ECE 2330 DIGITAL LOGIC DESIGN
UNIVERSITY OF VIRGINIA
STUDIO ASSIGNMENT ADDER DESIGN: PART 1

Standard honor statement.
This assignment is quite involved and will take a long time. Procrastinate at your own peril.

OBJECTIVE

This studio assignment will give you the opportunity to design and compare several alternative design approaches for a combinational data processing element. This is part 1 of a 2-part studio design problem.

PROBLEM DESCRIPTION

You are to design a binary adder which can add two signed 16-bit binary numbers in 2's complement representation. You will implement and evaluate several approaches using *Logisim* in a 2-part studio assignment. You will then extend this adder to allow it to perform subtraction as well. Use sections 4.3 and 6.4 in the text as a reference for this assignment. Read these sections very carefully so that you understand what you are building.

DESIGN PROCEDURE: PART 1

Be sure that you understand the full set of instructions before you begin. Your ability to answer the questions at the end may determine the design approach that you use for the different parts of this assignment.

1. Design a 2-level combinational network that adds two *n*-bit numbers. An *n* bit adder has $2n + 1$ inputs (the +1 is for the carry-in bit) and n+1 outputs (including a carry-out bit). The value of *n* can be 2,4,8 or 16. You choose which value of *n* is the most appropriate for your design. The choice of *n* is totally yours as the designer. **Do not use the built-in Logisim adder.** You need to use AND and OR gates to design this.
2. Connect an appropriate number of your *n*-bit adders to produce a 16-bit adder. Use the subcircuit capability of Logisim to build your 16-bit adder from your *n*-bit adders.
3. Using your 16-bit adder as a subcircuit, construct a 16-bit adder/subtractor. An input X=1 signals that the addends should be subtracted (sum = A - B). When X=0, add the addends (sum = A + B).
4. Add an overflow detector to the adder/subtractor that you built in step 3.
5. For each of your designs (from steps 1-4) calculate the **delay**, that is, the length (in gates) of the longest path from any input to any output.

PRE-WORK

Your designs should be complete before you come to the studio. Every input and output must be clearly labeled.

GRADING

- 5 points for the design and implementation of an n-bit adder (step 1) including explanation
- 1 point for building a 16-bit adder from the *n*-bit adder
- 1 point for the construction of the adder/subtractor from the 16-bit adder
- 1 point for overflow detection
- 2 points for delay discussion with TA
- 1 point for professionalism