CS 2810 – H.W. #17

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 > 6, which is clearly wrong. Hence, we must factor in overflow in the decision. Modify the 1-bit ALU in Figure C.5.10 on page C-33 to handle `slt` correctly. Make your changes on a photocopy of this figure to save time.

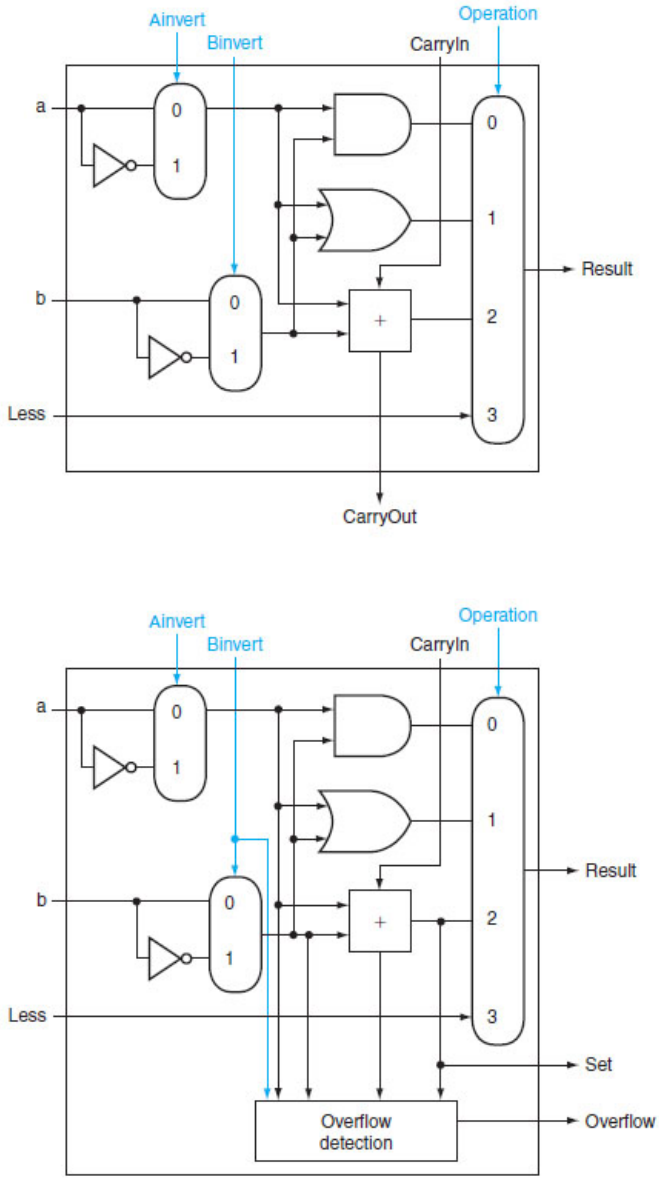

FIGURE C.5.10 (Top) A 1-bit ALU that performs AND, OR, and addition on a and b or $\bar{b}$, and (bottom) a 1-bit ALU for the most significant bit. The top drawing includes a direct input that is connected to perform the set on less than operation (see Figure C.5.11); the bottom has a direct output from the adder for the less than comparison called Set. (See Exercise C.24 at the end of this Appendix to see how to calculate overflow with fewer inputs.)