1 Problem 1

\[ CPU_A = \frac{(IC_A \times CPI_A)}{FREQ_A} \]

\[ CPU_A = \frac{10^5 \times 1.3}{600 \times 10^6} \]

Similarly,

\[ CPU_B = \frac{(IC_B \times CPI_B)}{FREQ_B} = CPU_A \]

\[ IC_B = \frac{CPU_A \times FREQ_B}{CPI_B} = \frac{1.3 \times 10^5}{600 \times 10^6} \times 750 \times 10^6 = 6.5 \times 10^4 \]

2 Problem 2

1. • Negative Consequences: Data Memory will be “bypassed”. For example, “lw” instruction will always fail

• Code that will fail: “lw $t1 0($t0) - when “a.stuck at 0”, the value loaded into $t1 is “$t0 + 0” rather than the value at memory location “$t0 + 0”

• Code that will work: “sw $t1 0($t0)” - “sw” does not care the signal “MemtoReg"

2. • Negative Consequences: “ALU” will always perform addition

• Code that will fail: sub $t0 $t1 $t2 - ALUOp = 00 means the ALU Control can only produce the operation code as 0010 which means addition
• Code that will work: \texttt{add \$t0 \$t1 \$t2}

3. • Negative Consequences: PC register never get incremented correctly, PC = PC + 4. Instead, PC will always point to a “far” instruction.

• Code that will fail: \texttt{add \$t0 \$t1 \$t2}
  \texttt{add \$t0 \$t1 \$t2}
  the second instruction will not be executed, because “c.stuck at 1” makes PC register get some “garbage” address instead of PC + 4.

• Code that will work: \texttt{beq \$t0 \$t0 L}
  \texttt{L: add \$t0 \$t1 \$t2}
  Because \$t0 always equals to \$t0, this instruction will always jump to “L”.

### 3 Problem 3

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Reg-Dis</th>
<th>ALU-Src</th>
<th>Memto-Reg</th>
<th>Reg-Write</th>
<th>Mem-Read</th>
<th>Mem-Write</th>
<th>Branch</th>
<th>ALU-Op1</th>
<th>ALU-Op2</th>
</tr>
</thead>
<tbody>
<tr>
<td>addi</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0 (or X)</td>
<td>0 (or X)</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Instruction Code</th>
<th>ALUop</th>
<th>Instruction Operation</th>
<th>Function Field</th>
<th>Desired ALU Action</th>
<th>ALU Control Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>addi</td>
<td>00</td>
<td>add immediate</td>
<td>xxxxxxx</td>
<td>add</td>
<td>0010</td>
</tr>
</tbody>
</table>