Principle of Microcomputer and Interface Technology

School of Automation Science and Electrical Engineering

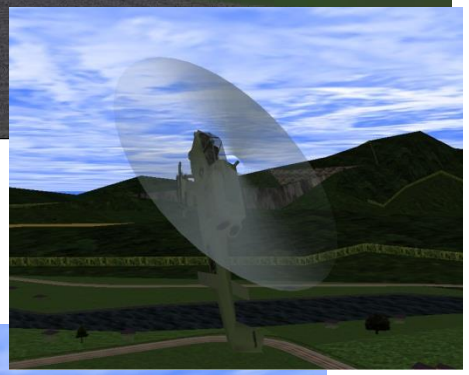
Dr. Prof. Ni LI

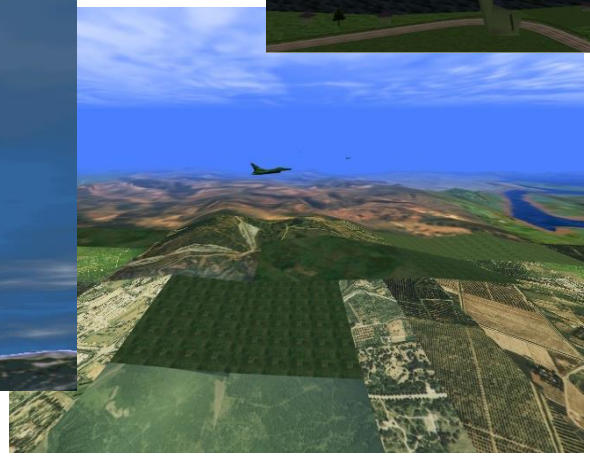
Self-introduction

- Beihang University： Ph.D, Full Professor, Ph.D Advisor
- Visiting scholar: Old Dominion University, Norfolk, USA
- Research field
  - System Modeling & Simulation Technology → Computer Application Technology
- Modern Modeling & Simulation Technology
  - **Comprehensive/Integrated Technology**
    - Construction of Simulation Systems by modeling
    - Method:
      - ***Research, Analysis, Experiment, Simulation & Evaluation of exited or imaged systems***
    - Tools：
      - ***Computer systems, physical equipment related with application and simulators***

Application of M&S technology
-Application of computer technology

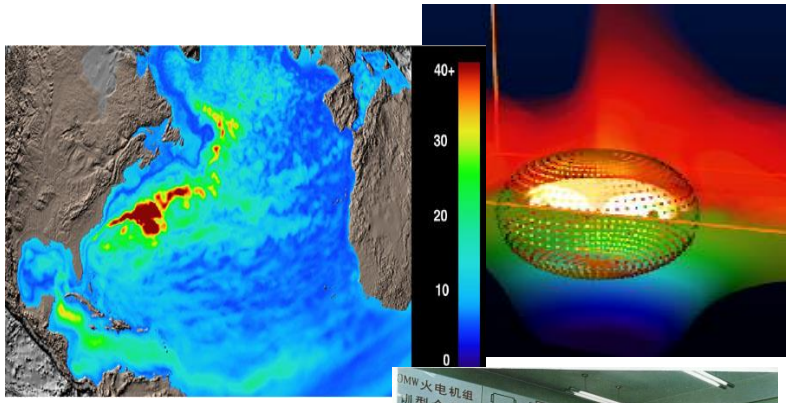
- **National defense**
  - **R&D of weapon system**
  - **Military training**
  - **Analysis of advanced concepts and military needs**

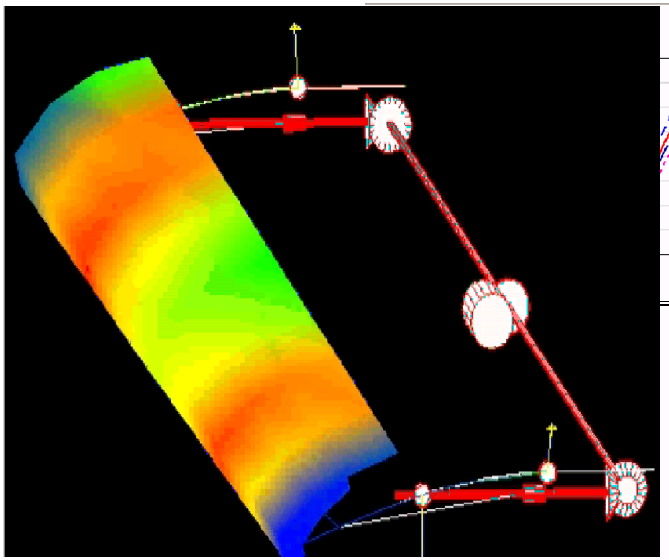


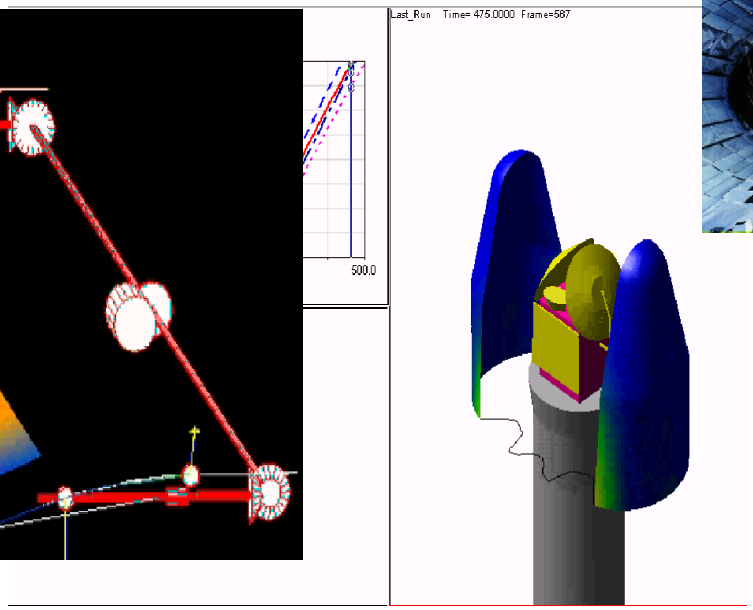


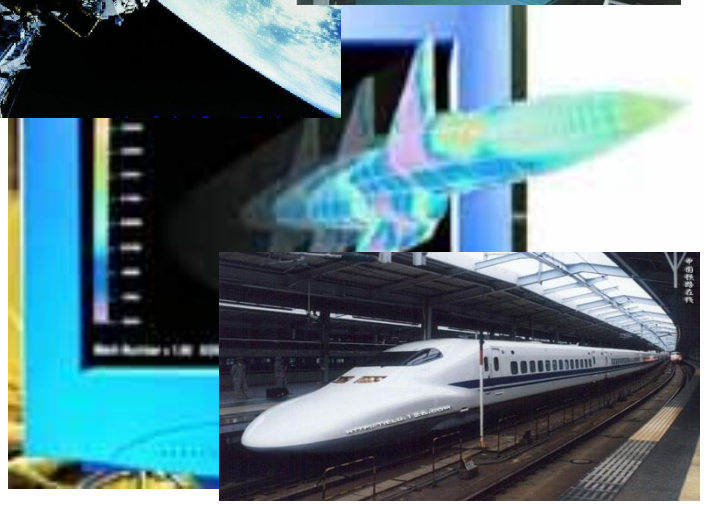
Application of M&S technology
-Application of computer technology

- **National economy**
  - **Electricity, nuclear power**
  - **Virtual manufacturing**
- **Other application area**
  - **Transportation, medicine**
  - **Education, entertainment**









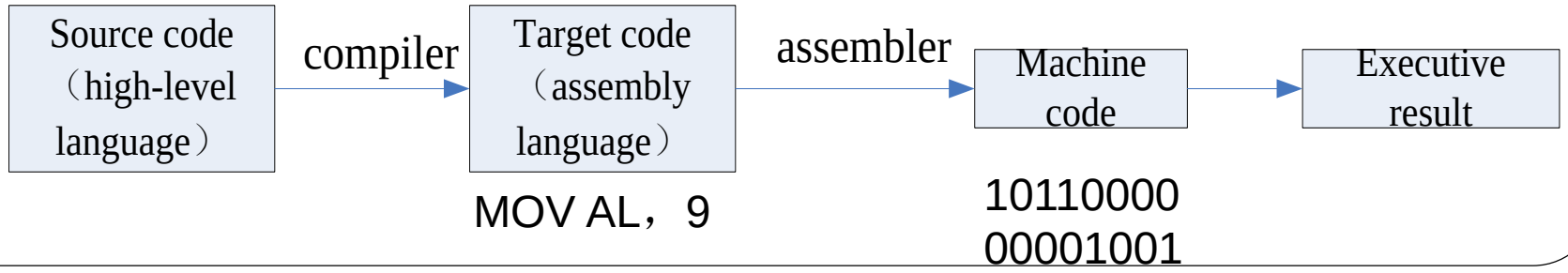
Application of computer technology

- Different fields
  - Industrial process control, intelligent robot, unmanned vehicle, computer-aided design and aided manufacturing
  - Automation

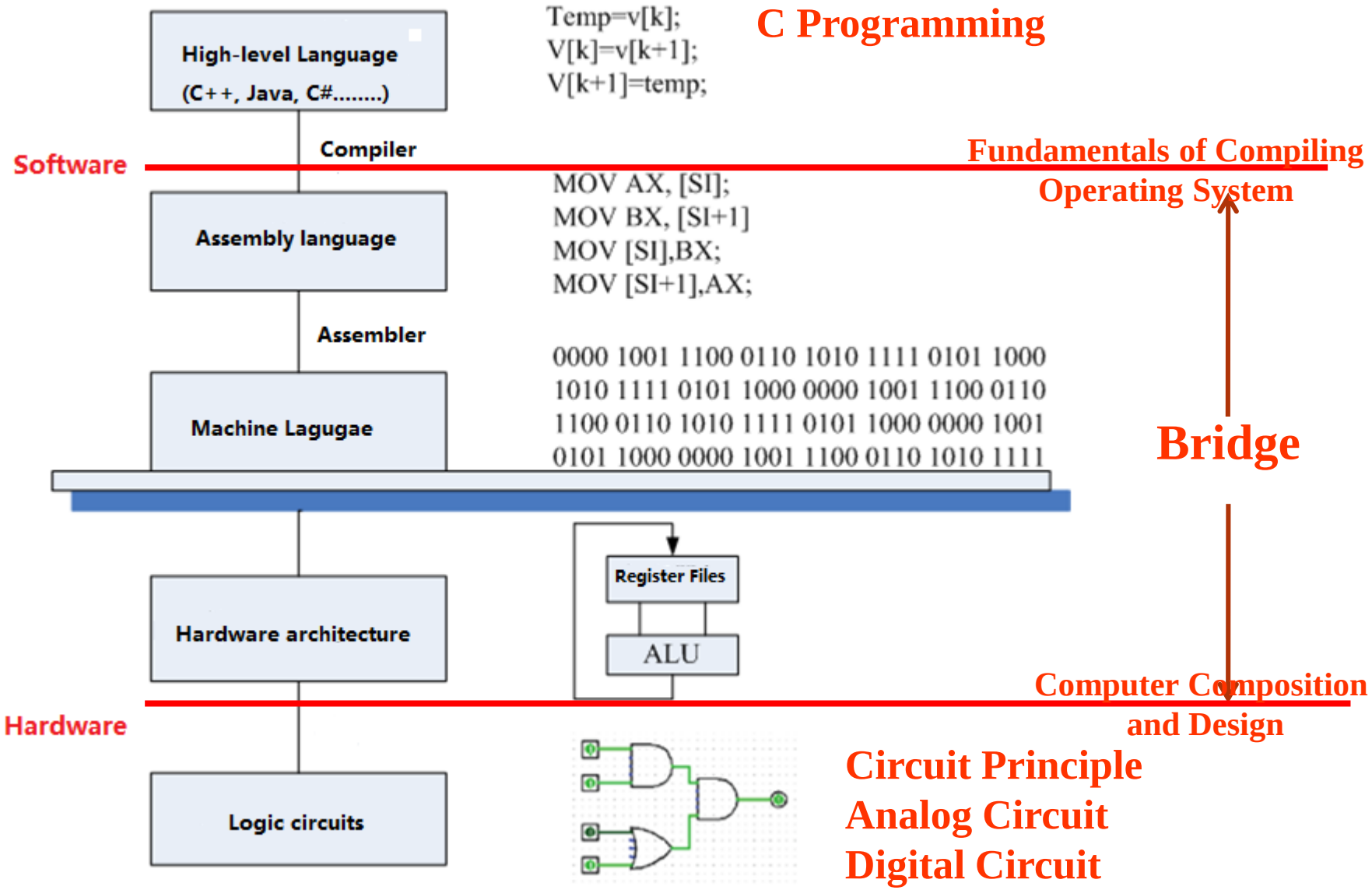
Principle of Microcomputer and interface technology !
Software & hardware interface

Syllabus

- Principle of microcomputer（chapter 1/2）
  - Computer composition
  - Basic principles of computer
  - CPU internal structure
- Software application（chapter 3/4）
  - Programing language
    - High-level: application-oriented, C/C++, Java, C#
    - Low-level: machine-oriented (hardware related), **Assembly language**, machine language
    - **For understanding principles and I/O interface control**
  - Addressing mode
  - CPU instruction set，assembler directive
  - Assembly language programing & debugging



Syllabus


High-level Language
(C++, Java, C#.....)
Compiler
Assembly language
Assembler
Machine Lagugae
Hardware architecture
Logic circuits
Software
Hardware
Temp=v[k];
V[k]=v[k+1];
V[k+1]=temp;
C Programming
MOV AX, [SI];
MOV BX, [SI+1]
MOV [SI],BX;
MOV [SI+1],AX;
0000 1001 1100 0110 1010 1111 0101 1000
1010 1111 0101 1000 0000 1001 1100 0110
1100 0110 1010 1111 0101 1000 0000 1001
0101 1000 0000 1001 1100 0110 1010 1111
Register Files
ALU
Fundamentals of Compiling
Operating System
Bridge
Computer Composition
and Design
Circuit Principle
Analog Circuit
Digital Circuit

Syllabus

- Hardware interface application (Chapter 6/7/8/9/10/11）
  - Interface technology (application of common interface chips)
  - Interrupt
  - Bus

- Class activities
  - PPT & blackboard-writing
  - Taking notes
  - Group discussions (8259A, 8251A)
  - Classroom quiz->After-class quiz in the course center
  - Homework, Midterm Exam
  - Attendance record

Curriculum

- Core curriculum, Optional courses for graduate entrance examinations
- Lectures + experiment (48 hours, 16 hours)
  - Software programing and hardware interface experiments
  - Late arrangement of time and place(starting from week 11)
- Exam
  - Final exam: 70%（Absence<=3）
  - Midterm Exam: 20% (Open Book, 8th week)
  - Everyday performance: 10% (Attendance & Homework & Quiz **-2** with each absence)
- Teaching assistant: graduate student
  - Homework: NMB D624

References

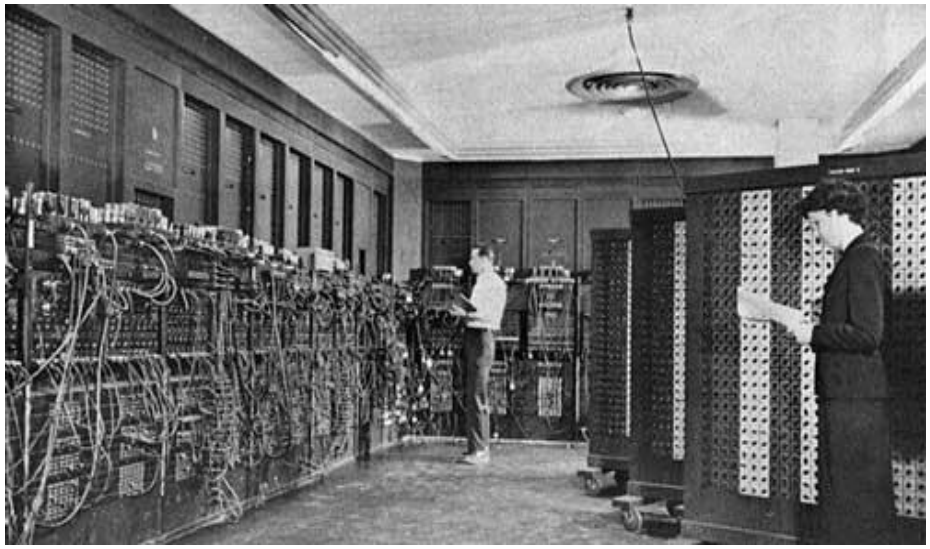
- Microprocessors and interfacing, Godse, 2009
- http://www-inst.eecs.berkeley.edu/~cs61c/sp10/ UC Berkeley CS61C Machine Structures
- http://pdos.csail.mit.edu/6.828/2009/index.html MIT 6.828: Operating System Engineering

- Q & A：
  - NMB E909
  - 82339016 lini@buaa.edu.cn

Chapter 1 Introduction

Development roadmap of computers

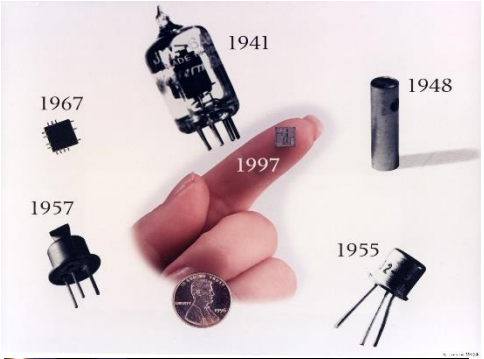
- Vacuum tube computer: The first electronic computer ERIAC, 1946, University of Pennsylvania, designed and manufactured by using 18800 tubes, and weighs 30 tons, covering an area of 150 square meters, with power consumption of 150 kilowatts, to complete 5000 addition operations per second
  - Military area：trajectory computation
  - Numerical Integration



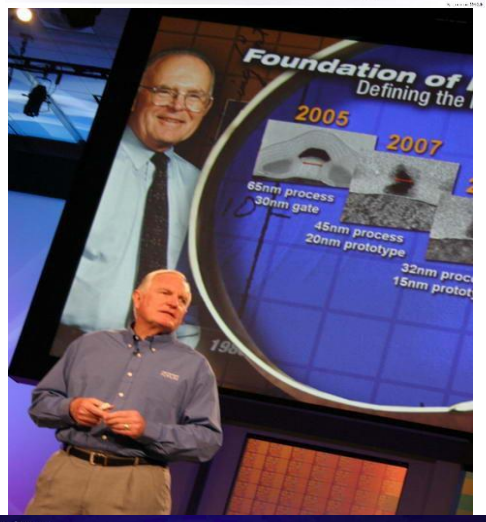
1946年

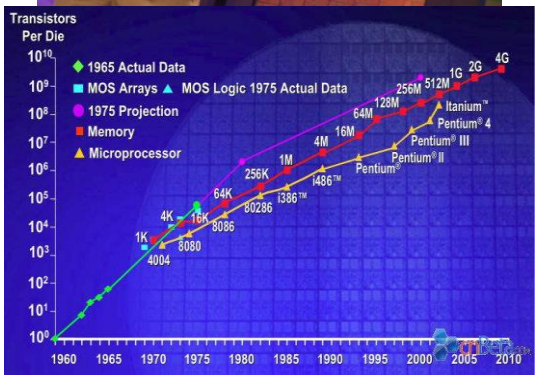
1958-1959年

Development roadmap of computers



- Transistorized computer: 1958, with transistors instead of vacuum tube, greatly reduces the cost and size of the computer, computational speed is greatly improved
  - Shockley etc. won the Nobel prize for physics by inventing transistors
- Small and medium-sized integrated circuit computer: 1965, reducing size, coupled with the operating system, and greatly improve the computational performance (scientific fields)
- Super LSI computer: 1970, micro-computer became typical representative of the fourth generation of computer
- Moore's Law: integration of integrated circuits (the number of transistors on a chip),18-24 months to be doubled (microprocessor performance improvements)



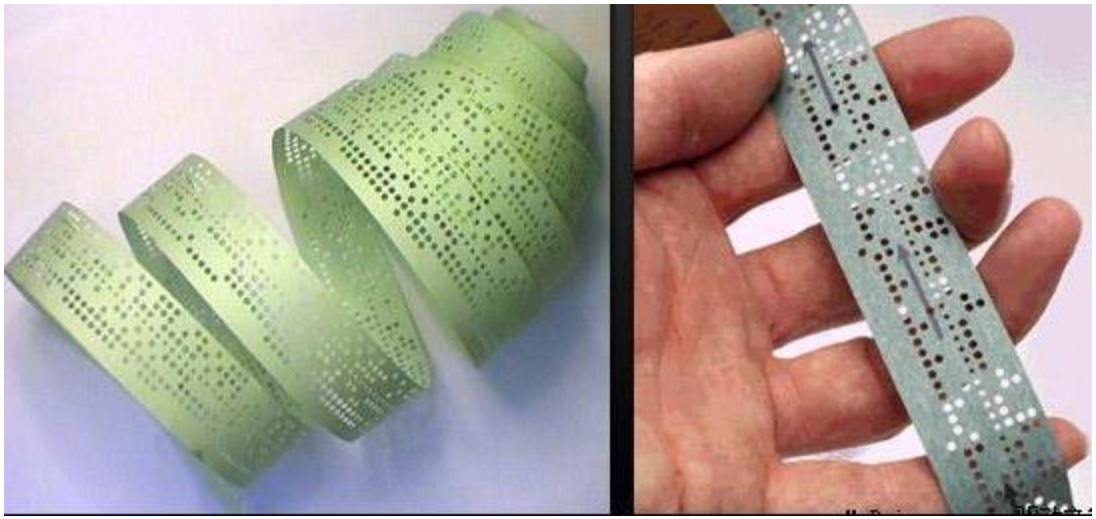


Performance limitation → eclipse in 2020?

Progress of information technology will not slow down

Development roadmap of microprocessors

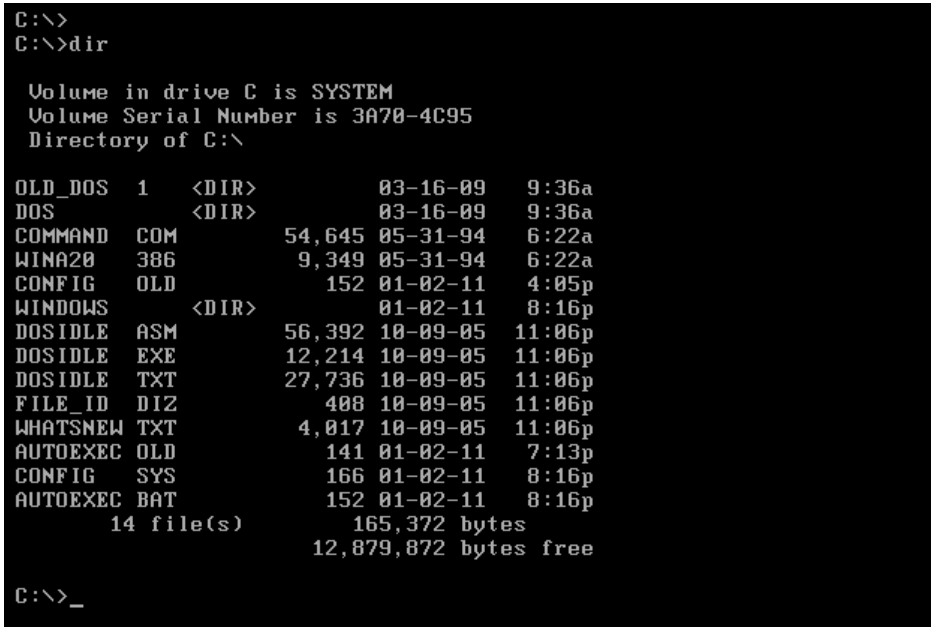
- First generation (from 1971)
  - 4 or 8-bit
  - **Word length**: the number of binary bits that can be directly processed, usually depending on the bits of the microprocessor internal registers and data bus width
  - Time frequency < 1MHz, average instruction execution time 10μs～15μs
  - Machine language programming, mainly used to control devices such as cash counters, traffic lights



10110000
00001001

Development roadmap of microprocessors

- Second generation (from 1974)
  - 8-bit
  - Time frequency 2 MHz, average instruction execution 1～2μs
  - Instruction set is more complete , with the assembly, BASIC and FORTRAN language
  - Single-user operating system



Development roadmap of microprocessors

- Third generation (from 1978)
  - 16-bit, Intel 8086
  - Time frequency 5,8,10 MHz, average instruction execution time 0.5μs
  - Rich instruction set
  - Intel 8086 was used as the IBM PC's CPU to be successful in the market, promoting the application of personal computers

Development roadmap of microprocessors

- Fourth generation (from 1983)
  - 32-bit
  - Time frequency 16MHz～33MHz, average instruction execution time <0.1μs
  - Intel 80386，80486
- Fifth generation (from 1993)
  - 32-bit, 64-bit
  - Time frequency 60MHz～200MHz ～3G～
  - Pentium series 586～PIV～ Dual Core ～Multi-core
  - Cluster: Multiple nodes, Multi-core

Development roadmap of X86 series

- X86：*86
  - Intel CPU, such as 8086, 80286
  - Instruction-compatible
  - Using X86 to identify a common instruction set

- Pentium (PII，PIV), Celeron, Core support the x86 instruction set
  - all belong to the X86 family

- AMD and other manufacturers have also produced CPUs with the X86 instruction set
  - Intel's CPU-compatible products
  - Huge X86 series and compatible CPU family

Intel microprocessor

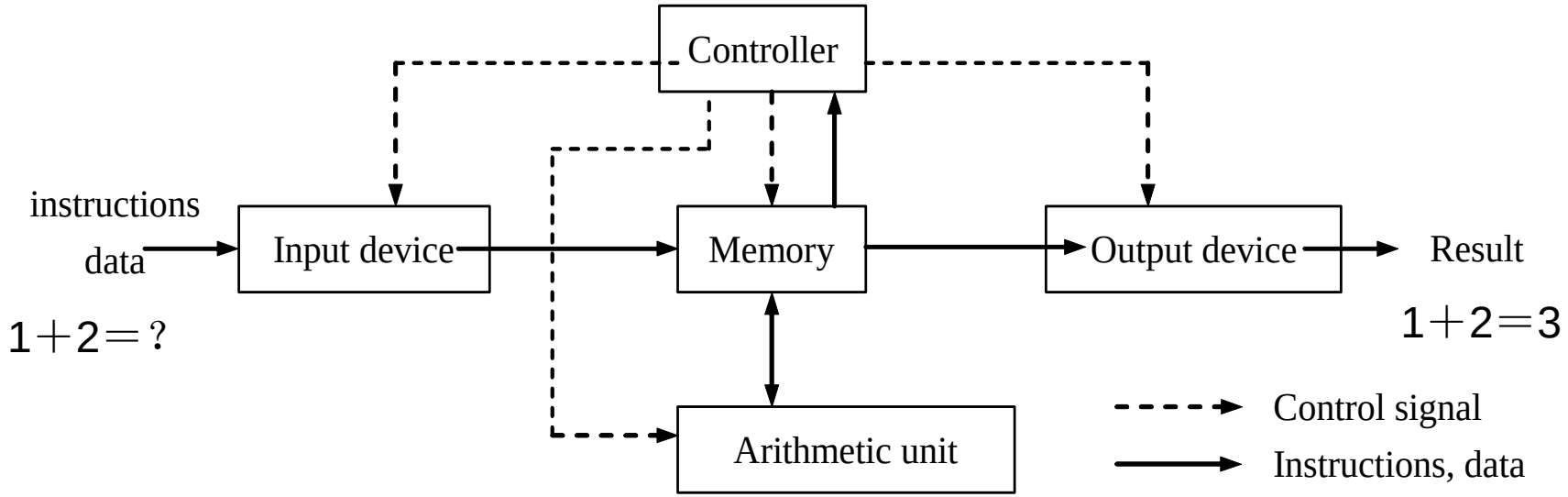
-------performance improvement

Chip	Address BUS	Data BUS	Memory Addressing	First cache	Second cache	Time frequency(Hz)	Integration（num/chip）
8080	16	8	64KB			2M	4500
8088	20	8	1MB			5M	29000
8086	20	16	1 MB			5M,8M,10M	29000
80286	24	16	16 MB			12M,20M,25M	134000
80386SX	24	16	16 MB			16M,25M,33M	275000
80386DX	32	32	4GB			16M,33M,40M	275000
80486DX	32	32	4 GB	8KB		25M ~100M	1.2 M
Pentium	32	64	4 GB	16 KB		66M ~200M	3.1 M
Pentium MMX	32(36)	64	64 GB	16 KB		200M ~300M	4.5 M
Pentium Pro	36	64	64 GB	16 KB	256 KB	150M ~200M	5.5 M
Pentium II	36	64	64 GB	32 KB	512 KB	233M ~450M	7.5 M
Pentium Xoen	36	64	64 GB	32 KB	512 KB	350M ~450M	7.5 M
Pentium III	36	64	64 GB	32 KB	512 KB	450M ~1.4GM	9.5 M
Pentium 4	36	64	64 GB	32 KB	256KB~1MB	1.3GM~2.8GM	12.5 B

Improve frequency / Integration of multiple computing cores

Basic structure of computer

- Microcomputer: LSI devices
- Von Neumann-type structure
  - Key contributions: Binary, memory
  - Data storage (instructions and data) and process control
  - 5 components → compared with a human



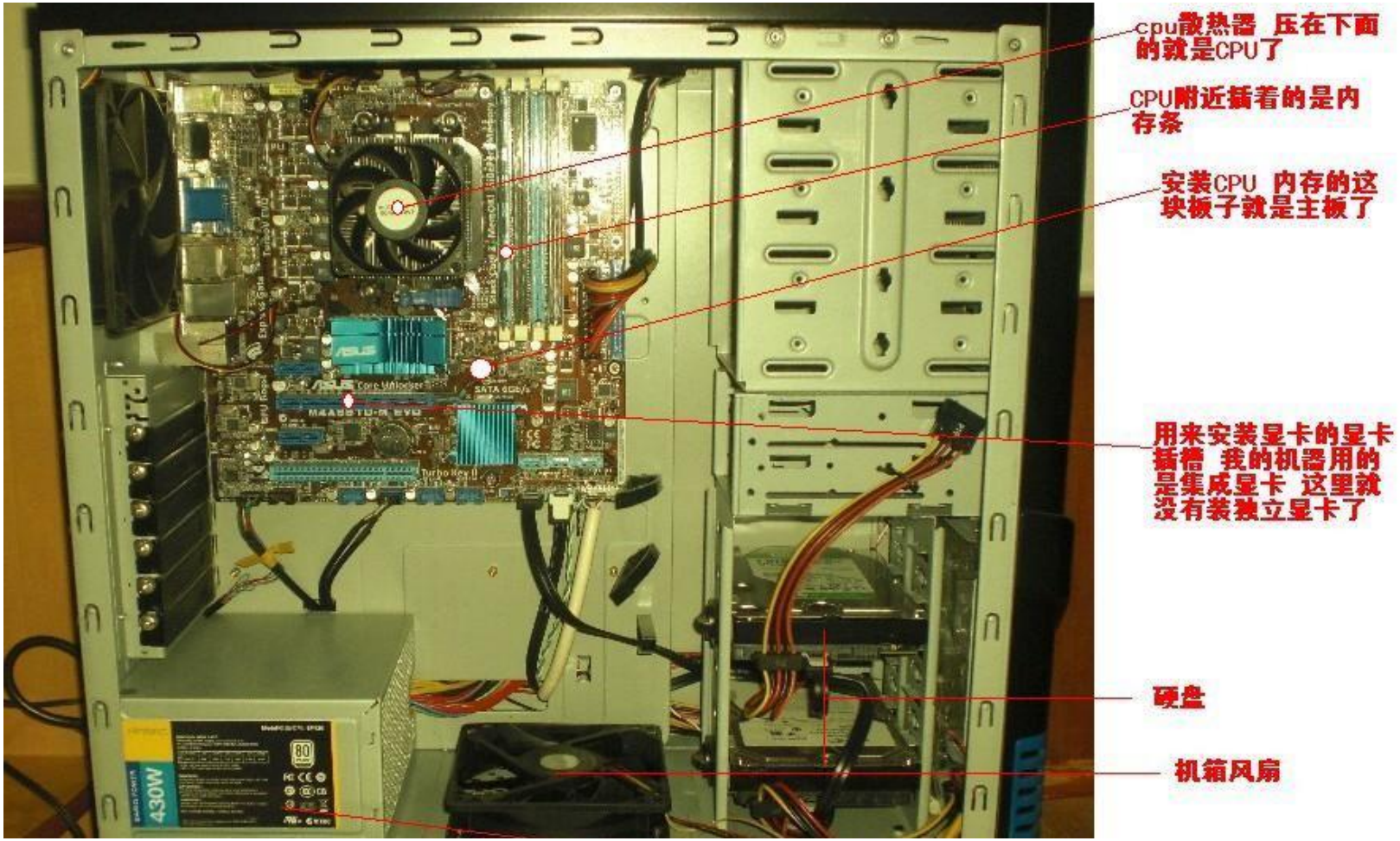
Microcomputer system

- Microcomputer system
  - **Structure**
    - Hardware system
    - Software system
      - System software
      - Application software

Microcomputer system

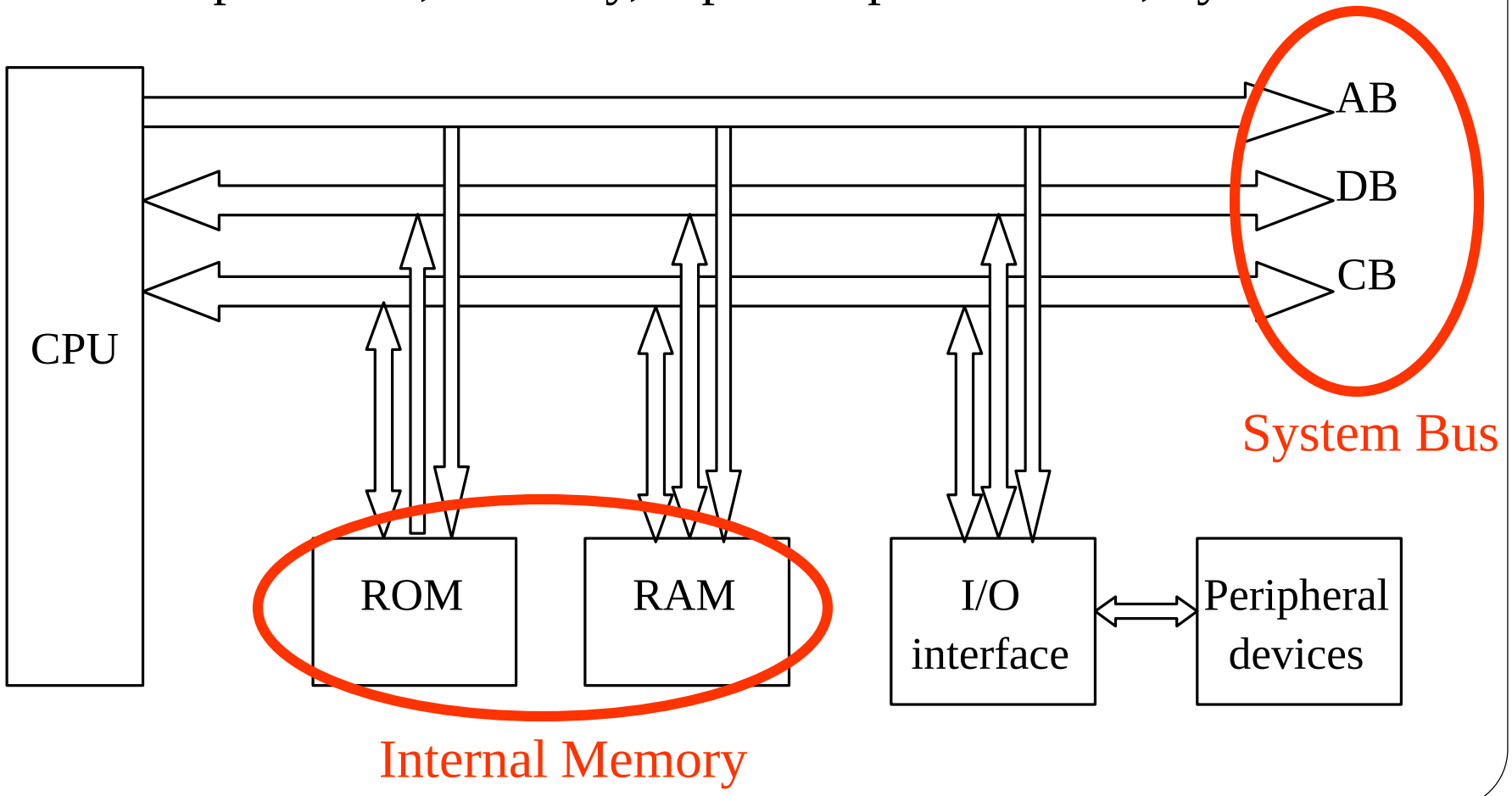
Von Neumann-type

- Hardware system

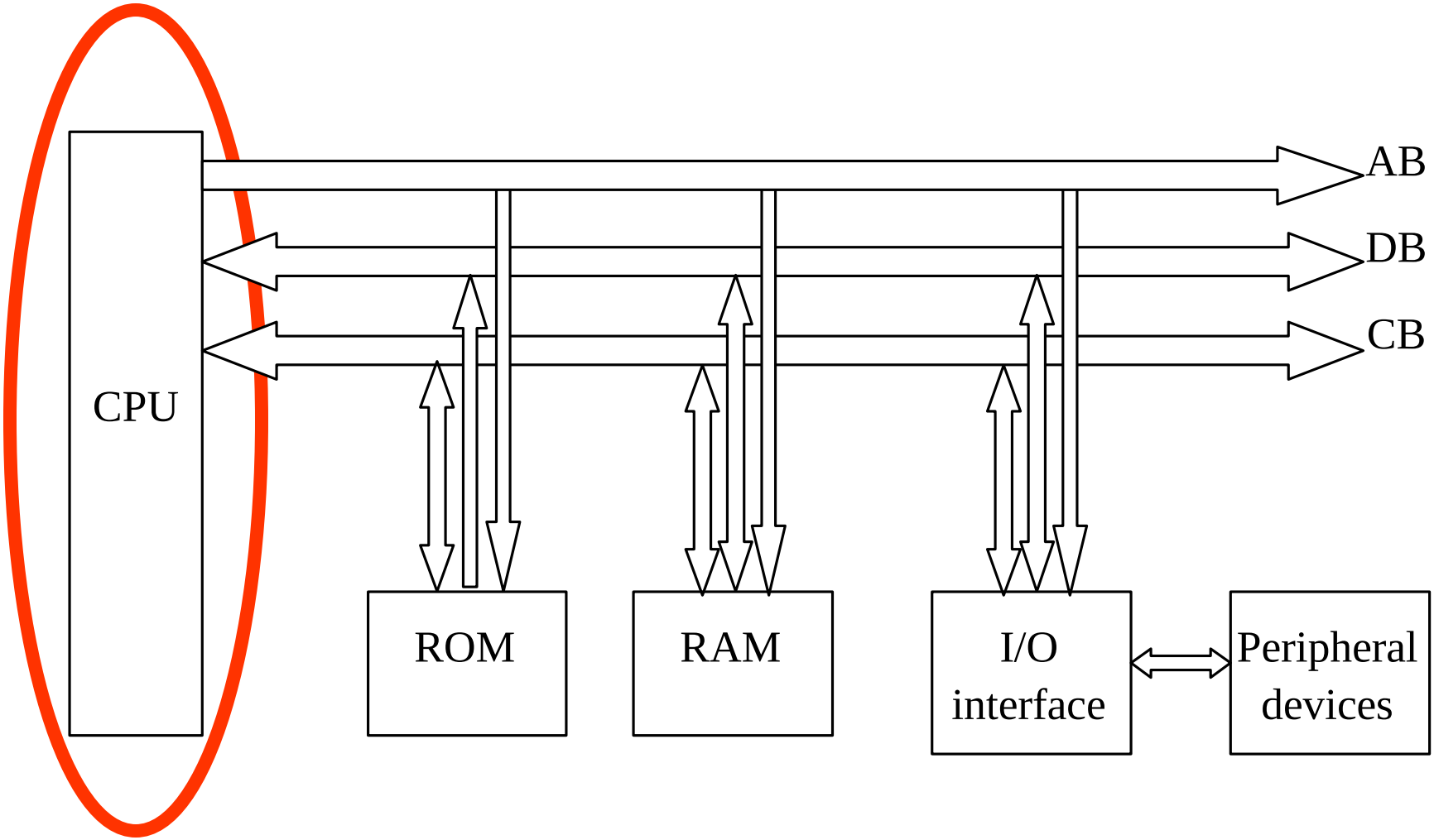


Microcomputer system

- Microcomputer
  - Microprocessor, Memory, Input/Output interface, System bus

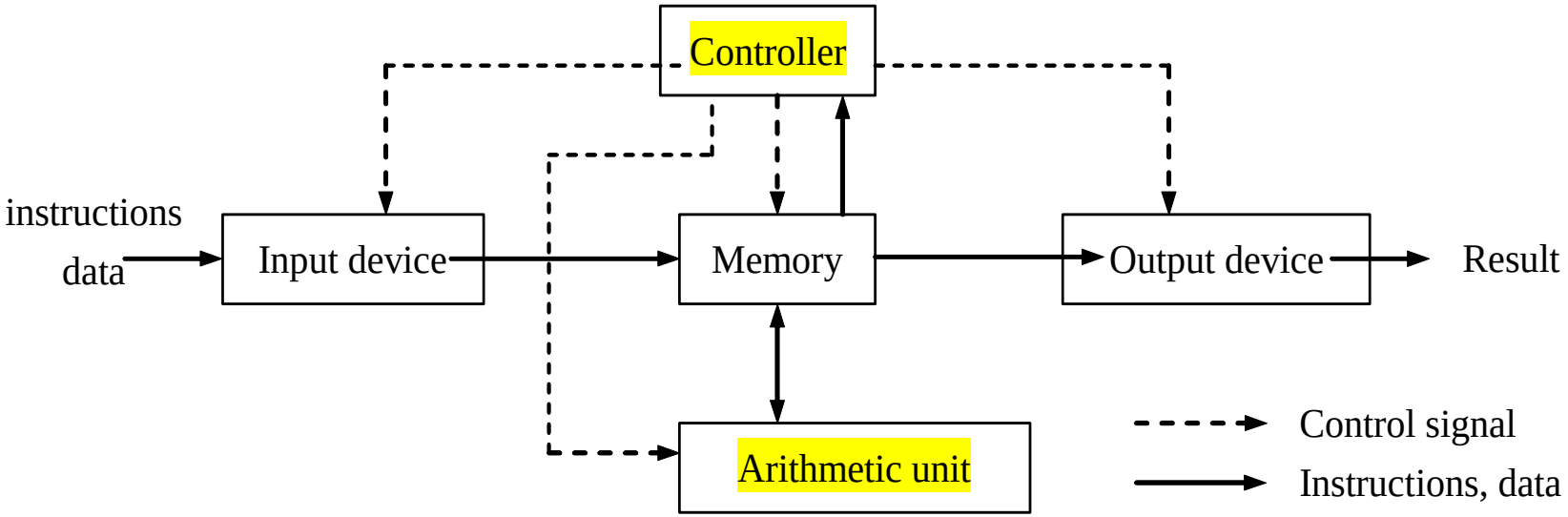


Microcomputer—CPU



Microcomputer—CPU

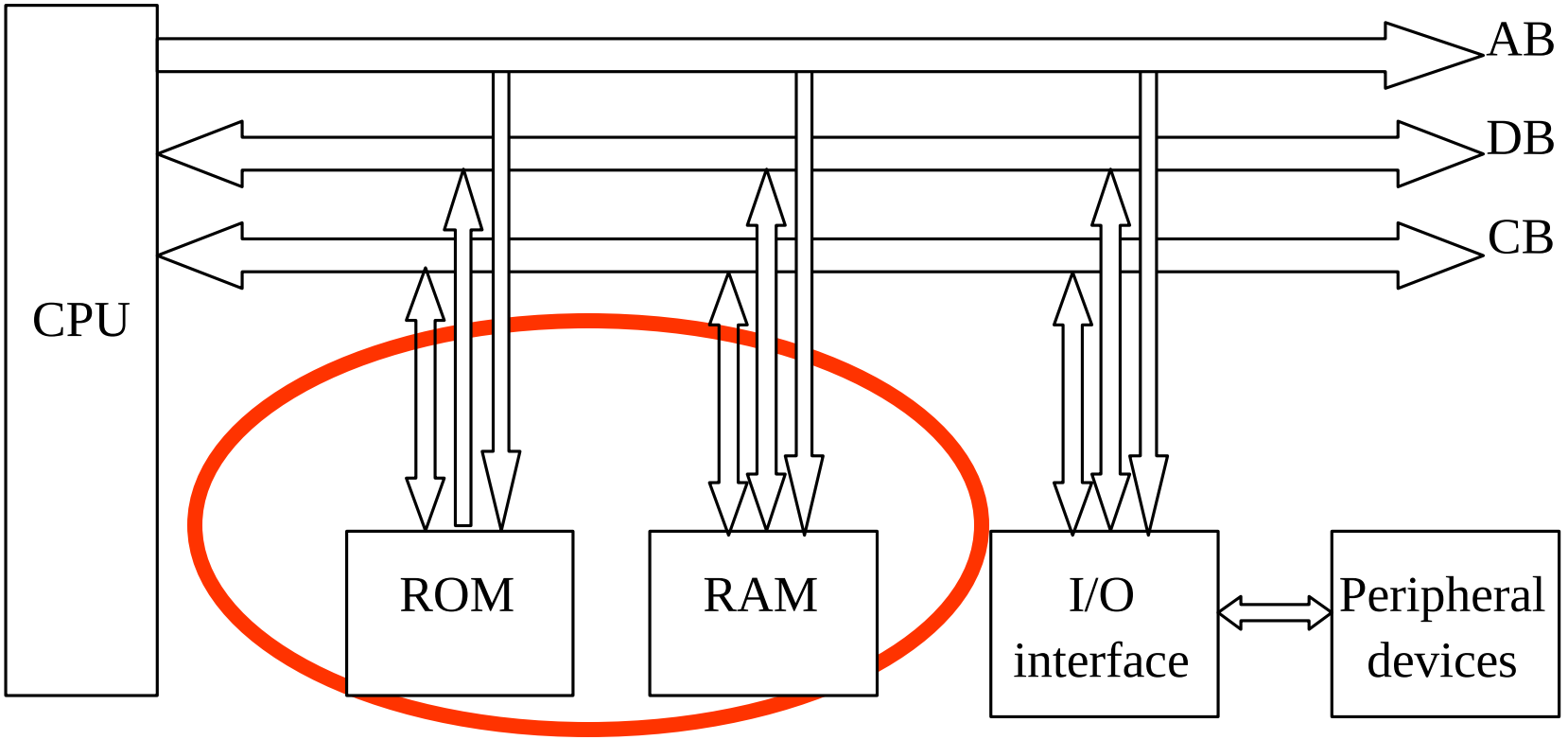
- Microprocessor
  - Core chip: CPU（Central Processing Unit）
  - Functions
    - Fetching instructions, decoding instructions, arithmetic operation, transferring data, controlling program flow



Microcomputer—CPU

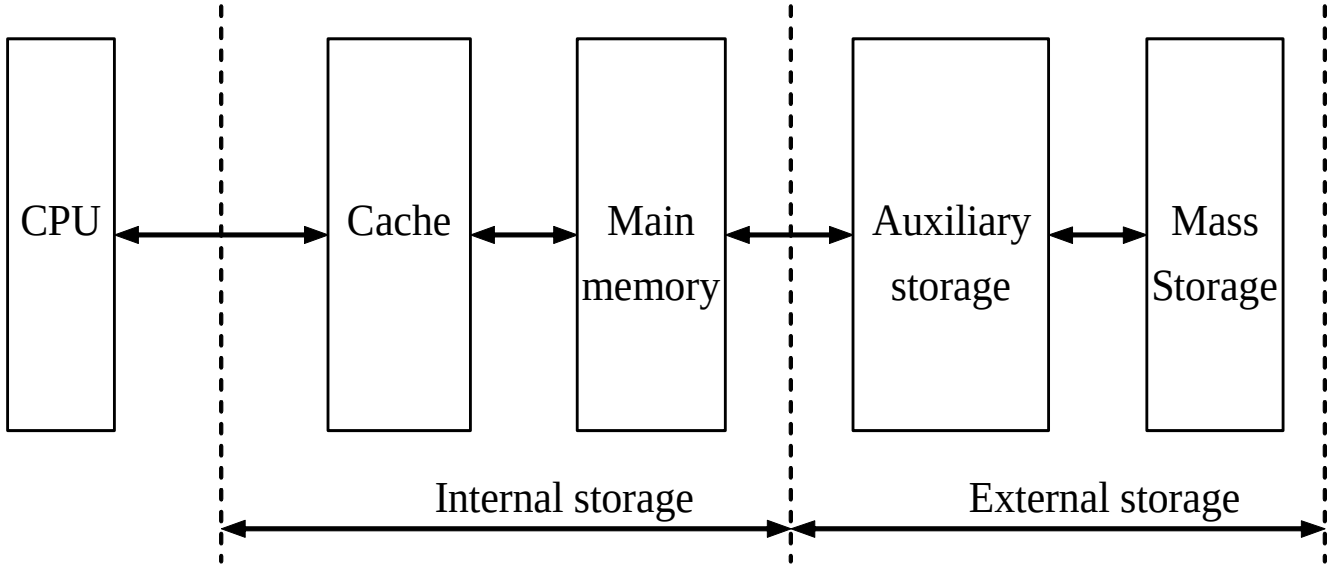
- Microprocessor
  - Composition (structure figure)
    - **Arithmetic unit** : ALU
    - **Controller**
      - Fetching instructions (Address generation), decoding, giving control signals (R/W/Enable)
    - **Register**: data storage
      - Data register: storage of data or processing result
      - Segment register: storage of segment address
      - Pointers and index register: storage of offset address, IP (offset address of next instruction)
      - Flag register: indicating the result of the operation status (overflow, sign, carry, etc.)

Microcomputer—Memory



Microcomputer—Memory

- Memory: storing data and programs (instructions)
- Internal memory and external memory
  - Internal memory for storing programs and data being used or frequently used
    - CPU can directly access - memory capacity, access speed
  - External memory for storing not frequently used data
    - Data should be transferred to internal memory first



Microcomputer—Memory

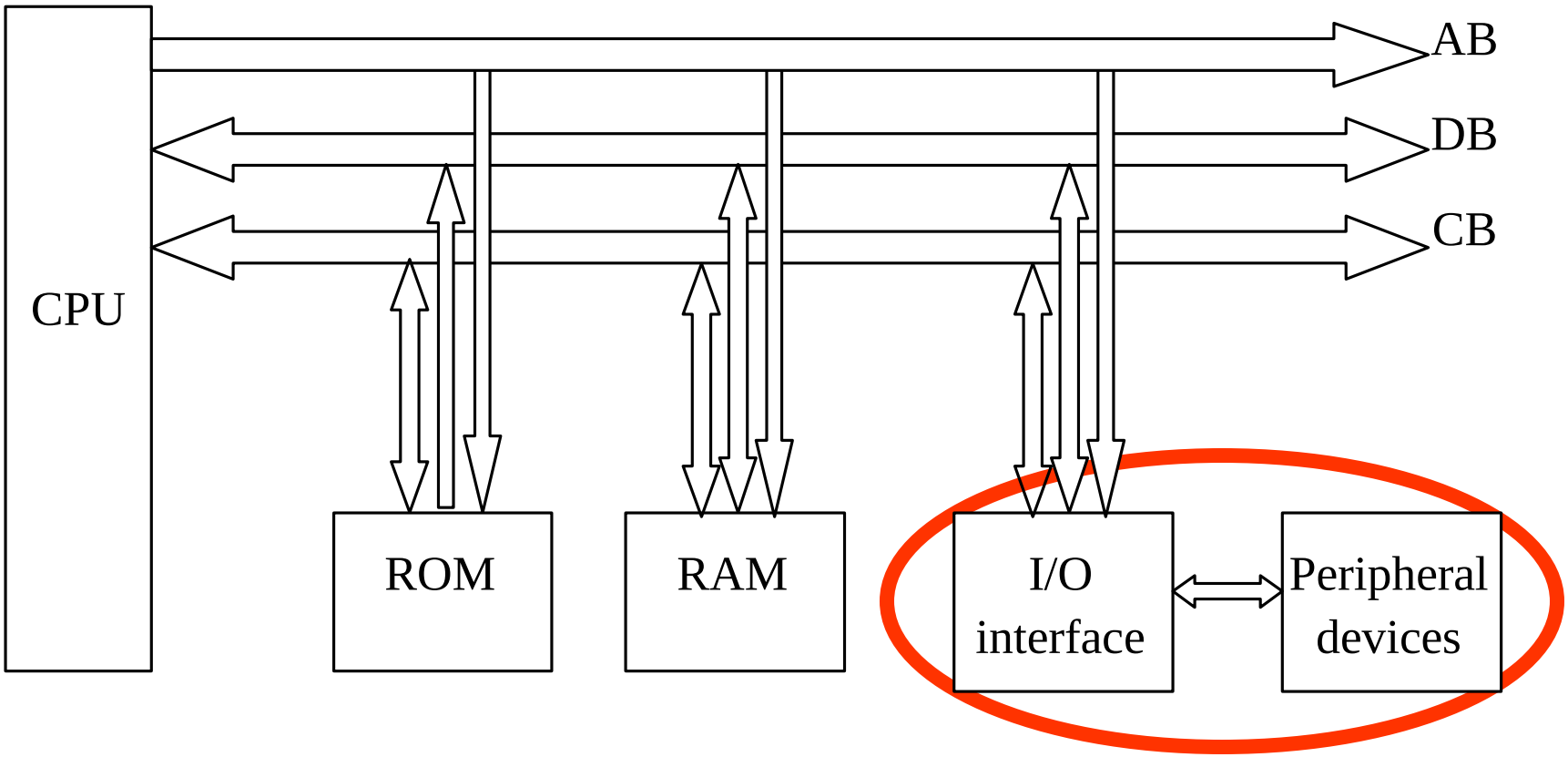
- Internal memory classification
  - ROM (Read Only Memory): Fixed procedures and data are written in advance by the manufacturer or user device – BIOS
  - RAM (Random Access Memory): Random access, erased after power failure

- Memory contains a lot of storage unit
  - Examples
  - Each unit has its own number called its address
  - Bit: the smallest basic unit of data that a computer can expressed, valued as a binary 0 or 1
  - Each memory unit has a fixed number of bits, called bytes; a byte contains 8 bits

Microcomputer—Memory

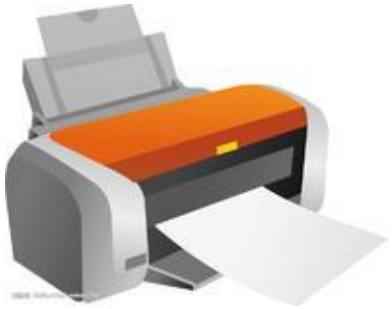
- External memory
  - Hard disk, floppy disk, tape, optical disk
- Removable storage devices
  - USB-based (FlashDisk), removable hard disk and so on

Microcomputer—I/O interface



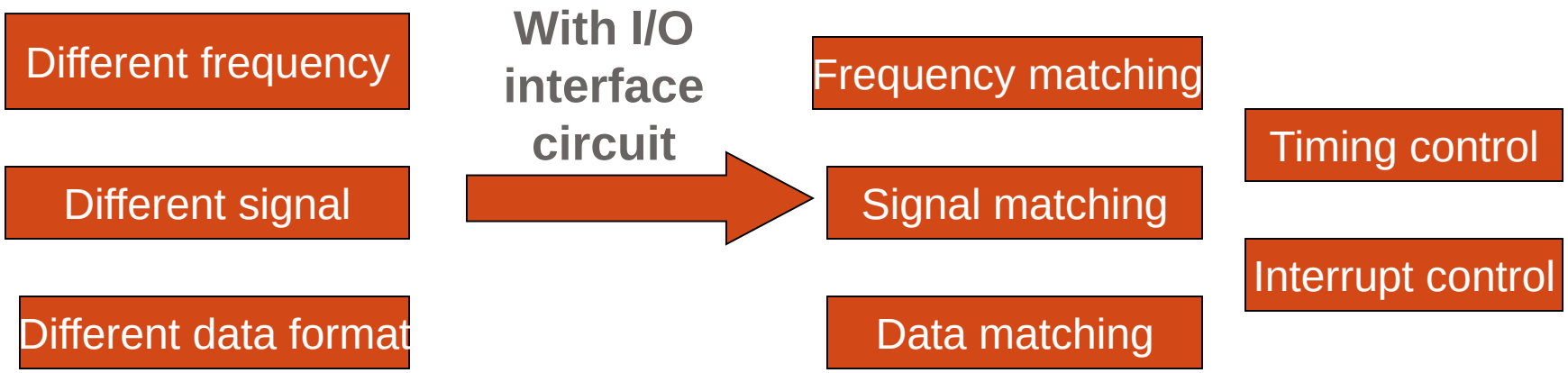
Microcomputer—I/O interface

- Peripheral devices
  - Input devices, display devices, printing devices, external memory, and network devices



Microcomputer—I/O interface

- Coordination of CPU and peripheral

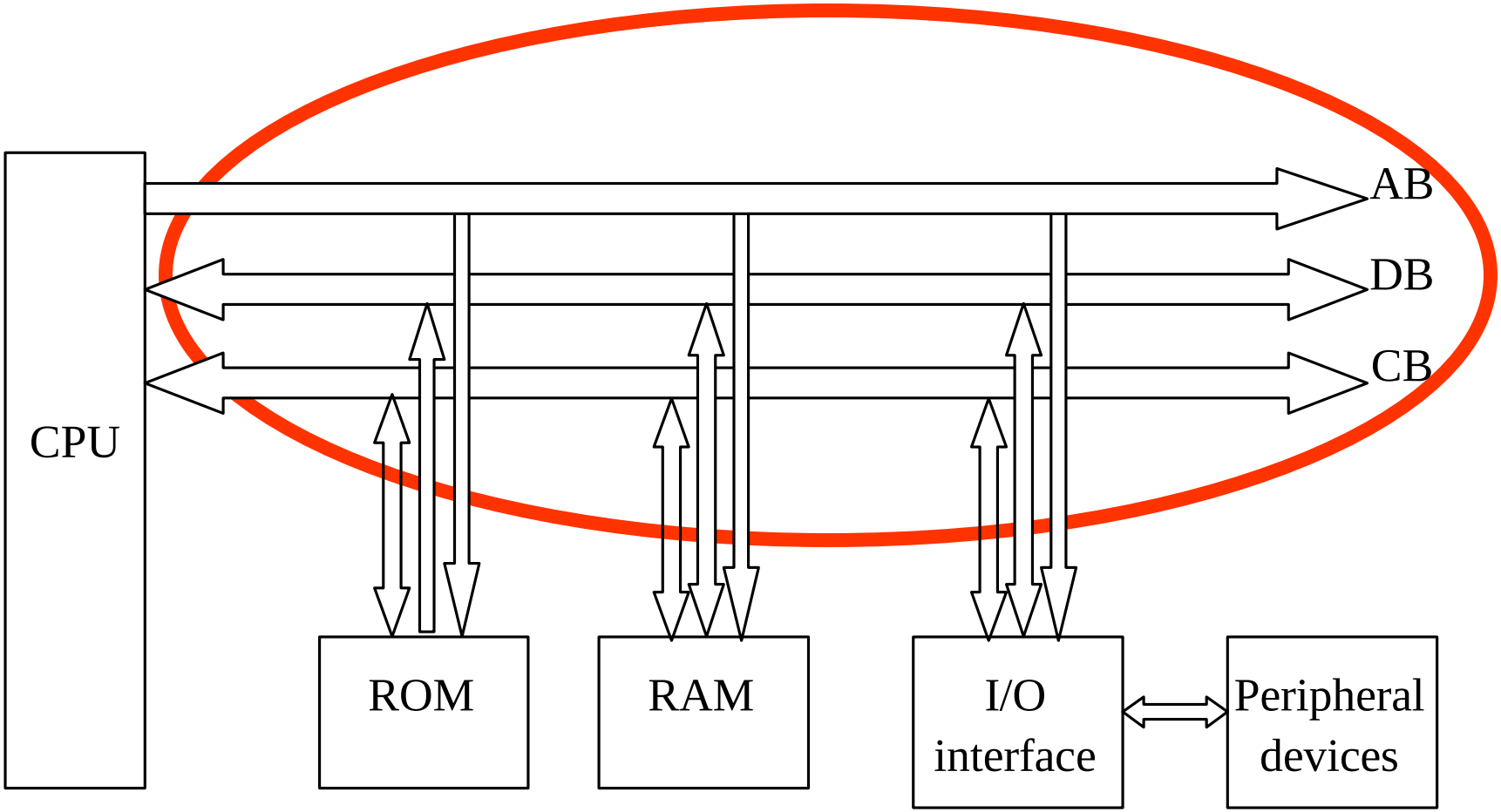


- Functions
  - For the exchange of information between the CPU (or memory) and peripheral devices
  - Address, control and data information

Microcomputer—I/O interface

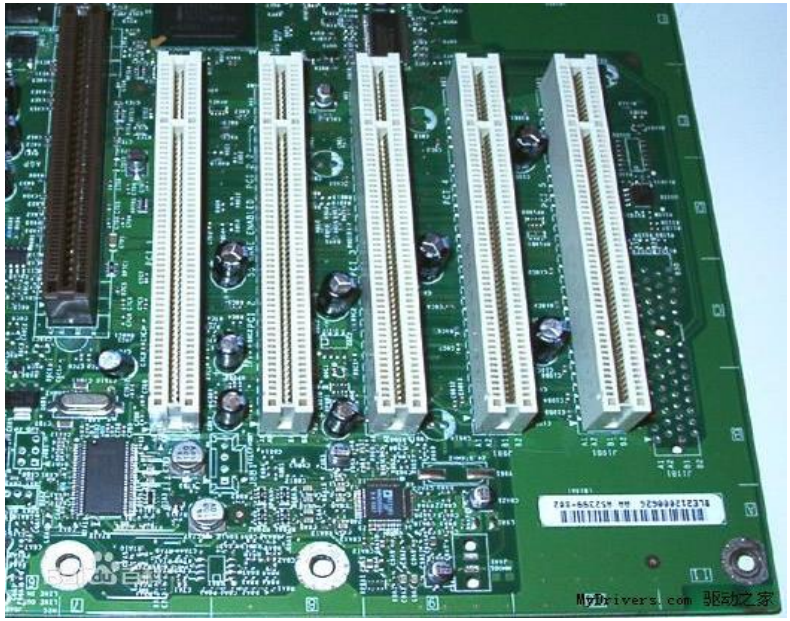
- Important interface chips
  - Programmable Interrupt Controller 8259A Chapter VII
  - Programmable counter / timer 8253 Chapter VIII
  - Programmable parallel interface 8255A Chapter IX
  - Programmable serial interface 8251A Chapter X
  - A / D (analog to digital), D / A (digital to analog) converter chip Chapter XI
- Learn how to use a chip

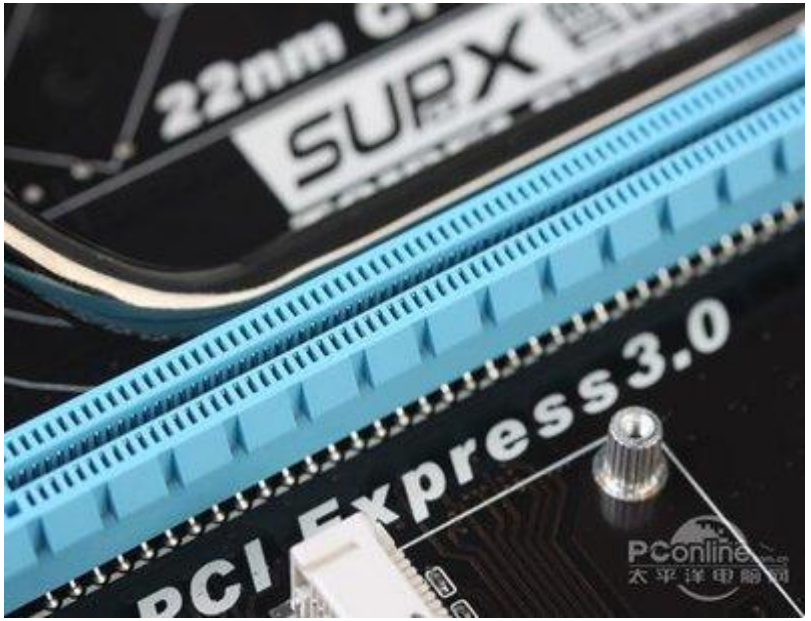
Microcomputer—System bus



Microcomputer—System bus

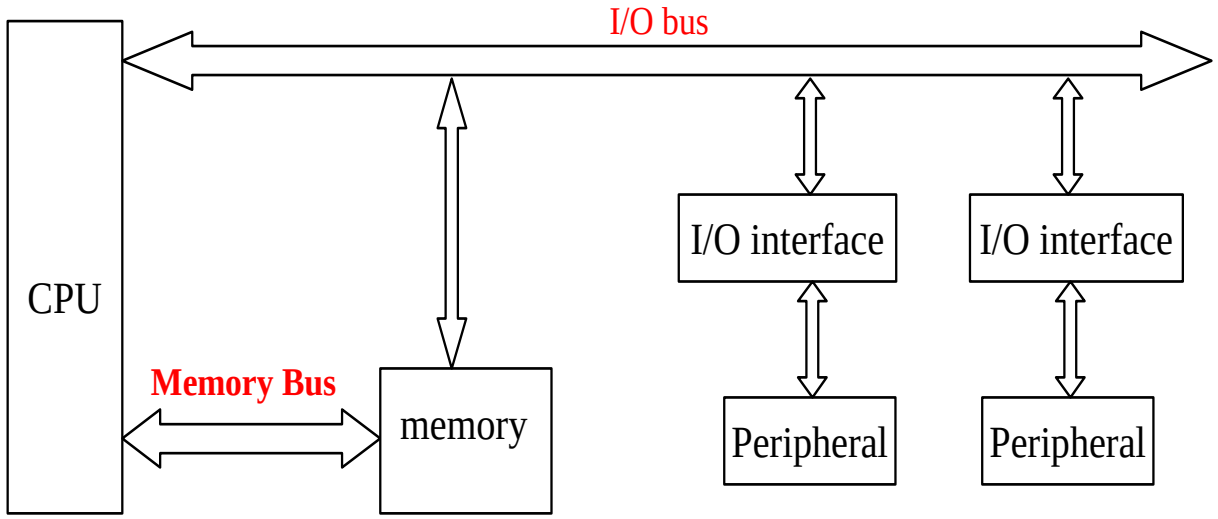
- The public access of information passed among various components in a computer system
- Physical form: a common set of wires and the corresponding driving circuit





Microcomputer—System bus

- **Memory-oriented dual bus architecture**



  - CPU/Memory CPU/IO Memory/IO
  - Improve the efficiency of information transmission, without reducing the efficiency of the CPU

Microcomputer—System bus

- **Bus Category**
  - By direction of the data transfer
  - By function

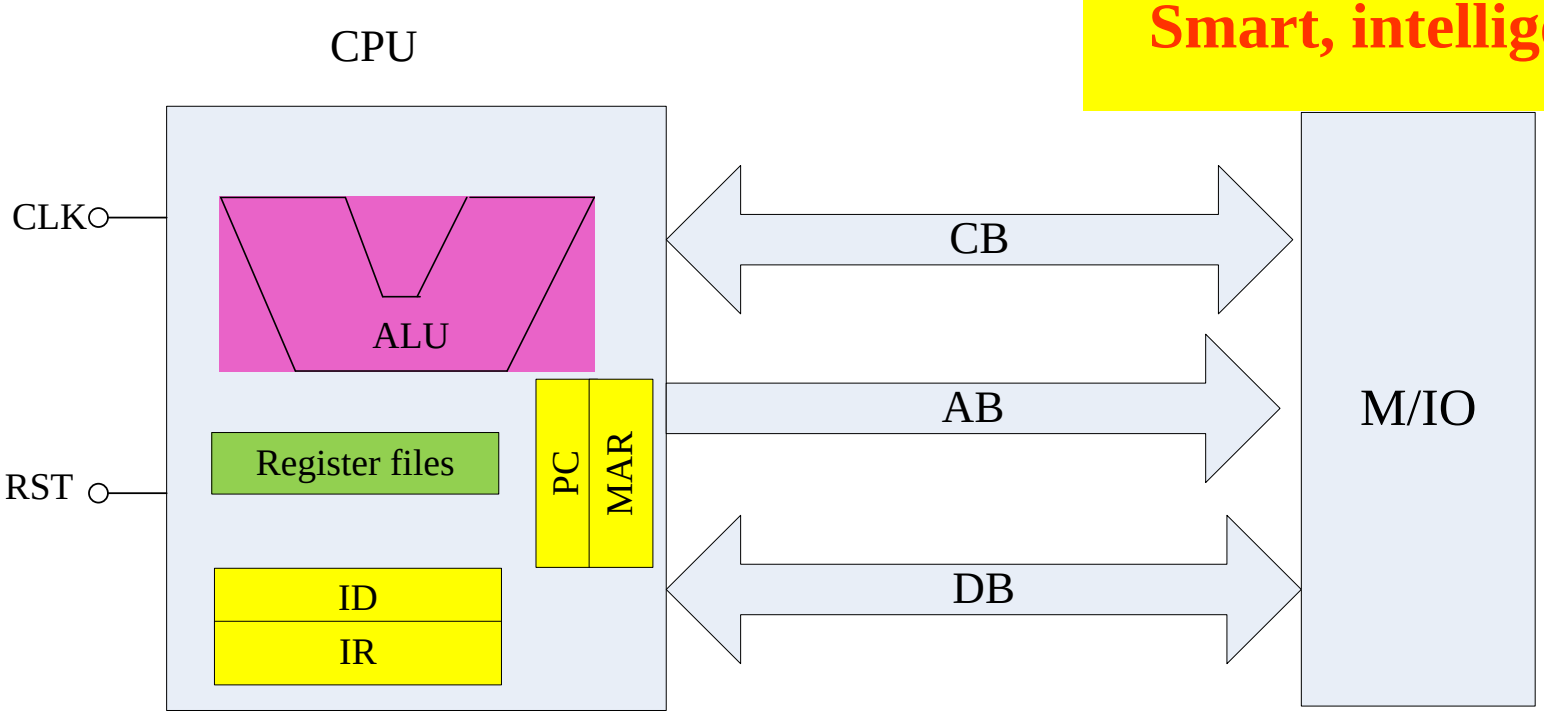
Working principle of microcomputer

- Basic concepts
  - Clock cycle: CPU works with clock cycles
    - Crystal oscillator generates a square wave signal, high + low
  - Machine code
    - Sequence composed of 0 and 1: instruction decoding
  - Instruction mnemonics
    - Add -> 11011000, assembler translation
  - Instruction system
    - User-oriented instruction set
- Microcomputer’s work process: Execute program
  - Instruction fetching, decoding and execution
  - Working process

Working principle of microcomputer

- Typical basic structure of microcomputer

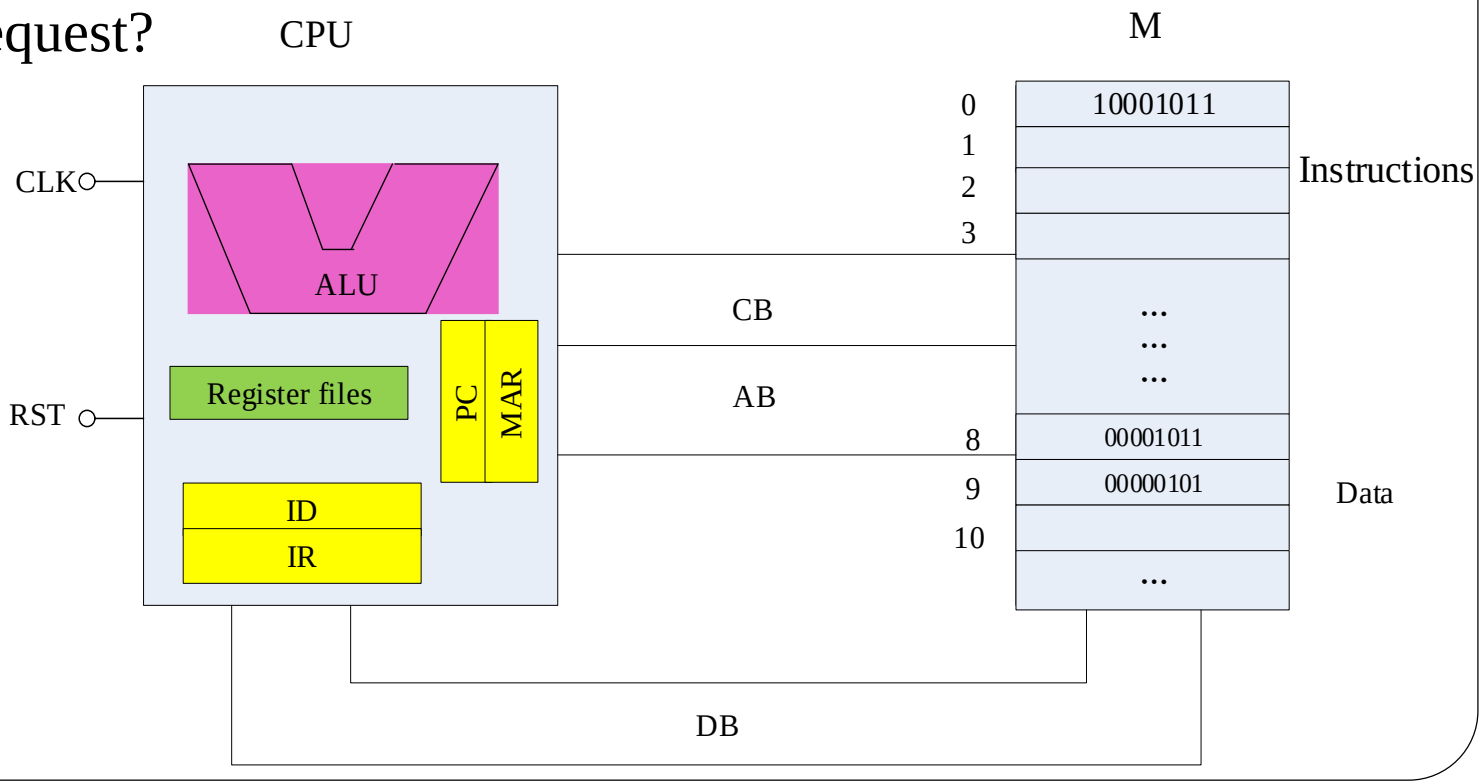
Smart, intelligent?



- Functions
  - **Arithmetic unit**, **register**, **controller**

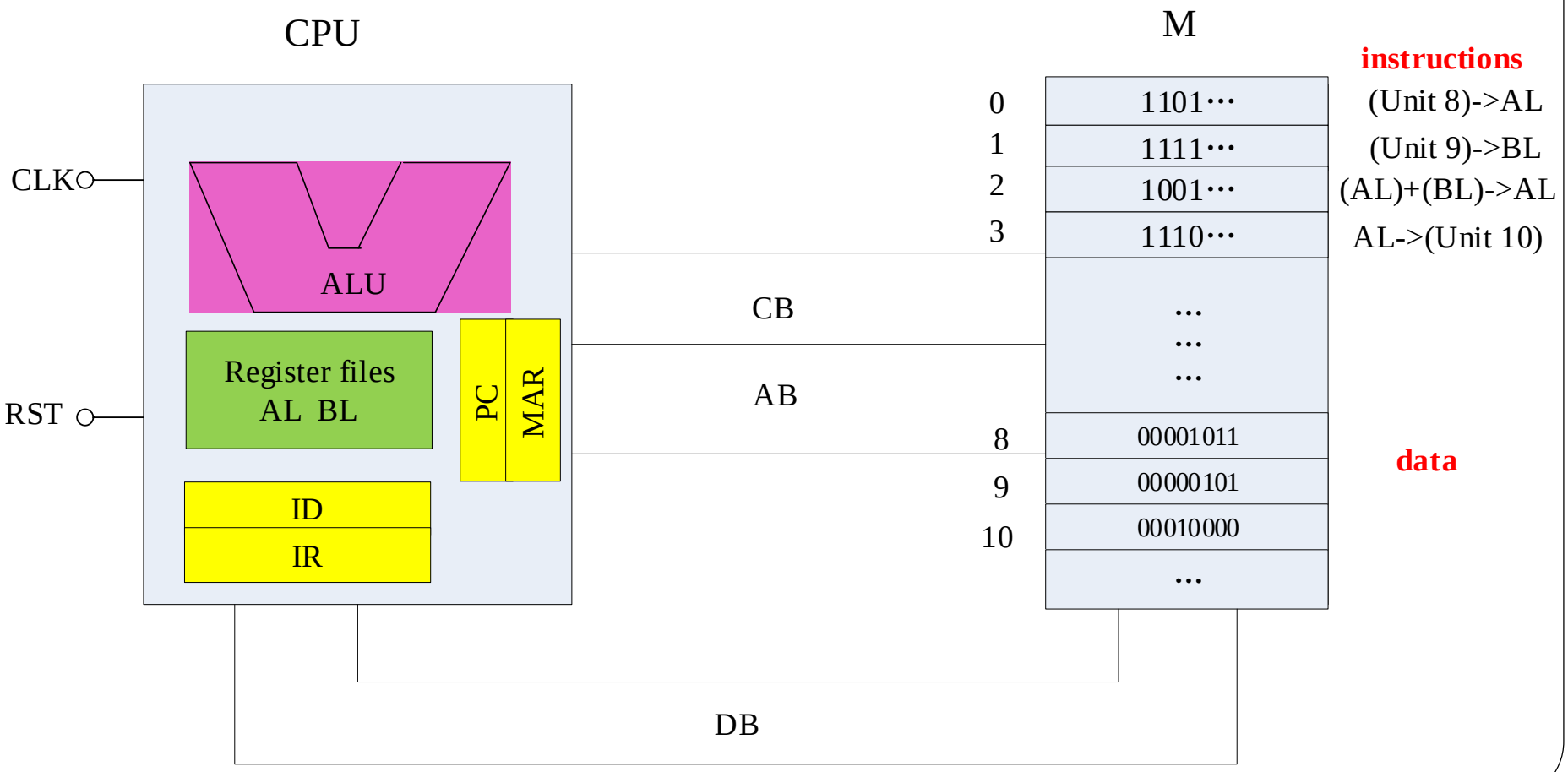
Working principle of microcomputer

- Working process
  - Fetching instructions：
    - PC, IP (instruction address)->AB； PC=PC+1, IP->next instruction;
  - Decoding, execution
  - Interrupt request?



Working principle of microcomputer

- Examples: 11+5=?
  - Unit 8->AL; Unit 9->BL; AL=AL+BL; Adding result->Unit 10



Working principle of microcomputer

- Conclusions
  - CPU execution
    - PC=PC+1(Von Neumann) ->IP (8086)
    - IP: offset address of the next instruction in the current code segment
  - Instruction cycle
    - Fetching, decoding and executing instructions
    - Multiple clock cycles
  - Functions of CB、DB and AB
  - Read/write
    - To CPU
  - Procedures (instructions) and data are data
    - Stored in different segment

Smart, intelligent?

Numeral systems

- Decimal
  - With suffix D or omitted eg: 231.1123
- Binary
  - With suffix B eg: 1101.101B
- Octal
  - With suffix Q or O eg: 625.77Q
- Hexadecimal
  - With suffix H eg: 37CF.56H
- Binary coded hexadecimal (BCH code)
- Binary coded decimal (BCD code)

Numeral systems

- Decimal
  - Base numbers: 0~9
  - $(X_nX_{n-1}\ldots X_1X_0X_{-1}\ldots X_{-m})_{10} = \sum_{-m}^{n} X_i \times 10^i$ 10---base 10^i --weight
  - **Examples**

- Binary
  - Base numbers: 0 1
  - $(X_nX_{n-1}\ldots X_1X_0X_{-1}\ldots X_{-m})_2 = \sum_{-m}^{n} X_i \times 2^i$ 2---base 2^i --weight
  - **Examples**

Numeral systems

- Octal
  - Base numbers: 0 ~7
  - $(X_n X_{n-1} \ldots X_1 X_0 X_{-1} \ldots X_{-m})_8 = \sum_{-m}^{n} X_i \times 8^i$
  - **Examples**
- Hexadecimal
  - Base numbers: 0~9 A B C D E F
  - $(X_n X_{n-1} \ldots X_1 X_0 X_{-1} \ldots X_{-m})_{16} = \sum_{-m}^{n} X_i \times 16^i$
  - **Examples**

Numeral systems

- BCH
  - Four binary digits represent a hexadecimal numeral
  - **BCH code table**
  - **Examples**

- BCD: convenient to store and display a decimal digit
  - A decimal digit represented by binary coding
    - **0～9**
    - **0000～1001**
  - Packed (two numerals in a single byte) and Unpacked (a byte represents a numeral)
  - **Examples**
  - **Different from Binary**
    - 01101001BCD
    - 01101001B

Numeral systems conversion

- Others －>Decimal
  - Expressed with weight
  - **Examples**

- Decimal －> others
  - Integer fraction (Divide the base and get the remainder) **examples**
    - Until the quotient is zero / get the remainder from back to front
  - Decimal fraction(Multiply the base and get the integer) **examples**
    - Until the decimal fraction is zero / get the integer from front to back
  - 0.7 -> Binary?
    - Bit length limitation

Numeral systems conversion

- Octal<—>Binary
  - Three binary bits correspond to an octal digit
  - Octal —> Binary **examples**
    - Each octal digit represented by three binary digits
  - Octal <— Binary **examples**
    - Divide from the decimal point to the left and to the right respectively/ three digits a group/ add 0 when digits are less than three

- Hexadecimal <—> Binary
  - Four binary bits correspond to an hexadecimal digit
  - Hexadecimal —> Binary **examples**
    - Each hexadecimal digit represented by four binary digits
  - Hexadecimal <— Binary **examples**
    - Divide from the decimal point to the left and to the right respectively/ four digits a group/ add 0 when digits are less than four

Numeral systems conversion

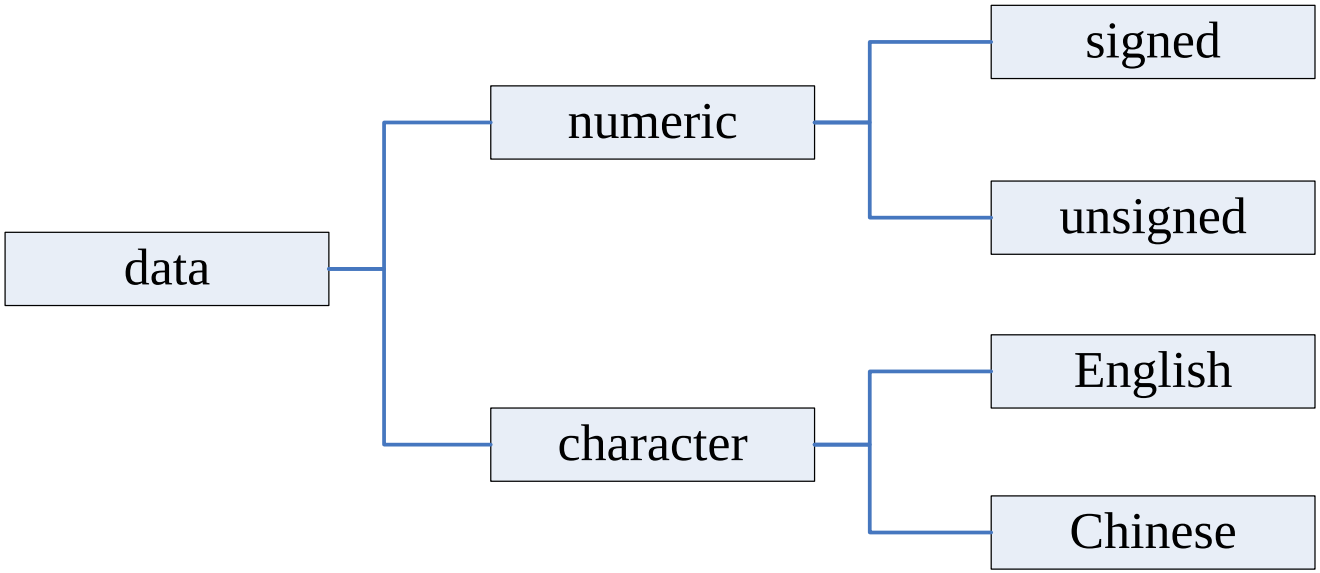
- Decimal <—>BCD
  - Decimal —>BCD **examples (packed)**
  - Decimal <—BCD **examples (packed)**
- Binary<—>BCD
  - Binary —> Decimal —>BCD
  - Binary <— Decimal <—BCD

Binary computation rules

- Arithmetic operations
  - Addition, subtraction, multiplication, division
  - Examples

- Logical operations
  - Or, And, Not, Xor
  - Examples

Data storage in a computer



Storage of numeric data

- Sign-and-magnitude coding
  - Allocate one sign bit to represent the sign
    - positive number: 0 negative number: 1
  - The remaining bits in the number indicate the magnitude (or absolute value)
  - **Examples**
  - Representation range
- Ones' complement coding
  - Positive number: the same with sign-and-magnitude
  - Negative number: unchanged sign bit and bitwise not magnitude bits
  - **Examples**
  - Representation range

Storage of numeric data

- Two's complement coding
  - Positive number: the same with sign-and-magnitude
  - Negative number: Ones' complement + 1
  - **Examples**
  - Representation range
  - Two's complement -> numeric value
    - Positive number
    - Negative number **Examples**
- Expansion
  - Positive number: Preceded by 0
    - 1 00000001 0000000000000001
  - Negative number: Preceded by 1
    - -1 11111111 1111111111111111

Storage of numeric data

- Application of Two's complement coding
  - $[X + Y]_{\text{Two's complement}} = [X]_{\text{Two's complement}} + [Y]_{\text{Two's complement}}$

	Decimal Addition	Binary Addition	
①			
	+66	0100 0010	= $[+66]_{补}$
	+) +51	+) 0011 0011	= $[+51]_{补}$
	+117	0111 0101	= $[+117]_{补}$
②			
	+66	0100 0010	= $[+66]_{补}$
	+) - 51	+) 1100 1101	= $[- 51]_{补}$
	+15	1 0000 1111	= $[+15]_{补}$
③			
	- 66	1011 1110	= $[- 66]_{补}$
	+) +51	+) 0011 0011	= $[+51]_{补}$
	- 15	1111 0001	= $[- 15]_{补}$
④			
	- 66	1011 1110	= $[- 66]_{补}$
	+) - 51	+) 1100 1101	= $[- 51]_{补}$
	- 117	1 1000 1011	= $[- 117]_{补}$

Storage of numeric data

- Application of Two's complement coding
  - $[X - Y]_{\text{Two's complement}} = [X]_{\text{Two's complement}} + [-Y]_{\text{Two's complement}}$

	Decimal Addition	Binary Addition	
①			
	+66	0100 0010	$= [+66]_{补}$
	—) +51	+) 1100 1101	$= [-51]_{补}$
	+15	1 0000 1111	$= [+15]_{补}$
②			
	+66	0100 0010	$= [+66]_{补}$
	—) - 51	+) 0011 0011	$= [+51]_{补}$
	+117	0111 0101	$= [+117]_{补}$
③			
	+51	0011 0011	$= [+51]_{补}$
	—) +66	+) 1011 1110	$= [-66]_{补}$
	- 15	1111 0001	$= [-15]_{补}$
④			
	- 51	1100 1101	$= [-51]_{补}$
	—) - 66	+) 0100 0010	$= [+66]_{补}$
	+15	1 0000 1111	$= [+15]_{补}$

Storage of numeric data

- Benefits of Two's complement coding- simplify the hardware
  - Turn subtraction into addition (eliminating the subtractor)
  - Applicable to addition of unsigned and signed numbers
    - **Example: 11110001B+ 00001100B**

Storage of numeric data

- Overflow
  - Beyond the representation range
    - **Examples**
  - Judgment of overflow
    - [-128,+127] [-32768,+32767]
    - Directly check the decimal arithmetic results

Storage of character data

- English characters
  - ASCII (American Standard Code for Information Interchange)
  - 8 binary bit coding, with the MSB is 0
  - The remaining seven bits can be used to define 128 characters
  - 33 non-printing control characters that affect how text and space is processed
  - 95 printable characters, including the space

Storage of character data

ASCII Characters

H / L	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	**0**	@	P	`	p
0001	SOH	DC1	!	1	**A**	Q	**a**	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENG	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	**LF**	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	|
1101	**CR**	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	↑	n	~
1111	SI	US	/	?	O	←	o	DEL

Storage of character data

- “0”~”9”
  - From 0110000(30H)
- “A”~”Z”
  - From 1000001(41H)
- “a”~”z”
  - From 1100001(61H)
- 0AH Line feed，0DH Carriage return
- Examples：
  - Printer: 41H 41H
    0AH, 0DH, 61H 0DH, 61H

Storage of character data

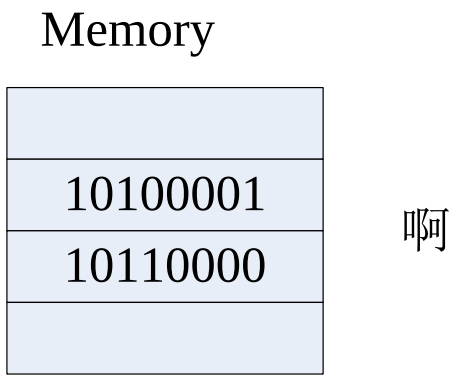
- Storage of simplified Chinese characters
- GB2312-80
  - GB2312 covers 99.75% of the characters used for Chinese input, each character is expressed with 2 bytes
  - Characters in GB2312 are arranged in a 94x94 grid (the quwei form), which specifies a row (qu) and the position of the character within the row (wei).

Quwei	01 02 03 …… 94
01 …	symbols
16 … 55	First plane for Chinese characters
56 … 94	Second plane for Chinese characters

Storage of character data

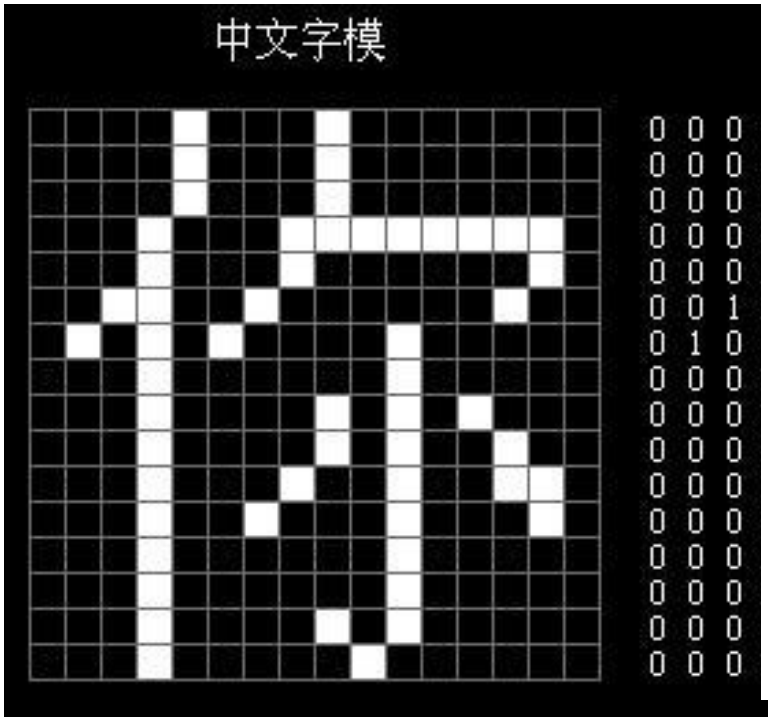
- Quwei code: encoding according to quwei position
  - Eg: “。” 01 Qu 03 Wei, 0103H
  - “啊” 16 qu 01 wei, 1001H

- EUC-CN: encoding of GB2312 for storage
  - Maintaining compatibility with ASCII
  - EUC-CN code = Quwei code + A0A0H
  - Started with 1, different with ASCII
  - Eg: “啊” 1001H + A0A0H = B0A1H



Display of character data

- Character font library
  - bitmap font(16×1
  - Vector font library



Homework

1. 将二进制数转换为十进制数

（1）1101.01B

（2）111001.0011B

(3) 101011.0101B

(4) 111.0001B

2. 将十六进制转换为十进制数

(1)A3.3H

(2)129.CH

(3)AC.DCH

(4)FAB.3H

3. 将十进制分别转换为 8 位原码、补码和反码

（1）+32 （2）-12 （3）+100 （4）-92

4. 将十进制转换为压缩和非压缩格式的 BCD 码

(1)102 (2)44 (3)301 (4)1000

5. 将下列补码转换为有符号十进制数

(1)10000000B (2)00110011B (3)10010010B (4)10001001B