Midterm Exam

This is a preview of the published version of the quiz

Started: Oct 12 at 2:02pm

Quiz Instructions

Total: 80 points.

Unless stated otherwise, assume the processors use the MIPS ISA in this exam.

IMPORTANT! Hexadecimal number rules in your answers:

- Do not add the "0x" prefix. For example, if a value is 0x1234, write **1234**, not **0x1234**
- Do not add the leading zeros. For example, if a value is 0x1234, write **1234**, not **00001234**
- Do not mix uppercase and lowercase letters in youranswer. For example, **ABCD** or **abcd** is okay, but **aBcd** is not allowed.

(Questions 1-2)

The following table shows the execution times of two computer systems in seconds.

Program	CPU 1	CPU 2
A	100	200
B	400	300
C	1000	100
D	60	500

Question 1

5 pts

Calculate how faster CPU 1 is than CPU 2 using the **geometric mean**.

Since it is difficult to calculate $\sqrt[4]{}$ without a calculator, write the value of x^4 as your answer where x is the geometric mean.

Question 2

5 pts

If you execute programs A, B, C, and D just once per each, which CPU completes the execution earlier?

- ○ CPU 1
- ○ CPU 2
- ○ Same

Question 3

10 pts

Write the value of the registers in **hexadecimal** after executing the following instructions in order. DO NOT write in decimal (base 10).

Also, as we noted above, please do not write leading zeros nor "0x" (0x0000ABC0 should be written as ABC0).

	Code	Destination Register	Value
1	`addi $1, $0, 222`	$1	0x
2	`addi $2, $0, 82`	$2	0x
3	`nor $3, $2, $0`	$3	0x
4	`sll $4, $1, 24`	$4	0x
5	`sll $5, $3, 24`	$5	0x
6	`srl $6, $5, 8`	$6	0x
7	`ori $7, 0xC000`	$7	0x
8	`add $8, $7, $1`	$8	0x
9	`add $9, $8, $6`	$9	0x
10	`add $10, $9, $4`	$10	0x

Question 4

10 pts

Suppose you're an engineer at a CPU company. Your company designed a new CPU, but it seems like the new CPU has a bug.

You're suspecting that the **sign extension logic** is not working. To verify your hypothesis, you designed the following test program.

```
addi $1, $zero, 1
sll $1, $1, 16
addi $1, $1, -1
ori $2, $zero, 0x8000
sw $2, 0($zero)
lh $3, 2($zero)
add $10, $1, $3
```

After running the program above. Write the value of the register $10.

Assume you can freely access any memory address range, and the memory accesses are in big endian)

If **all** sign extension logics are working properly: 0x (<- This should be the same result as a bug-free MIPS CPU)

If **all** sign extension logics are NOT working: 0x

The answers should be in hexadecimal, and beware of the hexadecimal rule of this exam!

(Questions 5-6)

MIPS opcode and funct code samples

Instruction	Opcode	funct
`add`	000000	100000
`addi`	001000	-
`beq`	000100	-
`bne`	000101	-
`jal`	000011	-
`lw`	100011	-
`or`	000000	100101
`sll`	000000	000000
`slt`	000000	101010
`sub`	000000	100010
`sw`	101011	-
`xor`	000000	100110

Question 5

10 pts

Convert the following binary machine codes to MIPS assembly codes.

Since we're using assist the automated grading, please follow the following rules.

1. Write the opcode in lowercase (O: `add`, X: `ADD`)
2. Write the registers by their number ID not by their name, (O: `$10`, X: `$zero`)
3. Write a SINGLE space between the opcode and registers (O: `sub $10, $11, $11`, X: `sub $10,  $12,  $13`)
4. Between the registers and immediate values, write a single comma, and do not write the comma at the end (X: `addi $1, $2 $3,`)
5. Do not add any space between the offset and the base register for the memory instructions. (O: `sw $1, 124($2)`, X: `lw $2, 0 ( $3)`)

0x029E5026:

0xAFBEFFEC:

Question 6

10 pts

Convert the last BEQ instruction in the following code to binary machine code. Write the answers in hexadecimal.

```
L1: sw $10, 1000($8)
    addi $8, $8, 4
    slti $9, $8, 100
    beq $9, $zero, L1
```

0x