Computer Organization and Structure

Homework #3
Due: 2005/11/22

1. Convert 512_{ten}, $-1{,}023_{ten}$, and $-4{,}000{,}000_{ten}$ into 32-bit two's complement binary numbers, respectively, and convert the following two's complement binary numbers to be decimal numbers:

   a. $1111\ 1111\ 1111\ 1111\ 1111\ 1110\ 0000\ 1100_{two}$;
   b. $1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111_{two}$;
   c. $0111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111_{two}$.

2. Find the shortest sequence of MIPS instructions to determine the absolute value of a two's complement integer. Convert this instruction (accepted by the MIPS assembler):

   ```
   abs      $t2, $t3
   ```

   This instruction means that register `$t2` has a copy of register `$t3` if register `$t3` is positive, and the two's complement of register `$t3` if `$t3` is negative. (Hint: It can be done with three instructions.)

3. The ALU supported set on less than (`slt`) using just the sign bit of the adder. Let's try a set on less than operation using the values -7_{ten} and 6_{ten}. To make it simpler to follow the example, let's limit the binary representations to 4 bits: 1001_{two} and 0110_{two}.

   $1001_{two} - 0110_{two} = 1001_{two} + 1010_{two} = 0011_{two}$

   This result would suggest that $-7_{ten} > 6_{ten}$, which is clearly wrong. Hence we must factor in overflow in the decision. Modify the 1-bit ALU in the following figures to handle `slt` correctly.

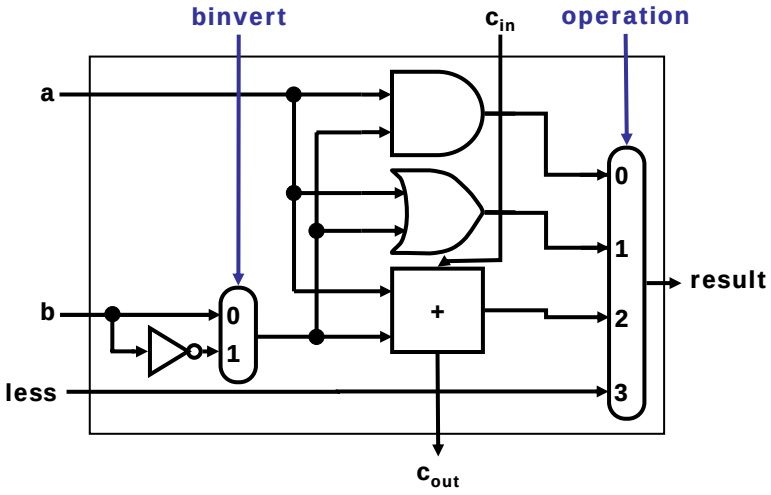

Figure 1: A 1-bit ALU that performs AND, OR, and addition on a and b or b'.

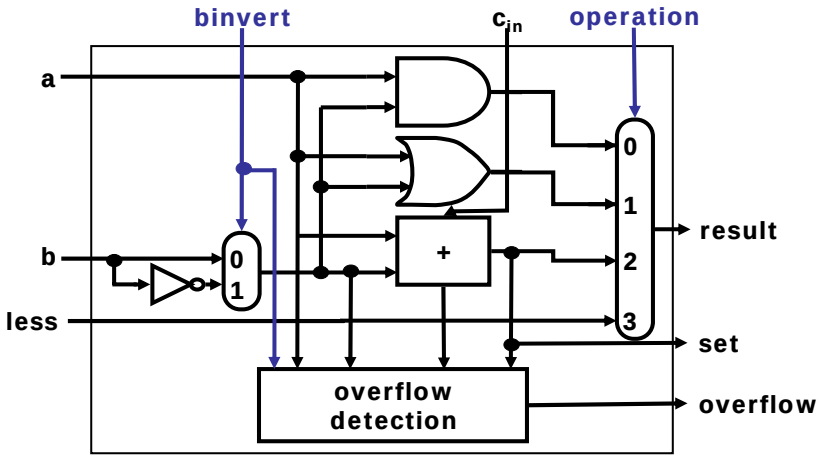

Figure 2: A 1-bit ALU for the most significant bit.

4. Add 2.85_{ten} x 10^3 to 9.84_{ten} x 10^4 and add 3.63_{ten} x 10^4 to 6.87_{ten} x 10^3, respectively, assuming that you have only three significant digits, first with guard and round digits and then without them.

5. Show the IEEE 754 binary representation for the floating-point number 10_{ten}, 10.5_{ten}, 0.1_{ten}, and -2/3, respectively.