# *Entregable 1*

## *Minimización de funciones y simulación con ISE*

*Carlos Eduardo Vargas Torrealba*
*1º Grado Informática*
*Universidad de La Laguna*
*10/3/2023*

**Índice**

# 1.- Introducción y objetivos

En esta práctica se va a proceder a la simplificación de funciones booleanas empleando los mapas de Karnaugh. A continuación, se muestra el código VHDL de dichas funciones lógicas simplificadas, el test-bench y la simulación.

Las funciones lógicas asignadas son las siguientes:

| Alumno | F1 | F2 |
|---|---|---|
| 210 Vargas Torrealba Carlos Euduardo | 2BF4 | 2E72 |

# 2.- Tablas de verdad

Las tablas de verdad de las funciones son:

| | abcd | f1 | f2 |
|---|---|---|---|
| 0 | 0000 | 0 | 0 |
| 1 | 0001 | 0 | 1 |
| 2 | 0010 | 1 | 0 |
| 3 | 0011 | 0 | 0 |
| 4 | 0100 | 1 | 1 |
| 5 | 0101 | 1 | 1 |
| 6 | 0110 | 1 | 1 |
| 7 | 0111 | 1 | 0 |
| 8 | 1000 | 1 | 0 |
| 9 | 1001 | 1 | 1 |
| 10 | 1010 | 0 | 1 |
| 11 | 1011 | 1 | 1 |
| 12 | 1100 | 0 | 0 |
| 13 | 1101 | 1 | 1 |
| 14 | 1110 | 0 | 0 |
| 15 | 1111 | 0 | 0 |

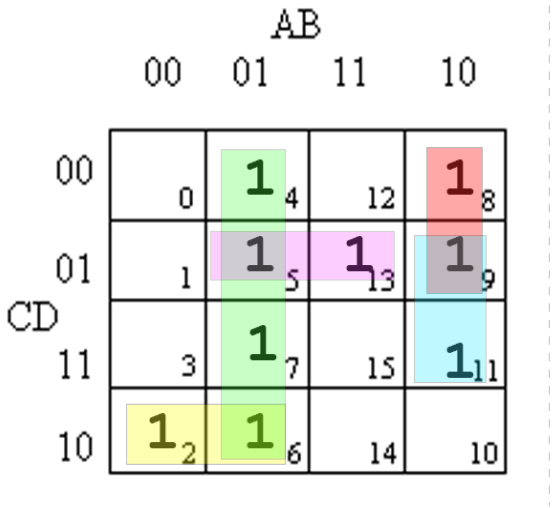
## 3.- Mapas de Karnaugh

Los mapas de Karnaugh de las funciones son:

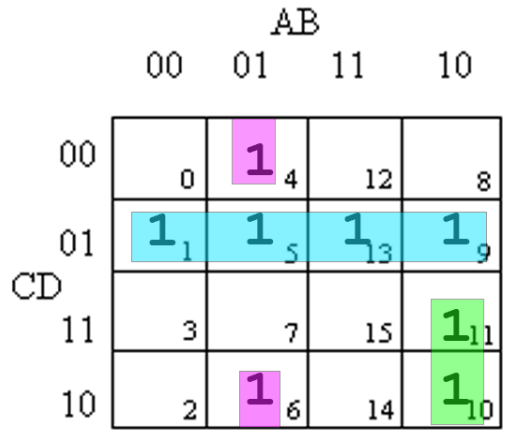
Para F1=F(A,B,C,D) = (2,4,5,6,7,8,9,11,13)

- (4, 5, 6, 7) = $\overline{C}D$
- (2, 6) = $A\overline{B}\overline{C}$
- (5, 13) = $\overline{A}BC$
- (8, 9) = $\overline{A}C\overline{D}$
- (9, 11) = $BC\overline{D}$

F1= $\overline{C}D+A\overline{B}\overline{C}+\overline{A}C\overline{D}+BC\overline{D}+\overline{A}BC$




Para F2= f(A,B,C,D) = (1,4,5,6,9,10,11,13)



- (4, 6) = a’bd’
- (1, 5, 9, 13) = c’d
- (10, 11) = ab’c

F2 = c'd + a'bd' + ab'c

## 4.- Código VHDL

```
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 p1 is
    Port ( a : in  STD_LOGIC;
           b : in  STD_LOGIC;
           c : in  STD_LOGIC;
           d : in  STD_LOGIC;
           f1 : out  STD_LOGIC;
           f2 : out  STD_LOGIC);
end p1;

architecture Behavioral of p1 is

begin
     f1 <= (not a and b) or (not a and c and not d) or (a
and not b and d) or (b and not c and d) or (a and not b and
not c);

     f2 <= (not a and b and not d) or (a and not b and c)
or (not c and d);

end Behavioral;
```

## 5.- Test bench

```
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;

ENTITY p1_tb IS
```

```
END p1_tb;

ARCHITECTURE behavior OF p1_tb IS

    -- Component Declaration for the Unit Under Test (UUT)

    COMPONENT p1
    PORT(
         a : IN  std_logic;
         b : IN  std_logic;
         c : IN  std_logic;
         d : IN  std_logic;
         f1 : OUT  std_logic;
         f2 : OUT  std_logic
        );
    END COMPONENT;


   --Inputs
   signal a : std_logic := '0';
   signal b : std_logic := '0';
   signal c : std_logic := '0';
   signal d : std_logic := '0';

     --Outputs
   signal f1 : std_logic;
   signal f2 : std_logic;

     --Control
     constant Tb_semiPeriod : time := 10 ns;
     signal TbSimEnded : std_logic := '0';


BEGIN

     -- Instantiate the Unit Under Test (UUT)
   uut: p1 PORT MAP (
          a => a,
          b => b,
          c => c,
          d => d,
          f1 => f1,
          f2 => f2
        );

     d <= not d after Tb_semiPeriod    when TbSimEnded /=
'1' else '0';
     c <= not c after Tb_semiPeriod*2 when TbSimEnded /=
'1' else '0';
     b <= not b after Tb_semiPeriod*4 when TbSimEnded /=
'1' else '0';
```

```
     a <= not a after Tb_semiPeriod*8 when TbSimEnded /=
'1' else '0';

     --Stimulus process
     stim_proc: process
     begin
          wait for Tb_semiPeriod*16;
               TbSimEnded <= '1';
          wait;
     end process;

END;
```

## 6.- Cronograma de simulación

A continuación, se muestra el cronograma de simulación de las funciones implementadas: