

Experiment 2

Multi-Function Gate

Digital Systems

EEE3342C-20FALL 0011

Equipment:

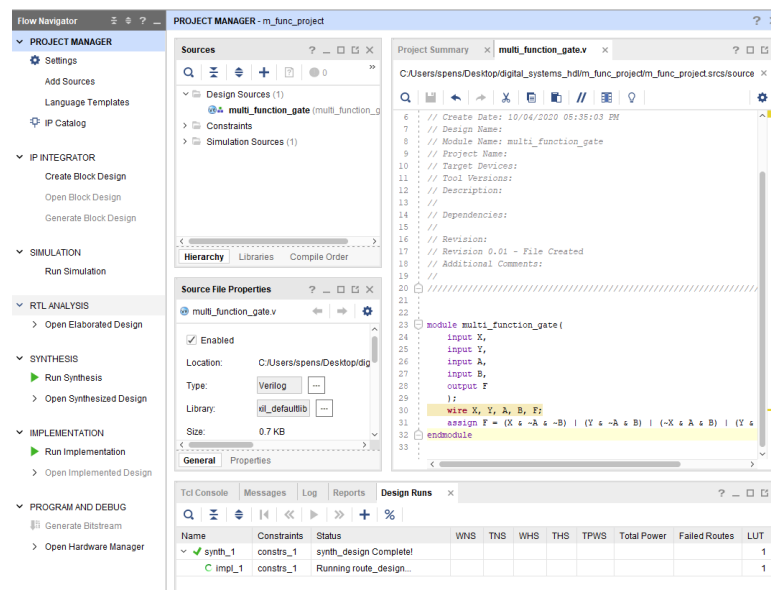
The Xilinx's FPGA VIVADO HLx Editions design tools are available in the laboratory. These tools can also be downloaded from Xilinx's web site at www.xilinx.com. The WebPack version of this tool that we use for the laboratory experiments are located under the support download section at the related website (<https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/vivado-design-tools/2017-4.html>). Please take note that the file download size exceeds 1 GB and also during the installation process updates may have to be installed. It can take you up to a few hours to download and install the software on your computer. The user does not need to have the BASYS development board interface to the computer to design and simulate an FPGA.

Objective:

The objective is to design and implement a multi-function gate with the Xilinx's FPGA Vivado tools and to display the results of the experiment. The Vivado tools will be used to design, simulate, and create (virtually) a multi-function gate.

Design Steps:

I created the project (m_func_proj) and added all inputs and the single output. I then programmed the logic of the multi-functional gate.

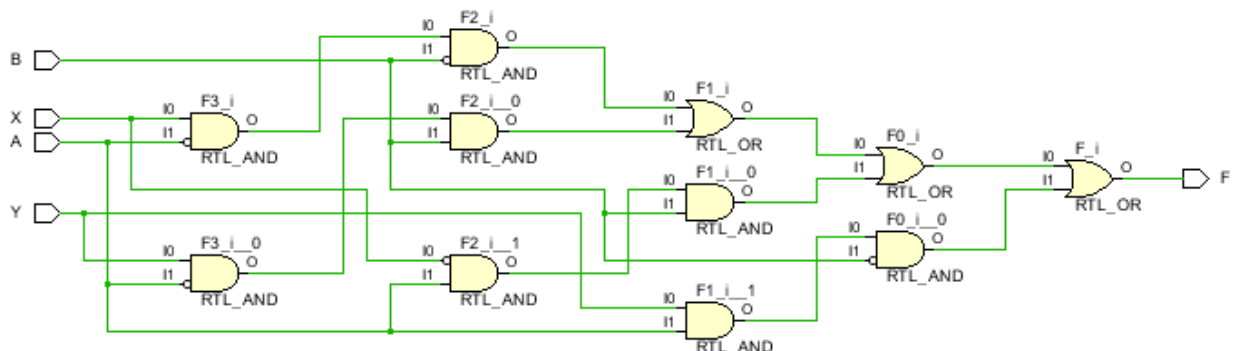


I then ran synthesis and implementation, to implement I/O Planning and Schematics. I then connected all the inputs and outputs to the I/O Planning.

The screenshot displays the Xilinx Vivado IDE for the project 'IMPLEMENTED DESIGN - xc7a35tcbg236-1'. The left sidebar shows the 'Flow Navigator' with sections for RTL ANALYSIS, SYNTHESIS, and IMPLEMENTATION. The 'IMPLEMENTATION' section is active, showing options like 'Run Implementation' and 'Open Implemented Design'. The 'Sources' panel lists 'multi_function_gate' with 'Nets (10)' and 'Leaf Cells (6)'. The 'Package' pin table is visible, showing scalar ports A, B, F, X, and Y. The top-right window shows a top-level schematic of the multi-functional gate, which is a complex logic circuit with multiple inputs and outputs.

Name	Direction	Neg Diff Pair	Package Pin	Fixed	Bank	I/O Std	Vcco	Vref	Drive Stren
Scalar ports (5)									
A	IN		V14	✓	14	LVCMS33*	3.300		
B	IN		V15	✓	14	LVCMS33*	3.300		
F	OUT		U16	✓	14	LVCMS33*	3.300		12
X	IN		V16	✓	14	LVCMS33*	3.300		
Y	IN		V17	✓	14	LVCMS33*	3.300		

Then I generated the schematic of the multi-functional gate.

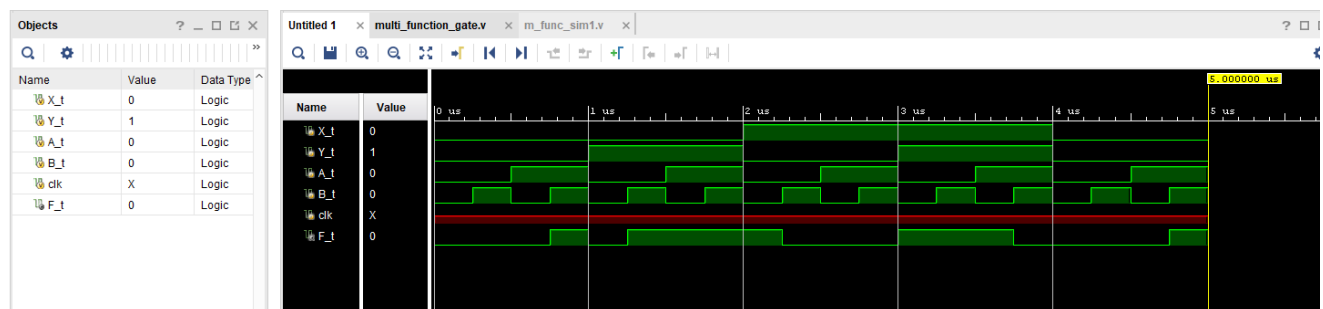


I then programmed the logic for the simulation.

```
C:/Users/spens/Desktop/digital_systems_hdl/m_func_project/

17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 //////////////////////////////////////////////////
21
22
23 module m_func_sim1(
24
25 );
26 reg X_t;
27 reg Y_t;
28 reg A_t;
29 reg B_t;
30 reg clk;
31 wire F_t;
32
33 multi_function_gate UUT(
34     .X(X_t),
35     .Y(Y_t),
36     .A(A_t),
37     .B(B_t),
38     .F(F_t)
39 );
40 initial begin
41     X_t = 1'b0;
42     Y_t = 1'b0;
43     A_t = 1'b0;
44     B_t = 1'b0;
45     //clk = 1'b0;
46 end
47 //always #5 clk = ~clk
48 always #2000 X_t = ~X_t;
49 always #1000 Y_t = ~Y_t;
50 always #500 A_t = ~A_t;
51 always #250 B_t = ~B_t;
52 endmodule
```

I then ran a behavioral simulation of the logic design.



Pre-Lab:

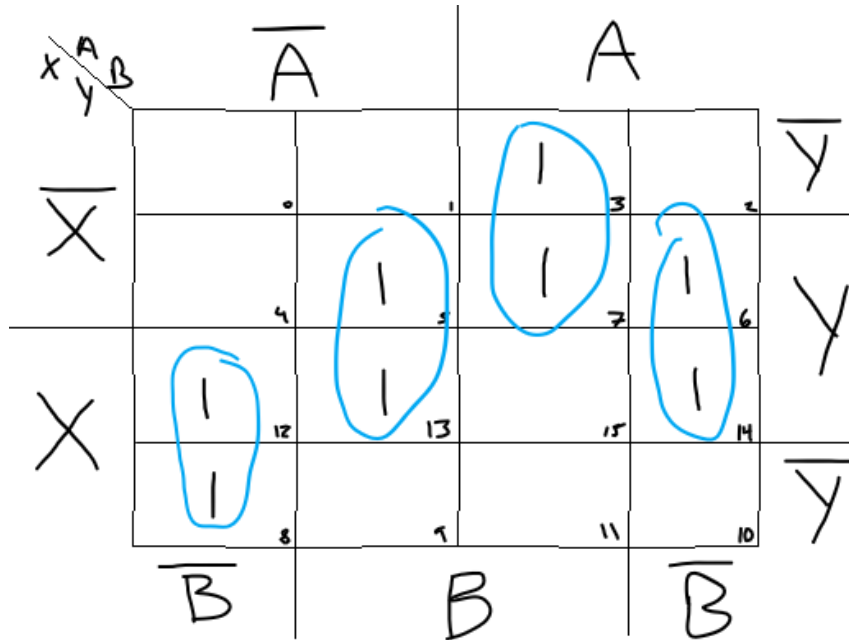
1. Read this experiment carefully to become familiar with the experiment.
2. Represent the output F as a function of X, Y, A and B on a truth table.

X	Y	A	B	OUT
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	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

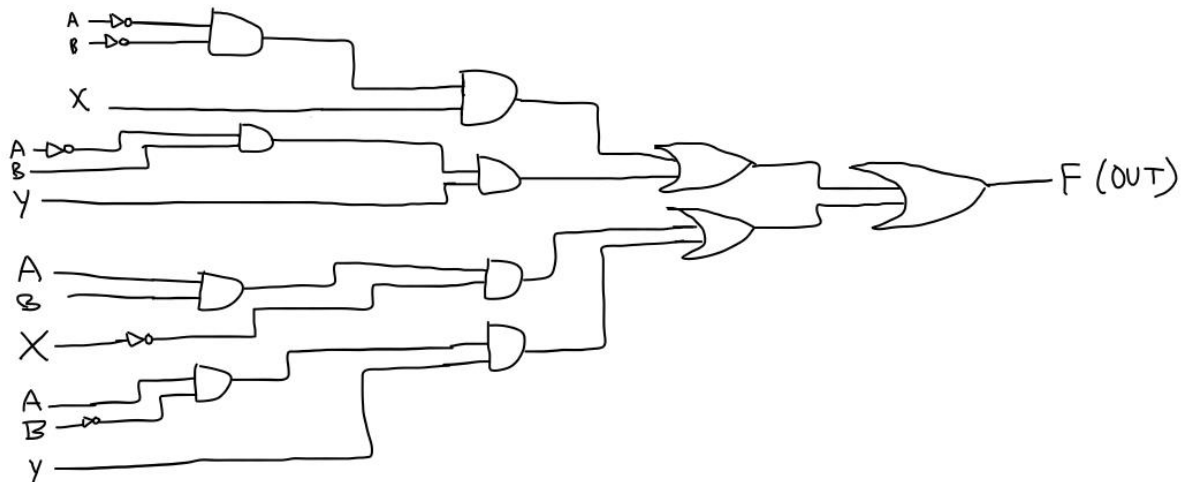
$$F = \sum m(3,5,6,7,8,12,13,14)$$

3. Write the minimum logic expression, as a sum-of-products for the function F.



$$F = XA'B' + A'BY + ABX' + YB'A$$

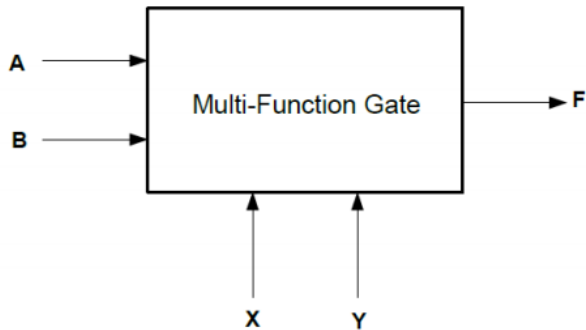
4. Draw logic diagrams for the above expressions using AND's, OR's, and Inverters.



Multi-Function Gate Schematic

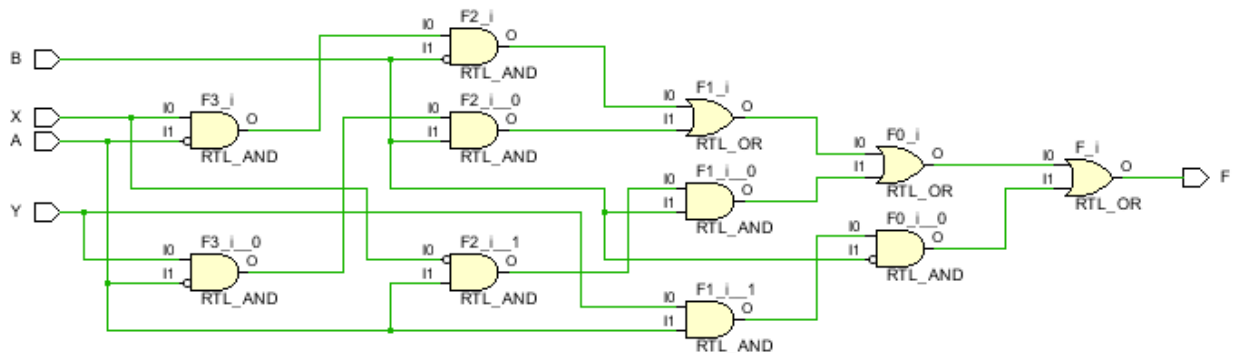
Logic Diagram:

The resulting schematic diagram for the MULTI-FUNCTIONAL gate.



Design Specification Plan:

This is the design specification for an MULTI-FUNCTIONAL gate. It has four inputs (X, Y, A, and B) and one output (F).



Flow Navigator

- Run Simulation
- RTL ANALYSIS
 - Open Elaborated Design
 - Report Methodology
 - Report DRC
 - Report Noise
 - Schematic
- SYNTHESIS
 - Run Synthesis
 - Open Synthesized Design
- IMPLEMENTATION**
 - Run Implementation
 - Open Implemented Design
 - Constraints Wizard
 - Edit Timing Constraints
 - Report Timing Summary
 - Report Clock Networks
 - Report Clock Interaction
 - Report Methodology
 - Report DRC
 - Report Noise
 - Report Utilization
 - Report Power
 - Schematic
- PROGRAM AND DEBUG
 - Generate Bitstream
 - Open Hardware Manager

Sources Netli x Device ? _ □ □

- multi_function_gate
 - Nets (10)
 - Leaf Cells (6)

Package x Device x multi_function_gate.v x m_func_sim1.v

Source File x Clock Region ? _ □ □

multi_function_gate.v

☒ Enabled

Location: C:/Users/spens/Desktop

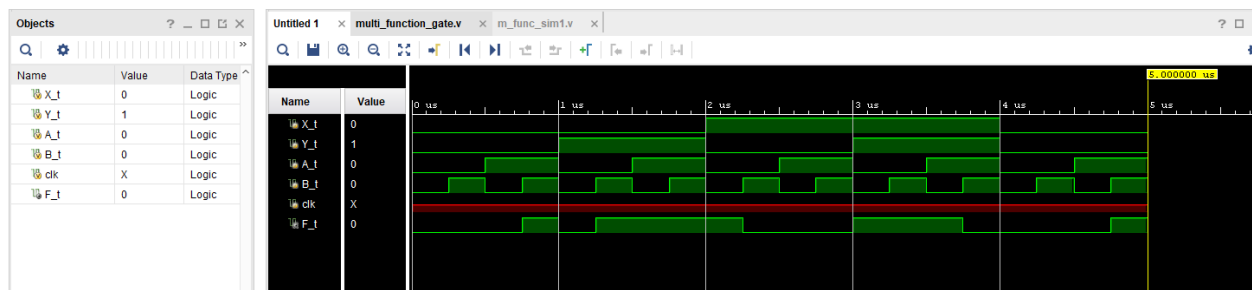
General Properties

Tcl Console Messages Log Reports Design Runs Timing Power DRC Package Pins I/O Ports

Name	Direction	Neg Diff Pair	Package Pin	Fixed	Bank	I/O Std	Vcco	Vref	Drive Stren
Scalar ports (5)									
A	IN		V14	<input checked="" type="checkbox"/>	14	LVC MOS33*	3.300		
B	IN		V15	<input checked="" type="checkbox"/>	14	LVC MOS33*	3.300		
F	OUT		U16	<input checked="" type="checkbox"/>	14	LVC MOS33*	3.300		12
X	IN		V16	<input checked="" type="checkbox"/>	14	LVC MOS33*	3.300		
Y	IN		V17	<input checked="" type="checkbox"/>	14	LVC MOS33*	3.300		

Results Statement:

Here is the simulation results for the MULTI-FUNCATIONAL gate.

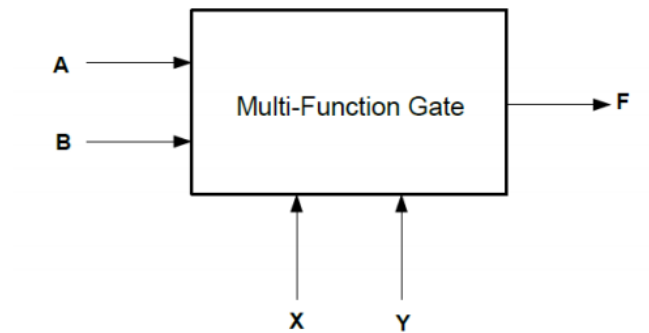


SW6	SW7	SW0	SW1	LED7
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Multi-Function Gate Verilog

Logic Diagram

The results from the simulation testing for the simulation for a MULTI-FUNCTIONAL gate.



Design Specification Plan

This is the design specification for an MULTI-FUNCTIONAL gate. It has four inputs (X, Y, A, and B) and one output (F).

```
6 // Create Date: 10/04/2020 05:35:03 PM
7 // Design Name:
8 // Module Name: multi_function_gate
9 // Project Name:
10 // Target Devices:
11 // Tool Versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 ///////////////////////////////////////////////////////////////////
21
22
23 module multi_function_gate(
24     input X,
25     input Y,
26     input A,
27     input B,
28     output F
29 );
30     wire X, Y, A, B, F;
31     assign F = (X & ~A & ~B) | (Y & ~A & B) | (~X & A & B) | (Y &
32 endmodule
33
```

Flow Navigator

- Run Simulation
- RTL ANALYSIS
 - Open Elaborated Design
 - Report Methodology
 - Report DRC
 - Report Noise
 - Schematic
- SYNTHESIS
 - Run Synthesis
 - Open Synthesized Design
- IMPLEMENTATION**
 - Run Implementation
 - Open Implemented Design
 - Constraints Wizard
 - Edit Timing Constraints
 - Report Timing Summary
 - Report Clock Networks
 - Report Clock Interaction
 - Report Methodology
 - Report DRC
 - Report Noise
 - Report Utilization
 - Report Power
 - Schematic
- PROGRAM AND DEBUG
 - Generate Bitstream
 - Open Hardware Manager

IMPLEMENTED DESIGN - xc7a35tcbg236-1 (active)

Sources | **Netli** | **Device** | **Package** | **Device** | **multi_function_gate.v** | **m_func_sim1.v**

multi_function_gate

- Nets (10)
- Leaf Cells (6)

Source File | **Clock Region**

multi_function_gate.v

☒ Enabled

Location: C:/Users/spens/Desktop

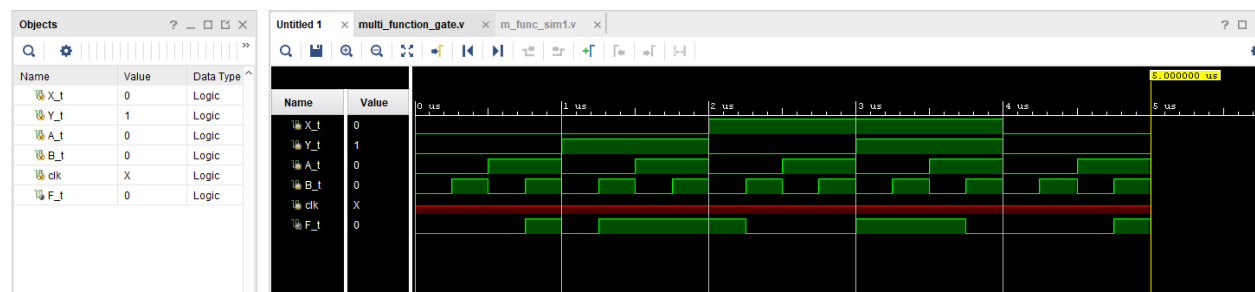
General | **Properties**

Tcl Console | **Messages** | **Log** | **Reports** | **Design Runs** | **Timing** | **Power** | **DRC** | **Package Pins** | **I/O Ports**

Name	Direction	Neg Diff Pair	Package Pin	Fixed	Bank	I/O Std	Vcco	Vref	Drive Stren
Scalar ports (5)									
A	IN		V14	<input checked="" type="checkbox"/>	14	LVC MOS33*	3.300		
B	IN		V15	<input checked="" type="checkbox"/>	14	LVC MOS33*	3.300		
F	OUT		U16	<input checked="" type="checkbox"/>	14	LVC MOS33*	3.300		12
X	IN		V16	<input checked="" type="checkbox"/>	14	LVC MOS33*	3.300		
Y	IN		V17	<input checked="" type="checkbox"/>	14	LVC MOS33*	3.300		

Results Statement

Explanation of pictures and tables.

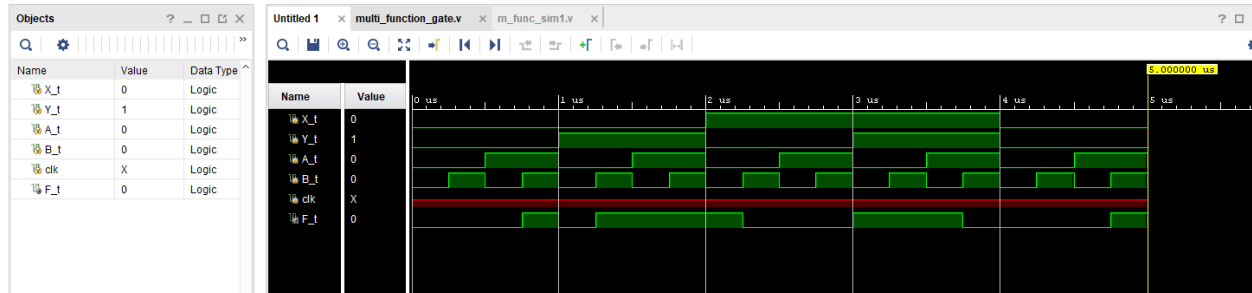


SW6	SW7	SW0	SW1	LED7
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Test Plan

The initial design was tested through a simulation based off of the test bench, which was programmed for every possible input, which would give every output.

(Multi-functional gate)



Conclusion

My results were what I expected, and all the test and simulation results were correct and agreed with the logic that was programmed. By running the gate through a simulation and then analyzing it was interesting, and lead to further understanding of the logic gates functionality and how it functions.

1. Can this Multi-Function Gate be operated as an Inverter? If yes, explain how.

It can be used as a NOT gate if both inputs are joined together and the X and Y input are high (set as 1). When this is done, X and Y function as a NAND gate, and when that happens, the NAND gate operates like an inverter.

2. Will the change in the number of inputs or outputs affect the number of operation select lines? Explain.

The number of outputs does not have an effect on the number of operation select lines. This is because each output is connected to a certain operation. So, by changing the amount of operations will increase the amount of outputs, but increasing the amount of outputs does not effect on the amount of operation select lines in the circuit.

3. Will the change in the number of functions alter the number of operation select lines? Explain.

Yes, if the number of functions is increased, the number of operation select lines are also increased, since there will be more functions to select, so there will need to be more operation select lines. The square root of the number of functions is how many select lines you will need.

4. Have you met all the requirements of this lab (Design Specification Plan)?

Yes.

5. How should your design be tested (Test Plan)?

My design was tested by running the circuit design through a detailed simulation that tested each input value, in all the combinations, and after checking each one, the results match the truth table. That would verify that the logic design is correct.

Answers To In-Class Activities:

1. C (RTL Analysis > Schematics)

2. D ($F = !X!YAB + !XY!AB + !XYA!B + !XYAB + X!Y!A!B + XY!A!B + XY!AB + XYA!B$)