

# 汇编语言基础

张国良 编译

国防工业出版社

# 汇编语言基础

张国良 编译

国防工业出版社

## 内 容 简 介

本书是一本介绍 IBM360、370 系统 (OS/VS 及 DOS/VS) 汇编语言的教学参考书。全书共分 16 章。本书在讲解汇编语言的基本概念的前提下,通过一些实用程序的例子,系统地介绍了使用汇编语言编写程序的各种技术。本书取材精炼,深入浅出,详尽地阐明了汇编语言中最基本的内容。每章后附有习题。

本书不仅适用于从事小型事务处理方面工作的程序员学习,而且可作为大专院校有关专业的师生参考书。对想了解 IBM 370 系统的读者也有一定的参考价值。

## 汇 编 语 言 基 础

张 国 良 编 译

\*

国防工业出版社出版

新华书店北京发行所发行 各地新华书店经售

北京昌平兴华印刷厂印装

\*

757×1092  $\frac{1}{16}$  印张 30  $\frac{1}{2}$  718千字

1987年11月第一版 1987年11月第一次印刷 印数: 0,001—4,920册

ISBN 7-118-00187-2/TP18 定价: 6.20元

## 前 言

本书向读者介绍如何使用汇编语言来编写计算机程序的基本原理。众所周知,和其他几种高级语言相比,汇编语言和计算机的结构及机器语言的关系甚为密切。正因为这一点,汇编语言才显示出和其他高级语言所不同的特点。尽管各种不同的计算机系统的汇编语言也有所不同,但由于基本原理大同小异,所以在介绍汇编语言时总是以某一具体系统为背景来讲解。本书就是以 IBM360、370 系统的 OS/VS 及 DOS/VS 操作系统为背景介绍汇编语言的。IBM370 系统是世界市场上用户最多的计算机系统之一,因此,以它为背景是不失其一般性和通用性的。读者只要掌握了本书的内容,对其他计算机系统的汇编语言也就不难掌握了。经验证明,要想成为一个优秀的程序设计者,只掌握一些高级语言(如 FORTRAN、COBOL 等)是不够的,还必须懂得汇编语言,这对更好理解其他各种高级语言是很有帮助的。

本书共有十六章,可分为六部分。

第一部分包括第一、二、三章,介绍汇编语言的基本概念。

第二部分包括第四、五、六章,主要介绍十进制指令系统,讨论了十进制整数、小数的算术运算及简单的编辑操作。

第三部分包括第七、八、九章,主要介绍标准指令系统,讨论了二进制制、十六进制以及二进制整数、非整数的运算操作。

第四部分包括第十、十一章,介绍了复杂的编辑功能以及程序的转移和循环控制。

第五部分包括第十二、十三、十四章,介绍操作系统的概念,讨论了子程序及其连接的标准约定,还讨论了虚存储器概念。最后介绍了汇编语言程序的调试方法。

第六部分包括第十五、十六章。第十五章简单地介绍科学计算中常用的浮点操作指令。第十六章讨论了大型系统中磁带和磁盘文件的使用。

本书前十四章译自 RINA·YARMISH 等编著的《Assembly Language Fundamentals 360/370 OS/VS DOS/VS》一书。第十五、十六章是根据下列几本书编译的:Shan S·Kuo 的《Assembler Language for FORTRAN, COBOL and PL/1 Programmer》;Ned Chapin 的《360/370 Programming in Assembly Language》以及 Naney Stern 等的《370/360 Assembler Language programming》。

在本书的编译中得到许多同志的热情帮助和鼓励,在此谨表示衷心的感谢!由于译者水平不高,难免有错误和不妥之处,望读者给予指正。

# 目 录

|                                   |    |
|-----------------------------------|----|
| 第一章 计算机简介 .....                   | 1  |
| 1.1 什么是计算机? .....                 | 1  |
| 1.2 计算机系统 .....                   | 1  |
| 1.2.1 存储器 .....                   | 2  |
| 1.2.2 算术及逻辑部件 .....               | 5  |
| 1.2.3 控制器 .....                   | 6  |
| 1.2.4 输入设备 .....                  | 6  |
| 1.2.5 输出设备 .....                  | 8  |
| 1.2.6 辅助存储器 .....                 | 8  |
| 1.3 数据结构: 字段、记录、文件 .....          | 8  |
| 1.4 和计算机通信 .....                  | 9  |
| 1.4.1 计算机“语言” .....               | 9  |
| 1.4.2 机器语言 .....                  | 10 |
| 1.4.3 汇编语言 .....                  | 20 |
| 1.4.4 为什么需要汇编语言? .....            | 21 |
| 习题 .....                          | 22 |
| 第二章 第一个程序 .....                   | 25 |
| 2.1 第一个问题的说明 .....                | 25 |
| 2.2 程序指令 .....                    | 28 |
| 2.2.1 汇编语言指令: 汇编指令、机器指令及宏指令 ..... | 28 |
| 2.2.2 指令的共同特点 .....               | 29 |
| 2.2.3 编码格式纸 .....                 | 31 |
| 2.3 计算机的工作步骤: 程序的翻译和执行 .....      | 34 |
| 2.4 S先生的程序指令的解释 .....             | 35 |
| 2.5 作业控制语言 (JCL) 介绍 .....         | 54 |
| 2.6 程序的全貌 .....                   | 55 |
| 2.6.1 编码 .....                    | 55 |
| 2.6.2 穿孔 .....                    | 58 |
| 2.6.3 汇编、连接编辑及程序的输出 .....         | 59 |
| 2.6.4 概括, 这个程序做了些什么? .....        | 64 |
| 习题 .....                          | 65 |
| 第三章 存储区及常数的定义 .....               | 68 |
| 3.1 为什么以及怎样把数据放入存储器中? .....       | 68 |
| 3.2 DC 指令: 常数的定义 .....            | 68 |
| 3.2.1 EBCDIC 常数 .....             | 70 |
| 3.2.2 装配常数 .....                  | 73 |
| 3.2.3 区标常数 .....                  | 77 |
| 3.2.4 十六进制常数 .....                | 79 |
| 3.2.5 二进制常数 .....                 | 80 |
| 3.3 直接常数 .....                    | 83 |
| 3.4 DS 指令: 存储区的定义 .....           | 84 |

|                                   |            |
|-----------------------------------|------------|
| 3.5 第二个程序 .....                   | 87         |
| 3.6 求解问题的步骤 .....                 | 87         |
| 3.6.1 精确地理解该问题的含义 .....           | 88         |
| 3.6.2 解的编码 .....                  | 93         |
| 习题 .....                          | 99         |
| <b>第四章 十进制运算——装配整数值的操作</b> .....  | <b>102</b> |
| 4.1 工资表的问题 .....                  | 102        |
| 4.1.1 问题的说明 .....                 | 102        |
| 4.1.2 区标及装配形式 .....               | 102        |
| 4.1.3 工资表程序的逻辑 .....              | 104        |
| 4.1.4 工资表程序——编码 .....             | 107        |
| 4.1.5 工资表程序——输出 .....             | 111        |
| 4.1.6 指令完成了什么? .....              | 113        |
| 4.2 程序中指令的解释 .....                | 116        |
| 4.2.1 AP (十进制加法) 指令 .....         | 116        |
| 4.2.2 SP (十进制减法) 指令 .....         | 117        |
| 4.2.3 ZAP (十进制清加) 指令 .....        | 119        |
| 4.2.4 MP (十进制乘法) 指令 .....         | 121        |
| 4.2.5 DP (十进制除法) 指令 .....         | 122        |
| 4.2.6 PACK (装配) 指令 .....          | 125        |
| 4.2.7 UNPK (拆卸) 指令 .....          | 127        |
| 习题 .....                          | 129        |
| 程序练习 .....                        | 132        |
| <b>第五章 简单的比较和编辑</b> .....         | <b>133</b> |
| 5.1 程序输出的解释 .....                 | 133        |
| 5.2 改进程序输出的样式 .....               | 135        |
| 5.2.1 MVZ 指令 .....                | 136        |
| 5.3 比较和控制的转换 .....                | 138        |
| 5.3.1 装配数的比较; CP 指令 .....         | 139        |
| 5.3.2 带有扩展助记符的转移 .....            | 142        |
| 5.3.3 工资表程序——加班的测试 .....          | 142        |
| 5.3.4 主存中的逻辑比较 .....              | 147        |
| 习题 .....                          | 150        |
| 程序练习 .....                        | 152        |
| <b>第六章 非整数值的十进制运算</b> .....       | <b>154</b> |
| 6.1 工资表问题——第3版 .....              | 154        |
| 6.2 非整数十进制运算技术 .....              | 155        |
| 6.2.1 非整数字段的加法和减法 .....           | 155        |
| 6.2.2 非整数字段的乘法 .....              | 161        |
| 6.2.3 非整数字段的除法 .....              | 165        |
| 6.3 非整数值的工资表问题 .....              | 166        |
| 6.4 十进制字段移位指令 .....               | 172        |
| 6.5 IBM370 系统中十进制字段的移位和舍入指令 ..... | 176        |
| 习题 .....                          | 180        |
| 程序练习 .....                        | 181        |
| <b>第七章 二进制数的操作</b> .....          | <b>183</b> |
| 7.1 标准指令系统 .....                  | 183        |

|                                               |     |
|-----------------------------------------------|-----|
| 7.2 二进制和十六进制的位置表示法 .....                      | 183 |
| 7.2.1 二进制制 .....                              | 184 |
| 7.2.2 十六进制制 .....                             | 195 |
| 7.2.3 二进制、十六进制及十进制的整数转换 .....                 | 199 |
| 7.3 程序中的二进制数 .....                            | 204 |
| 7.3.1 转换的必要性 .....                            | 204 |
| 7.3.2 二进制数在哪儿? .....                          | 205 |
| 7.3.3 定点常数, 存储区的定义及直接常数 .....                 | 205 |
| 7.3.4 使用二进制值的程序例子 .....                       | 208 |
| 7.3.5 二进制转换指令: CVB, CVD .....                 | 209 |
| 习题 .....                                      | 211 |
| 第八章 二进制整数的算术运算 .....                          | 214 |
| 8.1 库存报表问题 .....                              | 214 |
| 8.1.1 问题的说明 .....                             | 214 |
| 8.1.2 库存报表的程序逻辑 .....                         | 215 |
| 8.1.3 库存报表程序的编码 .....                         | 219 |
| 8.1.4 库存报表程序的输出 .....                         | 219 |
| 8.1.5 程序是怎样产生该报表的 .....                       | 220 |
| 8.2 程序指令的解释: 定点运算指令 .....                     | 222 |
| 8.2.1 定点加法指令: AR, A, AH .....                 | 222 |
| 8.2.2 定点减法指令: SR, S, SH .....                 | 226 |
| 8.2.3 定点乘法指令: MR, M, MH .....                 | 229 |
| 8.2.4 定点除法指令: DR, D .....                     | 233 |
| 8.2.5 定点除法的准备操作 .....                         | 236 |
| 8.2.6 某些程序例子 .....                            | 237 |
| 8.2.7 数据传送: 寄存器到寄存器, 主存储器到寄存器, 寄存器到主存储器 ..... | 238 |
| 习题 .....                                      | 243 |
| 程序练习 .....                                    | 245 |
| 第九章 二进制比较和非整数值的运算 .....                       | 247 |
| 9.1 库存报表问题——第2版 .....                         | 247 |
| 9.2 非整数定点运算的技术 .....                          | 248 |
| 9.2.1 非整数定点字段的加减法 .....                       | 248 |
| 9.2.2 非整数定点字段的乘、除法 .....                      | 250 |
| 9.2.3 长度因子在二进制运算中的应用 .....                    | 253 |
| 9.3 定点数的比较 .....                              | 254 |
| 9.4 求解带有非整数值的库存报表问题 .....                     | 257 |
| 9.5 代数移位指令 .....                              | 263 |
| 习题 .....                                      | 268 |
| 第十章 打印输出形式的改进: 编辑和格式控制 .....                  | 270 |
| 10.1 编辑(ED)指令 .....                           | 270 |
| 10.1.1 取消数值高位上无意义的零 .....                     | 271 |
| 10.1.2 标点的插入 .....                            | 273 |
| 10.1.3 有效开始符: 提前建立有效指示符 .....                 | 274 |
| 10.1.4 带符号的字段: 负号的后缀 .....                    | 275 |
| 10.1.5 用一条ED指令编辑几个字段 .....                    | 277 |
| 10.1.6 ED指令综述 .....                           | 279 |
| 10.2 “编辑并说明”(EDMK)指令 .....                    | 281 |

|                                          |            |
|------------------------------------------|------------|
| 10.2.1 固定的与浮动的符号                         | 281        |
| 10.2.2 EDMK 指令的应用                        | 282        |
| 10.2.3 EDMK 指令综述                         | 283        |
| 10.3 格式控制和接卡箱选择                          | 285        |
| 10.3.1 托架控制                              | 285        |
| 10.3.2 CNTRL 宏指令                         | 286        |
| 10.3.3 PRTOV 宏指令                         | 287        |
| 10.3.4 Spool 系统上的托架控制                    | 290        |
| 10.4 COMRG 及 TIME 宏指令                    | 291        |
| 10.5 带有编辑和格式控制的报表程序                      | 292        |
| 习题                                       | 295        |
| 程序练习                                     | 297        |
| <b>第十一章 转移和循环</b>                        | <b>298</b> |
| 11.1 社会保险问题                              | 298        |
| 11.1.1 问题的说明                             | 298        |
| 11.1.2 条件码和指令地址寄存器                       | 299        |
| 11.1.3 BC 和 BCR 指令                       | 300        |
| 11.1.4 社会保险问题——编码                        | 303        |
| 11.1.5 扩展助记符指令                           | 305        |
| 11.2 表处理问题                               | 306        |
| 11.2.1 什么是表?                             | 306        |
| 11.2.2 问题的说明                             | 306        |
| 11.2.3 循环结构                              | 307        |
| 11.2.4 表访问中基址寄存器和变址寄存器的应用                | 309        |
| 11.2.5 用变址转移指令处理表问题                      | 313        |
| 11.2.6 用 BCT 和 BCTR 指令控制循环               | 317        |
| 习题                                       | 320        |
| <b>第十二章 子程序及其连接</b>                      | <b>322</b> |
| 12.1 子程序的必要性                             | 322        |
| 12.1.1 程序设计上的麻烦是每页都要打印标题栏                | 322        |
| 12.1.2 什么是子程序?                           | 324        |
| 12.2 内部子程序                               | 325        |
| 12.2.1 内部子程序的调用和退出: BAL、BALR 及 BR 指令     | 325        |
| 12.2.2 标题程序的编码, 使用子程序的例子                 | 328        |
| 12.3 外部子程序                               | 330        |
| 12.3.1 连接约定                              | 330        |
| 12.3.2 外部子程序的调用和退出                       | 331        |
| 12.3.3 调用和被调用子程序间的数据传送                   | 334        |
| 12.3.4 寄存器内容的保存和恢复                       | 337        |
| 12.3.5 摘要: 调用和被调用程序的职责                   | 340        |
| 12.3.6 SAVE, RETURN 及 CALL 宏指令           | 343        |
| 12.3.7 程序举例                              | 345        |
| 12.4 连接编辑程序和子程序连接                        | 346        |
| 习题                                       | 347        |
| 程序练习                                     | 349        |
| <b>第十三章 操作系统的一些性能: 虚存储器, 程序状态字, 中断系统</b> | <b>350</b> |
| 13.1 什么是操作系统                             | 350        |



|                                 |     |
|---------------------------------|-----|
| 13.2 虚存储器系统 .....               | 350 |
| 13.2.1 存储器分配 .....              | 351 |
| 13.2.2 动态再定位技术: 段和页 .....       | 353 |
| 13.2.3 虚存储器 (VS): 它的方法和结构 ..... | 359 |
| 13.2.4 虚存储器系统中程序的执行 .....       | 360 |
| 13.2.5 单虚存储器和多虚存储器 .....        | 362 |
| 13.2.6 虚存储器中的编码 .....           | 363 |
| 13.2.7 虚存储器的优点 .....            | 363 |
| 13.3 系统设计的某些关键性能 .....          | 363 |
| 13.3.1 BC 方式的程序状态字 .....        | 364 |
| 13.3.2 EC 方式的程序状态字 .....        | 366 |
| 13.3.3 中断系统 .....               | 368 |
| 习题 .....                        | 374 |
| 第十四章 程序的调试 .....                | 377 |
| 14.1 预防为主 .....                 | 377 |
| 14.2 人工检查 .....                 | 377 |
| 14.3 诊断 .....                   | 378 |
| 14.4 程序的测试 .....                | 385 |
| 14.4.1 测试数据 .....               | 385 |
| 14.4.2 程序中断: 程序校验信息 .....       | 386 |
| 14.4.3 程序转储 .....               | 388 |
| 14.4.4 转储的分析: 排错的例子 .....       | 397 |
| 习题 .....                        | 400 |
| 第十五章 浮点操作 .....                 | 401 |
| 15.1 浮点数的格式 .....               | 401 |
| 15.2 浮点数的运算 .....               | 403 |
| 15.2.1 浮点寄存器 .....              | 403 |
| 15.2.2 浮点常数的定义 .....            | 403 |
| 15.2.3 浮点数的算术运算 .....           | 404 |
| 15.3 浮点操作指令 .....               | 405 |
| 15.3.1 指令格式 .....               | 405 |
| 15.3.2 浮点加减法指令 .....            | 405 |
| 15.3.3 浮点乘法指令 .....             | 407 |
| 15.3.4 浮点除法指令 .....             | 408 |
| 15.3.5 浮点数比较和存数指令 .....         | 409 |
| 15.3.6 浮点取数指令 .....             | 409 |
| 15.4 浮点操作举例 .....               | 410 |
| 习题 .....                        | 413 |
| 第十六章 磁带和磁盘的使用 .....             | 415 |
| 16.1 五种类型数据结构 .....             | 415 |
| 16.2 输入输出操作 .....               | 416 |
| 16.2.1 输入输出宏指令及其格式 .....        | 417 |
| 16.2.2 磁带和磁盘的 DCB 语句 .....      | 428 |
| 16.2.3 磁带和磁盘的 DTF 语句 .....      | 430 |
| 16.3 索引顺序文件中数据和索引格式 .....       | 433 |
| 16.4 索引顺序文件中的 OS 宏指令 .....      | 439 |
| 16.5 索引顺序文件中的 DOS 宏指令 .....     | 440 |

|                     |     |
|---------------------|-----|
| 16.5.1 装入宏指令        | 441 |
| 16.5.2 记录的插入和删除     | 442 |
| 16.5.3 随机处理         | 443 |
| 16.5.4 顺序处理         | 444 |
| 16.6 索引顺序文件的DCB语句   | 445 |
| 16.7 索引顺序文件的DTFIS语句 | 447 |
| 16.8 带和盘的标记及作业控制卡   | 453 |
| 习题                  | 454 |
| 附录A 370 系统摘要        | 455 |
| 附录B 输入输出宏指令         | 469 |
| 附录C 作业控制语句          | 472 |

# 第一章 计算机简介

本章简单地介绍一下计算机的性能、结构以及同它通信的方法。同时，还将讨论有助于理解本书所需要的一些基本术语。

本章扼要地复习需要几学期才能讲授完的，介绍数据处理课程的某些内容。虽然本章所介绍的内容，对理解本书来说是足够的，但对想更深入地研究某些具体范畴的读者来说，却需要参看已有的许多更好的入门教程。这里所介绍的都是一些实质性的必需的内容，可作为读者阅读过程中的参考。

## 1.1 什么是计算机？

在生活中，无处不谈到和用到“计算机”。在银行，出纳员使用计算机存款和取款；在实验室，用计算机处理实验结果；军事上使用计算机灵活地把导弹引向目标；计算机能控制很多工厂的操作，甚至用于交通指挥灯的管制等等。这些只是现代计算机的许多应用中的一些个别事例。

究竟什么是计算机呢？概括地说，它是一种特殊类型的“数据处理器”。它是把进来的数据（输入），在事先给它的一组指令（被存储的程序）的控制下转换为输出信息（输出）的一种电子机器。

如果机器完全是电子的，它就没有传动部件。这就意味着，数据的任何传送和处理，从它进入计算机（输入），直到它离开（输出），都是在计算机内以极大速度运动的电信号（实际上就是以极大的速度通过导线传输的电子）来进行的。没有传动的部件，也说明这种机器所需的维修工作量很小。

但是，使计算机具有如此功能的真正原因，是它能按照一组指令（程序）的控制来进行工作。我们可以用改变程序的办法，使计算机也跟着改变它正在进行的工作。因为能够编写的程序的数目可以无限多，所以，计算机也就能完成无限多种作业。计算机本身并不能指挥自己，“思维”必须由人来进行。计算机没有“智力”，即没有思维能力；它之所以重要，原因在于它是一种精确、快速、操作可控的高可靠通用工具。

如果你读了这本书，你就有可能学会如何通过编写程序来指挥计算机的工作。

## 1.2 计算机系统

一个典型的计算机系统包括有六个基本部分。有时可以将前三个（图 1.1）安装在同一个物理部件中，它们是：

- (1) 存储器（有时称为主存储器，内存或基本存储器）；
- (2) 算术和逻辑部件，通常简称为运算器（ALU）；
- (3) 控制器。

这三个部分通常总称为中央处理机（简称为 CPU）。〔按照美国国家标准学会（ANSI）的建议，CPU 只包括 ALU 及控制器，而存储器则作为一个独立的部件〕。数据的实际

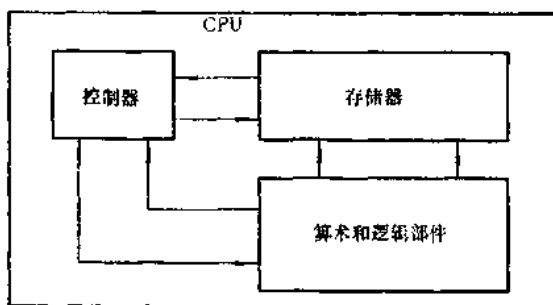


图 1.1 中央处理机 (CPU)。图中箭头表示信息通道

处理是在 CPU 中进行的。在本书中，术语“CPU”和“计算机”是可互换的。

另外的三个部分是：

- 输入设备
- 输出设备
- 辅助存储器

经常用 I/O 表示输入/输出

上述六个部分构成了计算机系统。现在简要地研究一下每一部分。

### 1.2.1 存储器

正像人们在按照某些信息行动之前，必须先把这些信息存放在大脑中一样，计算机在操作之前，也必须事先将数据，以及用来告诉计算机做什么的指令存放在存储器中。这些指令构成一个存储程序。物理上，一些计算机的存储器是由磁心组成的。这些磁心以阵列方式穿在导线上。因此，有时把计算机的存储器和磁心相提并论。通过这些导线的电流可以有选择地按两个方向（顺时针或逆时针）之一，磁化单个磁心。

实际上，计算机工艺目前正趋向用更快的固态线路及单片线路构成存储器。激光技术也开始用到许多现代计算机的设计和结构中。但是，由于磁心存储器仍作为主存储器来使用，所以仍然保留着这个术语。

#### 二进制数

电灯开关可以处于打开（连通）或关闭（断开）状态；电子线路可以导通或断路；插头也可以连通或断开。在这些情况中，都存在两种可能的状态。技术上可以把这两种状态分别看作为“是”或“否”，或者更一般讲，“1”或“0”。当数据被存储在计算机存储器中时，它是用一系列磁心的状态来表示的。用磁心的两个磁化方向，来表示数字 0 和 1。如图 1.2 所示。因此，一个磁心可看作是一个二进制位（二进制数字）。正像莫尔斯（Morse）电码中每个字母符号都用唯一的点划线序列组成一样，计算机中的符号由唯一的二进制序列组成。不过在莫尔斯编码中，各符号的长度有

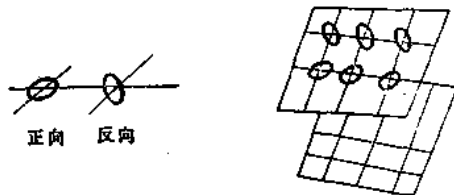


图 1.2

所不同；而计算机的存储器是被分成相等长度的段，称之为字节。每个字节可放一个符号〔诸如字母、数字、专用符号（如“&”）等等〕。在 IBM 计算机上，每个字节长度为 8 位。

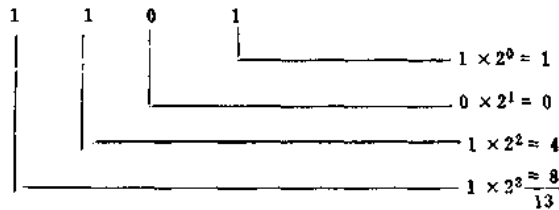
计算机中采用二进制来表示数值，即以 2 的幂次值作为权，由数字 0 和 1 组成的序列来表示一个数值。

二进制中各位的权为：

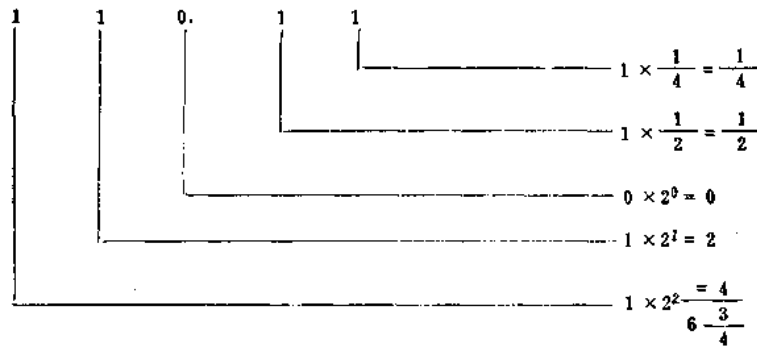
$$\cdots 2^3 = 8 \quad 2^2 = 4 \quad 2^1 = 2 \quad 2^0 = 1 \quad 2^{-1} = 1/2 \quad 2^{-2} = 1/4 \quad 2^{-3} = 1/8 \cdots$$

二进制数列中的数字“1”是对该列值本身有“价值”的量。数字“0”没有对该列的值作出贡献，只是占据一个位置。

例如，二进制数 1101 等于十进制 13，对应关系如下所示：



类似地，二进制数 110.11 等于十进制数  $6 \frac{3}{4}$ ，对应关系如下所示：



### 十六进制表示法

用二进制数进行计算的一个主要缺点是，数目太长和阅读不便。为此，经常采用十六进制，作为表示 0 和 1 数字串的一种简洁的表示法。

十六进制表示法，以一个等值的十六进制数字代替 4 个二进制数字，其对应关系如下：

| 十六进制 | 二进制  | 十六进制 | 二进制  |
|------|------|------|------|
| 0    | 0000 | 8    | 1000 |
| 1    | 0001 | 9    | 1001 |
| 2    | 0010 | A    | 1010 |
| 3    | 0011 | B    | 1011 |
| 4    | 0100 | C    | 1100 |
| 5    | 0101 | D    | 1101 |
| 6    | 0110 | E    | 1110 |
| 7    | 0111 | F    | 1111 |

因此，

二进制数 1101 0101 1010 1111 可表示为

十六进制 B 5 A F

在第七章中还要详细讨论二进制和十六进制的问题。

注意, 用 4 位所能表示的最大二进制数是 1111 或 15 (即  $2^4-1$ ); 8 位所能表示的最大二进制数是 11111111 或 255 (即  $2^8-1$ )。一般讲,  $n$  位所能表示的最大二进制数是  $2^n-1$ 。因此, 能够用  $n$  位表示的二进制数的个数等于  $n$  位的最大二进制数加 1 (包括 0), 即  $(2^n-1)+1$  或  $2^n$  个。

### EBCDIC 编码

用 8 位 ( $2^8=256$ ) 能够表示 256 个 0 和 1 的不同组合。计算机厂家已把许多符号同这些组合——对应起来。本书中所使用的是一种特殊组合, 即 EBCDIC 码 (扩充的二进制编码的十进制交换代码)。通常, 将 8 位的字节分为两部分, 各位的次序是从左到右, 从 0 开始, 以 7 结束。现图示如下:

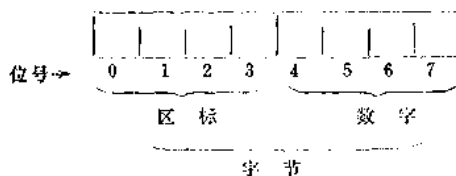


图1.3给出了数字和英文字母的EBCDIC 码

| 符 号 | EBCDIC | 编 码  |
|-----|--------|------|
| A   | 1100   | 0001 |
| B   | 1100   | 0010 |
| C   | 1100   | 0011 |
| D   | 1100   | 0100 |
| E   | 1100   | 0101 |
| F   | 1100   | 0110 |
| G   | 1100   | 0111 |
| H   | 1100   | 1000 |
| I   | 1100   | 1001 |
| J   | 1101   | 0001 |
| K   | 1101   | 0010 |
| L   | 1101   | 0011 |
| M   | 1101   | 0100 |
| N   | 1101   | 0101 |
| O   | 1101   | 0110 |
| P   | 1101   | 0111 |
| Q   | 1101   | 1000 |
| R   | 1101   | 1001 |
| S   | 1110   | 0010 |
| T   | 1110   | 0011 |
| U   | 1110   | 0100 |
| V   | 1110   | 0101 |
| W   | 1110   | 0110 |
| X   | 1110   | 0111 |
| Y   | 1110   | 1000 |
| Z   | 1110   | 1001 |

字母 A~I 的区标总是 1100, 而数字部分由 4 位二进制数 1~9 组成

字母 J~R 的区标总是 1101, 而数字部分由 4 位二进制数 1~9 组成

字母 S~Z 的区标总是 1110, 而数字部分由 4 位二进制数 2~9 组成

→

|                                                        |   |      |      |
|--------------------------------------------------------|---|------|------|
| 十进制数字 0~9 的区<br>标总是1111, 而数字部分<br>由 4 位二进制数 0~9 组<br>成 | 0 | 1111 | 0000 |
|                                                        | 1 | 1111 | 0001 |
|                                                        | 2 | 1111 | 0010 |
|                                                        | 3 | 1111 | 0011 |
|                                                        | 4 | 1111 | 0100 |
|                                                        | 5 | 1111 | 0101 |
|                                                        | 6 | 1111 | 0110 |
|                                                        | 7 | 1111 | 0111 |
|                                                        | 8 | 1111 | 1000 |
|                                                        | 9 | 1111 | 1001 |

图1.3 字母和数字的 EBCDIC表示法

例如, 单词“RAIN”的 EBCDIC 码存储时, 需要 4 个连续的字节, 即:

|      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|
| 1101 | 1001 | 1100 | 0001 | 1100 | 1001 | 1101 | 0101 |
| R    |      | A    |      | I    |      | N    |      |

一个典型的计算机存储器可以存储成千上万个 EBCDIC 字符。

#### 存储器的寻址

邮局中的信箱号是唯一且连续的编号。类似地, 计算机厂家把主存储器中的各字节位置, 同程序员用来识别某一特殊字节或存储单元的编号一一对应起来, 这个编号称为地址。

因此, 我们可以在概念上把计算机存储器看作由许多编了号的小盒子组成的一个大箱子, 而每个小盒子的大小等于一个字节。

|     |     |     |     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|     | 000 | 001 | 002 | 003 | 004 | 005 | 006 | 007 | 008 | 009 |
|     |     |     | R   | A   | I   | N   |     |     |     |     |
|     | 010 | 011 | 012 | 013 |     |     |     |     |     |     |
| 010 |     |     |     |     | ... |     |     |     |     |     |
| ... | 020 | 021 | 022 | 023 |     |     |     |     |     |     |
| ... |     |     |     |     | ... |     |     |     |     |     |

上图中, 单词“RAIN”在存储器中, 从 002 地址开始共占用了 4 个字节。

IBM 公司习惯于以  $k$  为单位来表示存储器容量的大小, 每个  $k$  等于 1024 个字节 ( $1024=2^{10}$ )。一个典型的计算机存储器容量有 128  $k$  个字节。如果需要, 可以从卖主那里购买附加存储器。磁心存储器通常以 4  $k$  为单位出售。

#### 1.2.2 算术及逻辑部件

算术及逻辑部件是由电子线路组成的。这些电子线路的功能是执行诸如加、减、乘、除等操作。计算机把除法看成是减法的扩展。ALU 还可以比较两个值, 以确定一个值是否大于、等于或小于另一个值, 并且它还能判定逻辑状态是“真”还是“假”。

为了便于操作, 还可以利用几个 32 位的寄存器作为存储器。这些寄存器在操作期

间用来暂时存储少量的信息，如计算中产生的一些中间结果。

### 1.2.3 控制器

控制器扮演着“总经理”的角色。它控制着计算机的各种操作。例如，控制计算机的输入和输出，以及计算机六个部分之间的通信（输入、输出、控制器、ALU、存储器及辅助存储器）；控制器解释程序指令、启动 I/O 设备、启动计算机程序的执行，以及保证每件事都能以正确的时间和正确顺序来进行。在这方面，它很像是在繁忙的公路交叉路口指挥来往车辆的交警的工作。

### 1.2.4 输入设备

#### 输入和输出

把信息送进（输入）计算机以及从计算机中取出（输出）信息的方法很多。它们的区别仅在于所使用的介质、速度、价格、方便性，以及它们的适用性。

在本书中，我们使用穿孔卡片及读卡机分别作为输入介质及输入设备。使用打印机作为输出设备，它的介质是连续的纸。之所以选择这些设备，主要是由于它们是最常用的设备，并且特别适合于初学者使用。

#### 穿孔卡片

图 1.4 中所示的穿孔卡，是由 H. Hollerith 在 1880 年末发明的。此后，它被命名为 Hollerith 卡片。而近年来多称为 IBM 卡片，简称为穿孔卡片。

如果从横向上看，该卡片上有 12 个横行，从顶部开始，前三行分别称为 12-行、11-行（或 X-行）及 0-行，这三行总称为区标行。在这些行上的任何穿孔都称为区标孔。接下去，在 0-行之下是 1-行、2-行等等，一直到 9-行。0-行到 9-行称为数字行。并且这几行中所有的孔都称为数字孔。注意，0-行既属于区标行也属于数字行。

现在让我们研究一下卡片的纵向划分。图上卡片的最下边印刷行标有“列号”（这一行和 0-行下面印刷的数字相同），从 1 到 80。每一个数字标一列，全卡片共有 80 列，故俗称为 80 列卡片。在想看某一列时，可使用这些列号。

还可以从这个图上看到，每一个符号都有一列唯一的孔数来表示它。但是，每一个符号只能占用一列，正像打字机上的符号一样，一个典型的符号只能占据一个打印位置。

对于数字 0~9 的表示，在各数字行的穿孔处印出这个数字。例如，卡片的第 8 列中印有 4 的位置，就意味着表示数字 4。只需要一个孔就能表示 0~9 中的某一数字。

在一列中要用两个孔表示一个字母符号，即一个区标孔和一个数字孔。字母 A~I 分别对应 12-行和数字行 1~9 的穿孔；字母 J~R 分别对应 11-行和数字行 1~9 的穿孔；字母 S~Z 分别对应 0-行和数字行 2~9 的穿孔。可以看出，这种编码和计算机存储器中的 EBCDIC 编码是不同的。

在穿孔卡片上印刷的标识行和列的数值或其它印刷信息，以及卡片的颜色等都是和卡片的存储信息不相干的，仅用孔的位置或卡片上的穿孔来确定表示什么信息。

#### 读卡机

只有用读卡机，才能把穿孔卡片上的内容传送到存储器中。典型的读卡机的“阅读”



