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,i,` and j are assigned to registers `$s0, $s1, $s2, $s3`, and `$s4`, 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?

Solution:

```
2.4.1

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

add $s0, $s1, $s2
lw $t0, 16($s7)
add $s0, $s0, $t0

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

lw $t0, 16($s7)
sll $t0, $t0, 2
add $t0, $t0, $s6
lw $t0, 0($t0)
sub $s0, $s1, $t0
```

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:
```

Solution:

```
      2.16.1
      a. 1
      b. 1
```

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. Assume that the value of `a,b,i,j` are in registers `$s0, $s1, $t0, $t1,` respectively. Also, assume that register `$s2` holds the base address of the array `D`.

Solution:

```
      2.18.2
      a. for (i=0;i<10;i++)
            a+=b;

      ori $t0, $0, 0
loop: add $s0, $s0, $s1
      addi $t0, $t0, 1
      slti $t2, $t0, 10
      bne $t2, $0, loop


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

      sll $t2, $s0, 2
      add $t2, $t2, $s2
loop: slti $t3, $s0, 10
      beq $t3, $0, done
      add $t3, $s0, $s1
      sw $t3, 0($t2)
      addi $s0, $s0, 1
      addi $t2, $t2, 4
      j loop
one:
```

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 and $s3 is register 19
2)       sw    $s1,  12($sp) # $sp is register 29 (stack pointer)
3)       add   $t2, $s3, $s4 # $t2 is register 10, $s4 is register 20
```

Solution:

1. addi $s1, $s3, 3
   I-format instr: op = 8, rs = 19, rt = 17, imm = 3.
   0010 0010 0111 0001 0000 0000 0000 0011
   Encoding: 0x22710003
2. sw $s1, 12($sp)
   I-format: op = 43, rs = 29, rt = 17, imm = 12
   1010 1111 1011 0001 12
   Encoding: 0xAFB1000C
3. add $t2, $s3, $s4
   R-format, op = 0, rs = 19, rt = 20, rd = 10, shamt = 0, funct = 32
   Encoding: 0x02745020