Chapter 1: Fundamentals of Computer Design (Part 2)

What is computer architecture?
Why study computer architecture?

Common principles
- Performance
  - What is performance: latency, throughput
  - The performance equation
  - Measuring performance
  - Improving performance: parallelism, locality, Amdahl's law
- Cost
- Power
- Reliability

What is Performance?

Two Metrics
- Latency (or response time or execution time)
- Throughput (or bandwidth)

Performance (Cont.)

Definition: \( X \) is \( n \)% faster than \( Y \) if

\[
\frac{\text{Execution Time}_Y}{\text{Execution Time}_X} = 1 + \frac{n}{100}
\]

Example: \( X = 1 \) minute, \( Y = 2 \) minutes
- \( X \) is 100% faster than \( Y \)

Key Performance Equation

\[
CPU_{time} = \frac{\text{instructions}}{\text{program}} \times \frac{\text{cycles}}{\text{instruction}} \times \frac{\text{time}}{\text{cycle}}
\]

Instructions per program (path length)
- ISA and compiler

Cycles per instruction (CPI)
- ISA and organization (e.g., cache misses)

Time per cycle (clock time, cycle time)
- Organization and hardware
Measuring Performance

MIPS, MFLOPS don't mean much

Benchmarks
- Real programs
  - Representative of real workload
  - Only way to characterize performance
  - SPEC89 → SPEC92 → SPEC95 → SPEC CPU2000 → CPU2006
  - SPECFS, SPECWeb, SPECjbb, SPECvirt_sc2010, TPC

Kernels
- "Representative" program fragments
- Often not representative of full applications
- EEMBC for embedded systems
- Toy benchmarks and synthetic benchmarks
  - Don't mean much

Improving Performance – Basic Principles

Parallelism

Locality

Focus on common case – Amdahl's law

Amdahl's Law

(Or why the common case matters most)

Let

\[ Speedup = \frac{\text{new rate}}{\text{old rate}} = \frac{\text{old latency}}{\text{new latency}} \]

Consider an enhancement \( x \) that speeds up fraction \( f_x \) of a task by \( S_x \)

\[ Speedup_{overall} = \frac{\text{old latency}}{\text{new latency}} = \frac{(1 - f_x) + (f_x S_x)}{(1 - f_x) \times \text{old latency} + f_x S_x \times \text{old latency}} \]

Amdahl's law gives

\[ Speedup_{overall} = \frac{1}{(1 - f_x) + f_x S_x} \]

Amdahl's Law, cont.

Example: \( f_x = 95\% \) and \( S_x = 1.10 \)

\[ Speedup_{overall} = \frac{1}{(1 - 0.95) + (0.95/1.10)} = 1.094 \]

Example: \( f_x = 5\% \) and \( S_x = 10 \)

\[ Speedup_{overall} = \frac{1}{(1 - 0.05) + (0.05/10)} = 1.047 \]

Example: \( f_x = 5\% \) and \( S_x = \infty \)

\[ Speedup_{overall} = \frac{1}{(1 - 0.05) + (0.05/\infty)} = 1.052 \]
**Amdahl’s Law Corollary**

Since $S_x \to \infty$ implies

$$Speedup_{overall} = \frac{1}{(1-f_x) + (f_x/\infty)}$$

For all real speedups:

$$Speedup_{overall} < \frac{1}{1-f_x}$$

Or make the common case fast

An application?

<table>
<thead>
<tr>
<th>$f_x$</th>
<th>$1/(1-f_x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1%</td>
<td>1.01</td>
</tr>
<tr>
<td>2%</td>
<td>1.02</td>
</tr>
<tr>
<td>5%</td>
<td>1.05</td>
</tr>
<tr>
<td>10%</td>
<td>1.11</td>
</tr>
<tr>
<td>20%</td>
<td>1.25</td>
</tr>
<tr>
<td>50%</td>
<td>2.00</td>
</tr>
</tbody>
</table>

**Example**

Cost

Cost is very important in most real designs
But usually hard to quantify for the architect

Costs change over time
Learning curve lowers manufacturing costs
Technology improvements lower costs

Focus on IC costs bigger price variable
Cost vs. price

---

**A Wafer**

Figure 1.15 This 300 mm wafer contains 280 full Sandy Bridge dies, each 20.7 by 10.5 mm in a 32 nm process. (Sandy Bridge is Intel’s successor to Nehalem used in the Core i7.) At 216 mm², the formula for dies per wafer estimates 282. (Courtesy Intel.)

Copyright © 2011, Elsevier Inc. All rights Reserved.

**A Die**

Figure 1.13 Photograph of an Intel Core i7 microprocessor die, which is evaluated in Chapters 2 through 5. The dimensions are 18.9 mm by 13.6 mm (257 mm²) in a 45 nm process. (Courtesy Intel.)

Copyright © 2011, Elsevier Inc. All rights Reserved.
**Integrated Circuit Cost**

Cost of IC = \[ \frac{\text{Cost of Die} + \text{Cost of Testing} + \text{Cost of Packaging}}{\text{Final Test Yield}} \]

Cost of Die = \[ \frac{\text{Cost of Wafer}}{\text{Dies per Wafer} \times \text{Die Yield}} \]

Dies per Wafer = \[ (\pi \times \left(\frac{\text{Wafer Diameter}}{2}\right)^2 - \frac{\text{Die Area}}{\text{Die Area}}) \]

(Correction factor for Edge Effects)

\[ \text{Die Yield} = \frac{\text{Wafer Yield} \times \frac{1}{(1 + \text{Defects per unit area} \times \text{Die Area})^\alpha}}{\text{Die Area}} \]

\[ \alpha = 11.5 \text{ to } 15.5 \text{ for } 40\text{nm in } 2010 \Rightarrow \text{large dependence of cost on die area} \]

Cost per die grows roughly as the square of the die area

Price only loosely related to cost

---

**Power**

Power = Dynamic power + Static power

Energy = Power \times Time

Dynamic Power \propto \text{Capacitance} \times \text{Voltage}^2 \times \text{Frequency}

Static power = Static current \times \text{Voltage}

---

**Reliability**

Many sources of unreliability

Soft errors due to radiation, hard errors due to wearout, ...

Common metrics

Mean time to failure – MTTF

For exponentially distributed time to failure

Define failures in time or FITs

\[ \text{FIT} = \frac{1}{\text{MTTF}} \]

FIT of system = Sum of FITs of components

Common solution