ECE 385 Lab 4 Report Outline

- ❑ **Introduction**
  - ❑ Summarize the high level function performed by the serial logic processor and the three adders
- ❑ **Part 1 - Serial Logic Processor**
  - ❑ Include a block diagram can be adapted from lab manual or the top level schematic generated by the RTL viewer. *Please only include the top level design if using the RTL viewer*.
  - ❑ Include a short description should include what was done with the provided code to extend it from 4 bits to 8 bits
  - ❑ Include a simulation of the processor that has notes that give information such as what operation is being performed, where the result was stored, etc.
- ❑ **Part 2 - Adders**
  - ❑ Ripple Carry Adder
    - ❑ Written description of the architecture of the adder
    - ❑ Block diagram.
  - ❑ Carry Lookahead Adder
    - ❑ Written description of the architecture of the adder
      - ❑ Describe how the P and G logic are used
      - ❑ Describe how you partitioned the adder (Did you chain together 4 4-bit CLA blocks to make a 16-bit CLA block?)
    - ❑ Block diagram
      - ❑ Block diagram inside a single CLA (usually 4-bits)
      - ❑ Block diagram of how each CLA was chained together
  - ❑ Carry Select Adder
    - ❑ Written description of the architecture of the adder
      - ❑ Describe at a high level how the CSA speculatively computes multiple sums in parallel and rapidly chooses the correct one later.
    - ❑ Block Diagram of the whole CSA circuit containing adders, multiplexers, and glue logic.
  - ❑ Written Description of all .sv Modules
    - ❑ List all modules used in a format shown in the appendix of this document
  - ❑ Describe at a high level the area, complexity, and performance tradeoffs between the adders.
  - ❑ Document the performance of each adder by creating a graph as specified in Prelab part C (page 4.6 in the manual).
- ❑ **Answers to the 2 Postlab Questions**
  - ❑ Found on page 4.9 of the lab manual
- ❑ **Conclusion**
  - ❑ Describe any bugs and countermeasures taken during this lab.

- ❏ 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.
- ❏ Any additional summary you want to include

Notes: The state diagram asked for in the Lab 4 part of the lab manual (page 4.10) is no longer required, as the top level design for the adder circuit is now given to students. This requirement can be ignored when writing the report.

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.