

# *Entregable 2*

*Contador aleatorio*

**CARLOS EDUARDO VARGAS TORREALBA**

***Alu0101604077@ull.edu.es***

***1º Grado Informática***

***Universidad de La Laguna***

***30/3/2023***

# Índice

1. Introducción y objetivos.....	1
2. Tabla de transiciones.....	1
3. Mapas de Karnaugh.....	2
4. Circuito del contador implementado.....	3
5. Código VHDL del contador .....	3
6. Cronograma de simulación .....	5

## 1.- Introducción y objetivos

En esta práctica se va a proceder a la implementar un contador aleatorio. Éste tendrá como señales de control una señal de reinicio (reset) y una señal de reloj; y como salidas se muestra la cuenta en binario y codificada para su representación en un display de 7 segmentos.

El contador aleatorio asignado es el siguiente:

Alumno	Secuencia	Flip-flops				Cíclico
210 Vargas Torrealba, Carlos Eduardo	1,9,8,7,5,0,4,3,2,6	T	D	D	D	Sí

## 2.- Tabla de transiciones

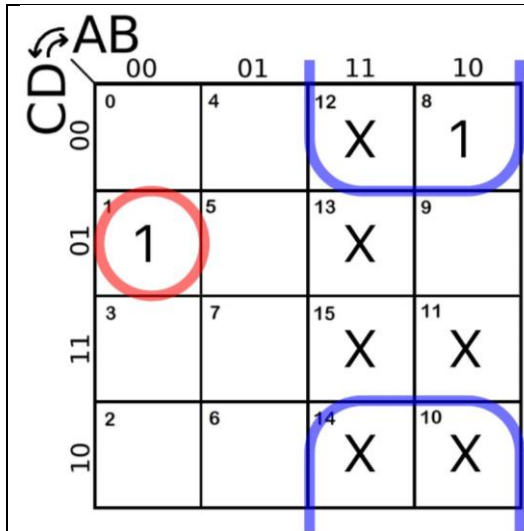
La tabla de transiciones del contador aleatorio asignado es:

	Estado actual				Estado siguiente							
	Q <sub>A</sub>	Q <sub>B</sub>	Q <sub>C</sub>	Q <sub>D</sub>	Q <sub>A</sub> <sup>+</sup>	Q <sub>B</sub> <sup>+</sup>	Q <sub>C</sub> <sup>+</sup>	Q <sub>D</sub> <sup>+</sup>	T <sub>A</sub>	D <sub>B</sub>	D <sub>C</sub>	T <sub>D</sub>
0	0	0	0	0	0	1	0	0	0	1	0	0
1	0	0	0	1	1	0	0	1	1	0	0	1
2	0	0	1	0	0	1	1	0	0	1	1	0
3	0	0	1	1	0	0	1	0	0	0	1	0
4	0	1	0	0	0	0	1	1	0	0	1	1
5	0	1	0	1	0	0	0	0	0	0	0	0
6	0	1	1	0	0	0	0	1	0	0	0	1
7	0	1	1	1	0	1	0	1	0	1	0	1
8	1	0	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	0	0	0	0	0	0	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

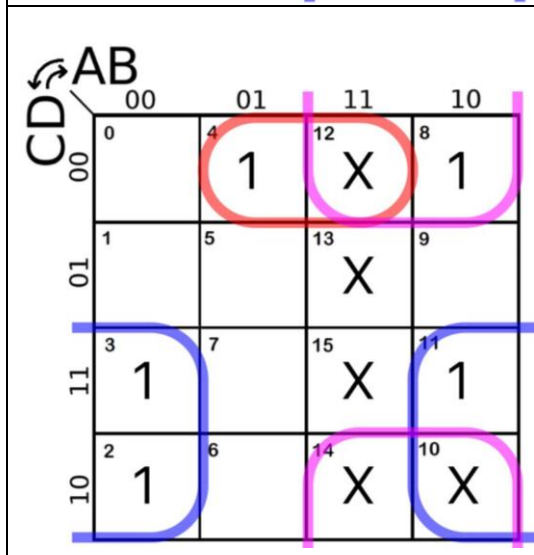
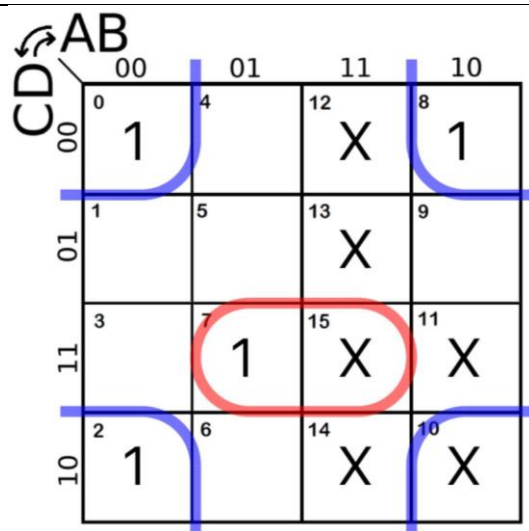
### 3.- Mapas de Karnaugh

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

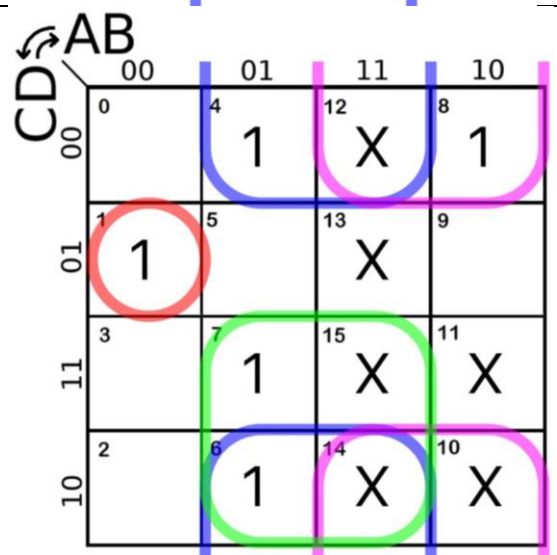
MAPA 1



MAPA 2



MAPA 3



MAPA 4

MAPA 1:  $Ta(Qa, Qb, Qc, Qd) = Qa' Qb' Qc' Qd + QaQd'$

**ROJO:**  $Qa' Qb' Qc' Qd$

**AZUL:**  $QaQd'$

MAPA 2:  $Ta(Qa, Qb, Qc, Qd) = QbQcQd + Qb'Qd'$

**ROJO:**  $QbQcQd$

**AZUL:**  $Qb'Qd'$

MAPA 3:  $Ta(Qa, Qb, Qc, Qd) = QbQc'Qd' + Qb'Qc + QaQd'$

**ROJO:**  $QbQc'Qd'$

**AZUL:**  $Qb'Qc$

**MORADO:**  $QaQd'$

MAPA 4:  $Ta(Qa, Qb, Qc, Qd) = Qa'Qb'Qc'Qd + QbQd' + QaQd' + QbQc$

**ROJO:**  $Qa'Qb'Qc'Qd$

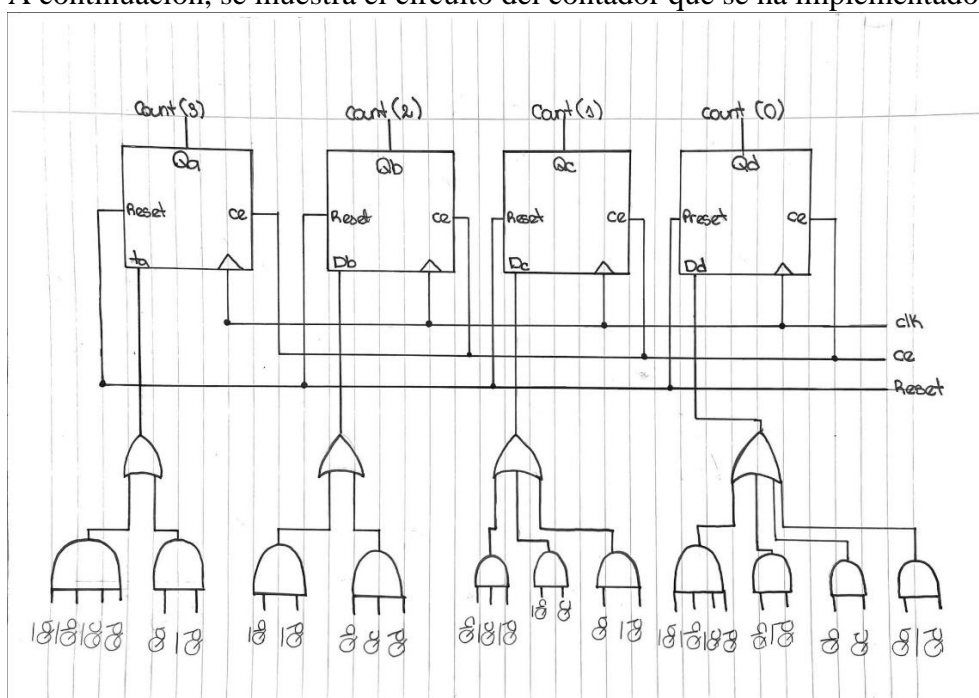
**AZUL:**  $QbQd'$

**MORADO:**  $QaQd'$

**VERDE:**  $QbQc$

## 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;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity contador is
  Port ( ce : in  STD_LOGIC;
        reset : in  STD_LOGIC;
        count : out STD_LOGIC_VECTOR (3 downto 0);
        clk : in  STD_LOGIC);
end contador;

architecture Behavioral of contador 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_reset
    PORT(
      clk : IN STD_LOGIC;
      reset : IN STD_LOGIC;
      ce : IN STD_LOGIC;
      d : 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;

signal ta, db, dc, dd : 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_reset PORT MAP(
    clk => clk,
    reset => reset,
    ce => ce,
    d => db,
    q => qb
);

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

unitD: ffD_preset PORT MAP(
    clk => clk,
    preset => reset,
    ce => ce,
    d => dd,
    q => qd
);

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

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.

