



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 3:

CONTADOR JOHNSON Y CONTADOR ANILLO

CIUDAD DE MEXICO ABRIL DE 2021

Para esta práctica utilizaremos la GAL C22V10 en el Simulador Galaxy, esta práctica consistió en programar un contador Johnson y un contador anillo de 8 bits que serán seleccionados mediante un selector de 1 bit además 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.

### El contador Johnson

Es un registro de desplazamiento básico, pero con la característica en la cual los datos no se pierden al desplazarse, sino que la información rota debido a la realimentación.

Un contador Johnson generara un módulo de  $2n$  estados siendo  $n$  el número de flipflop(etapas) del contado.

Impulso de reloj	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7
0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0
2	1	1	0	0	0	0	0	0
3	1	1	1	0	0	0	0	0
4	1	1	1	1	0	0	0	0
5	1	1	1	1	1	0	0	0
6	1	1	1	1	1	1	0	0
7	1	1	1	1	1	1	1	0
8	1	1	1	1	1	1	1	1
9	0	1	1	1	1	1	1	1
10	0	0	1	1	1	1	1	1
11	0	0	0	1	1	1	1	1
12	0	0	0	0	1	1	1	1
13	0	0	0	0	0	1	1	1
14	0	0	0	0	0	0	1	1
15	0	0	0	0	0	0	0	1

Figura 1. La secuencia de 8 bits tiene un total de 16 estados

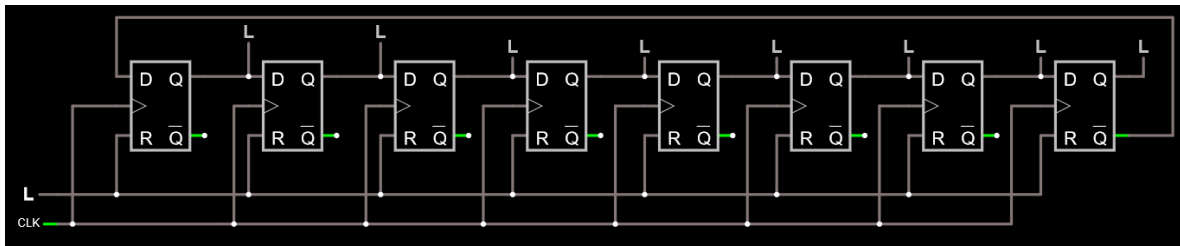


Figura 2. Arreglo de 8 biestables para el contador

Link en falstad: <https://tinyurl.com/yj5bp8go>

### El contador anillo

El funcionamiento del contador de anillo es similar al del contador Johnson, pero en este caso no se necesita lógica combinacional a la salida del ultimo flip-flop, se realimenta directamente como se muestra en la figura de tal forma que los datos se desplazan en forma de anillo

Impulso de Reloj	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7
0	1	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0
3	0	0	0	1	0	0	0	0
4	0	0	0	0	1	0	0	0
5	0	0	0	0	0	1	0	0
6	0	0	0	0	0	0	1	0
7	0	0	0	0	0	0	0	1

Figura 4. Secuencia del contador anillo

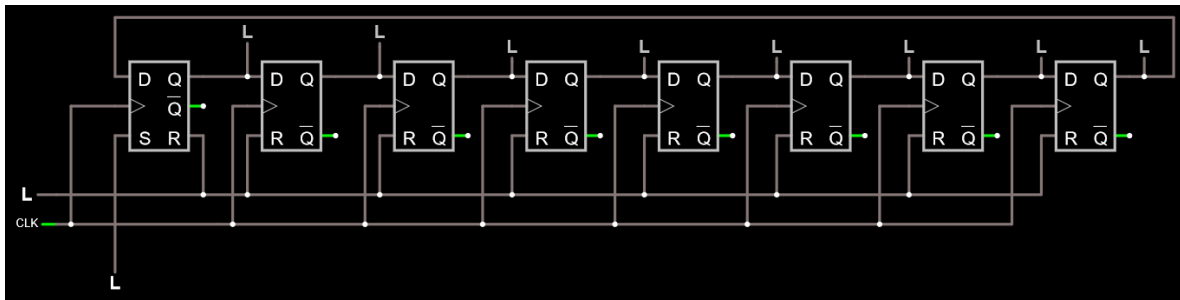


Figura 5. Arreglo de 8 biestables para el contador

Link en falstad: <https://tinyurl.com/yk6oufjq>

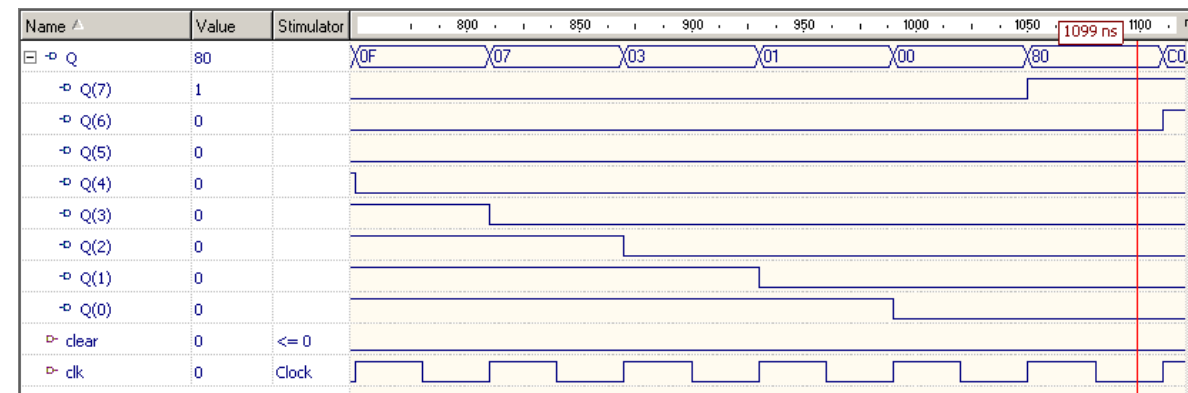
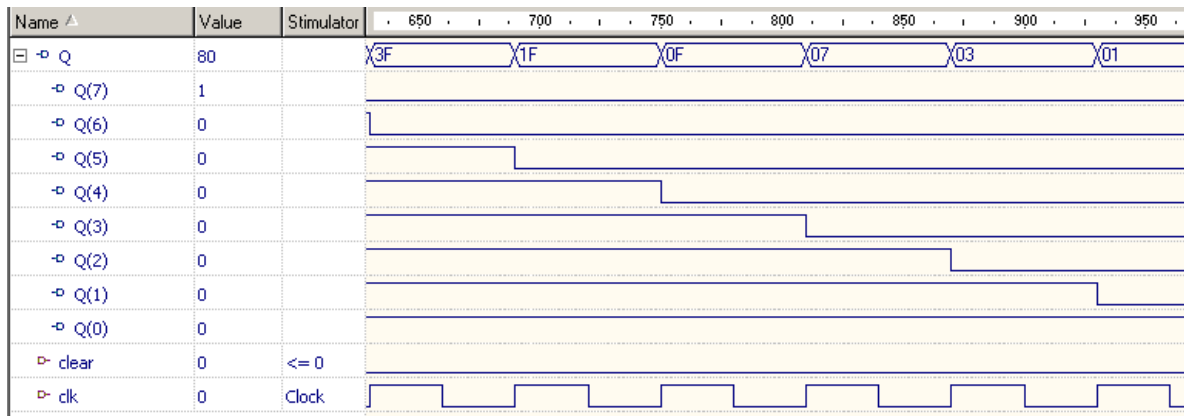
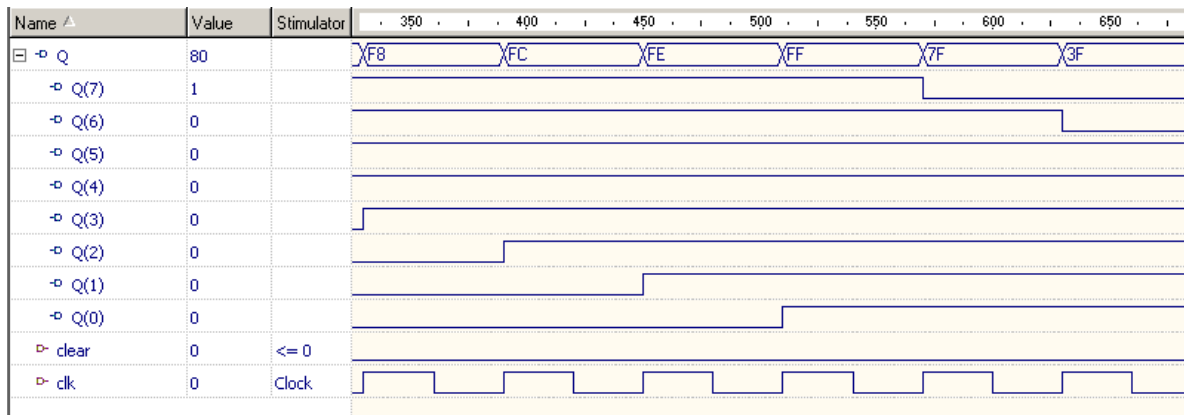
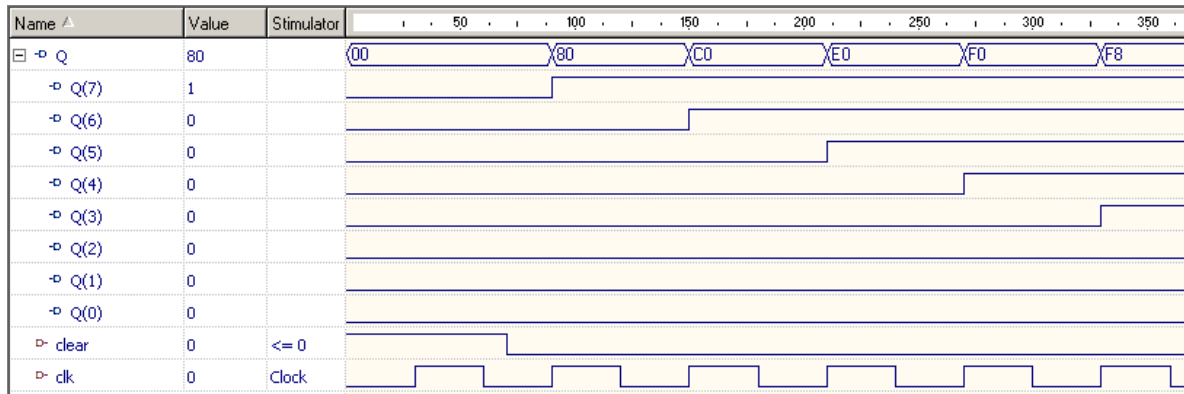
### Código y simulación para el Contador Johnson

```

1 LIBRARY IEEE;
2
3 USE IEEE.STD_LOGIC_1164.ALL;
4
5 ENTITY CJOHNSON IS
6
7     PORT( clk, clear, SEL : IN STD_LOGIC;
8           Q : OUT STD_LOGIC_VECTOR(7 downto 0));
9
10 END CJOHNSON;
11
12 ARCHITECTURE BEHAVIOR OF CJOHNSON IS
13     SIGNAL Q_aux : STD_LOGIC_VECTOR(7 downto 0);
14
15 BEGIN
16     PROCESS (clk, clear, SEL)
17     BEGIN
18         IF (clear = '1') THEN
19             Q_aux <= "00000000";
20         ELSIF ((clk'EVENT) AND (clk = '1')) THEN
21             Q_aux <= (NOT Q_aux(0)) & Q_aux(7 downto 1);
22         END IF;
23     END PROCESS;
24     Q <= Q_aux;
25 END BEHAVIOR;
26

```

WARP done.  
Compilation successful.  
genvhdl -s 1164\_VHDL -i "CJOHNSON.vhd"  
Running: batnova -v -f -lstd\_logic CJOHNSON  
genvhdl completed  
Done.



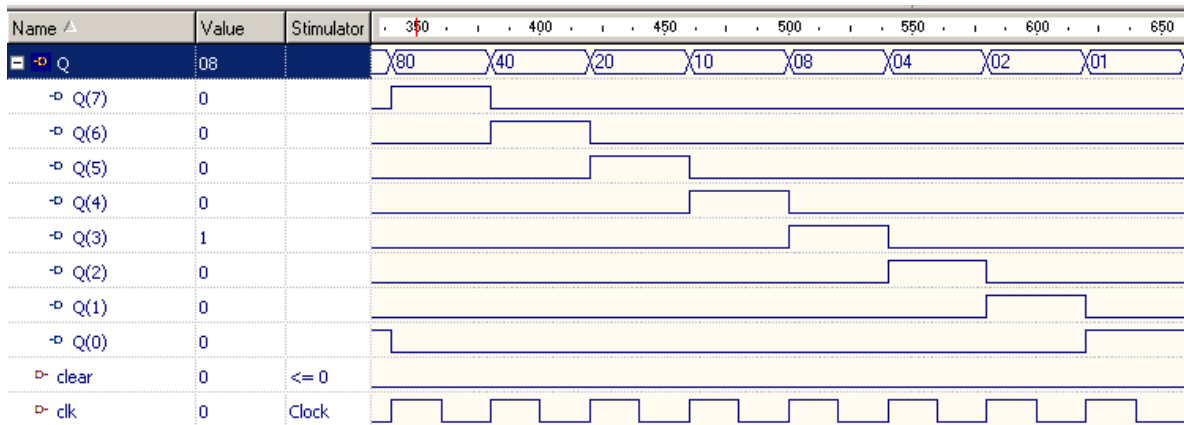
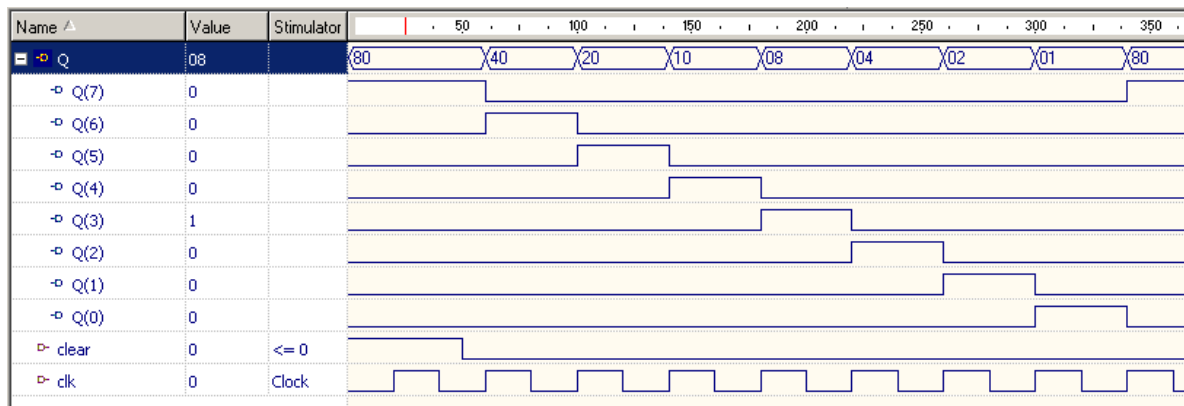
## Código y simulación para el Contador Anillo

The image shows a screenshot of a VHDL editor window titled "Contador\_Anillo [Device - PALCE22V10-25PC/PI] - Galaxy - [CANILLO.vhd]". The editor displays the VHDL code for a Ring Counter. The code is as follows:

```
1 LIBRARY IEEE;
2
3 USE IEEE.STD_LOGIC_1164.ALL;
4
5 ENTITY CANILLO IS
6
7     PORT( clk, clear : IN STD_LOGIC;
8           Q : OUT STD_LOGIC_VECTOR(7 downto 0));
9
10 END CANILLO;
11
12 ARCHITECTURE BEHAVIOR OF CANILLO IS
13
14     SIGNAL Q_aux : STD_LOGIC_VECTOR(7 downto 0);
15
16 BEGIN
17     PROCESS (clk, clear)
18     BEGIN
19         IF (clear = '1') THEN
20             Q_aux <= "10000000";
21         ELSIF ((clk'EVENT) AND (clk = '1')) THEN
22             Q_aux(6) <= Q_aux(7);
23             Q_aux(5) <= Q_aux(6);
24             Q_aux(4) <= Q_aux(5);
25             Q_aux(3) <= Q_aux(4);
26             Q_aux(2) <= Q_aux(3);
27             Q_aux(1) <= Q_aux(2);
28             Q_aux(0) <= Q_aux(1);
29             Q_aux(7) <= Q_aux(0);
30         END IF;
31     END PROCESS;
32     Q <= Q_aux;
33 END BEHAVIOR;
```

The simulation results at the bottom of the window show the following output:

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

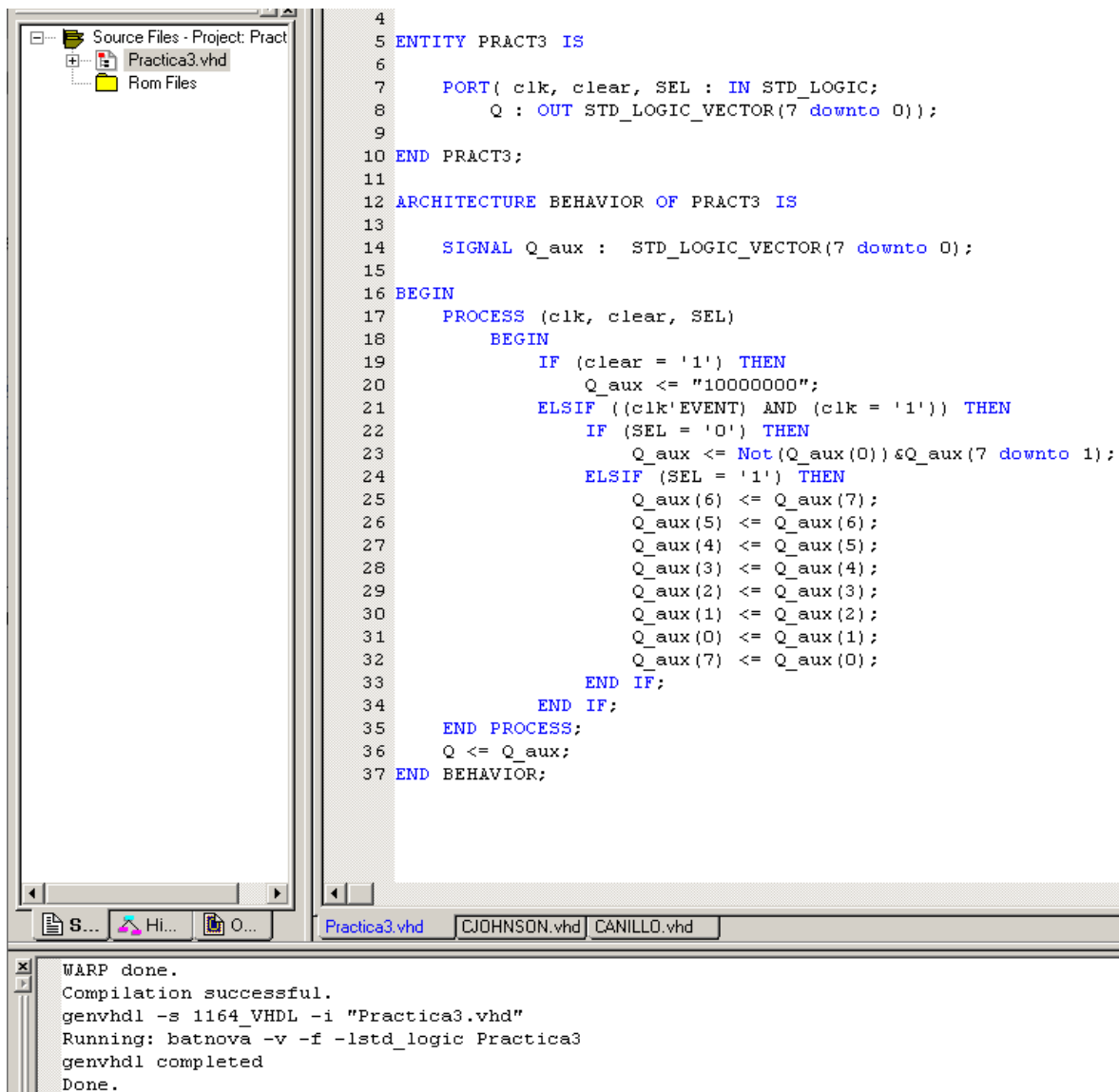


## Código y simulación para el Contador Johnson y Contador Anillo

Para la selección de contadores:

Contador	SEL
JOHNSON	0
ANILLO	1

Nota en el selector Johnson omito el estado 0 puesto que el valor del clear asigna un vector "10000000" el cual corresponde al estado 1, durante la conclusión detallare esto.



```
4
5 ENTITY PRACT3 IS
6
7     PORT( clk, clear, SEL : IN STD_LOGIC;
8           Q : OUT STD_LOGIC_VECTOR(7 downto 0));
9
10 END PRACT3;
11
12 ARCHITECTURE BEHAVIOR OF PRACT3 IS
13
14     SIGNAL Q_aux : STD_LOGIC_VECTOR(7 downto 0);
15
16 BEGIN
17     PROCESS (clk, clear, SEL)
18     BEGIN
19         IF (clear = '1') THEN
20             Q_aux <= "10000000";
21         ELSIF ((clk'EVENT) AND (clk = '1')) THEN
22             IF (SEL = '0') THEN
23                 Q_aux <= Not(Q_aux(0)) & Q_aux(7 downto 1);
24             ELSIF (SEL = '1') THEN
25                 Q_aux(6) <= Q_aux(7);
26                 Q_aux(5) <= Q_aux(6);
27                 Q_aux(4) <= Q_aux(5);
28                 Q_aux(3) <= Q_aux(4);
29                 Q_aux(2) <= Q_aux(3);
30                 Q_aux(1) <= Q_aux(2);
31                 Q_aux(0) <= Q_aux(1);
32                 Q_aux(7) <= Q_aux(0);
33             END IF;
34         END IF;
35     END PROCESS;
36     Q <= Q_aux;
37 END BEHAVIOR;
```

WARP done.  
Compilation successful.  
genvhdl -s 1164 VHDL -i "Practica3.vhd"  
Running: batnova -v -f -lstd\_logic Practica3  
genvhdl completed  
Done.



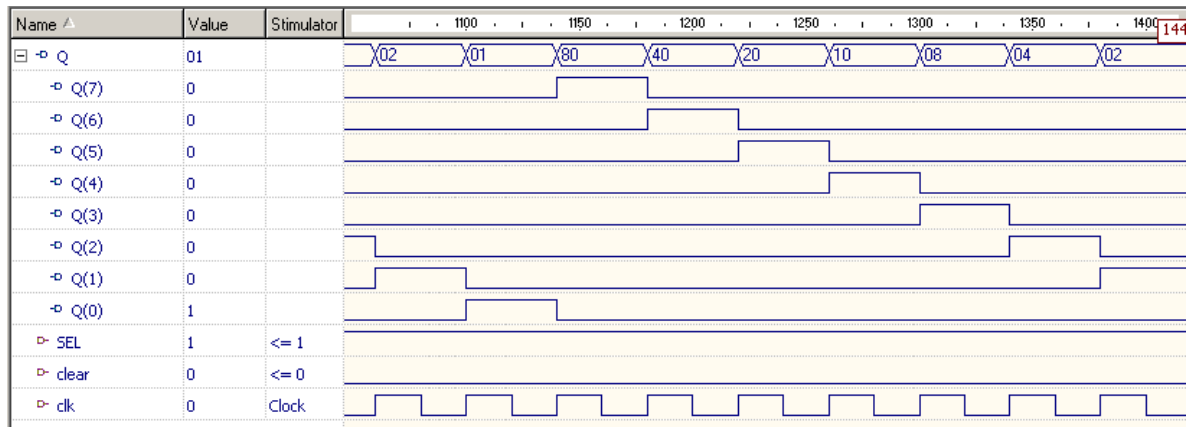
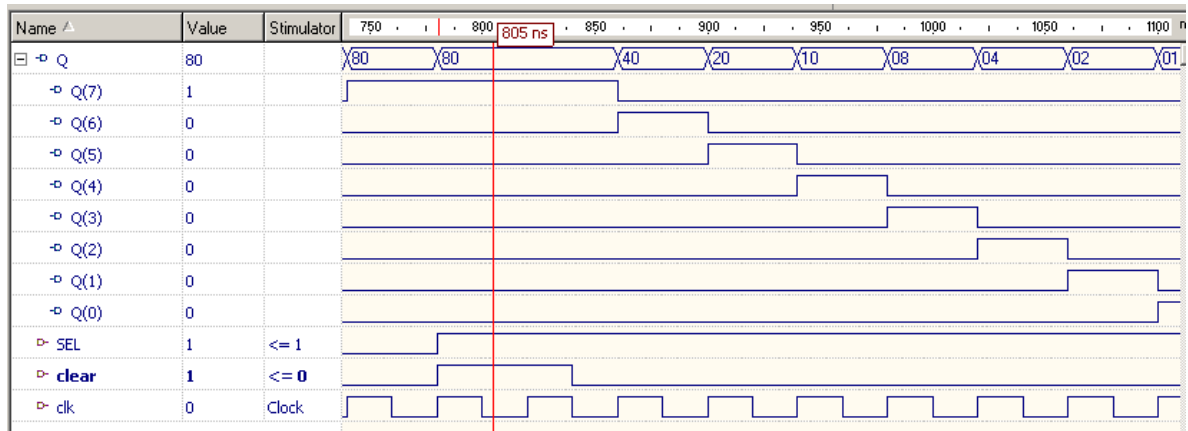
## Contador Johnson

Name ▲	Value	Stimulator										
Q	C0		UU	80	C0	E0	F0	F8	FC	FE		
Q(7)	1											
Q(6)	1											
Q(5)	0											
Q(4)	0											
Q(3)	0											
Q(2)	0											
Q(1)	0											
Q(0)	0											
SEL	0	<= 0										
clear	0	<= 0										
clk	1	Clock										

Name ▲	Value	Stimulator	350	400	450	500	550	600	650		
Q	C0		FE	FF	7F	3F	1F	0F	07	03	01
Q(7)	1										
Q(6)	1										
Q(5)	0										
Q(4)	0										
Q(3)	0										
Q(2)	0										
Q(1)	0										
Q(0)	0										
SEL	0	<= 0									
clear	0	<= 0									
clk	1	Clock									

Name ▲	Value	Stimulator	650	700	750	
Q	C0		3	01	00	80
Q(7)	1					
Q(6)	1					
Q(5)	0					
Q(4)	0					
Q(3)	0					
Q(2)	0					
Q(1)	0					
Q(0)	0					
SEL	0	<= 0				
clear	0	<= 0				
clk	1	Clock				

## Contador Anillo



## **Conclusiones y observaciones**

Esta práctica fue posible elaborarla puesto que ambos contadores son muy similares a un registro de salida paralelo con una retroalimentación, dado que previamente ya tenía conocimiento acerca de los registros no fue difícil elaborar estos contadores, los cuales están conformados por una entrada retroalimentada y salidas, para esta práctica simulamos dos contadores en un mismo proceso los cuales son el contador Johnson y anillo de 8 bits, los cuales como ya había mencionado tienen una entrada retroalimentada y para este caso 7 salidas, es decir 8 Flip – Flops tipo D conectados, además están integrados por una señal de reloj, así como por el clear y una entrada SEL que establece que contador ejecutar.

Durante el desarrollo de la práctica únicamente ocurrió un problema puesto que el simulador Galaxy marcaba un error cuando asignas más de una condición en la arquitectura, para cada contador es necesario iniciarlo de una manera para el contador Johnson el clear debe ser de esta forma “000....N” puesto que este representaría el primer estado (Estado 0), por otro lado el contador de anillo debe tener un clear de la forma “10000...N” para que funcione correctamente o establecer un preset para el primer Flip - Flop, sin embargo en Galaxy esto no es posible, así que para solucionar esto decidí omitir un estado para el contador Johnson, es decir, establecer el clear de esta forma “10000000” lo que implica que iniciaría en el segundo estado (Estado 1) para el contador Johnson esto puede apreciarse mejor en la Figura 1, entonces de esta manera ya es posible tener ambos contadores en un mismo proceso.

## **Anexos y bibliografía**

### **Referencias bibliográficas:**

Floyd, T. L. (2021). *Fundamentos De Sistemas Digitales* (9.<sup>a</sup> ed.) [Libro electrónico]. PRENTICE HALL/PEARSON.  
[https://www.academia.edu/34699883/Libro\\_fundamentos\\_de\\_sistemas\\_digitales\\_floyd\\_9ed\\_PDF](https://www.academia.edu/34699883/Libro_fundamentos_de_sistemas_digitales_floyd_9ed_PDF)

Maxinez, D. (2013). *Programación de sistemas digitales con VHDL* (1.<sup>a</sup> ed.). Grupo Editorial Patria. <https://editorialpatria.com.mx/pdf/files/9786074386219.pdf>

### **Referencias electrónicas.**

[https://cs.uns.edu.ar/~pmd/ac\\_ing/downloads/Slides/ACI-Clase-3.pdf](https://cs.uns.edu.ar/~pmd/ac_ing/downloads/Slides/ACI-Clase-3.pdf)

<https://riverglennapts.com/es/digital-counters/283-johnson-counter.html>

[https://ikastaroak.birt.eus/edu/argitalpen/backupa/20200331/1920k/es/IEA/ELEC/ELEC03/es\\_IEA\\_ELEC03\\_Contenidos/website\\_47\\_contador\\_johnson.html](https://ikastaroak.birt.eus/edu/argitalpen/backupa/20200331/1920k/es/IEA/ELEC/ELEC03/es_IEA_ELEC03_Contenidos/website_47_contador_johnson.html)

[https://www.ecured.cu/Contador\\_en\\_anillo](https://www.ecured.cu/Contador_en_anillo)