1. The figure below demonstrates the design of a 1-bit ALU. Construct a 4-bit ALU that can also perform the set-less-than operation.

2. P&H (3.14) Calculate the time necessary to perform a multiply using the approach given in Figures 3.4 and 3.5 (refer to slide P13 of the wk3-1 lecture notes) if an integer is 8 bits wide and each step of the operation takes 4 time units. Assume that in step 1a an addition is always performed – either the multiplicand will be added, or a 0 will be. Also assume that the registers have already been initialized (you are just counting how long it takes to do the multiplication loop itself). If this is being done in hardware, the shifts of the multiplicand and the multiplier can be done simultaneously. If this is being done in software, they will have to be done one after the other. Solve for each case.

3. P&H (3.17). As discussed in the textbook, one possible performance enhancement is to do a shift and add instead of an actual multiplication. Since 9*6, for example, can be written (2*2*2+1)*6, we can calculate 9*6 by shifting 6 to the left 3 times and then adding 6 to that result. Show the best way to calculate 0x33*0x55 using shifts and adds/subtracts. Assume both input are 8-bit unsigned integers.

4. P&H (3.23). Write down the binary representation of the decimal number 63.25 assuming the IEEE 754 single precision format.

5. P&H (3.27). IEEE 754-2008 contains a half precision that is only 16 bits wide. The leftmost bit is still the sign bit, the exponent is 5 bits wide and has a bias of 15, and the mantissa is 10 bits long. A hidden 1 is assumed. Write down the bit pattern to represent -1.5625*10^{-1} assuming the half precision format.

6. P&H (3.37). Assume the half precision format is used. Does (3.41796875*10^{-3} * 6.34765625*10^{-3} ) * 1.05625 * 10^{2} = 3.41796875*10^{-3} *(6.34765625*10^{-3} * 1.05625 * 10^{2} )?