

8086/8088

汇编语言 程序设计

齐志儒 高福祥 编

科学出版社

3
/1

8086/8088 汇编语言 程序设计

齐志儒 高福祥 主编

科学出版社

1994

(京) 新登字 092 号

内 容 简 介

本书以 IBM PC 机为背景机, 系统介绍汇编语言程序设计的基本理论和方法。

全书共分 17 章。第一至第七章主要讲述计算机的基础知识、8086/8088 的指令系统、汇编语言、程序的基本控制结构及程序设计理论、方法和技巧。第八至第十章介绍汇编语言在数值运算、代码转换及列表处理方面的应用, 讨论了多种算法, 并给出了程序实例。第十一、十二章剖析 IBM PC 及其兼容机的输入输出接口, 详细介绍输入输出及中断程序设计的方法, 并以实例说明显示器、键盘、打印机、定时、通信的程序设计及应注意的问题。第十三章介绍磁盘结构和编程方法及文件操作。第十四章介绍汉字处理技术。第十五章介绍 IBM PC 的宏汇编技术。第十六章介绍 8087/80287/80387 的指令系统及程序设计方法, 并给出了应用实例。第十七章介绍 80286, 80386 的特点、指令系统及存储管理的有关知识。

书中每章都附有大量习题。

本书语言通俗易懂, 叙述由浅入深、循序渐进, 结构清晰严谨, 可作为大专院校计算机专业和自动化专业的教材, 也可作为工程技术人员自学、进修的教科书或参考书。

JS386/32

8086/8088 汇编语言程序设计

齐志儒 高福祥 主编

责任编辑 王春晖

科 学 出 版 社 出 版

北京东黄城根北街 16 号

邮政编码: 100717

北京市海淀久利印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

1994 年 3 月第 1 版 开本: 787×1092 1/16

1994 年 3 月第一次印刷 印张: 26 1/2

印数: 1—3200 字数: 609 000

ISBN 7-03-003811-8/TP·293

定价: 23.60 元

前 言

汇编语言是一种强有力的语言,它能透彻地反映、巧妙充分地运用计算机的硬件功能及特点,便于编程人员根据自己的需要灵活地编制高级语言能实现和无法实现的各种程序,随心所欲地控制计算机的运行。汇编语言是计算机提供的最快而又最有效的语言,也是能够利用计算机所有硬件特性的唯一语言。在许多对运行速度要求很高的场合,汇编语言是必不可少的,如操作系统、编译程序等,多数是用汇编语言编写的。近年来,微型计算机的迅速发展,特别是它在控制和计算机网络与通信等方面的广泛应用,使汇编语言程序设计技术不仅为系统程序设计,而且为广大微型计算机用户所普遍使用。

“汇编语言程序设计”是高等院校计算机、自动化等专业学生的必修课程,是计算机原理、操作系统等其它课程的必要先修课,对于训练学生掌握程序设计技术、熟练地上机操作和调试程序有重要作用。由于汇编语言的特点,本课程必须结合具体的计算机来教学。我们认为,在目前情况下汇编语言程序设计课程应着重结合微型计算机进行。目前 IBM PC 机及其兼容机、长城 0520 机是我国最流行、应用最广泛的 16 位微机。近几年来,以 80286 和 80386 为 CPU 的微型机也涌入我国微型机市场及各应用领域。分析、消化、扩充、改造、移植、汉化、开发各种系统软件,研制和调试应用软件是计算机应用领域的重要课题。我们根据多年的教学和科研经验,参阅了大量的资料,以 IBM PC 机为背景机,在东北大学计算机科学与工程系汇编语言教材的基础上编写了本书。

全书共分十七章,全面、清楚地讲述 8086/8088 的宏汇编语言,详细地讨论程序的基本控制结构、设计原理及编程技巧,以大量的实例说明数值运算、代码转换、列表处理、分类检索的算法,深入剖析 IBM PC 机的接口电路、中断系统,透彻地阐述键盘、显示器、打印机、音响、图形、通信等程序的设计方法,清晰地介绍磁盘结构、磁盘文件操作和汉字处理技术,并给出实用程序;完整地介绍 8087,80287,80387 的结构、指令系统,并以若干程序实例说明了指令系统的应用;最后概括地介绍 80286,80386 的扩充功能、指令系统,讲述存储管理及特权级的概念,为读者深入学习和掌握 80286,80386 奠定基础。书中所有程序都是经过上机验证的。各章后附有大量习题。

本书在编写上力求语言通俗易懂,叙述由浅入深、循序渐进,结构清晰严谨,可作为大专院校计算机专业和自动化专业的教材,也可作为工程技术人员自学、进修的教科书或参考书。

本书第一至第九章、十五章、十七章由齐志儒执笔,第十章由李秀芳执笔,第十一章由杨文福执笔,第十二、十六章由高福祥执笔,第十三、十四章由王剑执笔。全书由齐志儒、高福祥主编。

本书由全国高等计算机教育研究会副理事长、教材编委会主任、北京工业大学计算机系教授李大友审阅,经教材编委会讨论,推荐给科学出版社出版。编者对李大友教授和教材编委会成员所做的工作表示感谢。

限于编者水平,书中错误和不妥之处在所难免,敬请读者不吝批评指正。

目 录

| | |
|------------------------------------|----|
| 第一章 概述 | 1 |
| 1.1 计算机语言 | 1 |
| 1.1.1 机器语言 | 1 |
| 1.1.2 汇编语言 | 2 |
| 1.1.3 通用语言 | 3 |
| 1.2 计算机系统 | 4 |
| 1.2.1 硬件子系统 | 4 |
| 1.2.2 软件子系统 | 6 |
| 习题 | 7 |
| 第二章 8086/8088 微处理器的结构 | 9 |
| 2.1 8086/8088 CPU 的结构 | 9 |
| 2.2 8086/8088 寄存器和标志 | 10 |
| 2.2.1 通用寄存器 | 10 |
| 2.2.2 指针和变址寄存器 | 11 |
| 2.2.3 段寄存器 | 12 |
| 2.2.4 标志位 | 12 |
| 2.3 存储器 | 13 |
| 2.3.1 存储器的分段结构 | 13 |
| 2.3.2 实际地址的产生 | 14 |
| 2.4 寻址方式 | 15 |
| 2.4.1 操作数的种类 | 15 |
| 2.4.2 寻址方式 | 15 |
| 2.4.3 段更换和段跨越 | 18 |
| 2.4.4 有效地址的计算时间 | 18 |
| 2.5 8086/8088 指令系统 | 19 |
| 习题 | 19 |
| 第三章 汇编语言 | 21 |
| 3.1 汇编语言的语句格式 | 21 |
| 3.1.1 字符集 | 21 |
| 3.1.2 汇编语句格式 | 21 |
| 3.2 汇编语言中数据的表示方法 | 24 |
| 3.2.1 数据在机内的表示方法和范围 | 24 |
| 3.2.2 汇编语言中数据书写形式 | 26 |
| 3.3 运算符号 | 26 |
| 3.3.1 算术运算符 | 26 |
| 3.3.2 逻辑运算符 | 27 |
| 3.3.3 关系运算符 | 27 |

| | | |
|------------|---------------|-----------|
| 3.3.4 | 分析算符 | 27 |
| 3.3.5 | 组合算符 | 29 |
| 3.3.6 | 字节分离算符 | 29 |
| 3.3.7 | 记录专用算符 | 29 |
| 3.4 | 伪指令(汇编命令) | 30 |
| 3.4.1 | 符号定义伪指令 | 30 |
| 3.4.2 | 数据定义伪指令 | 31 |
| 3.4.3 | 记录定义和结构定义伪指令 | 34 |
| 3.4.4 | 段定义伪指令 | 36 |
| 3.4.5 | 过程(子程序)定义伪指令 | 39 |
| 3.4.6 | 程序段连接伪指令 | 40 |
| 3.4.7 | 其它伪指令 | 42 |
| | 习题 | 43 |
| 第四章 | 顺序结构程序 | 45 |
| 4.1 | 程序设计的步骤 | 45 |
| 4.2 | 流程图的应用 | 46 |
| 4.3 | 程序的基本控制结构 | 47 |
| 4.4 | 数据传送指令 | 48 |
| 4.5 | 算术运算指令 | 52 |
| 4.5.1 | 加法运算指令 | 52 |
| 4.5.2 | 减法运算指令 | 53 |
| 4.5.3 | 乘法运算指令 | 54 |
| 4.5.4 | 除法运算指令 | 55 |
| 4.6 | 逻辑操作指令 | 57 |
| 4.7 | 移位操作指令 | 58 |
| 4.8 | 状态标志位操作指令 | 59 |
| 4.9 | 简单的 I/O 功能调用 | 59 |
| 4.10 | 顺序结构程序举例 | 60 |
| | 习题 | 67 |
| 第五章 | 分支结构程序 | 70 |
| 5.1 | 分支结构程序的引出 | 70 |
| 5.2 | 转移指令 | 71 |
| 5.2.1 | 无条件转移指令 | 71 |
| 5.2.2 | 条件转移指令 | 73 |
| 5.3 | 标志寄存器 | 73 |
| 5.4 | 分支结构程序设计 | 77 |
| 5.5 | 多分支结构程序的设计 | 82 |
| | 习题 | 86 |
| 第六章 | 循环结构程序 | 89 |
| 6.1 | 数据串操作指令 | 89 |

| | | |
|------------|---------------------------|------------|
| 6.2 | 重复控制指令 | 93 |
| 6.3 | 循环结构程序的引出 | 94 |
| 6.4 | 循环结构程序的组成 | 96 |
| 6.5 | 循环程序的控制方法 | 97 |
| 6.6 | 单重循环程序举例 | 101 |
| 6.7 | 多重循环程序 | 106 |
| 6.8 | 循环程序的效率 | 109 |
| | 习题 | 111 |
| 第七章 | 子程序 | 115 |
| 7.1 | 调用和返回指令 | 115 |
| 7.2 | 子程序的引出 | 117 |
| 7.2.1 | 子程序的调用与返回 | 118 |
| 7.2.2 | 主程序与子程序间信息交换 | 118 |
| 7.2.3 | 保存现场和恢复现场 | 124 |
| 7.2.4 | 子程序说明文件的编制 | 124 |
| 7.3 | 子程序设计 | 125 |
| 7.4 | 子程序嵌套 | 130 |
| 7.5 | 递归子程序 | 133 |
| 7.6 | 可重入子程序 | 136 |
| 7.7 | 程序的连接 | 137 |
| | 习题 | 140 |
| 第八章 | 数值运算 | 144 |
| 8.1 | 多字节加减运算 | 144 |
| 8.2 | 多字节整数乘除运算 | 145 |
| 8.2.1 | 一般整数乘法运算 | 145 |
| 8.2.2 | 多字节整数乘法运算 | 146 |
| 8.2.3 | 一般整数除法运算 | 149 |
| 8.2.4 | 多字节整数除法运算 | 151 |
| 8.3 | BCD 码运算 | 156 |
| 8.3.1 | BCD 码运算规则 | 156 |
| 8.3.2 | BCD 码运算调整指令 | 157 |
| 8.3.3 | 非组合的 BCD 码运算 | 158 |
| 8.3.4 | 组合的 BCD 码加减运算 | 159 |
| 8.3.5 | BCD 码数据的符号表示方法及运算方法 | 160 |
| 8.4 | 浮点数据的表示方法 | 161 |
| 8.5 | 浮点数加减运算 | 162 |
| 8.5.1 | 浮点数加法 | 162 |
| 8.5.2 | 浮点数减法 | 166 |
| 8.6 | 浮点数乘除运算 | 166 |
| 8.6.1 | 浮点数乘法 | 166 |

| | |
|------------------------------------|------------|
| 8.6.2 浮点数除法 | 167 |
| 习题 | 167 |
| 第九章 代码转换 | 169 |
| 9.1 十进制数的 ASCII 码串转换为二进制定点数 | 169 |
| 9.2 二进制定点数转换为十进制数的 ASCII 码串 | 171 |
| 9.3 十进制数的 ASCII 码串转换为浮点数 | 174 |
| 9.3.1 普通十进制数的 ASCII 码串转换为浮点数 | 174 |
| 9.3.2 科学表示法的 ASCII 码串转换为浮点数 | 177 |
| 9.4 浮点数据转为十进制数的 ASCII 码串 | 178 |
| 9.4.1 浮点数转换为普通十进制数的 ASCII 码串 | 178 |
| 9.4.2 浮点数转换为科学表示法的 ASCII 码串 | 179 |
| 习题 | 179 |
| 第十章 列表处理 | 181 |
| 10.1 概述 | 181 |
| 10.1.1 列表的存储结构 | 181 |
| 10.1.2 自定义数据结构 | 182 |
| 10.1.3 列表的种类及其运算 | 186 |
| 10.2 列表的插入与删除 | 187 |
| 10.2.1 插入 | 187 |
| 10.2.2 删除 | 188 |
| 10.3 排序 | 190 |
| 10.3.1 交换排序 | 190 |
| 10.3.2 插入排序 | 193 |
| 10.3.3 快速排序 | 195 |
| 10.4 检索 | 198 |
| 10.4.1 顺序检索法 | 198 |
| 10.4.2 折半检索 | 200 |
| 习题 | 202 |
| 第十一章 输入输出程序设计 | 204 |
| 11.1 输入输出指令 | 204 |
| 11.2 CPU 与外设间的数据传送 | 205 |
| 11.2.1 输入输出接口 | 205 |
| 11.2.2 CPU 与外设之间的接口信号 | 205 |
| 11.2.3 CPU 与外设之间的数据传送方式 | 206 |
| 11.3 彩色显示器 | 207 |
| 11.3.1 彩色/图形适配器 | 208 |
| 11.3.2 彩色显示器编程 | 213 |
| 11.4 打印机编程 | 221 |
| 11.5 并行接口 | 223 |
| 11.5.1 8255A 的结构及其工作方式 | 224 |
| 11.5.2 8255A 在 IBM PC 上的用法 | 226 |

| | | |
|-------------|----------------------------------|------------|
| 11.6 | 8253 可编程定时器/计数器 | 227 |
| 11.6.1 | 概述 | 227 |
| 11.6.2 | 8253 的基本功能和内部结构 | 228 |
| 11.6.3 | 8253 的工作方式 | 229 |
| 11.6.4 | 8253 的应用 | 230 |
| | 习题 | 233 |
| 第十二章 | 中断程序设计 | 234 |
| 12.1 | 中断、外同步和空操作指令 | 234 |
| 12.1.1 | 中断指令 | 234 |
| 12.1.2 | 外同步指令和空操作指令 | 235 |
| 12.2 | 中断系统概述 | 236 |
| 12.2.1 | 中断请求与中断源 | 236 |
| 12.2.2 | 中断系统的功能 | 237 |
| 12.2.3 | 中断响应 | 237 |
| 12.3 | 8086/8088 的中断系统 | 238 |
| 12.3.1 | 外部中断 | 238 |
| 12.3.2 | 内部中断 | 240 |
| 12.3.3 | 中断向量表 | 241 |
| 12.4 | 中断程序设计 | 242 |
| 12.4.1 | 主程序设计 | 242 |
| 12.4.2 | 中断服务程序设计 | 242 |
| 12.4.3 | 中断服务程序设计中应注意的几个问题 | 243 |
| 12.5 | 8259A 中断控制器及其程序设计 | 243 |
| 12.5.1 | 8259A 的结构 | 244 |
| 12.5.2 | 8259A 的程序设计 | 245 |
| 12.5.3 | 8259A 在 IBM PC 系列机中的应用 | 248 |
| 12.6 | IBM PC 的 BIOS 调用及 DOS 功能调用 | 250 |
| 12.6.1 | BIOS 调用 | 250 |
| 12.6.2 | DOS 中断和功能调用 | 252 |
| 12.7 | 键盘中断程序设计 | 256 |
| 12.8 | 打印机中断程序设计 | 259 |
| 12.9 | 异步串行通信程序设计 | 261 |
| 12.9.1 | 异步串行通信简介 | 261 |
| 12.9.2 | 异步串行通信适配器 | 262 |
| 12.9.3 | 8250 的程序设计 | 263 |
| 12.9.4 | 通信程序设计 | 266 |
| | 习题 | 278 |
| 第十三章 | 磁盘操作及文件管理 | 280 |
| 13.1 | 磁盘结构 | 280 |
| 13.2 | 磁盘上的数据组织 | 281 |
| 13.2.1 | 保留区(引导记录) | 282 |

| | | |
|-------------|------------------------------------|------------|
| 13.2.2 | 文件分配表(FAT)..... | 282 |
| 13.2.3 | 文件目录表(FDT)..... | 285 |
| 13.3 | 磁盘 I/O 参数表及其用法..... | 286 |
| 13.3.1 | 磁盘 I/O 参数表(BPB)..... | 286 |
| 13.3.2 | BPB 表的用法..... | 286 |
| 13.4 | 磁盘基数表及其用法..... | 288 |
| 13.4.1 | 磁盘基数表..... | 288 |
| 13.4.2 | 软盘基数表的用法..... | 289 |
| 13.5 | DOS 文件管理..... | 296 |
| 13.5.1 | DOS 文件结构..... | 297 |
| 13.5.2 | 文件控制块 FCB 及其用法..... | 297 |
| 13.5.3 | DOS 文件句柄功能及其用法..... | 299 |
| | 习题..... | 301 |
| 第十四章 | 汉字处理技术 | 302 |
| 14.1 | 汉字的机内表示..... | 302 |
| 14.2 | 汉字的输入..... | 303 |
| 14.2.1 | 区位码和国标码..... | 303 |
| 14.2.2 | 拼音码..... | 306 |
| 14.2.3 | 拼形码..... | 310 |
| 14.2.4 | 形音码..... | 312 |
| 14.3 | 汉字的显示..... | 313 |
| 14.3.1 | 汉字的字模..... | 313 |
| 14.3.2 | 汉字显示的实现..... | 314 |
| 14.4 | 汉字的打印..... | 319 |
| 14.4.1 | 汉字打印机..... | 319 |
| 14.4.2 | 汉字打印的实现..... | 320 |
| | 习题..... | 325 |
| 第十五章 | 条件汇编与宏指令 | 327 |
| 15.1 | 条件汇编伪操作..... | 327 |
| 15.2 | 宏伪操作..... | 329 |
| 15.2.1 | 宏定义与宏结束伪操作命令..... | 329 |
| 15.2.2 | 参数的使用..... | 330 |
| 15.2.3 | 宏中的标号处理..... | 333 |
| 15.2.4 | 宏嵌套..... | 335 |
| 15.2.5 | 宏与子程序的区别..... | 338 |
| 15.3 | 重复块宏操作命令..... | 339 |
| 15.4 | 特殊宏操作命令..... | 341 |
| | 习题..... | 342 |
| 第十六章 | 8087/80287/80387 程序设计 | 344 |
| 16.1 | 概述..... | 344 |
| 16.2 | 8087/80287/80387 的逻辑结构及运行..... | 345 |

| | | |
|-------------|-------------------------------|------------|
| 16.2.1 | 8087/80287/80387 的逻辑结构 | 345 |
| 16.2.2 | 8087/80287/80387 的运行 | 348 |
| 16.3 | 数据类型及格式 | 350 |
| 16.4 | 8087/80287/80387 的指令系统 | 351 |
| 16.4.1 | 数据传送指令 | 352 |
| 16.4.2 | 算术运算指令 | 353 |
| 16.4.3 | 比较指令 | 356 |
| 16.4.4 | 函数指令 | 357 |
| 16.4.5 | 常数指令 | 359 |
| 16.4.6 | 处理器控制指令 | 359 |
| 16.5 | 8087/80287/80387 程序设计举例 | 361 |
| 16.5.1 | 整数运算 | 362 |
| 16.5.2 | 实数运算 | 368 |
| | 习题 | 372 |
| 第十七章 | 80286 与 80386 系统 | 373 |
| 17.1 | 80286 CPU 结构 | 373 |
| 17.2 | 80286 系统存储器管理 | 375 |
| 17.2.1 | 80286 的操作方式 | 375 |
| 17.2.2 | 有关特权的概念 | 375 |
| 17.2.3 | 保护虚地址方式的存储管理 | 375 |
| 17.3 | 特权级 | 378 |
| 17.4 | 80286 的中断系统 | 379 |
| 17.5 | 80286 增强与扩充的指令 | 380 |
| 17.5.1 | 增强的指令 | 380 |
| 17.5.2 | 扩充的指令 | 381 |
| 17.6 | 80386 CPU 结构 | 384 |
| 17.7 | 80386 存储器管理 | 386 |
| 17.8 | 80386 的寻址方式 | 388 |
| 17.9 | 80386 扩充的指令 | 389 |
| | 习题 | 392 |
| 附录 A | 8086/8088 指令系统 | 393 |
| 附录 B | 伪指令 | 400 |
| 附录 C | BIOS 调用说明 | 401 |
| 附录 D | DOS 功能调用说明 | 405 |
| 附录 E | IBM PC 的键盘输入码和 CRT 显示码 | 409 |
| | 参考文献 | 411 |

第一章 概 述

汇编语言是一种强有力的语言，它能透彻地反映、巧妙而充分地运用计算机硬件的功能及特点，便于编程人员根据自己的需要灵活地编制高级语言能实现和无法实现的各种程序，随心所欲地控制计算机的运行。汇编语言是计算机能提供的最快而又最有效的语言，也是能够利用计算机所有硬件特性的唯一语言。在许多对运行速度要求很高的场合，汇编语言是必不可少的，如操作系统、编译程序、实时控制等软件多数是用汇编语言编写的。下面我们首先介绍计算机语言的种类及其优缺点。

1.1 计算机语言

语言是人们之间用来表达意愿，交流思想，抒发感情的工具。两个人使用同种语言，则不必通过翻译而能直接通话，倘若两人各懂得一种语言，则必须通过翻译，通话才能顺利进行。人要使用计算机，让计算机做某些工作，人与计算机之间就必须交流信息。交流信息的工具是什么呢？就是计算机语言。人们利用计算机语言，告诉计算机某个问题应如何处理，先做什么，后做什么，遇到什么情况应如何处理等，即人们用计算机语言安排好处理步骤，每一步都是用计算机语言编写的，然后送入计算机内，计算机就按处理步骤，不折不扣地完成人们所规定的工作。我们用计算机语言描述的处理步骤，称为程序。而编制处理步骤的过程称为程序设计。

计算机语言可分为三类：机器语言、汇编语言和通用语言。前两种语言是面向机器的，一般称为低级语言，后一种是面向程序设计人员的，一般称为高级语言。

1.1.1 机器语言

我们在电子线路课程中已经学过了与门、或门、非门、触发器、寄存器、计数器、译码器、驱动器等等，它们的功能就是记忆、传送、改变二进制信息“1”或“0”。计算机的硬件设备均由这些部件组合而成，所以计算机运算过程也就是加工、传输二进制信息的过程。简单地说，这些二进制信息就是机器语言，计算机只认识这些二进制信息。进一步说，这些二进制信息代表不同的意义，有的代表数据，有的代表存数的地址，有的代表控制信号。我们把可以使计算机完成各种操作的“1”和“0”的不同组合的数码串称为机器指令，机器语言就是与机器硬件有紧密联系的机器指令的集合。

下面我们应用 8086/8088 机器语言编制计算两数之和的程序。设两个数据分别用 8 个二进制位表示（字节数据），存放在内存当前数据区的 1 号和 2 号单元，计算结果存入 3 号单元。程序清单如下：

```
1010 0000 }  
0000 0001 } 1号单元内容取入 AL 寄存器  
0000 0000 }
```

```

1000 1010 }
0010 0110 } 2号单元内容取入 AH 寄存器
0000 0010 }
0000 0000 }
0000 0000 } AL 的内容与 AH 的内容相加,结果存入 AL 中
1110 0000 }
1010 0010 }
0000 0011 } AL 的内容送入 3 号单元
0000 0000 }
1111 0100  停机

```

上述程序涉及两种操作：数据传送和数据相加。尽管要解决的问题很简单，涉及的操作少，但我们看到机器语言完全由“0”和“1”代码串组成，因此程序的编制困难、麻烦，很容易出错，而且一旦有错，很难发现。另一方面，机器的指令与机器硬件紧密联系，不同的硬件结构，机器的指令代码不同，用某种机器指令编写的程序在其它类型的计算机上就不能使用。综上所述，用机器语言编写的程序，有以下缺点：

- 1) 机器语言难记，程序难编，易错，调试困难。
- 2) 机器语言通用性差。

尽管这样，最初的计算机都是用机器语言编制程序，其优点如下：

- 1) 计算机能直接认识、执行。
- 2) 算法刻画细致。
- 3) 程序紧凑，占用内存空间少，执行速度快。
- 4) 能充分发挥计算机的硬件功能。

1.1.2 汇编语言

随着时间的推移和计算机的发展，更复杂的问题需要用计算机去解决（要求编制更大的程序），从而产生了新的思想，那就是用易于记忆的缩写符号来代替数字码，这些缩写符号类似于英文单词的缩写，有明确的意义，代表机器指令的操作码、操作数、操作数地址、寄存器名等等。这就是符号语言，上面的程序用符号语言编写如下：

```

MOV    AL,DATA1    ; 取第一个数据
MOV    AH,DATA2    ; 取第二个数据
ADD    AL,AH        ; 求和
MOV    RLT,AL      ; 保存结果
HLT                    ; 停机

```

计算机不能直接认识、执行符号语言程序，需要先将其译为机器语言程序，然后由机器执行。翻译工作可以由人工完成，也可以由机器完成。但由于人工翻译繁琐且容易出错，所以一般由计算机完成。让计算机翻译时，必须规定组成符号语言的字符集、符号名字的组成规则、数据的表示方法、语句的书写格式等等。这就是符号语言的词法和语法。另外，还要告诉计算机一些必要的信息，如数据以什么形式存放，放在何处，程序代码存于何处等。这些信息用具有一定意义的符号表示，通常称之为伪指令或汇编命令。伪指令符号及符号语言的词法、语法规则构成了汇编语言。

在计算机中，把汇编语言程序翻译成机器语言程序的过程称为汇编过程，完成汇编过程的程序称为汇编程序。汇编程序加工的对象是汇编语言程序，称为源程序，而汇编产

生的结果是机器语言程序，称为目标程序或目的程序。

前面所述程序用 IBM PC 汇编语言编写如下：

```

DSEG  SEGMENT                                ; 数据段开始
DATA1 DB 15H
DATA2 DB 26H
RLT   DB 0
DSEG  ENDS                                    ; 数据段结束
CSEG  SEGMENT
      ASSUME CS,CSEG,DS,DSEG
START: MOV AX,DSEG                            ; 设置数据段基址
      MOV DS,AX
      MOV AL,DATA1                            ; 取第一个数据
      MOV AH,DATA2                            ; 取第二个数据
      ADD AL,AH                                ; 求和
      MOV RLT,AL                              ; 保存结果
      HLT                                     ; 停机
CSEG  ENDS                                    ; 代码段结束
      END START
    
```

把汇编语言程序变为可执行程序的过程如图 1.1 所示。

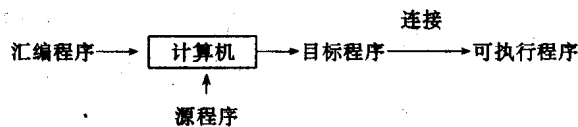


图 1.1 汇编语言程序执行过程

汇编语言程序与机器语言程序比较起来，具有容易编制、出错机会少、容易调试的优点，还保持了机器语言程序的算法刻画细致、程序紧凑、占用内存空间少、执行速度快以及可以直接使用计算机的全部资源的优点。但是汇编语言程序还存在着通用性差的缺点。同时产生了一个新的缺点：计算机不能直接认识、执行汇编语言程序。

1.1.3 通用语言(高级语言)

通用语言是用来对计算机操作步骤进行描述的一整套标记符号、表达格式、结构及其使用的语法规则。如 BASIC, FORTRAN, COBOL, Pascal 和 C 语言。目前世界上高级语言有数百种。用通用语言编写程序容易表达算法，不依赖具体的计算机和机器指令，通用性强，其缺点是计算机不能直接认识、执行，需通过翻译变成机器语言程序，运行速度低，处理过程刻画得不具体。

通用语言程序译为可执行程序的过程如图 1.2 所示。

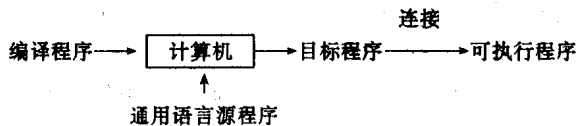


图 1.2 通用语言程序执行过程

三种语言各有利弊，一般地说，对于使用计算机进行科学计算或事务处理的人员，只掌握高级语言就可以了，而对开发计算机系统软件或把计算机用于工业过程控制、实时控制的人员，必须掌握汇编语言，以分析、修改、扩充计算机的系统软件，增加计算机的功能。

过程控制和实时控制程序要求程序执行和处理速度高,能灵活地驱动各种设备。

汇编语言因机型不同(因汇编程序不同)而异,但编程的原理、方法与技巧相同,只是指令的助记符号、伪指令的符号不同,数量的多少及语法不同,只要熟练掌握一种汇编语言,则对其它的汇编语言仅稍看一下说明书便可轻而易举地掌握。本书虽以 IBM PC 为背景机讲述程序设计,但所涉及的程序设计概念、原理、方法和技巧具有普遍的指导意义,适合于任何汇编语言程序设计。汇编语言与具体的计算机结构有关,进行汇编语言程序设计,不仅要考虑解决问题的本身的逻辑,还要考虑计算机硬件资源的使用,这就要求汇编语言程序设计人员对计算机的硬件结构在逻辑上有比较清楚的了解。

1.2 计算机系统

相关事物组成的集合体称为系统,组成系统的事物称为该系统的元件或部件,计算机系统由软件子系统和硬件子系统组成。

1.2.1 硬件子系统

硬件子系统是指组成计算机系统的所有电子的、机械的、光学的和磁性的元部件。硬件子系统通常包括中央处理器,主存储器,接口装置,外部设备(输入设备、输出设备、外存储器),外围设备(A/D 转换器、D/A 转换器、开关量输入/输出)。

计算机硬件系统的示意图如图 1.3 所示。

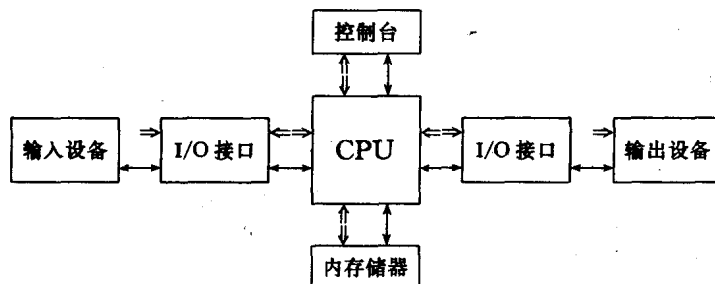


图 1.3 计算机硬件子系统示意图

(1) 中央处理器

中央处理器通常由算术逻辑部件,寄存器组,标志寄存器和控制部件等组成。

算术逻辑部件完成算术和逻辑运算。

寄存器组由若干寄存器组成,有的用于寄存参加运算的数据、运算后的结果;有的用作控制指令执行的指令计数器;有的用作堆栈指针寄存器等。

标志寄存器用于保存运算结果的状态和 CPU 用的逻辑控制标志,以使 CPU 根据这些标志位进行判断,决定程序的执行顺序。

控制部件用于保存机器指令,对指令进行译码,产生控制各个部件的信号。中央处理器是计算机硬件子系统的核心。

(2) 存储器

存储器分内存和外存。内存又称主存储器,用于存储计算机的各种语言程序,要处理的原始数据,中间结果及最终结果等。

主存储器存放信息的容量称为内存容量，通常用存储空间的多少来表示。存储空间的基本单位为字节(8位二进制位)，每1024个字节称为1KB。每个存储空间又称为一个单元，每个单元有一个编号，称为地址，向存储空间存数据或从存储空间读取数据时，都必须指出它的地址。从使用计算机的角度上说，内存空间越大越好，但由于受到技术上的限制及价格因素的影响，内存空间不能太大，以Z-80为CPU的微型机的内存空间最大为64KB，以8086/8088为CPU的微型机内存空间最大为1MB(1024KB)。

外存又称为辅助存储器，如磁盘(硬磁盘和软磁盘)和磁带等。以8086/8088为CPU的微型机一般配有硬盘(温氏盘)，容量为10MB、20MB或40MB等。软盘每个盘片的容量为180KB、360KB或1.2MB等，当要存储的信息较多时，可使用多片软盘，从这个意义上说，它的容量是无限的，所以又称为“海量”存储器。

(3) 接口装置

人们和计算机交换信息都是通过外部设备进行的，外部设备一般是机械设备，其工作速度与中央处理器的工作速度比起来低得多，而且不同的外部设备传输数据的编码格式、所需要的控制信号种类多少也不相同。因此，外部设备与CPU之间不能直接交换信息，需要有一个具有若干个寄存器和逻辑控制电路部件作为二者交换信息的缓冲部件，该部件称为接口装置，简称接口。

接口又分为并行接口和串行接口。在并行接口中，每一位数据都有自己的通路(数据传输线)，譬如，并行传送一个字节，需要8根数据传输线；在串行接口中，数据是一位接一位传输的，只要一根传输线。为了使接口电路具有通用性和灵活性，其中设有用于指定工作方式的控制寄存器，用户可以根据连接设备的种类及对控制信号的要求，对接口电路中的控制寄存器通过指令进行设置，这样的接口电路称为可编程接口电路。接口电路的通用性越强，内部的控制寄存器越多，需设置的信号越多，编程就越复杂。

不同的厂家为自己生产的计算机研制了配套的并行、串行接口电路，如Z-80系列的接口电路有PIO, SIO, CTC等，Intel系列的接口电路有8255, 8251, 8253, 8259等等。

(4) 输入输出设备

输入输出设备又称外部设备，是用来实现人与计算机交换信息的装置。输入设备是用来向计算机的主存储器或CPU送入程序或数据的装置。常见的输入设备有键盘，读卡机，光笔，数字化仪等；输出设备是用来将计算机处理的结果以某种形式输出给用户的装置，常见的输出设备有显示器，打印机，绘图仪和穿卡机等。

不同的计算机配置的外部设备种类、数量、所需的控制信号、数据编码格式等都不统一，使用时需参看有关技术说明书。

(5) 外围设备

当计算机用于控制时，现场信号要输入到计算机，计算机要输出控制信号到现场。这些信号分为模拟量和开关量，为区别一般的外部设备，完成这些信号输入/输出的设备，称为外围设备。

模拟量用于控制计算机，大量的现场信号经过传感器把非电量(例如温度，压力，流量，位移等)转换为电量，并经放大即得模拟电压和电流。这些模拟量必须先经过A/D转换才能输入计算机(位数由A/D转换精度确定)；计算机输出的控制信号，也必须先经过D/A转换，变为模拟量才能去控制执行机构。

开关量是一些两个状态的量，如计算机控制电机的运转与停止、开关的接通与断开、阀门的打开与关闭等等。这些量只要用一位二进制数即可表示，这就是开关量的输出。但有时在控制过程中也需要开关量的输入。字长为 8 位的机器一次可输入或输出 8 个这样的开关量。

1.2.2 软件子系统

软件子系统是指为了充分发挥计算机硬件子系统的功能，方便用户使用计算机，提高计算机系统效率而编制的各种程序。

软件子系统对于计算机系统来说是极为重要的，在硬件子系统已经确定的情况下，计算机系统功能的强弱，应用范围的大小，使用的方便性、灵活性等完全取决于软件功能的强弱。

软件子系统由系统软件和应用软件组成，而系统软件又分为三类：面向用户的软件，面向计算机维护人员的软件和面向计算机本身的软件；应用软件是用户为解决自己的问题而编制的软件。

1. 面向用户的软件

这类软件有利于用户使用计算机解决自己的问题，还有利于用户编制、调试、装配自己的应用程序。

(1) 语言加工软件

它用来把用户用各种计算机语言编制的源程序转换为计算机硬件能直接处理的机器代码(目标程序)。如：汇编程序、编译程序、解释程序等。

汇编程序是将汇编语言程序转换为目标程序的软件。

编译程序是将通用语言源程序转换为目标程序的软件。任何一种通用语言都有自己的编译程序。如：FORTRAN 语言有 FORTRAN 编译程序，Pascal 语言有 Pascal 编译程序等等。

(2) 辅助加工软件。用户借助辅助加工软件来编辑、修改自己的源程序，装配、连接、调试自己的目标程序，这类软件有编辑程序(行编辑程序，全屏幕编辑程序)，连接程序，纠错程序等等。

行编辑程序：用户利用此程序将自己的源程序送入内存，逐句逐字进行编辑操作，删除、插入、查找、替换字或字符串。IBM PC 微机的行编辑程序名是 EDLIN。

全屏幕编辑程序：其编辑功能与行编辑程序相同，只不过它可以使光标在整个显示屏上任意移动，对光标指示的任何字符进行各种编辑操作，它比行编辑程序更灵活、更方便。在 IBM PC 上有很多种全屏幕编辑程序，如 WS, WPS, XE 等等。

连接程序：把经过汇编或编译产生的目标程序(一个或几个)连接在一起成为可执行的机器语言程序，以备装入运行。IBM PC 上的连接程序名为 LINK。

纠错程序：用户借助此程序来控制自己程序的执行，设置断点、逐条跟踪、检查每条指令或某段程序运行的结果、检查修改寄存器或存储单元的内容等。IBM PC 上的纠错程序名为 DEBUG。