
CMSC 411

Computer Systems Architecture

Lecture 1

Computer Architecture at Crossroads

Slides from Alan Sussman, Pete Keleher...

CMSC 411 - 1

2

Administrivia

- Class web page
 - <http://www.cs.umd.edu/class/fall2009/cmssc411>
 - Linked in from CS dept class web pages
- Class accounts
 - CSIC Linux cluster
- Class textbook
 - Hennessy & Patterson, Computer Architecture: A Quantitative Approach, 4th Edition
 - Start reading Chapter 1

Introduction

- Why are you taking this course?
 - You really liked the material in 311 and want to learn more?
 - The course time fit into your schedule well?
 - You needed upper level CS courses and chose this one at random?
 - All the courses you really wanted to take were filled?

CMSC 411 - 1

3

What can you expect to learn?

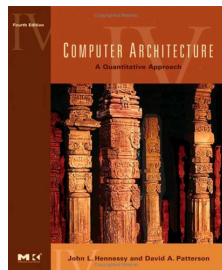
- What to look for in buying a PC
 - Can brag to parents and friends!
- How computer architecture affects programming style
- How programming style affect computer architecture
- How processors/disks/memory work
- How processors exploit instruction/thread parallelism
- A great deal of jargon

CMSC 411 - 1

4

The Textbook – H&P

- Everyone complains about it
- Virtually everyone uses it
- You can handle it, but you have to work at it – *do the reading*
- Through lecture notes, other references, etc., I'll try to help you put it all together



CMSC 411 - 1

5

Chapter 1 of H&P

- Read Chapter 1
- Historical Perspective - Section 1.13
 - Computers as we know them are roughly 60 years old
 - The *von Neumann machine* model that underlies computer design is only partially von Neumann's
 - Konrad Zuse say he had "the bad luck of being too early"
 - » Optional: Read his own recollections in TR 180 of ETH, Zürich,
<http://www.inf.ethz.ch/research/disstechreps/techreports/show?serial=180&lang=en> (contains both German and English)
 - No one was able to successfully patent the idea of a stored-program computer, much to the dismay of Eckert and Mauchly

CMSC 411 - 1

6

Early development steps

- Make input and output easier than wiring circuit boards and reading lights
- Make programming easier by developing higher level programming languages, so that users did not need to use binary machine code instructions
 - First compilers in late 1950's, for Fortran and Cobol
- Develop storage devices

CMSC 411 - 1

7

Later development steps

- Faster
- More storage
- Cheaper
- Networking and parallel computing
- Better user interfaces
- Ubiquitous applications
- Development of standards

CMSC 411 - 1

8

Perspective: An example

- Most powerful computer in 1988: CRAY Y-MP
- 1993: a desktop workstation (IBM Power-2) matched its power at less than 10% of the cost
- How did this happen?
 - hardware improvements, e.g., squeezing more circuits into a smaller area
 - improvements in instruction-set design, e.g., making the machine faster on a small number of frequently used instructions
 - improvements in compilation, e.g., optimizing code to reduce memory accesses and make use of faster machine instructions

CMSC 411 - 1

9

COMPUTER ARCHITECTURE AT A CROSSROADS

CMSC 411 - 1

10

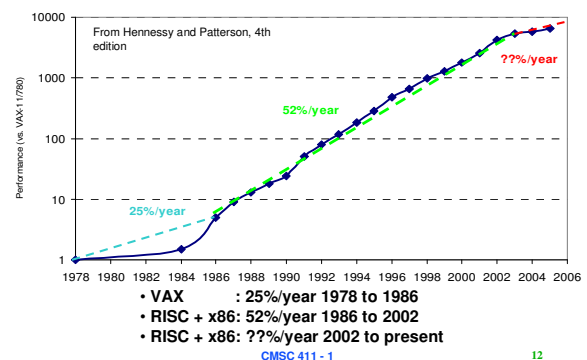
Crossroads: Conventional Wisdom in Comp. Arch

- Old Conventional Wisdom: Power is free, Transistors expensive
 - New Conventional Wisdom: "Power wall" Power expensive, transistors free (Can put more on chip than can afford to turn on)
 - Old CW: Sufficiently increasing Instruction Level Parallelism (ILP) 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

CMSC 411 - 1

11

Crossroads: Uniprocessor Performance

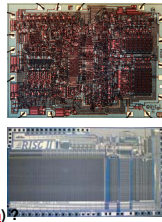


CMSC 411 - 1

12

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)
 - Proximity Communication via capacitive coupling at > 1 TB/s ? (Ivan Sutherland @ Sun / Berkeley)



• Processor is the new transistor?

CMSC 411 - 1

13

Multiprocessors - 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."
 - David Mitchell, *The Transputer: The Time Is Now* (1989)
- 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

CMSC 411 - 1

14

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 just by computer architects and compiler writers alone, but also cannot be solved *without* participation of computer architects
- This 4th Edition of textbook *Computer Architecture: A Quantitative Approach* explores shift from Instruction Level Parallelism to Thread Level Parallelism / Data Level Parallelism

CMSC 411 - 1

15