SOFTWARE 2:
4-BIT ADDERS

STUDENT RUBRIC

DEMO RUBRIC

This lab will test your understanding of binary addition and 2's Complement. The first part of the lab is effectively a "warm-up" and should be checked by TAs during the lab period, as it serves as a precursor to part 2. For part 2, from two hexadecimal keyboards and an inversion switch, students are tasked with implementing a 4-bit adder, where two four-bit inputs are converted to 2s complement, then added together, and returned to four unsigned bits AND the sign AND the overflow check. The real difficulty in this lab is figuring out how to program the overflow check to catch all relevant states.

The following overflow states need to be accounted for…

1) Positive + positive greater than 7.
2) Negative + negative less than -8.

…and the following are "optional".

3) Overflow should NOT display if one of the inputs is negative 0.
4) A check that identifies whether the hex input is too high (cannot exceed 7 for a positive and 8 for a negative).

Completion Requirements:

- ✓ Complete part A, where three 4-bit positive numbers are added.
- ✓ Part B circuit has 2 hex keyboard inputs with associated "sign switches".
- ✓ There are three displays: a hex display for the unsigned value, a binary probe displaying the sign, and a binary probe displaying whether or not there is overflow (1 = yes).
- ✓ Overflow is correctly detected based on the requirements above.

REPORT RUBRIC

Unique to this report is the need to learn how to label circuits and export the resultant display from LogicWorks. This is not present in other labs, but it is important to learn because it forces you to understand pin-out diagrams. In the lab instructions, there is a picture with four NAND gates in a column that say (1, 2, 3), then (4, 5, 6), then (13, 12, 11). As a student, I looked at this and said "what?". Working through the process of labelling the chips should allow you to come to a greater understanding of ICs than before.

We recommend you use the actual ICs in their LogicWorks design. By default, there is no 74_32 but I have designed one for lab use, which you can download from the same place you got this rubric.

Scoring (out of 3 points):

- ✓ **[0.8 points]** Theory:
  - [0.3] What is 2's Complement? How does it work and why do we use it instead of signed binary for digital logic applications?
  - [0.3] Why don't you *actually* have overflow when adding together numbers like negative 3 and positive 5 in 2's Complement, even though the hardware says you do?

- ✓ **[1.6 points]** Deliverables:
  - A section including the following tests with weight, sign, and overflow:
    - [0.1] Positive zero plus negative zero
    - [0.15] Three examples of a positive plus a positive, one of which causes an overflow
    - [0.15] Three examples of a negative plus a negative, one of which causes an overflow
    - [0.1] Two examples of any positive number and any negative number
  - Fully labelled printout of the circuit including the following:
    - [0.6] All switches, displays, and ICs labelled
    - [0.5] The pins on all gates are labelled correctly. Namely, if there are four XORs in a column, pins 1-6 and 13-8 should be indicated for that set.

- ✓ **[0.4 points]** Discussion section. Should conform to standard lab report guidelines.
- ✓ **[0.4 points]** Question 1: Refer to the instructions for details.