

Els circuits lògics seqüencials

Montse Peiron Guàrdia
Fermín Sánchez Carracedo

Índex

Introducció

Objectius

1. Caracterització dels circuits lògics seqüencials

- 1.1. Necessitat de memòria en els circuits lògics
- 1.2. Relotge. Sincronització

2. El biestable D

- 2.1. Dispositiu elemental de memòria. El biestable D
- 2.2. Senyal de càrrega
- 2.3. Entrades asíncrones

3. Blocs seqüencials

- 3.1. Registre
- 3.2. Banc de registres
- 3.3. Memòria RAM

Resum

Bibliografia

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 depenen del valor dels senyals d'entrada en aquest mateix moment. Quan els senyals d'entrada varien, llavors els de sortida també variaran en conseqüència (després del retard introduït per les portes i blocs, que en aquest curs no tenim en compte).

Ara bé, en algunes aplicacions es necessita que el valor dels senyals de sortida no depengui només de les entrades en el mateix moment, sinó que tingui també en compte els valors que han pres les entrades amb anterioritat. En els circuits que hem conegut fins ara, això no és possible: fan falta els elements que conformen els "circuits lògics seqüencials".

En aquest mòdul es coneixerà 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.

Objectius

L'objectiu fonamental d'aquest mòdul és conèixer els circuits lògics seqüencials, és a dir, saber com estan formats i poder utilitzar-los amb agilitat.

Per a arribar a aquest punt caldrà haver satisfet els objectius següents:

- a partir de la funcionalitat que es vol que tingui un circuit lògic, saber discernir si el circuit ha de ser de tipus seqüencial o combinacional
- entendre el concepte de memòria, la necessitat d'una sincronització en els circuits lògics seqüencials i el funcionament del senyal de rellotge
- conèixer el funcionament del biestable D i de totes les entrades de control de què pot disposar
- conèixer la funcionalitat dels diferents blocs seqüencials, i saber utilitzar-los en el disseny de circuits

Després de l'estudi d'aquest mòdul hem de ser capaços de construir un circuit qualsevol que combini tant dispositius seqüencials com combinacionals, així com d'entendre la funcionalitat d'un circuit donat que contingui tots aquests elements.

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 d' n bits, que interpretem com a números representats en complement a 2. Suposem que volem que $Z = X+2$. Amb els elements estudiats al mòdul "Els circuits lògics combinacionals" sabem com fer-ho, inclús de moltes maneres diferents. Quan el valor present a l'entrada X variï, llavors Z també canviarà de valor consegüentment.

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 d' X ha variat). Amb els dispositius lògics que coneixem fins ara no ho podem aconseguir, perquè en canviar el valor d' X , el valor anterior ha "desaparegut" i ja no el podem fer servir per a calcular la suma.

És necessari 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.

La denominació...

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

1.2. Relotge. 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 "tots els valors que han estat presents a l'entrada X durant un temps determinat"? El senyal X pot anar canviant de valor de forma aleatòria en el temps: pot valdre 13 durant 4 ns., després -25 durant 10 ns., després 0 durant 1 ns., 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 poder determinar-ho, el circuit ha de disposar d'un mecanisme de *sincronització*. En els circuits seqüencials que estudiarem en aquest mòdul s'utilitza un senyal de *relotge* com a forma de sincronització.

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

A aquesta tasca que duu a terme el senyal de relotge se l'anomena **sincronització** dels circuits.

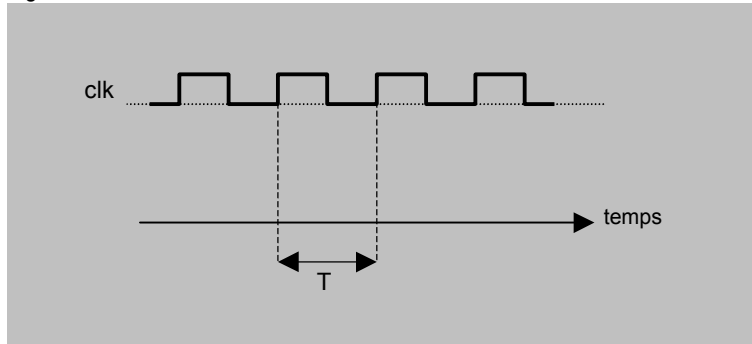
El que fa el relotge...

...és **discretitzar** el temps: en lloc de veure'l com una dimensió contínua, els circuits el veuen com una seqüència d'instants.

Concretament, el senyal de relotge 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 fer referència al senyal de relotge (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 mesura en segons o, més habitualment, en nanosegons (mil·lionèsimes de segon).

Figura 1



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 i el següent s'anomena **cicle** o **cicle de rellotge**. Per tant, la duració d'un cicle és un període, T segons.

La **frequència** 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 Hertz (cicles per segon); el més habitual és usar el múltiple "megaHertz" (milions de cicles per segon), que s'abreuja "MHz". Per exemple, si tenim un rellotge amb un període de 2 ns., la seva freqüència és de

$$\frac{1 \text{ cicle}}{2 \cdot 10^{-9} \text{ segons}} = 0'5 \cdot 10^9 \text{ cicles/segon} = 500 \cdot 10^6 \text{ cicles/segon} = 500 \text{ MHz}$$

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 a l'apartat següent.

2. El biestable D

2.1. Dispositiu elemental de memòria. El biestable D

A 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.

Tal com s'ha vist al 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.

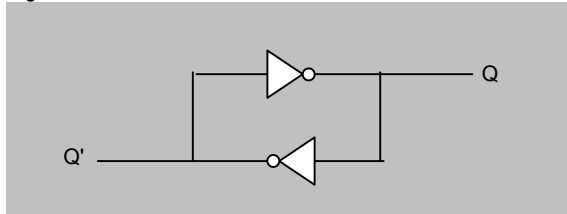
Físicament, el senyal de rellotge es genera a partir de cristalls de quars, un mineral que té la propietat anomenada **piezoelectricitat**: en rebre corrent elèctrica vibra amb una freqüència extremadament gran i regular.

A finals de l'any 2000, els processadors comercials més ràpids tenen freqüències de rellotge pròximes al GHz ("gigaHertz", 1000 milions de Hz), amb períodes propers a 1 nanosegon.

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

Examinant el circuit que es mostra a la figura 2 veiem que el valor que hi hagi en els punts Q i Q' (0 o 1) s'hi 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 dissenyar un circuit que tingui aquesta mateixa capacitat de memòria, però que a més permeti que el valor al punt Q pugui canviar en funció dels requeriments de l'usuari. Un circuit amb aquestes característiques s'anomena *biestable*.

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

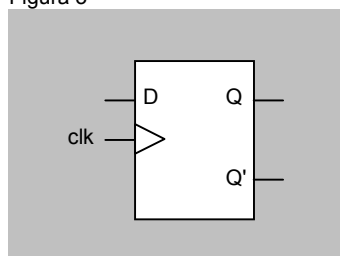
Un biestable té dues sortides, Q i Q'. Es diu que Q és "el valor que guarda el biestable" en cada moment, o "el valor emmagatzemat al biestable", i Q' és la seva negació.

La denominació "biestable"...

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

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

Figura 3



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

El biestable D funciona de la manera següent:

La sortida Q pren el valor que hi hagi a 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 \uparrow 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 del propi senyal Q en un instant futur: a partir del moment en què es produeixi el proper flanc. Aquesta notació, doncs, ens permet descriure amb precisió l'evolució temporal dels senyals en un circuit lògic seqüencial.

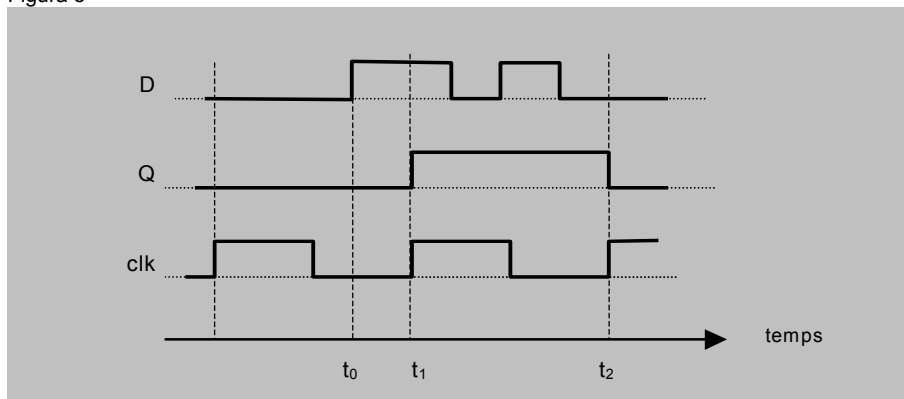
Figura 4

D	clk	Q^+
0	\uparrow	0
1	\uparrow	1

La figura 5 mostra el cronograma del comportament d'un biestable D. Es pot observar que, tot i que D es posa a 1 en l'instant t_0 , Q no canvia de valor fins a l'instant t_1 , perquè no és fins en aquest moment que es produeix un flanc ascendent de rellotge.

Malgrat les subseqüents variacions de D , Q es manté inalterat fins a l'instant t_2 , en què es produeix el proper flanc de rellotge.

Figura 5

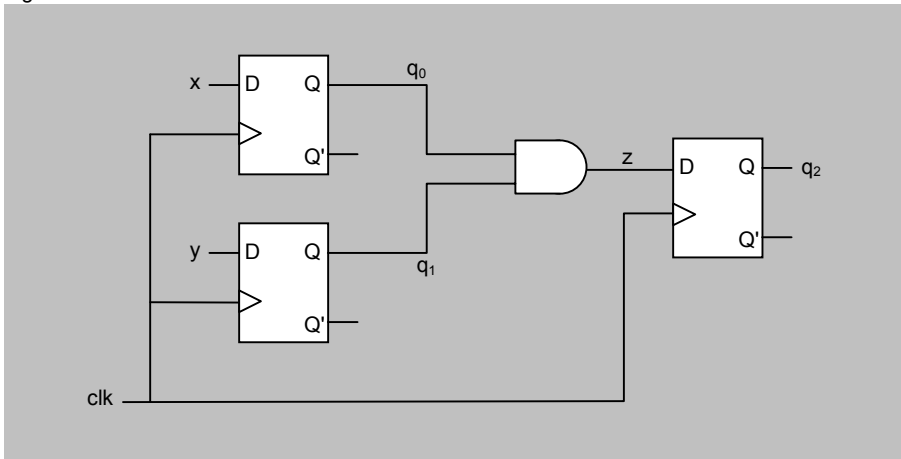


Notem que...

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

En els circuits reals, és habitual realitzar alguna funció combinacional sobre la sortida d'un o més biestables i connectar el resultat a l'entrada d'un altre biestable. La figura 6 en mostra un exemple.

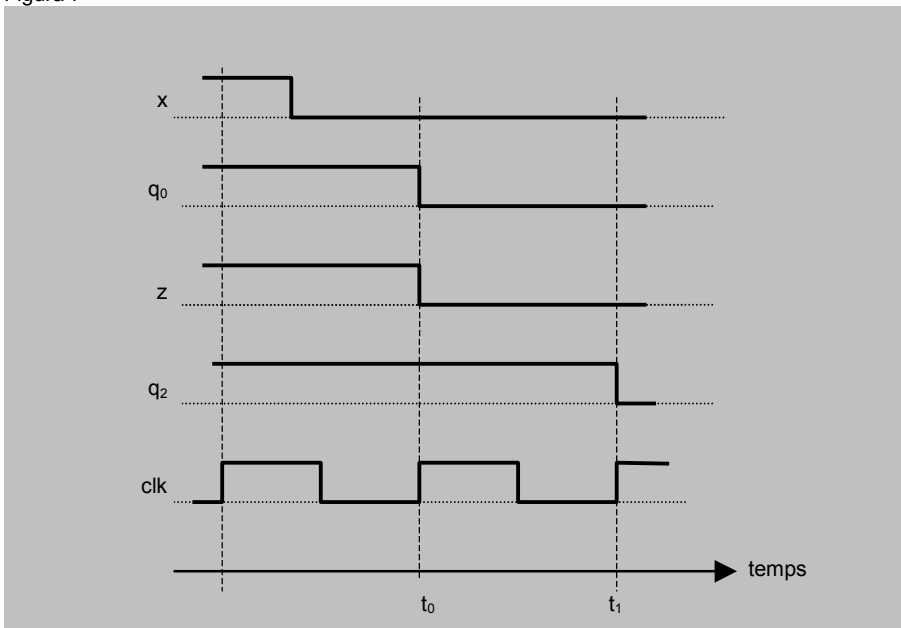
Figura 6



Aquesta estructura de circuit és també molt habitual usant **registres** en lloc de biestables. Els registres són blocs seqüencials que s'estudiaran a l'apartat següent.

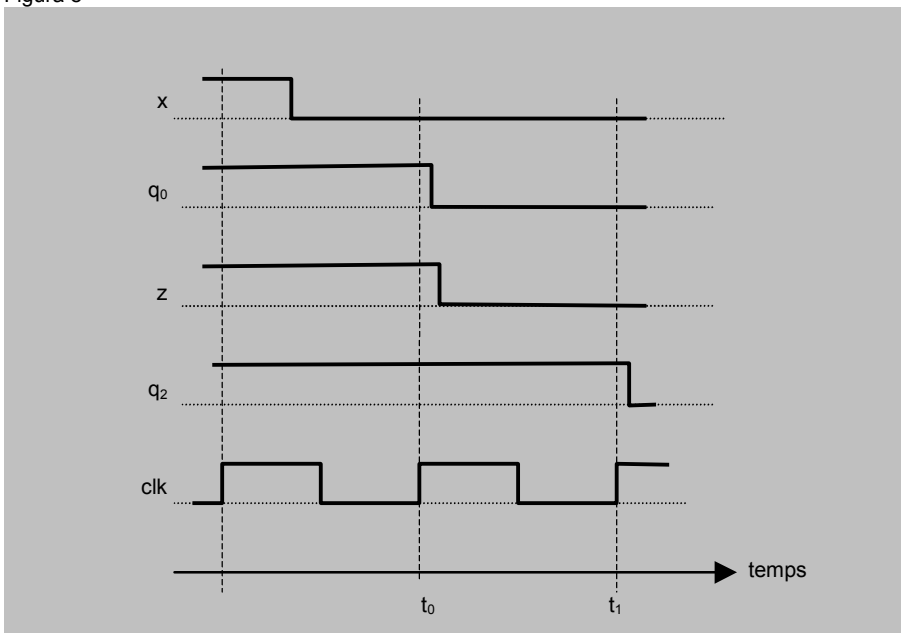
La figura 7 mostra el cronograma de l'evolució d'aquest circuit durant un cert interval de temps (per tal de simplificar el dibuix, no hi hem posat ni y ni q_1 ; assumim que tots dos senyals es mantenen a 1 tota l'estona). En l'instant t_0 , q_0 es posa a 0, perquè a l'entrada D del biestable corresponent hi ha un 0 ($x = 0$); en conseqüència, z també es posa a 0. En dibuixar la línia del cronograma corresponent a q_2 , podríem dubtar de si s'ha de posar a 0 en aquest mateix instant, ja que sobre la línia vertical corresponent a t_0 , z està tant a 1 com a 0.

Figura 7



Aquest dubte sorgeix pel fet que no tenim en compte els retards que es produeixen en les transicions dels valors dels senyals. En la realitat, z es posa a 0 com a conseqüència de què q_0 s'ha posat a 0, i això s'ha produït com a conseqüència del flanc indicat per t_0 . Per tant, en realitat tant q_0 com z es posen a 0 *després* de t_0 . Per tant, en aquest instant q_2 es posa a 1 (ja hi era, de fet), perquè z val 1. q_2 no es posarà a 0 fins a t_1 , amb el següent flanc de rellotge. La figura 8 mostra el mateix cronograma, però amb els retards reflexats.

Figura 8



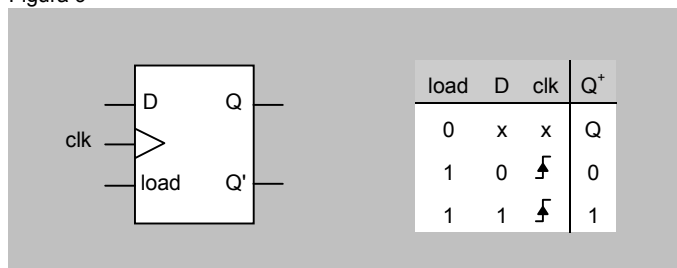
2.2. Senyal de càrrega

En posar un biestable dins d'un circuit amb altres components, a vegades ens interessarà que el contingut del biestable no canviï encara que variï el valor de D: voldrem que el biestable sigui "insensible" a les variacions en 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 a l'apartat anterior.

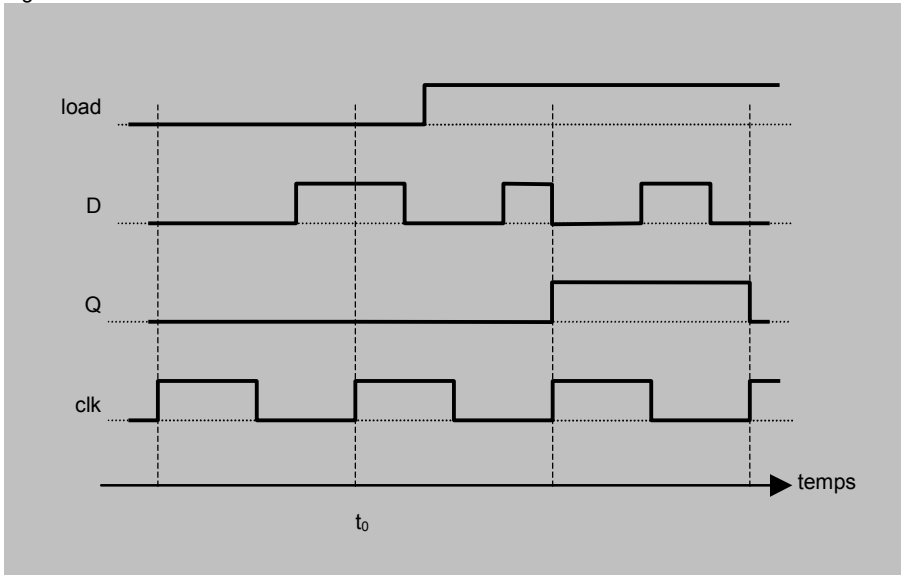
La figura 9 mostra la representació gràfica d'un biestable D amb senyal de càrrega (s'identifica per la paraula *load*), i la taula de veritat del seu funcionament. Com és habitual, les "x"s indiquen valors qualssevol dels senyals.

Figura 9



La figura 10 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 a l'apartat anterior.

Figura 10



2.3. Entrades asíncrones

El valor d'un biestable D pot variar en els instants de flancs ascendents en funció del 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 en posar-se en marxa un circuit.

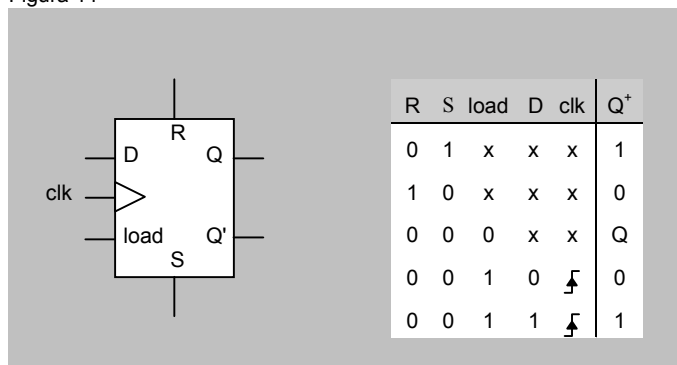
Les **entrades asíncrones** d'un biestable permeten modificar el seu valor de forma instantània, independentment del valor del senyal de rellotge i de les entrades D i *load*. Es diu que les entrades asíncrones tenen més **prioritat** que la resta d'entrades.

Els biestables solen disposar de dues entrades asíncrones:

- *R* (de l'anglès *Reset*): Quan 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 11 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.

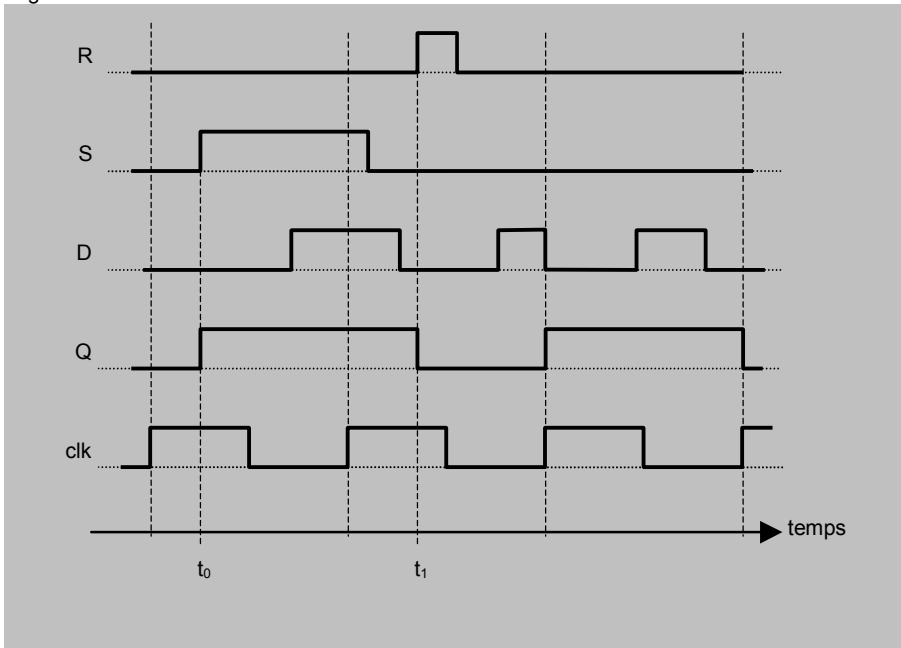
Figura 11



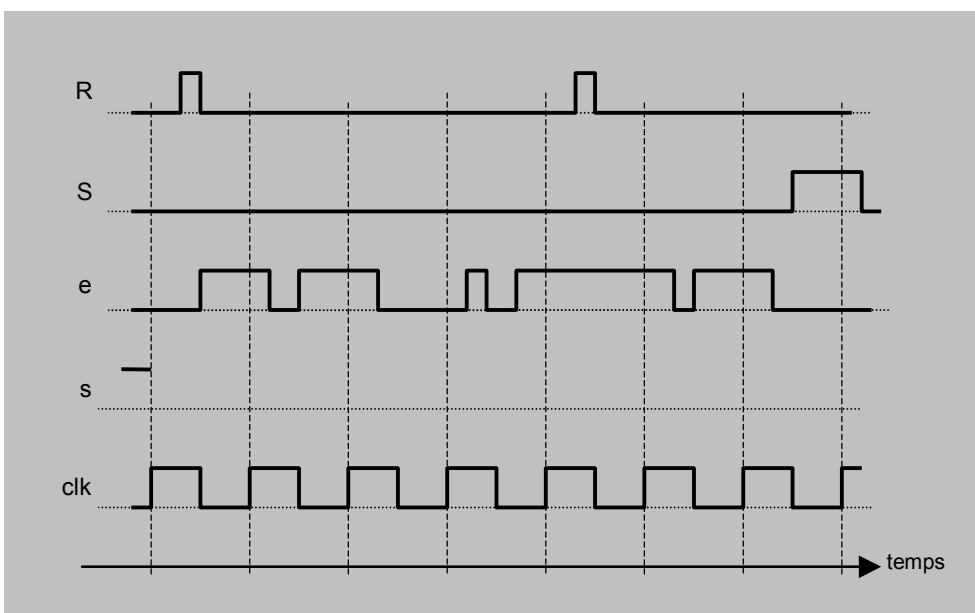
Si tant R com S valen 1, el comportament del biestable no es pot predir. En els circuits reals sempre es garanteix que aquesta situació no es donarà mai.

La figura 12 mostra el cronograma del comportament d'un biestable D amb entrades asíncrones i senyal de càrrega (per 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 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 12



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



Els circuits seqüencials solen disposar d'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). En posar-se el circuit en funcionament, el senyal *Inici* val 1 durant un cicle de rellotge (es diu que *fa un pols a 1*), i després baixa a 0 i s'hi manté fins al final.

3. Blocs seqüencials

3.1 Registre

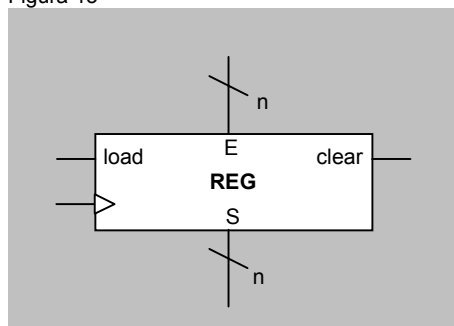
Hem vist que un biestable permet guardar el valor d'un bit. Per a guardar el valor d'un mot d' n bits, caldran n biestables D.

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

La figura 13 mostra la representació gràfica d'un registre. Es pot veure que disposa dels senyals següents:

- una entrada de dades d' 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 d' 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 13



El funcionament del registre és el següent:

- el senyal *clear* serveix per posar el contingut del registre a 0. Donat 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*: 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 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

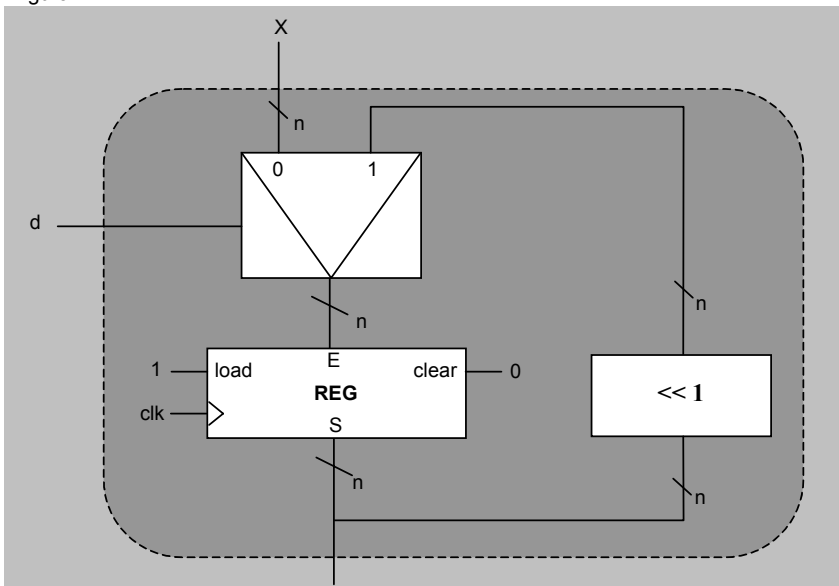
Si en un registre no hi dibuixem els senyals *load* o *clear*, assumirem que estan a 1 i a 0 respectivament.

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**.

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 14 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 14



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

3.2. Banc de registres

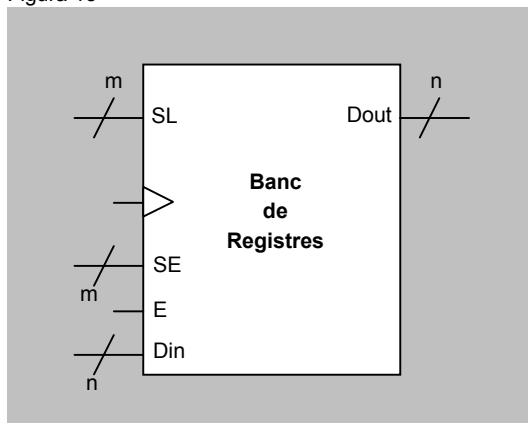
Un **banc de registres** és una agrupació d'un cert nombre de registres, tots ells del mateix nombre de bits. El contingut dels registres es pot llegir i modificar gràcies als *ports* de lectura i d'escriptura.

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

- una entrada de selecció de lectura, *SL*, d' 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*, d' 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 15



El banc de registres funciona de la manera següent:

- per 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 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 al registre indicat per *SE*.

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.

3.3. Memòria RAM

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 2 ports de lectura i 1 d'escriptura, es poden llegir dos registres i escriure'n un altre simultàniament.

Fixem-nos que...

... el banc "sempre 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.

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

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 algunes desenes de mots, mentre que una memòria RAM en pot guardar diversos milions
- la velocitat: 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 realitzen coincidint amb els flancs ascendents del rellotge. La memòria no disposa de senyal de rellotge: les escriptures són efectives un cert interval de temps després d'haver donat l'ordre d'escriure (uns pocs nanosegons).

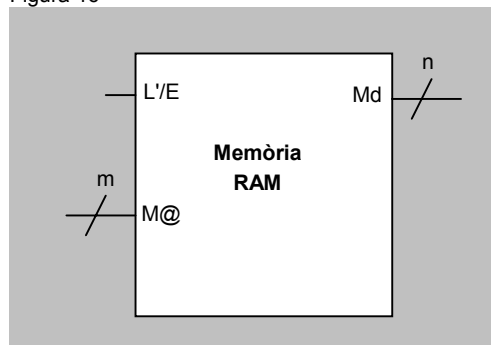
En aquest curs no s'estudiaran en detall aquestes qüestions. És suficient amb quedar-nos amb la idea que un banc de registres és petit i ràpid, i una memòria RAM és gran i lenta.

Igual que en el cas de la memòria ROM que s'ha vist al 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 al calaix amb adreça i .

La figura 16 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 , d' n bits (si els mots que guarda la memòria són d' n bits)
- una entrada de control, L/E , que indica en tot moment si s'ha de realitzar una lectura o una escriptura.

Figura 16



El funcionament de la memòria és el següent:

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 en fer una lectura o una escriptura no depèn de quin mot s'estigui accedint (a diferència del que passava en altres dispositius de memòria que es feien servir en els primers computadors).

La memòria s'estudiarà amb més detall a les assignatures *Estructura de Computadors I i II*.

Les memòries...

... també poden disposar d'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.

- si $L/E = 0$, llavors es realitza una lectura: pel bus Md surt el valor del mot que està guardat a l'adreça indicada per $M@$
- si $L/E = 1$, llavors es realitza una escriptura: el mot indicat per $M@$ pren el valor que hi ha a Md (un cert interval de temps després que L/E s'hagi posat a 1).

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 8 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 significat següent:

lletra	significat	exemple
k	$2^{10} = 1024 \cong 10^3$	16 kb (16 kbytes) = 2^{16} 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

Resum

En aquest mòdul s'han estudiat els circuits lògics seqüencials. S'ha vist que el que els caracteritza és la capacitat de memòria, i per tant són capaços de determinar el valor dels senyals de sortida en funció no només del 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 es pot "congelar" el seu valor mitjançant un senyal de càrrega.

Finalment, s'han presentat els diferents blocs seqüencials que permeten 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).

Els blocs seqüencials i combinacionals que s'han estudiat fins en aquest punt del curs constitueixen un conjunt de dispositius suficient per a dissenyar un computador senzill, tal com es veurà al mòdul "Estructura bàsica d'un computador".

Bibliografía

Hermida R., Del Corral A., Pastor E. y Sánchez F. (1998)
Fundamentos de Computadores Madrid: Editorial Síntesis

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