Computer Organization and Design
The Hardware/Software Interface

Chapter 1 - Computer Abstractions and Technology

Dr. Feng Li
fli@sdu.edu.cn
https://funglee.github.io

- **A website is constructed for this course.**

https://funglee.github.io/cod.html

Basic Information

- **SDU Course Number**
  - **SD01331470 计算机组成与设计**
  - **SD01331480 计算机组成与设计课程设计**
- **This course consists of two parts: lectures (SD01331470) and project design (SD01331470). The grades for SD01331470 will be based on the students' performance on assignments (20%) and final exam (80%), while the ones for SD01331470 will be given according to seven experiments.**

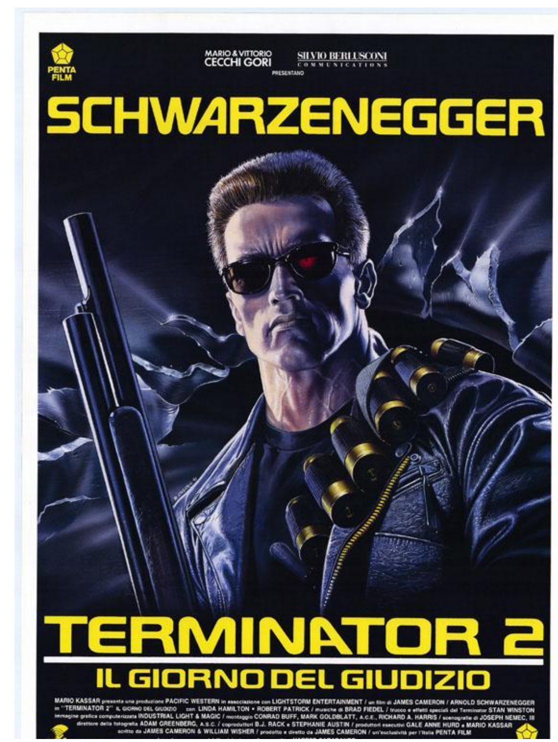
Tentative calendar

Course	Period (Weeks)	Venue	Type
Mon 7-8	1st-16th	Teaching Building 1-103	Lecture
Tue 1-2	1st-6th	Teaching Building 1-103	Lecture
Tue 1-2	7th-12th	Laboratory Building 403	Experiment
Tue 9-10	5th-12th	Teaching Building 4-109	Experiment
Thu 9-12	8th-15th	Laboratory Building 503	Experiment
Fri 1-2	3rd-12th	Laboratory Building 403	Experiment

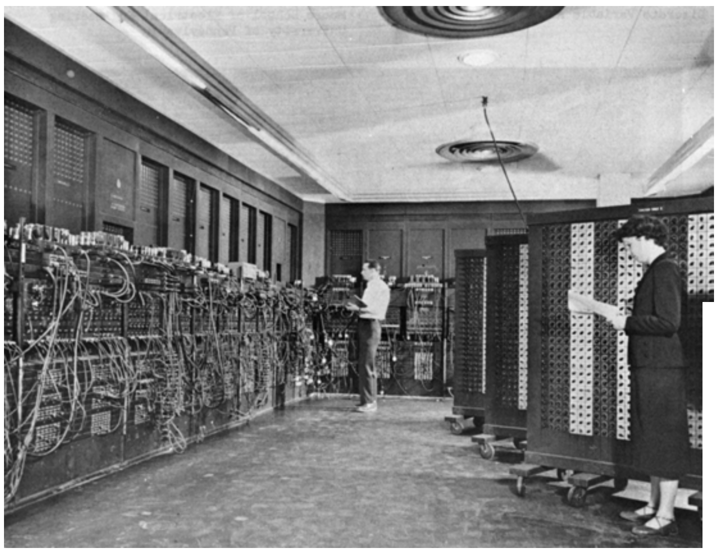
Contents of Chapter 1

Introduction

- Computers have led to a **third revolution** for civilization
- The following applications used to be “computer science fiction”
  - Computers in automobiles
  - Cell phones
  - Search engines
  - World Wide Web
  - Human genome project



History of Computing and Computers

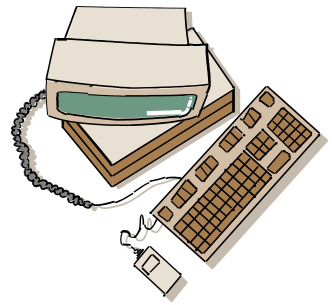


1946

2016

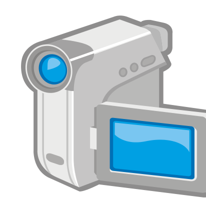
Classes of computer applications

- **Personal Computer**
  - E.g. Desktop, laptop



- **Server**
  - High performance
  - E.g. Mainframes, minicomputers, supercomputers, data center
  - Application
    - WWW, search engine, weather broadcast

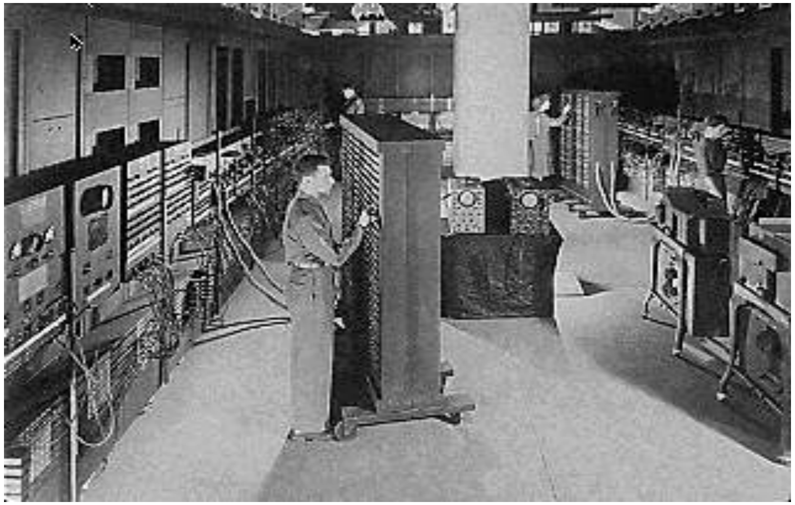
- **Embedded Computers**
  - **a computer system with a dedicated function within a larger mechanical or electrical system**
  - **E.g. Cell phone, microprocessors in cars/ television**

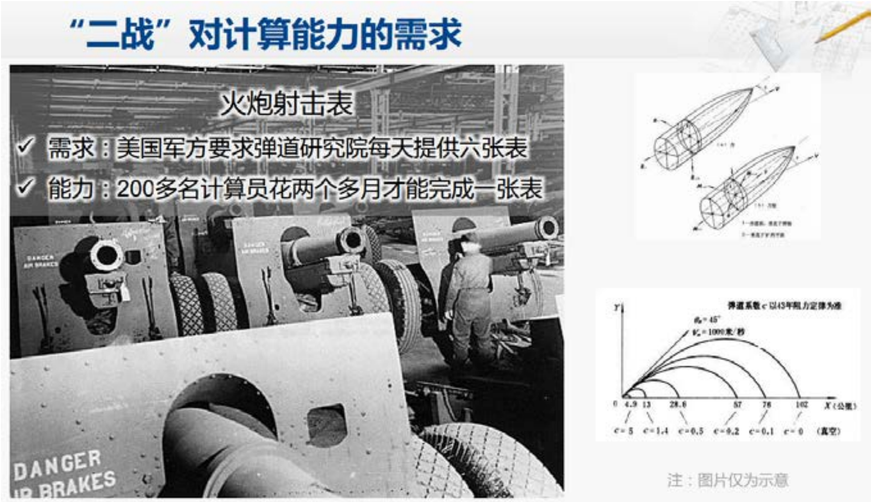


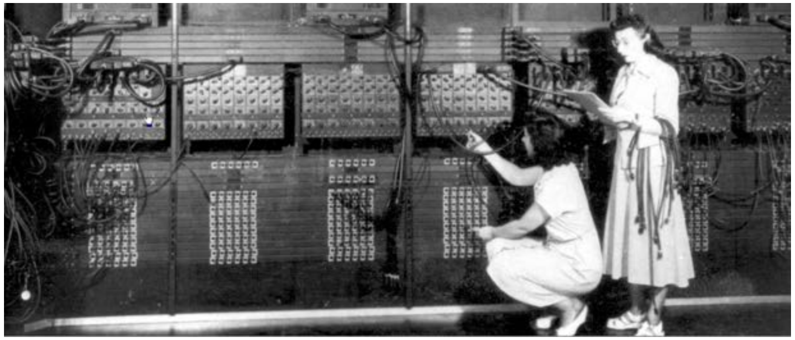
The first electronic general-purpose computer

ENIAC (Electronic Numerical Integrator and Computer)

- **Designed and built by Eckert and Mauchly at the University of Pennsylvania during 1943-45.**
- **It was Turing-complete, digital, and could solve “a large class of numerical problems” through reprogramming.**
  - **30 tons, 72 square meters, 200KW**
- **Performance**
  - **Read in 120 cards per minute**
  - **Addition took 20us, division 6ms**
- **Applications: ballistic calculations**

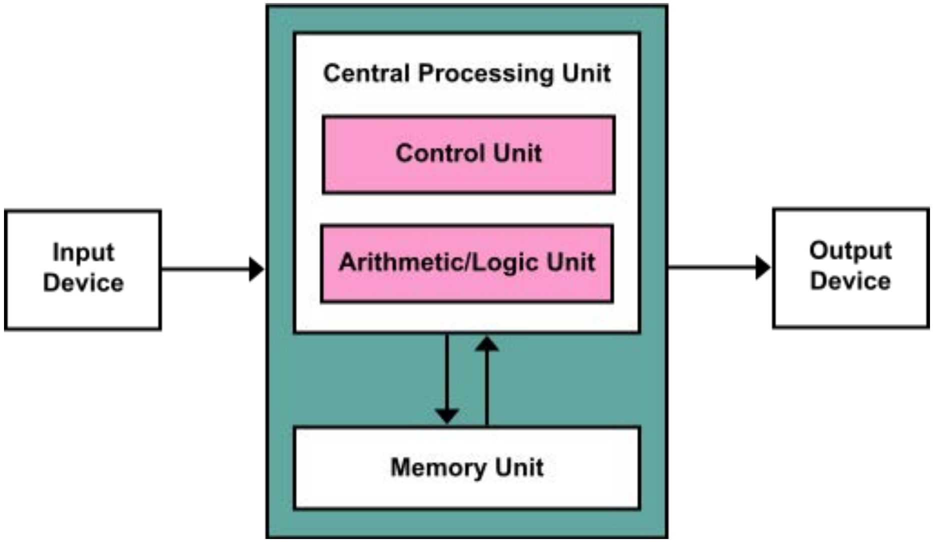






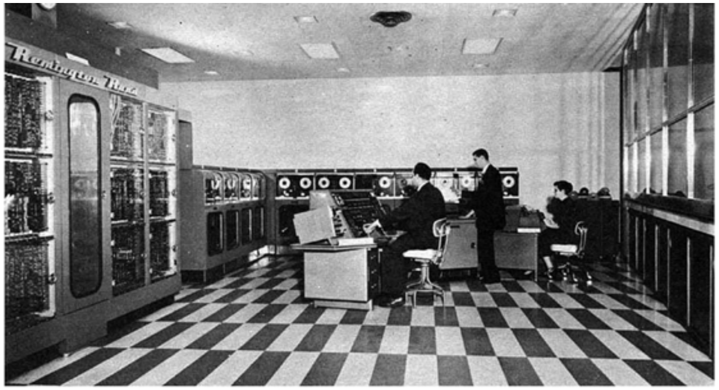
Von Neumann Architecture

- **Proposed by John von Neumann in 1945**



The First Commercial Computer

- **UNIVAC I (UNIVersal Automatic Computer I) designed by J. Presper Eckert and John Mauchly in 1951**
  - **5200 vacuum tubes**
  - **29000 pounds (13 tons)**
  - **125W**
  - **1905 operations per second**
  - **2.25 MHz clock**
  - $4.3m \times 2.4m \times 2.6m$



The TOP500 project ranks and details the 5000 most powerful (non-distributed) computer systems in the world.

Top 10 positions of the 46th TOP500 in November 2015

Rank	Rmax Rpeak (PFLOPS)	Name	Computer design Processor type, interconnect	Vendor	Site Country, year	Operating system
1	33.863 54.902	*Tianhe-2*	**NUDT** Xeon E5–2692 + Xeon Phi 31S1P, TH Express-2	NUDT	National Supercomputing Center in Guangzhou China, 2013	Linux (Kylin)
2	17.590 27.113	*Titan*	**Cray XK7** Opteron 6274 + Tesla K20X, Cray Gemini Interconnect	Cray Inc.	Oak Ridge National Laboratory United States, 2012	Linux (CLE, SLES based)
3	17.173 20.133	*Sequoia*	**Blue Gene/Q** PowerPC A2, Custom	IBM	Lawrence Livermore National Laboratory United States, 2013	Linux (RHEL and CNK)
4	10.510 11.280	*K computer*	**RIKEN** SPARC64 VIIIfx, Tofu	Fujitsu	RIKEN Japan, 2011	Linux
5	8.586 10.066	*Mira*	**Blue Gene/Q** PowerPC A2, Custom	IBM	Argonne National Laboratory United States, 2013	Linux (RHEL and CNK)
6	8.101 11.079	Trinity	**Cray XC40** Xeon E5-2698v3, Cray Aries Interconnect	Cray Inc.	DOE/NNSA/LANL/SNL United States, 2015	Linux (CLE)
7	6.271 7.779	*Piz Daint*	**Cray XC30** Xeon E5–2670 + Tesla K20X, Aries	Cray Inc.	Swiss National Supercomputing Centre Switzerland, 2013	Linux (CLE)
8	5.640 7.404	Hazel Hen	**Cray XC40** Xeon E5-2680v3, Cray Aries Interconnect	Cray Inc.	HLRS - Höchstleistungsrechenzentrum, Stuttgart Germany, 2015	Linux (CLE)
9	5.537 7.235	*Shaheen II*	**Cray XC40** Xeon E5–2698v3, Aries	Cray Inc.	King Abdullah University of Science and Technology Saudi Arabia, 2015	Linux (CLE)
10	5.168 8.520	*Stampede*	**PowerEdge C8220** Xeon E5–2680 + Xeon Phi, Infiniband	Dell	Texas Advanced Computing Center United States, 2013	Linux (CentOS)[13]

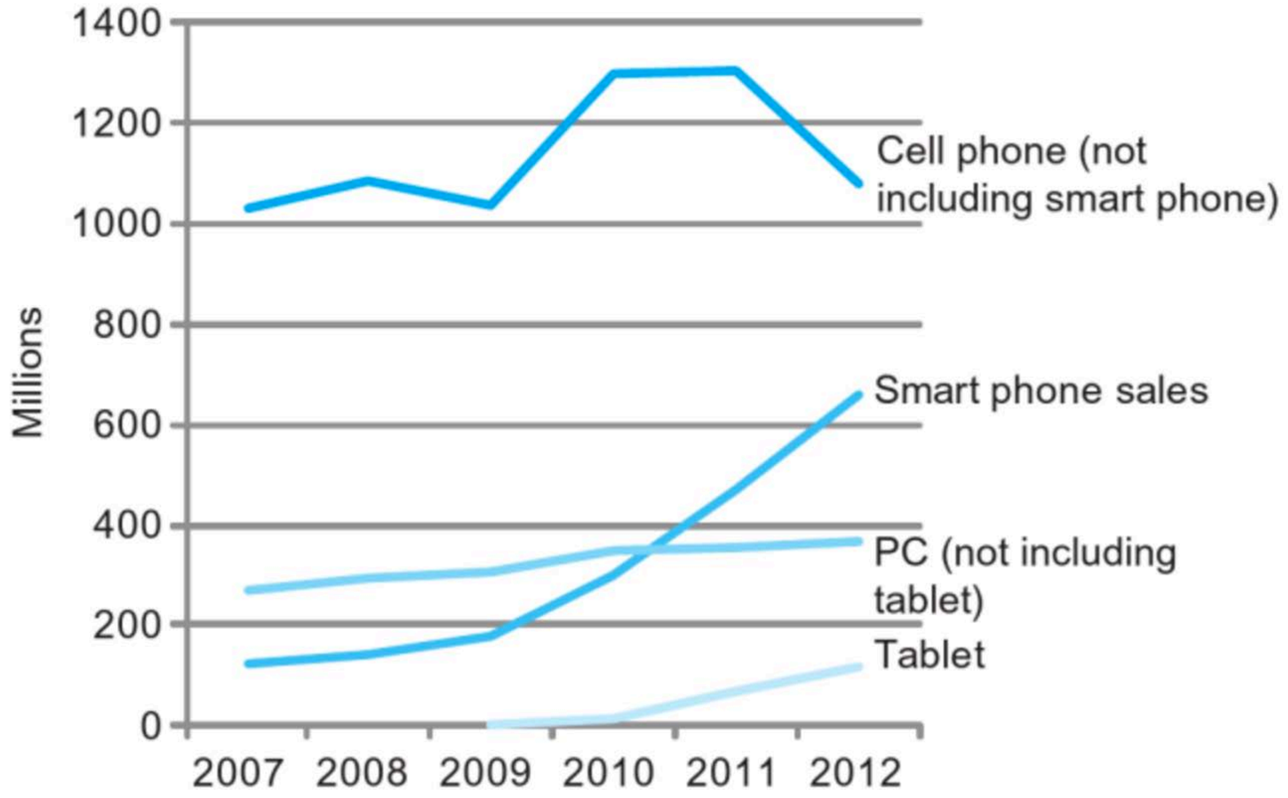
FLOPS (Floating-point operations per second), PFLOPS = 10^{15}FLOPS

Tianhe-2

- 16000 computer nodes, each comprising two Intel Ivy Bridge Xeon processors and three Xeon Phi coprocessor chips.
- Each node is equipped with memory of 88 GiB
- HD array 12.4 PiB
- Price: 3.9 million US$
- Applications:
    - scientific computing

Welcome to the PostPC era

- Personal Mobile Devices (PMDs)
  - E.g., Smart phones/watches/glasses, cell phones, tablet computers
  - Low price
  - Wireless access
  - Powered by batteries
  - Equipped with touch-sensitive screen or speech input
- Cloud Computing
  - Warehouse Scale Computers (WSCs)
  - Virtualization technology



The number of manufactured per year of tablets and smart phones, which reflect the PostPC era, versus personal computers and traditional cell phones

Cloud Computing

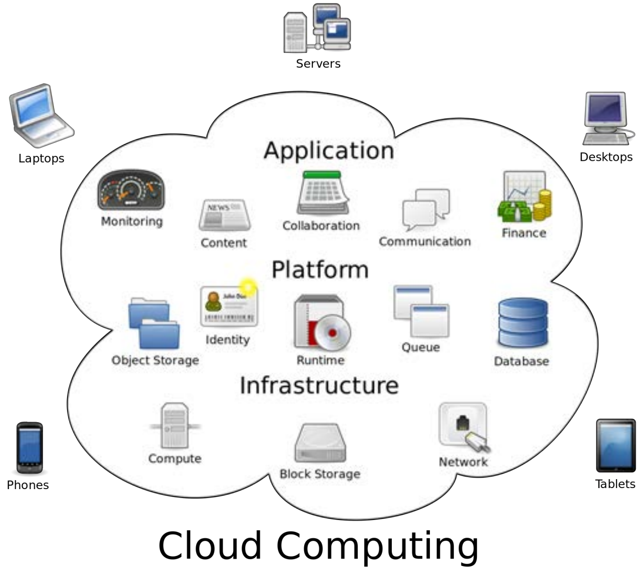
- Everyone is talking about Cloud Computing, but what is it?
  - Computing service is managed, scheduled, and delivered to users over Internet.
  - For example
    - Google Drive
    - One Drive
    - Hotmail
    - Gmail

Characteristics

- On demand self-service
- Access to networks anywhere, anytime, on any devices
- Location independent resource pooling
- Deployment flexibility
- Pay as you go

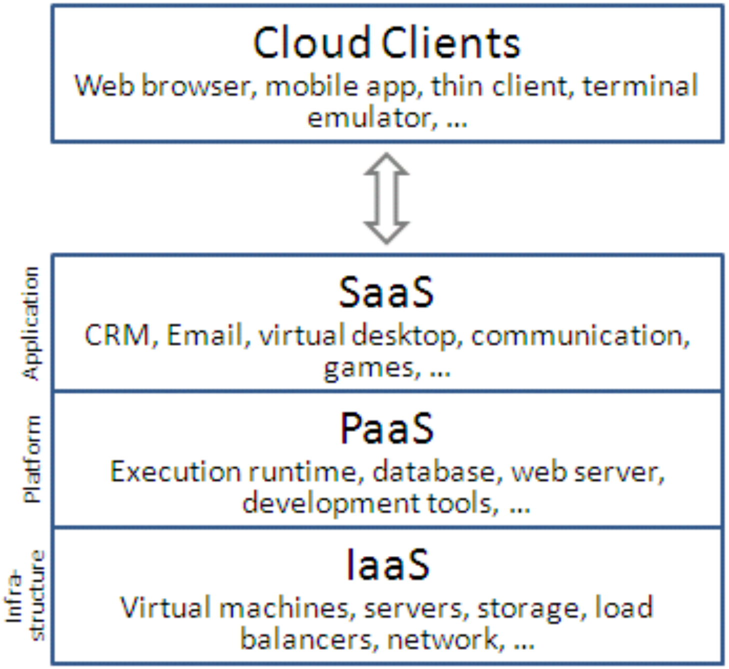
Infrastructures for Cloud Computing

- Development of computing capability
- Virtualization technology
- Distributed Storage
- Automated Storage
- Fast internet access



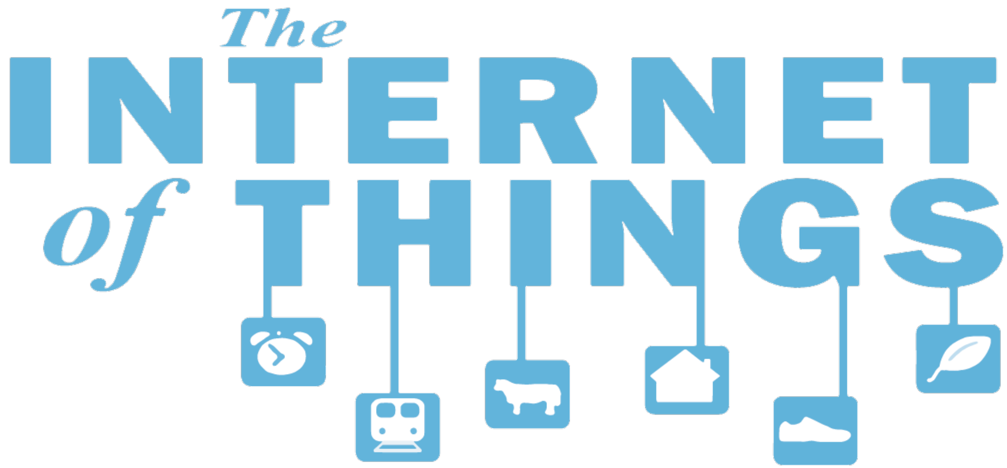
Services of Cloud Computing

- SaaS: Software as a Service
  - Gmail, Hotmail, Flickr, OfficeLive
- PaaS: Platform as a Service
  - Amazon EC2, Microsoft Azure
- IaaS: Infrastructure as a Service
  - AT&T Hosting and Storage
  - Amazon EC2



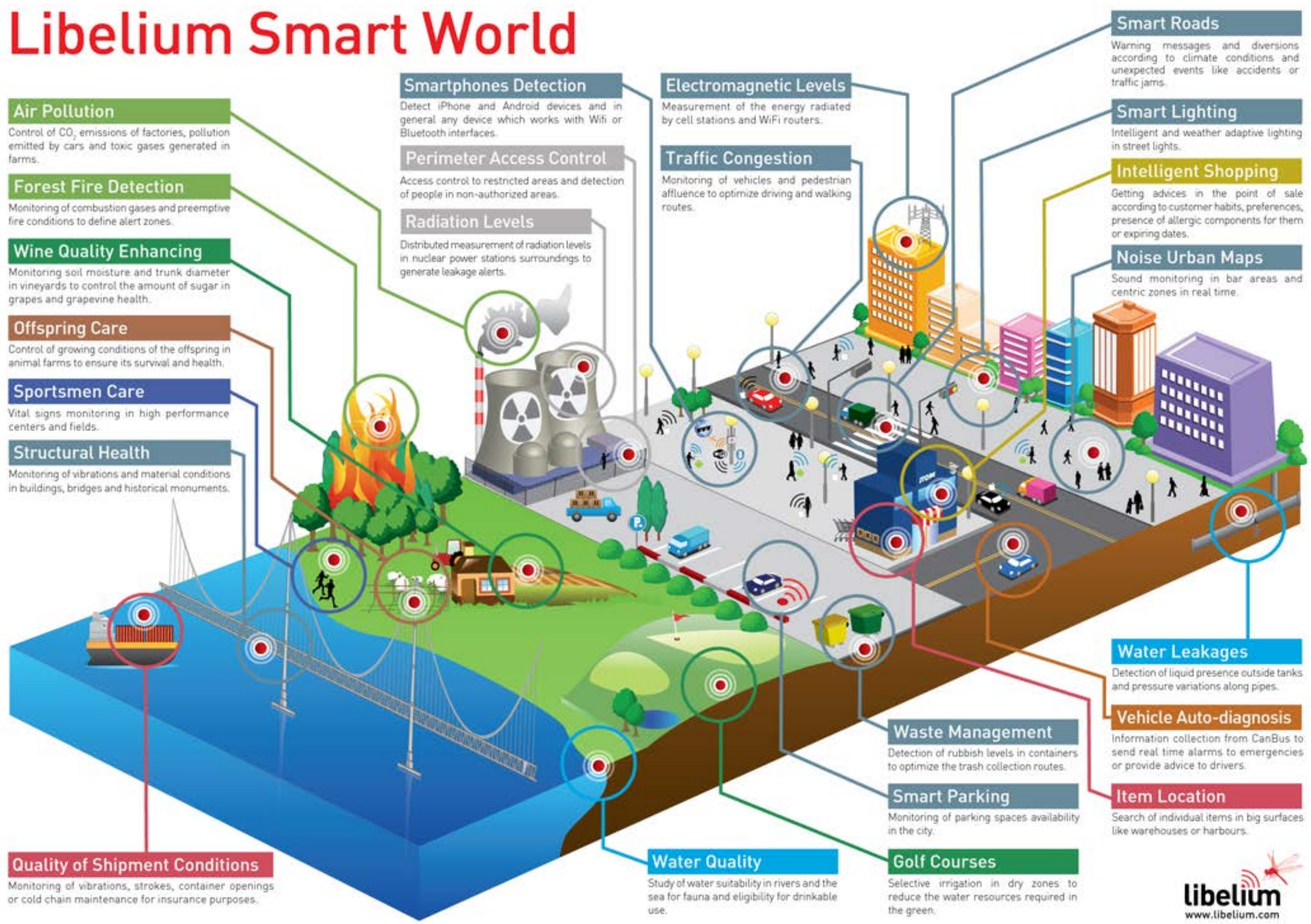
Internet of Things (IoT)

- Smart + X
  - Smart City
  - Smart Traffic
  - Smart Building
  - Smart …
- Wireless Sensor Networks
  - Sensor Motes
  - Mobile Phones
- RFID Systems



IoT Systems include...

- **Sensors**
  - **We look at the world through sensors, e.g., light sensors, cameras, microphones, motion sensors, accelerators, gyroscopes, magnetic sensors, barometers, GPS.**
- **Networks and communications**
  - **The sensed data are transmitted, stored and processed in a networked fashion, e.g., WAN, MAN, LAN, PAN.**
  - **Various communication techniques are combined in the systems, e.g., 3G, 4G, Bluetooth, WiFi, ZigBee, RFID, NB-IoT.**
- **Applications, people and processes**
  - **All the data are fed back to applications, people and processes for further process and analysis, and finally are sued to make better decision, e.g., remote monitoring, mobile apps, security, supply chain management, locating and tracking, control and automation.**


Libelium Smart World
Air Pollution
Control of CO_2 emissions of factories, pollution emitted by cars and toxic gases generated in farms.
Forest Fire Detection
Monitoring of combustion gases and preemptive fire conditions to define alert zones.
Wine Quality Enhancing
Monitoring soil moisture and trunk diameter in vineyards to control the amount of sugar in grapes and grapevine health.
Offspring Care
Control of growing conditions of the offspring in animal farms to ensure its survival and health.
Sportsmen Care
Vital signs monitoring in high performance centers and fields.
Structural Health
Monitoring of vibrations and material conditions in buildings, bridges and historical monuments.
Smartphones Detection
Detect iPhone and Android devices and in general any device which works with Wifi or Bluetooth interfaces.
Perimeter Access Control
Access control to restricted areas and detection of people in non-authorized areas.
Radiation Levels
Distributed measurement of radiation levels in nuclear power stations surroundings to generate leakage alerts.
Electromagnetic Levels
Measurement of the energy radiated by cell stations and WiFi routers.
Traffic Congestion
Monitoring of vehicles and pedestrian affluence to optimize driving and walking routes.
Smart Roads
Warning messages and diversions according to climate conditions and unexpected events like accidents or traffic jams.
Smart Lighting
Intelligent and weather adaptive lighting in street lights.
Intelligent Shopping
Getting advices in the point of sale according to customer habits, preferences, presence of allergic components for them or expiring dates.
Noise Urban Maps
Sound monitoring in bar areas and centric zones in real time.
Water Leakages
Detection of liquid presence outside tanks and pressure variations along pipes.
Vehicle Auto-diagnosis
Information collection from CanBus to send real time alarms to emergencies or provide advice to drivers.
Item Location
Search of individual items in big surfaces like warehouses or harbours.
Waste Management
Detection of rubbish levels in containers to optimize the trash collection routes.
Smart Parking
Monitoring of parking spaces availability in the city.
Golf Courses
Selective irrigation in dry zones to reduce the water resources required in the green.
Water Quality
Study of water suitability in rivers and the sea for fauna and eligibility for drinkable use.
Quality of Shipment Conditions
Monitoring of vibrations, strokes, container openings or cold chain maintenance for insurance purposes.
libelium
www.libelium.com

What can we learn in this lesson?

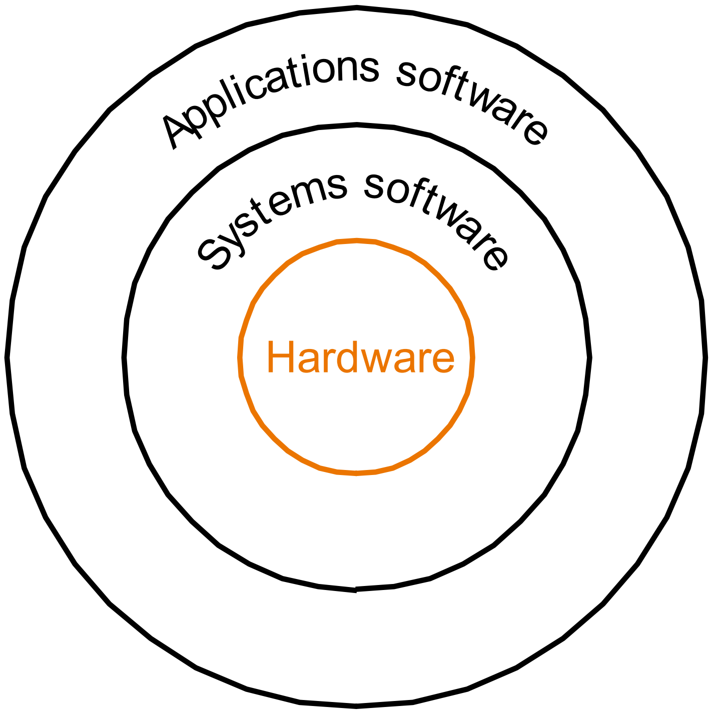
- How are the programs written in high-level languages (e.g., C, C++, or Java) translated into the language of hardware? How does the hardware execute the resulting program.
- What is the interface between software and hardware, and how does software instruct the hardware to perform required functions?
- What determines the performance of a program, and how can a programmer improve the performance?
- What techniques can be employed by hardware designers to improve energy efficiency?
- What are the reasons for and the consequences of the recent switch from sequential processing to parallel processing?
- …

Eight great ideas in computer architecture

- Design for Moore's Law
- 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

Below your programs

A simplified view of hardware and software as hierarchical layers



- **Systems software**
  - **aimed at programmers**
  - **E.g. Operation Systems, Compiler**
- **Applications software**
  - **aimed at users**
  - **E.g. Word, IE, QQ, WeChat**

Computer language and software system

- **Computer language**
  - **Computers only understands electrical signals**
  - **Easiest signals: *on* and *off***
  - **Binary numbers express machine instructions**

  e.g. 1000110010100000 means to add two numbers
  - **Very tedious to write**
- **Assembly language**
  - **Symbolic notations**

    e.g. add a, b, c #a=b+c
  - **The assembler translates them into machine instruction**
  - **Programmers have to think like the machine**

- **High-level programming language**
  - **Notations more closer to the natural language**
  - **The compiler translates them into assembly language statements**
  - **Advantages over assembly language**
    - **Programmers can think in a more natural language**
    - **Improved programming productivity**
    - **Programs can be independent of hardware**
  - **Subroutine library ---- reusing programs**
- **Which one faster?**
  - **Asm、C、C++、Java**
  - **Lower, faster**

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;
}
```

Compiler

Assembly language program (for MIPS)

```
swap:
      multi $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
```

Assembler

Binary machine language program (for MIPS)

```
00000000101000100000000100011000
00000000100000100001000000100001
10001101111000100000000000000000
10001110000100100000000000000100
10101110000100100000000000000000
10101101111000100000000000000100
00000011111000000000000000001000
```

Under the covers

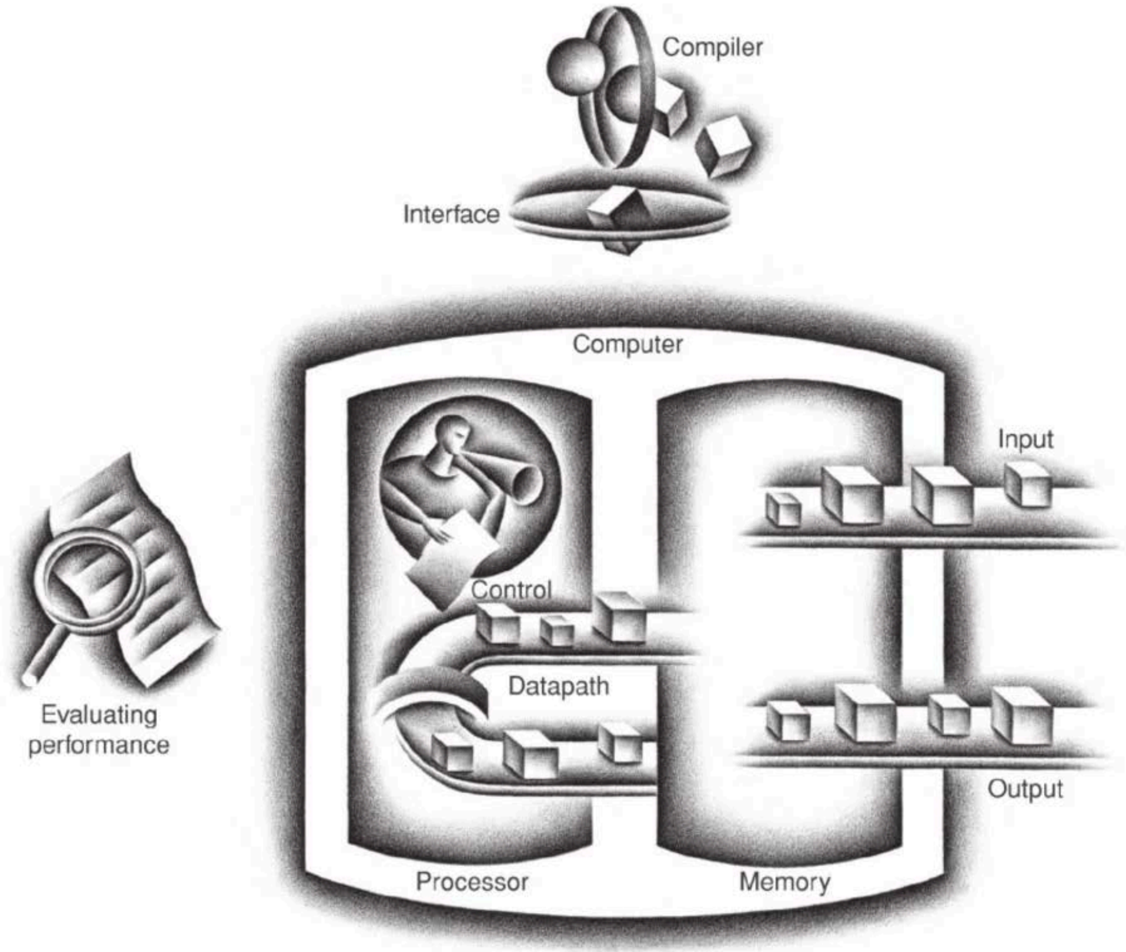
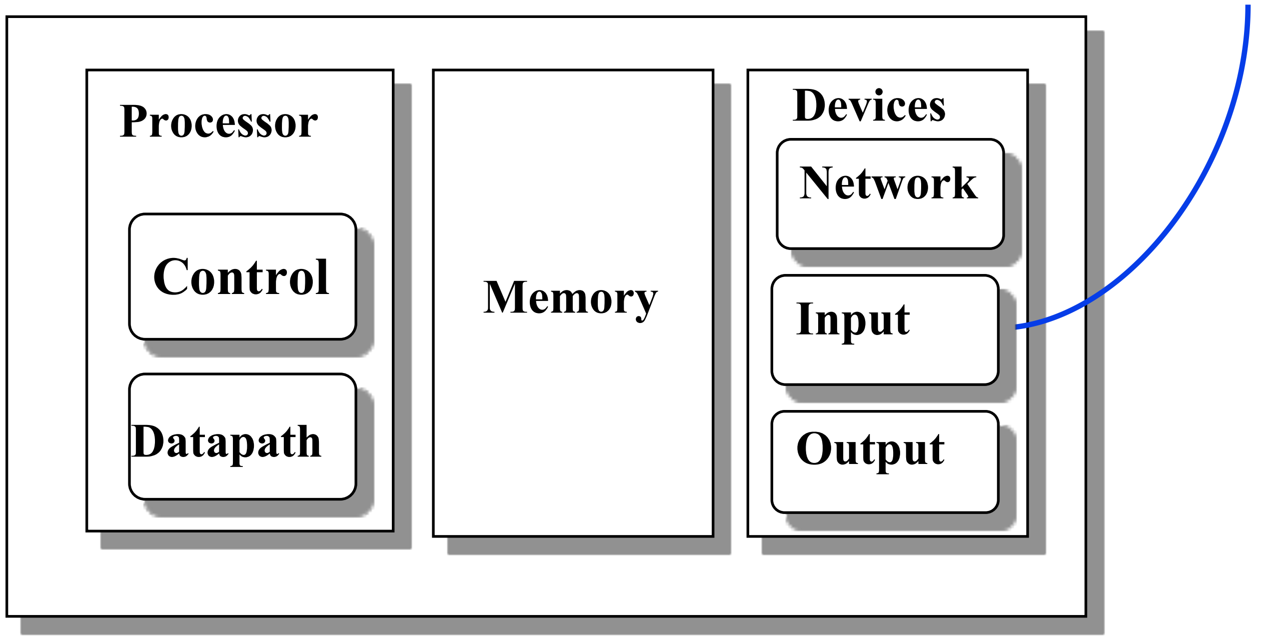


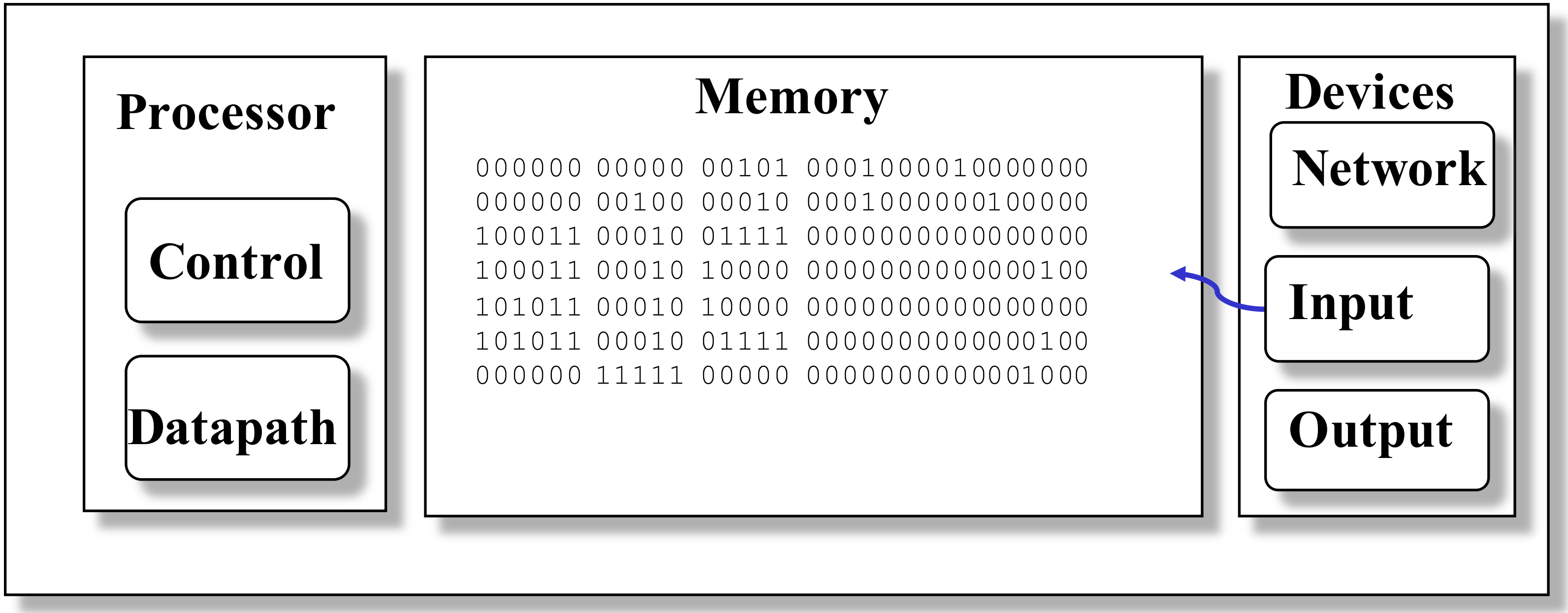
FIGURE 1.5 The organization of a computer, showing the five classic components. The processor gets instructions and data from memory. Input writes data to memory, and output reads data from memory. Control sends the signals that determine the operations of the datapath, memory, input, and output.

Input Device Inputs Object Code

```
000000 00000 00101 0001000010000000
000000 00100 00010 0001000000100000
100011 00010 01111 0000000000000000
100011 00010 10000 0000000000000100
101011 00010 10000 0000000000000000
101011 00010 01111 0000000000000100
000000 11111 00000 0000000000001000
```

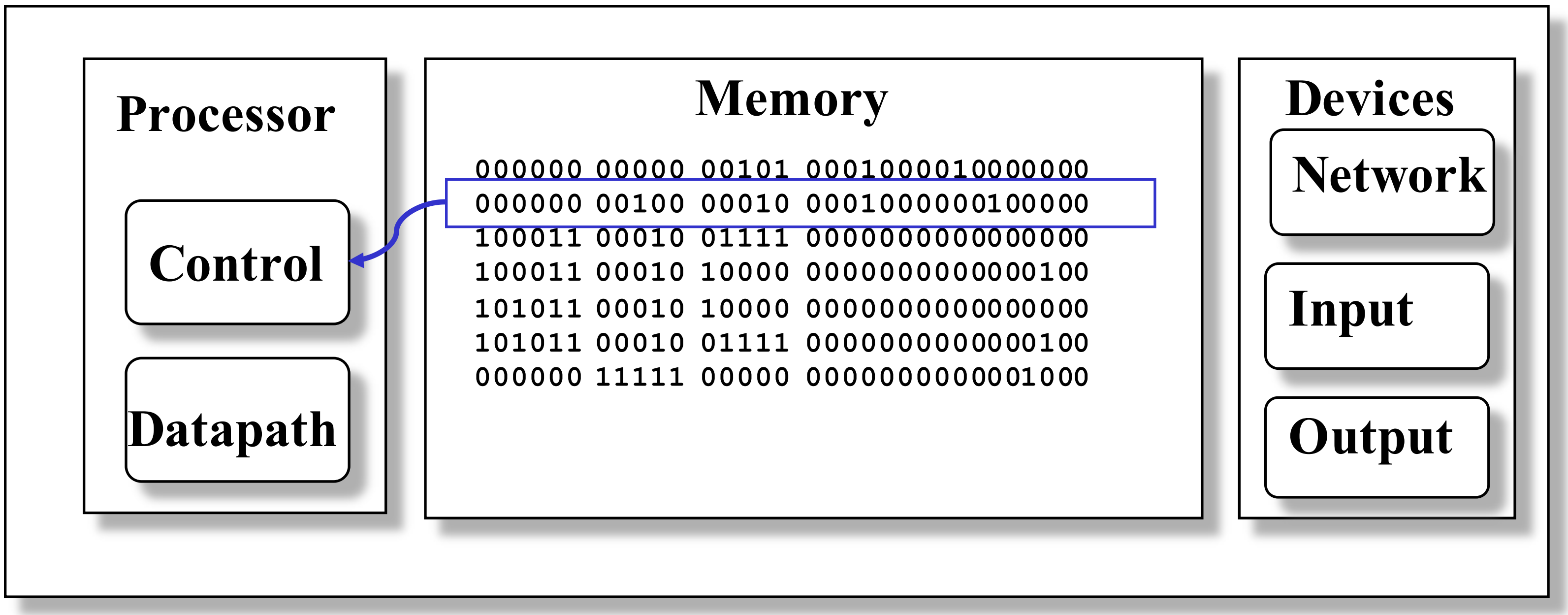


Object Code Stored in Memory



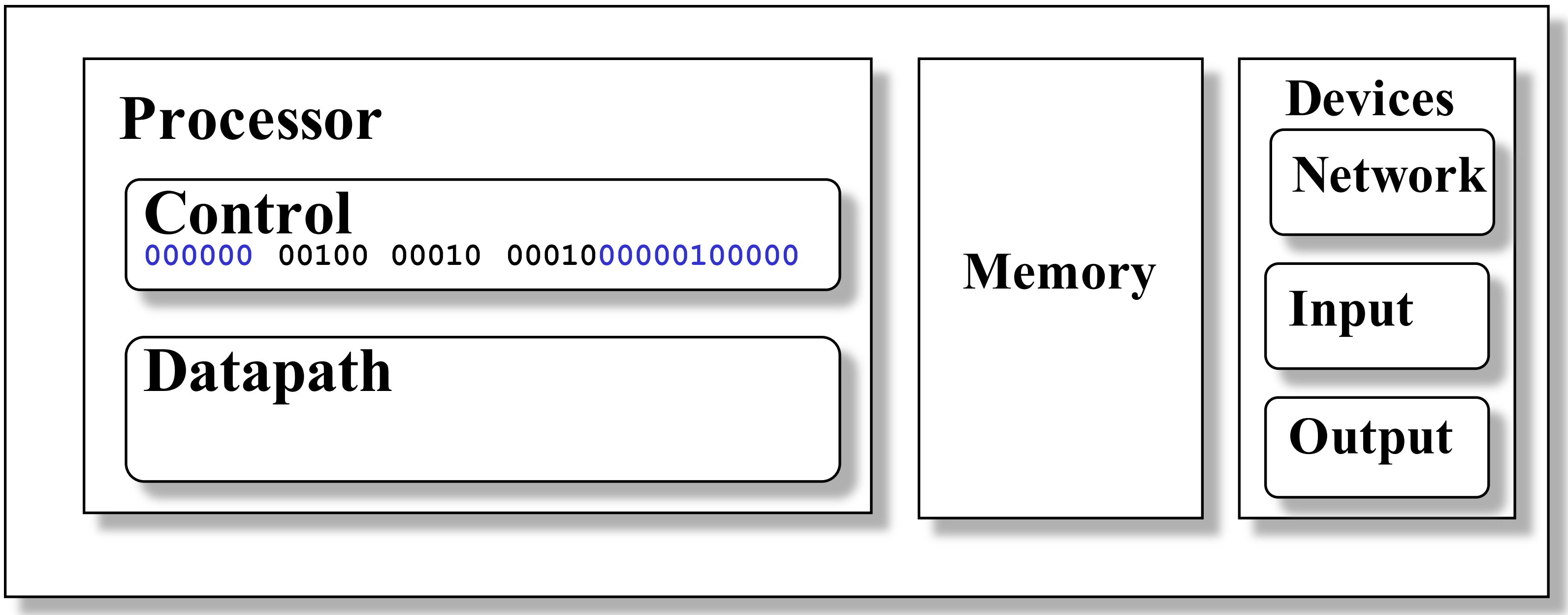
Processor Fetches an Instruction

Processor fetches an instruction from memory



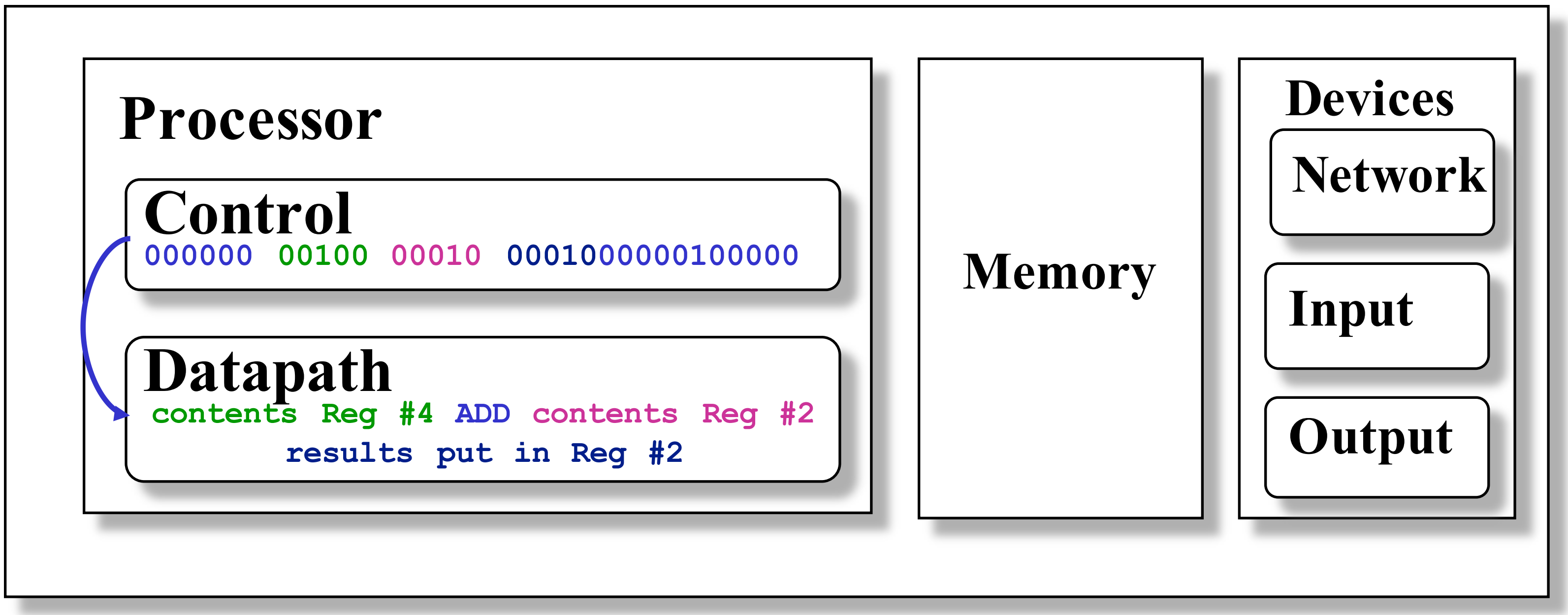
Control Decodes the Instruction

Control decodes the instruction to determine what to execute



Datapath Executes the Instruction

Datapath executes the instruction as directed by control



- **Display**
  - CRT (raster cathode ray tube) display
    - Scan an image one line at a time, 30 to 75 times / s
    - Pixels and the bit map, 512×340 to 1560×1280
    - The more bits per pixel, the more colors to be displayed

  - LCD (liquid crystal display)
    - Thin and low-power
    - The LCD pixel is not the source of light
    - Rod-shaped molecules in a liquid that form a twisting helix that bends light entering the display

- **Touchscreens**
  - **A substitute for keyboard and mouse**
  - **Since people are electrical conductors, if an insulator (e.g., glass) is covered with a transparent conductor, touching distorts the electrostatic field of the screen, which results in a change in capacitance.**

Opening the box

- I/O devices
  - Capacitive multi-touch LCD display
  - Front facing camera
  - Ear facing camera
  - Microphone
  - Headphone jack
  - Speaker
  - Accelerometer
  - Gyroscope
  - WiFi network
  - Bluetooth network

iPad 2

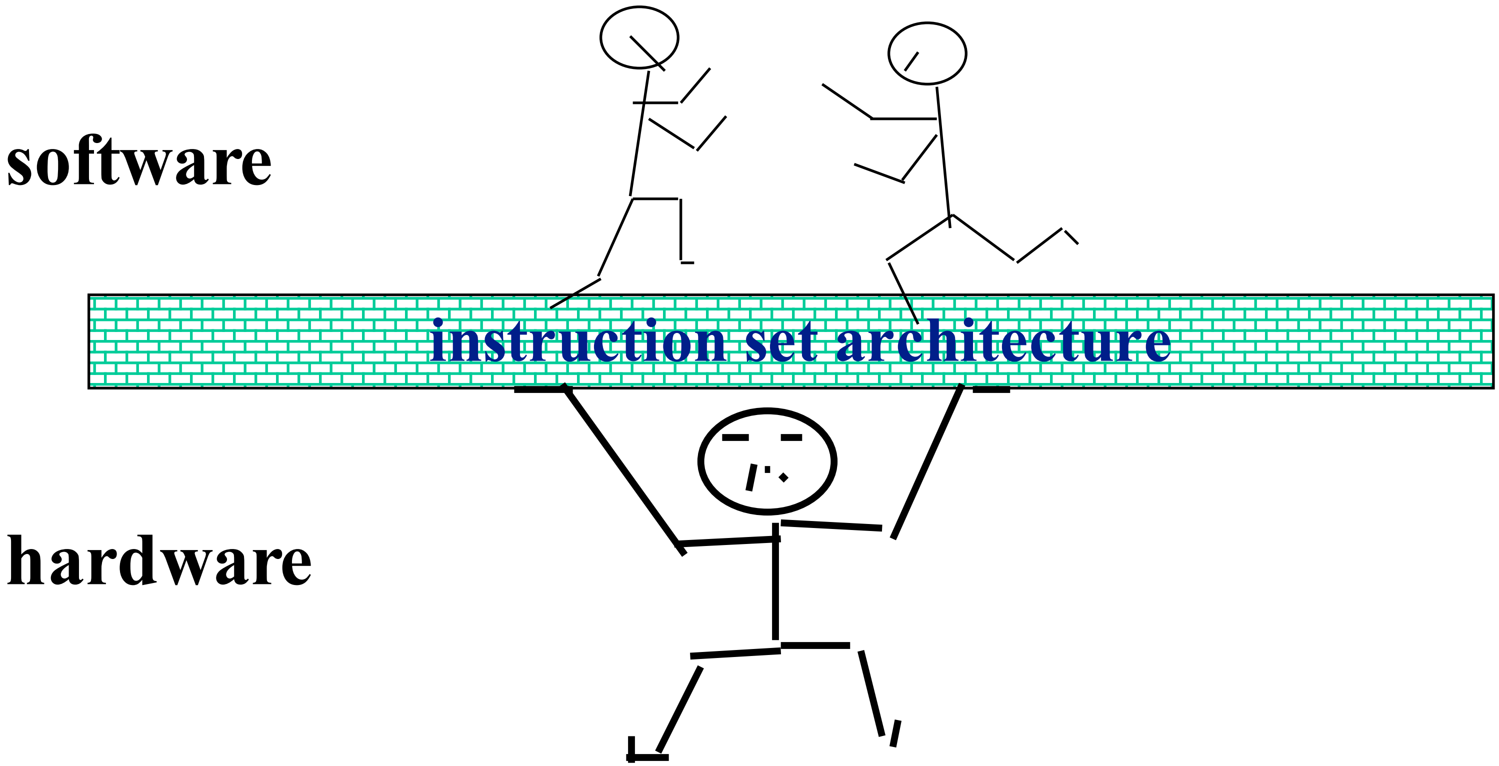
- **Integrated circuits (chips)**
  - **Apple A5 chips consisting of two 1 GHz ARM processor cores as well as 512 MB DRAM main memory**
  - **Flash memory chips for non-volatile storage**
  - **Power controller**
  - **I/O Controller**

- **The processor integrated circuit inside the A5 package**
  - **Data path performs the arithmetic operations**
  - **Control tells the data path, memory, and I/O devices what to do according to the wishes of the instructions of the program**

Memory architecture

- **Cache memory**
  - **A small, fast (but expensive) memory that acts as a buffer for the DRAM memory**
  - **Built by a different memory technology, Static Random Access Memory (SRAM)**
- **Volatile main memory**
- **Nonvolatile secondary memory**

Instruction Set Architecture (ISA)



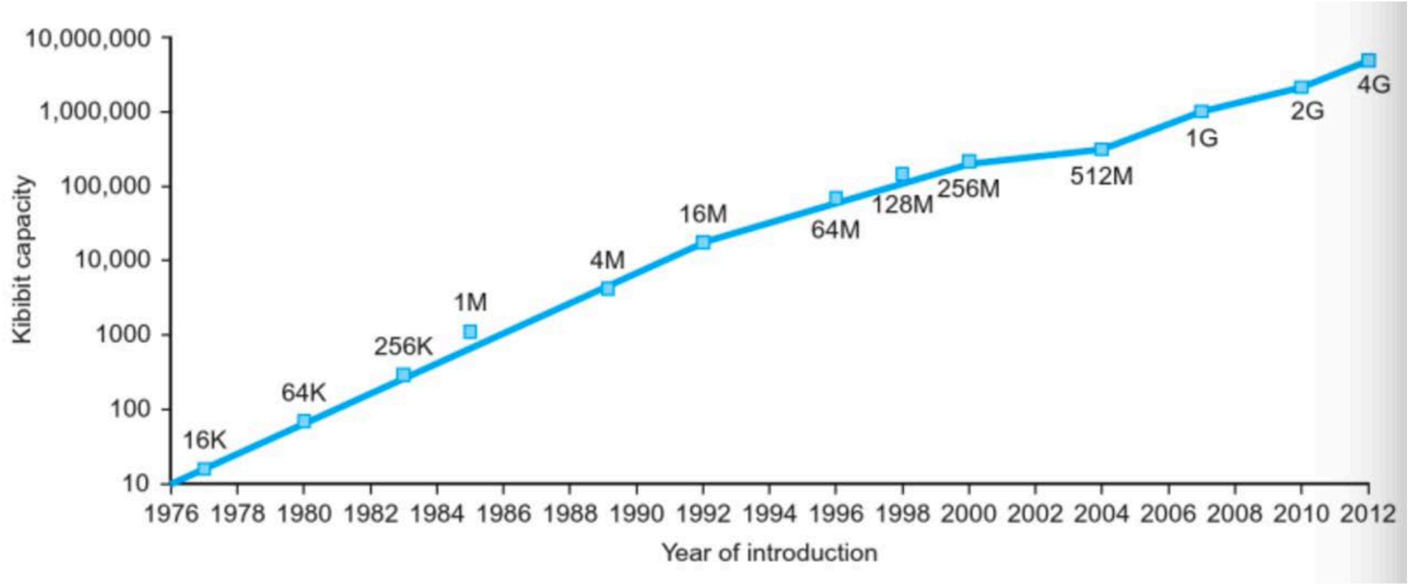
The interface description separating the software and hardware

Communicating with other computers

- Individual computers can be networked to exchange messages
- Advantages
  - Communication
  - Resource sharing
  - Nonlocal access

Technologies for building processors and memory

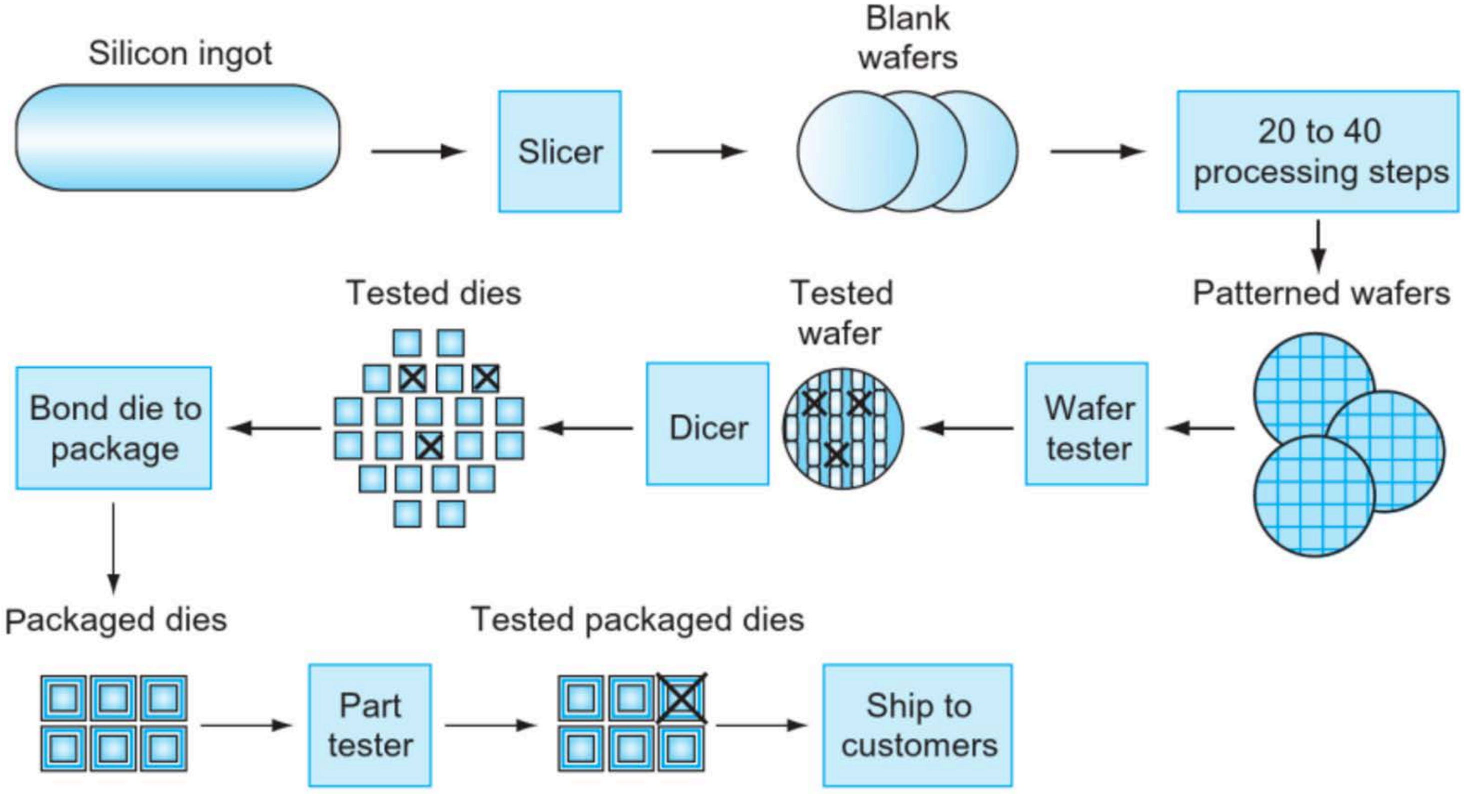
- Recent years have witnessed fast improvement of processors and memory
- Integrated circuits
  - A transistor is an on/off switch controlled by electricity
  - An integrated circuit consisting of hundreds of transistors. Even, a Very Large-Scale Integrated circuit (VLSI) contains billions of transistors.



Growth of capacity per DRAM chip over time

Manufacturing Process

- The manufacture of a chip begins with silicon that is a semiconductor.
- Through a special chemical process, we can add materials to silicon such that its tiny areas are transformed into one the the following three devices: i) excellent conductor; ii) excellent insulator; iii) transistors



Processor Technology Trends

- Shrinking of transistor sizes: 250nm (1997) → 130nm (2002) → 70nm (2008) → 35nm (2014)
- Transistor density increases by 35% per year and die size increases by 10-20% per year… functionality improvements!
- Transistor speed improves linearly with size (complex equation involving voltages, resistances, capacitances)

Performance

Performance metrics:

- *Response time, wall-clock time,* or *elapsed time*
  - Total time to complete a task, including disk access, memory access, I/O activities, OS overhead …
- *CPU Execution time (or CPU time)*
  - The time CPU spends computing for certain program, not include time spent waiting for I/O or running others
  - User CPU time and system CPU time
- *Throughput (or bandwidth)*
  - the total amount of work done in a given time.

- For a computer X,

$$\text{Performance} = \frac{1}{\text{Execution time}_X}$$

- *"X is faster than Y"*
  - the execution time on Y is longer than that on X.
- *"X is n times faster than Y"*

$$\frac{\text{Execution time}_Y}{\text{Execution time}_X} = n$$

- *"The throughput of X is 1.3 times higher than Y"*
  - The number of tasks completed per unit time on machine X is 1.3 times the number completed on Y.

Example

- **If computer A runs a program in 10 seconds and computer B runs the same program in 15 seconds, how much faster is A than B?**

- **The performance ratio is 15/10=1.5; therefore, A is 1.5 times faster than B**

Measuring Performance

- *wall-clock time*, *response time*, or *elapsed time*,
  - Including disk accesses, memory accesses, input/output activities, operating system overhead —everything.
- *CPU Execution time (or CPU time)*
  - The time CPU spends computing for certain program, not include time spent waiting for I/O or running others
  - *CPU time= user CPU time+ system CPU time*.
    - **User CPU time: the CPU time spent in a program itself**
    - **System CPU time: the CPU time spent in the operating system**

Clock

- **Measuring how fast the hardware can perform basic functions**
- **For all computers, their behaviors are conducted by a clock.**
- **The discrete time intervals are called clock cycles (or ticks, clock ticks, clock periods, clocks, cycles)**
- **Clock cycle (also tick): the time for one clock period.**
  - 250ps (1 Picosecond $= 10^{-12}$ second)
- **Clock rate: the count of clocks in one second.**
  - 4GHz (1 Gigahertz $= 10^{-9}$ Hertz)

CPU Performance

- $$\frac{\text{CPU execution time}}{\text{for a program}} = \frac{\text{CPU clock cycles}}{\text{for a program}} \times \text{Clock cycle time}$$

- **One program runs in 10 seconds on computer A, which has a 2 GHz clock. Computer B requires 1.2 times as many clock cycles as computer A for this program. To run this program in 6 seconds, what clock rate should the computer B supply?**

$$CPU\ clock\ cycles_A = CPU\ time_A \times Clock\ rate_A = 10s \times 2\times10^9 Hz = 2\times10^{10} cycles$$

$$Clock\ rate_B = \frac{1.2 \times CPU\ clock\ cycles_A}{CPU\ time_B} = \frac{1.2\times20\times10^9\ cycles}{6s} = 4\ GHz$$

Instruction Performance

- CPU clock cycles = Instructions for a program × Average clock cycles per instruction
- Clock cycles per instruction (CPI): average number of clock cycles per instruction for a program
  - Different instructions may take different amounts of time depending on what they do.
- CPI provides one way of comparing two different implementations of the same instruction set architecture.
- Instruction set architecture (ISA)
  - The number of instructions executed for a program will be the same, if the program run in two different implementations of the same instruction set architecture.
- CPU performance equations
  - CPU time=Instruction count ×CPI×Clock cycle time
  - CPU time=Instruction count ×CPI/Clock rate

Example

- Suppose we have two implementations of the same instruction set architecture. Computer A has a clock cycle time for 250ps and a CPI of 2.0 for some program, and computer B has a clock cycle time of 500ps and a CPI of 1.2 for the same program. Which computer is faster for this program and by how much?

I: the number of instructions for the program

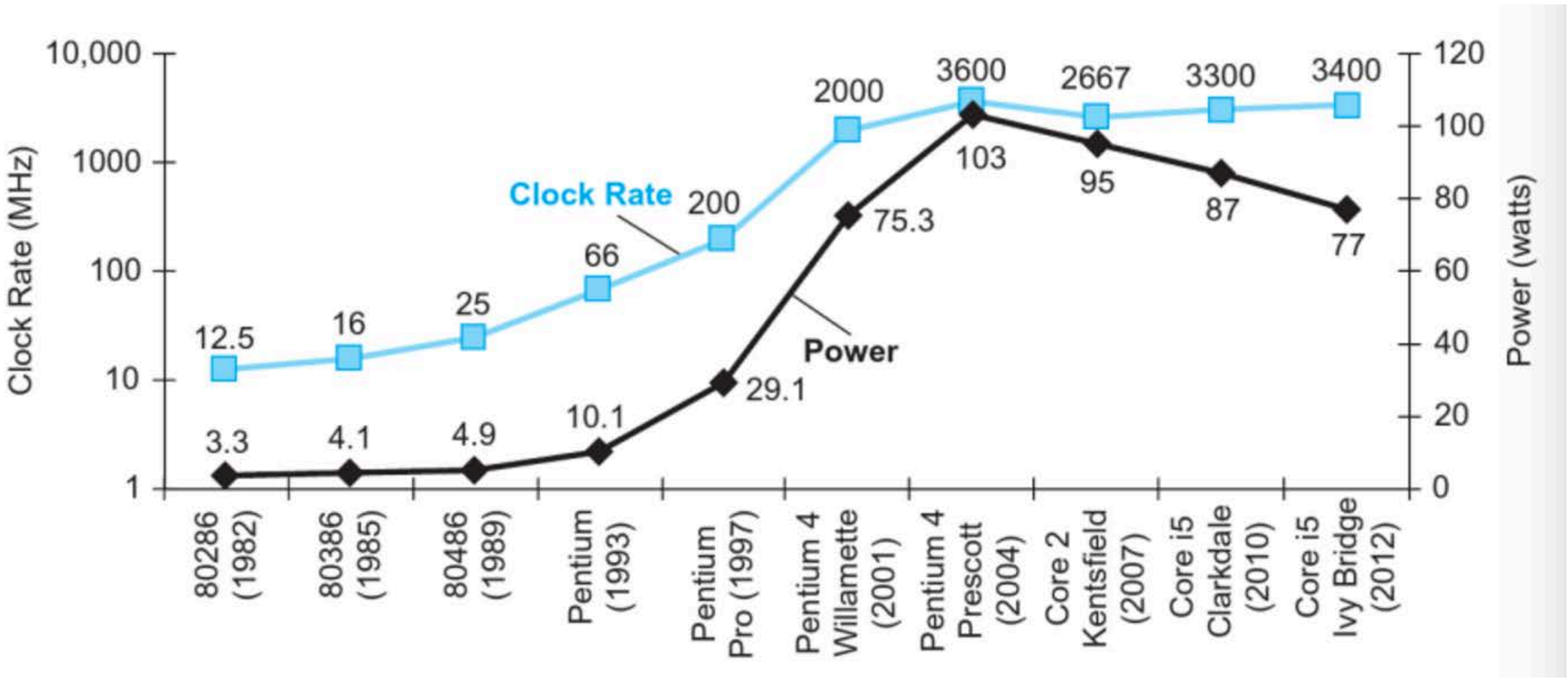
CPU $time_A = I \times 2 \times 250$ ps $= 500 \times I$ ps

CPU $time_B = I \times 1.2 \times 500$ ps $= 600 \times I$ ps

Computer A is 1.2 times as fast as computer B

The power wall

- The advancement of clock rate significantly increases power consumption
- In PostPC Era, battery life can trump performance in the PMDs, while the the power consumed in cooling the warehouse scale computers is huge



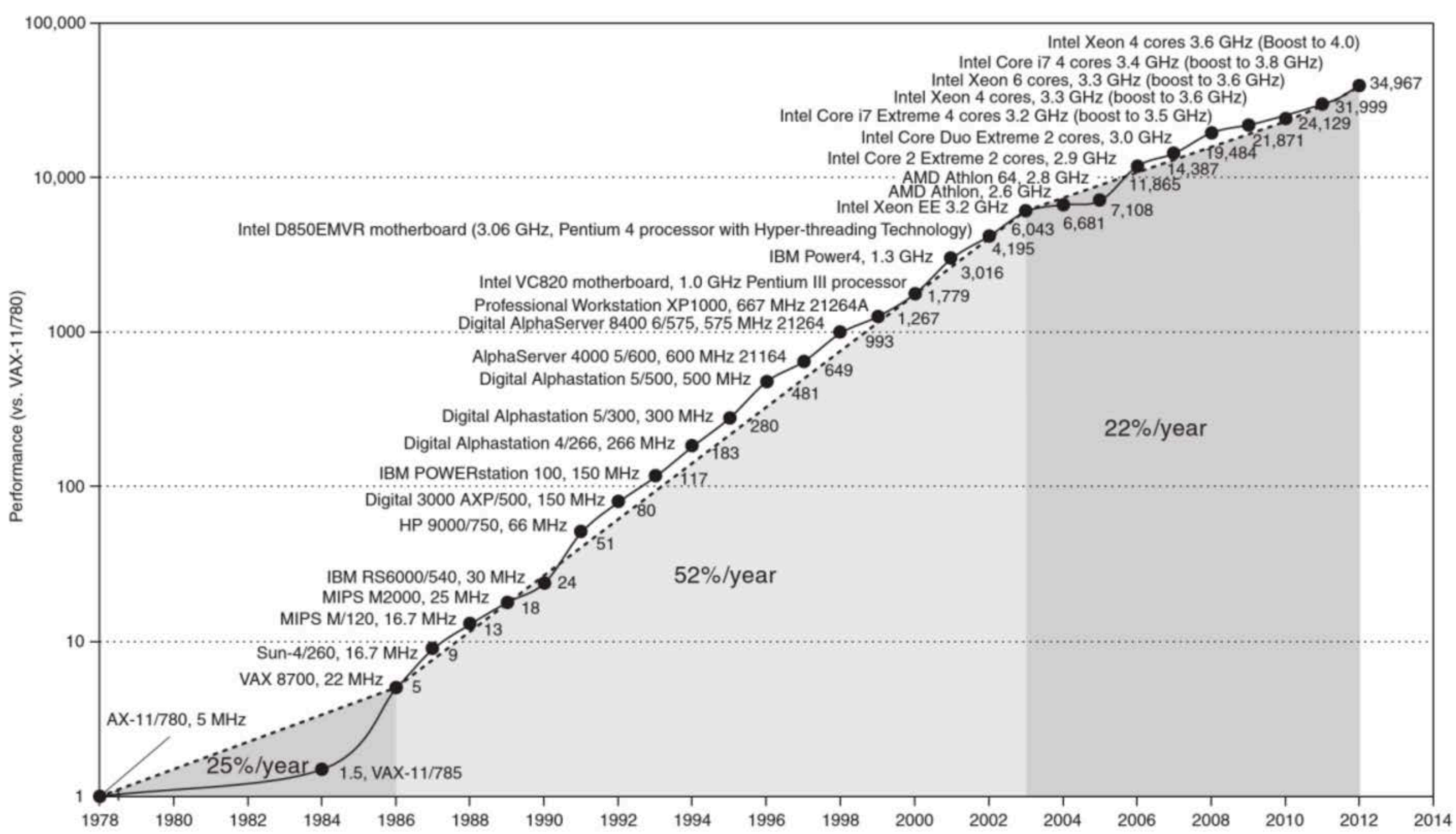
- **For CMOS (Complementary Metal Oxide Semiconductor, 互补金属氧化物半导体), the primary source of energy consumption is so-called dynamic energy: the energy is consumed when transistors switch their state**
- **The power required per transistor is**

Dyn power ∝ capacitance load × voltage² × frequency switched

- **Voltage are decreasing (from 5V to 1V in 20 years), but the number of transistors and frequency are increasing at a faster rate (by a factor of 1000)**
- **Unfortunately, today's problem is further lowering of the voltage makes the transistors to leaky. E.g., about 40% of the power consumption in server chips is due to leakage.**
  - **Cooling**
  - **Turn off parts of the chip in a given clock cycle**

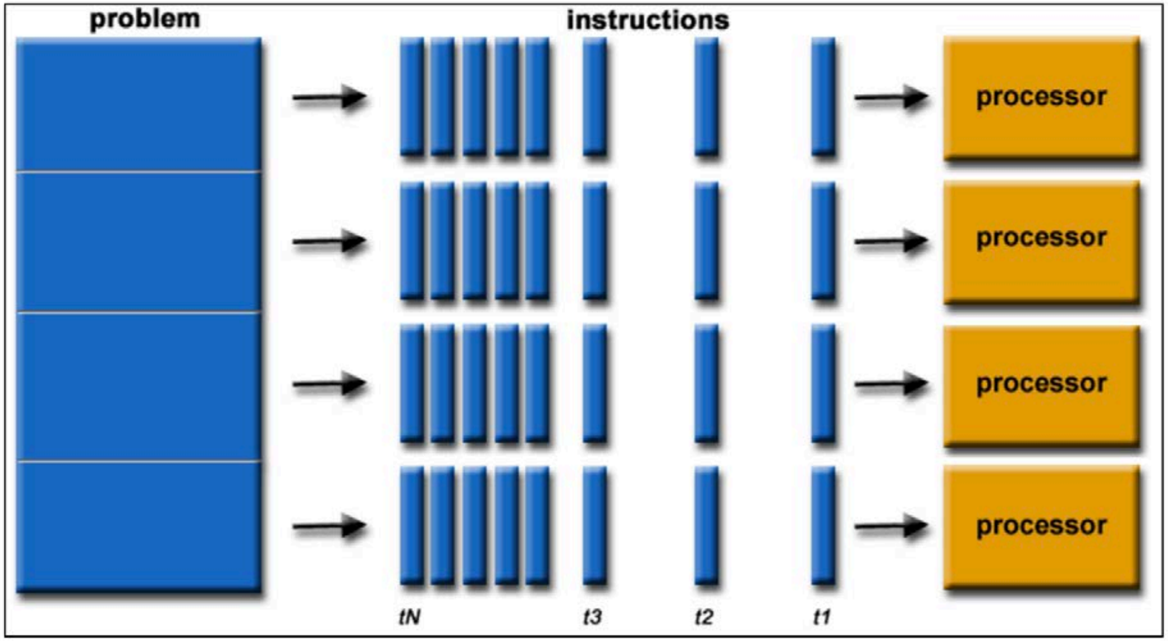
The sea change: the switch from uniprocessors to multiprocessors

- **Is there another way to improve the performance?**



Parallelism and pipelining

- Challenges
  - It is hard to write explicitly parallel programs
  - Load balance on each core
  - Appropriately scheduling sub-tasks (or instructions)
  - Communication and synchronization overhead



Benchmarking the Intel Core i7

- Benchmark: programs specifically chosen to measure performance
- SPEC (System Performance Evaluation Cooperative)
  - E.g., SPEC CPU2006 consisting of a set of 12 integer benchmarks (CINT2006) and 17 floating-point benchmarks (CFP2006)

Description	Name	Instruction Count x 10^9	CPI	Clock cycle time (seconds x 10^{-9})	Execution Time (seconds)	Reference Time (seconds)	SPECratio
Interpreted string processing	perl	2252	0.60	0.376	508	9770	19.2
Block-sorting compression	bzip2	2390	0.70	0.376	629	9650	15.4
GNU C compiler	gcc	794	1.20	0.376	358	8050	22.5
Combinatorial optimization	mcf	221	2.66	0.376	221	9120	41.2
Go game (AI)	go	1274	1.10	0.376	527	10490	19.9
Search gene sequence	hmmer	2616	0.60	0.376	590	9330	15.8
Chess game (AI)	sjeng	1948	0.80	0.376	586	12100	20.7
Quantum computer simulation	libquantum	659	0.44	0.376	109	20720	190.0
Video compression	h264avc	3793	0.50	0.376	713	22130	31.0
Discrete event simulation library	omnetpp	367	2.10	0.376	290	6250	21.5
Games/path finding	astar	1250	1.00	0.376	470	7020	14.9
XML parsing	xalancbmk	1045	0.70	0.376	275	6900	25.1
Geometric mean	–	–	–	–	–	–	25.7

FIGURE 1.18 SPECINTC2006 benchmarks running on a 2.66 GHz Intel Core i7 920. As the equation on page 35 explains, execution time is the product of the three factors in this table: instruction count in billions, clocks per instruction (CPI), and clock cycle time in nanoseconds. SPECratio is simply the reference time, which is supplied by SPEC, divided by the measured execution time. The single number quoted as SPECINTC2006 is the geometric mean of the SPECratios.

SPEC Power Benchmark

- It reports power consumption of servers at different workload levels, divided into 10% increments, over a period of time

Target Load %	Performance (ssj_ops)	Average Power (watts)
100%	865,618	258
90%	786,688	242
80%	698,051	224
70%	607,826	204
60%	521,391	185
50%	436,757	170
40%	345,919	157
30%	262,071	146
20%	176,061	135
10%	86,784	121
0%	0	80
Overall Sum	4,787,166	1922
$\sum$ssj_ops / $\sum$power =		2490

FIGURE 1.19 SPECpower_ssj2008 running on a dual socket 2.66 GHz Intel Xeon X5650 with 16 GB of DRAM and one 100 GB SSD disk.

- CPU performance is dependent upon three characteristics:
  - clock cycle (or rate)
  - clock cycles per instruction
  - and instruction count.
- It is difficult to change one parameter in complete isolation from others because the basic technologies involved in changing each characteristic are interdependent:

Thanks