Review #1/3: Pipelining & Performance

- Just overlap tasks; easy if tasks are independent
- Speed Up $\leq$ Pipeline Depth; if ideal CPI is 1, then:
  \[ \text{Speedup} = \frac{\text{Cycle Time}_{\text{ideal}}} {\text{Cycle Time}_{\text{pipelined}}} \]

- Hazards limit performance on computers:
  - Structural: need more HW resources
  - Data (RAW, WAR, WAW): need forwarding, compiler scheduling
  - Control: delayed branch, prediction

- Time is measure of performance: latency or throughput

- CPI Law:

<table>
<thead>
<tr>
<th>CPU time (Seconds)</th>
<th>Program</th>
<th>Program</th>
<th>Instruction</th>
<th>Cycle</th>
</tr>
</thead>
</table>

Review #2/3: Caches

- The Principle of Locality:
  - Program access a relatively small portion of the address space at any instant of time.
    - Temporal Locality: Locality in Time
    - Spatial Locality: Locality in Space

- Three Major Categories of Cache Misses:
  - Compulsory Misses: sad facts of life. Example: cold start misses.
  - Capacity Misses: increase cache size
  - Conflict Misses: increase cache size and/or associativity.

- Write Policy:
  - Write Through: needs a write buffer.
  - Write Back: control can be complex

- Today CPU time is a function of (ops, cache misses) vs. just f(ops): What does this mean to Compilers, Data structures, Algorithms?

Now, Review of Virtual Memory

Basic Issues in VM System Design

- size of information blocks that are transferred from secondary to main storage (M)
- block of information brought into M, and M is full, then some region of M must be released to make room for the new block \( \rightarrow \) replacement policy
- which region of M is to hold the new block \( \rightarrow \) placement policy

missing item fetched from secondary memory only on the occurrence of a fault \( \rightarrow \) demand load policy

Paging Organization

Address Map

- virtual address space \( n > m \)
- physical address space \( M = \{0, 1, \ldots, m-1\} \)

MAP: \( V \rightarrow M \cup \{0\} \) address mapping function

- \( \text{MAP}(a) = a' \) if data at virtual address \( a \) is present in physical address \( a' \) in M
- \( = 0 \) if data at virtual address \( a \) is not present in M

- OS performs this transfer

- Name Space
  - Memory
    - Fault
    - Handler
  - Main Memory
    - Page Map
      - Pages

Page 1
Paging Organization

Virtual Address and a Cache

Virtual Address and a Cache

TLBs

Translation Look-Aside Buffers

Reducing Translation Time

Overlapped Cache & TLB Access
Problems With Overlapped TLB Access
Overlapped access only works as long as the address bits used to index into the cache do not change as the result of VA translation. This usually limits things to small caches, large page sizes, or high n-way set associative caches if you want a large cache.

Example: suppose everything the same except that the cache is increased to 8 K bytes instead of 4 K:

<table>
<thead>
<tr>
<th>virt page #</th>
<th>disp</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>12</td>
</tr>
</tbody>
</table>

This bit is changed by VA translation, but is needed for cache lookup. Solutions:
- go to 8K byte page sizes;
- go to 2 way set associative cache; or

SPEC: System Performance Evaluation
Cooperative

- Fourth Round 2000: SPEC CPU2000
  - 12 Integer
  - 14 Floating Point
  - 2 choices on compilation: "aggressive" (SPECint_base2000, SPECfp_base): flags same for all programs, no more than 4 flags, same compiler for conservative, can change for aggressive
  - multiple data sets so that can train compiler if trying to collect data for input to compiler to improve optimization

How to Summarize Performance
- Arithmetic mean (weighted arithmetic mean) tracks execution time: \( \frac{\sum T_i}{n} \) or \( \frac{\sum (W_i* T_i)}{\sum W_i} \)
- Harmonic mean (weighted harmonic mean) of rates (e.g., MFLOPS) tracks execution time: \( n/\sum (1/R_i) \) or \( n/\sum (W_i/R_i) \)
- Normalized execution time is handy for scaling performance (e.g., X times faster than SPARCstation 10)
- But do not take the arithmetic mean of normalized execution time, use the geometric mean:
  \( \left( \frac{\sum T_j}{N} \right)^{1/n} \)

Impact of Means on SPECmark89 for IBM 550

<table>
<thead>
<tr>
<th>Program</th>
<th>Before</th>
<th>After</th>
<th>Before</th>
<th>After</th>
<th>Before</th>
<th>After</th>
</tr>
</thead>
<tbody>
<tr>
<td>gcc</td>
<td>30</td>
<td>29</td>
<td>49</td>
<td>51</td>
<td>8.91</td>
<td>9.22</td>
</tr>
<tr>
<td>espresso</td>
<td>35</td>
<td>34</td>
<td>65</td>
<td>67</td>
<td>7.64</td>
<td>7.86</td>
</tr>
<tr>
<td>spice</td>
<td>47</td>
<td>47</td>
<td>510</td>
<td>510</td>
<td>5.69</td>
<td>5.69</td>
</tr>
<tr>
<td>doduc</td>
<td>46</td>
<td>49</td>
<td>41</td>
<td>38</td>
<td>5.81</td>
<td>5.45</td>
</tr>
<tr>
<td>nasa7</td>
<td>78</td>
<td>144</td>
<td>258</td>
<td>140</td>
<td>3.43</td>
<td>1.88</td>
</tr>
<tr>
<td>li</td>
<td>34</td>
<td>34</td>
<td>183</td>
<td>183</td>
<td>7.86</td>
<td>7.86</td>
</tr>
<tr>
<td>eqntott</td>
<td>40</td>
<td>40</td>
<td>28</td>
<td>28</td>
<td>6.68</td>
<td>6.68</td>
</tr>
<tr>
<td>matrix300</td>
<td>78</td>
<td>730</td>
<td>58</td>
<td>6</td>
<td>3.43</td>
<td>0.37</td>
</tr>
<tr>
<td>fpppp</td>
<td>90</td>
<td>87</td>
<td>34</td>
<td>35</td>
<td>2.97</td>
<td>3.07</td>
</tr>
<tr>
<td>tomcatv</td>
<td>33</td>
<td>138</td>
<td>20</td>
<td>19</td>
<td>2.01</td>
<td>1.94</td>
</tr>
<tr>
<td>Mean</td>
<td>54</td>
<td>72</td>
<td>124</td>
<td>108</td>
<td>54.42</td>
<td>49.99</td>
</tr>
</tbody>
</table>

Impact Ratio: Arithmetic Ratio: 1.33 Weighted Ratio: 1.33

SPEC First Round
- One program: 99% of time in single line of code
- New front-end compiler could improve dramatically

SPEC: System Performance Evaluation Cooperative

- First Round 1989
  - 10 programs yielding a single number ("SPECmarks")
- Second Round 1992
  - SPECint92 (6 integer programs) and SPECfp92 (14 floating point programs)
  - Compiler Flags unlimited. March 93 of DEC 4000 Model 610:
    - spice: unix:/def=(sysv,has_bcopy,"bcopy(a,b,c)=memcpy(b,a,c)"
    - wave5: /ali=(all,dcom=nat)/ag=a/ur=4/ur=200
    - nasa7: /norecu/aga/ur=4/ur=200/lc=blas
- Third Round 1995
  - new set of programs: SPECint95 (8 integer programs) and SPECfp95 (10 floating point)
  - "benchmarks useful for 3 years"
  - Single flag setting for all programs: SPECint_base95, SPECfp_base95
Performance Evaluation

• "For better or worse, benchmarks shape a field"
• Good products created when have:
  - Good benchmarks
  - Good ways to summarize performance
• Given sales is a function in part of performance relative to competition, investment in improving product as reported by performance summary
• If benchmarks/summary inadequate, then choose between improving product for real programs vs. improving product to get more sales; Sales almost always wins!
• Execution time is the measure of computer performance!

Integrated Circuits Costs

\[
\text{Die Cost} = \text{Die cost} + \text{Testing cost} + \text{Packaging cost}
\]

\[
\text{Die cost} = \frac{\text{Die per wafer}}{\text{Die/Area}} = \frac{\text{Wafer diam} \cdot \text{Die Area}}{\text{Die Area}
\]

\[
\text{Die Yield} = \frac{\text{Wafer yield}}{1 - \text{Defect Density} \cdot \text{Die area}}
\]

Cost/Performance

What is Relationship of Cost to Price?

• Component Costs
  • Direct Costs (add 25% to 40%) recurring costs: labor, purchasing, scrap, warranty
  • Gross Margin (add 82% to 186%) nonrecurring costs: R&D, marketing, sales, equipment maintenance, rental, financing cost, pretax profits, taxes
• Average Discount to get List Price (add 33% to 66%): volume discounts and/or retailer markup

Real World Examples

<table>
<thead>
<tr>
<th>Chip</th>
<th>Metal Line</th>
<th>Wafer Defect Area</th>
<th>Dies/ Yield Die Cost</th>
<th>Avg. Selling Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>386DX</td>
<td>2</td>
<td>0.90</td>
<td>$900</td>
<td>71%</td>
</tr>
<tr>
<td>486DX2</td>
<td>3</td>
<td>0.80</td>
<td>$1200</td>
<td>54%</td>
</tr>
<tr>
<td>PowerPC 601</td>
<td>4</td>
<td>0.80</td>
<td>$1700</td>
<td>28%</td>
</tr>
<tr>
<td>HP PA 7100</td>
<td>3</td>
<td>0.80</td>
<td>$1300</td>
<td>27%</td>
</tr>
<tr>
<td>DEC Alpha</td>
<td>3</td>
<td>0.70</td>
<td>$1500</td>
<td>19%</td>
</tr>
<tr>
<td>SuperSPARC</td>
<td>3</td>
<td>0.70</td>
<td>$1700</td>
<td>13%</td>
</tr>
<tr>
<td>Pentium</td>
<td>3</td>
<td>0.80</td>
<td>$1500</td>
<td>9%</td>
</tr>
</tbody>
</table>


Chip Prices (August 1993)

• Assume purchase 10,000 units

<table>
<thead>
<tr>
<th>Chip</th>
<th>Area</th>
<th>Mfg.</th>
<th>Price Multi</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>386DX</td>
<td>43</td>
<td>$9</td>
<td>$31</td>
<td>3.4 Intense Competition</td>
</tr>
<tr>
<td>486DX2</td>
<td>81</td>
<td>$35</td>
<td>$245</td>
<td>7.0 No Competition</td>
</tr>
<tr>
<td>PowerPC 601</td>
<td>121</td>
<td>$77</td>
<td>$280</td>
<td>3.6</td>
</tr>
<tr>
<td>DEC Alpha</td>
<td>234</td>
<td>$202</td>
<td>$1231</td>
<td>6.1 Recoup R&amp;D?</td>
</tr>
<tr>
<td>Pentium</td>
<td>296</td>
<td>$473</td>
<td>$965</td>
<td>2.0 Early in shipments</td>
</tr>
</tbody>
</table>
CS 252 Course Focus

Understanding the design techniques, machine structures, technology factors, evaluation methods that will determine the form of computers in 21st Century

- Technology
- Parallelism
- Programming Languages
- Interface Design (ISA)
- Compilers
- Computer Architecture: Instruction Set Design
- Organization
- Hardware/Software Boundary
- Operating Systems
- Measurement & Evaluation
- History
- Applications

Topic Coverage

Research Papers -- Handed out in class

- 1 week: Review: Fundamentals of Computer Architecture (Ch. 1), Pipelining, Performance, Caches, Virtual Memory, Cost, Ics
- 1 week: Memory Hierarchy (Chapter 5)
- 2 weeks: Fault Tolerance, Queuing Theory, Input/Output and Storage (Ch. 6)
- 2 weeks: Networks and Clusters (Ch. 7)
- 2 weeks: Multiprocessors (Ch. 8)
- 2 weeks: Instruction Sets, DSPs, SIMD (Ch. 2), Vector Processors (Appendix B).
- 1 week: Dynamic Execution (Ch 3)
- 1 week: Static Execution (Ch 4)
- Rest: Project strategy meetings, presentations, quizzes

Lecture style

- 1-Minute Review
- 20-Minute Lecture/Discussion
- 5-Minute Administrative Matters
- 25-Minute Lecture/Discussion
- 5-Minute Class Discussion or Break (water, stretch)
- 25-Minute Lecture/Discussion
- Instructor will come to class early & stay after to answer questions

Attention

20 min. Break "In Conclusion, ..."

Quizzes

- Reduce the pressure of taking quizzes
  - Only 2 Graded Quizes:
    - Tentative: Wed Mar 7th and Wed. Apr 18th
    - Our goal: test knowledge vs. speed writing
    - 3 hrs to take 1.5-hr test (5:30-8:30 PM, TBA location)
    - Both mid-term quizzes can bring summary sheet
    - Transfer ideas from book to paper
    - Last chance Q&A: during class time day of exam
  - Students/Faculty meet over free pizza/drinks at La Vals:
    - Wed Oct. 18th (8:30 PM) and Wed Apr 18th (8:30 PM)

Original

Big Fishes Eating Little Fishes

1988 Computer Food Chain

Big Fishes Eating Little Fishes

- Massively Parallel Processors
- Supercomputer
- Mini
- Work Station
- Work PC
- Mainframe

Page 5
1998 Computer Food Chain

Massively Parallel Processors

Mini-supercomputer

Mainframe

Supercomputer

PC Workstation

Server

Workstation

Now who is eating whom?

Why Such Change in 10 years?

- Performance
  - Technology Advances
    - CMOS VLSI dominates older technologies (TTL, ECL) in cost AND performance
    - Computer architecture advances improves low-end
      - RISC, superscalar, RAID, ...
  - Price: Lower costs due to ...
    - Simpler development
      - CMOS VLSI: smaller systems, fewer components
    - Higher volumes
      - CMOS VLSI: same dev. cost 10,000 vs. 10,000,000 units
    - Lower margins by class of computer, due to fewer services
- Function
  - Rise of networking/local interconnection technology

Technology Trends: Microprocessor Capacity

Moore’s Law

"Graduation Window"

Alpha 21264: 15 million
Pentium Pro: 5.5 million
PowerPC 620: 6.9 million
Alpha 21064: 9.3 million
Sparc Ultra: 5.2 million

CMOS improvements:
- Die size: 2x every 3 yrs
- Line width: halve / 7 yrs

Memory Capacity (Single Chip DRAM)

<table>
<thead>
<tr>
<th>Year</th>
<th>size(Mb)</th>
<th>cyc time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1980</td>
<td>0.0625</td>
<td>250 ns</td>
</tr>
<tr>
<td>1983</td>
<td>0.25</td>
<td>220 ns</td>
</tr>
<tr>
<td>1986</td>
<td>1</td>
<td>190 ns</td>
</tr>
<tr>
<td>1989</td>
<td>4</td>
<td>165 ns</td>
</tr>
<tr>
<td>1992</td>
<td>16</td>
<td>145 ns</td>
</tr>
<tr>
<td>1996</td>
<td>64</td>
<td>120 ns</td>
</tr>
<tr>
<td>2000</td>
<td>256</td>
<td>100 ns</td>
</tr>
</tbody>
</table>

Technology Trends (Summary)

Capacity      Speed (latency)
Logic         2x in 3 years 2x in 3 years
DRAM          4x in 3-4 years 2x in 10 years
Disk          4x in 2-3 years 2x in 10 years

Processor Performance Trends

1000

100

10

1

0.1


Year
**Processor Performance**
(1.35X before, 1.55X now)

**Performance Trends**
(Summary)
- Workstation performance (measured in SPEC marks) improves roughly 50% per year (2X every 18 months)
- Improvement in cost performance estimated at 70% per year

**Moore’s Law Paper**
- Discussion
- What did Moore predict?
- 35 years later, how did it hold up?
- In your view, what was biggest surprise in paper?

**Review #3/3: TLB, Virtual Memory**
- Caches, TLBs, Virtual Memory all understood by examining how they deal with 4 questions: 1) Where can block be placed? 2) How is block found? 3) What block is replaced on miss? 4) How are writes handled?
- Page tables map virtual address to physical address
- TLBs make virtual memory practical
  - Locality in data => locality in addresses of data, temporal and spatial
- TLB misses are significant in processor performance
  - Funny times, as most systems can’t access all of 2nd level cache without TLB misses!
- Today VM allows many processes to share single memory without having to swap all processes to disk; today VM protection is more important than memory hierarchy

**Summary**
- Performance Summary needs good benchmarks and good ways to summarize performance
- Transistors/chip for microprocessors growing via “Moore’s Law” 2X 1.5x/yr
- Disk capacity (so far) is at a faster rate last 4-5 years
- DRAM capacity is at a slower rate last 4-5 years
- In general, Bandwidth improving fast, latency improving slowly