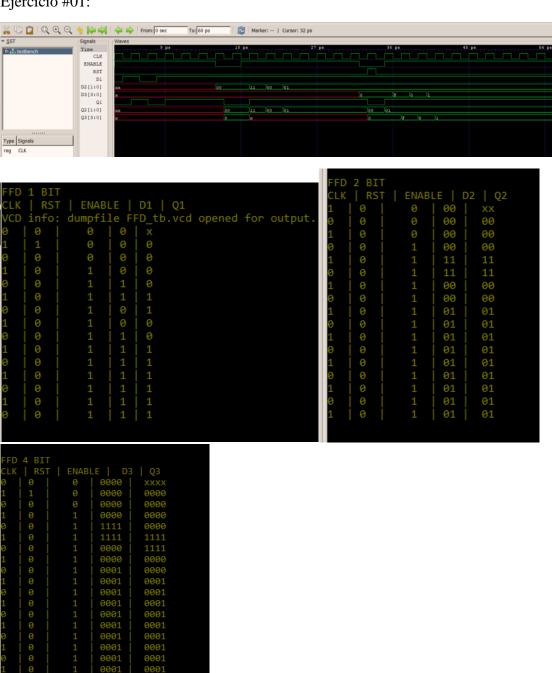
Universidad del Valle de Guatemala

Digital 1

Rodrigo García 19085

Laboratorio #09

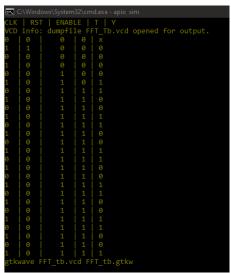
Ejercicio #01:



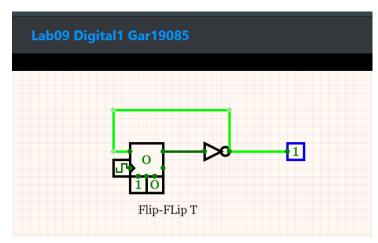
Se creo un modulo para el flip flip tipo D en el cual se indica en que momento sucede una acción dependiendo del flanco de reloj, se establecieron los parámetros de funcionamiento del flip flip para su funcionamiento con el enable y el reset. Luego para implementar el Flip Flip de 2 bits solo se llamo al modulo del FF de 1 bit, luego para el Flip Flop de 4 bits se hizo lo mismo solo que se llamo dos veces al modulo del Flip Flop de 2 bits, el cual también puede realizarse llamando 4 veces al modulo del Flip Flop de 1 bit.

Ejercicio #02:

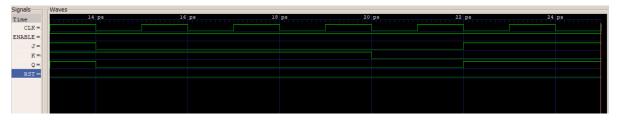


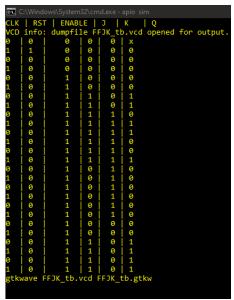


Se creo un modulo para un flip flop de 1 bit, para así implementar el Flip-Flop T en el cual solo se agrega una compuerta not para que así este pueda invertir la salida en cada flanco de reloj.

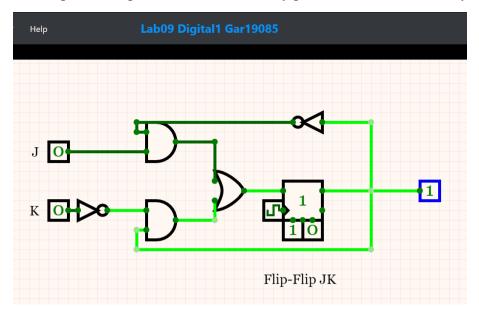


Ejercicio #03:





Se creo un modulo para un flip flop de 1 bit, para así implementar el Flip-Flop JK el cuál funciona por medio de una nube combinacional conformada por dos nots, dos ands y 1 or. Ya que este depende del estado futuro y presente de la salida, de J y de K.



Ejercicio #04:

```
AC DPS[3:0]

OUT (3:0)

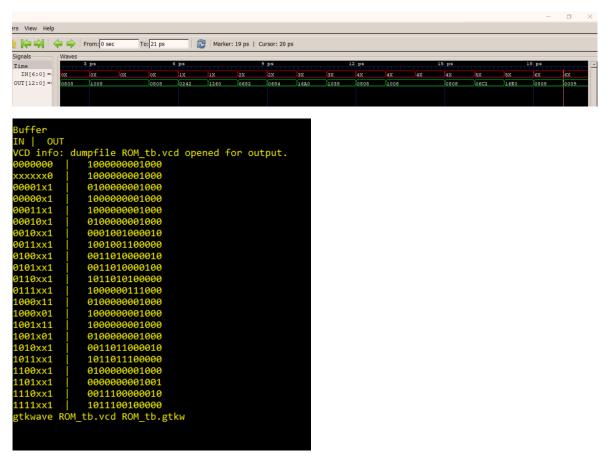
Buffer A | AC | OUT | VCD | OUTPUT=Buffer_tb | C:/Users/ro |

VVD info: dumpfile Buffer_tb.vcd opened for output.

00000 | 0 | zzzz |
0100 | 1 | 0100 |
0101 | 1 | 0100 |
0101 | 1 | 0101 |
0101 gtkwave Buffer_tb.vcd Buffer_tb.gtkw
```

Para implementar el Buffer Tri-Estado se utilizo el comando Case, para que el input que habilita o deja pasar la entrada al buffer tenga indicadas sus instrucciones para dejar pasar la información del input o dejarla como alta impedancia.

Ejercicio #05:



Para realizar el Rom se indico el tamaño del input y el tamaño del output para poder realizar las operaciones indicadas por la tabla, por medio del uso de Case en el modulo.