

Entregable 2

Contador aleatorio

*Rafael Gómez Guillén
alu0101462578@ull.edu.es
1º Grado Informática
Universidad de La Laguna
13/5/2022*

Índice

1. Introducción y objetivos1
2. Tabla de transiciones1
3. Mapas de Karnaugh2
4. Circuito del contador implementado3
5. Código VHDL del contador3
6. Cronograma de simulación5

1.- Introducción y objetivos

El contador aleatorio asignado es el siguiente:

Alumno	Secuencia	Flip-flops				Cíclico
86 Rafael Gómez Guillén	4,5,0,6,7,9,3,2,1,8	T	D	D	T	No

2.- Tabla de transiciones

La tabla de transiciones del contador aleatorio asignado 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	1	1	0	0	1	1	0
1	0	0	0	1	1	0	0	0	1	0	0	1
2	0	0	1	0	0	0	0	1	0	0	0	1
3	0	0	1	1	0	0	1	0	0	0	1	1
4	0	1	0	0	0	1	0	1	0	1	0	1
5	0	1	0	1	0	0	0	0	0	0	0	1
6	0	1	1	0	0	1	1	1	0	1	1	1
7	0	1	1	1	1	0	0	1	1	0	0	0
8	1	0	0	0	1	0	0	0	0	0	0	0
9	1	0	0	1	0	0	1	1	1	0	1	0
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:

$T_a = (\text{not } Q_b \text{ and not } Q_c \text{ and } Q_d) \text{ or } (Q_b \text{ and } Q_c \text{ and } Q_d)$

- Rojo = (not Q_b and not Q_c and Q_d)
- Azul = (Q_b and Q_c and Q_d)

$Q_a, Q_b/Q_c, Q_d$	00	01	11	10
00	0	1	0	0
01	0	0	1	0
11	x	x	x	x
10	0	1	x	x

$D_b = (\text{not } Q_a \text{ and not } Q_c \text{ and not } Q_d) \text{ or } (Q_b \text{ and not } Q_d)$

- Rojo = (not Q_a and not Q_c and not Q_d)
- Azul = (Q_b and not Q_d)

$Q_a, Q_b/Q_c, Q_d$	00	01	11	10
00	1	0	0	0
01	1	0	0	1
11	x	x	x	x
10	0	0	x	x

$D_c = (\text{not } Q_a \text{ and not } Q_b \text{ and not } Q_c \text{ and not } Q_d) \text{ or } (\text{not } Q_b \text{ and } Q_c \text{ and } Q_d) \text{ or } (Q_b \text{ and } Q_c \text{ and not } Q_d) \text{ or } (Q_a \text{ and } Q_d)$

Rojo = (not Q_a and not Q_b and not Q_c and not Q_d)

Azul = (Q_b and Q_c and not Q_d)

Amarillo = (not Q_b and Q_c and Q_d)

Verde = (Q_a and Q_d)

$Q_a, Q_b/Q_c, Q_d$	00	01	11	10
00	1	0	1	0
01	0	0	0	1
11	x	x	x	x
10	0	1	x	x

$T_d = (\text{not } Q_a \text{ and not } Q_b \text{ and } Q_d) \text{ or } (Q_c \text{ and not } Q_d) \text{ or } (Q_b \text{ and not } Q_c)$

$\text{Rojo} = (Q_c \text{ and not } Q_d)$

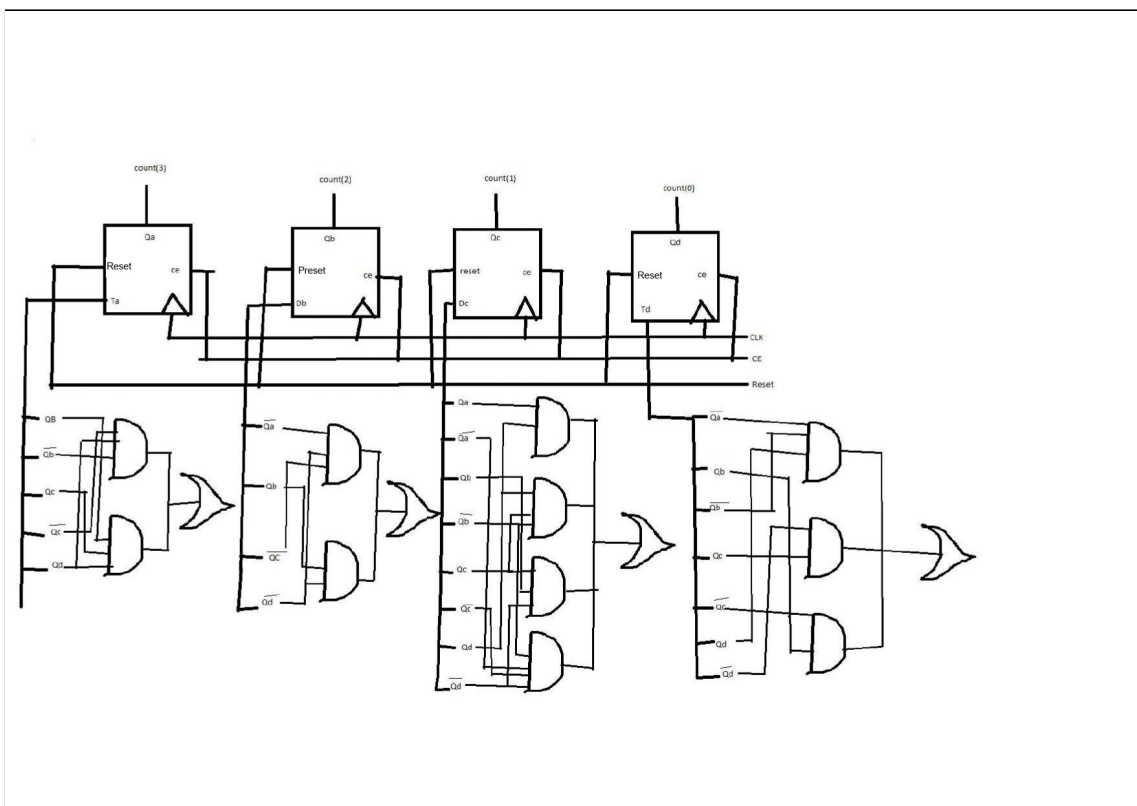
$\text{Azul} = (Q_b \text{ and not } Q_c)$

$\text{Verde} = (\text{not } Q_a \text{ and not } Q_b \text{ and } Q_d)$

$Q_a, Q_b / Q_c, Q_d$	00	01	11	10
00	0	1	1	1
01	1	1	0	1
11	x	x	x	x
10	0	0	x	x

4.- Circuito del contador implementado

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



5.- Código VHDL del contador

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

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity contador86 is
    Port ( ce : in  STD_LOGIC;
          reset : in  STD_LOGIC;
          count : out  STD_LOGIC_VECTOR (3 downto 0);
          clk : in  STD_LOGIC);
end contador86;

architecture Behavioral of contador86 is

    COMPONENT fft_reset
    PORT(
        reset : IN STD_LOGIC;
        clk : IN STD_LOGIC;
        ce : IN STD_LOGIC;
        t : IN STD_LOGIC;
        q : OUT STD_LOGIC
    );
END COMPONENT;

    COMPONENT ffD_preset
    PORT(
        clk : IN STD_LOGIC;
        preset : IN STD_LOGIC;
        ce : IN STD_LOGIC;
        d : IN STD_LOGIC;
        q : OUT STD_LOGIC
    );
END COMPONENT;

    COMPONENT ffD_reset
    PORT(
        clk : IN STD_LOGIC;
        reset : IN STD_LOGIC;
        ce : IN STD_LOGIC;
        d : IN STD_LOGIC;
        q : OUT STD_LOGIC
    );
END COMPONENT;

    signal ta, db, dc, td : STD_LOGIC;
    signal qa, qb, qc, qd : STD_LOGIC;

begin

    unitA: fft_reset PORT MAP(
```

```

        reset => reset,
        clk => clk,
        ce => ce,
        t => ta,
        q => qa
    );

    unitB: ffD_preset PORT MAP(
        clk => clk,
        preset => reset,
        ce => ce,
        d => db,
        q => qb
    );

    unitC: ffD_reset PORT MAP(
        clk => clk,
        reset => reset,
        ce => ce,
        d => dc,
        q => qc
    );

    unitD: ffT_reset PORT MAP(
        reset => reset,
        clk => clk,
        ce => ce,
        t => td,
        q => qd
    );

    ta <= (not qb and not qc and qd) or (qb and qc and
qd);
    db <= (not qa and not qc and not qd) or (qb and not
qd);
    dc <= (not qa and not qb and not qc and not qd) or
(not qb and qc and qd) or (qb and qc and not qd) or (qa and
qd);
    td <= (not qa and not qb and qd) or (qc and not qd) or
(qb and not qc);

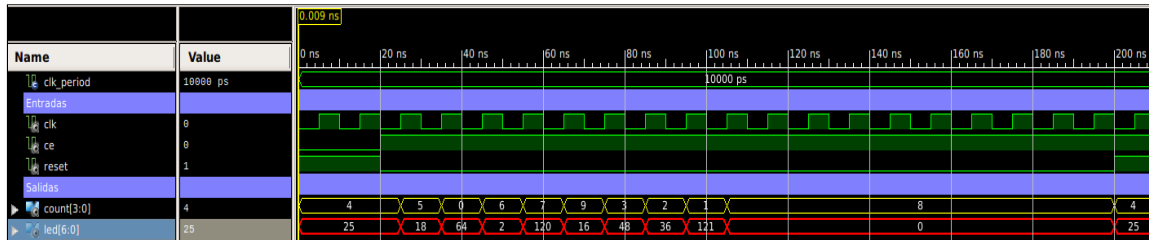
    count <= qa & qb & qc & qd;

end Behavioral;

```

6.- Cronograma de simulación

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



No se inserta ningún video, porque el profesor ya vio la implementación del contador.