

COMP2611: Computer Organization

Introduction (For L2 students)

Course's homepage <http://course.cse.ust.hk/comp2611>

Lecture

4:30-5:50, Wed/Fri, Room 2502 (Lift 25/26)

Instructor: Cindy LI Xin lixin@cse.ust.hk

Office: RM 3535

You also need to attend **Tutorials and **Labs**, which are ~~useful~~ **necessary** supplements to lectures**

Reading the textbook is also a very important part in the workflow of this course.

Course Facebook: search **HKUST CSE COMP2611 Spring 2015**

❑ Grading

- **2 Quizzes 15% (2 x 7.5%)**
- **Course Project 15%**
- **Midterm Exams 30%**
 - **Mar 21 (Sat) 10:00am**
- **Final Exam 40%**

❑ Policies

- **Course project should be individual work; both 'provider' and 'copier' will be penalized equally and harshly**
- **Skipping the midterms or final examination without prior approval will automatically lead to an "F" grade for the course**

- I II III ~~IIII~~ (herringbone) - T F F F F

- ❑ When dealing with a **size** (e.g., Memory or file)

- Kilo – 2^{10} or 1024
- Mega – 2^{20} or 1024 Kilo
- Giga – 2^{30} or 1024 Mega
- Tera – 2^{40} or 1024 Giga
- Peta – 2^{50} or 1024 Tera
- ...

Example:

- The memory in my computer is 4 Gigabytes
- The PPT file for this lecture is 2.5 Megabytes

- ❑ When dealing with a **rate/frequency** (e.g., # instructions per second, # clock ticks per second)

- Kilo – 10^3 or 1000
- Mega – 10^6 or 1000 Kilo
- Giga – 10^9 or 1000 Mega
- Tera – 10^{12} or 1000 Giga
- Peta – 10^{15} or 1000 Tera
- ...

Example:

- The speed of my network card is 1 Gigabit per second
- The speed of my Intel processor is 2.89 Gigahertz

Computers have led to a **third revolution** for civilization:
agricultural -> industrial -> **information**

❑ **Desktop computers:**

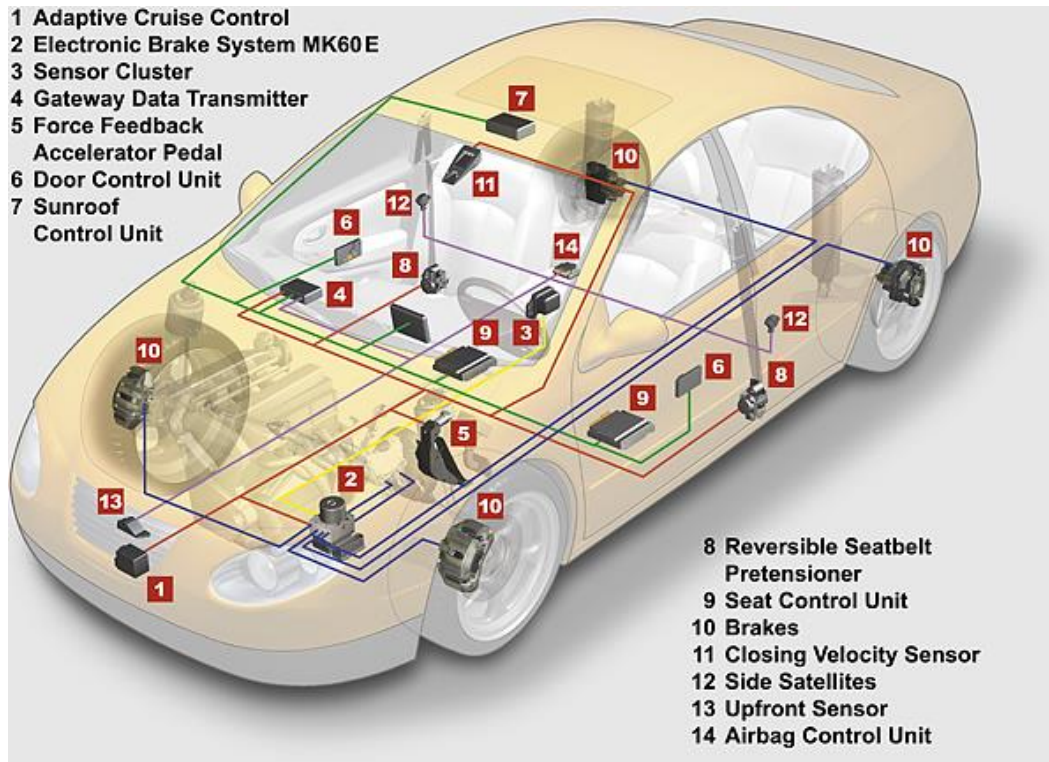
- Run a variety of general purpose software
- Designed to achieve good performance at low cost

❑ **Embedded computers:**

- Usually hidden as a component of a system (e.g., mobile phone, cars, airplanes, ATM machines, Smart card, ...)
- Run a predefined program
- Subject to a stringent power/performance/cost constraint

❑ **Servers and Networked computers:**

- High storage and computing capacity, performance and reliability
- Used to run large programs for multiple users
- Only accessible via a network
- Range from small servers to building sized, to several thousand computers in a grid

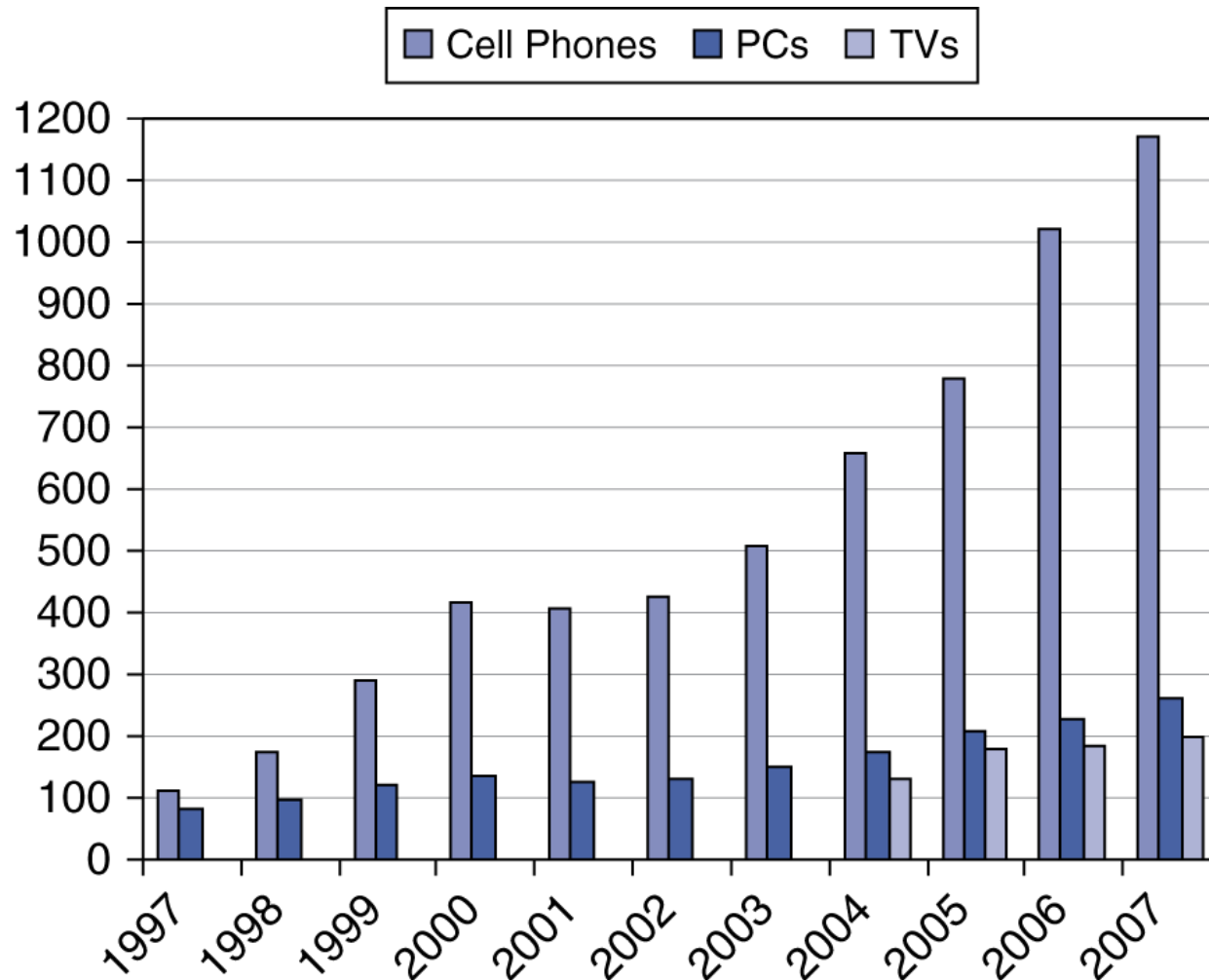


❑ Examples of embedded system



❑ Examples of server

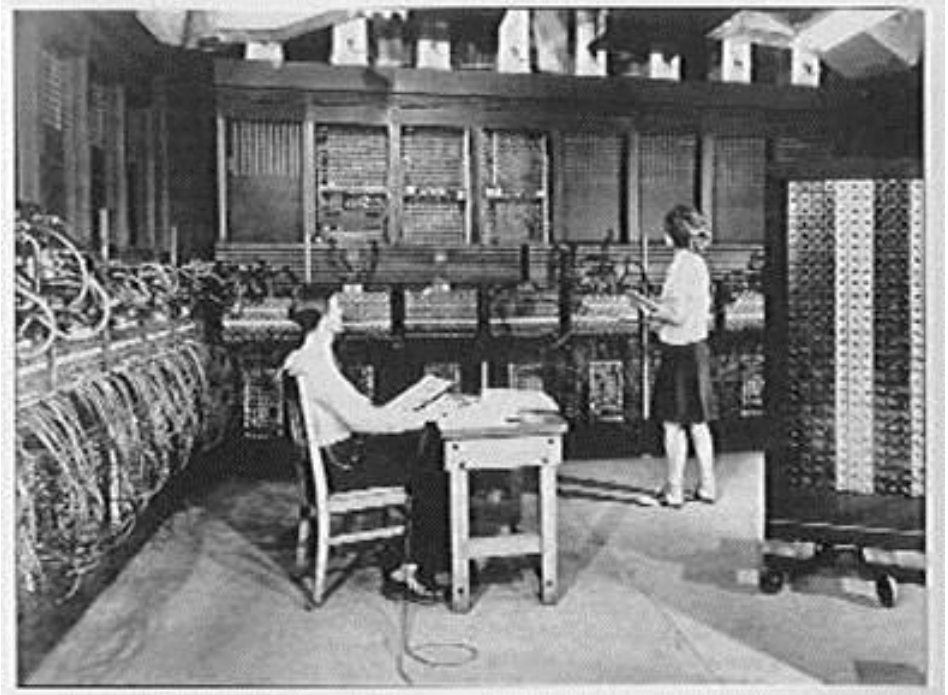
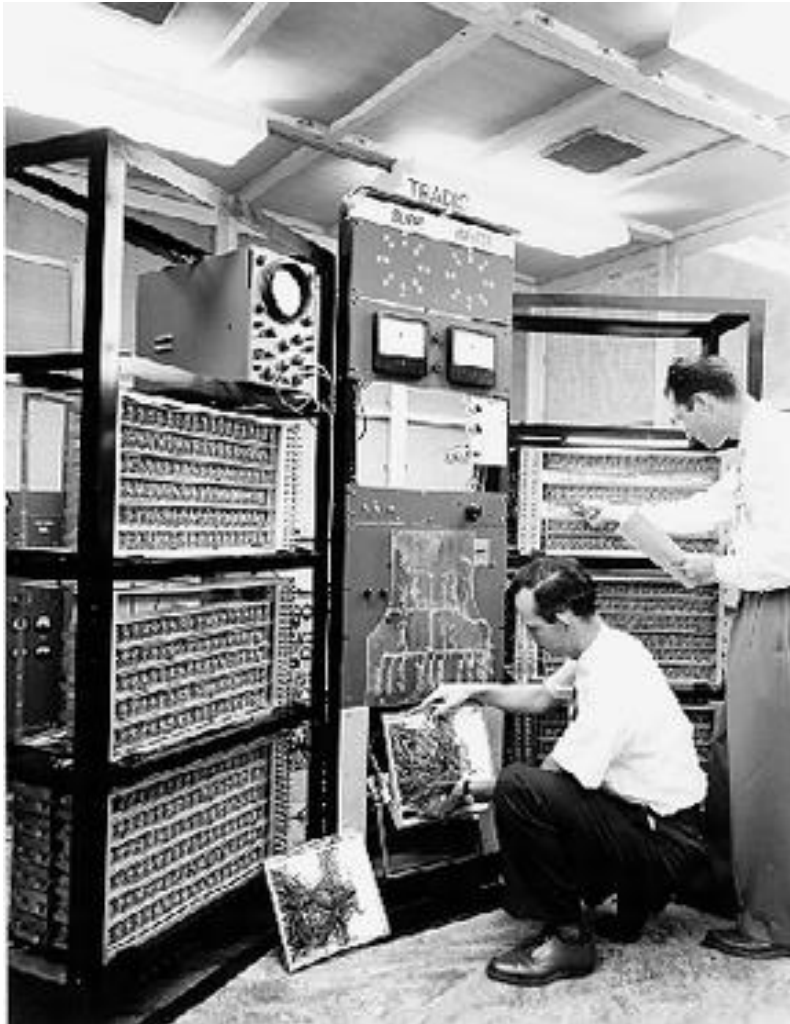
□ Number of Distinct Processors sold between 1997 and 2007 in millions



Why you're Here?

- ☐ Why do you take COMP2611?
- ☐ What have you heard about COMP2611 from senior students?
- ☐ What do you expect to learn from COMP2611?

Computer in the Oooooooooooooold Days



What did the Programmer Do?

```

00000000101000010000000000011000
00000000100011100001100000100001
10001100011000100000000000000000
100011001111001000000000000000100
10101100111100100000000000000000
101011000110001000000000000000100
00000011111000000000000000001000
    
```

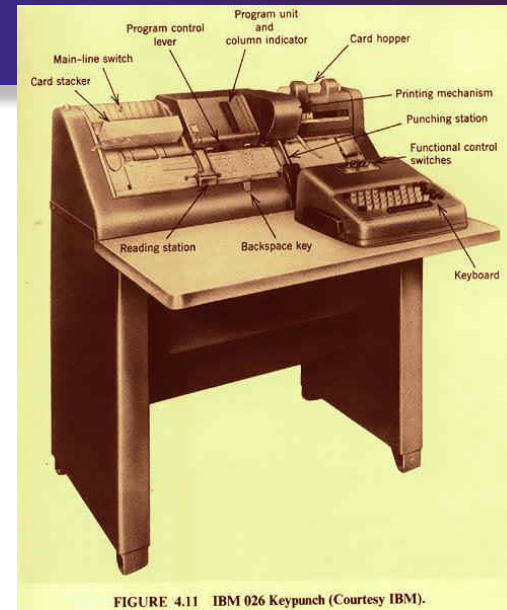
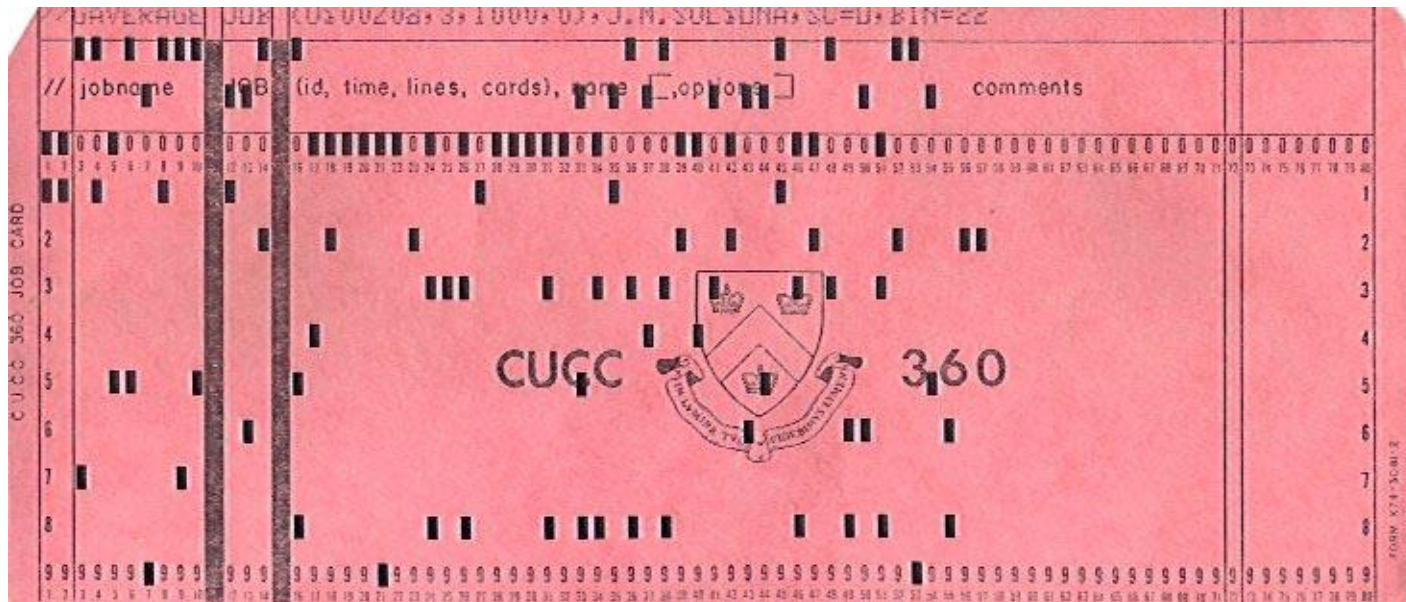


FIGURE 4.11 IBM 026 Keypunch (Courtesy IBM).



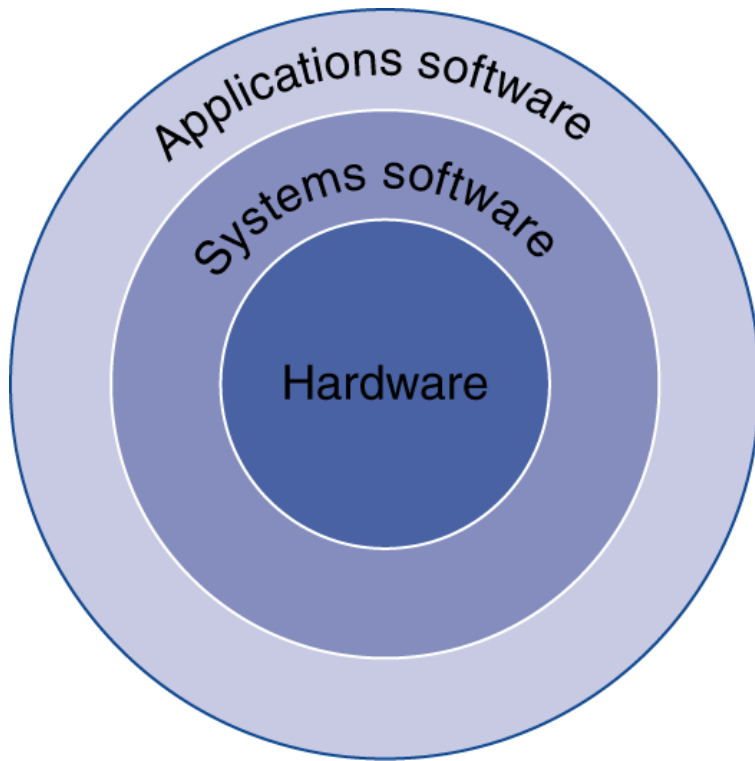
How does computer understand your instruction?



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

- ❑ How programs are translated from high level programming language to machine language
- ❑ How the hardware executes programs written in machine language
- ❑ The interface between the hardware and the software or the Instruction Set Architecture (ISA)
- ❑ What determines the performance of a program and how it can be improved
- ❑ How hardware designers improve the performance

- ❑ How to measure and analyze computer performance
 - To tell why a design is good or bad – Chapter 1
- ❑ How computers work
 - Computer Arithmetic and implementation – Chapter 3
 - Issues affecting design of modern processors – Chapters 2, 4 (and 7)
 - Exploiting memory hierarchy – Chapter 5
 - I/O – Chapter 6 (depending on the progress)



❑ Application software

- Written in high-level language
- Ex: Comp2011 assignment written in C++

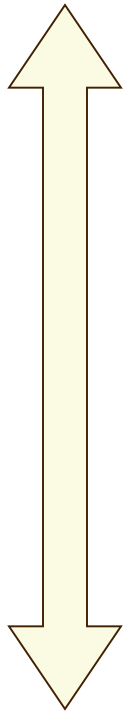
❑ System software

- **Compilers**: translates HLL code to machine code
- **Operating System**: service code
 - Handle input/output
 - Manage memory and storage
 - Schedule tasks & share resources

❑ Hardware

- Processor,
- memory,
- I/O controllers

for human



for machine

High-level language
program (in C)

Assembly (low-level)
language program (for MIPS)

Binary machine language
program (for MIPS)

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

C compiler

```
swap:
    muli $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

```
000000001010000100000000000011000
00000000100011100001100000100001
10001100011000100000000000000000
10001100111100100000000000000100
10101100111100100000000000000000
10101100011000100000000000000100
00000011111000000000000000001000
```

❑ High-level language

- Level of abstraction closer to the problem domain
- Helps increase productivity, portability and simplify debugging

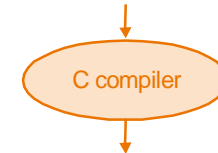
❑ Assembly language

- Binary instructions represented in symbolic notation
- One to one mapping with binary instructions
- Assemblers translate from Assembly language to machine language

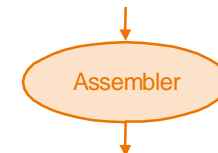
❑ Hardware representation

- Computers only deal with binary digits (bits)
- Instructions and data are encoded as bit strings

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

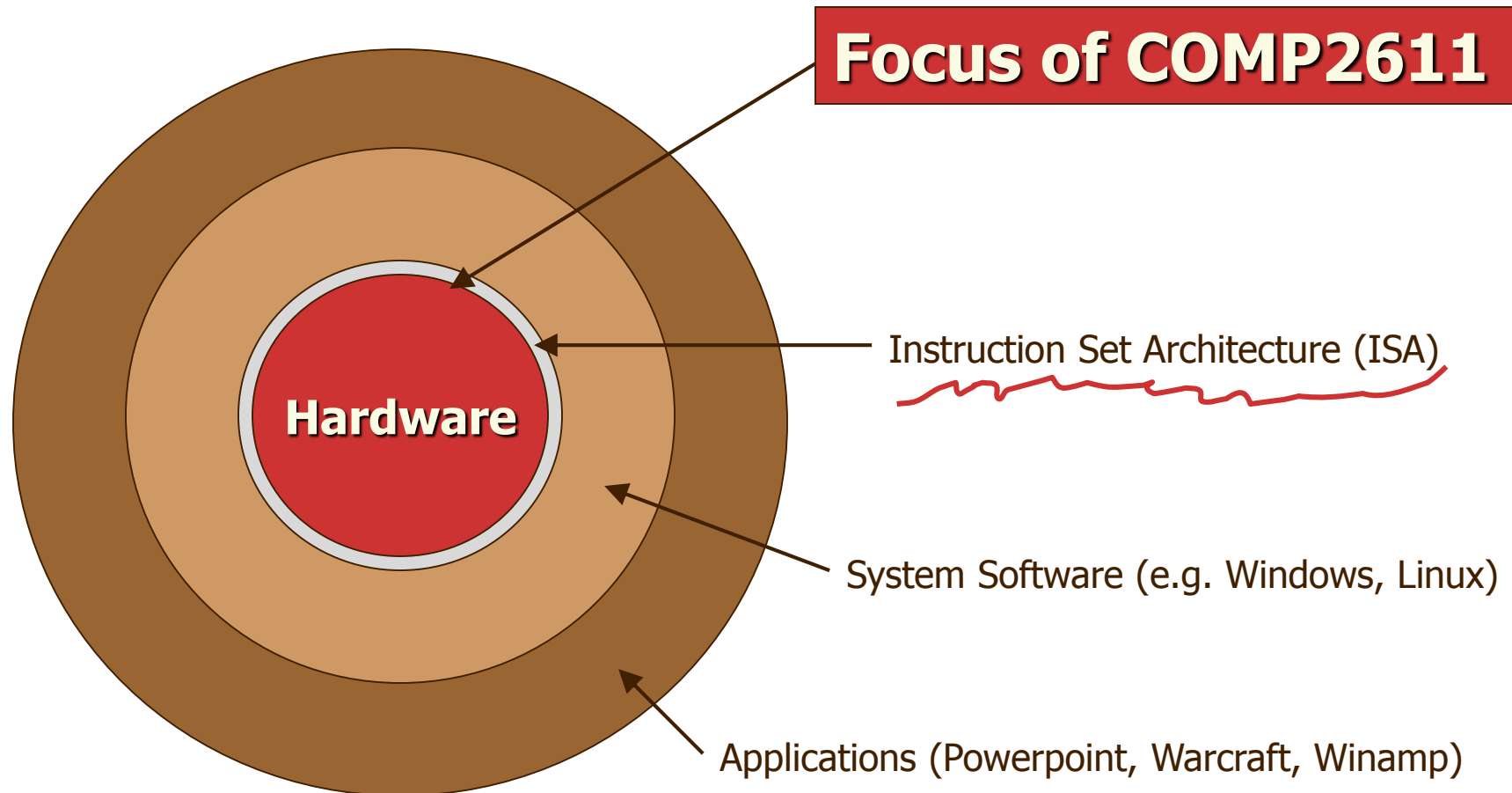


```
swap:
    muli $2, $5, 4
    add $2, $4, $2
    lw $15, 0($2)
    lw $16, 4($2)
    sw $16, 0($2)
    sw $15, 4($2)
    jr $31
```



```
000000001010000100000000000011000
00000000100011100001100000100001
10001100011000100000000000000000
10001100111100100000000000000100
10101100111100100000000000000000
10101100011000100000000000000100
00000011111000000000000000001000
```

Impossible to understand computer components by looking at every single transistor. Instead, **abstraction** is needed.



❑ Key ideas:

- Both hardware and software are organized into **hierarchical layers**.
- Hierarchical organization helps to cope with system **complexity**.
- Lower-level details are **hidden** to offer a simpler view at the higher levels.
- Interaction between levels occurs only through well-defined **interface**.

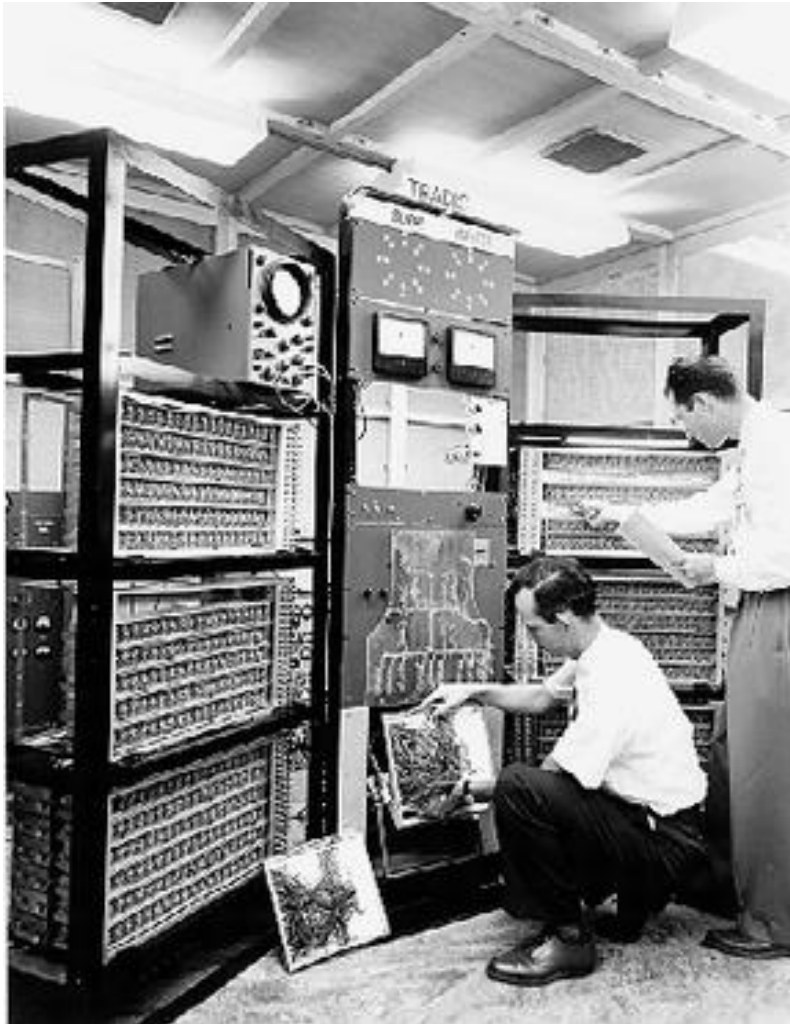
❑ Example:

- Interface between hardware and software: Instruction set architecture (ISA)

An **instruction set architecture (ISA)** provides an **abstract interface** between hardware and low-level software.

- ❑ **Advantage**: allows different implementations of varying cost and performance to follow the same instruction set architecture (i.e., to **run the same software**).
 - Example: 80x86, Pentium, Pentium II, Pentium III, Pentium 4 all implement the same ISA
- ❑ Some instruction set architectures:
 - **80x86/Pentium/K6** (offers different implementations)
 - **MIPS**
 - **PowerPC**
 - **ARM**
 - **HP PA-RISC**
 - **Sparc**

Anything in Common?



Five Basic Components (all kinds of computers)

❑ **Input:**

- To communicate with the computer
- Data and instructions transferred to the memory

❑ **Output:**

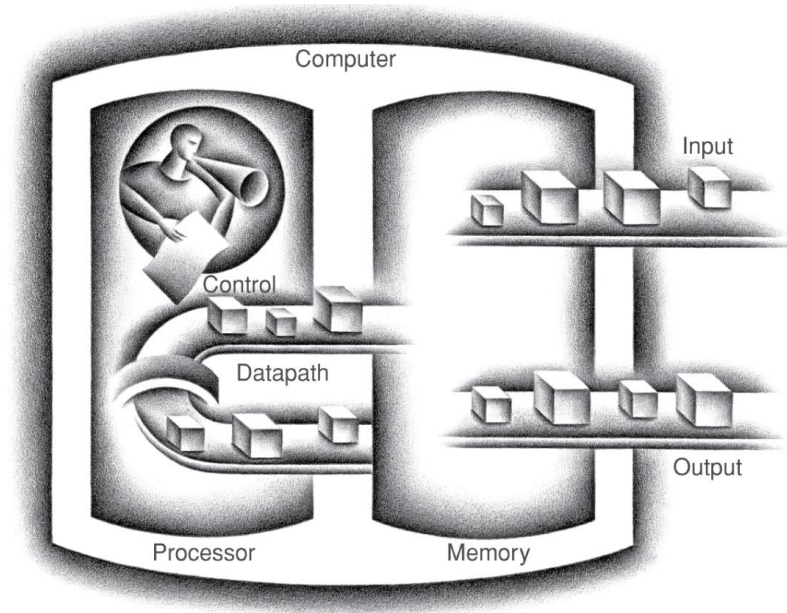
- To communicate with the user
- Data is read from the memory

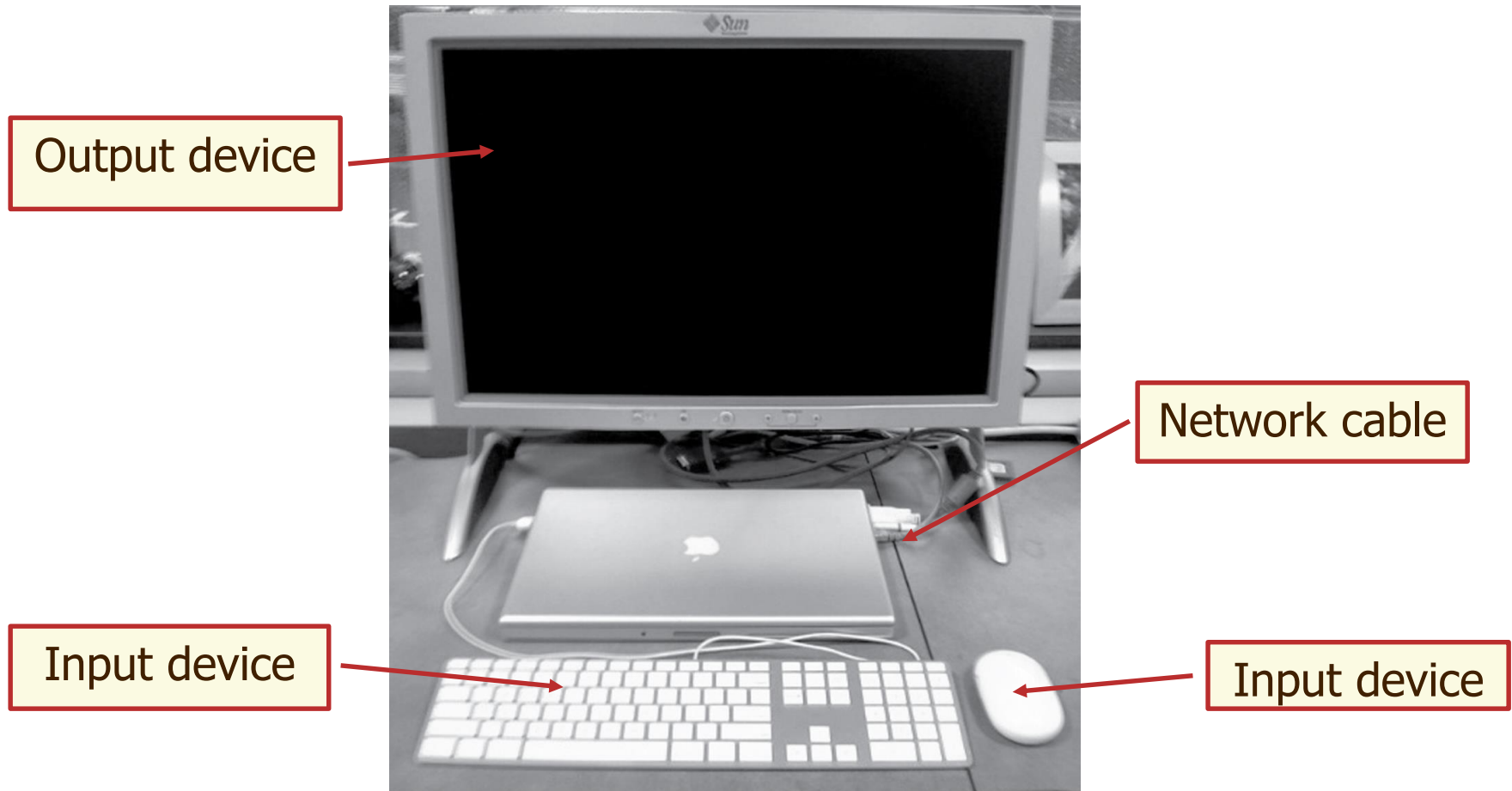
❑ **Memory:**

- Large store to keep instructions and data

❑ **Processor**, which consists of:

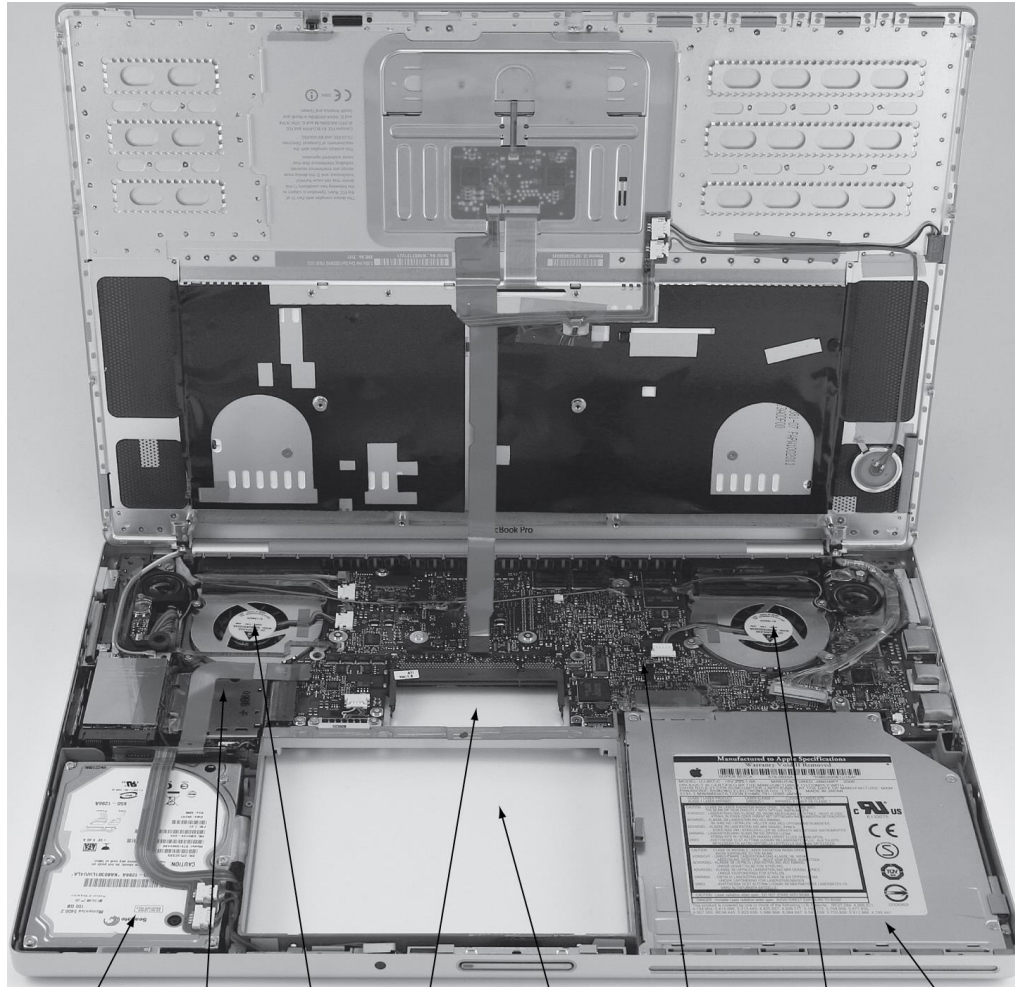
- **Datapath**: processes data according to instructions.
- **Control**: commands the operations of input, output, memory, and datapath according to the instructions.



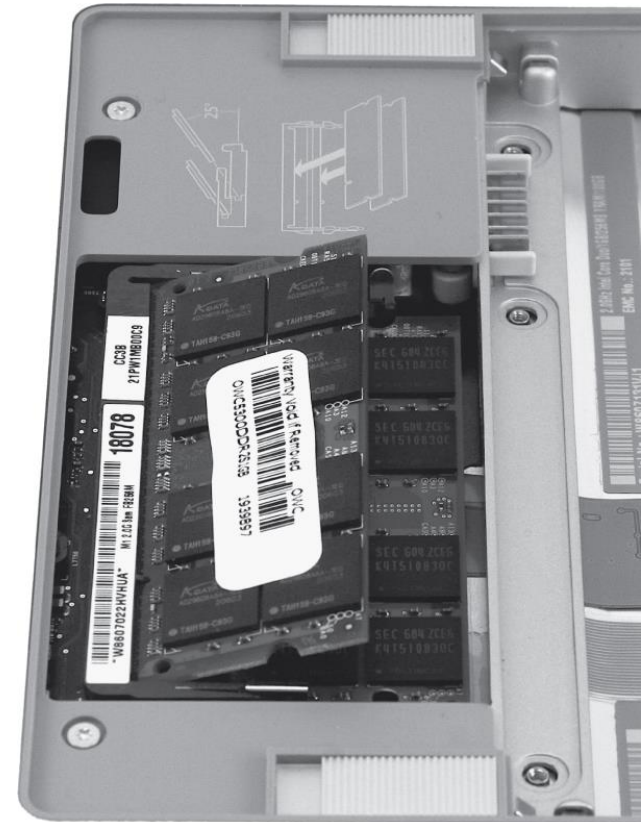


Opening the Box: in My College Days

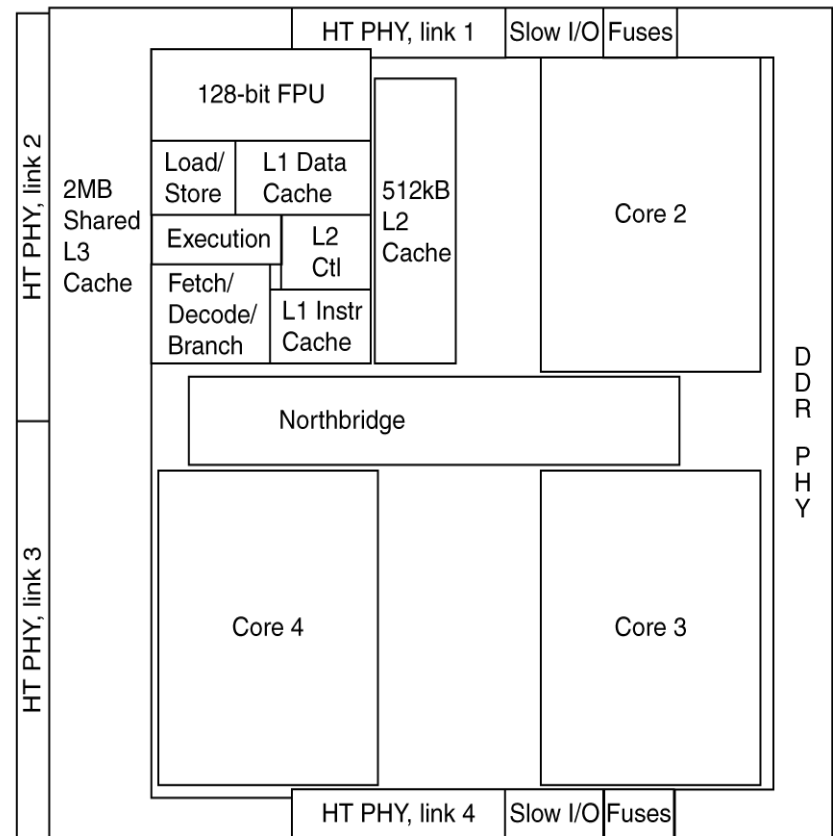
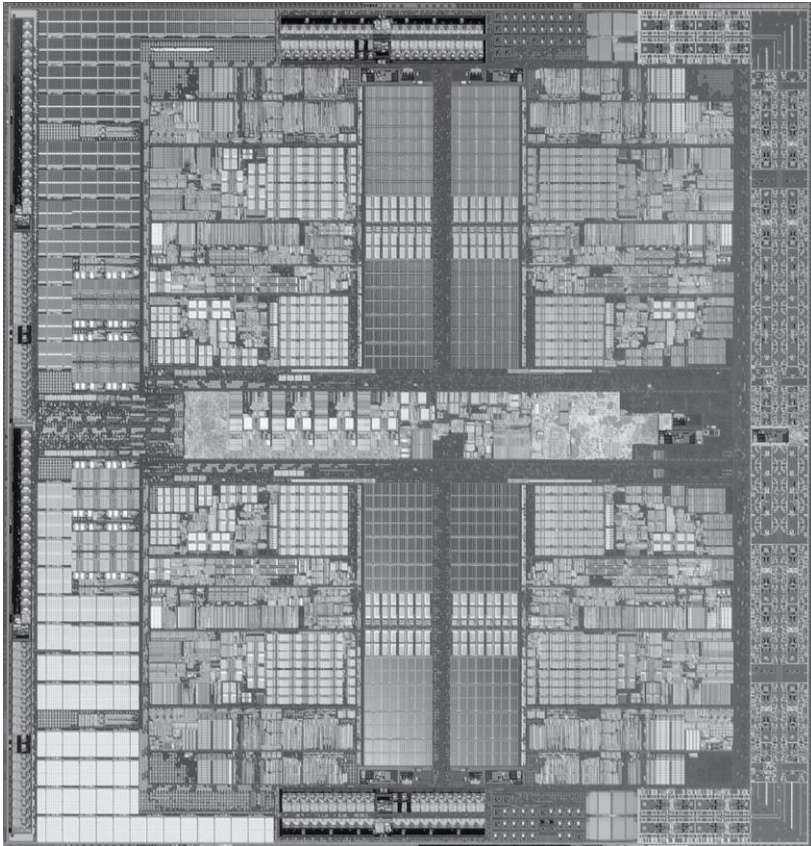




Hard drive Processor Fan with cover Spot for memory DIMMs Spot for battery Motherboard Fan with cover DVD drive



- ❑ AMD Barcelona: 4 processor cores

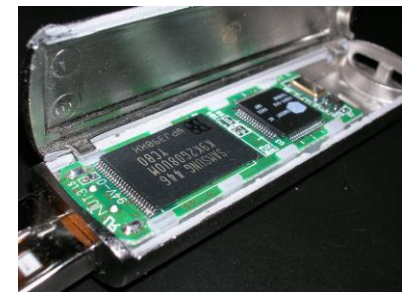


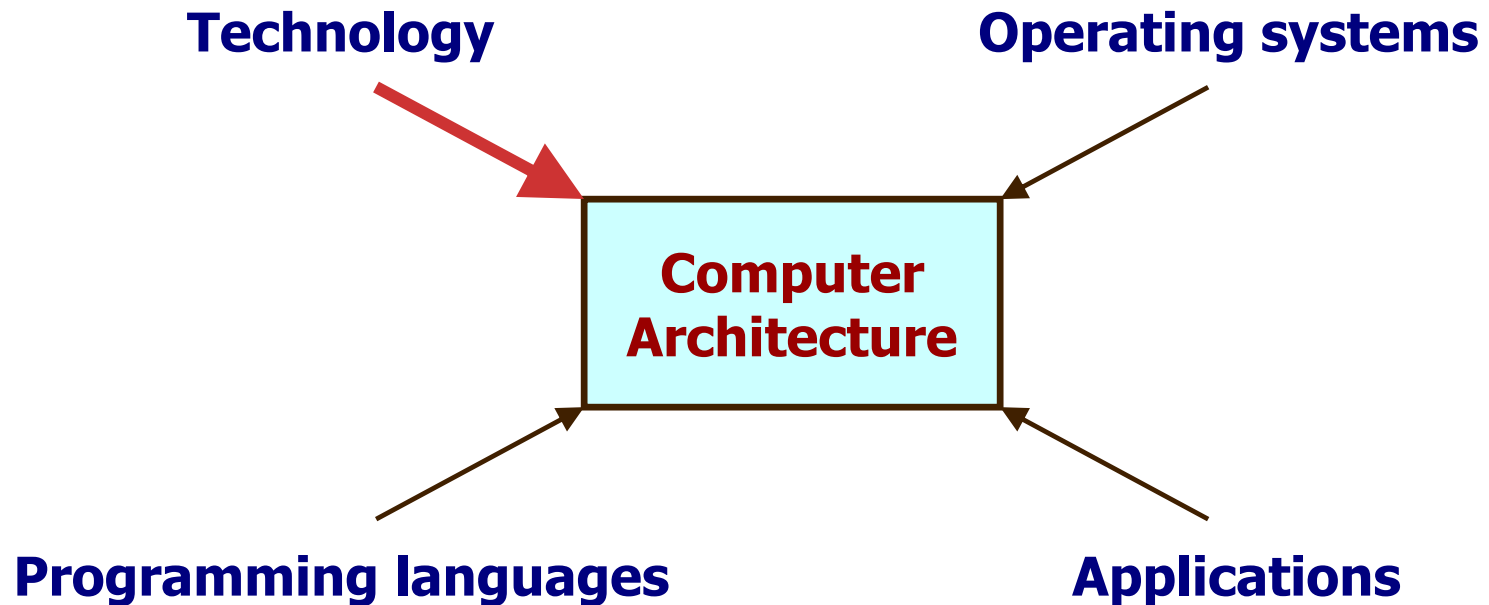
❑ Volatile main memory (RAM)

- Used by the processor to store programs and data
- Loses instructions and data when powered off

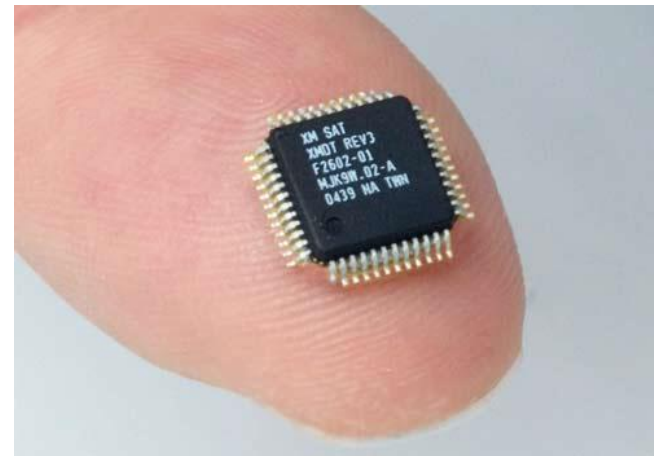
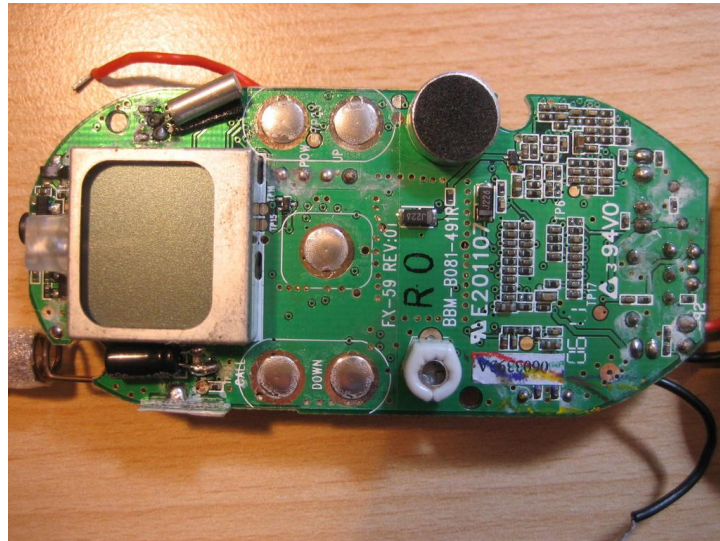
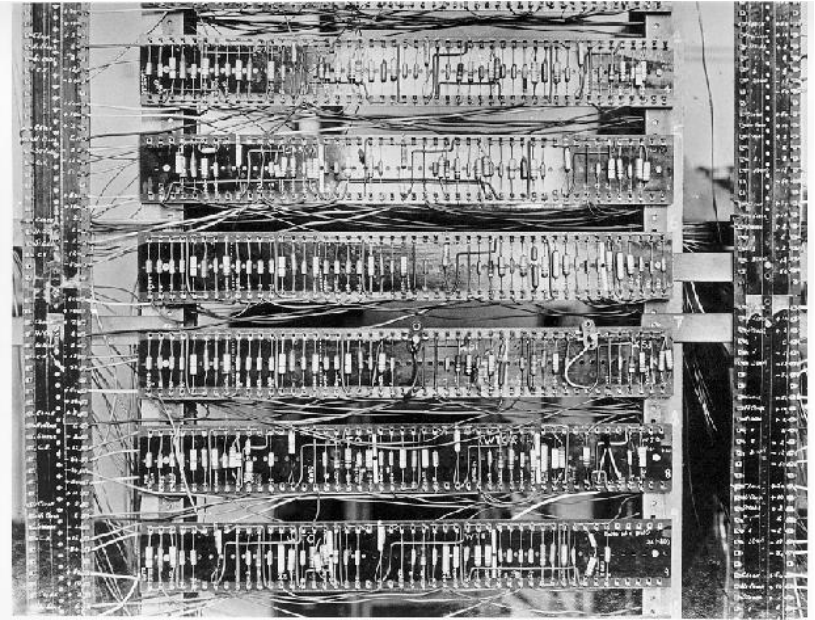
❑ Non-volatile secondary memory

- Magnetic disk
- Flash memory
- Optical disk (CDROM, DVD)





What are These Technologies?



- ❑ Increased capacity and performance
- ❑ Reduced cost

❑ Processor:

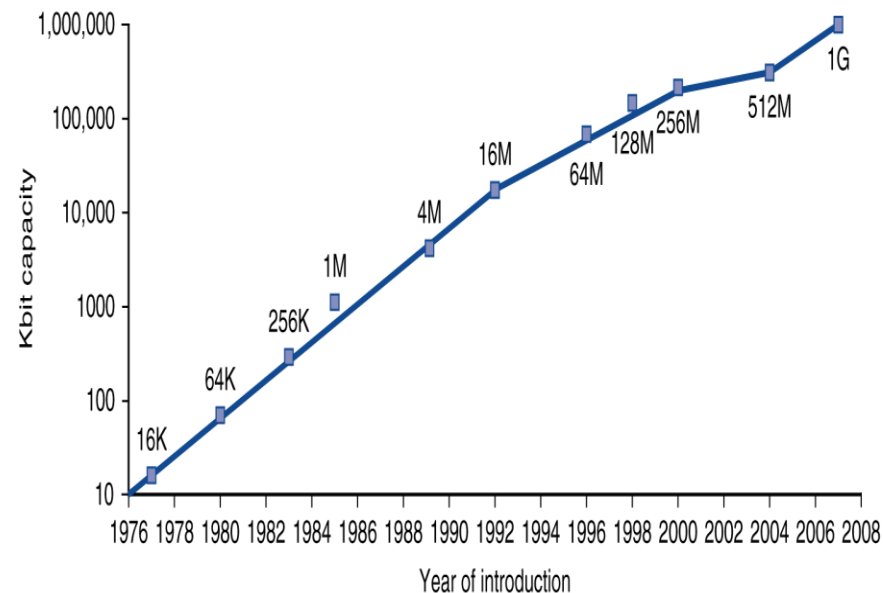
- **Logic capacity:** ~30% per year
- **Clock rate:** ~20% per year

❑ Memory:

- **DRAM capacity:** ~60% per year (or ~4X every 3 years)
- **Memory speed:** ~10% per year
- **Cost per bit:** decreases ~25% per year

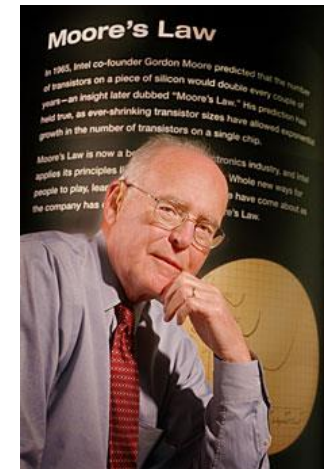
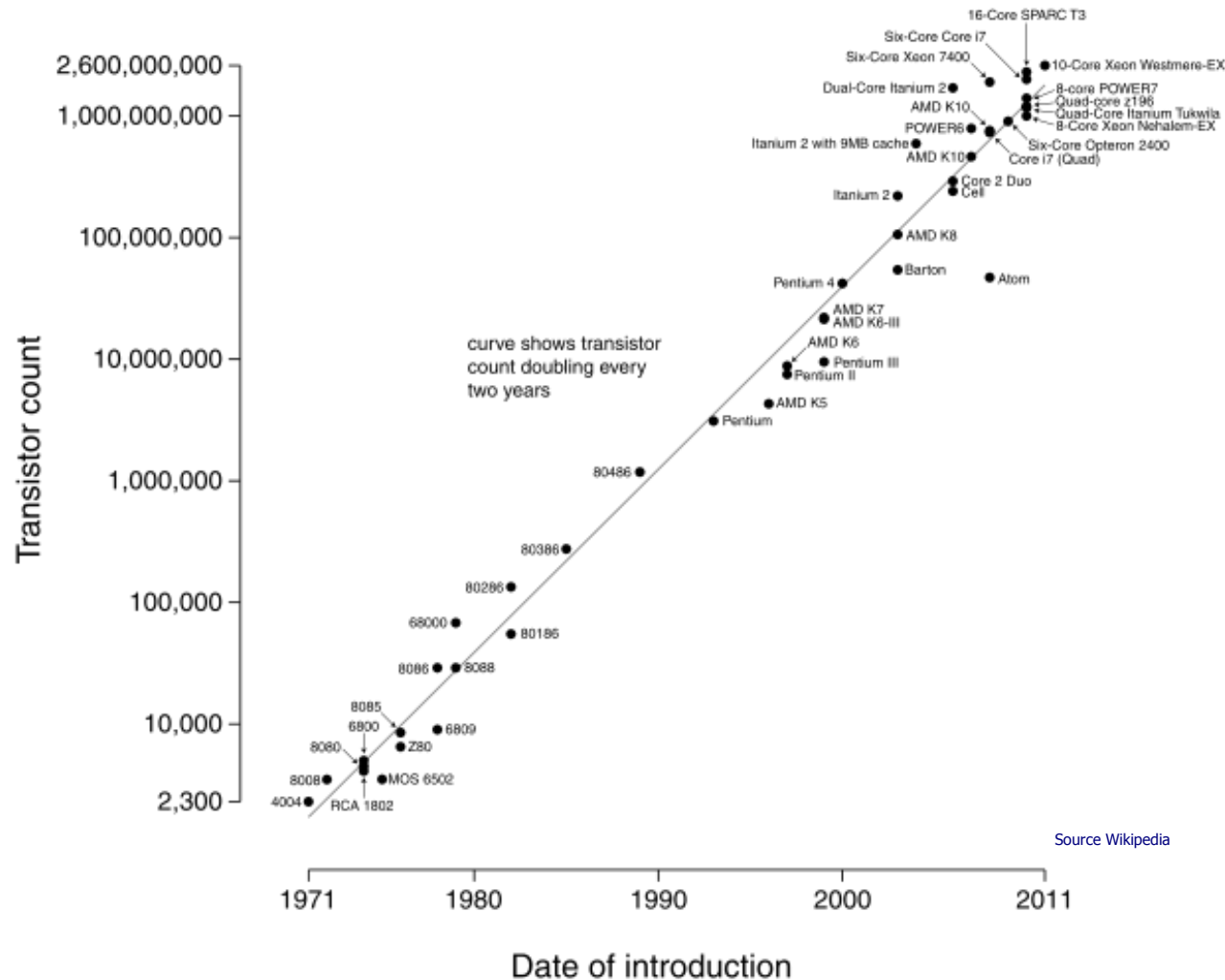
❑ Disk:

- **Capacity:** ~60% per year



Year	Technology used in computers	Relative performance per unit cost
1951	Vacuum tube	1
1965	Transistor	35
1975	Integrated circuit (IC)	900
1995	Very large scale integrated (VLSI) circuit	2,400,000
2005	Ultra large scale integrated circuit	6,200,000,000

Microprocessor Transistor Counts 1971-2011 & Moore's Law



- ❑ Five basic components of a computer
 - **input, output, memory, processor** (**datapath** + **control**)
- ❑ **Principle of abstraction**
 - Help cope with design complexity by hiding low level details
- ❑ **Instruction set architecture**
 - Important abstraction interfaces hardware with low-level software