



计算机组成原理

方淼

计算机与通信工程学院



第 1 章 计算机系统概论

- ▶ 1.1 计算机系统简介
- ▶ 1.2 计算机的基本组成与工作原理
- ▶ 1.3 计算机硬件的主要技术指标
- ▶ 1.4 知识结构



1.1 计算机系统简介

- ▶ 计算应用的分类及特性
- ▶ 后PC时代特点
- ▶ 计算机系统结构的8个伟大思想
- ▶ 计算机系统概念
- ▶ 计算机系统的层次结构
- ▶ 计算机系统的组成和体系结构



1.1.1 计算系统的分类和特性

- ▶ 计算机促进了人类的第三次革命。
- ▶ 过去曾经的计算机科学幻想，如今已经实现
 - ▶ 车载计算机
 - ▶ 手机
 - ▶ 人类基因项目
 - ▶ 万维网
 - ▶ 搜索引擎
- ▶ 如今的科学幻想也正在逐步实现
 - ▶ 虚拟现实
 - ▶ 无现金社会
 - ▶ 自动驾驶



1.1.1 计算系统的分类和特性

▶ 个人计算机 (PC)

- ▶ 用于个人使用的计算机，通常包含图形显示器、键盘和鼠标等。

▶ 服务器 (Server)

- ▶ 用于为多用户运行大型程序的计算机，通常由多个用户并行使用，并且一般通过网络访问。

▶ 超级计算机 (super comeputer)

- ▶ 具有最高性能和成本的一类计算机，一般配置为服务器。

▶ 嵌入式系统 (embedded computer)

- ▶ 嵌入到其他设备中的计算机，一般运行预定义的一个或者一组应用程序。



1.1.2后PC时代的特点

- ▶ 个人移动设备代替了PC
 - ▶ Personal mobile device
 - ▶ 连接到网络上的小型无线设备
 - ▶ 由电池供电
 - ▶ 通过下载app的方式安装软件
 - ▶ 智能手机和平板电脑
- ▶ 云计算代替了传统的服务器
 - ▶ 依赖称为仓储规模计算机的巨型数据中心
 - ▶ 在网络上提供服务的大型服务器集群，一些运营商根据应用需求出租不同数量的服务器。
 - ▶ 软件即服务SAAS
 - ▶ 在网络上以服务的方式提供软件和数据。其运行方式通常不是在本地上设备上运行所有的二进制代码，而是通过诸如运行在本地客户端的浏览器等小程序登录到远程服务器上执行。



1.1.2后PC时代的特点

► 计算性能需求发生变化

- 计算机和内存设计技术有了长足进步，除了嵌入式系统，大多数用户对少占内存的需求降低
- 现代计算机的特征是处理器的并行性和存储器的层次性
- 程序员需要考虑运行在PMD或云上的程序的能效，要求对代码之下的细节更多的理解，增加对计算机组成原理的认知。



1.1.3 计算机系统结构的8个伟大思想

- ▶ 面向摩尔定律的设计
- ▶ 使用抽象简化设计
- ▶ 加速大概率事件
- ▶ 通过并行提高性能
- ▶ 通过流水线提高性能
- ▶ 通过预测提高性能
- ▶ 存储器层次
- ▶ 通过冗余提高可靠性

1.1.4 计算机系统的概念

▶ 系统软件

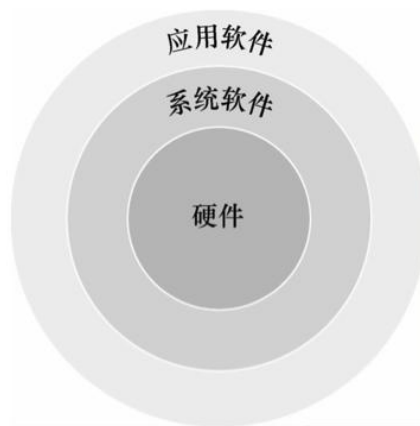
- ▶ 提供常用服务的软件，包括操作系统、编译程序、加载程序、汇编程序等等。

▶ 操作系统：为了使程序更好地在计算机上运行而管理计算机资源的监控程序

- ▶ 处理基本的输入输出
- ▶ 为多个应用程序提供共享计算资源的服务
- ▶ 分配内存和外存

▶ 编译程序

- ▶ 将高级语言翻译为计算机所能识别的机器语言的程序



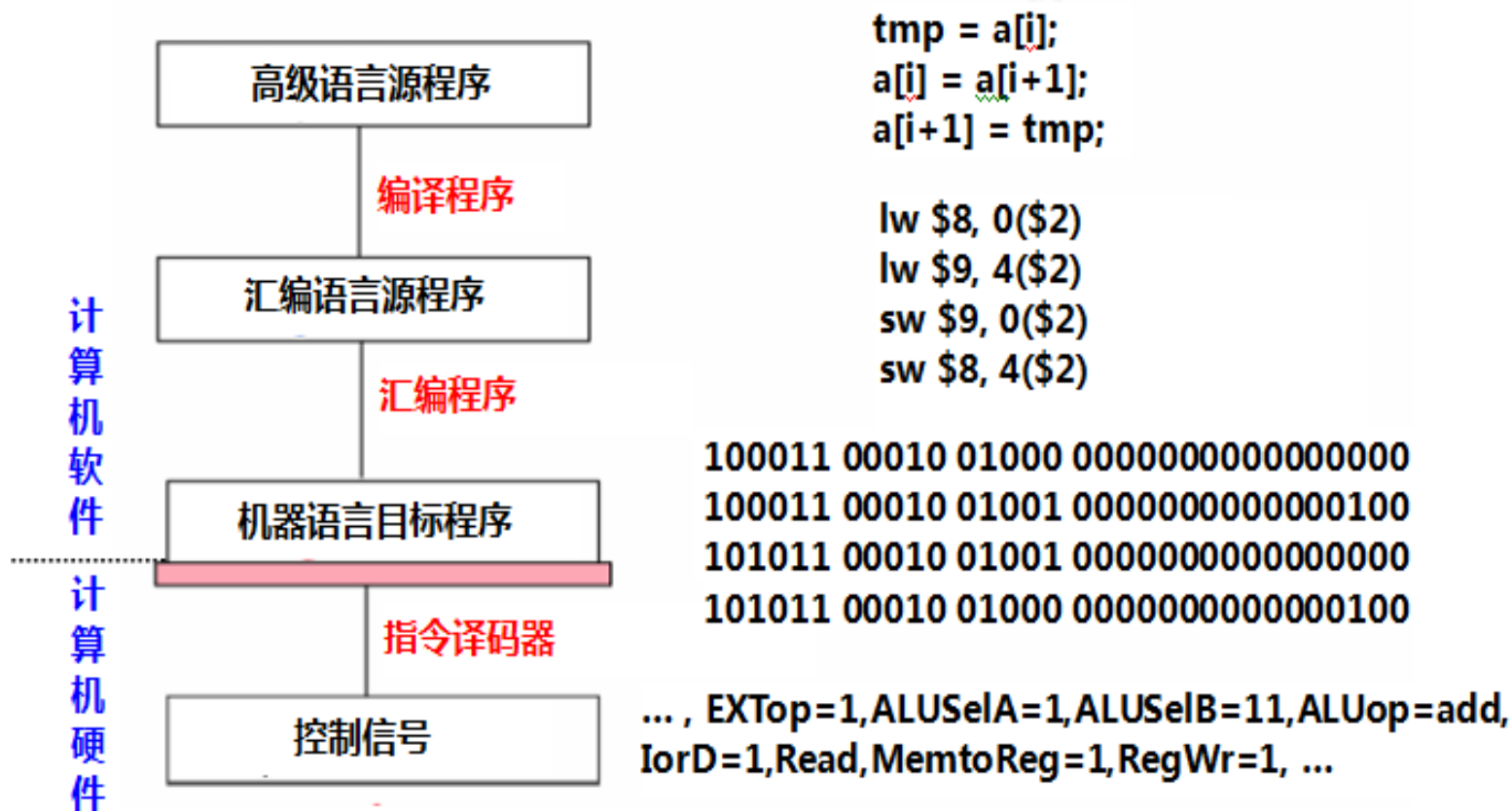
简化的硬件和软件层次图，
将硬件作为同心圆的中心，
应用程序软件作为最外层

在复杂的应用中，通常存在多层应用软件层。例如，一个数据库系统可运行于系统软件之上，而驻留在该系统软件上的某应用又反过来运行在该数据库之上。

<http://blog.csdn.net/jackhej>

1.1.4 计算机系统的概念

► 高级语言到硬件语言再到硬件执行





一个典型程序的转换处理过程

经典的“hello.c”C-源程序

```
#include <stdio.h>

int main()
{
    printf("hello,world\n");
}
```

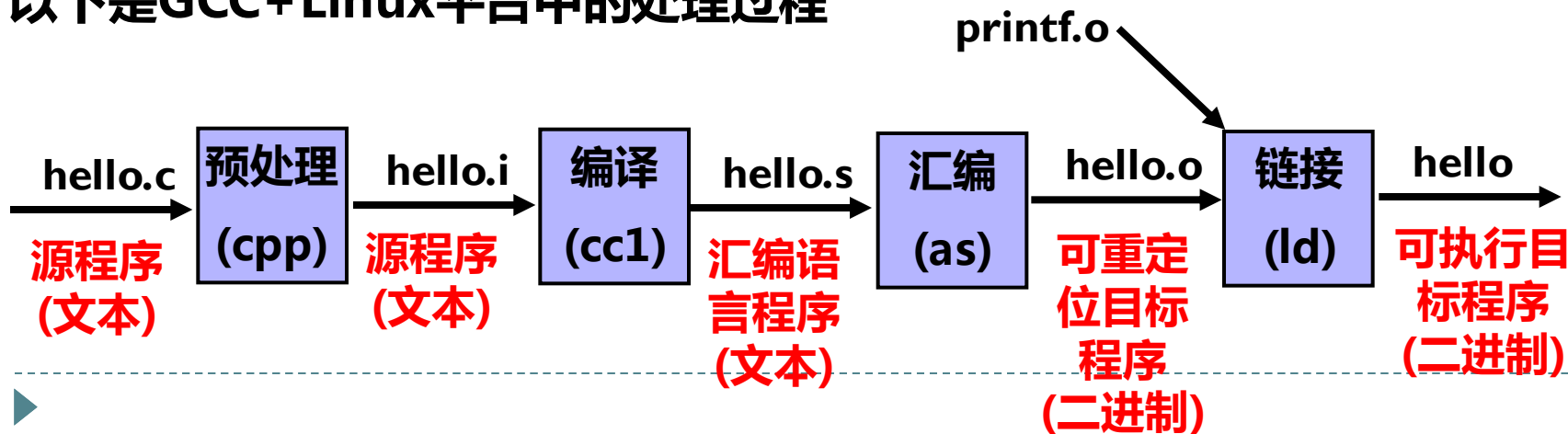
hello.c的ASCII文本表示

```
# i n c l u d e < s p > < s t d i o .
35 105 110 99 108 117 100 101 32 60 115 116 100 105 111 46
h > \n \n i n t < s p > m a i n ( ) \n {
104 62 10 10 105 110 116 32 109 97 105 110 40 41 10 123
\n < s p > < s p > < s p > < s p > p r i n t f ( " h e l
10 32 32 32 32 112 114 105 110 116 102 40 34 104 101 108
l o , < s p > w o r l d \n " ) ; \n }
108 111 44 32 119 111 114 108 100 92 110 34 41 59 10 125
```

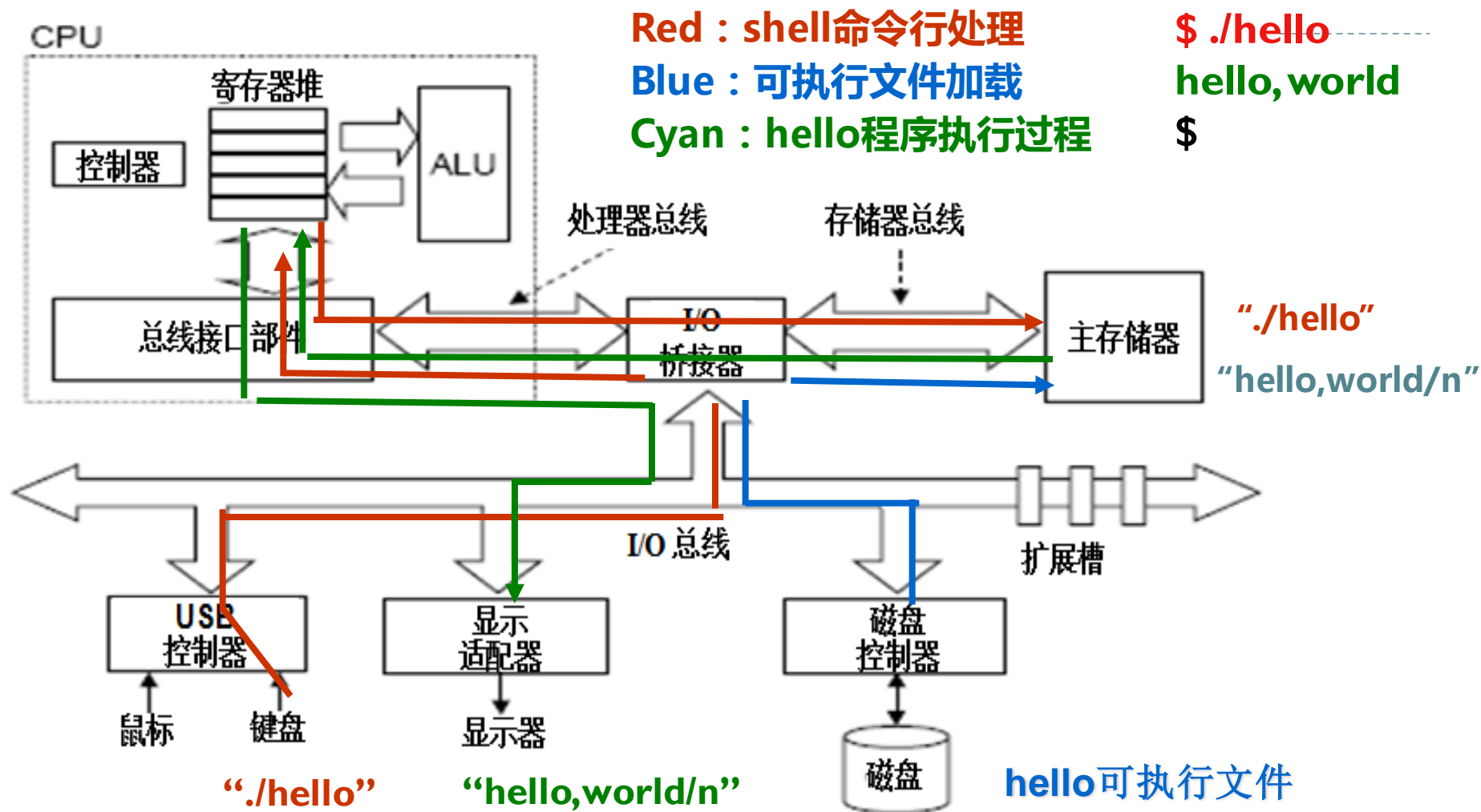
功能：输出“hello,world”

计算机不能直接执行hello.c！

以下是GCC+Linux平台中的处理过程



Hello程序的数据流动过程

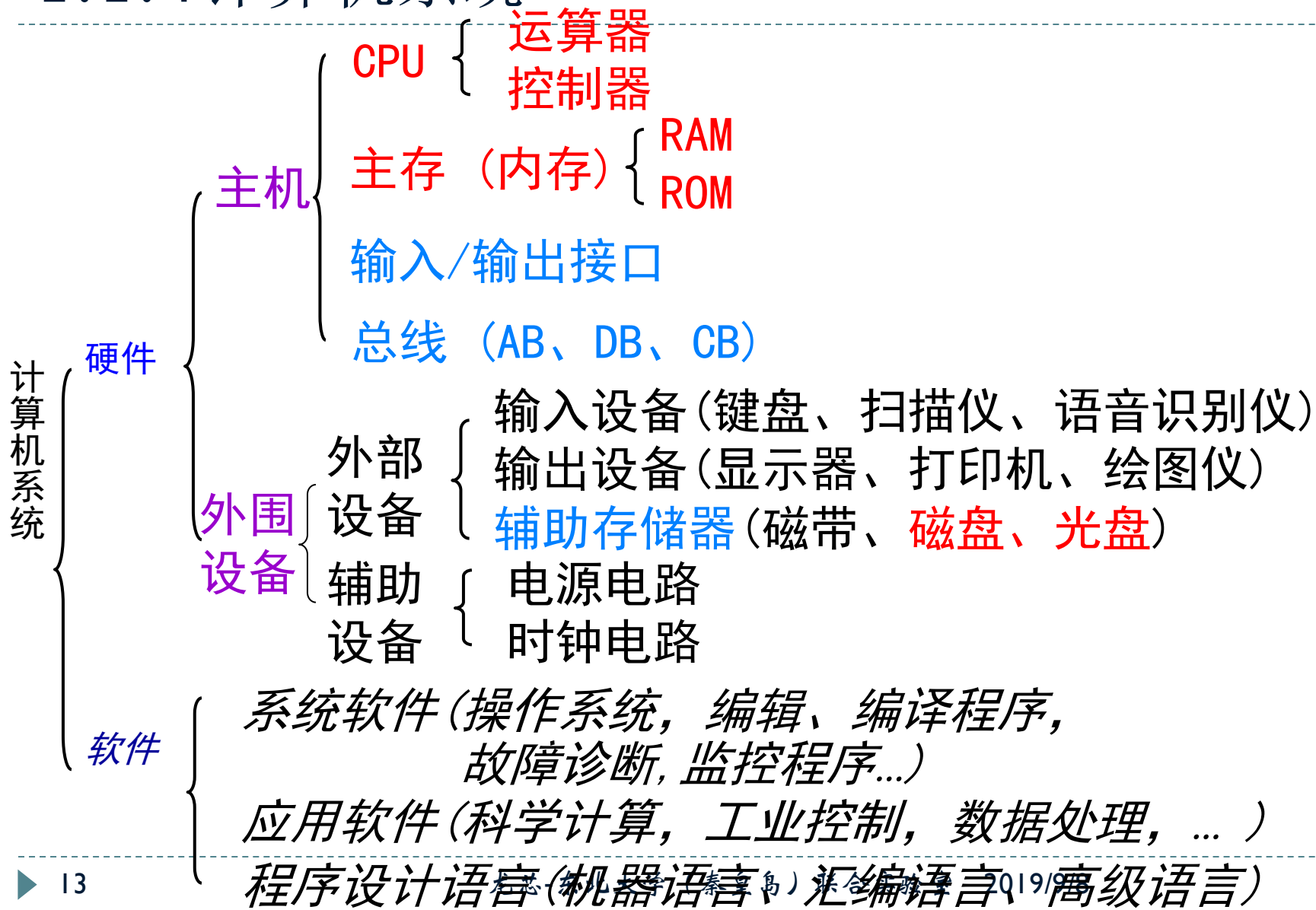


数据经常在各存储部件间传送。故现代计算机大多采用“缓存”技术！

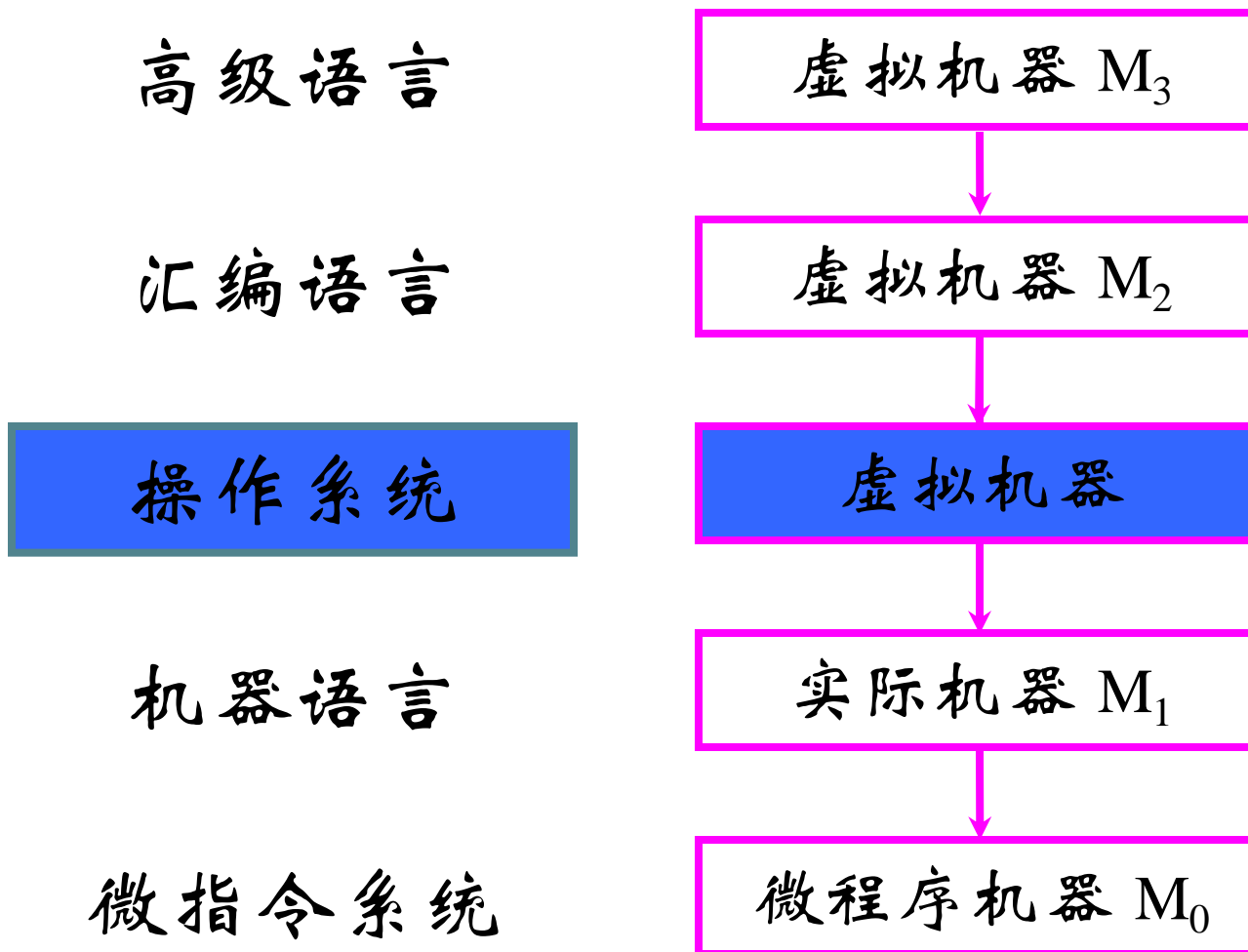
所有过程都是在CPU执行指令所产生的控制信号的作用下进行的。



1.1.4 计算机系统



1.1.5 计算机系统的层次结构



1.1.5 计算机系统的层次结构

软件

虚拟机器 M_4

用编译程序翻译成汇编语言程序

虚拟机器 M_3

用汇编程序翻译成机器语言程序

虚拟机器 M_2

用机器语言解释操作系统

硬件

实际机器 M_1

用微指令解释机器指令

微程序机器 M_0

由硬件直接执行微指令



计算机系统抽象层的转换

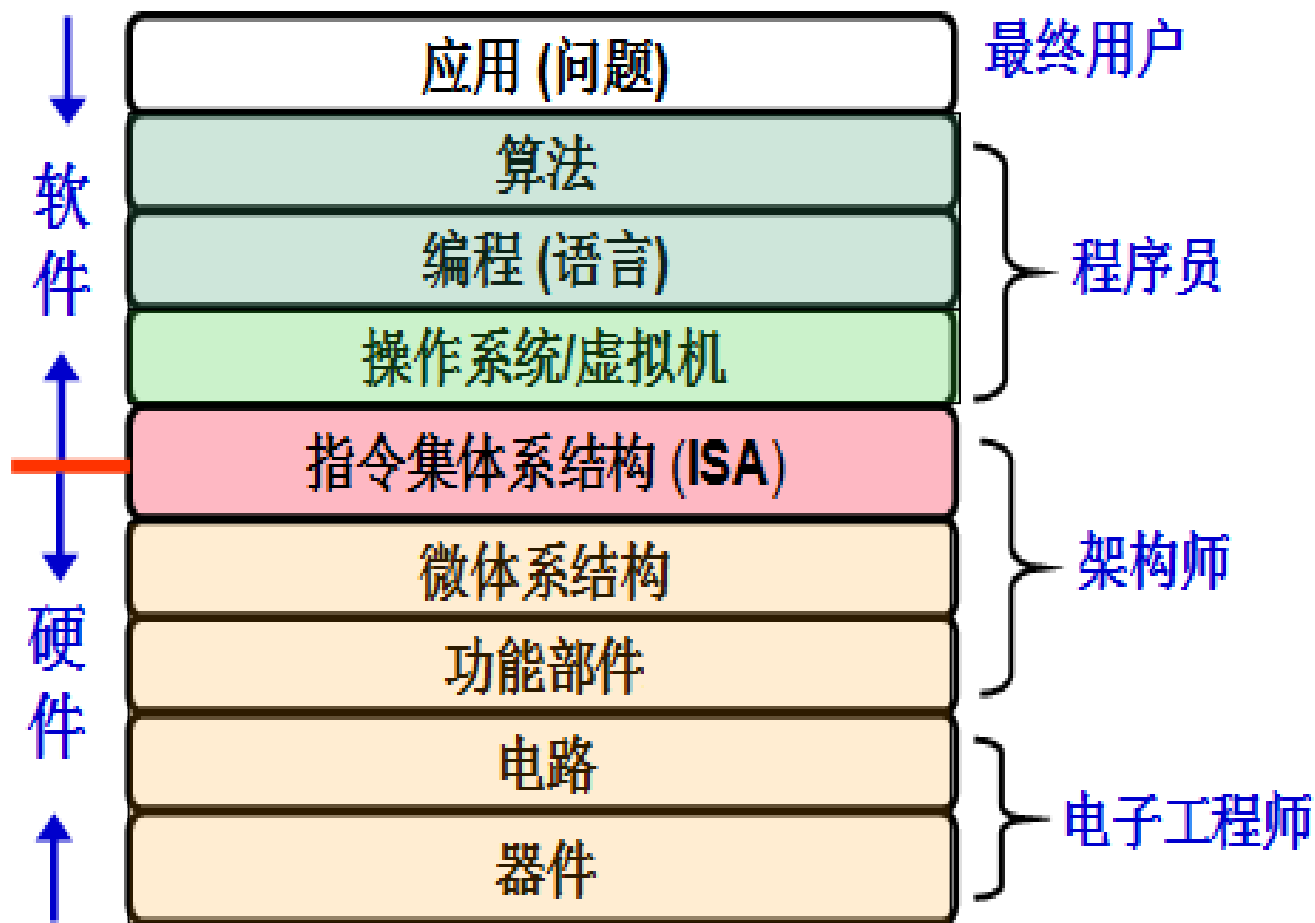
程序执行结果

不仅取决于
算法、程序编写
而且取决于
语言处理系统
操作系统
ISA
微体系结构

不同计算机课程
处于不同层次

必须将各层次关
联起来解决问题

功能转换：上层是下层的抽象，下层是上层的实现
底层为上层提供支撑环境！



最高层抽象就是点点鼠标、拖拖图标、敲敲键盘，但这背后有多少层转化啊！



计算机系统的不同用户

最终用户工作在由应用程序提供的最上面的抽象层

系统管理员工作在由操作系统提供的抽象层

应用程序员工作在由语言处理系统（**主要有编译器和汇编器**）的抽象层

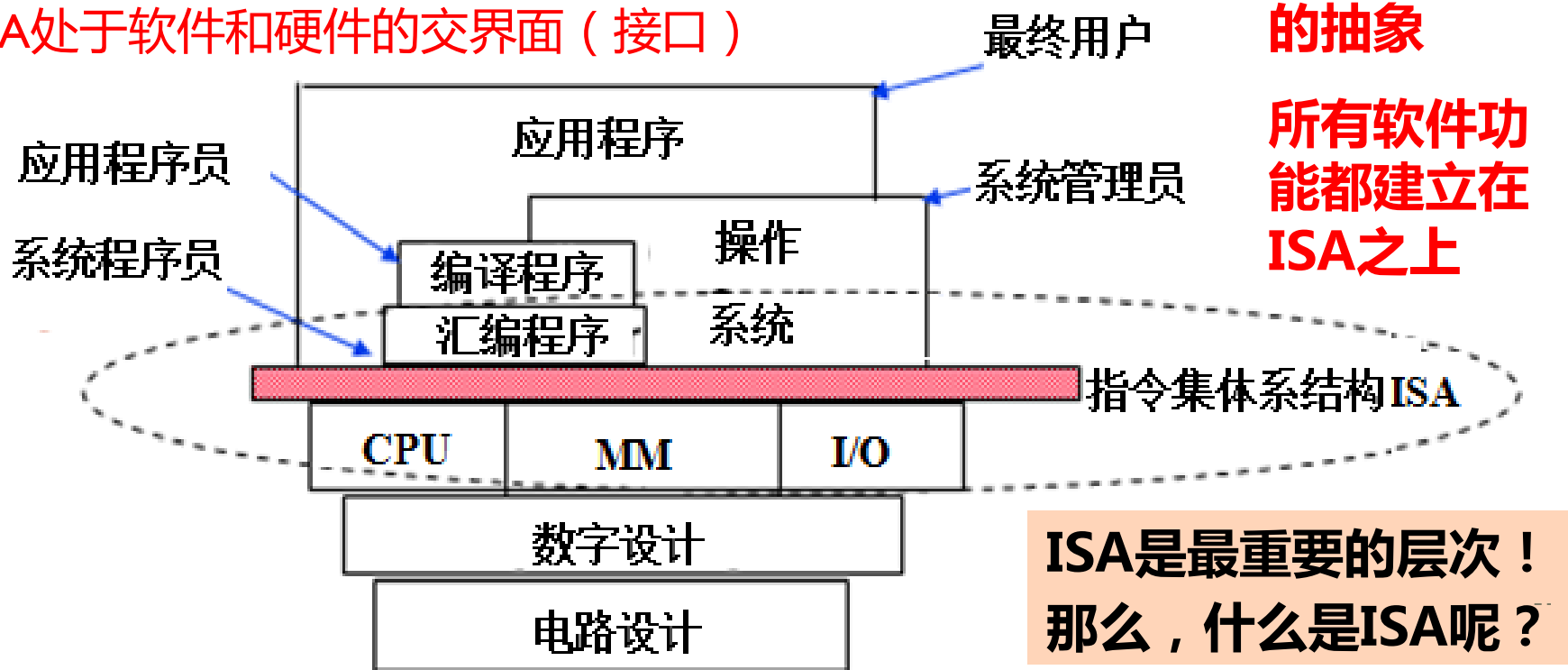
语言处理系统建立在**操作系统**之上

系统程序员（实现系统软件）工作在ISA层次，必须对ISA非常了解

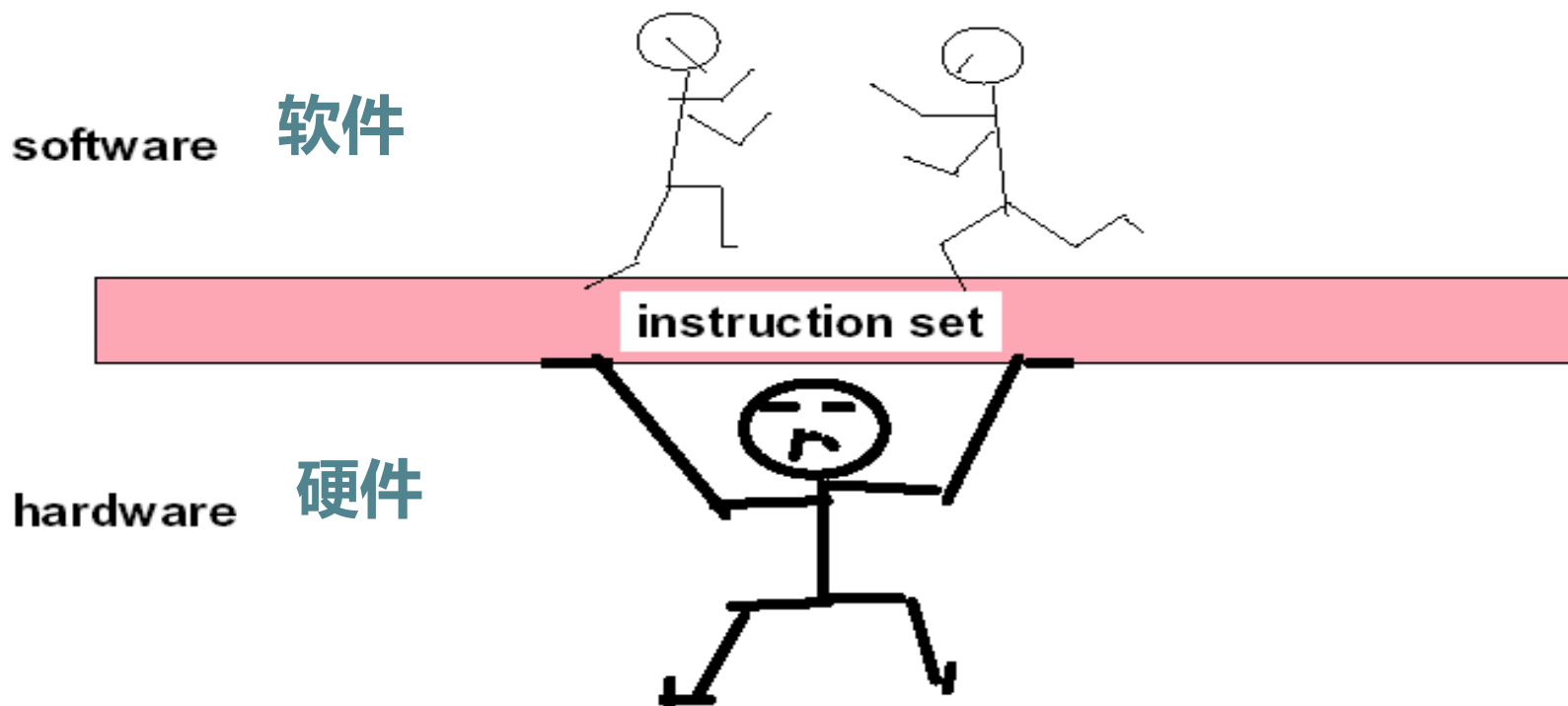
编译器和汇编器的目标程序由机器级代码组成

操作系统通过指令直接对硬件进行编程控制

ISA处于软件和硬件的交界面（接口）



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。

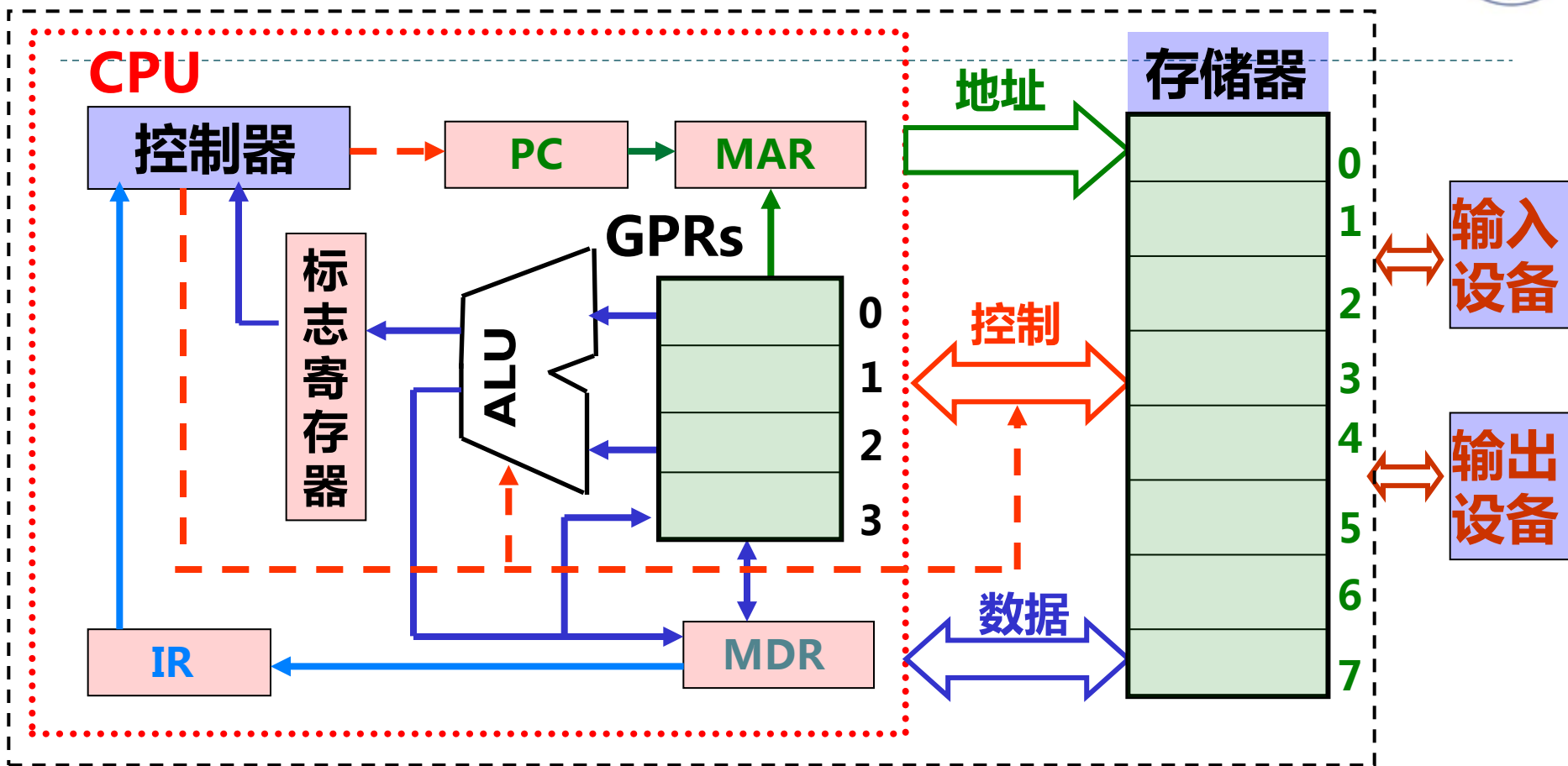


指令集体系结构（ISA）

- ▶ ISA指Instruction Set Architecture，即指令集体系结构
- ▶ ISA是一种规约（Specification），它规定了如何使用硬件
 - ▶ 可执行的指令的集合，包括指令格式、操作种类以及每种操作对应的操作数的相应规定；
 - ▶ 指令可以接受的操作数的类型；
 - ▶ 操作数所能存放的寄存器组的结构，包括每个寄存器的名称、编号、长度和用途；
 - ▶ 操作数所能存放的存储空间的大小和编址方式；
 - ▶ 操作数在存储空间存放时按照大端还是小端方式存放；
 - ▶ 指令获取操作数的方式，即寻址方式；
 - ▶ 指令执行过程的控制方式，包括程序计数器、条件码定义等。
- ▶ ISA在计算机系统中是必不可少的一个抽象层，Why？
 - ▶ 没有它，软件无法使用计算机硬件！
 - ▶ 没有它，一台计算机不能称为“通用计算机”微体系结构

ISA和计算机组成（Organization，即MicroArchitecture）是何关系？

ISA和计算机组成（微结构）之间的关系

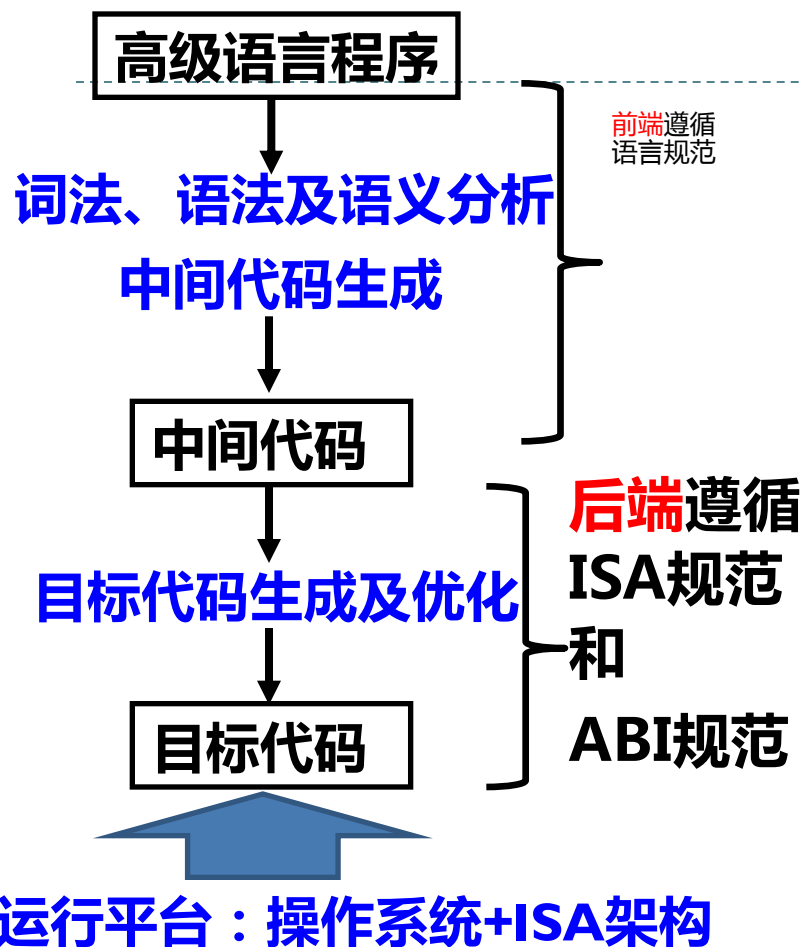


ISA是计算机组成的抽象

不同ISA规定的指令集不同，如，IA-32、MIPS、ARM等
计算机组成必须能够实现ISA规定的功能，如提供GPR、标志、运算电路等
同一种ISA可以有不同的计算机组成，如乘法指令可用ALU或乘法器实现



计算机系统核心层之间的关联



执行结果不符合程序开发者预期举例：

C90中， $-2147483648 < 2147483647$

结果为false

```
int x=1234 ;  
printf( "%lf" ,x);
```

} 未定义行为

不同平台结果不同，相同平台每次结果不同

结果不符合预期的原因通常有两种：

- (1) 程序员不了解语言规范；
- (2) 程序含有未定义行为 (undefined behavior) 或未确定行为 (unspecified behavior) 的语句

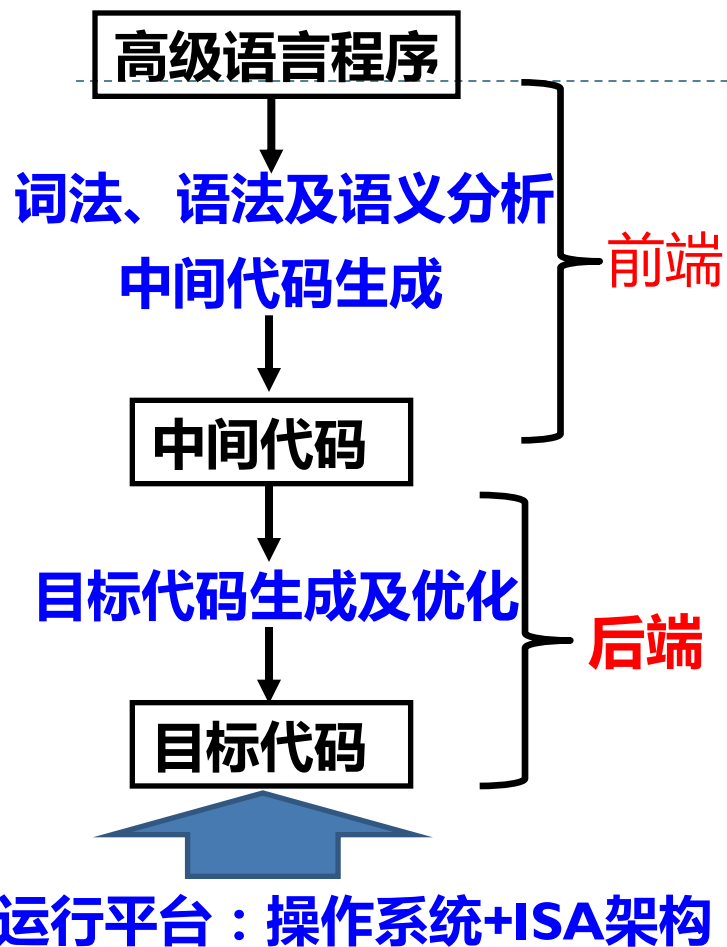
ABI是为运行在特定ISA及特定操作系统之上的应用程序所遵循的一种机器级目标代码层接口

描述了应用程序和操作系统之间、应用程序和所调用的库之间、不同组成部分 (如过程或函数) 之间在较低层次上的机器级代码接口。

后端根据ISA规范和应用程序二进制接口 (Application Binary Interface, ABI) 规范进行设计实现。



计算机系统核心层之间的关联



ABI是为运行在**特定ISA及特定操作系统**之上的应用程序所遵循的一种机器级目标代码层**接口规约**。例如：

过程间调用约定（参数和返回值传递等）
系统调用约定（系统调用的参数和调用号如何传递以及如何从用户态陷入操作系统内核等）
目标文件的二进制格式
函数库使用约定
寄存器使用规定
程序的虚拟地址空间划分
等等

后端根据ISA规范和**应用程序二进制接口**（**Application Binary Interface, ABI**）规范进行设计实现。

要了解程序的确切行为，最好的方法就是查手册！



三、计算机体系结构和计算机组成

有无乘法指令

计算机
体系结构

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



三、计算机体系结构和计算机组成

▶ 1. 计算机系统结构的定义

- ▶ 计算机体系结构就是程序员所看到的计算机的基本属性，即概念性结构与功能特性。

▶ 2. 计算机组成的定义

- ▶ 计算机组成是指实现体系结构功能特性的操作部件及其内在联系，包括机器内部的数据流和控制流的组成以及逻辑设计等。

▶ 3. 计算机实现的定义

- ▶ 计算机实现是指计算机组成的物理实现。它着眼于器件技术与组装技术

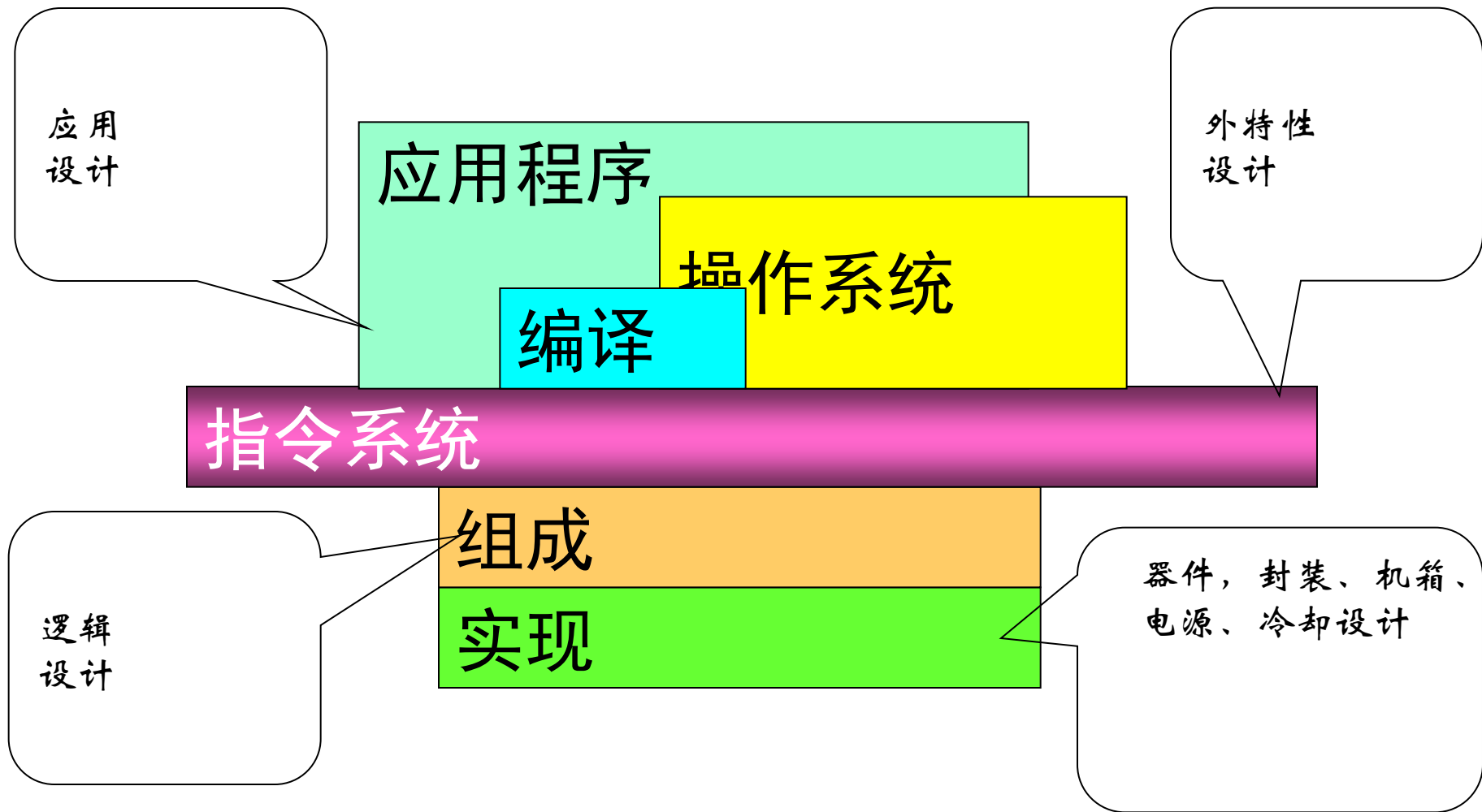


三、计算机体系结构和计算机组成

▶ 4、计算机系统结构、组成、实现三者的关系

- ▶ 计算机系统结构、计算机组成和计算机实现是三个不同的概念。
 - ▶ 计算机系统结构是指计算机系统的软件与硬件之间的接口
 - ▶ 计算机组成是指计算机系统结构的逻辑实现
 - ▶ 计算机实现是指计算机组成的物理实现。
- ▶ 三者包含的内容不同，但却有着紧密的联系。

补充：计算机系统设计



补充：计算机系统设计（续）

设计级	基本部件	信息单元	目标部件	依据和描述方法
处理器级	CPU, IOP, MEM, I/O设备	字块, 数据块	复杂系统	用途
寄存器级 ▲（硬件组成级）	寄存器, 组合与时序逻辑电路	字	CPU, IOP, MEM, 接口电路, 专用功能部件等	指令系统
门级(硬件实现级)	逻辑门, 触发器	位	寄存器, 组合与时序逻辑电路	真值表, 状态表, 布尔代数, 卡诺图等

▲ **计算机组成的任务：**是在计算机系统结构确定分配给硬件子系统的功能及其概念结构之后，研究各部分内部构造和相互联系，以实现机器指令级（处理器级）的各种功能和特性，包括各功能部件的配置，相互连接和相互作用。



补充：计算机系统设计（续）

▶ 计算机系统的逐级生成过程：

- ▶ 拟定指令系统（它是CPU硬、软界面所在）
- ▶ 创建硬件系统（称为硬核。能实现指令功能）
- ▶ 配置操作系统（它是系统软件的核心和基础）
- ▶ 配置语言处理程序及软件资源（将它们置于操作系统的管理调度之下）
- ▶ 输入用户程序



1.2 计算机的基本组成和工作原理

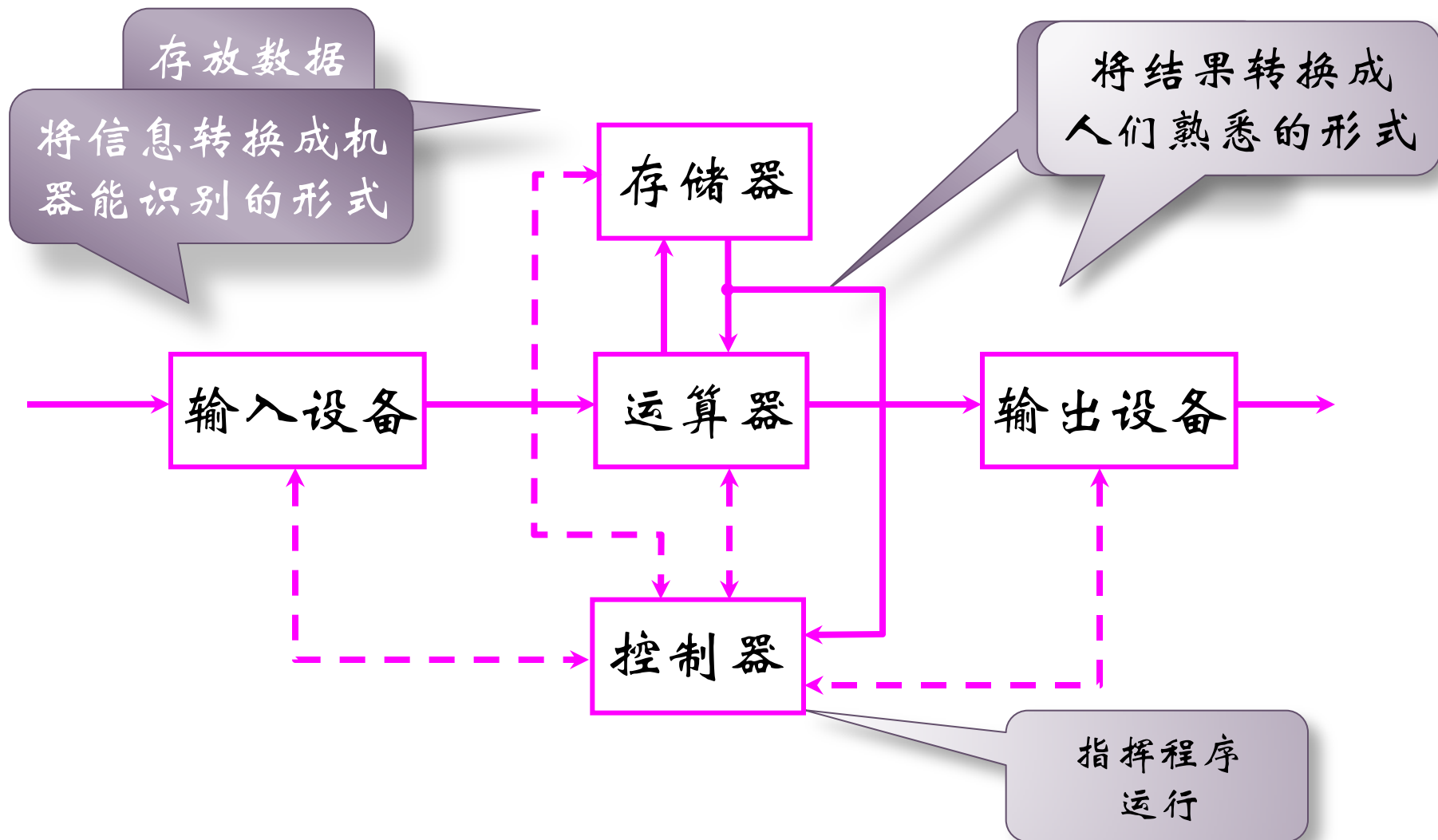
- ▶ 冯·诺依曼机的特点
- ▶ 计算机的硬件框图
- ▶ 计算机的工作过程



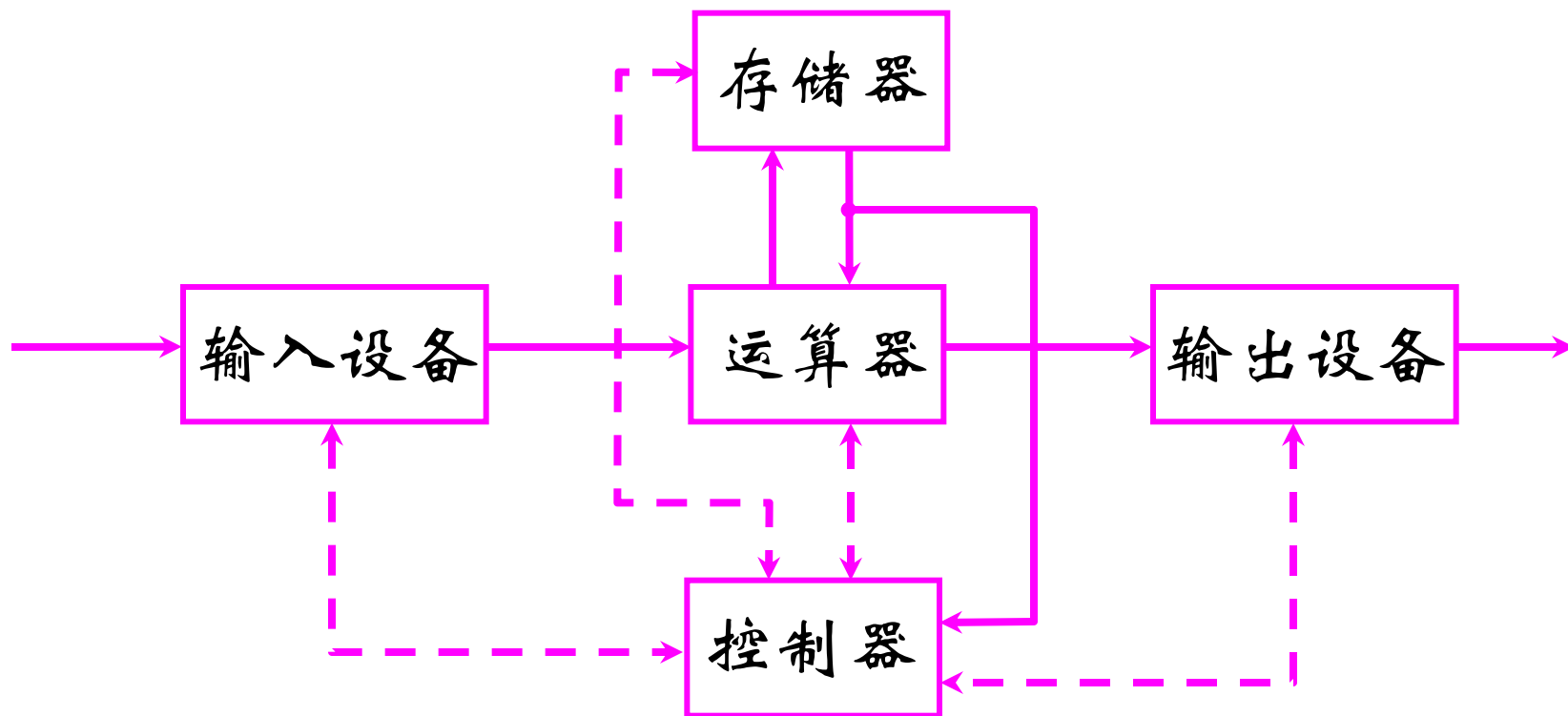
1.2.1 冯·诺依曼计算机的特点

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

1.2.2 冯 诺 依 曼 计 算 机 硬 件 框 图

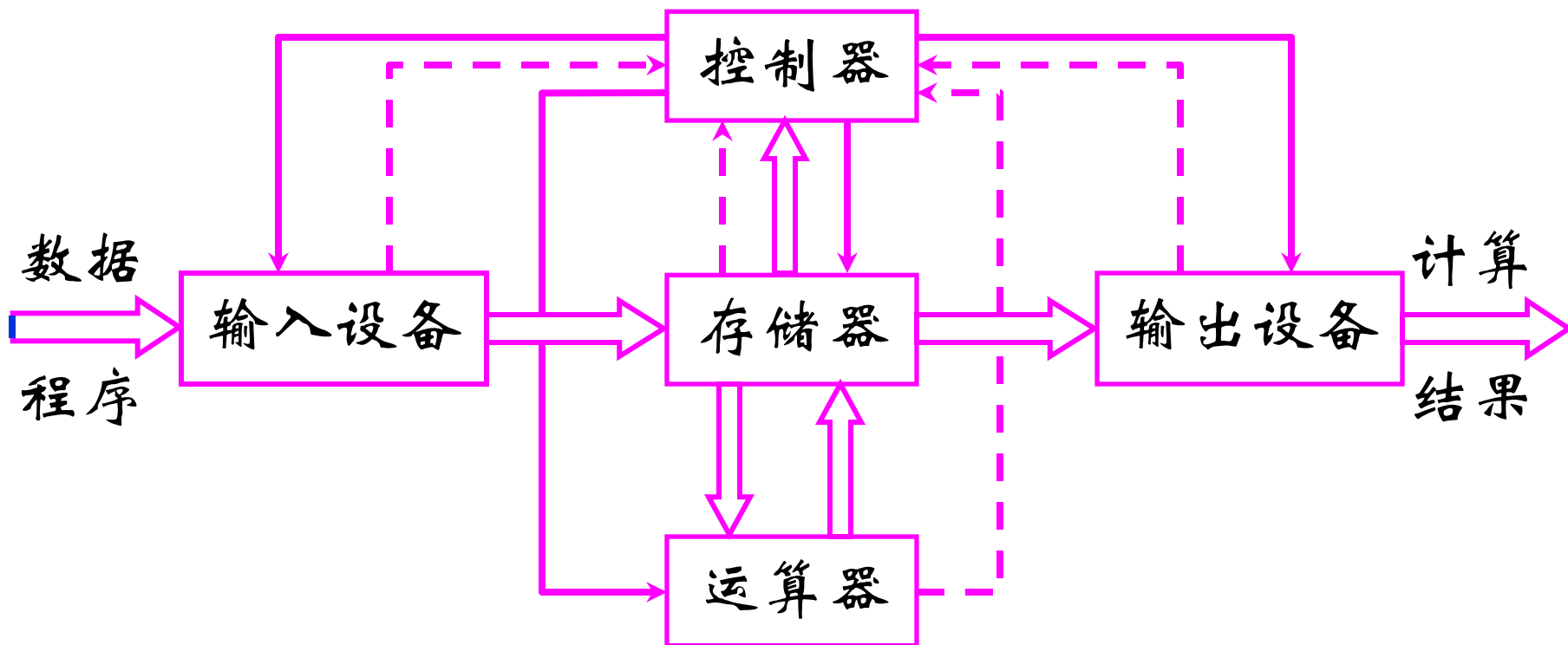


1.2.2 冯 诺 依 曼 计 算 机 硬 件 框 图



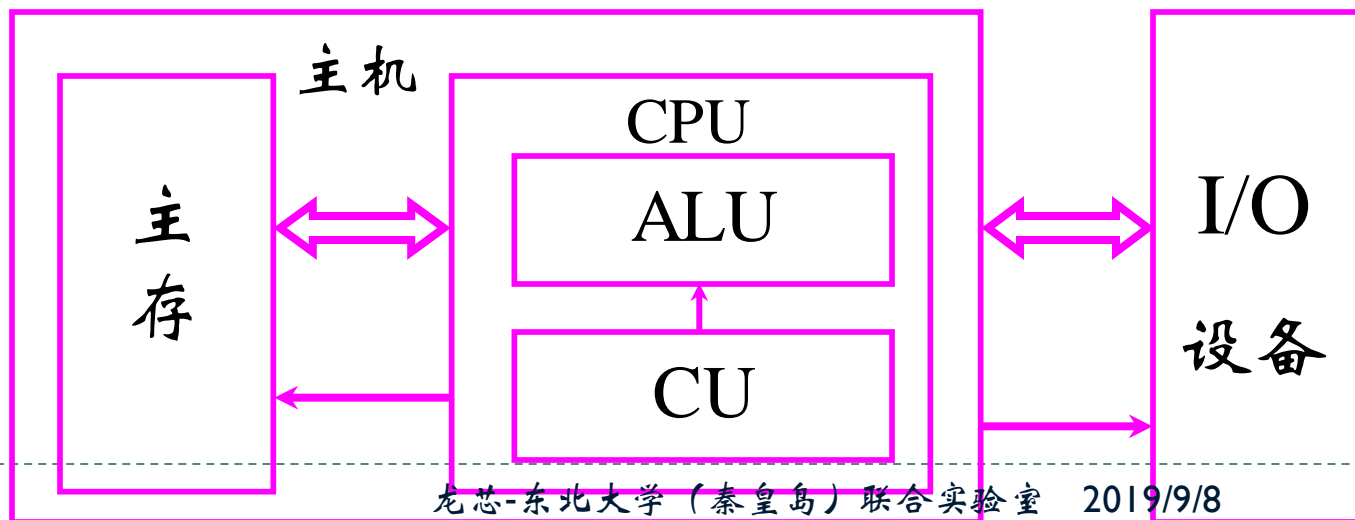
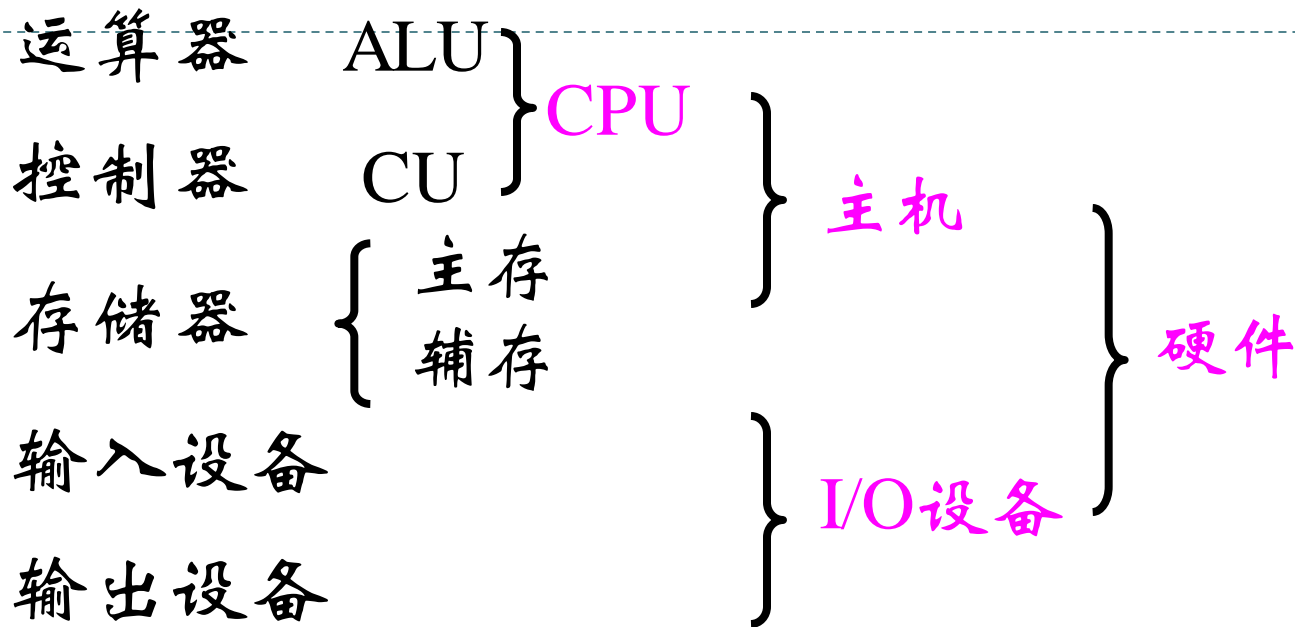
1.2.2 计算机硬件框图

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



三、计算机的工作步骤

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

编程举例

$$\text{计算} \quad ax^2 + bx + c = (ax + b)x + c$$

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

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

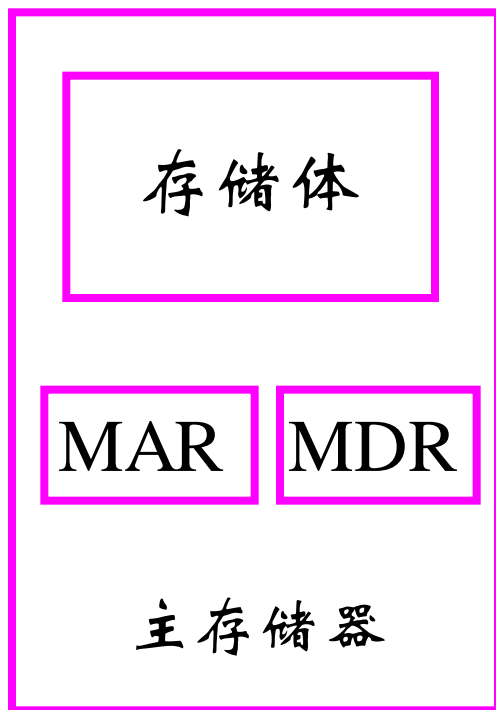


计算 $ax^2 + bx + c$ 程序清单

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
38	12	龙芯-东北大学(秦皇岛)联合实验室 2019/9/8	

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

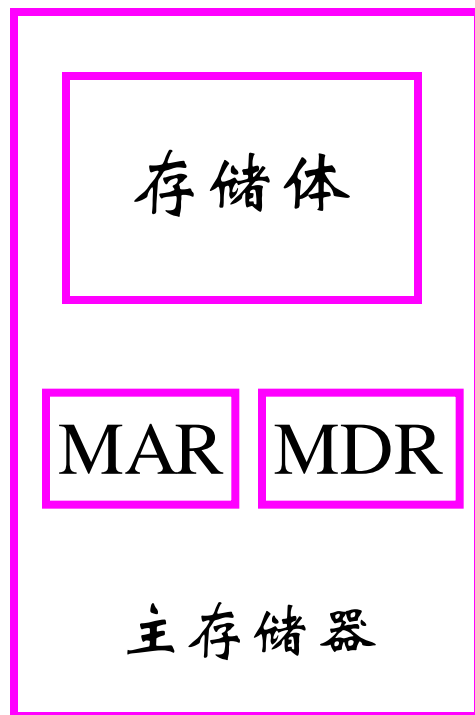
存储字 存储单元中二进制代码的组合

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

按地址寻访

(1) 存储器的基本组成



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



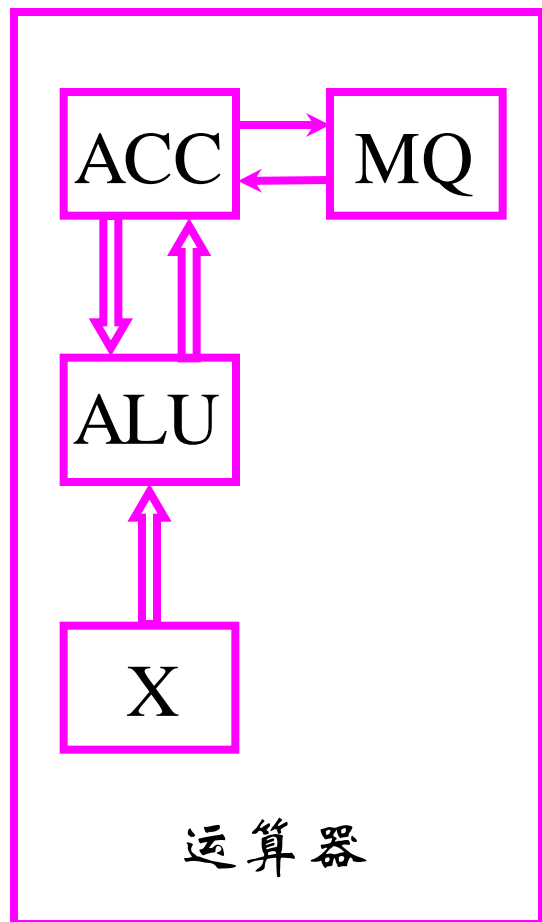
设 MAR = 4 位

MDR = 8 位

存储单元个数 16

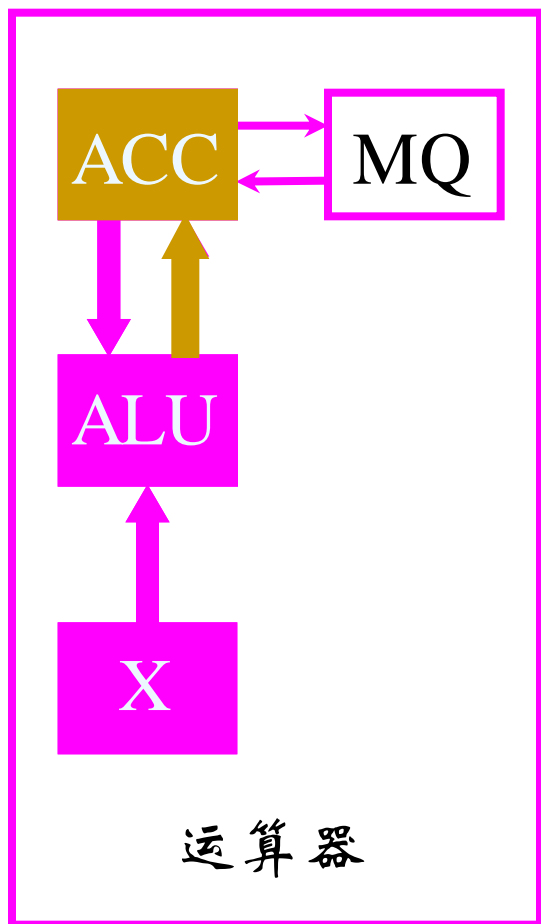
存储字长 8

(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

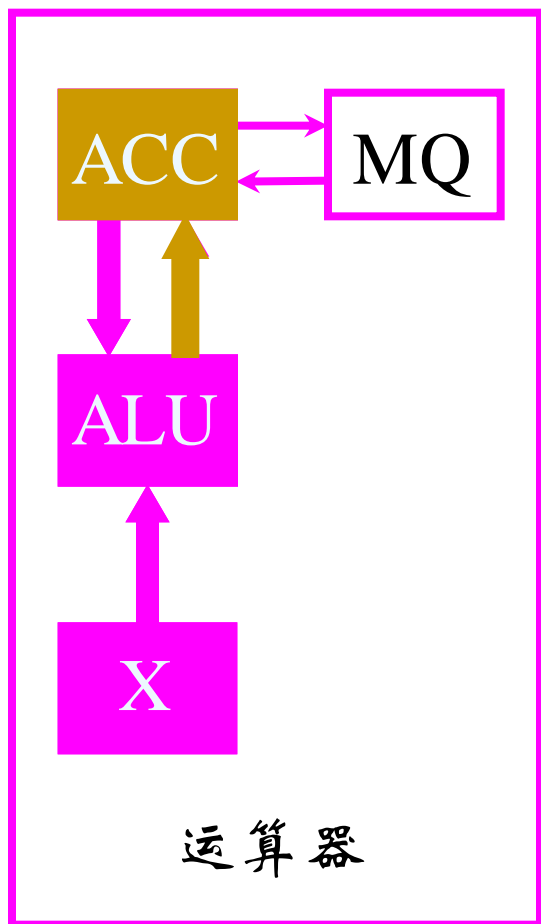
ACC

被加数

$[M] \longrightarrow X$

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

② 减法操作过程



指令

减

M

初态

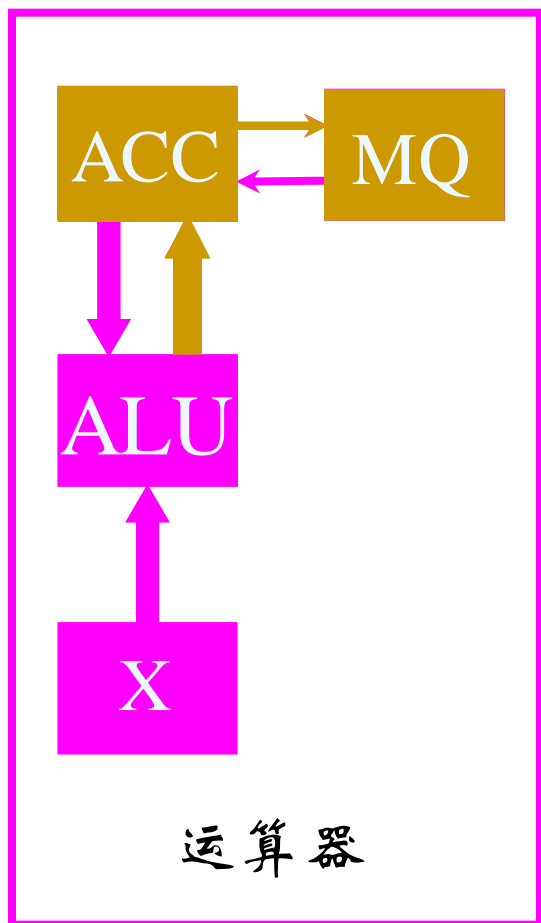
ACC

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

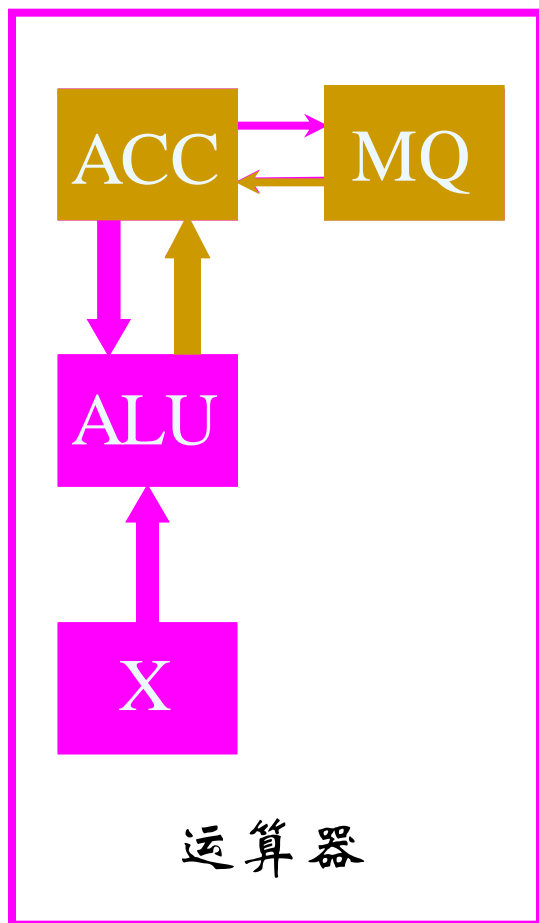
$[M] \longrightarrow MQ$

$[ACC] \longrightarrow X$

$0 \longrightarrow ACC$

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

④ 除法操作过程



指令

除

M

初态

ACC

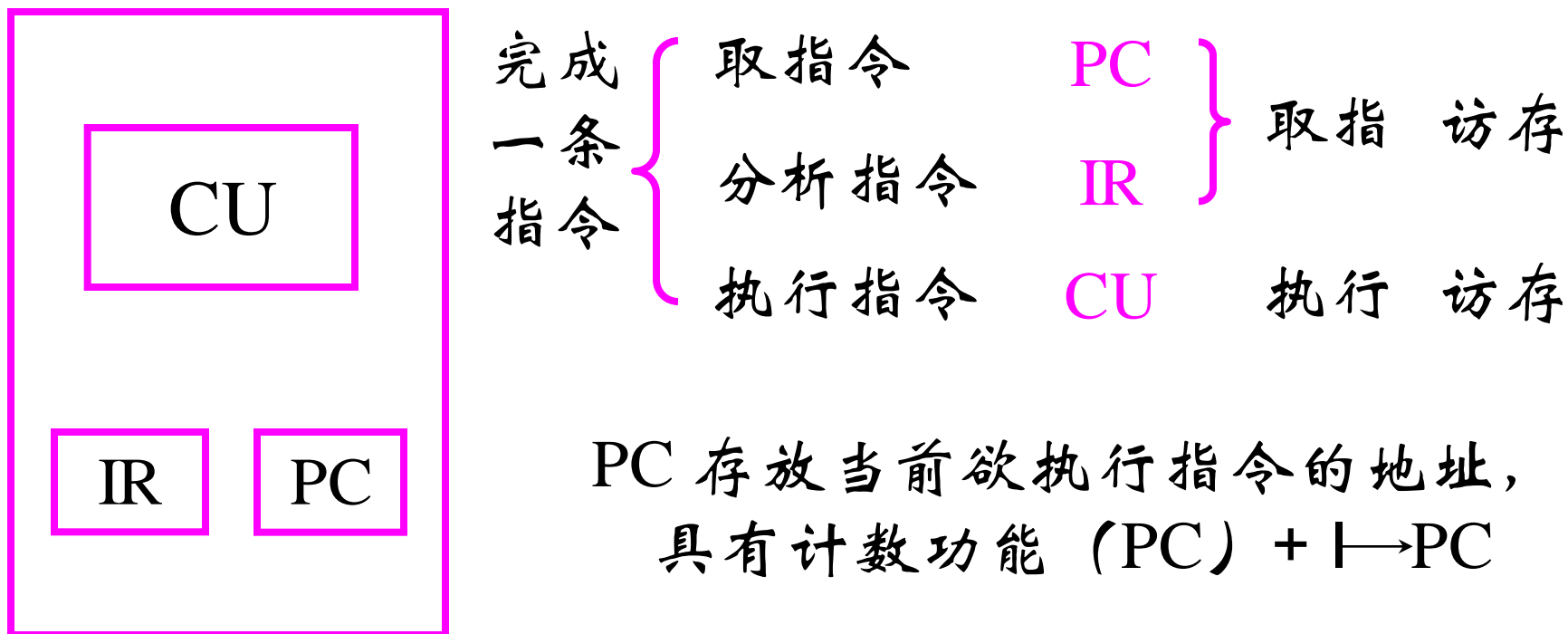
被除数

$[M] \rightarrow X$

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

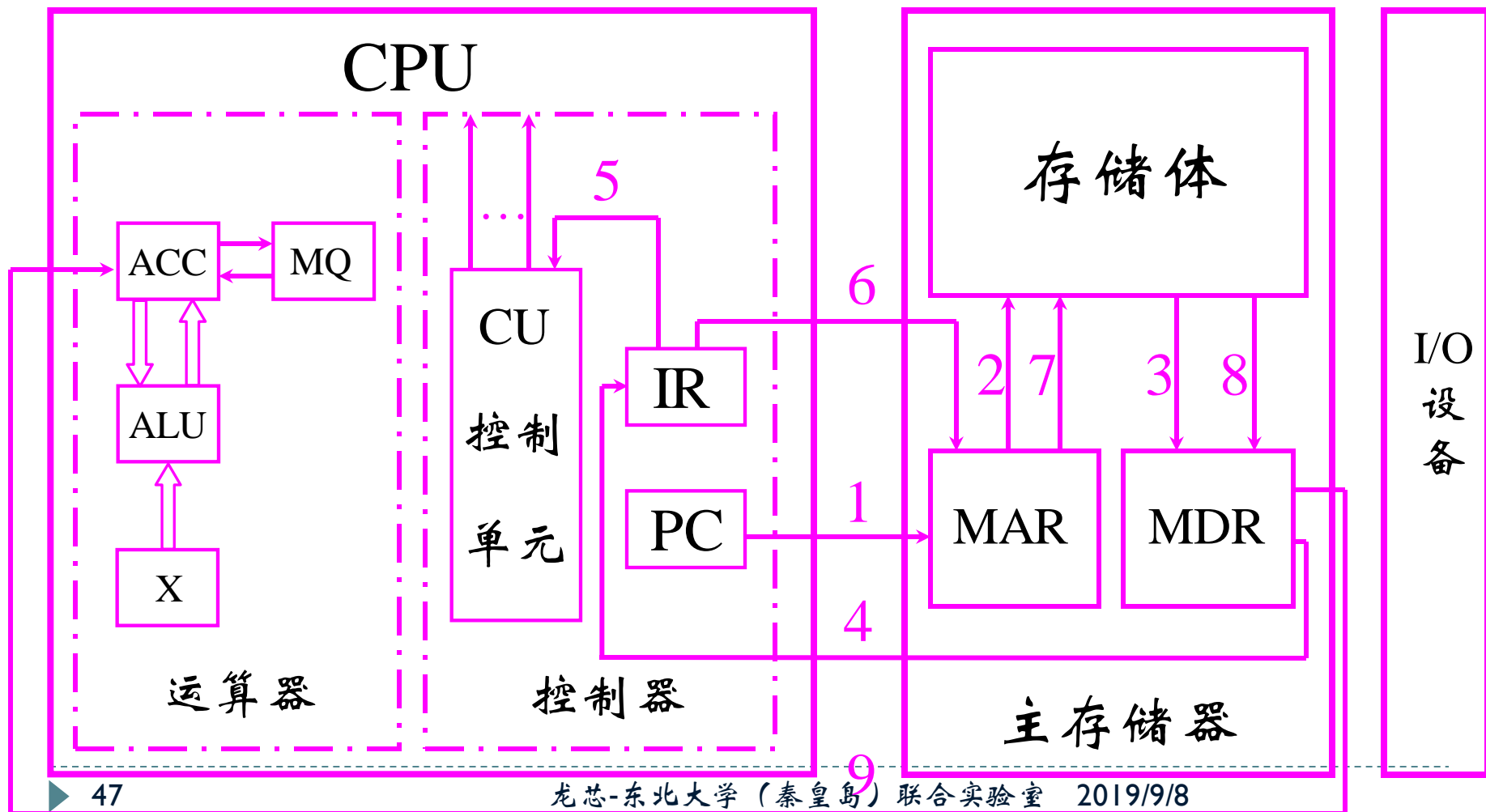
余数在ACC中

(3)控制器的基本组成



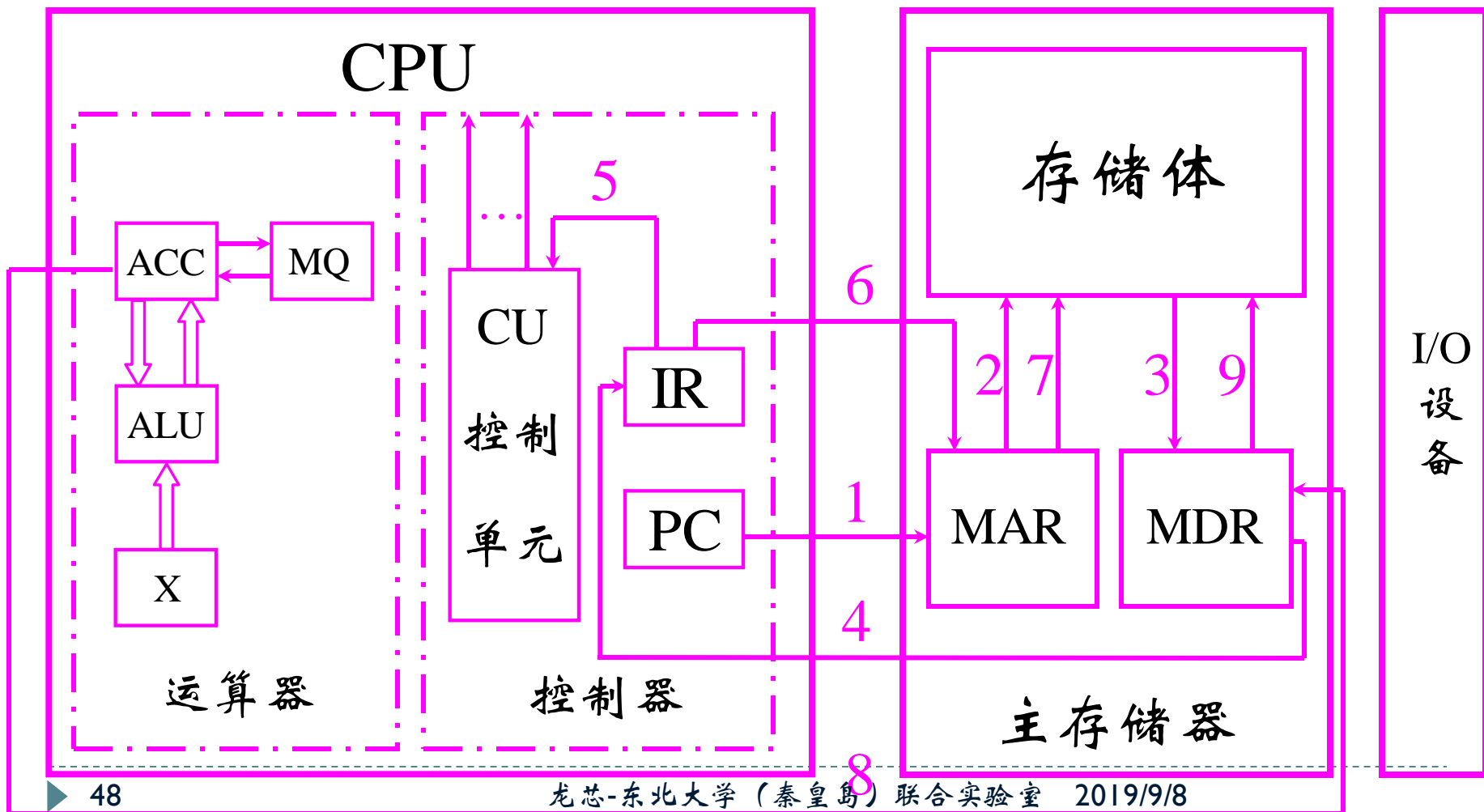
(4)主机完成一条指令的过程

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 $PC \rightarrow MAR \rightarrow M \rightarrow MDR \rightarrow IR$, $(PC) + I \rightarrow PC$
- 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
- 4 停机

1.3 计算机硬件的主要技术指标

- ▶ 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
- ▶ 2.运算速度
 - ▶ 主频：
 - ▶ 即CPU的时钟频率，计算机的操作在时钟信号的控制下分步执行，每个时钟信号周期完成一步操作，时钟频率的高低在很大程度上反映了CPU速度的快慢。主频和实际的运算速度存在一定的关系，但并不是一个简单的线性关系。主频表示在CPU内数字脉冲信号震荡的速度，CPU的运算速度还要看CPU的流水线、总线等各方面的性能指标。
 - ▶ 吉普森法： $T_M = \sum f_i * t_i$
 - ▶ T_M 机器运行速度， f_i 第i种指令占全部操作的百分数， t_i 第i种指令的执行时间

1.3 计算机硬件的主要技术指标

▶ 2.速度

- ▶ MIPS: Million Instruction Per Second, 每秒百万条指令
- ▶ CPI: Cycle Per Instruction, 执行一条指令所需的时钟周期数
- ▶ IPC: Instruction Per Cycle, 每个时钟周期平均执行的指令数。
- ▶ FLOPS: Floating Point Operation Per Second, 每秒执行浮点运算次数
- ▶ MFLOPS, GFLOPS, TFLOPS
- ▶ 加速比:

$$\text{系统加速比} = \frac{\text{系统性能}_{\text{改进后}}}{\text{系统性能}_{\text{改进前}}} = \frac{\text{总执行时间}_{\text{改进前}}}{\text{总执行时间}_{\text{改进后}}}$$

加速比

▶ 1. 大概率事件优先原则

对于大概率事件（最常见的事件），赋予它优先的处理权和资源使用权，以获得全局的最优结果。

▶ 2. Amdahl定律（阿姆达尔定律）

加快某部件执行速度所获得的系统性能加速比，受限于该部件在系统中所占的重要性。

▶ 3. 系统加速比依赖于两个因素：

可改进比例：可改进部分在原系统计算时间中所占的比例，它总是小于等于1的。

例如：一个需运行60秒的程序中有20秒的运算可以加速，那么该比例就是 $20/60$ 。

▶ **部件加速比：**可改进部分改进以后的性能提高，一般情况下它是大于1的。

例如：系统改进后执行程序，其中可改进部分花费2秒的时间，而改进前该部分需花费5秒，则性能提高为 $5/2$ 。

加速比

- ▶ 总执行时间_{改进后} = 不可改进部分的执行时间 + 可改进部分改进后的执行时间

系统加速比为改进前与改进后总执行时间之比：

$$\text{总执行时间}_{\text{改进后}} = (1 - \text{可改进比例}) \times \text{总执行时间}_{\text{改进前}}$$

$$+ \frac{\text{可改进比例} \times \text{总执行时间}_{\text{改进前}}}{\text{部件加速比}}$$

$$\text{系统加速比} = \frac{\text{总执行时间}_{\text{改进前}}}{\text{总执行时间}_{\text{改进后}}}$$

1

$$= \frac{1}{(1 - \text{可改进比例}) + \frac{\text{可改进比例}}{\text{部件加速比}}}$$

$$= \left[(1 - \text{可改进比例}) + \frac{\text{可改进比例}}{\text{部件加速比}} \right] \times \text{总执行时间}_{\text{改进前}}$$

$$= \frac{1}{(1 - Fe) + \frac{Fe}{Se}}$$

加速比

- ▶ 例1：将计算机系统中某一功能的处理速度加快10倍，但该功能的处理时间仅为整个系统运行时间的40%，则采用此增强功能方法后，能使整个系统的性能提高多少？

解：由题可知，可改进比例 $F_e = 40\% = 0.4$ ，部件加速比 $S_e = 10$ ，根据Amdahl定律可知：

$$\text{SpeedUp} = 1.5625$$



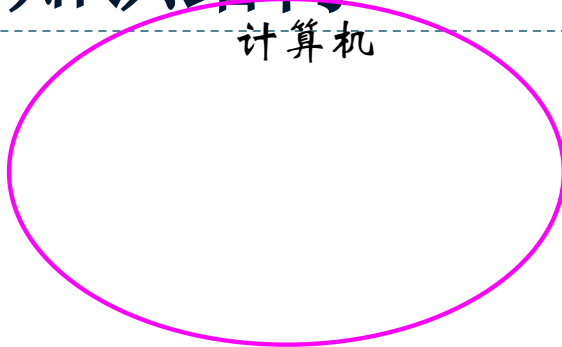
1.3 计算机硬件的主要技术指标

- ▶ 存储容量：存放二进制信息的总位数
- ▶ 主存容量：
 - ▶ 存储单元个数 \times 存储字长
 - ▶ 如 MAR MDR 容量
 - ▶ 10 8 1 K \times 8位
 - ▶ 16 32 64 K \times 32位
 - ▶ 字节数
 - ▶ 如： $2^{13} = 1$ KB
 - ▶ $2^{21} = 256$ KB
- ▶ 辅存容量：
 - ▶ 字节数
 - ▶ 如： 80 GB



1.4 本书知识结构

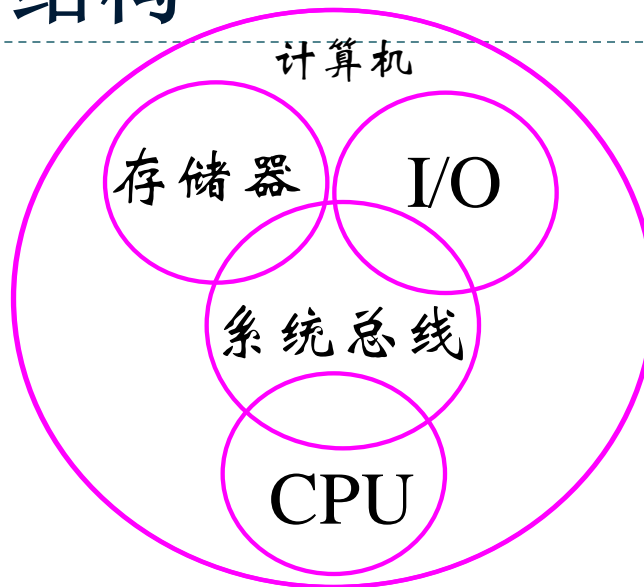
计算机



第 1 篇 概论



1.4 本书结构

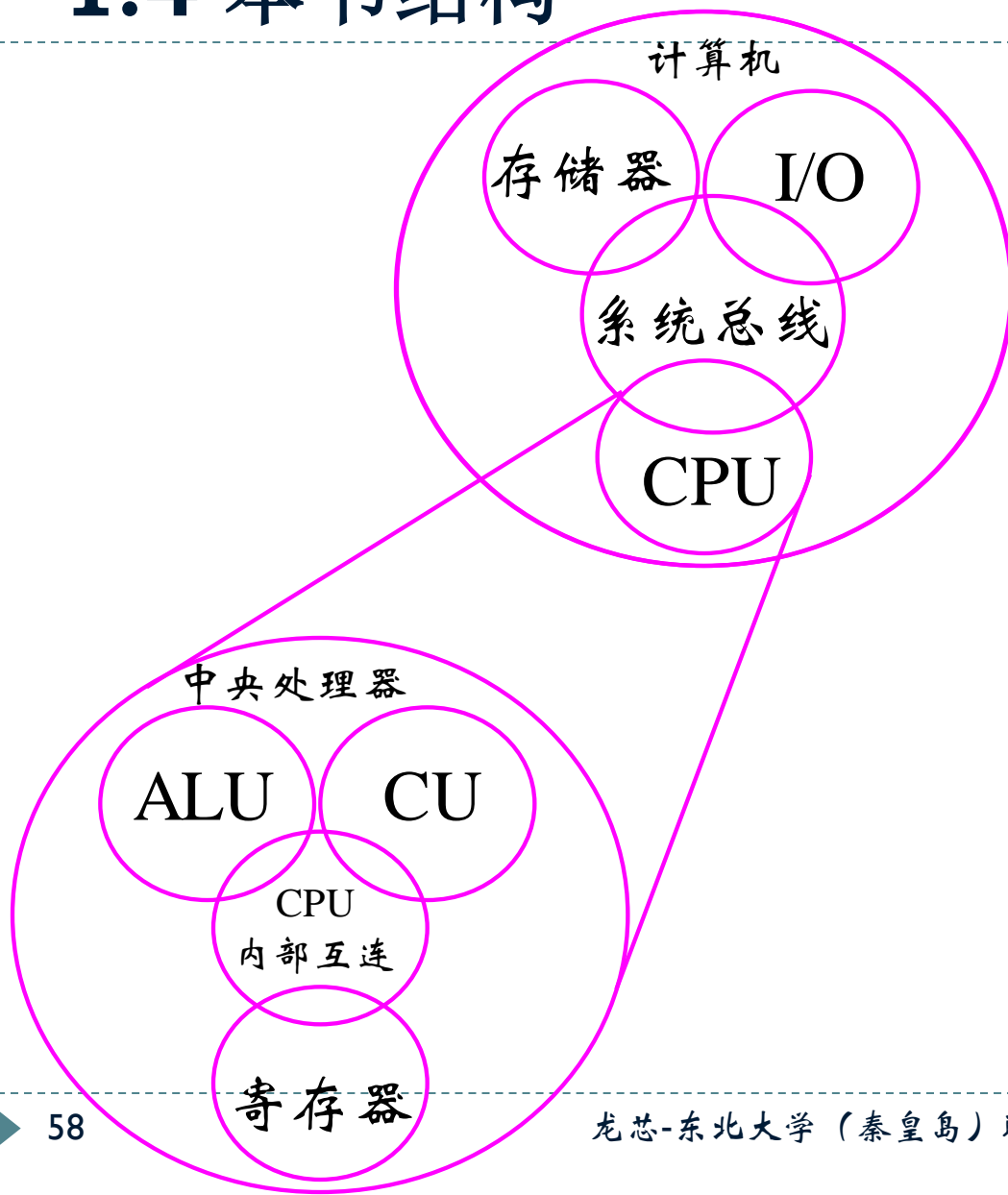


第2篇 计算机系统的硬件结构



1.4 本书结构

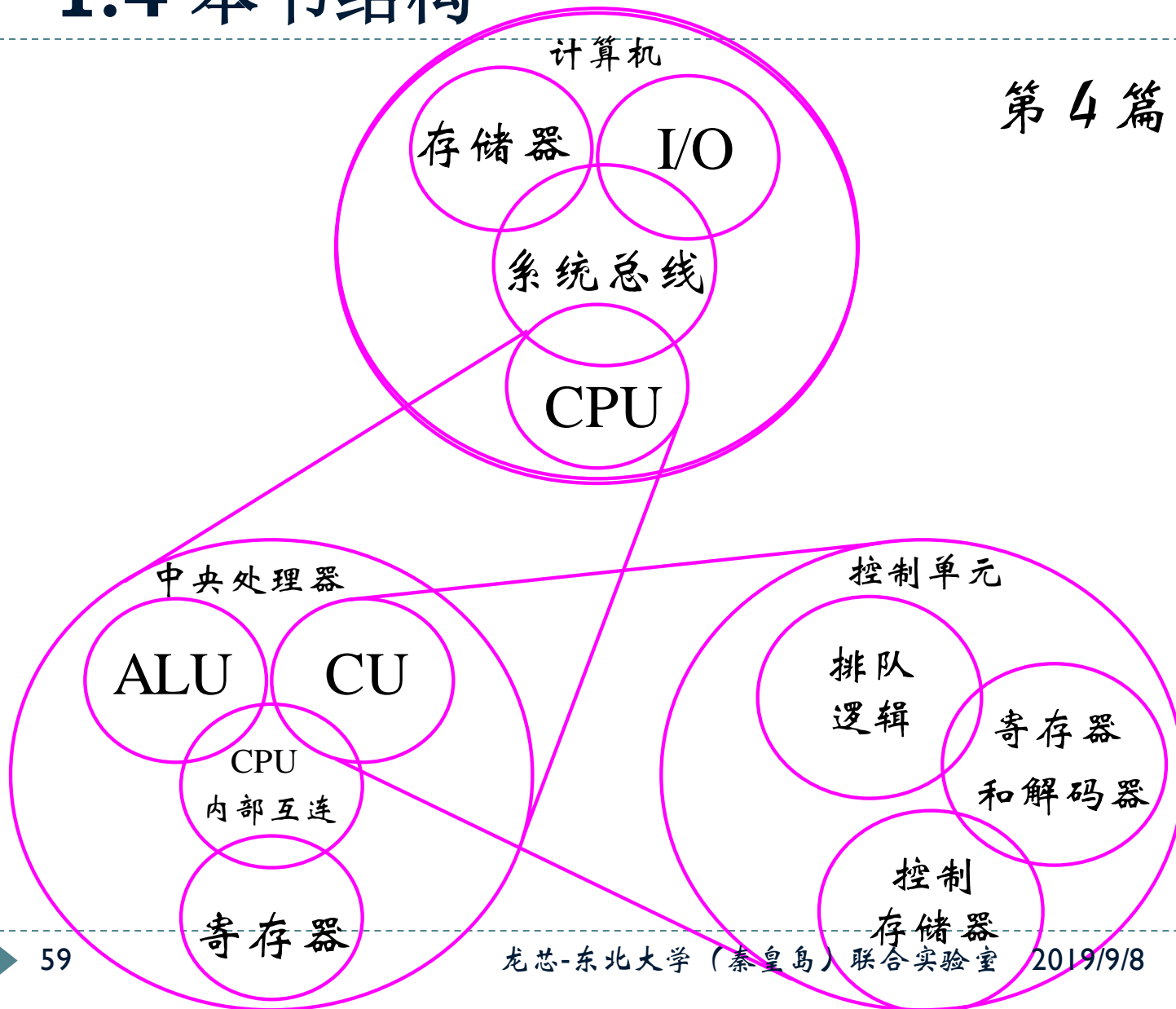
第3篇 CPU





1.4 本书结构

第4篇 CU



例题解析

- **例2** 某台主频为**400MHz**的计算机执行标准测试程序，程序中指令类型、执行数量和平均时钟周期数如下：

指令类型	指令执行数量	平均时钟周期数
整数	45000	1
数据传送	75000	2
浮点	8000	4
分支	1500	2

求该计算机的有效**CPI**、**MIPS**和程序执行时间。

解：（1） $\text{CPI} = (45000 \times 1 + 75000 \times 2 + 8000 \times 4 + 1500 \times 2) / 129500 = 1.776$

（2） $\text{MIPS速率} = f / \text{CPI} = 400 / 1.776 = 225.225 \text{MIPS}$

（3） $\text{程序执行时间} = (45000 \times 1 + 75000 \times 2 + 8000 \times 4 + 1500 \times 2) / 400 = 575 \mu\text{s}$



例题解析

- ▶ 假定基准程序A在某计算机上的运行时间为100秒,其中90秒为CPU时间,其余为I/O时间。若CPU速度提高50%, I/O速度不变, 则运行基准程序A所耗费的时间是() (2012考研)
- A. 55 秒 B. 60 秒
C. 65 秒 D. 70 秒

答案: D

令原始主频为 f , 则原始时钟周期 $T = 1/f$ 。CPU速度提高50%, 则 $f' = 1.5f$, 故 $T' = 1/(1.5f) = 2/(3f)$

CPU执行时间 $t = \text{CPU时钟周期数} C \times \text{CPU时钟周期} T$,
CPU运行的时钟周期数 C 不变, 则 t 与 T 成正比。故

$$t' = (2/3)t = 90 \times (2/3) = 60(s)$$

基准程序A所耗费的时间 = $60 + 10 = 70(s)$



例题解析

- ▶ (2015 考研)
- ▶ 计算机硬件能够直接执行的是 ()
I、机器语言程序 II、汇编语言程序 III、硬件描述语言程序
- ▶ A、仅 I B、仅 I II
C、仅 I III D、I II III

答案：A

例题解析

- 某计算机主频为 1.2GHz，其指令分为 4 类，它们在基准程序中所占比例及 CPI 如下表所示。该机的 MIPS 数是（ ）（2013 考研）
- A. 100 B. 200
C. 400 D. 600

指令类型	所占比例	CPI
A	50%	2
B	20%	3
C	10%	4
D	20%	5

答案：C