1. Suppose we are considering a change to an instruction set. The base machine initially has only loads and stores to memory, and all operations work on the registers. Measurements of the load-store machine showing the instruction mix and clock cycle counts per instruction are given in the table below:

<table>
<thead>
<tr>
<th>Instruction Type</th>
<th>Frequency</th>
<th>Clock Cycle Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALU Operations</td>
<td>43%</td>
<td>1</td>
</tr>
<tr>
<td>Loads</td>
<td>21%</td>
<td>2</td>
</tr>
<tr>
<td>Stores</td>
<td>12%</td>
<td>2</td>
</tr>
<tr>
<td>Branches</td>
<td>24%</td>
<td>2</td>
</tr>
</tbody>
</table>

Assume that 25% of the ALU Operations directly use a loaded operand that is not used again.

We propose adding ALU instructions that have one source operand in memory. These new register-memory instructions have a clock cycle count of 2. Suppose that the extended instruction set increases the clock cycle count for branches by 1 (so branches now take 3 cycles total), but it does not affect the clock cycle time. Would this change improve CPU performance? Justify your answer by showing your work.

2. Consider the following code fragment:

```assembly
loop: ld x1, 0(x2)
    addi x1, x1, #1
    sd 0(x2), x1
    addi x2, x2, #4
    sub x4, x3, x2
    bne x4, x0, loop
```

a) Show the timing of this instruction sequence for a processor that uses the 5-stage RISC-V pipeline, but without any data forwarding and without a delayed branch slot. Determine the number of cycles required to execute this loop.

b) Show the timing of this instruction sequence with all possible forwarding and bypassing hardware, and with a single-cycle delayed branch slot. Determine the number of cycles required to execute this loop.

c) Calculate the Speedup of the second machine vs the original one.