

Ejercicio #1

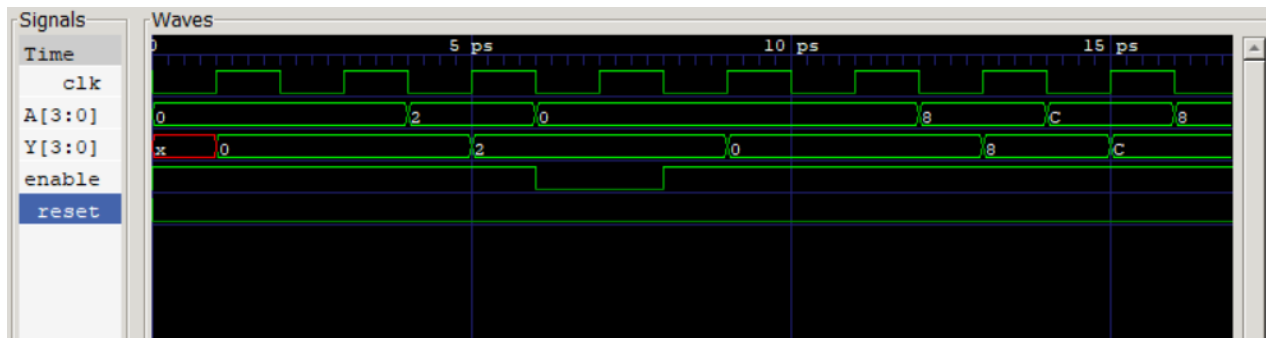


Gráfico 1. FlipFlop de 4bits

En este ejercicio se realizó un flipflop hasta de 4 bits. El funcionamiento de los flipflops de varios bits consiste en la manipulación de cada un bit independiente del resto. Cada bit cuenta con su flipflop el cual indicará a que dato cambiará después del flanco. Como se puede observar en la grafica 2, donde se muestra un flipflop de 2bits.

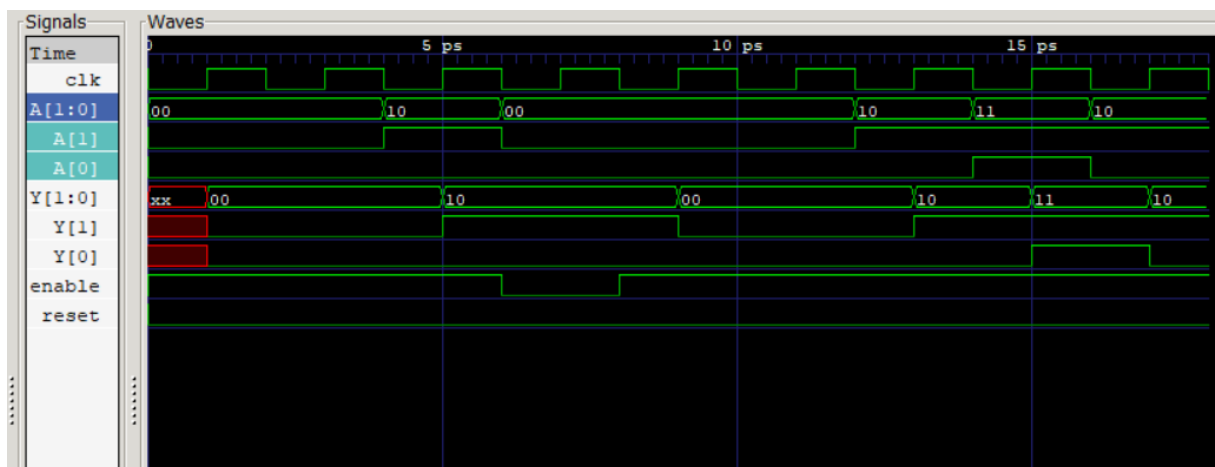


Gráfico 2. FlipFlop de 2 bits

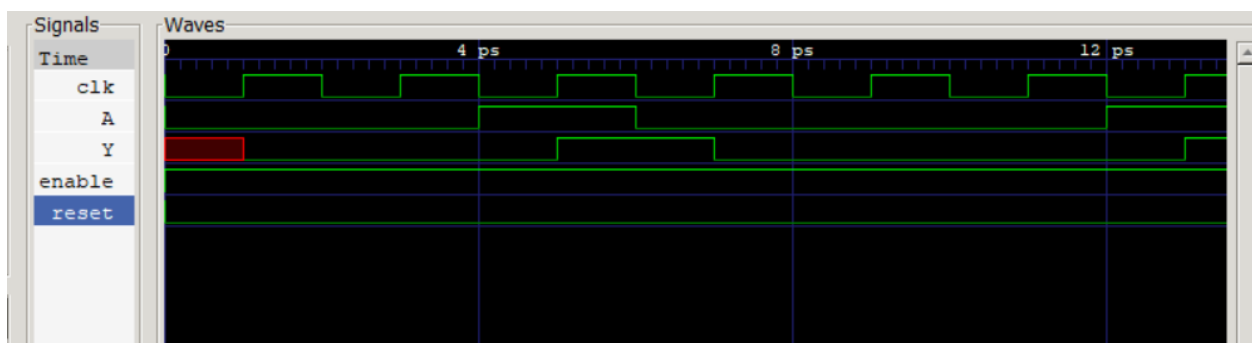


Gráfico 3. FlipFlop sencillo

Ejercicio 2



Gráfico 4. FlipFlop T

En este ejercicio se realizó un FlipFlop tipo Toggle. Este tipo de FF no cuenta con una entrada ya que este se retroalimenta. Realmente lo único que hace es invertir lo que se encuentra en Q. Como se puede observar en el gráfico es que el periodo del FF es el doble de la del reloj, esto ocurre debido a que cuenta con un flipflop interno que retrasa el cambio hasta el próximo flanco. Si se desactiva este deja de contar.

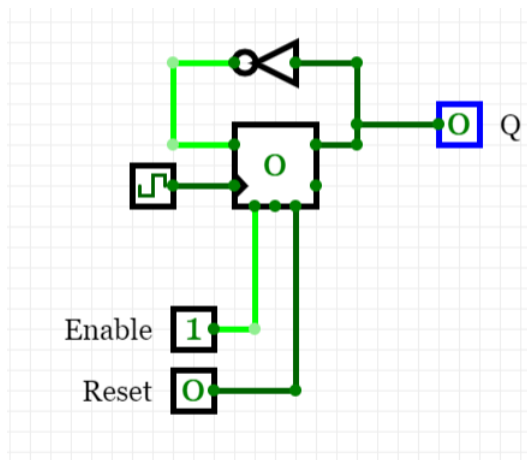


Ilustración 1. FlipFlop T

Ejercicio 3

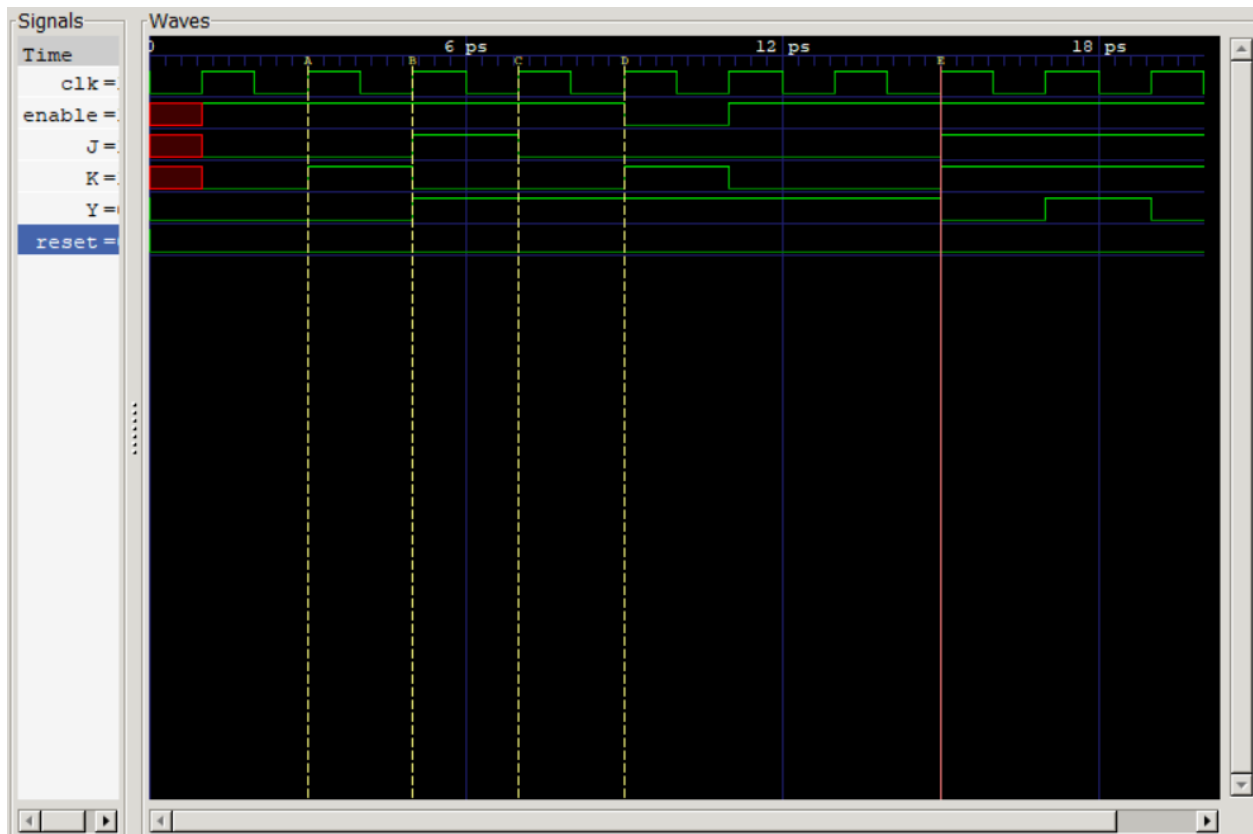


Gráfico 5. FlipFlop JK

En este ejercicio Podemos observar un flip flop JK este fliplop cuenta con 4 operaciones que se puede observar en la gráfica. Al probar $J=0$ y $K=1$ podemos observar en el marcador A que $Y=0$, luego al utilizar $J=1$ y $K=0$ podemos observar que $Y=1$ en el marcador B, en el marcador C observamos como $J=0$ y $K=0$ no tienen efecto en Y. En el marcador D se pudo verificar el funcionamiento del enable, ya que si hubiese estado encendido Y sería igual a 0. Por último, podemos observar $J=1$ y $K=1$ en el marcador E donde este alterna el resultado en cada flanco de reloj. En la siguiente imagen podemos observar el circuito en CircuitVerse

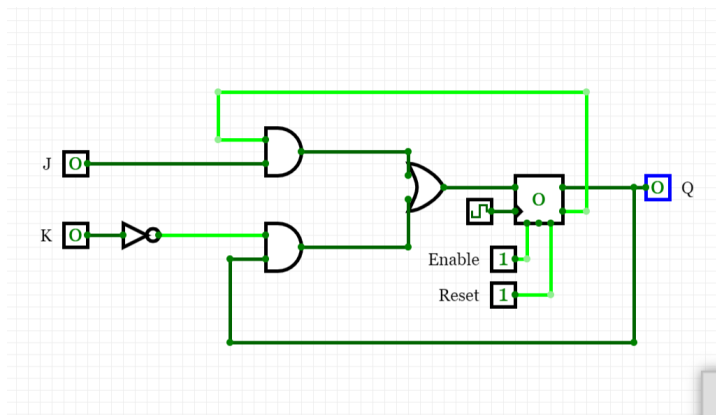


Ilustración 2. FlipFlop JK Circuitverse

Ejercicio #4



Gráfico 6. Buffer Tri-estado de 4bits

En este ejercicio se creó un buffer triestado de 4bits, como se puede observar en el gráfico anterior. El funcionamiento de este componente es bastante sencillo. Cuando se encuentra enabled, el dispositivo deja pasar los datos de A a Y, pero al tener el enable en 0 se coloca en alta impedancia.

Ejercicio #5

En este ejercicio se implemento una memoria en modo de cases. En cada case se contiene una serie de datos de 13bits

ADDRESS	DATA
0000000	1000000001000
1000000	1000000001000
0000101	0100000001000
0000111	0100000001000
0000001	1000000001000
0000011	1000000001000
0001101	1000000001000
0001001	0100000001000
0001101	1000000001000
0010001	0001001000010
0011001	1001001100000
0100001	0011010000010
0100011	0011010000010
0101001	0011010000100
0110001	1011010100000
0111001	1000000111000
1000011	0100000001000
1000001	1000000001000
1001011	1000000001000
1001001	0100000001000
1010001	0011011000010
1011001	1011011100000
1100001	0100000001000
1101001	0000000001001
1110001	0011100000010
1111001	1011100100000

Tabla 1. Datos

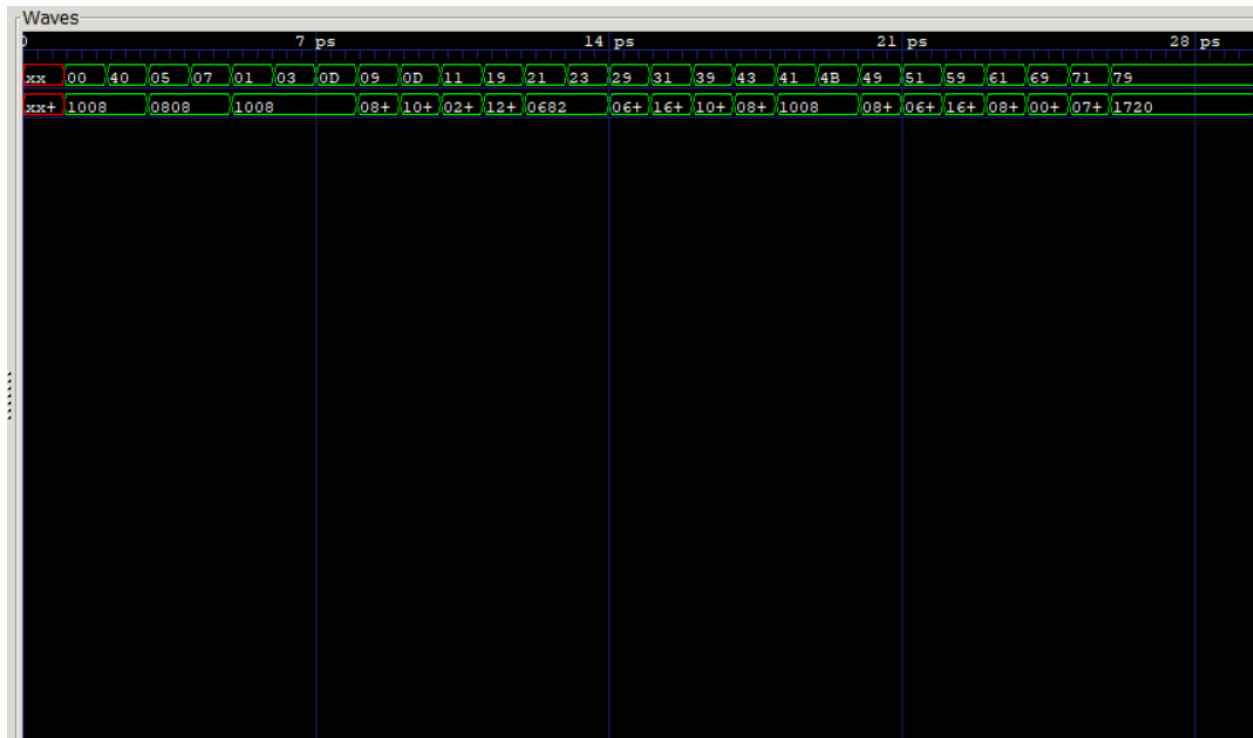


Gráfico 7. Memoria de la table

Links:

<https://circuitverse.org/users/29347/projects/lab-09-08f1ad43-461a-4f49-9b45-c52b2b664f71>

<https://github.com/Men19673/Electronica-Digital/tree/master/Lab%2009>