Write your name and UNI on your solutions

Show your work for each problem; we are more interested in how you get the answer than whether you get the right answer.
1. (25 pts.) Imagine each of the five control wires (A - E) in the above processor are stuck at a particular value. For each fault, provide an instruction that will still work and a second that will not. You should assume in each of the five cases, that all other wires are operating properly. Opcodes alone are sufficient, unless the operands are required to specify the scenario.
<table>
<thead>
<tr>
<th>Fault</th>
<th>Working Instruction</th>
<th>Broken Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>lw</td>
<td>any R-Type, e.g., add</td>
</tr>
<tr>
<td>B</td>
<td>anything that does not write memory, e.g., lw</td>
<td>sw</td>
</tr>
<tr>
<td>C</td>
<td>beq</td>
<td>any instr where ALUResult=0 that should not branch, e.g., add $t0, $0, $0</td>
</tr>
<tr>
<td>D</td>
<td>and</td>
<td>any non-and instr that uses the ALU, e.g., beq, lw, sw, other R-types</td>
</tr>
<tr>
<td>E</td>
<td>any instr that feeds immediate into ALU, e.g., lw or sw</td>
<td>any instr that doesn’t, e.g., beq, R-type</td>
</tr>
</tbody>
</table>
2. (25 pts.) Extend the single-cycle MIPS processor to support jal (j-format, opcode=000011).
3. (25 pts.) Consider a program, P, with 1 billion dynamic instructions, 50% R-Rype, 10% each of loads and stores, and 30% branches.

(a) How long would P take to execute on a single cycle processor with a 100MHz clock?

\[
\frac{10^9 \text{instructions}}{P} \cdot \frac{1 \text{cycle}}{\text{instruction}} \cdot \frac{10 \times 10^{-9} \text{seconds}}{\text{cycle}} = 10 \text{ seconds}
\]

(b) Assuming a multicycle processor where R-type instructions take 4 cycles, loads and stores 5 cycles, and branches 3 cycles, what is the CPI of P?

\[
CPI = 0.5 \cdot 4 + 0.1 \cdot 5 + 0.1 \cdot 5 + 0.3 \cdot 3 = 3.9
\]

(c) Assuming the multicycle processor operated at 400MHz, how long would it take to execute P?

\[
\frac{10^9 \text{instructions}}{P} \cdot \frac{3.9 \text{cycles}}{\text{instruction}} \cdot \frac{2.5 \times 10^{-9} \text{seconds}}{\text{cycle}} = 9.75 \text{ seconds}
\]
4. (25 pts.) Consider a new MIPS instruction “conditional move” or cmov. The instruction cmov $1, $2, $3 means “copy the value in register 2 into register 1 if register 3 is nonzero”.

If a processor does not support the cmov instruction, the function of the cmov instruction can be executed via software:

```
beq $3, $0, DONE
add $1, $2, $0
DONE
```

Imagine two single-cycle MIPS processors, $P$ and $P_{cmov}$. $P$ runs at 100MHz and does not implement the cmov instruction. $P_{cmov}$ runs at 90MHz does.

(a) Consider a 1 billion instruction application, called $A_{cmov}$, of which 5% of the instructions are cmov instructions. How long would it take to execute $A_{cmov}$ on $P_{cmov}$?

\[
\frac{1 \times 10^9 \text{instructions}}{\text{program}} \times \frac{1 \text{cycle}}{\text{instruction}} \times \frac{11.11 \times 10^{-9} \text{s}}{\text{cycle}} = 11.11 \text{s}
\]

(b) If $A_{cmov}$ were modified to remove all instances of cmov, creating $A$, how many dynamic instructions would $A$ have? Assume that the condition was true for all dynamic cmovs in $A_{cmov}$. 

\[
0.95 \times 10^9 + (2 \times 0.05 \times 10^9) = 1.05 \times 10^9
\]

(c) How long would
i. A take to execute on $P$?
\[
\frac{1.05 \times 10^9 \text{ instructions}}{\text{program}} \times \frac{1 \text{ cycle}}{\text{instruction}} \times \frac{10 \times 10^{-9} \text{ s}}{\text{cycle}} = 10.5 \text{ s}
\]

ii. A take to execute on $P_{cmov}$?
\[
\frac{1.05 \times 10^9 \text{ instructions}}{\text{program}} \times \frac{1 \text{ cycle}}{\text{instruction}} \times \frac{11.11 \times 10^{-9} \text{ s}}{\text{cycle}} = 11.66 \text{ s}
\]