

航空高等院校教材

Z80程序设计基础

舒忠正 林钧海 李立新 编



TP312

157

航空专业教材编审组

TP312
157

Z80 程序设计基础

舒忠正 林钧海 李立新 编

航空专业教材编审组

内 容 提 要

本书是以微型计算机 (Z80 作 CPU) 为背景的程序设计基础教材, 书中详细介绍了各指令的功能及程序设计的基本方法。为了学以致用, 在介绍程序设计方法过程中, 给出了一些常用程序的例子, 书中例题丰富, 文字通俗易懂, 便于自学; 有关章节都配有上机实习指导, 以利于实践操作的训练, 书中所有程序 (包括例题和习题) 均在 Z80 计算机上进行了调试, 书后列有指令系统一览表, 作为学完此书的读者的“使用手册”。

本书可作为理工科院校计算机各专业的学生教材, 也可作为从事计算机工作有关人员的参考书。

Z80 程 序 设 计 基 础

舒忠正 林钧海 李立新 编

航空专业教材编审组出版
南京航空学院印刷厂印装

*

787×1092 1/16 印张 18.75 480.千字

1983 年 12 月第一版 1983 年 12 月第一次印刷 印数: 0,001—1,500 册

统一书号: 35195j 定价: 1.93元

前 言

“程序设计基础”是计算机各专业的核心基础课之一，也是从事计算机工作人员的必读之课。随着微型计算机(一般以 Z80 为 CPU)的普及，“Z80程序设计基础”就应运而生了，本书就是由航空工业部计算机软件教材编审小组研究确定，责成我们编写的。部教材编审小组召集了有关院校的同行人讨论了该书的编写大纲，在系的领导下，我们安排了三名教学经验较丰富的中年教师来撰写。为保证质量及便于同志们今后调用方便，我们对所有程序都进行了上机调试。

本书可作为理工科院校计算机各专业的学生教材，也可作为从事计算机工作的有关人员的参考书。具有一般计算机知识的同志，第一章可以不读；掌握了一种计算机的程序设计基础的同志，第二章也只需作对比过渡式的阅读；学过微处理机原理的同志，第一、二章都可以不读。根据我们的教学实践全书约 60~80 学时即可教完（*号内容除外，不包括上机实习），为搞好上机实习，除在第七章专门作了理论指导外(可分散插在有关章节讲)按章都作了上机实习部署，共上机实习七次，每人需14个机时。

由于各厂家提供的 Z80 微机系统上配置的系统软件如操作系统、动态调试工具、文本编辑程序等略有不同，读者在使用计算机时，应根据厂家提供的用户手册进行操作。

我们是分工撰写，统一组稿的，分工情况如下：第一章（除 §1.1 外）、第三章、第四章由舒忠正同志编写；第二章、第六章由林钧海同志编写；第一章 §1.1、第五章、第七章由李立新同志编写。在编写过程中，院教务处的有关同志、兄弟教研室及本教研室的其它老师都提出了许多宝贵意见；周以铨同志参与了本书编写的组织领导和编辑工作；夏建同志仔细阅读并清抄了全部书稿，提出了一些改进意见；在上机调试过程中，404机房的同志给予大力的支持，等等，等等，在此一并致谢！

部教材编审室的有关同志及西北工业大学张遵濂付教授、韩云霞老师审阅了全书，提出了许多宝贵意见，在此表示感谢！

鉴于我们水平有限，缺点和错误在所难免，恳切希望同志们提出批评意见。

编 者

83.1.

目 录

第一章 预备知识	(1)
§ 1.1 Z80 微型计算机简介	(1)
一、Z80 微型机的组成	(2)
二、Z80的CPU	(2)
三、Z80 微型机的存贮器	(4)
四、Z80 微处理器的技术指标	(5)
五、Z80 微型计算机的软件系统	(5)
六、国产DJS-050系列计算机	(6)
§ 1.2 数的表示法	(6)
一、各种进位计数制	(6)
二、各种进位制之间的转换关系	(10)
三、数的原码、补码和反码表示	(15)
四、数的定点和浮点表示法	(21)
§ 1.3 程序设计的一般概念	(23)
一、地址、字节和字长	(23)
二、指令的组成部分及其形式	(23)
三、有效地E的概念	(26)
四、字符码与奇、偶校验	(26)
五、源(SOURCE)程序和目标(OBJECT)程序	(27)
习题	(28)
第二章 Z80 指令系统介绍	(30)
§ 2.1 寻址方式	(30)
一、立即数寻址(IMMEDIATE ADDRESSING)	(30)
二、直接寻址(DIRECT ADDRESSING)	(31)
三、寄存器寻址(REGISTER ADDRESSING)	(31)
四、寄存器间接寻址(REGISTER INDIRECT ADDRESSING)	(31)
五、变址寻址(INDEXED ADDRESSING)	(31)
六、相对寻址(RELATIVE ADDRESSING)	(31)
§ 2.2 数据传送指令	(32)
一、立即数的传送指令	(33)
二、寄存器之间的数据传送指令	(35)
三、寄存器与存贮器之间的数据传送指令	(36)
四、内存贮器之间的数据传送指令	(40)

§ 2.3 算术运算指令	(42)
一、加法运算指令	(42)
二、减法运算指令	(48)
三、加1运算指令	(51)
四、减1运算指令	(53)
五、求补运算指令	(55)
六、十进制调整累加器	(55)
§ 2.4 逻辑运算指令	(57)
一、逻辑运算的概念	(57)
二、逻辑运算指令	(59)
A. “或”运算指令	(59)
B. “与”运算指令	(61)
C. “非”运算指令	(62)
D. “异或”运算指令	(63)
§ 2.5 堆栈操作指令	(64)
一、设置堆栈指针	(66)
二、数据进栈	(66)
三、数据出栈	(67)
四、寄存器对的内容与栈顶内容的交换	(68)
§ 2.6 转移指令	(70)
§ 2.7 比较指令与检索指令	(76)
一、比较指令	(76)
二、检索指令	(78)
§ 2.8 移位操作指令	(81)
一、循环移位指令	(81)
A. 连同进位C的循环左移指令	(81)
B. 不连进位C的循环左移指令	(84)
C. 连同进位C的循环右移指令	(87)
D. 不连进位C的循环右移指令	(89)
二、位移指令	(91)
A. 算术左移指令	(91)
B. 算术右移指令	(94)
C. 逻辑右移指令	(96)
§ 2.9 位操作指令	(98)
一、位设置指令(SET BIT)	(98)
二、位清除指令(RESET BIT)	(99)
三、位测试指令(TEST BIT)	(100)
§ 2.10 上机实习	(102)
实习一	(102)

实习二	(105)
习题	(107)
第三章 程序设计的基本方法	(114)
§ 3.1 框图(FLOWCHART)	(114)
§ 3.2 循环程序的编制	(115)
一、单重循环程序	(116)
二、多重循环程序	(122)
§ 3.3 分枝程序设计	(126)
一、简单分枝程序的编制	(126)
二、多分枝程序的编制	(126)
§ 3.4 子程序的设计	(135)
一、子程序 (SUBROUTINE) 的概念	(135)
二、子程序的编制和使用	(136)
三、传送参数的方法	(139)
四、子程序举例	(143)
五、递归子程序(RECURSIVE SUBROUTINES)	(146)
六、关于编制子程序的一些注意事项	(149)
§ 3.5* 程序设计的一般方法	(150)
一、在数字计算机上进行工作的一般步骤	(150)
二*、程序设计的其它技术	(150)
§ 3.6 上机实习	(151)
一、关于汇编语言源程序文件的建立	(151)
二、关于源文件的修改	(152)
三、使用汇编程序进行汇编	(152)
四、目标程序的连接和装入命令	(154)
五、目标程序的调试命令	(154)
实习三	(155)
实习四	(155)
习题	(156)
第四章 常用程序的设计	(160)
§ 4.1 代码转换和数制转换	(160)
§ 4.2 定点运算程序	(166)
一、四字节的加法和减法	(166)
二、乘法和除法	(167)
三、十进制的加法和减法	(173)
四、通用多精度运算程序	(174)
五、多精度乘法程序	(177)
§ 4.3 字符串和表处理	(179)

一、字符串·····	(179)
二、表格·····	(181)
三、查表方法·····	(182)
四、表的删除和插入·····	(188)
§ 4.4 队列与链接·····	(190)
§ 4.5 上机实习·····	(194)
实习五·····	(194)
习题·····	(195)
第五章 输入/输出指令和程序中断·····	(196)
§ 5.1 计算机和外部设备之间的数据交换·····	(196)
一、计算机和外部设备之间数据传送方式·····	(196)
二、外部设备·····	(196)
§ 5.2 输入/输出指令·····	(197)
一、设备口地址·····	(197)
二、输入/输出指令·····	(198)
§ 5.3 磁盘控制和磁盘存贮管理·····	(202)
一、磁盘和内存之间的信息交换·····	(202)
二、磁盘控制·····	(203)
三、磁盘读写应用举例·····	(206)
§ 5.4 程序中断和Z80微机的中断系统·····	(208)
一、程序中断概念·····	(208)
二、程序中断对计算机硬件的要求·····	(209)
三、中断系统提高主机和外部设备之间的并行工作能力·····	(209)
四、Z80 微型计算机的中断系统·····	(211)
§ 5.5 其它有关输入/输出指令·····	(215)
§ 5.6 上机实习·····	(218)
实习六·····	(218)
习题·····	(218)
第六章 Z80 汇编语言·····	(219)
§ 6.1 汇编语言·····	(219)
一、机器语言·····	(219)
二、汇编语言·····	(220)
§ 6.2 汇编语言程序的组成格式·····	(220)
§ 6.3 标号域和操作数域·····	(222)
一、标号域·····	(222)
二、操作数域·····	(223)
§ 6.4 操作码、硬指令与伪指令·····	(226)
一、数据定义伪指令·····	(226)

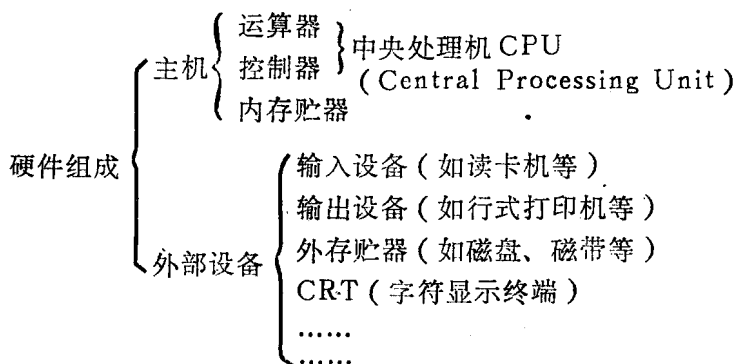
二、保留存储单元的伪指令DEFS.....	(227)
三、伪指令ORG.....	(227)
四、伪指令END.....	(228)
五、伪指令EQU.....	(228)
§ 6.5 汇编语言程序举例.....	(229)
§ 6.6 上机实习.....	(242)
实习七.....	(242)
习题.....	(242)
第七章 上机实习指导	(244)
§ 7.1 CP/M软盘操作系统.....	(244)
一、CP/M软盘操作系统的特点.....	(244)
二、CP/M操作系统功能.....	(245)
三、CP/M操作系统的调用方式.....	(249)
四、CP/M操作系统的命令调用.....	(250)
§ 7.2 CP/M操作系统文本编辑程序ED.....	(258)
一、文本编辑程序 ED(EDITOR) 的主要功能.....	(258)
二、ED程序的操作命令.....	(258)
三、ED程序的命令串.....	(262)
四、文本检索和修改命令.....	(263)
五、ED文本编辑错误标志.....	(264)
§ 7.3 动态调试程序DDT.....	(265)
一、动态调试程序DDT的功能及其调用方法.....	(265)
二、DDT程序的命令.....	(266)
习题.....	(269)
附录 I Z80 指令系统一览表	(270)
表 1 8 位数据传送指令.....	(270)
表 2 16 位数据传送指令 (包括堆栈操作).....	(272)
表 3 块数据传送和检索指令.....	(275)
表 4 8 位算术运算 (包括比较) 和逻辑运算指令.....	(276)
表 5 通用算术指令与 CPU 控制指令.....	(278)
表 6 16 位算术运算指令.....	(279)
表 7 移位指令.....	(280)
表 8 位操作指令.....	(282)
表 9 转移指令.....	(283)
表10 CALL指令和RETURN指令.....	(285)
表11 输入/输出指令.....	(286)
附录 II ASCII代码字符表	(288)
参考资料.....	(289)

第一章 预 备 知 识

本章介绍与程序设计有关的预备知识, 其中涉及计算机组成的内容, 在“计算机组成原理”课程中有专门的阐述, 这里仅介绍学习程序设计和使用计算机时涉及到的一般常识, “程序设计一般概念”一节介绍后继各章要用到的一些名词术语, 要求读者对它们有初步的了解。关于数的表示及其相互转换方法则是学习本课程以及其它计算机专业课的基础。

§ 1.1 Z80 微型计算机简介

一般计算机的硬件组成如下表所示:



Z80 微型计算机也具有上述主要组成部分, 下面我们就 Z80 微型计算机作一简要介绍。

Z80 微型计算机 (Z80 MICROCOMPUTER), 是美国 ZILOG 公司研制的以 Z80 微处理器作为 CPU 的一种微型计算机。由于 Z80 微处理器的功能完善, 它已被国内外许多厂家选用, 用来组装成微型计算机系统。如 DYNABYTE 公司的 DB-8 型微型计算机 (也称 A 档机), CROMEMCO 公司的系统 III (SYSTEM THREE) 微型计算机等等。

Z80 微处理器兼容了 INTEL 公司的 8080 微处理器的全部功能。并得到了较大的扩充, 基本指令由 8080 微处理器的 78 条增加到 158 条。同时, 指令执行的时间也较快。Z80 微型机目前已广泛地应用在各个方面。如科学计算, 数据处理以及商业统计等等。图 1—1 中给出了 DYNABYTE 公司的 DB-8 机的结构, 以此说明微型计算机系统的一般情况。同时, 对各个部件作一个简要的说明。

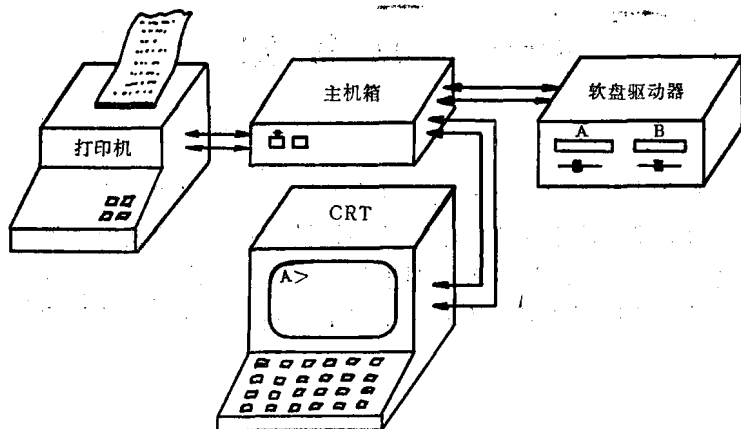


图 1—1

一、Z80 微型机的组成

1. 主机箱

Z80 主机箱中有 S-100 母线框架和电源，框架可以插入 12 块电路板，目前仅插有三块电路板，它们是：

(1) CPU 板 上面有 Z80 的芯片以及输入输出接口电路等。

(2) 64K RAM 板 主要由 32 块 SC74, 8216 MOS 片组成，另外还有辅助电路及体选开关。

(3) 磁盘控制主板 FDC 它最多可带 8 台软盘控制器。

2. 软盘控制器

每个机箱装有两个软盘驱动器，它们各自有独立的控制电路，可以插入两个盘片，图中指出的是 A 盘驱动器和 B 盘驱动器。

3. CRT

CRT 是带有键盘的字符显示器，DB-8 系统中 CRT 作为控制台输入输出用，编号为 1410。

4. 行式打印机

打印机的型号为 810 型，打印格式为 5×7 的针式打印。

根据实际情况的需要，还可以配置纸带输入机和快速穿孔机（有关外部设备的详细情况将在第五章中讨论）。

DB-8 微型计算机的电路逻辑由图 1—2 指出。

二、Z80 的 CPU

为了程序设计的需要，下面对 Z80 CPU 的结构作一个介绍。

1. 运算器

运算器能完成十二种基本算术运算和逻辑运算，即加法、减法、与、或、比较、异或、位测试、位置位、位清除、增 1、减 1、移位等等。

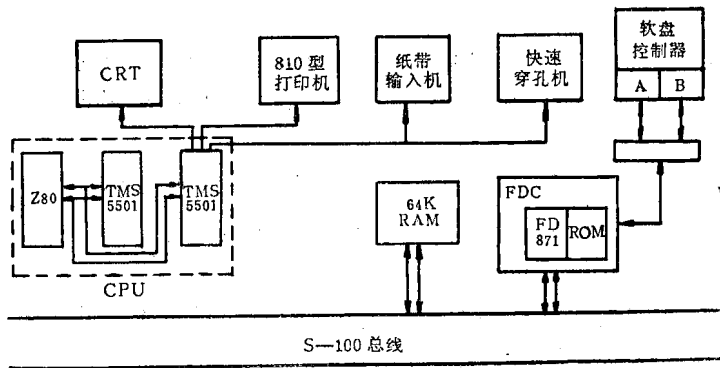


图 1—2

2. 控制器

主要是控制内部总线和寄存器之间的数据传送以及指令译码等功能。

3. 寄存器

Z80 微处理器的寄存器阵列包括 208 个二进制位 (BIT) 的读/写存储器。分别由 18 个 8 位寄存器和 4 个 16 位寄存器组成。全部 Z80 的寄存器是由 静态 RAM (RANDOM ACCESS MEMORY) 实现的。如图 1—3 中指出。

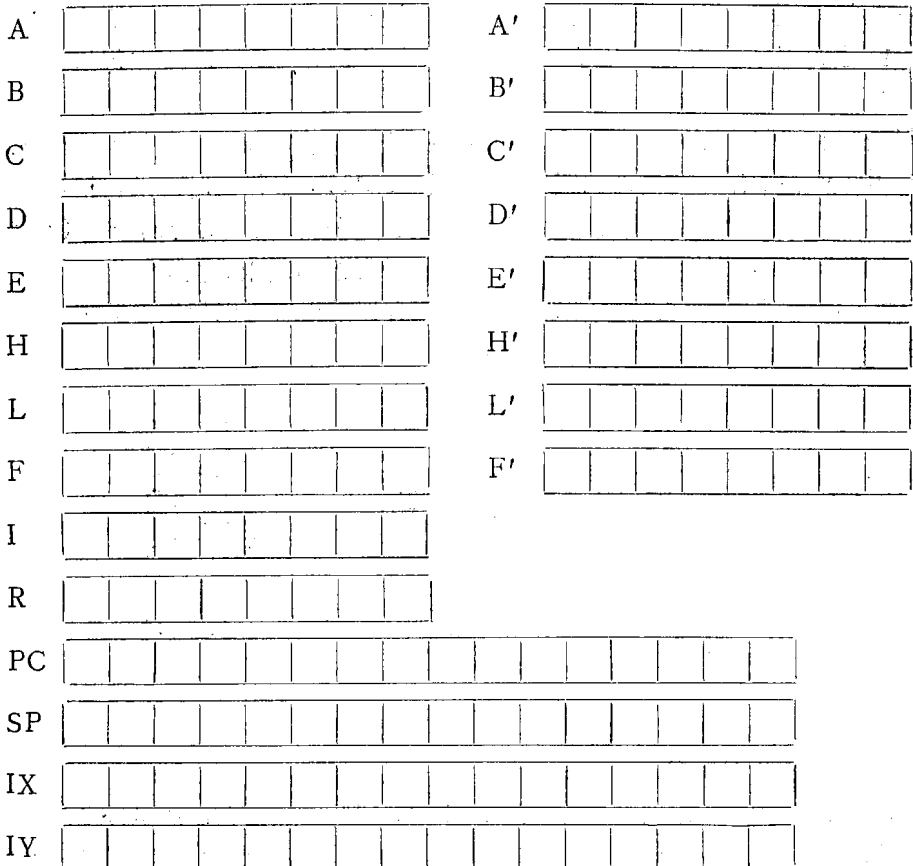


图 1—3

其中 A 是累加器。寄存器共有两组，分别作为前台和后台（供替换用，是前台的一组寄存器的备份）使用。

B、C、D、E、H、L 为 8 位寄存器，可以单独使用，也可以成对使用。

F 是状态寄存器，也称为标志寄存器，它指出运算之后的状态标志。每一个标志位的意义如下：

	7	6	5	4	3	2	1	0
F	S	Z		AC		P/O	N	C

其中：

C——进位标志 在算术运算的过程中，当有进位时，C 置 1。进位标志也可由其它操作设置（有关进位概念后面详细介绍）。

Z——零标志 当运算结果为零时，Z 位置 1，否则 Z 位置 0。

S——符号标志 在带符号数运算时，当运算结果为负时，符号标志位置 1，否则置 0。即符号位标志应与累加器 A 的最高位（左起第一位）相一致。

P/O——奇偶和溢出标志 是一个双重使用的标志位。在进行逻辑操作时，它表示奇偶标志；在进行算术运算和某些运算时，它表示溢出标志。当运算结果有溢出时，该位置 1，否则置 0。

AC——辅助进位标志 若低 4 位有进位（加法）或借位（减法），则该位置 1。

N——减法运算标志 当进行的最近一次运算为减法时，该位置 1，否则置 0。

PC 是 16 位的程序计数器。PC 的内容表示程序将要执行的地址，在每执行完一条指令后，PC 中的内容将进行修改。

SP 是堆栈指示器(STACK POINTER)，有时称为堆栈指针。它指出堆栈地址，此地址可以设在 RAM 存储器中的任何位置。当压入(PUSH)信息代码时，堆栈指针将自动地向小的地址方向减少；当弹出(POP)信息时，堆栈指针将自动向大的方向修改（见图 1—4）。

IX, IY 是两个 16 位的变址寄存器，它们在指令寻址时是很有用的。

I 是中断页面寄存器。在进行中断处理时，I 寄存器的内容可以提供中断地址的高 8 位。

R 是刷新寄存器。该寄存器用户是不用的。

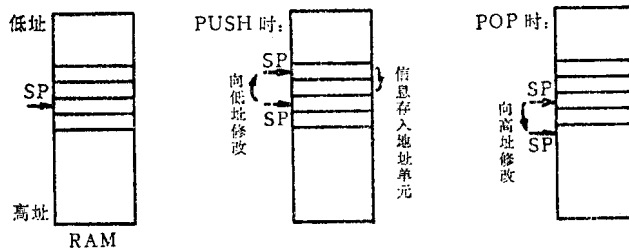


图 1—4

三、Z80 微型机的存储器

在 Z80 微机系统中，提供了各种容量的 ROM (READ ONLY MEMORY) 只读存储器 和 RAM (RANDOM ACCESS MEMORY) 随机访问存储器。ROM 存储器通常用来存放系统程序，如监控程序 MONITOR。RAM 通常用来存放程序、数据等信息。

RAM 存储器的地址编码

在 Z80 系统中, RAM 存储器是以字节为单位的, 存储器的容量分别有 16K, 32K 以及 64K 等等。其地址编码是以十六进制数来表示, 下面给出三种内存容量的地址编码示意图 (见图 1—5)。

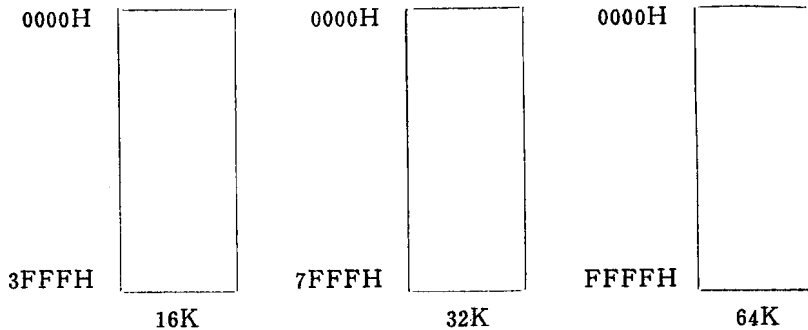


图 1—5

四、Z80 微处理器的技术指标

中央处理机 (CPU) 主振频率: 4MHZ

周期时间: 250nS

最小指令执行时间: 1μS

指令条数: 158 条

总线: 工业标准 S-100 总线

软盘驱动器: 8 个

磁盘容量: 8" 单面单密度盘 256K

8" 双面双密度盘 1000K

5" 单面单密度盘 80K

5" 双面单密度盘 181K

RAM 存储器容量: 64K

五、Z80 微型计算机的软件系统

由于 Z80 微型计算机在各方面得到广泛的应用, 许多厂家, 根据自己的特点和需要, 先后研制了各种软件系统。下面以 DB-8 微型计算机系统为例来说明软件系统的组成。

1. CP/M 软盘操作系统 (CONTRAL PROGRAM/MONITOR)

该系统主要是外部设备的输入输出管理程序和磁盘文件管理系统, 它是人与计算机系统的界面。操作人员是通过 CP/M 操作系统来对计算机进行控制的, 其它各种软件均以 CP/M 操作系统作为支持 (有关 CP/M 操作系统的内容在后面详细讨论)。与 CP/M 操作系统相当的有 CROMEMCO 公司的 CDOS 操作系统。

2. 文本编辑程序 ED (EDITOR)

该软件是用来建立、修改 ASCII 代码字符集源文件, 如汇编语言中的源文件, BASIC 语言中的源文件等等。当磁盘上没有该源文件时, ED 程序可以建立该源文件; 当源文件有错误时, ED 程序可以对源文件进行修改, 包括插入, 删除, 替换等等。

3. 动态调试程序 DDT (DYNAMIC DEBUGGING TOOL)

该程序能对可执行的文件进行跟踪调试, 如 COM 文件, HEX 文件。在调试过程中, 可以根据用户的需要设置断点, 能够显示寄存器和存贮器中的内容, 同时也能对寄存器和存贮器中的内容进行修改。

4. 汇编程序 ZASM (Z80 ASSEMBLER)

该程序能将 Z80 汇编语言规定语法编写的符号指令程序汇编成机器代码的目标程序。要指出的是通常的汇编程序 ASM, 仅仅能对 8080 的指令进行汇编 (关于汇编语言, 以后将作详细介绍)。

5. 高级语言

Z80 微型计算机系统中已配置了多种的高级语言。例如

BASIC 语言 包括有基本 BASIC 语言, 16K 扩展 BASIC 语言, 32K 的结构 BASIC 语言, CBASIC 语言 (它是一个面向商业的编译—解释系统) 及 MBASIC 语言。

FORTRAN 语言, COBOL 语言, PL/M 语言以及 PASCAL 语言等等。

六、国产 DJS-050 系列计算机

近年来国产微型计算机正在大量地问世, 其中以 Z80 微处理器作为 CPU (或以 8080 作为 CPU) 的微型计算机系统占了相当的数量, 国内统称为 DJS-050 系列计算机, 它在功能上和 Z80 (或 8080) 微型机相一致, 软件也可相容, 但是, 由于计算机的硬件环境不同以及结构上的差别, 使得软件中的核心 CP/M 操作系统要作适当的修改。这样, 才可能使计算机的原有资源 (包括硬件资源和软件资源) 得到充分的利用。

§ 1.2 数的表示法

计算机中的数的表示方法与我们日常所使用数的十进制表示不一样, 它是用符号 0 和 1 的组合来表示的, 在计算机中, 这些符号由开关元件产生, 这样的数我们称之为二进制数。

下面我们介绍数的各种表示形式及其相互之间的转换方法。

一、各种进位计数制

1. 数的十进制 (Decimal) 表示

这是日常生活中用得最多的一种进位制, 它有十个数码: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 计数原则为“逢十进一”。0, 1, 2, 3, ……8, 9, 10, 11, ……任意一个十进制数都可以表示为以 10 为基数的幂运算形式。

例如十进制数 2.0375 可表示为

$$2.0375 = 2 \times 10^0 + 0 \times 10^{-1} + 3 \times 10^{-2} + 7 \times 10^{-3} + 5 \times 10^{-4}$$

一般任意一个十进制数 $S_{(+)}$ 都可表示为

$$\begin{aligned} S_{(+)} &= K_n \cdot 10^n + K_{n-1} \cdot 10^{n-1} + \cdots + K_1 \cdot 10^1 + K_0 \cdot 10^0 \\ &\quad + K_{-1} \cdot 10^{-1} + K_{-2} \cdot 10^{-2} + \cdots + K_{-m} \cdot 10^{-m} \\ &= \sum_{j=-n}^m K_j \cdot 10^j \end{aligned}$$

其中 K_i 是数码 0~9 中的任意一个, m 、 n 为正整数; 10^j 为基数 10 的幂称为位值。

故数 $S_{(+)}$ 是数码与位值相乘的总和。

2. 数的二进制 (Binary) 表示

这是一种最简单的进位制, 计算机中采用的是二进制计数法。

它有两个数码: 0, 1, 计算原则为“逢二进一”。0, 1, 10, 11, 100, 101, 110, 111, ……任意一个二进制数都可以表示为以 2 为基数的幂运算形式。

例如二进制数 11.01(二) 可表示为

$$\begin{aligned} 11.01(\text{二}) &= 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} \\ &= 2 + 1 + \frac{1}{4} = 3.25 \end{aligned}$$

一般地有

$$\begin{aligned} S(\text{二}) &= K_n \cdot 2^n + K_{n-1} \cdot 2^{n-1} + \cdots + K_0 \cdot 2^0 + K_{-1} \cdot 2^{-1} + \cdots + K_{-m} \cdot 2^{-m} \\ &= \sum_{j=-m}^n K_j \cdot 2^j \end{aligned}$$

其中 K_j 是数码 0 或 1 中的任意一个; m 、 n 为正整数;

2^j 为基数 2 的幂也称为位值。

故数 $S(\text{二})$ 也是数码与位值相乘的总和。

计算机为什么要用二进制表示, 它有哪些优点?

二进制有如下一些优点

(1) 可以用任何只具有两个不同稳定状态的元件来标记二进制数的每一位, 其中的一种状态表示数字“0”, 另一种状态表示数字“1”。制造具有两个稳定状态的元件要比制造多个稳定状态的元件容易得多。例如氖灯的亮和灭, 继电器的闭合和断开, 晶体管的截止和导通等都可以用来表示二进制的“0”和“1”, 也即二进制数的任一位都可用上述的两种状态之一来表示。因此, 二进制在电子技术上是比较容易实现的。

(2) 二进制的算术运算比较简单

例如

加法	$\begin{array}{r} 0 \\ + 0 \\ \hline 0 \end{array}$	$\begin{array}{r} 1 \\ + 0 \\ \hline 1 \end{array}$	$\begin{array}{r} 0 \\ + 1 \\ \hline 1 \end{array}$	$\begin{array}{r} 1 \\ + 1 \\ \hline 10 \end{array}$
乘法	$\begin{array}{r} 0 \\ \times 0 \\ \hline 0 \end{array}$	$\begin{array}{r} 1 \\ \times 0 \\ \hline 0 \end{array}$	$\begin{array}{r} 0 \\ \times 1 \\ \hline 0 \end{array}$	$\begin{array}{r} 1 \\ \times 1 \\ \hline 1 \end{array}$

二进制运算的计数原则是“逢二进一, 借一当二”。例如

$$\begin{array}{r} 101.01 \\ + 110.11 \\ \hline 1100.00 \end{array} \qquad 101.01(\text{二}) + 110.11(\text{二}) = 1100(\text{二})$$

相当于十进制数 $5\frac{1}{4} + 6\frac{3}{4} = 12$

$$\begin{array}{r} 1100 \\ - 110.11 \\ \hline 101.01 \end{array}$$

$$1100_{(2)} - 110.11_{(2)} = 101.01_{(2)}$$

$$\text{相当于十进制数 } 12 - 6\frac{3}{4} = 5\frac{1}{4}$$

$$\text{又例如 } 10.101_{(2)} \times 101_{(2)} = 1101.001_{(2)}$$

$$\begin{array}{r} 10.101 \\ \times 101 \\ \hline 10 \ 101 \\ 000 \ 00 \\ 1010 \ 1 \\ \hline 1101.001 \end{array}$$

$$\text{相当于十进制数 } 2.625 \times 5 = 13.125$$

$$\begin{array}{r} 10.101 \\ 101 \overline{) 1101.001} \\ \underline{101} \\ 11 \ 0 \\ \underline{10 \ 1} \\ 101 \\ \underline{101} \\ 0 \end{array}$$

$$\text{相当于十进制数 } 13.125 \div 5 = 2.625$$

由上可知二进制数的运算比十进制数运算要简单得多，故计算机运算部件的结构可以设计得比较简单。

(3) 可以节省存贮器设备

在说明原因之前，我们先引入“数字一位”的概念。所谓“数字一位”即是为表示在一定范围内的任意数所需的元件数目和元件稳定状态数目的乘积。例如用十进制表示0~999的整数需要三位，每位可能出现十种状态（0~9数码之一），这时的“数字一位”为 $3 \times 10 = 30$ 。用二进制表示上面的数需要十位，每位只可能有两种状态，（0，1数码之一），“数字一位”为 $10 \times 2 = 20$ 。

我们来作一般性的研究。设要表示从0到 N 的整数，进位制的基数为 R ，需要的位数为 n ，求使“数字一位” $x = R \cdot n(R)$ 为最小值时的 R 。

显然我们有

$$N = R^n - 1, \quad R^n = N + 1 = M$$

$$\ln R^n = \ln M \quad \text{即 } n = \frac{\ln M}{\ln R}$$

$$\therefore x(R) = \frac{R \ln M}{\ln R}$$

为求 x 的最小值，求 $\frac{dx}{dR}$ ，并令它为0则得