Decoupled Access/Execute Computer Architectures + Retrospective

James E. Smith

Kasia and Lillie, cse 548 wi05
Summary

- Decoupling access from execution
- Implementation issues
  - Stores
  - Conditional branches
  - Queues implemented with registers
Issues

- Deadlock prevented by compiler
- How to merge instruction streams?
- Going from this to this

\[ q = 0.0 \]
\[ \text{Do } 1 \ k = 1, 400 \]
\[ x(k) = q + y(k) \times (r \times z(k+10) + t \times z(k+11)) \]

<table>
<thead>
<tr>
<th>Access</th>
<th>Execute</th>
</tr>
</thead>
<tbody>
<tr>
<td>\cdot</td>
<td>\cdot</td>
</tr>
<tr>
<td>\cdot</td>
<td>\cdot</td>
</tr>
<tr>
<td>AEQ + z + 10, A2</td>
<td>X4 + X2 *f AEQ</td>
</tr>
<tr>
<td>AEQ + z + 11, A2</td>
<td>X3 + X5 *f AEQ</td>
</tr>
<tr>
<td>AEQ + y, A2</td>
<td>X6 + X3 +f X4</td>
</tr>
<tr>
<td>A7 + A7 + 1</td>
<td>EAQ + AEQ *f X6</td>
</tr>
<tr>
<td>x, A2 + EAQ</td>
<td>\cdot</td>
</tr>
<tr>
<td>A2 + A2+ A3</td>
<td>\cdot</td>
</tr>
<tr>
<td>\cdot</td>
<td>\cdot</td>
</tr>
<tr>
<td>\cdot</td>
<td>\cdot</td>
</tr>
</tbody>
</table>

Fig. 2c. Access and execute programs for straight-line section of loop
Benefits

• Decoupling performance gains:
  – Processor-memory communication speed is less of an issue
  – One instruction per cycle bottleneck not an issue
  – Improvement = 1.71 on average

• Reduction of programmer responsibility

• Two PCs makes interrupts easier to deal with than in other multi-processor architectures
Critique

• Instruction stream merge
  – Performance evaluation?
• Deadlock prevention
  – Moving the problem to the compiler
• Hand-compiled code
• They assume optimum conditions in their evaluation
• How did they come up with the timings?
  – Human error seems to be likely
• “Does it work? Nyeeeh . . .” - Schwerin
Questions

• Max speedup from decoupled processor = 2.5, while for a pair of strictly serial processors = 2.0. If there were efforts to improve performance today, would a decoupled architecture be an option, or would the “design from scratch” be too costly?

• Arithmetic mean is apparently a faux pas. How are average speedups calculated today?

• What was the result of the study on the performance impact of the WAQ length?
More questions

• How do they decide queue length?
• What else can we decouple from what?
• What would happen if this joined forces with out-of-order processing?
• Did anyone get very far building a compiler for this? Are there any terribly clever compiler tricks we can apply?
• DEA vs. Superscalar in a fight to the death: who wins? Who gets the most points for style?