

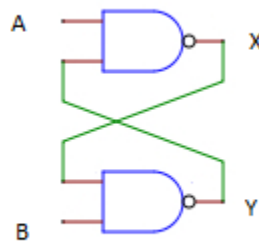
“Esperá que lo anoto, sino me olvido”

Además de hacer operaciones con datos, como se mostró en la práctica 5, para poder hacer cálculos es necesario tener un mecanismo que almacene datos, una MEMORIA. Para esto el circuito no puede depender solamente del estado de sus entradas, sino que debe depender de estados anteriores, es decir de la SECUENCIA de datos, de allí el nombre de Lógica Secuencial.

LATCH

El elemento mas simple que hace esto se denomina LATCH (cerrojo, pestillo), y es un ejemplo de realimentación positiva.

El circuito que se muestra cumple con la siguiente tabla de verdad:



A	B	X	Y
0	0	1	1
0	1	1	0
1	0	0	1
1	1	0 1	1 0

TABLA 1: Tabla de verdad de un LATCH

Los primeros 3 estados se determinan fácilmente si uno nota que la tabla de verdad de una compuerta NAND es:

A	B	Z
0	0	1
0	1	1
1	0	1
1	1	0

TABLA 2: Tabla de verdad de una compuerta NAND

NOTE que si en una compuerta **NAND ALGUNA** de las entradas en **CERO**, la salida es **UNO**.

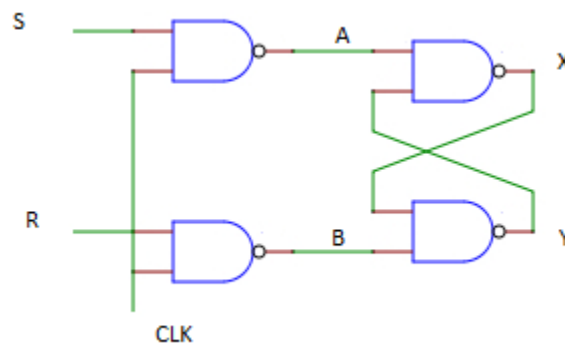
Sin embargo, si **AMBAS** entradas son **UNO**, existen dos posibles estados, es decir que es un estado biestable (marcado en rojo en la tabla 1). El sistema PUEDE estar en cualquiera de los dos estados y uno no tiene forma de saber en cual de los dos estará si no conoce la HISTORIA, es decir que pasó antes.

Con esto ya puedo guardar el dato que quiera. Si quiero guardar un 1 en X entonces pongo A=0 y B=1. Si quiero guardar un 0 en X entonces pongo A=1 y B=0.

Si ahora quiero que el sistema se quede en el estado en que lo puse, pongo en las entradas A=1 y B=1. **El estado A = 0 y B = 0 no es útil para lo que se quiere hacer con estos circuitos, y se eliminará luego evitando esa posibilidad.**

CLOCKED FLIP FLOP

Un LATCH es un dispositivo de memoria que trabaja de manera asincrónica, es decir, que no tiene sincronización temporal. Cuando un LATCH trabaja con sincronización temporal se lo denomina FLIP FLOP (FF).



El Clock (CLK) puede valer 0 o 1. Si **Clock = 0**, entonces por la tabla 2 A=1 y B=1, sin importar el estado de S y R. Por lo tanto, **si Clock = 0 el sistema mantiene el estado anterior.**

Si **Clock = 1**, entonces usando las tablas 1 y 2:

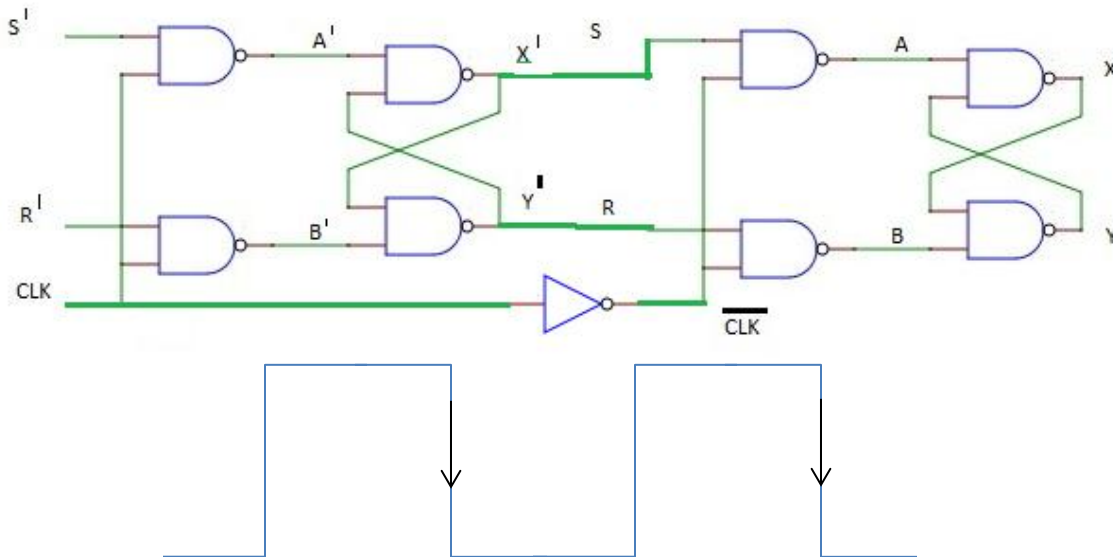
S	R	A	B	X	Y
0	0	1	1	0	1
0	1	1	0	0	1
1	0	0	1	1	0
1	1	0	0	1	1

TABLA 3: Tabla de verdad de un FLIP FLOP SR

Es decir, en este circuito los cambios SOLAMENTE ocurren cuando el Clock está en **UNO**. Las letras **S** y **R** vienen de **SET** y **RESET**, dado que **S** pone $X = 1$ mientras que **R** pone $X = 0$.

FLIP FLOP MASTER – SLAVE

El FLIP Flop Master – Slave es una sofisticación del anterior. Se conectan dos etapas iguales, una detrás de la otra, pero el Clock de la segunda es el de la primera NEGADO (Invertido). Por lo visto en el punto anterior, cada etapa solamente puede cambiar si su Clock está en **UNO**. Eso quiere decir que el estado que se encuentre en la primera mitad (llamada Master) justo en el instante en que el Clock pasa de **UNO** a **CERO** será el que tendrá algún efecto en la segunda etapa (llamada Slave). Todo lo que pase en otro momento es irrelevante, porque será transferido a la salida.



SEÑAL DE CLOCK: Solo importa lo que pasa en los flancos descendentes

Flip Flop Tipo D

Un tipo de Flip Flop Master – Slave particular es el que tiene $R' = \overline{S'}$. Este Flip Flop se llama D.

Comprueba que la tabla de verdad de un Flip Flop D es:

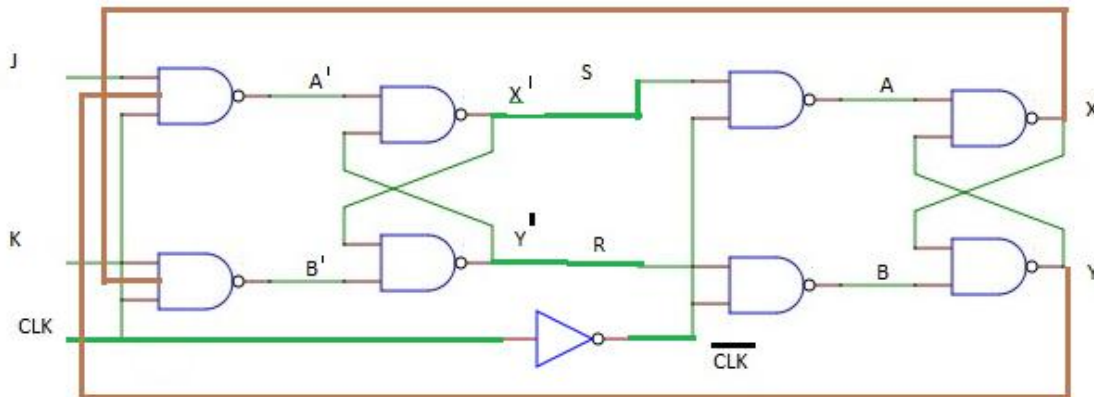
D	X	Y
0	0	1
1	1	0

TABLA 4: Tabla de verdad de un Flip Flop D

donde D es el estado de entrada en el momento del flanco descendente de la señal. Note que $Y = \bar{X}$. **Note que en este Flip Flop ya se eliminó el estado indeseado que ponía $X = 1$ e $Y = 1$.**

Flip Flop JK

El Flip Flop JK es otro tipo de Flip Flop Master – Slave que tiene realimentaciones de las salidas con las entradas. Las compuertas NAND de las entradas son de 3 entradas, donde una es el Clock, otra es J o K y la otra es Y o X, respectivamente.



Note que sigue siendo válido que solamente importa lo que pasa en el flanco descendente del Clock. Por lo tanto voy a analizar que pasa si las entradas tienen un determinado valor en ese momento.

Existen 4 posibles estados a la entrada:

- a) $J = 0, K = 0$
- b) $J = 0, K = 1$
- c) $J = 1, K = 0$
- d) $J = 1, K = 1$

a) Si $J = 0, K = 0$, como están conectados a compuertas NAND, no importa que pasa con las demás entradas, $A' = 1$ y $B' = 1$. Pero usando la Tabla 1 X' e Y' mantendrán el estado que tenían. Pero si X' e Y' no cambian, entonces no hay forma de que cambie X e Y.

J	K	A'	B'	X_{N+1}	Y_{N+1}
0	0	1	1	X_N	Y_N

b) Si $J = 0, K = 1$, entonces $A' = 1$. El valor de B' dependerá de cuanto valga X, dado que está conectada a la entrada. Dado que $CLK = 1$ y $K = 1, B' = \bar{X}_N$. Entonces tengo:

J	K	X_N	A'	B'	X_{N+1}	Y_{N+1}
0	1	1	1	0	0	1
		0	1	1	0	1

Pero usando la Tabla 1, $A' = 1$, $B' = 0$ pone $X' = 0$ e $Y' = 1$, mientras que $A' = 1$, $B' = 1$ mantiene el estado anterior, que en ese caso era $X' = 0$, $Y' = 1$. Por lo tanto, no importa que estado tenía a la salida, $J = 0$, $K = 1$ pone $X_{N+1} = 0$, $Y_{N+1} = 1$

- c) Si $J = 1$, $K = 0$, entonces $B' = 1$. El análisis es el mismo del circuito anterior por su simetría. El valor de A' dependerá de cuanto valga Y , dado que está conectada a la entrada. Dado que $CLK = 1$ y $J = 1$, $A' = \overline{X_N}$. Entonces tengo:

J	K	Y_N	A'	B'	X_{N+1}	Y_{N+1}
1	0	1	0	1	1	0
		0	1	1	1	0

Pero usando la Tabla 1, $A' = 0$, $B' = 1$ pone $X' = 1$ e $Y' = 0$, mientras que $A' = 1$, $B' = 1$ mantiene el estado anterior, que en ese caso era $X = 1$, $Y = 0$. Por lo tanto, no importa que estado tenía a la salida, $J = 1$, $K = 0$ pone $X_{N+1} = 1$, $Y_{N+1} = 0$

- d) Si $J = 1$, $K = 1$, entonces estoy en la situación donde cada una de las compuertas NAND de las entradas tiene dos de sus patas en UNO, ya sea J y Clock o K y Clock. Por lo tanto $A' = Y_N$ y $B' = X_N$. Por lo tanto

J	K	A'	B'	X_{N+1}	Y_{N+1}
1	1	Y_N	X_N	Y_N	X_N

Pero entonces resulta que los casos a) y b) no fijan X_N e Y_N ; o lo mantienen o los intercambian. Y los casos c) y d) ponen $Y_N = \overline{X_N}$, condición que entonces se verifica siempre. Entonces, en este Flip Flop el estado no deseado de la Tabla 1 también fue eliminado.

En la práctica 6 usará estos Flip Flops y además circuitos armados con Flip Flops, como los 7490 y 7493, que son CONTADORES. Un contador es un componente que le suma una unidad en cada pulso de Clock al número que tenía antes. Por lo tanto debe tener memoria de ese número para poder realizar la operación.

También usará una plaqueta con un Display de 7 segmentos, que muestra el número binario de 4 bits BCD como un número digital. El circuito que se usa es un 74HC4511. Este circuito, además de tener una memoria que permite “guardar” el número que tiene a la entrada, y “congelarlo”, debe hacer una operación de conversión para prender los LEDS que correspondan en función de cuál es el número binario de 4 bits BCD que está en la entrada. Por ejemplo, si las cuatro entradas están en CERO, se deben prender TODOS los leds salvo el del medio, mientras que el 1000 debe prender todos los leds, dado que equivale a un 8 y el 0001 debe prender solamente los leds de la derecha.



Este es un buen ejemplo de uso de lógica combinacional que aprendió en la práctica 5. Vea la Tabla de verdad en la hoja de datos del 74HC4511. ¿Puede escribirla en forma de ecuaciones?