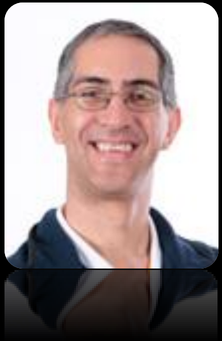


UC Berkeley
Teaching Professor
Dan Garcia

CS61C

Great Ideas in Computer Architecture
(a.k.a. Machine Structures)

Welcome, everyone!! Course Introduction

A Janet Jackson Song Could Crash Windows XP Laptops!

…a sound frequency in Janet Jackson's song "Rhythm Nation" could crash a model 5400rpm laptop hard drive used in certain Windows XP notebooks… "Playing the music video on one laptop caused a laptop sitting nearby to crash, even though that other laptop wasn't playing the video!" Microsoft determined the song had a frequency that matched the laptop hard drive's natural resonant frequency, which caused its moving disks to over-vibrate and induce a crash..

www.pcmag.com/news/a-janet-jackson-song-could-crash-windows-xp-laptops

Agenda

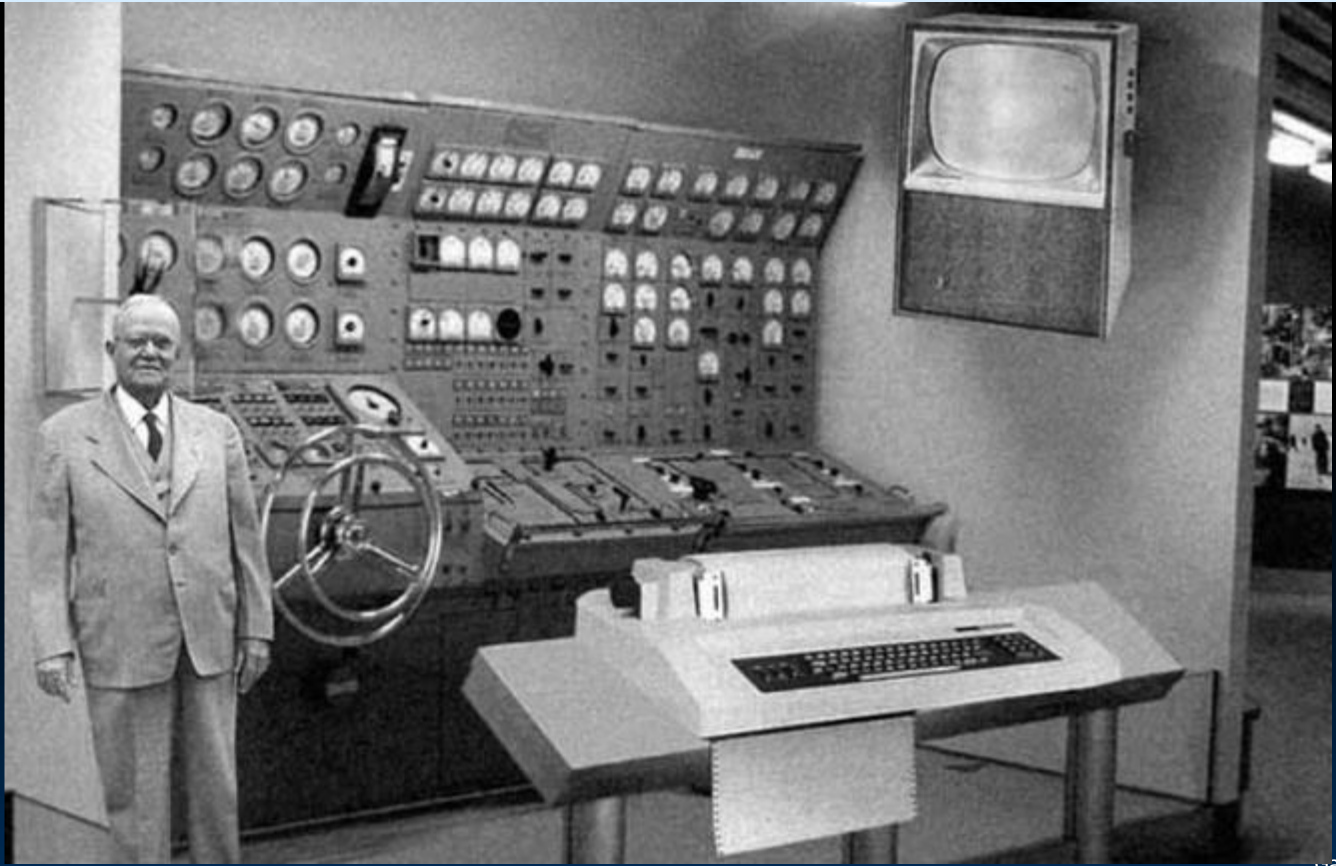
Thinking about Machine Structures

- Thinking about Machine Structures
- Great Ideas in Computer Architecture

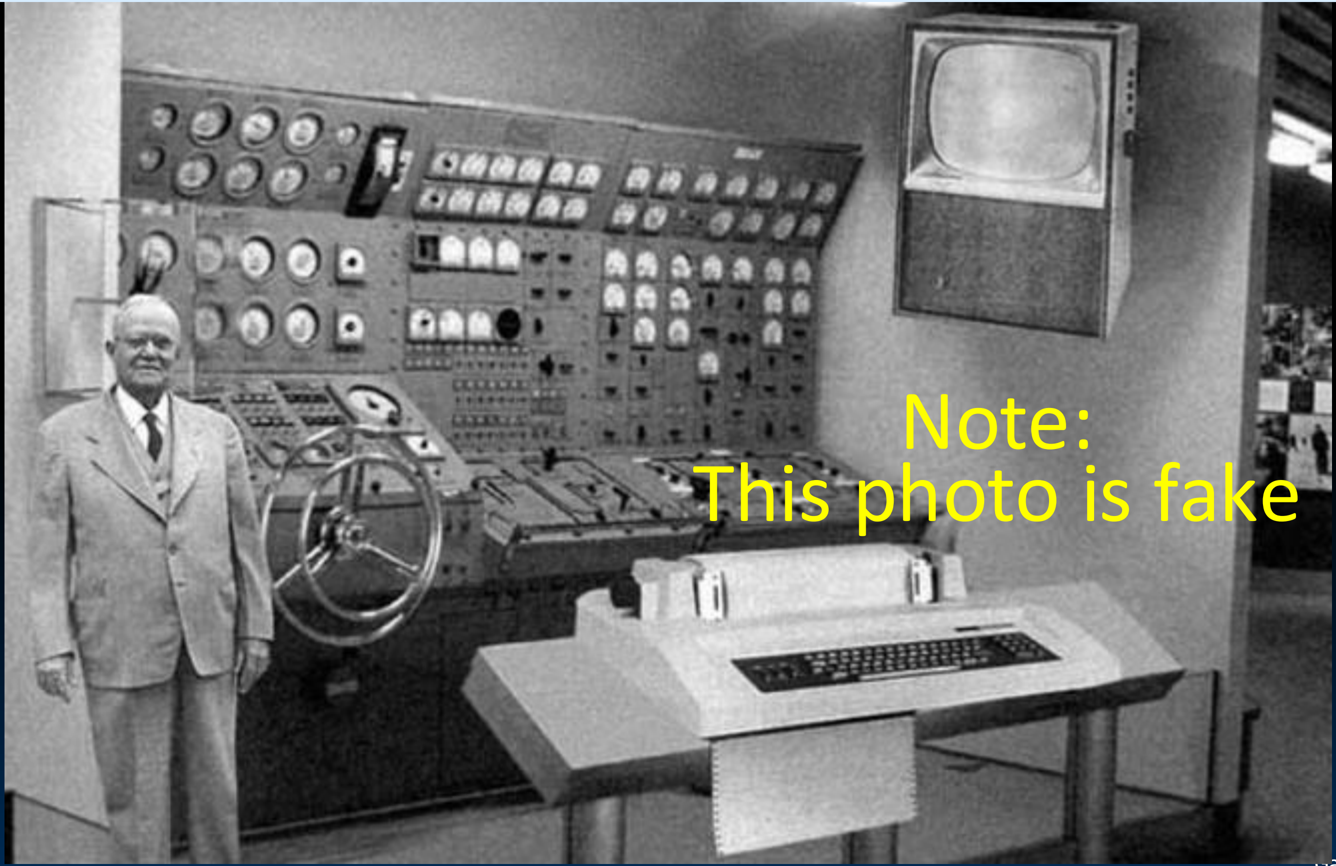
CS61C is NOT about C Programming

- It is about the hardware-software interface
  - What does the programmer need to know to achieve the highest possible performance
- Languages like C are closer to the underlying hardware, unlike languages like Snap!, Python, Java
  - We can talk about hardware features in higher-level terms
  - Allows programmer to explicitly harness underlying hardware parallelism for high performance

Old School CS61C



Old School CS61C



New School CS61C (1/3)

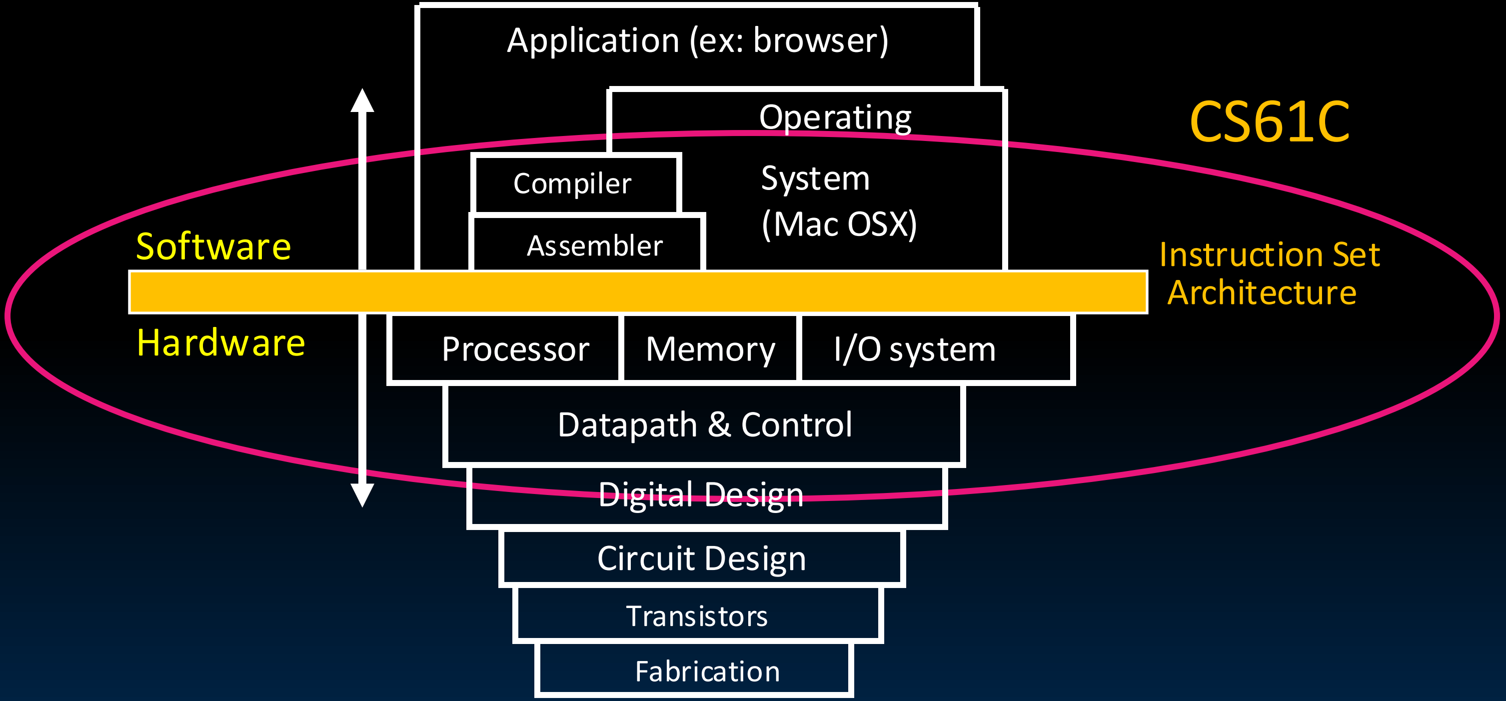
Personal
Mobile
Devices

Network
Edge Devices

New School CS61C (2/3)

New School CS61C (3/3)

Old School Machine Structures



New-School Machine Structures (It’s a bit more complicated!)

Software

Parallel Requests
Assigned to computer
e.g., Search “Cats”

Parallel Threads
Assigned to core e.g., Lookup, Ads

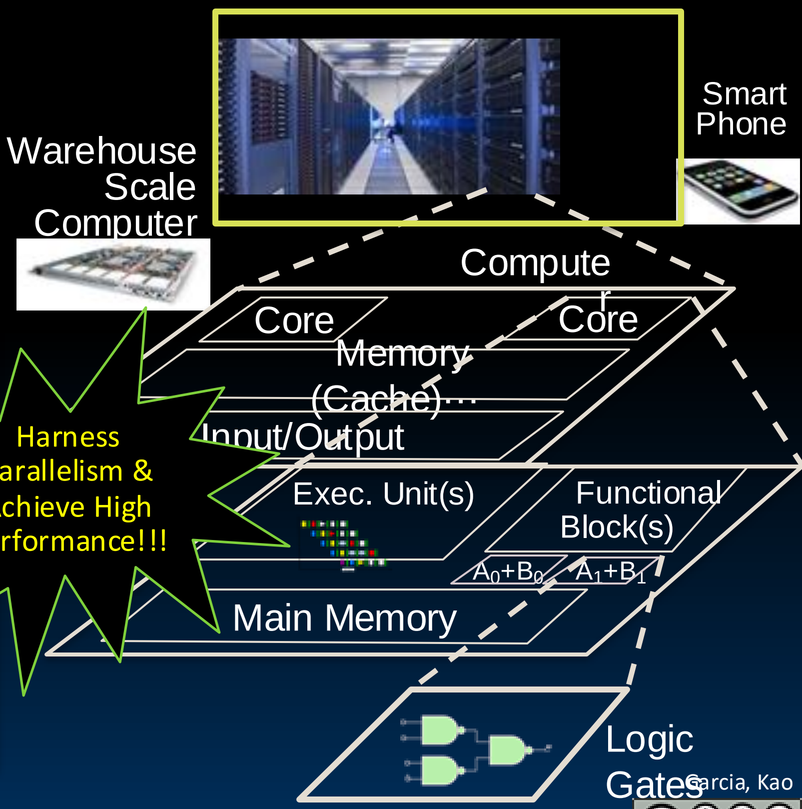
Parallel Instructions
>1 instruction @ one time
e.g., 5 pipelined instructions

Parallel Data
>1 data item @ one time
e.g., Add of 4 pairs of words

Hardware descriptions
All gates work in parallel at same time

Harness Parallelism & Achieve High Performance!!!

Hardware



Agenda

Great Ideas in Computer Architecture

- Thinking about Machine Structures
- Great Ideas in Computer Architecture

6 Great Ideas in Computer Architecture

1. Abstraction (Layers of Representation/Interpretation)
2. Moore's Law
3. Principle of Locality/Memory Hierarchy
4. Parallelism
5. Performance Measurement & Improvement
6. Dependability via Redundancy

Great Idea #1: Abstraction (Levels of Representation/Interpretation)

- Structure and Interpretation of Computing Programs, my good friend…

```
import numpy

x = 3
x = "hello, world"
x = [3, "cs61a", True]
x = len
x = numpy
x = lambda x: x + 2

# anything can be a Python name!
```

Great Idea #1: Abstraction
(Levels of Representation/Interpretation)

High Level Language Program (e.g., C)

```
temp = v[k];
v[k] = v[k+1];
v[k+1] = temp;
```

Compiler

Assembly Language Program (e.g., RISC-V)

```
lw   x3, 0(x10)
lw   x4, 4(x10)
sw   x4, 0(x10)
sw   x3, 4(x10)
```

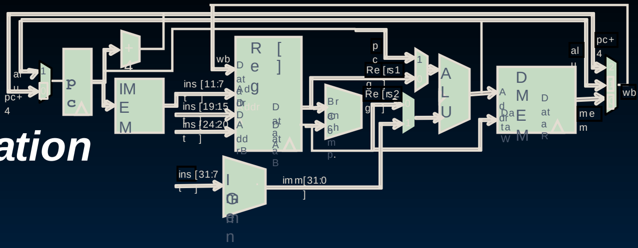
Anything can be represented as a number, i.e., data or instructions

Assembler

Machine Language Program (RISC-V)

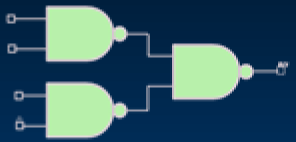
```
1000 1101 1110 0010 0000 0000 0000 0000
1000 1110 0001 0000 0000 0000 0000 0100
1010 1110 0001 0010 0000 0000 0000 0000
1010 1101 1110 0010 0000 0000 0000 0100
```

Hardware Architecture Description (e.g., block diagrams)

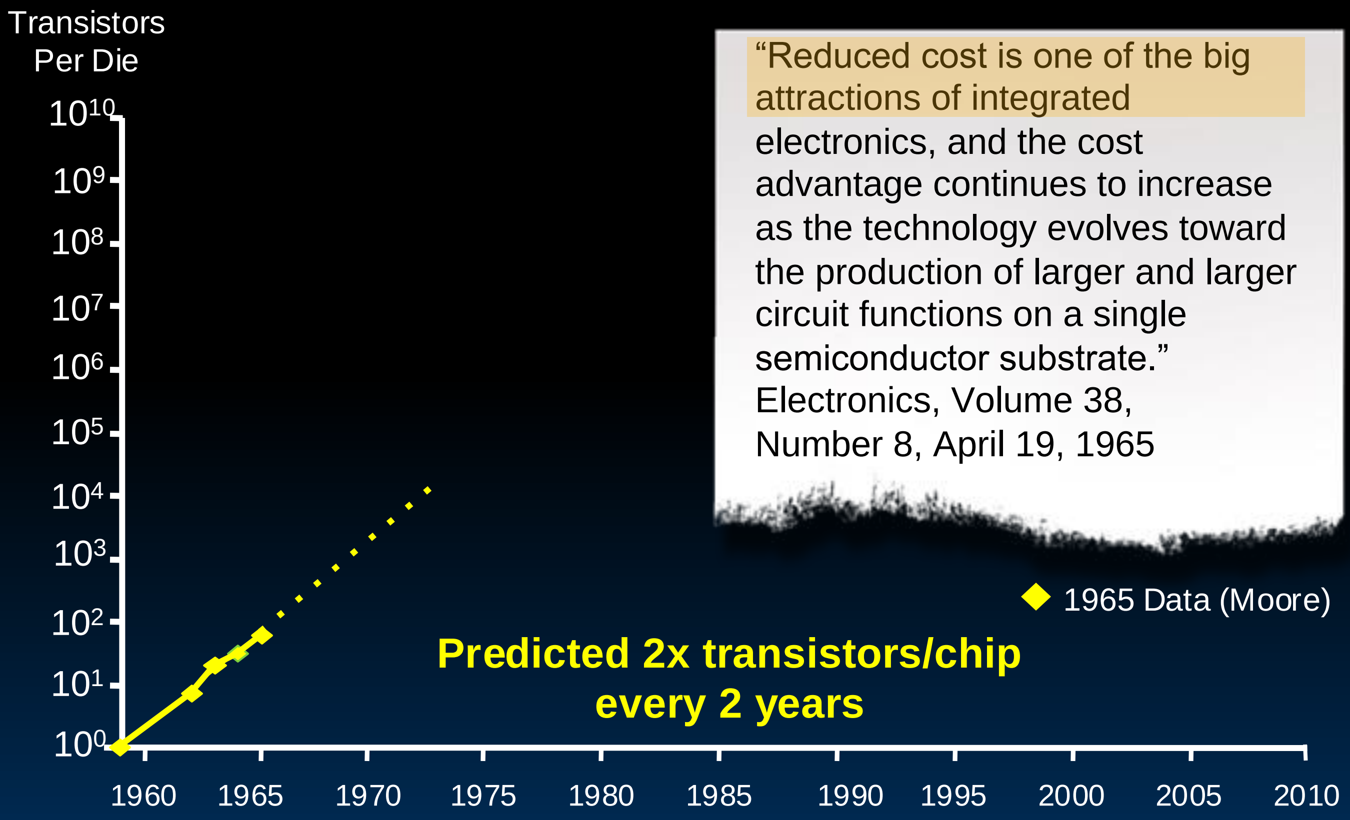


Architecture Implementation

Logic Circuit Description (Circuit Schematic Diagrams)



Great Idea #2: Moore's Law - 2005



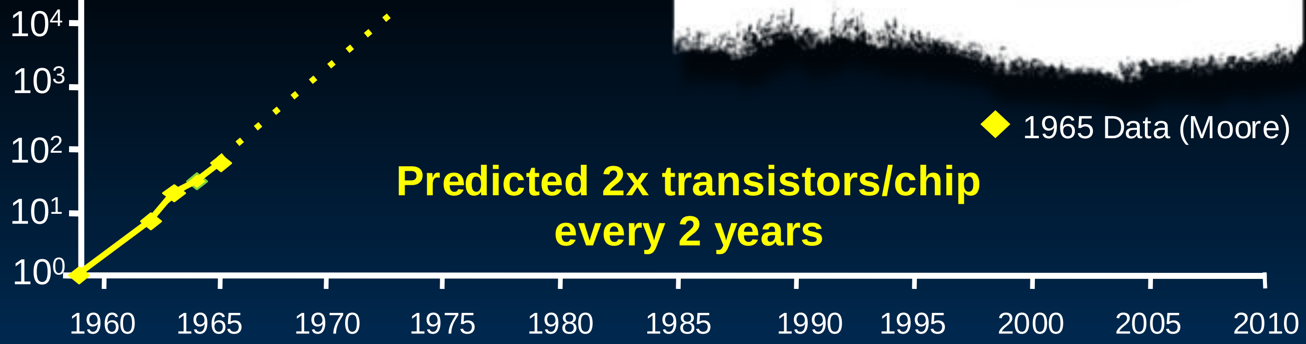
"Reduced cost is one of the big attractions of integrated electronics, and the cost advantage continues to increase as the technology evolves toward the production of larger and larger circuit functions on a single semiconductor substrate."
Electronics, Volume 38, Number 8, April 19, 1965

Source: Intel

Great Idea #2: Moore's Law - 2005

Gordon Moore
Intel Cofounder
B.S. Cal 1950!

"Reduced cost is one of the big attractions of integrated electronics, and the cost advantage continues to increase as the technology evolves toward the production of larger and larger circuit functions on a single semiconductor substrate."
Electronics, Volume 38, Number 8, April 19, 1965

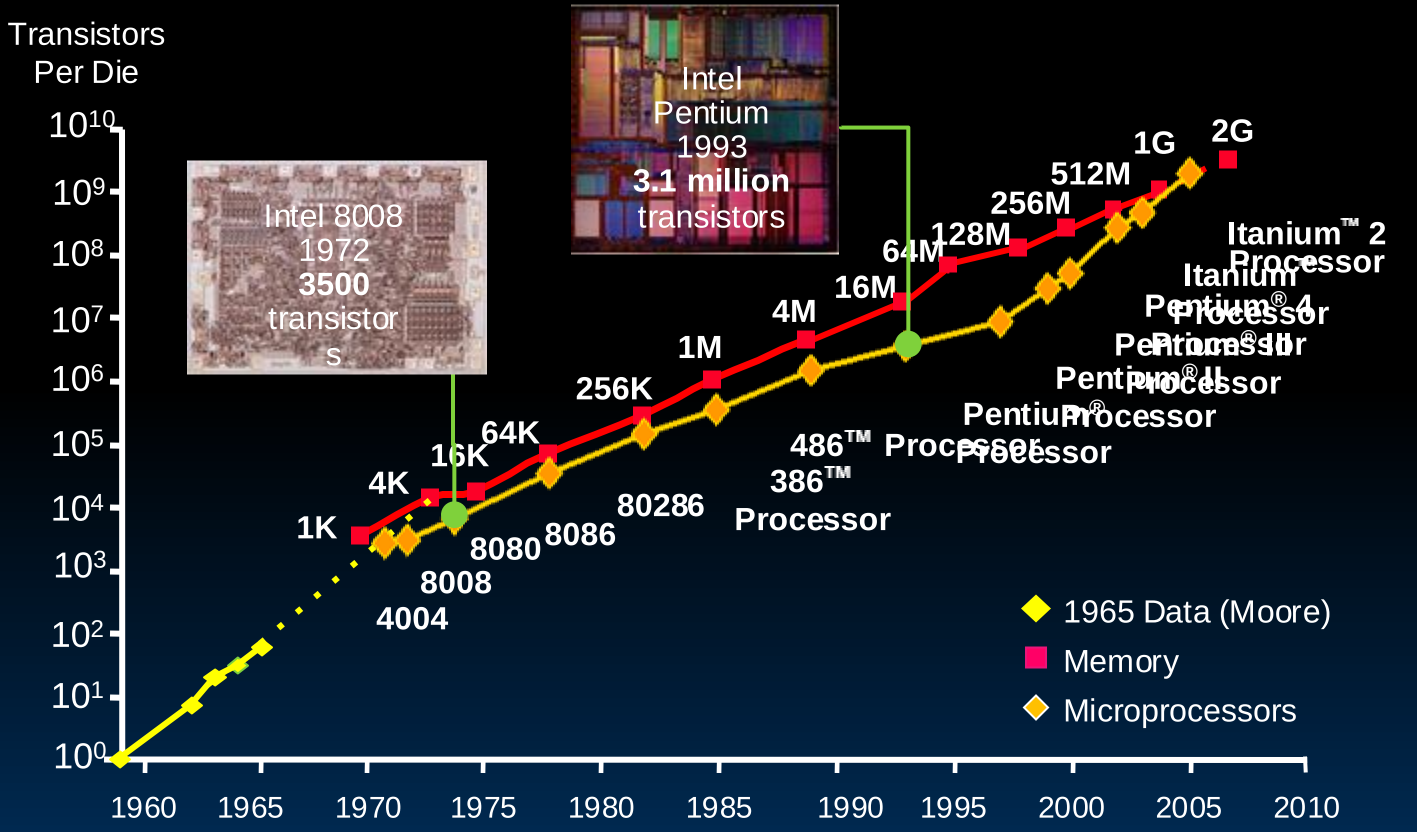


Predicted 2x transistors/chip every 2 years

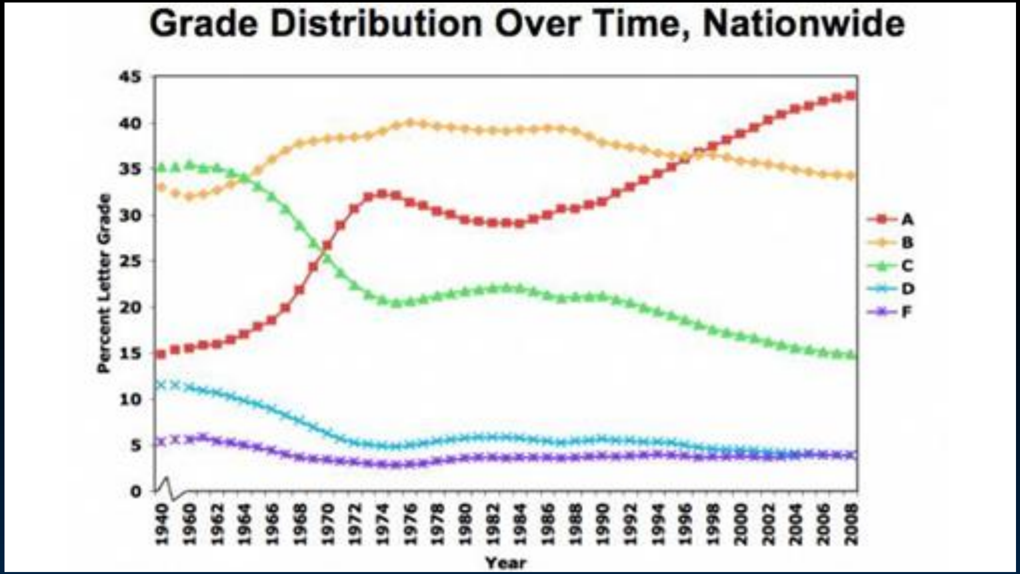
Source: Intel

Garcia, Kao

Great Idea #2: Moore's Law - 2005



A’s Law

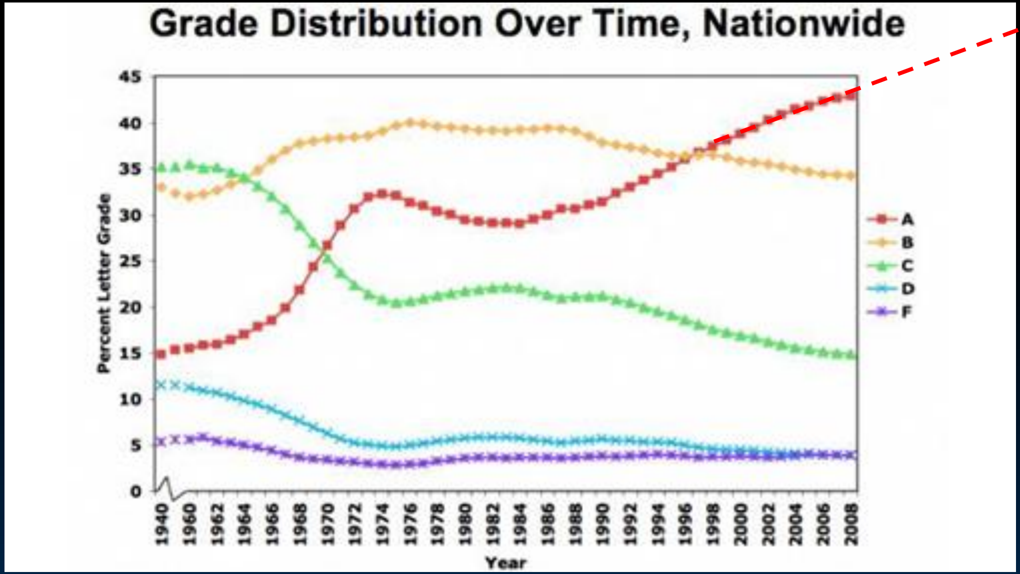


Teachers College Record Volume 114 Number 7, 2012, p. 1-23

A’s Law

Great news:
Your grandchildren WILL all get As!

100%

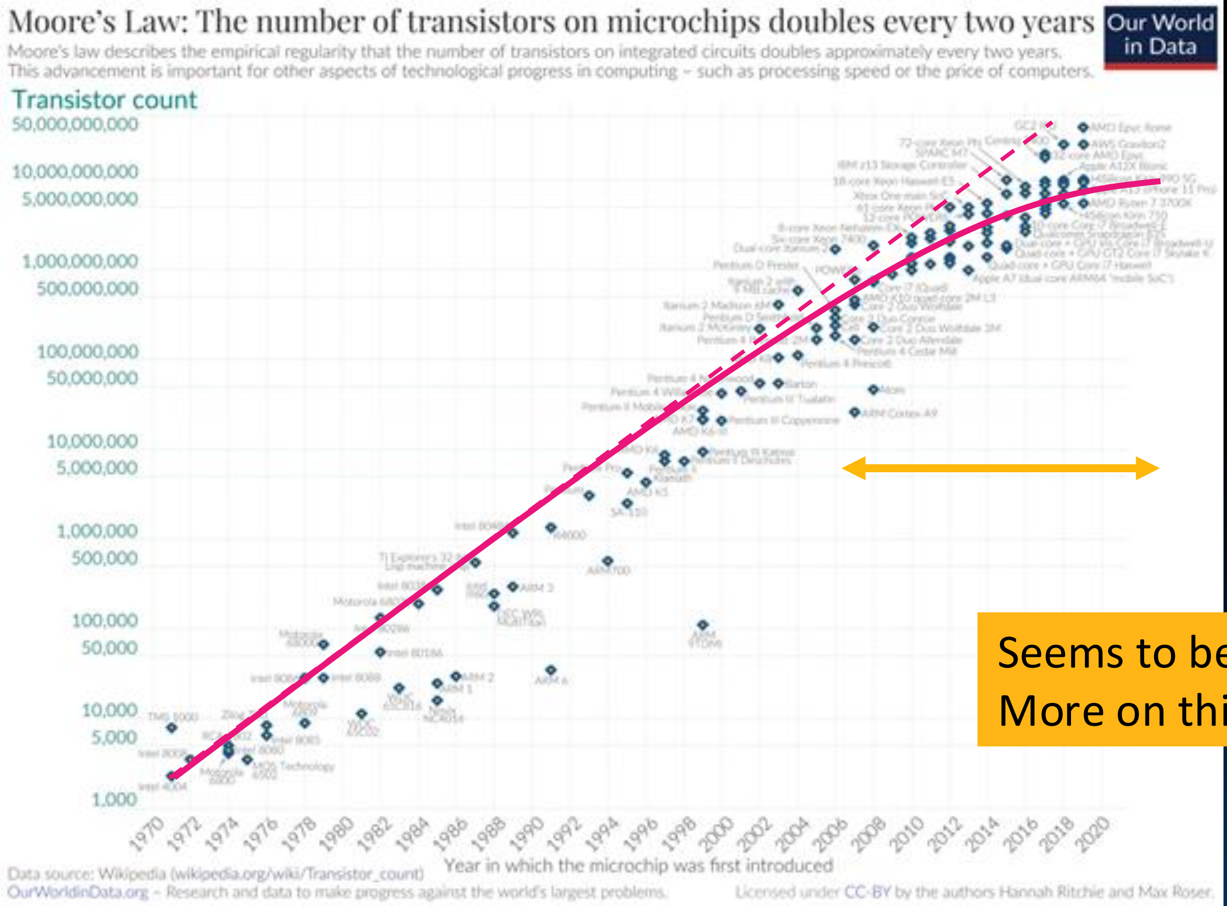


…

2070

Teachers College Record Volume 114 Number 7, 2012, p. 1-23

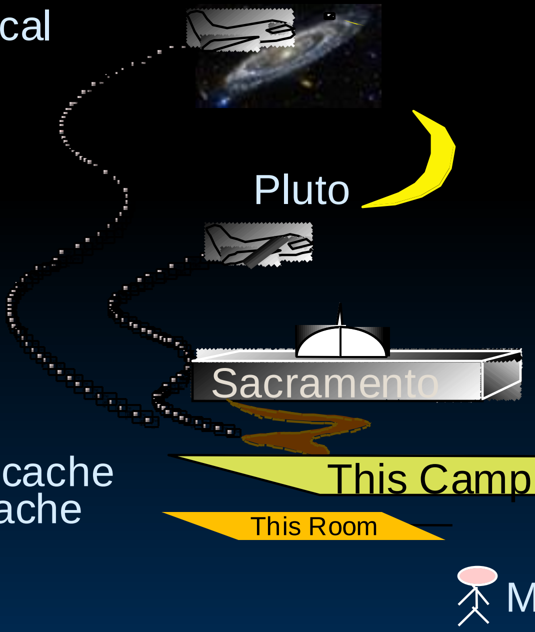
Moore's Law…?

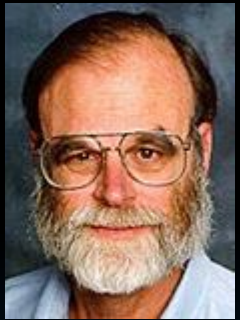


Seems to be tapering?
More on this in a bit…

Storage Latency Analogy: How Far Away is the Data?

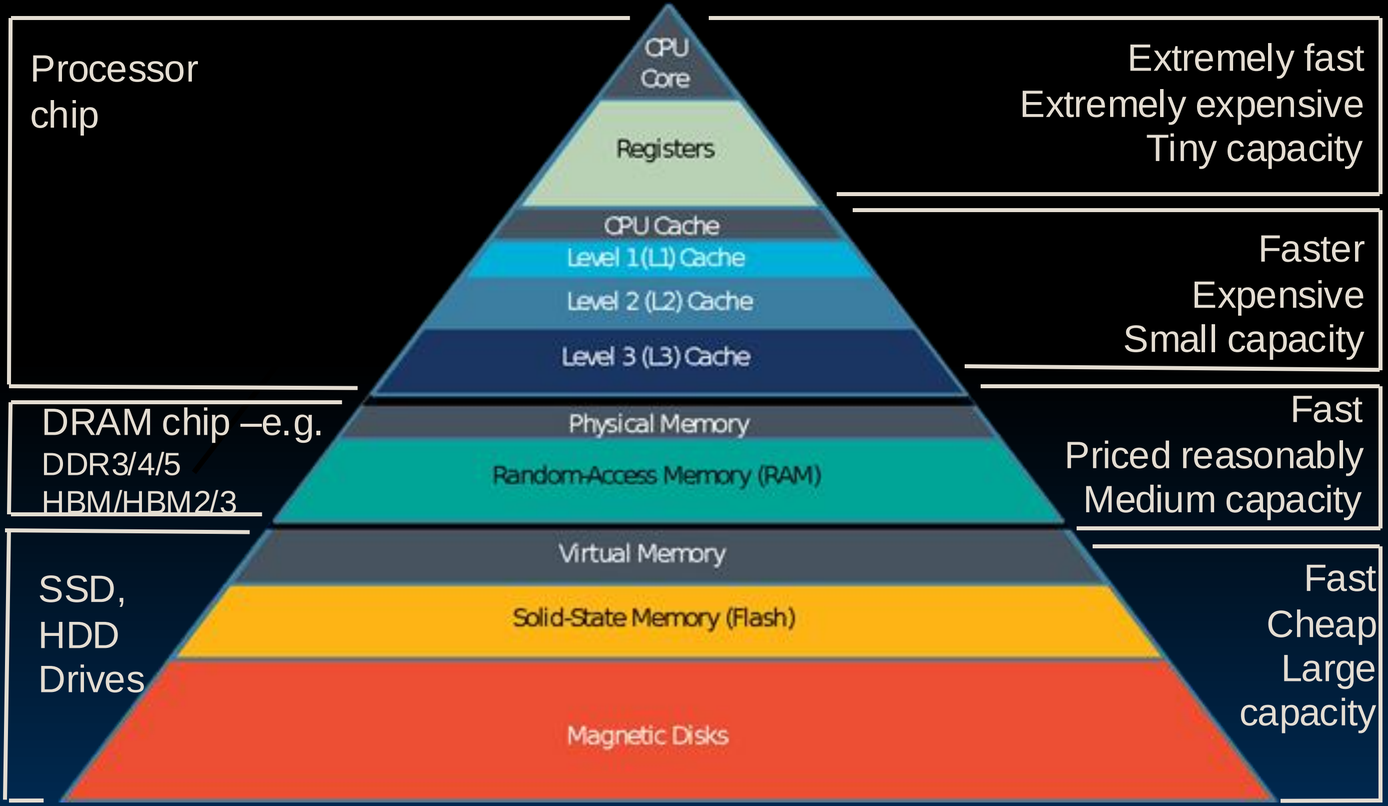
10^9	Tape /optical robot	Andromeda	2,000 Years
10^6	Disk	Pluto	2 Years
100	Memory	Sacramento	1.5 hr
10	On-board cache	This Campus	10 min
2	On-chip cache	This Room	
1	Registers	My Head	1 min
[ns]			





Jim Gray
Turing Award
B.S. Cal 1966
Ph.D. Cal 1969

Great Idea #3: Principle of Locality / Memory Hierarchy



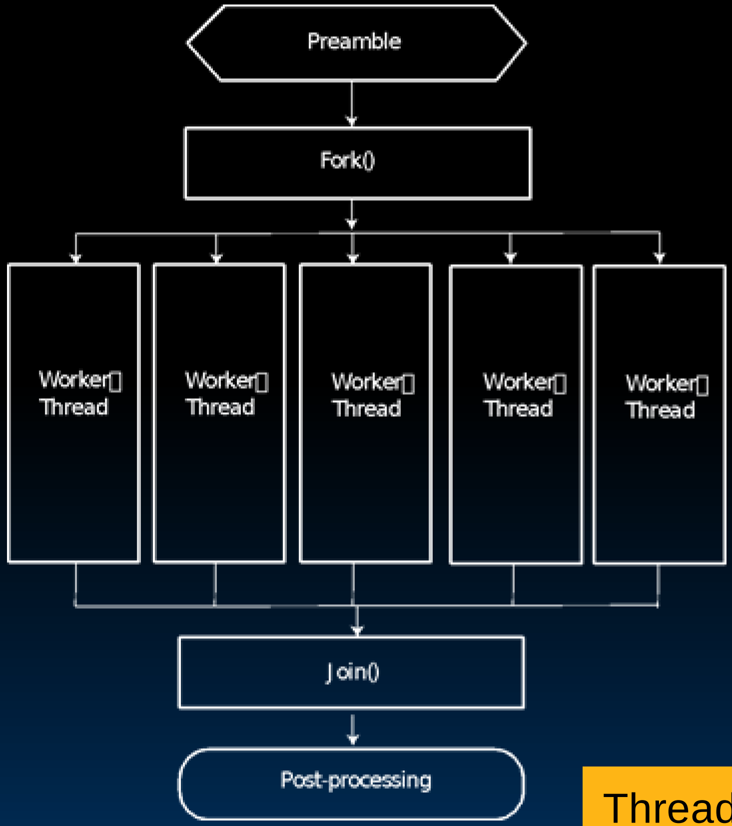
Great Idea #4: Parallelism (1/3)

	Time 1	Time 2	Time 3	Time 4	Time 5	Time 6	Time 7	Time 8	Time 9
Instruction 1	Instruction Fetch	Instruction Decode	Execute	Memory Access	Write Back				
Instruction 2		Instruction Fetch	Instruction Decode	Execute	Memory Access	Write Back			
Instruction 3			Instruction Fetch	Instruction Decode	Execute	Memory Access	Write Back		
Instruction 4				Instruction Fetch	Instruction Decode	Execute	Memory Access	Write Back	
Instruction 5					Instruction Fetch	Instruction Decode	Execute	Memory Access	Write Back

In time slot 3,
Instruction 1 is being executed
Instruction 2 is being decoded
And Instruction 3 is being fetched from memory

Instruction-Level Parallelism

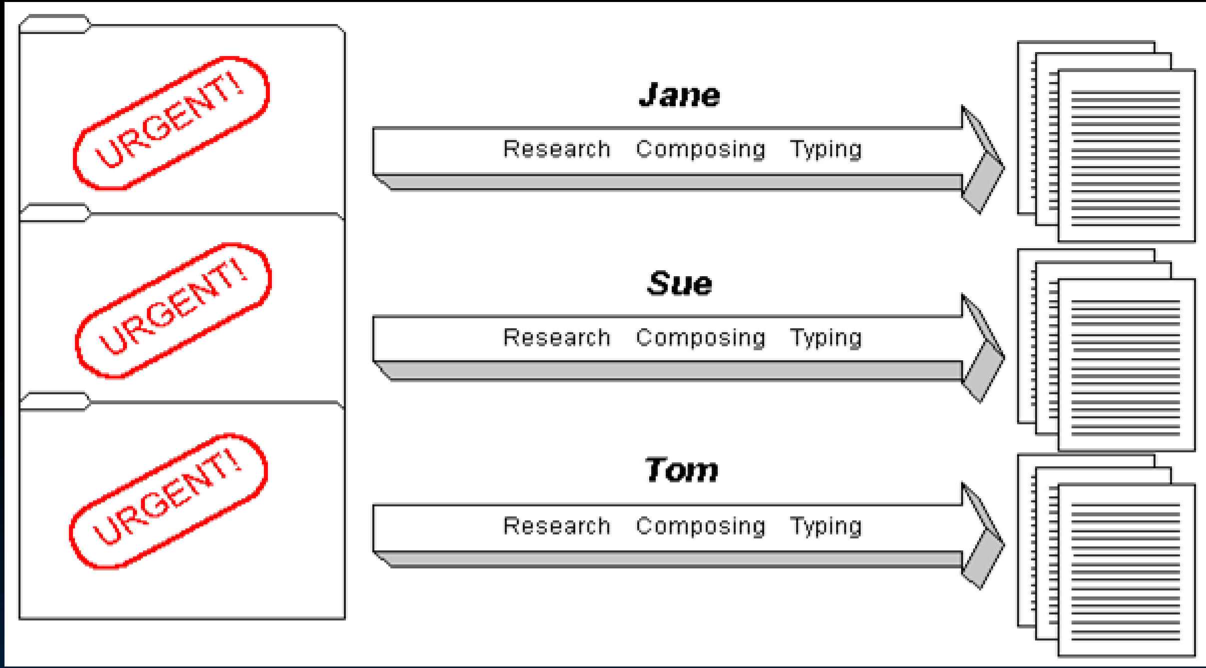
Great Idea #4: Parallelism (2/3)



Thread-Level Parallelism

Garcia, Kao

Great Idea #4: Parallelism (3/3)



Data-Level Parallelism

Caveat! Amdahl's Law

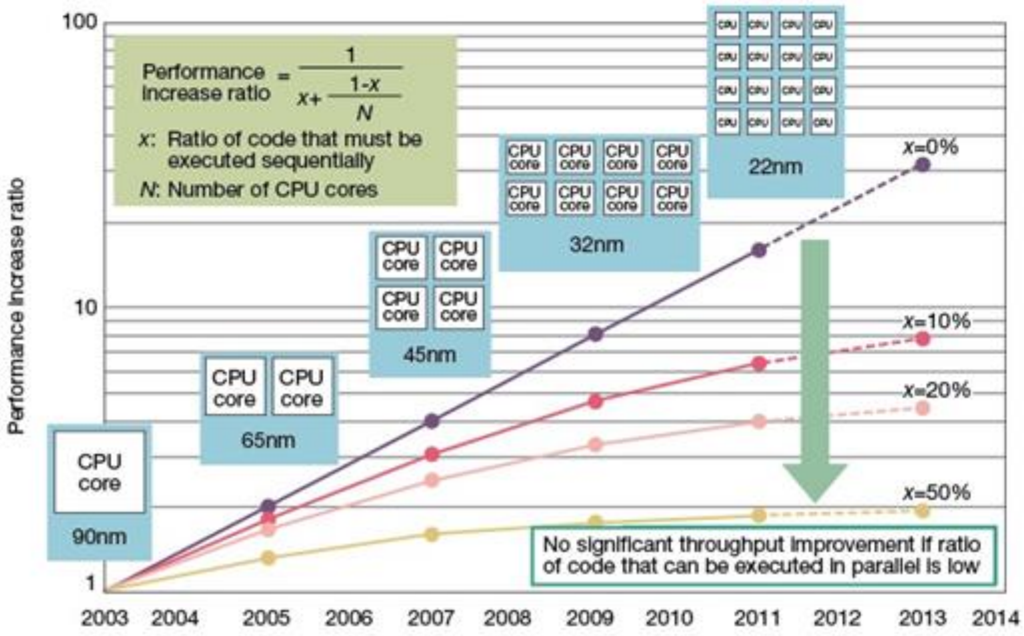


Fig 3 Amdahl's Law an Obstacle to Improved Performance Performance will not rise in the same proportion as the increase in CPU cores. Performance gains are limited by the ratio of software processing that must be executed sequentially. Amdahl's Law is a major obstacle in boosting multicore microprocessor performance. Diagram assumes no overhead in parallel processing. Years shown for design rules based on Intel planned and actual technology. Core count assumed to double for each rule generation.

Gene Amdahl

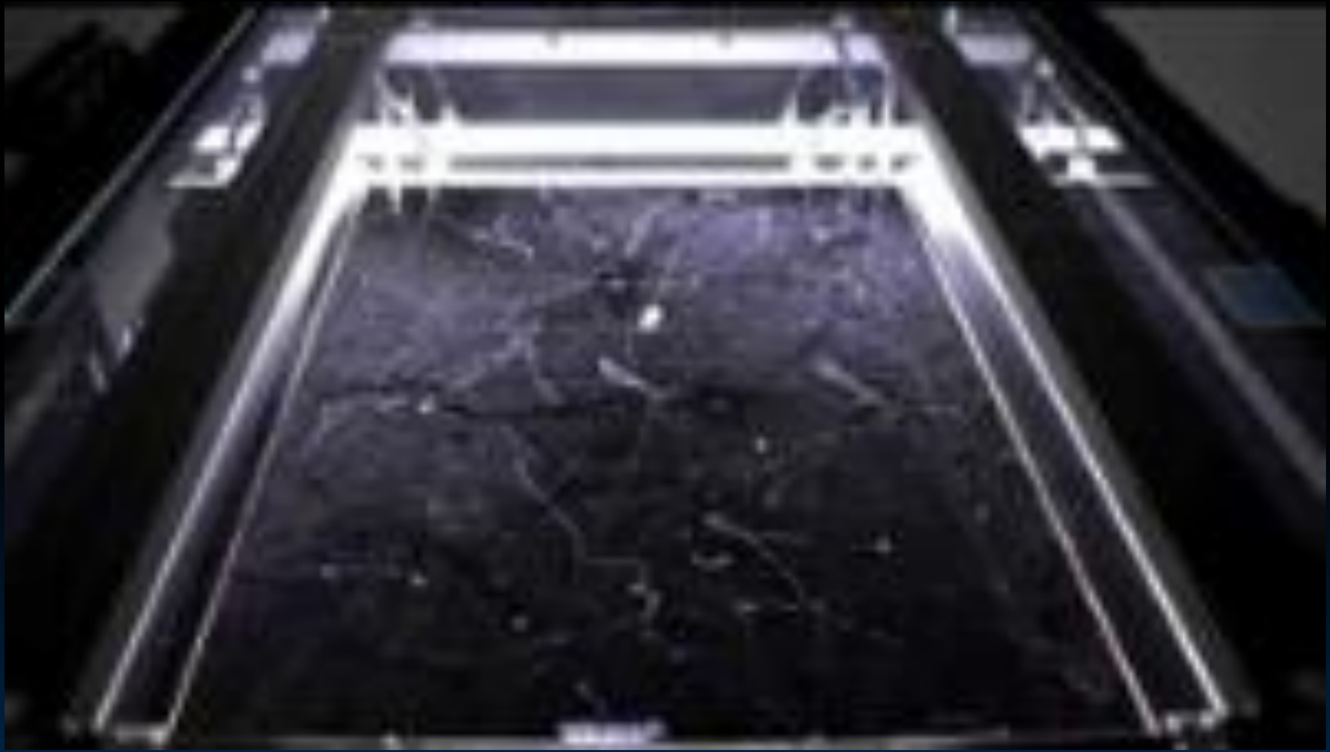
Computer Pioneer

Great Idea #5: Performance Measurement & Improvement

- Match application to underlying hardware to exploit:
  - Locality;
  - Parallelism;
  - Special hardware features, like specialized instructions (e.g., matrix manipulation).
- Latency/Throughput:
  - How long to set the problem up and complete it (or how many tasks can be completed in given time)
  - How much faster does it execute once it gets going
- Latency is all about time to finish.

Great Idea #6: Dependability via Redundancy

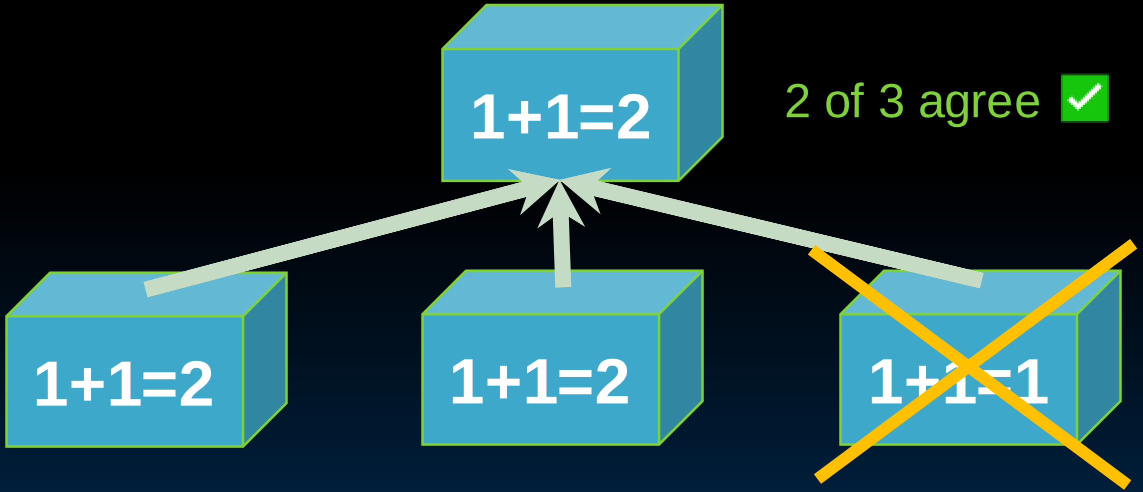
Unintended transistor behavior can be caused by unintended electron flow from cosmic rays (among other reasons)!



https://www.exploratorium.edu/exhibits/cloud-chamber

Great Idea #6: Dependability via Redundancy

- Design with redundancy so that a failing piece doesn't make the whole system fail.



Increasing transistor density reduces the cost of redundancy

Great Idea #6: Dependability via Redundancy

- Applies to everything from datacenters to storage to memory to instructors!
  - Redundant **datacenters** so that can lose 1 datacenter but Internet service stays online;

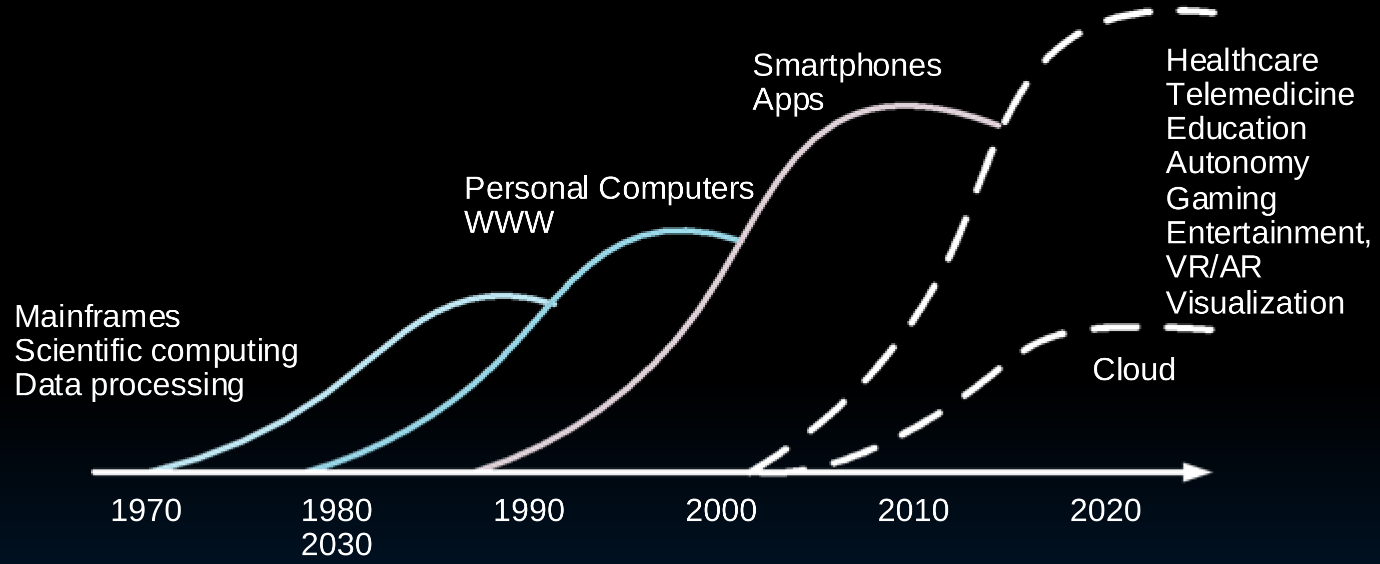
  - Redundant **disks** so that can lose 1 disk but not lose data (Redundant Arrays of Independent Disks/RAID);

  - Redundant **memory bits** so that can lose 1 bit but no data (Error Correcting Code/ECC Memory).

(dramatic pause)

Why is computer architecture exciting today?

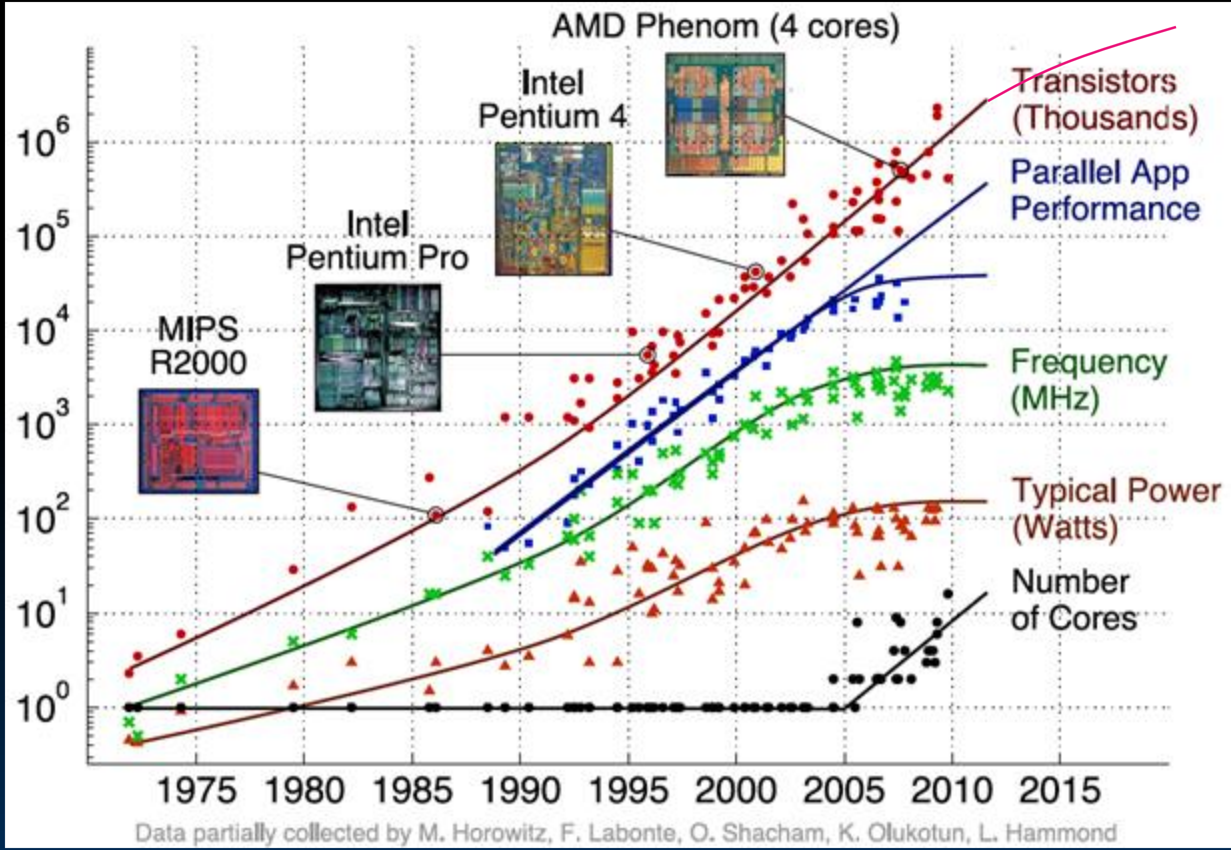
Why Is Computer Architecture Exciting Today?



- Number of deployed devices continues to grow, but there is no single killer application.
  - Diversification of needs, architectures
  - Machine learning is common for most domains

Reason 1: Changing Constraints

- Moore's Law ending
- Power limitations
- Amdahl's Law



Reason 2: Era of Domain-Specific Computing

- Each domain requires heterogeneous systems
  - Multiple processor cores
  - GPUs,
  - NPUs,
  - accelerators,
  - interfaces,
  - memory, …

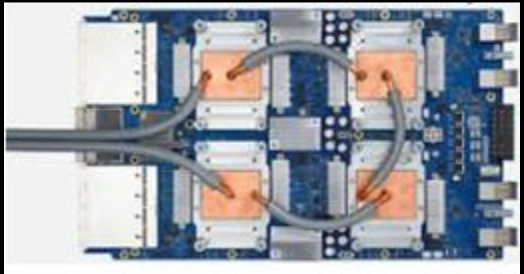
Apple A15 Bionic
Source: SemiAnalysis

Old Conventional Wisdom

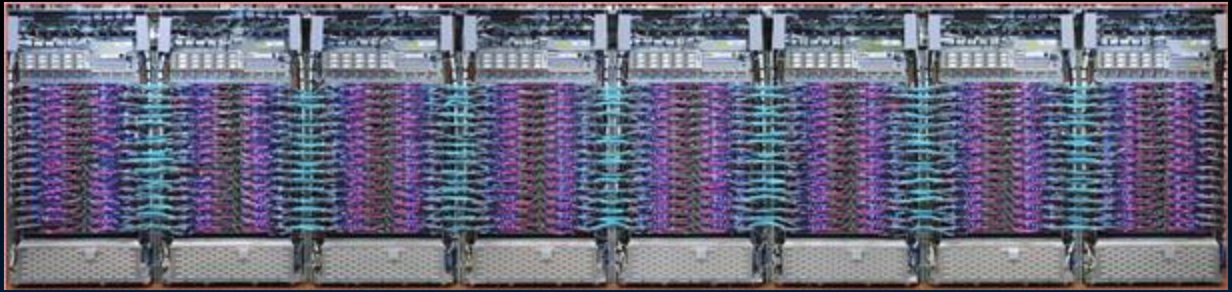
- Moore's Law + Dennard Scaling = faster, cheaper, lower-power general-purpose computers each year
- In glory days, 1%/week performance improvement!

- Dumb to compete by designing parallel or specialized computers
- By time you've finished design, the next generation of general-purpose will beat you

New Conventional Wisdom



Google TPU3
Specialized Engine for training Neural Networks
Deployed in cloud



1024 chips, > 100PetaFLOPs

Patterson and Hennessy win Turing!

Innovations in computer architecture have a convention of defying conventional wisdom

https://engineering.berkeley.edu/news/2018/06/patterson-wins-turing-award/

Garcia, Kao

Summary

- CS61C: Learn 6 great ideas in computer architecture to enable high performance programming via parallelism, not just learn C.
  1. Abstraction (Layers of Representation / Interpretation)
  2. Moore's Law
  3. Principle of Locality/Memory Hierarchy
  4. Parallelism
  5. Performance Measurement and Improvement
  6. Dependability via Redundancy