

**COMP180: Computer Organization**  
**Spring 2001, Lecture Section 1, Dit-Yan Yeung**

**Quiz #2**  
7 March 2001

Name: \_\_\_\_\_ Student ID: \_\_\_\_\_ E-mail: \_\_\_\_\_

Answer all questions in the space provided. Time allowed: 20 minutes

1. For each of the following statements, write down 'true' if it is true and 'false' otherwise.

- (a) The address field in an I-format instruction always represents an address offset in bytes. Ans: \_\_\_\_\_
- (b) RISC computers can usually accommodate more registers than CISC computers. Ans: \_\_\_\_\_
- (c) The instruction `jal sort` saves the value of `$ra` to the program counter before jumping to the instruction labeled `sort`. Ans: \_\_\_\_\_
- (d) A jump instruction can never jump beyond a range of  $2^{26}$  words. Ans: \_\_\_\_\_
- (e) Each real MIPS assembly instruction corresponds to only a single MIPS machine instruction. Ans: \_\_\_\_\_
- (f) Each MIPS pseudoinstruction corresponds to multiple MIPS machine instructions. Ans: \_\_\_\_\_
- (g) The MIPS instruction `beq $s0, $s1, loop` does not necessarily lead to the execution of the instruction labeled `loop`. Ans: \_\_\_\_\_
- (h) The pseudodirect addressing mode computes the target address by finding the sum of a full 32-bit address and an address offset. Ans: \_\_\_\_\_
- (i) The two's complement scheme can represent more numbers than the one's complement scheme. Ans: \_\_\_\_\_
- (j) The MIPS instruction `addiu` may lead to overflow. Ans: \_\_\_\_\_

2. The absolute value function is defined as:

$$\text{abs}(x) = \begin{cases} x & \text{if } x \geq 0 \\ -x & \text{otherwise} \end{cases}$$

Give a sequence of real MIPS assembly language instructions which takes the value held in register `$t0`, applies the absolute value function to it, and then saves the result into register `$t1`.

3. Suppose the values of two registers `$s0` and `$s1` are `159e 5091` and `ec25 7a15`, respectively, in hexadecimal representation. Give the value of register `$t1` in hexadecimal representation after performing the MIPS instruction `add $t1, $s0, $s1`. Does this instruction lead to overflow?