4-bit Adders

Prerequisites

You need to be familiar with the construction of circuits using Logicworks. You should know how to add binary numbers together and how to use the 2's complement system to subtract numbers. You need to know the limitations of negative and positive numbers in the 2's complement system.

Objectives

Your objectives include the building of simulation circuitry to add and subtract binary numbers. You are expected to be able to demonstrate to your TA that your circuit works correctly given numbers within the range of the design.

Theory

The theory of the numbers to be used should be familiar to you. However, all the numbers needed for this laboratory exercise are 4-bit strings. In the case of negative numbers the range will be from decimal -8 to +7 or binary 1000 to 0111.

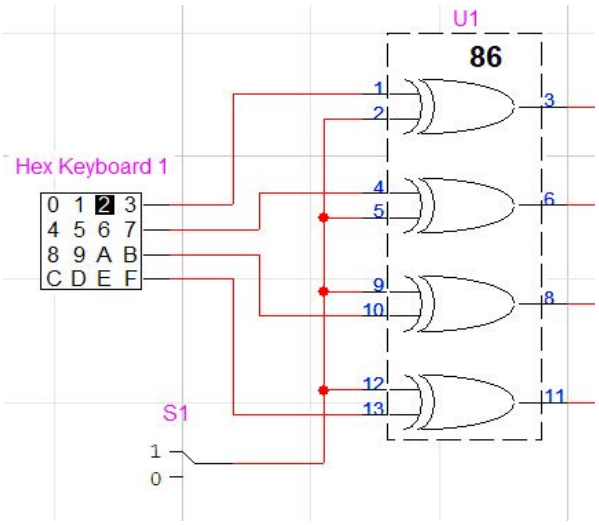
If you are adding with positive and negative numbers you need to observe the following.

- Positive numbers all begin with a 0, negative number begin with a 1
- Using 2's complement arithmetic you do not change a positive number
- You do, however, need to create the 2's complement version for a negative number. The simplest method is to invert all the bits and add 1.
- Once the addition has taken place you accept the result as positive if it starts with a 0
- If it starts with a 1 you need to convert it to a magnitude which will be its negative value.

Procedure

- Using the 4-bit adders in Logicworks, build a circuit that will add three 4-bit positive binary numbers. Test it thoroughly though not exhaustively! Document your results and show them to your TA.
- Using the 4-bit adders in Logicworks, design a circuit to add two 4-bit numbers. Test your circuit by adding positive to positive numbers, positive to negative numbers, negative to negative numbers, and cases where an overflow occurs.

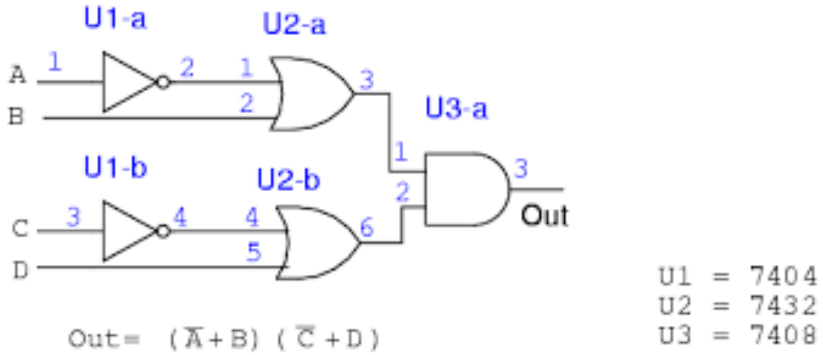
Hint: To invert a bit you can use an XOR gate. Below, the inputs are inverted if the toggle T is a 1 but not if T = 0.



- Create a simple circuit to detect an overflow condition. Demonstrate to your TA that the circuit works in all the above conditions.
- Your circuits should have hex keyboards for input and some hex displays for outputs. Other node values should be indicated by probes.

Deliverables

Your report should contain all your results with explanations if necessary. Your circuit printout should show a tidy circuit with clear labeling. The enclosed circuits show the appropriate method of annotation of logic diagrams. Tabular test results should be included with comments as appropriate.



Note the chip list at the bottom right (above)

Question

Since you cannot, in practice, test all digital circuits exhaustively, how would decide which tests to make?