



南京大學

NANJING UNIVERSITY

计算机组织结构

殷亚凤

智能软件与工程学院

苏州校区南雍楼东区225

yafeng@nju.edu.cn , <https://yafengnju.github.io/>



课程介绍

• 课程内容在计算机系统中位置

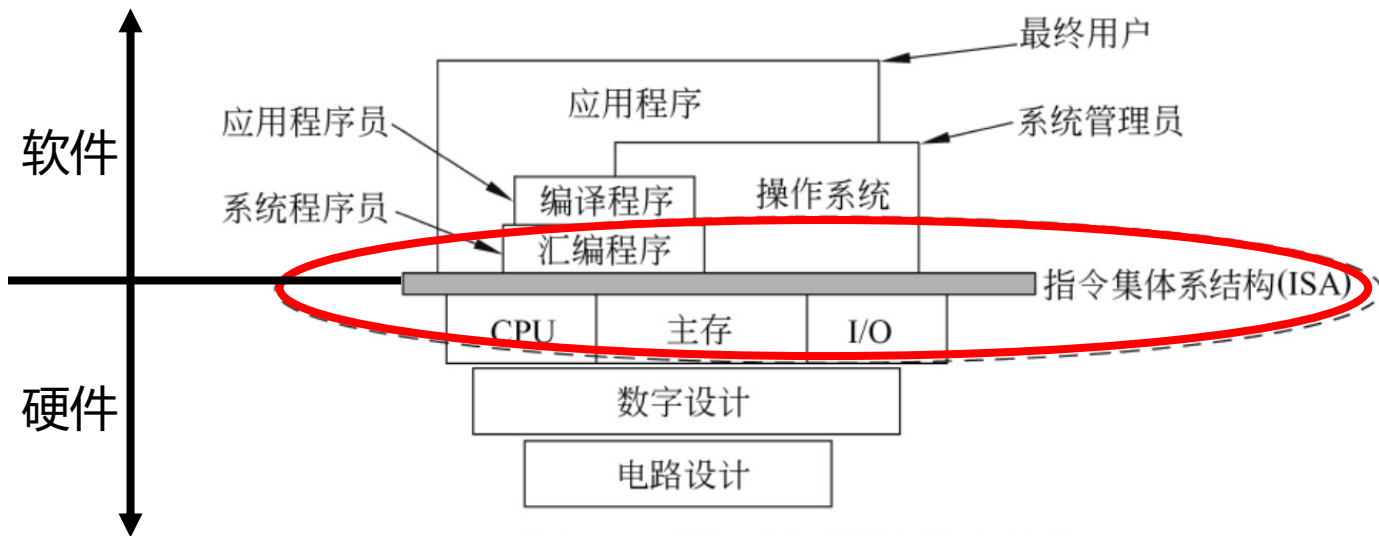


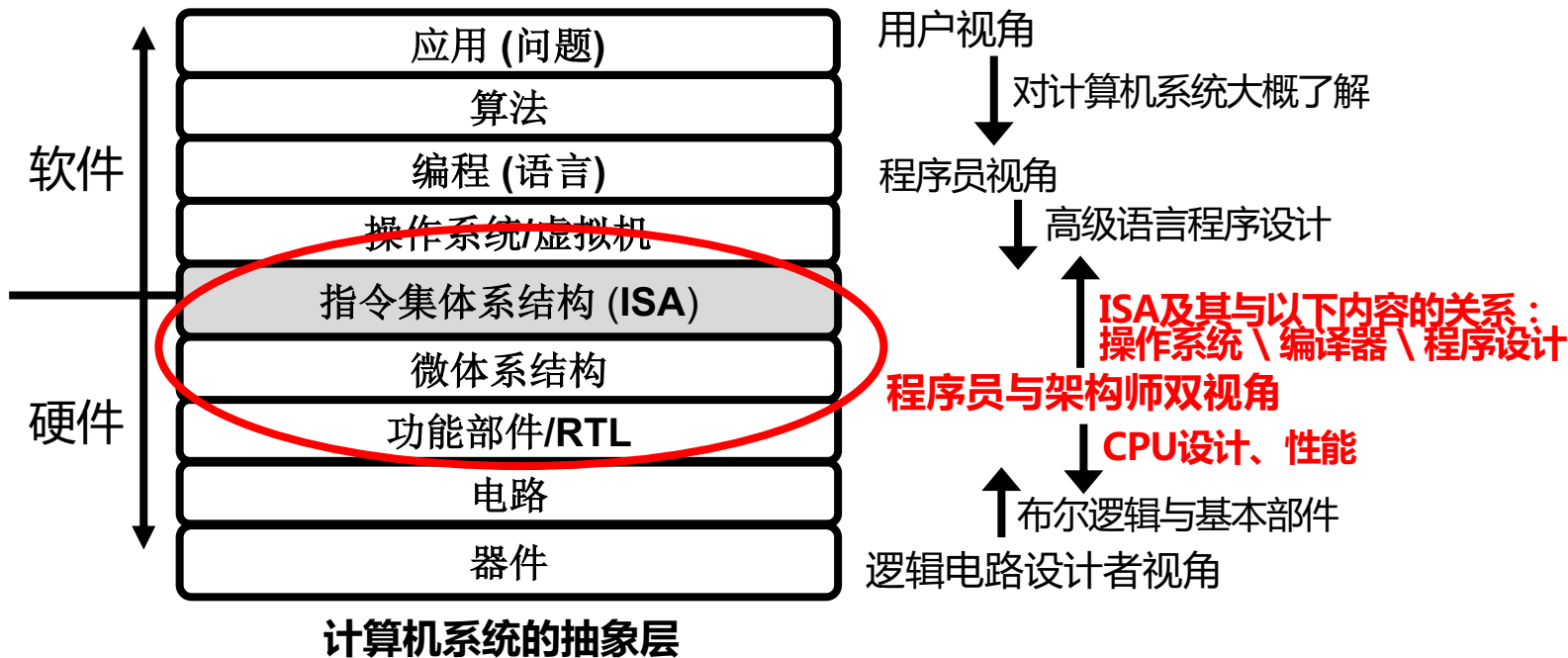
图 1.7 计算机系统的层次化结构

计算机系统的“基石”，计算机软件的“立足点”，属于计算机系统核心技术内容



课程介绍

本课程与其他课程的关系



计算机组织结构：
构建系统框架，以建立完整计算机系统概念

先行课：概论、程设、高级程设、数电设计等



南京大学
NANJING UNIVERSITY



教学目标

基本目标是什么？

- 指令集体系结构设计原理
- 计算机硬件设计原理
- 计算机整机概念的建立

要求掌握的重要知识和能力有哪些？

- 利用硬件知识提高调试程序的能力
(如：数据格式转换、大端/小端方式)
- 从硬件角度出发编制高效程序的能力
(如：Cache的局部性、函数调用)
- 操作系统和硬件如何分工、如何衔接？
(如：异常、中断)
- 如何从硬件角度出发进行编译优化？
(如：流水线调度)
- 为后续课程打下坚实基础
(如：操作系统、编译原理、体系结构、
微机原理、嵌入式技术等)





教学内容

- 计算机系统概述
- 数据的机器级表示
- 运算方法和运算部件
- 指令系统
- 中央处理器
- 指令流水线
- 存储器层次结构
- 系统互连及输入输出组织
- 并行处理系统





教学安排

- **课堂教学环节**
 - 总体框架、主干概念、结合例子、提问互动
- **其它环节**
 - 作业/习题报告（随机抽查上台讲解）
 - 编程实验报告（随机抽查上台讲解）
 - **没有期中考试**
 - 课后交流（网上讨论区、邮件、课间答疑）



课程主页

- 主页链接：<https://yafengnju.github.io/>

计算机组织结构

[\[Course Information\]](#) [\[Slides\]](#) [\[Assignments\]](#)

Course Information

To:	B.Sc. students of School of Intelligent Software and Engineering, Nanjing University
Teacher:	Yafeng Yin
Classroom:	Room 426, West of Nanyong Building
Class time:	9:00 - 12:00, Monday
Textbook:	袁春风主编, 唐杰、杨若瑜、李俊编著. 计算机组成与系统结构(第3版). 清华大学出版社, 2023.
Grading:	Final exam (60%) + Assignments (15%+15%)+ Lessons (10%)

Slides

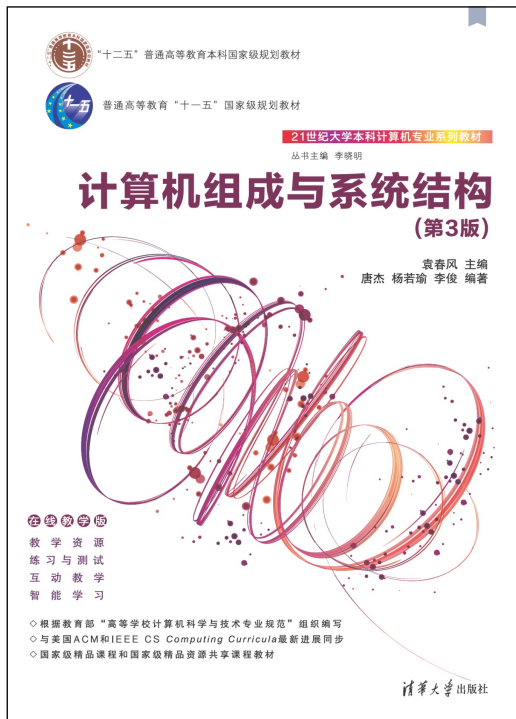
[1-计算机系统概述](#)

Assignments





教材



参考文献

- [1] PATT Y N, PATEL S J. 计算机系统该论. 梁阿磊, 蒋兴昌, 林凌, 译. 机械工业出版社.
- [2] STALLINGS W. 计算机组成与体系结构性能设计. 彭蔓蔓, 吴强, 任小西, 等译. 机械工业出版社.
- [3] 袁春风, 余子濠. 计算机系统基础. 机械工业出版社.
- [4] TANENBAUM A S. 现代操作系统. 陈向群, 马洪兵, 译. 机械工业出版社.

.....





考核形式

- **平时成绩（包括上课、课后作业和实验报告）：40%**
 - 随机抽查点名；
 - 约每节课均有课后作业，可选交其中5次（即取5次最高分）；
 - 3次实验作业，均需提交；
- **考试成绩（闭卷）：60%**

成绩=上课(10%)+课后作业(15%)+实验报告(15%)+期末考试(60%)



计算机系统概述

- 计算机的发展历程
- 计算机系统的基本组成
- 计算机系统层次结构
- 程序开发与执行过程
- 计算机系统性能评价





计算机的发展历程

- **第一代：真空管（电子管Vacuum Tube）** 1946 ~ 1957年



真空管

- **1946年诞生第1台电子计算机 ENIAC**

- 体积大，重30吨，有18000多个真空管，5000次加法/s
- 十进制表示/运算，存储器由20个累加器组成，每个累加器存10位十进制数，每一位由10个真空管表示。
- 采用手动编程，通过设置开关和插拔电缆来实现。

- **典型代表：冯·诺依曼机（Von Neumann Machine）**

- 45年冯·诺依曼提出 **“存储程序(Stored-program)”** 思想，并于46年开始设计“存储程序”计算机。
- **“存储程序”思想**：将事先编好的程序和原始数据送入主存中，然后启动执行。计算机能在不需操作人员干预下，自动完成逐条取出指令和执行指令的任务。



计算机的发展历程

- **第一代：真空管（电子管Vacuum Tube）1946~1957年**

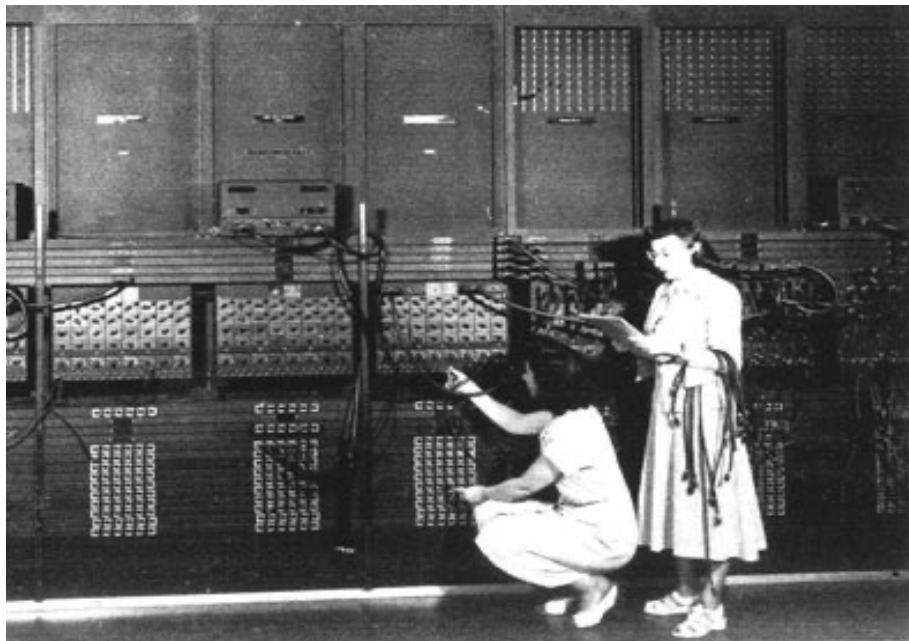


第1台实用的电子计算机 - ENIAC



计算机的发展历程

- **第一代：真空管（电子管Vacuum Tube）1946~1957年**



ENIAC：**不是**冯·诺依曼结构





计算机的发展历程

• 冯·诺依曼的故事



- 1944年，冯·诺依曼参加原子弹的研制工作，涉及到极为困难的计算。
- 1944年夏的一天，诺依曼巧遇美国弹道实验室的军方负责人戈尔斯坦，他正参与ENIAC的研制工作。
- 冯·诺依曼被戈尔斯坦介绍加入ENIAC研制组，1945年，在共同讨论的基础上，冯·诺依曼以“关于EDVAC的报告草案”为题，起草了长达101页的总结报告，发表了全新的“**存储程序通用电子计算机方案**”。
- 一向专搞理论研究的普林斯顿高等研究院批准让冯·诺依曼建造计算机，其依据就是这份报告。

Electronic Discrete Variable Automatic Computer



南京大學
NANJING UNIVERSITY



计算机的发展历程

• 冯·诺依曼结构的主要思想

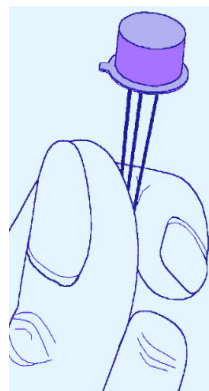
1. 计算机应由运算器、控制器、存储器、输入设备和输出设备五个基本部件组成。
2. 各基本部件的功能是：
 - **存储器**不仅能存放数据，而且也能存放指令，形式上两者没有区别，但计算机应能区分数据还是指令；
 - **控制器**应能自动执行指令；
 - **运算器**应能进行加/减/乘/除四种基本算术运算，并且也能进行一些逻辑运算和附加运算；
 - 操作人员可以通过**输入设备**、**输出设备**和主机进行通信。
3. 内部以**二进制表示**指令和数据。每条指令由操作码和地址码两部分组成。操作码指出操作类型，地址码指出操作数的地址。由一串指令组成程序。
4. 采用“**存储程序**”工作方式。



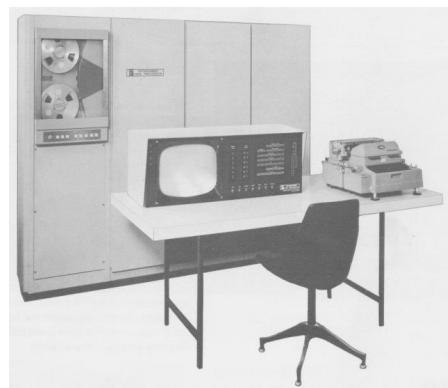


计算机的发展历程

- **第二代：晶体管 1958 ~ 64年**
 - **元器件**：逻辑元件采用晶体管，内存由磁芯构成，外存为磁鼓与磁带。
 - **特点**：变址，浮点运算，多路存储器，I/O处理机，中央交换结构(非总线结构)。
 - **软件**：使用高级语言，提供了系统软件。
 - **代表机种**：IBM 7094 (scientific)、1401 (business)和 DEC PDP-1



晶体管：
Transistor



DEC PDP-1





计算机的发展历程

- **第三代：SSI/MSI 1965 ~ 1971年**
 - **元器件**：逻辑元件与主存储器均由集成电路（IC）实现。
 - **特点**：微程序控制，Cache，虚拟存储器，流水线等。
 - **代表机种**：IBM 360和DEC PDP-8（大/巨型机与小型机同时发展）
 - 巨型机(Supercomputer)：Cray-1
 - 大型机(Mainframe)：IBM360系列
 - 小型机(Minicomputer)：DEC PDP-8



Cray-1



计算机的发展历程

• IBM System/360系列计算机

- IBM公司于1964年研制成功
- 引入“兼容机”（系列机）概念
 - 兼容机的特征：
 - 相同的或相似的指令集
 - 相同或相似的操作系统
 - 更高的速度
 - 更多的I/O端口数
 - 更大的内存容量
 - 更高的价格

低端机指令集是高端机的一个子集，称为“**向后兼容**”。原来机器上的程序可以不动而在新机器上运行，但性能不同。



IBM 360

问题1：引入“**兼容机**”有什么好处？

问题2：保持“**兼容**”的关键是什么？



计算机的发展历程

- **DEC公司的PDP-8机**

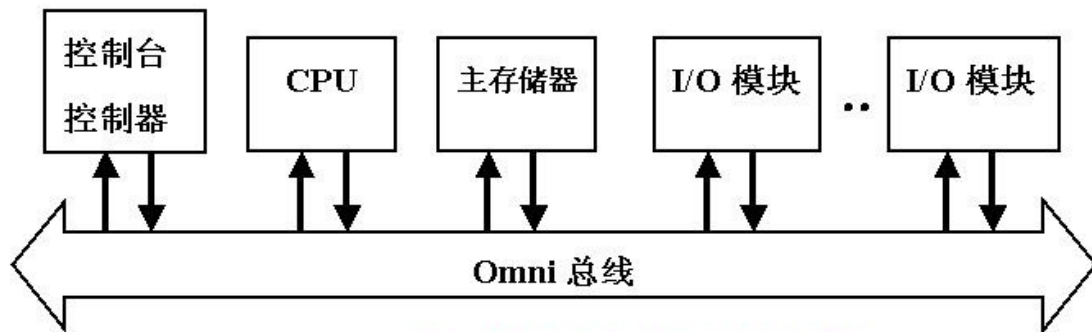
- 同在64年出现。与IBM 360相比，价格更低、更小巧，因而被称为小型机（Minicomputer）
- PDP-8 “创造了小型机概念，并使之成为数十亿美元的工业”，使DEC成为了最大的小型机制造商。
- **主要特点：首次采用总线结构。**
具有高度的灵活性，允许将模块插入总线以形成各种配置。



计算机的发展历程

- PDP-8/E计算机系统框图

Omnibus总线包含了96个独立的信号通道，用以传送控制、地址和数据信号。



PDP-8 计算机总线结构图

“总线结构”有什么好处？

可扩充性好（允许将新的符合标准的模块插入总线形成各种配置）、节省器件，体积小，价格便宜。



计算机的发展历程

• 第四代：大规模/超大规模集成电路 1972 ~ 至今

- 微处理器和半导体存储器技术发展迅猛，微型计算机出现。使计算机以办公设备和个人电脑的方式走向普通用户。

半导体存储器

- 70年Fairchild公司生产出第一个相对大容量半导体存储器
- 74年位价格低于磁芯的半导体存储器出现，并快速下跌
- 从70年起，存储密度呈4倍提高（几乎是每3年）

微处理器

- 微处理器芯片密度不断增加，使CPU中所有元件放在一块芯片上成为可能。71年开发出第一个微处理器芯片4004。

- **特点：**共享存储器，分布式存储器及大规模并行处理系统

以后几代（标准、意见不一）（注：有称第四代是超大规模集成电路VLSI，从80年代开始；也有称第四代是大规模集成电路LSI，从72年开始；有的又分成LSI时代和VLSI时代）



计算机系统概述

- 计算机的发展历程
- **计算机系统的基本组成**
- 计算机系统层次结构
- 程序开发与执行过程
- 计算机系统性能评价





计算机系统的基本组成

- 什么是计算机？
 - 计算机是一种能对数字化信息进行自动、高速算术和逻辑运算的处理装置。
- 计算机系统有硬件和软件两部分组成：
 - **硬件**：具体物理装置的总称（如芯片、板卡、外设、电缆等）
 - **软件**：运行在硬件上的程序和数据以及相关的文档。
 - 程序：指挥计算机如何操作的指令序列
 - 数据：指令操作的对象
- 计算机实现的所有任务都是通过执行一条一条指令完成的！





计算机系统的基本组成

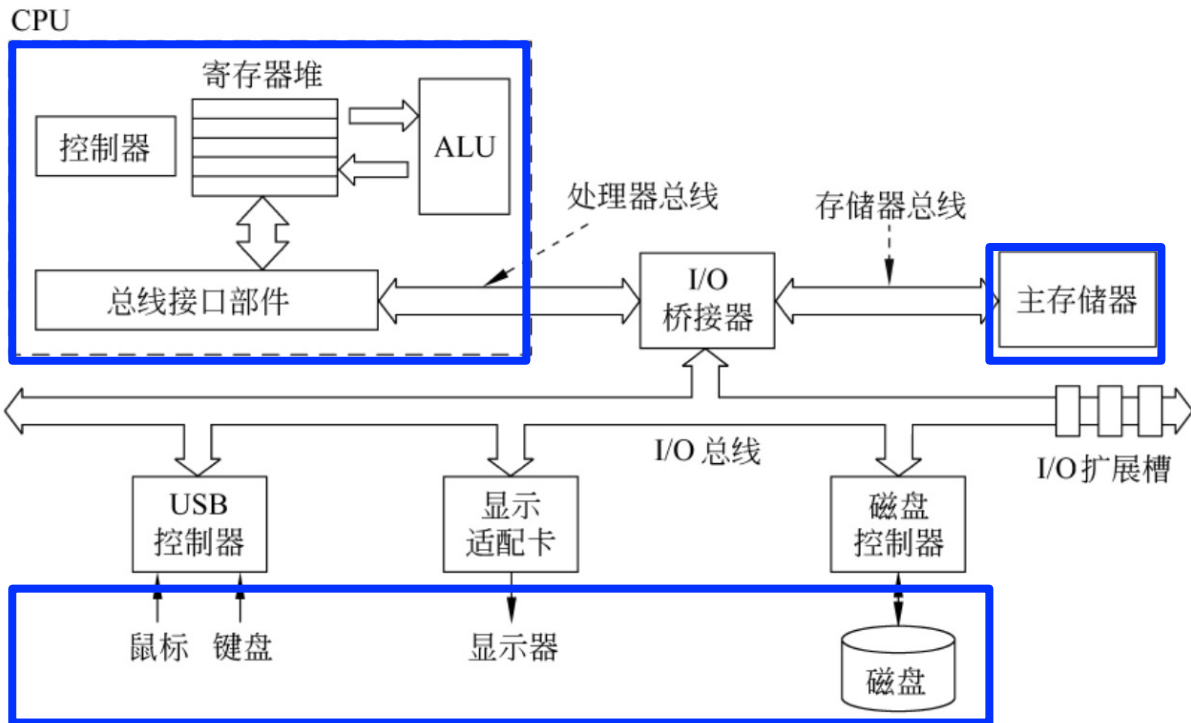
- **计算机系统的硬件组成**
- **冯·诺依曼结构**：计算机由运算器、控制器、存储器、输入设备、输出设备构成。
- **现代计算机**：把运算器、控制器和各类寄存器等互连在一个中央处理器中(CPU)。
 - **中央处理器**：核心部件，用于指令执行；包含数据通路和控制器。
 - **存储器**：分为内存和外存。
 - **外部设备和设备控制器**：外设通过设备控制器连接到主机上。
 - **总线**：传输信息的介质，用于在部件之间传输信息。





计算机系统的基本组成

计算机系统的硬件组成

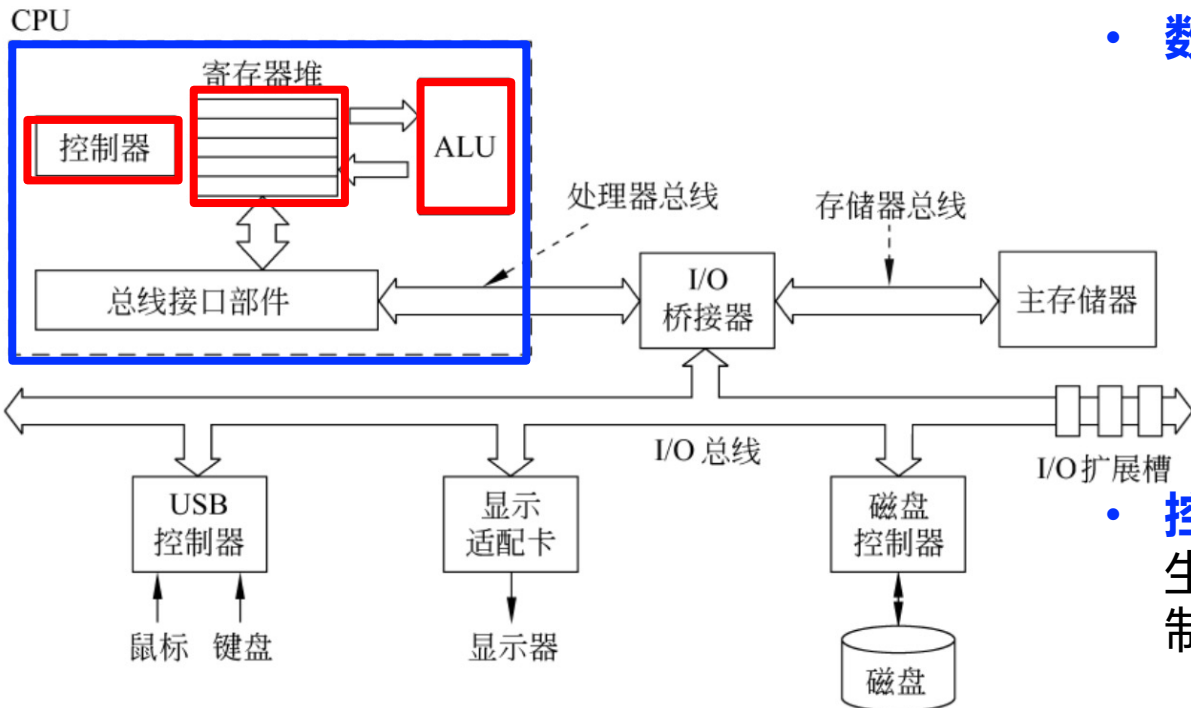


- **CPU**: 通过处理器总线、I/O桥接器等与主存储器和I/O设备交换信息；
- **主存**：通过存储器总线、I/O桥接器与CPU和I/O设备交换信息；
- **I/O设备**：通过各自的外设控制器连到I/O总线上。



计算机系统的基本组成

• 中央处理器 (CPU)



• 数据通路：

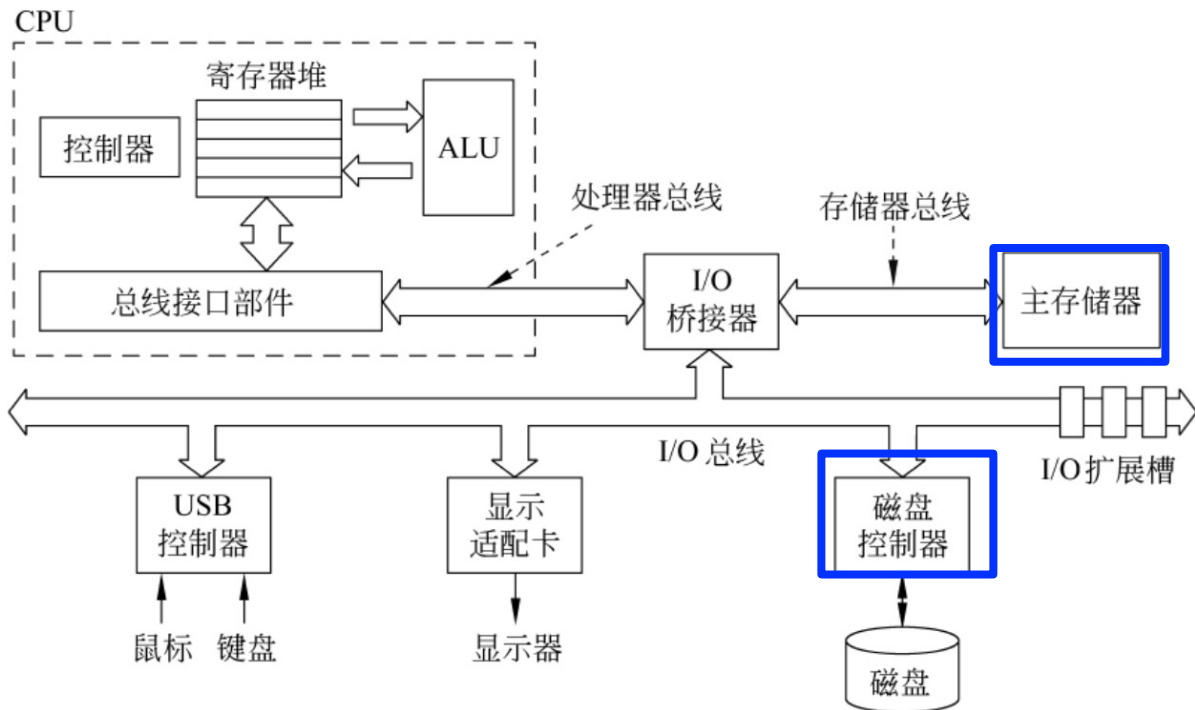
- **算术逻辑部件ALU**：执行算术和逻辑运算等操作；
- **通用寄存器**：暂存指令所用的操作数或执行结果。

- **控制器**：对指令进行译码，生成相应的控制信号，控制数据通路进行正确操作。



计算机系统的基本组成

• 存储器

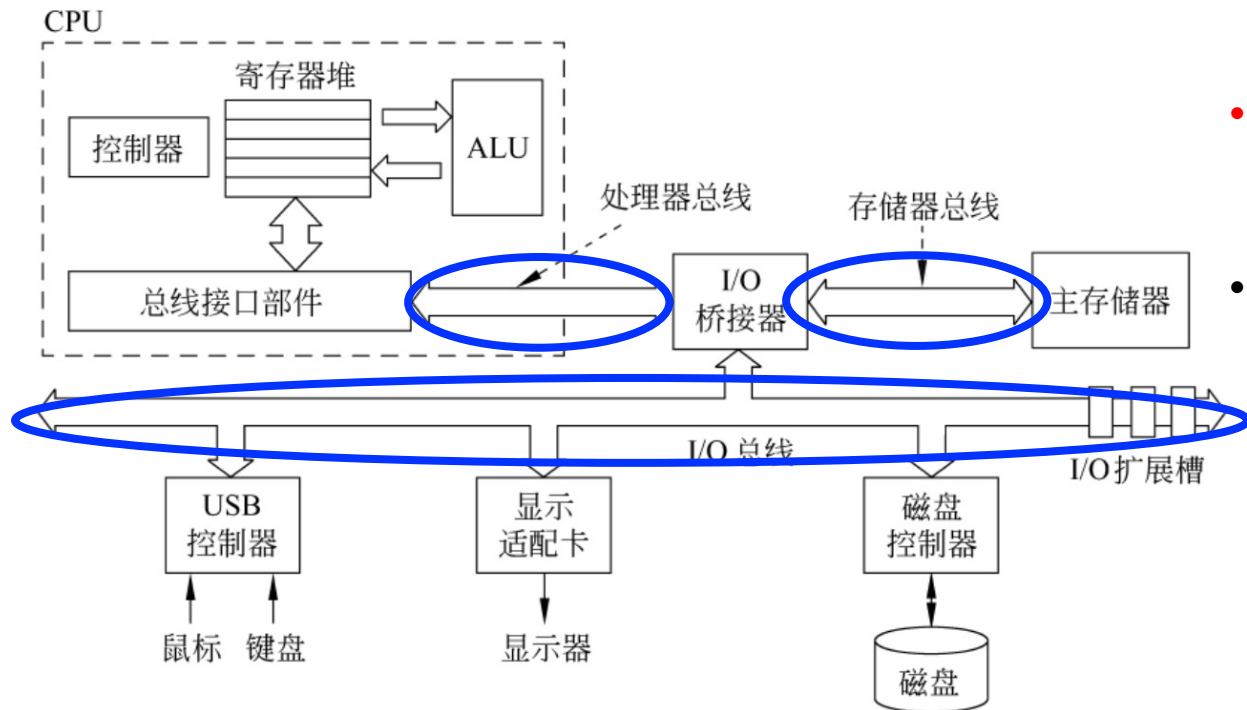


- **内存：**
 - **主存；**
 - **高速缓存；**
- **外存：**磁盘存储器、固态硬盘等。



计算机系统的基本组成

• 总线



- CPU、主存、I/O模块通过**总线**互连；
- 各类总线及总线接口部件、I/O桥接线、I/O扩展槽、I/O控制器、显示适配器等互连，用于数据传输和缓存任务。



计算机系统的基本组成

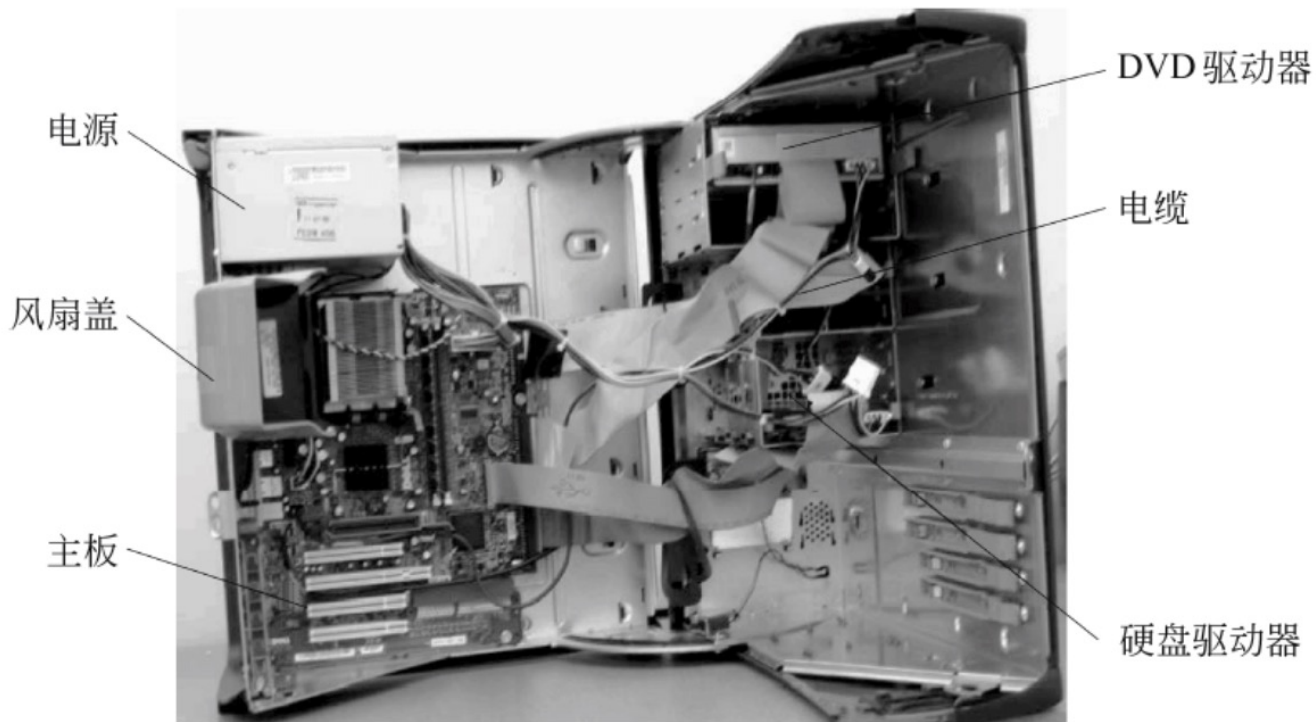


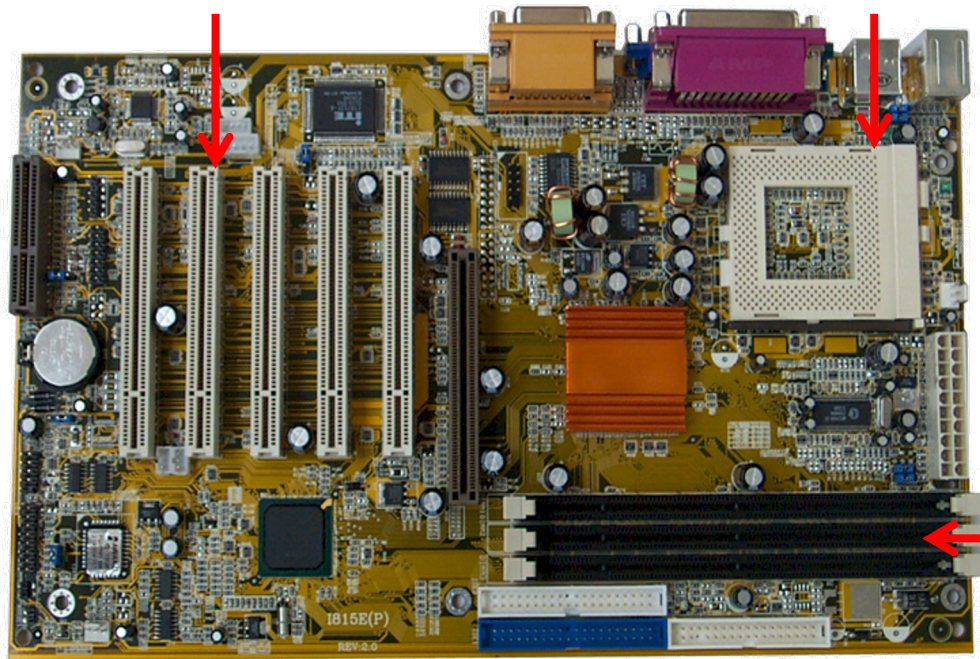
图 1.2 台式个人计算机机箱内的部件



计算机系统的基本组成

PCI总线插槽

CPU芯片插座



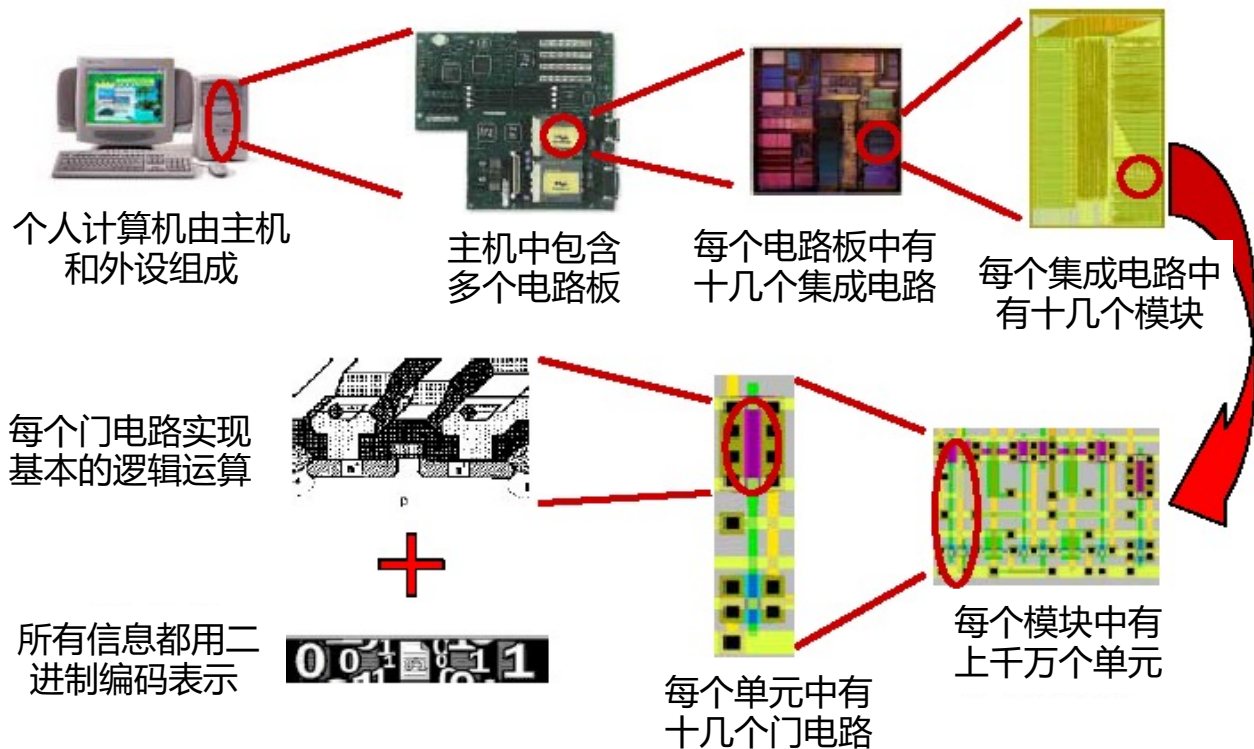
内存条插槽

台式个人计算机主板



计算机系统的基本组成

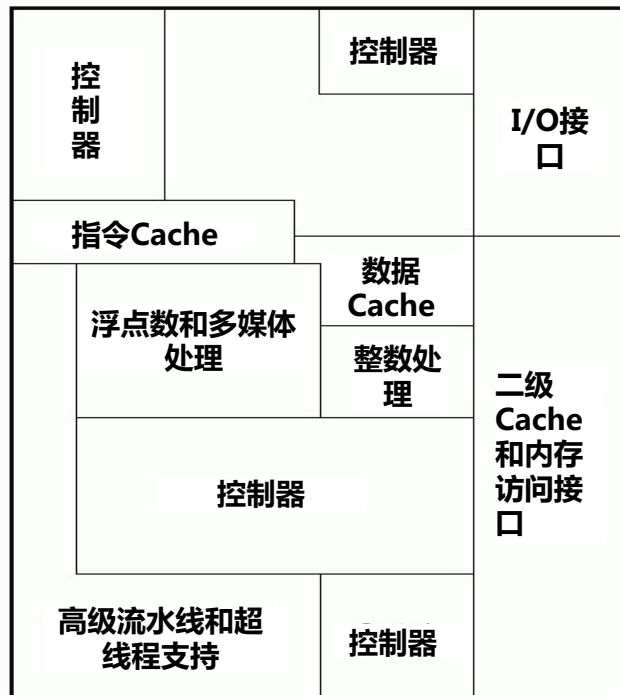
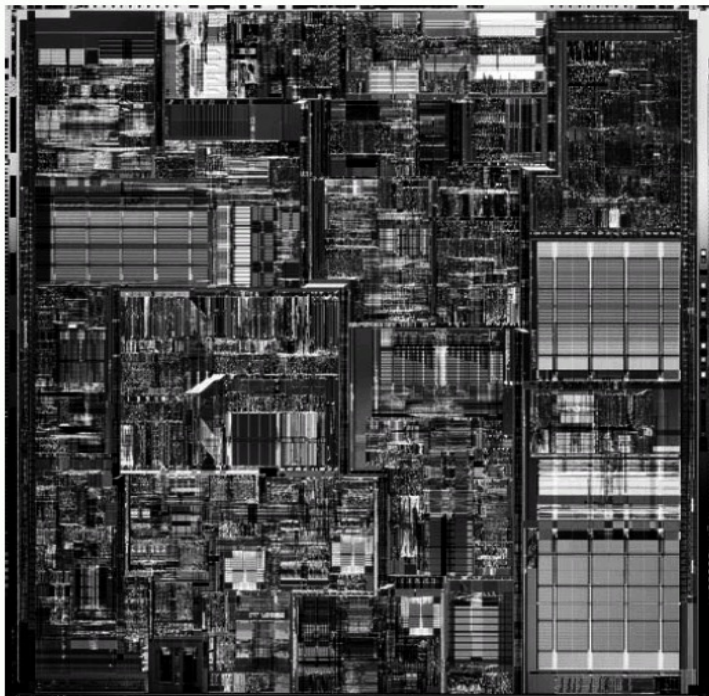
个人计算机的硬件结构解剖





计算机系统的基本组成

- Pentium 4处理器芯片的内部组成





计算机系统的基本组成

- **计算机软件**

- **应用软件**：专门为某种应用编写的程序（如电子邮件收发软件、视频播放软件、文字处理软件等）。
- **系统软件**：为有效、安全地**使用和管理计算机**以及为**开发和运行应用软件**而提供的各类软件；介于**计算机硬件与应用软件之间**。
 - 操作系统：
 - 语言处理系统：
 - 数据库管理系统：





计算机系统概述

- 计算机的发展历程
- 计算机系统的基本组成
- **计算机系统层次结构**
- 程序开发与执行过程
- 计算机系统性能评价





计算机系统层次结构

• 计算机系统抽象层的转换

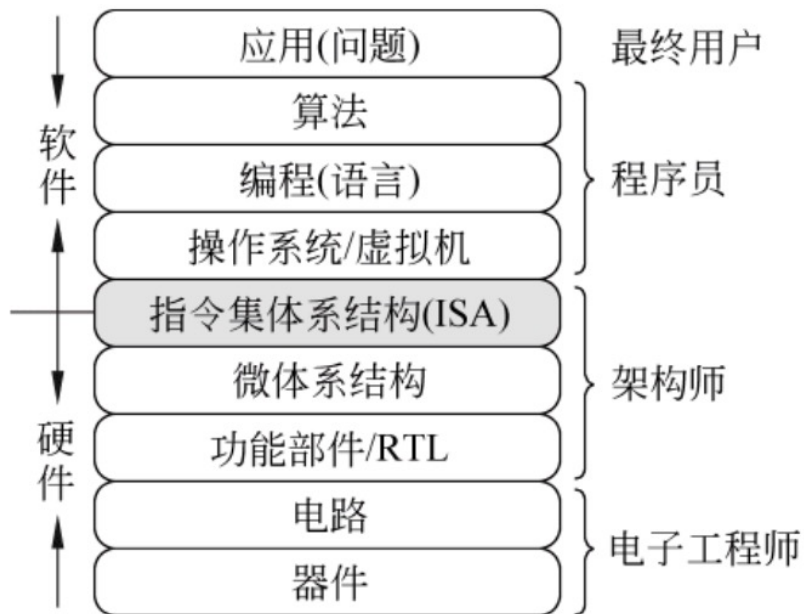


图 1.6 计算机系统抽象层及其转换

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！

程序执行：

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、指令集体系结构、微体系结构等。



• 算法和程序

- 首先将应用问题转换为算法描述；
- 将算法转换为用编程语言描述的程序。

• 编程语言

- **高级语言**：与底层计算机结构无关
- **低级语言（机器级语言）**：与计算机底层结构密切相关
 - **机器语言**：二进制序列
 - **汇编语言**：机器语言的符号（简短英文字符）表示语言





计算机系统层次结构

- **语言处理系统**：将高级语言程序转换成机器语言程序
 - **汇编程序**：将**汇编语言**源程序翻译成**机器语言**目标程序；
 - **解释程序**：将**源程序**中的语句逐条翻译成**机器指令**并立即执行；
 - **编译程序**：将**高级语言**源程序翻译成**汇编语言或机器语言**目标程序。
- **操作系统**
 - 所有语言处理系统都必须在操作系统提供的环境中运行；
 - 是对计算机底层结构和计算机硬件的一种抽象；
 - 构成了一台让程序员使用的虚拟机。



计算机系统层次结构

- **指令集体系结构（ISA）**：是硬件和软件之间接口的一个完整**定义**
 - 定义了一台计算机可以执行的所有指令集合；
 - 指令规定了计算机执行的操作，所处理的操作数存放的位置以及类型等；
 - **软件能感知**，属于软件可见部分。
- **微体系结构**：指令系统**具体实现**的组织
 - **软件不可感知**：如加法器是采用串行还是并行进位方式，软件不知；
 - 相同的ISA可能具有不同的微体系结构；
 - 由逻辑电路实现。



计算机系统层次结构

• 计算机系统的不同用户

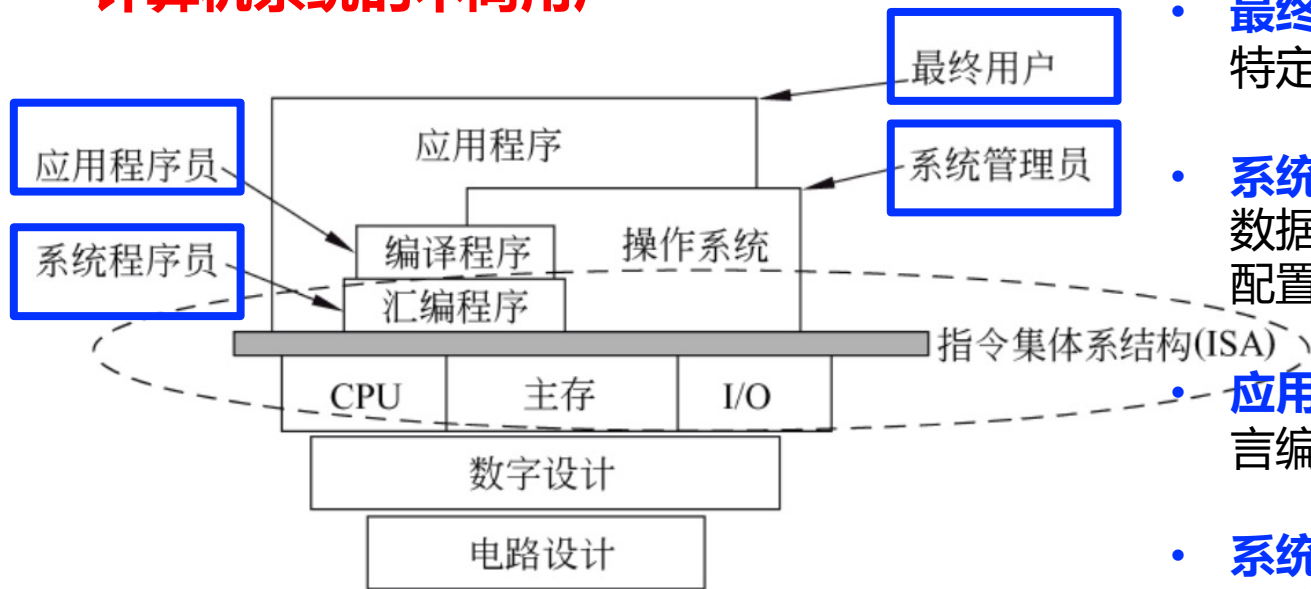


图 1.7 计算机系统的层次化结构

- **最终用户**：使用应用程序完成特定任务的计算机用户；
- **系统管理员**：利用操作系统、数据库管理系统等对系统进行配置、管理、维护的操作人员；
- **应用程序员**：使用高级编程语言编制应用程序的程序员；
- **系统程序员**：设计和开发操作系统、编译器、数据库管理程序等系统软件的程序员。



计算机系统概述

- 计算机的发展历程
- 计算机系统的基本组成
- 计算机系统层次结构
- **程序开发与执行过程**
- 计算机系统性能评价





程序开发与执行过程

从源程序到可执行程序

经典的 “hello.c” C-源程序

```

1 #include <stdio.h>
2
3 int main()
4 {
5     printf("hello, world\n");
6 }

```

程序的功能是：输出 “hello,world”

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

```

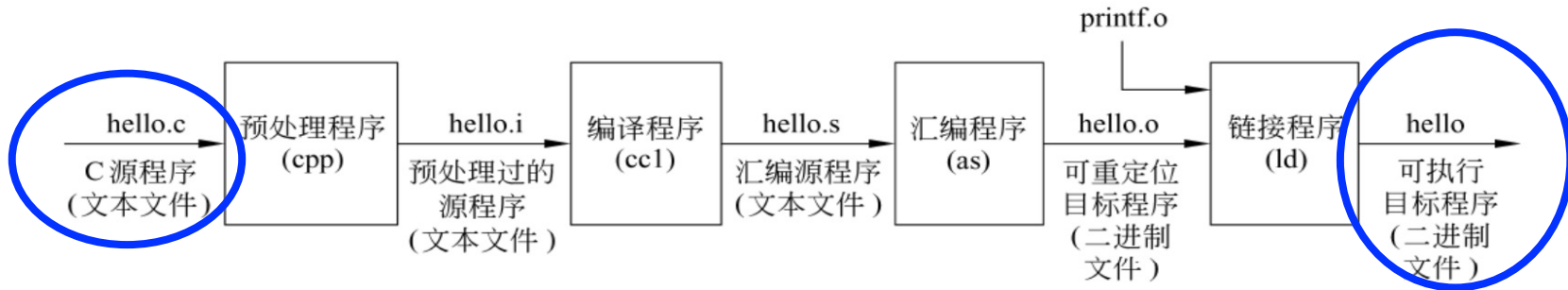


图 1.9 hello.c 源程序文件到可执行目标文件的转换过程



程序开发与执行过程

可执行程序的启动和执行

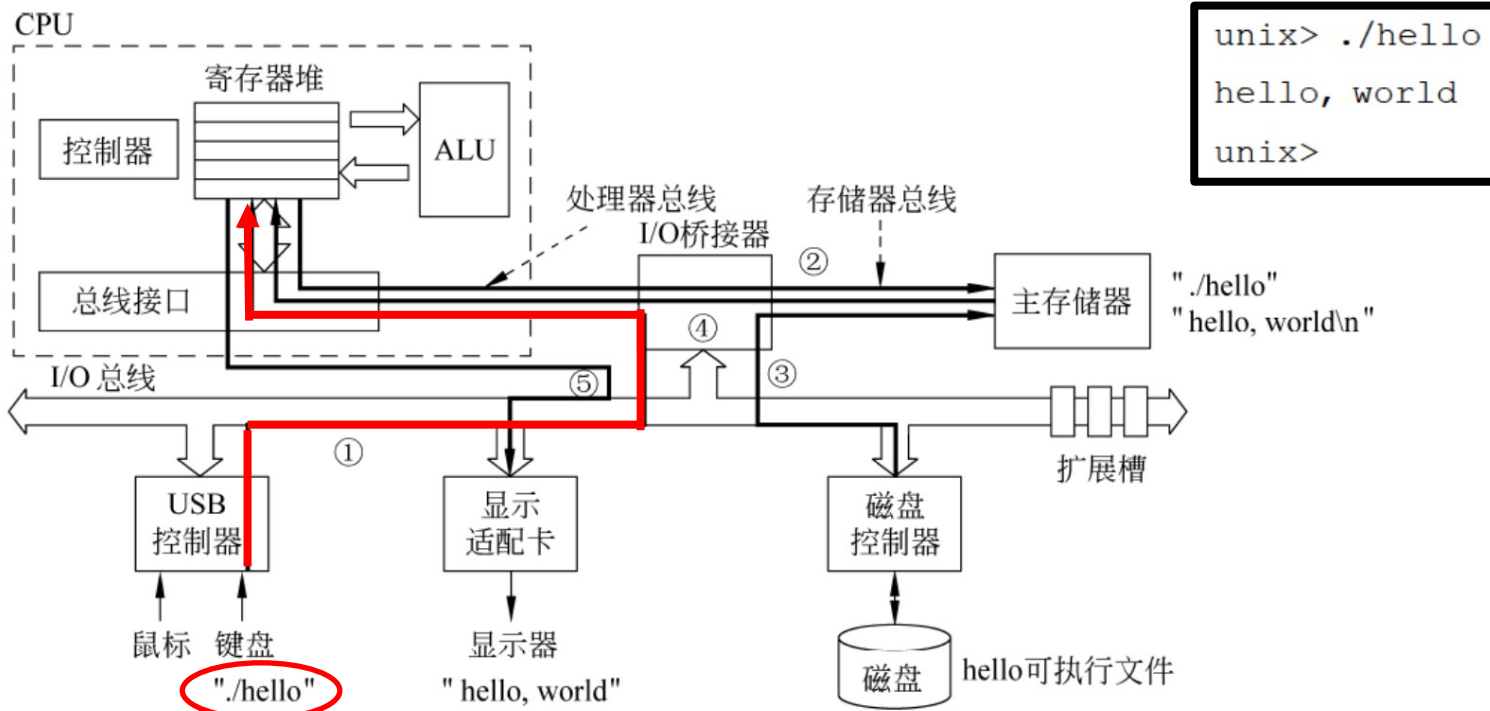


图 1.10 启动和执行 hello 程序的整个过程



程序开发与执行过程

可执行程序的启动和执行

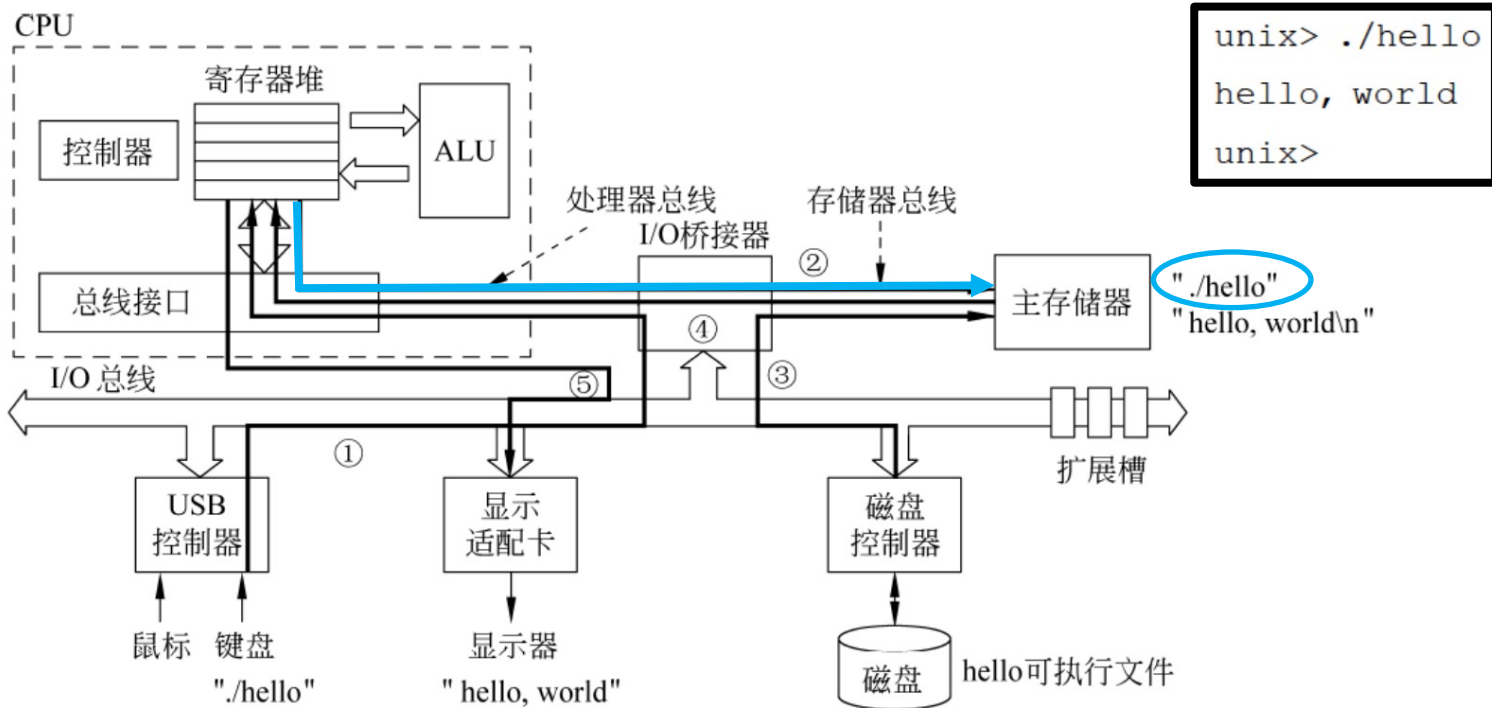


图 1.10 启动和执行 hello 程序的整个过程



程序开发与执行过程

可执行程序的启动和执行

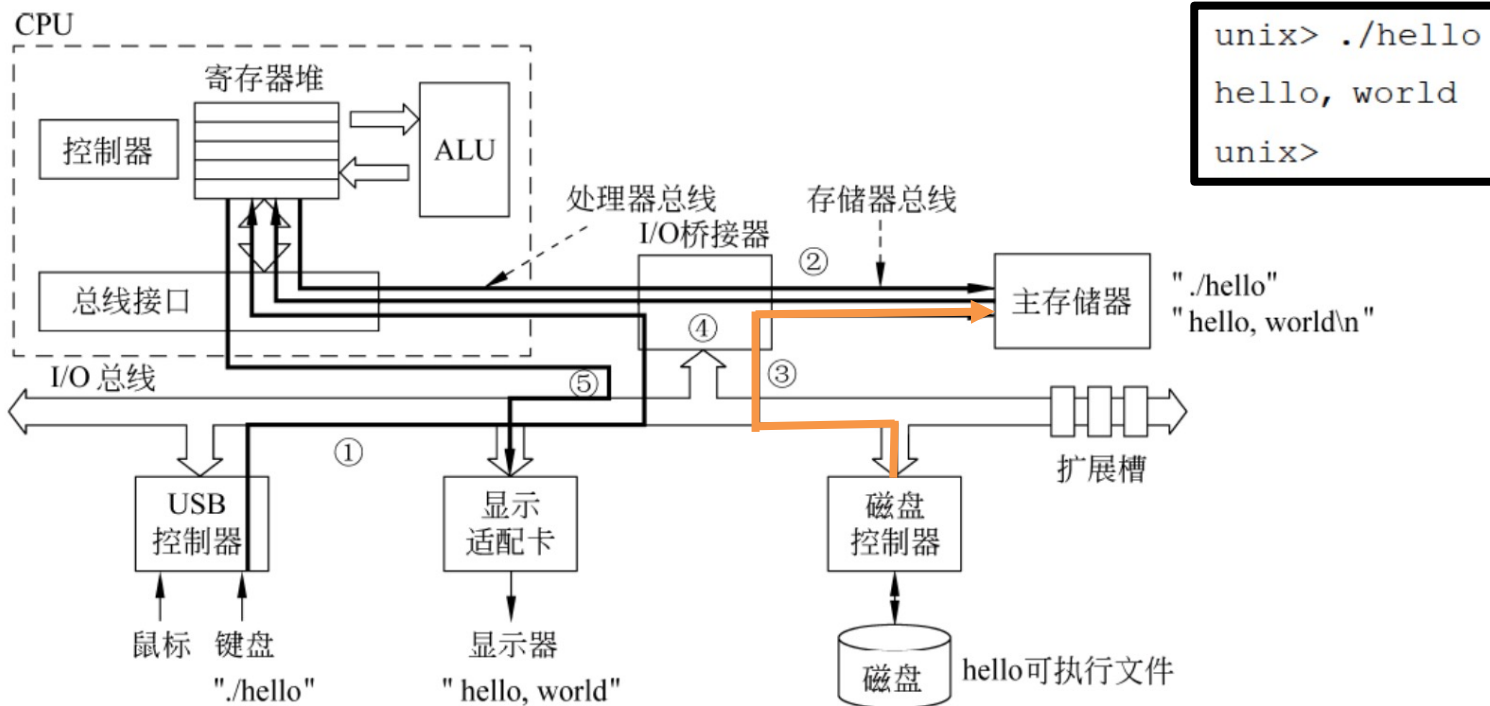


图 1.10 启动和执行 hello 程序的整个过程



程序开发与执行过程

可执行程序的启动和执行

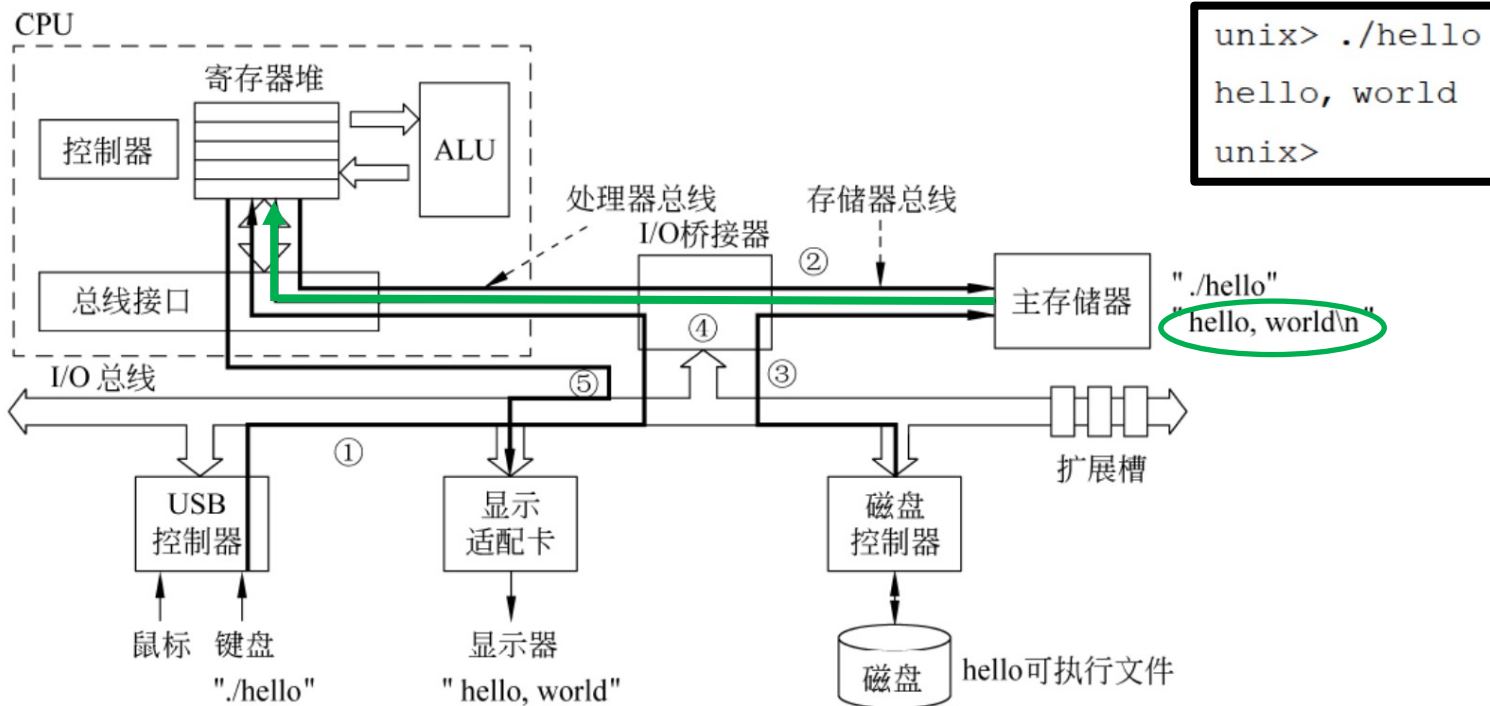


图 1.10 启动和执行 hello 程序的整个过程



程序开发与执行过程

可执行程序的启动和执行

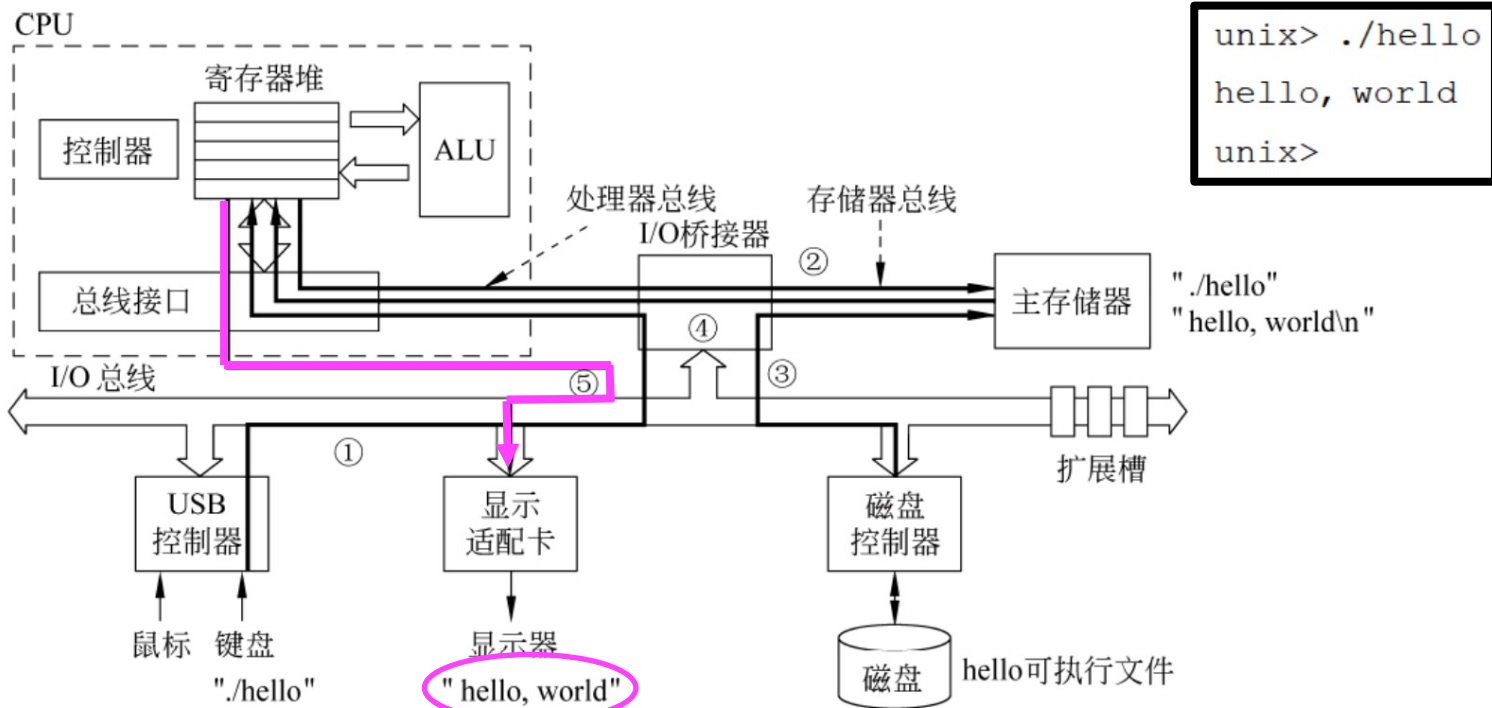


图 1.10 启动和执行 hello 程序的整个过程



程序开发与执行过程

程序与指令及控制信号的关系

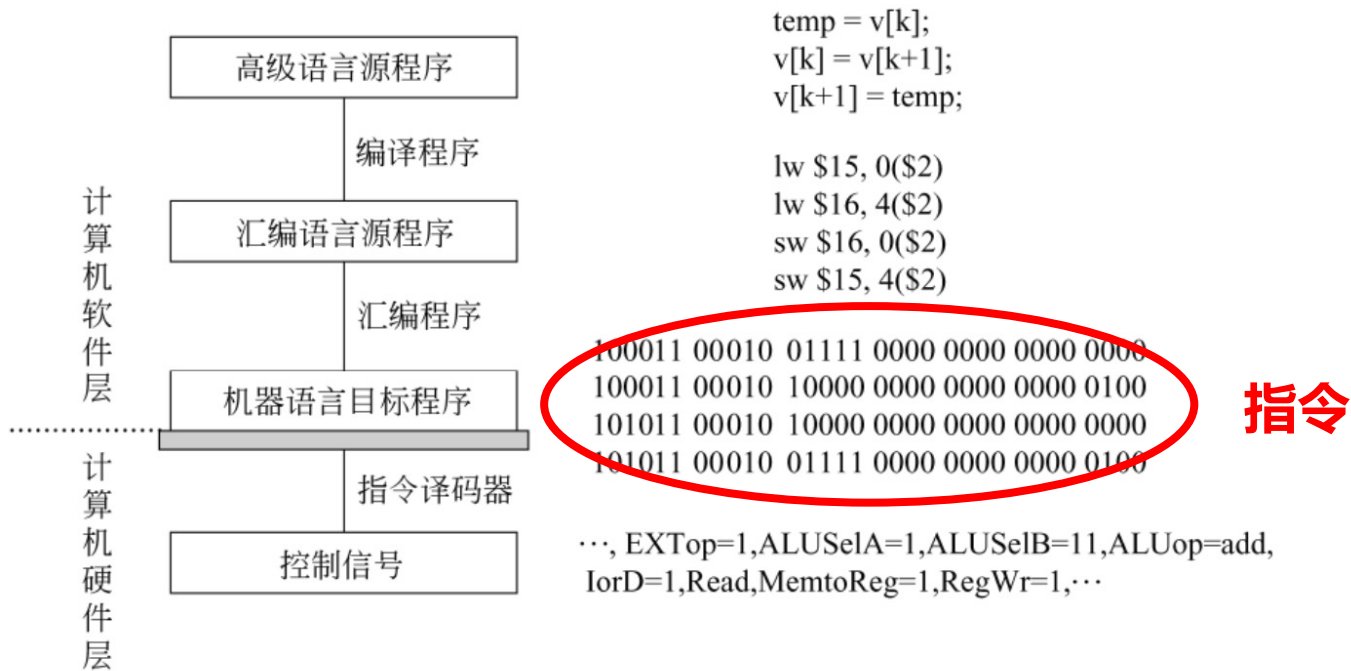


图 1.11 不同层次语言之间的等价转换



程序开发与执行过程

指令的执行过程

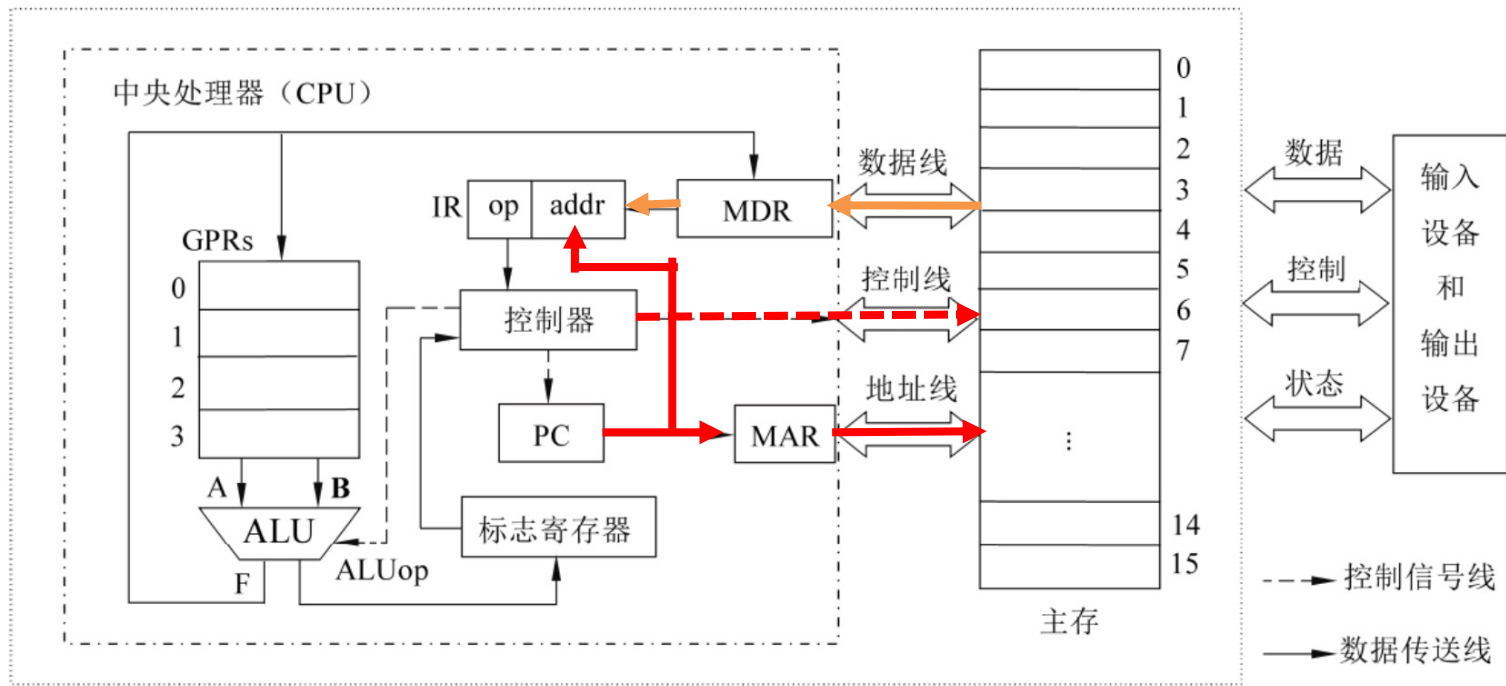


图 1.12 冯·诺依曼结构模型机



程序开发与执行过程

指令的执行过程

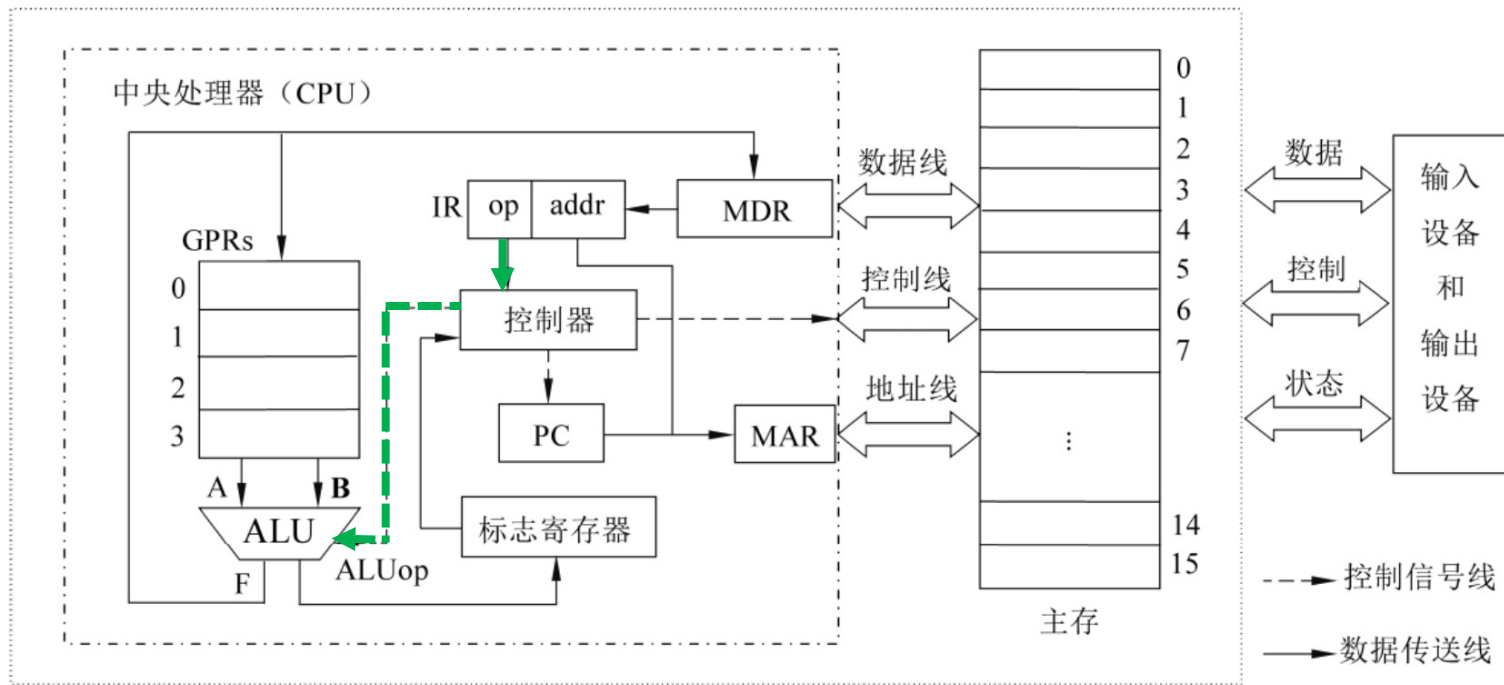


图 1.12 冯·诺依曼结构模型机



程序开发与执行过程

指令的执行过程

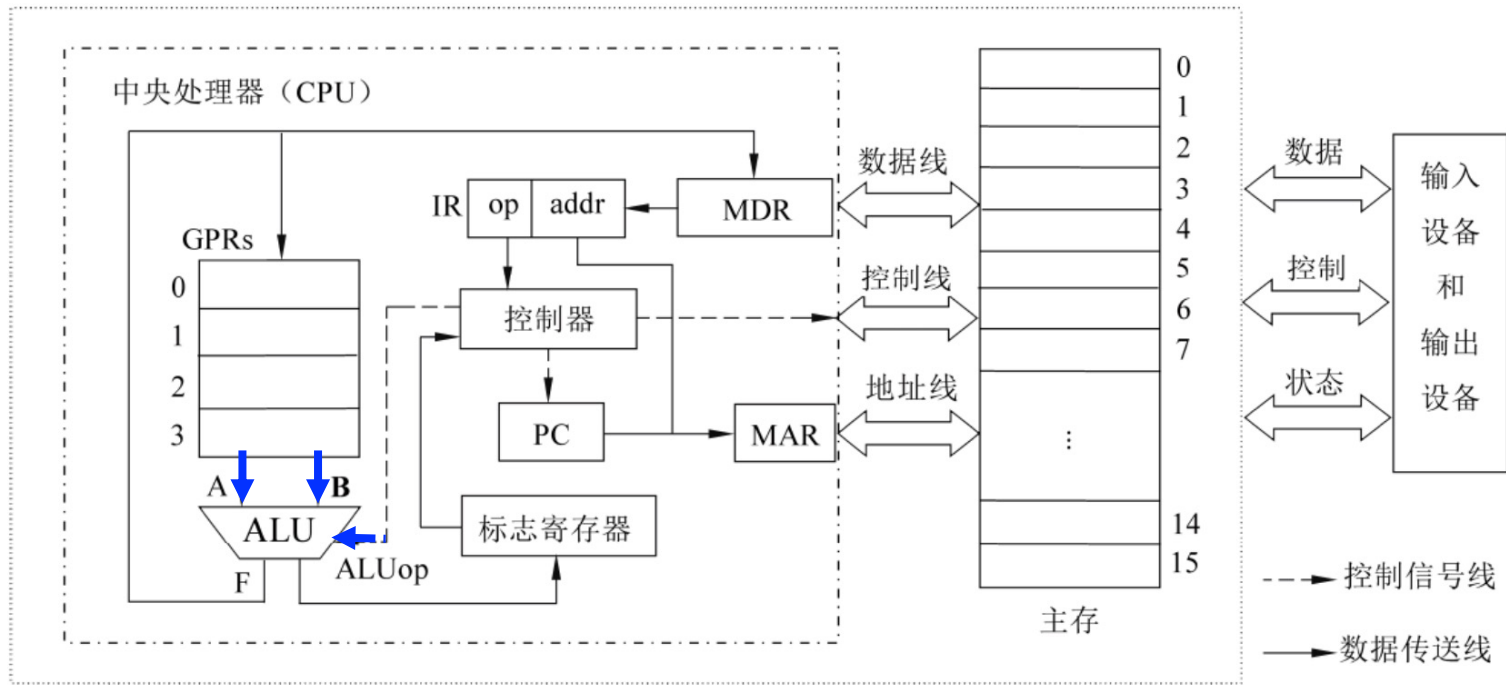


图 1.12 冯·诺依曼结构模型机



程序开发与执行过程

指令的执行过程

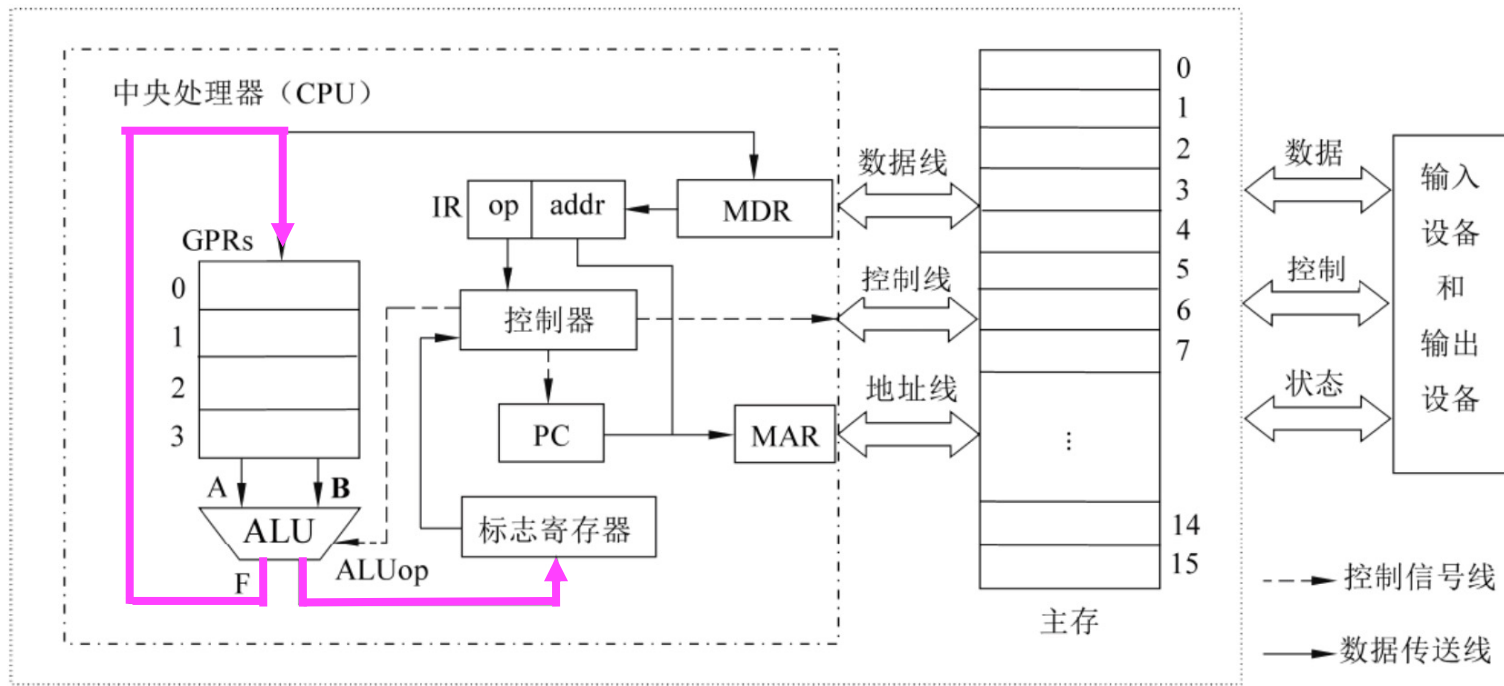


图 1.12 冯·诺依曼结构模型机



计算机系统概述

- 计算机的发展历程
- 计算机系统的基本组成
- 计算机系统层次结构
- 程序开发与执行过程
- **计算机系统性能评价**





计算机系统性能评价

• 计算机性能的定义

• 两个基本指标

• 作业提交开始到完成所有的时间

- 响应时间
- 执行时间
- 等待时间

• 单位时间所完成的工作量

- 吞吐率

• 不同应用场合用户关心的性能不同：

- 要求吞吐率高的场合，例如：
多媒体应用（音/视频播放要流畅）
- 要求响应时间短的场合：例如：
事务处理系统（存/取款的速度要快）
- 要求吞吐率高且响应时间短的场合：
ATM、文件服务器、Web服务器等



计算机系统性能评价

- **比较计算机的性能时，常常用执行时间来衡量**
 - 完成同样工作量所需时间最短的那台计算机就是性能最好的
 - 处理器时间往往被多个程序共享使用，因此用户感觉到的程序执行时间并不是程序真正的执行时间。
- **通常把用户感觉到的响应时间分成：**
 - **用户CPU时间**：用来运行用户代码的时间
 - **其他时间**：CPU运行操作系统、等待I/O操作完成、CPU花在其他用户程序的时间
- **系统性能和CPU性能不等价：**
 - **系统性能**：系统响应时间，与CPU外的其他部分也有关系
 - **CPU性能**：用户CPU时间
- **计算机系统的性能主要考虑CPU性能，即用户CPU时间！**





• 用户CPU时间涉及的概念和参数

- **时钟周期**：计算机产生的同步时钟定时信号(CPU主脉冲信号)的宽度
- **时钟频率**：主脉冲信号的时钟频率，CPU时钟周期的倒数
- **CPI**：执行一条指令所需的时钟周期数。对于一条指令，CPI是确定值；对于一个程序或机器，综合CPI是所有指令的平均时钟周期数。



计算机系统性能评价

- **CPU执行时间：** 用户 CPU 时间 = 程序总时钟周期数 ÷ 时钟频率
= 程序总时钟周期数 × 时钟周期

$$\text{程序总时钟周期数} = \text{程序总指令条数} \times \text{CPI}$$

$$\text{程序总时钟周期数} = \sum_{i=1}^n (\text{CPI}_i \times C_i)$$

CPI_i 、 F_i 是各指令的 CPI 和在程序中的出现频率

$$\text{CPI} = \sum_{i=1}^n (\text{CPI}_i \times F_i)$$

CPI_i 和 C_i 分别为第 i 类指令的 CPI 和指令条数

$$\text{用户 CPU 时间} = \text{程序总指令条数} \times \text{CPI} \times \text{时钟周期}$$

- 计算机的**性能之比**是用户**CPU时间之比**的**倒数**！



计算机系统性能评价

- **时钟周期、指令条数、CPI相互制约**，提高时钟频率不能保证速度同倍数提高。

例题：程序P在机器A上运行需10s，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6s. 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？

解: CPU时间A = 时钟周期数A / 时钟频率A

$$\text{时钟周期数A} = 10 \text{ sec} \times 400\text{MHz} = 4000\text{M个}$$

$$\text{时钟频率B} = \text{时钟周期数B} / \text{CPU时间B}$$

$$= 1.2 \times 4000\text{M} / 6 \text{ sec} = 800 \text{ MHz}$$

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！



计算机系统性能评价

- **时钟周期、指令条数、CPI相互制约**，指令条数最少的程序不一定执行最快

例 1.2 假设计算机 M 的指令集中包含 A、B、C 三类指令，其 CPI 分别为 1、2、4。程序 P 在 M 上被编译成两个不同的目标代码序列 P1 和 P2，P1 所含 A、B、C 三类指令的条数分别为 8、2、2，P2 所含 A、B、C 三类指令的条数分别为 2、5、3。请问：哪个代码序列执行速度快？它们的 CPI 分别是多少？

解：P1 和 P2 的指令条数分别为 12 和 10，P2 的指令条数更少。

P1 的时钟周期数为 $8 \times 1 + 2 \times 2 + 2 \times 4 = 20$ 。

P2 的时钟周期数为 $2 \times 1 + 5 \times 2 + 3 \times 4 = 24$ 。

因为两个指令代码序列在同一台机器上运行，所以时钟周期一样，故时钟数少的代码序列所用时间短，执行速度更快。显然，P1 比 P2 快。

评价计算机性能时，仅考虑单个因素是不全面的，**必须3个因素同时考虑。**



计算机系统性能评价

• 用指令执行速度进行性能评估

➤ **指令速度计量单位**：平均每秒钟执行多少百万条指令，（**MIPS**，Million Instructions Per Second）；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
- 峰值MIPS:（不实用）

**所以MIPS数不能说明性能的好坏
（下页中的例子可说明）**

➤ **浮点操作速度**：**MFLOPS**，Million Floating-point Operations Per Second

- 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！



计算机系统性能评价

• 用指令执行速度进行性能评估

假设我们为加载/存储机器构建了一个优化编译器。该编译器会丢弃50%的ALU指令。

(1) CPI是多少？

(2) 假设时钟周期时间为20纳秒（时钟频率为50兆赫兹）。优化代码与未优化代码的MIPS是多少？MIPS是否与执行时间的评级一致？

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		$50M / 1.57 = 31.8MIPS$	1.73
MIPS	31.8		$50M / 1.73 = 28.9MIPS$	28.9

→ 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，MIPS与执行时间评级不一致！。



计算机系统性能评价

• 用基准程序进行性能评估

- 基准测试程序是**专门用来进行性能评价的一组程序**；
- 在不同机器上运行相同的基准程序比较运行时间来测评性能；
- 基准程序能够反映计算机在运行实际负载时的性能。

• 基准程序的缺陷

- **现象**：基准程序的性能与某段短代码密切相关时，会被利用以得到**不当的性能评测**结果；
- **手段**：硬件系统设计人员或编译器开发者针对这些代码片段进行**特殊的优化**，使得执行这段代码的速度非常快。
 - ❑ 例1：Intel Pentium处理器运行SPECint时用了公司内部使用的特殊编译器，使其性能极高
 - ❑ 例2：矩阵乘法程序SPECmatrix300有99%的时间运行在一行语句上，有些厂商用特殊编译器优化该语句，使性能达VAX11/780的729.8倍！





计算机系统性能评价

• SPEC : 引用最广泛也是最全面的基准程序集

- 1988年, 5家公司 (Sun, MIPS, HP, Apollo, DEC) 联合提出了**SPEC (Systems Performance Evaluation Committee)**
- **SPEC给出了一组标准的测试程序、标准输入和测试报告。**它们是一些实际的程序, 包括 OS calls、I/O等。
- ❑ 版本89 : 10 programs=4 for integer+6 for FP, 用每个程序的执行时间求出一个综合性能指标。
- ❑ 版本92 : SPECInt92 (6 integer programs)和SPECfp92 (14 floating point programs)
 - ✓ 整数和浮点数单独提供衡量指标 : SPECInt92和SPECfp92
 - ✓ 增加 SPECbase: 禁止使用任何与程序有关的编译优化开关
- ❑ 版本95 : 8 int + 10fp
- ❑ 较新版本 : SPEC HPC96, SPEC JVM98, SPEC WEB99, SPEC OMP2001. SPEC CPU2000 (<http://www.spec.org>)



计算机系统性能评价

• 如何给出综合评价结果

问题：如果用一组基准程序在不同机器上测出了运行时间，那么如何综合评价机器的性能呢？

程序1：机器A上执行1秒，机器B上执行10秒；

程序2：机器A上执行1000秒，机器B上执行100秒。

- 对于程序1，A的速度是B的10倍；
- 对于程序2，B的速度是A的10倍。

这个结论无法比较A和B的好坏
必须用一个综合的值来表示！

执行时间总和是一个综合度量值，可以据此得出结论：A: 1001, B: 110, B的速度是A的 $1001/110=9.1$ 倍。

→ 实际上，可考虑每个程序在作业中的使用频度，即**加权平均**。



计算机系统性能评价

• 综合性能评价的方法

- 可用以下两种平均值来评价：
 - **Arithmetic mean(算术平均)**：求和后除n
 - **Geometric mean(几何平均)**：求积后开根号n
- 根据**算术平均**执行时间能得到**总平均执行时间**
- 根据几何平均执行时间不能得到程序总的执行时间
- 执行时间的规格化（测试机器相对于参考机器的性能）：
 - $\text{time on reference machine} \div \text{time on measured machine}$
- **平均规格化**执行时间不能用算术平均计算，而应该用**几何平均**
 - program A going from 2s to 1s **as important as** program B going from 2000s to 1000s. （**算术平均值不能反映这一点！**）

算术平均和几何平均各有长处，可灵活使用！





计算机系统性能评价

• Amdahl定律

- 阿姆达尔定律是计算机系统设计方面重要的定量原则之一
 - 基本思想：对系统中某部分（硬件或软件）进行更新所带来的系统性能改进程度，取决于该部分被使用的频率或其执行时间占总执行时间的比例。

$$\text{改进后的执行时间} = \frac{\text{改进部分执行时间}}{\text{改进部分的改进倍数}} + \text{未改进部分执行时间}$$

$$\text{整体改进倍数} = \frac{1}{\text{改进部分执行时间比例}/\text{改进部分的改进倍数} + \text{未改进部分执行时间比例}}$$

$p=1/(t/n + 1-t)$

- 若整数乘法器改进后可**加快10倍**，整数乘法指令在程序中占40%，则整体性能可改进多少倍？若占比达60%和90%，则整体性能分别能改进多少倍？

40% : $1/(0.4/10+0.6)=1.56$; 60% : $1/(0.6/10+0.4)=2.17$; 90% : $1/(0.9/10+0.1)=5.26$



计算机系统性能评价

• Amdahl定律

例：某程序在某台计算机上运行所需时间是100秒，其中，80秒用来执行乘法操作。要使该程序的性能是原来的5倍，若不改进其他部件而仅改进乘法部件，则乘法部件的速度应该提高到原来的多少倍？

解：根据公式 $p=1/(t/n + 1-t)$ 知：

$$5=1/(0.8/n+0.2) , 0.8/n+0.2 = 1/5 = 0.2$$

要使上述公式满足，则**必须 $0.8/n=0$ ，即 $n \rightarrow \infty$**

→ 也就是说，即使乘法运算时间占80%，也**不可能**通过对乘法部件的改进，使整体性能提高到原来的5倍。

→ 当乘法运算时间占比 **$\leq 80\%$** ，则无论如何对乘法部件进行改进，都不能使整体性能提高到原来的5倍。



课程习题（作业）——截止日期：9月16日晚23:59

- **课本21-22页**：第2、8、9、10题
- 提交方式：<https://selearning.nju.edu.cn/>（教学支持系统）

教学支持系统

课程

- ▾ 2024 Fall
 - 本科生一年级
 - 本科生二年级
 - 本科生三年级
 - 本科生四年级
 - 研究生一年级
 - 智能软件与工程学院

计算机组织结构-智软院

教师: 殷亚凤

课后习题

第1章-计算机系统概述-课后习题

第1章-计算机系统概述-课后习题

课本21-22页：第2、8、9、10题

- 命名：学号+姓名+第*章。
- 若提交遇到问题请及时发邮件或在下一次上课时反馈。



课程习题（作业）——截止日期：9月16日晚23:59

2. 简单回答下列问题。

- (1) 冯·诺依曼计算机由哪几部分组成？各部分的功能是什么？采用什么工作方式？
- (2) 摩尔定律的主要内容是什么？
- (3) 计算机系统的层次结构如何划分？计算机系统的用户可分为哪几类？每类用户工作在哪个层次？
- (4) 程序的 CPI 与哪些因素有关？
- (5) 为什么说性能指标 MIPS 不能很好地反映计算机的性能？

8. 假设机器 M 的时钟频率为 4GHz, 程序 P 在 M 上的指令条数为 8×10^9 , 其 CPI 为 1.25, 则 P 在 M 上的执行时间是多少？若在机器 M 上从程序 P 开始启动到执行结束所需的时间是 4s, 则 P 占用的 CPU 时间的百分比是多少？





课程习题（作业）——截止日期：9月16日晚23:59

9. 假定编译器对某段高级语言程序编译生成两种不同的指令序列 S1 和 S2, 在时钟频率为 500MHz 的机器 M 上运行, 目标指令序列中用到的指令类型有 A、B、C 和 D 四类。每类指令在 M 上的 CPI 和两个指令序列所用的各类指令条数如下表所示。

	A	B	C	D
各指令的 CPI	1	2	3	4
S1 的指令条数	5	2	2	1
S2 的指令条数	1	1	1	5

请问：S1 和 S2 各有多少条指令？CPI 各为多少？所含的时钟周期数各为多少？执行时间各为多少？

10. 假定机器 M 的时钟频率为 1.2GHz, 程序 P 在机器 M 上的执行时间为 12s。对 P 优化时, 将其所有的乘 4 指令都换成了一条左移两位的指令, 得到优化后的程序 P'。已知在 M 上乘法指令的 CPI 为 5, 左移指令的 CPI 为 2, P 的执行时间是 P' 执行时间的 1.2 倍, 则 P 中有多少条乘法指令被替换成了左移指令被执行？



提问

Q & A

殷亚凤

智能软件与工程学院

苏州校区南雍楼东区225

yafeng@nju.edu.cn , <https://yafengnju.github.io/>



南京大學
NANJING UNIVERSITY