

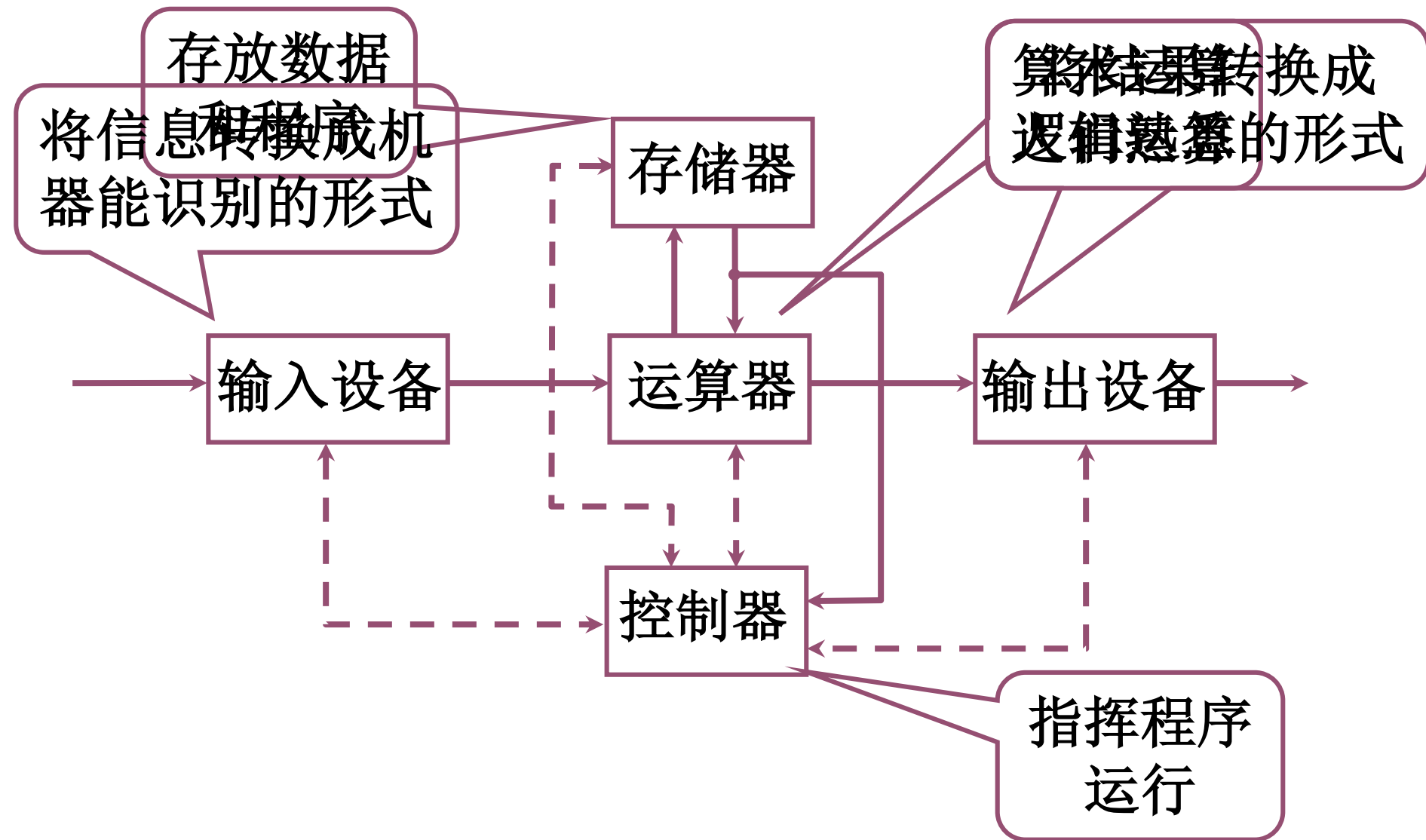
1.2 计算机的基本组成

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

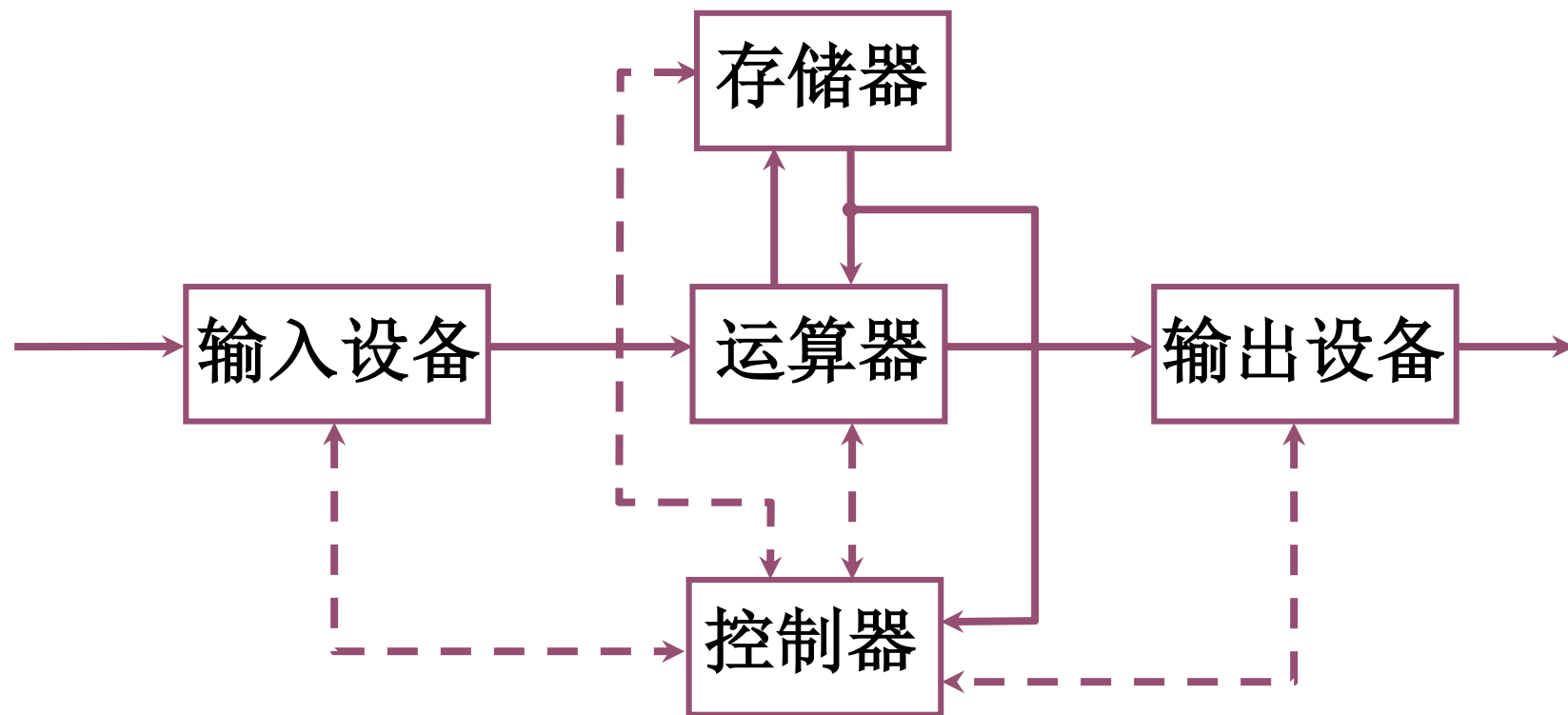
冯·诺依曼计算机硬件框图

1.2



冯·诺依曼计算机硬件框图

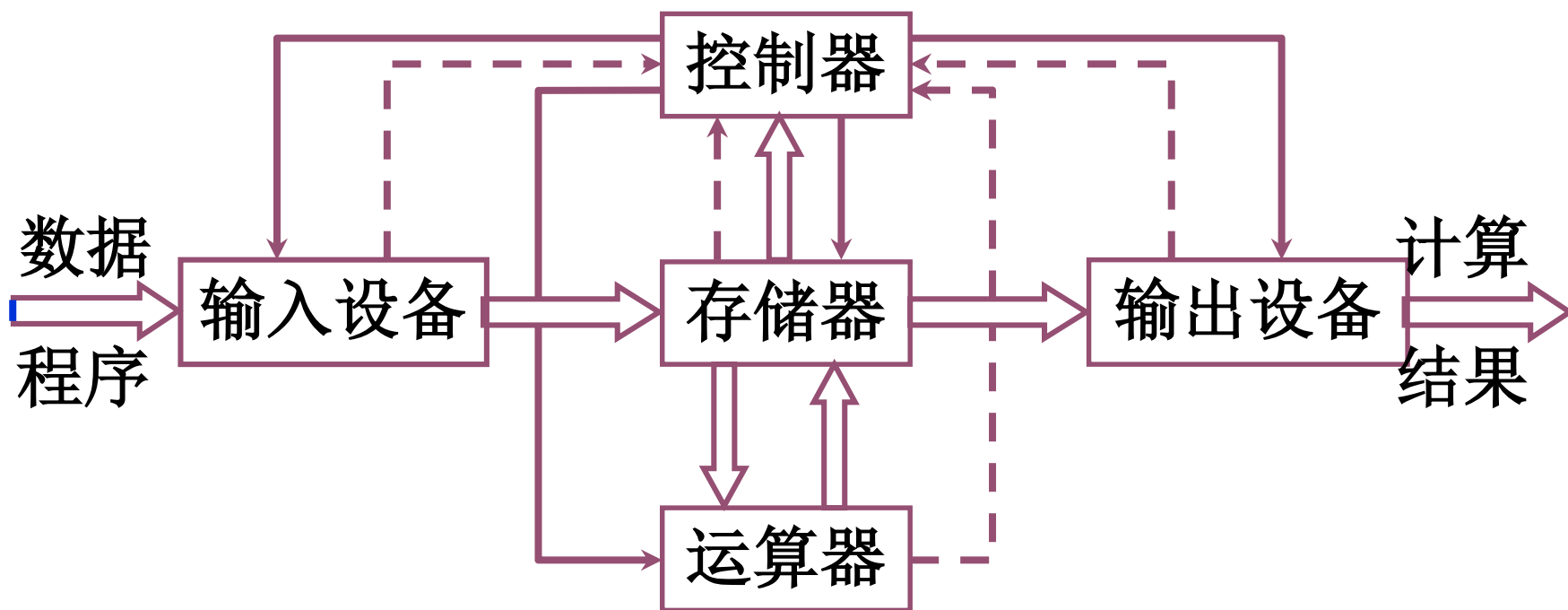
1.2



二、计算机硬件框图

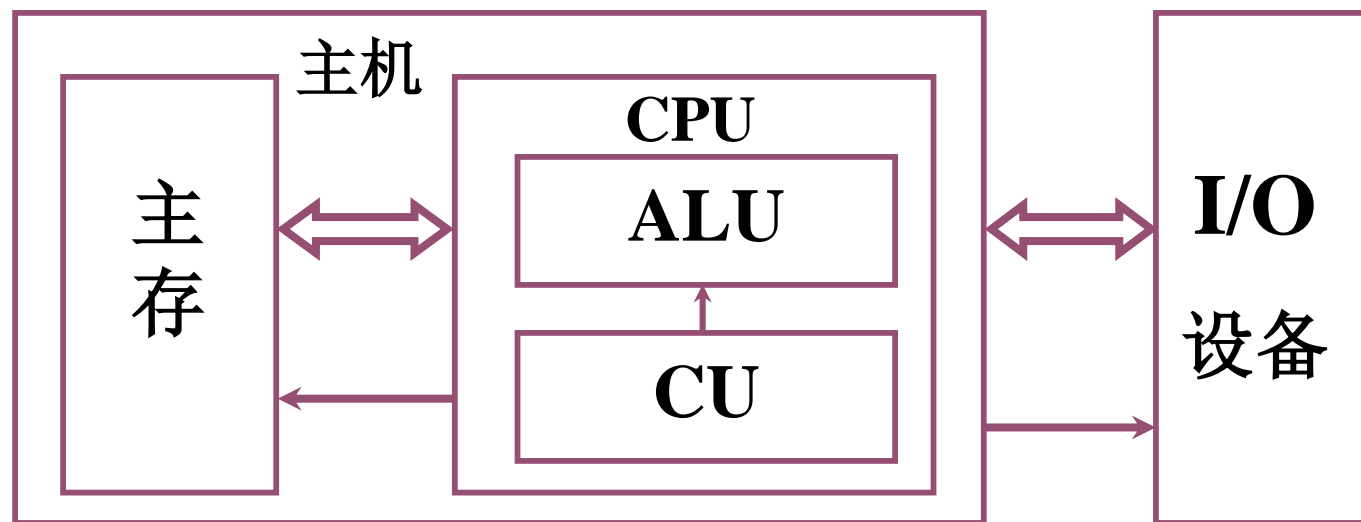
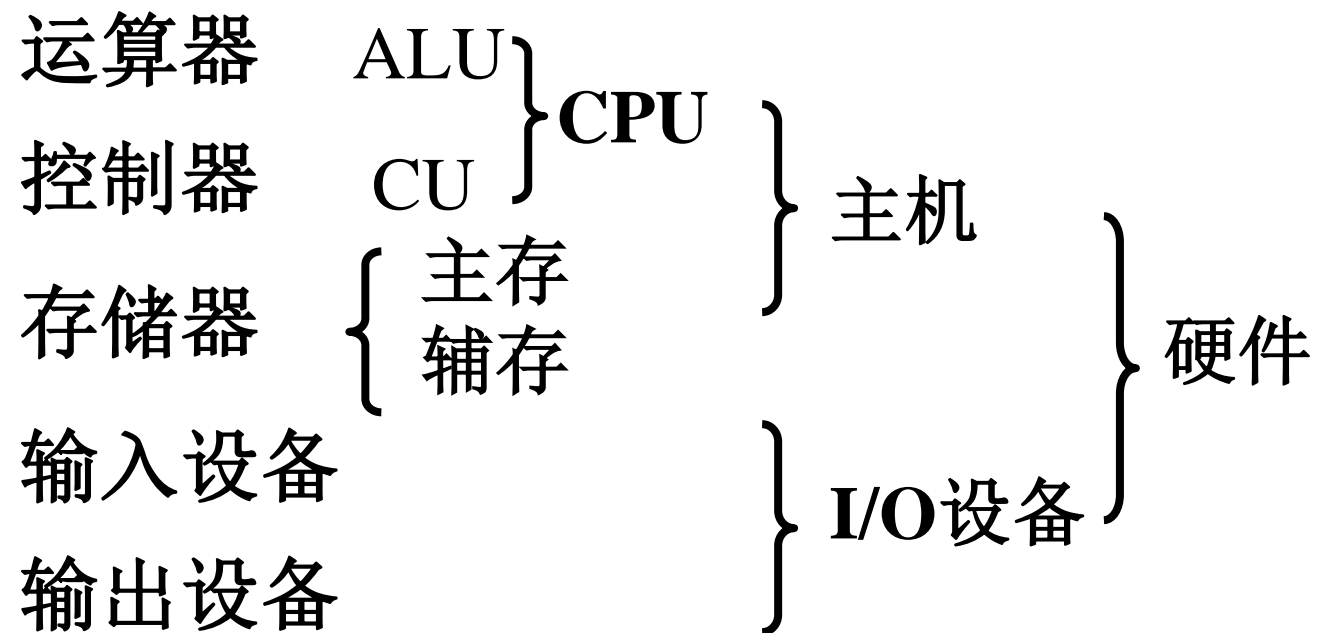
1.2

1. 以存储器为中心的计算机硬件框图



2.现代计算机硬件框图

1.2



2.现代计算机硬件框图

- 系统复杂性管理的方法-2（3'Y）
 - 层次化（**Hierachy**）：将被设计的系统划分为多个模块或子模块
 - 模块化（**Modularity**）：有明确定义（**well-defined**）的功能和接口
 - 规则性（**regularity**）：模块更容易被重用

问题：一个现实中的问题，如何用计算机来解决？

问题：是不是所有的问题都可以用计算的方法来解决

假设我们面对的是一个可以有计算机解决的问题，
如何用计算机来解决这个问题呢？

三、计算机的工作步骤

1.2

1.上机前的准备

- 建立数学模型

$$u = U_m \sin \omega t$$

- 确定计算方法

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

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

- 编制解题程序

程序 —— 运算的 全部步骤

指令 —— 每 一个步骤

编程举例

1.2

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

指令格式举例

操作码	地址码
-----	-----

取数 α $[\alpha] \rightarrow ACC$

000001 0000001000

存数 β $[ACC] \rightarrow \beta$ 加 γ $[ACC] + [\gamma] \rightarrow ACC$ 乘 δ $[ACC] \times [\delta] \rightarrow ACC$ 打印 σ $[\sigma] \rightarrow \text{打印机}$

停机

计算 $ax^2 + bx + c$ 程序清单

1.2

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

- 指令和数据都是保存在存储器中的

存储器的结构？

如何进行访问？

每次访问获得的数据的位数是多少呢？

2.计算机的解题过程

1.2

(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

按地址寻访

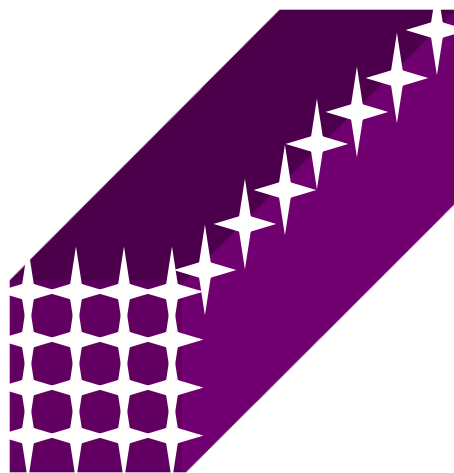
(1)存储器的基本组成

1.2



MAR 存储器地址寄存器
反映存储单元的个数

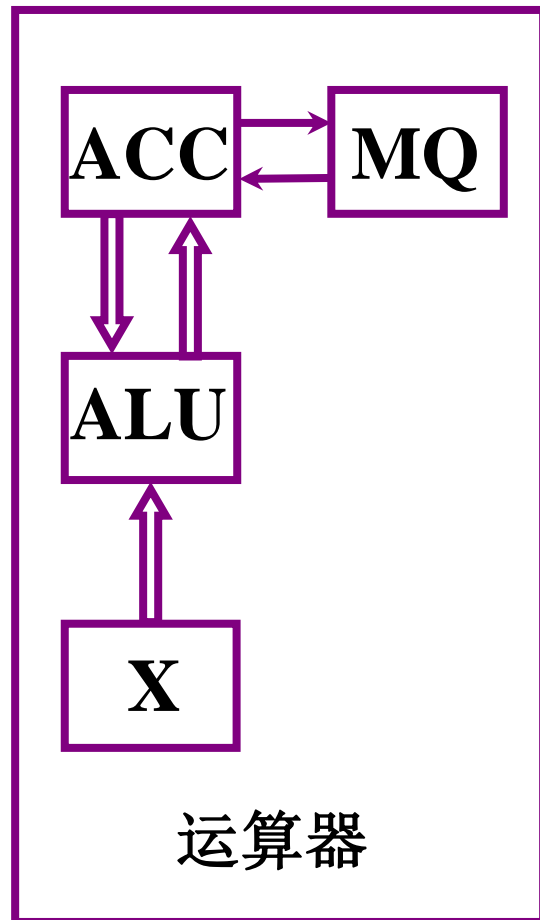
MDR 存储器数据寄存器
反映存储字长



设 **MAR=4 位**
MDR=8 位
存储单元个数 **16**
存储字长 **8**

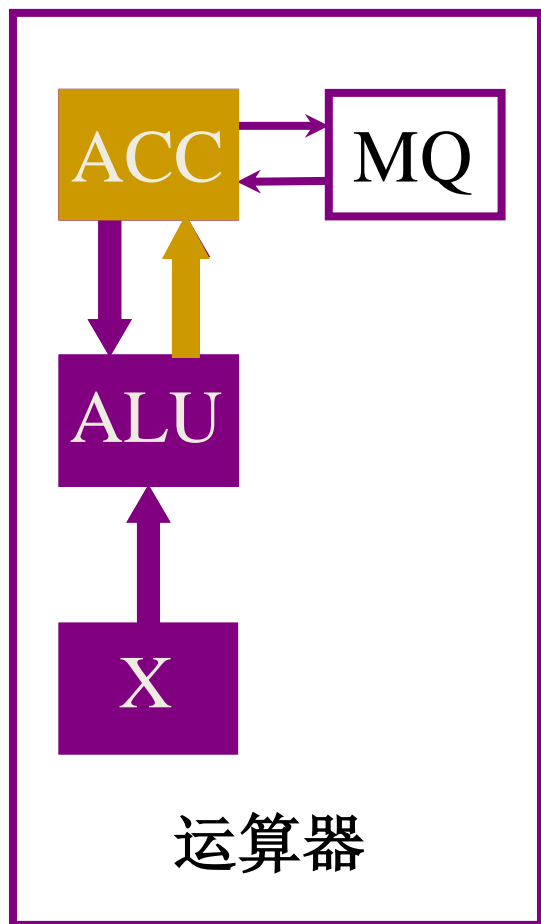
- 已经知道了存储器的基本组成
- 运算器的结构是什么？
- 运算器功能是什么？，如何工作的？
 - 加法？
 - 乘法？

(2)运算器的基本组成及操作过程



	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



指令

加

M

初态

ACC

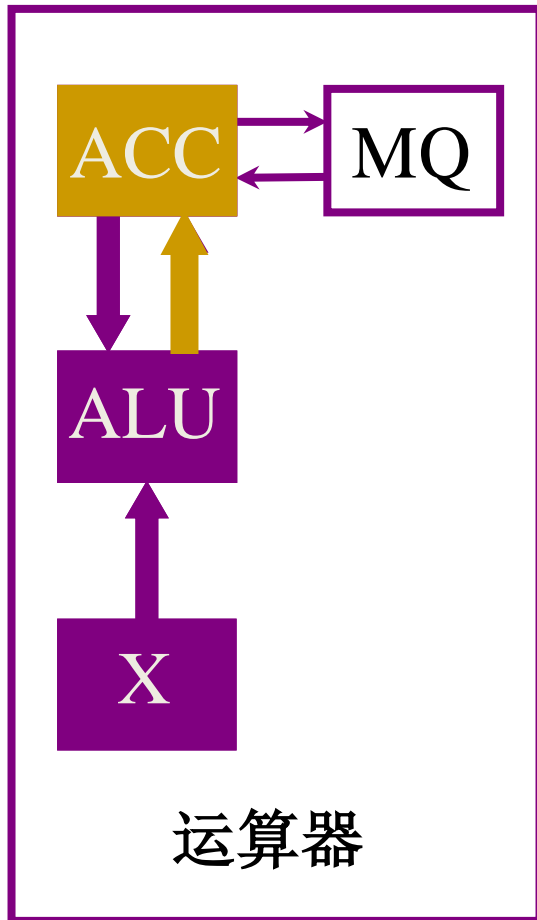
被加数

$[M] \longrightarrow X$

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

② 减法操作过程

1.2



指令

减

M

初态

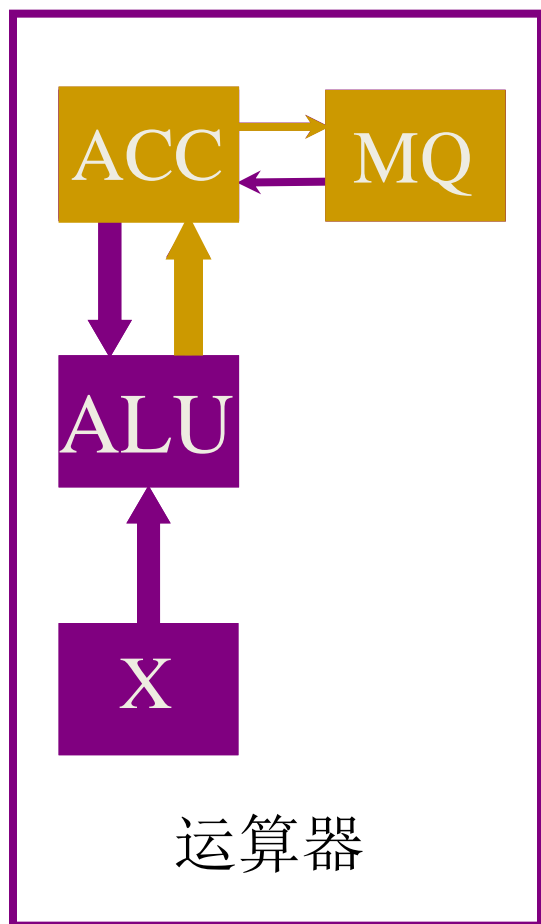
ACC

被减数

$[M] \rightarrow X$

$[ACC] - [X] \rightarrow ACC$

③ 乘法操作过程



指令

乘

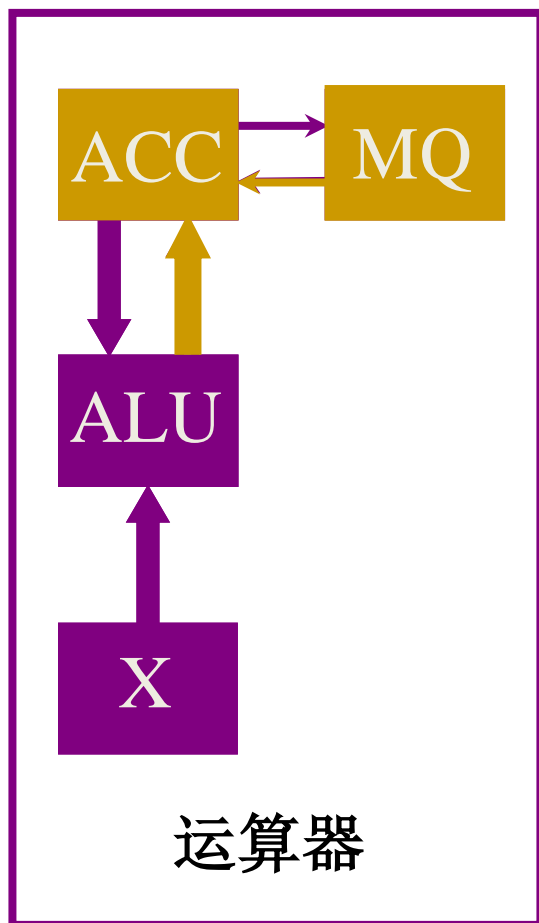
M

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



指令

除

M

初态

ACC 被除数

 $[M] \longrightarrow X$ $[ACC] \div [X] \longrightarrow MQ$

余数在ACC中

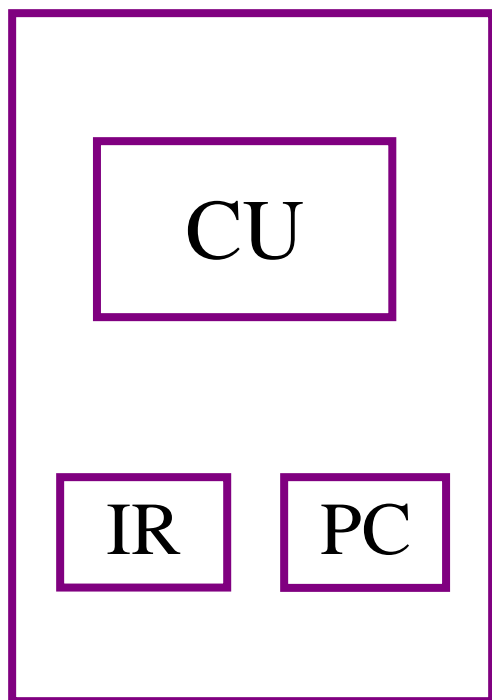
控制器的基本结构如何呢？

控制器的功能？

- ✓ 解释指令
- ✓ 保证指令的按序执行

(3)控制器的基本组成

1.2



完成 一条 指令	{	取指令	PC
		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 $(PC) + 1 \rightarrow PC$

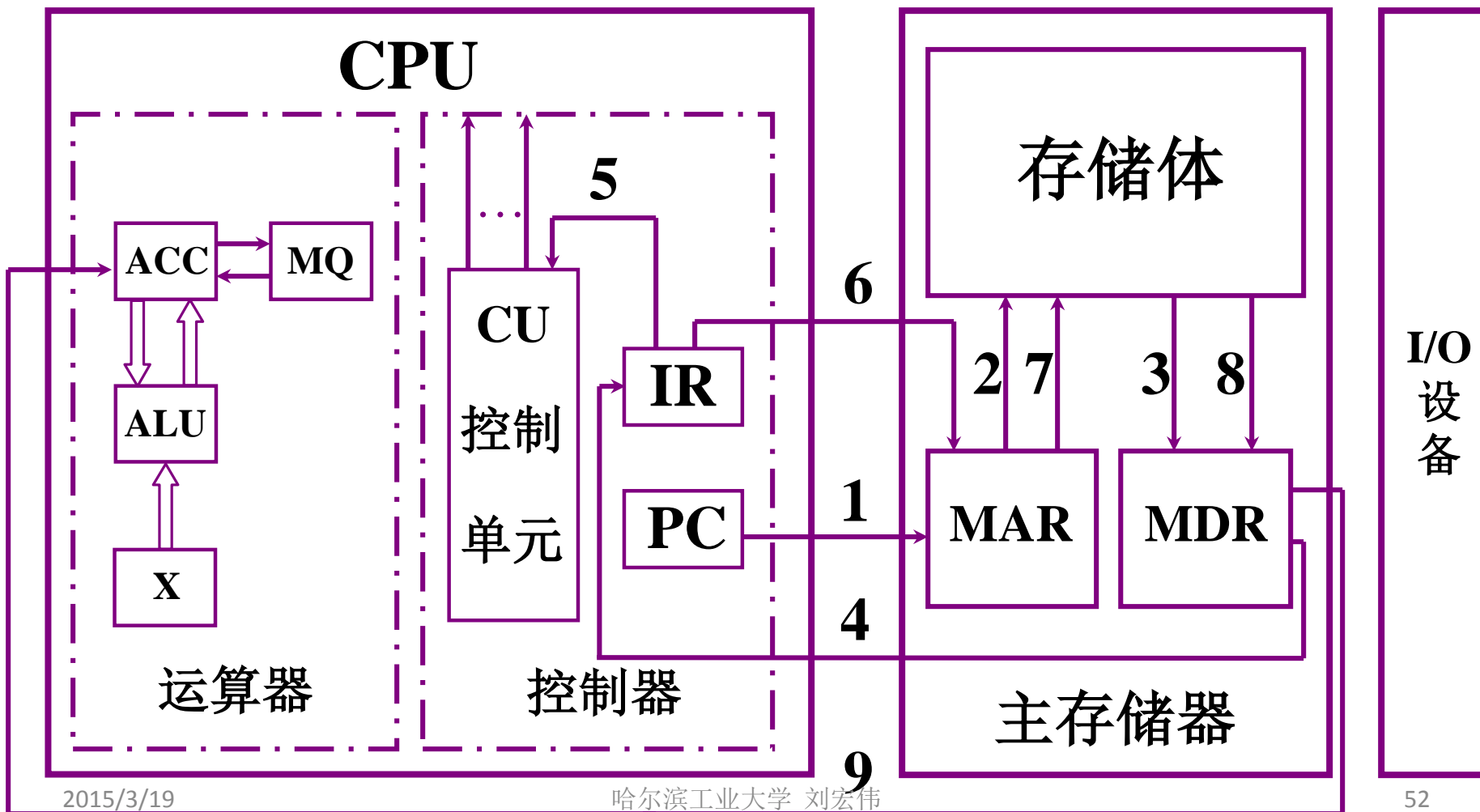
IR 存放当前欲执行的指令

- 运算器、控制器、存储器构成了什么
 - ✓一条指令在主机上的完成过程
 - ✓程序在主机上是如何执行的

(4)主机完成一条指令的过程

以取数指令为例

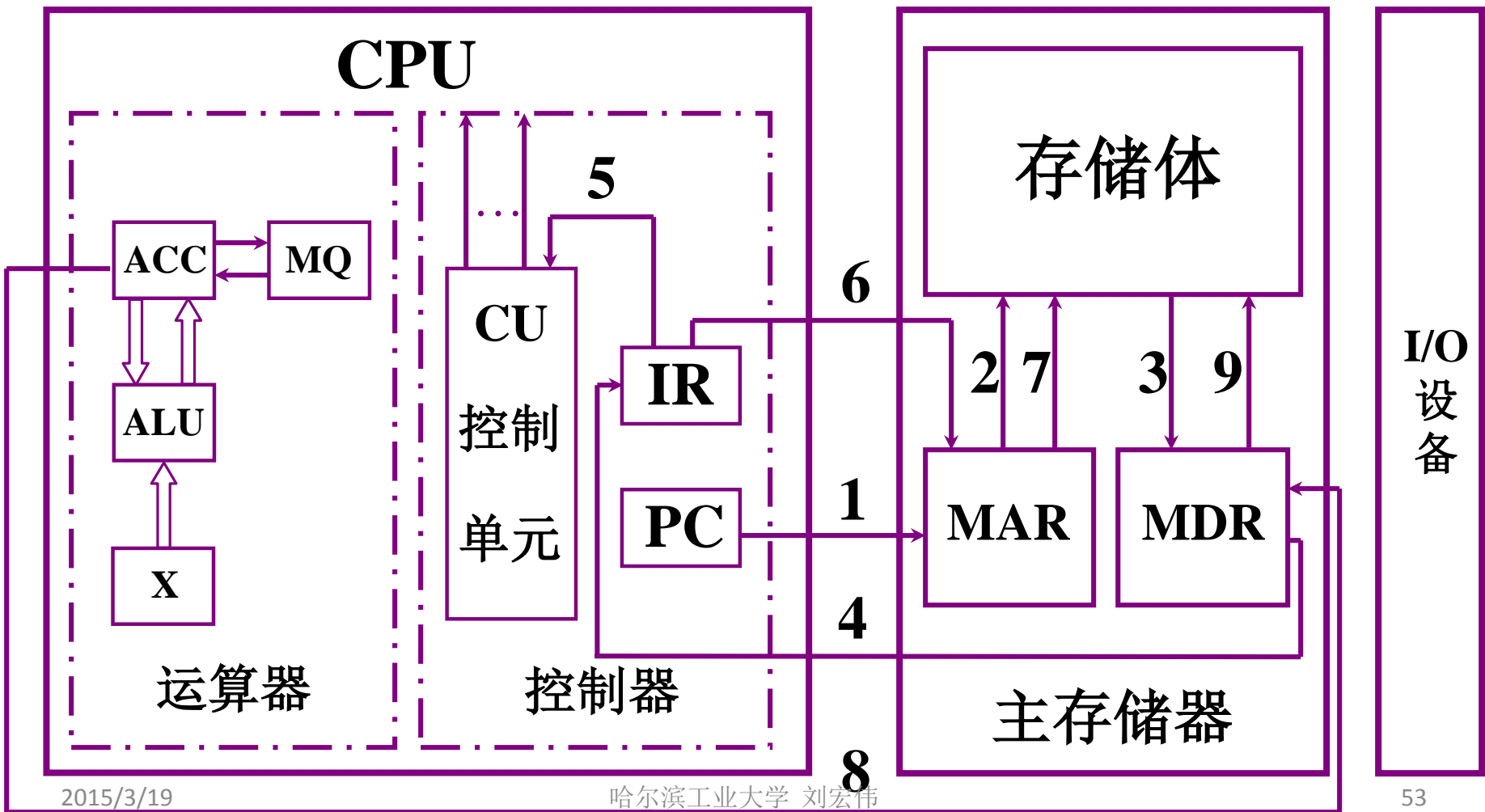
1.2



(4)主机完成一条指令的过程

1.2

以存数指令为例



(5) $ax^2 + bx + c$ 程序的运行过程

1.2

- 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 $PC \longrightarrow MAR \longrightarrow M \longrightarrow MDR \longrightarrow IR$, $(PC) + 1 \longrightarrow PC$
- 分析指令 $OP(IR) \longrightarrow CU$
- 执行指令 $Ad(IR) \longrightarrow MAR \longrightarrow M \longrightarrow MDR \longrightarrow ACC$
- \vdots
- 打印结果
- 停机

- 如果打算买一台机器，如何进行合适的选择呢？

- ✓ 买这台机器做什么

- ✓ 你有多少钱

- ✓ 机器的性能能否满足你的要求

- 如何在购买前对计算机的性能进行评价

- 处理速度快、内存容量大