Pipeline and Hazards
Data Forwarding and Pipeline Scheduling

Dr. A. Sahu
CSE IIT Guwahati

Types of Pipelined processors
• Degree of overlap
  – Serial, Overlapped, Pipelined,
  – Super-pipelined/Superscalar
• Depth
  – Shallow, Deep
• Structure
  – Linear, Non - linear
• Scheduling of operations
  – Static, Dynamic

Degree of overlap

Serial

Overlapped

Pipelined

Depth

Shallow

Deep

Pipeline Structure

Linear Pipeline

Non-linear Pipeline

Sequence: A, B, C, B, C, A, C, A

Scheduling/timing alternatives

• Static
  – same sequence of stages for all instructions
  – all actions in order
  – if one instruction stalls, all subsequent instructions are delayed
• Dynamic
  – above conditions are relaxed
  – higher throughput is achieved

Dynamic Scheduling

• type 1 : beginnings (decode) and endings (put away) in order
• type 2 : only beginnings in order
• type 3 : no order restrictions except dependencies
• type 1 extended : beginnings in order, references that effect memory state are in order
  [note that a memory reference may lead to page fault]
Pipelining and CPI

<table>
<thead>
<tr>
<th>Type</th>
<th>CPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Serial</td>
<td>5 – 6</td>
</tr>
<tr>
<td>Overlapped</td>
<td>3</td>
</tr>
<tr>
<td>Pipelined (static)</td>
<td>1.5 – 2</td>
</tr>
<tr>
<td>Pipelined (dynamic)</td>
<td>1.2 – 1.5</td>
</tr>
<tr>
<td>Multiple instruction issue</td>
<td>&lt; 1.0</td>
</tr>
</tbody>
</table>

Hazards in Pipelining

- Data dependencies => Data hazards
  - RAW (read after write)
  - WAR (write after read)
  - WAW (write after write)
- Resource conflicts => Structural hazards
  - use of same resource in different stages
- Procedural dependencies => Control hazards
  - conditional and unconditional branches, calls/returns

Data Hazards

Data forwarding path P1

I: \( \text{i} \) \( \text{IM} \rightarrow \text{RF} \rightarrow \text{DM} \rightarrow \text{EX} \rightarrow \text{IF} \)

I+1: \( \text{i+1} \) \( \text{IM} \rightarrow \text{RF} \rightarrow \text{DM} \rightarrow \text{EX} \rightarrow \text{IF} \)

Data forwarding path P2

I: \( \text{i} \) \( \text{IM} \rightarrow \text{RF} \rightarrow \text{DM} \rightarrow \text{EX} \rightarrow \text{IF} \)

I+1: \( \text{i+1} \) \( \text{IM} \rightarrow \text{RF} \rightarrow \text{DM} \rightarrow \text{EX} \rightarrow \text{IF} \)

Data forwarding path P2

I: \( \text{i} \) \( \text{IM} \rightarrow \text{RF} \rightarrow \text{DM} \rightarrow \text{EX} \rightarrow \text{IF} \)

I+1: \( \text{i+1} \) \( \text{IM} \rightarrow \text{RF} \rightarrow \text{DM} \rightarrow \text{EX} \rightarrow \text{IF} \)
Data forwarding path P3

Data forwarding path P4

Data forwarding paths

Data forwarding path list

• P1 from ALU out (EX/DM) to ALU in1/2
• P2 from DM/ALU out (DM/WB) to ALU in1/2
• P3/P4 from DM/ALU out (DM/WB) to DM in

Drawing Dependence Graph

Reference


Chapter 6.4/6.5, third edition
Ebook can be found
Structural Hazards
Caused by Resource Conflicts
- Use of a hardware resource in more than one cycle
- Different sequences of resource usage by different instructions
- Non-pipelined multi-cycle resources

Analysis of Structural Hazards
Multi-functional Pipeline

Reservation Table for X
(Required Resources of Instruction in Cycle)

With Initiation Interval = 3

Collisions with Initiation Interval = 2

Collisions with Initiation Interval = 5
Latency Sequences and Cycles

No Collision for 1, 8, 3 and 6 interval

1, 8, 1, 8, .... (1, 8) avg = 4.5
3, 3, 3, 3, .... (3) avg = 3
6, 6, 6, 6, .... (6) avg = 6

Minimum Average Latency?

Latency Cycles from State Diagram

Latency Cycles
(1, 8)  (1, 8, 6, 8)  (3)  (6)  (3, 8)  (3, 6, 3)
Simple Latency Cycles (no figure repeats)
(1, 8)  (3)  (6)  (3, 8)  (6, 8)
Greedy Latency Cycles
(1, 8)  (3) - from different starting states

Collision Free Scheduling for X

Minimum Average Latency (MAL)

MAL ≥ max no. of check marks in any row
MAL ≤ avg latency of any greedy cycle

avg latency of any greedy cycle ≤
no. of 1’s in initial collision vector + 1

Upper Bound on MAL

- Consider a greedy cycle \((k_1,k_2,...,k_n)\)
- Let \(p = \text{no. of 1's in initial collision vector}\)
  \[ \Rightarrow k_1 \leq p + 1 \]
  \[ k_2 \leq 2p - k_1 + 2 \]
  \[ k_3 \leq 3p - k_1 - k_2 + 3 \]
  ....
  \[ k_n \leq np - k_1 - k_2 \ldots - k_{n-1} + n \]
  \[ \Rightarrow k_1 + k_2 \ldots + k_n \leq np + n \Rightarrow \text{MAL} \leq p + 1 \]

Reference

Kai Hwang, "Advanced Computer Architecture: Parallelism, Scalability, Programmability"
Chapter 6
Control Hazards

- Branch instr
- Next inline instr
- Target instr

- Cond eval
- Target addr gen

- Delay = 2
- Delay = 5

- The order of cond eval and target addr gen may be different
- Cond eval may be done in previous instruction

Improving Branch Performance

- Branch Elimination
  - Replace branch with other instructions
- Branch Speed Up
  - Reduce time for computing CC and TIF
- Branch Prediction
  - Guess the outcome and proceed, undo if necessary
- Branch Target Capture
  - Make use of history