# 计算机系统概述

## 一、计算机的发展历程

### 第一代：真空管（电子管Vacuum Tube）

> *典型代表：冯·诺依曼机（Von Neumann Machine）*

> *“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。*

- 冯·诺依曼结构的主要思想
    - 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。
    - 各基本部件功能：
      • 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。
- 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。
- 采用“存储程序”工作方式。

### 第二代：晶体管

- 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
- 代表机种：.....

### 第三代：SSI/MSI

- 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：.....

> *“兼容机”(系列机)概念：“向后兼容”*

> *PDP-8机：首次采用总线结构：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*

### 第四代：大规模/超大规模集成电路

- 摩尔定律：由于硅技术的不断改进，每18个月。集成度将翻一番，速度将提高一倍，而其价格将降低一半

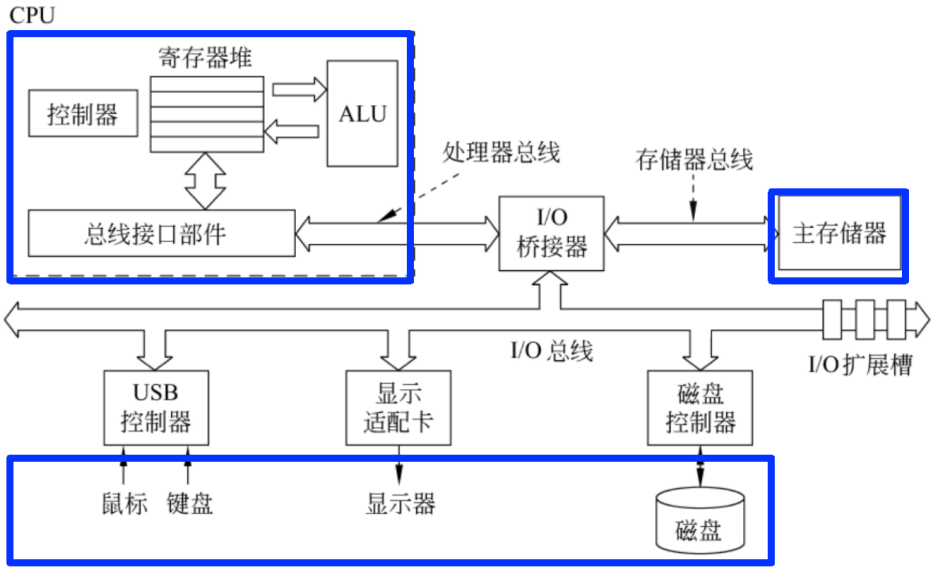
## 二、计算机系统的基本组成

计算机系统有硬件和软件两部分组成

### 硬件

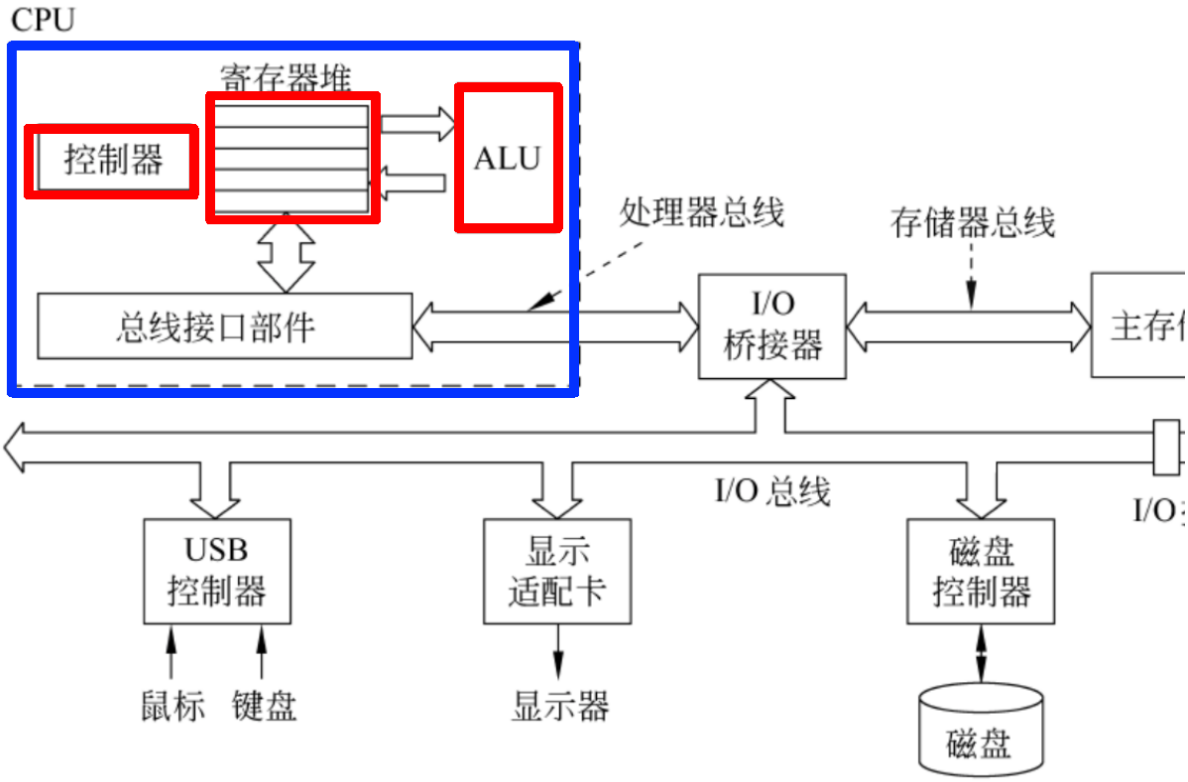
具体物理装置的总称（如芯片、板卡、外设、电缆等）

- 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




- **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。

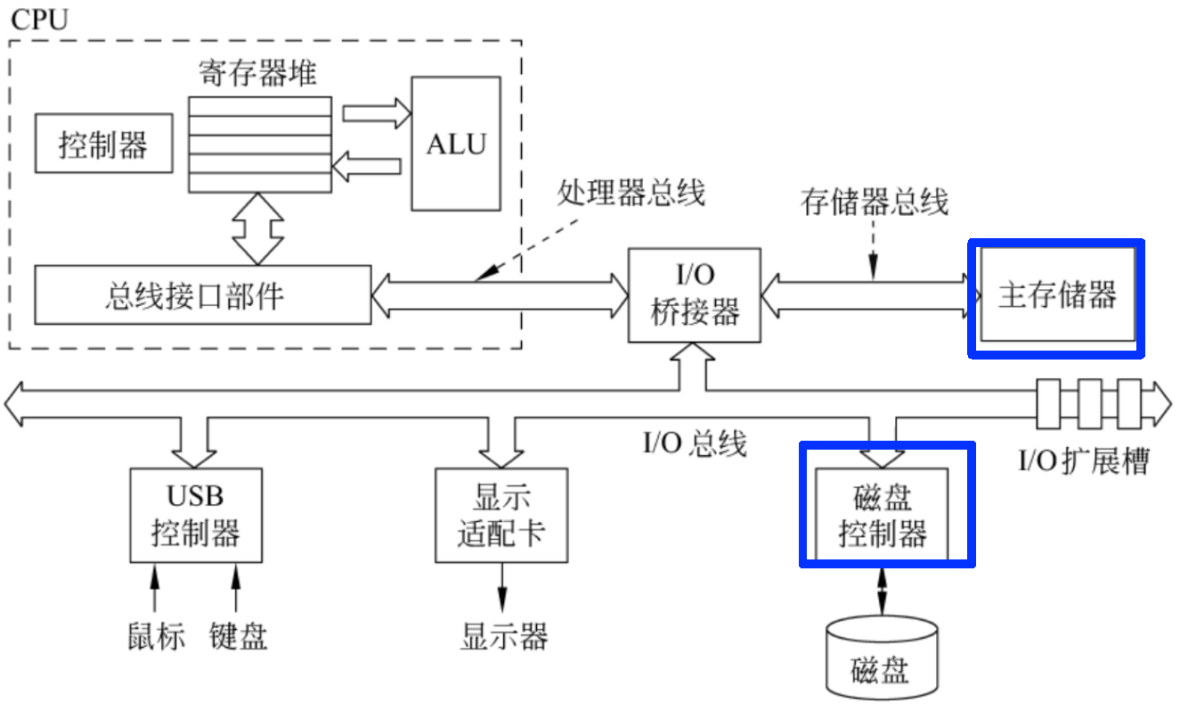
1. 中央处理器（**CPU**）
    a. 数据通路
        i. **算术逻辑部件（ALU）**：执行算术和逻辑运算等操作
        ii. **通用寄存器**：暂存指令所用的操作数或执行结果
    b. 控制器：对指令进行译码，生成相应的控制信号，控制数据通路进行正确操作




2. 存储器

a. 内存：主存+高速缓存

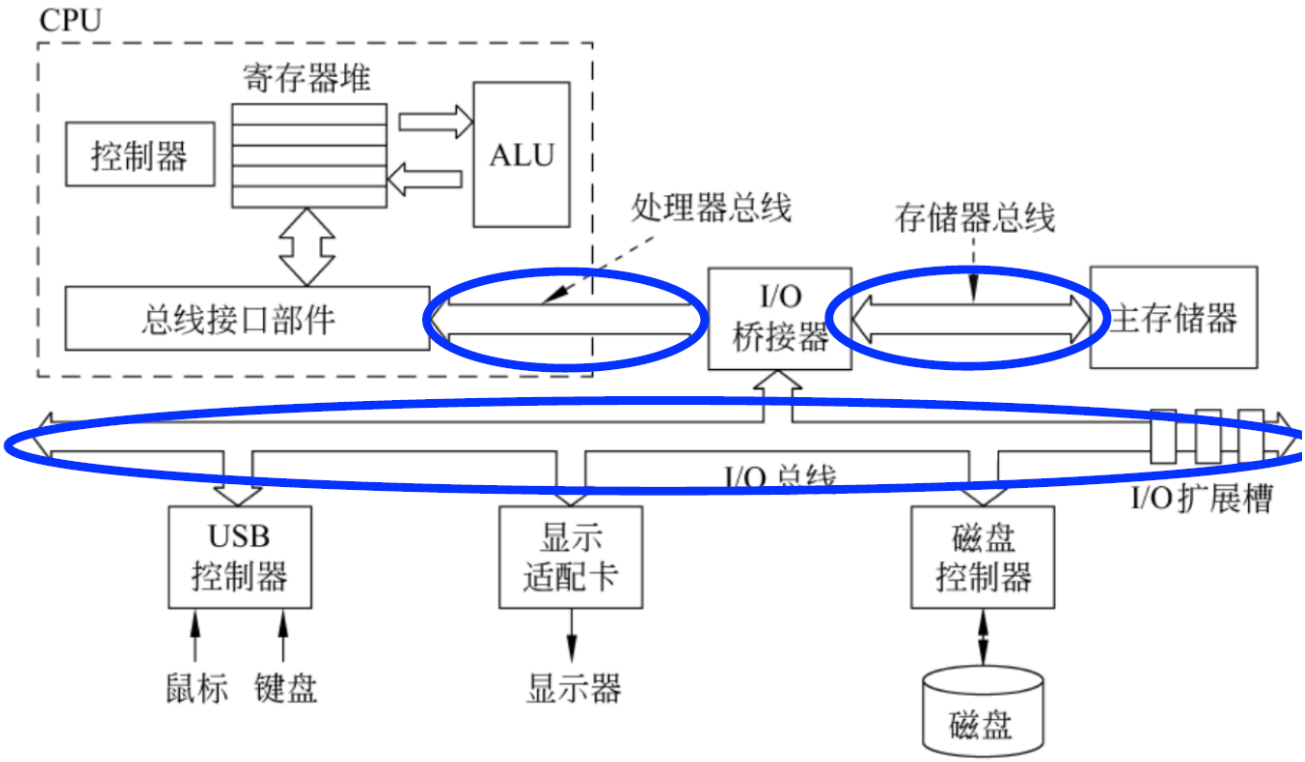
b. 外存：磁盘存储器、固态硬盘等




3. 总线

a. CPU、主存、I/O模块通过主线互连

b. 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




4. 外设和设备控制器

a. 外设：I/O设备

i. 机械部分：外设本身

ii. 电子部分：控制外设的设备控制器

b. 设备控制器（I/O控制器）：键盘接口、显卡.....

## 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

> *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*

---

# 三、计算机系统层次结构

## 1. 计算机系统抽象层的转换

> *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。*

1. **算法和程序**
    a. 首先将应用问题转换为算法描述
    b. 将算法转换为用编程语言描述的程序
2. **编程语言**
    a. **高级语言**：与底层计算机结构无关
    b. **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
3. **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
4. **操作系统**
5. **指令集体系结构（ISA）**：硬件和软件之间接口的一个完整定义
    - 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
6. **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

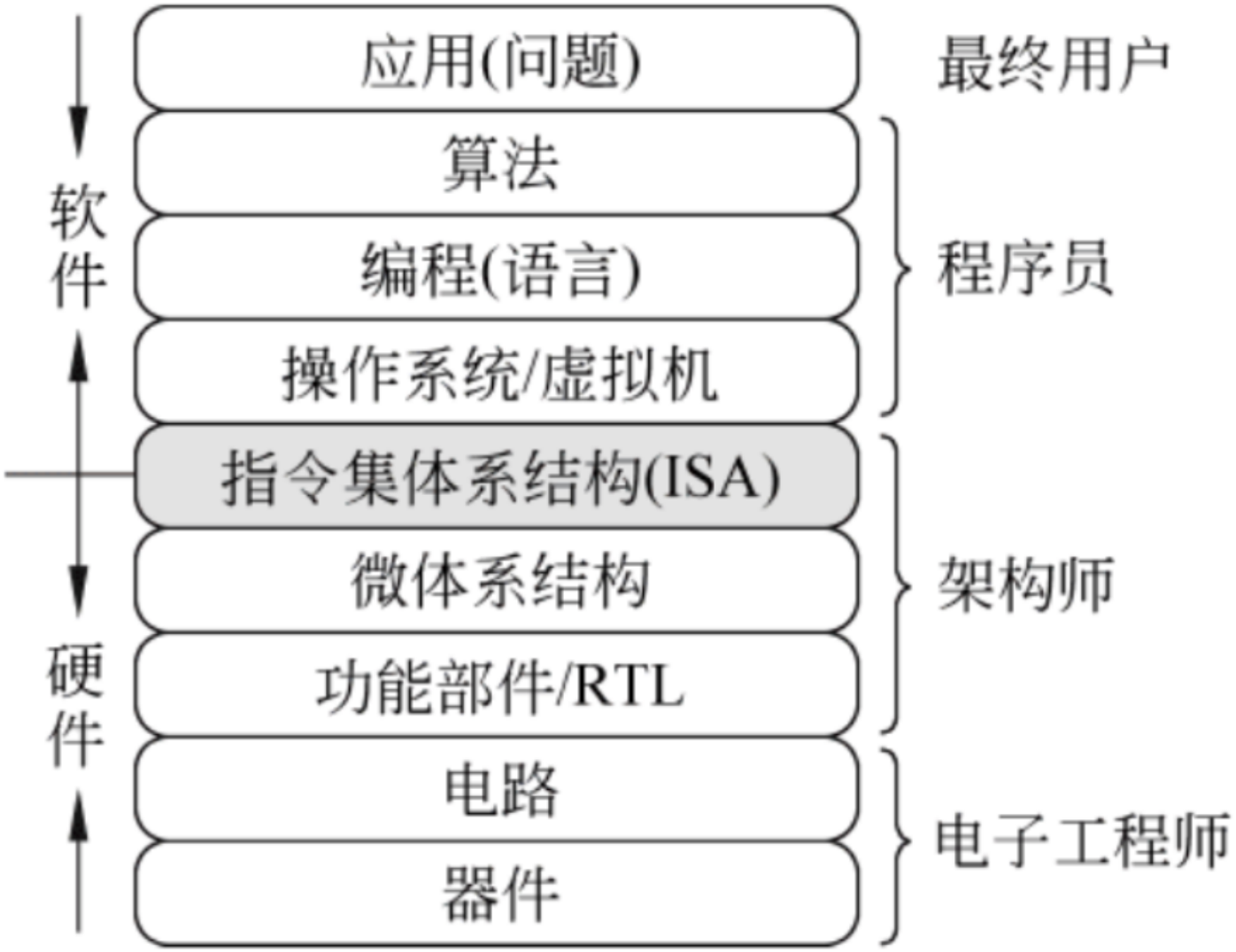



图 1.6　计算机系统抽象层及其转换

2. 计算机系统的不同用户

**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

**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

**应用程序员**：使用高级编程语言编制应用软件的程序员

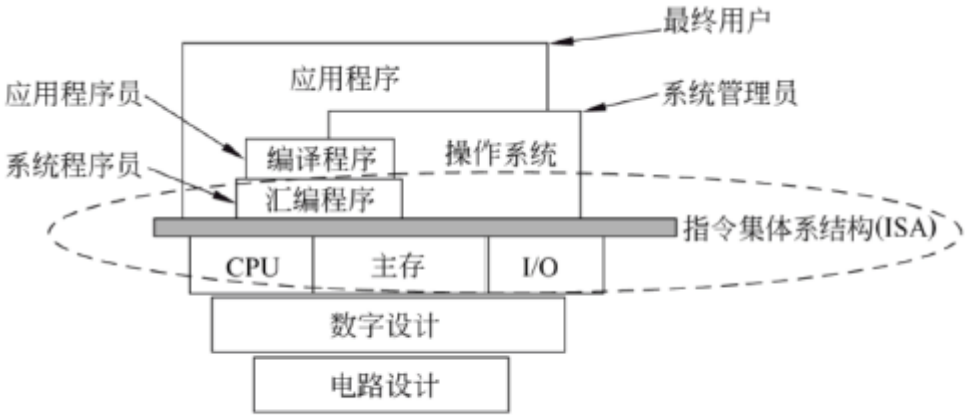



图 1.7　计算机系统的层次化结构

## 四、程序开发与执行过程

从源程序到可执行程序

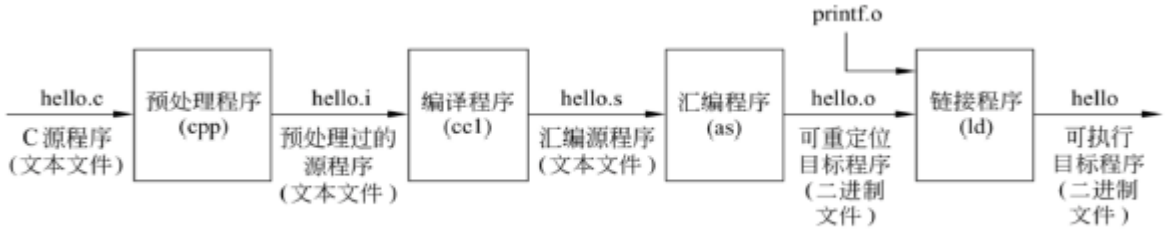



图 1.9　hello.c 源程序文件到可执行目标文件的转换过程

可执行程序的启动和执行

1

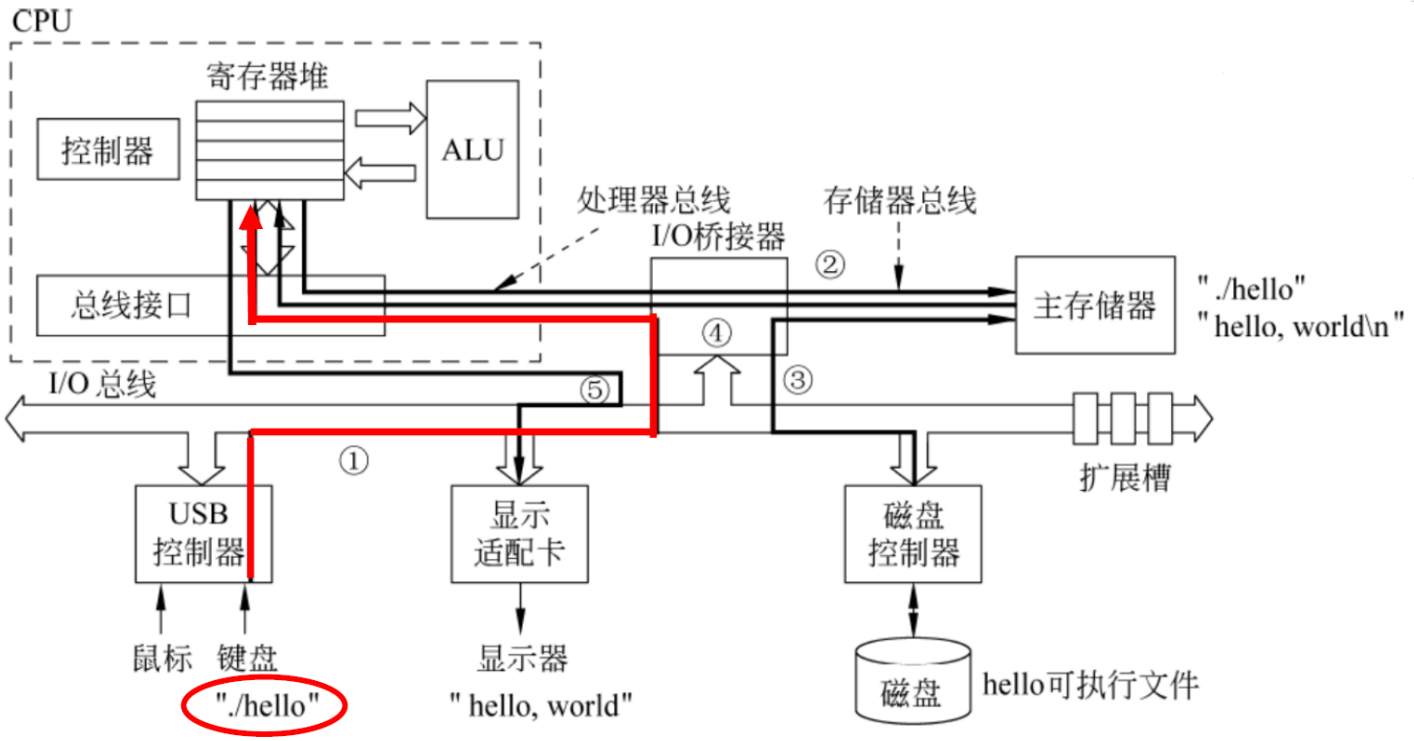



图 1.10　启动和执行 hello 程序的整个过程

3

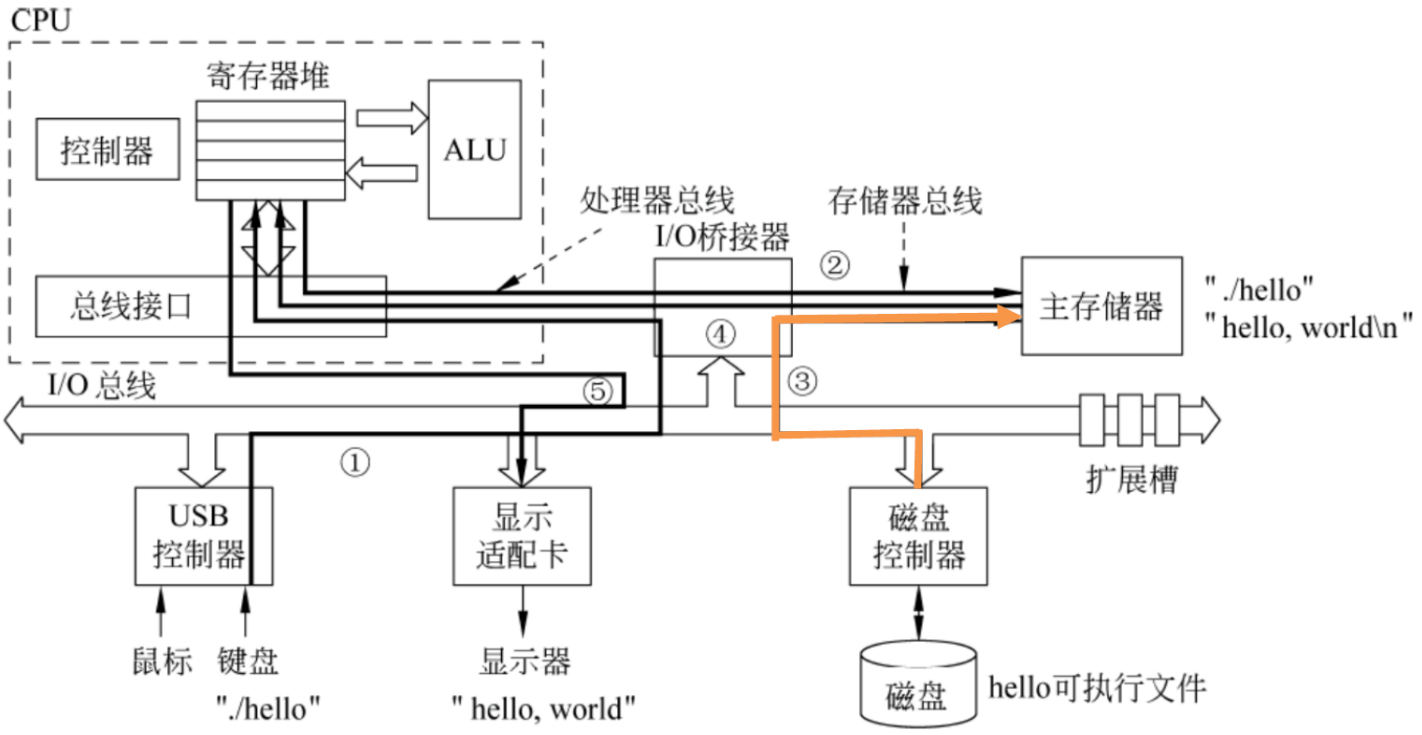



图 1.10　启动和执行 hello 程序的整个过程

5

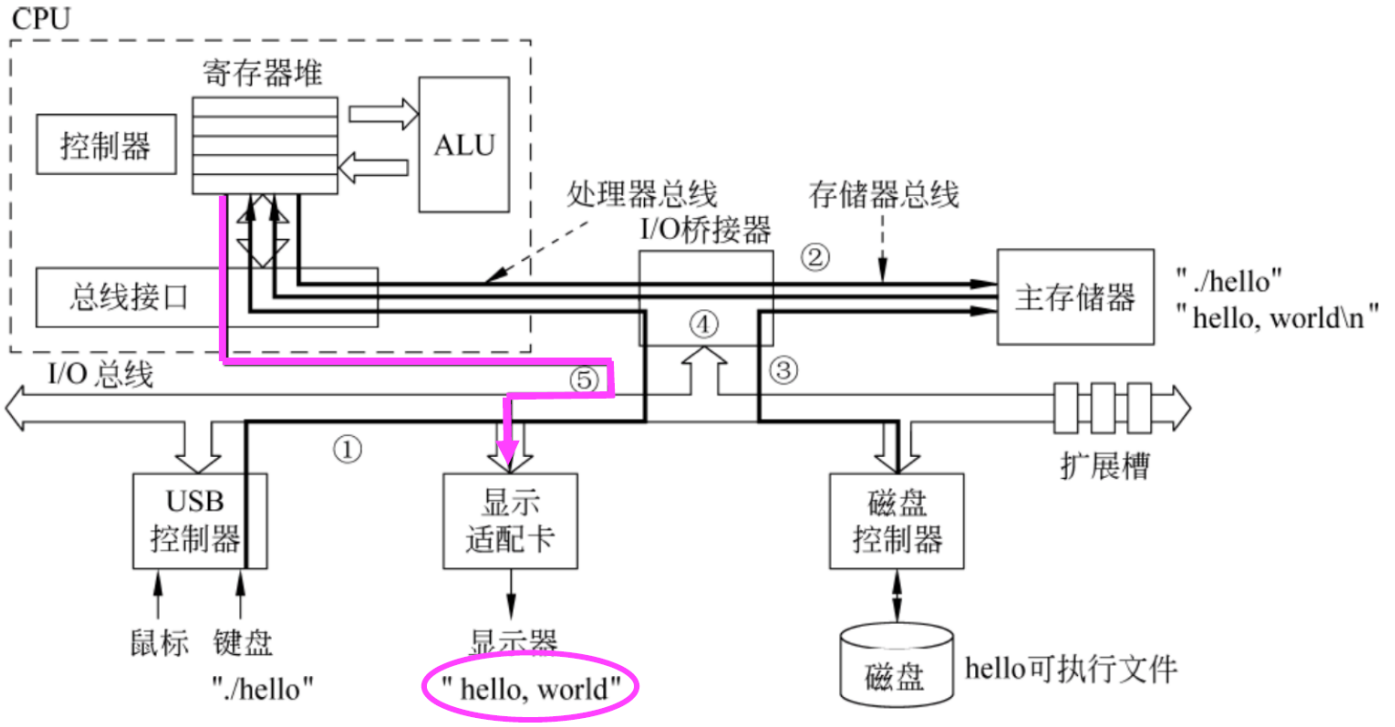



图 1.10　启动和执行 hello 程序的整个过程

2

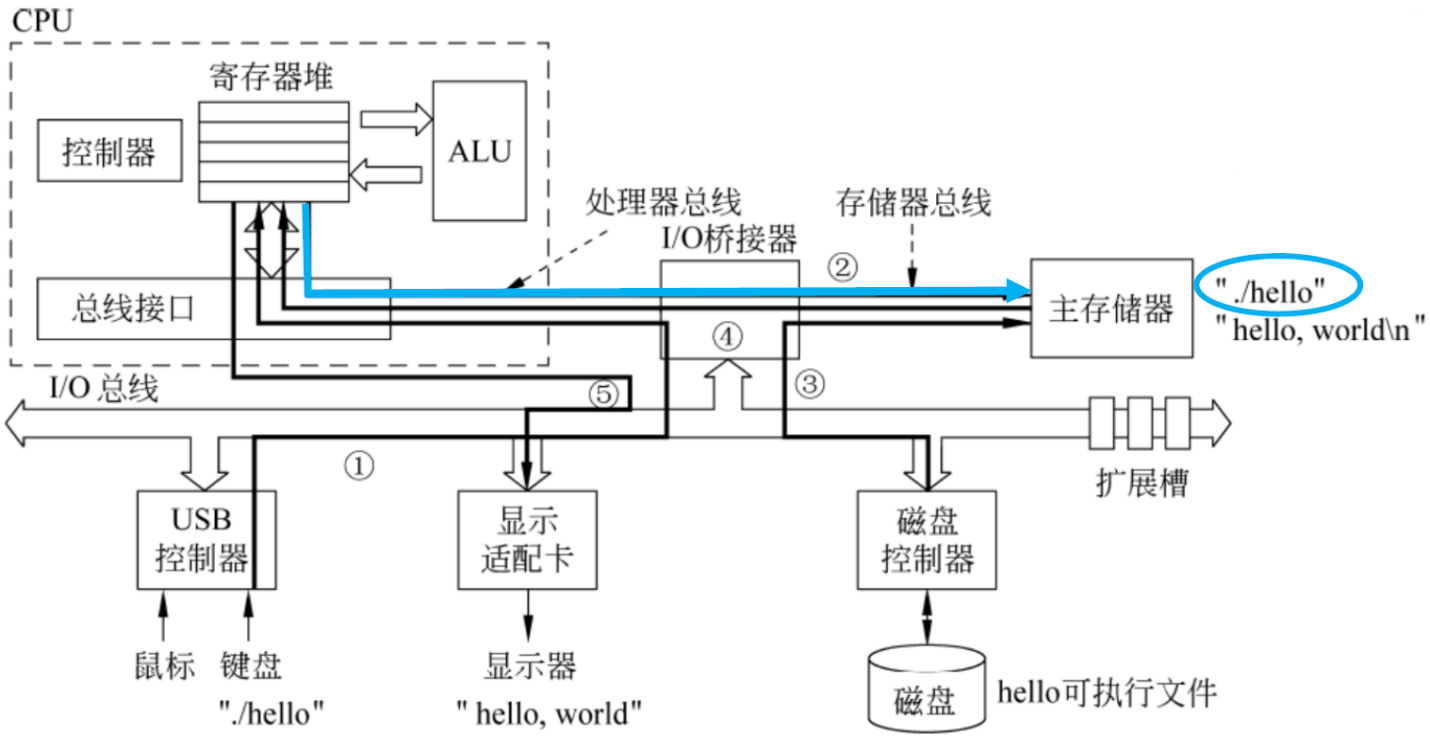



图 1.10　启动和执行 hello 程序的整个过程

4

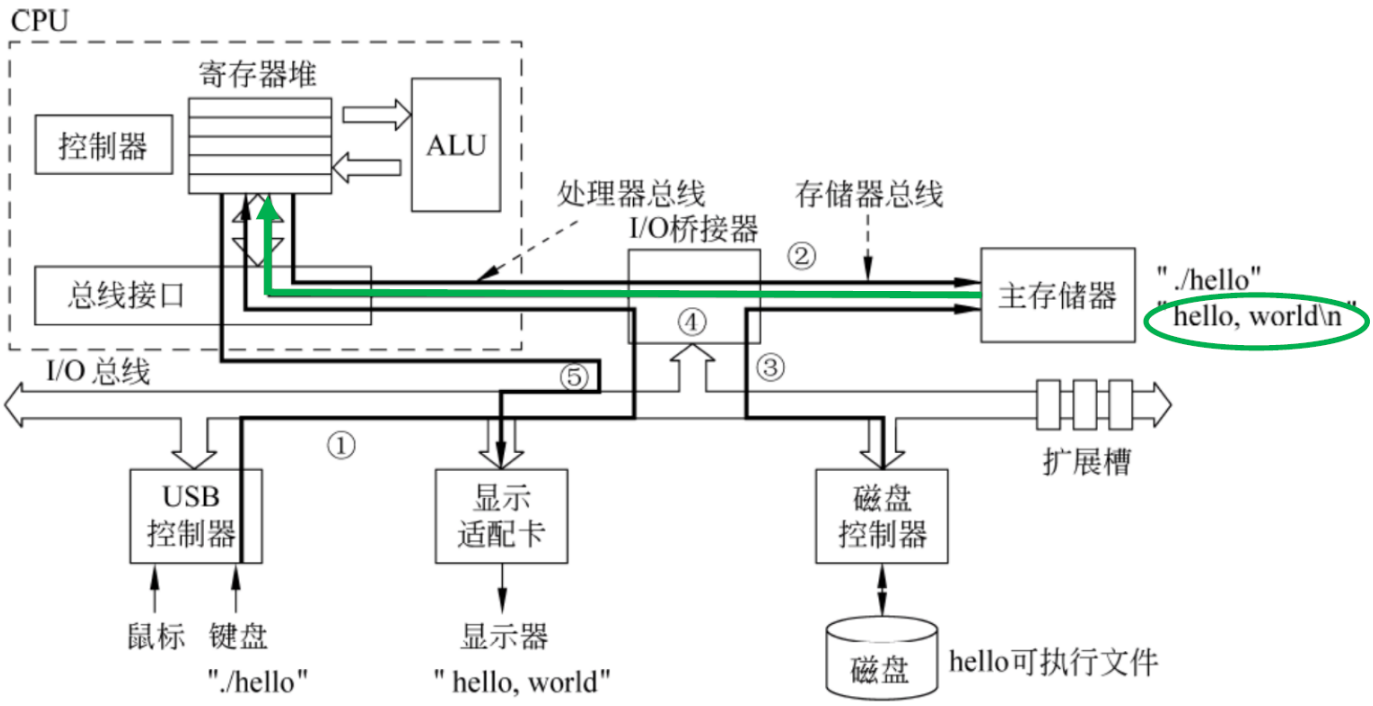



图 1.10　启动和执行 hello 程序的整个过程

## 程序与指令及控制信号的关系

程序以可执行文件形式存放在外存（包含机器代码）。可执行文件的执行实际上是所包含的机器代码段执行的过程。

指令：用0和1表示的一串0/1序列，用来指示CPU完成一个特定的原子操作

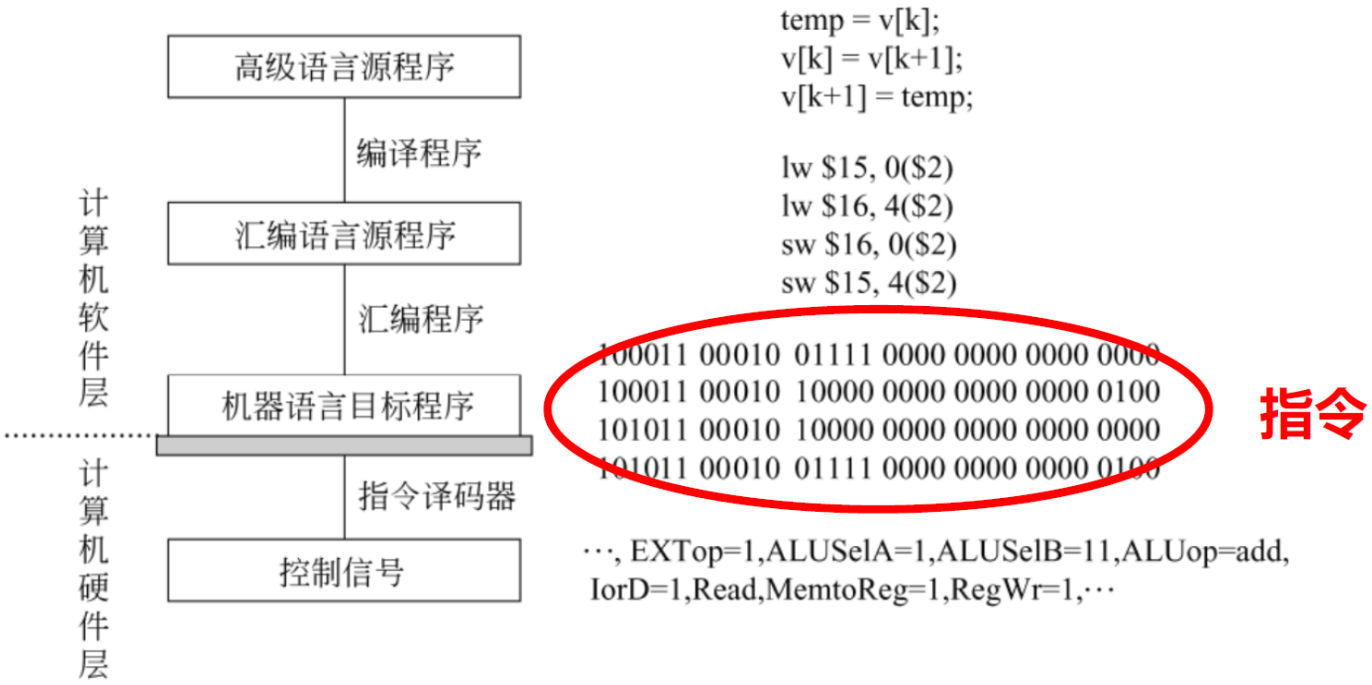
指令划分：

- 操作码字段：指令的操作类型

- 地址码字段：指令所处理的操作数所存放的寄存器的编号或所在主存单元的地址
- 立即数字段：具体的一个操作数或偏移地址

CPU通过逻辑电路直接执行用二进制表示的机器指令。

- 控制器中的指令译码器将指令操作码进行译码，以解释成**控制信号**来控制数据通路的执行




## 指令的执行过程

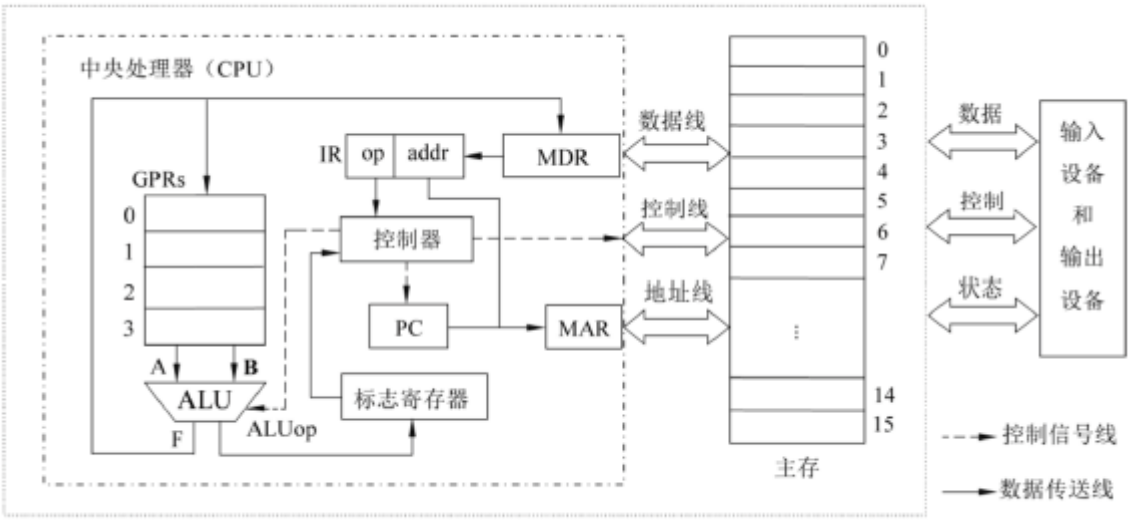



图 1.12　冯・诺依曼结构模型机

ALU：算术运算和逻辑运算

通用寄存器：指令指定编号的寄存器中的数据作为ALU运算的操作数

标志寄存器：存放ALU运算得到的一些标志信息

IR：存放从主存读出的指令

指令中的操作码op被送到控制器进行

主存（一个字节（8位））：存储指令和操作数

CPU和主存间通过一组**总线**相连

- 地址线：MAR中的地址信息会直接送到地址线上
- 控制线：读/写信号线，根据控制信号决定MDR中的数据送到数据线上或数据线上的数据送到MDR

1. 根据PC取指令到IR
2. 指令译码并送出控制信号
3. 读寄存器并进行ALU运算
4. 写结果寄存器或读写内存

---

## 五、计算机系统性能评价

### 计算机性能的定义

**两个基本指标：**

1. 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
2. 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

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！**

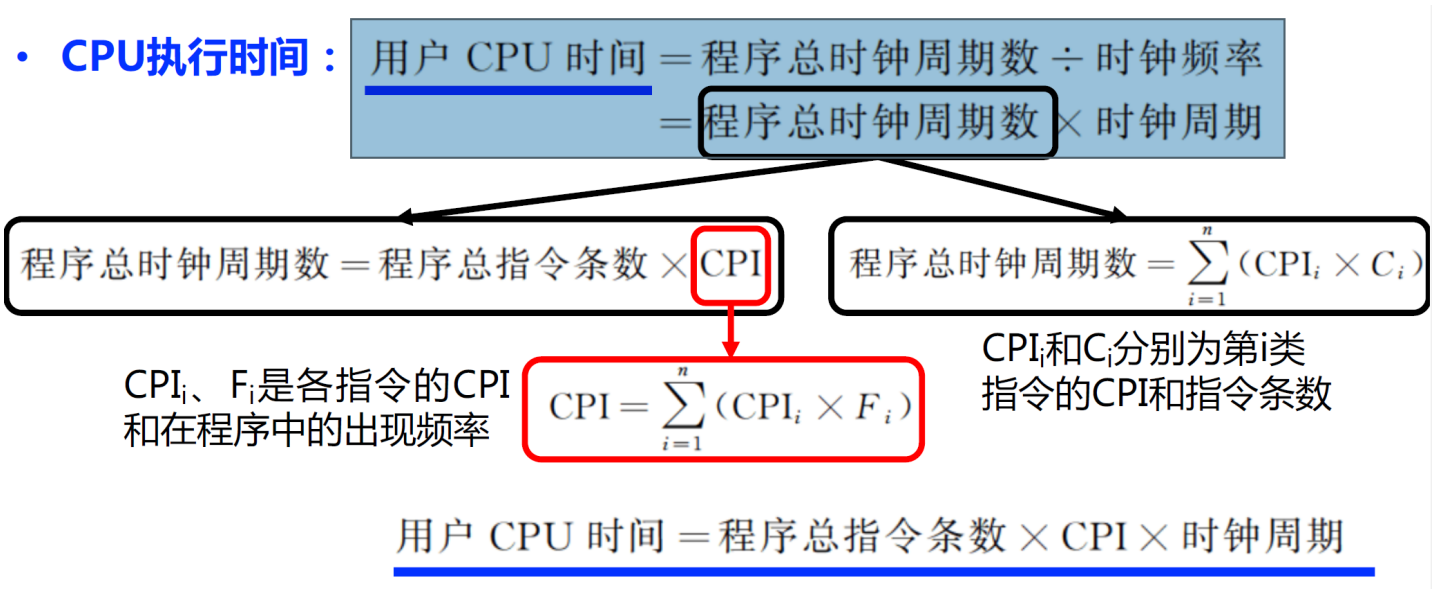
**用户CPU时间设计的概念和参数**

**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。

计算机的性能之比是用户**CPU**时间之比的倒数！

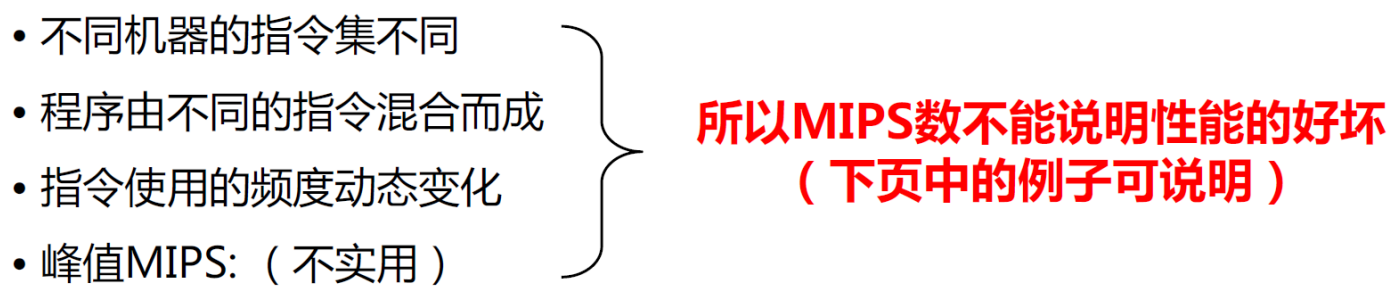



时钟周期、指令条数、CPI相互制约：

- 提高时钟频率不能保证速度同倍数提高。
- 指令条数最少的程序不一定执行最快

## 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。




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


**MIPS=指令数/（执行时间 × 1000000）=主频/(CPI × 1000000)**

## 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机器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。

SPEC：引用最广泛也是最全面的基准程序集

## 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。

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

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

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