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)
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

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

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

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

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

Amdahl’s law gives

\[ \text{Speedup}_{\text{overall}} = \frac{1}{(1 - f_x) + f_x / S_x} \]
Amdahl's Law, cont.

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

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

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

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

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

\[
\text{Speedup}_{\text{overall}} = \frac{1}{(1 - 0.05) + (0.05/\infty)} = 1.052
\]
Amdahl's Law Corollary

Since $S_x \to \infty$ implies

$$\text{Speedup}_{\text{overall}} = \frac{1}{(1 - f_x) + \left(\frac{f_x}{\infty}\right)}$$

For all real speedups:

$$\text{Speedup}_{\text{overall}} < \frac{1}{1 - f_x}$$

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>
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
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.)
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.)
Figure 1.14 Floorplan of Core i7 die in Figure 1.13 on left with close-up of floorplan of second core on right.
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}}

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

\text{(Correction factor for Edge Effects)}

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

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

\text{Cost per die grows roughly as the square of the die area}

\text{Price only loosely related to cost}
Power

Power = Dynamic power + Static power

Energy = Power * Time

Dynamic Power $\propto$ Capacitance * Voltage$^2$ * Frequency

Static power = Static current * 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

   FIT = 1/MTTF

   FIT of system = Sum of FITs of components

Common solution