

Semana # 8 – Flips Flops, RS, JK, D, T / Semisumador y sumador completo

BIESTABLES O FLIP FLOPS

Hoy, hemos aprendido que todo circuito en electrónica emplea datos binarios para poder funcionar correctamente. Que estos datos binarios se emplean en señales digitales y que los valores de estos datos varían dependiendo del funcionamiento de las compuertas lógicas que se empleen, esto es lo mismo que decir que los datos de salida cambian de acuerdo con sus entradas.

Los Flips Flops o Biestables se definen como multivibradores que se emplean para memorizar información o en otras palabras para mantener un dato o serie de datos en un lugar sin que varíe, hasta que lo necesitemos.

Las memorias entonces y como ya sabemos, pueden almacenar datos evitando que se pierdan y pueden variar su contenido a nuestra voluntad. El Flip Flop es el corazón de una memoria y es un circuito basado en una combinación de compuertas lógicas.

Un Flip Flop puede tener múltiples entradas pero siempre sólo dos salidas:

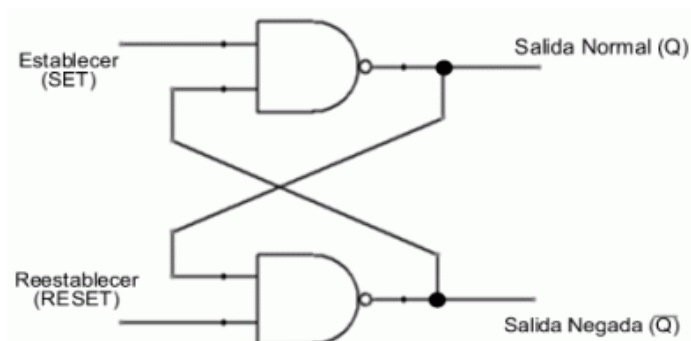
- Q (salida normal)
- \bar{Q} (salida 1 o 0)

Salida	Símbolo	Estado 1	Estado 2	Información
Normal	Q	1	0	Estado 1 "Establecer" (SET) Estado 2 "Reestablecer" (RESET)
Invertida o Negada	\bar{Q}	0	1	

Flip Flop con compuertas NAND y NOR

Estos podrían considerarse los circuitos más sencillos y básicos de un Flip Flop. Se pueden construir con dos compuertas NAND o dos compuertas NOR cada uno con dos entradas.

El circuito o la conexión se verían de la siguiente manera para la versión con compuertas NAND:



La tabla que se muestra a continuación muestra la opción cuando ambas entradas están ACTIVAS lo cual quiere decir que el FF se encuentra en estado de “reposo”. Una vez que se cambian los valores de entrada por valores bajos, es cuando el Flip Flop comienza a actuar:

Entrada	Estado	Q	-Q
Establecer	1	1	0
Reestablecer	1	0	1

Ahora veamos la tabla que resulta cuando hay cambios en las entradas y por lo tanto provoca cambios en las salidas:

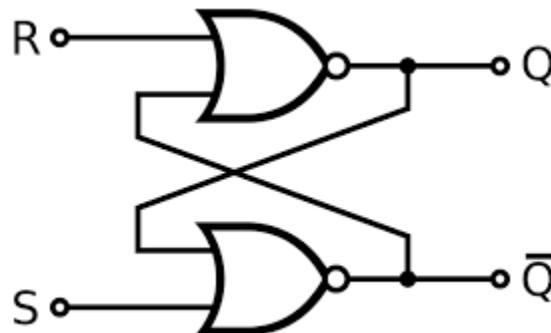
Cambios	Entrada	Estado	Q	-Q
Estado ACTUAL	SET	1	Sin cambio	Sin cambio
	RESET	1		
Pulso BAJO a RESET	SET	1	0	1
	RESET	0		
Pulso BAJO a SET	SET	0	1	0
	RESET	1		
Pulso BAJO a AMBOS	SET	0	1	1
	RESET	0		

Podríamos decir que la tabla de la verdad se vería de la siguiente manera:

Set	Reset	Q	-Q
1	1	Sin cambio	Sin cambio
1	0	0	1
0	1	1	0
0	0	1	1

Ahora bien, con la compuerta **NOR** vendría a ser la misma lógica, sólo que las salidas vendrían a ser completamente diferentes.

El circuito se vería de la siguiente manera:



Y la tabla de la verdad quedaría de la siguiente manera:

Set	Reset	Q	-Q
1	1	No definido	No definido
1	0	0	1
0	1	1	0
0	0	Q	-Q

Notas: Con el estado 0:0 nada cambia y entonces el FF memoriza.

Algunos usos que se le pueden dar a estos registros básicos NAND y NOR son:

- Sistemas de Encendido (ON) y Apagado (OFF).
- Sensores de movimientos mecánicos.
- Control digital sobre otros circuitos.

Control por Reloj (CLOCK)

Los dos circuitos que vimos anteriormente se conocen con el nombre de Flip Flops de Registros Básicos y son aquellos que responden de manera predecible a las dos variables de entrada, pero ¿es posible hacer que el registro actúe cuando nosotros lo queramos y no al momento de cambiar los valores de entrada?

Esto puede conseguirse con señales de RELOJ, pero antes debemos repasar que todo sistema digital puede operar de dos maneras:

- De forma **ASINCRONA**, que es cuando las salidas cambian de manera automática siguiendo las órdenes de entrada.
- De forma **SINCRONA**, que es cuando las salidas cambian siguiendo las órdenes de entrada, pero únicamente cuando una señal de control llamada RELOJ (CLOCK, CLK, CP) es aplicado al registro.

Biestable RS

Es un dispositivo de almacenamiento de dos estados y cuyas entradas al ser activadas permiten:

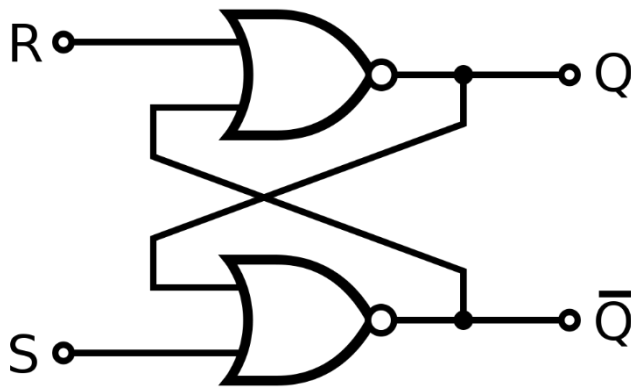
- R o borrado (Reset) cuando es 0.
- S o grabado (Set) cuando es 1.

Si ninguna entrada se activa, el Flip Flops permanece con el estado anterior de borrado o grabado.

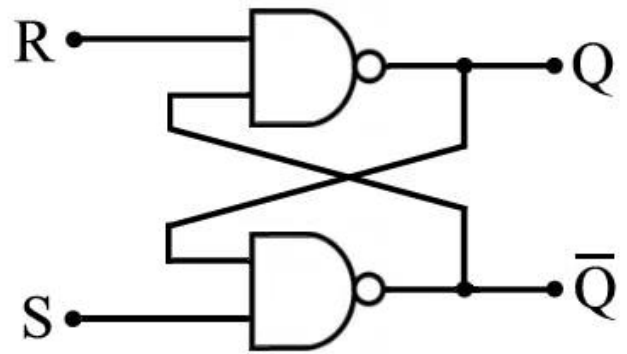
Biestable RS Asíncrono

Se puede construir empleando dos compuertas NAND or NOR, muy similar a los ejemplos anteriores:

En este caso, el biestable no debería de poseer ambas entradas activas o altas o ambas entradas bajas ya que esto provocará que las salidas ambas posean el mismo valor y esto provocará que al desactivar ambas entradas no se pueda determinar el último valor de la salida.



a) Flip Flop RS con NOR



b) Flip Flop RS con NAND

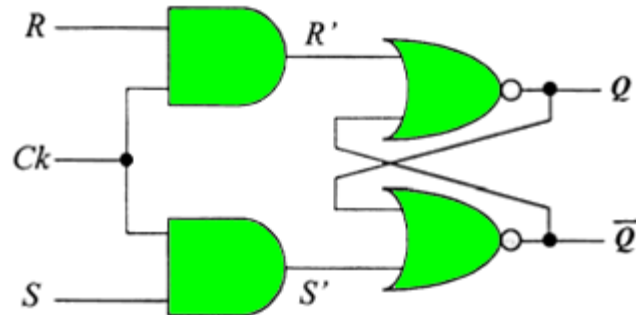
La tabla de la verdad quedaría de la siguiente manera:

R	S	Q (NOR)	Q (NAND)
0	0	q	N. D.
0	1	1	0
1	0	0	1
1	1	N. D.	Q

q = Estado de memoria, ND = No Deseado

Biestable RS Síncrono

El siguiente es un ejemplo de un Flip Flop síncrono:

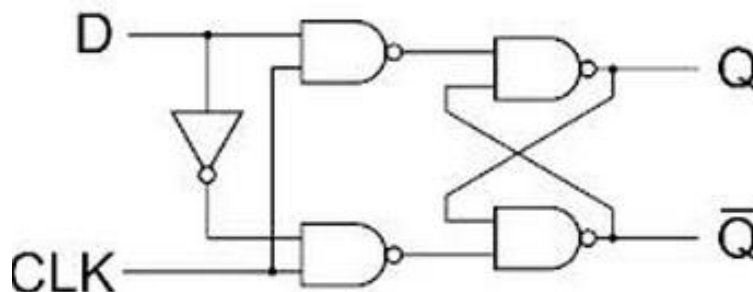


La tabla de la verdad se vería de la siguiente manera:

C	R	S	Q (NOR)
0	X	X	q
1	0	0	q
1	0	1	1
1	1	0	0
1	1	1	N. D.
X=no importa			

Biestable D (Data o Relay)

Se emplea cuando se requiere almacenar un único dato (1 o 0). Es un Flip Flop que posee una sola entrada y que puede combinarse con un Flip Flop de tipo RS, por lo que se considera una mejora. Veamos el siguiente diagrama:



En este caso podemos observar que el Flip Flop posee una única entrada D y luego se combina con un Flip Flop tipo RS con compuertas NAND. Podemos observar también que este biestable emplea una señal de reloj por lo tanto es de tipo síncrono.

La tabla de la verdad quedaría de la siguiente manera:

Q	D	Q (t+1)
0	0	0
0	1	1
1	0	0
1	1	1

Flip Flop JK

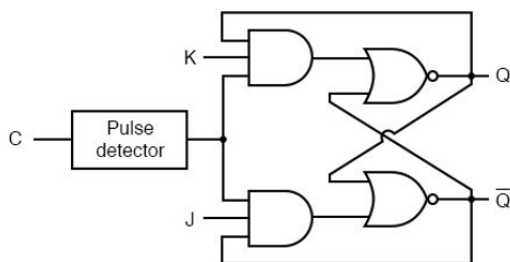
Este Flip Flop es de los más utilizados. Posee las mismas condiciones SET y RESET de los biestables RS y el mismo estado de permanencia, con la única diferencia de que el FF JK no posee condiciones no válidas (no deseadas) como sí sucede en el RS.

Las entradas se definen como J y K y cuando son activadas permiten:

- J: grabar (SET), valor 1 o alto en la salida.
- K: borrar (RESET), valor 0 o bajo en la salida.

En caso de que ambas entradas no se activen, el Flip Flop permanece en el estado que poseía tras la última operación de borrado o grabado. Ahora bien, si ambas entradas se activan a la vez (estado 1:1 del FF RS), este estado ya no será no deseado, la salida adquirirá el estado contrario al que tenía.

En la siguiente imagen podemos ver el diagrama, el mapa de Karnaugh y la tabla de verdad:

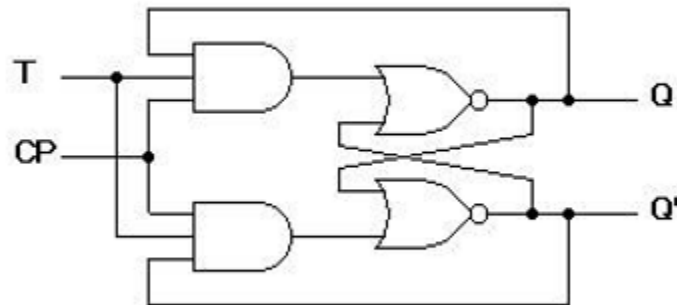


CK	J	K	Q	Q'
0	X	X	Q	Q'
1	0	0	Q	Q'
1	0	1	0	1
1	1	0	1	0
1	1	1	Q'	Q

Flip Flop T (Toggle)

Es un circuito del JK pero refinado a una sola entrada como veremos más adelante en el diagrama. Como vimos en el título T viene de la palabra en inglés **Toggle** que vendría a ser **conmutar** en español.

El Flip Flop T cambia de estado cada vez que la entrada de reloj se dispara y la entrada T está en nivel alto, si T está en nivel bajo, el biestable retiene el nivel previo.



SEMISUMADOR Y SUMADOR COMPLETO

Semisumador

El semisumador es un circuito que admite dos entradas y que genera dos salidas las cuales son:

- 1.El bit que representa el resultado de la suma de los dos bits de entrada.
- 2.Un bit que representa el acarreo generado por la suma.

Esto es algo que ya ustedes dominan, repasemos:

$$0 + 0 = \underline{\quad}$$

$$0 + 1 = \underline{\quad}$$

$$1 + 0 = \underline{\quad}$$

$$1 + 1 = \underline{\quad}$$

Entonces lo que sucedía con el $1 + 1 = 10$, es que ese uno del resultado tenemos que acarrearlo. La tabla de la verdad para un semisumador es la siguiente:

A	B	Co	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Para construir el circuito, podemos utilizar una compuerta XOR para el valor de suma y una compuerta AND para la operación de acarreo. ¿Tiene sentido? Repasemos esas tablas de verdad y veamos lo que tenemos arriba:

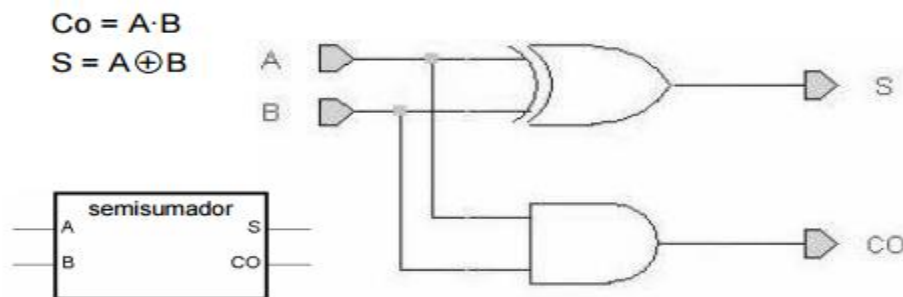
Tabla AND

p	q	$p \wedge q$
V	V	V
V	F	F
F	V	F
F	F	F

Tabla XOR

p	q	$p \text{ XOR } q$
V	V	F
V	F	V
F	V	V
F	F	F

¿Ya lo vieron? Entonces si partimos de esta premisa, el circuito quedaría de la siguiente manera:



Sumador completo

El sumador completo a diferencia del semisumador, es que éste admite un valor de acarreo de entrada, por lo que posee tres valores de entrada en lugar de dos.

La tabla de la verdad quedaría de la siguiente manera:

Ci	A	B	Co	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

El circuito para implementar un sumador completo sería el siguiente:

