Homework #8 Solutions – Computer Architecture Exercises

Note: All CLO’s in this problem set tie to ABET program-level criterion a.

MIPS Pipeline Architecture: On your answer sheet, fill in the instruction data and check the control lines that are active. Remember that you ONLY turn in your answer sheet.

- **Problem 1:** (CLO 1—Comp. Arch.) On your answer sheet, fill in the data values for add $t2,$t0,$t1. Assume that $t0 contains 0x0000 0011 and $t1 contains 0x0000 002c. This will enable you to fill in the numerical data on your answer sheet. Remember to check the five (5) control lines which are active in the control line list on your answer sheet.

- **Problem 2:** (CLO 1—Comp. Arch.) On your answer sheet, fill in the data values for and $t9,$t8,$t7. Assume that $t7 contains 0x00ff 0812 and $t8 contains 0xfe77 4956. This will enable you to fill in the numerical data on your answer sheet. Remember to check the five (5) control lines which are active in the control line list on your answer sheet. Also, remember that AND is a bitwise logic function (i.e., AND bit 1 of the two operands together, store the result in bit 1 of the destination, and do the same for bit 2, bit 3, etc.).

- **Problem 3:** (CLO 1—Comp. Arch.) On your answer sheet, fill in the data values for lw $t0,16($t1). Assume that $t1 contains 0x1001 0248, and that the data memory location contains 0x7a61394f. This will enable you to fill in the numerical data on your answer sheet. Remember to check the six (6) control lines which are active in the control line list on your answer sheet.

- **Problem 4:** (CLO 1—Comp. Arch.) On your answer sheet, fill in the data values for sw $t2,32($t3). Assume that $t3 contains 0x1001 000f, and that $t2 contains 0x000044e3. This will enable you to fill in the numerical data on your answer sheet. Remember to check the three (3) control lines which are active in the control line list on your answer sheet.
• Problem 1

```
add $t2, $t0, $t1
```

```
<table>
<thead>
<tr>
<th>Op Code</th>
<th>Fn. Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>2</td>
</tr>
</tbody>
</table>
```

![Diagram](image-url)
Problem 2.

<table>
<thead>
<tr>
<th>Op Code</th>
<th>Fn. Code</th>
<th>( a )</th>
<th>( n )</th>
<th>( &amp; )</th>
<th>( t_9 ), ( t_8 ), ( t_7 )</th>
</tr>
</thead>
</table>

IF/ID | ID | EX | MEM | MEM/WB |

ADD

Instruction Memory

Instr. Address
Instr. Bits 0-31

PC

Op Code

Control Decode

Branch

Register Block

[\$t8]
[\$t7]

[\$t9]

Rs
Rt
Rd
Write

Read Data 1
Read Data 2

Sign Extend

Bits 0-15

Bits 16-20

Bits 11-15

Op Code

MUX

Add

Left Shift 2

ALU Srce.

ALU

Branch

Mem/ALU

ALU Bypass

Data Memory

[\$t9]

Write Data

[0x0077 0812]

[0x00ff 0812]

[0x00ff 0812]

[0xfe77 4956]

[0x0077 0812]

and \$t9, \$t8, \$t7

Instruction Bits 0-31

IF/ID

ID/EX

ID/EX

EX/MEM

MEM/WB

MUX


Problem 3:

```
• Problem 3:

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Memory</th>
<th>ALU Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>+4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Instr. Addr</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ 0x1001\ 0248 \]

\[ 0x10 \]

\[ 0x7a61 394f \]

lw \$t0, 16($t1)
```
Problem 4:

The diagram illustrates the processor pipeline stages and data flow for the instruction `sw $t2,32($t3)`. The stages include Instruction Memory, IF/ID, ID/EX, and MEM/WB. The operations involve reading and writing to memory, arithmetic operations (ADD), and control logic (sign extension, ALU bypass, etc.).