



中国计算机学会教育专业委员会 推荐
全国高等学校计算机教育研究会 出版
高等学校规划教材

计算机组成原理与 汇编语言程序设计

俸远祯 王正智 徐洁 俸智丹 编
朱怀芳 主审

计算机学科教学计划1993



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
URL: <http://www.phei.co.cn>

高等学校规划教材

计算机组成原理与汇编语言程序设计

俸远楨 王正智 徐 洁 俸智丹 编

朱怀芳 主审

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书系统地介绍计算机组成原理与汇编语言程序设计方法,强调整机概念与软硬结合。全书分为四篇:第一篇介绍 CPU 子系统,首先从机器指令级描述其基本组成,然后介绍计算机中数据信息的表示方法及指令系统,再通过一台模型机的设计阐明指令执行流程与微操作。第二篇介绍 8086/8088 汇编语言,讲解程序设计方法,并提供较多程序实例。第三篇讨论运算功能的实现方法,解释运算指令功能的硬件实现方法,以及通过程序扩展运算功能的方法。第四篇介绍存储器子系统与 I/O 子系统,包含主存储器设计、存储系统组织、I/O 控制方法、总线、接口、常用 I/O 设备工作原理、I/O 程序设计等内容。

本书可作为高校计算机专业教材,也可供从事计算机专业的科技人员参考。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,翻版必究。

图书在版编目(CIP)数据

计算机组成原理与汇编语言程序设计/俸远祯等编. - 北京:电子工业出版社,1999.7

高等学校规划教材

ISBN 7-5053-3863-3

I. 计… II. 俸… III. ①电子计算机-基础理论-高等学校-教材 ②汇编语言-程序设计-高等学校-教材 IV. TP301

中国版本图书馆 CIP 数据核字(1999)第 30268 号

丛 书 名:高等学校规划教材

书 名:计算机组成原理与汇编语言程序设计

编 者:俸远祯 王正智 徐 洁 俸智丹

审 校 者:朱怀芳

责任编辑:张凤鹏

特约编辑:袁 英

排版制作:电子工业出版社计算机排版室监制

印 刷 者:民族出版社印刷厂

出版发行:电子工业出版社 URL:<http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036

经 销:各地新华书店

开 本:787×1092 1/16 印张:33.5 插页:1 字数:883.3 千字

版 次:1999 年 7 月第 2 版 2001 年 7 月第 5 次印刷

书 号:ISBN 7-5053-3863-3
G·303

定 价:38.00 元

凡购买电子工业出版社的图书,如有缺页、倒页、脱页、所附磁盘或光盘有问题者,请向购买书店调换;若书店售缺,请与本社发行部联系调换。电话 68279077

出版说明

中国计算机学会教育专业委员会和全国高等学校计算机教育研究会(以下简称“两会”),为了适应培养我国 21 世纪计算机各类人材的需要,根据学科技术发展的总趋势,结合我国高等学校教育工作的现状,立足培养的学生能跟上国际计算机科学技术发展水平,于 1993 年 5 月参照 ACM 和 IEEE/CS 联合教程专题组 1990 年 12 月发表的《Computing Curricula 1991》,制定了《计算机学科教学计划 1993》,并组织编写与其配套的首批 18 种教材。现推荐给国内有关院校,作为组织教学的参考。

《计算机学科教学计划 1993》是从计算机学科的发展和社会需要出发提出的最基本的公共要求,不是针对某一具体专业(如计算机软件或计算机及应用专业),因此它适用于不同类型的学校(理科、工科及其它学科)、不同专业(计算机各专业)的本科教学。各校可以根据自己的培养目标和教学条件有选择地组织制定不同的教学计划,设置不同的课程。本教学计划的思想是将计算机学科领域的知识,分解为九个主科目(算法与数据结构、计算机体系结构、人工智能与机器人学、数据库与信息检索、人-机通信、数值与符号计算、操作系统、程序设计语言、软件方法学与工程)作为学科的公共要求;对计算机学科的教学归结为理论(数学)、抽象(实验)和设计(工程)三个过程,并强调专业教学一定要与社会需要相结合。另外,还提出了贯穿于计算机学科重复出现的十二个基本概念,在深层次上统一了计算机学科,对这些概念的理解和应用能力,是本科毕业生成为成熟的计算机学科工作者的重要标志。

为了保证这套教材的编审和出版质量,两会成立了教材编委会,制定了编写要求和编审程序。编委会对编者提出的编写大纲进行了讨论,其中一些关键性和难度较大的教材还进行了多次讨论。并且组织了部分编委对教材的质量和进度分片落实,有的教材在编审过程中召开了部分讲课教师座谈会,广泛听取意见。参加这套教材的编审者都是在该领域第一线从事教学和科研工作多年,学术水平较高,教学经验丰富,治学态度严谨的教师。这套教材的出版得到了电子工业出版社的积极支持。他们把这套教材列为出版社的重点图书出版,并制定了专门的编审出版暂行规定和出版流程,组织了专门的编辑和协调机构。

这套教材的编审出版凝聚了参加这套教材编审教师和关心这套教材的教师、参与编辑和出版工作者、以及编委会成员的汗水,他们为此作出了努力。

这套教材还得到电子工业部计算机专业教学指导委员会的支持,其中 11 本被选入 1996~2000 年全国工科电子类专业规划教材。

限于水平和经验,这套教材肯定还会有缺点和不足,希望使用教材的单位、教师和学生积极提出批评建议,共同为提高教学质量而努力。

中国计算机学会教育专业委员会
全国高等学校计算机教育研究会

教材编审委员会成员名单

主任：王义和 哈尔滨工业大学计算机系
副主任：杨文龙 北京航空航天大学计算机系（兼北京片负责人）
委员：朱家铨 东北大学计算机系（兼东北片负责人）
龚天富 电子科技大学计算机系（兼成都片负责人）
邵军力 南京通信工程学院计算机系（兼南京片负责人）
张吉锋 上海大学计算机学院（兼上海福州片负责人）
李大友 北京工业大学计算机系
袁开榜 重庆大学计算机系
王明君 电子工业出版社
朱 毅 电子工业出版社（特聘）

前 言

ACM/IEEE-CS《计算教学计划 1991》发表后,对我国计算机专业的教学体系也产生了重大的影响。以此为依据,中国计算机学会教育专业委员会和全国高等学校计算机教育研究会组织了数十位专家,经过多次讨论,制订了《计算机学科教学计划 1993》,并提出了相应的配套教材编写计划,本书就是其中的一本。此后,本书由全国工科电子类专业专业教材《计算机》教材编审委员会评选择优,作为“九五”规划教材,推荐出版。

本书将传统教学计划中的两门主干课程《计算机组成原理》与《汇编语言程序设计》,有机地组织成为一门课程,这是建立新教学体系的一种努力。其目的是围绕机器指令级(传统机器级),及与此紧密相关的微操作级和汇编语言级,更好地体现软硬结合,了解内部工作机制与编程求解问题相结合。从CPU和整机系统两个层次,从逻辑组成、内部工作机理、程序设计等三个方面,使学生建立比较清晰的整机概念。

本课程的参考教学时数约为100~120学时,全书内容可能比教学时数所允许的分量稍多一些,可供选取或让学生自学,部分实例可随技术发展而更新。由于涉及面较宽,我们将全书分为四篇,以体现模块性,具体教学顺序则可以适当改变。

本书的编写顺序体现了下述的教学思路:第一篇讲解CPU的组织及其内部工作机理,并从CPU的角度说明主存储器的基本特性,及主机与外部进行信息交换的基本方法,这就为讲解程序设计奠定比较坚实的硬件基础。第二篇在此基础上阐明程序设计方法。在第三篇讨论运算方法时,就可以既讨论硬件实现也讨论软件实现,既解释运算指令功能的硬件实现,也提供通过软件扩展运算功能的方法。第四篇介绍存储器子系统与I/O子系统,从CPU级发展为整机系统级。在讨论I/O子系统时可以软硬结合,既讨论硬件接口与I/O设备的逻辑组成及工作机理,也介绍软件调用方法与相应的I/O程序设计。因此,这样的组织方法并非简单地将《计算机组成原理》与《汇编语言程序设计》相拼合,而是一种有机的结合,在一定程度上避免了以往在课程先后顺序上所遇到的困难。

本书由俸远祯、王正智主编。俸远祯执笔第一章,并负责全书的统稿;王正智编写第五、六、八章;徐洁编写第二、三、四、七章;俸智丹编写第九、十、十一章。西南交大朱怀芳教授担任主审,他认真地审阅了书稿,提出了许多宝贵而重要的意见。《计算机学科教学计划 1993》配套教材编委会编委,北京航空航天大学杨文龙教授、电子科技大学龚天富教授、重庆大学袁开榜教授等,对本书的编写自始至终地给予了热情的指导和帮助。在编写过程中,还得到了电子科技大学计算机学院有关领导和老师们的关怀和支持。全国工科电子类专业教材《计算机》教材编委会(课程指导委员会)对本书的出版给予了支持和鼓励。谨向上述同志表示衷心的感谢。

作为建立新教学体系的一种尝试,本书肯定是不成熟的,而且编者水平有限。对书中的错误和不足,恳切希望大家批评指正。

编 者

1996年9月

目 录

开 篇	(1)
第一章 绪论	(1)
第一节 基本概念	(1)
一、存储程序工作方式与诺依曼机	(1)
二、信息的数字化表示	(3)
三、数字计算机的特点	(5)
第二节 计算机系统的硬软组成与层次结构	(6)
一、硬件系统	(6)
二、软件系统	(9)
三、层次结构模型	(11)
四、本书的教学组织	(13)
第三节 计算机的性能指标	(14)
第一篇 中央处理机子系统	(17)
第二章 机器指令级的组成	(18)
第一节 中央处理器的总体结构	(18)
一、ALU 部件与寄存器	(18)
二、总线	(20)
三、CPU 内部数据通路	(20)
第二节 CPU 与主存储器间的信息交换	(21)
一、CPU 与主存间的信息交换	(22)
二、主存储器的基本特性	(22)
三、堆栈	(23)
第三节 主机与 I/O 间的数据通路与信息交换方式	(24)
一、数据通路	(25)
二、信息交换的控制方式	(26)
第四节 时序控制方式与时序系统	(29)
一、同步控制方式	(29)
二、同步控制方式的多级时序系统	(30)
三、指令的执行过程	(31)
第五节 典型 CPU 举例	(32)
一、8086/8088 结构	(33)
二、8086/8088 寄存器	(34)
三、8086/8088 的主存储器	(37)

四、8086/8088 的堆栈组织	(40)
习题	(41)
第三章 数据信息表示与指令系统	(43)
第一节 数据信息表示	(43)
一、进位计数制	(43)
二、带符号数的表示	(49)
三、定点数与浮点数	(55)
四、字符表示	(61)
第二节 指令系统基本概念	(63)
一、指令格式	(63)
二、寻址方式	(67)
三、指令类型	(72)
第三节 指令系统举例	(76)
一、8086/8088 寻址方式	(77)
二、8086/8088 指令	(82)
习题	(94)
第四章 CPU 的工作机理与微程序设计	(97)
第一节 模型机总体设计	(97)
一、总体结构	(97)
二、指令系统	(101)
第二节 指令流程与组合逻辑控制器	(109)
一、时序系统	(109)
二、指令流程与微操作时间表	(111)
三、微命令综合	(121)
第三节 微程序控制原理	(123)
一、微程序控制概念	(123)
二、微指令编码方式	(125)
三、微程序的顺序控制	(127)
四、微指令格式	(130)
五、典型微指令举例	(131)
第四节 微程序设计	(133)
一、时序系统	(133)
二、微指令格式	(134)
三、微程序编制	(137)
第五节 典型 CPU 举例	(142)
一、80386/80486	(142)
二、RISC 的概念	(147)
三、MIPS R4000	(149)
四、Intel Pentium	(154)
习题	(156)
第二篇 汇编语言级程序设计	(158)
第五章 汇编语言	(159)

第一节 概述	(159)
第二节 汇编语言语句种类与格式	(161)
第三节 符号定义语句	(163)
一、等值语句	(164)
二、等号语句	(164)
第四节 数据定义语句	(165)
第五节 汇编语言数据与运算符	(168)
一、常数	(168)
二、变量	(169)
三、标号	(170)
四、运算符	(171)
第六节 段结构伪指令	(181)
一、段定义伪指令	(181)
二、段寻址伪指令	(183)
三、段寄存器的装入	(184)
第七节 过程定义伪指令	(187)
第八节 其他伪指令	(187)
一、ORG 伪指令	(187)
二、EVEN 伪指令	(189)
三、INCLUDE 伪指令	(189)
四、PUBLIC 伪指令	(189)
五、EXTRN 伪指令	(190)
六、TITLE 伪指令	(191)
第九节 宏指令	(191)
一、宏指令的使用过程	(191)
二、宏操作符	(194)
三、LOCAL 伪指令	(196)
四、宏嵌套	(198)
五、宏库	(200)
第十节 重复汇编和条件汇编	(201)
一、重复汇编	(201)
二、条件汇编	(203)
习题	(205)
第六章 程序设计基本技术	(209)
第一节 概述	(209)
第二节 分支程序设计	(213)
一、转移指令	(214)
二、分支程序设计举例	(216)
第三节 循环程序设计	(225)
一、循环控制指令	(225)
二、循环程序结构	(231)
三、循环控制方法	(233)

四、多重循环程序设计	(235)
第四节 子程序设计	(238)
一、调用与返回	(238)
二、子程序设计方法	(241)
三、子程序设计举例	(244)
四、子程序的嵌套与递归调用	(254)
第五节 系统功能子程序的调用	(257)
一、DOS 功能子程序的调用	(258)
二、BIOS 功能子程序的调用	(262)
第六节 汇编语言程序的开发	(263)
一、编辑——建立源程序	(263)
二、汇编——生成目标程序	(265)
三、链接——获得可执行文件	(268)
四、调试与运行	(269)
习题	(273)
第三篇 运算功能的实现	(280)
第七章 数值运算	(281)
第一节 算术逻辑运算基础	(281)
一、定点加减运算	(281)
二、溢出判别	(283)
三、移位	(285)
四、十进制运算	(287)
五、逻辑运算	(288)
第二节 算术逻辑运算部件 ALU	(289)
一、加法单元	(289)
二、并行加法器进位链	(291)
三、ALU 举例	(295)
第三节 定点乘法运算	(299)
一、无符号数一位乘	(299)
二、带符号数一位乘	(302)
三、二位乘简介	(305)
四、阵列乘法器	(308)
第四节 定点除法运算	(309)
一、无符号数一位除	(309)
二、带符号数一位除	(314)
三、阵列除法器	(315)
第五节 浮点算术运算	(316)
一、浮点加减运算	(317)
二、浮点乘法运算	(321)
三、浮点除法运算	(323)
四、浮点运算器举例	(324)
第六节 开平方运算	(329)

习题	(331)
第八章 非数值运算	(333)
第一节 串操作	(333)
一、串操作指令	(333)
二、串操作指令使用示例	(335)
三、串操作应用举例	(338)
第二节 表的处理	(341)
一、排序	(341)
二、查找	(343)
第三节 代码转换	(346)
一、二进制数或十六进制数与其对应的 ASCII 码相互转换	(347)
二、十进制数与二进制数之间的相互转换	(350)
习题	(355)
第四篇 存储与输入输出子系统	(361)
第九章 存储系统	(362)
第一节 半导体存储器芯片	(362)
一、静态 RAM 芯片 (SRAM)	(363)
二、动态 RAM 芯片 (DRAM)	(367)
三、半导体只读存储器芯片	(370)
第二节 主存储器组织	(374)
一、主存储器逻辑设计	(374)
二、动态存储器的刷新	(376)
三、主存储器与 CPU 的连接	(378)
第三节 存储系统组织	(380)
一、三级存储体系及其存取方式	(380)
二、高速缓存 (Cache)	(383)
三、虚拟存储器	(386)
四、并行主存系统	(390)
第四节 校验技术	(391)
一、奇偶校验	(392)
二、海明校验	(393)
三、循环校验码 (CRC)	(394)
习题	(395)
第十章 接口与总线	(397)
第一节 概述	(397)
一、接口的功能与分类	(397)
二、主机与外围设备间的连接方式	(400)
三、总线的功能与分类	(402)
四、接口的编址与 I/O 指令	(405)
第二节 直接程序控制方式	(407)
第三节 程序中断方式	(409)
一、中断方式及其应用	(409)

二、中断源与中断向量	(411)
三、中断的过程	(415)
四、中断接口逻辑	(417)
五、中断接口举例	(423)
第四节 DMA 方式	(428)
一、DMA 方式的一般概念	(428)
二、DMA 过程	(429)
三、DMA 接口组成	(430)
第五节 通道与 IOP	(441)
一、通道	(441)
二、输入输出处理机与外围处理机	(444)
第六节 系统总线	(446)
一、概述	(446)
二、总线的组成	(448)
三、总线操作时序	(450)
四、典型总线举例	(453)
习题	(454)
第十一章 I/O 设备与 I/O 程序设计	(455)
第一节 概述	(455)
一、I/O 设备的分类	(455)
二、工作机制中的几个层次	(457)
第二节 键盘	(458)
一、键的结构	(458)
二、键盘组织结构	(459)
三、IBM-PC 机键盘及软件界面	(462)
第三节 打印机	(464)
一、概述	(464)
二、针式点阵打印机	(465)
三、打印机管理软件	(469)
四、激光打印机和喷墨式打印机简介	(470)
第四节 显示器	(472)
一、概述	(472)
二、CRT 显示器工作原理	(474)
三、显示器软件调用	(482)
第五节 磁表面存储器	(483)
一、外存储器概述	(483)
二、磁表面存储原理	(484)
三、磁记录编码方式	(486)
四、硬磁盘存储器	(489)
五、软磁盘存储器	(495)
六、磁带存储器简介	(498)
第六节 光盘存储器	(499)
习题	(501)

附录 A 8086/8088 指令系统汇总表 (503)

附录 B MASM5.0 出错信息 (515)

参考文献 (520)

开 篇

第一章 绪 论

通常所讲的计算机，其全名是电子式数字计算机，它是一种能存储程序，能自动连续地对各种数字化信息进行算术、逻辑运算的快速工具。在这一定义中包含两个重要的基本概念：信息数字化，存储程序工作方式。本章一开始就强调它们，作为了解计算机组成及工作机制的基本出发点。

计算机系统是由硬件与软件组成的综合体，人们常采用层次结构观点去描述系统的组成与功能，分层次地分析与设计计算机系统。本章在简要叙述计算机系统的硬件、软件组成之后，将分别从程序设计角度及系统内部有机组成角度，介绍两种常用的层次结构模型。然后从教学组织的角度，再进一步强调有关的几种层次。

第一节 基 本 概 念

一、存储程序工作方式与诺依曼机

1945年，冯·诺依曼在领导设计EDVAC计算机的过程中，提出了现代数字计算机的若干设计思想，被后人称为诺依曼体制，这是计算机发展史中的一个里程碑。采用诺依曼体制的计算机就被称为诺依曼机。几十年来计算机体系结构发生了许多演变，但诺依曼体制的核心理念——存储程序工作方式，仍是普遍采用的原则，绝大多数实用的计算机仍属于诺依曼机。我们将诺依曼体制中至今仍广泛采用的要点归纳如下。

1. 采用二进制代码表示数据和指令

计算机是处理信息的工具，所处理的信息泛称为数据。计算机的具体工作表现为执行程序，而程序的最终可执行形态是指令序列，即若干用数字代码表示的机器指令，它们是产生控制命令的信息基础。数据和指令都采用二进制代码表示，则它们在外形上并无区别，都是由0或1组成的代码序列，只是各自约定的含义不同而已。采用二进制，使信息数字化容易实现，可以用二值逻辑电路进行处理，这可称之为处理功能逻辑化。不仅数据，程序本身也可以作为被处理的对象，例如对源程序进行编译，就是将源程序当作被加工处理的对象。

2. 采用存储程序工作方式

这是数字计算机不同于其他计算工具之处，它意味着：

(1) 事先编制程序。为了用计算机求解问题，需要编制程序，它告诉计算机需要做哪些事，按什么步骤去做，并提供所要加工处理的数据。一件事往往要分步去做，每一步要求计算机执行的操作命令称为一条指令。计算机最终执行的程序，包含指令序列和有关的原始数

据。

(2) 将程序存储于计算机的存储器之中。

(3) 计算机在运行时将自动地、连续地从存储器中依次取出指令加以执行。因此，计算机的工作体现为执行程序，计算机功能的扩展在很大程度上取决于所存储程序的扩展。

存储程序工作方式是所有计算机都需采用的方式，是其他具体工作方式的基础，但传统诺依曼机在执行程序时，始终是以控制信息流（指令流）为驱动程序执行的因素，而数据信息流（数据流）则被动地被调用处理，因而称为控制流驱动方式。即按照指令的执行序列，依次读取指令，根据指令所含的控制信息，调用数据进行运算处理。

为了控制指令序列的执行顺序，可设置一个程序（指令）计数器 PC（Program Counter），让它存放当前指令所在存储单元的地址。如果是顺序执行，则读取一个存储单元的指令代码之后，PC 内容加 1。如果需要程序转移，则将转移后的地址送入 PC。因此 PC 就像一个指针，指引着程序的执行。虽然程序与数据都采用二进制代码，在外形上没有区别，但可按照 PC 的内容作为地址读取指令，再按照指令给出的操作数地址去读取数据。由于多数情况下程序将顺序执行，所以大多数指令需要依次地紧挨着存放，除了个别立即使用的数据可以紧挨着指令存放外，一般将指令与数据分别存储于该程序区中的不同区域。

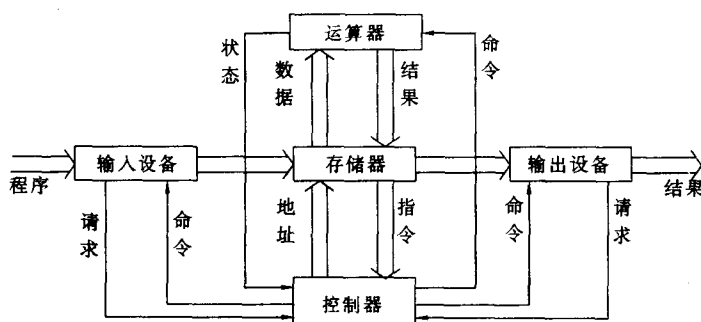


图 1-1 数字计算机的简单框图

3. 由运算器、存储器、控制器、输入装置、输出装置等五大部件，构成计算机硬件系统

如图 1-1 示，所编好的程序（包含指令与数据）通过输入设备送入计算机的存储器中。例如用键盘，将程序语句的字符转换为机器所能识别的二进制字符编码，然后送入存储器。

存储器按一定顺序存放所输入的程序，计算机处理所获得的结果也作为数据存放在存储器中。这就要求将存储器分成若干单元，并给每个单元分配一个地址，因此存储器的一个重要特性是能按地址存放或读取内容。对存储器而言，寻址的依据是地址码，存取的内容称为数据。

启动运行后，计算机将自动连续地执行程序。控制器向存储器送出某个地址码，从指定的存储单元中读取一条指令，送入控制器进行识别，分析该指令要求做何操作，操作数在何处。根据该指令的要求，控制器发出相应的操作命令，例如将某存储单元中存放的操作数取出，送往运算器进行所指定的运算处理，再将运算结果送回存储器中指定单元。运算过程中的某些状态信息，如有无进位、结果是否为 0、结果正负、是否溢出等，可送回控制器作为下一步操作的参考依据。

计算机执行完一段程序后, 将获得所需的处理结果, 再通过输出设备将结果送出。例如通过显示器在屏幕上显示程序运行情况与结果, 或通过打印机将结果打印出来。

信息的输入、输出, 存储器的读写, 信息的运算处理等, 都必须在控制器的控制下有序地进行。控制器控制着整个计算机的工作, 决定在什么时候发出什么命令、进行什么操作。例如向存储器发出读命令或写命令及单元地址码, 向运算器发出某种四则运算命令。在输入/输出设备与控制器之间常采取这样一种协调方式: 当输入/输出设备做好相应准备之后, 向中央控制器发出请求信号, 然后控制器发出输入命令或输出命令。

早期的计算机常设置一个专门的控制台, 操作人员可通过控制台启动或停止计算机的运行, 或设置有关的运行参数, 或在运行过程中进行某些必要的干预。现在的计算机则往往通过键盘实现上述操作, 取代了传统的控制台。

二、信息的数字化表示

计算机是处理信息的工具。如前所述, 在计算机中的信息可分为两大类: 控制信息流、数据信息流。前者是指指令代码序列, 及以此为基础所产生的控制信号(微命令)序列, 它们是控制计算机工作的有关信息。数据流是计算机加工处理的对象, 可分为数值型与非数值型(如字符、文字、图像、声音、逻辑信息等)。

为什么称为数字计算机呢? 就是因为计算机中的上述信息都采用数字化表示, 简单地讲它有两层含义。

1. 在计算机中的各种信息用数字代码表示

例 1-1 用数字代码表示数值大小, 正负数符也用代码表示, 如约定符号位为 0 表示数为正, 符号位为 1 表示数为负。

例 1-2 用数字编码表示字符, 以字符为基础可表示范围广泛的各种文字, 以及程序设计语言等。西文字符、常用算符与控制符等, 可用七位编码表示, 每个字符编码占一个 8 位存储单元(即一个字节单元)。对于中文字符, 可用二个字节的编码表示。

例 1-3 程序在计算机中的最终可执行形态是指令序列, 每种指令按约定的指令格式, 用一组数字代码表示。其中往往分为: 操作码字段, 操作数(或地址)字段。例如用 0000 表示传送, 0001 表示加, 0010 表示减……等操作命令。用某种程序设计语言编写的程序, 先以字符编码形式输入计算机, 再转换为指令代码。

例 1-4 将设备的工作状态抽象化, 用不同编码分别表示, 如: 00 表示空闲, 01 表示忙, 10 表示完成一次操作。计算机根据某设备的状态编码, 即可判定该设备所处状态。

例 1-5 用点阵近似地表示图像。

与字符相比, 图像信息是随机分布的, 规律性差。但可以将一幅图像细分为若干点, 用这些点的组合逼近真实图像。如果分得足够细即点数很多, 则在人的视觉中, 这幅由许多点组成的图像几乎是连续的。相应地在计算机中用一位数字代码表示一个像点, 如用 1 表示一个亮点, 用 0 表示一个暗点。再按一定的扫描规律, 将这些位(点)的信息以数字形式组织并存储起来, 就可以对图像进行处理。

例 1-6 用数字表示声音。

在传统的电话装置中, 受话器将声波转换为模拟信号形式的电流波, 如要将其数字化, 则可按一定频率对电流波进行采样, 即等时间间隔地对电流波进行测量, 将每次测得的电流幅

值用一个相应的数字量表示。如果采样频率足够高，则所获得的数字信息能逼真地保持声波信息，还原后发话器能逼真地再现原来的声音。采用数字化转换后，就可以用计算机对声音信息进行各种处理。

例 1-7 用数字代码表示逻辑信息。

计算机中的运算可分为：算术运算、逻辑运算。常用二值逻辑描述与处理逻辑问题，如用逻辑变量表示逻辑条件，1 表示该条件成立，0 表示条件不成立。将逻辑处理的可能结果分解为若干个逻辑命题，1 表示命题真，0 表示命题假，则计算机可以用数字逻辑部件实现其各种功能，对数据信息实行逻辑运算。例如一个多数表决器，每个投票者的意见可用一个逻辑变量表示，1 表示赞成，0 表示反对；表决结果也用一个逻辑变量表示，为 1 表示通过，为 0 表示否决。

计算机可以处理的信息类型极其广泛，我们不可能也没有必要穷举所有的例子。通过上面的几个例子，希望读者能举一反三。请注意，善于用约定的数字代码去表示各种需要描述的信息，这是从事计算机技术工作的重要基础。

2. 在物理机制上，数字代码以数字型信号表示

从物理实现这一层次看，数字代码需用电信号去体现，才能用电路部件实现信息的传送与处理。电信号可分为两大类型，即模拟信号及数字信号。

模拟信号是一种在时间上连续的信号，用信号的某些参量去模拟信息，例如用电信号的幅值去模拟数值的大小，所以称为模拟信号或模拟量。当通过传感器将一些非电量转换为电信号时，最初获得的常常是模拟信号，例如受话器将声音转换为电流波，温度传感器将温度转换为相应的电压或电流信号等。需要将模拟信号转换为数字信号，数字计算机才能进行处理。

数字信号是一种在时间上或空间上离散的信号。单个数字信号仅取有限的几种状态，目前常用的是二值逻辑，非 0 即 1。依靠彼此离散的多位信号的组合，能表示广泛的信息，而处理时则可以逐位处理。

例 1-8 用一串脉冲型数字信号表示一个四位数 1101。

数字信号的一种形式是脉冲，如图 1-2 示，它呈脉动形状，随时间的分布不连续。脉冲信号的特点是，信号电平向正方向（或负方向）跳变，并维持较短的时间，然后回到原来的状态。脉冲信号依靠边沿作用，可利用其上升边沿或下降边沿表示某一时刻，起定时作用。也可利用脉冲的有无表示 1 或 0，如在约定时间内有脉冲为 1，无脉冲为 0。如图 1-2 示，通过同一根信号线在一段时间内发出一组脉冲序列，表示数字代码 1101。

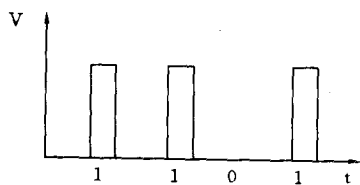


图 1-2 用一串脉冲表示数字代码

例 1-9 用一组电平型数字信号表示四位数 1101。

数字信号的另一种形式是电平信号，利用信号电平的高低表示不同状态值，通常定义高电平（如 +5V）表示 1，低电平（如 0V）表示 0。由于是依据电平的高低起作用，所以电平信号维持某一状态的时间相对要长一些。如图 1-3 示，用一组信号线表示多位数，每一根线产生一位电平信号，各线（位）之间在空间上是独立的，即离散的。事先约定各位的高低顺序，如图表示 1101。