Lecture: Pipelining Extensions

- Topics: bypassing, deeper pipelines, control hazards
Bypassing: 5-Stage Pipeline

Source: H&P textbook
Problem 6

- Show the instruction occupying each stage in each cycle (with bypassing)
  if I1 is R1+R2 → R3 and I2 is R3+R4 → R5 and I3 is R3+R8 → R9.
  Identify the input latch for each input operand.

<table>
<thead>
<tr>
<th>CYC-1</th>
<th>CYC-2</th>
<th>CYC-3</th>
<th>CYC-4</th>
<th>CYC-5</th>
<th>CYC-6</th>
<th>CYC-7</th>
<th>CYC-8</th>
</tr>
</thead>
<tbody>
<tr>
<td>IF</td>
<td>IF</td>
<td>IF</td>
<td>IF</td>
<td>IF</td>
<td>IF</td>
<td>IF</td>
<td>IF</td>
</tr>
<tr>
<td>D/R</td>
<td>D/R</td>
<td>D/R</td>
<td>D/R</td>
<td>D/R</td>
<td>D/R</td>
<td>D/R</td>
<td>D/R</td>
</tr>
<tr>
<td>ALU</td>
<td>ALU</td>
<td>ALU</td>
<td>ALU</td>
<td>ALU</td>
<td>ALU</td>
<td>ALU</td>
<td>ALU</td>
</tr>
<tr>
<td>DM</td>
<td>DM</td>
<td>DM</td>
<td>DM</td>
<td>DM</td>
<td>DM</td>
<td>DM</td>
<td>DM</td>
</tr>
<tr>
<td>RW</td>
<td>RW</td>
<td>RW</td>
<td>RW</td>
<td>RW</td>
<td>RW</td>
<td>RW</td>
<td>RW</td>
</tr>
</tbody>
</table>
Problem 6

- Show the instruction occupying each stage in each cycle (with bypassing) if I₁ is \( R₁ + R₂ \rightarrow R₃ \) and I₂ is \( R₃ + R₄ \rightarrow R₅ \) and I₃ is \( R₃ + R₈ \rightarrow R₉ \).

Identify the input latch for each input operand.

<table>
<thead>
<tr>
<th>CYC-1</th>
<th>CYC-2</th>
<th>CYC-3</th>
<th>CYC-4</th>
<th>CYC-5</th>
<th>CYC-6</th>
<th>CYC-7</th>
<th>CYC-8</th>
</tr>
</thead>
<tbody>
<tr>
<td>IF I₁</td>
<td>IF I₂</td>
<td>IF I₃</td>
<td>IF I₄</td>
<td>IF I₅</td>
<td>IF</td>
<td>IF</td>
<td>IF</td>
</tr>
<tr>
<td>D/R I₁</td>
<td>D/R I₂</td>
<td>D/R I₃</td>
<td>D/R I₄</td>
<td>D/R I₅</td>
<td>D/R</td>
<td>D/R</td>
<td>D/R</td>
</tr>
<tr>
<td>ALU</td>
<td>ALU</td>
<td>ALU</td>
<td>ALU</td>
<td>ALU</td>
<td>ALU</td>
<td>ALU</td>
<td>ALU</td>
</tr>
<tr>
<td>DM</td>
<td>DM</td>
<td>DM</td>
<td>DM</td>
<td>DM</td>
<td>DM</td>
<td>DM</td>
<td>DM</td>
</tr>
<tr>
<td>RW</td>
<td>RW</td>
<td>RW</td>
<td>RW</td>
<td>RW</td>
<td>RW</td>
<td>RW</td>
<td>RW</td>
</tr>
</tbody>
</table>
Pipeline Implementation

- Signals for the muxes have to be generated – some of this can happen during ID
- Need look-up tables in decode stage to identify situations that merit bypassing/stalling – the number of inputs to the muxes goes up

![Pipeline Implementation Diagram]
Problem 7

• For the 5-stage pipeline (RR and RW take half a cycle)

IF  D/ RR  AL  DM  RW

• For the following pairs of instructions, how many stalls will the 2\textsuperscript{nd} instruction experience (with and without bypassing)?

- ADD R3 ← R1+R2
  ADD R5 ← R3+R4
- LD R2 ← [R1]
  ADD R4 ← R2+R3
- LD R2 ← [R1]
  SD R3 → [R2]
- LD R2 ← [R1]
  SD R2 → [R3]
Problem 7

- For the 5-stage pipeline (RR and RW take half a cycle)

- For the following pairs of instructions, how many stalls will the 2\textsuperscript{nd} instruction experience (with and without bypassing)?

- ADD R3 $\leftarrow$ R1+R2
  - ADD R5 $\leftarrow$ R3+R4
  - without: 2   with: 0

- LD R2 $\leftarrow$ [R1]
  - ADD R4 $\leftarrow$ R2+R3
  - without: 2   with: 1

- LD R2 $\leftarrow$ [R1]
  - SD R3 $\rightarrow$ [R2]
  - without: 2   with: 1

- LD R2 $\leftarrow$ [R1]
  - SD R2 $\rightarrow$ [R3]
  - without: 2   with: 0
Summary

• For the 5-stage pipeline, bypassing can eliminate delays between the following example pairs of instructions:
  add/sub       R1, R2, R3
  add/sub/lw/sw R4, R1, R5

  lw        R1, 8(R2)
  sw        R1, 4(R3)

• The following pairs of instructions will have intermediate stalls:
  lw        R1, 8(R2)
  add/sub/lw R3, R1, R4 or sw R3, 8(R1)

  fmul     F1, F2, F3
  fadd     F5, F1, F4
Problem 8

- Consider this 8-stage pipeline (RR and RW take a full cycle)

```
IF  DE  RR  AL  AL  DM  DM  RW
```

- For the following pairs of instructions, how many stalls will the 2nd instruction experience (with and without bypassing)?

- ADD R3 \(\leftarrow\) R1+R2
  ADD R5 \(\leftarrow\) R3+R4
- LD R2 \(\leftarrow\) [R1]
  ADD R4 \(\leftarrow\) R2+R3
- LD R2 \(\leftarrow\) [R1]
  SD R3 \(\rightarrow\) [R2]
- LD R2 \(\leftarrow\) [R1]
  SD R2 \(\rightarrow\) [R3]
Problem 8

• Consider this 8-stage pipeline (RR and RW take a full cycle)

IF  DE  RR  AL  AL  DM  DM  RW

• For the following pairs of instructions, how many stalls will the 2\textsuperscript{nd} instruction experience (with and without bypassing)?

- ADD R3 ← R1+R2
  ADD R5 ← R3+R4
  without: 5  with: 1

- LD R2 ← [R1]
  ADD R4 ← R2+R3
  without: 5  with: 3

- LD R2 ← [R1]
  SD R3 → [R2]
  without: 5  with: 3

- LD R2 ← [R1]
  SD R2 → [R3]
  without: 5  with: 1
Stalls from Control Hazards

Source: H&P textbook
Control Hazards

• Simple techniques to handle control hazard stalls:
  ➢ for every branch, introduce a stall cycle (note: every 6th instruction is a branch on average!)
  ➢ assume the branch is not taken and start fetching the next instruction – if the branch is taken, need hardware to cancel the effect of the wrong-path instructions
  ➢ predict the next PC and fetch that instr – if the prediction is wrong, cancel the effect of the wrong-path instructions
  ➢ fetch the next instruction (branch delay slot) and execute it anyway – if the instruction turns out to be on the correct path, useful work was done – if the instruction turns out to be on the wrong path, hopefully program state is not lost
Branch Delay Slots

(a) From before

DADD R1, R2, R3
if R2 = 0 then

Delay slot

becomes

if R2 = 0 then

DADD R1, R2, R3

(b) From target

DSUB R4, R5, R6

DADD R1, R2, R3
if R1 = 0 then

Delay slot

becomes

DSUB R4, R5, R6

(c) From fall-through

DADD R1, R2, R3
if R1 = 0 then

Delay slot

OR R7, R8, R9

DSUB R4, R5, R6

becomes

OR R7, R8, R9

DSUB R4, R5, R6
Problem 1

• Consider a branch that is taken 80% of the time. On average, how many stalls are introduced for this branch for each approach below:
  ▪ Stall fetch until branch outcome is known
  ▪ Assume not-taken and squash if the branch is taken
  ▪ Assume a branch delay slot
    o You can’t find anything to put in the delay slot
    o An instr before the branch is put in the delay slot
    o An instr from the taken side is put in the delay slot
    o An instr from the not-taken side is put in the slot
Problem 1

• Consider a branch that is taken 80% of the time. On average, how many stalls are introduced for this branch for each approach below:
  ▪ Stall fetch until branch outcome is known – 1
  ▪ Assume not-taken and squash if the branch is taken – 0.8
  ▪ Assume a branch delay slot
    o You can’t find anything to put in the delay slot – 1
    o An instr before the branch is put in the delay slot – 0
    o An instr from the taken side is put in the slot – 0.2
    o An instr from the not-taken side is put in the slot – 0.8