CONESTOGA
Connect Life and Learning

# DIGITAL DESIGN PRINCIPLES

# LAB 7 – SEVEN SEGMENT DISPLAY

**Name & Surname :** Sinan KARACA

**Date :** 23 / 07 / 2021

**Student ID :** 8743013

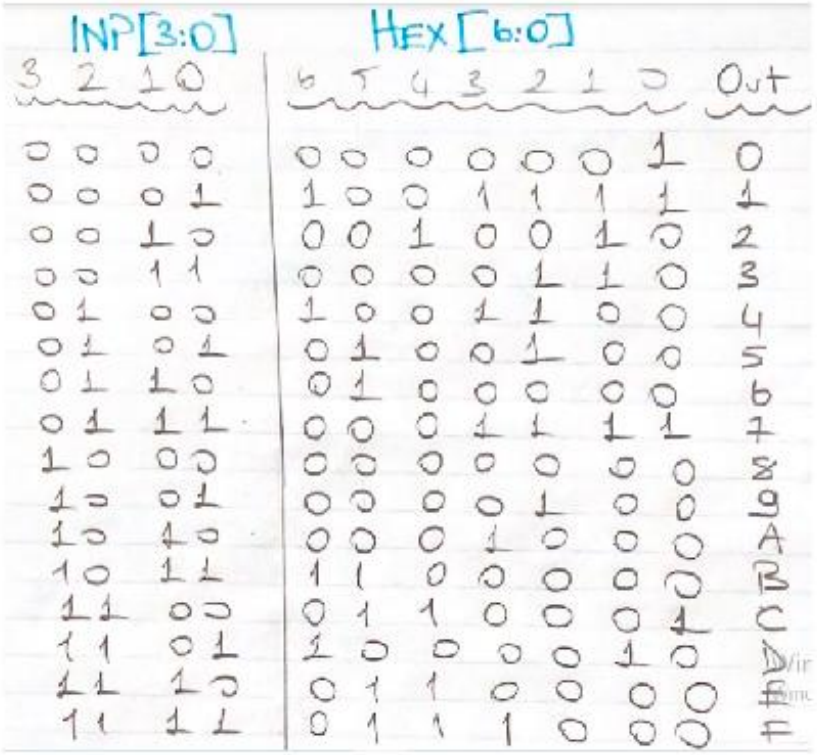
## INTRODUCTION

The purpose of the experiment is to run the 7 segment display with fpga board wit VHDL and Verilog codes. The instruction of lab notes says the 7 segment is common anode, so in the design, leds on 7 segments are high when regarding ouput is 0, off when regarding output is 1.

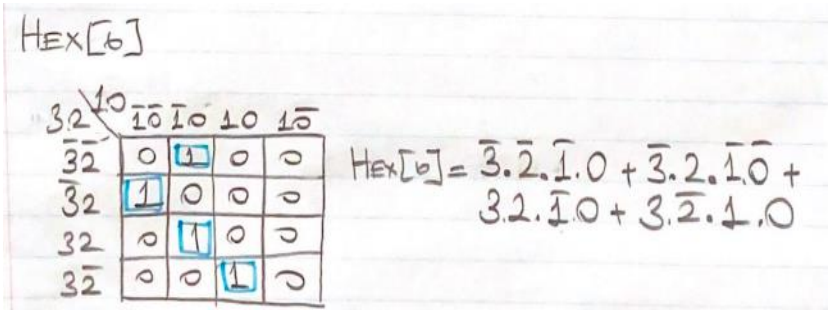
## WORKING PRINCIPLE

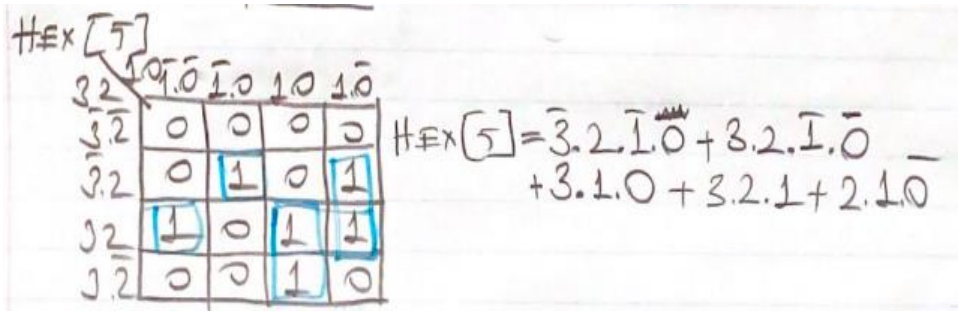
## HALF ADDER

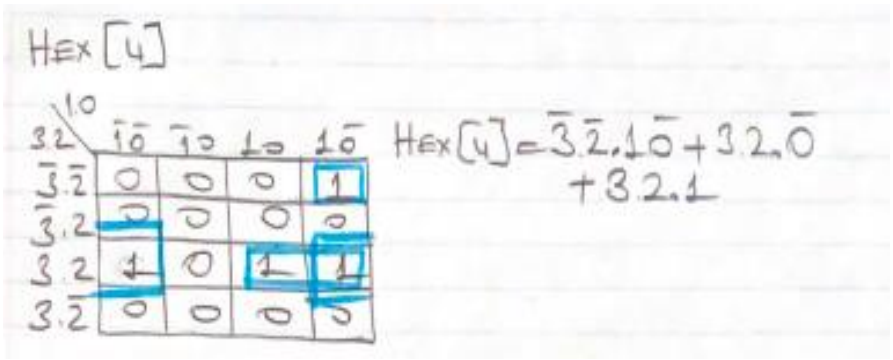
The design is implemented from K map of each output from the truth table of 7 segment display, also it is proven with the multisim program. Truth table is Show down below,

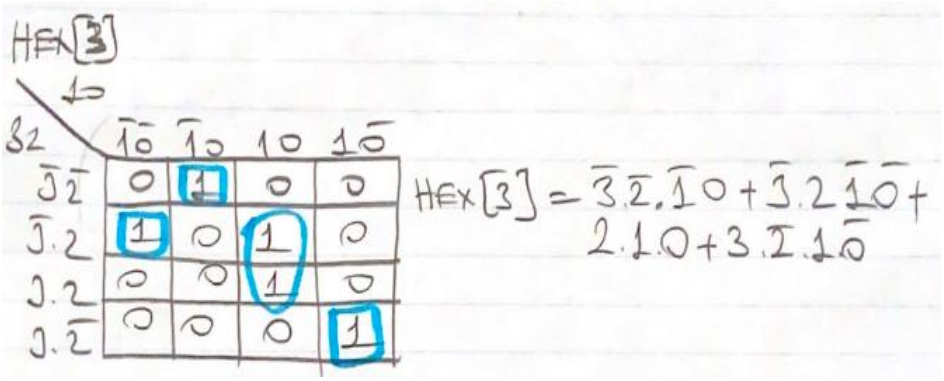


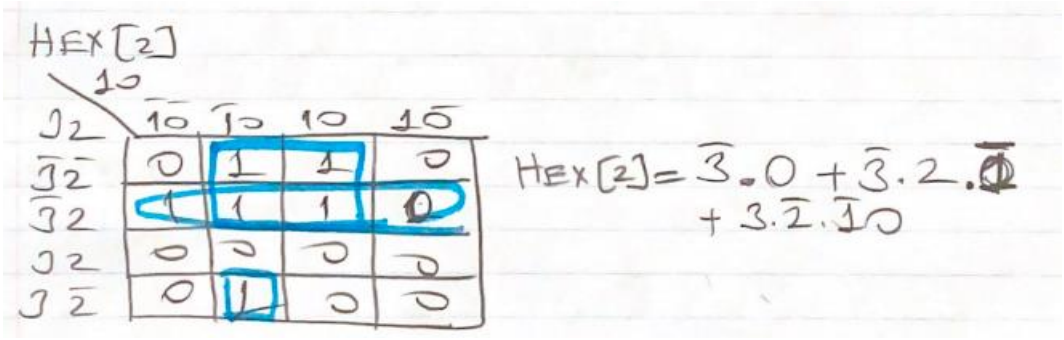
Karnough maps of each outputs shown below,

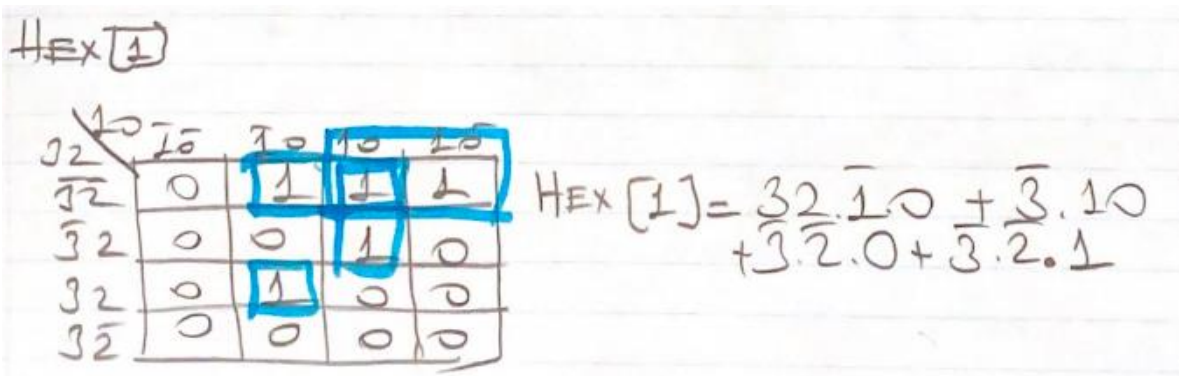


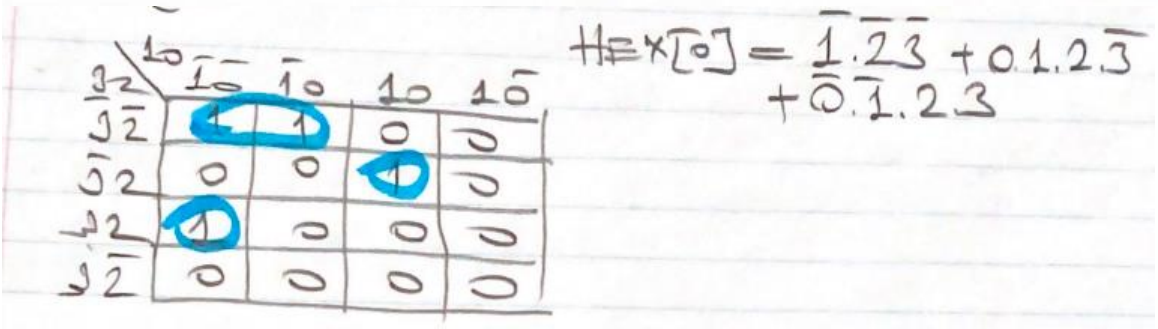



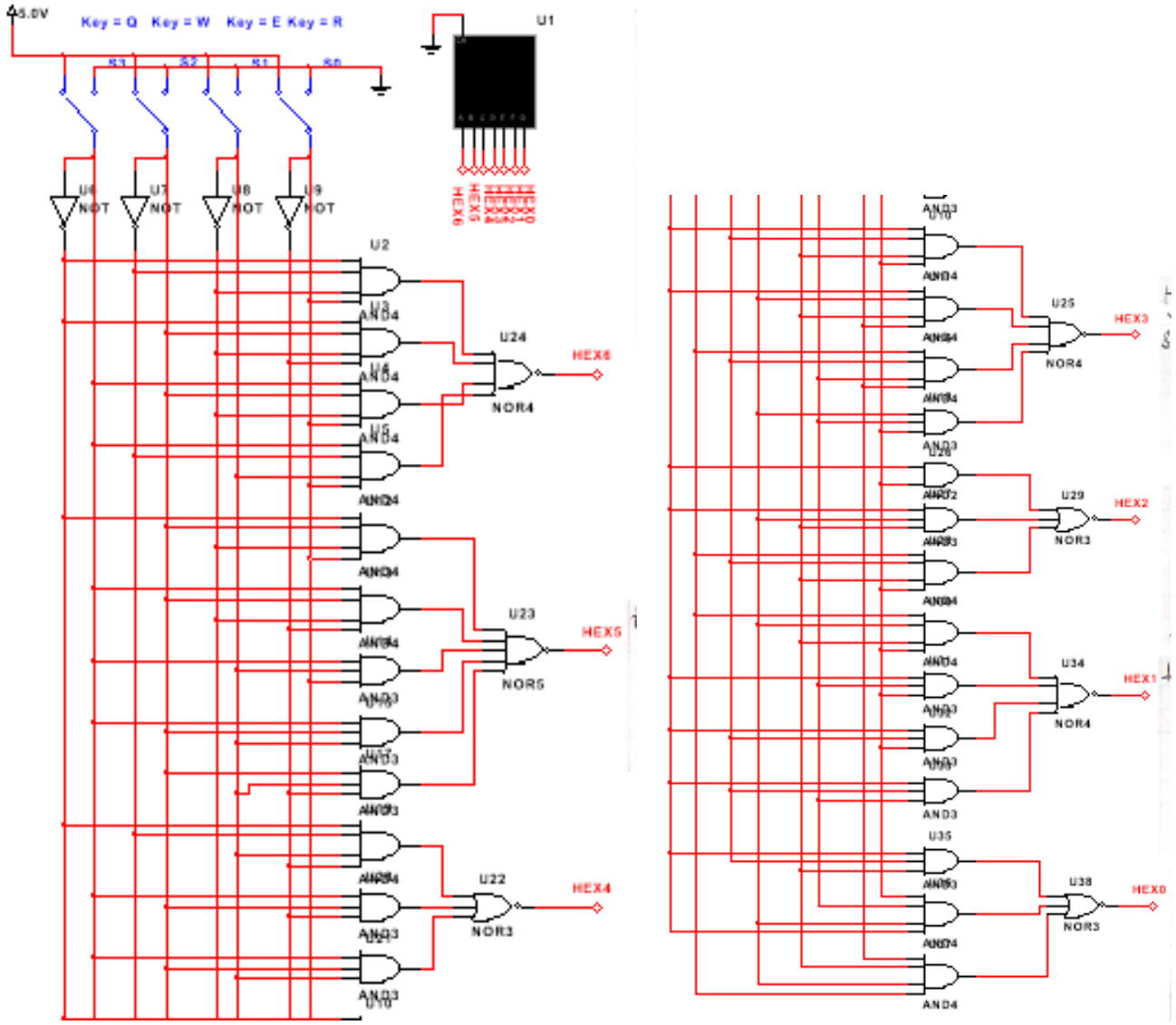



Derived logic circuit shown down below, the circuit created with multisim. Before implementing VHDL and Verilog Codes, the design approved with multisim program.

# LOGIC CIRCUIT OF DIAGRAM (MULTISIM)



# VHDL MAIN PROGRAM

```
----------------------------------------
----------------------------------------
-- Project Name  : 7 Segment Display
-- File          : skaraca_lab7
-- Creator       : Sinan KARACA
-- Student number: 8743013
-- Date          : 22.07.2021
----------------------------------------
----------------------------------------

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity skaraca_Lab6 is
   port(
      INP    : in STD_LOGIC_VECTOR(3 downto 0);
      HEX    : out STD_LOGIC_VECTOR(6 downto 0));
   end skaraca_Lab6;


architecture Behavioral of skaraca_Lab6 is


--Component initialization
component OutputComponent is
   port(
      Input     : in STD_LOGIC_VECTOR(3 downto 0);
      Output    : out STD_LOGIC_VECTOR(6 downto 0));
   end component OutputComponent;


begin

--Component declerations
   H1: OutputComponent port map (INP, HEX);


end Behavioral;
```

# VHDL COMPONENT PROGRAM

```
----------------------------------------
----------------------------------------
-- Project Name  : OutputComponent
-- File          : skaraca_lab4
-- Creator       : Sinan KARACA
-- Student number: 8743013
-- Date          : 14.07.2021
----------------------------------------
----------------------------------------

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity OutputComponent is
   port(
      Input     : in STD_LOGIC_VECTOR(3 downto 0);
      Output    : out STD_LOGIC_VECTOR(6 downto 0));
   end OutputComponent;


architecture Behavioral of OutputComponent is

begin

   Output(6) <=((not Input(3)) and (not Input(2)) and ( not Input(1)) and Input(0)) or
               ((not Input(3)) and Input(2) and (not Input(1)) and  (not Input(0))) or
               (Input(3) and Input(2) and (not Input(1)) and Input(0)) or
               (Input(3) and (not Input(2)) and Input(1) and Input(0));

   Output(5) <= ((not Input(3)) and Input(2) and (not Input(1)) and Input(0)) or
               ( Input(3) and Input(2) and (not Input(1)) and (not Input(0))) or
               (Input(3) and Input(1) and Input(0)) or (Input(3) and Input(2) and Input(1)) or
               (Input(2) and Input(1) and (not Input(0)));


   Output(4) <= ((not Input(3)) and (not Input(2)) and Input(1) and (not Input(0))) or
               (Input(3) and Input(2) and (not Input(0))) or (Input(3) and Input(2) and Input(1));

   Output(3) <= ((not Input(3)) and (not Input(2)) and (not Input(1)) and Input(0)) or
               ((not Input(3)) and Input(2) and (not Input(1)) and (not Input(0))) or
               (Input(2) and Input(1) and Input(0)) or (Input(3) and ( not Input(2)) and
               Input(1) and (not Input(0)));

   Output(2) <= ((not Input(3)) and Input(0) ) or ((not Input(3)) and Input(2) and (not Input(1)))
                or (Input(3) and (not Input(2)) and (not Input(1)) and Input(0));




   Output(1) <= (Input(3) and Input(2) and (not Input(1)) and Input(0)) or ((not Input(3))
               and Input(1) and Input(0)) or ((not Input(3)) and (not Input(2)) and Input(0))
               or ((not Input(3)) and (not Input(2)) and Input(1));

   Output(0) <= ((not Input(1)) and (not Input(2)) and (not Input(3)))
               or (Input(0) and Input(1) and Input(2) and  (not Input(3)))
               or ( (not Input(0)) and (not Input(1) ) and Input(2) and Input(3));


end Behavioral;


 endmodule
```

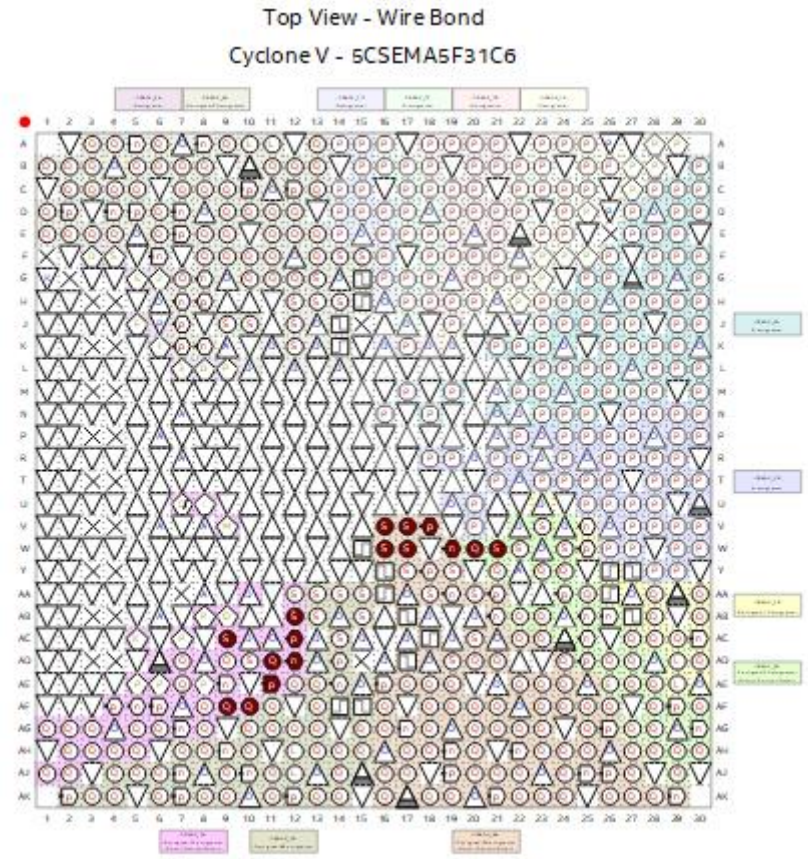
# VERILOG MAIN CODE

```
//////////////////////////////////////////////////////////////////////////////////
//////////////////////////////////////////////////////////////////////////////////
// Project Name  : sevenSegment
// File          : sevenSegment
// Creator       : Sinan KARACA
// Student number: 8743013
// Date          : 14.07.2021
//////////////////////////////////////////////////////////////////////////////////
//////////////////////////////////////////////////////////////////////////////////

module skaraca6_verilog(INP, HEX);

    input  [3:0]INP;
    output [6:0]HEX;

    OutputComponent(INP ,  HEX);


endmodule
```

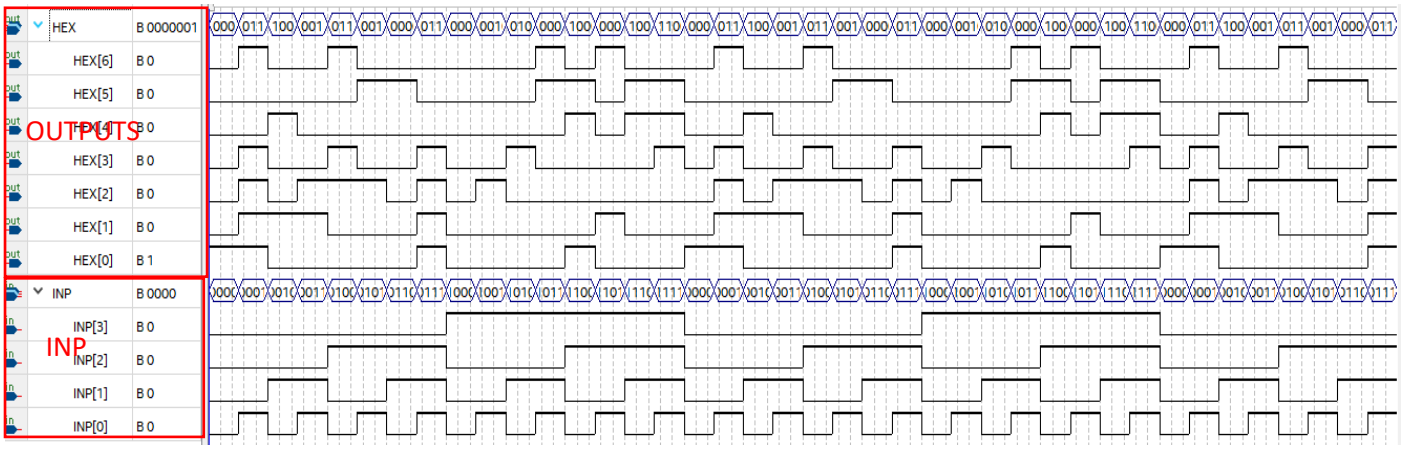
# VERILOG COMPONENT CODE

```
//////////////////////////////////////////
//////////////////////////////////////////
// Project Name  : OutputComponent
// File          : OutputComponent
// Creator       : Sinan KARACA
// Student number: 8743013
// Date          : 21.07.2021
//////////////////////////////////////////
//////////////////////////////////////////

module OutputComponent(Input, Output);

    input  [3:0]Input;
    output [6:0]Output;

    assign Output[6] = (( !Input[3] & !Input[2]) & !Input[1] & Input[0]) |
                ((! Input[3]) & Input[2] & (! Input[1]) &  (! Input[0])) |
                (Input[3] & Input[2] & (! Input[1]) & Input[0]) |
                (Input[3] & (! Input[2]) & Input[1] & Input[0]);

    assign Output[5] = ((! Input[3]) & Input[2] & (! Input[1]) & Input[0]) |
                ( Input[3] & Input[2] & (! Input[1]) & (! Input[0])) |
                (Input[3] & Input[1] & Input[0]) | (Input[3] & Input[2] & Input[1]) |
                (Input[2] & Input[1] & (! Input[0]));


    assign Output[4] = ((! Input[3]) & (! Input[2]) & Input[1] & (! Input[0])) |
                (Input[3] & Input[2] & (! Input[0])) | (Input[3] & Input[2] & Input[1]);

    assign Output[3] = ((! Input[3]) & (! Input[2]) & (! Input[1]) & Input[0]) |
                ((! Input[3]) & Input[2] & (! Input[1]) & (! Input[0])) |
                (Input[2] & Input[1] & Input[0]) | (Input[3] & ( ! Input[2]) &
                Input[1] & (! Input[0]));

    assign Output[2] = ((! Input[3]) & Input[0] ) | ((! Input[3]) & Input[2] & (! Input[1]))
                 | (Input[3] & (! Input[2]) & (! Input[1]) & Input[0]);

    assign Output[1] = (Input[3] & Input[2] & (! Input[1]) & Input[0]) | ((! Input[3])
                & Input[1] & Input[0]) | ((! Input[3]) & (! Input[2]) & Input[0])
                | ((! Input[3]) & (! Input[2]) & Input[1]);

    assign Output[0] = ((! Input[1]) & (! Input[2]) & (! Input[3]))
                | (Input[0] & Input[1] & Input[2] &  (! Input[3]))
                | ( (! Input[0]) & (! Input[1] ) & Input[2] & Input[3]);
```

# PIN ASSIGNMENT




# SIMALATION WAVEFORM




Simulation waveform shows the all combinations of 4 bit inputs. 16 different input are shown up below, it is ascended to 0 to F.

# CONCLUSION

Design of 7 segment display control through vhdl and verilog has been experienced in this lab. There are others methods which are more easier and understandable to create, but I choose to gather the design from K-Maps of output signals. To learn more about and make practise about K-Maps. Also, the result of the K-Maps are implemented to multisim platform to prove, if the K-Maps result works fine.