信息科学与技术学院
School of Information Science and Technology

CS 110
Computer Architecture
Introduction

Instructors:

Siting Liu & Chundong Wang

Course website: https://toast-lab.sist.shanghaitech.edu.cn/courses/CS110@ShanghaiTech/Spring-2023/index.html

School of Information Science and Technology (SIST)

ShanghaiTech University

2023/2/7

Outline

- What is this course about?
- Teaching team
- Course rules/elements
- Great ideas in computer architecture

What is a computer?

- A **computer** is a machine that can be programmed to carry out sequences of arithmetic or logical operations (computation) automatically. Modern digital electronic computers can perform generic sets of operations known as programs. These programs enable computers to perform a wide range of tasks.

- A **computer system** is a nominally complete computer that includes the hardware, operating system (main software), and peripheral equipment needed and used for full operation. This term may also refer to a group of computers that are linked and function together, such as a computer network or computer cluster.

From https://en.wikipedia.org/wiki/Computer

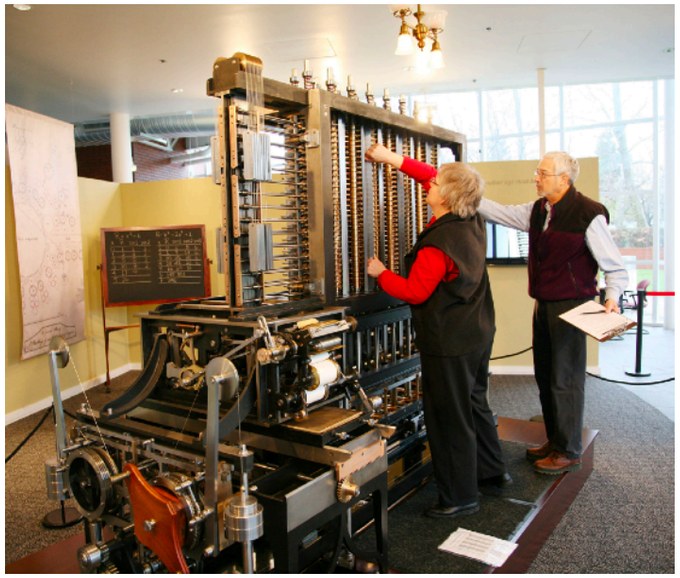
Computer History

- Early computers (calculators)

Chinese abacus

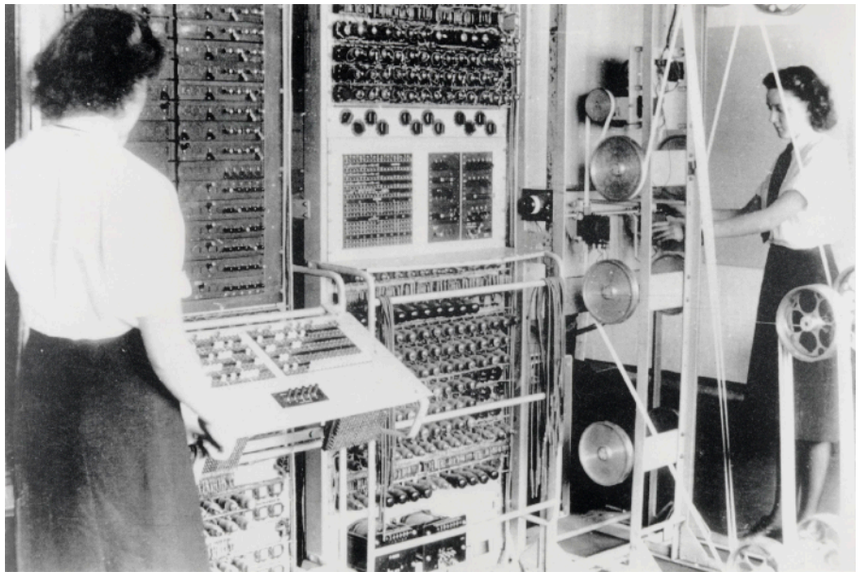
By Loadmaster (David R. Tribble). This image was made by Loadmaster (David R. Tribble). CC BY-SA 3.0.

Automatic mechanical calculator (1820)



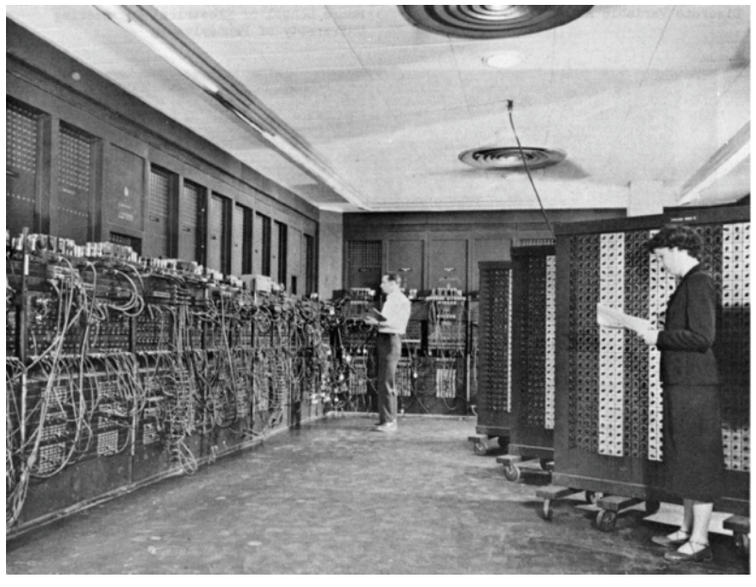
By Jitze Couperus from Los Altos Hills, California, USA. Uploaded by oxyman, CC BY 2.0.

Colossus (WWII, 1943-1945)



This file is from the collections of The National Archives (United Kingdom), catalogued under document record FO850/234.

ENIAC (1945-1946)

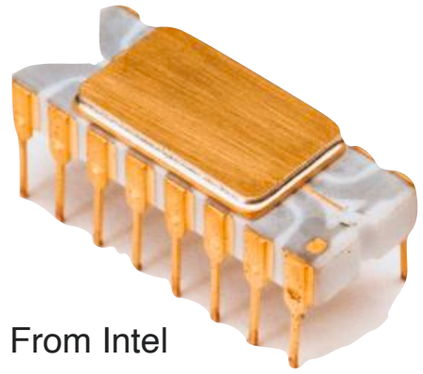


- Programmed by manually setting plugs and switches
- Vacuum tubes

- 5000 additions or subtractions per second
- Modules to multiply, divide, and square root
- 30 tons, 200 kW, 18,000 vacuum tubes

Computer History

- Evolution of theory
  - Alan Turing proposed theoretical basis for the stored-program computer (1936)
  - The *First Draft of a Report on the EDVAC* by Von Neumann (1945)
  - Konrad Zuse suggested and implemented similar ideas (Harvard architecture)
- Evolution of hardware & devices
  - Random-access digital storage (Williams tube, 1947)
  - Metal-oxide-silicon field-effect transistor (MOSFET, 1970-ish)
  - Integrated circuits (ICs) and then VLSI
  - Too many to list ...
- Microprocessor CPU (Intel 4004, 1971)



From Intel

The first delivery of a fully operational 4004 was in March 1971 to Busicom for its 141-PF printing calculator engineering prototype (now displayed in the Computer History Museum in Mountain View, California).

Modern Computer Systems

From Huawei website

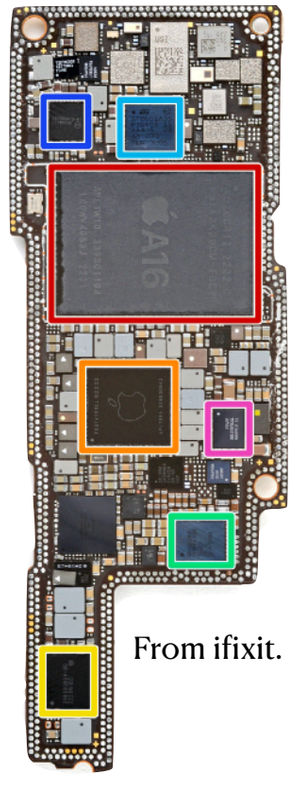
Inside an Intel Core i7 CPU

- 4 cores
- Up to 5.00 GHz
- Max 64 GB memory
- Include Intel Iris Xe Graphics (GPU)
- Windows/Linux (for desktop)

Inside an Apple A16 processor (SoC)

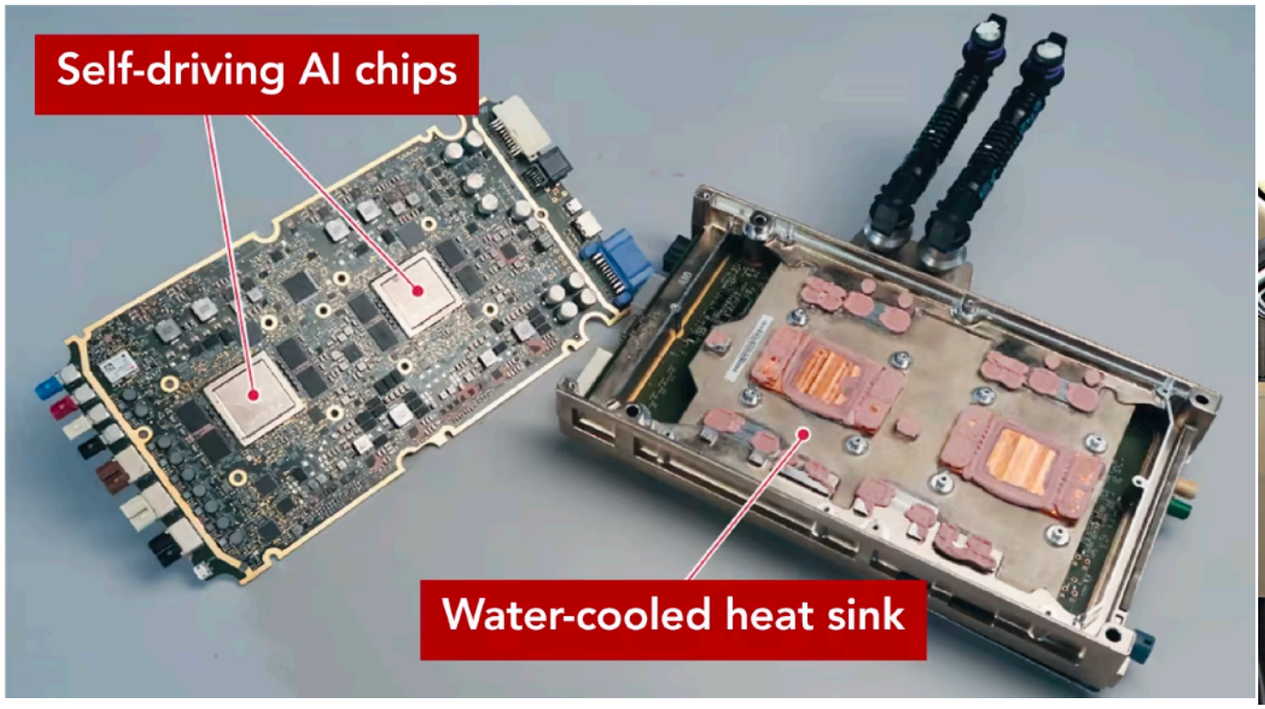
- 6 cores
- Up to 3.46 GHz
- 6 GB memory
- Include Apple-designed GPU, image processor, 16-core neural engine, etc.
- 17 TOPS on neural engine
- iOS

From Wikipedia



From ifixit.

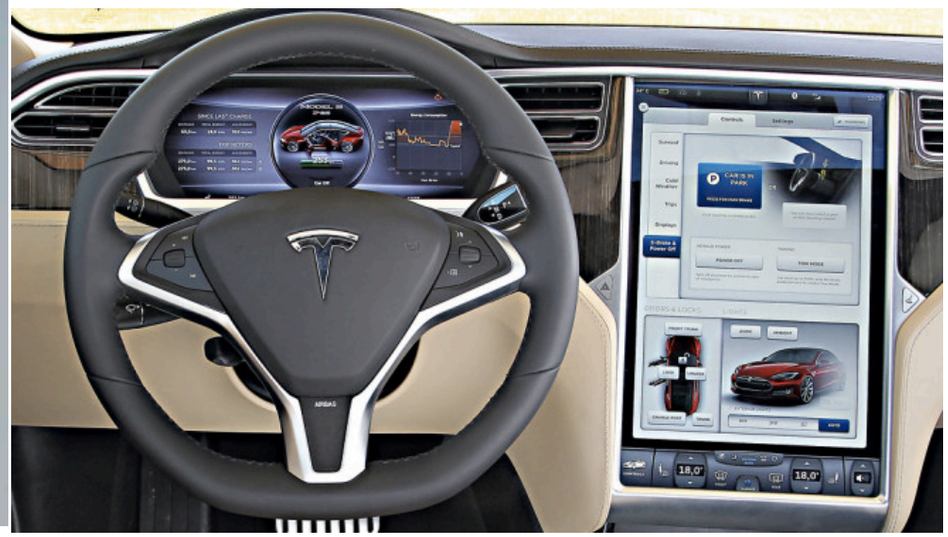
Modern Computer Systems



FSD computer, inside a FSD (full self-driving) chip (SoC)

- 12 CPU cores, 2.2 GHz
- Up to 8 GB memory
- Include a Mali GPU @ 1 GHz, 2 neural processing units @ 2 GHz, etc.
- GPU 600 GFLOPS, NPU 36.86 TOPS
- Specialized software

From Wikichip.



Main screen (infotainment) computer

- Ubuntu-based OS
- AMD Ryzen & GPU/Intel Atom

From Nikkei-xTech, Ingineerix & Dongchedi.
https://www.youtube.com/watch?v=ODdIRr5RzI8
https://www.dongchedi.com/article/7122294255204712972

Modern Computer Systems

SIST computing center

- Hundreds/thousands of servers
- Each has one or multiple CPUs
- Mostly runs Linux OS


cooling towers
warehouse-scale computer
power substation

Hardware Revolution (Cloud Services)

Providers	CPU	GPU	FPGA*	ASIC (DSA)
Alibaba	X86/ARM/RISC-V	Nvidia/AMD	Intel/AMD	AliNPU
AWS (Amazon)	X86/Graviton (ARM)	Nvidia/AMD	AMD	Trainium
Azure (MS)	X86	Nvidia	Intel	N/A
Baidu	X86	Nvidia	AMD	Kunlun
Google	X86	Nvidia	N/A	TPU
Huawei	X86/Kunpeng (ARM)	Nvidia & Ascend	AMD	Ascend
Tencent	X86	Nvidia & Xinghai	AMD	Enflame (燧原)

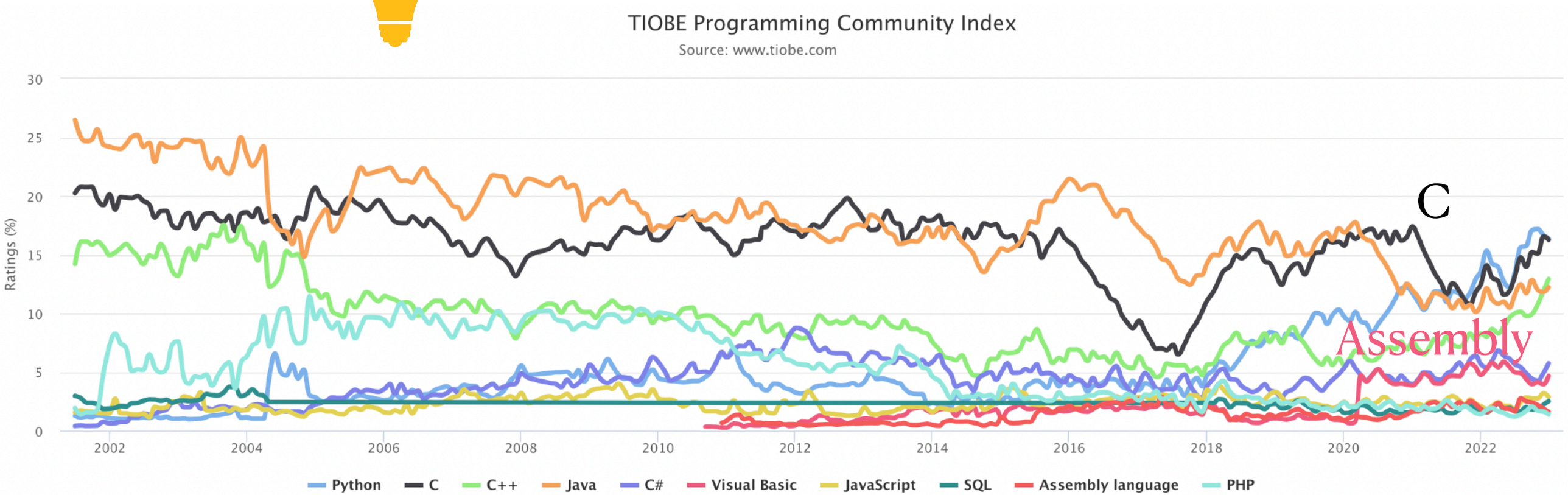
*Intel: formerly a.k.a. Altera
AMD: formerly a.k.a. Xilinx

Computer Evolution

- Stronger (process complex information, e.g. AI)
  - High performance, etc.
- Easier to use (touch screen, voice control, etc.)
- Diversity (edge/cloud)
- But, all developed based on computer architecture theory

Computer Programming (Software)

- Write program (as CS students/programmers)
- It works!



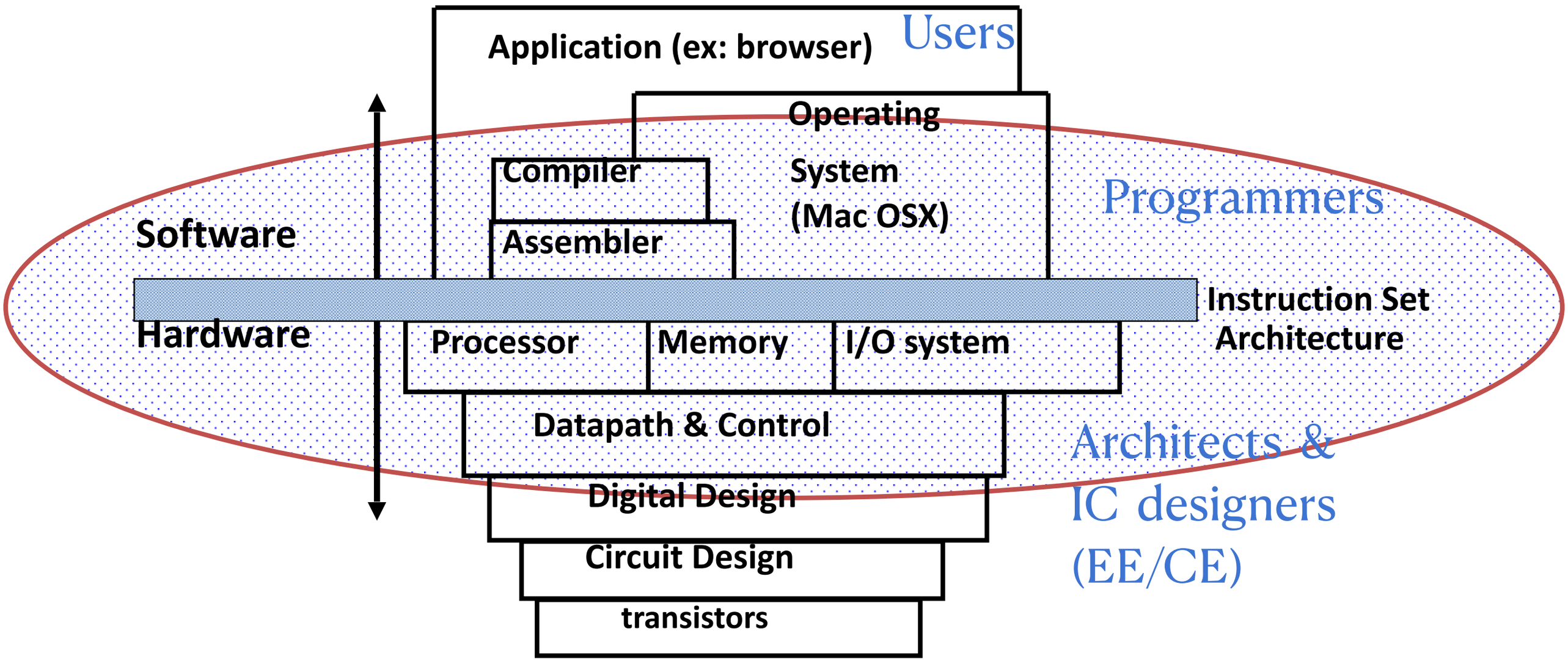
- How these programs run on the hardware?

CS 110 is NOT really about C Programming

- It is about the *hardware-software* interface
  - What does the programmer need to know to achieve the highest possible performance
- C is close to the underlying hardware, unlike languages like Python, Java!
  - Allows us to talk about key hardware features in higher level terms
  - Allows programmer to explicitly harness underlying hardware parallelism for *higher performance* and *power efficiency*
- We will also learn *assembly*
  - Allows us to talk about the hardware in lower level terms
  - Uses RISC-V assembly/ISA as an example
- We will focus on *CPU-based computer systems* for simplicity
  - xPUs work similarly (GPU, TPU, etc.)

Computer Architecture (a.k.a. Machine Structure)

- Abstraction of computers



Course Contents

- How computer (CPU & main memory) program works?
  - Info representation/binary
  - Intro to C
  - Assembly (RISC-V ISA)
  - Compiler, assembler, linker & loader (in general)
- CPU hardware design & How CPU works (RISC-V ISA)
  - Logic & synchronous digital systems
  - Functional Units, FSM & Datapath
  - Pipeline, superscalar (Parallelism 1)
- Memory systems (Cache, Main memory)
- Parallelism 2 (SIMD, TLP, Sync & OpenMP)
- Topic studies (OS, Interrupts, Virtual Memory, FPGA, Warehouse-Scale Computing, DMA, external memory, fault-tolerance, security, etc.)

Learning Goals

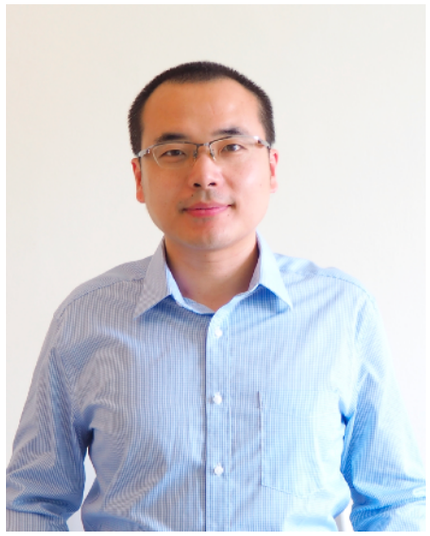
- Be an advanced programmer
- Know how computer works in DEPTH (CPU, memory, I/O, etc.)
- Be prepared for further CS/EE course studies

Outline

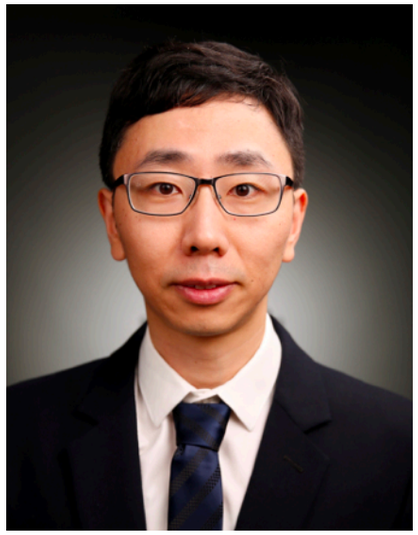
- What is this course about?
- Teaching team
- Course rules/info/elements
- Great ideas in computer architecture

Meet the Teaching Team!

- Greetings!

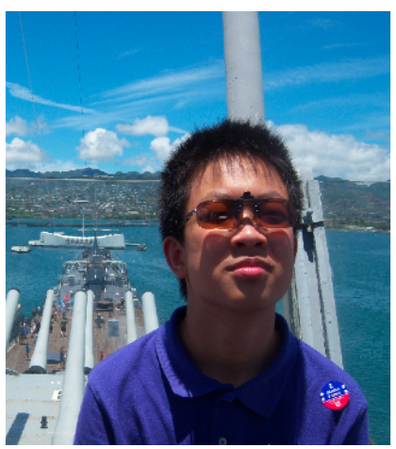


[Chundong Wang](#)
<wangchd>



[Siting Liu](#)
<liust>

- Head TA



Yanpeng Hu
<huyp>

- TAs

Suting Chen
<chenst>

Meng Chen
<chenmeng>

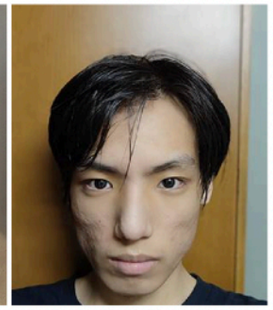
Zongze Li
<lizz>

Haiyue Chen
<chenhy5>

Haoran Jiang
<jianghr1>

Weiming Hu
<huwm1>

Yulu Song
<songyl1>



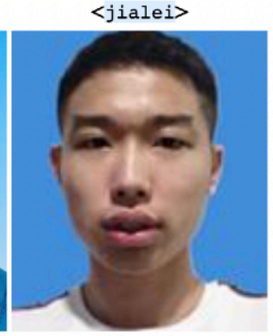
Lei Jia
<jialei>

Cheng Peng
<pengcheng2>

Jintian Hu
<hujt>

Qing Xu
<xuqing2>

Linjie Ma
<malj>



Han Li
<lihan>

Huizhe Su
<suhzh>

Outline

- What is this course about?
- Teaching team
- Course rules/info/elements
- Great ideas in computer architecture

Course Info

CS110

Lecture	Tuesday	8:15-9:55	Teaching Center 301	Siting & Chundong
Lecture	Thursday	8:15-9:55	Teaching Center 301	

Lab 1	Thu.	19:50-21:20	SIST 1B-108	TA: Yulu Song
Lab 2	Tue.	19:50-21:20	SIST 1B-108	TA: Huizhe Su
Lab 3	Tue.	19:50-21:20	SIST 1B-110	TA: Linjie Ma
Lab 4	Mon.	19:50-21:20	SIST 1B-110	TA: Lei Jia
Lab 5	Mon.	19:50-21:20	SIST 1B-108	TA: Haoran Jiang
Lab 6	Thu.	19:50-21:20	SIST 1B-106	TA: Zongze Li
Lab 7	Tue.	19:50-21:20	SIST 1B-106	TA: Suting Chen
Lab 8	Mon.	19:50-21:20	SIST 1B-106	TA: Cheng Peng
Lab 9	Thu.	19:50-21:20	SIST 1B-110	TA: Haiyue Chen

*Lab starts next week; Discussion starts next week.

Computer Architecture

- The most important course this semester

  6 credit points in total

  4 credit lecture (CS110); 2 credit projects (CS110P)

  - Your first CS only course
  - Spend a lot of time on:

    Textbook reading before class;

    HW and projects;

    Lab preparation;

    Mid-terms and final;

Course Info

- Course webpage: https://toast-lab.sist.shanghaitech.edu.cn/courses/CS110@ShanghaiTech/Spring-2023/index.html
- **Acknowledgement**: **UC Berkeley's CS61C: https://cs61c.org/**; 一生一芯: https://ysyx.oscc.cc/
- Instructors:
  - Chundong Wang & Siting Liu
- Teaching Assistants: (see course webpage and piazza for office hour)
- https://piazza.com/shanghaitech.edu.cn/spring2023/cs110
- Textbooks: Average 15 pages of reading/week
  - Patterson & Hennessey, Computer Organization and Design RISC-V edition!
  - Kernighan & Ritchie, The C Programming Language, 2nd Edition
  - RTFM: C & RISC-V
- Piazza:
  - Every announcement, discussion, clarification happens there
- Materials this year are similar to previous years, but there might be differences!
- https://robotics.shanghaitech.edu.cn/courses/ca/22s/

CS110 vs CS110P

- Project course CS110P and CA I course are graded separately =>
  - Different grades in both are likely
  - Its possible to fail one but not the other

- The scheduled time for CS110P is used for the LAB which is part of the course!
- CS110P has no assigned time – it is for the projects.
- Announcements for the projects are held in CS110 lectures AND posted on piazza.
- Discussions often cover topics from the lectures AND the project (tools, linux, etc.)
- You can get help the projects in piazza and from the TAs during your lab and during the office hours.

CS110 Grading

- Homework: 40%
  - 4 programming HW: 6% each
  - 3 paper HW: 5% each
  - HW1: 1%
- Lab: 5%
- Exams: 53%
  - Midterm I: 14%
  - Midterm II: 14%
  - Final: 25%
- Participation: 2%
  - Based on participation (reading and writing posts) in piazza

Project CS110P Grading

- Projects: 100%
    - P1.1: 17%
    - P1.2: 17%
    - P2.1: 16%
    - P2.2: 16%
    - P3: 17%
    - P4: 17%

- Discuss & ask questions after each class
  - General topics can be discussed outside of class thread
- Ask general questions about HW/projects
- Announcements will be posted on piazza only!
  - Check email & piazza at least once a day!
- Participation is recorded & goes into grade!

Labs

- Labs: Find one partner for your lab-work and project from you lab class!
  - Labs start next week
  - Projects are done in the same 2-person teams!
- Need Linux for the labs!
  - Recommendation: install natively (dual boot)! (STFW)
  - Virtual Machine works fine
  - If in doubt: Ubuntu 20.04
  - Many labs/ homework/ project may work with Mac, but no guarantee, no support. Use Linux!

Discussion

- Time:
  - Monday 20:30-21:30 Teaching center 301
  - Friday 20:30-21:30 Teaching center 301
- Content: The TA will give a presentation about current topics. Then you can ask questions.
- Participation is encouraged.

Office Hours

- Meet a TA in person and ask questions/get (high-level) help with HW or projects
- Before going to office hour:
  - Is you question already answered on piazza?
  - If the question is suitable (i.e. not giving away a HW answer), prefer to ask on piazza
- Office hours of TAs are posted in piazza
- Office hours with the Profs. are also possible – best write an email first to make an appointment.

- CA will use Autolab for grading HW & projects
  - Setup maintained by TAs
    - => Please be patient and report any bugs/problems in piazza or (if sensitive) via email.
    - Your logins will be created soon!
    - http://autolab.sist.shanghaitech.edu.cn
- HW 1 will be available on Autolab soon.
- Gradescope/Blackboard (more likely) for text-based HW and exams.

- Gitlab for projects!
- Use for collaboration.
- Autolab will directly pull from your gitlab!
- http://autolab.sist.shanghaitech.edu.cn/gitlab
- Accounts will be created soon

Late Policy … Slip-Days!

- Assignments due at 11:59:59 PM
- You have 3 slip-day tokens in total (NOT hour or min)
- Every day your project or homework is late (even by a minute) we deduct a token
- After you've used up all tokens, it's 25% deducted per day.
  - No credit if more than 3 days late
  - Save your tokens for projects, worth more!!
- No need for sob stories, just use a slip day!
- Autolab will take care of this!

Policy on Assignments and Independent Work

- ALL PROJECTS WILL BE DONE WITH A PARTNER
- With the exception of laboratories and assignments that explicitly permit you to work in groups, all homework are to be YOUR work and your work ALONE.
- PARTNER TEAMS MAY NOT WORK WITH OTHER PARTNER TEAMS
- You can discuss your assignments with other students, and credit will be assigned to students who help others by answering questions on Piazza (participation), but we expect that what you hand in is yours.
- Level of detail allowed to discuss with other students: Concepts (Material taught in the class/ in the text book)! Pseudocode is NOT allowed!
- Use the Office Hours of the TA and the Profs. if you need help with your homework/ project!
- Rather submit an incomplete homework with maybe 0 points than risking an F!
- It is NOT acceptable to copy solutions from other students.
- You can never look at homework/ project code not by you/ your team!
- You cannot give your code to anybody else –> secure your computer when not around it
- It is NOT acceptable to copy (or start your) solutions from the Web.
- It is NOT acceptable to use PUBLIC github archives (giving your answers away)
- It is NOT acceptable to give anyone other than your project partner access to your gitlab!
- We have tools and methods, developed over many years, for detecting this. You WILL be caught, and the penalties WILL be severe.
- At the **minimum F** in the course, and **a letter to your university record** documenting the incidence of cheating.
- Both Giver and Receiver are equally culpable and suffer equal penalties

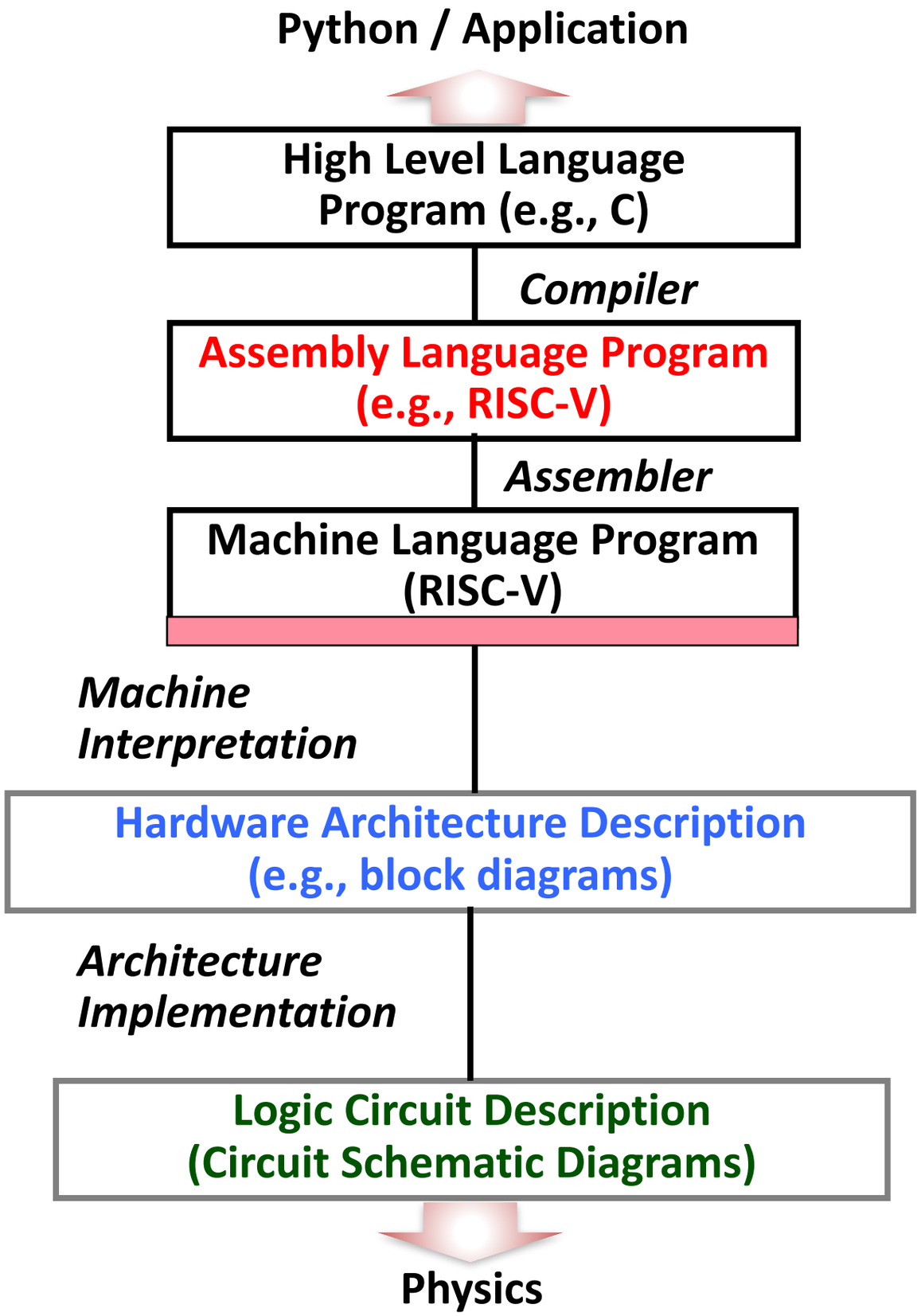
Outline

- What is this course about?
- Teaching team
- Course rules/info/elements
- Great ideas in computer architecture

Great Ideas in Computer Architecture

- Abstraction
  (Layers of Representation/Interpretation)
- Moore's Law (Designing through trends)
- Principle of Locality (Memory Hierarchy)
- Parallelism
- Performance Measurement & Improvement
- Dependability via Redundancy

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

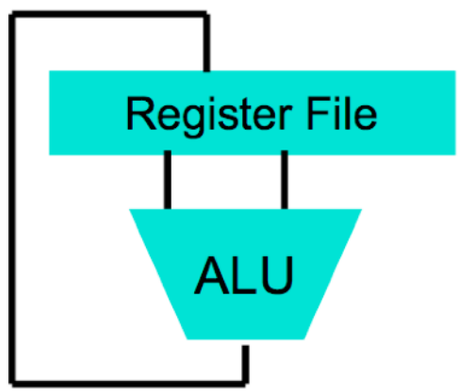


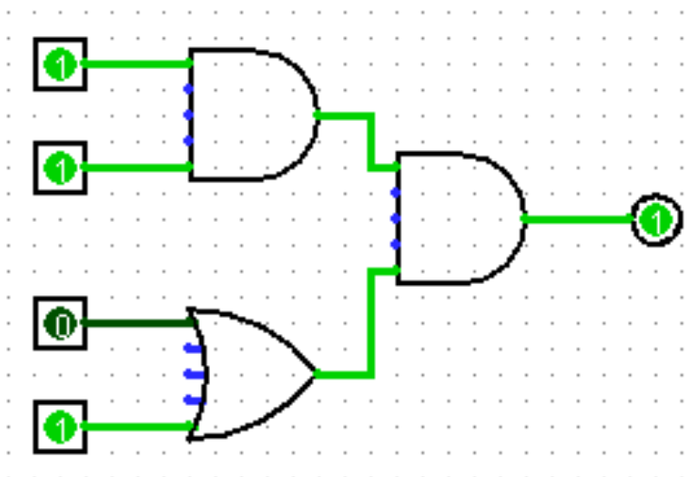

```
temp = v[k];
v[k] = v[k+1];
v[k+1] = temp;
```

```
lw    t0, 0(s2)
lw    t1, 4(s2)
sw    t1, 0(s2)
sw    t0, 4(s2)
```

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

```
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
```





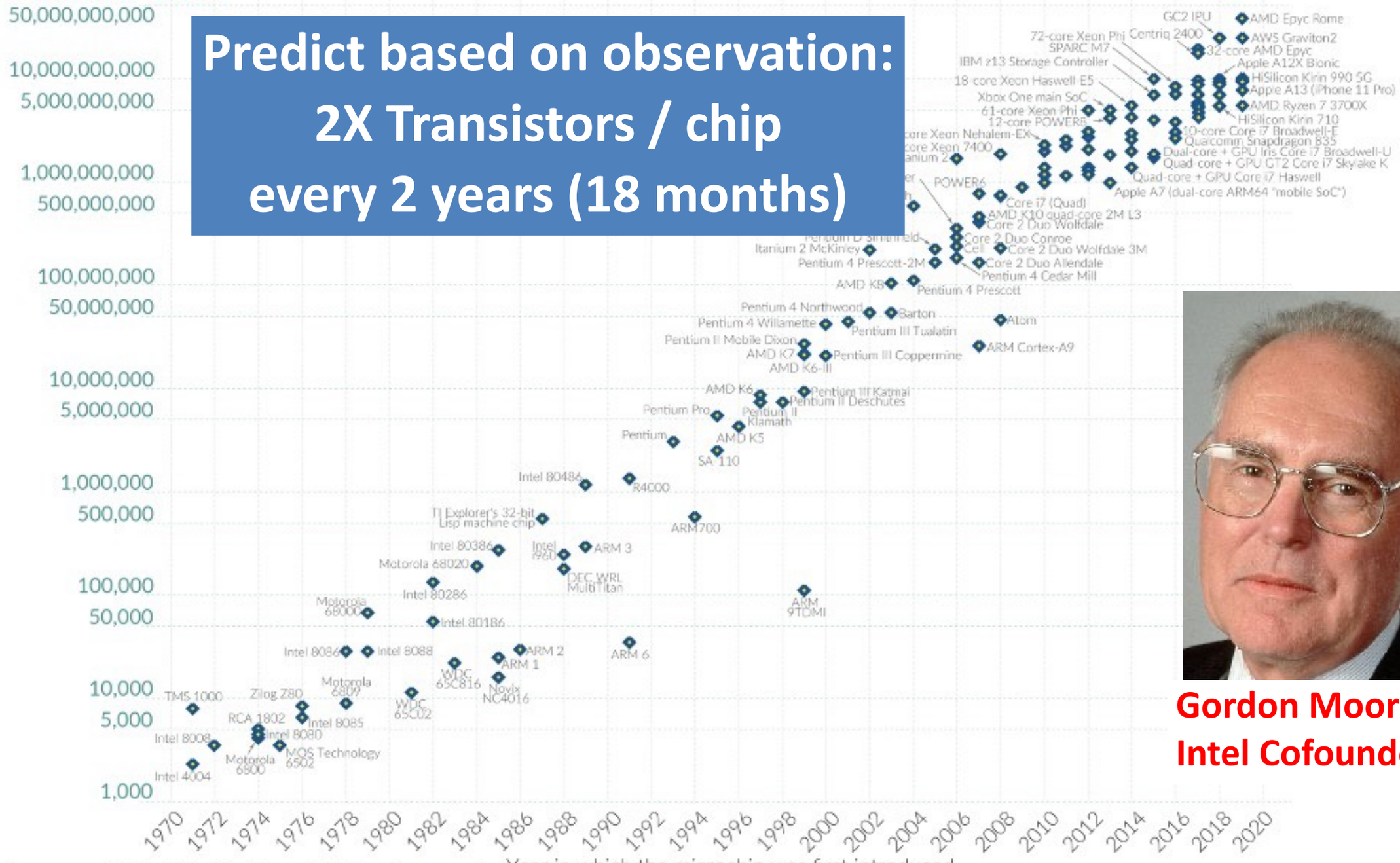
2. Moore's Law?

Moore's Law: The number of transistors on microchips doubles every two years

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important for other aspects of technological progress in computing – such as processing speed or the price of computers.

Our World in Data

Predict based on observation:
2X Transistors / chip
every 2 years (18 months)



Gordon Moore
Intel Cofounder

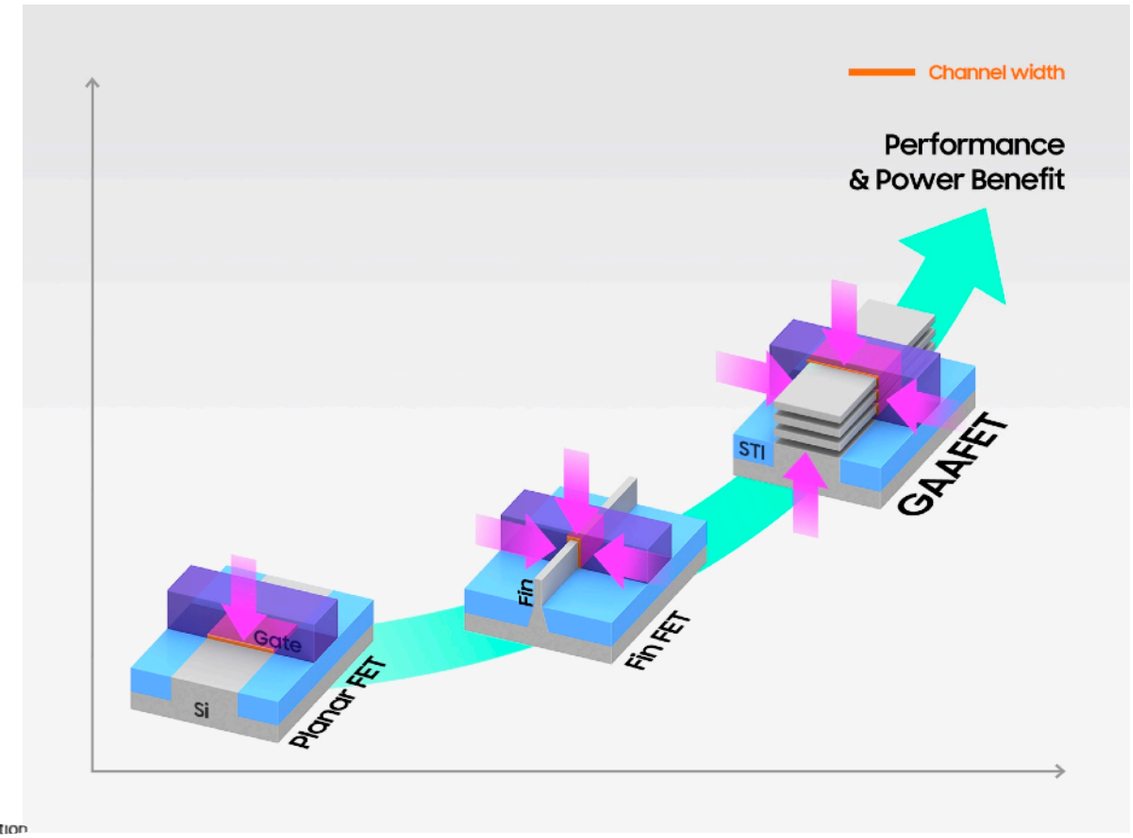
Data source: Wikipedia (wikipedia.org/wiki/Transistor_count)
OurWorldinData.org – Research and data to make progress against the world's largest problems.

Interesting Times

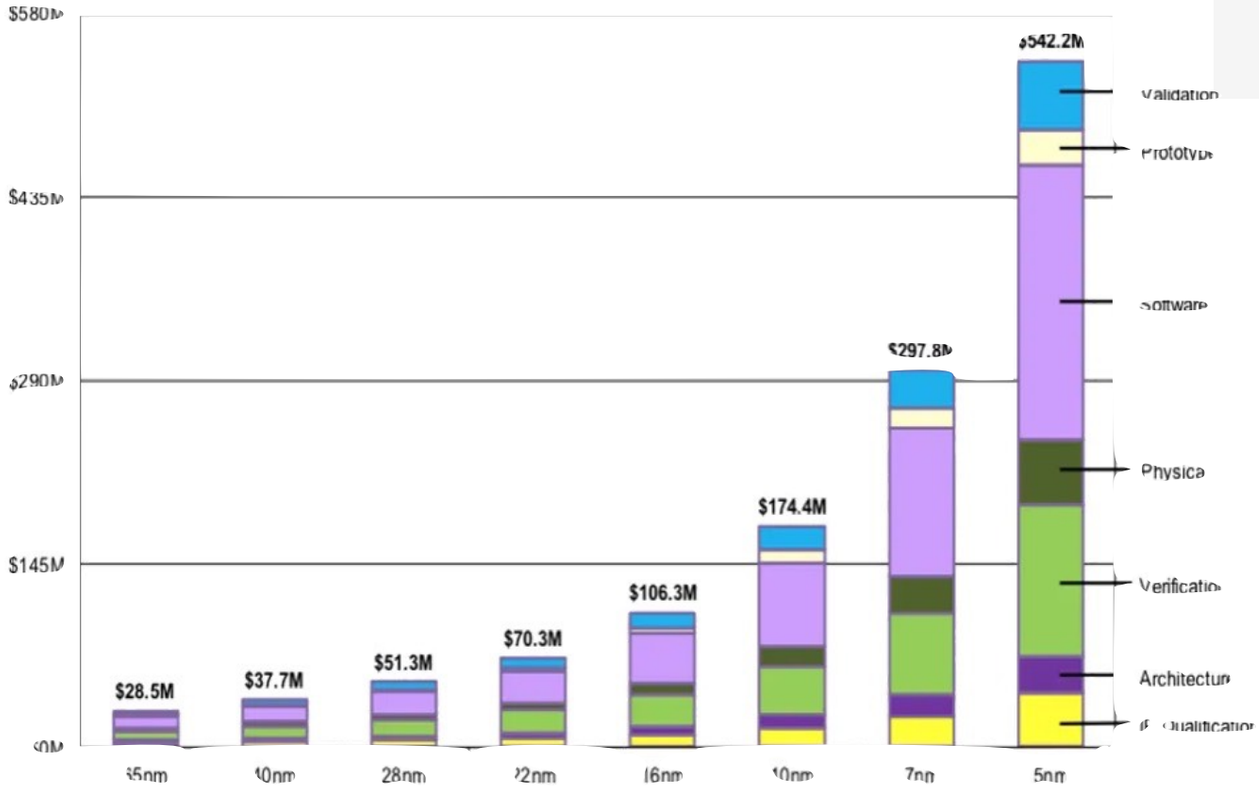
- Moore's Law relied on the cost of transistors scaling down as technology scaled to smaller and smaller feature sizes.
- BUT newest, smallest fabrication processes <5nm, might have greater cost/transistor !!!!
- Power budget (TDP)
- So, why shrink????

Advanced Technology Nodes

- Now: 4 nm process (A16 Bionic, TSMC)
- Future: up to 3 nm (rumors on A17)
- Other notable players: Samsung, Intel, GlobalFoundries (AMD), IBM
- China: SMIC: 14-nm production

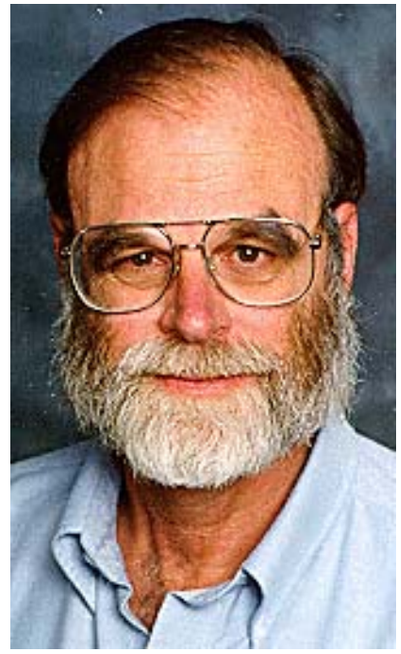


From Samsung Semiconductor



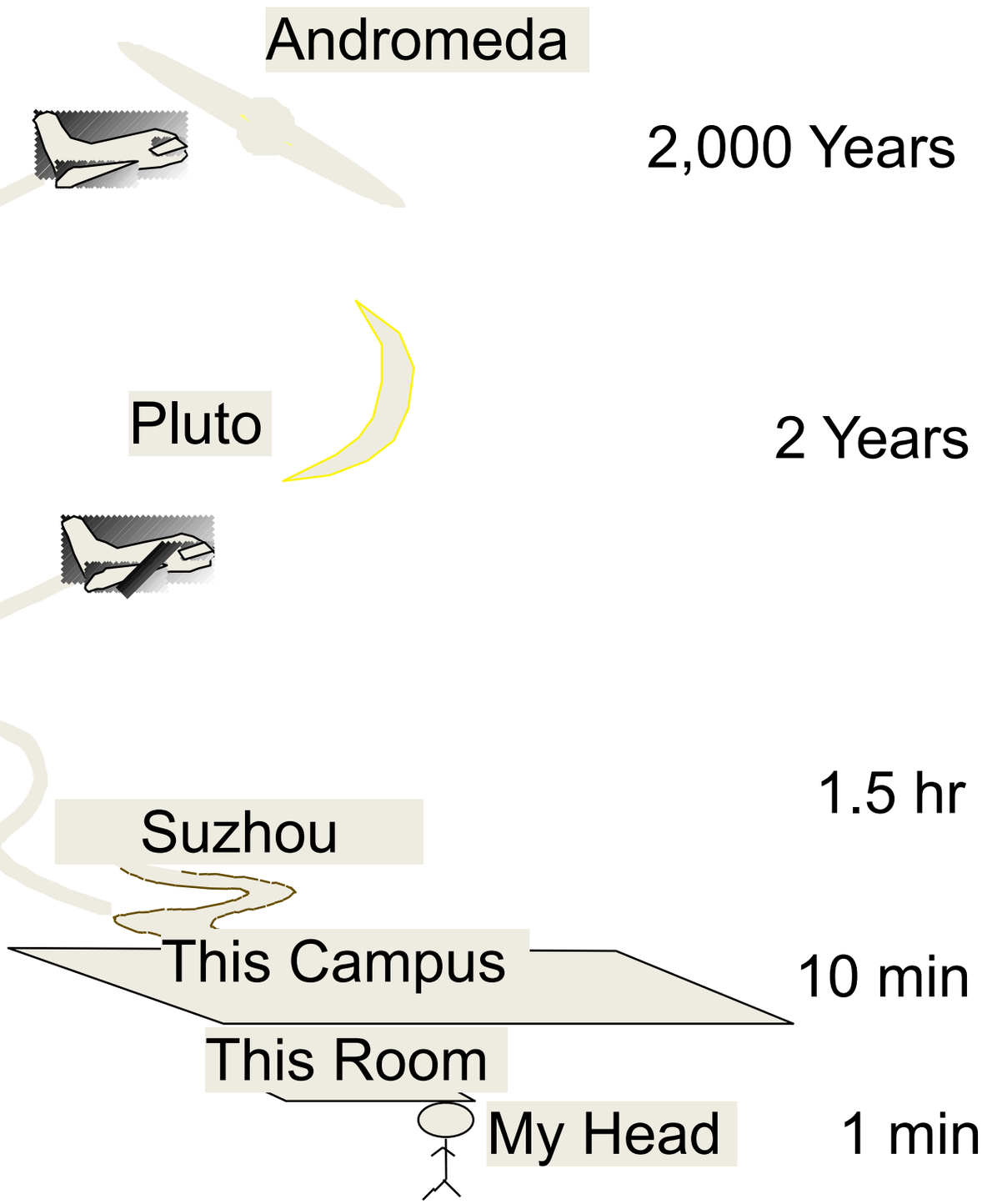
Design cost at different technology nodes.
Data source: Handel Jones, IBS, 2018.

Jim Gray's Storage Latency Analogy: How Far Away is the Data?

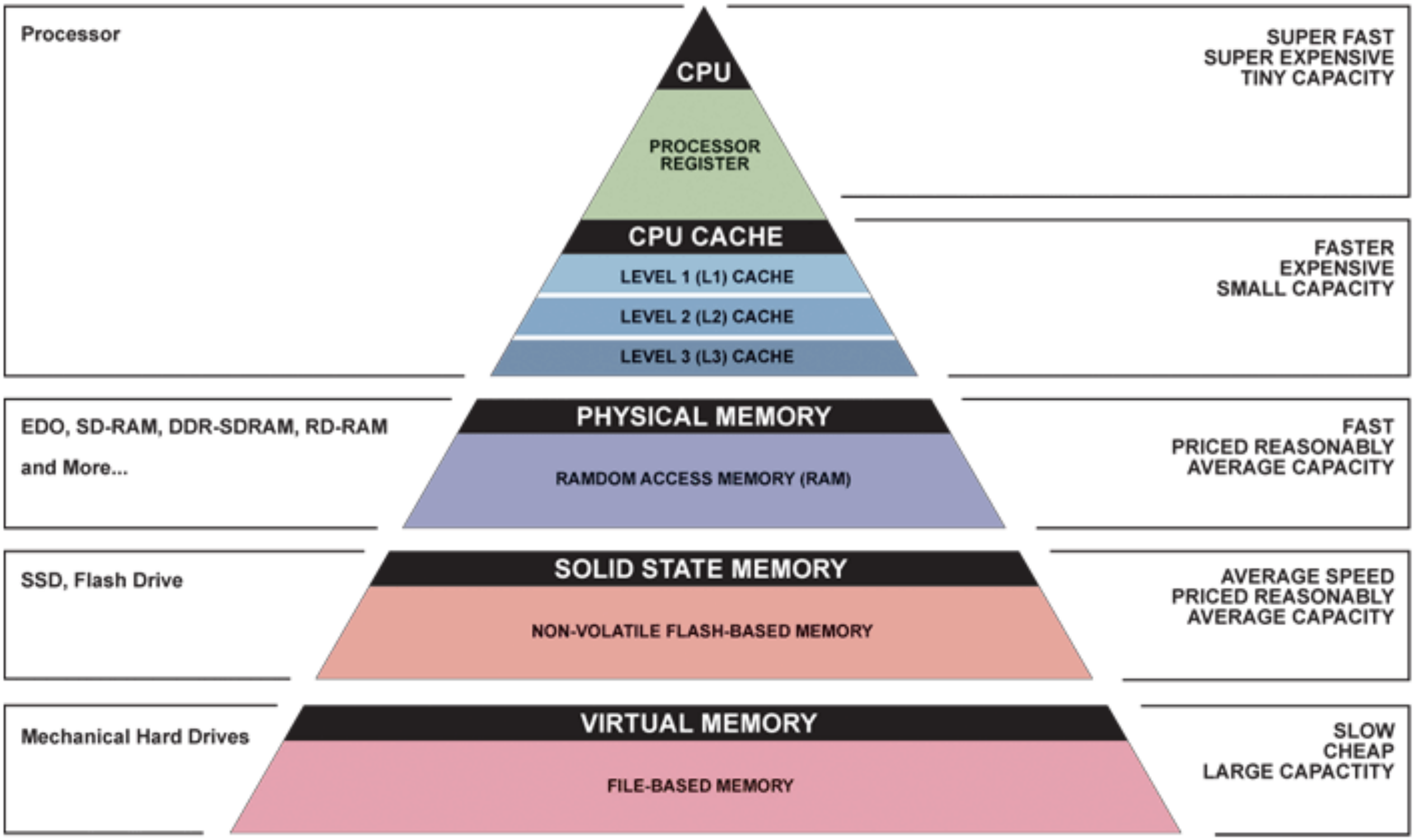


Jim Gray
Turing Award

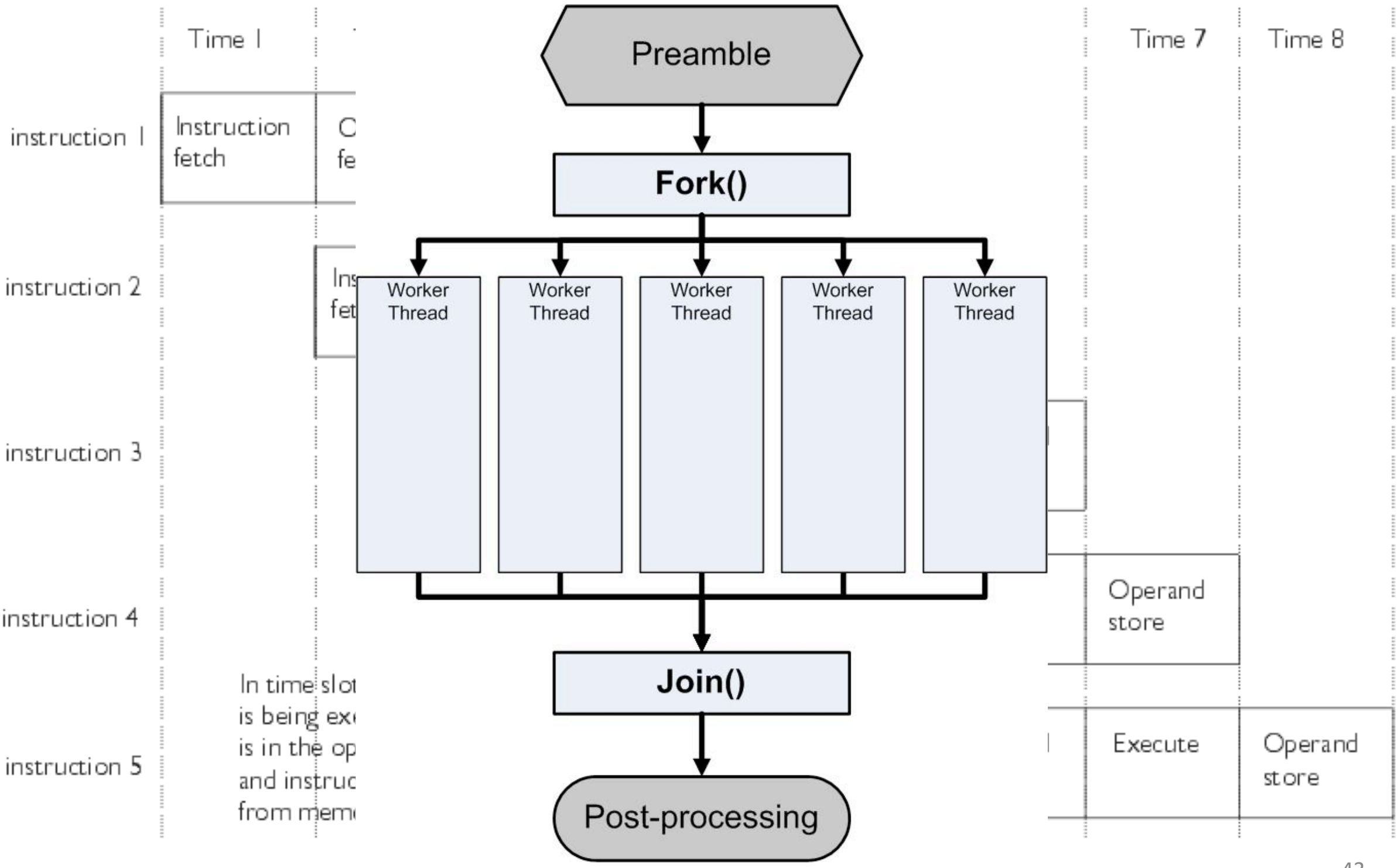
10^9	Tape /Optical Robot	2,000 Years
10^6	HardDisk	2 Years
100	Main Memory	1.5 hr
10	On Board Cache	10 min
2	On Chip Cache	
1	Registers	1 min
(ns)		



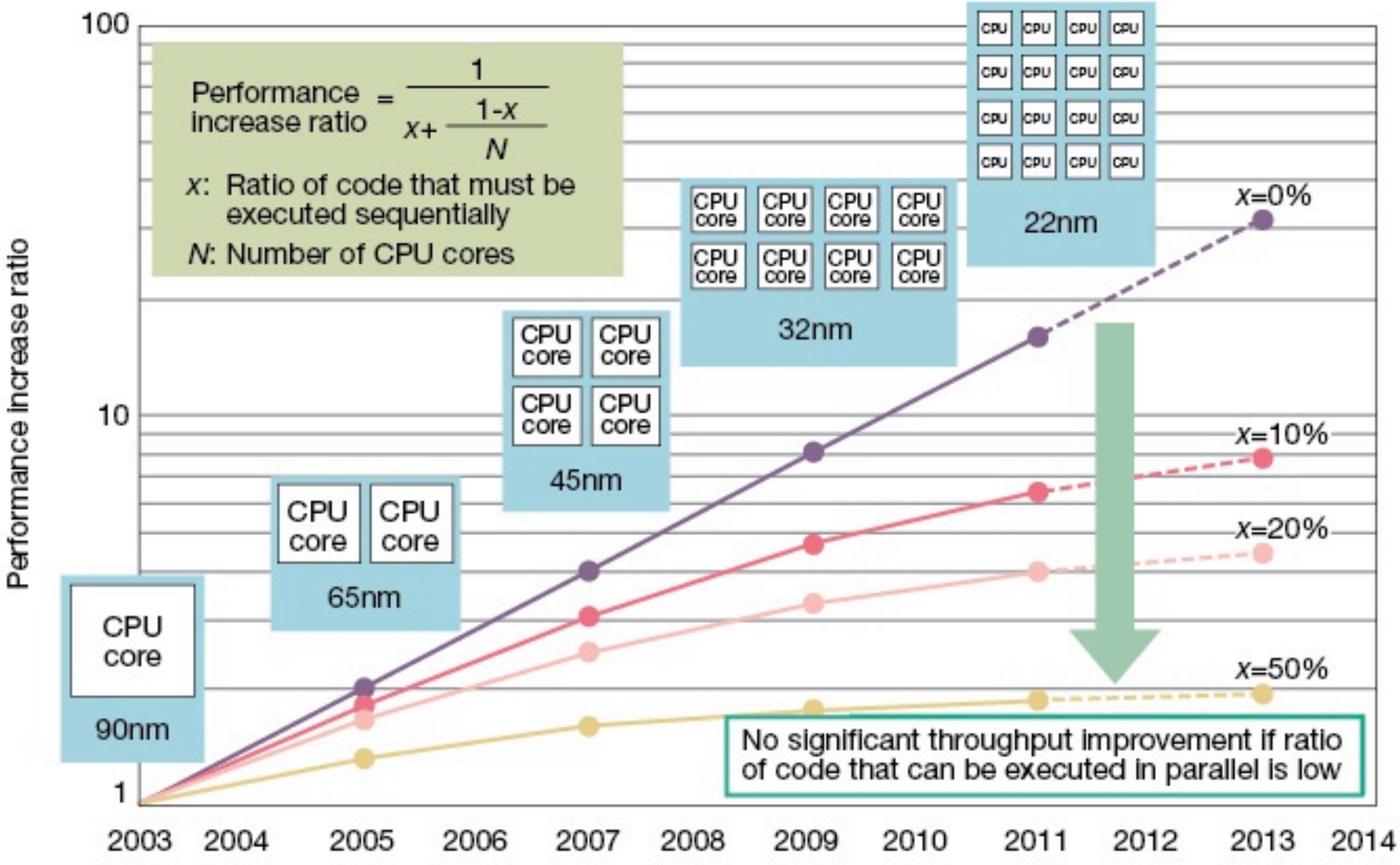
Great Idea 3: Principle of Locality/ Memory Hierarchy



Great Idea 4: Parallelism



Caveat: Amdahl's Law



Gene Amdahl
Computer Pioneer

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.

Great Idea 5: Performance Measurement & Improvement

- Tuning application to underlying hardware to exploit:
  - Locality; parallelism; special hardware features, like specialized instructions (e.g., matrix manipulation)
- Latency
  - How long to set the problem up. It is all about time to finish.
- Power/energy consumption
  - How much energy consumed to perform certain tasks.
- Benchmark

Coping with Failures

- 4 disks/server, 50,000 servers
  - Assume 4% annual failure rate
- On average, how often does a disk fail?
  - 1 / month
  - 1 / week
  - 1 / day
  - 1 / hour

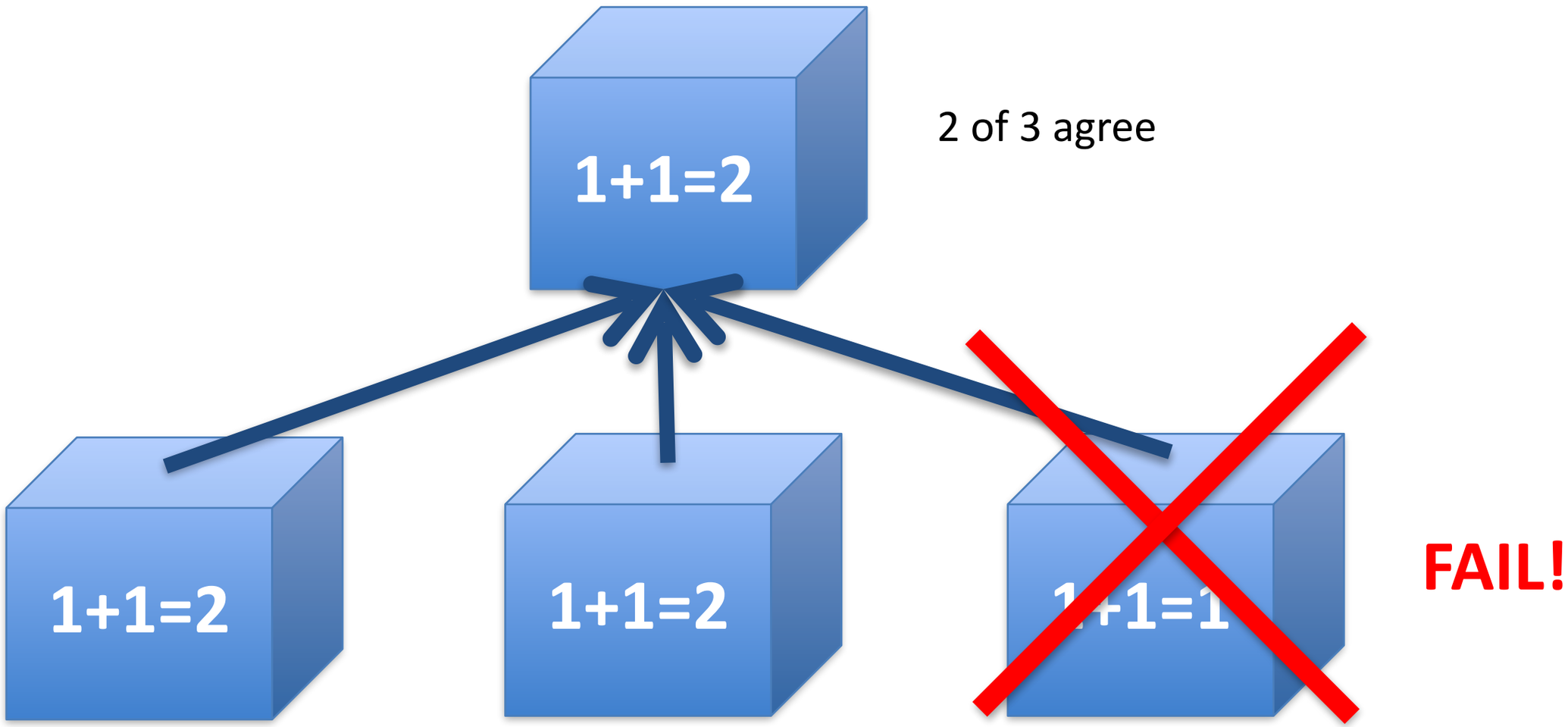
50,000 x 4 = 200,000 disks

200,000 x 4% = 8000 disks fail

365 days x 24 hours = 8760 hours

Great Idea 6: Dependability via Redundancy

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



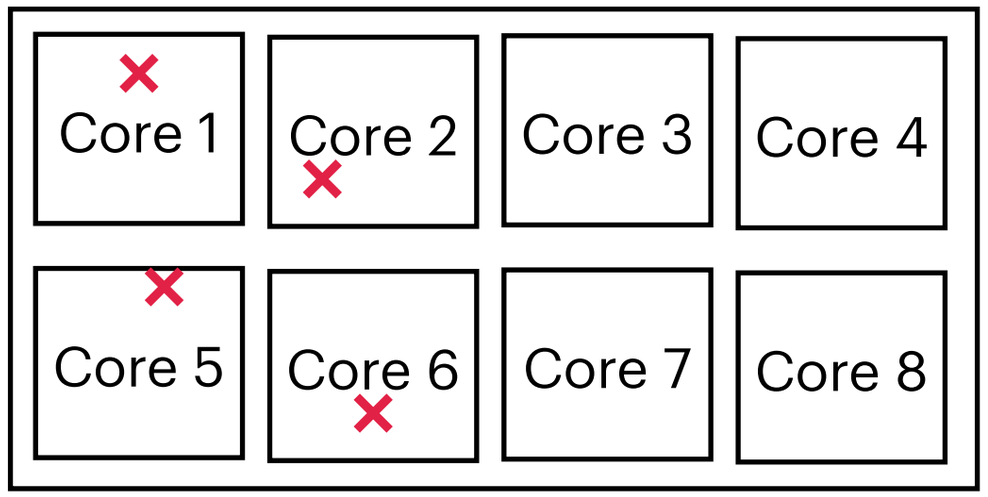
Increasing transistor density reduces the cost of redundancy

Great Idea 6: Dependability via Redundancy

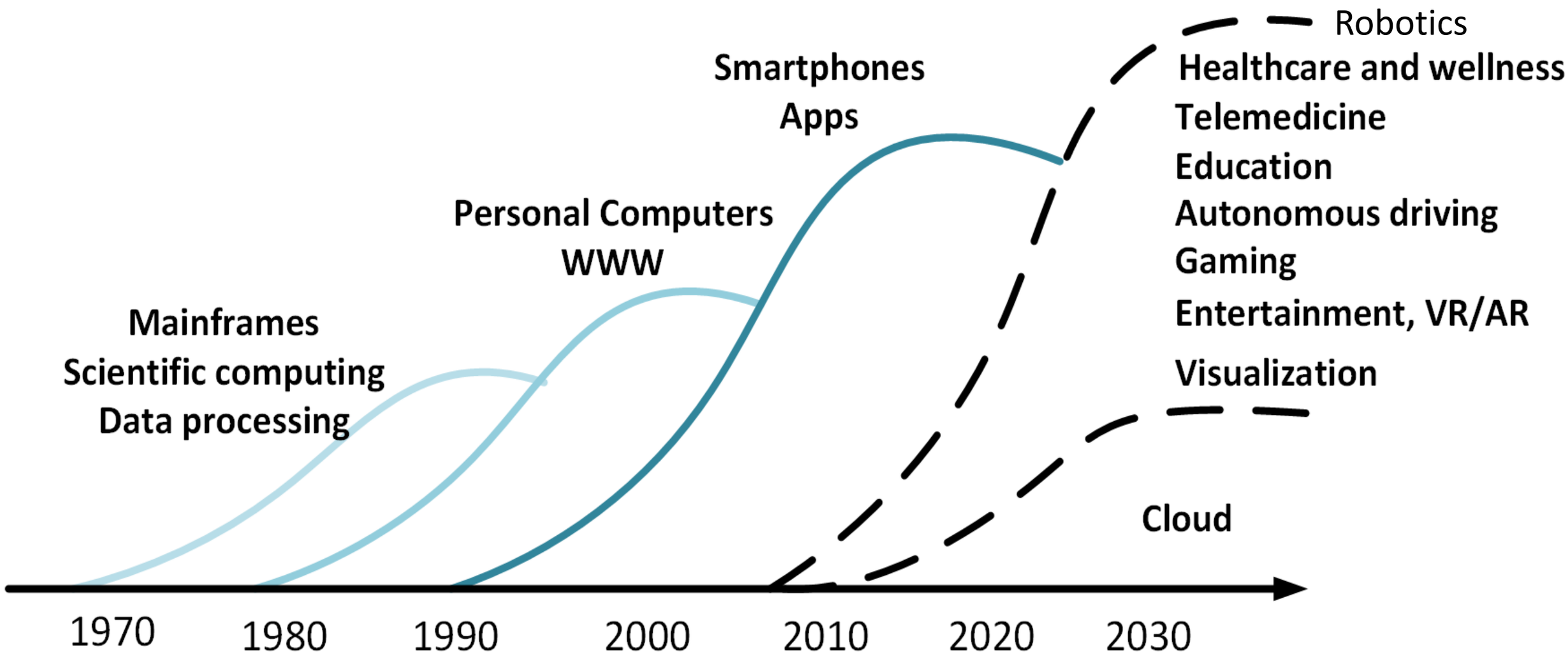
- Applies to everything from datacenter to storage to memory to instructors
  - Redundant datacenters so that can lose 1 datacenter but Internet service stays online
  - Redundant disks so that can lose 1 disk but not lose data (Redundant Arrays of Independent Disks/RAID)
  - Redundant memory bits so that can lose some bits but not the data (Error Correcting Code/ECC Memory)

Great Idea 6: Dependability via Redundancy

- Redundancy improves yield
  - Considering a CPU with 10^9 transistors
  - Each transistor has a probability 10^{-9} to be failure during production

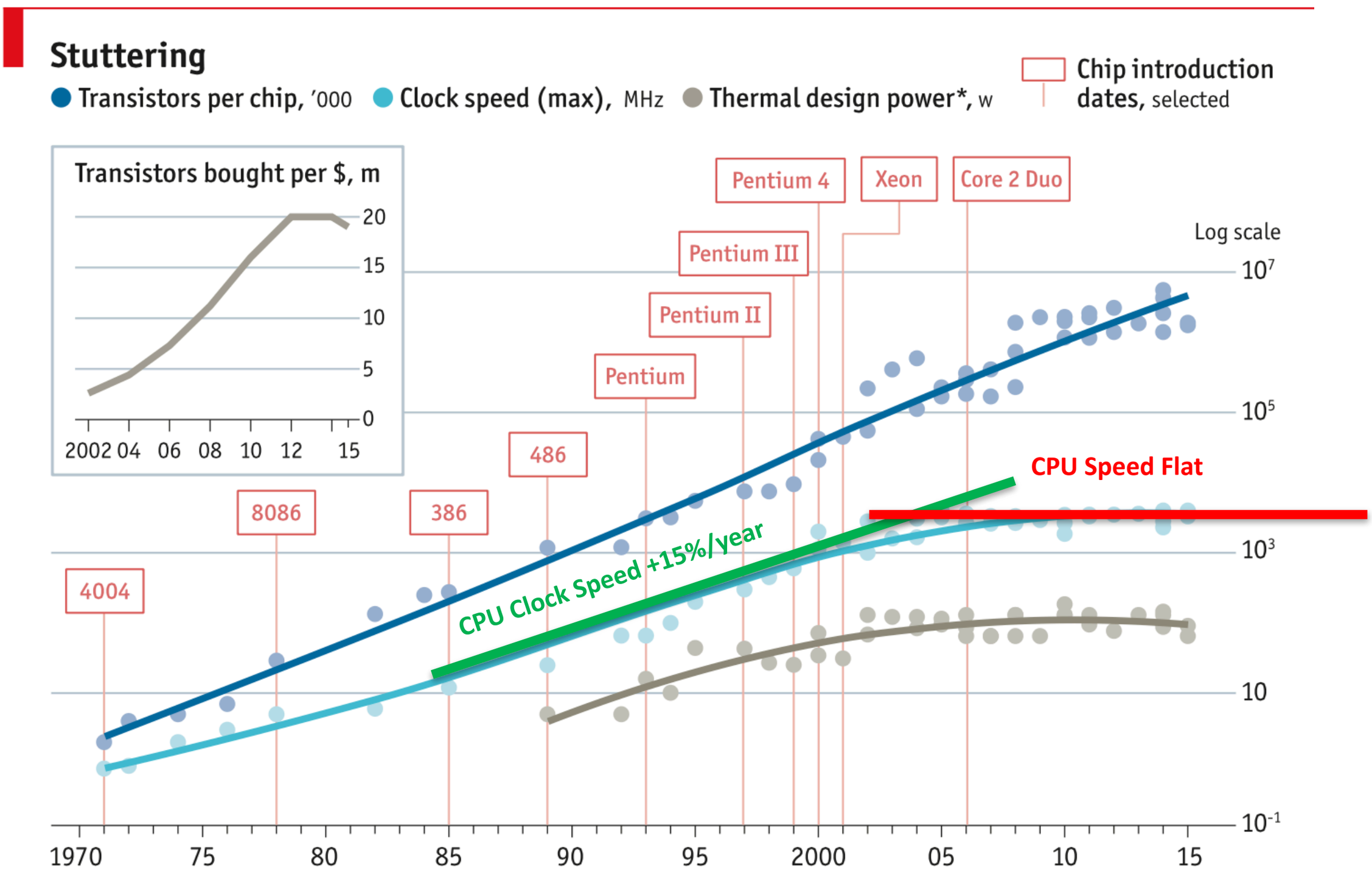


Why is Architecture Exciting Today?



- Number of deployed devices continues growing, but no single killer app
  - Diversification of needs, architectures

Why is Architecture Exciting Today?



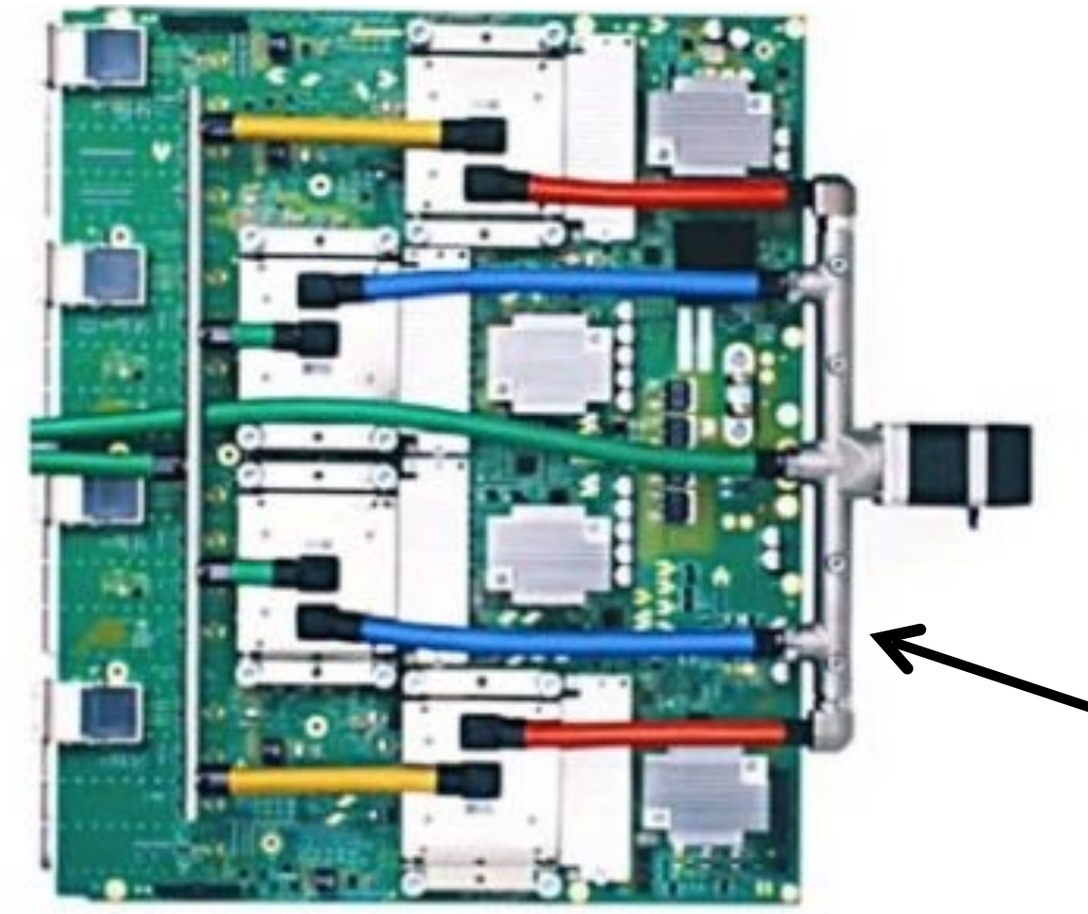
Sources: Intel; press reports; Bob Colwell; Linley Group; IB Consulting; *The Economist*

*Maximum safe power consumption

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 specialized computers
- By time you've finished design, next generation of general-purpose will beat you

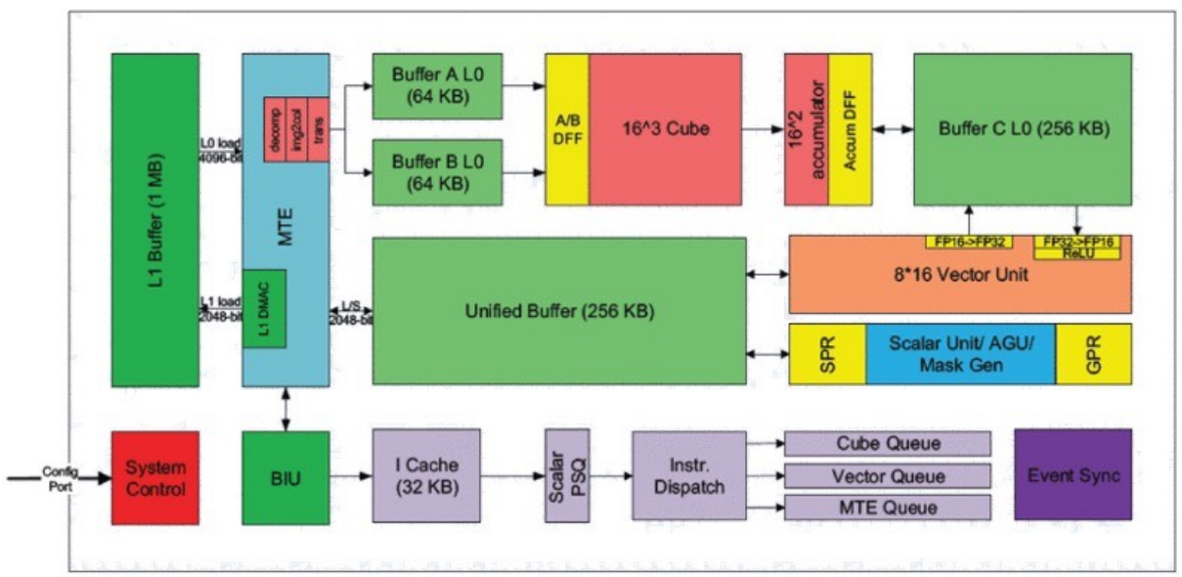
New Conventional Wisdom



- Modern CPUs massively parallel
- Many specialized chips

  Google TPUv4
  Specialized Engine for NN training
  Deployed in cloud
  180+ TFLOPS/chip

  Water Cooling!



Domain-specific architectures

Huawei edge device
Specialized for NN inference
11 TFLOPS FP16; 22 TOPS INT8

Summary

- Great ideas in CA
  - Abstraction (Layers of Representation / Interpretation)
  - Moore's Law
  - Principle of Locality/Memory Hierarchy
  - Parallelism
  - Performance Measurement and Improvement
  - Dependability via Redundancy