期中.

Computer Organization & Assembly Language

Midterm Exam – 2008/11/12

Dept. of Engineering Science, National Cheng Kung University

1. Answer "True" or "False" to the following statements. (36%)

T (1) Memory space is a major constraint for executing computer programs in early years.

F (2) A single instruction in assembly language may map to one or more operations in machine codes.

T (3) In a processor, the control unit is responsible for decoding an instruction while the datapath unit is responsible for executing an instruction.

F (4) Moore's law indicates that the number of transistors in a processor grows linearly as time advances.

F (5) The "shamt" field is not used by the `add` instruction but is used by the `lw` instruction.

T (6) The `nor` instruction can help on performing the logical NOT operation.

F (7) For the stack in MIPS, a data element located at a higher address can be read before a data element located at a lower address is read.

T (8) The data stored in `$t0~$t9` is never preserved during procedure calls.

F (9) Writing codes in assembly language usually results in a better efficiency of developing time.

T (10) When representing a negative number, the MSB is used as the sign bit.

(11) When performing the binary multiplication operation for computers, it may not be necessary to add the Multiplicand to the Product in all iterations.

F (12) When performing the binary division operation for computers, it may not be necessary to subtract the Divisor from the Remainder in all iterations.

2. Spell out the full names of the following abbreviations. (6%)

(1) DLL Dynamically Linked Library

(2) JVM Java Virtual Machine

(3) ALU Arithmetic Logic Unit

3. What are the two most important parts in an object file for further linking process? (4%)

relocation information - symbol table

4. What kind of instructions is supported by PC-relative addressing? Explain in detail how PC-relative addressing works. (6%)

PC relative instruction ⇒ branching instructions

⇒ PC = register (PC+4) + branch address

5. The following two versions of assembly language codes perform the same task of clearing contents in an integer array. Note that `$a0` stores the base address of this array and `$a1` stores the array size whose value is 15.

A. array version

```
        move  $t0, $zero
Loop1:  sll   $t1, $t0, 2
        add   $t2, $a0, $t1
        sw    $zero, 0($t2)
        addi  $t0, $t0, 1
        slt   $t3, $t0, $a1
        bne   $t3, $zero, loop1
```

B. pointer version

```
        move  $t0, $a0
loop2:  sw    $zero, 0($t0)
        addi  $t0, $t0, 4
        sll   $t1, $a1, 2
        add   $t2, $a0, $t1
        slt   $t3, $t0, $t2
        bne   $t3, $zero, loop2
```

(1) Which instruction (in both versions) is a pseudoinstruction? What is the equivalent MIPS instruction for the pointer version? (6%)

(2) How many instructions would be executed in the array version and the pointer version, respectively? (6%)

(3) How would you modify the codes in the pointer version to reduce the required number of instructions for execution? Explain your answer briefly. Also, how many instructions would be executed then? (6%)

(4) If this program is for handling a character array, how would you modify the codes in the array version? (6%)

6. Add $-4.63_{ten} \times 10^4$ to $-5.96_{ten} \times 10^3$, assuming that you have only three significant digits, first with guard and round digits and then without them. (6%)

7. Show the IEEE 754 binary representation for the floating-point number -4.875_{ten} and $1/6_{ten}$ in single precision, respectively. (8%)

8. Explain why there is a trade-off between *precision* and *range* when representing a floating point number in computer arithmetic? (4%)

9. When using the IEEE 754 format, what is the meaning of *overflow* and *underflow*, respectively? (4%)

10. Explain the difference between the computer arithmetic and the paper-and-pencil arithmetic. (4%)