

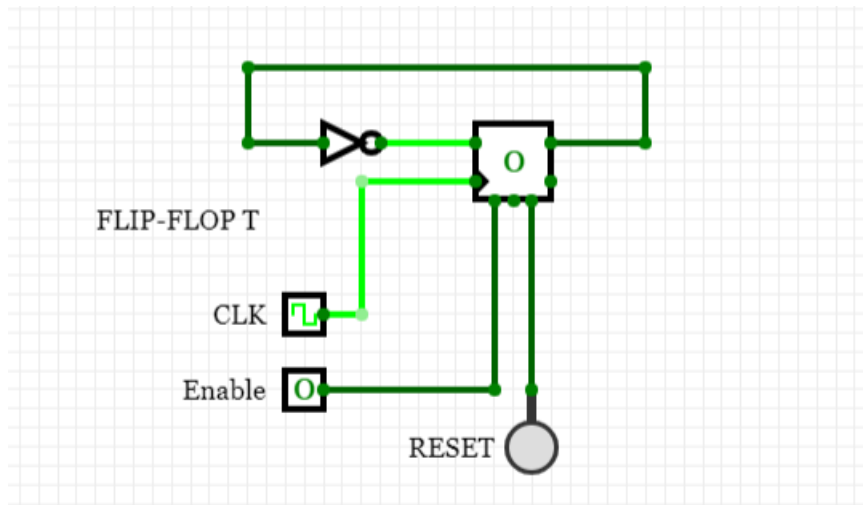
## LAB 09

### Ejercicio #1:

Para este ejercicio implemente inicialmente mi flip-flop de un bit, y luego de ello procedí con el mismo a implementar el de dos bits, calculando cada parte individual de la salida Q y de la misma manera para el de 4 bits.

### Ejercicio #2:

Para la implementación del flip-flop tipo T, utilice el módulo del tipo D inicial en donde la única variante a realiza fue que para la entrada del valor D seria el negativo de la salida Q, y con esto logre realizar la alternancia de valores entre 0 y 1.



Implementación en circuit verse.

### Ejercicio # 3:

Para la implementación de este nuevo flip-flop tipo JR, realice una nube combinacional la cual posea la siguiente lógica.

entradas del FFD		salida del FFD	
J	K	Q (actual)	D para Q futuro
0	0	Qact	Qact
0	1	x	0
1	0	x	1
1	1	Qact	$\sim Qact$

De una forma codificada se obtuvo que:

J	K	Q	D'
0	0	1	1
0	1	x	0
1	0	x	1
1	1	1	0
1	1	0	1

Con esto se logró determinar que la nube final es:

J	K	Q	=>	D'
1	X	0		1
X	0	1		1

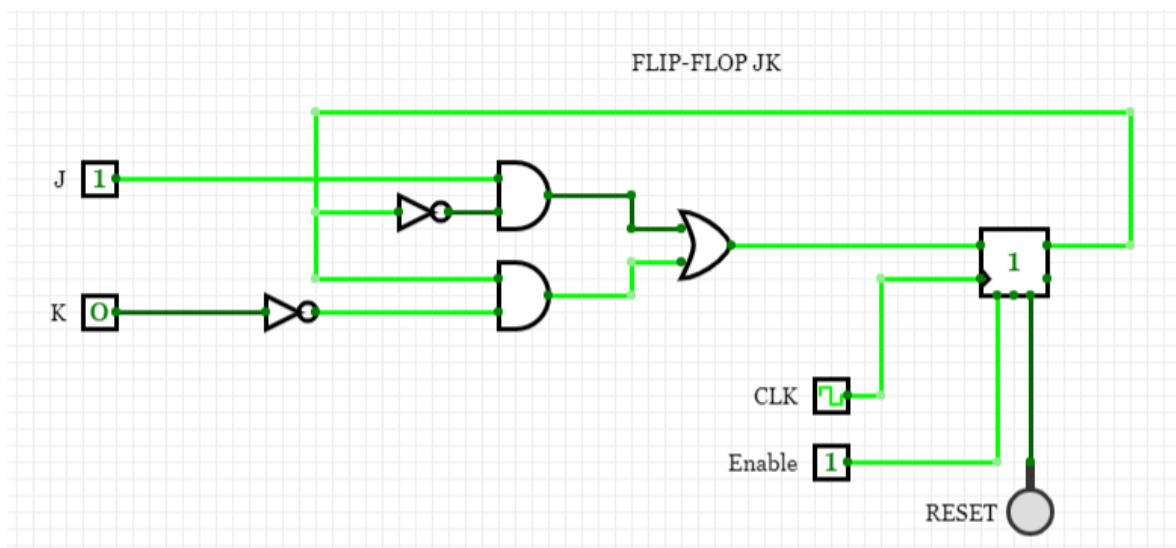
Imported from file:

$$D' = J' K' Q + J K' Q' + J K' Q + J K Q';$$

Minimized:

$$D' = J Q' + K' Q;$$

Implementación en circuit verse.



#### Ejercicio #4:

Para el buffer triestado de 4bits me base en la utilización de un always el cual se activa por cada cambio positivo de los valores de entrada en donde en las condicionales agregue que para cualquier valor de entrada cuando ENABLE es 0, da una alta impedancia de lo contrario si esta en 1, deja pasar de largo el valor de entrada.

Link repositotio

[https://github.com/Angel-Cuellar/laboratorios\\_digital01](https://github.com/Angel-Cuellar/laboratorios_digital01)