

计算机组成原理

主讲：丁贤庆、阙夏

课程 qq群号：661046675

提 纲

1

概 述

2

计算机系统的硬件结构

3

中央处理器 (CPU)

4

控 制 单 元 (CU)

课程学习目的

- ➡ **深入理解**现代计算机的体系结构、基本问题, 工程中的权衡;
- ➡ 如何**设计**计算机系统;
- ➡ **理解**计算机系统的工作方式是什么? 各种操作为什么这样执行?

大纲要求

计算机组成原理考查目标

- ➡ **理解**单处理器计算机系统中各部件的内部工作原理、组成结构以及相互连接方式，具有完整的计算机系统的整机概念。
- ➡ **理解**计算机系统层次化结构概念，熟悉硬件与软件之间的界面，掌握指令集体系结构的基本知识和基本实现方法。
- ➡ **能够运用**计算机组成的基本原理和基本方法，对有关计算机硬件系统中的理论和实际问题进行计算、分析，并能对一些基本部件进行简单设计。

课程考核成绩的组成

- 作业 15%
- 专题讨论，课堂表现 10%
- 测验 / 期中考试 20%
- 期末考试成绩占40%
- 实验 15%

成绩类型	分项成绩规则	比例	记分形式	合计百分比
期末考试	1%-100%	40	百分制	合计百分比 100 %
课堂测试	1%-100%		...	
期中考试	1%-100%	20	百分制	
作业撰写	1%-100%	15	百分制	
实验分析报告	1%-100%	15	...	
期末报告	1%-100%		...	
课堂演讲	1%-100%	10	百分制	
论文撰写	1%-100%		...	

- 每一章的学习目标将在每章学习前说明，要求同学们在学习过程中能够针对每个知识点认真总结。

学习要求

作业要求：

1、所有作业来源于配套习题册,习题册上的选择题和填空题必做，其他题目每个同学自己选做,多做有加分。

(75基础分+5分排版分+20选做分)

2、按章节顺序或者学习顺序排列，图片摆正。

3、共收两次作业，只收pdf电子档（手写拍照，ipad手写笔均可，一次作业一个文档，控制文件大小，能看清楚就行，不需要高清）

4、文档前面要写明自己完成情况。

样例如下：

完成情况：

第七章~第十章 选择、填空题及校对

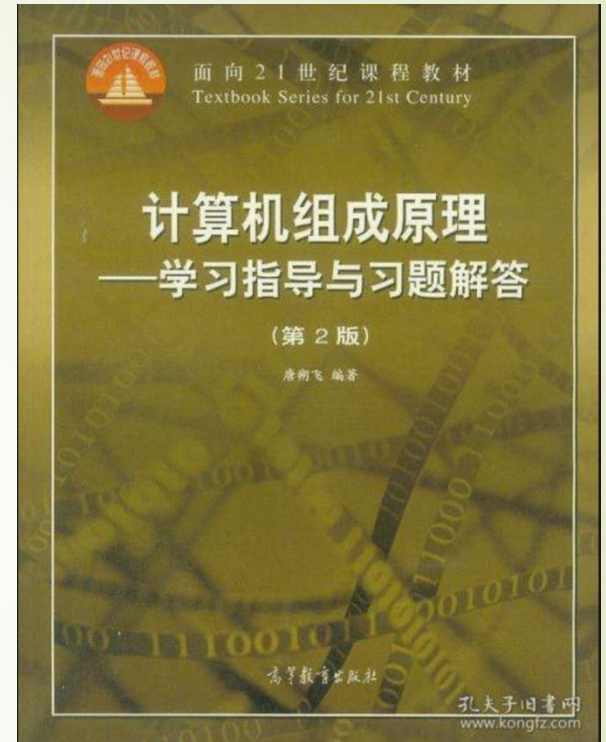
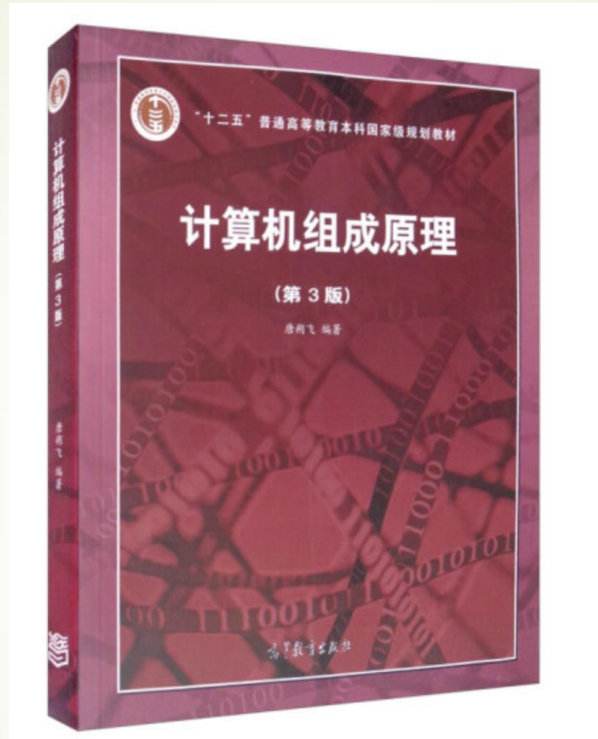
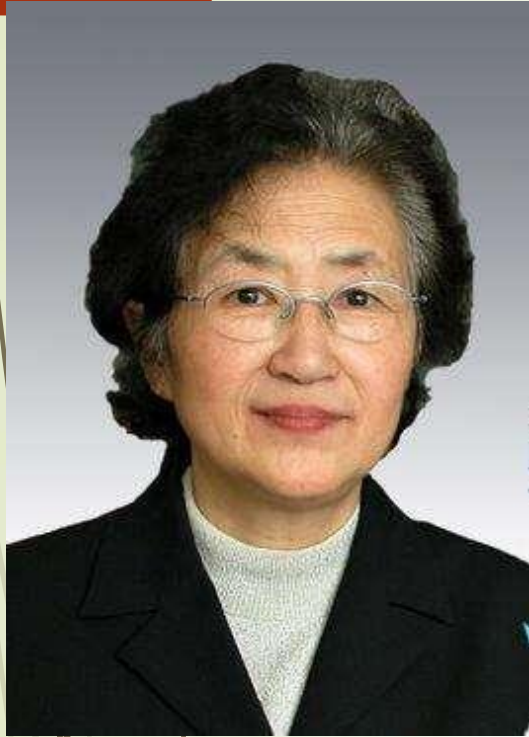
问答题：第七章 20/36

第八章 全部（除指令流水相关知识点）

第九章 8/14

第十章 4/14

教材和辅助习题册(必须有)



➤ 《计算机组成原理》（第3版）唐朔飞 高等教育出版社

➤ 《计算机组成原理--学习指导与习题解答》唐朔飞 高等教育出版社

参考书目

《Computer Organization and Design》 hardware and software interface, Patterson and Hennessy, 4th Edition, Morgan Kaufmann Pub.

➤ 中文版: 《计算机组成和设计: 硬件/软件接口 (第4版或第5版)》 (美) 帕特森、亨尼西 著

➤ 《Computer Systems A Programmer's Perspective》 Randal E. Bryant David O' Hallaron, Prentice-Hall International Inc.

➤ 中文译本: 《深入理解计算机系统》

➤ 《计算机组成与系统结构》 袁春风编著 清华大学出版社

➤ 《计算机系统基础》 袁春风编著 机械工业出版社

章节学时分配

<u>第1章 计算机系统概论</u>	0.5周
<u>第2章 计算机的发展及应用</u>	0.5周
<u>第3章 系统总线</u>	0.5周
<u>第4章 存储器</u>	1周
<u>第5章 输入输出系统</u>	0.5周
<u>第6章 计算机的运算方法</u>	3周
<u>第7章 指令系统</u>	2周
<u>第8章 CPU 的结构和功能</u>	2周
<u>第9章 控制单元的功能</u>	2周
<u>第10章 控制单元的设计</u>	2周

第1章 计算机系统概论

第1章 计算机系统概论

1.1 计算机系统简介

1.2 计算机的基本组成

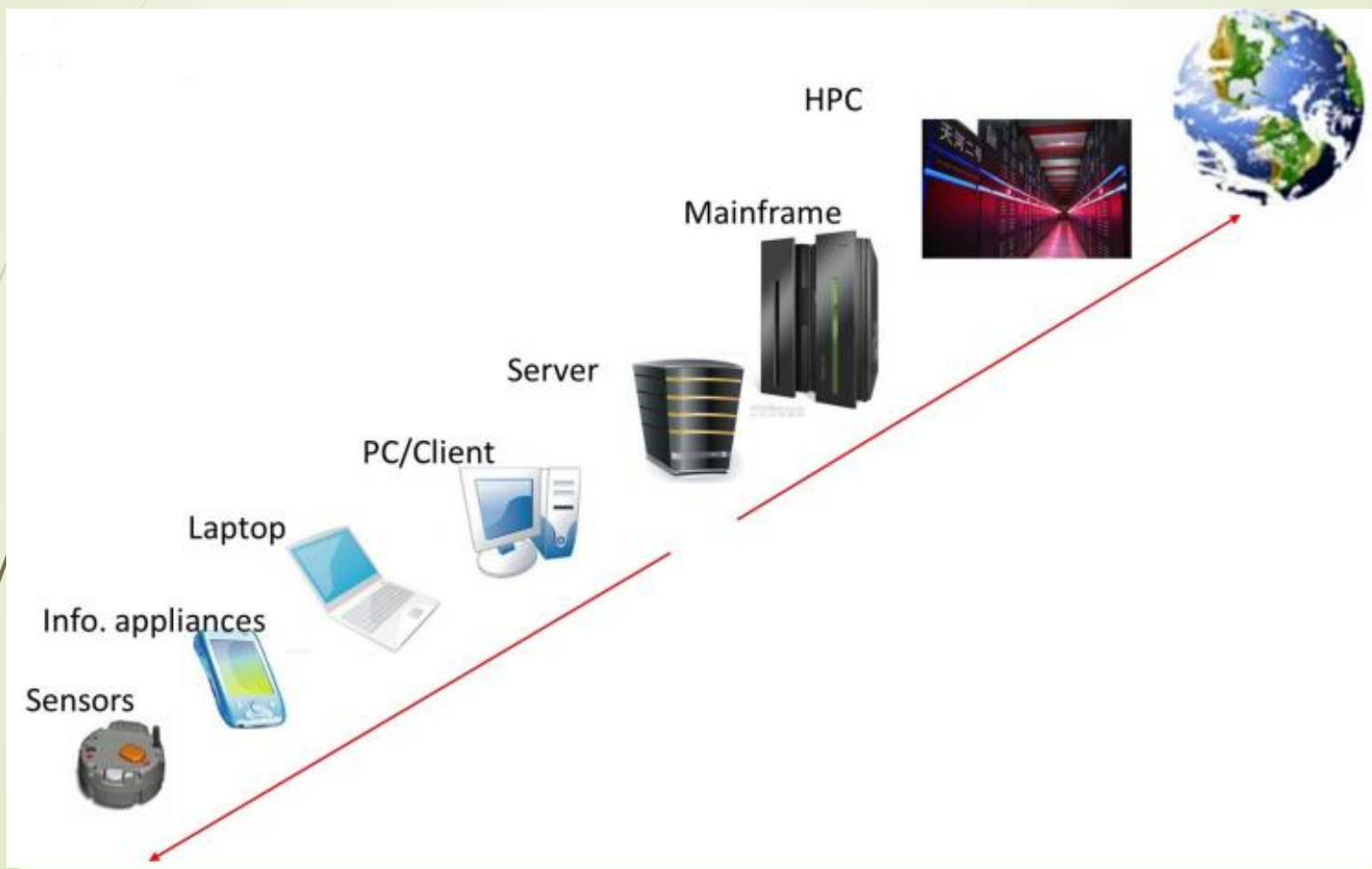
1.3 计算机硬件的主要技术指标

1.4 本书结构

第1章 计算机系统概论

1.1 计算机系统简介

现代计算机的多态性

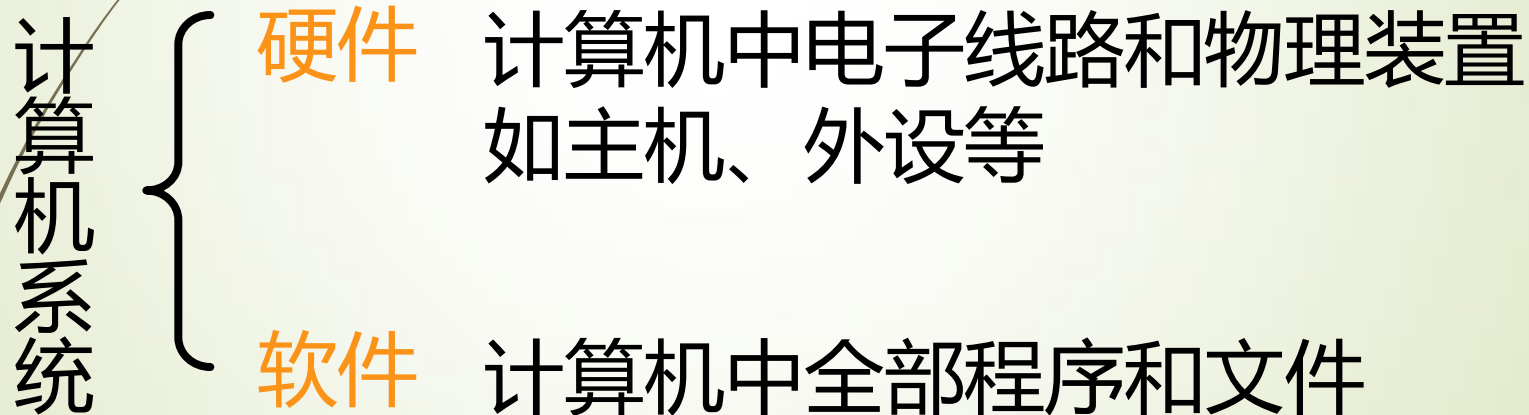


1.1 计算机系统概述



一、计算机软、硬件的概念

1. 计算机系统



软件

系统软件

用来管理整个计算机系统

操作系统

语言处理程序

服务性程序

数据库管理系统

网络软件

应用软件

按任务需要编制成的各种程序

软件

❖ System software(系统软件) - 简化编程过程，使硬件资源被有效利用

- 操作系统 (Operating System) : 硬件资源管理，用户接口
- 语言处理系统：翻译程序+ Linker, Debug, Loader, etc ...
 - 翻译程序(Translator)有三类：
 - 汇编程序(Assembler)：汇编语言源程序→机器语言目标程序
 - 编译程序(Compiler)：高级语言源程序→汇编/机器语言目标程序
 - 解释程序(Interpreter)：将高级语言语句逐条翻译成机器指令并立即执行,不生成目标文件。
- 其他实用程序: 如：磁盘碎片整理程序、备份程序等

❖ Application software(应用软件) - 解决具体应用问题/完成具体应用任务

- 各类媒体处理程序：Word/ Image/ Graphics/...
- 管理信息系统 (MIS)
- Game, ...

翻译方式

编译：使用编译程序把高级语言的源程序**翻译成机器代码**的目标程序，并以**文件形式保留**，然后执行。

- C, PASCAL

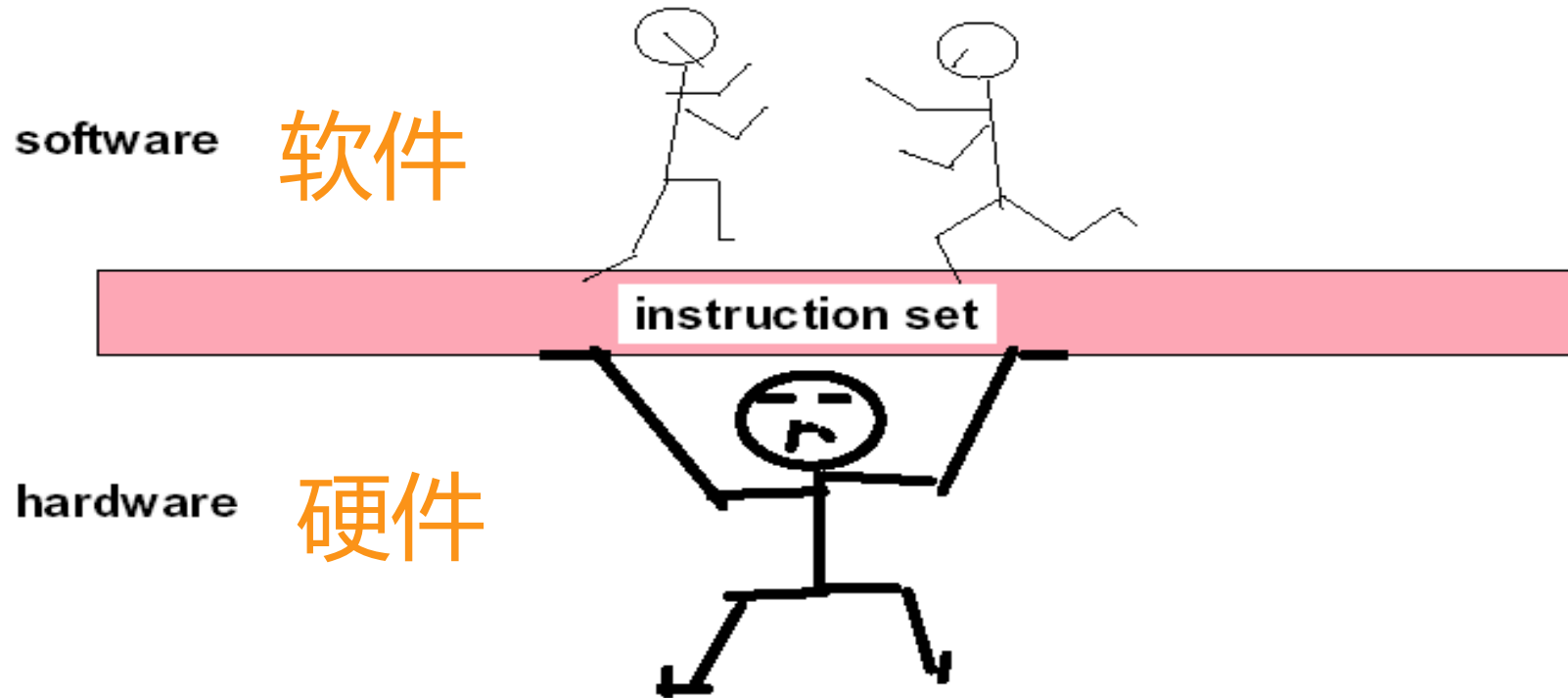
- 大多数编译程序直接产生机器语言的目标代码，形成可执行的目标文件，少数先产生汇编语言一级的符号代码文件，然后再调用汇编程序进行翻译加工处理，最后产生可执行的机器语言目标文件。

- **解释**：使用解释程序，将源程序的一条语句翻译成对应机器语言的语句，并**立即执行这条语句**，接着再翻译源程序的**下一条语句**，并**立即执行**，并**不形成**目标程序，如此重复，直至完成源程序的全部翻译任务

- JAVA

- BASIC的翻译有解释和编译两种方式。

Hardware/Software Interface (界面)



软件和硬件的界面： ISA (Instruction Set Architecture)

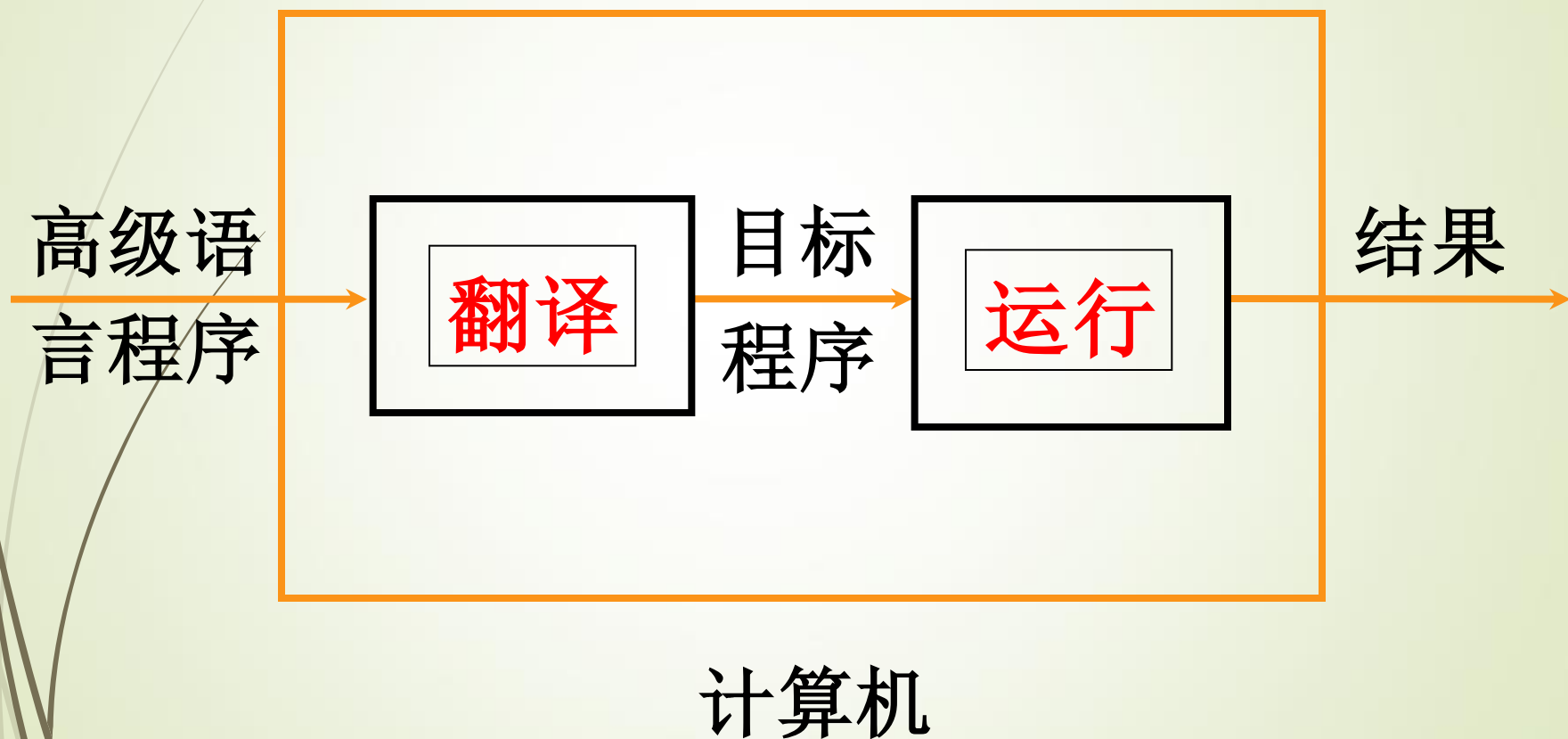
指令集体系结构

机器语言由指令代码构成，能被硬件直接执行。

2. 计算机语言

- 1) **机器语言**：二进制代码表示的计算机语言，可以直接执行。
- 2) **汇编语言**：用助记符（如ADD、SUB、...）编写的语言，需要通过汇编程序翻译成目标程序后才执行。
- 3) **高级语言**：如PASCAL，C，JAVA等，便于编写、阅读，接近自然语言，提高了软件的产量，便于移植，独立于硬件环境。

3. 计算机的解题过程

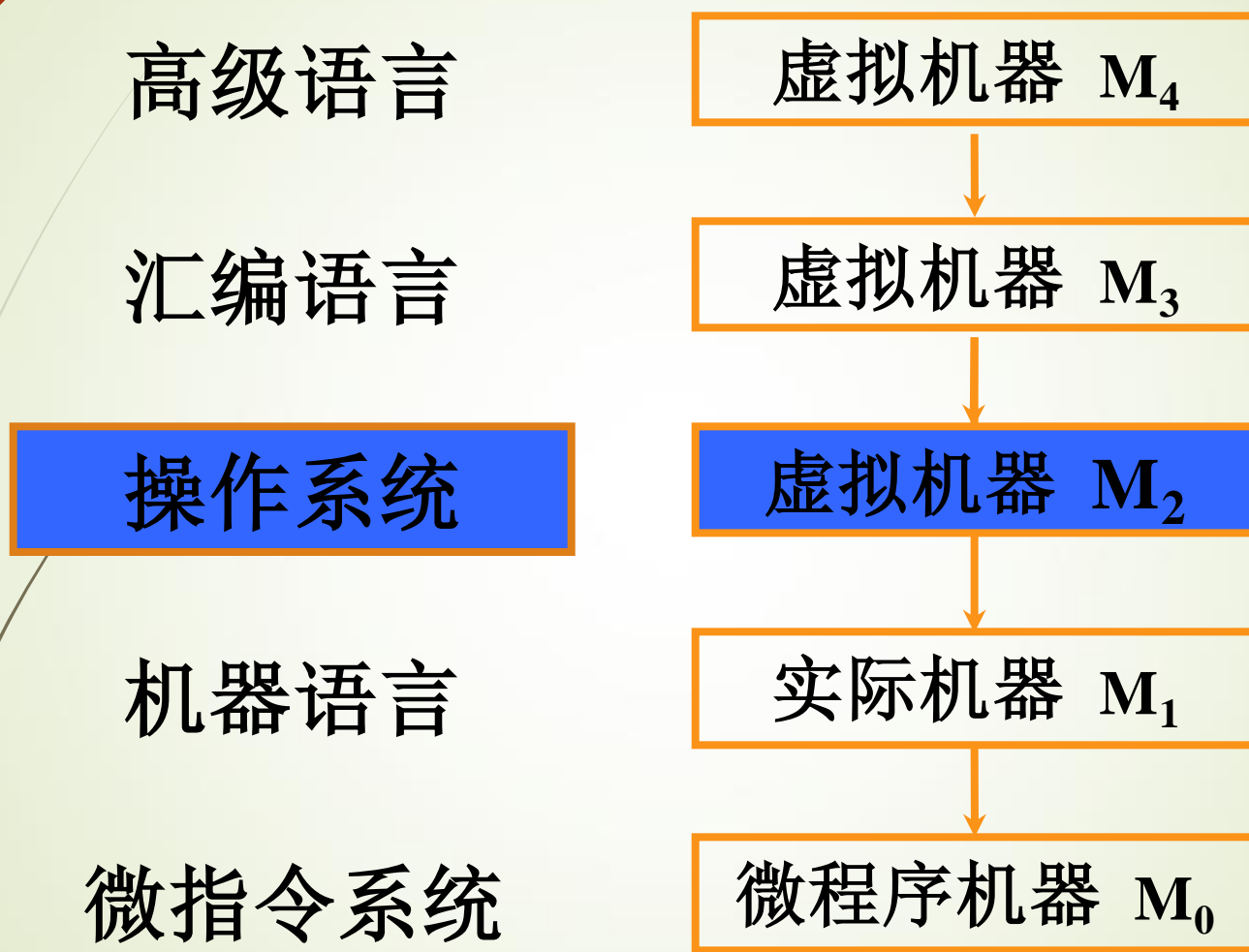


下列_____不属于高级语言。

- ☐ A C语言
- ☒ B 汇编语言
- ☐ C Java语言
- ☐ D Pascal语言

提交

二、计算机系统的层次结构



二、计算机系统的层次结构



- **实际机器**：指由**硬件**或**固件**实现的机器。
- **虚拟机器**：指以**软件**为主实现的机器。
 - 虚拟机器只对该级的观察者存在，某一级观察者只需通过**该级的语言**来了解和使用计算机，不必关心下级是如何工作和实现的。
 - 高级语言级的用户可以不了解机器的具体组成，不熟悉指令系统，直接用所指定的语言描述所要解决的问题。

二、计算机系统的层次结构



软件

虚拟机 M4

用编译程序翻译成汇编语言程序

虚拟机 M3

用汇编程序翻译成机器语言程序

虚拟机 M2

用机器语言解释操作系统

硬件

实际机器 M1

用微指令解释机器指令

微程序机器 M0

由硬件直接执行微指令

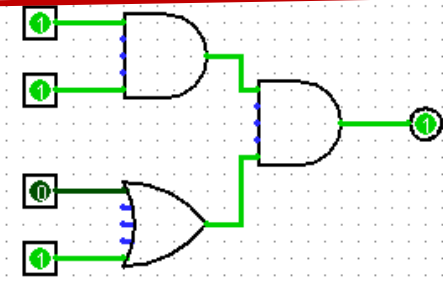
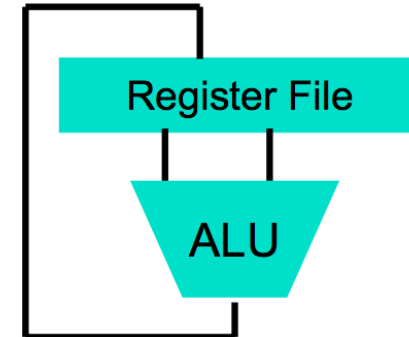
高级语言

```
temp = v[k];  
v[k] = v[k+1];  
v[k+1] = temp;
```

Anything can be represented
as a *number*,
i.e., data or instructions

```
lw  $t0, 0($2)  
lw  $t1, 4($2)  
sw  $t1, 0($2)  
sw  $t0, 4($2)
```

```
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
```



High Level Language
Program (e.g., C)

Compiler

Assembly Language Program (e.g.,
MIPS)

Assembler

Machine Language Program
(MIPS)

*Machine
Interpretation*

Hardware Architecture Description
(e.g., block diagrams)

*Architecture
Implementation*

Logic Circuit Description
(Circuit Schematic Diagrams)

三、计算机体系结构和计算机组成

1.1

计算机 体系结构

有无乘法指令

程序员所见到的**计算机系统的属性**
概念性的结构与功能特性

（指令系统、数据类型、寻址技术、I/O机理）

计算机 组成

实现计算机体系结构所体现的**属性**

（具体指令的实现）

如何实现乘法指令

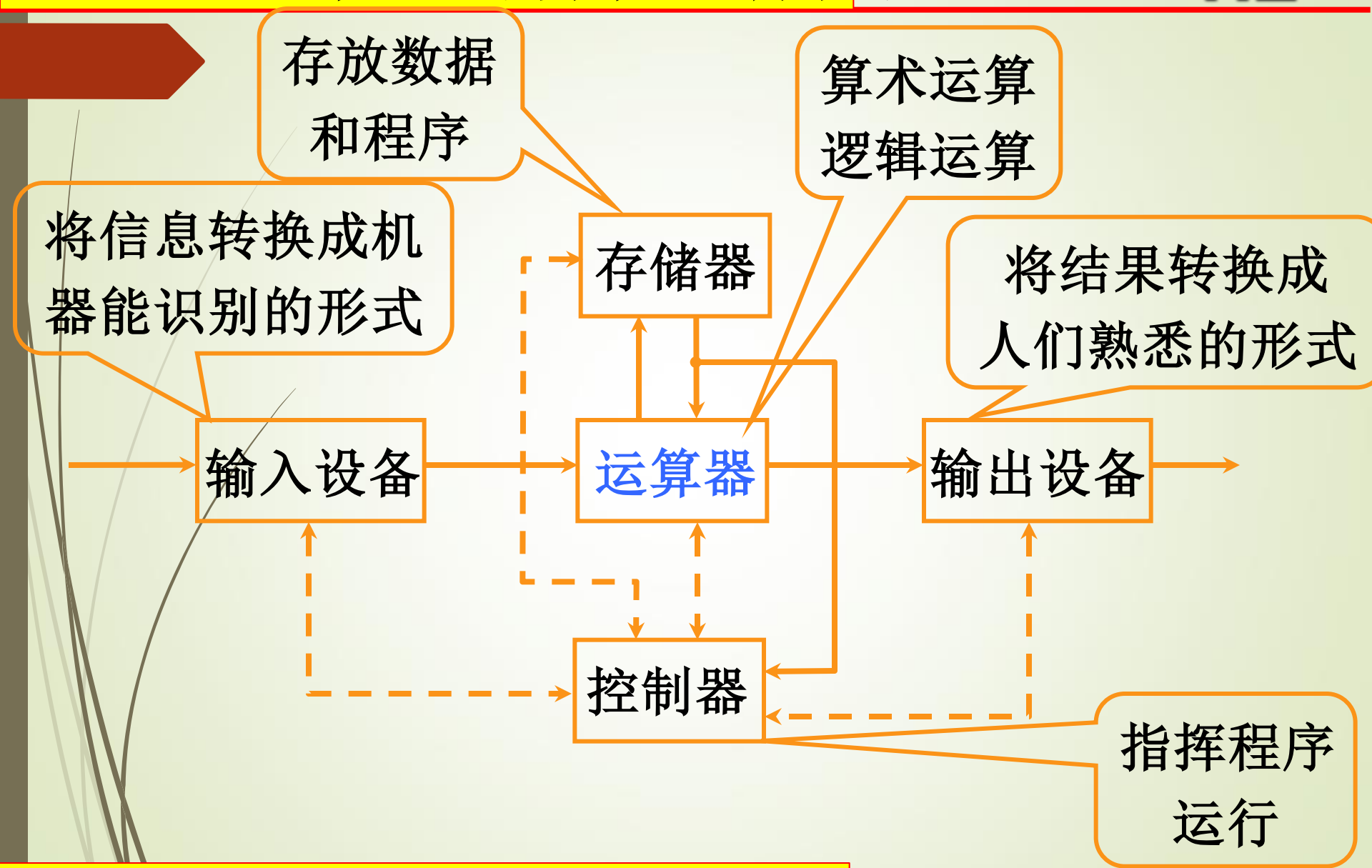
1.2 计算机的基本组成

一、冯·诺依曼计算机的特点

1. 计算机由五大部件组成
2. 指令和数据以同等地位存于存储器，可按地址寻访
3. 指令和数据用二进制表示
4. 指令由操作码和地址码组成
5. 存储程序
6. 以**运算器**为中心

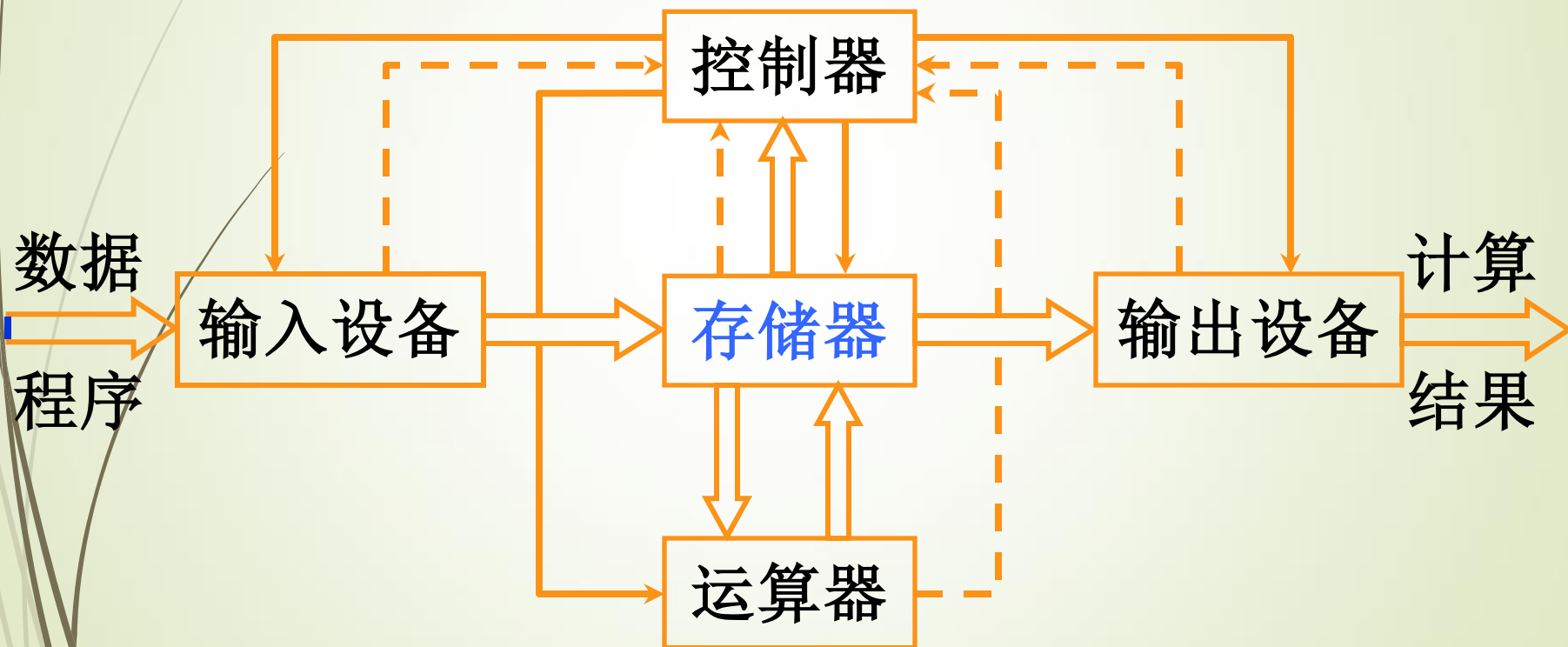
冯·诺依曼计算机硬件框图

1.2



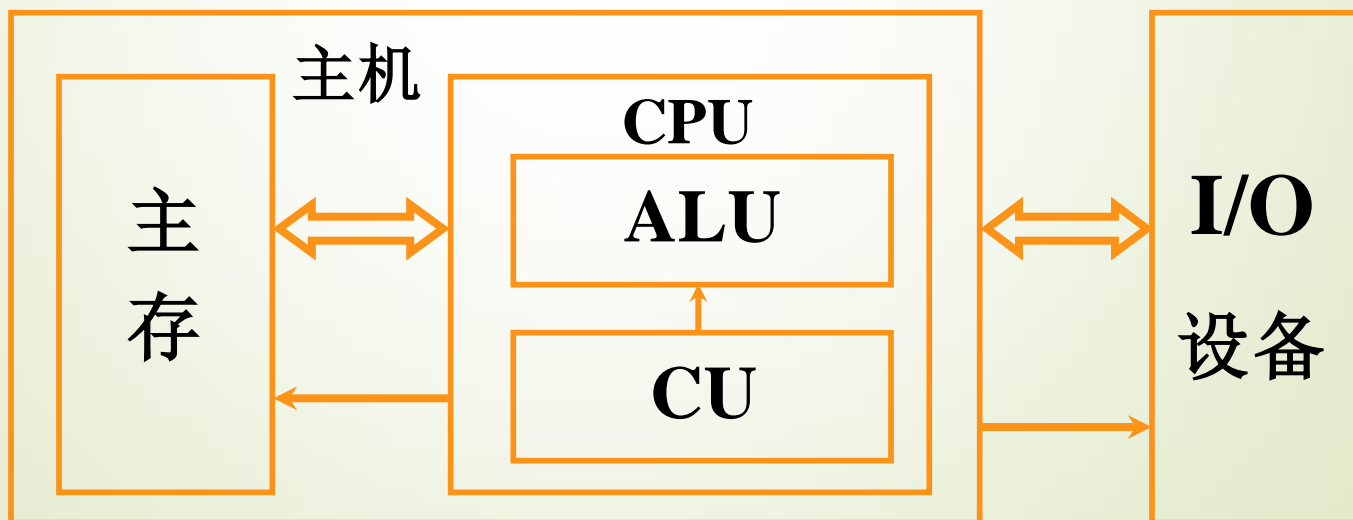
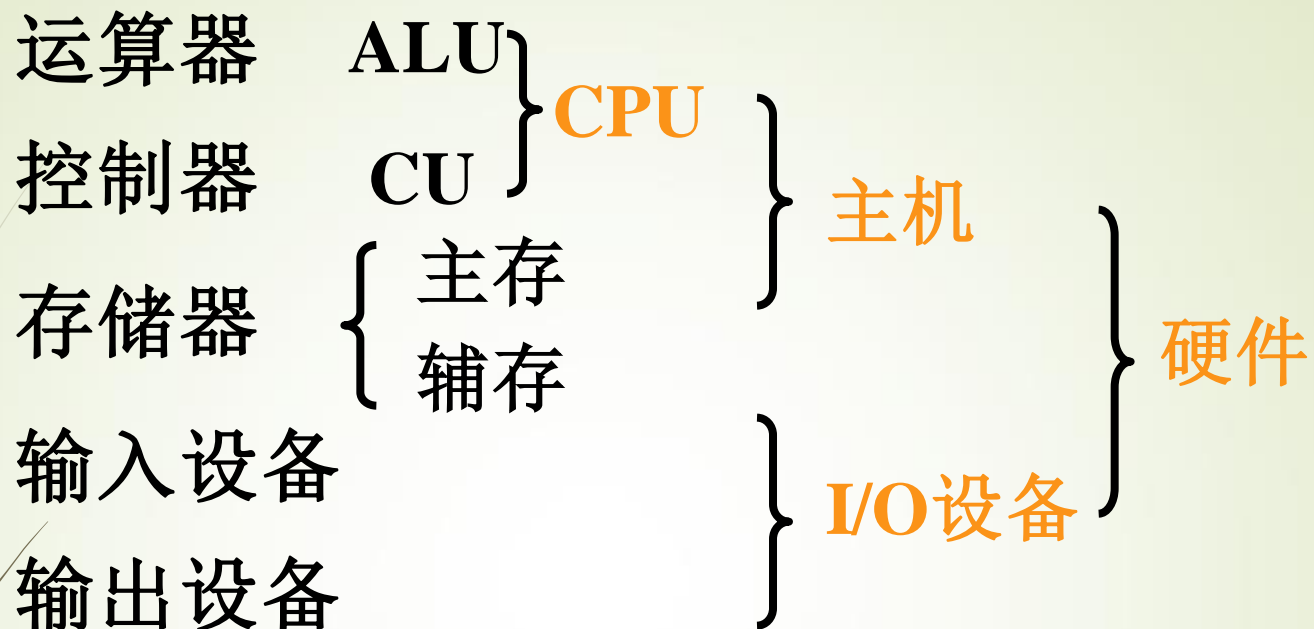
二、计算机硬件框图

1. 以存储器为中心的计算机硬件框图（现代）

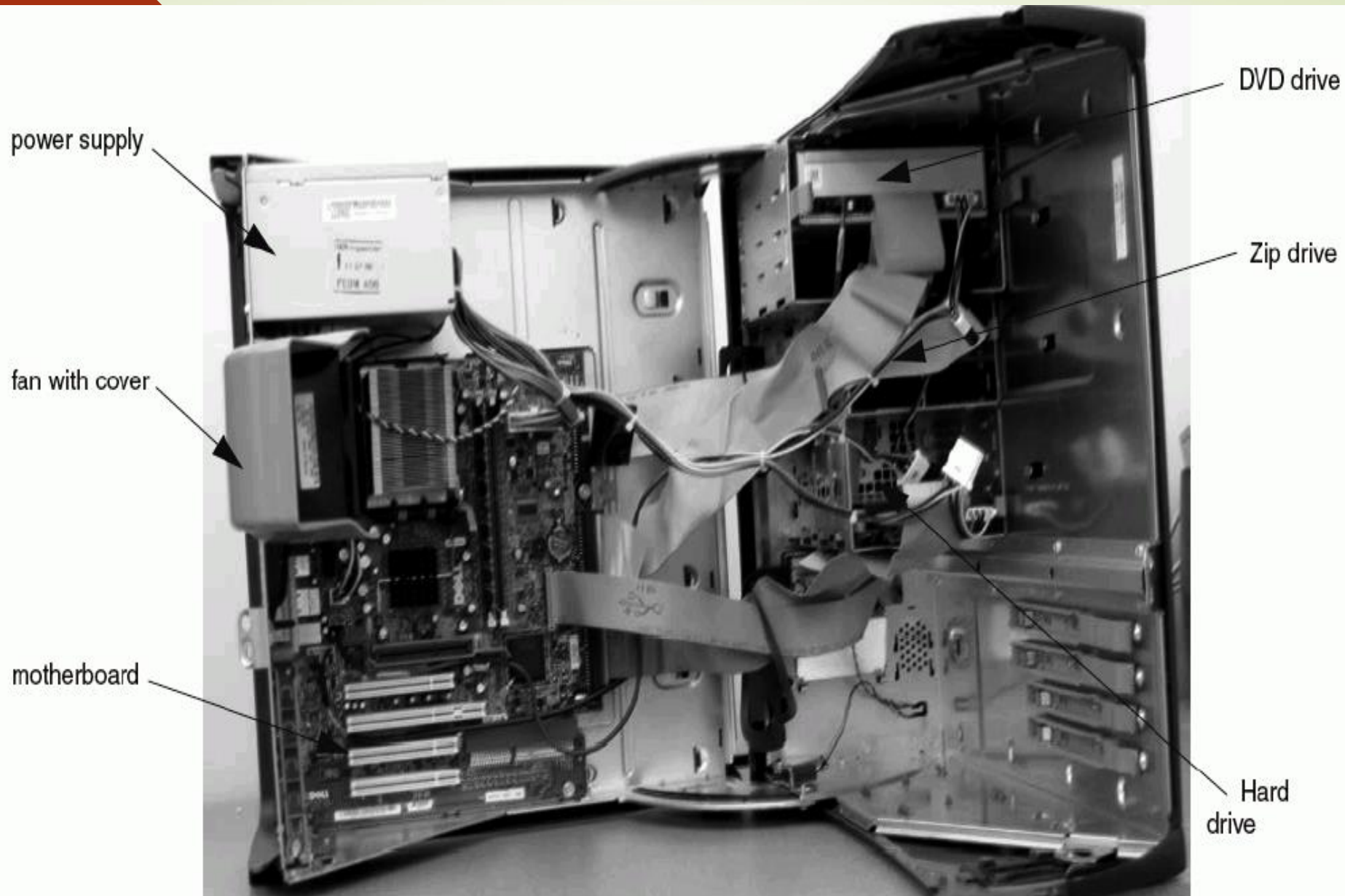


2. 现代计算机硬件框图

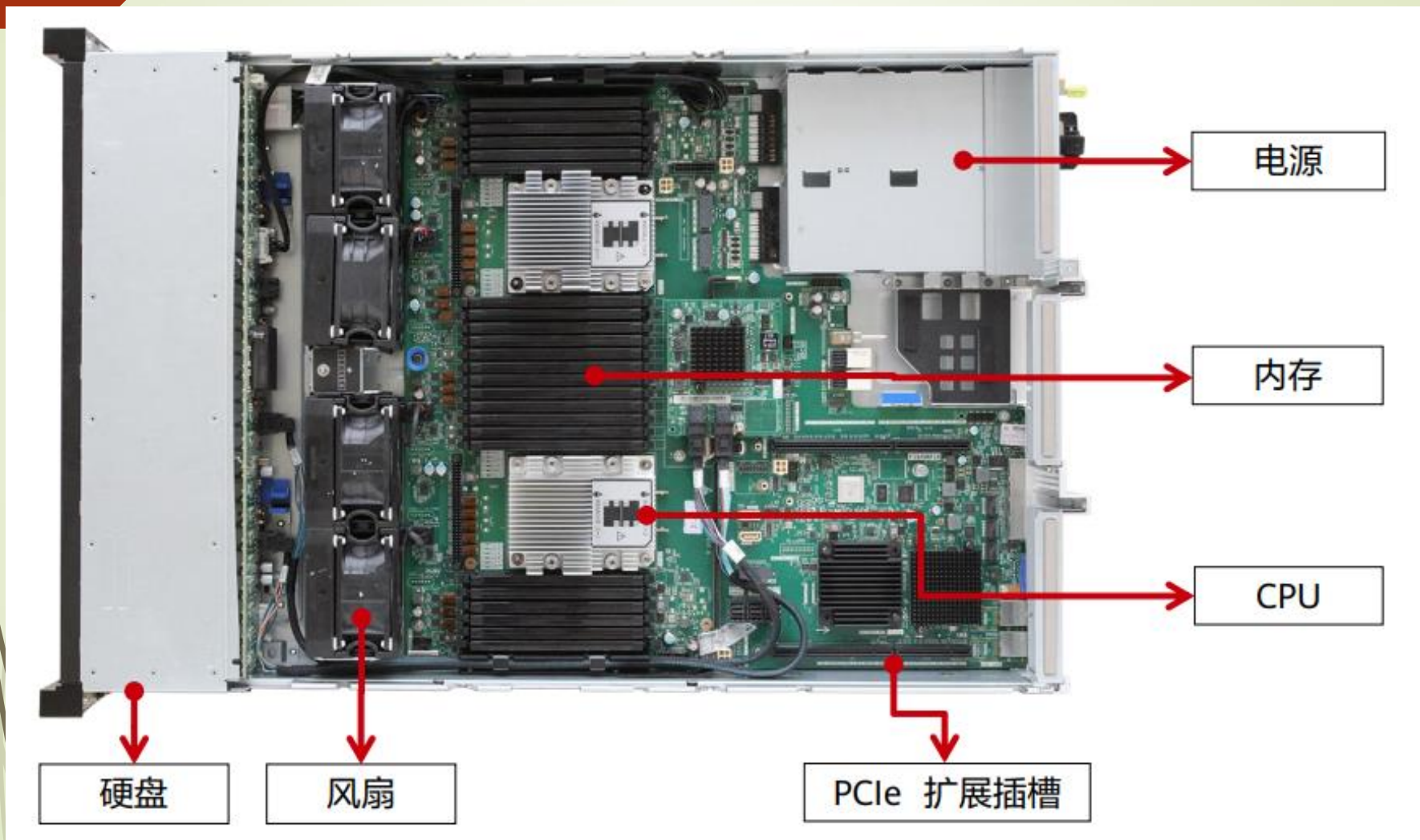
1.2



计算机硬件：打开计算机来看看



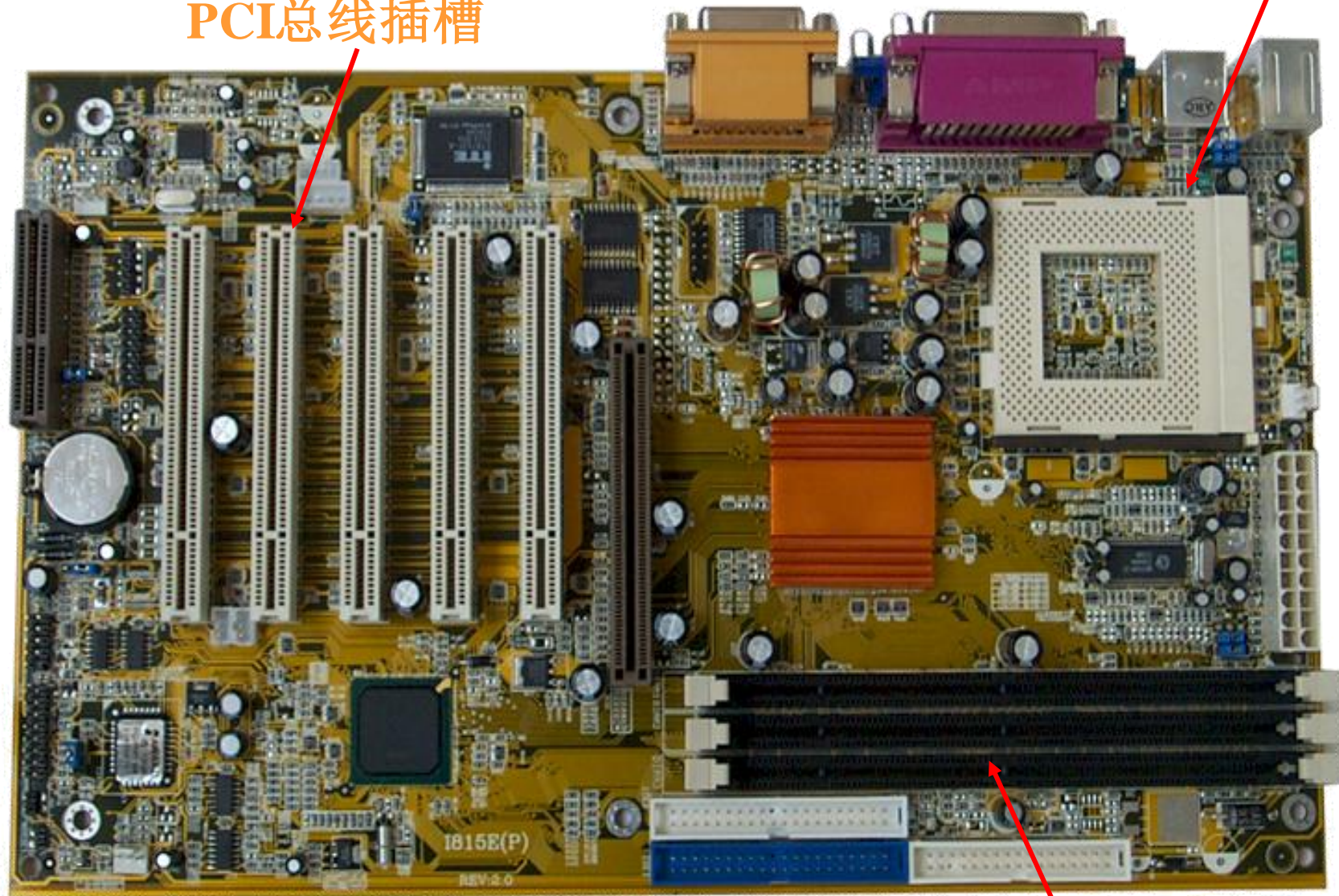
刀片式服务器内部视图



PC主板

PCI总线插槽

CPU插座



内存条

冯·诺依曼计算机中，以_____为中心。

- ☐ A 指令
- ☐ B 存储器
- ☒ C 运算器
- ☐ D 输入输出设备

提交

1. 上机前的准备

- 建立数学模型

- 确定计算方法

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \dots$$

$$\sqrt{x} = \frac{1}{2} \left(y_n + \frac{x}{y_n} \right) \quad (n = 0, 1, 2, \dots)$$

- 编制解题程序

程序 —— 运算的 全部步骤

指令 —— 每 一个步骤

计算 $ax^2 + bx + c = (ax + b)x + c$

取 x 至运算器中

乘以 x 在运算器中

乘以 a 在运算器中

存 ax^2 在存储器中

取 b 至运算器中

乘以 x 在运算器中

加 ax^2 在运算器中

加 c 在运算器中

取 x 至运算器中

乘以 a 在运算器中

加 b 在运算器中

乘以 x 在运算器中

加 c 在运算器中

操作码

地址码

取数

α

$[\alpha] \rightarrow \text{ACC}$

000001

0000001000

存数

β

$[\text{ACC}] \rightarrow \beta$

加

γ

$[\text{ACC}] + [\gamma] \rightarrow \text{ACC}$

乘

δ

$[\text{ACC}] \times [\delta] \rightarrow \text{ACC}$

打印

σ

$[\sigma] \rightarrow \text{打印机}$

停机

计算 $ax^2 + bx + c$ 程序清单(假定指令字长16位: 6+10)

指令和数据存于主存单元的地址	指令		注释
	操作码	地址码	
0	000001	0000001000	取数 x 至ACC
1	000100	0000001001	乘 a 得 ax ,存于ACC中
2	000011	0000001010	加 b 得 $ax+b$,存于ACC中
3	000100	0000001000	乘 x 得 $(ax+b)x$,存于ACC中
4	000011	0000001011	加 c 得 $ax^2 + bx + c$,存于ACC
5	000010	0000001100	将 $ax^2 + bx + c$,存于主存单元
6	000101	0000001100	打印
7	000110		停机
8		x	原始数据 x
9		a	原始数据 a
10		b	原始数据 b
11		c	原始数据 c
12			存放结果

指令

数据

2. 计算机的解题过程

(1) 存储器的基本组成



存储体 – 存储单元 – 存储元件 (0/1)

大楼 – 房间 – 床位 (无人/有人)

存储单元 存放一串二进制代码

存储字 存储单元中二进制代码的组合

存储字长 存储单元中二进制代码的位数

每个存储单元赋予一个地址号

1	0	1	0	1	0	0	1
1	1	0	0	0	0	0	1
1	0	0	1	0	0	1	1
1	1	1	0	0	0	1	0
0	1	1	0	1	1	0	1
1	0	1	0	1	0	0	1

按地址寻访

(1) 存储器的基本组成



MAR 存储器地址寄存器（参见第4章）
反映**存储单元**的个数

MDR 存储器数据寄存器
反映**存储字长**

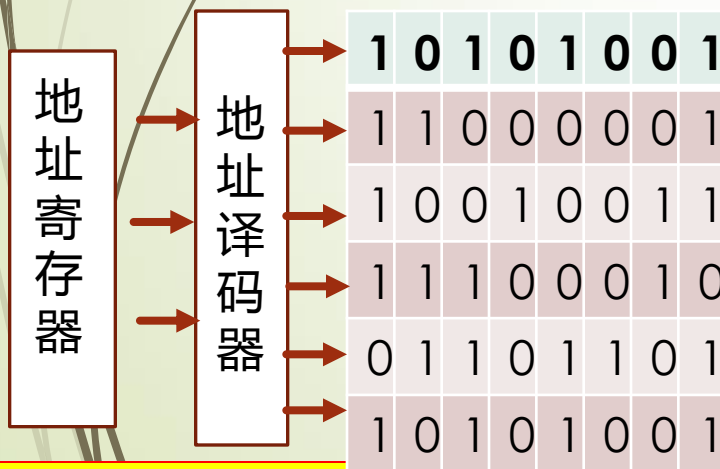
设 **MAR=4位**

MDR=8位

存储单元个数 $2^4 = 16$

存储字长 8

存储容量=16*8=128位

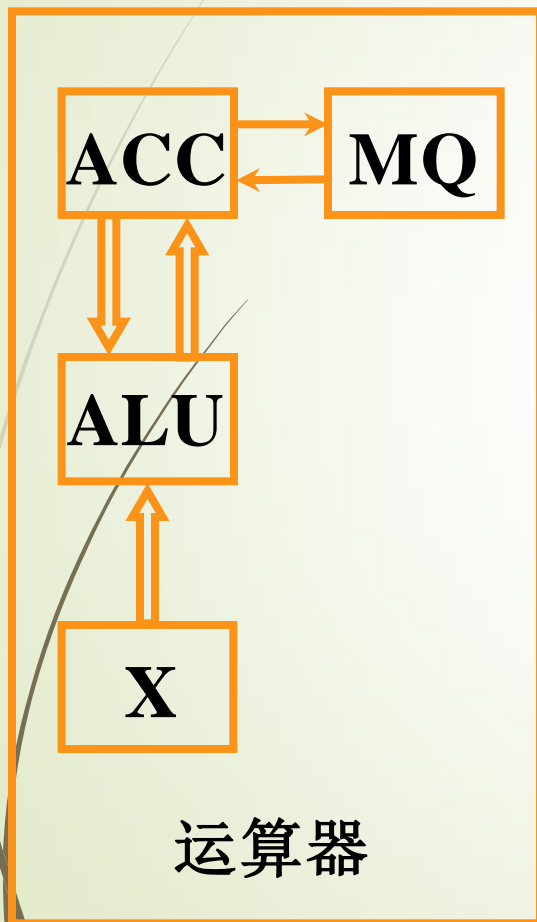


MAR: Memory Address Register

MDR: Memory Data Register

(2)运算器的基本组成及操作过程

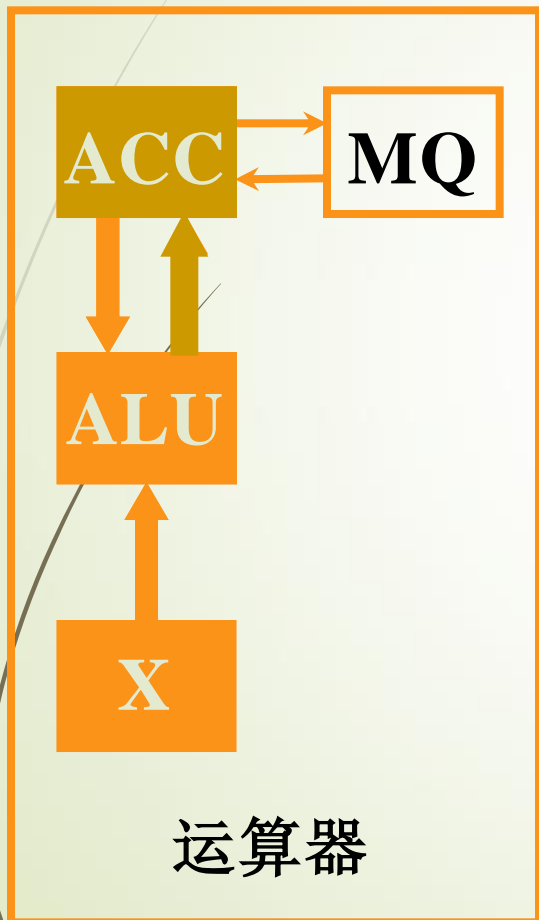
不同的运算，所需要的存储单元的数量是不同的。



	ACC	MQ	X
加法	被加数 和		加数
减法	被减数 差		减数
乘法	乘积高位	乘数 乘积低位	被乘数
除法	被除数 余数	商	除数

① 加法操作过程

例如： $x1+y1=z1$



指令

加

M

初态

ACC

被加数

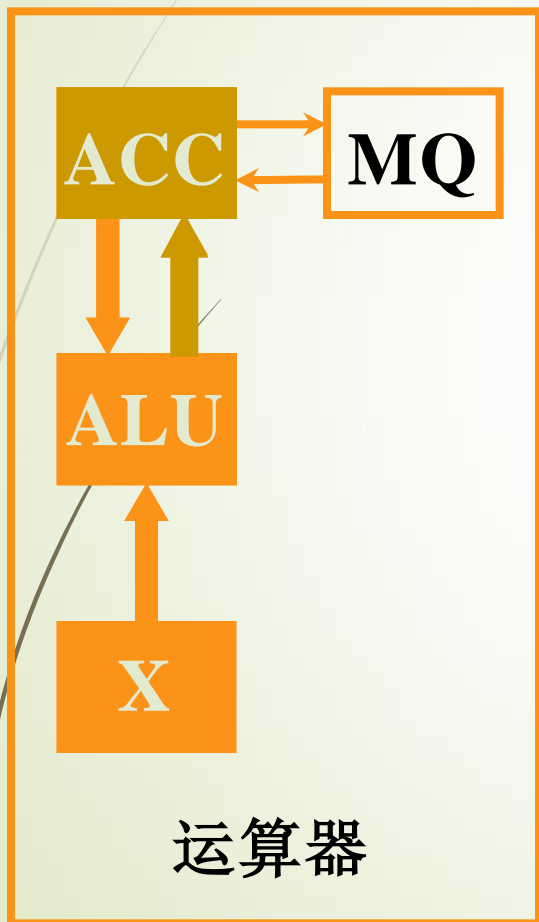
$[M] \longrightarrow$

X

$[ACC] + [X] \longrightarrow ACC$

② 减法操作过程

例如： $x1 - y1 = z1$



指令

减

M

初态

ACC

被减数

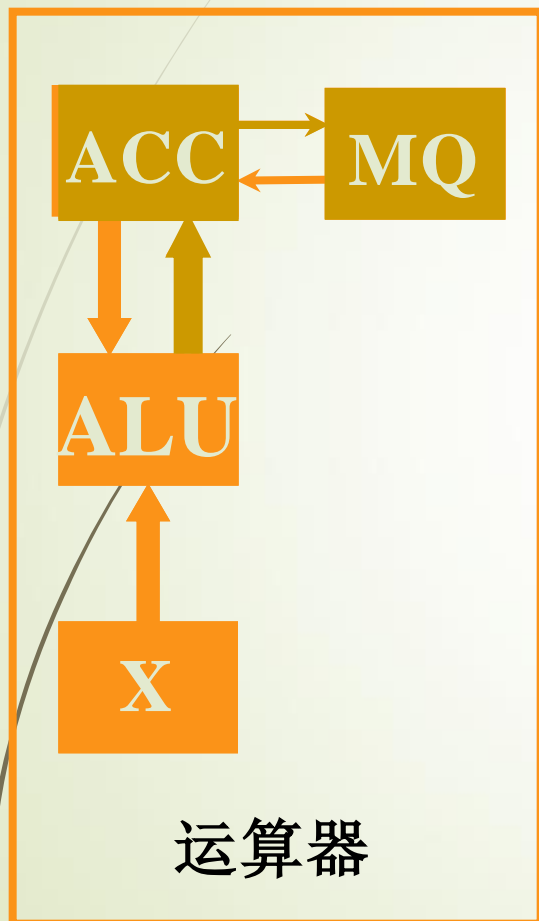
$[M] \longrightarrow X$

$[ACC] - [X] \longrightarrow ACC$

③ 乘法操作过程

1.2

例如： $x1 * y1 = z1$ (高位) $w1$ (低位)



指令

乘

M

初态

ACC

被乘数

$[M] \longrightarrow MQ$

$[ACC] \longrightarrow X$

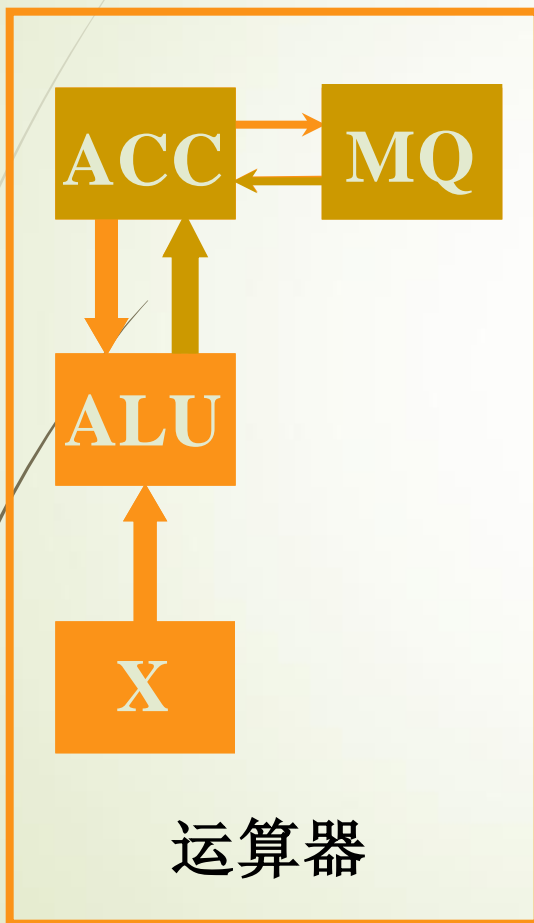
$0 \longrightarrow ACC$

$[X] \times [MQ] \longrightarrow ACC // MQ$

④ 除法操作过程

1.2

例如： $x1 / y1 = z1$ (商) $w1$ (余数)



指令

除

M

初态

ACC

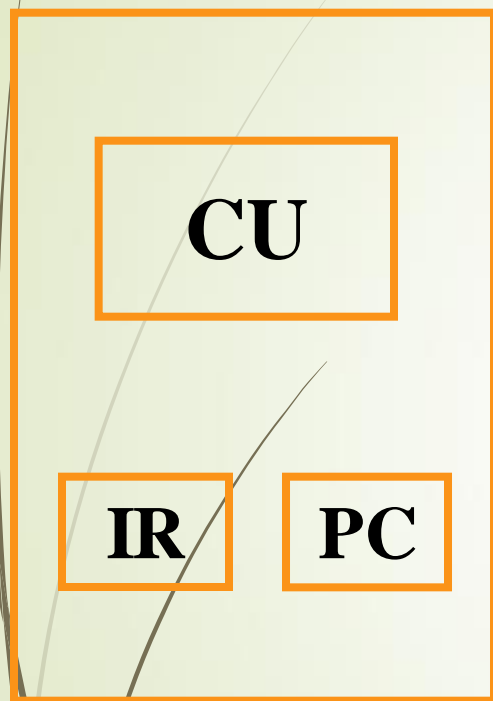
被除数

$[M] \rightarrow X$

$[ACC] \div [X] \rightarrow MQ$

余数在ACC中

(3) 控制器的基本组成



完成 一条 指令	{	取指令	PC	}	取指 访存
		分析指令	IR		
		执行指令	CU		

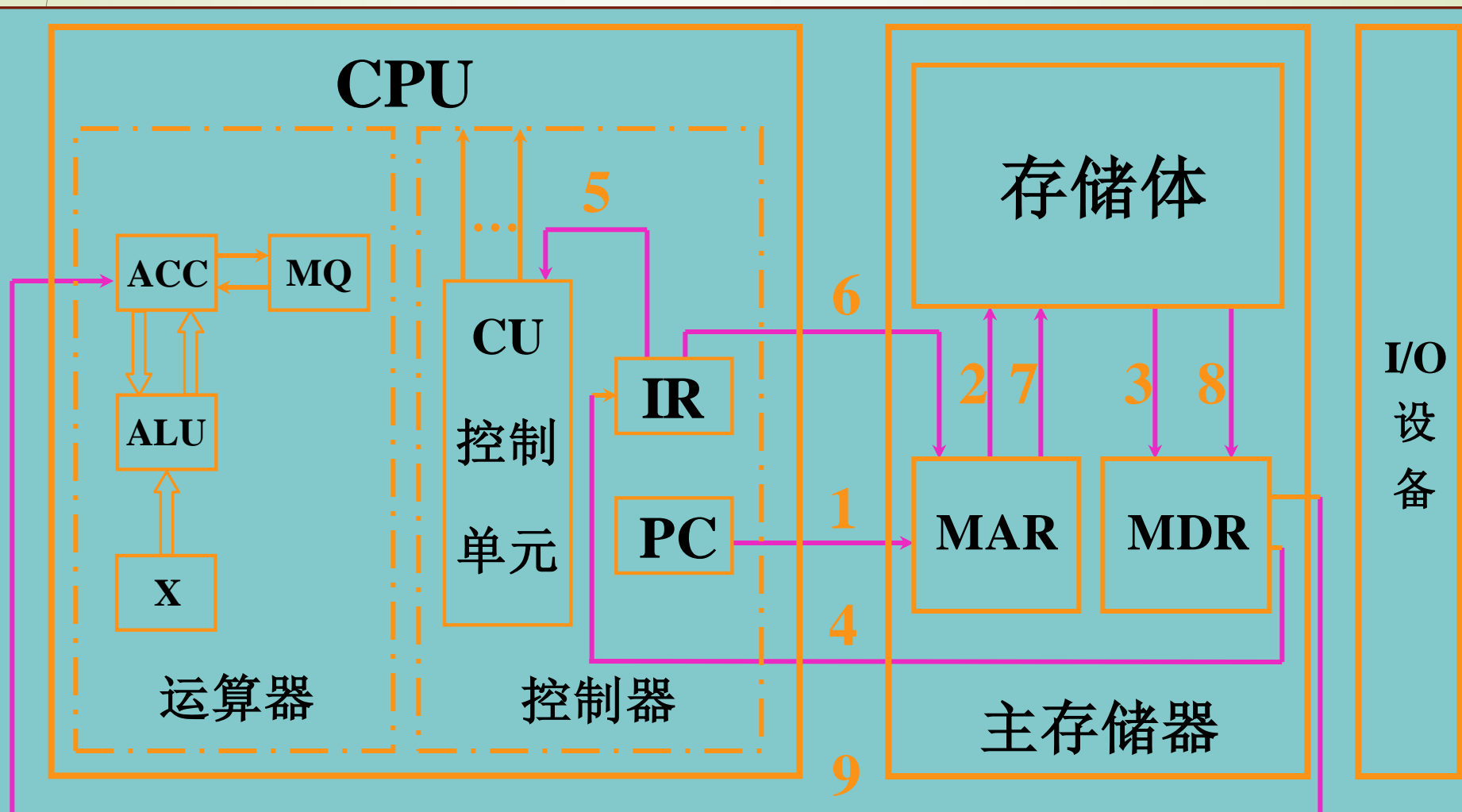
PC 存放当前欲执行指令的地址，
具有计数功能 $(PC) + 1 \rightarrow PC$

IR 存放当前欲执行的指令

CU 控制部件，发出对应指令的控制信号

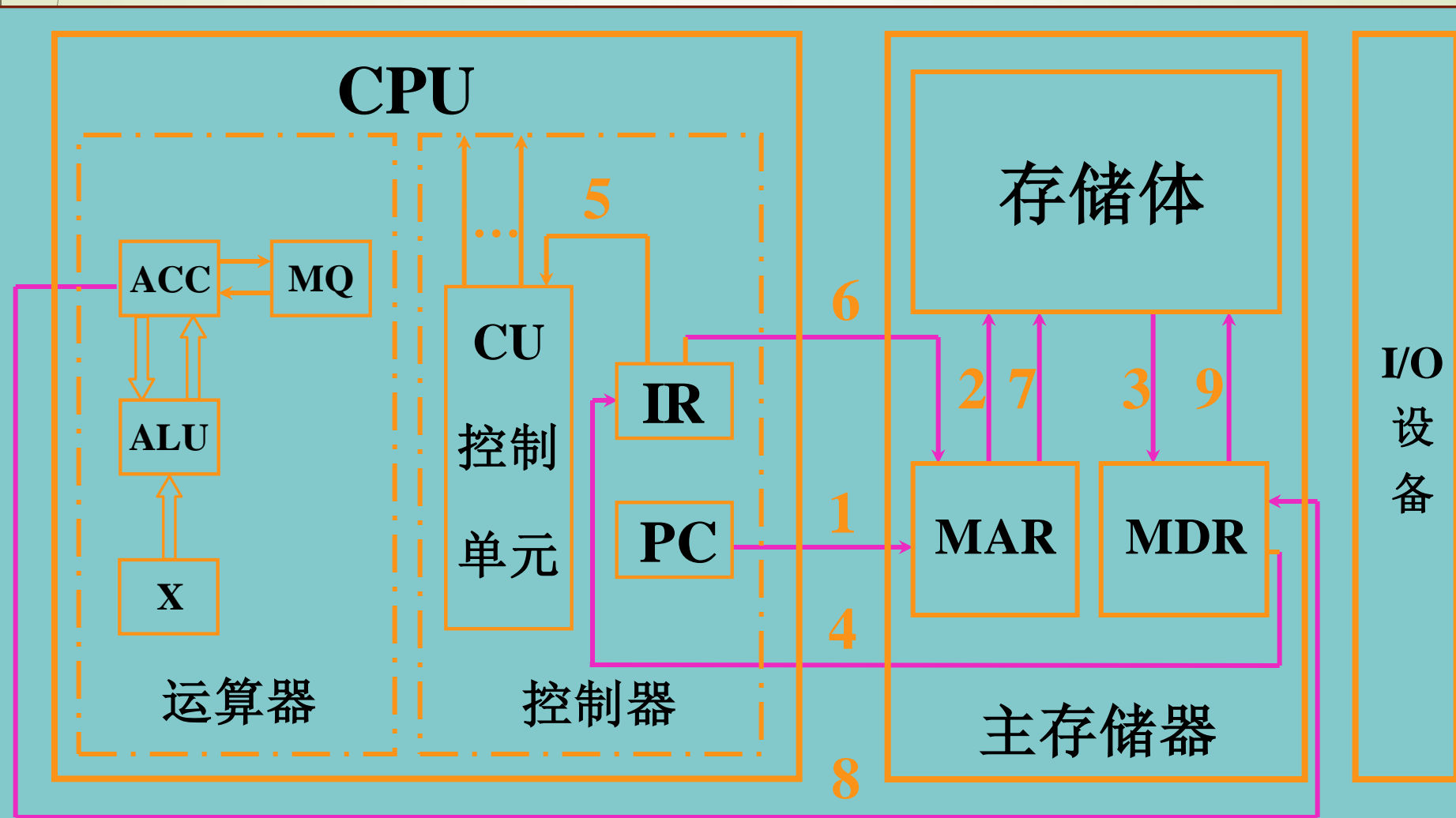
(4) 主机完成一条指令的过程

以取数指令为例



(4) 主机完成一条指令的过程

以存数指令为例



(5) $ax^2 + bx + c$ 程序的运行过程

- 将程序通过输入设备送至计算机
- 程序首地址 \longrightarrow PC
- 启动程序运行
- 取指令 $PC \rightarrow MAR \rightarrow M \rightarrow MDR \rightarrow IR, (PC) + 1 \rightarrow PC$
- 分析指令 $OP(IR) \rightarrow CU$
- 执行指令 $Ad(IR) \rightarrow MAR \rightarrow M \rightarrow MDR \rightarrow ACC$
- \vdots
- 打印结果
- 停机

1.3 计算机硬件的主要技术指标（掌握）

1. 机器字长

CPU 一次能处理数据的位数
与 CPU 中的 寄存器位数 有关

主频

吉普森法

$$T_M = \sum_{i=1}^n f_i t_i$$

其中 f_i 是第 i 种指令占有所有操作的百分比。
 t_i 是第 i 条指令的执行时间。

2. 运算速度

MIPS

每秒执行百万条指令

CPI

执行一条指令所需时钟周期数

FLOPS

每秒浮点运算次数

3. 存储容量

1.3

存放二进制信息的**总位数**

存储单元个数 \times 存储字长

主存**容量**

如 **MAR** **MDR** **容量**

10 **8** **1 K \times 8位**

16 **32** **64 K \times 32位**

$$1\text{K} = 2^{10}$$

字节数

如

$$2^{13} = 1 \text{ KB}$$

$$2^{21} = 256 \text{ KB}$$

$$1\text{B} = 2^3\text{b}$$

辅存**容量**

字节数

80 GB

$$1\text{GB} = 2^{30}\text{b}$$

此题未设置答案，请点击右侧设置按钮

下列选项中，描述浮点数操作速度指标的是：_____。

- ☐ A MIPS
- ☐ B CPI
- ☐ C 主频
- ☐ D FLOPS

提交

几个重要概念及技术指标

- **机器字长**: CPU 一次能处理数据的位数与 CPU 中的 **寄存器位数** 有关
- **时钟周期**(clock cycle,tick,clock tick,clock):又称为节拍或T周期, 计算机执行指令的过程被分成若干步骤和相应的动作来完成, 每一步动作都要有相应的控制信号进行控制, 这些控制信号何时发出、作用时间多长, 都要有相应的定时信号进行同步。CPU 产生的同步时钟定时信号, 也就是CPU主脉冲信号, 宽度称为时钟周期
- **时钟频率** (clock rate,主频): CPU中的主脉冲信号的时钟频率, 是CPU时钟周期的倒数。

几个重要概念及技术指标



- **CPI** (Cycle Per Instruction) : 执行一条指令所需时钟周期数
- 吉普森法(Gibson): 根据不同类型指令在计算过程中出现的频繁程度, 乘以不同系数, 求得统计平均值, 即**平均运算速度**。

$$T_M = \sum_{i=1}^n f_i t_i$$

T_M : 机器运行速度;

f_i : 第*i*种指令占全部操作的百分比;

t_i : 第*i*种指令的执行时间

几个重要概念及技术指标

• CPU执行时间的计算：

$$\begin{aligned}\text{CPU 执行时间} &= \text{CPU时钟周期数 (程序)} \times \text{时钟周期} \\ &= \text{CPU时钟周期数 (程序)} \div \text{时钟频率} \\ &= \text{指令条数 (程序)} \times \text{CPI} \times \text{时钟周期}\end{aligned}$$

$$\text{CPU时钟周期数 (程序)} = \text{指令条数 (程序)} \times \text{CPI}$$

$$\text{CPI} = \text{CPU时钟周期数 (程序)} \div \text{指令条数 (程序)}$$

CPI 用来衡量以下各方面的综合结果

- **Instruction Set Architecture (ISA)**
- **Implementation of that architecture**
- **Program (Compiler、 Algorithm)**

如何计算CPI？

对于某一条特定的指令而言，其CPI是一个确定的值。
但是，对于某一类指令、或一个程序、或一台机器而言，其CPI是一个平均值，表示该类指令或该程序或该机器的指令集中每条指令执行时平均需要多少时钟周期。

假定 CPI_i 和 C_i 分别为第 i 类指令的CPI和指令条数，则程序的总时钟数为：

$$\text{总时钟数} = \sum_{i=1}^n CPI_i \times C_i \quad \text{所以,} \quad \text{CPU时间} = \text{时钟周期} \times \sum_{i=1}^n CPI_i \times C_i$$

假定 CPI_i 、 F_i 是各指令CPI和在程序中的出现频率，则程序综合CPI为：

$$CPI = \sum_{i=1}^n CPI_i \times F_i \quad \text{where} \quad F_i = \frac{C_i}{Instruction_Count}$$

已知CPU时间、时钟频率、总时钟数、指令条数，则程序综合CPI为：

$$CPI = (\text{CPU 时间} \times \text{时钟频率}) / \text{指令条数} = \text{总时钟周期数} / \text{指令条数}$$

问题：指令的CPI、机器的CPI、程序的CPI各能反映哪方面的性能？

单靠CPI不能反映CPU的性能！为什么？如：单周期处理器CPI=1，但性能差！

Example 1

程序P在机器A上运行需10 s，机器A的时钟频率为400MHz。现在要设计一台机器B，希望该程序在B上运行只需6 s.

机器B时钟频率的提高导致了其CPI的增加，使得程序P在机器B上时钟周期数是在机器A上的1.2倍。机器B的时钟频率达到A的多少倍才能使程序P在B上执行速度是A上的 $10/6=1.67$ 倍？

Answer:

CPU时间A = 时钟周期数A / 时钟频率A

时钟周期数A = 10 sec x 400MHz = 4000M个

时钟频率B = 时钟周期数B / CPU时间B

= 1.2 x 4000M / 6 sec = 800 MHz

机器B的频率是A的两倍，但机器B的速度并不是A的两倍！

MIPS:平均每秒执行百万条指令数

$$\begin{aligned}\text{MIPS} &= \text{Instruction Count} / \text{Time} \times 10^6 \\ &= \text{Clock Rate} / \text{CPI} \times 10^6\end{aligned}$$

Million **I**nstructions **P**er **S**econd

因为每条指令执行时间不同，所以**MIPS**总是一个平均值。

- 不同机器的指令集不同
- 程序由不同的指令混合而成
- 指令使用的频度动态变化
- **Peak MIPS**: (不实用)

用**MIPS**数表示性能有没有局限？

所以**MIPS**数不能说明性能的好坏（用下页中的例子来说明）

$$\text{MFLOPS} = \text{FP Operations} / \text{Time} \times 10^6$$

Million **F**loating-point **O**perations **P**er **S**econd

- 与机器相关性大
- 并不是程序中花时间的部分

用**MFLOPS**数表示性能也有局限！

Example: MIPS数不可靠!

Assume we build **an optimizing compiler** for the load/store machine. The compiler discards 50% of the ALU instructions.

- 1) What is the CPI ? 仅仅在软件上进行优化，没有涉及到任何硬件措施。
- 2) Assuming a 20 ns clock cycle time (50 MHz clock rate). What is the MIPS rating for optimized code versus unoptimized code? Does the MIPS rating agree with the rating of execution time?

Op	Freq	Cycle	Optimizing compiler	New Freq
ALU	43%	1	$21.5 / (21.5 + 21 + 12 + 24) = 27\%$	27%
Load	21%	2	$21 / (21.5 + 21 + 12 + 24) = 27\%$	27%
Store	12%	2	$12 / (21.5 + 21 + 12 + 24) = 15\%$	15%
Branch	24%	2	$24 / (21.5 + 21 + 12 + 24) = 31\%$	31%
CPI	1.57		$50\text{M} / 1.57 = 31.8\text{MIPS}$	1.73
MIPS	31.8		$50\text{M} / 1.73 = 28.9\text{MIPS}$	28.9

结果：因为优化后减少了**ALU**指令（其他指令数没变），所以程序执行时间一定减少了，但优化后的**MIPS**数反而降低了。

浮点运算速度



➤ FLOPS:

- 1 Megaflop/s指每秒执行100万次浮点运算，即 10^6 次方FLOPS；
- 1 Gigaflop/s指每秒执行10亿次浮点运算，即 10^9 次方FLOPS；
- 1 Teraflop/s指每秒执行**1万亿**次浮点运算，即 10^{12} 次方FLOPS
- 1 Petaflop/s指每秒执行**1000万亿**次浮点运算，即 10^{15} 次方FLOPS
- 1 Exaflop/s即每秒执行100万万亿次浮点运算， 10^{18} 次方FLOPS。

K、M、G、T、P的定义

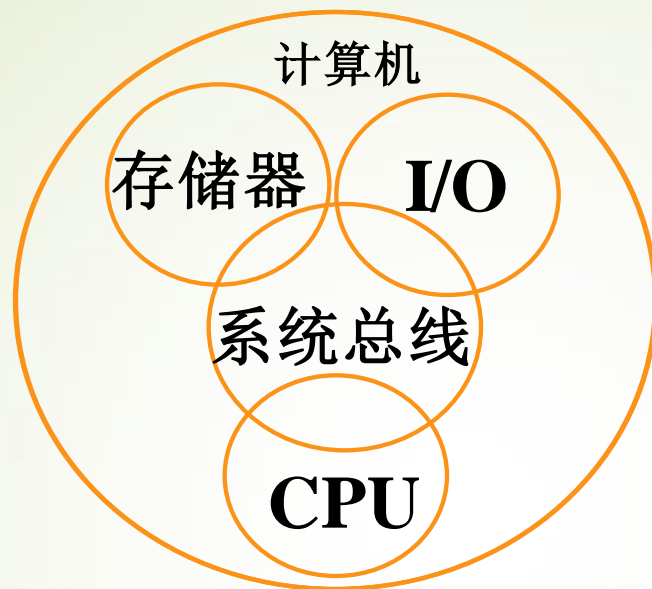
单 位	通常意义	实际表示
K(Kilo)	10^3	$2^{10}=1\ 024$
M(Mega)	10^6	$2^{20}=1\ 048\ 576$
G(Giga)	10^9	$2^{30}=1\ 073\ 741\ 824$
T(Tera)	10^{12}	$2^{40}=1\ 099\ 511\ 627\ 776$
P(Peta)	10^{15}	$2^{50}=1\ 125\ 899\ 906\ 842\ 624$

1.4 本书结构

计算机

第 1 篇 概论

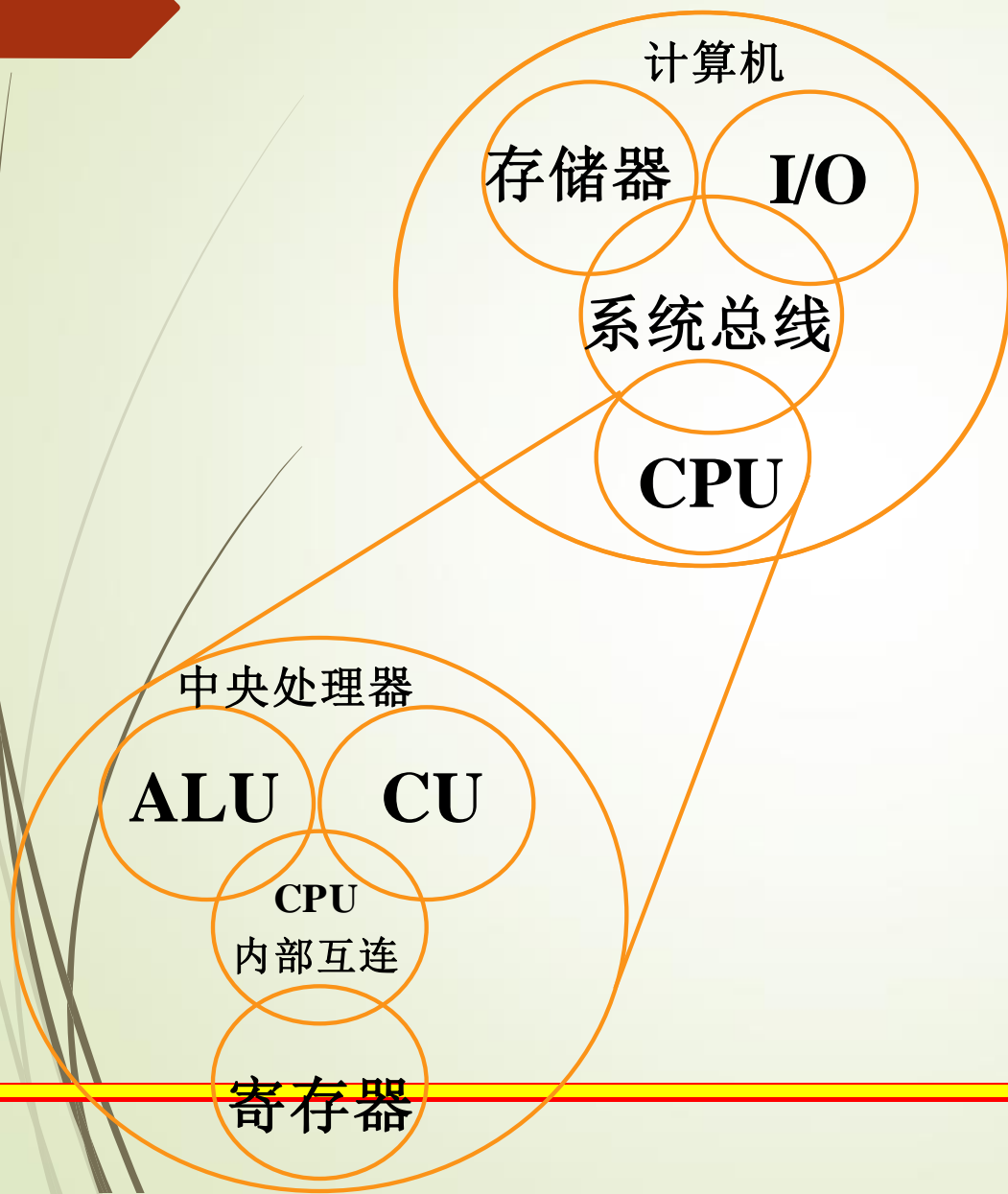
1.4 本书结构



第 2 篇 计算机系统的硬件结构

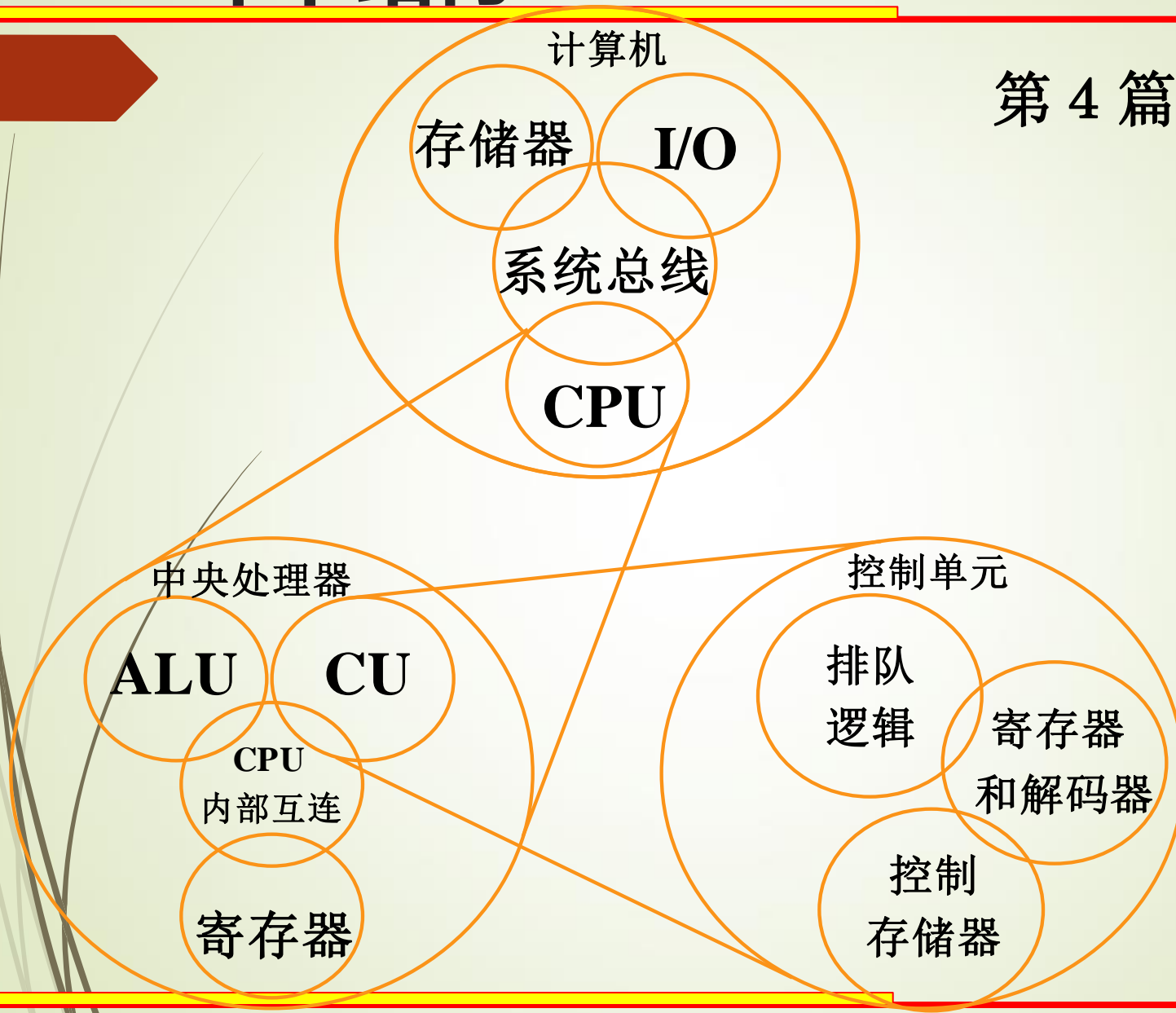
1.4 本书结构

第 3 篇 CPU



1.4 本书结构

第 4 篇 CU



Thank You !

