

Tarea 2

Implementación de un contador aleatorio

***Acoidan Martín Conrado
Sistemas Electrónicos Digitales
1º Grado Ingeniería Informática
15/04/2024***

Índice

1. Secuencia asignada.....	2
2. Tabla de transiciones y excitación.....	2
3. Mapas de Karnaugh.....	3
4. Circuito del contador implementado.....	3
5. Códigos VHDL.....	4
6. Cronograma de simulación.....	7

Nota: La plantilla utilizada es una guía para la realización del informe, el estudiante la debe usar un punto de partida. Se valorará todas las explicaciones para el entendimiento del proceso seguido.

Tarea 2

Implementación de un contador aleatorio

1.- Secuencia asignada

El contador aleatorio asignado se muestra en la Tabla 1.

Alumno	Secuencia	Flip-flops				Cíclico
119 Acoidan Martín Conrado	3,5,8,1,7,4,6,9,0,2	T	D	D	T	No

Tabla 1. Contador aleatorio asignado

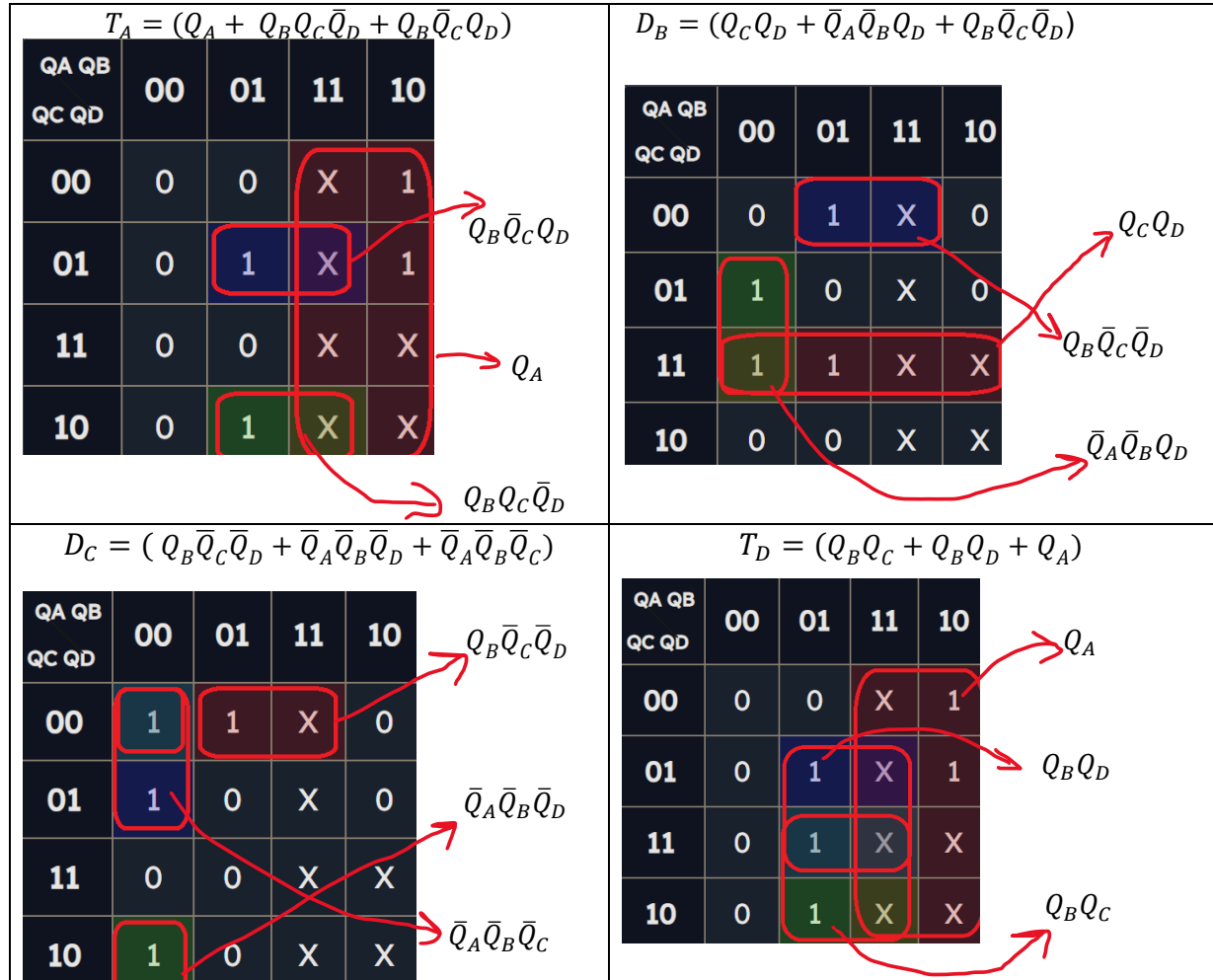
2.- Tabla de transiciones y excitación

La tabla de transiciones del contador aleatorio asignado y su tabla de excitación es:

	Estado actual				Estado siguiente							
	Q _A	Q _B	Q _C	Q _D	Q _A ⁺	Q _B ⁺	Q _C ⁺	Q _D ⁺	T _A	D _B	D _C	T _D
0	0	0	0	0	0	0	1	0	0	0	1	0
1	0	0	0	1	0	1	1	1	0	1	1	0
2	0	0	1	0	0	0	1	0	0	0	1	0
3	0	0	1	1	0	1	0	1	0	1	0	0
4	0	1	0	0	0	1	1	0	0	1	1	0
5	0	1	0	1	1	0	0	0	1	0	0	1
6	0	1	1	0	1	0	0	1	1	0	0	1
7	0	1	1	1	0	1	0	0	0	1	0	1
8	1	0	0	0	0	0	0	1	1	0	0	1
9	1	0	0	1	0	0	0	0	1	0	0	1
10	1	0	1	0	X	X	X	X	X	X	X	X
11	1	0	1	1	X	X	X	X	X	X	X	X
12	1	1	0	0	X	X	X	X	X	X	X	X
13	1	1	0	1	X	X	X	X	X	X	X	X
14	1	1	1	0	X	X	X	X	X	X	X	X
15	1	1	1	1	X	X	X	X	X	X	X	X

3.- Mapas de Karnaugh

Los mapas de Karnaugh de las entradas de los biestables son:



4.- Circuito del contador implementado

A continuación, se muestra el circuito del contador que se ha implementado:

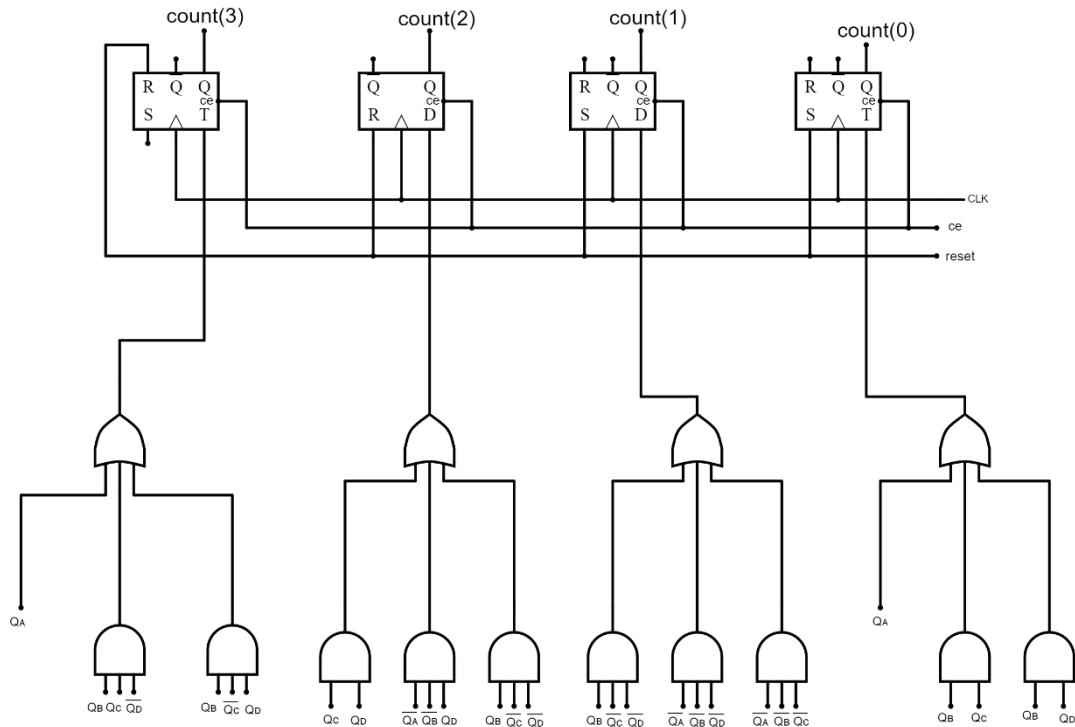


Figura 1. Circuito implementado del contador 119.

5.- Códigos VHDL

En este apartado se muestra el código VHDL del contador implementado:

```

1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3
4  entity contador119 is
5      Port ( clk      : in std_logic;
6            ce       : in std_logic;
7            reset    : in std_logic;
8            count    : out std_logic_vector (3 downto 0));
9  end contador119;
10

```

```
1  architecture Behavioral of contador119 is
2
3      component ffD_reset
4          port (
5              clk      : in std_logic;
6              ce       : in std_logic;
7              reset    : in std_logic;
8              d        : in std_logic;
9              q        : out std_logic
10             );
11  end component;
12
13  component ffT_preset
14      port (
15          clk      : in std_logic;
16          ce       : in std_logic;
17          preset   : in std_logic;
18          t        : in std_logic;
19          q        : out std_logic
20      );
21  end component;
22
23  component ffT_reset
24      port (
25          clk      : in std_logic;
26          ce       : in std_logic;
27          reset    : in std_logic;
28          t        : in std_logic;
29          q        : out std_logic
30      );
31  end component;
32
33  component ffD_preset
34      port (
35          clk      : in std_logic;
36          ce       : in std_logic;
37          preset   : in std_logic;
38          d        : in std_logic;
39          q        : out std_logic
40      );
41  end component;
42
43  signal qa, qb, qc, qd : std_logic;
44  signal ta, db, dc, td : std_logic;
```

```
1  begin
2
3      unitA: ffT_reset port map (
4          clk    => clk,
5          ce     => ce,
6          reset  => reset,
7          t      => ta,
8          q      => qa
9      );
10
11     unitB: ffD_reset port map (
12         clk    => clk,
13         ce     => ce,
14         reset  => reset,
15         d      => db,
16         q      => qb
17     );
18
19     unitC: ffD_preset port map (
20         clk    => clk,
21         ce     => ce,
22         preset => reset,
23         d      => dc,
24         q      => qc
25     );
26
27     unitD: ffT_preset port map (
28         clk    => clk,
29         ce     => ce,
30         preset => reset,
31         t      => td,
32         q      => qd
33     );
34
35     ta <= (qa) or (qb and qc and not qd) or (qb and not qc and qd);
36     db <= (qc and qd) or (not qa and not qb and qd) or (qb and not qc and not qd);
37     dc <= (qb and not qc and not qd) or (not qa and not qb and not qd) or (not qa and not qb and not qc);
38     td <= (qa) or (qb and qc) or (qb and qd);
39
40     count <= qa & qb & qc & qd;
41
42 end Behavioral;
```

Figura 2. Código VHDL del contador 119.

También se muestra el código VHDL del test bench.

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3
4  entity sistema119_tb is
5  -- Port ( );
6  end sistema119_tb;
7
8  architecture Behavioral of sistema119_tb is
9
10     component sistema119
11         port (clk    : in std_logic;
12              ce     : in std_logic;
13              reset   : in std_logic;
14              count   : out std_logic_vector (3 downto 0);
15              led     : out std_logic_vector (6 downto 0));
16     end component;
17
18     signal clk    : std_logic := '0';
19     signal ce     : std_logic := '0';
20     signal reset   : std_logic := '0';
21     signal count   : std_logic_vector (3 downto 0);
22     signal led     : std_logic_vector (6 downto 0);
23
24     begin
25
26         dut : sistema119
27         port map (clk => clk,
28                 ce  => ce,
29                 reset => reset,
30                 count => count,
31                 led  => led);
32
33         -- Clock Generation
34         clk <= not clk after 10 ns;
35
36         stimuli : process
37         begin
38
39             ce <= '0'; reset <= '1';
40             wait for 20 ns;
41             reset <= '0'; ce <= '1';
42             wait for 180 ns;
43             reset <= '1';
44         end process;
45
46     end Behavioral;
```

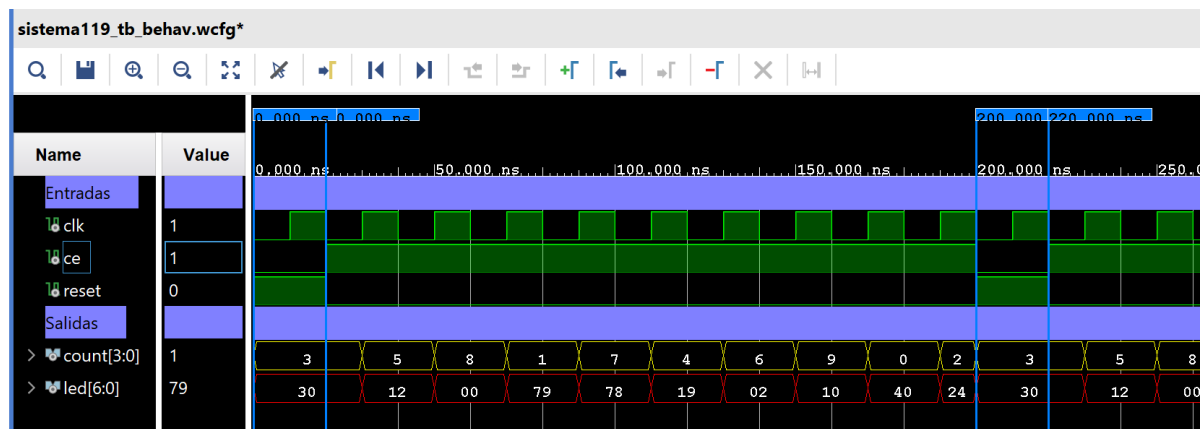

Figura 3. Código VHDL del test bench.

6.- Cronograma de simulación

Instrucciones para realizar la simulación:

- Crea un pulso de reset de 0ns a 20ns.
- Activa la señal ce después de 20ns.
- Vuelve a activar el reset a los 180ns. se resetea el sistema para comprobar que volvemos al estado de partida.
- Se debe añadir 2 divisores de señales (Entradas y Salidas)
- Count en amarillo y Led en rojo.
- Marcar en el cronograma las etapas de reset y la etapa inicial de ce. Sugerencias: se puede realizar en el propio cronograma según criterio propio.

A continuación, en la Figura 3, se muestra el cronograma de simulación del sistema implementado con las salidas del contador y del decodificador de 7 segmentos.

**Figura 4.** Cronograma de simulación del contador 119 con la secuencia completa recorrida.

Los resultados coinciden con la secuencia asignada que aparece en la tabla del apartado 1 (copia la tabla 1 del apartado 1 en esta sección):

Alumno	Secuencia	Flip-flops				Cíclico
119 Acoidan Martín Conrado	3,5,8,1,7,4,6,9,0,2	T	D	D	T	No

Tabla 1. Contador aleatorio asignado

7.- Realización de vídeo

Generar un vídeo con la tarjeta Nexys A7 en el que se aprecie:

1. La secuencia entera correcta.
2. Los funcionamientos de reset y ce.

Video: <https://youtu.be/at4IEZEKSBc>