University of Guelph

ENGG*2410: Digital Design

Combinational Logic Design “Majority Circuit” via Schematic Capture

Group (# 8)

Layal Alzaydi (1041786)

Josiah Varghese (1043468)

Giuliana Lowry (1013417)

Mazen hassan (0984322)

Preformed On: September 24th, 2019

Due Date: October 1st, 2019

Problem Statement

This Lab was composed of two main parts. For each part there was a problem statement for which a Truth Table was to be created. Each circuit was to be drawn and simulated for verification.

For Part 1 of the lab, the FPGA LED light would light when the Majority of the inputs are ON. The circuit has three inputs (A, B,C) and one output (F). The function (output) is TRUE when the Majority of the inputs (A,B,C) are 1's and FALSE when the Majority of the inputs (A,B,C) are 0's.

For Part 2 , the FPGA LED light would light when the inputs are of different binary values (i.e → 0,1) The Circuit has two inputs (X and Y) and one output (F). The Function F (output) is TRUE when X and Y are different, and FALSE when they are the same.

Assumptions and Constraints

Some of the assumptions and constraints used in this lab are as follows:

- Assumed that Xilinx would perform any optimizations needed
- No hand optimizations were done
- Only AND/OR gates were to be used

System Overview & Justification of Design

The purpose of this lab is to develop a truth table for a given problem, derive its Boolean function, and design its respective circuit. The results were then to be verified using a simulation on Xilinx. Xilinx ISE Foundation design software was used to design the circuit then it was implemented using the Digilent NEXYS 3 FPGA Board.

This lab consisted of two parts, the group decided to split into two teams and each subteam took one part to work on. Both parts had the same steps to reach to the desired outputs, however each part required its own truth table, logic function, and circuit design. Both teams started by understanding and analyzing the given problem then a truth table was produced for each problem. A Boolean function was derived from each truth table. Using the Boolean functions, two circuits were drawn using AND/OR gates. The design was then inputted using Schematic capture and simulated to verify its functionality. The first circuit was designed using four AND gates connected to an OR gate and the second circuit was designed using two AND gates connected to an OR gate. Inverters were used as needed to get the complement of the

inputs. Furthermore, for the simulation, the code from the tutorial "Bench test used for behavioral simulation" was copied and edited to match the truth table for each problem. The tests worked as expected which proved that the circuits were drawn correctly. Both parts 1 and 2 follow the same steps however different conditions were used to match the problem statement for each part.

Part 1

Truth Table

A (input)	B (input)	C (input)	F (output)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Figure 1: Truth Table for part 1 majority circuit

Logic Function

From the Truth table (Figure 1) above the following information was derived:

- From the Fourth row → where the output is 1 the inputs are A'BC
- From the Sixth row → where the output is 1 the inputs are AB'C
- From the Seventh row → where the output is 1 the inputs are ABC'
- From the Eighth row → where the output is 1 the inputs are ABC

The Logic function for this Truth Table is as follows: **F = A'BC + AB'C + ABC' + ABC**

Circuit Diagram

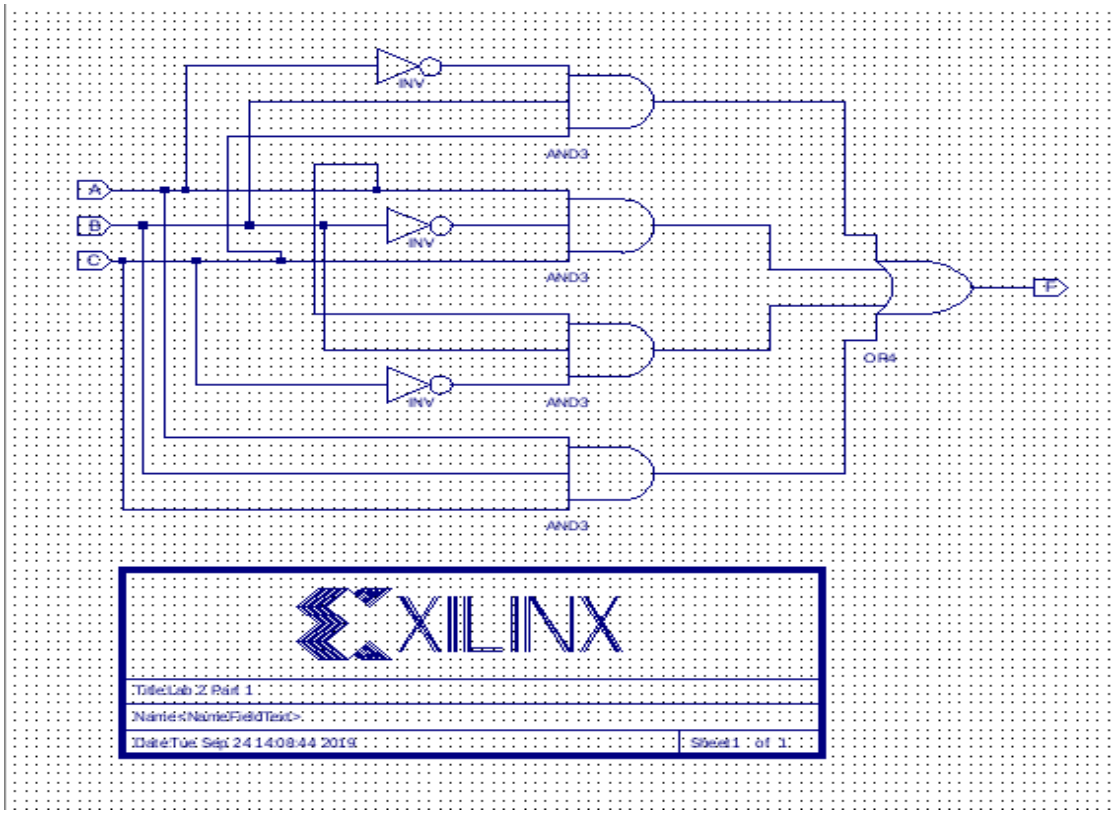

Figure 2: Circuit Diagram for part 1 majority circuit

The diagram above (Figure 2) shows the circuit for part 1. The inputs in this design are A, B, and C and the output is F. It consists of four AND gates connected to an OR gate as well as inverters on the top three AND gates to get A', B', and C' respectively. F is one if the majority of the inputs are one.

Symbol Diagram

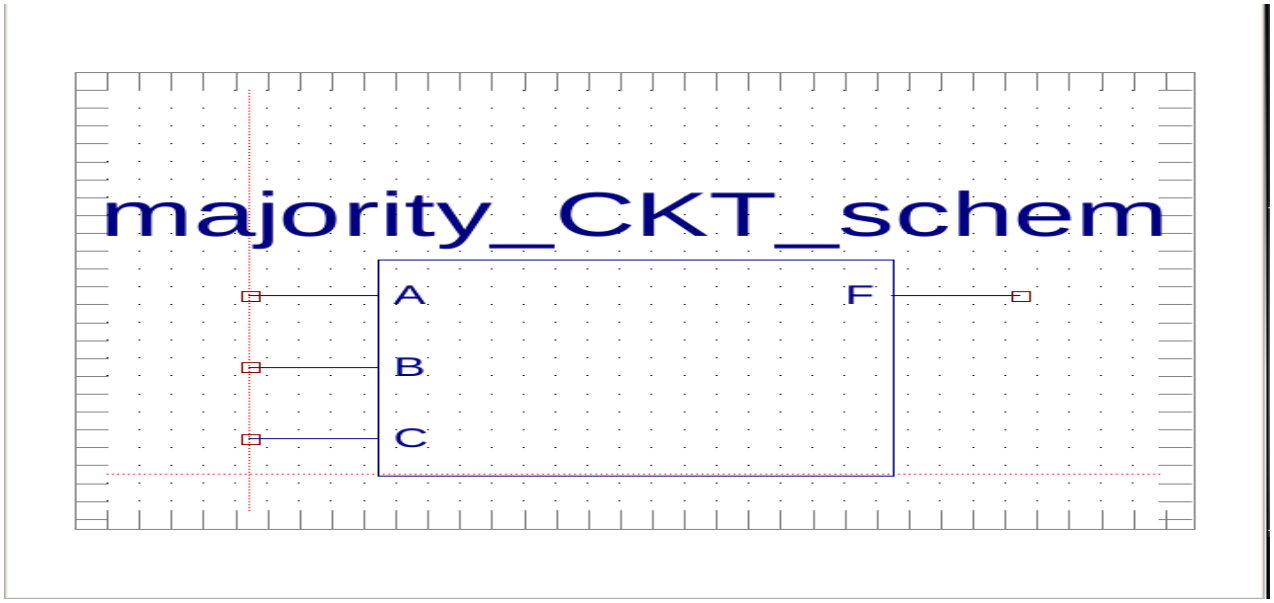

Figure 3: Symbol Diagram for part 1 majority circuit

Simulation

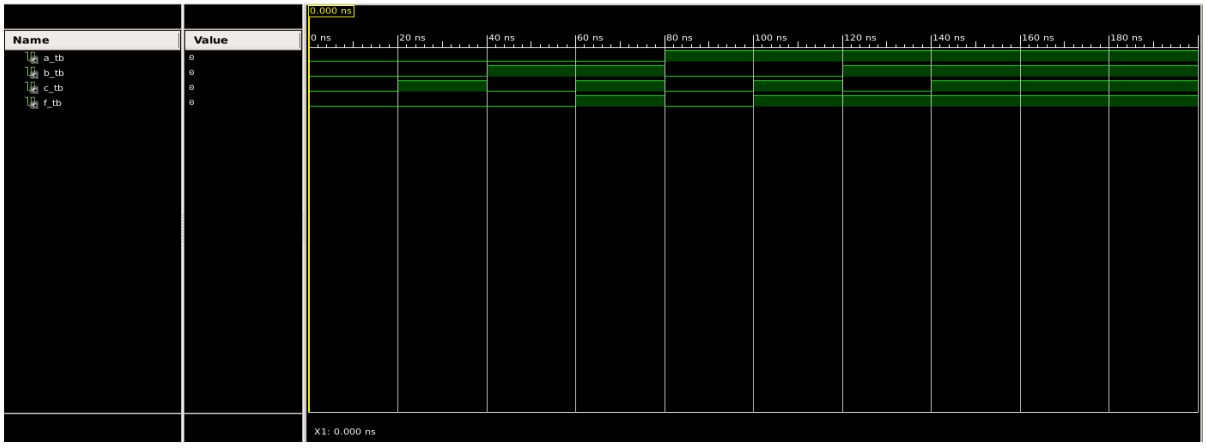

Figure 4: Simulation for part 1 majority circuit

The simulation above (Figure 4) is the simulation for the majority circuit. A, B, and C are listed as inputs and F is listed as the output. When the majority of the inputs are one, the output is also one.

Test Bench

```
44
45  -- *** Test Bench - User Defined Section ***
46     tb : PROCESS
47     constant period: time := 20 ns;
48     BEGIN
49     A_tb <= '0'; -- apply input combination 00 and check outputs
50     B_tb <= '0';
51     C_tb <= '0';
52     wait for period;
53     assert (F_tb = '0')
54     report "test failed for input combination 000" severity error;
55     A_tb <= '0'; -- apply input combination 01 and check outputs
56     B_tb <= '0';
57     C_tb <= '1';
58     wait for period;
59     assert (F_tb = '0')
60     report "test failed for input combination 001" severity error;
61     A_tb <= '0'; -- apply input combination 10 and check outputs
62     B_tb <= '1';
63     C_tb <= '0';
64
65     wait for period;
66     assert (F_tb = '0')
67     report "test failed for input combination 010" severity error;
68     A_tb <= '0'; -- apply input combination 11 and check outputs
69     B_tb <= '1';
70     C_tb <= '1';
71     wait for period;
72     assert (F_tb = '1')
73     report "test failed for input combination 011" severity error;
```

```
74
75    A_tb <= '1'; -- apply input combination 11 and check outputs
76    B_tb <= '0';
77    C_tb <= '0';
78    wait for period;
79    assert (F_tb = '0')
80    report "test failed for input combination 100" severity error;
81
82    A_tb <= '1'; -- apply input combination 11 and check outputs
83    B_tb <= '0';
84    C_tb <= '1';
85    wait for period;
86    assert (F_tb = '1')
87    report "test failed for input combination 101" severity error;
88
89    A_tb <= '1'; -- apply input combination 11 and check outputs
90    B_tb <= '1';
91    C_tb <= '0';
92    wait for period;
93    assert (F_tb = '1')
94    report "test failed for input combination 110" severity error;
95
96    A_tb <= '1'; -- apply input combination 11 and check outputs
97    B_tb <= '1';
98    C_tb <= '1';
99    wait for period;
100   assert (F_tb = '1')
101   report "test failed for input combination 111" severity error;
102   WAIT; -- will wait forever
103   END PROCESS;
```

Figure 5: Bench Test for part 1 majority circuit

Part 2

Truth Table

X (input)	Y(input)	F(output)
0	0	0
0	1	1
1	0	1
1	1	0

Figure 6: Truth Table for part 2

Logic Function

From the Truth table (Figure 6) above the following information was derived:

- From the second row → where the output is 1 the inputs are X'Y
- From the third row → where the output is 1 the inputs are XY'

The Logic function for this Truth Table is as follows: **F = X'Y + XY'**

Circuit Diagram

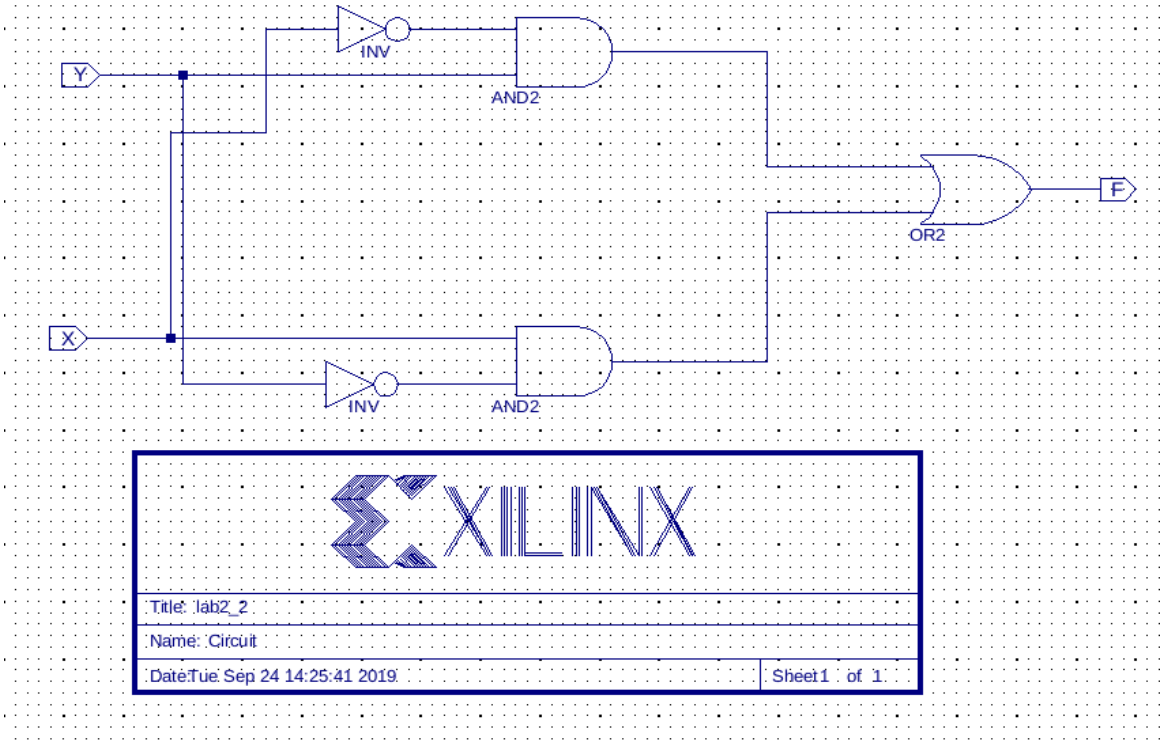

Figure 7: Circuit Diagram for part 2 majority circuit

The diagram above (Figure 7) shows the circuit diagram for the second part. The circuit Diagram consists of two AND gates with two inputs (X,Y) that are connected to an OR gate . The OR gate is then connected to output F. The system is programmed so that the output only lights up if the two inputs have different values.

Symbol Diagram

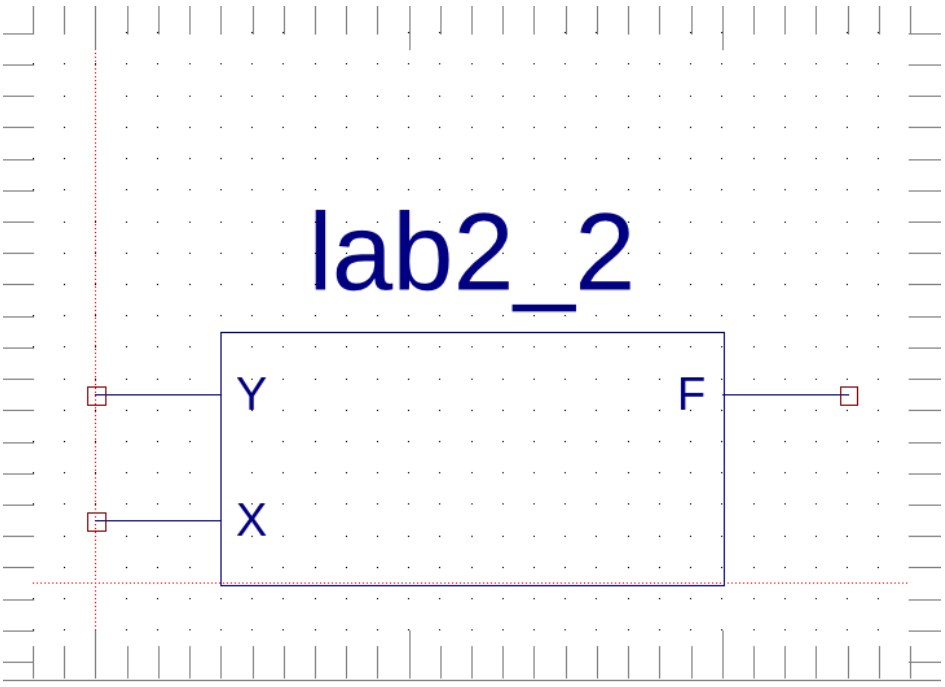

Figure 8: Symbol Diagram for part 2

Simulation

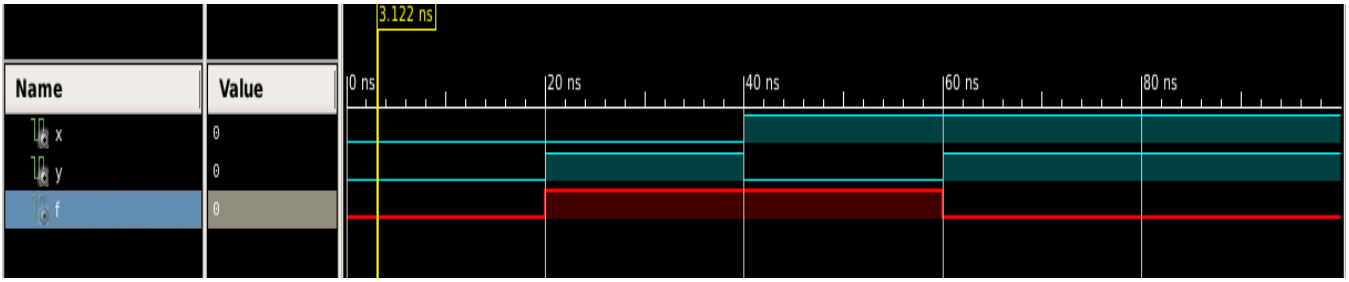

Figure 9: Simulation for part 2

The simulation above (Figure 9) shows that the output is 1/ON when the inputs are different (one is ON and the other is OFF).

*The team apologies if the picture is too small, we tried taking another one but we could not run our simulation again.

Test Bench

```
40      );
41
42    -- *** Test Bench - User Defined Section ***
43       tb : PROCESS
44       constant period: time := 20 ns;
45       BEGIN
46       X_tb <= '0'; -- apply input combination 00 and check outputs
47       Y_tb <= '0';
48    wait for period;
49    assert ((F_tb = '0'))
50    report "test failed for input combination 00" severity error;
51
52    X_tb <= '0'; -- apply input combination 00 and check outputs
53       Y_tb <= '1';
54    wait for period;
55    assert ((F_tb = '1'))
56    report "test failed for input combination 00" severity error;
57
58    X_tb <= '1'; -- apply input combination 00 and check outputs
59       Y_tb <= '0';
60    wait for period;
61    assert ((F_tb = '1'))
62    report "test failed for input combination 00" severity error;
63
64    X_tb <= '1'; -- apply input combination 00 and check outputs
65       Y_tb <= '1';
66    wait for period;
67    assert ((F_tb = '0'))
68    report "test failed for input combination 00" severity error;
69          WAIT; -- will wait forever
70       END PROCESS;
71    -- *** End Test Bench - User Defined Section ***
72
73    END;
74
```

Figure 10: Test Bench for part 2

Error Analysis

The final circuits worked as expected and all outputs matched what was described in the problem statement. There were a few errors that occurred during the lab which led to improper circuit designs and simulation outputs which were then corrected. At first, the circuit design did not include inverters and instead used the compliments of the inputs directly. When drawing the circuit in Xilinx, some of the inputs were connected to the wrong AND gates which made the outputs incorrect. The design was corrected prior to drawing the circuit in Xilinx. The circuit diagram was corrected and the simulation was run again to ensure proper outputs. There were two errors made while writing the test bench which included misnaming variables and declaring too long of a wait time which made some of the intervals too long. The test bench was edited to correct these issues and the simulation was run again with the corrected test bench.

Conclusion

To conclude, the group believes that the purpose of the lab was met. For each part, a truth table, Boolean function, and circuit diagrams were successfully obtained. Both circuits were tested and no error was given, which further proves that the circuits made are correct and satisfy the main objective of the lab. Overall the group didn't face any challenges in conducting the lab however some minor errors were present as mentioned in the error analysis section. The important thing is that the group got adjusted to the situations quickly and the results were precise and understandable.