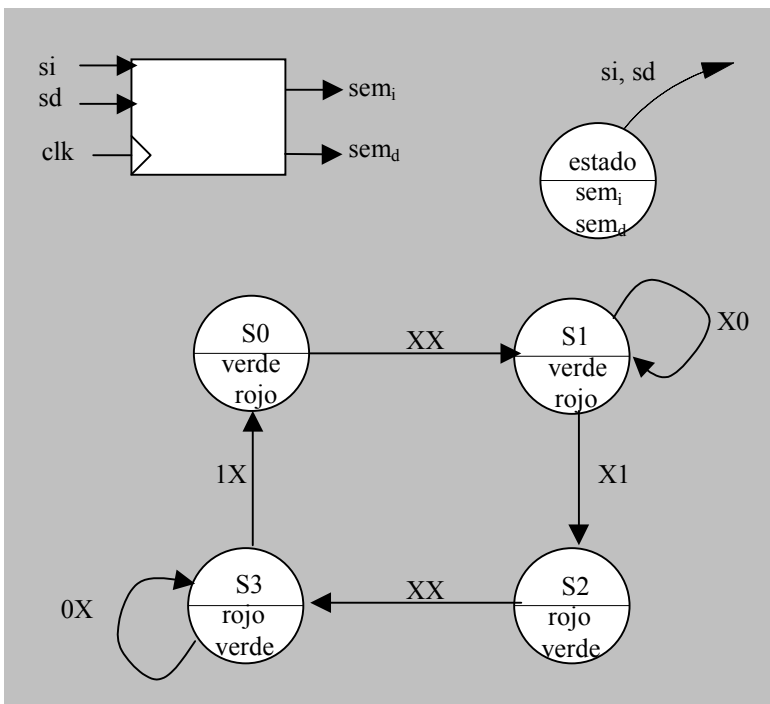
34.

En primer lugar, podemos ver que ambos semáforos no podrán estar nunca simultáneamente en verde o en rojo. Por tanto, si asignamos un 1 al color verde y un 0 al color rojo, siempre se cumplirá que $sem_d = sem_i'$. Bastaría por tanto que el sistema tuviera un único bit de salida, y con este bit podrían gestionarse los dos semáforos. No obstante, por claridad se han incluido los dos bits en el grafo.

Como un semáforo ha de mantenerse verde al menos dos ciclos consecutivos de reloj, una vez se produzca el cambio de color en un semáforo pasaremos SIEMPRE a otro estado en el que el sistema mantendrá la misma salida, para asegurar que durante estos dos ciclos de reloj se mantiene el semáforo en verde.

Los cambios de color en los semáforos se producen después de dos ciclos consecutivos de reloj con el mismo color y cuando haya coches en el otro lado del puente, INDEPENDIEN- TEMENTE de que haya coches en el lado del puente que tiene el semáforo en verde. Por eso, las transiciones en las que cambia el color de los semáforos dependen únicamente de que haya o no vehículos en el otro lado del puente.

A continuación se muestra el grafo de estados del sistema. Supondremos que inicialmente el sistema arranca con el semáforo izquierdo en verde durante dos ciclos consecutivos de reloj (S0)



39.

Descrivint el contingut de la ROM en funció dels valors dels senyals connectats a l'entrada d'adreces (X, Q_2, Q_1, Q_0), obtenim la taula de transicions del circuit. La taula de sortides s'obté directament de les línies Z_2, Z_1, Z_0 .

X	Q ₂	Q ₁	Q ₀	D ₂	D ₁	D ₀
0	0	0	0	0	0	1
0	0	0	1	0	1	0
0	0	1	0	0	1	1
0	0	1	1	1	0	0
0	1	0	0	1	0	1
0	1	0	1	0	0	0
0	1	1	0	0	0	0
0	1	1	1	0	0	0
1	0	0	0	1	0	1
1	0	0	1	0	0	0
1	0	1	0	0	0	1
1	0	1	1	0	1	0
1	1	0	0	0	1	1
1	1	0	1	1	0	0
1	1	1	0	0	0	0
1	1	1	1	0	0	0

taula de transicions

Q ₂	Q ₁	Q ₀	Z ₂	Z ₁	Z ₀
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	0
1	1	1	1	1	1

taula de sortides

Cada combinació dels biestables correspon a un estat diferent; donem un nom a cada estat segons la taula de noms següent. Sabent que $D = Q^+$, podem reescriure les taules de transicions i sortides fent servir el nom de cada estat:

Q ₂	Q ₁	Q ₀	nom
0	0	0	A
0	0	1	B
0	1	0	C
0	1	1	D
1	0	0	E
1	0	1	F
1	1	0	G
1	1	1	H

taula de noms

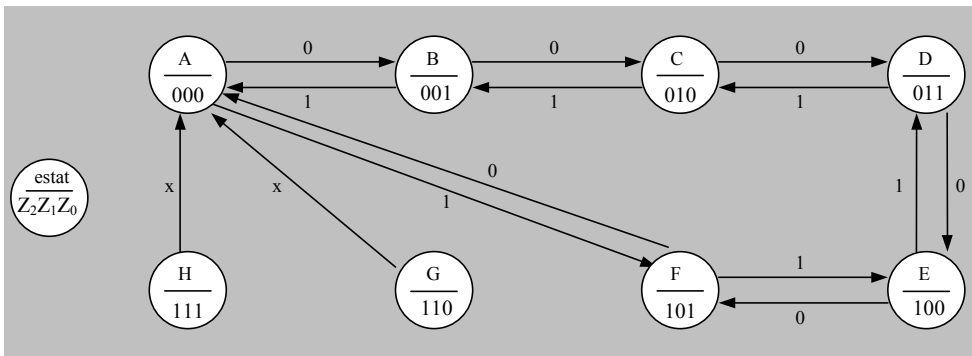
X	estat	estat futur
0	A	B
0	B	C
0	C	D
0	D	E
0	E	F
0	F	A
0	G	A
0	H	A
1	A	F
1	B	A
1	C	B
1	D	C
1	E	D
1	F	E
1	G	A
1	H	A

taula de transicions

estat	Z ₂	Z ₁	Z ₀
A	0	0	0
B	0	0	1
C	0	1	0
D	0	1	1
E	1	0	0
F	1	0	1
G	1	1	0
H	1	1	1

taula de sortides

Només falta expressar la informació de les taules en forma de graf d'estats:



Veiem que es tracta d'un comptador reversible mòdul 6. Als estats G i H no s'hi arribarà mai, a no ser que sigui en inicialitzar el sistema. En aquest cas, es passarà en el cicle següent a l'estat A, començant el funcionament normal del comptador a partir d'aquest.

40.

a)

$$D_0 = x' Q_0' + x Q_0$$

$$D_1 = x' Q_1' Q_0 + x' Q_1 Q_0' + x \cdot Q_1' \cdot 1 + x \cdot Q_1 \cdot 0$$

$$= x' Q_1' Q_0 + x' Q_1 Q_0' + x Q_1'$$

b)

Taula de transicions:

Q_1	Q_0	x	D_1	D_0	Q_1^+	Q_0^+
0	0	0	0	1	0	1
0	0	1	1	0	1	0
0	1	0	1	0	1	0
0	1	1	1	1	1	1
1	0	0	1	1	1	1
1	0	1	0	0	0	0
1	1	0	0	0	0	0
1	1	1	0	1	0	1

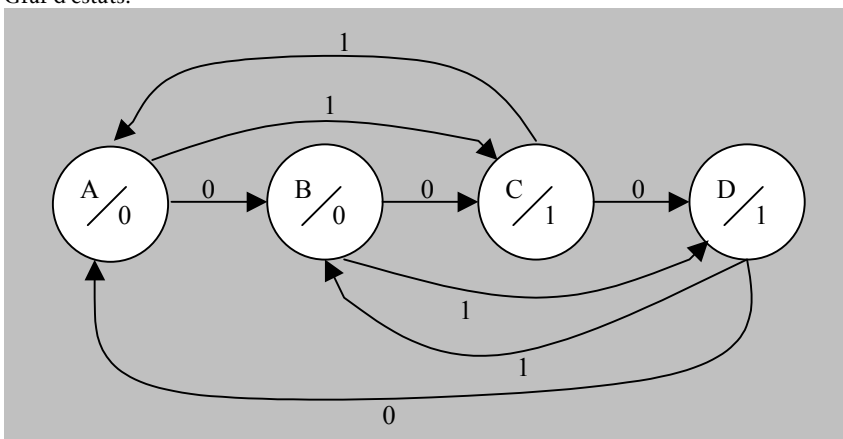
Taula de sortides:

Q_1	Q_0	Z
0	0	0
0	1	0
1	0	1
1	1	1

c) Donem nom als estats:

Q_1	Q_0	nom
0	0	A
0	1	B
1	0	C
1	1	D

Graf d'estats:



41.

a) Para dibujar el cronograma correspondiente a los puntos a, b, c, d, f y g se ha de tener en cuenta la ecuación de transición del biestable D:

$$Q^+ = D$$

Esta ecuación indica que el estado futuro pasa a ser el valor que había en la entrada del biestable en el momento en que se produce el flanco ascendente del reloj.

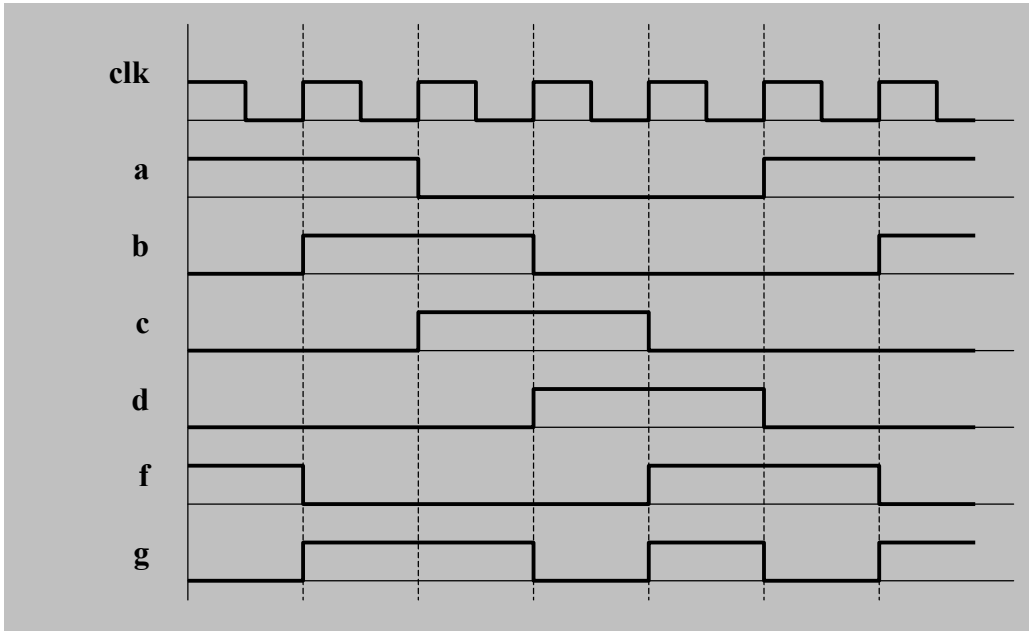
$$b^+ = c'd'$$

$$c^+ = b$$

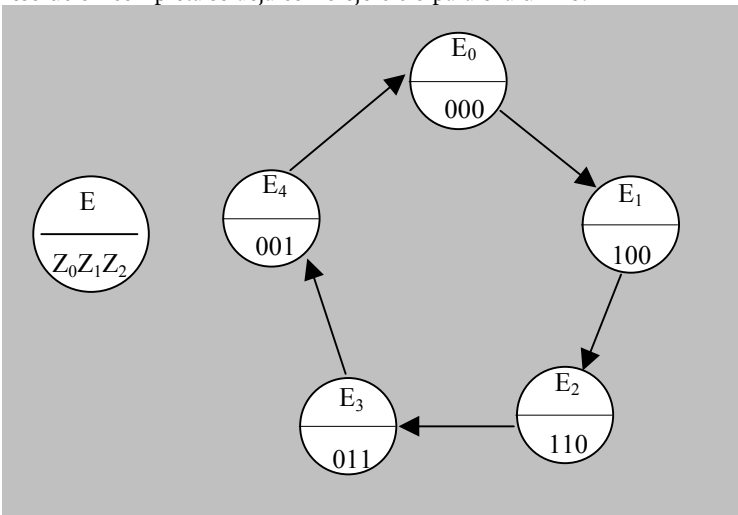
$$d^+ = c$$

$$f^+ = d$$

La funció g es combinacional: $g = (d \oplus f)'$



b) El sistema segueix una seqüència de cinc estats i no té cap entrada. A continuació es descriu el gra d'estats. Els altres tres estats no han estat considerats en la resolució del problema. No obstant, l'anàlisi complet del sistema requeriria especificar com es comporta el sistema en cada un d'aquests estats. La seva resolució completa se deixa com a exercici per a l'alumne.



42.

Codificació de estats

Estado	Q_2	Q_1	Q_0
S_0	0	0	0
S_1	0	0	1
S_2	0	1	0
S_3	0	1	1
S_4	1	0	0

Tabla de transiciones de estados

Entrada	Estado	Estado'
0	S ₀	S ₂
0	S ₁	S ₁
0	S ₂	S ₄
0	S ₃	S ₃
0	S ₄	S ₀
1	S ₀	S ₁
1	S ₁	S ₄
1	S ₂	S ₃
1	S ₃	S ₄
1	S ₄	S ₄

En la tabla anterior se han incluido únicamente los estados del grafo. A continuación escribiremos la misma tabla usando la codificación de estados utilizada. Asumiremos que los estados no definidos no pueden producirse nunca, por lo que de cara a la simplificación del circuito permitiremos que se produzca cualquier transición (ponemos X en la tabla).

Entrada	Q ₂	Q ₁	Q ₀	Q ₂ ⁺	Q ₁ ⁺	Q ₀ ⁺
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	1	0	0
0	0	1	1	0	1	1
0	1	0	0	0	0	0
0	1	0	1	X	X	X
0	1	1	0	X	X	X
0	1	1	1	X	X	X
1	0	0	0	0	0	1
1	0	0	1	1	0	0
1	0	1	0	0	1	1
1	0	1	1	1	0	0
1	1	0	0	1	0	0
1	1	0	1	X	X	X
1	1	1	0	X	X	X
1	1	1	1	X	X	X

Tabla de salidas

Estado	Salida
S ₀	0
S ₁	1
S ₂	0
S ₃	1
S ₄	1

c)

D₂

	D₂			
	00	01	11	10
XQ₂				
Q₁Q₀				
00	0	0	1	0
01	0	X	X	1
11	0	X	X	1
10	1	X	X	0

$D_2 = x Q_2 + x Q_0 + x' Q_1 Q_0'$

D₁

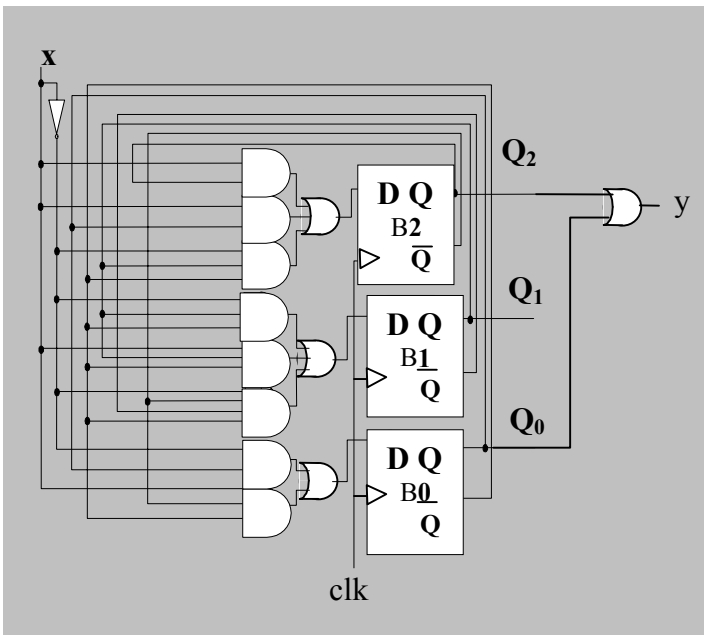
	D₁			
	00	01	11	10
XQ₂				
Q₁Q₀				
00	1	0	0	0
01	0	X	X	0
11	1	X	X	0
10	0	X	X	1

$D_1 = x' Q_1 Q_0' + x Q_1 Q_0 + x' Q_2' Q_1' Q_0'$

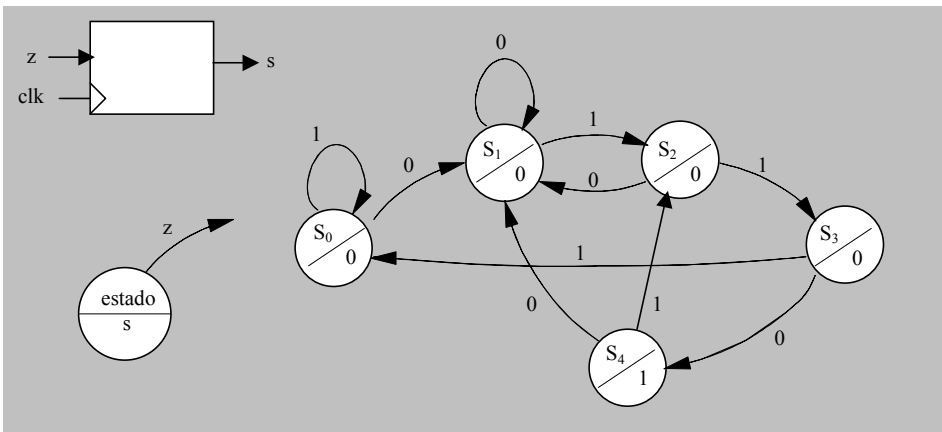
		D_0			
		00	01	11	10
$Q_1 Q_0$	00	0	0	0	1
	01	1	X	X	0
	11	1	X	X	0
	10	0	X	X	1

$D_0 = x' Q_0 + x Q_2' Q_0'$

En cuanto a la salida y, tenemos $y = Q_2 + Q_0$ (la simplificación por Karnaugh se deja como ejercicio para el alumno). El circuito es el siguiente:



43.
a)



Estado	Q_2	Q_1	Q_0
S_0	0	0	0
S_1	0	0	1
S_2	0	1	0
S_3	0	1	1
S_4	1	0	0

z	Q_2	Q_1	Q_0	Q_2^+	Q_1^+	Q_0^+
0	0	0	0	0	0	1
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	1	0	0
0	1	0	0	0	0	1
0	1	X	X	X	X	X
1	0	0	0	0	0	0
1	0	0	1	0	1	0
1	0	1	0	0	1	1
1	0	1	1	0	0	0
1	1	0	0	0	1	0
1	1	X	X	X	X	X

D_2

zQ_2	Q_1Q_0			
	00	01	11	10
00	0	0	0	0
01	0	X	X	0
11	1	X	X	0
10	0	X	X	0

$$D_2 = z Q_1 Q_0$$

D_1

zQ_2	Q_1Q_0			
	00	01	11	10
00	0	0	1	0
01	0	X	X	1
11	0	X	X	0
10	0	X	X	1

$$D_1 = z Q_2 + z Q_1' Q_0 + z Q_1 Q_0'$$

D_0

xQ_2	Q_1Q_0			
	00	01	11	10
00	1	1	0	0
01	1	X	X	0
11	0	X	X	0
10	1	X	X	1

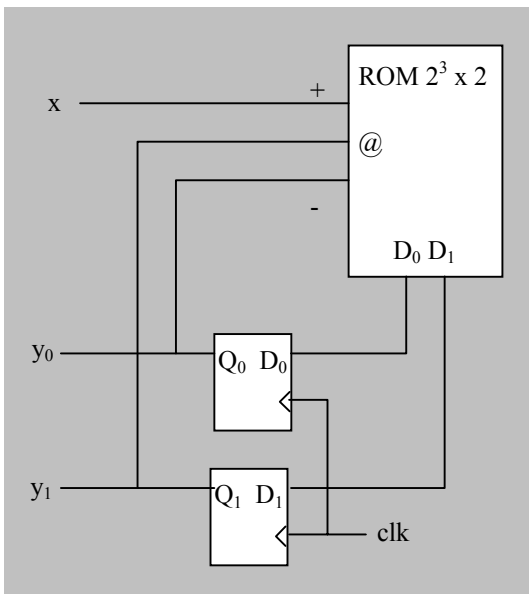
$$D_0 = Q_1 Q_0' + Q_1' z'$$

x	Q ₁ Q ₀	Q ₁ ⁺ Q ₀ ⁺	D ₁	D ₀
0	0 0	0 1	0	1
0	0 1	1 1	1	1
0	1 0	0 1	0	1
0	1 1	1 1	1	1
1	0 0	0 0	0	0
1	0 1	1 0	1	0
1	1 0	0 0	0	0
1	1 1	1 0	1	0

El tamaño de la ROM necesario para implementar el circuito es de $2^3 \times 2$ bits. Su contenido será :

@	Contenido
XQ ₁ Q ₀	D ₁ D ₀
0 0 0	0 1
0 0 1	1 1
0 1 0	0 1
0 1 1	1 1
1 0 0	0 0
1 0 1	1 0
1 1 0	0 0
1 1 1	1 0

A continuación se muestra el diseño del sistema. Por simplicidad en el diseño, para evitar cruces de cables, se han intercambiado de orden las salidas D₀ y D₁ en la ROM con respecto a las que se presentan en la tabla de verdad.

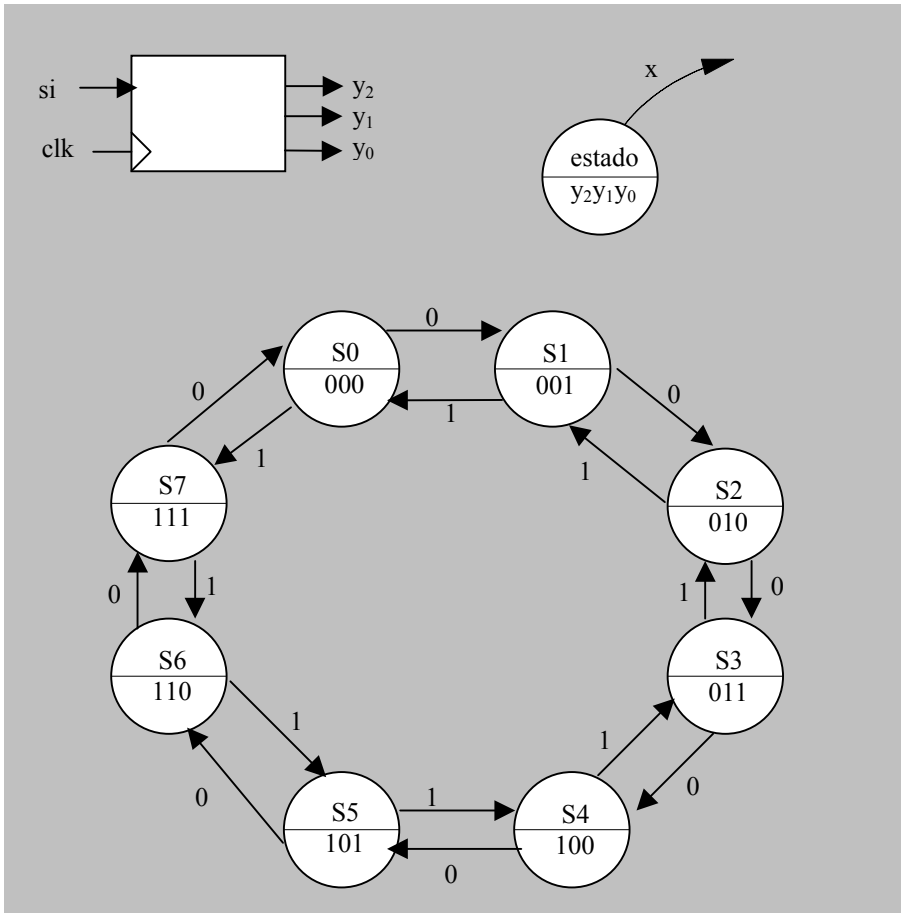


48.

En primer lugar diseñaremos el grafo de estados del contador.

El sistema tendrá 8 estados que corresponderán a cada una de las posibles salidas de tres bits. Los estados se numeran desde S₀ a S₇. La codificación del estado se hará coincidir con la codificación de la salida por simplicidad en el diseño (podemos usar una memoria ROM de la mitad de tamaño).

A continuación se presenta el grafo de estados.



Como tenemos 8 estados necesitamos 3 biestables. La tabla de codificación de estados del sistema será la siguiente:

Estado	Q_2	Q_1	Q_0
S0	0	0	0
S1	0	0	1
S2	0	1	0
S3	0	1	1
S4	1	0	0
S5	1	0	1
S6	1	1	0
S7	1	1	1

La tabla de transiciones del sistema es la siguiente:

x	$Q_2Q_1Q_0$	$Q_2^+Q_1^+Q_0^+$	D_2	D_1	D_0
0	0 0 0	0 0 1	0	0	1
0	0 0 1	0 1 0	0	1	0
0	0 1 0	0 1 1	0	1	1
0	0 1 1	1 0 0	1	0	0
0	1 0 0	1 0 1	1	0	1
0	1 0 1	1 1 0	1	1	0
0	1 1 0	1 1 1	1	1	1
0	1 1 1	0 0 0	0	0	0
1	0 0 0	1 1 1	1	1	1
1	0 0 1	0 0 0	0	0	0
1	0 1 0	0 0 1	0	0	1
1	0 1 1	0 1 0	0	1	0
1	1 0 0	0 1 1	0	1	1
1	1 0 1	1 0 0	1	0	0
1	1 1 0	1 0 1	1	0	1
1	1 1 1	1 1 0	1	1	0

El tamaño de la ROM necesario para implementar el circuito es de $2^4 \times 3$. Su contenido será:

Direcció				Contenid		
x	Q ₂	Q ₁	Q ₀	D ₂	D ₁	D ₀
0	0	0	0	0	0	1
0	0	0	1	0	1	0
0	0	1	0	0	1	1
0	0	1	1	1	0	0
0	1	0	0	1	0	1
0	1	0	1	1	1	0
0	1	1	0	1	1	1
0	1	1	1	0	0	0
1	0	0	0	1	1	1
1	0	0	1	0	0	0
1	0	1	0	0	0	1
1	0	1	1	0	1	0
1	1	0	0	0	1	1
1	1	0	1	1	0	0
1	1	1	0	1	0	1
1	1	1	1	1	1	0

A continuació se mostra el disseny del sistema.

