2023-1 Computer Architecture Homework #2

Due: 4/18 (Tue) 11:59 p.m.

2.4 [10] <§§2.2, 2.3> For the MIPS assembly instructions above, what is the corresponding C statement? 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 in registers $s6 and $s7, respectively.

```
sll  $t0, $s0, 2   # $t0 = f * 4
add  $t0, $s6, $t0 # $t0 = &A[f]
sll  $t1, $s1, 2   # $t1 = g * 4
add  $t1, $s7, $t1 # $t1 = &B[g]
lw   $s0, 0($t0)   # f = A[f]
addi $t2, $t0, 8
lw   $t0, 0($t2)
add  $t0, $t0, $s0
sw   $t0, 0($t1)
```

2.10 Assume that registers $s0 and $s1 hold the values 0x80000000 and 0xD0000000, respectively.

2.10.1 [5] <§2.4> What is the value of $t0 for the following assembly code?

```
add $t0, $s0, $s1
```

2.10.2 [2.5] <§2.4> Is the result in $t0 the desired result, or has there been overflow?

2.10.3 [5] <§2.4> What is the value of $t0 for the following assembly code?

```
sub $t0, $s0, $s1
```

2.10.4 [2.5] <§2.4>Is the result in $t0 the desired result, or has there been overflow?

2.12 The binary representation of instructions `add, sub, addi, lw, sw` are 32, 34, 8, 35, 43 respectively.

2.12.1 [5] <§§2.4, 2.5> Provide the type and assembly language instruction for the following binary value.

0000 0010 0001 0000 1000 1000 0010 0000_{two}.

2.12.2 [5] <§§2.4, 2.5> Provide the type and hexadecimal representation of following instruction.

```
sw $t1, 32($t2)
```

2.12.3 [5] <§2.5> Provide the type, assembly language instruction, and binary representation of instruction described by the following MIPS fields.

op=0, rs=9, rt=10, rd=8, shamt=0, funct=34

2.17 [10] <§2.6>Assume the following register contents.

$t0=0xBBBBBBBB, $t1=0x12345678

For the register values shown above, what is the value of $t2 for the following sequence of instructions?

```
sll $t2, $t0, 4
or $t2, $t2, $t1
```

2.21 [5] <§2.7> Assume $t0 holds the value 0x10100000. What is the value of $t2 after the following instructions?

```
slt $t2, $0, $t0
bne $t2, $0, ELSE
j   DONE
ELSE: addi $t2, $t2, 2
DONE:
```

2.22 Suppose the *program counter* (PC) is set to 0x10000000.

2.22.1 [3] <§2.10> What range of addresses can be reached using the MIPS *jump-and-link* (jal) instruction? Use hexadecimal representation.

2.22.2 [7] <§2.10> What range of addresses can be reached using the MIPS *branch if equal* (beq) instruction? Use hexadecimal representation.

2.24 Consider the following MIPS loop:

```
LOOP: slt  $t2, $0, $t1
      beq  $t2, $0, DONE
      subi $t1, $t1, 1
      addi $s2, $s2, 3
      j    LOOP
DONE:
```

2.24.1 [5] <§2.7> For the loops written in MIPS assembly above, assume that the register $t1 is initialized to the value N. How many MIPS instructions are executed?

2.24.2 [10] <§2.7> For each of the loops above, write the equivalent C code routine. Assume that the registers $s1, $s2, $t1, and $t2 are integers A, B, i, and temp, respectively.