22341 Computer Organization (Fall 2010)

First Exam Solution

الاسم: رقم التسجيل: رقم الشعبة:

==========

Instructions: Time **60** min. Closed books & notes. No calculators or mobile phones. No questions are allowed. Show your work clearly. Every problem is for 5 marks.

==========

Q1. Consider two different implementations of the same instruction set architecture, P1 and P2. Processor P1 runs on a clock rate of **1.5 GHz** and P2 runs on **2.0 GHz**. There are four classes of instructions, A, B, C, and D. The CPIs of each implementation are given in the following table.

	Class A	Class B	Class C	Class D
CPIs of P1	1	2	3	4
CPIs of P2	2	2	2	2
Frequency	10%	10%	50%	30%

Given a program with 10^6 instructions divided into the four classes according to the frequencies in the above table, which implementation is faster? And how much?

CPU Time $= IC * \sum_{i=1}^{n} CPI_i * Freq_i / f$

CPU Time $_{P1}$ $= 10^6 * (1*0.1 + 2*0.1 + 3*0.5 + 4*0.3) / 1.5*10^9$

$= 2.0 * 10^{-3}$ sec

CPU Time $_{P2}$ $= 10^6 * (2*0.1 + 2*0.1 + 2*0.5 + 2*0.3) / 2.0*10^9$

$= 1.0 * 10^{-3}$ sec

Processor P2 is ($2.0 * 10^{-3} / 1.0 * 10^{-3} = 2$) faster than Processor P1.

Q2. Assume that registers `$s0`, `$s1`, and `$s2` hold the initial values shown in the table below.

Register	Contents
$s0	0x00000001
$s1	0x00000003
$s2	0x100000F8

Given the following 6 MIPS instructions, specify the contents of the registers shown in the following table after executing these 6 instructions (use the hexadecimal numbering system).

```
sub  $t0, $s0, $s1
sw   $t0, 0($s2)
lbu  $t1, 2($s2)
or   $t2, $s1, $s2
sll  $t3, $s2, 3
slti $t4, $s0, 4
```

Register	Contents
$t0	**0xFFFFFFFE**
$t1	**0x000000FF**
$t2	**0x100000FB**
$t3	**0x800007C0**
$t4	**0x00000001**

Q3. Translate the C code shown below to MIPS assembly code. Use a minimum number of instructions. Assume that the values `A` and `B` are in registers `$s0` and `$s1`, respectively. Also, assume that register `$s2` holds the base address of the array `D`. Note that each element of Array `D` is 4 bytes long.

```
while (A < 10) {
   D[A] = B + A;
   A += 1;
}
```

```
loop:
    slti $t0, $s0, 10
    beq  $t0, $zero, exit
    add  $t1, $s0, $s1
    sll  $t2, $s0, 2
    add  $t2, $t2, $s2
    sw   $t1, 0($t2)
    addi $s0, $s0, 1
    j    loop
exit:
```

Q4. The following table shows the code of a C factorial procedure and its translation to MIPS assembly procedure. However, the MIPS code shown has five errors (some errors have multiple instances). Circle these errors and correct them in the space below.

C Code	MIPS Code	Line

`int fact (int n)`
`{`
`  if (n < 1) return f;`
`  else return n * fact(n - 1);`
`}`
	`fact:`
	`  addi $sp, $sp, -2`
	`  sw  1($sp), $at`
	`  sw  0($sp), $a0`
	`  slti $t0, $a0, 1`
	`  beq  $t0, $zero, L1`
	`  addi $v0, $zero, 1`
	`  addi $sp, $sp, 8`
	`  j  $at`
	`L1: addi $a0, $a0, -1`
	`  jal  fact`
	`  lw  $a0, 0($sp)`
	`  lw  $at, 1($sp)`
	`  addi $sp, $sp, 2`
	`  mul  $v0, $a0, $v0`
	`  j  $at`
		1
		2
		3
		4
		5
		6
		7
		8
		9
		10
		11
		12
		13
		14

Error 1:

In Lines 2 and 12, the stack should be increased by 8 bytes, not 2 bytes.

Error 2:

In Lines 3 and 4, the `sw`'s operands should be swapped.

Error 3:

In Lines 3, 8, 11, and 14, should use the `$ra` register, not `$at`.

Error 4:

In Lines 3 and 11 the displacement should be 4, not 1.

Error 5:

In Lines 8 and 14 should use `jr`, not `j`.

Q5. It is required to translate the five MIPS instructions shown below to machine code. For each instruction, specify the fields of the instruction word then convert the word to a 32-bit binary number. Note that the needed MIPS reference data is enclosed in the last page.

Address	Instruction	Instruction Word's Fields
0x00200000	L1: add $t1, $s0, $a1	0, 10, 5, 9, 0, 20
Binary: 0000 0010 0000 0101 0100 1000 0010 0000		
0x00200004	lbu $t2, 0($t1)	24, 9, A, 0
Binary: 1001 0001 0010 1010 0000 0000 0000 0000		
0x00200008	beq $t2, $zero, L2	4, A, 0, 2
Binary: 0001 0001 0100 0000 0000 0000 0000 0010		
0x0020000C	addi $s0, $s0, 1	8, 10, 10, 1
Binary: 0010 0010 0001 0000 0000 0000 0000 0001		
0x00200010	j L1	2, 080000
Binary: 0000 1000 0000 1000 0000 0000 0000 0000		
0x00200014	L2:	

MIPS Reference Data:

Instruction	OPCODE/FUNCT (Hex)
`add`	$0/20_h$
`lbu`	24_h
`beq`	4
`addi`	8
`j`	2

REGISTERS

NAME	NMBR	USE	STORE?
$zero	0	The Constant Value 0	N.A.
$at	1	Assembler Temporary	No
$v0-$v1	2-3	Values for Function Results and Expression Evaluation	No
$a0-$a3	4-7	Arguments	No
$t0-$t7	8-15	Temporaries	No
$s0-$s7	16-23	Saved Temporaries	Yes
$t8-$t9	24-25	Temporaries	No
$k0-$k1	26-27	Reserved for OS Kernel	No
$gp	28	Global Pointer	Yes
$sp	29	Stack Pointer	Yes
$fp	30	Frame Pointer	Yes
$ra	31	Return Address	Yes
$f0-$f31	0-31	Floating Point Registers	Yes

<Good Luck>