

CSC 411

Computer Organization (Spring 2024)
Lecture 9: Computer systems

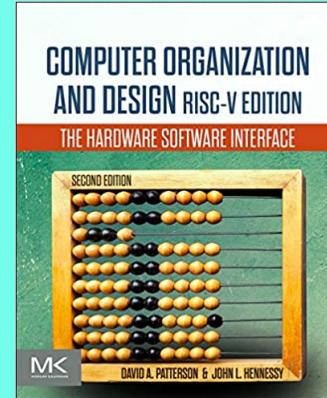
Prof. Marco Alvarez, University of Rhode Island

Units of measure

Disclaimer

Some figures and slides are adapted from:

Computer Organization and Design (Patterson and Hennessy)
The Hardware/Software Interface



SI prefixes

Factor	Name	Symbol	Factor	Name	Symbol
10^1	deca	da	10^{-1}	deci	d
10^2	hecto	h	10^{-2}	centi	c
10^3	kilo	k	10^{-3}	milli	m
10^6	mega	M	10^{-6}	micro	μ
10^9	giga	G	10^{-9}	nano	n
10^{12}	tera	T	10^{-12}	pico	p
10^{15}	peta	P	10^{-15}	femto	f
10^{18}	exa	E	10^{-18}	atto	a
10^{21}	zetta	Z	10^{-21}	zepto	z
10^{24}	yotta	Y	10^{-24}	yocto	y
10^{27}	ronna	R	10^{-27}	ronto	r
10^{30}	quette	Q	10^{-30}	quecto	q

SI prefixes are a set of 24 prefixes used in the International System of Units (SI) to indicate multiples and submultiples of SI units. They are based on powers of 10, and each prefix has a unique symbol.

Binary prefixes

The SI prefixes refer strictly to powers of 10. They should not be used to indicate powers of 2 (for example, one kilobit represents 1000 bits and not 1024 bits). The names and symbols for prefixes to be used with powers of 2 are recommended as follows:

kibi	Ki	2^{10}
mebi	Mi	2^{20}
gibi	Gi	2^{30}
tebi	Ti	2^{40}
pebi	Pi	2^{50}
exbi	Ei	2^{60}
zebi	Zi	2^{70}
yobi	Yi	2^{80}

Computer systems

The computer revolution

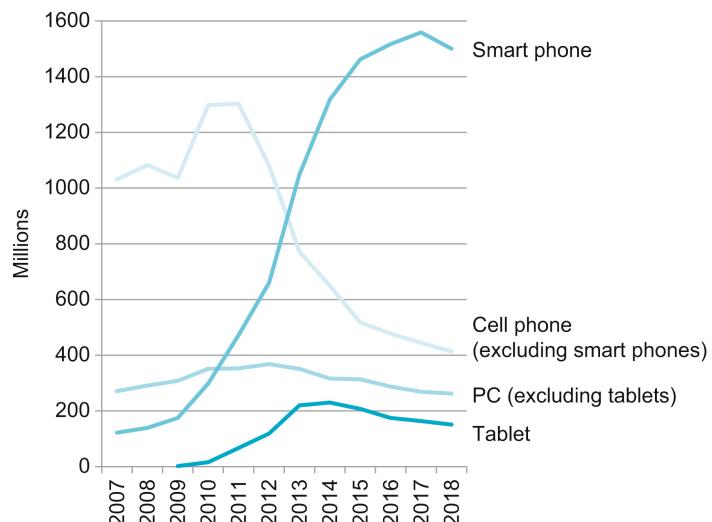
- Progress in computer technology
 - underpinned by domain-specific accelerators
- Novel applications (not long ago considered fiction):
 - generative AI, computers in automobiles, smartphones, human genome project, web, search engines
- Although common hardware technologies ...
 - different design requirements

Computers in all forms are now pervasive

Classes of computers

- Personal computers (desktops/laptops)
 - low cost, general purpose, variety of software
 - subject to cost/performance tradeoff
- Server computers
 - network based, large workloads
 - high capacity, performance, reliability
 - range from small servers to building sized supercomputers (high-end scientific and engineering calculations)
- Embedded computers
 - largest class
 - hidden as components of systems (Internet of Things)
 - stringent power/performance/cost constraints

The post PC era (# of manufactured devices)



The post PC era

- Personal Mobile Device (PMD) — taking over PCs
 - battery operated
 - connects to the Internet
 - hundreds of dollars
 - smart phones, tablets, electronic glasses
- Cloud computing — taking over servers
 - Warehouse Scale Computers (WSC) — e.g. Amazon AWS
 - Software as a Service (SaaS)
 - portion of software run on a PMD and a portion run in the Cloud
 - Amazon and Google

Accelerators

- Multiple cores in one chip
 - heterogeneous collection of cores
 - System-on-a-Chip (SoC) design
- Graphics processing units (GPUs)
 - throughput-oriented multicore processors
 - great for gaming and machine learning
- Focus on performance and energy efficiency

High performance computing

Top 500 project

- 500 most powerful computers in the world
- Updated twice a year
 - ISC'xy in June, Germany and SC'xy in November, U.S.

NOVEMBER 2023		Manufacturer	Computer	Country	Cores	Rmax [petaflop]	Power [MW]
1	Oak Ridge National Laboratory	HPE	Frontier HPE Cray EX235a, AMD EPYC 64C 2.0GHz, Instinct MI250X, Slingshot-11	USA	8,730,112	1,102	21.1
2	Argonne National Laboratory	HPE	HPE Cray EX Xeon CPU Max 9470 52C 2.4GHz, Intel Data Center GPU Max, Slingshot-11	USA	4,742,808	585.3	24.6
3	Microsoft Azure	Microsoft	Foale Microsoft NDv5 Xeon Platinum 8480C 48C 2.0GHz, NVIDIA H100, NVIDIA Infiniband NDR	USA	1,123,200	561.2	
4	RIKEN Center for Computational Science	Fujitsu	Fugaku Supercomputer Fugaku, A64FX 48C 2.0GHz, Tofu interconnect D	Japan	7,630,848	442.0	29.9
5	EuroHPC / CSC	HPE	LUMI HPE Cray EX235a, AMD EPYC 64C 2.0GHz, Instinct MI250X, Slingshot-11	Finland	2,069,760	309.1	6.0

credit: Berkeley's CS267 lectures

Top computer in the world



Frontier (#1) System Overview

System Performance

- Peak performance of 1.6 double precision exaFLOPS
- Measured Top500 performance (Rmax) was 1.102 exaFLOPS

Each node has

- 3rd Gen AMD EPYC CPU with 64 cores
- 4 Purpose Built AMD Instinct 250X GPUs
- 4X128 GB of fast memory, 1 per GPU
- 5 terabytes of flash memory



The system includes

- 9,472 nodes
- Slingshot interconnect



credit: Berkeley's CS267 lectures

Exascale applications at LBNL

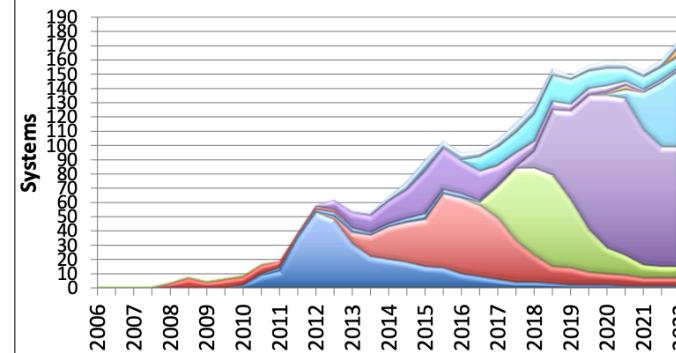


credit: Berkeley's CS267 lectures

Heterogenous devices

TOP 500

Nvidia Fermi	Nvidia Kepler	Nvidia Pascal	Nvidia Volta	Nvidia Ampere
Nvidia Turing	ATI Radeon	IBM Cell	Clearspeed	Intel Xeon Phi
Xeon Phi Main	AMD Instinct	Others		



2022: Over 1/3rd of all systems have accelerators or co-processors

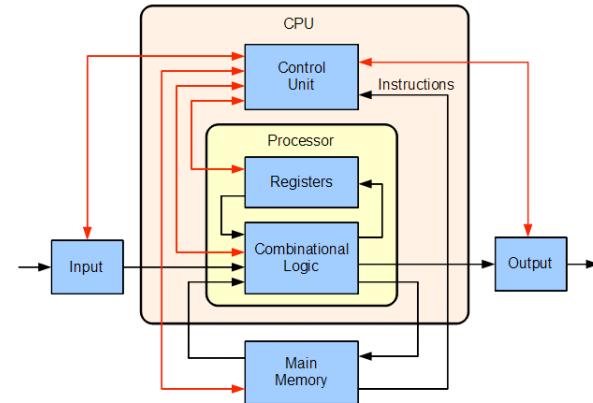
The actual "performance share" is higher

credit: Berkeley's CS267 lectures

Abstractions

The Von-Neumann model

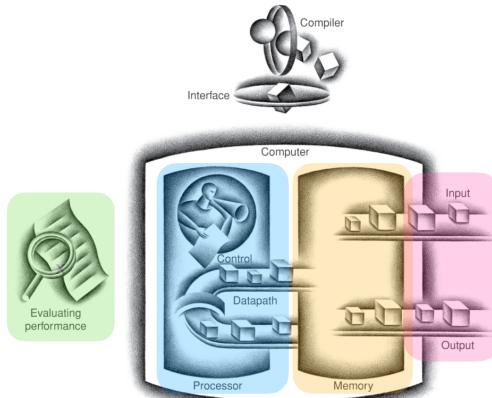
- High-level organization of computer hardware



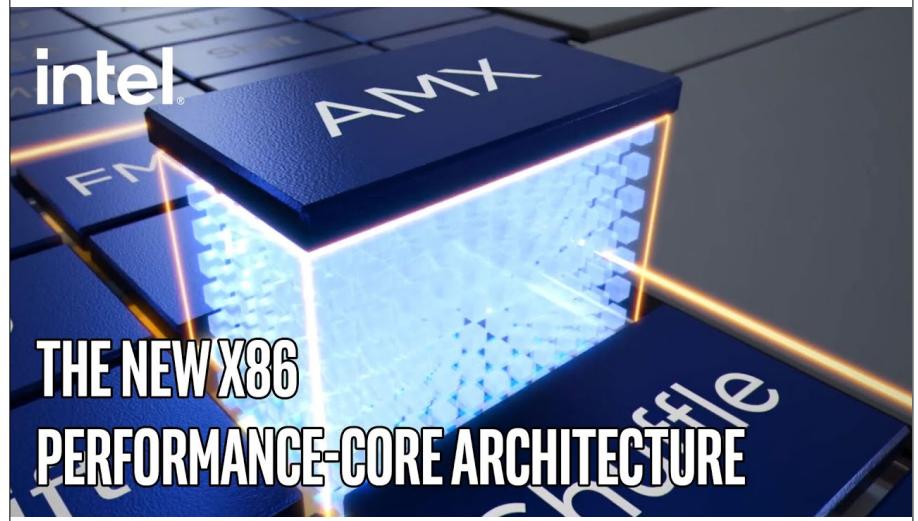
https://en.wikipedia.org/wiki/Computer_architecture

Components

- Same components for all kinds of computers
 - 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



Intel's x86 core



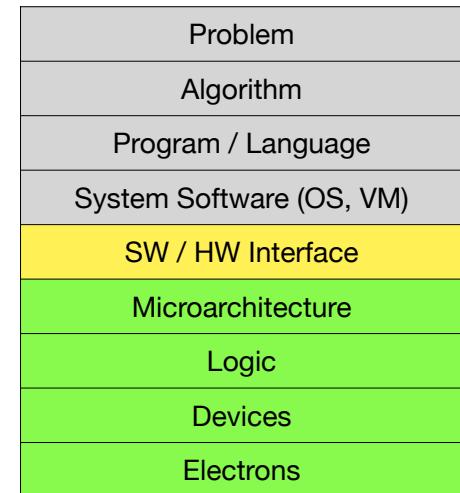
<https://www.youtube.com/watch?v=ijTRvIQV7bE>

Seven 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

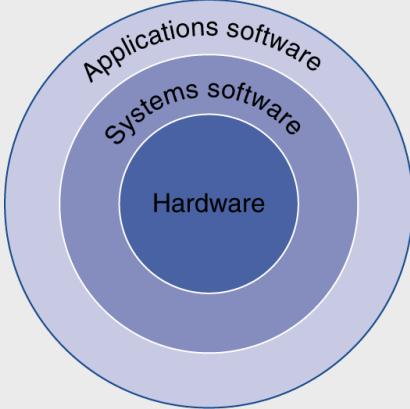


Abstraction layers (computing system)



Below your program

- Application software
 - written in high-level language
 - System software
 - compiler: translates HLL code to machine code
 - operating system: service code
 - handling input/output
 - managing memory and storage
 - scheduling tasks & sharing resources
 - Hardware
 - processor, memory, I/O controllers



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

High-level
language
program
(in C)

```
swap(int v[], int k)
{int temp;
    temp = v[k];
    v[k] = v[k+1];
    v[k+1] = temp;
}
```

Assembly
language
program
(for MIPS)

```
swap:  
    mul $2, $5,4  
    add $2, $4,$2  
    lw $15, 0($2)  
    lw $16, 4($2)  
    sw $16, 0($2)  
    sw $15, 4($2)  
    jr $31
```

Binary machine language program
(for MIPS)

The screenshot shows the Compiler Explorer interface. On the left, there's a sidebar with a gear icon labeled 'COMPILER EXPLORER' and buttons for 'Add...', 'More...', 'C++ source #1', and '(2, 1)'. The main area has tabs for 'A', 'Output...', 'Filter...', 'Libraries', and '+ Add n'. The code editor contains the following C++ code:

```

1 #include <stdio.h>
2
3 int main() {
4     printf("hello world !\n");
5
6     return 0;
7 }

```

Below the code, the assembly output is shown:

```

1 main:
2     addi    sp, sp, -32
3     sd     ra, 24(sp)
4     sd     s0, 16(sp)
5     addi   s0, sp, 32
6     li     a0, 0
7     sd     a0, -32(s0)
8     sw     a0, -20(s0)
9     lui    a0, %hi(.L.str)
10    addi   a0, a0, %lo(.L.str)
11    call   printf
12    ld     a0, -32(s0)
13    ld     ra, 24(sp)
14    ld     s0, 16(sp)
15    addi   sp, sp, 32
16    ret
17 .L.str:
18     .asciz "hello world !\n"

```

At the bottom, there's a link: <https://godbolt.org/>

Instruction set architecture

“the words of a computer’s language are called *instructions*, and its vocabulary is called an *instruction set*” [Computer Organization and Design, P&H]

the basic job of a CPU is to execute instructions

Looking into an executable

```

#include <stdio.h>

int main() {
    printf("hello world !\n");

    return 0;
}

```

\$ objdump -d

Definitions

- **Instruction set**
 - the **repertoire** of instructions of a computer
 - different computers have different instruction sets
 - ... with many aspects in common
- **Instruction set architecture (HW/SW interface)**
 - parts of a processor design that one needs to understand for writing correct machine/assembly code
 - a **contract** between “hw” and “sw”, defining format, syntax, and semantics
 - e.g., instructions set, registers
- **Microarchitecture**
 - implementation of the architecture, i.e., how the processor physically executes the instructions
 - e.g., cache sizes, core frequency, pipelining, branch prediction

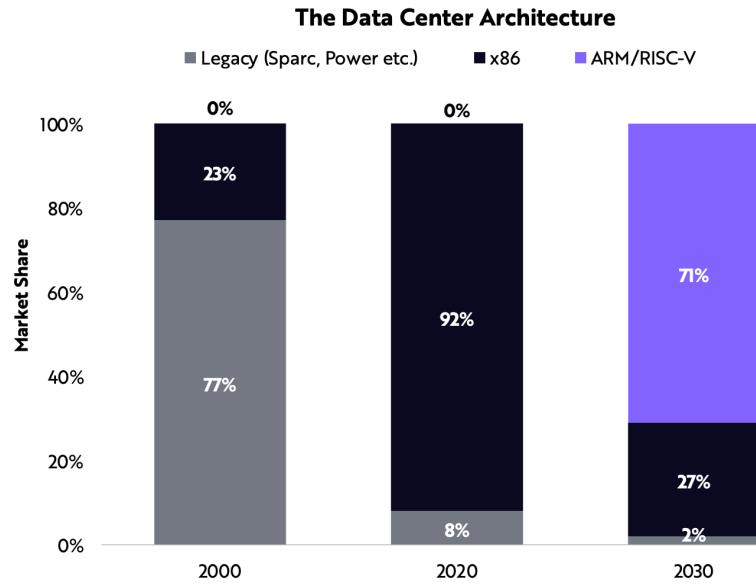
Major ISAs

‣ Major ISAs

- X86 (Intel and AMD)
 - desktop computers, laptops, servers
- ARM
 - widely used on embedded, phones, recently on laptops, servers
- RISC-V
 - fastest growing, embedded systems, servers, personal computers
 - “one ISA rules them all” => focus on optimizing the implementation

‣ Even more microarchitectures

- Apple/Samsung/Qualcomm have their own microarchitecture (implementation) of ARM
- Intel/AMD have different microarchitectures for x86



ISA design principles

‣ Keep hardware simple

- chip must only implement **basic primitives** and **run fast**
- simplicity enables higher performance at lower cost

‣ Keep the instructions regular

- **regularity** makes implementation simpler
 - simplifies decoding/scheduling of instructions

CISC vs RISC

‣ Design “philosophies” for ISAs

- RISC vs. CISC

‣ Complex Instruction Set Computer (CISC)

- X86, X86_64 (Intel and AMD, desktop/laptop/server)
- X86* internally are still RISC

‣ Reduced Instruction Set Computer (RISC)

- ARM (smartphone/pad)
- **RISC-V** (free ISA, closer to MIPS than other ISAs)
- Others: Power ISA, SPARC, etc

CISC vs RISC

› CISC

- early trend was to add more and more instructions
- VAX architecture had an instruction to multiply polynomials!
- operations may directly manipulate memory

› RISC philosophy

- keep the instruction set **small and simple**
- makes it easier to build fast hardware.
- let software do complicated operations by composing simpler ones



ACM has named John L. Hennessy, former President of Stanford University, and David A. Patterson, retired Professor of the University of California, Berkeley, recipients of the **2017 ACM A.M. Turing Award** for pioneering a systematic, quantitative approach to the design and evaluation of computer architectures with enduring impact on the microprocessor industry.

<https://www.youtube.com/watch?v=3LVeEjsn8Ts>

Chip Manufacturing

Semiconductor technology

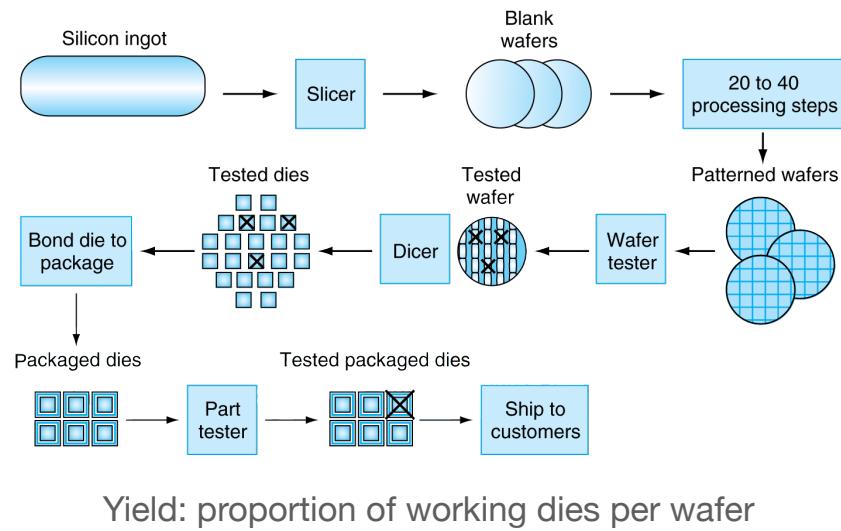
› Silicon



› Add materials to transform properties

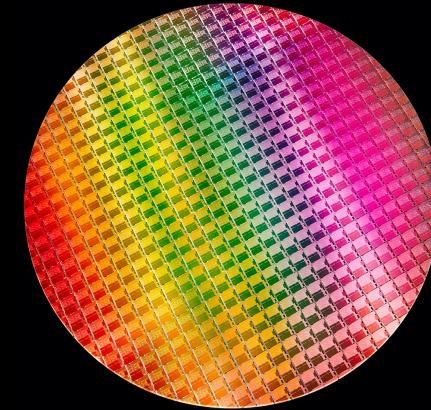
- conductors of electricity
- insulators
- semiconductors (can conduct or insulate under specific conditions)

Manufacturing integrated circuits

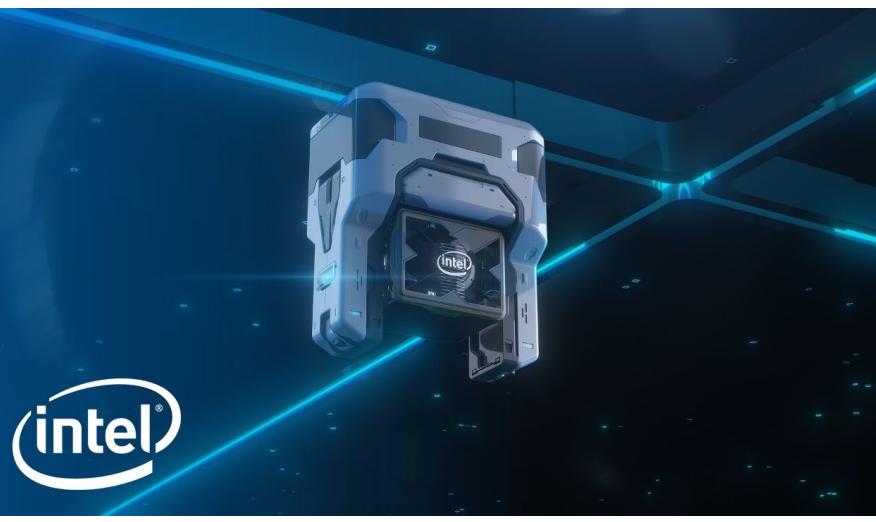


Intel® Core 10th Gen

- 300mm wafer (almost 1 foot), 506 chips, 10nm technology
- Each chip is 11.4 x 10.7 mm (a bit less than .5 in)



From sand to silicon



[https://www.youtube.com/watch?v= VMYPLXnd7E](https://www.youtube.com/watch?v=VMYPLXnd7E)