EECS 252 Graduate Computer Architecture

Lec 1 - Introduction

David Patterson
Electrical Engineering and Computer Sciences
University of California, Berkeley

http://www.eecs.berkeley.edu/~patterson
http://www-inst.eecs.berkeley.edu/~cs252

Outline

• The following slides are taken from the supplemental slides for the Hennessy & Patterson text (which I highly recommend)
• Overall material is at a grad level, but I have selected slides that appropriate to this class
Manufacturing Process

- Wafers contain multiple die (typically hundreds)
- As features size gets smaller, fabrication costs increase

Crossroads: Conventional Wisdom in Comp. Arch

- Old Conventional Wisdom: Power is free, Transistors expensive
- New Conventional Wisdom: “Power wall” Power expensive, Xtors free (Can put more on chip than can afford to turn on)
- Old CW: Sufficiently increasing Instruction Level Parallelism via compilers, innovation (Out-of-order, speculation, VLIW, …)
- New CW: “ILP wall” law of diminishing returns on more HW for ILP
- Old CW: Multiplies are slow, Memory access is fast
- New CW: “Memory wall” Memory slow, multiplies fast (200 clock cycles to DRAM memory, 4 clocks for multiply)
- Old CW: Uniprocessor performance 2X / 1.5 yrs
- New CW: Power Wall + ILP Wall + Memory Wall = Brick Wall
  - Uniprocessor performance now 2X / 5(?) yrs
  ⇒ Sea change in chip design: multiple “cores” (2X processors per chip / ~ 2 years)
  » More simpler processors are more power efficient
Crossroads: Uniprocessor Performance


- VAX: 25%/year 1978 to 1986
- RISC + x86: 52%/year 1986 to 2002
- RISC + x86: ??%/year 2002 to present

Sea Change in Chip Design

- Intel 4004 (1971): 4-bit processor, 2312 transistors, 0.4 MHz, 10 micron PMOS, 11 mm² chip

- RISC II (1983): 32-bit, 5 stage pipeline, 40,760 transistors, 3 MHz, 3 micron NMOS, 60 mm² chip

- 125 mm² chip, 0.065 micron CMOS = 2312 RISC II+FPU+Icache+Dcache
  - RISC II shrinks to ~ 0.02 mm² at 65 nm
  - Caches via DRAM or 1 transistor SRAM ([www.t-ram.com](http://www.t-ram.com))
  - Proximity Communication via capacitive coupling at > 1 TB/s? (Ivan Sutherland @ Sun / Berkeley)

- **Processor is the new transistor?**
Déjà vu all over again?

- Multiprocessors imminent in 1970s, ‘80s, ‘90s, …
- “… today’s processors … are nearing an impasse as technologies approach the speed of light…”
- Transputer was premature
  ⇒ Custom multiprocessors strove to lead uniprocessors
  ⇒ Procrastination rewarded: 2X seq. perf. / 1.5 years
- “We are dedicating all of our future product development to multicore designs. … This is a sea change in computing”
  Paul Otellini, President, Intel (2004)
- Difference is all microprocessor companies switch to multiprocessors (AMD, Intel, IBM, Sun; all new Apples 2 CPUs)
  ⇒ Procrastination penalized: 2X sequential perf. / 5 yrs
  ⇒ Biggest programming challenge: 1 to 2 CPUs

Problems with Sea Change

- Algorithms, Programming Languages, Compilers, Operating Systems, Architectures, Libraries, … not ready to supply Thread Level Parallelism or Data Level Parallelism for 1000 CPUs / chip,
- Architectures not ready for 1000 CPUs / chip
  - Unlike Instruction Level Parallelism, cannot be solved by just by computer architects and compiler writers alone, but also cannot be solved *without* participation of computer architects
**Instruction Set Architecture: Critical Interface**

- Properties of a good abstraction
  - Lasts through many generations (portability)
  - Used in many different ways (generality)
  - Provides convenient functionality to higher levels
  - Permits an efficient implementation at lower levels

**Levels of the Memory Hierarchy**

<table>
<thead>
<tr>
<th>Capacity</th>
<th>Access Time</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Registers</td>
<td>100s Bytes</td>
<td>300-500 ps (0.3-0.5 ns)</td>
</tr>
<tr>
<td>L1 and L2 Cache</td>
<td>10s-100s K Bytes</td>
<td>~1 ns - ~10 ns, $1000s/6Byte</td>
</tr>
<tr>
<td>Main Memory</td>
<td>6 Bytes</td>
<td>80ns-200ns, ~$100/6Byte</td>
</tr>
<tr>
<td>Disk</td>
<td>10s T Bytes, 10 ms (10,000,000 ns)</td>
<td>~$1/6Byte</td>
</tr>
<tr>
<td>Tape</td>
<td>infinite sec-min</td>
<td>~$1/6Byte</td>
</tr>
</tbody>
</table>

**Staging Unit**
- Upper Level
  - faster
  - cache cntl 32-64 bytes
  - cache cntl 64-128 bytes
  - OS 4K-8K bytes
  - user/operator Mbytes

**Lower Level**
- larger
**Moore’s Law: 2X transistors / “year”**

- “Cramming More Components onto Integrated Circuits”
  - Gordon Moore, Electronics, 1965
- # on transistors / cost-effective integrated circuit double every $N$ months ($12 \leq N \leq 24$)

**Latency Lags Bandwidth (for last ~20 years)**

- **Performance Milestones**
- **Disk:** 3600, 5400, 7200, 10000, 15000 RPM (8x, 143x)
  (latency = simple operation w/o contention
  BW = best-case)
Latency Lags Bandwidth (last ~20 years)

- **Performance Milestones**

  - **Memory Module**: 16bit plain DRAM, Page Mode DRAM, 32b, 64b, SDRAM, DDR SDRAM (4x,120x)
  - **Disk**: 3600, 5400, 7200, 10000, 15000 RPM (8x, 143x)

  \[
  \text{Relative Latency Improvement} = \frac{\text{Latency improvement}}{\text{Bandwidth improvement}}
  \]

  (latency = simple operation w/o contention, BW = best-case)

- **Ethernet**: 10Mb, 100Mb, 1000Mb, 10000 Mb/s (16x,1000x)

- **Memory Module**: 16bit plain DRAM, Page Mode DRAM, 32b, 64b, SDRAM, DDR SDRAM (4x,120x)

- **Disk**: 3600, 5400, 7200, 10000, 15000 RPM (8x, 143x)

  (latency = simple operation w/o contention, BW = best-case)
**CPUs: Archaic (Nostalgic) v. Modern (Newfangled)**

- **1982 Intel 80286**
  - 12.5 MHz
  - 2 MIPS (peak)
  - Latency 320 ns
  - 134,000 x tors, 47 mm²
  - 16-bit data bus, 68 pins
  - Microcode interpreter, separate FPU chip
  - (no caches)

- **2001 Intel Pentium 4**
  - 1500 MHz (120X)
  - 4500 MIPS (peak) (2250X)
  - Latency 15 ns (20X)
  - 42,000,000 x tors, 217 mm²
  - 64-bit data bus, 423 pins
  - 3-way superscalar,
    Dynamic translate to RISC,
    Superpipelined (22 stage),
    Out-of-Order execution
  - On-chip 8KB Data caches,
    96KB Instr. Trace cache,
    256KB L2 cache

---

**Latency Lags Bandwidth (last ~20 years)**

- **Performance Milestones**
  - **Processor**: ‘286, ‘386, ‘486, Pentium, Pentium Pro, Pentium 4 (21x,2250x)
  - Ethernet: 10Mb, 100Mb, 1000Mb, 10000 Mb/s (16x,1000x)
  - Memory Module: 16bit plain DRAM, Page Mode DRAM, 32b, 64b, SDRAM, DDR SDRAM (4x,120x)
  - Disk: 3600, 5400, 7200, 10000, 15000 RPM (8x, 143x)
Reasons Latency Lags Bandwidth

1. Moore’s Law helps BW more than latency
   - Faster transistors, more transistors, more pins help Bandwidth
     » MPU Transistors: 0.130 vs. 42 M xtors (300X)
     » DRAM Transistors: 0.064 vs. 256 M xtors (4000X)
     » MPU Pins: 68 vs. 423 pins (6X)
     » DRAM Pins: 16 vs. 66 pins (4X)
   - Smaller, faster transistors but communicate over (relatively) longer lines: limits latency
     » Feature size: 1.5 to 3 vs. 0.18 micron (8X,17X)
     » MPU Die Size: 35 vs. 204 mm² (ratio sqrt ⇒ 2X)
     » DRAM Die Size: 47 vs. 217 mm² (ratio sqrt ⇒ 2X)

2. Distance limits latency
   - Size of DRAM block ⇒ long bit and word lines ⇒ most of DRAM access time
   - Speed of light and computers on network
   - 1. & 2. explains linear latency vs. square BW?

3. Bandwidth easier to sell (“bigger=better”)
   - E.g., 10 Gbits/s Ethernet (“10 Gig”) vs. 10 μsec latency Ethernet
   - 4400 MB/s DIMM (“PC4400”) vs. 50 ns latency
   - Even if just marketing, customers now trained
   - Since bandwidth sells, more resources thrown at bandwidth, which further tips the balance
Define and quantity power (1/2)

- For CMOS chips, traditional dominant energy consumption has been in switching transistors, called dynamic power.

\[ \text{Power}_{\text{dynamic}} = \frac{1}{2} \times \text{CapacitiveLoad} \times \text{Voltage}^2 \times \text{Frequency}_{\text{Switched}} \]

- For mobile devices, energy better metric

\[ \text{Energy}_{\text{dynamic}} = \text{CapacitiveLoad} \times \text{Voltage}^2 \]

- For a fixed task, slowing clock rate (frequency switched) reduces power, but not energy.

- Capacitive load a function of number of transistors connected to output and technology, which determines capacitance of wires and transistors.

- Dropping voltage helps both, so went from 5V to 1V.

- To save energy & dynamic power, most CPUs now turn off clock of inactive modules (e.g. Fl. Pt. Unit).