

# 4. CIRCUITS SEQUENCIALS

## 4.1. OSCIL·LADORS, SENYALS DE RELLOTGE.

## 4.2. CARACTERISTIQUES DELS OPERADORS SEQUENCIALS.

## 4.3. BIESTABLES.



4.3.1 R-S.

4.3.2 D.

4.3.3 J-K.

4.3.4 T.

4.3.5. IMPORTANCIA DE L'ACTIVACIÓ PER NIVEL O PER FLANC.

## 4.4. REGISTRES D'EMAGATZEMAMENT. MEMÒRIES RAM.



Tipus de Memòries RAM.

Memòries ROM.

## 4.5. REGISTRES DE DESPLAÇAMENT.

## 4.6. CONVERSORS.



4.6.1 PARALLEL - SÈRIE

4.6.2 SÈRIE - PARALLEL.

## 4.7. COMPTADORS.




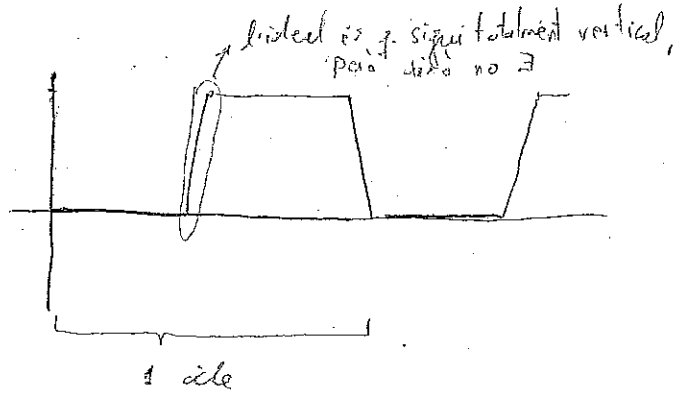
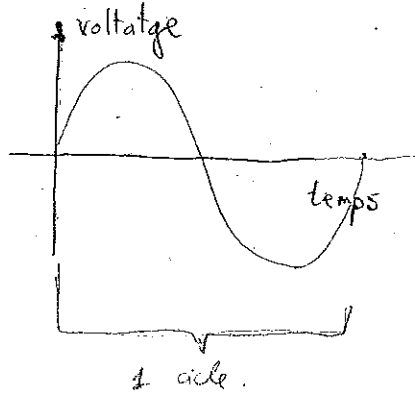
4.7.1 ASINCRON.

4.7.2 SINCRO.

4.7.3 ALTRES

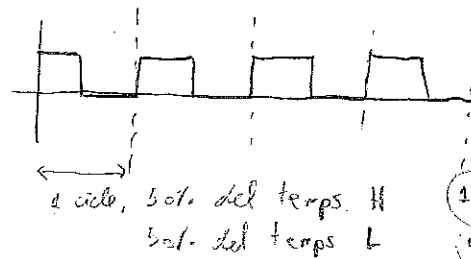
# 4.1. OSCIL·LADORS: SENYALS DE RELOTGE.

- Ja n'hem parlat, simbolitzant-lo així  i aliquid q. incorporen un cristall de SiO<sub>2</sub>
- En realitat, un generador d'impulsos és un circuit més complicat, que incorpora resistències (R), i condensadors, de manera que es pugui generar una ona quadrada. (C)



La velocitat d'un circuit ve determinada pel n<sup>o</sup> de cicles/segon, o Hertz.

Frequència =  $F = \nu$



$\rightarrow \frac{4 \text{ cicles}}{1 \text{ segon}} = 4 \text{ Hz} = 4 \text{ s}^{-1}$

Període =  $\frac{1}{F} = \frac{1}{\nu}$  (Duració d'un cicle)

En aquest cas,

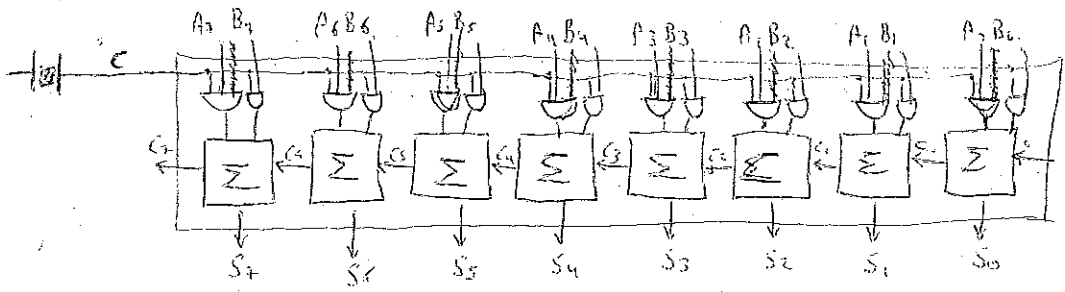
$\Rightarrow 4 \text{ Hz} \rightarrow \text{Període} = \frac{1}{4} = \frac{1}{4 \text{ s}^{-1}} = 0,25 \text{ segons}$

\* Els circuits que estudiarem es poden "activar" per nivell o per flanc de pujada.<sup>o baixada.</sup>  
 Es pot suposar que el ~~segon cas ens~~ ~~permet~~ ~~que~~ el període,  $\frac{T}{2}$ , és el temps mínim que es manté l'activació. (més endavant en parlarem). El fil per on entra aquest senyal s'etiqueta com "clock", o "ck".

Exercici Càlcul de la F màxima del sumador de 8 bits anterior, suposant tecnologia CMOS alimentada amb 5 volts.

Conclusió Hi ha altres tipus de Σ, que no calculen el carry en sèrie, sinó que sumen "una mica més en paral·lel". Poden funcionar a major F, però són més complexes.

- Suposant que tenim un sumador de 8 bits com el que hem explicat abans, i que l'hem construït amb tecnologia CMOS alimentada amb 5 volts, i que les entrades  $A_7 \dots A_0$ ,  $B_7 \dots B_0$  estan controlades per un dispositiu síncron; calcular la Freqüència màxima a la que pot funcionar aquest sumador.



El carry ha de travessar 8  $\Sigma$  unitaris, 2 portes lògiques en cada un.

Total = 16 portes lògiques.

Retard / porta lògica en CMOS a 5V = 60 ns } Temps total = 960 ns

1 ns =  $10^{-9}$  segons:

(10V  $\Rightarrow$  30 ns)

960 ns =  $960 \cdot 10^{-9}$  segons = 0,000000960 segons =  $9.6 \times 10^{-7}$

Això és el temps mínim que  $A_i B_i$  s'han de mantenir activades.

El període ha de mantenir el nivell 1, o H, al menys aquest temps.

$$\frac{T}{2} > 9.6 \times 10^{-7}$$

$$T > 1.92 \times 10^{-6}$$

$$\frac{1}{F} > 1.92 \times 10^{-6}$$

$$\frac{1}{1.92 \times 10^{-6}} > F$$

$$\frac{1}{1.92} \times 10^6 > F$$

$$0.52 \times 10^6 > F$$

~~1.92 / 10^6~~

$$0.52 \text{ MHz} > F$$

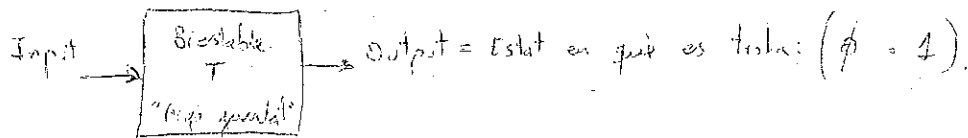
$$F \text{ ha de ser menor que } 0.52 \text{ MHz}$$

Overclocking, massa velocitat, pot ser perjudicial.

Observació: Hi ha altres tipus de sumadors que no calculen el carry en sèrie, sinó "en paral·lel". Són més complexos. Poden funcionar a major F.

## 4.2 CARACTERITZACIÓ DELS OPERADORS SEQUÈNCIALS.

- Un operador seqüencial és una entitat més complexa que una porta lògica, però que no arriba a ser un bloc funcional.
- En el cas de la lògica combinatòria, un operador combinatori seria, per exemple, la porta OREX.
- D'operadors seqüencials n'hi ha 4: el biestable R-S, el biestable D, el biestable J-K i el biestable T.
- El nom biestable deriva del fet de que és un dispositiu que pot emmagatzemar la unitat mínima de memòria, un bit, i es deu, aleshores, que el dispositiu està en estat 0 o en estat 1 en un moment donat.
- Noms equivalents biestable = bàscula = Flip-Flop.



Es tenen en compte aquestes tres coses:

a) Sincronisme d'activació (dispar)

→ Sincron Si intervé el rellotge.

→ Asincron Si no intervé el rellotge.

$$\text{Estat Futur} = F(\text{Estat actual, input, rellotge})$$

$$\text{Estat Futur} = F(\text{Estat actual, input})$$

b) Tipus de dispar: per flanc, o per nivell - Pujada o baixada  
 - Manteniment de 0 o 1

c) Per la lògica de dispar, segons com s'objectua el canvi d'un estat a un altre donats els possibles inputs i l'estat actual:

Inputs	E. Actual	E. Futur
000		:
...		:
111		:

Identifica un dels 4 biestables existents: R-S, D, J-K, T.

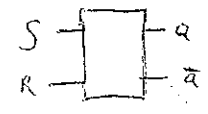
Es semblant a les taules de veritat que descriuen els circuits combinatorials.

4.3. BIESTABLES.

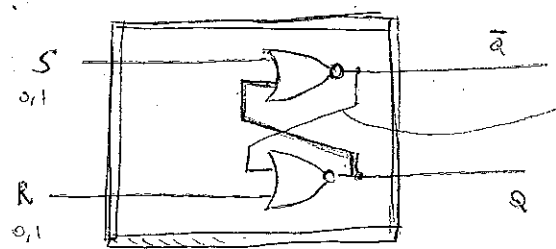
4.3.1 R-S.

⊗ Té 2 entrades, R i S, (que responen a RESET i SET). L'estat és Q, i les dues sortides son Q i  $\bar{Q}$ .

⊗ Es pot implementar amb portes NOR o portes NAND.



⊗ La implementació NOR és aquesta:



Aquest bucle és el que "captra" el corrent elèctric, de 0V o 5V, que hem anomenat "estat Q".  
Q és la part remarcada. L'altre és  $\bar{Q}$ .

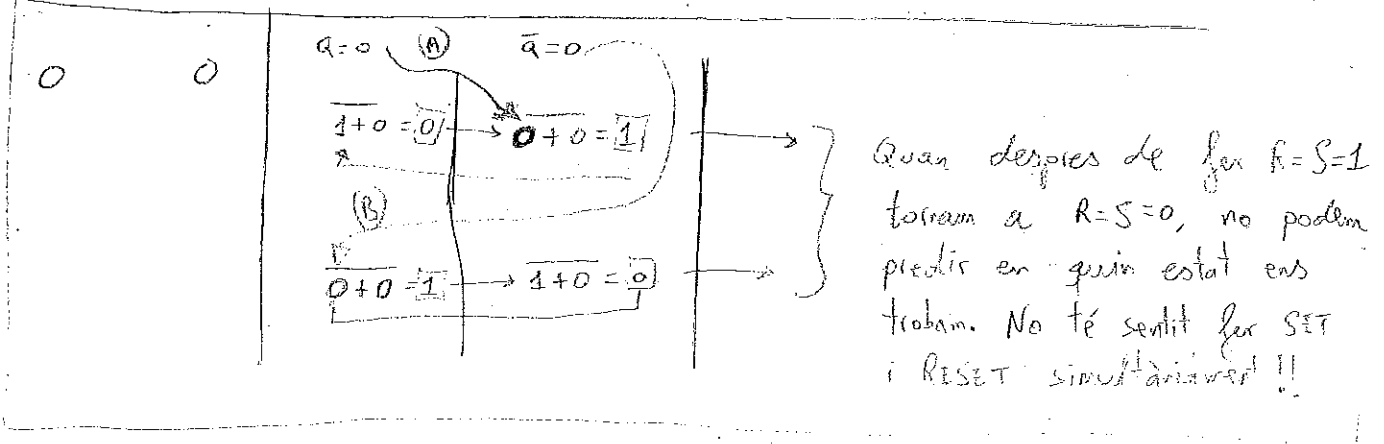
Comportament del biestable R-S.

→ Les entrades R-S poden prendre 4 valors, (00, 01, 10, 11), i l'estat inicial de Q i  $\bar{Q}$  pot ser 0 i 1 o 1 i 0. (En principi no s'admeten d'altre.)

R	S	Estat Inicial		Notes
		Q=0	$\bar{Q}=1$	
0	0	$Q^+ = \bar{Q} + R$ $1 + 0 = 1$	$\bar{Q}^+ = Q + S$ $0 + 0 = 0$	S=R=0 no provoca canvis, independentment de l'estat inicial.  R=0, S=1, independentment del valor inicial, Q sempre acaba = 1, $\bar{Q}$ sempre acaba = 0. Hem fet un SET. Quan tornem a R=S=0, ja sabem que OK.
0	1	$Q^+ = \bar{Q} + R$ $0 + 0 = 0$	$\bar{Q}^+ = Q + S$ $1 + 1 = 0$	
1	0	$Q^+ = \bar{Q} + R$ $1 + 1 = 0$	$\bar{Q}^+ = Q + S$ $0 + 0 = 0$	

R	S	Estat inicial		Estat inicial	
		$Q=0$	$\bar{Q}=1$	$Q=1$	$\bar{Q}=0$
		$Q^* = \bar{Q} + R$	$\bar{Q}^* = \bar{Q} + S$	$Q^* = \bar{Q} + R$	$\bar{Q}^* = \bar{Q} + S$
1	1	$0+1=1 \rightarrow 0+1=1$ $1+1=0 \rightarrow 0+1=1$ $0+1=1$	$0+1=1$ $1+1=0$ $0+1=1$	$0+1=1$ $1+1=0$ $0+1=1$	Sempre s'inverteix amb $Q = \bar{Q} = 0$

Problema de  $Q = \bar{Q} = 0$ , quan tornam a  $R = S = 0$ ,

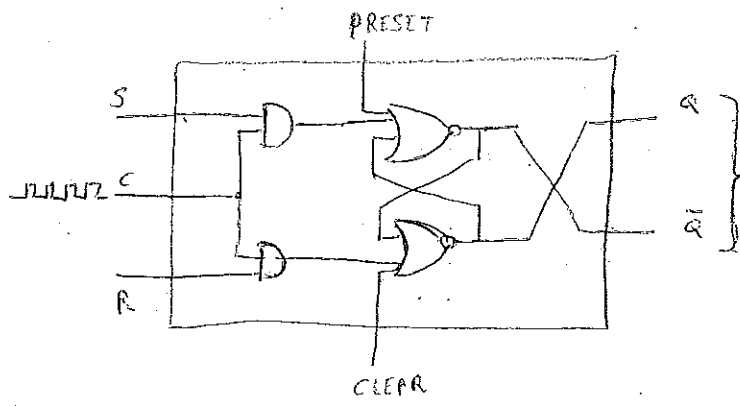


Taula d'estats del bistable R-S amb portes NOR:

R	S	Q	$Q^* = F(R, S, Q)$
0	0	0	0 → No fa res
0	0	1	1 → No fa res
0	1	0	1 → SET
0	1	1	1
1	0	0	0 → RESET
1	0	1	0
1	1	0	?
1	1	1	?

El bistable es posa a 1 quan  $S=1$ .  
 S'activa per nivell alt.

- Amb portes NAND tb. es pot construir un biestable R-S. Molt senzill. (S'actua per nivell baix).
- El biestable que acabam de descriure és **asíncron** no depen d'un senyal de rellotge. Si afegim el senyal C, obtenim un circuit **síncron** on  $Q^+ = F(R, S, Q, C)$ .

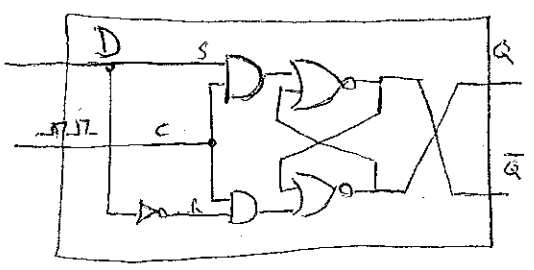


Invertim el gràfic, per tenir Q "a dalt"  
 PRESET → "antic SET". Per preposicionar a 1  
 CLEAR → Col·locació inicial a 0.

Aquest biestable s'anomena SR-C, (Set, Reset, Clock).

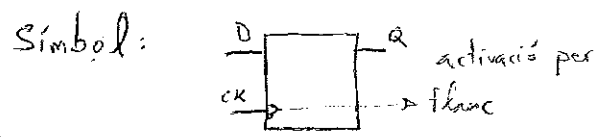
4.3.2 BIESTABLE D.

Es un biestable SRC precedit d'un inversor que distribueix les entrades S i R. (sense clear i Preset o amb ells).



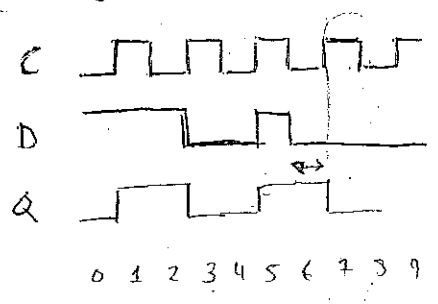
$D=0 \Rightarrow S=0, R=1$   
 $D=1 \Rightarrow S=1, R=0$   
 S'emmagatzema el valor D al cap d'un cicle de rellotge.

"D" ve de "Data" S'emmagatzema 1 bit.



Taula d'estats

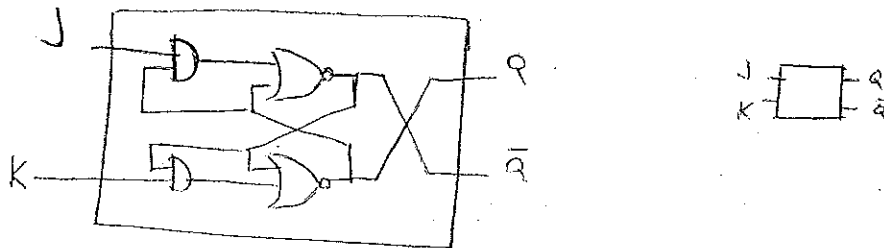
Cronograma:



En el cas de l'activació per nivell.

### 4.3.3. BIESTABLE J-K

Es un biestable R-S al qual se li elimina la indeterminació que presenta l'input  $R=S=1$  amb 2 portes AND disposades de la següent manera:

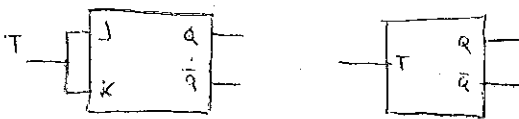


• No ho demostrarem, però la taula d'estats del biestable J-K amb portes NOR és la següent:

J	K	Q	Q <sup>+</sup>	
0	0	0	0	→ El deixa igual
0	0	1	1	
0	1	0	0	→ RESET
0	1	1	0	
1	0	0	1	→ SET
1	0	1	1	
1	1	0	1	→ canvia de valor !!
1	1	1	0	

També existeix l'equivalent síncron.

### 4.3.4. BIESTABLE T . És un J-K amb una sola entrada:

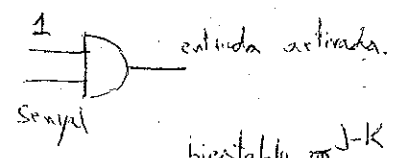


T	Q	Q <sup>+</sup>	
0	0	0	T=0, deixa Q igual
0	1	1	
1	0	1	T=1, inverteix Q.
1	1	0	



### 4.3.5. Importància de l'activació per nivell o per flanc.

o Els biestables síncrons activats per nivell canvien d'estat quan el senyal de rellotge és 1.

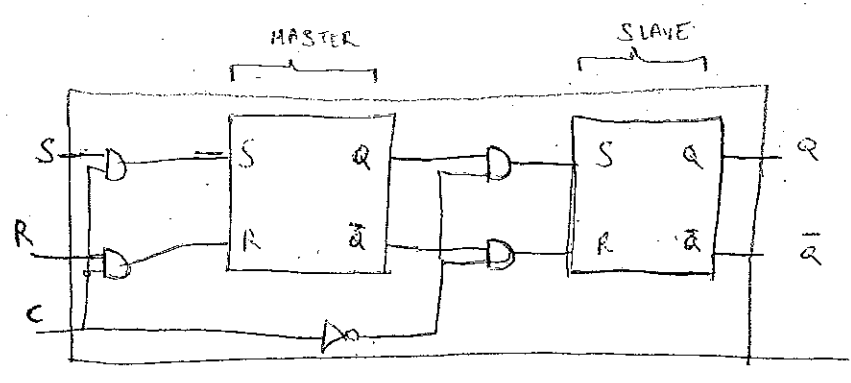


Per exemple al biestable J-K,  $J=1, K=1$  "entra" quan  $C=1$ .

Problema: Si  $C=1$  durant molt de temps,  $J=1$  val 1 durant molt de temps, la qual cosa pot ~~canviar~~ <sup>implicar</sup> si (tenim  $J=K=1$ ) que es canvi de valor varies vegades.

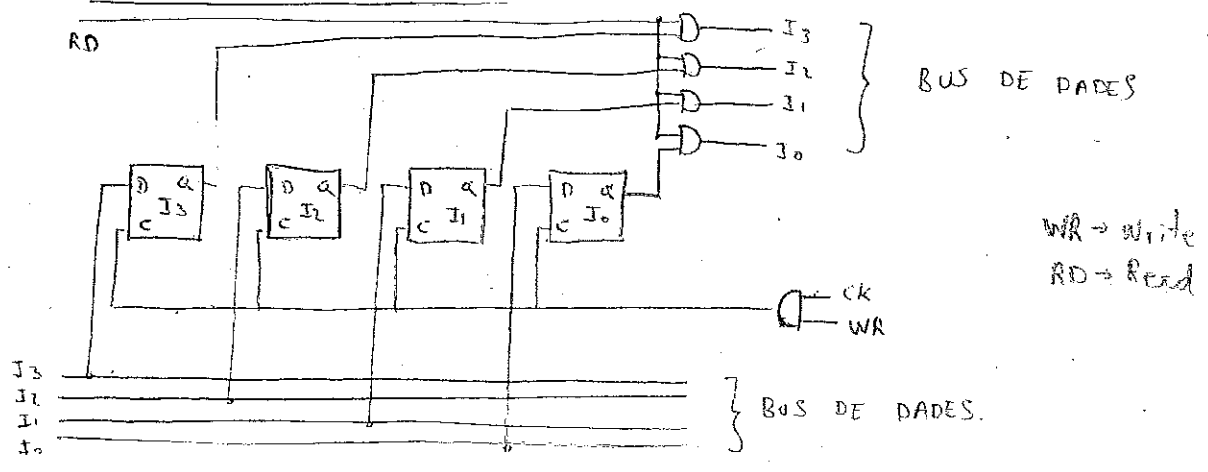
Per a evitar això, s'utilitzen els biestables MASTER-SLAVE, que emmagatzemen en un flip-flop l'estat inicial amb el senyal de rellotge alt i el traspassen quan cau el flanc a un segon flip-flop, inhibint l'entrada de dades del flip-flop inicial.

Exemple: (amb R-S.)



# 4.4. REGISTRES D'EMmagatzement.

## 4.4.1. MEMORIES RAM.



Fotocòpia TANNERBAUM, (pàg. 120.) Memòria 4x3.

Detall:	Input	Output	Control	Output
			0	(Res) → Degut a q. el bus de dades d'Entrada i Sortida és el mateix
			1	(Quan RD=0, es desactiva la sortida !!)

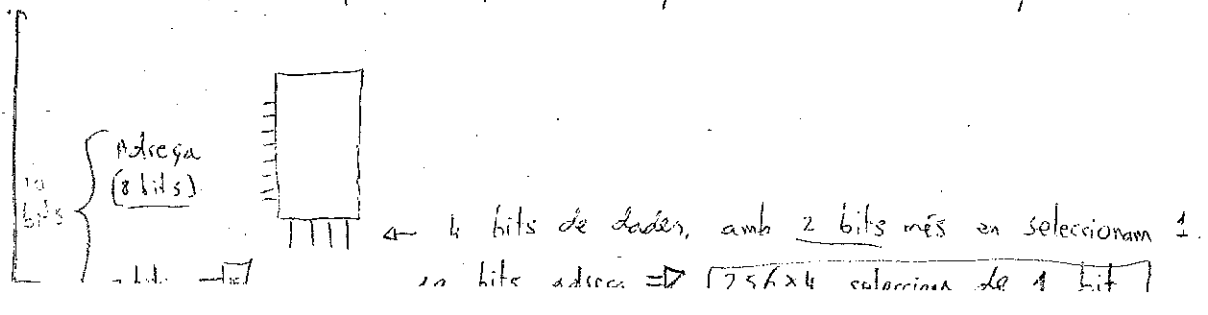
Es un dispositiu triestat: 0,1, Res (a nivell electrònic, "semblant a un transistor").

Sobre el direccionament: 0 formes de recuperar paraules:

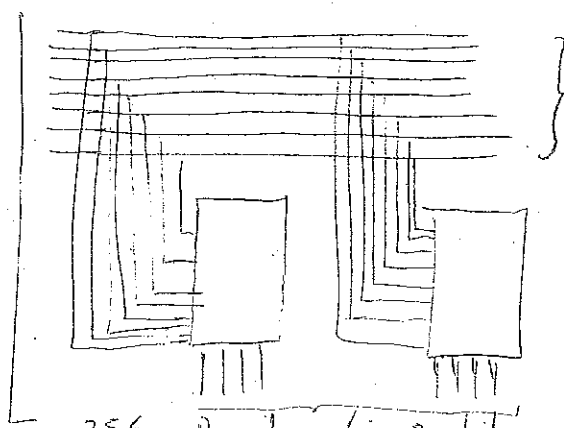
- Hem vist un "xip" de 4 paraules de 3 bits.
- les adreces han tenir:  $2^n = 4 \rightarrow n = 2$  bits.
- Si tenguéssim un xip de 16 paraules, les adreces han de tenir  $2^n = 16 \rightarrow n = 4$  bits.

El nº de bits de les adreces (longitud de l'adreça) depèn de la quantitat de paraules a adreçar.

• Amb un xip de, per exemple 256 paraules, de 4 bits, podem:



• Amb el mateix tipus de xips:

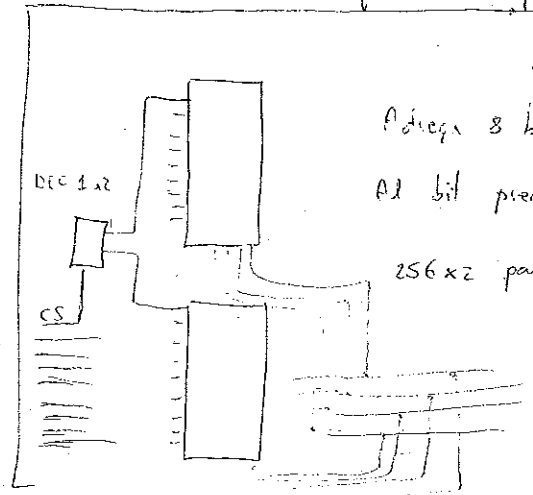


Adreça de 8 bits

256 paraules de 8 bits.

Pe → Jun codi ASCII.  
 } una adreça d'aquest mateix conjunt de xips.  
 } un valor numèric en binari natural entre 0 i 255  
 } Dos dígit hexadecimals.

• Amb el mateix tipus de xips



Adreça 8 bits + 1 astreisc = Adreça 9 bits

Al bit precedent se l'anomena Chip Select: CS.

256x2 paraules de 4 bits.

Bus de DATES (a) Bus d'ADRECES.

[RAM] Random Access Memory. Memòria d'accés aleatori (és seqüencial o altres mètodes).

- Amb bistables es construeix el que es diu [SRAM], Static-RAM. Molt ràpid accés.
- També existeix la [D-RAM] basada no en portes lògiques, sinó en condensadors.
- És més fàcil de construir (i per tant més barata) però necessita refresc elèctric permanent (i.e., el seu funcionament és més car). Més lenta

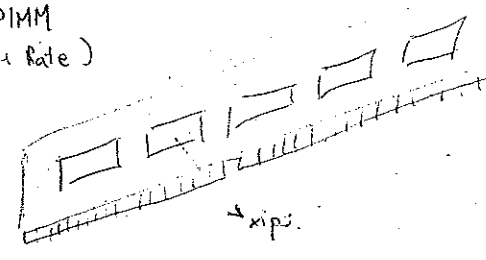
EDOP = Extended Data Output. Poden ser escrites a la vegada que llegides (no la mateixa posició) (Tècnica constructiva)

Els xips de D-RAM es solen presentar en format SIMM → DIMM

- ⊗ Single In-line Memory Module
- ⊙ Dual In-line
- ⊕ DDR-DIMM (Double Data Rate)

Solen ser EDO-RAM o DRAM.

+bits/cicle



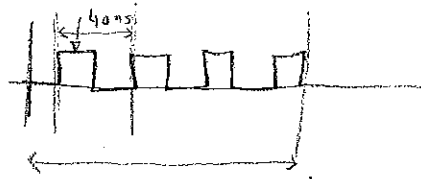
→ contactes elèctrics, (20) (72) (68)  
 s'insereixen al bus de la placa base.

xips.

## Exemple d'això darrer:

⊗ EDO-RAM de 70 ns de temps d'accés.

⊗ CPU 486 SX, a 25 MHz,



1 segon = 25 millions de cicles.

$$\text{Període} = \frac{1}{F} = \frac{1}{25 \cdot 10^6} = \frac{1}{25} \cdot 10^{-6} = \frac{100}{25} \cdot 10^{-8} = 4 \cdot 10^{-8} = 40 \cdot 10^{-9} = \text{40 ns}$$

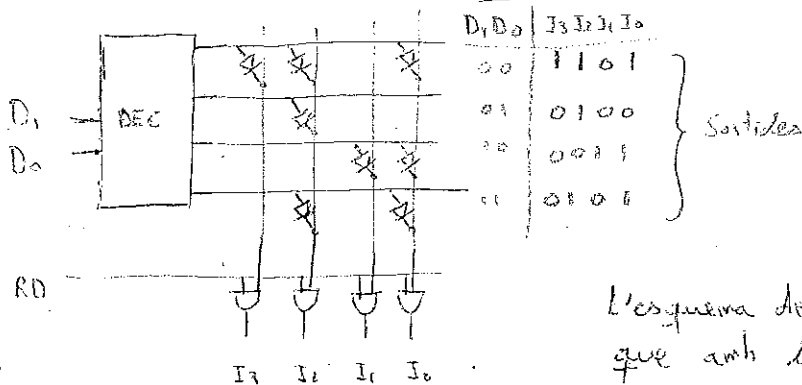
Temps d'accés EDO-RAM = 70 ns  $\Rightarrow$  necessitem 2 períodes complets. (2 "pulsos").  
(Velocitat de rellotge q. es desajusta).

## 4.4.2 MEMÒRIES ROM.

DDR-DIM, va trequant paraules cada semiperíode. (No s'ha d'esperar.)

ROM = Read Only Memory. No es poden escriure. Tan sols llegir.

Es poden construir amb diodes.



L'esquema de direccionament és el mateix que amb les memòries RAM.

## Altres tipus de ROMs:

$\rightarrow$  Així és una espècie de "màscara" sobre una matriu de 1s i 0s.

En principi, els transistors són inamovibles.

Si tenim però un circuit on les connexions són totes "1", i podem "forçar" els llucos on volem un "0", (màscara invertida), tenim una PROM Programmable-ROM.

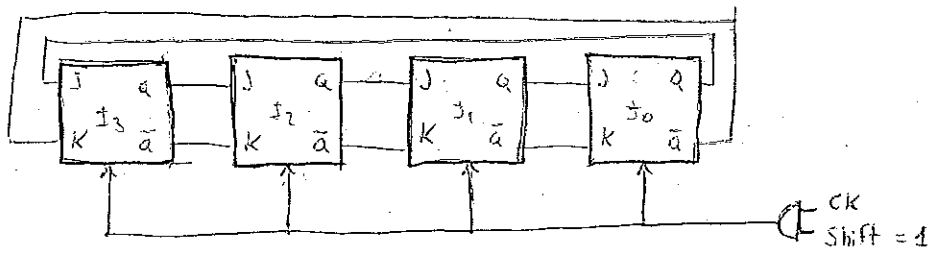
Es pot destruir. Amb raigs UV.

Existeixen també les EEPROM Erasable-Programmable ROM.

Es poden "borrar", i "re-escriure" de forma permanent.

## 4.5 REGISTRES DE DESPLAÇAMENT.

- És un número indeterminat de biestables sinans en els quals els bits emmagatzemats es poden desplaçar lateralment en un sentit o en un altre:



Exemple:

Cicle	$J_3$	$J_2$	$J_1$	$J_0$	Estat
0	0	1	0	0	Initial
1	0	0	1	0	
2	0	0	0	1	
3	1	0	0	0	
...	...	...	...	...	

Les parelles JK són 01 o 10. Sempre es passa J a la Q següent ( $J=1 \rightarrow SET$ ,  $J=0 \rightarrow RESET$ )

Tipus:

Simple (vs) Doble. { Per dins el mateix registre  
o D'un registre a un altre

→ Simple: Dreta a Esquerra, o viceversa.

→ Doble: Amunt o Avall.

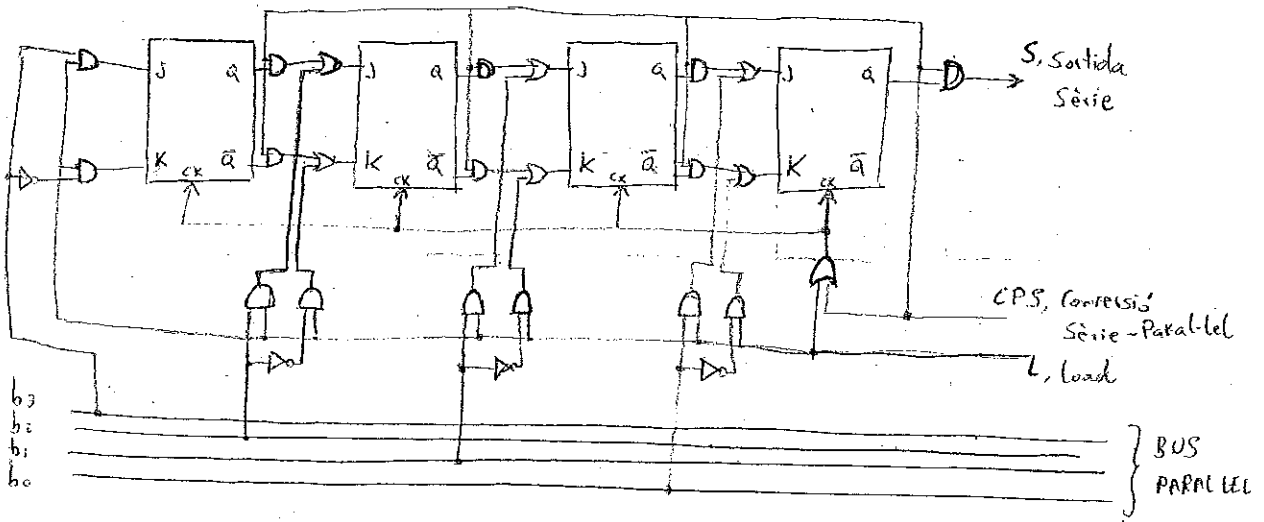
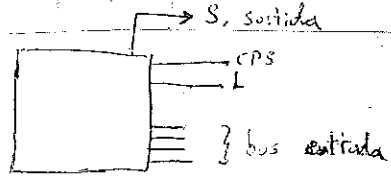
Obert (vs) Tancat: { La informació q surt per un costat entra per l'altre  
o no

L'exemple és: Simple, esquerra-dreta, tancat, 4 bits

## 4.6 CONVERSORS

- Transmissió en paral·lel: tots els bits que constitueixen una paraula viatgen simultàniament per fils diferents.
- Transmissió en sèrie: tots els bits que constitueixen una paraula viatgen un darrere de l'altre pel mateix fil.

### 4.6.1. CONVERSORS PARALLEL-SÈRIE.



#### CÀRREGA

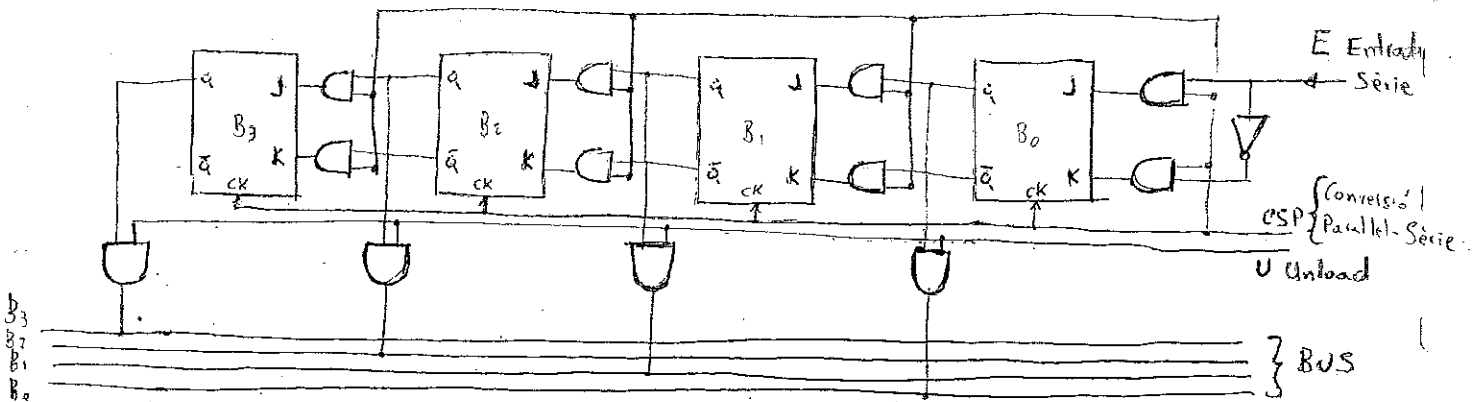
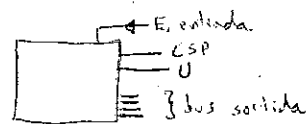
$L=1$ , (Load) a cada JK s'envia  $b_i$ . La portic OR davant cada JK permet l'entrada. Això és equivalent a un bistable D. És a dir: carregam la paraula de 4 bits  $b_3b_2b_1b_0$  que circula pel bus al registre de 4 posicions.

#### DESCÀRREGA

$CPS=1$ , (Conversió Sèrie-Paralel) S'obria les portes AND q. hi ha davant cada  $Q$  i  $\bar{Q}$ , que entien al JK següents directament. Entien parelles 01 o 10, i podem dir que cada  $Q$  pren el valor de la  $J$  anterior, (desplaçament). L'entrada CK es posa a nivell alt, tant amb  $L=1$  com  $CPS=1$ . Quan tots 2 són 1, els bistables estan "tancats".

$\downarrow$   
 1 cicle, tot dies  
 controlat per CK, n cicles per sortir tot

### 4.6.2. CONVERSORS SÈRIE-PARALLEL.



#### CÀRREGA

Necessitam 1 cicles de rellotge a  $CSP$  i per E van entrar  $b_i$ , que accedixen als JK a través de les AND com si fossen bistables D.

#### DESCÀRREGA

$U=1$ , Unload, i s'obria les 4 AND de baix, que recullen  $Q_i$

## 4.7 COMPTADORS

⊕ Un circuit comptador modifica les dades emmagatzemades en un registre de forma que el valor numèric de la configuració binària emmagatzemada augmenta o disminueix en 1.

⊕ La informació pot estar codificada en qualsevol codi binari. (Binari Natural!)

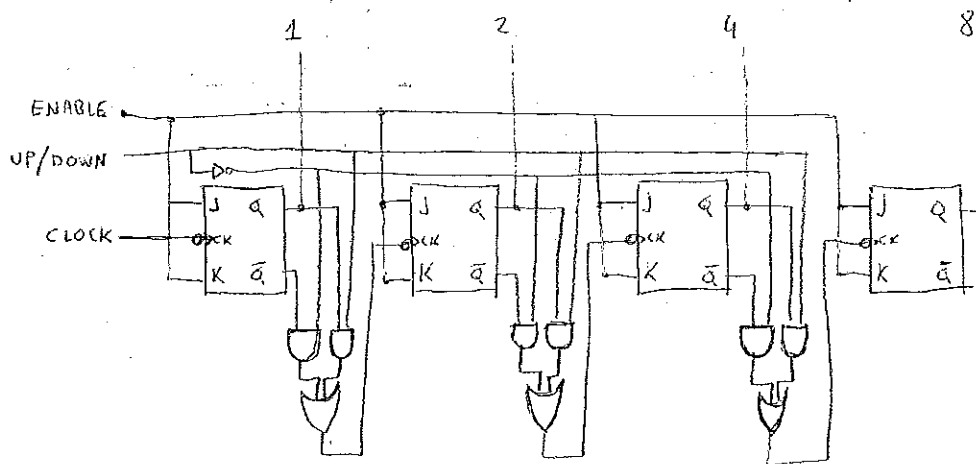
Dues funcions:

- Emmagatzemar temporalment la informació (bistables ⇒ memòria)
- Generar una seqüència progressiva de configuracions binàries a partir de la paraula emmagatzemada.

Dos grans grups:

- Asíncrons, o de propagació en sèrie.
- Síncrons, o de propagació en paral·lel.

### 4.7.1. COMPTADOR ASÍNCRON. (Exemple). LITTLE ENDIAN



→ cada bit és una potència de 2.

Comptador mòdul 16  
( $16 = 2^4$ )  
n de bistables

**ENABLE=1** vol dir que es canvia d'estat cada vegada que s'activa  $J=K=1$ .

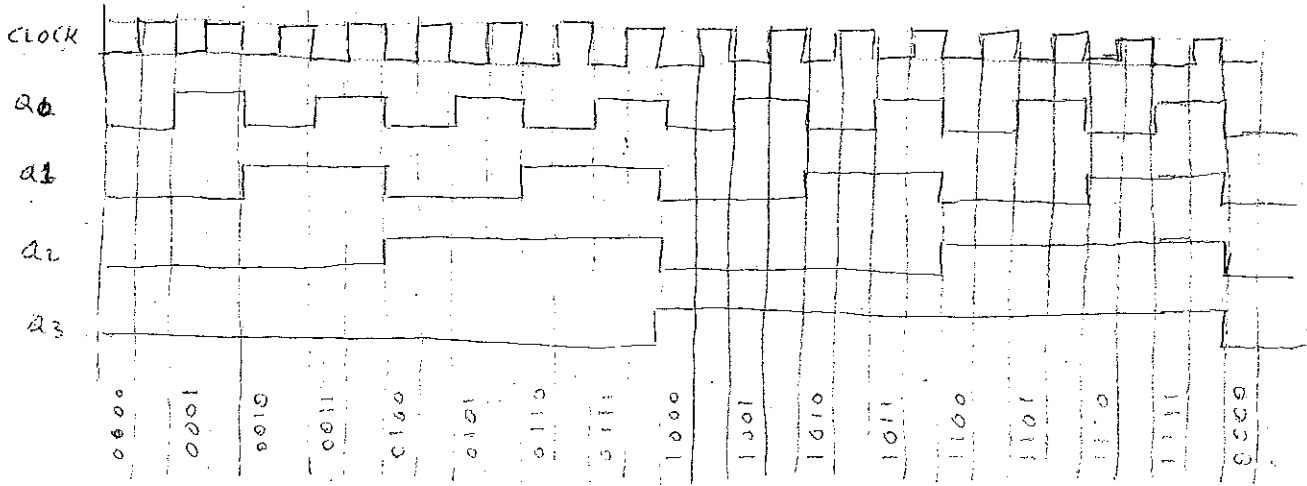
Els flip-flops van de 0 a 1 a 0... **ENABLE=0** El comptador està bloquejat.

**UP/DOWN** 1 (UP), es capturen les sortides Q, el comptador compta de forma ascendent.

0 (DOWN) es capturen les  $\bar{Q}$ , comptador descendent.

- Negació a **CLOCK** → canvi d'estat T, per flanc de baixada. Cada on ho fa per 2 de contorns (CRONOGRAMA)

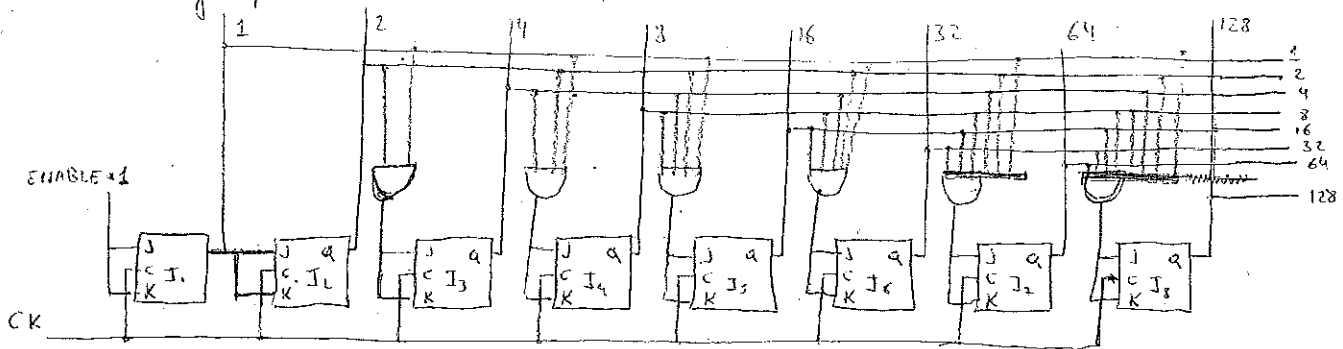
Si en els J:Ks es tenen connectades les senyals CLEAR/PRESET, fent CLEAR a tots els bistables simultàniament es "reinicialitza" el comptador.



Problema d'aquest comptador: l'entrada CK s'ha de transmetre fins al darrer bít, la qual cosa requereix períodes llargs → Freqüències baixes → **Es lent**

#### 4.7.2 COMPTADOR SINCRON (Exemple: LITTLE ENDIAN)

Amb aquest, tots els bít <sup>bits</sup> reben el senyal CK simultàniament. El cicle de rellotge pot ser més curt.



Es tracta cada JK com un T, que canvia de valor quan tots els anteriors són 1 en el mateix cicle en que tots passen a valor 0.

	J <sub>1</sub>	J <sub>2</sub>	J <sub>3</sub>	J <sub>4</sub>	...
0	0	0	0	0	← Estat inicial (els JK poden tenir PRESET/CLEAR)
1	1	0	0	0	
2	0	1	0	0	
3	1	1	0	0	
4	0	0	1	0	
5	1	1	1	0	
6	0	1	1	0	
7	1	1	1	0	

Problema d'aquest ~~circuit~~ <sup>circuit</sup>:  
 necessita moltes portes AND.  
 Per a mòduls grans és inrealitzable



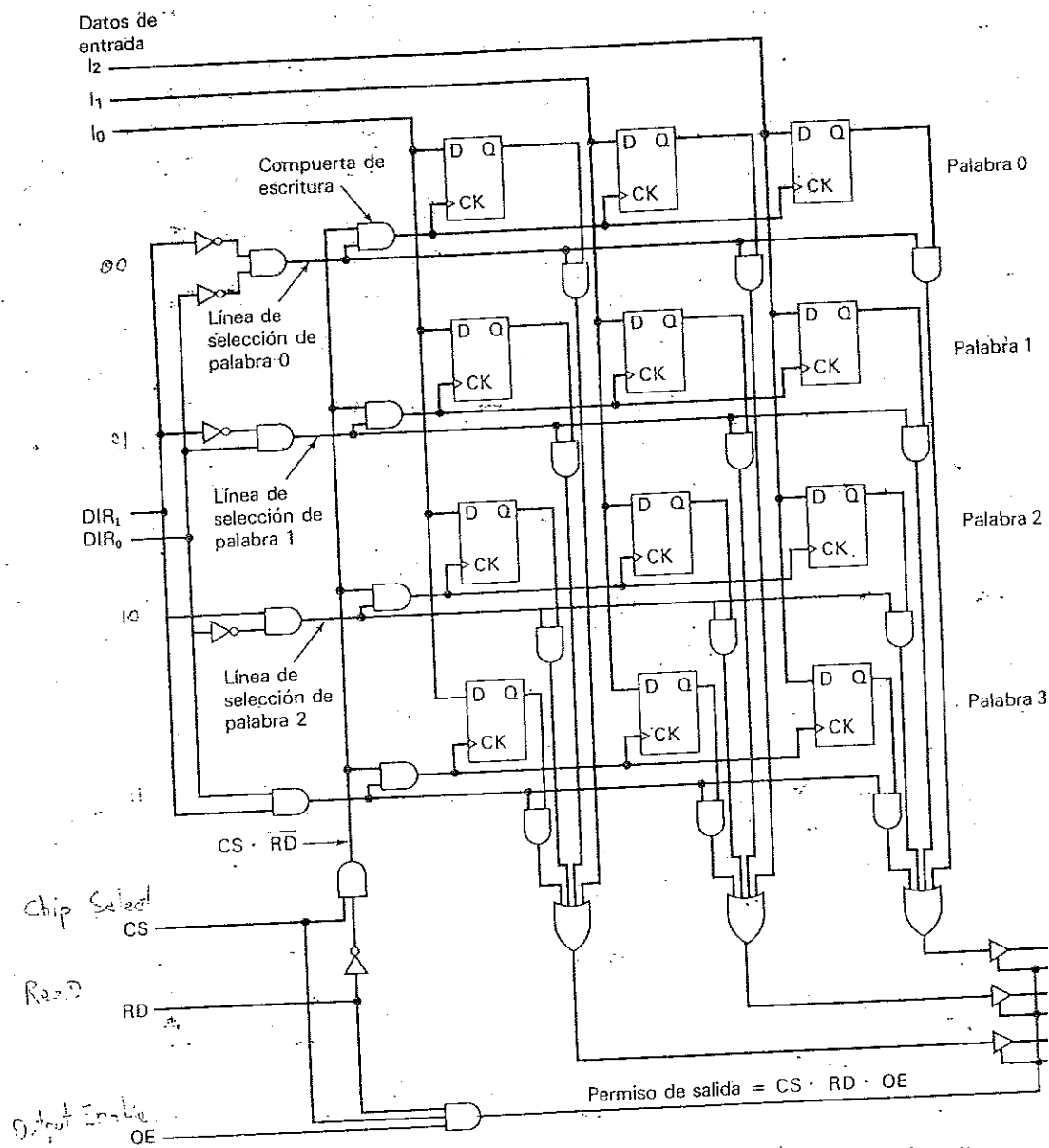


Fig. 3-28. Diagrama lógico de una memoria de  $4 \times 3$ . Cada fila es una de las cuatro palabras de tres bits. Las operaciones de lectura y escritura siempre leen o escriben una palabra completa.

**BIESTABLE R-S** Reset-Set

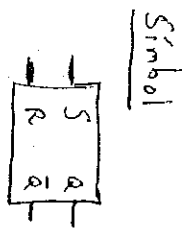
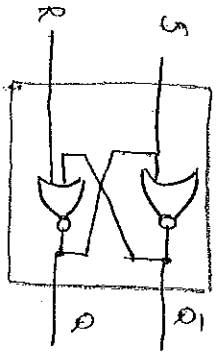
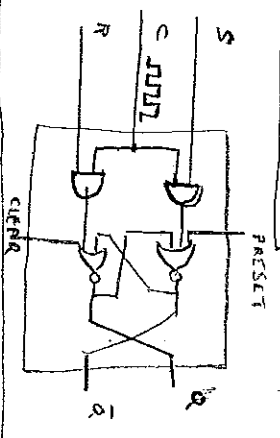


Tabla d'estats:

Reset	Set	Q	Q+
0	0	0	0
0	1	0	1
1	0	1	0
1	1	?	?

No Fa: 0 0  
 res: 0 1  
 SET: 1 0  
 RESET: 1 1  
 Indeterminat: ? ?

**Biestable RSC**



**BIESTABLE D** Data

- ⊗ RSC amb entrada D i D-bar a S i R
- ⊗ Capturam D

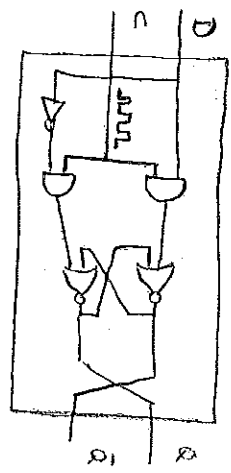


Tabla d'estats

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

Capturam 0  
 Capturam 1

Divs (S,R) entra (D,D-bar)

**BIESTABLE J-K**

- ⊗ Modificació de R,S, de forma que J=K=1 te sentit.

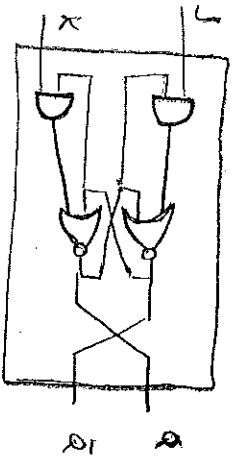


Tabla d'estats

Set	Reset	Q	Q+
0	0	0	0
0	1	0	1
1	0	1	0
1	1	0	1

No Fa: 0 0  
 res: 0 1  
 RESET: 1 0  
 SET: 1 1  
 Canvi d'estat: 1 1

\* També existeix la versió sincronitzada, "J-K-C"

**BIESTABLE T** Toggle

- ⊗ JK amb entrada T a J i K

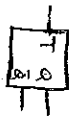
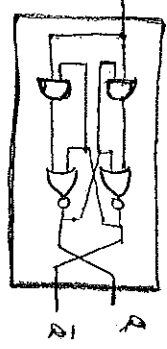


Tabla d'estats

T	Q	Q+
0	0	0
1	0	1
0	1	0
1	1	0

No Fa res: 0 0  
 Canvi d'estat: 1 0, 0 1, 1 1

\* També existeix la versió síncrona

Divs (J,K) entra (T,T)

\* També es pot implementar el D a partir de JK. Divs (J,K) entra (D,D-bar)