Els circuits lògics seqüencials

Montse Peiron Guàrdia
Fermín Sánchez Carracedo

PID_00163595

UOC
Universitat Oberta de Catalunya
www.uoc.edu

Índex

Introducció 5

Objectius 6

1. Caracterització dels circuits lògics seqüencials 7
1.1. Necessitat de memòria en els circuits lògics 7
1.2. Rellotge. Sincronització 7

2. El biestable D 10
2.1. Dispositiu elemental de memòria. El biestable D 10
2.2. Senyal de càrrega 13
2.3. Entrades asíncrones 14

3. Blocs seqüencials 18
3.1. Registre 18
3.2. Banc de registres 23
3.3. Memòria RAM 25

4. El model de Moore 29
4.1. Estat. Transicions 29
4.2. Representació gràfica: grafs d'estats 34
4.2.1. Mecànica de disseny 35
4.2.2. Notació 37
4.2.3. Circuits sense entrades 38
4.3. Sincronització 40
4.4. Implementació 45

Resum 50

Exercicis d'autoavaluació 51

Solucionari 56

Bibliografia 75

Introducció

En el mòdul "Els circuits lògics combinacionals" s'ha vist que els circuits computen funcions lògiques dels senyals d'entrada: el valor dels senyals de sortida en un instant determinat depèn del valor dels senyals d'entrada en aquest mateix moment. Quan els senyals d'entrada varien, llavors com a conseqüència també variaran els de sortida (després del retard introduït per les portes i blocs, que en aquest curs no tenim en compte).

Ara bé, en algunes aplicacions cal que el valor dels senyals de sortida no depengui només del valor de les entrades en el mateix moment, sinó que també tingui en compte els valors que han pres les entrades anteriorment. En els circuits que hem conegut fins ara, això no és possible: fan falta elements que tinguin alguna "capacitat de recordar", que són els que conformen els circuits lògics seqüencials.

En aquest mòdul coneixerem el concepte de *sincronització* i s'estudiaran els *biestables*, que són els dispositius seqüencials més bàsics, i els blocs seqüencials, que es construeixen a partir de biestables i tenen una funcionalitat determinada.

Després es presentarà una de les maneres de formalitzar el funcionament d'un circuit seqüencial, l'anomenat *model de Moore*. El model de Moore fa servir els conceptes d'*estat* i de *transicions entre estats* per a descriure l'evolució temporal del funcionament d'un circuit seqüencial, que es representa gràficament mitjançant un *graf d'estats*.

Aquest model permet descriure el comportament de molts circuits seqüencials.

Objectius

L'objectiu fonamental d'aquest mòdul és conèixer els circuits lògics seqüencials, és a dir, saber com estan formats i poder-los utilitzar amb agilitat. Per a arribar a aquest punt caldrà haver satisfet els objectius següents:

1. Saber discernir, a partir de la funcionalitat que es vol que tingui un circuit lògic, si el circuit ha de ser de tipus seqüencial o combinacional.

2. Entendre el concepte de memòria, la necessitat d'una sincronització en els circuits lògics seqüencials i el funcionament del senyal de rellotge.

3. Conèixer el funcionament del biestable D i de totes les entrades de control que pot tenir.

4. Conèixer la funcionalitat dels diferents blocs seqüencials, i saber-los utilitzar en el disseny de circuits.

5. Comprendre els conceptes d'estat i de transició entre estats. Entendre tots els elements d'un graf d'estats, i ser capaços de construir el graf d'estats d'un circuit qualsevol a partir de la descripció de la seva funcionalitat.

6. Saber deduir l'evolució temporal d'un circuit a partir del graf d'estats que en descriu el funcionament.

1. Caracterització dels circuits lògics seqüencials

1.1. Necessitat de memòria en els circuits lògics

Sigui un circuit amb un senyal d'entrada X i un de sortida Z, tots dos de n bits, que interpretem com a nombres representats en complement a 2. Suposem que volem que $Z = X + 2$. Amb els elements estudiats en el mòdul "Els circuits lògics combinacionals" sabem com s'ha de fer, fins i tot de moltes maneres diferents. Quan el valor present a l'entrada X varií, llavors Z també canviarà conseqüentment de valor.

Suposem ara que volem que el valor de Z correspongui a la suma de tots els valors que han estat presents a l'entrada X durant un interval de temps determinat (durant el qual el valor de X ha variat). Amb els dispositius lògics que coneixem fins ara no ho podem aconseguir, perquè quan canviem el valor de X, el valor anterior ha "desaparegut" i ja no el podem fer servir per a calcular la suma.

Cal que aquest circuit sigui capaç de recordar o retenir els valors anteriors d'alguns senyals, és a dir, ha de tenir memòria. Aquesta és la funcionalitat que distingeix els circuits lògics seqüencials dels combinacionals.

Seqüencial

La denominació "seqüencial" deriva justament de la capacitat de recordar la seqüència de valors que prenen els senyals.

1.2. Rellotge. Sincronització

En els circuits combinacionals, l'única noció temporal que intervé és el present. En canvi, en els circuits seqüencials es té en compte l'evolució temporal dels senyals (i apareix, com es veurà més endavant, la noció de futur).

Ara bé, en la descripció del circuit de l'exemple anterior, què vol dir exactament que "tots els valors que han estat presents en l'entrada X durant un temps determinat"? El senyal X pot anar canviant de valor de manera aleatòria en el temps: pot valer 13 durant quatre nanosegons, després –25 durant deu nanosegons, després 0 durant un nanosegon, etc. Com pot determinar el circuit en quin moment X canvia de valor, és a dir, en quin moment ha de considerar que "X ha deixat de tenir el valor antic" i "comença a tenir el valor nou"? Per a poder-ho determinar, el circuit ha de tenir un **mecanisme de sincronització**. En els circuits seqüencials que estudiarem en aquest mòdul s'utilitza un senyal de rellotge com a forma de sincronització.

El **rellotge** és un senyal que serveix per a determinar els instants en què un circuit seqüencial "veu" el valor dels senyals, o "hi és sensible", i respon en conseqüència.

Discretització

El rellotge discretitza el temps: en lloc de veure'l com una dimensió contínua, els circuits el veuen com una seqüència d'instants.

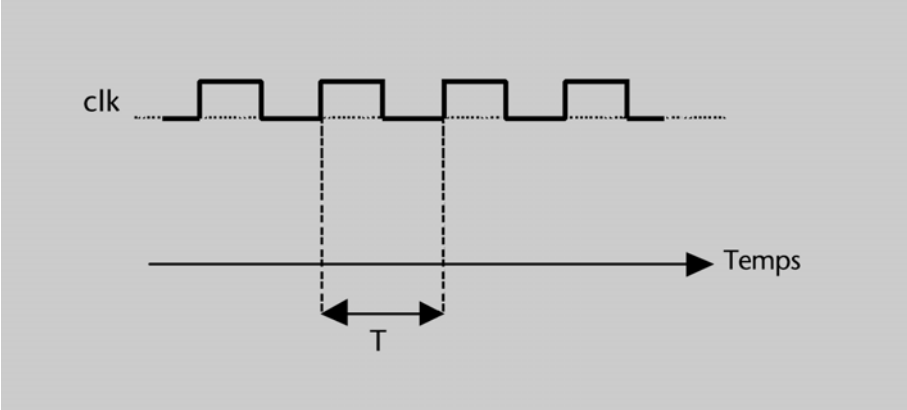
Aquesta tasca que duu a terme el senyal de rellotge s'anomena **sincronització dels circuits**.

Concretament, el senyal de rellotge pren els valors 0 i 1 de manera cíclica i contínua des de la posada en marxa d'un circuit i fins que aquest s'atura. Usualment, es fa servir la notació *clk* per a fer referència al senyal de rellotge (deriva de l'anglès *clock*).

La figura 1 mostra el cronograma del senyal de rellotge. El cicle que forma la seqüència de valors 0 i 1 té una durada determinada i constant, *T*, que s'anomena **període**. Es pot mesurar en segons o, més habitualment, en nanosegons (mil milionèsimes de segon).

Tal com s'ha vist en el mòdul "Els circuits lògics combinacionals", les transicions entre els valors 0 i 1 d'un senyal tenen un cert retard. Nosaltres, però, considerarem que els canvis de valor, del senyal de rellotge o de qualsevol altre, són instantanis.

Figura 1



Generació d'un senyal de rellotge

Físicament, el senyal de rellotge es genera a partir de cristalls de quars, un mineral que té la propietat anomenada *piezoelectricitat*: quan rep corrent elèctric vibra amb una freqüència extremadament gran i regular.

Els instants en què el senyal de rellotge passa de 0 a 1 s'anomenen **flancs ascendents**. L'interval de temps que hi ha entre un flanc ascendent i el següent s'anomena **cicle** o **cicle de rellotge**. Per tant, la duració d'un cicle és un període, *T* segons.

La **freqüència** del rellotge és la inversa del període, és a dir, és el nombre de cicles de rellotge que tenen lloc durant un segon. Es mesura en hertzs (cicles per segon); el més habitual és usar el múltiple gigahertzs (mil milions de cicles per segon), que s'abreuja GHz. Per exemple, si tenim un rellotge amb un període de 0,75 nanosegons, la seva freqüència és la següent:

$$\frac{1 \text{ cicle}}{0,75 \cdot 10^{-9} \text{ segons}} = 1,33 \cdot 10^{9} \text{ cicles/segon} = 210^{-9} \text{ segons} = 1,33 \text{ GHz.}$$

Valors típics de les freqüències

Al començament de l'any 2010, els processadors comercials més ràpids tenen freqüències de rellotge pròximes als 3,5 GHz, amb períodes propers a 0,3 nanosegons. Tanmateix, la freqüència base es pot multiplicar utilitzant la tècnica anomenada *overclocking*.

El senyal de rellotge pot sincronitzar els circuits de diverses formes. En aquest curs només es veurà la que s'usa més habitualment, anomenada **sincronització per flanc ascendent**. Aquesta forma de sincronització estableix que els dispositius seqüencials d'un circuit seran sensibles als valors dels senyals en els instants dels flancs ascendents, tal com veurem en l'apartat següent. En la resta dels apunts, si escrivim només *flanc* ens referim a *flanc ascendent*.

Altres formes de sincronització

Altres formes de sincronització són per nivell 0, per nivell 1 i per flanc descendent.

Activitats

1. Es vol dissenyar un sistema que reconegui si es produeix la combinació 1010 en una entrada de quatre bits. Indiqueu si el sistema és seqüencial o combinacional, i per què.

2. Es vol dissenyar un sistema que reconegui una seqüència de quatre dígits decimals per a identificar el número secret d'una targeta de crèdit. El sistema té una entrada de dades única de quatre bits, que codifiquen cada dígit. Indiqueu si el sistema és seqüencial o combinacional, i per què.

3. Quin és el període del rellotge d'un processador Intel Celeron E1200 que funciona a 1,6 GHz?

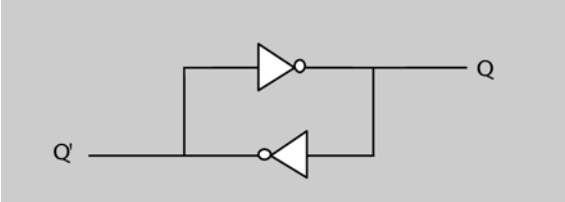
2. El biestable D

2.1. Dispositiu elemental de memòria. El biestable D

En l'apartat anterior hem vist la necessitat que els circuits lògics tinguin capacitat de memòria. En aquest apartat veurem com es construeixen els dispositius que poden "recordar" els valors dels senyals.

Si examinem el circuit que es mostra en la figura 2, veiem que el valor que hi ha en els punts *Q* i *Q'* (0 o 1) es mantindrà indefinidament, ja que la sortida de cada inversor està connectada amb l'entrada de l'altre. Per tant, podem dir que aquest circuit és capaç de "recordar", o mantenir en el temps, un valor lògic.

Figura 2



Ara bé, aquest circuit no és gaire útil, perquè no admet la possibilitat de modificar el valor recordat. Interessa de dissenyar un circuit que tingui aquesta mateixa capacitat de memòria, però que a més permeti que el valor en el punt *Q* pugui canviar segons els requeriments de l'usuari. Un circuit amb aquestes característiques s'anomena *biestable*.

Els **biestables** són els dispositius de memòria més elementals: permeten de guardar un bit d'informació.

Un biestable té dues sortides, *Q* i *Q'*. Es diu que *Q* és "el valor que guarda el biestable" en cada moment, o "el valor emmagatzemat al biestable", i *Q'* és la seva negació.

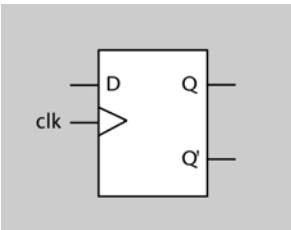
La denominació *biestable*

Prové del fet que el biestable pot estar "en dos estats": *Q* = 0 o *Q* = 1.

Hi ha diferents tipus de biestables. En aquesta assignatura només en veurem un, el **biestable D**. La figura 3 mostra la seva representació gràfica. Podem observar que té una entrada de rellotge, ja que es tracta d'un dispositiu seqüencial.

L'entrada de rellotge d'un dispositiu seqüencial sincronitzat per flanc ascendent s'identifica amb el símbol >.

Figura 3



Implementació interna

La implementació interna d'un biestable D es basa en el circuit de la figura 2, però no és objectiu d'aquesta assignatura conèixer-la.

El biestable D funciona de la manera següent:

> La sortida *Q* pren el valor que hi hagi en l'entrada *D* en cada flanc ascendent de rellotge. Durant la resta del cicle, el valor de *Q* no canvia.

És a dir, el biestable només és sensible al valor present en l'entrada *D* en els instants dels flancs ascendents.

La figura 4 mostra la taula de veritat que descriu el comportament del biestable *D* (no hi posem la columna corresponent a *Q'*, perquè és la negació de *Q*). En aquesta figura s'introdueixen algunes notacions que s'usaran d'ara en endavant:

- El símbol ↑ representa un flanc ascendent de rellotge.
- El símbol $^+$ a la dreta del nom d'un senyal es refereix al valor que prendrà aquest senyal quan es produeixi el proper flanc ascendent de rellotge.

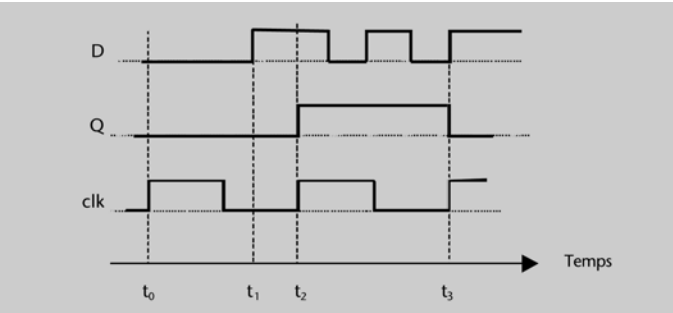
Per tant, Q^+ no identifica cap senyal del circuit, sinó el valor que tindrà el senyal *Q* en un instant futur: a partir del moment en què es produeixi el proper flanc. Aquesta notació, doncs, ens permet de descriure amb precisió l'evolució temporal dels senyals en un circuit lògic seqüencial.

Figura 4

D	clk	Q^+
0	↑	0
1	↑	1

La figura 5 mostra el cronograma del comportament d'un biestable D durant dos cicles: el que va de l'instant t_0 a l'instant t_2 i el que va de t_2 a t_3.

Figura 5



Nota

Notem que no tindria sentit el fet de dibuixar una línia amb el nom Q^+ en un cronograma, ja que Q^+ no correspon a cap senyal.

A la figura 5 podem observar que D pren diferents valors durant un cicle. En el cicle que va de t_0 a t_2 val primer 0 i després 1, i en el cicle que va de t_2 a t_3 pren els valors 1, 0, 1 i 0, i passa de nou a 1 en l'instant t_3 (coincidint amb el tercer flanc ascendent del rellotge). Ens podríem preguntar quin d'aquests valors és el que es carrega al biestable quan arriba el flanc ascendent, en els instants t_2 i t_3.

Valor que es carrega al biestable

En els circuits reals, el valor que es carrega al biestable és el que està present de manera estable a l'entrada D un cert interval de temps anterior al flanc, que depèn del retard en la pujada i baixada de tensió dels senyals i de la tecnologia emprada per a construir el biestable.

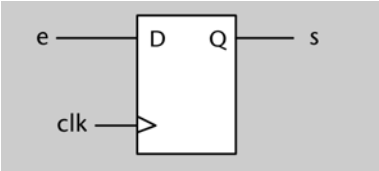
Per a respondre a aquesta pregunta, usarem la convenció següent: el valor que es carrega en un biestable en l'instant d'un flanc és el que té l'entrada D en el moment que arriba el flanc.

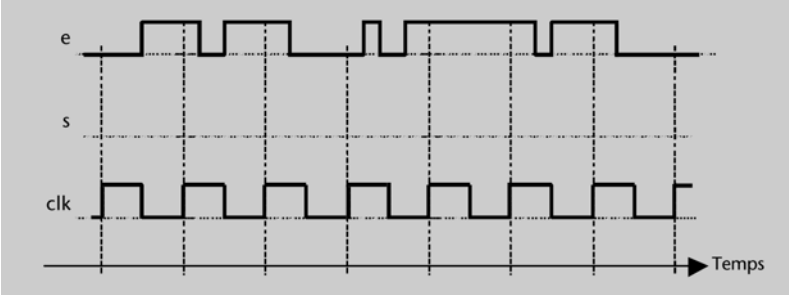
En un cronograma, aquesta idea es tradueix en el següent: el valor que es carrega al biestable en un flanc determinat és el que té la línia corresponent a l'entrada D quan toca la línia vertical corresponent a aquest flanc per l'esquerra (ja que, en un cronograma, el temps transcorre d'esquerra a dreta).

A la figura 5 podem veure que, en l'instant t_2, Q pren el valor 1 (en aquest cas no hi ha dubte), i en l'instant t_3, Q pren el valor 0. La convenció anterior ens diu que el valor que pren Q és 0, perquè és el que hi ha a la línia corresponent a D quan aquesta toca el flanc per l'esquerra.

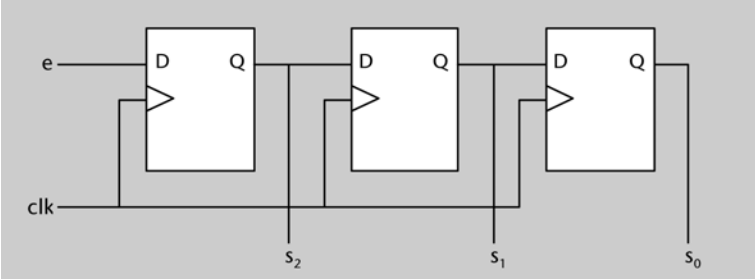
Activitats

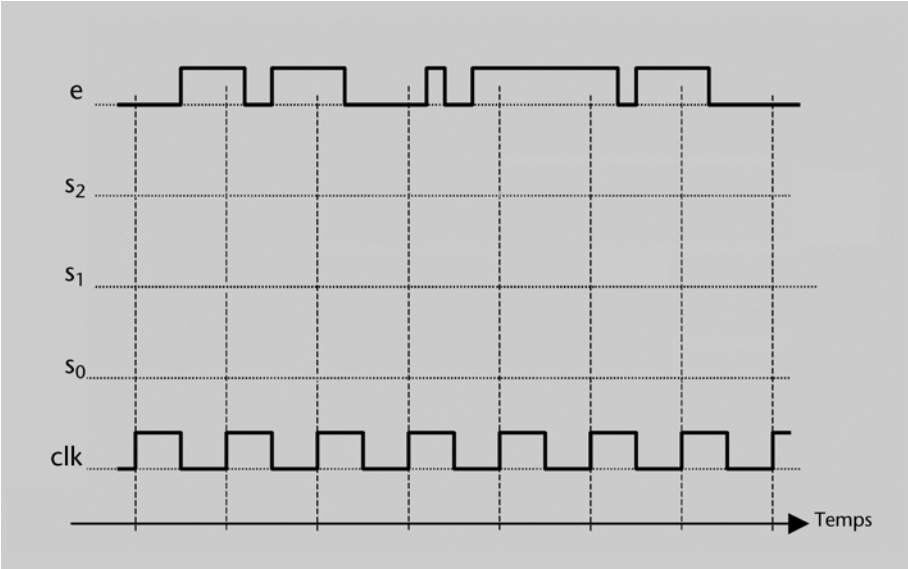
4. Completeu el cronograma que correspon al circuit de la figura.





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





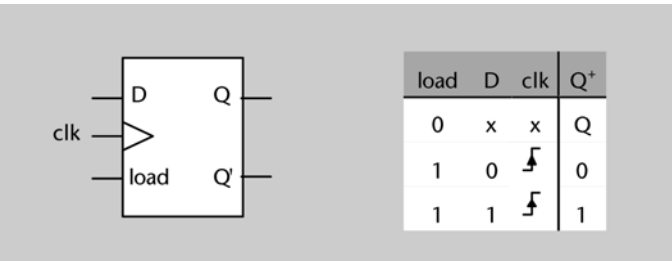
2.2. Senyal de càrrega

Quan es posa un biestable dins un circuit amb altres components, a vegades ens interessarà que el contingut del biestable no canviï, ni tan sols en els instants dels flancs, encara que varii el valor de *D*: voldrem que el biestable sigui "insensible" a les variacions de *D* quan així ho requerim.

Amb aquest fi s'afegeix al biestable un **senyal de càrrega**, que funciona de la manera següent: si val 0, el valor del biestable no canvia. Si val 1, el biestable funciona tal com s'ha explicat en l'apartat anterior.

La figura 6 mostra la representació gràfica d'un biestable *D* amb senyal de càrrega (s'identifica per la paraula *load*), i la taula de veritat del seu funcionament. Com és habitual, les *x* indiquen valors qualssevol dels senyals.

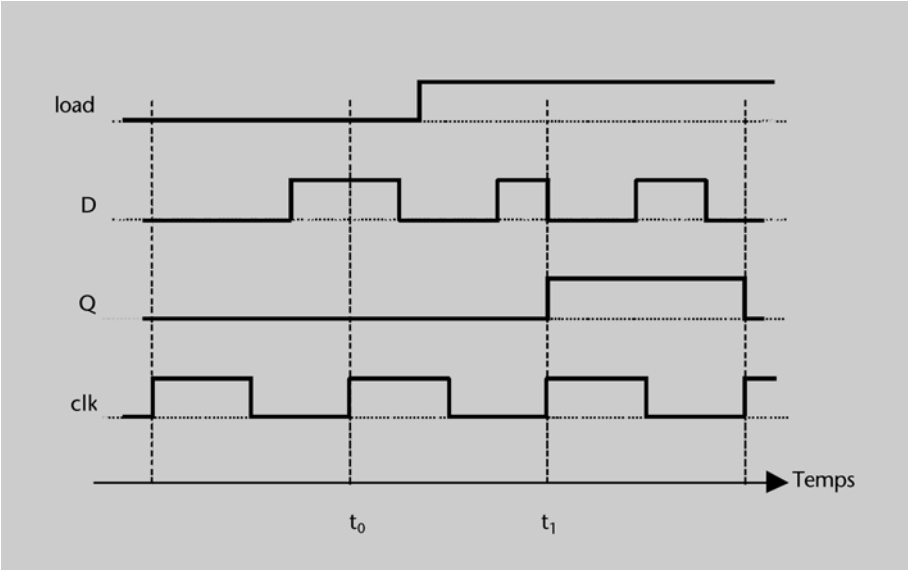
Figura 6



load	D	clk	Q^+
0	x	x	Q
1	0	↑	0
1	1	↑	1

La figura 7 mostra el cronograma del comportament d'un biestable D amb senyal de càrrega. Podem veure que en l'instant t_0 el valor de *Q* no varia, tot i que $D = 1$, perquè el senyal *load* està a 0. Quan *load* = 1, llavors el biestable funciona tal com s'ha estudiat en l'apartat anterior.

Figura 7



2.3. Entrades asíncrones

El valor d'un biestable D pot variar en els instants de flancs ascendents segons el valor que hi hagi a les entrades *D* i *load*. Ara bé, cal tenir la capacitat de donar-li un valor inicial: el valor que prendrà quan es posi en marxa un circuit.

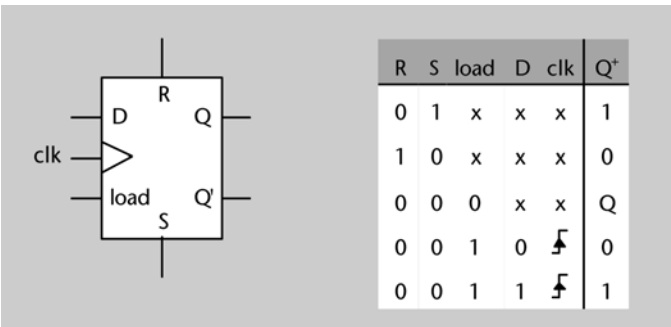
Les **entrades asíncrones** d'un biestable permeten de modificar el seu valor de manera instantània, independentment del valor del senyal de rellotge i de les entrades *D* i *load*. Es diu que les entrades asíncrones tenen més prioritat que la resta d'entrades.

Els biestables solen tenir dues entrades asíncrones:

- *R* (de l'anglès *Reset*): en el moment en què es posa a 1, el biestable es posa a 0.
- *S* (de l'anglès *Set*): en el moment en què es posa a 1, el biestable es posa a 1.

La figura 8 mostra la representació gràfica d'un biestable D amb entrades asíncrones i senyal de càrrega, i la taula de veritat que descriu el seu comportament.

Figura 8

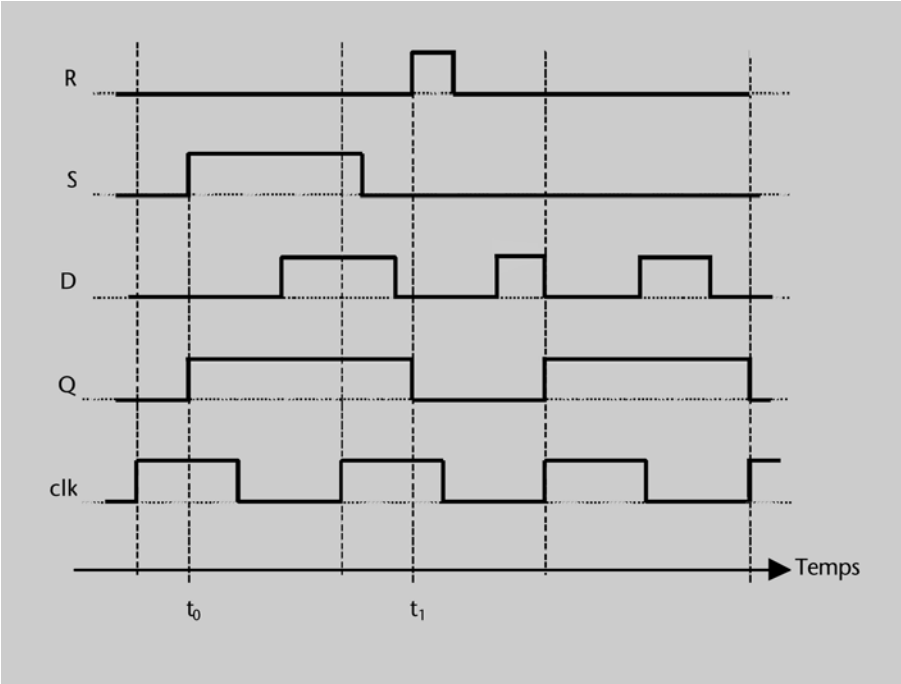


R	S	load	D	clk	Q^+
0	1	x	x	x	1
1	0	x	x	x	0
0	0	0	x	x	Q
0	0	1	0	↑	0
0	0	1	1	↑	1

Si tant *R* com *S* valen 1, el comportament del biestable no es pot predir. En els circuits reals sempre es garanteix que aquesta situació no es produirà mai.

La figura 9 mostra el cronograma del comportament d'un biestable D amb entrades asíncrones i senyal de càrrega (per a simplificar el dibuix, suposem que *load* = 1, tota l'estona). Es pot observar que el biestable es posa a 1 en l'instant t_0, encara que aquest no coincideixi amb un flanc ascendent,i es manté a 1 mentre $S = 1$, independentment del valor de D. Igualment, es posa a 0 en l'instant t_1. En canvi, mentre totes dues entrades asíncrones estan a 0, el biestable modifica el seu valor només en els moments d'un flanc ascendent, d'acord amb el valor de D.

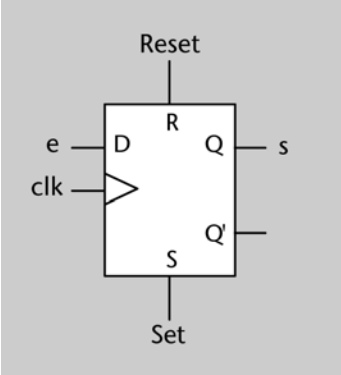
Figura 9

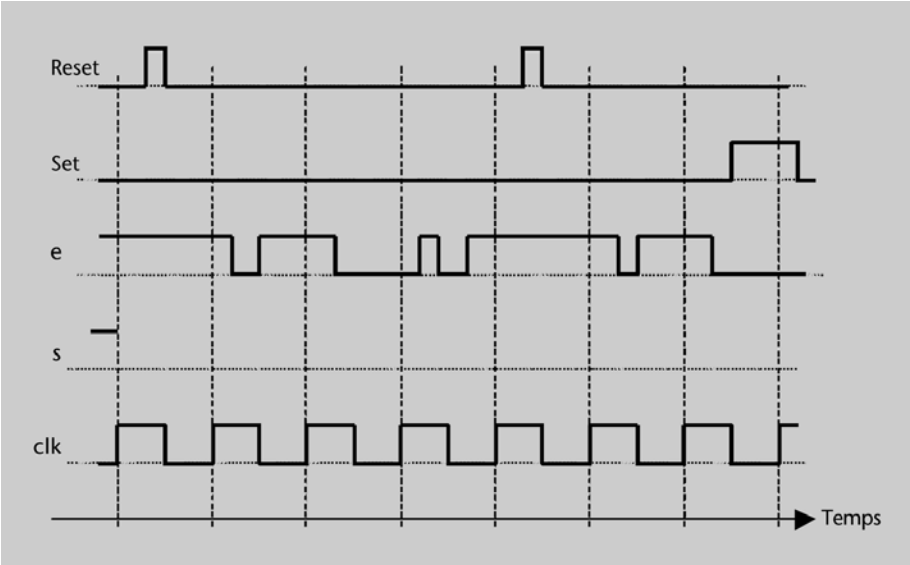


Quan no dibuixem en un circuit els senyals *load*, *R* i *S* d'un biestable, assumirem per defecte que valen 1, 0 i 0, respectivament.

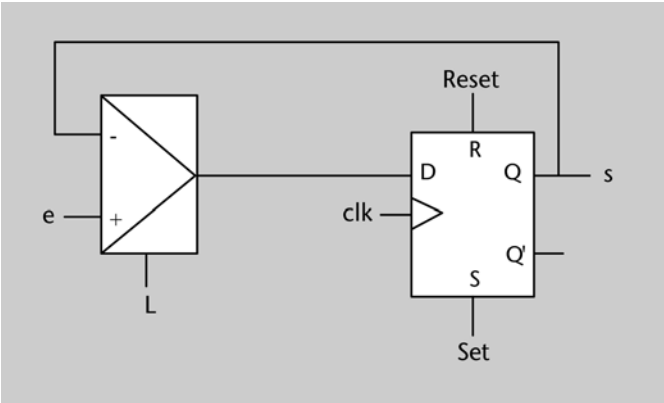
Activitats

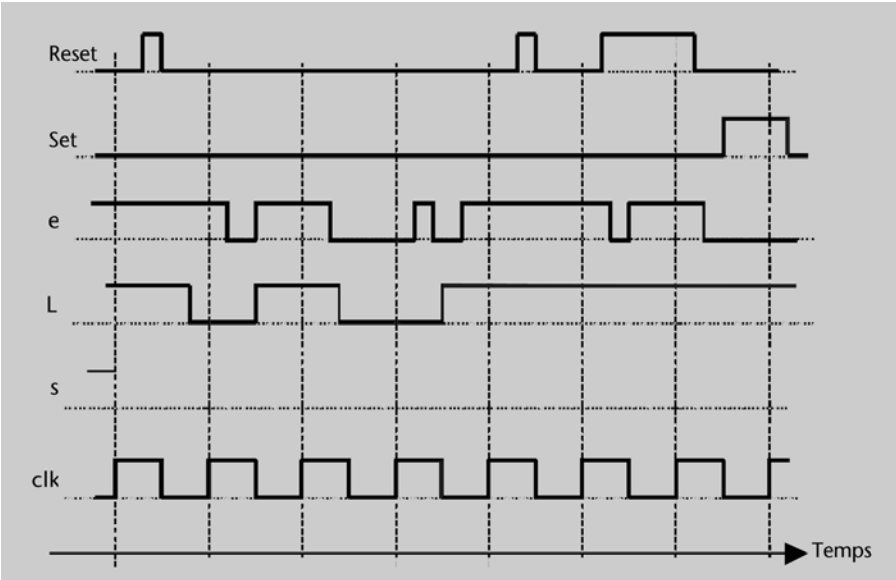
6. Completeu el cronograma que correspon al circuit de la figura, suposant que inicialment la sortida *Q* val 1.





7. 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?



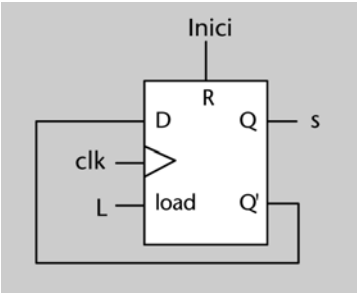


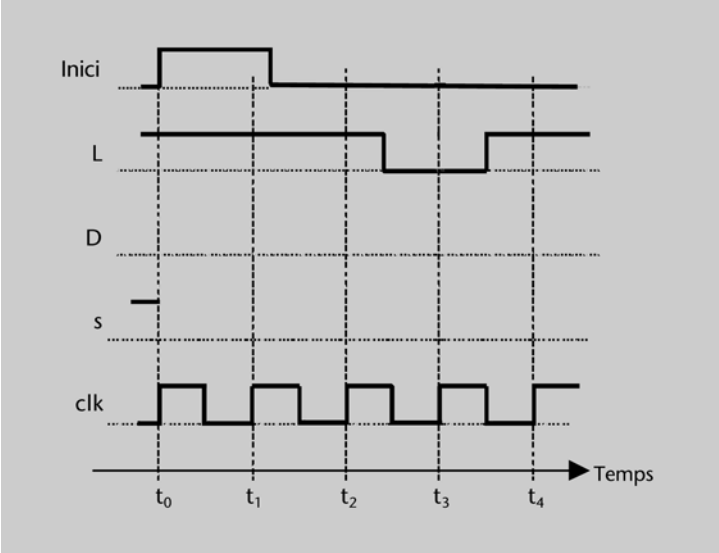
Inicialització dels circuits

Els circuits seqüencials solen tenir un senyal que actua de manera asíncrona i té per missió inicialitzar el circuit. Aquest senyal, que anomenarem *Inici*, està connectat a les entrades asíncrones dels biestables (a *R* o *S* segons si el valor inicial ha de ser 0 o 1). Quan el circuit es posa en funcionament, el senyal *Inici* val 1 durant un cert interval de temps, i després baixa a 0 (es diu que **fa un pols** a 1); un pols sempre dura més d'un cicle de rellotge (és a dir, sempre es produeix almenys un flanc ascendent mentre *Inici* val 1). Durant el funcionament normal del circuit, *Inici* roman a 0. Si en algun altre moment *Inici* fa un altre pols a 1, el circuit es reinicialitza.

Activitats

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





3. Blocs seqüencials

3.1. Registre

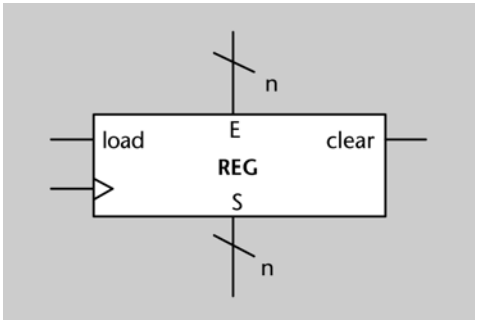
Hem vist que un biestable permet guardar el valor d'un bit. Per a guardar el valor d'un mot de *n* bits, caldran *n* biestables D.

> Un **registre** és un bloc seqüencial format per *n* biestables D, que permet guardar el valor d'un mot de *n* bits.

La figura 10 mostra la representació gràfica d'un registre. Es pot veure que té els senyals següents:

- Una entrada de dades de *n* bits, *E*. Cadascun dels bits d'aquest bus està connectat amb l'entrada *D* d'un dels *n* biestables que formen el registre.
- Una sortida de dades de *n* bits, *S*, que és un bus format per les sortides *Q* dels *n* biestables que formen el registre.
- Dues entrades de control d'un bit, *load* i *clear*. Aquests dos senyals estan connectats respectivament al senyal *load* i a l'entrada asíncrona *R* de cadascun dels biestables del registre.
- Una entrada de rellotge, connectada a les entrades de rellotge de tots els biestables.

Figura 10



El funcionament del registre és el següent:

- El senyal *clear* serveix per a posar el contingut del registre a 0. Atès que es connecta amb les entrades *R* dels biestables, és un senyal asíncron, és a dir, actua independentment del rellotge, i és el més prioritari, de manera que quan està a 1 els *n* bits del registre es posen a 0, independentment del valor dels altres senyals.

- Quan *clear* està a 0, llavors els *n* biestables que formen el registre es comporten com a *n* biestables *D* amb senyal de càrrega.

Aquest funcionament es pot expressar mitjançant aquesta taula de veritat:

clear	load	clk	S^+
1	x	x	0
0	0	x	S
0	1	↑	E

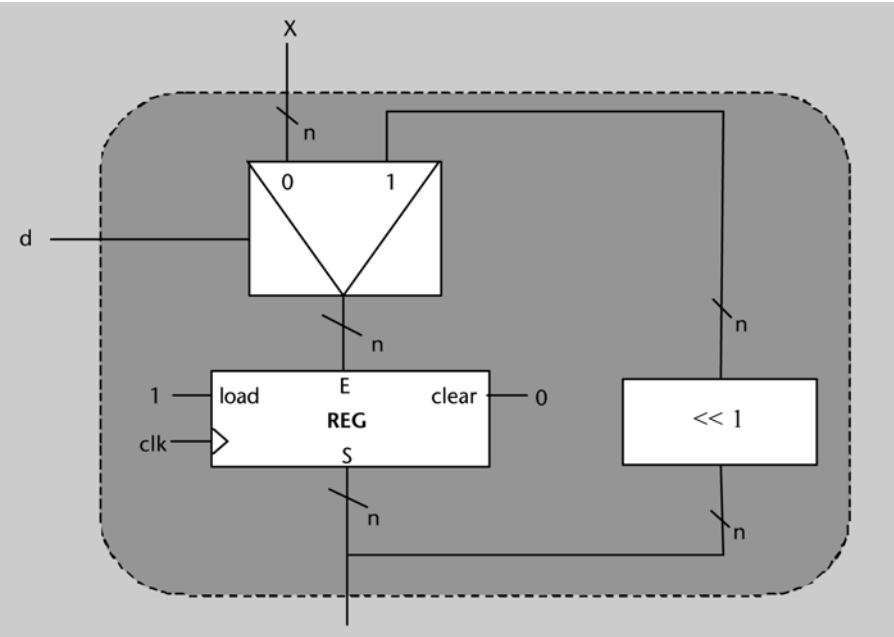
Si en un registre no hi dibuixem els senyals *load* o *clear*, assumirem que estan a 1 i a 0, respectivament.

Quan modifiquem el valor d'un registre fent que es carregui amb el valor que hi ha a l'entrada *E*, diem que fem una **escriptura** en el registre.

Quan analitzem el contingut d'un registre a partir de la sortida S, diem que fem una **lectura**.

A partir d'un registre i blocs combinacionals, es poden dissenyar circuits amb una funcionalitat determinada. Per exemple, el circuit de la figura 11 permet que el registre es pugui carregar amb el valor de l'entrada *X* o que pugui desplaçar el seu contingut 1 bit a l'esquerra, en funció del senyal *d*.

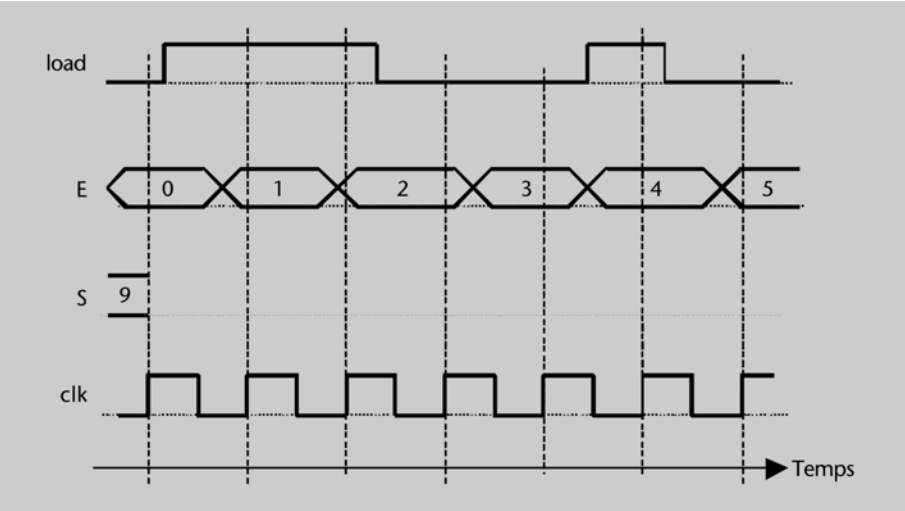
Figura 11



Activitats

9.La figura següent mostra els valors dels senyals *E* i *load* d'un registre de vuit bits durant un cert interval de temps. Indiqueu en la línia etiquetada com a *Temps* els ins-

tants 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 de *S* és 9.

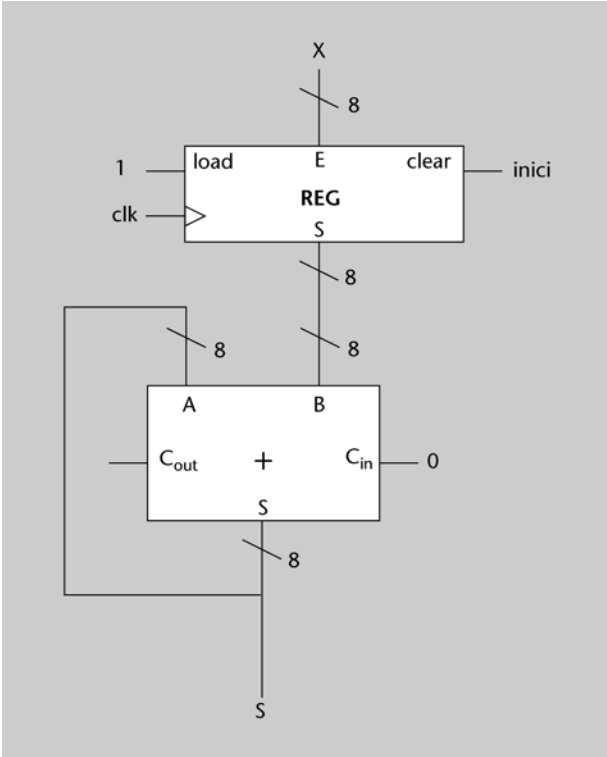


Interpretació d'un cronograma

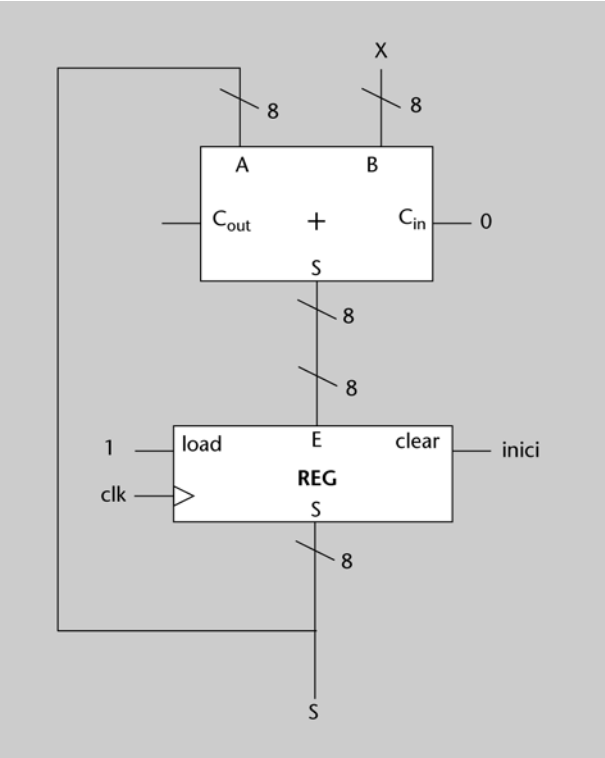
En un cronograma, els valors de mots de *n* bits es dibuixen mitjançant un hexàgon, dins el qual s'escriu el valor del senyal en decimal. Els punts de contacte entre els extrems dels hexàgons indiquen el moment en què el mot canvia de valor.

10. Es vol dissenyar un circuit amb una entrada *X* de 8 bits que codifica un nombre natural en binari, i una sortida *S* també de 8 bits que mostri en tot moment la suma acumulada dels valors que ha tingut *X* des de la inicialització del circuit i fins al cicle present.

a) Un dissenyador inexpert proposa aquest circuit. Per què no és vàlid?



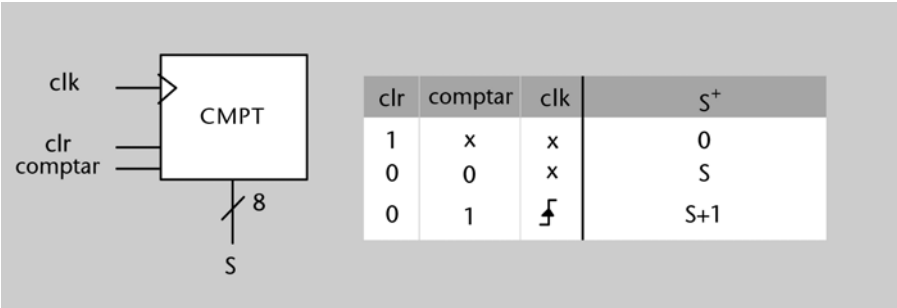
b) Un dissenyador amb una mica més d'experiència proposa aquest altre circuit. En què millora l'anterior? Aquest circuit també té un inconvenient, quin és? És possible evitar-lo?



11. Dissenyeu un bloc seqüencial comptador, CMPT, amb una sortida *S* de 8 bits. Aquesta sortida és un nombre natural (codificat en binari) que, sempre que el senyal d'entrada *comptar* valgui 1, s'ha d'incrementar a cada flanc del rellotge segons aquesta expressió:

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

(per tant, després de valer 255 passa a valer 0). El bloc té, a més, una entrada asíncrona *clr* que quan val 1 posa la sortida a 0. El funcionament del circuit es mostra en la figura següent.



clr	comptar	clk	S^+
1	x	x	0
0	0	x	S
0	1	↑	S+1

12. Dissenyeu un circuit seqüencial amb una entrada de dades *E* de 8 bits, dues entrades de control c_1 i c_0 d'1 bit i una sortida *S* de 8 bits. *E* i *S* codifiquen nombres naturals en binari. El funcionament del circuit el descrivim en la taula següent:

c_1	c_0	*clk*	S^+
0	0	↑	0
0	1	↑	*E*
1	0	↑	(*S*+1) mod 256
1	1	↑	(*S*+*E*) mod 256

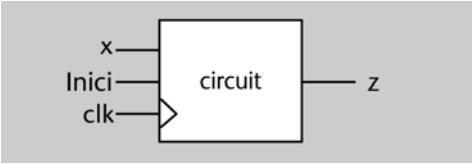
Inicialment *S* ha de tenir el valor 0. El circuit no pot contenir més d'un bloc sumador.

13. Es vol dissenyar un circuit que controli l'estat d'ocupació d'un pàrquing, en concret posant en verd un semàfor mentre hi quedin places lliures i posant-lo en vermell quan estigui ple. El semàfor es posa verd quan el senyal *ple* val 0 i es posa vermell quan val 1.

El pàrquing té capacitat per a 500 vehicles. Quan hi entra un cotxe es produeix un pols al senyal *entra*, i quan en surt un cotxe es produeix un pols al senyal *surt*. Altres sistemes de control del pàrquing garanteixen que en un mateix cicle mai no entrarà i sortirà un cotxe alhora, i generen un pols al senyal *Inici* quan el pàrquing obre les portes, moment en el qual no hi haurà cap cotxe.

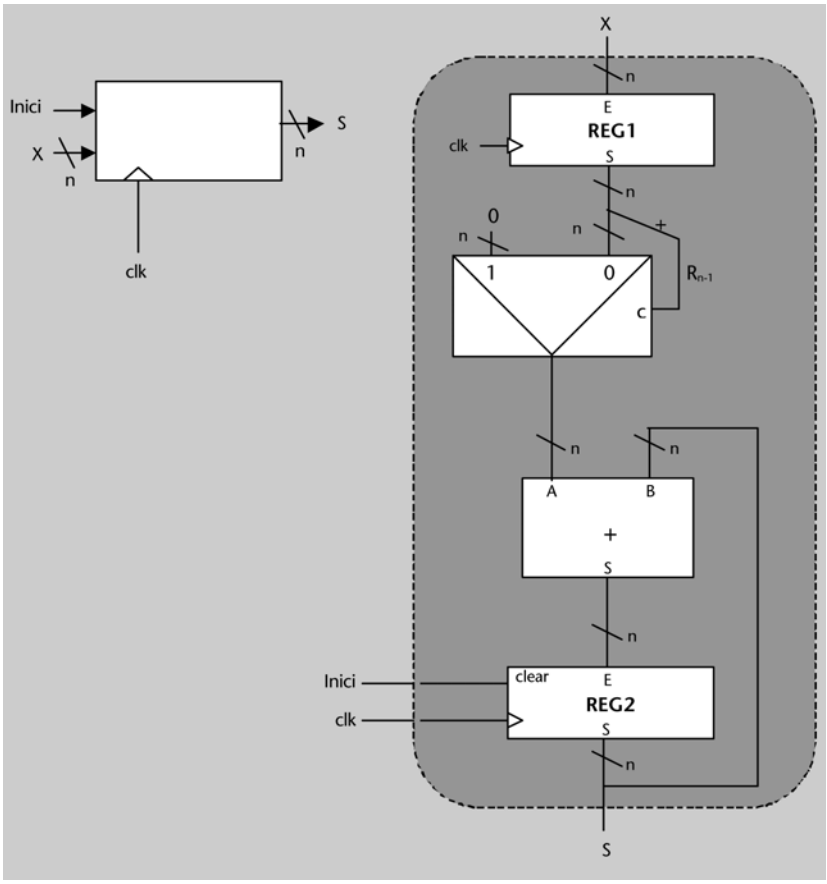
a)Quins senyals d'entrada i de sortida ha de tenir el circuit? De quants bits cadascun?
b)El circuit ha de ser seqüencial, per què?
c)Dissenyeu el circuit, indicant l'amplada de tots els busos.

14. Utilitzant un registre de quatre 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. Els diferents valors de *x* són els que tingui aquest senyal en arribar cada flanc de rellotge. Quan reconeix la seqüència, el circuit ha de posar el senyal de sortida *z* a 1. El circuit té un altre senyal d'entrada, *Inici*, que fa un pols a 1 per a inicialitzar el circuit.



15. Analitzeu què fa el circuit de la figura, tenint en compte el següent:

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

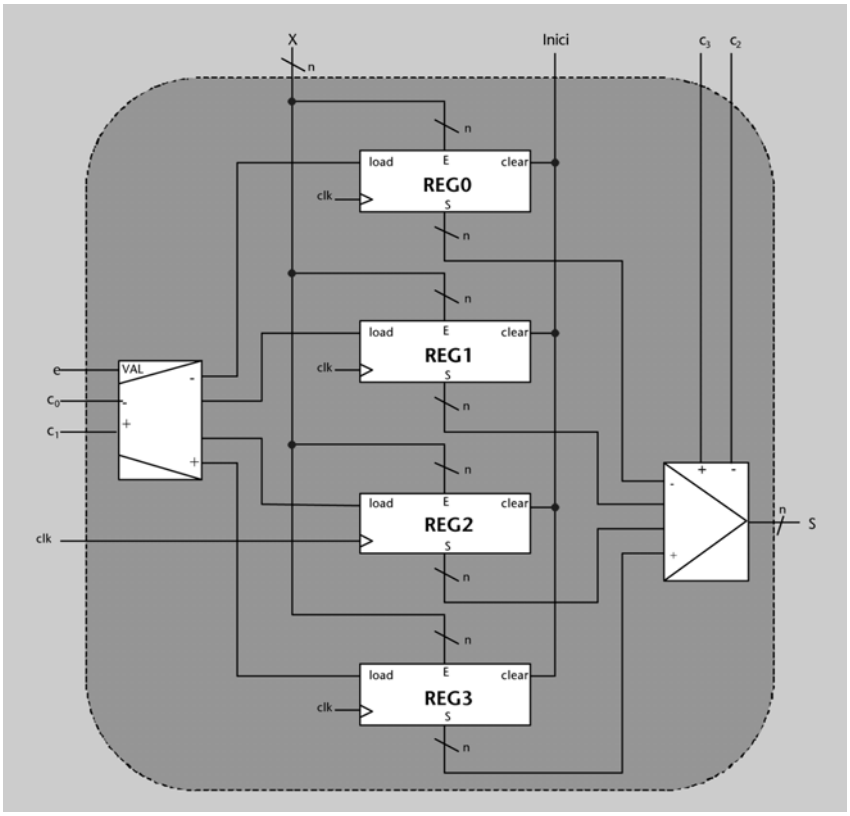


Precisions sobre els gràfics

En els circuits seqüencials, assumirem sempre que hi ha un senyal únic de rellotge (*clk*). En les figures, però, a vegades no es connecten totes les entrades de rellotge amb una mateixa línia, per tal d'aclarir el dibuix.

En general, si en un circuit hi ha més d'un punt identificat per un mateix nom de senyal s'entén que els punts estan connectats, encara que no estiguin units per una línia. Per exemple, a la figura de l'esquerra s'escriu dues vegades "clk", però totes dues corresponen al mateix senyal.

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



Intersecció de cables

A vegades s'introdueixen punts en les interseccions per a aclarir el disseny de circuit.

17. Es vol dissenyar un circuit que controli l'entrada a un edifici d'accés restringit, a la porta del qual hi ha un dispositiu amb cinc tecles numèriques (del 0 al 4) i una tecla *Entrar*. Qui vulgui entrar a l'edifici ha de teclejar un codi al teclat numèric i prémer *Entrar*; això generarà un pols a 1 al senyal *entrar*, d'un cicle de rellotge de durada, i posarà al senyal *codi* la representació en binari del codi que s'acaba d'escriure. Es pot prémer qualsevol nombre de tecles numèriques, però només es tindran en compte les tres últimes que s'hagin escrit abans de prémer *Entrar*.
Si el codi coincideix amb la contrasenya correcta, s'encén un llum verd (s'aconsegueix activant el senyal *verd*) i la porta s'obre. Si el codi és diferent de la contrasenya, s'encén un llum groc (s'aconsegueix activant el senyal *groc*) i la porta no s'obre. Si s'escriu un codi equivocat tres vegades seguides, s'encén un llum vermell (s'aconsegueix activant el senyal *vermell*) i la porta es bloqueja fins que no vingui el porter i reinicialitzi el sistema (generant un pols al senyal *Inici*).
Els llums s'han d'apagar un cert nombre de segons després que s'hagin encès, i també s'han de mantenir apagats mentre ningú no hagi escrit cap codi. Però d'això se n'ocupa un altre subsistema (és a dir, només us heu de preocupar que quan algú hagi escrit un codi s'encengui el llum apropiat).

a) Quines entrades i sortides té el circuit? De quants bits cadascuna?
b) Dissenyeu el circuit, suposant que la contrasenya correcta no varia mai i que disposeu d'un registre que estarà sempre carregat amb la seva codificació en binari. Indiqueu l'amplada de tots els busos.
c) Quins elements caldria afegir-hi per permetre que es pogués canviar la contrasenya?

3.2. Banc de registres

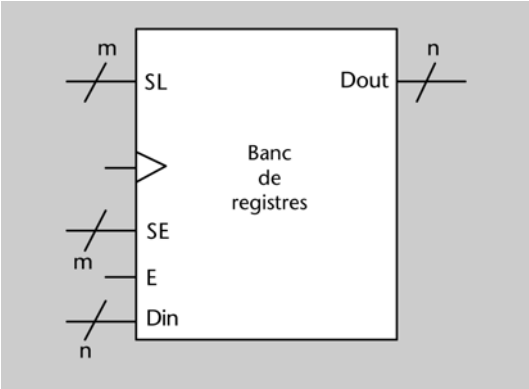
Un **banc de registres** és una agrupació d'un cert nombre de registres, tots del mateix nombre de bits. El contingut dels registres es pot llegir i modificar gràcies als **ports de lectura i d'escriptura**.

El nombre de registres d'un banc sempre és una potència de 2. Els 2^m registres estan numerats des de 0 fins a $2^m - 1$.

La figura 12 mostra la representació gràfica d'un banc de registres. Es pot veure que disposa dels senyals següents:

- Una entrada de selecció de lectura, *SL*, de *m* bits (si 2^m és el nombre de registres del banc).
- Una sortida *Dout*, de tants bits com els registres del banc. *SL* i *Dout* formen el **port de lectura del banc**.
- Una entrada de selecció d'escriptura, *SE*, de *m* bits.
- Una entrada de permís d'escriptura, *E*, d'un bit.
- Una entrada *Din*, de tants bits com els registres del banc. *SE*, *E* i *Din* formen el **port d'escriptura del banc**.

Figura 12



Banc de registres

En general, un banc de registres pot tenir diversos ports de lectura i d'escriptura. El nombre de ports de cada tipus determina el nombre d'operacions de lectura i d'escriptura que es poden fer simultàniament. Per exemple, si té dos ports de lectura i un d'escriptura, es poden llegir dos registres i escriure'n un altre simultàniament. En aquesta assignatura sempre tindran un port d'escriptura i un de lectura.

El banc de registres funciona de la manera següent:

- Per a fer una lectura, cal posar a l'entrada *SL* la codificació binària del número del registre que es vulgui llegir. Llavors, el contingut d'aquest registre estarà present a la sortida *Dout*.
- Per a fer una escriptura, cal posar a *SE* la codificació binària del número de registre que es vulgui escriure i posar l'entrada *E* a 1. Quan es produeixi el proper flanc ascendent de rellotge, el valor que hi hagi a *Din* s'escriurà en el registre indicat per *SE*.

Atenció

Fixem-nos que el banc "sempre s'està llegint"; dit d'una altra manera, a *Dout* hi ha en tot moment el contingut del registre indicat per *SL*. En canvi, només s'hi escriu en els cicles en què *E* = 1; l'escriptura es realitzarà al final del cicle, coincidint amb el proper flanc de rellotge.

Com es pot veure, l'entrada *E* funciona com un senyal de càrrega: si està a 0, no es pot modificar el contingut de cap registre del banc.

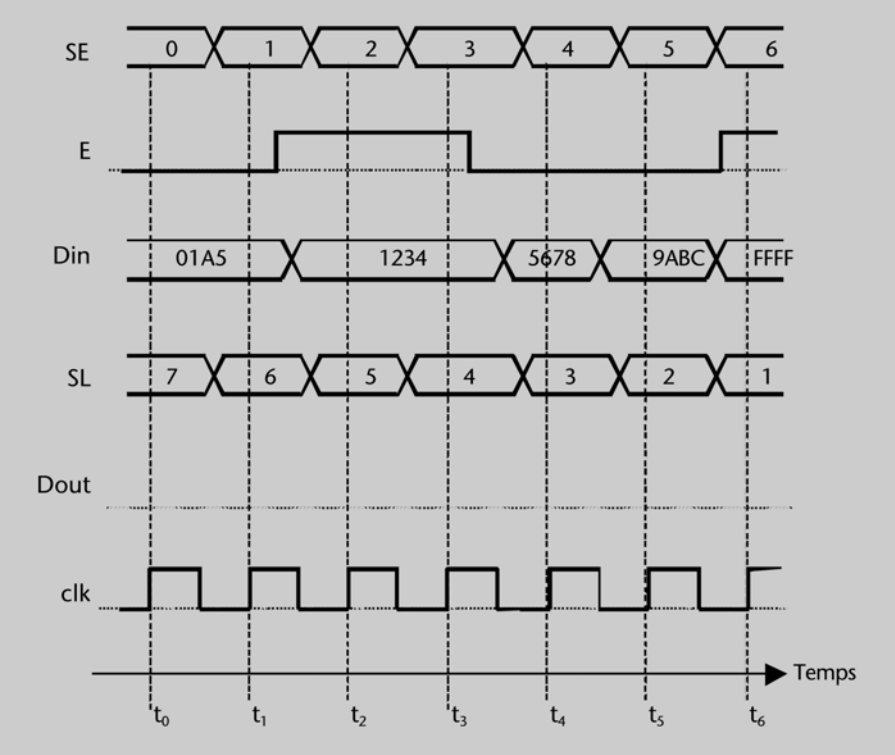
Activitats

18. Es disposa d'un banc de registres de vuit registres de 16 bits. A partir del cronograma que es mostra a continuació, feu el següent:

a) Indiqueu quins registres s'escriuen, en quin moment i amb quin valor.
b) 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.

19. Implementeu un circuit amb la mateixa funcionalitat que el de l'activitat 16, però sense el senyal *Inici*, usant només un banc de quatre registres.

3.3. Memòria RAM

La **memòria RAM** és un bloc seqüencial que permet de guardar el valor d'un cert nombre de mots (2^m) d'un cert nombre de bits (n).

RAM

La denominació RAM prové de l'anglès *random access memory* (memòria d'accés aleatori). Se li va donar aquest nom perquè el temps que es triga a fer una lectura o una escriptura no depèn del mot al qual s'accedeixi (a diferència del que passava en altres dispositius de memòria que es feien servir en els primers computadors).

La funcionalitat d'una memòria RAM, doncs, és similar a la d'un banc de registres. Les diferències entre ambdós blocs són les següents:

- La mida: un banc de registres sol guardar unes quantes desenes de mots, mentre que una memòria RAM en pot guardar diversos milions.

- La velocitat: per com s'implementen físicament una i altra, el temps de resposta (retard) d'una memòria RAM és molt més gran que el d'un banc de registres (i, per tant, aquest últim és més ràpid).

- La implementació interna d'ambdós blocs és molt diferent.

- En un banc de registres les escriptures es fan coincidint amb els flancs ascendents del rellotge. En canvi, la memòria no té senyal de rellotge: les escriptures són efectives un cert interval de temps després d'haver donat l'ordre d'escriure.

En aquest curs no s'estudiaran detalladament aquestes qüestions. En tenim prou amb la idea que un banc de registres és petit i ràpid, i una memòria RAM és gran i lenta. També assumirem que la memòria RAM està sincronitzada de la mateixa manera que els bancs de registres, amb un senyal de rellotge.

Com en el cas de la memòria ROM que s'ha vist en el mòdul "Els circuits lògics combinacionals", la memòria RAM es pot veure com un arxivador amb calaixos, numerats amb una adreça. Cada calaix guarda un mot. Anomenem *M*[*i*] el mot guardat en el calaix amb adreça *i*.

Nota

Les memòries també poden tenir un cert nombre de ports de lectura i d'escriptura, que determinen el nombre d'operacions que s'hi poden fer simultàniament. En una memòria amb un sol port de lectura/escriptura com la que presentem en aquests apunts, només es pot fer o bé una lectura o bé una escriptura en cada moment.

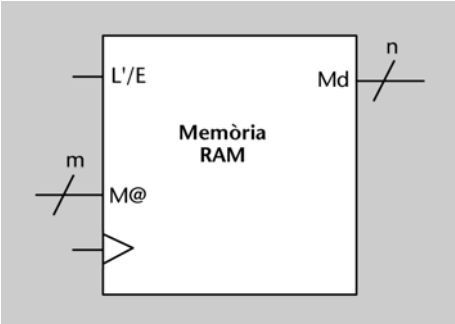
La figura 13 mostra la representació d'una memòria RAM amb un sol **port de lectura/escriptura**. Es pot veure que disposa dels senyals següents:

- Una entrada d'adreces, *M@*. Si la memòria té capacitat per a 2^m mots, l'entrada d'adreces tindrà *m* bits.

- Una entrada/sortida de dades, *Md*, de *n* bits (si els mots que guarda la memòria són de *n* bits).

- Una entrada de control, *L'/E*, que indica en tot moment si s'ha de fer una lectura o una escriptura.

Figura 13



El funcionament de la memòria és el següent:

- Si *L'/E* = 0, llavors es fa una lectura: pel bus *Md* surt el valor del mot que està guardat en l'adreça indicada per *M@*. Si *M@* canvia mentre *L'/E* = 0, *Md* variarà també immediatament (en aquest curs assumim que les lectures a memòria tenen retard 0).

- Si *L'/E* = 1, llavors es fa una escriptura: el mot indicat per *M@* pren el valor que hi ha a *Md* en el primer flanc de rellotge que es produeixi després

d'activar *L'/E* (assumim que les escriptures tampoc no tenen retard). Mentre *L'/E* = 1, el bus *Md* pren el valor 0 (a diferència del que passa en els bancs de registres, que "sempre estan llegint").

La taula de veritat següent resumeix el funcionament de la memòria RAM.

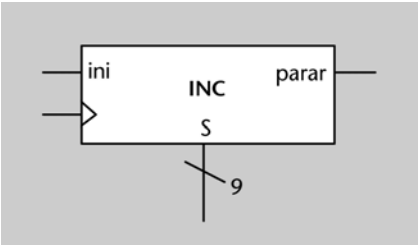
L'/E	
0	Md := M[M@]
1	M[M@] := Md

La capacitat d'una memòria RAM se sol mesurar en bytes (mots de vuit bits). Com hem dit, sol contenir diversos milions de mots, i, per això, per a indicar la seva capacitat se solen fer servir les lletres k, M i G, que tenen els significats següents:

Lletra	Significat	Exemple
k	$2^{10} = 1.024 \cong 10^3$	16 kb (16 kilobytes) = 2^{14} bytes
M	$2^{20} = k \cdot k \cong 10^6$	32 Mb (32 Megabytes) = 2^{25} bytes
G	$2^{30} = k \cdot M \cong 10^9$	2 Gb (2 Gigabytes) = 2^{31} bytes

Activitats

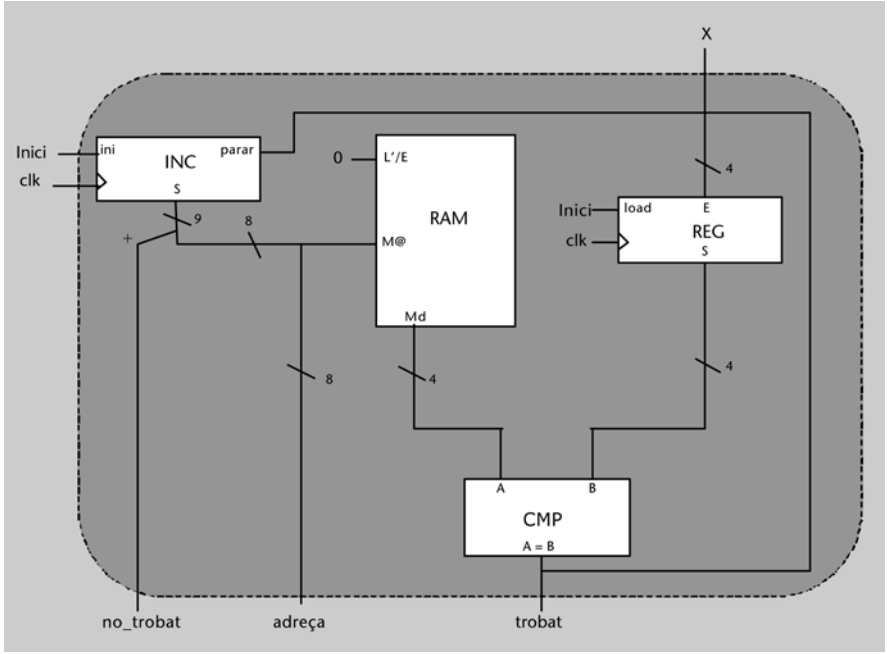
20.
a) Dissenyeu un bloc incrementador, INC, amb una sortida *S* de nou bits (semblant al que s'ha dissenyat en l'activitat 11), que funcioni de la manera següent:



- Quan l'entrada *ini* està a 1, *S* es posa a 0.
- En 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 a l'entrada *ini* hi torni a haver un 1.

A partir del circuit de la figura següent (el bloc INC és el dissenyat en 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 són 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).


X
Inici
clk
ini
INC
parar
S
9
8
+
0
L'/E
RAM
M@
Md
Inici
clk
load
E
REG
S
4
8
4
4
A
B
CMP
A = B
no_trobat
adreça
trobat

4. El model de Moore

4.1. Estat. Transicions

El **model de Moore** és una forma d'expressar o modelitzar el funcionament d'un circuit lògic seqüencial. Es fonamenta en els conceptes d'*estat* i *transicions entre estats*.

Altres modelitzacions

Una altra manera força usual de modelitzar un circuit seqüencial és l'anomenat **model de Mealy** (que no s'estudia en aquesta assignatura).

Per a introduir aquests conceptes farem servir un exemple.

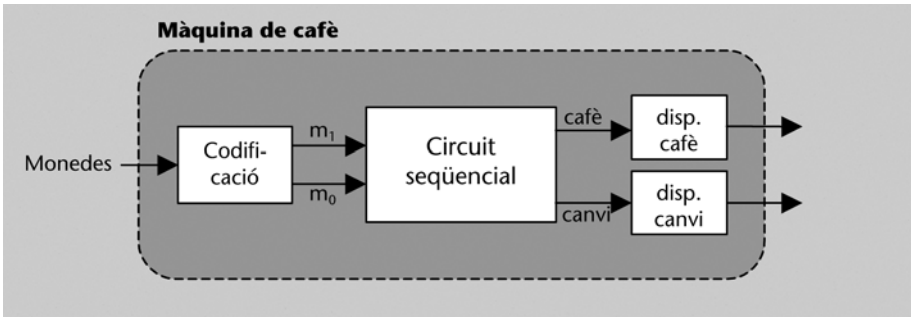
Imaginem un circuit que controla el funcionament d'una màquina expenedora de cafè. Per tal de simplificar, assumirem que la màquina només serveix un únic tipus de cafè sol, i que només admet monedes de 0,5 euros i d'1 euro. La informació sobre les monedes introduïdes es codifica mitjançant dos senyals lògics m_1 i m_0, tal com es mostra a la taula del marge. Els senyals m_1 i m_0 seran les entrades del circuit.

Taula 1

Monedes introduïdes	m_1	m_0
Cap moneda	0	0
Moneda de 0,5 euros	0	1
Moneda d'1 euro	1	0

El preu d'un cafè és d'1,5 euros. La màquina té dos dispositius de sortida, un per a servir el cafè i un per a tornar el canvi. Aquest dos dispositius estan controlats respectivament pels senyals lògics *cafè* i *canvi*, de manera que la màquina donarà cafè o canvi quan el senyal corresponent estigui a 1. La figura 14 mostra l'esquema del funcionament de la màquina.

Figura 14



Per tal de determinar en cada moment si ha de servir cafè o donar canvi, la màquina necessita saber quants diners s'hi han introduït fins al moment. Es poden donar les situacions següents:

- s'hi han introduït 0 euros.
- s'hi han introduït 0,5 euros.
- s'hi ha introduït 1 euro.
- s'hi han introduït 1,5 euros.
- s'hi han introduït 2 euros.

S'anomena **estat** cada situació diferent en què un circuit es pot trobar.

En el nostre exemple, la màquina de cafè pot trobar-se en cinc estats diferents, els descrits per les cinc possibilitats anteriors.

Els senyals de sortida del circuit prendran un valor o un altre segons en quin estat es trobi el circuit. En el cas de la màquina de cafè, el senyal *cafè* estarà a 1 quan s'hi hagin introduït almenys 1,5 euros, i el senyal *canvi* estarà a 1 si se n'hi han introduït dos.

El valor que prenen els senyals de sortida en cada estat s'expressa mitjançant la **taula de sortides**.

La taula de sortides té a l'esquerra els diferents estats i a la dreta el valor que prenen els diferents senyals de sortida en cada estat. La taula de sortides del circuit de la màquina de cafè és la següent:

Estat	cafè	canvi
S'hi han introduït 0 euros	0	0
S'hi han introduït 0,5 euros	0	0
S'hi ha introduït 1 euro	0	0
S'hi han introduït 1,5 euros	1	0
S'hi han introduït 2 euros	1	1

Explicació de la taula de sortides

Quan el senyal *cafè* estigui a 1, la màquina servirà un cafè. Quan el senyal *canvi* estigui a 1 (és a dir, quan l'estat sigui "s'han introduït 2 euros"), la màquina donarà 0,5 euros de canvi (recordem que un cafè val 1,5 euros).

A mesura que el temps avança, el circuit canvia d'estat en funció dels valors que vagin arribant per les entrades. Mentre no s'hi hagi introduït cap moneda, la màquina de cafè es trobarà en l'estat "s'hi han introduït 0 euros". Quan s'introdueixi una moneda de 0,5 euros, passarà a l'estat "s'hi han introduït 0,5 euros"; si la moneda és d'1 euro passarà a l'estat "s'hi ha introduït 1 euro".

El circuit no ha de recordar necessàriament tots els valors que han arribat per les entrades, sinó que els ha de resumir en informacions que siguin rellevants per al seu funcionament. En l'exemple de la màquina de cafè, s'hi pot haver introduït 1 euro mitjançant una sola moneda d'1 o dues monedes de 0,5 euros. A la màquina li és indiferent com s'hagi fet, ja que totes dues accions porten a la mateixa situació: que "ja s'hi ha introduït 1 euro". Per això no hi ha els estats "s'hi han introduït dues monedes de 0,5 euros" i "s'hi ha introduït una moneda d'1 euro", sinó que totes dues informacions es resumeixen en l'estat "s'hi ha introduït 1 euro".

Anomenem **estat actual** l'estat en què es troba la màquina en un instant donat. El valor nou (moneda o absència de monedes) que arribi per l'entrada determinarà quin serà el proper estat en què es trobarà el circuit: l'**estat futur**.

S'anomena **transició** el pas de l'estat actual a un estat futur.

Els circuits seqüencials són sincronitzats per un senyal de rellotge. A cada flanc ascendent, es produeix una transició cap a un estat futur o un altre en funció del valor de les entrades. Les consideracions respecte a l'evolució temporal dels circuits es veuen en un altre subapartat.

La sincronització dels circuits seqüencials s'estudia amb detall en el subapartat 4.3 d'aquest mòdul.

Totes les transicions que es poden donar en un circuit seqüencial s'especifiquen mitjançant la **taula de transicions**.

La taula de transicions té a l'esquerra totes les combinacions possibles d'estats actuals i valors de les entrades, i a la dreta, l'estat futur a què duu cada combinació. A continuació es mostra la taula de transicions de l'exemple de la màquina de cafè (per a fer la taula més llegible, en lloc d'escriure "s'hi han introduït 0 euros" escrivim "zero", i de manera anàloga per a tots els estats).

Estat actual	Entrada	Estat futur
zero	Cap moneda	zero
zero	Moneda de 0,5 euros	mig
zero	Moneda d'1 euro	un
mig	Cap moneda	mig
mig	Moneda de 0,5 euros	un
mig	Moneda d'1 euro	un-i-mig
un	Cap moneda	un
un	Moneda de 0,5 euros	un-i-mig
un	Moneda d'1 euro	dos
un-i-mig	Cap moneda	zero
un-i-mig	Moneda de 0,5 euros	mig
un-i-mig	Moneda d'1 euro	un
dos	Cap moneda	zero
dos	Moneda de 0,5 euros	mig
dos	Moneda d'1 euro	un

En cada transició, l'estat futur pot coincidir o no amb l'estat actual.

Nota

Quan ja s'hi hagin introduït 1,5 o 2 euros, la màquina servirà un cafè i, si és el cas, tornarà el canvi. Si en una d'aquestes situacions s'hi introdueix una altra moneda, el circuit entendrà que correspon a una nova petició de cafè.

Un circuit seqüencial sempre té un estat que reflecteix la situació "encara no ha passat res", és a dir, "no cal recordar cap dels valors que han arribat per l'entrada". Aquest estat s'anomena **estat inicial**.

Quan un circuit es posa en funcionament, és a l'estat inicial. Ara bé, també hi pot ser en altres moments, si la funcionalitat del circuit així ho requereix. En el cas de la màquina de cafè, l'estat inicial és "s'hi han introduït 0 euros". Quan la màquina hagi servit un cafè i mentre no s'hi introdueixin més monedes, tornarà a l'estat inicial.

Una vegada especificat quin és l'estat inicial, la taula de transicions i la taula de sortides descriuen completament el comportament d'un circuit lògic seqüencial d'acord amb el model de Moore.

La taula de transicions també es pot escriure amb els senyals d'entrada codificats en binari (ja que tot circuit treballa només amb senyals lògics, com ja sabem). En l'exemple de la màquina de cafè, les entrades es poden codificar en binari mitjançant els senyals m_1 i m_0, tal com s'ha vist a la taula 1.

Quan escrivim una taula de transicions amb les entrades codificades en binari, hi posem totes les combinacions de les variables d'entrada possibles, encara que algunes no es produeixin mai.

En el nostre exemple, les variables m_1 i m_0 no prendran mai els valors [1 1]; això no obstant, posem aquestes combinacions a la taula de transicions, que es mostra a continuació. El valor de l'estat futur en aquests casos serà *x*, ja que es tracta de combinacions *don't care*.

Estat actual	m_1	m_0	Estat futur
zero	0	0	zero
zero	0	1	mig
zero	1	0	un
zero	1	1	x
mig	0	0	mig
mig	0	1	un
mig	1	0	un-i-mig
mig	1	1	x
un	0	0	un
un	0	1	un-i-mig
un	1	0	dos
un	1	1	x
un-i-mig	0	0	zero
un-i-mig	0	1	mig
un-i-mig	1	0	un
un-i-mig	1	1	x
dos	0	0	zero
dos	0	1	mig
dos	1	0	un
dos	1	1	x

Exemple d'especificació d'un circuit seqüencial amb el model de Moore

Vegem un altre exemple d'especificació d'un circuit seqüencial mitjançant el model de Moore.

Sigui un circuit amb un senyal lògic d'entrada, *x*, i un de sortida, *z*. El senyal de sortida ha de valer 1 sempre que es compleixi que per l'entrada han arribat un nombre parell d'uns i un nombre parell de zeros (recordem que zero és un nombre parell).

El circuit es pot trobar en les situacions, o estats, següents:

- Ha arribat un nombre parell d'uns i un nombre parell de zeros.
- Ha arribat un nombre parell d'uns i un nombre senar de zeros.

- Ha arribat un nombre senar d'uns i un nombre parell de zeros.
- Ha arribat un nombre senar d'uns i un nombre senar de zeros.

El valor del senyal de sortida en cada estat ve donat per la taula de sortides següent:

Estat Ha arribat un nombre...	**Sortida** z
... parell d'uns i parell de zeros	1
... parell d'uns i senar de zeros	0
... senar d'uns i parell de zeros	0
... senar d'uns i senar de zeros	0

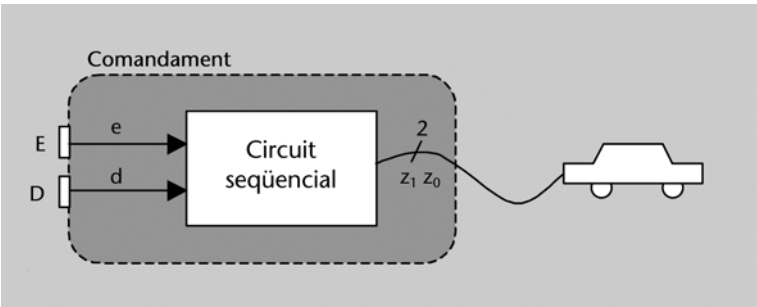
El circuit anirà passant a un estat o un altre segons si l'entrada val 0 o 1. Concretament, la taula de transicions serà la següent:

Estat actual Ha arribat un nombre...	**Entrada** x	**Estat futur** Ha arribat un nombre...
... parell d'uns i parell de zeros	0	... parell d'uns i senar de zeros
... parell d'uns i parell de zeros	1	... senar d'uns i parell de zeros
... parell d'uns i senar de zeros	0	... parell d'uns i parell de zeros
... parell d'uns i senar de zeros	1	... senar d'uns i senar de zeros
... senar d'uns i parell de zeros	0	... senar d'uns i senar de zeros
... senar d'uns i parell de zeros	1	... parell d'uns i parell de zeros
... senar d'uns i senar de zeros	0	... senar d'uns i parell de zeros
... senar d'uns i senar de zeros	1	... parell d'uns i senar de zeros

Per a referir-nos a *estat actual* i *estat futur* també podem escriure Estat i Estat$^+$.

Activitats

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



Si el cotxe està parat, en pitjar qualsevol botó es posa en moviment: gira a l'esquerra si es pitja *E*, gira a la dreta si es pitja *D* i tira endavant si es pitgen tots dos botons alhora. Mentre el cotxe està en moviment, si es pitja *E* farà el següent:

- girarà a l'esquerra si anava recte,
- anirà recte si girava a la dreta,
- continuarà girant a l'esquerra si ja ho feia.

I de manera anàloga quan es pitgi *D*. Quan es pitgin tots dos botons alhora es pararà.

El comandament disposarà d'un circuit seqüencial que rep com a entrada dos senyals *e* i *d* connectats als botons *E* i *D* respectivament (1: pitjat; 0: no pitjat), i genera dos senyals z_1 i z_0, que governaran el cotxe segons la taula que veieu al marge.

z_1	z_0	Acció del cotxe
0	0	Girar a la dreta
0	1	Girar a l'esquerra
1	0	Parar
1	1	Moure's endavant

a) Quins estats té el circuit?
b) Quin és l'estat inicial?
c) Escriviu la taula de sortides i la de transicions.

22. Sigui un circuit seqüencial amb dos senyals d'entrada x i y d'un bit, i un senyal de sortida z també d'un bit. El senyal de sortida s'ha de posar a 1 quan en almenys tres ocasions els valors de x i y hagin estat iguals.

a) Quins estats té el circuit?
b) Quin és l'estat inicial?
c) Escriviu la taula de sortides i la de transicions.

4.2. Representació gràfica: grafs d'estats

L'especificació del funcionament d'un sistema seqüencial mitjançant el model de Moore es representa gràficament amb un **graf d'estats**, de la manera següent:

1) Per cada estat es dibuixa un cercle, amb el nom de l'estat a la part superior. El cercle corresponent a l'estat inicial se senyala amb una fletxa i la paraula *Inici*.

Recordem

Els circuits seqüencials solen tenir un senyal d'entrada *Inici* que fa un pols a 1 per a indicar al circuit que es posi en funcionament.

2) A la part inferior de cada cercle s'hi escriu el valor que prenen els senyals de sortida quan el circuit es troba en l'estat corresponent a aquest cercle.

3) Les transicions es representen mitjançant fletxes o **arcs**, que tenen l'origen en el cercle corresponent a l'estat actual i la punta en el cercle corresponent a l'estat futur. El valor dels senyals d'entrada associat amb la transició s'escriu al costat de l'arc.

Els valors escrits al costat d'un arc també s'anomenen *etiquetes* de l'arc.

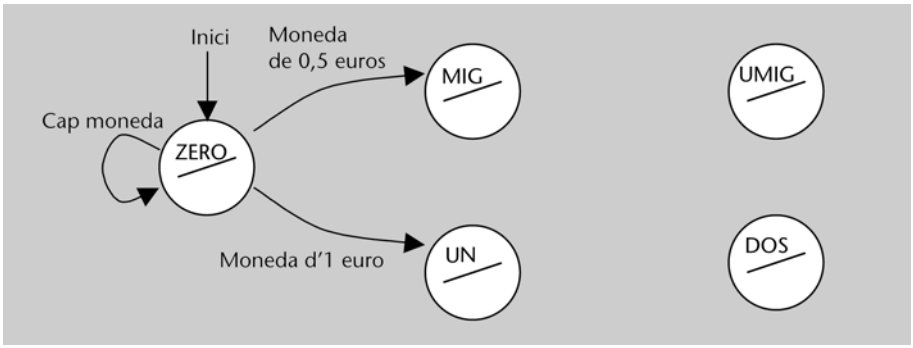
La figura 15 mostra la representació gràfica dels estats de la màquina de cafè, i les transicions que parteixen de l'estat "s'hi han introduït 0 euros". En aquesta figura hem donat als diferents estats aquests noms:

Estat	Nom
S'hi han introduït 0 euros	ZERO
S'hi han introduït 0,5 euros	MIG
S'hi ha introduït 1 euro	UN
S'hi han introduït 1,5 euros	UMIG
S'hi han introduït 2 euros	DOS

Consell pràctic

En especificar un circuit seqüencial mitjançant un graf d'estats, se sol donar a cada estat un nom curt, ja que així es pot escriure còmodament dintre del cercle corresponent. El nom que es doni a cada estat és indiferent, però resulta pràctic que sigui un mnemotècnic que ens remeti d'alguna manera a la situació reflectida per cada estat.

Figura 15



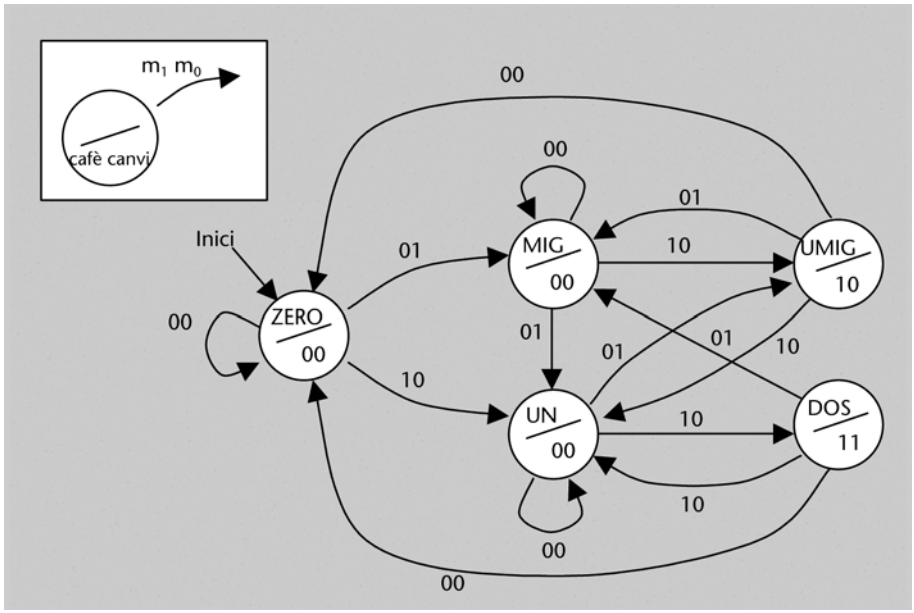
El graf complet es mostra a la figura 16, amb les entrades codificades en binari (d'acord amb la taula 1). Al requadre de la part superior esquerra hi trobem la llegenda del graf.

La **llegenda** d'un graf d'estats indica l'ordre en què s'escriuen els senyals d'entrada a les etiquetes dels arcs i els senyals de sortida dins dels cercles.

La llegenda d'un graf d'estats

Si un graf amb més d'un senyal d'entrada o de sortida no disposa de llegenda, és impossible desxifrar-ne el significat.

Figura 16



Combinacions impossibles

En general, en un graf no hi apareixen les combinacions de les variables d'entrada que no es donaran mai. Per això, a la figura 16 no hi ha cap etiqueta amb la combinació [1 1].

Gràcies a la llegenda sabem que, en les parelles de valors que hi ha a les etiquetes de la figura 16, el de l'esquerra correspon a m_1 i el de la dreta a m_0 (i no al revés). La llegenda ens diu també que, dels valors de sortida, el de l'esquerra correspon al senyal *cafè* i el de la dreta, al senyal *canvi*.

4.2.1. Mecànica de disseny

A partir de l'enunciat del comportament d'un circuit seqüencial, per a trobar-ne el graf d'estats, podem seguir l'algorisme següent:

1) Analitzar quines entrades i sortides té el circuit i determinar la llegenda del graf.

2) Dibuixar un cercle per a l'estat inicial, donar-li un nom i escriure el valor de les sortides en aquest estat.

3) Fer una llista de totes les combinacions de valors que poden prendre els senyals d'entrada en aquest estat. Per a cadascuna, deduir quina transició provoca. Si la transició comporta l'aparició d'un estat inexistent fins al moment, incorporar-lo al graf, donar-li un nom i escriure el valor adequat per a les sortides. Dibuixar l'arc corresponent a la transició.

4) Repetir el pas **3** per a tots els estats nous que hagin aparegut, fins que no n'aparegui cap de nou.

Circuits reconeixedors de seqüència

Vegem un exemple de construcció del graf a partir de l'especificació del funcionament d'un circuit.

Es vol dibuixar el graf d'estats d'un circuit amb un senyal d'entrada *x* i un de sortida *z*, tots dos d'un bit. Inicialment la sortida ha de valer 0. Quan a l'entrada s'hagi produït la seqüència de valors 101, la sortida s'ha de posar a 1. La sortida s'ha de tornar a posar a 0 quan per l'entrada hagi arribat la seqüència de valors 001.

Els circuits que generen un 1 a la sortida quan s'ha produït una seqüència de valors determinada a l'entrada s'anomenen *reconeixedors de seqüència.*

En aquest cas els senyals d'entrada i de sortida són d'un únic bit; per tant, no cal determinar l'ordre dels senyals a la llegenda, només hi hem de posar el nom.

El circuit tindrà un estat inicial, anomenat *INI*, amb sortida 0. Com que l'entrada és d'un únic bit, només pot prendre els valors 0 i 1.

Si estant en l'estat *INI*, per l'entrada arriba un 1, ho hem de recordar, ja que pot ser el principi de la seqüència 101 que volem reconèixer. Apareix, doncs, un estat nou, que anomenem "ha arribat un 1"; la sortida en aquest estat continua valent 0. En canvi, si estant en l'estat inicial arriba un 0, no cal recordar-ho, i ens quedarem en el mateix estat *INI*.

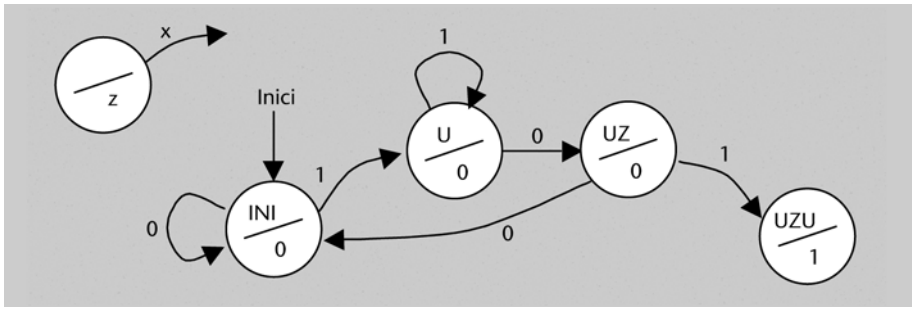
Estant en l'estat "ha arribat un 1", si arriba un 0 hem de passar a un estat nou, que anomenarem "ha arribat la subseqüència 10", amb sortida 0. Si arriba un 1, s'ha trencat la seqüència anterior, però aquest nou 1 pot ser al seu torn l'inici d'una nova seqüència 101. Per tant, ens hem de quedar en el mateix estat "ha arribat un 1".

Situem-nos ara en el nou estat que ha aparegut, "ha arribat la subseqüència 10". Si arriba un 0, tenim que els tres darrers valors que han arribat per l'entrada són 100 i, per tant, cap d'ells no pot formar part de la seqüència 101. Tornem, doncs, a l'estat *INI*. En canvi, si arriba un 1, tenim que els tres darrers valors que han arribat són 101, que és justament la seqüència que el circuit ha de reconèixer. Passarem, doncs, a un estat nou, "ha arribat la seqüència 101", en el qual la sortida val 1.

Donem als estats que han aparegut fins ara els noms següents:

Estat	Nom
Ha arribat un 1	U
Ha arribat la subseqüència 10	UZ
Ha arribat la seqüència101	UZU

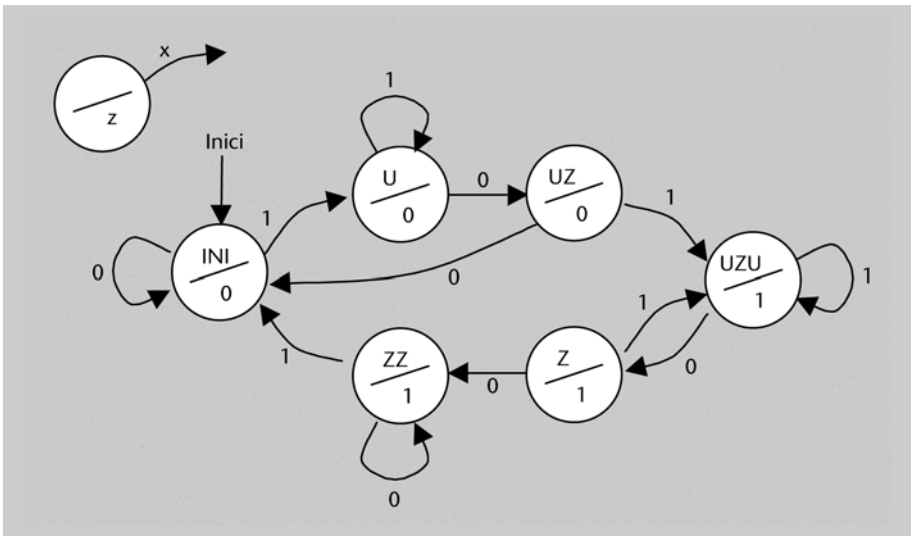
La part del graf que hem construït fins ara es mostra a continuació:



Una vegada ha arribat a l'estat *UZU*, el circuit ha de reconèixer la seqüència 001 per a saber quan ha de tornar a posar la sortida a 0. Raonem de manera anàloga al cas anterior i obtenim que haurà de tenir també els estats següents:

Estat	Nom
Ha arribat un 0	Z
Ha arribat la subseqüència 00	ZZ
Ha arribat la seqüència 001	ZZU

Veiem, però, que l'estat ZZU coincideix amb l'estat inicial, ja que la sortida ha de valer 0 i el circuit ha de reconèixer a partir d'aquest moment la seqüència 101; és a dir, el circuit es troba en la mateixa situació que en començar a funcionar. Finalment obtenim el graf complet següent:

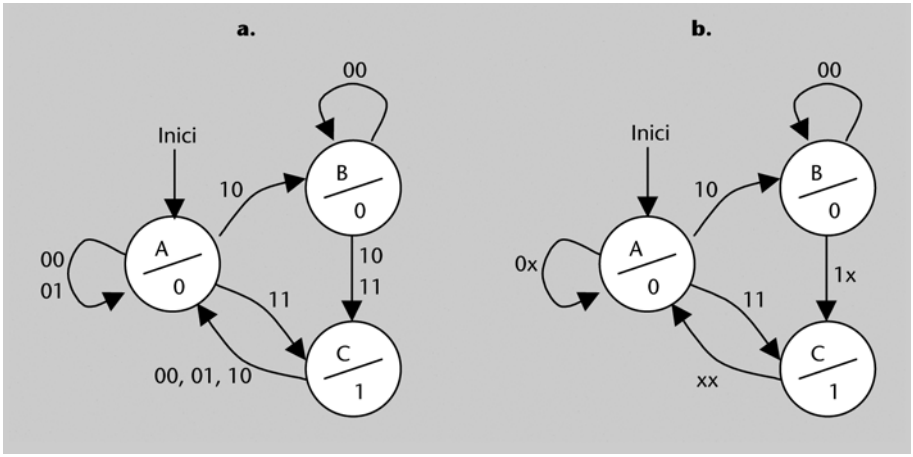


4.2.2. Notació

En un graf, si es produeix una transició d'un estat determinat cap a un mateix estat futur per més d'una combinació de valors dels senyals d'entrada, escriurem les etiquetes corresponents una sota l'altra, o bé separades per comes, tal com es mostra al gràfic **a** de la figura 17. Observem que, en el circuit corresponent a aquest graf, estant en l'estat *B* no es donarà mai la combinació d'entrada [0 1], i estant en l'estat *C* no es donarà mai la combinació [1 1].

En una etiqueta podem escriure també *x* per a referir-nos a un valor qualsevol d'un senyal d'entrada (igual com es fa a les taules de veritat). Quan d'un estat es passa sempre a un mateix estat futur, independentment del valor de les entrades (com és el cas de l'estat *C* en el graf **a** de la figura 17), es pot posar una sola etiqueta amb *x* com a valor de totes les variables, fins i tot si algunes combinacions no es donen mai. Així, els dos grafs de la figura 17 són equivalents (fixem-nos que, estant en l'estat *C*, la combinació d'entrada 11 no es pot produir).

Figura 17



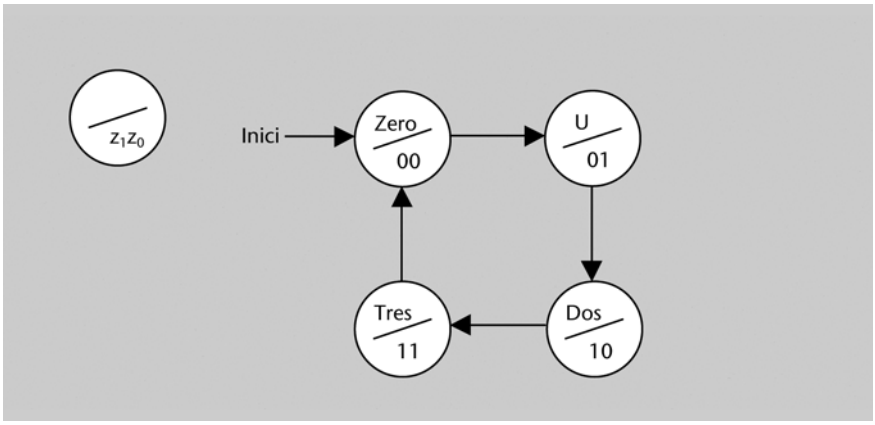
4.2.3. Circuits sense entrades

Un circuit seqüencial pot no tenir cap senyal d'entrada. En aquest cas, sempre es produiran les mateixes transicions entre estats i, per tant, la seqüència de valors a les sortides serà sempre la mateixa.

Els comptadors mòdul *n*

En general, un comptador mòdul *n* és un circuit que genera cíclicament la seqüència de valors 0, 1, ..., *n* – 1.

Imaginem un circuit la missió del qual és generar cíclicament la seqüència de números 0, 1, 2 i 3, codificats en binari. Aquest circuit rep el nom de *comptador mòdul 4*, i el seu graf d'estats és el que es mostra a la figura següent.



Nota

Fixeu-vos que quan el circuit no té cap entrada, en dissenyar-ne el graf d'estats, la llegenda no té cap arc ni cap etiqueta associada al mateix.

Activitats

23. Dibuixeu el graf d'estats del circuit que es descriu a l'activitat 21.
24. Dibuixeu el graf d'estats del circuit seqüencial que es comporta tal com descriuen les taules següents:

a) Estat inicial: *A*

Taula de transicions			
Estat	x_1	x_0	Estat$^+$
A	0	0	A
A	0	1	C
A	1	0	B
A	1	1	B
B	0	0	B
B	0	1	D
B	1	0	A
B	1	1	A
C	0	0	A
C	0	1	C
C	1	0	D
C	1	1	D
D	0	0	D
D	0	1	B
D	1	0	C
D	1	1	C

Taula de sortides			
Estat	y_2	y_1	y_0
A	1	0	0
B	0	1	0
C	0	0	1
D	1	1	1

b) Estat inicial: *C*

Taula de transicions			
Estat	**e_1**	**e_0**	**Estat$^+$**
A	0	x	B
A	1	0	C
A	1	1	A
B	0	0	B
B	0	1	C
B	1	0	A
B	1	1	x
C	0	x	A
C	1	0	A
C	1	1	x

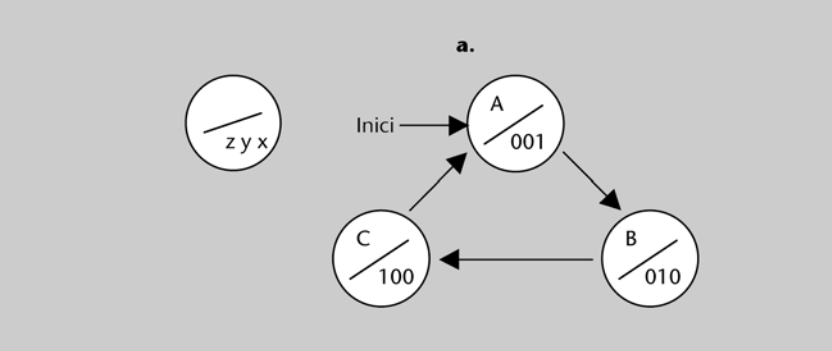
Taula de sortides	
Estat	**z**
A	1
B	1
C	0

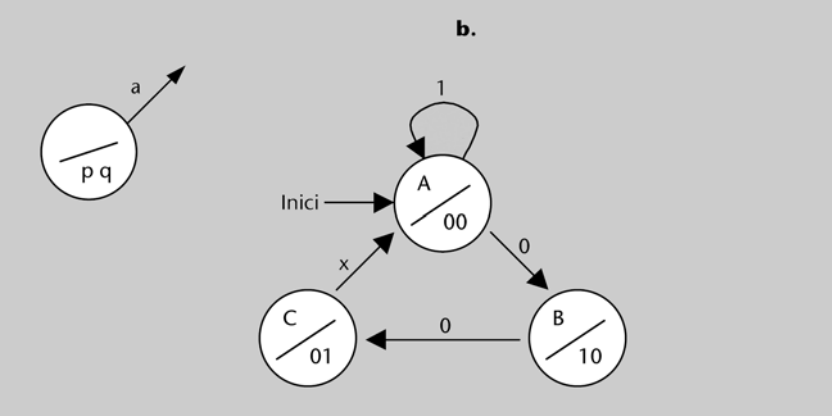
c) Estat inicial: *E2*

Taula de transicions	
Estat	**Estat$^+$**
E0	E2
E1	E0
E2	E1

Taula de sortides	
Estat	**z**
E0	1
E1	0
E2	1

25. Escriviu les taules de transicions i sortides dels circuits que es comporten tal com descriuen els grafs d'estats següents:





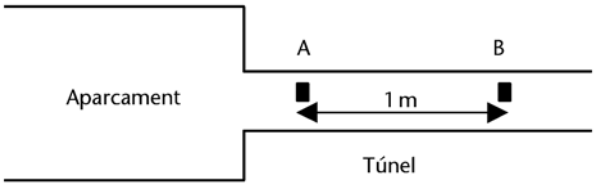
26. Dibuixeu el graf d'estats d'un circuit que funcioni com un comptador reversible mòdul 5. Un comptador reversible compta endavant o endarrere en funció d'un senyal d'entrada *x*:

- $x = 0$: compta endavant
- $x = 1$: compta endarrere

Inicialment la sortida ha de valer 0. La sortida no es codificarà en sistema binari, sinó com s'indica en aquesta taula:

Valor de la sortida	Senyals de sortida $z_3z_2z_1z_0$
0	0 0 0 0
1	0 0 0 1
2	0 0 1 1
3	0 1 1 1
4	1 1 1 1

27. En un aparcament es necessita saber el nombre de cotxes que hi ha en cada moment. Els cotxes entren i surten de l'aparcament 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 entra o surt segons l'ordre en què s'activin els sensors (se suposa que tots els cotxes fan més d'un metre i que entre un cotxe i el següent hi ha més d'un metre). Dos cotxes no es trobaran mai de cara al túnel. Tampoc no hi haurà mai vianants.



Quan un cotxe ha entrat totalment a l'aparcament s'incrementa el nombre de cotxes aparcats, i quan n'ha sortit totalment es decrementa. Mentre passa pel túnel, un cotxe pot parar o fer marxa enrere en qualsevol moment.

Dibuixeu el graf d'estats d'un circuit seqüencial que a partir dels senyals *a* i *b*, provinents respectivament dels sensors *A* i *B* (valdran 1 si hi ha un cotxe davant del sensor i 0 si no n'hi ha cap), generi dos senyals de sortida *més* i *menys* que governaran el comptador de cotxes que hi ha a l'aparcament en cada moment.

4.3. Sincronització

Ja sabem que els circuits seqüencials estan sincronitzats per un senyal de rellotge que descriu cicles periòdics entre els valors 0 i 1.

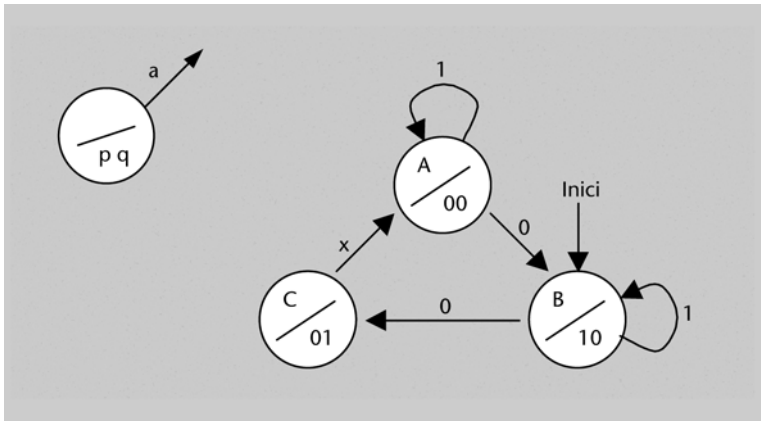
Les transicions entre estats tenen lloc a cada flanc ascendent de rellotge (perquè els estats s'implementen físicament mitjançant biestables).

Vegeu la implementació dels estats mitjançant biestables en el subapartat 4.4 d'aquest mòdul.

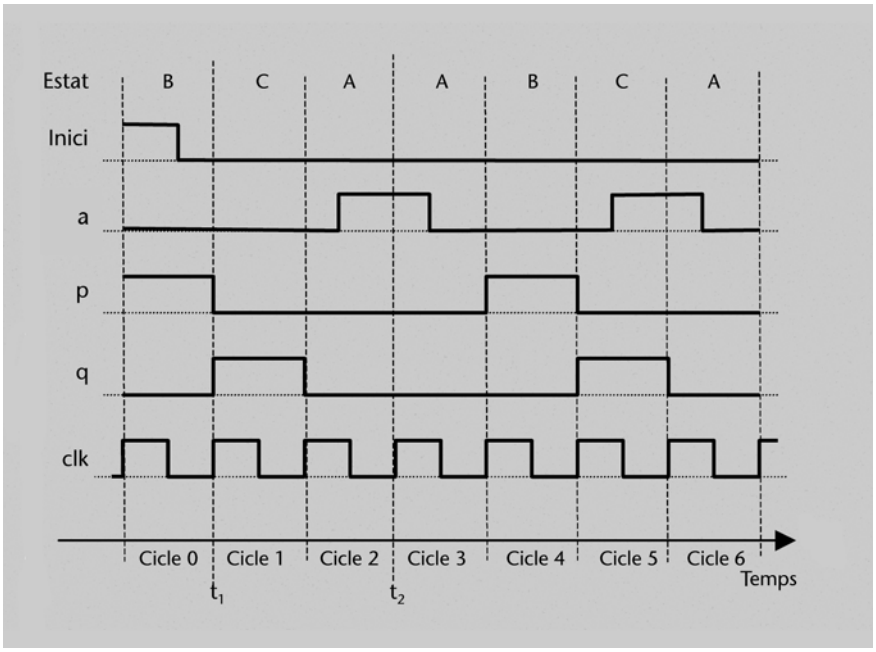
Per tant, el circuit examina el valor dels senyals d'entrada a cada cicle de rellotge. En concret, el valor que fa que es prengui una transició o una altra és el que tenen les entrades en arribar l'instant del flanc. Si dibuixem les entrades en un cronograma, el valor que decideix quina transició es pren és el que tenen en tocar, per l'esquerra, la línia vertical corresponent a un flanc.

Exemple de transicions entre estats

Sigui el graf d'estats que es mostra en aquesta figura:



La figura següent mostra com evoluciona el circuit amb el temps a partir d'una seqüència de valors determinada a l'entrada *a*. En el cicle 0, el senyal *Inici* fa un pols a 1 i fa que el circuit es posi en l'estat *B*. En l'instant t_1 (per l'esquerra), el senyal d'entrada *a* val 0, la qual cosa provoca que el circuit passi a l'estat *C* en aquest instant. Durant el cicle 2, el circuit es troba en l'estat *A*. Com que a l'instant t_2 l'entrada val 1, en aquest moment es produeix una transició cap al mateix estat *A*. La resta del cronograma es determina de manera anàloga.

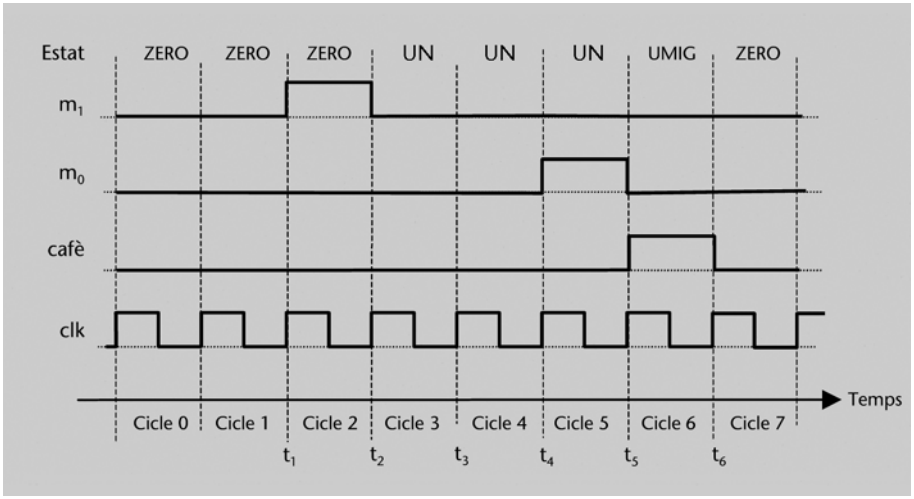


Recordem que el valor dels senyals de sortida en cada moment ve determinat per l'estat en què es troba el circuit.

Reprenem l'exemple de la màquina de cafè. Quan s'introdueix una moneda, passa un interval de temps des que es fica a la ranura fins que cau a la caixa corresponent. Suposem que el sistema que codifica els senyals m_1 i m_0 (recordem la taula 1 i la figura 14) genera polsos d'un cicle de durada: m_0 estarà a 1 durant un cicle quan s'hagi introduït una moneda de 0,5 euros, i m_1 estarà a 1 durant un cicle quan s'hagi introduït una moneda d'1 euro.

La figura 18 mostra una possible evolució temporal del circuit, partint de l'estat *ZERO*. Durant els dos primers cicles no s'ha introduït cap moneda i, per tant, les transicions que s'han produït han dut sempre a l'estat *ZERO*. En el cicle 2, m_1 fa un pols a 1 i indica que s'ha introduït una moneda d'1 euro. Per tant, el circuit passa a l'estat *UN* en l'instant t_2. Les properes dues transicions portaran també a l'estat *UN*, ja que $[m_1\ m_0] = [0\ 0]$ en els instants t_3 i t_4. En el cicle 5, $m_0 = 1$ (s'ha introduït una moneda de 0,5 euros) i, per tant, el circuit passa a l'estat *UMIG* en l'instant t_5. Per tant, durant el cicle 6 la sortida *cafè* val 1 (això farà que s'activi el dispositiu que serveix un cafè). Com que durant aquest cicle les entrades valen 0 (no s'ha introduït cap moneda), el circuit passa a l'estat *ZERO* en l'instant t_6 i, per tant, la sortida *cafè* torna a 0.

Figura 18



Representació dels cicles en un cronograma

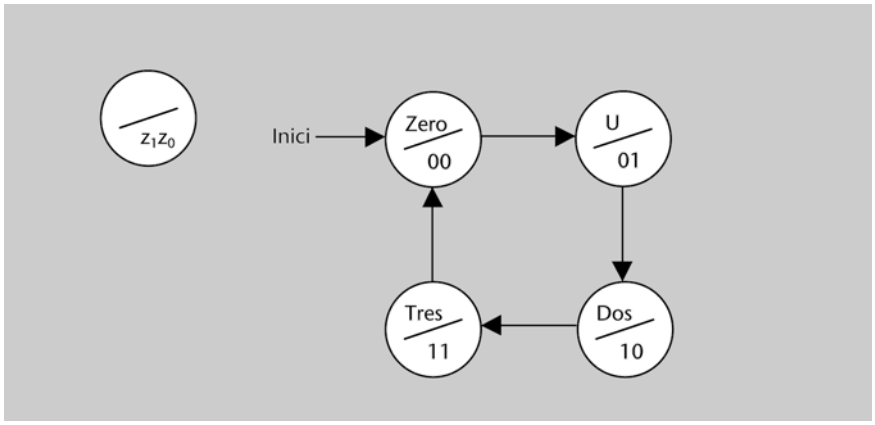
En aquest cronograma no hi hem dibuixat el senyal *Inici*. En general, els cicles que mostrem en un cronograma no tenen per què ser els inicials, sinó que poden correspondre a un moment qualsevol del funcionament del circuit.

Veiem doncs que, en els circuits seqüencials, l'estona que el circuit estigui en cada estat i, per tant, la durada dels diferents valors dels senyals de sortida ve determinada per la sincronització.

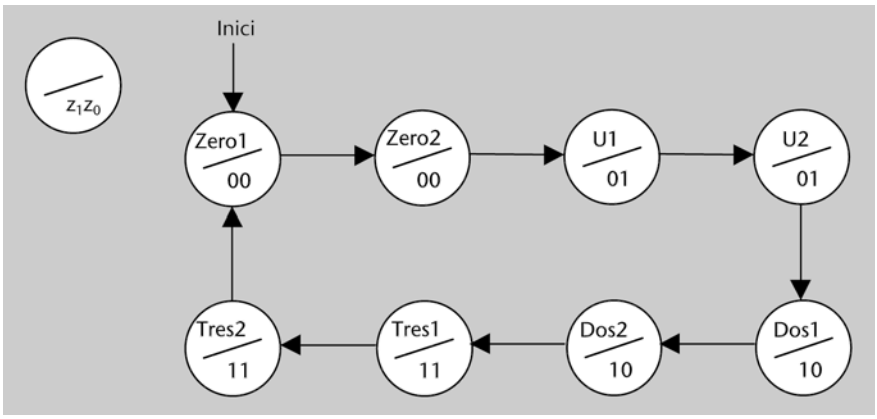
Un comptador mòdul 4

Volem dissenyar un comptador mòdul 4 com el de la figura següent en el qual cada valor de la sortida duri 100 ns:

Vegeu l'exemple dels comptadors mòdul *n* en el subapartat 4.2.3



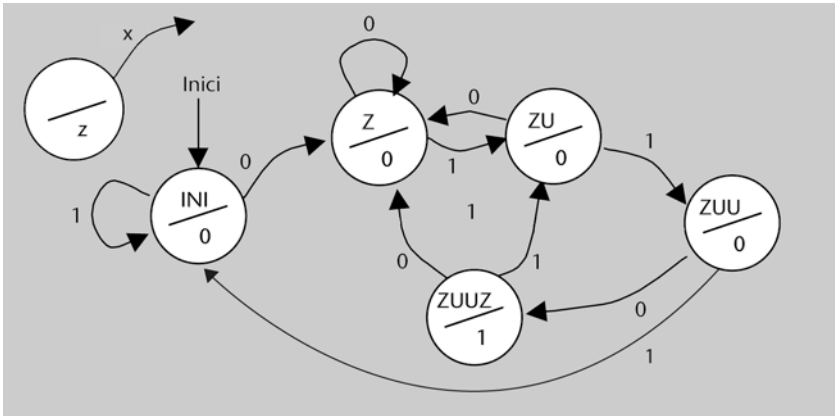
Si disposem d'un senyal de rellotge amb un període de 100 ns, llavors el graf d'estats del circuit és el que es mostra a la figura anterior. Però si el període del rellotge és de 50 ns, llavors el graf ha de ser el següent:



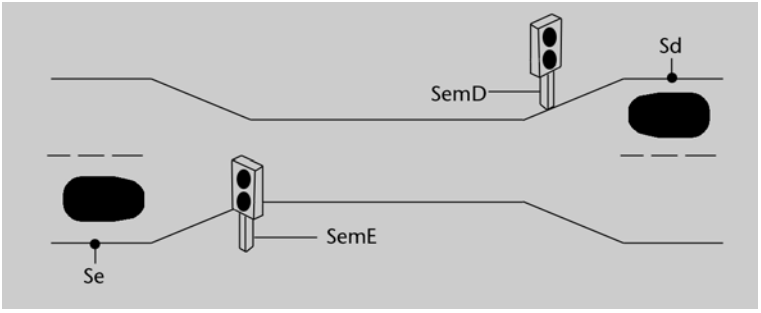
Activitats

28. Dibuixeu el graf d'estats d'un circuit que reconegui la seqüència 0110 a l'entrada *x* (d'un bit). En reconèixer-la, el senyal de sortida *z* (d'un bit) s'ha de posar a 1 durant un cicle de rellotge. Inicialment la sortida ha d'estar a 0.
El circuit no detecta encavalcament entre dues seqüències consecutives. És a dir, si arriben els valors d'entrada 0110110, la sortida només es posarà a 1 després dels quatre primers valors.

29. El graf d'estats següent correspon a un circuit que reconeix una seqüència determinada de valors en el senyal d'entrada *x* i posa la sortida *z* a 1 durant un cicle quan s'ha produït. Quina és aquesta seqüència? Descriviu amb detall els casos en què es produeix el reconeixement.



30. Dibuixeu el graf d'estats d'un circuit seqüencial que controli els semàfors d'un pont pel qual en cada moment només poden passar cotxes en un sentit.



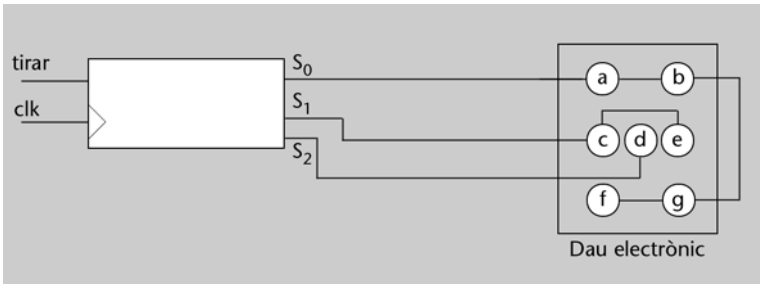
A cada extrem del pont hi ha:

- un sensor que posa el senyal *Sd* (o *Se*) a 1 quan hi ha un cotxe al davant.
- un semàfor que es posa en vermell quan li arriba un 0 pel senyal S*emD* (o *SemE*) i es posa en verd quan li arriba un 1.

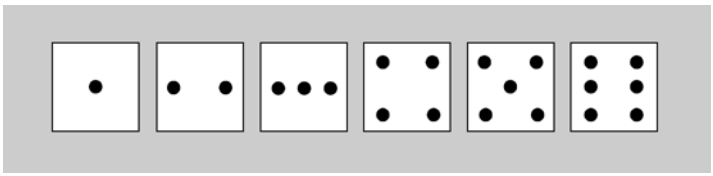
El circuit rep com a entrada els senyals *Sd* i *Se* i genera com a sortida els senyals *SemD* i *SemE*. Ha de funcionar de la manera següent:

- Els semàfors es posen en verd de forma alternada durant dos cicles de rellotge com a mínim.
- Després del segon cicle de rellotge, el semàfor que està en verd s'hi queda fins que no arribi un cotxe per l'altre extrem.
- En posar-se el sistema en funcionament, el semàfor de l'esquerra ha d'estar en verd durant dos cicles.

31. Es vol dissenyar un circuit seqüencial per a tirar un dau. L'entrada és el senyal *tirar*, que està connectat a un polsador que manipula el jugador (pitjat: *tirar* = 1; no pitjat: *tirar* = 0). Les sortides del circuit estaran connectades als punts d'un "dau electrònic" de la manera següent:



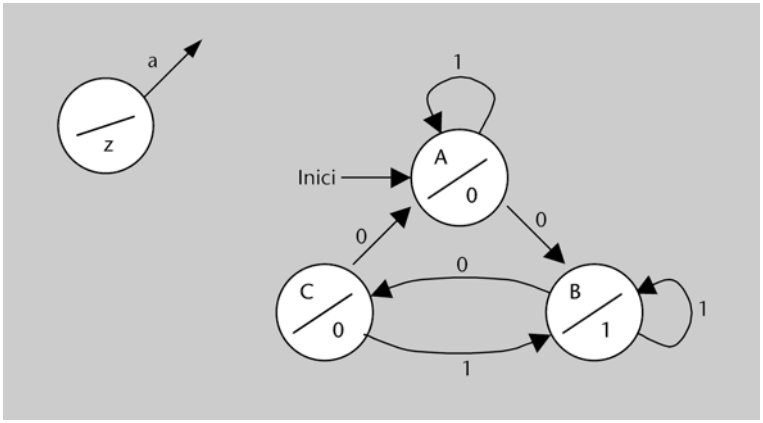
Els punts s'il·luminen quan els arriba un 1. Les combinacions possibles del dau són les següents:



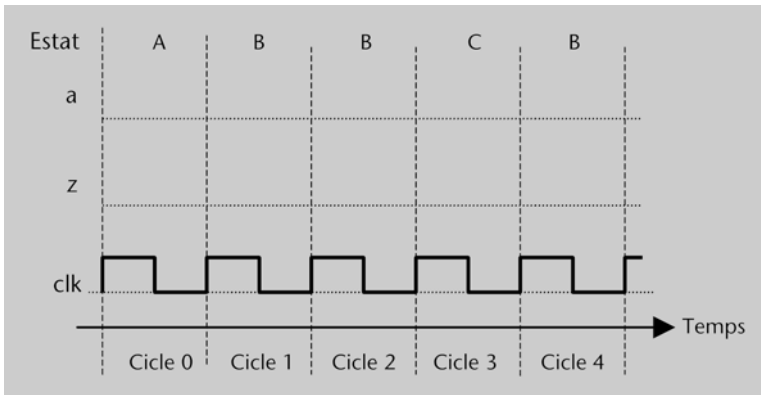
Quan el dau comenci a funcionar, ha de mostrar un "1". A partir del moment que el jugador pitgi el polsador i mentre el mantingui pitjat, el circuit generarà cíclicament i en ordre les sis combinacions del dau. Quan el jugador alliberi el polsador, no es produirà cap transició, i veurà la combinació que ha sortit al dau (se suposa que la freqüència del rellotge és molt alta i no es poden arribar a distingir les combinacions intermèdies).

Dibuixeu el graf d'estats del circuit.

32. Sigui el següent graf d'estats:



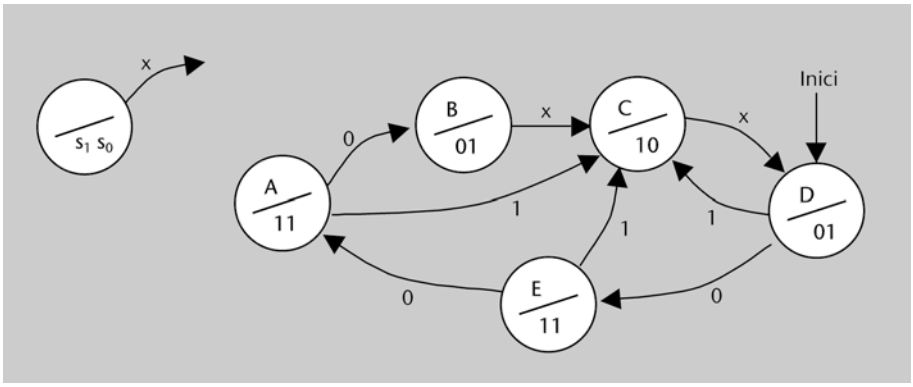
Completeu el cronograma següent. Suposeu que el senyal d'entrada *a* només canvia de valor en els instants dels flancs.



4.4. Implementació

Vegem com es pot implementar físicament un circuit seqüencial descrit d'acord amb el model de Moore. Prenem com a exemple el circuit descrit pel graf de la figura següent:

Figura 19



A continuació es mostren les taules de transicions i de sortides corresponents a aquest graf:

Taula de transicions

Estat	x	Estat$^+$
A	0	B
A	1	C
B	0	C
B	1	C
C	0	D
C	1	D
D	0	E
D	1	C
E	0	A
E	1	C

Taula de sortides

Estat	s_1	s_0
A	1	1
B	0	1
C	1	0
D	0	1
E	1	1

Aquestes dues taules es poden convertir fàcilment en taules de veritat de funcions lògiques si codifiquem els estats mitjançant variables lògiques. En con-

cret, si hi ha n estats, caldran $\lceil \log_2 n \rceil$ variables per a codificar-los. En el nostre exemple, aquesta és una possible codificació d'estats:

Estat	q_2	q_1	q_0
A	0	0	0
B	0	0	1
C	0	1	0
D	0	1	1
E	1	0	0

Una vegada codificats els estats, les transicions i sortides del circuit són funcions lògiques que es descriuen amb les taules de veritat següents:

Taula de transicions						
Estat				Estat$^+$		
q_2	q_1	q_0	x	q_2^+	q_1^+	q_0^+
0	0	0	0	0	0	1
0	0	0	1	0	1	0
0	0	1	0	0	1	0
0	0	1	1	0	1	0
0	1	0	0	0	1	1
0	1	0	1	0	1	1
0	1	1	0	1	0	0
0	1	1	1	0	1	0
1	0	0	0	0	0	0
1	0	0	1	0	1	0
1	0	1	0	x	x	x
1	0	1	1	x	x	x
1	1	0	0	x	x	x
1	1	0	1	x	x	x
1	1	1	0	x	x	x
1	1	1	1	x	x	x

Taula de sortides				
Estat				
q_2	q_1	q_0	s_1	s_0
0	0	0	1	1
0	0	1	0	1
0	1	0	1	0
0	1	1	0	1
1	0	0	1	1
1	0	1	x	x
1	1	0	x	x
1	1	1	x	x

Les variables que codifiquen els estats, q_i, es guarden en biestables. D'aquesta manera, el circuit guarda memòria en tot moment de l'estat en què es troba. En el nostre exemple, seran necessaris tres biestables; quan aquests valguin, per exemple, $[q_2\ q_1\ q_0] = [0\ 1\ 0]$, sabrem que el circuit es troba en l'estat C.

Els senyals de sortida s_1 i s_0 es poden implementar com a funcions lògiques de q_2, q_1 i q_0, a partir de la taula de veritat anterior, de qualsevol de les maneres que coneixeu.

Vegeu diverses maneres d'implementar funcions lògiques en el mòdul "Els circuits lògics combinacionals" d'aquesta assignatura.

Pel que fa a les transicions, les columnes q_2^+, q_1^+ i q_0^+ ens indiquen els valors que han de prendre els biestables en el flanc de rellotge següent. Com que un biestable pren el valor que hi ha a la seva entrada D, sabem que a les entrades dels biestables hi hem de posar, per cadascuna de les combinacions possibles

d'estats i entrades, el que es mostra a la taula següent (en la qual d_i correspon a l'entrada D de cadascun dels biestables):

q_2	q_1	q_0	x	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	0
0	0	1	1	0	1	0
0	1	0	0	0	1	1
0	1	0	1	0	1	1
0	1	1	0	1	0	0
0	1	1	1	0	1	0
1	0	0	0	0	0	0
1	0	0	1	0	1	0
1	0	1	0	x	x	x
1	0	1	1	x	x	x
1	1	0	0	x	x	x
1	1	0	1	x	x	x
1	1	1	0	x	x	x
1	1	1	1	x	x	x

Aquesta taula s'anomena **taula d'excitacions**, ja que ens diu com cal "excitar" els biestables per tal que tinguin lloc les transicions adequades. Fixem-nos que les columnes d_i de la taula d'excitacions coincideixen amb les columnes q_i^+ de la taula de transicions.

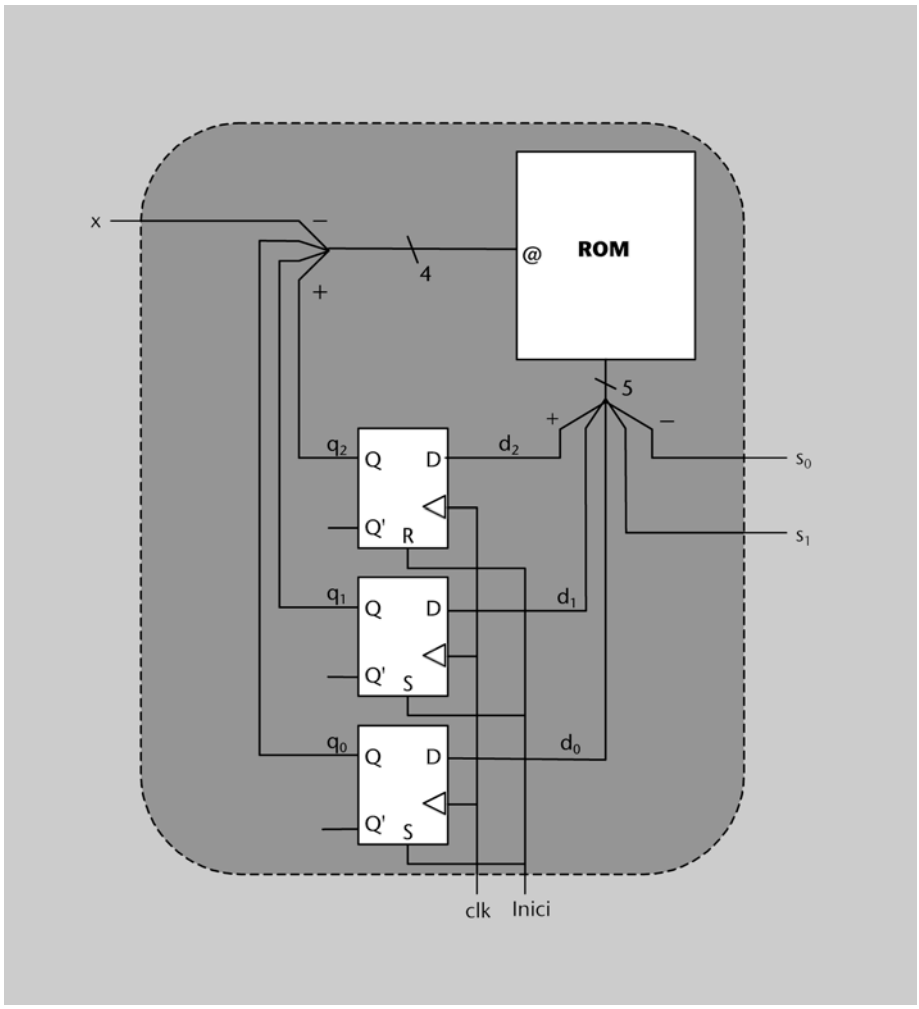
Els senyals d_2, d_1 i d_0, anomenats **funcions d'excitació**, són funcions lògiques de q_2, q_1, q_0 i de l'entrada x. Podem implementar-les, doncs, per qualsevol dels mètodes que coneixem.

Podem escriure una sola taula de veritat que inclogui les funcions d'excitació i les de sortida, tal com es mostra a la taula següent. Fixem-nos que, pel fet que les funcions de sortida depenen només de l'estat (variables q_i), el seu valor és el mateix en les files corresponents a una mateixa combinació de q_i i a diferents valors de l'entrada x.

q_2	q_1	q_0	x	d_2	d_1	d_0	s_1	s_0
0	0	0	0	0	0	1	1	1
0	0	0	1	0	1	0	1	1
0	0	1	0	0	1	0	0	1
0	0	1	1	0	1	0	0	1
0	1	0	0	0	1	1	1	0
0	1	0	1	0	1	1	1	0
0	1	1	0	1	0	0	0	1
0	1	1	1	0	1	0	0	1
1	0	0	0	0	0	0	1	1
1	0	0	1	0	1	0	1	1
1	0	1	0	x	x	x	x	x
1	0	1	1	x	x	x	x	x
1	1	0	0	x	x	x	x	x
1	1	0	1	x	x	x	x	x
1	1	1	0	x	x	x	x	x
1	1	1	1	x	x	x	x	x

Una vegada que s'ha expressat el comportament del circuit mitjançant aquesta única taula de veritat, és possible implementar-lo de manera molt senzilla fent servir una memòria ROM, tal com es mostra a la figura que presentem a continuació:

Figura 20



Nota

En aquest circuit els biestables estan dibuixats amb les entrades a la dreta i les sortides a l'esquerra.

La memòria ROM de la figura 20 es podria substituir per qualsevol altra forma d'implementació de funcions lògiques.

El contingut de la memòria ROM, que correspon a la taula anterior, és el següent:

Adreça	d_2	d_1	d_0	s_1	s_0
0	0	0	1	1	1
1	0	1	0	1	1
2	0	1	0	0	1
3	0	1	0	0	1
4	0	1	1	1	0
5	0	1	1	1	0
6	1	0	0	0	1
7	0	1	0	0	1
8	0	0	0	1	1
9	0	1	0	1	1
10	x	x	x	x	x
11	x	x	x	x	x
12	x	x	x	x	x
13	x	x	x	x	x
14	x	x	x	x	x
15	x	x	x	x	x

A la figura 20 es pot veure també com es du a terme la inicialització del circuit, connectant el senyal *Inici* a les entrades asíncrones dels biestables (recordem que el senyal *Inici* fa un pols a 1 per a indicar al circuit que es posi en funcionament). Com que en el nostre exemple l'estat inicial és el *D* (vegeu la figura 19), en començar a funcionar el circuit, els biestables prenen els valors $[q_2\ q_1\ q_0] = [0\ 1\ 1]$. El circuit tornarà a aquest estat sempre que *Inici* faci un pols.

Resum

En aquest mòdul s'han estudiat els circuits lògics seqüencials. S'ha vist que allò que els caracteritza és la capacitat de memòria, i, per tant, són capaços de determinar el valor dels senyals de sortida d'acord no solament amb el valor actual dels senyals d'entrada, sinó també del valor que han tingut aquests senyals d'entrada en moments anteriors.

S'ha vist la necessitat d'un mecanisme de sincronització per a controlar l'evolució temporal dels diferents senyals, i s'ha presentat el senyal de rellotge.

S'ha conegut el dispositiu més elemental de memòria, el biestable D, que és capaç de guardar el valor d'un bit. S'ha vist que el seu valor es pot modificar de manera síncrona i també asíncrona, gràcies a les entrades *R* i *S*. També s'ha vist que se'n pot "congelar" el valor mitjançant un senyal de càrrega.

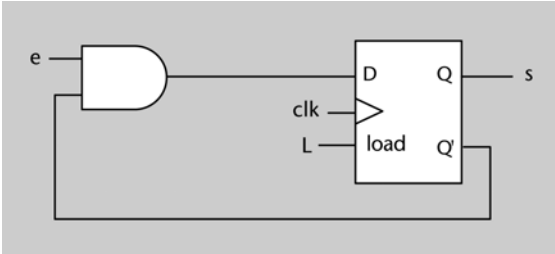
Després, s'han presentat els diferents blocs seqüencials que permeten de guardar el valor d'un mot (el registre), d'un nombre petit de mots (el banc de registres) o d'un gran volum de mots (la memòria RAM).

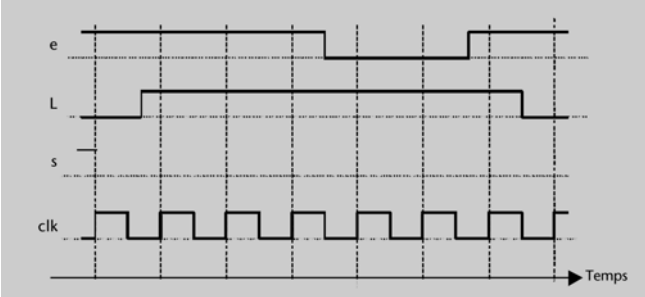
Finalment, s'ha conegut la manera d'especificar el comportament de circuits seqüencials anomenada *model de Moore*, que es fonamenta sobre els conceptes d'*estat* i *transició*. S'ha vist que el comportament del circuit es pot expressar mitjançant les taules de sortides i transicions, o bé gràficament mitjançant grafs d'estats. S'ha après a dibuixar l'evolució temporal d'un circuit sobre un cronograma.

Els blocs seqüencials i combinacionals que s'han estudiat en aquest curs constitueixen un conjunt de dispositius suficient per a dissenyar un computador senzill.

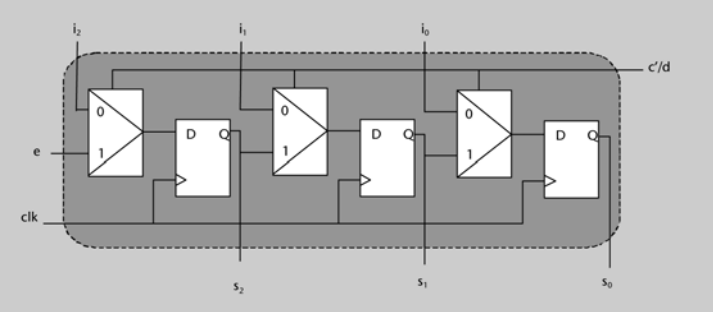
Exercicis d'autoavaluació

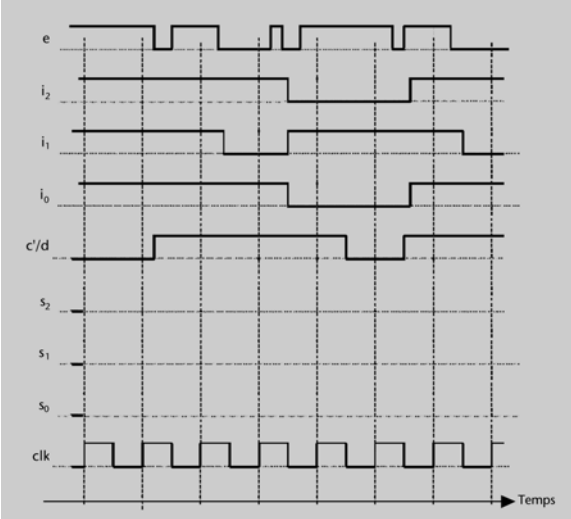
1. Completeu el cronograma que correspon al circuit de la figura, suposant que inicialment la sortida *Q* val 1. Quin és el paper del senyal *e* en el circuit?





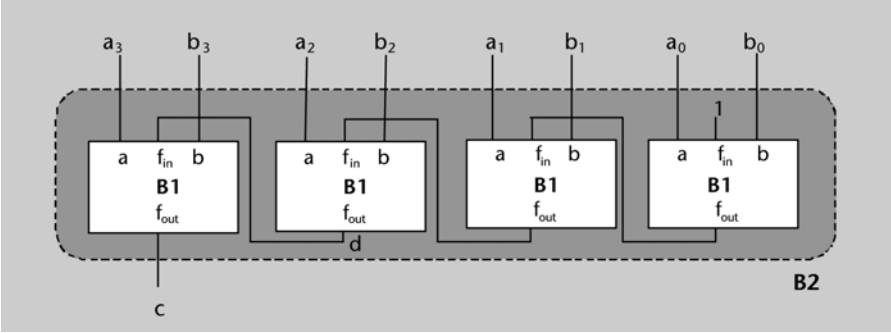
2. Completeu el cronograma que correspon al circuit de la figura, suposant que inicialment les sortides *Q* de tots els biestables valen 0. Descriviu en poques paraules què fa el circuit segons el senyal *c'/d*.





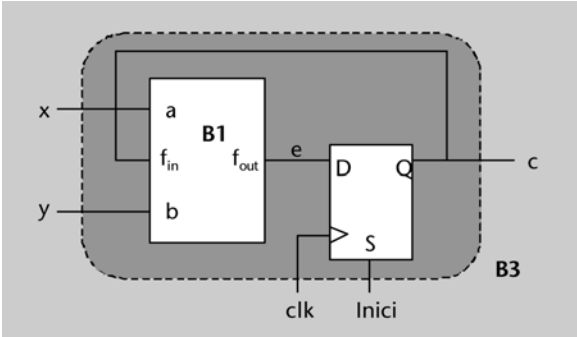
3. El circuit combinacional B2 és un comparador de dos nombres naturals *A* i *B* representats en binari. El seu funcionament és el següent:

c = 0 si *A* és menor que *B*.
c = 1 si *A* és més gran o igual que *B*.

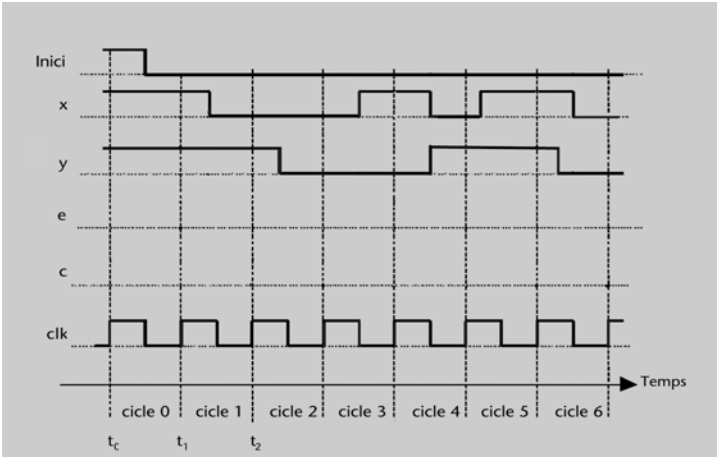


a) Veient de quina manera s'ha construït el circuit B2 i la seva funcionalitat, deduïu la taula de veritat del bloc B1.

b) Utilitzant el mateix bloc B1 s'ha construït aquest altre circuit, B3. Descriviu quina funció fa aquest circuit, i compareu-lo amb el circuit B2.

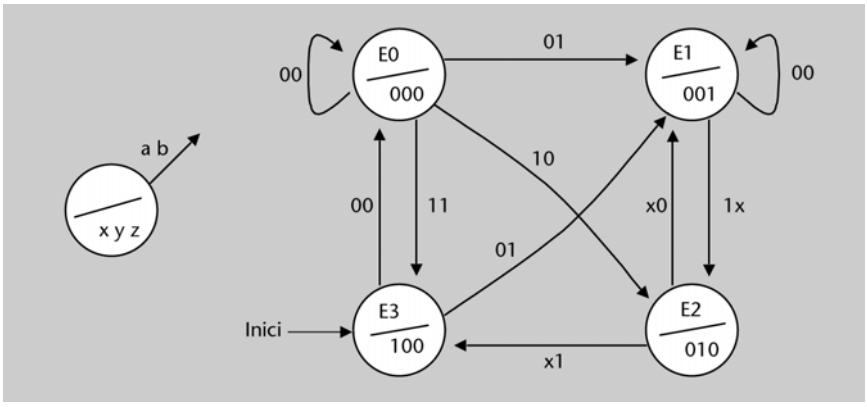


c) Completeu el cronograma següent, que correspon al circuit B3. Si interpretem les entrades *x* i *y* en cada cicle de rellotge com els diferents bits d'una parella de números *A* i *B*, quins són aquests números *A* i *B*?

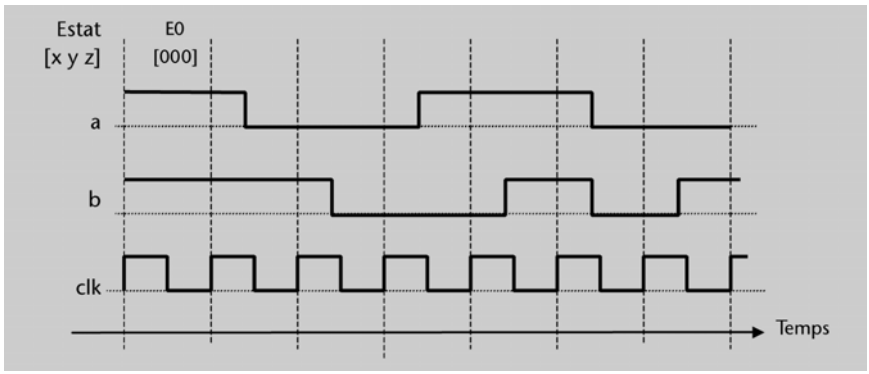


4. Es vol dissenyar un circuit seqüencial que controli un semàfor. De dia (de 8:00 a 20:00), el semàfor ha d'estar en verd durant 3 cicles, en groc durant 1 cicle i en vermell durant 2 cicles. A la nit, ha d'estar en verd durant 2 cicles, en groc durant 1 cicle i en vermell durant 3 cicles.
a) Quines entrades i sortides ha de tenir el circuit?
b) Expresseu el comportament del circuit mitjançant un graf d'estats.

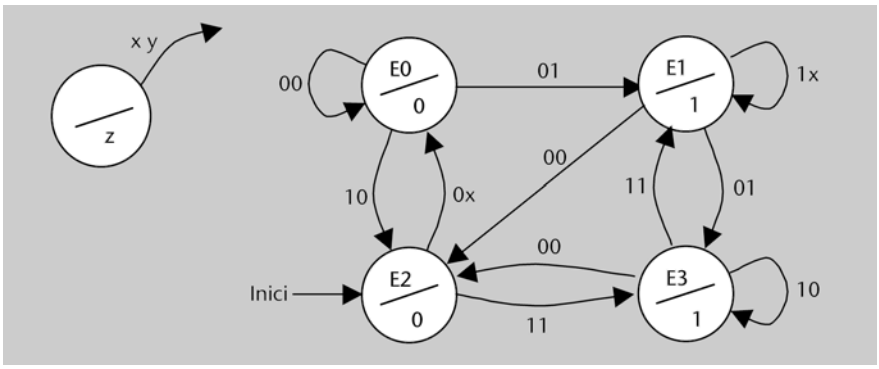
5. Donat el graf d'estats següent:

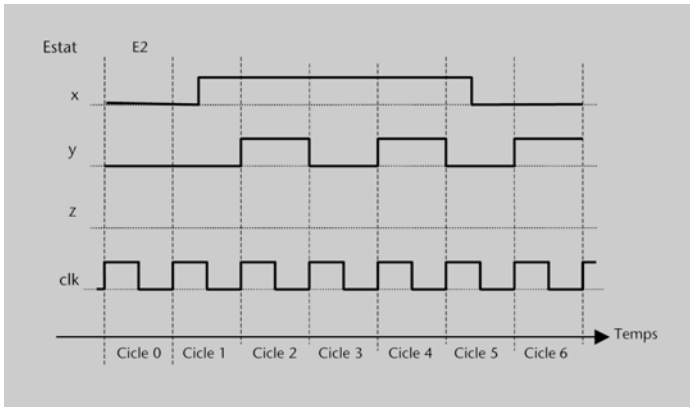


a) Escriviu la taula de sortides i la taula de transicions del circuit.
b) Completeu les línies corresponents a l'estat i a les sortides del cronograma següent:



6. Completeu el cronograma següent, que correspon a un circuit amb aquest graf d'estats:



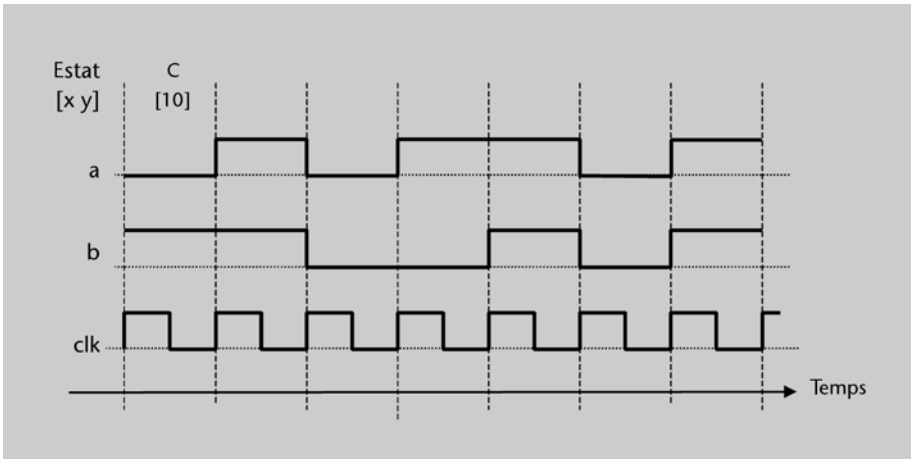


7. A continuació es mostren la taula de sortides i la de transicions d'un circuit seqüencial.

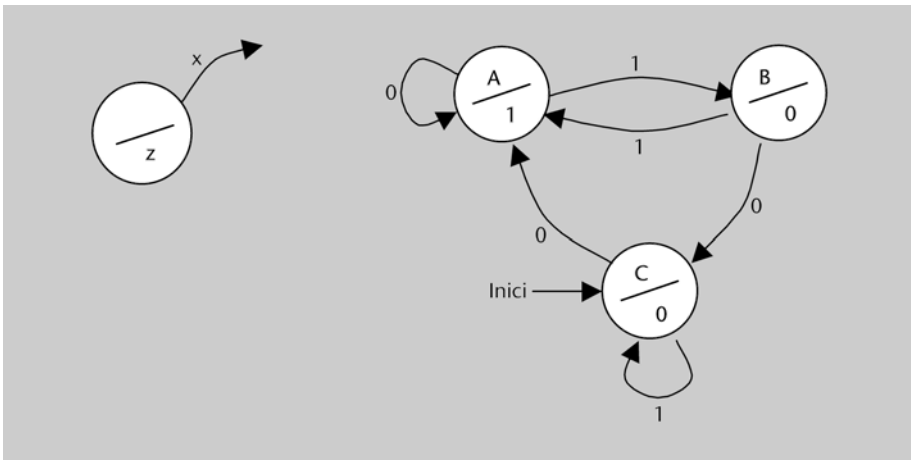
Taula de sortides	
Estat	**x y**
A	0 0
B	0 1
C	1 0

Taula de transicions			
Estat	**a**	**b**	**Estat$^+$**
A	0	0	B
A	0	1	A
A	1	0	B
A	1	1	A
B	0	0	x
B	0	1	x
B	1	0	C
B	1	1	C
C	0	0	C
C	0	1	B
C	1	0	A
C	1	1	A

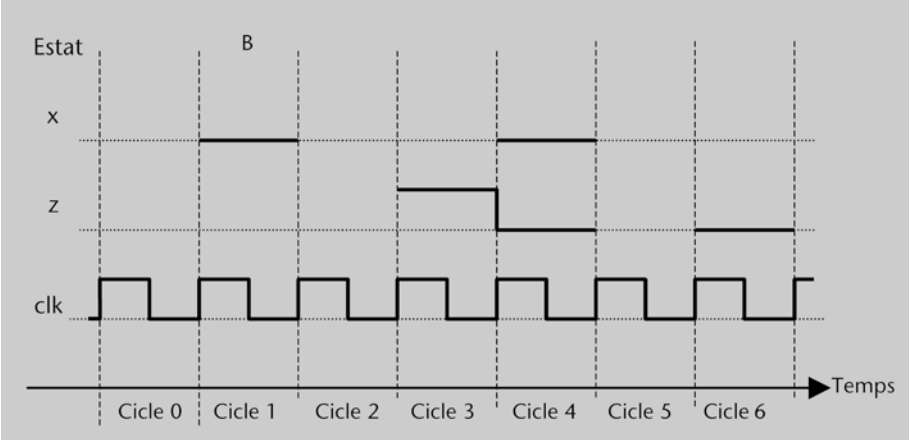
a) Dibuixeu el graf d'estats del circuit, suposant que l'estat inicial és l'*A*.
b) Completeu les línies corresponents a l'estat i a les sortides del cronograma següent:



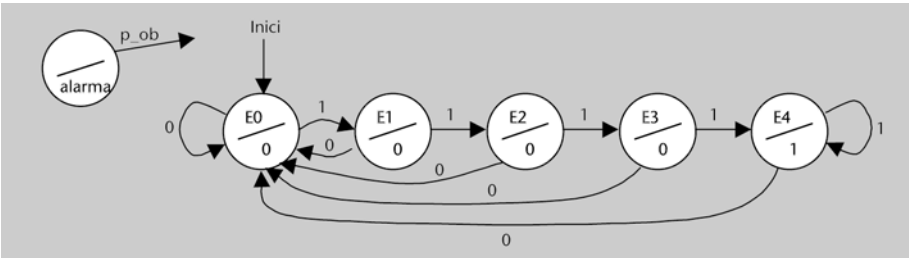
8. Sigui el graf d'estats següent:



Completeu el cronograma següent. Suposeu que el senyal d'entrada *x* només canvia de valor en els instants dels flancs.



9. Un circuit seqüencial té una entrada *p_ob* i una sortida *alarma*, totes dues d'un bit. L'entrada *p_ob* prové de la caixa forta d'un banc, i val 1 quan està oberta i 0 quan està tancada. La sortida *alarma* està connectada a un senyal d'alarma, que s'activa quan *alarma* = 1. El funcionament del circuit ve descrit per aquest graf d'estats:



Si sabem que el rellotge del circuit té un període de 30 segons, descriviu el comportament del sistema d'alarma de la caixa forta. Suposeu que la porta de la caixa només es pot obrir o tancar cada 30 segons, coincidint amb els flancs ascendents del rellotge.

Solucionari

Activitats

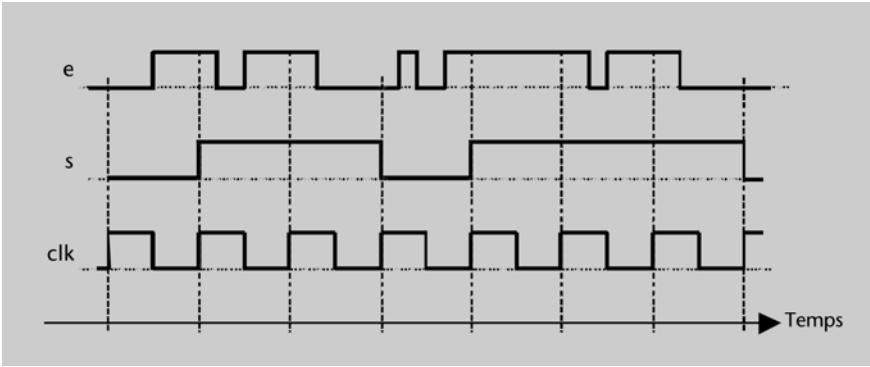
1. Es tracta de reconèixer si els quatre bits que arriben per l'entrada del sistema valen 1010 o no. Per a saber-ho n'hi ha prou d'examinar el valor del mot d'entrada en el moment actual. Per tant, el sistema és de tipus combinacional.
En canvi, si l'entrada del circuit fos d'un bit el circuit hauria de ser de tipus seqüencial, perquè en cada moment hauria de recordar els tres últims valors que han arribat per l'entrada, a més de l'actual.

2. Un dígit decimal (rang del 0 al 9) requereix quatre bits per a ser codificat. L'entrada del sistema que s'ha de 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 darrere l'altre per l'entrada del sistema, i el circuit ha de recordar els dígits que han entrat amb anterioritat per tal de reconèixer la seqüència.
Es tracta, doncs, clarament, d'un sistema seqüencial.

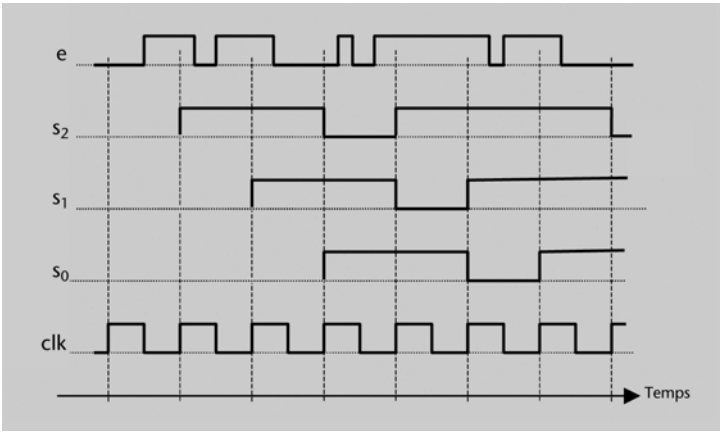
3. El període T és l'invers de la freqüència: $T = 1/F$.
Com que la freqüència és de $1{,}6 \cdot 10^9$ Hz, el període és el següent:

$$T = 1/(1{,}6 \cdot 10^9)\ \text{s} = 0{,}625 \cdot 10^{-9}\ \text{s} = 0{,}625\ \text{ns (nanosegons)}$$

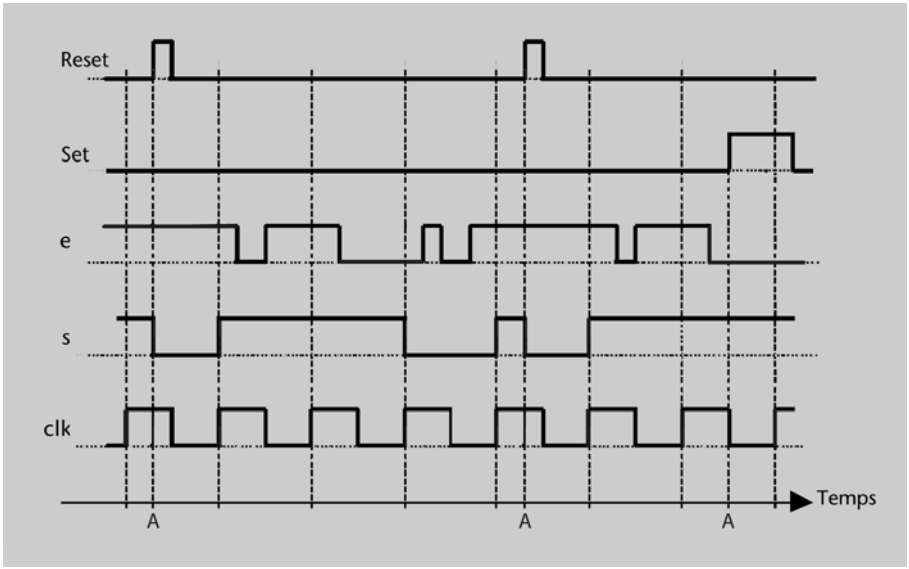
4. Com es pot veure en el cronograma, en la sortida s del biestable hi ha el valor llegit de l'entrada e en cada flanc ascendent del rellotge.
Fixeu-vos que la sortida del biestable només canvia en els flancs ascendents del rellotge (és a dir, canvia de manera síncrona), mentre que l'entrada pot canviar en qualsevol moment (és a dir, canvia de manera asíncrona).
És important notar que el valor de 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. Per tant, com es pot veure en la figura, s no té cap valor.



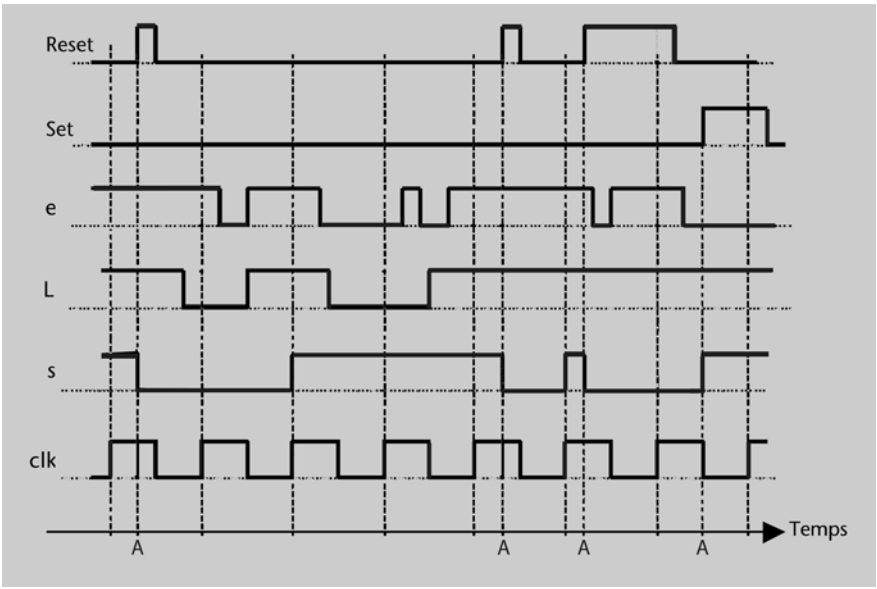
5. En aquest cas es pot veure que l'entrada e es "desplaça" successivament per les sortides s_2, s_1 i s_0 dels biestables.
La millor manera de fer aquest tipus de cronogrames és dibuixar primer la línia corresponent al senyal s_2, que només depèn de e, sencera. Una vegada aquesta s'ha dibuixat, es dibuixa 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 de s_1 no es pot determinar fins al segon flanc de rellotge, i el de s_0 fins al tercer.



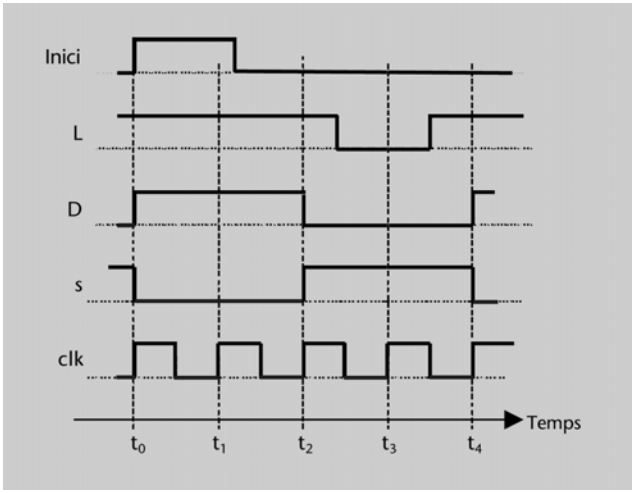
6. Per a resoldre aquest exercici hem de tenir en compte que les entrades asíncrones tenen prioritat sobre les síncrones, és a dir, primer s'avaluen els valors de les entrades *R* i *S* del biestable, i només quan ambdues estan a 0 s'avalua el valor present a l'entrada *D* en cada flanc de rellotge (com que el biestable no té senyal de càrrega, assumim que està a 1).
Per aquesta raó, en el cronograma estan marcats amb línies verticals amb la lletra A els moments en què canvien les entrades asíncrones, a més dels flancs ascendents del rellotge.



7. 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.
Per contra, quan $L = 1$, s'emmagatzema en el 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 es deu al fet que *Set* està a 1.
Com en l'activitat 6, en el cronograma hi ha marcats els moments en què canvien les entrades asíncrones amb la lletra A a més dels flancs de pujada del rellotge.

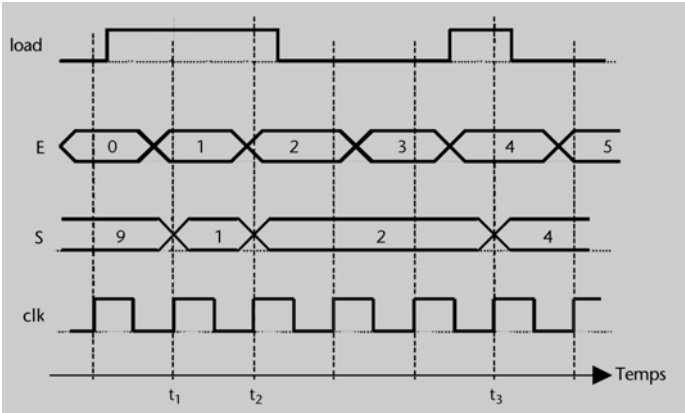


8. Inicialment, $D = 0$, perquè *s* està a 1. En l'instant t_0 *Inici* es posa a 1, i, per tant, el biestable es posa a 0. S'hi quedarà almenys fins a t_2, perquè a t_1 *Inici* continua valent 1. A t_2 es carrega amb el que hi ha a l'entrada *D*, és a dir, un 1. A t_3, el biestable no es carrega, perquè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



9. Podem observar en el cronograma que el registre només es carrega en els flancs en els quals *load* = 1: t_1, t_2 i t_3. 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 en el cronograma, és la següent:

9 1 2 4.



10.
a) En aquest circuit, la sortida del sumador es connecta directament a una de les seves entrades. Això implica que *S* mai serà estable, contínuament estarà variant.

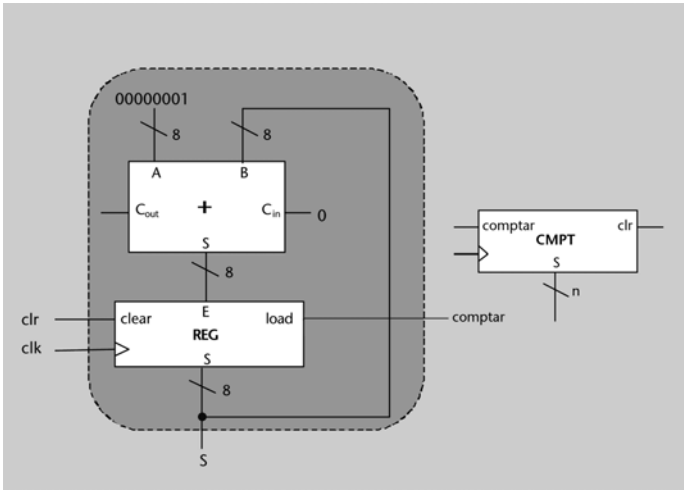
b) En aquest cas, *S* expressa correctament la suma acumulada dels valors que ha pres *X* (en els instants anteriors a cada flanc, tal com indica la convenció que s'ha establert en aquests apunts) des de la inicialització del circuit. Tanmateix, en el moment en què aquesta suma acumulada sigui més gran que 255 el valor de *S* deixarà de ser correcte, perquè no n'hi ha prou amb 8 bits per a representar-lo. Podríem pensar a fer que el registre sigui de més de 8 bits, però sigui quin sigui el nombre de bits que tingui sempre arribarà un moment que es produirà sobreeiximent (tret que *Inici* faci un nou pols a 1 abans d'arribar a produir-se el sobreeiximent, però que això passi o no és incontrolable).

11. Usarem un registre que contindrà en tot moment la sortida *S* del circuit. Sempre que *comptar* valgui 1, el seu 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 en el proper flanc de rellotge si *comptar* val 1. Per tant, connectarem *comptar* a l'entrada *load* del registre.

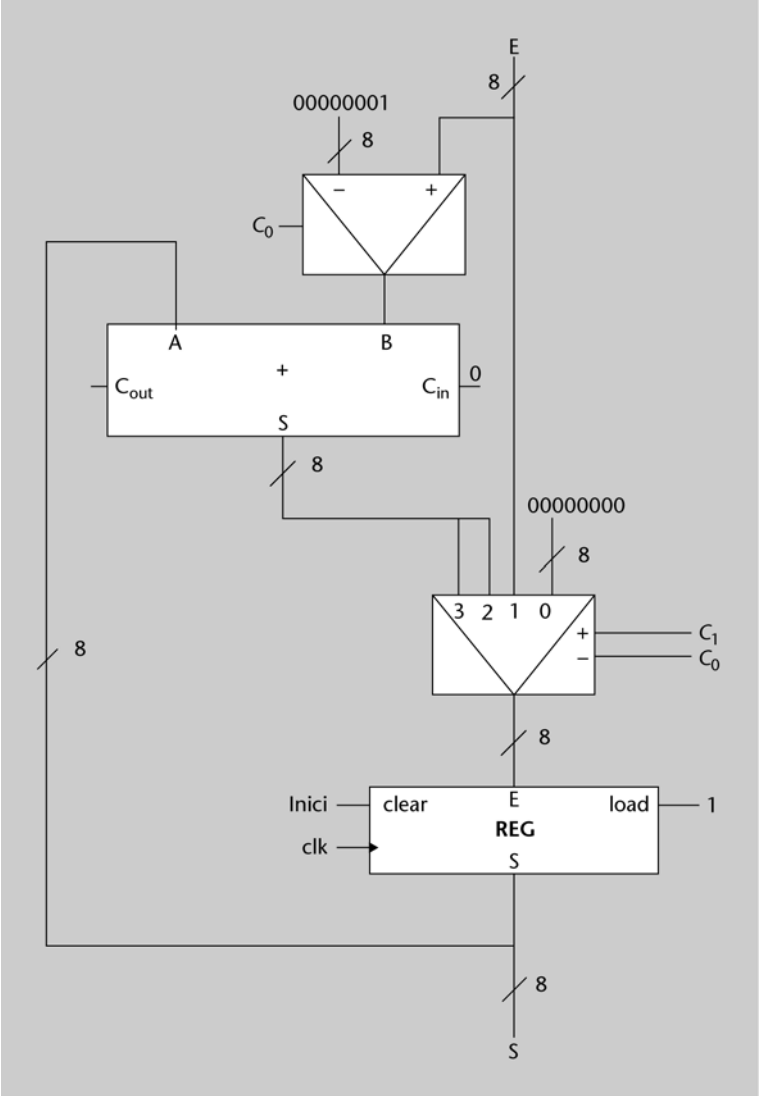
Una altra possibilitat és posar un 0 a l'altra entrada del sumador i sumar l'1 per mitjà de l'entrada de transport del sumador, que en el disseny mostrat té un 0.

D'altra banda, el senyal *clr* estarà connectat a l'entrada *clear* del registre per tal de posar-lo a 0 de manera asíncrona.

La figura mostra el circuit que conforma el bloc, i el dibuix del bloc amb les seves entrades i sortides.



12. El valor de *S* s'ha d'actualitzar a cada cicle, per tant ha d'estar guardat en un registre, a l'entrada *clear* del qual connectem el senyal *Inici* i a l'entrada *load* del qual connectem un 1. Per tal de determinar quin valor s'hi carregarà a cada flanc ascendent usarem un multiplexor 4-1 governat pels senyals de control c_1 i c_0. Quan $c_1 = 1$, el valor que s'ha de carregar al registre és el seu contingut actual sumat amb 1 (si $c_0 = 0$) o *E* (si $c_0 = 1$). Això ho aconseguirem amb un sumador, a una entrada del qual connectem *S* i a l'altra connectem la sortida d'un multiplexor 2-1 governat per c_0.



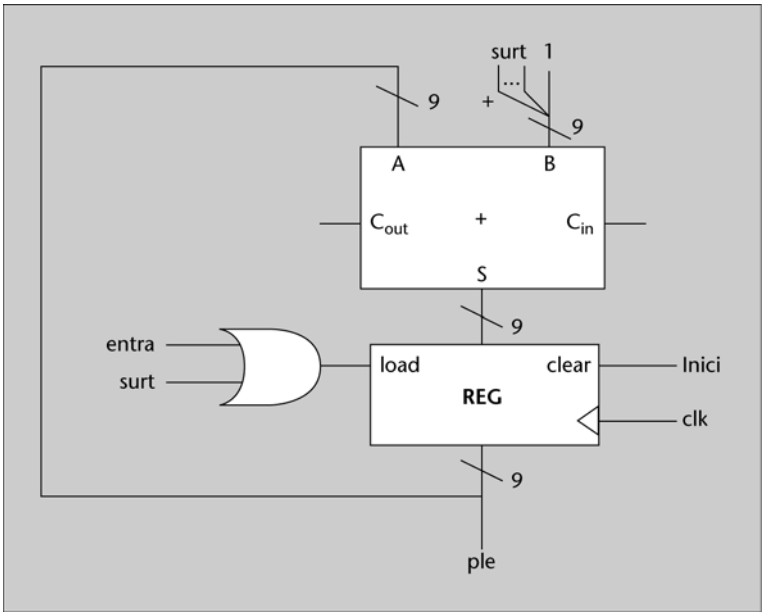
13.

a) El circuit ha de saber quan entra o surt un cotxe, per tant *entra* i *surt* han de ser senyals d'entrada, i també el senyal *Inici*. I la seva missió és donar en tot moment el valor correcte al senyal *ple*, que serà el seu senyal de sortida. Tots els senyals són d'1 bit.

b) El circuit ha de ser seqüencial perquè ha de saber en tot moment quants cotxes hi ha al pàrquing, per tal de poder determinar si ja està ple o no. Per tant, ha de tenir memòria de quants n'han entrat i sortit des de la inicialització i fins al moment actual.

c) El circuit ha de tenir un registre que guardi el nombre de vehicles que hi ha al pàrquing. Com que n'hi pot haver 500 com a molt, el registre ha de tenir 9 bits. Connectarem *Inici* a la seva entrada *clear*.

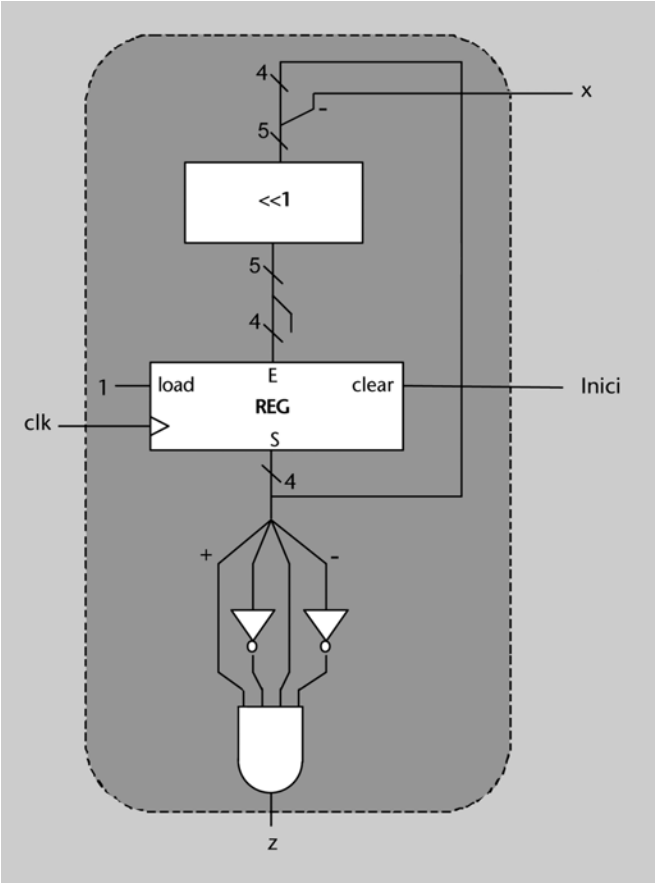
Aquest registre s'ha d'incrementar o decrementar en una unitat sempre que es produeixi un pols al senyal *entra* o *surt*, respectivament. Per tant, l'entrada *load* ha d'estar a 1 quan algun d'aquests dos senyals estigui a 1. Caldrà disposar d'un sumador que sumi el contingut del registre més 1 o –1 segons el cas. La suma en binari i en complement a 2 es fan de la mateixa manera (és a dir, el resultat d'una suma serà correcte tant si interpretem les entrades i sortides del sumador en binari com en complement a 2). Per tant, en cas que vulguem sumar 1 podem posar a l'entrada del sumador 000000001, i en cas que vulguem sumar –1 hi podem posar 111111111 (que és –1 codificat en complement a 2). Tenint en compte que sempre que *surt* valgui 1 *entra* valdrà 0, podem aconseguir el valor desitjat en l'entrada *B* del sumador tal com es mostra en el circuit: el bit de menys pes val sempre 1 (recordem que el registre només es carregarà si *entra* o *surt* valen 1), i els altres 8 bits es formen replicant 8 vegades el senyal *surt*.



14. Volem reconèixer si a l'entrada *x* d'un bit es produeix la seqüència de valors 1010, i disposem d'un registre de quatre bits. Per tant, hem de fer que els valors que arribin per l'entrada *x* es desplacin pels biestables del registre, per exemple, de dreta a esquerra. El biestable que conté el bit de menys pes es carregarà a cada flanc amb el valor de *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 de *x*.
D'aquesta manera, obtenim que la sortida *z* ha de valer 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 el següent:

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

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 de *x* fos 0, el circuit reconeixeria la seqüència 1010 després del primer flanc de rellotge, de manera errònia. Per això connectem el senyal *Inici* a l'entrada *clear* del registre.



15. 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 nombres estan representats en complement a 2, el bit R_{n-1} és el bit de signe de l'últim nombre 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 nombre que hi ha a *REG1* si aquest és positiu, i deixa passar un 0 si és negatiu.
- El sumador suma aquest nombre amb el contingut del registre *REG2*, que inicialment està a 0, i el resultat es guarda una altra vegada 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 nombres positius que entren per *X*.

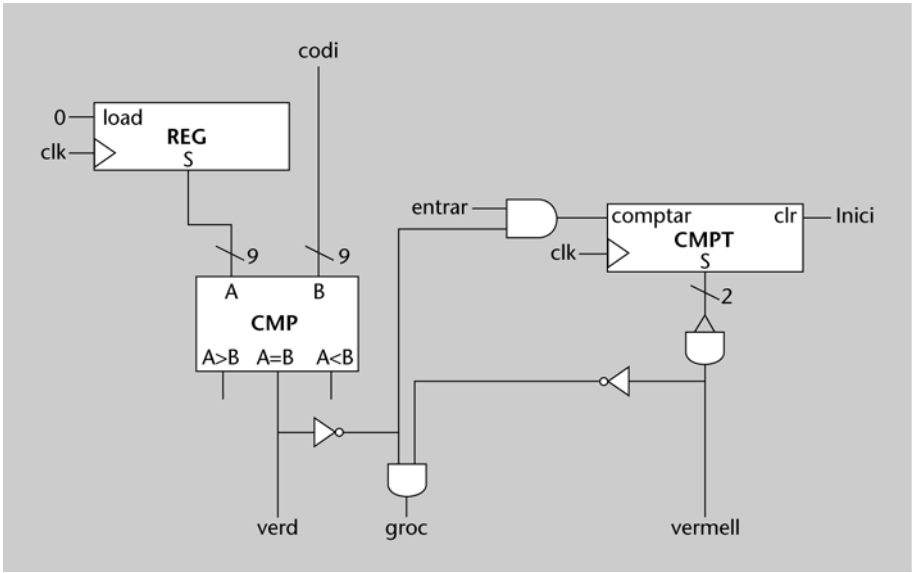
16. Veiem que el senyal *Inici* està connectat a totes les entrades *clear* dels registres. Per tant, tots es posen a 0 quan comença a funcionar el circuit.
Les entrades *load* de cada registre estan connectades a les sortides d'un descodificador. Per tant, només una d'aquestes està a 1 en cada moment, 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 registre 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 quatre registres. Per tant, deduïm que c_3 i c_2 controlen quin dels continguts dels quatre registres surt en cada moment per la sortida.

17.
a) Les entrades són:

- *entrar*, d'un bit.
- *codi*, de 9 bits, perquè el màxim valor que pot tenir és 444.
- *Inici*, d'1 bit.

I les sortides són *verd*, *groc* i *vermell*, totes tres d'1 bit.

b) El circuit ha de tenir un comparador per comparar el codi que s'ha teclejat amb la contrasenya correcta. Quan la sortida A = B valgui 1, s'ha d'activar el senyal *verd*, mentre que si val 0 s'han d'activar o bé *groc* o bé *vermell* depenent de quantes vegades s'ha entrat un codi incorrecte.
Per a saber quantes vegades s'ha escrit un codi incorrecte usarem un bloc comptador com el que es dissenya a l'activitat 11; com que només ha de comptar fins a 3, pot tenir només 2 bits. S'haurà d'incrementar en 1 sempre que la sortida $A = B$ del comparador valgui 0 i s'hagi entrat un nou codi; això ho aconseguim connectant a l'entrada *comptar* del comptador la sortida d'una porta AND, a les entrades de la qual connectem *entrar* i la negació de la sortida $A = B$ del comparador. Fixem-nos que, com que *entrar* val 1 només durant un cicle de rellotge, el comptador s'incrementarà només un cop per cada nou codi que s'escrigui. El senyal *vermell* s'haurà d'activar quan la sortida del comptador sigui 11, i quan això passi el senyal *groc* ha d'estar a 0. Per tant, el llum groc s'ha d'encendre quan el codi escrit no coincideixi amb la contrasenya correcta però el comptador encara no hagi arribat a 3.
La figura mostra el circuit sencer. El registre de l'esquerra és el que conté la contrasenya correcta.

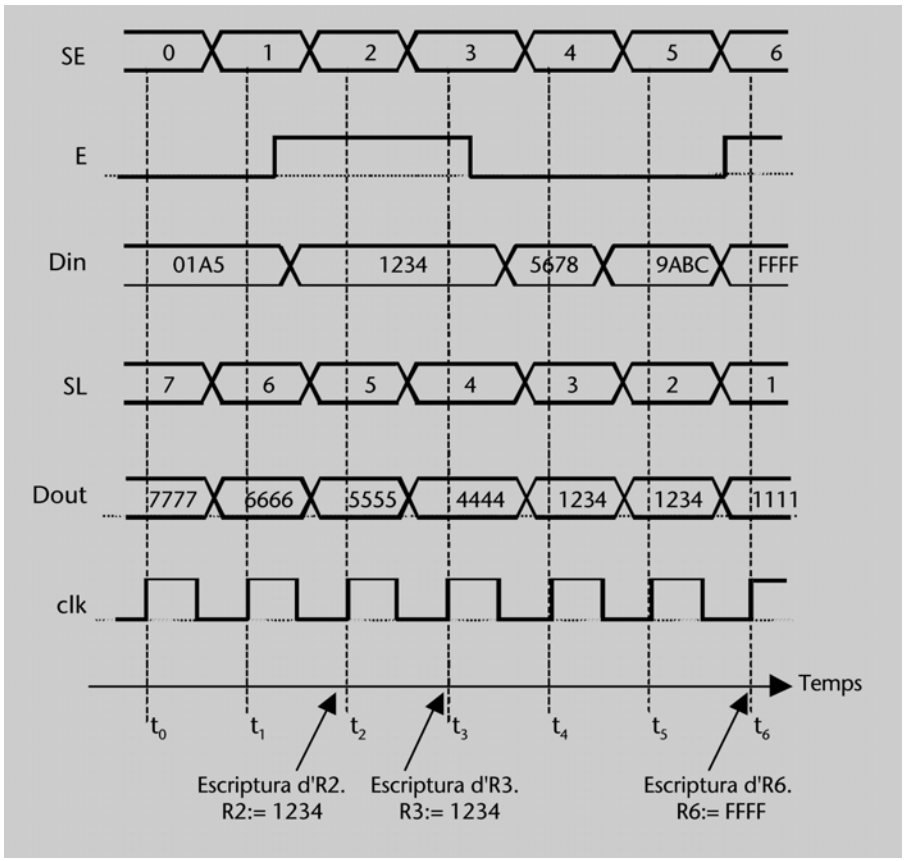


c) Caldria definir un senyal *canvi_contrasenya* que s'activaria quan es volgués fer el canvi i que s'hauria de connectar a l'entrada *load* del registre que guarda la contrasenya correcta. També caldria establir alguna manera d'escriure la nova contrasenya, i fer arribar el seu valor codificat en binari a l'entrada de dades del registre.

18.
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 a saber quin registre s'escriu analitzem el valor d'*SE* en aquests instants; per a 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`.

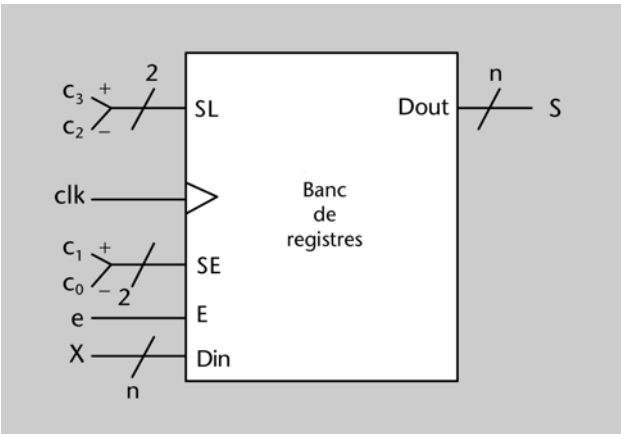
b) En el cronograma es pot veure la seqüència de valors a *Dout*. Per a deduir-lo, cal mirar en cada moment quin registre es llegeix (*SL*) i quin és el seu contingut (recordem que les lectures es fan de manera asíncrona).



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.

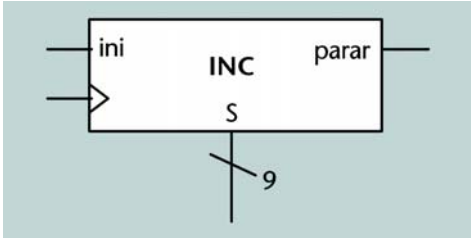
19. La funcionalitat del circuit de l'activitat 16 (sense l'entrada *Inici*) s'aconsegueix amb un banc de registres que connecta els senyals a les entrades i sortides tal com es mostra en la figura.



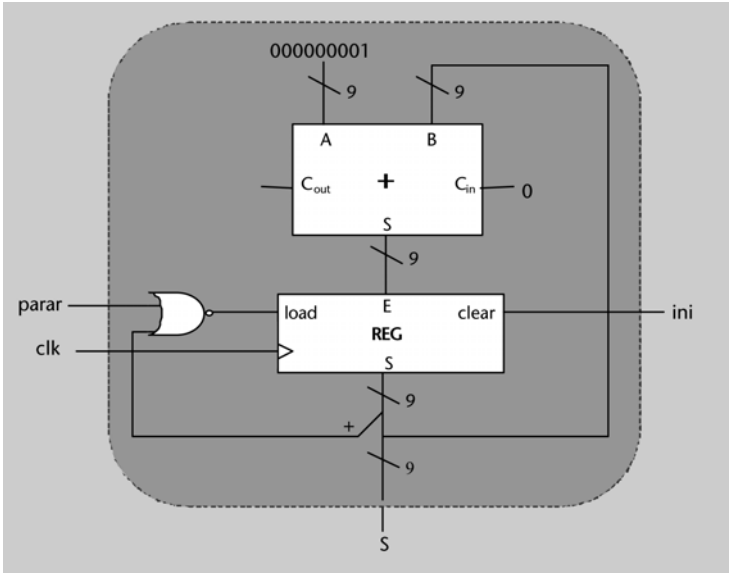
Deduïm, doncs, que el circuit de l'activitat 16 correspon a una possible implementació d'un banc de quatre registres, al qual se li ha afegit la funcionalitat d'inicialitzar tots els registres a 0.

20.

a) El circuit que s'ha de dissenyar és el següent:



- Per a fer aquest bloc utilitzarem, bàsicament, un registre i un sumador. El registre emmagatzemarà la sortida *S* i el sumador permetrà d'incrementar-la. Per a fer això, a una entrada del sumador hi arribarà el valor de *S* i a l'altra un 1.
- Implementarem el senyal *ini* amb el *clear* del registre.
- Implementarem el senyal *parar* amb el senyal de *load* del registre. Quan *parar* sigui 1 o el bit de més pes de *S* sigui 1 posarem un 0 a *load*. En qualsevol altre cas, el senyal de *load* serà 1. Per tant, $load = (parar + S_8)'$

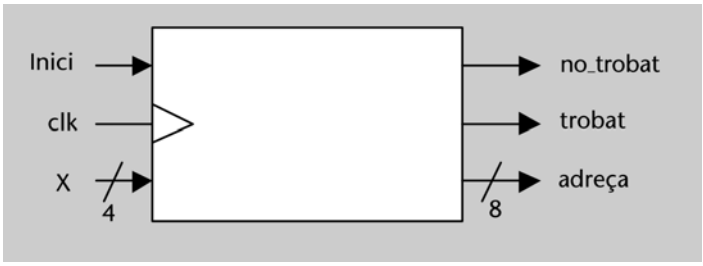


b) El circuit té tres entrades i tres sortides.

Entrades:
- Una entrada (*Inici*) d'un bit, que inicialitza el circuit.
- Una entrada de rellotge: *clk*.
- Una entrada de dades de quatre bits: *X*.

Sortides:
- Una sortida de vuit bits: *adreça*.
- Dues sortides d'un bit: *trobat*, *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 a veure què fa el circuit l'analitzarem per parts:
- Quan el circuit comença a funcionar, l'entrada *X* s'emmagatzema en el 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 (100000000) pararà 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 vuit bits de menys pes d'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 que és 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 allò que fa aquest circuit és buscar en 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.

21.
a) El cotxe pot estar en quatre situacions: parat, girant cap a la dreta, girant cap a l'esquerra i movent-se endavant. El circuit tindrà, doncs, quatre estats, que anomenarem respectivament *PARAT*, *DRETA*, *ESQUERRA* i *ENDAVANT*.
b) El més habitual és que, en pitjar el botó "on" del cotxe per a començar-hi a jugar, aquest resti parat. Per tant, podem dir que l'estat inicial és *PARAT*.
c) A partir de la taula que descriu l'acció dels senyals z_1 i z_0 sobre el cotxe, obtenim que els senyals de sortida han de tenir els valors següents:

Estat	z_1	z_0
PARAT	1	0
DRETA	0	0
ESQUERRA	0	1
ENDAVANT	1	1

L'enunciat ens diu que les transicions que es produiran seran les següents:

Estat	e	d	Estat$^+$
PARAT	0	0	PARAT
PARAT	0	1	DRETA
PARAT	1	0	ESQUERRA
PARAT	1	1	ENDAVANT
DRETA	0	0	DRETA
DRETA	0	1	DRETA
DRETA	1	0	ENDAVANT
DRETA	1	1	PARAT
ESQUERRA	0	0	ESQUERRA
ESQUERRA	0	1	ENDAVANT
ESQUERRA	1	0	ESQUERRA
ESQUERRA	1	1	PARAT
ENDAVANT	0	0	ENDAVANT
ENDAVANT	0	1	DRETA
ENDAVANT	1	0	ESQUERRA
ENDAVANT	1	1	PARAT

22.
a) El circuit necessita saber si els valors de *x* i *y* han estat iguals en almenys tres ocasions. Per a això, ha de ser capaç de recordar aquestes situacions:
- Els valors de *x* i *y* no han estat iguals en cap ocasió.
- Ho han estat en una ocasió.
- Ho han estat en dues ocasions.
- Ho han estat en tres ocasions o més.

Aquests seran els estats del circuit, que anomenarem respectivament *CAP*, *UNA*, *DUES* i *TRES_O_MÉS*.

b) Quan el circuit es posi en funcionament, com que encara no haurà arribat cap valor per les entrades, haurà de trobar-se en l'estat *CAP*.

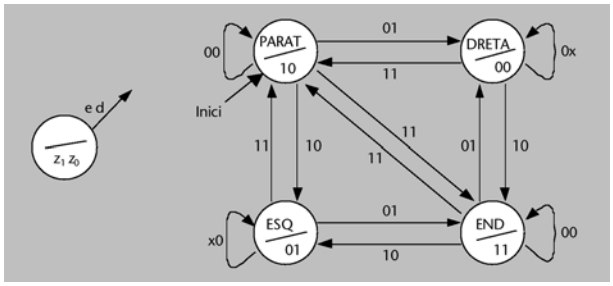
c) La sortida només ha de valer 1 quan *x* i *y* hagin estat iguals almenys 3 vegades i, per tant, la taula de sortides és la següent:

Estat	z
CAP	0
UNA	0
DUES	0
TRES_O_MÉS	1

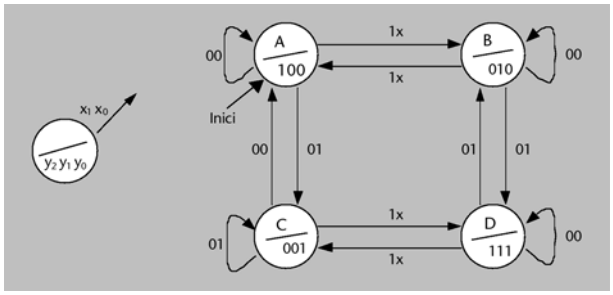
Sempre que *x* i *y* siguin diferents, el circuit romandrà en l'estat en què es trobi. Quan siguin iguals, passarà a recordar que han estat iguals en una ocasió més. Una vegada hagi arribat a l'estat *TRES_O_MÉS*, ja no en sortirà. Per tant, la taula de transicions és aquesta:

Estat	x	y	Estat$^+$
CAP	0	0	UNA
CAP	0	1	CAP
CAP	1	0	CAP
CAP	1	1	UNA
UNA	0	0	DUES
UNA	0	1	UNA
UNA	1	0	UNA
UNA	1	1	DUES
DUES	0	0	TRES_O_MÉS
DUES	0	1	DUES
DUES	1	0	DUES
DUES	1	1	TRES_O_MÉS
TRES_O_MÉS	x	x	TRES_O_MÉS

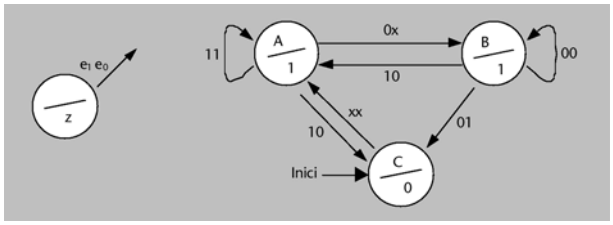
23.
Anomenem "ESQ", l'estat *ESQUERRA* i "END", l'estat *ENDAVANT*. Recordem que l'estat inicial és *PARAT*; aleshores traduïm directament de les taules de transicions i de sortides que hem obtingut a l'activitat 21 i podem dibuixar el graf següent:



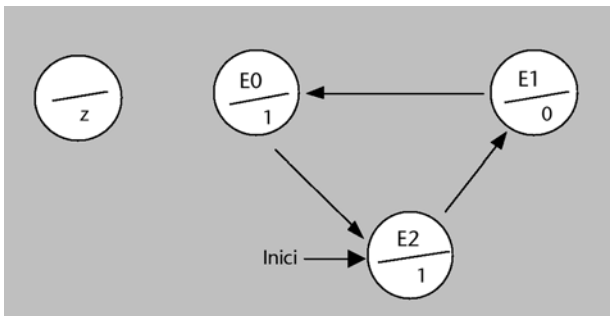
24.
Si traduïm directament de les taules de transicions i de sortides, podem dibuixar els grafs següents:
a)



b) En aquest cas, estant en els estats *B* o *C*, no es produirà mai la combinació d'entrada [1 1]. D'altra banda, de l'estat *C* passem sempre a l'estat *A*.



c) Fixem-nos que aquest circuit no té cap senyal d'entrada.



25.
a) Veiem que es tracta d'un circuit sense cap senyal d'entrada. Fixem-nos que a la taula de sortides hem escrit els senyals en un ordre diferent del que apareix a la llegenda del graf (els podem escriure en qualsevol ordre, sempre que mantinguem la coherència).

Taula de transicions	
Estat	**Estat⁺**
A	B
B	C
C	A

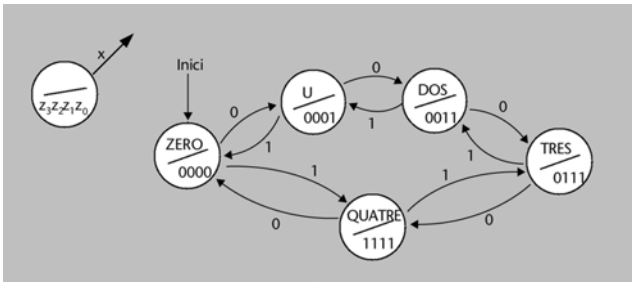
Taula de sortides			
Estat	**x**	**y**	**z**
A	1	0	0
B	0	1	0
C	0	0	1

b) Si mirem el graf, veiem que, estant en l'estat *B*, l'entrada no val mai 1. No obstant això, incloem a la taula de transicions la fila corresponent a aquesta combinació.

Taula de transicions		
Estat	**a**	**Estat⁺**
A	0	B
A	1	A
B	0	C
B	1	x
C	0	A
C	1	A

Taula de sortides		
Estat	**p**	**q**
A	0	0
B	1	0
C	0	1

26.
El graf d'estats és el següent:

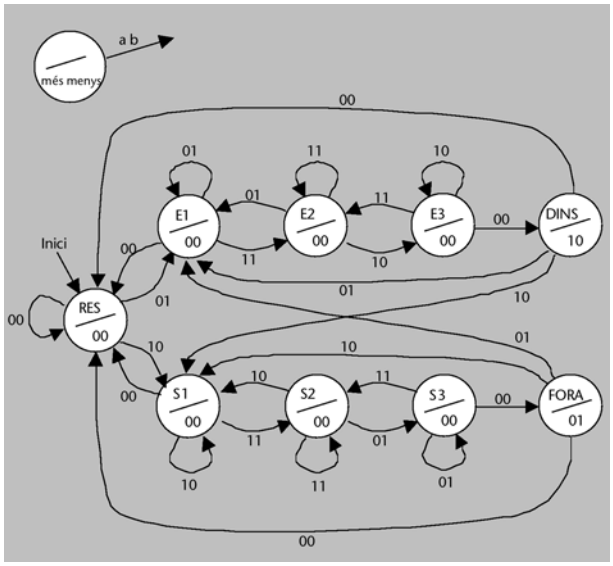


27.
Per a entrar a l'aparcament, un cotxe ha de passar quatre etapes: primer passarà per davant del sensor *B*, després serà davant de tots dos sensors, després, només davant del sensor *A* i, finalment, entrarà a l'aparcament. Per a sortir-ne també haurà de passar per quatre etapes anàlogues, però en sentit contrari. En conseqüència, per a poder determinar el valor dels senyals *més* i *menys*, el circuit ha de ser capaç de distingir les situacions que es mostren a la taula següent. S'hi inclou també la taula de sortides del circuit; veiem que el senyal *més* només s'ha de posar a 1 quan un cotxe ja ha entrat (estat *DINS*), i el senyal *menys* només s'ha de posar a 1 quan un cotxe ja ha sortit (estat *FORA*). L'estat inicial és *RES*.

Estat	Nom	més	menys
No hi ha cap moviment	RES	0	0
Entra un cotxe, etapa 1	E1	0	0
Entra un cotxe, etapa 2	E2	0	0
Entra un cotxe, etapa 3	E3	0	0
Ha entrat un cotxe completament	DINS	1	0
Surt un cotxe, etapa 1	S1	0	0
Surt un cotxe, etapa 2	S2	0	0
Surt un cotxe, etapa 3	S3	0	0
Ha sortit un cotxe completament	FORA	0	1

Les transicions entre estats es mostren al graf següent. Veiem que hi ha estats en els quals no es donaran algunes combinacions d'entrada. Per exemple, en l'estat E1 (hi ha un cotxe que entra davant del sensor *B*) no es donarà mai la combinació [*a b*] = [1 0] (perquè l'enunciat ens diu que mai no es trobaran dos cotxes de cara).

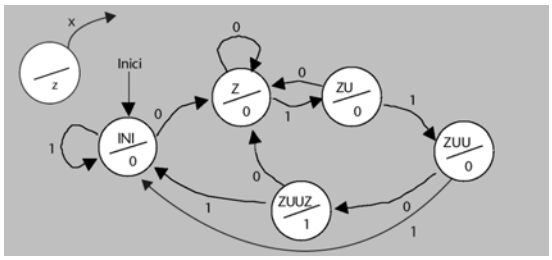
Recordem que un cotxe pot parar o fer marxa enrere en qualsevol moment. Per això, per exemple, si en l'estat *E2* (el cotxe és davant de tots dos sensors) es produeix la combinació [*a b*] = [0 1], anem a l'estat *E1* (el cotxe ha fet marxa enrere fins a situar-se davant del sensor *B*). Si es produeix la combinació [*a b*] = [1 1], ens quedem a l'estat *E2* (el cotxe continua davant dels dos sensors).



28.
a) El graf d'estats es mostra a continuació. Hem fet servir els mnemotècnics següents per als noms dels estats:

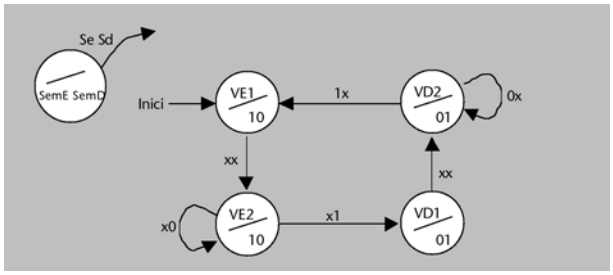
Estat	nom
Estat incial	INI
Ha arribat un 0	Z
Ha arribat la subseqüència 01	ZU
Ha arribat la subseqüència 011	ZUU
Ha arribat la seqüència 0110	ZUUZ

Fixem-nos que de l'estat *ZUUZ* es passa sempre a un estat en què la sortida val 0 i, per tant, quan es reconeix la seqüència, la sortida està a 1 durant un únic cicle de rellotge.



29.
Aquest graf d'estats és molt semblant a l'obtingut en l'activitat 28. La diferència és que, si arriba un 1 estant en l'estat *ZUUZ*, anem a l'estat *ZU*, és a dir, el circuit reconeix que ha arribat la subseqüència 01. Per tant, deduïm que el 0 que ens ha portat fins a l'estat *ZUUZ* es considera com el primer d'una seqüència nova. Així, doncs, el circuit reconeix la seqüència 0110, però permet l'encavalcament entre seqüències consecutives.

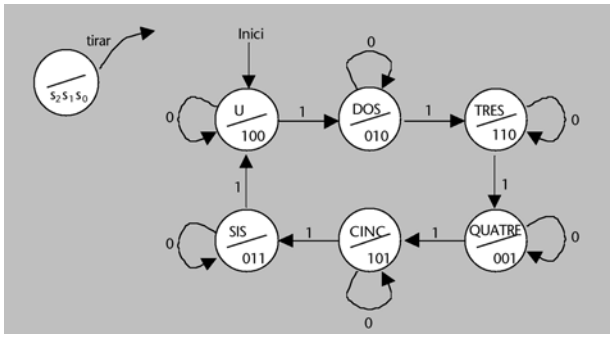
30.
Les entrades del circuit són *Sd* i *Se*, i les sortides, *SemD* i *SemE*. En cada moment hi pot haver només un semàfor en verd i, per tant, sempre es complirà que *SemD = SemE'*.
Quan un semàfor es posa en verd s'hi ha d'estar durant almenys dos cicles de rellotge; per tant, hi haurà dos estats (*VD1* i *VD2*) en els quals *SemD* = 1 i dos estats (*VE1* i *VE2*) en els quals *SemE* = 1.
Quan un semàfor ja ha restat en verd durant dos cicles, no es tornarà a posar en vermell fins que arribin cotxes per l'altra banda.
Atès que inicialment el semàfor esquerre ha s'ha de mantenir en verd durant dos cicles, l'estat inicial és *VE1*. El graf, doncs, és el següent:



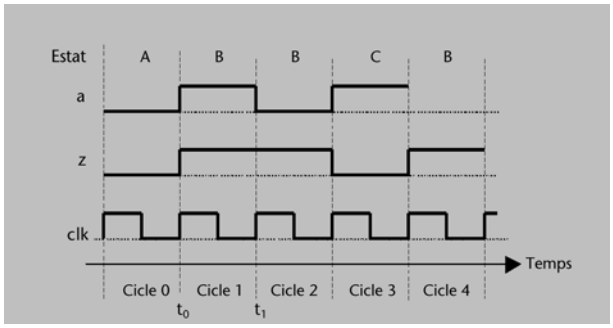
31.
Vegem com cal il·luminar els punts del dau per a formar les diferents combinacions:

Combinació	Punts il·luminats	Sortides s_2 s_1 s_0
1	d	1 0 0
2	c, e	0 1 0
3	c, d, e	1 1 0
4	a, b, f, g	0 0 1
5	a, b, d, f, g	1 0 1
6	a, b, c, e, f, g	0 1 1

El graf d'estats és el següent (l'estat inicial és U):

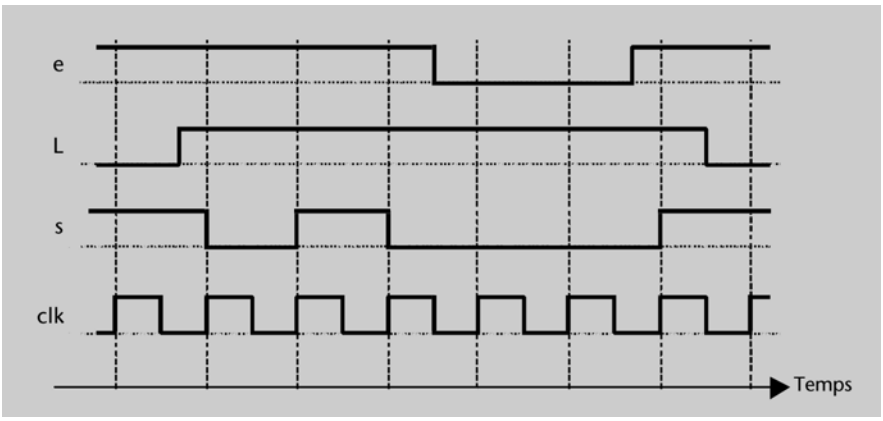


32.
En el cicle 0, el circuit està en l'estat *A*. En el cicle 1, passa a l'estat *B*. Per tant, en l'instant t_0 l'entrada havia de valer 0. Atès que suposem que l'entrada només canvia de valor en els flancs, obtenim que $a = 0$ durant tot el cicle 0.
Perquè el circuit continuï en l'estat *B* en el cicle 2, cal que en l'instant t_1 (i, per tant, durant tot el cicle 1), l'entrada valgui 1. Durant el cicle 2, l'entrada ha de valer 0 perquè el circuit passi a l'estat *C* en el cicle 3. Perquè torni a l'estat *B* en el cicle 4, l'entrada ha d'haver valgut 1 durant el cicle 3. No disposem de cap informació que ens permeti determinar quant val l'entrada durant el cicle 4.
El valor del senyal de sortida ve determinat per l'estat en què es troba el circuit: 0 mentre està en l'estat *A* o al *C*, i 1 mentre està en l'estat *B*.
El cronograma complet es mostra a continuació:



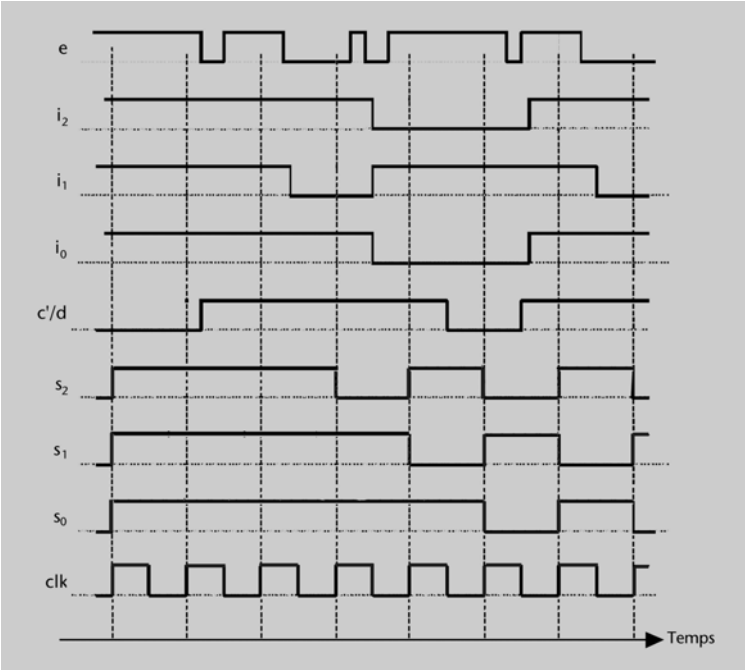
Exercicis d'autoavaluació

1. 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.



2. Com es pot veure en el 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 en cadascun dels biestables) prenen el valor de les entrades i_2, i_1 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*.



3.

a) El bit c val 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 valer 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 valer 0.
- Si $a_3 = 1$ i $b_3 = 0$, llavors $A > B$, i, per tant, c (i, per tant, f_{out}) ha de valer 1.
- Si $a_3 = b_3$, llavors cal saber quant valen els bits de menys pes de A i B per a decidir el valor que haurà de prendre c.

Fixem-nos en el valor del punt d del circuit. Aquesta seria la sortida del circuit B2 en el cas que els números fossin de tres bits en lloc de quatre. Per tant, $d = 0$ si $[a_2\ a_1\ a_0] < [b_2\ b_1\ b_0]$ i 1 en el cas contrari. Aquesta és la informació que fa falta per a decidir el valor de c en el cas que $a_3 = b_3$. Per tant, com que d 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 $[a_2\ a_1\ a_0] \geq [b_2\ b_1\ b_0]$

Els bits de pes –1 no existeixen, però 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}.

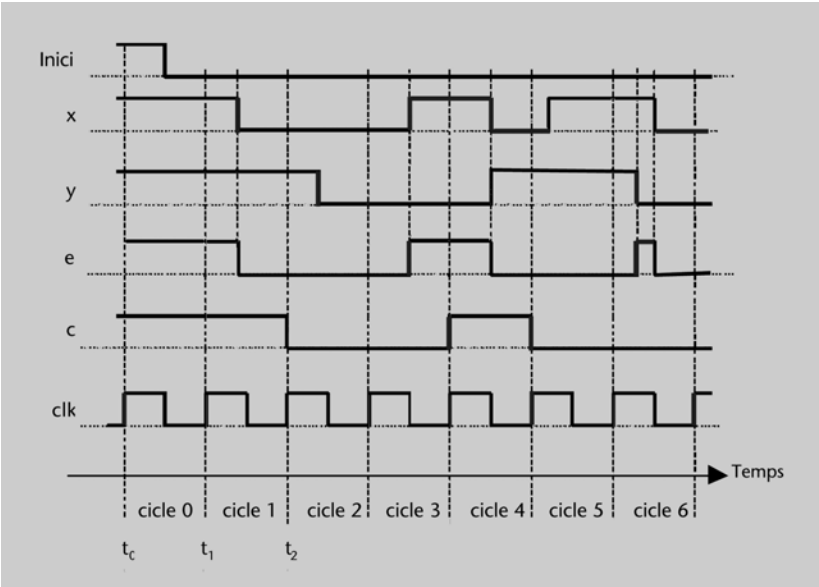
b) En l'apartat **a** hem vist que cada bloc B1 fa la comparació de dos bits de A i B. Aquesta comparació consulta, quan no pot decidir només amb els bits a_i i b_i, el resultat de la comparació dels bits de menys pes.

El circuit B3 compara dos bits d'entrada, x i y, i emmagatzema el resultat de la comparació en un biestable. El bloc B1 consultarà aquest resultat quan compari els bits x i y en el flanc de rellotge següent, 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 a carregar un 1 al biestable quan comença 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 que hi ha 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 a 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 de A i B es comparen parella per parella, de manera seqüencial en el temps. El circuit ha de tenir un biestable per a poder recordar en cada moment el resultat de la comparació de la parella anterior. El resultat final de la comparació estarà disponible quatre cicles després que el circuit comenci a funcionar.

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



Durant el cicle 0, $x = y = 1$. 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 de x, y i e al final de cicle 1 (en l'instant t_2), i, per tant, direm que $a_1 = 0$. Si seguim el mateix raonament en tots els casos, obtenim que els números que s'han comparat són els següents:

$$A = 0101001,\ B = 0110011.$$

Es compleix que $A < B$. Podem comprovar que, efectivament, la sortida c del circuit val 0.

4.
a) El circuit ha de saber si és de dia o de nit per a regular adequadament la durada de cada llum. Per tant, necessita un senyal d'entrada, que anomenarem d'/n, i suposarem que val 0 quan és de dia i 1 quan és de nit.

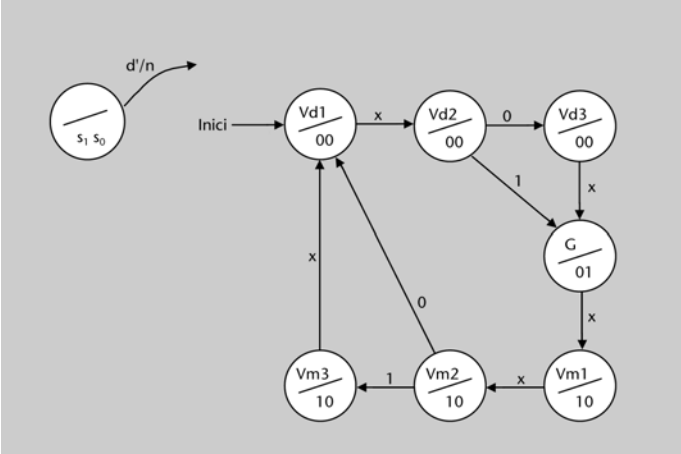
Les sortides han d'indicar quin llum està encès en cada moment. Per exemple, el circuit podria generar 3 sortides, una associada a cada llum. Una altra possibilitat és que generi només dos senyals de sortida, que controlin el semàfor de la manera següent:

s_1	s_0	Semàfor
0	0	Verd
0	1	Groc
1	0	Vermell

Prendrem aquesta segona possibilitat.

b) El graf d'estats es mostra a continuació. Hem identificat per *Vdi* els estats durant els quals el semàfor està en verd, per *G* l'estat durant el qual el semàfor està en groc, i per *Vmi* els estats durant els quals el semàfor està en vermell. Hem suposat que l'estat inicial és *Vd1*, però l'enunciat no ens deia res al respecte, per tant podíem haver pres qualsevol estat com a inicial.

Observem que mentre és de dia només es recorren 2 dels estats en què el semàfor està en vermell, i mentre és de nit només es recorren 2 dels estats en què el semàfor està en verd.

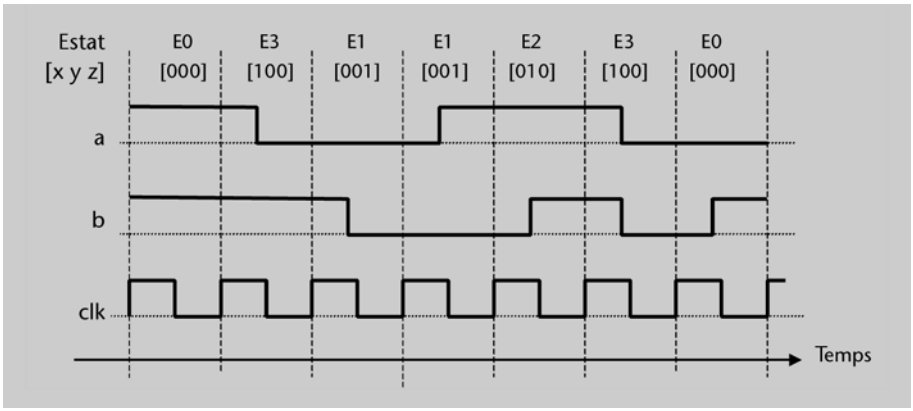


5.
a) Les taules es mostren a continuació:

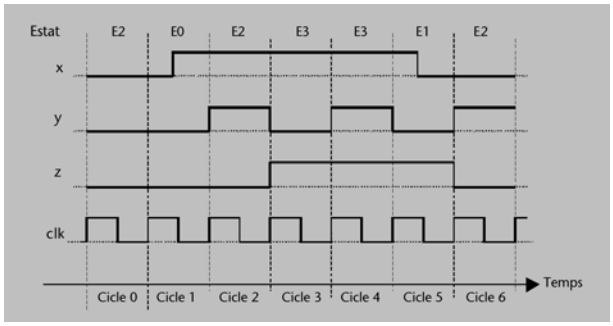
Taula de sortides	
Estat	**x y z**
E0	0 0 0
E1	0 0 1
E2	0 1 0
E3	1 0 0

Taula de transicions			
Estat	**a**	**b**	**Estat$^+$**
E0	0	0	E0
E0	0	1	E1
E0	1	0	E2
E0	1	1	E3
E1	0	0	E1
E1	0	1	x
E1	1	0	E2
E1	1	1	E2
E2	0	0	E1
E2	0	1	E3
E2	1	0	E1
E2	1	1	E3
E3	0	0	E0
E3	0	1	E1
E3	1	0	x
E3	1	1	x

b) El cronograma es mostra continuació.

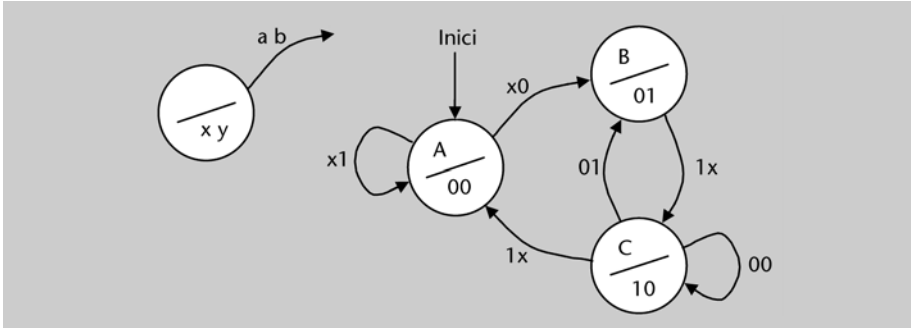


6.
El cronograma complet es mostra a continuació:

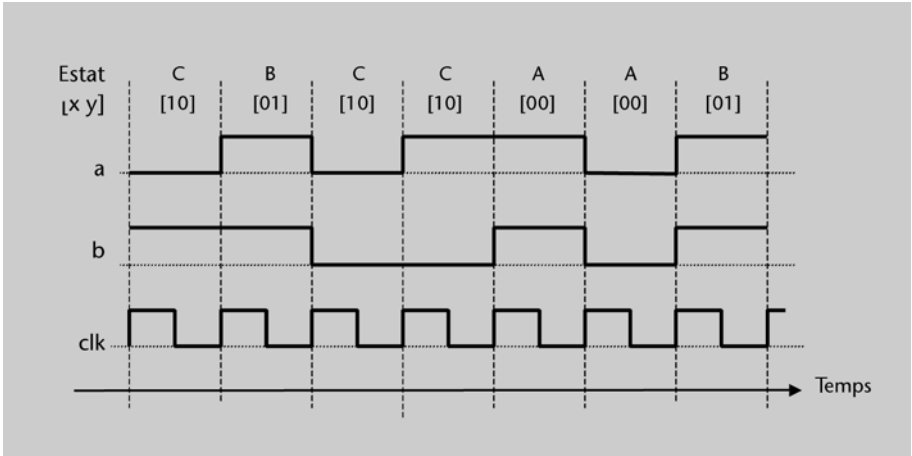


Recordeu que en el circuits seqüencials en el moment del flanc ascendent s'ha de tenir en compte el valor de les entrades a l'esquerra del flanc. Per exemple, a l'inici del cicle 2, *x* val 1 i *y* val 0.

7.
a) El graf es mostra a continuació.



b) El cronograma es mostra continuació:

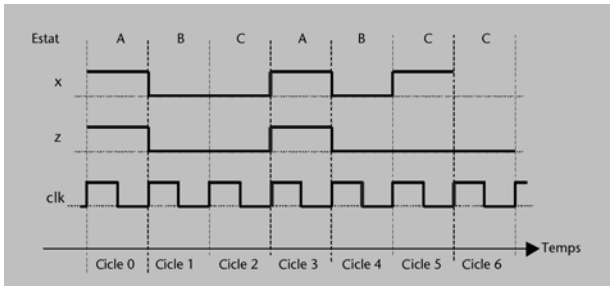


8.
Completarem el cronograma per passos:

- Durant el cicle 1 el circuit està en l'estat *B* i, per tant, la sortida val 0. Només es pot arribar a l'estat *B* des de l'*A* amb entrada 1. Per tant, deduïm que, durant el cicle 0, el circuit estava en l'estat *A* (la sortida valia 1, doncs) i que l'entrada valia 1.
- Durant el cicle 1, estem en l'estat *B* i l'*entrada* és 0; per tant, en el cicle següent anem a l'estat *C* amb sortida 0.
- Durant el cicle 3, la sortida val 1. L'únic estat en el qual la sortida val 1 és l'*A*. Per a arribar a l'estat *A* a partir del *C* s'ha de complir que, durant el cicle 2, l'entrada valgui 0.
- Durant el cicle 4, la sortida val 0. Si tenim en compte que venim de l'estat *A*, deduïm que en el cicle 4 estem en l'estat *B* (ja que no es pot anar al *B* des de l'*A*); per tant, durant el cicle 3, l'entrada valia 1.
- Durant el cicle 4 estem en l'estat *B* i l'entrada val 0; per tant, en el cicle 5 estarem en l'estat *C*, i la sortida valdrà 0.

- En el cicle 6 anem a un estat en el qual la sortida val 0. Si tenim en compte que durant el cicle 5 estàvem en l'estat *C*, deduïm que ens hem quedat en el mateix estat; per tant, l'entrada durant el cicle 5 havia de valer 1.
- No tenim cap informació que ens permeti saber quant val l'entrada durant el cicle 6.

El cronograma complet es mostra a continuació:



9.
Inicialment, el circuit està en l'estat *E0*, i s'hi queda mentre *p_ob* = 0. Si *p_ob* val 1, el circuit passa a l'estat *E1*. Deduïm, doncs, que l'estat *E1* indica que la porta de la caixa forta ha estat oberta durant 30 segons (un cicle de rellotge). Podem seguir el mateix raonament per a veure que l'estat *E2* assenyala que la caixa ha estat oberta durant un minut, i així successivament fins a l'estat *E4*, que indica que la caixa ha estat oberta durant dos minuts seguits. La sortida *alarma* val 1 només en l'estat *E4*.
A partir de les transicions, concloem que el senyal d'alarma s'activa quan la caixa resta oberta durant dos minuts seguits o més, i es desactiva un cicle després que es tanqui la porta de la caixa forta.

Bibliografia

Gajsky, D. D. (1997). *Principios de diseño Digital*. Prentice Hall.

Hermida, R.; Corral, A. del.; Pastor, E.; Sánchez, F. (1998). *Fundamentos de Computadores*. Madrid: Síntesis.