Notes:
- Do not forget to write your name.
- There are 6 pages and 4 questions in total. Please check for missing/blank pages before you start answering.
- Read each question carefully and answer to the point.
- You may use your book or notes during the examination.
- If you make any assumptions, please state them clearly.
- Write your solutions in the space below the questions. You may use both sides of a page if necessary.
- Do not remove the staples from the exam sheets.
- Good Luck!

<table>
<thead>
<tr>
<th>Problem</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem 1</td>
<td>14</td>
</tr>
<tr>
<td>Problem 2</td>
<td>10</td>
</tr>
<tr>
<td>Problem 3</td>
<td>12</td>
</tr>
<tr>
<td>Problem 4</td>
<td>14</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>50</strong></td>
</tr>
</tbody>
</table>
Problem 1: (14 points) Answer briefly.
1) (2 points) What specific abstraction does hardware provide to the low-level software?

2) (2 points) What is the purpose of function field (funct) and shift amount (shamt) in R-type instructions in MIPS?

3) (2 points) MIPS 2000 machine allows only a 26-bit jump address. What is the implication of that? How can software get around the limitations of this addressing mode?

4) (2 points) Many instruction sets contain the instruction NOOP, meaning no operation, which has no effect on the state of CPU (like changing contents of any register or memory location) other than incrementing the program counter. Give two one-instruction examples, one with slt and the other with beq, that effectively achieve no operation.

5) (2 points) Consider the instruction sequence below.

```assembly
loop: lw $t1, 0($s1)  # load the word in Memory[$s1]
      sw $t1, 0($s2)  # copy the word to Memory[$s2]
      addi $s1, $s1, 4  # increment $s1 pointer by one word (4 bytes)
      addi $s2, $s2, 4  # increment $s2 pointer by one word (4 bytes)
      bne $s1, $s3, loop # repeat the copying until $s1 = $s3
exit: ...
```

When encoding the “bne $s1, $s3, loop” instruction in MIPS, what is the 16-bit offset encoded in the instruction?

<table>
<thead>
<tr>
<th>opcode</th>
<th>Rs</th>
<th>Rt</th>
<th>16-bit number</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>17</td>
<td>19</td>
<td>??</td>
</tr>
</tbody>
</table>

Answer: ____________
6) **(2 points)** Why do we choose the 2’s complement notation over 1’s complement notation for binary arithmetic in our processor design?

7) **(2 points)** If $s0$ contains the base address for integer array $A$, and $t0$ contains the constant $h$, write a MIPS code that is equivalent of the following C-statement: $A[30] = A[15] + h$.

MIPS Code:

**Problem 2: (10 points)** You are going to enhance a machine that executes our favorite program. The program has three classes of instructions: Class-A, Class-B, and Class-C. The instruction mix and the CPI for each of the instruction class is shown below:

<table>
<thead>
<tr>
<th>Class type</th>
<th>% occurrence</th>
<th>Number of cycles</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>40</td>
<td>3</td>
</tr>
<tr>
<td>B</td>
<td>40</td>
<td>4</td>
</tr>
<tr>
<td>C</td>
<td>20</td>
<td>5</td>
</tr>
</tbody>
</table>

We have two alternatives to improve the performance of the machine. The first alternative is to make Class-A instructions run three times faster and the second is to make Class-B instructions run two times faster.

1) **(4 points)** What is the speedup if you improve only Class-A instructions?

2) **(4 points)** What is the speedup if you improve only Class-B instructions?

3) **(2 points)** What is the speedup if both improvements are made?
Problem 3: (12 points) We are interested in finding out if a number in a given register (say $s0$) is a power of 2; i.e., the value of register $s0$ can be written as $2^i$ for some integer value of $i$ ($1 \leq i < 32$). Write a MIPS assembly language program to identify if the value in register $s0$ is a power of 2 or not. Store the result (1 if true, 0 otherwise) in the register $t0$.

[Useful instruction: sll $v1, $v0, 1. This instruction shifts the value of register $v0$ by one bit to the left and stores the result in $v1.] Write the logic of how you would identify if a number is a power of 2, an algorithm of how you would do this, and then convert it into MIPS code. Show your work clearly to obtain partial credits.
Problem 4: (14 points) A program that is of interest to us has been written using the standard MIPS assembly language to be executed on a standard MIPS processor. The program has 50% R-type instructions, 20% load word instructions, 10% store word instructions, and 20% branch instructions. Among the branch instructions in the program, 75% of them are used in the context of branch-if-less-than (blt), where the code uses set-if-less-than (slt) followed by branch-if-equals (beq) instruction.

The students of this class have devised a simple modification to the hardware that could directly execute branch-if-less-than (blt) instructions in the hardware. Assume that this hardware modification does not impact the clock cycle time. Also assume that blt and beq instructions would require the same number of cycles to execute in the modified hardware.

1) (5 points) With a YES, NO, or DEPENDS, identify which of the following metrics will be affected by the above enhancement. Write a brief comment if your answer any of the metric is DEPENDS stating the condition(s) under which it will be affected.

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Affected?</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of instructions executed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MIPS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Execution time</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2) (2 points) What is the speedup obtained with the above enhancement assuming a single-cycle implementation (all instructions take one clock cycle to complete)?

3) (4 points) If the hardware implementation uses varying number of cycles for different instructions, with R-type instructions requiring 4 cycles, load word instructions requiring 5 cycles, store word instructions requiring 4 cycles, and all branch instructions requiring 3 cycles, what is the CPI of the MIPS processor with and without enhancement?
4) (3 points) What is the speed-up obtained with the above enhancement made to the hardware with hardware implementation in Part 3 of this problem?