

周明德 主编

微型计算机

BM-PC(0520)系统

原理及应用

8088-8086
BASIC
BASE

微 型 计 算 机

IBM-PC (0520) 系 统

原 理 及 其 应 用

周明德 宋翰涛 冯 云 张喜英编

周明德 主编

清 华 大 学 出 版 社

内 容 简 介

本书以我国高档微型机的优选系列——长城0520 (IBM-PC兼容)系列为背景,系统地、完整地叙述了微型计算机的原理和应用。全书分为四个部分:第一部分为原理部分,介绍了计算机的基础知识;8088的结构、指令和汇编语言的程序设计;8088的时序;介绍了输入输出和中断的原理,数据传送的方法,全面地介绍了8088的中断分式。第二部分通过一种操作系统——PC-DOS,介绍了DOS的结构和使用,重点从DOS的命令和功能调用两方面详细介绍了PC-DOS的使用。第三部分提供了IBM-PC上的各种BASIC语言;第四部分介绍了一种功能很强、应用广泛、使用方便的关系数据库——汉字dBASE II。全书有大量的实例可供学习和模仿。

此书可作为长城0520以及IBM-PC及兼容机的培训教材,也可作为以IBM-PC为样机普及微型计算机的教材,适用于各种技术培训班和高等学校。

微 型 计 算 机 IBM-PC (0520) 系统原理及应用

周明德 主编

☆

清华大学出版社出版

北京 清华园

清华大学印刷厂排版

北京市联华印刷厂印装

新华书店北京发行所发行

☆

开本 787×1092/16 印张: 42 1/4 字数: 1054千字

1985年9月第1版 1985年9月第6次印刷

印数: 265000~311000 定价: 9.50元

ISBN 7-302-00243-6/TP·97

前 言

今年以来，微型计算机的发展十分迅猛。当我们选择计算机时，应该考虑一些什么原则呢？

第一，我们所选择的机型，技术上应该是先进的，在世界上处于主流地位，这样就可以有大量的、广泛的软硬件支持。

长城 0520 系列，IBM-PC 或 PC/XT 以及兼容机是符合这样要求的。

到 1983 年底，我国微型机的主流是以 Z-80CPU 为核心的，而 IBM-PC 是以 Intel 8088 作为 CPU。Intel 8088 在性能上比 Z-80 CPU 要高一个数量级，这主要反映在：

1. 寻址方式上，8088 除了有变址寻址外，还有间址加变址寻址，更适用于高级语言中的数组和记录等数据结构。

2. 在指令系统上功能更强。8088 的内部是 16 位的，即它的寄存器是 16 位的，运算也是 16 位的。所以，能完成广泛的 16 位的数据传送；16 位的算术运算，包括各种寻址方式的加减法运算，特别是增加了 16 位的乘法和除法指令；16 位的逻辑运算指令；16 位的移位和循环操作等等。

3. 8088 的地址线为 20 条，直接寻址能力可达 1M 字节，这样就便于大型软件的使用，特别是汉字处理。

4. 输入输出指令的功能也扩大了，可以做到 16 位数据的输入和输出；端口寻址也可扩大到 64K 个。

5. 中断功能更强。最多可达到 256 个中断源，而且有内部中断指令，溢出中断指令，以及单步工作方式。

以 8088 CPU 为核心的长城 0520 系列，IBM-PC 及兼容机的内存容量可扩展到 10K 字节，可配有 10M 字节的硬盘（温盘），在性能上确实是先进的、优越的，而且在世界上处于主流地位。

第二，我们所选择的机型应该有一个完整的系列，它们在软件上应是兼容的。

应用的范围是广泛的，各种各样的，有的应用场合只要用单片机或单板机就可以了；有的就要求用 0520A 系统；有的就可能要求有多用户、多任务系统；有的就要求有一个局部网络等等。这个系列在软件上应该是兼容的。

IBM 公司推出了一个完整的系列，可根据不同的要求加以选择。

第三，要有强大的、良好的技术服务。通常在购买机器之前要进行咨询；买了机器以后，就要求有高质量的培训来培养人材。使用机器的人的素质，是充分发挥机器效益的关键。能否获得高质量的培训，是能否迅速地把机器应用起来的先决条件。机器在经过了一段时间运行以后总是会坏的，这时，能否得到及时的良好的维修是十分重要的。为了开发应用，能否得到源源不断的软硬件支持也至关重要。

长城 0520 系列，在全国有近 3000 人的从咨询、技术培训、维修到应用开发的完整

技术服务。

为了促进我国的优选系列——长城 0520 的广泛普及，我们编写了此教材。在编写时，我们充分考虑了培训教材的特点，一方面要脱离技术手册、使用说明书的框框，按照教材的要求，要有系统性、完整性，由浅入深、循序渐进；另一方面又充分突出实用性，在教材中引进了大量的实例以供学习和模仿。

这本教材贡献给读者四方面的内容：通过 IBM-PC，叙述微型计算机的原理，提供一种学习和使用微型计算机的基本工具——汇编语言；介绍一种操作系统——PC-DOS 的使用（命令和功能调用）；介绍一种高级语言——IBM-PC BASIC 语言；介绍一个功能很强，用途很广而又容易使用的关系数据库——汉字 dBASE II。

此教材是长城 0520 的用户培训教材，也是以 Intel 8088 为典型的普及微型计算机的教材，适用于各种技术培训班和高等学校。

本书的第一部分由周明德同志编写，第二部分由宋翰涛、关维忠、张雪兰同志编写，第三部分由冯云同志编写，第四部分由张喜英同志编写，全书由周明德同志主编。

限于编者的水平，且时间十分仓促，缺点与错误在所难免，敬请读者批评指正。

周明德

1984.6

目 录

第一部分 IBM-PC 系统原理及 Intel 8088 汇编语言程序设计	1
第一章 概述	1
第一节 引言	1
第二节 计算机中的数和编码系统	2
第三节 计算机基础	14
第四节 计算机的硬件和软件	36
第五节 微型计算机的结构	38
第六节 Intel 8088 的结构	40
第七节 IBM-PC 的基本配置	49
第二章 8088 的指令系统和汇编语言	54
第一节 8088 的寻址方式	54
第二节 8088 中的标志寄存器	58
第三节 8088 的指令系统	60
第四节 IBM-PC 宏汇编	115
第五节 汇编语言程序举例	141
第六节 建立、汇编、连接、运行和调试汇编语言源程序	163
第三章 8088 的时序	213
第四章 输入和输出	219
第一节 输入输出的寻址方式	219
第二节 CPU 与外设数据传送的方式	220
第五章 中断	230
第一节 引言	230
第二节 最简单的中断情况	231
第三节 中断优先权	233
第四节 8088 的中断方式	236
第二部分 IBM-PC DOS	239
第一章 DOS 与文件简介	239
第一节 DOS 简介	239
第二节 文件简介	240
第二章 启动 DOS	246
第一节 如何启动 DOS	246
第二节 程序的自动执行	249
第三节 软盘和驱动器的兼容性	249
第四节 单驱动器系统	250
第五节 建立 DOS 2.00 的工作盘片	250
第六节 DOS 用的一些键	256
第七节 DOS 编辑键	258

第三章 DOS 命令	260
第一节 简介	260
第二节 DOS 命令类型	260
第三节 命令格式表示法	261
第四节 FORMAT (格式化) 命令	262
第五节 DIR (目录) 命令	263
第六节 关于盘维护命令	265
第七节 RENAME (REN 重新命名) 命令	271
第八节 TYPE (显示) 命令	271
第九节 DATE (日期) 命令	272
第十节 TIME (时间) 命令	272
第十一节 MODE (模式) 命令	273
第十二节 批处理命令	275
第十三节 DOS 命令一览表	277
第四章 树结构的目录	282
第一节 简介	282
第二节 目录类型	282
第三节 当前目录	282
第四节 指定文件的路径	283
第五节 目录命令	284
第五章 行编辑程序	286
第一节 简介	286
第二节 如何启动行编辑程序	286
第三节 EDLIN 命令的参数	288
第四节 插入行命令	289
第五节 显示行命令	290
第六节 删除行命令	292
第七节 编辑行命令	294
第八节 替换文字段命令	295
第九节 查找文字段命令	297
第十节 附加行命令	298
第十一节 写入行命令	299
第十二节 结束编辑程序命令	299
第十三节 退出编辑程序命令	299
第十四节 复制行命令	300
第十五节 移动行命令	300
第十六节 页命令	301
第十七节 传输行命令	301
第十八节 EDLIN 命令总结	302
第六章 连接程序	303
第一节 简介	303
第二节 文件	303
第三节 段、组和类的定义	304

第四节	命令提示.....	305
第五节	命令提示详述	305
第六节	如何启动连接程序	309
第七节	连接阶段举例	310
第七章	中断和子程序调用	314
第一节	DOS 中断.....	314
第二节	子程序调用	316
第三节	中断和子程序调用举例	324
附录 1	DOS 技术资料.....	330
附录 2	DOS 盘分配.....	333
附录 3	DOS 控制块和工作区.....	337
第三部分	IBM-PC BASIC 语言	343
第一章	概述	343
第一节	PC 系统 BASIC 语言概况	343
第二节	使用 BASIC 时的操作	344
第二章	语言的基本规定	353
第一节	BASIC 的程序行.....	353
第二节	BASIC 的字符集和保留字.....	354
第三节	常量、变量、数组	355
第四节	数值的精度类型	359
第五节	表达式和运算符	361
第三章	基本语句和函数	367
第一节	赋值语句	367
第二节	循环语句	376
第三节	显示和打印	379
第四节	类型说明语句和函数	393
第五节	数值函数	400
第四章	调用和转移	411
第一节	转移语句	411
第二节	调用语句	422
第三节	程序链接	431
第五章	图形功能	434
第一节	辅助性语句、函数和变量	435
第二节	有关图形颜色的语句和函数	439
第三节	绘图语句	445
第六章	文件和输入输出语句	454
第一节	有关文件操作的几条命令	455
第二节	顺序文件的输入、输出	459
第三节	随机文件的输入和输出	468
附录 1	字符串函数	474
附录 2	附加的部分常用 BASIC 命令	478
附录 3	错误信息及其意义	481
附录 4	ASCII 码表	487

第四部分 汉字 dBASE I	491
第一章 概述	491
第一节 数据库的基本概念	491
第二节 dBASE I 关系式数据库管理系统	497
第二章 最基本的数据库操作	511
第一节 建立数据库文件	511
第二节 数据库数据的输入	513
第三节 数据库的显示及定位	517
第三章 dBASE I 的常数、变量、表达式及函数	525
第一节 常数	525
第二节 变量	525
第三节 表达式	528
第四节 函数	532
第四章 数据库中数据的修改与删除	538
第一节 数据的修改	538
第二节 数据的删除	544
第五章 数据库的排序、索引及查寻	546
第一节 数据库的排序	546
第二节 数据库的索引	548
第三节 数据库的查寻	551
第六章 数据库的复制、结构的移转、数据的转移及数据结构的修改	556
第一节 数据库的复制	556
第二节 数据结构的转移	562
第三节 数据的转移	563
第四节 数据结构的修改	568
第七章 双工作区操作及数据库的连接、数据的更新	575
第一节 双工作区操作	575
第二节 数据库的连接	581
第三节 数据的更新	584
第八章 自动计数与求和及几个通用命令	587
第一节 自动计数	587
第二节 求和	587
第三节 几个通用命令	589
第九章 数据文件摘要及数据的报表输出	590
第一节 数据文件摘要	590
第二节 建立报表格式	593
第三节 报表格式的调用	601
第十章 dBASE I 程序设计	604
第一节 程序(命令文件)的建立和执行	604
第二节 先进的结构式程序设计语言	605
第三节 程序交互性命令	614
第四节 用于程序设计的几个辅助命令	617
第五节 格式控制命令	618

第六节	SET 命令.....	628
第七节	程序实例	633
附录 1	全屏幕编辑控制键一览表	645
附录 2	dBASE III (V2.3D) 命令一览表	648
附录 3	SET 命令一览表.....	652
附录 4	新版本 (V2.4) 所提供的新函数、新命令及对原有命令的扩充.....	654
附录 5	错误信息及改正方法一览表	665

第一部分 IBM-PC 系统原理及Intel 8088 汇编语言程序设计

第一章 概 述

第一节 引 言

微处理器 (Micro processor) 和微型计算机 (Microcomputer) 自 70 年代初崛起以来, 发展极为迅猛, 在短短的十年里已经经历了四代。应用的发展也极为迅速, 已经渗透到各个技术领域, 渗透到文化、教育领域, 渗透到家庭、日常生活的各个领域。

十多年来微型计算机的主要发展趋势是两大方面。

(一) 提高性能

微处理器片子的集成度越来越高, 几乎每两年翻一番, 且性能提高一个数量级。拿 Intel 公司的产品来说, 1971 年的 4004, 集成度为 2500 个/片; 1976 年的 8085, 集成度为 9000 个/片; 1978 年的 8086, 集成度为 29000 个/片; 1980 年的 iAPX43201, 集成度为 100000 个/片。

现在 16 位、32 位的微处理器已经大量出现。

半导体存储器的集成度也越来越高, 64K 位的早已有商品出售, 256K 位的也已经出现。

软磁盘的存储密度日益提高, Winchester 硬磁盘 (温盘) 的研制成功, 为微型机系统提供了一种价格低廉而存储容量很大的外存储设备, 大大扩大了微型机系统的功能。

各种微型机的操作系统, 如 CP/M、MP/M、CP/M86、Unix (以及各种变型)、P 系统等等, 以及在各种操作系统支持下的大量高级语言, 象雨后春笋般涌现, 从而大大丰富了微型计算机的系统软件。

为各种微型计算机配制硬件配件、选件, 特别是编制应用软件, 出售软件包的公司也大量涌现, 成为一种新的行业。现在, 没有应用软件支持的微型机已成为裸机, 大大影响销售量。

总之, 微型计算机系统的性能已经赶上甚至超过了 70 年代小型机的水平。

(二) 降低价格

微型机发展的另一个重要趋势是降低价格。一方面片子的价格降低, 另一方面制造了各种价格低廉的微型机。例如, 利用家用电视机及录音机, 能运行 BASIC 语言的机器, 价格在 \$100 以下。

价格低廉，是微型机真正能够在各行各业应用，能深入到办公室自动化，甚至深入家庭，形成个人计算机（Personal Computer）的重要条件。

目前，各种微处理器片子的年产量为数千万片；各种微型计算机的年产量和销售量为数百万台。

但微型计算机的普及应用方兴未艾，正以越来越高的势头飞速发展。

微型计算机的普及应用，已经引起了各种科学技术领域的深刻变革，甚至引起了生活领域的变革。越来越多的人在关心在议论第四次工业革命的到来。

IBM 公司生产的个人计算机简称 IBM-PC，是 1981 年下半年推出来的，但是，由于它的性能价格比较好；也由于 IBM 公司在计算机行业中的地位，赢得了用户的信任；也由于有上千家公司围绕 IBM-PC 做硬件的配件、选件、扩充件，配制各种系统软件和语言，出售各种软件包；也由于 IBM 公司计划在微型机方面形成系列，考虑到软件的兼容、标准化与系列化，以及与大型机在软件上的兼容性等等。因而 IBM-PC 发展十分迅速，在 83 年大约销售了 40 万台，预计 84 年将生产 200 万台。

电子工业部计算机管理局已选定与 IBM-PC 兼容的长城 0520 机，作为我国 8 位微型机的重点机件，要配上汉字，要能连成网络。在研制、生产、应用服务等方面形成一个完整的体系，近几年内在国内大量推广与普及。

为了适应这样的形势，本教材以 IBM-PC 机为样机，从应用的角度叙述微型计算机的原理；介绍 Intel 8088 的结构、指令系统和汇编语言的程序设计；介绍 PC-DOS 的主要命令及其使用，以及 PC-DOS 所提供的系统调用程序的使用；介绍 IBM 中的主要 BASIC 语言及其应用；又向大家提供一个功能较强，使用方便的关系数据库-dBASE II。这些，我们都是从工程技术人员如何把微型计算机应用于本技术领域的角度来论述问题的。

第二节 计算机中的数和编码系统

一、计算机中的数制

计算机最早是作为一种计算工具出现的，所以它的最基本的功能是对数进行加工和处理。数在机器中是以器件的物理状态来表示的，一个具有两种不同的稳定状态且能相互转换的器件，就可以用来表示一位二进制数。所以，二进制数的表示是最简单而且可靠的，另外，二进制的运算规则也是最简单的。因此，目前在计算机中，数几乎全是用二进制表示的。

（一）二进制数

一个二进制数，具有以下两个基本特点：

1. 具有两个不同的数字符号，即 0 和 1。
2. 逢二进位。

由于是逢二进位的，所以同一个数字符号在不同的数位所表示的值是不同的。例如

111. 11

小数点左边第一位的“1”代表的值就是它本身；小数点左边第二位的“1”，是由第一位逢二进上来的，所以它的值为 1×2^1 ；则左边第三位的“1”的值为 1×2^2 ；小数点右面第一

位的“1”代表 1×2^{-1} ；右面第二位的“1”代表 $1 \times 2^{-2} \dots \dots$ 。

可见，每一个数位有一个基值与之相对应，这个基值就称为权。

一个二进制数的权，小数点左面的是 2 的正次幂；小数点右面的是 2 的负次幂。

一个二进制数的值，就可以用它的按权展开式来表示。即

$$(111.11)_2 = 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} = (7.75)_{10}$$

$$(1011.101)_2 = 1 \times 2^3 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-3} = (11.625)_{10}$$

于是，一个任意的二进制数可以表示为

$$(B)_2 = B_{n-1} \times 2^{n-1} + B_{n-2} \times 2^{n-2} + \dots + B_1 \times 2^1 + B_0 \times 2^0 \\ + B_{-1} \times 2^{-1} + B_{-2} \times 2^{-2} + \dots + B_{-m} \times 2^{-m}$$

$$= \sum_{i=-m}^{n-1} B_i \times 2^i$$

其中 n 为整数部分的位数， m 为小数部分的位数； B_i 的值为 0 或 1 取决于一个具体的数。

(二) 十六进制数

目前，大部分微型机的字长是 4 的整数倍，所以广泛地采用十六进制数来表示。一个十六进制数的特点为：

1. 具有十六个数字符号，采用 0—9 和 A—F。这 16 个数字符号与十进制数和二进制数之间的关系如表 1—1 所示。

表 1—1

十 进 制	十六进制	二 进 制	十 进 制	十六进制	二 进 制
0	0	0000	9	9	1001
1	1	0001	10	A	1010
2	2	0010	11	B	1011
3	3	0011	12	C	1100
4	4	0100	13	D	1101
5	5	0101	14	E	1110
6	6	0110	15	F	1111
7	7	0111	16	10	10000
8	8	1000			

2. 逢 16 进位

由于是逢 16 进位，所以同一个数字符号，在不同的数位所代表的值是不同的，即每一个数位有一个权与之相对应。小数点左边的权是 16 的正次幂，小数点右边的权是 16 的负次幂。一个 16 进制数的值，可以用它的按权展开式来表示。

$$(32)_{16} = 3 \times 16^1 + 2 \times 16^0 = (50)_{10}$$

$$(FF)_{16} = 15 \times 16^1 + 15 \times 16^0 = (255)_{10}$$

$$(3AB.11)_{16} = 3 \times 16^2 + 10 \times 16^1 + 11 \times 16^0 + 1 \times 16^{-1} + 1 \times 16^{-2} \\ = (939.0664)_{10}$$

于是一个任意的 16 进制数 D 可以表示为:

$$(D)_{16} = D_{n-1} \times 16^{n-1} + D_{n-2} \times 16^{n-2} + D_1 \times 16^{-1} + D_0 \times 16^0 \\ + \dots D_{-1} \times 16^{-1} + D_{-2} \times 16^{-2} + \dots D_{-m} \times 16^{-m} \\ = \sum_{i=-m}^{n-1} D_i \times 16^i$$

其中, n 是整数部分的位数, m 是小数部分的位数; D_i 的值在范围 0—9 和 A—F 中。

但是, 在机器中, 数并不是用 16 进制表示的, 由于一开始提到的理由, 在机器中数仍是用二进制表示的。由于二进制和十六进制之间存在着一种特殊关系, 即 $2^4 = 16$ 。

于是一位十六进制数, 可以用四位二进制数表示, 它们之间存在着直接的而又唯一的对应关系, 如表 1—1 所示。

因此, 二进制数和十六进制数之间的转换是十分简捷而又方便的。

1. 16 进制转换为二进制

不论是 16 进制的整数或小数, 只要把每一位 16 进制的数用相应的四位二进制数代替, 就可以转换为二进制数

例: $(3AB)_{16}$ 可转换为

$$\begin{array}{ccc} 3 & A & B \\ | & | & | \\ 0011 & 1010 & 1011 \end{array}$$

$$\therefore (3AB)_{16} = (0011 \ 1010 \ 1011)_2 = (11 \ 1010 \ 1011)_2$$

$(0.7A53)_{16}$ 可转换为

$$\begin{array}{cccc} 0.7 & A & 5 & 3 \\ | & | & | & | \\ 0111 & 1010 & 0101 & 0011 \end{array}$$

$$\therefore (0.7A53)_{16} = (0.0111 \ 1010 \ 0101 \ 0011)_2$$

2. 二进制转换为 16 进制

二进制的整数部分由小数点向左, 每四位一分, 最后不足四位的前面补 0; 小数部分由小数点向右, 每四位一分, 最后不足四位的后面补 0。然后把每四位二进制数用相应的 16 进制数代替, 即可转换为 16 进制数。

例: $(11011 \ 1110 \ 0011.1001 \ 01111)_2$ 可转换为

$$\begin{array}{ccccccc} 0001, & 1011, & 1110, & 0011, & 1001 & 0111 & 1000 \\ | & | & | & | & | & | & | \\ 1 & B & E & 3 & 9 & 7 & 8 \end{array}$$

$$\therefore (11011 \ 1110 \ 0011.1001 \ 01111)_2 = (1BE3.978)_{16}$$

总之, 数在机器中是用二进制表示的, 但是, 一个二进制数书写起来太长, 且容易出错。而目前大部分微型机的字长是四位的, 八位的, 十六位或三十二位的, 都是四的整数倍, 我们在书写时用十六进制来表示。一个字节 (八位) 就可以用两位十六进制数表示, 两个字节 (十六位) 可以用四位十六进制数表示等, 书写方便且不容易出错。

二、二进制编码

如上所述, 在计算机中, 数是用二进制表示的。而计算机又应能识别和处理各种字符, 如大小写的英文字母, 标点符号, 运算号等等, 这些又如何表示呢? 由于计算机中的基本编

理器件是具有两个状态的器件，所以各种字符又只能用若干位的二进制码的组合来表示，这就称为二进制编码。

(一)、二进制编码的十进制数

因二进制数实现容易、可靠，二进制的运算规律十分简单。所以，在计算机中采用二进制。但是，二进制数不直观，于是在计算机的输入和输出时通常还是用十进制数表示。不过这样的十进制数，要用二进制编码来表示。

一位十进制数用四位二进制编码来表示，表示的方法可以极多，较常用的是 8421 BCD 码，表 1—2 列出了一部分编码关系。

8421BCD 码有十个不同的数字符号，且它是逢“十”进位的，所以，它是十进制数；但它的每一位是用四位二进制编码来表示的，因此，称为二进制编码的十进制数 (BCD-Binary Coded Decimal)。

表 1—2 BCD 编码表

十进制数	8412 BCD 码	十进制数	8421 BCD 码
0	0000	8	1000
1	0001	9	1001
2	0010	10	0001 0000
3	0011	11	0001 0001
4	0100	12	0001 0010
5	0101	13	0001 0011
6	0110	14	0001 0100
7	0111	15	0001 0101

BCD 码是比较直观的。

例：(0100 1001 0111 1000.0001 0100 1001)_{BCD}

可以很方便地认出为：

4978. 149

即，只要熟悉了 BCD 的十位编码，立即可以很容易地实现十进制与 BCD 码之间的转换。

但是 BCD 码与二进制之间的转换是不直接的，要先经过十进制。即：BCD 码先转换为十进制码然后再转换为二进制，反之亦然。

(二) 字母与字符的编码

如上所述，字母和各种字符也必须按特定的规则用二进制编码才能在机中表示。编码也可以有各种方式——即规定。目前在微型机中最普遍的是采用 ASCII (American Standard Code for Information Interchange 美国标准信息交换码) 码，编码表见附录 1。

它是用七位二进制编码，故可表示 128 个字符，其中包括数码(0—9)，以及英文字母等可打印的字符。从表中可看到，数码 0—9，它是相应用 0110001—0111001 来表示的。因微型机通常字长为 8 位，所以通常 bit 7 用作奇偶校验位，但在机中表示时，常使其为零，故用一个字长(即一个字节)来表示一个 ASCII 字符。于是 0—9 的 ASCII 码为 30H—39H；大写字母 A—Z 的 ASCII 码为 41H—5AH。

三、二进制数的运算

一种数字系统可进行两种基本的算术运算：加法和减法。利用加法和减法，就可以进行乘法、除法以及其他数值运算。

(一) 二进制加法

二进制加法的规则为：

- ① $0+0=0$
- ② $0+1=1+0=1$
- ③ $1+1=0$ 进位 1
- ④ $1+1+1=1$ 进位 1

若有两数 1101 和 1011 相加，则加法过程如下：

进 位	1 1 1
被加数	1 1 0 1
加 数	+1 0 1 1
	1 1 0 0 0

可见，两个二进制数相加，每一位有三个数——即相加的两个数以及低位的进位，用二进制的加法规则得到本位的和以及向高位的进位。

微型机中，通常字长为 8 位。例：两个八位数相加

进 位	1 1 1 1 1 1
被加数	1 0 1 1 0 1 0 1
加位数	+0 0 0 0 1 1 1 1
和	1 1 0 0 0 1 0 0

(二) 二进制减法

二进制减法的运算规则为：

- ① $0-0=0$
- ② $1-1=0$
- ③ $1-0=1$
- ④ $0-1=1$ 有借位

例：11000100 - 001100101，列出式子为：

借 位	1 1 1 1 1 1
借位以后的被减数	1 0 1 1 1 0 1
被 减 数	1 1 0 0 0 1 0 0
减 数	-0 0 1 0 0 1 0 1
差	1 0 0 1 1 1 1 1

与加法类似，每一位有三个数参加运算：本位的被减数和减数，以及低位来的借位。为了便于计算，式中列出了低位向高位的借位，在运算时先用被减数和借位相运算，得到考虑了借位以后的被减数，然后再减去减数，最后可得到每一位的差，以及所产生的借位。下面再举一个例子，说明这样的运算过程。

例：11101110 - 10111010，式子为：

借	位	0 1 1 0 0 0 0
借位后的被减数		1 0 0 0 1 1 1
被 减 数		1 1 1 0 1 1 1 0
减 数		-1 0 1 1 1 0 1 0
		0 0 1 1 0 1 0 0

(三) 二进制乘法

二进制乘法的运算规则为:

- ① $0 \times 0 = 0$
- ② $0 \times 1 = 0$
- ③ $1 \times 0 = 0$
- ④ $1 \times 1 = 1$

是十分简单的, 只有当两个 1 相乘时, 积才为 1, 否则积为 0。

二进制的乘法也与十进制的类似:

被乘数	1 1 1 1
乘 数	$\times 1 1 0 1$
	1 1 1 1
	0 0 0 0
	1 1 1 1
	1 1 1 1
	1 1 0 0 0 1 1

可用乘数的每一位去乘被乘数, 乘得的中间结果的最低有效位与相应的乘数位对齐, 若乘数位为 1, 所得的中间结果即为被乘数; 若乘数位为 0, 则中间结果为 0。最后把这些中间结果一起加起来, 就可得到乘积。这种做法由于重复性差, 不便于在机器中实现。为了便于在机器中实现, 我们把运算方法改变一下。

1. 被乘数左移的方法:

乘 数	被 乘 数	部 分 积
① 乘数最低位为 1, 把被乘数加至部分积上, 然后把被乘数左移	1101 1111	0 0 0 0 + 1 1 1 1 1 1 1 1
② 乘数为 0, 不加被乘数, 被乘数左移	11110	
③ 乘数为 1, 加被乘数 (已左移后的)	1111000	1 1 1 1 1 1 1 1 0 0 1 0 0 1 0 1 1
④ 乘数为 1, 加被乘数得乘积		1 1 1 1 0 0 0 1 1 0 0 0 0 1 1

从上例中可看到两个 n 位数相乘, 乘积为 $2n$ 位。在运算过程中, 这 $2n$ 位都有可能要有相加的操作, 故需 $2n$ 个加法器。

2. 部分积右移的乘法

上例中是以被乘数左移的方法来实现的, 则两个 n 位数相乘, 乘积为 $2n$ 位, 在运算