COMP 230: Computer Architecture and Organization

Homework 3

Assigned: September 17, 2017 *Due: September 25, 2017*

Complete this assignment on a separate sheet of paper.

1. Exercise B.36 from the text.

2. Exercise 2.7.

3. Convert each of the following MIPS assembly instructions to both machine code binary and hexadecimal.

   (a) `sw $t0 8($s0)`

   (b) `sub $s4,$s1,$t2`

   (c) `addi $t5,$t8,100`

4. Convert each instruction from hexadecimal to machine code binary and to MIPS assembly.

   (a) `000b 8180`$_{16}$

   (b) `2319 ff9c`$_{16}$

5. Convert each of the following C statements to MIPS assembly statements. Assume each lowercase variable `f, g, h, i,` and `j` are 32-bit integers as declared in a C or C++ program and are stored in registers `$s0` through `$s4`, respectively. Arrays are denoted with uppercase letters, with the base address of `A` in register `$s5`, the base address of `B` in `$s6`, etc. You are not allowed to use any `mul` or `div` instructions.

   (a) `f = g + (h - 5);`

   (b) `B[8] = A[i - j] + B[i];`

   (c) `i = (f + j) / 8;`

   (d) `g = C[0] << 4; //` (In C, `x << y` shifts the bits of `x` to the left by `y` positions)

6. Exercise 2.18.

7. Exercise 2.26, parts 1 (2.26.1) and 2 (2.26.2) only. Note that part 2 refers to "each of the loops above," but there is actually just one loop. Also note that C++'s brace-initialization syntax is not valid in C.

8. Implement the C code from exercise 2.31 in MIPS assembly.

9. Write an iterative (instead of recursive) version of the C code in exercise 2.31 and write the MIPS assembly for it.