ESE345 Computer Architecture Logistics

Instructor: Ryan Thielke
E-mail: ryan.thielke@stonybrook.edu

Office Hours: M/W 4:00-5:45PM
Light Engineering Room 258A

Class: MW 6:05-7:25 PM
Earth & Space 131

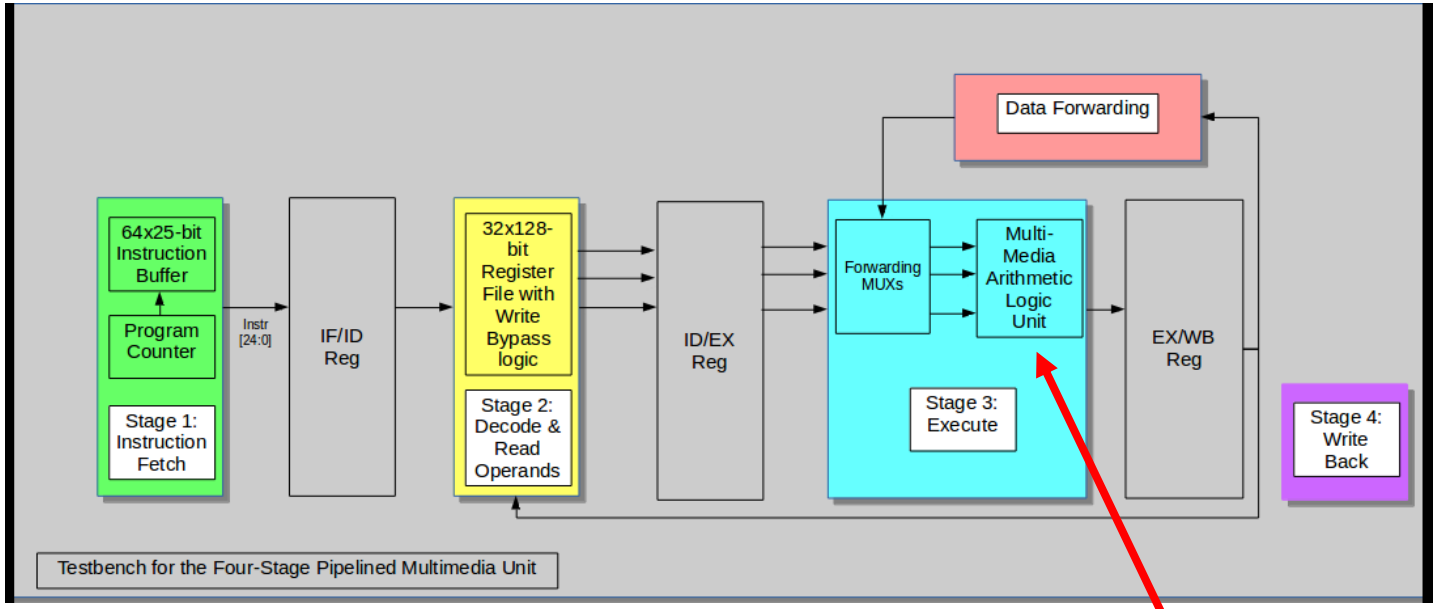
Teaching Assistants:	Xiaokun Zhao	Mondays 2-3pm	(Project and Course Material)
	Jingwei Li	Tuesdays 2-3pm	(HW and Course Material)

Text: *David A. Patterson and John L. Hennessy "Computer Organization & Design The Hardware/ Software Interface," 5th ed., 2014 by Elsevier Inc. ISBN: 978-0-12-407726-3.*

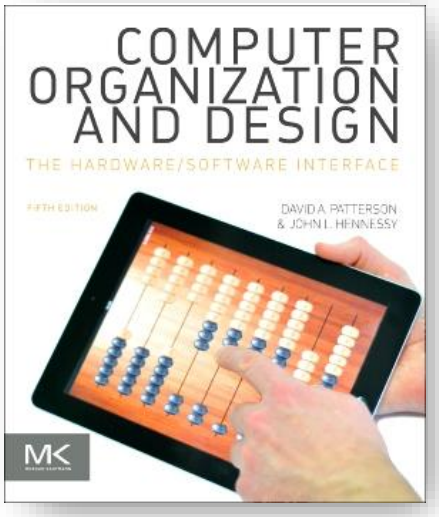
Course Grading: Six homeworks + 1 programming assignment : 14%
Exams (two midterms) : 64%
Project: 22%

Discord: https://discord.gg/dS6qXUT9DU

Project: VHDL Pipelined Multimedia Unit Model



- Read **Chapter 3.6** on subword parallelism to understand the original concept of multimedia processing
- **Refresh your knowledge** of VHDL/Verilog in the HDL design of digital circuits
- **Develop and submit behavioral VHDL/Verilog code and its verification results** for all multimedia ALU functions at the 3rd stage
- **Develop the HDL model** of the four-stage pipelined multimedia unit and its modules
- **Verify individual modules** of your design with their testbenches before instantiating them in higher order modules. **Verify the final model** with a testbench and show correct pipeline operation with and without forwarding



ESE 345 Computer Architecture

Introduction and Technology Trends

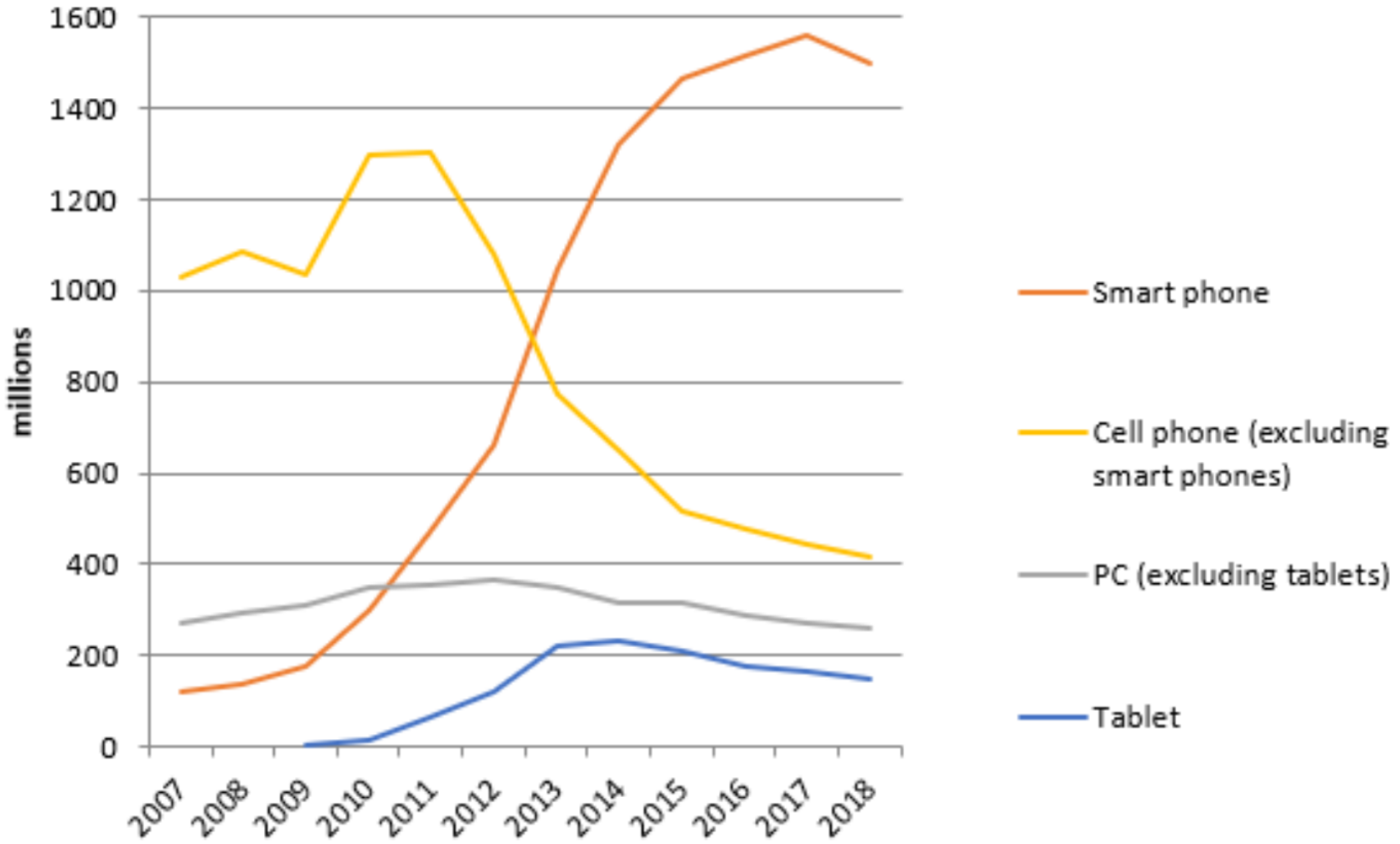
Classes of Computers

- Personal Computers
  - General purpose, variety of software
  - Subject to cost/performance tradeoff
- Server Computers
  - Network based
  - High capacity, performance, reliability
  - Range from small servers to building sized

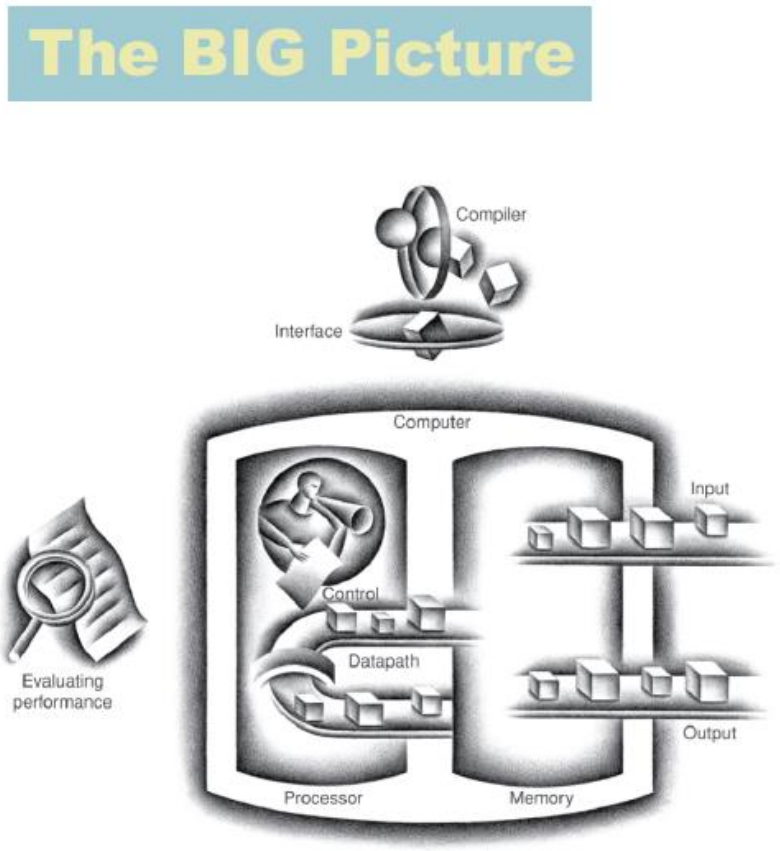
Classes of Computers

- Supercomputers
  - Type of server
  - High-end scientific and engineering calculations
  - Highest computing capability but represent a small fraction of the overall computer market
- Embedded Computers
  - Hidden as component of systems
  - Stringent power/performance/cost constraints

The PostPC Era



Components of a Computer



- Same components for all kinds of computers since 1946
  - Desktop, server, embedded
- Input/output includes
  - User-interface devices
    - Display, keyboard, mouse
  - Storage devices
    - Hard disk, CD/DVD, flash
  - Network adapters
    - For communicating with other computers

Inside the Processor Core (CPU)

- Datapath: performs operations on data
- Control: sequences datapath, memory, etc.
- Cache memory
  - Small fast SRAM memory for immediate access to data

What You Will Learn

- How programs are translated into the machine language
  - And how the hardware executes them
- The hardware/software interface
- What determines program performance
  - And how it can be improved
- How hardware designers improve performance
- What is parallel processing

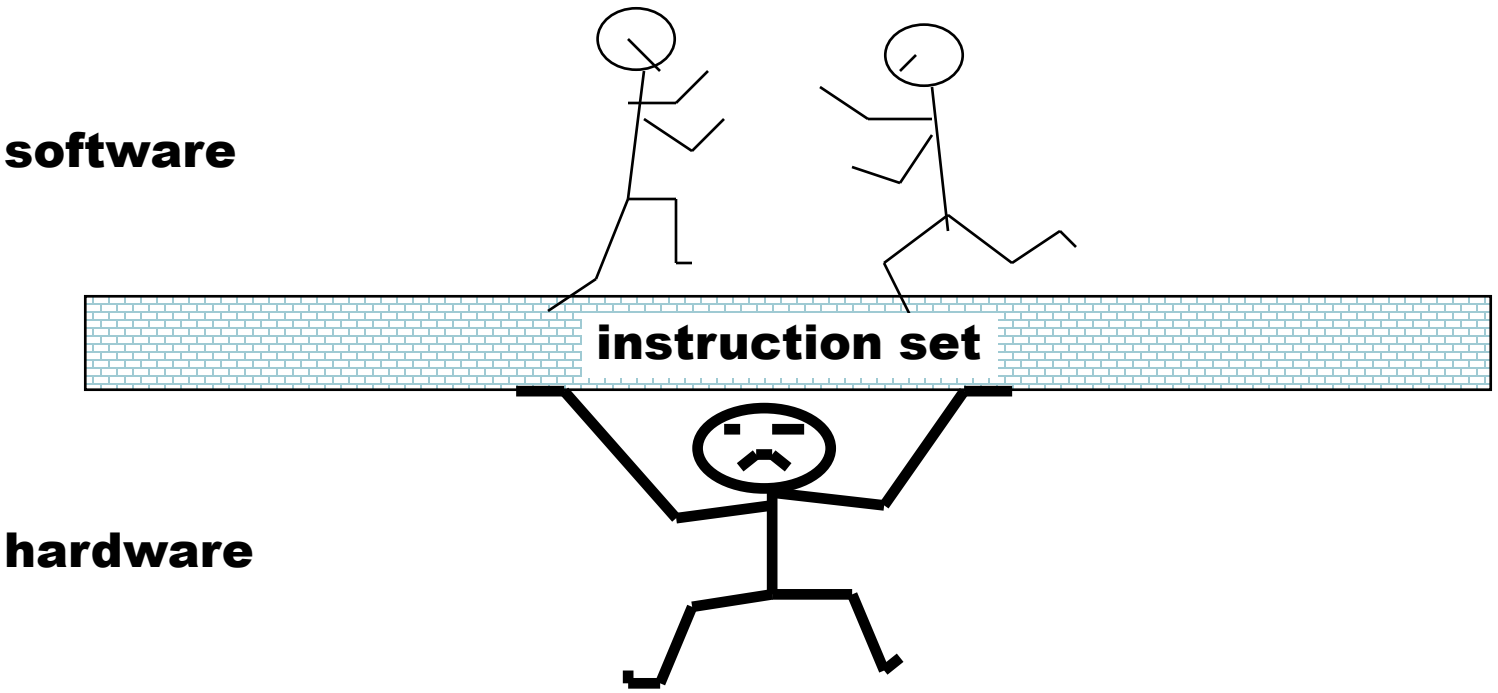
Understanding Performance

- Algorithm
  - Determines number of operations executed
- Programming language, compiler, architecture
  - Determine number of machine instructions executed per operation
- Processor and memory system
  - Determine how fast instructions are executed
- I/O system (including OS)
  - Determines how fast I/O operations are executed

What is Computer Architecture?

- Old definition of computer architecture = instruction set architecture (ISA)
- Our view is computer architecture >> ISA
- Architect's job much more than instruction set design; technical hurdles today *more* challenging than those in instruction set design

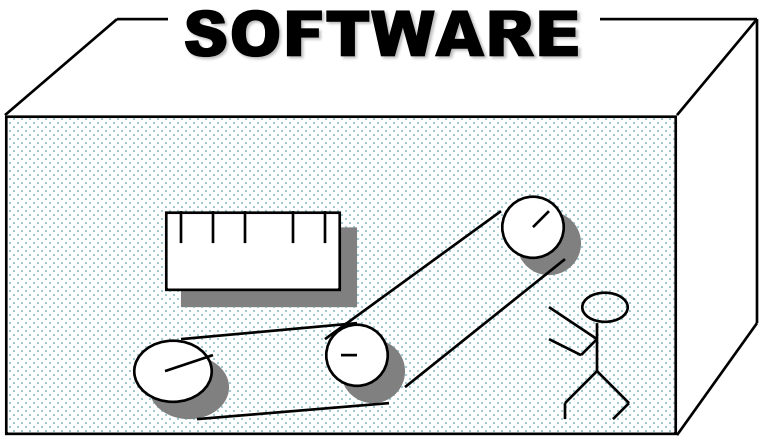
Instruction Set Architecture:Critical Interface



- 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

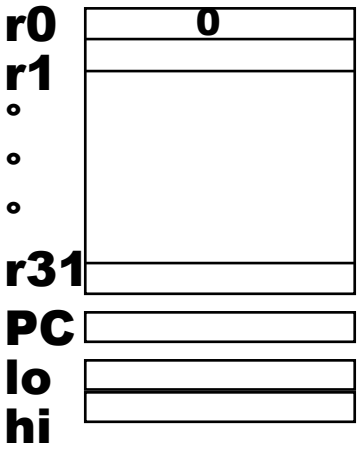
Instruction Set Architecture

"... the attributes of a [computing] system as seen by the programmer, *i.e.* the conceptual structure and functional behavior, as distinct from the organization of the data flows and controls the logic design, and the physical implementation." – Amdahl, Blaauw, and Brooks, 1964



- **Organization of Programmable Storage**
- **Data Types & Data Structures: Encodings & Representations**
- **Instruction Formats**
- **Instruction (or Operation Code) Set**
- **Modes of Addressing and Accessing Data Items and Instructions**
- **Exceptional Conditions**

Example: MIPS



Programmable storage

- 2^32 x bytes
- 31 x 32-bit GPRs (R0=0)
- 32 x 32-bit FP regs (paired DP)
- HI, LO, PC

Data types ?

Format ?

Addressing Modes?

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, BLEZ,BGTZ,BLTZ,BGEZ,BLTZAL,BGEZAL

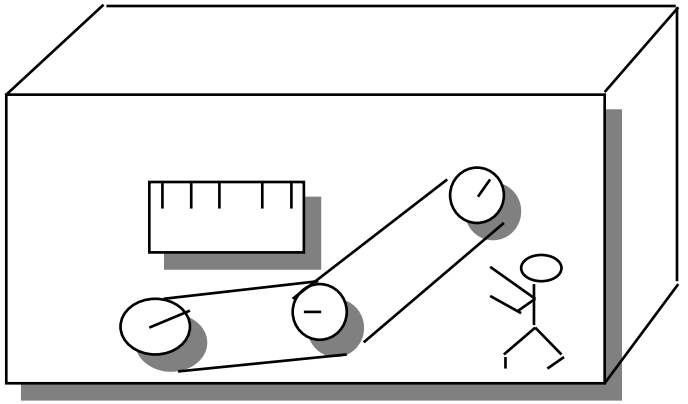
Organization (=Microarchitecture)

- Capabilities & Performance Characteristics of Principal Functional Units
  - (e.g., Registers, ALU, Shifters, Logic Units, ...)
- Ways in which these components are interconnected
- Information flows between components
- Logic and means by which such information flow is controlled.
- Choreography of FUs to realize the ISA
- Register Transfer Level (RTL) Description

Logic Designer's View

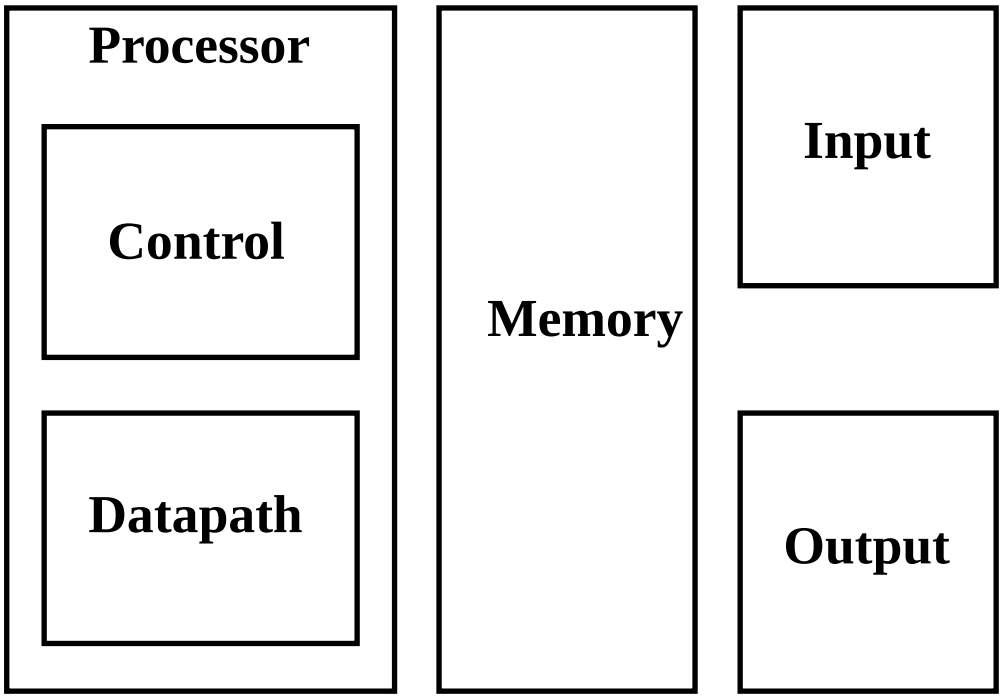
ISA Level

FUs & Interconnect

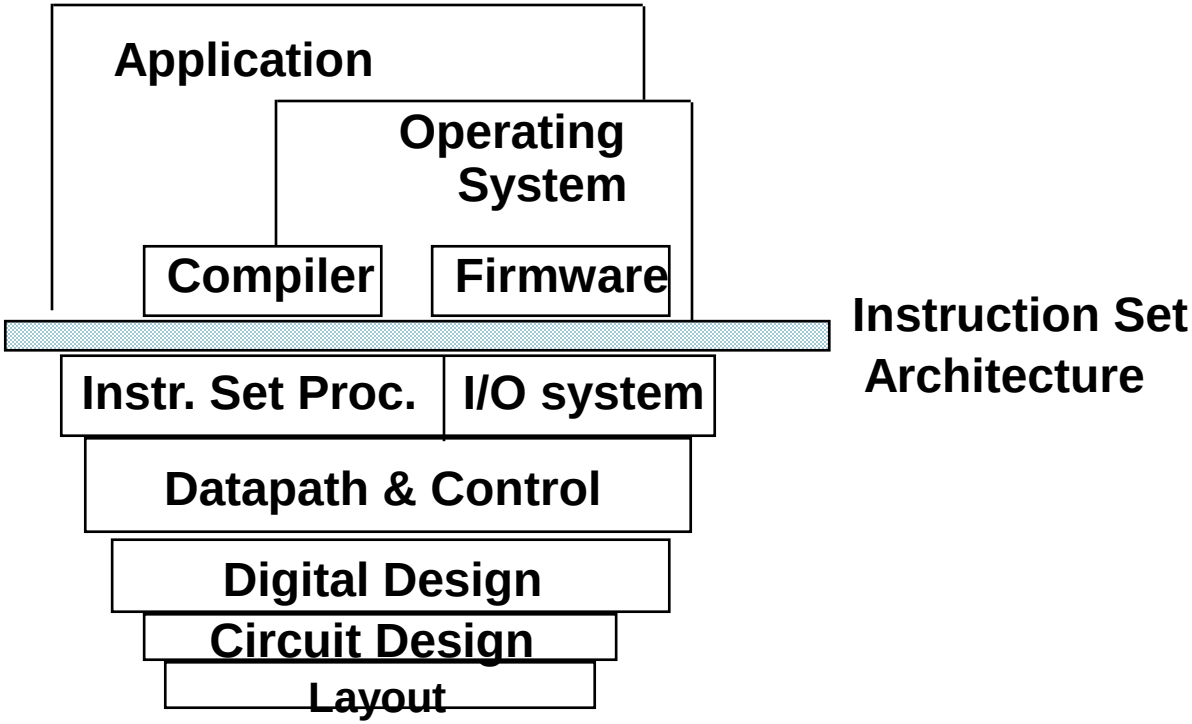


The Big Picture

- Since 1946 all computers have had 5 components



What is "Computer Architecture" Now?



- In its broad definition, computer architecture is the *design of the abstraction layers* to maximize performance within constraints (e.g., cost, power, and availability) that allow us to implement information processing applications efficiently using available manufacturing technologies.

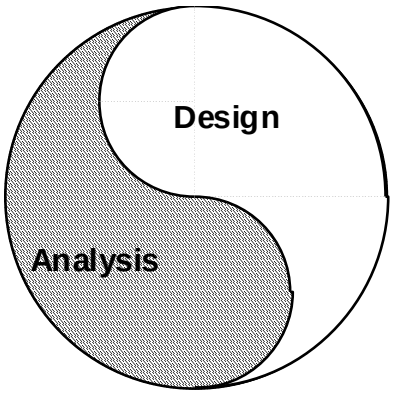
Great Ideas

- Use ***abstraction*** to simplify design
- Make the ***common case fast***
- Performance via ***parallelism***
- Performance via ***pipelining***
- Performance via ***prediction***
- ***Hierarchy*** of memories
- ***Dependability*** via redundancy
- ***Technology trends***

Computer Architecture is an Integrated Approach

- What really matters is the functioning of the complete system
  - hardware, runtime system, compiler, operating system, and application
  - In networking, this is called the “End to End argument”
- Computer architecture is not just about transistors, individual instructions, or particular implementations
  - E.g., Original RISC projects replaced complex instructions with a compiler + simple instructions

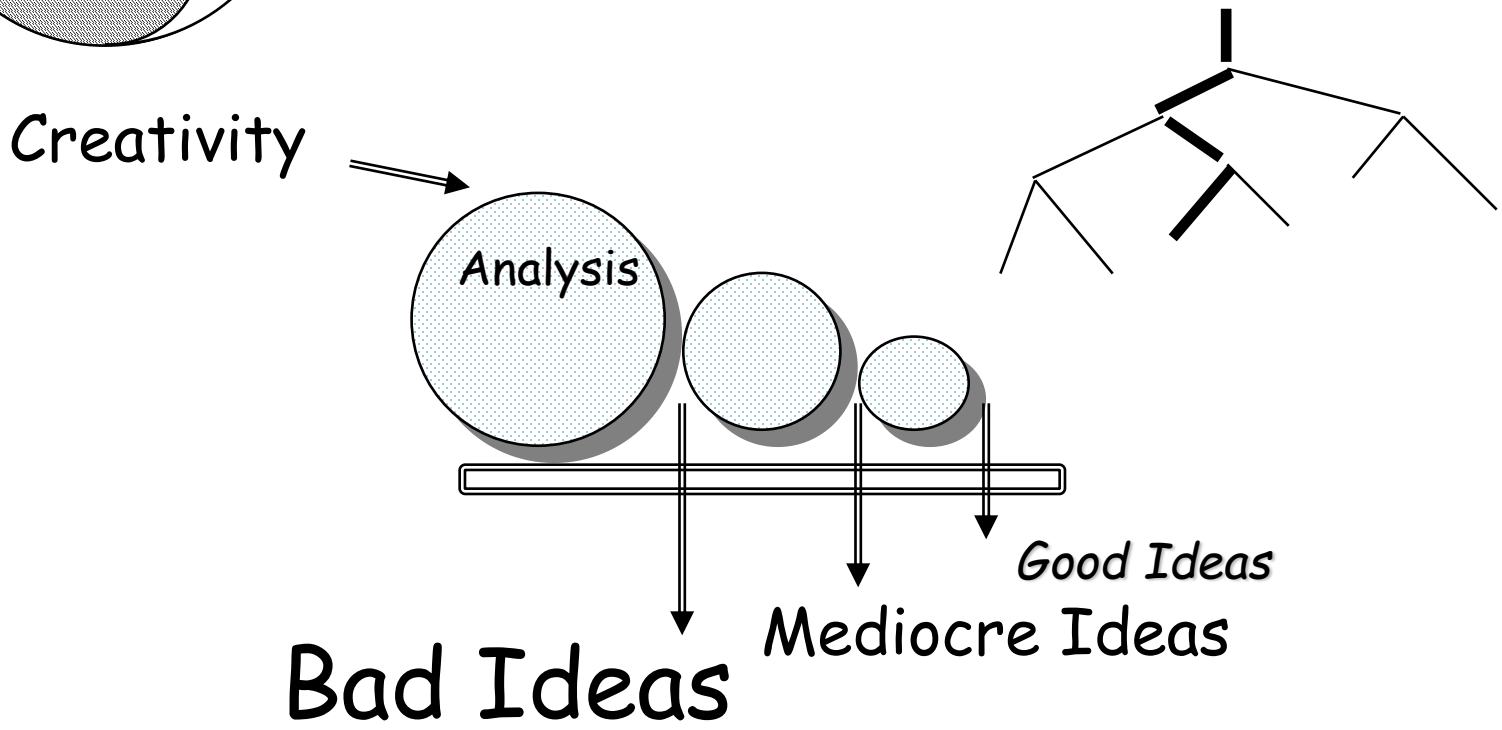
Computer Architecture is Design and Analysis



There is no single recipe for right architecture design

Architecture design is an iterative process:

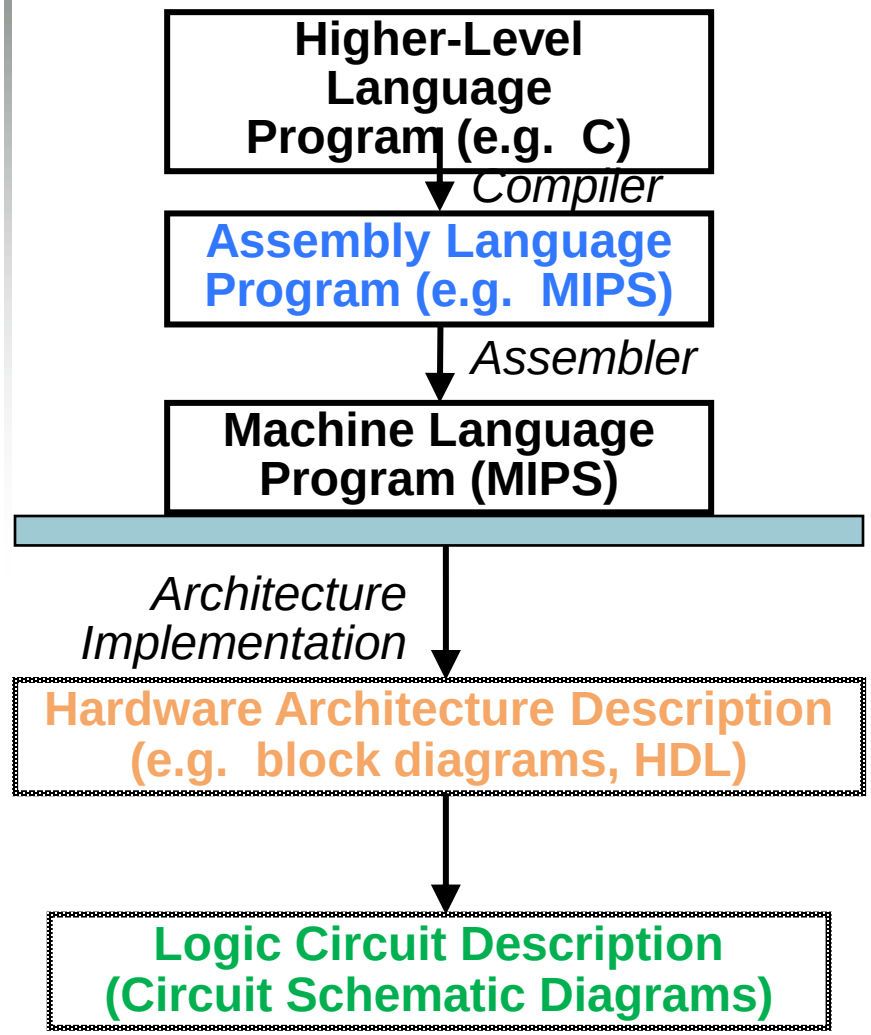
- Searching the space of possible designs
- At all levels of computer systems



Ideas in Computer Architecture

1. Levels of Representation
2. Technology Trends
3. Principle of Locality/Memory Hierarchy
4. Parallelism
5. Performance Measurement
6. Dependability via Redundancy

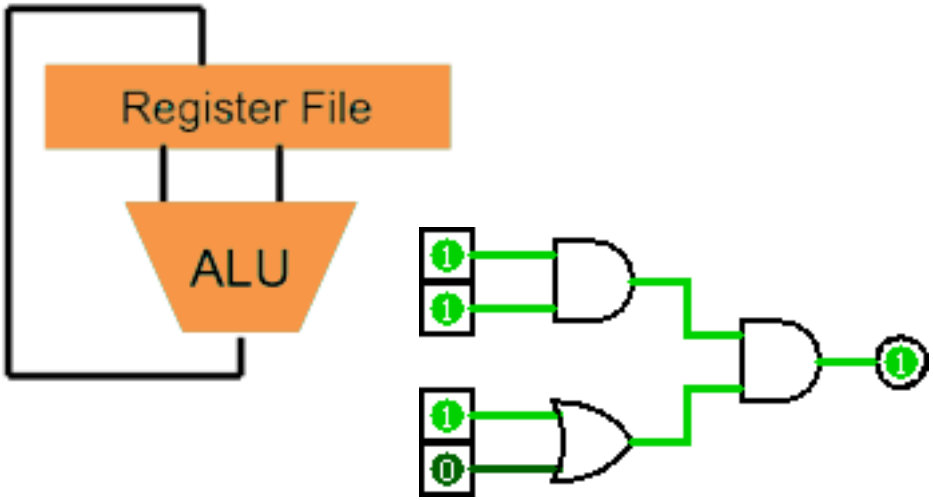
Idea #1: Levels of Representation




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

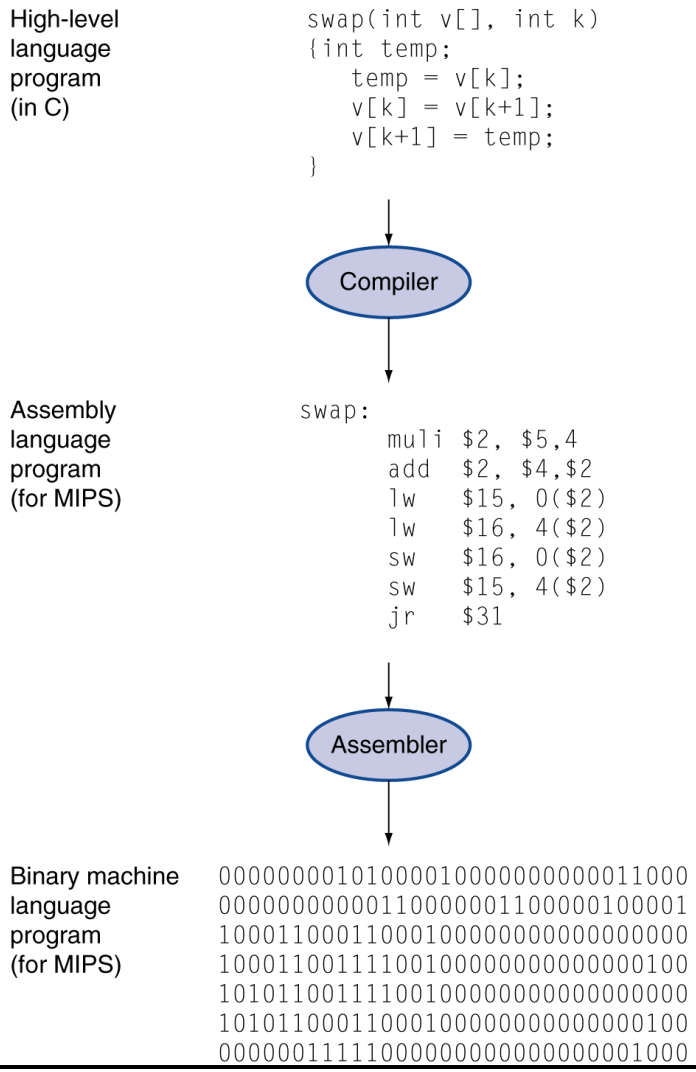
```
lw    $t0, 0($2)
lw    $t1, 4($2)
sw    $t1, 0($2)
sw    $t0, 4($2)
```

```
0000 1001 1100 0110 1010 1111 0101 1000
1010 1111 0101 1000 0000 1001 1100 0110
1100 0110 1010 1111 0101 1000 0000 1001
0101 1000 0000 1001 1100 0110 1010 1111
```

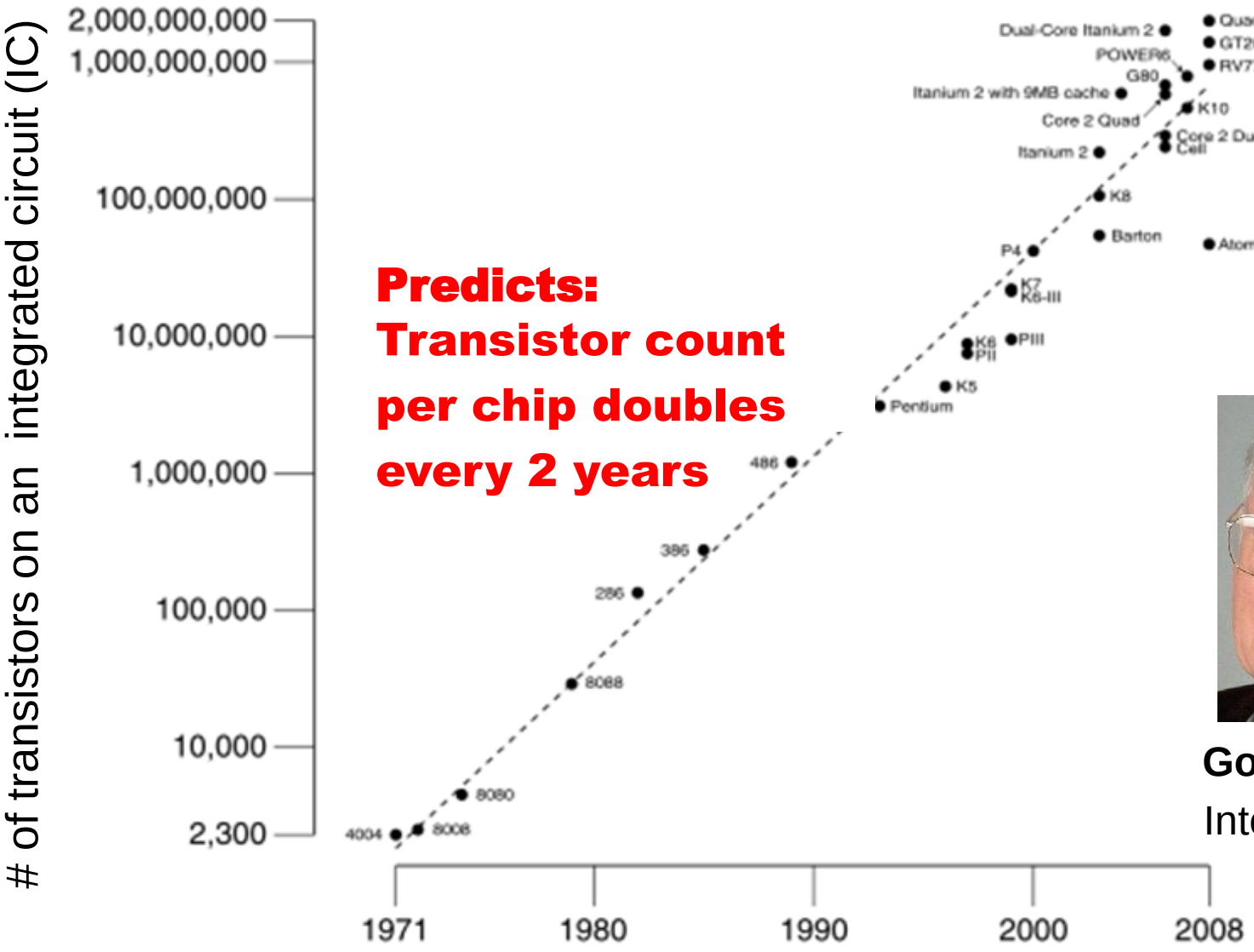


Levels of Program Code

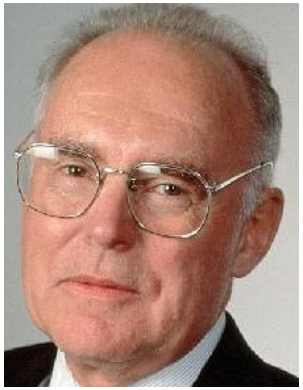
- High-level language
  - Level of abstraction closer to problem domain
  - Provides for productivity and portability
- Assembly language
  - Textual representation of instructions
- Hardware representation
  - Binary digits (bits)
  - Encoded instructions and data



Idea #2: Technology Trends

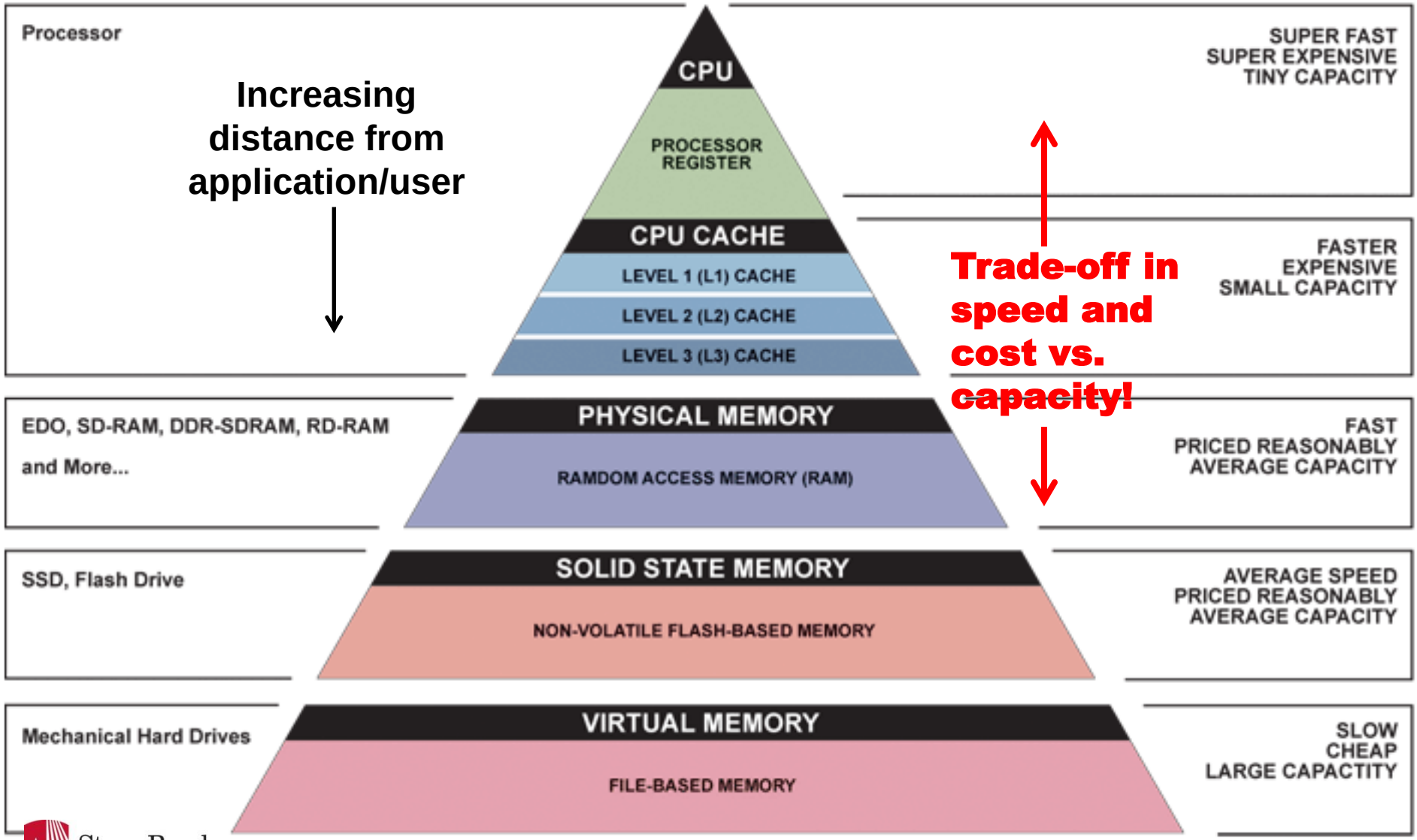


Predicts:
Transistor count per chip doubles every 2 years

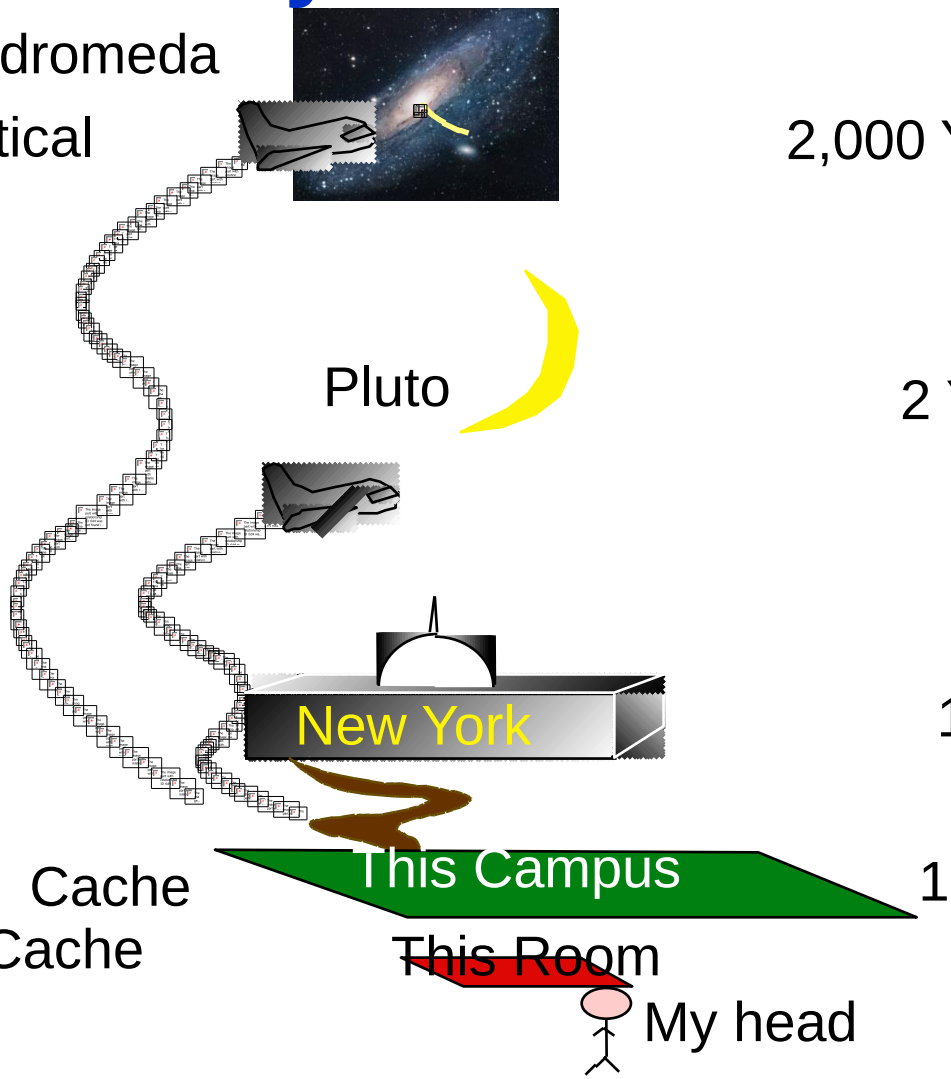


Gordon Moore
Intel Cofounder

Idea #3: Principle of Locality/ Memory Hierarchy



Storage Latency Analogy: How Far Away is the Data?



10^9	Tape /Optical Robot	2,000 Years
10^6	Disk	2 Years
100	Memory	1.5 hr
10	On Board Cache	10 min
2	On Chip Cache	
1	Registers	1 min

(ns)

Memory Technology Costs

- Static RAM (SRAM)
  - 0.5ns – 2.5ns, $500 – $1000 per GB
- Dynamic RAM (DRAM)
  - 50ns – 70ns, $3 – $6 per GB
- Magnetic disk
  - 5ms – 20ms, $0.01 – $0.02 per GB
- Ideal memory
  - Access time of SRAM
  - Capacity and cost/GB of disk

Idea #4: Parallelism

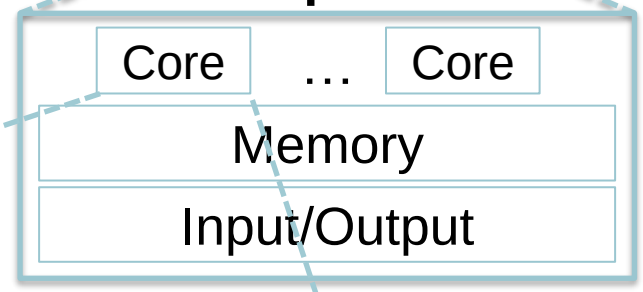
Software *Hardware*

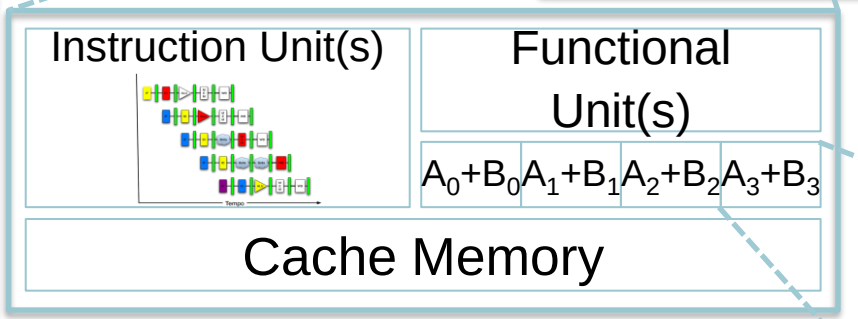
- Parallel Requests
  - Assigned to computer
  - e.g. search “Garcia”
- Parallel Threads
  - Assigned to core
  - e.g. lookup, ads
- Parallel Instructions
  - > 1 instruction @ one time
  - e.g. 5 pipelined instructions
- Parallel Data
  - > 1 data item @ one time
  - e.g. add of 4 pairs of words
- Hardware descriptions
  - All gates functioning in parallel at same time

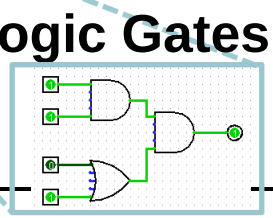
Leverage Parallelism & Achieve High Performance

Warehouse Scale Computer

Smart Phone





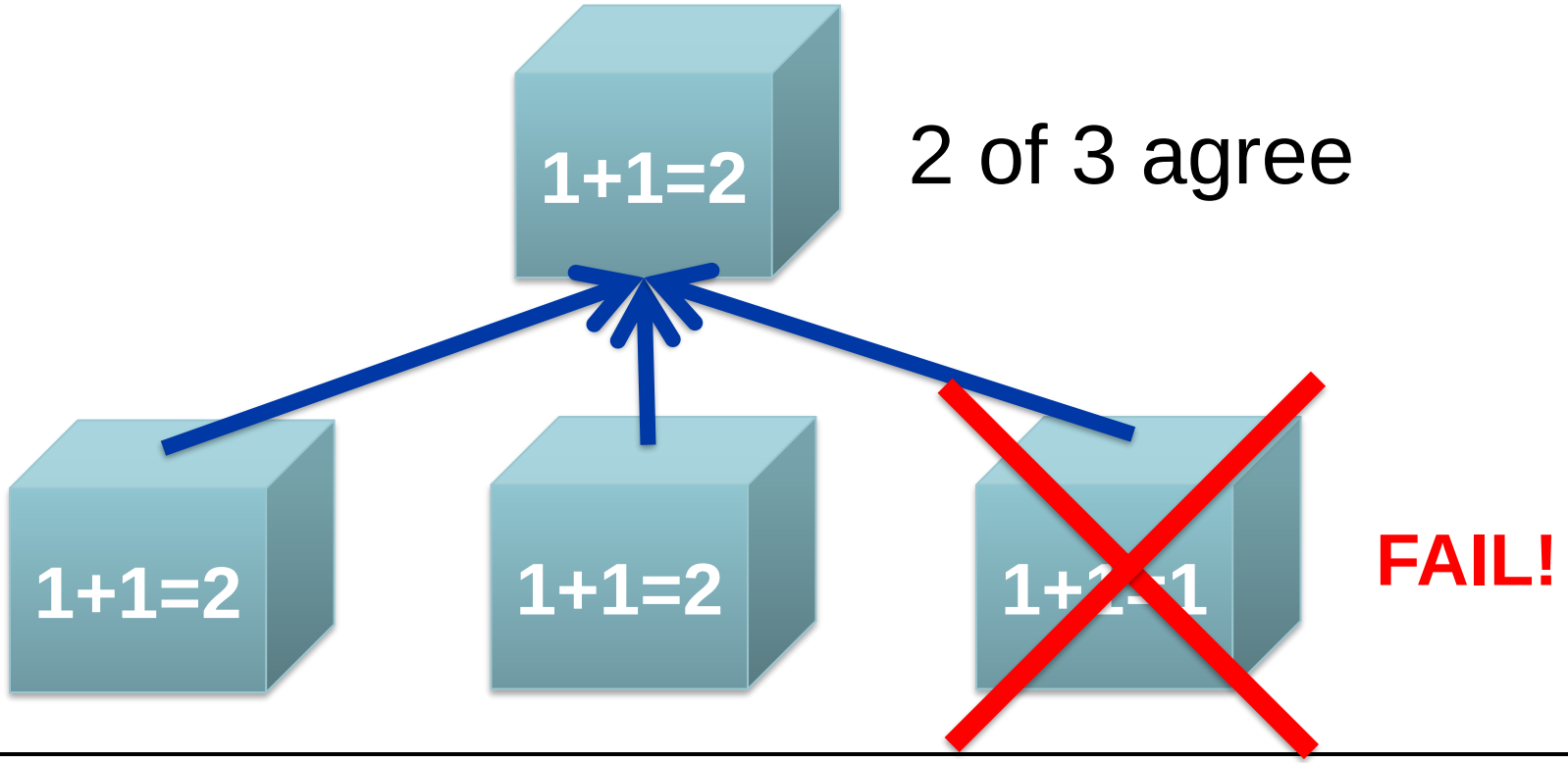


Idea #5: Performance Measurement and Improvement

- Allows direct comparisons of architectures and quantification of improvements
- Most common measures are *time to finish* (latency) and *rate of execution* (throughput)
- Match application and hardware to exploit:
  - Locality, parallelism, special hardware features

Idea #6: Dependability via Redundancy

- Redundancy so that a failing piece doesn't make the whole system fail

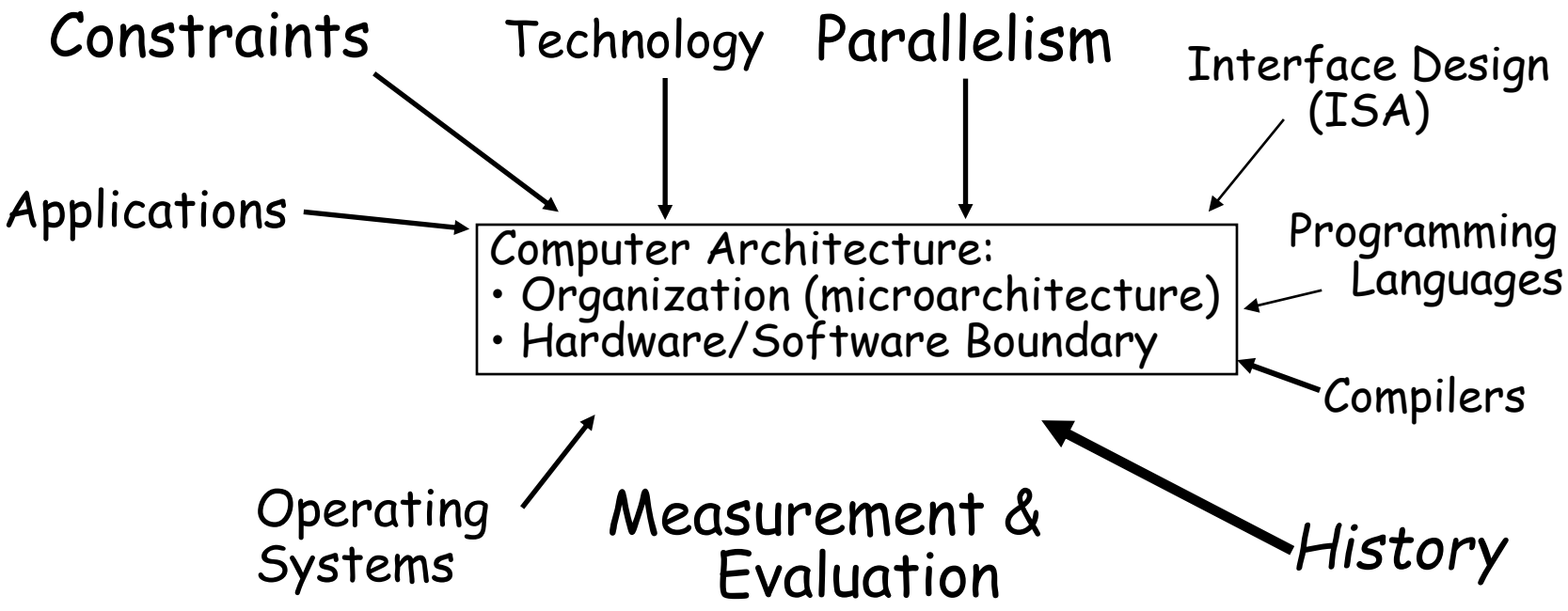


Idea #6: Dependability via Redundancy

- Applies to everything from datacenters to storage to memory
  - 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 of so that can lose 1 bit but no data (Error Correcting Code/ECC Memory)
- Increasing transistor density reduces the cost of redundancy

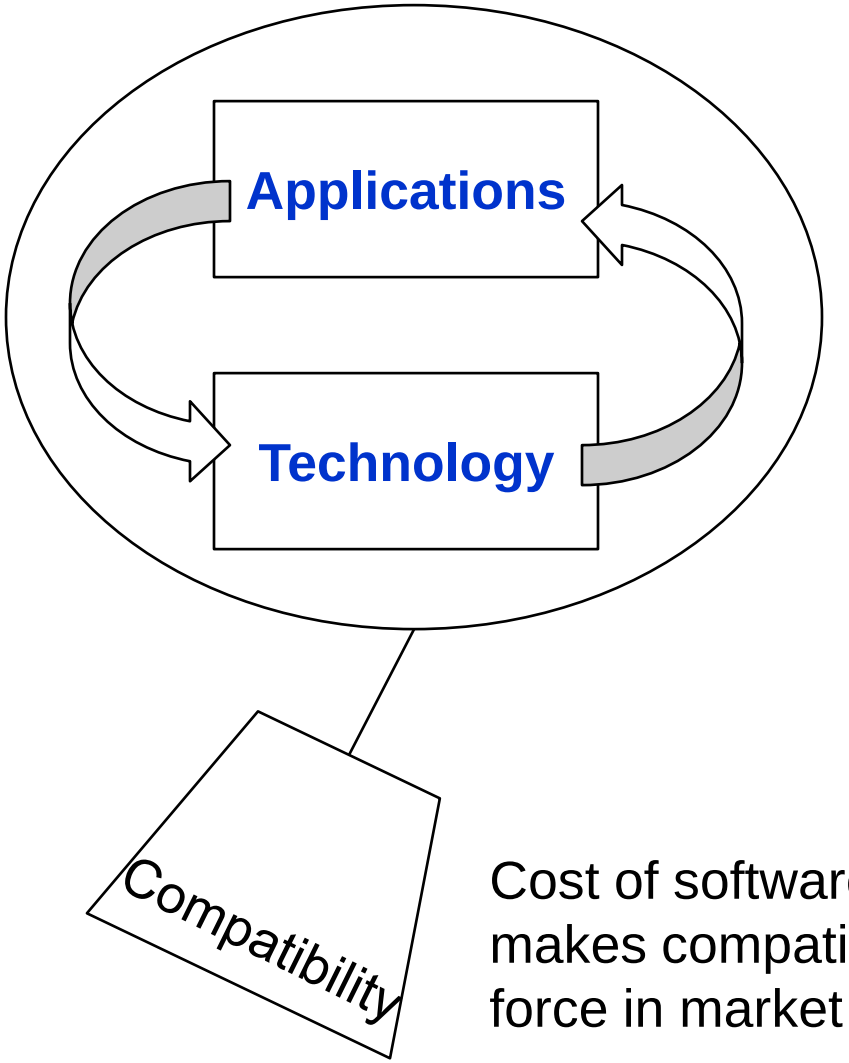
Computer Architecture Course Focus

Understanding the design techniques, machine structures, technology factors, evaluation methods that will determine the form of computers in 21st Century



Architecture Continually Changing

Applications suggest how to improve technology, provide revenue to fund development

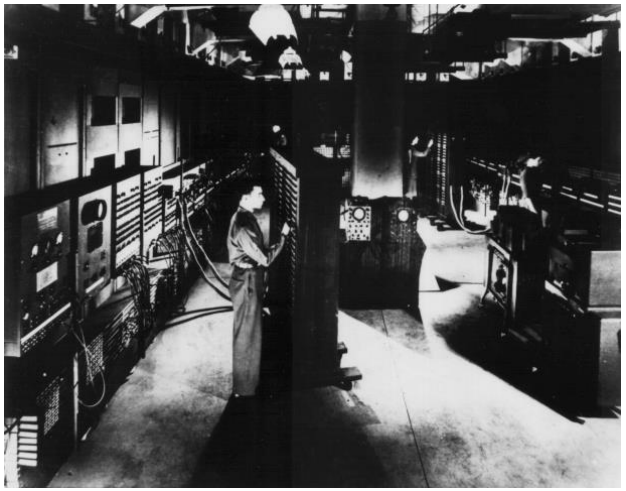


Improved technologies make new applications possible

Cost of software development makes compatibility a major force in market

Electronic Numerical Integrator and Computer (ENIAC)

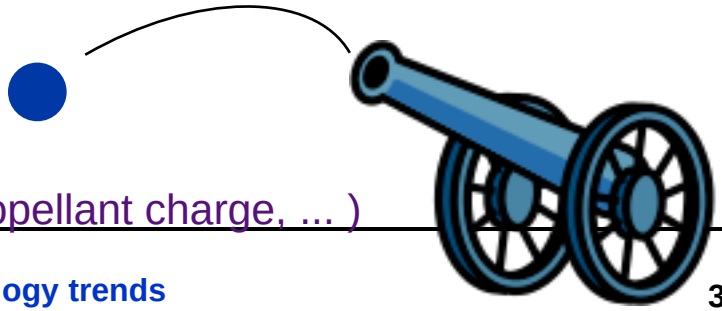
- Inspired by Atanasoff and Berry, Eckert and Mauchly designed and built ENIAC (1943-45) at the University of Pennsylvania
- The first, completely electronic, operational, general-purpose analytical calculator!
  - 30 tons, 72 square meters, 200KW
  - Each of the twenty 10 digit registers was 2 feet long
  - Used 18,000 vacuum tubes
- Performance
  - Read in 120 cards per minute
  - Addition took 200 ms, Division 6 ms
  - 1000 times faster than Mark I
- Not very reliable!



The slogan was: The ENIAC is able to calculate the trajectory of a large-caliber naval shell in less time than the shell takes to reach its target!

Application: Ballistic calculations

angle = f (location, tail wind, cross wind,
air density, temperature, weight of shell, propellant charge, ...)

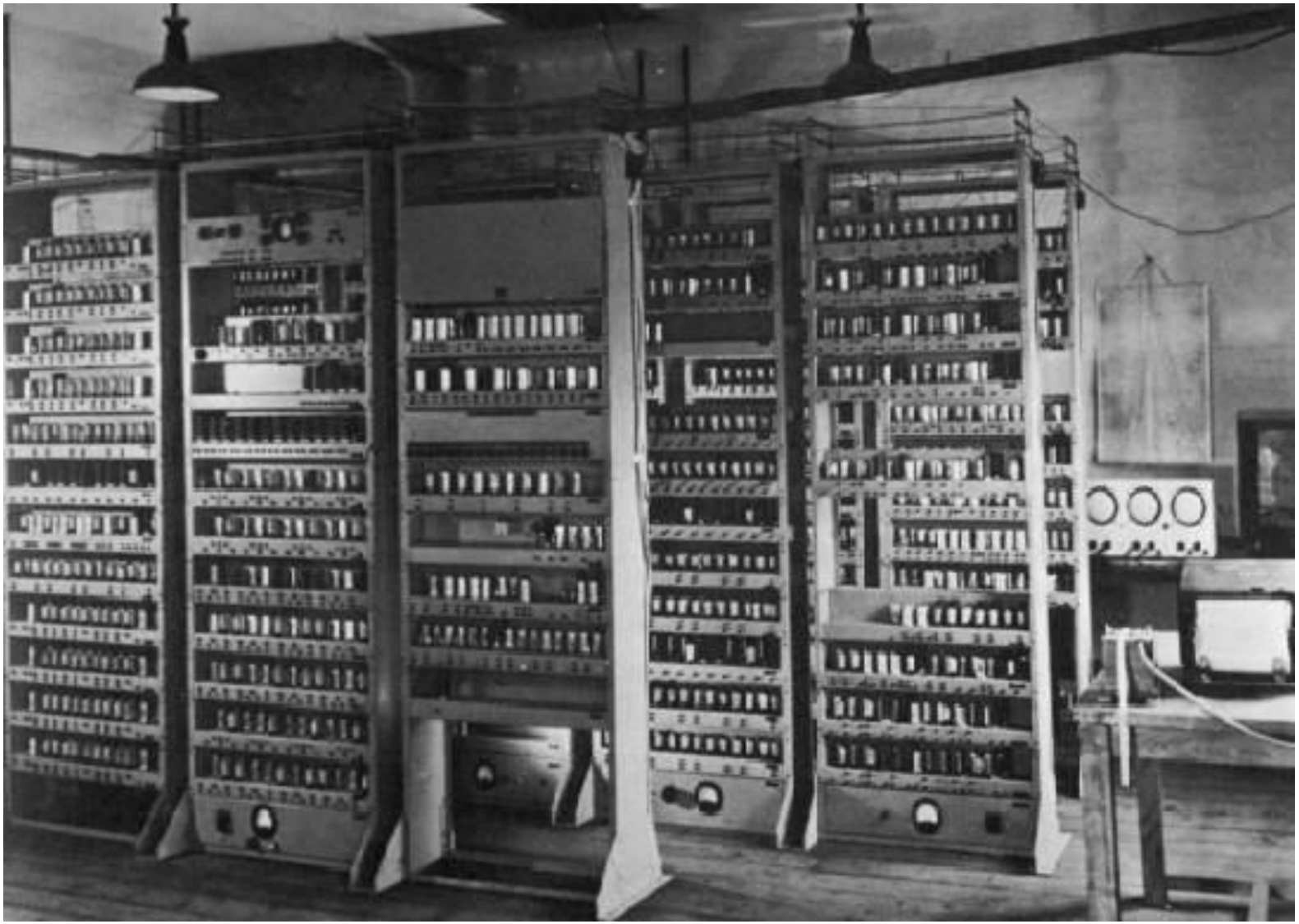


Electronic Discrete Variable Automatic Computer (EDVAC)

- ENIAC's programming system was external
  - Sequences of instructions were executed independently of the results of the calculation
  - Human intervention required to take instructions "out of order"
- Eckert, Mauchly, John von Neumann and others designed EDVAC (1944) to solve this problem
  - Solution was the *stored program computer*

    $\Rightarrow$ "*program can be manipulated as data*"
- *First Draft of a report on EDVAC* was published in 1945, but just had von Neumann's signature!
  - In 1973 the court of Minneapolis attributed the honor of *inventing the computer* to John Atanasoff

EDSAC, University of Cambridge, UK, 1949



Dominant Problem: *Reliability*

Mean time between failures (MTBF)

MIT's Whirlwind with an MTBF of 20 min. was perhaps the most reliable machine !

Reasons for unreliability:

1. Vacuum Tubes

2. Storage medium
   - acoustic delay lines
   - mercury delay lines
   - Williams tubes
   - Selections

Reliability solved by invention of Core memory by J. Forrester 1954 at MIT for Whirlwind project

Technology Issues

ENIAC	⇒	EDVAC
18,000 tubes		4,000 tubes
20 10-digit numbers		2000 word storage mercury delay lines

ENIAC had many asynchronous parallel units but only one was active at a time

BINAC : Two processors that checked each other for reliability.

Didn't work well because processors never agreed

Computers in mid 50's

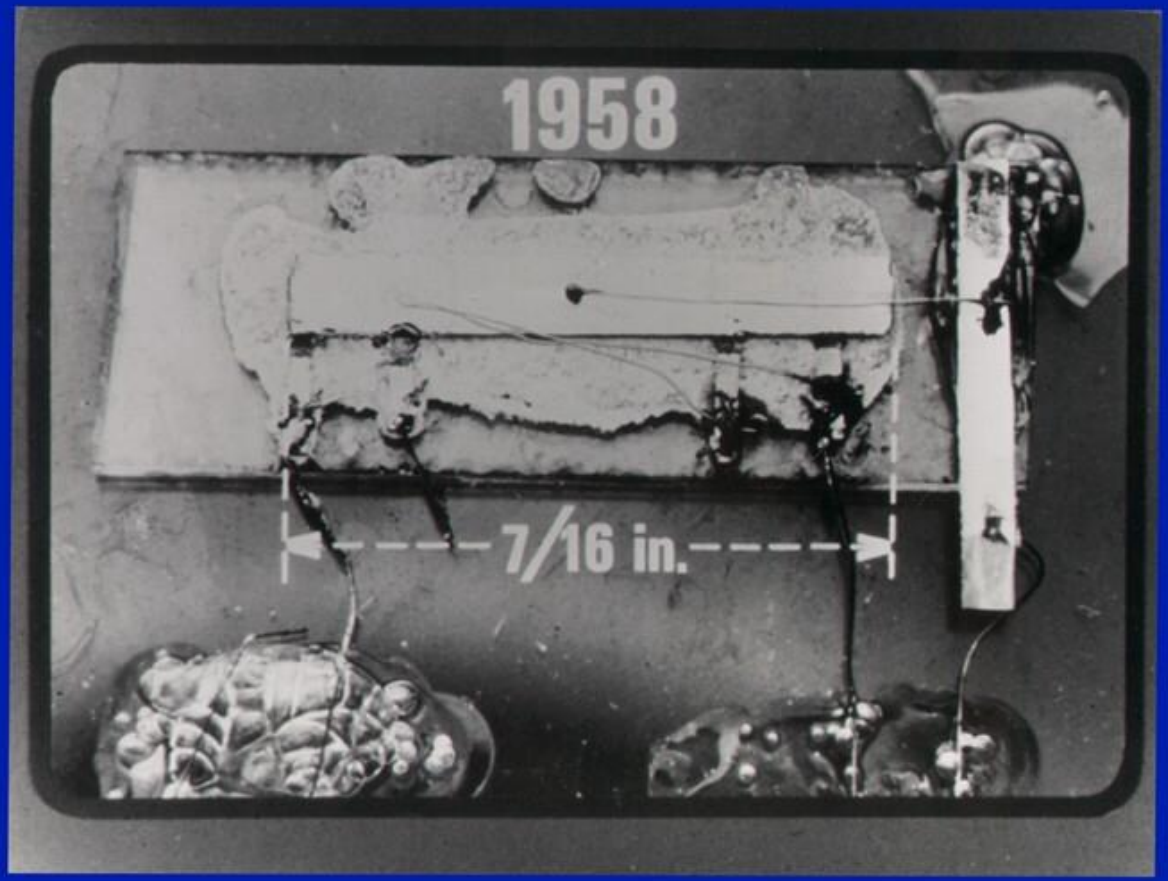
- Hardware was expensive
- Stores were small (1000 words)

  $\Rightarrow$ No resident system software!
- Memory access time was 10 to 50 times slower than the processor cycle

  $\Rightarrow$ Instruction execution time was totally dominated by the *memory reference time*.
- The *ability to design complex control circuits* to execute an instruction was the central design concern as opposed to *the speed* of decoding or an ALU operation
- Programmer's view of the machine was inseparable from the actual hardware implementation
  - No software abstraction layer

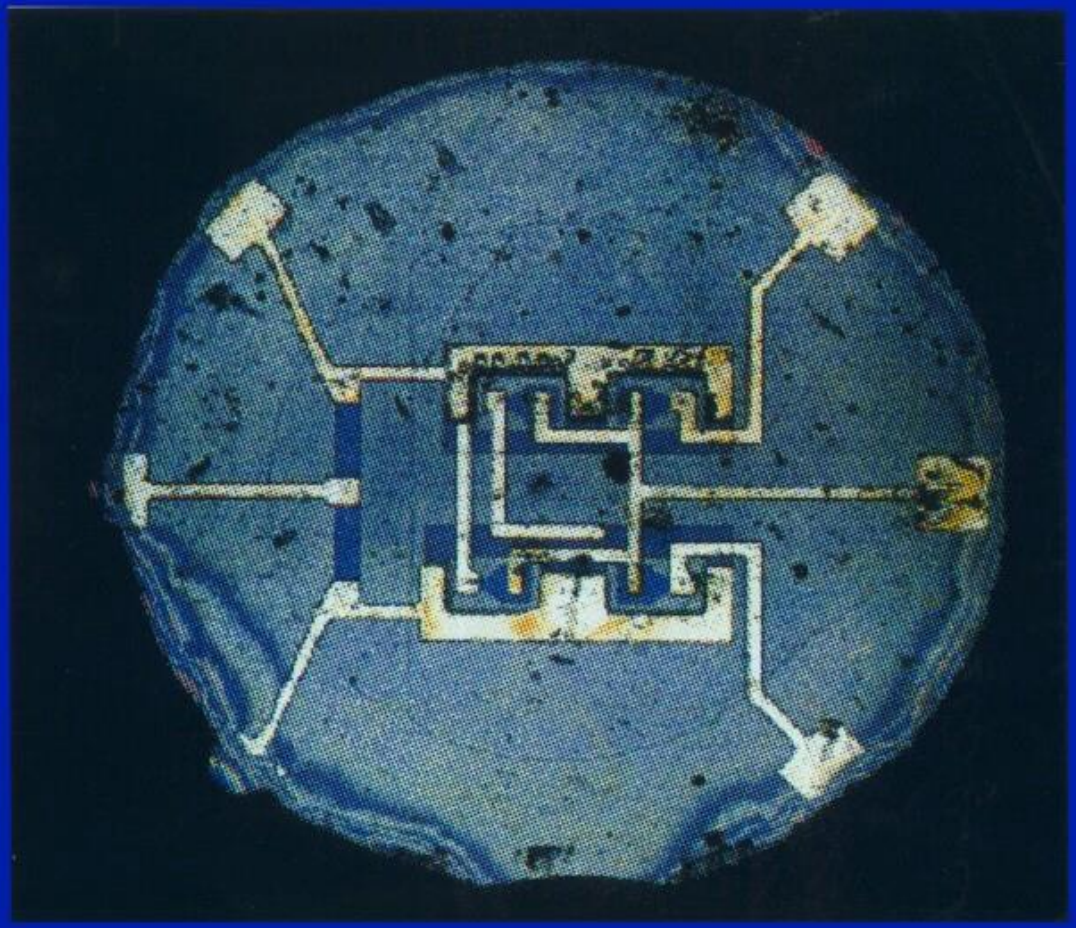
FIRST INTEGRATED CIRCUIT BY J. S. KILBY

(US Patent 3,138,763 filed Feb. 1959, granted 1964)



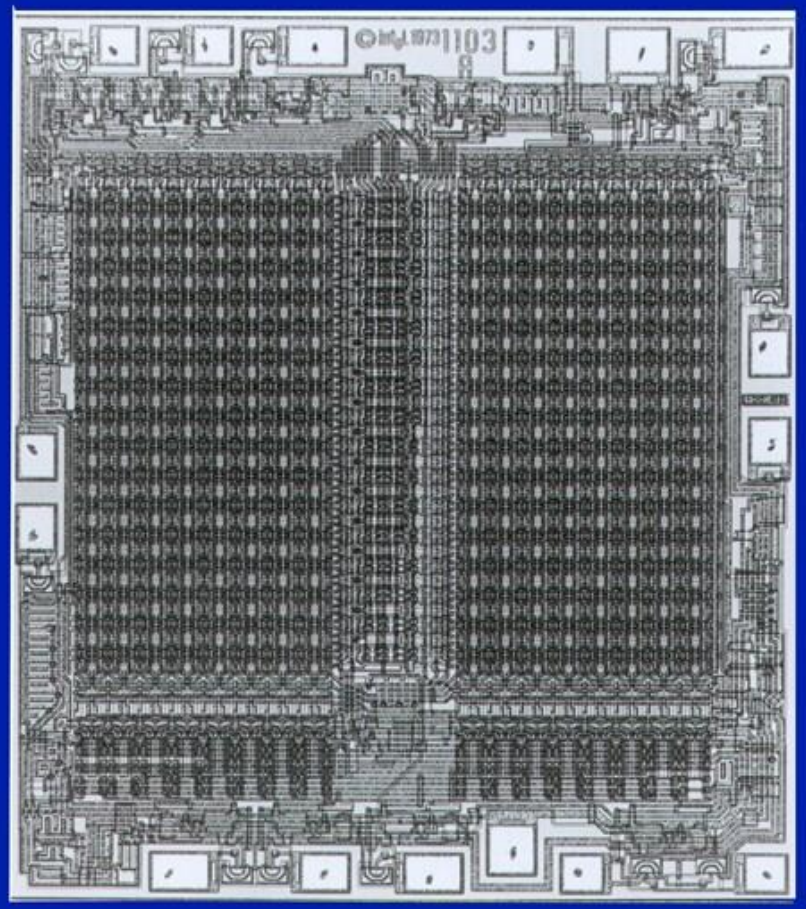
FIRST MONOLITHIC IC BY R. N. NOYCE

(US Patent 2,981,877 filed July 1959, granted 1961)



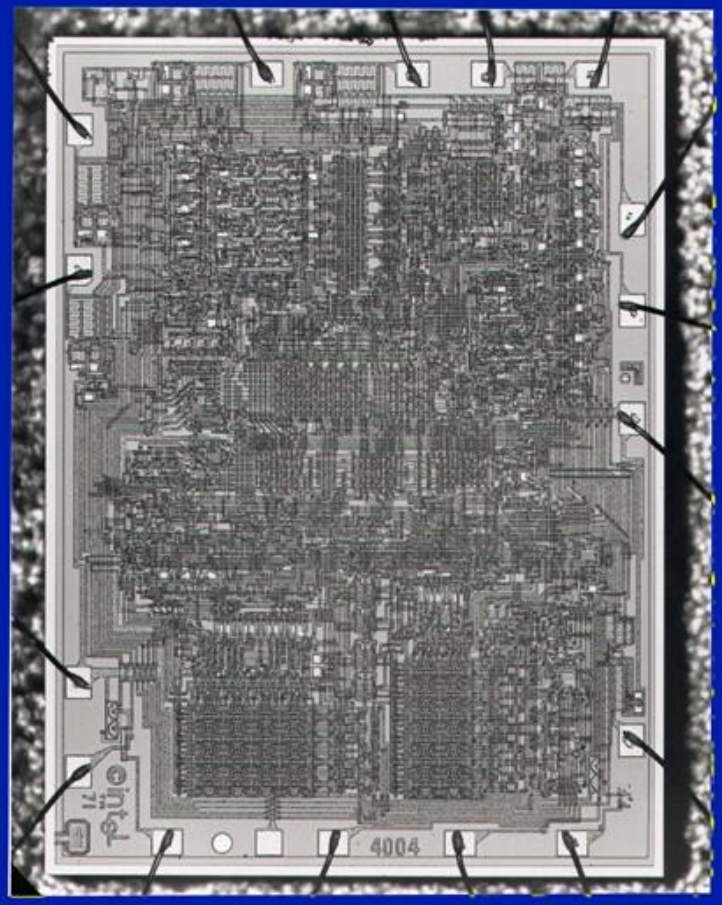
FIRST DRAM

(1103 by Intel 1970)



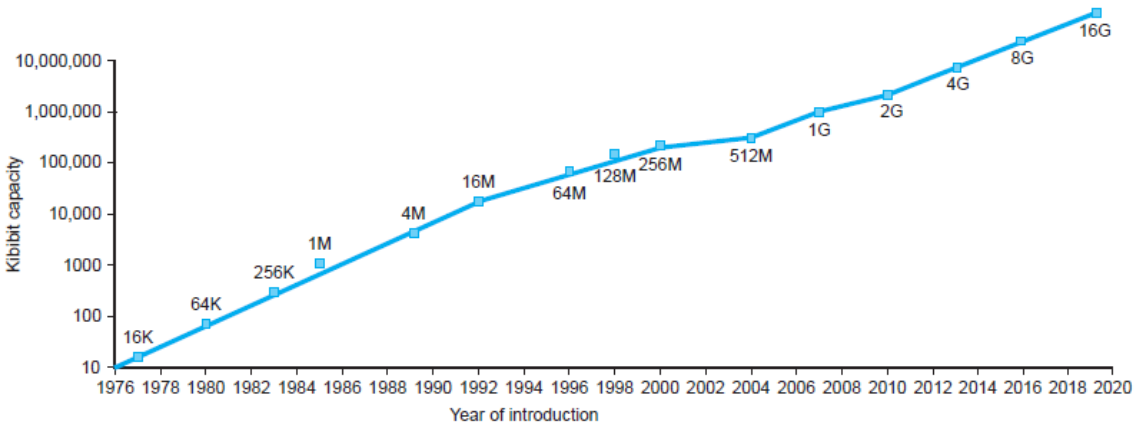
FIRST MICROPROCESSOR

(4004 by Intel 1971)



Technology Trends

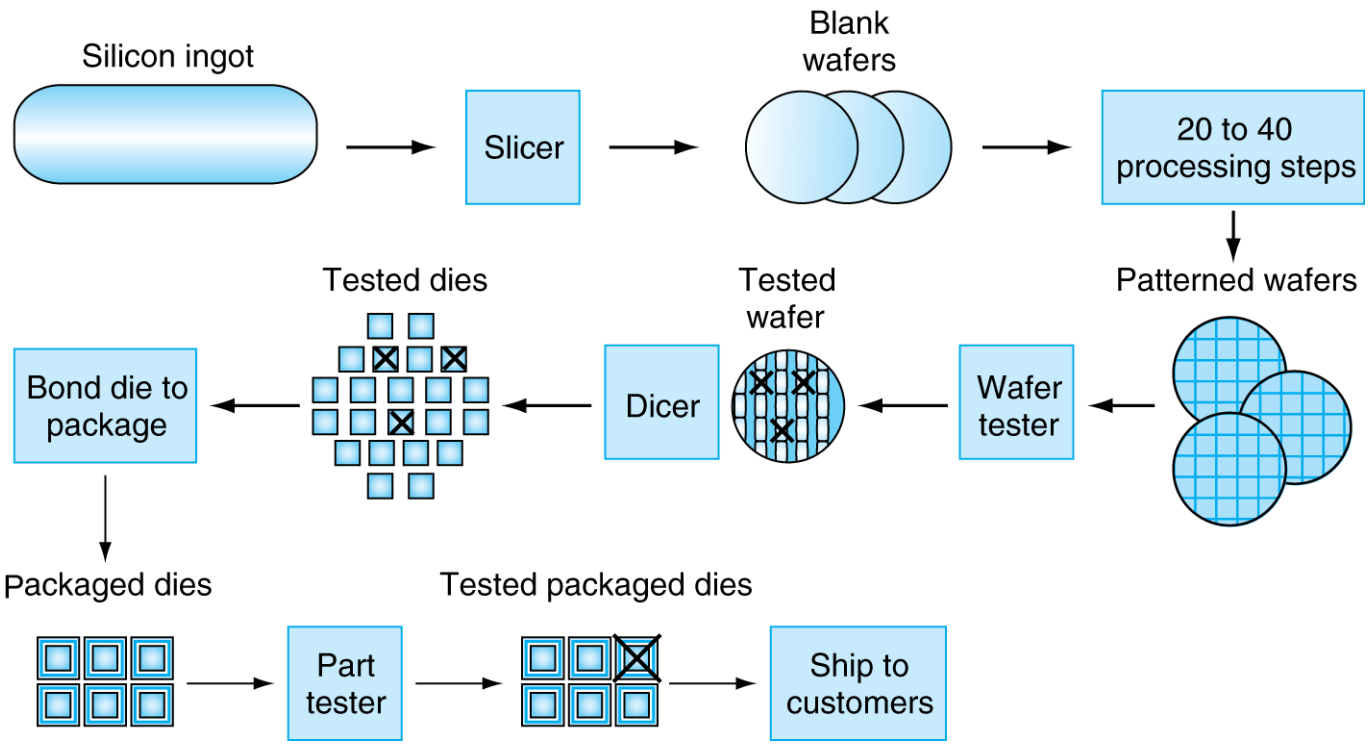
- Electronics technology continues to evolve
  - Increased capacity and performance
  - Reduced cost



DRAM capacity

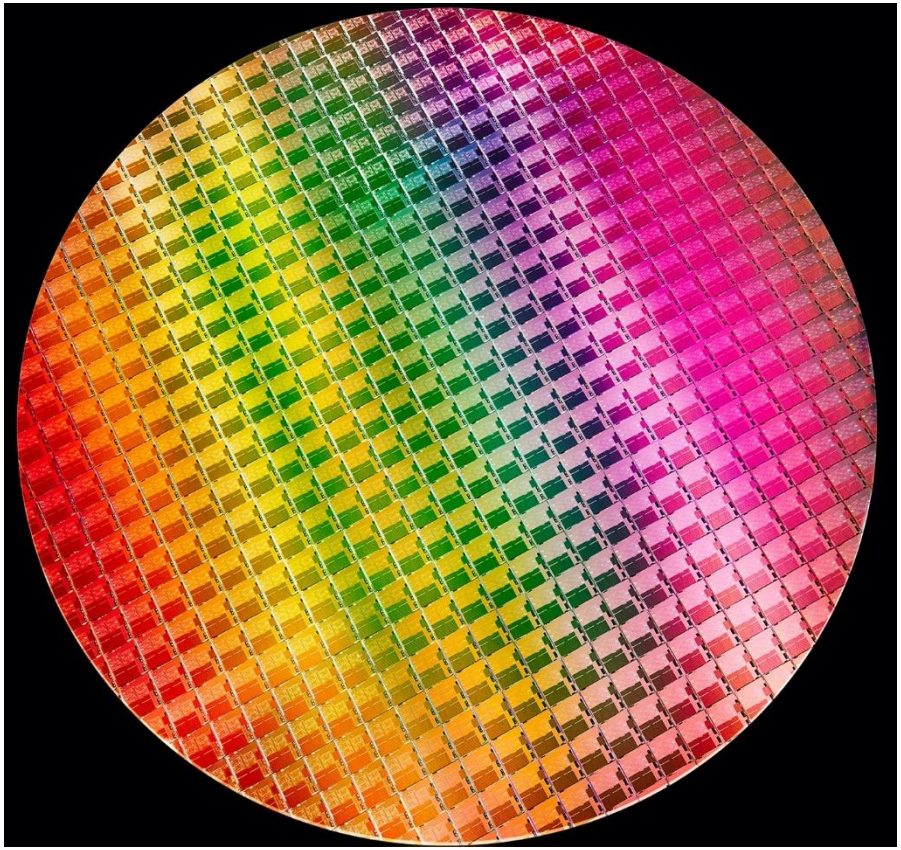
Year	Technology	Relative performance/cost
1951	Vacuum tube	1
1965	Transistor	35
1975	Integrated circuit (IC)	900
1995	Very large scale IC (VLSI)	2,400,000
2020	Ultra large scale IC	500,000,000,000

Manufacturing ICs



- Yield: proportion of working dies per wafer

Intel® Core 10th Generation



- 300mm wafer, 506 chips, 10nm technology
- Each chip is 11.4 x 10.7 mm

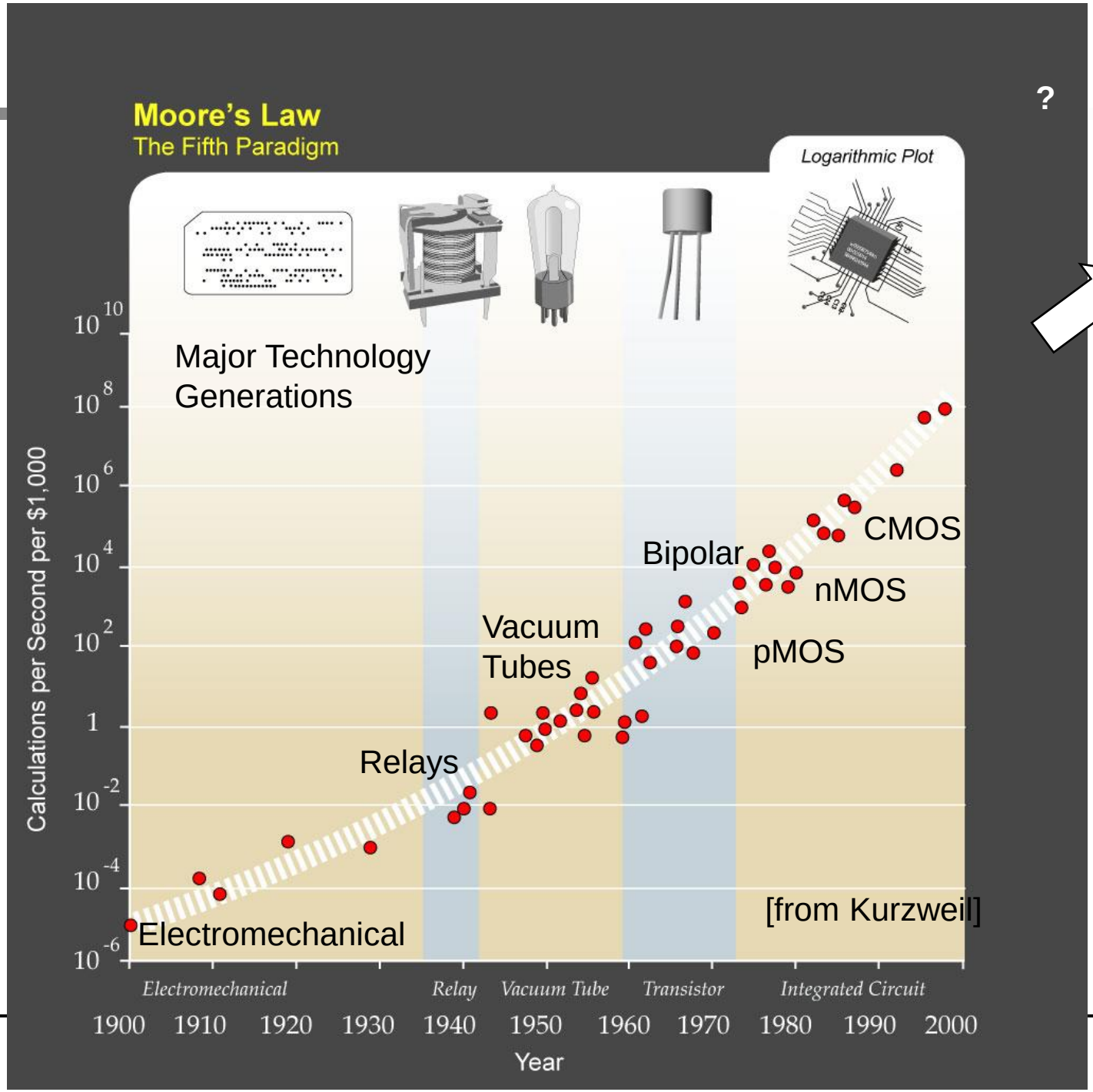
Integrated Circuit Cost

$$\text{Cost per die} = \frac{\text{Cost per wafer}}{\text{Dies per wafer} \times \text{Yield}}$$

$$\text{Dies per wafer} \approx \text{Wafer area}/\text{Die area}$$

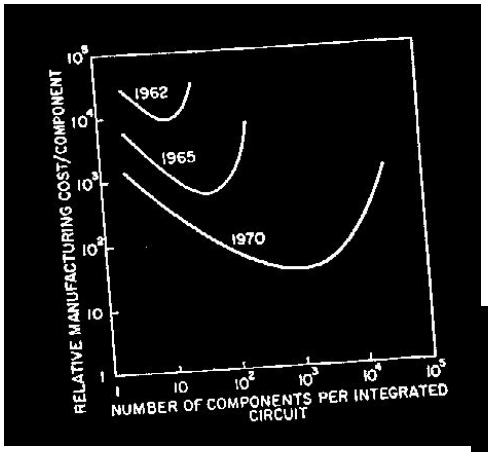
$$\text{Yield} = \frac{1}{(1+(\text{Defects per area} \times \text{Die area}/2))^2}$$

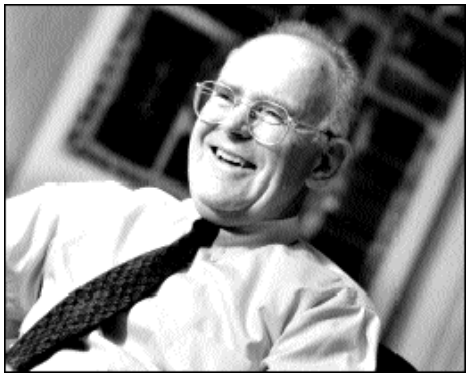
- Nonlinear relation to area and defect rate
  - Wafer cost and area are fixed
  - Defect rate determined by manufacturing process
  - Die area determined by architecture and circuit design

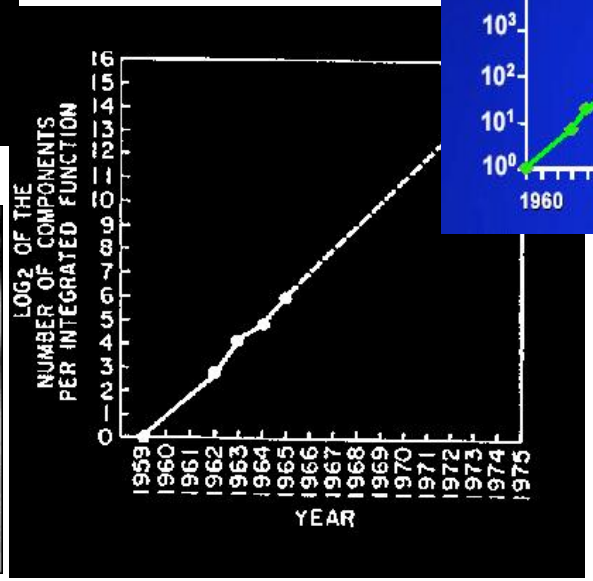

Moore's Law
The Fifth Paradigm
Logarithmic Plot
Major Technology Generations
Calculations per Second per $1,000
10^{10}
10^{8}
10^{6}
10^{4}
10^{2}
1
10^{-2}
10^{-4}
10^{-6}
Electromechanical
Relays
Vacuum Tubes
Bipolar
CMOS
nMOS
pMOS
[from Kurzweil]
Electromechanical
Relay
Vacuum Tube
Transistor
Integrated Circuit
1900
1910
1920
1930
1940
1950
1960
1970
1980
1990
2000
Year

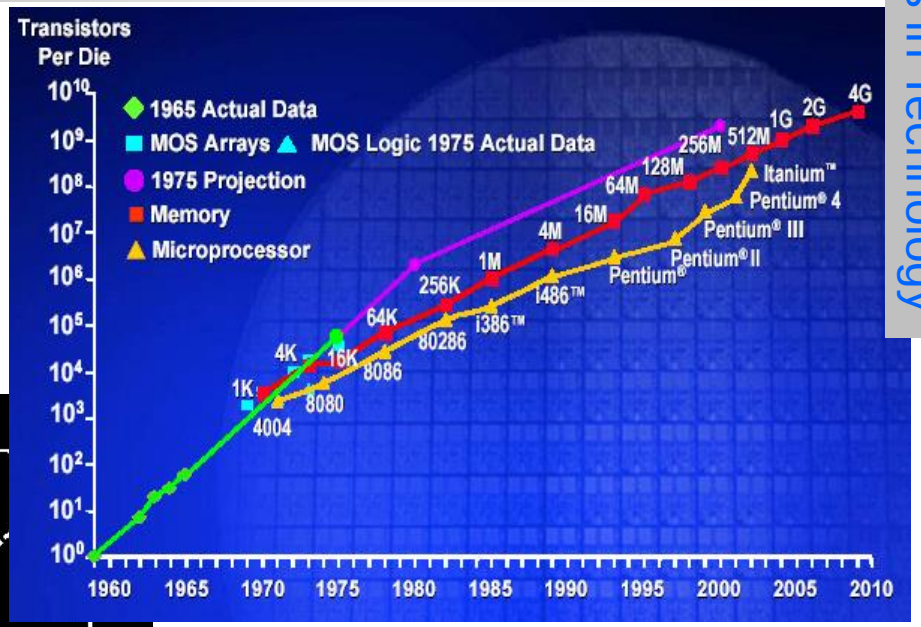
?

Moore's Law: 2X transistors / "year"



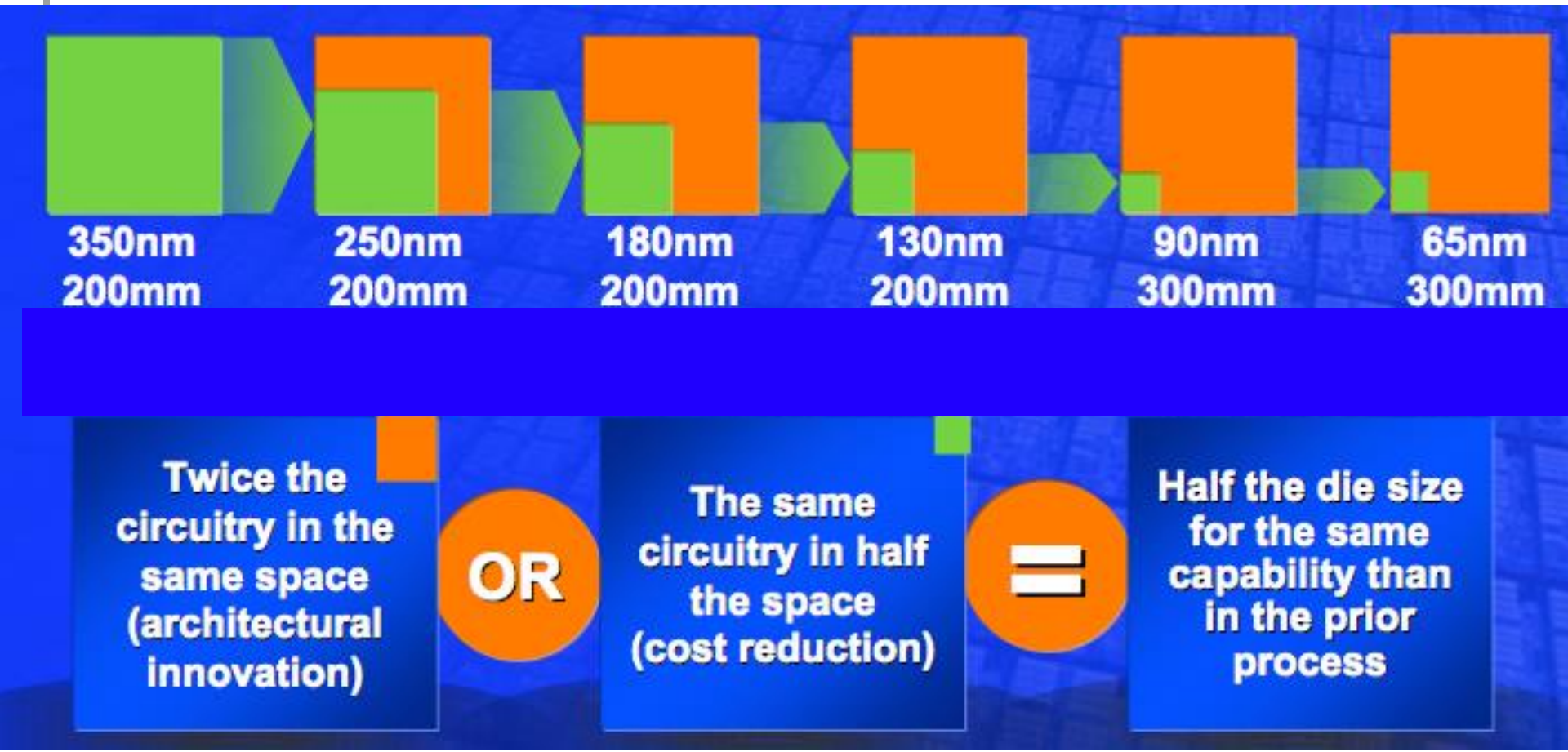






- "Cramming More Components onto Integrated Circuits"
  - Gordon Moore, Electronics, 1965
- # on transistors / cost-effective integrated circuit double every N months (12 ≤ N ≤ 24)

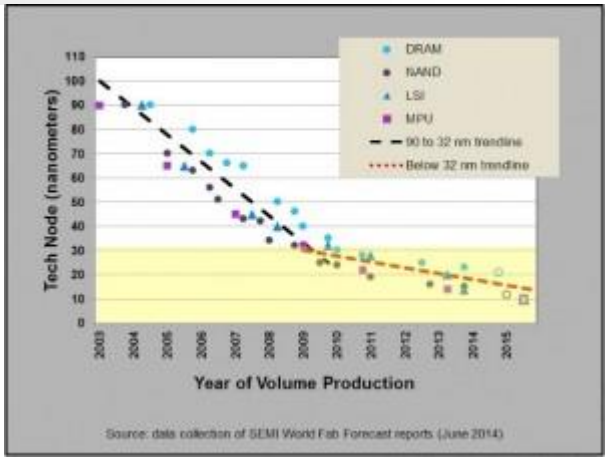
Main driver: device scaling ...



From: “Facing the Hot Chips Challenge Again”, Bill Holt, Intel, presented at Hot Chips 17, 2005.

Trends in Technology

- Integrated circuit technology
  - Transistor density: 35%/year (slowing)
  - Die size: 10-20%/year
  - Integration overall: 40-55%/year (slowing)
  - Moore's Law is no more!

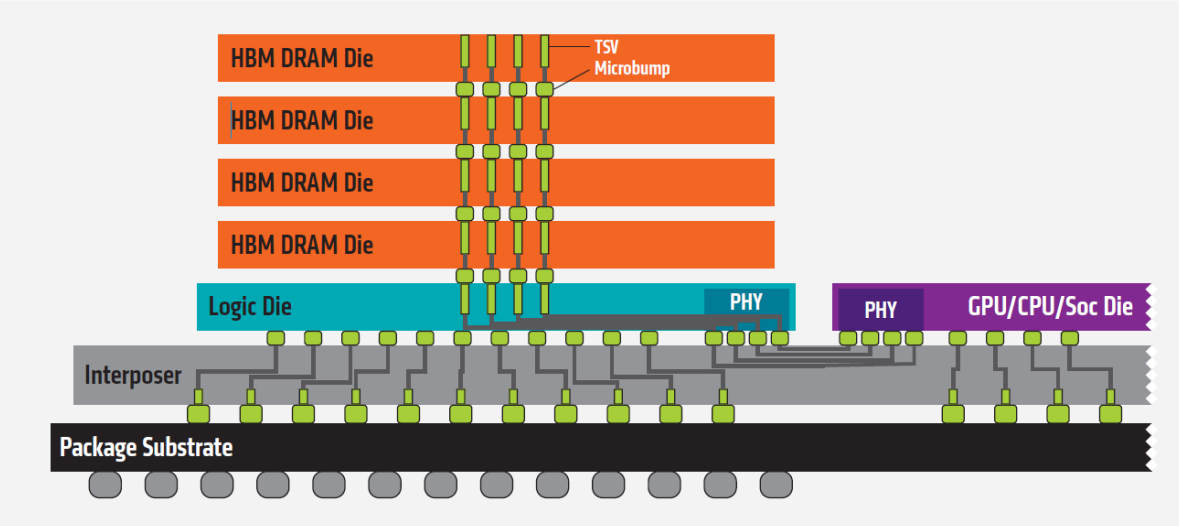


- DRAM capacity: was 25-40%/year (slowed dramatically now) (8Gbit in 2014, 16Gbit in 2019, 32Gbit ??)
- Flash capacity: 50-60%/year
  - 8-21X cheaper/bit than DRAM
- Magnetic disk technology: was 40%/year (~5% now)
  - 15-25X cheaper/bit then Flash
  - 300-500X cheaper/bit than DRAM

3D DRAM Stacking Technologies

Recent enabling technology: 3D stacking of DRAM chips

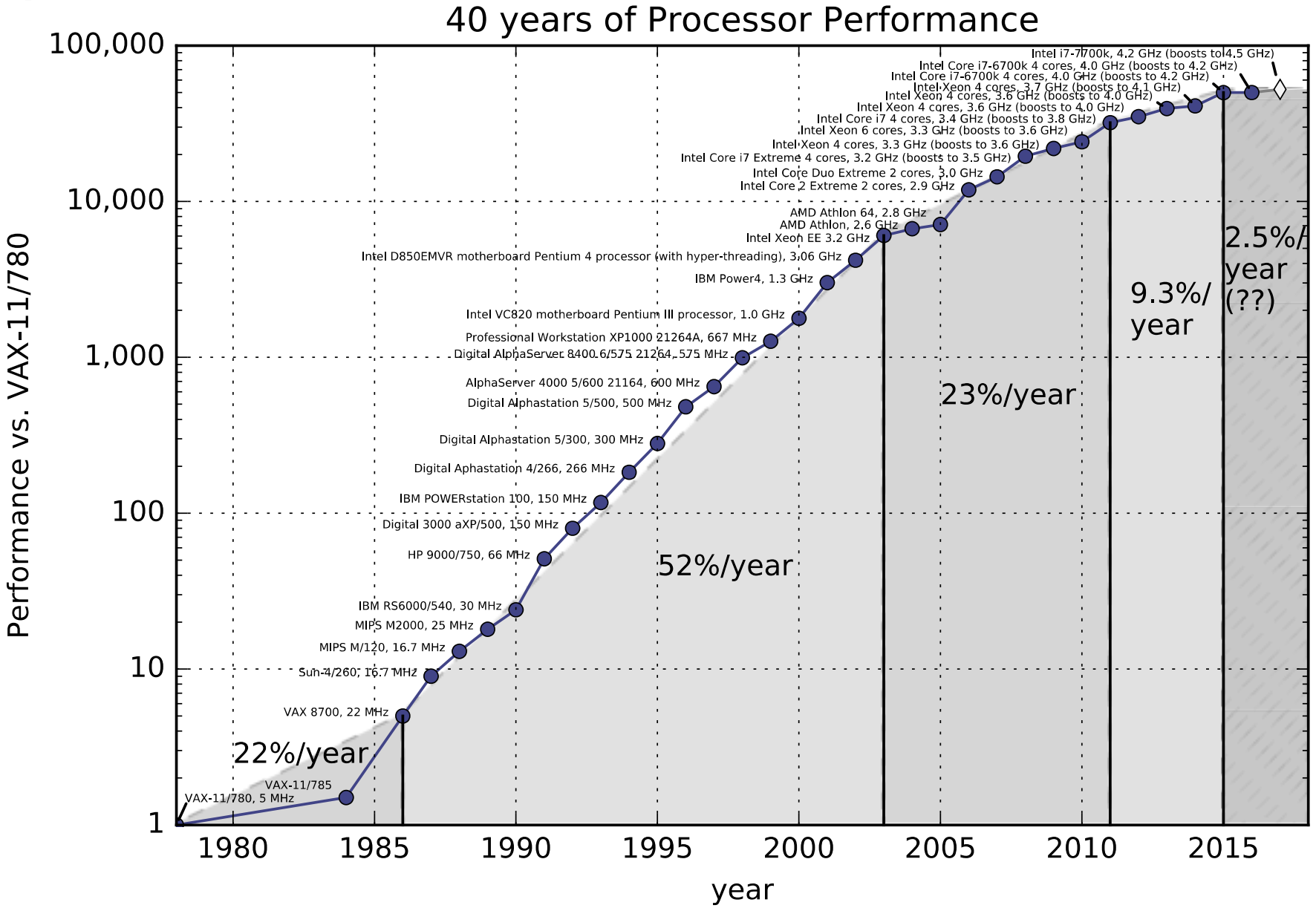
- DRAMs connected via through-silicon-vias (TSVs) that run through the chips
  - TSVs provide highly parallel connection between logic layer and DRAMs
- Base layer of stack "logic layer" is memory controller, manages requests from processor
- Silicon "interposer" serves as high-BW interconnect between DRAM stack and processor



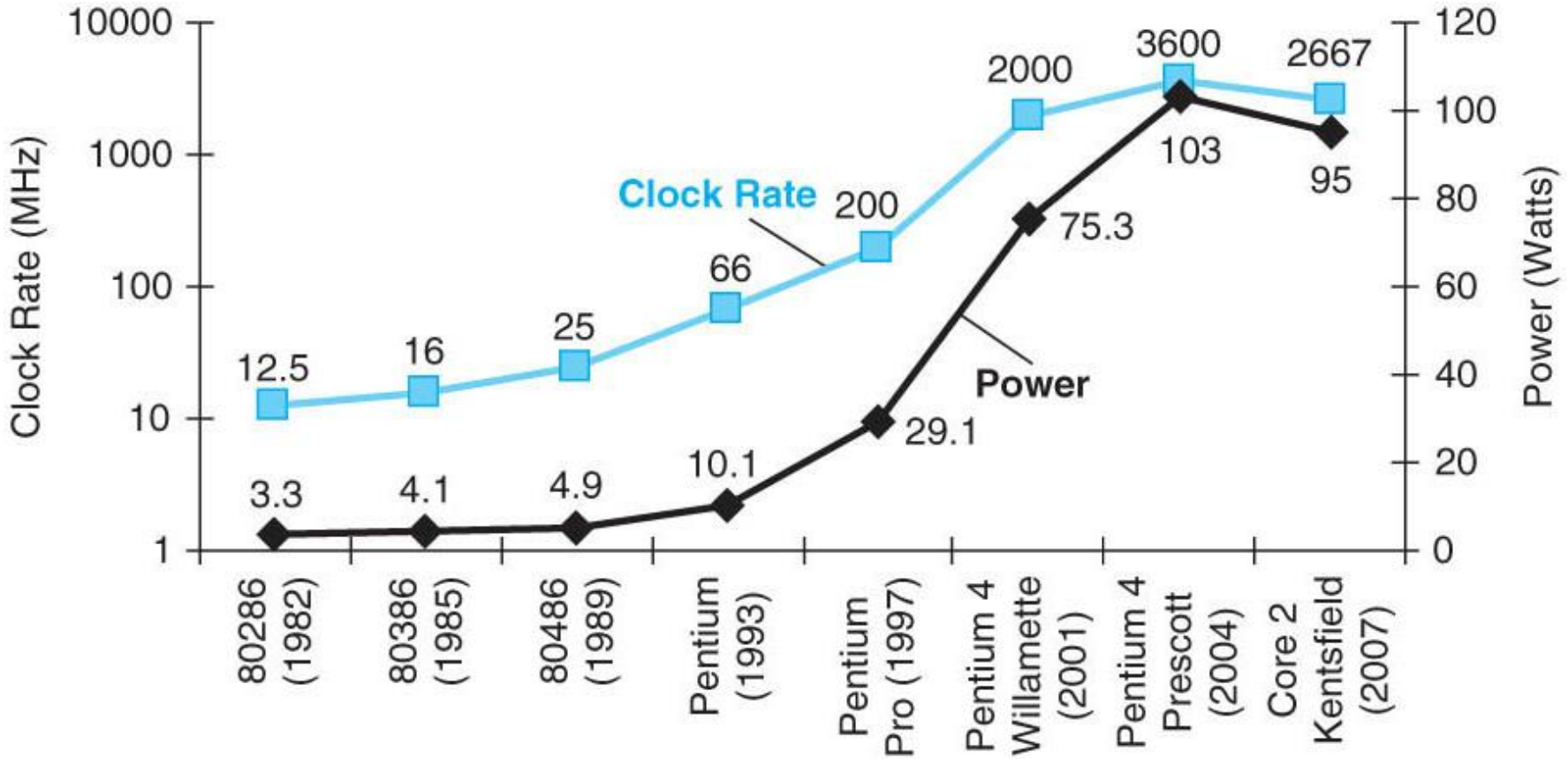
Technologies:

- Micron/Intel's Hybrid Memory Cube (HMC)
- AMD's High-Bandwidth Memory (HBM): 1024 bit interface to stack

Single-Thread Processor Performance

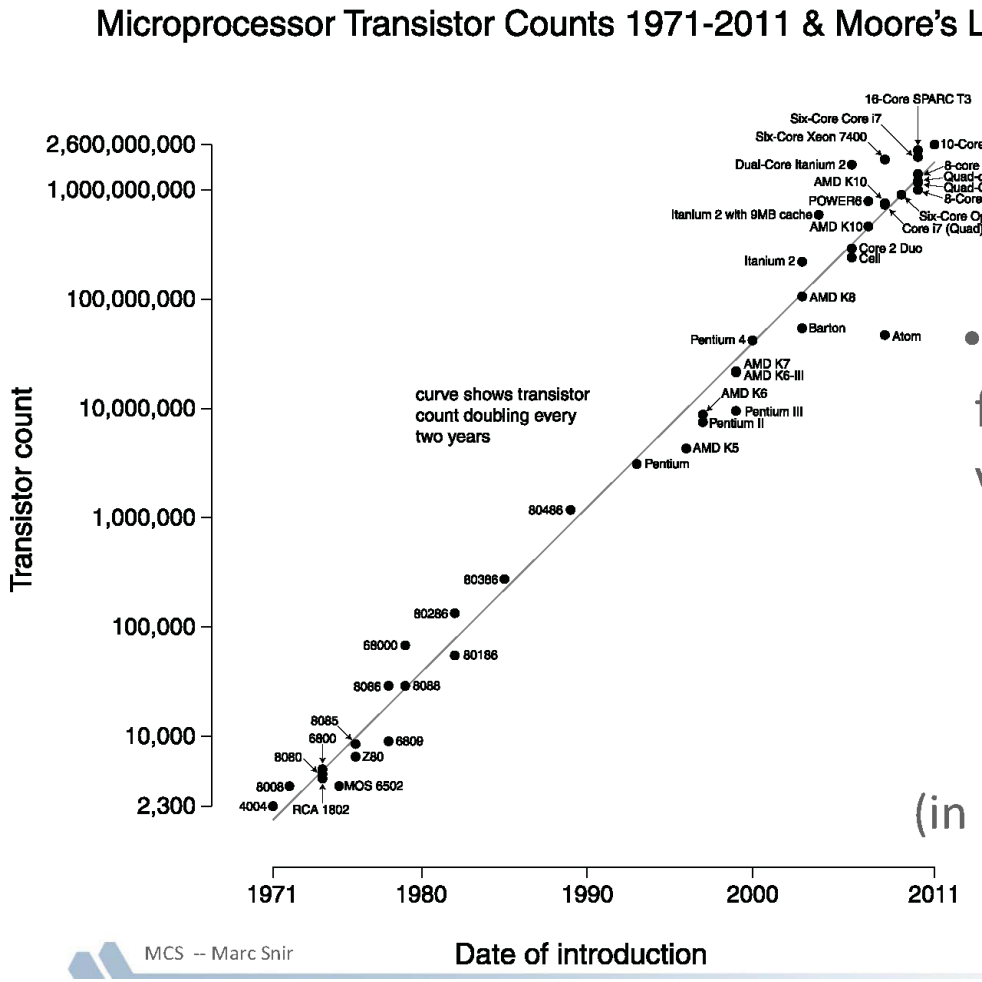


Limits to Performance: Faster Means More Power



The CMOS Age: Moore's Law & Dennard Scaling

Moore's Law: *The number of transistors per chip doubles every 2+ years*



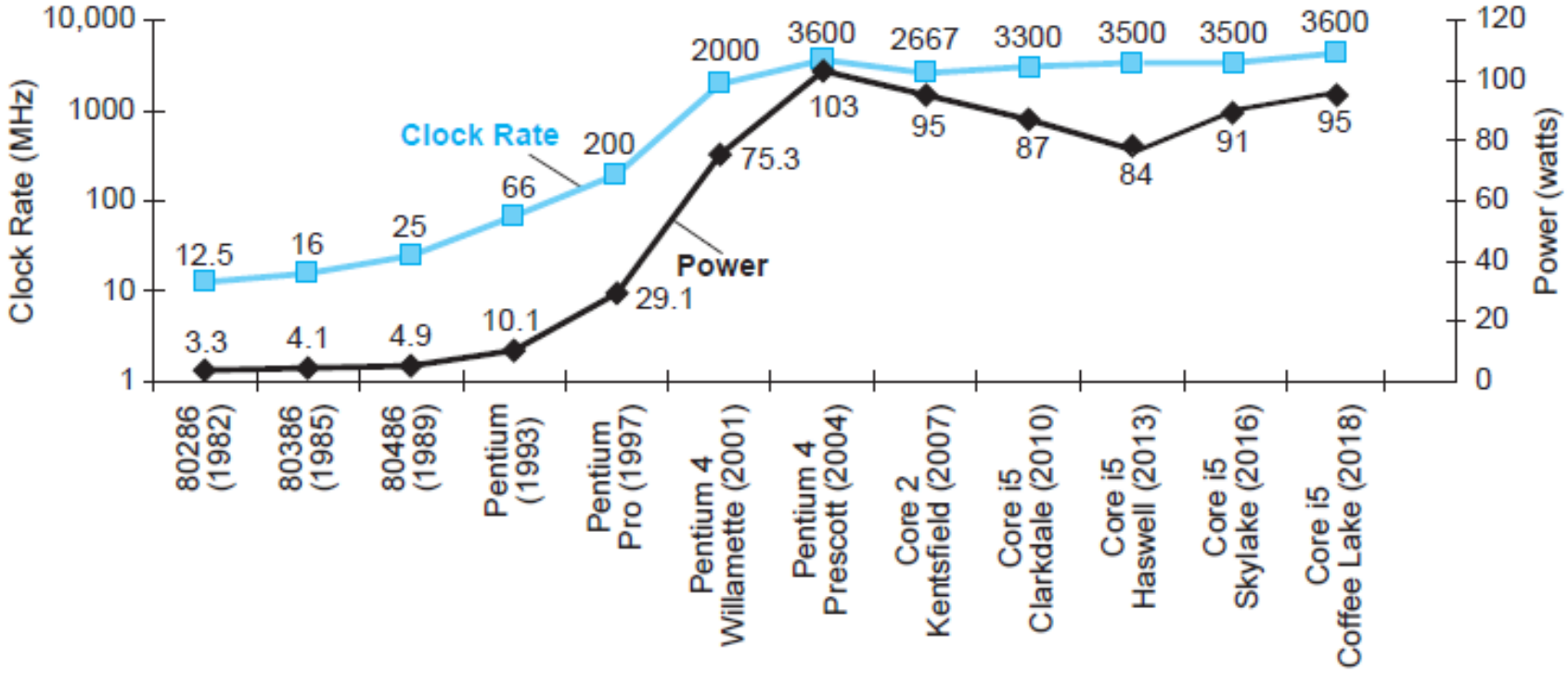
Dennard Scaling:

- Decrease feature size by a factor of λ and decrease voltage by a factor of λ; then
  - # transistors increase by λ^2
  - Clock speed increases by λ
  - Energy consumption does not change

(in reality, voltage decrease was slower; clock speed and energy consumption increased faster)

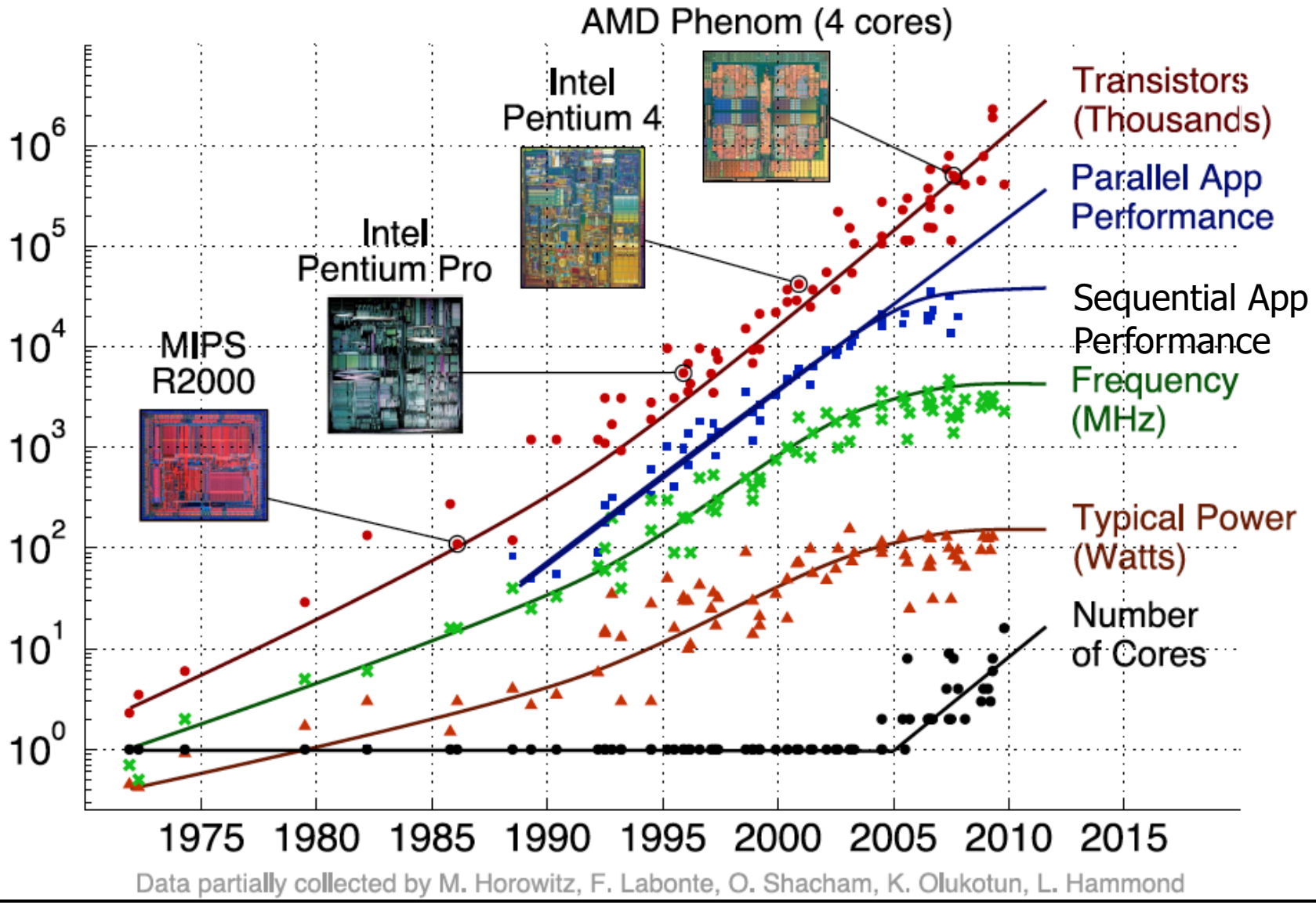
The Power Wall

Why? We can no longer fully scale voltage Vdd because MOS transistor leakage current is no longer a negligible part of the power budget.



Water Cooling in a Google Data Center

Transition to Multicore



Multicore Machines Today

Examples from Apple's product line:

Mac Pro
28 Intel Xeon W cores

iMac Pro
18 Intel Xeon W cores

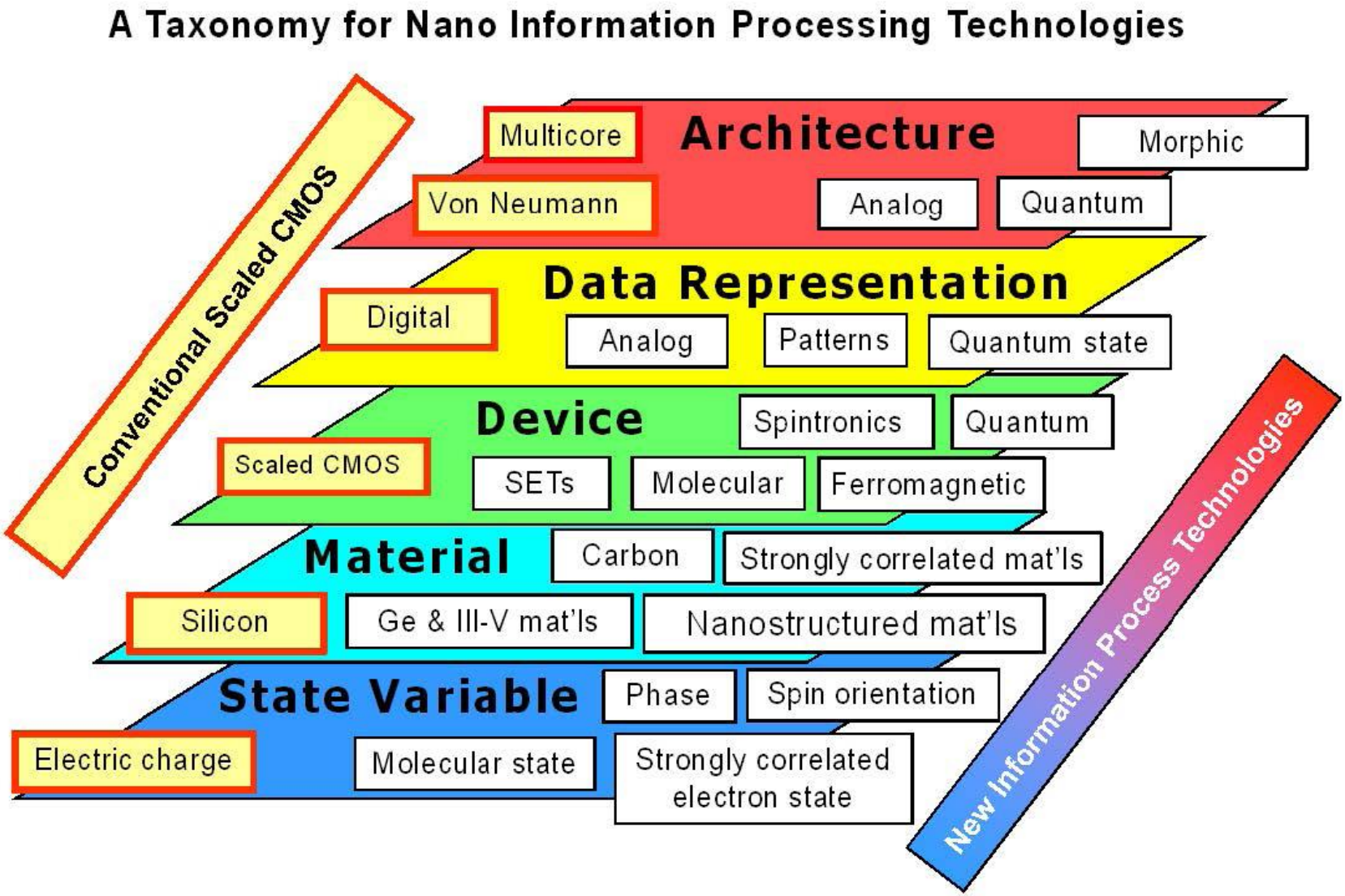
MacBook Pro Retina 15"
8 Intel Core i9 cores

iPad Pro
8 A12X cores
(4 fast + 4 low-power)

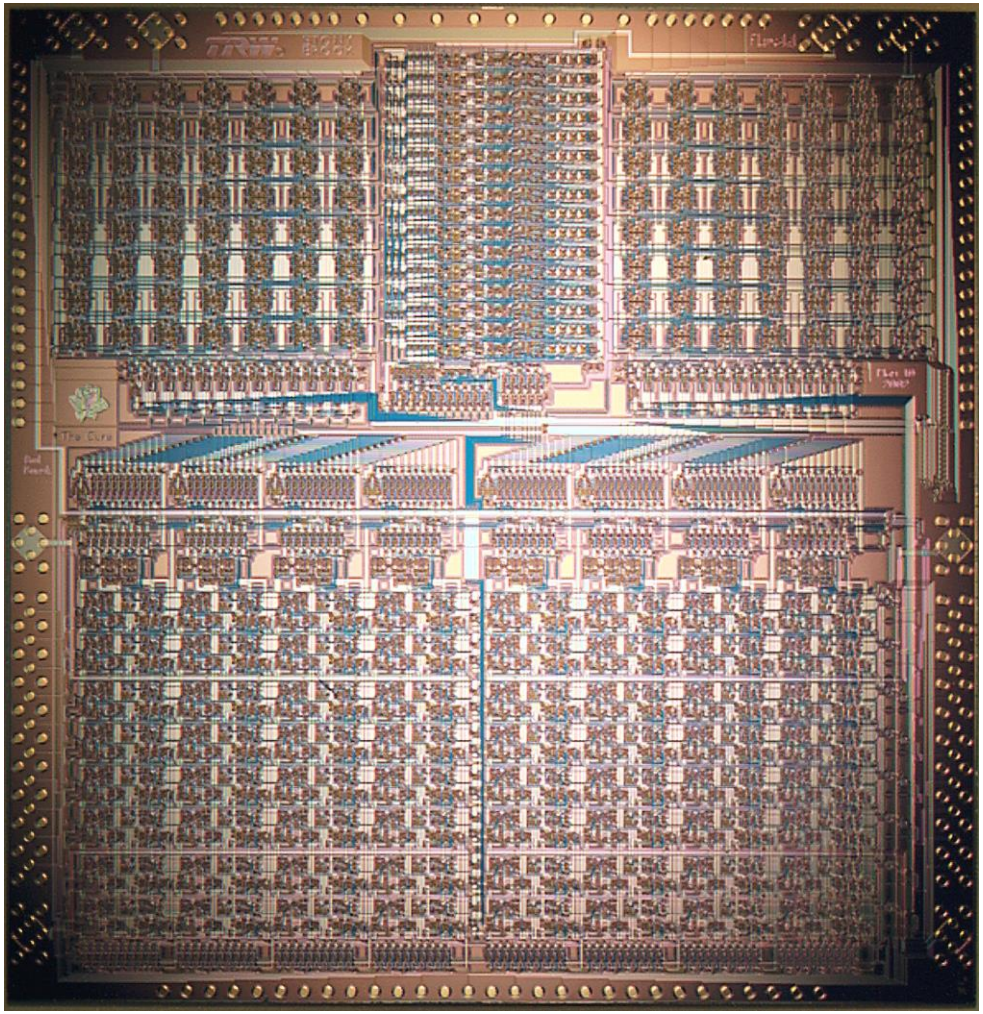
iPhone XS
6 A12 cores
(2 fast + 4 low-power)

(images from apple.com)

What's Next After CMOS?



Superconductor Flux Quantum Computing: 20 GHz 8-bit RSFQ Microprocessor (2002)



- **Designed at Stony Brook University (ECE and Physics dept. teams) in collaboration with TRW (CA)**
- **Technology**: 1.75-μm min. JJ size (TRW)
- **Circuit family**: RSFQ
- **Target Clock**: 20 GHz
- **JJ count**: 63,107
- **Power**: ~9.5 mW at 4.2K
- **Chip size**: 10.35 mm x 10.65 mm

- More about flux quantum computing at **www.quantarctic.com**

What About Bio/Neuro-Inspired Computing?

Human Brain

Laid out flat – 0.25 m^2 (not 3D) 2mm thick with 6 layers
wire diameter 10 nm, neuron diameter 4 microns
100 billion neurons (1 x 10^{11}),
0.15 quadrillion connections (1.5 x 10^{14})
150,000 km of wiring
Wt 1.3 Kg
Frequency 30 hz
Power 25 W ← System architects eat your heart out

Continuous failure – 85,000 die/day, millions misfire/sec

Errors in calculations – make mistakes (examples visual illusions)

Self correction – sensor fusion, redundant calculation

Self healing – reprogram around physical damage

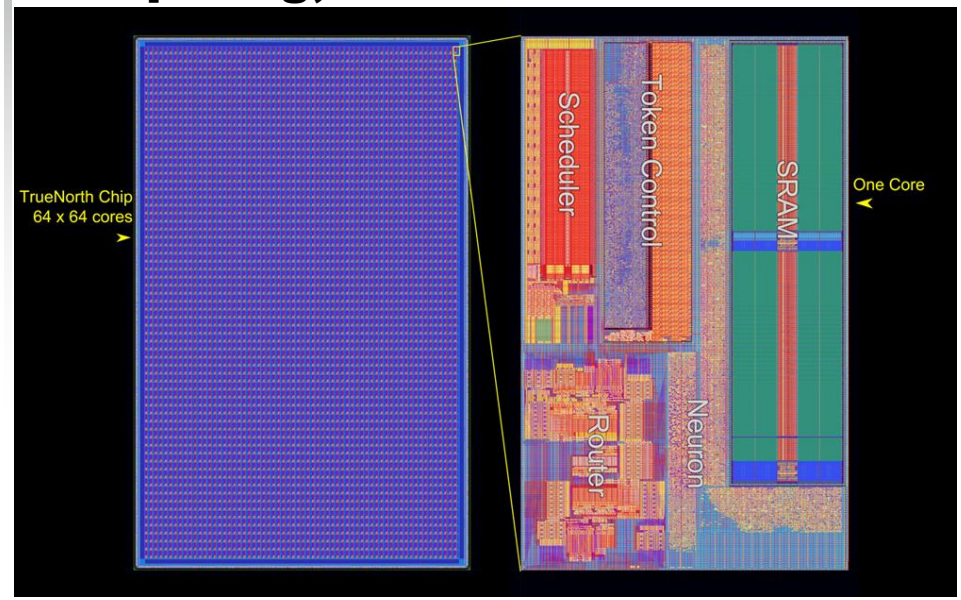
6 Computer Science and Mathematics Division

OAK RIDGE National Laboratory

From A. Geist's presentation at Exaflops workshop (CA), Aug. 3, 2011

TrueNorth Neuromorthic Digital Chip (2014)

Low-power neuromorphic chip designed for applications in mobile sensors, cloud computing, and so on.



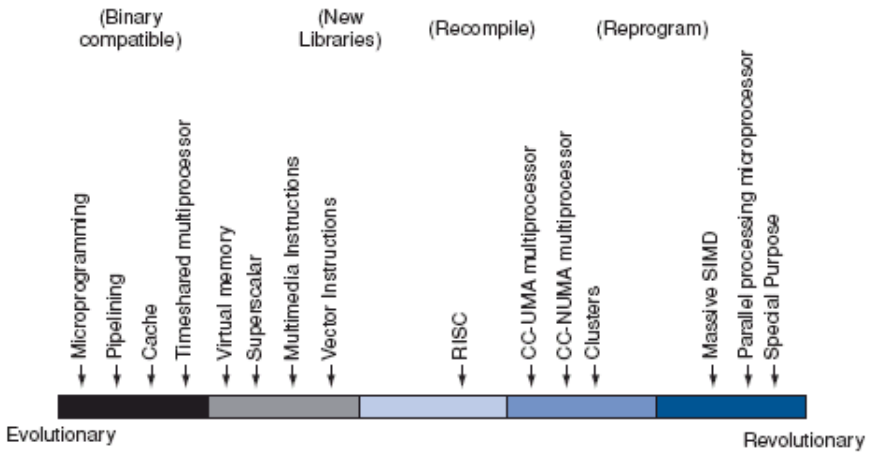
Each core has 256 inputs (axons), 256 outputs (neurons), a big bank of SRAM (which stores the data for each neuron), and a router that allows for any neuron to transmit to any axon up to 255 cores away.

- **IBM, DARPA SyNAPSE**
- **1 million neurons**
- **256 million synapses**
- **4096 neurosynaptic cores**
- **5.4 billion transistors**
- **Asynchronous circuit design**
- **28 nm Samsung Technology:**
- **72 mW/op**

- **Largest current configuration:**
- 16 chips; 16 million neurons; 4 billion synapses
- **Next configuration:** 4,096 chips; 4 billion neurons; 1 trillion synapses
- **Final configuration:** 10 billion neurons; 100 trillion synapses

And in Conclusion …

- Computer Architecture >> ISAs
- Computer architecture is shaped by technology and applications
  - History provides lessons for the future
- Computer Science at the crossroads from sequential to parallel computing
  - Salvation requires innovation in many fields, including computer architecture
- Evolution vs. Revolution
  - "More often the expense of innovation comes from being too disruptive to computer users"



  - "Acceptance of hardware ideas requires acceptance by software people; therefore hardware people should learn about software. And if software people want good machines, they must learn more about hardware to be able to communicate with and thereby influence hardware engineers."

Acknowledgements

- These slides contain material developed and copyright by:
  - Morgan Kauffmann (Elsevier, Inc.)
  - Arvind (MIT)
  - Krste Asanovic (MIT/UCB)
  - Joel Emer (Intel/MIT)
  - James Hoe (CMU)
  - John Kubiatowicz (UCB)
  - David Patterson (UCB)
  - Justin Hsia (UCB)
  - Mikhail Dorojevets (SBU)