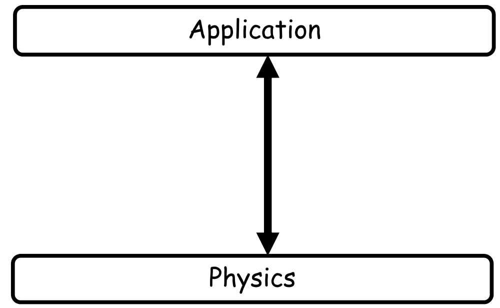
CS740

Overview
September 12, 2014

- Topics
  - What is Computer Architecture
  - Underlying Technology
  - Information about the class
  - History

What is computer architecture?

- The science and art of selecting and interconnecting hardware components to create computers that meet functional, performance and cost goals. [wikipedia]
- Abstractions to bridge gap



Responsive to Technology Changes

- Underlying components:
  - Relays → Tubes → Transistors → VLSI → ?
  - Magnetic core → SRAM → DRAM → FLASH → ?
- What to optimize for:
  - Transistors
  - Memory
  - Instructions
  - Power
  - Parallelism
- Technology constantly changing

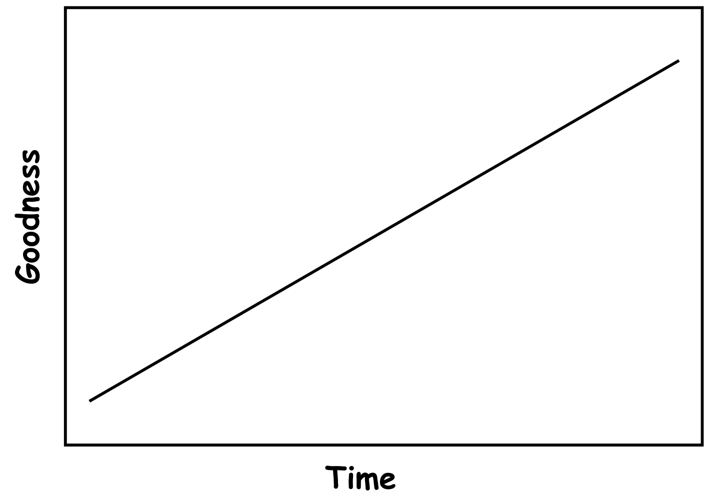
Responsive to Applications

Course Constantly Changing

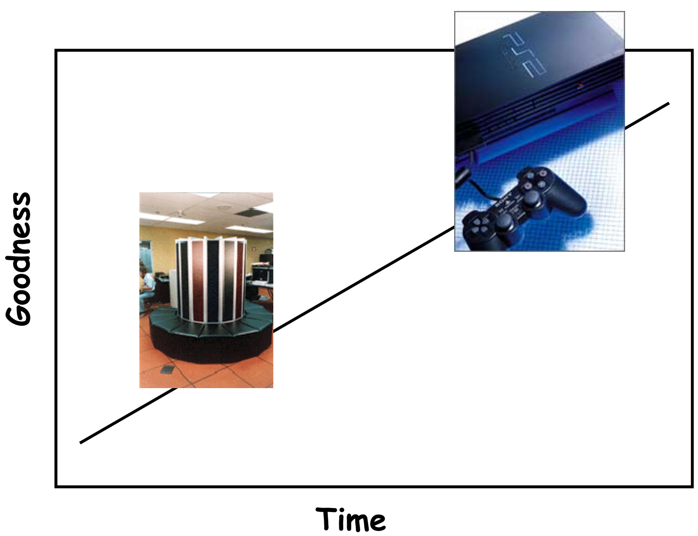
- As technology and application space change, so do the focus of computer architecture:

- Computer Arithmetic
- Instruction Set Architecture
- CPU Design
- Memory System, I/O, Networks
- Power, Multicore

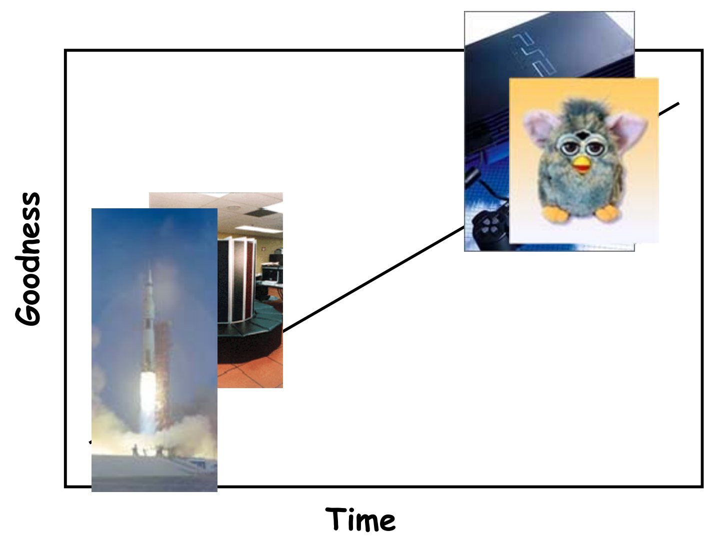
Moore's Law



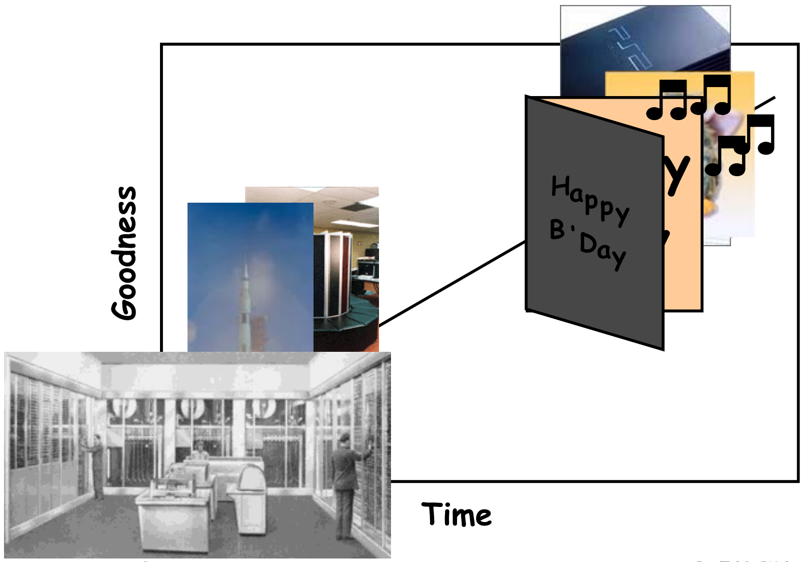
Moore's Law



Moore's Law

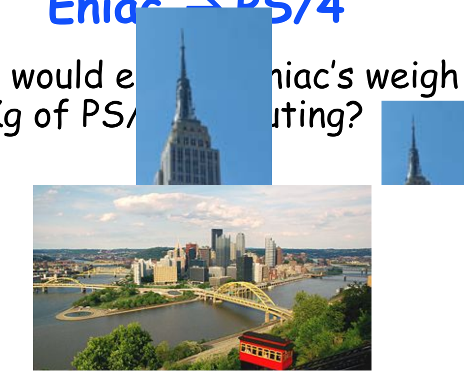


Moore's Law



Eniac → PS/4

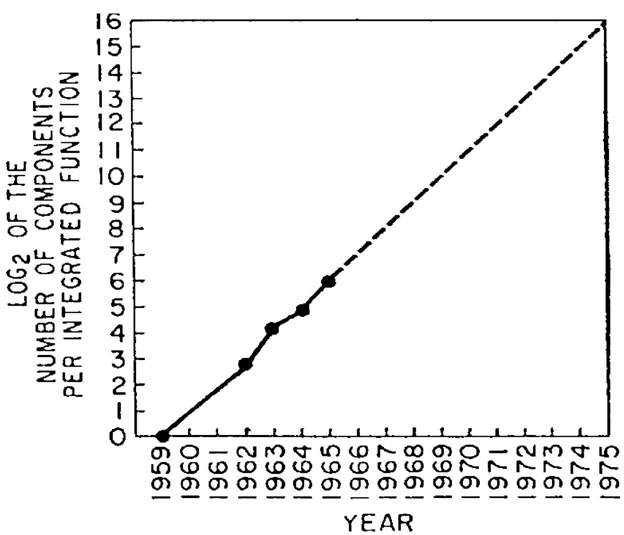
- How much would e niac's weigh to equal 2.8Kg of PS/ uting?



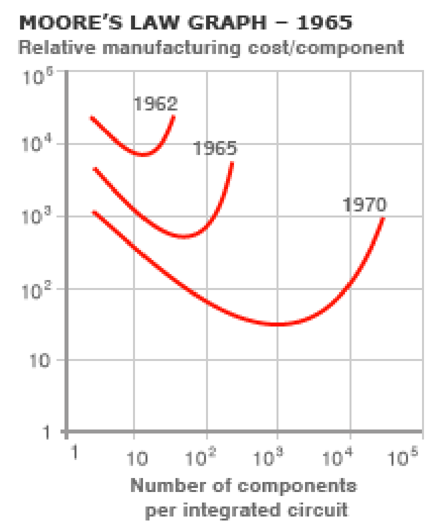
Alternatively, more than all the buildings in Pittsburgh!

s!

Moore's Law - 1965



Essential Argument - Economics?

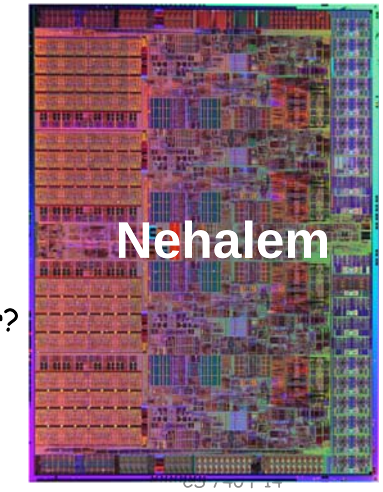


Technology Trends

- It isn't just transistor density
  - Transistor size, density, speed, power, cost
  - Memory size, density, latency, throughput
  - Disks
  - Networks
  - Communication
- These trends lead to exponential increse in ops/sec-\$-m^3-watt
- Which in turn leads to changes in applications Mainframes → Desktops → Mobile
- Which leads to new design goals

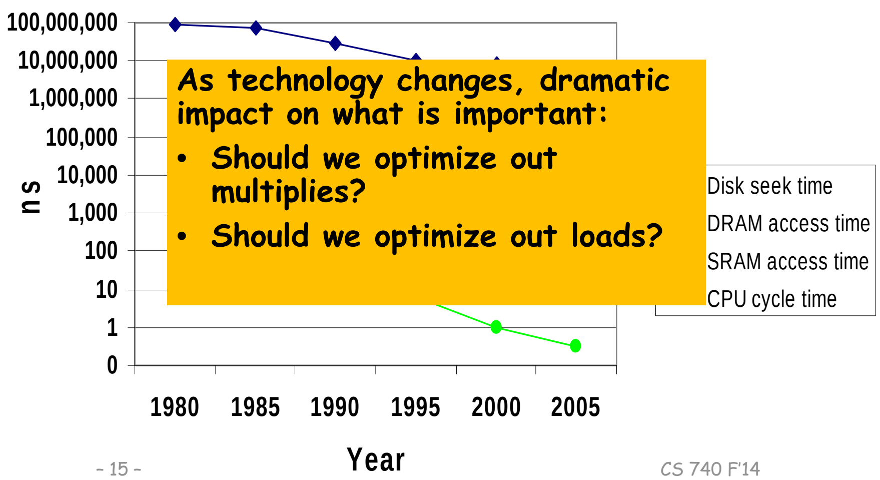
Technology constantly on the move!

- Not Optimizing for Num of transistors
  - Currently > 1 billion transistors/chip
- Issues:
  - Complexity
  - Power
  - Heat
  - Latency
  - Parallelism
- Huge Change in thinking
  - Improve ILP or decrease power?



The CPU-Memory Gap

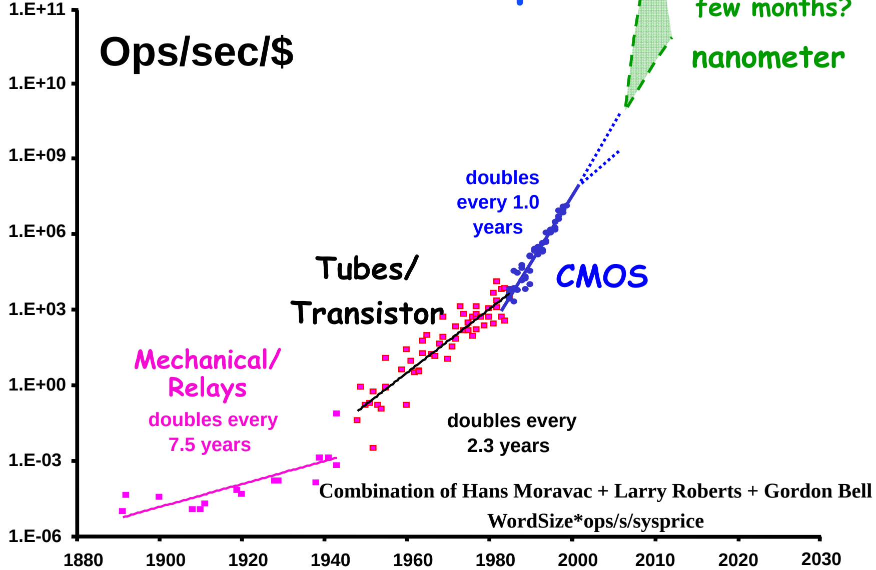
The gap widens between DRAM, disk, and CPU speeds.



Why Study Computer Architecture

- Understand how computers work
- Understand computer performance
- Its not just how to build them:
  - Why does my program run slowly?
  - How do I increase performance?
  - How do I improve reliability?
  - What can I expect tomorrow?
- We are at a crossroads

A Historical Perspective



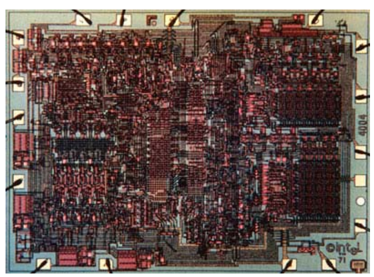
[Gray Turing Award Lecture]

The Microprocessor

- **Microprocessor revolution**
  - One significant technology threshold was crossed in 1970s
  - Enough transistors (~25K) to put a 16-bit processor on one chip
  - Huge performance advantages: fewer slow chip-crossings
  - Even bigger cost advantages: one "stamped-out" component
- Create New Applications
  - Desktops, CD/DVD players, laptops, game consoles, set-top boxes, mobile phones, digital camera, mp3 players, GPS, automotive
- And replaced incumbents in existing segments
  - Microprocessor-based system replaced supercomputers, "mainframes", "minicomputers", etc.

First Microprocessor

- Intel 4004 (1971)
  - Application: calculators
  - Technology: 10000 nm
  - 2300 transistors
  - 13 mm^2
  - 108 KHz
  - 12 Volts
  - 4-bit data
  - Single-cycle datapath



Tracing the Microprocessor Revolution

- How were growing transistor counts used?
- Initially to widen the datapath
  - 4004: 4 bits → Pentium4: 64 bits
- … and also to add more powerful instructions
  - To amortize overhead of fetch and decode
  - To simplify programming (which was done by hand then)

Implicit Parallelism

- Then to **extract implicit instruction-level parallelism (ILP)**
  - Hardware provides parallel resources, figures out how to use them
  - Software is oblivious
- Initially using pipelining …
  - Which also enabled increased clock frequency
- … caches …
  - Which became necessary as processor clock frequency increased
- … and integrated floating-point
- Then deeper pipelines and branch speculation
- Then multiple instructions per cycle (superscalar)
- Then dynamic scheduling (out-of-order execution)

Near End of 1-Core Microprocessors

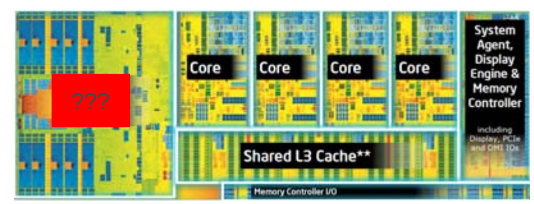
- Intel Pentium4 (2003)
  - Application: desktop/server
  - Technology: 90nm (1% of 4004)
  - 55M transistors (20,000x)
  - 101 mm^2 (10x)
  - 3.4 GHz (10,000x)
  - 1.2 Volts (1/10x)
  - 32/64-bit data (16x)
  - 22-stage pipelined datapath
  - 3 instructions per cycle (superscalar)
  - Two levels of on-chip cache
  - data-parallel vector (SIMD) instructions, hyperthreading

Explicit Parallelism

- Then to support **explicit data & thread level parallelism**
  - Hardware provides parallel resources, software specifies usage
  - Why? diminishing returns on instruction-level-parallelism
- First using (subword) vector instructions…, Intel's SSE
  - One instruction does four parallel multiplies
- … and general support for multi-threaded programs
  - Coherent caches, hardware synchronization primitives
- Then using support for multiple concurrent threads on chip
  - First with single-core multi-threading, now with multi-core
- Graphics processing units (GPUs) are highly parallel
  - Converging with general-purpose processors (CPUs)

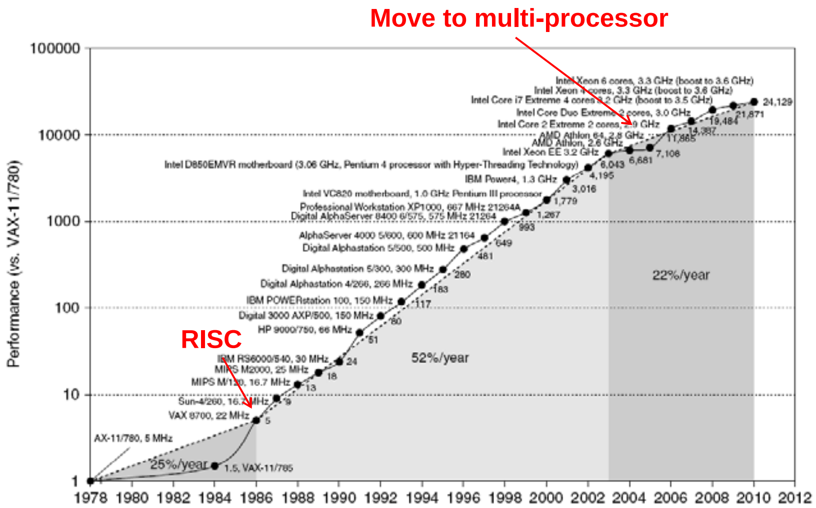
Modern Multicore Processor

- Intel Core i7 (2013)
  - Application: desktop/server
  - Technology: 22nm (25% of P4)
  - 1.4B transistors (30x)
  - 177 mm^2 (2x)
  - 3.5 GHz to 3.9 Ghz (~1x)
  - 1.8 Volts (~1x)



  - 256-bit data (2x)
  - 14-stage pipelined datapath (0.5x)
  - 4 instructions per cycle (1x)
  - Three levels of on-chip cache
  - data-parallel vector (SIMD) instructions, hyperthreading
  - **Four-core multicore** (4x)

Performance



What Computer Architects Do

- Given Constraints of:
  - Technology
  - Application
- Use Essential Themes:
  - Exploit locality (AKA caching)
  - Prediction / Speculation
  - Pipelining
  - Parallelism
  - Virtualization / Indirection
  - Specialization
- And, always using abstraction

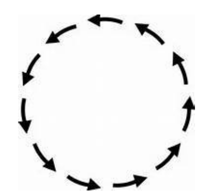
 [Mudge'14]

Abstraction Layers in Modern Systems

Application
Algorithm
Programming Language
Operating System/Virtual Machine
Instruction Set Architecture (ISA)
Microarchitecture
Gates/Register-Transfer Level (RTL)
Circuits
Devices
Physics

What Computer Architects Do

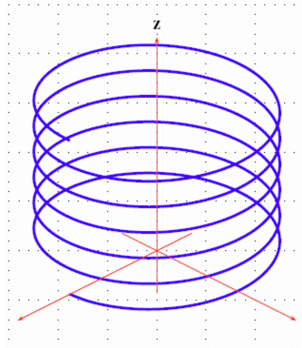
- Given Constraints of:
  - Technology
  - Application
- Use Essential Themes:
  - Exploit locality (AKA caching)
  - Prediction / Speculation
  - Pipelining
  - Parallelism
  - Virtualization / Indirection
  - Specialization
- Often seems like going in circles



 [Mudge'14]

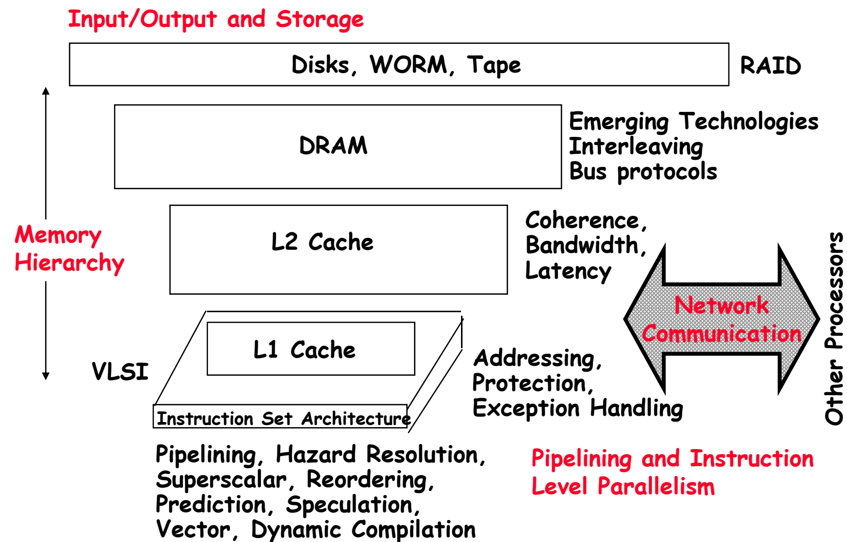
What Computer Architects Do

- Given Constraints of:
  - Technology
  - Application
- Use Essential Themes:
  - Exploit locality (AKA caching)
  - Prediction / Speculation
  - Pipelining
  - Parallelism
  - Virtualization / Indirection
  - Specialization
- But, there is progress

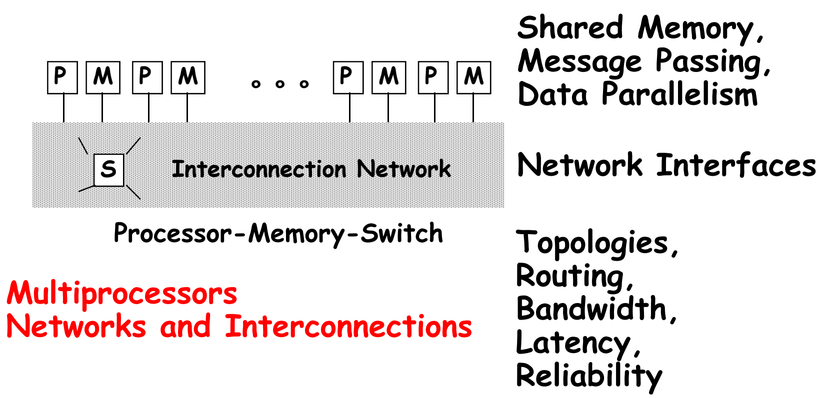


[Mudge'14]

Computer Architecture Topics



Computer Architecture Topics



The Course Logistics

- Lectures
- Paper Readings & Reviews
- Paper Presentations
- Labs
- Project
- Exams

Lectures

- Please come
- Please come prepared
- Participation

Papers

- No required text
- Required readings
- Reviews/Summaries
  - This is very important
    - Your grade
    - But, more importantly, an essential skill
    - Submit before class
  - Review contents
    - Identify essential content
    - Understand context
    - At most half a page

Paper Summaries

- Identify Essential (good) Idea
- Goal of paper
- Relationship to other papers/ideas
- What questions does it raise?
- Methodology
- What are the conclusions

- Submit a pdf before class

Paper Presentations

- Logisitics:
  - Pick a topic by 9/24
  - Groups of 2
  - 20 minute presentation
  - Submit powerpoint before class
- Presentation
  - Background question/problem they are investigating
  - What are the good ideas?
  - How do they come to their conclusions
  - Some follow on ideas

Labs

- There will be 2 labs
- Work in groups of 2 or 3
- Goal:
  - Become familiar with tools
  - Understand performance measurement
  - Understand optimization aka How architecture affects use

Project

- Do some real research
- Work in groups of 2 (or possibly 3)
- Lectures are front-loaded
- Timeline:
  - Proposal
  - Milestone
  - Poster
  - Paper

Exams

- 2 in-class exams