Computer Organization & Assembly Language

Midterm Exam – 2011/11/18

Dept. of Engineering Science, National Cheng Kung University

1. For the following set of variables, identify all of the subsets that can be used to calculate execution time. Each subset should be minimal; that is, it should not contain any variable that is not needed. Note that MIPS stands for "*million instructions per second*". (12%)

{CPI, clock rate, cycle time, MIPS, number of instructions in program, number of cycles in program}

{CPI, clock rate

2. You are the lead designer of a new processor. The processor design and compiler are complete, and now you must decide whether to produce the current design as it stands or spend additional time to improve it. You discuss this problem with your hardware engineering team and arrive at the following options:

a. *Leave the design as it stands*. Call this base machine *Mbase*. It has a clock rate of 500 MHz, and the following measurements have been made using a simulator:

Instruction class	CPI	Frequency
A	2	40%
B	3	25%
C	3	25%
D	5	10%

b. *Optimize the hardware*. The hardware team claims that it can improve the processor design to give it a clock rate of 600 MHz. Call this base machine *Mopt*. The following measurements have been made using a simulator:

Instruction class	CPI	Frequency
A	2	40%
B	2	25%
C	3	25%
D	4	10%

(A) What is the CPI for each machine? (4%)

(B) What are the MIPS ratings for each machine? (4%)

(C) How much faster is *Mopt* than *Mbase*? (4%)

On the other hand, the compiler team proposes to improve the compiler for the machine to further enhance the performance. Call this combination of the improved compiler and the base machine *Mcomp*. The instruction improvements from this enhanced compiler have been estimated as follows.

Instruction class	Percentage of instructions executed vs. base machine
A	90%
B	90%
C	85%
D	95%

For example, if the base machine executed 500 class A instructions, *Mcomp* would execute 0.9*500=450 class A instructions for the same program.

(D) What is the CPI for *Mcomp*? (4%)

(E) How much faster is *Mcomp* than *Mbase*? (4%)

(F) If both the hardware and compiler improvements are implemented, yielding machine

Mboth, how much faster is *Mboth* than *Mbase*? (4%)

(G) Given that the following time would be required to implement the optimizations:

Optimization	Time to implement
Hardware (*Mopt*)	6 months
Comiler (*Mcomp*)	6 months
Both (*Mboth*)	8 months

Recall that CPU performance improves by approximately 50% per year, or about 3.4% per month. Assuming that the base machine has performance equal to that of its competitors, which optimizations (if any) would you choose to implement? (6%)

3. Please use one or two sentences to answer/explain each of the following questions.

(A) When and why does *register preservation* has to be done? (8%)

(B) How *register preservation* works? (4%)

n<1

4. Given the following MIPS code and assume that \$t0 and \$t1 correspond to *i* and *j*, respectively. Also, assume that \$a0 initially contains *n*, a positive number.

```
begin:   addi    $t0, $zero, 0
         addi    $t1, $zero, 1
loop:    slt     $t2, $a0, $t1
         bne     $t2, $zero, finish
         add     $t0, $t0, $t1
         addi    $t1, $t1, 2
         j       loop
finish:  add     $v0, $t0, $zero
```

→ finish

(A) Show the code in C. (8%)

(B) What is the returned value when n=1, 2, ..., 5, respectively? (10%)

(C) Describe in one sentence what this code segment computes. (4%)

5. Given your understanding of PC-relative addressing, explain why an assembler might have problems directly implementing the branch instruction in the following code sequence: (6%)

```
here:    beq $t1, $t2, there
...
there:   add $t1, $t1, $t1
```

6. The following program tries to copy words from the address in register \$a0 to the address in register \$a1, counting the number of words copied in register \$v0.The program stops copying when it finds a word equal to 0. You do not have to preserve the contents of registers \$v1, \$a0, and \$a1. The terminating word should be copied but not counted.

```
loop:   lw      $v1, 0($a0)          # Read next word from source
        addi    $v0, $v0, 1          # Increment count words copied
        sw      $v1, 0($a1)          # Write to destination
        addi    $a0, $a0, 1          # Advance pointer to next source
        addi    $a1, $a1, 1          # Advance pointer to next dest
        bne     $v1, $zero, loop     # Loop if word copied != zero
```

There are multiple bugs in this MIPS program; please fix them. (12%)

7. Show the single MIPS instruction or minimal sequence of instructions for this C statement:

```
x[10] = x[11] + c;
```

Assume that *c* corresponds to register \$t0 and the base address of the integer array *x* is stored in \$s0. (6%)