



UNIVERSIDAD  
POLITECNICA  
DE VALENCIA



etsinf



---

# Fundamentos de computadores

---

## TEMA 4. CIRCUITOS SECUENCIALES

---

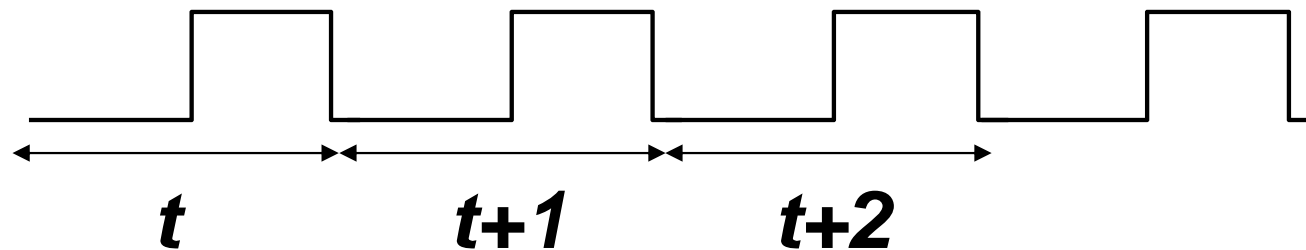
- Estudiar los circuitos secuenciales más sencillos.
- Introducir el concepto de cronograma.
- Comprender el funcionamiento básico de los biestables.
- Conocer los sistemas secuenciales básicos más importantes
  - Registros, Banco de registros, Memoria, Contadores

- Introducción
  - Circuitos secuenciales, reloj, cronogramas, símbolos lógicos
- Biestables
  - Biestable S-R
  - Biestable D
    - Activo por nivel
    - Activo por flanco
    - Con entradas asíncronas
  - Biestable J-K
  - Biestable T

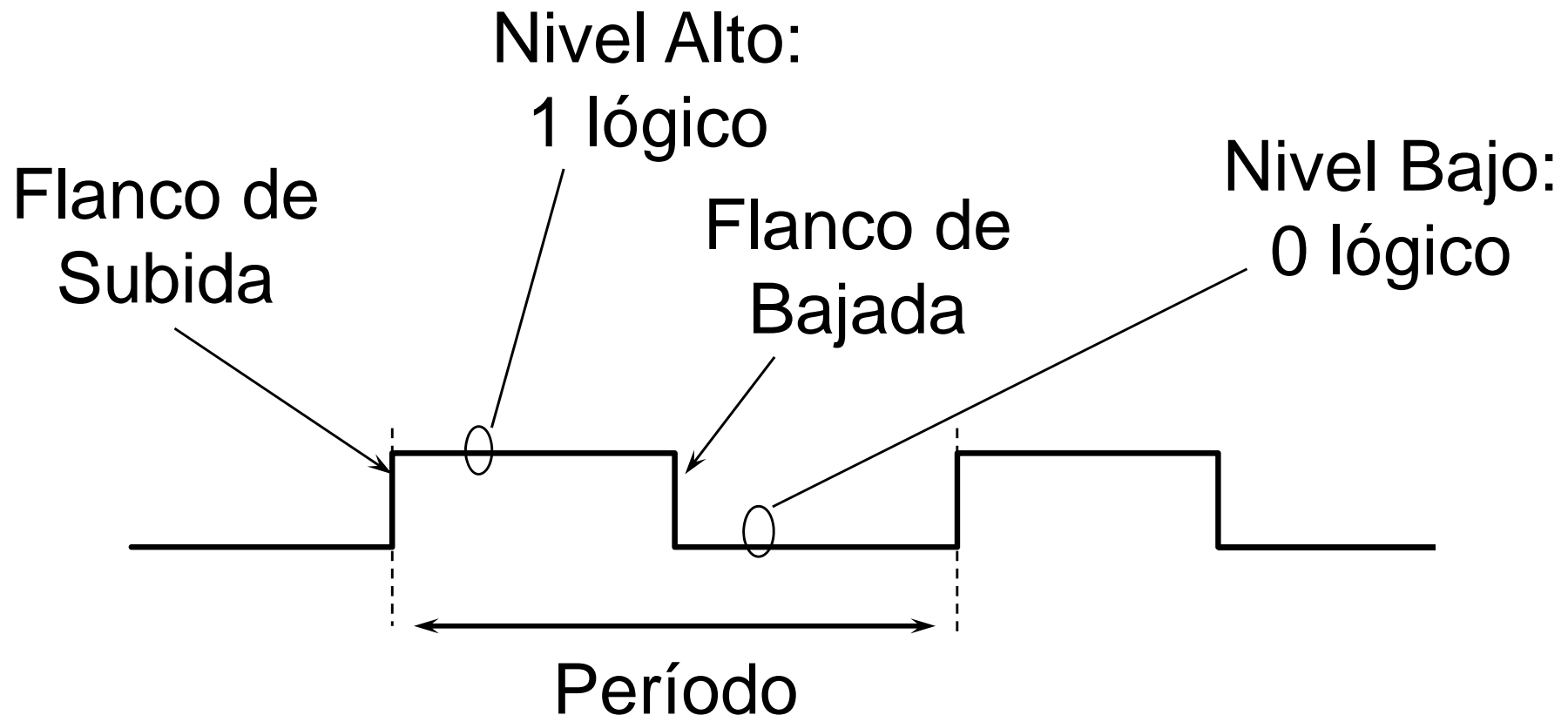
- Bloques secuenciales básicos
  - Registros de almacenamiento
  - Banco de Registros, Memoria
  - Registros de desplazamiento
  - Contadores
- Análisis de SS por cronograma

- 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 internamente el estado del sistema

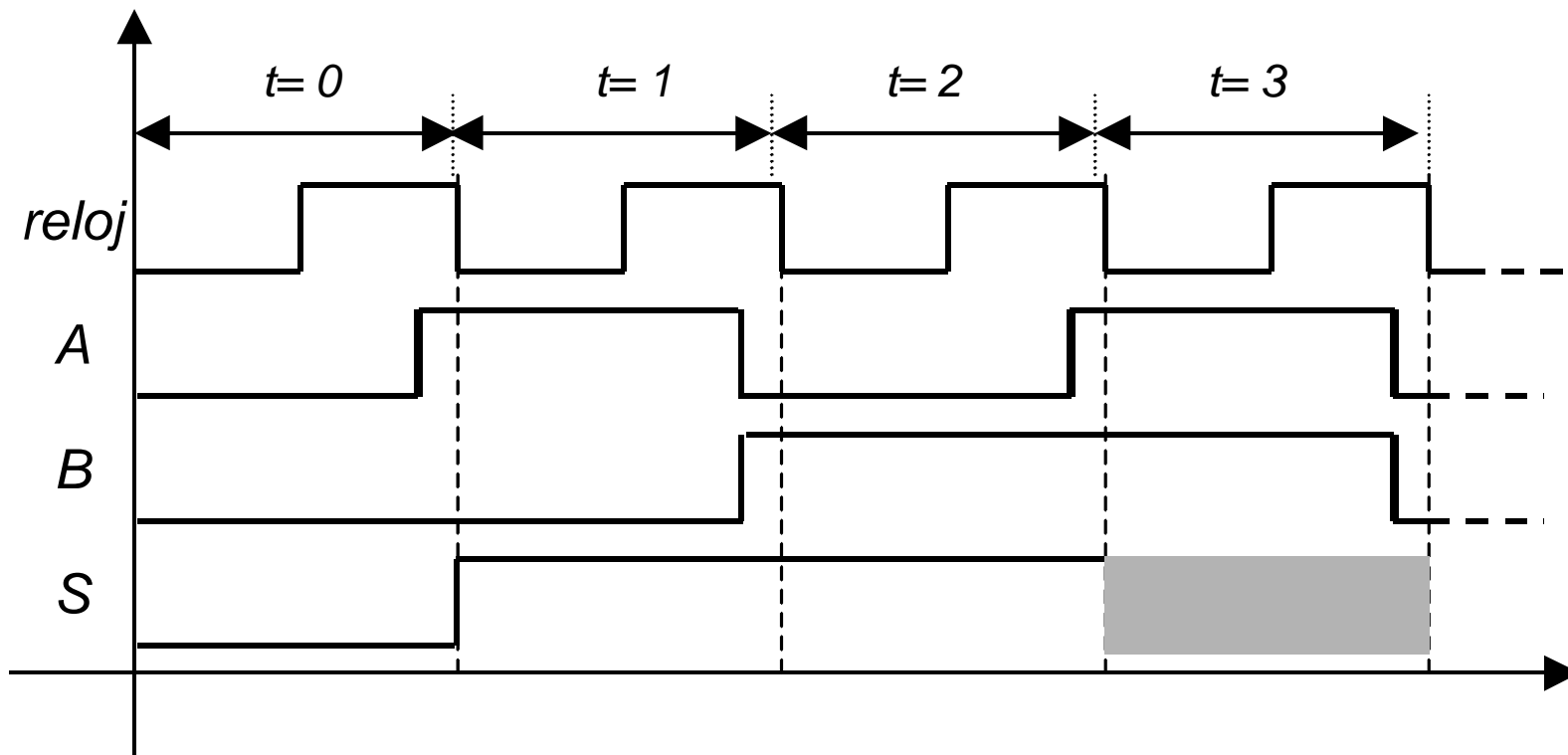
**Reloj**



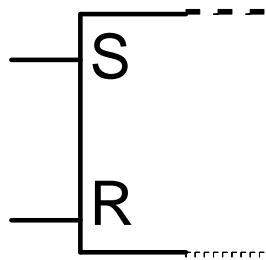
- La señal de reloj indica a los elementos de memoria cuándo deben cambiar su estado



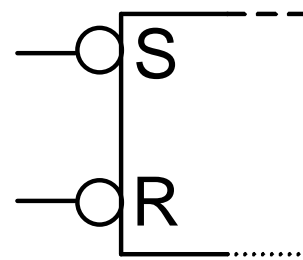
- Cronograma: Representación de la evolución temporal de las entradas y salidas de un circuito.
  - Los valores desconocidos se representan sombreados



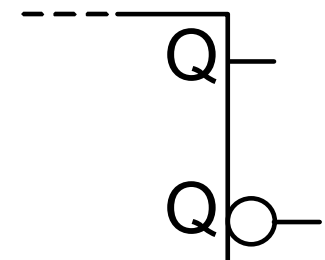
- Símbolos lógicos
  - Entradas y salidas



Entradas activas a nivel alto

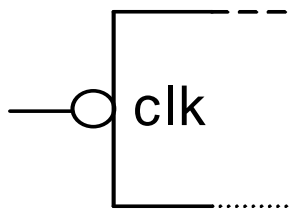


Entradas activas a nivel bajo

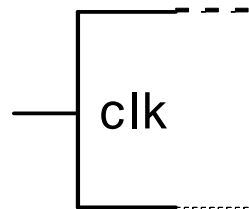


Salidas Q y /Q

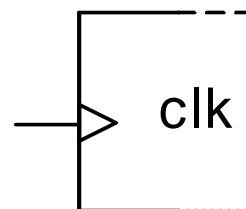
- Señal de reloj



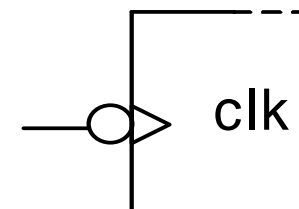
Activa a nivel bajo



Activa a nivel alto



Activa por flanco de subida

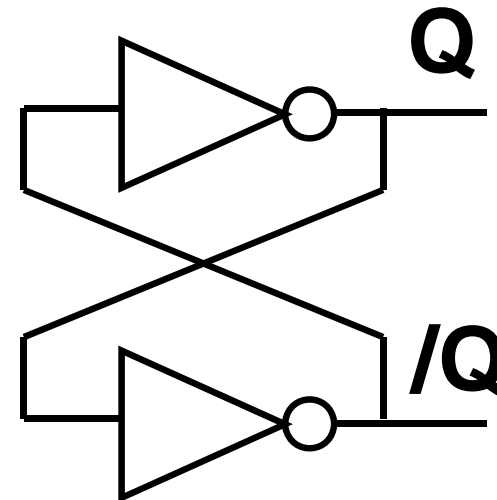
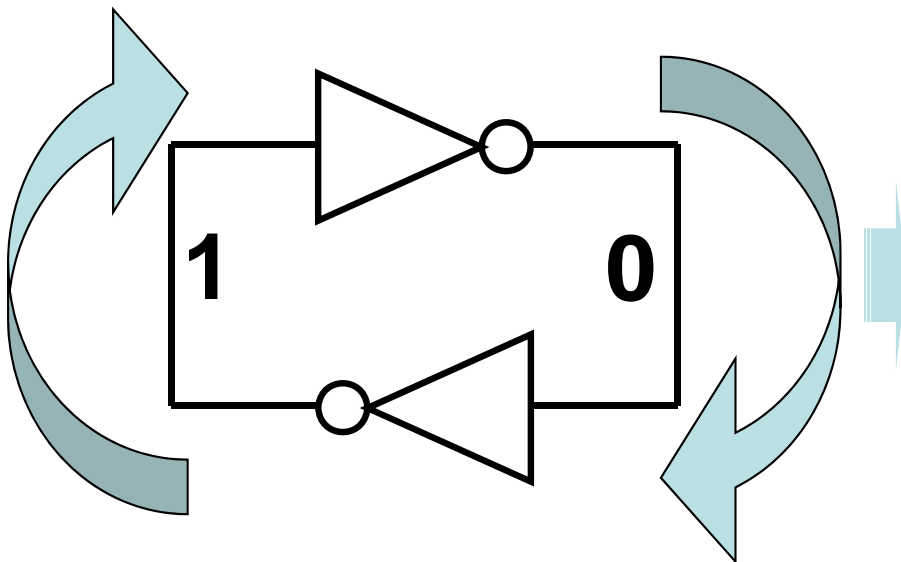


Activa por flanco de bajada



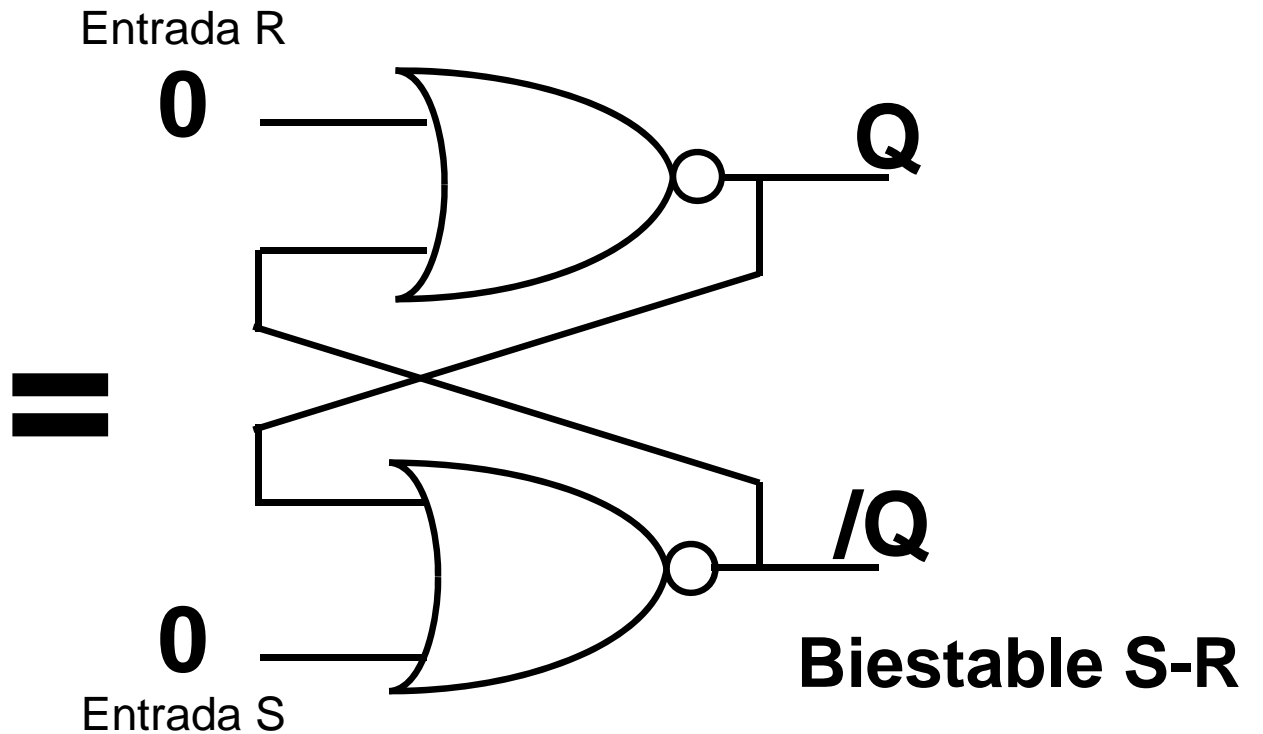
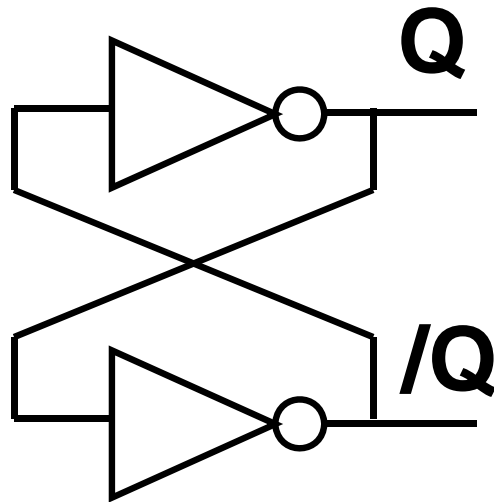
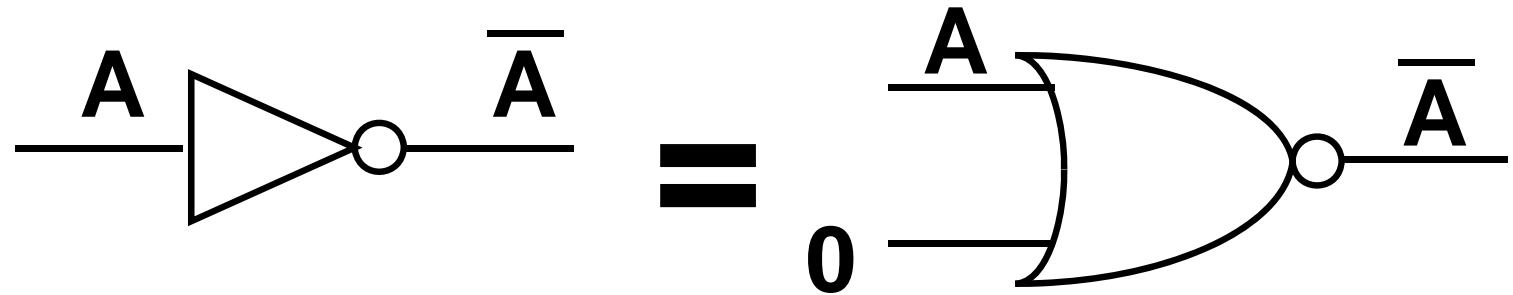
- Biestable: Circuito secuencial con dos estados estables (0 y 1)

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?



- Biastable S-R con puertas NOR
  - ¿Cómo se determina su comportamiento?
    1. Obtención de una tabla con todos los casos
      - A partir de las ecuaciones (sin realimentación)
      - Las entradas externas y el estado actual ( $t$ ) son las entradas y el estado siguiente ( $t+1$ ) son las salidas
    2. Análisis de dicha tabla
      - a) Los casos se agrupan para cada combinación de valores de las entradas externas
      - b) Se aplica la tabla una y otra vez (la realimentación del circuito) hasta que el estado permanece estable (siempre sin cambiar las entradas externas)
  - Este método sirve para cualquier biastable por nivel

- Tabla: ecuaciones  $Q(t+1) = R + \overline{Q(t)}$  y  $\overline{Q(t+1)} = S + Q(t)$

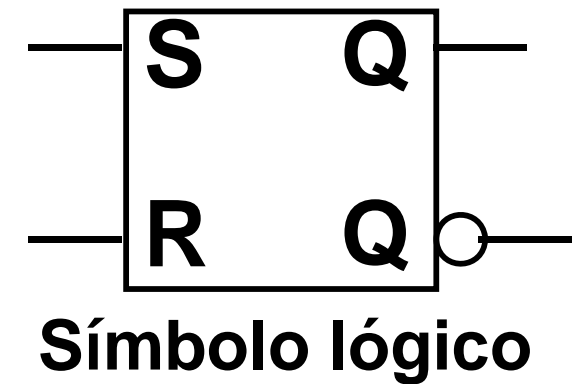
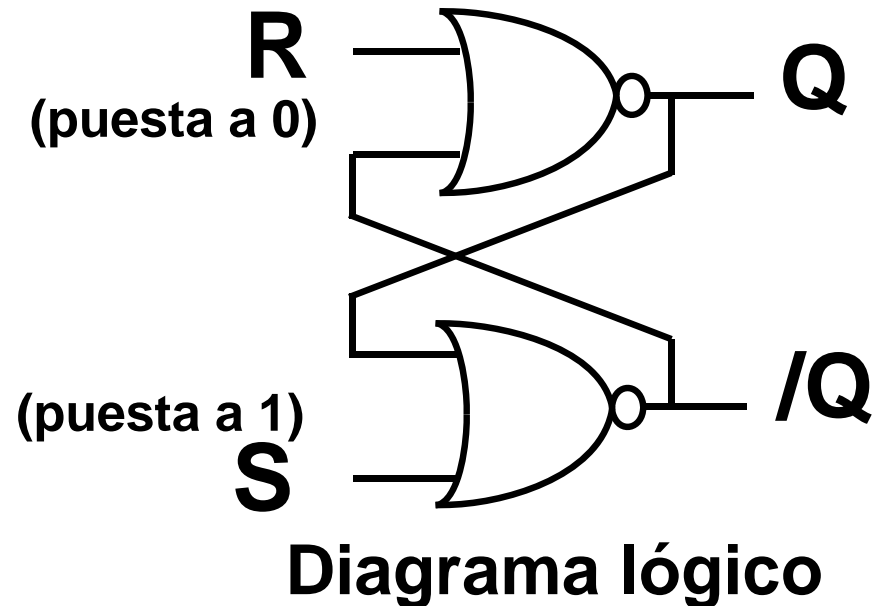
S	R	Q(t)	$\overline{Q(t)}$	Q(t+1)	$\overline{Q(t+1)}$
0	0	0	0	1	1
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	0	0
0	1	0	0	0	1
0	1	0	1	0	1
0	1	1	0	0	0
0	1	1	1	0	0
1	0	0	0	1	0
1	0	0	1	0	0
1	0	1	0	1	0
1	0	1	1	0	0
1	1	0	0	0	0
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	0	0

- Análisis: Agrupando por los casos de S y R

	S	R	Q(t)	/Q(t)	Q(t+1)	/Q(t+1)		Q(t+1)	/Q(t+1)
A0	0	0	0	0	1	1	→ A3 (Oscila)	Q(t)	/Q(t)
A1	0	0	0	1	0	1	→ A1 (Estable)		
A2	0	0	1	0	1	0	→ A2 (Estable)		
A3	0	0	1	1	0	0	→ A0 (Oscila)		
B0	0	1	0	0	0	1	→ B1	0	1
B1	0	1	0	1	0	1	→ B1 (Estable)		
B2	0	1	1	0	0	0	→ B0		
B3	0	1	1	1	0	0	→ B0		
C0	1	0	0	0	1	0	→ C2	1	0
C1	1	0	0	1	0	0	→ C0		
C2	1	0	1	0	1	0	→ C2 (Estable)		
C3	1	0	1	1	0	0	→ C0		
D0	1	1	0	0	0	0	→ D0 (Estable)	0*	0*
D1	1	1	0	1	0	0	→ D0		
D2	1	1	1	0	0	0	→ D0		
D3	1	1	1	1	0	0	→ D0		

- Biastable S-R con puertas NOR
  - Comentarios a los valores de resumen del análisis
    1. Cuando el estado siguiente (los valores de  $Q(t+1)$  y  $/Q(t+1)$ ) es el mismo para todos los casos agrupados, resumir es fácil
      - Se indica el valor numérico (casos  $Bn$ ,  $Cn$  y  $Dn$ )
      - En el caso ( $Dn$ ) en el que los valores resumen cumplan  $Q(t+1) = /Q(t+1)$  se indica con un \* que la situación es “no deseada” ya que **no** se cumple la condición habitual  $/Q(t+1) = \overline{Q(t+1)}$
    2. Cuando los valores de estado siguiente de los casos agrupados es diferente (casos  $An$ ), el resumen es más difícil
      - Se expresa  $Q(t+1)$  y  $/Q(t+1)$  a partir de los valores de  $Q(t)$  y  $/Q(t)$
      - **La oscilación** (el estado siguiente pasa de 00 a 11 y de 11 a 00 de forma indefinida) **no aparece en el resumen** (casos  $A0$  y  $A3$ )

- Biastable S-R con puertas NOR

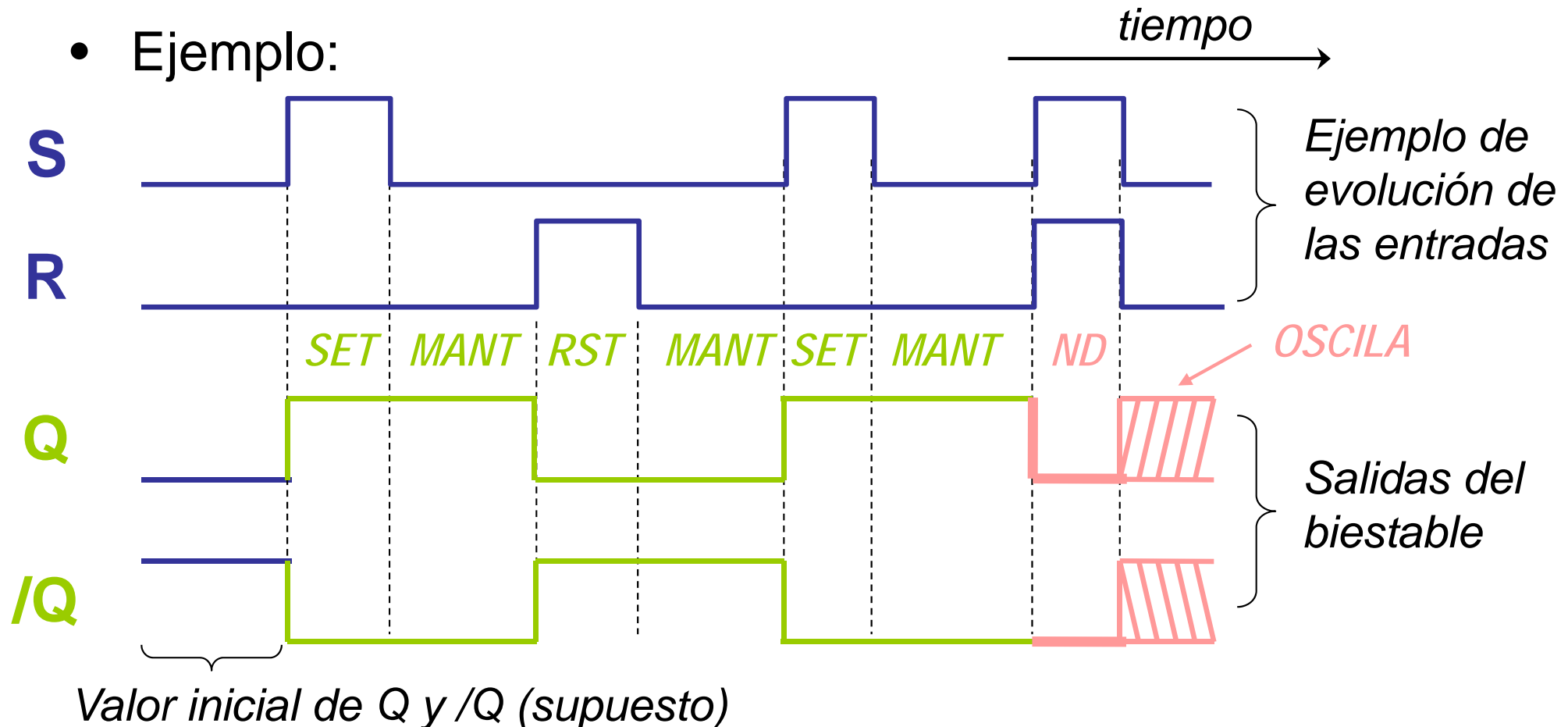


S	R	Q(t+1)	/Q(t+1)
0	0	Q(t)	/Q(t)
0	1	0	1
1	0	1	0
1	1	0*	0*

Tabla de funcionamiento

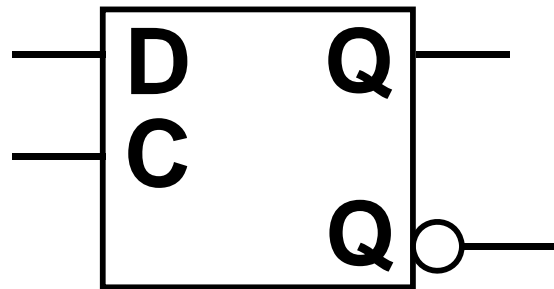
\* =situación no deseada

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





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

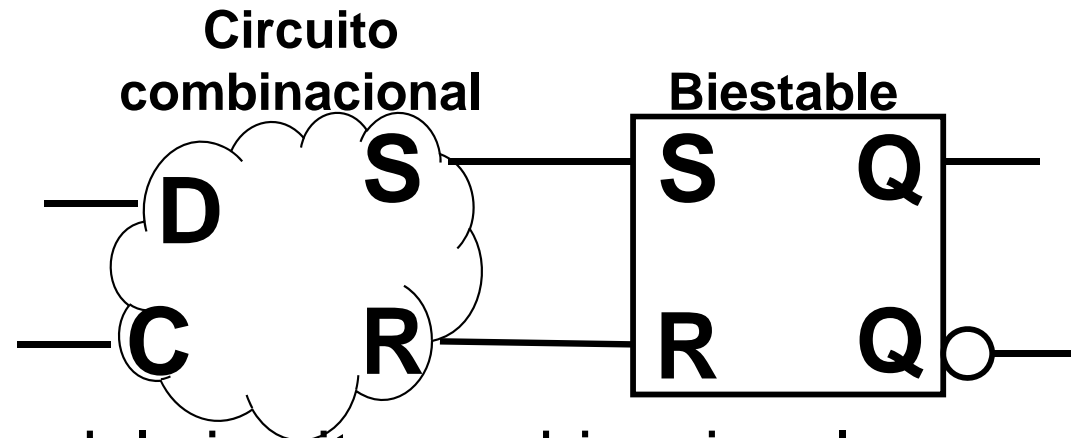


Símbolo lógico

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

Tabla de funcionamiento

- Podemos construir un biastable D a partir de un S-R



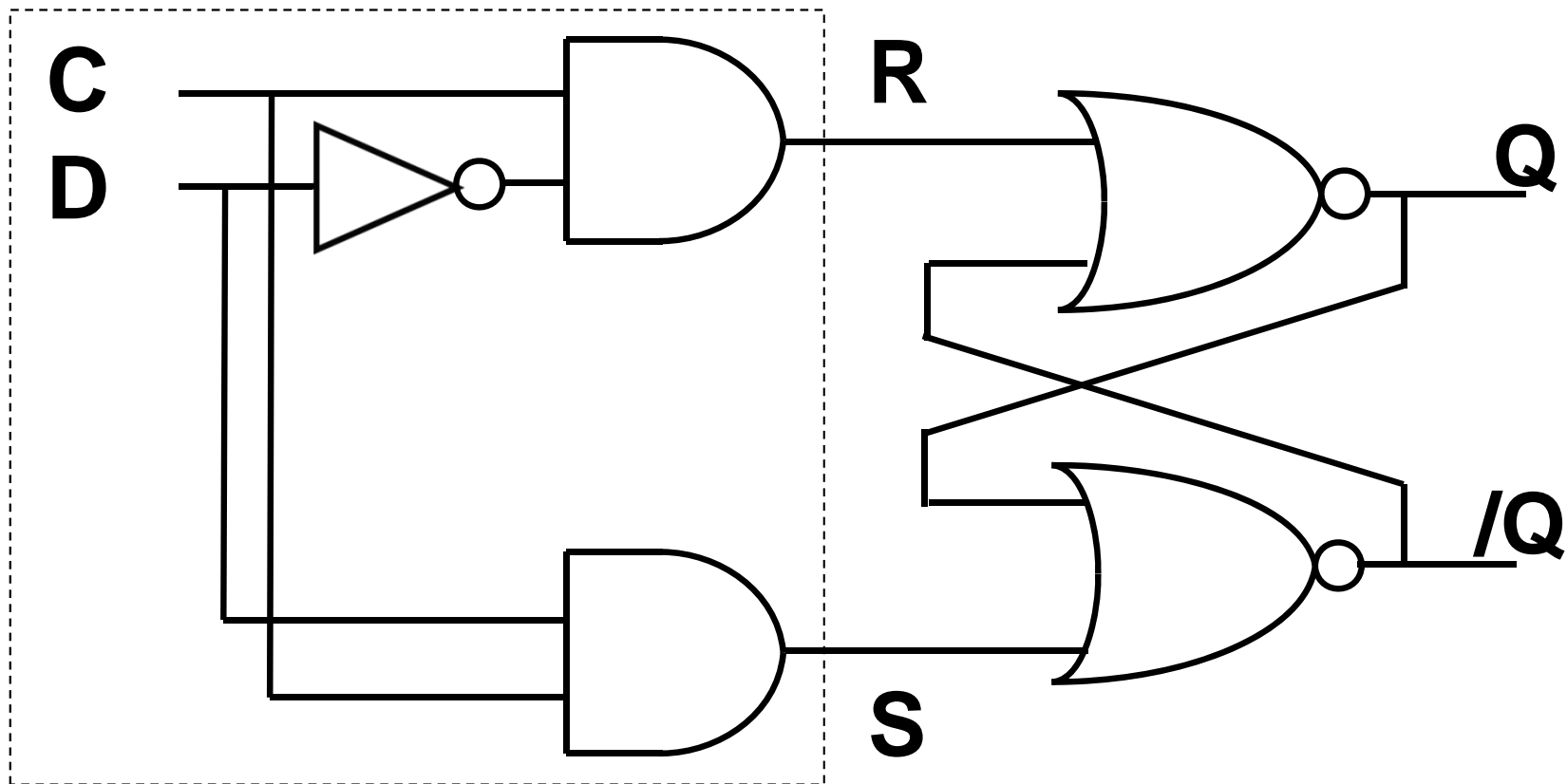
- Especificación del circuito combinacional

- Si  $C=0$ , queremos que  $Q(t+1) = Q(t)$ , por tanto la salida debe ser  $S=R=0$
- Si  $C=1$  y  $D=0$ , queremos que  $Q(t+1) = 0$ , por tanto la salida debe ser  $S=0$  y  $R=1$
- Si  $C=1$  y  $D=1$ , queremos que  $Q(t+1) = 1$ , por tanto la salida debe ser  $S=1$  y  $R=0$

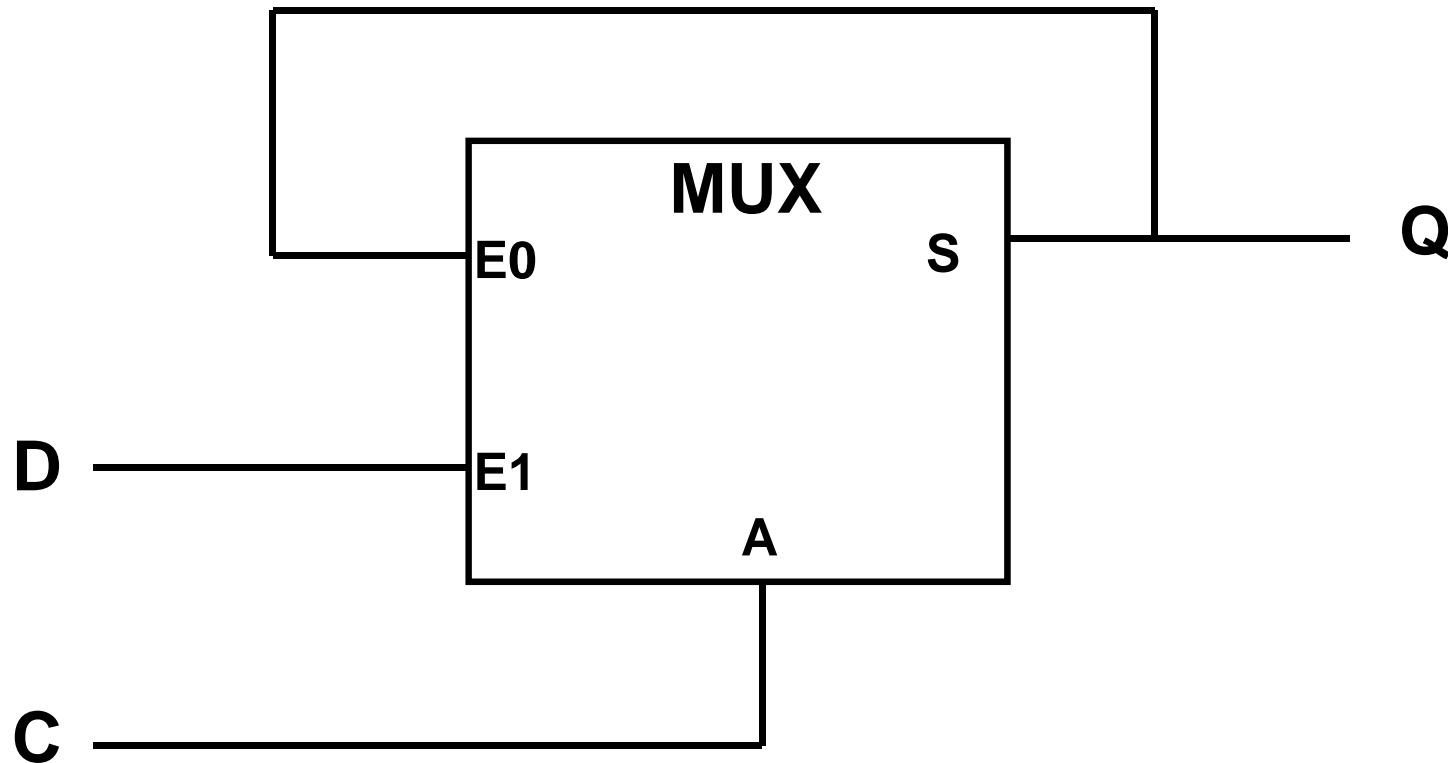
Entradas		Salidas	
C	D	S	R
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	0

Efecto en S-R	
$Q(t+1)$	$/Q(t+1)$
$Q(t)$	$/Q(t)$
$Q(t)$	$/Q(t)$
0	1
1	0

- De la tabla del circuito combinacional se obtiene  $S=CD$  y  $R=C\bar{D}$



- Una implementación alternativa:

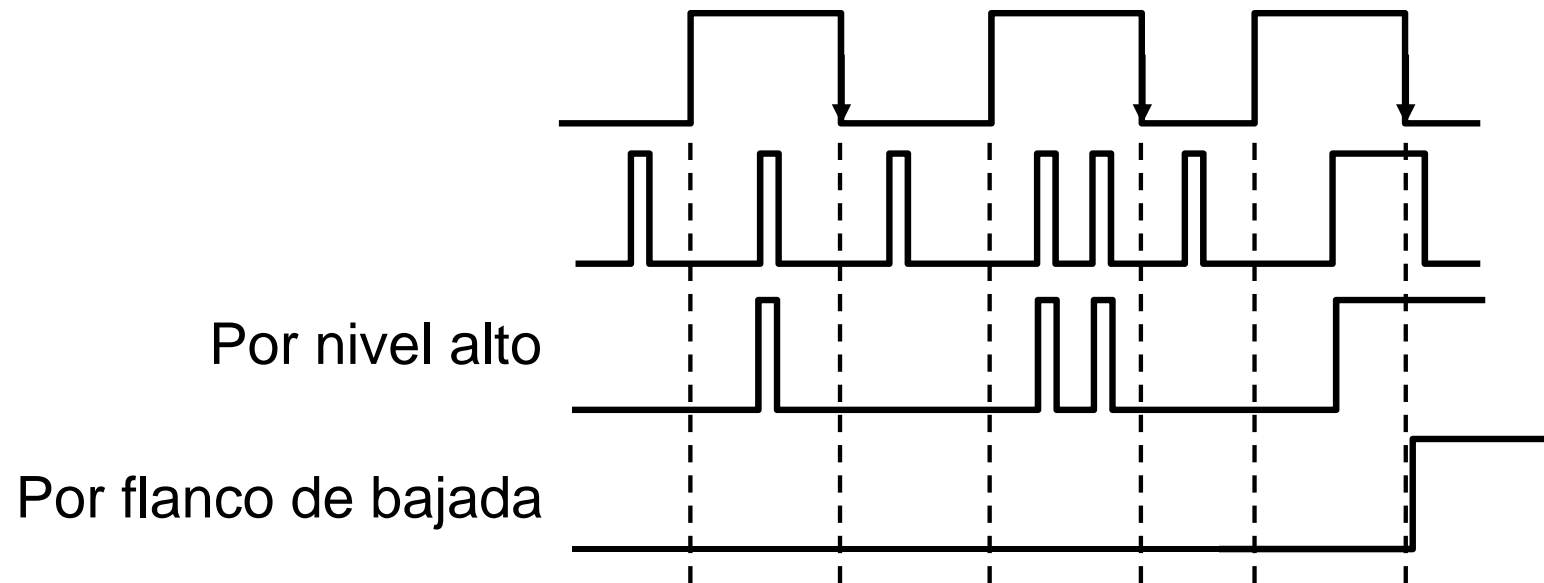


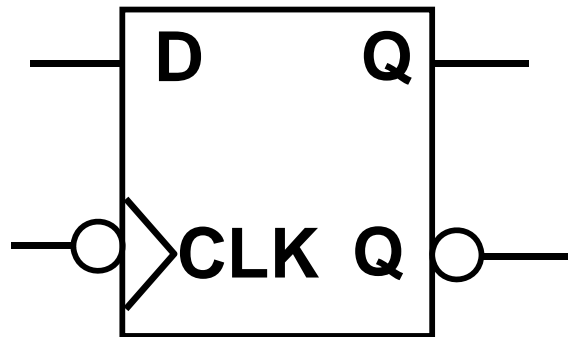
Si se activa el biestable por nivel  
y hay un pulso no deseado  
en la entrada de datos



el pulso no deseado  
se trasladará a la salida

Podemos hacer que la señal de reloj actúe por flanco



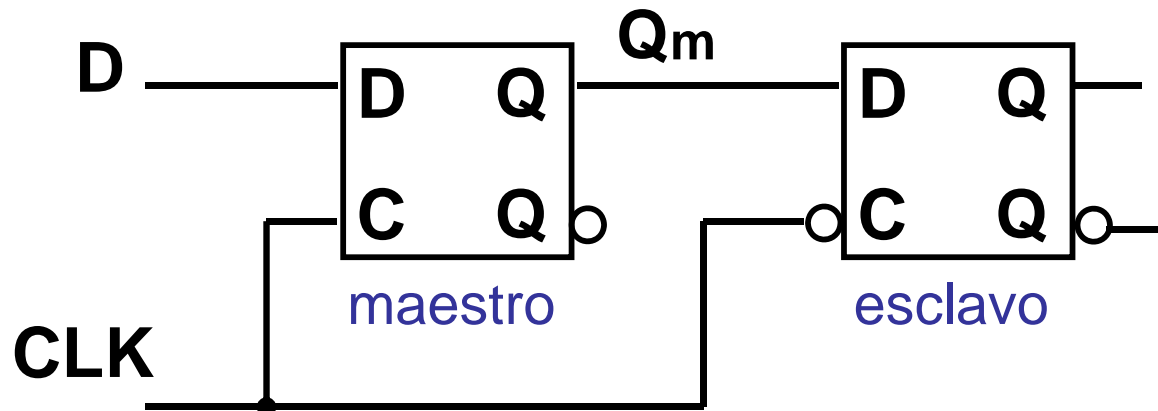


Símbolo lógico

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

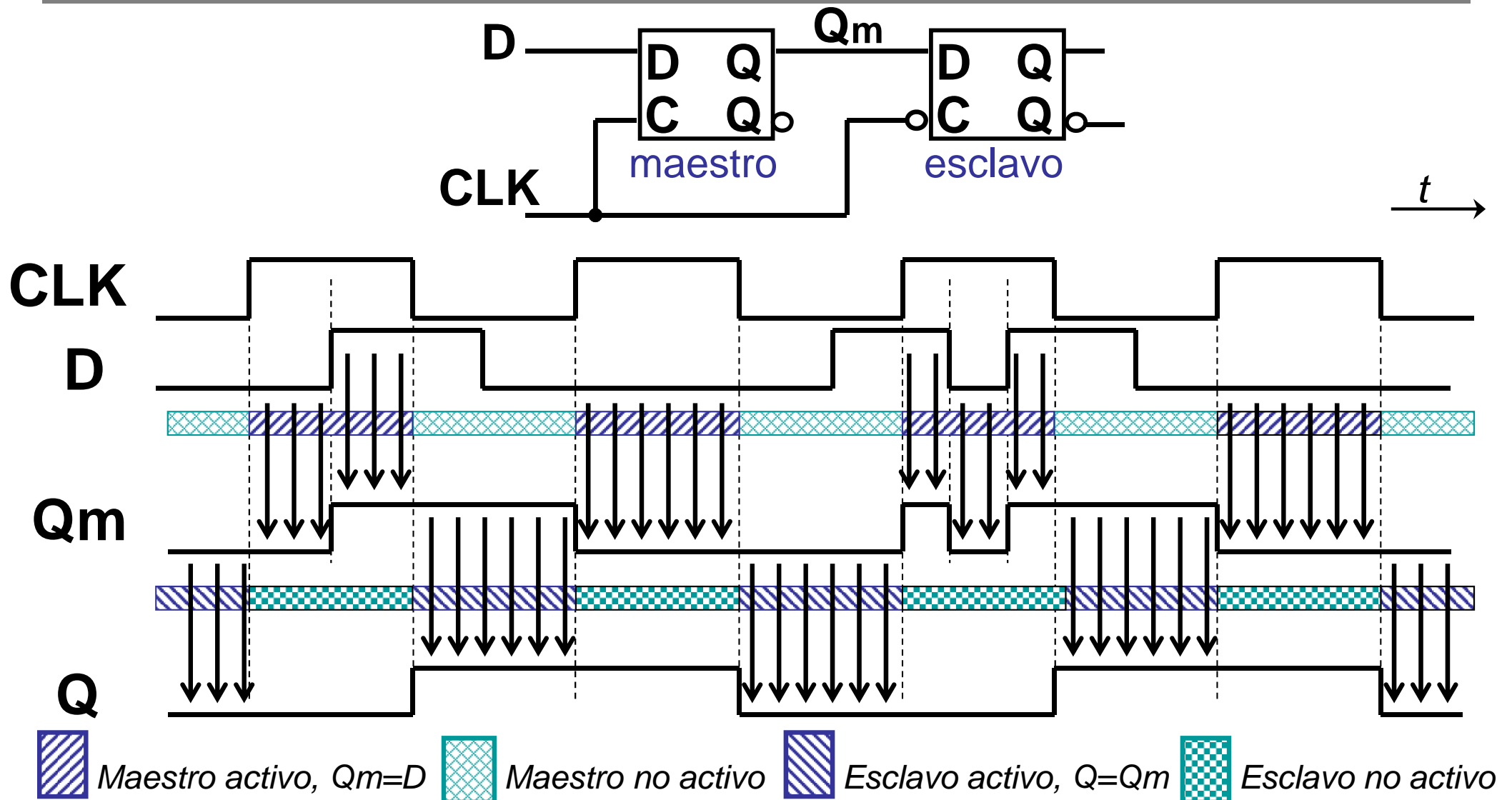
Tabla de funcionamiento

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

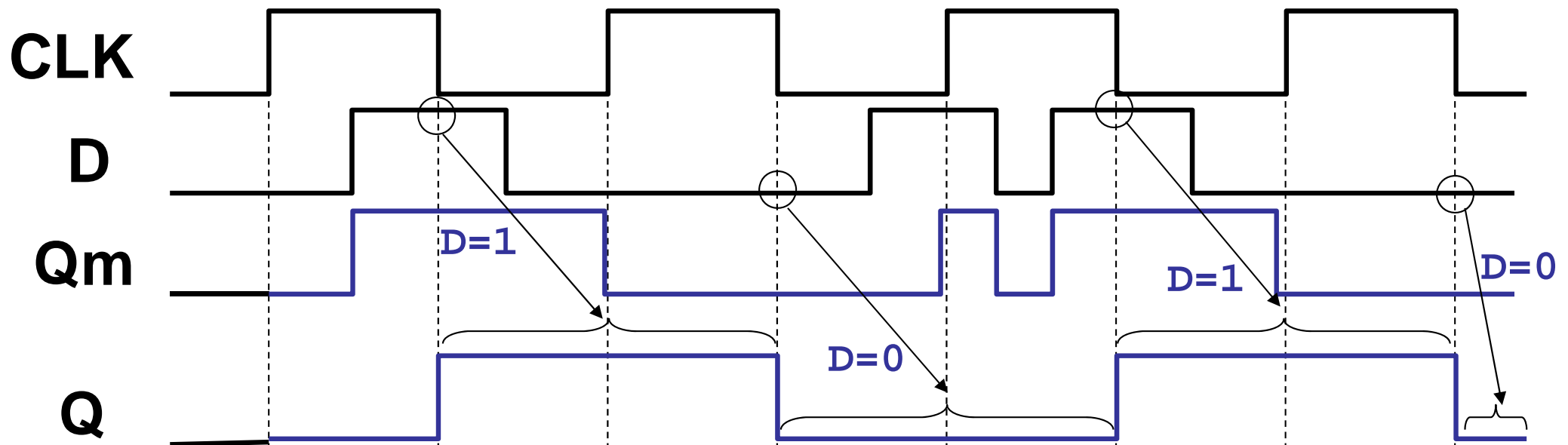


# Biestable D activo por flanco de bajada FCO

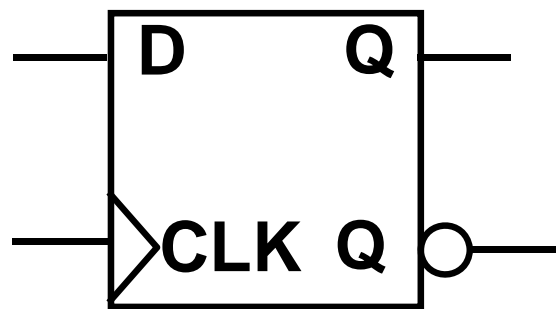
FCO



- Como se puede observar:
  - La línea interna  $Q_m$  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  $Q$





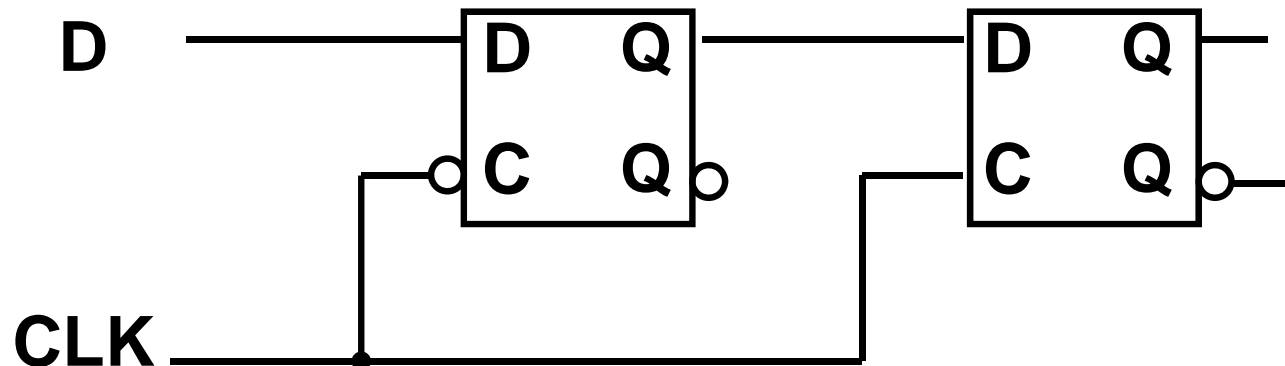


Símbolo lógico

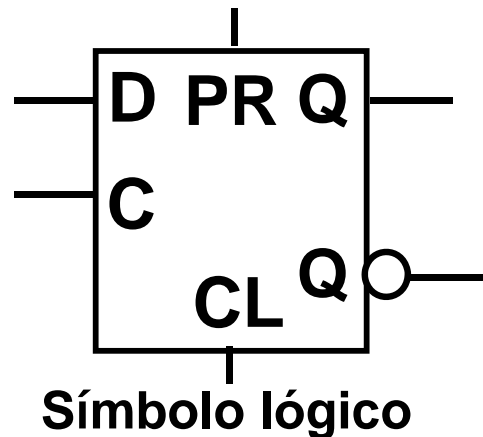
CLK	D	Q(t+1)	/Q(t+1)
0	X	Q(t)	/Q(t)
1	X	Q(t)	/Q(t)
↑	1	1	0
↑	0	0	1

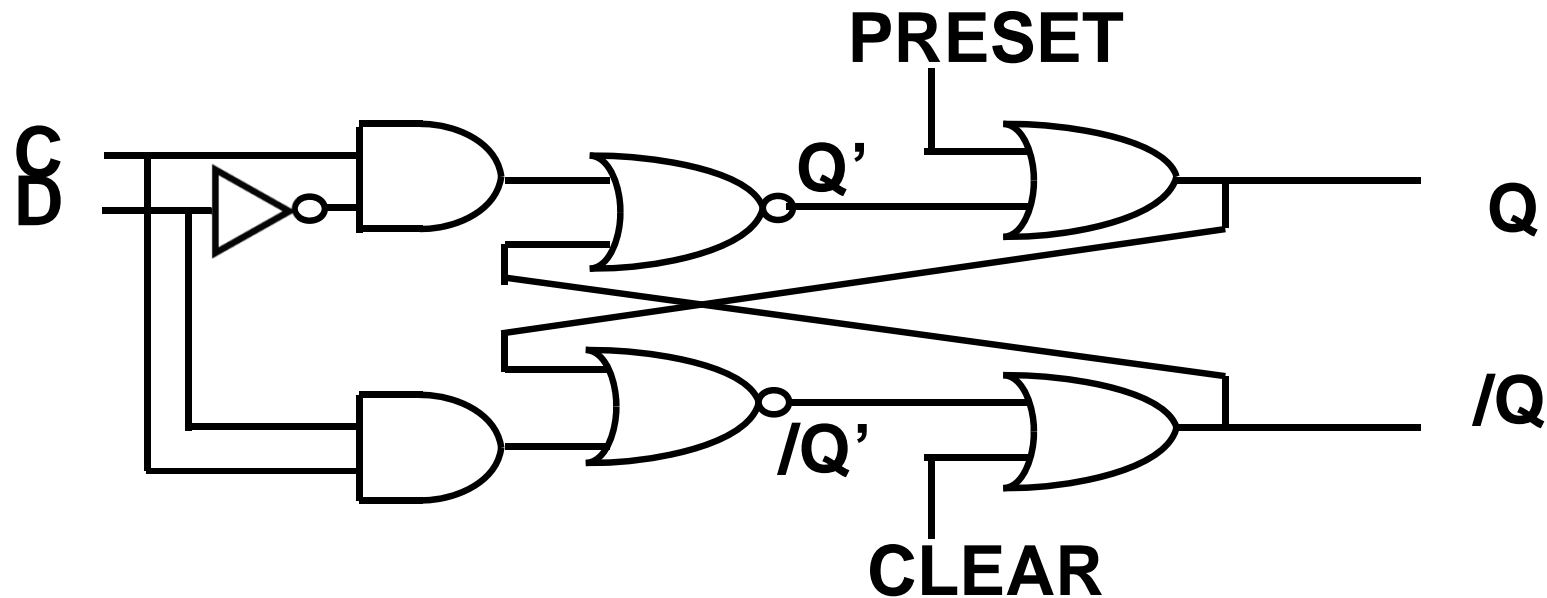
Tabla de funcionamiento

- Implementación (maestro activo a nivel bajo, esclavo a nivel alto)



- A un biestable D activo por nivel (alto o bajo) se le pueden añadir entradas asíncronas del tipo:
  - CLEAR (CL): puesta a cero asíncrona
  - PRESET (PR): puesta a uno asíncrona
- Las entradas asíncronas:
  - Tienen prioridad sobre las demás entradas
  - Permiten cambiar el estado del biestable en cualquier momento (independientemente del valor de las demás entradas)





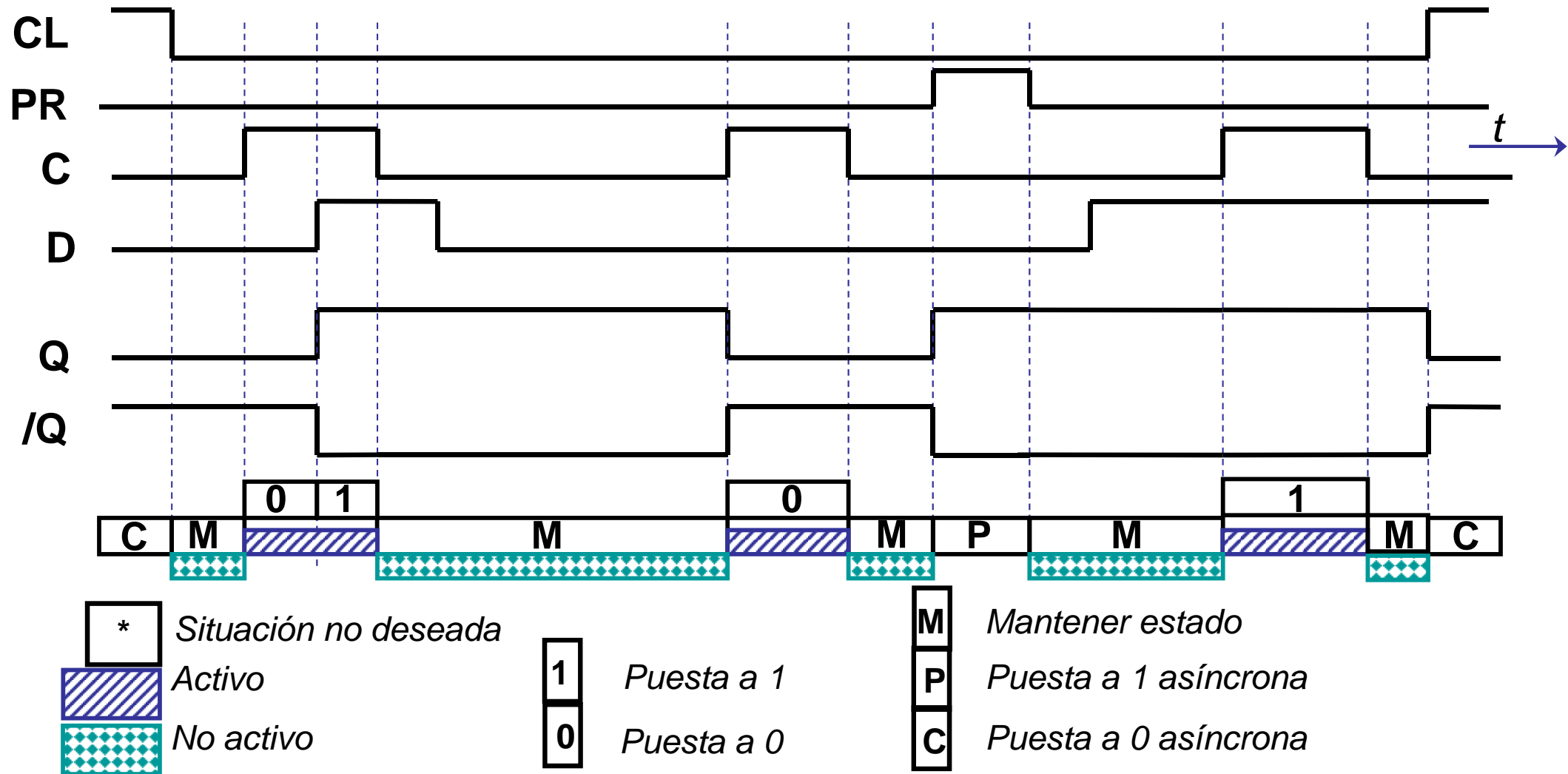
- Si  $CLEAR=PRESET=0 \Rightarrow Q = Q', /Q = /Q'$
- Si  $CLEAR=1, PRESET=0 \Rightarrow /Q = 1 \Rightarrow Q' = 0 \Rightarrow Q = 0$
- Si  $CLEAR=0, PRESET=1 \Rightarrow Q = 1 \Rightarrow /Q' = 0 \Rightarrow /Q = 0$
- Si  $CLEAR=1, PRESET=1 \Rightarrow Q=/Q=1^*$

- Tabla de funcionamiento

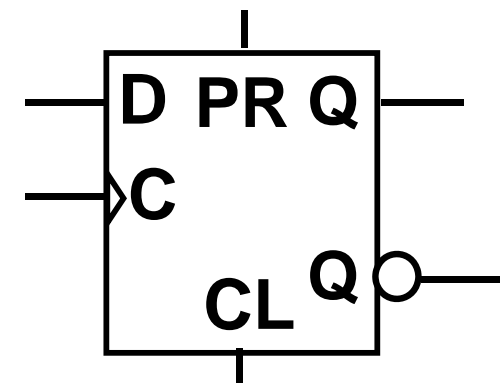
PR	CL	C	D	$Q(t+1)$	$/Q(t+1)$
0	1	X	X	0	1
1	0	X	X	1	0
1	1	X	X	1*	1*
0	0	1	1	1	0
0	0	1	0	0	1
0	0	0	X	$Q(t)$	$/Q(t)$

# Biestable D con entradas asíncronas

FCO

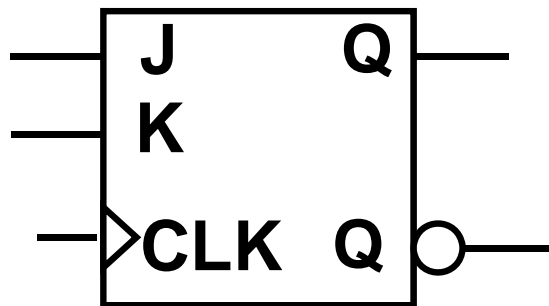


- A un biestable D activo por flanco (subida o bajada) también se le pueden añadir entradas asíncronas
  - Se puede implementar usando un diseño maestro-esclavo donde tanto maestro como esclavo disponen de entradas asíncronas
- Ejemplo: Biestable D activo por flanco de subida con entradas PRESET y CLEAR



Símbolo lógico

- El biestable S-R presenta problemas cuando se activan simultáneamente las dos entradas S y R
- El biestable J-K tiene un funcionamiento similar al S-R, pero evita el problema anterior invirtiendo el estado cuando J y K están activas simultáneamente

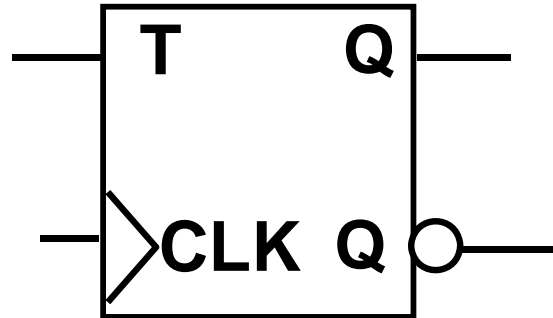


Símbolo lógico

CLK	J	K	Q(t+1)	/Q(t+1)
0	X	X	Q(t)	/Q(t)
1	X	X	Q(t)	/Q(t)
↑	0	0	Q(t)	/Q(t)
↑	0	1	0	1
↑	1	0	1	0
↑	1	1	/Q(t)	Q(t)

Tabla de funcionamiento

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



Símbolo lógico

Clk	T	Q(t)	Q(t+1)
↑	0	0	0
↑	0	1	1
↑	1	0	1
↑	1	1	0

Tabla de funcionamiento

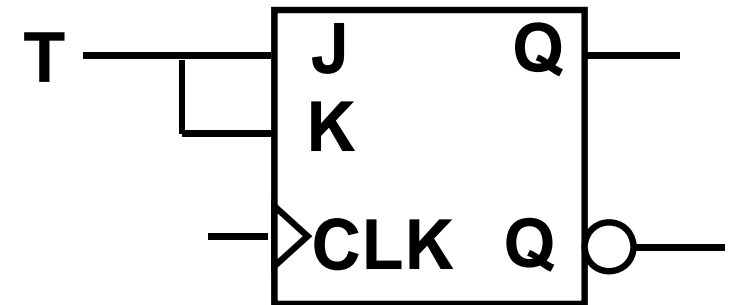


Diagrama lógico

- De manera análoga, se puede construir un biestable T activo por flanco de bajada usando un biestable J-K activo por flanco de bajada



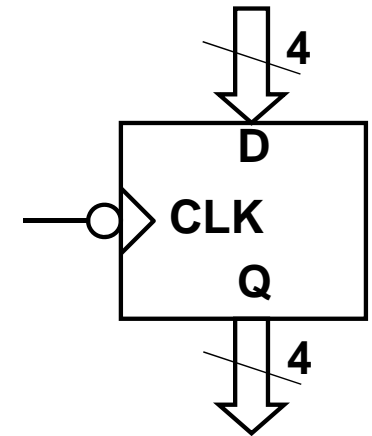
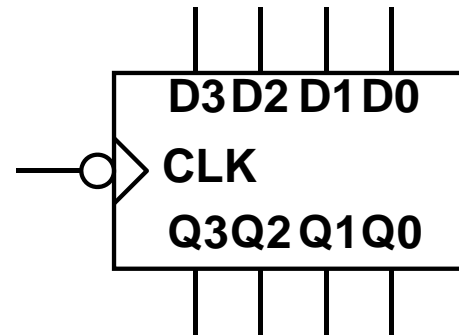
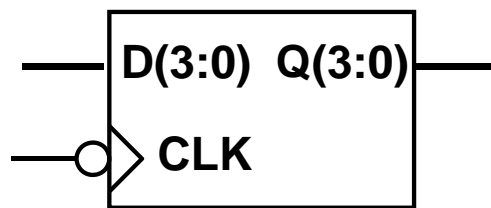
- Existen bloques secuenciales tan importantes y de uso tan común que se les ha dado nombre
  - Registro de almacenamiento
    - Almacén de un dato de  $N$  bits
  - Banco de Registros y Memoria
    - Agrupación de varios registros de almacenamiento
    - Misma funcionalidad básica. Difieren fundamentalmente en aspectos de capacidad, velocidad de funcionamiento y diseño

- Existen bloques secuenciales tan importantes y de uso tan común que se les ha dado nombre (cont.)
  - Registro de desplazamiento
    - Almacén de un dato de  $N$  bits en los se necesitan  $N$  ciclos de reloj para que la información entre (escritura) y/o salga (lectura)
  - Contador
    - Circuito que cambia de valor de forma autónoma en cada ciclo de reloj siguiendo una secuencia de valores predeterminada
      - Generalmente la cuenta es binaria (ascendente o descendente)

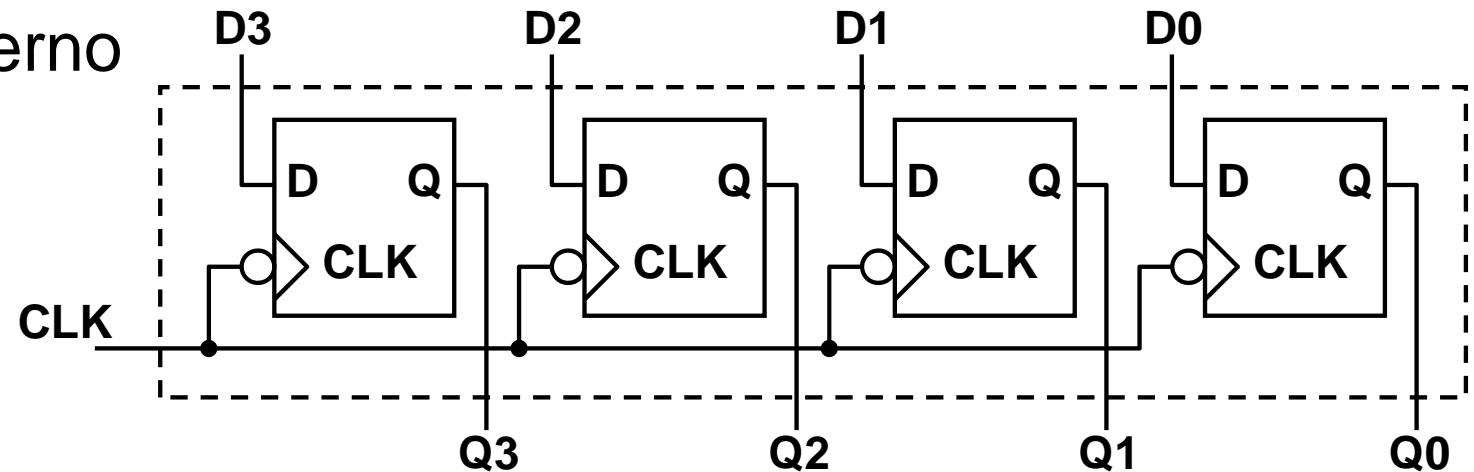
- Agrupación síncrona de biestables D activos por flanco
  - Tantos biestables D como bits queramos almacenar
    - Cada biestable D almacena un bit
  - Las entradas/salidas de datos del registro coinciden con las de los biestables D
  - Un único reloj (sistema síncrono) interconecta las entradas de reloj de todos los biestables
    - Todos los biestables deben ser activos en el mismo flanco de reloj

- Ejemplo: Registro de almacenamiento de 4 bits activo por flanco de bajada

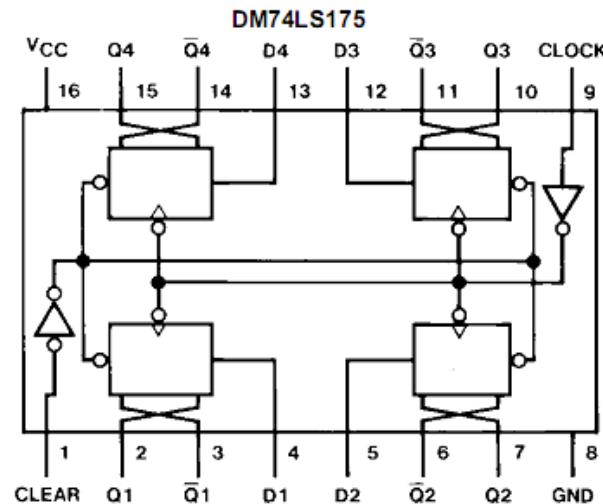
- Símbolos lógicos posibles



- Esquema interno



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



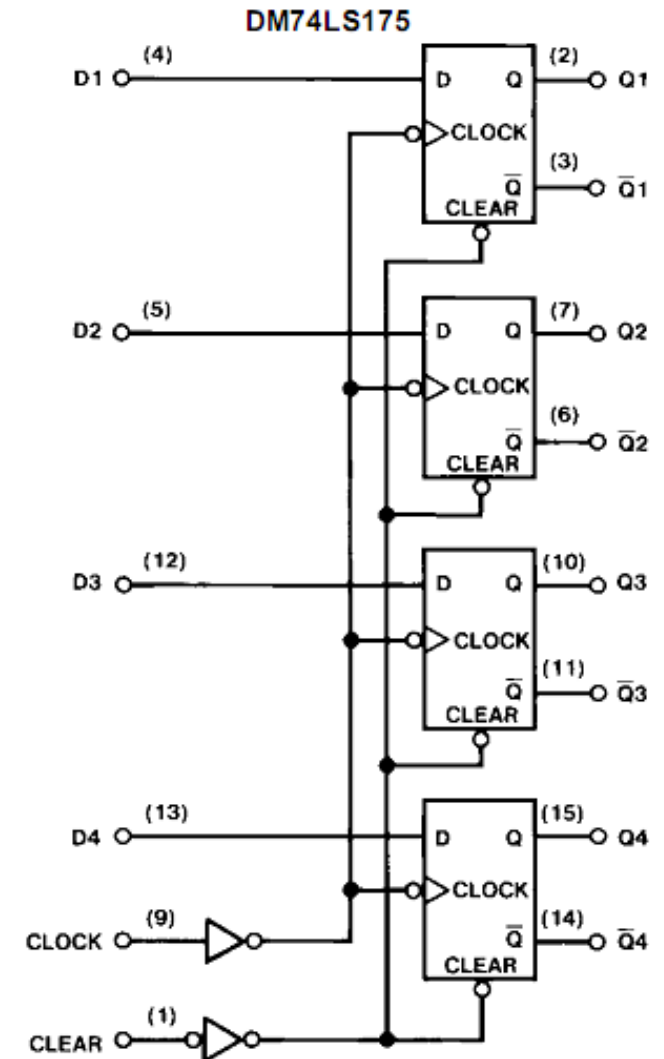
**Function Table**

(Each Flip-Flop)

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

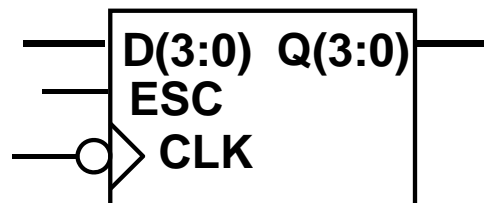
H = HIGH Level (steady state)

L = LOW Level (steady state)

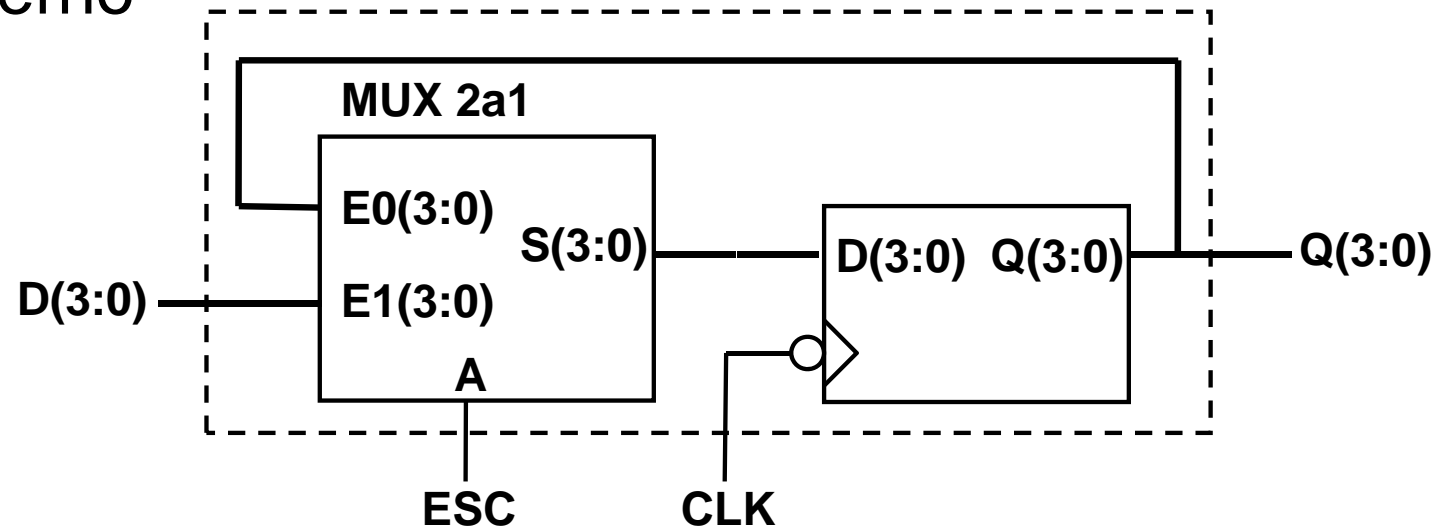


- Señal de escritura en paralelo (*parallel load*)
  - Permite habilitar o deshabilitar la escritura en el registro durante el flanco activo de reloj

- Símbolo lógico



- Esquema interno

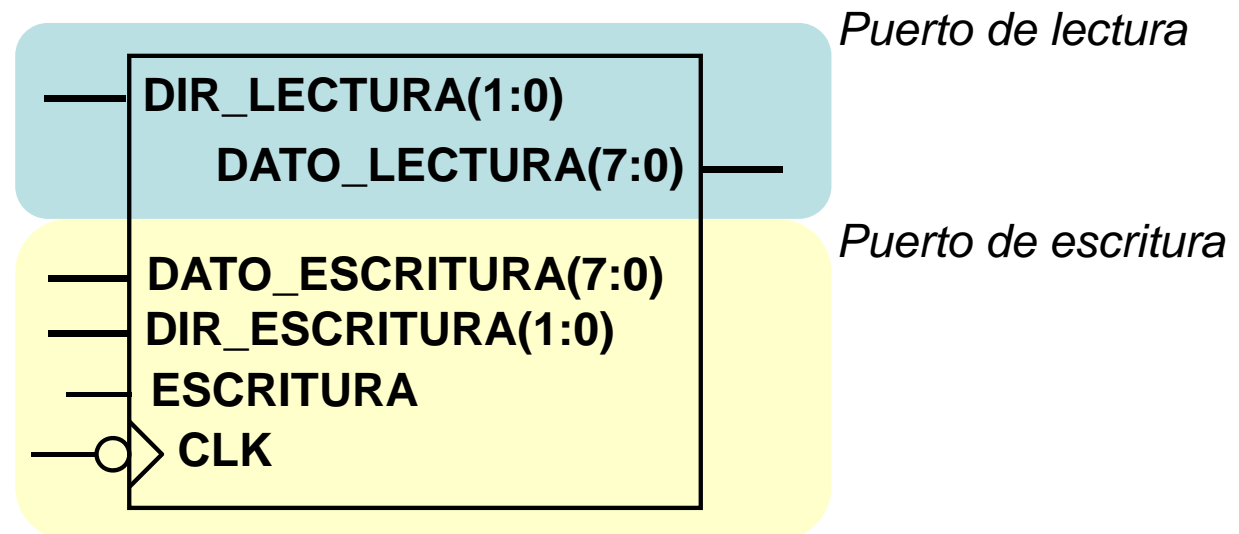


- Operación de escritura
  - Operación destructiva
    - El dato (previamente) almacenado desaparece y es sobre-escrito con el dato de la operación de escritura
  - Operativa
    - 1) Establecer el valor de las entradas D de los biestables
    - 2) Activar la señal de escritura
    - 3) La escritura se hará efectiva en el flanco activo de la señal de reloj
- Operación de lectura
  - Operación no destructiva
    - El dato almacenado permanece inalterado
  - Operativa
    - Examinar las salidas Q de los biestables

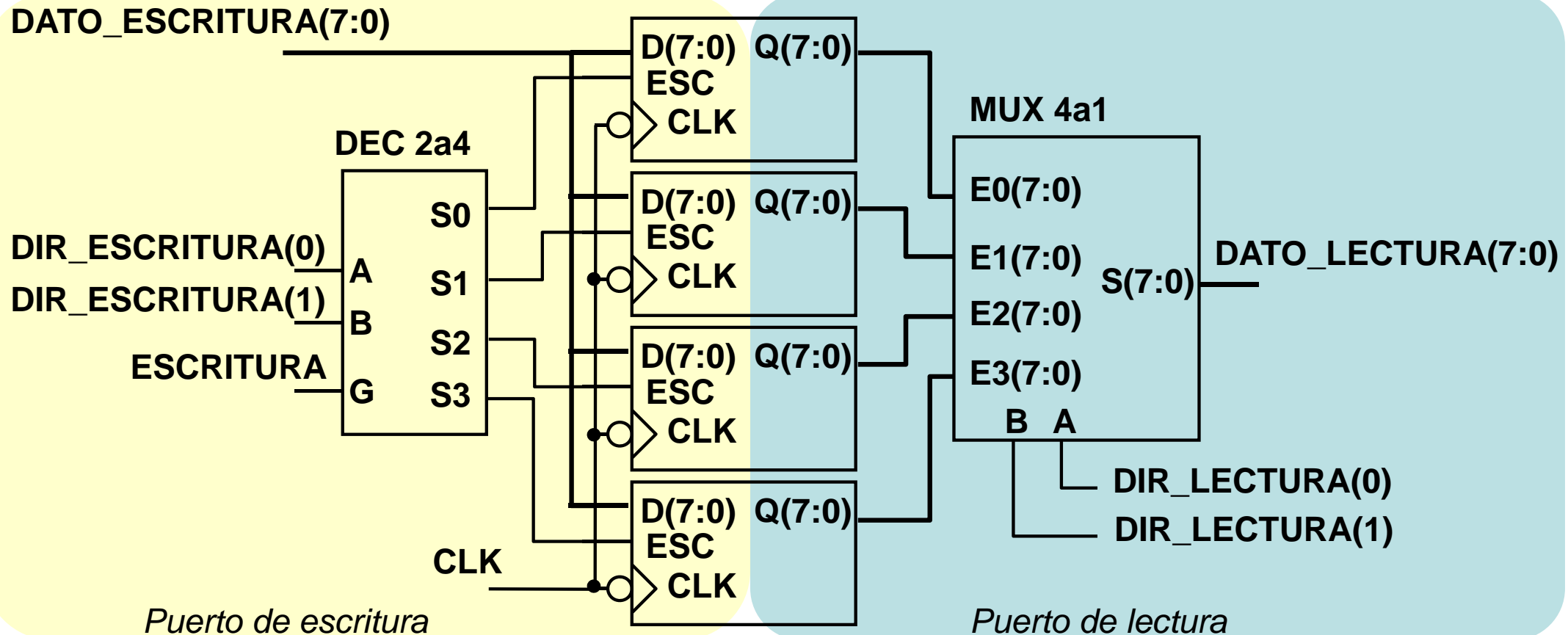
- Agrupación de registros
  - Permite la escritura de un dato sobre un registro
    - Hay que seleccionar qué registro queremos escribir
      - Esta información es la **dirección de escritura**
    - Internamente, un decodificador selecciona qué registro trabaja
  - Permite la lectura de, al menos, un dato almacenado
    - Hay que seleccionar qué registro queremos leer
      - Esta información es la **dirección de lectura**
    - Internamente, un multiplexor selecciona qué dato se obtiene
    - Para permitir la lectura simultánea de dos o más datos es necesario disponer de tantas entradas de dirección de lectura y multiplexores internos como datos se deseen leer



- Ejemplo: Banco de registros de 4 registros de 8 bits con un puerto (vía de acceso) de escritura y otro de lectura
  - 4 registros => 2 bits de dirección
  - Datos de entrada/salida de 8 bits
  - Símbolo lógico



- Ejemplo (cont):
  - Esquema interno



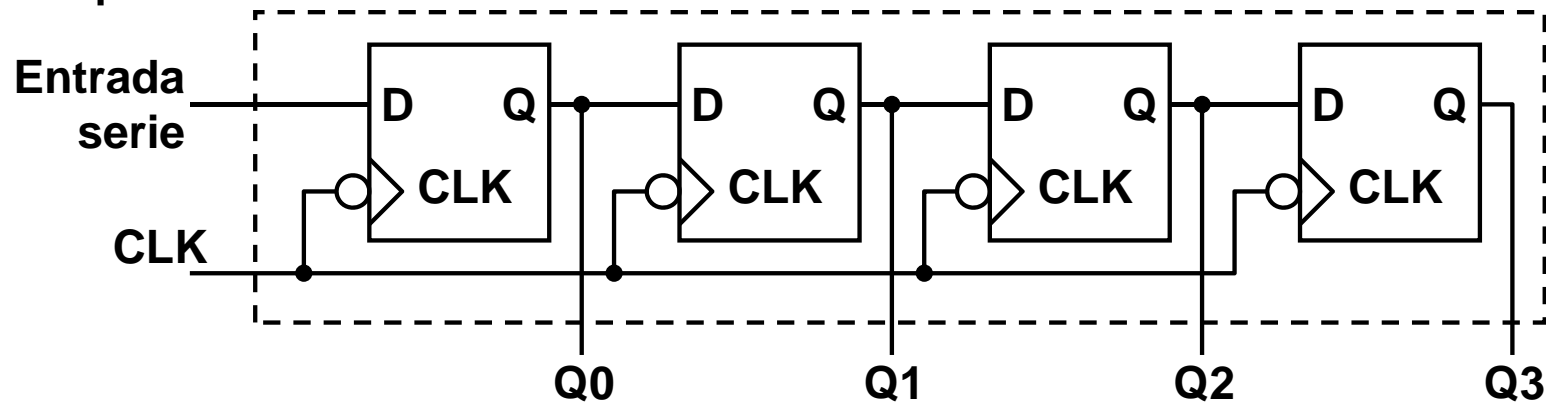
- Operación de escritura
  - Operativa
    - 1) Establecer el valor del dato de escritura
    - 2) Establecer el valor de la dirección de escritura
    - 3) Activar la señal de escritura (habilita el decodificador)
    - 4) La escritura se efectuará en el flanco activo de la señal de reloj
- Operación de lectura
  - Operativa
    - 1) Establecer el valor de la dirección de lectura
    - 2) Examinar el valor del dato de lectura

- Misma funcionalidad que un banco de registros, pero
  - Mucha más capacidad (Kbytes, Mbytes, Gbytes, ...)
  - Mucho más lenta
  - Tecnología diferente
- Una única operación (lectura/escritura) en un momento dado
  - Una única entrada de dirección y otra de dato
  - Líneas de órdenes de lectura y escritura
    - Para que la memoria sepa qué hacer en cada momento
      - Leer, escribir o nada (si no se activa ni lectura ni escritura)

- Agrupación síncrona de biestables D por flanco
  - Tantos biestables D como bits queramos almacenar
  - La información necesita varios ciclos de reloj para entrar (escritura) o salir (lectura)
    - Se conocen como entrada serie y salida **serie**, respectivamente
    - Cuando todos los bits entran o salen en el mismo ciclo de reloj se dice que el registro tiene entrada o salida **paralelo**
  - Estructura con entrada serie
    - El primer biestable conecta su entrada con la única entrada de datos
    - El resto, cada entrada con la salida del anterior
  - Estructura con salida serie
    - Sólo es accesible la salida del último biestable

- Entrada serie, salida paralelo

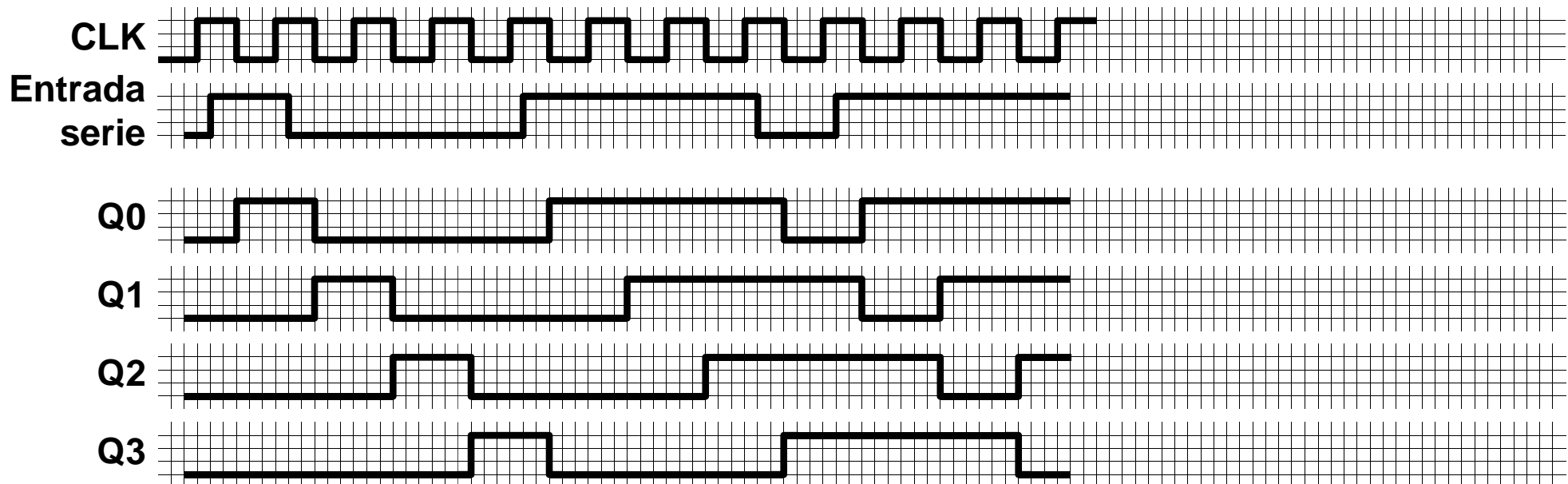
- Esquema interno



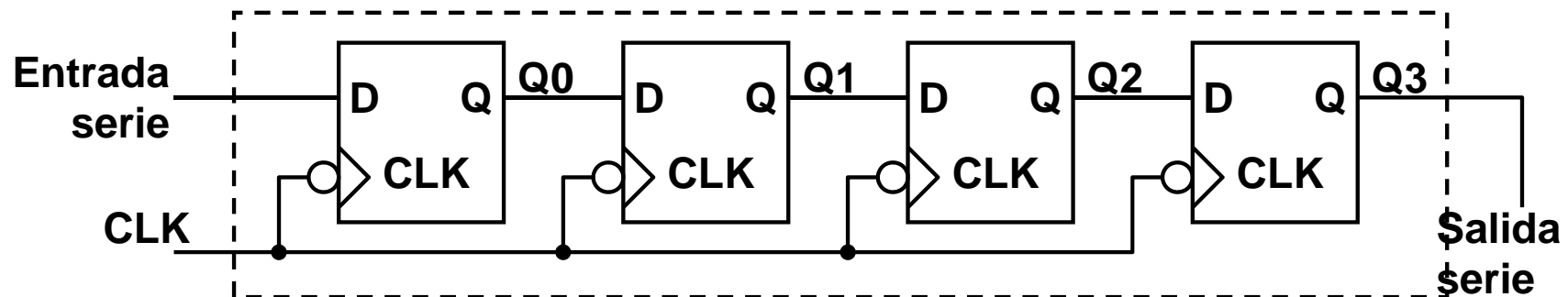
- Funcionamiento

- Un bit (el valor de la entrada serie) entra por la izquierda y desplaza la información almacenada una posición hacia la derecha
    - El bit almacenado en el extremo derecho se pierde

- Entrada serie, salida paralelo (cont.)
  - Ejemplo de funcionamiento
    - Estado inicial  $Q_3Q_2Q_1Q_0 = 0000$
    - La entrada serie toma los valores indicados en el cronograma (como ejemplo de secuencia de valores)



- Entrada serie, salida serie
  - Mismo circuito interno que con salida paralelo, excepto que la única salida disponible es la del último biestable



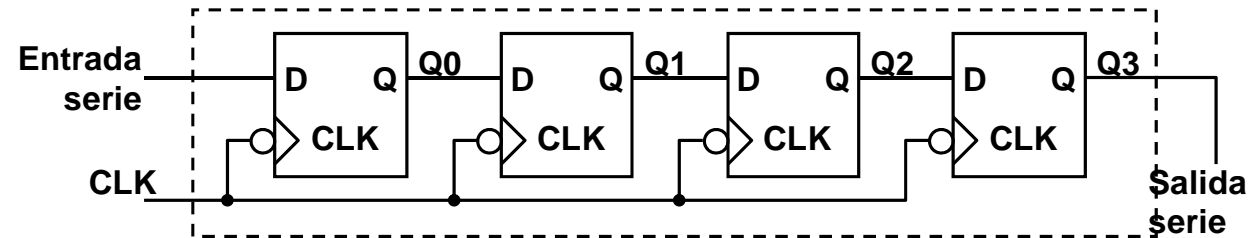
- Entrada paralelo, salida serie
  - Necesitamos una entrada adicional (que podemos llamar, por ejemplo, LOAD) para determinar si el sistema debe cargar el dato de entrada (LOAD=1) o desplazar (LOAD=0)



- Registros de desplazamiento (resumen)
  - Un registro de desplazamiento puede desplazar
    - A izquierdas o a derechas
    - No hay unanimidad en lo que significa “a izquierdas” o “a derechas”
      - Mejor indicar el sentido explícitamente
    - Incluso puede desplazar en ambos sentidos
      - No simultáneamente, a veces a izquierdas y a veces a derechas
      - Con una entrada de control adicional que indique el sentido
      - En este caso puede disponer de
        - » *Una única entrada de datos serie*
        - » *Una entrada de datos serie para cada uno de los sentidos de desplazamiento (entrada serie por la izquierda, entrada serie por la derecha)*

- Registros de desplazamiento (resumen, cont.)
  - La entrada de datos puede ser
    - Serie: Cuando la entrada de datos es de un bit por ciclo de reloj
    - Paralelo: Cuando todo el registro se escribe en un único ciclo de reloj
  - La salida puede ser
    - Serie: Cuando para observar el valor almacenado en el registro necesitamos tantos ciclos de reloj como biestables
    - Paralelo: Cuando podemos observar el valor de todos los biestables en un único ciclo de reloj

- Preguntas:

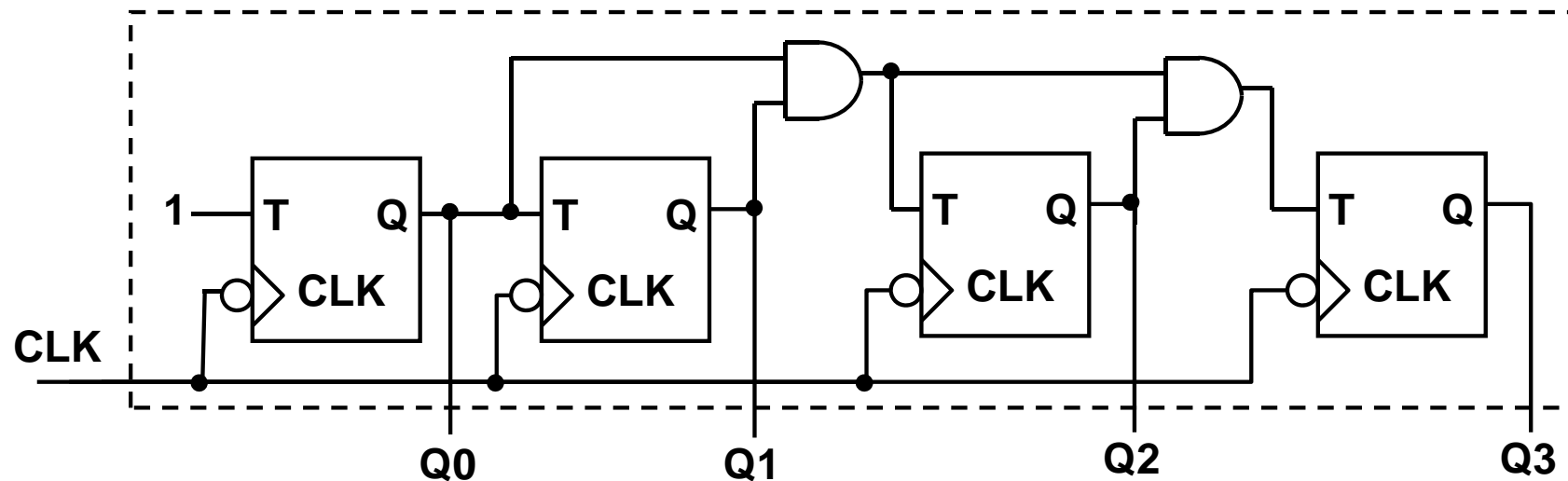


- ¿Cómo podemos construir un registro de desplazamiento con entrada y salida serie, con desplazamiento de Q3 a Q0?
- ¿Y que se pueda escoger el sentido del desplazamiento con una señal llamada “sentido” y con dos entradas serie?
- Ahora, ¿cómo le añadimos carga paralela?
- ¿Y hacerlo cíclico a veces sí y a veces no?

¿Multiplexores?

- Contador
  - Circuito que cambia de valor de forma autónoma en cada ciclo de reloj siguiendo una secuencia, generalmente una cuenta binaria (ascendente o descendente)
  - La cuenta puede ser ascendente o descendente
  - La cuenta también puede ser ascendente/descendente (reversible)
    - No simultáneamente, a veces ascendente y a veces descendente
    - Una entrada de control adicional indica el sentido

- Contadores síncronos
  - Suelen estar contruidos con biestables T o J-K (con  $J=K$ )
    - El circuito resultante utiliza menos puertas lógicas que si se usan biestables D
  - Ejemplo: Contador síncrono ascendente de 4 bits
    - Salida 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 0, ...



- Contadores: Clasificación por el tipo de cuenta
  - Binarios
    - Hacen todas las cuentas posibles entre 0 y  $2^{\text{variables de estado}} - 1$
  - De módulo N
    - Hacen N cuentas distintas, siendo  $N < 2^{\text{variables de estado}}$ 
      - Suelen incluir la cuenta 0...00, contando entonces de 0 a N-1
    - Ejemplo
      - Un contador de décadas es un contador de módulo 10 que cuenta de 0 a 9



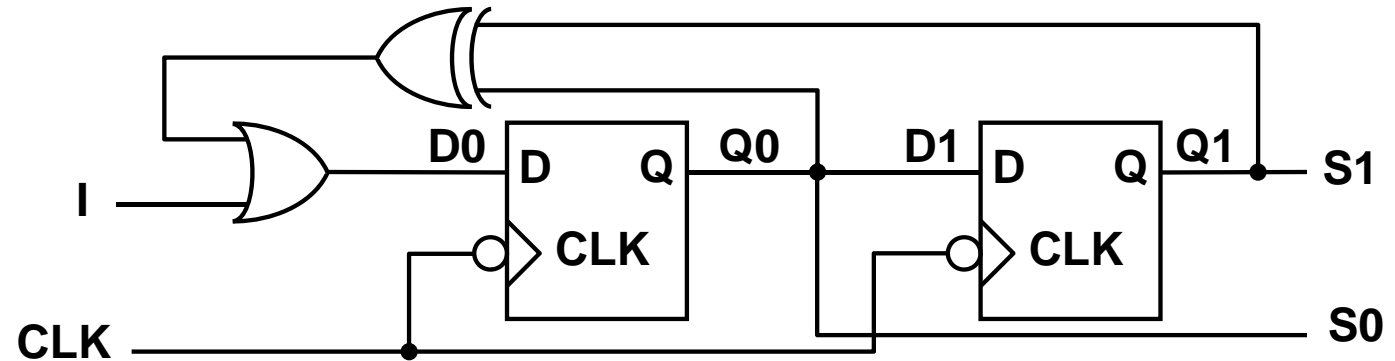
- Análisis de sistemas secuenciales
  - Consiste en obtener la salida del sistema a partir del circuito
  - Existen varios métodos
    - Cronograma:
      - Obtiene la salida del sistema para una secuencia de entradas particular



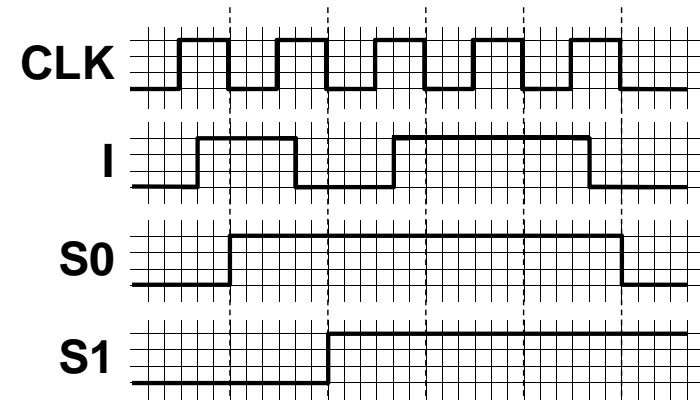
- Análisis por cronograma
  - Diagrama temporal
    - Incorpora todas las entradas y salidas del sistema
    - Puede ser de utilidad para simplificar el análisis añadir también señales internas del circuito
  - Necesitamos
    - El circuito
    - El estado inicial del sistema (si no se puede deducir por las entradas)
    - Una secuencia de entradas
  - Obtenemos
    - La secuencia de salidas del sistema para la secuencia de entradas concreta que hemos empleado en el análisis

- Análisis por cronograma
  - Para realizar el análisis debemos recurrir a las tablas de verdad
    - De cada biestable del circuito
    - Para cada una de las combinaciones de entrada que tenga a lo largo del tiempo
  - Debemos aplicar las combinaciones de entrada una por una en el tiempo
    - Porque el nuevo valor (de un biestable) puede influir en el comportamiento futuro (del mismo biestable o de otro)

- Ejemplo
  - Analizar



- Valor inicial  $Q1Q0 = 00_2$
- Secuencia de entrada  $I = 1 - 0 - 1 - 1 - 0$
- Cronograma



(S1S0 en decimal) Salida 0 1 3 3 3 2



UNIVERSIDAD  
POLITECNICA  
DE VALENCIA



etsinf



---

# Fundamentos de computadores

---

## TEMA 4. CIRCUITOS SECUENCIALES

---