

Apuntes de Clases

Flip-Flops.

Realizado por Sergio Noriega

Introducción a los Sistemas Lógicos y Digitales
Departamento de Electrotécnica
Facultad de Ingeniería
Universidad Nacional de La Plata
2003

INDICE

- 1 - Introducción.
- 2 - Flip-Flops asincrónicos (latches):
 - Biestable elemental.
 - El flip-flop RS:
 - Implementación con NAND y NOR.
 - Análisis por Karnaugh.
 - Limitaciones.
- 3 - Flip-Flops sincrónicos:
 - Flip-flops disparados por nivel:
 - Estructura simple y master-slave.
 - El flip-flop tipo RS.
 - El flip-flop tipo JK.
 - Flip-flops derivados: tipo T y D.
 - Limitaciones.
 - Flip-flops Disparados por flanco:
 - El flip-flop tipo JK.
 - El latch (flip-flop tipo D).
- 4 - Bibliografía.

1 - Introducción.

El empleo de circuitos combinatorios, es decir, aquellos circuitos cuyas salidas solo dependen del estado de sus entradas, tienen limitaciones a la hora de poder implementar ciertas operaciones.

Tanto las compuertas básicas (inversor, NAND, AND, NOR, OR, OR-Exclusivo, NOR-Exclusivo, etc.), como circuitos formados a base de ellas: sumadores, restadores, MUXs, DeMUXs, Decodificadores, Codificadores, etc. adolecen de la falta de capacidad para memorizar un nivel lógico.

Por ejemplo, supongamos que con lo que sabemos hasta ahora deseamos implementar la lógica electrónica para comandar un ascensor en un edificio de tres pisos.

En principio, el control del motor del ascensor debe ser capaz de arrancarlo, pararlo y hacer que éste suba ó baje.

Esto lo podemos hacer con un circuito de sólo dos salidas: una para activar/desactivar al motor y otra para que éste gire en un sentido o en el otro (subir ó bajar).

De ahí en mas, deberemos manejar con etapas de potencia los contactores del circuito eléctrico del motor.

Con respecto a las entradas, debemos tener por un lado indicación de donde está el ascensor, para saber si se debe mover al llamarlo y en caso afirmativo saber si debe bajar ó subir y donde detenerse.

Esto se puede implementar con fines de carrera dispuestos uno en cada piso.

Además faltan los pulsadores de llamada del ascensor en cada piso.

A manera de simplificar el problema nos centraremos sólo en como implementar la lógica de llamada del ascensor.

Con lo que sabemos hasta ahora, podemos diseñar un circuito que al poner un "1" lógico cuando la llave de un piso se presione, active al motor, si es que no está en ese lugar (lo cual se puede saber por los contactos de los fines de carrera).

El mismo se deberá detener cuando una guía del ascensor toque el fin de carrera del piso donde se presionó la llave.

El primer problema que tenemos es que no es práctico emplear una llave que tengamos que conectar cuando llamamos al ascensor y luego desconectarla cuando haya llegado, a fin de permitir que sea llamado desde otro piso.

Lo que hace falta es un pulsador, donde bastaría sólo conectarlo una vez para llamar al ascensor y éste quede liberado para un posterior uso.

Pero esa conexión breve, debería poner la entrada del circuito de accionamiento del ascensor del piso desde donde se lo llama, en "1" hasta que el ascensor llegue, y luego al arribar éste, poner dicha entrada a "0" otra vez.

Esta operación se debe realizar con un circuito cuya entrada proviene del pulsador y su salida al circuito principal de accionamiento del ascensor.

Dicho circuito debe estar en "0" y al recibir un "1" al conectarse brevemente el pulsador, ponerse en "1" y quedarse en ese estado aunque la entrada vuelva a "0" (pulsador liberado). Deberá quedar así en "1", hasta que con otra entrada proveniente de un fin de carrera, sea puesta a "0" nuevamente.

Con lógica combinatoria no es posible implementar tal función ya que una tabla de verdad dictaría que si hay un "1" desde el pulsador debe haber un "1" en la salida y viceversa, cuando hay un "0", la salida debe estar en "0".

Pregunta: Cómo hacer para que el circuito una vez que haya recibido un "1" del pulsador conectado, quede en "1" aunque el pulsador se vaya a "0" lógico después?.

Respuesta: Se necesita de un circuito con memoria.

En este caso, la salida no sólo debe tener en cuenta el nivel lógico de la entrada, sino además el nivel lógico de la salida, o sea, debe haber una realimentación.

En la figura 1 vemos como se puede implementar esto:

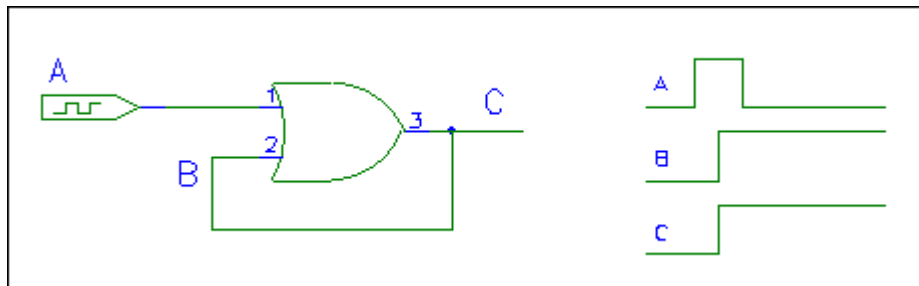


Figura 1

Vemos una compuerta OR donde una de las entradas proviene del pulsador (entrada A), mientras que la otra (B) es una realimentación de la salida C.

Suponiendo $A = "0"$, es decir, pulsador no presionado, y salida también en "0", tendremos que cuando A vaya a "1" (llamada al ascensor), por la tabla de verdad de la compuerta OR, C pasará a "1", quedándose en ese nivel lógico aunque A pase a "0" cuando se deje de oprimir el pulsador, debido a que la salida C está conectada a la entrada B, y ese "1" forzará a la OR a quedarse en nivel alto.

Tenemos así, un circuito que ha memorizado esa entrada de "1" de A, a través de la realimentación efectuada desde su salida.

Obviamente el circuito que se necesita para implementar toda la operación es mas complejo ya que deberíamos tener otra entrada proveniente del fin de carrera del piso respectivo para volver a poner a "0" a dicha salida.

La función de memorizar una operación lógica es la base de los llamados **circuitos secuenciales**, los cuales son imprescindibles para poder implementar dispositivos mas complejos como contadores, microprocesadores, memorias, etc.

Circuito secuencial: Es aquél que posee la habilidad de memorizar estados. Su arquitectura implica que las salidas del mismo no sólo dependen de las entradas sino también de estados anteriores de dichas salidas.

A partir de ahora veremos los circuitos básicos denominados flip-flops, los cuales son componentes necesarios para generar otros mas elaborados.

2 - Flip-Flop asincrónicos (latch):

El Biestable elemental.

Un elemento de memoria es algo que mantiene su estado durante largo tiempo, una vez que éste ha sido cambiado por un estímulo externo aún cuando éste haya desaparecido.

Supongamos que tenemos una compuerta **no inversora**, es decir, un buffer, entonces, según su tabla de verdad, si tengo un "1" en la entrada, la salida estará en "1" y viceversa.

Si ahora uno la salida con la entrada tendré el denominado **circuito biestable** ó elemento fundamental de memoria (figura 2).

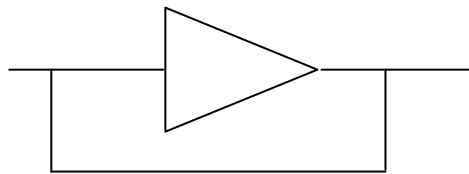


Figura 2

Físicamente a un "0" lógico le corresponde un rango de valores de tensión, mientras que a un "1" otro rango.

Por ejemplo, en forma ideal, un **no inversor** alimentado con una tensión de +5V, tendrá un rango de tensiones de entrada donde si la misma varía entre 0 y 2,5V , la salida será "0", mientras que si la entrada varía desde 2,5V a +5V, la salida irá a "1".

Tenemos entonces, un entorno infinitesimal alrededor de 2,5V, donde un valor que apenas supere ese umbral conmutará a la compuerta a "1" y viveversa.

El problema técnico es como definirle un estado u otro a la entrada.

En la siguiente figura vemos como podemos mejorar el anterior circuito para "**setear**" (poner a "1" lógico) ó **resetear** (poner a "0" lógico) la salida del biestable.

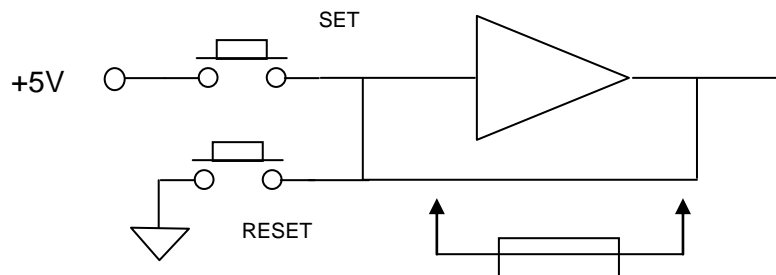


Figura 3

En este caso estoy momentáneamente forzando a la entrada, a la tensión +5V ó a masa, si presiono el pulsador de "SET" ó "RESET" respectivamente.

Existe un problema físico, y es que estoy temporalmente poniendo también a +5V ó a masa a la salida, pudiéndose dañar la compuerta internamente.

Una mejora sería conectar una resistencia entre la entrada y la salida como se indica en la misma figura, a fin de limitar la corriente que entrega la misma. Esta opción no es recomendable ya que generaría una caída de tensión en la resistencia y por lo tanto habría diferencia de tensión entre la entrada y la salida, lo cual puede hacer funcionar mal a la compuerta buffer por no tener los niveles de tensión correctos para interpretar lo que es un "1" ó un "0" (esto se verá con mas detalles en el capítulo de familias lógicas).

El Flip-Flop /S /R

Continuando con la idea de perfeccionar el circuito para que éste pueda almacenar un estado definido a través de estímulos externos, tenemos el circuito de la figura 4.

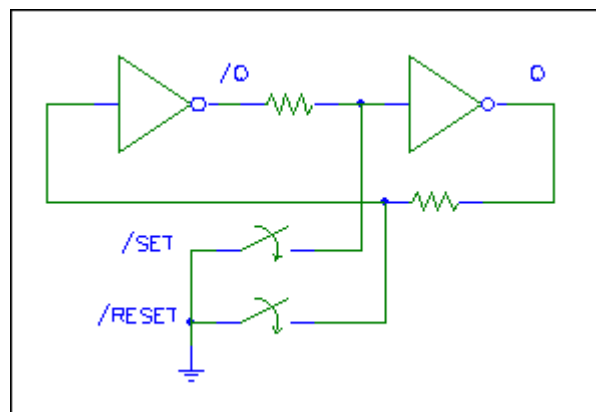


Figura 4

Aquí hemos reemplazado al no inversor por dos inversores en cascada, tal que la salida Q, sigue siguiendo el nivel lógico de la entrada.

Las resistencias se ponen para evitar cortocircuitos entre entradas y salidas.

Dos pulsadores /SET ó /PRESET y /RESET ó /CLEAR, tienen la función de programar la memoria.

Cuando estos están liberados, el biestable mantiene el estado anterior a su salida.

Si /SET está en "0" (**la barra "/" indica que se activa en nivel bajo**), fuerza un "1" a la salida del segundo inversor que es la salida del biestable.

Dado que hay una realimentación entre esta salida y la entrada del primer inversor, el circuito queda en un estado estable, con la salida en "1".

Si /RESET se pone a "0", fuerza a "1" la salida de la primera compuerta y como sigue la cascada con la segunda, la salida será "0".

Otra vez la realimentación logra que el estado sea estable, quedando en "0" la salida Q.

Si llego a activar los dos pulsadores simultáneamente, estoy ante una situación donde la salida se pone a "1", quedando la segunda resistencia en paralelo con la salida general.

Una mejora que se puede realizar para evitar que queden las resistencias en serie con las compuertas, las cuales degradan los niveles lógicos, es reestructurar el circuito anterior y replantearlo según se puede ver en la figura 5:

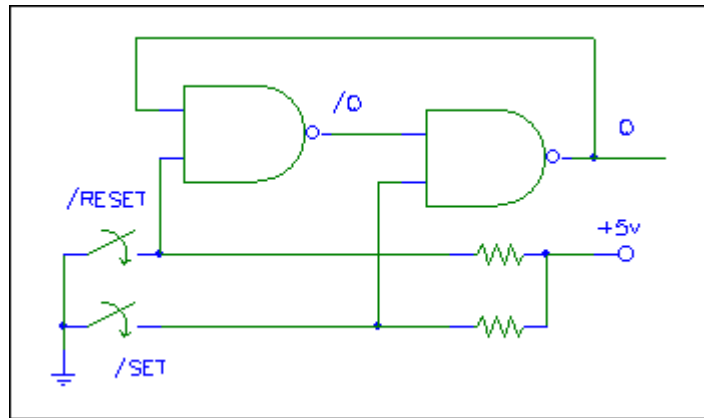


Figura 5

Las compuertas NAND reemplazan a los inversores, donde se indica el camino de la realimentación entre entrada y salida.

Con este arreglo es posible ahora aislar los pulsadores de las salidas evitando que queden las resistencias en serie.

A fin de no cortocircuitar la fuente de +5V a masa se ponen entre un pulsador y la línea de alimentación una resistencia de **"pull-up"** (tirar para arriba), la cual cuando no está presionado el mismo, lleva la tensión del punto medio a +5V.

Este circuito se puede reordenar, quedando como se muestra en la figura 6:

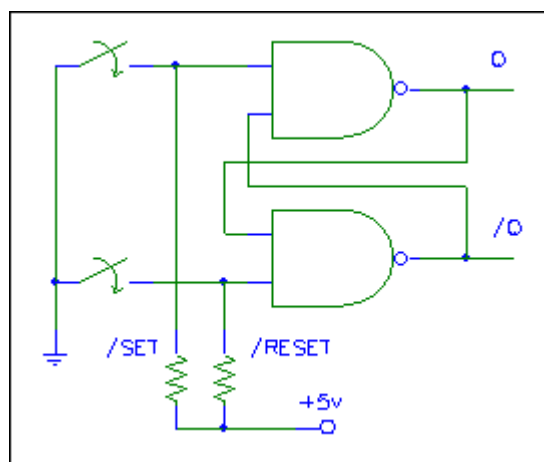


Figura 6

Si /S (/SET) está en "0" y /R está en "1", se fuerza la salida a "1".
Si /R (/RESET) está en "0" y /S queda en "1" la salida irá a "0".

Este circuito constituye una celda elemental de memoria práctica llamada **Flip-Flop** ó **Latch** (cerrojo).

Comunmente se le conoce como **Flip-Flop /S /R**, donde la barra "/" indica, como dijimos anteriormente, que la entrada se activa en nivel bajo ó "0".

Si quisieramos implementar esta misma función de memoria pero ahora con los pulsadores conectados a +5V en vez de a masa, tendríamos el siguiente circuito implementado con inversores (figura 7) y su posterior modificación con compuertas NOR (figura 8):

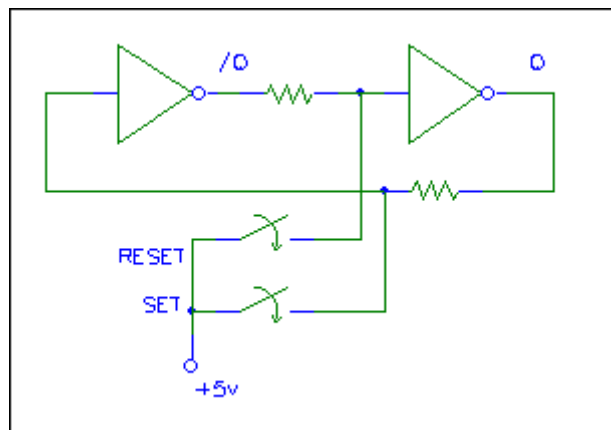


Figura 7

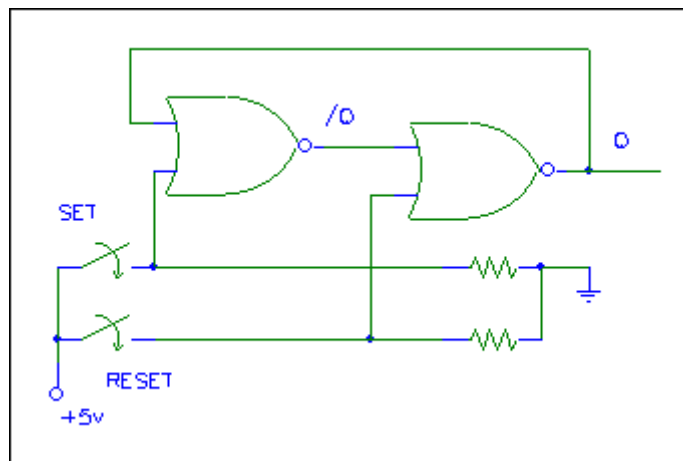


Figura 8

Aquí las resistencias que se agregan son de **"pull-down"** (tirar para abajo).
Los pulsadores están invertidos y además se activan con niveles opuestos al circuito implementado con compuertas NAND.

Dicho Flip-Flop ó Latch se denomina "S R".

Cuando Set está en "1" y R en "0" la salida pasa a "1" y viceversa.

Si en cambio tanto R y S están en "0", es decir, ninguno está presionado, dichas entradas no modifican la salida de cada compuerta, por lo tanto el Flip-Flop mantiene su estado anterior. Esto es así ya que esas entradas desde los pulsadores, al estar en "0", no alteran a las respectivas compuertas NOR.

Lo mismo pasa en el ejemplo anterior con el latch /S /R, implementado con compuertas NAND, donde si no se presionaban los pulsadores, las entradas asociadas, quedaban en "1".

En la figura 9 vemos el circuito de la figura 8 reordenado.

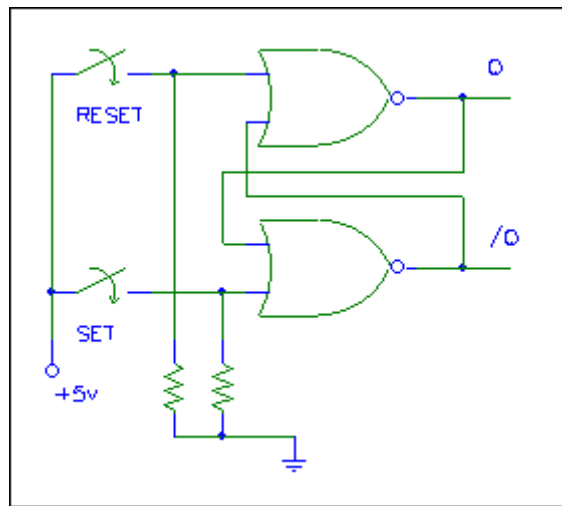


Figura 9

Análisis de funcionamiento por Karnaugh:

En la figura 10 tenemos nuevamente el circuito del Flip-Flop /S /R implementado con compuertas NAND, con el agregado de dos bloques τ_1 y τ_2 , los cuales representan los tiempos de respuesta que tienen las salidas de ambas compuertas ante un cambio en sus entradas.

Esto es debido a que todo dispositivo físico no puede responder ante una excitación en tiempo cero.

En otras palabras habrá un retardo temporal que representa esta inercia al cambio.

Por simplicidad del siguiente análisis los consideramos fuera del símbolo de las compuertas, ubicados en el camino de realimentación de cada una de ellas.

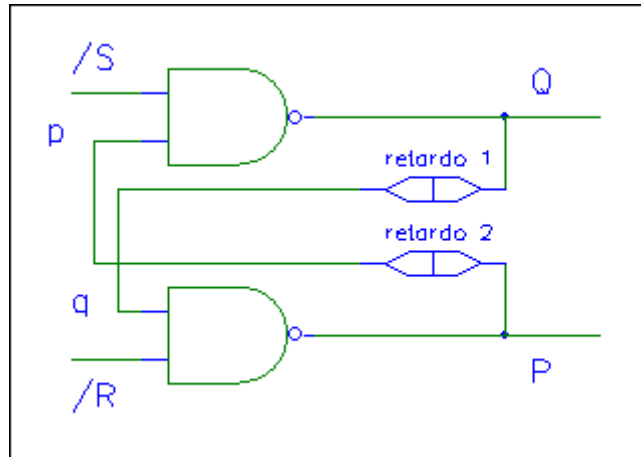


Figura 10

Para el análisis, debemos plantearnos todas las combinaciones posibles de estados de las entradas y salidas en un dado instante para saber como responderá el circuito.

Esto puede visualizarse en el siguiente diagrama de Karnaugh doble, donde indicamos con **/S** y **/R** las entradas /Set y /Reset, con **p** y **q** las salidas en un estado inicial (también entradas a las compuertas) y con **Q** y **P**, a las salidas futuras que resultan de aplicar la función Nand de las entradas anteriores (en una /S y p y en la otra /R y q).

El diagrama de Karnaugh definitivo es el siguiente, el cual analizaremos a continuación:

q p		/S /R			
		00	01	11	10
00	11	11	11	11	
01	11	11	01	01	
11	11	10	00	01	
10	11	10	10	11	

QP

Figura 11

NOTA: Las salidas q y Q por un lado y p y P por el otro, físicamente son las mismas, sólo que en distinto instante de tiempo, dado por los retardos τ_1 y τ_2 .

Para realizar el Karnaugh abrimos las realimentaciones.

Entonces, inicialmente con un juego de valores en las entradas y salidas vemos a que nuevo estado iremos.

Para analizar el funcionamiento del circuito luego cerramos las realimentaciones y vemos como irá comportándose las salidas a eventuales cambios en las variables de entrada.

Se denomina **estado estable** a aquella condición en donde se cumpla que $q = Q$ y $p = P$, es decir cuando las salidas queden en un nivel permanente sin modificarse, mientras se mantengan los niveles lógicos de las entradas.

Se denomina **estado inestable** a aquella condición en donde se cumpla que $q \neq Q$ y/o $p \neq P$, es decir cuando las salidas queden en un nivel transitorio, donde ese nuevo estado modificará de nuevo las salidas.

Los casilleros donde las salidas están marcadas con "**negritas**" y subrayadas corresponden a estados estables, es decir, luego de "inyectar" al circuito esa combinación de /R, /S, q y p, si no hay cambios en los niveles lógicos de las mismas, las salidas Q y P quedarán permanentemente en ese estado.

Por ejemplo, si suponemos que partimos de /S /R q p = 0011, las salidas no se modificarán, ya que un "0" en una de las entradas en una NAND nos dá siempre "1" a la salida de la misma sin importar el nivel lógico en la otra entrada.

Además partimos de que qp = 11, por lo tanto no hay cambios en Q y P, con lo que concluimos que corresponde a un estado estable.

En cambio si partimos con /S /R = 00 y qp = 00 ó 01 ó 10, habrá luego de un tiempo de retardo un cambio de la(s) salida(s) que esté(n) en "0" a "1", ya que será(n) forzada(s) por los "0" en las entradas.

Estos casilleros, donde tenemos /S /R = 00 y qp = 00 ó 01 ó 10 se les denomina estados inestables. Son estados de paso hasta llegar a un estado estable ó incluso a una situación de inestabilidad como veremos a continuación.

De esta manera, para el resto de las combinaciones posibles se va llenando el Karnaugh doble.

Consideramos a τ_1 y τ_2 de igual valor, a fin de examinar como funciona este circuito cuando se modifican las entradas /S y /R.

Lo que se intenta es buscar, en definitiva, una tabla de verdad que pueda resumir el funcionamiento del mismo.

Camino 1:

Suponemos que inicialmente tenemos /S /R = 11 y QP = 01.

Cambiamos /R tal que ahora tenemos /S /R = 10.

Esta nueva condición nos deja en el mismo estado ya que como q era "0" en la NAND inferior, el cambio de /R de "1" a "0" no modifica el "1" en P.

Por lo tanto ese cambio en /R no tiene efecto.

Si ahora volvemos a poner /R en "1", tampoco hay cambio en las salidas, donde volvemos en el Karnaugh a la celda inicial.

Los retardos τ_1 y τ_2 , en la realidad son diferentes y dependen del proceso de fabricación ya que es improbable que se puedan construir dos compuertas con los mismos valores de retardo.

Esto trae una condición particular en el funcionamiento del flip-flop, que es la siguiente:

Supongamos que la compuerta superior es mas veloz que la inferior, entonces:

$$\tau_1 < \tau_2$$

Camino 4:

Si por ejemplo partimos de la condición inicial donde /S y /R valen "0", las salidas Q y P valdrán "1" igual que p y q.

Si en el mismo instante ponemos ambas entradas a "1", cada NAND forzará sus salidas Q y P a "0", pero como la 1 es mas veloz que la 2, luego de τ_1 q pasará a "0" cambiando inmediatamente P a "1" la cual mantendrá a p todavía en "1" (se considera que no alcanza a reaccionar la compuerta 2 por ser mas lenta) lo que significa que Q seguirá en "0" manteniéndose un estado estable a partir de allí con QP = 01.

Camino 5:

Si en cambio es la compuerta 2 mas veloz que la 1, la salida estable será QP = 10.

Este análisis es simplista ya que aunque una compuerta reaccione tarde puede cambiar su salida.

En general lo que sucede es que tanto Q como P oscilarán varias veces entre un nivel lógico y otro hasta que se estabilicen las salidas como vimos.

De este último análisis se concluye que aunque la combinación inicial de /S /R= 00 genera un estado estable QP = 11, existe lo que se denomina "**carrera**" entre compuertas, cuyo resultado dependerá de la velocidad de respuesta relativa entre ambas.

El fabricante no puede garantizar chip por chip que compuerta es mas rápida. Por tal motivo salvo que se haga una medición, se considera esta combinación de entradas como **prohibida**.

Además aunque la condición QP = 11 sea estable, el flip-flop siempre se define con salidas Q y P donde Q = /P, es decir se tiene Q y /Q, dos salidas, una la negación de la otra.

Llamaremos entonces desde ahora /Q a P.

Si se hace un resumen del Karnaugh doble, se puede concluir que el funcionamiento del flip-flop /S /R implementado con compuertas NAND es el siguiente:

- 1 - Cuando las entradas /S /R están en "1", las salidas mantienen el estado anterior que tenían (Q /Q = 01 ó 10).
Esto dá al flip-flop la propiedad de poder **memorizar** un estado previo, que es lo que se buscaba.
- 2 - Cuando se fuerza a /S a "0" con /R en "1", las salidas pasan al estado estable Q /Q = 10.
- 3 - Cuando se fuerza a /R a "0" con /S en "1", las salidas pasan al estado estable Q /Q = 01.
- 4 - Inicialmente en /S /R = 00 si se pasa a /S /R = 11, los estados de las salidas pasarán a 01 ó 10 dependiendo de la diferencia de retardos entre ambas compuertas.

Con esto podemos escribir la siguiente tabla de verdad:

/S	/R	Q(n+1)	/Q(n+1)
0	0	Prohibido	Prohibido
0	1	1	0
1	0	0	1
1	1	Qn	/Qn

Figura 14

Donde:

Qn y /Qn es el estado actual de las salidas Q y /Q.

Q(n+1) y /Q(n+1) es el estado futuro de las salidas Q y /Q.

En la siguiente figura podemos ver un diagrama de tiempos donde se representan las entradas /S como DSTM3, /R como DSTM4, Q como U1A:Y y P como U1B:Y.

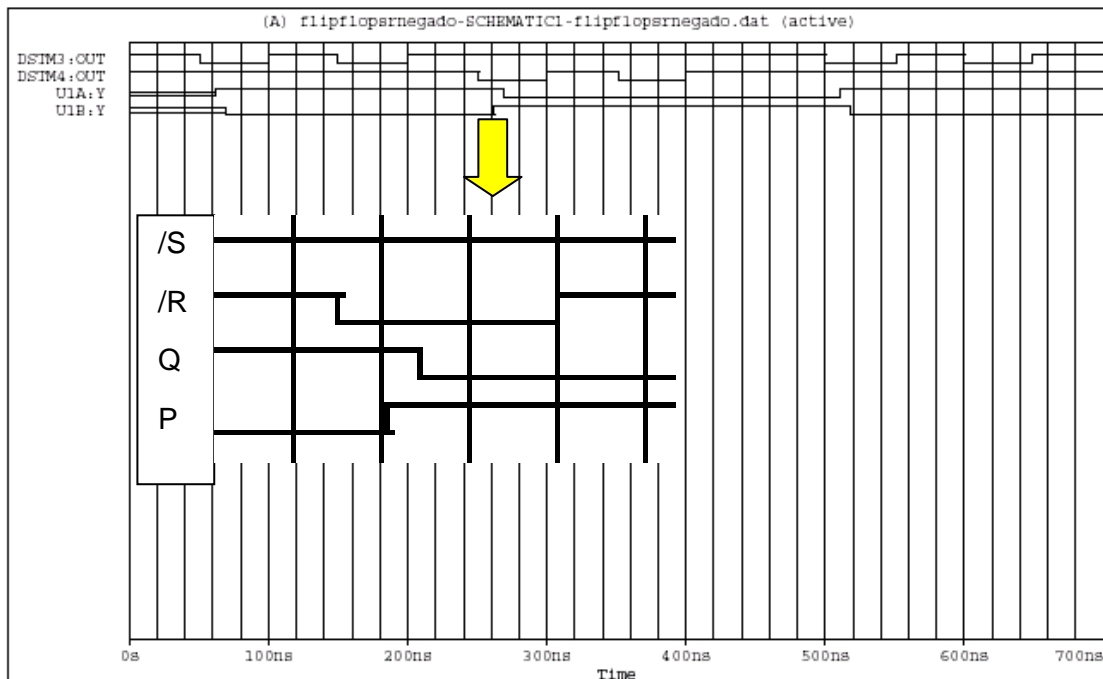


FIGURA 15

Esta forma de representación es otra alternativa para describir el funcionamiento de circuitos digitales.

Se puede notar como responden las compuertas con un cierto retardo. Por ejemplo al cambiar /S de "1" a "0", U1A:Y cambia antes que U1B:Y, siendo lo contrario cuando cambia /R de "1" a "0". Esto último se ha magnificado en dicha figura.

El Flip-Flop S R

Otra forma de implementar un flip-flop, es empleando compuertas tipo NOR. En la figura 16 vemos un ejemplo:

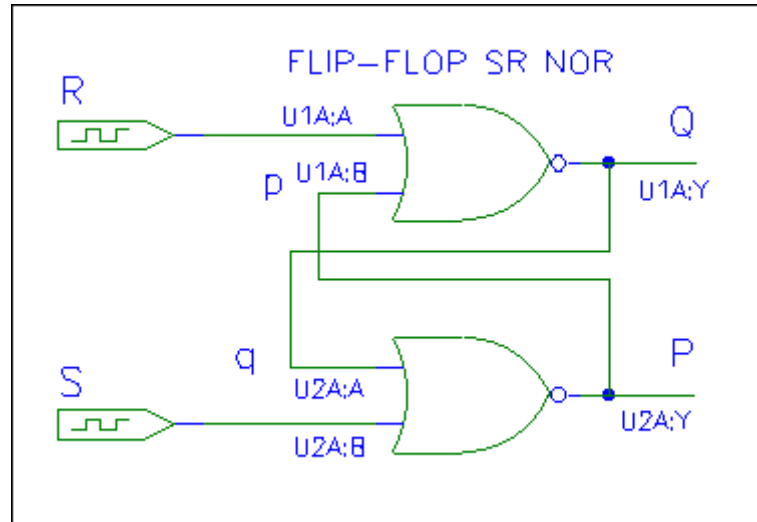


Figura 16

En este caso, las entradas son S y R, sin negar lo que significa que se activan con valor lógico "1", a diferencia del flip-flop /S /R, donde por ejemplo cuando /S = 0 la salida Q se ponía a "1" y cuando /R = 0 Q pasaba a "0".

Para analizar este circuito basta con aplicar el mismo criterio que con el anterior flip-flop.

Del mismo se obtiene la siguiente tabla de verdad:

S	R	Q(n+1)	/Q(n+1)
0	0	Q _n	/Q _n
0	1	0	1
1	0	1	0
1	1	Prohibido	Prohibido

Figura 17

Se puede observar que ahora la combinación SR = 11 se considera prohibida, ya que si se parte de la condición inicial SR = 11 y se pasa a la combinación SR = 00, las salidas Q y P de estar en QP = 00 pasarán a 01 ó 10 dependiendo de la diferencia relativa de retardos entre ambas compuertas.

Además ahora cuando S = 1 se fuerza a Q a valer "1" (acción de **set**), mientras que si R = 1 se tiene que la salida Q pasará a "0" (acción de **reset**).

Cuando ponemos SR = 00 tendremos que se memorizará el estado anterior.

Al igual que en el flip-flop precedente, esta combinación es la que le dá al circuito la característica de **memorizar** un estado previo.

En la figura 18 podemos ver un diagrama de tiempos donde se dan varias combinaciones de entradas R y S para ver como responden las salidas.

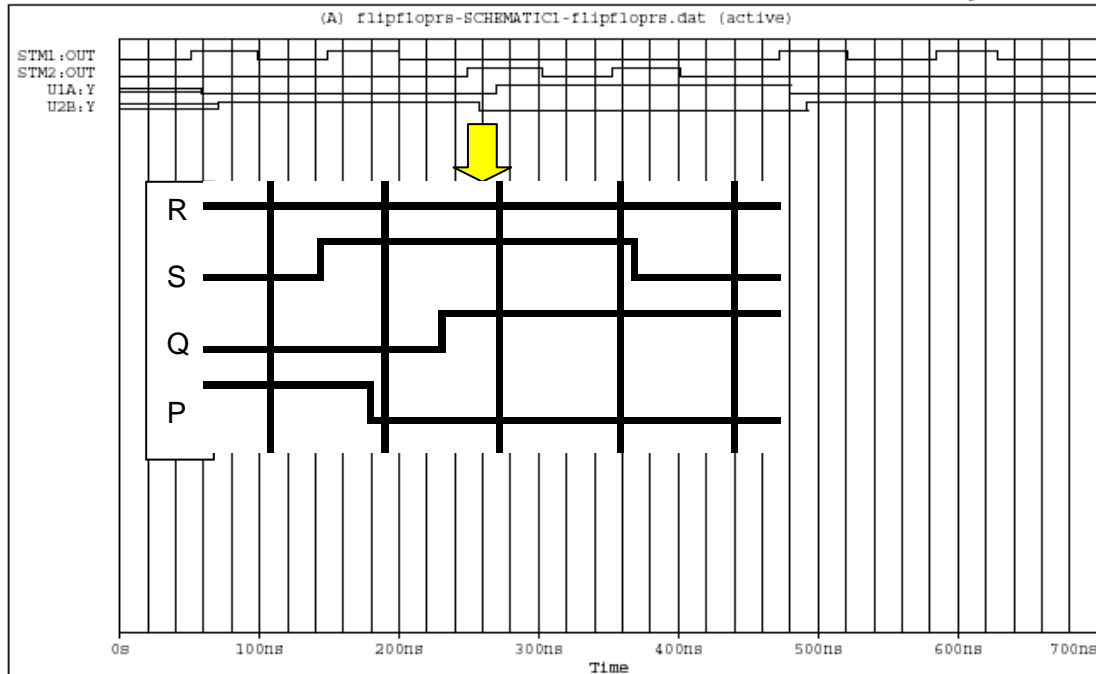


Figura 18

Ejemplo de aplicación: Pulsador antirebote.

Es sabido que cuando pulsamos una llave, el contacto al llegar al punto de cierre, rebota varias veces hasta que queda definitivamente solidario al terminal aislado.

Estos rebotes producen una serie de oscilaciones que se traducen en una ráfaga de "unos" y "ceros" al alentrada de algún circuito digital, pudiendo causar un mal funcionamiento ya que solo se espera una sola transición.

Vemos esto el la siguiente figura.

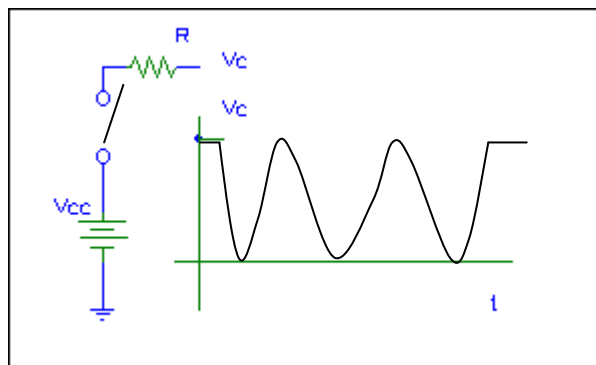


Figura 19

Una forma de solucionar esto es empleando un flip-flop, por ejemplo uno implementado con dos compuertas NAND, dos resistencias y una llave con punto medio (figura 20).

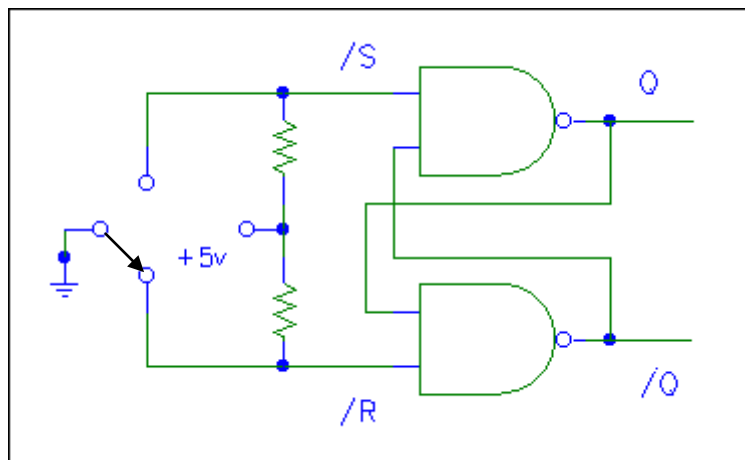


Figura 20

Se observa que el punto medio de la llave se conecta a masa (GND) mientras que el punto medio de las resistencias a V_{cc} , es decir, a un "1"lógico.

Inicialmente suponemos que con la llave en la posición inferior se tiene $\bar{S} = 1$ y $\bar{R} = 0$, por lo que la tabla de verdad nos indica que Q / \bar{Q} será el del estado anterior, que suponemos 01.

Cuando movemos la palanca de la llave hacia el otro borne, tendremos que ya no hará contacto con ninguno, por lo que las entradas serán $\bar{S} / \bar{R} = 11$ (se mantiene el estado anterior $Q = 0$).

Cuando se hace el primer contacto, se fuerza a \bar{S} a "1", mientras \bar{R} queda siempre en "0".

En ese momento y según la tabla de verdad, la salida pasará a $Q = 1$.

A partir de allí aunque la palanca rebote varias veces sobre el borne superior, las entradas cambiarán sucesivamente de $\bar{S} / \bar{R} = 11 \rightarrow 01 \rightarrow 11 \rightarrow 01 \rightarrow 11$ hasta que se amortiguen las oscilaciones y quede definitivamente en 01.

Por la tabla vemos que esas combinaciones de entrada no modificarán el estado después de haber hecho el primer contacto, es decir, la salida Q permanecerá en "1" hasta que se mueva la palanca hacia el borne inferior donde se repite el mismo mecanismo que terminará poniendo a Q otra vez en "0".

3 - Flip-Flop sincrónicos:

Lo visto hasta ahora comprende circuitos donde las salidas responden sólo a entradas externas de Set y Reset.

Hay aplicaciones en donde es necesario que los cambios que se efectúen en las salidas de un flip-flop, (debido a cambios en sus entradas de acuerdo a la tabla de verdad que regule el funcionamiento del mismo), estén sincronizadas con una señal maestra, a fin de poder realizar operaciones entre dos dispositivos ó mas, en forma ordenada para la transferencia de información ya sea de control como de datos.

Es por eso que existen los denominados flip-flops sincrónicos, los cuales además de sus entradas de datos (como por ejemplo: Set y Reset) tienen una entrada adicional de control, denominada clock (reloj).

Las salidas entonces, se modificarán si y solo si la combinación de las entradas de datos lo permite y si además el clock las habilita a generar dicho cambio.

En general esa entrada de clock en los flip-flops sincrónicos, activa a los mismos de una de dos formas posibles:

Por un nivel lógico de la señal de clock (nivel alto ó bajo)

Por un flanco de la señal de clock (flanco ascendente [positivo] ó descendente [negativo]).

Examinaremos primero el primer caso, es decir, a aquellos flip-flops que se activan por un nivel de la señal del clock.

Flip-flops activados por nivel

Una clasificación de flip-flops según su función es la siguiente:

Flip-flop RS

Flip-flop JK

Flip-flop D

Flip-flop T

Comenzaremos con el primero, es decir, el Flip-flop tipo RS.

Para poder implementar un flip-flop RS activado por el nivel de una señal de control, podemos partir del circuito de un flip-flop /S /R NAND asincrónico.

En la figura 22 vemos un circuito donde se indica el bloque que debemos diseñar para que se cumpla la siguiente tabla de verdad (figura 21):

S	R	Clk	Qn+1	/Qn+1
0	0	1	Qn	/Qn
0	1	1	0	1
1	0	1	1	0
1	1	1	Proh.	Proh.
x	x	0	Qn	/Qn

Figura 21

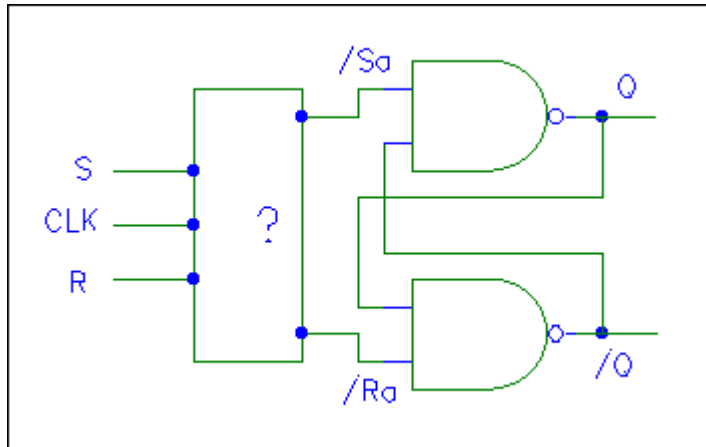


Figura 22

Según dicha tabla vemos que la misma es idéntica al flip-flop asincrónico RS sólo cuando la señal de clock es "1".

Si ésta es "0" las "x" indican que no importa que valores tomen S y R, las salidas del flip-flops mantendrán el estado anterior.

Para poder sintetizar el bloque en cuestión debemos por un lado tener en cuenta la tabla de verdad de la figura 21 y también la del flip-flop asincrónico del cual partimos, es decir, del /Sa /Ra implementado con NAND, en este caso.

Dicho bloque tendrá como entradas R, S y CLK y como salidas, /Sa y /Ra.

Podemos entonces realizar la siguiente tabla de verdad que describa como debe funcionar.

S	R	Clk	/Sa	/Ra
0	0	1	1	1
0	1	1	1	0
1	0	1	0	1
1	1	1	x	x
x	x	0	1	1

Figura 23

Por ejemplo: cuando CLK es "0" queremos que el flip-flop sincrónico mantenga el estado anterior, entonces la entrada del FF /S /R debe recibir "11".

Si queremos setear al FF (poner Q en "1") cuando S = "1", debemos poner /S /R = 01 y viceversa.

Para la combinación SR = 11 que se considera un estado prohibido, podemos poner /S /R = xx (indeterminado ó don't care) ya que se supone no se va a presentar dicha combinación, pudiendo de este modo simplificar las funciones de /Sa y /Ra.

Con la tabla ya armada, solo falta sintetizar las funciones de /Sa y /Ra.
 Los Karnaugh quedarán:

Clk	SR			
	00	01	11	10
0	1	1	1	1
1	1	1	x	0

/Sa

Figura 24

Clk	SR			
	00	01	11	10
0	1	1	1	1
1	1	0	x	1

/Ra

Figura 25

De las figuras 24 Y 25 tenemos:

$$/Sa = /CLK + /S = \overline{CLK S}$$

$$/Ra = /CLK + /R = \overline{CLK R}$$

Por lo tanto se puede implementar el bloque con compuertas tipo NAND, quedando el circuito del flip-flop sincrónico RS de la siguiente manera:

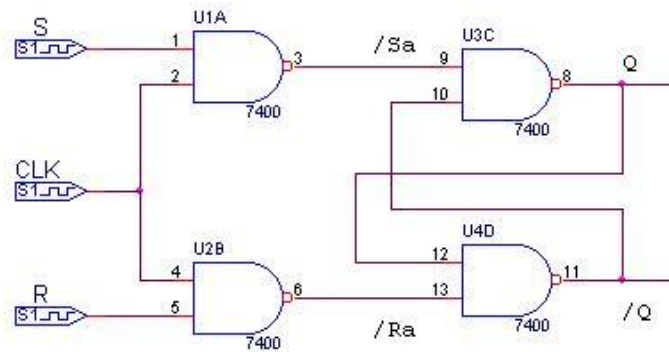


Figura 26

Se observa que por ejemplo cuando el CLK vale "0" las entradas /Sa y /Ra quedan en "11" lo que significa "mantener el estado anterior de las salidas".

Cuando tenemos SR = 10 en /Sa /Ra vale 01, poniendo Q en "1" y viceversa, cuando SR = 01, tenemos /Sa /Ra = 10, forzando a Q a ir a "0".

En la siguiente figura vemos un diagrama temporal el cual representa el funcionamiento de este flip-flop.

En el diagrama se presentan varias combinaciones de valores de S y R, donde se aprecia que cuando el CLK es "0", dichas entradas no tienen efecto sobre las salidas.

Las variables graficadas corresponden a:

U3C:Y es la salida Q. U4D:Y es la salida /Q

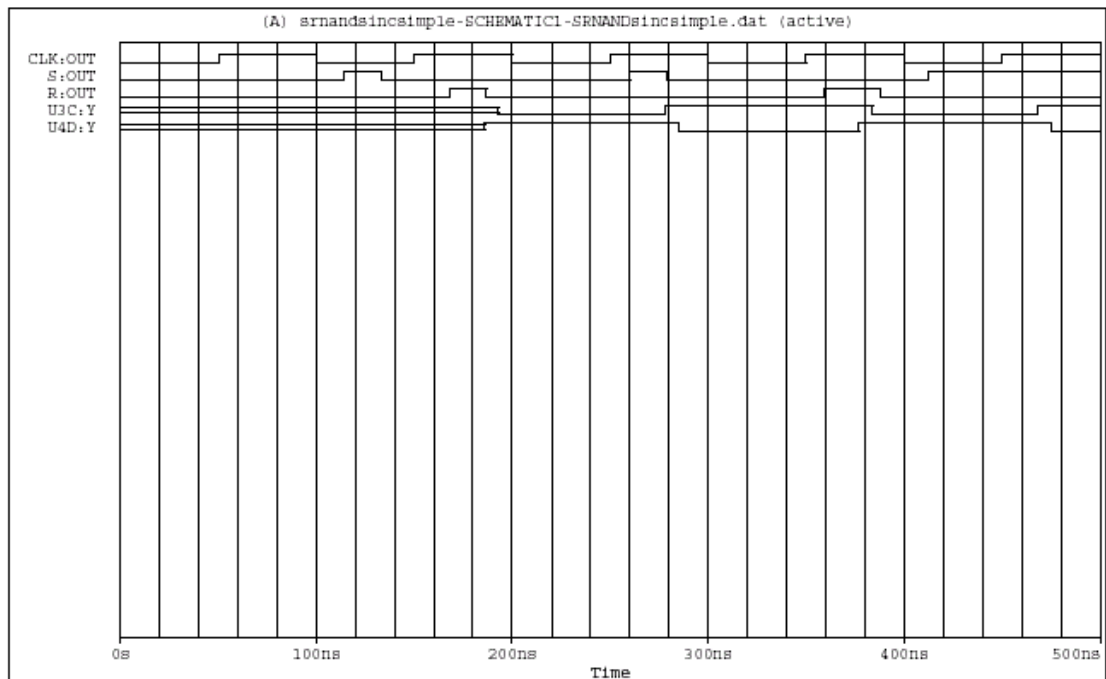


Figura 27

Flip-Flop sincrónico JK disparado por nivel

Un tipo de flip-flop muy empleado es el denominado flip-flop tipo "JK". Este presenta la siguiente tabla de verdad:

J	K	Clk	Q _{n+1}	/Q _{n+1}
0	0	1	Q _n	/Q _n
0	1	1	0	1
1	0	1	1	0
1	1	1	/Q _n	Q _n
x	x	0	Q _n	/Q _n

Figura 28

Esta tabla es similar a la del FF (Flip-Flop) RS, con la diferencia que ahora no hay estado prohibido con JK = 11, sino que se obtiene a la salida el estado anterior negado.

Esto como se verá mas adelante, es muy útil para implementar otros dispositivos digitales como los contadores.

Con el mismo razonamiento empleado para sintetizar el FF RS, partimos del FF asincrónico, debiendo llegar a uno síncrono disparado por nivel que cumpla con la tabla anterior.

El circuito inicial entonces será:

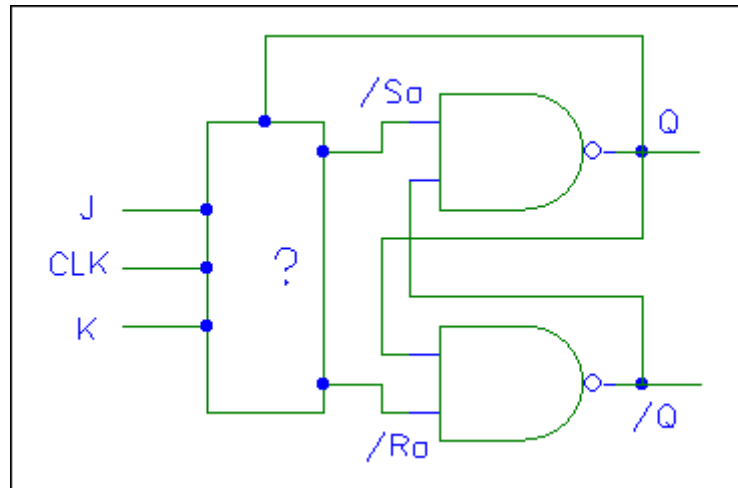


Figura 29

Como se nota, hay una diferencia importante y es que ahora debemos tener información de la salida general Q del FF síncrono, para cuando se de la combinación de entrada JK = 11.

Debemos, por lo tanto, incluirla como entrada al bloque a sintetizar.

Teniendo en cuenta las tablas del FF JK y del asincrónico podemos armar la tabla correspondiente al bloque incógnita.

Qn	J	K	Clk	/Sa	/Ra
0	0	0	1	1	x
0	0	1	1	1	x
0	1	0	1	0	1
0	1	1	1	0	1
1	0	0	1	x	1
1	0	1	1	1	0
1	1	0	1	x	1
1	1	1	1	1	0
x	x	x	0	1	1

Figura 30

De esta tabla podemos hacer los Karnaugh correspondientes a /Sa y /Ra para sintetizar el bloque:

Clk Qn	JK			
	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	x	1	1	x
10	1	1	0	0

/Sa

Figura 31

Clk Qn	JK			
	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	1	0	0	1
10	x	x	1	1

/Ra

Figura 32

De las figuras 31 y 32 obtenemos:

$$/Sa = /CLK + /J + Qn = \overline{CLK \cdot J \cdot /Qn}$$

$$/Ra = /CLK + /K + /Qn = \overline{CLK \cdot K \cdot Qn}$$

El circuito definitivo del FF JK sincrónico disparado por nivel alto, será:

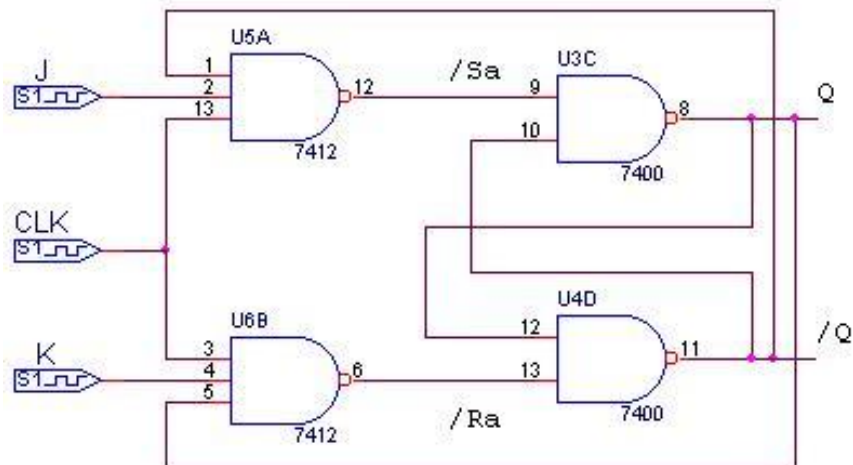


Figura 33

En la siguiente figura vemos un diagrama de tiempos donde se pueden observar como se comportan las salidas, ante cambios en las entradas:

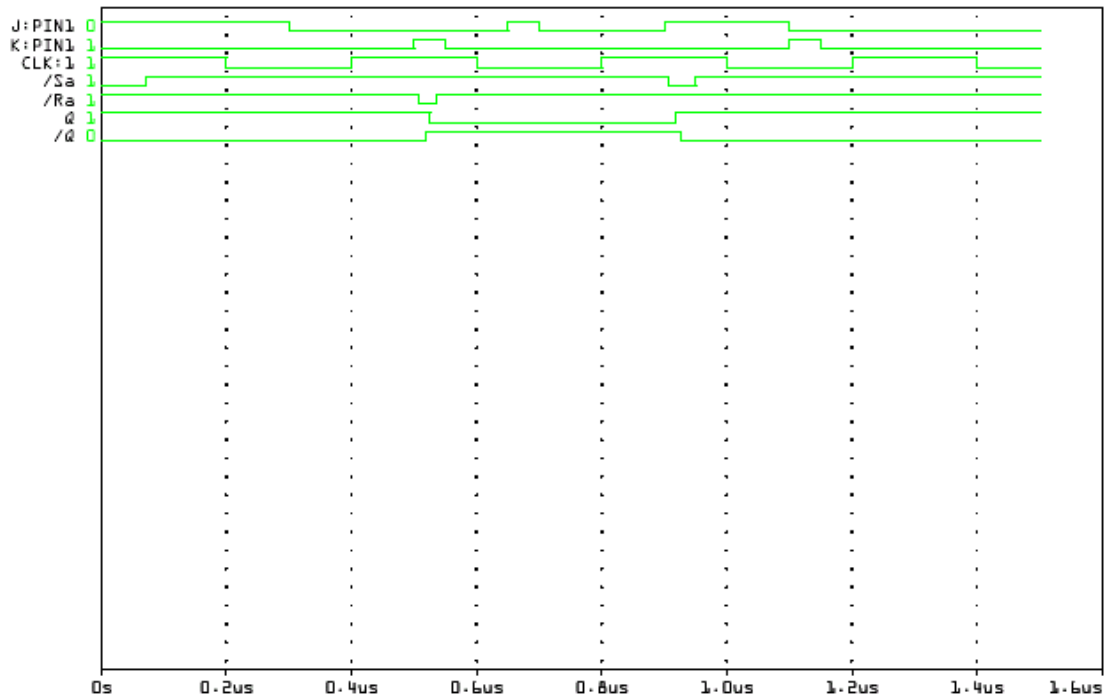


Figura 34

Como punto interesante, vemos como con la combinación JK = 11, las salidas cambian de un estado al otro con una velocidad definida por los retardos propios de las compuertas, siempre que el clock se los permita (CLK en "1").

Flip-Flop derivados del JK:

Hay dos flip-flops también de uso intensivo en los circuitos digitales que son derivados del JK, es decir, sus tablas de verdad pueden sintetizarse fácilmente desde este último con una o ninguna compuerta adicional.

Flip-Flop tipo T disparado por nivel:

Este flip-flop cumple con la siguiente tabla de verdad:

T	Clk	Qn+1	/Qn+1
0	1	Qn	/Qn
1	1	/Qn	Qn
x	0	Qn	/Qn

Figura 35

Si se observa la misma es idéntica a la del FF JK para las combinaciones $JK = 00$ que corresponde a $T = 0$ y $JK = 11$ que corresponde a $T = 1$.

Esto quiere decir que si unimos las entradas J y K, y formamos una única entrada, tendremos un flip-flop tipo "T". esto se muestra en la siguiente figura.

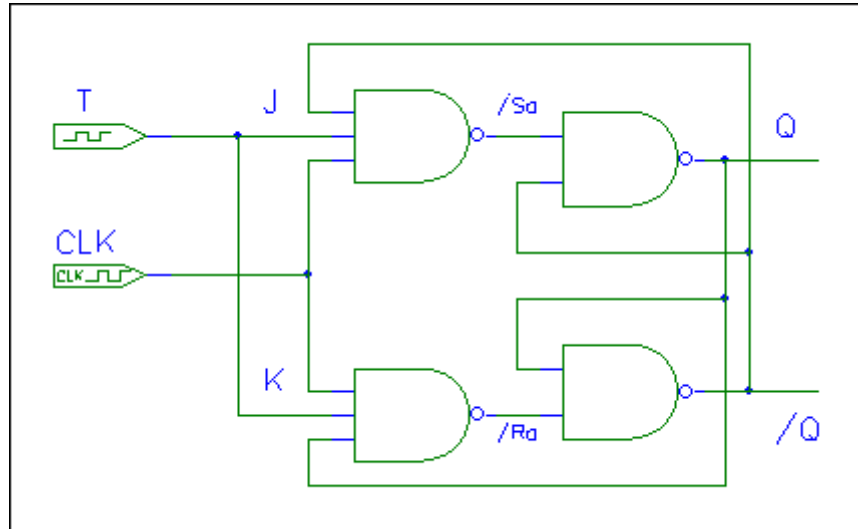


Figura 36

En la figura 37 vemos un diagrama de tiempos de este circuito, para los casos en que la entrada T vale "0" y "1", notándose que cuando $T = 1$, la salida Q alterna entre los dos niveles lógicos cada vez que el clock esté en "1" lógico (ampliado en la figura 37b).

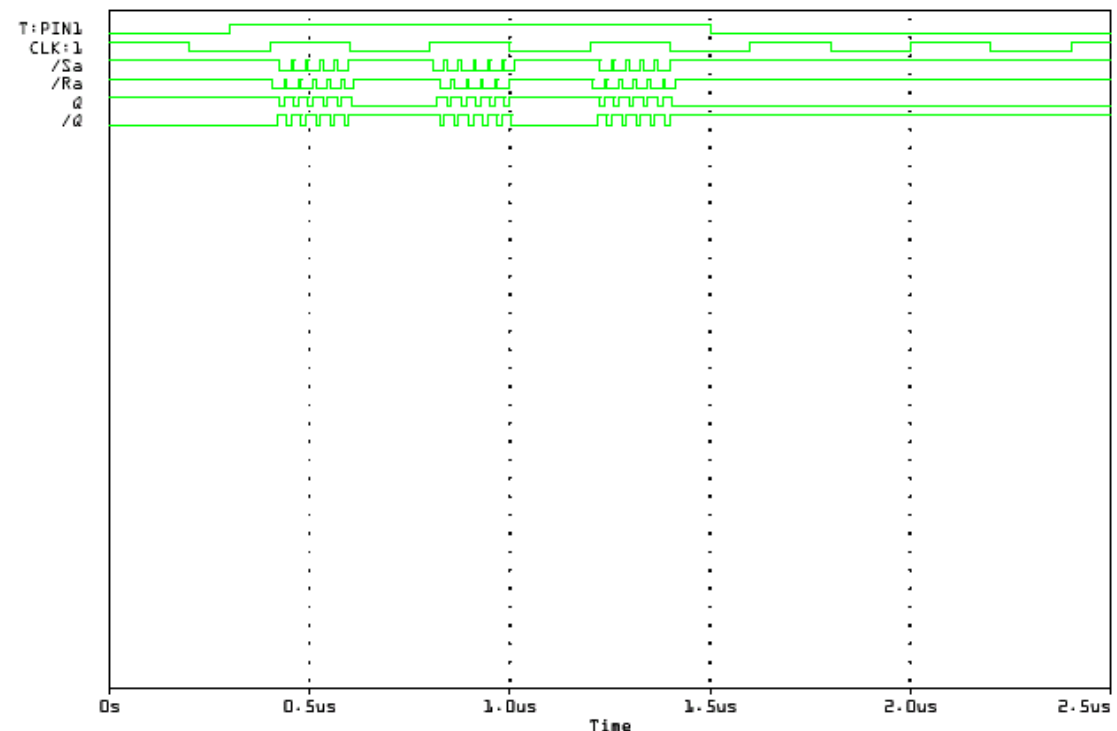


Figura 37a

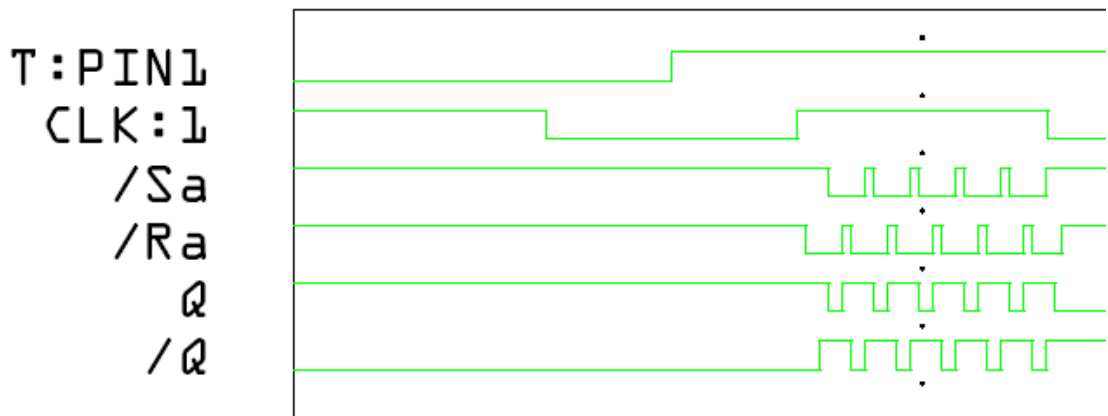


Figura 37b

Otra forma de obtener un flipflop "T", sería implementarlo con un flip-flop tipo /S /R asincrónico como veníamos haciendo.

La tabla de verdad para la síntesis del bloque combinatorio se muestra en la figura 38, mientras que los diagramas de Karnaugh en la figura 39.

T	Qn	Clk	/Sa	/Ra
0	0	1	1	x
0	1	1	x	1
1	0	1	0	1
1	1	1	1	0
x	x	0	1	1

Figura 38

Clk	T Qn				
	00	01	11	10	
0	1	1	1	1	/Sa
1	1	x	1	0	

Clk	T Qn				
	00	01	11	10	
0	1	1	1	1	/Ra
1	x	1	0	1	

Figura 39

Las expresiones de /Sa y /Ra quedarán:

$$/Sa = /T + /CLK + Qn = \overline{T + CLK + Qn}$$

$$/Ra = /T + /CLK + \overline{Qn} = \overline{T + CLK + Qn}$$

Vemos en la figura 40, el circuito completo:

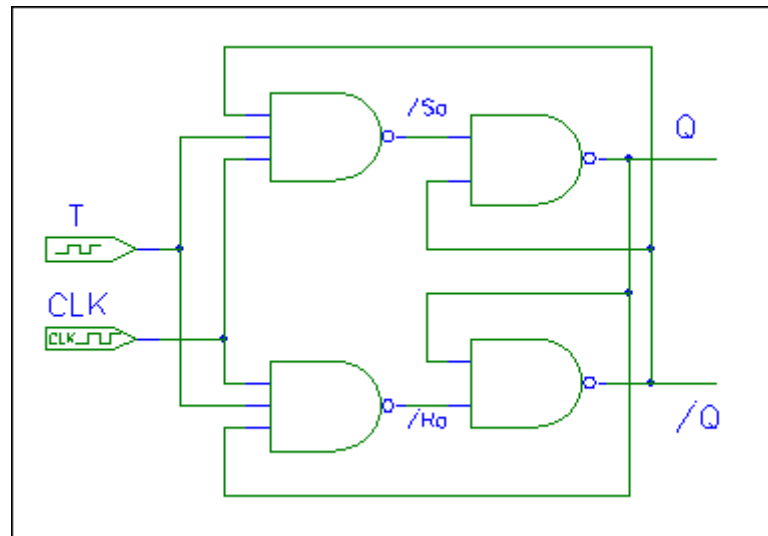


Figura 40

Como podemos ver es exactamente igual al anterior.

Ejemplo de aplicación:

Supongamos que tenemos un circuito oscilador, el cual genera una señal de reloj de 2MHz, y necesitamos para una aplicación una señal pero de frecuencia mitad, es decir, 1MHz.

Una manera de poder lograr esto con la señal original de 2 MHz es la de implementar un divisor de frecuencias binario, y esto puede hacerse empleando un flip-flop tipo T.

Según se puede ver en el diagrama de tiempos anterior, cuando la entrada T está a "1" lógico, la salida Q cambia de nivel en cada período de la señal de su CLK, o lo que es lo mismo genera en Q una señal de período doble al del clock, que es lo que estamos buscando.

En resumen, el circuito se obtiene con sólo inyectar la señal de 2MHz a la entrada de reloj del FF tipo T y poner $T = 1$.

Si disponemos de un FF JK, la diferencia es que las entradas J y K se unen a la salida del circuito generador de los 2MHz.

Flip-Flop tipo D disparado por nivel:

Este flip-flop cumple con la siguiente tabla de verdad:

D	Clk	Q _{n+1}	/Q _{n+1}
0	1	0	1
1	1	1	0
x	0	Q _n	/Q _n

Figura 41

Si se observa la misma es idéntica a la del FF JK para las combinaciones JK = 01 que corresponde a D = 0 y JK = 10 que corresponde a D = 1.

Esto quiere decir que si unimos la entrada "D" a la entrada J y conectamos esa misma entrada a través de un inversor a la entrada K del FF, tendremos un flip-flop tipo "D". Esto se muestra en la siguiente figura.

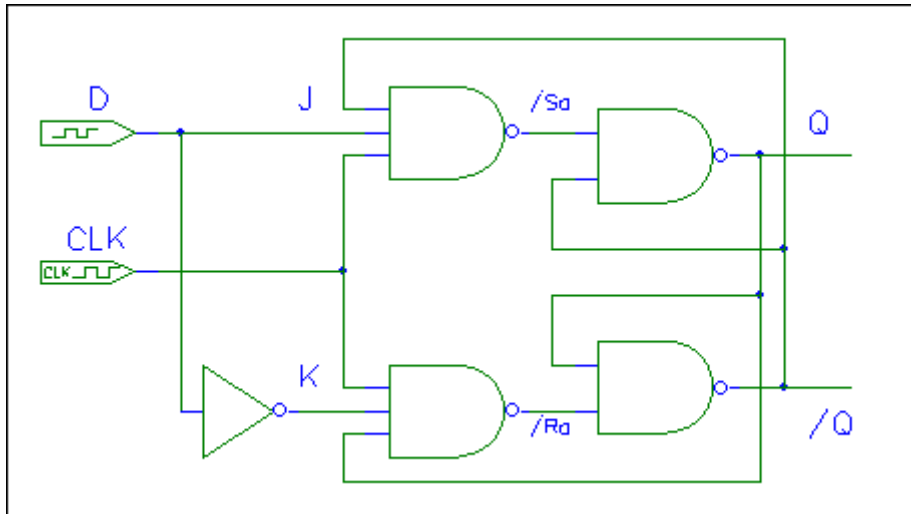


Figura 42

De igual forma que en los casos anteriores, este FF tipo D puede implementarse en base a uno /S /R asincrónico.

En las figuras 43 y 44 vemos la tabla de verdad del bloque a sintetizar y los diagramas de Karnaugh.

D	Clk	/Sa	/Ra
0	1	1	0
1	1	0	1
x	0	1	1

Figura 43

Clk	D	
	0	1
0	1	1
1	1	0

/Sa

Clk	D	
	0	1
0	1	1
1	0	1

/Ra

Figura 44

Las ecuaciones de /Sa y /Ra quedarán entonces:

$$/Sa = /D + /CLK = \overline{D \cdot CLK}$$

$$/Ra = D + /CLK = \overline{/D \cdot CLK}$$

NOTA: Este tipo de flip-flop también se puede generar en base a un SR sincrónico donde se debe conectar la entrada D a S y la R a D previo uso de un inversor.

El circuito se ve en la figura 45a y los diagramas de tiempos en las figuras 45b y 45c (esta última es una ampliación de la anterior):

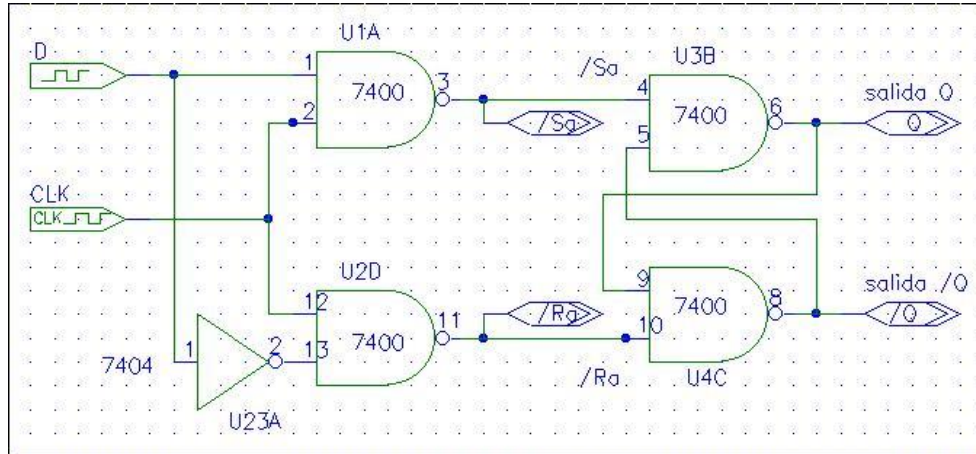


Figura 45a

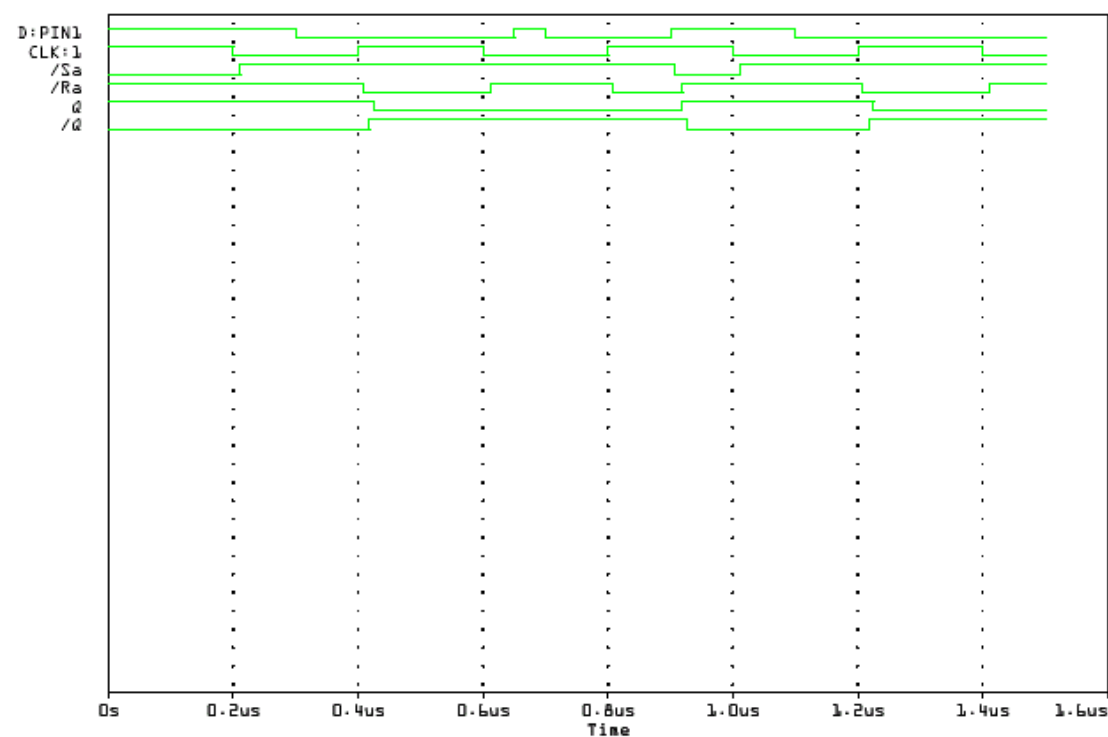


Figura 45b

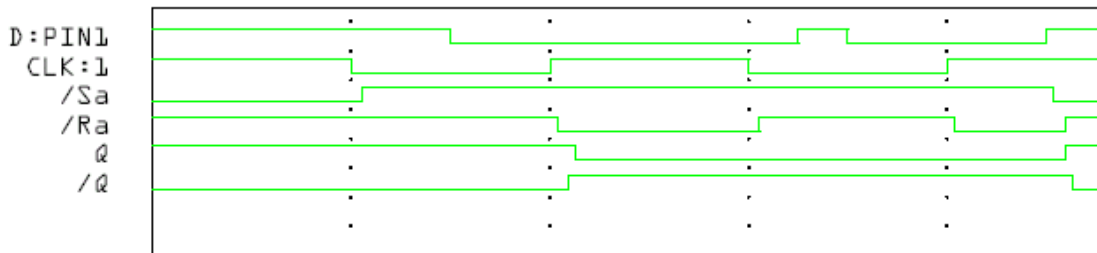


Figura 45c

Flip-Flops disparados por nivel tipo Master - Slave (Maestro - Esclavo):

Los flip-flops vistos anteriormente son disparados por nivel lógico alto (ó por nivel lógico bajo, si se quisiera), y según se observa en los diagramas de tiempo, siempre que el CLK esté en alto, las entradas SR, JK, T ó D pueden modificar las salidas.

Esto puede ser un inconveniente, cuando por ejemplo se quieren implementar dispositivos tales como registros de desplazamientos

Si bien se verán estos circuitos con mas detalle en otro capítulo, podemos dar un resumen de su funcionamiento.

Un registro de desplazamiento es un circuito compuesto por una cadena de flip-flops tipo D, los cuales se disponen en cascada, es decir, la salida del primero se conecta a la entrada del segundo y así sucesivamente, todos con sus líneas del CLK conectados a un reloj único de control.

De esta manera lo que se quiere por ejemplo es poder entrar datos por el 1er FF en forma serie e ir corriendo los bits que llegan uno a uno por los FF en cadena, todo esto controlado por la señal de CLK externa.

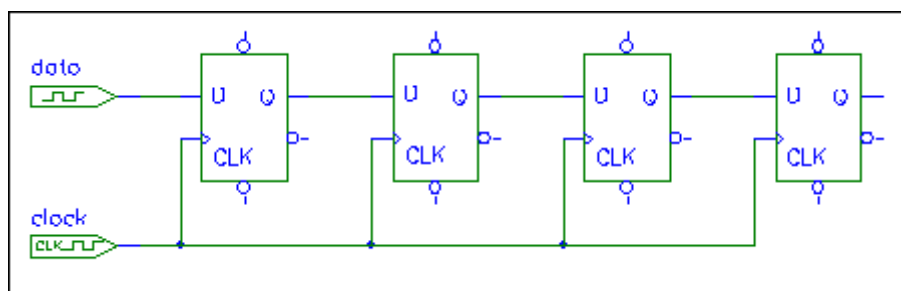


Figura 46

Lo que se pretende es que con cada pulso de CLK que se reciba, copie un bit de dato de la entrada y la ya almacenada por el 1er FF se la pase a l segundo y así siguiendo.

Si se quisiera implementar este tipo de registro con los FF tipo D vistos hasta ahora, no funcionaría ya que si por ejemplo conectamos varios FF en cascada, cuando el CLK general se ponga en "1", el dato en el 1er FF luego del tiempo de retardo del mismo aparecerá en la salida Q1, si sigue el CLK en alto, luego de otro tiempo de retardo del 2do FF aparecerá también en la salida de éste (Q2) y así siguiendo mientras esté el CLK en "1".

En resumen, no hay manera de controlar que los datos se transfieran de a uno por vez con cada pulso de CLK, salvo que el tiempo en que dure el CLK en alto sea un poco mayor al retardo de un FF, donde cada uno sólo alcanza a copiar lo que tiene en su entrada respectiva.

Pero esto es poco práctico ya que nos limita en el valor del semiperíodo del CLK, además de que dependiendo que familia lógica usemos (TTL ó CMOS) los retardos pueden variar en un orden de magnitud entre ellos.

Una solución a este problema es el empleo de los denominados Flip-flops Master - Slave, los cuales si bien son también disparados por nivel, tienen una circuitería un poco más compleja pero que evita este problema de "avalancha".

En la siguiente figura vemos un flip-flop tipo SR master-slave.

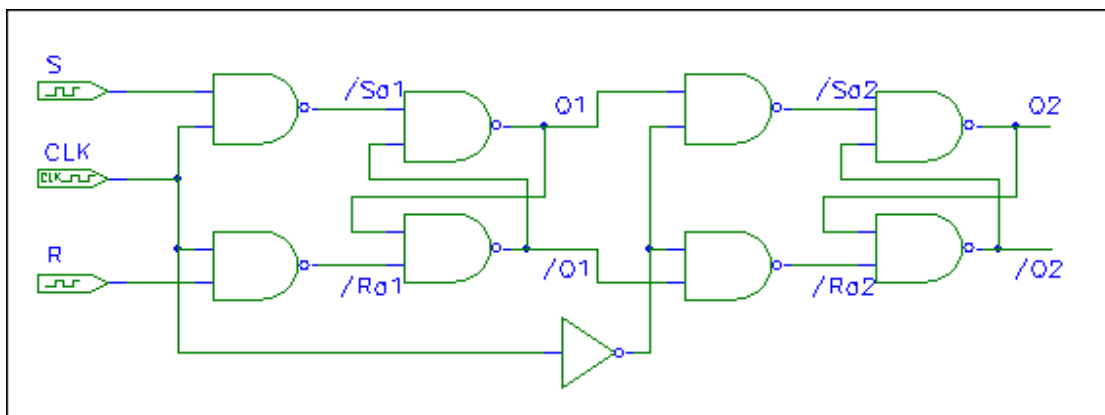


Figura 47

Se puede observar que el mismo está formado por dos flip-flops sincrónicos S R en cascada, disparados por nivel.

El primero es el que da la función al flip-flop, mientras que el segundo sólo funciona como copiator (flip-flop tipo "D"), ya que sus entradas siempre están conectadas a Q1 y /Q1 de la primera etapa (S2 R2 serán 10 ó 01, lo cual da a la salida también 10 ó 01 cuando lo permita el CLK).

Además se observa que el primer FF está conectado directamente a la línea de CLK, mientras que el 2do. FF se conecta a través de un negador.

Esto indica que cuando el CLK está en nivel lógico alto, funciona el 1er FF poniendo a la salida de Q1, el dato que resulte de la combinación de S1 R1.

Mientras tanto el 2do FF estará deshabilitado ya que su CLK estará en nivel bajo, el cual mantendrá el estado anterior.

Cuando se baja la línea de CLK, el 1er FF mantendrá su salida con el último valor obtenido, mientras que el 2do FF ahora estará habilitado copiando justamente la información que se transfirió al 1er FF cuando el CLK estaba en alto.

De esta manera el conjunto de ambos FF funciona como un FF SR donde su salida Q = Q2 responde a las entradas SR (que se presentaron con el nivel de CLK en alto) recién cuando el nivel del CLK es bajo.

En las siguientes figuras vemos un diagrama temporal de este FF.

En la 48a, el diagrama completo. En la 48b, para el caso de dejar S y R en "1" simultáneamente y en la 48c, para el caso de funcionamiento normal.

Se puede observar en todas ellas que las salidas Q y/Q, cambian luego de que lo hace la primera etapa, cuando el reloj está en "0".
 La 48b resume la inestabilidad que puede ocurrir cuando se presenta la combinación SR = "11".

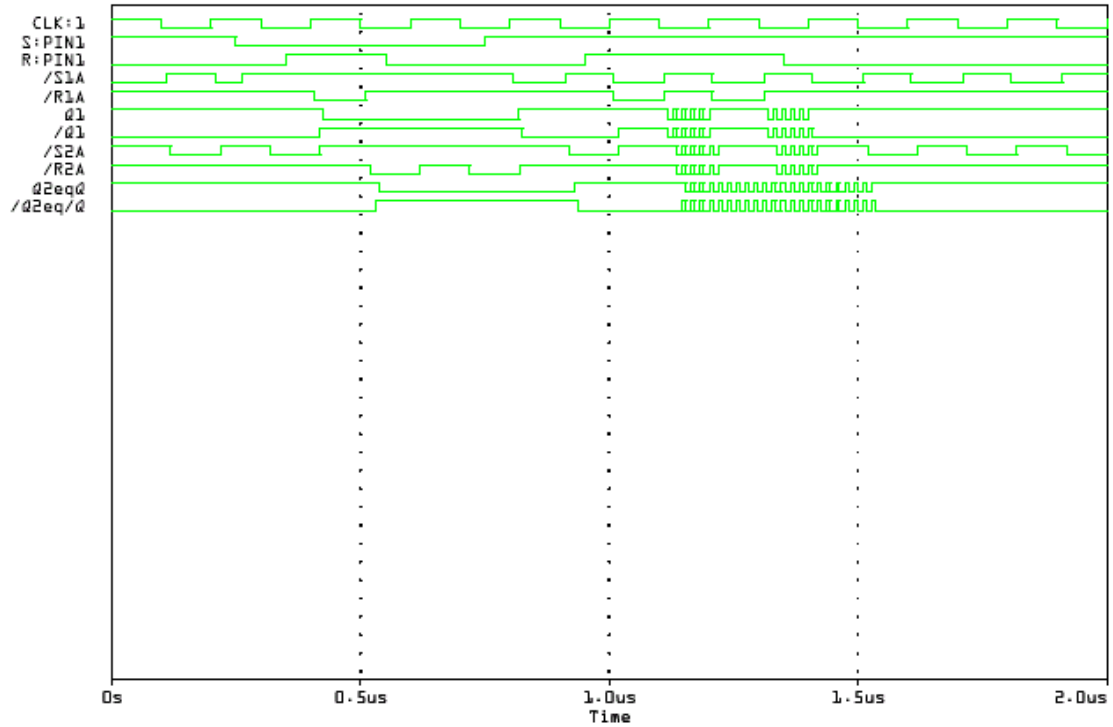


Figura 48a

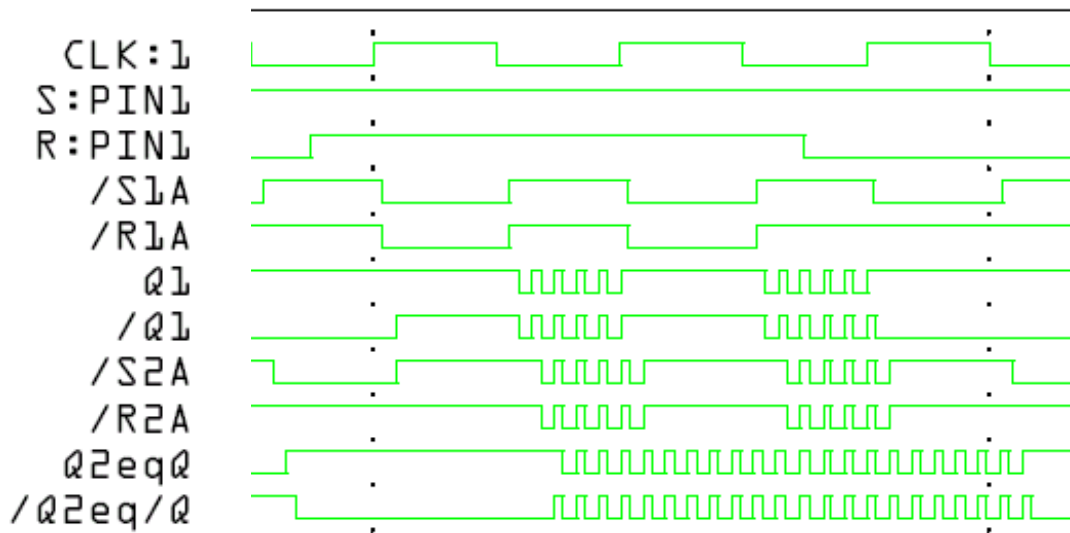


Figura 48b

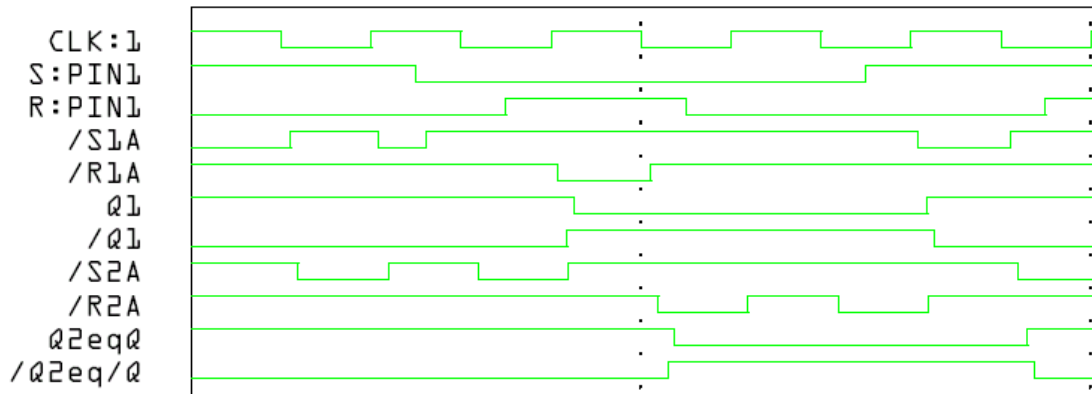


Figura 48c

Gracias a que las salidas Q y /Q, sólo cambian (con el reloj en "0") dada la última combinación presente en las entradas antes de que el clock pase a "0", es posible ahora implementar una cadena de Flip-Flops tipo D con este SR Master-Slave, ya que no hay posibilidad de avalancha.

Se debe si o si generar secuencialmente pulsos de CLK para que la información se vaya trasladando de FF a FF.

De la misma manera que con el SR, se puede implementar un flip-flops tipo JK.

En la siguiente figura vemos el circuito correspondiente:

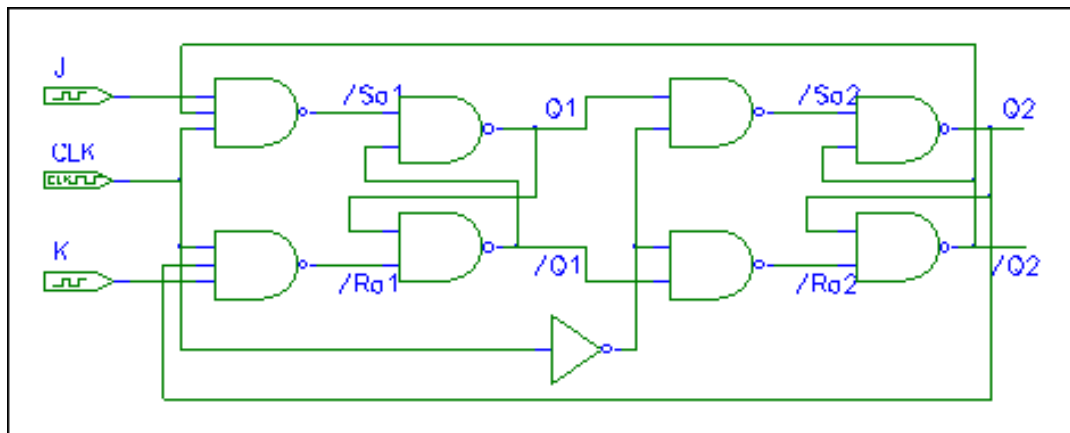


Figura 49

Se puede observar como la segunda etapa cumple la misma función que antes, es decir, de copiador, para que la salida se actualice recién en el nivel bajo del CLK.

Además en este caso como se necesita disponer de información del estado de la salida Q, se hace la realimentación de esta hacia la primera etapa que es la que conforma la función JK.

Los flip-flops tipo D y T se pueden implementar como se vió anteriormente.

Flip-flops activados por flanco

Problema de captura de "unos": Los flip-flops sincrónicos master-slave tienen la particularidad que eliminan el problema de la cascada de flip-flops, pero adolecen de otro que se denomina: habilidad de captura de "unos".

Si volvemos a analizar por ejemplo el FF SR anterior, veremos que como se dispara por nivel, si en un primer momento estando $SR = 00$ con $CLK = 1$, alguna de esas entradas se pone en "1" el suficiente tiempo para vencer los retardos internos de la primera etapa, ésta podrá modificar su valor.

Por ejemplo inicialmente teníamos $Q1 = 0$, $SR = 00$ y se pone brevemente $S = 1$, pasando $Q1$ a 1 sin pretenderlo (idem $Q1$ a 0 si teníamos $Q1 = 1$ y pasa R a 1 momentáneamente, estando S en 0).

El problema radica en que en cualquier circuito electrónico puede inducirse ruido ya sea en cables, la fuente de alimentación, etc. ó la aparición de glitches en las salidas de otros dispositivos, lo que puede generar este tipo de "pulso positivo no deseado", con consecuencias desastrosas en el funcionamiento de equipo en cuestión.

Una solución a esto y que es ya utilizada desde hace algunos años, es el empleo de flip-flops **disparados por flanco**.

Esto quiere decir que a diferencia de los anteriores, ahora las salidas podrán modificar sus estados (si la combinación de entradas lo permite) recién con una de las dos transiciones posibles del clock (flanco de subida o de bajada).

Es decir, la habilitación de las entradas de datos del flip-flop se realiza no con un nivel lógico, sino con el cambio de un nivel lógico a otro preestablecido.

Una de las especificaciones importantes de este tipo de FF es el denominado **Tiempo de establecimiento (Set-Up Time)**, el cual define el tiempo en que el dato debe estar estable antes de que llegue la transición que genera cambios en la salida del flip-flop a fin de que la salida sea la correcta.

Otro parámetro temporal importante, es el denominado **Tiempo de mantenimiento (Hold Time)**, el cual se define como el tiempo mínimo que debe seguir siendo estable las entradas de datos, para que efectivamente sea tomada como válida por el flip-flop.

Cuál es la ventaja de esto? Que para que el ruido que se presente en las entradas de datos, modifique las salidas, tendría que ocurrir dentro de ese intervalo de tiempo antes de la transición correspondiente del clock y mantenerse durante un breve lapso más ya que si es de muy corta duración no tendrá efecto alguno sobre el funcionamiento del flip-flop, lo cual nos beneficia ya que no queremos su ocurrencia.

NOTA: El Hold Time se especifica en los FF tipo D. En los tipo JK es nulo.

Los flip-flops que generalmente se pueden adquirir en el mercado (familia TTL 74LS/ALS/F y CMOS CD4XXX y 74HC/HCT/ACT), son todos disparados por flanco del tipo D y JK, en general con entradas asincrónicas de Set/Reset ó sólo Reset.

Si se requiere de un flip-flop tipo T, éste deberá ser implementado con algunos de los anteriormente mencionado.

Flip-Flop tipo D disparado por flanco:

En la siguiente figura vemos un circuito que corresponde a un flip-flop tipo D, disparado por flanco ascendente ó de subida (de nivel bajo a alto).

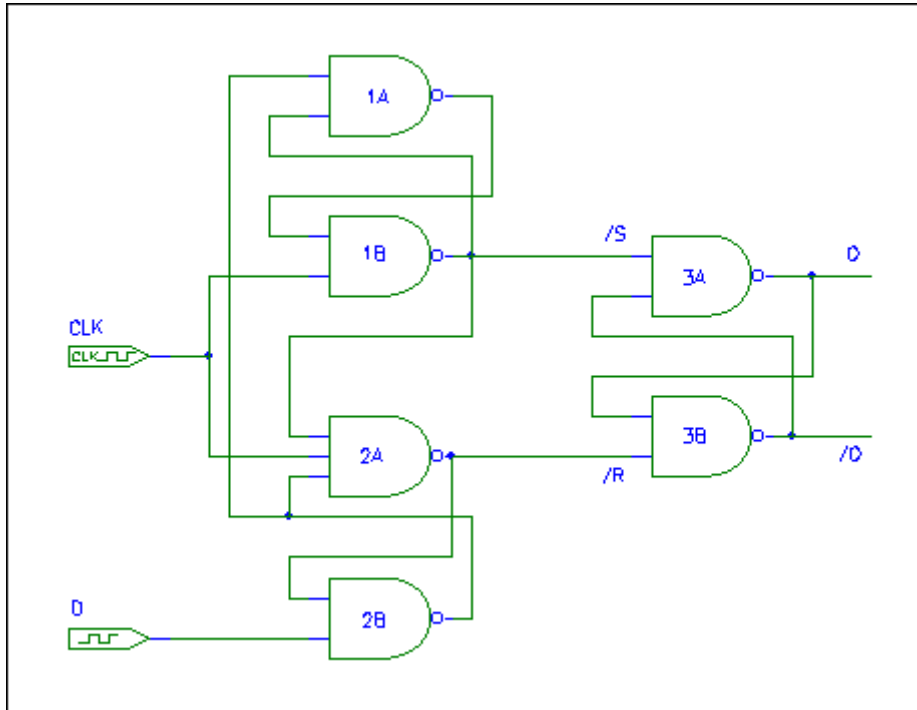


Figura 50

Podemos observar que el mismo está formado en base a un flip-flop asincrónico tipo /S /R mas cuatro compuertas NAND las cuales hacen las funciones de conversión del formato tipo D al SR y de habilitación de la entrada de datos sólo en la transición positiva del reloj.

NAND 1B y NAND 2A se emplean para forzar a /S y /R a 11 cuando el CLK esté en "0", mientras que NAND 1A y NAND 2B tienen por misión modificar los niveles de /S y /R por cambios en la entrada D.

Para analizar su funcionamiento comenzaremos, suponiendo que CLK = 0, por lo tanto /S y /R serán 11, manteniendo el estado anterior.

En esta condición tendremos:

Q1A = según estado anterior

Q1B = 1

Q2A = 1

Q2B = según estado anterior.

Si hacemos D = 0 en estas condiciones, sólo modificaríamos el estado de Q2B a "1" y el de Q1A a "0". Siguen aún Q1B y Q2A ambos en "1", forzados por el "0" del CLK y como consecuencia la salida Q se mantendrá sin cambios.

Si ahora llevamos CLK a "1", tendremos que:

$$Q2A = 0 \text{ ya que } Q2B = 1, \text{ CLK} = 1 \text{ y } /S = 1.$$

Esto fuerza a que el FF asincrónico cambie a $Q = 0$, pues tenemos $/S /R = 10$.
Esto fuerza a la salida Q a cambiar a "0" (reset).

Si mientras el CLK está en alto hay cambios en la entrada D, las entradas $/S /R$ no van a cambiar ya que para que D afecte al FF debe cambiar la NAND 2B y su salida Q2B está forzada a 1 por el 0 en $/R$.

Si el CLK ahora se va al nivel bajo, es decir, a "0", fuerza a las salidas Q1B y Q2A a pasar a "1", manteniéndose el estado anterior y volviendo al punto de partida del análisis.

En estas condiciones:

$$\begin{aligned} Q1B &= 1 \\ Q2A &= 1 \\ Q2B &= 1 \\ Q1A &= 0 \end{aligned}$$

Por otro lado, si suponemos que D pasa a nivel alto mientras está el CLK en bajo, tendremos que:

$$Q2B = 0$$

Este valor no afectará a NAND 2A ya que se ve forzada a 1 por el CLK en "0".

Por otro lado Q1A pasa a "1" pues $Q2B = 0$, pero Q1B no se modifica pues manda el CLK que sigue en "0".

Si bien nada cambió en la salida Q ya Q1A puso un "1" en una de las entradas de la NAND 1B para que cuando venga la transición del CLK de "0" a "1", haga que Q1B pase a "0".

Efectivamente, si CLK pasa a "1", tendremos $/S = 0$, mientras que $/R$ no se modificará pues con $D = 1$, sigue estando Q2B en bajo, forzando a la NAND 2A a estar en alto.

En ese momento del cambio del CLK a "1", pasará la salida Q a 1 (set).

NOTA: Si antes de que el CLK pase a "1", hay cambios de nivel en la entrada D, sólo tendrá validez la que se presente un instante antes de la transición 0 a 1 del CLK.

Luego de esta transición, donde el CLK queda en "1", aunque se modifique el estado de la entrada D, (habrá cambios en la salida Q2B), el "0" de $/S$ forzará a Q2A a mantenerse en "1" por un lado y forzará a Q1A a estar en "1" impidiendo que la otra entrada de NAND 1A que se podría modificar con el cambio en la entrada D, tenga efecto alguno sobre Q1B.

Si pasamos luego al CLK a un nivel bajo, tampoco habrá modificaciones ya que tendremos $Q1B = Q2A = 1$ permanentemente, manteniéndose el estado anterior en la salida Q.

En las hojas de datos de los manuales de circuitos integrados tales como Motorola, National, RCA, Fairchild, Texas, etc. encontramos una variedad de flip-flops catalogados como latches ó también como flip-flops tipo D, casi todos disparados por flanco ya sea descendente ó ascendente salvo alguna rara excepción.

De la familia Low Power Schottky TTL podemos citar por ejemplo:

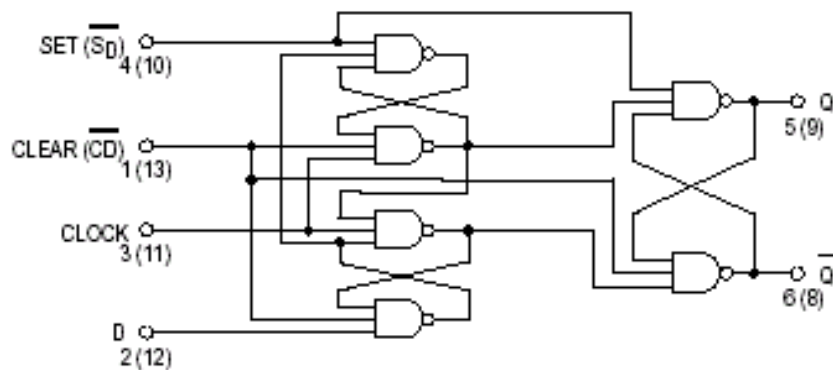
- 74LS74A Doble FF D disparado por flanco ascendente (positive edge triggered)
- 74LS75 Latch (cerrojo) D de 4 bits con salidas Q y /Q
- 74LS77 Latch D de 4 bits con salidas Q
- 74LS373 Octuple latch con salidas tri-state **asincrónico**
- 74LS374 Octuple latch con salidas tri-state sincrónico.

De la familia CMOS serie 4000:

- CD4013BM Doble flip-flop D.
- CD4042BM Cuádruple latch D.
- CD40174BM Séxtuple flip-flop D

Ejemplo comercial: El 74LS74

En la siguiente figura vemos el diagrama circuital, la tabla de verdad y un diagrama en bloques con el conexionado de este doble flip-flop tipo D, disparado por flanco ascendente (positive edge - triggered).



OPERATING MODE	INPUTS			OUTPUTS	
	SD	SD	D	Q	Q̄
Set	L	H	X	H	L
Reset (Clear)	H	L	X	L	H
*Undetermined	L	L	X	H	H
Load "1" (Set)	H	H	h	H	L
Load "0" (Reset)	H	H	l	L	H

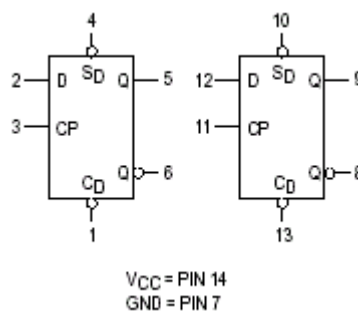


Figura 51

En la figura 52 tenemos las especificaciones sobre la evolución temporal de las salidas Q y /Q en función de la entrada D y el reloj, mientras que en la figura 53, muestra como varían dichas salidas en función de las entradas asincrónicas de reset y set.

Además se indican en esas formas de onda los diferentes tiempos asociados al funcionamiento de este flip-flop.

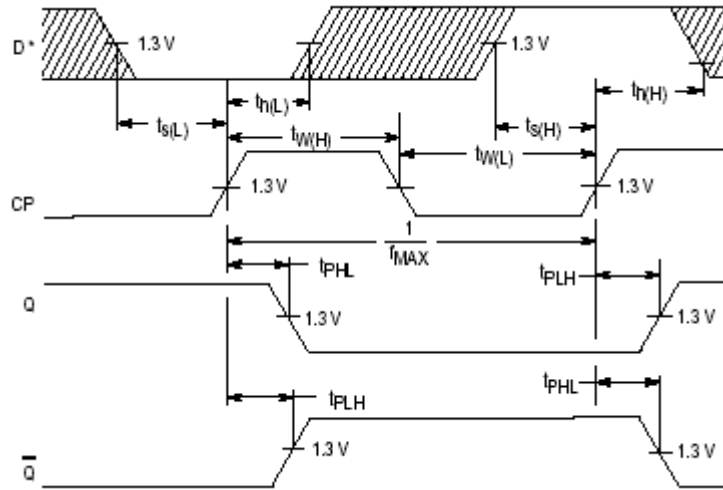


Figura 52

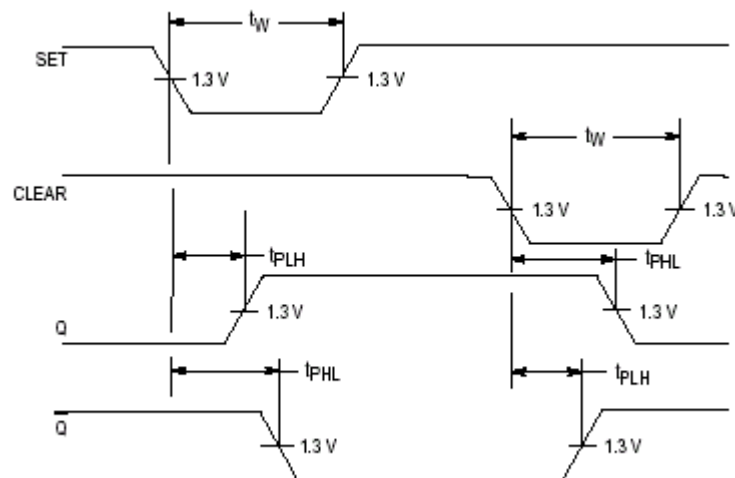


Figura 53

Como se puede apreciar, el circuito es similar al analizado anteriormente con la adición de mas entradas a las compuertas a fin de poder agregar las funciones de Set y Reset asincrónicas a través de las líneas /Set y /Clear respectivamente.

Según la tabla de verdad una combinación /Sd /Cd = 01 fuerza la salida a 1 y viceversa para /Sd /Cd = 10.

$\overline{Sd} / \overline{Cd} = 00$ se considera prohibido ya que no se puede garantizar el estado final en la salida.

$\overline{Sd} / \overline{Cd} = 11$ inhabilita las entradas asincrónicas las cuales no tendrán efecto en el funcionamiento del FF.

Por otro lado, según se comentó anteriormente, se definen los tiempos de **set-up** y de **hold**, los cuales son importantes a tener en cuenta para garantizar el buen funcionamiento del circuito

Para este dispositivo en particular (tecnología LS TTL), tenemos que el tiempo de establecimiento será como mínimo de $t_s = 20\text{ns}$, mientras que el tiempo de mantenimiento será igual ó mayor a 5ns

Esto significa que el dato presente en la entrada D debe estar estable al menos 20ns antes de generarse el flanco positivo del CLK y mantenerse en ese nivel por lo menos durante 5ns después de pasado dicho flanco.

Otras especificaciones temporales importantes son por ejemplo el tiempo de retardo de propagación desde que se genera la transición positiva del CLK al momento en que cambia la salida Q debido a esa transición si la entrada lo permite.

Su nomenclatura es t_{PLH} (tiempo de propagación bajo a alto) ó t_{PHL} (tiempo de propagación de alto a bajo) cuyo valor típico es de 13ns y 25ns , respectivamente.

Este tiempo también es definido para medir el retardo desde que se generan cambios en las entradas asincrónicas y la respuesta de la salida Q ante dichos cambios.

Flip-Flop tipo JK disparado por flanco:

Este flip-flop al igual que el visto anteriormente, tiene la particularidad que la salida puede conmutar (si las entradas J y K por la tabla de verdad habilitan el cambio) con uno de los flancos de la señal de Clock.

En la siguiente figura vemos un circuito de un FF JK el cual corresponde al esquema interno del 74LS112 (dual JK negative edge-triggered flip-flop):

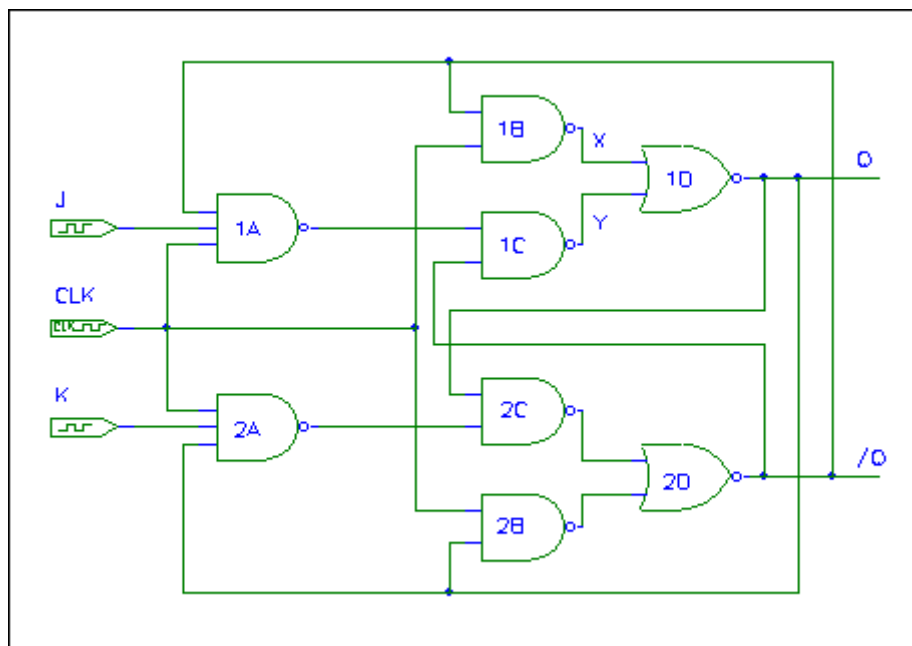


Figura 54

La clave del mecanismo de disparo por flanco del clock está centrada en el empleo de los pares de compuertas AND 1B - AND 1C por un lado y AND 2C - AND 2B por el otro.

En el primer par se puede notar que tienen en común una entrada que proviene de la realimentación de \bar{Q} , pero la habilitación del clock viene por distintos caminos, a la AND 1B en forma directa, mientras que a la AND 1C a través de una compuerta AND 1A. Con esto lo que se logra es generar un retardo en la generación de la señal indicada como Y, respecto de la X.

Veremos como funciona esto a continuación.

Para el análisis vamos a suponer que el FF está en un estado tal que:

$$\begin{aligned} \text{CLK} &= 0 \\ \text{Q} &= 0 \\ \text{J} &= 1 \\ \text{K} &= 0 \end{aligned}$$

Esto quiere decir que en algún momento la salida Q del FF va a ir a "1", ya que J y K lo estarían definiendo así a través de la tabla de verdad.

En esta condición con el CLK en bajo, tendremos que

$$\begin{aligned} \text{Y} &= 1 \text{ ya que } \bar{\text{Q}} = 1 \text{ y } \text{Q1A} = 1 \\ \text{X} &= 0 \text{ ya que } \text{CLK} = 0 \\ \text{Q1A y Q2A} &= 1 \text{ pues } \text{CLK} = 0 \\ \text{Q} &= \text{Q1D} = 0 \text{ pues } \text{Y} = 1 \text{ siendo su salida estable.} \\ \text{Q2C} &= 0 \text{ y } \text{Q2B} = 0 \text{ pues } \text{Q1D} = 0. \\ \bar{\text{Q}} &= \text{Q2D} = 1 \text{ pues } \text{Q2C} = 0 \text{ y } \text{Q2B} = 0. \end{aligned}$$

Supongamos ahora que el CLK cambia a nivel alto, tendremos:

$$\begin{aligned} \text{Q1A} &= 0 \text{ pues } \text{J} = 1, \text{CLK} = 1 \text{ y } \bar{\text{Q}} = 1 \\ \text{Q2A} &= 0 \text{ ya que } \text{K} = 0 \\ \text{Q1B} &= \text{X} \text{ cambiará a } 1 \text{ pues } \text{CLK} = 1 \text{ y } \bar{\text{Q}} = 1 \\ \text{Q1C} &= \text{Y} \text{ cambiará a } 0 \text{ pues } \text{Q1A} = 0 \end{aligned}$$

Pero estas dos últimas compuertas tienen distintos tiempos de respuesta, ya que X cambiará antes que Y pues Y tiene un retraso mas debido al tiempo de retardo de la compuerta Q1A.

Por lo tanto los cambios que se generan en XY son $01 \rightarrow 11 \rightarrow 10$.

Como se observa, el efecto neto sobre la entrada de NOR 1D es que siempre hay una entrada en nivel alto, por lo tanto Q1D no se modificará cuando el CLK pasa de bajo a alto.

Por otro lado tenemos que $\text{Q2C} = 0$ y $\text{Q2B} = 0$, forzados por $\text{Q} = 0$.

Analizando ahora que pasa cuando el CLK baja, tendremos:

En estas condiciones vuelve a ocurrir que responde X antes que Y, pero con la diferencia que aquí si habrá cambio en Q1D pues la secuencia de respuesta XY será: 10 → 00 → 01, es decir, habrá un momento en donde las dos entrada serán "0" habilitando a NOR 1D a ir a "1".

Con respecto a las compuertas inferiores, tendremos que al bajar el CLK a "0", Q2B pasa a "0" y Q2C pasa a "1" habilitada por Q.

Este "1" en la NOR 2D pone a /Q = Q1D en "0" y se mantiene.

Con respecto a la habilidad de no capturar "unos", podemos ver que si cuando el CLK está todavía en alto, pasamos J de 1 a 0 antes de que venga la transición 1 a 0 del CLK, pasará Y de 0 a 1 y se quedará en ese nivel ya que /Q = 1

Quiere decir que el FF responde a último momento antes de generarse la transición negativa del CLK.

En realidad, las entradas J y K deben estar al menos estables un poco mas del tiempo de retardo promedio de las compuertas para que como en este caso la salida Y pueda reaccionar.

Con respecto al análisis del funcionamiento del FF ante la combinación JK = 01, es similar a lo anterior pero la diferencia es que ahora trabajan las compuertas AND 2C y AND 2B en forma análoga a las 1B y 1C.

En el FF analizado, existe un problema potencial en el empleo de este esquema particular para generar disparo por flancos, dado que se necesita que se generen transiciones del clock muy rápidas a fin de lograr por diferencias de tiempos de respuesta entre compuertas, que funcione correctamente.

Si dichas transiciones son lentas puede fallar la operación.

Otro ejemplo comercial de flip-flop JK disparado por flancos:

El 74LS73A: Doble FF JK disparado por flanco descendente.

En la figura 55a vemos el diagrama circuital y el esquemático de este flip-flop, en la figura 55b la tabla de verdad del mismo y en la figura 56 las especificaciones temporales.

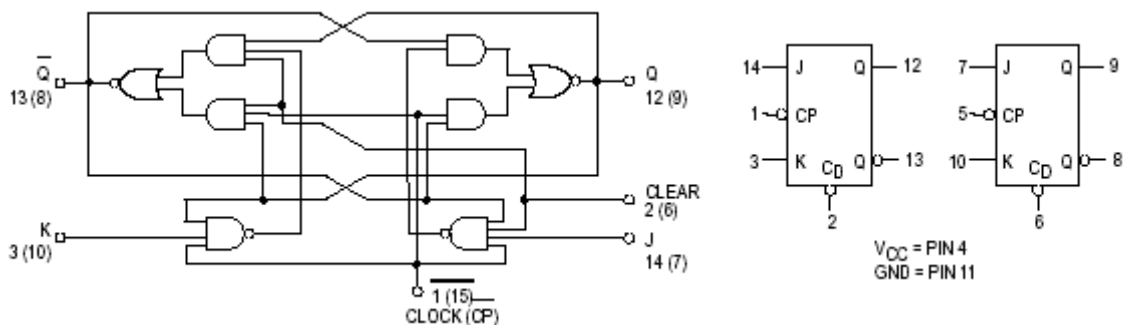


Figura 55a

OPERATING MODE	INPUTS			OUTPUTS	
	C _D	J	K	Q	\bar{Q}
Reset (Clear)	L	X	X	L	H
Toggle	H	h	h	q	q
Load "0" (Reset)	H	l	h	L	H
Load "1" (Set)	H	h	l	H	L
Hold	H	l	l	q	q

H, h - HIGH Voltage Level

L, l - LOW Voltage Level

X - Don't Care

l, h (q) - Lower case letters indicate the state of the referenced input (or output) one set-up time prior to the HIGH to LOW clock transition.

Figura 55b

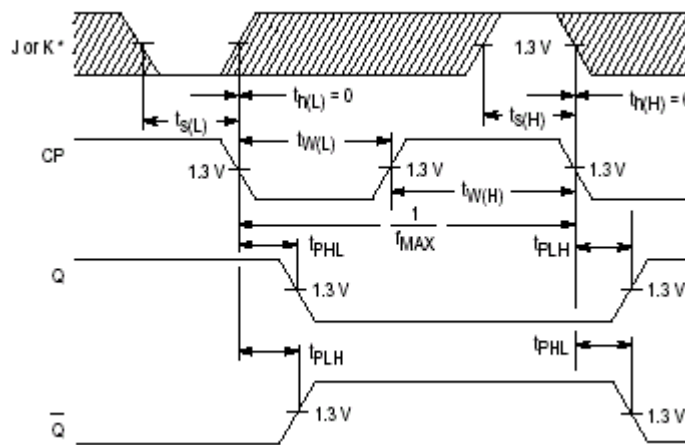


Figura 56a

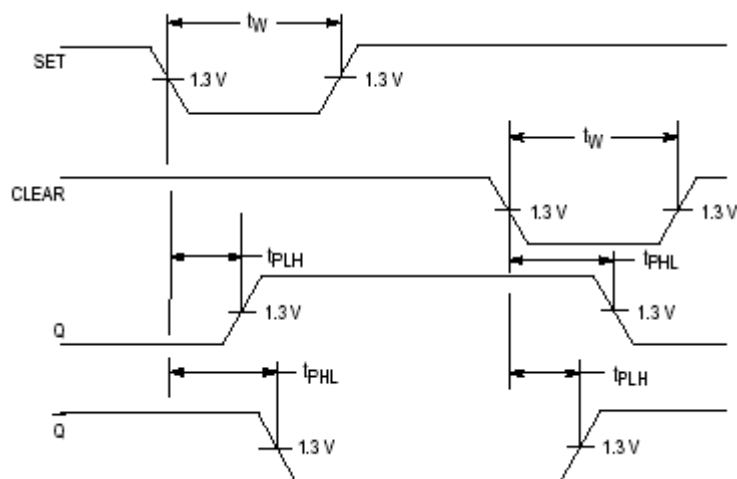


Figura 56b

De las especificaciones temporales se puede observar que el tiempo de hold es nulo, lo que implica que para que la salida responda correctamente basta con que el dato presente en J y K debe mantenerse estable hasta que se genere la transición negativa del clock.

Existe además una restricción en cuanto al mínimo valor que debe mantenerse el clock en nivel alto (20ns).

En cuanto al tiempo de set-up, su valor es similar al del flip-flop 74LS74 (20ns).

En la figura 56a se puede ver una forma tradicional de indicar con una zona rayada, en un intervalo de tiempo, la condición de don't care (no importa) del nivel lógico de una entrada ó ambas (aquí pueden ser J y/ó K).

Esto significa que durante esa condición no interesa que varíe(n) la(s) entrada(s), ya que el flip-flop no la(s) reconocerá(n).

Sí, es importante, por ejemplo en el intervalo t_s , antes de que se genere la transición activa del clock (flanco descendente) que permanezca(n) estable(s).

Luego de dicha transición, la misma(s) puede(n) modificarse ($t_h = 0$).

Ejemplo de aplicación con flip-flops disparados por flanco:

Se dispone de un sensor de velocidad optoelectrónico, el cual consta de dos discos ranurados, solidarios a un mismo eje.

Cada uno lleva asociado un par emisor-detector óptico que da una señal eléctrica (0V si el detector no recibe luz y +5V si la recibe).

De este modo, cuando el eje (solidario al eje cuya velocidad se quiere medir) gire, las señales provenientes de estos sensores darán una onda cuadrada, cuya frecuencia será proporcional a dicha velocidad.

La razón de tener dos discos, es que éstos están espacialmente espaciados de tal manera que las formas de onda cuadradas que se obtienen están desfasadas en 90° eléctricos.

El problema consiste en implementar un circuito que sea capaz de detectar el sentido de giro del eje, a partir de estas dos señales.

En la siguiente figura tenemos las formas de onda mencionadas para los dos sentidos de giro:

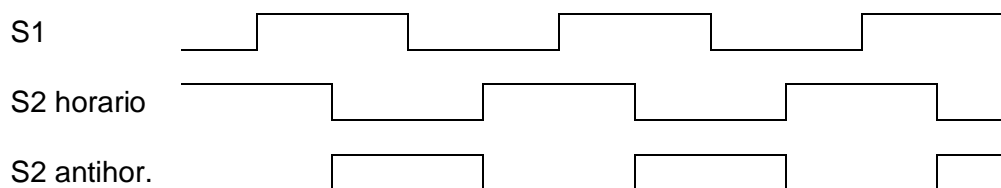


Figura 57

Si tomamos la superior (sensor 1) como referencia, vemos que en un caso la señal del sensor 2 está 90° adelantada y en el otro 90° atrasada respecto de S1.

Recordando que los flip-flops disparados por flanco, capturan la(s) entrada(s) un instante antes de generarse el flanco de activación del flip-flop, podemos pensar en emplear uno tipo D (por ejemplo el 74LS74), donde la entrada de datos se conecta a S2 y la de reloj a S1 (figura 59).

De este modo cuando S2 esté adelantada con respecto a S1, en cada flanco ascendente el flip-flop copiará un "1", mientras que si S2 está atrasada con respecto a S1, el mismo copiará un "0".

La línea /CLR está conectada a una serie R-C a fin de que cuando se dá alimentación al circuito, el flip-flop esté momentáneamente con esta línea en "0" (capacitor descargado) y luego vuelva a "1" lógico (al cargarse el capacitor), todo esto mientras /PRESET está fijo a un "1" (a Vcc).

De esta manera se genera un RESET en el power-up (al conectar la alimentación).

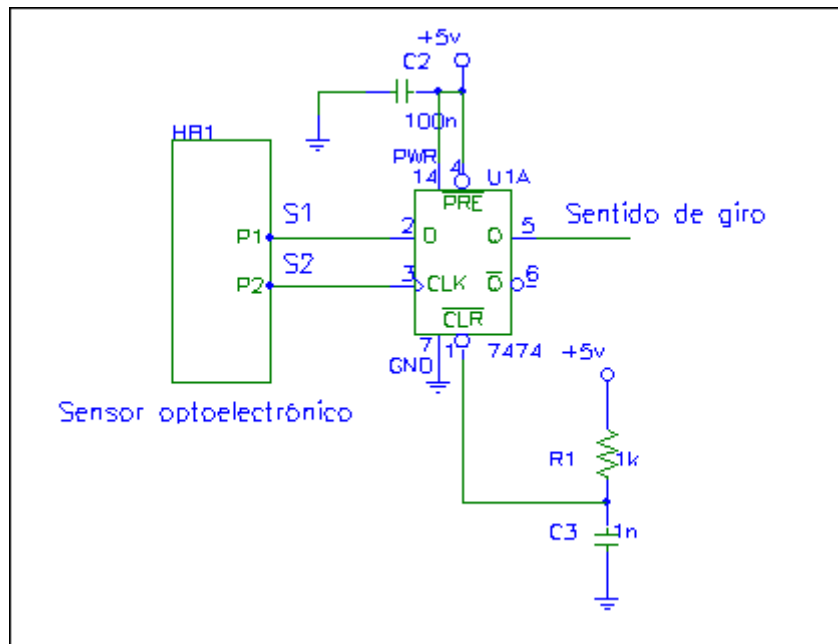


Figura 58

4 - Bibliografía.

- 1 - Circuitos Digitales y Microprocesadores. Herbert Taub. Editorial Mc Graw-Hill.
- 2 - Sistemas Digitales. Principios y Aplicaciones. Ronald Tocci. Editorial Prentice Hall.
- 3 - Electrónica Digital. James Bignell & Robert Donovan. Editorial CECSA.
- 4 - Manual Motorola Low Power Schottky.
- 5 - Manual National Semiconductor Corporation CMOS Serie CD4000. Editorial EMEDE.