Homework Assignment #2 – MIPS Instructions

CDA 3100, Computer Organization I

Submission: A hard copy required.

Problem 1 (30 points) Exercise 2.4.1(p. 182).

The following problems deal with translating from C to MIPS. Assume that the variables `f,g,h` are assigned to registers `$s0, $s1, $s2`, respectively. Assume that the base address of the arrays `A` and `B` are registers `$s6` and `$s7`, respectively.

a. `f=g+h+B[4];`

b. `f=g-A[B[4]];`

2.4.1. For the C statements above, what is the corresponding MIPS assembly code?

Problem 2 (10 points) Exercise 2.16.1 (p. 194).

For these problems, there are various binary values for register `$t0`. Given the value for `$t0`, you will be asked to evaluate the outcome of different branches.

a. $1010\ 1101\ 0001\ 0000\ 0000\ 0000\ \ 0000\ 0010_{two}$

b. $1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111_{two}$

2.16.1. Suppose that the register `$t0` contains a value from the above and `$t1` has the value

$0011\ 1111\ 1111\ 1000\ 0000\ 0000\ 0000\ 0000_{two}$

What is the value of `$t2` after the following instructions?

```
      slt $t2, $t0, $t1
      beq $t2, $zero, ELSE
      j DONE
ELSE: addi $t2, $zero, 2
DONE:
```

Problem 3 (40 points) Exercise 2.18.2. (p. 196).

For these problems, you are given some C code. You will be asked to evaluate these C code in MIPS assembly code.

```
a. for(i=0;i<10;i++)
     a+=b;

b. while (a<10){
     D[a] = b+a;
     a+=1;
   }
```

2.18.2. For the code above, translate the C code to MIPS assembly code. Use a minimum number of instructions (I won't take off points, but you should try to complete this in a few instructions as possible to prepare for the midterm). Assume that the value of `a, b, i` are in registers `$s0, $s1, $t0` respectively. Also, assume that register `$s2` holds the base address of the array `D`.

Problem 4 (20 points) Encode the following MIPS instructions. For each instruction, you should identify the format type (R, I, or J format) and the decimal values of each field and then give the hexadecimal representation. (You may find the Appendix B helpful (pp. B-49 – B-80), where the encoding of MIPS instructions is described in detail.)

```
1)       addi  $s1, $s3, 3   # $s1 is register 17, $s3 is register 19
2)       sw    $s1,  12($sp) # $s1 is register 17, $sp is register 29
3)       add   $t2, $s3, $s4 # $t2 is register 10, $s3 is register 19, $s4
   is register 20
```