



INSTITUTO POLITÉCNICO NACIONAL



ESCUELA SUPERIOR DE CÓMPUTO

INGENIERIA EN SISTEMAS COMPUTACIONALES

MATERIA: DISEÑO DE SISTEMAS DIGITALES

PROFESOR: TESTA NAVA ALEXIS

PRESENTA:

RAMIREZ BENITEZ BRAYAN

GRUPO: 2CV18

PRACTICA 4:

CONTADOR CON FSM MOORE

CIUDAD DE MEXICO MAYO DE 2021

Para esta práctica utilizaremos la GAL C22V10 en el Simulador Galaxy, esta práctica consistió en programar un contador con FSM Moore con 3 bits donde tendrá un selector para sus estados incluye, clear asíncrono y una señal de reloj, a continuación, las salidas y entradas de ocupamos de la GAL y el código correspondiente a la práctica.

Contador FSM Moore

Los contadores pueden tener varias señales de control dependiendo de las operaciones que realiza. Las más comunes son:

Señal de dirección de conteo (UD – Up/Down). Cuando esta activa, esta señal permite el conteo ascendente del contador, en caso contrario el conteo es descendente.

Autómata del Contador

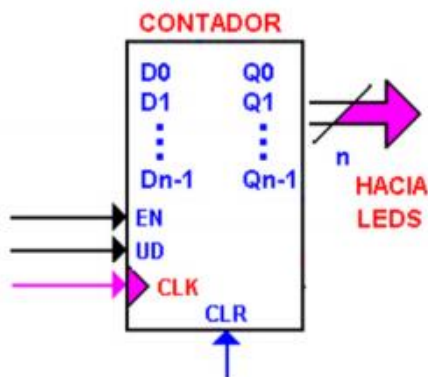
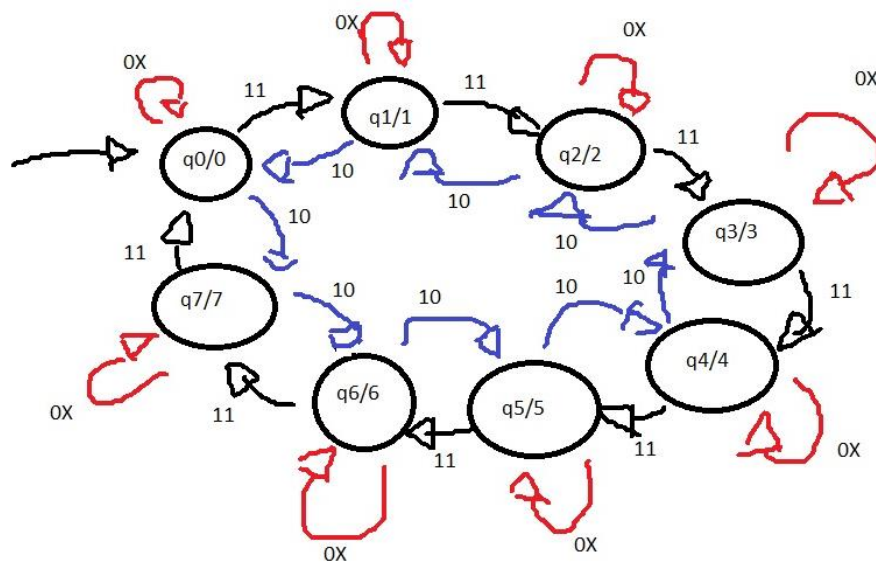


Tabla de operación del contador

EN	UD	OPERACIÓN
0	X	RETENCIÓN
1	1	CONTEO ASCENDENTE
1	0	CONTEO DESCENDENTE

Entrada Ascendente		Estado Actual			Estado siguiente						
1	1	Q2	Q1	Q0	Q2	Q1	Q0	D	J	K	T
1	1	0	0	0	0	0	1	0	0	X	1
1	1	0	0	1	0	1	0	0	1	X	1
1	1	0	1	0	0	1	1	0	X	0	1
1	1	0	1	1	1	0	0	1	X	1	1
1	1	1	0	0	1	0	1	1	0	X	1
1	1	1	0	1	1	1	0	1	1	X	1
1	1	1	1	0	1	1	1	1	X	0	1
1	1	1	1	1	0	0	0	0	X	1	1

Entrada Descendente		Estado Actual			Estado siguiente						
1	0	Q2	Q1	Q0	Q2	Q1	Q0	D	J	K	T
1	0	0	0	0	1	1	1	1	1	X	1
1	0	0	0	1	0	0	0	0	0	X	1
1	0	0	1	0	0	0	1	0	X	1	1
1	0	0	1	1	0	1	0	0	X	0	1
1	0	1	0	0	0	1	1	0	1	X	1
1	0	1	0	1	1	0	0	1	0	X	1
1	0	1	1	0	1	0	1	1	X	1	1
1	0	1	1	1	1	1	0	1	X	0	1

Para la parte Ascendente:

Q1 \ Q0	00	01	11	10
Q2 0			1	
Q2 1	1	1		1

$$D = Q2'Q1Q0 + Q2Q1' + Q2Q0'$$

Q1 \ Q0	00	01	11	10
Q2 0		1	X	X
Q2 1	1	1	X	X

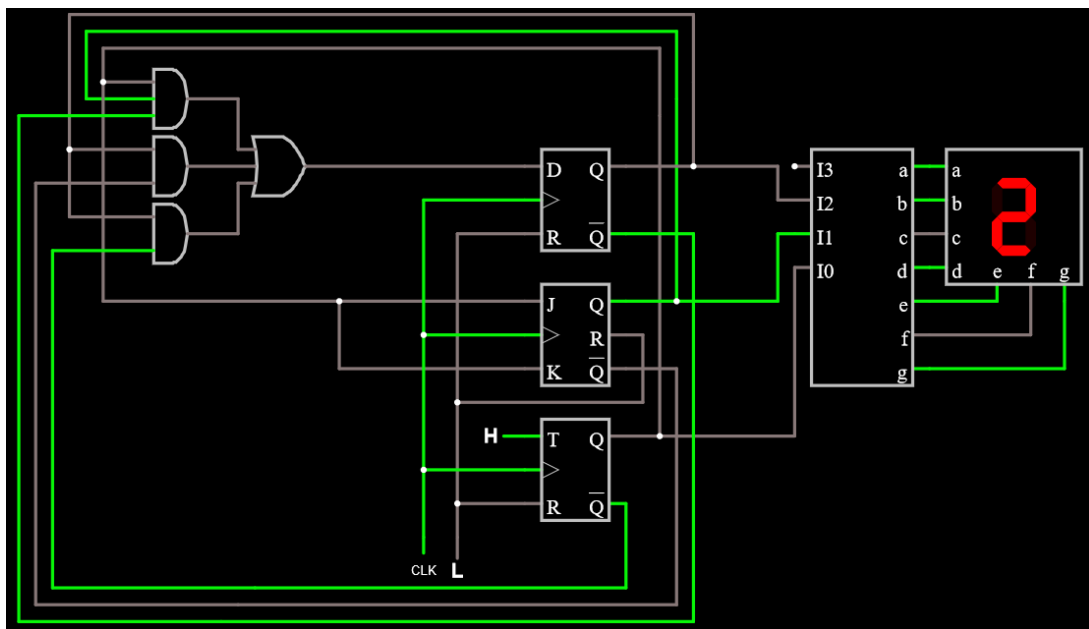
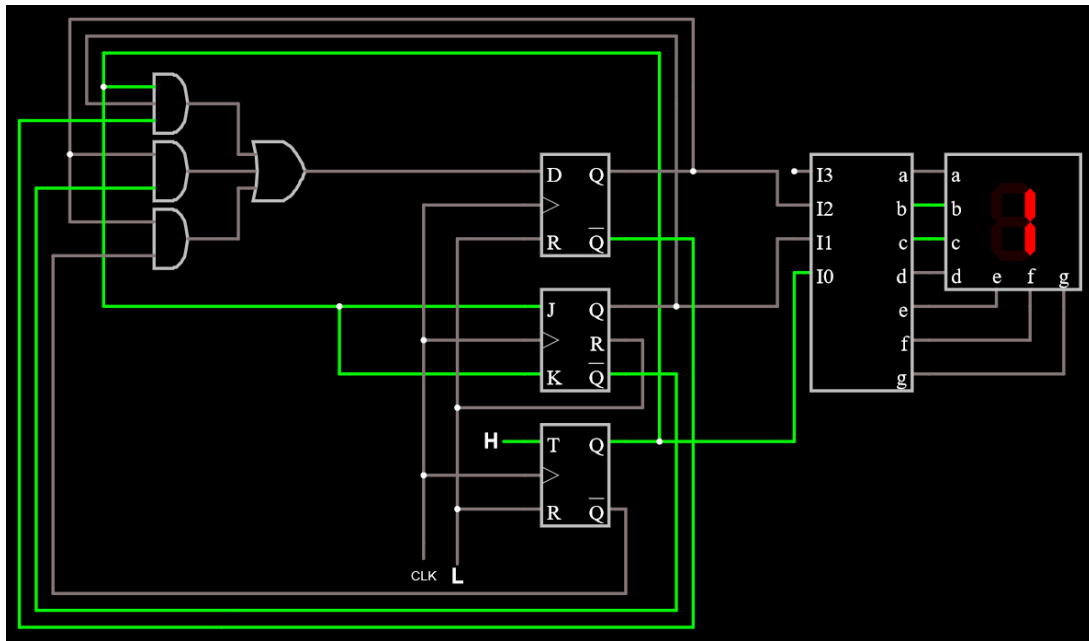
$$J = Q0$$

Q1 \ Q0	00	01	11	10
Q2 0	X	X	1	
Q2 1	X	X	1	

$$K = Q0$$

Q1 \ Q0	00	01	11	10
Q2 0		1	1	1
Q2 1	1	1	1	1

$$T = 1$$



Link: <https://tinyurl.com/yg7tw7ui>

Para la parte Descendente:

Q1 \ Q0	00	01	11	10
0	1			
1		1	1	1

$$D = Q2'Q1'Q0' + Q2Q0 + Q2Q1$$

Q1 \ Q0	00	01	11	10
0	1		X	X
1	1		X	X

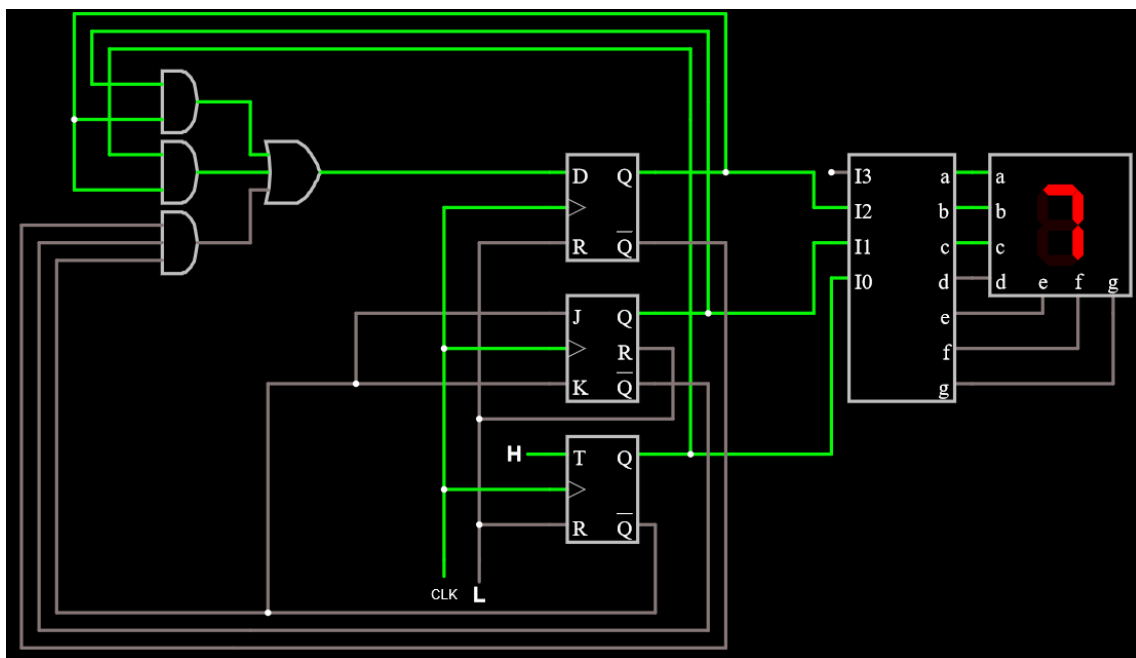
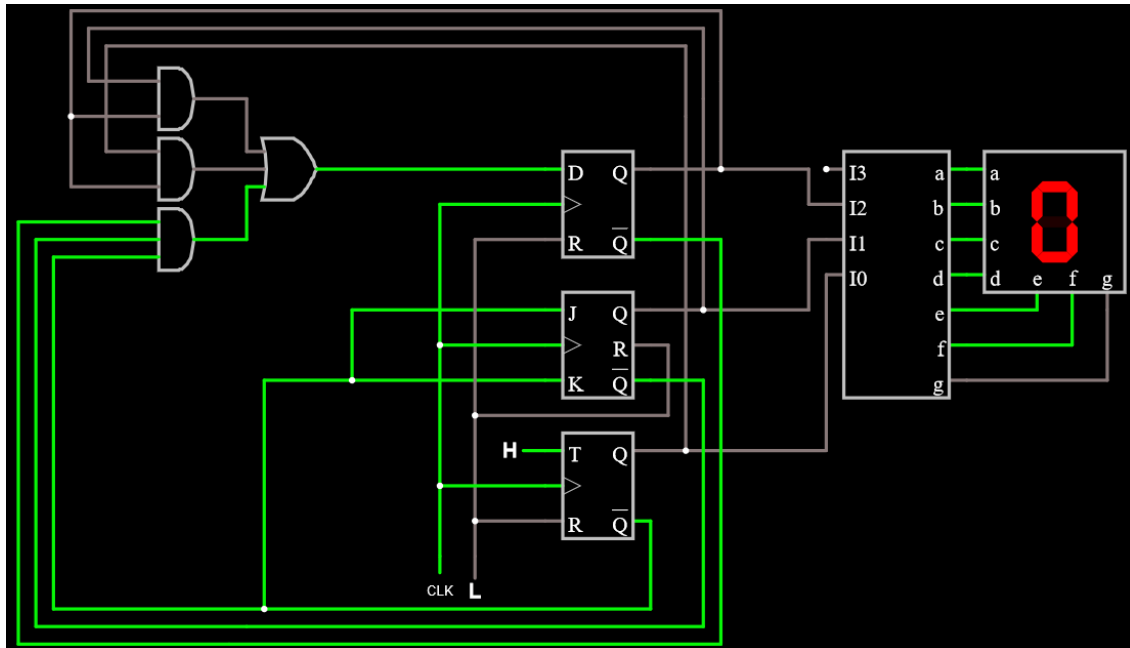
$$J = Q0'$$

Q1 \ Q0	00	01	11	10
0	X	X		1
1	X	X		1

$$K = Q0'$$

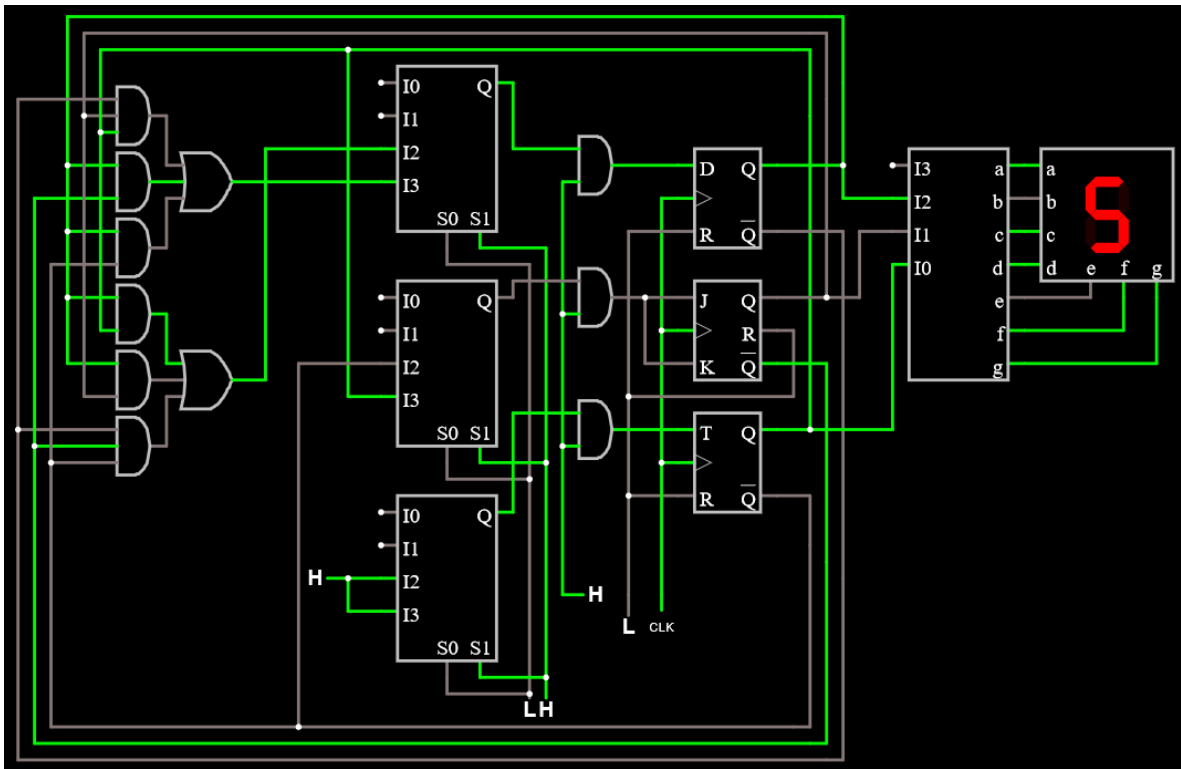
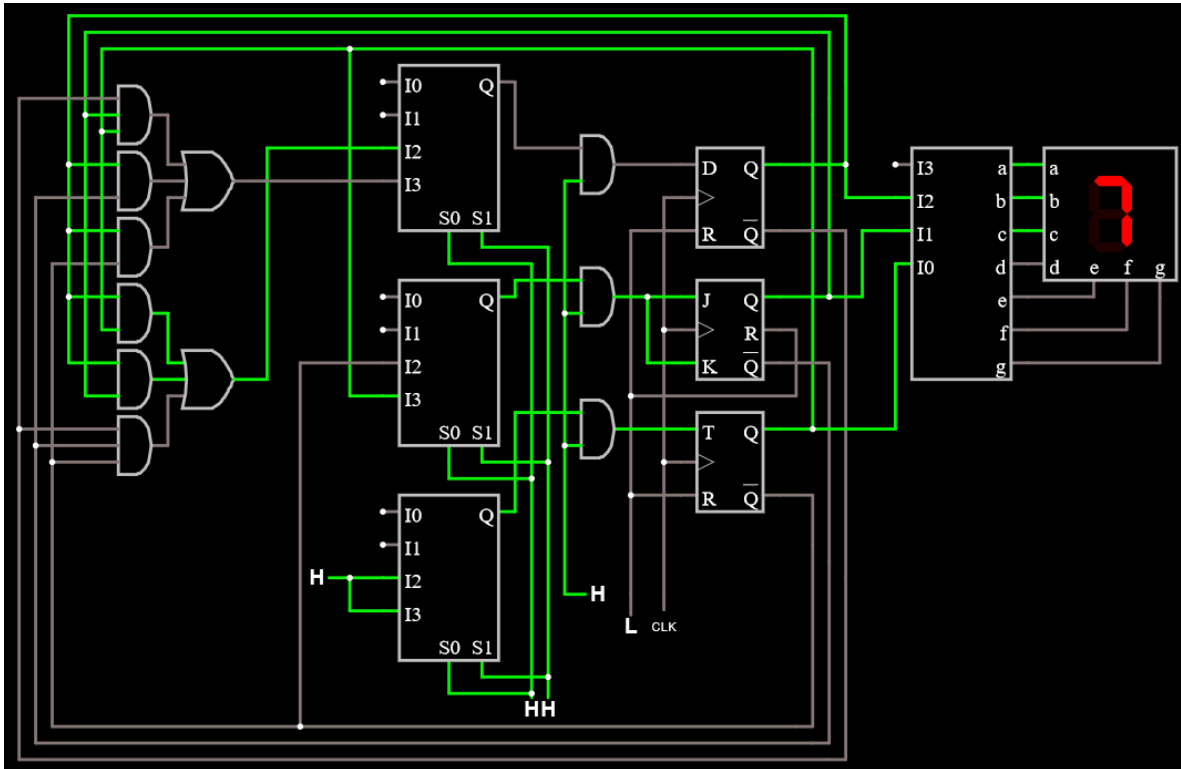
Q1 \ Q0	00	01	11	10
0	1	1	1	1
1	1	1	1	1

$$T = 1$$



Link: <https://tinyurl.com/yhiqqluy>

Para la parte Ascendente/Descendente con retención y enable.

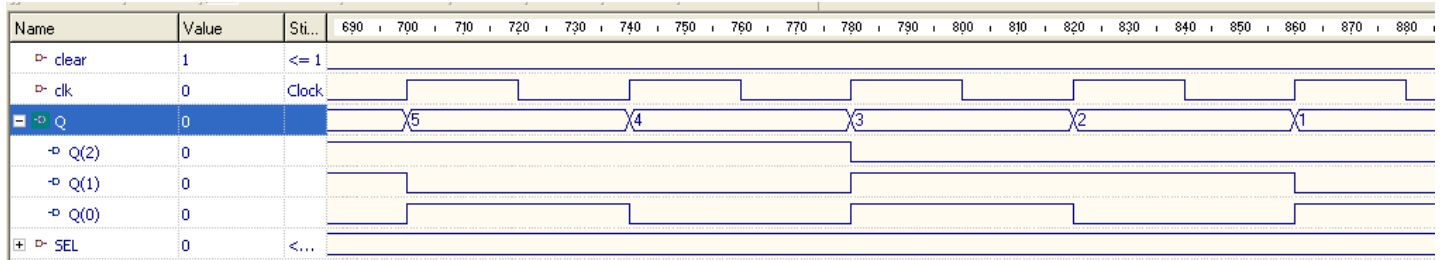
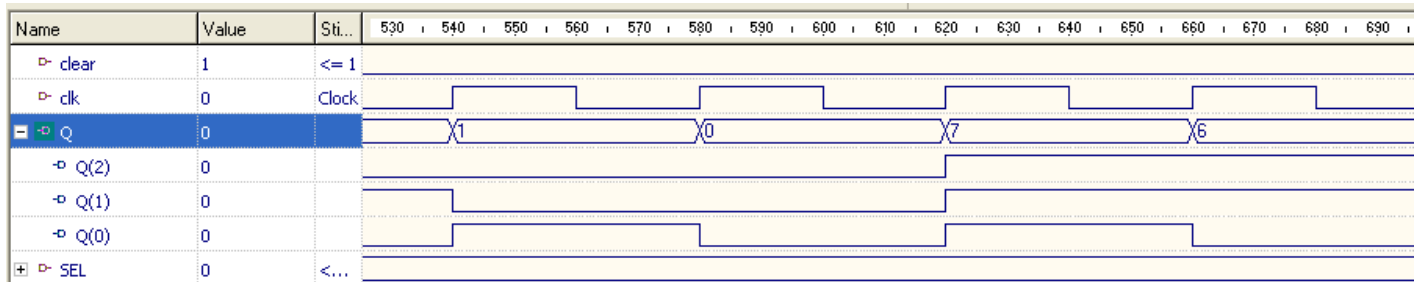
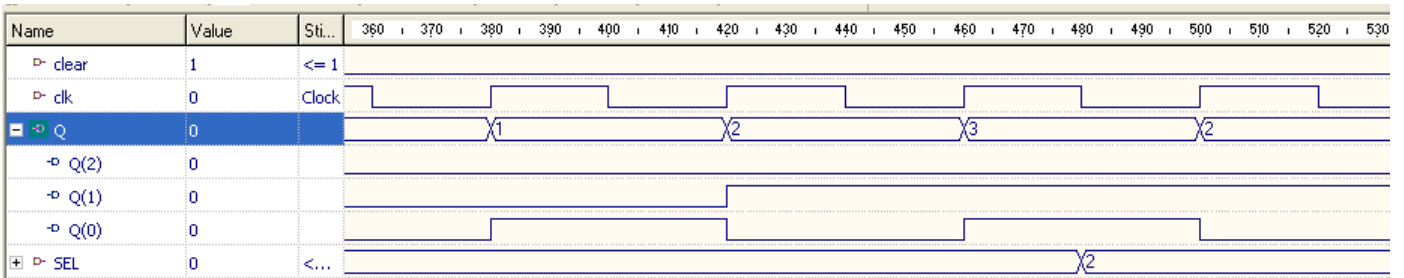
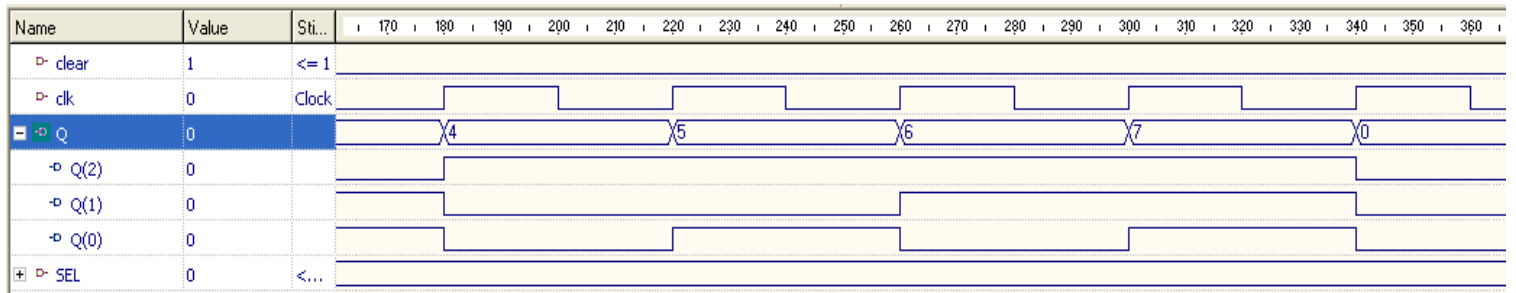
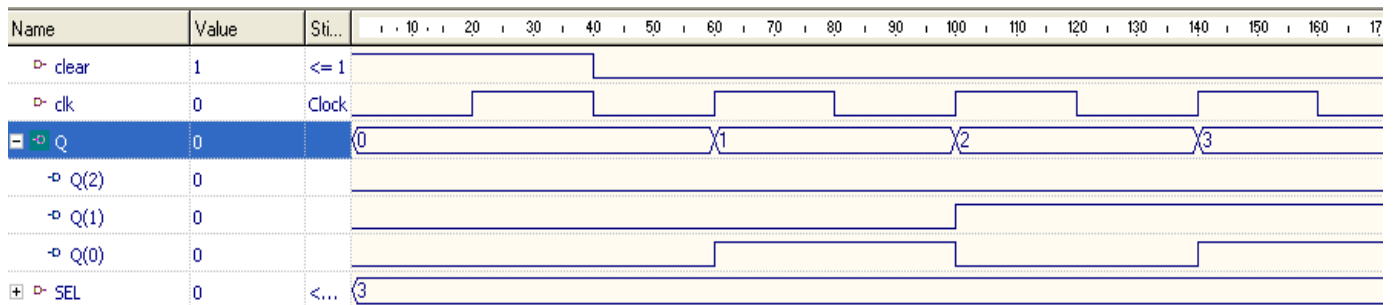


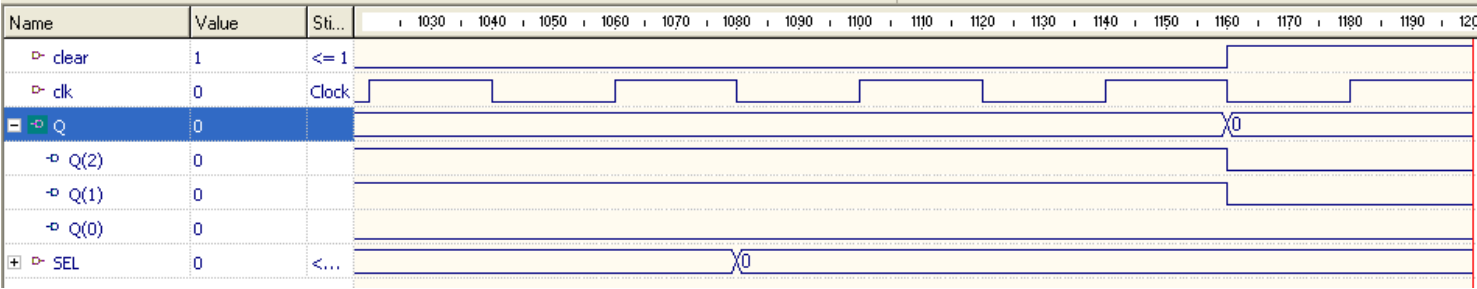
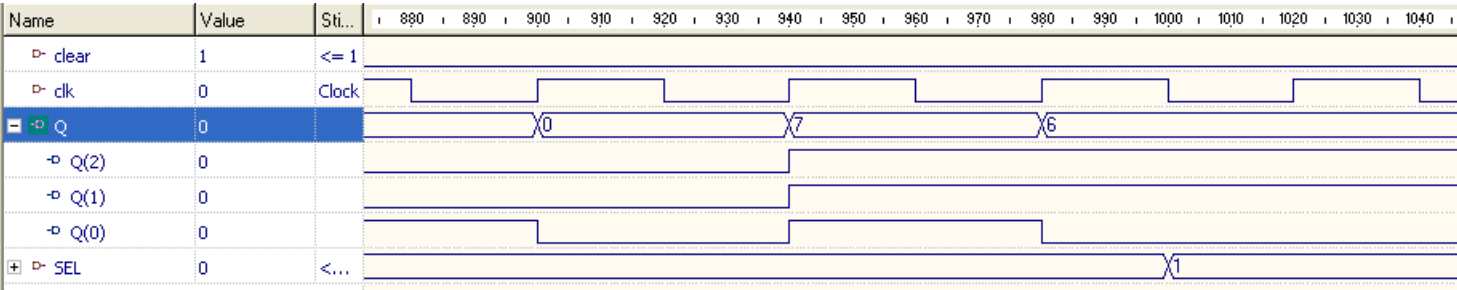
Link: <https://tinyurl.com/yfzr6njw>

Código y simulación del contador FSM Moore

```
1 LIBRARY IEEE;
2
3 USE IEEE.STD_LOGIC_1164.ALL;
4
5 ENTITY pract4 IS
6
7     PORT( clk, clear : IN STD_LOGIC;
8           SEL : IN STD_LOGIC_VECTOR(1 downto 0);
9           Q : OUT STD_LOGIC_VECTOR(2 downto 0));
10
11 END pract4;
12
13 ARCHITECTURE behavior OF pract4 IS
14
15     SIGNAL Q_aux : STD_LOGIC_VECTOR(2 downto 0) := "000";
16     SIGNAL T : STD_LOGIC := '1';
17
18 BEGIN
19     PROCESS (clk, clear)
20     BEGIN
21         IF (clear = '1') THEN
22             Q_aux <= "000";
23         ELSIF ((clk'EVENT) AND (clk = '1')) THEN
24             CASE SEL IS
25                 WHEN "00" => Q_aux <= Q_aux;
26                 WHEN "01" => Q_aux <= Q_aux;
27                 WHEN "10" =>
28                     Q_aux(2) <= ((NOT Q_aux(2)) AND (NOT Q_aux(1)) AND (NOT Q_aux(0))) OR (Q_aux(2) AND Q_aux(0)) OR (Q_aux(2) AND Q_aux(1));--d
29                     Q_aux(1) <= ((NOT Q_aux(0)) AND (NOT Q_aux(1))) OR (Q_aux(0) AND Q_aux(1));--jk
30                     Q_aux(0) <= (T XOR Q_aux(0));--t
31                 WHEN "11" =>
32                     Q_aux(2) <= ((NOT Q_aux(2)) AND (Q_aux(1)) AND (Q_aux(0))) OR (Q_aux(2) AND (NOT Q_aux(1))) OR (Q_aux(2) AND (NOT Q_aux(0)));--d
33                     Q_aux(1) <= ((Q_aux(0) AND (NOT Q_aux(1))) OR ((NOT Q_aux(0)) AND Q_aux(1)));--jk
34                     Q_aux(0) <= (T XOR Q_aux(0));--t
35                 WHEN OTHERS => Q_aux <= "000";
36             END CASE;
37         END IF;
38     END PROCESS;
39     Q <= Q_aux;
40 END behavior;
```

WARP done.
Compilation successful.
genvhdl -s 1164_VHDL -i "practica4.vhd"
Running: batnova -v -f -lstd_logic practica4
genvhdl completed
Done.





Conclusiones y observaciones

Esta práctica consistió en elaborar un contador FSM Moore Ascendente/Descendente, para elaborarlo fue necesario modificar el autómata propuesto para la practica el cual contaba con dos características importantes ya que era ascendente y podía estar en retención, su elaboración consistió en encontrar su tabla de estados y las ecuaciones para cada entrada, dado que previamente ya tenía conocimiento acerca de los contadores no fue difícil elaborar esta práctica, utilizamos 3 bits para conseguir la cantidad de estados necesarios y usamos 3 Flip – Flops, los cuales son el tipo D, JK y T, además añadí una señal de reloj, así como por el clear y una entrada SEL que establece de qué manera se ejecutara.

Por último, durante el desarrollo de la práctica no ocurrió ningún problema, aunque fue un poco confuso elaborar los contadores sin el uso de los operadores + y – para el funcionamiento de la práctica.

Anexos y bibliografía

Referencias bibliográficas:

- Floyd, T. L. (2021). *Fundamentos De Sistemas Digitales* (9.^a ed.) [Libro electrónico]. PRENTICE HALL/PEARSON.
https://www.academia.edu/34699883/Libro_fundamentos_de_sistemas_digitales_floyd_9ed_PDF
- Maxinez, D. (2013). *Programación de sistemas digitales con VHDL* (1.^a ed.). Grupo Editorial Patria. <https://editorialpatria.com.mx/pdf/files/9786074386219.pdf>