



UNIVERSIDAD
POLITECNICA
DE VALENCIA



Fundamentos de los computadores

Tema 4. Circuitos Secuenciales.



poli **OCW**
OPENCOURSEWARE

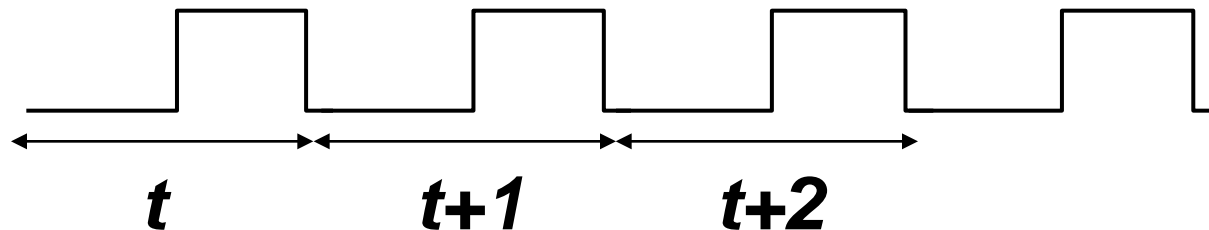


- Estudiar los circuitos secuenciales más sencillos.
- Introducir el concepto de cronograma.
- Comprender el funcionamiento básico de los biestables.

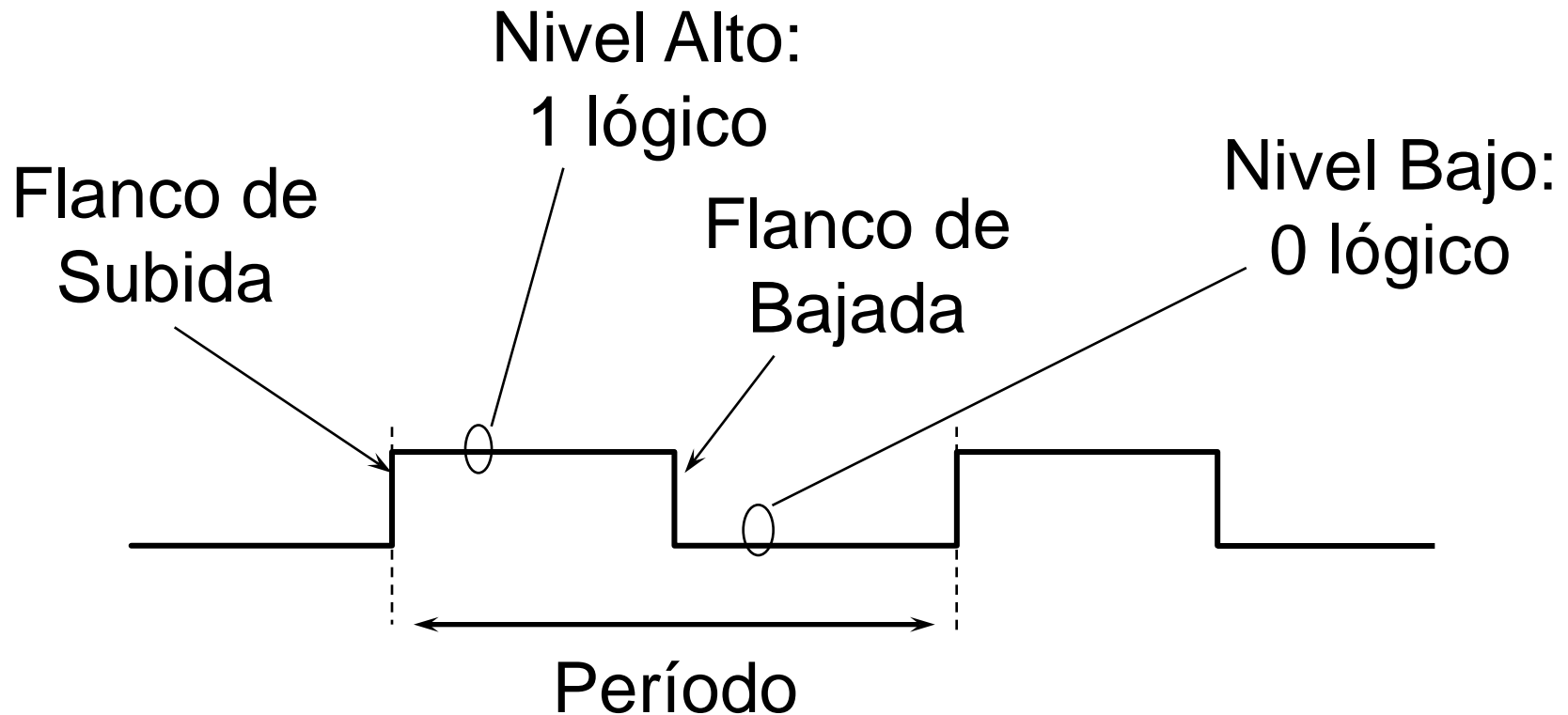
- Introducción
 - Circuitos secuenciales, reloj, cronogramas, símbolos lógicos.
- Biestables
 - Biestable RS asíncrono
 - Biestable D
 - Síncrono
 - Activo por flanco
 - Con entradas asíncronas
 - Biestable JK
 - Biestable T

- Circuitos secuenciales:
 - las salidas del circuito en el instante actual $S(t)$ dependen no sólo del valor actual de las entradas del circuito $E(t)$, sino también de su “memoria” o “estado almacenado” $Q(t)$
 - Están formados por
 - un bloque combinacional
 - un bloque de elementos de memoria para almacenar el estado Q
 - una entrada de reloj que decide cuando pasamos del tiempo t al $t+1$, y que marca cuando se almacena interiormente el estado del sistema

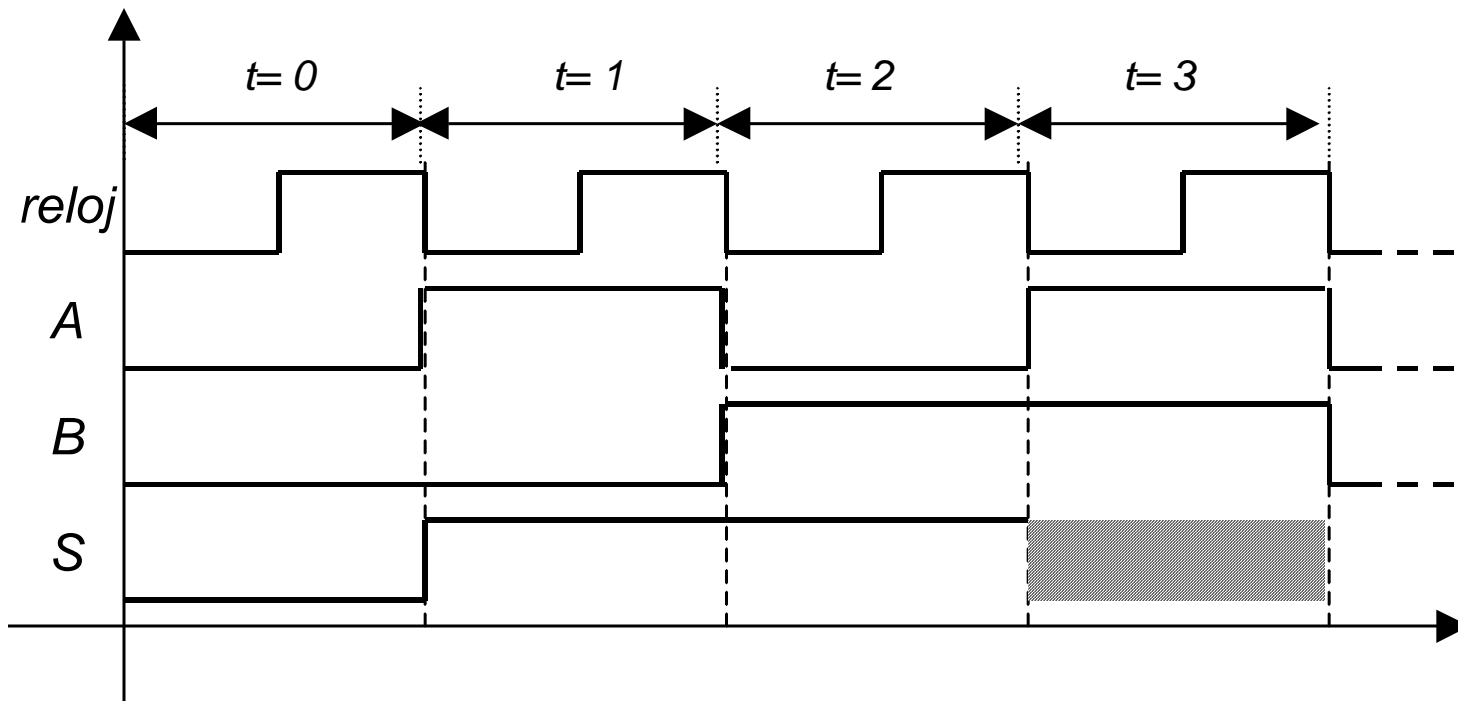
Reloj



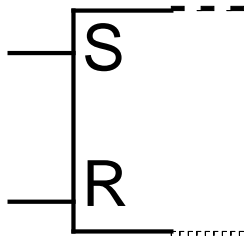
- La señal de reloj indica a los elementos de memoria cuándo deben cambiar su estado



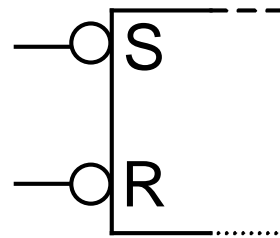
- Cronograma: Representación de la evolución temporal de las entradas y salidas de un circuito.
 - El valor indefinido se representa sombreado



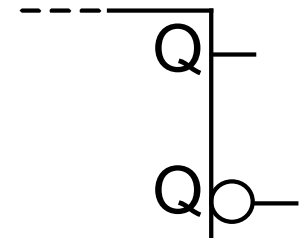
- Símbolos lógicos
 - Entradas y salidas



Entradas activas a nivel alto

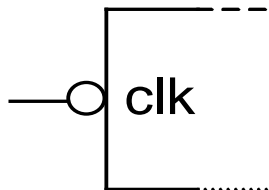


Entradas activas a nivel bajo

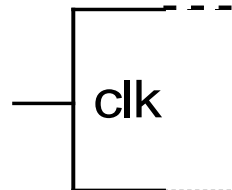


Salidas Q y /Q

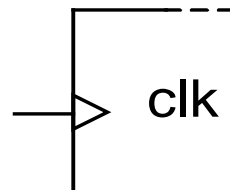
- Señal de reloj



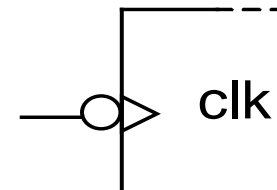
Activa a nivel bajo



Activa a nivel alto



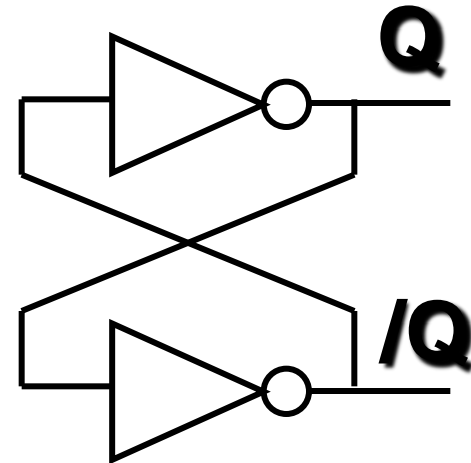
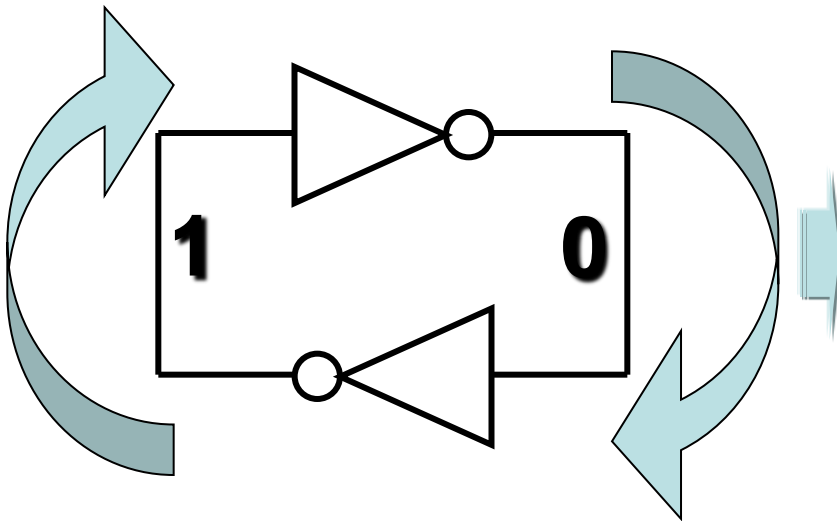
Activa por flanco de subida



Activa por flanco de bajada

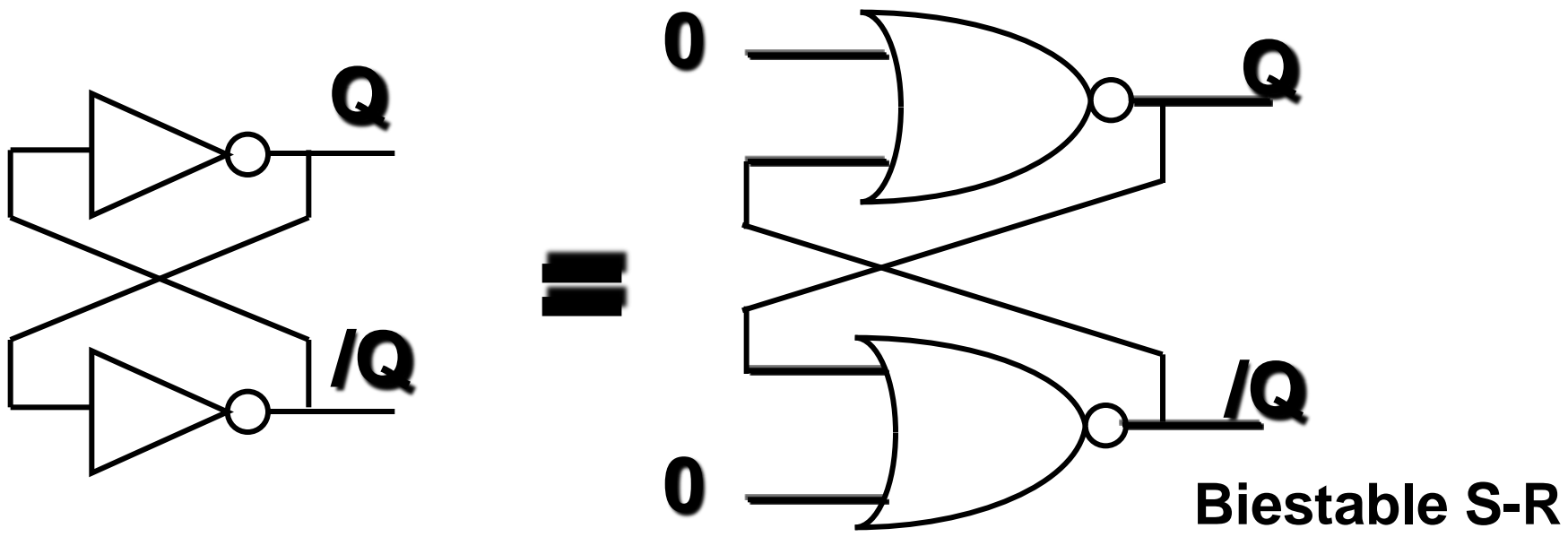
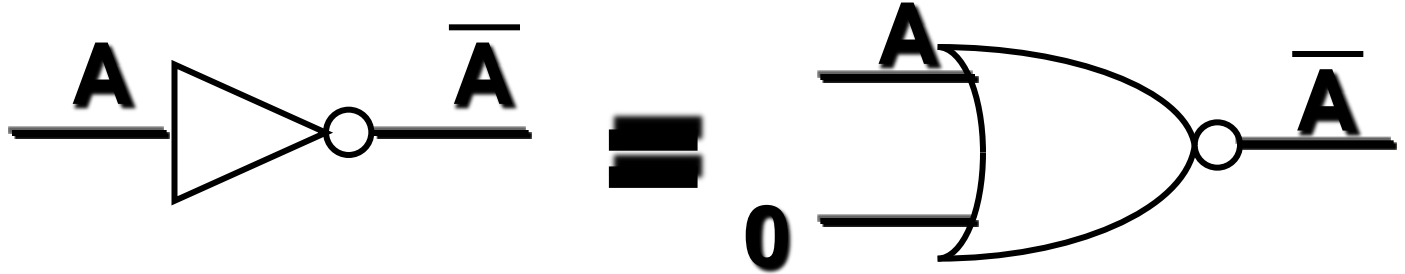
- Biestable: Circuito secuencial con dos estados estables (0 y 1)

Podemos almacenar un bit en un circuito haciendo que de vueltas y vueltas, de manera que la información recircule indefinidamente



No podemos modificar el estado

¿Cómo cambiar el estado del biestable?



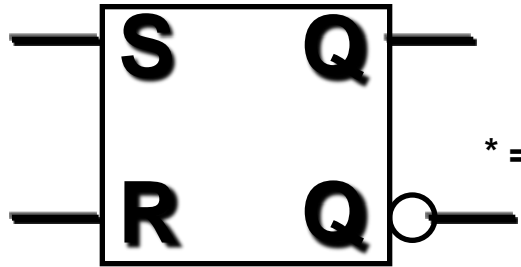
- Biestable R-S con puertas NOR
 - Activar entrada S para hacer $Q = 1$
 - Activar entrada R para hacer $Q = 0$
- Si $S=R=0$:
$$Q = \overline{\overline{Q} + 0} = \overline{\overline{Q}}$$
$$\overline{Q} = \overline{Q + 0} = \overline{\overline{Q}}$$
 y el estado se mantiene
- Si $S=0, R=1$:
$$Q = \overline{\overline{Q} + 1} = \overline{1} = 0$$
$$\overline{Q} = \overline{Q + 0} = \overline{\overline{Q}} = 1$$
 y el estado se pone a 0
- Si $S=1, R=0$:
$$\overline{Q} = \overline{Q + 1} = \overline{1} = 0$$
$$Q = \overline{\overline{Q} + 0} = \overline{0} = 1$$
 y el estado se pone a 1

- Biestable R-S con puertas NOR
 - Si se activan al mismo tiempo S y R, llegamos a una situación no deseada
 - Es el único caso en el que $\neg Q \neq \overline{Q}$

- Si $S=R=1$: $Q = \overline{\neg Q + 1} = \overline{1} = 0$
 $\neg Q = \overline{Q + 1} = \overline{1} = 0$

y si después hacemos $S=R=0$, dado que $\neg Q \neq \overline{Q}$, el biestable evolucionará a un estado indefinido

- Biastable R-S con puertas NOR



Símbolo lógico

Tabla de verdad

* =situación no deseada

S	R	Q(t)	Q(t+1)	/Q(t+1)
0	0	0	0	1
0	0	1	1	0
0	1	0	0	1
0	1	1	0	1
1	0	0	1	0
1	0	1	1	0
1	1	0	0*	0*
1	1	1	0*	0*

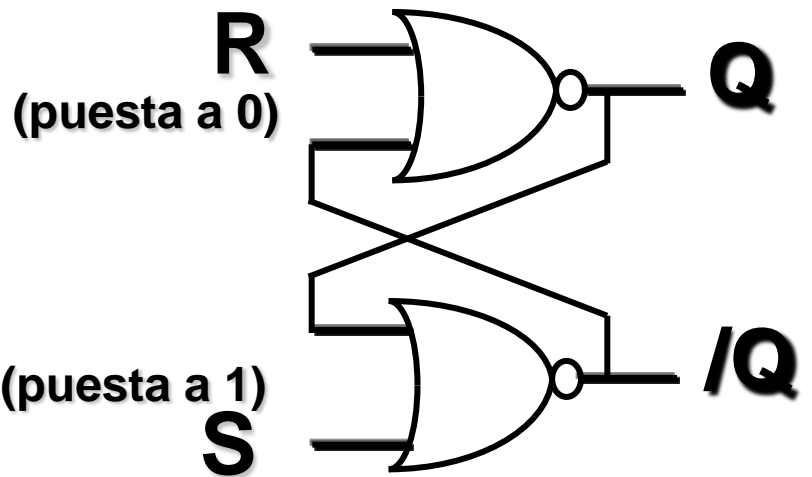


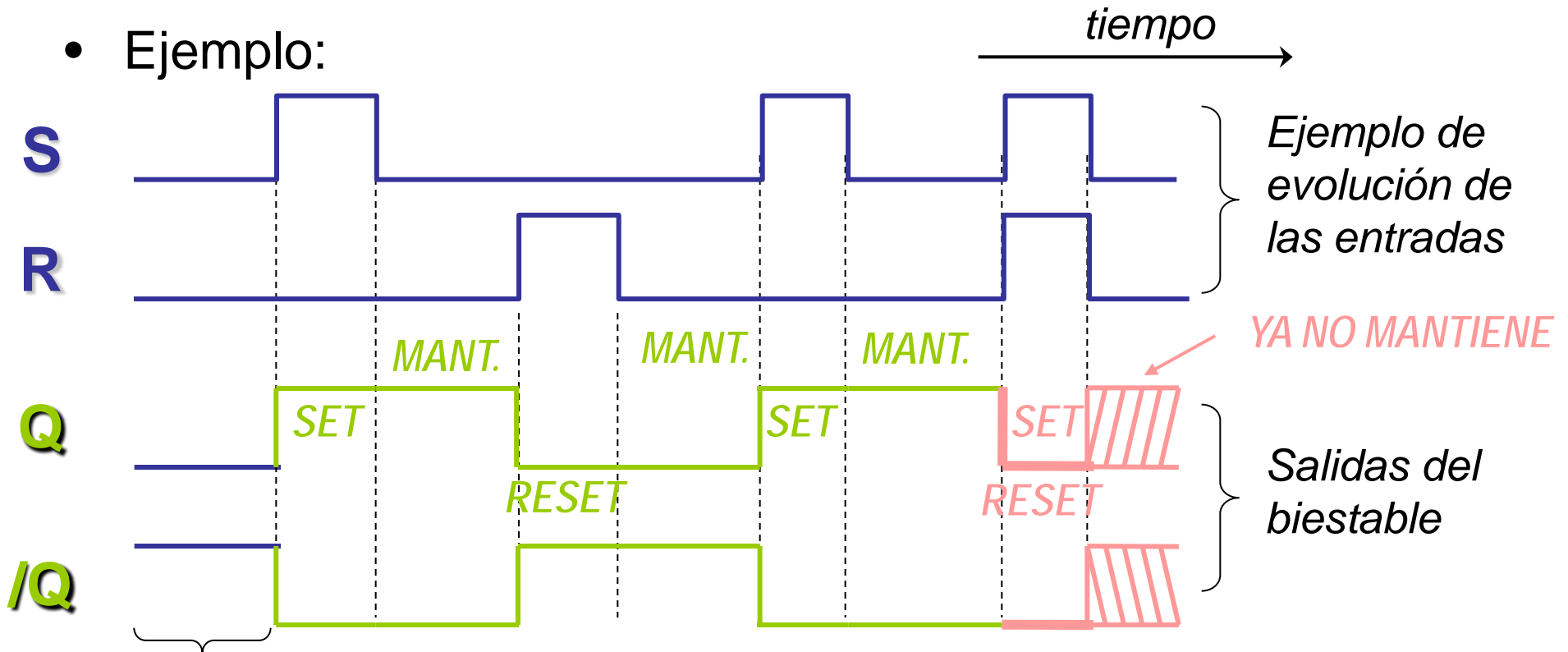
Diagrama lógico

Tabla de verdad reducida

* =situación no deseada

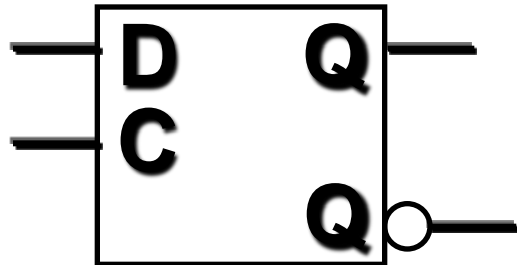
S	R	Q(t+1)	/Q(t+1)
0	0	Q(t)	/Q(t)
0	1	0	1
1	0	1	0
1	1	0*	0*

- El cronograma se utiliza para conocer la evolución temporal del estado cuando cambian sus entradas
- Ejemplo:



Valor inicial de Q y /Q (supuesto)

- Se utilizan para la implementación de elementos de memoria, cuya única finalidad es almacenar el valor de una línea de información (un bit)



Símbolo lógico

C	D	$Q(t+1)$	$/Q(t+1)$
0	X	$Q(t)$	$/Q(t)$
1	1	1	0
1	0	0	1

Tabla de verdad

- Podemos construir un biestable D a partir de un RS.

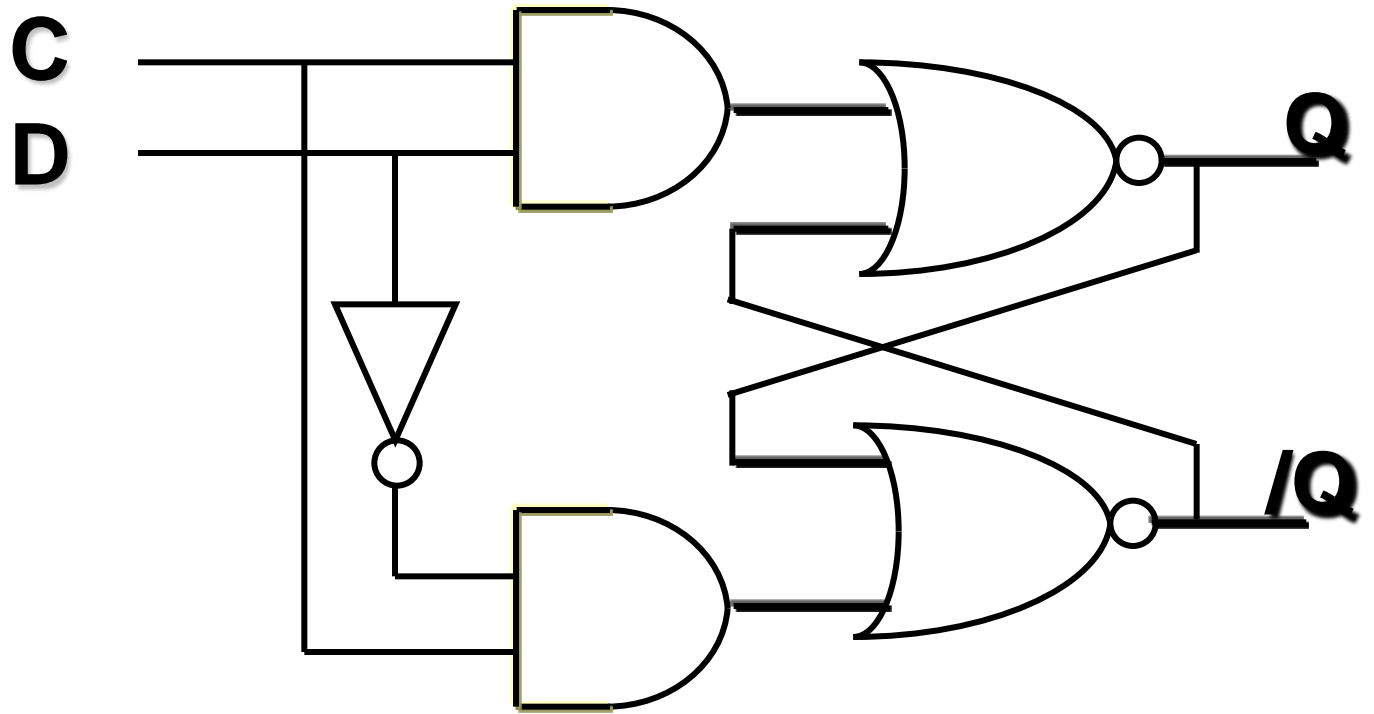
C	D	Q(t+1)	/Q(t+1)	R	S
0	X	Q(t)	/Q(t)	0	0
1	1	1	0	0	1
1	0	0	1	1	0

- Queremos que el estado se mantenga cuando $C=0$ ($R=S=0$)
- $Q(t+1)=1$ si $C=1$ y $D=1$ ($R=0$ y $S=1$)
- $Q(t+1)=0$ si $C=1$ y $D=0$ ($R=1$ y $S=0$)

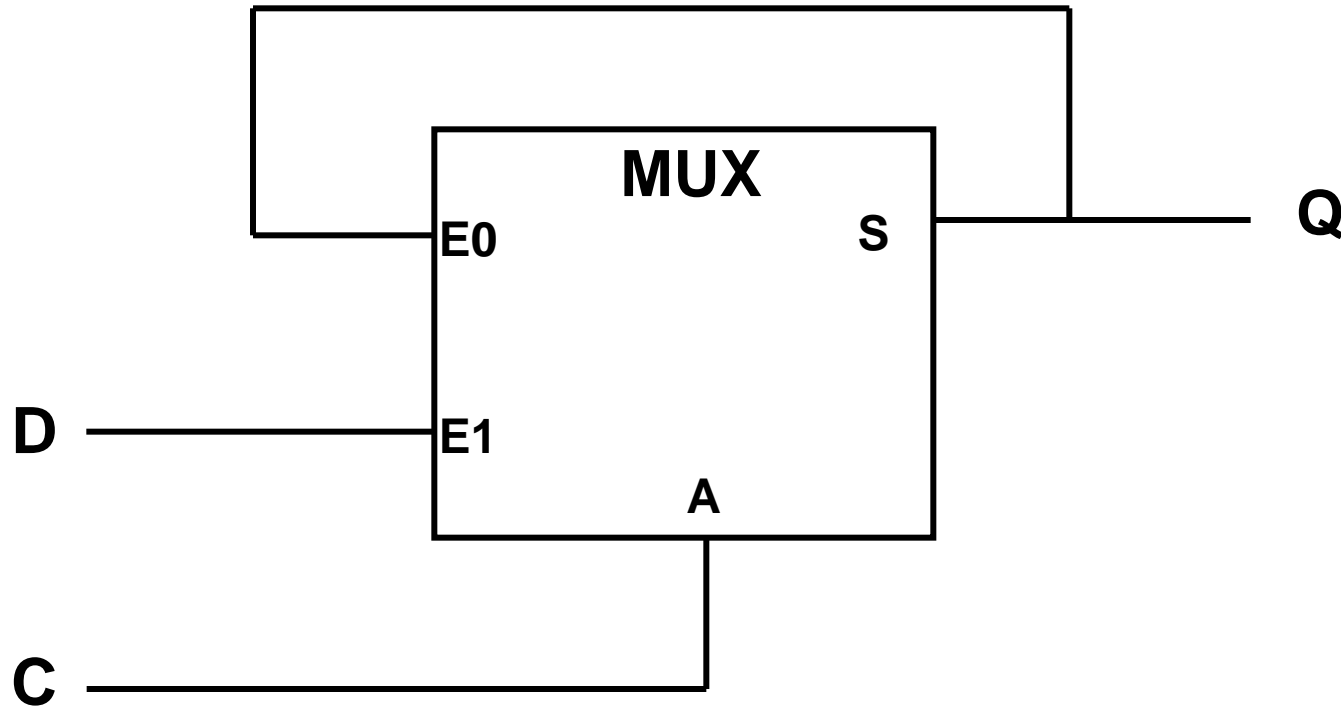
$$R=C\bar{D}$$

$$S=CD$$

- Una posible implementación del biestable D:



- Otra posible implementación del biestable D:

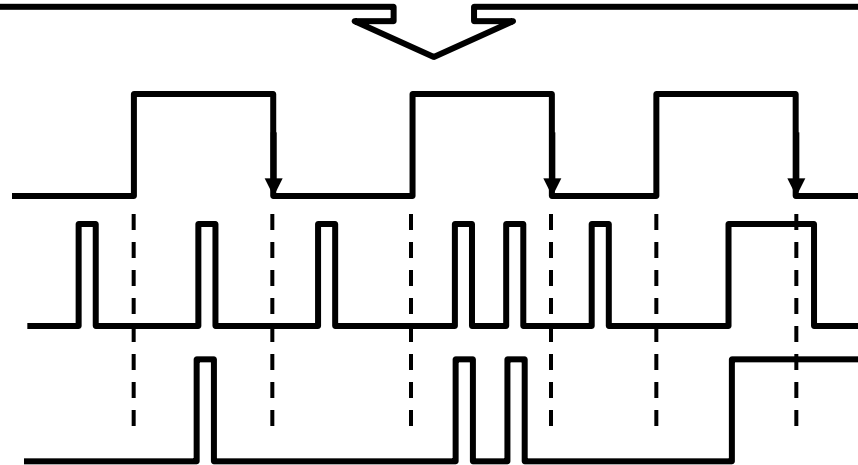


Si se activa el biestable por nivel
y hay un pulso no deseado
en la entrada de datos

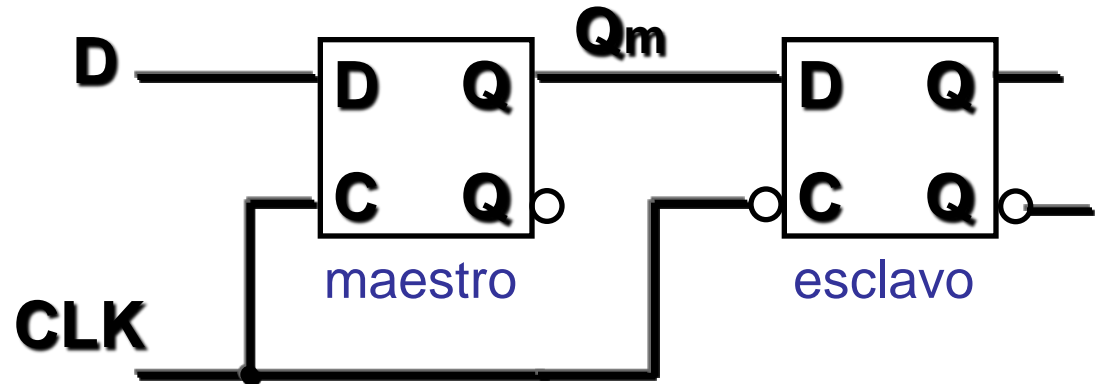
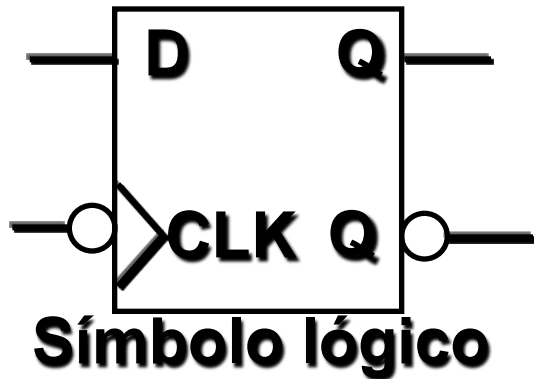


el pulso no deseado
se trasladará a la salida

Podemos hacer que la señal de reloj actúe por flanco



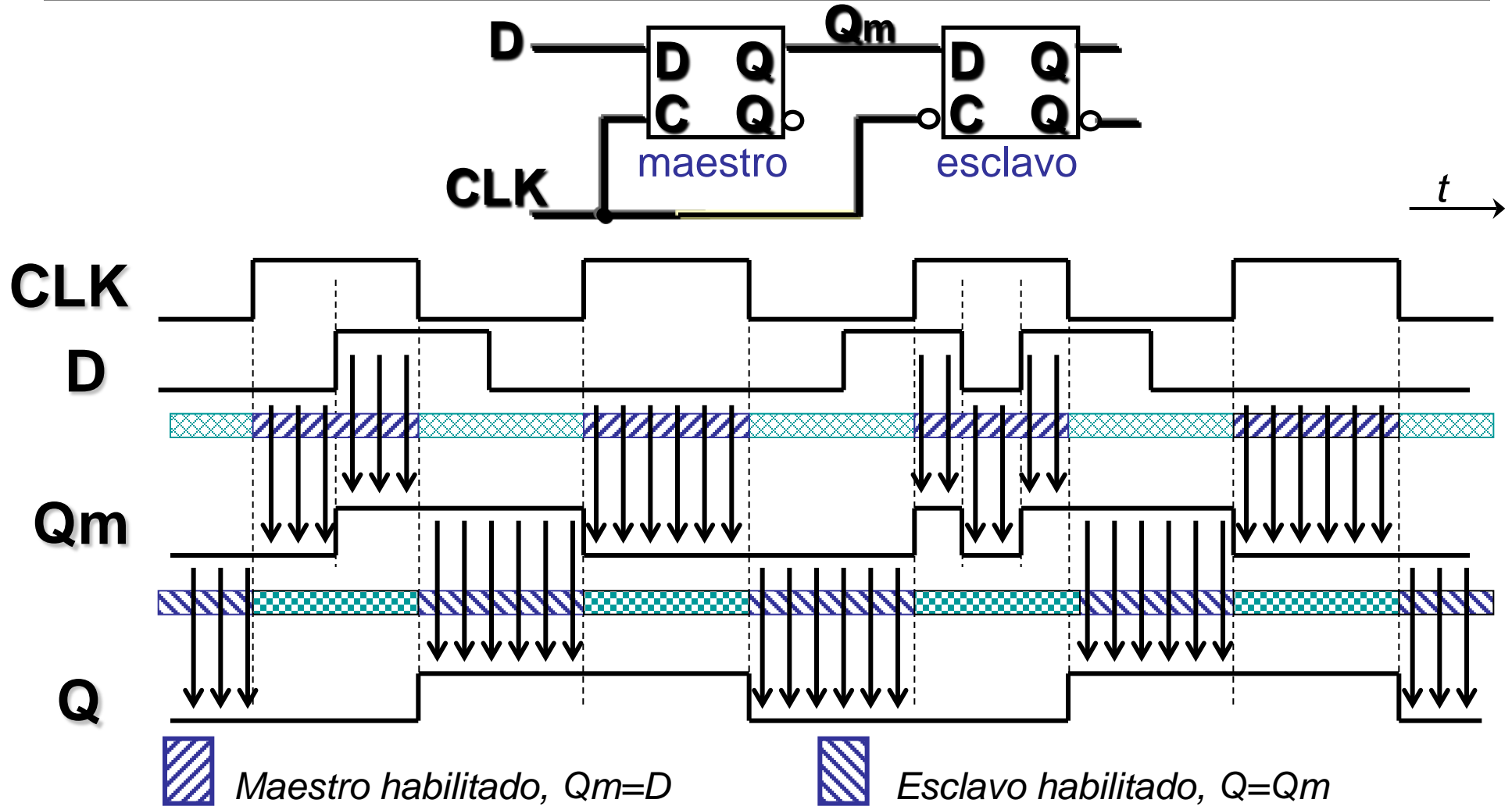
- Para implementar biestables que se activen por flanco se utiliza con frecuencia un par de biestables por nivel en configuración MASTER-SLAVE (maestro-esclavo)



CLK	D	Q(t+1)	/Q(t+1)
0	X	Q(t)	/Q(t)
1	X	Q(t)	/Q(t)
↓	1	1	0
↓	0	0	1

Tabla de verdad

Bi stable D por flanco de bajada (ii)

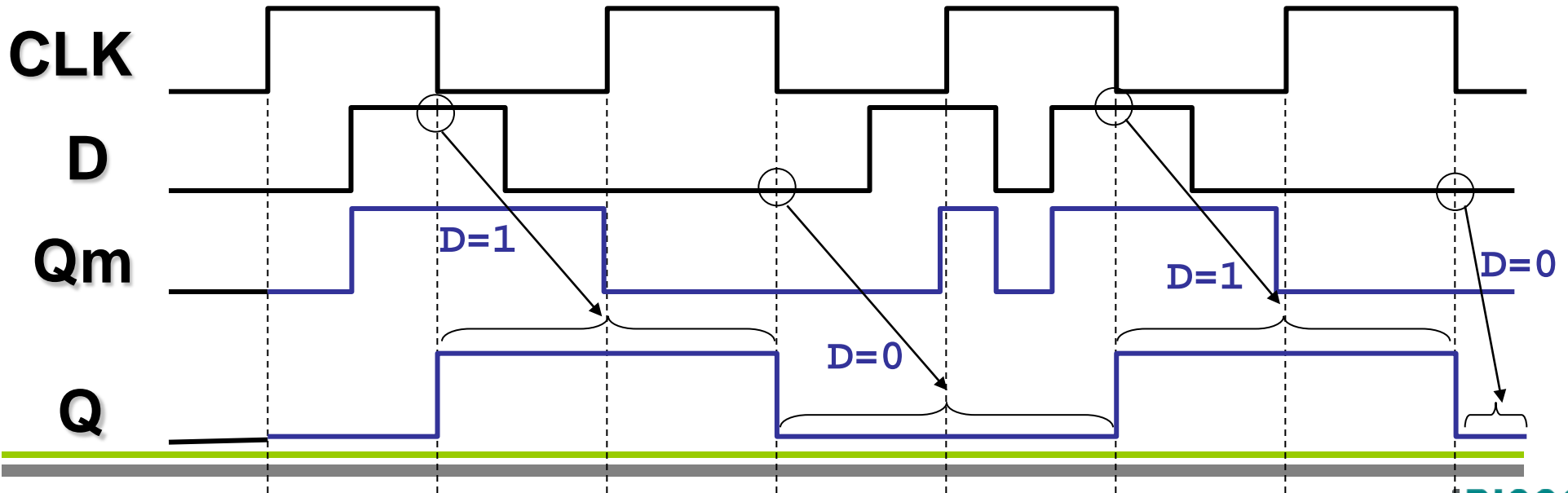


Maestro no habilitado



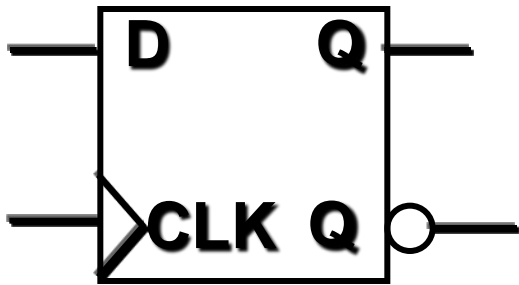
Esclavo no habilitado

- Como se puede observar:
 - La línea interna Qm cambia cuando CLK = 1, siguiendo la evolución de la entrada D
 - La salida Q sólo cambia en los flancos de CLK
- *El efecto neto de la configuración maestro-esclavo es que el último valor de la entrada D justo antes del flanco es el valor que aparece en la salida Q*



Biastable D por flanco de subida

Sólo es necesario añadir un inversor a la entrada CLK del biastable D activo por flanco de bajada



Símbolo lógico

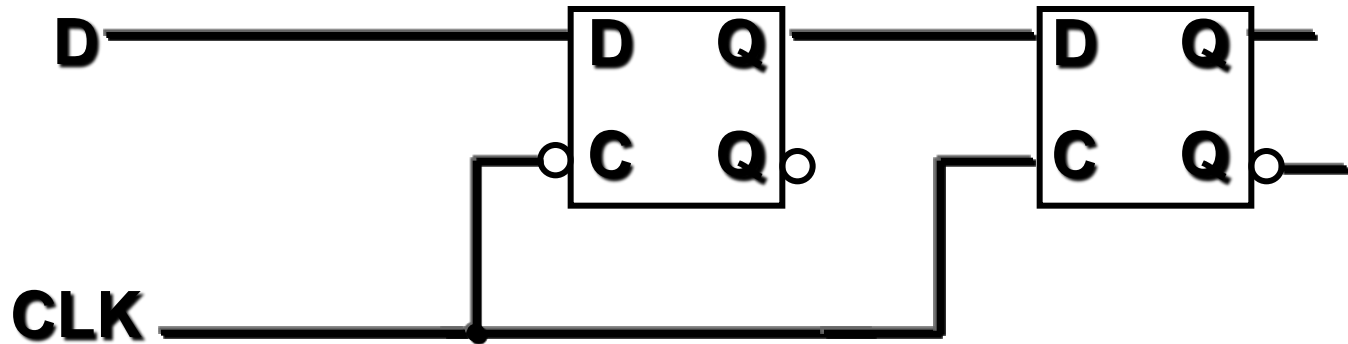
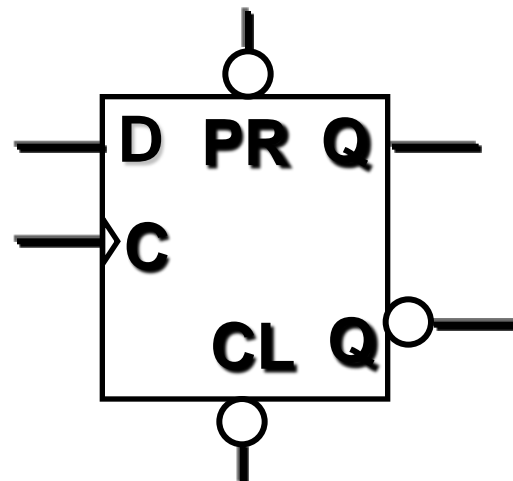


Diagrama lógico

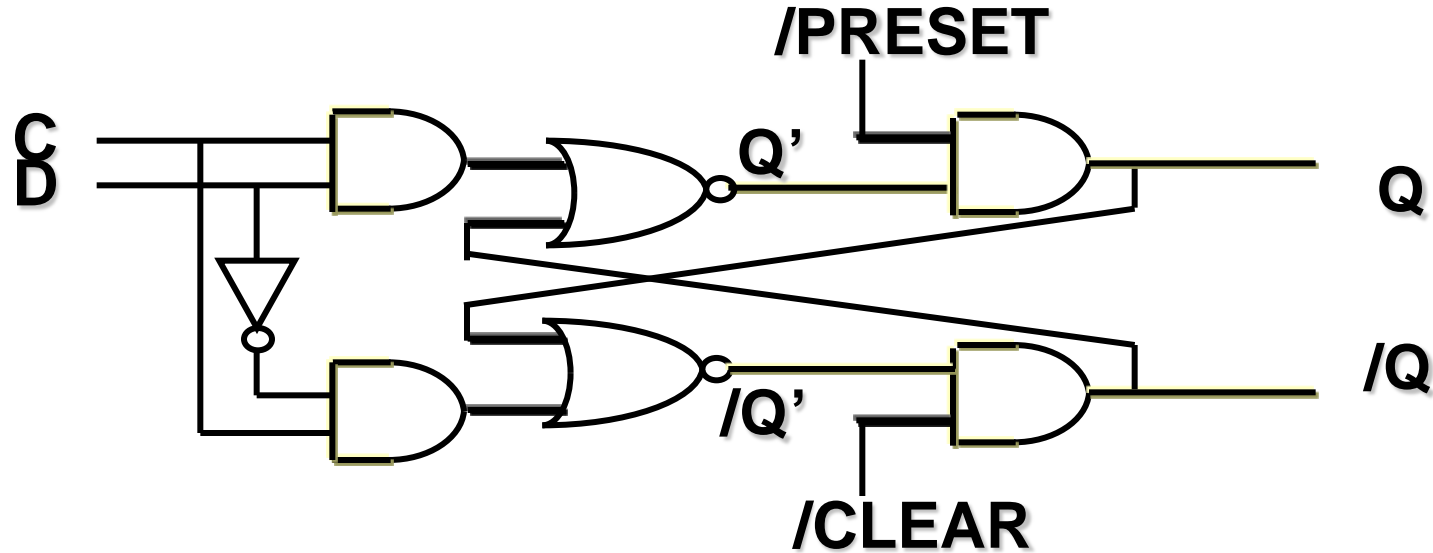
CLK	D	Q(t+1)	/Q(t+1)
0	X	Q(t)	/Q(t)
1	X	Q(t)	/Q(t)
↑	1	1	0
↑	0	0	1

Tabla de verdad

- Biastable D con habilitación y con entradas asíncronas
 - /CLEAR: puesta a cero asíncrona
 - /PRESET: puesta a uno asíncrona
- Tienen prioridad sobre la señal de reloj y permiten poner el estado a uno o a cero



Símbolo lógico

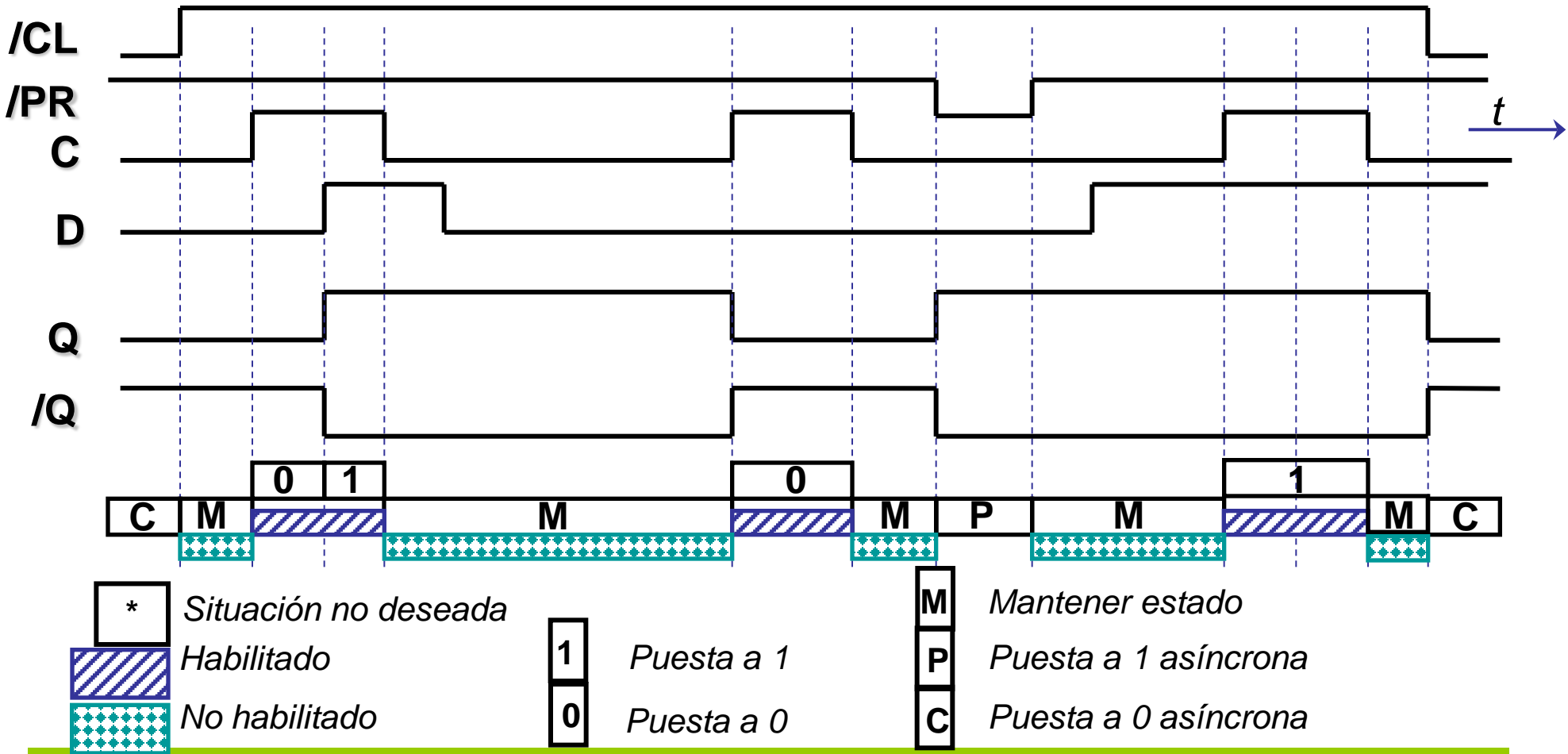


- Si $\text{/CLEAR}=\text{/PRESET}=1 \Rightarrow Q = Q', \text{/Q} = \text{/Q}'$
- Si $\text{/CLEAR}=0, \text{/PRESET}=1 \Rightarrow Q = 0, \text{/Q}' = 1, \text{/Q} = 1$
- Si $\text{/CLEAR}=1, \text{/PRESET}=0 \Rightarrow \text{/Q} = 0, Q' = 1, Q = 1$
- Si $\text{/CLEAR}=0, \text{/PRESET}=0 \Rightarrow Q=\text{/Q}=0^*, Q'=\text{/Q}'=1^*$

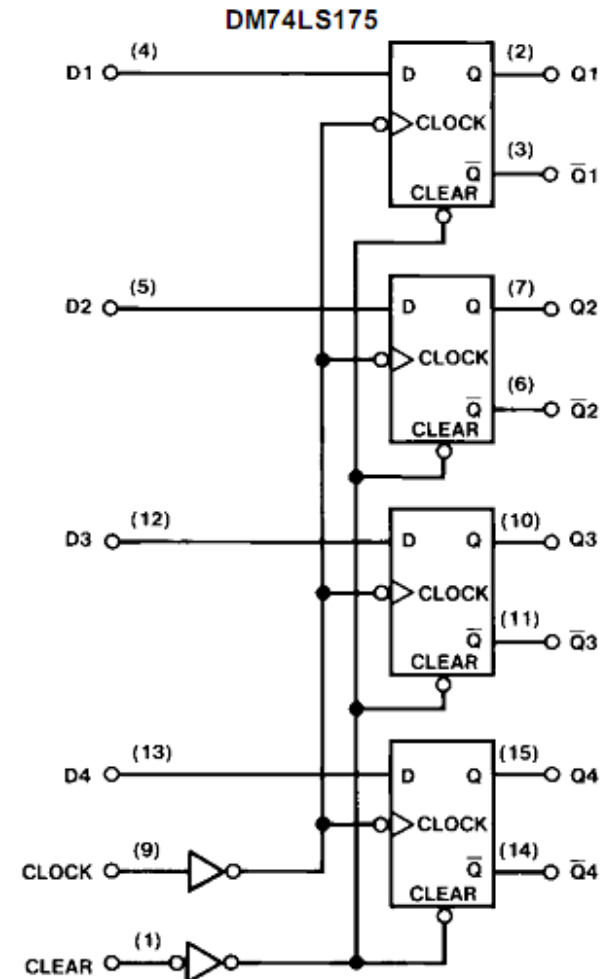
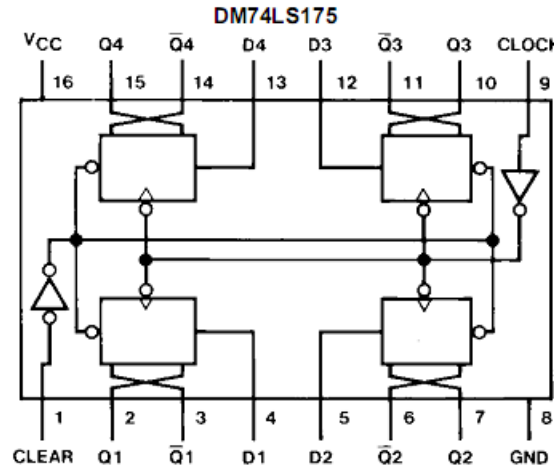
- Biestable D con habilitación y entradas asíncronas
 - Tabla de verdad

/PR	/CL	C	D	Q(t+1)	/Q(t+1)
0	1	X	X	1	0
1	0	X	X	0	1
0	0	X	X	0*	0*
1	1	1	1	1	0
1	1	1	0	0	1
1	1	0	X	Q(t)	/Q(t)

- Biestable D con habilitación y entradas asíncronas



- Circuito integrado '175
- 4 biestables tipo D, activos por flanco de subida con entrada asíncrona de puesta a 0



Function Table

(Each Flip-Flop)

Inputs			Outputs	
Clear	Clock	D	Q	$\bar{Q} \uparrow$
L	X	X	L	H
H	\uparrow	H	H	L
H	\uparrow	L	L	H
H	L	X	Q_0	\bar{Q}_0

H = HIGH Level (steady state)

L = LOW Level (steady state)

X = Don't Care

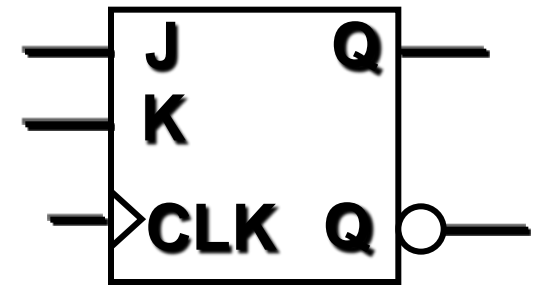
\uparrow = Transition from LOW-to-HIGH level

Q_0 = The level of Q before the indicated steady-state input conditions were established.

- El biastable RS presenta problemas cuando se activan simultáneamente las dos entradas S y R
- Podemos diseñar un biastable similar que no presente problemas a partir de un biastable D:
 - Entrada J o de puesta a 1; cuando se active, $Q(t+1) = 1$
 - Entrada K o de puesta a 0; cuando se active, $Q(t+1) = 0$
 - Si no se activa ninguna, mantener el estado: $Q(t+1) = Q(t)$
 - Si se activan al tiempo, invertir el estado: $Q(t+1) = \neg Q(t)$

CLK	J	K	$Q(t+1)$	$\neg Q(t+1)$
0	X	X	$Q(t)$	$\neg Q(t)$
1	X	X	$Q(t)$	$\neg Q(t)$
↑	0	0	$Q(t)$	$\neg Q(t)$
↑	0	1	0	1
↑	1	0	1	0
↑	1	1	$\neg Q(t)$	$Q(t)$

Tabla de verdad



Símbolo lógico

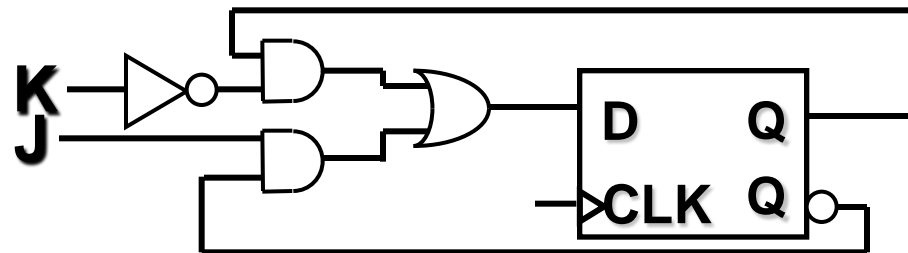
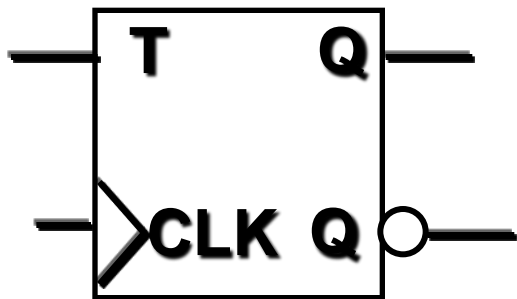


Diagrama lógico

- Sólo tiene una entrada llamada T (*toggle*)
- Mantiene el estado (si T=0) o lo cambia (si T=1) cada vez que llega un flanco de reloj
- No se construye comercialmente, pero se puede implementar fácilmente utilizando un D



Símbolo lógico

T	Q(t)	D
0	0	0
0	1	1
1	0	1
1	1	0

Tabla de verdad

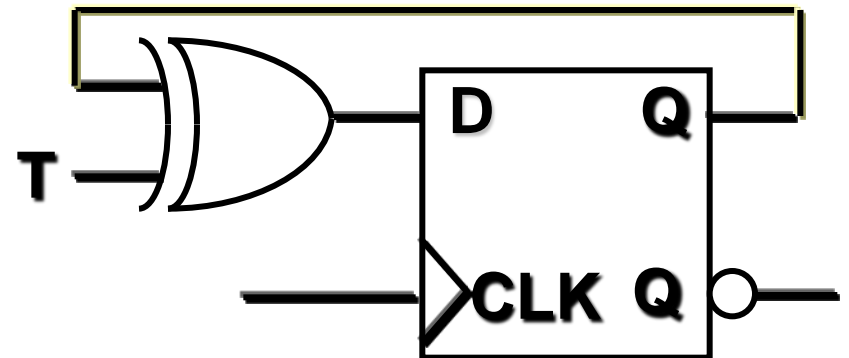


Diagrama lógico

De manera análoga, se puede construir un biestable T activado por flanco de bajada usando un biestable D activo por flanco de bajada