CS150 Fall 2012 — Worksheet for Discussion #6

MIPS150 ISA

- Do not only go by what you find in P&H or other textbooks – utilize the course’s ISA for all implementation details if they differ from other sources (e.g. ‘green sheet’)
- Memory operations can be especially tricky – different bit widths of desired quantities

Problem 1:
Consider the following descriptions of memory operations:

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LB</td>
<td>[ R[rt] = SEXT (BMEM ([R[rs] + SEXT (imm)][31 : 0])] ] delayed</td>
</tr>
<tr>
<td>LH</td>
<td>[ R[rt] = SEXT (BMEM ([R[rs] + SEXT (imm)][31 : 1])] ] delayed</td>
</tr>
<tr>
<td>LW</td>
<td>[ R[rt] = BMEM ([R[rs] + SEXT (imm)][31 : 2]) ] delayed</td>
</tr>
<tr>
<td>SB</td>
<td>[ BMEM ([R[rs] + SEXT (imm)][31 : 0]) = R[rt][7 : 0] ]</td>
</tr>
<tr>
<td>SH</td>
<td>[ BMEM ([R[rs] + SEXT (imm)][31 : 1]) = R[rt][15 : 0] ]</td>
</tr>
<tr>
<td>SW</td>
<td>[ BMEM ([R[rs] + SEXT (imm)][31 : 2]) = R[rt][31 : 0] ]</td>
</tr>
</tbody>
</table>

Assume the word at memory address 0x12345678 (which is a word-aligned address) contains the data word 0xDEADBEEF. Determine the contents of the registers \$r2 and \$r3 after the following instruction sequences if the contents are known (assume big-endianness).

1. 

   \[
   \begin{align*}
   &LUI \quad \$r3, 0x1234 \\
   &LW \quad \$r3, 0x5678(\$r3) \\
   &ADDU \quad \$r2, \$r0, \$r0 \\
   &ADDIU \quad \$r2, \$r3, 0x1
   \end{align*}
   \]

2. 

   \[
   \begin{align*}
   &ADDU \quad \$r2, \$r0, \$r0 \\
   &LUI \quad \$r3, 0x1234 \\
   &LW \quad \$r3, 0x5678(\$r3) \\
   &ADDIU \quad \$r2, \$r3, 0x1
   \end{align*}
   \]

3. 

   \[
   \begin{align*}
   &LUI \quad \$r3, 0x1234 \\
   &LB \quad \$r3, 0x5679(\$r3) \\
   &ADDU \quad \$r2, \$r0, \$r0 \\
   &ADDIU \quad \$r2, \$r3, 0x1
   \end{align*}
   \]

MIPS Pipelining

- Be aware of the effect of synchronous elements (regfile, memories) on the placement of pipelining registers
- Typical MIPS pipelining stages are IF, ID, EX, MEM, and WB.
  - You have worked with 1,2, and 5 stages – how will you use three?
  - Where will register fetch fit in?
  - Where will writeback fit in? How will the memory delay slot affect forwarding?