

Ejercicio 1:

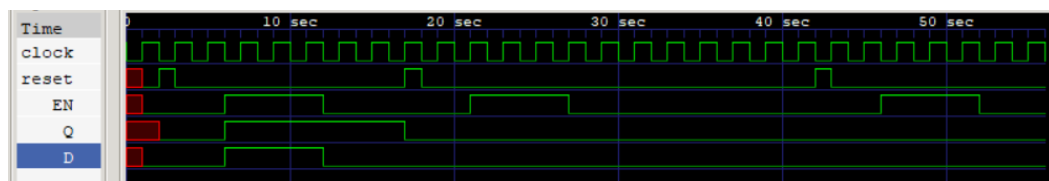
En este ejercicio se realizo un flip flop D de un bit con enable, reset en un bloque always. Posteriormente, se concatenaron 2 de estos módulos para hacer un flip flop D de 2 bits y 4 para hacer un flip flop de 4 bits.

Tablas:

Flip Flop D de un bit:

FLIP FLOP D 1 BIT			
clk	en	D	Q
1	0	0	x
0	0	0	0
1	0	0	0
0	0	0	0
1	0	0	0
0	1	1	1
1	1	1	1
0	1	1	1
1	1	1	1
0	1	1	1
1	1	1	1
0	0	0	1
1	0	0	1
0	0	0	1
1	0	0	1

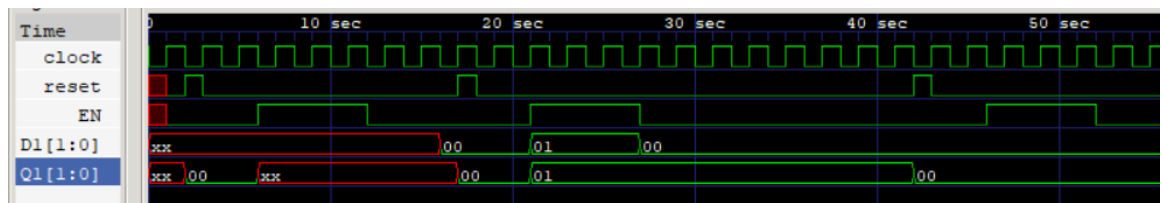
Diagrama de timing:



Flip Flop D de dos bits:

FLIP FLOP D 2 BIT			
clk	en	D	Q
0	0	00	xx
1	0	00	00
0	0	00	00
1	0	00	00
0	0	00	00
1	1	01	01
0	1	01	01
1	1	01	01
0	1	01	01
1	1	01	01
0	1	01	01
1	0	00	01
0	0	00	01
1	0	00	01
0	0	00	01
1	0	00	01
0	0	00	01
1	0	00	01
0	0	00	01
1	0	00	01
0	0	00	01
1	0	00	01
0	0	00	01
1	0	00	01
0	0	00	01

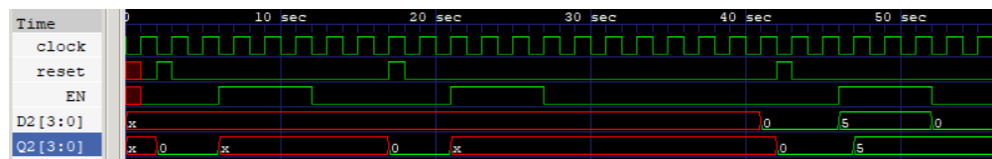
Diagrama de timing:



Flip Flop D de cuatro bits:

FLIP FLOP D 4 BIT			
clk	en	D	Q
1	0	0000	xxxx
0	0	0000	0000
1	0	0000	0000
0	0	0000	0000
1	0	0000	0000
0	1	0101	0000
1	1	0101	0101
0	1	0101	0101
1	1	0101	0101
0	1	0101	0101
1	1	0101	0101
0	0	0000	0101
1	0	0000	0101
0	0	0000	0101
1	0	0000	0101
0	0	0000	0101

Diagrama de timing:



Ejercicio2:

En este ejercicio se realizo un toggle flip flop, que alterna el valor de la salida con cada flanco de reloj. Este es esencialmente un flip flop D con Q! conectado a D.

Tabla:

FLIP FLOP D		
clk	en	Q
1	0	x
0	1	0
1	1	1
0	1	1
1	1	0
0	1	0
1	1	1
0	1	1
1	1	0
0	1	0
1	1	1
0	1	1
1	1	0
0	1	0

Circuitverse:

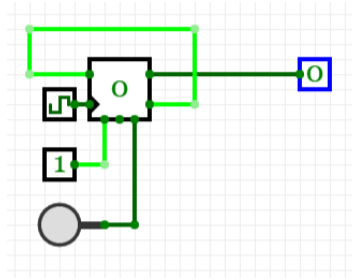
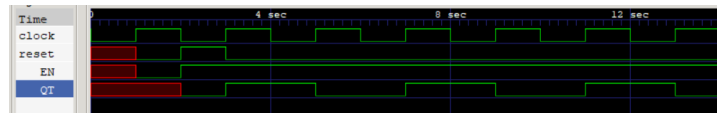


Diagrama de timing:



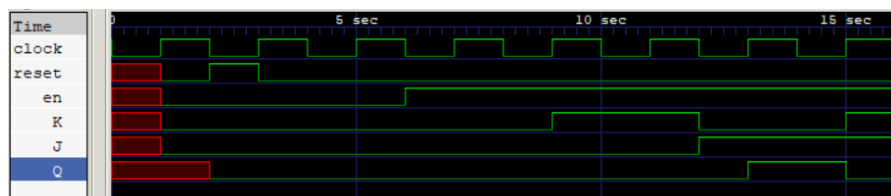
Ejercicio 3:

En este ejercicio se construyo un Flip Flop JK que tiene de entradas J, K, enable y reset. Este consiste en una nube combinacional conectada a un flip flop D, donde la entrada del FFD es la salida de la nube combinacional y dicha nube tiene como entradas, la salida del FFD, J y K.

Tabla:

FLIP FLOP JK				
clk	en	J	K	Q
1	0	0	0	x
0	0	0	0	0
1	0	0	0	0
0	0	0	0	0
1	0	0	0	0
0	1	0	0	0
1	1	0	0	0
0	1	0	0	0
1	1	0	1	0
0	1	0	1	0
1	1	0	1	0
0	1	1	0	0
1	1	1	0	1
0	1	1	0	1
1	1	1	1	0
0	1	1	1	0

Diagrama de Timing:



Ecuaciones booleanas:

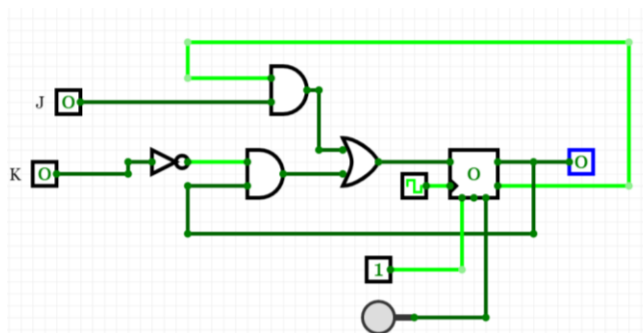
Logic Friday

Función	Inputs	Outputs	True	False	DC	PI	Gates
D	3	1	4	4	0	2	Not mapped

J	K	Qprev	=>	D
X	0	1		1
1	X	0		1

Entered by truthtable:
 $D = J' K' Q_{prev} + J K' Q_{prev}' + J K' Q_{prev} + J K Q_{prev}'$;
 Minimized:
 $D = K' Q_{prev} + J Q_{prev}'$;

Circuitverse:



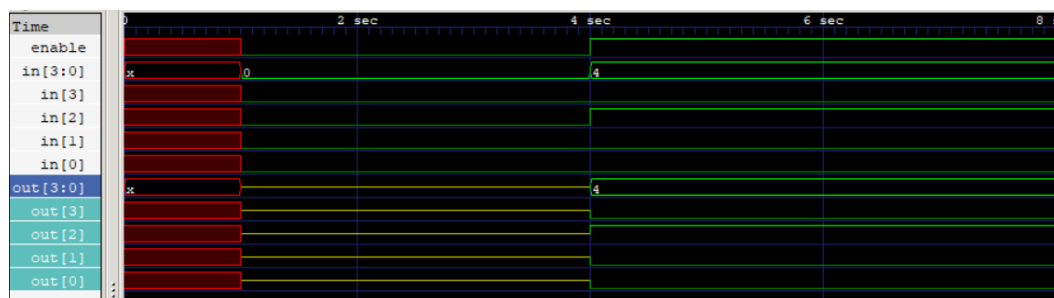
Ejercicio 4:

En este ejercicio se construyo un buffer triestado. Este elemento funciona de la siguiente manera, cuando el enable esta encendido, deja pasar a la salida lo que este en la entrada y cuando el enable esta en 0, la salida esta en alta impedancia.

Tabla:

IN	enable	out
0000	0	zzzz
0100	1	0100

Diagrama de Timing:

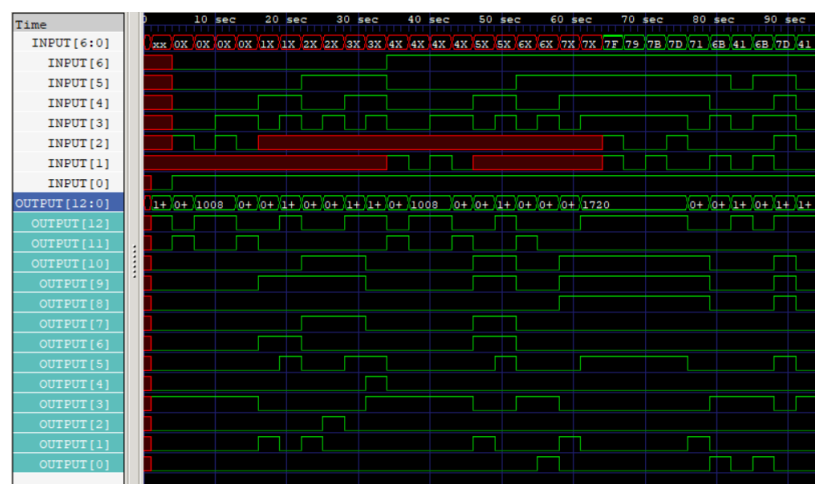


En este ejercicio se implemento la tabla de verdad de la codificación de un nibbler en forma de memoria ROM implementada con cases.

Tabla:

MICROCODE	
INPUT	OUTPUT
xxxxxxx0	1000000001000
00001x1	0100000001000
00000x1	1000000001000
00011x1	1000000001000
00010x1	0100000001000
0010xx1	0001001000010
0011xx1	1001001100000
0100xx1	0011010000010
0101xx1	0011010000100
0110xx1	1011010100000
0111xx1	1000000111000
1000x11	0100000001000
1000x01	1000000001000
1001x11	1000000001000
1001x01	0100000001000
1010xx1	0011011000010
1011xx1	1011011100000
1100xx1	0100000001000
1101xx1	0000000001001
1110xx1	0011100000010
1111xx1	1011100100000
1111111	1011100100000
1111001	1011100100000
1111011	1011100100000
1111101	1011100100000
1110001	0011100000010
1101011	0000000001001
1000001	1000000001000
1101011	0000000001001
1111101	1011100100000
1000001	1000000001000

Diagrama de timing:



Link del repositorio: https://github.com/gil19443/Digital_1.git

