What is Computer Architecture?

In its broadest definition, computer architecture is the design of the abstraction layers that allow us to implement information processing applications efficiently using available manufacturing technologies.

Abstraction Layers in Modern Systems

<table>
<thead>
<tr>
<th>Application</th>
<th>Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programming Language</td>
<td>Operating System/Virtual Machine</td>
</tr>
<tr>
<td>Instruction Set Architecture (ISA)</td>
<td>Microarchitecture</td>
</tr>
<tr>
<td>Gates/Register-Transfer Level (RTL)</td>
<td>Circuits</td>
</tr>
<tr>
<td>Devices</td>
<td>Physics</td>
</tr>
</tbody>
</table>

Parallel computing, security, …

Domain of more recent computer architecture (‘90s)

Reliability, power, …

Reinvigoration of computer architecture, mid-2000s onward.

Computer Architecture: Changing Definition

- 1950s to 1960s
  - Computer Arithmetic
- 1970s to mid 1980s
  - Instruction Set Design, especially ISA appropriate for compilers
- 1990s
  - Design of CPU, memory system, I/O system, Multiprocessors, Networks
- 2000s
  - Multi-core design, on-chip networking, parallel programming paradigms, power reduction
- 2010s
  - Self adapting systems? Self organizing structures?
Sea Change in Chip Design

- Intel 4004 (1971): 4-bit processor, 2312 transistors, 0.4 MHz, 10 µm pMOS, 11 mm² chip
- RISC II (1983): 32-bit, 5 stage pipeline, 40,760 transistors, 3 MHz, 3 µm nMOS, 60 mm² chip (6 x 10 mm)
- As of 2006: 125 mm² chip, 0.065 µ CMOS = 2312 RISC II + FPU + I-cache + D-cache (~ 100M transistors)
  - RISC II shrinks to ~ 0.02 mm² at 65 nm

Processor is the new transistor?

Technology constantly on the move!

- # of transistors is not limiting factor
  - Currently, ~ 2 billion transistors/chip
  - Problems:
    - Too much Power, Heat, Latency
    - Not enough Parallelism
- 3-dimensional chip technology?
  - Sandwiches of silicon
  - “Through-Vias” for communication
- On-chip optical connections?
  - Power savings for large packets
- The Intel® Core™ i7 (“Nehalem”)
  - 4 cores/chip
  - 45 nm, Hafnium hi-k dielectric
  - 731 million transistors
  - Shared L3 Cache - 8MB
  - L2 Cache - 1MB (256K x 4)

Conventional Wisdom in Comp. Arch

- Old CW: Power is free, transistors expensive
  - New CW: Power wall - Power expensive, transistors free (Can put more on chip than can afford to turn on)
- Old CW: Can increase Instruction Level Parallelism more 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 access 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)
    - Increase on-chip number of simple processors that are power efficient
    - Simple processor “cores” use less power per useful calculation done

Single Processor Performance

Move to multi-processor

- VAX: 25%/year 1978 to 1986
- RISC + x86: 52%/year 1986 to 2003
- RISC + x86: ~20%/year 2003 to 2010
The End of the Uniprocessor Era

Single biggest change in the history of computing systems

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
  ⇒ 2X seq. perf. / 1.5 years
- “We are dedicating all of our future product development to multi-core designs. … This is a sea change in computing”
  - Paul Otellini, President, Intel (2004)
- Difference is all microprocessor companies switch to multi-core (AMD, Intel, IBM; all new Apples 2-4 CPUs)
  ⇒ 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
- Shift from Instruction Level Parallelism to Thread Level Parallelism / Data Level Parallelism

Many Core Chips: The future is here

- Intel 80-core multi-core chip (Feb 2007)
  - 80 simple cores
  - Two FP-engines / core
  - Mesh-like network
  - 100 million transistors
  - 65nm feature size
  - 24 “tiles” with two 1X cores per tile
  - 24-router mesh network with 256 Gb/s bisection
  - 4 integrated DDR3 memory controllers
  - Hardware support for message-passing
- “Many Core” refers to many processors/chip
  - 64? 128? Hard to say exact boundary
- How to program these?
  - Use 2 CPUs for video/audio
  - Use 1 for word processor, 1 for browser
  - 76 for virus checking??
- Something new is clearly needed here...
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
- What matters today is performance of complete computer systems

Example: MIPS R3000

<table>
<thead>
<tr>
<th>r0</th>
<th>Programmable storage</th>
<th>Data types?</th>
</tr>
</thead>
<tbody>
<tr>
<td>r1</td>
<td>2^32 x bytes</td>
<td>Format?</td>
</tr>
<tr>
<td></td>
<td>31 x 32-bit GPRs (R0=0)</td>
<td>Addressing Modes?</td>
</tr>
<tr>
<td>r31</td>
<td>32 x 32-bit FP regs (paired DP)</td>
<td></td>
</tr>
<tr>
<td>PC</td>
<td>Hi, LO, PC</td>
<td></td>
</tr>
</tbody>
</table>

Arithmetic logical

- Add, AddU, Sub, SubU, And, Or, Xor, Nor, SLT, SLTU, AddI, AddIU, SLTI, SLTIU, AndI, OrI, Xori, LUI
- SLL, SRL, SRA, SLLV, SRLV, SRAV

Memory Access

- LB, LBU, LH, LHU, LW, LWL, LWR
- SB, SH, SW, SWL, SWR

Control

- J, JAL, JR, JALR
- BEQ, BNE, BGEZ, BLTZ, BGEZAL, BLTZAL