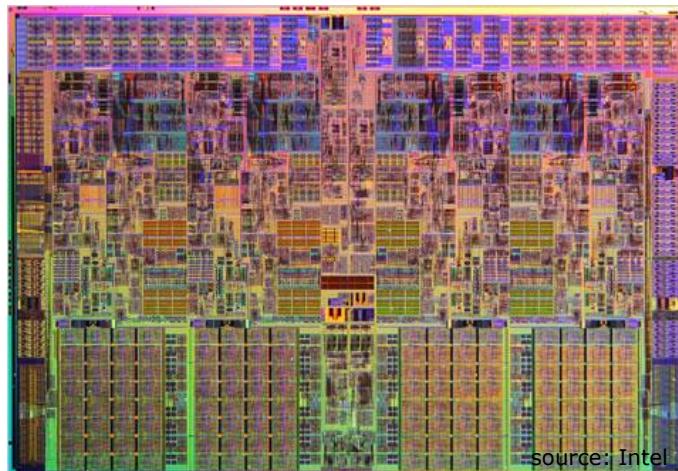


# Introduction to Computer Architecture



# Computer Architecture

“Computer architecture is a **specification detailing how a set of software and hardware technology standards interact to form a computer system or platform**. In short, computer architecture refers to how a computer system is designed and what technologies it is compatible with. “

source: Technopedia

- Computer architecture teaches you
  - how a computer is ***controlled***
  - how a computer is ***built***

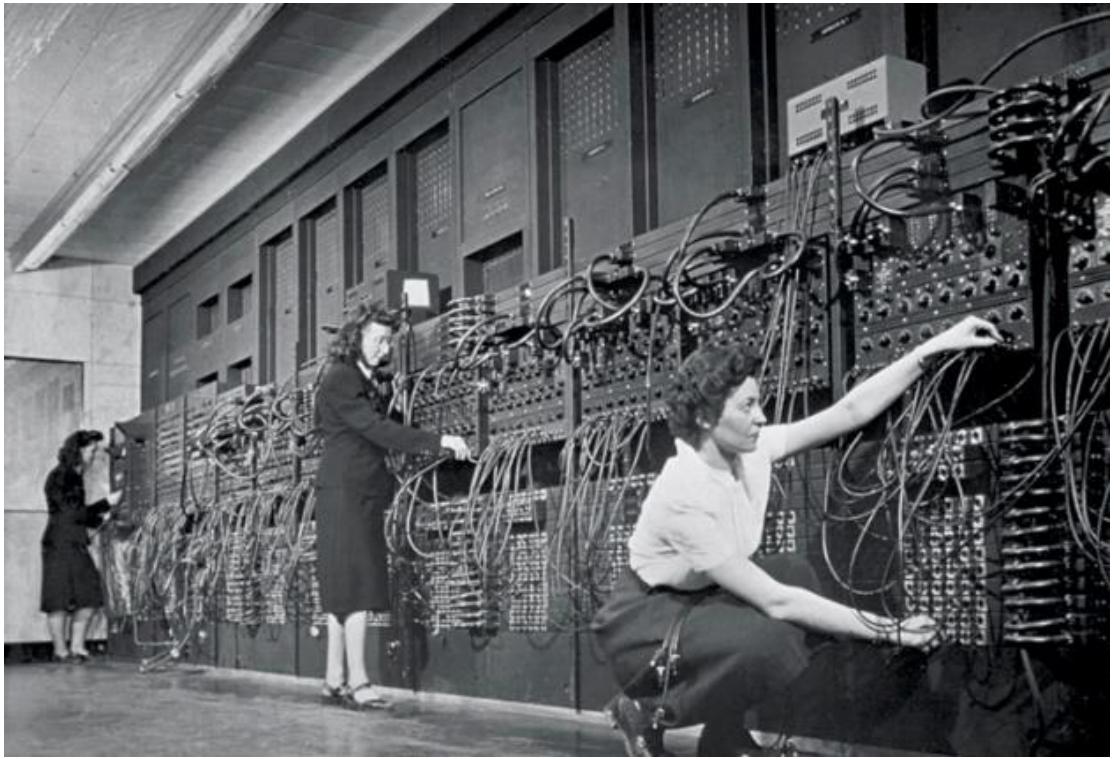
# Credits For The Entire Course

- Slides and material adapted mainly from
  - slides provided with the COD textbook
  - slides provided with the CS:APP textbook
  - Tannenbaum's "Structured Computer Organization"
  - slides and projects from CMU
  - slides from David Black-Schaffer (introduction)
  - slides from Jin-Soo Kim

# Module Outline

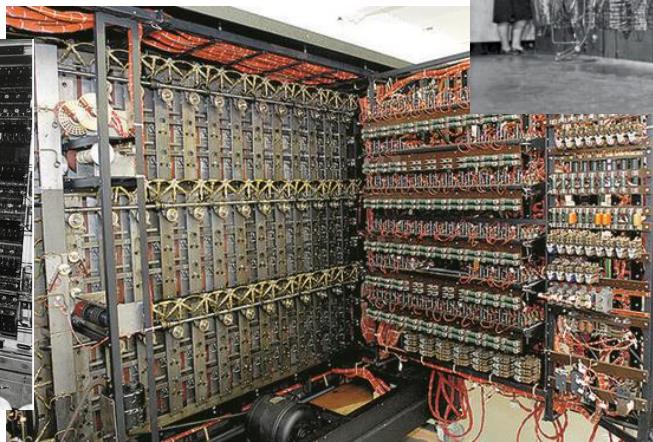
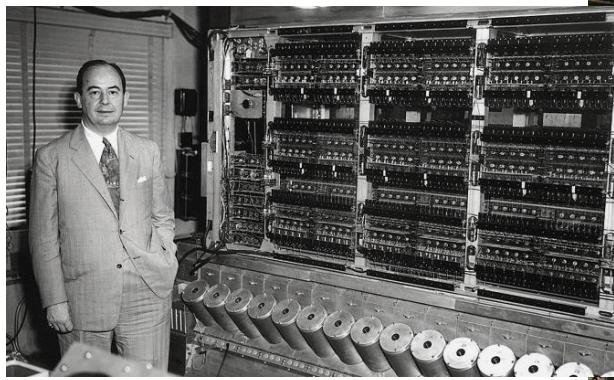
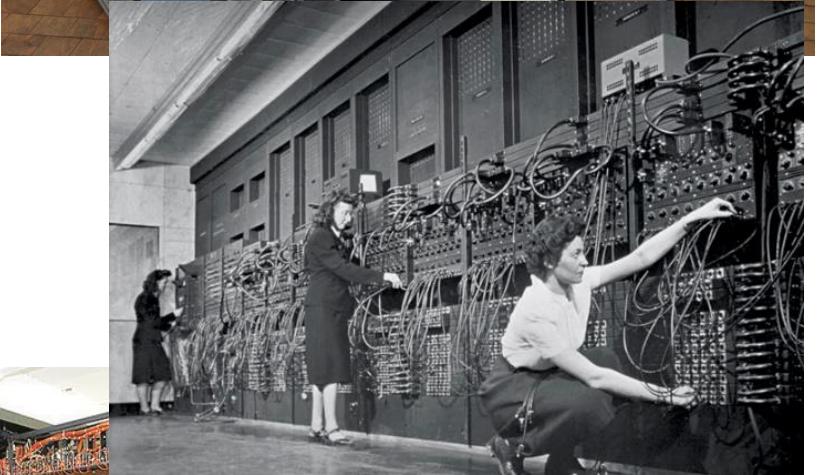
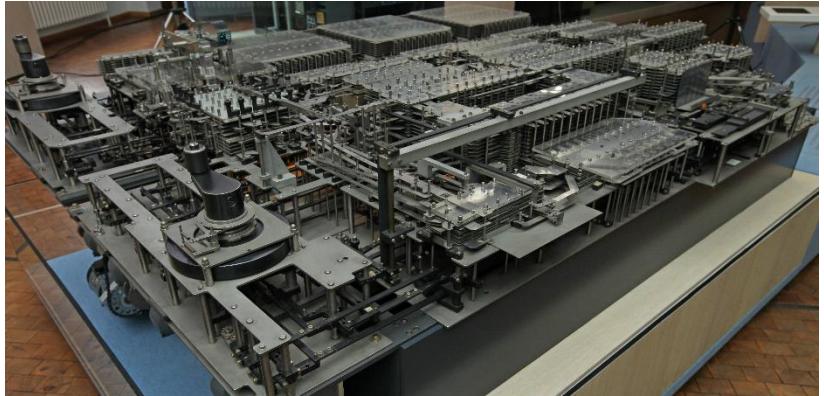
- **Computer History**
  - From Batch To Parallel Processing
  - Processor Architecture Evolution
- **Hardware Organization of a Computer System**
- **How Did We Get Here?**
- **Eight Great Ideas in Computer Architecture**
- **Course Outlook**
- **Module Summary**

# From Batch to Parallel Processing



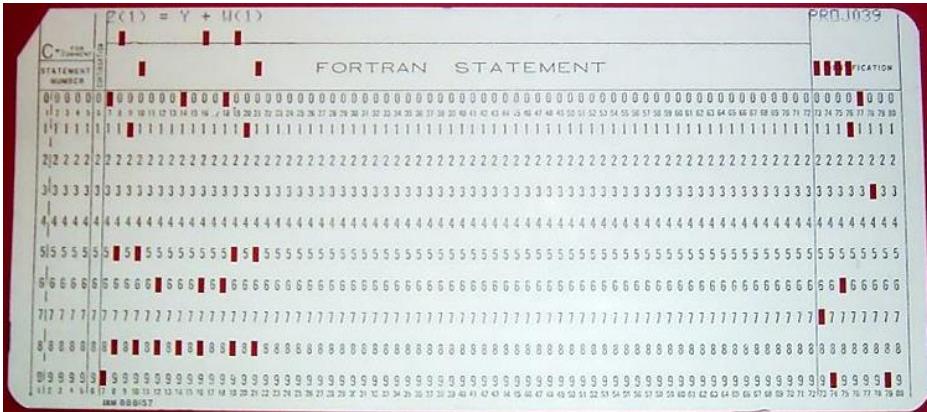
# From Batch To Parallel Processing

- 1940's: special-purpose computers
  - Z1, Colossus, ENIAC  
“Programmable” by rewiring the system
  
- Early 1950's: general-purpose computers
  - EDVAC, bombe
  - programs stored in memory
  - CPU fetch-execute cycle
  - single program, single user at a time



# From Batch To Parallel Processing

- Mid 1950's: batch programming
  - operator combines programs into batches of programs
  - executing a batch meant executing the programs one by one
  - results available after all jobs in the batch had completed
  - “resident monitor”: a first primitive version of system software (OS)
    - ▶ control card interpreter
    - ▶ loader
    - ▶ device drivers



# From Batch To Parallel Processing

- Mid 1950's: batch programming (cont'd)
  - no protection:  
a faulty job reading too many cards, over-writing the resident monitor's memory, or entering an endless loop would affect the entire system
  - lead to:
    - ▶ operating modes (user/monitor)
    - ▶ memory protection
    - ▶ execution timers



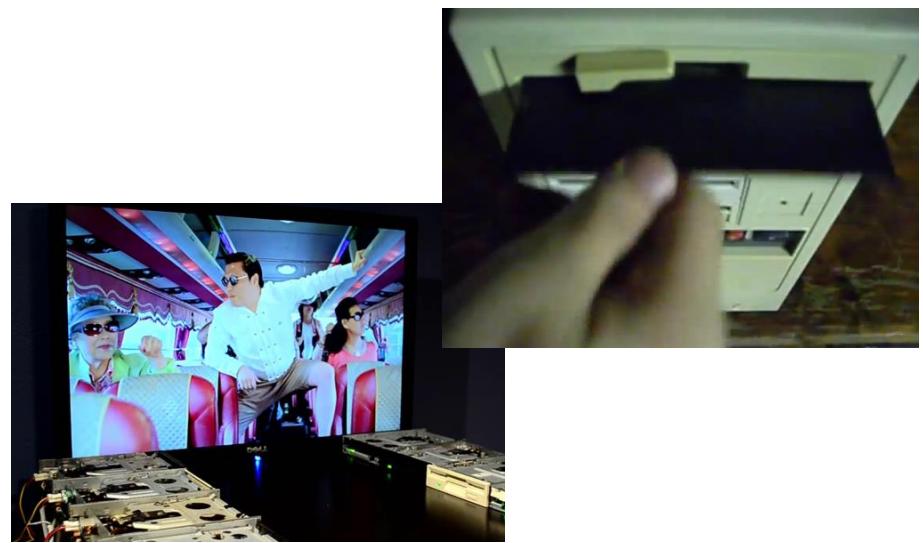
# From Batch To Parallel Processing

- Early 1960's: multiprogramming
  - more memory → keep several programs in memory at once  
(memory partitioning to separate the different jobs)
  - OS monitor could switch between jobs when one became idle (i.e., waiting for I/O)
  - e.g., IBM OS/360
  
- Mid 1960's: timesharing
  - switch between jobs periodically
  - access via remote terminals
  - e.g., CTSS, MULTICS, UNIX



# From Batch To Parallel Processing

- Late 1970's: personal computers
  - single user, dedicated workstation
  - WIMP user interface
  - peripherals connected directly
  - single processor, time-sharing



# From Batch To Parallel Processing

- Since the mid 2000's: portable computing
  - enormous computing power in your pocket
  - single user, dedicated device,  
touch WIMP interface
  - today
    - ▶ octa-core @ up to 3GHz
    - ▶ GPU w/ 768 cores (1.8 GFLOPS FP16)
    - ▶ 8 GB RAM
    - ▶ 256 GB storage



# The Fastest Computer Today

## Frontier (Oak Ridge National Laboratory)

- 8,730,112 cores
- AMD CPU + GPUs
- 74 cabinets
- Interconnect
  - ▶ CPU-GPU: AMD Infinity
  - ▶ system: Slingshot network
- 1.6 EF peak performance
- 1.1 EF Linpack
- #1 on the top500.org list since June 2022



# The Development of Computing Power

| System     | Year | Speed |        |
|------------|------|-------|--------|
| Z1         | 1938 | 1.00  | IPS    |
| ENIAC      | 1946 | 5.00  | kIPS   |
| Atlas      | 1962 | 1.00  | MFLOPS |
| Cray-2     | 1985 | 1.41  | GFLOPS |
| ASCI Red   | 1997 | 1.06  | TFLOPS |
| Roadrunner | 2008 | 1.02  | PFLOPS |
| Frontier   | 2022 | 1.10  | EFLOPS |

- Your smartphone provides about the same performance as the world's fastest supercomputer from 1997 – 2000, ASCI Red. Yet it (reference: Samsung Galaxy S22),
  - requires about 25 times fewer processors to do so
  - is about 70'000 times cheaper
  - consumes about 450'000 times less power



# Processor Architecture Evolution

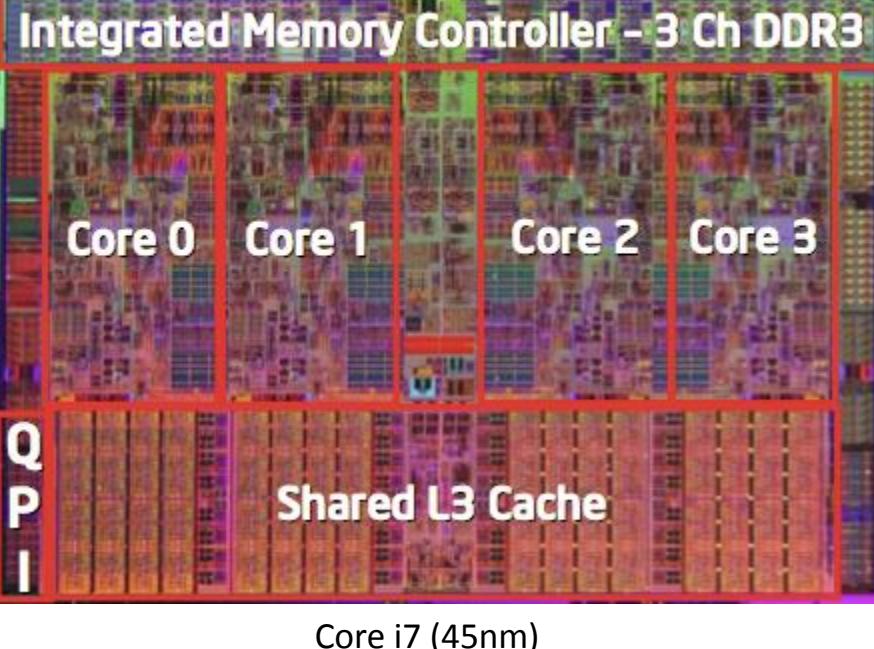
# Intel x86 Evolution: Milestones

| Name   | Date | Transistors | MHz       |
|--|------|-------------|-----------|
| ■ 8086   | 1978 | 29K         | 5-10      |
|  |      |             |           |
| ● First 16-bit processor. Basis for IBM PC & DOS                   |      |             |           |
| ● 1MB address space (RAM)  |      |             |           |
| ■ 386  | 1985 | 275K        | 16-33     |
|  |      |             |           |
| ● First <u>32 bit processor</u> , referred to as IA32              |      |             |           |
| ● Added “flat addressing”  |      |             |           |
| ● Capable of running Unix  |      |             |           |
| ● 32-bit Linux/gcc uses no instructions introduced in later models |      |             |           |
| ■ Pentium 4F   | 2004 | 125M        | 2800-3800 |
|  |      |             |           |
| ● First 64-bit processor, referred to as <u>x86-64</u>             |      |             |           |
| ■ Core i7  | 2008 | 731M        | 2667-3333 |
|  |      |             |           |
| ● multiple cores   |      |             |           |

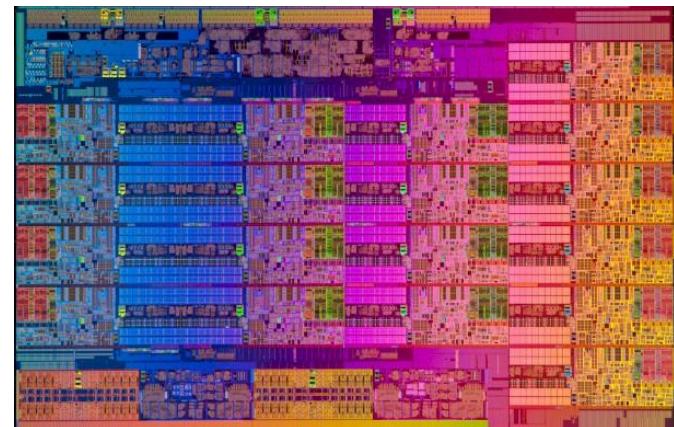
# Intel x86 Processors

## Machine Evolution

| Name                            | Date | Transistors   |
|---------------------------------|------|---------------|
| • 386                           | 1985 | 0.3M          |
| • 486                           | 1989 | 1.9M          |
| • Pentium                       | 1993 | 3.1M          |
| • Pentium/MMX                   | 1997 | 4.5M          |
| • PentiumPro                    | 1995 | 6.5M          |
| • Pentium III                   | 1999 | 8.2M          |
| • Pentium 4                     | 2001 | 42M           |
| • Core 2 Duo                    | 2006 | 291M          |
| • Core i7 (4 cores)             | 2008 | 731M          |
| • Core i7 (6 cores)             | 2011 | 2'270M        |
| • Xeon E5 v4 (22 cores)         | 2016 | 7'200M (est.) |
| • <u>Xeon Plat. 8284 (28 c)</u> | 2019 | 8'000M (est.) |



Core i7 (45nm)



Xeon E5 v3 (22nm)

## Added Features

- Instructions to support multimedia operations
  - ▶ Parallel operations on 1, 2, and 4-byte data, both integer & FP
- Instructions to enable more efficient conditional operations

# x86 Clones: Advanced Micro Devices (AMD)

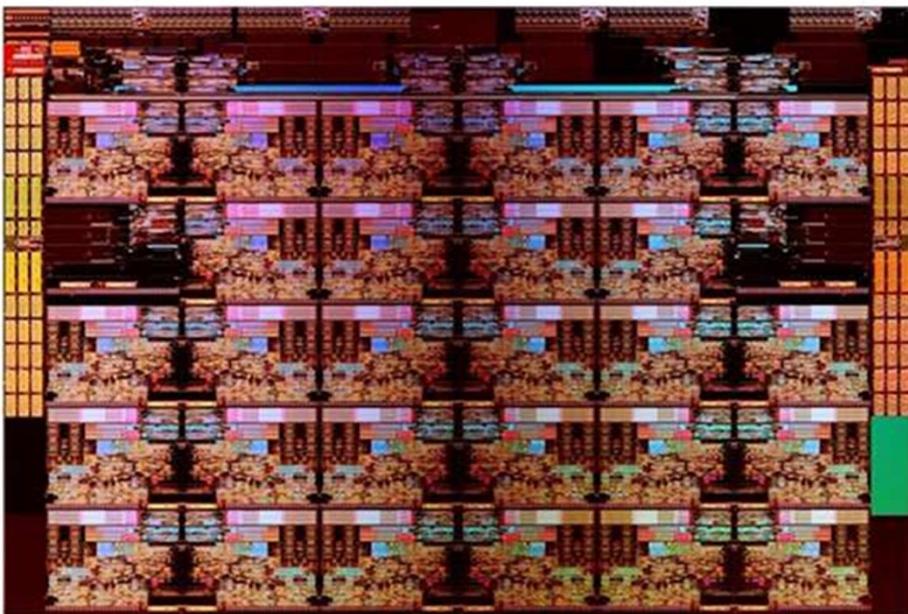
- Historically
  - AMD has followed just behind Intel
  - A little bit slower, a lot cheaper
- Then
  - Recruited top circuit designers from Digital Equipment Corp. and other downward trending companies
  - Built Opteron: tough competitor to Pentium 4
  - Developed x86-64, their own extension to 64 bits
- About 10 years back
  - Intel much quicker with multi-core design
  - Intel far ahead in performance
  - Intel em64t backwards compatible to x86-64
- Today (2022)
  - AMD outperforms Intel in terms of cost, #cores, design, and matches single-thread performance!

# Intel's 64-Bit

- Intel Attempted Radical Shift from IA32 to IA64
  - Totally different architecture (Itanium)
  - Executes IA32 code only as legacy
  - Application performance disappointing
- AMD Stepped in with Evolutionary Solution
  - x86-64 (now called “AMD64”)
- Intel Felt Obligated to Focus on IA64
  - Hard to admit mistake or that AMD is better
- 2004: Intel Announces EM64T extension to IA32
  - Extended Memory 64-bit Technology
  - Almost identical to x86-64

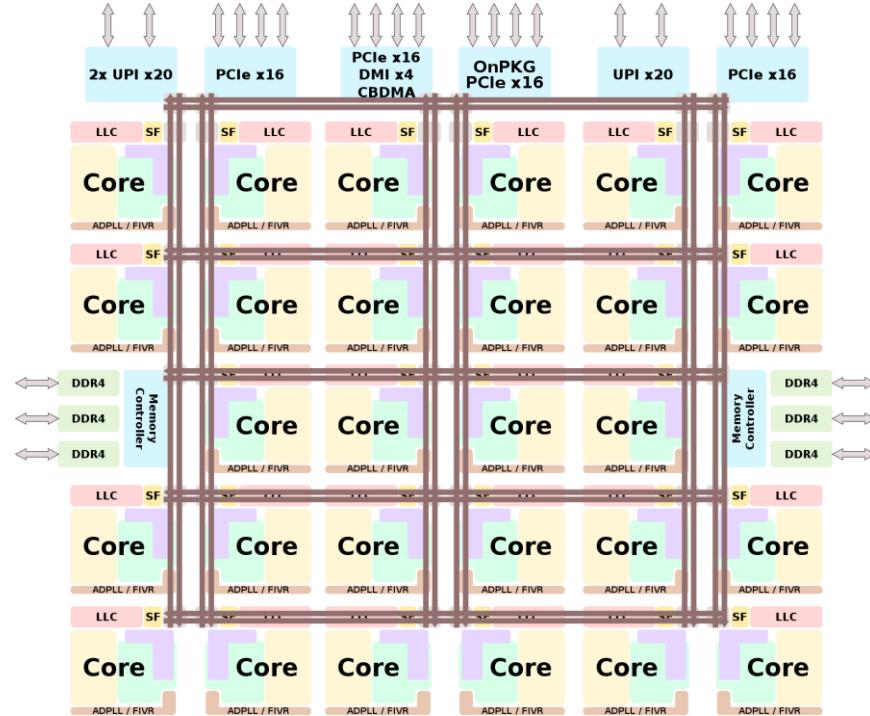
# Intel Cascade Lake SP

- Current leader: Xeon Platinum 8284  
(Cascade Lake, July 2019)
  - ~8 billion transistors
  - 3/4 GHz
  - virtualization support
  - no GPU on chip



source: [WikiChip](#)

19



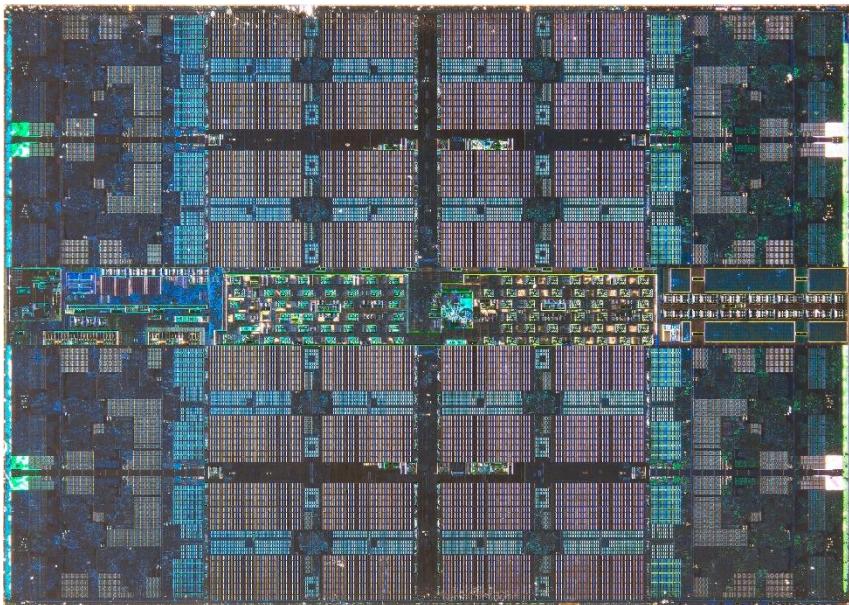
Source: [WikiChip](#)

- up to 28 cores, 56 threads
- per core: 896KB L1(I/D),  
1MB L2 cache
- 38.5MB shared L3 cache
- 240W TPD

# AMD Zen Architecture

## ■ AMD Threadripper 3990X (Zen2, July 2019)

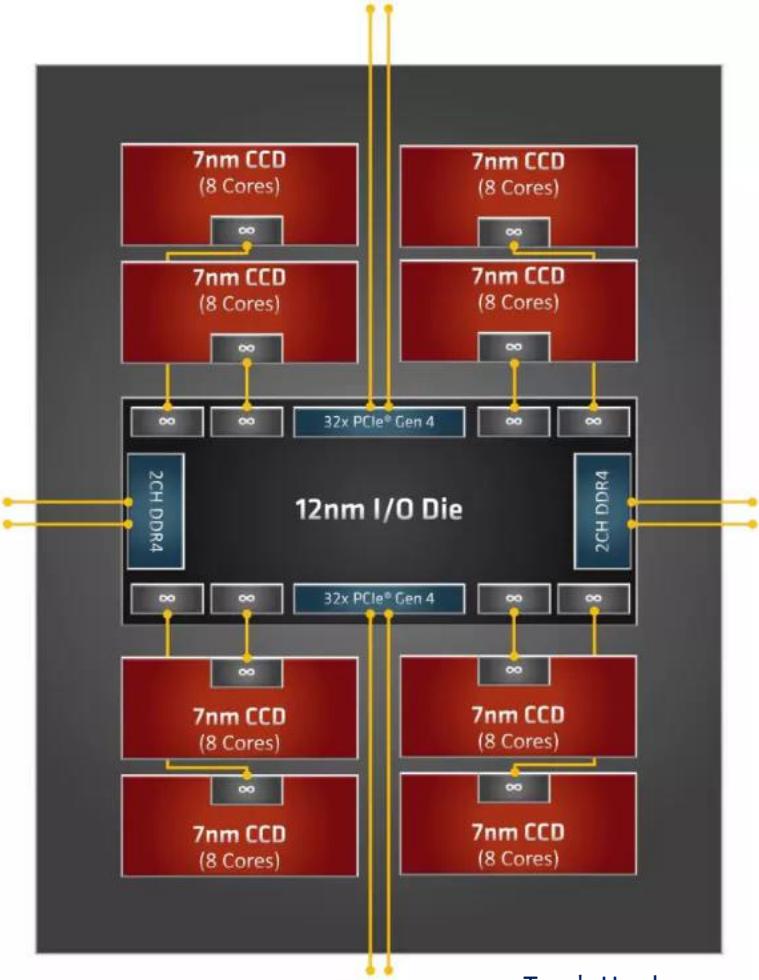
- 64 cores, 128 threads
- 40 billion transistors  
(4 billion per 8-core chiplet)
- 3/4 GHz



4190.308 Computer Architecture, Spring 2023

source: [flickr](#)

20



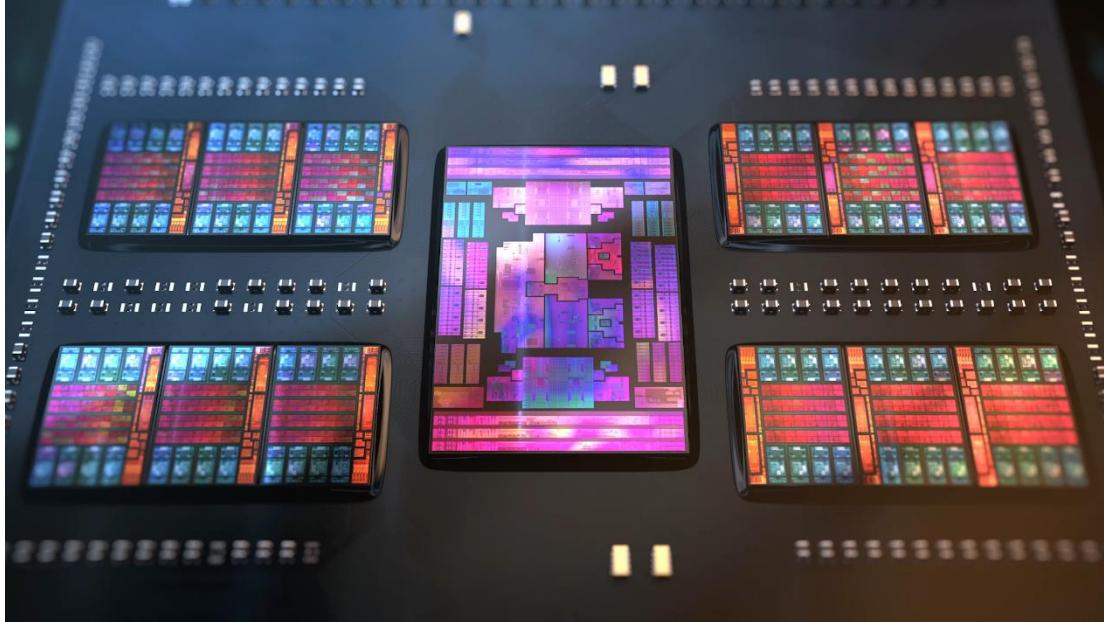
source: [Tom's Hardware](#)

- Built by combining 8-core ‘core-chiplets’
- 32MB L2, 256MB L3 cache
- 280W TPD

# AMD Zen Architecture

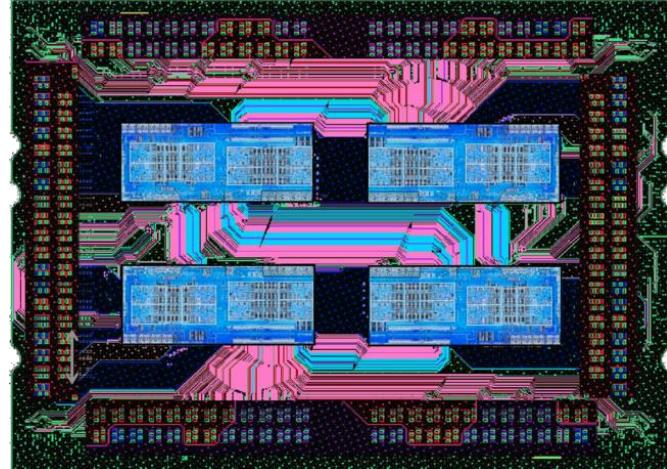
## ■ Zen 4 (October 2022)

- 5nm technology
- CCD (Core Chiplet Die)  
8 cores/16 threads, 72 mm<sup>2</sup>, 6.57 billion transistors
- IOD: 397 mm<sup>2</sup>



Zen 4 Architecture (October 2022)

source: [WCCFtech](#)



Zen Architecture (2017)

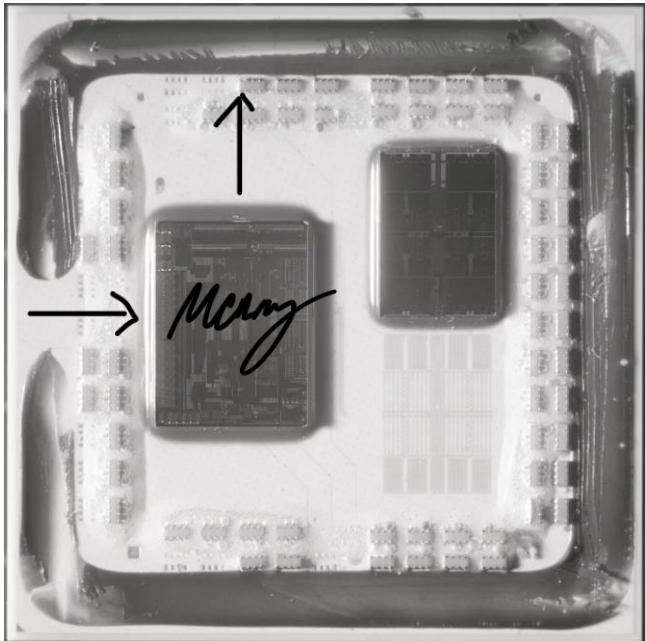
source: [Wikichip](#)

# AMD Zen 4: 32, 64, and 96 Core Configurations



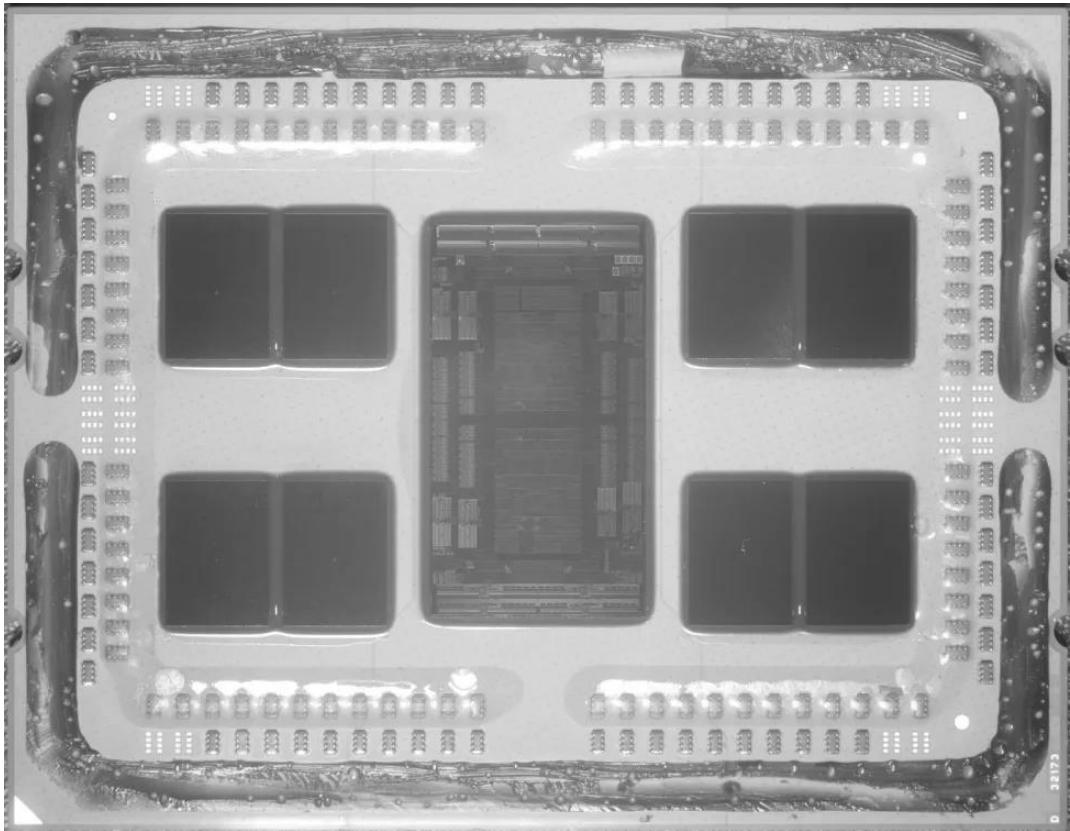
source: AMD

# AMD Zen 4 CCD and IOD



AMD Ryzen 5 3600

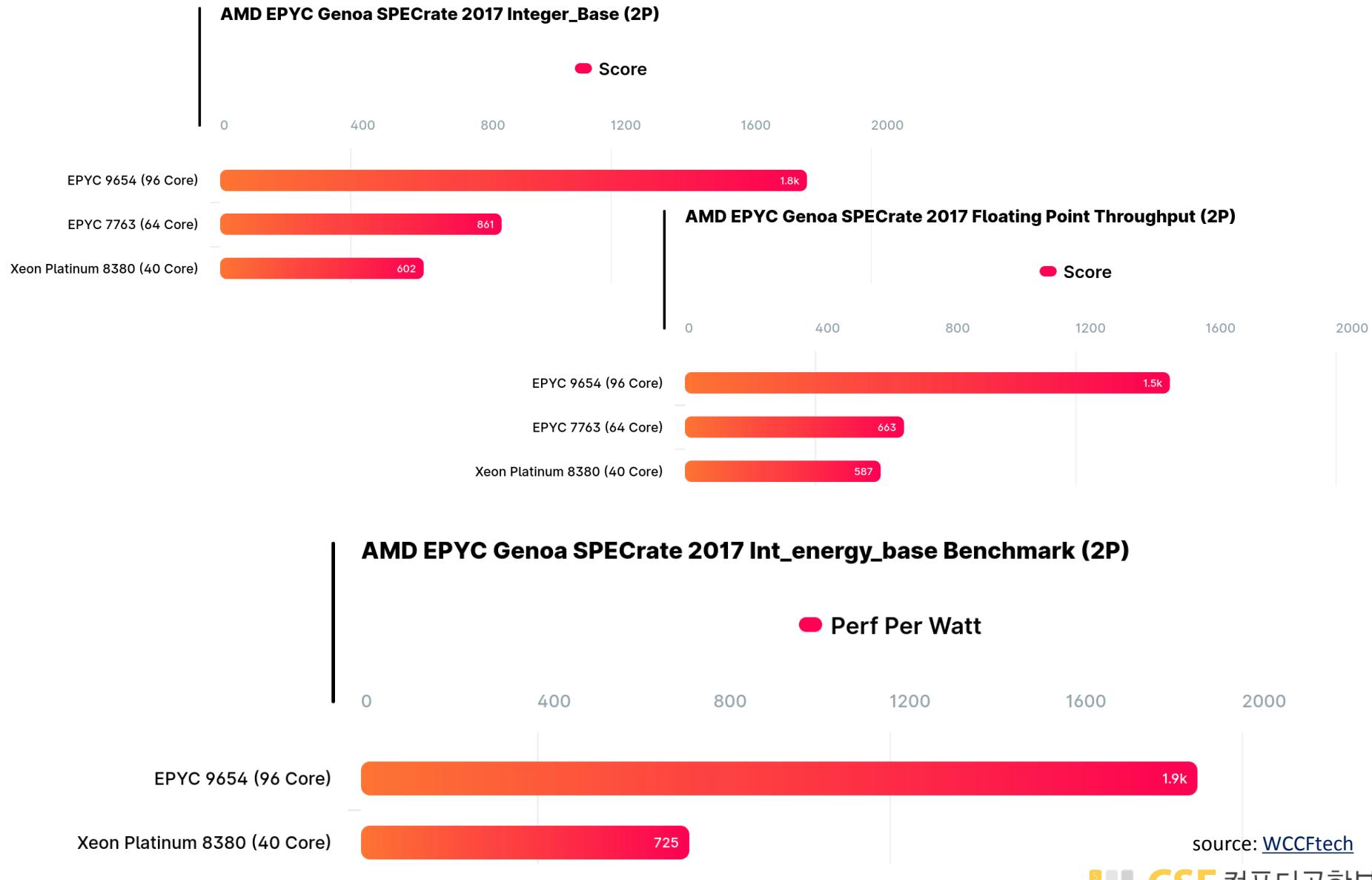
source: [Tom's Hardware](#)



AMD Rome CPU (64 cores)

source: [Tom's Hardware](#)

# AMD Zen 4 Performance



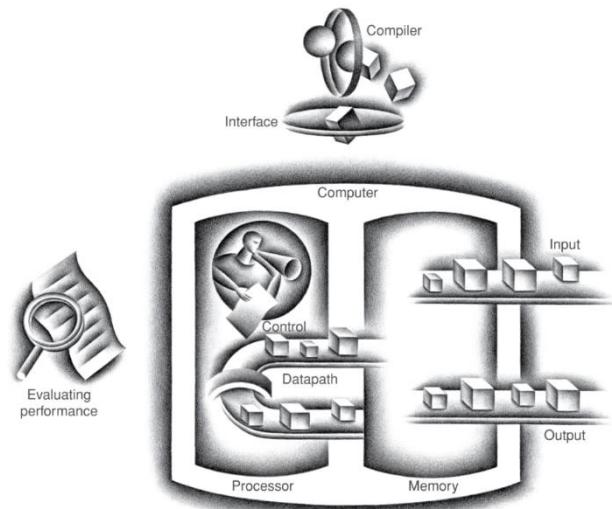
# AMD EPYC 9004 "Genoa Zen 4"

## AMD EPYC 9004 "Genoa Zen 4" Server CPU SKUs:

| CPU NAME   | ARCHITECTURE | FAMILY | TOTAL CCDS | CORES / THREADS | L3 CACHE | BASE / MAX CLOCKS | TDP             | PRICE (1000 UNIT MSRP) |
|------------|--------------|--------|------------|-----------------|----------|-------------------|-----------------|------------------------|
| EPYC 9664  | 5nm Zen 4    | Genoa  | 12         | 96/192          | 384 MB   | 2.25-3.80 GHz     | 400W (320-400W) | TBD                    |
| EPYC 9654  | 5nm Zen 4    | Genoa  | 12         | 96/192          | 384 MB   | 2.40 / 3.70 GHz   | 360W (320-400W) | \$11,805               |
| EPYC 9654P | 5nm Zen 4    | Genoa  | 12         | 96/192          | 384 MB   | 2.40 / 3.70 GHz   | 360W (320-400W) | \$10,625               |
| EPYC 9634  | 5nm Zen 4    | Genoa  | 12         | 84/168          | 384 MB   | 2.25 / 3.70 GHz   | 290W (320-400W) | \$10,304               |
| EPYC 9554  | 5nm Zen 4    | Genoa  | 8          | 64/128          | 256 MB   | 3.10 / 3.75 GHz   | 360W (320-400W) | \$9,087                |
| EPYC 9554P | 5nm Zen 4    | Genoa  | 8          | 64/128          | 256 MB   | 3.10 / 3.75 GHz   | 360W (320-400W) | \$7,104                |

source: [WCCFtech](#)

# Hardware Organization of a Computer System



# Various Classes of Computers

## ■ Personal computers

- General purpose, variety of software
- Subject to cost/performance tradeoff

## ■ Supercomputers

- High-end scientific and engineering calculations
- Highest capability but represent a small fraction of the overall computer market

## ■ Server computers

- Network based
- High capacity, performance, reliability
- Range from small servers to large data centers

## ■ Embedded computers

- Hidden as components of systems
- Stringent power/performance/cost constraints

# Opening the Box: Desktop Computer



# Opening the Box: Desktop Computer



# Opening the Box: Desktop Computer



# Opening the Box: Samsung Galaxy S20 Ultra



image source: Samsung Electronics

# Opening the Box: Samsung Galaxy S20 Ultra



image source: iFixIT

# Opening the Box: Samsung Galaxy S20 Ultra

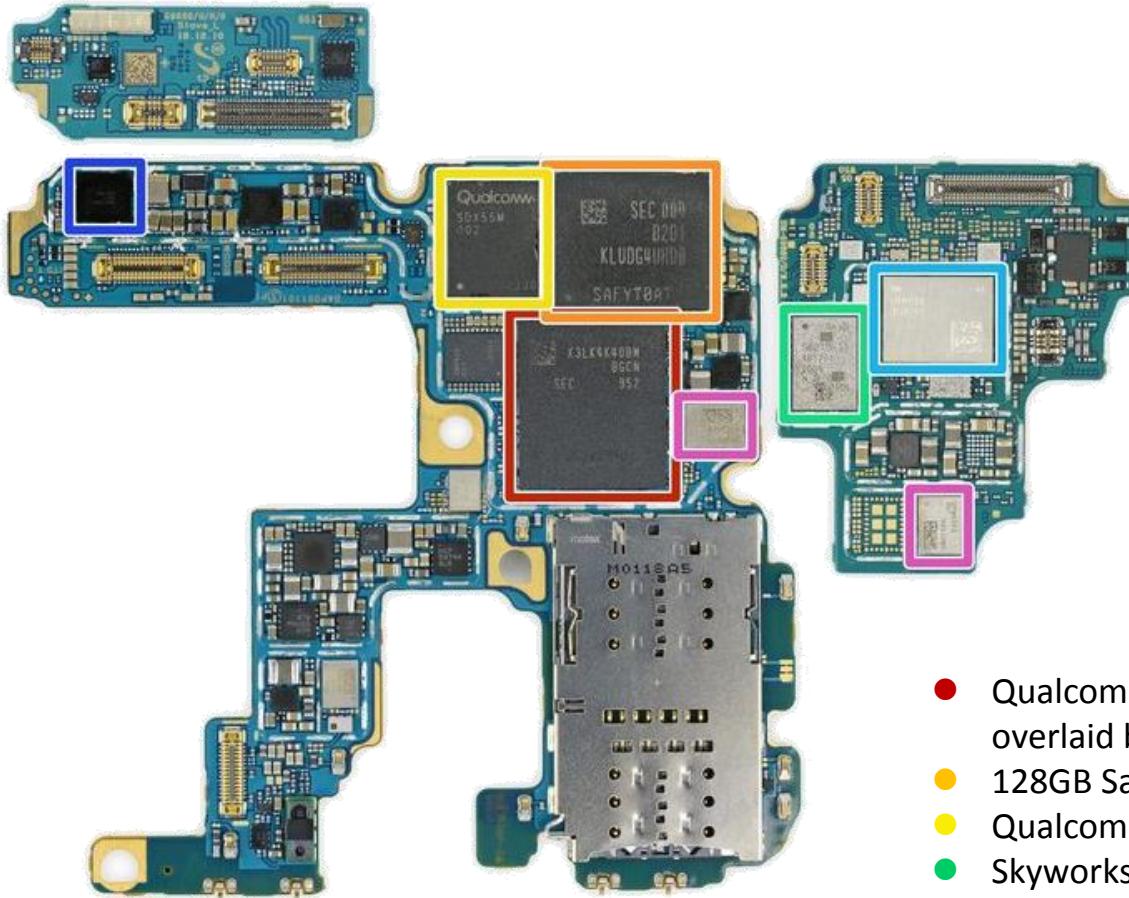


image source: iFIXIT

- Qualcomm Snapdragon 865 processor (8 cores), overlaid by 12GB Samsung LPDDR5 RAM
- 128GB Samsung flash storage
- Qualcomm 5G modem
- Skyworks RF module
- Qorvo WiFi module
- Maxim power management IC
- Qualcomm power amplification modules

# Opening the Box: Apple iPad 7



image source: Apple

# Opening the Box: Apple iPad 7

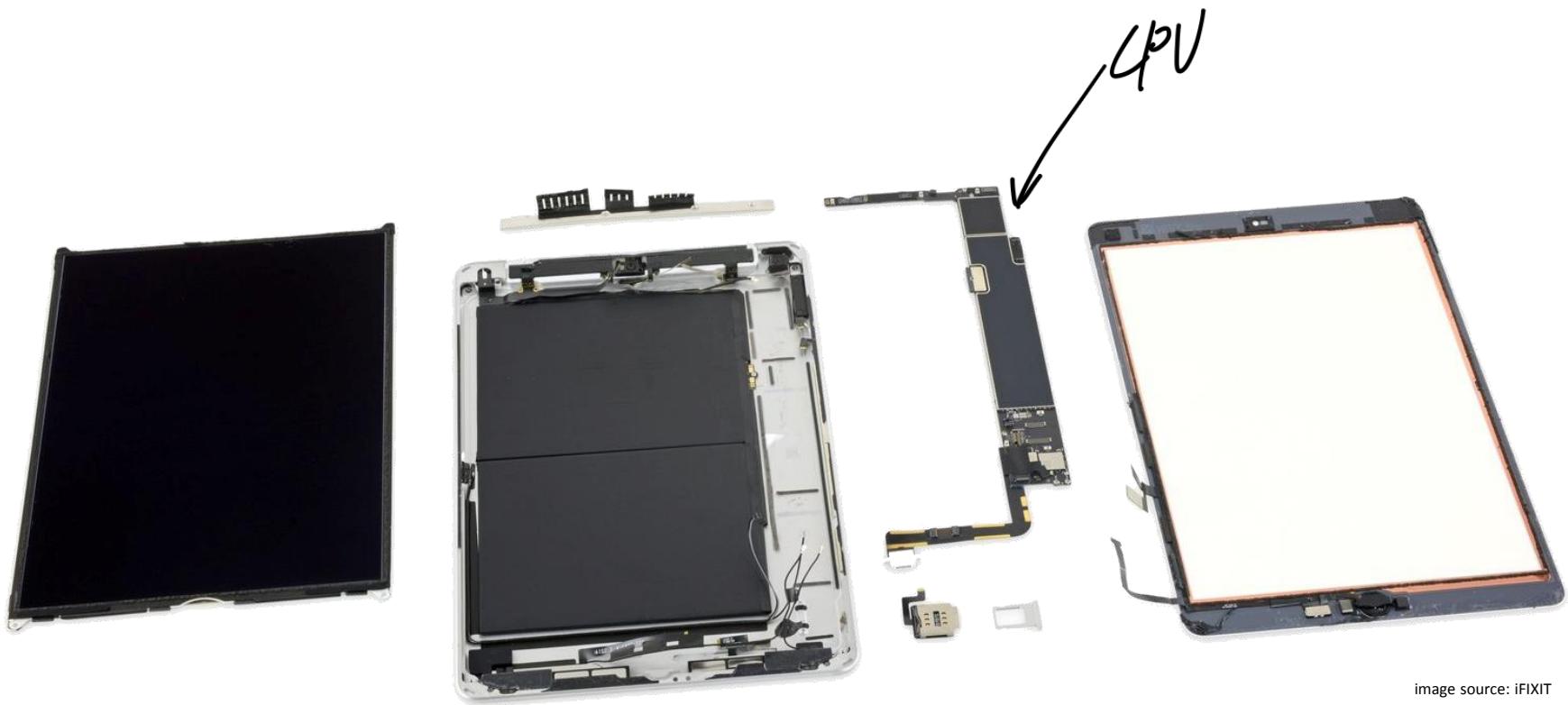
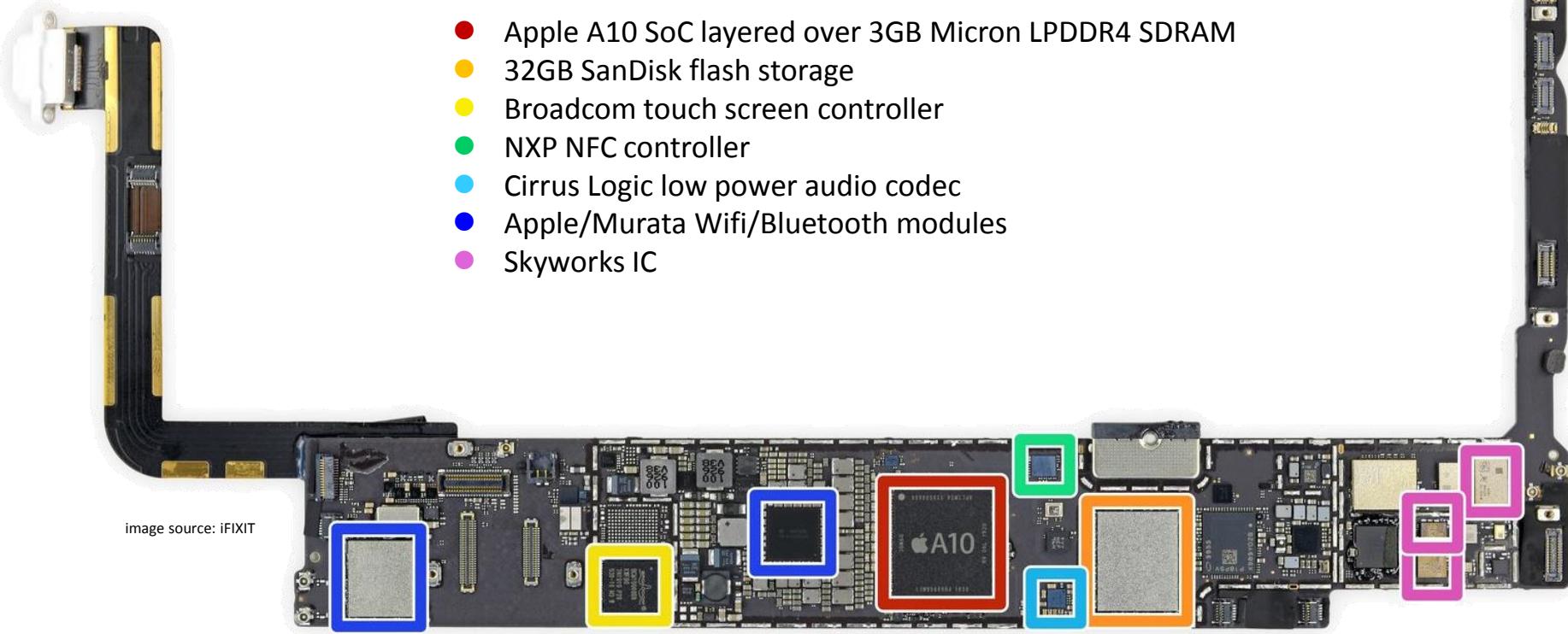


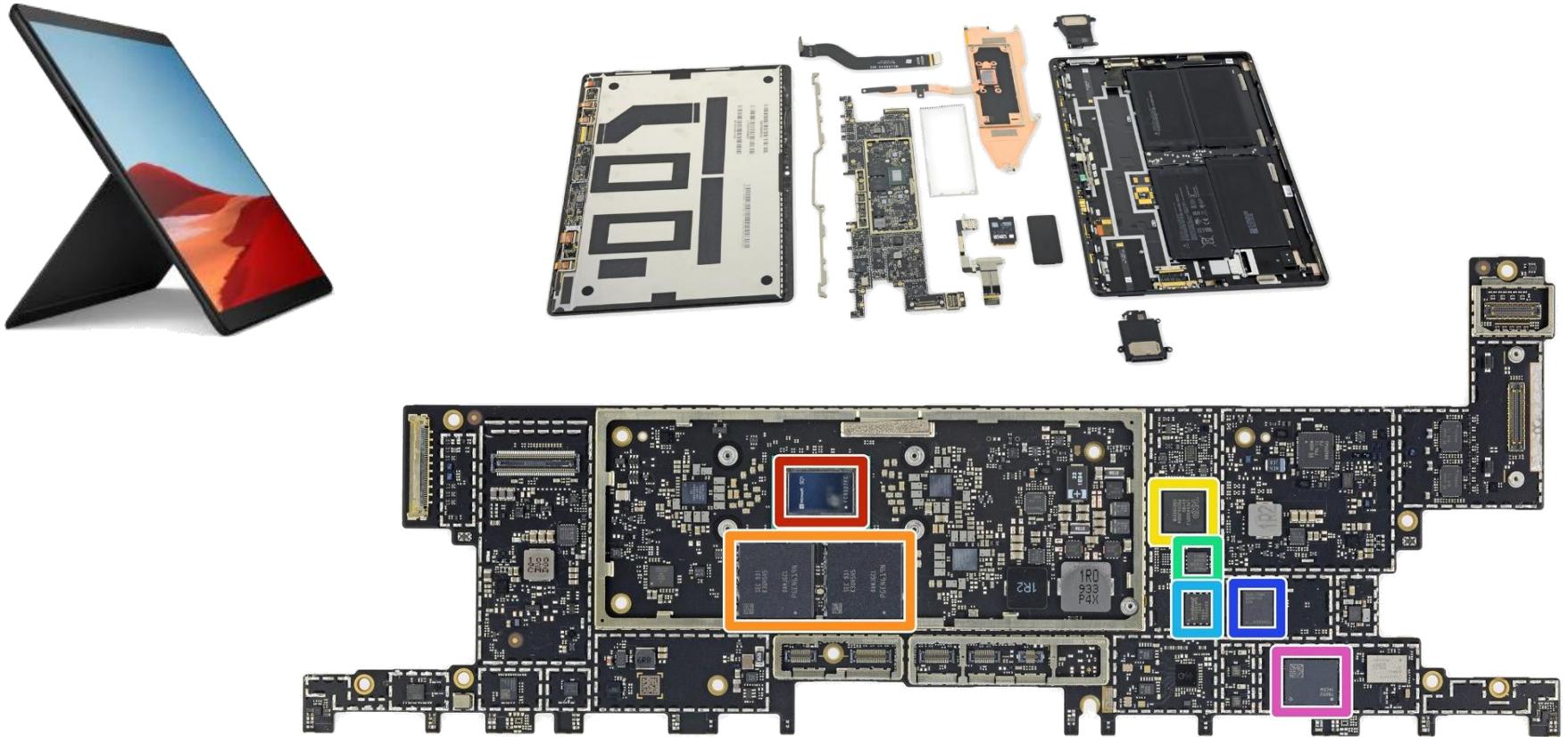
image source: iFIXIT

# Opening the Box: Apple iPad 7

- Apple A10 SoC layered over 3GB Micron LPDDR4 SDRAM
- 32GB SanDisk flash storage
- Broadcom touch screen controller
- NXP NFC controller
- Cirrus Logic low power audio codec
- Apple/Murata Wifi/Bluetooth modules
- Skyworks IC



# Opening the Box: Microsoft Surface Pro



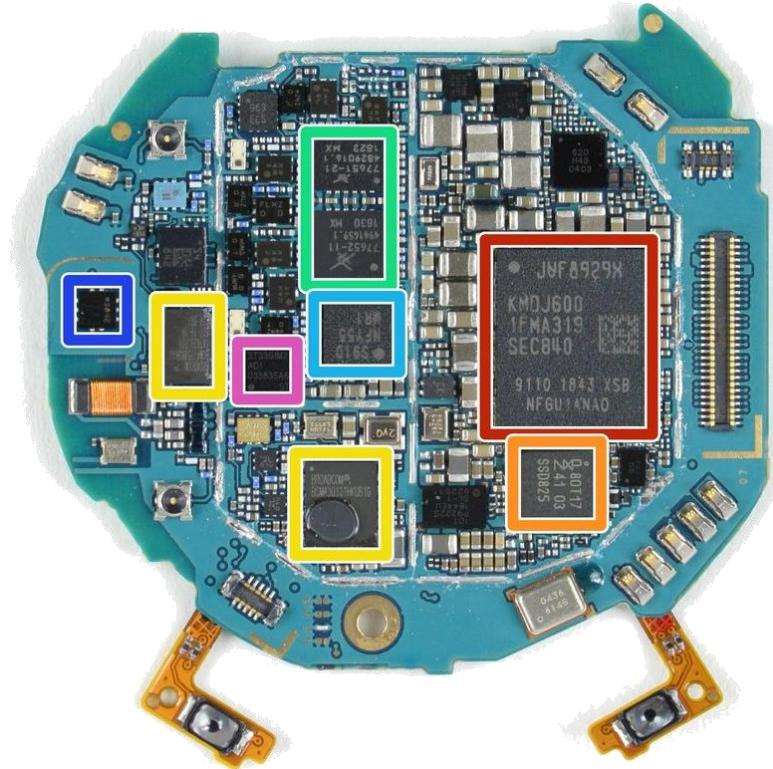
- Microsoft ARM processor
- 2x4GB Samsung LPDDR4X RAM
- NXP EV180 microcontroller
- Macronix 16Mb serial NOR flash memory
- Winbond 256 Mb serial flash memory
- Qualcomm RF module
- Qorvo WiFi module

image sources: Microsoft, iFixit

# Opening the Box: Samsung Galaxy Watch



- Samsung Exynos 9110 (dual core)
- NXP NFC module
- Broadcom Wifi/Bluetooth modules



- Skyworks power amplifiers
- STMicroelectronics barometric pressure sensor
- ST Micro 32-bit ARM SecurCore

image sources: Samsung, iFixit

# Opening the Box

- Not all cases are that tidy



- Or server rooms...



# The Fastest Computer Today

## Frontier (Oak Ridge National Laboratory)

- 8,730,112 cores
- AMD CPU + GPUs
- 74 cabinets
- Interconnect
  - ▶ CPU-GPU: AMD Infinity
  - ▶ system: Slingshot network
- 1.6 EF peak performance
- 1.1 EF Linpack
- #1 on the top500.org list since June 2022



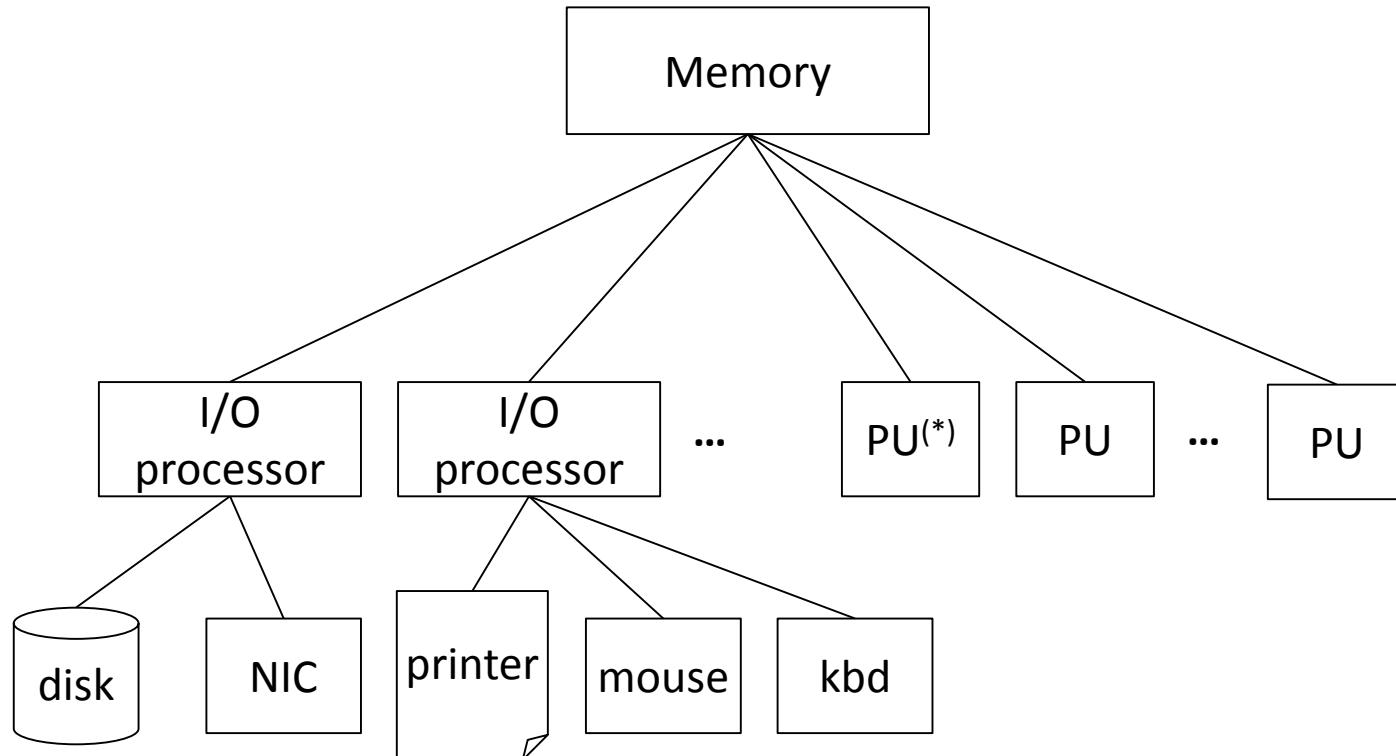
# The Development of Computing Power

| System     | Year | Speed |        |
|------------|------|-------|--------|
| Z1         | 1938 | 1.00  | IPS    |
| ENIAC      | 1946 | 5.00  | kIPS   |
| Atlas      | 1962 | 1.00  | MFLOPS |
| Cray-2     | 1985 | 1.41  | GFLOPS |
| ASCI Red   | 1997 | 1.06  | TFLOPS |
| Roadrunner | 2008 | 1.02  | PFLOPS |
| Frontier   | 2022 | 1.10  | EFLOPS |

- Your smartphone provides about the same performance as the world's fastest supercomputer from 1997 – 2000, ASCI Red. Yet it (reference: Samsung Galaxy S22),
  - requires about 25 times fewer processors to do so
  - is about 70'000 times cheaper
  - consumes about 450'000 times less power

# Components of a Computer

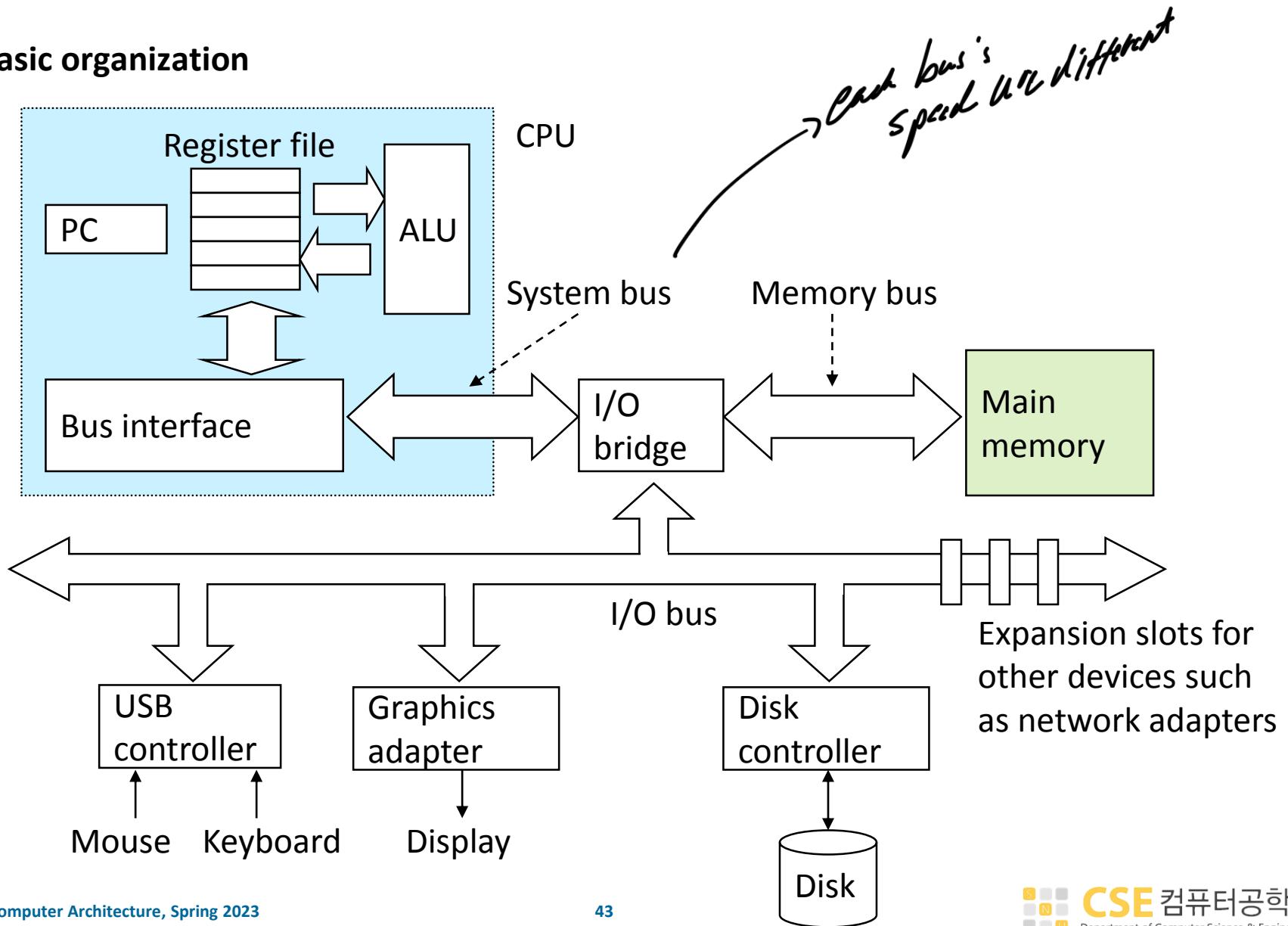
## ■ Abstract hardware organization



(\*) PU = Processing Unit

# Components of a Computer

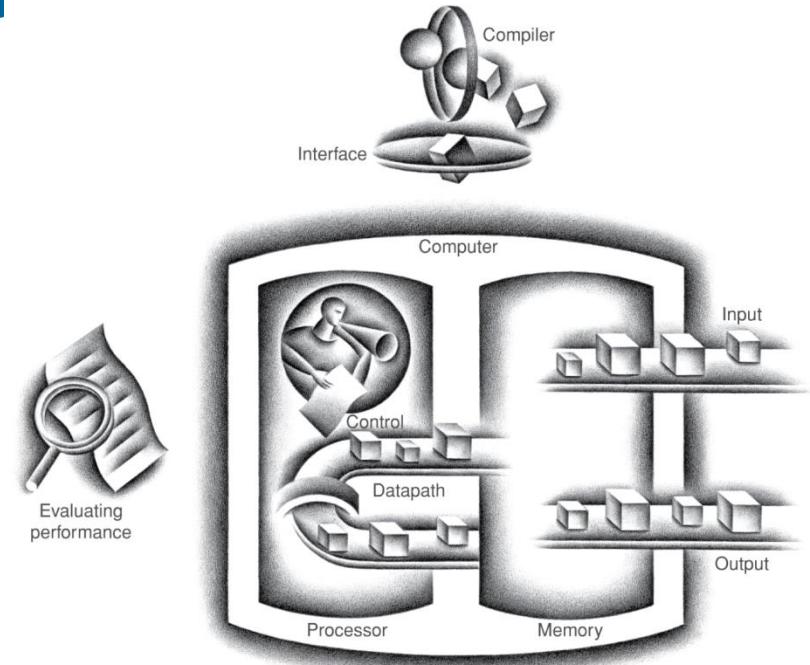
## ■ Basic organization



# Components of a Computer

- CPU: Control + Datapath
- Memory
- I/O

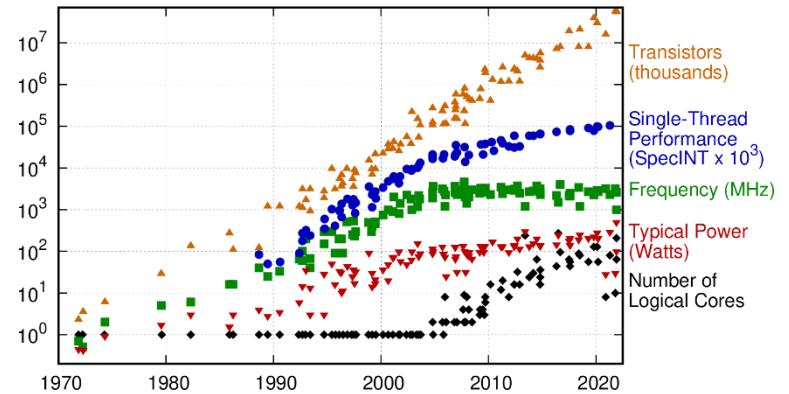
- User-interface devices:  
Display, keyboard, mouse, sound, ...
- Storage devices:  
HDD, SSD, CD/DVD, ...
- Network adapters:  
Ethernet, 3G/4G/5G, WiFi, Bluetooth, ...



# Computer Organization

## ■ Still to this day

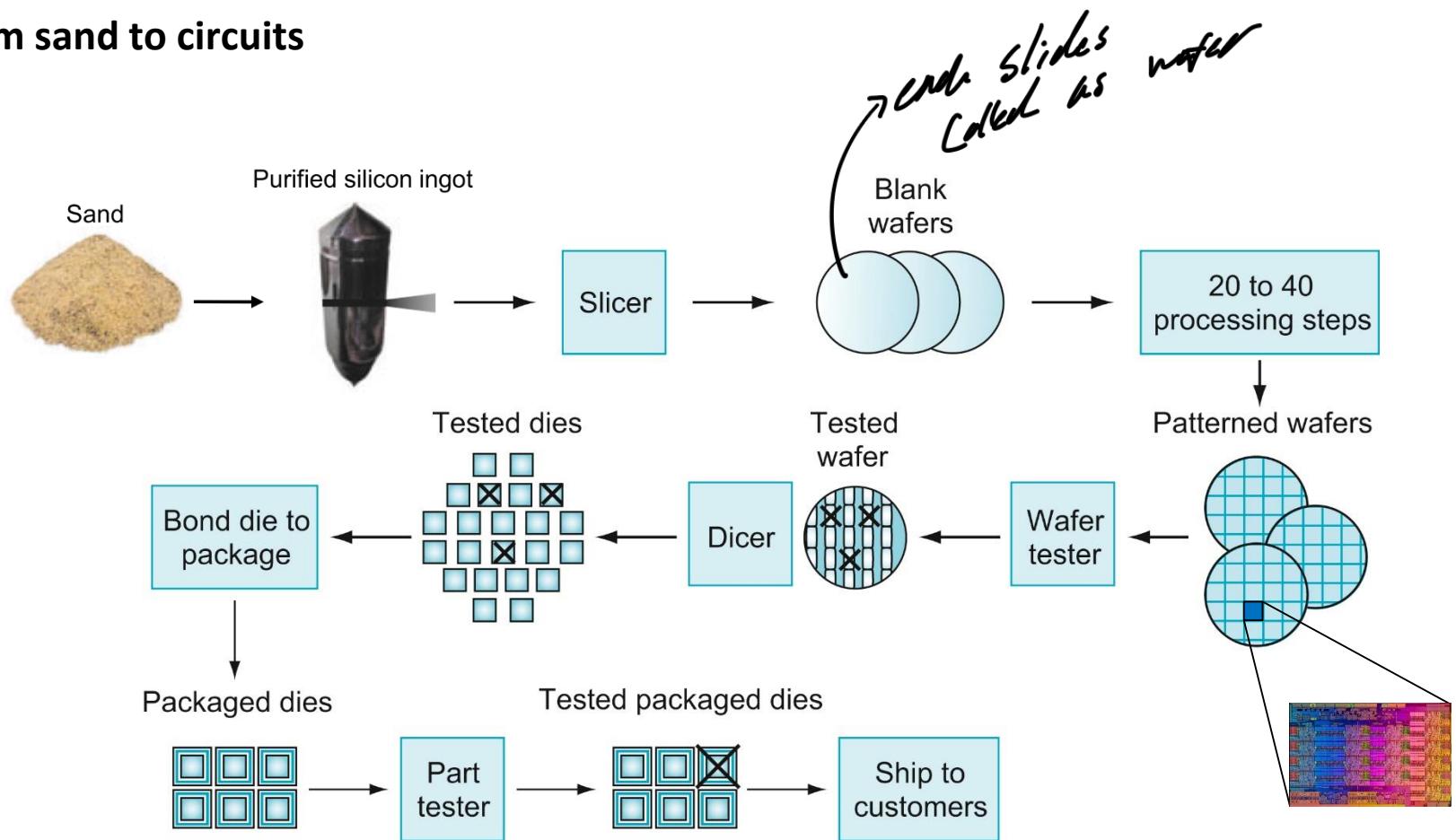
- a single (single-threaded) application executes under the assumption it runs exclusively on the hardware
  - ▶ private memory
  - ▶ its own CPU
  - ▶ uninterrupted access to peripherals
- yet, programs may
  - ▶ be multi-threaded
  - ▶ communicate with each other



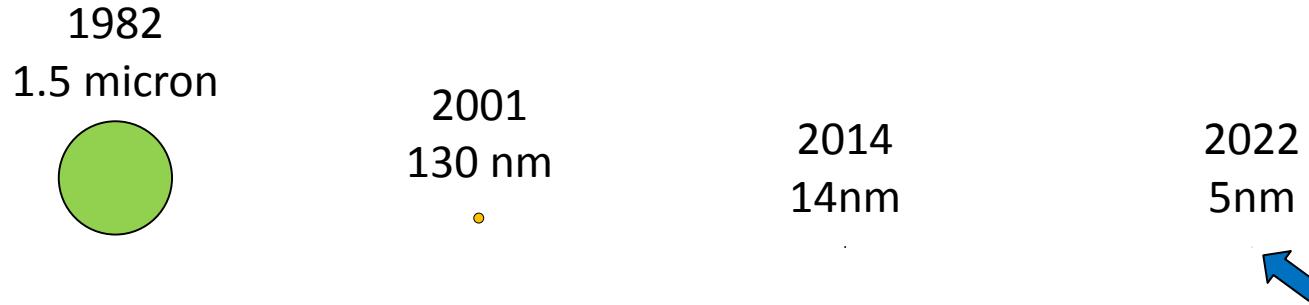
# How Did We Get Here?

# Transistor Technology

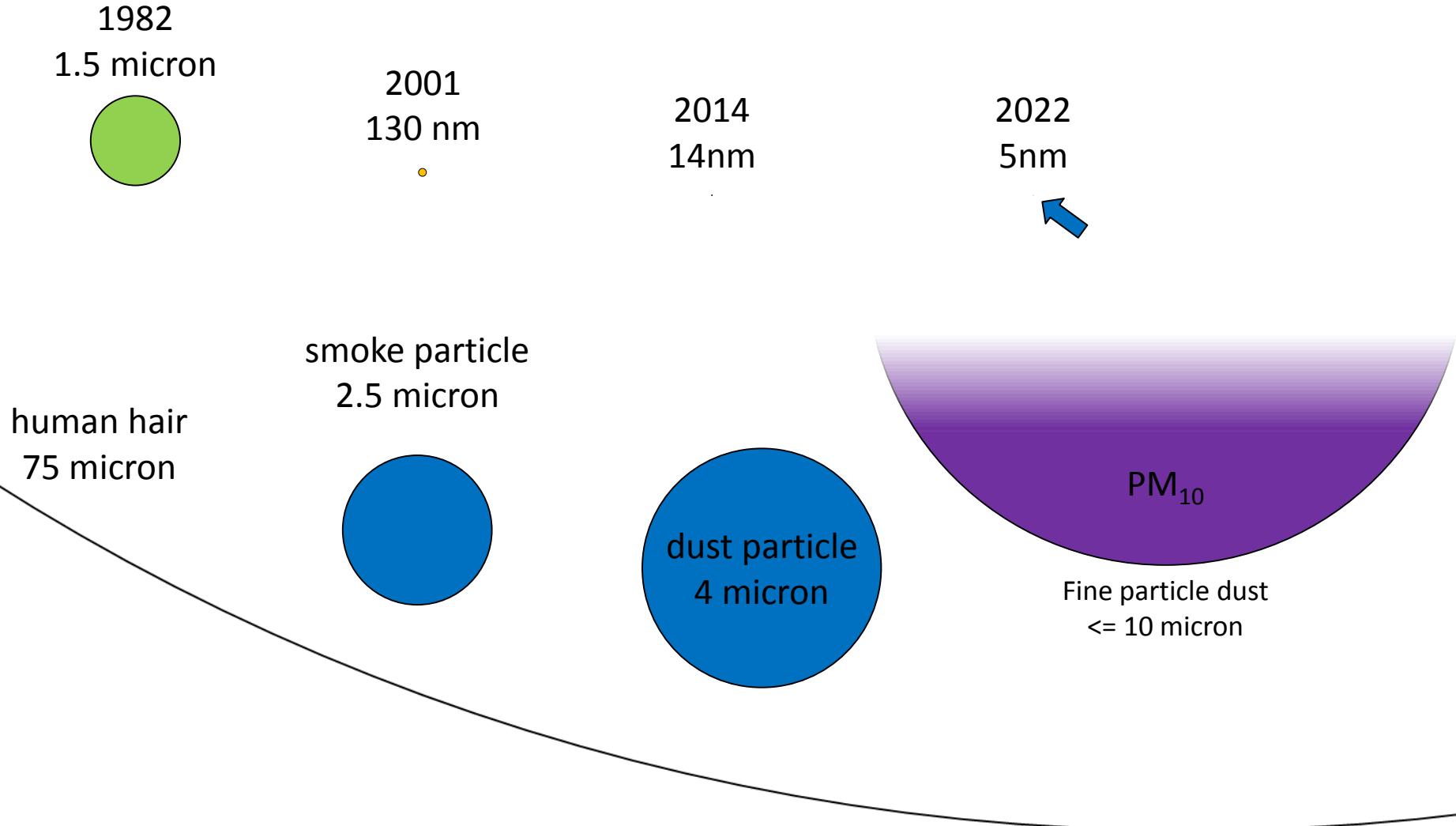
## From sand to circuits



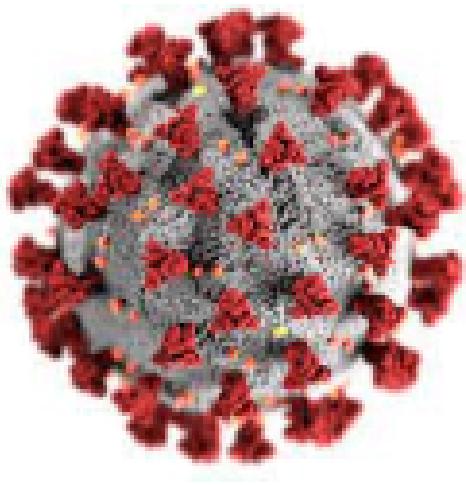
# Transistor Technology



# Transistor Technology – A Comparison



# Transistor Tech – A Contemporary Comparison



2020  
90nm  
(50-140 nm)



2014  
14nm



2022  
5nm

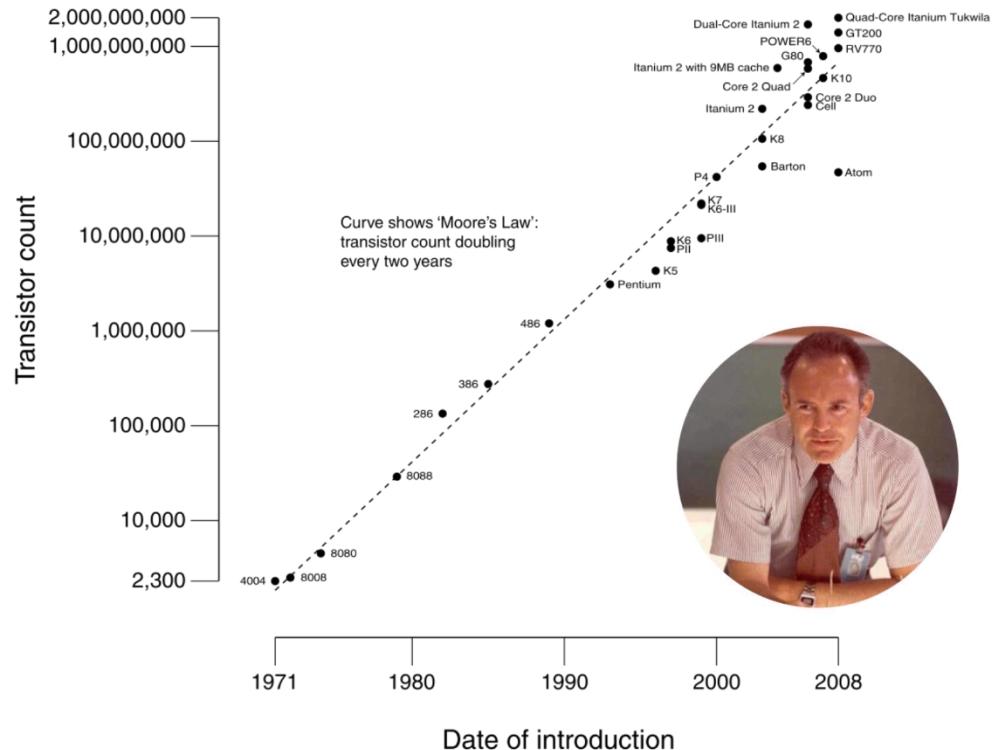
# Transistor Technology – Moore's Law

## ■ Gordon Moore, Intel Co-Founder, 1965

“The number of transistors incorporated into a chip will approximately double every 24 months.”

CPU Transistor Counts 1971-2008 & Moore's Law

## ■ Moore's Law held for 50 years!



# Transistor Technology - Yield

## ■ Intel Core 10<sup>th</sup> Generation (Ice Lake)

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

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

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

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

- cost of wafer fixed
- die area determined by architecture, circuit design, and technology
- defect rate determined by manufacturing process

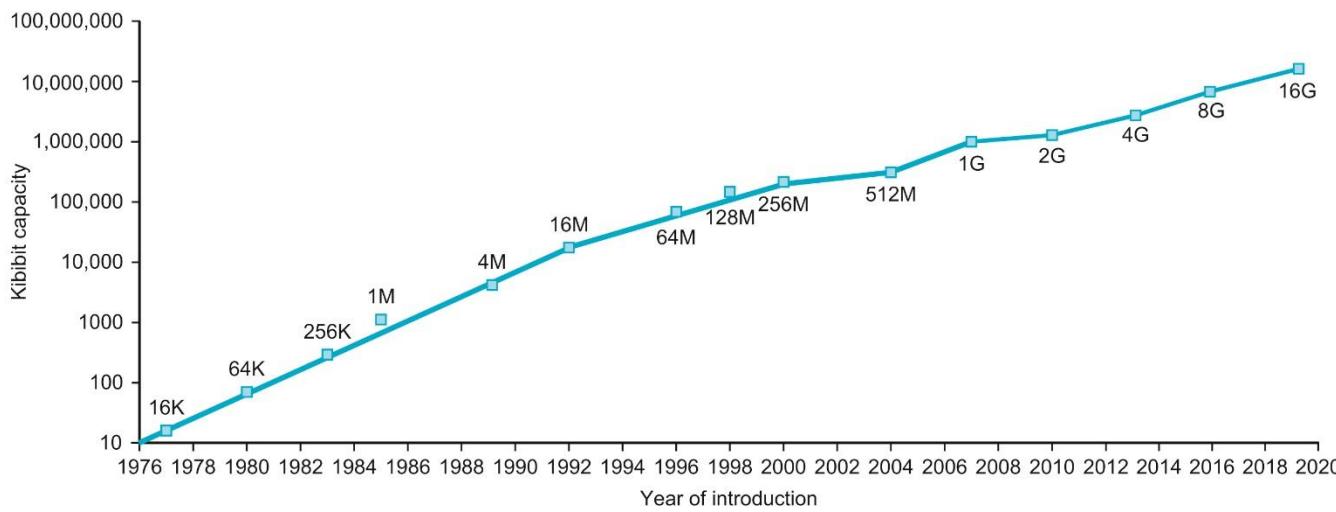
# Technology Trends

## ■ Processors

- Logic capacity: ~ +30% per year
- Clock rate: ~ +20% per year

## ■ Memory (DRAM)

- Capacity: ~ +60% per year (4x every three years)
- Speed: ~ +10% per year



# Dennard Scaling

## ■ MOFSET (Dennard) Scaling (valid until ~2005)

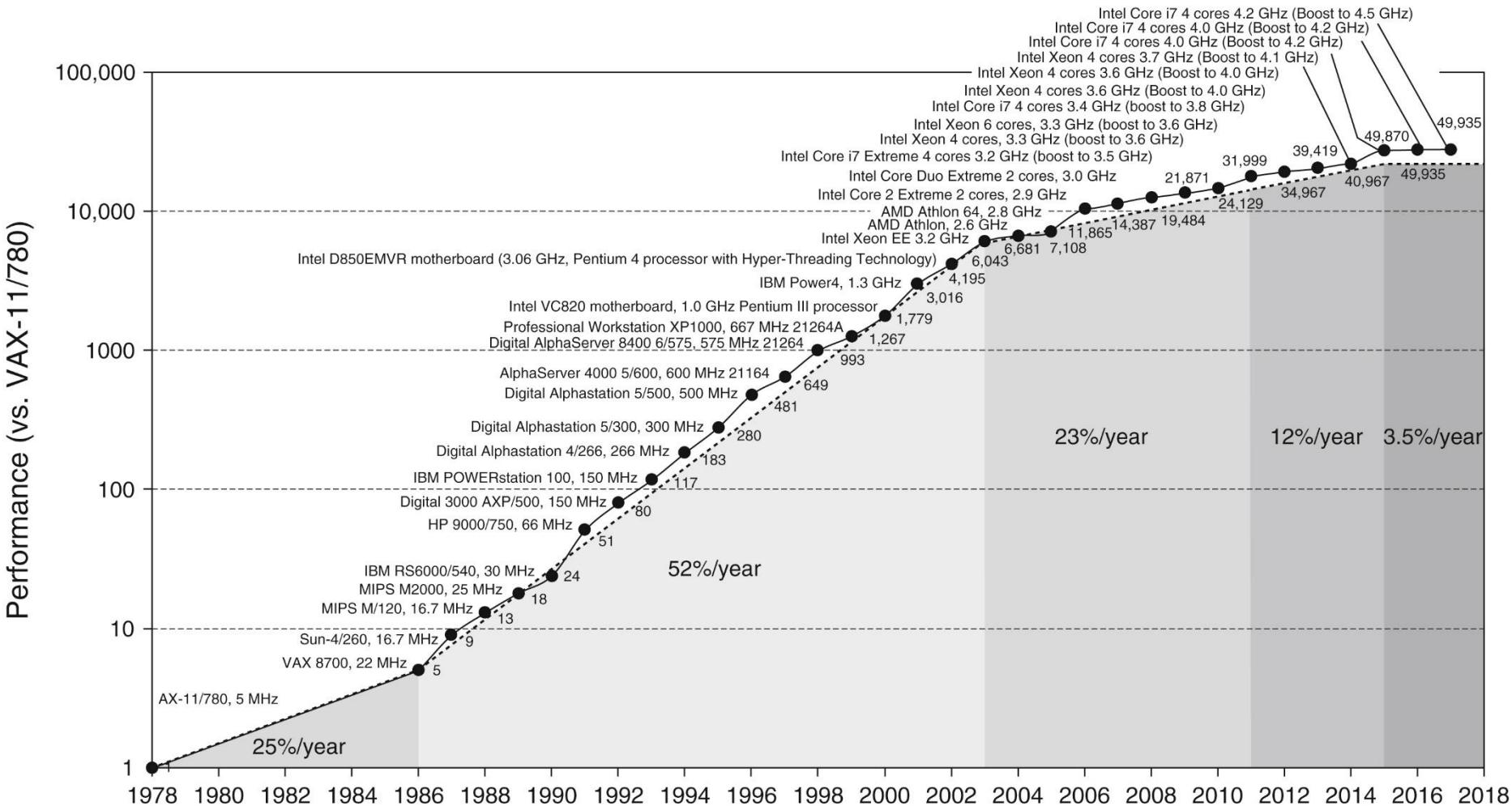
$$\text{Power} = \text{Capactive Load} \times \text{Voltage}^2 \times \text{Frequency}$$

- As transistors are scaled down, their power density stays constant
- Example:
  - ▶ transistors, capacitance, voltage: 0.7x
  - ▶ frequency: 1.4x

$$\frac{P_{new}}{P_{old}} = \frac{C_{old} \times 0.7 \times (V_{old} \times 0.7)^2 \times F_{old} \times 1.4}{C_{old} \times V_{old}^2 \times F_{old}} = 0.48$$

→ maintaining the same power consumption, we can double the number of transistors with faster performance!

# Uniprocessor Performance



# Everything was good...then we hit “three walls”

## ■ The ILP Wall (Instruction Level Parallelism)

There is only so much parallelism a processor can exploit

- control dependencies
- data dependencies

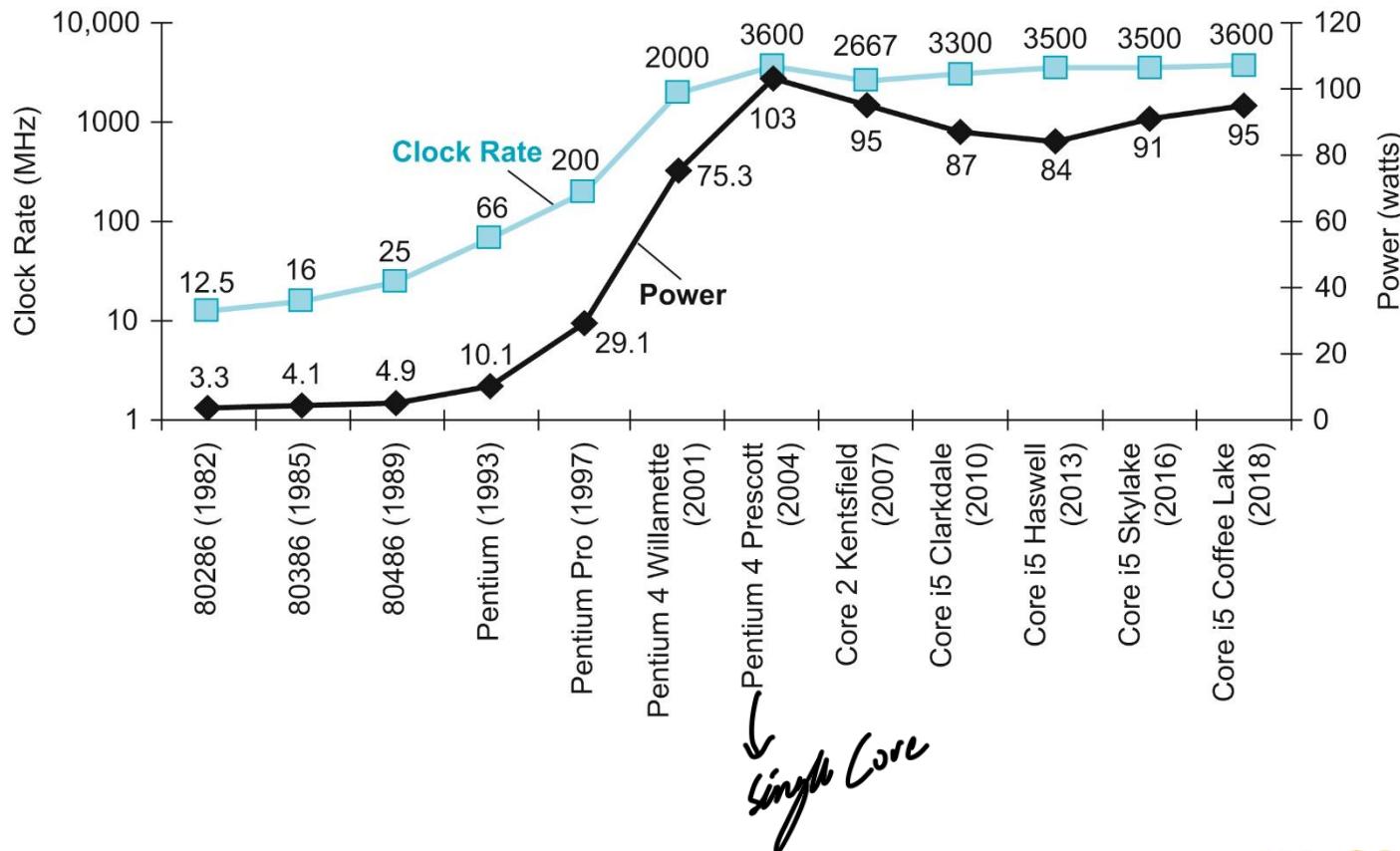
## ■ The Memory Wall

- Processor speedup > memory speedup, year after year
- Huge gap
- Caches show diminishing returns

## ■ The Power Wall (next slide)

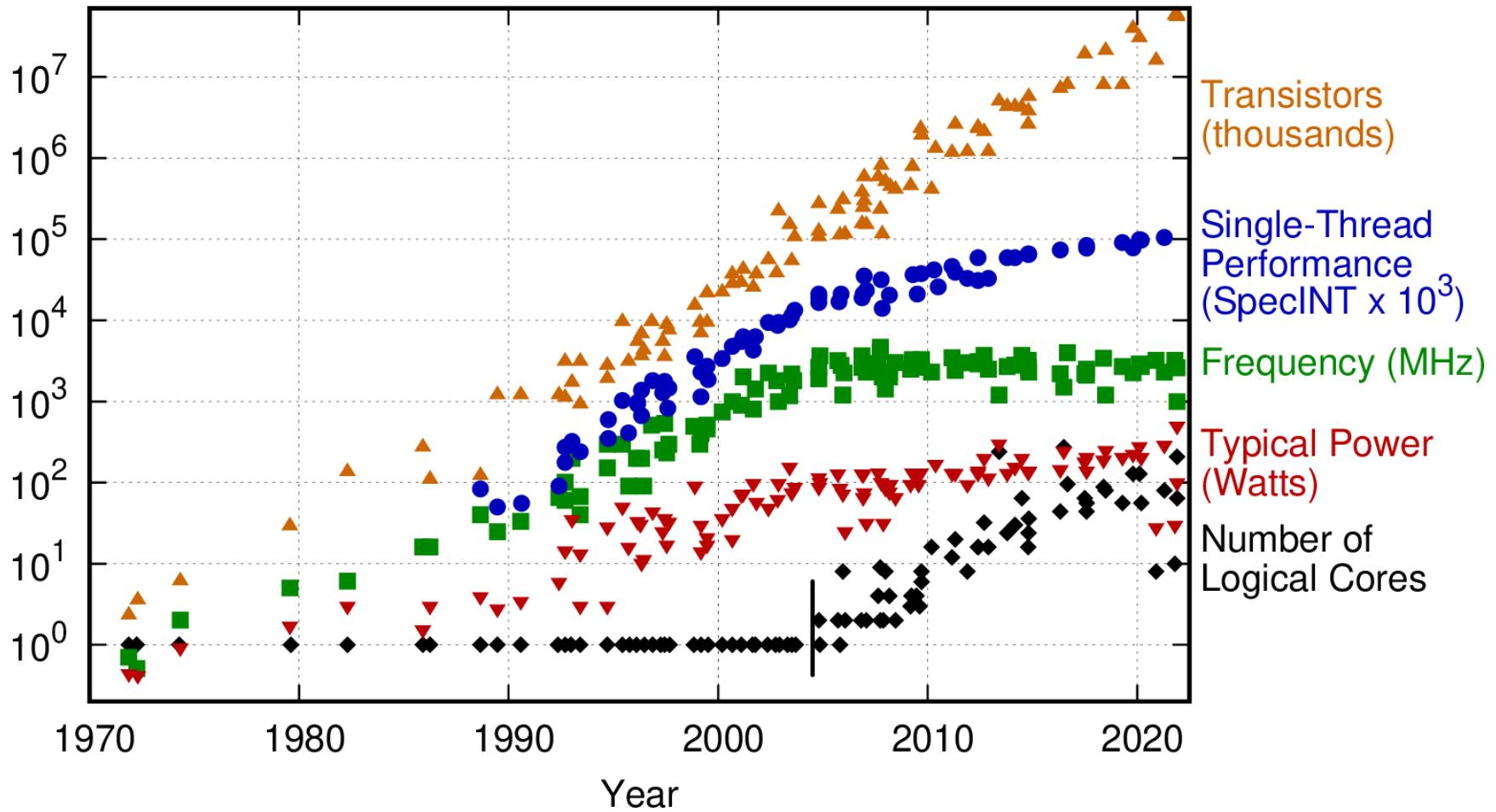
# The Power Wall

- Cannot reduce voltage anymore
- Cannot remove more heat



# From Single Cores to Multicores

50 Years of Microprocessor Trend Data



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten  
New plot and data collected for 2010-2021 by K. Rupp

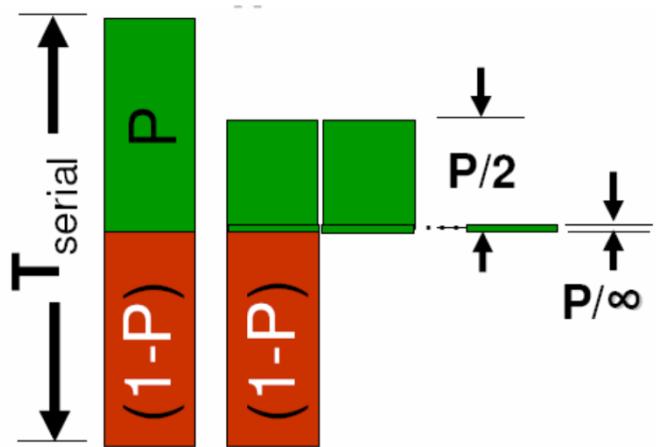
# Amdahl's Law

- Speedup is limited by the serial portion of a job

$$\text{Speedup} = \frac{1}{(1 - P) + \frac{P}{n}}$$

- P: parallelizable part
- S (= P-1): serial part
- n: number of cores

→ Corollary: make the common case fast!



# Where are we going today?

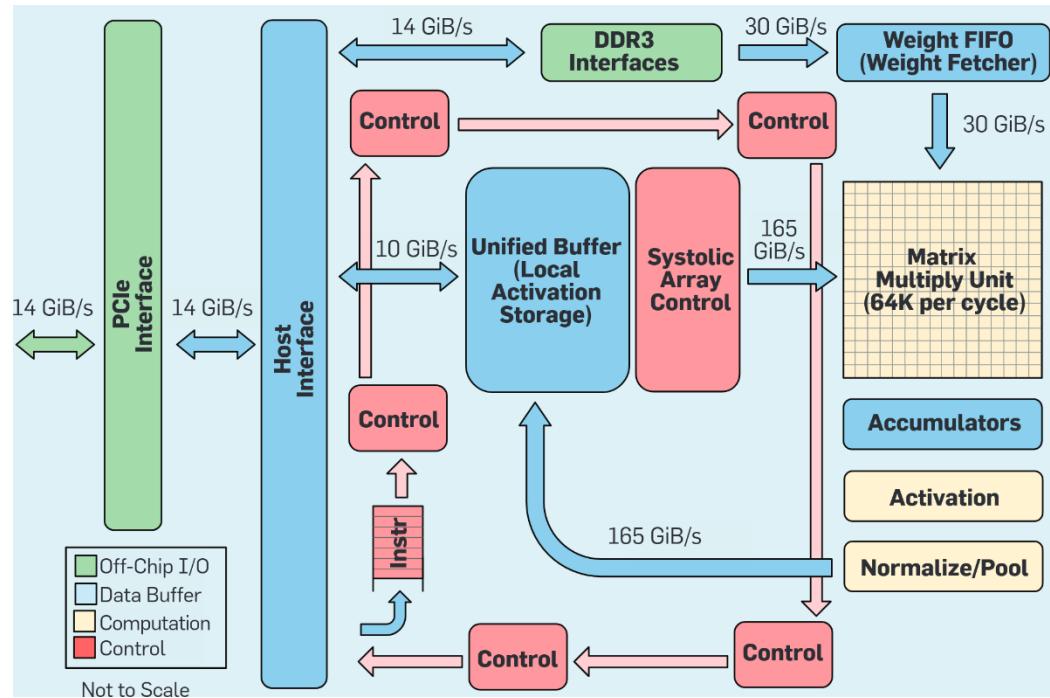
## ■ Specialized instructions for general-purpose processors

- specialized instructions
  - ▶ vector operations
  - ▶ FMA

## ■ Application-specific processors

- neural processing units
- Example: Google's TPU v1

## ■ Lots of interesting opportunities for computer architects, compiler builders, and system people!



Source: Hennessy and Patterson, "A New Golden Age for Computer Architecture", CACM, 2019.

# Where are we going today?

## NVIDIA

- GPUs (Turing, 2018)

Quadro RTX 6000

- 4608 CUDA cores
- 576 tensor cores
- 72 RT cores
- 1.5 ~ 1.8 GHz clock
- 18.6B transistors



- mobile CPUs



source: NVIDIA

# Where are we going today?

## ■ Intel

- Many Integrated Cores (“Xeon Phi”)

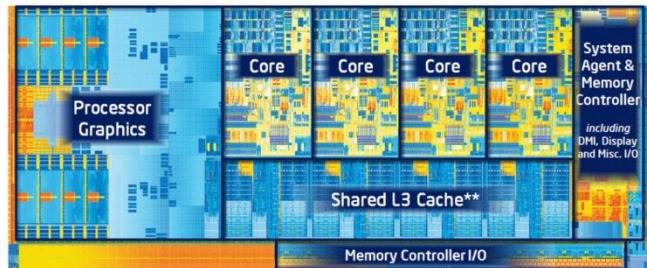
Knights Mill (Dec. 2017)

- 72 cores / 288 threads
- 36MB cache
- 1.5GHz clock

- Xeon Server CPUs

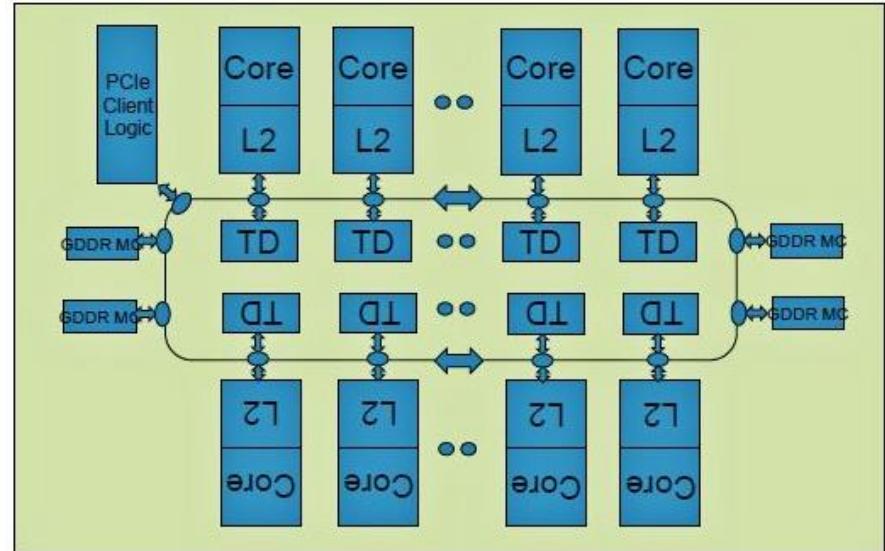
- up to 28 cores / 56 threads
- 38MB cache
- 2 ~ 3.5GHz clock

- Desktop/mobile CPUs



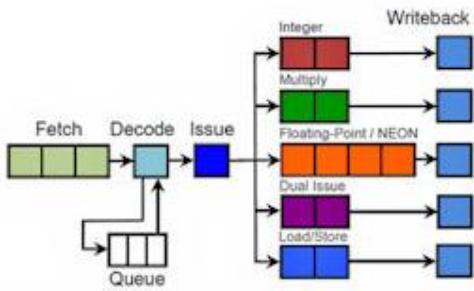
Whiskey Lake (Aug. 2018)

- 4 cores / 8 threads
- 8MB cache
- 1.8 – 4.6 GHz clock
- integrated graphics

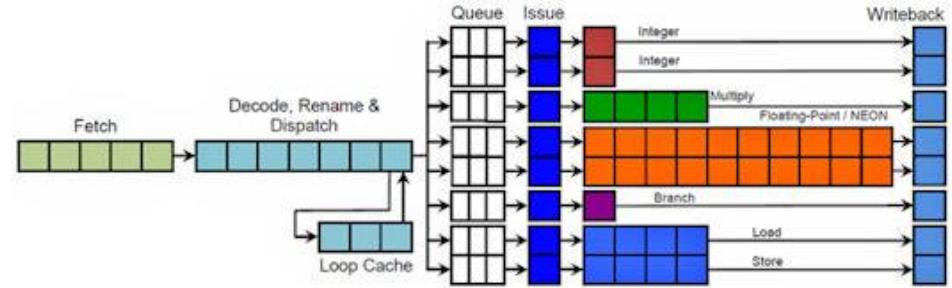


source: Intel

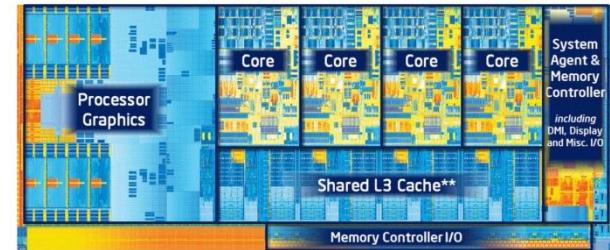
# This is Computer Architecture!



VS



VS



- Understanding the building blocks of processors and computer systems
- Understanding design tradeoffs such as performance vs efficiency
- Building the hardware
- Making it programmable

# Why You Should Care...

- Understanding computer architecture is important for many other core subjects in computer science (system programming, OS, compilers, programming models and languages, ...)
- Understanding computer architecture will make you a better programmer

```
for (i=0; i<N; i++) {  
    for (j=0; j<N; j++) {  
        C[i,j] = A[i,j] + B[i,j];  
    }  
}
```

vs

```
for (j=0; j<N; j++) {  
    for (i=0; i<N; i++) {  
        C[i,j] = A[i,j] + B[i,j];  
    }  
}
```

- Understanding assembly and a processor's ISA is still an important skill
- It is actually quite fun!



MOORE'S LAW



ABSTRACTION



COMMON CASE FAST



PARALLELISM



PIPELINING



PREDICTION



HIERARCHY



DEPENDABILITY

# Eight Great Ideas in Computer Architecture

# Great Ideas in Computer Architecture

## ■ Design for Moore's Law

- Anticipate state of technology when the product ships

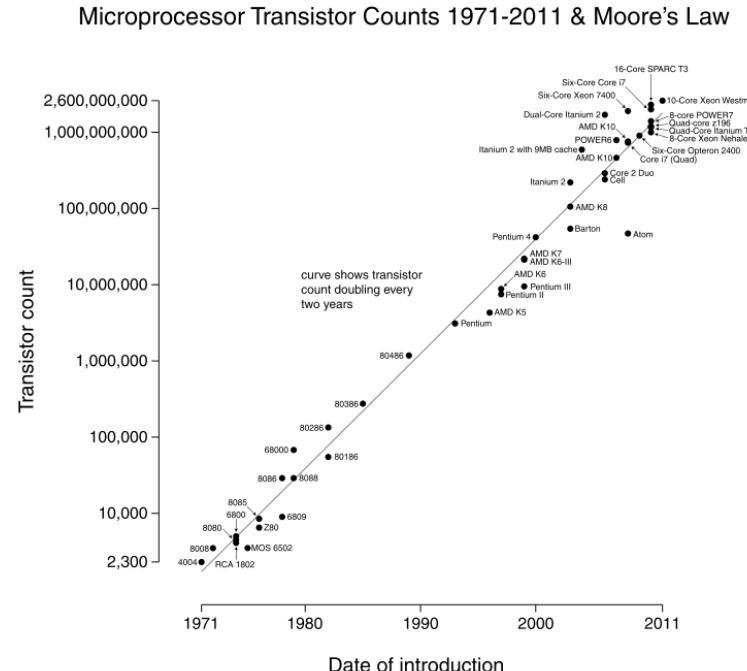
## ■ Use Abstraction to Simplify Design

- Abstractions are everywhere

## ■ Make the Common Case Fast

- Amdahl's Law

$$speedup = \frac{1}{(1 - parallel) + \frac{parallel}{\#cores}}$$



source: Wikipedia

# Great Ideas in Computer Architecture

## ■ Performance via Parallelism

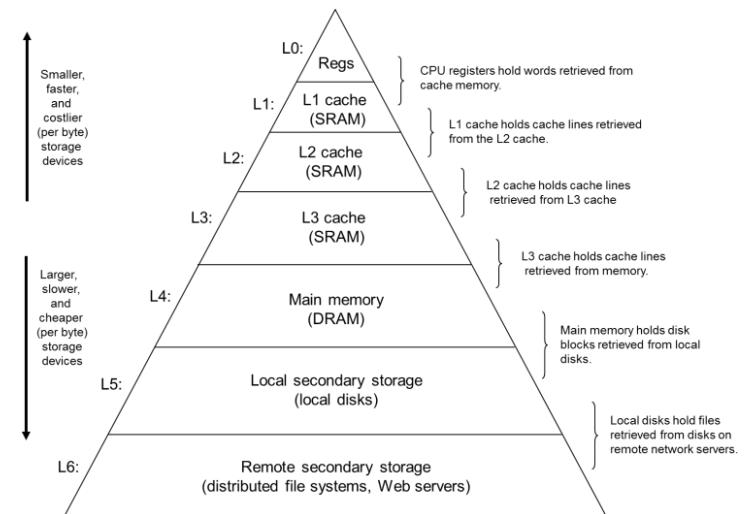
## ■ Performance via Pipelining

## ■ Performance via Prediction

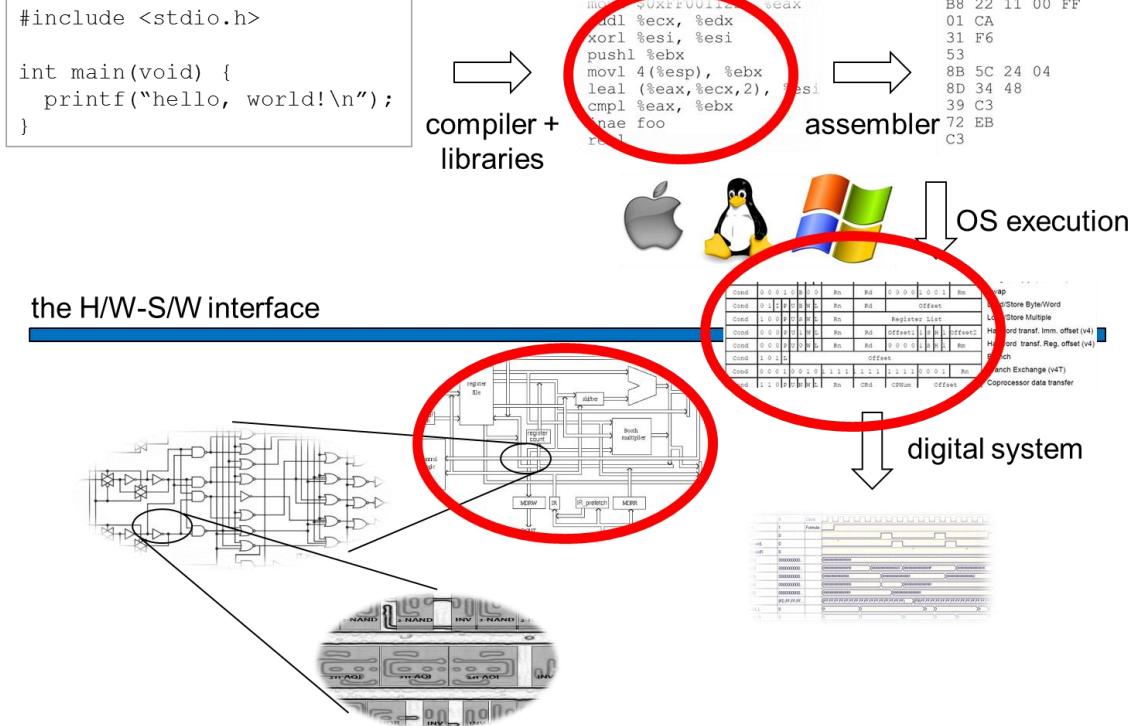
## ■ Hierarchy of Memories

- Exploit spatial and temporal locality  
(공간적/시간적 지역성)

## ■ Dependability via Redundancy



# Course Outlook



# What Are We Going To Do In This Course?

```
#include <stdio.h>  
  
int main(void) {  
    printf("hello, world!\n");  
}
```

compiler +  
libraries

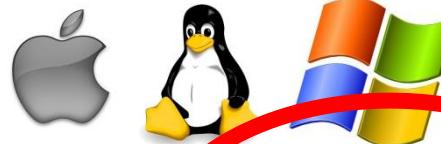
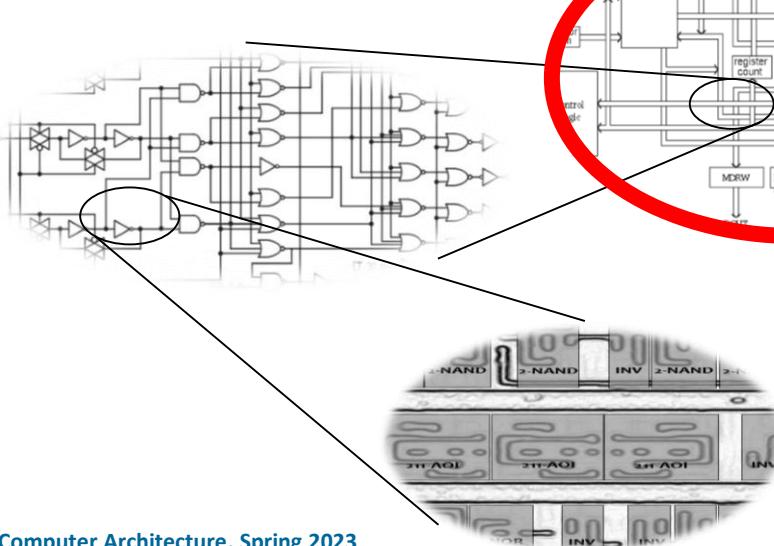
```
movl $0xFF001122, %eax  
addl %ecx, %edx  
xorl %esi, %esi  
pushl %ebx  
movl 4(%esp), %ebx  
leal (%eax,%ecx,2), %esi  
cmpl %eax, %ebx  
jae foo  
retl
```

B8 22 11 00 FF  
01 CA  
31 F6  
53  
8B 5C 24 04  
8D 34 48  
39 C3  
72 EB  
C3

assembler

# this course

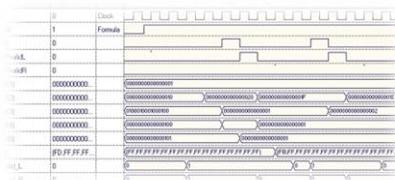
the H/W-S/W interface



OS execution

|      |           |         |         |         |         |                 |    |                                    |
|------|-----------|---------|---------|---------|---------|-----------------|----|------------------------------------|
| Cond | 0 0 0 1 0 | B       | 0 0     | Rn      | Rd      | 0 0 0 0 1 0 0 1 | Rm | swap                               |
| Cond | 0 1 -     | P       | U       | W       | L       | Rn              | Rd | End/Store Byte/Word                |
| Cond | 1 0 0     | P       | U       | S       | W       | L               | Rn | Loc/Store Multiple                 |
| Cond | 0 0 0     | P       | U       | 1       | W       | L               | Rn | Half-word transf. Imm. offset (v4) |
| Cond | 0 0 0     | P       | U       | 0       | W       | L               | Rn | Half-word transf. Reg. offset (v4) |
| Cond | 1 0 1     | L       | -       | -       | -       | -               | Rm | Branch                             |
| Cond | 0 0 0 1   | 0 0 1 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 0 0 0 1         | Rn | Branch Exchange (v4T)              |
| Cond | 1 1 0     | P       | U       | N       | W       | C               | Rn | Coprocessor data transfer          |

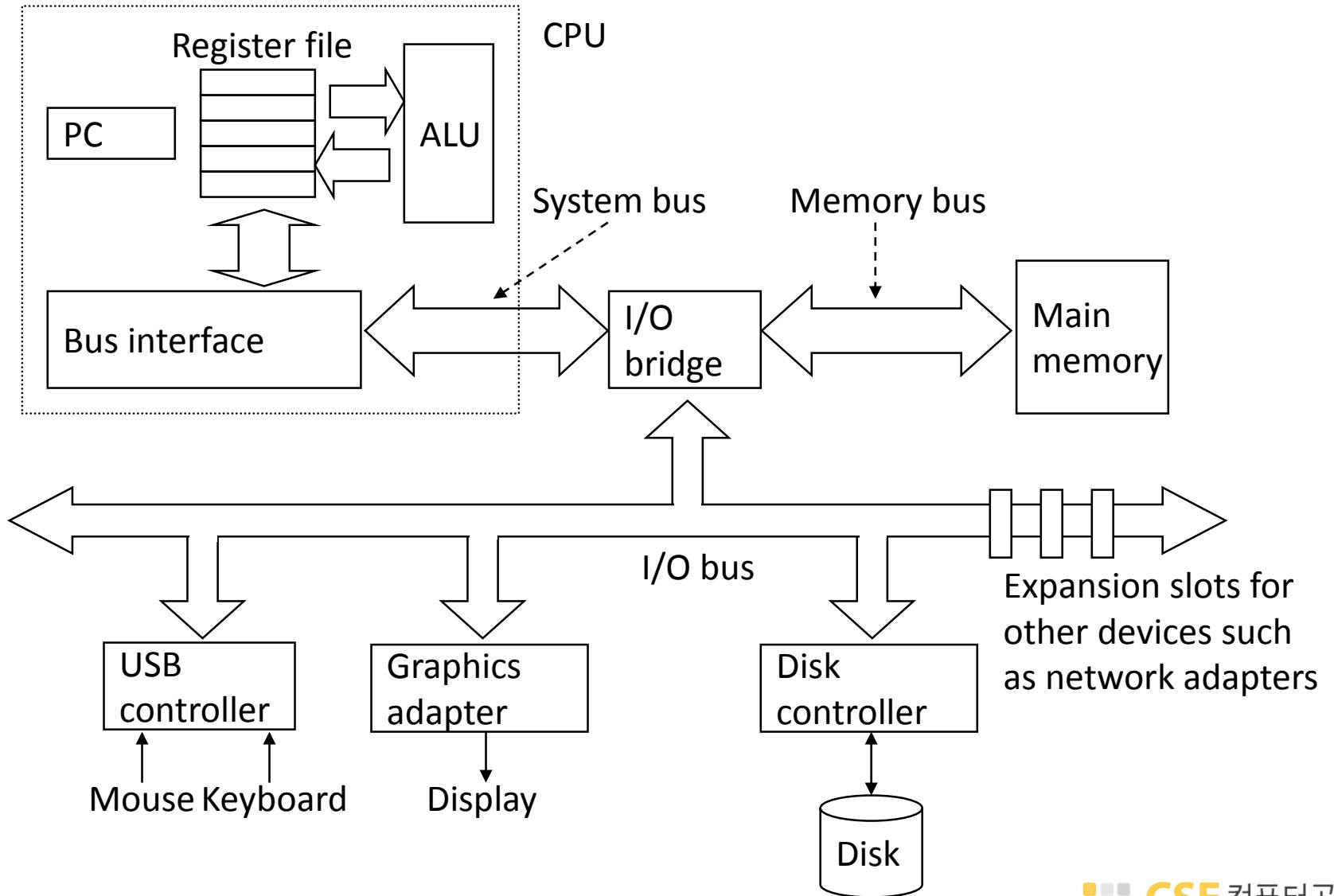
digital system



# What Are We Going To Do In This Course?

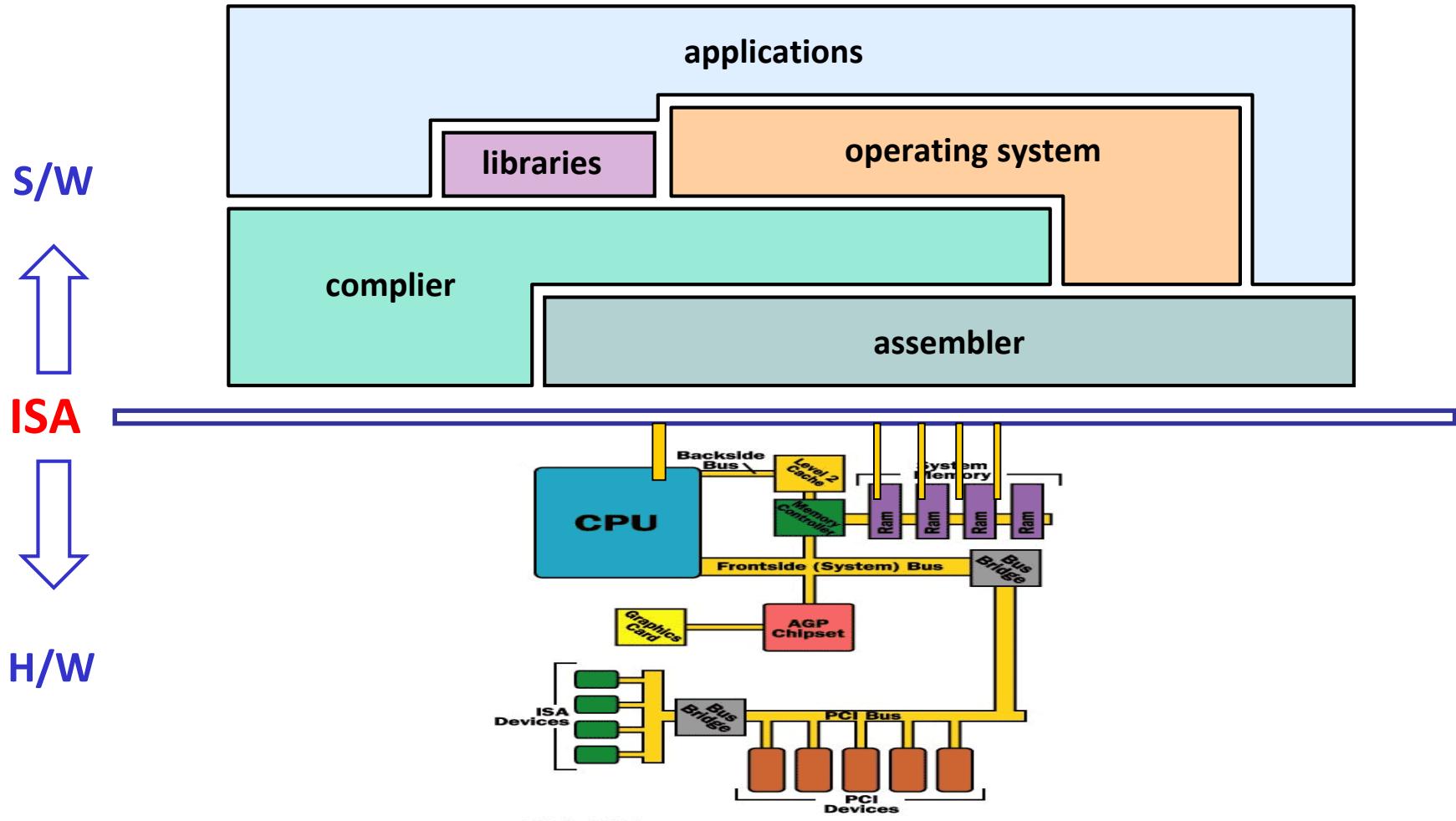
- We will *not*
  - design the next Core i11 with a 64-stage pipeline
  - study how logic gates work (you should know that already)
  - write entire programs in assembly
  
- We will, however,
  - learn an ISA (RISC-V) and how to read and understand assembly programs
  - learn how a (simple) pipelined processor is built
  - learn about the memory hierarchy in modern computer systems
  - have a quick look at modern state-of-the-art processors

# Part I: The Computer in A Nutshell



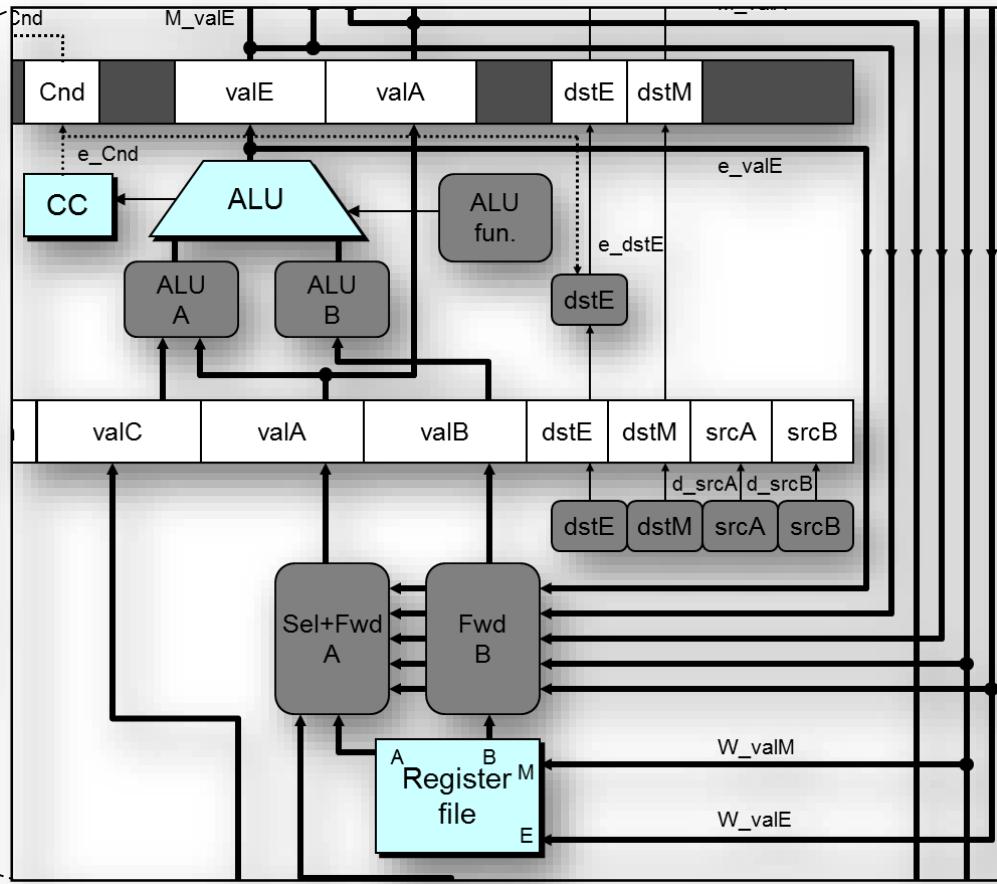
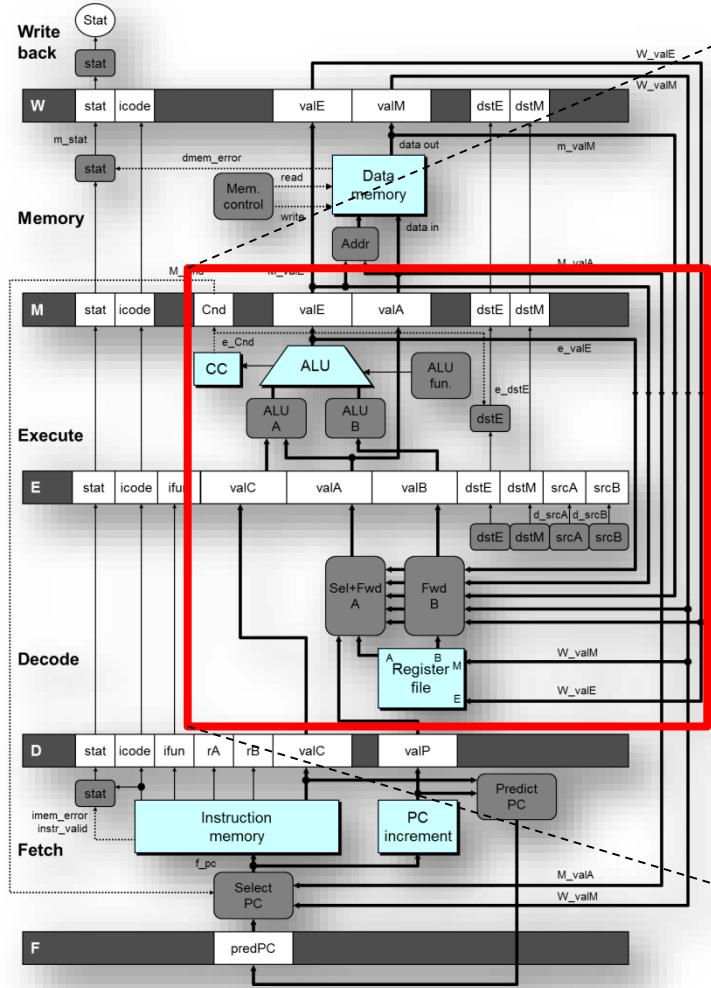
# Part II: The Instruction Set Architecture

- The ISA (Instruction Set Architecture) as the hardware/software interface



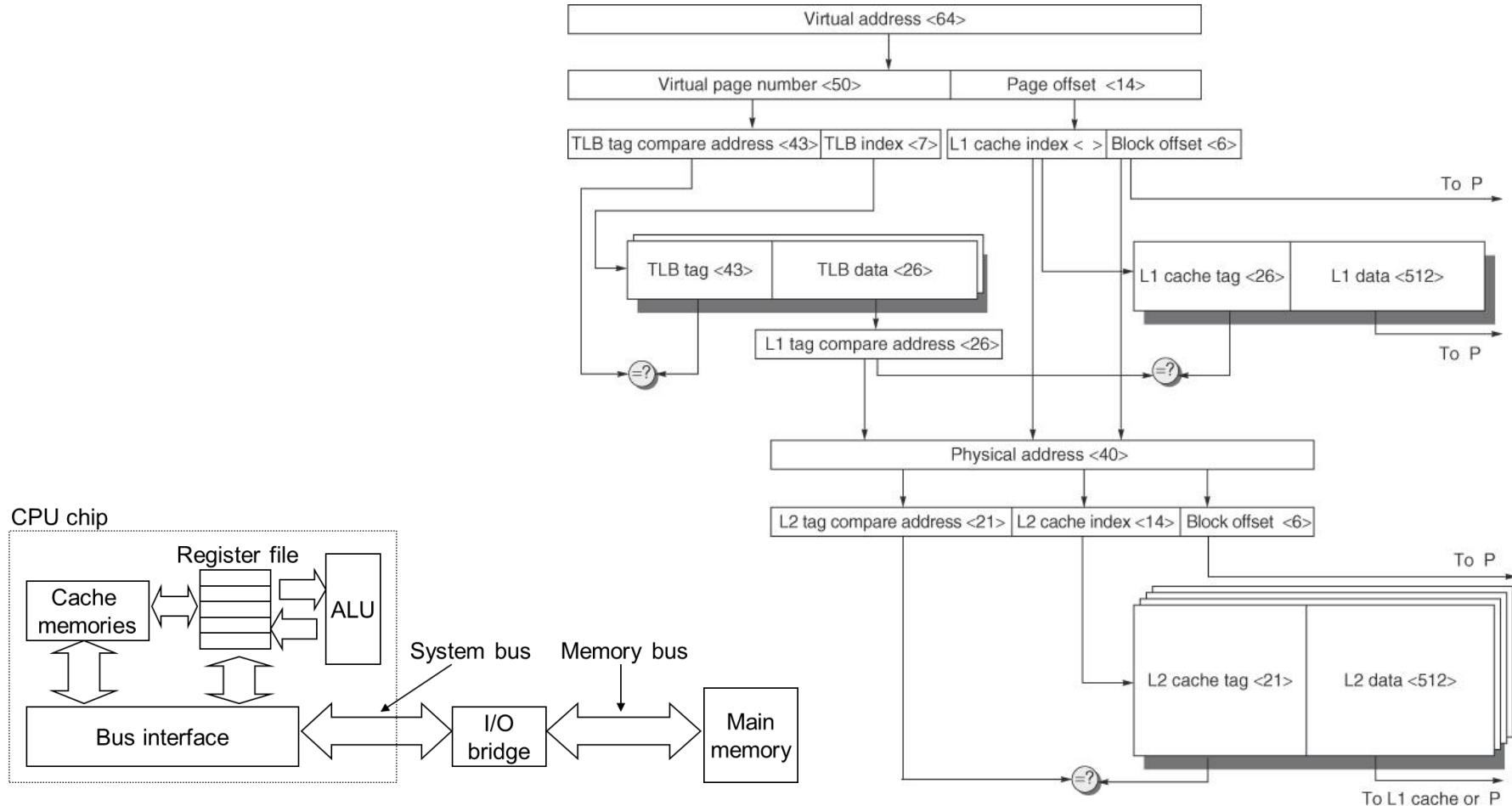
# Part III: Processor Architecture

## Study and modify a simple pipelined processor



# Part IV: The Storage Hierarchy

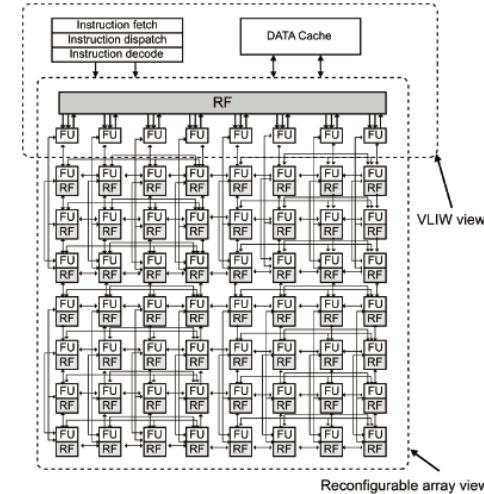
## Study the memory hierarchy in modern computer systems



# Part V: Modern Processor Architectures

## ■ What other (modern) architectures are out there?

### NPUs and GPUs



VLIW and CGRA  
architectures

## Summary

- -----
- -----
- -----
- -----
- -----
- -----

# Module Summary

# Summary

- Modern Computer Architecture is about managing and optimizing across several levels of abstraction w.r.t. dramatically changing technology and application load
- This course focuses on
  - RISC-V Instruction Set Architecture (ISA) – a new open interface
  - An implementation based on Pipelining (Microarchitecture) – how to make it faster?
  - Memory hierarchy – how to make trade-offs between performance and cost?
- Understanding Computer Architecture is vital to other “systems” courses:
  - System programming, Operating systems, Compilers, Embedded systems, Computer networks, Multicore computing, Distributed systems, Mobile computing, Security, Machine learning, Quantum computing, etc.