



中山大學 软件工程学院
SUN YAT-SEN UNIVERSITY SCHOOL OF SOFTWARE ENGINEERING

计算机组成原理

授课老师：吴炜滨

- 计算机的基本组成
 - 冯·诺依曼计算机的特点
 - 计算机的硬件框图
 - 计算机的工作步骤

- 计算机硬件的主要技术指标

- 计算机的基本组成
 - 冯·诺依曼计算机的特点

冯·诺依曼计算机的特点



■ 计算机由五大部件组成

- 运算器、控制器、存储器、输入设备、输出设备

■ 指令和数据以同等地位存于存储器，并按地址寻访

■ 指令和数据用二进制表示

■ 指令由操作码和地址码组成

- 操作码：操作的性质；地址码：操作数所在地址

■ 存储程序

- 核心特征，具有存储程序特征的计算机统称为**冯·诺依曼计算机**

■ 以运算器为中心

- 输入输出设备与存储器间的数据传送是通过运算器完成的



➤ 计算机的基本组成

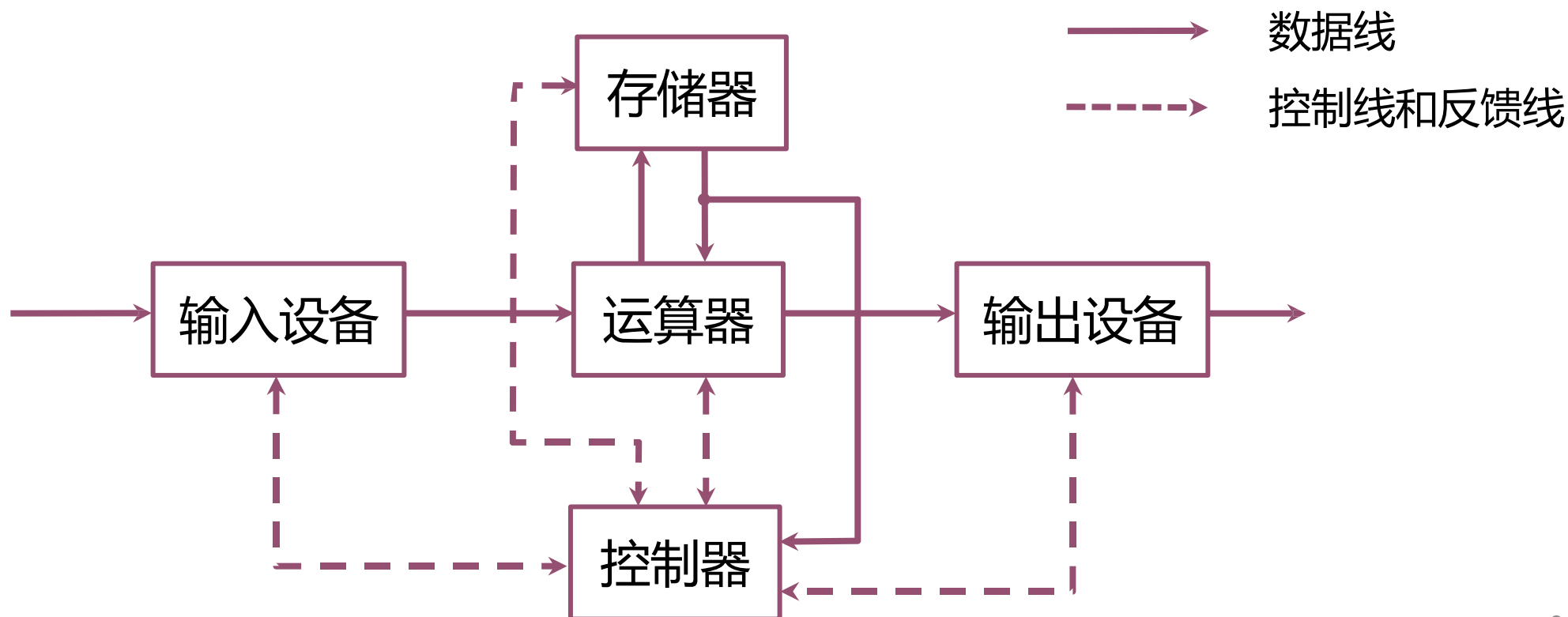
- 计算机的硬件框图

以运算器为中心的计算机硬件框图



■ 各部件功能

- 运算器：算术运算、逻辑运算
- 存储器：存放数据和程序

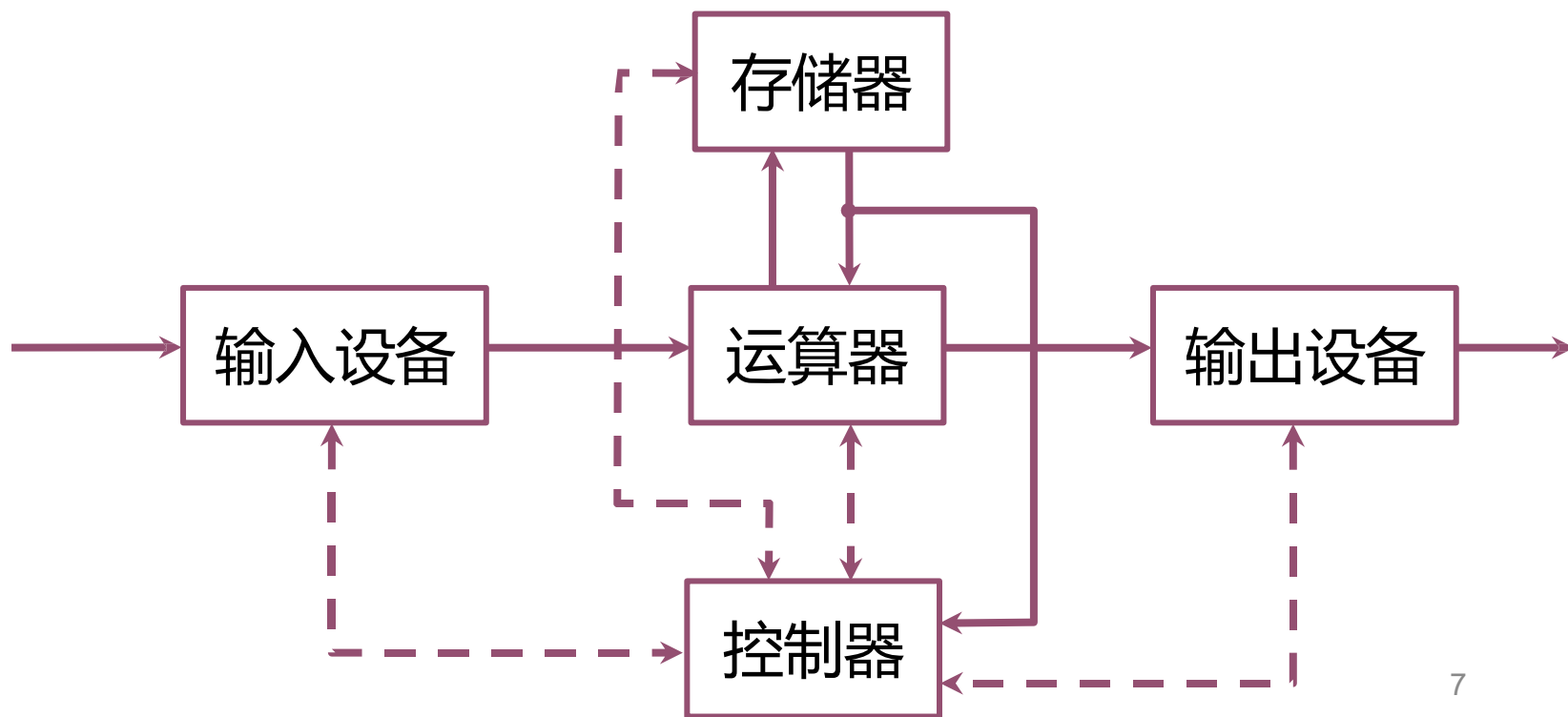


以运算器为中心的计算机硬件框图

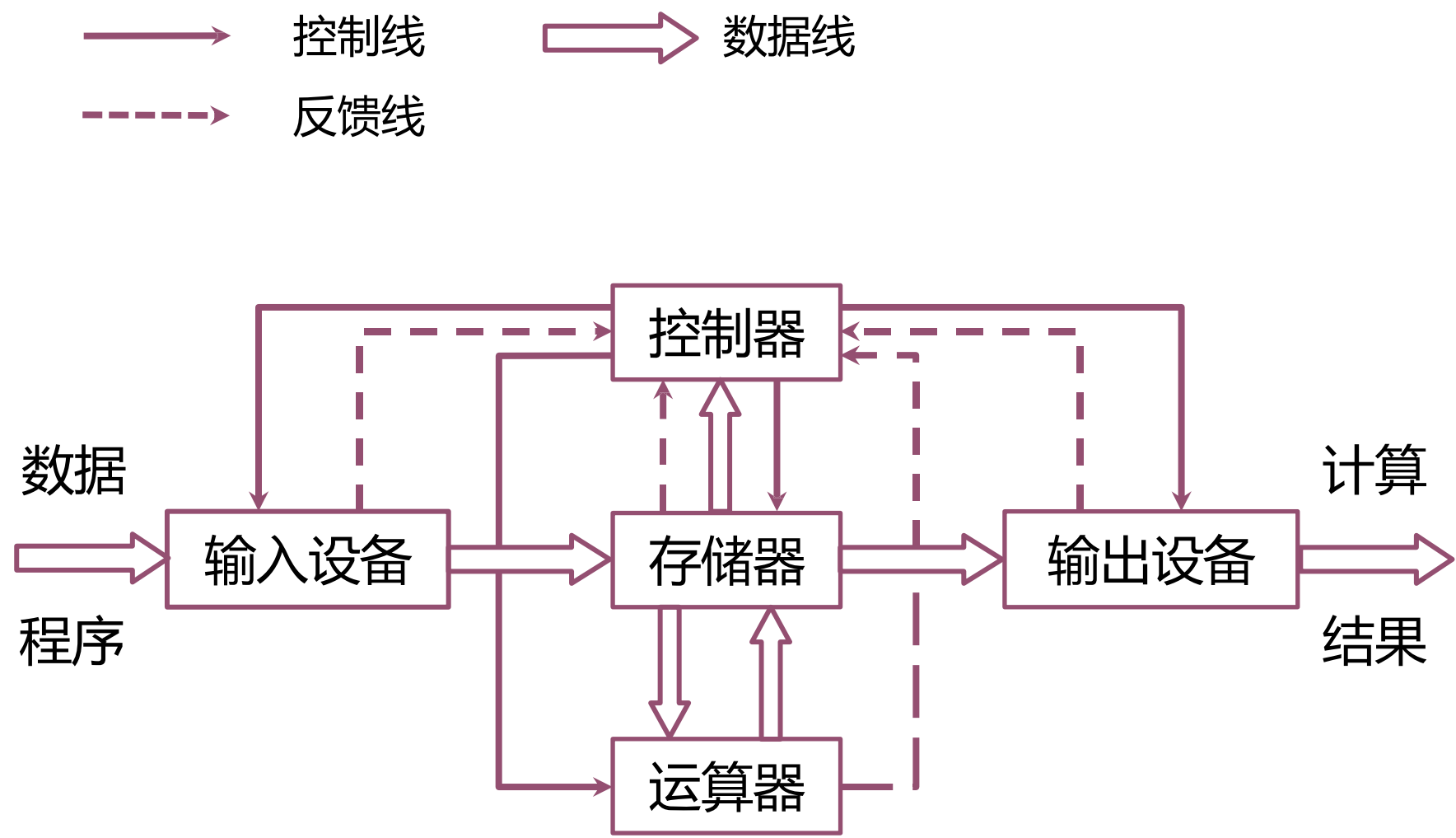


■ 各部件功能

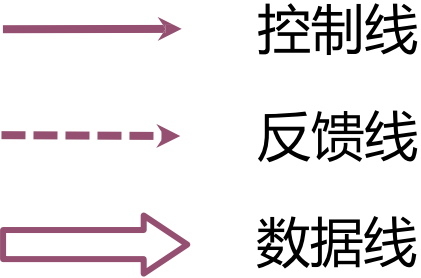
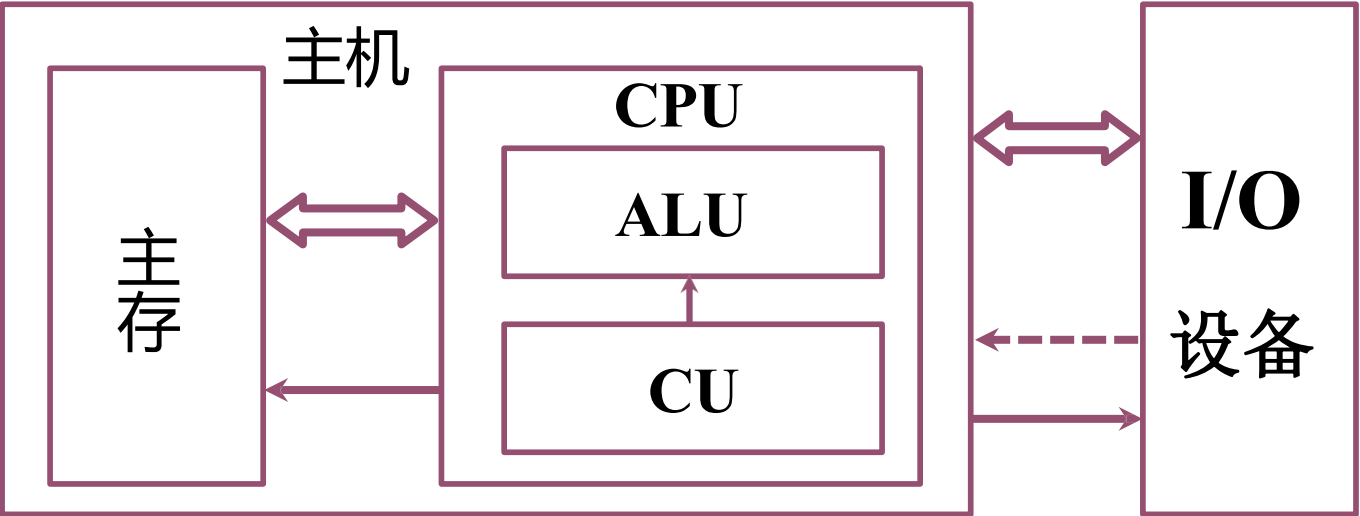
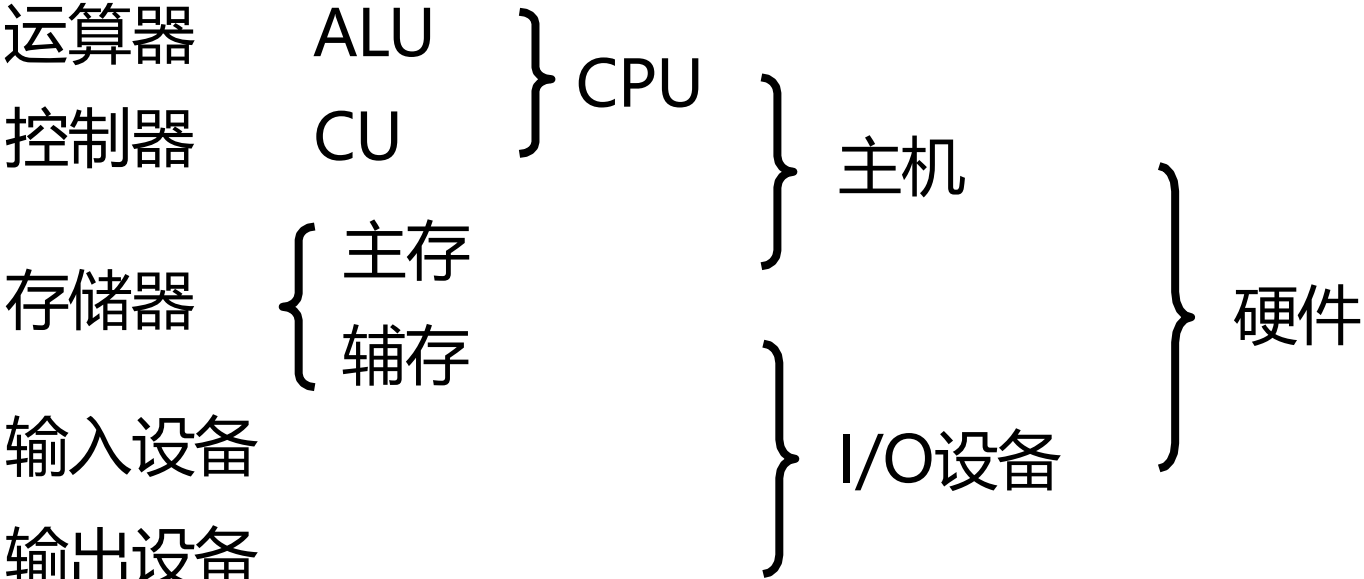
- 控制器：控制、指挥程序和数据的输入、运行，处理运算结果
- 输入设备：将人类熟悉的信息形式转换成机器能识别的信息形式
- 输出设备：将机器运算结果转换成人们熟悉的信息形式



以存储器为中心的计算机硬件框图



现代计算机硬件框图



系统复杂性管理的方法 (3' Y)



■ 层次化 (Hierarchy)

- 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



- 计算机的基本组成
 - 计算机的工作步骤

问题



- 一个现实中的问题，如何用计算机来解决？
 - 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？
 - 如何计算正弦函数的值？

上机前的准备



■ 建立数学模型

$\sin x$

■ 确定计算方法

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

■ 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：计算 $ax^2 + bx + c$



■ 假设

- 累加器型的机器
 - 一个操作数要把它保存在累加器当中，另外的一个操作数来自于计算机的主存储器
- 包含了六条指令
 - 加法指令
 - 乘法指令
 - 取数指令
 - 存数指令
 - 打印指令
 - 停机指令

■ 计算 $ax^2 + bx + c$

- 假设 a, b, c, x 这些操作数已经保存到了存储器当中
- 方法一
 - 取 x 至运算器中的累加器中（取数指令）
 - 乘以 x ，结果 x^2 保存在运算器中的累加器中（乘法指令）
 - 乘以 a ，结果 ax^2 保存在运算器中的累加器中（乘法指令）
 - 存 ax^2 至存储器中（存数指令）
 - 取 b 至运算器中的累加器中（取数指令）
 - 乘以 x ，结果 bx 保存在运算器中的累加器中（乘法指令）
 - 加 ax^2 ，结果 $ax^2 + bx$ 保存在运算器中的累加器中（加法指令）
 - 加 c ，结果 $ax^2 + bx + c$ 保存在运算器中的累加器中（加法指令）

■ 计算 $ax^2 + bx + c$

- 假设 a, b, c, x 这些操作数已经保存到了存储器当中
- 方法二: $ax^2 + bx + c = (ax + b)x + c$
 - 取 x 至运算器中的累加器中 (取数指令)
 - 乘以 a , 结果 ax 保存在运算器中的累加器中 (乘法指令)
 - 加 b , 结果 $ax + b$ 保存在运算器中的累加器中 (加法指令)
 - 乘以 x , 结果 $(ax + b)x$ 保存在运算器中的累加器中 (乘法指令)
 - 加 c , 结果 $(ax + b)x + c$ 保存在运算器中的累加器中 (加法指令)
- 指令的条数较少, 执行速度较快, 所需存储空间较小

指令格式举例



■ 取数 a

- 000001 0000001000
- $[a] \rightarrow \text{ACC}$ (累加器)

■ 存数 b

- $[\text{ACC}] \rightarrow b$

■ 加 c

- $[\text{ACC}] + [c] \rightarrow \text{ACC}$

■ 乘 d

- $[\text{ACC}] \times [d] \rightarrow \text{ACC}$ // MQ

操作码

地址码

■ 打印 e

- $[e] \rightarrow \text{打印机}$

■ 停机

计算 $ax^2 + bx + c$ 程序清单

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

- 指令和数据以同等的地位保存在存储器当中
- 指令和数据都是二进制

存储器的基本组成



■ 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?

存储器的基本组成



存储器核心结构



存储体 – 存储单元 – 存储元件 (0/1)

大楼 – 房间 – 床位 (无人/有人)

存储体

MAR

MDR

主存储器

存储单元： 存放一串二进制代码 (指令/数据)

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

按地址寻访： 每个存储单元赋予一个地址号

存储器的基本组成



■ MAR

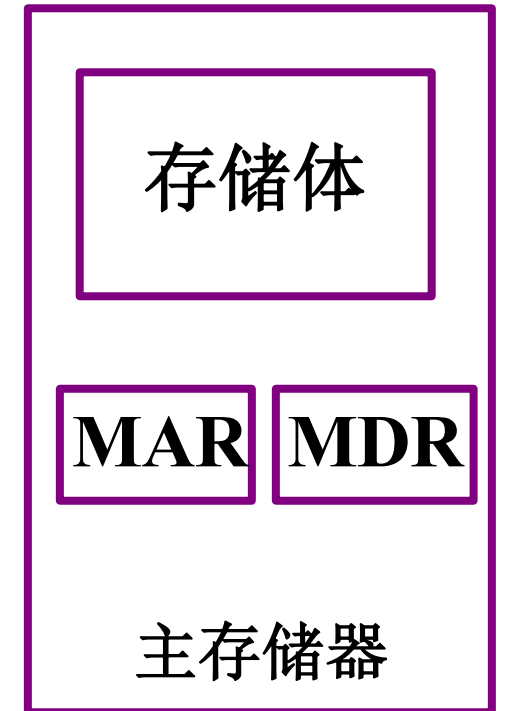
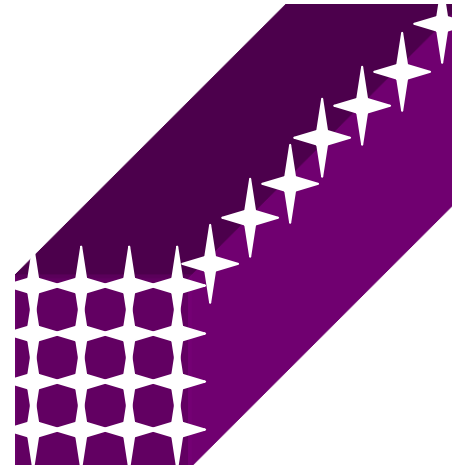
- 存储器地址寄存器
- 保存欲访问的存储单元的地址，位数反映存储单元的个数

■ MDR

- 存储器数据寄存器
- 保存来自或要送往存储体的数据，位数反映存储字长

■ 设MAR= 4 位，MDR= 8 位

- 存储单元个数16
- 存储字长8



运算器的基本组成及操作过程



- 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？

运算器的基本组成及操作过程

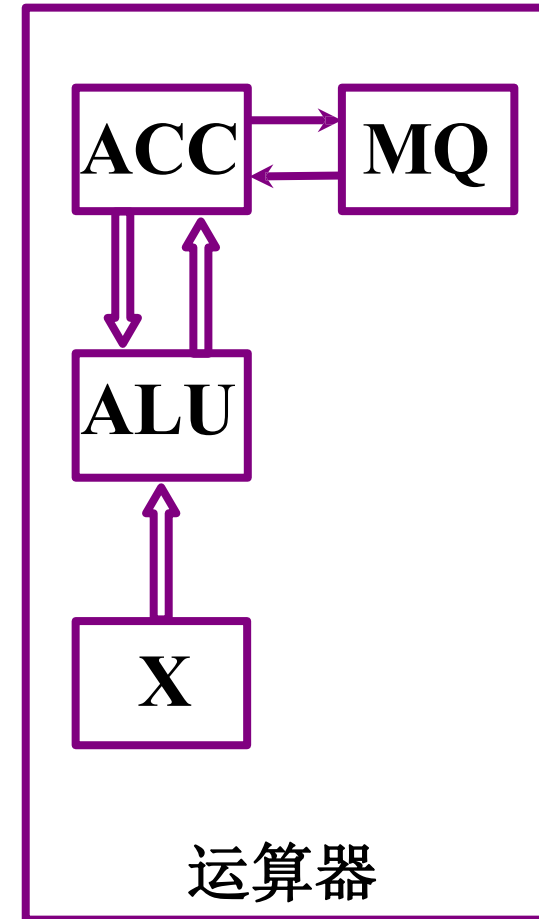


■ 运算器的最基本的功能，就是完成运算

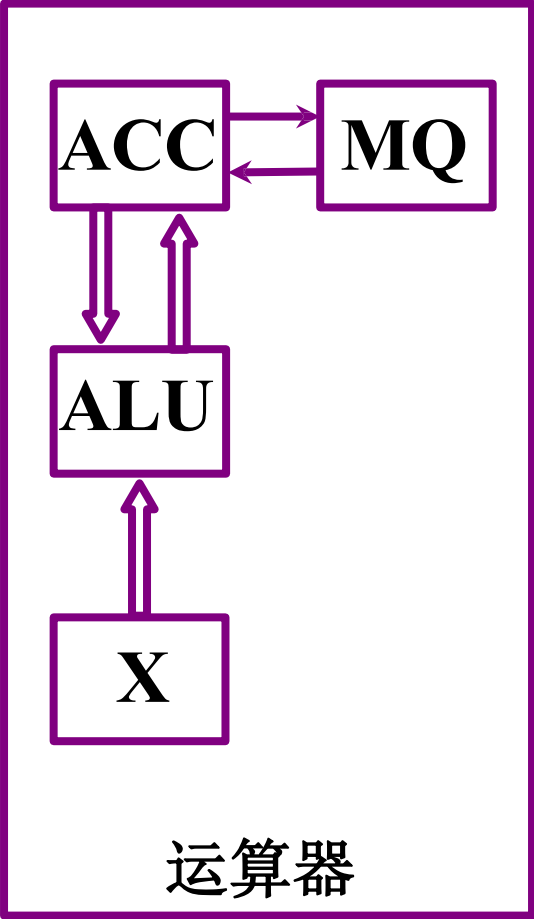
- 核心就是算逻运算单元（ALU）
- ALU通常是组合逻辑电路

■ 对输入、输出进行保存

- ACC：累加器
- X：操作数寄存器
- 乘法结果的长度是操作数的两倍
 - MQ寄存器：保存增加出来的部分



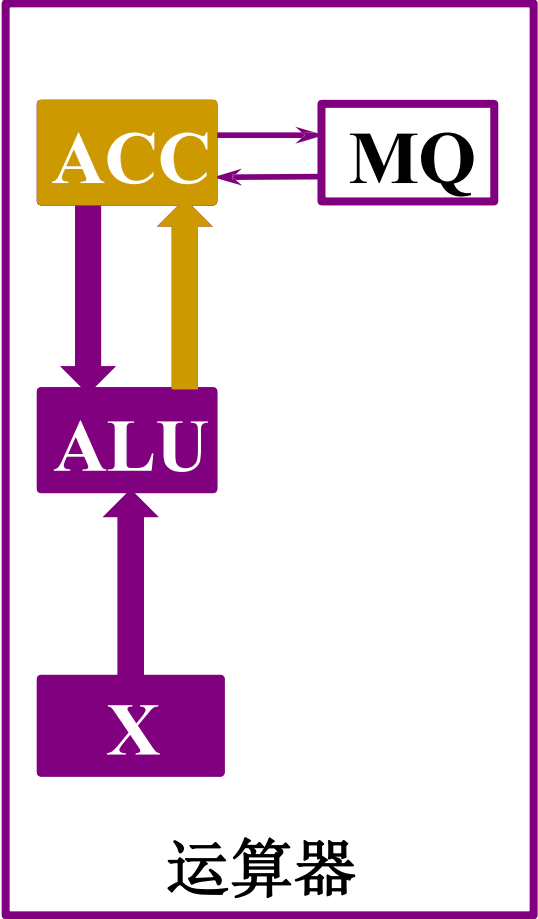
运算器的基本组成及操作过程



	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

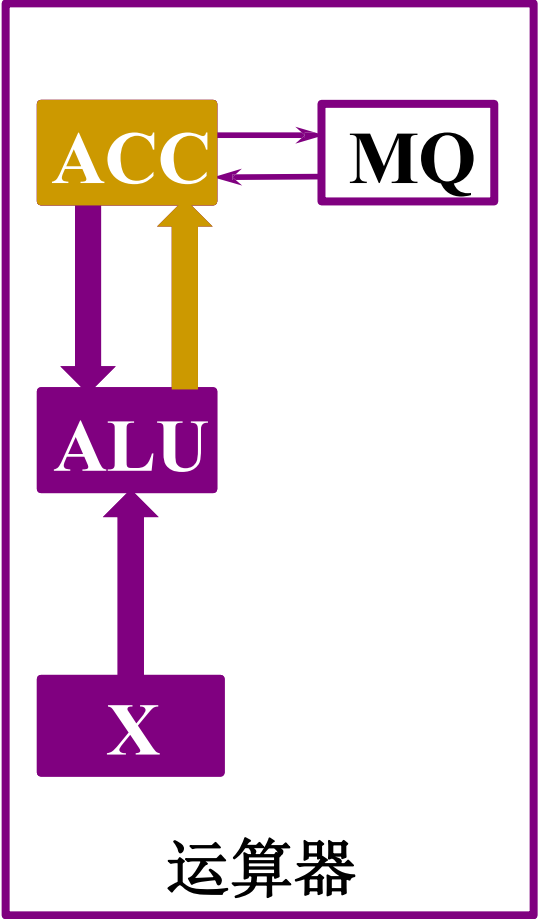
- 类似笔算乘法
- 中间结果累加：累加器
- 通过减法实现

加法操作过程



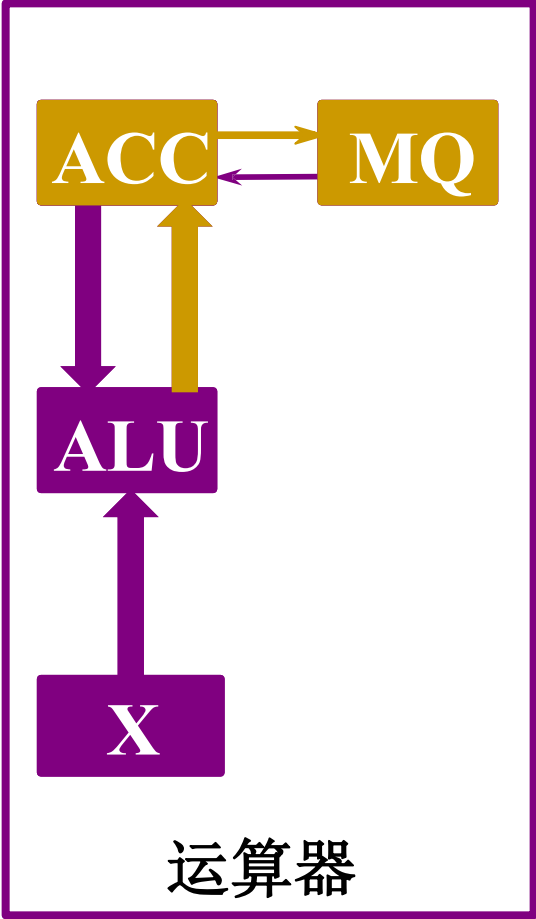
指令	加	M
初态	ACC	被加数
	$[M] \rightarrow X$	
	$[ACC] + [X] \rightarrow ACC$	

减法操作过程



指令	减	M
初态	ACC	被减数
	$[M] \rightarrow X$	
	$[ACC] - [X] \rightarrow ACC$	

乘法操作过程



指令

乘	M
---	---

初态 ACC 被乘数

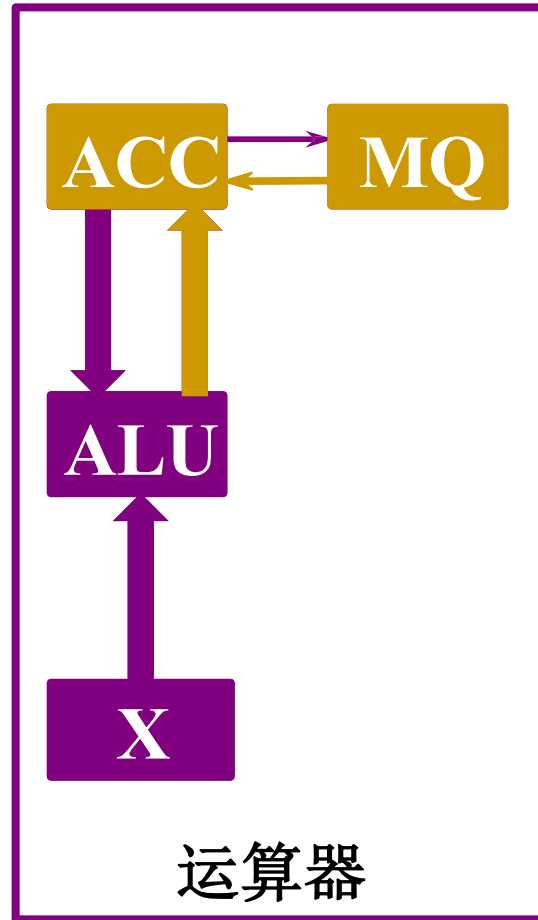
$[M] \rightarrow MQ$

$[ACC] \rightarrow X$

$0 \rightarrow ACC$

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

除法操作过程



指令

除	M
---	---

初态

ACC 被除数

$[M] \rightarrow X$

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

余数在ACC中

MQ: 乘商寄存器

控制器的基本组成



■ 控制器的基本结构如何呢？

- 控制器的功能？
 - 完成指令
 - 保证指令的按序执行

控制器的基本组成



■ 完成一条指令

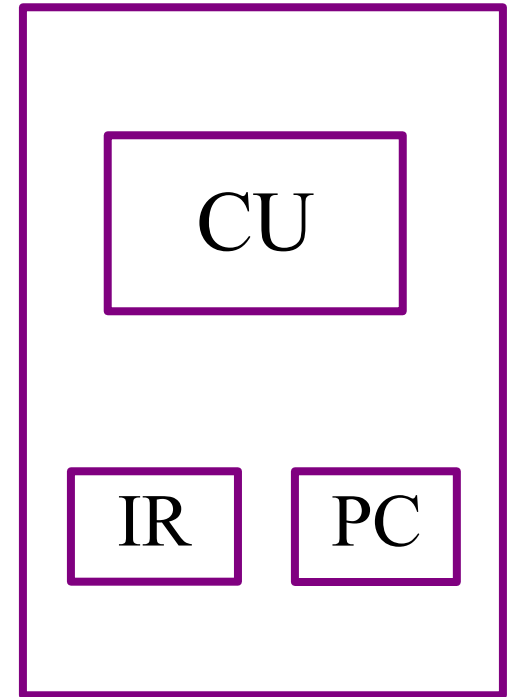
- 取指令：把内存单元当中保存的指令取出来，送入到控制器当中
- 分析指令：把指令的操作码部分送给控制单元进行分析
- 执行指令：由控制单元控制相应的执行部件去完成这条指令所需要的操作

控制器的基本组成



■ 完成一条指令

- 取指令
 - PC: 程序计数器
 - PC 存放当前欲执行指令的地址, 具有计数功能 $(PC) + 1 \rightarrow PC$
- 分析指令
 - IR: 指令寄存器
 - IR 存放当前欲执行的指令
- 执行指令
 - CU: 控制单元
 - CU 发出控制信号, 控制执行部件去按序执行指令要求的各种操作



主机

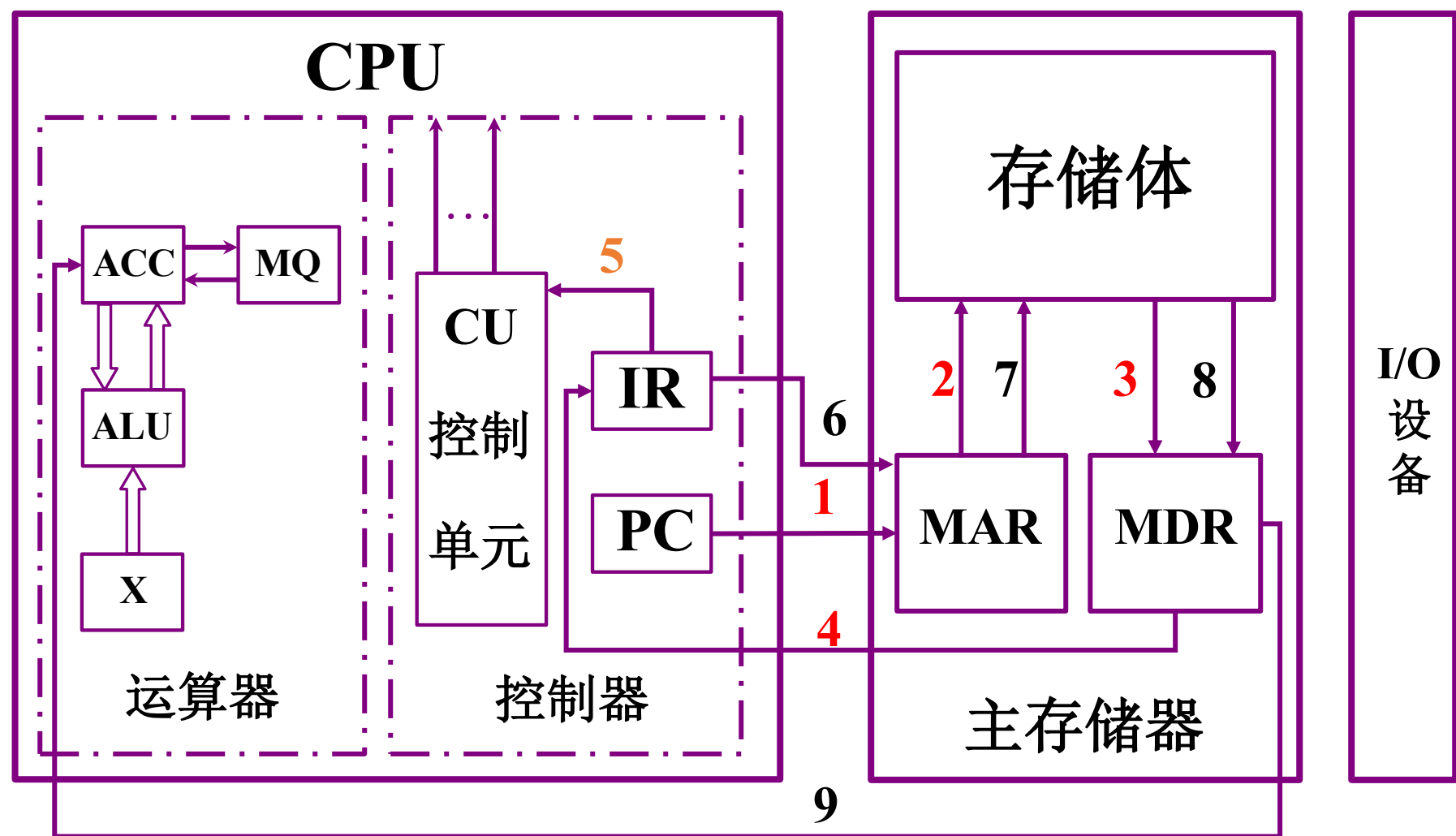


- 运算器、控制器、主存储器构成了什么
 - 主机
- 一条指令在主机上的完成过程
- 一个程序在主机上是如何执行的

完成一条指令的过程



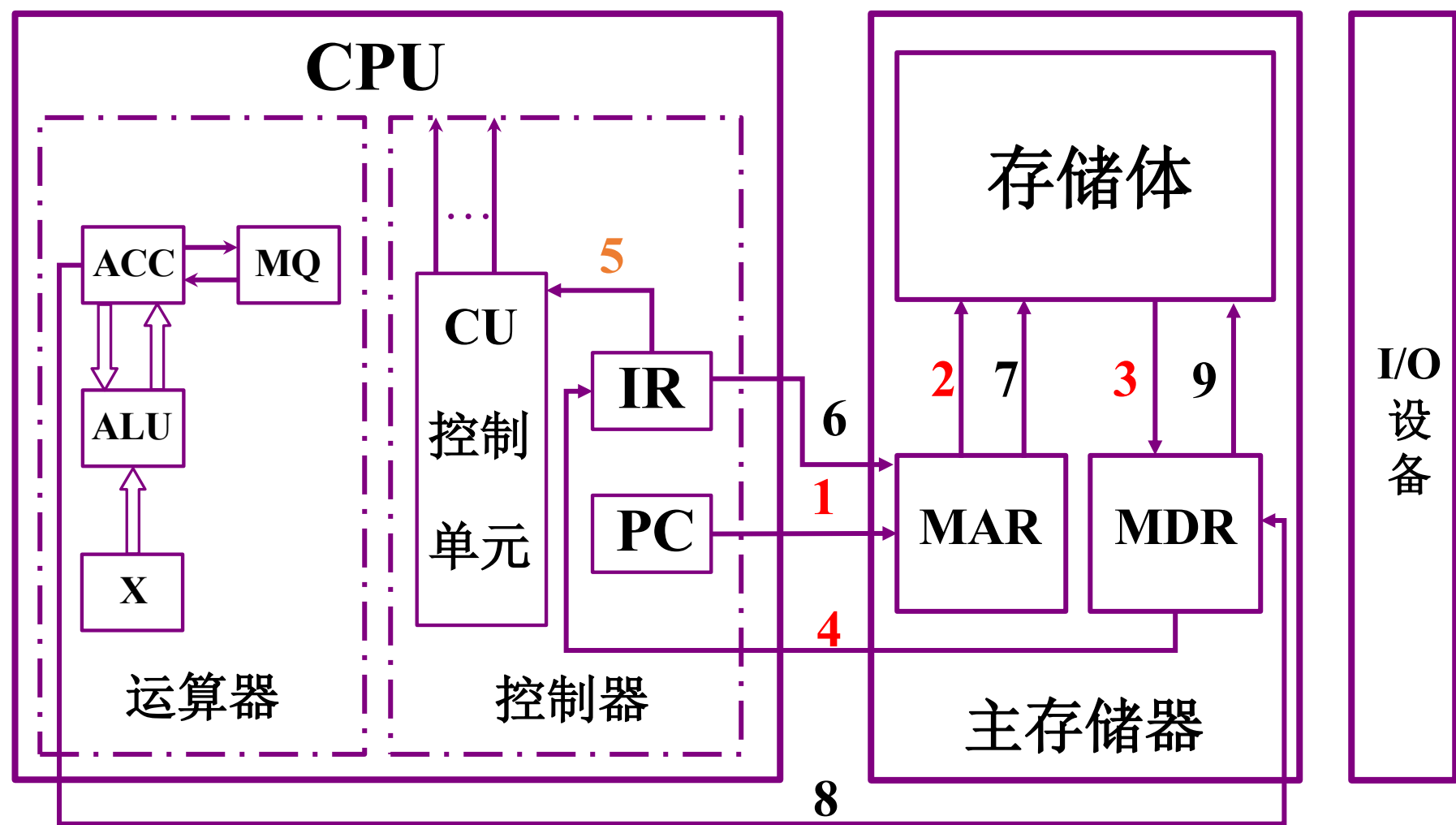
■ 以取数指令为例：取指令，分析指令，执行指令； $[M] \rightarrow ACC$



完成一条指令的过程



■ 以存数指令为例：取指令，分析指令，执行指令；[ACC] → [M]



完成一个程序的过程



■ $ax^2 + bx + c$ 程序的运行过程

- 将程序和数据通过输入设备送至计算机，保存到存储器中
- 程序首地址 \rightarrow PC
- 启动机器执行程序
- 取数指令： $[M] \rightarrow \text{ACC}$
 - 取指令： $\text{PC} \rightarrow \text{MAR} \rightarrow \text{M (存储体)} \rightarrow \text{MDR} \rightarrow \text{IR}$, $(\text{PC})+1 \rightarrow \text{PC}$
 - 分析指令： $\text{OP}(\text{IR}) \rightarrow \text{CU}$
 - 执行指令： $\text{Ad}(\text{IR}) \rightarrow \text{MAR} \rightarrow \text{M (存储体)} \rightarrow \text{MDR} \rightarrow \text{ACC}$
- ...
- 打印结果
- 停机



➤ 计算机硬件的主要技术指标

计算机硬件的主要技术指标



■ 机器字长

- CPU一次能处理数据的位数，通常与CPU中的寄存器位数相等

■ 运算速度

- T_M ：指令执行的速度
 - 吉普森法
 - a_i ：第*i*种指令占全部操作的百分比数
 - t_i ：第*i*种指令的执行时间
 - $T_M = \sum_{i=1}^n a_i t_i$

计算机硬件的主要技术指标



■ 机器字长

- CPU一次能处理数据的位数，通常与CPU中的寄存器位数相等

■ 运算速度

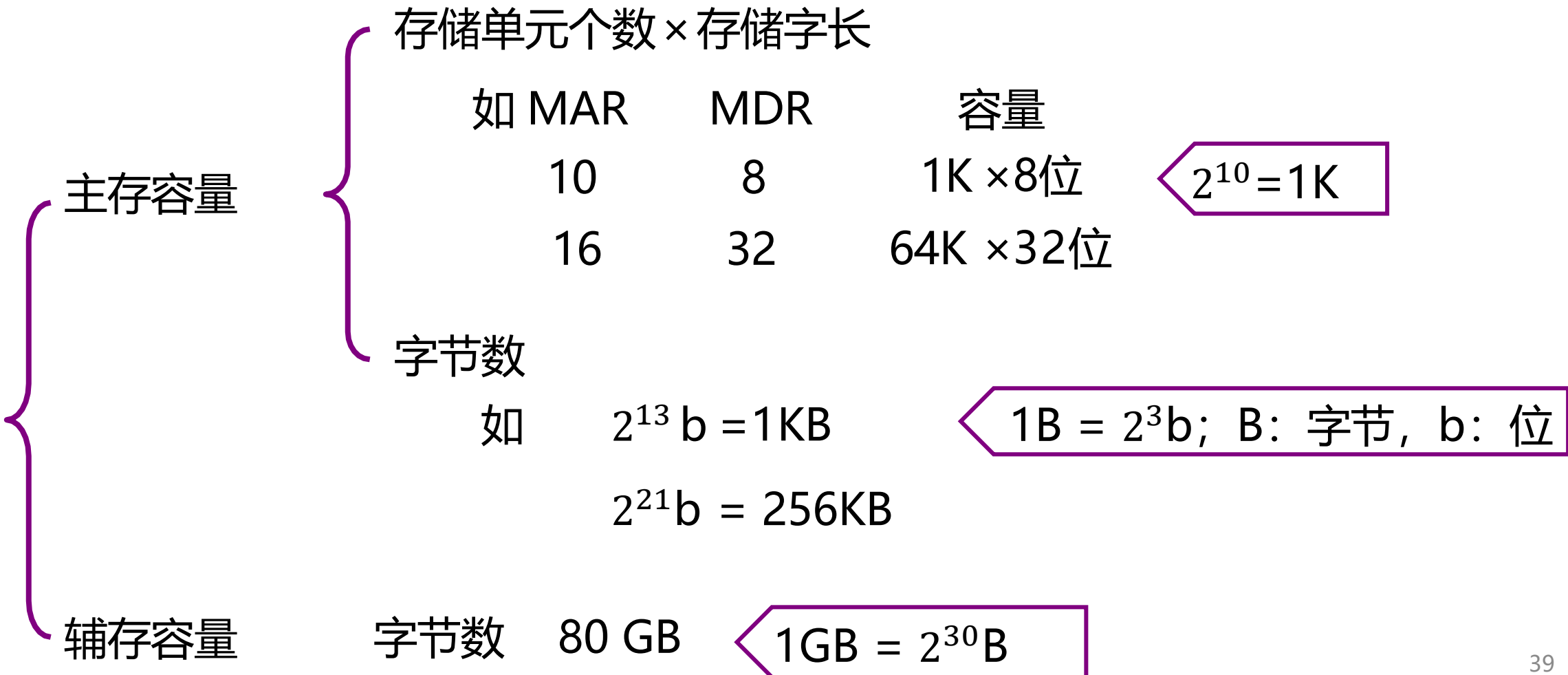
- 时钟周期 T ：控制计算机操作的最小时间单位
- 主频 f ：CPU的时钟频率，是时钟周期 T 的倒数
- CPI：执行每条指令所需的平均时钟周期数
- IPC：每个时钟周期平均能执行的指令条数，CPI的倒数
- MIPS：每秒能执行多少百万条指令
 - $MIPS = IPC \times f = f / CPI$
- FLOPS：浮点运算次数每秒

吉普森法

计算机硬件的主要技术指标



■ 存储容量：存放二进制信息的总位数





谢谢！