Department of Computer Science
National Tsing Hua University

CS4100 Computer Architecture, Spring 2019

Homework 3
Deadline: 2019/04/25 10:10 AM

1. The ALU design presented in the class supports set-on-less-than (slt) using just the sign bit of the adder. Unfortunately, it ignores the overflow case. Consider the set-on-less-than operation assuming A = -7_{10} and B = 6_{10}, and both of them are represented by four bits
   (a) Use the ALU design in Fig. A.5.12 to show that it outputs a wrong result for the above case.
   (b) Modify the 1-bit ALU in Fig. A.5.10.b to handle slt correctly.(You should only use mux, xor or xnor gates to finish this job)

2. Suppose the 32-bit pattern, 0000000 11000 01001 001 01100 1100011, is stored in memory starting from location $5A470B20_{16}$.
   (a) What RISC-V instruction will be executed if PC = $5A470B20_{16}$? Show your derivation.
   (b) What value (in decimal) is loaded into the floating-point register f0 if the following RISC-V instruction is executed, where x20 contains $5A470B20_{16}$. Show your derivation.

```
flw     f0, 0(x20)
```

3. IEEE 754-2008 contains a half precision that is only 16 bits wide. The leftmost bit is 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 present the following decimal numbers. Please show the calculation procedure.
   (a) -0.39
   (b) 12.3456

4. Suppose in IEEE 754-2008 a half-precision number representation with a 00000 in the exponent field and a non-zero mantissa indicates a denormalized number: $(-1)^S \times (0 + Fraction) \times 2^{-14}$.
   (a) What is the binary representation of the smallest "normalized" positive number, denoted *a0*?
   (b) What is the binary representation of the largest "denormalized" number, denoted *a1*? How about the *second* largest "denormalized" number, denoted *a2*?
   (c) Find the differences between *a0* and *a1*, and between *a1* and *a2*.
   (d) Repeat (a), (b) and (c) if a denormalized number is given by $(-1)^S \times (0 + Fraction) \times 2^{-15}$.

5. Please write down the steps of 00001101 ÷ 0011 according to version 2 of the divide algorithm. Show the contents of the two registers, Divisor and Remainder, in each step.