

Tema 12: Sistemas Secuenciales (BIESTABLES)

Objetivos:

Introducción.

Biestables Asíncronos

Biestables Síncronos.

Entradas asíncronas.

Parámetros de los biestables.

INTRODUCCIÓN

- En los circuitos secuenciales, el estado de la salida depende no sólo del estado lógico de las entradas, sino también de la historia pasada del circuito (por ejemplo Contador).
- El nivel lógico de la salida puede ser diferente para la misma combinación de entradas, dependiendo del estado del que se viene
- Genéricamente, un circuito secuencial está formado por una parte combinacional y elementos de memoria

INTRODUCCIÓN

- En los circuitos secuenciales se distinguen dos tipos en función de su modo de funcionamiento :

» ASíncronos

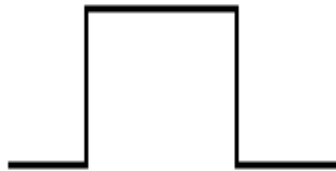
» Síncronos

Los circuitos secuenciales **Asíncronos** son aquellos circuitos secuenciales en los cambios en el estado lógico de sus salidas se producen en el instante en que se modifican las variables de entrada (sin olvidar los retardos propios de los dispositivos electrónicos)

- Los circuitos secuenciales **Síncronos** , además de las señales de entrada, es necesaria la presencia de una señal particular, llamada de sincronismo o reloj para que se propague a la salida.

INTRODUCCIÓN

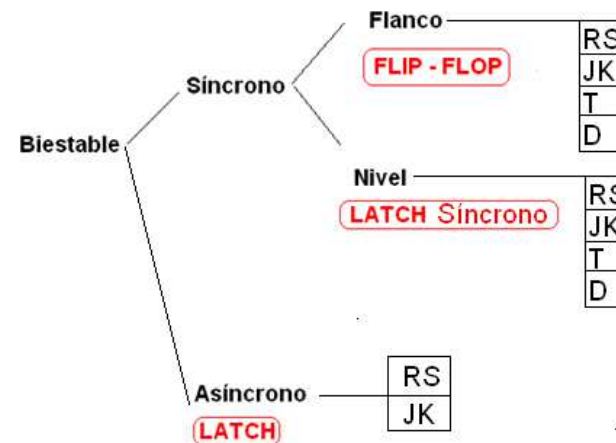
- En los circuitos secuenciales asíncronos, el nivel lógico de las salidas sólo cambia en determinados instantes de la señal de sincronismo. El tipo de sincronismo puede ser:



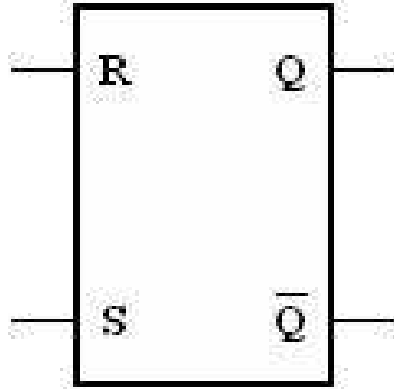
- » Síncrono a nivel alto (Latch)
- » Síncrono a nivel bajo (Latch)
- » Síncrono por flanco de subida (Flip – Flop)
- » Síncrono por flanco de bajada (Flip – Flop)

BIESTABLES

• Los **biestables** son circuitos lógicos secuenciales caracterizados por su capacidad de permanecer indefinidamente en uno cualquiera de sus dos estados estables (H ó L) aunque desaparezca la señal de excitación que provocó la transición al estado en el que se encuentra, debido a esta característica, podemos denominarlos “**elementos de memoria de un solo bit**”.



BIESTABLES ASINCRONOS



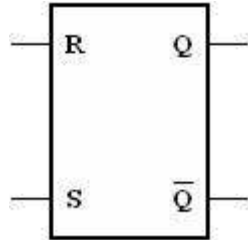
- **Biestables RS asíncronos**

- Existen dos formas de implementar biestables RS, basándonos en puertas NOR o en puertas NAND.

- La entrada R será conocida como entrada Reset y su función será la de “*apagar*” el biestable, esto es, poner $Q=0$.
- La entrada S será conocida como entrada Set y su función será la de “*Encender*” el biestable, esto es, poner $Q=1$.
- Q será la salida del biestable.

BIESTABLES ASINCRONOS

- Latch R-S NOR



- Al aplicar una combinación de entradas en S, R su efecto se acusa en las salidas dependiente de los niveles lógicos de S y R y del estado lógico anterior del latch

- Se desea:

- Si $S=R=0$ la salida Q permanezca en el mismo estado indefinidamente.
- Si $S=1$ y $R=0$ “encendemos” el latch, es decir, la salida Q se pondrá o continuará a 1 dependiendo del estado anterior
- Si $S=0$ y $R=1$ “apagamos” el latch, es decir, la salida Q se pondrá o continuará a 0 dependiendo del estado anterior.
- Si $S=1$ y $R=1$ estamos ordenando el encendido y apagado simultáneo del biestable, lo cual no es lógico, por lo que será un estado prohibido.

BIESTABLES ASINCRONOS

- Latch R-S NOR

S	R	$Q_{(t)}$	$Q_{(t+1)}$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	X	X



S	R	$Q_{(t+1)}$	$\bar{Q}_{(t+1)}$
0	0	$Q_{(t)}$	$\bar{Q}_{(t)}$
0	1	0	1
1	0	1	0
1	1	NO USADO	

S \ R	00	01	11	10
0	0	0	X	1
1	1	0	X	1

S \ R	00	01	11	10
0	1	1	X	0
1	0	1	X	0

$Q_{(t+1)}$

$\bar{Q}_{(t+1)}$

$$Q_{(t+1)} = S \cdot \bar{R} + Q_{(t)} \cdot \bar{R} = \bar{R} \cdot (S + Q_{(t)})$$

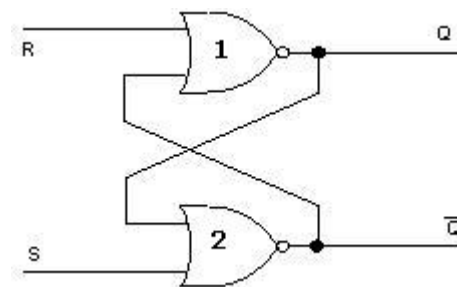
Aplicando los teoremas de Morgan

$$Q_{(t+1)} = \overline{\bar{R} \cdot (S + Q_{(t)})} = \bar{R} + \overline{(S + Q_{(t)})}$$

$$\bar{Q}_{(t+1)} = \bar{S} \cdot \bar{Q} + \bar{S} \cdot R = \bar{S} \cdot (\bar{Q} + R)$$

Aplicando los teoremas de Morgan

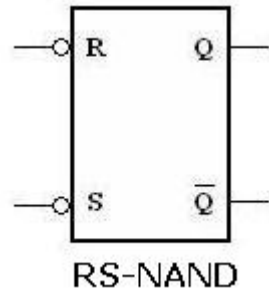
$$\bar{Q}_{(t+1)} = \overline{\bar{S} \cdot (\bar{Q} + R)} = S + \overline{(\bar{Q} + R)}$$



•Se puede ver que si $S=R=1$ simultáneamente ambas salidas son 0 lo cual no es lógico ya que una debe ser el complemento de la otra.

BIESTABLES ASINCRONOS

- LATCH R-S NAND



- El funcionamiento será similar al caso anterior exceptuando que la activación de las entradas se realizará con ceros en lugar que con unos.

- Se desea:

- Si $S=R=1$ la salida Q permanezca en el mismo estado indefinidamente.
- Si $S=0$ y $R=1$ “encendemos” el latch, es decir, la salida Q se pondrá o continuará a 1 dependiendo del estado anterior
- Si $S=1$ y $R=0$ “apagamos” el latch, es decir, la salida Q se pondrá o continuará a 0 dependiendo del estado anterior.
- Si $S=0$ y $R=0$ estamos ordenando el encendido y apagado simultáneo del biestable, lo cual no es lógico, por lo que será un estado prohibido.

BIESTABLES ASINCRONOS

• LATCH R-S NAND

S	R	$Q_{(t)}$	$Q_{(t+1)}$
0	0	X	X
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1



S	R	$Q_{(t+1)}$	$\bar{Q}_{(t+1)}$
1	1	$Q_{(t)}$	$\bar{Q}_{(t)}$
1	0	0	1
0	1	1	0
0	0	NO USADO	

S \ R	00	01	11	10
Q				
0	X	1	0	0
1	X	1	1	0

$Q_{(t+1)}$

S \ R	00	01	11	10
Q				
0	X	0	1	1
1	X	0	0	1

$\bar{Q}_{(t+1)}$

$$Q_{t+1} = (\bar{S} + Q) * (\bar{S} + R) = \bar{S} * \bar{S} + \bar{S} * R + Q * \bar{S} + Q * R$$

$$= \bar{S} + \bar{S} * R + Q * \bar{S} + Q * R = \bar{S} * (1 + R + Q) + Q * R =$$

$$= \bar{S} + QR$$

Aplicando Morgan

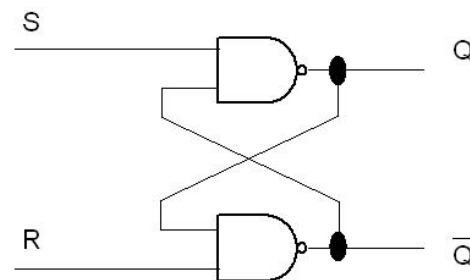
$$Q_{t+1} = \bar{S} + Q * R = \bar{S} * Q * R$$

$$Q_{(t+1)} = (S + \bar{R}) * (\bar{Q} + \bar{R}) = S * \bar{Q} + S * \bar{R} + \bar{R} * \bar{Q} + \bar{R} * \bar{R} =$$

$$= S * \bar{Q} + \bar{R} * (S + \bar{Q} + 1) = S * \bar{Q} + \bar{R}$$

Aplicando Morgan

$$Q_{(t+1)} = \bar{S} * \bar{Q} + \bar{R} = \bar{S} * \bar{Q} * \bar{R}$$

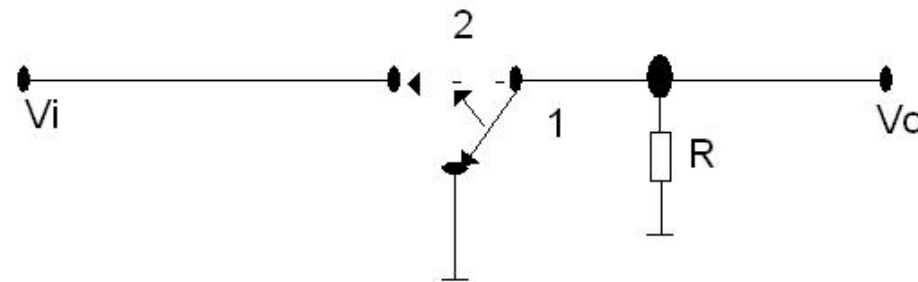


•Se puede ver que si $S=R=0$ simultáneamente ambas salidas son 1 lo cual no es lógico ya que una debe ser el complemento de la otra.

BIESTABLES ASINCRONOS

- APLICACIÓN

- Una aplicación práctica de las básculas RS es el diseño de pulsadores sin rebotes.

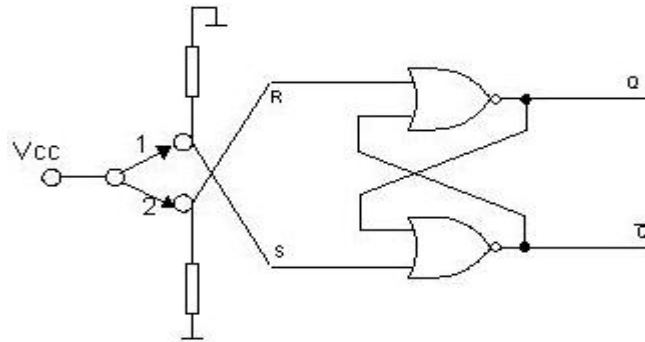


- Este circuito tiene dos problemas:
 - El primero es que la entrada se queda al aire mientras el interruptor está entre la posición 1 y la 2, por lo que la salida estaría indefinida
 - El segundo problema pueden ser los posibles rebotes que tenga el interruptor por su carácter mecánico.

BIESTABLES ASINCRONOS

- APLICACIÓN

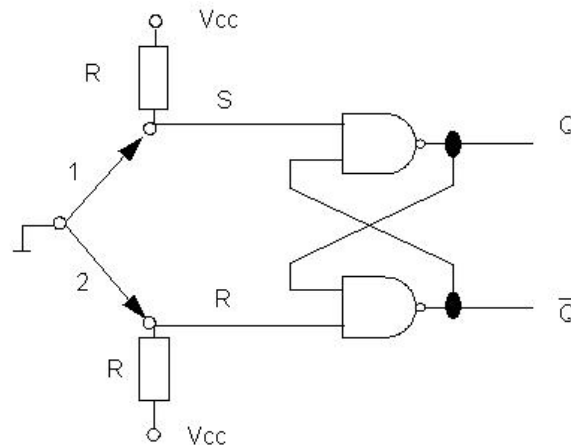
- Estos problemas se pueden resolver usando los biestables, por ejemplo con un latch RS-NOR.



- En principio el interruptor está colocado en la posición 1, por lo que las entradas del latch son $S = 1$ y $R = 0$ y las salidas $Q = 1$, en el camino de 1 a 2 tenemos las entradas $R = 0$ y $S = 0$, por lo que las salida no varía, y cuando llegamos al 2 tenemos $S = 0$ y $R = 1$, y las salidas $Q = 0$
- Con esto hemos conseguido que las entradas del latch nunca queden al aire, y a la vez, en caso de tener rebotes, durante los mismos, las entradas R y S están a cero, por lo que el biestable no varía.

BIESTABLES ASINCRONOS

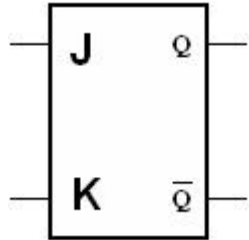
- SOLUCIÓN CON RS-NAND



- En principio tengo el interruptor colocado en la posición 1, por lo que las entradas del latch son $S = 0$ y $R = 1$ y las salidas $Q = 1$, en el camino de 1 a 2 tenemos las entradas $R = 1$ y $S = 1$, por lo que las salida no varían, y cuando llegamos al 2 tenemos $S = 1$ y $R = 0$, y las salidas $Q = 0$
- Con esto hemos conseguido que las entradas del latch nunca queden al aire, y a la vez, en caso de tener rebotes, durante los mismos, las entradas R y S están a uno, por lo que el biestable no varía.

BIESTABLES ASINCRONOS

- LATCH J-K



- Un nuevo tipo de latch, muy similar a los RS, pero que solucionan los problemas que presentaban estos últimos al activar simultáneamente sus entradas, son los biestables JK asíncronos.

- En los biestables JK, la J hace las funciones del SET, y la K hace las funciones del reset .
- El funcionamiento es similar al expuesto para las básculas RS-NOR, con la diferencia de que si llegamos al estado ($J = K = 1$), el latch conmutará al estado opuesto al que se encontraba previamente, es decir, si previamente Q era 1 conmutará a cero y viceversa.

BIESTABLES ASINCRONOS

- LATCH J-K

- Resumiendo el funcionamiento del Latch JK asíncrono en tablas de verdad:

<u>J</u>	<u>K</u>	<u>Q(t)</u>	<u>Q(t+1)</u>
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

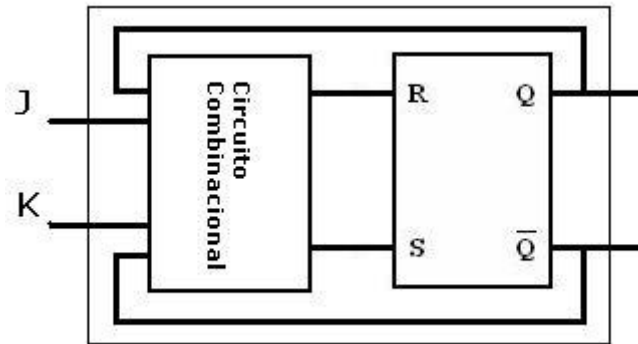


<u>J</u>	<u>K</u>	<u>Q(t+1)</u>
0	0	Q(t)
0	1	0
1	0	1
1	1	$\overline{Q(t)}$

- La tabla es muy similar a la del latch RS-NOR Asíncrono, con la activación de entrada mediante unos.
- El estado R=S=1, se sustituye por J=K=1 y se le asigna la salida inversa a la actual.

BIESTABLES ASINCRONOS

- BIESTABLE J-K Para el diseño del biestable JK nos basaremos en el RS ya diseñado anteriormente y añadiendo a su entrada una etapa combinacional:



BIESTABLES ASINCRONOS

- Para el diseño del C. Combinacional plantearemos la siguiente tabla:

J	K	Q_t	Q_{t+1}	S	R
0	0	0	0	0	X
0	0	1	1	X	0
0	1	0	0	0	X
0	1	1	0	0	1
1	0	0	1	1	0
1	0	1	1	X	0
1	1	0	1	1	0
1	1	1	0	0	1

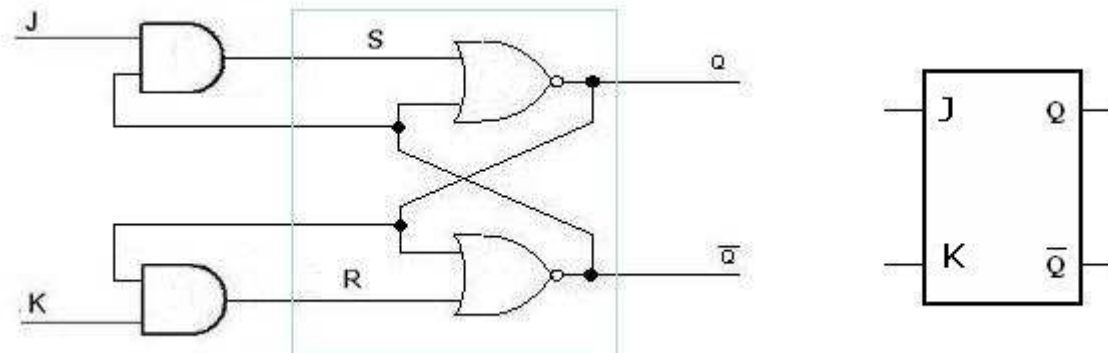
Si $J=K=Q_{(t)}=0$ implica que $Q_{(t+1)}=0$, es decir, mirando la entrada del latch RS, si lo tenemos apagado y queremos que continúe apagado, debemos asegurar que $R=0$ y "no importa" lo que valga S, con este mismo razonamiento rellenamos el resto de la tabla recordando que $S=R=1$ es un estado prohibido.

Haciendo los mapas de Karnaugh obtendremos:

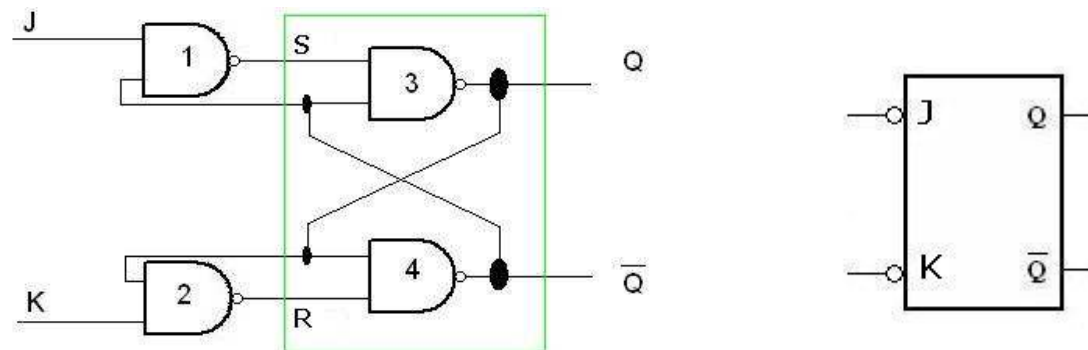
$$S = J * \overline{Q}$$

$$R = K * Q$$

BIESTABLES ASINCRONOS



- Si se plantea el mismo problema partiendo del latch RS-NAND solamente debemos negar las salidas de la parte combinacional :



$$S = J \cdot \bar{Q}$$

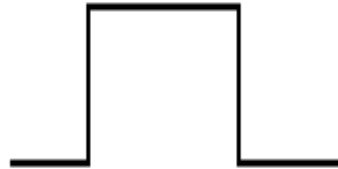
$$R = K \cdot Q$$

BIESTABLES SINCRONOS

- Hasta el momento todos los biestables estudiados eran síncronos, es decir, su salida cambia en el momento que cambie su entrada siempre y cuando la nueva combinación de entrada provoque estado lógico diferente a la salida.
- Pero es habitual que nos interese decidir nosotros en que momento se debe producir dicho cambio a la salida, para ello añadiremos a los biestables vistos unas entradas de sincronismos a través de las cuales se decidirá el momento exacto en que la salida debe reflejar lo que la entrada le pide, y no antes ni después (*siempre teniendo en cuenta los retrasos de tiempos de propagación*)

BIESTABLES SINCRONOS

- El tipo de sincronismo puede ser:



Síncrono a nivel alto (Latch)

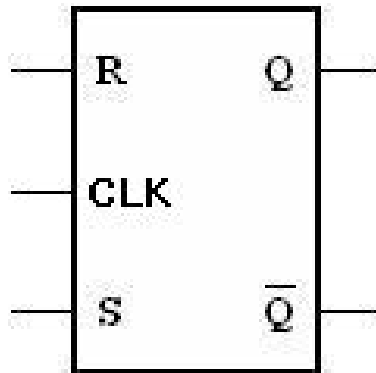
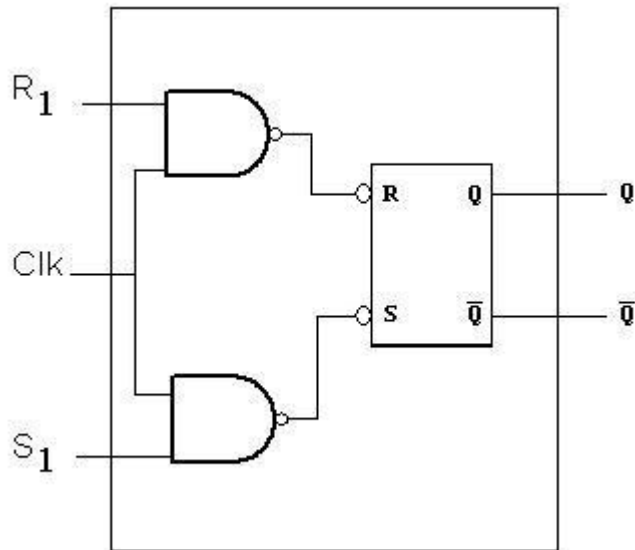
Síncrono a nivel bajo (Latch)

Síncrono por flanco de subida (Flip-Flop)

Síncrono por flanco de bajada (Flip-Flop)

- Cuando un circuito digital es activo por nivel alto, su salida sólo podrá modificarse cuando tengamos un nivel alto en la señal de sincronismo, análogamente podemos definir los circuitos síncronos por nivel bajo, por flanco de subida y por flanco de baja.
- Habitualmente se toma la señal de sincronismo como la correspondiente a una señal de reloj (CLK)

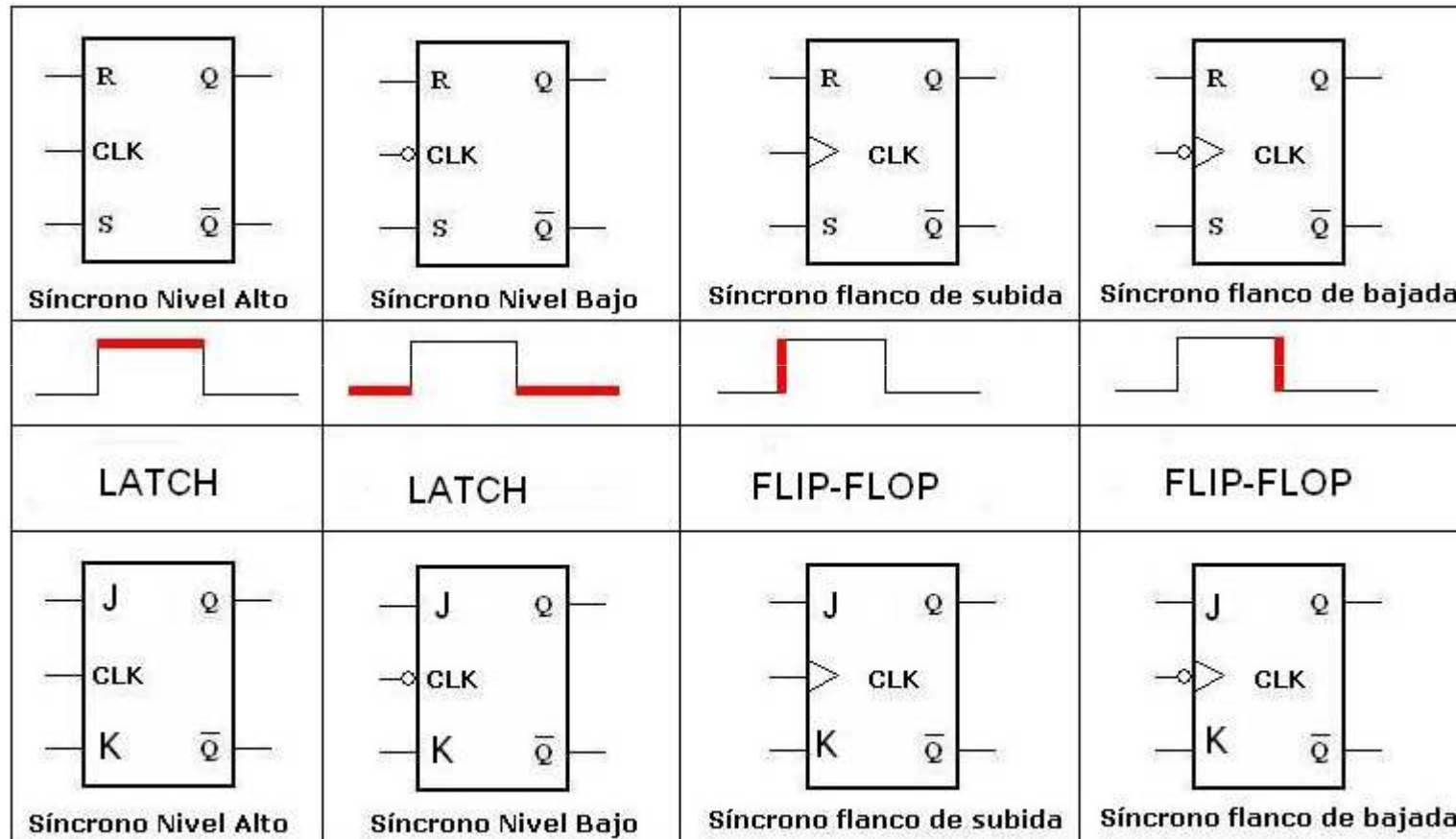
BIESTABLES SINCRONOS



- Como ejemplo vemos un latch RS-NAND al que se le ha añadido una entrada de sincronismo por nivel alto:
- Mientras la señal **CLK = 0**, hace que tanto la entrada R como la S del latch asíncrono sean 1, por lo que la salida no varía.
- Cuando **CLK = 1**,
 - $R_1=0$ y $S_1=0$ implica que $R=S=1$, el latch no varía.
 - $R_1=0$ y $S_1=1$ implica que $R=1$, $S=0$, el latch se enciende
 - $R_1=1$ y $S_1=0$ implica que $R=0$, $S=1$, el latch se apaga
 - $R_1=1$ y $S_1=1$ implica que $R=S=0$, prohibido

BIESTABLES SINCRONOS

- De la misma forma se puede hacer que el biestable sea síncrono a nivel bajo o incluso a uno de los flancos (ascendente o descendente) de la señal de reloj.



BIESTABLES SINCRONOS

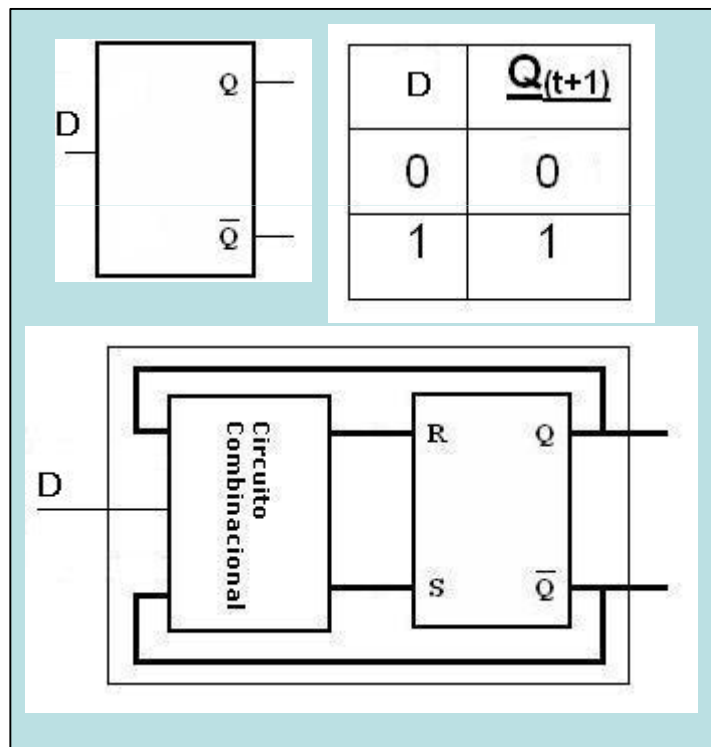
- **BIESTABLE D**

- Un tipo de biestable muy empleado es el biestable D, este tipo de biestable dispondrá de una entrada de datos “D”, por otro lado dispondremos de la entrada de sincronismo y de la salida.
- En los instantes de sincronismo, la salida del biestable D es transparente a su entrada. El resto del tiempo, la salida del biestable quedará fija o memorizada.

BIESTABLES SINCRONOS

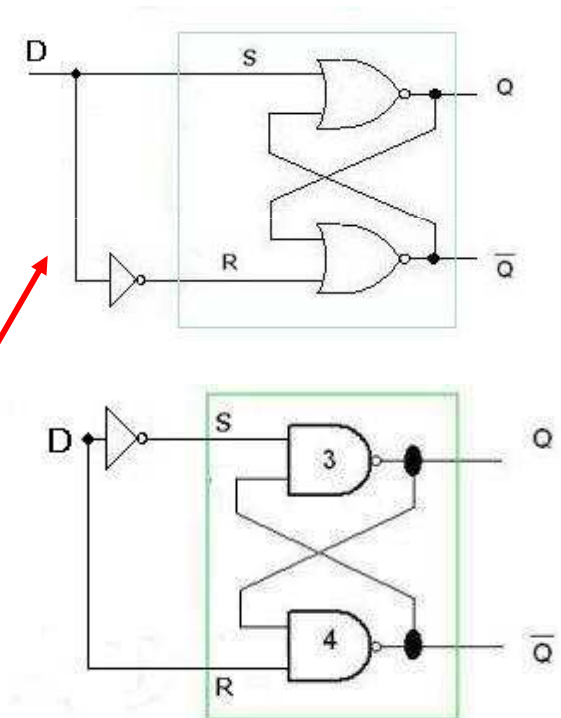
- LATCH D

- Como ejemplo realizaremos el diseño de un latch D, y comenzaremos diseñándolo de forma asíncrona e inmediatamente le añadiremos la entrada de sincronismo por nivel



D	Q_t	Q_{t+1}	S	R
0	0	0	0	X
0	1	0	0	1
1	0	1	1	0
1	1	1	X	0

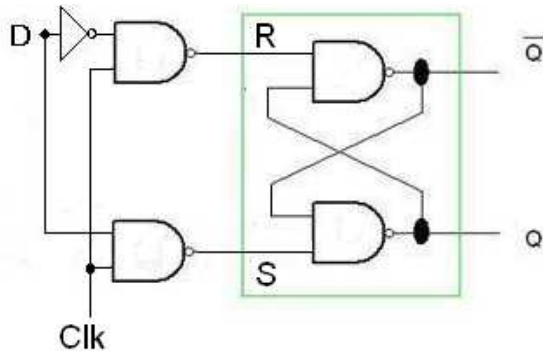
$$S = D$$
$$R = \bar{D}$$



BIESTABLES SINCRONOS

- LATCH D

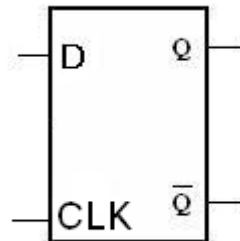
- Añadimos la entrada de sincronismo por nivel para crear el LATCH-D



- Si $\text{Clk} = 0$ implica que $R = S = 1$, luego $R = S = \text{inactivo}$, por lo que la salida queda como estaba antes.

- Si $\text{Clk} = 1$ y $D = 1$, implica que $S = 0$ y $R = 1$, luego $S = \text{Activo}$ y $R = \text{inactivo}$, por lo que la salida = 1

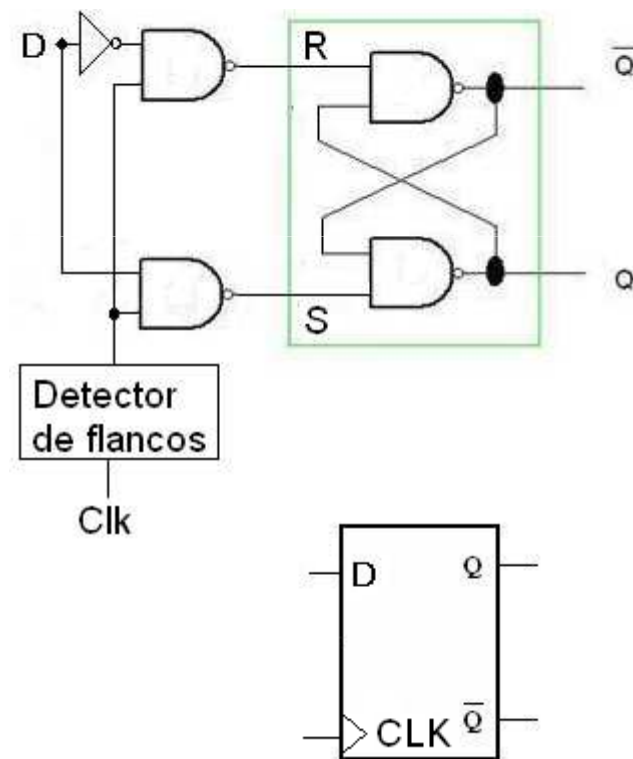
- Si $\text{Clk} = 1$ y $D = 0$, implica que $S = 1$ y $R = 0$, luego $S = \text{Activo}$ y $R = \text{inactivo}$, por lo que la salida = 0



BIESTABLES SINCRONOS

- Flip-Flop D

- Del mismo modo se puede implementar un flip-flop tipo D, pero para ello necesitaríamos un “Detector de flancos” en la entrada de sincronismo.

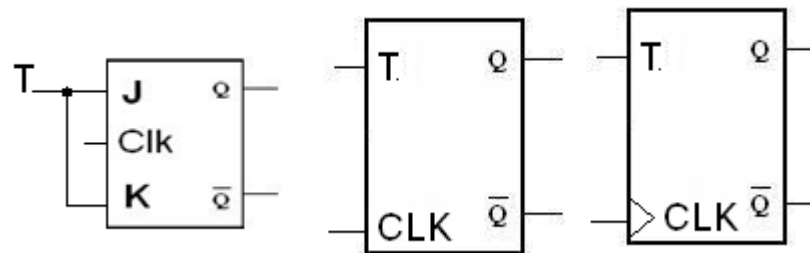


BIESTABLES SINCRONOS

- **BIESTABLE T**

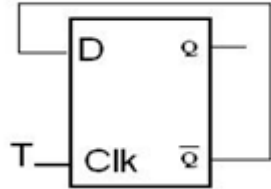
- Otro tipo de biestable es el tipo T, este tipo de biestable dispondrá de una entrada de datos “T”, por otro lado dispondremos de la entrada de sincronismo y de la salida.
- En los instantes de sincronismo, si la entrada $T = 1$, la salida cambiará de 0 a 1 ó de 1 a 0 dependiendo de cómo estuviera previamente, y si $T = 0$ la salida permanecerá sin variar su estado. El resto del tiempo, las salidas del biestable quedará fija o memorizada.
- La forma más sencilla de obtener este tipo de biestable es partiendo de un JK con sus entradas J y K unidas

<u>T</u>	<u>$Q_{(t+1)}$</u>
0	$Q_{(t)}$
1	$\overline{Q_{(t)}}$



BIESTABLES SINCRONOS

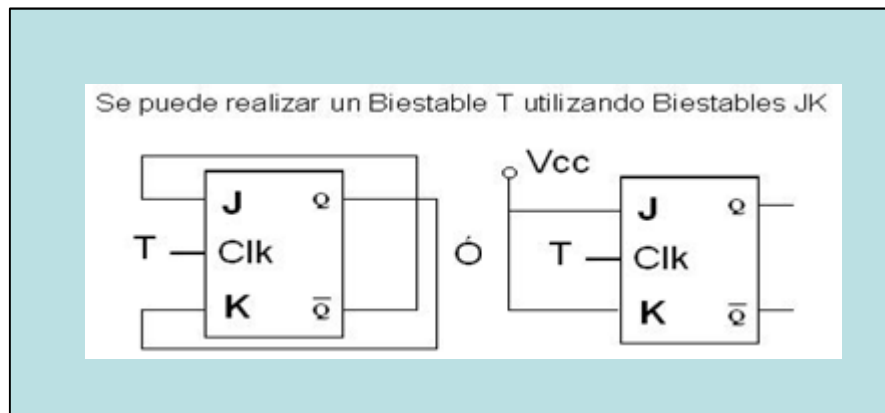
- Una forma sencilla de conseguir una Bascula tipo T con una de tipo D



Observar que el latch D es síncrono a nivel alto y en su entrada de sincronismo conectamos la entrada T

- Cuando $T = 0$, la entrada de sincronismo es cero y el latch D no varía su salida.
- Cuando $T = 1$, la entrada de sincronismo es uno y el latch D recoge la salida negada de Q.

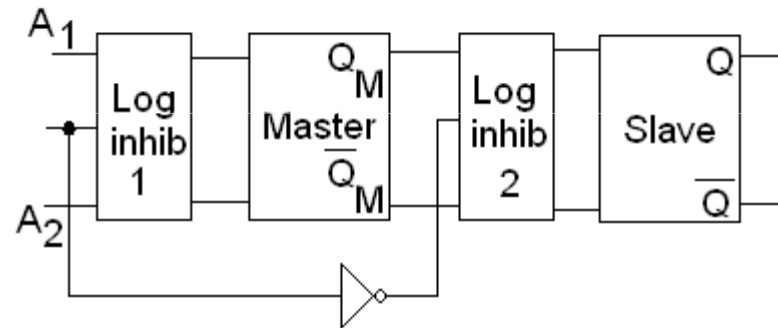
<u>T</u>	<u>$Q_{(t+1)}$</u>
0	$Q_{(t)}$
1	$\overline{Q_{(t)}}$



BIESTABLES SINCRONOS

• Biestables Master-Slave (Maestro-esclavo)

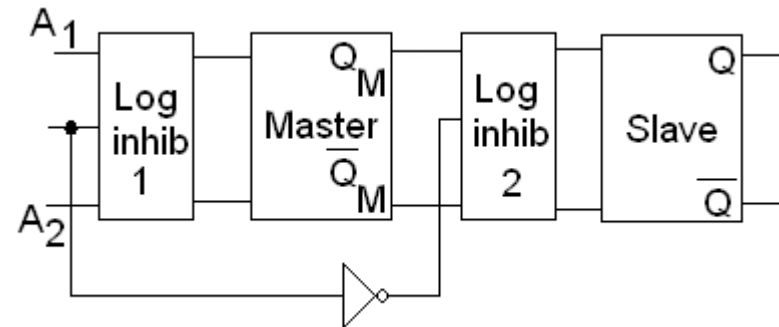
- En este tipo de biestables, se toman datos durante un nivel o flanco de la señal de sincronismo y se llevan a la salida en otro flanco de la misma señal y se denominan flip-flop



El circuito consta de dos biestables (Maestro y Esclavo) y dos circuitos lógicos de inhibición que se encuentran siempre en distinto estado

BIESTABLES SINCRONOS

FLIP-FLOP M/S POR FLANCO NEGATIVO

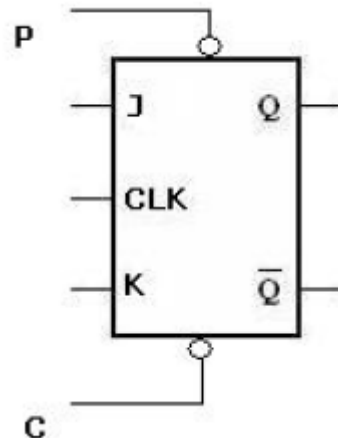


- Con la señal de reloj a nivel alto, el circuito lógico de inhibición 1 permite que la información presente en A1, A2, actúe sobre el biestable maestro. En cambio, el circuito lógico de inhibición 2 impide que la información del maestro actúe sobre el esclavo que seguirá memorizando su último estado.
- Cuando la señal de reloj pasa de nivel alto a nivel bajo, el circuito lógico de inhibición 1 impide que la información presente en A1, A2, actúe sobre el biestable maestro. En cambio, el circuito lógico de inhibición 2 permanece transparente, permitiendo que la información del maestro actúe sobre el esclavo y sea la salida del biestable M/S.

BIESTABLES SINCRONOS

ENTRADAS ASINCRONAS EN BIESTABLES SINCRONOS

- Con el fin de construir circuitos más flexibles, es normal que además de las entradas síncronas, es decir, controladas por el reloj, existan a la vez otras entradas independientes que actúan de forma asíncrona y que generalmente se les denomina PRESET (Puesta a uno) y CLEAR (Puesta a Cero)



Parámetros de los Biestables

- **Tiempo de asentamiento (“ Set Up Time o setting time”)**: Es el tiempo previo al flanco activo de toma de datos durante el cual las entradas no deben cambiar para que los biestables las interpreten de forma correcta.
- **Tiempo de mantenimiento (“ Hold time o release time “)**: Es el tiempo posterior al flanco activo de toma de datos durante el cual las entradas no deben cambiar.
- **Frecuencia máxima de reloj (“ f max “)**: Es la máxima frecuencia de reloj a la que el biestable responde de forma correcta.
- **Anchura mínima de pulso de reloj en alto (twidth(H))**: Es el mínimo ancho de señal de reloj a nivel alto que puede captar el biestable.
- **Anchura mínima de pulso de reloj en bajo (twidth(L))**: Es el mínimo ancho de señal de reloj a nivel bajo que puede captar el biestable.
- **Tiempo de recuperación (trec)**: Tiempo mínimo que debe pasar desde que las entradas asíncronas dejan de ser activas hasta el flanco activo de la señal de reloj.

Parámetros de los Biestables

Tiempo alto/bajo de preset/Clear : Es el tiempo mínimo que deben permanecer activadas las entradas asíncronas para garantizar un correcto funcionamiento.

Tiempo de propagación (tpHL, tpLH): Son los tiempos que transcurren desde que una entrada actúa hasta que se produce cambio en la salida, tp_{HL} será el tiempo que transcurre desde que una entrada pasa de alto a bajo hasta que se produce el cambio en la salida, y tp_{LH} será el tiempo que transcurre desde que una entrada pasa de bajo a alto hasta que se produce el cambio en la salida.