Winter ’11 CIS 314 Final Review Solutions

1. [25] Consider the following MIPS code:

```
add $t0 $t1 $t2
add $t3 $t4 $t5
lw $t6 0($t0)
add $t7 $t0 $t6
```

a. [5] Identify all data dependencies in the above code:

- `lw` - $t0 depends on first `add`
- `third add` - $t0 depends on first `add`, $t6 depends on `lw`

b. [5] Assuming no forwarding or load delay slot but that registers can be read and written in the same cycle, how many cycles are stalled when executing the above code? Why?

<table>
<thead>
<tr>
<th>cycle</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>IF</td>
<td>ID</td>
<td>EX</td>
<td>MEM</td>
<td>WB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>add</td>
<td>IF</td>
<td>ID</td>
<td>EX</td>
<td>MEM</td>
<td>WB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lw</td>
<td>IF</td>
<td>ID</td>
<td>EX</td>
<td>MEM</td>
<td>WB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>add</td>
<td>IF</td>
<td>ID</td>
<td>EX</td>
<td>MEM</td>
<td>WB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- `lw` - stall 1 cycle so that the $t0 is written in the same cycle that the `lw` reads $t0
- `third add` - stall 2 cycles so that $t6 is written in the same cycle that the `add` reads $t6
c. [5] Assuming forwarding (including register read/write in the same cycle) but no load delay slot, how many cycles are stalled when executing the above code? Why?

<table>
<thead>
<tr>
<th>cycle</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>IF</td>
<td>ID</td>
<td>EX</td>
<td>MEM</td>
<td>WB</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>add</td>
<td>IF</td>
<td>ID</td>
<td>EX</td>
<td>MEM</td>
<td>WB</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lw</td>
<td>IF</td>
<td>ID</td>
<td>EX</td>
<td>MEM</td>
<td>WB</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>add</td>
<td>IF</td>
<td>ID</td>
<td>EX</td>
<td>MEM</td>
<td>WB</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*add* - stall one cycle so that loaded value can be forwarded to the last add’s EX.

d. [5] Assuming forwarding (including register read/write in the same cycle) and a filled load delay slot, how many cycles are stalled when executing the above code? Why?

no stalls - third add is delayed by one cycle while we execute the load delay instruction so that the loaded value can be immediately forwarded to the third add’s EX.

e. [5] Which instruction above could you use to fill the load delay slot? Why?

*add $t3 $t4 $t5 because is has no dependencies with the other instructions*
2. [20] Consider the following single-cycle processor diagram:

![Single-cycle processor diagram]

**Figure 4.17 (p. 322)**

a. [10] What are the values of the RegDst, Branch, MemRead, MemtoReg, MemWrite, ALUSrc, and RegWrite, and zero binary control lines for a `beq` instruction?

RegDst = doesn’t matter, RegWrite is 0, Branch = 1, MemRead = 0, MemtoReg = doesn’t matter, RegWrite is 0, MemWrite = 0, ALUSrc = 0, RegWrite = 0

b. [10] What decimal register-number values are sent to the Read register 1, Read register 2, and Write register inputs of the register unit for an `add $5 $6 $7` instruction?

Read register 1 = rs = 6, Read register 2 = rt = 7, Write register = rd = 5

3. [10] How many control lines are required for a decoder circuit with 1024 outputs?

\[ \lg 1024 = 10 \text{ (and } 2^{10} = 1024) \]