

## Índex de conceptes

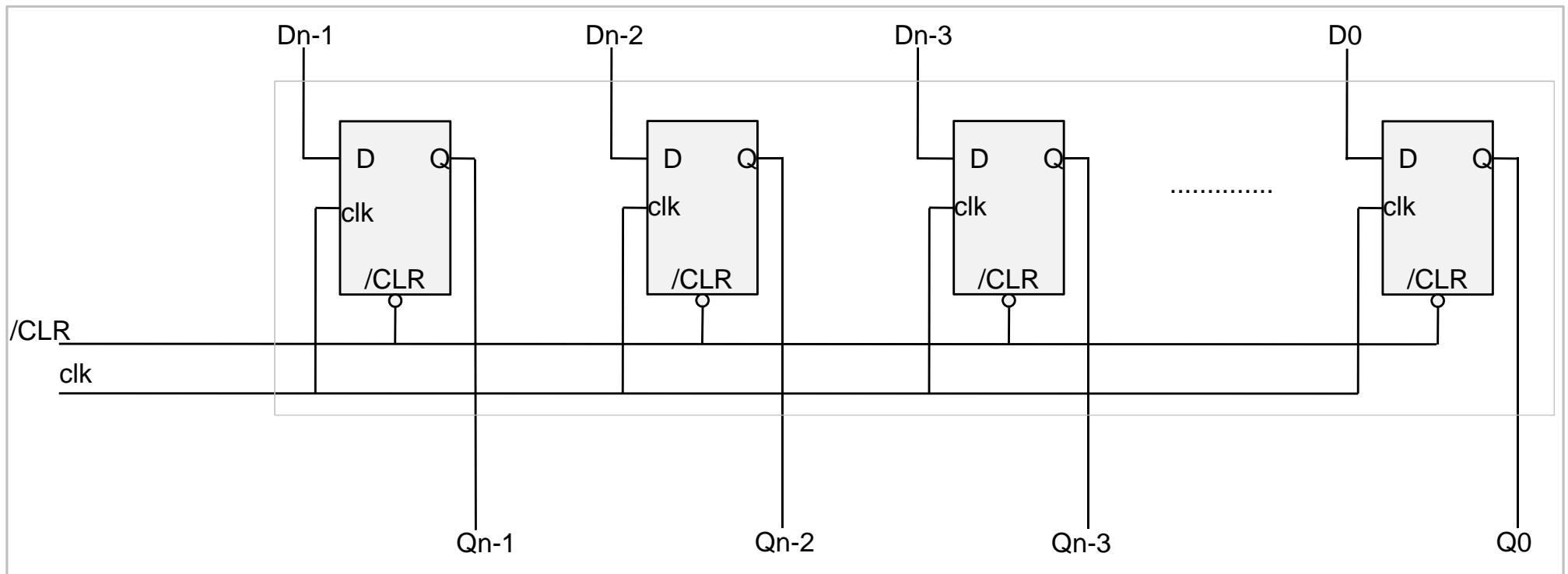
- **Estructura d'un registre**
- **Buses i Buffers**
- **Registre de desplaçament**
- **Comptador asíncron**
- **Comptador síncron**

# 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:

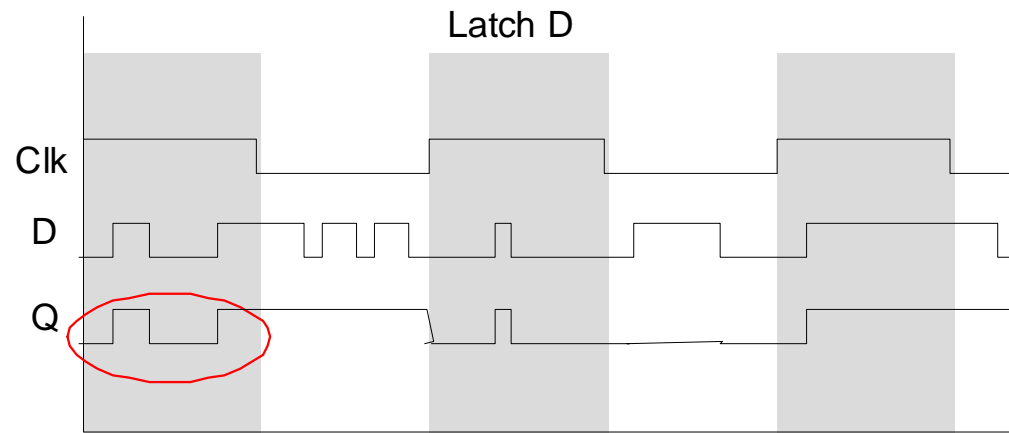
- Flancs: **Flip-Flop edge triggered**
- Nivells: Latch



$n$  elements =  $n$  bits

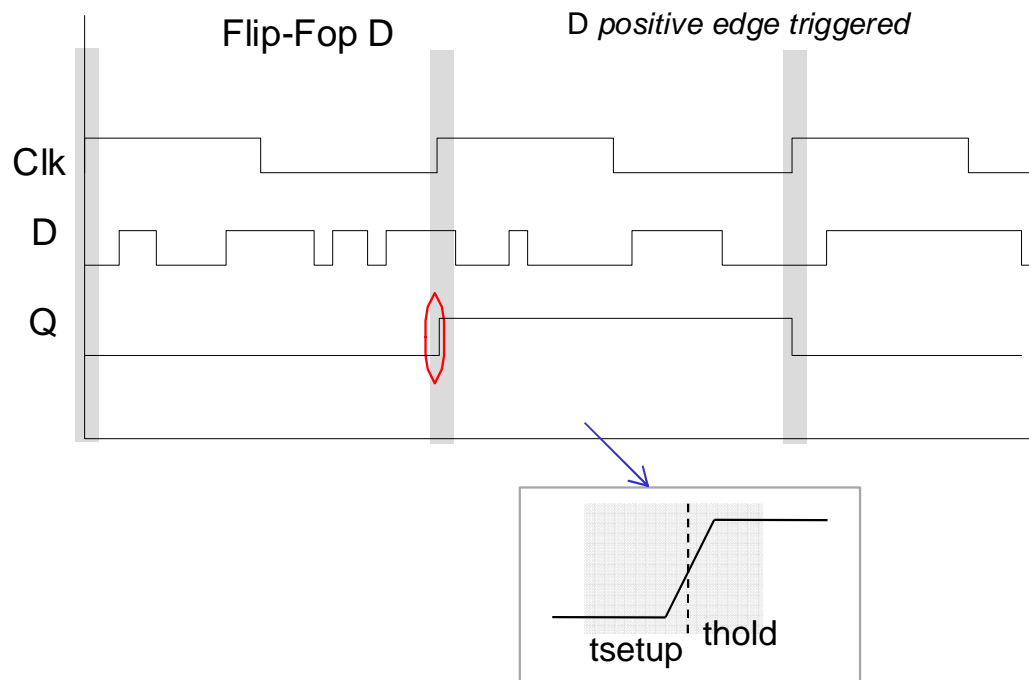
Les dades  $D_{n-1}$ - $D_0$  entren en paral·lel (simultàniament) i al següent flanc del senyal de rellotge aquestes dades es presenten a les sortides  $Q_{n-1}$ - $Q_0$ . Al pols de rellotge següent es perd la informació i es carrega de nou el registre.

# Latch vs. Flip-Flop



## Latch

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



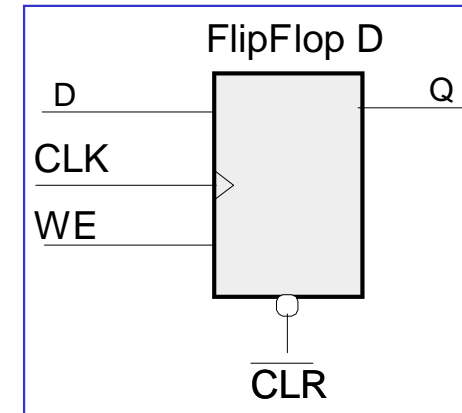
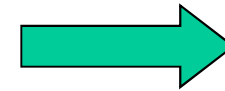
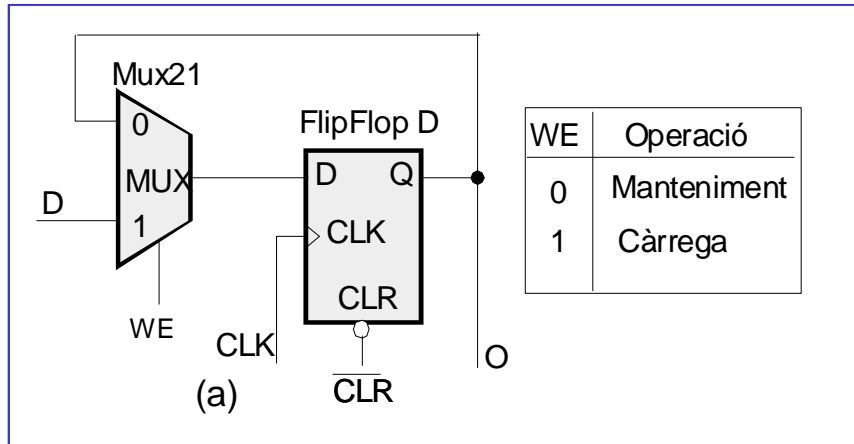
## Flip-Flop

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

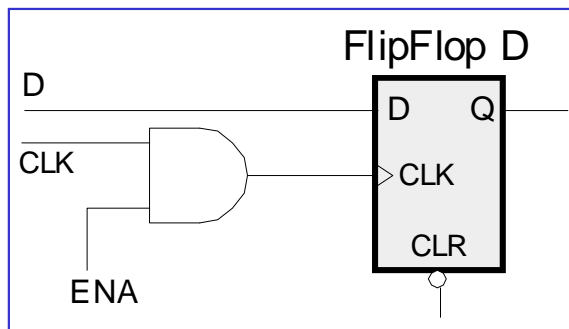
# Estructura dels Registres

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

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



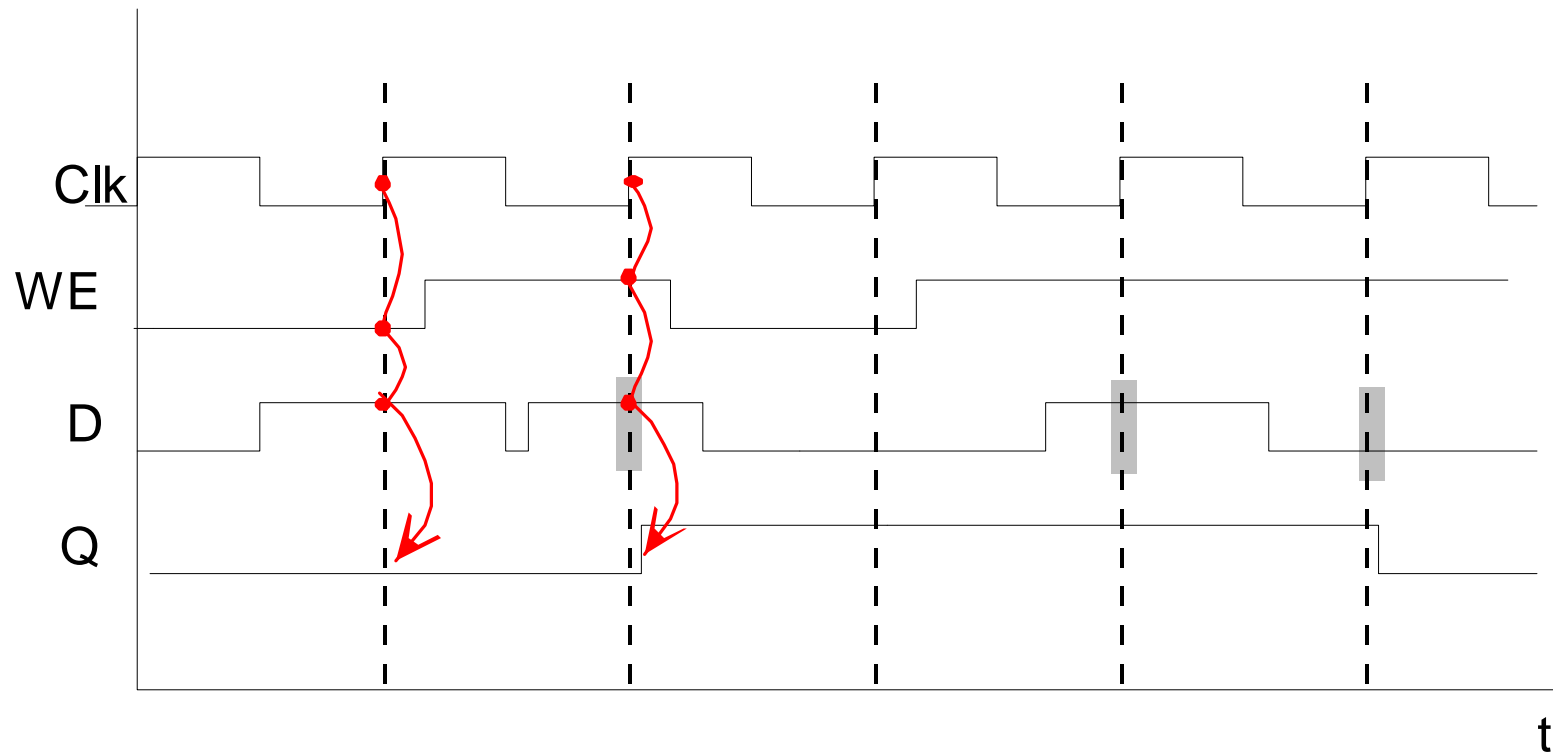
Altra solució: Gated Lock. Habilitar/inhibir el clock amb una porta AND. Habilitació actuant sobre el senyal de clock. No es aconsellable en general i s'utilitza en casos especials (exemple: reduir consum)



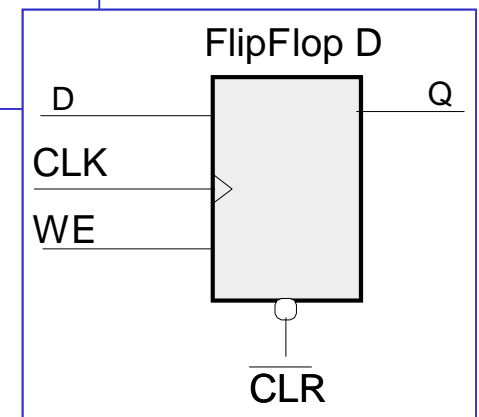
*Gated clock*

- Retard en clock
- Pèrdua de sincronisme

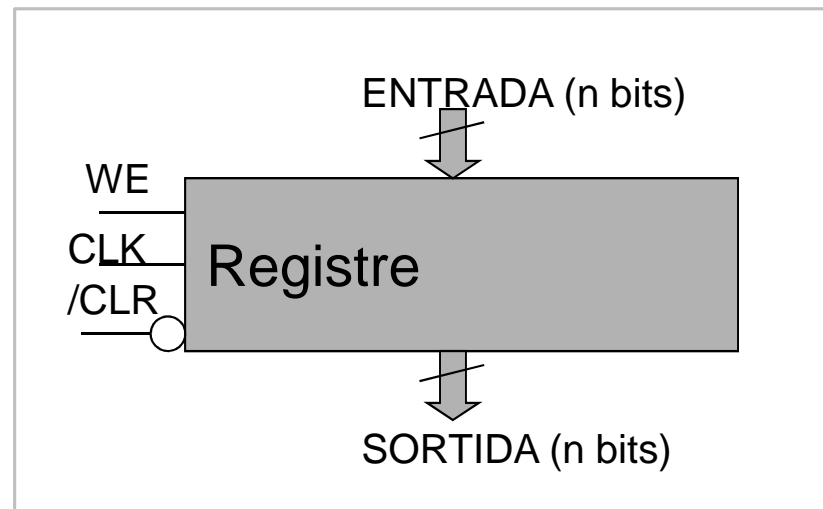
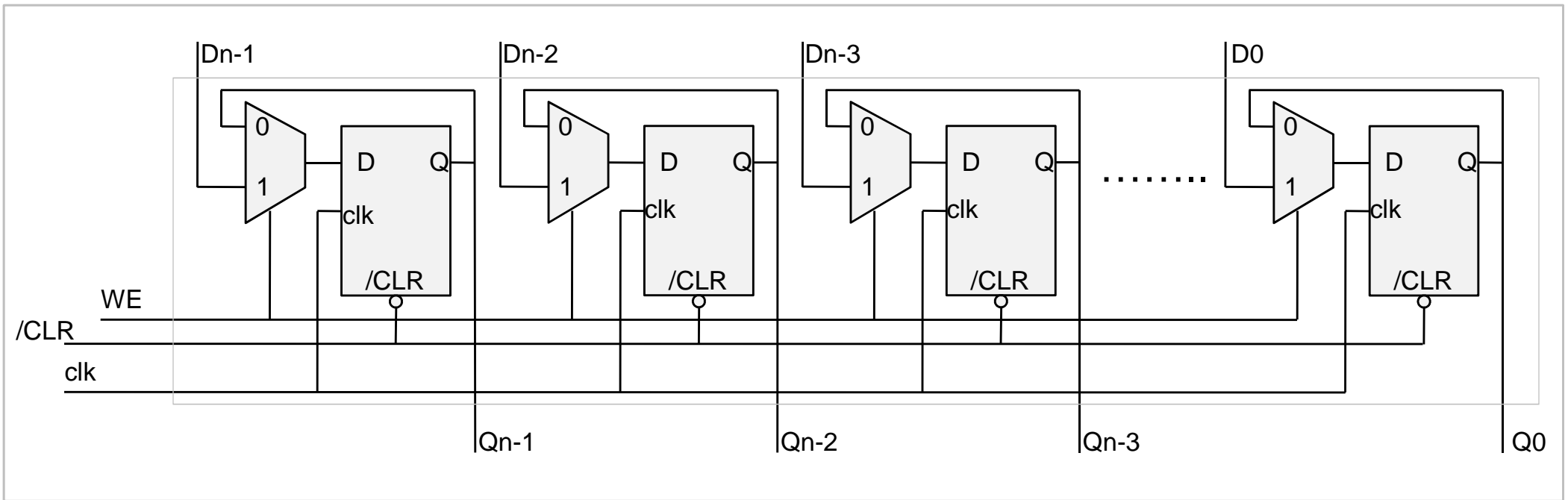
# Sincronisme en Registres



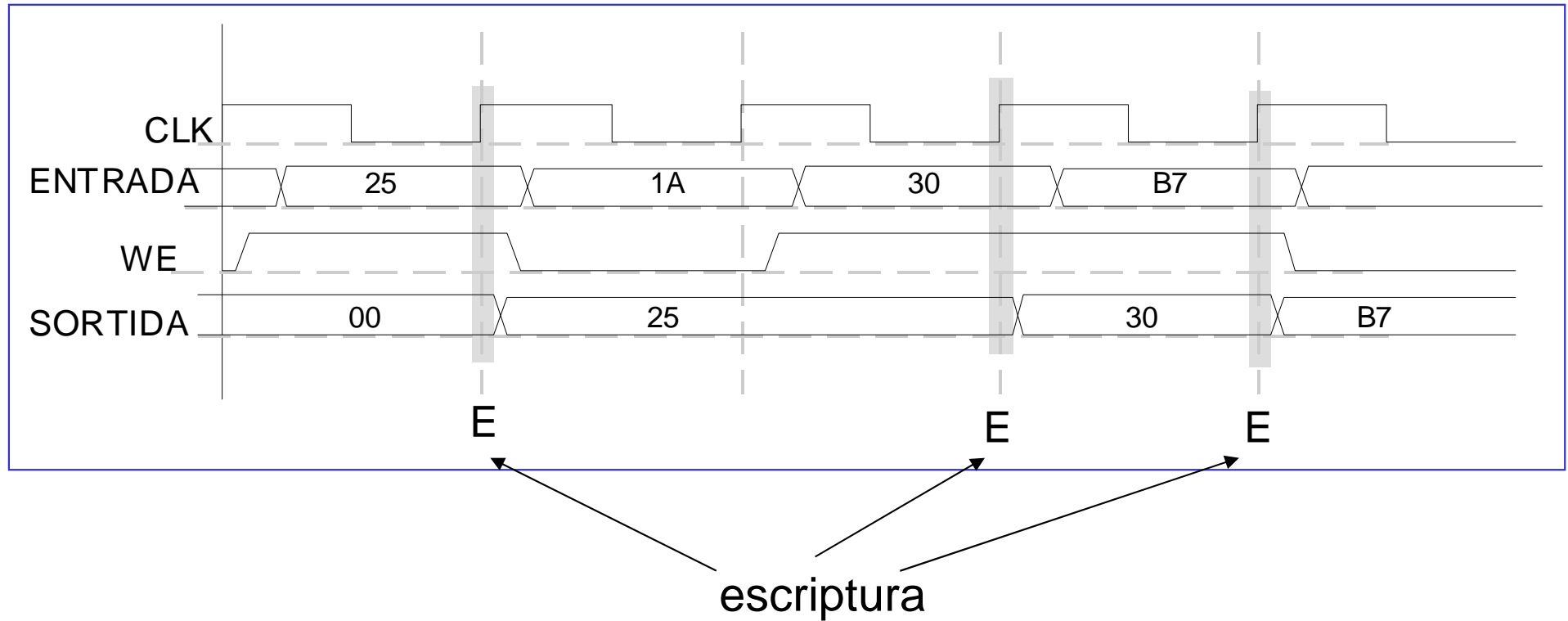
Registre governat per senyal d'habilitació (WE)



# Registre d'n bits

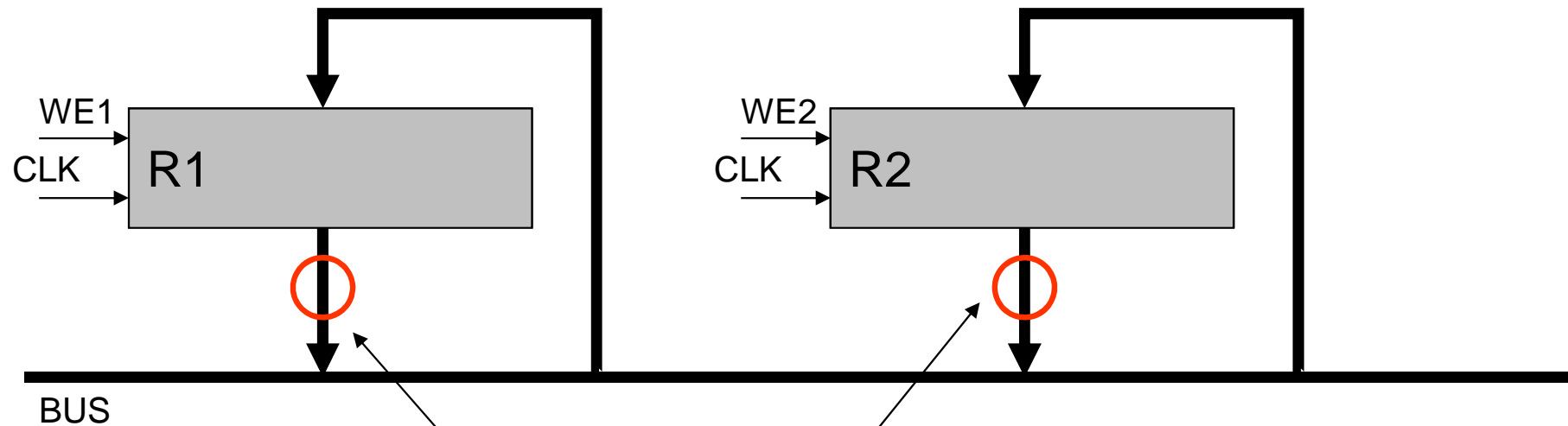


## Espectura SÍNCRONA dels Registres





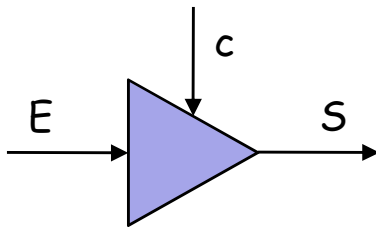
### Conflictes de BUS



- 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

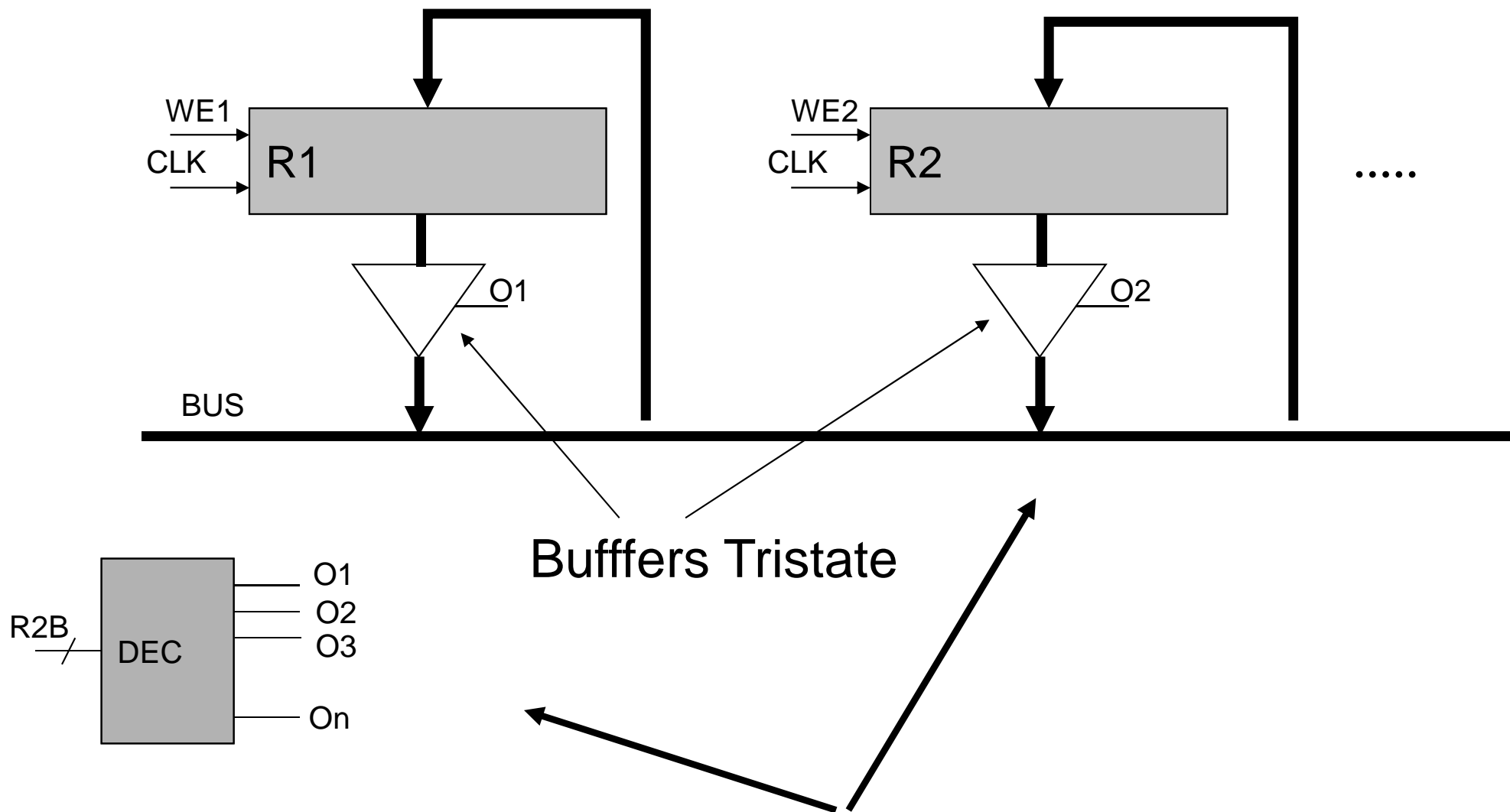
# BUFFERS TriState

- 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 intermig entre 0 i 1)
- permeten inhibir la sortida, a partir d'una entrada de control, c.
- permeten connectar diverses sortides entre sí, permetent seleccionar-ne una d'elles



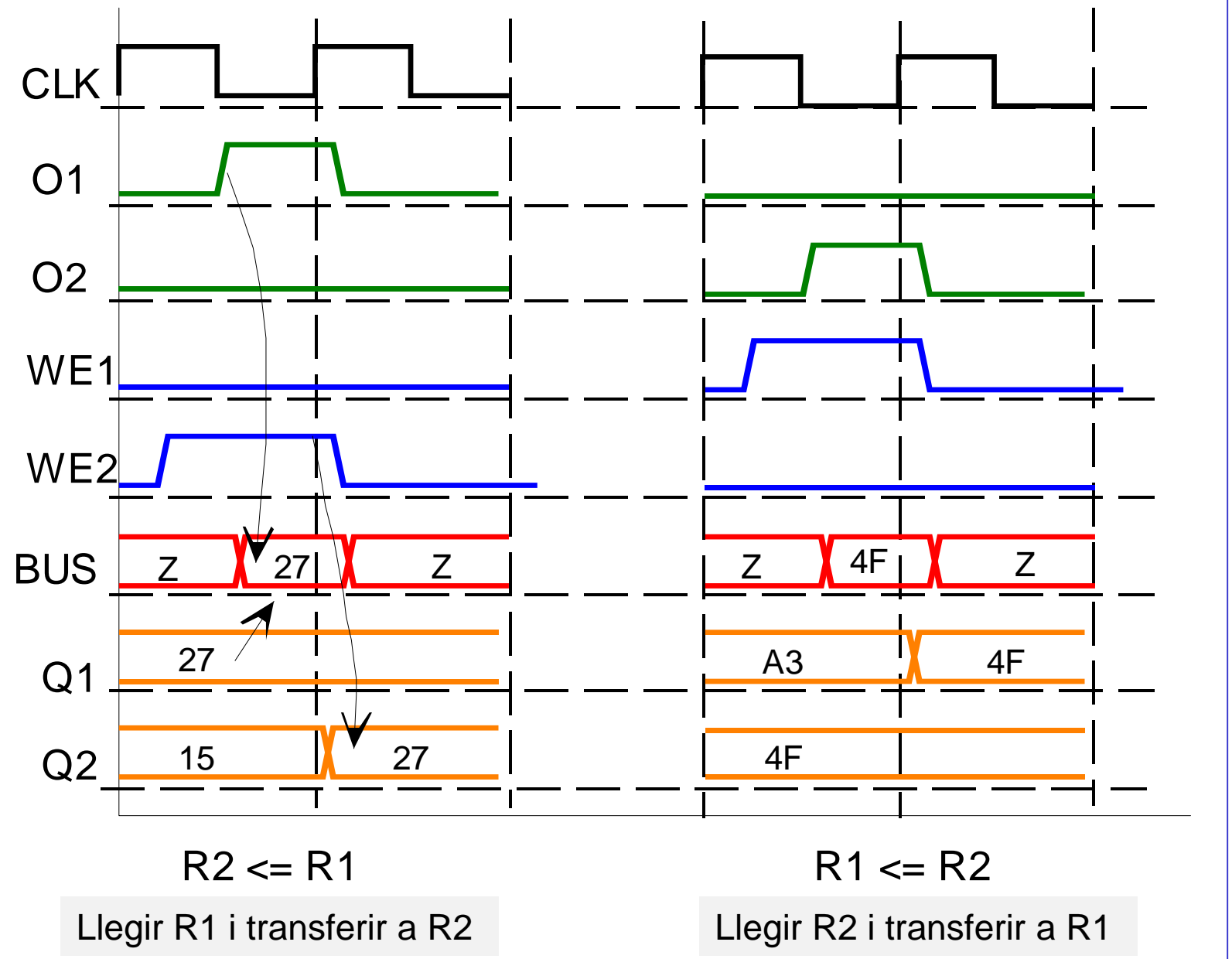
c	E	S
0	0	Z
0	1	Z
1	0	0
1	1	1

**Z** = Estat d'alta impedància,  
equivalent a desconnectat



# Exemple

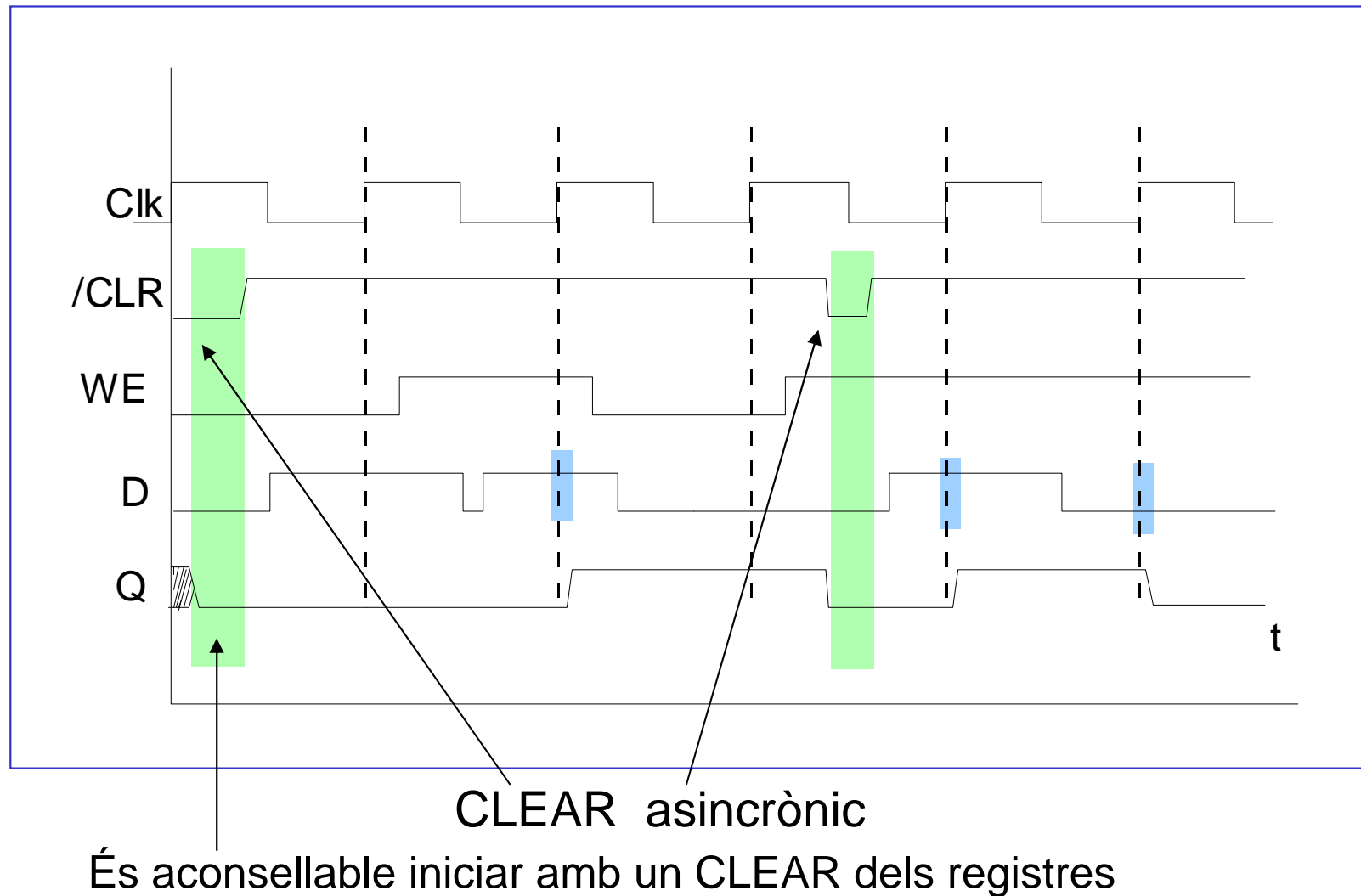
Transferència de dades entre registres



## RESET/CLEAR en Registres

Generalment els flip-flops consten de senyal de RESET (CLEAR) asíncron

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



## Registre de desplaçament (*Shift Register*)

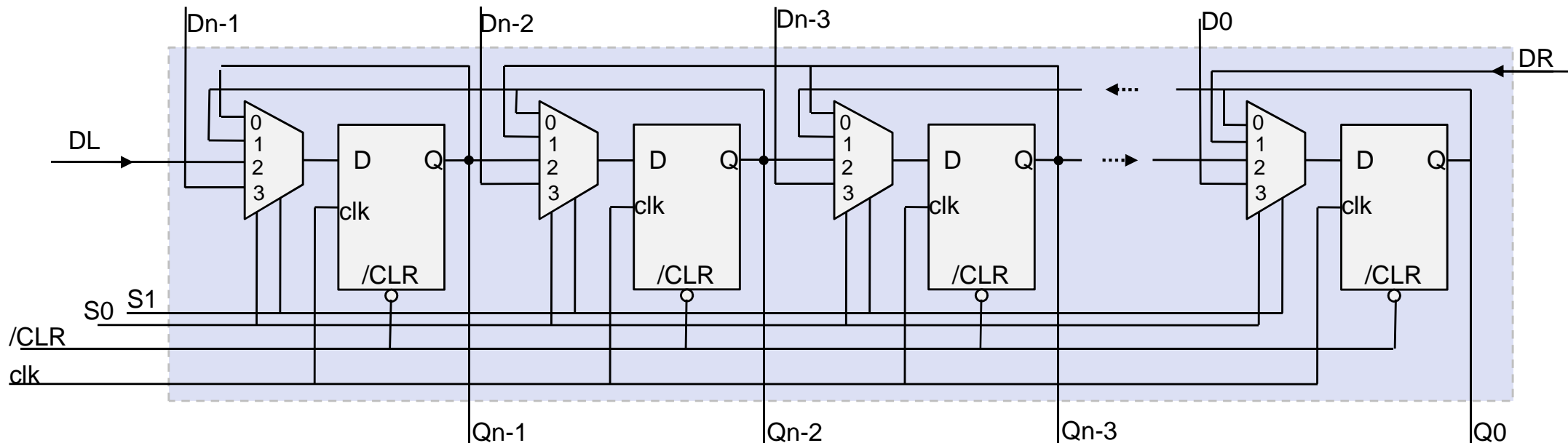
Son registres en els quals la informació que contenen es pot desplaçar cap a la dreta o cap a la esquerra dins del mateix registre.

La seva principal utilitat és la **transferència de dades de format paral·lel a format sèrie** i viceversa

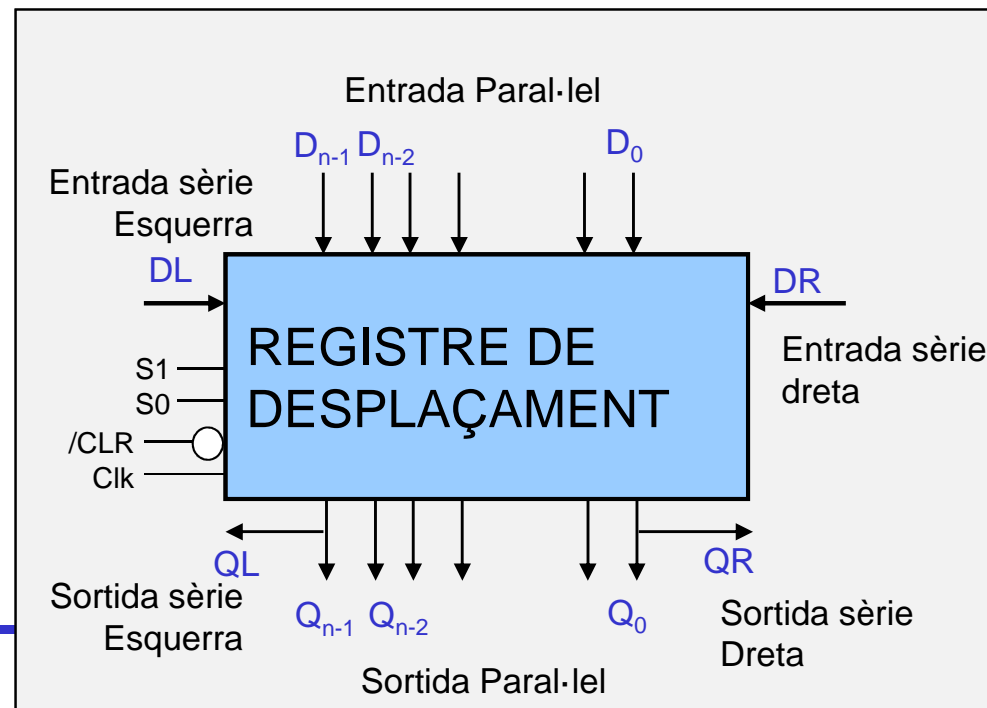
Tenen també utilitat aritmètica:

- Desplaçar a la dreta = dividir per 2
- Desplaçar a l'esquerra = Multiplicar per 2

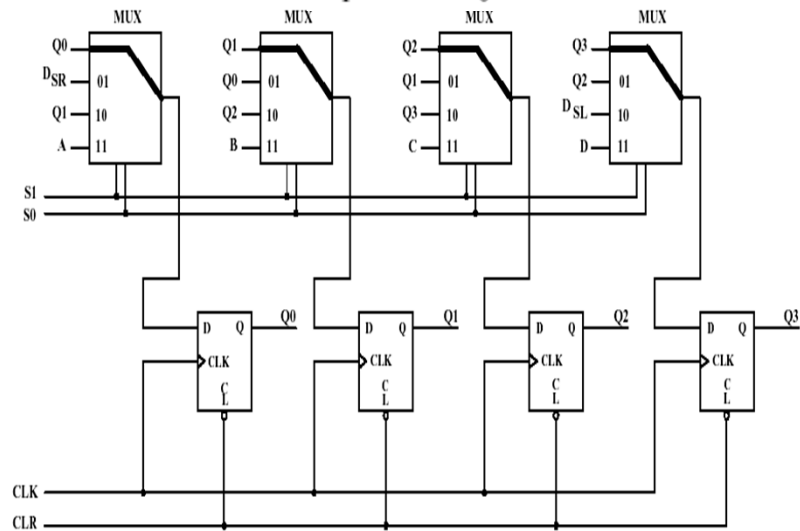
# Registre de desplaçament (*Shift Register*)



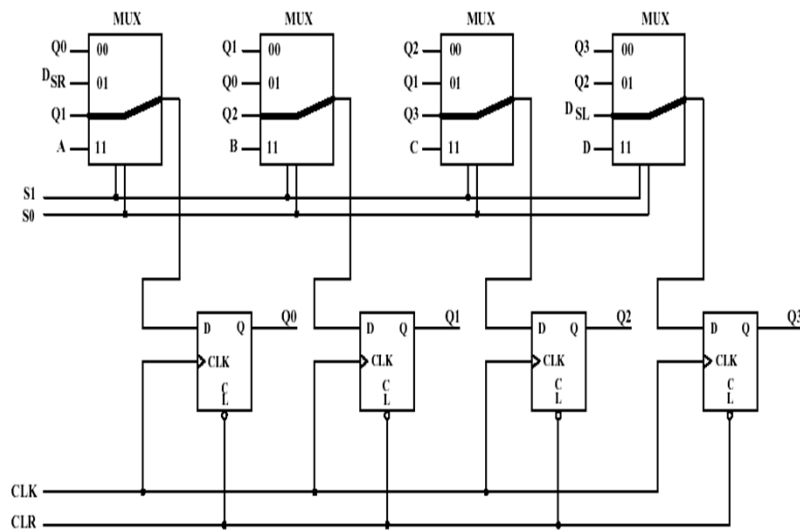
Selecció S1 S0		Operació		Senyals de Sortida $Q_{n-1} \quad Q_{n-2} \quad \dots \quad Q_1 \quad Q_0$					
0	0	Mantenir	HOLD	$Q_{n-1}$	$Q_{n-2}$	$\dots$	$Q_1$	$Q_0$	
0	1	Despl. esquerra	SHR	$Q_{n-2}$	$Q_{n-3}$	$\dots$	$Q_0$	DR	
1	0	Despl. Dreta	SHL	DL	$Q_{n-1}$	$\dots$	$Q_2$	$Q_1$	
1	1	Càrrega	LOAD	$D_{n-1}$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



$S_1=0$  y  $S_0=0$

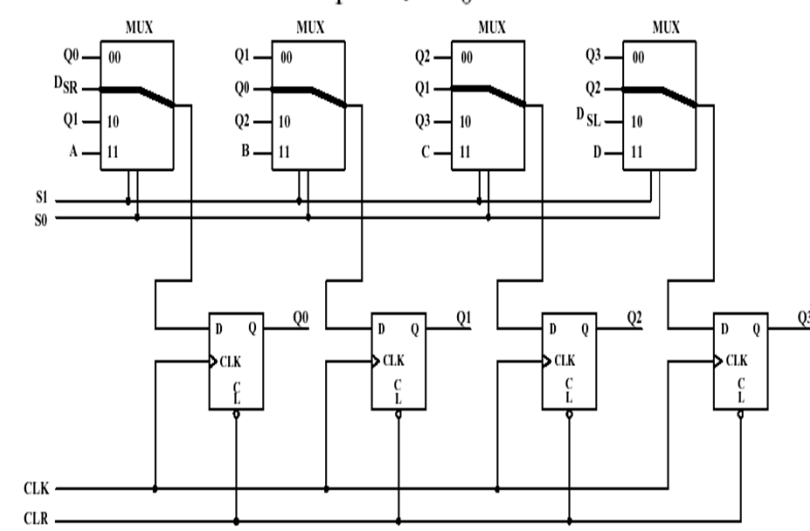


$S_1=1$  y  $S_0=0$

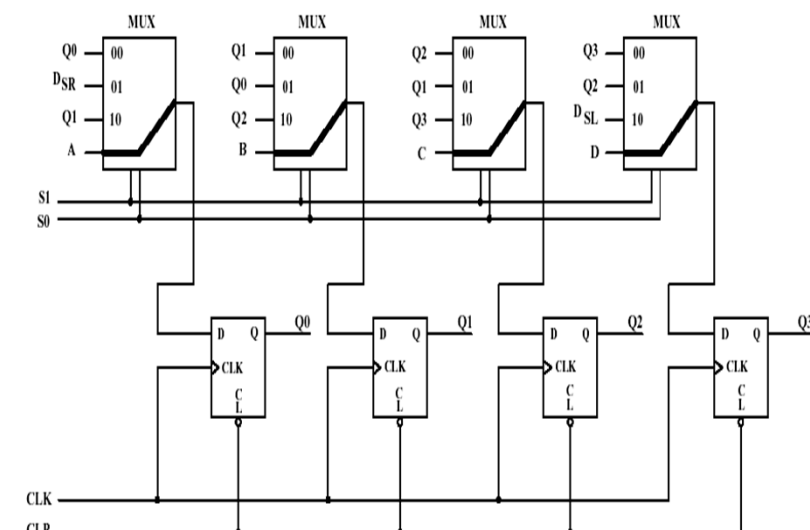


S1	S0	Operació
0	0	Manteniment (HOLD)
0	1	Desplaçament dreta (SR)
1	0	Desplaçament esquerra (SL)
1	1	Càrrega paral·lel (LOAD)

$S_1=0$  y  $S_0=1$



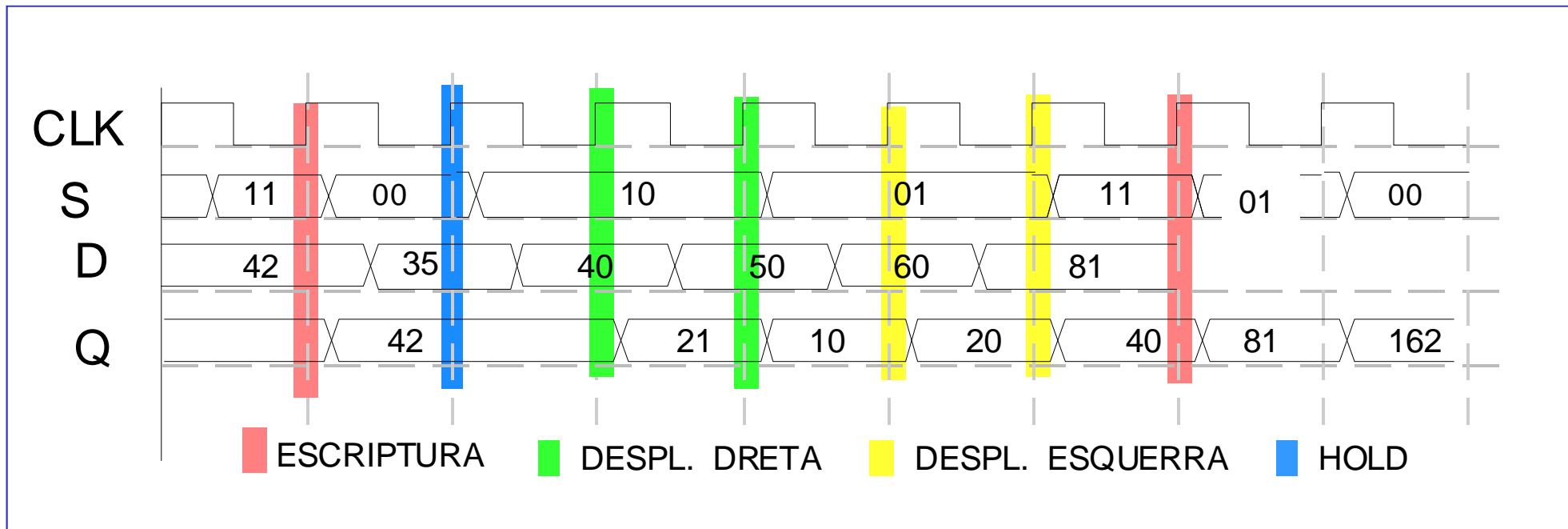
$S_1=1$  y  $S_0=1$





## Exemple

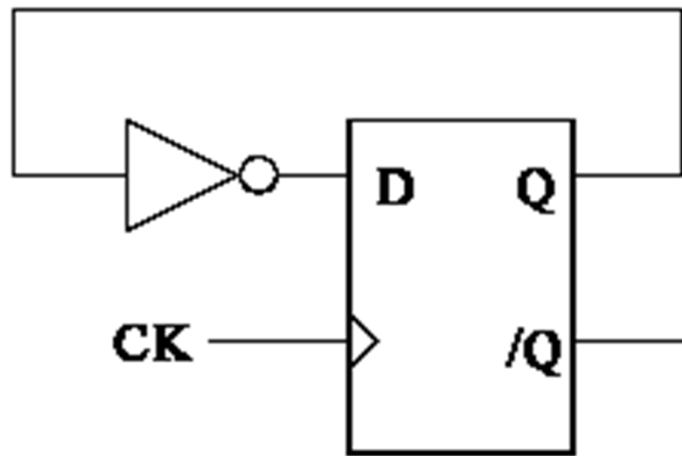
### Funcionament d'un Registre de Desplaçament



# 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).



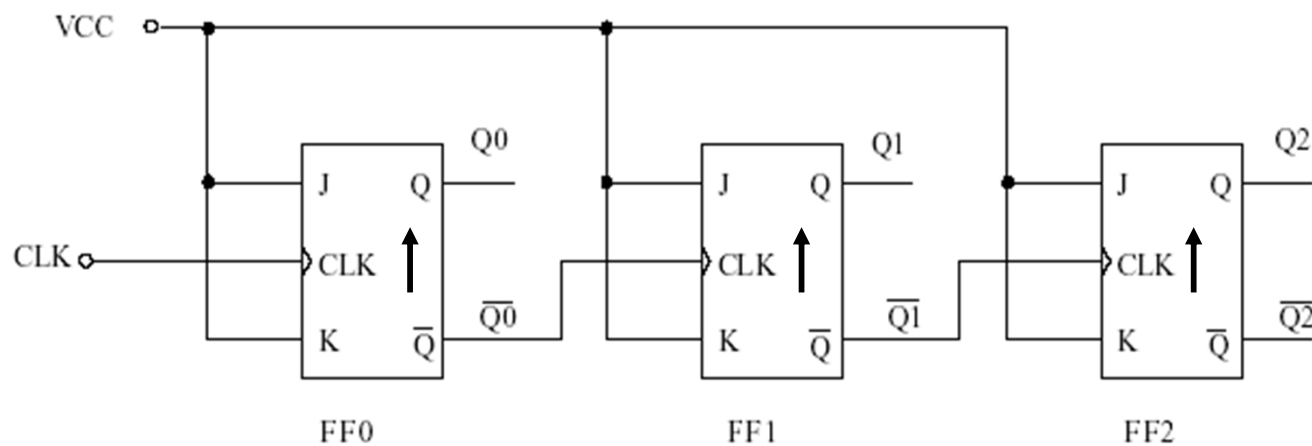
Comptador binari mòdul 2: la seqüència, partint del valor inicial 1, és: 1,0,1,0,1,0,1,...

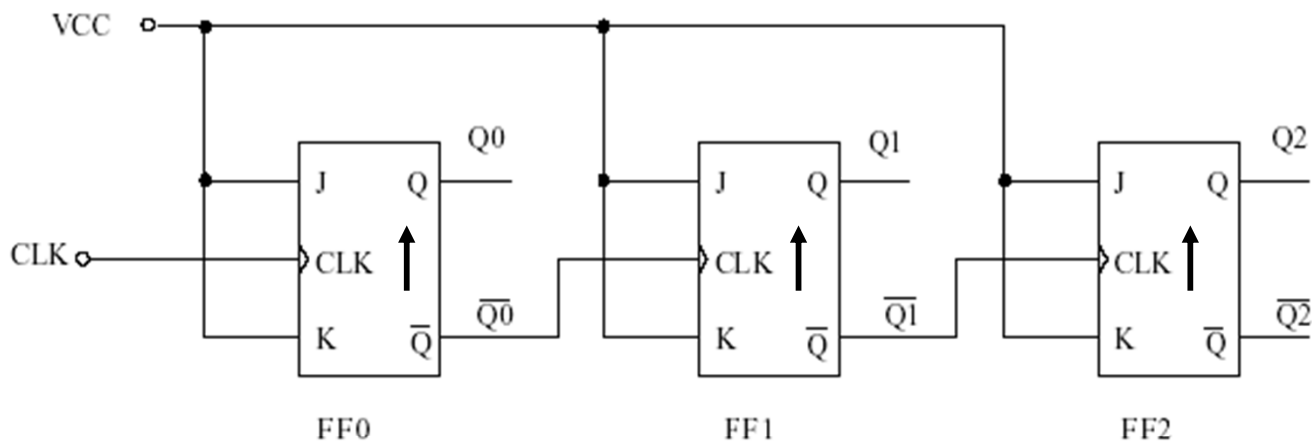
## Comptadors asíncrons

Un comptador és asíncron 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).

En general, el sincronisme depèn del senyal de sincronisme extern i de l'estat de tots els FF.

## Comptadors asíncrons binaris



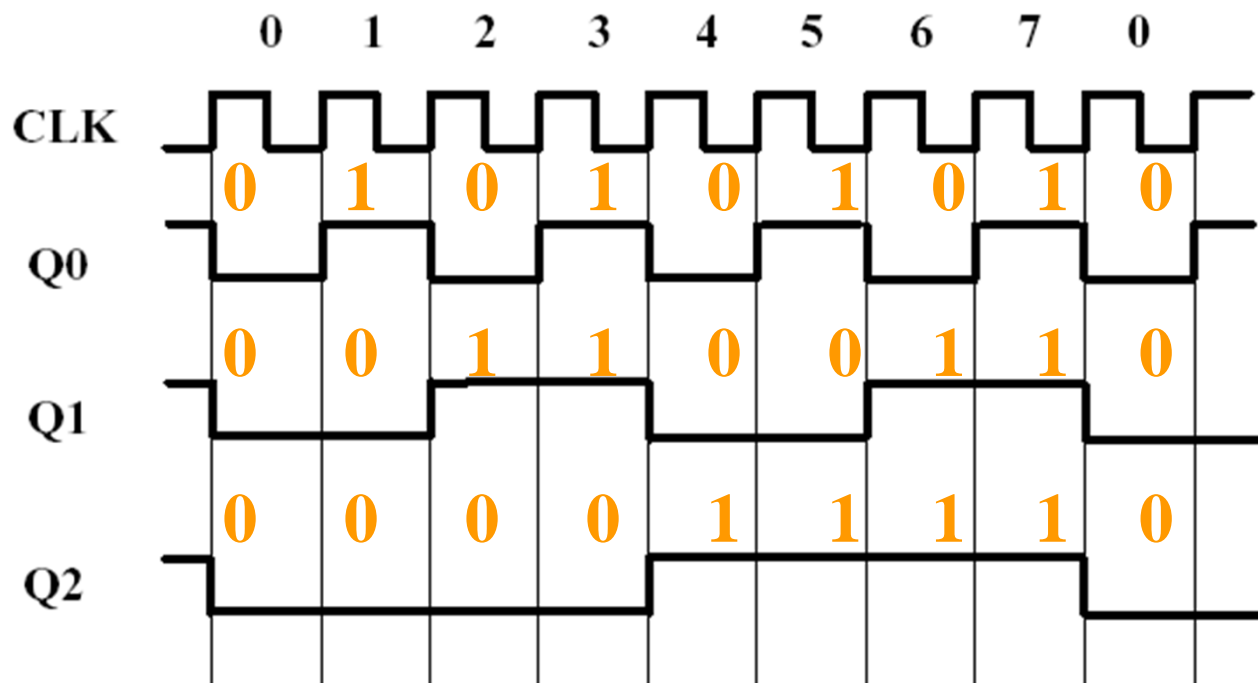


La seqüència d'aquest comptador és la natural 000, 001, 010, 011, 100, 101, 110, 111, 000, ...

Q0 canvia cada cicle de rellotge.

Q1 canvia cada 2 cicles de rellotge.

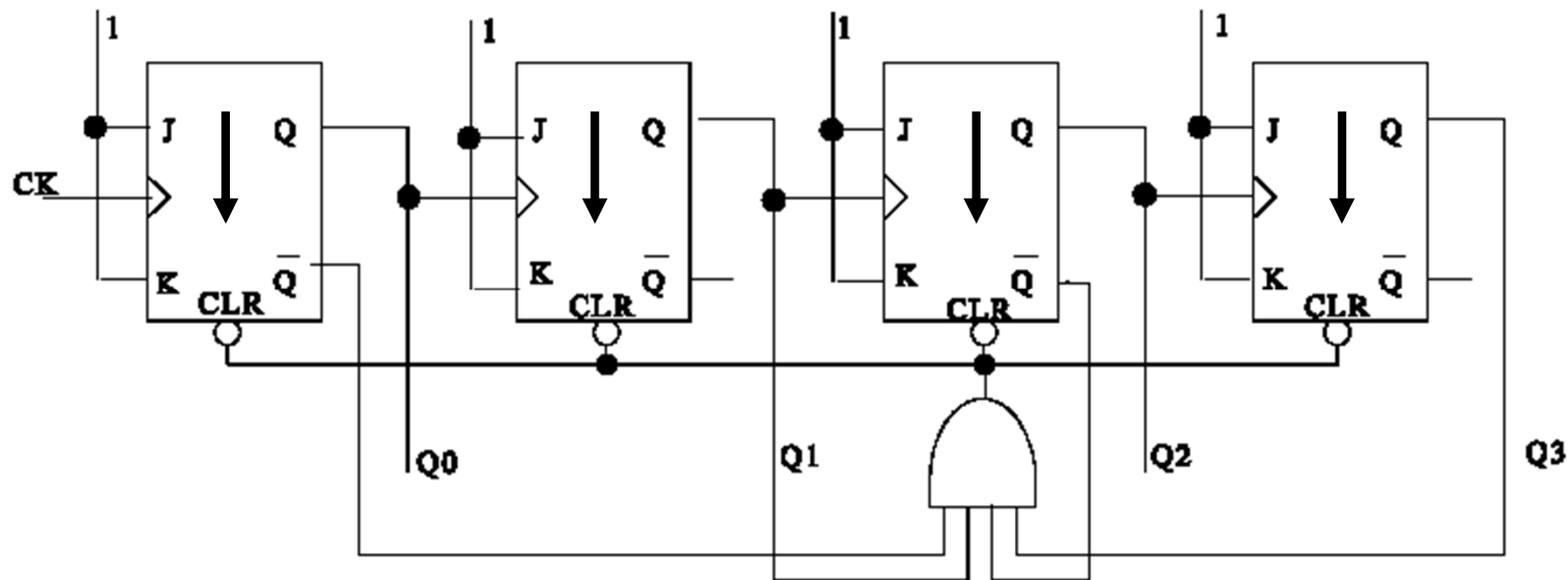
Q2 canvia cada 4 cicles de rellotge.

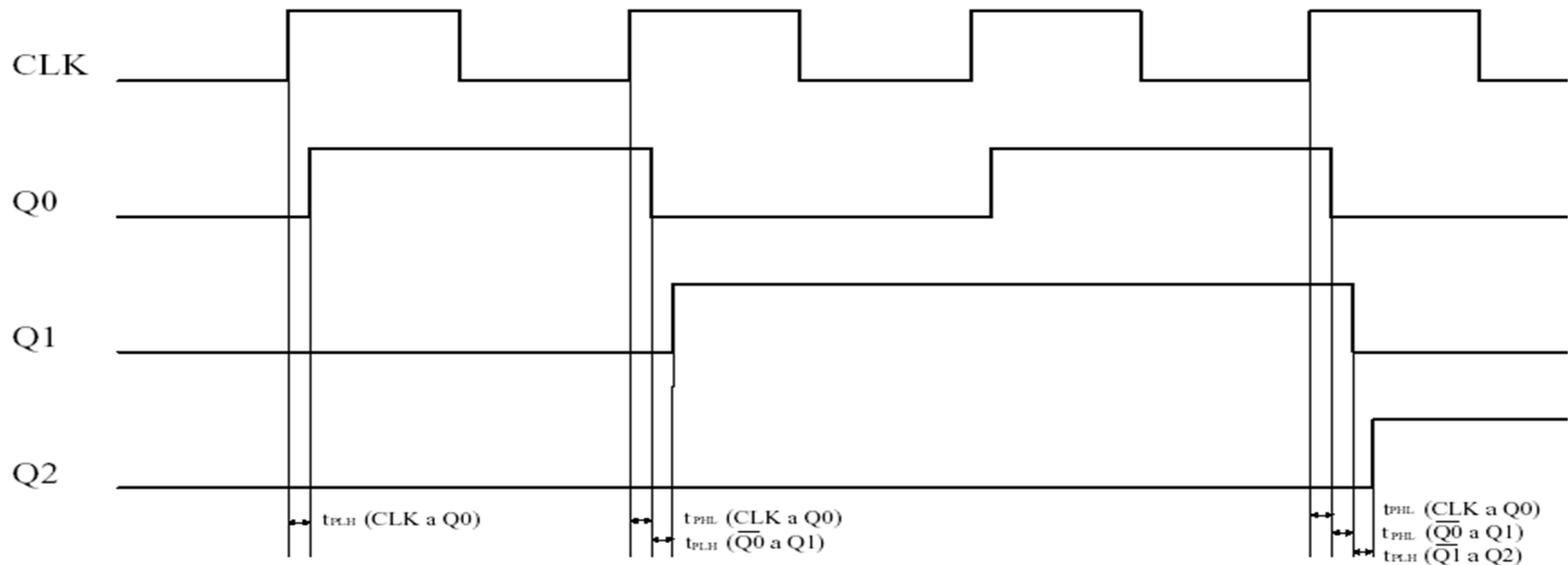


La seqüència canvia a descendent si es connecten les sortides Q a les entrades de rellotge

## 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 0 (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 asincrònics 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 ( $t_{pd}$ ), més el temps de set-up ( $T_s$ )

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

Retard acumulat



## Comptadors sincrònics

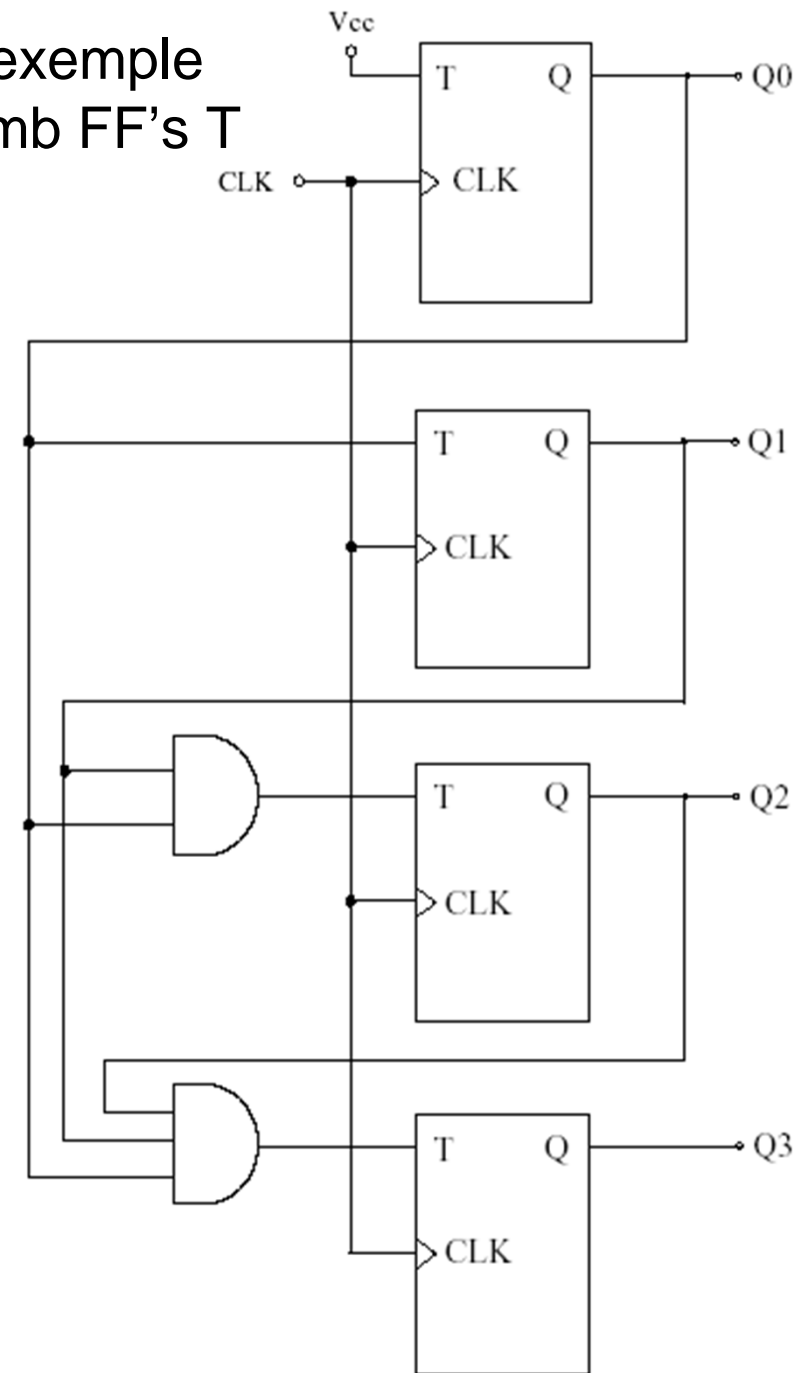
El senyal de rellotge és comú per a tots els FF. Son un exemple concret de **màquines d'estats finits**

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

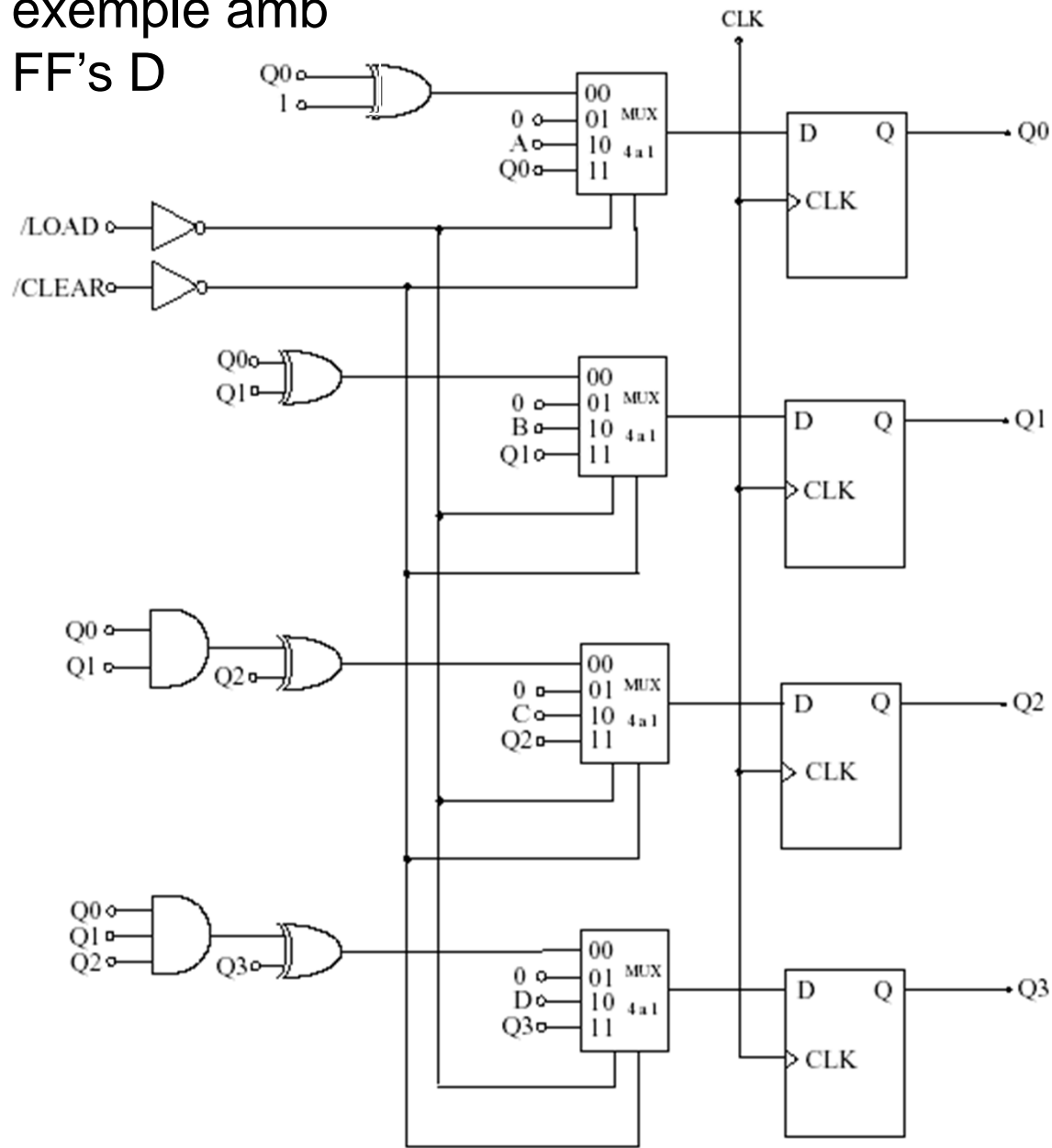
El retard màxim és el temps de commutació d'un FF més el d'una porta AND més el temps de set-up. Així la freqüència màxima serà:

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

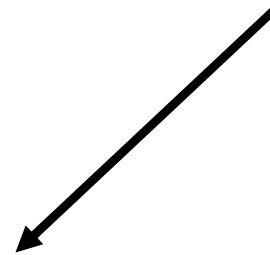
exemple  
amb FF's T



## exemple amb FF's D



Es pot fer un comptador més  
complet amb entrades de  
LOAD i CLEAR sincròniques

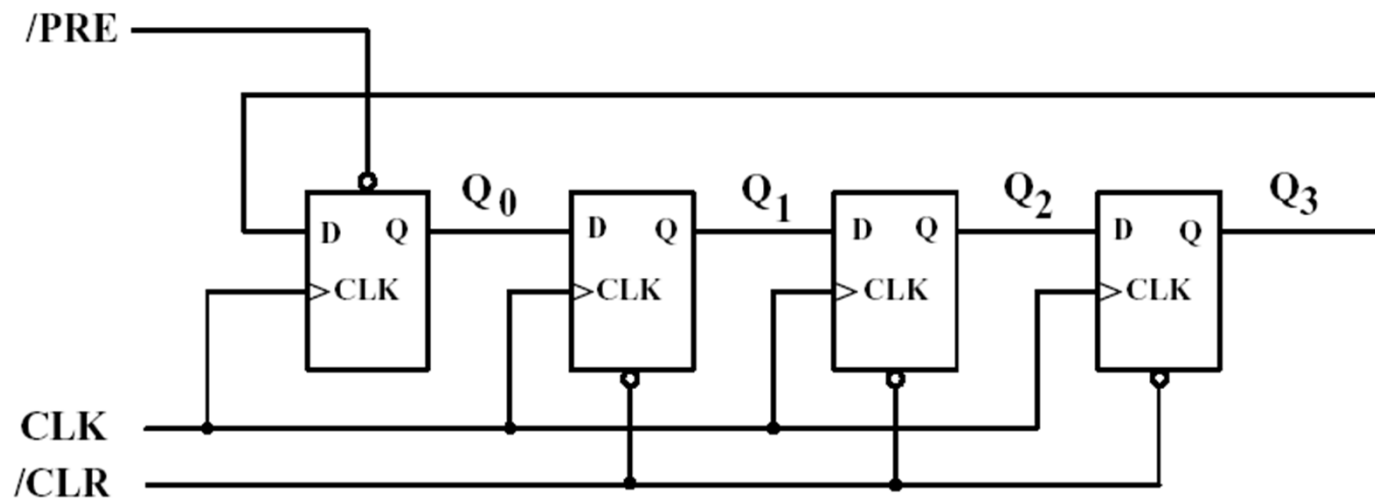




## Comptadors d'anell

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 reloxj	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:  $D0^+ = \overline{Q_{n-1}}$ .

Pulso de reloi	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

