

Muchos sistemas digitales deben trabajar en sincronismo con una secuencia de señales binarias (un tren de impulsos). Veamos por ejemplo el funcionamiento de una máquina calculadora en el momento de hacer una operación; el primer paso para hacer una suma, es introducir el primer sumando, luego se introduce el operador (suma), el tercer paso es introducir el segundo sumando y por último introducir el igual para obtener el resultado. ¿Qué ocurre dentro de la calculadora? Introducir el primer sumando implica guardar en una memoria un valor dado, luego guardo en otra memoria el valor asociado a la operación a realizar, el tercer paso es análogo al primero y en el cuarto se acciona la el programa interno que le “dice” a la calculadora que pasos tiene que ejecutar, teniendo en cuenta el valor asociado al operador de entrada (tomar el dato de la primer memoria ponerlo en la entrada del circuito sumador, tomar el dato de la segunda memoria ponerlo como segundo dato del circuito del sumador, accionar el circuito sumador, tomar los valores del resultado y mostrarlos en la pantalla. Cada uno de estos pasos se debe realizar en un momento preciso, ninguno de ellos puede comenzar antes o después del indicado. Por lo tanto hará falta que exista también un circuito que marque el compás para que todo esto realice con la precisión debida (contador). De lo visto asumimos que también hace falta que existan circuitos que guarden datos (memorias).

El bloque básico para realizar este tipo de operaciones se llama circuito biestable. Ya sea para marcar tiempos (Temporizadores) como para establecer registros de memoria.

Los circuitos de este tipo son los llamados sistemas secuenciales, a diferencia de los combinacionales, pues el estado de salida de este tipo de circuitos no sólo depende del estado de las variables de entrada sino que también depende del estado en que se encontraban las variables de salida al comenzar el proceso.

Pero recientemente vimos el funcionamiento de una ROM o sea una memoria sólo de lectura, en realidad la ROM no guarda un dato durante el tiempo que se le pide sino que, como un sistema combinatorio establece una relación entre las variables de salida y de entrada. La ROM no tiene memoria en el sentido que no se le puede asignar el valor de una variable para que la guarde durante el tiempo necesario.

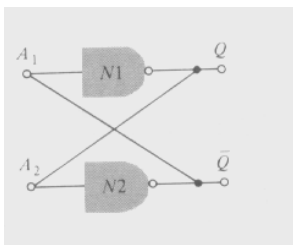
La afirmación genérica que los circuitos combinacionales no poseen memoria se refiere al hecho que el estado de las variables de salida dependen sólo del estado de las variables de entrada y no les importa en absoluto el estado que tenía ese mismo circuito en un instante anterior. Un sistema que guarde un valor de memoria tiene que tener esa capacidad, o sea mantener el estado de la variable después de producida la excitación y aún ante cambios aleatorios en las variables de entrada.

Estudiemos el problema que surge en un teclado: en un instante dado uno presiona una tecla determinada y lo que nos interesa es que de esa situación el procesador interprete que tiene que ingresar en el texto correspondiente la letra deseada. Para esto la tecla deberá estar apretada durante un tiempo mínimo, el tiempo necesario para que se realice dentro de la computadora el paso siguiente. Ahora bien es clásico que cuando uno aprieta una tecla se produce un rebote de la misma (al llegar al final del recorrido), dependiendo de las características mecánicas del teclado y de las características de quien realiza la operación. El resultado es que, en la parte eléctrica se producen una serie de pulsos, ante los cuales deseo que conduzcan siempre al mismo resultado.

Vamos a ver un circuito que me independiza de esta situación. Para ello veamos primero un circuito básico formado por dos puertas NOT, sean éstos N_1 y N_2 (como ya vimos puertas NAND con una sola entrada). La salida de cada puerta se conecta en la entrada de la otra. ¿Qué ocurre en esta situación?, veamos: si introducimos

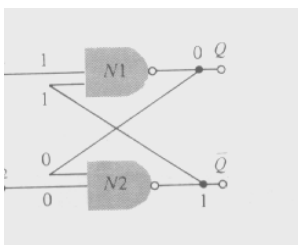
un estado 1 en la entrada A_1 de la puerta N_1 tendremos en su salida un 0; el que a su vez al introducirse en la entrada A_2 de la puerta N_2 me dará como salida correspondiente un 1 que se vuelve a introducir en la entrada A_1 de la puerta N_1 ; esto es refuerza la entrada inicial.

Observemos que si introducimos un 0 como valor inicial en la entrada A_1 de la puerta N_1 , el circuito refuerza este mismo valor. Como conclusión tenemos que éste circuito mantiene dos estados en forma estable. Un circuito de éste tipo se llama circuito biestable o Báscula-Cerrojo o Latch. Llamaremos los estados de salida Q y \bar{Q} a los estados de salida de N_1 y N_2 respectivamente (dado que uno es la inversión del otro)



Hasta ahora no hemos resuelto el problema, pues si por azar la puerta A_1 pasara del estado 1 al estado 0 la salida Q también cambiaría.

Para evitar esto estudiemos el circuito siguiente:



En éste se han cambiado las puertas NOT por puertas NAND, (de fácil fabricación), y se ha mantenido conectada la salida de cada una de ellas a una entrada de la otra.

Analicemos que ocurre en las salidas Q_1 y Q_2 cuando en dicho circuito se introducen los siguientes estados:

- a) $B_1=1$ y $B_2=0$
- b) $B_1=0$ y $B_2=1$

Veamos primero el caso a); si tenemos la entrada B_2 de N_2 es 0, por tratarse de una puerta NAND su salida será siempre 1 cualquiera sea el valor de la otra entrada de N_2 .

L	M	NAND
0	0	1
0	1	1
1	0	1
1	1	0

Ahora bien con la salida de $N_2=1$ en la una de las entradas de N_1 y en la otra entrada, $B_1=1$, la salida de la misma es 0 tenemos como conclusión que

B_1	B_2	Q	$\sim Q$
1	0	0	1

De analizar el caso $B_1=0$ y $B_2=1$ tenemos que por estar B_1 en 0 la salida de N_1 será 1 independientemente de la otra entrada luego las entradas de N_2 serán 1 y $B_2=1$ por lo que su salida será $N_2=0$ tenemos entonces que

B_1	B_2	Q	$\sim Q$
0	1	1	0

O sea que cuando tenemos en las entradas $B_1=1$ y $B_2=0$ guardamos en Q un 0 y en $\sim Q$ un 1 en tanto que si tenemos en $B_1=0$ y en $B_2=1$ guardamos en Q un 1 y en $\sim Q$ un 0.

¿Qué ocurre ahora si estando en uno de los estados estables ya descriptos, accidentalmente cambiamos la entrada que tiene el valor 0 al valor 1?

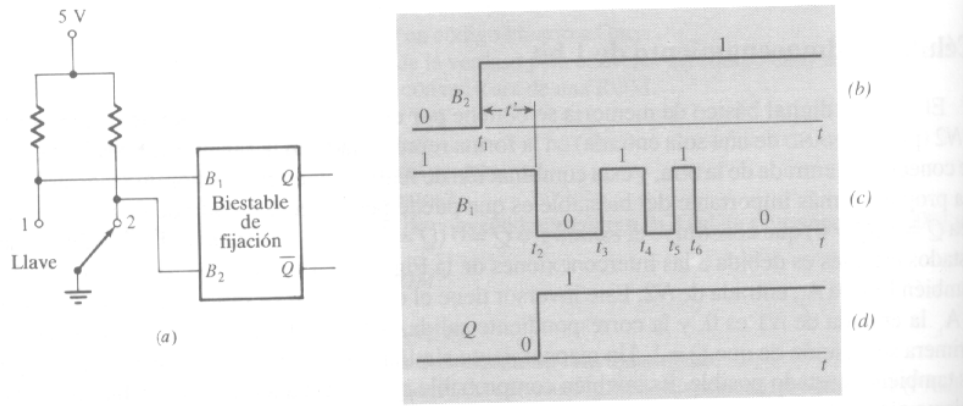
Caso a) si partimos de la situación en que $B_1=1$ y $B_2=0$ la única posibilidad es que B_2 pase del valor 0 al valor 1; en tal caso, (verifiquemos con la tabla NAND) la puerta N_2 no cambia su estado y el sistema no se altera.

B_1	B_2	Q	$\sim Q$
1	0	0	1
1	1	0	1

Caso b) si partimos de la situación en que $B_1=0$ y $B_2=1$ la única posibilidad es que B_1 pase del valor 0 al valor 1; en tal caso la que no se altera es la puerta N_1 (pasó de tener entradas 0 y 0 a valores 1 y 0) con lo que no cambia su estado y el sistema no se altera.

B_1	B_2	Q	$\sim Q$
0	1	1	0
1	1	1	0

Como aplicación práctica veamos en el sistema del teclado el siguiente ejemplo

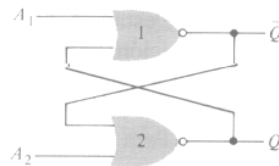


mientras no hay teclas apretadas todas las salidas Q están en estado 0 o sea que las entradas correspondientes están $B_1=1$ máxima tensión (5V) y $B_2=0$ conectada a tierra con la llave inversora. Cuando presiono una tecla, paso la conexión a tierra de la entrada B_2 o sea la quito del estado 0 para pasarlo al estado 1 y completo el paso haciendo que la llave conecte la entrada B_1 a tierra. Alcanzada esta situación el sistema biestable necesita de cierto tiempo mínimo para pasar del estado $Q = 0$ al $Q = 1$, pero si a continuación hay un rebote que me desconecta de tierra a la entrada B_1 durante un lapso dado y la lleva momentáneamente al estado $B_1=1$ la salida Q permanecerá en el estado 1 . Esta situación se mantendrá por mas que B_1 oscile entre por los estados 0 y 1 . En el gráfico anterior se muestra los estados de las entradas como de las salidas como función del tiempo, se ve el paso de B_1 del estado 0 al 1 en el instante t_1 , el paso de la entrada B_2 del estado 1 al 0 y el consecuente pasaje del latch del estado 0 al 1 . Se observa la representación de los sucesivos rebotes de la tecla en los tiempos t_3 y t_5 sin que se altere el estado de Q , evitándose así errores en la salida del teclado.

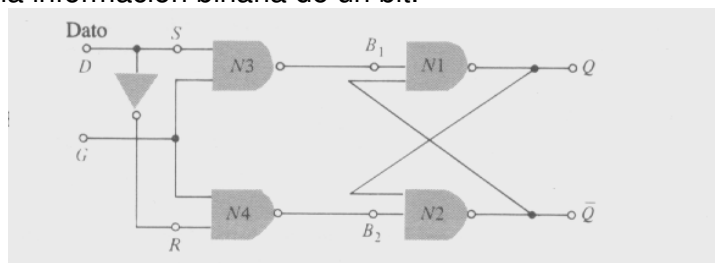
La situación analizada es totalmente simétrica para el pasaje de la llave cuando conmuta en sentido inverso, o sea cuando paso la entrada B_1 al estado 1 y a continuación la entrada B_2 al estado 0 con sucesivos rebotes entre los estados 0 y 1 de la misma.

Es de hacer notar que la situación $B_1=0$ y $B_2=0$ nos conduce a estados en los que las salidas de ambas puertas no resultan ser una la inversa de la otra por lo que no nos conducen a sistemas estables.

También cabe hacer notar que todo este estudio se podía haber hecho con puertas NOR llegando a resultados totalmente equivalentes a los hallados.



Si le agregamos dos puertas NAND a N_1 y a N_2 del biestable original, y una puerta inversora obtendremos el sistema de la figura siguiente que nos servirá para almacenar una información binaria de un bit.



Se puede observar que cuando la entrada \bar{E} (puerta de habilitación) está en estado 1 cualquiera sea el valor del dato de entrada D queda automáticamente almacenado en la salida Q .

como vimos la salida para éste dispositivo biestable responde a la siguiente tabla de verdad

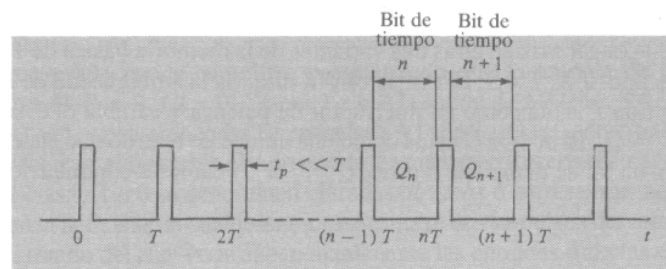
B_1	B_2	Q
0	0	?
0	1	1
1	0	0
1	1	NC

Los circuitos biestables o latch pueden ser de dos tipos según el instante en que pueden cambiar de estado o lo que es lo mismo en el instante en que pueden cambiar el valor de sus salidas. Debido a esta condición los latch se dividen en asincrónicos y sincrónicos o temporales.

a) Serán asincrónicos cuando pueden cambiar de estado en cualquier momento. En tal caso los circuitos ya vistos hasta ahora son latch asincrónicos. Las puertas de entrada serán S y R en el caso de ser puertas (Set-Reset) con o sin puerta de habilitación (\bar{E}) según sea el caso. Habitualmente se construyen conectando entre sí puertas del tipo NOR o NAND.

b) Serán sincrónicos o temporales cuando además de tener una o dos entradas también tienen una puerta de habilitación conectada a un reloj o clock (Ck) que determina el momento en que la/s entrada/s pueden alterar a las salidas (Q y \bar{Q})

El reloj o clock es un circuito que introduce una serie de pulsos con período T en el que habitualmente el lapso reservado para el tiempo en el que el pulso está en el estado 1 llamado t_p es mucho menor que el período del pulso T como se indica en la figura



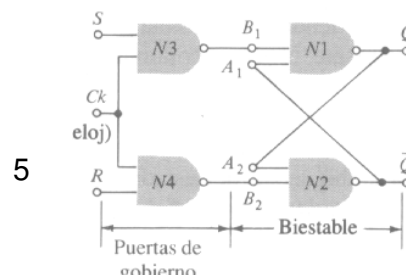
siguiente:

Se destaca en la misma el nombre de los pulsos, al comenzar el período número n -ya habrá transcurrido un tiempo $(n-1)T$ - en ese instante el bit correspondiente al pulso n tendrá un estado dado por la variable Q_n en tanto que un período posterior el estado de la variable estará dado por el valor que adopte Q_{n+1} . De la misma manera los cambios en las variables S y R que ocurran durante el tiempo T_n los llamamos S_n y R_n , normalmente éstos tendrán efecto en un tiempo T_{n+1}

En la figura siguiente vemos un latch S-R temporizado hecho con puertas NAND

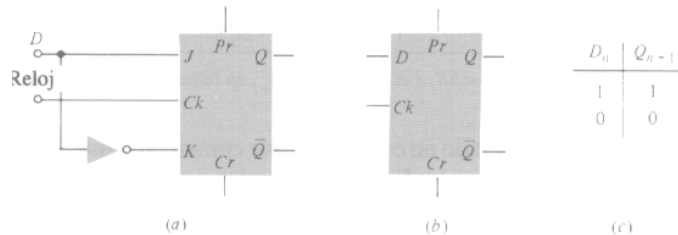
Vemos que se podrá cambiar la entrada de este dispositivo sólo cuando el Ck me habilite a hacerlo pues si Ck es igual a cero las únicas salidas posibles para las puertas NAND son 1 y si tengo ambas iguales a 1 en un dispositivo biestable, la salida Q no se altera con respecto a su último estado.

S_n	R_n	Q_{n+1}
0	0	Q_n
1	0	1
0	1	0
1	1	?



No obstante en el caso en que C_k esté en el estado 1 podrá verificarse la tabla descrita, lógicamente, en la salida Q_{n+1} ahora se producirá una escritura (Set) cuando $S_n=1$ mientras que se producirá un Reset cuando $R_n=1$ a consecuencia de las puertas NAND ante el sistema biestable (al revés que la tabla anterior)

Si se modifica el latch J-K agregándole un inversor antes de la entrada K y unimos la entrada de éste con la entrada J tendremos que una será siempre el complemento de la otra y por lo tanto no existe posibilidad de ambigüedad.



Pues, de acuerdo con la tabla de verdad del latch J-K, tendremos un estado de salida $Q_{n+1}=1$ para el caso en que $D_n=J_n=\sim K_n=1$ análogamente tendremos un estado de salida $Q_{n+1}=0$ para el caso en que $D_n=J_n=\sim K_n=0$ con lo que $Q_{n+1}=D_n$. O sea que la salida Q_{n+1} después del pulso $n+1$ (período T_{n+1}) se iguala al de la entrada D_n antes del impulso (período T_n).

En la figura anterior se muestra la imagen de un latch D hecho a partir de un J-K, su representación y su tabla de verdad

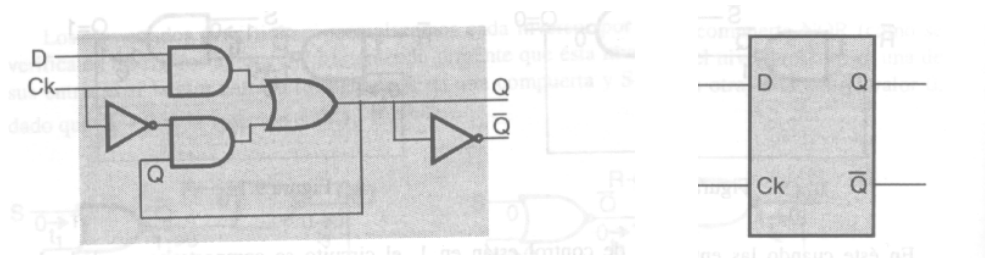
Es de hacer notar que también tiene las entradas de preset y clear para comenzar su acción.

Como dijimos en su presentación el uso de este dispositivo es de guardar el valor de una variable para el pulso siguiente.

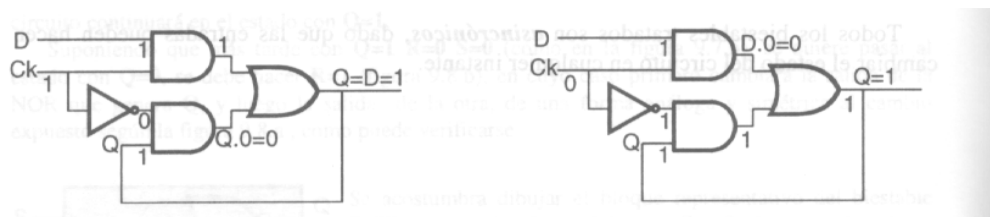
Otra forma de ver el latch D es mediante puertas AND, OR, e inversoras

Que tiene el mismo símbolo que el latch D ya visto hecho a partir del latch J-K

El tiempo establecido para presentar en la entrada D el valor a memorizar está dado



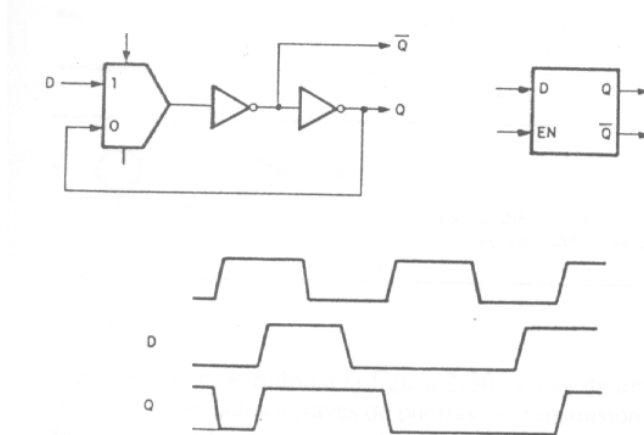
por la duración del nivel alto del clock, el tiempo que debe mantenerse y memorizarse, ese valor quedará memorizado durante el tiempo en que el clock se mantenga en el estado bajo y quedará fijado por la realimentación de Q a la puerta AND que tiene como otra entrada la del clock invertida. Los cambios de nivel de Ck marcan los tiempos en que se pasa de una situación a la otra.



La forma en que opera se representa en el esquema siguiente

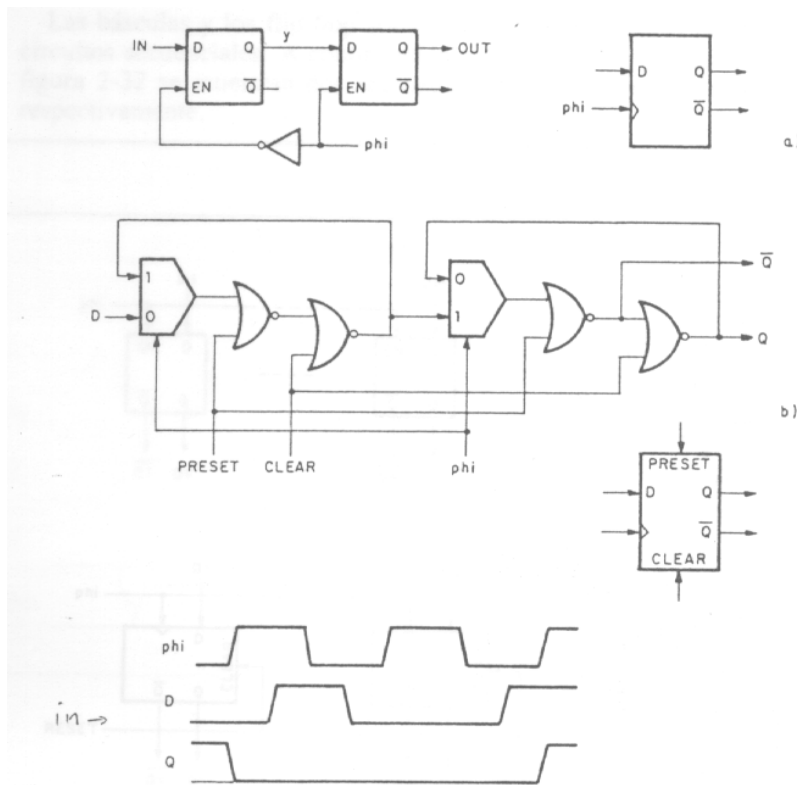
Cuando Ck está en 1 se podrá ingresar por la entrada D el valor a memorizar tanto sea 1 como 0 la puerta AND entregará ese valor a la entrada de la puerta OR que tomará ese valor pues de la otra puerta AND sólo podrá recibir un cero por la condición

de $\sim Ck$ una vez que Q adopte el valor de D , en el intervalo siguiente en el que Ck pasa al valor cero ya no importa el valor que tenga la entrada D pues la puerta OR tendrá como entradas un cero proveniente de la puerta AND que también incluye a D y considerará sólo habrá el valor de la otra puerta AND la que entregará el valor que tenía Q que ahora estará multiplicada por un 1 proveniente del $\sim Ck$



Otra forma de ver el latch D es formada por un multiplexor con una entrada D , otra proveniente de la realimentación habilitadas por el clock Ck (puerta EN en el símbolo) en los gráficos de pulsos se ve que cuando Ck habilita por nivel la entrada de D tendremos en Q el mismo valor hasta que Ck vuelva a habilitar por nivel la entrada siguiente.

Obsérvese que la salida Q cambia su valor, adoptando el que le corresponda, cuando se dan las condiciones adecuadas, independientemente de cual haya sido la última puerta que cambió su valor, esto significa que puede alcanzar el valor que le corresponda tanto cuando la curva del reloj se encuentra en el estado superior desde hace algún tiempo y es habilitado el estado por un ingreso de un valor adecuado en D , como cuando Ck pasa al estado superior y la condición sobre D era anterior.



Nótese la diferencia con la gráfica de curvas anterior del latch D en la que algunas salidas de Q aparecían cuando el nivel de Ck habilitaba y había un cambio en la entrada D . Esta es precisamente la diferencia entre latch y Flip-Flop.

En los Flip-Flop los cambios en las salidas sólo aparecen por flancos en la gráfica del clock; es el reloj quien coordina las salidas.

En tanto que en los latch las salidas aparecerán entonces por nivel, ya sea que el clock haya habilitado el cambio antes que se produjera en las variables o en el caso que el clock habilite un cambio en la salida después que las condiciones hayan sido fijadas, el cambio se producirá sólo después que el clock haya alcanzado su estado superior.

Otra diferencia de hacer notar entre el flip-flop J-K presentado anteriormente y el flip-flop D es la orientación del inversor, y consecuentemente, el estado de habilitación del clock al maestro o al esclavo durante el período T_n .

En el flip-flop J-K hemos puesto que el clock habilita primero al bloque Master mientras el Slave está deshabilitado, cuando el clock pasa al estado negativo deshabilita al bloque maestro en tanto que por culpa del inversor se habilita el bloque esclavo y en consecuencia la salida Q . Esto que significa que la salida de éste flip-flop aparece en el pasaje del clock de un estado alto a un estado bajo, por lo que se llama un flanco descendiente.

Por el contrario en el flip-flop D el clock entrega un pulso positivo al bloque Slave por lo que habilita la salida en la primera parte del período T_n . En tanto que en la segunda parte del período (que como vimos anteriormente es la más larga) está habilitado el bloque maestro. De modo tal que durante el lapso en que el clock está en su estado bajo se producen los cambios necesarios en el bloque maestro para encontrar la salida buscada, luego cuando el clock pasa a su estado alto aparecerán los cambios mostrados en la salida del bloque maestro al bloque esclavo, en el período siguiente T_{n+1} . El hecho que la aparición de los resultados en la salida del flip-flop ocurre cuando el clock pasa del estado bajo al estado alto del mismo, determina que la salida de este dispositivo es por flanco ascendente.

Además del S-R existen otros tipos de dispositivos que almacenan un bit de memoria básica, comercialmente se los conoce como J-K, T y D. El primero elimina la ambigüedad en la tabla de verdad del S-R, el T actúa como un interruptor de palanca y cambia el estado de la salida a cada impulso del reloj logrando que $Q_{n+1} = \sim Q_n$. El tipo D funciona como una unidad de retardo que hace que la salida Q siga a la entrada D pero con un bit de tiempo de retraso $Q_{n+1} = Q_n$.

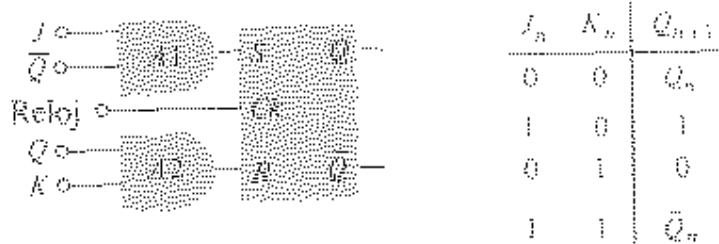
Éste dispositivo se construye poniendo dos puertas AND (A_1 y A_2) frente a cada una de las entradas S y R de un latch S-R. Una entrada de cada una de las puertas será la entrada J y de la otra entrada K . La otra entrada de cada puerta AND es realimentada por las salidas $\sim Q$ y Q respectivamente de modo que en la salida de la puerta A_1 tendremos $S = J \cdot \sim Q$ y a la salida de la puerta A_2 tendremos $R = K \cdot Q$.

Es evidente que cuando Ck habilita el funcionamiento de la puerta S-R tendremos para cada una de las entradas J y K dos valores posibles; tendremos entonces cuatro estados de la puerta J-K no obstante dado que a priori no conocemos los valores posibles para las salidas Q y $\sim Q$ transformamos la tabla en ocho posibilidades asignando 2 posibilidades más por el estado de Q .

Columna	1	2	3	4	5	6	7	8
Fila	J_n	K_n	Q_n	$\sim Q_n$	S_n	R_n	Q_{n+1}	
1	0	0	0	1	0	0	Q_n	Q_n
2	0	0	1	0	0	0	Q_n	
3	1	0	0	1	1	0	1	1
4	1	0	1	0	0	0	Q_n	

5	0	1	0	1	0	0	Q_n	0
6	0	1	1	0	0	1	0	
7	1	1	0	1	1	0	1	
8	1	1	1	0	0	1	0	

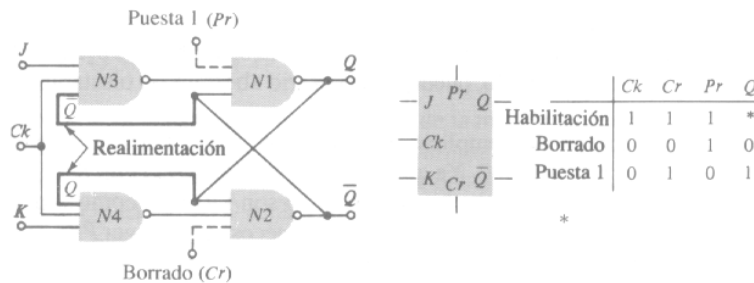
La columna 8 de la tabla anterior se obtiene a partir de la columna 7 observemos que en las filas 1 y 2 los valores de S y R coinciden en dar como salida el



último estado de Q. En tanto que para la línea 4 hay que obtener el valor de Q a partir del valor propuesto (1) situación análoga a la línea 5 con Q=0. los valores encontrados para las líneas 7 y 8 coinciden con los valores propuestos para ~Q

La tabla de verdad ya presentada para el latch J-K coincide con las columnas 1, 2 y 8 de la última tabla, no era extraño pensar que la abigüedad que generaba la situación de entrada 1-1 podría anularse con las salidas Q y ~Q complementando la entrada con puertas AND dado que una de ellas es la inversa de la otra.

Usando éste mismo concepto se puede hacer un latch J-K a partir de la S-R hecho con puertas NAND con tan sólo agregar una entrada adicional a las puertas de entrada y conectarles la realimentación de las salidas Q y ~Q directamente como se muestra en la figura sin tener en cuenta las líneas de trazo punteado.



OJO ESTE DISPOSITIVO NO PUEDE ANDAR!!!!!! PORQUE A LA SALIDA DE n3 Y n4 ESTARÁN SIEMPRE Q Y Q NEGADO CON LO QUE N1 Y N2 ANDARÁN MAL

Dado que en estos casos la salida hallada por la tabla de verdad nos informa los estados posibles cuando el Ck está habilitando en realidad no sabemos que valor nos indica la primera vez que el Ck pasa de 0 a 1.

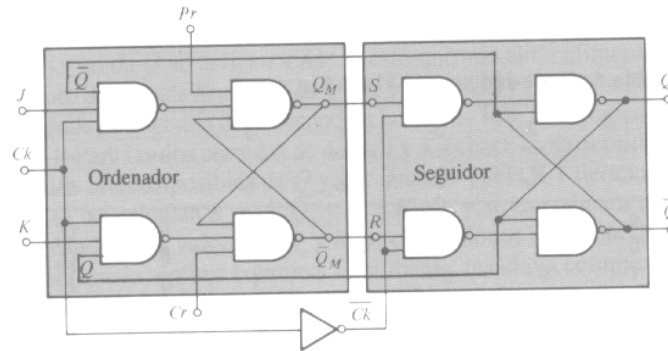
Por ello es necesario ajustar el latch al comenzar a trabajar, por ejemplo asegurándose que Q valga 0 cuando el Ck vale 0. (La llamamos operación de borrado)

Esto se logra haciendo que las entradas extras Pr = 1 (Preset) y Cr=0 (Clear) cuando el reloj esté en Ck=0. Veamos como: cuando Cr=0 la salida de N2 está en estado 1 para cualquier valor de las otras entradas, luego ~Q=1; al tener la entrada Ck=0 la salida de N3 deberá ser 1 tenemos entonces que las entradas de N1 son 1 (por N3) 1 (por ~Q) y 1 (por Pr) con lo que logramos que Q=0.

Debido a la duración de los pulsos del Ck , al retardo en la realimentación por Q y \bar{Q} es posible que este tipo de circuitos comience a oscilar. Para resolver este problema se creó la lógica de Maestro-esclavo (Master-Slave) o como figura en algunas traducciones (Ordenador-Seguidor)

En ésta lógica el reloj alimenta una etapa de entrada mientras deshabilita la etapa siguiente. Y da origen al oficialmente llamado Flip-Flop.

El flip-flop tipo J-K Master-Slave en realidad está compuesto por dos latches S-R en cascada con realimentación desde la salida del segundo a la entrada del primero.

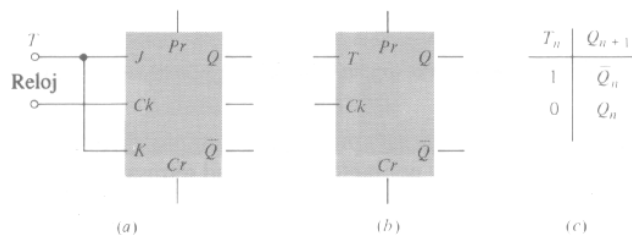


El concepto de éstas ideas pasa por eliminar la indeterminación del estado 1 - 1 con la realimentación de las puertas Q y \bar{Q} y eliminar los estados de oscilación asegurando que cuando una etapa cambia, la otra no puede hacerlo. A continuación se muestra una figura de dicha configuración

Veamos el funcionamiento cuando el clock está en uno, puedo comenzar los cambios en el block Master si hago que $Pr=1$ y $Cr=1$ habilitando el valor de Q_M , al pasar el clock al estado cero se habilita la segunda etapa (Slave) con los valores de entrada sin posibilidad de ambigüedad pues como entrada ya tengo Q_M y \bar{Q}_M , y con la imposibilidad de una oscilación.

Dado que Q_M sigue la lógica del latch J-K es importante tener bien definidos los tiempos en los que los valores de J y K mantengan la señal buscada para lograr su efecto durante el tiempo de habilitación del clock de lo contrario se induciría a un error.

El flip flop tipo T se puede obtener a partir de un flip flop tipo J-K (Maestro Esclavo) con una entrada simultanea en ambas puertas dado que el flip-flop adopta valores \bar{Q}_n y Q_n cuando las entradas J y K son 1-1 y Q_n respectivamente y atendiendo que en un sistema maestro esclavo cada etapa está activada cuando la otra no, tendremos que cuando $T=1$ la salida estará cambiando cada vez que el clock habilita



Resumen

Como resultado del estudio encontramos que puede haber biestables tipo latch y flip-flop. En general en los circuitos biestables primarios se define la existencia de dispositivos capaces de guardar un bit; en el S-R adecuamos que la salida Q tenga el

¹ En la bibliografía de origen español se encontrará esta disposición como ordenador -seguidor no así en el resto de la bibliografía , en particular la de origen norteamericana.

mismo estado que la entrada S (Set =1 Q=1) en los latch J-K, D y T resolvemos el tema de la ambigüedad para completar la tabla de verdad, en el primero completamos la tabla de verdad con un $\sim Q$ en el D reducimos las entradas posibles y en el T tenemos las otras dos salidas que teníamos en el J-K lo que nos permite construir un interruptor. Por último, en los dispositivos flip-flop se eliminan las inestabilidades propias de las diferencias constructivas de los transistores que se manifiestan cuando se introducen lazos de realimentación esto es que los dispositivos comiencen a oscilar o a cambiar de estado aleatoriamente.

Queda claro que los latch son dispositivos que habitualmente se activan por un nivel de un clock u otro dispositivo en tanto que los flip-flop sólo se activan por un flanco de un clock (que por lo general es el positivo, para que se generen los cambios en el master en el período T_n y se entreguen en el período siguiente T_{n+1})

A manera de resumen presentamos la siguiente tabla de verdad de los flip-flop

S-R			J-K			D		T		Entradas directas			
S_n	R_n	Q_{n+1}	J_n	K_n	Q_{n+1}	D_n	Q_{n+1}	T_n	Q_{n+1}	Ck	Cr	Pr	Q
0	0	Q_n	0	0	Q_n	1	1	1	$\sim Q_n$	0	1	0	1
1	0	1	1	0	1	0	0	0	Q_n	0	0	1	0
0	1	0	0	1	1					1	1	1	*
1	1	?	1	1	$\sim Q_n$								

Cabe acotar que los valores de las entradas directas corresponden al caso de trabajar con puertas NAND. Si estudiamos los estados de Preset & Clear del Flip-Flop D de la página 8 nos encontramos que la tabla correspondiente para las entradas directas con puertas NOR son:

Entradas directas			
Ck	Cr	Pr	Q
0	0	1	1
0	1	0	0
1	0	0	*