CS373 LM8: Introduction to FPGA’s and VHDL using   
Xilinx Vivado® and the BASYS 3 Board

Name(s) Brycen Martin

|  |  |  |
| --- | --- | --- |
| **CATEGORY** | **POINTS** |  |
| Majority4 | 48 | 50 |
| TOTAL | 48 | 50 |
| I don’t think you ran the simulation correctly. See section 4 for what the simulation trace should look like. |  |  |

## Learning Goals for This Study Guide

* Understand where to locate additional information about FPGA design with the Basys 3
* Gain an introductory understanding of how and FPGA works and how they are programmed.
* Learn the key features of the Bassys 3 board.
* Understand how to test and verify that the Basys 3 board is working correctly.
* Learn beginning VHDL.
* Learn how to simulate a design.
* Challenge critical thinking skills by implementing a new project.

## Instructions for This Project

* You may work by yourself or with another person. You must **both** submit the code to whitgit along with your changes to this document. You must each answer the questions in this document by yourself, but you may consult each other on your answers.
* Answer the questions on the last page of this document and submit this document (with your answers in it) to Blackboard.
* Follow the instructions given in class to install Vivado on your machine (or use the machines in the lab – if you use lab machines which already have Vivado installed, you will need to install the board files for the BASYS 3)
* Create the majority project using the TCL/Bash scripts provided in class. You will need to configure the TCL script so that it builds your majority project.

# Majority4 Project

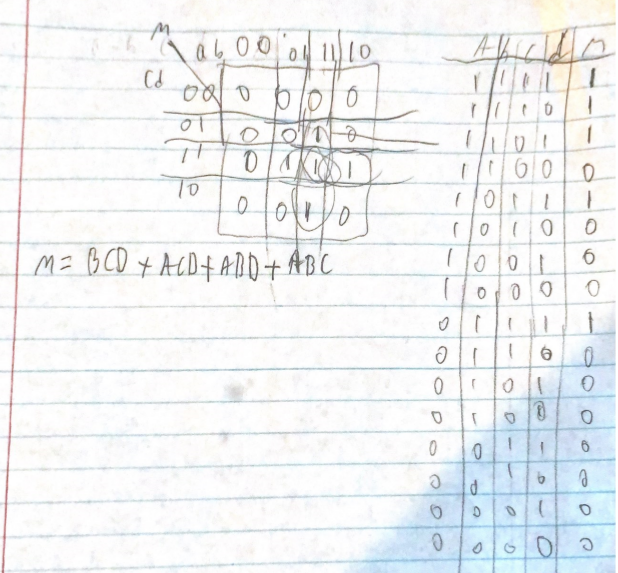
Now it’s time to put your newly learned skills to use. For this project you will design and implement the logic for a **Majority Circuit** that has four input bits.

## Marjority4 Project Specifications

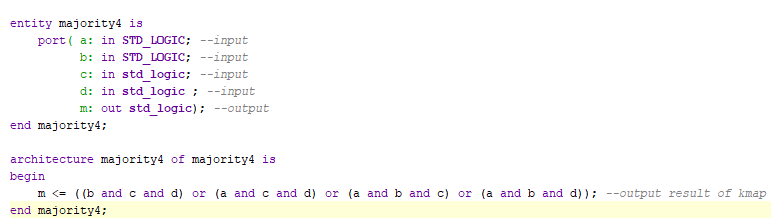
* **Pull the LM8\_Majority starter project from whitgit using vscode**
* Create a **Majority** VHDL module that has:
  + Four STD\_LOGIC inputs, a, b, c and d.
  + One STD\_LOGIC output m :
    - This output will be high if 3 or 4 of the a,b,c,d inputs are active.
    - This output will be low if 0, 1, or 2 of the a, b, c, d inputs are active.
* **Create a testbench vhdl file to test all combinations of a, b, c, d of your Majority component**.
  + You could do this using nested simulation code loops for loops in a test bench, but this is not required for this lab. We will cover how to write more complex test bench code in class.
  + Make sure to configure the setup TCL script so that this simulation will be generated.
* Create a **Majority\_top** VHDL module that interfaces to the BASYS 3 board.
  + This should include the Majority component and map the board inputs/outputs to this component.
  + Setup the project so that SW(3), SW(2) … SW(0) map to the inputs of your majority circuit **a**, **b, c**, and **d** respectively.
  + Setup the project so that LED(0) is mapped to the output **m** of your majority circuit.
* **You must program the FPGA board and verify the majority circuit works: if any 3 of the four switches are high the LED should turn on. It should also turn on if all four switches are set to high.**
  + Modify the setup TCL script so that it will generate both the simulation and the bitstream

## Majority Circuit Design

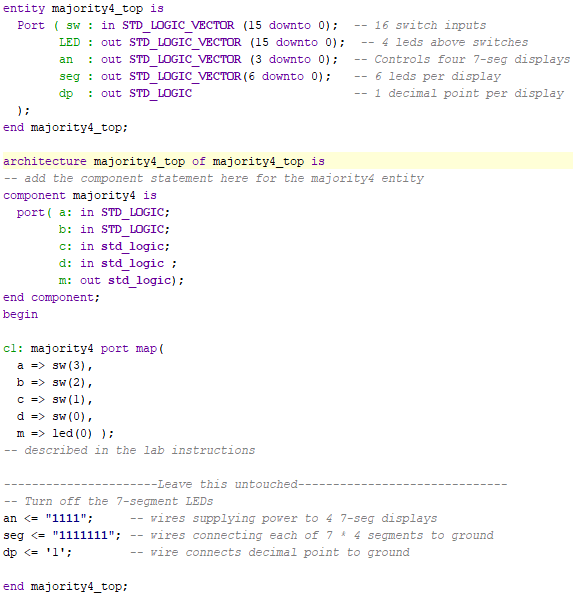
1. Create a Karnaugh map for the Majority circuit include the k-map and the resulting Boolean equation for the Majority circuit here:



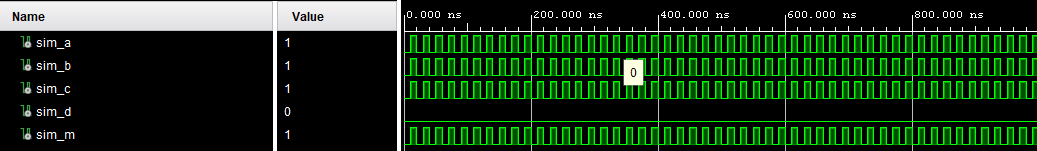
1. Include a copy of your Majority VHDL module code here. Make sure it is commented well!



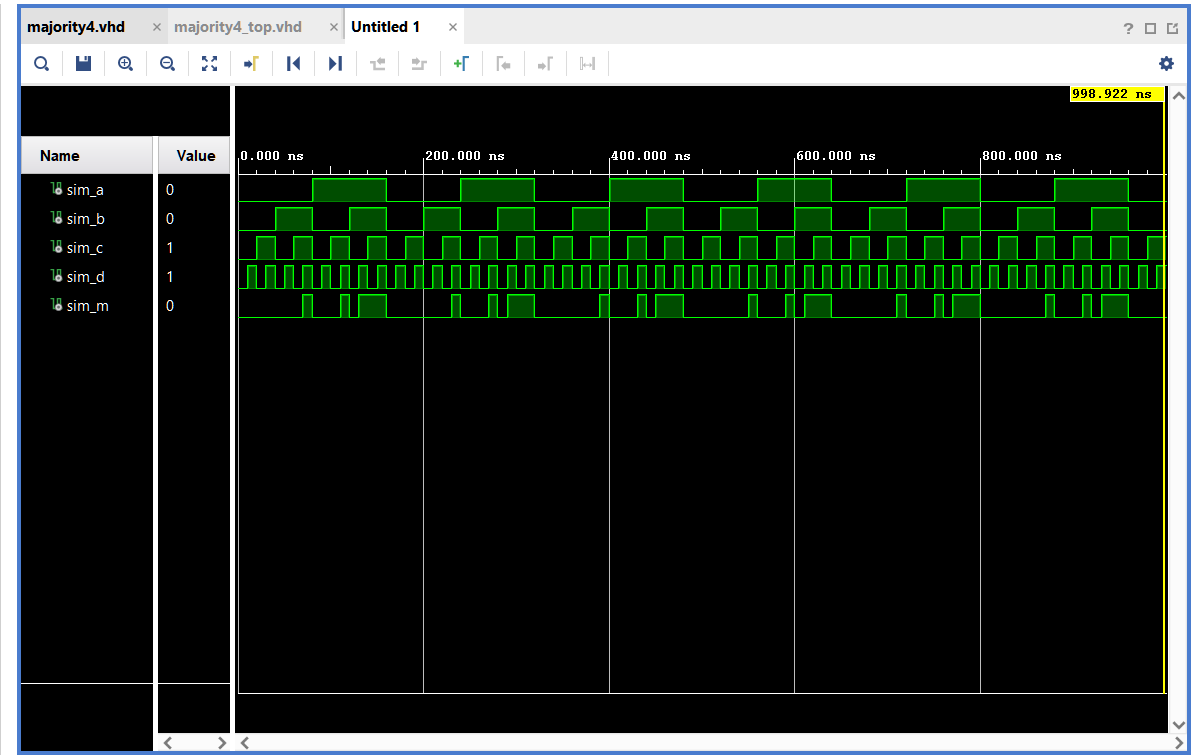
1. Include a copy of your Majority\_top level VHDL module code here. Make sure you comment it well!



1. Include a screen snip of your Majority circuit simulation testbench running here:



-2 A correct simulation trace will look like this:



1. Describe the design process, what issues you faced, and how it works.

To start the process the first thing we had to do was design the Kmap to find what our output would look like thus, enabling us to build logic for it. After that we had to use the examples to help us build our majority 4 module code and then after we did our top-level code as it requires the module code to work. From their it was just making sure we hit the right button to see our testbenches. The problems we faced were mainly gen.sh errors and its unwillingness to compile for us as we had no experience with how it like to work yet and as such it made it difficult for us to track down and fix. The next biggest problem we faced was syntax as it is similar to c++ but not the same so we kept getting tripped up with errors due to incorrect syntax.