全国计算机软件专业技术资格水平考试

电视函授教学丛书

高级程序员级考试辅导
计算机硬件系统

唐　毅　主编
叶祥生　主审

机械电子工业部计算机技术培训中心

全国计算机软件专业技术资格水平考试

电视函授教学丛书

高级程序员级考试辅导
计算机硬件系统

唐　毅　　主编
叶祥生　　主审

机械电子工业部计算机技术培训中心

内 容 简 介

本书是全国计算机软件专业技术资格考试电视函授教学丛书之一，专门讲解高级程序员级硬件知识。内容包括：计算机系统的基本组成、计算机的控制方式、存储体系、输入输出系统、数据通信与计算机网络、计算机系统可靠性技术与性能评价及计算机系统结构的发展等。每章末附有试题及例题选解。本书可作为高级程序员任职资格考试培训班的教材或考生的自学辅导用书，还可供大专院校有关专业师生和在职软件人员学习参考。

出 版 说 明

由国家人事部、国家科委、机电部、国务院电子信息系统推广应用办公室等组织的“全国计算机软件专业技术资格考试”从一九九〇年开始在全国统一举行。这是深化职称改革的一项重要措施，它将大大促进计算机应用人才的培养，促进人才和软件的国际交流，促进我国软件事业的发展，推动电子信息技术的广泛应用。

为配合这一考试，帮助广大软件技术人员做好应试准备和提高实际应用水平，经国家教委和机电部批准，在国务院电子信息办的领导和支持下，在中国计算机应用软件人员考试中心的指导和支持下，机电部计算机技术培训中心组织了北京、上海、南京、杭州和合肥等地区二十多个高等院校、研究所、企事业单位的三十多位专家教授，组成了编审委员会，按照考试大纲，在分析了历届考试试题的基础上，编写了这套全国电视函授教学丛书。这套丛书将于一九九一年初先出版以下六本：

程序员级考试辅导——硬软件基础知识

程序员级试题分析与解答（’87—’90）

高级程序员级考试辅导——计算机硬件系统

高级程序员级考试辅导——软件知识

高级程序员级考试辅导——软件设计方法和工具

高级程序员级试题分析与解答（’87—’90）

为了在中国教育电视台上通过卫星向全国播放辅导课程，机电部计算机技术培训中心组织了京沪地区十七名专家教授，由培训中心电教室拍摄了与上述教材配套的教学录像带（约120学时）。我们期望这套教材和配套录像带能对考生在复习应考时有所指导和帮助，也能受到广大在职计算机软件人员和大专院校师生的欢迎。

机电部计算机技术培训中心是部属的教育事业单位。该中心依托的中国计算机软件与技术服务总公司在全国组织的培训网，在三十多个省市建有分中心。该培训网的年培训量占全国同类培训量的一半以上，是我国唯一的全国性跨行业的计算机技术培训实体单位。我们衷心感谢中央和各地电子厅局、电振办、教委和仪表厅局的支持，衷心感谢上海市计算机应用软件培训中心以及各有关高等院校、研究所、企事业单位的专家教授们在编写教材、拍摄录像带和开展全国电视函授工作中给予的支持和帮助。

为慎重起见，本套教材将先作内部发行，待以后广泛征求意见并作必要的修改补充后再正式发行。欢迎各界同行提出宝贵意见，来函地址：北京学院南路55号电脑大厦机电部计算机技术培训中心，邮政编码：100081。

机械电子工业部计算机技术培训中心

一九九一年春

编 审 委 员 会

前　言

本书是全国计算机软件专业技术资格考试电视函授教学丛书之一，专门介绍高级程序员级硬件知识。主要内容包括：计算机系统的基本组成、计算机的控制方式、存储体系、输入输出系统、数据通信与计算机网络、计算机系统可靠性技术与性能评价、计算机系统结构的发展等。每章后面都配有前几届考试试题及解答，还增加了一些与试题风格相同的例题，并给出了解答。这些题解将帮助读者检查和巩固正文中学到的知识。

本书由机电部计算机技术培训中心组织，由上海科技大学唐毅副教授主编，上海第二工业大学叶祥生副教授主审。本书第一、三、七章由唐毅副教授编写，第二、四、五、六章由周颖讲师编写。

本书是根据高级程序员级考试大纲，同时参考国内外的有关资料，并在分析历届试题的基础上编写的。在编写过程中上海科技大学计算机科学系的忻琳萍、王文利、李淞江、许静鸿、水群强、周明海等同志协助做了大量工作。机电部计算机技术培训中心主任邵祖英和办公室主任邓小敏、北京软件考试实施办负责人沈林兴等同志也对本书作了审校。参加本书编校工作的还有李大方、高伟红、吕跃军等同志，在此一并表示感谢。

由于编写时间仓促，错误之处请读者与有关专家指正。

编者

1991年1月

目　录

第一章　计算机的组成

计算机系统是由硬件和软件组成的，硬件是计算机中的实际装置。计算机系统的硬件一般由中央处理机、存储器、输入/输出设备等组成。它以机器语言（即指令系统）提供给软件人员使用。本章主要介绍《考试大纲》中要求高级程序员必须掌握的硬件知识中计算机组成的基础知识。

考虑到机内代码及运算在《程序员级考试辅导——硬软件基础知识》一书中已有详细的介绍，所以，在此不再述及。

本章主要介绍了指令系统、主要部件的功能及其相互关系、控制器的实现原理、中断系统、总线等内容。

1.1　指　令　系　统

指令系统是计算机所有指令的集合。它是提供软件设计人员编制程序的基本依据。指令系统反映了计算机的基本功能及其功能的强弱。程序员用各种语言编制的程序，一般都要翻译到以指令形式表示的机器语言后才能运行。

指令系统是以程序设计者角度看的机器的主要属性，是软硬件的主要交界面。

早期计算机的指令系统是比较简单的，比如对数据的运算只能支持定点数据表示和逻辑数据表示，只有定点运算指令和逻辑运算指令。对于复杂的运算（如浮点运算）只能采用子程序的方法来实现。随着硬件价格的下降，以及实际应用的需要，指令系统逐渐由简单向复杂发展。如增加了浮点运算类指令，并增设专用硬件来实现浮点运算。指令系统的逐步扩充，指令的功能也不断得到增强。指令系统改进的途径是围绕着缩小与高级语言的语义差异，以及有利于操作系统的优化而进行的。

1.1.1　指令格式

指令由操作码、地址码两部分组成。地址码可以有零地址、一地址、二地址、三地址等几种形式。随着指令类型的不同，对地址码的长度要求变化很大，例如：操作数在寄存器内比在存储器内其地址码的长度短得多。为了缩小程序代码所占的存储容量，在计算机中，各类指令的长度可以不一致。从压缩代码的观点出发，我们希望常用指令的操作码短些，以达到最后使程序的长度也短些的目的。如某计算机模型有7条指令（$I_1 \sim I_7$），它们在程序中出现的概率用P_i表示，则可考虑图1—1所示的方案，这就是扩展操作码。使用频率高的指令的操作码为2倍，低的用四位。这不是压缩到最小代码的方案，因为在计算机中的操作码还是希望有一定的规整性，否则会引起硬件实现的复杂性。

指令	概率Pi	操作码	操作码长度
I_1	45%	00	2位
I_2	28%	01	2位
I_3	17%	10	2位
I_4	5%	1100	4位
I_5	3%	1101	4位
I_6	1%	1110	4位
I_7	1%	1111	4位

图 1—1　指令出现的概率与操作码长度的选择

另外在计算机内存放的指令的长度，应是字节的整数倍。所以操作码与地址码两部分长度之和应是字节的整数倍。在考虑操作码长度时，也应考虑地址码的要求。

图1—2～图1—4是指令在存储器中存放的例子，假设存储器字宽为32位。

图1—2为变长度指令的存放。在一个存储单元中可存放1到4条指令，由于要求指令字边界对齐，因而在存储器中可能有无用的空白。

图1—3为定长度指令（定长度操作码）的存放。由于操作码长度固定，而不同指令对地址码要求不同，甚至有的指令不需要地址码，因而也会出现空白。

图1—4是定长度指令（变长度操作码）的存放。操作码长度主要由地址码决定，这时可能有较多的，没有用到的操作码存在。

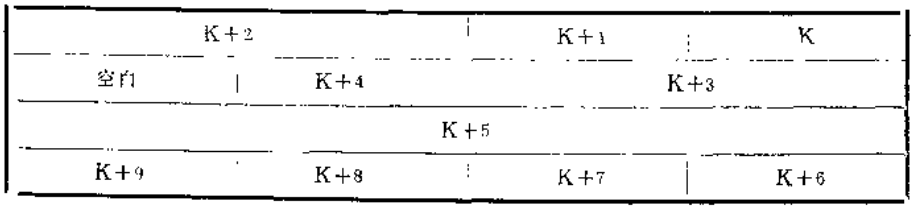

图 1—2 变长度指令的存放（K，K+1，……都是指令）

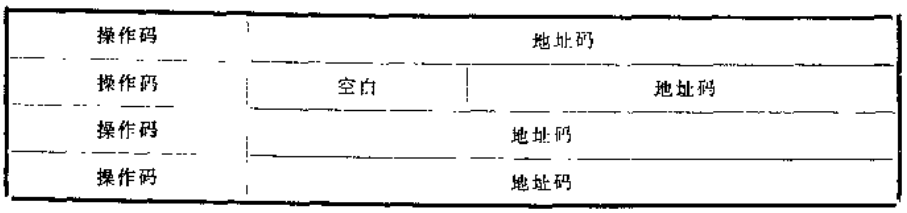

图 1—3 定长度指令（定长度操作码）的存放

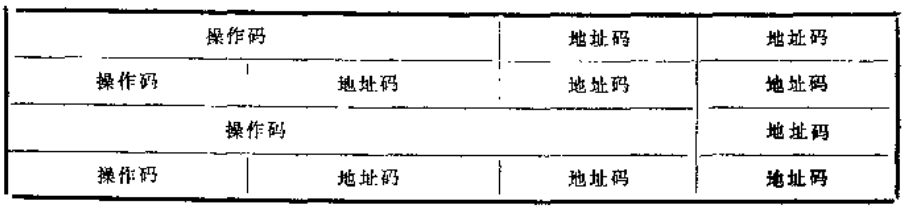

图 1—4 定长度指令（变长度操作码）的存放

总之，指令格式的变化是很大的，其方案也很多，而且总是有冗余情况存在。

1.1.2 数据表示

数据表示是指能由硬件直接辨认的数据类型。如定点数，浮点数等。运算类指令以及运算器的基本结构主要是按机器有什么样的数据表示来确定的，也就是说机器的运算类指令以及运算部件就是围绕数据表示设置的。对一种数据表示，可设置专门的指令来处理，如浮点运算指令就专门处理浮点数的计算。也可以仅设置最简单的算术逻辑运算指令，通过程序的执行来对其它数据表示进行处理，但是这种方法将使速度下降很多。为提高速度，在巨型机中设置了能直接对矩阵，向量数据（数组）进行运算的指令及硬件。这就大大提高了对向量，数组的处理速度。

目前计算机所用数据字长一般为32位，即使是微型机的字长也从8位，16位发展到32位。在存储器中的地址，一般按字节编址。但近来出现的一些机器，随着数据表示的发展，出现了按位编址方式以及相应的寻址硬件。所以计算机的指令系统可支持对字节、半字、字、双

字的运算，需要的话，还应提供位操作运算。为使于硬件实现，一般要求数据对准边界，如图1—5所示。

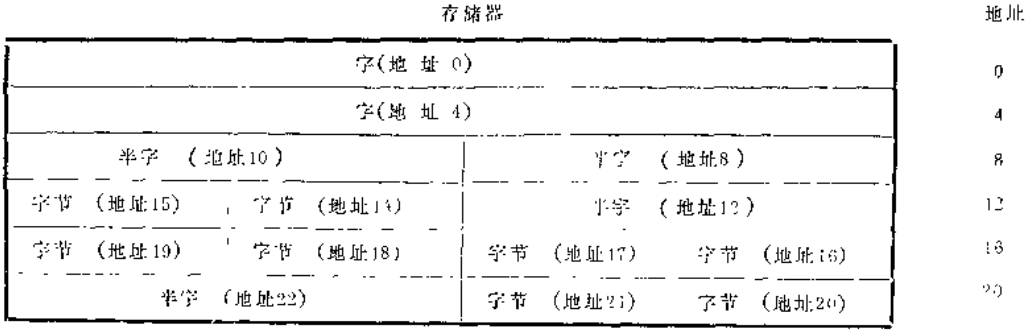

图 1—5 存储器中数据的存放

1.1.3 寻址方式

寻址方式要解决两个问题：1）指令如何访问到所需要的操作数，即如何确定操作数所在地址；2）确定转移指令的下一条要执行的指令地址。

每条指令的寻址方式由指令本身确定，或者按某些预先约定的规则进行。可以有按地址访问，按堆栈访问和按内容访问等方式，另外还可以在指令中直接表示出数据，称为立即数。与指令系统一样，各种计算机的寻址方式也是各不相同的。

1.按堆栈访问：堆栈是在存储器中的一个先进后出存储区，它的用途一方面通过专门的堆栈指令，对存在栈顶的二个数据进行运算，这便于以逆波兰表达式作为编译时的中间语言；另外，按堆栈访问也用于程序嵌套、递归时保留现场数据以及程序返回地址。在堆栈结构中，有一堆栈指针指出栈顶地址。当往堆栈内送一数时称为压入（PUSH），从堆栈内取出一数时称为弹出（POP）。当进行压入，弹出操作时，栈顶将变化，堆栈指针也跟着变化，始终指向栈顶的地址。

2.按内容访问：按内容访问适用于查询。按内容访问的相联存储器在“存储体系”中加以讨论。

3.按地址访问：按地址访问是指令系统中用得较多的一类寻址方式。对一般指令来讲就是由机器指令指明（或经计算后得到）操作数所在地址，对转移类指令来讲，就是下一条即将执行的指令地址。该寻址方式中，地址形成的方式，即编址方式在基址编址、变址编址、直接编址、间接编址、相对编址、寄存器编址等。在介绍这些编址方式之前，先了解逻辑地址，物理地址的概念。

逻辑地址：逻辑地址指的是程序员编程时所指定的程序（指令）地址和数据地址；

物理地址：物理地址指的是机器实际执行时的程序（指令）地址和数据地址。

在早期，程序员编制程序时用的程序地址和数据地址就是实际地址，因此当时并不区分物理地址和逻辑地址。后来随着多道程序以及汇编语言，高级语言的出现，由于各道程序是独立编写的，在编写程序时，程序员不可能预知他所编的这道程序会存在主存中哪个实际地址开始的空间，所以，各道程序的编写都是从零地址开始编址。而主存只有一个从零地址开始的编址空间，于是程序员编程时设定的地址（逻辑地址）与主存实际地址（物理地址）发生不符。所以，在将程序装入主存时要先进行地址的变换，即：逻辑地址空间到物理地址空间的变换，形成物理实际地址。另外，由于存储器容量的可扩充性，采用虚拟存储器方案等，决定了用户的源程序经编译后得到的是逻辑地址，然后在程序的链接时，或程序的运行

时，转换成物理地址。

以下具体介绍几种常用的编址方法。

（1）基址编址：如图1—6所示，有A、B二道程序，每道程序的逻辑地址都从零开始，然后由操作系统各自分配到a～a+l和b～b+k，在实现时可采用基址编址的方法。基址编址的原理如图1—7所示。在执行A道程序之前，先在基址寄存器中存放存储器的起始地址a，然后在执行指令时，由硬件将指令中所表示的地址（逻辑地址）和基址值a（基址寄存器的内容）相加，形成有效地址（物理地址）这就是基址编址的原理。

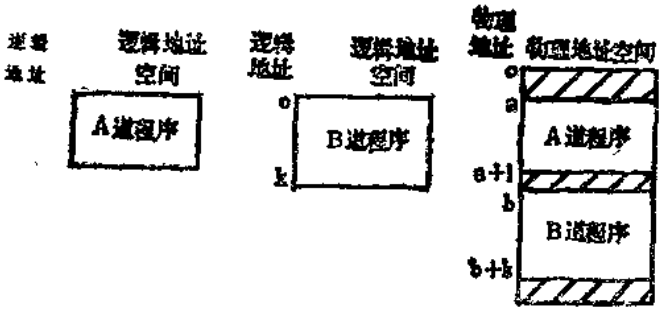

图 1—6 逻辑地址空间和物理地址空间

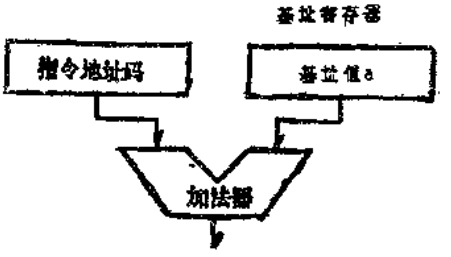

图 1—7 基址编制

另外，也可以用基址扩充寻址范围，如在16位微机中，16位地址码的寻址范围为64K字节，如将基址左移四位，与指令地址相加，即可得到20位物理地址，寻址范围可扩大到1M字节。IBM—PC就是这样处理的。

（2）变址编址：这是几乎所有计算机都采用的一种编址方式。变址操作指的是将指令中所指出的地址码与变址寄存器的内容相加，形成有效地址（图1—8）。

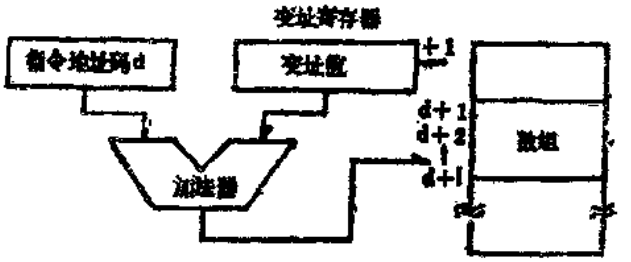

图 1—8 变址编址

图1—8表示变址操作对处理一维数组的支持，用户需编写对数组中一个元素进行运算的程序，然后改变变址寄存器的值（从1→l），对程序循环执行l次，就可以对数组中的l个元素逐个进行处理，这就是利用变址操作与循环执行程序的方法对整个数组进行运算的例子，在整个执行过程中，不改变原程序，因此对实现程序的重入性是有好处的。至于二维数组，也可用变址操作实现，但需要二个变址寄存器。

（3）直接编址、间接编址：在指令中直接指出操作数地址，称为直接编址，根据此地址即可直接取得操作数。如果按上述法取得的不是操作数，而仍是操作数地址，则称为间接编址。有的计算机还存在多级间接编址方式。

采用间接编址方式可将某存储单元作为程序的地址指针，用以指示操作数在存储器中的位置，指针值可根据程序的需要而变化。间接编址方式还可以用于提供程序转移地址。当发生中断时，执行间接编址转移指令，可直接进入相应的中断程序入口（在某一存储区，已存放中断向量表）

（4）相对编址：程序计数器PC给出当前指令的地址，在指令中给出相对于PC的偏移

值（可正，可负），二者之和形成有效地址，如有下列指令格式：

操作码	地址码 d

执行本条指令后，将转移到PC+d。相对编址方式用于转移指令。

（5）寄存器编址：为提高计算机运算速度，减少访问存储器的次数，在计算机内设置多个通用寄存器，参加运算的操作数就存放在这些寄存器内，指令对这些寄存器直接寻址。

（6）立即数：在指令中地址码部分存放的是直接参与运算的操作数。

以上介绍的是一些常用的编址方式，这些编址方式可以单独使用，也可以组合使用，如在一条指令执行过程中同时实现基址编址，变址编址，则有效地址为：
基址寄存器内容十变址寄存器内容十指令地址码。

一台计算机的编址方式如何，需要查阅有关指令系统的说明书，同一种编址方式在不同的计算机中的实现还会有差别。

用户假如用高级语言编程，根本不用考虑编址方式，因为这是编译程序的事。但若用汇编语言编程，则应对它有确切的了解，才能编出正确而有效的程序。

对虚拟存储器来讲，由于逻辑空间比实存空间大很多用上述方法求得的有效地址仍称为逻辑地址，需要通过硬件与软件的配合才能完成逻辑地址到物理地址的转换。

1.1.4 指令分类

从指令功能的角度考虑，指令系统一般应含有如下类型的指令

1.算术逻辑运算类指令：这类指令中的算术运算一般指的是对定点数进行计算，如对定点数的加、减、乘、除、算术移位等。逻辑运算指的是对字、字节进行逻辑操作处理，如逻辑与“∧”、逻辑或“∨”，逻辑移位等。

2.传送指令：指的是在寄存器、存储器之间；寄存器与输入、输出端口之间；存储器与存储器之间进行各种传送操作的指令。该类指令大部分有时也称为存数，取数指令。存数指令是把某寄存器的内容写入存储器某一单元中，取数指令是把存储器某一单元的内容读入某寄存器中。

3.程序转移指令：该类指令提供了程序流的控制机制，或者说该类指令的功能是改变程序的执行顺序。包括无条件转移指令、条件转移指令、转子程序指令、返回指令、调用（Call）指令、循环控制指令等。

4.控制指令：这类指令的功能有：允许程序去控制处理机与外部事件同步、进行存储管理、对程序状态字、或标志寄存器（或其中某几位）进行设置与修改。在多道程序和多处理机系统中，有相当一部分控制指令被称为特权指令。特权指令只能由操作系统使用。相应可由用户使用的指令称为非特权指令。在这种系统中，程序被区分成运行在操作系统环境和用户环境中两种情况，前者称管态（管理状态），后者称目态（用户状态）。在管态情况下，可运行全部指令，而在目态情况下，不允许运行特权指令，若不慎误用，处理机将自动按出错处理，目的是防止多个用户之间，或多个处理机之间的相互干扰而破坏对方（其它用户）的正常工作。

5.输入／输出指令：这类指令使主机与外围设备之间进行各种信息交换，如输入或输出数据、主机向外围设备发出各种控制命令、了解外围设备的各类工作状态。从广义的角度来看，该类指令可归入传送类指令。有些机器就是这样做的，在这些机器中，输入或输出操作

就由传送类指令来实现，而不再另设专用的输入／输出指令。在这样的实现中，应将外围设备的数据寄存器，控制寄存器，存储器统一编址。

6.浮点运算类指令：这类指令提供了对浮点数进行算术运算的功能。在科学计算中，要大量用到对实数的计算，高级语言中的实数经常是先转换成浮点数的形式而后再进行处理。若无浮点运算指令，则对浮点数的处理只能采用子程序的方法，这种方法效率很低。所以对于主要用科学计算的计算机应设有浮点运算指令，使运算器能直接对浮点数进行运算，即设置浮点运算用硬件。浮点运算的主要功能有：加、减、乘、除、比较、存取等操作，同时应能对单精度（32位）、双精度（64位）数据都进行处理。

7.十进制运算指令：该类指令使运算器直接对二—十进制数进行运算。因为在人机交互作用时，输入输出的数据都是以十进制形式表示的。在不具有十进制运算指令时，输入的十进制数须转换成二进制形式，然后进行运算，计算结果也是转换成十进制后输出。所以在输入输出数据频繁的计算机系统中设置十进制运算指令能提高数据处理的速度。

8.字符串处理指令：这类指令包括字符串传送，字符串比较，字符串转换等指令。其中“字符串传送”指的是数据块从主存储器的一个区域移动到另一个区域；“字符串比较”可以是一个字符串与另一字符串进行比较，也可以是从某一字符串中寻找某一指定的符号（字符或字符串）；“字符串转换”指的是从一种数据表达形式转换成另一种形式。

1.1.5 指令系统的兼容性

各计算机公司设计的计算机，指令的数量以及各条指令的功能是各异的，即使是一些常用的基本指令，如算逻运算指令、转移指令等也是各不相同的。因此尽管各种型号的计算机的高级语言基本相同，但将高级语言程序编译成机器语言后，其差别是很大的。将用机器语言表示的程序移植到其它机器上去几乎是不可能的。从计算机的发展过程已经看到：由于构成计算机的基本硬件发展迅速，计算机的更新换代是很快的，这就存在软件如何跟上的问题。一台计算机推出交付使用时，仅有少量系统软件（如操作系统）可提交用户，大量软件是不断充实的，尤其是应用程序，有相当一部分是用户在使用机器时不断产生的。为了缓解新机器的推出与原有应用程序的继续使用之间的矛盾，1964年在设计IBM—360计算机中所采用的系列机思想较好的解决了这一问题。从此以后，各个计算机公司生产的同一系列的计算机尽管其硬件实现方法可以不同，但指令、系统数据格式、I／O系统等（即所谓系统结构）保持相同，因而软件完全兼容（在此基础上产生了兼容机）。当研制该系列计算机的新型号或高档产品时，尽管指令系统可以有较大的扩充，但仍保留原来的全部指令、保持软件向上兼容的特点，即低档机或旧机型上的软件不加修改即可在新机器上运行，以保护用户在软件上的投资。保证软件兼容还可以通过模拟、仿真技术来实现。系列机方式所能实现的软件兼容，局限于具有相同系统结构的各机器之间，而模拟、仿真技术可以实现在具有不同的系统结构的机器之间软件兼容。有关模拟、仿真技术的详细内容请参看第六章。

1.2 计算机的基本组成

目前绝大多数计算机的结构仍然没有摆脱冯·诺伊曼型的范畴。冯·诺伊曼型计算机结构指的是冯·诺伊曼（Von Neumann）等人于1946年提出来的结构，它由运算器，控制

器，存储器和输入/输出设备组成。本节将介绍冯·诺伊曼型计算机的基本组成，并说明计算机各组成部件功能及其相互关系。

1.2.1 计算机的主要部件

按照冯·诺伊曼等人的设想：为了让计算机能按照人们的意图进行运算，就必须事先将计算方法和解题步骤翻译成机器能够理解的语言，即二进制代码形式的机器语言。人们使用机器语言来编制解题步骤，这就是编制程序的过程。接着，把编好的程序连同原始数据，通过输入设备存入计算机的存储器中。然后，启动计算机，计算机便在程序控制下，按人的意图自动地进行操作，直到全部计算完毕后，通过输出设备送出结果，计算机的基本组成如图1—9所示，

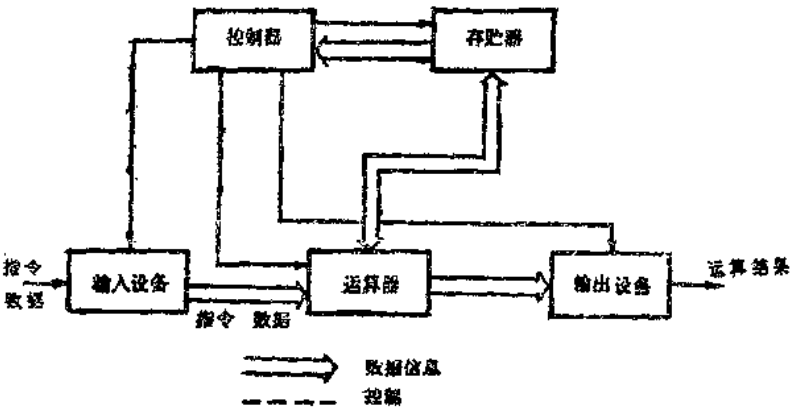

图 1—9 计算机的基本组成

其中因为运算器与控制器是计算机的重要核心部件，一般统称为中央处理器CPU（Central Processing Unit）；又把存储器与CPU统称为主机。

1.2.2 主要部件的功能及其相互关系

1.运算器：运算器是对数据进行运算的部件，它能快速地进行加、减、乘、除等算术运算及基本的逻辑运算。

在运算过程中，运算器不断地得到由存储器提供的数据，并能将求得的结果（包括中间结果）送回存储器暂时保存起来（少量可暂时存在运算器内部）。它的整个运算过程是在控制器统一指挥下，按程序中编排的先后次序有规律地进行的。

2.存储器：存储器的主要功能是保存大量信息。在使用时，根据实际的需要可以把原来记录的内容抹去而重新记录新内容，或者把原记录的内容取出但不破坏原有的记录。在计算机的存储器内保存的信息主要有数据及指令。

在解题之初，程序和原始数据从计算机外经输入设备送入存储器保存起来；在计算过程中，存储器一方面不断向运算器提供运算所需的数据，另一方面还能保存从运算器送出的结果。此外，存储器中保存着的计算程序，是用来决定计算机的具体工作过程的：计算机从存储器不断地取出指令送往控制器，然后控制器分析和解释指令的含意，并据此向运算器或其它部件发出相应的命令，指挥、控制各部件执行指令规定的操作。

3.控制器：控制器的主要作用是使整个计算机能自动地执行程序。它从存储器顺序地取出指令，并向各部件发出相应的命令，使它们一步步地执行程序所规定的任务。因此，控制器

是统一指挥和控制计算机各部件的中央机构。它一方面向各个部件发出执行任务的命令，另一方面又接受“执行部件”向控制器发回的有关任务执行情况的反馈信息，如运算器向控制器“报告”计算结果的大小是否超出预定界限等。这种由运算器、存储器及输入／输出部件发回控制器的“反馈信息”，将对控制器下一步工作产生重要影响。

4.输入/输出部件：输入/输出部件是计算机和外界进行联系的桥梁和通道，其主要功能是实现人—机对话、数据的输入与输出以及各种形式的数据变换等。输入输出部件的逻辑框图如图1—10所示。

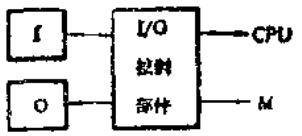

图 1—10　输入—输出部件的逻辑框图

输入设备I是把程序与数据转换为计算机能识别和处理的数据形式的设备。输入的程序和数据存入计算机存储器中。计算机的输入设备种类很多，如卡片输入机，电传打字机、软磁盘等等，它们的工作特点大同小异，一般都是将人工编制的程序与原始数据在某种媒介物上，以某种二进制编码的数字形式表现出来。载有信息的媒介物通过相应的输入设备，将信息转换为电信号形式为计算机接收，并存入存储器。

输出设备O是将计算机中二进制信息转换为用户所需要的数据格式的设备。它将计算机中的信息（计算结果等）以十进制或字符、图形、表格等形式显示或打印出来，也可以记录在磁盘上。输出设备种类很多，如打印机，电传打字机、绘图仪、磁盘、磁带或显示器等。它们的工作特点与输入设备正好相反，是将计算机中的二进制信息转换为相应的控制电信号，以十进制等人们熟悉的形式记录在媒介物上。许多设备既可以作为输入设备，又可以作为输出设备。

I／O控制部件是计算机专门用来管理输入输出设备工作的控制部件。大多数I／O设备都有机电元件，工作速度要比计算机慢几个数量级。如果由中央处理直接控制I／O设备的工作，计算机的工作速度会大大下降。为了改变这种不合理现象，使用专门的I／O控制部件，由它管理多种I／O设备工作以减轻中央处理机的负担，实现CPU与I／O设备并行工作，大大提高机器的工作速度和效率。当前，多数通用计算机都有I／O控制部件，一般称为通道式I/O处理器，使其功能进一步扩大便形成了I／O处理机。

1.2.3　计算机的工作过程

计算机各个部件能协调地工作都是在控制器的控制下完成。计算机的工作过程可以归结如下：

1.控制器控制将数据、程序从输入设备经过运算器输入到存储器（以运算器为中心的计算机）或直接输入到存储器（以存储器为中心的计算机）；

2.控制器控制从存储器取出指令送入控制器；

3.控制器分析指令，控制运算器、存储器执行指令操作；

4.运算结果由控制器控制送存储器或经输出设备输出；

5.返回到第2步，直至程序结束。

1.3　运　算　器

运算器是对各种信息进行加工处理的部件。它完成算术运算和逻辑运算。通常由寄存