CS 352: Computer Systems Architecture

Lecture 1:
What is Computer Architecture?

January 17, 2003

Kathryn S McKinley
Professor of Computer Science
University of Texas at Austin
mckinley@cs.utexas.edu

The simple view

All a computer does is

- Store and move data
- Communicate with the external world
- Do these two things conditionally
- According to a recipe specified by a programmer

Questions we will address in this course

- How do we separate software from hardware?
  - So that new computers can run old software
- How is computer hardware organized?
  - Processor, Memory, I/O, etc.
- How is the processor organized? Why?
- How do we measure and improve computer performance?
- How do we think about parallelism?
  - Doing more than one thing at once

Logistics

Lectures T/Th 9:30-11am, NOA 1.126

Instructor Prof. Kathryn S McKinley,
Office Hours: Tu 1:30-2:30 & by appointment

TA Renee St. Amant
Office Hours: M,W 1:30-2:30

Grading

Final Exam	1	15%
Midterm Exam	2	15% each
Homework	~7	20%
Quizzes	~10	10%
Project	1	25%

Ethics

If you cheat, you fail

Text

Patterson & Hennessy, *Computer Organization and Design* (Fourth Edition). Including CD.

CS352 Online

URL: www.cs.utexas.edu/users/mckinley/352

email list: 352-mckinley@cs.utexas.edu

mandatory: send email with your name and email to stamant@cs.utexas.edu

Computer Architecture Seminar Series:

www.cs.utexas.edu/users/cart/arch

CS352 Topics

- What is a computer system?
- Technology Trends
- Computer Performance
- Instruction set architectures
- Pipelining
- Modern pipelined architectures
  - Dynamic ILP machines
  - Static ILP machines
- Cache memory systems
- Virtual memory
- Multiprocessors
- Computer system implementation

What is a Computer System?

Specification — compute the fibonacci sequence

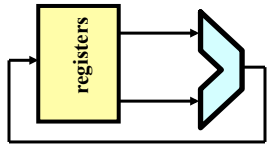
Program

```
for(i=2; i<100; i++) {
  a[i] = a[i-1]+a[i-2];}
```

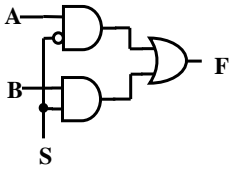
ISA (Instruction Set Architecture)

```
load r1, a[i];
add  r2, r2, r1;
```

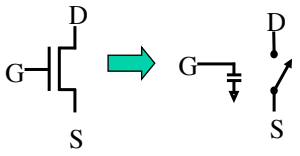
Microarchitecture



Logic

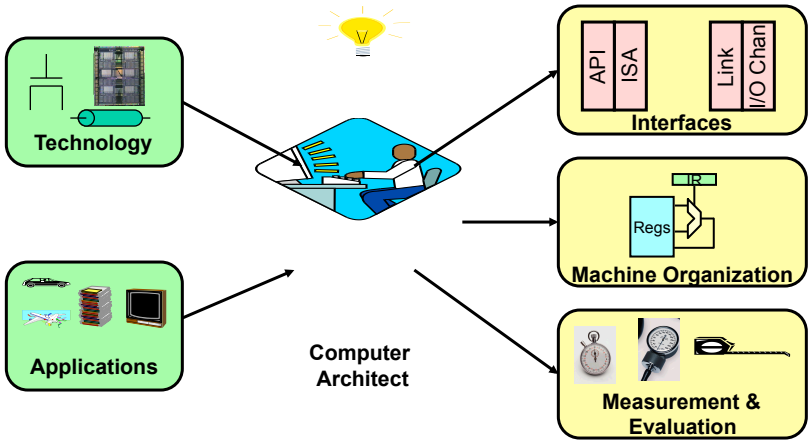


Transistors

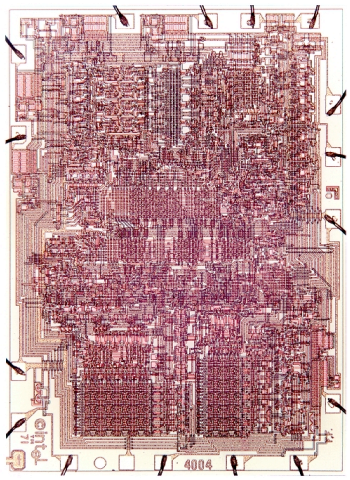


Physics/Chemistry

What is Computer Architecture?

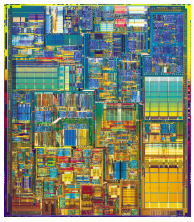


Intel 4004 - 1971

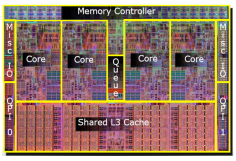


- The first microprocessor
- 2,300 transistors
- 108 KHz
- 10μm process

Some Recent Chips!

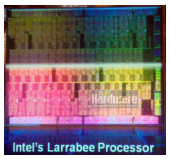


Intel Pentium IV
- 1 CPU 2005
- 42 million transistors
- 4 GHz
- 0.13μm process
- Fits ~15,000 4004s!

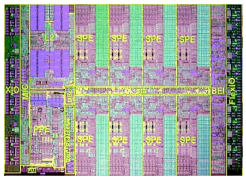


Intel i7
- 4 CPUs 2008
- 731 million transistors
- 3 GHz
- 0.045μm (45nm) process
- Fits ~300,000 4004s

NVidia - GeForce 6800
- 2006
- 222 million transistors
- 400 MHz
- 0.13μm process



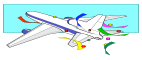
Intel- Larabee
- 2009
- 1.7 billion transistors
- 1 GHz
- 45nm process

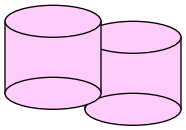


IBM Cell
- 2008
- 8 vector processors + 1 PPC
- 4 GHz
- 90nm process

Many kinds of systems and applications

- Personal:
  - Desktop, Laptop
  - Cell phone / PDA
  - Game machine
- Server:
  - Web servers
  - Transaction processing
- Engineering/Scientific:
  - Weather simulation
  - Drug design
- Embedded Control:
  - Anti-lock brake system
  - Microwave oven





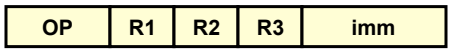
What is an "interface"

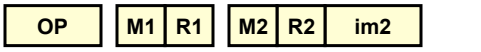
- *Interfaces* are visible, *Implementations* are not
  - Same interface can have multiple implementations
  - We allow performance (time behavior) to change!
- Example interfaces:
  - Ethernet connector / protocol
  - X86 Instruction Set Architecture (ISA)
  - Operating System Application Program Interface (API)
  - C/Java language
- Examples that are NOT interfaces
  - Power connector for cell phone charger
- Good interfaces are simple

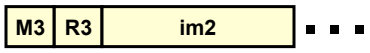
Instruction-Set Architecture (ISA)

Hardware/Software Interface

- Software impact
  - support OS functions
    - restart instructions
    - memory relocation and protection
  - a good compiler target
    - simple
    - orthogonal
  - dense
- Hardware impact
  - admits efficient implementation
    - across multiple hardware generations
  - admits parallel implementation
    - no 'serial' bottlenecks
- Abstraction without interpretation

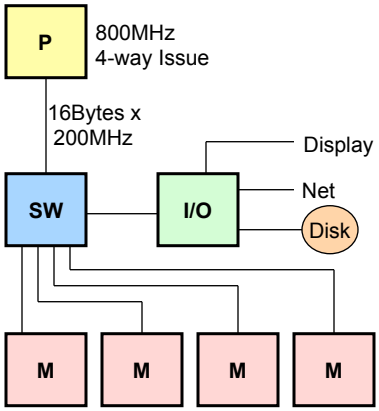






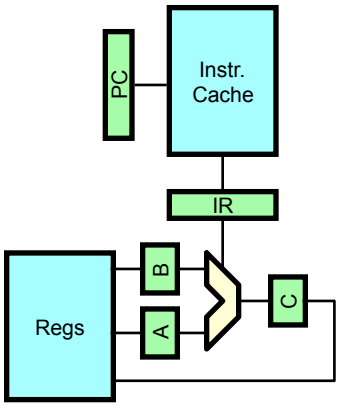
System-Level Organization

- Design at the level of processors, memories, and interconnect.
- More important to application performance than CPU design
- Feeds and speeds
  - constrained by inter-connect pin count, module pin count, and signaling rates
- System balance
  - for a particular application
- Driven by
  - performance/cost goals
  - available components (cost/perf)
  - technology constraints

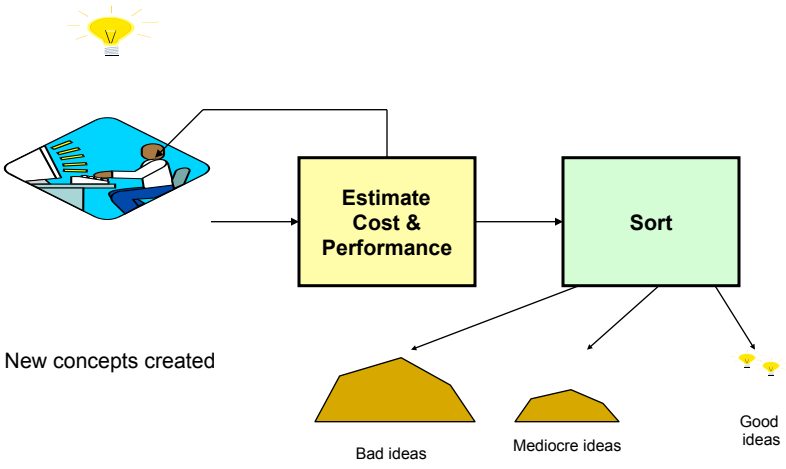


Microarchitecture

- Register-transfer-level (RTL) design
- Implements the ISA
- Exploit capabilities of technology
  - locality and concurrency
- Iterative process
  - generate proposed architecture
  - estimate cost
  - measure performance
- Emphasis is on overcoming sequential nature of programs
  - deep pipelining
  - multiple issue
  - dynamic scheduling
  - branch prediction/speculation
  - speculative parallelism



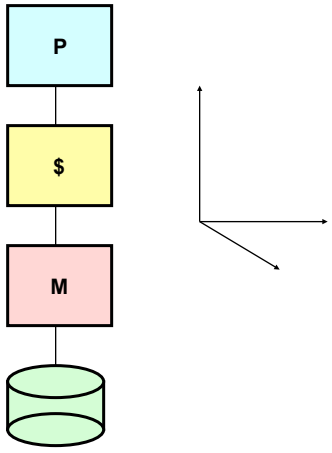
The Architecture Process



Performance Measurement and Evaluation

Many Dimensions to Performance

- CPU execution time
  - by instruction or sequence
    - floating point
    - integer
    - branch performance
- Cache bandwidth
- Main memory bandwidth
- I/O performance
  - bandwidth
  - seeks
  - pixels or polygons per second
- Relative importance depends on applications

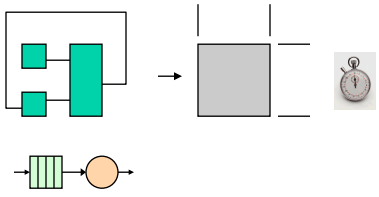


Evaluation Tools

- Benchmarks, traces, & mixes
  - macrobenchmarks & suites
    - application execution time
  - microbenchmarks
    - measure one aspect of performance
  - traces
    - replay recorded accesses
    - cache, branch, register
- Simulation at many levels
  - ISA, cycle accurate, RTL, gate, circuit
    - trade fidelity for simulation rate
- Area and delay estimation
- Analysis
  - e.g., queuing theory

MOVE	39%
BR	20%
LOAD	20%
STORE	10%
ALU	11%

```
LD 5EA3
ST 31FF
....
LD 1EA2
....
```



Don't forget the simple view

All a computer does is
- Store and move data
- Communicate with the external world
- Do these two things conditionally
- According to a recipe specified by a programmer

It's complex because
- We want it to be fast
- We want it to be reliable and secure
- We want it to be simple to use
- It must obey the laws of physics

Next Time

- Basic computer elements
  - transistors, wires, memory
- How chips are made
- Technology trends

- Reading assignment
  - P&H Chapter 1.1—3, 1.7-9
- More on transistors (optional)
  - http://en.wikipedia.org/wiki/Transistor