ECE 385 Lab 5 Report Outline

- ❑ **Introduction**
  - ❑ Summarize the basic functionality of the multiplier circuit
- ❑ **Pre-lab question**
  - ❑ Rework the multiplication example on page 5.2 of the lab manual, as in compute 00000111 * 11000101 in a table similar to the example
- ❑ **Written description and diagrams of multiplier circuit**
  - ❑ Summary of operation
    - ❑ Explain in words how operands are loaded, how the multiplier computes its result, how the result is stored, etc.
  - ❑ Top Level Block Diagram
    - ❑ This can be generated from the RTL viewer. *Please only include the top level diagram and not the RTL view of every module*.
  - ❑ Written Description of .sv Modules
    - ❑ List all modules used in a format shown in the appendix of this document
  - ❑ State Diagram for Control Unit
    - ❑ This can be done in a program like Visio, but if the Quartus state diagram generator is used, you must label the states and transitions or you will lose points!
- ❑ **Annotated pre-lab simulation waveforms**
  - ❑ Create your own testbench for your multiplier which displays the following
    - ❑ 4 operations where operands have signs (+*+), (+*-), (-*+) and (-*-)
    - ❑ Waveform must have notes that clearly show the operands as well as the result, etc.
- ❑ **Answers to two post-lab questions**[1]
  - ❑ Fill in the table shown in 5.6 with your design's statistics
  - ❑ Write down several ideas on how the maximum frequency of your design could be increased or the gate count could be decreased
- ❑ **Conclusion**
  - ❑ Discuss functionality of your design. If parts of your design didn't work, discuss what could be done to fix it
  - ❑ Was there anything ambiguous, incorrect, or unnecessarily difficult in the lab manual or given materials which can be improved for next semester? You can also specify what we did right so it doesn't get changed.

---

[1]Although the manual numbers it as 1, the two questions asked are distinct

APPENDIX

Module descriptions are an important part of the reports in ECE 385, and since this is the first significant FPGA lab, a brief example of how to write a module description is shown below.

Let's say you needed two 16 bit registers to store operands A and B in an adder that computes the sum of A and B. Here is example code of a 16 bit register with asynchronous reset and synchronous load that can be used for that purpose.

```
module reg16
(input [15:0] Din,
input logic Clk, Load, Reset,
output logic [15:0] Dout);

always_ff @(posedge Clk or posedge Reset)
        begin
                if(Reset)
                        Dout <= 16'h0000;
                else if(Load)
                        Dout <= Din; //If load=1, perform parallel load on clock edge
        end

endmodule
```

And here is how a section of the report would describe it:

Module: reg16.sv
Inputs: [15:0] Din, Clk, Load, Reset
Outputs: [15:0] Dout
Description: This is a positive-edge triggered 16-bit register with asynchronous reset and synchronous load.
Purpose: This module is used to create the registers that store operands A and B in the adder circuit.

Simple modules can have a description and purpose that are just a sentence long each, but more complicated modules require more detailed descriptions.