Un experto que resuelve problemas debe estar dotado con dos cualidades incompatibles, una inquieta imaginación y una paciencia tenaz.

Howard W. Eves

Tema 5: Sistemas secuenciales.

5.1: Introducción.

5.2: Sistemas secuenciales asíncronos.

- 5.2.1: Biestables asíncronos.

5.3: Sistemas secuenciales síncronos.

- 5.3.1: Biestables síncronos.
- 5.3.2: Análisis y síntesis de sistemas secuenciales síncronos.
  - Modelos de *Mealy* y *Moore*.
- 5.3.3: Bloques funcionales síncronos
  - 5.3.3.1: Contadores.
  - 5.3.3.2: Registros.

▪ Los Sistemas Digitales se pueden clasificar en dos grupos, de acuerdo con su comportamiento:

$$\textit{Sistemas Digitales}\begin{cases} \cdot\ \textit{Sistemas combinacionales.} \\ \cdot\ \textit{Sistemas secuenciales}\begin{cases} \textit{asíncronos} \\ \textit{síncronos} \end{cases} \end{cases}$$

DEF. 1: Los sistemas combinacionales se caracterizan porque el valor lógico de sus salidas, en cualquier instante de tiempo *t* que se considere, *sólo* depende del valor lógico de sus entradas en dicho instante de tiempo *t*.

DEF. 2: Los sistemas secuenciales (*sequential machine or state machine*) se caracterizan porque los estados lógicos de sus salidas, en cualquier instante de tiempo *t* que se considere, en general, dependen de:

i) Las condiciones iniciales en las que ha comenzado a funcionar el sistema.

ii) La secuencia de todas las entradas que se han aplicado al sistema (circuito) hasta dicho instante de tiempo *t*.

iii) El estado lógico de las entradas en dicho instante *t*.

Ejemplos:

- Circuito de control de un ascensor.
- Circuito de control del cambio de canal de una T.V.

Nota: los sistemas secuenciales necesitan guardar información, mientras que los sistemas combinacionales no lo hacen.

DEF. 3: Los sistemas secuenciales se caracterizan porque el valor lógico de sus salidas, en cualquier instante de tiempo *t* que se considere, dependen de:

_ El *estado interno* del sistema [equivale a *i)* y *ii)* en la Def.1].

_ El *estado lógico* de sus entradas [equivale a *iii)* en la Def.1].

DEF. 4: Se define el *estado interno* de un sistema secuencial como un conjunto de variables lógicas, denominadas *variables de estado*, cuyos valores en cualquier instante de tiempo *t* que se considere contienen la información necesaria acerca del pasado del sistema para explicar su comportamiento futuro.

Las *variables de estado* se caracterizan por lo siguiente:

- ✓ Son variables lógicas que crea el propio sistema secuencial.
- ✓ No representan necesariamente magnitudes físicas conocidas.
- ✓ En la mayoría de los sistemas secuenciales se pueden definir diferentes grupos de variables de estado para representar el *estado* del sistema.
- ✓ Un sistema secuencial con n variables de estado tiene como mucho 2^n estados distintos.

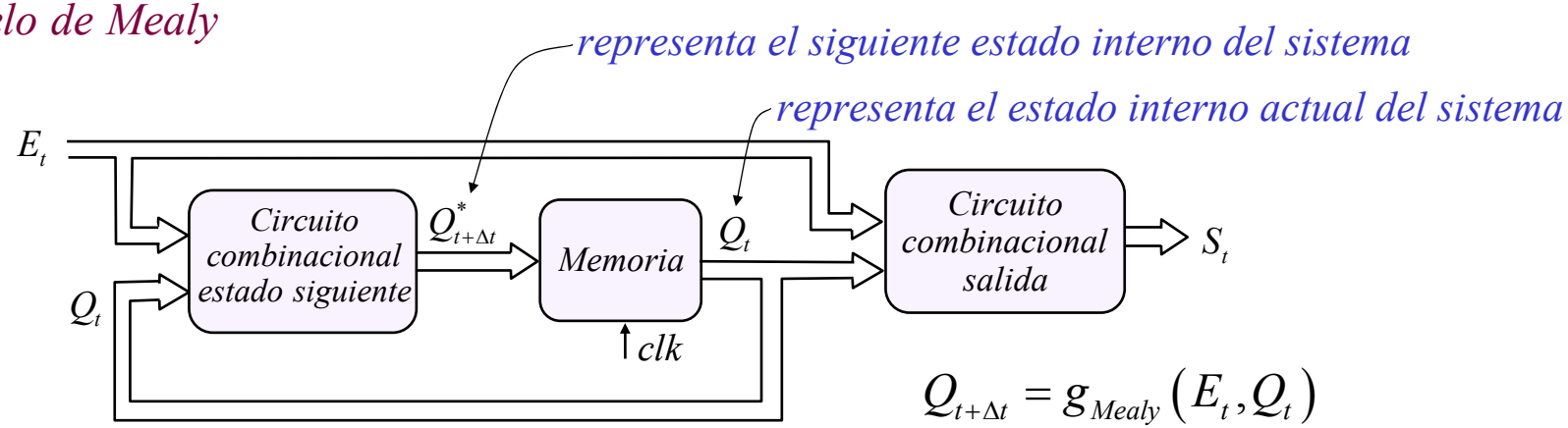
número de estados finito $\Rightarrow$ *sistemas* (*o máquinas*) *de estados finitos*.

Los sistemas secuenciales se diseñan (y se analizan) a partir de los conceptos anteriores siguiendo dos estructuras o modelos distintos, denominados modelos de *Mealy* y de *Moore*.

Nota 1: se puede demostrar que para todo sistema diseñado siguiendo el modelo de *Mealy* existe un modelo de *Moore* equivalente y viceversa. Ahora bien, aunque un diseño se pueda sustituir por otro, su funcionamiento no es exactamente el mismo.

Nota 2: Aunque en general el modelo de *Mealy* proporciona circuitos más sencillos, hay sistemas cuyo comportamiento se describe mucho mejor mediante un modelo de *Moore*.

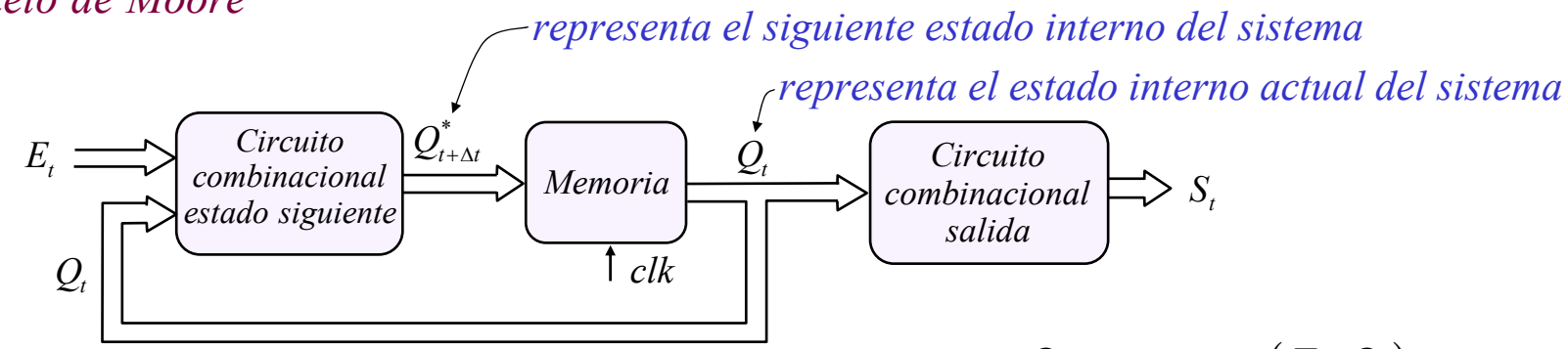
Modelo de Mealy



$$Q_{t+\Delta t} = g_{Mealy}\left(E_t, Q_t\right)$$

$$S_t = f_{Mealy}\left(E_t, Q_t\right)$$

Modelo de Moore



$$Q_{t+\Delta t} = g_{Moore}\left(E_t, Q_t\right)$$

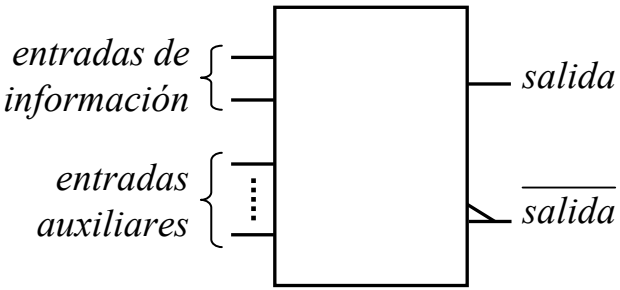
$$S_t = f_{Moore}\left(Q_t\right)$$

E_t : *entradas del sistema*

S_t : *salidas del sistema*

Q_t : *estado actual*

clk : señal de sincronismo



Dispositivos de memoria: Biestables.

Características:

- Un *biestable* es un circuito digital muy sencillo que guarda (memoriza) 1 bit de información.

- Básicamente, un *biestable* posee: 1 salida (Q), 1 ó 2 entradas de información (según el tipo de *biestable* que se considere) y, en algunos casos, entradas auxiliares.

- Las entradas de información sirven para indicarle al *biestable* el valor lógico que debe memorizar (0 ó 1).

- La salida proporciona en todo momento el valor guardado por el *biestable*.

- Puede tener entradas auxiliares que permiten controlar su funcionamiento

- Un biestable constituye por si solo un sistema secuencial ⇒ los *biestables* se pueden clasificar en *asíncronos* y *síncronos*.

Una posible clasificación de los biestables es la siguiente:

- *Biestables*
  - *Asíncronos* (*Latches*)
    - *Sin entrada de habilitación*: *SR* (*D*, *JK*, *T*)
    - *Con entrada de habilitación*: *D* (*SR*, *JK*, *T*)
  - *Síncronos* (*Flip-flops*)
    - *Disparados por flanco* (*edge triggered*): *D*, *JK*, *T*, (*SR*, *PQ*)

- A nivel funcional, el comportamiento de un biestable se puede definir mediante:

_ Una *ecuación característica*: proporciona el nuevo valor a guardar (Q_{t+1}) en función del valor de las entradas y del valor guardado actualmente.

$$Q_{t+1} \equiv Q_{t+\Delta t} \equiv Q_t^* = f(E_t, Q_t)$$

_ Una *tabla de transición de estados*.

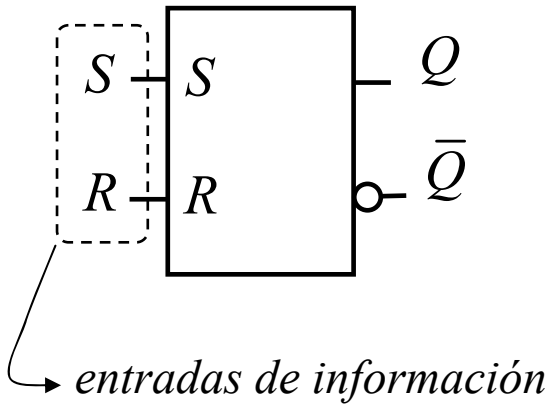
_ Un *diagrama de flujo* o *diagrama de estados*.

_ Un *cronograma*.

Biestables asíncronos: LATCHES

Se caracterizan porque obedecen en todo momento a sus entradas de información

Latch S-R:



- ✓ Es el más sencillo de <u>todos</u> los biestables
- ✓ Se utiliza como punto de partida para diseñar cualquier otro tipo de biestable
- ✓ Tiene dos entradas (de información):

  S ≡ set → *sirve para indicarle que guarde un* 1

  R ≡ reset → *sirve para indicarle que guarde un* 0

Nomenclatura para biestables

Q_t : *representa el valor actual o presente de la salida Q de un biestable. Dicho de otra forma, representa el valor de la salida de un biestable* <u>*antes*</u> *de que ésta cambie de valor debido a un cambio de valor en una de sus entradas.*

$Q_{t+\Delta t}$: *representa el siguiente valor que tomará la salida Q de un biestable. Es decir, es el valor que pasará a tener la salida de un biestable como consecuencia de un cambio de valor de una entrada (≡ es el siguiente valor que guardará el biestable)*

Nomenclatura para variables de estado

Q_t : *representa el valor actual o presente de una variable de estado. Dicho de otra forma, representa el valor de una variable de estado antes de que el sistema cambie de estado.*

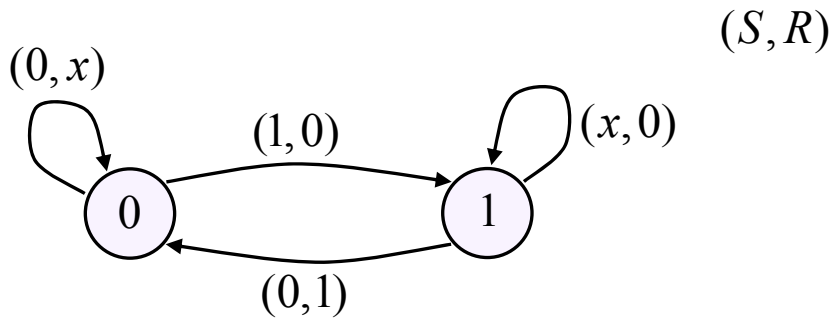
$Q_{t+\Delta t}$: *representa el siguiente valor que tomará una variable de estado. Es decir, es el valor que tomará la variable de estado al cambiar el sistema a un nuevo estado.*

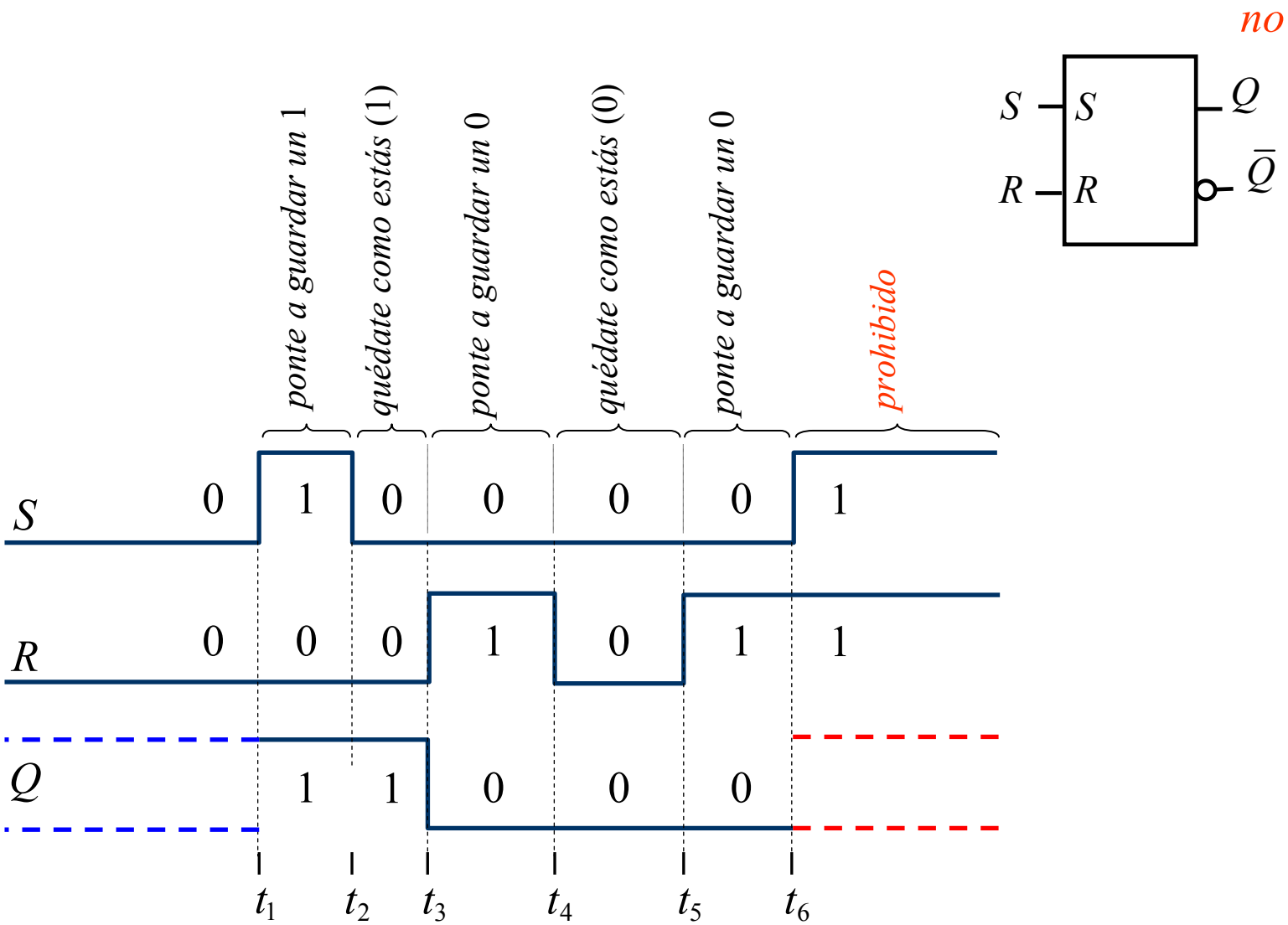
Nomenclatura: $Q_t \equiv Q \qquad Q_{t+\Delta t} \equiv Q_{t+1} \equiv Q^*$

- Definición de su comportamiento:

valor de una misma salida en dos instantes de tiempo distintos

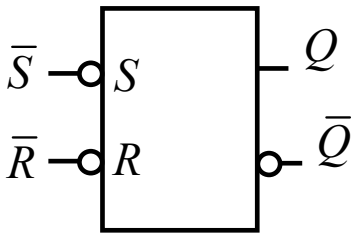
S_t	R_t	Q_t	$Q_{t+\Delta t}$	
0	0	0	0	*no hagas nada, quédate como estás* $\left(Q_{t+\Delta t} = Q_t\right)$.
0	0	1	1	
0	1	0	0	*ponte a memorizar un cero* $\left(Q_{t+\Delta t} = 0\right)$.
0	1	1	0	
1	0	0	1	*ponte a memorizar un uno* $\left(Q_{t+\Delta t} = 1\right)$.
1	0	1	1	
1	1	0	x	*orden prohibida*
1	1	1	x	

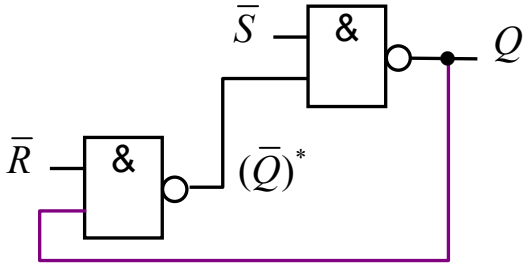


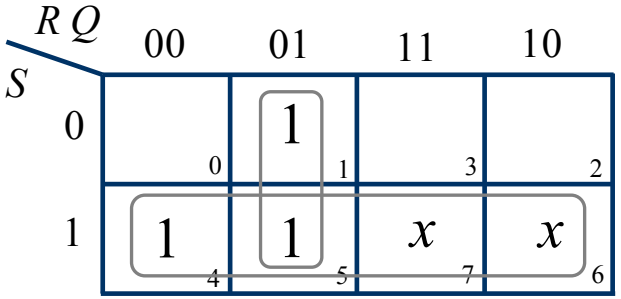

no
S
R
S
R
Q
$\bar{Q}$
ponte a guardar un 1
quédate como estás (1)
ponte a guardar un 0
quédate como estás (0)
ponte a guardar un 0
prohibido
S
0 1 0 0 0 0 1
R
0 0 0 1 0 1 1
Q
1 1 0 0 0
t_1 t_2 t_3 t_4 t_5 t_6

▪ *Esquema*: *no*

(*grabado prioritario*)



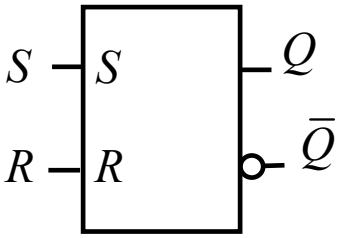


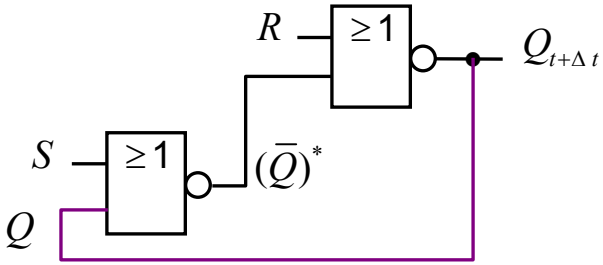


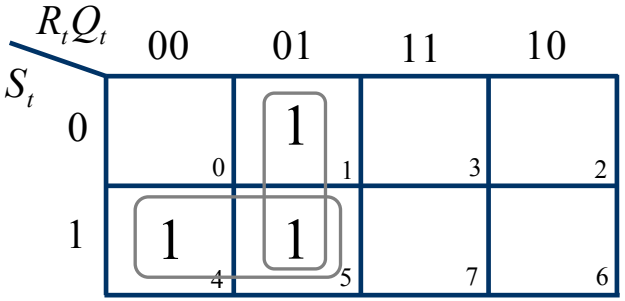
$$Q_{t+\Delta t}\left(S_t, R_t, Q_t\right) = \bar{R}_t Q_t + S_t$$

$$Q^*\left(S, R, Q\right) = \bar{R} \cdot Q + S$$

(*borrado prioritario*)





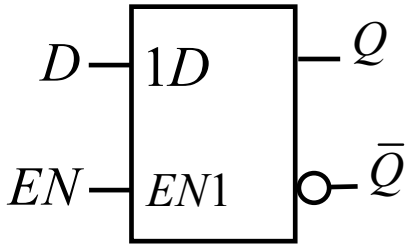


$$Q_{t+\Delta t}\left(S_t, R_t, Q_t\right) = \bar{R}_t Q_t + S_t \bar{R}_t = \bar{R}\left(S + Q\right)$$

$$Q^*\left(S, R, Q\right) = \bar{R} \cdot Q + S \cdot \bar{R} = \bar{R}\left(S + Q\right)$$

no

Latch *D* con entrada de habilitación:

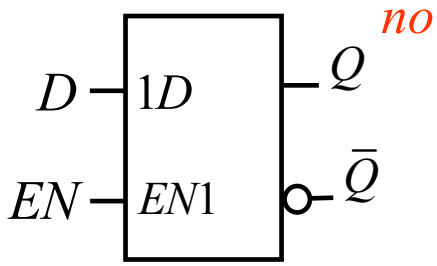


Características:

✓ Tiene una entrada de información (*D*)

✓ Tiene una entrada auxiliar (*EN*) con la que se controla cuándo el *latch* obedece a la entrada de información y cuándo no.

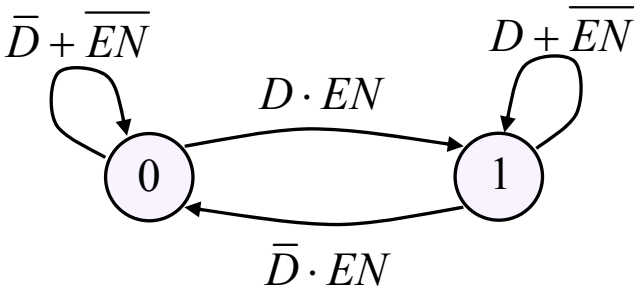
- Definición de su comportamiento:



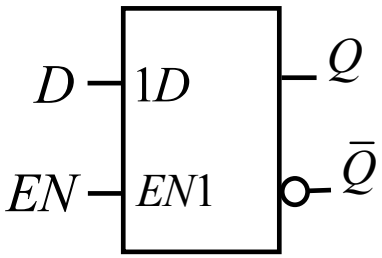
EN_t	D_t	Q_t	$Q_{t+\Delta t}$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

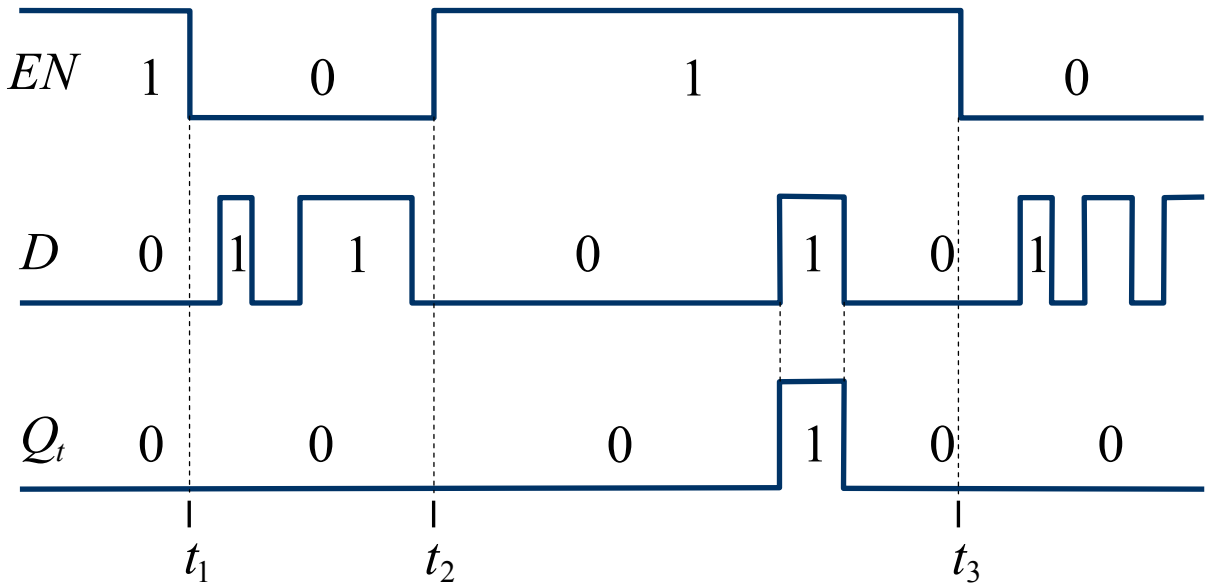
no obedece a la entrada de información $\left(Q_{t+\Delta t} = Q_t\right)$.

guarda el valor que le indica la entrada de información $\left(Q_{t+\Delta t} = D_t\right)$.



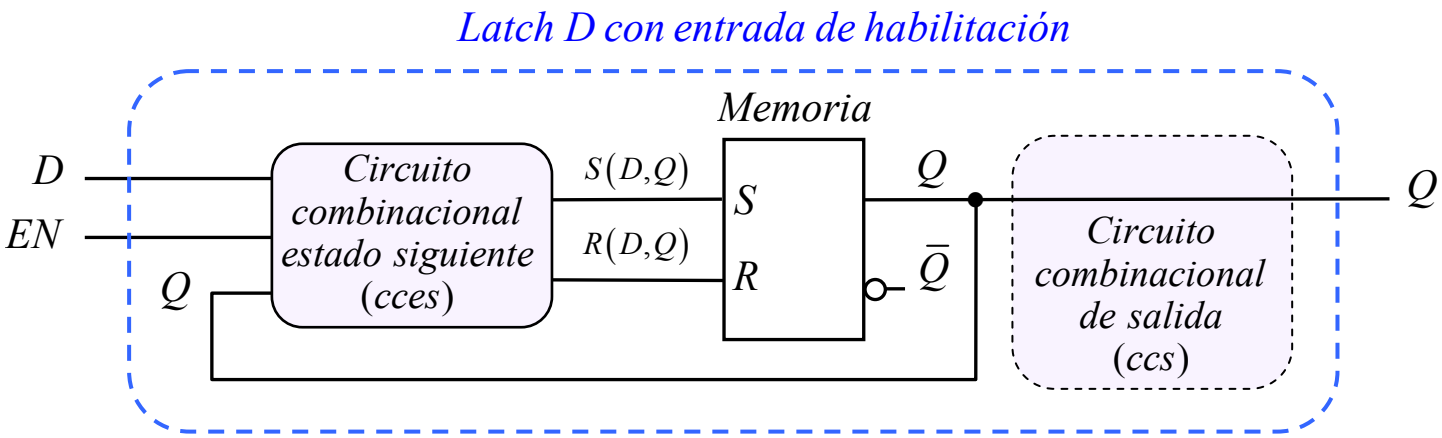
no



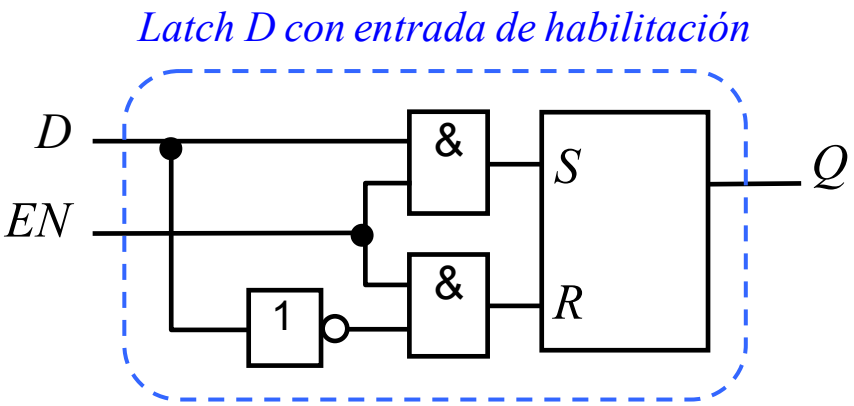


no

- Diagrama de bloques:



- Esquema:



Biestables síncronos: Flip-Flops

Se caracterizan por lo siguiente:

✓ Básicamente, tienen dos tipos de entradas:

_ entradas de información (1 ó 2).

_ entrada de sincronismo (1).

✓ La entrada de sincronismo es la encargada de habilitar las entradas de información

✓ Las entradas de información están habilitadas durante intervalos de tiempo muy pequeños (en torno a los flancos de sincronismo)

✓Algunos *flip-flops* tienen entradas auxiliares (asíncronas). La acción de una entrada *asíncrona* siempre prevalece sobre la acción de una entrada *síncrona*.

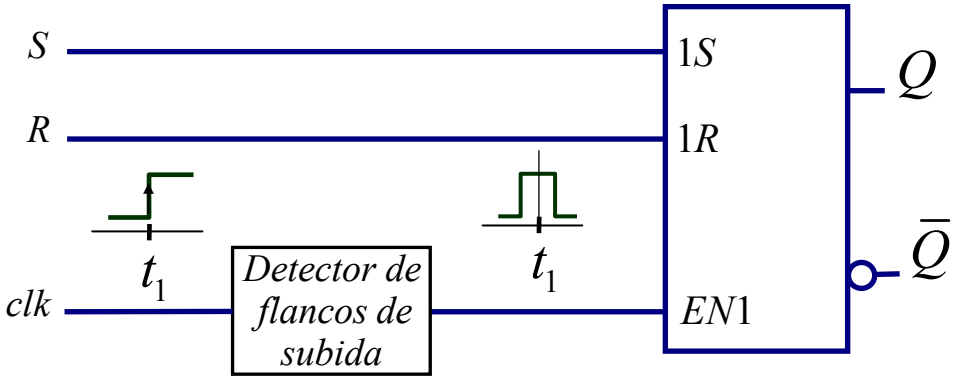
✓ Los *flip-flops* se pueden clasificar en dos grupos:

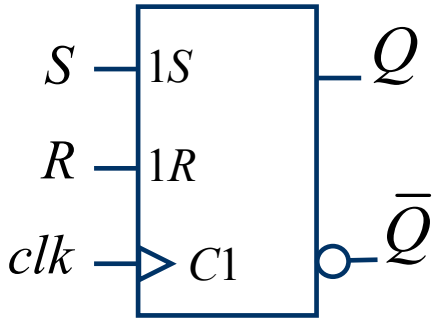
- Biestables *master-slave* (obsoletos!!!).
- Biestables disparados por flanco (*edge triggered*).

Biestables sincronizados por flanco (*edge triggered*)

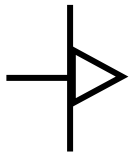
Se caracterizan por lo siguiente:

✓ Se comportan como biestables asíncronos (*latches*) con una entrada de habilitación de las entradas de información.

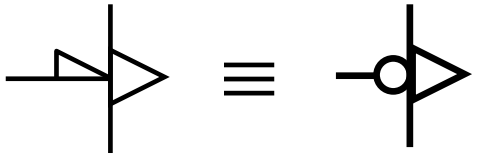




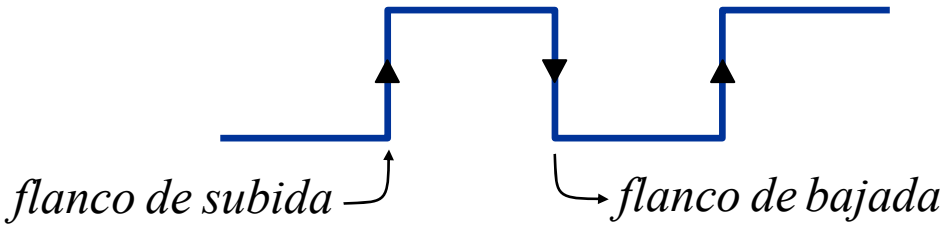
✓ Las entradas de información están habilitadas durante intervalos de tiempo muy pequeños. Dichos intervalos están centrados en torno a los flancos de subida o bien de bajada que describe la señal de sincronismo (≡ señal de reloj ≡ *clk*).

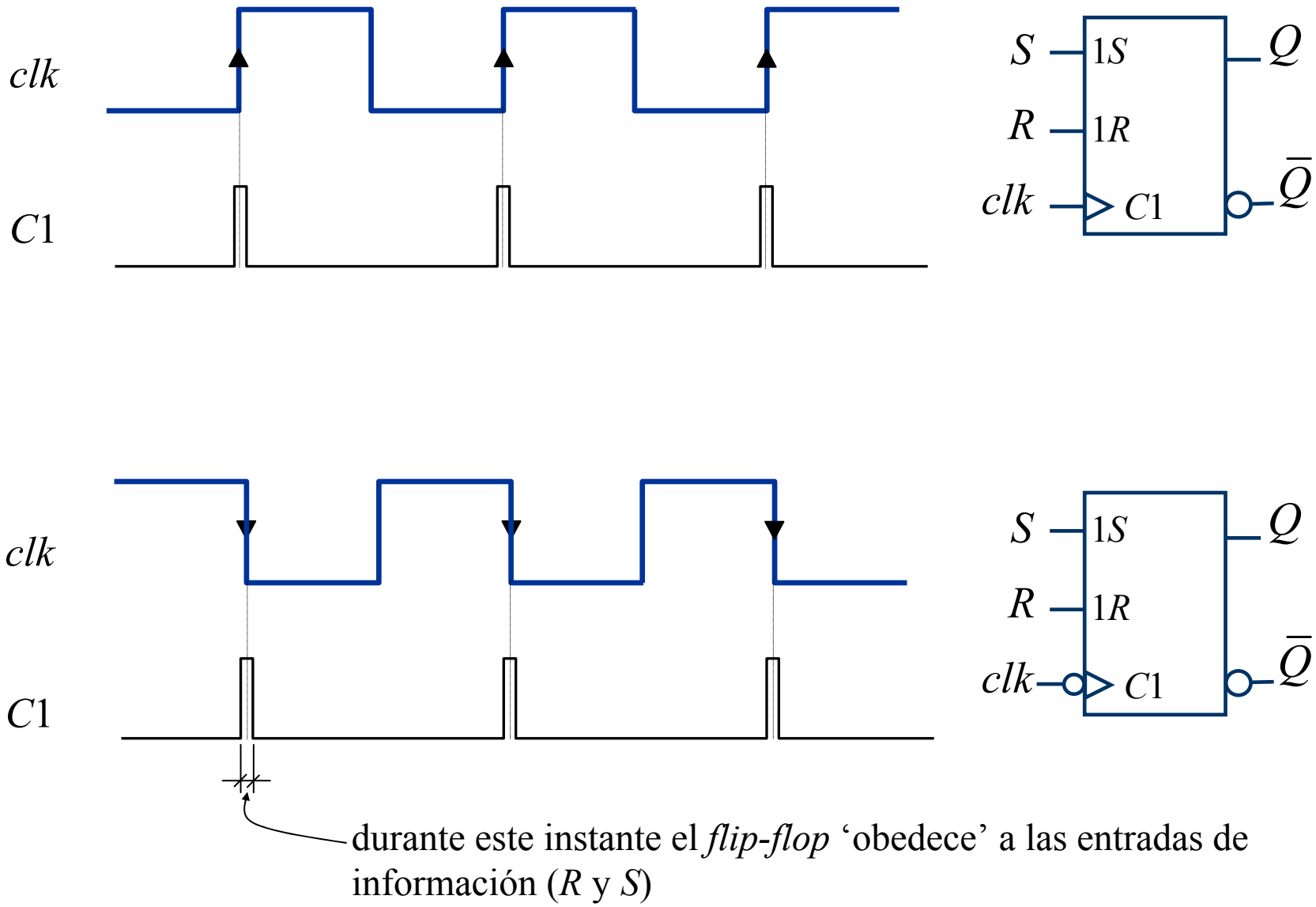


entrada dinámica sensible a los flancos de subida



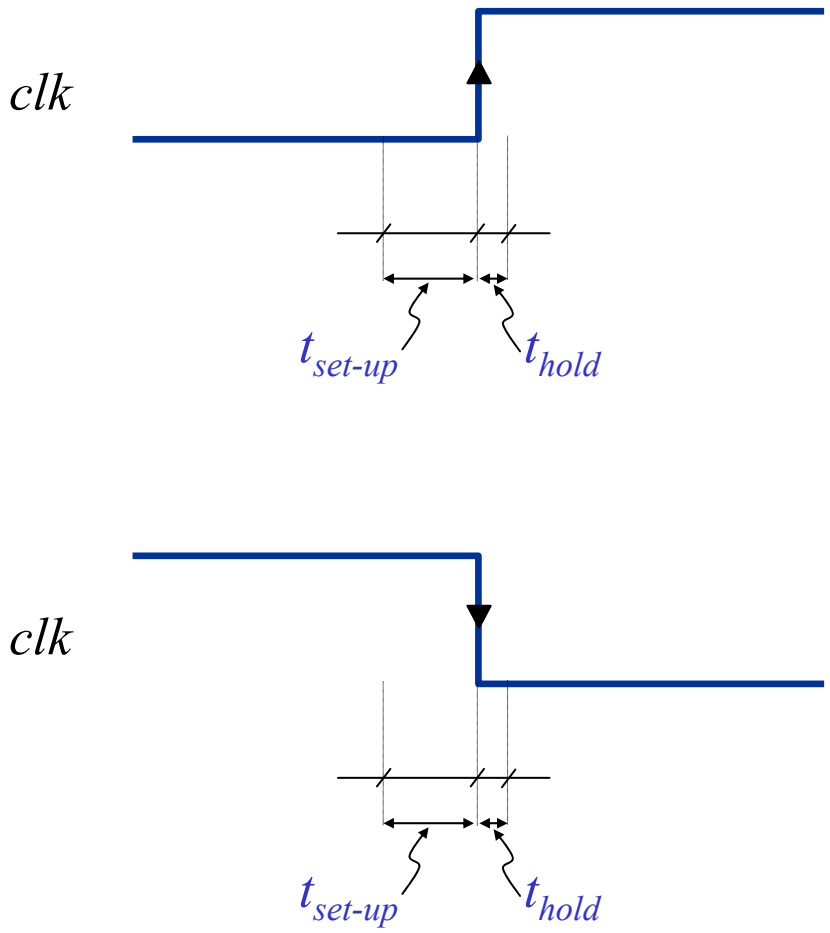
entrada dinámica sensible a los flancos de bajada



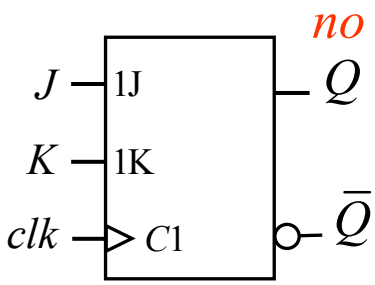

clk
C1
S
1S
R
1R
clk
C1
Q
$\bar{Q}$
clk
C1
S
1S
R
1R
clk
C1
Q
$\bar{Q}$
durante este instante el *flip-flop* 'obedece' a las entradas de información (*R* y *S*)

no

✓ El valor de las entradas de información debe permanecer *constante* durante el tiempo que están habilitadas ($t_{set\text{-}up}$ + t_{hold}). Si no se cumple esta condición, el biestable entra en un estado *metaestable*.



Flip-flop J-K disparado por flanco:



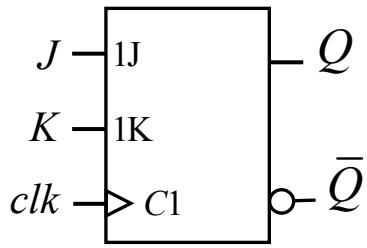
- Tiene 2 entradas de información (J y K) y una entrada de sincronismo (clk)

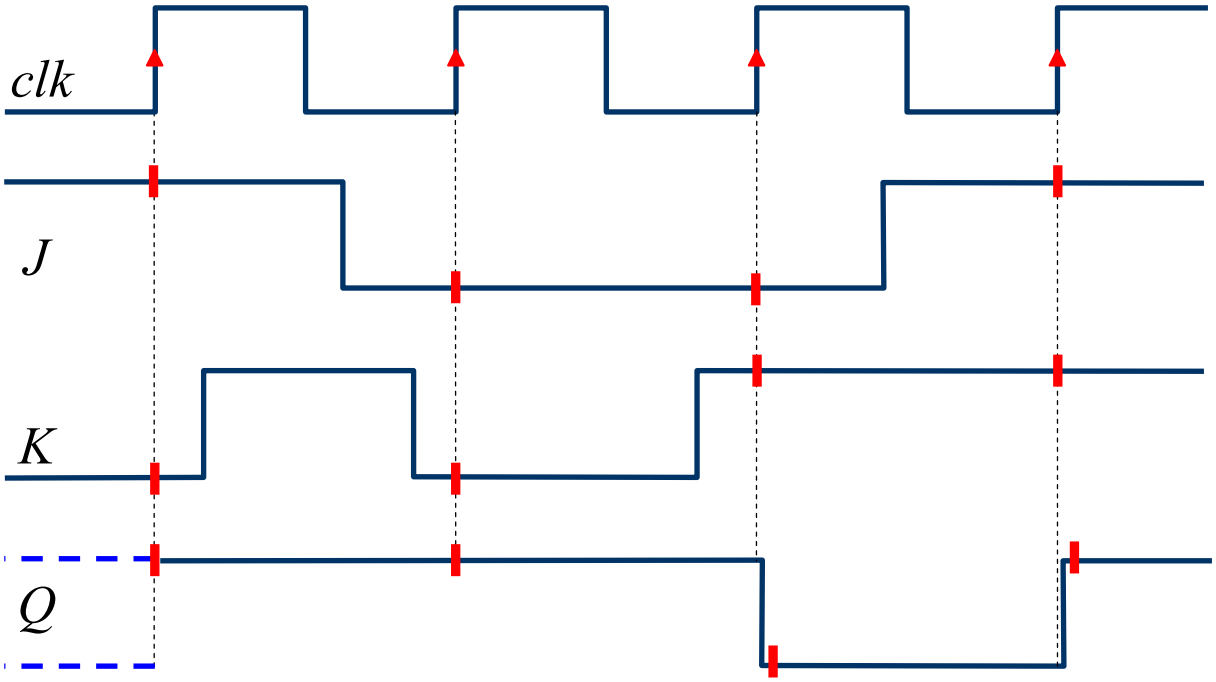
- Definición de su comportamiento:

clk	J_t	K_t	Q_t	$Q_{t+\Delta t}$	
↑	0	0	0	0	*no hagas nada, quédate como estás* $(Q_{t+\Delta t} = Q_t)$.
↑	0	0	1	1	
↑	0	1	0	0	*ponte a memorizar un cero* $(Q_{t+\Delta t} = 0)$.
↑	0	1	1	0	
↑	1	0	0	1	*ponte a memorizar un uno* $(Q_{t+\Delta t} = 1)$.
↑	1	0	1	1	
↑	1	1	0	1	*cambia el valor guardado*
↑	1	1	1	0	

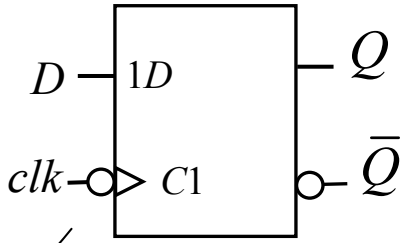
esta tabla sólo es cierta durante los flancos de subida de la señal de sincronismo *clk*

no





Flip-flop *D* disparado por flanco:



flip-flop D sincronizado con los flancos de bajada de clk

- Tiene 1 entrada de información (*D*) y 1 entrada de sincronismo (*clk*)
- Definición de su comportamiento:

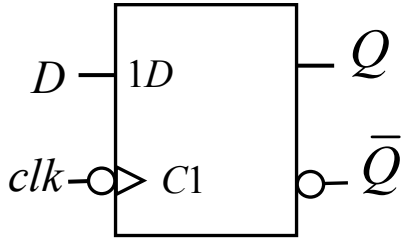
valores antes de que la señal clk describa un flanco de bajada

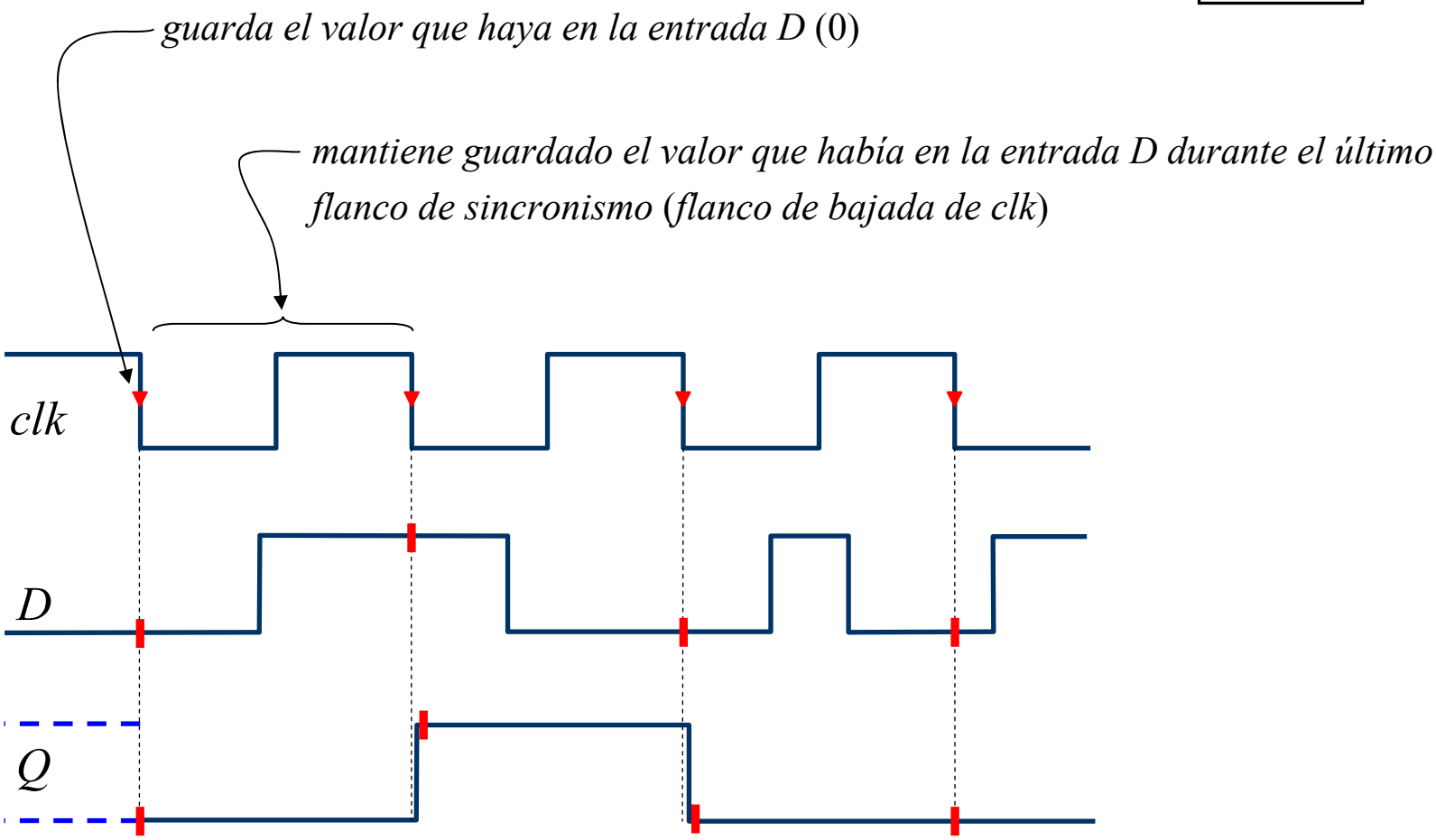
valor guardado por el flip flop después de que la señal clk describa un flanco de bajada

clk	D_t	Q_t	$Q_{t+\Delta t}$	
↓	0	0	0	*memoriza un cero* $\left(Q_{t+\Delta t} = D_t = 0\right)$.
↓	0	1	0	
↓	1	0	1	*memoriza un uno* $\left(Q_{t+\Delta t} = D_t = 1\right)$.
↓	1	1	1	

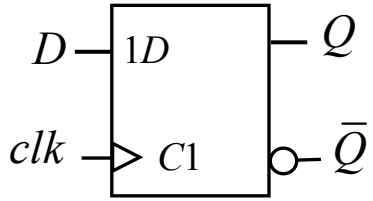
esta tabla sólo es cierta durante los flancos de bajada de la señal de sincronismo (clk)

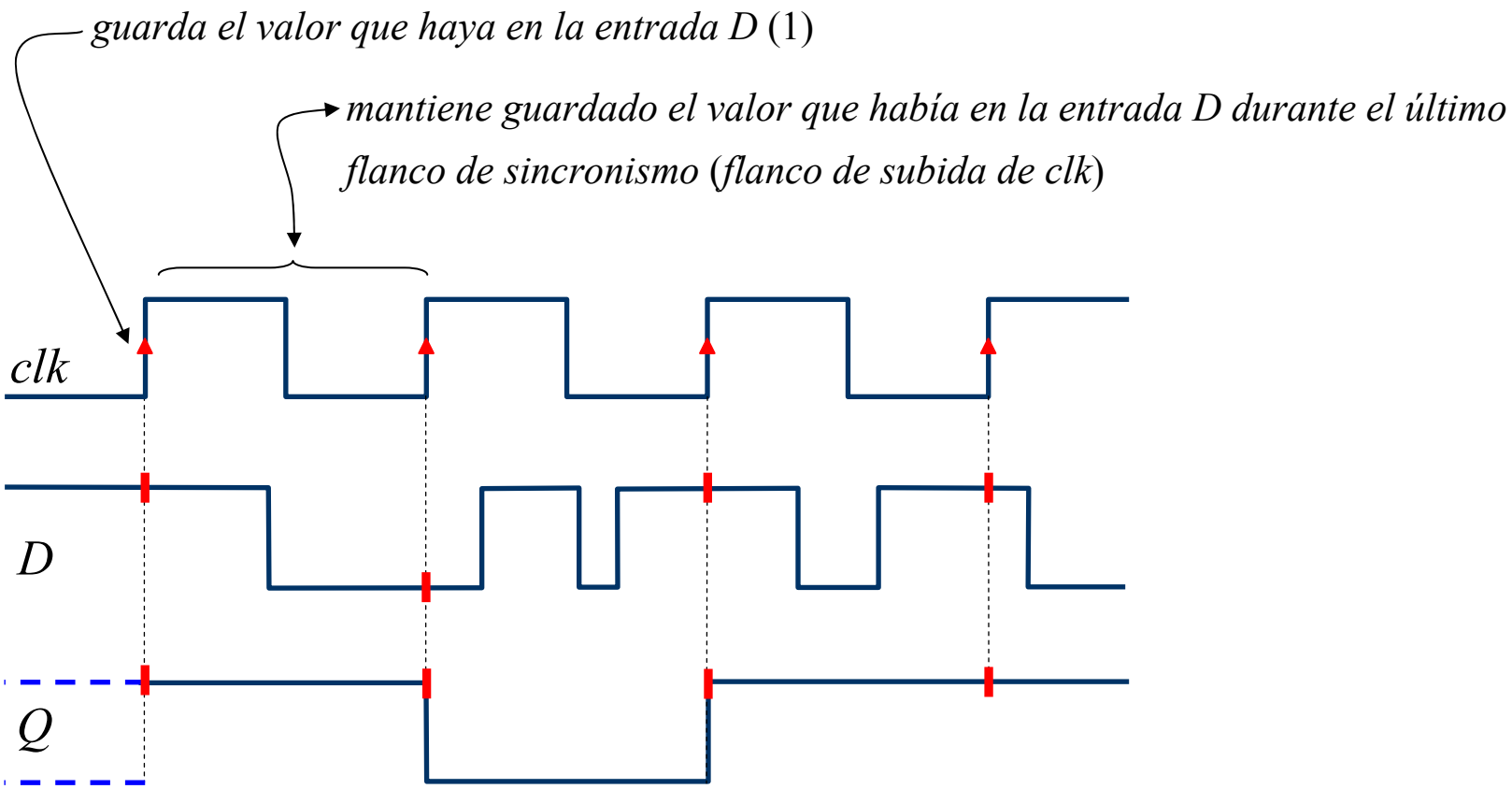
- *Flip-flop D sincronizado con los flancos de bajada de clk*





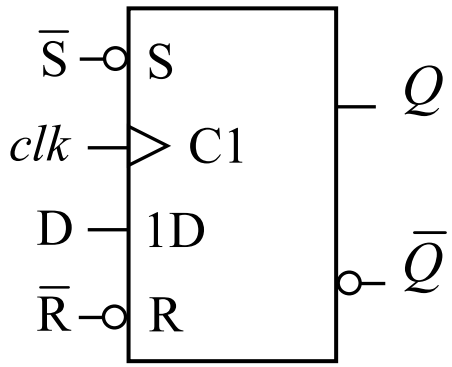
- *Flip-flop D sincronizado con los flancos de subida de clk*





Circuito 74*xxx*74

no



clk	$\overline{S}$	$\overline{R}$	D	Q	$Q_{t+\Delta t}$	
x	0	0	x	x	*prohibida*	*funcionamiento asíncrono*
x	0	1	x	x	1	
x	1	0	x	x	0	
↑	1	1	0	x	0	*funcionamiento síncrono*
↑	1	1	1	x	1	

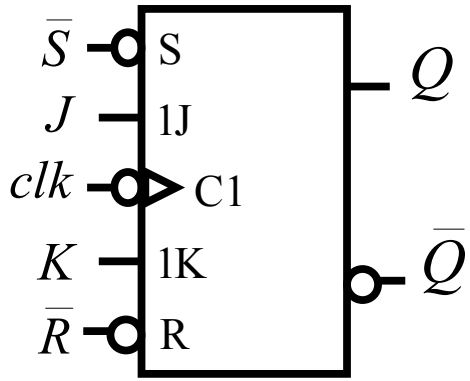
✓ Las entradas asíncronas <u>siempre</u> prevalecen sobre las entradas síncronas.

_ Entradas de información asíncronas: $\overline{S}$, $\overline{R}$

_ Entadas de información síncronas: D

no

Circuito 74*xxx*112

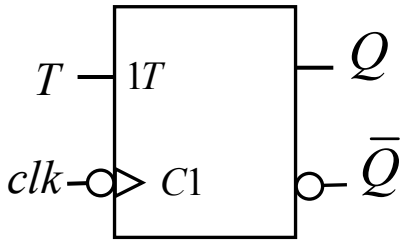


clk	J	K	Q_t	$\bar{S}$	$\bar{R}$	$Q_{t+\Delta t}$	
x	x	x	x	0	0	*prohibida*	*funcionamiento asíncrono*
x	x	x	x	0	1	1	
x	x	x	x	1	0	0	
↓	0	0	x	1	1	Q_t	*funcionamiento síncrono*
↓	0	1	x	1	1	0	
↓	1	0	x	1	1	1	
↓	1	1	x	1	1	$\bar{Q}_t$	

✓ Las entradas asíncronas prevalecen siempre sobre las entradas síncronas.

no

Flip-flop *T* disparado por flanco (de bajada):



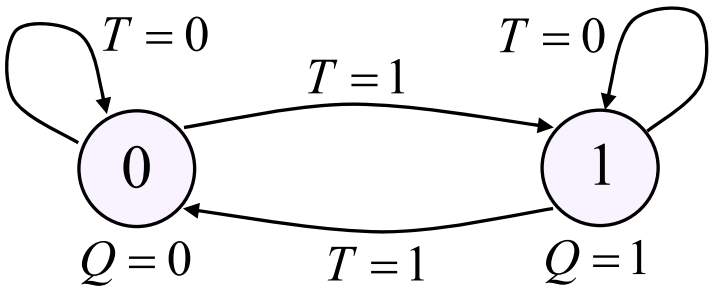
- Tiene 1 entrada de información (*T*) y 1 entrada de sincronismo (*clk*)
- Definición de su comportamiento:

clk	T_t	Q_t	$Q_{t+\Delta t}$
↓	0	0	0
↓	0	1	1
↓	1	0	1
↓	1	1	0

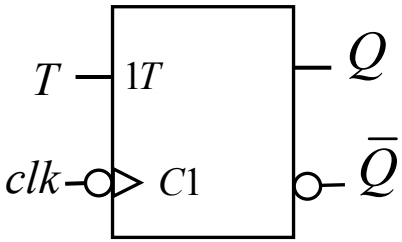
permanece en el estado en el que se encuentre $\left(Q_{t+\Delta t} = Q_t\right)$.

cambia de estado $\left(Q_{t+\Delta t} = \bar{Q}_t\right)$.

esta tabla sólo es cierta durante los flancos de bajada de la señal de sincronismo



no



Notas:

En el diagrama de flujo anterior, las transiciones sólo pueden tener lugar durante los flancos de bajada de la señal de sincronismo (clk)

Los flip flops de tipo T aparecen en las librerías de los programas de configuración de las FPGAs y de las CPLDs

- Procedimiento general de diseño de un sistema secuencial síncrono.

Descripción del funcionamiento del circuito a diseñar

↓

_ Se determina el número de entradas del circuito a diseñar y se le asigna una variable lógica (distinta) a cada una de ellas.

_ Se determina el número de salidas del circuito a diseñar y se le asigna una función lógica (distinta) a cada una de ellas.

(1) *Se dibuja un diagrama de flujo que describa el comportamiento del circuito a diseñar.*

alternativo

Se determina la tabla de funcionamiento del sistema. Dicha tabla está formada por la tabla de transición de estados y por la tabla de verdad del circuito combinacional de salida (ccs).

↓

Se minimiza el número de estados del modelo planteado (opcional)

Se codifican los estados del modelo propuesto, asignándole un número binario distinto a cada uno. El número de dígitos de los 'nombres (≡ números)' asignados a los distintos estados establece el número de variables de estado que se necesitan para representar el estado (interno) del sistema.

Cada vez que el sistema se encuentre en un determinado estado, las variables de estado (interno) tomarán el valor (número) asignado a dicho estado (interno).

Se elige el tipo de flip-flops que se van a utilizar en el bloque de memoria para guardar el valor de las variables de estado ≡ valor del estado interno. El valor de cada variable de estado se guardará en 1 flip-flop.

Se determina la tabla de excitación de los flip-flops elegidos.

Se determina la tabla de verdad del circuito combinacional del estado siguiente (cces)

Se determinan las expresiones más sencillas de las funciones que describen el comportamiento a nivel lógico de los circuitos combinacionales del estado siguiente y de salida.

↓

Dibujar un esquema del circuito.

↓

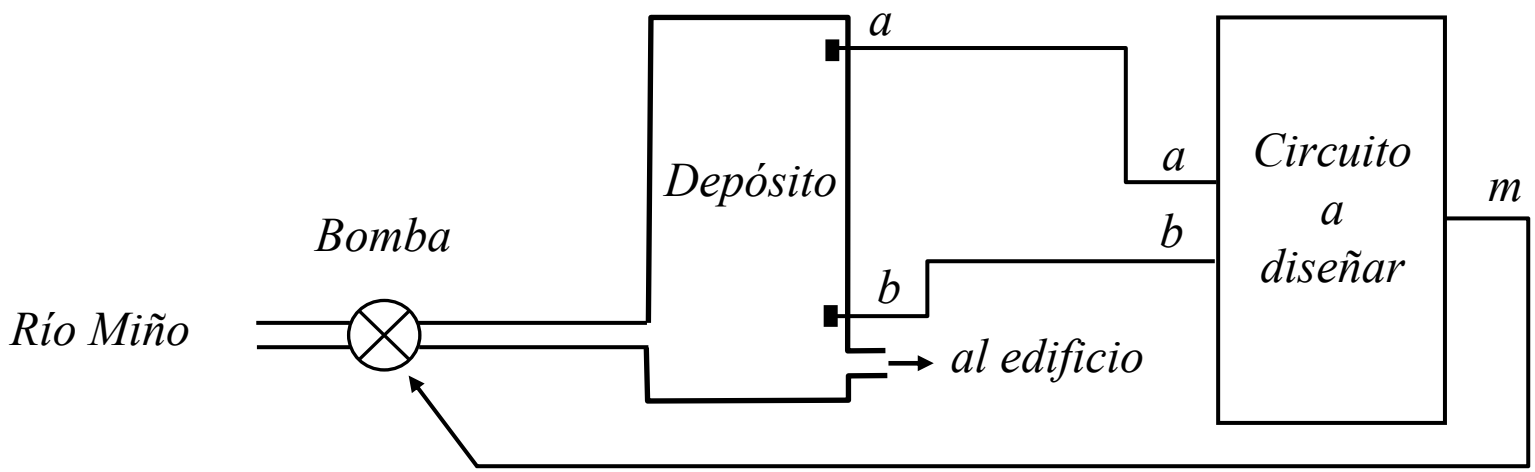
Simular el funcionamiento del circuito diseñado (*opcional*). *Si el circuito diseñado no funciona de acuerdo con las especificaciones se vuelve al punto* (1)

↓

Construcción de un prototipo. Si su funcionamiento no cumple las especificaciones se vuelve al punto (1)

Ejemplo de obtención de un diagrama de flujo (*modelo de Moore*) I :

no



$$a = \begin{cases} 1 & \text{si el nivel del agua supera la posición del sensor (a)} \\ 0 & \text{si el nivel del agua es inferior a la posición del sensor (a)} \end{cases}$$

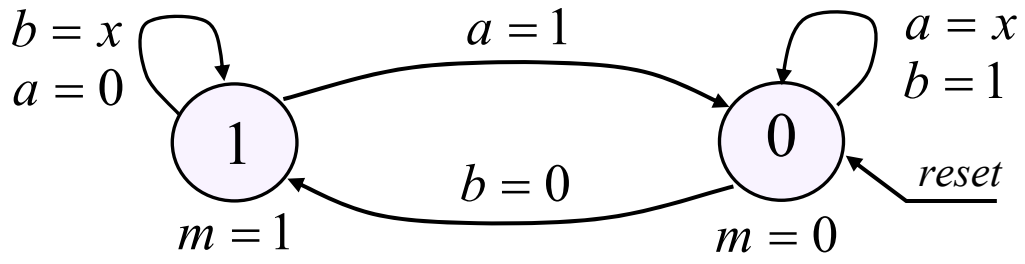
$$b = \begin{cases} 1 & \text{si el nivel del agua supera la posición del sensor (b)} \\ 0 & \text{si el nivel del agua es inferior a la posición del sensor (b)} \end{cases}$$

$$m = \begin{cases} 1 & \text{el motor bombea agua al depósito} \\ 0 & \text{el motor está parado} \end{cases}$$

no

El sistema secuencial que controla el funcionamiento del motor debe garantizar en todo momento que el nivel del agua en el interior del depósito esté siempre comprendido entre los niveles que establecen los sensores *a* y *b*.

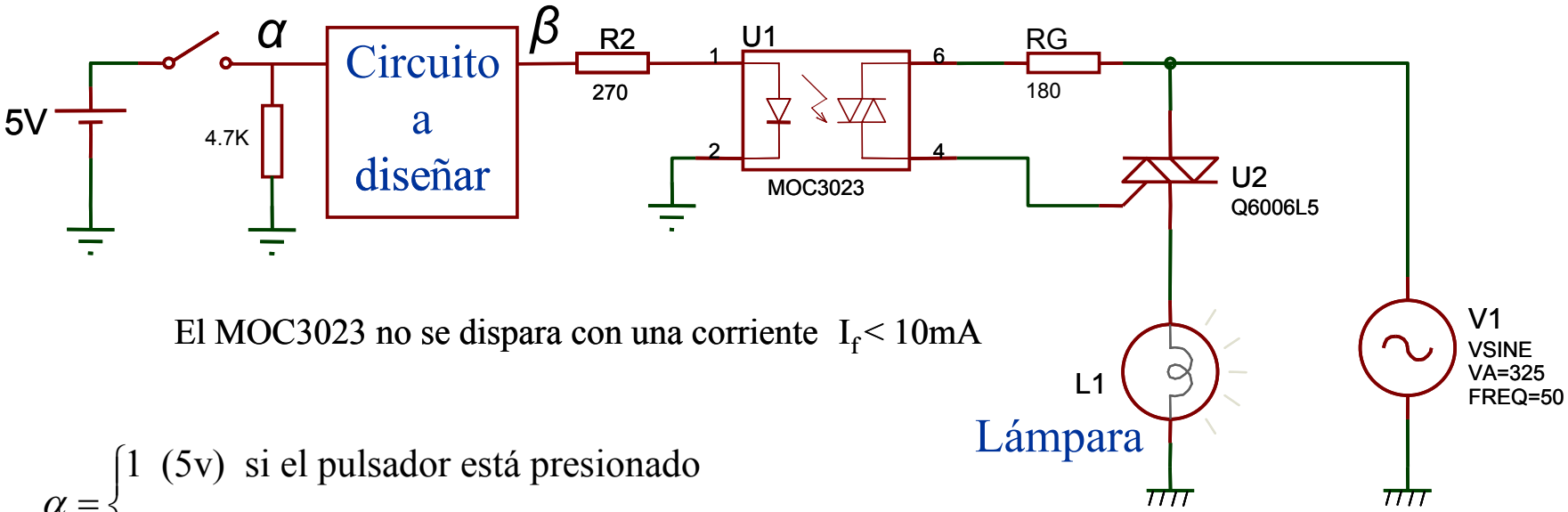
Diagrama de flujo (*modelo de Moore*):



En el estado 0, el motor está parado (m = 0) y el depósito se está vaciando. El sistema se mantiene en este estado mientras el nivel del agua sea superior al marcado por el sensor *b*. Cuando el nivel descienda por debajo de la posición del sensor *b* (b = 0), el motor se pondrá en funcionamiento (el sistema pasa al estado 1) hasta que se llene el deposito de agua (a = 1). En cuyo momento se deberá detener el motor (el sistema pasa al estado 0).

Ejemplo de obtención de un diagrama de flujo (*modelo de Moore*) II: *no*

En este caso se trata de diseñar un sistema de control que haga que cada vez que se presione un pulsador α se encienda una bombilla, en el caso de que esté apagada, o se apague en el caso de que esté encendida.



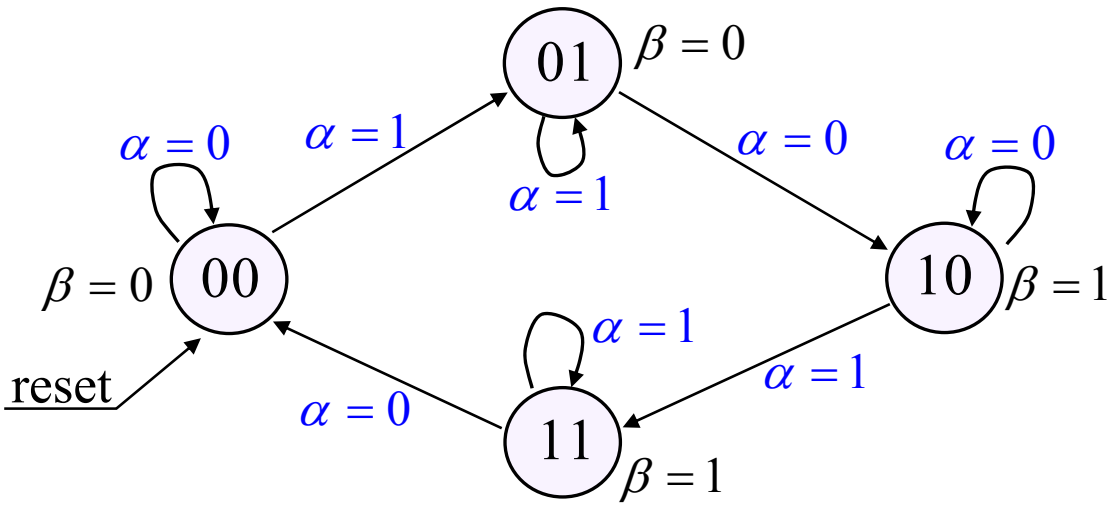
El MOC3023 no se dispara con una corriente $I_f < 10mA$

$$\alpha = \begin{cases} 1 \ (5v) \ \text{si el pulsador está presionado} \\ 0 \ (0v) \ \text{si el pulsador no está presionado} \end{cases}$$

$$\beta = \begin{cases} 1 \ \text{la bombilla está encendida} \\ 0 \ \text{la bombilla está apagada} \end{cases}$$

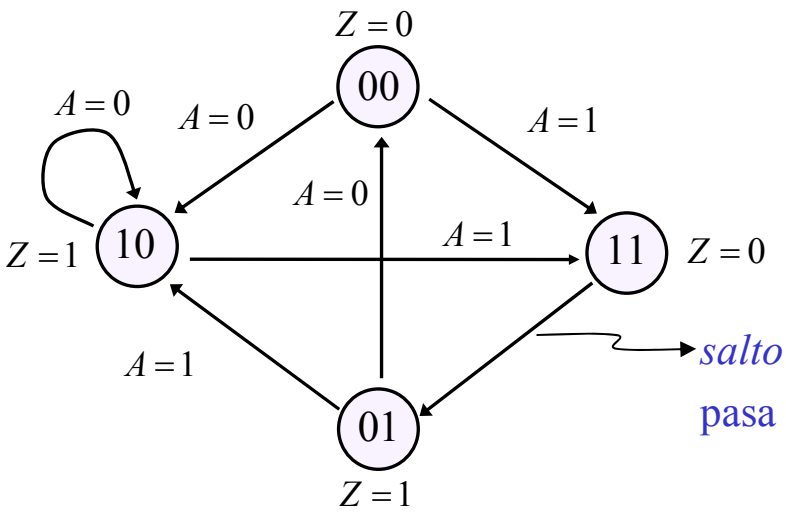
Nota: pulsador $\neq$ interruptor

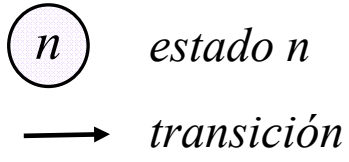
Diagrama de flujo (*modelo de Moore*): *no*



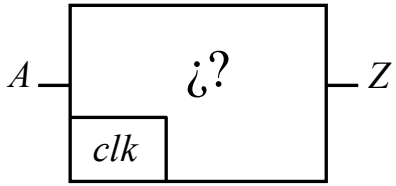
En el *estado* 0, la bombilla está apagada ($\beta = 0$) y el pulsador no está presionado ($\alpha = 0$). Estando en este estado, si se presiona el pulsador ($\alpha = 1$), el sistema pasa al *estado* 1, en el que permanece hasta que se deja de presionar el pulsador ($\alpha = 0$), en cuyo momento el sistema pasa al *estado* 2. En dicho estado la bombilla está encendida ($\beta = 1$) y el pulsador no está presionado ($\alpha = 0$). Estando en el estado 2, si se presiona el pulsador el sistema pasa al estado 3, en el que permanece hasta que se deja de presionar el pulsador. En cuyo momento el sistema pasa al estado 0, en el que la bombilla permanece apagada.

Ejemplo de implementación de un diagrama de flujo (*modelo de Moore*)





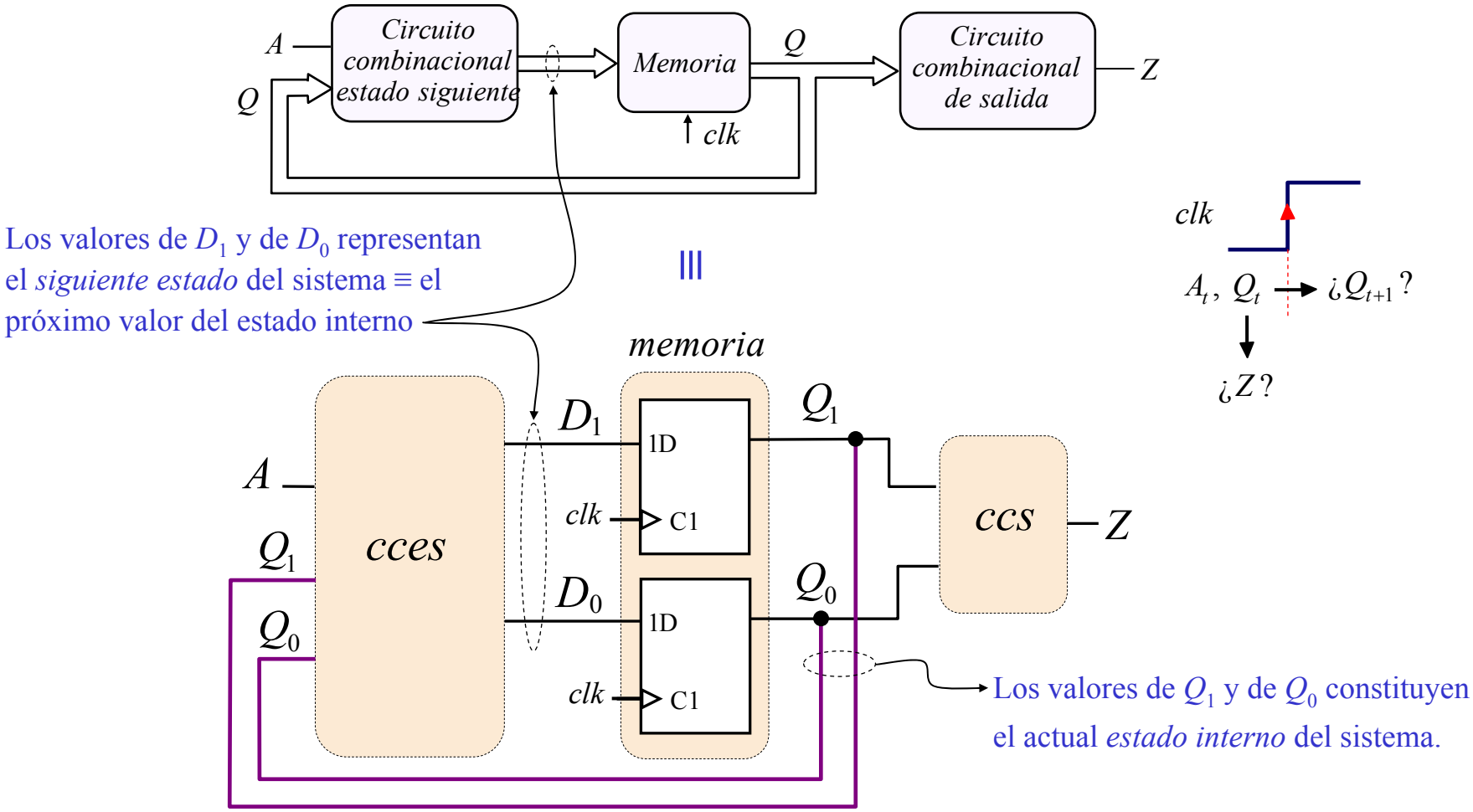
salto incondicional: estando en el estado 3, el sistema pasa al estado 1 tanto si $A = 0$ como si $A = 1$.



Del diagrama de flujo se deduce lo siguiente:

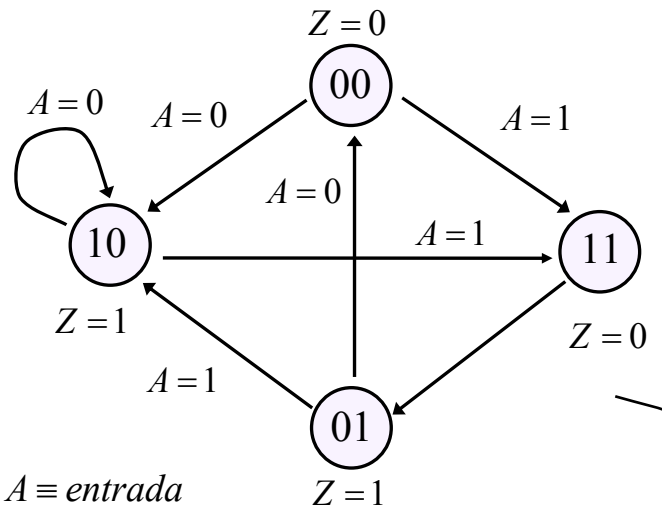
- *tiene una única entrada*: A
- *tiene una única salida*: Z
- *tiene* 4 *estados. El estado más alto es* $3_{10} = 11_2 \Rightarrow$ *el sistema tendrá* 2 *variables de estado* $Q(Q_1, Q_0)$ *y, por lo tanto, el bloque de memoria estará formado por* 2 *flip-flops* (*en este caso se van a utilizar flip-flops de tipo D*).

El diagrama de bloques del sistema secuencial a diseñar es el siguiente:



Nota: el sistema sólo puede cambiar de estado durante los flancos de subida de la señal *clk*. ¿por qué?

Para poder diseñar los *circuitos combinacionales del estado siguiente* (*cces*) y *de salida* (*ccs*) hay que determinar la *tabla de transición de estados* y la *tabla del circuito combinacional de salida* a partir del *diagrama de flujo* o *de estados*:



$A \equiv$ *entrada*

$Z \equiv$ *salida*

tabla de verdad del ccs

Q_{1t}	Q_{0t}	Z
0	0	0
0	1	1
1	0	1
1	1	0

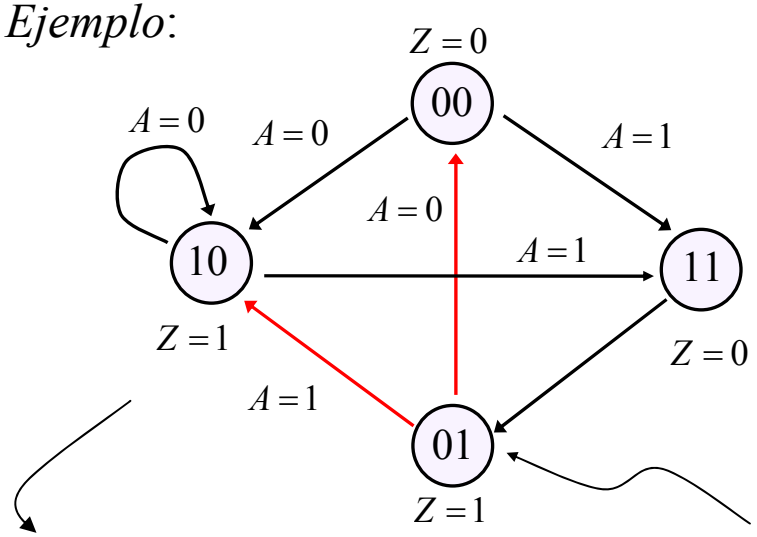
tabla de transición de estados

Q_{1t}	Q_{0t}	A_t	Q_{1t+1}	Q_{0t+1}
0	0	0	1	0
0	0	1	1	1
0	1	0	0	0
0	1	1	1	0
1	0	0	1	0
1	0	1	1	1
1	1	0	0	1
1	1	1	0	1

estado actual (*dura hasta el próximo flanco de sincronismo*)

estado siguiente (*será el estado en el que estará el sistema a partir del próximo flanco de sincronismo*)

Ejemplo:



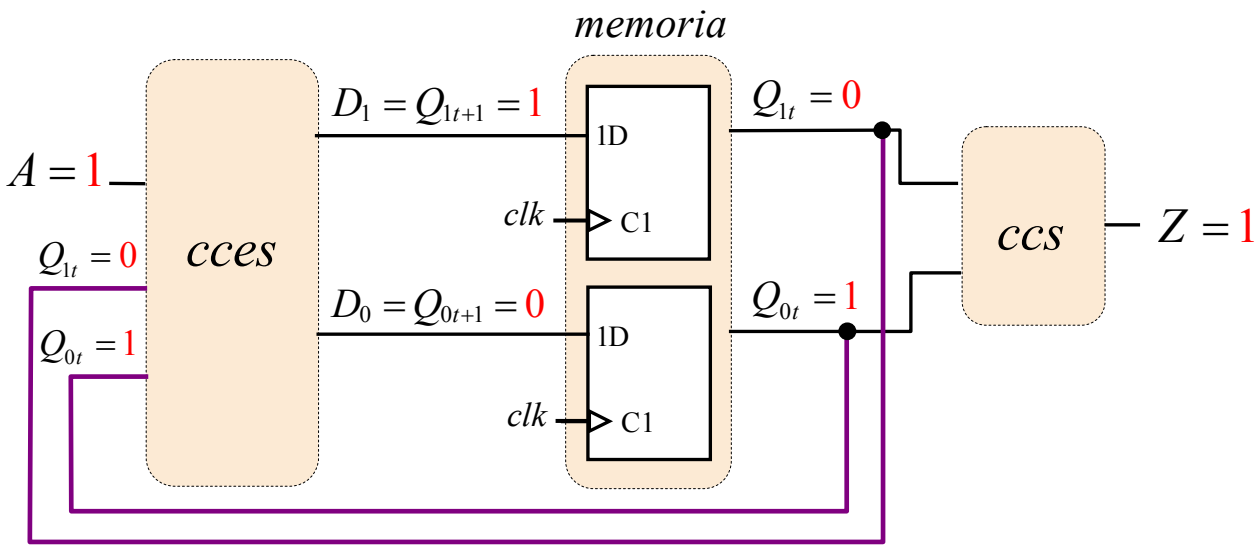
Q_{1t}	Q_{0t}	A_t	Q_{1t+1}	Q_{0t+1}
0	0	0	1	0
0	0	1	1	1
0	1	0	0	0
0	1	1	1	0
1	0	0	1	0
1	0	1	1	1
1	1	0	0	1
1	1	1	0	1

Siempre que el sistema esté en el estado $Q_{1t}\,Q_{0t} = 01$, se cumple que:

_ la salida es $Z = 1$

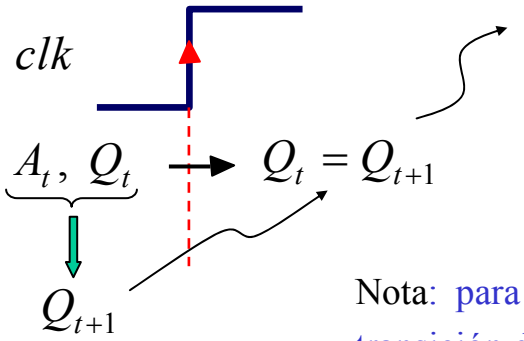
_ si $A = 1$, a partir del próximo flanco de subida de *clk*, el estado del sistema será el 10

_ si $A = 0$, a partir del próximo flanco de subida de *clk*, el estado del sistema será el 00



Q_{1t}	Q_{0t}	Z
0	0	0
0	1	1
1	0	1
1	1	0

La *tabla de transición de estados* indica el nuevo valor (Q_{t+1}) que tendrá el estado interno después del próximo flanco de sincronismo, en función de los valores actuales del estado (Q_t) y de la entrada (A_t). Para que se produzca el cambio de estado correcto durante los flancos de sincronismo de la señal *clk* es necesario que los valores D_{1t} y D_{0t} presentes en las entradas de los *flip-flops*, tanto antes como durante un flanco de sincronismo, correspondan a los valores Q_{1t} y Q_{0t} que deberá tener el estado interno del sistema después de que ocurra dicho flanco de sincronismo (y que en la tabla de transición de estados se representan por Q_{1t+1} y Q_{0t+1}). Los valores de D_{1t} y de D_{0t} los genera el *circuito combinacional del estado siguiente* a partir del estado actual (Q_{1t} y Q_{0t}) del sistema y del valor actual de la entrada (A_t).



después del flanco de sincronismo de *clk*, el nuevo valor de Q_t es igual al valor $Q_{t+1} = D_t$ existente antes (y durante) el flanco de sincronismo de *clk*.

Nota: para cada combinación de valores de Q_t y A_t, en la tabla de transición de estados se indica el nuevo valor (Q_{t+1}) que tendrá el estado interno Q_t del sistema después del próximo flanco de sincronismo.

Para determinar el valor D_t (D_{1t}, D_{0t}) que debe haber en las entradas de información de los *flip flops* resulta útil determinar previamente la *tabla de excitación* de los biestables que se vayan a utilizar en el bloque de memoria.

clk ↑

D_t	Q_t	Q_{t+1}
0	0	0
0	1	0
1	0	1
1	1	1

$$Q_{t+1} = D_t$$

clk ↑

$Q_t \rightarrow Q_{t+1}$	D_t
$0 \rightarrow 0$	0
$0 \rightarrow 1$	1
$1 \rightarrow 0$	0
$1 \rightarrow 1$	1

$\Rightarrow D_t = Q_{t+1}$

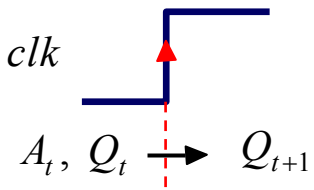
tabla de excitación de un biestable D

De la tabla de excitación de un *flip flop* de tipo *D* se deduce que el valor que debe de haber en la entrada de información (D_t) *antes de* cada flanco de sincronismo debe ser igual al valor (Q_{t+1}) que debe de guardar el correspondiente *flip flop* *después de* cada flanco de sincronismo.

A partir de la tabla de transición de estados y de la tabla de excitación de un *flip flop D* se deduce la siguiente tabla de verdad de las funciones $D_1(Q_{1t},Q_{0t},A)$ y $D_0(Q_{1t},Q_{0t},A)$ a implementar por el circuito combinacional de entrada (*cce*):

Q_{1t}	Q_{0t}	A	Q_{1t+1}	Q_{0t+1}	D_1	D_0
0	0	0	1	0	1	0
0	0	1	1	1	1	1
0	1	0	0	0	0	0
0	1	1	1	0	1	0
1	0	0	1	0	1	0
1	0	1	1	1	1	1
1	1	0	0	1	0	1
1	1	1	0	1	0	1

$Q_t \rightarrow Q_{t+1}$	D_t
$0 \rightarrow 0$	0
$0 \rightarrow 1$	1
$1 \rightarrow 0$	0
$1 \rightarrow 1$	1



Nota: en el caso de utilizar *flip flops D* para guardar el *estado interno* del sistema, se cumple que: $D_1(Q_{1t}, Q_{0t}, A) = Q_{1t+1}$ y $D_0(Q_{1t}, Q_{0t}, A) = Q_{0t+1}$

A partir de las tablas de verdad de los circuitos combinacionales del *estado siguiente* (*cces*) y de *salida* (*ccs*), se deducen las siguientes expresiones lógicas:

$$D_{1t}(Q_{1t}, Q_{0t}, A_t) = \sum_3 (0,1,3,4,5)$$

$$D_{0t}(Q_{1t}, Q_{0t}, A_t) = \sum_3 (1,5,6,7)$$

$$Z(Q_{1t}, Q_{0t}) = \sum_2 (1,2)$$

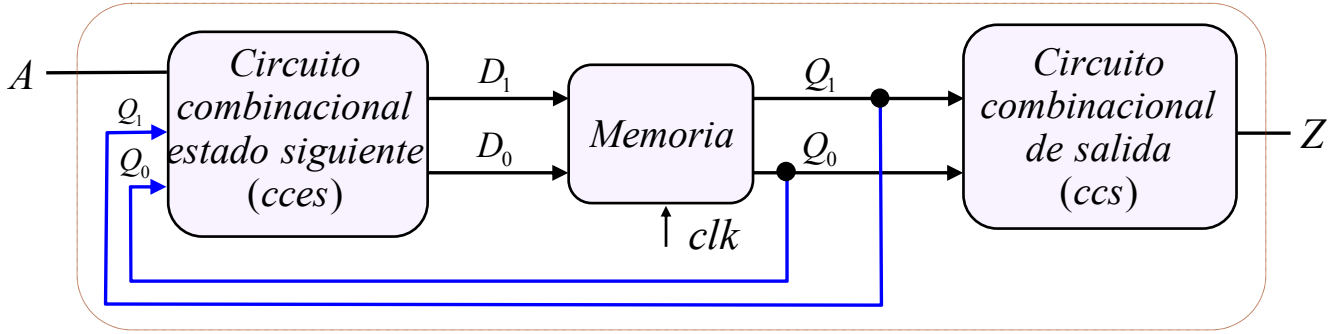
Simplificando por *Karnaugh-Veitch* las funciones anteriores se obtiene lo siguiente:

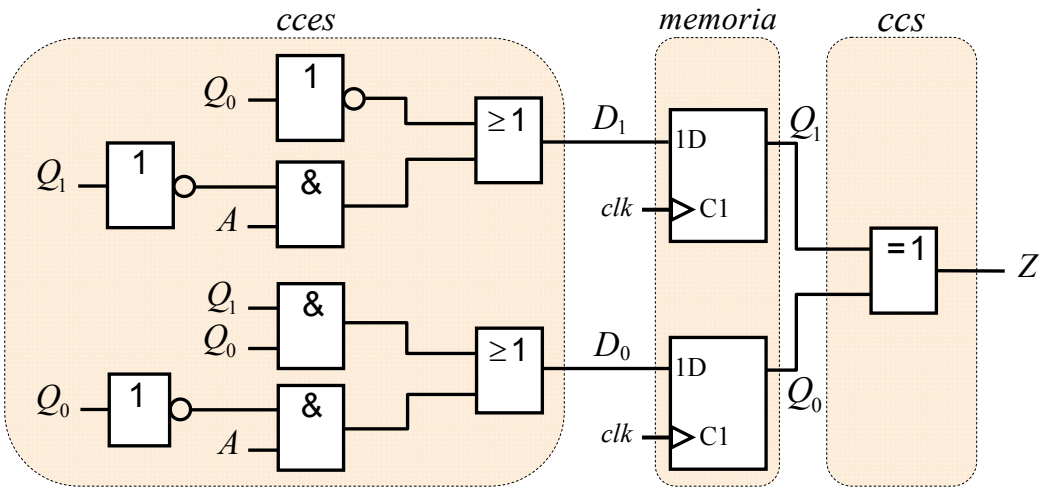
$$D_{1t}(Q_{1t}, Q_{0t}, A_t) = \bar{Q}_{1t} A_t + \bar{Q}_{0t}$$

$$D_{0t}(Q_{1t}, Q_{0t}, A_t) = \bar{Q}_{0t} A_t + Q_{1t} Q_{0t}$$

$$Z(Q_{1t}, Q_{0t}) = \bar{Q}_{1t} Q_{0t} + Q_{1t} \bar{Q}_{0t} = Q_{1t} \oplus Q_{0t}$$

A partir de las expresiones anteriores se deduce el siguiente circuito:



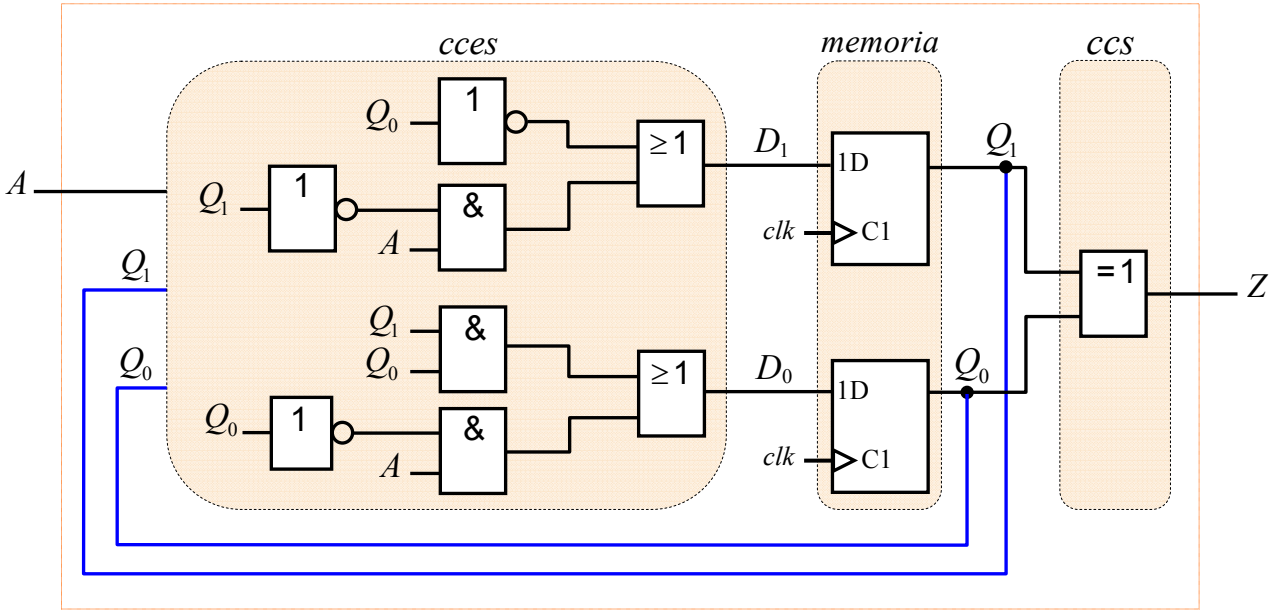


$$D_1(Q_1, Q_0, A) = \bar{Q}_1 A + \bar{Q}_0$$

$$D_0(Q_1, Q_0, A) = \bar{Q}_0 A + Q_1 Q_0$$

$$Z(Q_1, Q_0) = Q_1 \oplus Q_0$$

Nomenclatura: $X_t = X$

* *clk* es una señal auxiliar que sincroniza el funcionamiento de los *flip - flops*

El circuito anterior, indicando las realimentaciones de las variables de estado: