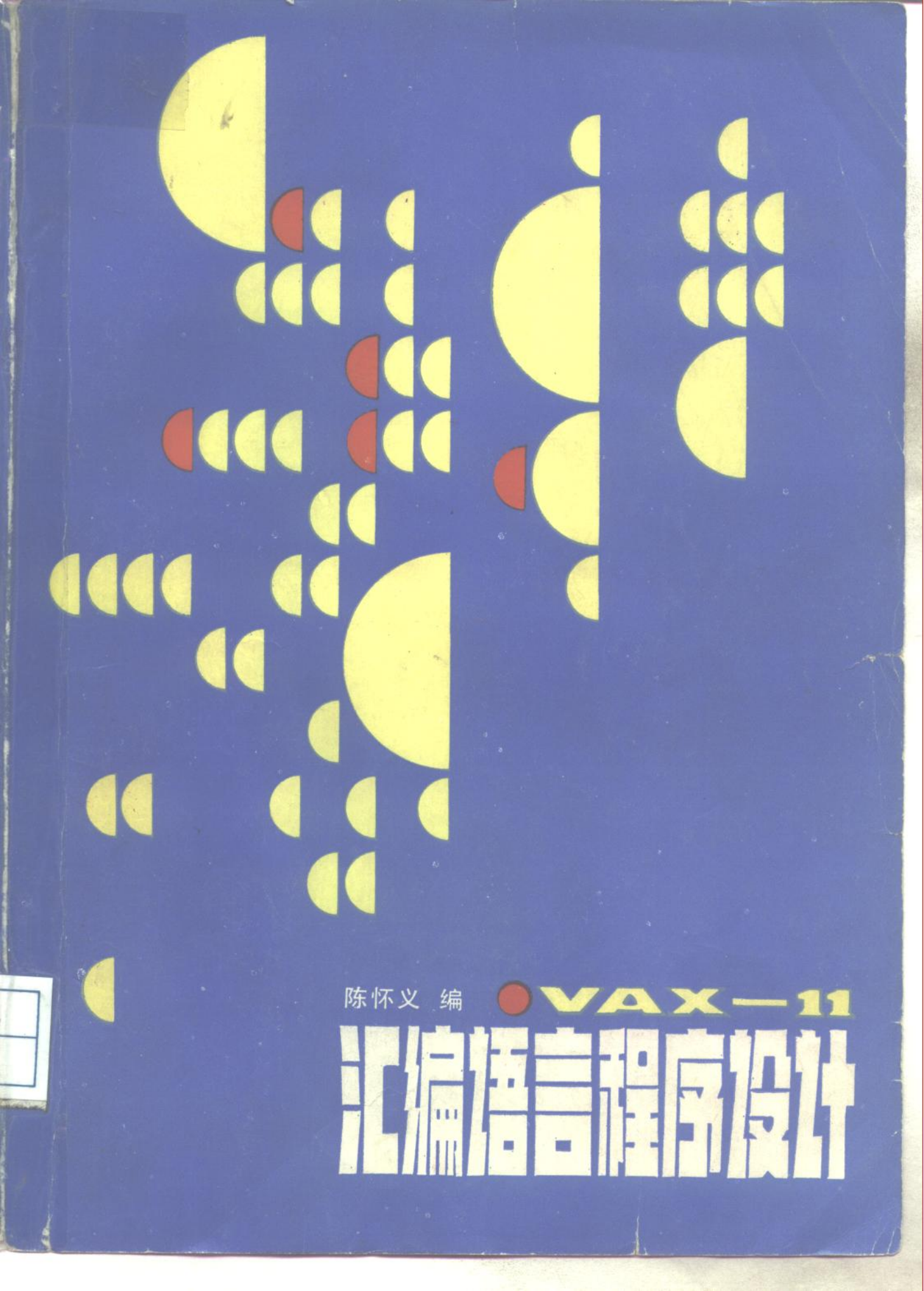


陈怀义 编

VAX—11

汇编语言程序设计

陈怀义 编

IBM-PC/XT/AT/X—11 汇编语言程序设计

国防科技大学出版社

内 容 简 介

本书以VAX-11为硬件背景，介绍计算机基本原理、数的计算机表示方法和指令寻址方式；由浅入深地阐述了汇编程序设计的基本概念、基本方法和技巧；最后还介绍了程序调试的意义、原则、方法及其软件工具；是一部颇具特色的汇编语言程序设计课教材，亦是一本汇编级程序员不可多得的参考书。

VAX-11 汇编语言程序设计

陈怀义 编

责任编辑 钟 平

*

国防科技大学出版社出版发行

（湖南长沙市 邮政编码：410073）

湖南省新华书店经销

国防科技大学印刷厂印装

*

开本：787×1092 1/16 印张：$27^{8}/_{16}$ 字数：630千

1990年2月第1版第1次印刷 印数：3000册

ISBN 7-81024-092-7

TP·19 定价：5.40元

前　言

汇编语言程序设计是计算机专业低年级的基础课程。该课程的主要目的是学习程序设计的基本原理和基本方法，用汇编语言作为工具，来进行程序设计技术和技巧的基本训练。

本书是作为汇编语言程序设计课程的教材而编写的，它的读者是程序设计的初学者，或者是已学习一点高级语言，希望进一步加深自己对程序设计的认识和提高自己程序设计能力的人。本书从描述计算机所执行的基本动作开始，由易到难，由简到繁，逐步介绍程序元素、程序结构以及程序的建造方法，一步一步引导读者自己编写出具有一定水平的汇编语言程序来。本书自始至终强调程序质量的观点，并用专门的章节来讨论程序质量的概念和介绍提高程序质量的方法。作为程序设计过程中一个必经的重要环节，本书还介绍了程序调试的意义和方法。书中所列举的大量程序实例偏重于非数值处理，以便为后续课程的学习打下一些有益的基础。每章之后都附有丰富的习题，以利于读者检查和巩固自己所学的知识。

汇编语言是面向机器的语言，书中的内容不可避免地要涉及到特定的计算机，特定计算机的选择一般有二种方法，一是设计一个抽象的计算机，一是选定一种正在实际运行着的具体计算机。本书以有代表性的VAX-11为硬件背景，采用VAX-11MACRO汇编语言为编程工具。这样做，虽有一定的局限性，但由于有具体的实践环境的支持，不仅可以巩固从书本所获得的知识，而且可以开拓学习的广度和深度，同时又培养了使用这种计算机的实际能力，故也有一定的优越性。书中的内容虽然是以VAX-11为背景写的，但除去细节，基本概念和基本方法仍是普遍适用的。

实践是本课程的一个重要学习环节。在条件允许的情况下，上机实习在整个学习中应占有相当的比重，并且上机实习与课堂讲授应该并行。光“纸上谈兵”，是很难真正学好程序设计的。

本书一共分为十三章。

第一章介绍有关程序设计的一般知识。

第二章介绍数制和数据的机器表示。

第三章介绍VAX-11计算机硬件的一般知识和VAX-11 MACRO汇编语言的基本内容。

第四章介绍VAX-11寻址方式。

从第五章到第八章，详细介绍了程序设计的基本概念和基本方法，并通过大量的例子来演示这些概念和方法的实际应用。

第九章介绍宏汇编语言特有的宏指令程序设计。

第十章介绍在各种不同环境中输入输出的程序设计方法。

第十一章简单介绍汇编过程以及连接和装入的有关知识。

第十二章介绍程序质量的概念和提高程序质量的方法。

第十三章介绍程序调试的意义、一般方法和一种系统调试工具。

在学习中，可根据具体情况节选书中的内容或改变先后顺序。如果已经学过计算机原理和数字逻辑，则第一、二章可以忽略。为了上机实习，第十章的部分内容可以提到第六章之前。如果时间不够，第十一、十二章也可以删去或一带而过。

本书在编写过程中得到陈火旺教授、齐治昌副教授的关心和指导，王广芳副教授、李健讲师、邹鹏讲师提供了宝贵意见和帮助，在此，仅表示衷心感谢。

限于水平，书中可能存在错误和不妥之处，敬请读者批评指正。

作　者

1988年2月

目　录

第一章　程序设计概述

对于正要进入计算机专业学习的人来说，最急于了解的是有关计算机和程序设计的概要性的知识，以便尽快缩短自己同计算机专业的距离。这一章从计算机软件的角度出发，向初学者介绍了计算机的基本知识、程序设计的概念及发展历史、运用计算机解题的基本步骤和各种程序设计语言的简要分析。

1.1　计算机的基本知识

程序设计是一种以计算机为基础的智力活动，任何程序最终都必须在某台计算机上实现运行才能产生实际的效果。因此，凡是学习程序设计的人首先都应该了解计算机，至少应该具备有关计算机的基本知识，这样才能较好地理解和掌握程序设计的概念和技术。学习汇编语言的程序设计尤其是这样，需要在寄存器一级上了解计算机的结构组成和工作原理，才能有效掌握汇编级上的编程技术。

1.1.1　计算机组成和工作原理

计算机是一种计算工具，为了完成一个计算过程，计算机应该有哪些基本的组成部分？这些基本的组成部分又是如何工作的呢？

在回答上面的问题之前，我们先来考察以算盘为工具的人工计算过程是怎样进行的。

假设要计算算术式

$$1024\div16+64\times16$$

的值。人工计算可按下述步骤进行：

1．将算术式$1024\div16+64\times16$写在纸上

2．按除法口诀，在算盘上做1024除以16的运算，得

$$1024\div16=64$$

3．将商64记在纸上

4．按乘法口诀，在算盘上做64乘以16的运算，得

$$64\times16=1024$$

5．将积1024记在纸上

6．按加法口诀，在算盘上做64加1024的运算，得

$$64+1024=1088$$

7．将和1088记在纸上，计算结束。

这样，就完成了该算术式的人工计算过程，

即 $$1024\div16+64\times16=1088$$

从上述计算过程可以看出，参加计算的有纸、笔、算盘以及使用这些工具的人（当然，参加计算的人必须懂得珠算的口诀）。计算过程的每一个步骤都是由人操纵的，离不开人的直接动作。

我们仍以上面所说的那个算术式为例，来考察以计算机为工具的计算过程是怎样进行的。

1．将解题的顺序（除→乘→加）以及参加运算的数据1024、64和16都存放在计算机的存贮设备里。

2．将数据1024和16取到计算机的运算设备中，做除法运算，得

$$1024\div16=64$$

3．将商64记入存贮设备

4．将数据64和16取到运算设备中，做乘法运算，得

$$64\times16=1024$$

5．将积1024记入存贮设备

6．将商64和积1024取到运算设备中，做加法运算，得

$$64+1024=1088$$

7．将和1088记入存贮设备，计算结束。

可以看出，以计算机为工具的计算过程与以算盘为工具的人工计算过程很相似，在这里，存贮设备相当于纸，运算设备相当于算盘。但是与人工计算过程又不同，这里没有人的干预，每一个步骤都是在计算机中自动进行的。计算机之所以能使计算过程自动进行，是因为计算机中还有一个控制设备，它懂得运算法则并能够按照存贮在存贮设备中的解题步骤指挥计算机动作，相当于人工计算过程中懂得珠算口诀的人。因此，与人工计算过程相比，以计算机为工具的计算过程可以称为自动计算过程。

计算机只不过是一个为人们服务的计算工具，它终究要同人发生关系。人们通过输入设备将需要完成的计算任务告诉计算机；通过输出设备从计算机那里得到计算的结果。因而，输入、输出设备构成了人与计算机的接口，实现人机之间的通信。

由以上叙述不难看出，计算机至少要由**存贮、运算、控制、输入、输出**这五种设备组成。但有了这五种设备，我们还只能说，计算机具备了自动计算的能力，它必须按人的意志行事，才能使这种能力得到应用和发挥。由人所决定的运算步骤必须以一种计算机能理解的形式加以描述，并存入存贮设备，计算机才能实现某个特定的计算过程。计算任务的处理对象和处理规则的描述就称为**程序**。

上面所说的计算机的五种设备就是计算机最基本的组成部件，任何类型的计算机的基本组成部分都离不开这五大部件。它们之间的联系如图1-1所示。

下面再对这五大部件作进一步的介绍。

1．**存贮器**(Memory)。存贮器是用来存贮程序和数据的重要部件。它不仅能保存大量信息，而且允许快速（从存贮器）读出信息和快速（向存贮器）写入信息。

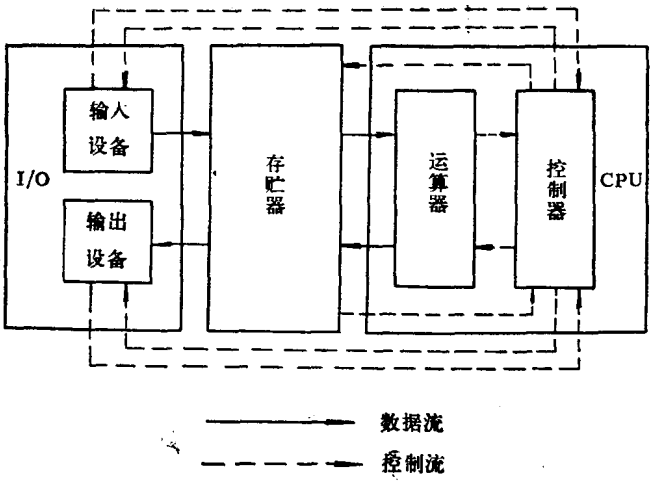


图 1-1 计算机组成

计算机的设计，要求被处理的信息在计算机内采取二进制的表示形式。不论是程序还是数据，都是以二进制形式存贮在存贮器中的。

存贮器的基本构成单元是存贮元件。一个存贮元件有二个稳定状态，可存贮一位二进制信息。如具有双稳态的半导体触发器或具有两种剩磁状态的铁淦氧磁芯等都可作为存贮元件。若干个按一定顺序排列的存贮元件组成一个**存贮单元**（Location）。如果一个存贮单元由8个存贮元件组成，则该存贮单元有8位，可以存贮一个8位的二进制信息。对于某一种型号的计算机来说，所有的存贮单元都有相同的固定位数。但对不同型号的计算机，其存贮单元的位数可以不同，有8位，16位，32位，48位，60位，64位等多种。

为了区分存贮器内不同的存贮单元，通常对存贮单元进行统一编号，这个号码称为存贮单元的**地址**（Address）。如果存贮器有n个存贮单元，其地址编号则从0到n－1．每个存贮单元都有一一对应的地址，通过这个地址可唯一地访问这个存贮单元。存贮器的组织如图1-2所示。

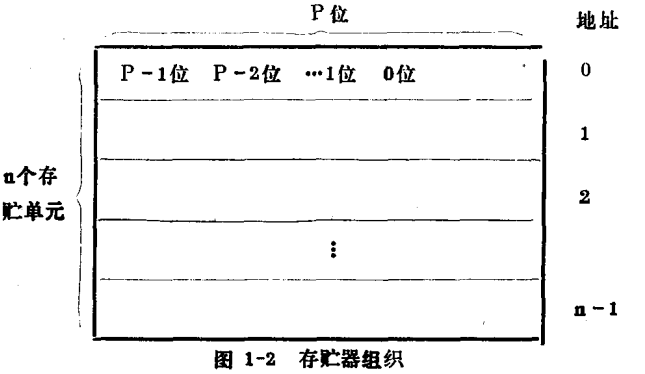


图 1-2 存贮器组织

计算机是通过地址在存贮器中存取信息的。当我们想要从某个存贮单元取出信息或者要将信息存入某存贮单元时，总是先将相应的单元地址送往存贮器，存贮器根据地址寻找出相应的位置以后，才能将所需的信息读出或写入。在通常情况下，读出时，被读存贮单元的内容不变，以便继续使用；写入时，被写存贮单元的原有内容被破坏而代之以新写入的内容。因为存贮单元是可寻址的最小存贮单位，所以一个存贮单元中所包含的信息是计算机能够存取的最小信息单位。一般称这个信息单位为**字**（Word）。存贮单元的位数称为**字长**（Word Length）。但是对于存贮单元位数较少的情况，例如8位，由

于能表示的数值范围太小，甚至满足不了一般的计算需要，所以有的计算机一次存取二个单元、四个单元或八个单元的存贮单元序列。将这个存贮单元序列中所包含的信息称为字，相应的计算机字长则为16位，32位或64位。在这种情况下，一个存贮单元所包含的信息则称为一个**字节**（Byte）。一个字由二个、四个或八个相邻的字节组成。

存贮器所具有的存贮单元总数称为存贮器的**容量**（Memory Capacity）。存贮器的容量一般为2^n个单元。当n＝10时，存贮器单元数为1024，并简记为1K，当n＝11时，存贮器容量为2048个单元，记作2K．容量的单位可以是字，也可以是字节。假设某种计算机一个字由两个字节组成，如果它的存贮器容量为64K字节，则也可以说容量为32K字。

一个存贮单元对应于一个地址，所有地址的集合称为**地址空间**（Address Space）。显然，存贮器容量的大小也就是地址空间的大小。地址空间大小的设计取决于计算机规定用于表示地址的二进制位数，一般这个位数都等于或小于字长。例如许多小型机都采用16位地址，它们的地址空间即由$0,1,\cdots,2^{16}-1$这2^{16}个地址组成。

存贮器的容量越大、读写速度越快，意味着机器的功能越强，但造价也就越高。出于成本和性能等因素的综合考虑，一般都把存贮器分为二级。一级为**内存贮器**，也称为主存贮器，由半导体、磁芯等存贮元件组成，其存取速度较快，但容量较小，直接同运算器打交道；另一级为**外存贮器**，也称辅助存贮器，如磁带、磁盘等，其存取速度较慢，但容量很大，它不直接同运算器打交道，必须将它的信息传送到主存后才能由运算器进行处理。一般所称存贮器均指主存。

存贮器的工作是通过选址系统、读写系统、存贮控制线路和缓冲寄存器来完成的，如图1-3所示。

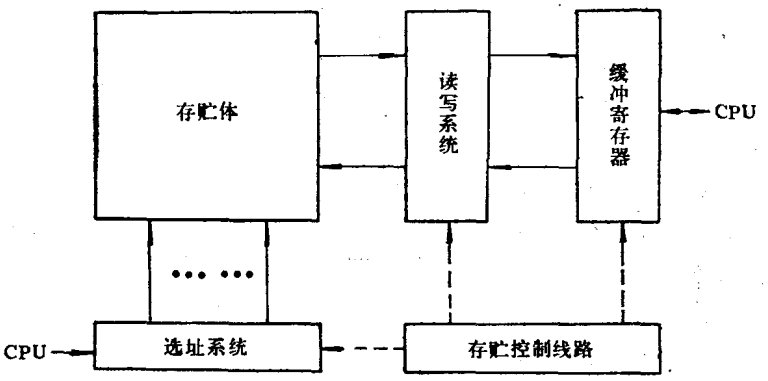


图 1-3 存贮器工作原理

选址系统包括地址寄存器和地址译码器。读写系统包括读写驱动线路和读出放大器。存贮控制线路是一个复杂的逻辑延时网络。

读出时，首先由地址寄存器获取欲读出的存贮单元的地址，选址系统选中存贮单元，在控制线路控制下，将被选单元内容读到缓冲寄存器中以备使用。写入时，在地址寄存器得到欲写存贮单元地址的同时，欲写的代码内容也送到了缓冲寄存器中。然后在控制线路的控制下，将缓冲寄存器里的内容经读写系统写入到选址系统选中的存贮单元中。

2．**运算器**（Arithmetic Unit）。运算器是计算机进行算术运算和逻辑运算的主要部件。除了算术和逻辑运算之外，运算器还能完成一些其它的动作，如数码移位，数据传送等。运算器从存贮器中取得运算数据，经过运算后所得运算结果，或者保留在运算器以备下一次运算使用，或者再送回存贮器。一般计算机可以进行几十种到几百种不同的运算，这些运算也可以称为**操作**（Operation）。运算过程是在控制器控制下自动进行的，计算机每秒可进行几十万次，几百万次甚至几亿次基本运算。

运算器主要是由寄存器和算术逻辑线路构成，如图 1-4 所示。图 1-4 中忽略了许多控制环节和其他一些细节。图中的三个寄存器是运算器的基本寄存器。

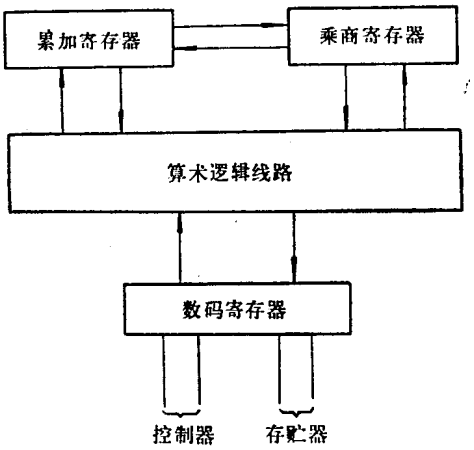


图 1-4　运算器工作原理

寄存器（Register）是用来存放数据的电子器件，一般由有一定顺序的若干位高速触发器组成。每位触发器的状态与所存数据的对应二进制位的数值一一对应。参加运算的数据和运算的结果都可保存在寄存器中。寄存器的位数根据它的用途而定，用于数据运算的寄存器的位数一般都等于或大于字长。算术逻辑线路是根据各种运算规则而设计的组合逻辑线路，它的主要任务是完成各种运算，一般没有寄存的功能。

运算器中的三个基本寄存器是累加寄存器、数码寄存器和乘商寄存器。累加寄存器既可存放一个准备参与运算的数据，又可暂时存放运算的结果。当做加法时，它先存放被加数，运算完成后又存放和，原来的被加数被和取代，就象原被加数加上了加数一样，故称为累加寄存器，简称**累加器**（Accumulator）。数码寄存器（Data Register）存放参与运算的另一个数据，如加数、被乘数等。同时它又是运算器与存贮器，运算器与控制器交换信息的通路。乘商寄存器（Multiplier-Quotient Register）用来实现乘除法运算，它存放乘数或商。

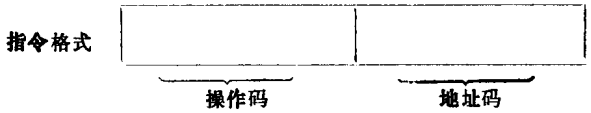
算术逻辑线路的核心部分是加法器，因为加、减、乘、除等运算都归结为加法和移位操作。

运算器进行什么运算完全是由控制器决定的，运算器与控制器是计算机的核心部件，一般合称为**中央处理机**（Central Processing Unit），简记为CPU.

3．**控制器**（Control Unit）。控制器是指挥和控制计算机工作的中心部件。它的任务是根据人们预先确定的算法步骤，控制和协调计算机各部件的自动工作。而这些算法步骤的形式描述就是程序，因此可以说控制器的工作又是由程序来支配的。

算法步骤的描述可以通过安排计算机自动操作的顺序和每步操作的内容来表示。其中每一步操作都是计算机所能进行的基本操作。要求计算机进行基本操作的命令称为**指令**（Instruction）。指令在计算机中形式化地表示为二进制位的序列。这个二进制位的

序列一般分为前后两部分，如下所示：



其中操作码指明计算机基本操作的类型，如传送、加法、乘法等；地址码指明参与操作的数据存贮在什么地方。一种计算机所有指令的集合称为**指令系统**（Instruction Set）。程序则是指令的序列。由于指令数码化，故程序可以象数据一样存贮在存贮器中。程序存贮在计算机中是计算机摆脱人工的干预，实现自动工作的最根本的条件。因为控制器控制和协调计算机各部件自动工作，正是通过从程序中逐条读取指令、分析指令和执行指令来实现的。

控制器的基本组成如图1-5所示。按功能可分为指令控制、地址形成、时序信号产生和微操作控制四个部分。指令控制部件包括程序计数器、指令寄存器和指令译码器等。地址形成部件包括地址寄存器、变址寄存器和地址计算部件。时序信号产生部件主要包括时钟脉冲源和时序信号产生器。微操作控制部件是一个复杂的逻辑网络控制线路。

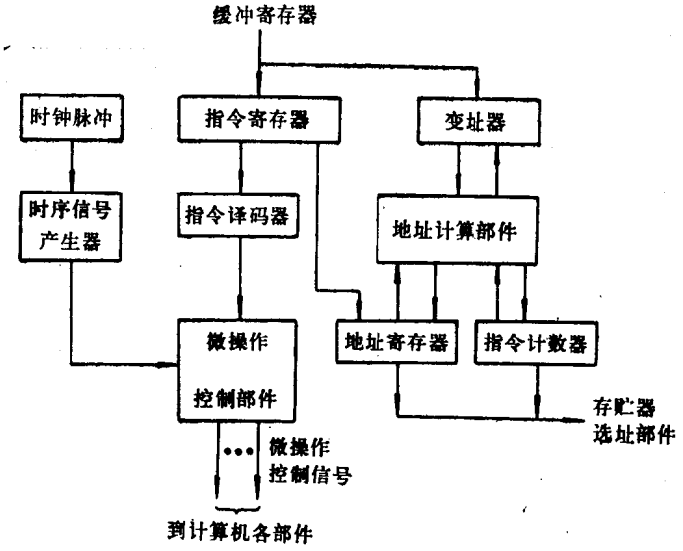


图 1-5 控制器工作原理

指令的编排顺序是由算法步骤的先后顺序决定的。程序是指令的序列。程序存入存贮器时指令依编排顺序连续存放，这样，存贮指令的单元地址顺序与指令的编排顺序就是一致的。计算机为了执行规定的算法步骤，只需按地址递增的顺序从存贮器读取指令并加以处理即可。

程序计数器（Program Counter）又称指令计数器，简记为PC，是用于管理指令的读取顺序的。PC中存放待执行指令在存贮器中的存贮地址。控制器工作时，按PC中的地址从存贮器中读取指令。具体做法是，控制器将PC的内容送到存贮器的选址部件，再向存贮器发出读命令，把所需指令通过缓冲寄存器读到控制器。当一条指令读出以后，PC中的地址就自动增加一个固定的量，以便指出紧接在后面的那条指令的地址。当前

一条被读取的指令执行完后，控制器就立即又按PC的新值从存贮器中读取下一条指令。自动修改PC内容的地址增量取决于存贮器的编址方式和指令的长短。如果一个存贮单元只存放一条指令，则地址增量为1；如果二个存贮单元存放一条指令，则地址增量为2，等等。当程序要求改变指令的执行顺序时，只需将PC的内容强行作相应的修改（由当前执行的指令操作实现），指令就可以按新的顺序执行。这种指令执行顺序的改变称为“转移”。

控制器从存贮器中读取的指令，保存在指令寄存器中，直到这条指令执行结束，被读取的下一条指令取代为止。在指令执行时，指令的操作码部分送入指令译码器。指令译码器将指令操作码转换成控制信号，控制计算机各部件进行相应的操作。指令的地址码部分则送入地址寄存器，用来形成操作数的存贮地址。

一般指令给出的地址码部分称为形式地址，真正的有效地址，要在考虑变址寄存器的值，通过地址计算部件计算以后才能得到。地址计算的规则取决于指令格式，指令格式不同，计算地址的方法也不一样。这种由形式地址变换为有效地址的过程称为**寻址**（Addressing）。

时钟脉冲源是计算机各部件协调而准确地进行操作的同步时钟标准。实际上它是一个具有一定频率的信号发生器，其工作频率称为计算机主频。计算机的速度技术指标主要取决于主频。时序信号产生器在每一条指令执行期间，按时间先后次序发出若干个节拍信号。两个时钟脉冲的时间间隔（主频周期）即一个节拍，它是执行指令时完成最基本操作所需的时间间隔。指令执行的每个基本操作与时序信号产生器发生的每个节拍信号一一对应。

微操作是指计算机各部件在一个节拍中能完成的基本操作。指令都是分解成许多微操作来执行的。微操作控制部件的功能就是根据指令译码器给出的操作控制信号和时序信号产生器给出的节拍信号，向计算机的各部件发出各种微操作命令。

4．**输入输出设备**（Input-Output Device）。输入输出设备又称为I/O设备，是计算机与外界相互联系的部件。主要用于人-机对话和信息的输入输出。

输入设备是把程序和数据转换成计算机能识别和处理的数据形式的设备。程序和数据一般都是以某种二进制编码的形式、通过输入设备存入存贮器的。

输出设备是将计算机中的二进制信息转换成用户所需数据格式的设备。计算机中的信息以十进制数字、字符、图形、表格等形式显示或打印出来，或者录入磁带、磁盘中。

纸带输入机，卡片输入机等是输入设备。打印机，绘图仪等是输出设备。而控制打字机，显示器终端，磁带机，磁盘机等既是输入设备又是输出设备。

在输入输出设备的组成中大都有机电元件，其工作速度与CPU和存贮器的工作速度极不匹配，因此往往都不由CPU直接控制，而设置一个I/O控制部件，由它来直接管理I/O设备的工作，CPU只同I/O控制部件打交道。这样，CPU与I/O设备就可并行工作，大大提高了整个计算机的工作速度和效率。这种I/O控制部件一般称为**通道**或**I/O处理机**。

1.1.2 计算机系统

如果一台计算机仅仅只有前面所说的五大部件，那么这台计算机的能力是极其有限

的，使用起来也是极不方便的。五大部件和必要的电源、机架、散热系统等组成的计算机，在每一次计算过程中，其一举一动都得由人通过指令不厌其烦地去具体安排。即使对计算机构造细节了解得十分清楚的人，使用这种计算机也会感到十分不方便。这种计算机习惯上称为裸机（Bare Machine）。裸机基本上只是由一些集成电路、电子器件等物理实体组成，我们称它们为**硬件**（Hardware）。

为了方便用户使用计算机，为了提高计算机的效率或扩展硬件功能，人们编制了一些程序永久性地存放在计算机中。如汇编程序、编译程序、操作系统、数据库管理程序以及各种各样的标准子程序和实用程序等。有了这些程序，人们就不需要再去安排计算机的动作细节，人们在使用裸机时所必须做的许多工作可由常驻计算机的这些程序代劳。这些程序不仅能根据用户要求自动管理存贮空间，调度作业（计算任务），组织数据的物理结构等，而且，由于这些程序的帮助，用户用形式化的程序设计语言编写的计算过程（与机器指令完全不同）也能在计算机上执行。因此，对计算机构造细节了解不多，甚至完全不了解的人，也能方便地使用计算机。另外，用户在使用计算机中经常遇到的一些重复的工作，例如求平方根，求三角函数的值，文件的复制，数据记录的修改等等，也只需要用户以某种简单的方式提出要求而具体工作由这些程序去完成。这些程序常驻计算机中，成了计算机的一部分，但由于它们只不过是一些二进制信息编码，因此我们称它们为**软件**（Software）。另外，当计算机用于特定环境解决某一特定问题时，为了减轻用户的工作量需要编制一套特定的程序，如线性方程组的求解程序，数值气象预报程序，运输调度程序，企业管理程序等等。在计算机的某一特定应用领域，这些程序也是计算机所不能缺少的，所以它们也被称为软件。但这种类型的软件的作用是针对某个特定应用环境的，因此称它们为**应用软件**。前面所说的软件是适用于任何环境，任何用户的，因此称它们为**系统软件**。因为软件成了计算机整体组成的一部分，所以软件又称为软设备。

由硬件和软件所组成的计算机整体，我们称为**计算机系统**（Computing System）。其组成如图 1-6 所示。任何一个有实用意义的计算机整体都应该是一个计算机系统。没有硬件固然不行，软件再好再多也发挥不了作用；没有软件也不行，纯硬件功能的计算机在它必须应付的任务面前常常是软弱无力的。只有软硬件相互配合，计算机才能有效地发挥出它的巨大能力。

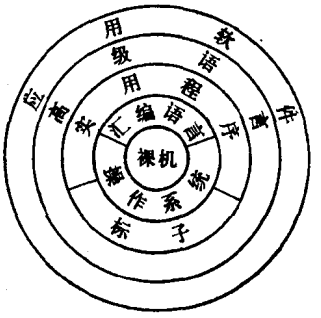


图 1-6 计算机系统组成

计算机可以理解为面向算法的机器，因为在程序的控制下它能实现某个特定的算法。一个算法可以由硬件实现，也可以由软件来实现。例如，在早期的计算机中，硬件只能实现四则运算中的加减法，乘除法则是由程序来完成的。而在后来的计算机中，乘除法都作为一条基本指令出现，完全由硬件来实现了。又如向量运算在大多数计算机中是依靠软件完成的，但在巨型机或向量处理机中则是由硬件来实现的。因此，硬件和软件在逻辑功能上是等效的。

早期的计算机，由于硬件成本高，可靠性差，为了取得较高的性能价格比，许多功

能常常用软件来实现。随着计算机的发展，硬件技术迅速发展，成本迅速下降。而相对来说，软件技术发展较慢，随着软件日益复杂和规模增大，软件的成本越来越高。所以，现在凡是能用硬件实现的功能都尽量采用硬件设计，以降低计算机的成本。某些技术已经成熟的软件（尚限于较小规模），也尽量与硬件相结合，设法用硬件设计来实现其软件功能，这就是所谓软件的固化，现在软件固化成为计算机设计的技术之一。

1.2 程序设计与计算机

从计算机诞生的那一天起，程序设计与计算机就联系在一起了。虽然程序和程序设计的概念在计算机问世之前就已存在，但计算机的发展给程序设计提供了从未有过的广阔天地，程序设计的概念和技术在计算机的发展中得到了极大的丰富和迅速的进步。为学习计算机的程序设计，了解一些程序设计的历史和程序设计在当前社会中的作用是十分有益的。

1.2.1 程序设计的概念

程序的一般概念是指计划，方案，大纲等。演出的节目单，会议的议程都可以称为程序。而这些节目单的编排，议程的制定则被称为程序设计。将程序与程序设计同计算机联系起来，就是**计算机程序**（Program）和**计算机程序设计**（Programming）了。

计算机是顺应军事和工程数值计算的迫切需要而发展起来的计算工具，但由于计算机逻辑判断功能的存在，使得它在非数值计算的数据处理领域也得到了迅速的广泛的应用。无论是数值计算或非数值计算，应用计算机来进行数据处理，其实质不过是对给定的初始数据进行某种规定的加工，以求得与给定初始数据有关的结果数据。这种规定的加工方法称为算法（Algorithm）。在这个意义上来说，计算机就是执行算法的机器，计算机程序就是用计算机所能接受的语言编写的算法，计算机程序设计就是要在计算机上实现某种算法。

严格一点来讲，一个**算法**就是一个有穷规则的集合，其中的规则规定了一个解决某一特定类型问题的运算序列。此外，一个算法还有五个重要特性：

1．有穷性。一个算法必须在执行有穷步之后结束。

2．确定性。算法的每一步骤都是确定的，待执行的动作对每一种情况都有严格确切的规定。

3．输入。算法有 0 个或多个输入，即在算法开始执行之前，对算法最初给出的量，这些输入取自确定的对象集合。

4．输出。算法有一个或多个输出，即与输入有确定关系的量。

5．能行性。算法中所有有待实现的运算必须都是十分基本的，用手工也能在有限的时间内做完。

描述算法的工具有多种，如自然语言、图、表、程序设计语言等等，用自然语言描述算法固然便当，但由于自然语言本身的二义性，使得存在对算法误解的可能性，而且，目前尚没有一种计算机能理解和执行用自然语言描述的算法。图、表虽然明显直