1 Administrivia

Announcements

Assignment

Read 6.3.

From Last Time

Overview of pipelining.

Outline


2. Simple example: a single lw.

Coming Up

Pipelined control.
2 Pipelining

A pipelined datapath:

Consider four instructions: R-mode, a branch, LW, SW.

Observations:

1. Not a true pipeline: feedback.

2. How do we re-design control?

2.1 Hazards

1. Structural hazards.

   Example: unified L1 cache/memory.

2. Control hazards. Consider the following example:
Solutions:

(a) Stall.

(b) Predict.

Static prediction. Truly static. Compile-time determined.

Dynamic prediction. Branch history tables. One-, two-bit counters.

(c) Delayed branch.

Assumes you know branch outcome early.

Code scheduling:
Consideration: deeper pipelines.

3. Data hazards.

Data not available when needed.

ALU example:

<table>
<thead>
<tr>
<th>Value of register $2$</th>
<th>CC 1</th>
<th>CC 2</th>
<th>CC 3</th>
<th>CC 4</th>
<th>CC 5</th>
<th>CC 6</th>
<th>CC 7</th>
<th>CC 8</th>
<th>CC 9</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10-20</td>
<td>-20</td>
<td>-20</td>
<td>-20</td>
<td>-20</td>
</tr>
</tbody>
</table>

Program execution order (in instructions):

- sub $2, $1, $3$
- and $12, $2, $5$
- or $13, $6, $2$
- add $14, $2, $2$
- sw $15, 100($2$)
Fixed by forwarding.

Memory example:

How can this be fixed?

### 2.2 Simple Example

Let’s follow a lw. What’s going on during each clock cycle?
Figure 06.15