CIRCUITS SEQÜENCIALS ESTÀNDARDS

Índex de conceptes

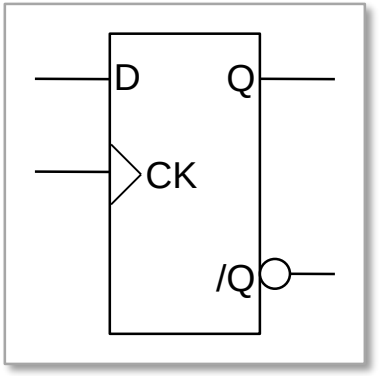
- **Estructura d'un registre**
- **MUXs**
- **Buses i Buffers**
- **DECs**
- **Registre de desplaçament**
- **Comptador asíncron**
- **Comptador síncron**

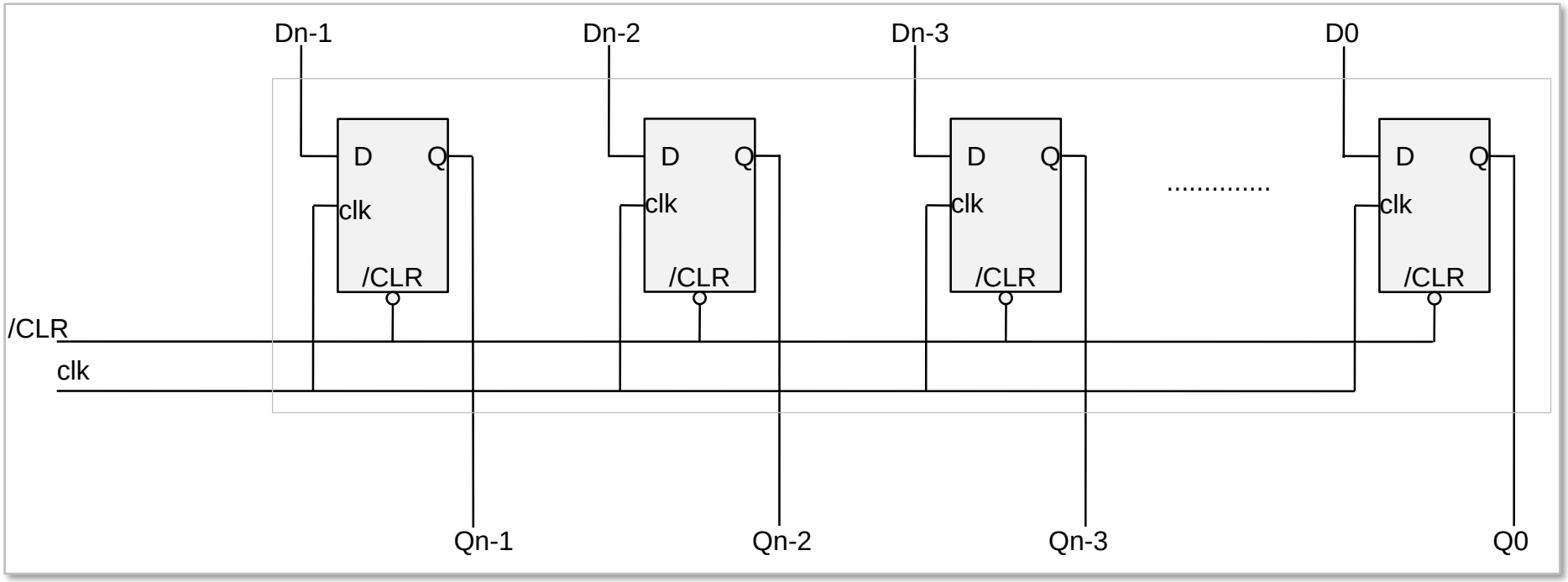
Registres

Són circuits que serveixen per emmagatzemar informació binària en forma de paraules d'n bits.

Es constitueixen per **BIESTABLES tipus D**, cadascun dels quals es pot interpretar com un registre d'1 bit (emmagatzema 1 bit), i es troben governats per el mateix *clock.* Aquest sincronisme pot estar activat per:

- Flancs: Flip-Flop edge triggered
- Nivells: Latch

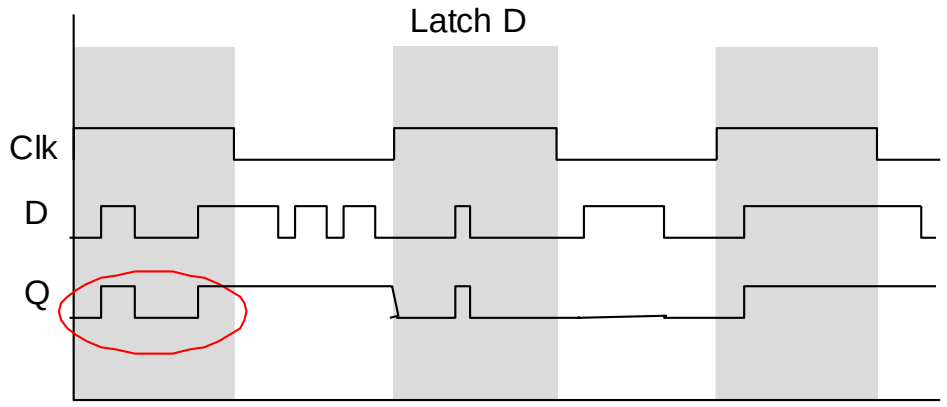




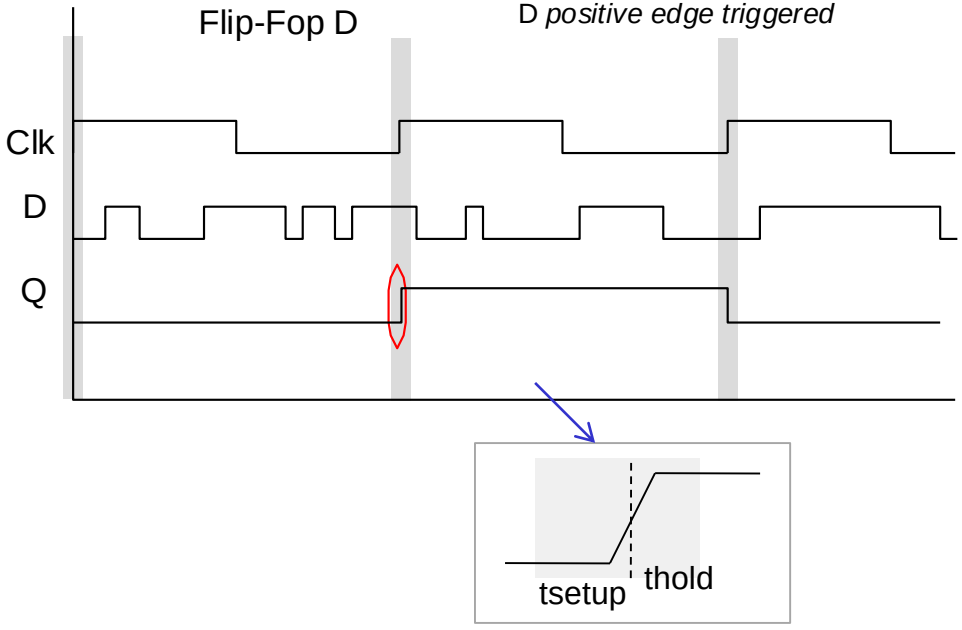
n elements = n bits

Les dades D_{n-1}-D_0 entren en paral·lel (simultàniament) i al següent flanc del senyal de rellotge aquestes dades es presenten a les sortides Q_{n-1}-Q_0. Al pols de rellotge següent es perd la informació i es carrega de nou el registre.

Latch vs. Flip-Flop



Latch
- Actiu per nivells
- Útil per guardar informació
- Asincrònic (!?)



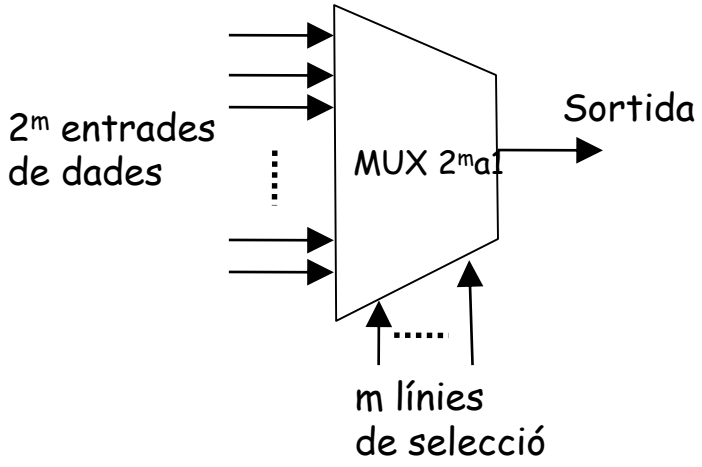
Flip-Flop
- Actiu per canvis de nivells
- Útil per guardar informació
- Sincrònic

Estructura dels Registres

Com es pot fer per mantenir la informació en un registre?

Multiplexors (MUX)

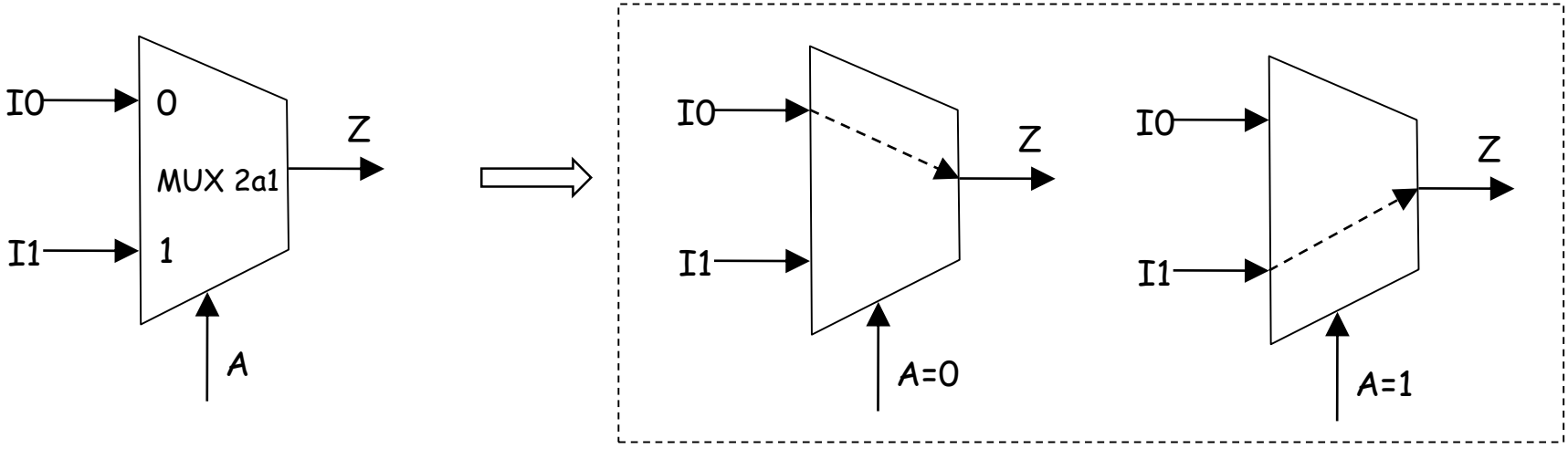
Multiplexor és un circuit combinacional amb n entrades de dades, m entrades de selecció, tal que $n=2^m$, i una sortida. Amb les entrades de selecció es selecciona el valor de l'entrada que apareixerà a la sortida.



Estructura dels Registres

Com es pot fer per mantenir la informació en un registre?

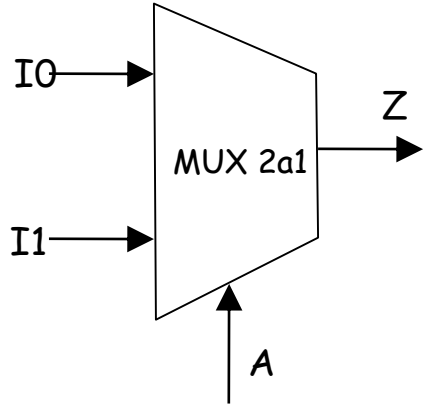
Funcionament d'un MUX



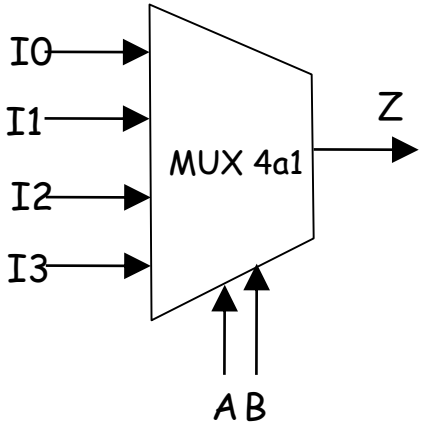
Estructura dels Registres

Com es pot fer per mantenir la informació en un registre?

Descripció de la funció lògica de sortida que genera un MUX



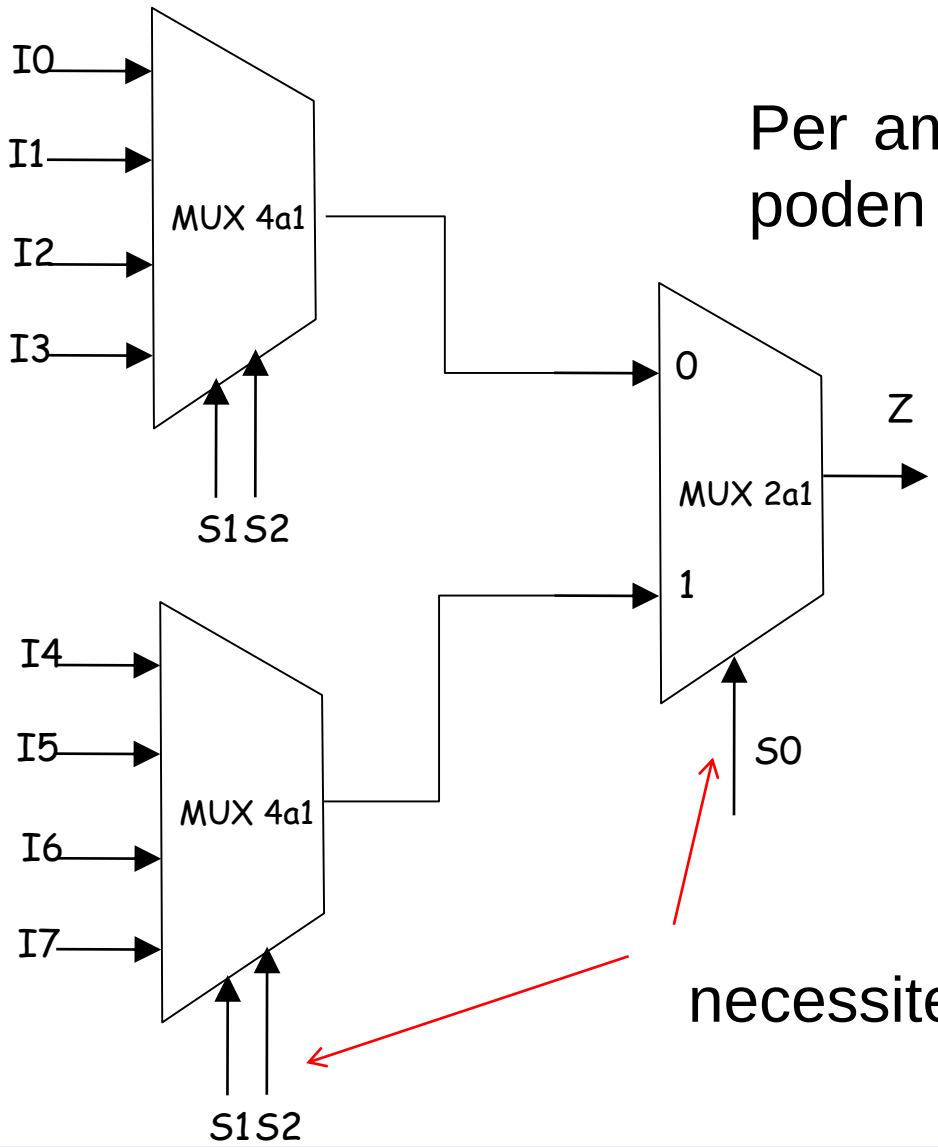
$$Z = \overline{A} \cdot I_0 + A \cdot I_1$$



$$Z = \overline{A}\,\overline{B} \cdot I_0 + \overline{A}B \cdot I_1 + A\overline{B} \cdot I_2 + AB \cdot I_3$$

Estructura dels Registres

Com es pot fer per mantenir la informació en un registre?



Per ampliar el nombre de canals es poden connectar diversos MUXs

MUX de 8 canals realitzat a partir de 2 MUX de 4 canals i un MUX de 2 canals

necessitem 3 entrades de selecció

Estructura dels Registres

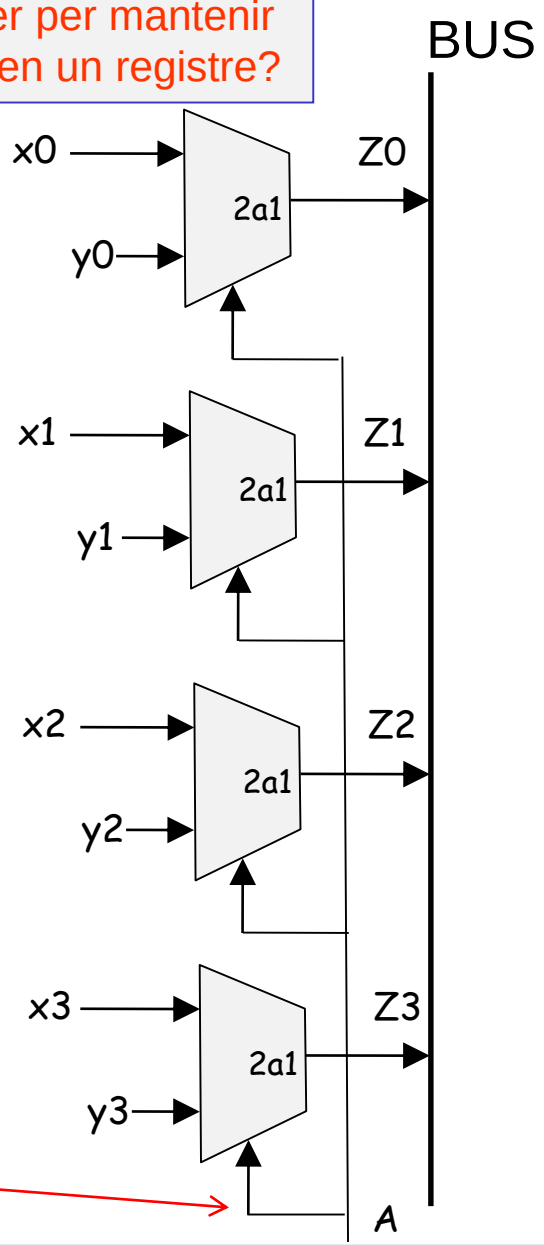
Com es pot fer per mantenir la informació en un registre?

Els MUXs es poden utilitzar per seleccionar grups de dades que es posen a un BUS

Exemple:

Selecció entre dues paraules de 4 bits X i Y
$\{x_3,x_2,x_1,x_0\}$, $\{y_3,y_2,y_1,y_0\}$

que s'envien al BUS Z, $\{Z_3,Z_2,Z_1,Z_0\}$

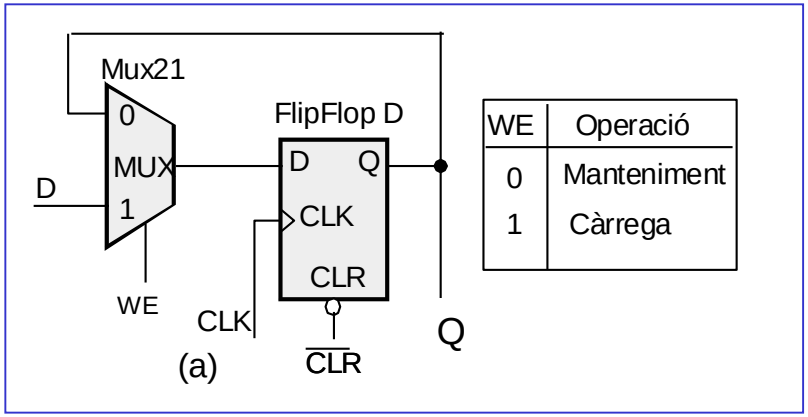


mateixa entrada de selecció 'A' de tots els MUXs

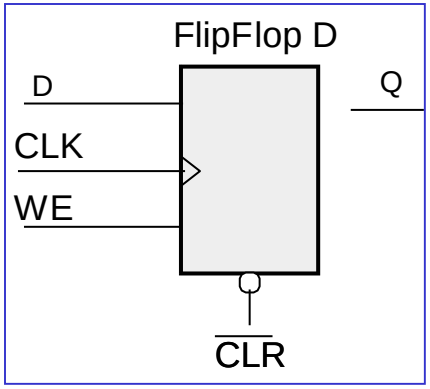
Estructura dels Registres

Com es pot fer per mantenir la informació en un registre?

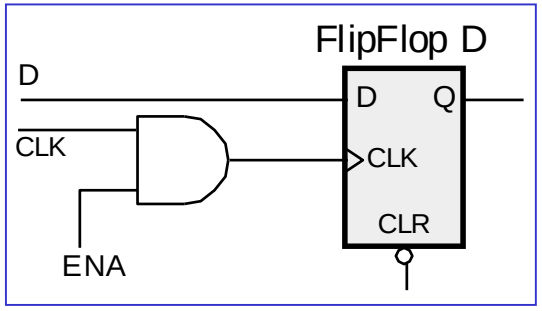
Disseny correcte: Registre governat per senyal d'habilitació d'escriptura WE (Write Enable).



WE	Operació
0	Manteniment
1	Càrrega



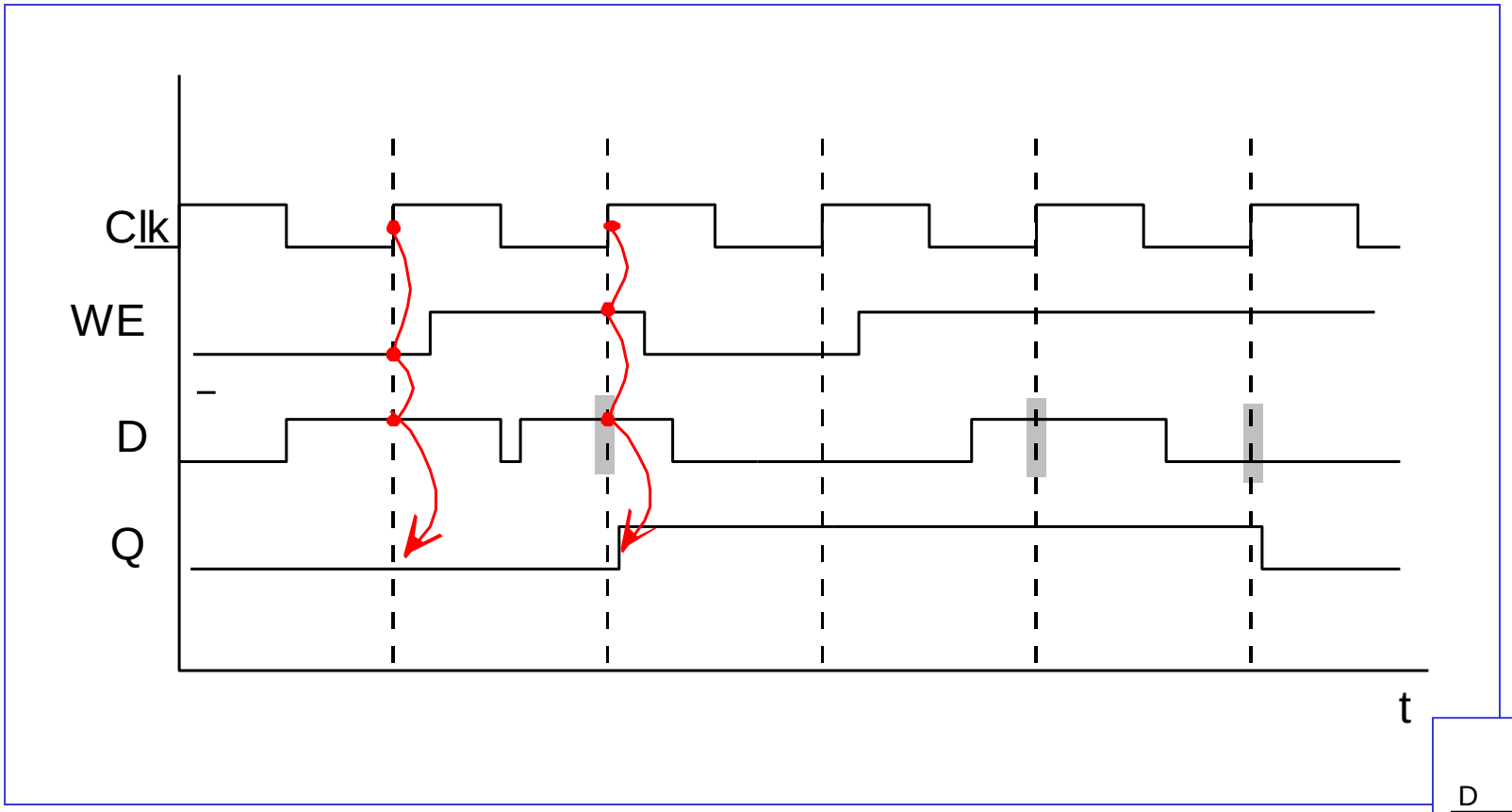
Altra solució: Gated Lock. Habilitar/inhibir el clock amb una porta AND. Habilitació actuant sobre el senyal de clock. No es aconsellable en general i s'utilitza en casos especials (exemple: reduir consum)

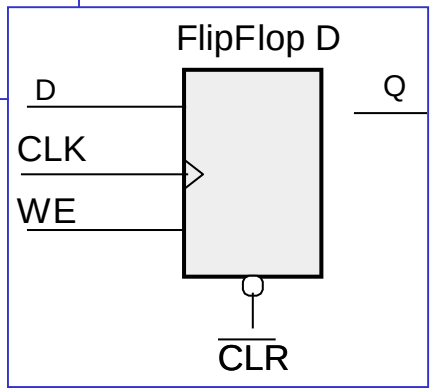


Gated clock

- Retard en clock
- Pèrdua de sincronisme

Sincronisme en Registres



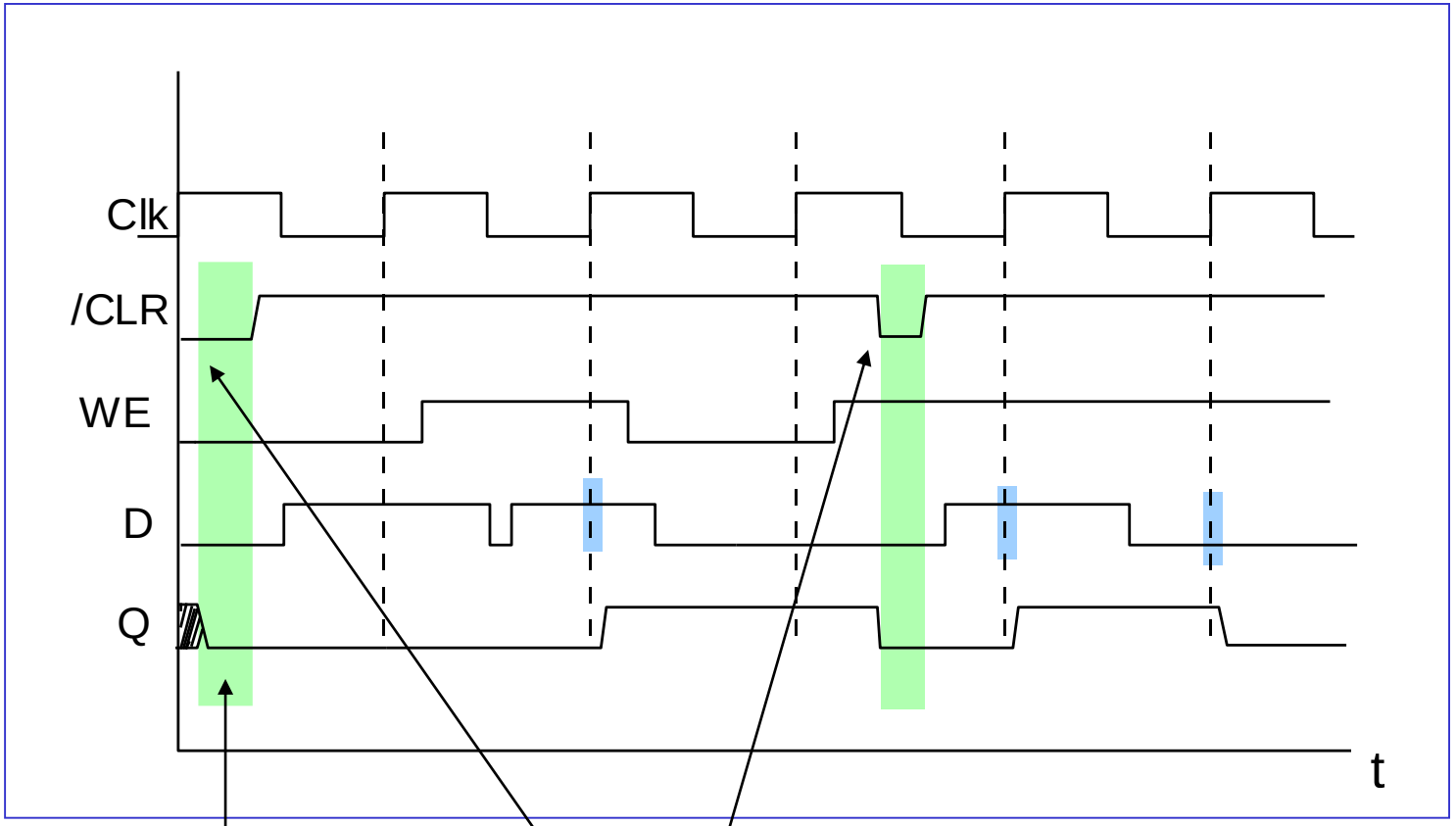


Registre governat per senyal d'habilitació (WE)

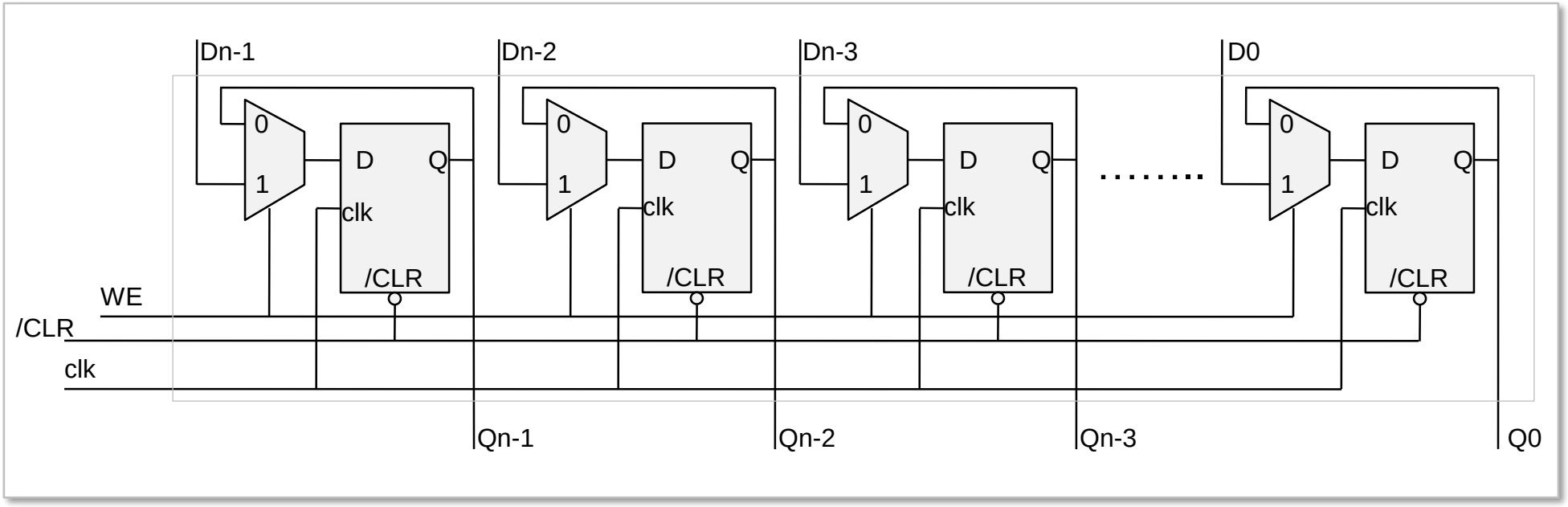
CLEAR en Registres

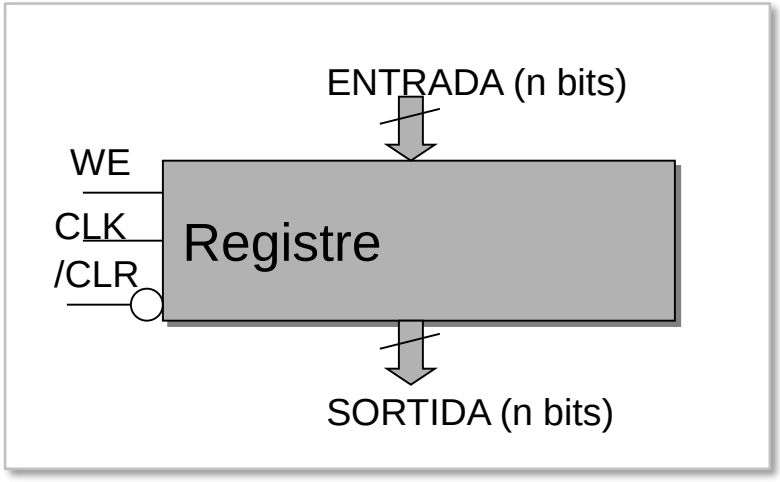
Generalment els Flip-Flops consten d'una senyal de CLEAR asíncron que permet posar a 0 tot el registre

El CLEAR sol ser *active low* (/CLR)

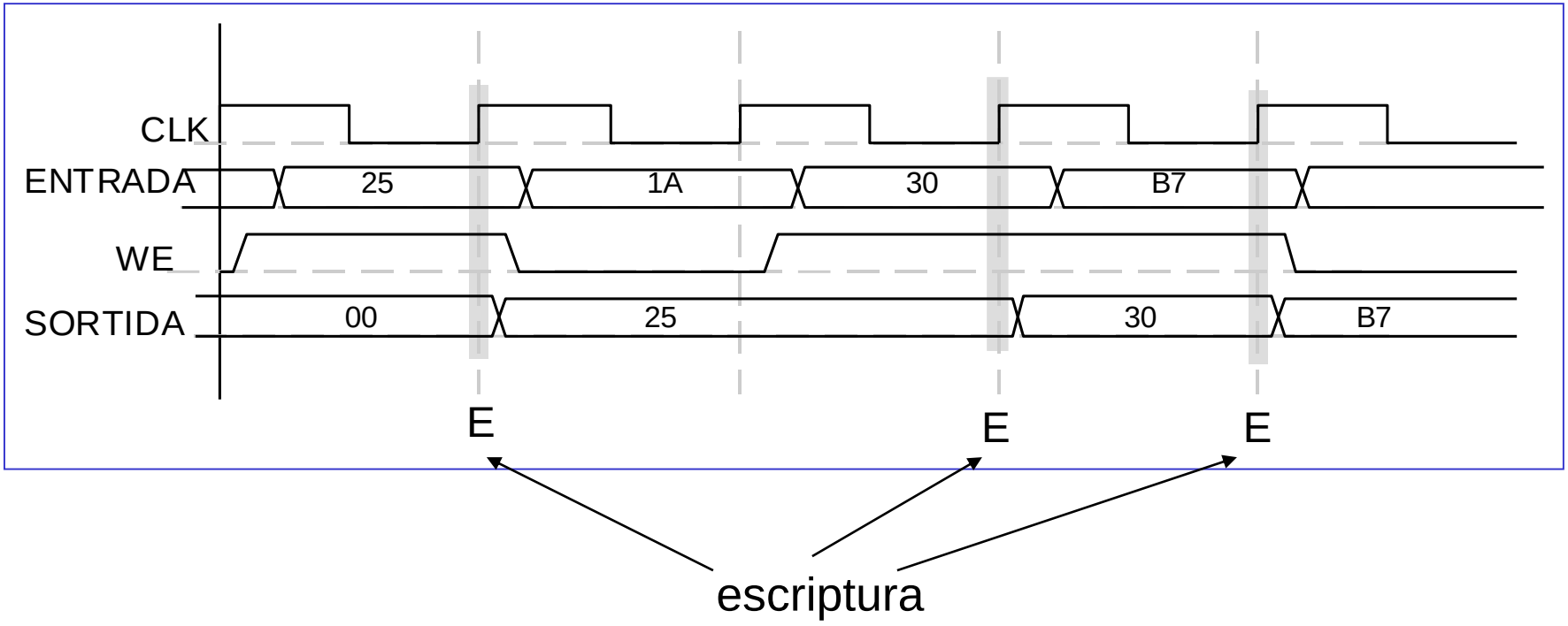


Registre d'n bits



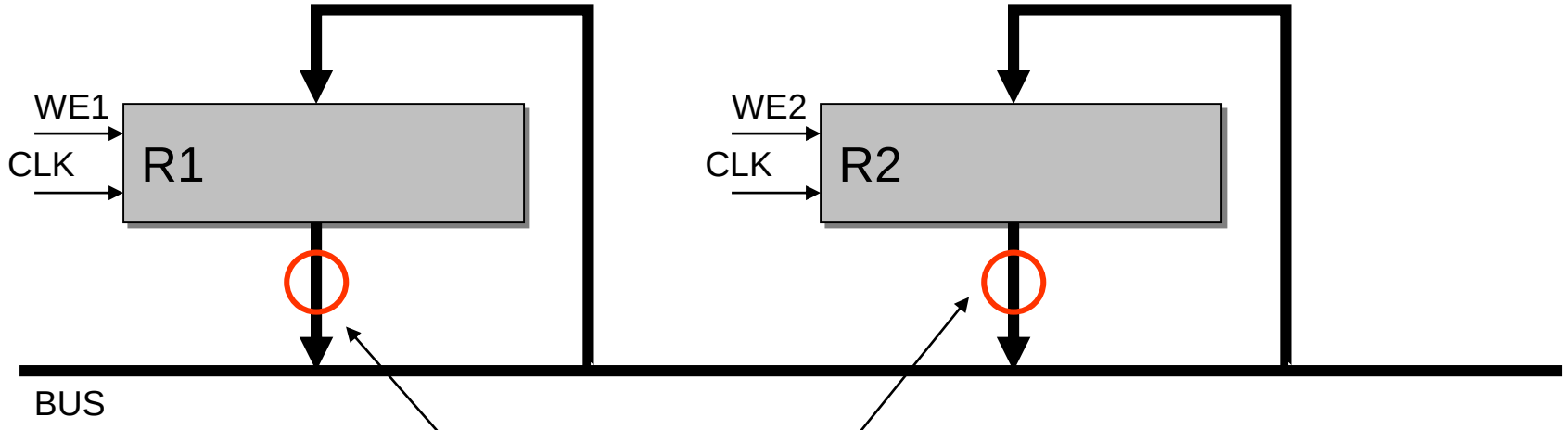


Escriptura SÍNCRONA dels Registres



Lectura de REGISTRES

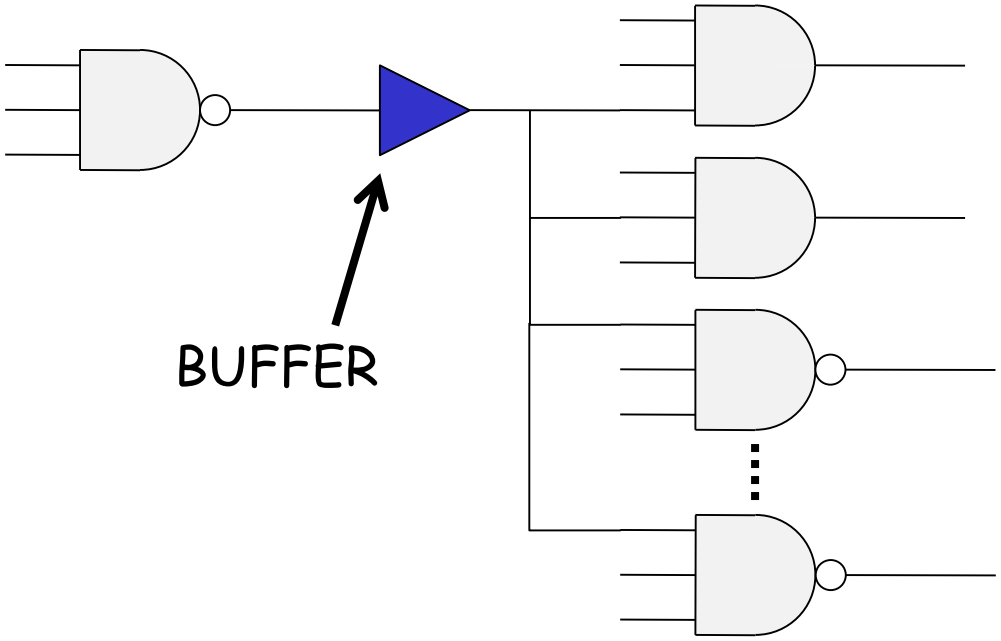
Conflictes de BUS



- L'escriptura està controlada per els senyals WE1, WE2.
- Però si es vol llegir de dos registres i passar les dades a BUS es produeix un conflicte de BUS
- Cal controlar l'accés a BUS

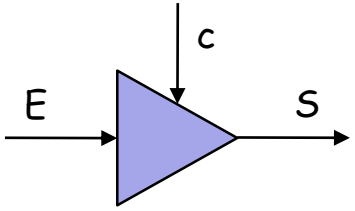
BUFFERS

- Una sortida d'un circuit lògic només es pot connectar a un número limitat d'entrades (fan-out), sense que es degradi el senyal.
- Per augmentar la connectivitat de la sortida d'un circuit es pot usar un Buffer, que augmenta la capacitat d'excitació d'una sortida.

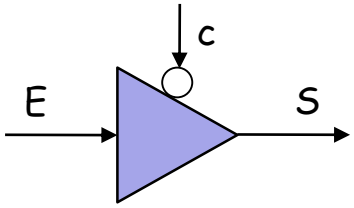


BUFFERS TriState

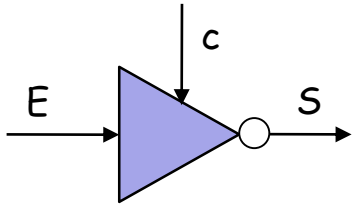
Normalment no es poden connectar dues sortides entre si, (generalment provoca un curtcircuit entre alimentació i massa i la tensió de la sortida sol quedar en un valor entre 0 i 1). Amb un buffer, si es poden connectar diverses sortides entre sí, permetent seleccionar-ne una d'elles.



c	E	S
0	0	Z
0	1	Z
1	0	0
1	1	1



c	E	S
0	0	0
0	1	1
1	0	Z
1	1	Z



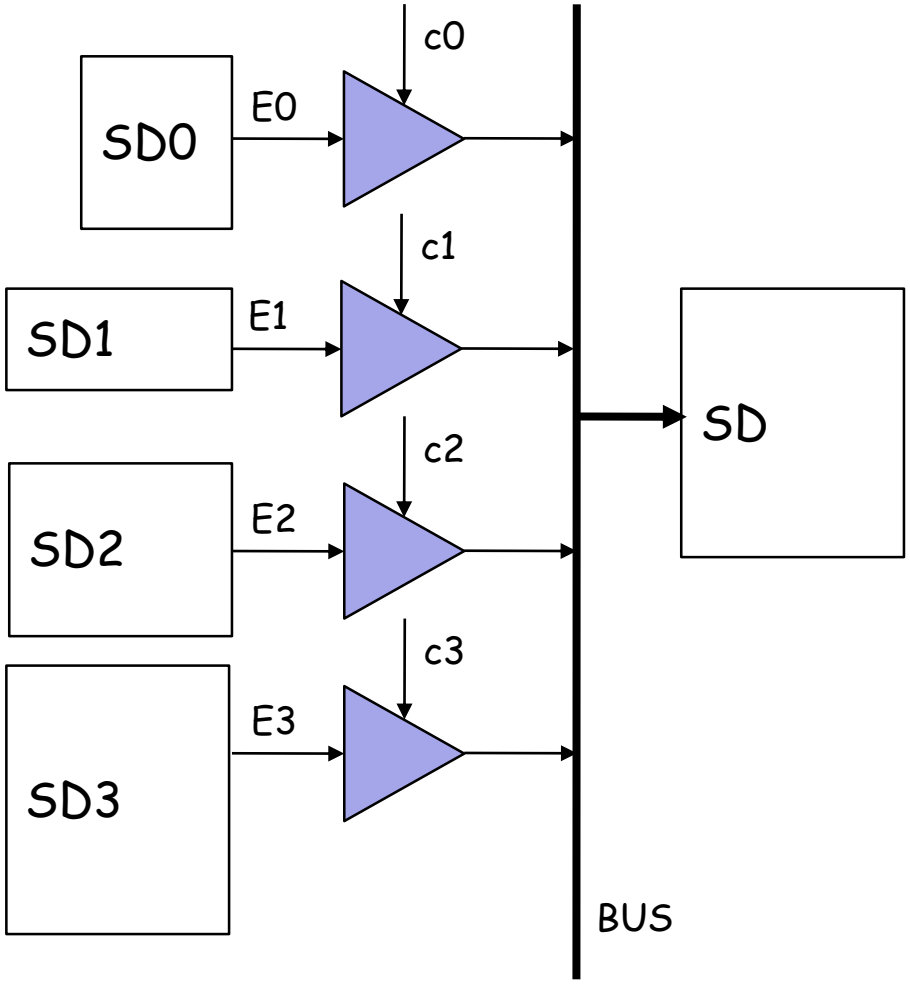
c	E	S
0	0	Z
0	1	Z
1	0	1
1	1	0

Z =Estat d'alta impedància, equivalent a desconnectat

La entrada de control c inhibeix la sortida

Lectura de REGISTRES

Utilitat BUFFERS TriState: Connexió de diversos sistemes digitals a un **BUS** comú

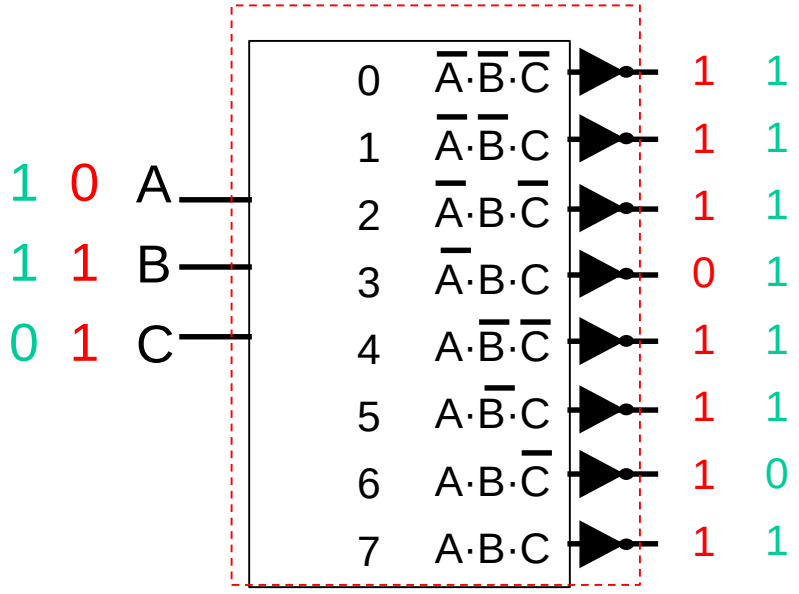


Com es controlen adequadament les c's ?

Descodificadors (DEC)

Són circuits combinacionals que **generen els productes canònics d'una combinació binària aplicada a les seves entrades**. Tenen **n entrades** i **2^n sortides**

Exemple de descodificador de 3 a 8 línies amb sortida **active-low**



Lectura de REGISTRES

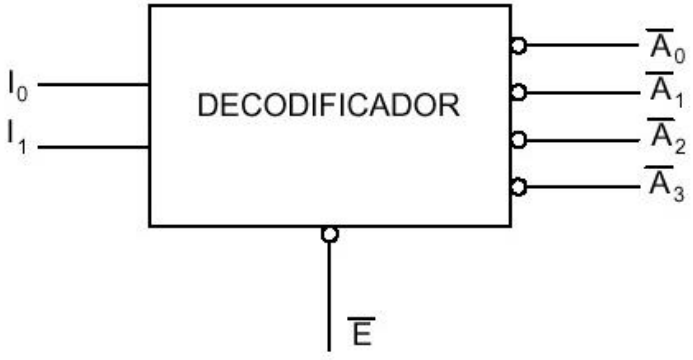
Alguns dels diferents tipus més comuns:

- De 3 a 8 línies: descodificador binari natural
- De 4 a 16 línies: descodificador binari natural
- De 4 a 10 línies: descodificador BCD–decimal
- Hexadecimal a 7 segments

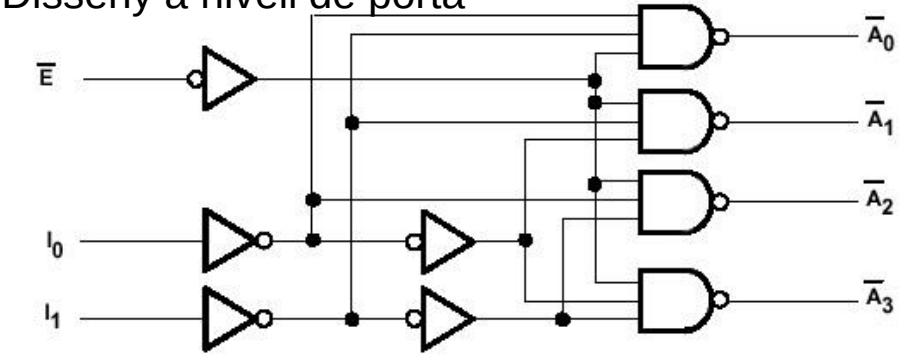
Exemple de descodificador de 2 a 4 línies amb sortida active-low i funció Enable (active-low)

Taula de funcionament

$\overline{E}$	I_1	I_0	$\overline{A}_0$	$\overline{A}_1$	$\overline{A}_2$	$\overline{A}_3$
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0



Disseny a nivell de porta



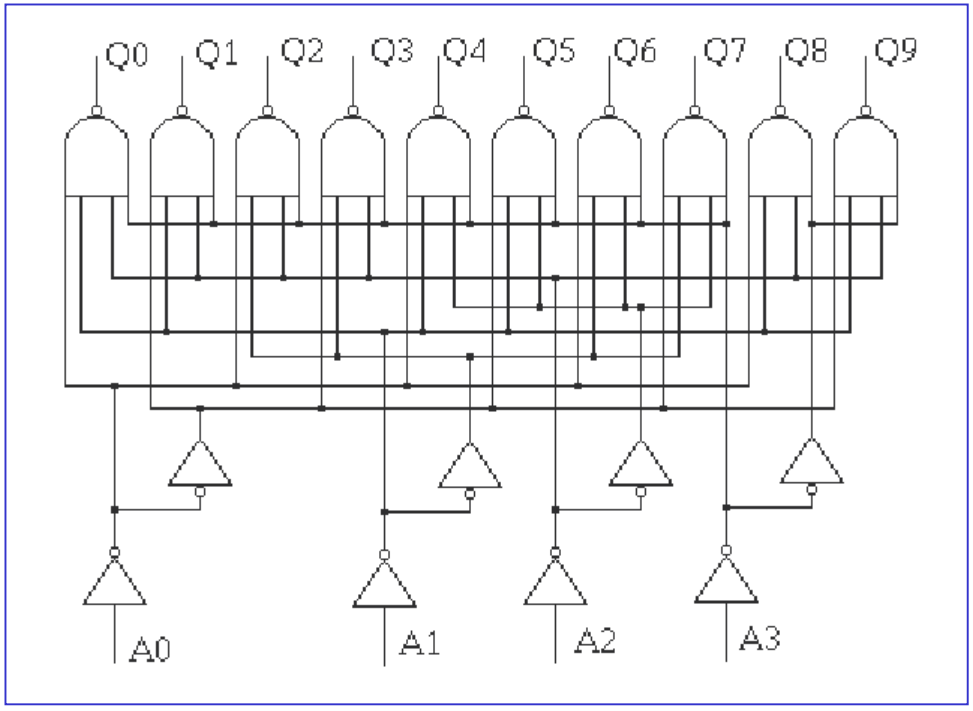
Lectura de REGISTRES

Alguns dels diferents tipus més comuns:

- De 3 a 8 línies: descodificador binari natural
- De 4 a 16 línies: descodificador binari natural
- De 4 a 10 línies: descodificador BCD–decimal (amb sortida active-low)
- Hexadecimal a 7 segments

Entrades $A_3\ A_2\ A_1\ A_0$	Sortides $Q_0\ Q_1\ Q_2\ Q_3\ Q_4\ Q_5\ Q_6\ Q_7\ Q_8\ Q_9$
0 0 0 0	0 1 1 1 1 1 1 1 1 1
0 0 0 1	1 0 1 1 1 1 1 1 1 1
0 0 1 0	1 1 0 1 1 1 1 1 1 1
0 0 1 1	1 1 1 0 1 1 1 1 1 1
0 1 0 0	1 1 1 1 0 1 1 1 1 1
0 1 0 1	1 1 1 1 1 0 1 1 1 1
0 1 1 0	1 1 1 1 1 1 0 1 1 1
0 1 1 1	1 1 1 1 1 1 1 0 1 1
1 0 0 0	1 1 1 1 1 1 1 1 0 1
1 0 0 1	1 1 1 1 1 1 1 1 1 0
1 0 1 0	1 1 1 1 1 1 1 1 1 1
1 0 1 1	1 1 1 1 1 1 1 1 1 1
1 1 0 0	1 1 1 1 1 1 1 1 1 1
1 1 0 1	1 1 1 1 1 1 1 1 1 1
1 1 1 0	1 1 1 1 1 1 1 1 1 1
1 1 1 1	1 1 1 1 1 1 1 1 1 1

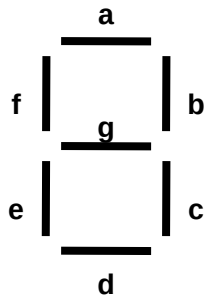
Disseny a nivell de porta



Lectura de REGISTRES

Alguns dels diferents tipus més comuns:

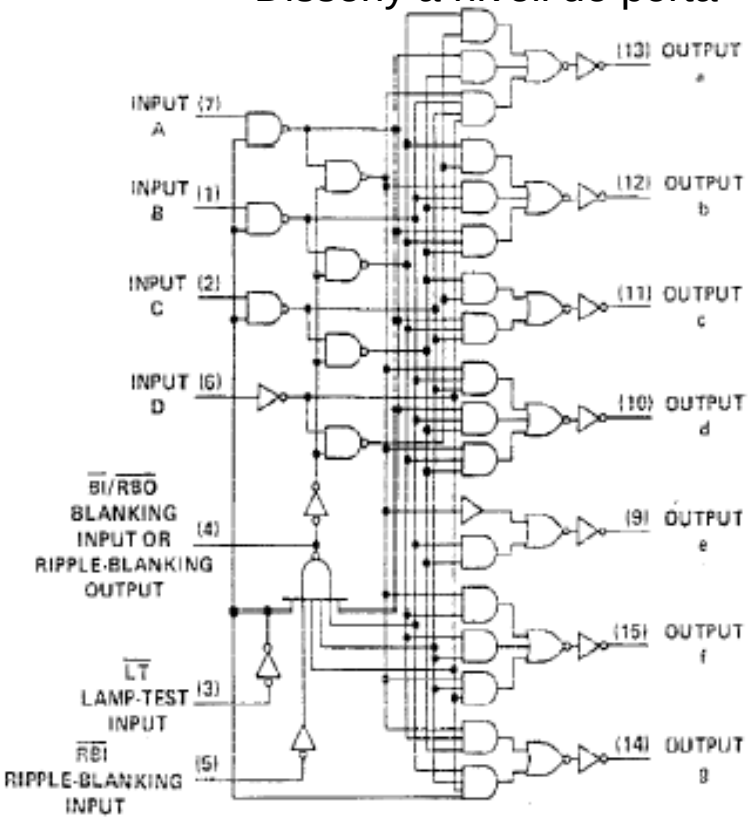
- De 3 a 8 línies: descodificador binari natural
- De 4 a 16 línies: descodificador binari natural
- De 4 a 10 línies: descodificador BCD–decimal
- Hexadecimal a 7 segments



poden activar vàries sortides simultàniament

hex	Inputs	Outputs						
	A B C D	a	b	c	d	e	f	g
0	L L L L	ON	ON	ON	ON	ON	ON	OFF
1	L L L H	OFF	ON	ON	OFF	OFF	OFF	OFF
2	L L H L	ON	ON	OFF	ON	ON	OFF	ON
3	L L H H	ON	ON	ON	ON	OFF	OFF	ON
4	L H L L	OFF	ON	ON	OFF	OFF	ON	ON
5	L H L H	ON	OFF	ON	ON	OFF	ON	ON
6	L H H L	ON	OFF	ON	ON	ON	ON	ON
7	L H H H	ON	ON	ON	OFF	OFF	OFF	OFF
8	H L L L	ON	ON	ON	ON	ON	ON	ON
9	H L L H	ON	ON	ON	OFF	OFF	ON	ON
A	H L H L	ON	ON	ON	OFF	ON	ON	ON
B	H L H H	OFF	OFF	ON	ON	ON	ON	ON
C	H H L L	OFF	OFF	OFF	ON	ON	OFF	ON
D	H H L H	OFF	ON	ON	ON	ON	OFF	ON
E	H H H L	ON	OFF	OFF	ON	ON	ON	ON
F	H H H H	ON	OFF	OFF	OFF	ON	ON	ON

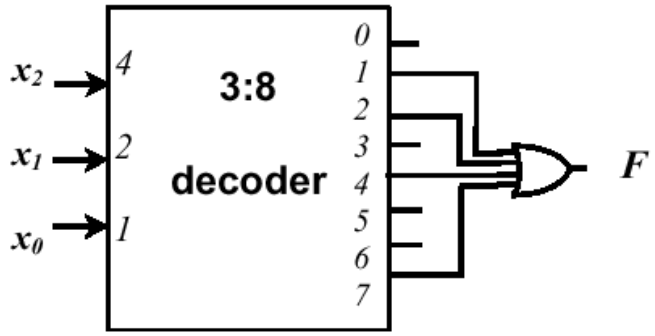
Disseny a nivell de porta



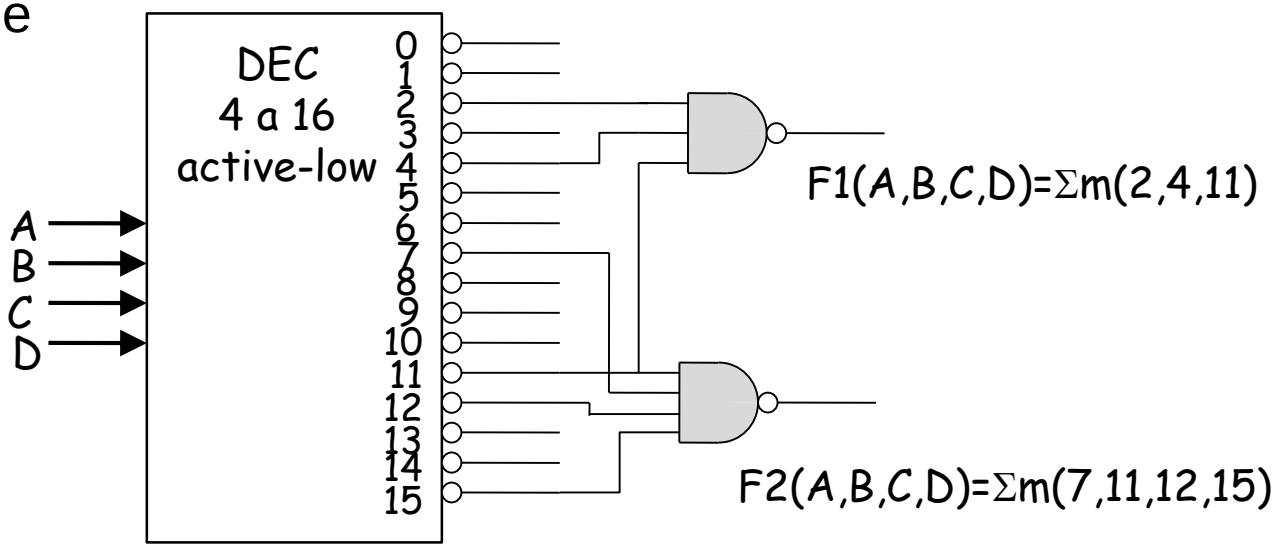
Realització de funcions lògiques mitjançant descodificadors.

Donat que un DEC active-high genera tots els minterms, es poden implementar funcions lògiques amb un DEC i una porta OR amb les entrades connectades als minterms de la funció.

$F(x_2,x_1,x_0)=\Sigma m(1,2,4,7)$



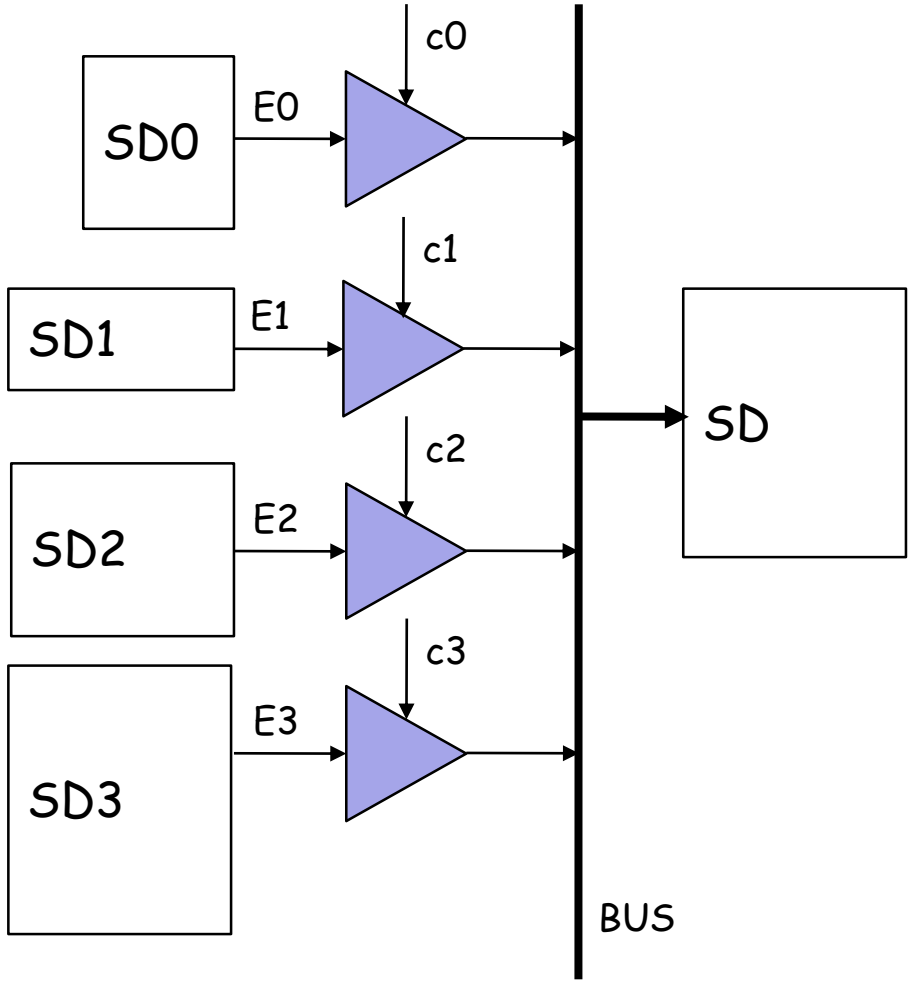
I si el DEC proporciona les sortides negades (active-low) es col·loca una porta NAND (utlizant els minterms)



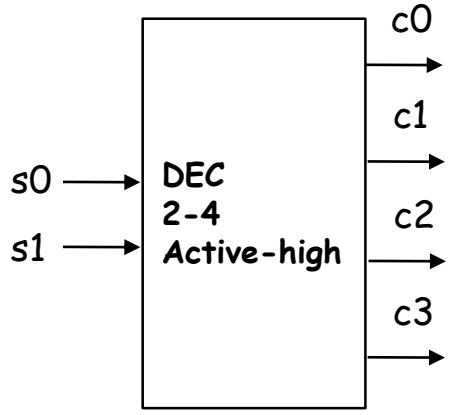
Es poden implementar *diferents funcions* alhora

Lectura de REGISTRES

Utilitat BUFFERS TriState: Connexió de diversos sistemes digitals a un **BUS** comú

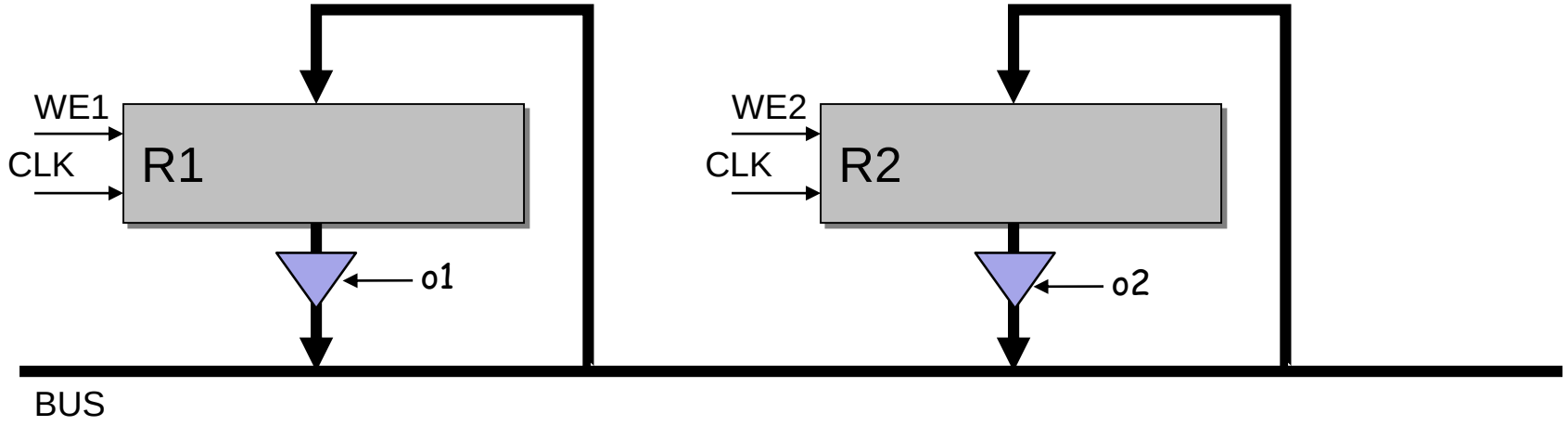


Generalment les entrades de control provenen d'un Descodificador (DEC), perquè només una de les seves sortides pot estar activada



Lectura de REGISTRES

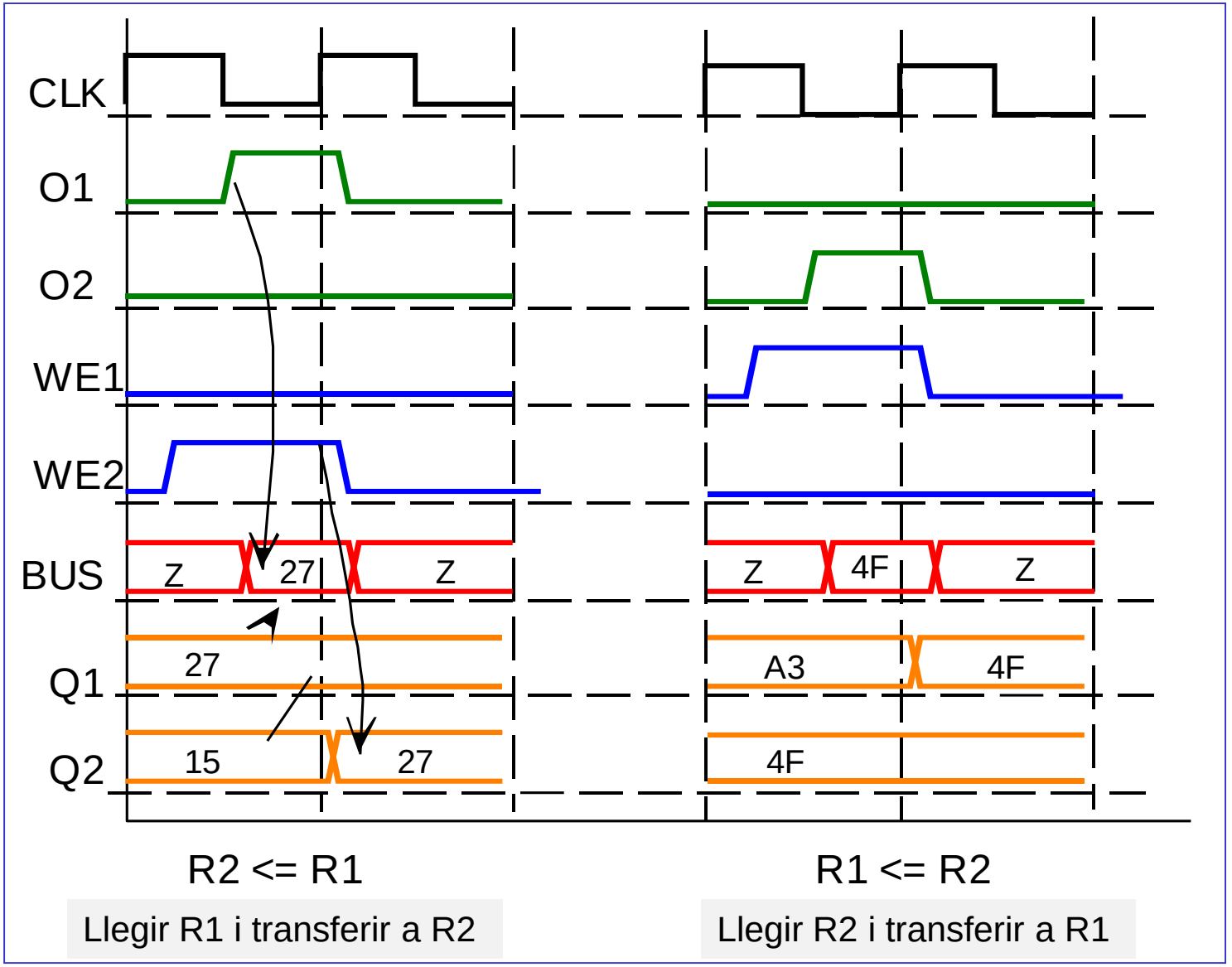
Conflictes de BUS



- L'escriptura està controlada per els senyals WE1, WE2.
- Però si es vol llegir de dos registres i passar les dades a BUS es produeix un conflicte de BUS

Exemple

Transferència de dades entre registres



Registre de DESPLAÇAMENT (*Shift Register*)

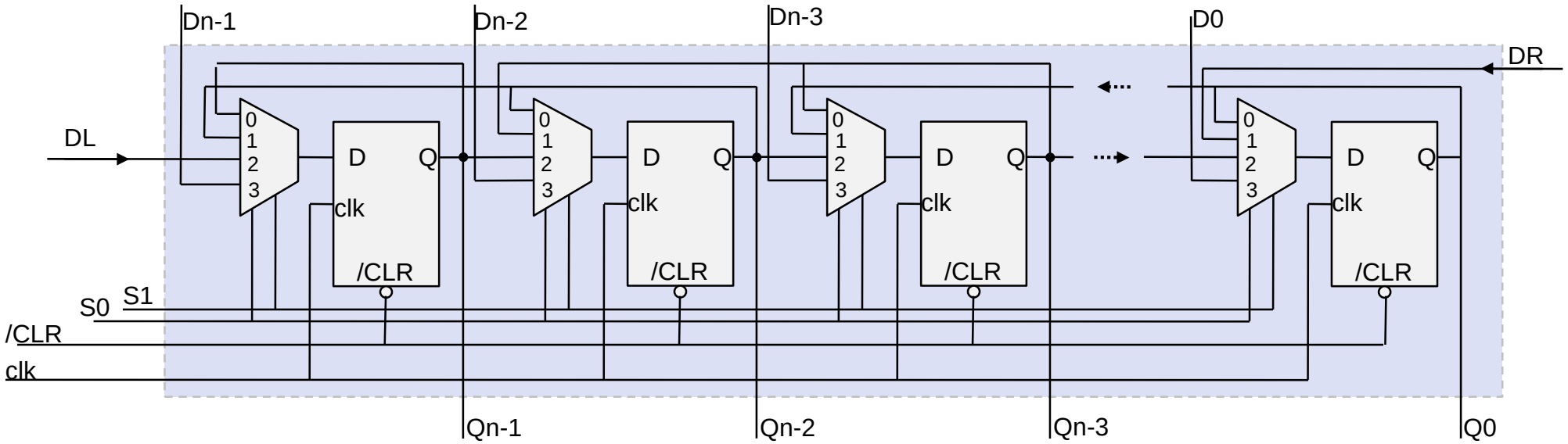
Son registres en els quals la informació que contenen es pot ***desplaçar*** cap a la dreta o cap a la esquerra dins del mateix registre. La seva principal utilitat és la transferència de dades de format **paral·lel** a format **sèrie** i viceversa (com a exemple en un ***USB pen drive***).

Selecció S1 S0	**Operació**		**Senyals de Sortida** Q_{n-1} Q_{n-2} ..... Q_1 Q_0
0 0	Mantenir	HOLD	Q_{n-1} Q_{n-2} ..... Q_1 Q_0
0 1	Despl. esquerra	SHL	Q_{n-2} Q_{n-3} ..... Q_0 DR
1 0	Despl. Dreta	SHR	DL Q_{n-1} ... Q_2 Q_1
1 1	Càrrega	LOAD	D_{n-1} D_{n-2} ..... D_1 D_0

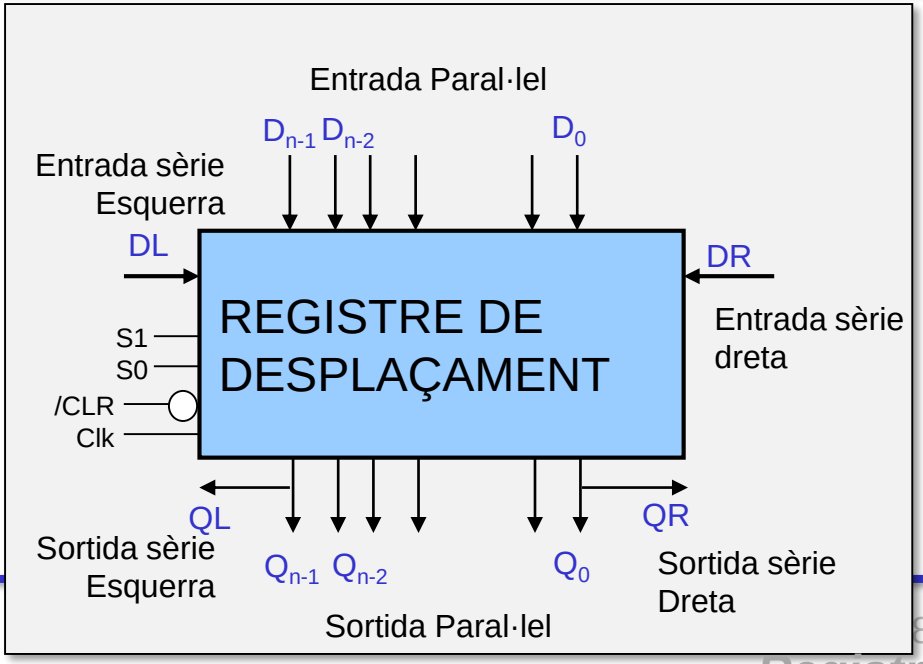
Tenen també utilitat aritmètica:

- Desplaçar a la dreta = dividir per 2 (amb DL=0)
- Desplaçar a l'esquerra = multiplicar per 2 (amb DR=0)

Registre de DESPLAÇAMENT (*Shift Register*)

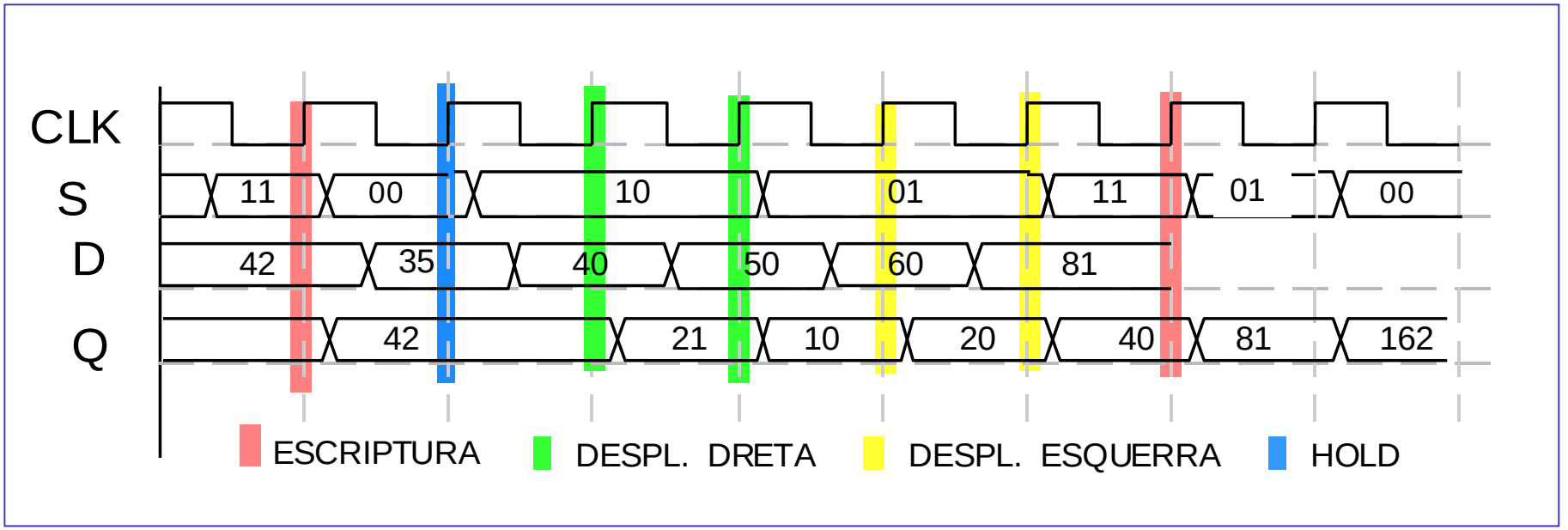


Selecció S1	S0	Operació		Senyals de Sortida Q_{n-1} Q_{n-2} ..... Q_1 Q_0
0	0	Mantenir	HOLD	Q_{n-1} Q_{n-2} ..... Q_1 Q_0
0	1	Despl. esquerra	SHL	Q_{n-2} Q_{n-3} ..... Q_0 DR
1	0	Despl. Dreta	SHR	DL Q_{n-1} ... Q_2 Q_1
1	1	Càrrega	LOAD	D_{n-1} D_{n-2} ..... D_1 D_0



Exemple

Funcionament d'un Registre de Desplaçament

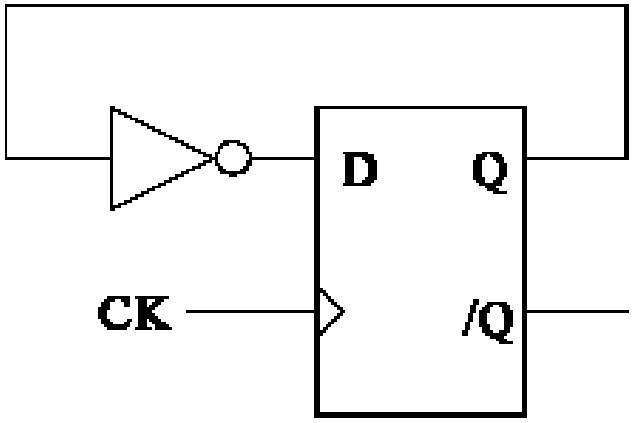


Selecció S1 S0	Operació		Senyals de Sortida Q_{n-1} Q_{n-2} ..... Q_1 Q_0
0 0	Mantenir	HOLD	Q_{n-1} Q_{n-2} ..... Q_1 Q_0
0 1	Despl. Esquerra	SHL	Q_{n-2} Q_{n-3} ..... Q_0 DR
1 0	Despl. Dreta	SHR	DL Q_{n-1} ... Q_2 Q_1
1 1	Càrrega	LOAD	D_{n-1} D_{n-2} ..... D_1 D_0

Comptadors

Comptador és un circuit que genera una seqüència predefinida d'estats, de manera que el circuit canvia el seu estat quan rep un pols de rellotge i l'estat futur només depèn de l'estat actual.

Si el comptador reinicia la seqüència al cap de K cicles de rellotge, es diu que és un **comptador mòdul K**. Si $K=2^n$ (on n és el número de FF) el comptador és **binari** (amb 1 FF tenim 2 estats 0,1, i amb n FF tenim 2^n estats possibles).



Comptador binari mòdul 2: la seqüència, partint del valor inicial 1, és: 1,0,1,0,1,0,1,...

Comptadors

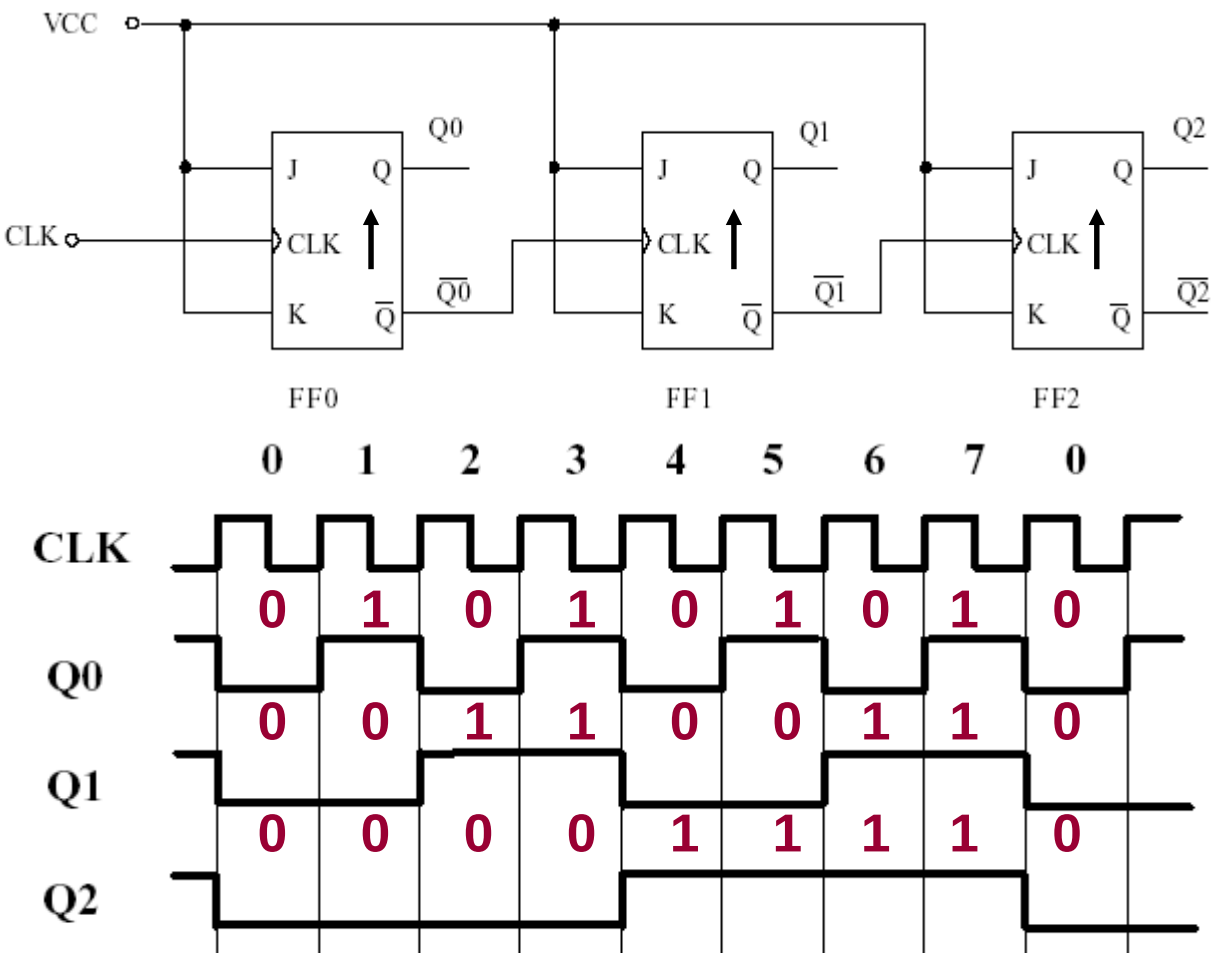
Comptador és un circuit que genera una seqüència predefinida d'estats, de manera que el circuit canvia el seu estat quan rep un pols de rellotge i l'estat futur només depèn de l'estat actual.

Si el comptador reinicia la seqüència al cap de K cicles de rellotge, es diu que és un **comptador mòdul K**. Si $K=2^n$ (on n és el número de FF) el comptador és **binari** (amb 1 FF tenim 2 estats 0,1, i amb n FF tenim 2^n estats possibles).

Hi ha de dos tipus en funció de la sincronització:

- Comptadors **asíncrons**: si l'entrada de sincronisme (rellotge) de tots els FF que el formen no és única (cada FF canvia el seu estat en moments diferents).
- Comptadors **síncrons**: El senyal de rellotge és comú per a tots els FF. Son un exemple concret de **màquines d'estats finits**.

Comptadors asíncrons binaris



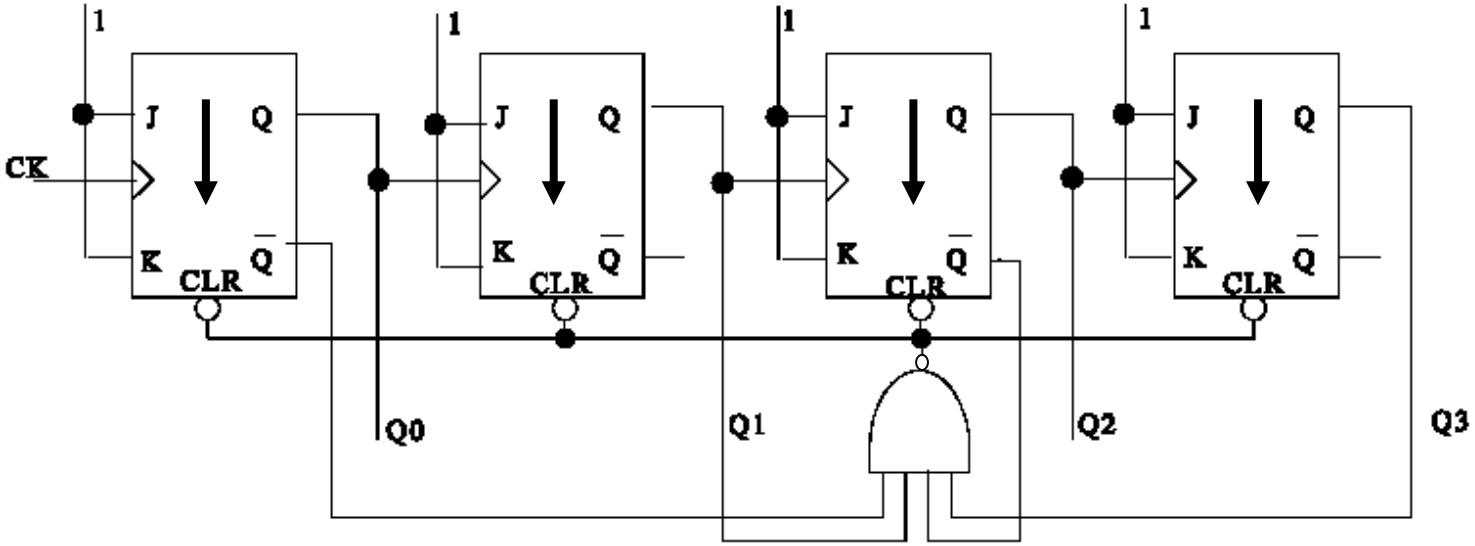
La seqüència d'aquest comptador és la natural 000, 001, 010, 011, 100, 101, 110, 111, 000, ...

Q0 canvia cada cicle de rellotge.
Q1 canvia cada 2 cicles de rellotge.
Q2 canvia cada 4 cicles de rellotge.

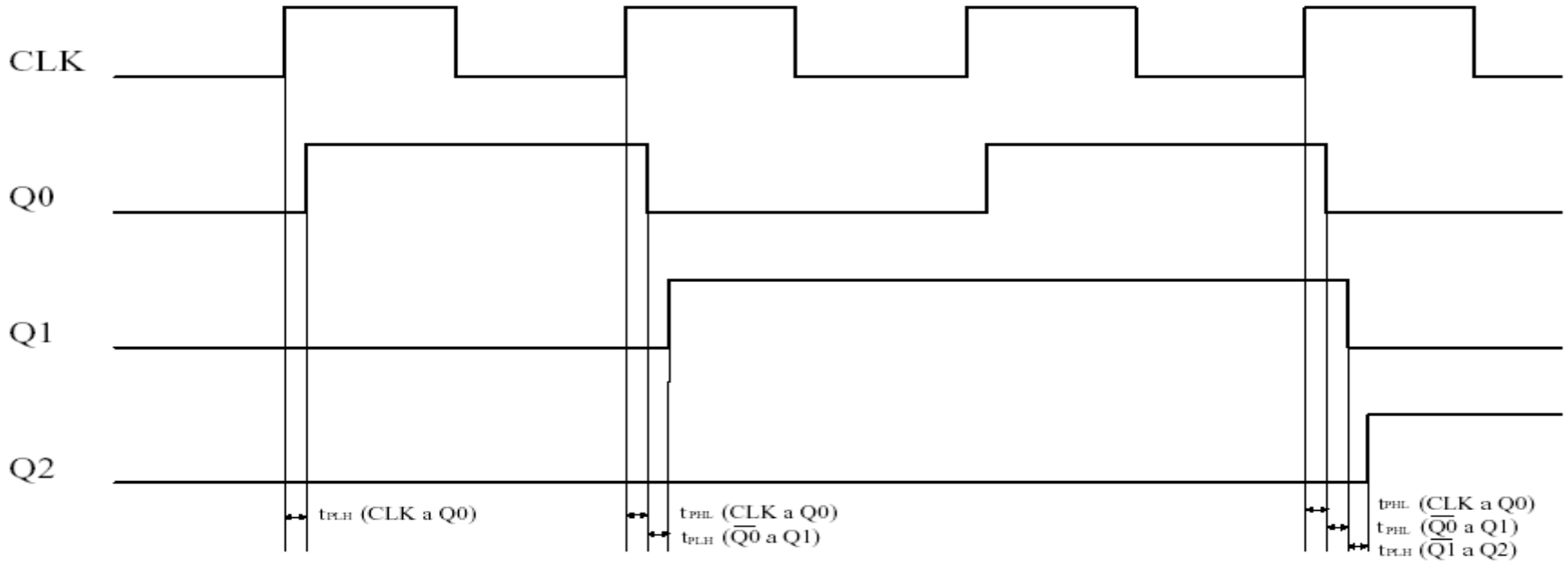
La seqüència pot canviar a descendent (7,6,5,4,...) si es connecten les sortides Q a les entrades de rellotge

Comptadors asíncrons no-binaris

Són comptadors asíncrons amb mòdul K menor que 2^n, de forma que no s'utilitzen tots els estats disponibles.



Un exemple pot ser un comptador mòdul 10, que utilitza 4 FF ($2^3<10<2^4$). Quan l'estat del comptador arribi a 10 (1010), cal que torni a l'estat 0 (0000). Això es pot realitzar amb un circuit combinacional que actua sobre el senyal asíncron clear



Els comptadors asincròns presenten l'inconvenient que no tots els FF commuten al mateix moment. Per exemple en aquest cas l'últim FF canvia el seu estat, en el pitjor dels casos, en un temps corresponent 3 vegades el temps que el un FF triga en canviar d'estat (tpd), més el temps de set-up (Ts)

$$T = nt_{pd} + T_s \Rightarrow f_{max} = 1/(nt_{pd} + T_s)$$

Retard acumulat

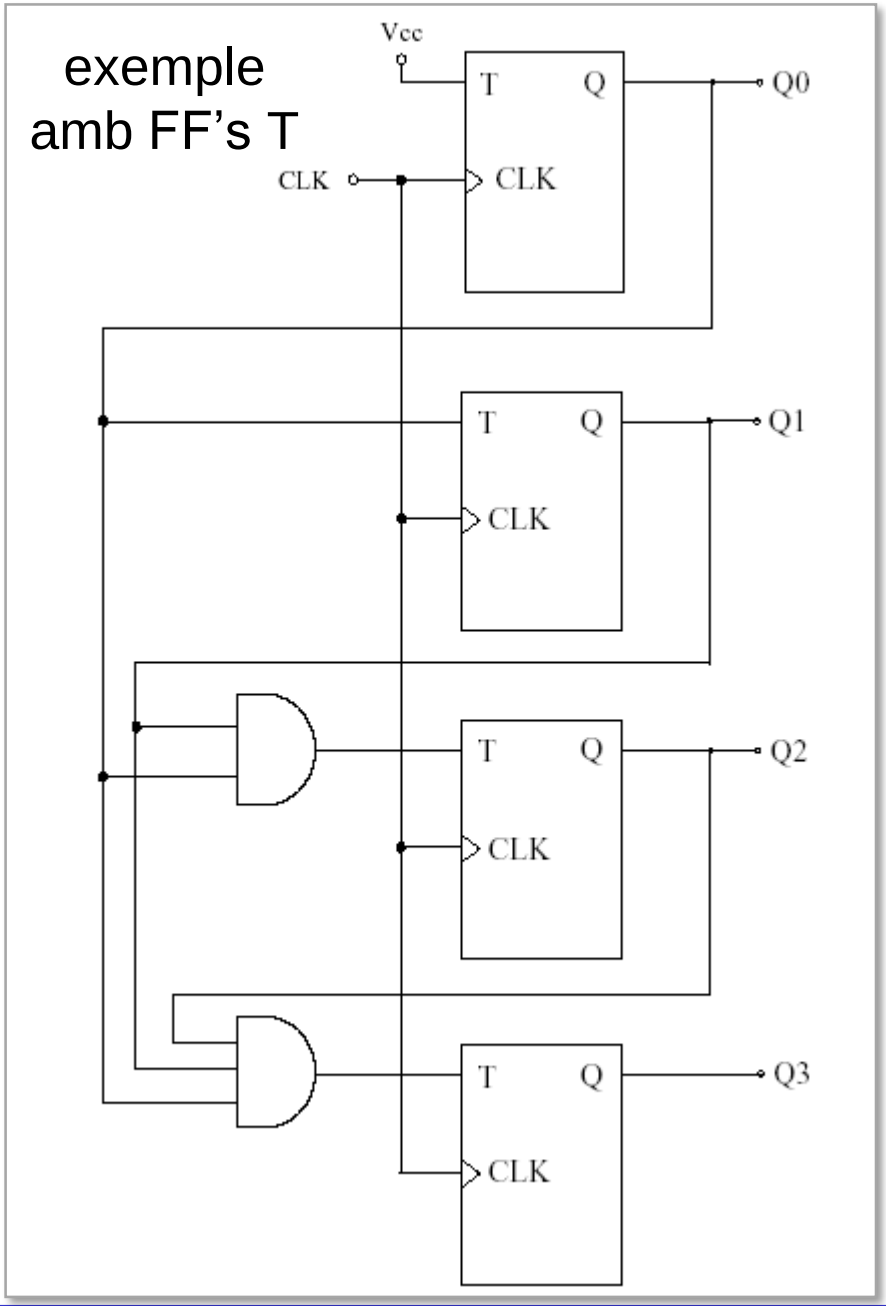
Comptadors SÍNCRONS

El senyal de rellotge és comú per a tots els FF. Son un exemple concret de màquines d'estats finits.

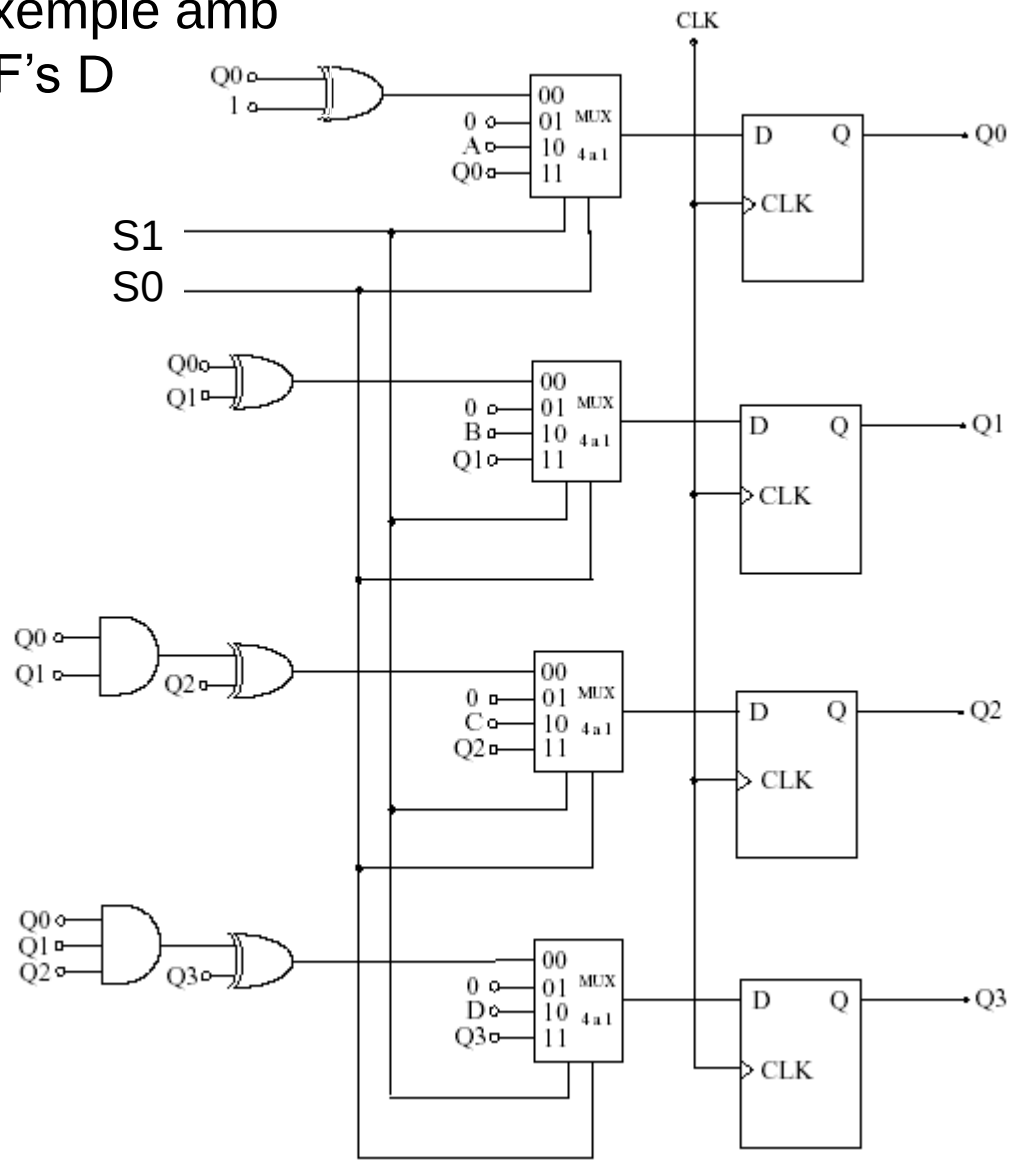
Com que tots els FF commuten simultàniament, aquests comptadors són més ràpids que els comptadors asíncrons.

El retard màxim és el temps de commutació d'un FF més el d'una porta AND més el temps de set-up. Així la freqüència màxima serà:

$$f_{max} = \frac{1}{t_{pd}(FF) + t_{pd}(AND) + t_{setup}}$$



exemple amb FF's D



Es pot fer un comptador més complert amb entrades síncrones S1 i S0 que facin:

S1	S0	operació
0	0	Compta
0	1	Posar a 0
1	0	Càrrega
1	1	Mantenir

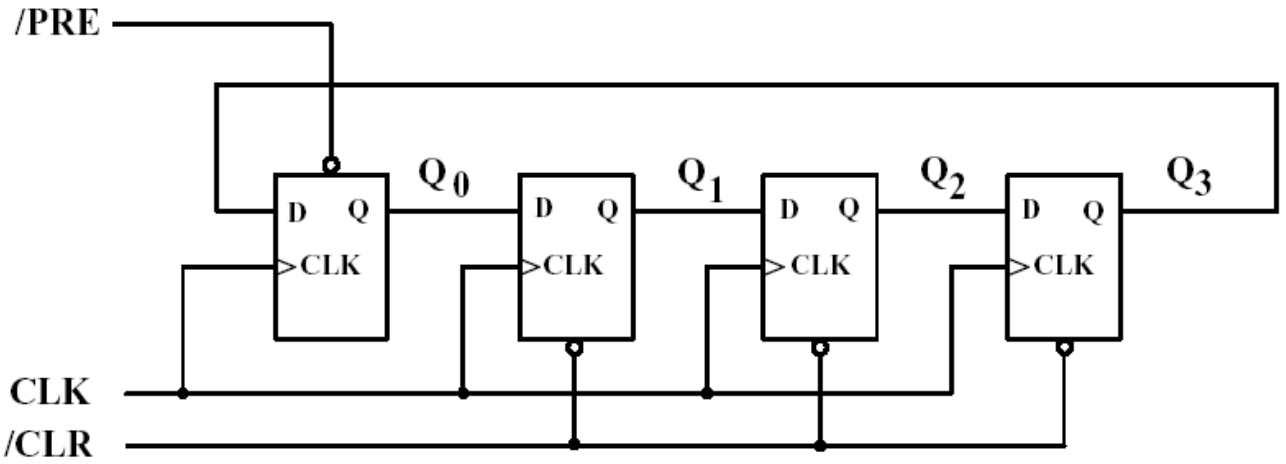
Aquest comptador permet:

- Comptar
- Posar a 0 sincrònicament
- Carregar un valor a partir del qual començar el compte
- Mantenir un valor durant un temps

Comptadors d'anell

És un tipus de comptador realitzat amb FF D connectats de manera que: $D_{i+1}^{+}=Q_i$ i $D_0^{+}=Q_{n-1}$.

Pulso de reloj	Q0	Q1	Q2	Q3
0	1	0	0	0
1	0	1	0	0
2	0	0	1	0
3	0	0	0	1



Es pot considerar també un divisor de freqüència (apareix un 1 cada n polsos de rellotge, on n és el número de FF del comptador).

Comptadors d'anell trenat o Johnson

És similar al comptador d'anell per a tots els FF tret del primer, per al qual la relació amb l'últim és: $D0^{+}=\overline{Qn-1}$.

Pulso de reloj	Q0	Q1	Q2	Q3
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1