CMPUT 229: Computer Organization and Architecture I
Fall 2000–2001, Section A2
Midterm #2
Instructor: Paul Lu
(November 15, 2000)

Name: _____ SID: _____

Carefully read all of these instructions and the questions. Good luck!

1. Duration of the examination is 50 minutes.
2. Check that your exam package has 7 pages.
3. Answer all parts of all problems. There are 5 questions worth a total of 50 marks (i.e., one mark per minute of time).
4. No books, no notes, and no calculators.
5. You may use the provided MIPS Assembly Language reference pages taken from your textbook. Please return these pages to the instructor at the end of the exam.
6. Be concise and give clear answers.
7. Write all answers on the front of the exam pages and **within the space provided**. You may use the back of these pages for rough work, but it will **not** be marked.
8. If your answer is NOT legible, I cannot mark it.
9. **NOTE**: Here is a decimal, binary and hexadecimal conversion table.

Decimal	Binary	Hexadecimal	Decimal	Binary	Hexadecimal
0	0000	0	8	1000	8
1	0001	1	9	1001	9
2	0010	2	10	1010	A
3	0011	3	11	1011	B
4	0100	4	12	1100	C
5	0101	5	13	1101	D
6	0110	6	14	1110	E
7	0111	7	15	1111	F

Problem 1 (1 mark for each blank, 5 marks in total)
Fill in the blanks.

1. Assuming two's complement representation,
   $00001110_2 + 11111011_2 =$ _____ (give answer in decimal).

2. In digital logic, a _____ circuit is used to select one output line from n input lines. (The answer is the name of a kind of circuit we have studied.)

3. In the single-cycle CPU that we have studied, the arithmetic logic unit (ALU) is used to compute arithmetic expressions, logical expressions, and _____.

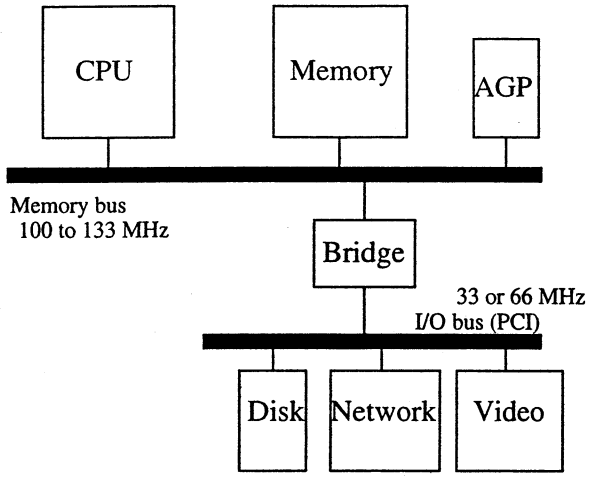
4. When I/O devices are accessed using regular load and store instructions to special addresses, that is known as _____.

5. The technique of using a loop to check for when a condition is true is known as _____.

Problem 2 (5 marks for each question, 15 marks in total)

Short answers (2 or 3 sentences) are expected for the following questions. Keep your answers **brief** and to the point.

Consider the basic computer architecture we studied in class:



1. What is one **significant advantage** to having a separate main memory bus and I/O bus in a computer?

2. What is one **significant disadvantage** to having a separate main memory bus and I/O bus in a computer?

3. What is one **significant disadvantage** to having special CPU instructions to perform I/O operations (e.g., `DISK_OUTPUT $s2`)?

Problem 3 (5 marks for question, 10 marks in total)

The next 2 questions (on 2 pages) refer to the following **buggy** MIPS assembly language code fragment, which is taken from SPIM's `trap.handler`.

NOTE: Register `$13` is the Cause Register. Register `$14` is the Exception Program Counter (EPC)

```
        .ktext 0x80000080
PointA:
PointB:
        mfc0 $k0 $14     # EPC
        jr $k0
        rfe
PointC:
```

1. Joe Programmer wants to write a simple exception handler that ignores all exceptions. If an instruction causes an exception, it is simply ignored and skipped over.

   Joe wrote the code shown above, but there are 2 significant bugs in the code. Re-write the code to do what Joe originally intended. You may have to re-write some instructions, add instructions, or move instructions.

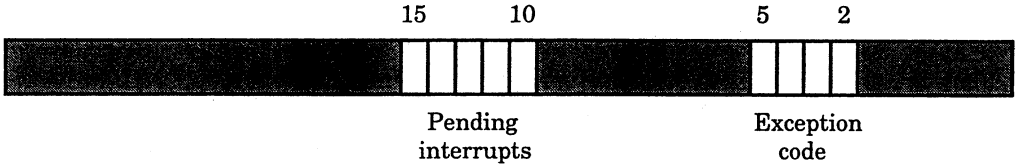
   Change as little of Joe's code as possible (i.e., do not write a new handler from scratch).

Problem 3 (continued)

2. Now, write **new** code that checks the Cause Register (`$13`) and branches to label `PointC` if a Reserved Instruction exception (value of decimal 10) was what occurred.

   The new code would go between the labels `PointA` and `PointB`. Assume that valid code already exists at label `PointC`, although it is now shown.

   Recall that the Cause Register looks like this:



   You only need to show the **new** code that goes between the labels `PointA` and `PointB`; do not copy any of the existing code shown above.

Problem 4 (1 mark for each blank, 8 marks in total)

Suppose you are asked to convert the following for-loop (which is written in C) to MIPS assembly language.

```
for( i = 0; i < j; i++ )
{
        /* Loop body */
}
```

Fill in the blanks (_____) in the following code to properly implement the for-loop.

NOTE: Register `$s0` is used to hold variable `i`. Register `$s1` is used to hold variable `j`.

```
Init-Top:

        li $s0, ______

Test:
        ______  $s0, ______, ______

    # /* Loop body */

        addi ______, ______, ______

        j ______

Exit-For:
```

Problem 5 (1 mark for each entry, 12 marks in total)

All of the rows in the following table contain different representations of the same number. Fill in the missing entries in the following table. As an example, the first row is complete.

Two's complement representation should be used. Sign extension should be used, where necessary.

Decimal	32-bit Hexadecimal	8-bit Binary
10	0000000A	0000 1010
	FFFFFFF0	
-2		
		0111 1101
80		
	00000101	
-10		

Page 2	Page 3	Page 4–5	Page 6	Page 7	
#1	#2	#3	#4	#5	Total
/5	/15	/10	/8	/12	50