

第 1 章 概 论

在本章中我们将从存储程序的概念入手，讨论计算机的基本组成与工作原理，使读者对于计算机系统先有一个简单的整体概念，为今后深入讨论各个部件打下基础。

1.1 电子计算机与存储程序控制

电子计算机是一种不需要人工直接干预，能够自动、高速、准确地对各种信息进行处理和存储的电子设备。电子计算机从总体上来说可以分为两大类：电子模拟计算机和电子数字计算机。电子模拟计算机中处理的信息是连续变化的物理量，运算的过程也是连续的；而电子数字计算机中处理的信息是在时间上离散的数字量，运算的过程是不连续的。我们通常所说的计算机都是指电子数字计算机。

1.1.1 电子计算机的发展

一、计算机的发展历史

人们习惯把电子计算机的发展历史分“代”，其实分代并没有统一的标准。若按计算机所采用的微电子器件的发展，可以将电子计算机分成以下几代：

第一代：1946—1959 年，电子管计算机

第二代：1959—1964 年，晶体管计算机

第三代：1964—1975 年，小、中规模集成电路（SSI, MSI）计算机

第四代：1975—1990 年，大、超大规模集成电路（LSI, VLSI）计算机

第五代：1990—200? 年，甚大规模集成电路（ULSI）计算机

第六代：200?—20?? 年，极大规模集成电路（ELSI）计算机

1. 电子管计算机时代

这一时期的计算机采用电子管作为基本器件，初期使用延迟线作为存储器，以后发明了磁芯存储器。早期的计算机主要用于科学计算，为军事与国防尖端科技服务。

2. 晶体管计算机时代

这一时期计算机的基本器件由电子管改为晶体管，存储器采用磁芯存储器。运算速度从每秒几千次提高到几十万次，存储器的容量从几千存储单元提高到 10 万存储单元以上。这不仅使计算机在军事与尖端技术上的应用范围进一步扩大，而且在气象、工程设计、数据处理以及其他科学研究等领域内也应用起来。

3. 小、中规模集成电路计算机时代

这一时期的计算机采用小、中规模集成电路为基本器件，因此功耗、体积、价格等进一步下降，而速度及可靠性相应的提高，使得计算机的应用范围进一步扩大。

4. 大、超大规模集成电路计算机时代

20 世纪 60 年代后，微电子技术发展迅猛，半导体存储器问世，迅速取代了磁芯存储器，

并不断向大容量、高集成度、高速度发展。

从 1973 年开始出现了包含 CPU 的单片 IC (微处理器) 以微处理器为核心的电子计算机就是微型计算机, 简称微机。微机的出现, 形成了计算机发展史上的又一次革命, 使计算机进入了几乎所有的行业。

5. 甚大规模集成电路计算机时代

20 世纪 90 年代初, 计算机进入第五代。其主要标志有两个: 一个是单片集成电路规模达 100 万晶体管以上, 另一个是超标量技术的成熟并被广泛应用。

6. 极大规模集成电路计算机时代

第六代计算机可能是极大规模集成电路计算机, 单片集成电路规模可达一亿到十亿晶体管。

二、计算机的发展趋势

现在, 世界已进入了计算机时代, 计算机的发展趋势正向着“两极”分化。一极是微型机向更微型化、网络化、高性能、多用途方向发展。微型机分为台式机、便携机、笔记本、亚笔记本、掌上机等。由于它们体积小、成本低而占领了整个国民经济和社会生活的各个领域。另一极则是巨型机向更巨型化、超高速、并行处理、智能化方向发展。它是一个国家科技水平、经济实力、军事威力的象征。在解决天气预报、地震分析、航空气动、流体力学、卫星遥感、激光武器、海洋工程等方面的问题上, 巨型机将大显身手。

随着新的元器件及其技术的发展, 新型的超导计算机、量子计算机、光子计算机、生物计算机、纳米计算机等将会在 21 世纪走进我们的生活, 遍布各个领域。

2 存储程序的概念

世界上第一台电子数字计算机是 1946 年 2 月 15 日在美国宾夕法尼亚大学诞生的 ENIAC (Electronic Numerical Integrator And Computer)。ENIAC 是一个庞然大物, 它共用了 18 000 多个电子管, 1 500 个继电器, 重达 30t, 占地面积 170m², 耗电 160kW, 其运算速度为 5 000 次/s 左右。ENIAC 有一个很大的缺点, 即它的存储容量极小, 只能存 20 个字长为 10 位的十进制数, 所以只能用线路连接的方法来编排程序, 每次解题都要依靠人工来改变接线, 准备时间大大超过了实际计算时间。

在研制 ENIAC 的同时, 以美籍匈牙利数学家冯·诺依曼 (John Von Neumann) 为首的研制小组提出了“存储程序控制”的计算机结构, 并开始了存储程序控制的计算机 EDVAC (Electronic Discrete Variable Automatic Computer) 的研制。由于种种原因, EDVAC 直到 1951 年才问世。而吸收了冯·诺依曼的设计思想, 由英国剑桥大学研制的 EDSAC (Electronic Delay Storage Automatic Computer) 则先于它两年诞生, 成为事实上的第一台存储程序的计算机。

存储程序概念是冯·诺依曼等人在 1946 年 6 月首先提出来的, 它可以简要地概括为以下几点:

(1) 计算机 (指硬件) 应由运算器、存储器、控制器、输入设备和输出设备五大基本部件组成;

(2) 计算机内部采用二进制来表示指令和数据;

(3) 将编好的程序和原始数据事先存入存储器中, 然后再启动计算机工作, 这就是存储程序的基本含义。

冯·诺依曼对计算机界的最大贡献在于“存储程序控制”概念的提出和实现。50多年来，虽然计算机的发展速度是惊人的，但就其结构原理来说，目前绝大多数计算机仍建立在存储程序概念的基础上。我们把符合存储程序概念的计算机统称为冯·诺依曼型计算机。当然，现代计算机与早期计算机相比在结构上还是有许多改进的。

随着计算机技术的不断发展，也暴露出了冯·诺依曼型计算机的一些缺点。目前已出现了一些突破冯·诺依曼结构的计算机，统称非冯结构计算机，如数据驱动的数据流计算机、需求驱动的归约计算机和模式匹配驱动的智能计算机等。本书讨论的范围仅限于冯·诺依曼结构计算机的组成，有关非冯结构的计算机将在后继的课程中进行讨论。

1.2 计算机的硬件组成

原始的冯·诺依曼计算机在结构上是以运算器为中心的，而发展到现在，已转向以存储器为中心了。图 1-1 为计算机最基本的组成框图。

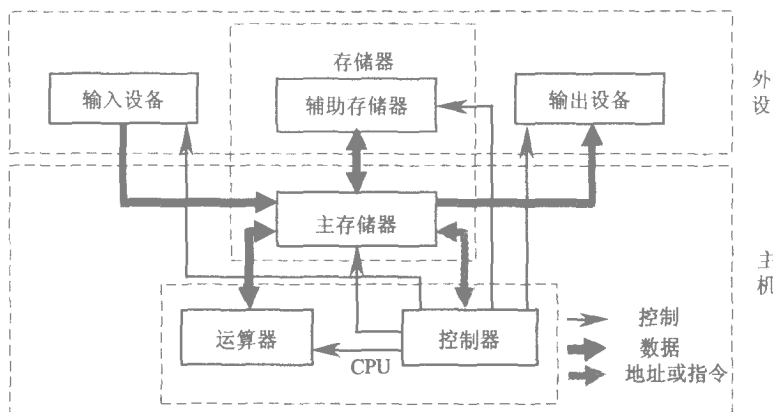


图 1-1 计算机的组成框图

通常将运算器和控制器合称为中央处理器 CPU (Central Processing Unit)。在由超大规模集成电路构成的微型计算机中，往往将 CPU 制成一块芯片，称为微处理器。

中央处理器和主存储器（内存储器）一起组成主机部分。除去主机以外的硬件装置（如输入设备、输出设备、辅助存储器等）称为外围设备或外部设备。

1.2.1 计算机的主要部件

1. 输入设备

输入设备的任务是把人们编好的程序和原始数据送到计算机中去，并且将它们转换成计算机内部所能识别和接收的信息方式。

按输入信息的形态可分为字符（包括汉字）输入、图形输入、图像输入及语音输入等。目前，常见的输入设备有键盘、鼠标、扫描仪等。辅助存储器（磁盘、磁带）也可以看做输入设备。另外，自动控制和检测系统中使用的模/数 (A/D) 转换装置也是一种输入设备。

2. 输出设备

输出设备的任务是将计算机的处理结果以人或其他设备所能接受的形式送出计算机。

目前最常用的输出设备是打印机和显示器。辅助存储器也可以看做输出设备。另外数/模(D/A)转换装置也是一种输出设备。

3. 存储器

存储器是用来存放程序和数据的部件，它是一个记忆装置，也是计算机能够实现“存储程序控制”的基础。

在计算机系统中，规模较大的存储器往往分成若干级，称为存储系统。图 1-2 所示的是常见的三级存储系统。主存储器可由 CPU 直接访问，存取速度快但容量较小，一般用来存放当前正在执行的程序和数据。辅助存储器设置在主机外部，它的存储容量大，价格较低，但存取速度较慢，一般用来存放暂时不参与运行的程序和数据，这些程序和数据在需要时可传送到主存，因此它是主存的补充和后援。当 CPU 速度很高时，为了使访问存储器的速度能与 CPU 的速度相匹配，又在主存和 CPU 间增设了一级 Cache（高速缓冲存储器）。Cache 的存取速度比主存更快，但容量更小，用来存放当前最急需处理的程序和数据，以便快速地向 CPU 提供指令和数据。

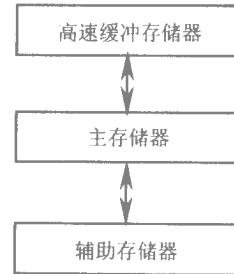


图 1-2 三级存储系统

4. 运算器

运算器是对信息进行处理和运算的部件。经常进行的运算是算术运算和逻辑运算，所以运算器又称为算术逻辑运算部件（Arithmetic and Logical Unit, ALU）。

运算器的核心是加法器。运算器中还有若干个通用寄存器或累加寄存器，用来暂存操作数，并存放运算结果。寄存器的存取速度比存储器的存取速度快得多。

5. 控制器

控制器是整个计算机的指挥中心，它的主要功能是按照人们预先确定的操作步骤，控制整个计算机的各部件有条不紊地自动工作。

控制器从主存中逐条地取出指令进行分析，根据指令的不同来安排操作顺序，向各部件发出相应的操作信号，控制它们执行指令所规定的任务。

控制器中包括一些专用的寄存器。

2.2 计算机的总线结构（小型、微型机的典型结构）

将前述的各大基本部件，按某种方式连接起来就构成了计算机的硬件系统。在各大基本部件之间，有着大量的信息需要交换、传送，例如：程序和数据要经过输入设备送入存储器；CPU 要发送存储单元地址和读命令，以便从存储器中读取指令和数据；存储器要把数据传送给运算器去加工、处理；指令执行的结果又可能要返回来写入存储器；全部程序的最后执行结果由存储器传送到输出设备等等。早期的计算机往往是在各个部件间直接地传送，数据通路结构复杂而凌乱，控制很不方便，且没有多少扩充的余地，故这种方式现在已不再采用。目前许多计算机（主要指小型、微型计算机）的各大基本部件之间是用总线（Bus）连接起来的。

所谓总线是一组能为多个部件服务的公共信息传送线路，它能分时地发送与接收各部件的信息。计算机中采用总线结构，既可以大大减少信息传送线的数目，又可以提高计算机扩

充主存及外部设备的灵活性。

最简单的总线结构是单总线结构，如图 1-3 所示。各大部件都连接在单一的一组总线上，故将这个单总线称为系统总线。CPU 与主存、CPU 与外设之间可以直接进行信息交换，主存与外设、外设与外设之间也可以直接进行信息交换，而无须经过 CPU 的干预。

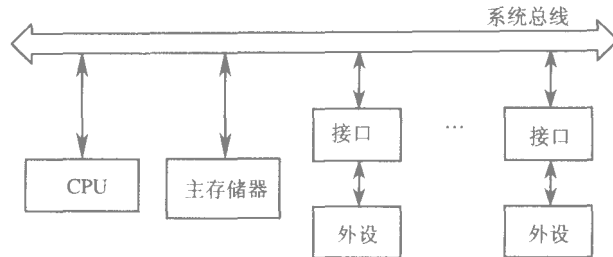


图 1-3 单总线结构

单总线结构提高了 CPU 的工作效率，而且外设连接灵活，易于扩充。但由于所有部件都挂在同一组总线上，而总线又只能分时工作，故同一时刻只允许一对设备（或部件）之间传送信息。

所谓单总线并不是指只有一根信号线。系统总线按传送信息的不同又可以细分为地址总线、数据总线和控制总线。地址总线（Address Bus）由单方向的多根信号线组成，用于 CPU 向主存、外设传输地址信息；数据总线（Data Bus）由双方向的多根信号线组成，CPU 可以沿这些线从主存或外设读入数据，也可以沿这些线向主存或外设送出数据；控制总线（Control Bus）上传输的是控制信息，包括 CPU 送出的控制命令和主存/外设反馈给 CPU 的状态信号。

总线结构是小型、微型机的典型结构。这是因为小型、微型机的设计目标是以较小的硬件代价组成具有较强功能的系统，而总线结构正好能满足这一要求。

2.3 大型、中型机的典型结构

大型、中型计算机系统的设计目标更着重于系统功能的扩大与效率的提高。图 1-4 为大型、中型计算机的典型结构。在系统连接上分为四级：主机、通道、设备控制器和外部设备。

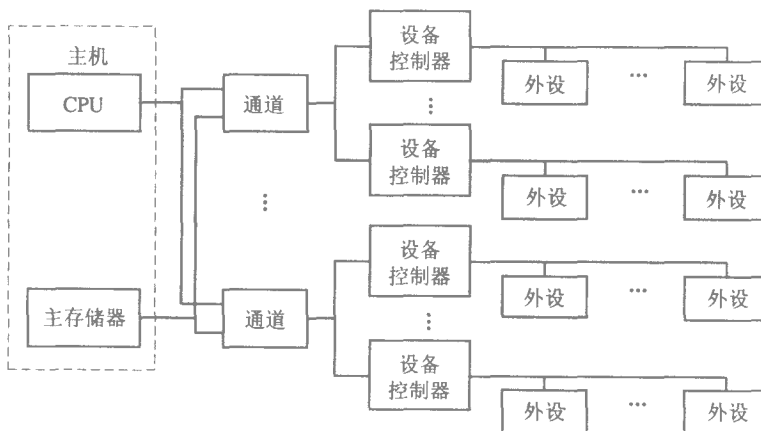


图 1-4 大型、中型计算机系统的典型结构

通道是承担 I/O 操作管理的主要部件，能使 CPU 的数据处理和与外部设备交换信息这两项操作同时进行。每个通道可以接一台或几台设备控制器，每个设备控制器又可接一台或几台外部设备，这样整个系统就可以连接很多的外部设备。这种结构具有较大的扩充变化余地。对较小的系统来说，可将设备控制器与外设合并在一起，将通道与 CPU 合并在一起；对较大的系统，则单独设置通道部件；对更大的系统，通道可发展成为具有处理功能的外围处理机。

1.2.4 不同对象观察到的计算机硬件系统

从第一台计算机诞生到至今，已经经历了半个多世纪的风风雨雨，目前计算机可以说已经无所不在、无事不通。在绝大多数人看来，计算机（主要指微型计算机）已不再是陌生和充满玄机的神秘之物，而仅仅是一种常用工具而已。然而不同对象所观察到的计算机硬件系统是不相同的，所涉及的技术问题也是不同层次的，下面仅以微型计算机为例来分析它们之间的区别。

一、一般用户观察到的计算机硬件系统

一般用户是指那些仅局限于使用计算机最基本用途的用户。他们观察到的只是计算机的用户界面，如人机交互用的键盘、鼠标、显示器，用于存储信息的磁盘、光盘等，计算机本身对于他们来说只是一个或立或卧在那里的铁箱子，至于它内部的结构和组成、工作原理等都是不必关心的，也就是说完全透明的。一般用户观察到的计算机硬件系统如图 1-5 所示。

二、专业用户观察到的计算机硬件系统

专业用户观察到的计算机硬件系统要比一般用户深入许多，他们可能更多地关注计算机机箱内的结构和组成，专业用户观察到的计算机硬件系统如图 1-6 所示。



图 1-5 一般用户观察到的计算机硬件系统

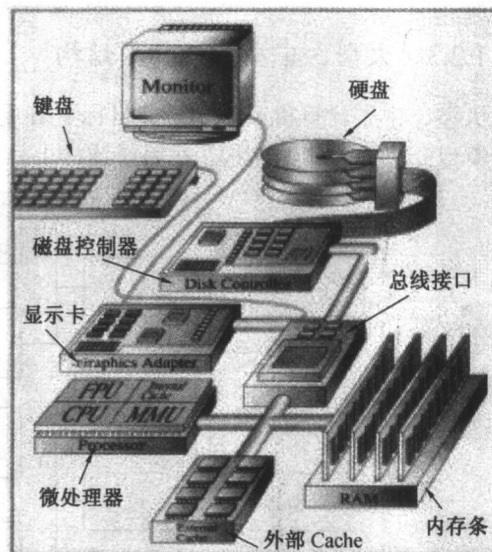


图 1-6 专业用户观察到的计算机硬件系统

图 1-6 中除键盘、显示器以外，其余部分都在机箱内。机箱中的核心是主机板，微处理

器、内存条、外部高速缓冲存储器(Cache)、显示卡、磁盘控制器等可以直接制作在主板上或插在主板上。

三、计算机设计者观察到的计算机硬件系统

计算机的设计者更多地关心计算机的组成原理和实现方法，图 1-7 是计算机设计者观察到的计算机硬件系统，此时计算机的硬件系统被进一步分解。

从图 1-7 可以看出，计算机硬件系统是由运算器、控制器、存储器和外部设备组成的，我们称为系统级。系统级是由寄存器级组成的，寄存器级又是由门级组成的，门级最后由晶体管级组成，而晶体管的物理实现主要有两种：金属氧化物半导体（MOS）和晶体管-晶体管逻辑（TTL）。

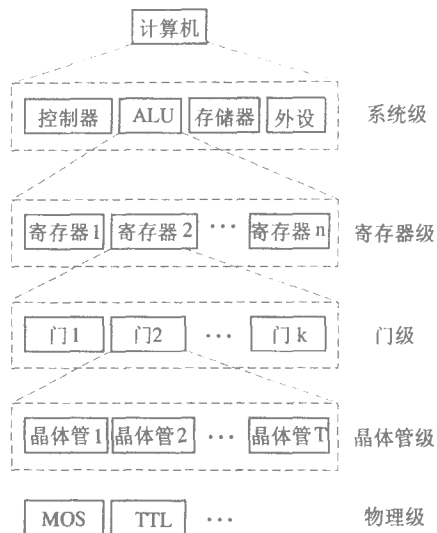


图 1-7 计算机设计者观察到的计算机硬件系统

1.3 计算机系统

一个完整的计算机系统包含硬件系统和软件系统两大部分。硬件通常是指一切看得见、摸得到的设备实体；软件通常是泛指各类程序和文件，它们实际上是由一些算法以及其在计算机中的表示所构成的。

1.3.1 硬件与软件的关系

硬件是计算机系统的物质基础，正是在硬件高度发展的基础上，才有软件赖以生存的空间和活动场所。没有硬件对软件的支持，软件的功能就无从谈起；同样，软件是计算机系统的灵魂，没有软件的硬件“裸机”将犹如一堆废铁，不能提供给用户使用。因此，硬件和软件是相辅相成的、不可分割的整体。

当前，计算机的硬件和软件正朝着互相渗透、互相融合的方向发展，在计算机系统中没有一条明确的硬件与软件的分界线。原来一些由硬件实现的功能可以改由软件模拟来实现，这种做法称为硬件软化，它可以增强系统的功能和适应性；同样，原来由软件实现的功能也可以改由硬件来实现，称为软件硬化，它可以显著降低软件在时间上的开销。由此可见，硬件和软件之间的界面是浮动的，对于程序设计人员来说，硬件和软件在逻辑上是等价的。一项功能究竟采用何种方式实现，应从系统的效率、速度、价格和资源状况等诸多方面综合考虑。

既然硬件和软件不存在一条固定的一成不变的界限，那么今天的软件可能就是明天的硬件，今天的硬件也可能就是明天的软件。

除去硬件和软件以外，还有一个概念需要引起大家的注意，这就是固件（Firmware）。固件一词是在 1967 年由美国人 A·Opler 首先提出来的。固件是指那些存储在能永久保存信息

的器件（如 ROM）中的程序，是具有软件功能的硬件。固件的性能指标介于硬件与软件之间，吸收了软、硬件各自的优点，其执行速度快于软件，灵活性优于硬件，是软、硬件结合的产物，计算机功能的固件化将成为计算机发展中的一个趋势。

3.2 系列机和软件兼容

计算机技术是飞速发展的技术，随着元器件、硬件技术和工业生产能力的迅猛发展，新的高性能的计算机不断地被研制和生产出来。用户希望在新的计算机系统推出后，原先已开发的软件仍能继续在升级换代后的新型号的机器上使用，这就要求软件具有可兼容性。

所谓系列机是指一个厂家生产的、具有相同的系统结构但具有不同组成和实现的一系列不同型号的机器。

系列机从程序设计者的角度看具有相同的机器属性，即相同的系统结构。这里的相同是指在指令系统、数据格式、字符编码、中断系统、控制方式和输入/输出操作方式等多个方面保持统一，从而保证了软件的兼容。系列机的软件兼容分为向上兼容、向下兼容、向前兼容和向后兼容四种。向上（下）兼容指的是按某档次机器编制的程序，不加修改就能运行在比它更高（低）档的机器上；向前（后）兼容是指按某个时期投入市场的某种型号机器编制的程序，不加修改就能运行在它之前（后）投入市场的机器上。图 1-8 形象地说明了兼容性的概念。对系列机的软件向下和向前兼容可以不作要求，但必须保证向后兼容，力争做到向上兼容。

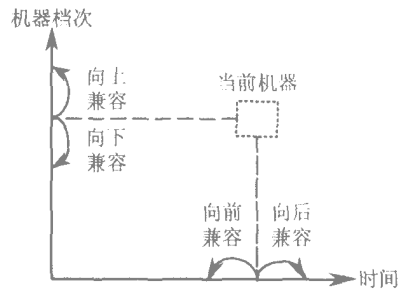


图 1-8 兼容性示意图

3.3 计算机系统的多层次结构

现代计算机系统是一个硬件与软件组成的综合体，我们可以把它看成是按功能划分的多级层次结构，如图 1-9 所示。

第 0 级为硬件组成的实体。

第 1 级是微程序级。这级的机器语言是微指令集，用微指令编写的微程序一般是直接由硬件执行的。

第 2 级是传统机器级。这级的机器语言是该机的指令集，用机器指令编写的程序可以由微程序进行解释。

第 3 级是操作系统级。从操作系统的基本功能来看，一方面它要直接管理传统机器中的软硬件资源，另一方面它又是传统机器的延伸。

第 4 级是汇编语言级。这级的机器语言是汇编语言，完成汇编语言翻译的程序叫做汇编程序。

第 5 级是高级语言级。这级的机器语言就是各种高级语言，通常用编译程序来完成高级语言翻译的工作。

第 6 级是应用语言级。这一级是为了使计算机满足某种用途而专门设计的，因此这一级语言就是各种面向问题的应用语言。

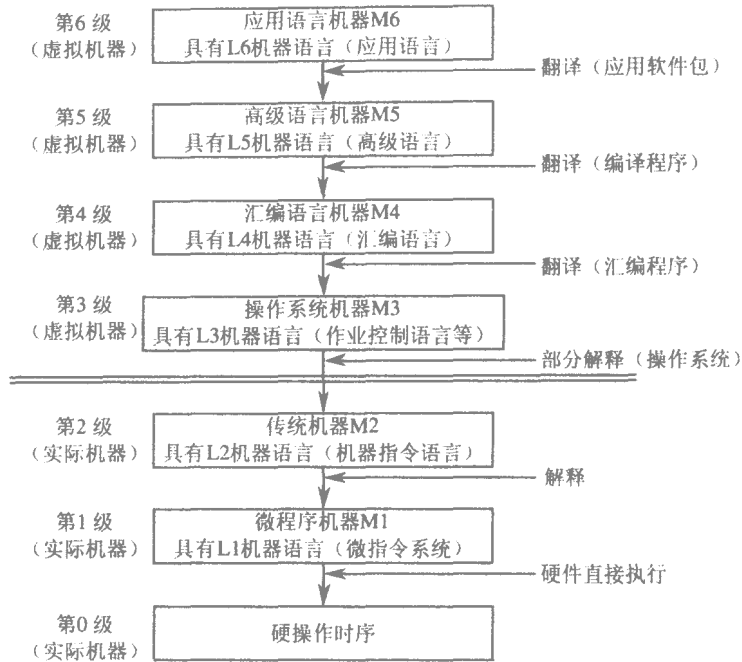


图 1-9 计算机系统的多层次结构

把计算机系统按功能划分成多级层次结构，有利于正确理解计算机系统的工作过程，明确软件、硬件在计算机系统中的地位和作用。

1.3.4 实际机器和虚拟机器

在图 1-9 的多级层次结构中，对每一个机器级的用户来说，都可以将此机器级看成是一台独立的、使用自己特有的“机器语言”的机器。

实际机器是指由硬件或固件实现的机器，如图 1-9 中的第 0~第 2 级。虚拟机器是指以软件或以软件为主实现的机器，如图 1-9 中的第 3~第 6 级。

虚拟机器只对该级的观察者存在，即在某一级观察者看来，他只需要通过该级的语言来了解和使用计算机，至于下级是如何工作和实现就不必关心了。如高级语言级及应用语言级的用户，可以不了解机器的具体组成，不必熟悉指令系统，直接用所指定的语言描述所要解决的问题。

1.4 计算机的工作过程和主要性能指标

为使计算机按预定要求工作，首先要编制程序。程序是一个特定的指令序列，它告诉计算机要做哪些事，按什么步骤去做。指令是一组二进制信息的代码，用来表示计算机所能完成的基本操作。衡量一台计算机的性能是由多项技术指标综合确定的，不能片面强调某一项指标。

1.4.1 计算机的工作过程

编制好的程序放在主存中，由控制器控制逐条取出指令执行，下面以一个例子来加以说明

例如：计算 $a+b-c=?$ （设 a 、 b 、 c 为已知的三个数，分别存放在主存的 5~7 号单元中，结果将存放在主存的 8 号单元），如果采用单累加寄存器结构的运算器，完成上述计算至少需要五条指令，这五条指令依次存放在主存的 0~4 号单元中，参加运算的数也必须存放在主存指定的单元中，主存中有关单元的内容如图 1-10 (a) 所示。运算器的简单框图如图 1-10 (b) 所示，参加运算的两个操作数一个来自累加寄存器，一个来自主存，运算结果则放在累加寄存器中。图 1-10 (b) 中的存储器数据寄存器是用来暂存从主存中读出的数据或写入主存的数据的，它本身不属于运算器的范畴。

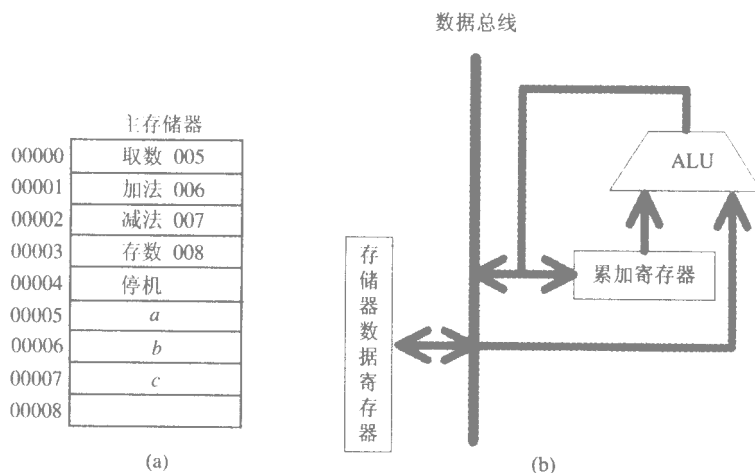


图 1-10 计算机执行过程实例

计算机的控制器将控制指令逐条的执行，最终得到正确的结果。

第一步，执行取数指令，从主存 5 号单元取出数 a ，送入累加寄存器中；

第二步，执行加法指令，将累加寄存器中的内容 a 与从主存 6 号单元取出数 b 一起送到 ALU 中相加，结果 $a+b$ 保留在累加寄存器中；

第三步，执行减法指令，将累加寄存器中的内容 $a+b$ 与从主存 7 号单元取出的数 c 一起送到 ALU 中相减，结果 $a+b-c$ 保留在累加寄存器中；

第四步，执行存数指令，把累加寄存器的内容 $a+b-c$ 存至主存 8 号单元；

第五步，执行停机指令，计算机停止工作。

1.4.2 计算机的主要性能指标

为了进一步表征计算机的特性，全面衡量一台计算机的性能，下面介绍计算机的主要性能指标。

1. 机器字长

机器字长是指参与运算的数的基本位数，它是由加法器、寄存器的位数决定的，所以机器字长一般等于内部寄存器的大小。字长标志着精度，字长越长，计算的精度就越高。倘若字长较短，又要计算位数较多的数据，那么需要经过两次或多次的运算才能完成，这样势必影响整机的运行速度。

在计算机中为了更灵活地表达和处理信息，又以字节（byte）为基本单位，用大写字母 B 表示。一个字节等于 8 位二进制位（bit）。

不同的计算机，字（word）的长度可以不相同，但对于系列机来说，在同一系列中，字长却是固定的，如 80X86 系列中，一个字等于 16 位；IBM303X 系列中，一个字等于 32 位。

2. 数据通路宽度

数据总线一次所能并行传送信息的位数，称为数据通路宽度。它影响到信息的传送能力，从而影响计算机的有效处理速度。这里所说的数据通路宽度是指外部数据总线的宽度，它与 CPU 内部的数据总线宽度（内部寄存器的大小）有可能不同。有些 CPU 的内、外数据总线宽度相等，例如 Intel 8086、80286、80486 等；有些 CPU 的外部数据总线宽度小于内部，例如 8088、80386SX 等；也有些 CPU 的外部数据总线宽度大于内部，例如 Pentium 等。所有的 Pentium 都有 64 位外部数据总线和 32 位内部寄存器。这一结构看起来似乎有问题，但这是因为 Pentium 有两条 32 位流水线，它就像两个合在一起的 32 位芯片，64 位数据总线可以满足高效地充满多个寄存器的需要。

3. 主存容量

一个主存储器所能存储的全部信息量称为主存容量。通常，以字节数来表示存储容量，这样的计算机称为字节编址的计算机，也有一些计算机是以字为单位编址的，它们用字数乘以字长来表示存储容量。在表示容量大小时，我们经常用到 K、M、G、T、P 之类的字符，它们与通常意义下的 K、M、G、T、P 有些差异，如表 1-1 所示。

表 1-1 K、M、G、T、P 的定义

单位	通常意义	实际表示
K (Kilo)	10^3	$2^{10}=1\ 024$
M (Mega)	10^6	$2^{20}=1\ 048\ 576$
G (Giga)	10^9	$2^{30}=1\ 073\ 741\ 824$
T (Tera)	10^{12}	$2^{40}=1\ 099\ 511\ 627\ 776$
P (Peta)	10^{15}	$2^{50}=1\ 125\ 899\ 906\ 842\ 624$

每 1 024 个字节称为 1 KB，每 1 024KB 称为 1 MB，每 1 024MB 称为 1 GB，……计算机的主存容量越大，存放的信息就越多，处理问题的能力就越强。

4. 运算速度

计算机的运算速度与许多因素有关，如机器的主频、执行什么样的操作、主存本身的速度等。对运算速度的衡量有不同的方法。

(1) 根据不同类型指令在计算过程中出现的频繁程度，乘上不同的系数，求得统计平均值，这时所指的运算速度是平均运算速度。

(2) 以每条指令执有所需时钟周期数 CPI (Cycles Per Instruction) 来衡量运算速度。

(3) 以 MIPS 和 MFLOPS 作为计量单位来衡量运算速度

MIPS (Million Instructions Per Second) 表示每秒执行多少百万条指令。对于一个给定的程序, MIPS 定义为

$$\text{MIPS} = \frac{\text{指令条数}}{\text{执行时间} \times 10^6}$$

这里所说的指令一般是指加、减运算这类短指令。

MFLOPS (Million Floating-point Operations Per Second) 表示每秒执行多少百万次浮点运算。对于一个给定的程序, MFLOPS 定义为

$$\text{MFLOPS} = \frac{\text{浮点操作次数}}{\text{执行时间} \times 10^6}$$

衡量一台计算机的性能指标还有很多,如系统吞吐量、外部设备配置、系统软件配置,还有可靠性、可用性、可维护性、完整性、安全性和兼容性等,在此不一一说明。值得一提的是,衡量一台计算机的优劣,不能片面强调某一项指标,而需要综合、全面考虑。通常以性能价格比作为综合性指标,它是上述各项性能与售价的比值,性能价格比越高的计算机就越受到用户的欢迎。

习 题

- 1-1 电子数字计算机和电子模拟计算机的区别在哪里?
- 1-2 冯·诺依曼计算机的特点是什么?其中最主要的一点是什么?
- 1-3 计算机的硬件是由哪些部件组成的?它们各有哪些功能?
- 1-4 什么叫总线?简述单总线结构的特点。
- 1-5 简单描述计算机的层次结构,说明各层次的主要特点。
- 1-6 衡量计算机性能的主要技术指标有哪些?

第 2 章 计算机中数据信息的表示

数据信息是计算机加工和处理的对象，数据信息的表示法将直接影响到计算机的结构和性能本章主要介绍进位计数制、数值数据的编码、数的定点与浮点表示方法、字符和汉字的编码方法、数据校验码等。熟悉并掌握本章的内容，是学习计算机的最基本要求。

2.1 进位计数制

在人类的生产实践和日常生活中，总要遇到数的表示问题。人们创造了许多种表示数的方法，这些表示数的方法统称为进位计数制。

2.1.1 进位计数制的基本概念

讨论进位计数制首先要涉及两个基本问题，即基数（Radix）和权（Weight）。

在进位计数制中，每个数位所用到的不同数码的个数叫做基数。十进制是人们最熟悉的一种进位计数制，每个数位允许选用 0~9 共 10 个不同的数码中的某一个，因此十进制的基数为 10；每个数位计满 10 就向高位进位，即逢十进一。

在一个数中，数码在不同的数位上所表示的数值是不同的。每个数码所表示的数值就等于该数码本身乘以一个与它所在数位有关的常数，这个常数叫做位权，简称权。例如：十进制数 4952，数码“4”表示 4000，该位的权为 1000，这一位所代表的数值等于数码 4 乘以常数 1000，即 4×10^3 ，“9”表示 900，该位的权为 100，这一位所代表的数值等于数码 9 乘以常数 100，即 9×10^2

所以，一个数的数值大小就是它的各位数码按权相加，如：

$$4952 = 4 \times 10^3 + 9 \times 10^2 + 5 \times 10^1 + 2 \times 10^0$$

由此可见，任何一个十进制数都可以用一个多项式来表示：

$$\begin{aligned} (N)_{10} &= K_n \times 10^n + K_{n-1} \times 10^{n-1} + \cdots + K_0 \times 10^0 + K_{-1} \times 10^{-1} + \cdots + K_{-m} \times 10^{-m} \\ &= \sum_{i=n}^{-m} K_i \times 10^i \end{aligned}$$

式中： K_i 的取值是 0~9 中的一个数码， m 和 n 为正整数。

推广来看，一个基数为 r 的 r 进制数可表示为：

$$\begin{aligned} (N)_r &= K_n \times r^n + K_{n-1} \times r^{n-1} + \cdots + K_0 \times r^0 + K_{-1} \times r^{-1} + \cdots + K_{-m} \times r^{-m} \\ &= \sum_{i=n}^{-m} K_i \times r^i \end{aligned}$$

式中： r^i 为第 i 位的权， K_i 的取值可以是 0, 1, ..., $r-1$ 共 r 个数码中的任意一个。

r 进制数的进位原则是：逢 r 进一。

2.1.2 计算机中常用的进位计数制

人们在日常生活中最常用的是十进制，这恐怕和人有十个手指头是分不开的。然而，在计算机中常用的却是二进制、八进制和十六进制。

一、二进制表示

二进制是一种最简单的进位计数制，它只有两个不同的数码：“0”和“1”。即基数为 2 逢二进一。任意数位的权是 2^i 。

因此，任何一个二进制数都可表示为：

$$(N)_2 = K_n \times 2^n + K_{n-1} \times 2^{n-1} + \cdots + K_0 \times 2^0 + K_{-1} \times 2^{-1} + \cdots + K_{-m} \times 2^{-m}$$

$$= \sum_{i=n}^{-m} K_i \times 2^i$$

表 2-1 列举了常用的二进制数的权值，请读者自行熟记。

表 2-1 常用的二进制数的权值

权	权值	权	权值
2^{-4}	0.0625	2^6	64
2^{-3}	0.125	2^7	128
2^{-2}	0.25	2^8	256
2^{-1}	0.5	2^9	512
2^0	1	2^{10}	1 024
2^1	2	2^{11}	2 048
2^2	4	2^{12}	4 096
2^3	8	2^{13}	8 192
2^4	16	2^{14}	16 384
2^5	32	2^{15}	32 768

计算机采用二进制的主要原因有以下几点：

(1) 容易实现 二进制在技术上最容易实现，可以用任何一个具有两种不同稳定状态的物理器件来表示两个数码：“0”或“1”。

(2) 运算简单 二进制运算规则简单，操作实现简便。一般来说， r 进制数的运算共有 $r \times (r+1)/2$ 个和与积的运算公式，对十进制而言，就有 $10 \times (10+1)/2=55$ 个和与积的运算公式；而对二进制，只有 $2 \times (2+1)/2=3$ 个和与积的运算公式，算术运算规则特别简单。

(3) 工作可靠 由于采用两种稳定状态来表示数码，使数据的存储、传送和处理都变得可靠得多。

(4) 逻辑判断方便 采用二进制后，能够方便地使用逻辑代数这一有力工具来分析、综合与设计计算机的逻辑线路。

但是，用二进制来表示一个数所使用的位数要比用十进制数表示时所使用的位数长得多，书写起来不方便，也不好记。为此，在计算机中通常还使用八进制和十六进制。

二、八进制与十六进制表示

八进制数的基数为 8，逢八进一，每个数位可取 8 个不同的数码（0~7）中的任意一个，任何一个八进制数都可表示为：

$$(N)_8 = \sum_{i=n}^{-m} K_i \times 8^i$$

因为 $r = 8 = 2^3$ ，所以只要把二进制中的 3 位数码编为一组就是一位八进制数码，两者之间的转换是很方便的。

十六进制数的基数为 16，逢十六进一，每个数位可取 16 个不同的数码和符号（0, 1, ..., 9, A, ..., F）中的任意一个，其中 A~F 表示十进制数值 10~15。任何一个十六进制数可表示为：

$$(N)_{16} = \sum_{i=n}^{-m} K_i \times 16^i$$

因为 $r = 16 = 2^4$ ，故 4 位二进制数码与一位十六进制数码相对应，同样，这两种进位计数制之间的转换也是很容易的。

既然有不同的进位计数制，那么在给出一个数的同时就必须指明它是哪个数制的数，例如：(1010)₂、(1010)₈、(1010)₁₀、(1010)₁₆ 所代表的数值就不同，如果不用下标加以标注，就会出现误会。除了用下标来表示不同的数制以外，还可以用后缀字母来表示不同的数制。后缀 B 表示这个数是二进制数（Binary）；后缀 Q 表示这个数是八进制数（Octal），本来八进制数的英文单词的第一个字母应当是 O，因为字符 O 与数字 0 太容易混淆，所以常使用字符 Q 作为八进制数的后缀；后缀 H 表示这个数是十六进制数（Hexadecimal）；而后缀 D 表示这个数是十进制数（Decimal）。十进制数在书写时后缀 D 可以省略，其他任意数制在书写时后缀不可省略。例如：有四个数为 375D、101B、76Q、A17H，从后缀字母就可以知道它们分别是十进制数、二进制数、八进制数和十六进制数。

由二进制数转换成八进制或十六进制数是很容易的，但需要注意的是，对于一个二进制混合数，在转换时应以小数点为界。其整数部分，从小数点开始往左数，将一串二进制数分为三位一组或四位一组，在数的最左边可根据需要随意加“0”；对于小数部分，从小数点开始往右数，也将一串二进制数分为三位一组或四位一组，在数的最右边也可根据需要随意加“0”。最终使总的位数成为三或四的倍数，然后分别用相应的八进制或十六进制数取代之。例如：

$$\begin{aligned} 11011.1010011011B &= \underline{011} \underline{011} . \underline{101} \underline{001} \underline{101} \underline{100}B \\ &= 33.5154Q \\ &= \underline{0001} \underline{1011} . \underline{1010} \underline{0110} \underline{1100}B \\ &= 1B.A6CH \end{aligned}$$

八进制和十六进制数之间的转换，可以通过二进制作作为中间桥梁。例如：

$$\begin{aligned} 3.145Q &= \underline{011} \underline{001} \underline{100} \underline{101}B \\ &= \underline{0011} \underline{0011} \underline{0010} \underline{1000}B \\ &= 3.328H \end{aligned}$$

大多数计算机都是用八进制或十六进制来描述计算机中的指令和数据的，例如：PDP-11 系列机就使用八进制，而大部分微型计算机，如 Intel 80X86、MC68000 等都使用十六进制。

表 2-2 列出了几种常用进位计数制对照表。

表 2-2 几种常用进位计数制对照表

十进制	二进制	八进制	十六进制
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

2.1.3 各种数制间的相互转换

前面已经提到二进制同八进制、十六进制之间的转换，这里着重讨论的是任意进制数与十进制数之间的转换。

一、任意进制数转换为十进制数

一个任意进制的数转换成十进制数，只要把各位数码与它们的权相乘，再把乘积相加，就得到了一个十进制数，这种方法称为按权展开相加法。例如：

$$100011.1011\text{B} = 1 \times 2^5 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-3} + 1 \times 2^{-4} = 35.6875\text{D}$$

$$37.2\text{Q} = 3 \times 8^1 + 7 \times 8^0 + 2 \times 8^{-1} = 31.25\text{D}$$

$$4\text{B}6.\text{CH} = 4 \times 16^2 + 14 \times 16^1 + 6 \times 16^0 + 12 \times 16^{-1} = 1254.75\text{D}$$

二、十进制数转换为任意进制数

一个十进制数转换成任意进制数，常采用基数乘法。这种转换方法对十进制数的整数部分和小数部分将分别进行处理，对于整数部分用除基取余法；对于小数部分用乘基取整法，最后把它们拼接起来。

1. 十进制整数转换为任意进制整数（除基取余法）

十进制整数转换为任意进制整数的规则是：除（以）基取余；先余为低，后余为高。

对于十进制整数转换为二进制整数来说，就是用 2 连续去除十进制整数，直至商等于 0 为止；然后逆序地排列每次的余数（先取得的余数为低位），便得到与该十进制数相对应的二进制数各位的数值。例如：

2	327	… 余数 1	(低位)
2	163	… 余数 1	
2	81	… 余数 1	
2	40	… 余数 0	
2	20	… 余数 0	
2	10	… 余数 0	
2	5	… 余数 1	
2	2	… 余数 0	
2	1	… 余数 1	
	0		↑

所以 $327D=101000111B$

用与此类似的方法也可以完成十进制整数 八进制整数，十进制整数 十六进制整数的转换，不同的是此时应当用 8 或 16 连续地去除十进制整数。例如：

8	653	… 余数 5	(低位)
8	81	… 余数 1	
8	10	… 余数 2	
8	1	… 余数 1	
	0		↑

所以 $653D=1215Q$

2. 十进制小数转换为任意进制小数（乘基取整法）

十进制小数转换为任意进制小数的规则是：乘（以）基取整，先整为高，后整为低。

对于十进制小数转换为二进制小数来说，就是连续用 2 去乘十进制小数，直至乘积的小数部分等于 0 或者达到精度要求为止；然后顺序地排列每次乘积的整数部分（先取得的整数为高位），便得到与该十进制数相对应的二进制数各位的数值。例如：

	0.6875		
×	2		
	1.3750	… 整数 1	(高位)
	0.3750		
×	2		
	0.7500	… 整数 0	
×	2		
	1.5000	… 整数 1	
	0.5000		
×	2		
	1.0000	… 整数 1	(低位)

所以 $0.6875D=0.1011B$

用与此类似的方法也可以完成十进制小数 八进制小数，十进制小数 十六进制小数的转换，不同的是此时应当用 8 或 16 连续地去乘十进制小数。例如：