Homework 10
Assembly Language & Architecture

1. Answer the questions for each MIPS processor in this table, whose stages have the latencies shown.

<table>
<thead>
<tr>
<th></th>
<th>IF</th>
<th>ID</th>
<th>EX</th>
<th>MEM</th>
<th>WB</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>300 ps</td>
<td>400 ps</td>
<td>350 ps</td>
<td>500 ps</td>
<td>100 ps</td>
</tr>
<tr>
<td>P2</td>
<td>200 ps</td>
<td>150 ps</td>
<td>120 ps</td>
<td>190 ps</td>
<td>140 ps</td>
</tr>
</tbody>
</table>

a. How long will a clock cycle be, in both a pipelined and nonpipelined system? (500 ps, 1650 ps; 200 ps, 800 ps)

b. What will the total latency of a `lw` instruction be, in both a pipelined and nonpipelined system? (2500 ps, 1650 ps; 1000 ps, 800 ps)

c. If one could split any single stage into two new stages (each with half the latency) which stage should be split and what would the new period of the processor be, assuming a pipelined system? What would the total latency of `lw` be? (MEM, 400 ps, 2400 ps; IF, 190 ps, 1140 ps)

2. For the following questions, consider the MIPS code on the right. Assume that each of these instructions is divided into five stages (IF, ID, EX, MEM, and WB), and that each stage takes exactly one cycle to complete.

```
lw $t1, 0($s0)
lw $t2, 4($s0)
add $t3, $t1, $t2
sw $t3, 0($s0)
sub $t4, $s4, $t3
lw $t5, 0($s1)
```

a. How many cycles will it take to execute this code, assuming that the system is not pipelined? (30 cycles)

b. How many cycles will it take, assuming that the system is pipelined, but cannot support forwarding? (16 cycles)

c. How many cycles will it take, assuming that the system is pipelined, and allows forwarding? (11 cycles)

c. How many cycles will it take, assuming that the system is pipelined, allows forwarding, and a good compiler has optimized this code (reordering instructions)? (10 cycles)