COMP 2421 Computer Organization
Assignment 1

Firm Deadline: 11:59 pm, Feb. 25 (Sunday), 2024.
Note: please submit a PDF or Word file with file name: "Name_ID.pdf".

1 Questions with Short Answers [10 pts]

For each of the question, you need to write down the detailed steps.

1. Assuming 12-bit 2's complement representation, what is the decimal value of the hexadecimal number FD2? [**2 pts**]

2. What is the decimal value of the following floating point number represented in binary form (8 bits for biased exp and 23 bits for significand)? [**2 pts**]

   1 1000 1001 111 1100 1101 0000 0000 0000

3. Suppose that we use **4 bits** to represent integers in 2's complement form. Consider the following two cases of binary addition. First, calculate the result. Second, decide whether there is overflow and explain why. [**2 pts**]

   1) Case 1: $0111 + 1111$
   2) Case 2: $1110 + 1000$

4. Examine the following program fragment:

   addiu $t1,$0,-13
   addiu $t2,$0,23
   the third instruction

   What is the third instruction to set $t3 to 1 [**2 pts**]? Requirement: using 'sltu'.

5. Complete the following program to push the two values stored in $s1 and $s2 into stack (note: $s1 is pushed first) [**2 pts**]:
   addi $sp, $sp, __
   sw $s1, __($sp)
   sw $s2, __($sp)

2 MIPS: Translate Pseudo-instructions [10 pts]

For each of the following Pseduo-instruction with comment, translate it into actual MIPS instructions as specified.
(1) [**3 pts**]

```
ror $t1, 7
#rotate right: rotate the bits in $t1 to the right by 7 positions.
#here, "rotate" means that the bits on the right side are filled into
#the vacated bits on the left (see Fig.1)
Requirement: use a sequence of three instructions: srl, sll, or,
             and another register.
```

Figure 1: Question 2-(1): rotate right.

(2) [**3 pts**]

```
multiply $t1, $t2, 31
#multuply $t2 with a constant 31 and store the result in $t1
#suppose there's no overflow
Requirement: use a sequence of two instructions: sll, sub.
```

(3) [**4 pts**]

```
lw $t4, 0x00010002($t1)
#load the word stored at memery address $t1 + 0x00010002 into register $t4,
#where $t1 stores the base address
Requirement: use a sequence of four instructions: lui, ori, add, lw,
             and another resigter
```

3 MIPS: Translate MIPS program into C program [12pts]

Read the following MIPS code segment and comments. Translate it into C code. Specifically, the registers `$s0, $s1, $s2, $s3` store signed integers x, y, z, w, respectively. Complete the C code using expressions of x, y, z, w. Do not care too much about the grammars of C. You will get full marks as long as the meanings are correct.

MIPS code segment:

```
bge $s0, $s1, L1   # go to L1 if $s0 >= $s1
bgt $s2, $s3, skip # go to skip if $s2 > $s3
L1:
bne $s0, $s2, skip # go to skip if $s0 != $s2
L2:                    # inner if statement
bne $s2, $s3, L3   # go to L3 if $s2 != $s3
addu $s0, $s1, $s2
j skip
L3:
addiu $s2, $s1, -2
skip:
```

C code you need to complete:

```
if (some condition 1  [5pts]){
if (some condition 2   [3pts])
{
some code 1   [1pt]
}else{
some code 2  [1pt]
}
}
```

4 MIPS: Understand MIPS Code [18 pts]

An array of integers `S` is defined in the following code. Try to understand the code and answer the following questions.

```
S: .word 14, -29, 18, 30, -12, 12, 106, -7

la $a0, S   # load address of S into $a0; suppose $a0 = 0x20060000
addi $a1, $a0, 28
move $v0, $a0  #move the value of $a0 into $v0
lw $v1, 0($v0)
move $t0, $a0
loop: addi $t0, $t0, 4
lw $t1, 0($t0)
ble $t1, $v1, skip # go to skip if $t1 <= $v1
move $v0, $t0
move $v1, $t1
skip: bne $t0, $a1, loop
```

(1) What flow-control statement does `ble $t1, $v1` implement? [**1 pts**]

(2) To show that you fully understand the function of this program, briefly explain the usage of the following 4 registers in the program. That is, what are these registers used for in the program. For example, for register `$a1`, it stores the address of the last element of array, indicating the end of array. [**4 pts**]

Registers: `$t0, $t1, $v0, $v1`

(3) Briefly explain the usage of the two instructions `move $v0, $t0; move $v1, $t1`. [**4 pts**]

(4) Briefly explain the usage of the instruction `bne $t0, $a1, loop`. [**2 pts**]

(5) Briefly explain the function of this program, suppose the desired outputs of the program are the contents of the registers `$v0` and `$v1`. [**3 pts**]

(6) Determine the contents of the registers `$v0` and `$v1` after executing the code. [**4 pts**]