

Link repositorio: <https://github.com/mon19379/DIGITAL-Montufar-19379.git>

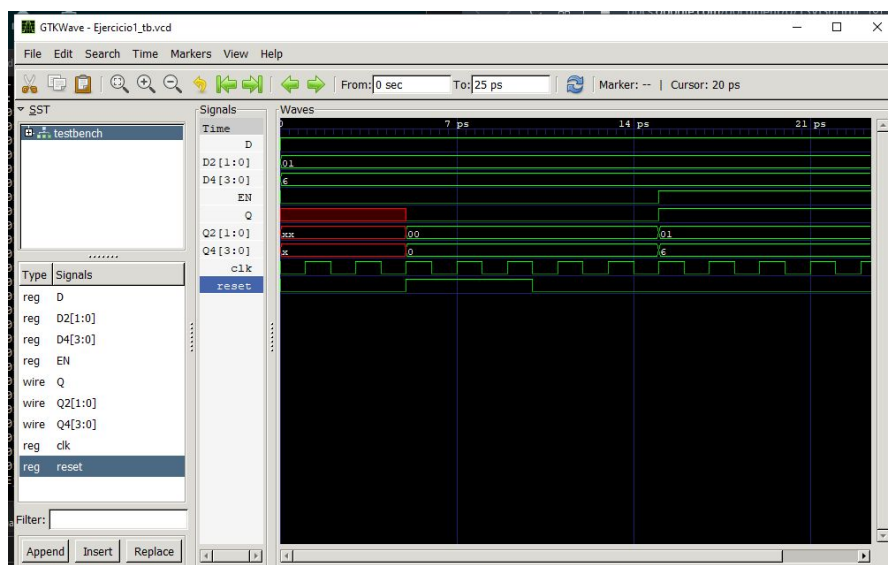
Este ejercicio consistía en implementar un FlipFlop tipo D de 1, 2 y 4 bits. Para lograr esto, se implementó primero el FF de 1 bit, en donde se definían las entradas y las salidas de este. Luego se establecía que este funcionara cuando existiera un flanco positivo tanto en el clk como en el reset. Luego se definen las condicionales que establecen que si reset estaba activo, todos los bits se vuelven 0, de lo contrario, si el EN esta activo, los valores de entrada y salida serán los mismos. Para lograr hacer los otros FF solo se define el módulo con sus respectivas entradas y salidas y se llamaba el módulo del FF de un bit la cantidad de veces necesaria.

```
C:\Windows\System32\cmd.exe - apio sim
```

```
VCD info: dumpfile Ejercicio1_tb.vcd opened for output.
```

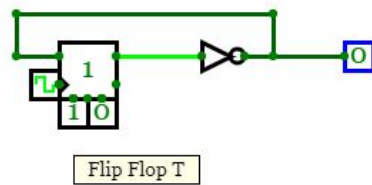
0 0 0 1 01 0110	x xx xxxx
1 0 0 1 01 0110	x xx xxxx
0 0 0 1 01 0110	x xx xxxx
1 0 0 1 01 0110	x xx xxxx
0 0 0 1 01 0110	x xx xxxx
1 1 0 1 01 0110	0 00 0000
0 1 0 1 01 0110	0 00 0000
1 1 0 1 01 0110	0 00 0000
0 1 0 1 01 0110	0 00 0000
1 1 0 1 01 0110	0 00 0000
0 0 0 1 01 0110	0 00 0000
1 0 0 1 01 0110	0 00 0000
0 0 0 1 01 0110	0 00 0000
1 0 0 1 01 0110	0 00 0000
0 0 0 1 01 0110	0 00 0000
1 0 1 1 01 0110	1 01 0110
0 0 1 1 01 0110	1 01 0110
1 0 1 1 01 0110	1 01 0110
0 0 1 1 01 0110	1 01 0110
1 0 1 1 01 0110	1 01 0110
0 0 1 1 01 0110	1 01 0110
1 0 1 1 01 0110	1 01 0110
0 0 1 1 01 0110	1 01 0110
1 0 1 1 01 0110	1 01 0110
0 0 1 1 01 0110	1 01 0110
1 0 1 1 01 0110	1 01 0110

```
gtkwave Ejercicio1_tb.vcd Ejercicio1_tb.gtkw
```

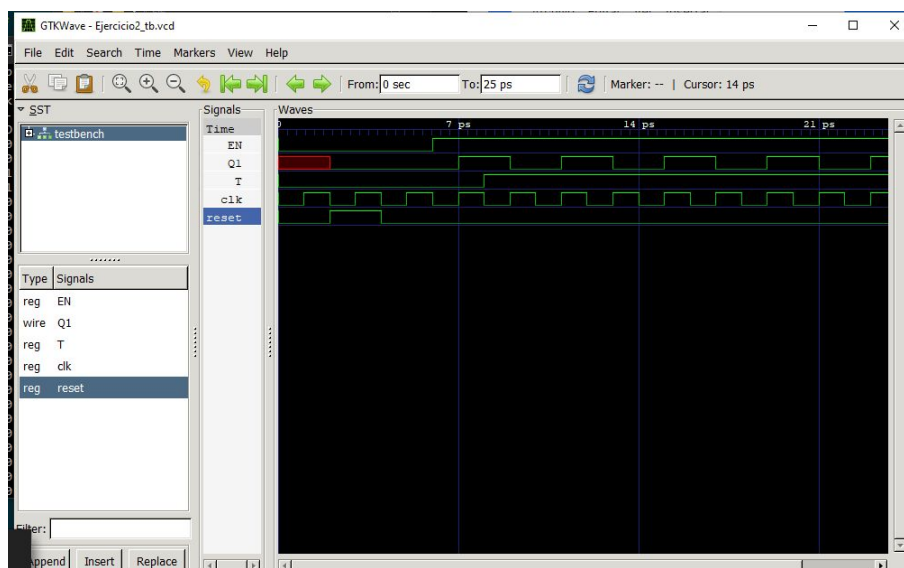


Ejercicio 2

Este ejercicio consistía en implementar un FlipFlop tipo T de 1 bit. Para lograr esto, se implementó primero el FF de 1 bit, en donde se definían las entradas y las salidas de este. Luego se establecía que este funcionara cuando existiera un flanco positivo tanto en el clk como en el reset. Luego se definen las condicionales que establecen que si reset estaba activo, todos los bits se vuelven 0, de lo contrario, si el EN esta activo, los valores de entrada y salida serán los mismos. Para lograr hacer el FF tipo T se utilizó el FF de un bit y se le agregó un not a la salida del FF y esta salida fue conectada a la entrada del FF.

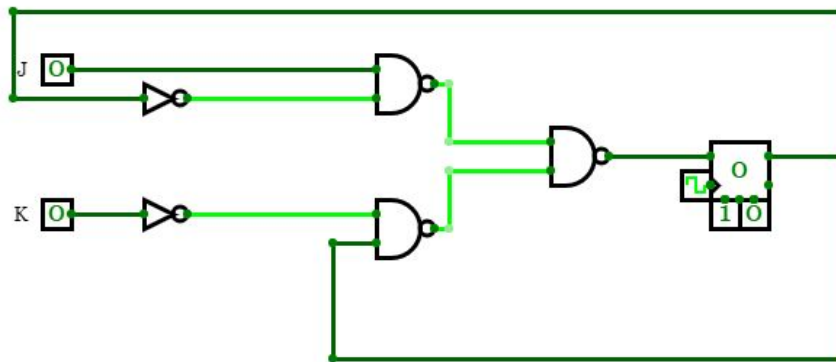


```
C:\Windows\System32\cmd.exe - apio sim
vvp Ejercicio2_tb.out
ejercicio 2
clk RST EN | T | Q1
-----|-----
VCD info: dumpfile Ejercicio2_tb.vcd opened for output.
0 0 0 0 | x
1 0 0 0 | x
0 1 0 0 | 0
1 1 0 0 | 0
0 0 0 0 | 0
1 0 0 0 | 0
0 0 1 0 | 0
1 0 1 0 | 1
0 0 1 1 | 1
1 0 1 1 | 0
0 0 1 1 | 0
1 0 1 1 | 1
0 0 1 1 | 1
1 0 1 1 | 0
0 0 1 1 | 0
1 0 1 1 | 1
0 0 1 1 | 1
1 0 1 1 | 0
0 0 1 1 | 0
1 0 1 1 | 1
0 0 1 1 | 1
1 0 1 1 | 0
0 0 1 1 | 0
1 0 1 1 | 1
0 0 1 1 | 1
1 0 1 1 | 0
0 0 1 1 | 0
1 0 1 1 | 1
0 0 1 1 | 1
1 0 1 1 | 0
0 0 1 1 | 0
1 0 1 1 | 1
0 0 1 1 | 1
```



Ejercicio 3

Este ejercicio consistía en implementar un FlipFlop JK de 1 bit. Para lograr esto, se implementó primero el FF de 1 bit, en donde se definían las entradas y las salidas de este. Luego se establecía que este funcionara cuando existiera un flanco positivo tanto en el clk como en el reset. Luego se definen las condicionales que establecen que si reset estaba activo, todos los bits se vuelven 0, de lo contrario, si el EN esta activo, los valores de entrada y salida serán los mismos. Para lograr el funcionamiento del FFJK se creó una nube combinacional con nots y nands. Esta nube combinacional dependía de las entradas J y K y de la salida del FF para poder lograr las salidas esperadas.



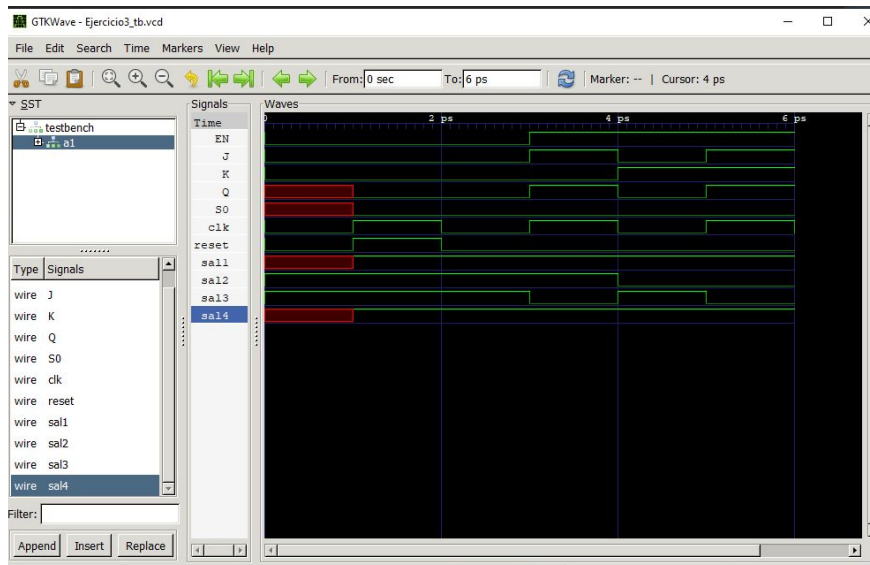
Flip Flop JK

```
C:\Windows\System32\cmd.exe - apio sim
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\franc\Documents\GitHub\DIGITAL-Mont-far-193879\LAB9\Ej3>apio clean
---> WARNING: no PCF file found (.pcf)
Removed Ejercicio3_tb.out
Removed Ejercicio3_tb.vcd
===== [SUCCESS] Took 0.34 seconds =====

C:\Users\franc\Documents\GitHub\DIGITAL-Mont-far-193879\LAB9\Ej3>apio sim
---> WARNING: no PCF file found (.pcf)

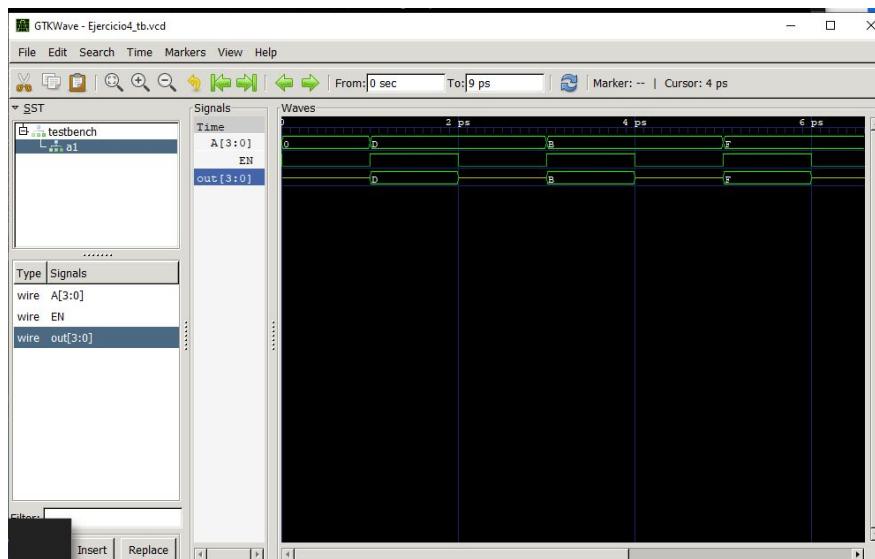
iverilog -o Ejercicio3_tb.out -D VCD_OUTPUT=Ejercicio3_tb C:/Users/franc/.apio/packages/toolchain-yosys/share/yosys/ice4
0/cells_sim.v Ejercicio3_tb.v Ejercicio3.v
vvp Ejercicio3_tb.out
ejercicio3
clk rst en j k | q
-----|-----
VCD info: dumpfile Ejercicio3_tb.vcd opened for output.
0 0 0 0 0 | x
1 1 0 0 0 | 0
0 0 0 0 0 | 0
1 0 1 1 0 | 1
0 0 1 0 1 | 0
1 0 1 1 1 | 1
0 0 1 1 1 | 1
gtkwave Ejercicio3_tb.vcd Ejercicio3_tb.gtkw
```



Ejercicio 4

Este ejercicio consistía en implementar un buffer tri-estado de 4 bits. Para implementarlo, primero se definieron las entradas y salidas del mismo. Luego se asignó a la salida con un operador ternario para que cuando el selector estuviera en 1, la salida fuera igual a la entrada y cuando el selector estuviera en cero, la salida estaría en alta impedancia.

```
C:\Windows\System32\cmd.exe - apio sim
C:\Users\franc\Documents\GitHub\DIGITAL-Mont-far-193879\LAB9\Ej4>apio clean
---> WARNING: no PCF file found (.pcf)
Removed Ejercicio4_tb.out
Removed Ejercicio4_tb.vcd
===== [SUCCESS] Took 0.34 seconds =====
C:\Users\franc\Documents\GitHub\DIGITAL-Mont-far-193879\LAB9\Ej4>apio sim
---> WARNING: no PCF file found (.pcf)
iverilog -o Ejercicio4_tb.out -D VCD_OUTPUT=Ejercicio4_tb C:/Users/franc/.apio/packages/toolchain-yosys\share\yosys\ice4
0/cells_sim.v Ejercicio4_tb.v Ejercicio_4.v
vvp Ejercicio4_tb.out
ejercicio 4
EN A | OUT
-----
VCD info: dumpfile Ejercicio4_tb.vcd opened for output.
0 0000 | zzzz
1 1101 | 1101
0 1101 | zzzz
1 1011 | 1011
0 1011 | zzzz
1 1111 | 1111
0 1111 | zzzz
1 0100 | 0100
0 0100 | zzzz
gtkwave Ejercicio4_tb.vcd Ejercicio4_tb.gtkw
```



Ejercicio 5

Este ejercicio consistía en implementar una tabla. Para implementarlo, se utilizaron cases, en este caso, la función de casex, debido a que trabajamos con don't cares. En el case se asignar un valor para un resultado pre definido.

```
C:\Windows\System32\cmd.exe - apio sim
A      |      Y
-----|-----
VCD info: dumpfile Ejercicio5_tb.vcd opened for output.
xxxxxxx| xxxxxxxx
xxxxxx0| 100000001000
00001x1| 010000001000
00000x1| 100000001000
00011x1| 100000001000
00010x1| 010000001000
0010xx1| 0001001000010
0011xx1| 1001001100000
0100xx1| 0011010000010
0101xx1| 0011010000100
0110xx1| 1011010100000
0111xx1| 1000000111000
1000x11| 010000001000
1000x01| 100000001000
1001x11| 100000001000
1001x01| 010000001000
1010xx1| 0011011000010
1011xx1| 1011011100000
1100xx1| 010000001000
1101xx1| 000000001001
1110xx1| 0011100000010
1111xx1| 1011100100000
gtkwave Ejercicio5_tb.vcd Ejercicio5_tb.gtkw
```

