Chapter 1

Fundamentals of Quantitative Design and Analysis
In the beginning...

- Complex Instruction Set Computing (CISC):
  - Lots of assembly instructions – easy to program

- Reduced Instruction Set Computing (RISC):
  - Fewer instructions – relies on High-Level Languages (HLL) and compilers
  - Simpler (i.e., faster) hardware
Single Processor Performance

Move to multicore

Copyright © 2012, Elsevier Inc. All rights reserved.
Current Trends in Architecture

- Cannot continue to leverage Instruction-Level parallelism (ILP)
  - Single processor performance improvement has slowed since 2003

- New models for performance:
  - Data-level parallelism (DLP)
  - Thread-level parallelism (TLP)
  - Request-level parallelism (RLP)

- These require explicit restructuring of the application
Flynn’s Taxonomy

- Single instruction stream, single data stream (SISD)
- Single instruction stream, multiple data streams (SIMD)
  - Vector architectures
  - Multimedia extensions
  - Graphics processor units
- Multiple instruction streams, single data stream (MISD)
  - No commercial implementation
- Multiple instruction streams, multiple data streams (MIMD)
  - Tightly-coupled MIMD
  - Loosely-coupled MIMD
Computer Technology

Performance improvements:

- Improvements in semiconductor technology
  - Feature size, clock speed
- Improvements in computer architectures
  - Enabled by HLL compilers, UNIX
  - Lead to RISC architectures

Together have enabled:

- Lightweight computers
- Productivity-based managed/interpreted programming languages
Classes of Computers

- **Personal Mobile Device (PMD)**
  - e.g. start phones, tablet computers
  - Emphasis on energy efficiency and real-time interactivity

- **Desktop Computing**
  - Emphasis on price-performance

- **Servers**
  - Emphasis on availability, scalability, throughput

- **Clusters / Warehouse Scale Computers**
  - Used for “Software as a Service (SaaS)”
  - Emphasis on availability and price-performance
  - Sub-class: Supercomputers, emphasis: floating-point performance and fast internal networks

- **Embedded Computers**
  - Emphasis: price
Defining Computer Architecture

“Old” view of computer architecture:
- Instruction Set Architecture (ISA) design
- i.e. decisions regarding:
  - registers, memory addressing, addressing modes, instruction operands, available operations, control flow instructions, instruction encoding
# MIPS Instruction Formats

## Basic instruction formats

<table>
<thead>
<tr>
<th>R</th>
<th>instruction format</th>
<th>opcode</th>
<th>rs</th>
<th>rt</th>
<th>rd</th>
<th>sham</th>
<th>funct</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>31 26 25 21 20 16 15 11 10 6 5</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>instruction format</td>
<td>opcode</td>
<td>rs</td>
<td>rt</td>
<td>immediate</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>31 26 25 21 20 16 15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J</td>
<td>instruction format</td>
<td>opcode</td>
<td>address</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>31 26 25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Floating-point instruction formats

<table>
<thead>
<tr>
<th>FR</th>
<th>instruction format</th>
<th>opcode</th>
<th>fmt</th>
<th>ft</th>
<th>fs</th>
<th>fd</th>
<th>funct</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>31 26 25 21 20 16 15 11 10 6 5</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FI</td>
<td>instruction format</td>
<td>opcode</td>
<td>fmt</td>
<td>ft</td>
<td>immediate</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>31 26 25 21 20 16 15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Defining Computer Architecture

“Real” computer architecture:
- Specific requirements of the target machine
- Design to maximize performance within constraints: cost, power, and availability
- Includes ISA, microarchitecture, hardware
Intel Core i7 (18.9 x 13.6 mm)
Trends in Technology

- Integrated circuit technology
  - Transistor density: 35%/year
  - Die size: 10-20%/year
  - Integration overall: 40-55%/year

- DRAM capacity: 25-40%/year (slowing)

- Flash capacity: 50-60%/year
  - 15-20X cheaper/bit than DRAM

- Magnetic disk technology: 40%/year
  - 15-25X cheaper/bit than Flash
  - 300-500X cheaper/bit than DRAM
Bandwidth and Latency

- Bandwidth or throughput
  - Total work done in a given time
  - 10,000-25,000X improvement for processors
  - 300-1200X improvement for memory and disks

- Latency or response time
  - Time between start and completion of an event
  - 30-80X improvement for processors
  - 6-8X improvement for memory and disks
Bandwidth and Latency

Log-log plot of bandwidth and latency milestones
Transistors and Wires

- Feature size
  - Minimum size of transistor or wire in x or y dimension
  - 10 microns in 1971 to .032 microns in 2011
  - Transistor performance scales linearly
    - Wire delay does not improve with feature size!
  - Integration density scales quadratically
## Feature Sizes

<table>
<thead>
<tr>
<th>Name</th>
<th>Count</th>
<th>Year</th>
<th>Co</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel 4004</td>
<td>2,300</td>
<td>1971</td>
<td>Intel</td>
<td>10 μm</td>
</tr>
<tr>
<td>Intel 8008</td>
<td>3,500</td>
<td>1972</td>
<td>Intel</td>
<td>10 μm</td>
</tr>
<tr>
<td>Intel 8080</td>
<td>4,500</td>
<td>1974</td>
<td>Intel</td>
<td>6 μm</td>
</tr>
<tr>
<td>Intel 8088</td>
<td>29,000</td>
<td>1979</td>
<td>Intel</td>
<td>3 μm</td>
</tr>
<tr>
<td>Intel 80286</td>
<td>134,000</td>
<td>1982</td>
<td>Intel</td>
<td>1.5 μm</td>
</tr>
<tr>
<td>Intel 80386</td>
<td>2/5,000</td>
<td>1985</td>
<td>Intel</td>
<td>1.5 μm</td>
</tr>
<tr>
<td>Intel 80486</td>
<td>1,180,000</td>
<td>1989</td>
<td>Intel</td>
<td>1 μm</td>
</tr>
<tr>
<td>Pentium</td>
<td>3,100,000</td>
<td>1993</td>
<td>Intel</td>
<td>0.8 μm</td>
</tr>
<tr>
<td>AMD K5</td>
<td>4,300,000</td>
<td>1996</td>
<td>AMD</td>
<td>0.5 μm</td>
</tr>
<tr>
<td>Pentium II</td>
<td>7,500,000</td>
<td>1997</td>
<td>Intel</td>
<td>0.35 μm</td>
</tr>
<tr>
<td>AMD K6</td>
<td>8,800,000</td>
<td>1997</td>
<td>AMD</td>
<td>0.35 μm</td>
</tr>
<tr>
<td>Pentium III</td>
<td>9,500,000</td>
<td>1999</td>
<td>Intel</td>
<td>0.25 μm</td>
</tr>
<tr>
<td>AMD K6-III</td>
<td>21,300,000</td>
<td>1999</td>
<td>AMD</td>
<td>0.25 μm</td>
</tr>
<tr>
<td>AMD K7</td>
<td>22,000,000</td>
<td>1999</td>
<td>AMD</td>
<td>0.25 μm</td>
</tr>
<tr>
<td>Pentium 4</td>
<td>42,000,000</td>
<td>2000</td>
<td>Intel</td>
<td>180 nm</td>
</tr>
<tr>
<td>Atom</td>
<td>47,000,000</td>
<td>2008</td>
<td>Intel</td>
<td>45 nm</td>
</tr>
<tr>
<td>Barton</td>
<td>54,300,000</td>
<td>2003</td>
<td>AMD</td>
<td>130 nm</td>
</tr>
<tr>
<td>AMD K8</td>
<td>105,900,000</td>
<td>2003</td>
<td>AMD</td>
<td>130 nm</td>
</tr>
<tr>
<td>Itanium 2</td>
<td>220,000,000</td>
<td>2003</td>
<td>Intel</td>
<td>130 nm</td>
</tr>
<tr>
<td>Cell</td>
<td>241,000,000</td>
<td>2006</td>
<td>Sony/</td>
<td>90 nm</td>
</tr>
<tr>
<td>Core 2 Duo</td>
<td>291,000,000</td>
<td>2006</td>
<td>Intel</td>
<td>65 nm</td>
</tr>
<tr>
<td>AMD K10</td>
<td>463,000,000</td>
<td>2007</td>
<td>AMD</td>
<td>65 nm</td>
</tr>
<tr>
<td>Itanium 2</td>
<td>592,000,000</td>
<td>2004</td>
<td>Intel</td>
<td>130 nm</td>
</tr>
<tr>
<td>Core i7 (Quad)</td>
<td>731,000,000</td>
<td>2008</td>
<td>Intel</td>
<td>45 nm</td>
</tr>
<tr>
<td>POWER6</td>
<td>789,000,000</td>
<td>2007</td>
<td>IBM</td>
<td>65 nm</td>
</tr>
<tr>
<td>6-Core Opteron</td>
<td>904,000,000</td>
<td>2009</td>
<td>AMD</td>
<td>45 nm</td>
</tr>
<tr>
<td>2-Core Itanium 2</td>
<td>1,700,000,000</td>
<td>2006</td>
<td>Intel</td>
<td>90 nm</td>
</tr>
<tr>
<td>6-Core Xeon 7400</td>
<td>1,900,000,000</td>
<td>2008</td>
<td>Intel</td>
<td>45 nm</td>
</tr>
<tr>
<td>4-Core Tukwila</td>
<td>2,000,000,000</td>
<td>future</td>
<td>Intel</td>
<td>65 nm</td>
</tr>
<tr>
<td>8-Core Nehalem-E</td>
<td>2,300,000,000</td>
<td>future</td>
<td>Intel</td>
<td>45 nm</td>
</tr>
</tbody>
</table>
Power

- Intel 80386 consumed ~ 2 W
- 3.3 GHz Intel Core i7 consumes 130 W
- Heat must be dissipated from 1.5 x 1.5 cm chip
- This is the limit of what can be cooled by air
Reducing Power

Techniques for reducing power:
- Do nothing well
- Dynamic Voltage-Frequency Scaling
- Low power state for DRAM, disks
- Overclocking, turning off cores
AMD Opteron DVS

![chart showing power consumption and DVS savings](image-url)
Trends in Cost

- Cost driven down by learning curve
  - Yield

- DRAM: price closely tracks cost

- Microprocessors: price depends on volume
  - 10% less for each doubling of volume
300 mm wafer (280 full dies)
Dependability

- Module reliability
  - Mean time to failure (MTTF)
  - Mean time to repair (MTTR)
  - Mean time between failures (MTBF) = MTTF + MTTR
  - Availability = MTTF / MTBF
Measuring Performance

- Typical performance metrics:
  - Response time
  - Throughput

- Speedup of X relative to Y
  - Execution time\(_Y\) / Execution time\(_X\)

- Execution time
  - Wall clock time: includes all system overheads
  - CPU time: only computation time

- Benchmarks
  - Kernels (e.g. matrix multiply)
  - Toy programs (e.g. sorting)
  - Synthetic benchmarks (e.g. Dhrystone)
  - Benchmark suites (e.g. SPEC06fp, TPC-C)
<table>
<thead>
<tr>
<th>SPEC2006 benchmark description</th>
<th>SPEC2006</th>
<th>SPEC2000</th>
<th>SPEC95</th>
<th>SPEC92</th>
<th>SPEC89</th>
</tr>
</thead>
<tbody>
<tr>
<td>GNU C compiler</td>
<td>gcc</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interpreted string processing</td>
<td>perf</td>
<td></td>
<td></td>
<td></td>
<td>espresso</td>
</tr>
<tr>
<td>Combinatorial optimization</td>
<td>mcf</td>
<td></td>
<td></td>
<td></td>
<td>li</td>
</tr>
<tr>
<td>Block-sorting compression</td>
<td>bzip2</td>
<td></td>
<td></td>
<td></td>
<td>egltott</td>
</tr>
<tr>
<td>Go game (AI)</td>
<td>go</td>
<td>vortex</td>
<td></td>
<td></td>
<td>sc</td>
</tr>
<tr>
<td>Video compression</td>
<td>h264vvc</td>
<td>gzip</td>
<td></td>
<td></td>
<td>ljpng</td>
</tr>
<tr>
<td>Games/path finding</td>
<td>astar</td>
<td>eon</td>
<td></td>
<td></td>
<td>m88ksim</td>
</tr>
<tr>
<td>Search gene sequence</td>
<td>hmmer</td>
<td>twolf</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quantum computer simulation</td>
<td>libquantum</td>
<td>vortex</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Discrete event simulation library</td>
<td>omnetpp</td>
<td>vpr</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chess game (AI)</td>
<td>sierr</td>
<td>crafty</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XML parsing</td>
<td>xalanbmk</td>
<td>parser</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CFD/blast waves</td>
<td>bwaves</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Numerical relativity</td>
<td>cactusADM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Finite element code</td>
<td>calculix</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Differential equation solver framework</td>
<td>dealll</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quantum chemistry</td>
<td>genomes</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EM solver (freq/time domain)</td>
<td>GemsFDTD</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scalable molecular dynamics (NAMDA)</td>
<td>gromacs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lattice Boltzmann method (fluid/air flow)</td>
<td>lbm</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Large eddy simulation/turbulent CFD</td>
<td>LESlie3d</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lattice quantum chromodynamics</td>
<td>mic</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Molecular dynamics</td>
<td>namd</td>
<td>gaige</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Image ray tracing</td>
<td>povray</td>
<td>mesa</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sparse linear algebra</td>
<td>soplex</td>
<td>art</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Speech recognition</td>
<td>sphinx3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quantum chemistry/object oriented</td>
<td>tonto</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weather research and forecasting</td>
<td>wrf</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Magneto hydrodynamics (astrophysics)</td>
<td>zeusmp</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Copyright © 2012, Elsevier Inc. All rights reserved.
Performance/Power

Graph showing performance vs. power consumption for different target workloads. The graph compares the performance of 710 Intel 12 core, 815 AMD 24 core, and 815 AMD 48 core systems. The x-axis represents the target workload (%), while the y-axis shows the ssi.ops/watt. The graph indicates that as the workload decreases, the performance and power consumption decrease as well.
Principles of Computer Design

- Take Advantage of Parallelism
  - e.g. multiple processors, disks, memory banks, pipelining, multiple functional units

- Principle of Locality
  - Reuse of data and instructions

- Focus on the Common Case
  - Amdahl’s Law
    \[
    \text{Execution time}_{\text{new}} = \text{Execution time}_{\text{old}} \times \left( 1 - \frac{\text{Fraction}_{\text{enhanced}}}{\text{Speedup}_{\text{enhanced}}} \right) + \frac{\text{Fraction}_{\text{enhanced}}}{\text{Speedup}_{\text{enhanced}}} \\
    \text{Speedup}_{\text{overall}} = \frac{\text{Execution time}_{\text{old}}}{\text{Execution time}_{\text{new}}} = \frac{1}{\left( 1 - \frac{\text{Fraction}_{\text{enhanced}}}{\text{Speedup}_{\text{enhanced}}} \right) + \frac{\text{Fraction}_{\text{enhanced}}}{\text{Speedup}_{\text{enhanced}}}}
    \]
The Processor Performance Equation

\[
\text{CPU time} = \text{CPU clock cycles for a program} \times \text{Clock cycle time}
\]

\[
\text{CPU time} = \frac{\text{CPU clock cycles for a program}}{\text{Clock rate}}
\]

\[
\text{CPI} = \frac{\text{CPU clock cycles for a program}}{\text{Instruction count}}
\]

\[
\text{CPU time} = \text{Instruction count} \times \text{Cycles per instruction} \times \text{Clock cycle time}
\]
Different instruction types having different CPIs

\[ \text{CPU clock cycles} = \sum_{i=1}^{n} \text{IC}_i \times \text{CPI}_i \]

\[ \text{CPU time} = \left( \sum_{i=1}^{n} \text{IC}_i \times \text{CPI}_i \right) \times \text{Clock cycle time} \]
Fallacies

- Multiprocessors are a silver bullet
- Benchmarks remain valid indefinitely
- Peak performance tracks observed performance
- Disks practically never fail
Pitfalls

- Falling prey to Amdahl’s heartbreaking law
- Fault detection can lower availability