

# 第 1 章 概论

“计算机组成原理”这门课程的主要内容是以单机系统为对象，阐述计算机系统的硬件组成，其核心是建立计算机系统的整机概念。这里的整机概念包括两个方面，即计算机系统的逻辑组成与工作机制。本书将从 CPU 级和硬件系统级这两个层次逐步建立整机概念。为此，在概论这一章首先阐明 3 个重要的基本概念：信息的数字化表示，存储程序工作方式和计算机系统的层次结构；将以上概念作为了解计算机的逻辑组成与工作机制的基本出发点。

## 1.1 计算机的基本概念

### 1.1.1 计算机与诺依曼体制

计算机是 20 世纪人类最伟大的发明之一。它通过自动、连续地运行程序，能够代替人类完成各种复杂的计算和实现对各类信息的处理。

一个计算机系统由硬件和软件两大部分组成。硬件是指看得见、摸得着的设备实体，包括运算器、控制器、存储器、输入设备、输出设备等，如图 1-1 所示。软件则不能直接触摸，比如程序、文档等。构造硬件的基本思想是处理功能逻辑化，即用逻辑电路构造各种功能部件，如用门电路、触发器来构造运算器、控制器、存储器等。在硬件基础上，可以根据需要配置各种软件，如操作系统、编程语言、各种支撑软件等。硬件与软件按层次结构组成复杂的计算机系统。

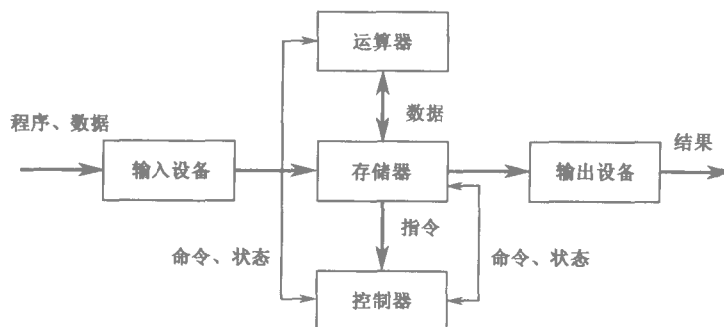


图 1-1 计算机组成示意图

一个计算机系统是如何工作的呢？不管做一次复杂的数学计算，还是对大量的数据进行查询，或者对一个过程实现自动控制，用户都必须按照处理的步骤，用编程语言事先编写程序，然后通过输入设备（如键盘）将程序和需要处理的数据送入计算机，存放在存储器中。用户编写的程序称为源程序，是不能被计算机直接执行的。计算机只能执行机器指令，即要求计算机完成某种操作的命令，简称指令，如执行加法操作的加法指令、执行乘法操作的乘法指令、执行传送操作的传送指令，等等。因此，计算机在运行程序之前，必须将源程序转换为指令序列，并将这些指令按一定顺序存放在存储器的若干单元中。每个单元都有一个固

定的编号，称为地址。只要给出某个地址，就能访问相应的存储单元，对该单元的内容进行读 / 写操作。

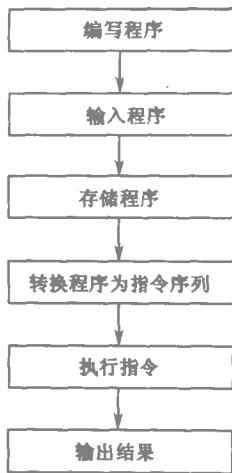


图 1-2 计算机工作流程

当计算机启动运行后，控制器将某个地址送往存储器，从该地址单元取回一条指令。控制器根据这条指令的含义，发出相应的操作命令，控制该指令的执行。比如执行一条加法指令，先要从存储单元或寄存器中取出操作数，送入运算器，再将两个操作数相加，并将运算处理的结果送回存储单元或寄存器存放。如果用户需要了解处理结果，则计算机通过输出设备（如显示器、打印机等），将结果显示在屏幕上，或打印在纸上。图 1-2 给出了计算机的简单工作流程。

从以上的描述可以看出，计算机作为一个处理信息的工具，首先需要解决两个最基本的问题：第一，信息如何表示，才能被计算机识别；第二，采用什么工作方式，才能使计算机自动地对信息进行处理。对这两个问题的解决做出杰出贡献，并且产生深远影响的是一位美籍匈牙利数学家冯·诺依曼。他在 1945 年提出的 EDVAC 计算机设计方案中，总结了计算机的设计思想，被称为诺依曼思想，采用这一思想体制的计算机就称为诺依曼机。几十年来，尽管计算机的体系结构发生了许多演变，但是诺依曼体制的核心思想仍然是普遍采用的结构原则，现在绝大多数实用的计算机仍属于诺依曼计算机。

诺依曼体制的主要思想包括：

- 采用二进制代码形式表示信息（数据、指令）；
- 采用存储程序工作方式（这是诺依曼思想最核心的概念）；
- 计算机硬件系统由五大部件（存储器、运算器、控制器、输入设备和输出设备）组成。

这些思想奠定了现代计算机的基本结构，并且开创了程序设计的新时代。当然，从本质上讲，传统的诺依曼机采用串行处理的工作机制，即逐条执行指令序列。要想提高计算机的性能，其根本方向之一是采取并行处理机制，例如用多个处理部件形成流水线，依靠时间上的重叠来提高处理效率；或者用多个诺依曼机组成多机系统，以支持并行算法，等等。

学习计算机工作原理应该从了解诺依曼思想入门，因此本节先讨论信息的数字化表示和存储程序工作方式这两个要点，下一节再讨论硬件系统的组成。

### 1.1.2 信息的数字化表示

计算机是通过执行程序（指令序列）来实现对数据的加工处理的。因此，计算机中的信息可以分为两大类：控制信息和数据信息。控制信息用来控制计算机的工作。计算机执行指令时，用指令产生的控制命令（称为微命令）控制有关操作，所以指令序列和微命令序列属于控制信息类。数据信息是计算机加工处理的对象。计算机根据指令要求取出的操作数，以及对操作数进行处理的结果，等等，都属于数据信息类。数据信息又分为数值型数据和非数值型数据两类。数值型数据有数值大小与正负之分，如 6、-15 等。非数值型数据则无数值大小，也不分正数负数，如字符、文字、图像、声音等人们能够识别的信息，以及条件、状态、命令等用于判定的逻辑信息。那么，在计算机中如何表示这些信息呢？

计算机的主要部件是用逻辑电路，即电子电路构成的，所以，在电子数字计算机中传送与处理的信息都采用数字化表示方法。信息的数字化表示包含了两层含义：

- 用数字代码表示各种信息；
- 用数字信号表示数字代码。

信息表示数字化这个重要概念是理解计算机工作原理的一个基本出发点。下面通过多个例子说明这两层含义。

### 1. 在计算机中用数字代码表示各种信息

数字代码是指一组数字的集合，这里的数字代码通常指二进制数字代码。可以根据需要描述的信息（某类控制信息或某类数据信息），用一组约定含义的数字代码来表示它。

【例 1-1】用数字代码表示数值型数据。

6 和 -7 是两个数值型数据。可以约定，用一位二进制代码表示每个数的符号，比如用 0 表示正数，用 1 表示负数；再用 4 位二进制代码表示每个数的大小。这样，代码 00110 表示 6（左边第一个 0 代表正号），代码 10111 表示 -7（左边第一个 1 代表负号）。

当然，也可以用 8 位二进制代码表示一个数的大小。代码位数增多，则数的表示范围将扩大。例如，用 4 位二进制代码不能表示 200 这个数，但用 8 位二进制代码则可以表示它。

【例 1-2】用数字代码表示字符。

字符本身没有大小和正负之分，但仍然可用数字代码来表示它。计算机中常常约定用 7 位代码表示一个西文字符，如用 1000001 表示字符 A，用 1000010 表示字符 B；或用 7 位代码表示一个控制字符，如用 0001100 表示换页（FF），用 0001101 表示回车（CR）。字符的这种编码称为 ASCII 码，是国际上广泛采用的一种字符表示方法。另外，还可约定用两组 8 位二进制代码表示一个中文字符，如用 01010110 01010001 表示“中”，用 00111001 01111010 表示“国”，等等。总之，用数字代码可以表示各种字符，而以字符为基础又可以表示范围广泛的各种文字。

【例 1-3】用数字代码表示图像。

字符的种类总是有限的，因而可以用若干位编码来表示；图像则不然，其变化是无穷无尽的，那么如何用数字代码来表示这些随机分布的图像信息呢？实际上，一幅图像可以被细分为若干个点，这些点称为像点。也就是说，可用像点的组合来逼近真实的图像。图像划分得越细，像点越多，组成的图像也就越真实。按照信息表示数字化的思想，可用数字代码表示像点。例如用一位代码表示一个像点，若像点是亮的，则用代码 1 表示；若像点是暗的，则用代码 0 表示。再将表示一幅图像所有像点的代码按照像点在图像中的位置进行组织，就可以实现用数字代码来表示图像了。

【例 1-4】用数字代码表示声音。

为了对声音信息数字化，首先要将声波转换为电波，再按一定频率对电波进行采样，即在长度相同的时间间隔内分别对电流波的幅值进行测量，每次测到的电流幅值都用一个数字量来表示。只要采样频率足够高，所得到的数字信息就能逼真地保持声波信息，还原后真实地再现原来的声音。

【例 1-5】用数字代码表示指令。

指令属于控制信息。通常，一条指令需要提供要求计算机做什么操作，以及如何获取操作数等信息。因此，可用一段数字代码表示操作类型，这段代码称为操作码；用另一段代码

表示获取操作数的途径，这段代码称为地址码。将操作码和地址码组合在一起，就形成机器指令代码。例如，操作码取 4 位，可以约定用 0000 表示传送操作，0001 表示加法操作，0010 表示减法操作……；地址码取 12 位，用 6 位表示一个操作数的来源，用其余 6 位表示另一个操作数的来源。如约定用 000000 表示操作数来自 0 号寄存器，000001 表示另一操作数来自 1 号寄存器。这样，16 位代码 (0010000000000001 表示一条加法指令，它的含义是将 0 号寄存器的内容与 1 号寄存器的内容相加，结果存放在 1 号寄存器中。

#### 【例 1-6】用数字代码表示设备状态。

计算机在工作时往往需要了解外部设备的状态，根据外设状态决定做什么操作。不同的外部设备可能有不同的工作状态，例如打印机将字符打印在纸上，而显示器则将字符显示在屏幕上。我们可以将这些设备的状态抽象、归纳为 3 种：空闲（设备没有工作）、忙（设备正在工作）、完成（设备做完一次操作）。相应地，可用约定的数字代码表示这 3 种状态，比如，用 00 表示空闲，01 表示忙，10 表示完成。

### 2. 在物理机制上用数字信号表示数字代码

为什么要用数字代码表示各种信息呢？这就涉及计算机的物理机制了。计算机是一种复杂的电子线路，它传送和处理的实际对象是电信号。电信号又分为模拟信号和数字信号两种类型。

模拟信号是一种随时间连续变化的电信号，如电流信号、电压信号等。我们可用电流或电压的幅值来模拟数值或物理量的大小，比如模拟温度的高低、压力的大小等。处理模拟信号的计算机称为模拟计算机，只应用在极特殊的领域中。用模拟信号表示数据的大小有许多缺点，如表示的精度低、表示的范围小、抗干扰能力弱、不便于存储等。用数字信号表示信息则可以克服以上缺点。

数字信号是一种在时间上或空间上断续变化的电信号，如电平信号和脉冲信号。单个数字信号一般只取两种状态，例如电平的高或低、脉冲的有或无，这样，就可用这两种状态分别表示数字代码 1 和 0，称为二值逻辑。比如，用高电平状态表示 1，低电平状态表示 0；或者用有脉冲的状态表示 1，无脉冲的状态表示 0。用一位数字信号表示一位数字代码，用多位

数字信号的组合就可表示多位数字代码。处理数字信号的计算机称为数字计算机，电平信号和脉冲信号是数字计算机中最基本的信号形式。由于可用数字信号表示数字代码，用数字代码又可表示各种信息，因而数字计算机能够用于各行各业，处理广泛的信息。下面通过两个例子说明如何用多位数字信号的组合来表示多位数字代码。

#### 【例 1-7】用一组电平信号表示四位数字代码。

电平信号利用信号电平的高、低状态表示不同的代码，所以电平信号通常需要一段有效维持时间。可用 4 根信号线分别输出 4 个电平信号，每个电平信号表示一位代码。我们约定 +5V 为高电平，表示 1；0V 为低电平，表示 0。如图 1-3 所示，4 位电平信号表示 4 位数字代码 1011，它们可能表示一个 4 位的二进制数，也可能表示一个命令或一种状态的编码。

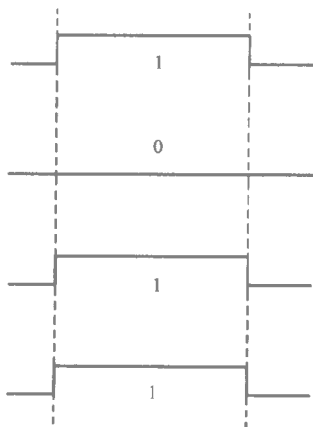


图 1-3 用一组电平信号表示多位数字代码

每位信号各占用一根信号线，因而这组电平信号在空间上的分布是离散的。在计算机中常用电平信号表示并行传送的信息，如用若干根信号线同时传送的数据、地址或其他信息的编码。

【例 1-8】用一串脉冲信号表示 4 位数字代码。

与电平信号不同，脉冲信号的电平维持时间很短，例如，信号电平从 0V 向 +5V（或 -5V）跳变，维持极短时间后再回到原来的 0V 状态。因此，信号出现时其电平为 +5V（或 -5V），信号未出现时其电平为 0V，如图 1-4 所示。由于脉冲信号在时间上的分布是离散的，因而可用一根信号线发出一串脉冲信号，在约定的时间内有脉冲表示 1 无脉冲表示 0。图 1-4 中的脉冲串表示 4 位数字代码 1011。

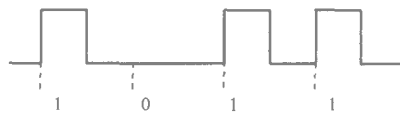


图 1-4 用一串脉冲信号表示多位数字代码

可以使用脉冲信号的上升边沿或下降边沿表示某一时刻，对某些操作定时。例如，在脉冲上升边沿将数据送入某个寄存器中。另外，在计算机中常用脉冲信号表示串行传送的数据。

从事计算机技术工作的重要基础，是善于使用约定的数字代码表示各种需要描述的信息。这里，再次强调信息数字化这个重要概念：

从事计算机技术工作的重要基础，是善于使用约定的数字代码表示各种需要描述的信息。这里，再次强调信息数字化这个重要概念：

- 计算机中的各种信息用数字代码表示，这些信息包括数值型的数字，非数值型的字符、图像、声音，逻辑型的命令、状态，等等；
- 数字代码中的每位用数字信号表示，数字信号可以是电平信号或脉冲信号。

### 3. 用数字化方法表示信息的主要优点

#### (1) 在物理上容易实现信息的表示与存储

每位信号只取两种可能的状态表示 1 或 0，因而在物理上可用多种方法来实现，如开关的接通或断开、晶体管的导通或截止、电容上有电荷或无电荷、磁性材料的正向磁化或反向磁化、磁化状态的变化或不变，等等。凡是具有两种稳定状态的物理介质均可用来存储信息，比如，用双稳态触发器存储信息，或用电容上存储的电荷来存储信息，还可以用磁性材料记录信息，或者用激光照射过的介质记录信息。

#### (2) 抗干扰能力强，可靠性高

由于单个数字信号的两种状态（高电平与低电平，或者有脉冲与无脉冲）差别较大，即使信号受到一定程度的干扰，仍然能比较可靠地鉴别电平的高低或信号的有无。例如高电平 +5V 表示 1，低电平 0V 表示 0，假设信号处于 0 状态，如果出现了 2V 的干扰信号，也不会将原来信号的 0 状态改变到 1 状态。

#### (3) 数值的表示范围大，表示精度高

一位数字信号的表示范围很窄，但用多位数字信号的组合表示一个数时，可以获得很大的表示范围和很高的精度。例如，用 4 位电平信号表示一个 4 位的二进制整数时（不考虑符号）能够表示的最大数值是 15。若表示一个 4 位的二进制小数，同样不考虑符号，则数的精度为  $2^{-4}$ 。位数越多，数的表示范围越大，或者数的表示精度越高。从理论上讲，位数的增加是有限制的，但位数增多，所花费的硬件开销也相应增大。

#### (4) 表示的信息类型极其广泛

各种非电量类型的信息可以先转换为电信号，模拟电信号又可以转换为数字电信号，因此表示的信息类型和范围几乎没有限制。

### (5) 能用数字逻辑技术进行信息处理

根据处理功能逻辑化的思想，计算机的所有操作最终是用数字逻辑电路来实现的。因此，用逻辑代数对信息进行处理，就形成了计算机硬件设计的基础，可以用非常有限的几种逻辑单元（如与门、或门、非门等）构造出变化无穷的计算机系统和其他数字系统。

### 1.1.3 存储程序工作方式

存储程序是诺依曼思想的核心内容，它表明了计算机的工作方式，包含以下 3 个要点：

- 事先编制程序；
- 事先存储程序；
- 自动、连续地执行程序。

这 3 点体现了用计算机求解问题的过程，下面分别加以说明。

#### 1. 根据求解问题事先编制程序

计算机处理任何复杂的问题都是通过执行程序来实现的。因此，在求解某个问题时，用户需要根据解决这个问题所采用的算法事先编制程序，规定计算机需要做哪些事情，按什么步骤去做。程序中还应提供需要处理的数据，或者规定计算机在什么时候、什么情况下从输入设备取得数据，或向输出设备输出数据。

#### 2. 事先将程序存入计算机中

如前所述，用户用某种编程语言编写的程序称为源程序，它是由字符组成的，计算机不能识别。因此，需要通过输入设备将源程序转换为二进制代码，送入计算机的存储器中。这时的程序还不是指令代码，不能被计算机执行，还需进一步转换为符合某种格式的机器指令序列。所以，事先编写的程序最终将变为指令序列和原始数据，且被保存在存储器中，提供给计算机执行。

#### 3. 计算机自动、连续地执行程序

程序已经存储在计算机内部。计算机被启动后，不需要人工干预，就能自动、连续地从存储器中逐条读取指令，按指令要求完成相应操作，直到整个程序执行完毕。当然，在某些采用人机对话方式工作的场合，也允许用户以外部请求方式干预程序的运行。

指令和数据都以二进制代码的形式存放在存储器中，那么计算机如何区分它们？又如何自动地从存储器中读取指令呢？首先，将指令和数据分开存放。由于多数情况下程序是顺序执行的，因此大多数指令需要依次紧靠着存放，而将数据放在该程序区中不同的区间。其次，可以设置一个程序计数器（Program Counter），称为 PC，用它存放当前指令所在的存储单元的地址。如果程序顺序执行，则在读取当前指令后将 PC 的内容加 1（当前指令只占用一个存储单元），指示下一条指令的地址。如果程序需要转移，则将转移目标地址送入 PC，以便按照转移地址读取后续指令。所以，依靠 PC 的指示，计算机就能自动地从存储器中读取指令，再根据指令提供的操作数地址读取数据。

对于传统的诺依曼机而言，存储程序工作方式是一种控制流驱动方式，即按照指令的执行序列依次读取指令，再根据指令所含的控制信息调用数据进行处理。这里的控制流也称指令流，是指在程序执行过程中，各条指令逐步发出的控制信息，它们始终驱动计算机工作。而依次被处理的数据信息称为数据流，它们是被驱动的对象。

## 1.2 计算机系统的组织

### 1.2.1 硬件系统

在诺依曼体制中，计算机硬件系统是由存储器、运算器、控制器、输入设备和输出设备等五大部件组成的。随着计算机技术的发展，硬件系统的组成已发生许多重大变化，如运算器和控制器组合成一个整体，称为中央处理机器（Central Processing Unit），简称 CPU；存储器成为多级存储器，它包含主存、外存和高速缓存 3 个层次。下面先以目前常见的计算机硬件系统组成为例，讨论系统中各部件应该具有哪些功能，以及这些部件通过什么方式相互连接，构成整机等硬件设计方面的问题；然后简单介绍一些典型的硬件系统结构。

#### 1. 常见计算机硬件系统组成

图 1-5 是一个常见的计算机硬件系统结构示意图。系统中有 CPU、存储器、I/O 设备和接口等部件，它们通过系统总线相连接。

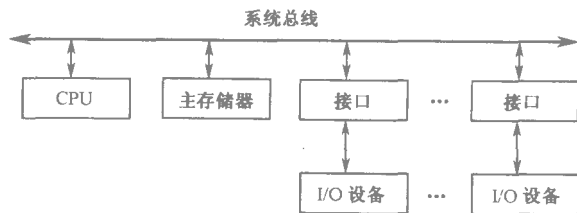


图 1-5 常见计算机硬件系统框图

#### (1) CPU

CPU 是计算机硬件系统的核心部件，在微机系统或其他应用大规模集成电路技术的系统中，它被集成在一块芯片上，构成微处理器。CPU 的主要功能是读取并执行指令，在执行指令的过程中，它向系统中的各个部件发出各种控制信息，收集各部件的状态信息，与各部件交换数据信息。

CPU 由运算部件、寄存器组和控制器组成，这些部件通过 CPU 内部的总线相互交换信息。运算部件完成算术运算（定点数运算、浮点数运算）和逻辑运算。寄存器组用来存放数据信息和控制信息。控制器提供整个系统工作所需的各种微命令，这些微命令通过若干组合逻辑电路产生，称为组合逻辑控制方式；也可通过执行微程序产生，称为微程序控制方式。

CPU 由运算部件、寄存器组和控制器组成，这些部件通过 CPU 内部的总线相互交换信息。运算部件完成算术运算（定点数运算、浮点数运算）和逻辑运算。寄存器组用来存放数据信息和控制信息。控制器提供整个系统工作所需的各种微命令，这些微命令通过若干组合逻辑电路产生，称为组合逻辑控制方式；也可通过执行微程序产生，称为微程序控制方式。

#### (2) 存储器

存储器用来存储信息，包括程序、数据、文档等。如果存储器的存储容量越大、存取速度越快，那么系统的处理能力也就越强、工作速度也就越高。一个存储器很难同时满足大容量、高速度的要求，因此常将存储器分为主存、外存、高速缓存等三级存储器。

主存用来存放 CPU 需要使用的程序和数据。主存的每个存储单元都有固定的地址，CPU 可按地址直接访问它们。因此，主存的速度较快，但容量有限。通常用半导体存储器构成主存。一般将 CPU 和主存合称为主机，因主存在主机之内，所以又称内存。

外存位于主机之外，用来存放大量的需要联机保存，但 CPU 暂不使用的程序和数据。需要时 CPU 并不直接按地址访问它们，而是按文件名将它们从外存调入主存。因此外存的容量很大，但存取速度比主存慢。像磁盘、光盘等，都是常用的外存。

高速缓存是为了提高 CPU 的访存速度，在 CPU 和主存之间设置的一级速度很快的存储器，容量较小，用来存放 CPU 当前正在使用的程序和数据。高速缓存的地址总与主存某一区间的地址相映射，工作时 CPU 首先访问高速缓存，如果未找到所需的内容，再访问主存。高速缓存由高速的半导体存储器构成。在现代计算机中，常将高速缓存集成在 CPU 内部，称为

片内 cache，或称一级 cache；也可以在 CPU 外部再设置一级高速缓存，称为片外 cache，或称二级 cache。

### (3) 输入/输出设备

输入设备将各种形式的外部信息转换为计算机能够识别的代码形式送入主机，常见的输入设备有键盘、鼠标等。输出设备将计算机处理的结果转换为人们所能识别的形式输出，常见的输出设备有显示器、打印机等。

从信息传送的角度来看，输入设备或输出设备都与主机之间传送数据，只是传送方向不同，因此常将输入设备和输出设备合称为 I/O 设备。它们在逻辑划分上位于主机之外，故又称为外围设备或外部设备，简称外设。磁盘、光盘等外存既可看成存储系统的一部分，也可看成具有存储能力的 I/O 设备。

### (4) 总线

总线是一组能为多个部件分时共享的信息传送线。现代计算机普遍采用总线结构，用一组系统总线将 CPU、存储器和 I/O 设备连接起来，各部件通过这组总线交换信息。任何时刻只能允许一个部件或设备向总线发送信息，否则会引起信息的碰撞；但允许多个部件同时从总线接收信息。

根据系统总线上传送的信息类型，可将系统总线分为地址总线、数据总线和控制总线。地址总线用来传送 CPU 或外设发向主存的地址码；数据总线传送 CPU、主存以及外设之间需要交换的数据；控制总线用来传送控制信号，如时钟信号、CPU 发向主存或外设的读/写命令、外设送往 CPU 的请求信号……，等等。

### (5) 接口

在图 1-5 中，为什么系统总线与 I/O 设备之间设置了接口部件？这是因为计算机系统通常采用标准的系统总线，每种总线标准都规定了其地址线和数据线的位数、控制信号线的种类和数量等。计算机系统所连接的各种外部设备并不是标准的，在种类与数量上都是可变的。为了将标准的系统总线与各具特色的 I/O 设备连接起来，需要在系统总线与 I/O 设备之间设置一些部件，它们具有缓冲、转换、连接等功能，这些部件称为 I/O 接口。

计算机的各种操作都可归结为信息的传送，信息在计算机中沿着什么途径传送将直接影响到硬件系统结构。我们将信息在计算机中的传送途径称为数据通路结构。因此硬件系统结构的核心是数据通路结构。不同类型的计算机，如传统的微型机、小型机和大、中型机，其功能的侧重点不同，因而它们的数据通路结构是有区别的。下面介绍几种典型的计算机硬件系统结构及其特点。

## 2. 典型的硬件系统结构

### (1) 以总线为基础的系统结构

微型机和小型机系统往往侧重于以较低的硬件代价实现较强的系统功能。因此，常用一组系统总线作为系统中互连的基础，连接 CPU、存储器和 I/O 接口，再通过接口连接外部设备，如图 1-5 所示。已在前面这种总线结构形态做了描述，它有如下 3 个主要特点。

数据通路结构简洁。用一组总线代替各部件之间杂乱的连线，系统中各个部件都通过这组总线传送信息。

数据传送控制方便。发送部件通过三态门或 OC 组件向总线发送数据，控制逻辑用接收信号选择一个或多个部件从总线接收数据。

系统扩展易于实现。I/O 设备都是通过接口挂接在总线上的，只要总线负载允许，就可以增加新的接口，往总线上挂接新的 I/O 设备。

在实际机器中可能采用不同的总线结构。例如在 IBM-PC 微型计算机中，采用多级总线结构。在系统板上设置一组总线，称为局部总线，它将 CPU、主存、一些外围芯片连接成一个最小规模的系统。局部总线通过总线控制器和锁存器，扩展为 PC 总线，即系统总线。系统总线再通过接口部件连接各种外部设备，组成最大系统模式。

又比如在多微处理器系统中，使用一种多机互连总线 Multibus II，它将系统总线划分为若干组子总线。其中，并行系统总线用来实现多个 CPU 之间的互连；存储总线用于高速访问存储器；系统扩充 I/O 总线、多通道 DMA I/O 总线等，用于访问外部设备。

总之，总线结构是一种将各个部件连接成整机的基本结构，在实现上允许有多种变化。

## (2) 采用通道或输入/输出处理机的大型系统结构

大、中型计算机系统的构成更着重于扩大系统的处理功能与提高系统的运行速度，这就使得系统需要处理的数据量大大增加，管理的外设种类和数量也大大增多。因此，在大、中型计算机中，CPU 内部常采用多个运算处理部件，主存采用多个存储体交叉访问体制，采用多级存储体系，采用通道或输入/输出处理机来管理 I/O 操作。典型的大型系统结构在系统连接上形成主机、通道、I/O 控制器、I/O 设备等四级结构，如图 1-6 所示。

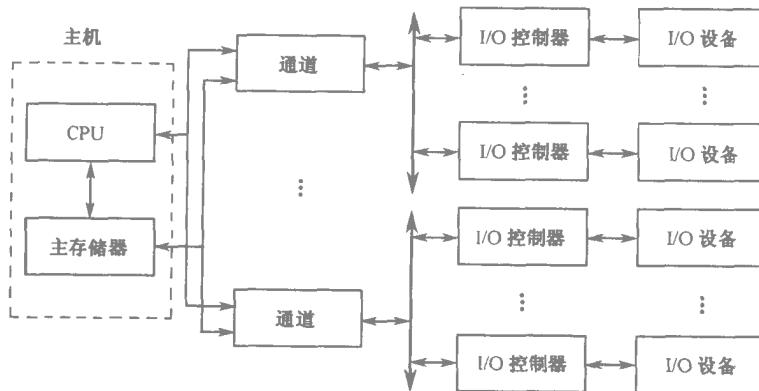


图 1-6 典型的大型系统结构

主机包括 CPU 与主存储器。由于主存负荷重，且整个系统的信息传送量大，在 CPU 与主存之间设置有专门的高速存储总线，CPU 与通道之间、主存与通道之间也都有各自独立的数据通路，使系统能够高速地工作。

通道也称通道控制器，能够执行专用的通道指令，是管理 I/O 操作的控制部件。CPU 启动通道后可以继续执行程序，进行本身的处理工作；通道则独立执行由通道指令编写的通道程序，控制 I/O 设备与主存的数据交换。这样，CPU 中的数据处理与 I/O 操作可以并行执行，使系统效率得到进一步提高。

I/O 控制器将通道发来的控制命令转换为具体操作命令，送往 I/O 设备以控制具体的 I/O 操作。

从系统结构来看，系统中可以设置几个通道，每个通道可以连接若干 I/O 控制器，每个 I/O 控制器又可连接若干相同类型的 I/O 设备。这样，整个系统就能够连接许多不同种类的外部设备。对于规模较小的系统，可将通道部件设置在 CPU 内部，组成一种结合型通道；对

于较大的系统，则可以将通道设置在 CPU 之外，成为独立的一级；对于更大的系统，可将通道发展成为功能更强的输入/输出处理机，称为 IOP。至于 I/O 控制器，它可能独立于 I/O 设备，也可能与 I/O 设备合为一体。需要指出的是，现在通道的概念不仅仅用于大、中型计算机，在微型计算机系统中也有集成化的 IOP 芯片，负责对 I/O 操作进行管理。

### (3) 多机系统

单个 CPU 的处理功能和运算速度总是有限的。因此，计算机系统结构的发展方向之一是多处理器化，即用多个 CPU 组成多机系统。根据多机之间连接的紧密程度，可将多机系统分为紧耦合与松耦合两种连接模式。

#### 紧耦合系统

这种系统的结构特点是通过总线或交叉开关矩阵进行多机互连，通过共享主存实现多机之间的信息交换。在较小规模的多机系统中，常常在系统总线上挂接多个处理单元，如图 1-7 所示。每个处理单元用微处理器作为自己的 CPU，并设置局部存储器或本地存储器供自己单独使用。在系统总线上挂接一个全局存储器，作为各个 CPU 都可访问的共享存储器。如果两个 CPU 交换数据，则其中一个 CPU 先将数据写入共享存储器，另一个 CPU 再从共享存储器读取数据。

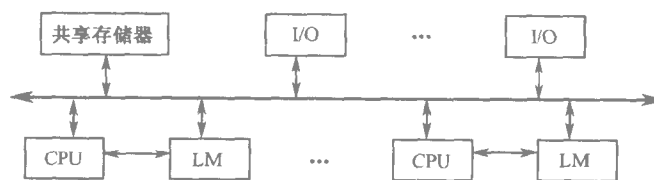


图 1-7 紧耦合多机系统结构

紧耦合多机系统还有一种应用模式，是用多个 CPU 连接成一块多处理器插件，该插件称为加速处理部件。将它插入某个通用单机系统中，起加速处理作用。在高速的并行处理多机系统中，使用一组总线往往不能满足速度要求，需要用一种高速的交叉开关矩阵来连接各个处理机。例如，研制成功的 GC 系列大规模并行处理机，用交叉开关矩阵 C104 将 16 个高速微处理器 T9000 连接成一个标准单元，再用这些标准单元组成积木式立体结构，系统中的微处理器数最多可达 16384 个。

#### 松耦合系统

松耦合系统的结构特点是用通信网络连接各节点，节点之间以中断方式传送信息包。每个节点内有一个 CPU、一个局部存储器，可能还有独自的外存或其他外设。通信网络的连接可以有星状、总线状、环状等多种模式。图 1-8 是一种典型的松耦合多机系统结构框图。每个节点是一个计算机模块，模块内用局部总线连接一个处理器 CPU、局部存储器 LM、一些 I/O 设备等。通过消息传送系统 MTS 将若干这样的节点连接成多机系统。MTS 可以是较简单的通信总线，也可以是比较复杂的互连网络。每个节点通过通信接口 CAS 与 MTS 相连。

## 1.2.2 软件系统

计算机软件通常包含各类程序和文件。一般来讲，程序是用字符和符号描述的某种算法的实现过程，最终体现为机器指令序列，要被计算机硬件执行。文件则是对编制程序和运行、维护程序所做的说明，例如对编程工具与运行环境的说明、帮助提示信息及其他参考信息等。

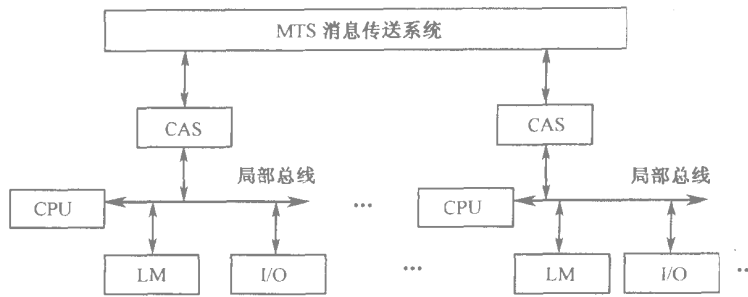


图 1-8 松耦合多机系统结构

在计算机系统中，各种软件有机地组合起来，构成了软件系统。从软件的功能与配置的角度出发，可将软件分为系统软件与应用软件两大类。

### 1. 系统软件

系统软件作为一种基础软件，其功能是负责系统的调度管理，提供程序的运行环境和开发环境，向用户提供各种服务，保证计算机系统能够良好地运行。从配置的角度来看，系统软件是用户所使用的计算机系统的一部分，通常作为系统资源（软设备）提供给用户使用。

#### (1) 操作系统

操作系统是软件系统的核心，负责管理和控制计算机系统的硬件资源、软件资源与运行的程序。它是用户和计算机之间的接口，为用户提供软件的开发环境和运行环境。

一个典型的操作系统由处理机调度，存储管理，设备管理，文件系统，作业调度等几大模块组成。这些模块相互配合，以尽可能优化的方式，调度管理系统的硬、软资源，合理地组织工作流程，提高系统的工作效率。例如，在具备多道程序运行环境的计算机系统中，需要处理机调度模块对处理机的分配和运行进行有效的管理；需要存储管理模块为各道程序分配内存空间，提供内存保护，使这些程序互不干扰；需要设备管理模块为用户程序分配 I/O 设备，提供良好的人机界面，完成相关的 I/O 操作；需要文件系统模块为大量的、以文件形式组织和保存的信息提供管理；需要作业调度模块对以作业形式存放在外存中的用户程序进行调度管理，将它们从外存调入主存，交由 CPU 运行。所以说，操作系统对计算机系统的硬件资源、软件资源和运行的程序进行了合理、高效的控制和管理。

对用户而言，往往是通过操作系统提供的各种功能来操作和使用计算机的，如 DOS 操作系统提供的键盘命令、Windows 操作系统提供的人机界面等。人们通过按键或移动鼠标，可以打开、关闭计算机，可以使用计算机完成任意合法操作。因此，操作系统提供了用户和计算机之间的接口。

当开发用户程序时，常常需要调用在操作系统管理下的某些软件资源，以便得到操作系统的最大支持。所编制的程序一般也作为文件，由操作系统进行管理。程序运行时可能需要调用操作系统管理的其他资源，使用操作系统的有关功能。所以，操作系统为用户提供了软件的开发环境和运行环境。

#### (2) 语言处理程序

用户通常使用程序设计语言来编写源程序。程序设计语言可以是 BASIC, Pascal, C, C++ 等高级编程语言，也可以是较低级的汇编语言。计算机硬件只能识别和执行由二进制代码表示的指令序列，所以需要语言处理程序将源程序转换为指令序列，即用机器语言表示

的目标程序。有两种基本的转换方式，一种是解释方式，另一种是编译方式。相应地，语言处理程序就分为解释程序和编译程序两种类型。

在解释方式中，操作系统调用某种编程语言的解释程序（如 BASIC 语言的解释程序），计算机执行该解释程序，将源程序逐段地转换为具有相同功能的指令序列。转换一段源程序后，就执行该段对应的指令序列；再转换下一段源程序，并且执行下一段指令序列，直到整个源程序被解释执行完。这是一种边翻译、边执行的工作方式，目标代码的执行始终离不开源程序和解释程序。

在编译方式中，操作系统调用的是某种编程语言的编译程序（如 C 语言的编译程序），计算机执行该编译程序，将整个源程序全部转换为指令序列，即可执行的目标程序。然后不再需要源程序和编译程序，由计算机单独执行目标程序。这是多数程序设计语言采用的处理方式：先翻译、后执行。

将用汇编语言编写的源程序转换为目标程序，这一过程称为“汇编”，也属于编译类型。用于转换的程序称为汇编程序，它和解释程序、编译程序一样，都是语言处理程序。在剖析某些重要的软件时，常需要将目标程序转换为用汇编语言表示的程序，这是汇编的逆过程，叫做“反汇编”。相应地，反汇编需要用反汇编程序来实现。

### （3）数据库管理系统

随着计算机技术在信息管理领域的广泛应用，数据管理的重要性更加突出，在这种背景下出现了数据库技术。所谓数据库，是指在计算机存储器中合理存放的、相互关联的数据集合，能够提供给不同的用户共享使用。在计算机中需要配置数据库管理系统软件（DBMS），负责数据装配、内容更新、查询检索、通信控制，以及对用数据库语言编写的程序进行翻译，控制相关的运行操作等。像 SQL Server, Sybase, ORACLE, INFORMIX 等，都是常见的 DBMS。

### （4）各种服务性支撑软件

为了帮助用户使用与维护计算机，向用户提供服务性手段而编制的一类程序，统称服务性程序。这类程序一般指输入与装配程序、编辑程序、调试程序、诊断程序、提示系统、窗口软件，以及一些可供调用的通用性应用软件，如文字处理软件、表格处理软件、图形处理软件等。这些服务程序往往作为操作系统可以调用的文件，根据需要进行配置，也可看成操作系统的可以扩充的部分。

为使用户更有效、更方便地操作计算机，现在软件开发中的一个重要趋势是将开发及运行过程中所要用到的各种软件集成为一个综合的软件系统，称为软件平台。这种软件平台技术为用户提供一个完善的集成环境，具有良好的人机界面和完善的服务支持。

### （5）各种标准程序库

这是由系统事先配置的通用、优化的标准子程序，作为库文件可供用户调用。例如许多编译程序中含有库文件，在对用户编写的源程序进行编译时，编译程序根据源程序中给出的调用名，便可调出相应的库文件进行装配。

## 2. 应用软件

应用软件直接面向用户需要，是用户在各自的应用领域中，为解决各类问题而编写的程序。计算机的应用领域极其广泛，因而应用软件几乎涉及各行各业。按照不同的应用目的，可将各种各样的应用软件大致划分为以下几种类型：

- 科学计算类程序；
- 工程设计类程序；
- 数据处理类程序；
- 信息管理类程序；
- 自动控制类程序；
- 情报检索类程序。

尽管将计算机软件划分为系统软件与应用软件两大类，但这种划分并不是一成不变的。一些具有通用价值的应用软件也可以归入系统软件的范畴，作为一种软件资源提供给用户使用。例如前面提到的数据库管理系统，是面向信息管理应用领域的，就其功能而言属于应用软件，但在计算机系统中需要事先配置，所以又是系统软件的一部分。

### 1.2.3 系统组成的层次结构

计算机系统以硬件为基础，通过配置各种软件来扩充系统功能，形成一个有机组合的复杂系统。为了对计算机系统的有机组成建立整机概念，便于对系统进行分析、设计和开发，常用一种层次结构的观点，将计算机系统从不同的角度分为若干层次。在分析计算机的工作原理时，可以根据不同的工作需要，从某一层去观察、分析计算机的组成、性能和工作机理。例如人们要了解一个计算机的硬件功能，可从该机器的指令系统级入手进行分析；了解主机对外设的控制情况，可从汇编语言级去分析设备驱动程序。在设计或构造一个计算机系统时，也常常分层次进行，如在硬件的基础上，逐级配置软件资源，逐级扩展功能。在开发应用程序时，也可以按不同层次的模块进行，如编程人员可以面向用户、面向系统分别编制不同功能的模块。

总之，按分层结构化设计策略实现的系统不仅易于建造、调试和维护，也易于扩充。根据不同的需要和目的，有多种划分层次的方法，下面主要介绍两种常见的层次结构模型。

#### 1. 从硬、软件组成角度划分的层次结构模型

这种层次结构表明一个计算机系统包括哪些硬件和软件，并且描述硬、软件之间的关系，如图 1-9 所示。层次结构模型分为 8 层，其中微程序级和逻辑部件级属于硬件部分，传统机器级可以看作硬、软件之间的界面，其他都属于软件部分。从下层向上层发展，反映了计算机系统逐级生成的过程；而从上层向下层观察，则可帮助了解应用计算机求解问题的过程。下面分别按这两个过程对层次结构模型加以说明。

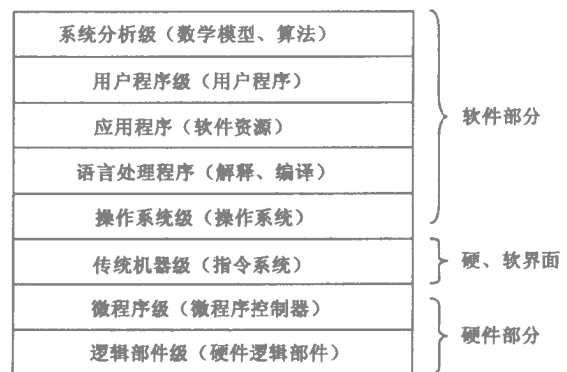


图 1-9 从组成角度划分的层次结构模型

#### (1) 自下而上，计算机系统的逐级生成过程

##### 拟定指令系统

首先规定指令系统所包含的各种基本功能，这些功能都要由硬件来实现；而各种软件最终也要转换为指令序列，才能被硬件识别和执行。所以指令系统，即传统机器级是连接硬件

和软件的界面。指令系统常用汇编语言来描述，便于人们分析和设计，但硬件最终执行的仍然是用机器语言表示的二进制代码序列。

#### 创建硬件系统

根据指令系统来设计和实现硬件系统。硬件系统也称硬核，其核心是 CPU 和主存，通过系统总线和接口将各种外存和外部设备连接起来，构成整机系统。从工作机制来看，很多计算机都采用微程序控制方式，即用微程序控制器来解释指令和执行指令，因而常常又将硬件分为两级。最下面一级是用连线连接的各种逻辑部件，称为硬连逻辑，包括寄存器与门电路；上面一级是微程序控制器，它通过执行微程序，发出各种命令控制逻辑部件的工作。

#### 配置操作系统

操作系统是系统软件的核心和基础，因此在有了硬件系统后，先要配置操作系统，然后根据硬件系统的特点改进、扩展操作系统，不断推出新的版本。例如在个人计算机上首先配置单用户操作系统 PC-DOS，后来又推出多任务操作系统 Windows。一些操作系统的核心部分源程序可以配置到多种计算机上，具有一定程度的通用性。一种计算机通常只需配置一种操作系统就可工作，但为适用于不同的应用领域，有些计算机也配置了两种或多种操作系统。

#### 配置语言处理程序及各种软件资源

根据系统需要，配置相应的语言处理程序，如某种编程语言的编译程序、解释程序或汇编程序，并且配置所需的各种软件资源。将这些软件归于操作系统的调度管理之下，形成一些通用的、或面向某种应用领域的软件平台，供用户随时调用。

#### 输入用户程序

当组成了一个硬、软件配置完备的计算机系统后，便可输入用户编写的应用程序，由计算机处理执行。

### (2) 自上而下，应用计算机求解问题的过程

#### 系统分析级

系统分析人员根据对任务的需求分析，设计算法，构建数学模型，并根据数学模型和算法进行概要设计和详细设计。

#### 用户程序级

编程人员根据详细设计，选择某种程序设计语言编写用户应用程序。

#### 操作系统级

计算机在操作系统的管理之下调用语言处理程序，如编译、解释或汇编程序，将用户源程序转换为用机器语言描述的目标程序。在源程序的输入、编辑、编译和调试过程中，通常需要调用软件开发平台所提供的各种有关的软件资源。

#### 传统机器级

所形成的目标程序是用机器语言描述的指令序列，这些能被计算机硬件识别和执行的二进制代码构成可执行文件，即可执行的目标代码。从这一级看到的程序与计算机的工作属于传统机器级，或称为机器语言级。

#### 硬件系统级

由硬件执行机器语言程序，完成指令规定的操作。一般用户所看到的计算机工作到这一级就可以了。但对于硬件设计者和维护人员来说，还需要了解硬核的工作情况，因此要深入到微程序级和逻辑部件级。对集成电路制造者而言，还要细化到电路级甚至元器件级。

## 2. 从语言功能角度划分的层次结构模型

图 1-10 是一个语言功能层次模型。这种层次划分的出发点是将计算机功能简化为执行由若干种语言编写的多个程序。

计算机硬件的物理功能是执行机器语言程序，因此相对于实际机器的这一级就称为机器语言物理机。换句话说，用户在这一级上看到的是一台实际的计算机。

与实际机器密切相关的还有汇编语言。因为它用助记符来表示指令系统的，所以常将助记符描述的指令称为汇编指令。注意，在这个层次以及后面的层次上使用了虚拟机的概念。所谓虚拟机，一般是指通过配置软件，扩充机器功能（如扩充某种语言功能）后所形成的一台计算机，而实际硬件在物理功能上并不具备这种语言功能。在汇编语言这一级，用户看到计算机能接收并执行用汇编语言编写的程序，但实际的物理机只能执行机器语言指令，它通过配置汇编程序才能处理汇编语言程序。所以用户在汇编语言这一级看到的是一台能够执行汇编语言功能的虚拟机。

与算法、数学模型甚至自然语言接近的程序设计语言称为高级语言，如 Pascal, C, C++……等通用的高级程序设计语言。用户在这一级看到的计算机是高级语言虚拟机。例如配置了 C 编译程序的计算机对用户而言，能够执行 C 语言程序，因而它是一台具有 C 语言功能的虚拟机；配置了 BASIC 解释程序的计算机对用户而言，则是一台能够执行 BASIC 语言程序的虚拟机。

对某些特殊的应用领域或特定的用户，也可使用某种专用语言。例如在数字电路设计中使用某种硬件描述语言 VHDL，描述电路中的组成与连接情况；在数据库应用中使用某种数据库语言，创建和管理数据库，等等。用户在这一级看到的是具有某种专用语言功能的虚拟机。

在图 1-10 中，各层之间有什么关系呢？我们知道，用程序设计语言编写的程序必须翻译为机器语言程序，才能被计算机理解和执行。通常，大多数编程语言是直接翻译为机器语言的，如图中实线所示，汇编语言、高级语言、专用语言都可以直接翻译为机器语言。少数高级语言也可能采用间接翻译的方式，先转换为层次较低的中间语言，再由中间语言翻译为机器语言。如图中虚线所示，专用语言先翻译为高级语言，高级语言再翻译为机器语言；或者高级语言先翻译为汇编语言，汇编语言再翻译为机器语言。

采用虚拟机概念是计算机设计中的一个重要策略。它将计算机提供给用户的功能抽象出来，脱离具体的物理机器，使用户摆脱真实机器细节的约束。例如，许多系统软件的层次结构常分为虚拟层和物理层；在虚拟层上开发的系统软件具有较强的通用性，只要改变其与物理层的接口，就能应用在不同的物理机上。

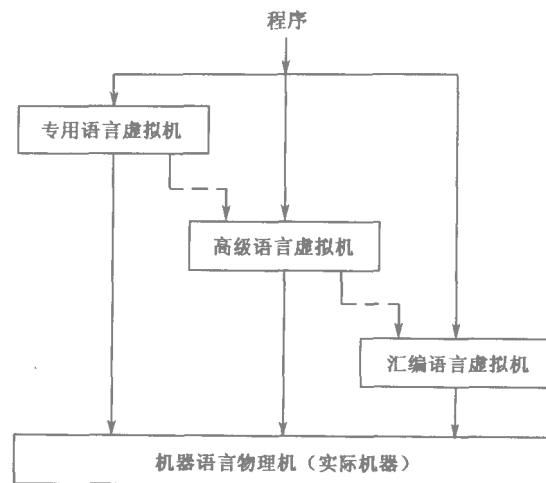


图 1-10 从语言角度划分的层次结构模型

## 1.2.4 硬件、软件的功能划分与逻辑上的等价

计算机系统以硬件为基础，通过配置软件扩充其功能，并采用执行程序的方式来体现其功能。一般来说，硬件只完成最基本的功能，复杂的功能往往通过软件来实现。硬件与软件之间的功能分配关系常常随着技术发展而变化，哪些功能分配给硬件，哪些功能分配给软件是没有固定模式的。在计算机中，实际上有许多功能既可直接由硬件实现，也可以在硬件支持下依靠软件来实现，对用户而言，在功能上是等价的。这种情况称为硬、软件在功能上的逻辑等价。例如乘法运算，可由硬件乘法器实现，也可以在加法器与移位器的支持下，通过执行乘法子程序实现。因而在用户看来，乘法器和乘法子程序在实现乘法运算功能上是没有区别的。那么，在设计一台计算机时，如何恰当地分配硬件、软件的功能？这既取决于所选定的设计目标、系统的性能价格比等因素，也与当时的技术水平有关。

早期曾采用“硬件软化”的技术策略。刚出现计算机时，各种基本功能均通过硬件来实现。随后为了降低造价，只让硬件完成较简单的指令操作，如传送、加法、减法、移位和基本逻辑运算，而乘法、除法、浮点运算等较复杂的功能则交给软件实现。这导致了在当时条件下小型计算机的出现。“硬件软化”使小型机结构简单，又具有较强的功能，推动了计算机技术的普及与应用。

随着集成电路技术的飞速发展，人们可以将功能很强的模块集成在一块芯片上，于是又出现了“软件硬化”的情况，将原来依靠软件才能实现的一些功能改由大规模或超大规模集成电路直接实现，如浮点运算、存储管理等。这就使系统具有更高的处理速度，在软件支持下有更强的功能。

微程序控制技术的出现使计算机结构和硬、软件功能分配发生了变化，对指令的解释与执行是通过运行微程序来实现的。因而又出现了另一种技术策略“软件固化”。利用程序设计和扩大微程序的容量，可使原来属于软件级的一些功能纳入微程序一级。微程序类似于软件，但被固化在只读存储器中，属于硬件 CPU 的范畴，称为固件。这种方式使 CPU 的结构得到简化。人们也常采用软件固化的策略，将系统软件的核心部分（如操作系统的内核、常用软件中固定不变的部分）固化在存储芯片中。从用户的角度看，它们属于系统硬件（如系统板）的一部分。例如 IBM-PC 微机系统将操作系统中的基本输入/输出系统 BIOS 固化在系统板上，Pentium 微处理器将存储管理功能集成于 CPU 芯片之内，等等。

系统设计者必须关心硬、软件之间的界面划分，决定系统功能哪些由硬件实现，哪些由软件实现。而用户则更关心系统究竟能提供哪些功能，至于这些功能是由硬件实现还是软件实现，在逻辑功能上是等价的，只是执行速度不同而已。

尽管常按层次结构来分析、设计计算机系统，但考虑到初学者的认识过程，本书不准备机械地分级讨论计算机组成原理，而是以讨论计算机各组成部件原理及整机连接为基线，即分别讨论 CPU 组织、存储器组织、外围设备组织、以总线和接口为基础的 I/O 系统组织。在分析这些计算机组织时，力求注意它们的硬、软件功能分配与界面关系，抽象出相应的概念模型，从寄存器级、微操作控制级、机器指令级等几个层次去观察分析计算机的硬件组成与工作机制。

例如分析 CPU、存储器、接口、I/O 设备中有哪些寄存器，以及与寄存器相当的部件，如加法器、输入/输出通道、选择器等，了解它们通过什么样的数据通路结构相连接。从寄存器级进行描述，容易获得清晰的结构概念。

CPU 的功能特性体现为指令系统，而硬件的任务就是执行指令，实现寄存器级的信息传送。为此，可先将指令流程分解为若干步寄存器级传送操作，再从微操作控制这一级产生微操作命令序列，控制完成寄存器级的传送操作。微操作命令的产生既可通过组合逻辑控制方式实现，也可通过微程序控制方式实现。

## 1.3 计算机的特点与性能指标

### 1.3.1 数字计算机的特点

由于数字计算机采用数字化的信息表示方法和存储程序工作方式，因而能够自动连续地对各种数字化信息进行算术、逻辑运算，并且能进行广泛的信息处理。它的主要特点表现在以下 5 个方面。

#### 1. 自动连续地执行程序

将程序输入计算机后，只要提供程序的运行条件和起始地址，计算机便能自动地从起始地址读取程序，解释并且执行程序。这是计算机区别于其他计算工具的最本质的特点。

#### 2. 运算速度快

计算机硬件目前是由高速的电子线路组成的，工作速度极快。一个复杂的数学计算或大量的数据处理可能需要若干人工作很长一段时间才能完成，而用计算机来处理很快就可获得结果。这不仅极大地提高了人类的工作效率，也大大增强了人类处理问题的能力，使许多复杂问题得到解决。随着更高速的新器件的诞生，以及系统结构的进一步优化，计算机的运算速度还将得到更大的提高。

#### 3. 运算精度高

计算机采用数字代码表示数据，代码的位数越多，数据的表示精度就可以越高。尽管在实际计算机中考虑到硬件成本等因素，对数据的基本位数有一定限制，但通过软件可以实现多位数据的运算，获得更高的精度。

#### 4. 存储能力强

计算机依靠存储器能够存储大量的程序和数据，这是保证计算机自动连续工作的先决条件。程序和数据是由二进制代码组成的，只要具有两种稳定状态的物理介质都能存储二进制代码。在计算机中，主存利用触发器或电容电荷存储信息，外存利用磁化状态或其他介质状态存储信息。因而计算机具有很强的信息存储能力，存储的程序和数据越多，计算机的处理功能也就越强。

#### 5. 通用性好

由于各类信息都可表示为数字化信息，都能被计算机处理，所以计算机的应用领域极其广泛。由于计算机用数字逻辑部件作为处理数字信号的统一逻辑基础，因此计算机既能实现算术运算，又能实现逻辑运算；既能进行数值计算，又能对各类非数值信息进行处理，如信息检索、图像处理、语音处理、逻辑判断，等等。计算机的这种极好的通用性使它能够应用到各行各业，渗透到人们的工作、学习、生活等各个方面。