Justin VanWinkle
EEE 3342C
Experiment #3
3 June 2015

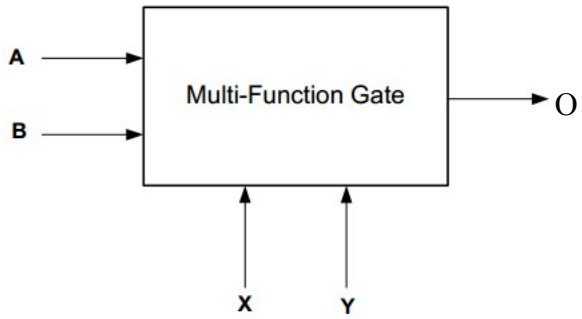
Justin VanWinkle
3 June 2015
Experiment #3

Objective

This experiment is intended to create and analyze a Multi-function logic gate using the Xilinx ISE design tools. The logic gate will be analyzed using the Xilinx ISE simulation tool as well as on a BASYS 2 FPGA development board.

Diagrams

The following is a block diagram that shows an over-arching view of the multi-function gate that will be created in this experiment.



By use of the selector lines, X and Y, the multi-function gate will change the function performed on inputs A and B.
The following logic diagrams show the expected output of the function performed on A and B based on the given inputs of selector lines X and Y

When X=0 and Y=0:

A	B	O
0	0	0
1	0	1
0	1	0
1	1	1

When X=1 and Y=0:

A	B	O
0	0	1
1	0	1
0	1	0
1	1	0

When X=0 and Y=1:

A	B	O
0	0	0
1	0	0
0	1	1
1	1	1

When X=1 and Y=1:

A	B	O
0	0	1
1	0	0
0	1	1
1	1	0

Equipment

The equipment used for this experiment included a computer with the Xilinx ISE design suite and Digilent ExPort installed as well as a Digilent BAYSYS 2 development board.

Procedure

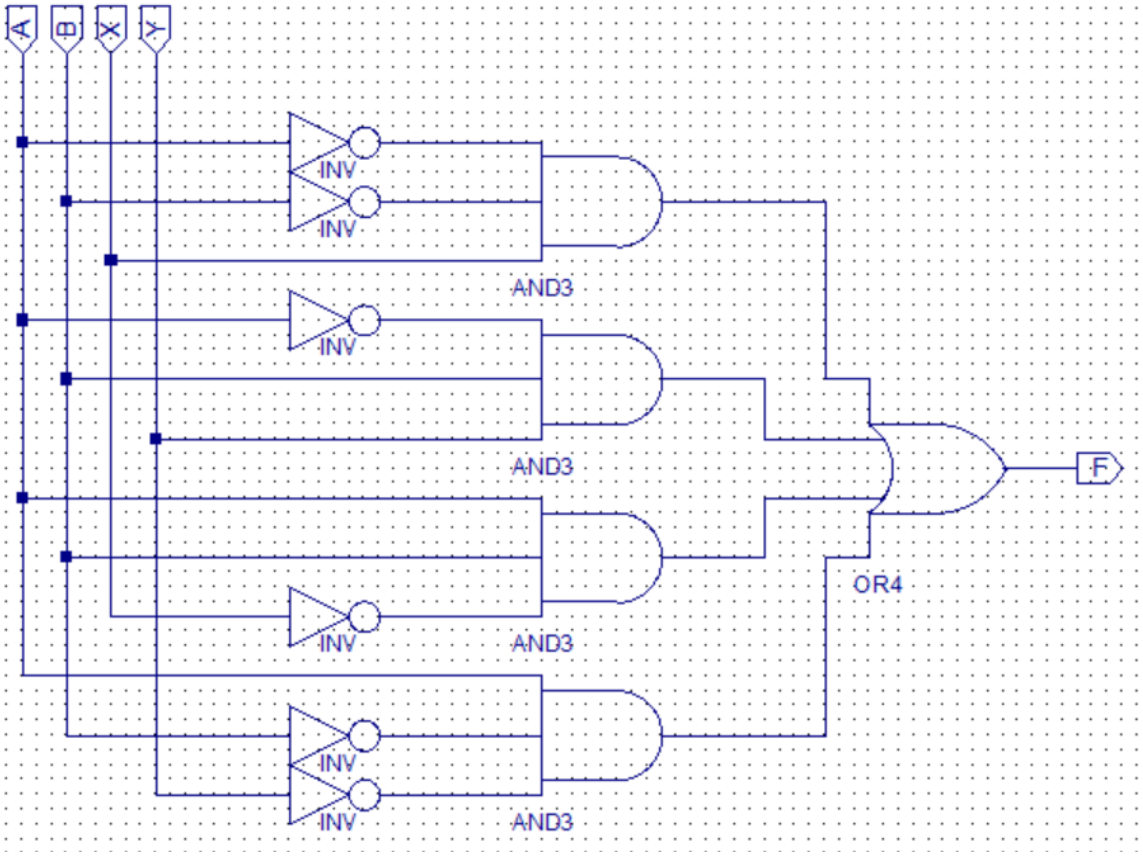
Part 1

The first step in this experiment should be analyzing and understanding what our expected output should be based on any given set of inputs. This will give us a better understanding of how to build the schematic of the circuit. The below table is a comprehensive view of the logic possibilities in the circuit at any given moment, matched with their respective output

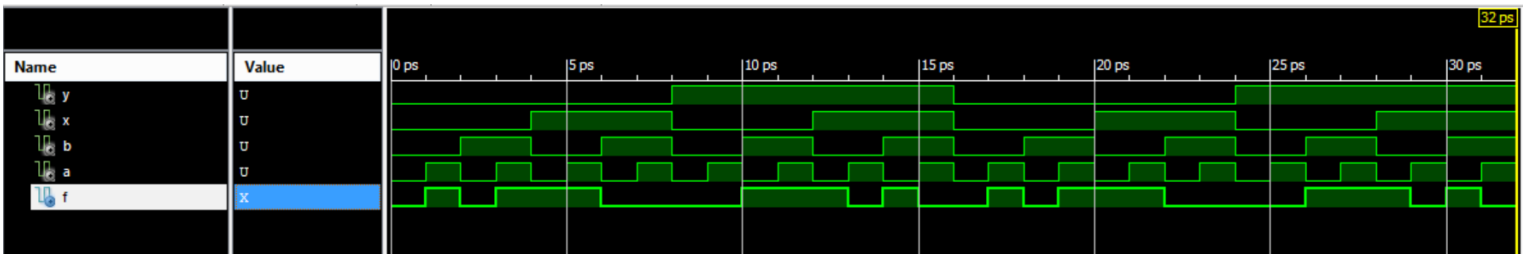
A	B	X	Y	O
0	0	0	0	0
1	0	0	0	1
0	1	0	0	0
1	1	0	0	1
0	0	1	0	1
1	0	1	0	1
0	1	1	0	0
1	1	1	0	0
0	0	0	1	0
1	0	0	1	0
0	1	0	1	1
1	1	0	1	1
0	0	1	1	1
1	0	1	1	0
0	1	1	1	1
1	1	1	1	0

The above table shows the expected output based on the function performed on A and B when X and Y are of certain inputs.

Justin VanWinkle
3 June 2015
Experiment #3



After completion of the above schematic capture, the operator then synthesized and implemented the design. Using the isim interface, the operator next simulated and tested the schematic design for its accuracy based on all possible combinations of input and function selects. The results of this simulation is as follows:



Once the simulation had been proven to provide proper output, the operator was ready to transfer the implementation over to the BASYS 2 development board. To do this, the operator used the ISE design tools to create area constraints such that the inputs were tied to individual switches on the BASYS 2 development board that represented high and low inputs into the gate and the output was tied to a single LED to show if the output was high or low (The chosen switches and LED can be seen in the expected results section). Next, the operator re-implemented the design with the new physical I/O assignments. Once the design was implemented, the ISE design tools were used to generate the programming file that is used to program the BASYS 2 development board. Doing this produced a *.bit file that was used for programming the board. To transfer the file to the

BASYS 2 board, Digilent Adept was used. Once in Adept, with the BASYS 2 board powered on, the operator selected the initialize chain button to establish communication between the computer and the BASYS 2 board. Next, the operator used the browse button next to the "FPGA" text to select the newly created *.bit file for programming the BASYS 2 baord. Finally, the "program chain" button was used to push the program to the BASYS 2 board and the operator could test the program on the board for proper output.

Part 2

For the second part of the experiment, the operator repeated the first part using the VERILOG module in place of the schematic capture. This very slightly changed the process up to the point of creating a simulation, but beyond that, the process is identical.

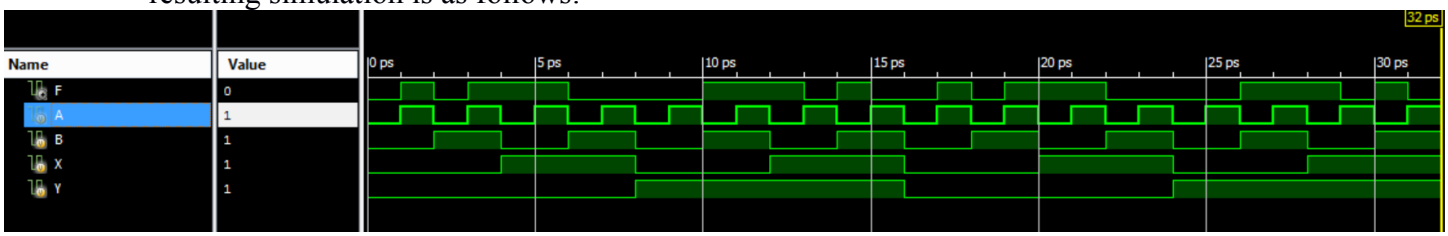
When creating a VERILOG module, the user will select inputs and outputs based on the need of the problem and will also select a naming convention for each. The code for the multi-function gate is as follows:

```
module Exp3Verilog(
    input A,
    input B,
    input X,
    input Y,
    output F
    );

    assign F = (~A & ~B & X) | (~A & B & Y) | (A & B & ~X) | (A & ~B & ~Y);

endmodule
```

After completion of each of the above VERILOG module, the operator then synthesized and implemented the design. Next, the operator used the newly written code to simulate the intended circuit for its accuracy based on all possible combinations of inputs. The resulting simulation is as follows:



Once the simulation had been proven to provide proper output, the operator was ready to transfer the implementation over to the BASYS 2 development board. To do this, the operator used the ISE design tools to create area constraints such that the inputs were tied to individual switches on the BASYS 2 development board that represented high and low inputs into the gate and the output was tied to a single LED to show if the output was high or low (The chosen switches and LED can be seen in the expected results section). Next, the operator re-implemented the design with the new physical I/O assignments. Once the design was implemented, the ISE design tools were used to generate the

programming file that is used to program the BASYS 2 development board. Doing this produced a *.bit file that was used for programming the board. To transfer the file to the BASYS 2 board, Digilent Adept was used. Once in Adept, with the BASYS 2 board powered on, the operator selected the initialize chain button to establish communication between the computer and the BASYS 2 board. Next, the operator used the browse button next to the "FPGA" text to select the newly created *.bit file for programming the BASYS 2 baord. Finally, the "program chain" button was used to push the program to the BASYS 2 board and the operator could test the program on the board for proper output.

Expected Results of Circuit

The following chart shows the expected output based on each possible scenario of inputs.

Mulit-function Gate:

A (SW0)	B (SW1)	X (SW6)	Y (SW7)	O (LED7)
0	0	0	0	0
1	0	0	0	1
0	1	0	0	0
1	1	0	0	1
0	0	1	0	1
1	0	1	0	1
0	1	1	0	0
1	1	1	0	0
0	0	0	1	0
1	0	0	1	0
0	1	0	1	1
1	1	0	1	1
0	0	1	1	1
1	0	1	1	0
0	1	1	1	1
1	1	1	1	0

From this chart, we derive the following minimal logic circuit:
O = ABX' + A'BY + A'B'X + AB'Y'

Design Specification Plan

The results of this experiment draw a sound conclusion that the objective has been met by the execution of the procedure. This is given by the fact that simulations and test results match verbatim to that of the expected results based on the calculations performed beforehand. The methodology chosen for this experiment followed a calculation-first route such that the operator would know what to expect. This also provided simplicity for the operator when designing the schematic as well as the VERILOG module. Given that a successful attempt was made to design, simulate, and implement a multi-function gate, which was the objective of this experiment, this experiment has met all of its

requirements.

Test Plan

This experiment should be tested and verified by the following means. The tester should obtain a BASYS 2 development board with this program loaded on it. Once the tester has set up the board and powered it on, he should test for output based on the minimal boolean expression we developed above. Any time one of the conditions is met using the proper switches, LED 7 should turn on. Otherwise, and in all other arrangements, the LED should be off.

Results

Please see the section entitled procedures for simulation results.
The outcome of this experiment accurately matched that of the calculations performed and presented in the expected results section of this report. This means that a multi-function gate was implemented such that the function chosen to be performed on its inputs were the result of the selection from its function selector inputs. Given two selector inputs, the multi-function gate was capable of performing 4 separate functions on the two input lines given, providing a total of 16 input-output combinations.

Conclusion

In summation, this experiment suggest evidence that any circuit could be simulated through the use of a multi-function gate. This could serve several purposes, the most notable of which is simplification of logic design - especially in vastly larger circuits. This experiment was particularly useful for learning to apply boolean simplification.

Questions

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

  Yes. This multi-function gate can serve as an inverter for both the AND and OR gates based on the input from the X and Y wires.

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

No. The operation select lines only serve the purpose of determining what operation is being performed on the inputs. Changing the number of inputs will only change the number of possible results.

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

Yes. More functions will require more operation select lines and fewer functions will require fewer operation select lines.

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

See section above entitled Design Specification Plan. As per this section, all requirements of this lab have been met.

- How should your design be tested (Test Plan)?

See section above entitled Test Plan.