Name: _____ ID# _____

Date Submitted: _____ Time Submitted _____

CSE 3341 Digital Logic Design II

CSE 5357 Advanced Digital Logic Design

Fall Semester 2024

Lab 5 – Verilog Modular Exponent Subtractor

Due Date – October 18, 2024 (11:59 PM)

Submit on Canvas Assignments
(Note new submission requirements below)

Note – Late submissions will not be accepted!

Submission Requirements

- Submit the entire Quartus project in a zipped folder to ensure all files are included for execution.
- Name the zip file as `netid_assignment_x` where `netid` is your university network ID, and `x` corresponds to the assignment number (e.g., `jdx1234_assignment_1`).
- Make sure the project files can be run seamlessly on another machine without any missing dependencies.

Contact the TA for any clarification or help.

CSE 3341/5357 **LAB ASSIGNMENT 5** **FALL 2024**

VERILOG MODULAR EXPONENT SUBTRACTOR

(100 POINTS)

ASSIGNMENT

Your assignment is to design a *paramerterized Verilog modular exponent subtractor* like those found in floating-point adders and subtractors as shown in Figure 1. The block diagram of an eight-bit modular exponent subtractor is shown in Figure 2.

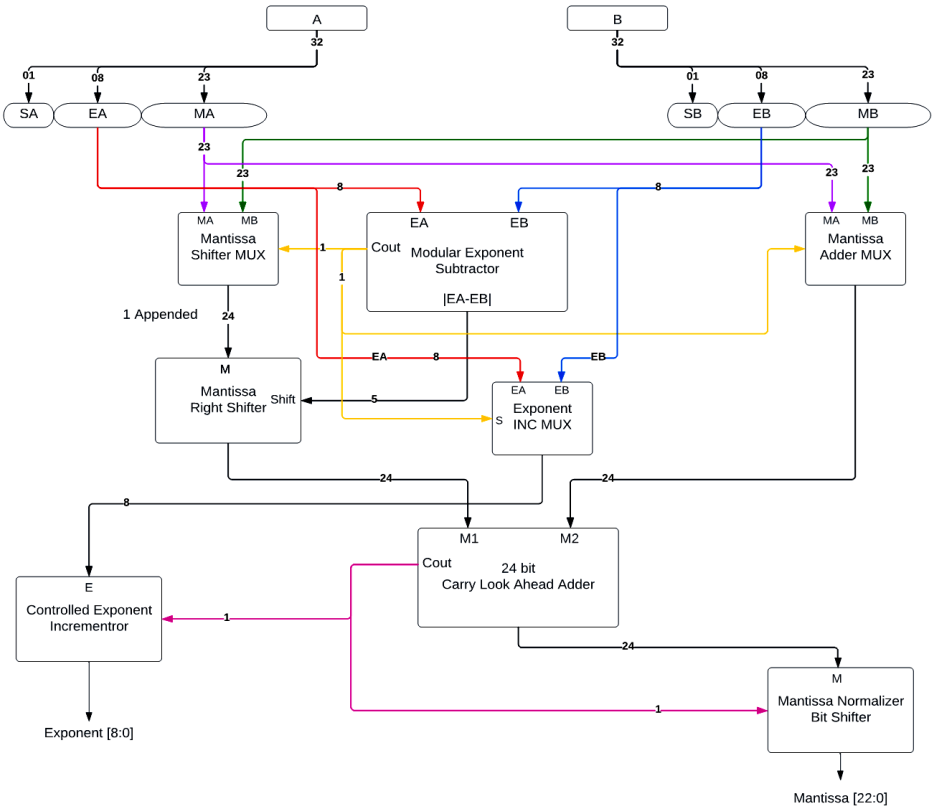

Figure 1. 32-bit Floating-Point Adder Architecture

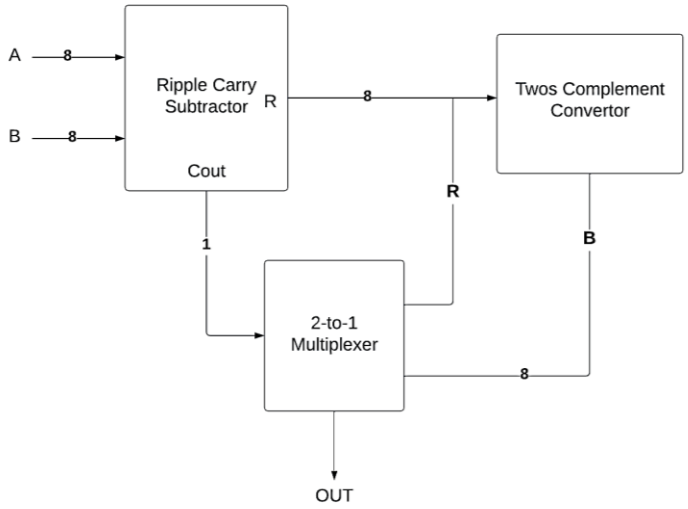

Figure 2. Eight-Bit Modular Exponent Subtractor

PURPOSE/OUTCOMES

To give you experience writing Verilog modules and instantiating these modules to realize more complex designs. In this lab you will implement a modular exponent subtractor. You will construct a Ripple Carry Subtractor to subtract two exponents and compute the difference between the two. Then, you will construct a component to compute the twos complement of the difference. You will then use a two-to-one multiplexer to choose the positive of these two values which will be the absolute difference which will then be used to perform floating point arithmetic in future labs.

After completing this lab, you will have demonstrated an ability to design eight-bit ripple carry subtractor, eight-bit twos complement to sign magnitude convertor, to write Verilog models of adders and subtractors, to capture and verify your designs using Model-Sim Altera on Quartus Prime, and to realize and test your designs on a DE10-Lite.

DESIGN REQUIREMENTS

1. Use hierarchical design by writing a SystemVerilog top-module that includes instantiations of the components shown in Figure 2.
2. Use hierarchical design to realize each component from Figure 2. Your final design should have a hierarchy the same as or similar to that shown in Figure 3.
3. Create an instantiation template of your Modular Exponent Subtractor to use in the floating-point adder-subtractor design later in the semester.

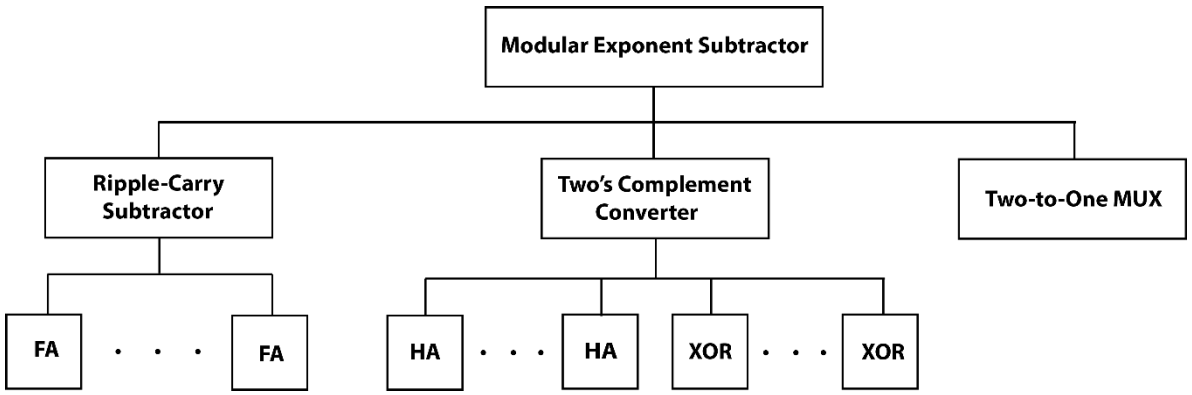

Figure 3. Eight-Bit Modular Exponent Subtractor Hierarchy Diagram

DESIGN VERIFICATION

1. Simulate your design to verify its correctness. Use the following values of A and B for your simulation inputs.

(a) 01010101 – 10101010

(b) 01111111 – 00000001

(c) 01111111 – 11111111

(d) 01100110 – 11011101

(e) 11111111 – 00000000

(f) 00000000 – 11111111

(g) 10101010 – 01010101

(h) 11111111 – 11111111

2. Include screen shots of your simulation waveform in your report.

3. Record the simulation results in the table below for your report.

A	B	R = |A – B|	Cout
01010101	10101010		
01111111	00000001		
01111111	11111111		
01100110	11011101		
11111111	00000000		
00000000	11111111		
10101010	01010101		
11111111	11111111		

DE10-Lite IMPLEMENTATION (*N=5*)

1. Implement a 5-bit version of your design on the DE-10 Lite using the following inputs/outputs using pin assignments of your choice.
   Inputs: A4, A3, A2, A1, A0, B4, B3, B2, B1, B0
   Outputs: R4, R3, R2, R1, R0, Cout

2. Include a table of your assignments in your report.

3. Program the DE10-Lite with your design.

4. Test your design with the following input pairs (take photos for your report)
   (a) 10101 – 01010
   (b) 01010 – 10101
   (c) 11111 – 00001
   (d) 00001 – 11111
   (e) 01101 – 00000
   (f) 10110 – 10110

5. Check your understanding
   (a) Explain how the circuit in Figure 2 computes R = |A – B|.
   (b) Explain the meaning of Cout = 0. Cout = 1.

REPORT REQUIREMENTS

1. Cover sheet
2. SystemVerilog code for the top module and all lower-level modules
3. Simulation waveform output and summary of results
4. DE10-Lite test results with photos