UNIVERSITY OF CANBERRA

INTRODUCTION TO COMPUTER ENGINEERING (8223/10096)

TUTORIAL WEEK 12 Assignment

FACULTY OF SCIENCE AND TECHNOLOGY

Assignment (Laboratory) Coversheet

Student ID number	U3312590
Student Name	Jasmine Bajracharya
Unit name	Introduction to Computer Engineering
Unit number	10096
Name of lecturer/tutor	Dr. Julio Romero
Assignment topic	HDL and Microprocessors
Due date	2nd November, 2025
Word Count	1404

You must keep a photocopy or electronic copy of your assignment.

Student declaration

I certify that the attached assignment is my own work. Material drawn from other sources has been appropriately and fully acknowledged as to author/creator, source and other bibliographic details. Such referencing may need to meet unit-specific requirements as to format and style.

I give permission for my assignment to be copied, submitted and retained for the electronic checking of plagiarism.

Signature of student: **Date: 30th October 2025**

(Students submitting work electronically can type their name in the space for signature above, but must produce a signed copy of this coversheet on request.)

Date of submission: 30th October 2025

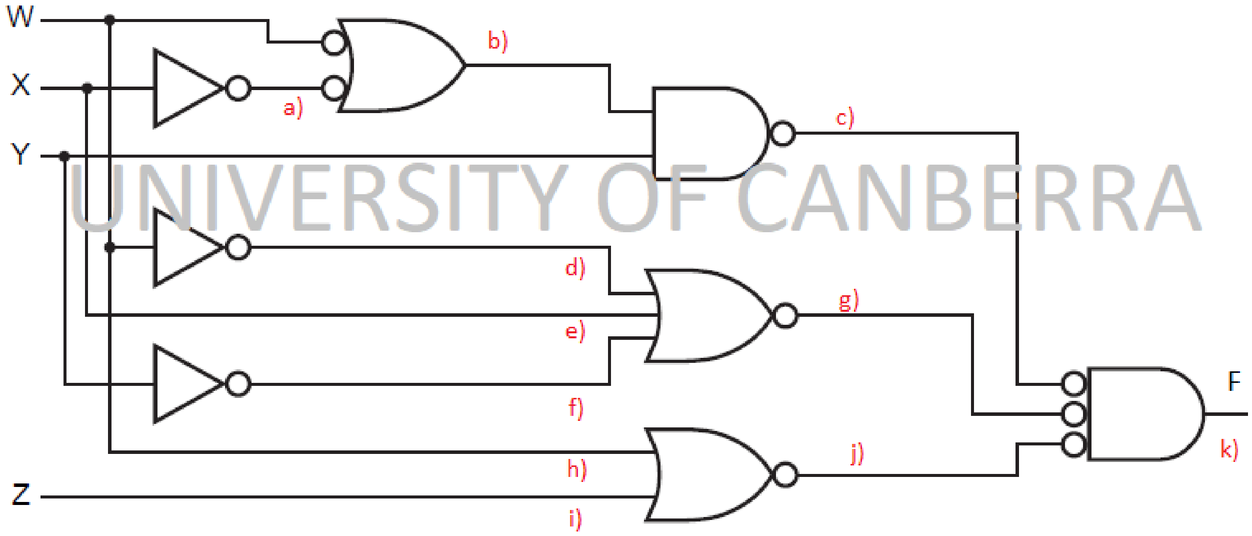


Figure 1: Boolean Circuit for Laboratory Week 12

1. Boolean Expression Derivation from Circuit (20 marks)

Derive the Boolean expression that describes the circuit shown in Figure 1. Show your steps in each branch of the circuit given.

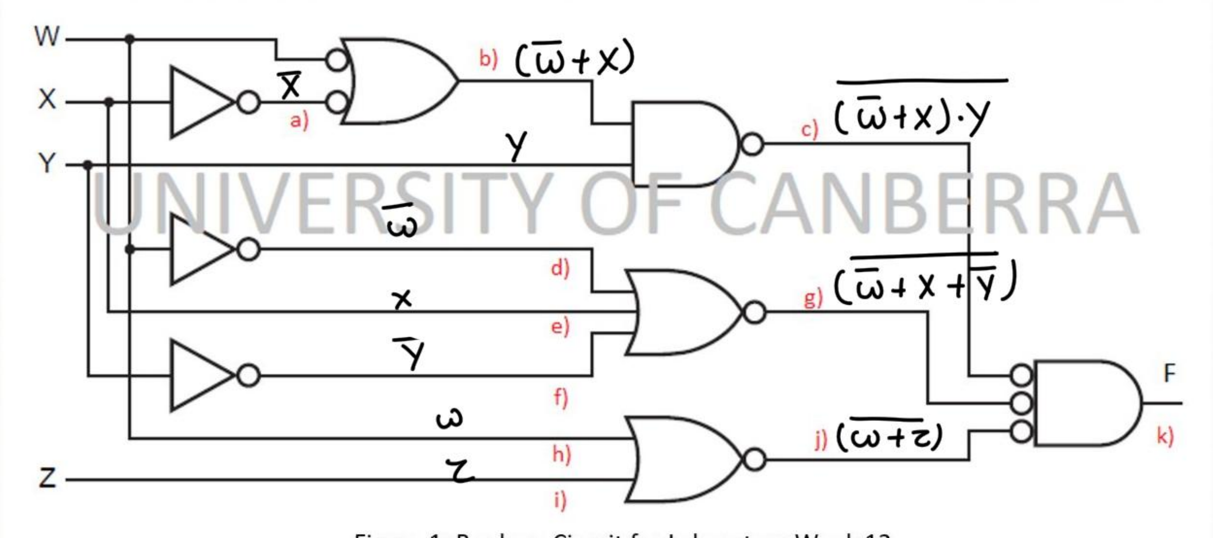

W
X
Y
Z
$\bar{X}$
a)
b) $(\bar{\omega}+X)$
y
c) $\overline{(\bar{\omega}+x)\cdot y}$
$\bar{\omega}$
x
$\bar{y}$
d)
e)
f)
g) $\overline{(\bar{\omega}+X+\bar{y})}$
ω
z
h)
i)
j) $\overline{(\omega+z)}$
k)
F

Figure 1: Boolean Circuit for Laboratory Week 12

$$F = \overline{\overline{\overline{(\bar{\omega}+x)\cdot y}} \cdot \overline{\overline{(\bar{\omega}+x+\bar{y})}} \cdot \overline{\overline{(\omega+z)}}}$$

$$F = \overline{\overline{\overline{(\bar{\omega}+x)\cdot y}} \cdot \overline{\overline{(\bar{\omega}+x+\bar{y})}} \cdot \overline{\overline{(\omega+z)}}} \cdot$$

Here, we have for each intermediate expression we get,

$$a = \overline{X}$$

$$b = \overline{W} + X = \overline{W.\overline{X}}$$

$$c = \overline{Y.\overline{W\overline{X}}}$$

$$d = \overline{W}$$

$$e = X$$

$$f = \overline{Y}$$

$$g = \overline{\overline{W} + X + \overline{Y}}$$

$$h = W$$

$$i = Z$$

$$j = (\overline{W + Z})$$

$$k = \overline{\overline{Y.\overline{W\overline{X}}} + \overline{\overline{W} + X + \overline{Y}} + \overline{W + Z}}$$

2. Boolean Expression Simplification Using Theorems (10 marks)

On simplification, the expression is given as:

$$F = \overline{\overline{Y.\overline{W\overline{X}}} + \overline{\overline{W} + X + \overline{Y}} + \overline{W + Z}}$$

$$F = \overline{\overline{Y.\overline{W\overline{X}}}}\,.\,\overline{\overline{\overline{W} + X + \overline{Y}}}\,.\,\overline{\overline{W + z}}\ \ [Applying\ De - morgan's\ theorem]$$

$$F = Y.\overline{W\overline{X}}\,.(\overline{W} + X + \overline{Y}).\ (W + Z)\ \ [Applying\ tautology]$$

$$F = Y(\overline{W} + \overline{\overline{X}}).(\overline{W} + X + \overline{Y}).\ (W + Z)\ \ [Applying\ De - Morgan's\ law]$$

$$F = Y(\overline{W} + X).(\overline{W} + X + \overline{Y}).\ (W + Z)\ \ [Applying\ tautology]$$

$$F = Y(\overline{W} + X + \overline{Y}).\ (W + Z)\,\overline{W} + Y.(\overline{W} + X + \overline{Y}).\ (W + Z)\,X\ \ [Applying\ distribution\ law]$$

$$F = Y\,(W+Z)\,\overline{W}\,.\,\overline{W} + Y\,(W+Z)\,X.\overline{W} + Y(W+Z)\overline{W}\,.\overline{Y} + Y\left(\overline{W}+X+\overline{Y}\right).\ (W+Z)\,X \quad [Applying\ idempodent\ law]$$

$$F = Y\,(W+Z)\,\overline{W} + Y\,(W+Z)\,X.\overline{W} + 0 + Y\left(\overline{W}+X+\overline{Y}\right).\ (W+Z)\,X \quad [Applying\ complement\ law]$$

$$F = Y\,(W+Z)\,\overline{W} + Y\,(W+Z)\,X.\overline{W} + Y\left(\overline{W}+X+\overline{Y}\right).\ (W+Z)\,X \quad [Applying\ identity\ law]$$

$$F = Y\,(W+Z)\,\overline{W} + Y\left(\overline{W}+X+\overline{Y}\right).\ (W+Z)\,X \quad [Applying\ absorption\ law]$$

$$F = YW\overline{W} + Y\overline{W}Z + Y\left(\overline{W}+X+\overline{Y}\right).\ (W+Z)\,X \quad [Applying\ distribution\ law]$$

$$F = 0 + Y\overline{W}Z + Y\left(\overline{W}+X+\overline{Y}\right).\ (W+Z)\,X \quad [Applying\ complement\ law]$$

$$F = Y\overline{W}Z + Y\left(\overline{W}+X+\overline{Y}\right).\ (W+Z)\,X \quad [Applying\ identity\ law]$$

$$F = Y\overline{W}Z + Y\,(W+Z)\,X\overline{W} + Y\,(W+Z)\,XX + Y\,(W+Z)\,X.\overline{Y} \quad [Applying\ distribution\ law]$$

$$F = Y\overline{W}Z + Y\,(W+Z)\,X\overline{W} + Y\,(W+Z)\,X + Y\,(W+Z)\,X.\overline{Y} \quad [Applying\ idempodent\ law]$$

$$F = Y\overline{W}Z + Y\,(W+Z)\,X\overline{W} + Y\,(W+Z)\,X + 0 \quad [Applying\ complement\ law]$$

$$F = Y\overline{W}Z + Y\,(W+Z)\,X\overline{W} + Y\,(W+Z)\,X \quad [Applying\ identity\ law]$$

$$F = Y\overline{W}Z + Y\,(W+Z)\,X \quad [Applying\ absorption\ law]$$

$$F = \overline{W}YZ + WXY + XYZ \quad [Applying\ distribution\ law]$$

$$F = \overline{W}YZ + WXY + XYZ$$

3. Boolean Expression Simplification Using Karnaugh Maps (10 marks)

The truth table for the above circuit is given as:

W	X	Y	Z	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0

1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

Now, using Karnaugh map for the simplification of the expression, we get:

	$\overline{Y}\,\overline{Z}$	$\overline{Y}\,Z$	YZ	$Y\,\overline{Z}$
$\overline{W}\,\overline{X}$	0	0	1	0
$\overline{W}\,X$	0	0	1	0
WX	0	0	1	1
$W\,\overline{X}$	0	0	0	0

On grouping the rows for 1's we get:

$$F = \overline{W}YZ\ + WXY$$

The results from the simplification and Karnaugh map are different due to the redundancy in the terms in the manually simplified expression. The results are, however, equivalent.

4. VHDL Code representation SA (20 marks)

From simplification obtained above, and using the structural approach, the code representation using structural approach is given as:

considering , W -> IN1, X -> IN2, Y-> IN3, Z-> IN4 and F -> OUT

```
entity Logic is
        port(IN1, IN2, IN3, IN4 : in bit; OUT5: out bit);
architecture LogicCircuit of Circuit is
        component NOT_gate is
                port(A:in bit;X:out bit);
        end component NOT_GATE;

        component AND_gate3 is
                port(A, B,C: in bit; X:out bit);
        end component AND_GATE;

        component OR_gate3 is
                port (A,B, C : in bit; X:out bit);
        end component OR_GATE;
```

```
signal OUT1, OUT2, OUT3, OUT4: bit;
begin
        G1: NOT_gate port map( A=>IN1, X=>OUT1);
        G2: And_gate3 port map (A=>IN1, B=>IN2, C=>IN3, X=>OUT2)
        G3: And_gate3 port map (A=>IN2, B=>IN3, C=>IN4, X=>OUT3)
        G4: And_gate3 port map (A=>IN2, B=>OUT1, C=>IN4, X=>OUT4)
        G5: OR_gate3 port map (A=> OUT2 , B=>OUT3, C=>OUT4, X=>OUT5)
end architecture LogicCircuit;
```

5. VHDL Code representation DF (20 marks)

From simplification obtained above, using the Data Flow approach, we get the code as:

```
entity Logic is
        port(
                W: in STD_LOGIC;
                X: in STD_LOGIC;
                Y : in  STD_LOGIC;
                Z : in STD_LOGIC;
                F : out STD_LOGIC;
);
End Logic;
Architecture Behavioral of  Logic is
begin
        F<= (W and X and Y) or (X and Y and Z) or (not W and Y and Z)
end Behavioral;
```

6. VHDL Implementation (20 marks)

From simplification obtained above, and using the data flow approach,

6.1 Correct structure of the code written in Vivado. It includes proper I/O mapping.

The code written in Vivado is as follows:

```
library IEEE;
use IEE.STD_LOGIC_1164.ALL;

entity tute_12 is
        port(
                W: in STD_LOGIC;
                X: in STD_LOGIC;
                Y : in  STD_LOGIC;
                Z : in STD_LOGIC;
                F : out STD_LOGIC;
);
End tute_12;
Architecture Behavioral of  tute_12 is
begin
        F<= (W and X and Y) or (X and Y and Z) or (not W and Y and Z)
end Behavioral;
```

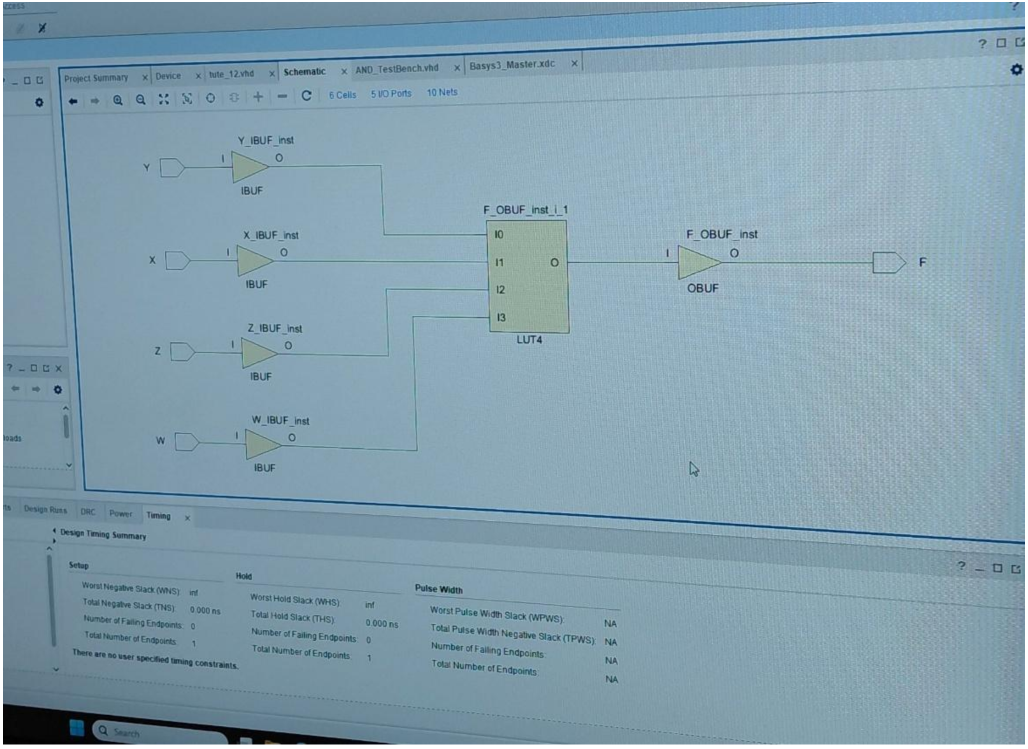
Similarly, the testbench for the above is written as:

```
Library IEEE;
Use IEEE.STD_LOGIC_1164.ALL;

Entity tb_and_gate is
End tb_and_gate;

architecture behavior of tb_and_gate is
        Signal A, B, Y; STD_LOGIC;
Begin
        uut.entity work.tute_12
                port map(
                        W => W,
                        X => X,
                        Y=> Y,
                        Z=>Z,
                        F=> F);

        -- Test process
process
begin
        W <= ‘0’ ; X <= ‘0’ ; Y <= ‘0’ ; Z<=’0’ wait for 10 ns;
        W <= ‘0’ ; X <= ‘0’ ; Y <= ‘0’ ; Z<=’1’ wait for 10 ns;
        W <= ‘0’ ; X <= ‘0’ ; Y <= ‘1’ ; Z<=’0’ wait for 10 ns;
        W <= ‘0’ ; X <= ‘0’ ; Y <= ‘1’ ; Z<=’1’ wait for 10 ns;
        W <= ‘0’ ; X <= ‘1’ ; Y <= ‘0’ ; Z<=’0’ wait for 10 ns;
        W <= ‘0’ ; X <= ‘1’ ; Y <= ‘0’ ; Z<=’1’ wait for 10 ns;
        W <= ‘0’ ; X <= ‘1’ ; Y <= ‘1’ ; Z<=’0’ wait for 10 ns;
        W <= ‘0’ ; X <= ‘1’ ; Y <= ‘1’ ; Z<=’1’ wait for 10 ns;
        W <= ‘1’ ; X <= ‘0’ ; Y <= ‘0’ ; Z<=’0’ wait for 10 ns;
        W <= ‘1’ ; X <= ‘0’ ; Y <= ‘0’ ; Z<=’1’ wait for 10 ns;
        W <= ‘1’ ; X <= ‘0’ ; Y <= ‘1’ ; Z<=’0’ wait for 10 ns;
        W <= ‘1’ ; X <= ‘0’ ; Y <= ‘1’ ; Z<=’1’ wait for 10 ns;
        W <= ‘1’ ; X <= ‘1’ ; Y <= ‘0’ ; Z<=’0’ wait for 10 ns;
        W <= ‘1’ ; X <= ‘1’ ; Y <= ‘0’ ; Z<=’1’ wait for 10 ns;
        W <= ‘1’ ; X <= ‘1’ ; Y <= ‘1’ ; Z<=’0’ wait for 10 ns;
        W <= ‘1’ ; X <= ‘1’ ; Y <= ‘1’ ; Z<=’1’ wait for 10 ns;
        Wait;
```

```
end process;
end behavior;
```

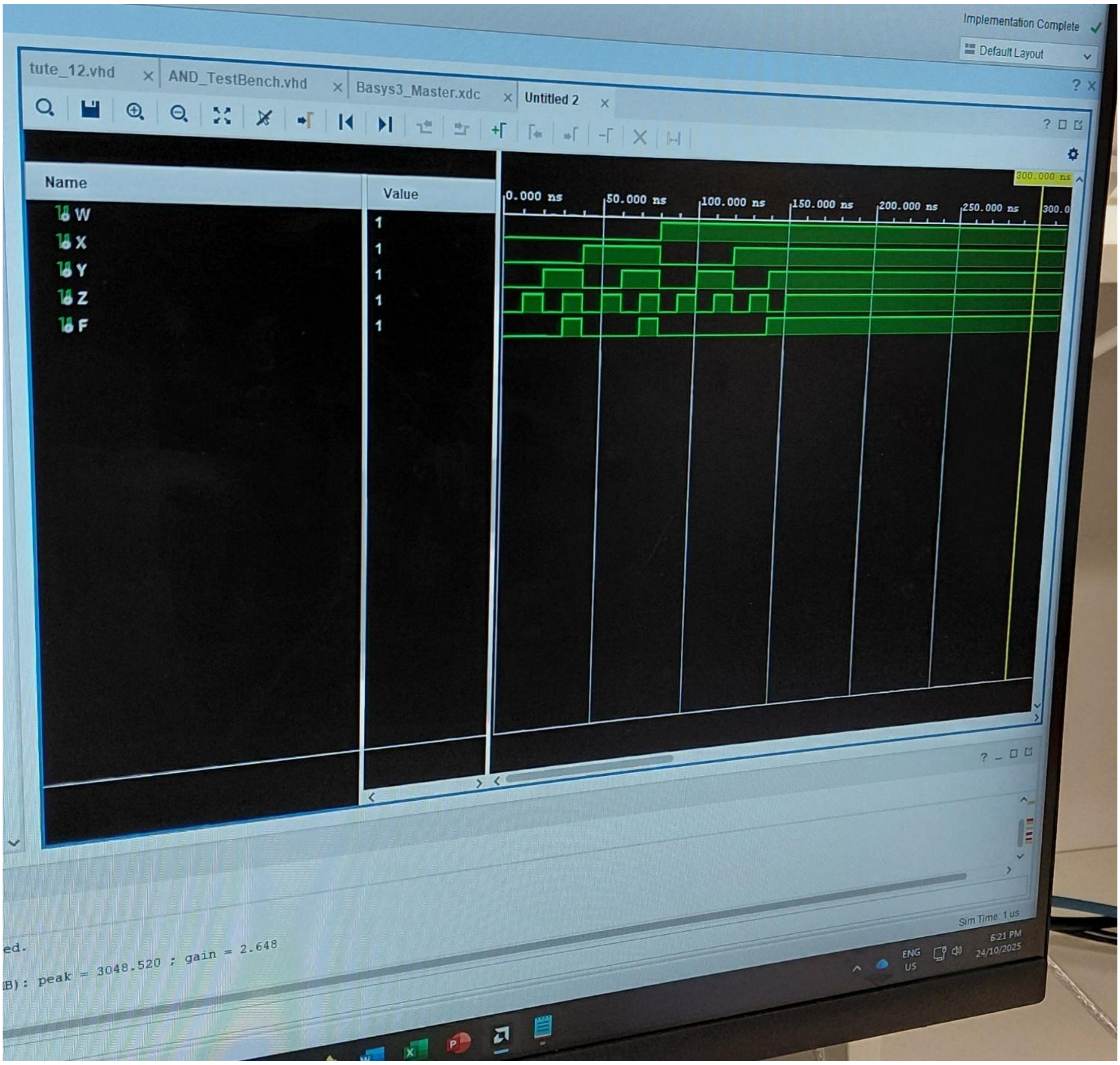
6.2 Correct simulation of the circuit in Vivado (time diagram).

The circuit was simulated, and the timing diagram was obtained as:



Design Timing Summary

```
Setup                                    Hold                                   Pulse Width
Worst Negative Slack (WNS):   inf        Worst Hold Slack (WHS):      inf       Worst Pulse Width Slack (WPWS):             NA
Total Negative Slack (TNS):   0.000 ns   Total Hold Slack (THS):      0.000 ns  Total Pulse Width Negative Slack (TPWS):    NA
Number of Failing Endpoints:  0          Number of Failing Endpoints: 0         Number of Failing Endpoints:                NA
Total Number of Endpoints:    1          Total Number of Endpoints:   1         Total Number of Endpoints:                  NA
There are no user specified timing constraints.
```



Note: The simulation video (timing diagram) can be accessed through the file timing_diagram_video/ timing_diagram.mp4

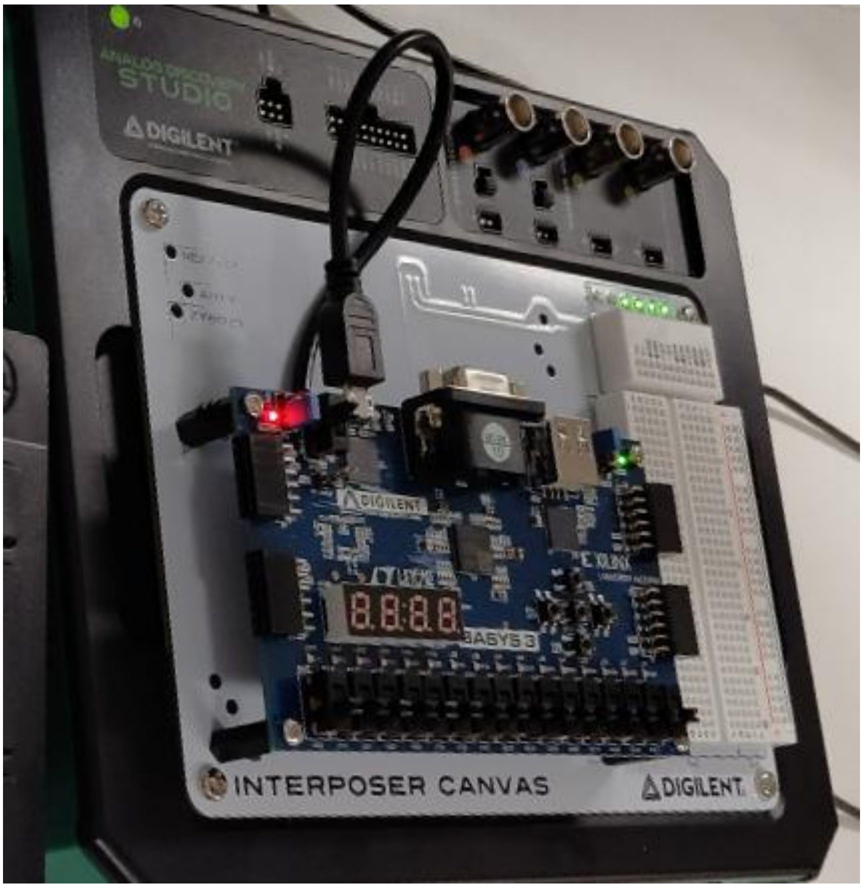
6.3 Correct functioning of the FPGA by verifying that your circuit works as intended using a truth table.

The truth table for the circuit is as follows:

W	X	Y	Z	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0

0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

The FPGA implementation of the circuit is given as:



**Note: The FPGA implementation can be accessed through the file : FPGA_video/ FPGA_implementation.mp4*

REFERENCES

[1] Floyd, Th.L. Digital Fundamentals, Pearson, 2015

[2] Boolean algebra website: https://www.boolean-algebra.com