Lab Report #3

Digital Logic Design

Submitted By:

Haider Ali (FA21-BEE-053)

Malik Ammar Murtaza (FA21-BEE-078)

Muhammad Ahmad (FA21-BEE-095)

Submitted to:

Dr. Adnan Qureshi

LAB#03 Introduction to Verilog and Simulation using XILINX ISE

Apparatus:

- **XILINX**
- **FPGA**

Concept:

Verilog is a hardware description language used to write code to describe circuits and gates. Its scope is from most simple circuits like AND gates to very complex output circuits with thousands of inputs.

Xilinx is an IDE which gives us programming environment to code in Verilog and test our code.

It provides drivers and connections to put our circuit on FPGA and test it too.

Part 1: Basic Gates Output Using Verilog:

Procedure:

- Create new project.
- Create a module and specify inputs and outputs.
- Implement all basic gates and give their output.

Verilog:

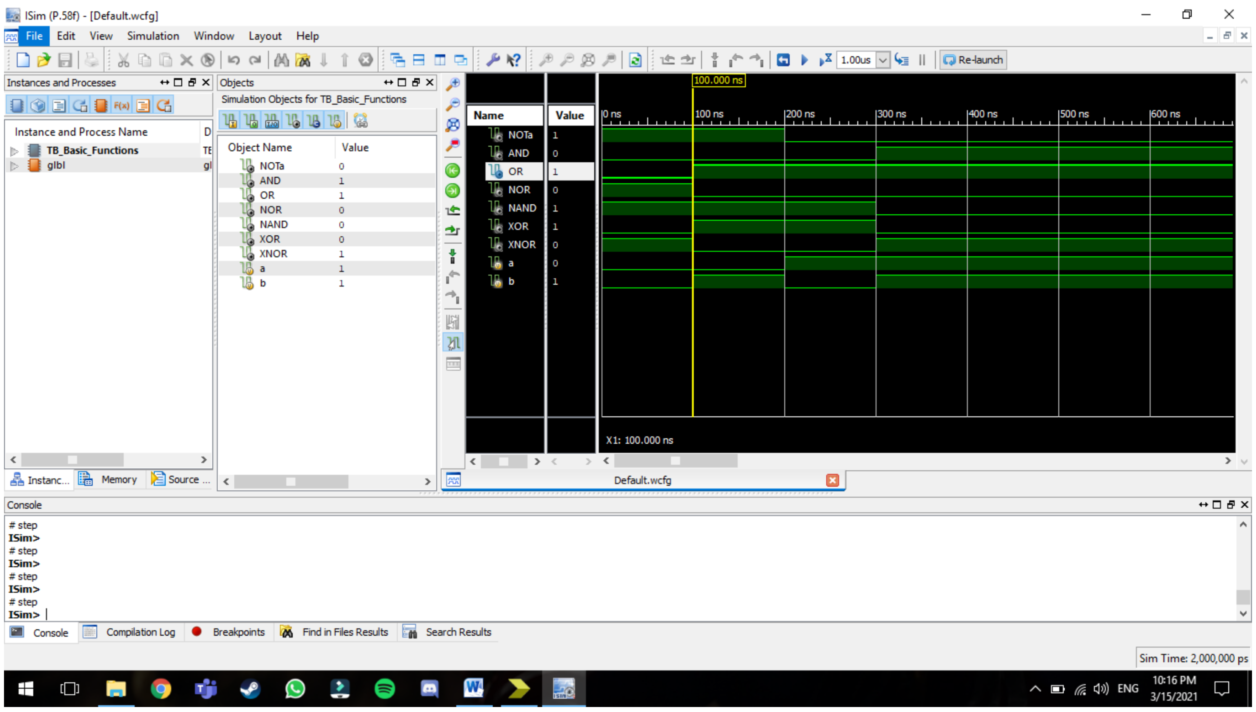
```
module Basic_Functions(a,b,NOTa,AND,OR,NOR,NAND,XOR,XNOR

   );

              input a,b;

              output NOTa,AND,OR,NOR,NAND,XOR,XNOR;

              assign NOTa = ~a;

              assign AND = a && b;

              assign OR = a || b;

              assign NOR = ~(a || b);

              assign NAND = ~(a && b);

              assign XOR = a^b;

              assign XNOR = ~(a^b);


endmodule
```

Test Bench:

```
module FuncTest;

	// Inputs
	reg a;
	reg b;

	// Outputs
	wire NOTa;
	wire AND;
	wire OR;
	wire NOR;
	wire NAND;
	wire XOR;
	wire XNOR;

	// Instantiate the Unit Under Test (UUT)
	BasicFuncs uut (
		.a(a),
		.b(b),
		.NOTa(NOTa),
		.AND(AND),
		.OR(OR),
		.NOR(NOR),
		.NAND(NAND),
		.XOR(XOR),
		.XNOR(XNOR)
	);

	initial begin
		a = 0;
		b = 0;
		#100;
		a = 0;
		b = 1;
		#100;
```

```
                a = 1;

                b = 0;

                #100;

                a = 1;

                b = 1;

                #100;



        end

endmodule
```

Behavior Test:



ISim (P.58f) - [Default.wcfg]

Simulation Objects for TB_Basic_Functions

Object Name	Value
NOTa	0
AND	1
OR	1
NOR	0
NAND	0
XOR	0
XNOR	1
a	1
b	1

Name	Value
NOTa	1
AND	0
OR	1
NOR	0
NAND	1
XOR	1
XNOR	0
a	0
b	1

X1: 100.000 ns

```
# step
ISim>
# step
ISim>
# step
ISim>
# step
ISim>
```

Sim Time: 2,000,000 ps

PART 2: Implement Boolean Expressions Using XILINX:

Procedure:

- Create project in Xilinx.
- Specify inputs and outputs in main module.
- Implement the Boolean expression using gates.
- Give final output as output function variable
- Create test file.
- Test wave form.

MAIN METHOD CODE(DATA FLOW):

```
module FuncDataFlow(x,y,z,F
   );
             input x,y,z;
             output F;
              assign F = x | (y&(~x)) | (y&(~z))
endmodule
```

Main Method Code(GATE LEVEL):

```
module FuncGateLevel(x,y,z,F
   );
              input x,y,z;
              output F;
              wire not_x,not_z,f1,f2,f3;
              not n1(not_x,x);
              not n2(not_z,z);
              and a1(f1,not_x,y);
              and a2(f2,y,not_z);
              or o1(f3,f1,f2);
              or o2(F,f3,x);




endmodule
```

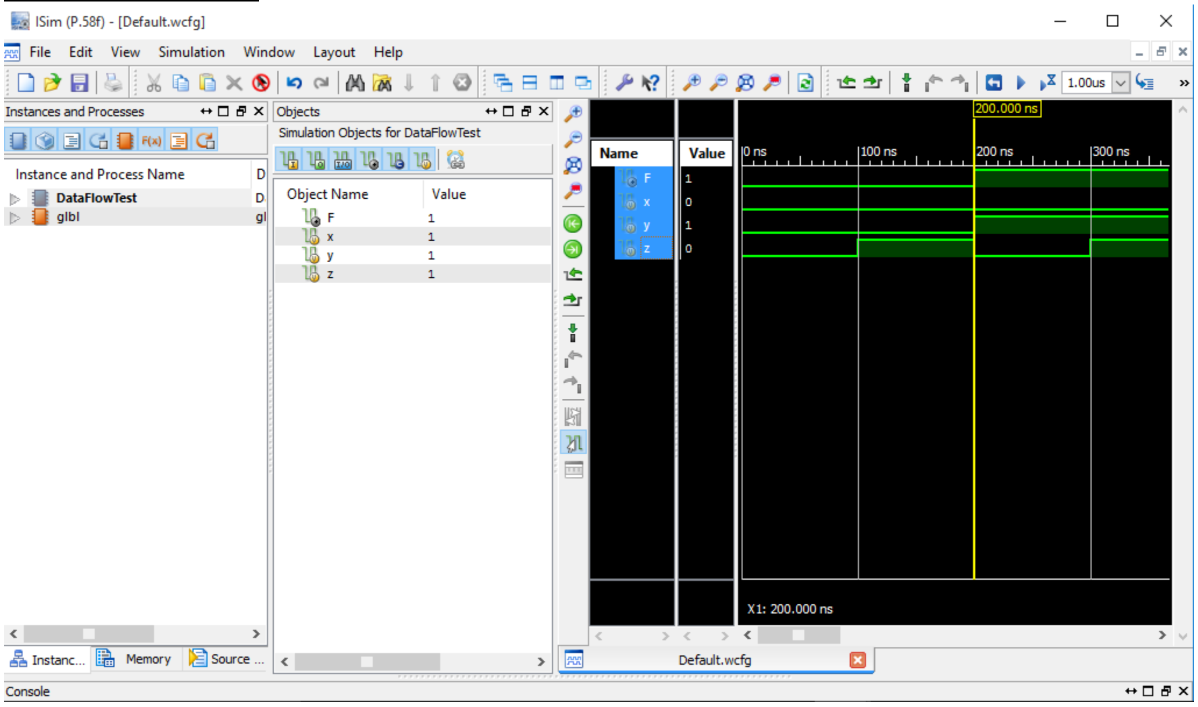
Test Bench:

```
module DataFlowTest;
```

```
// Inputs
reg x;
reg y;
reg z;

// Outputs
wire F;

// Instantiate the Unit Under Test (UUT)
FuncDataFlow uut (
        .x(x),
        .y(y),
        .z(z),
        .F(F)
);

initial begin
        x = 0;
        y = 0;
        z = 0;
        #100;
        x = 0;
        y = 0;
        z = 1;
        #100;
        x = 0;
        y = 1;
        z = 0;
        #100;
        x = 0;
        y = 1;
        z = 1;
        #100;
        x = 1;
        y = 0;
        z = 0;
```

```
                #100;
                x = 1;
                y = 0;
                z = 1;
                #100;
                x = 1;
                y = 1;
                z = 0;
                #100;
                x = 1;
                y = 1;
                z = 1;
                #100;


        end

endmodule
```

Behavior test:


ISim (P.58f) - [Default.wcfg]
File Edit View Simulation Window Layout Help
1.00us
Instances and Processes
Instance and Process Name
DataFlowTest
glbl
Objects
Simulation Objects for DataFlowTest
200.000 ns
0 ns
100 ns
200 ns
300 ns
X1: 200.000 ns
Instanc... Memory Source ...
Default.wcfg
Console

Object Name	Value
F	1
x	1
y	1
z	1

Name	Value
F	1
x	0
y	1
z	0

Critical Analysis:

In this lab 03 we started using new software named XILINX we learned some new commands in XILINX firstly we started our project by the procedure written in lab manual and then we wrote the Verilog code and synthesized it and then tested its behavior and the saw the results on test bench which are also attached in this lab report. There were two ways of doing test bench one was the GATE level method and the other one was DATA FLOW method we just need to put correct signs of particular gate and then test the behavior of that Verilog code.