Justin VanWinkle

EEE 3342C

Experiment #1

3 June 2015

## Objective

In this experiment, the author intends to familiarize himself with the Xilinx FPGA ISE design tools as well as the Digilent BASYS 2 development board through the design of several basic logic circuits via both schematic capture and the VERILOG language.

## Equipment

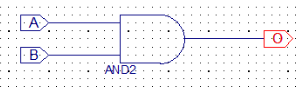
The equipment used for this experiment included a computer with the Xilinx ISE design suite and Digilent ExPort installed as well as a Digilent BAYSYS 2 development board.

## Procedure

### Parts 1 & 2

The initial portion of this experiment was focused on learning how everything connected and interfaced. Therefore, the operator used the Xilinx FPGA ISE design tools to create several basic logic gates through the schematic capture tool to interface with the BASYS 2 development board such that the BASYS 2 development board mirrored what was designed via the schematic capture tool. The logic gates and their individual schematics that were designed are shown below in order of creation. Each input and output show the physical switch or LED through which the input or output was routed as well as the pin number for the switch or LED, in respective order.

And Gate:

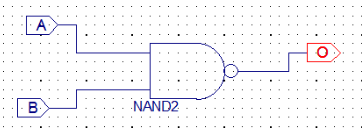


SW0 - P11

SW1 - L3

LED0 - M5

Nand Gate:

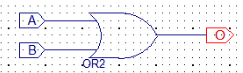


SW0 - P11

SW1 - L3

LED0 - M5

Or Gate:



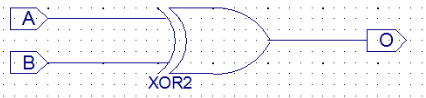
SW0 - P11

SW1 - L3

LED0 - M5

Xor Gate

SW0 - P11



SW1 - L3

LED0 - M5

Inverter Gate:

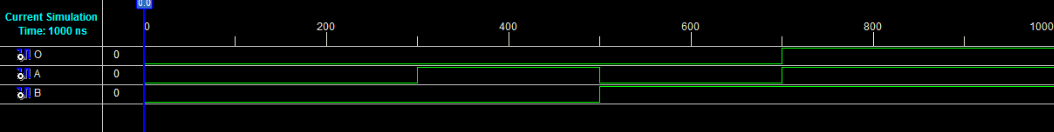


SW0 - P11

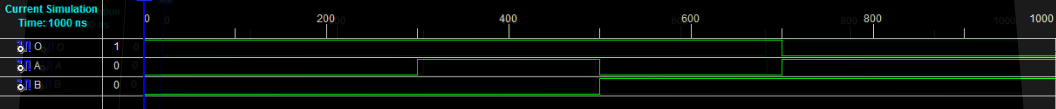
LED0 - M5

After completion of each of the above schematic captures, the operator then synthesized and implemented the design. Next, the operator created a test bench waveform so as to simulate and test each individual schematic design for its accuracy based on all possible combinations of input. The results of these simulations, in matching order to the schematics above, are as follows:

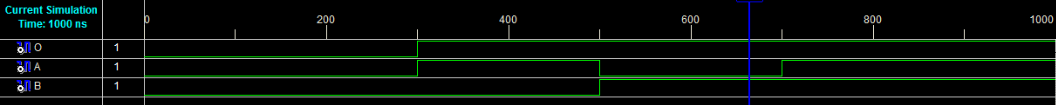
And Gate:



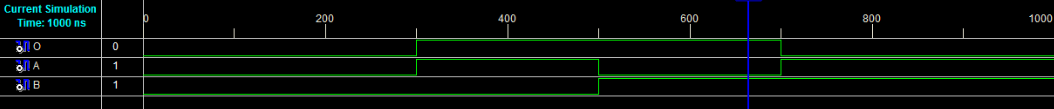
Nand Gate:



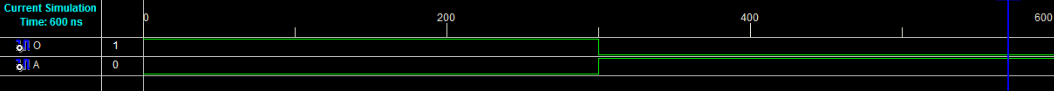
Or Gate:



Xor Gate:



Inverter Gate:



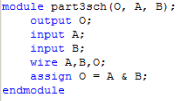
Once the design had been proven to provide proper output in simulation, the operator was ready to transfer the implementation over to the BASYS 2 development board. To do this, the operator used the ISE design tools to create area constraints such that the inputs were tied to individual switches on the 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. 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 3

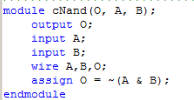
For the next portion of the experiment, the operator familiarized himself with the same process mentioned for parts 1 and 2, except by using a VERILOG module in place of schematic capture. This very slightly changed to process up to the point of creating a simulation, but beyond that, the process is identical. Therefore, the author will cover the differences and refer the reader to parts 1 and 2 for continuation.

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. For this portion of the experiment, the operator repeated the process using the same gates as used in parts 1 and 2. The code for each can be seen below in matching order to the previous parts.

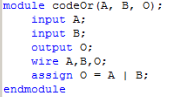
And Gate:



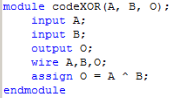
Nand Gate:



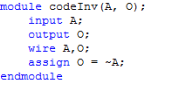
Or Gate:



Xor Gate:

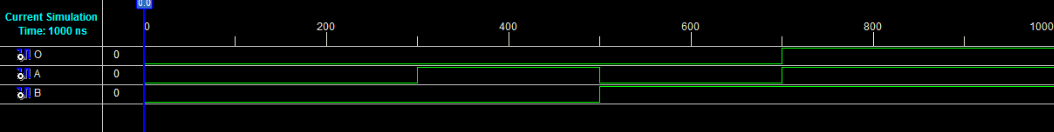


Inverter Gate:

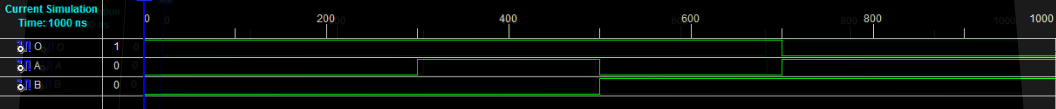


After completion of each of the above VERILOG modules, the operator then synthesized and implemented the design. Next, the operator created a test bench waveform so as to simulate and test each individual VERILOG module for its accuracy based on all possible combinations of input. The results of these simulations, in matching order to the code blocks above, are as follows:

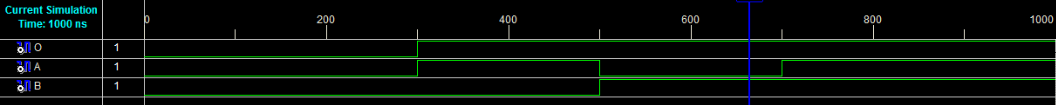
And Gate:



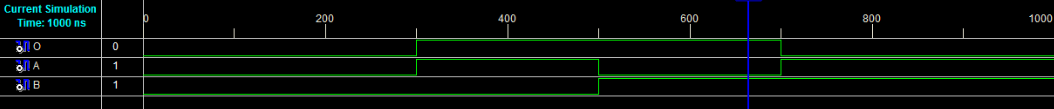
Nand Gate:



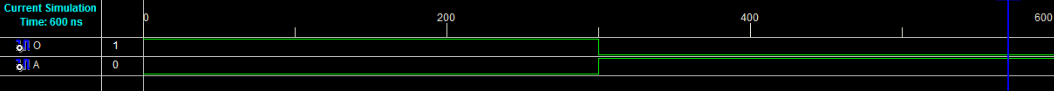
Or Gate:



Xor Gate:



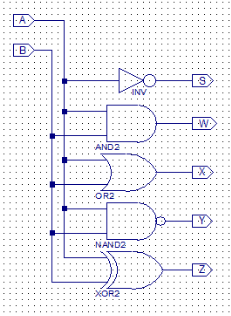
Inverter Gate:



As previously stated, the process beyond this point is identical to that of the previous section. The author urges the reader to use the text following the previous simulations for the procedure used beyond simulation.

### Part 4

The final portion of this experiment involves an implementation that combines all of the logic gates used in this experiment for a two-input five-output logic circuit through the use of both the schematic capture tool as well as the VERILOG module. Just as before, the operator created a schematic capture as follows:



LED0 - M5

LED1 - M11

LED2 - P7

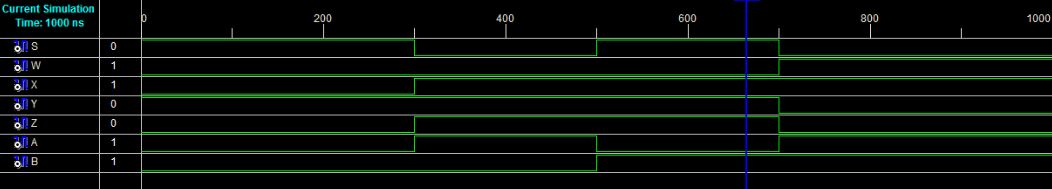
LED3 - P6

LED4 - N5

SW0 - P11

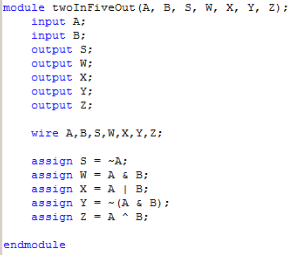
SW1 - L3

This schematic represents the inputs A and B with the outputs S (Inverter Gate), W (And Gate), X (Or Gate), Y (Nand Gate), and Z (Xor Gate). Using this schematic capture, the operator created a test bench simulation that yielded the following:

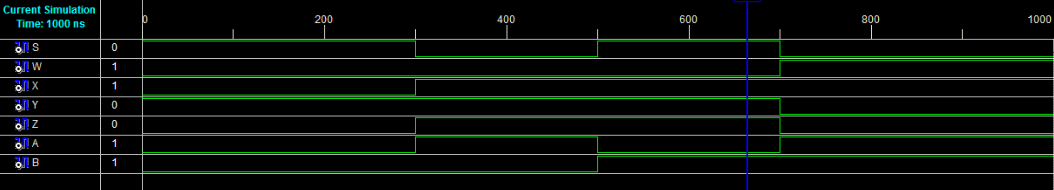


Upon verification of the proper outputs, the operator then followed the procedure outlined in the first section of this report to create an area constraint and load the program onto the BASYS 2 development board.

Once the output was verified on the BASYS 2 board, the operator persisted to complete the same circuit using the VERILOG module in place of the schematic caputre. Keeping the naming convention of the schematic capture, the resulting code was as follows:



Next, the operator created a test bench simulation of the VERILOG module to test for proper outputs and found the following results:



The operator then created an area constraint and loaded the program to the board (see the first section of this report for detailed information) and verified that the correct outputs were given.

## Expected Results of Circuits

The following charts are the expected results of each part of the experiment above and are to be referenced for accuracy.

And Gate:

|  |  |  |
| --- | --- | --- |
| A (SW0) | B (SW1) | O(LED0) |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Nand Gate:

|  |  |  |
| --- | --- | --- |
| A (SW0) | B (SW1) | O(LED0) |
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Or Gate:

|  |  |  |
| --- | --- | --- |
| A (SW0) | B (SW1) | O(LED0) |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Xor Gate:

|  |  |  |
| --- | --- | --- |
| A (SW0) | B (SW1) | O(LED0) |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Inverter Gate:

|  |  |
| --- | --- |
| A (SW0) | O(LED0) |
| 0 | 1 |
| 1 | 0 |

Two-Input Five-Output Circuit:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A (SW0) | B (SW1) | S(LED0) | W(LED1) | X(LED2) | Y(LED3) | Z(LED4) |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 |

## Design Specification Plan

In this experiment, the operator created several circuits through two different means while using the equipment provided which matched the equipment required for this experiment. As such, the operator was able to familiarize himself with all of the equipment and software needed to complete this experiment. And since the purpose of this experiment was to gain familiarization with modeling, simulation, and implementation of digital circuits using Xilinx’s FPGA ISE design tools, the operator was able to complete several scenarios through multiple means so as to gain the aforementioned familiarization.

## Test Plan

Given the simple nature of this experiment, the test plan will simply involve verification of the tables on page 8 of this report. To do this, the tester will set the switches to each of the positions in the table on page 8 that corresponds to the program that is currently loaded onto the BASYS 2 development board. At each given input scenario, the tester will verify that the LED that matches the corresponding output in the table is correct. If all LEDs match their corresponding output based on a given input scenario, then the test will have been successful.

## Results

The results of this experiment were found to match exactly those of the simulations that can be found on the previous pages of this report. The BASYS 2 development board clearly showed outputs that matched the tables on page 8 of this report based on their corresponding inputs in each given logic circuit.

## Conclusion

In this experiment, the operator learned much about the hardware and software required. His familiarization increased significantly. Moreover, his confidence in the interfaces provided increased as the operator was able to see and verify several circuit designs through multiple means.