**CMPEN 270: Digital Design: Theory and Practice**

**Module 1 Lab: Getting Started With Xilinx Vivado**

**Due: 1/16/2022**

**Robert Myers, rfm5594@psu.edu**

**Grading Rubric**

|  |  |
| --- | --- |
| **Criteria** | **Grade** |
| Initial grade, based on how well the functional specification is met  Excellent (exceptional achievement) 90-100%  Good (extensive achievement) 80-89%  Satisfactory (acceptable achievement) 72-79%  Poor (minimal achievement) 65 to 71%  Failure (inadequate achievement) 0-64% | / 25 |
| Modification for design documents  Block diagrams  State diagrams  State tables  Other |  |
| Modification for coding style, comments, efficiency  Header comments for files  Other comments (informative but not excessive)  Proper code indenting, alignment, use of whitespace  Code is clean (doesn't have commented out code without a good reason)  Self documenting code (good signal and component names, clear structure, etc.)  General approach (algorithms)  Coding details (operations)  Proper use of components  VHDL matches design documents  Other |  |
| Modification for sections in this report  Design  Verification  Evaluation  Questions  Other |  |
| Bonus (optional challenge, etc.) |  |
| Penalties  No grade until all deliverables are submitted, late submission penalty for anything submitted late  Late submission: (<1 day) -10%, (<1 week) -30%, (<2 weeks) -50%, (>=2 weeks) -99%  Attachments missing, not in order, instructions not followed: -10%  Other |  |
| TOTAL (Max is 100% of total points unless specified otherwise in handout) | / 25 |

**ACKNOWLEDGEMENT**

This work is entirely my own and I did not provide any assistance except as noted.

100% Robert Myers \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Robert Myers\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ATTACHMENTS**

The following are attached, in this order:

* Test Worksheet

**DESIGN**

There was no design for this lab.

**VERIFICATION**

My verification method was to start by computing the binary addition by hand. Once I had computed the sum (in 8-bit unsigned binary), I verified by converting the operands and sum to decimal. If the operands and sum were true, I knew I had the correct binary sum. I then moved to Vivado where I forced the operands as constants to A and B respectively. If there was a carry out in the 29 place it was noted in the manual calculation step, reflected in the C\_out output, and 25610 was added to the binary sum calculated by Vivado (converted to decimal).

**EVALUATION**

No performance metrics are required for this lab.

**QUESTIONS**

1. How many test cases are needed to test the 8-bit RippleCarryAdder exhaustively (every possible input combination)? Explain how you got this number.
   1. For the first operand (call this A) there are 255 combinations ranging from 0000 0000 to 1111 1111. For the second operand (call this B) there are 255 combinations as well. To test every possible case and assuming A + B = C and B + A = C should be treated as unique cases, there would need to be 2552 or **65,025** test cases.

