Today: Beyond the 5-stage pipeline

- Taxonomy: Introduction to advanced processor techniques.

- Superpipelining: Increasing the number of pipeline stages.

- Superscalar: Issuing several instructions in a single cycle.

Superpipelining: Add more stages

Goal: Reduce critical path by adding more pipeline stages.

Example: 8-stage ARM XScale: extra IF, ID, data cache stages.

Difficulties: Added penalties for load delays and branch misses.

Ultimate Limiter: As logic delay goes to 0, FF clk-to-Q and setup.

Superscalar: Multiple issues per cycle

Goal: Improve CPI by issuing several instructions per cycle.

Example: CPU with floating point ALUs: issue 1 FP + 1 integer instruction per cycle.

Difficulties: Load and branch delays affect more instructions.

Ultimate Limiter: Programs may be a poor match to issue rules.
Out of Order: Going around stalls

Goal: Issue instructions out of program order

Example:

\[
\begin{array}{c|c|c}
\text{Instruction} & \text{Program} & \text{Cycles} \\
\hline
\text{LD} & F_2, & 34(R2) \\
\text{ADD} & F_4, & 45(R3) \\
\text{MUL} & F_6, & J \\
\end{array}
\]

MULTP waiting on F4 to load ...

Difficulties: Bookkeeping is highly complex. A poor fit for lockstep instruction scheduling.

Ultimate Limiter: The amount of instruction level parallelism present in an application.

Throughput and multiple threads

Goal: Use multiple CPUs (real and virtual) to improve (1) throughput of machines that run many programs (2) execution time of multi-threaded programs.

Example: Sun Niagara (8 SPARCs on one chip).

Difficulties: Gaining full advantage requires rewriting applications, OS, libraries.

Ultimate limiter: Amdahl’s law, memory system performance.

Dynamic Scheduling: End lockstep

Goal: Enable out-of-order by breaking pipeline in two: fetch and execution.

Example: IBM Power 5:

Admintrivia: No class on Thursday!

- HW 4: due Weds 11/10, 5PM, 283 Soda.
- Friday 11/12: Lab 4 final demo in section.
- Monday 11/15: Lab 4 final report due, 11:59 PM.
- Final project (Lab 5) will be out soon

Superpipelining

Xilinx field trip date: 11/30. Details on bus transport from Soda Hall soon.
Add pipeline stages, reduce clock period

Q. Could adding pipeline stages reduce CPI for an application?
A. Yes, due to these problems:

<table>
<thead>
<tr>
<th>CPI Problem</th>
<th>Possible Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extra branch delays</td>
<td>Branch prediction</td>
</tr>
<tr>
<td>Extra load delays</td>
<td>Optimize code</td>
</tr>
<tr>
<td>Structural hazards</td>
<td>Optimize code, add hardware</td>
</tr>
</tbody>
</table>

FO4: How many fanout-of-4 inverter delays in the clock period.

Is there an optimal pipeline depth?
Methodology: Simulate standard benchmarks on many different pipeline designs.

Source: “The Optimal Logic Depth Per Pipeline Stage is 6 to 8 FO4 Inverter Delays”. N.S. Hrishikesh et al. ISCA 2002.

Superscalar: A simple example ...
Example: Superscalar MIPS. Fetches 2 instructions at a time. If first integer and second floating point, issue in same cycle.
Limitations of “lockstep” superscalar

- Only get 0.5 CPI for a 50/50 mix of float and integer ops with no hazards.
- Extending scheme to general mixes and more instructions is complicated.
- If one accepts building a complicated machine, there are better ways to do it.

Next time: Dynamic Scheduling

Recall: Branch Predictors are Caches

```plaintext
0b0110[...]01001000 BNEZ R1 Loop

Branch Target Buffer (BTB)
28-bit address tag target address
0b0110[...]0100 PC + 4 + Loop

Branch History Table (BHT)

"Taken" or "Not Taken"

Note: BHT can be larger than BTB; does not need a tag.
```

Conclusion: Superpipelining, Superscalar

- The 5 stage pipeline: a starting point for performance enhancements, a building block for multiprocessing.
- Superpipelining: Reduce critical path by adding more pipeline stages. Has the potential to increase the CPI.
- Superscalar: Multiple instructions at once. Programs must fit the issue rules. Adds complexity.