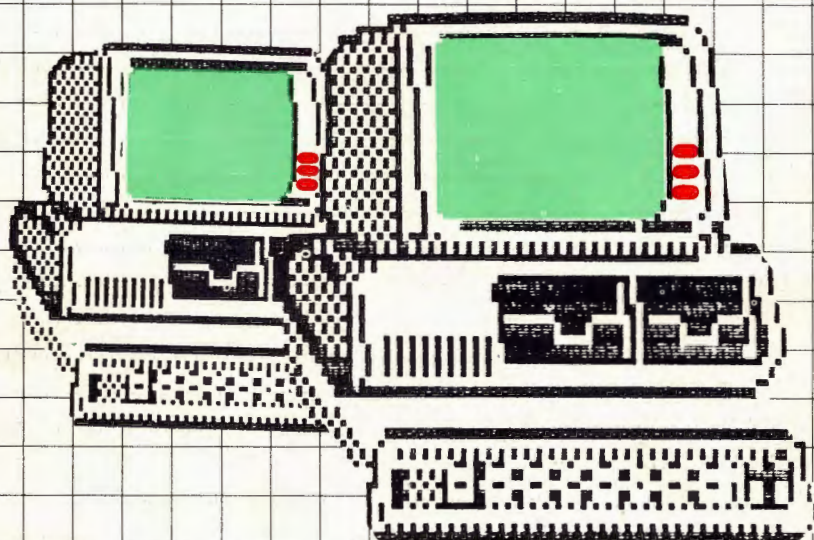


387
ISSN 1001-3695

计算机应用研究

APPLICATION RESEARCH OF COMPUTERS



1989 4

《计算机应用研究》杂志社

《计算机应用研究》杂志办刊单位

四川省电子计算机应用研究中心

贵州省科学技术电子计算机中心

安徽省计算中心

新疆电子计算中心

吉林省计算中心

青海省测试计算中心

甘肃省计算中心

四川省电子学会

《计算机应用研究》杂志编辑委员会

主任委员：顾昌贤

副主任委员：李泽民

委员：余凯 张国栋 贾洪钧

曾光初 王小华 朱景生

《计算机应用研究》杂志社董事会

董事长：周赛渝

董事：唐珍 韩圣谦 陆慰椿

秦小竹 龚宇清 黎瑰常

计算机应用研究（双月刊）

（公开发行）

一九八九年

第六卷 第四期（总30期）

主 编：顾昌贤

副主编：李泽民

本期责任编辑：齐墨之

编辑出版：《计算机应用研究》杂志社

印 刷：西南冶金地质印刷厂

出版日期：1989年7月31日

本刊通讯地址：成都市人民南路4段11号附1号

订阅处：全国各地邮局

总发行：成都市邮政局

每册定价：1.30元

本刊邮发代号：62—68

国内统一刊号：CN51—1196

广告经营许可证川蓉工商广字005号

目 录

软 件 篇

绘制电路图的一种专用语言.....	徐松君 龙克勇 (1)
PARADOX的QBE查询语言.....	李佐卿 (6)
建立IBM (0520) BASIC语言的超 64K二维数组.....	刘大雄 (12)
允许在输入数据文件中加注的输入子程序及其相容性、容错性.....	王孝纯 (14)
同音字替换法修改CCDOS拼音码错误.....	郭继展 (16)
DOS 3.3 空区利用集锦.....	邵爱楠 (18)
复合表达式的特殊功能与应用.....	陈庆祥 (20)
多分支判断的解除.....	沈友敏 (22)
计算任意形状图形面积的布尔运算法.....	孟庆春 (24)
介绍一种新的裁剪算法.....	刘卡林 孔庆复 (27)
多功能打印曲线程序.....	朱龙根 (30)
dBASE III PLUS (FOXBASE) 绘图功能.....	邓京明 (34)
用BASIC程序驱动打印机进行屏幕拷贝.....	魏 旭 (35)
利用BASIC产生屏幕美术字的编程技巧.....	瞿明德 (36)
C-DBASE-III下的屏幕装饰.....	瞿明德 (37)
决策表用于MIS开发的初探.....	张剑平 (38)
一个实用的高精度实时数据采集系统.....	李华贵等 (41)
Novell 网络上的生产文件管理系统.....	郑晓风 (45)
西服显示的逼真性探讨.....	赵元平 (47)
微计算机在测定岩石性能中的应用.....	罗 云 (51)

硬 件 篇

利用8279芯片实现多路D/A 转换.....	周鸣争 (58)
-------------------------	------------

系 统 篇

APPLE II微型机外中断源的扩展方法.....	鹿文宁 王砚方 (60)
TP801-A单板机EPROM编程器的简单制作.....	李华贵 (63)

维 修 篇

DEBUG程序在故障诊断中的应用.....	三晓武 (53)
速达微计算机系统板的故障维修.....	刘 洽 (59)

信 息 篇

征订启事等七则信息.....	(63, 67, 68, 目录页)
新颖实用的医院会计核算自动化软件.....	(封3)

新颖实用的医院会计核算自动化软件

——医院会计核算管理系统介绍

一种新颖实用的医院会计核算自动化软件——医院会计核算管理系统，最近由河北医学院第四医院、河北省卫生厅计财处、河北省医院、河北省结核病医院等单位共同研制成功，并由河北省科委主持通过了技术鉴定。

《医院会计核算管理系统》是一个拥有初始建帐、制单、记帐、月末结帐、汇总凭证、查帐、银行对帐、负数更正、报表处理、帐簿打印等会计作业功能的医院会计核算自动化软件，具有使用方便、自动化程度高的特色，对于减轻财会人员劳动强度、提高会计核算速度和准确程度，实现会计核算规范化有着积极的作用。

参加鉴定会的专家教授及有关用户一致认为：

一、该系统设计合理、构思新颖、结构严密，能够完成医院会计核算的编制记帐凭证、审核修改凭证、记帐、查询、银行对帐、编制会计报表、打印帐簿等日常工作。实现了对医院会计核算工作的自动化管理，不仅免去了会计人员大量繁杂的手工转录，而且保证了财会工作的准确性、安全性和保密性。

二、该系统将医院会计核算业务按卫生部、财政部1987年制订颁发的《医院财务管理办法》和《医院会计制度》融汇于系统程序中，既可促进医院财务管理与会计核算的改革，完善医院经营管理，又完全符合财会工作人员的工作程序和习惯，易于被财会人员所采用。

三、在施行新的《办法》、《制度》后，财会人员的工作量增加近一倍。就河北省情况而言，按1988年2月底统计，缺编40%。由于该系统使医院会计核算规范化、现代化和自动化，可以减轻财会人员的劳动强度，弥补会计人员缺编的困难。

四、该软件在1988年9月已由卫生部组织的十二个省市专家教授对系统进行了评测，同年12月份又举办了一期全国性培训班，进一步推及15个省市自治区，说明该系统有很好的推广前景，具有明显的社会效益和经济效益。

综上所述，由于该系统具有上述特点，与国内同类软件相比，在整体设计、编程技巧上都具有明显的先进性、实用性和科学性，技术水平处于国内领先地位。

需要此软件者，请同石家庄市合作路6号河北省卫生厅计财处张翼宗同志联系（电话：743563）。

亦可同石家庄市健康路河北医学院第4医院孙春秋同志联系（电话：633946）。

APPLICATION RESEARCH OF COMPUTERS

1989 Vol.6 No.4 (Total 30)

CONTENTS

SOFTWARE

- A Special Language for Drawing Circuits..... Xu Songjun Long Keyong (1)
QBE Inquire Language of PARADOX..... Li Zuoqing (6)
Establishing Two-dimensional Array Beyond 64K of IBM (0520) BASIC
Language Liu Daxiong (12)
Permitting Input Subroutine of Comment and It's Compatibility & Fault-
tolerance in the Input Data File..... Wang Xiaoxian (14)
Modifying Errors of CCDOS Phonetic Code with the Method of
Homophone Replacement Guo Jizhan (16)
Outstanding Examples of DOS 3.3 Space Utilization..... Shao Aibo (18)
The Specific Function and Application of Compound Expression
..... Chen Qingxiang (20)
Undoing of Multi-branch Judgment Shen Youmin (22)
Boolean Operation for Calculating Area of Arbitrary Form of
Figure Meng Qingchun (24)
Introducing a New Algorithm for Cutting Out.....Liu Qialin Kong Qingfu (27)
Teh Program of Multi-function Printing Curves.....Zhu Longgen (30)
Drawing Function of dBASE PLUS (Fox BASE)Deng Jingming (34)
Conducting Copy of Screen with BASIC Program Driving Printer
..... Wei Xu (35)
Editing Skill of Producing Artistic Calligraphy on Screen with
BASIC.....Zhai Mingde (36)
Adorning Screen of C-DBASE-III.....Zhai Mingde (37)
Initially Probing into Developing MIS with Decision Table
..... Zhang Jianping (38)
An Utility System of High Precision Real-time Data Collection
.....Li Huagui and others (41)
Producing File Management System on Novell Network Zheng Xiaofeng (45)
Probing into Fidelity of Displaying Western-style Clothes
..... Zhao Yuanping (47)
Microcomputer Application in Measuring Rock Property Luo Yun (51)

HARDWARE

- Achieving Multiple D/A Commutation by 8279 Chip..... Zhou Mingzheng (53)

SYSTEM

- Expansion Method of External Interrupt Source on APPLE II
Microcomputer..... Pang Wenning Wang Yanfang (60)
Simply Making EPROM Programmer on TP801-A Single Board Computer
..... Li Huagui (63)

MAINTENANCE

- The Application of DEBUG Program in Fault Diagnosed.....Wang Xiaowu (53)
Breakdown Maintenance of the System Panel on SUPETECH Microcomputer
..... Liu Qia (59)

INFORMATION

欢迎使用“造字及修改字库”软件

IBM-PC/XT及其兼容计算机配备 24×24 点阵后,打印出的汉字比较美观。但是因为国标码中没有象上标或下标等汉字字模,在实际应用中受到一定限制。要增加汉字字模,就得有造字和修改字库的软件。由于计算机采用了三种字库用于显示和打印,这给修改字库也带来了许多麻烦。这里介绍一个用BASIC编写的造字及修改字库的软件,该软件功能较强,可在修改 24×24 点阵的同时,自动形成 24×48 点阵和 16×16 点阵字库,使得显示和打印字库一次形成。

该软件采用菜单方式,所有操作均有提示,易于使用。汉字的输入方法可用操作系统所具备的所有输入方法。主菜单有四个功能:

一、修改字库:指修改 24×24 点阵汉字,修改完成后,自动形成 24×48 点阵,并在形成 16×16 点阵后转到功能四进一步修改。

二、造标号:该功能能自动形成上、下标号,并可转功能一,进一步完善。

三、合成字:该功能能将两个汉字合成为一个汉字。按上下重叠合成。

四、修改 16×16 字库:该功能可直接修改用于显示的 16×16 点阵,同时也能修改因 24×24 点阵压缩造成的 16×16 点阵变形。

该软件在BASIC, EXE系统下运行。此程序造出的程序框图表符和其它专业字符,适合用于文字及书刊编辑,有很好的效果。需此程序者,可同作者或与本刊联系。

(后勤工程学院自动化教研室 张昭)

APPLICATION RESEARCH OF COMPUTERS

(Publicly Publishing) (Bimonthly)

May 1989 Volume 6 No.4 (Total 30)

Chief Editor: Gu Chang Xian

Vice-Chief Editor: Li Ze min

Responsibility Editor for this Issue: Qi Mo zhi

Editor/Publisher: Editorial Department of this Periodical

Address: 11-1 Four section Ren Ming Nan Avenue Chengdu

Press, South West Metallurgical Geology Printing House

of Metallurgical Industry Ministry

Subscribe Address: The Post of the All Localities of the Country

Central Dispatching in the Inside of Country: The Post of Chengdu

Unified Number in the Inside of the Country: CN51-1196

Number of the Post Publishing: 62-68

绘制电路图的一种专用语言

成都科技大学电力系 徐松君 龙克勇

摘要: 本文提出一种绘制电路图的通用方法。这一方法是在 BASICA 的支持下设计的一种专用语言。它可以在 IBMPC, PC/XT 及其兼容机上运行。这一方法的优点是节省编程时间, 节省存储空间。

一、电路图形的特点

计算机在电原理这一领域的应用前景是宽广的。如计算机辅助分析, 计算机辅助教学, 建立试题库等。但无论那种情况, 都会遇到绘制电路图的问题。首先要解决在屏幕上绘制电路图的问题。

IBMPC 或 PC/XT 及其兼容机的 BASICA 语言是解决这一问题的良好工具, 用它的绘图语言编写的程序几乎可以完成所有电路的绘图工作。这一方面是因为 BASICA 有较强的绘图能力; 另一方面是因为电路理论所研究的电路图均是实际电路的模型, 它仅与元件的性质和联接方式有关, 与各元件及电路的尺寸无关。这一特点, 也便于用 BASICA 来完成。

电路图并不像机械零件图那样有严格的尺寸要求。它仅仅是由几个有限元件放在适当位置上的图形。图形可大可小, 线段也可长可短。用 BASICA 来编写在屏幕上绘制某一图形的程序应该说不难的。例如有一电路图(如图1所示)可用以下这段程序, 把该图绘制在屏幕上。

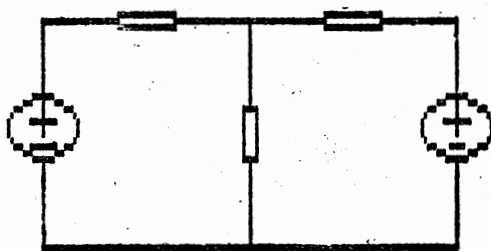


图1

```
1 CLS
2 SCREEN 2,0
3 DRAW"bm-273,26"
10 DRAW"r24u1r18d2l18u1bm+18,+0r48u1r18d2l18u1bm+18,+0r24d13"
20 CIRCLE STEP(0,4),11,1
30 DRAW"bm-0,-1nu2nd2nl4nr4bm+0,+4nl2nr2bm-0,+3d13l66u14l2u8"
35 DRAW"r4d8l2bm+0,-8u14bm-0,+36l66bm-0,-36d13"
40 CIRCLE STEP(0,4),11,1
50 DRAW"bm-0,-1nu2nd2nl4nr4bm-0,+4nl4nr4bm-0,+3d13"
60 END
```

可见绘制一个电路图所用的程序量并不大, 但是如果有成百上千个电路图, 我们也用这种方法一个电路编写一个程序, 总的程序量就非常可观。这种方法还要用应人员了解 BASICA 的程序设计方法; 更为繁琐的是每画一个图, 都必须先由人工去计算图中某些点在屏幕上的坐标, 例如直线的起点坐标, 终点坐标, 圆的圆心坐标等等。由此所花费的时间是很多的。最后还有一个更为严重的问题就是图形的存贮问题。

BASICA 提供了 GET、PUT 两个语句, 把屏幕上某一块矩形面积的内容保存在数组里面。用这种方法保存一帧中分辨屏幕图形约占 16K 字节的内存空间。如果把图形缩小到 1/4 也要占 4K 字节的内存空间, 这并非一个小的数字。

为克服上述缺点, 我们设计了一种专用的绘图语言。它由两部分组成, 一是对电路图的各种元件定义出专用语言的命令字和可选择的参数; 二是利用 BASICA 编制此专用语言每一命令的

绘图子程序。新的语言仍然是解释性的，用户只要按新语言的命令字及可选的参数写出一连续的命令行，该电路图就立即显示在屏幕上。

二、绘制电路图的专用语言的命令字的定义

电路图是由有限的基本元件组成的，它们是电阻、电容、电感、电压源、电流源、运算放大器、回转器、受控源、导线等等。如果把一个电路图按元件进行分解，每一元件用一种或几种符号表示，那么一个电路图就可以转换成由一系列的符号来描述，我们把这些符号定义为这一专用语言的命令字。每一个命令字用三位字符是表征。

第一位字符是表征元件图形的信息代码：

电阻——R
 电感——L
 电容——C
 阻抗——Z
 节点——P
 开关——K
 连接线——W
 波形图——F
 回转器——G
 (光标)转移——T
 晶体三极管——M
 理想电压源——E(e)
 理想电流源——S(s)
 受控电压源——U(u)
 受控电流源——I(i)
 电压参考方向——D
 电流参考方向——A
 运算放大器——O
 非线性元件——N
 晶体二极管——B
 理想变压器——H

第二位字符是表征元件放置情况的信息代码：

1, 4, 7 横向放置；
 2, 5, 8 纵向放置；
 3, 6, 9 左上右下斜向放置；
 A, B 右上左下斜向放置；

这样1, 2位字符一道被定义为专用语言每个命令的关键字。

例如：命令 R1 代表从左到右画出的电阻。

命令 E2 代表从上到下画出的理想电压源。

命令 L3 代表从右下到左上画出的电感。

命令 CA 代表从左下到右上画出的电容。

详见专用语言的命令字一览表(表1)。

第三位字符是每一种命令字的可选项参数，它是表征元件在图中附加的文字信息代码：

0——表示该元件无附加的文字符号；

非0——表示该元件有附加的文字符号，所显示的文字符号由该命令的第一位字符和第三位字符所决定。

第三位字符和数字，取值1~9以表示元件的编号。如命令为R11，表示要画一水平方向的电阻，它的附加文字信息为R1。

在作了这些规定后，对任意的电路图我们都可以很快的写出画该图的命令串。

例：图2所示电路其命令串为：

R10L10R20T03C20
 T30E20W10W10

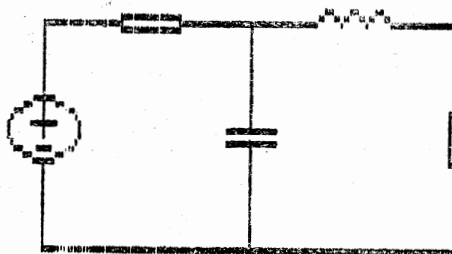


图2

由于图中未加注元件符号及编号，故每个元件的命令字的第三位代码当全部为0。命令为：

R11L12R23T30C24
 T30E21W10W10

绘出图3所示电路图，图中每条支路的元件都加注了符号，并进行了编号。

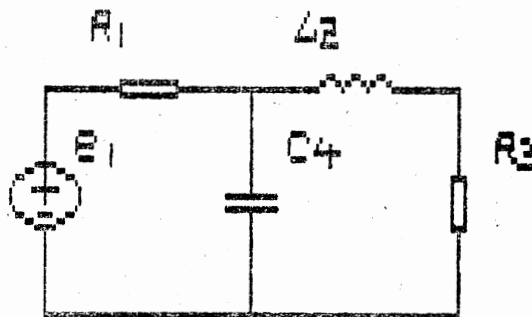


图3

表1 专用语言的命令字一览表

元件	图形及命令字							
电 阻	R1 	R2 	R3 				RA 	
电 感	L1 	L2 	L3 				LA 	
电 容	C1 	C2 	C3 				CA 	
电 压 源	E1 	E2 	E3 	E4 	E5 	E6 	EA 	EB
电 流 源	S1 	S2 	S3 	S4 	S5 	S6 	SA 	SB
受控电压源	U1 	U2 	U3 	U4 	U5 	U6 	UA 	UB
受控电流源	I1 	I2 	I3 	I4 	I5 	I6 	IA 	IB
开 关	K1 	K2 	K3 				KA 	
线 段	W1 	W2 	W3 	W4 	W5 	W6 	WA 	WB
转 移	T1 	T2 	T3 	T4 	T5 	T6 	TA 	TB
箭 头	A1 	A2 	A3 	A4 	A5 	A6 	AA 	AB
阻 抗	Z1 	Z2 	Z3 				ZA 	
参 考 方 向	D1 	D2 	D3 	D4 	D5 	D6 	DA 	DB

续表 1

二 极 管	B1	B2	B3	B4	B5	B6	BA	BB
电 表	X1	X2	X3	X4	X5	X	XA	XB
波 形	F1	F2	F3		F5	F6		X6
	F7	F8					FA	FB
三 极 管	Q1	Q2						
非线性符号	N1	N2	N3				NA	
运算放大器	O0							
回 转 器	G0							
理想变压器	H0							
节 点	P0							

如果还需要对节点编号, 其命令为:

R11P1L12R22T30C24

P02T30E21W10W10

立即绘出图4所示图形。

又如图1所示电路其命令串为 R10R10E20T30R20T30E20W10W10。如果把这些图形的命令的串用BASICA 的文件方式存入磁盘, 它所占令空间是相当少的。图1仅用了27个字符, 图5所示电路也仅用了 57个字符 (d11R11P01L12P02C13S21a21T33u21R32I21T33e21L011P03C14

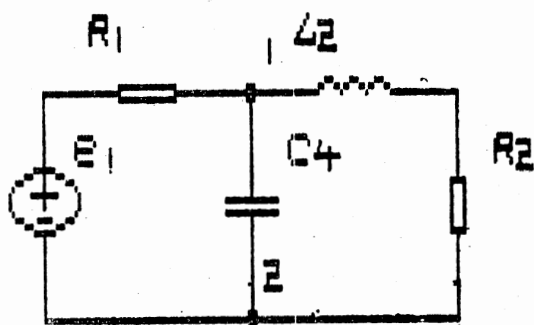


图4

P04R13), 它与前面所提到的BASICA的GET语言相比, 内存空间大为节省。

三、绘图软件的设计

绘图软件由两部分组成, 一是命令串的分解和解释; 二是画图子程序。

1. 命令串的分解和解释可以用BASICA的字符串处理语句和函数来实现。

例如:

```

10 INPUT "input graphics parameter";A$
20 INPUT "input graphics parameter";B$
30 C$=A$+B$
40 N=LEN(C$)
50 FOR I=1 TO N STEP 3
60 D$=MID$(C$,I,3)
70 E$=MID$(D$,1,1)
100 IF E$="R" OR E$="r" GOTO 300
110 IF E$="L" OR E$="l" GOTO 600
120 IF E$="C" OR E$="c" GOTO 900
130 IF E$="W" OR E$="w" GOTO 1200
140 IF E$="P" OR E$="p" GOTO 1500
150 IF E$="T" OR E$="t" GOTO 1800
160 IF E$="E" OR E$="e" GOTO 2100
170 IF E$="S" OR E$="s" GOTO 2400
180 IF E$="U" OR E$="u" GOTO 2700
190 IF E$="I" OR E$="i" GOTO 3000
200 IF E$="K" OR E$="k" GOTO 3300
210 IF E$="A" OR E$="a" GOTO 3600
220 IF E$="D" OR E$="d" GOTO 3900
230 IF E$="Z" OR E$="z" GOTO 4200
231 IF E$="b" OR E$="B" GOTO 4500
232 IF E$="f" OR E$="F" GOTO 4800
233 IF E$="o" OR E$="O" GOTO 5100
234 IF E$="g" OR E$="G" GOTO 5400
235 IF E$="n" OR E$="N" GOTO 5700
236 IF E$="h" OR E$="H" GOTO 6000
237 IF E$="x" OR E$="X" GOTO 6300
240 PRINT "sorry your parameter is mistake!"
250 GOTO 275
260 NEXT I
270 GOTO 285
275 INPUT "do your want to input again!(Y/N)";G$
280 IF G$="Y" OR G$="y" GOTO 30
285 END

```

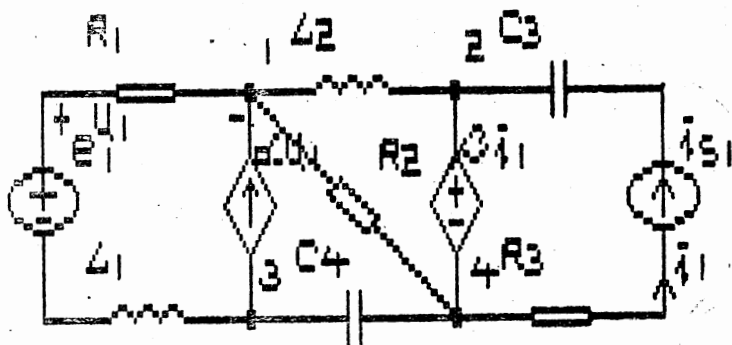


图5

第10、20号语句是考虑到电路的图形复杂时,其命令串可能较长,为此用两个输入语句,可以送两行命令串。这样当简单图形命令串少时也应分两次送入。50~237行语句为对命令串的分解和解释。将命令串按三个字符一组进行分解,然后对每一组第一个字符进行解释,以便转向相应的绘图子程序。

2. 绘图子程序

绘图子程序,是绘制上述各种电路元件的绘图程序。

例如:

```
300 J=1:F$=MID$(D$,2,1)
310 IF F$="1" GOTO 360
320 IF F$="2" GOTO 380
330 IF F$="3" GOTO 400
340 IF F$="A" OR F$="a" GOTO 440
350 GOTO 240
360 DRAW "r24u1r18d2l18u1bm+18,+0r24"
370 GOTO 8000
```

这是一个绘制水平方向的一个电阻元件的程序。实际使用的子程序是很多的,加上还要标注元件的各种文字符号,所以程序量也还是较大的。

四、利用此专用绘图语言绘制电路图的程序设计方法

对任何电路可按任意方向遍历该图的所有支路。在遍历的路径上,每经过一条支路,就写出

画该支路的元件的绘图命令。所有这些命令串在一起即组成了画该电路图的专用绘图程序。在遍历各支路时,可以重复遍历某一支路,也可以用转移命令(只移动不画图),以确保任何电路中的所有支路都会被遍历一次。如前面所介绍的图44为画该电路的程序设计步骤是:

1. 先画 R_1 , 它的命令为 $R11$ 。
2. 画 L_2 , 它的命令为 $L12$ 。
3. 画 R_3 , 它的命令为 $R23$ 。
4. 为画 C_4 , 先转移到 R_1 、 L_2 、 C_4 的交点处, 命令为 $T30$ 。
5. 画 C_4 , 命令为 $C24$ 。
6. 为画 e_1 , 先转移到 e_1 同 R_1 两元件相接处, 命令为 $T30$ 。

7. 画 e_1 , 命令为 $E21$ 。

8. 最后画下面的两段直线,

命令为 $W10W10$ 。

由此,组成了画电路图3的程序。

五、小结

用这一新的方法,可以用保存这样的命令串取代保存图形的点阵,可以大量的节省存储空间。当需要显示某一电路图时,只要执行一次该绘图程序,电路图就会很快的被画在屏幕上。这一方法很容易掌握。在电路的计算机辅助分析和辅助教学中,无疑是一个好的工具。

PARADOX的QBE查询语言

电子科技大学微型机研究所 李佐卿

PARADOX是美国Ansa Software公司研制的新一代关系数据库管理系统,能在IBM PC、XT、AT及其兼容机上运行。目前,它在国内还鲜为人知,而在国外已深受广大用户的欢迎和推崇。它的2.0版本曾被评为1987年年度美国微机上(在MS-DOS支持下)运行的最佳《数据库系统》软件产品。

PARADOX引人注目的特色之一,是它采用了QBE(Query-By-Example)示例查询语言。这种语言最初由IBM公司Yorktown高级研究实验室的M.M.Zloof于1975年提出,并于1978年在IBM的大型机下研制成功。1985年上述的Ansa Software公司在IBM的微型机上实现了QBE。近年来,有人称QBE是一种通用的第四代语言,它在IBM第四代语言工具中的地位与重要

性可以用图1表示。

QBE 查询语言与众不同，是因为它也是一种图形语言，具有二维语法。而大多数传统的计算机语言都具有线性语法。因此，“非常直观”是它的独特风格；“用户之所见即用户之所得”，它能：

- 帮助用户建立查询
- 充分利用全屏幕这一优势
- 作为导航的信号标志
- 提供帮助提示信息

使用时，用户只要指出“做什么”，系统就自动地解决“如何做”的问题。具体讲，用户只要在查询表格上填写查询要求，就能对数据进行操纵操作。操作的结果也以表格形式出现。这不仅对于非专业程序员有很强的吸引力，而且对于许多愿意用视觉或图形方式来思考问题的用户同样具有很强的吸引力。正如 Thomas 等人曾作过实验表明的结果那样：QBE 的用户性能最佳，SQL 次之。

QBE 的用户性能最佳，SQL 次之。

PARADOX的查询方法

先考虑这样一个模糊查询：在 EMPLOYEE（雇员）表中找出 Name（姓名）好像是 Mike Hansen 的雇员记录。

系统先显示查询表格（见图2）：

EMPLOYEE	Id #	Position	Name	Area Code

图2

用户在表格上填写查询要求（见图3）

EMPLOYEE	Id #	Position	Name	Area Code
			like Mike Hansen	

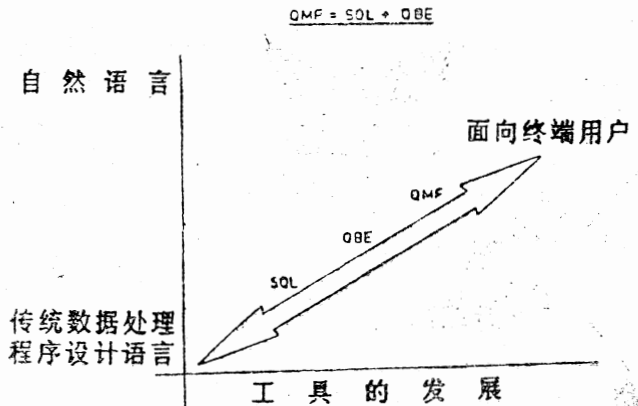
图3

按F2执行键后，系统显示的查询结果如图4所示。

ANSWER	Id #	Position	Name
1	893-47-6321	Salesman	Mike D. Hansen

图4

下面我们比较深入地来讨论查询表格、查询要求和查询结果。含有查询要求的查询表格构成了完整的PARADOX的QBE查询语句。



QMF = query management facility;

查询管理工具

SQL = structured query language;

结构化查询语言

QBE = query by example.

示例查询语言

图1 IBM第四代语言工具发展的里程碑

1. 查询表格

PARADOX的查询表格 (Query Form) 是QBE语句的框架, 以帮助用户建立查询。对应于数据库里的每一个表均有一个查询表格, 根据用户指定的表名, 系统就在屏幕上显示出相应的查询表格。它包含该表所有字段的结构框架, 但不含记录。因此, 它也被称为空白的查询表格, 让用户用来填写对数据进行操纵操作的各种要求。对于多表连接查询就要使用多个查询表格。

在图2中, EMPLOYEE是表名, Id*, Position, Name等均是字段名。它和最初的 QBE形式 [3] (见图5)相比, 又推进了一大步。PARADOX 把表名和字段名放在表格线之间, 而不是表格

EMPLOYEE	Id*	Position	Name	Area Code

图5

框内。这样, 既充分利用了屏幕的空间, 又明显地和用户输入的查询要求区分开来。而表格的字段宽度, 无须用户先用指定的功能键把它加宽, 系统会根据查询条件的长度自动加宽。

2. 查询要求

查询要求包括查询的目标与查询的约束条件 (简称为查询条件, 下同)。

查询目标就是指定查询结果中要出现的字段。这样的字段也称为查询字段或检查字段。为了在查询表格上便于标记, PARADOX设有专用的检查键, 即 F6。操作时, 先将光标移到相应字段内, 一按F6, 其标记符 (V) 以反向显示方式出现在该字段的最左端。此时, 系统按查询条件所得的查询结

表1

类 别	算 符	意 义
保 留 字	V	表示查询结果中要显示的字段
	V+	同上, 且包含重复元素
	Calc	计算新字段
	Insert	用指定的值插入新记录
	Delete	从表中删去所选的记录
	Change to	修改所选记录的字段
	Find	找出表中所选的记录
算术算符	+	加, 或者并接
	-	减
	*	乘
	/	除
	()	括号
范围算符	=	等于 (使用时, 可以省略)
	>	大于
	<	小于
	>=	大于或等于
	<=	小于或等于
字符串算符	...	任意字符串
特殊算符	@	单一字符串
	Like	类似
	Not	不匹配
	Blank	无值
	Today	当天日期
分组算符 (即库函数)	Average	求平均值
	Count	计数
	Max	求最大值
	Min	求最小值
	Sum	求和

果中自动消除冗余重复。若需重复记录的话,则使用检查+键,即〔Alt〕·F6,其相应的标记符为√+。由于上述两个功能键是按计数方式工作的,所以再按一次就能消除标记。若需要查阅全部字段的信息,则有一种简捷方法:将光标置于表名下,按F6或〔Alt〕·F6即可。

查询条件也称为查询标准。PARADOX提供了一套算符的保留字(见表1)来满足各种条件的表达。它除了一般表示具体的数据或指定某一范围外,还能生成新的字段或表达模糊查询(即非确切的匹配,如“like”、“...”、“@”等算符)等,这在某些场合下是很有用的。系统设有专用的示例键,即F5,用来输入查询示例(也称为示例元素,它在屏幕上以反向显示方式出现),以标识一个字段或指定几个表之间的相同字段。示例元素的具体值,包括数据类型或长度等,均不影响查询的要求。

对于逻辑上相“与”的查询条件,它们表示在表格的同一行内。若同一字段内出现“与”的条件,“,”分隔。

对于逻辑上相“或”的查询条件,它们表示在查询表格的不同行内。“或”的个数就等于使用的行数。要注数的一点是:在各行内出现的查询目标应该一致。否则,系统会在信息窗口区显示错误信息。

当一个查询涉及多个表时,就要在多个查询表格上填上相应的条件。

3. 查询结果

系统把满足查询要求的结果,自动地按第一个检查字段进行排序,然后显示在ANSWER表中。ANSWER表象图4所示的形式。若要查阅所有字段,可以用光标左右移动键来实现。它也可以用图表转换功能键,即F7来改变。那时它以数据输入的形式出现,即显示一个记录的所有字段(见图6)。

ANSWER表是一个临时表。若需要保留时,则用PARADOX的相应命令对它进行更名,即能作为永久表存贮。

ANSWER		#	1
Id #:	123-45-6789		
Position:	President		
Name:	James E. Smith		
Area Code:	205		
Phone #:	573-6210		
Birth Date:	2/10/45		
Dependents:			
Salary:	24,000.00		
Date Employed:	5/01/80		

图6

若干实例

为节省篇幅起见,下面所有实例均不显示查询结果,而只给出查询语句。由于这些查询语句是在屏幕上实际显示的缩影,所以,有时需要检查的字段不一定全部出现在图上。

例1:简单查询

找出职务是President的雇员记录。

EMPLOYEE	Id	Position	Name	Area Code
		President		

例2: 用 “...” 算符的查询

找出所有姓为Taylor的雇员姓名及其身份证明号码。

EMPLOYEE	Id	Position	Name	Area Code
			.. Taylor	

例3: 逻辑上相“或”条件的查询

找出职务为President或VP或Manager的雇员姓名、身分证号码与职务。

EMPLOYEE	Id	Position	Name	Area Code
		President		
		VP		
		Manager		

这里, Position 字段上有三个相“或”的查询条件, 分别写在三行上。图上三行的查询目标是一致的。而查询结果与各行出现的先后顺序无关。

例4: 逻辑上相“与”条件的查询

找出在80年至82年期间被雇用的人员姓名及雇用日期。

Dependents	Salary	Date Employed
		≥1/1/80, <=12/31/82

这里, 标有✓的检查字段Name已卷出屏幕。

例5: 生成新字段

由Salary (年工资) 字段产生一个Salary/12 (月工资) 字段。

Dependents	Salary	Date Employed
	pay, calc pay/12	

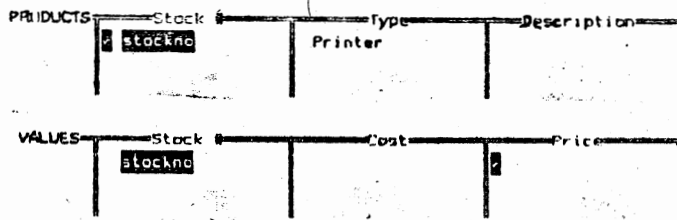
或

Dependents	Salary	Date Employed
	99999, calc 99999/12	

上面图示的两条语句是等效的。Pay和99999是示例元素, 它们标识的都是Salary 字段, 而不影响算符Calc的功能。

例6: 涉及多表的查询

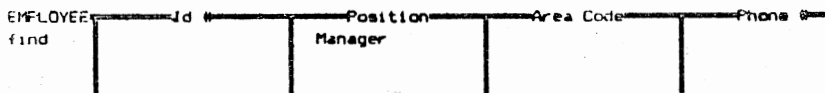
查出所有打印机产品的售价清单。



这里涉及二个表,即PRODUCTS (产品) 和 VALUES (价值) 表,通过 Stock* (库存产品代号) 把它们连接起来。

例7: 定位操作

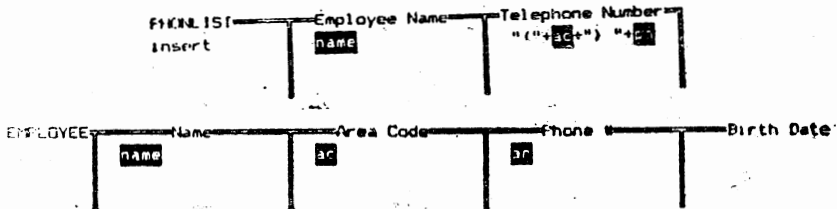
找出职务是Manger的记录



算符Find填在表名下。此时,系统不产生ANSWER表,仅显示含有记录的EMPLOYEE表,而把光标定位在所查找的记录序号处。同时,屏幕上有相应的文字说明。

例8: 插入操作

把EMPLOYEE表中 Name (姓名, Area Code (地区代码)、Phone* (电话号码) 等三个字段的数据,按要求 (主要是后二个字段组合) 插入到PHONLIST表中。

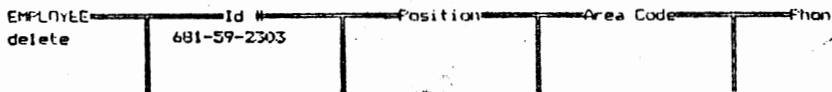


算符Insert填在PHONLIST表名下。此时,系统不产生ANSWER表,除了在PHONLIST表中插入 (或添加) 数据外,还产生一个Inserted表。它包含被插入的内容,以便对插入操作还原。

图中“+”号是起并接作用,它表达了两个字段组合的要求。

例9: 删除操作

把身分证明号码为681—59—2303的雇员记录删除。



算符Delete放在表名下。此时,系统不产生ANSWER表,除了对EMPLOYEE表删除所要求的记录外,还产生一个Deleted表。它含有刚被删除的记录,以便对删除操作还原。

例10: 修改操作

把所有职务是Salesman (售货员) 的年工资增加10%。

建立IBMPC (0520) BASIC语言的超64K二维数组

新疆阿克苏地区科委科技情报所 刘大雄

对于 IBM PC BASIC、PASCAL、FORTRAN语言2.0版本,规定数据段不超过64K。在西文DOS2.0支持下运行解释BASIC,占用内存不大于92K,编译BASIC也如此。对于拥有512K内存的用户,有连续420K未被利用,若在此存放数组,按单精度变量设置数组元素的话, $\sqrt{420 \times 1024 \div 4 \div 2} \approx 230$, 用伴随矩阵法将可解出 230×230 的逆矩阵,而常规下只能解到 80×80 。建立超64K二维数组的方法是可行的,在BASIC主程序中用CALL指令调用机器语言子程序就能实行。

假设内存共有512K,西文DOS及BASIC占用空间不作改动,数组元素按单精度从92K存起,机器语言从510K存起,则调用机器语言的BASIC程序如程序(1)。程序(1)中:C为数

```
10 F=5888+C*F:G=4*G
20 DEF SEG=&H7F80:F=P
30 CALL P(E,F,G,X)
```

程序1

组每一行要占用的字节总数,必须是整形变量;F为段基址,数组的每一行都用不同的段基址;G为数组元素每一列的地址偏移值,每个元素占用4个字节;X为传递数组的中间变量,必须是单精度;F、G必须是整形变量,程序中是用10进制表示,但在机器内部是以16进制贮存调用;E用来表示数值元素是存还是取。

现将机器语言子程序用DEBUG反汇编,以程序(2)表示,并分段说明语句的作用。100~103,保护BP,ES(与DS,SS同值)入栈,将SP给BP;104~10D,取出E值判断数组元素是存还是取,E总是取1或2;10F~121,存数组元素的准备阶段,将BASIC数据段的X变量设为源地址,将数组存放地址作为目的地址;122~132,取数组元素的准备阶段,将数组存放地址作为源地址,将BASIC数据段的X变量作为目的地址;133~138,完成数据传送;139~13F,恢复ES、DS、BP,段间返回,将控制权交BASIC主程序。

Dependents	Salary	Position	Name
	pay, changeto pay*1.10	Salesman	

这里算符Changeto放在年工资字段内。系统不产生ANSWER表,只修改相应的记录的年工资字段内容。同时产生一个Changed表,它保留修改前的数据,以便对修改操作还原。

由以上分析和实例展示,我们可以得出:PARADOX的QBE查询语言,是一种微机上值得推广的第四代语言。

主要参考文献

1. “一九八七年美国最佳微机产品”,《计算机世界》,1988年19期;
2. Dimitris N. Chorafas, “Fourth and Fifth Generation Programming Languages, Vol. I”, McGraw-Hill Book Company, 1987;
3. C. J. Date, “数据库系统导论”卷 I, 中译本, 科学出版社, 1986;
4. George Tsu-der Chou, “Using Paradox”, Que Corporation, 1986;
5. 萨师煊, 王珊, “数据库系统概论”, 高等教育出版社, 1985;

```

-U100,13F
3D1B:0100 55      PUSH    BP
3D1B:0101 8BEC     MOV     BP,SP
3D1B:0103 06      PUSH    ES
3D1B:0104 8B760C   MOV     SI,[BP+0C]
3D1B:0107 8B04     MOV     AX,[SI]
3D1B:0109 2401     AND     AL,01
3D1B:010B 50      PUSH    AX
3D1B:010C 59      POP     CX
3D1B:010D E313     JCXZ    0122
3D1B:010F 8B760A   MOV     SI,[BP+0A]
3D1B:0112 8B04     MOV     AX,[SI]
3D1B:0114 50      PUSH    AX
3D1B:0115 07      POP     ES
3D1B:0116 8B7608   MOV     SI,[BP+08]
3D1B:0119 8B04     MOV     AX,[SI]
3D1B:011B 50      PUSH    AX
3D1B:011C 5F      POP     DI
3D1B:011D 8B7606   MOV     SI,[BP+06]
3D1B:0120 EB11     JMP     0133
3D1B:0122 8B7E06   MOV     DI,[BP+06]
3D1B:0125 8B760A   MOV     SI,[BP+0A]
3D1B:0128 8B04     MOV     AX,[SI]
3D1B:012A 50      PUSH    AX
3D1B:012B 8B7608   MOV     SI,[BP+08]
3D1B:012E 8B04     MOV     AX,[SI]
3D1B:0130 50      PUSH    AX
3D1B:0131 5E      POP     SI
3D1B:0132 1F      POP     DS
3D1B:0133 B90400   MOV     CX,0004
3D1B:0136 FC      CLD
3D1B:0137 F3      REPZ
3D1B:0138 A4      MOVSB
3D1B:0139 07      POP     ES
3D1B:013A 06      PUSH    ES
3D1B:013B 1F      POP     DS
3D1B:013C 5D      POP     BP
3D1B:013D CA0800   RETF     0008

```

程序2

举例说明如何建立和调用数组。程序(3)是一个给二维数组置数、读数的原程序。此程序由于数组设置过大,必然溢出出错,只有存到BASIC64K用户程序之外。程序(4)是针对程序(3)的改编,为方便阅读和理解,两个程序的语言标号尽量统一。对比程序(3),程序(4)多了两个GOSUB子程序。400句开始的是为了建立机器语言子程序,500句开始的是为了调用

```

100 A=200:B=300
110 DIM A(A,B)
120 FOR I=1 TO A
130 FOR J=1 TO B
140 INPUT A(I,J)
150 NEXT J:NEXT I
200 FOR I=1 TO A
210 FOR J=1 TO B
220 PRINT A(I,J)
230 NEXT J:NEXT I
240 END

```

程序3

```

100 A=200:B=300:X=0
110 GOSUB 400
120 FOR I=1 TO A
130 FOR J=1 TO B
140 INPUT X
141 E=1:F=I:G=J
142 GOSUB 500
150 NEXT J:NEXT I
200 FOR I=1 TO A
210 FOR J=1 TO B
220 E=2:F=I:G=J
221 GOSUB 500
222 PRINT X,
230 NEXT J:NEXT I
240 END
400 C=INT((4*B+19)/16)
410 DEF SEG=&H7F80
420 FOR I=0 TO 63
430 READ J:POKE I,J
440 NEXT I:RETURN
450 DATA &H55,&H8B,&HEC,&H06
451 DATA &H8B,&H76,&H0C,&H8B
452 DATA &H04,&H24,&H01,&H50
453 DATA &H59,&HE3,&H13,&H8B
454 DATA &H76,&H0A,&H8B,&H04
455 DATA &H50,&H07,&H8B,&H76
456 DATA &H08,&H8B,&H04,&H50
457 DATA &H5F,&H8B,&H76,&H06
458 DATA &HEB,&H11,&H8B,&H7E
459 DATA &H06,&H8B,&H76,&H0A
460 DATA &H8B,&H04,&H50,&H8B
461 DATA &H76,&H08,&H8B,&H04
462 DATA &H50,&H5E,&H1F,&HB9
463 DATA &H04,&H00,&HFC,&HF3
464 DATA &HA4,&H07,&H06,&H1F
465 DATA &H5D,&HCA,&H08,&H00
500 F=5888+C*F:G=4*G
510 CALL P(E,F,G,X)
530 RETURN

```

程序4

机器语言子程序。程序(4)中删掉了DIM语句,对主程序中使用数组的部分作了一些修改,修改方法具有规律性,很容易掌握。

将程序(4)的语句分析一下。400句计算出

允许在输入数据文件中加注的输入 子程序及其相容性、容错性

福建省地震局地震研究所 王孝铨

摘 要

本文介绍一种用True BASIC 语言编写的具有特殊性能的数据文件输入子程序。它不仅可以和INPUT 语句一样正确地输入通常的数据文件,而且可以正确地输入添加了注释的数据文件。应用这个子程序,可以在准备的数据中加注,以增强其可读性和可维护性。

一、问题的由来

应用计算机解决科学计算、绘图等问题时,往往需要准备大量的数据,而这些数据一般都安排在一个数据文件中。我们时常会发现这样的情况:当数据文件准备好而运行程序时,其结果不合理或与原来设想的相不一致,甚至是错误的,固然这里的问题有可能发生在源程序,但更多的是由于数据有误。如果是这样,那么当数据文件中的数据量很大时,查错是比较麻烦的。比方说,准备了地图底图的数据文件,运行绘图程序后所画出的福建省界出现了异样或者短缺了,于是要更改和补充有关数据就要在一系列“面孔如一”的经纬度数值中寻查福建边界数据是在什么位

置。这确似“海底捞针”。鉴于这种情况,假如能象一般源程序一样,在数据文件中也允许对不同的数据加以注释,那么我们对数据文件的阅读、查找和修改就方便多了。

本文就介绍这样一种数值数据文件的输入程序(下面所讲的数据均指数值数据),它可以对通常的数据文件正确地读数,如同语言中的输入语句,还可以对添加了注释的数据文件正确地读数。这些注释仅仅提供用户阅读的方便,输入时它们是被忽略的成分。

二、程序清单

下面是用True BASIC 语言编写的数据文件输入子程序:

```
! input filename$;
! output array z(1)---z(k).
SUB inputdata(filename$,z(),k)
  OPEN #8:name filename$,access input,organization text
  LET k=0
```

数组每一行占用的字节数,为以后设置段服务,所以要除以16并取整;410≈440是以7F8DH(即510K处)为段基址,以0为偏移量,连续设置64个2位16进制机器语言代码,代码按前面反汇编表的左列顺序,段基址与0偏移值必须与510句一致;500≈520前面已介绍过,此处略;100句置X为零是早些定下X的确切贮存地址;140先给X赋值后才能存取;141是为确定段基址,偏移址的参数,E=1表示为存取数组元素;142调用子程序完成存取;220中E=2表示取数组元素;221是调用子程序完成存取;232显示刚取出的数组

元素。程序中所用的指令,请参考有关手册。

经过修改后的BASIC解释程序,由于数据量大,又是段间调用,因此运算速度明显下降,最好进行编译。为了使编译后的程序具有通用性,某些与建立、调用数组有关的变量及数组变量最好先建成顺序文件。

我们用以上方法修改了“投入产出模型”程序和“单纯型解线性规划”程序,有效地解决了数组超64K和数据总存储空间不够的矛盾。

建立双精度、整型变量及一维、多个维或多个数组的方法,请读者自我探索。


```

input filename$:
output array z(1) to z(k).
SUB inputdata(filename$,z(1),k)
OPEN #8:name filename$,access input,organization to
LET k=0
DO
LINE INPUT #8:xx$
LET nn=len(xx$)
LET digit=0
LET chr=0
LET jj=1
FOR j=1 to nn
SELECT CASE xx$(j:j)
CASE "0" to "9","+", "-", "."
LET stz=0
CASE else
LET stz=1
END SELECT
WHEN error in
IF j=nn then
IF stz=1 and digit(>0) then
LET k=k+1
LET yy$=xx$(jj:nn-1)
LET z(k)=val(yy$)
ELSEIF stz=0 then
LET k=k+1
LET yy$=xx$(jj:nn)
LET z(k)=val(yy$)
ELSE
END IF
ELSEIF stz=1 then
LET chr=chr+1
IF chr=1 then
LET k=k+1
LET z(k)=val(xx$(jj:j-1))
LET digit=0
LET jj=j+1
ELSE
END IF
ELSE
LET digit=digit+1
IF digit=1 then
LET jj=j
LET chr=0
ELSE
END IF
END IF
USE
LET k=k-1
END WHEN
NEXT j
LOOP until end #8
CLOSE #8
END SUB

```

三、调用要点

1. 子程序入口和出口

调用本子程序 inputdata 时, 必须给定所要读数的数据文件的文件名, 它替换第一个哑元: filename\$。这是子程序唯一的输入参量。

执行后, 数据文件中各有效的数据依序读入数组 z 中, 并给出所读数据的个数 k。当然, 根据哑实结合的原理, 在主程序中可以用别的实元名来替换数组名 z 和变量名 k。

2. 数据文件的格式

我们不妨把可形成数值的字符: “+”、“-”、“.” 和数字 0~9 以外的其它字符称为非数值字符, 那么, 输入的数据文件的格式要求为:

(1) 数据依序安排为若干行;

(2) 一行内各数据之间均有一个或一个以上的非数值字符相隔开;

(3) 各行的行首和行末, 可以是数据本身, 也可以是非数值字符串;

(4) 可以有全部由非数值字符组成的无数据行。

显而易见, 使用上述这些非数值字符串便可形成给数据文件加以注释的内容。

3. 数据文件名应具后缀.txt或.tru。

四、举例说明

为了具体理解附加有注释的输入数据文件是什么样子以及本子程序的调用方法, 我们就来看看试编的一个基本包含有各种情况的数据文件的典型例子, 并给出调用子程序后所输出的结果。

1. 主程序

```

! filename(main) : io.tru
DIM z(50)
CALL inputdata("a:datafile",z,k)
OPEN #2:name"prn."
FOR j=1 to k
PRINT #2:z(j),
NEXT j
CLOSE #2
END

```

2. 输入的数据文件 (已附加注释)

```

filename:datafile
this is a example:
first : 1, 2, 3, 4, 5, 6
       7, 8, 9,10
second: 11,12,13,14,15
third : 16,17,18 and 19,
       20.5 end

```

3. 输出的读进数组 z 中的数据

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20.5

五、相容性和容错性

对于True BASIC语言中 INPUT 语句所要求的正文文件的数据, 调用上述输入子程序也可以正确地输入。其实, 通常的数据文件就是附有注释的数据文件的一种特殊情况。可见, 本输

同音字替换法修改CCDOS拼音码错误

河北廊坊炮兵导弹学校 郭继展

CCDOS2.0、3.0各级版本是目前IBM—PC及其兼容机上广为使用的中文操作系统,但许多版本的拼音输入码都有一些错误。下面是我们在一级汉字中发现的,共有37个:

汉 字	杯	碑	坊	挂	褂	峻	蓝	浪	钎	抬	藤	腾	炯
区 位 码	1713	1714	2327	2550	2551	3094	3222	3243	3905	4407	4457	4458	3028
错误拼音码	bk	bk	fj	gul	gul	jue	kj	lih	qif	k[i	tk	dg	jy
正确拼音码	bei	bei	fh	gua	gua	jun	lj	lh	qij	tl	tg	tg	jis

	碗	舷	萧	楔	轩	攒	赃	脏	葬	泽	椎	锥	荆
	4567	4747	4784	4808	4889	5260	5263	5264	5265	5283	5521	5522	3003
	wl	xji	xih	xi	xu	j[n	zg	zg	zg	{e	auh	auh	jb
	wj	xij	xik	xie	xuj	zj	zh	zh	zh	ze	au	au	jy

	崩	绷	甬	虽	隋	铁	帖	睡	鞍	揣	酣
	1732	1733	1734	4368	4369	4490	4491	4315	1616	2007	2608
	bf	bf	bf	suj	suj	tik	tik	oui	i	iu	hl
	bg	bg	bg	sui	sui	tie	tie	uui	j	iul	hj

这就是说,当按正确的拼音码输入时,是根本找不到它们的。有的按首字母从头一一查询,可以查到,如“杯”、“峻”,这是因为它们拼音的第一个字母没有错;而另外一些字,如“蓝”、“睡”、按首字母也是永远查不到的,因为它们第一个字母就错了,在查找其他汉字时,有时会发现它们混在里头;至于“泽”字,就是翻遍所有字母也是找不到的,因为它的第一个符号“{”根本就不是字母。

入子程序对于语言中 INPUT 语句来讲是相容的,而且使用起来似更便利。

从上述介绍的数据文件的格式特点可以看出,本子程序还有一定的容错性能。众所周知,通常的数据文件格式要求各数据之间由一个逗号隔开。由于准备这样的数据文件,工作量往往

相当大,难免有时不慎,在数据间出现了连续的两个逗号,或者出现了非逗号的字符等等意外情况,假如使用语言中的 INPUT 语句来读这样的数据文件,就会出错,然而要是调用本文介绍的子程序,那么这些错误就会被忽略过去,而仍能正确地将数据读入。

二级汉字错误更多,我们粗略地检查了一遍,竟有108个,按区位码顺序列出如下:

丐 廿 丞 厓 刂 刳 劖 仃 佗 侏 侑 倬 倨 衰 衰 衰 衰 衰 衰 衰 衰 冥 汕 涯 洗 谄 谶
 谗 谄
 哧 咪 嗯 噤 岫 阍 狷 夥 儋 厖 汙 浚 洄 迳 矧 嬗 緌 緌 緌 枵 桎 槿 鞞 鞞 鞞
 华 阮 肫 膝 颡 桃 稂 稂 痧 袂 蚩 螂 螭 蚌 螭 螭 笄 笄 楷 越 越 越 罪 佳 蚩 蚩 蚩

对于这些错误,或从首字符查起,或借助于区位码表,“大海捞针”。当然,最好还是设法把错误的拼音码纠正过来。

我们知道,CCDOS的拼音码与首尾码共用一张扫描表,这张表在文件CCCC.EXE中,由DOS自举时带入内存。每个表项占4个字节,与字库中的汉字点阵(每个汉字占32个字节)按递增顺序一一对应。错误的拼音码,只是字母错,在扫描表中的位置并没有错乱。所以,修改拼音码的错误,只需在原处修改。CCDOS规定,首尾码、拼音码用小写英文字母a~z键入,它们的ASCII码在97~122之间,如果原原本本表示,要占7个二进制位。实际上,CCDOS将它们均减去96,缩至1~26之间,用5个二进制位就够了。又由于首尾码的第三个字符必然与拼音码的第一个字符相同,所以扫描表中,每个表项只有5个字母(不足时补以“ ”),共占25个二进制位,需用4个字节表示,如图1所示:

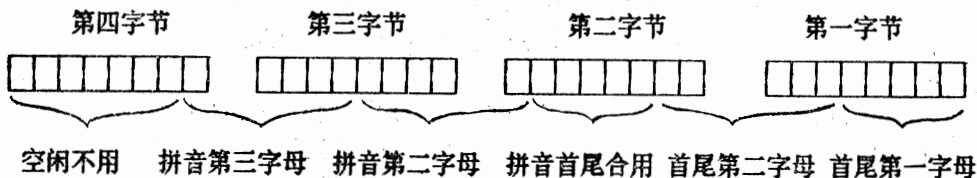


图1

有些报纸、杂志介绍过一些修改方法,要求用户熟悉DEBUG或首尾码,且需对四个字节的32个二进制位进行大量的手工逻辑运算,费时费力易出错。本文介绍一种同音字替换法,简单易行。基本思想是,把磁盘上的CCCC.EXE文件当作一个随机数据文件,拼音码和首尾码扫描表的首地址为1731。这就是说,区位码表中的第一个汉字“啊”的首尾码和拼音码存放在11731~11734字节中,第二个汉字“阿”的存放在11735~11738字节中。对任何一个汉字,根据它的区位码都可计算出它的首尾码和拼音码在扫描表中的位置。我们知道,汉字都是有同音字的,它们的拼音码当然相同。一个汉字的拼音码错了,只要它的某个同音字的拼音码正确,替换一下即可。至于首尾码,不管正确与否,不动它就是了。

对照图1,假定它含有某汉字错误的拼音码,并假定还有一张类似的图,是它的同音字的表项,是正确的。图1第一个字节不动(可以取,但不回存),而类似的图的第三、四字节全部存入图一的第三、四字节,类似图的第二字节取左六位、图1的第二字节取右两位合成一个字节(程序中140行),存入图1的第二字节。当CCDOS重新启动时,拼音码就是正确的了。

程序用BASIC语言编写,采用问答式。前面提到的145个错误(汉字可用区位码输入),已用它成功地改正过来了。程序清单如下:

```
5 REM 拼音码改错
10 CLS:DEFINT D-Q:DEFSTR R-Z:K1=11731
20 OPEN "cccc.exe" AS #1 LEN=1:FIELD #1,1 AS T
30 IF LOF(1)<>44356! THEN PRINT "cccc.exe not !":END:REM 此时应修改K1
40 PRINT:INPUT " 请输入拼音码可能错误的汉字 "; S1
50 IF S1="e" OR S1="E" THEN CLOSE:END
```

```

60 S=S1:GOSUB 170
70 U12=U2:C1=C+1
80 INPUT " 拼音码正确吗 (y/n) "; Y
90 IF Y="y" OR Y="Y" THEN 40
100 INPUT " 输入它的一个同音字"; S2
110 S=S2:GOSUB 170
120 U22=U2:U23=U3:U24=U4
130 INPUT " 拼音码正确吗 (y/n)"; Y: IF Y="n" OR Y="N" THEN 100
140 M2=ASC(U12) AND 3 OR ASC(U22) AND 252:U22=CHR$(M2)
150 RSET T=U22:PUT #1,C1:RSET T=U23:PUT #1:RSET T=U24:PUT #1
160 PRINT TAB(5); "已换好 !!":GOTO 40
170 Q=ASC(LEFT$(S,1))-160:M=ASC(MID$(S,2,1))-160
180 C=((Q-16)*94+M)*4+K1:rem Q 区号, M 位号, C 扫描表指针
190 GET #1,C:U1=T:I1=ASC(T):GET #1:U2=T:I2=ASC(T):GET #1:U3=T:
    I3=ASC(T):GET #1:U4=T:I4=ASC(T)
200 J1=I1 AND 31:rem J1~J5这5个字符从4个字节中分离出来, 值1~26, 或-5
210 J2=((I1 AND 224)/32) OR ((I2 AND 3)*8):IF J2>26 THEN J2=-5
220 J3=(I2 AND 124)/4
230 J4=((I2 AND 128)/128) OR ((I3 AND 15)*2):IF J4>26 THEN J4=-5
240 J5=((I3 AND 240)/16) OR ((I4 AND 1)*32):IF J5>26 THEN J5=-5
250 PRINT TAB(5); S; " 首尾码: "; CHR$(J1+96)+CHR$(J2+96);
260 PRINT TAB(32); "拼音码: "; CHR$(J3+96)+CHR$(J4+96)+CHR$(J5+96);
270 PRINT TAB(50); "区位码: ";
280 IF Q<10 THEN PRINT USING "0#"; Q:ELSE PRINT USING"###"; Q;
290 IF M<10 THEN PRINT USING"0#"; M:ELSE PRINT USING"###"; M
300 RETURN

```

参 考 资 料

- (1) 《CC-BIOS分析》，钱培德著。
- (2) 《计算机应用研究》1988年4期 王导渝、顾永平同志文章。

DOS 3.3 空区利用集锦

郑州市技工学校 邵爱柏

中华学习机DOS3.3的磁盘容量仅为140KB, 如何充分利用这一有限空间, 对中华机的推广应用, 无疑是有裨益的, 笔者亦愿在此“抛砖”, 其意在于“引玉”。下面举一些实例, 说明利用空区写入机器语言程序以增加DOS某些功能的方法。

1. 保护BASIC程序 (隐藏):

```

*B6B3: A2 00 A0 01 AD 01 08 85 06
        AD 02 08 85 07 A1 06 C9 00
        D0 0F B1 03 C9 00 D0 09 A5
        06 85 67 A5 07 85 68 60 A1
        06 85 08 B1 06 85 07 A5 08
        85 06 4C C1 B6

```

键入CALL 46771即可将内存中BASIC程序隐藏起来。

2. 连接两个BASIC程序(合并);

*B6E5: A9 01 85 67 A9 08 85 68 60

调入另一BASIC程序, 键入CALL46821
可将二程序合并。

3. 音乐功能;

*BCDF: A0 00 B1 B8 48 C8 20 98 D9
68 C9 53 D0 03 4C C2 B3 4C
C9 DE

*B3C2: 20 46 E7 86 51 AD 30 C0 88
D0 04 C6 51 F0 08 CA D0 F6
A6 50 4C C7 B3 60

*9E77: DF BC

DOS启动后, 键入&S<音阶值>, <节拍值>
可以发出乐音(其值均 ≤ 255)。〔注〕\$B3C2
一段内容在VTOC中(17轨0扇区)从第\$07字
节处重写一次(用《磁盘医生》)

4. GOTO <变量行号>功能;

*B4AA: 20 BE DE 20 67 DD 20 52 E7
68 68 20 41 D9 4C D2 D7

〔注〕在17轨0扇区从第\$EF字节处开始重
写一次。

键入CALL 46250, N: GOTO N

5. 低分辨状态下打底色功能;

*B49D: A0 27 84 2D A9 00 20 28 F8
88 10 F8 60

〔注〕在17轨0扇区从第\$E2字节处开始重
写一次。

键入COLOR=N: GR: POKE 48,
N+N*16: CALL 46237

6. 读指定的数据语句行中数据功能;

*B3DA: A9 C1 20 C0 DE 20 0C DA
20 1A D6 38 A5 9B E9 01
A4 9C 4C 50 D8

〔注〕在17轨0扇区从第\$1F字节处重写一
次。

键入CALL 46042 TO <DATA语句行
号>; READ<变量>

7. 列目录功能的改造;

①显示文件名中控制字符(反相)

*B6EE: C9 A0 90 04 20 ED FD 60 69
80 4C F2 B6

*AE1C: EE B6

②列长目录按回车键时退出列目录;

*BCF3: 20 0C FD C9 8D D0 03 4C
2C AE 4C 3C AE

*AE39: 4C F3 BC

③列目录先清屏功能;

*ADAC: 58 FC

④列目录时减少显示的文件数目;

*ADA4: 11

*AE3D: 10

8. 引导DOS时具有封锁CTRL—C和 CTRL—RESET功能;

*BA69: 85 FF AD 00 C0 C9 83 F0 05
A5 FF 4C 81 9E A9 00 8D 10
C0 4C DF A4

*9D02: 69 BA

*9E7D: DE A4 DF A4

*9E37: 00

启动DOS时, 若按住CTRL—C不放, 系
统不会中断, 放开时则运行内存中BASIC程序;
按CTRL—RESET时则重新启动磁盘(自举),
可以保护用户程序。

9. 直接删除加锁文件;

*B78D: 20 75 A2 4C 63 A2

*9D28: 8C B7

10. 磁盘加密功能;

*BA7F: 86 2B 85 2A C9 1E 90 02
E6 2A AD 78 04 C9 1E 90
03 EE 78 04 A5 2A 60

*B9A0: 20 7F BA EA

先将DOS3.3引入内存, 按上述方法修改,
格式化一张空盘(再写入有关内容), 则该盘就
具有上述新功能。这些已在本人改编的《中华学
习系列机》系统主盘(CEC—DOS)上实现,
同仁不妨一试。



人无远虑
必有近忧

包世雄 篆刻

复合表达式的特殊功能与应用

南充市第一中学 陈庆祥

通常所谓的表达式有三种,即算术表达式、关系表达式和逻辑表达式。算术表达式用于算术运算,其结果为数值型值,关系表达式和逻辑表达式用于逻辑判断,它们的结果均为逻辑值1或0(1表示为真,0表示为假)。一般说来,后两种表达式仅出现在IF/THEN等控制程序分支的语句中。如果你稍加留意,就不难觉察到:表示真假的1和0依然也是数值型值,那么它们同样可以参与算术运算。具体地讲,在一些场合,我们完全可以把关系表达式或逻辑表达式融于一般的算术表达式之中,从而构成一种复合表达式。这种复合表达式将同时具有算术运算和逻辑判断的双重功能,它可以脱离分支语句而单独发挥其作用。在编程中使用复合表达式能大大减少程序流向的转移、简化推理和计算过程,不仅可以提高其处理效率,而且能使程序变得精炼、清晰,有利于模块化与结构化。

例1. 销售某种化工原料的批发价有三种:凡一次购买50千克及以下者,每千克8元;一次购买50千克以上但不足100千克者,每千克7.5元;购买100千克及以上者,每千克7元。试编写计算销售金额的程序:

一、用一般方法编程如下:

```
10 INPUT W
20 IF W < =50 THEN 60
30 IF W < 100 THEN 80
40 D=7
50 GOTO 90
60 D=8
70 GOTO 90
80 D=7.5
90 PRINT "W=" ; W, "YUN FEI=" ;
  W*D
109 GOTO 10
```

二、改用复合表达式编程:

```
10 INPUT W
20 D=8*(W<+50)+7.5*(W>50 AND
```

```
W<100)+7*(W>=100)
30 PRINT "W=" ; W, "YUN FEI=" ;
  W*D
40 GOTO 10

RUN
?46
W=46 YUN FEI=368
?82
W=82 YUN FEI=615
?103
W=103 YUN FEI=721
```

很明显,这里用20语句取代上一程序的20~70语句,无需任何转向就一次性地完成了判断和决定单价的工作,各判断条件依次罗列,井然有序,程序十分简炼、清晰。

在解决某些问题时,常常需要一系列的条件判断,而判断后的处理又属同一类型,这时很容易用复合表达式来予以简化。

例2. 打印CEC—I中华学习机区位字符表。

中华学习机内两片各一兆位的ROM固化了7445个全点阵图形字符(包括一、二级汉字库多种外文字母和特殊制表符等),通过图形字符与ASCⅡ码的对应关系可以打印出区位字符表便于输入时查找。由于区位码与机内码并不一致,因此程序设计的关键是实现二者的转换。一般用多次调用如下的子程序的方法来解决:

```
100 IF Q >9 AND Q<16 THEN 300
200 IF X <6 THEN Y=28+X
210 IF X <16 THEN Y=29+X
220 IF X <28 THEN Y=30+X
230 IF X >=28 THEN Y=31+X
```

上面的子程序完全可以用复合表达式构成的自定义函数取代:

```
DEF FN Y(X)=28++X+(X>5)+
(X>14)+(X>27)
```


多分支判断的解除

安徽省冶金工业学校 沈友敏

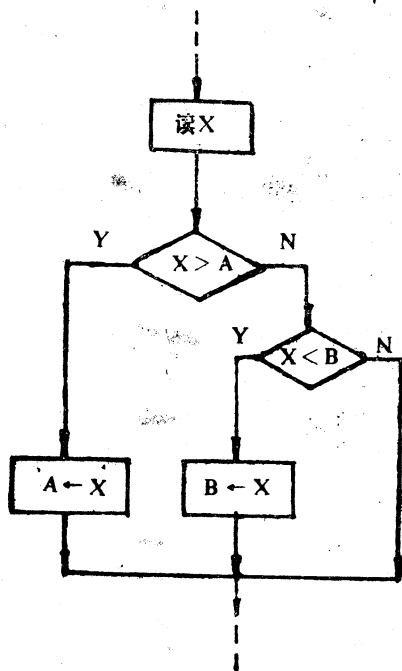
计算机具有逻辑判断能力,最基本的判断是对仅有两种情况的判断,即对给出的条件进行成立(是)与不成立(否)的判断。在程序中,它是由条件语句来实现的,经过一个判断之后简单地形成两个分支。但是,当需要判断的数量在两个以上时,便出现了多分支判断的情况。一般地,两个判断形成三个分支,三个判断形成四个分支,…… $n-1$ 个判断形成 n 个分支。

多分支判断的存在,往往使程序变得很复杂。因此,探索解除多分支判断的途径,对于优化程序设计是有意义的。本文将通过程序实例来阐明多分支判断的几种解除方法。

(一) 利用函数解除多分支判断

例1 打印出几个数据中的最大数与最小数。

这是数据处理中的一个选大元和选小元的问题。设以变量 X 表示数据, A 表示大元, B 表示小元。它们之间的关系构成为三支判断(框图1)。



(框图1)

```

GRUN
?1988
*1*
SU MO TU WE TH FR SA
          1 2
3 4 5 6 7 8 9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
*2*
SU MO TU WE TH FR SA
          1 2 3 4 5 6
7 8 9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29
*3*
SU MO TU WE TH FR SA
          1 2 3 4 5
6 7 8 9 10 11 12
13 14 15 16 17 18 19
20
  
```

```

GRUN
?1989
*1*
SU MO TU WE TH FR SA
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31
*2*
SU MO TU WE TH FR SA
          1 2 3 4
5 6 7 8 9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28
*3*
SU MO TU WE TH FR SA
          1 2 3 4
5 6 7 8 9 10 11
12 13 14 15 16
Break in 2
U
JLIST
  
```

利用绝对值函数可以解除分支判断, 因为

$ABS(A-X)$ 的值等于 $\begin{cases} A-X, & \text{当 } A > X \text{ 时} \\ X-A, & \text{当 } A < X \text{ 时} \end{cases}$

所以可以直接用赋值语句来选大元:

$$A = (ABS(A-X) + A + X) / 2$$

同样可以用赋值语句来选小元:

$$B = (B + X - ABS(B-X)) / 2$$

程序如下:

```

10 INPUT N
20 READ X
30 A = X: B = X
40 FOR I = 1 TO N - 1
50 READ X
60 A = (ABS(A - X) + A + X) / 2
70 B = (B + X - ABS(B - X)) / 2
80 NEXT I
90 PRINT "MAX="; A, "MIN="; B
100 END
110 DATA .....

```

程序1

例2 某商店对原价为P的商品以优惠价扩大销售, 规定:

购买数量Q	优惠幅度%
10件以下	0
10件~50件以下	5
50件~100件以下	8
100件~100件以上	10

要求根据商品的购买数量计算金额并打印出来。

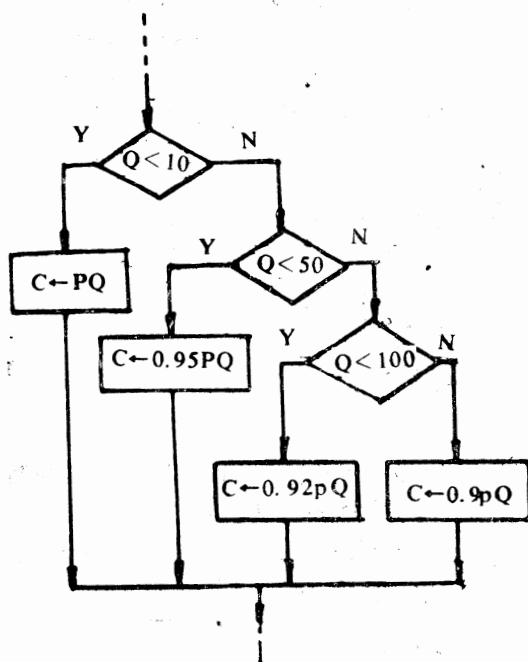
设以C表示金额, C、P、Q之间的关系构成四分支判断(框图2)

利用符号函数可以解出分支判断, 因为

$SGN(1 + SGN(Q-10))$ 的值等于 $\begin{cases} 0, & \text{当 } Q < 10 \\ 1, & \text{当 } Q \geq 10 \end{cases}$

对于Q-50、Q-100可以作同样处理, 所以

$$C = Q \cdot P - 0.05 \cdot Q \cdot P \cdot SGN(1 + SGN(Q-10)) - 0.03 \cdot Q \cdot P \cdot SGN(1 + SGN(Q-50)) - 0.02 \cdot Q \cdot P \cdot SGN(1 + SGN(Q-100))$$



(框图2)

程序如下:

```

10 INPUT P
20 INPUT Q
30 IF Q = -1 THEN 70
40 C = Q * P * (1 - .05 * SGN(1 + SGN(Q - 10)) - .03 * SGN(1 + SGN(Q - 50)) - .02 * SGN(1 + SGN(Q - 100)))
50 PRINT "QUANTITY:"; Q, "COST:"; C
60 GOTO 20
70 END

```

(程序2)

(二) 利用条件式解除多分支判断

分支判断的条件式也称关系式, 其运算的结果是取逻辑值, 即条件成立时, 条件式的值等于1, 条件不成立时, 条件式的值等于0。将条件式如符号函数一般参与运算, 同样可以解除多分支判断。

在例2中, 条件式 $Q \geq 10$ 的值等于

$$\begin{cases} 0, & \text{当 } Q < 10 \\ 1, & \text{当 } Q \geq 10 \end{cases} \quad \text{条件式 } Q \geq 50, Q \geq 100$$

有相似的结果, 所以

$$C = Q * P - 0.05 * Q * P * (Q \geq 10) \\ - 0.03 * Q * P * (Q \geq 50) \\ - 0.02 * Q * P * (Q \geq 100)$$

利用条件式解除多分支判断的程序如下:

```

10 INPUT P
20 INPUT Q
30 IF Q = - 1 THEN 70
40 C = Q * P * (1 - .05 * (Q > = 10) - .03 * (Q > = 50) - .02 * (Q > = 100))
50 PRINT "QUANTITY:";Q,"COST:";C
60 GOTO 20
70 END

```

(程序3)

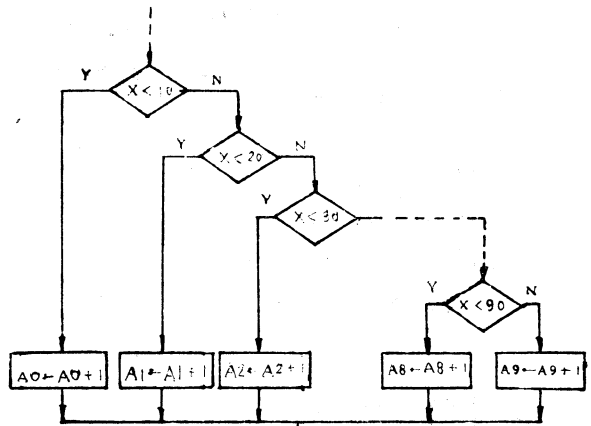
(三) 利用下标变量解除多分支判断

例3 统计N名参加升学考试的学生成绩,按10分为一个分数段分别打印各分数段的人数。

分数段	人数
0~10分以下	A0
10~20分以下	A1
20~30分以下	A2
30~40分以下	A3
40~50分以下	A4
50~60分以下	A5
60~70分以下	A6
70~80分以下	A7
80~90分以下	A8
90~100分	A9

设以X表示学生成绩, X与A0~A9的关系构成十分支判断(框图3)。

用下标变量替代简单变量, 确定变量序号的下标便可以 and 分数段联系起来。对于序号 $J = \text{INT}(X/10)$, 根据X值(从0~100), J可



(框图3)

能为0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10。因此只要用 $A(J) = A(J) + 1$ 就可以解除分支判断, 直接统计各分数段的人数。

考虑当 $X=100$ 时归并于 $A(9)$ 的要求, 在程序4中50程序行用了复合语句, 使在 $J=10$ 时重新赋值 $J=9$ 。

```

10 INPUT N
20 DIM A(9)
30 FOR I = 1 TO N
40 READ X
50 J = INT (X / 10): IF J = 10 THEN J = 9
60 A(J) = A(J) + 1
70 NEXT I
80 FOR J = 0 TO 9
90 PRINT 10 * J, "----", 10 * J + 10, ":", A(J)
100 NEXT J
110 END
120 DATA .....

```

(程序4)

计算任意形状图形面积的布尔运算法

沈阳航空工业学院航空系 孟庆春

摘要 本文给出了一种可用于求解任意形状图形面积的算法。该算法简洁、明了, 已得到具体应用。

一、基本图形面积单元

一个平面图总可由三类线段组成:

(1) 直线

(2) 圆弧

(3) 非圆曲线

其中非圆曲线可用圆弧逼近。所以本文考虑由真

线和圆弧组成的任意形状图形面积的解法。在此,将三角形、多边形及弓形称为基本图形单元。当平面图形边廓用直线段和圆弧描述后,任意形状的图形面积均可由这三种面积单元构成。

三角形面积可用行列式写为:

$$S_{\triangle} = \frac{1}{2} \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} = \frac{1}{2} \left(\begin{vmatrix} x_1 & y_1 \\ x_2 & y_2 \end{vmatrix} + \begin{vmatrix} x_2 & y_2 \\ x_3 & y_3 \end{vmatrix} + \begin{vmatrix} x_3 & y_3 \\ x_1 & y_1 \end{vmatrix} \right) \quad (1)$$

由上述求三角形面积的矩阵方法出发,可写出任意多边形面积的计算公式:

$$S_p = \frac{1}{2} \left(\begin{vmatrix} x_1 & y_1 \\ x_2 & y_2 \end{vmatrix} + \begin{vmatrix} x_2 & y_2 \\ x_3 & y_3 \end{vmatrix} + \dots + \begin{vmatrix} x_n & y_n \\ x_1 & y_1 \end{vmatrix} \right) \quad (2)$$

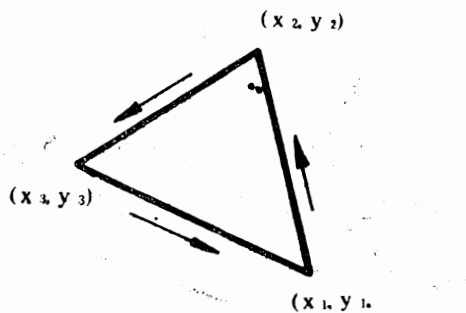


图1(a)

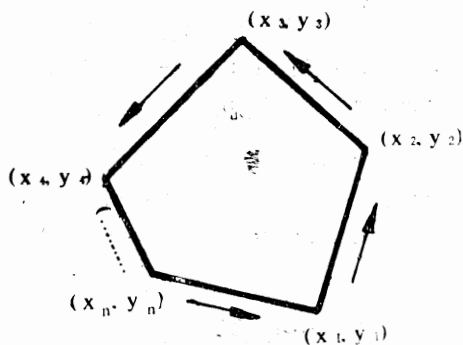


图1(b)

弓形面积有计算公式可直接使用:

$$S_{\text{弓}} = \frac{1}{2} R^2 (\theta - \sin \theta) \quad (3)$$

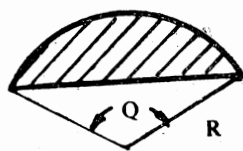


图2

二、布尔运算

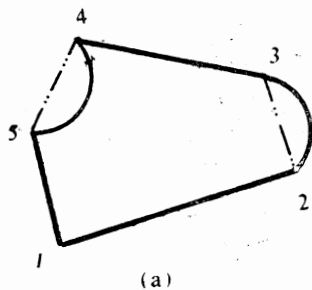
设A、B为两个集合, C为新的集合, 则三种布尔运算可表示为: (1) 并运算 $C = A \cup B$, (2) 差运算 $C = A - B$, (3) 交运算 $C = A \cap B$. 集合A与B代表什么取决于我们所研究的问题, 在此集合A、B代表基本图形面积单元。运用布尔运算的概念可得: 任一复杂的冲裁件图形, 一定可由基本图形面积单元经过若干布尔运算得到。因此, 可得到如下的图形面积算法。

三、计算图形面积的布尔运算法

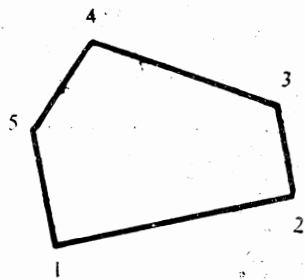
采用三个步骤求任意形状图形面积:

(1) 先计算特征多边形面积

特征多边形是指将图形边廓上的节点依次顺序直接相连所得到的多边形, 如图3-(b)中所示的多边形即为图3-(a)所示图形的特征多边形, 特征多边形面积用公式(2)计算。



(a)



(b)

图3

(2) 计算弓形面积

在求得特征多边形的面积后,再计算每个圆弧段所对应的弓形面积。如图3—4中 $\widehat{34}$ 弧与 $\widehat{61}$ 弧所对应的弓形面积 $S_{弓}^1$ 与 $S_{弓}^2$ 。

(3) 再计算真实的图形面积

基本约定:由起点到终点逆时针转过的圆弧为正向弧,从起点到终点顺时针转过的弧称为负

向弧。

将计算得到的特征多边形面积和各个弓形面积相加、减,便可得到真实的图形面积。对正向弧,在求图形面积时,是加上该弧所对应的弓形面积,即视正向弧对应的弓形面积为“正”,反之,对负向弧,则视其所对应的弓形面积为“负”。该方法用图示意如下。

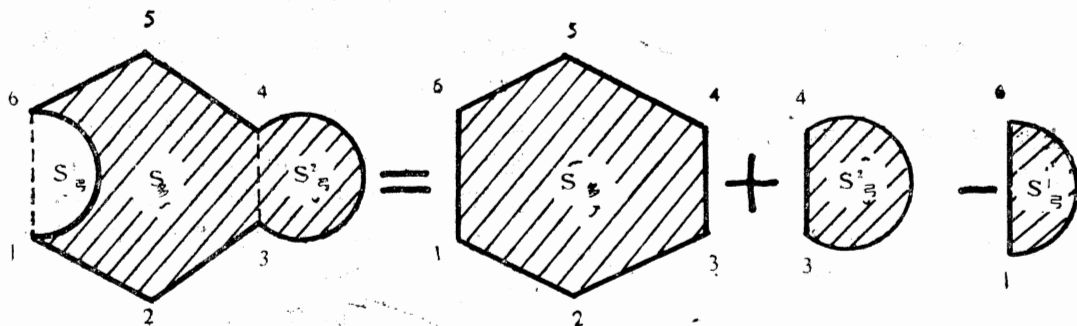


图 4

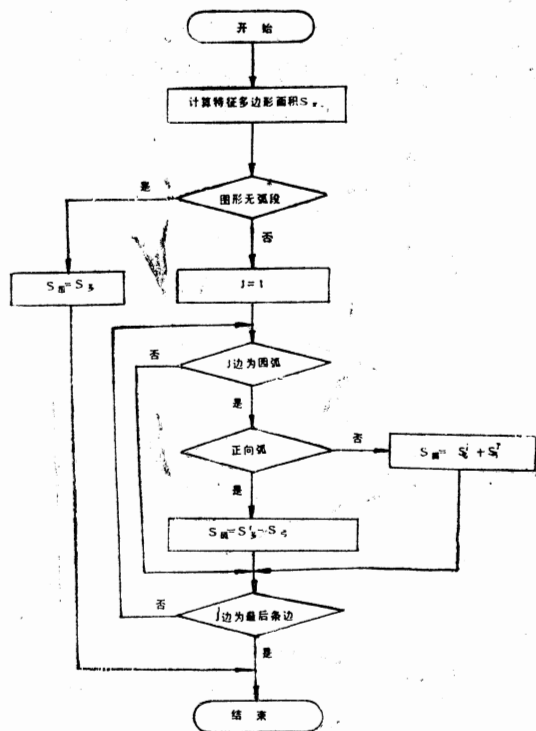


图 5

由图可见,图形面积 $S_{图}$ 是由 $S_{多}$ 与 $S_{弓}^1$ 进行差运算($S_{多} - S_{弓}^1$),再与 $S_{弓}^2$ 进行并运算($(S_{多} - S_{弓}^1) + S_{弓}^2$)。

该算法程序框图如图5中所示

四、结束语

本文给出的算法可用于求解由直线和圆弧组成的任意形状图形面积问题。作者已在计算机辅助最佳冲裁排样程序系统中应用了此种算法。

参考文献

- [1] 孟庆春、张钧:《微机辅助最佳冲裁排样》,沈阳航空工业学院学报 1988年, No1.



龙的传人

包世雄 篆刻

介绍一种新的裁剪算法

哈尔滨工业大学 刘卡林 孔庆复

摘 要

在计算机图形显示中,经常遇到的问题之一就是图形裁剪问题。本文在分析了已有的一些剪裁算法优缺点的基础上,提出了一种新的裁剪算法。算法以对线段的裁剪为核心,可以直接推广到对二维、三维图形进行裁剪,原理简单,实现方便,裁剪速度快。

一、引言

我们知道,要在视见区或取景窗口内正确地显示计算机所表示的图形,就必须把要显示的图形进行划分成可见部分和不可见部分的处理,这种处理的方法就称之为裁剪。例如如图1a所示的世

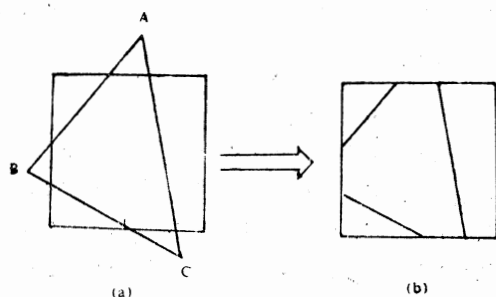


图1 裁剪前后的图形

界坐标系中的三角形经过裁剪处理就成为了图1b所示的屏幕坐标中的图形。显然,之所以要对图形进行裁剪,乃是由于屏幕的显示区域是有限的缘故。因此,针对这一问题各国的专家学者提出了各种各样的裁剪算法,我们对下面三种算法进行了分析:

1. COHEN-SUTHERLAND算法

(代码法):

这种方法出现较早,并被认为是一种较好的算法。它的原理是将平面按显示窗口的边框分成九个区,每个区用4位二进制码表示,如图2所示,任一线段的二个端点的编码都与它们所在的区号相对应,然后将端点的编码进行逻辑乘以判断线段的可见部分。显然,该算法能很快确定可见线段的端点,并舍弃不可见的线段,但不足的

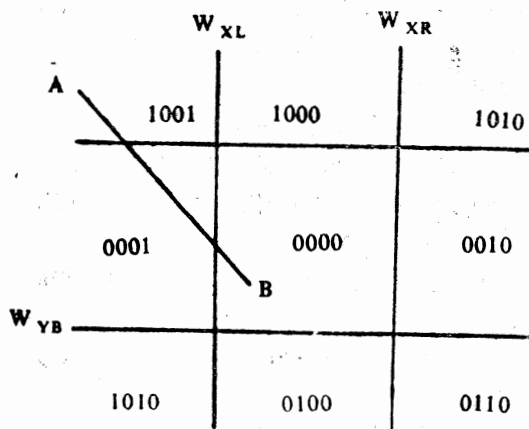


图2 Cohen-sutherland法

是该算法难以采用一些不便于进行逻辑乘的高级语言(如FORTRAN, ALGOL等)实现,且遇到图2中AB线段的情况,还需将线段分割进行判断。

2. 中点分割裁剪法

这种方法的主要思想是找到距端点最远的可见点,对两个端点同时进行这样的寻找,从而得到被判线段的可见部分,具体则通过多次判中点的可见性实现,因此,该算法由于要多次分割和判可见性,故裁剪速度不高,但它有一个显著优点,就是适用于没有乘除部件的机器,故硬件很容易实现。

3. 为了提高裁剪速度,并针对Cohen-sutherland算法在使用高级语言实现受到限制的情况,提出了一种用于交互式图形系统的裁剪算法。我们分析了这种算法,认为:该算法虽然步骤简单,但引入拓广矩形却没有充分利用线段和窗口位置关系所提供的信息,并且对拓广矩形

也存在多次分割判断的问题,因而裁剪速度提高不了多少。

从上面三种算法的分析可以看到,各种算法都有其优缺点,本文就是在这些分析的基础上,综合各算法的优点,并充分利用线段和窗口位置关系提供的信息,对线段进行了分类,提出了一种以对线段的裁剪为核心、逻辑简单、实现方便和裁剪速度快的新算法。

下面的讨论主要以线段的裁剪为主,对于二维多边形及三维情况的裁剪仅作原理性介绍。曲线我们则用有限条线段去逼近,从而也转化成线段裁剪问题。

二、裁剪算法原理和步骤

图3所示是线段相对于窗口的各种情况,我们分析这些线段和窗口的位置关系,可以知道,线段有下面三种情况:

1. 线段完全在窗口内,即全部可见。如ab线。
2. 线段完全在窗口外,即全部不可见。如cd线。
3. 线段被窗口分成几部分,则有且只有一部分是可见的。如线段ef、gh。

对于第一、二种情况处理很简单,第三种情况稍微复杂一点,但可以转化成对第一、二种情况的处理。并且我们可以推出如下的结论。

a. 线段如果和窗口有交点,则交点至多只有二个,且交点一定是该线段可见部分和不可见部分的分界点。

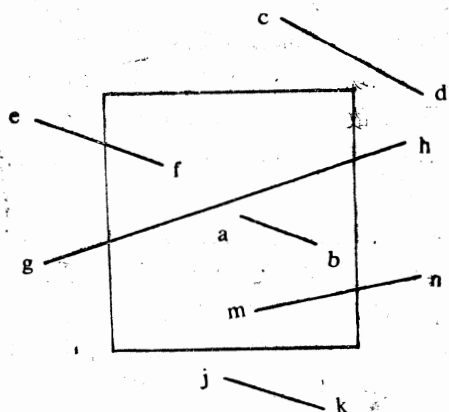


图3 线段和窗口的位置关系

b. 线段至多被窗口分成三部分,且不可见部分的下一段一定是可见部分,可见部分的下一段一定是不可见部分。由于可见部分有且只有一段,故经过一次可见性判断,就可以得到可见部分的端点,这个结论大大减少了裁剪的计算和判断。

c. 在被交点分成的小分段上注取一点(一般取中点),若该点可见,则小分段可见,若该点不可见,则小分段亦不可见。

线段和窗口无交点(即第一、二种情况),则只需取线段的中点判其可见性,再应用结论c即可。若线段和窗口有交点(即第三种情况),我们则首先求出线段和窗口的交点,取一小分段进行判断,并根据判断结果应用结论b、c即可完成对线段的裁剪。

以上就是本算法的基本原理,根据这个原理我们设计了如下的裁剪步骤,步骤中采用了一些优化措施。

一、Box比较:

Box比较的目的就是检查线段和窗口是否可能相交,若可能相交,则进行下一步的求交处理,若不相交,则可知该线段不可见,从而可以很快舍弃。显然,经过这样的粗判,可以避免不必要的求交计算,提高裁剪的速度。

比较的方法是给线段建立一个最小外接矩形盒,然后和窗口边框进行相交可能性判断,设线段和窗口的座标值如图4所示,则当

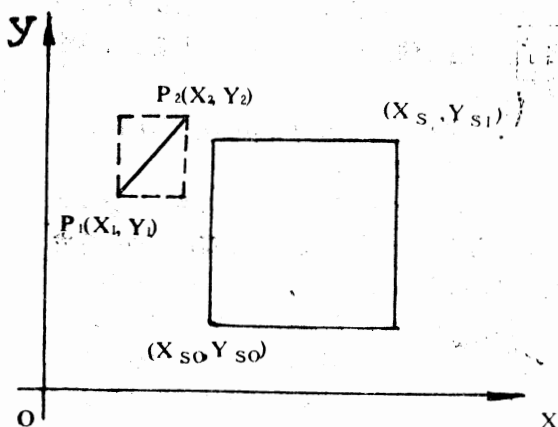


图4 Box比较

$\max(x_1, x_2) < x_{s0}$ 或

$\min(x_1, x_2) > x_{s1}$ 或

$\max(y_1, y_2) < y_{s0}$ 或

$\min(y_1, y_2) > y_{s1}$

时, 线段和窗口无交点, 否则进一步进行求交处理。

二、求交处理:

当线段和窗口可能相交时, 就要求它们的交点, 找出有效交点。有效交点就是既处于线段内又处于窗口边框上的那些交点。线段和窗口的求交, 我们将它分解成线段和窗口的四条边框的求交计算, 线段和窗口求交计算的具体方法在〔1〕, 〔2〕, 〔4〕中均有介绍, 在此不再赘述。

由于线段和窗口至多只有两个交点, 因此, 我们在具体编程中, 当求得有效交点为两个时就不再进行后面的求交计算, 并且可以肯定这两个有效交点就是线段可见部分的端点, 从而裁剪结束。若存在一个有效交点或没有交点, 则须进行小分段的可见性判断。

三、包容性测试:

根据结论C, 我们知道, 小分段的可见性判断可以通过对小分段上点的可见性判断来完成。而判断点的可见性只要测试一下该点是否在窗口内, 即包容性测试, 若点在窗口内则可见否则不可见。

为此, 我们在窗口的左侧引一直线 l , 取小分段的中点, 并由该点向直线 l 引垂线, 统计该垂线和窗口边框的交点数, 如图5所示。显然, 我们只须计算垂线和边框 W_1W_2 、 W_3W_4 的交点即可, 并且有当交点数为奇数, 则点在窗口内, 即被判线段可见, 输出端点即完成裁剪, 否则, 点在窗口外, 即被判小分段不可见, 但根据结论b可知该小分段的下一段一定是可见的, 故输出这个可见小分段的端点即可。至此, 算法结束。

对于二维多边形裁剪, 图6(a)所示, 我们定义: 二维多边形是有向的、封闭的、自身不相交的多边形。多边形的方向是这样规定的: 当沿着多边形的边前进, 其左侧应是多边形的内部, 右

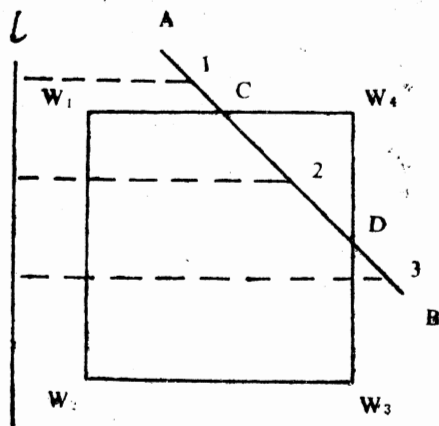


图5 包容性测试

侧是多边形的外部, 即多边形按逆时针走向。这样我们只须用窗口边框沿多边形走向逐一对其每一条边进行裁剪即可, 显然这样裁剪的结果得到的是非封闭的图形, 如图6(b), 为此我们将窗口也进行定向, 并按其走向用多边形裁剪它的每一条边框, 即可得到封闭的多边形图6(c)。

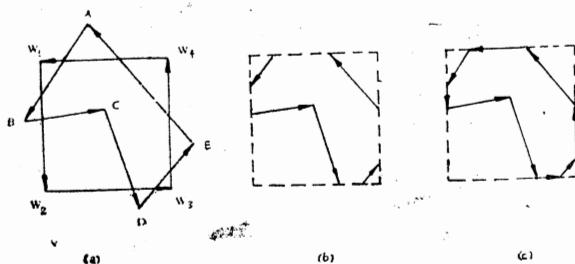


图6 二维多边形裁剪

将以上的讨论进行推广, 我们即可得到三维空间的裁剪算法。三维裁剪同样以裁剪空间线段为核心。以平引投影的裁剪为例, 设其观察体裁剪空间如图7所示。可知空间线段和这个观察体的位置关系也只有三种情况, 即:

1. 线段全部处于观察体内;
2. 线段全部在观察体外;
3. 线段被观察体分成几部分, 并有且只有一部分在观察体内。

因此, 我们同样可以得到和二维裁剪一样的步骤, 即: Box比较粗判, 求交计算和包容性测试三步。显然, 要注意的是: 在三维情况下,

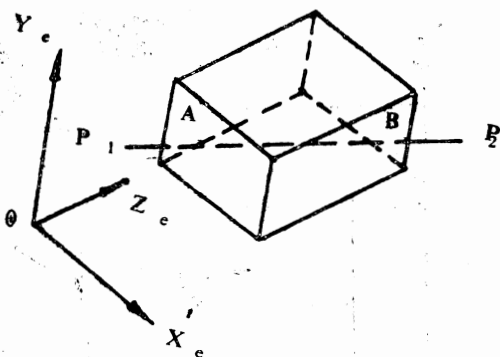


图7 平行投影裁剪空间

Box比较还需要增加对Z方向的比较,才能使判断彻底,并且交点也是指的线段和观察体上某个面的交点,包容性测试则是判断点是在观察体的内部还是外部,内部的点所在小分段是可见的,外部的点所在小分段是不可见的。

三、结束语

上面我们只讨论了处理一般情况下载剪问题的算法,在具体编程时则还要考虑对一些特殊情况的处理,如线段和窗口的边框重叠、线段的端点和窗口相交以及线段和窗口角点相交等,为此,在统计交点数时我们作了相应的规定,使本算法同样适合处理这些特殊情况。具体规定在此就不一一介绍了。

本算法由于从对线段的裁剪出发,因而适合对所有种类的多边形进行裁剪,无论该多边形是凹的还是凸的,并且也符合国际上要求图形系统应该把二维、三维情况进行统一处理的要求。从讨论中也可以看到,本算法原理简单并且充分利用了各种几何信息,大大减少了裁剪的计算和判断,不失为一种能提高裁剪速度的有效算法。

本算法已用 FORTRAN 语言在 IBM-PC/XT 微机上实现。

参 考 文 献

1. Wolfgang K.Gili, "Interactive Computer Graphics—Data Structures, Algorithms, Languages", Prentice-Hall, 1978.
2. 孙家广, 许隆文编著, "计算机图形学", 清华大学出版社, 1986.
3. 张文焘, "用于交互式图形系统的裁剪新算法" 机械工业自动化, 1983.2.
4. R.多尼〔法〕著, 田宝华译, "微型计算机BASIC语言绘图50例", 电子工业出版社, 1985.
5. W.M.纽曼, R.F.斯普劳尔著, 易晓东等译, "对话式计算机图形显示原理", 科学出版社, 1984.
6. 辜凯宁, 周嘉玉, "三维图形显示技术的实现" 计算机图形学及辅助设计第二届年会论文集, 1987.
7. D.F.罗杰斯〔美〕著, 梁友栋等译, "计算机图形学的算法基础", 科学出版社, 1987.

多功能打印曲线程序

国防科技大学 朱龙根

一、问题的提出

计算机计算的模型大多数是有明显的物理意义的,可是通过屏幕或打印机一般得到的是一大堆离散的数据。一般说来,图形比数字更为直观,更为形象。在工程和科学计算中最好能将计算结果实时地或事后地绘成曲线,研究其发展规律,用以交互式控制程序的运行。但有时由于受到计算机硬件和软件的限制,图形显示也会受到相应的限制。本文介绍的多功能打印曲线程序,是利用屏幕或打印机的打印字符能力,用各种符号组成曲线。它们虽然不如绘图屏幕或绘图仪等

画得那样好,但对于定性讨论是有益的。这种方法对硬件要求极低,无需具有绘图能力的图示器和相应的适配器,也不要求打印机具有点阵绘图能力,而只是利用打印机能打印的 ASCII 码组成绘图,一般可用于画简单的美术图、流程图;一元和二元函数曲线。本文介绍的程序用 IBM FORTRAN 2.0 版编制,适用于 IBM 微型机系统以及其兼容机(如 0520 等),打印机为 EPSON FX 系列,稍加修改可移植到所有配有 FORTRAN 语言的计算机系统。

二、多功能打印曲线程序——

DRAW·FOR.

工程和科学计算,绝大多数采用 FORTRAN 数值计算语言。本程序因此可以作为一个子程序,可供随时调用(见附录1)。下面介绍其哑元的规定:用户只要按要求写对哑元就可以完成打印曲线的工作。所有的哑元均是输入参数,并且符合隐式说明(I—N规则)。为了加快运行速度和减少数据区,整型数全用 INTEGER *2 定义。

- (1) N 打印曲线的条数; $N < 10$ 。
- (2) M 每条曲线中的数据个数。
- (3) Y $N * M$ 个元素的二维实形 *4 的数组,存放所要打印的曲线数据, $y_i, j = f_i(x_j), j=1, M, i=1, N$ 按行存放;第一条曲线数据放在第一行, ... 第N条放在第N行。
- (4) KFILE0 控制曲线数据的来源。当 KFILE0 = 0 时,则表示从磁盘文件 DRAW.DAT 输入。用此功能可使此程序独立运行或与其他应用软件进行通讯(例如,与 BASIC、dBASE、Pascal 等进行通讯)。数据存放格式为
FORMAT (TE11.4, 3x)
调用时它占用第12号通道
- (5) NY Y轴方向的长度;以字符为单位(下同)、用以调整曲线的幅值。
- (6) NDX X轴上刻度线的间距。当 $NDX > M$ 时,表示不画刻度线。
- (7) NDY Y轴上刻度线的间距。当 $NDY = 0$ 时,表示不画刻度线,并将坐标原点移到 $Y=0$ 处;其他情况是处于整个曲线数据的最小值处。
- (8) NXL X轴上刻度线的长度;当 $NXL = NY$ 时,形成X轴上的网格线。
- (9) NYL Y轴上刻度线的长度;当 $NYL = NY$ 时,形成Y轴上的网格线。
- (10) L2C1D0 控制输出通道

0	磁盘文件
1	屏幕
2	打印机

此形成的曲线磁盘文件名为 DRAW.PIC, 为 ASCII 文件, 可以进入文本处理程序等, 从而形成图文并茂的文章。一般其他绘图方法是不能实现这一点的。
调用时占用第11号通道。

- (11) NH0V1 控制曲线坐标系的放置方向。当 $NH0V1 = 0$ 时, 表示X轴是水平放置, 在打印纸上即为垂直走纸

方向。此时如在屏幕上显示曲线, 最好为

$$M \leq 79, NY \leq 3$$

以便使整个屏幕能完整地显示曲线。如用打印机打印, M要小于打印机可打印的行宽。当 $NH0V1 = 1$ 时, 表示X轴是垂直放置, 此时对屏幕显示要求 $NY \leq 79$, 对打印机则要求 $N < \text{打印行宽}$ 。

- (12) TITLE 图形标题, 是 Character *20 型字符变量。

本程序以不同的字符打印各条曲线, 所用字符依次为: + * : @ & = # \$ %。为了使打印形式精细, 利用了打印机的控制命令。这些命令与打印机的型号有关。此程序适合 EPSON 公司的 FX 系列打印机和 BROTHER 公司的 2024L。控制码 CHAR (14) 使当前打印行字体横向放大一倍。CHAR (15) 控制打印机进入压缩体状态, 横向缩小一倍。CHAR (27) + '3' + CHAR (10) 控制打印机纵向走纸为 (10/216) 时/行。CHAR (18) + CHAR (27) + '2' 控制打印机进入正常状态。屏幕显示时的坐标轴是由图形码组成的, 即扩充 ASCII 码 CHAR (179) 和 CHAR (196)。

三、使用方法

下面举例说明:

主程序见附录2, 输出参数和绘制的图形见图1。图中 $y(1, 1-80)$ 和 $y(2, 1-80)$ 分别为两条曲线在X网格线处的值。Ymin, all 和 Ymax, all 为两条曲线中的最小值和最大值。Ymin (1), Ymax (1) 等相类推。

Drawing Sample

1987.6.7 Version Zhu Long-gen

$y(1, 1) = .475417E-01$	$y(2, 1) = .452230E-01$
$y(1, 20) = .309560$	$y(2, 20) = .113881$
$y(1, 40) = .123060$	$y(2, 40) = .166544E-01$
$y(1, 60) = .702595E-02$	$y(2, 60) = .349801E-03$
$y(1, 80) = -.138613E-01$	$y(2, 80) = -.253879E-03$
$Ymin, all = -.179248E-01$	$Ymax, all = .322329$
$Ymin(1) = -.139248E-01$	$Ymax(1) = .322329$
$Ymin(2) = -.330379E-03$	$Ymax(2) = .176844$

Type: +:Y1, *:Y2,

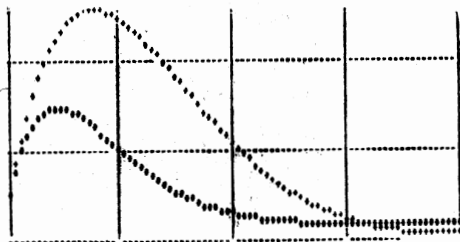


图1

附录1

```

SUBROUTINE Draw(N,M,Y,KFILE0,NY,NDX,NDY,NXL,NYL,L2C1D0,
&      NH0V1,Title)
  IMPLICIT INTEGER*2 (I-N)
  CHARACTER*1 T(9),BLANK,AINT,PAIM,LINE(131,100),CFORM*10,
&      CFORMT(10), Title*20,DAY
  LOGICAL PRNSDK
  DIMENSION Y(N,M),YMAX(9),YMIN(9)
  EQUIVALENCE(CFORM,CFORMT(1))
  DATA BLANK,AINT,PRIM/' ','|','_'/
  DATA CFORM/'(1X,130A1)'/,DAY/'N'/
  DATA T(1),T(2),T(3),T(4),T(5),T(6),T(7),T(8),T(9)
&  /'+','*',':','@','&','-','#','$','%'/
  IF(L2C1D0.EQ.0) OPEN(11,FILE='DRAW.PIC',STATUS='NEW')
  IF(L2C1D0.EQ.1) OPEN(11,FILE='CON:')
  IF(L2C1D0.EQ.2) OPEN(11,FILE='PRN:')
  IF(NH0V1.EQ.0) THEN
    IF(L2C1D0.EQ.1) THEN
      AINT=CHAR(196)
      PRIM=CHAR(179)
    ELSE
      AINT=' '
      PRIM='|'
    END IF
    ELSE IF(L2C1D0.EQ.1) THEN
      PRIM=CHAR(196)
      AINT=CHAR(179)
    END IF
    NY1=NY+1
    M1=M+1
    PRNSDK=(L2C1D0.EQ.2).OR.(L2C1D0.EQ.0)
    IF(KFILE0.EQ.0) THEN
      OPEN(12,FILE='Draw.DAT')
      READ(12,15,END=19) ((Y(I,J),J=1,M),I=1,N)
19    ENDFILE 12
    END IF
    IF(PRNSDK) WRITE(11,16) CHAR(14)
    WRITE(11,17) Title
    DO 21 K=1,M
      IF((MOD(K,NDX).EQ.0).OR.(K.EQ.1))
&        WRITE(11,25) (I,K,Y(I,K),I=1,N)
21    CONTINUE
    DO 26 I=1,N
      YMAX(I)=Y(I,1)
26    YMIN(I)=Y(I,1)
    DO 30 I=1,N
      DO 30 J=2,M
        YMAX(I)=AMAX1(YMAX(I),Y(I,J))
30      YMIN(I)=AMIN1(YMIN(I),Y(I,J))
      YMAXA=YMAX(1)
      YMINA=YMIN(1)
      DO 33 I=1,N
        YMAXA=AMAX1(YMAX(I),YMAXA)
33      YMINA=AMIN1(YMIN(I),YMINA)
      SCALE=NY/ABS(YMAXA-YMINA)
      WRITE(11,31) YMINA,YMAXA
      WRITE(11,32) (I,YMIN(I),I,YMAX(I),I=1,N)
      WRITE(11,27) (T(I),I,I=1,N)
      IF(PRNSDK) WRITE(11,65) CHAR(15),CHAR(27),'3',CHAR(10)
      IF(NDY.EQ.0) JZ=NINT(-YMINA*SCALE+1)

```

```

DO 60 J=1,M
  IF(J.LE.1) THEN
    DO 39 I=1,NY1
      LINE(I,J)=PRIM
    ELSE
      DO 40 K=1,NY1
        LINE(K,J)=BLANK
      END IF
      IF(NDY.EQ.0) THEN
        LINE(JZ,J)=AINT
      ELSE IF(J.LE.NYL) THEN
        DO 38 K=1,NY,NDY
          LINE(K,J)=AINT
        ELSE
          LINE(1,J)=AINT
        END IF
      IF(MOD(J,NDX).EQ.0) THEN
        DO 46 JK=1,NXL
          LINE(JK,J)=PRIM
        END IF
      DO 50 I=1,N
        IY=NINT((Y(I,J)-YMINA)*SCALE+1)
        IF(IY.GT.NY1) IY=NY1
        LINE(IY,J)=T(I)
      CONTINUE
      IF(NH0V1.EQ.0) THEN
        CALL IntChr(3,M1,CFORMT(5))
        WRITE(11,CFORM) ((LINE(I,J),J=1,M1),I=NY1,1,-1)
      ELSE
        CALL IntChr(3,NY1,CFORMT(5))
        WRITE(11,CFORM) ((LINE(I,J),I=1,NY1),J=1,M1)
      END IF
      IF(PRNDISK) WRITE(11,66) CHAR(18),CHAR(27),'2'
      CLOSE(11)
      WRITE(*,70)
      READ(*,'(A1)') DAY
      IF((DAY.EQ.'Y').OR.(DAY.EQ.'y')) GOTO 1
15  FORMAT(7E11.4)
16  FORMAT(1X,A1\))
17  FORMAT(3X,A//12X,'1987.6.7 Version  Zhu Long-gen'/)
25  FORMAT(3X,2('Y(',I2,',',I3,')=' ,G13.6,2X))
27  FORMAT(3X,'Type: ',9(A1,':Y',I1,',','))
31  FORMAT(3X,'Ymin,all =' ,G13.6,2X,'Ymax,all =' ,G13.6)
32  FORMAT(3X,'Ymin(',I2,') =' ,G13.6,2X,'Ymax(',I2,') =' ,G13.6)
65  FORMAT(' ',4A1////)
66  FORMAT(' ',3A1)
70  FORMAT(' Draw again(N/Y): \))
      RETURN
    END

SUBROUTINE IntChr(M,Num,Chr)
  CHARACTER*1 Chr(M)
  INTEGER*2 Pow10(5),Num,I,M,M1
  DATA Pow10/1,10,100,1000,10000/
  M1=M+1
  DO 10 I=M,1,-1
    CHR(I)=CHAR(MOD(Num/Pow10(M1-I),10)+48)
  CONTINUE
  RETURN
END

```

dBASE II PLUS (FOXBASE) 绘图功能

中国人民银行宝鸡市分行 邓京明

作为数据管理方面的能手, dBASE系列数据库管理系统赢得了绝大多数微型机用户的青睐, 成为目前世界上最为流行的微机数据库系统。然而无论是从dBASE I、II, 一直到拥有网络及多用户管理功能的第三代产品——dBASE II PLUS (FOXBASE), 都不具备绘图功能。这使得既需要很强的数据管理功能, 又需要灵活方便的作图功能的用户深感遗憾。他们或者忍痛割爱, 单独使用其它高级语言; 或者委屈求全, 采用dBASE与其它高级语言联合编程。这些都是不得已而为之。因此, 如果能在dBASE内部实现直接作图功能, 将会时大方便这方面的应用。

我们利用dBASE II PLUS(FOXBASE)调用汇编语言子程序的能力, 为其开发了一组屏幕绘图命令, 很好地实现了这种要求。绘图命令及其使用格式如下:

①LINE(直线): CALL FINE WITH "C X1 Y1 X2 Y2"; 颜色、两端点座标

②CIRCLE(圆): CALL CIRCLE WITH "C X Y R"; 颜色、圆心座标、半径

③ARC(圆弧、椭圆): CALL ARC WITH "C X Y R A AS AE ASPX ASPY"; 颜色、圆心座标、半径、特性(A为0与圆心无连线, 为1有连线)、起始角度、终止角度、X轴比例、Y轴比例

④RECTAN(矩形): CALL RECTAN WITH "C X1 Y1 X2 Y2"; 颜色、左上角座标、右下角座标

⑤FILL(填充): CALL FILL WITH "X Y C B"; 要填充区域内的任一点座标, 填充颜色、边界颜色

⑥CLS(清屏): CALL CLS

以上命令中C为颜色值, 取值范围为0~7, 对应的屏幕颜色如表1。X为横向点座标, Y为纵向点座标, 对于长城机的CH显示板, X、Y的取值范围分别为0~639和0~449。清屏命令CLS是为了弥补dBASE II PLUS (FOXBASE)的清屏命令CLEAR在长城机高分辨率下不能清除图形而编写的, 它可将文本和图形同时清除。

使用绘图功能时, 先用LOAD<程序名>

附录2

```

IMPLICIT INTEGER*2 (I-N)
CHARACTER Title*20
DIMENSION Y(2,80)
N=2
M=80
Title='Drawing Sample'
DO 10 I=1,N
  AI=FLOAT(I)
  DO 10 J=1,M
    FJ=0.05*FLOAT(J)
    Y(I,J)=EXP(-AI*FJ)*SIN(FJ)
  CALL Draw(N,M,Y,1,50,20,20,50,80,2,0,Title)
STOP
END

```

表1

值	0	1	2	3	4	5	6	7
色	黑	蓝	绿	浅蓝	红	紫	黄	白

将相应的绘图子程序装入内存,然后用CALL命令调用执行.当系统装入的子程序多于5个时,先用RELEASE MODULE<程序名>将暂时

不用的子程序释放,然后加载其它程序.下面是一个简单的演示程序及其运行结果(图1),读者可以看到绘图功能的使用方法.程序用8086/8088宏汇编语言编写,由于源程序太长,程序清单从略.适用机型和环境:长城(浪潮、SUPER)系列机,GWBIOS操作系统.(需要程序清单者,请与本文作者联系)

图1. 演示程序及运行结果

```
set talk off
load circle
load arc
load rectan
load cls
load zb
call cls
call circle with "4 320 220 50"
call arc with "3 320 220 60 1 120 240 1 1"
call arc with "2 320 220 60 1 300 60 1 1"
call rectan with "1 300 200 340 240"
@ 14,30 say "LQ-1000K SCREEN COPIER"
call zb with "4 15 31 15 52"
return
```



用BASIC程序驱动打印机进行屏幕拷贝

重庆建筑学院计算中心 魏 旭

在BASIC程序中有时需要把屏幕的数据或图形打印出来.但解释BASIC本身不具备这个功能.我们大家都知道用BASIC的CALL和USR命令可以调用汇编程序,所以笔者用DEBUG编了小段汇编程序,然后用BASIC的POKE命令把这段汇编程序代码写到内存的指定区域,再用CALL命令调用这段程序,进行全屏拷贝.

程序如下:

该程序首先把数据段DS进栈,然后把栈段地址SS和栈指针保存起来,产生一个05号中断,05号中断是打印屏幕,打印完后把栈段地址和栈指针恢复,这样不致于有时发生系统把返回地址

```
PUSH DS
MOV AX, 0000
MOV DS, AX
MOV [0000], SS
MOV [0002], SP
INT 05
MOV AX, 0000
MOV DS, AX
MOV SS, [0000]
MOV SP, [0002]
POP DS
RET
```

破坏而造成机器死锁,最后返回.这里因为是段间调用,所以返回时要用RET而不用RET.

BASIC程序如下:

```

10 REM PRINT SCREEN
20 DEF SEG=&H9FE0
30 FOR I=0 TO 31
40 READ J
50 POKE I,J
50 NEXT I
60 DATA &H1E,&HB8,&H00,&H00,&H8E,&HD8,&H8C,&H16
70 DATA &H00,&H00,&H89,&H26,&H02,&H00,&HCD,&H05
80 DATA &HB8,&H00,&H00,&H8E,&HD8,&H8E,&H16,&H00
90 DATA &H00,&H8B,&H26,&H02,&H00,&H1F,&HCB,&H00
100 SUB=0
110 CALL SUB
120 END

```

这段程序就能实现屏幕的拷贝, 但要注意两点

①全屏幕拷贝的内核为

```

INT 5H
RETF
3字节构成, 但

```

在有的机器上并不能顺利返回系统, 故加了几条保护栈指针的指令

②POKE的地址必须放在工作程序破坏不了的地方, 所以我们把它在系统的保护空间内, 对512K内存的机器, SEG=&H9FE0.

利用BASIC产生屏幕美术字的编程技巧

国营西北机器厂计算机中心 翟明德

在屏幕上显示美术字可以使屏幕提示更加醒目, 也富有艺术色彩, 增加屏幕的美观性, 现代广告艺术也离不开美观漂亮的美术字型, 笔者用BASIC编写了一个显示美术字的程序, 在输入一个汉字串后, 根据不同的选择可显示大立体, 斜宋体, 正宋体和大虚体等形式的美术字来, 本程序采用两种方法来实现美术字显示的, 一种是先定义一个数组, 然后将要显示的汉字点阵信息读进这个数组, 最后对此数组中的汉字点阵信息进行加工处理来显示美术字 (见程序中60—70行; 140—180行); 另一种是将要加工的汉

字信息显示在屏幕左上角, 然后逐次一边对汉字点阵采样, 一边对所采样信息进行处理来实现美术字显示的, 若挂上打印机即可打印屏幕上的美术字, 读者也可根据需要对本程序进行扩充或修改, 再加入自己的应用程序中去。

本程序在GW0520A, GW0520—CH中分辨率屏幕环境下运行通过, 使用汉字库为CCLIB, 打印屏幕时要用3070A, EXE打印机驱动程序。

下面是程序清单和运行举例:

```

10 '这是显示各种美术字的BASIC程序 88.7.28
20 SCREEN 2:KEY OFF:DIM H(24,24,18):CLS
30 INPUT "请输入要显示的汉字: ",PS
40 IF LEN(PS)>12 THEN PS=LEFT$(PS,12)
50 IF LEN(PS)/2<>INT(LEN(PS)/2) THEN L=LEN(PS)+1 ELSE L=LEN(PS):C
60 FOR K=1 TO L/2:LOCATE 1,1,0:PRINT MID$(PS,K*2-1,2):FOR X=0 TO 15
70 FOR Y=0 TO 15:H(X,Y,K)=POINT(X,Y):NEXT Y,X,K:LOCATE 1,1:PRINT "

```

C-DBASE-III 下的 屏幕装饰

国营西北机器厂计算机中心 翟明德

一个好的屏幕装饰不仅能使屏幕美观,使程序运行中的菜单提示更加醒目,而且使计算机与操作人员的界面更加柔和,令人心情舒畅。C-DBASEIII 虽然没有屏幕作图的功能,但只要巧妙地应用屏幕色彩设置命令 SET COLOR TO<颜色代码>和屏幕局部清屏命令 @<X,Y> CLEAR就可以设计出色彩丰富,灵活多样的屏幕画面。下面的DBASE 屏幕装饰子程序可以显示如图所示的画面,读者不妨试试用本程序的思路和方法将自己的应用程序“打扮”得更完美一些。

本程序在CW0520-CH 型机,高分辨率彩显环境下运行通过。

* C-DBASEIII 屏幕装饰子程序

set color to 15/9,14/12

clear

set color to 14/5

@ 1,2 clear

set color to 7/10

@ 3,8 clear

set color to 14/5

@ 3,72 clear

@ 19,8 clear

set color to 15/9,14/12

@ 1,78 clear

@ 21,2 clear

a='修改删除打印查询'

b='1----工资输入'

@ 7,20 get b

h=7

i=0

do while i<4

b=str(i+1,1)+'----工资'+subs(a,i*4+1,4)

if col(i)>50

@ h+3,20 get b

h=h+3

else

@ h,45 get b

endif

i=i+1

enddo

@ h,45 say 'Q----结 束'

set color to 15/6

@ 22,0 clear

set color to 14/12

wait '请选择:' to xz

set color to 15/9

retu

80 PRINT "Press any key to continue..."

90 AAS=INKEYS:IF AAS="" THEN 90 ELSE SCREEN 2:CLS:LOCATE 10,1

100 INPUT "请选择显示方式:1-大立体 2-斜宋体 3-正宋体 4-大虚体 5-结

110 FOR K=1 TO L/2:FOR X=0 TO 15:FOR Y=0 TO 15:IF H(X,Y,K)=0 THEN 1

120 IF ZT=5 THEN END

130 IF ZT=2 THEN 160 ELSE IF ZT=4 THEN 190 ELSE IF ZT=3 THEN 170

140 A=K*108-110+X*6:B=30+Y*5:CIRCLE(A,B),3:PAINT(A,B)

150 A=K*108-108+X*6:B=28+Y*5:CIRCLE(A,B),3:GOTO 180

160 A=(K+1)*50+15+X*3-Y:B=100+Y*1:LINE(A-1,B)-(A,B),1,B:GOTO 180

170 A=(K+1)*50+15+X*3:B=100+Y*1:LINE(A-1,B)-(A,B),1,B

180 NEXT Y,X,K:GOTO 80

190 SCREEN 1:CLS:PRINT PS

200 DEF FNC=C*2+30:DEF FND=50+D*2

210 FOR C=0 TO 130:FOR D=0 TO 16

220 DOT=POINT(C,D):IF DOT=0 THEN 240

230 PSET(FNC,FND),DOT

240 NEXT D,C:GOTO 80

决策表用于MIS开发的初探

浙江师范大学 张剑平

摘 要

本文简述了决策表的结构、特点、计算机实现方法以及和管理信息系统(MIS)开发中的应用。

运用软件工程思想开发管理信息系统(MIS)的特点之一,就是在系统开发过程中强调突出用户的作用,用户充分参与并提出新系统应该“做什么”和“怎么做”,有利于提高系统的合理性和实用性。在MIS的开发过程中,经常需进行大量的逻辑判断,为了综合运用管理人员(用户)的专业知识和思维方法,我们需要一种既准确、严谨,又简单明了,管理人员易于掌握的描述逻辑判断过程的方法。

通常采用语言(自然语言、形式语言)或流程图来描述一个判断过程,但是当条件多或过程很复杂时,前者不直观,后者太烦琐,况且一般管理人员也不习惯。

决策表(Decision table)是以表格形式来描述逻辑判断过程,使用该表不需正规的数学基础和特别的专业准备,管理人员只需花很少时间即可基本掌握。作为一种实用的描述工具,目前决策表在我国已逐渐受到人们的重视。本文将

叙述决策表的基本结构,通过例子说明它在MIS开发中的应用,特点和计算机实现方法。

一、决策表的基本结构

决策表由四部份组成:条件栏(Condition stub),条件记载(Condition entry),行动栏(Action stub)、行动记载(Action entry),如图1所示。

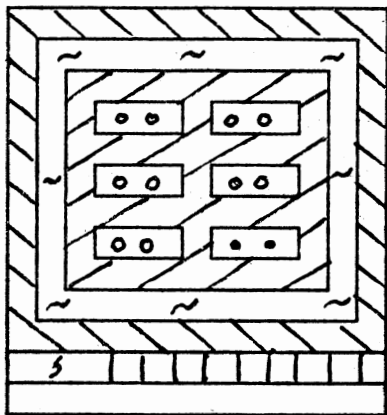
条件栏	条件记载
-----	-----
行动栏	行动记载

图1 决策表的结构








表格上方的两部份说明各项条件,用Y和N分别表示该条件成立与否,填写条件栏时应做到一个不漏,但不必考虑次序先后。表格下方的两部份说明各组条件下采取的相应行动,用X表示执行,填写时应按执行次序从头至尾依次列入。

例如,一种根据天气决定穿衣情况的“穿衣

附图:屏幕装饰后的菜单提示



图例:

-  --- 兰色区域
-  --- 绿色区域
-  --- 红色区域
-  --- 橘黄色区域
-  --- 红底黄字区域
-  --- 兰底白字区域
-  --- 闪烁显示区域

表”如下:

	项 目	规 则		
		1	2	3
C1	下雨	Y	Y	N
C2	天寒	Y	N	Y
A1	穿夹雨衣	X		
A2	穿单雨衣		X	
A3	穿羊毛衫			X

图2 穿衣表

上表的条件记载栏中仅有Y(是)、N(否)两种状态,在行动记载栏中也只有X(执行)和□(不执行)两种,因此也称此表为有限决策表。一般情况下,对于具有 n 个条件的决策表,因每个条件均有Y、N两种可能,所以在条件记载栏中最多有 2^n 列,但根据实际情况,并不一定满足该数目。比如上例中应有 $2^2=4$ 列,即

Y	Y	N	N
N	Y	N	N

可实际上允许仅考虑其中的3列。

两条行动情况完全相同的法则,若在各项条件中仅有一个条件 C_i 取值相反(一个是Y,另一个是N),则这两条法则可以合并,合并后条件 C_i 记为“—”,表示当 C_i 为Y或N时该法则都满足。

二、决策表应用于MIS开发的优点

在MIS的系统生命周期的四个阶段中,系

统分析和设计阶段的任务,是在对现行系统全面分析研究的基础上确定新系统的逻辑模型,而系统实现阶段则是用计算机语言实现和完善该模型。在上述两个阶段中,经常需要对某项活动的逻辑判断过程作详尽、准确的描述和化简,在这方面,同目前广泛采用的语言或框图描述相比,决策表是更为有效的工具。

在MIS开发中使用决策表有以下优点:

1.能清晰地表达复杂判断条件,比其他形式更接近企业管理人员和业务操作员的逻辑思维,同时系统分析及程序编制人员也能接受。在系统分析阶段,管理人员可借助决策表向设计人员传授有关事务处理的判断过程。

2.采用决策表能防止各种条件和行动的遗漏,及时纠正逻辑上的自相矛盾、前后不一致等情况的发生。在系统实现阶段有利于程序的设计、维护和修改。

3.由决策表转化成计算机程序比较方便、精炼。

4.由于决策表通俗易懂,绘制较容易,因而在系统的文档资料中(如《技术报告》、《用户手册》等)可代替某些流程框图及冗长的文字说明。

三、实例——订货决策表

因篇幅所限,仅以《物资管理系统》材料订货模块中的“订货决策表”为例:

	项 目	规 则					
		1	2	3	4	5	6
C1	库存 \leq 订货点	Y	N	Y	Y	Y	Y
C2	上月有销售	N	—	Y	Y	Y	Y
C3	库存 $<$ 临界点	—	—	Y	N	N	N
C4	是否可能脱库	—	—	—	N	Y	Y
C5	销售获利是否满意	—	—	—	—	N	Y
A1	按方案一订货			X			
A2	按方案二订货						X
A3	不订货	X	X		X	X	

图3 订货决策表

该表反映了材料供应部门制订订货决策时的判断过程。从表中可看出,若库存、销售等情况

满足规则3、6的条件组,采购员应分别按照方案一、二订货,否则均不需订货(此例已经过简化)

在系统分析和系统设计阶段,该表格由管理人员(或者请他们参与)建立,经设计人员化简,整理后作为编程的依据。最后将此表列入系统的技术文档交给用户,供系统的维护和二次开发时参考。

四、决策表的计算机实现

现以关系数据库语言 dBASE III 为例,介绍用计算机高级语言实现决策表的方法。

对于条件个数比较少的决策表,一般可直接采用CASE语句或 IF—ELSE 语句来实现,比如前述的“穿衣表”可写成:

程序1

```
.....
DO CASE
  CASE C1 .AND. C2
    DO A1
  CASE C1 .AND. (.NOT. C2)
    DO A2
  CASE (.NOT. C1).AND. C2
    DO A3
ENDCASE
.....
```

其中 C_i ($i=1, 2$) 取值 .T. (Y) 或 .F. (N) 表示决策表中相应的条件, A_j ($j=1, 2, 3$) 表示相应的行动的文件名称。

对于较复杂的决策表,可采用“开关法”、“分枝法”等不同方法编程。下面用“分枝法”来实现前面给出的“订货决策表”;

第一步:把决策表转化成决策树。

首先以条件 C_1 为树的根结点,引出Y、N两个分枝,分别连接, Y子表和“N子表”其中“Y子表”由 C_1 取“Y”或“—”的所对应的 $C_2, C_3 \dots C_5$ 条件行构成,“N子表”则由 C_1 中取“N”或“—”的 $C_2, C_3 \dots C_5$ 条件行构成,即如图4所示。

Y子表	1	3	4	5	6	N子表	2
C2	N	Y	Y	Y	Y	C2	-
C3	-	Y	N	N	N	C3	-
C4	-	-	N	Y	Y	C4	-
C5	-	-	-	N	Y	C5	-

图4 Y, N子表

然后以 C_2 的枝结点,从各子表引出新的Y、N分枝,作出新的子表……,直至作出的子表仅含一行(一条法则)或仅含一行(一个条件)为止,最后在各终结点填上相应的行动代码 A_i ($i=1, 2, 3$)。注意在选取下一结点时应优先取那些含“—”项少的条件 C_i 。

最终得到图5的决策树。

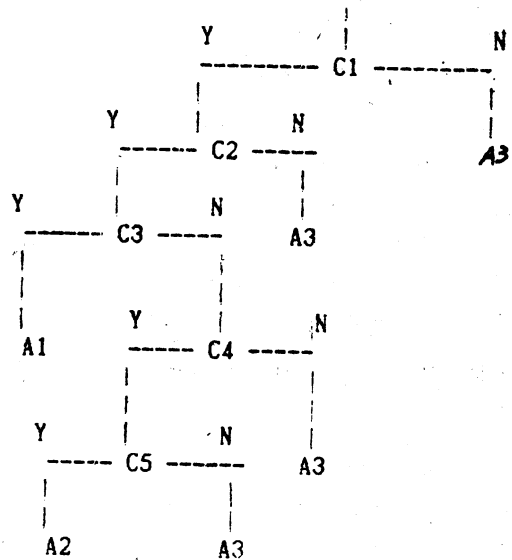


图5 决策树

第二步:由决策树写出计算机程序。本例的程序段如下。

由于每个IF语句中的条件均为简单类型的,并且对某一具体的执行过程,不必一一执行所有IF判断,因此“分枝法”写出的程序其运行效率较高,但可读性略为不足。该方法的详细情况请参见文献〔2〕p33。

在“分枝法”中,若事先对决策树进行优化,则程序还可进一步简化。

五、结语

1. 决策表的内容十分丰富,除了本文介绍的有限决策表(Limited Entry Table),根据表中条件的状态不同,还有扩展决策表(Extended Entry Table)和混合决策表(Mixed Entry Table),它们都各有特色,若能合理选择和灵活运用,则可描述、处理更为广泛、复杂的判断决策过程。

程序2

```

.....
IF C1
  IF C2
    IF C3
      DO A1
    ELSE
      IF C4
        IF C5
          DO A2
        ELSE
          DO A3
        ENDIF
      ELSE
        DO A3
      ENDIF
    ENDIF
  ELSE
    DO A3
  ENDIF
ELSE
  DO A3
ENDIF
.....

```

2. 管理信息系统 (MIS) 的开发需要科学的方法和手段, 决策表不仅适用于 MIS 开发的系统分析、系统设计, 还是程序编制和文档资料编写的有效工具。由于 MIS 这门学科本身刚处在开始阶段, 把决策表应用于 MIS 的开发更是一种新的尝试, 还有待于同行们进一步探索和完善, 本文仅仅为此起抛砖引玉的作用。

参 考 文 献

1. 薛华成, 汪授泓, 管理信息系统, 清华大学出版社, 1988.5.
2. Richard B. Hurley, Decision tables in software engineering, 1983.
3. R. Welland, Decision tables and computer programming, 1981.

一个实用的高精度实时数据采集系统

江汉石油学院 李华贵 李光祖
江汉油田仪表厂 陈家林

摘 要

本文介绍了由PC—1500A微型计算机为主体所组成的一个高精度实时数据采集系统, 阐述了系统工作原理、硬件与软件设计及特点等。

一、前 言

实时数据采集与处理是微机应用的一个重要方面, 为满足石油勘探开发现场的实际需要, 我们开发了一个高精度的实时数据采集系统, 该系统以PC—1500A为主体, 外扩32K字节存储器, 选用12位分辨率的AD574转换芯片。该系统具有诸如: 转换精度高, 具有一定的现场数据处理能力与绘图能力, 能使用BASIC语言及汇编语言编程, 可靠性高以及轻便等一系列特点, 特别适用于石油生产现场数据采集与数据存储以及实时诊断的需要。

二、系统硬件结构

其框图如图1所示: 由主机、接口电路、数据采集、数据存储及打印等部分组成, 重点阐述如下。

1. 32K存储器的扩充方法

为实现正确的联接, 先分析PC—1500A的CPU (型号为LH5801) 的读/写时序 (图2), 其中图a为读时序, 图b为写时序, $AB_0 \sim AB_{15}$ 为地址总线, 可寻址64K字节, 由于CPU内部设有两个存储区间的切换逻辑电路, 用以产生MEO及MEI两个允许存储信号, MEO控制内

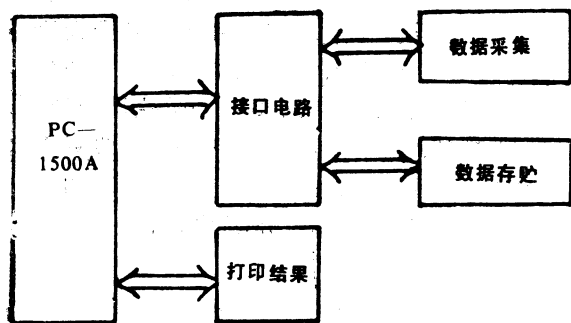


图1 系统框图

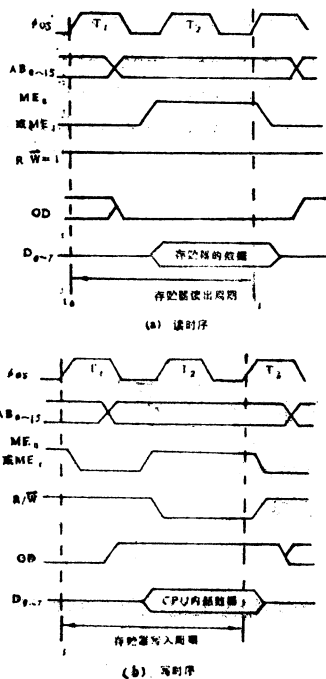


图2 L5H801读/写时序

64K字节，MEI控制外64K字节，并称之为井号区，所以实际可寻址128K字节， R/\overline{W} 为读/写控制信号，读存储器时， $R/\overline{W}=1$ ，写存储器时， $R/\overline{W}=0$ ，OD为禁止输出信号， $OD=0$ 时，对应于读存储器操作，因此，亦可把OD信号作为读信号使用。

32K RAM存储器由四片容量为8K的 HM6264LP-15组成，与PC-1500A的联接如图3所示，其选片信息号 \overline{CS} 利用地址高三位 A15、A

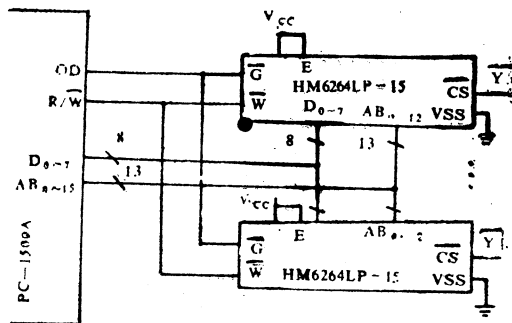


图3 32K字节存储器的扩充联接

14、A13通过74LS138译码得到（见图4），图中MEI=1时，方能产生外64K字节存储器的选片信号，上面一个74LS138译码器的低四位输出 $\overline{Y}_0 \sim \overline{Y}_3$ 分别接至四个8K存储器的选片端 \overline{CS} ，因而各芯片的地址范围相应为0000H~1FFFH、2000H~3FFFH、4000H~5FFFH、6000H~7FFFH，图中还利用 \overline{Y}_7 及下面一个74LS138译码器译出八个选片信号，注意， $\overline{Y}_0 \sim \overline{Y}_7$ 以及 $S_0 \sim S_7$ 均是井号区的空白区。图四中构成了两个并行接口，即由74LS273锁存器组成并行输出接口，

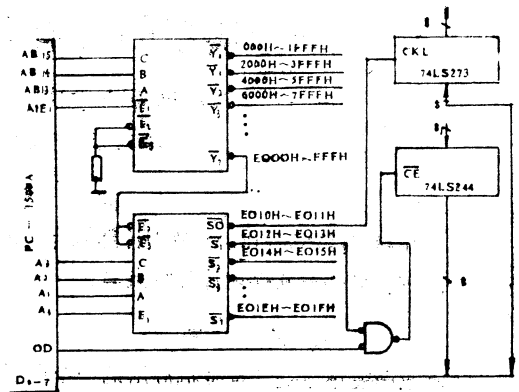


图4 输入/输出原理电路

即由74LS244组成并行输入接口，这两个接口是为AD574与CPU接口而设计的，前者用于输出通道选择信号，后者用于读入状态。

2. 数据采集

选用12位转换芯片AD574，片内提供基准电源，使得外围电路简单，还提供三态输出，可以与8位或12位字长的微处理器直接接口，转换

后的数据可采取12位并行输出方式,亦可先读出高8位而后后读出低4位分两次读出的方式,由于后一特点,容易实现AD574与8位字长微处理器的接口,与此同时,AD574除作12位转换外,也可作8位转换,模拟输入电压范围为 $0\text{V} \sim +10\text{V}$,或 $\pm 5\text{V}$,AD574有 $\overline{\text{CE}}$, $\overline{\text{CS}}$, $\text{R}/\overline{\text{C}}$, $12/\overline{8}$ 及 $\text{A}0$ 等5个控制端,不同组合下,实现的功能见表1)。

表1

CE	$\overline{\text{CS}}$	$\text{R}/\overline{\text{C}}$	$12/\overline{8}$	A0	功 能
1	0	0	×	0	启动12位A/D转换
1	0	0	×	0	启动8位A/D转换
1	0	1	接+5V	×	12位数据一起读出
1	0	1	接地	0	读出高八位二进制数
1	0	1	接地	1	读出低四位二进制数

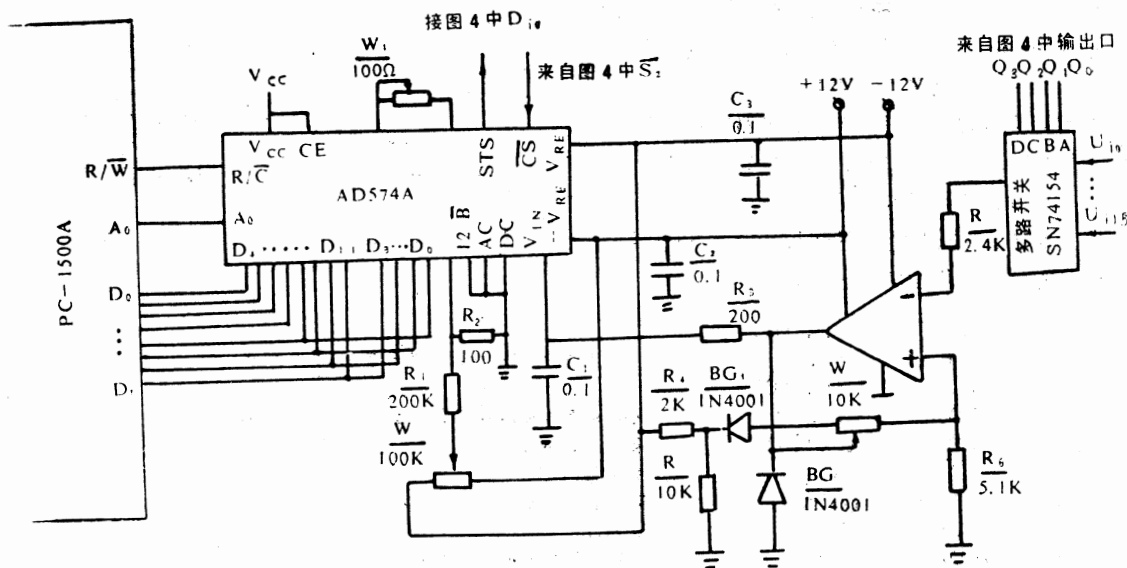


图5 采集原理图

数据采集系统电路如图五所示。

通过多路开关SN74154使得该系统可以分时采集16路模拟信号,利用图四所示输出端的低四位($Q_3 \sim Q_0$)进行通道选择,该端口的时钟接至下面一个74LS138的 $\overline{S_0}$,地址可定为E010H,由于LH5801的指令系统是统一编址的,故其输入输出时序与存储器的读写时序一致,当执行POKE并E010, DATA语句时,MEI的下降沿将引起 $\overline{S_0}$ 的正跳沿,此时数据将被锁存于上升沿触发的输出口中,利用其低四位($Q_3 \sim Q_0$)从16路模拟信号IN0~IN15中选择任一通道与AD574联通,考虑到AD574采集时负载电流大,多路开关又有一定的内阻,且各路内阻可能不一致,所以加一级运算放大器以保证精度用R3、C1进一步滤除噪声干扰

本系统中A/D转换器采用12位A/D转换,转换数据分两次读入的方式,根据表一的要求,可确定各控制引脚的联接。 $\overline{\text{CE}}$ 固定接高电平,选片端 $\overline{\text{CS}}$ 接译码电路中的 $\overline{S_2}$, $\overline{S_2}$ 占用两个地址,即E014H($\text{A}0=0$)和E015H($\text{A}0=1$),将CPU的最低地址A0接至AD574的A0,实现了选片端 $\overline{\text{CS}}$ 为0时A0有两种状态, $\text{R}/\overline{\text{C}}$ 接PC-1500A输出的读/写信号 $\text{R}/\overline{\text{W}}$,因此,可用写存储器指令启动转换,用读存储器指令分别读出高八位与低四位,因此 $12/\overline{8}$ 控制端接地。那么,选用E014H为地址启动12位A/D转换,以E014H和E015H为地址分别读出高八位与低四位。

由于PC-1500A对机外无法实现中断控制,因此可用查询方式查询采集间隔的定时时标和A/D转换结束信号,将AD574的转换结束信号

STS接图四中输入的一个引脚,譬如Dio,以供查询之用。

最后指出,电位器W1用于校准满刻度,W2用于调零,W3用于调节放大倍数。

三、软件设计

由上述的硬件电路可实现不同要求的数据采集,在此,提供一个通用的数据采集软件,用BASIC语言编制菜单提示程序(图6),主要回答:①采集路数(1~16),②各模拟通道的编号(0~15),③采集点数(≤ 255)根据需要可增加到两个以上字节的数量。将回答的参数按规定的格式存入MEO区的机器码存储区(例如从7C01H开始存放),由汇编语言采集时获取这些参数,照此要求进行采集。

根据图7的程序流程框图,用汇编语言编制如下的通用采集程序。

	LDI	YH, 00H	
	LDI	YL, 00H	: 数据区的首地址0000H(井号区)→Y寄存器
	LDI	XH, 7CH	
	LDI	XL, 02H	: X寄存器存放各路编号在所在内存的起始地址
	LDA	(7C11H)	: 采集点数→A
	STA	UH	: A→UH
-i①:	LDA	(7C01H)	: 采集路数→A
	STA	UL	: A→UL
-i②:	LDA	(X)	: 具体采集那一路的编号→A
	STA并(E010H)		: A→74LS273锁存, 选通模拟通道
	STA并(E014H)		: 启动AD574作12位A/D转换
-i③:	LDA并(E012H)		: 转换结束标志STS接图中74LS244的Dio, STS状态→A
	ROR		: 右移一位
	BCS	-i①	: 如果没转换完(C=1)则转
	LDA并(E014H)		: 读出高八位→A
	STA并(Y)		: 存入井号区
	INC	Y	: 地址增1
	LDA并(E015G)		: 读出低四位→A
	STA并(Y)		
	INC	Y	
	INC	X	: 指到下一路编号所在内存的地址
	DEC	UL	: 采集路数减1
	BZR	-i②	: 预置路数没转换完(Z=0)则转
	JMP	i, j	: 各路均采完一个子样后, 调采集间隔延时程序(略)
	DEC	UH	: 采集点数减1
	BZR	-i③	: 预置点数没完成(Z=0)则转
	RTN		: 返回到BaSIC状态

如果对任意两路模拟信号各采250个点,通过CE—150打印机所还原出来的波形见图8。应当指出,本系统具有较强的数据处理能力,根据需要还可以通过CE—158串行口或自设并行口将

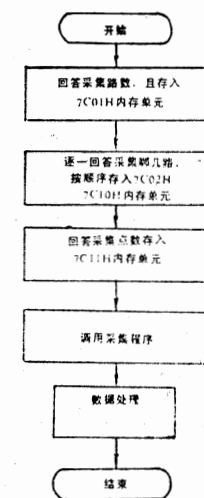


图6 总流程框图

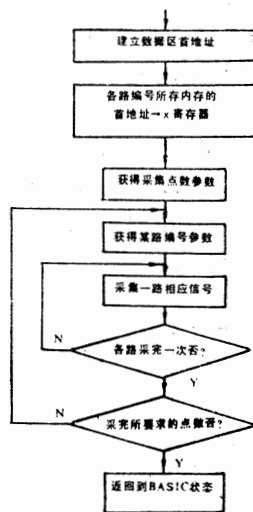


图7 采集程序框图

采集到的数据传送给其它系统机进行处理。

四、结束语

我们以本采集系统为硬件基础,在石油开采方面制作了两件仪器,通过使用,一致认为体积

Novell网络上的生产文件管理系统

深圳蛇口132信箱 郑晓风

随着计算机的日益普及,计算机网络已逐渐迈入大学、研究所、机关甚至工厂。对工厂管理者来讲,只有及时了解、分析并处理复杂纷多的生产文件如:工艺更改文件、另部件更改文件、生产图纸文件、QC(质量控制)文件等,才能有效地管理生产,高效率、高质量地生产出顾客满意的产品。

本着上述目的,作者在Novell网络上建立了生产文件管理系统。通过网络的应用,顾客或供应商可直接将有关文件输入网络,以便生产厂家及时了解最新的产品质量或其它有关信息,并作相应处理;生产厂家也可直接利用系统对生产进行迅速及时的管理。本系统是用当前国际上较新的数据库语言—dBASE 3 PLUS编写的,因而具有速度快、功能多等优点。系统的另一特点是所有文件处理工作都采用“菜单”形式,通过“选单”就可对文件进行各种处理,简单、明

瞭,用户使用方便。同时,本系统扩展性较强,只需稍作修改,就可作为一般文件、图书、资料的管理系统在Novell网络上运行。

下面对系统作一简单介绍。

一、系统总框图

系统总框图示于图1。

二、各模块简介

1. 系统标题信息显示模块

此模块向用户显示系统名称、版本、日期等有关信息。此模块不需用户作任何操作。

2. 产品类型及文件类型模块

此两模块的主要功能是根据用户选择的产品和文件类型查表搜索到其相应的文件库。系统对每种产品的不同类型的文件均分别建立了不同的文件库。例如有A、B两种产品、文件类型有QC(质量控制)、ECN(工艺更改通知),则系统为此建立图2所示数据库文件。

小,功耗低,功能强,携带方便。难以解决的是抗低温环境能力差。

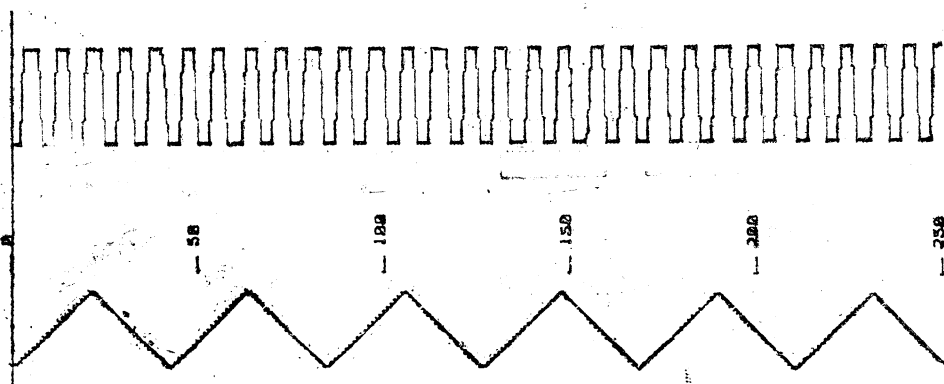


图8 恢复波形

参 考 文 献

- [1] “一九八五年世界集成电路大全”(3)“微处理器”,上海交通大学微机研究所,南洋电脑开发总公司,1985年10月
- [2] 刘植桢 王秀玲 岳震伍,“微型机控制系统设计——从芯片到系统”,1986年4月
- [3] Sharp Corporation, Sharp Pocket Computer Application Manual, Osaka, Japan, 1983

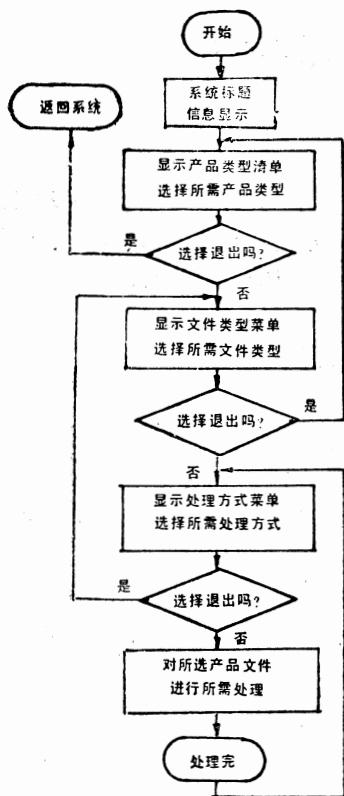


图1

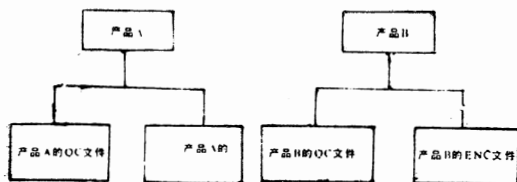


图2

通过分类建立文件库,提高了文件查找速度,增加了文件的易读性。

3. 处理方式模块

此模块是本系统对生产文件进行管理的最重要的模块,它提供以下几种文件处理方式:

A. 输入方式

在此方式下,用户可向相应文件库存入新文件的有关信息,框图示于图3。

B. 报告方式

本系统提供两种报告输出方式:屏幕显示及

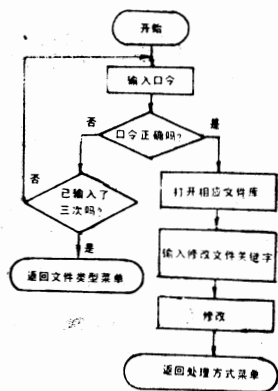


图3

打印。报告的内容有3种选择,即:(1)相应文件库所有文件的全部信息;(2)相应文件库某一文件的全部信息;(3)相应文件库所有文件的扼要清单。下面分别给出3种选择的流程图。

(a) 相应文件库所有文件的全部信息,见图4。

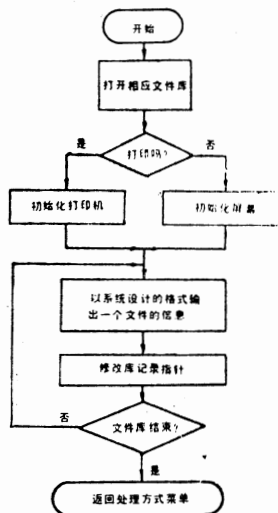


图4

(b) 相应文件库某一文件的全部信息,见图5。

(c) 相应文件库所有文件的扼要信息清单,见图6。

c. 修改方式

系统提供了两种常用的修改方式:(1)编辑方式;(2)删除方式。

西服显示的逼真性探讨

北京航空航天大学电子工程系 赵元平

摘 要

本文以增强服装显示的真实感为最终目的,采取光栅显示的手段,得到了西服的三维明暗效果图。众所周知,服装表面的复杂程度远远超过机械零件,服装CAD行业尚有许多问题需要解决,本文只是从问题的一个侧面开展工作,通过开窗变换、透视变换和有序颤抖(order dither)等基本处理过程,采取PHONE光照模型、HLS [hue, lightness, saturation] 颜色模型、Gourand亮度插值明暗法和Z—buffer消隐算法,利用Bezier曲面初步形成了一个交互式三维西服CAD系统,该系统暂时给出了几套西服(正在扩充过程中),而且能从任意方向观察,即显示的服装能以任意角度旋转。

引 言

随着科学技术的进步和人民生活水平的提高,特别是计算机辅助设计行业的突起,服装CAD也得到了迅速发展,人们不再满足于服装的二维平面图,而希望看到自己所需服装的逼真

效果,那么,由计算机生成具有真实感的服装在许多场合比起制做样品来说是一个更为容易、经济和更有效的方法,而且还允许设计者考虑更多的不同的设计方案。固然消除隐藏线和隐藏面的线架图具有一定的真实感,然而这跟实际生活中

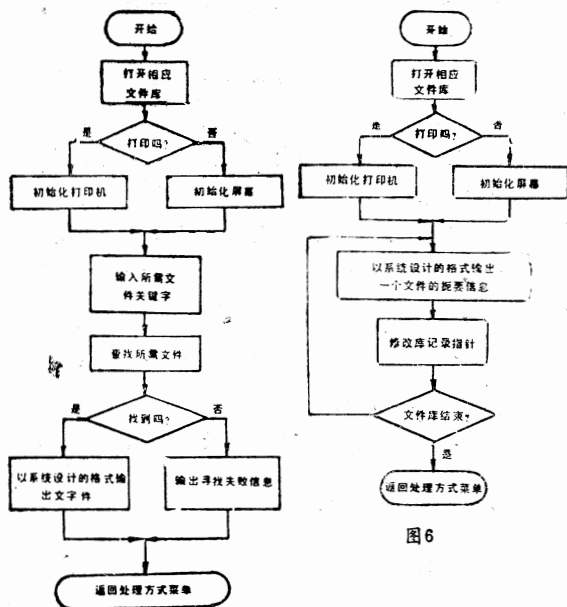


图5

(1) 编辑方式

在编辑状态下,可对需修改的文件记录进行修改,框图示于图7。

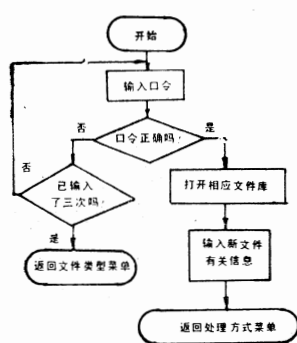


图7

(2) 删除方式

在编辑状态下,可方便地删除作废的文件记录。删除流程与编辑类似,在此就不多写了。

在选择输入或修改方式时,系统加设口令(Password),以防止非法用户破坏生产文件库。

以上对Novell网络上的生产文件管理系统作了一个简单介绍,希望它能早日成为生产管理者的好助手。

所见的服装还是相差太多,因此,本文探讨真实感更强的具有明暗效应的服装效果图。

基本原理、算法和结论

一、由于Bezier曲面是多项式曲面,其不同于样条曲面的特点之一就是独立性强,不会牵一发而动全身,将曲面所在区域划分得适当小,改动一片Bezier曲面所影响的区域是十分有限的,正是由于Bezier曲面所独具的几何直观性,使得它在交互设计方面有着巨大的潜力,本文就是采用Bezier曲面来构造服装的。

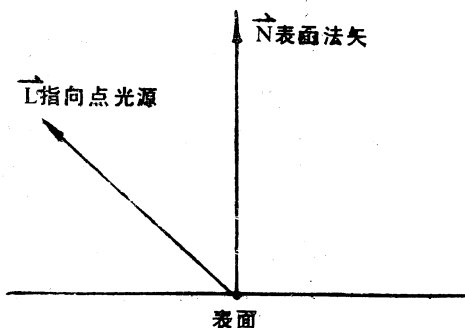
二、要得到服装的明暗效果图,增强其真实感,首先是消除隐藏面,其次是可见表面的明暗处理。

隐藏面的消除有许多种算法,本文采用Z—buffer算法与扫描线算法相结合进行处理,这些算法在此就不再复述。

一个物体的深度可用光强度来表示,离视点近的物体用较高的强度来显示,特别是眼睛对强度的分辨能力比对空间的分辨能力要弱,使得离散化后形成的误差,在视觉效应上得以消除或减弱,故而用明暗图来显示服装的方法是可取的。

要得到可见表面的明暗处理,首先要建立光照模型,因为,一个表面的外观取决于照射该物体的光源的类型、表面的性质(如颜色、纹理和反射)以及该表面相对于光源的位置和方向,还取决于它与其它表面的位置和方向。为简单起见,本文采用了PHONE的只包含环境光和点光源的光照模型:

$$I = I_a K_a + I_p K_p (\vec{L} \cdot \vec{N}) / (R + K)$$



\vec{L} , \vec{N} 均已规格化

I_a , I_p : 环境光和点光源光强

K_a , K_p : 环境光和点光源反射系数

R : 透视视点至表面的距离

K : 常数

以上各参数值的选择跟具体的调色系统有关,只能在实践中摸索确定。

服装与机械零件不同,产生阴影后效果反而不是很好,而且还花费大量的机时,本文把点光源设定与视点重合,就不会产生阴影,所附照片就是在这种假定下得到的。

由于Mach带效应很明显,给人形成的视觉效果是,所显示的服装是由一块一块的布料拼成的,即块与块之间虽然连续,但交线处一阶导数不连续,导致显示的服装的真实感明显下降,因此,要对多边形网格进行明暗处理,可采用的解决途径大致有三种:恒定的明暗度、亮度插值的明暗度和法向量插值明暗度。

恒定的明暗度法 (Constant shading) 是对整个多边形的明暗度只算出一个亮度值,这样,由于每一小平面有着与其相邻面不同的亮度,用来逼近曲面的每一个可见多边形都能分辨出来,相邻面间明暗度的差别因Mach带效应而得以加强,从而夸大了相邻面交界处亮度的变化,而法向量插值明暗度法对于解决本文的问题亦不太合适,因此,本文以亮度插值明暗度法为基础,提出了一种适合于服装CAD的明暗度处理法,其基本步骤为:

1. 计算各多边形平面的法向。如图1所示

\vec{N}_i , $i=1, 2, 3, 4$

2. 计算各顶点的法向矢量 N_v , 如图1所示,

$N_v = \frac{1}{4} \sum_{i=1}^4 \vec{N}_i$ 。对于服装的熨痕需要

突出表示,熨痕两侧的平面在熨痕处连续,但其一阶导数不连续,采用图2所示的处理方法,一个顶点计算两个法向量 \vec{N}_{v1} , \vec{N}_{v2} 。

其中: N_{v1} 用来插值1, 2两面的明暗度

N_{v2} 用来插值3, 4两面的明暗度。这样1—2和3—4的边界都成了均匀明暗度,而

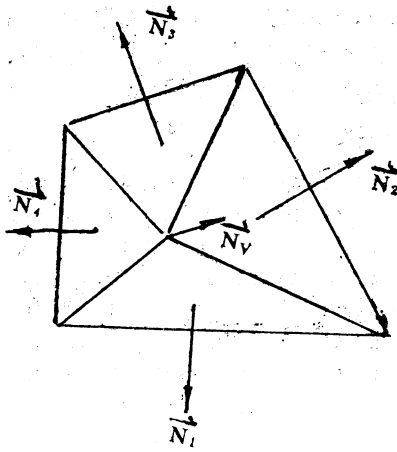


图1 顶点法向

$$\bar{N}_v = \frac{1}{4} (\bar{N}_1 + \bar{N}_2 + \bar{N}_3 + \bar{N}_4)$$

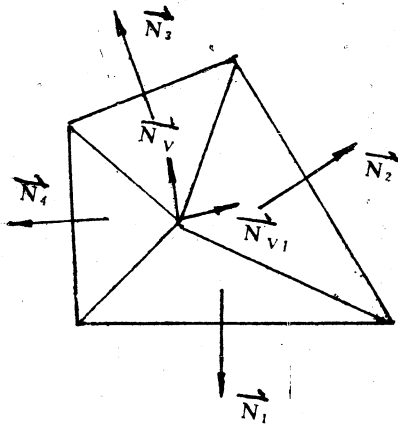


图2 顶点法向

$$\bar{N}_{v1} = \frac{1}{2} (\bar{N}_1 + \bar{N}_2)$$

$$\bar{N}_{v2} = \frac{1}{2} (\bar{N}_3 + \bar{N}_4)$$

1—4, 2—3的边界具有不连续的明暗度。

3. 由 $I = I_a K_a + I_b K_b (\bar{L} \cdot \bar{N}_v) / (R + K)$ 计算顶点的亮度。

4. 对顶点亮度值沿每一条边取线性插值, 然后又沿扫描线在各边之间作亮度的线性插值, 如图3所示。

(x_i, y_i) , $i=1, 2, 3, 4$ 为四个顶点坐标

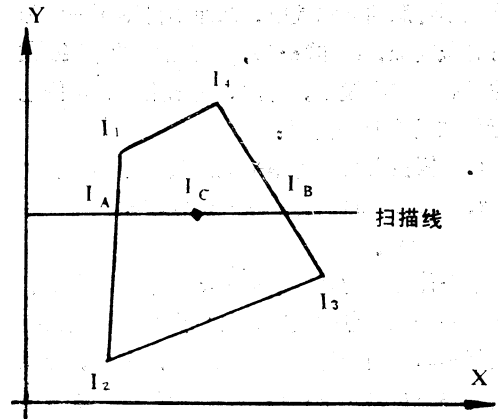


图3 亮度的线性插值

(x_i, y_i) , $i=A, B, C$ 相应为 A, B, C 三点的坐标

I_k , $k=1, 2, 3, 4, A, B, C$ 分别为相应点的亮度值

$$I_A = I_1 \frac{y_A - y_2}{y_1 - y_2} + I_2 \frac{y_1 - y_A}{y_1 - y_2}$$

$$I_B = I_4 \frac{y_A - y_3}{y_4 - y_3} + I_3 \frac{y_4 - y_A}{y_4 - y_3}$$

$$I_C = I_A \frac{x_B - x_C}{x_B - x_A} + I_B \frac{x_C - x_A}{x_B - x_A}$$

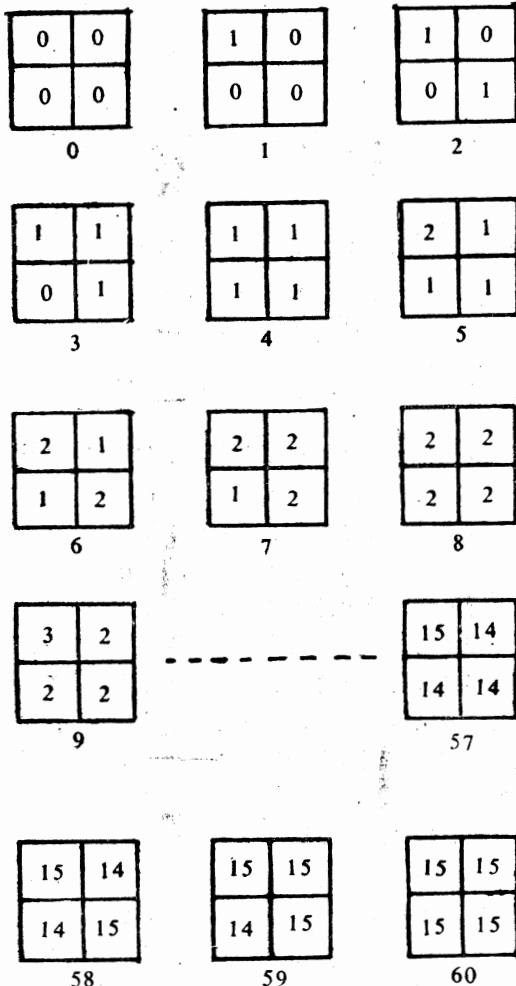
5. 对于缝合线采取降低线上各象素点的亮度值来实现。

虽然通过上述处理, 但得到的图形还是不理想, 不过这并不是理论上的问题, 而是跟实验过程中所用的显示设备有关。由于西服表面基本上是一种颜色, 在现实生活中由于各种光照效果, 使得其各点处的灰度略有差异, 要在显示设备中得到这种连续灰度的效果图显然是不可能的, 本人用4Bit的灰度级显示出的黑白图形效果很不理想, 灰度级0对应于黑色(背景色), 15对应于白色, 1—14为介于黑白之间的中间色调, 对于象人的头像等这种实际上黑白分明的图片, 4Bit的灰度分辨率或许能达到很好的显示效果, 但对于西服这种单颜色的图片, 就未免太少了。如果显示出的西服某些地方为绝对的白色(灰度级为15), 而有些地方又是绝对的黑色(灰度级为0), 显示出的图形很不柔和, 给人一种很不

顺眼很不舒服的视觉效果,从而不得不压缩实际使用的灰度级,比如 $3 \sim 10$,这显然使灰度级太少的矛盾更加尖锐了,为此,可采用两种中间色调近似法解决这一问题:

1. 牺牲空间分辨率来提高灰色分辨率。
2. 有次序的颤抖技术 (ordered dither)。

第一种方法的基本原理是用四个像素点来表示原来一个像素点的灰度,所付出的代价是沿每一坐标轴的空间分辨率降低到原来的一半,结果得到 $(16-1) \times 4 + 1 = 61$ 个灰度级,本文采用下图所示的增长序列:



这种方法的致命缺点是使图形的空间分辨率成倍地下降,因此,本文在比较这两种方法的显

示效果后,采用有次序颤抖技术作为最终解决办法。在有次序的颤抖过程中,像素点 (M, N) 是否增强其灰度取决于该点的灰度值 $I(M, N)$ 及一个 8×8 的颤抖矩阵 $A(8)$

$$A(8) = \begin{bmatrix} 7 & 0 & 32 & 8 & 40 & 2 & 34 & 10 & 42 \\ 6 & 48 & 16 & 56 & 24 & 50 & 18 & 58 & 26 \\ 5 & 12 & 44 & 4 & 36 & 14 & 46 & 6 & 38 \\ 4 & 60 & 28 & 52 & 20 & 62 & 30 & 54 & 22 \\ 3 & 3 & 35 & 11 & 43 & 1 & 33 & 9 & 41 \\ 2 & 51 & 19 & 59 & 27 & 49 & 17 & 57 & 25 \\ 1 & 15 & 47 & 7 & 39 & 13 & 45 & 5 & 37 \\ 0 & 63 & 31 & 55 & 23 & 61 & 29 & 53 & 21 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{bmatrix}$$

像素点 (M, N) 的处理方法为:

计算: $i = M \bmod (8)$

$j = N \bmod (8)$

如果: $I(M, N) > A_{ij}(8)$

则: $I(M, N) = I(M, N) + \delta$

当 $I(M, N)$ 取值 $0 \sim 60$ 时, δ 取3, 5, 7较合适,尤以 $\delta = 7$ 效果最好。

本文采用ordered dither技术的另一方面的原因是,用第一种方法显示的服装效果较差,好象是由金属片构造的,显得很生硬,跟现实生活中的服装相差悬殊,更不用说增强其真实感了;而ordered dither技术会给待显示的图形引入纹理,用〔2〕中介绍的方法选择合适的 n 值的颤抖矩阵 $A(n)$,可尽可能的减少这种纹理,微弱的纹使得显示效果更逼真,图形更柔和,具体的形成方法及各参数的选择跟所用的设备有关,在实验过程中摸索确定。

本文采用HLS (hue, lightness, saturation) 颜色模型,它是Ostwald颜色系统为基础产生一个双六面锥体子空间,如下图所示:具体使用时参考相应的使用手册即可。

结束语

服装CAD在我国还刚刚起步,特别是三维CAD还处于探索阶段,要使服装CAD达到一定的层次,本人认为显示过程中象裙子、夹克衫之类服装的皱折的显示是服装CAD的最大障碍,同时各种布料纹理及印花图案处理起来也不容

微计算机在测定岩石性能中的应用

大庆石油学院 罗云

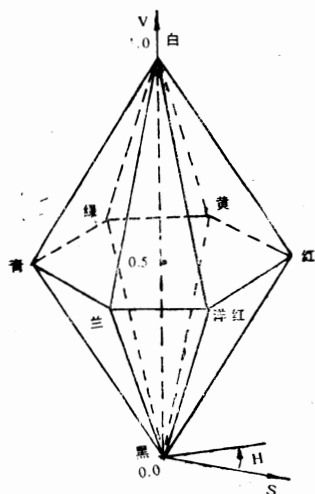
一、引言

岩石可钻性的大小是石油钻井工作者最感兴趣的问题之一,精确测定岩石可钻性,准确确定岩石可钻性大小,对于牙轮钻头的选型、钻井参数的优选,合理规定钻井生产订额,都具有重要的意义。目前石油钻井部门采用目测法测定岩石可钻性,使用华石——3型岩石可钻性测定仪测定岩石可钻性,规定微型钻头纯钻深 H 为2.4毫米,记录钻进时间 T ,由微型钻头的钻速 $V=H/T$ 确定岩石可钻性大小。微型钻头具有冲击破碎岩石的作用,测试过程中微型钻头上下跳动,显示微型钻头钻深的测深表指针左右摆动幅度大;对于抗破碎能力非常小的岩石,测深表指针转动很快,目测不能准确判断纯钻深为2.4毫米,也就不能精确测得要求纯钻深为2.4毫米的钻进时间。过去测定岩石可钻性,人们用眼睛观察钻深,用秒表记录钻进时间,用眼睛观察到的钻深 H' 实际上有的比2.4毫米大,有的比2.4毫

米小,但都认为是2.4毫米深,相应地钻进时间 T' 也增大或减小,用钻速 $V'=H/T'$ ($H=2.4$ 毫米)判断岩石可钻性大小是近似的。随着科学优化钻井、定向钻井、水平钻井技术的发展,对测定岩石性能的测试技术、测量精度要求越来越高,要求测量参数准确反映岩石可钻性的真实大小,近年来随着微计算机的广泛应用,这此要求已能实现。因此,提出岩石可钻性的微计算机自动采集处理系统,采用自动检测、微计算机采集处理技术,比原来的目测法更为科学和准确。微计算机自动采集钻深和钻时,能精确测得钻探 H 和相应的钻时 T ,用钻速 $V=H/T$ 确定岩石可钻性大小,准确反映岩石可钻性级值,为石油钻井工作者正确选择钻井参数、钻头类型等提供了可靠数据。

二、系统构成和原理

岩石可钻性的微计算机自动采集处理系统如图1所示。它由华石——3型岩石可钻性测定仪、



易,因此,服装CAD涉及面不仅很广,而且还非用CAD和图形学方面的高深理论解决不可,本人只是在探索服装显示的逼真性方面做了点工

作,显示效果还可以,达到了预期的目的,在我国服装CAD行业尚属首次,希望能起到抛砖引玉之目的。

参考文献

1. B.E.Bayer "An Optimum Method for two-level Rendition of continuous-tone pictures" INTERNATIONAL CONFERENCE RECORD ON COMMUNICATIONS 1973
2. Gouraud.H "Continuous shading curved surfaces" IEEE TRANSACTIONS ON COMPUTER Vol c-20, NO.6 JUNE 1971
3. Griffiths. J. G "A surface display algorithm" COMPUTER AIDED DESIGN Vol 10 NO.1 January 1978

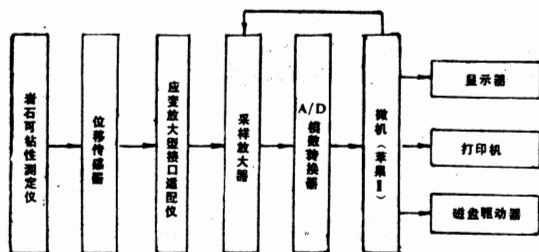


图1 系统框图

位移传感器、应变放大接口适配仪、采样放大器、A/D模数转换器、苹果Ⅱ微型计算机组成。

位移传感器安装在岩石可钻性测定仪微型钻头下部，直接检测微型钻头的钻深，将钻深转换成电信号。应变放大接口适配仪将电信号放大，采样放大器再把放大后的电信号转化为标准电平的电压量，以满足A/D模数转换器所要求的电压，A/D模数转换器把电压信号转换成数字量，输入到微计算机中进行数据处理，数据处理成二进制数后存入磁盘，待数据处理程序调用。钻时 T 是通过采样频率 F 求得，采样前进行标定时给定一个采样频率 F ，采样结束时采样总点数已知，找到纯钻深为2.4毫米的采样点数 N ，于是求得钻时 $T=N/F$ 。

三、软件设计

1. 采样程序

采样系统软件是BASIC程序和一系列汇编语言编写的子程序，采样程序流程如图2所示。用一个通道采集微钻头的钻深，微机备有32K（32768）字节的内存可供存放数据，采样容量预置大些，利用手动控制器控制采样容量，测试过程中发现岩石可钻性测定仪测深表指针指示值大于2.4毫米，按下手动控制器（与采样放大器相联）的停止按钮，立即中断采样，同时在屏幕上显示实际采集的采样点数，采集到的数据由程序处理成文件形式存盘，供数据处理程序调用。采样频率的选择应满足采样定理，即采样频率 F 应大于或等于被测信号最高频率的二倍。检测微钻头钻深的位移传感器输出信号是单极性的，且

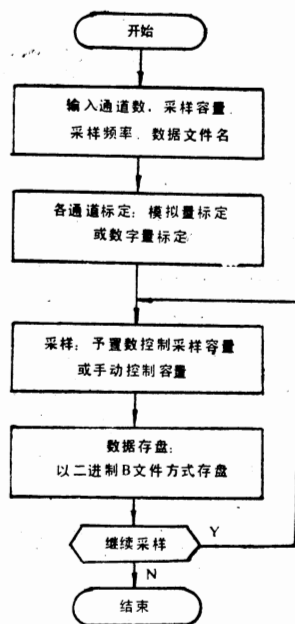


图2 采样程序流程

变化率较低，一般采样频率选取 $10\sim 50\text{Hz}$ 已满足要求，测定抗破碎能力高的岩石采样频率取较小值，反之取较大值。采样前先进行标定，标定的作用有两个，一是通过与标定信号的幅值比较，确定其它采样值的真实大小；二是确定模拟量（检测钻深的位移传感器输出信号）与实际物理量（微钻头钻深）的比例关系。在数据处理程序的计算中，都用采样所得的数字量运算，最后将结果折算成物理量输出。

2. 数据处理程序

岩石可钻性数据处理RD软件是用BASIC语言编制的，数据处理程序框图如图3所示。将某个测试孔采样所得数据逐个比较大小，判断该测试孔钻深 $h=0$ 和 $h=2.4$ 毫米的时刻，算出钻深在零到2.4毫米之间的采样点数 N ，到规定地址取出采样频率 F ，从而求得该测试孔的钻时 $t=N/F$ 和钻速 $v=h/t$ 。每个岩样求三个测试孔的钻速 v 、钻时 t 、纯钻深 h 的几何平均值作为该岩石的钻速 V 、钻时 T 、纯钻深 H 。根据钻速 V 的大小将岩样划分出类别和级别，试验数据结果和钻深与钻时曲线显示在微机屏幕上，并可打印钻深与钻时曲线、钻速 H 、钻时 T 、钻速 V 。

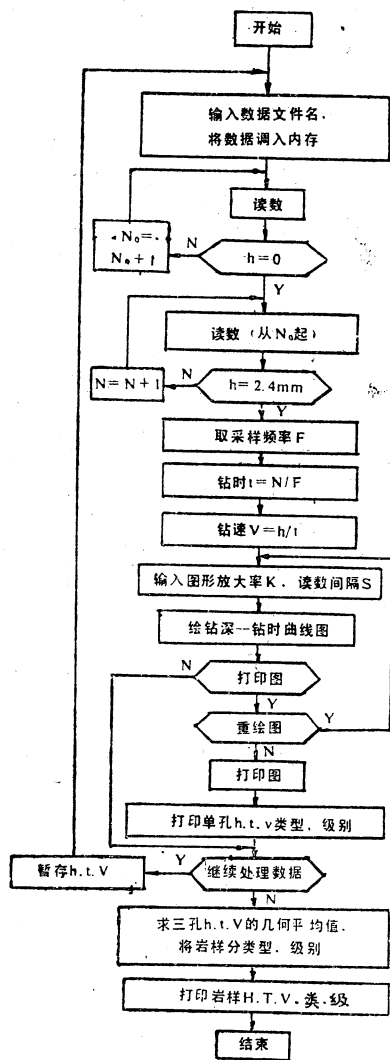


图3 数据处理程序框图

以及岩样的类别和级别。实测结果如图4所示, 由图4知, 该岩样的钻深为2.4065毫米, 钻时为176.9秒, 钻速为0.04897米/小时, 岩样属于二类七级。

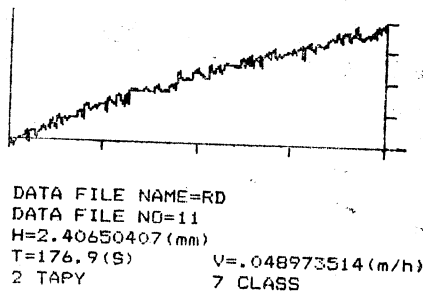


图4 实测结果

四、结束语

岩石可钻性的微机自动采集处理系统, 实现了自动采集实时处理试验数据, 显示、打印试验结果的功能。该系统的测量精度高、准确可靠, 使用效果良好, 测试结果图文并较为直观。比原来的目测法更为科学和精确, 为石油钻井系统测试岩石性能提供了科学方法和先进的测试手段。

参考文献

1. 北京师范大学计算中心编:《微型计算机实用手册》, 北京师范大学出版社出版, 1985.10.
2. 王秀玲、赵雁南、刘植楨编著:《微型计算机A/D、D/A转换接口技术及数据采集系统设计》, 清华大学出版社出版, 1984.8.
3. 北京钢铁学院王绍纯主编:《自动检测技术》, 冶金工业出版社出版, 1985.5.

DEBUG程序在故障诊断中的应用

总后勤部武汉基地指挥部自动化站 王晓武

IBM PC系列微机的各种版本DOS操作系统中, 均带有一个动态调试程序DEBUG.COM。该程序不仅是一个软件的动态调试工具, 而且还可以利用其独特的功能和丰富的命令来查找和诊断硬件系统的某些故障所在。因此, 可以说DEBUG程序同样是硬件故障的动态测试工具。

当一台微机某局部产生故障, 而没有造成整机不能运行的情况, 也就是说还能够启动运行DEBUG程序, 这时就可利用该程序, 采取软硬件结合的方法, 进一步诊断局部故障的故障点。本文将介绍利用DEBUG程序诊断故障的方法。

一、DEBUG程序的功能

在DOS手册中, DEBUG程序的功能被定义为以下三个方面:

1. 提供可控制的调试手段, 监视并控制要调试程序的执行。用户在自己的程序中直接确定问题, 然后立即执行程序, 以决定问题是否被解决。不必重新汇编程序, 便可检查自己的修改是否有效。

2. 装入、修改或显示任何文件。

3. 执行目标文件(目标程序是按机器语言格式表示的可执行文件)。

上述功能是对软件人员而言, 对于硬件维修人员来说, DEBUG程序的功能是: 提供一个诊断的测试环境, 以便检查故障所在。

二、DEBUG程序的使用

从诊断故障的角度, DEBUG程序的用法有两种情况, 一种是进入DEBUG程序后, 利用其命令进行简单的诊断; 另一种情况是用DEBUG程序调用已编好的维修小程序。其方法是:

1. 直接启动法

C>DEBUG

2. 调用法

C>DEBUG RAMTEST.COM

这里的RAMTEST.COM是一个测试内存的维修小程序。这种方法就是将RAMTEST.COM程序装入内存使用。

当启动DEBUG程序时, 要对调试的程序设置寄存器和标志位, 其设置内容如下:

1. 设置分段寄存器(CS, DS, ES, SS)为跟在DEBUG程序末的第一段。

2. 设置指令指针(IP)为X'0100'。

3. 设置堆栈指针(SP)为这个分段的末端。

4. 设置其它的寄存器(AX, BX, CX, DX, BP, SI和DI)均为0, 当用调用法启动后, BX: CX应为调入文件的长度。

5. 设置标志寄存器为清除的值:

NV UP EI PL NZ NA PO NC

6. 缺省磁盘传输地址设在代码分段X'80'。

三、DEBUG程序的命令

DEBUG程序共有18条命令, 每条命令都有

其不同的用途及格式, 下表详细介绍每条命令的用途及格式:

DEBUG程序命令表

命令	用途	格式
Assemble汇编	汇编语句	A [address]
Compare比较	比较存储器内容	C range address
Dump卸出	显示存储器内容	D [address]或D [range]
Edit进入	修改存储器内容	E address [list]
Fill填充	修改存储器块	F range List
Go运行	可选断点运行	G [-address] [address]
Hex十六进制	计算十六进制和与差	H Value Value
Input输入	从指定端口输入并显示	I port address
Load装入	装入文件或对扇区	L [add drive Sect Sector]
Move移动	移动存储器内容	M range address
Name命令	定义文件和参数	N filename [filename]
Output输出	向指定的端口输出字节	O port address Byte
Quit退出	结束DEBUG程序	Q
Register寄存器	显示寄存器/标志位	R Register name value
Search搜索	搜索字符或字符串	S range List
Trace跟踪	执行并显示	T [-address] [value]
Unassemble反汇编	对指令进行反汇编	U [address] [range]
Write写入	写入文件或扇区	W [add drive Sect Sector]

上述命令从诊断维修的角度来看, 从功能上可以将它们分成以下几类:

1. 简易诊断类。

对寄存器的检查命令。

对I/O口的检查命令。

对磁盘存储器的检查命令。

2. 文件编辑类。

包块加载文件, 编辑文件与存贮文件。

3. 执行文件类。

四、诊断故障的方法

应用DEBUG程序诊断故障的方法, 在此分两方面介绍, 一方面是一般诊断, 就是用命令直接诊断局部子系统; 另一方面是用自编的小汇编程序结合必要的维修设备诊断机器故障。

1. 一般诊断

当机器局部子系统有故障时, 可以用DEBUG程序的有关命令直接进行诊断。

①对通用寄存器进行读写和运算检查

使用R命令

键入R命令可以读出所有通用寄存器的内容和标志寄存器的内容, 显示信息如下:

```

AX=XXXX    BX=XXXX    CX=XXXX    DX=XXXX
SP=XXXX    BP=XXXX    SI=XXXX    DI=XXXX
DS=XXXX    ES=XXXX    CS=XXXX    SS=XXXX
IP=XXXX NV UP DI NG NZ AC PE NC
XXXX: XXXX  CD21  INT 21

```

以上前面显示的四行是从通用寄存器读出的16进制内容和标志寄存器读出的八个标志位。后面一行是一条指令，指令地址为CS: IP所指向的地址。

标志寄存器的八个标志位表示的意义如下：

标志位名称	置入	清除
溢出(是/不是)	OV	NV
方向(增加/减少)	DN	UP
中断(允许/禁止)	EI	DI
符号(负/正)	NG	PL
零标志(是/不是)	ZR	NZ
辅助进位(是/不是)	AC	NA
奇偶校验(偶校验/奇校验)	PE	PO
进位(是/不是)	CY	NC

通过键入R，接着键入某寄存器名称或标志寄存器名F，可读出相应寄存器和标志寄存器的内容；此时，还可进行写操作，键入适当内容后回车，写入操作完成了，并可以再次读出来检查正确与否。

例如：

```

R AX          ; 读出AX
AX AAC4
: -4CAB       ; 修改AX
R AX          ; 再读AX
AX 4CAB       ; 验证是否刚写入的内容，

```

例如：

```

RF
OV DN EI NG ZR AC PE CY
: -PO NZ DI NV
RF
NV DN DI NG NZ AC PO CY

```

说明已修改了四个标志位。

以上方法可以检查每个寄存器的读写功能和标志寄存器的读写功能。

②检查存储器读写、地址译码及数据传输功

能。

采用A、E、F（写内存）D、U（读内存）C（比较）M（移动）命令

例如：固化BASIC的EPROM四片2764中若有一片有故障，可用读内存命令去检查。也可以和标准ROM相比较。

U F600: 0 可显示固化BASIC程序清单，
D F600: 0 可显示固化BASIC程序数据。

把标准组件置F000，被测件置F800。可用比较法测试：

C F600: 0 L2000 2000

上述命令意味着从F600: 0起始和F600: 2000起始共200个字节进行比较。但是要注意，先把F600置入CS寄存器。

对于RAM检查，可写、可读、移动、比较。当扩充内存产生故障，可充分利用RAM检查手段来检测故障。扩充内存一般从256KB开始，当引导DOS时出现“扩充内存奇偶错”，便不能执行下去。这时需要修理扩充内存板，如果只有一台机器该如何处理呢？

我们知道基本内存和扩充内存奇偶错都会引起无条件中断，而NMI的中断处理程序显示错误信息后便停机处理，所以必须不使其产生NMI（可把8088的17端断开），扩充内存奇偶错将不再影响引导DOS。此时，就可以在DEBUG控制下对内存4000段以上的单元进行检查，使故障能准确定位。

F 4000: 0 LFFFF AAAA ; 写入4000段64K单元。

D 4000: 0 FFFF ; 读出

C 4000: 0 8000 4000: 8000 ; 64K对半比较。

64KB内容较前后32KB对半比较；找出地址，再用D命令确定其是哪位或几位错。有了具体地址，便可以在此地址内利用命令写入各种测试码，进一步定位。

E DS: n

4000: n 0, 55 ; 以55码换00码

D DS: n 1

4000: n 55 ; 读出一个单元

其中n为假定出错单元的地址偏移量。

③对I/O端口的诊断方法

I/O端口是CPU和外部设备进行信息交换的通道,为了使数据、控制命令以及I/O设备的状态信息便于区分,则安排了不同的地址端口,这些地址端口有的是可读可写的,有的仅为可写或可读的,利用I/O端口的读写功能,我们可以对其进行检测。

例如:打印机数据I/O口为378,它是可读写的。

```
O 378 FF; 送378口FF码
I 378      ; 读出的是FF
-EE        ; 可以换几组码试试。
```

打印机状态I/O口为379,它是可读的,可查打印机的状态。

```
I 379      ; 读出375口状态
-XX
```

379口读出的状态信息为:

```
bit7=BUIY (忙), bit6=ACK (应答), bit5=
=PEN
D (纸尽), bit4=SETECT (选择), bit3=
ERROR (错误)。
```

其它的I/O口地址可以按上述方法检测。

④外部存储器一般诊断

DEBUG程序支持读盘命令L和写盘命令W,利用这两个命令可以检测磁盘的读写功能。

读盘命令格式如下:

```
L [(地址) [(盘号)(相对扇区)(扇区数)]]
```

可以实现从盘的指定扇区装入信息到内存。

其中,地址指的是内存DS:XXXX,盘号0为A盘、1为B盘、2为硬盘C驱动器。

为了检查A驱动器文件目录表,可以打入如下命令:

```
L 100 0 5 7; 从A盘5扇区起读7扇区
D 100      ; 调入内存,显示内容
```

若读盘成功,可用D命令显示,读盘不成功,显示错误信息,此时,可以去显示查看40:41中的出错情况和40:42~48中的出错内容,以便分析原因。

读盘时,有时读一个扇区是成功的,读多个扇区失败,说明数据或数据窗口波形不十分稳定。有时读0面是成功的,读1面是失败的,说明磁头定位偏移。

写磁盘命令格式如下:

```
W [(地址) [(盘号)(相对扇区)(扇区数)]]
```

参数的定义与L命令相同,写入磁盘内容正确与否,可用L命令来检验。

现举例如下:

要求把内容存DS:1000起100个字节内容写入硬盘10扇区,然后读回到内存DS:2000,再与DS:1000进行比较,可打入下面三条命令:

```
W DS:1000 2 10 1
L DS:2000 2 10 1
C 1000 100 2000
```

2. 简单诊断程序的编写与使用

针对具体的一个故障编写一些简单短小、有助于检查故障的有效程序。通常用于检查故障之前或故障发因之后。编写简单诊断程序要掌握两个原则,一是短小,这就必须针时性强;二是利用循环指令,进行周期性测试,这样可以区分偶然性故障和固定性故障。循环测试为借助示波器等设备测试被测点波形提供了有利条件。这种小程序,应能熟练的随手编写,它不同于专用功能检查模块的编写,那样的程序多数采用宏汇编去编写。

在DEBUG中,编辑小程序使用A命令,然后用U命令反汇编检查,程序中个别参数的修常用E命令,程序中某些指令修改常用A命令。

编辑完一个小程序,往往写完后跟踪运行,则不必存储起来,这时可用T命令安排第一指令,然后用G命令执行。

如果存储起来,则必须用N命令起名。根据文件长度修改CX值,再用W命令存储起来。

下面举几个小实例:

①检查打印机口

```
DY: MOV DX, 378 ; DY、DY1标号是为转
DY1: MOV AX, AA55 ; 移指令所设,实际
      OUT DX, AL ; 上是绝对地址的代号
      MOV AL, AH
      OUT DX, AL
      JMP DY1
```

运行此程序,检查数据口LS374输出应为方波。

此程序中DY条指令中378换成37A则检查命令口74LS174和74LS05应为方波。

②扩充内存出错检查

检查74LS138译码器, 可编如下小程序:

```
TEST: MOV AX, 4000
      MOV DS, AX
      MOV CX, 2000
      MOV SI, 0
      MOV AL, AA
WRITE: DS,
      MOV [SI], AL
      INC SI
      LOOP WRITE
```

③检查只读存储器ROM

检测ROM, 可检查其代码和是否正确, 编小程序如下:

```
      MOV CX, 2000
      MOV AX, E000
      MOV AL, 0
ROMSUM:
      ADD AL, [BX]
      INC BX
      LOOP ROMSUM
      OR AL, AL
      INT 3
```

当这个程序运行结束, 可检查AL是否为0, 若为0, 则认为正确, 否则有错。

④检查显示器控制板

检查显示器控制板状态寄存器第三位(彩色显示器垂直同步信号)有无变化, 编写以下小程序:

```
      MOV DX, 3DA
      MOV AH, 08 ; 状态口第三位高
      SUB CX, CX
TEST1: IN AL, DX
      CMP AL, AH
      JNI TEST2 ; 为低转测变高
      LOOP TEST1 ; 为高继续查直到出现低
TEST2: SUB CX, CX
TEST3: IN AL, DX
      CMP AL, AH
      JZ STOP ; 为高, 转停机
      LOOP TEST3 ; 为低直到出现高
STOP: INT 3
```

⑤软盘读写检查

软盘读写检查要调用BIOS的INT13中断程序, 在进入INT13以前, 必须准备各种参数, 这些参数的定义如下:

入口参数:

AH为子功能命令字节,

AH=0 复位磁盘系统

AH=2 读磁盘扇区到内存

AH=3 内存数据写入磁盘

AH=4 校验磁盘扇区内容

AH=5 格式化磁道

AL=扇区数(最大为9, 格式化时不用)

CH=磁道号(0~39)

CL=扇区号(1~9)

DH=磁头号(0或1)

DL=驱动器号(0~3)

ES: BX为内存缓冲区的首址的段与偏移量。

出口参数:

AH=软盘操作出错状态标志(送40: 41单元)

CY=0操作成功, CY=1失败, 操作结果参数存入40: 42~48单元中(这些单元各位状态含义参见硬件技术手册)。

AL=实际读到的扇区数

举例: 复位功能检测, 可编程如下:

```
START: MOV AH, 0
      MOV DI, 0; 驱动器A
      INT 13
STOP: JMP START (INT 3)
```

执行到STOP时, 若是不循环, 改为INT3, 然后从40: 41~48中去检查出错原因。若是循环, 可用示波器测量UPD765的RST复位端和中断清示RQ6信号是否正确。

40: 41单元内容, 其含义如下:

01=I/O命令非法, 即入口AH大于5。

02=地址码错, 找不到标识符

03=写保护

04=找不到扇区

08=FDC的DMA超时

09=DMAC的DMA超时

10=CRC校验错

20=FDC失败, 包括主状态寄存器出错, 命令异常结束等。

40=寻道失败

利用8279芯片实现多路D/A转换

安徽机电学院电气系 周鸣争

一、引言

在大多数微型计算机控制系统中,通常要求有多路的模拟量输出,以使一台主机同时控制几个回路。本文针对这一问题提出了一种只使用一片D/A芯片,在8279芯片的控制下,可进行十六路模拟量输出的方法。这种方法既克服了以往对多路D/A转换每路均使用一片D/A芯片而造成芯片过多、成本高的缺点,又克服了采用多路保持器而需定时扫描刷新程序支持的缺点。该方法与CPU的接口规范,连接简单,占用CPU的工作时间少。同时由于对各路D/A输出的定时刷新过程由硬件自动完成,因此,编程方便,输出稳定,具有一定的实用性。

二、硬件结构及工作原理

利用8279芯片实现多路D/A转换电路如图1所示。图中8279芯片为一通用可编程键盘/显示器接口芯片,它内部配备有 16×8 的显示RAM单元。CPU对此16个字节的RAM可进行读/写操作。而RAM各单元的内容则通过输出端 $A_0 \sim A_3$ 和 $B_0 \sim B_3$ 端输出。在图中我们使用8279内部

的这16个单元,分别存放16路需转换的数字量,CPU可通过对8279写入适当的命令字,令其工作在每次读/写后显示器地址自动加一方式,使16显示RAM单元中的内容以一定的频率循环地由 $A_0 \sim A_3$ 和 $B_0 \sim B_3$ 端输出,作为D/A芯片的数字量输入。同时,通过8279命令字的控制,使8279内扫描计数器工作于编码方式。此时,四根扫描输出线($SL_0 \sim SL_3$)的输出信号为未经译码的二进制码,作为多路模拟开关的选择信号。扫描计数器的四位扫描输出与16个显示RAM单元相对应,且在时间上严格同步。当16个显示RAM单元中某一单元的内容(即某路需转换的数字量)由 $A_0 \sim A_3$ 和 $B_0 \sim B_3$ 端输出时,该数字量经DAC82转换成模拟量后,作为多路模拟量开关AD7506的输入。此时,四根扫描输出线($SL_0 \sim SL_3$)的输出为相应单元的编号,作为AD7506的选择信号,使相应的通道选通,DAC82输出的模拟量将加至该通道的保持电容上,经运放后输出。

8279的消隐输出信号 \overline{BD} 与AD7506的 \overline{EN} 端相接,当CPU向8279内16个RAM单元中写入新

80=FDC超时,包括主状态寄存器,中断超时。

磁盘控制线路复杂,仅靠使磁盘进行简单动作后检查40:41~48单元的内容是远远不够的。

另外,编写小程序时,仍然有技巧,例如:上面举复位磁盘的小程序,若把两种情况都考虑到,可借助于键盘中断进行人工干预。

```
START: MOV AH, 0
        MOV DL, 0
        INT 13
        MOV AH, 01
        INT 16 ; 按一键则程序停止
        JNZ START; 否则循环。
STOP: INT 3
```

关于编程后程序执行或调入程序执行,都有一个起始点问题。DEBUG调入程序必须放置在内存CS:100处,因此从此为起始地址执行。但是刚刚编辑完的程序,其IP值为刚编完的地址,一种方法用T命令:T=首址,一种方法用R命令修改IP值:R IP,填入首址,然后执行T或G命令。

结束语

应用DEBUG程序,以软硬兼施的方法查找和排除机器故障,可以避免修理过程中的许多不必要的环节,提高维修效率,许多常见故障采用本文所叙述的方法,可以事半功倍,用户不妨一试。

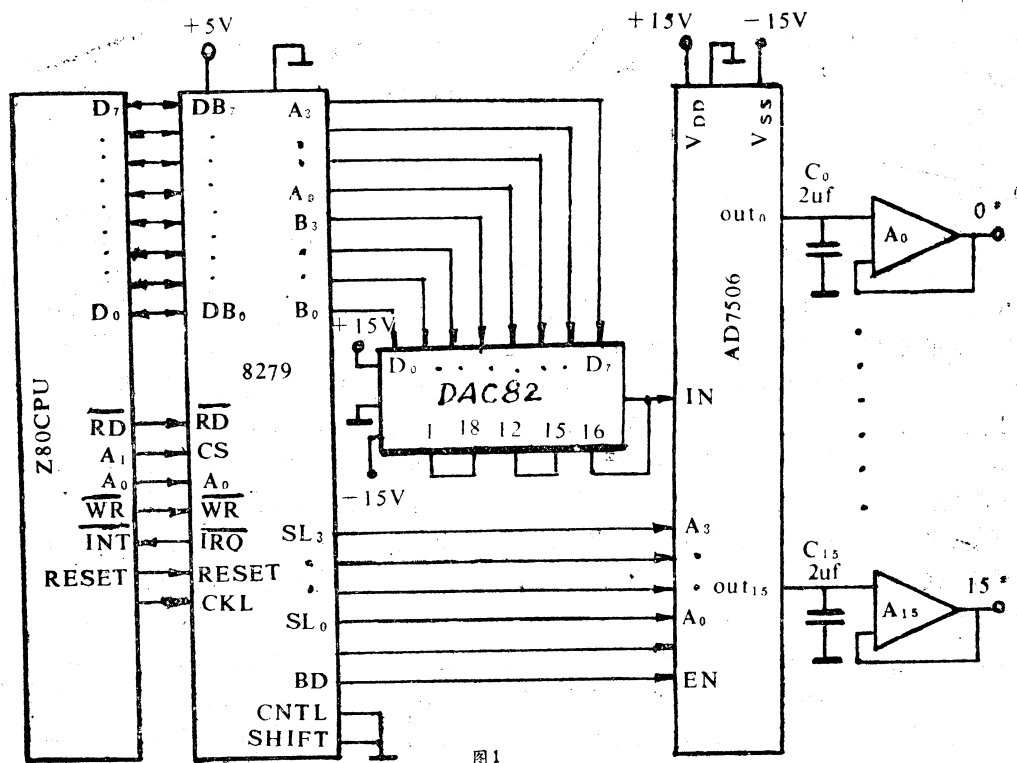


图1

的数字量时, \overline{BD} 有效, 使在数字切换过程中禁止 AD7506 工作, 避免输出出现“毛刺”现象。

8279 具有可编程的时钟, 可根据系统的不同要求通过预置内部定时器的初值不同来改变对各路 D/A 输出刷新周期的大小, 使输出稳定、以满足各种情况下系统的需要。

三、结束语

综上所述, 采用图 1 所示的电路实现多路 D/A 转换, 具有硬件省、连接方便等优点。同时由于 8279 为一标准接口芯片, 它与 CPU 的接口规范。编写其多路 D/A 转换的控制程序十分简单,

完全与 8279 的编程要求一致, 具体过程可参见文献 [1], 这里不再赘述。图一的电路是根据 8279 与 Z-80 CPU 接口而设计的。只要稍加改进, 与其它类型的 CPU 以及单片机的接口亦很方便。能满足大多数过程控制系统的需要。

参考文献

- [1] 张善德编者: 《微型计算机系统的设计方法和接口技术》, 人民邮电出版社 1985 年
- [2] 周明德年编著: 《微型计算机接口电路及应用》, 清华大学出版社 1987 年

速达微机系统板的维修

甘肃省计算中心 刘 洽

速达 (SUPETECH) 微机, 是与 IBM PC/XT 微机兼容的。该机出现的故障是: 打开显示器时, 显示屏幕上有光栅出现。接着打开主机电源后, 屏幕无光标, 但屏幕左上角出现

“02”的字样; 扬声器无声音; 软、硬盘驱动器的指示灯全不亮。

分析与处理: 在微机加电后, 电源就绪信号 PWRGOOD 经时钟发生器 8284A, 向处理器

8088 输出一个复单位脉冲信号 RESET, 直接送到 8088 芯片的 RESET 脚上, 使 8088 处于初始化状态。这时, 8088 内部硬件逻辑, 就置 8088 的代码寄存器值为 FFFFH, 并置指令寄存器的值为 0000H。于是, CPU 便从 FFFF0H 单元开始执行程序, 即自动地到了 FFFF: 0000H 处, 取出第一条指令: JMP F000: E05B。从而转移到 POST 程序的入口处: F000: B05B, 进入加电自诊断测试程序 POST。因为, POST 程序是 ROMBIOS 程序的主要部分。在自诊断测试过程中, 若以显示屏幕上是否出现光标为分界线, 可将 IBMPC/XT 微机的故障, 分成严重性错误和非严重性错误。严重性错误包括: CPU8088 的内部出错 (涉及系统总线出错); 8KB ROM BIOS 代码和出错; 计数器 8253 计数功能出错; DMA8237 芯片无法进行动态刷新控制; 前基本 16KB 内存读、写出错等 (非严重性错误; 与处理本故障无关, 故不在此处罗列)。

当微机在自诊断过程中, 出现了严重性错误, 则只能转入停机 (HLT)。如何处理此类故障呢? 我们的处理方法是: 使用示波器或逻辑测试笔, 测出 8255A 的 PA 端口标志值。因为, POST 在对关键部件的自诊断过程中, 把并行接口器件 8255A 的 PA 端口, 暂时作为诊断检测标志出口, 而 8253 和 8237 出错, 转入停机后, 都问 8255A 的 PA 端口, 输出检测标志。下表为严重性错误测试标志。(见右上)

由测得的 PA 端口标志值, 结合故障现象,

测量点	停机地址	出错原因
PA=01	E0AD	8088 处理器, 8KB ROM BIOS 代码和出错,
PA=02	E0F8 E100	8253——1 st 计数出错,
PA=03	E12D E15C	DMA8237 地址和计数出错。 8237—0 th 刷新周期不正常。
PA=04(被)	E184~E18C	基本 16KB 出错。死循环, PA ₂ =1 和 PA ₇₋₀ =RAM 出错, 交替方波。

来判断故障所在部件或其所在点。

根据测得 PA 标志值 “02 (这与屏幕上出现的字样 “02” 相符), 及其严重性错误的出错原因而判断, 估计故障的所在部件, 可能为 8253 芯片。我们知道: 8253 芯片的三个独立定时器的时钟信号, 都是来自 8284 芯片的第 2 脚 PCLK。但 PCLK 是经 U₁₂ (M74LS175P), 向 8253 提供 1.19MHz 的时钟信号。所以, 应从 8253 芯片的输入来着手检查。经检查发现 8253 芯片的输入脚 9、15 和 18, 均无时钟信号输入。因此, 顺藤摸瓜往前查找, 检查 LS175 芯片的输出脚 10 和输入脚 9。则既无输入也更无输出了。但是, 查 8284 芯片的第 2 脚有输出信号。那么问题就出在 8284 芯片的第 2 脚与 LS175 芯片的第 9 脚之间的印刷线路上。经查实, 果然是它们之间的印刷线路断裂, 且断裂点恰好在 8284 芯片下面, 若不细致和耐心, 难以查找和发现。经过重新焊接, 连接好此线之后, 加电启动, 故障即被解除, 系统畅通无阻, 运行正常。

APPLE II 微型机外中断源的扩展方法

中国科技大学四系微机室 庞文宁 王砚方

一、引言

微型计算机于用测量控制系统时, 往往要求具有优良的中断系统, 以提高 CPU 对外部异步事件的处理能力和响应速度。由于 APPLE II 系统拥有众多的软件和外部设备扩充卡以及很高的性能价格比, 所以在我国有许多 APPLE II 微型机应用于测量和过程控制领域。但是, APPLE II

微型机的 6502CPU 仅为用户提供了可展蔽的外中断源 (相当于 Z80CPU 的可屏蔽中断方式 1), 同时 6502 指令系统中的软件中断指令 BRK 与可屏蔽中断 $\overline{\text{IRQ}}$ 共用一个中断向量, 和 Z80, 8080 比较, 6502 由于硬件的限制, 中断方式不如前者灵活。在某些稍复杂的测量控制系统中, 往往会有多个中断源的中断请求同时出现, 而且这些中断请

求对其响应的紧迫程度也各不相同,为此,需进一步扩展。本文所提供的方法可将6502CPU的可屏蔽中断源($\overline{\text{IRQ}}$)扩展到8个或8个以上,并且,电路稍加改进,在软件上作些相应的处理,可实现8级中断嵌套。

二、硬件电路

中断优先级排队由硬件和软件来实现,使用

优先权编码器将每个中断源的中断请求转换成相应的优先权代码。这里我们采用74LS148 优先权编码器和74LS373 锁存器来进行此项工作。如图1,图2所示。

图1中74LS148的8个输入端0~7用作8个外中断源的输入端。3位优先权编码输出移A, B, C经锁存器连至6502CPU的数据总线 $D_0, D_1,$

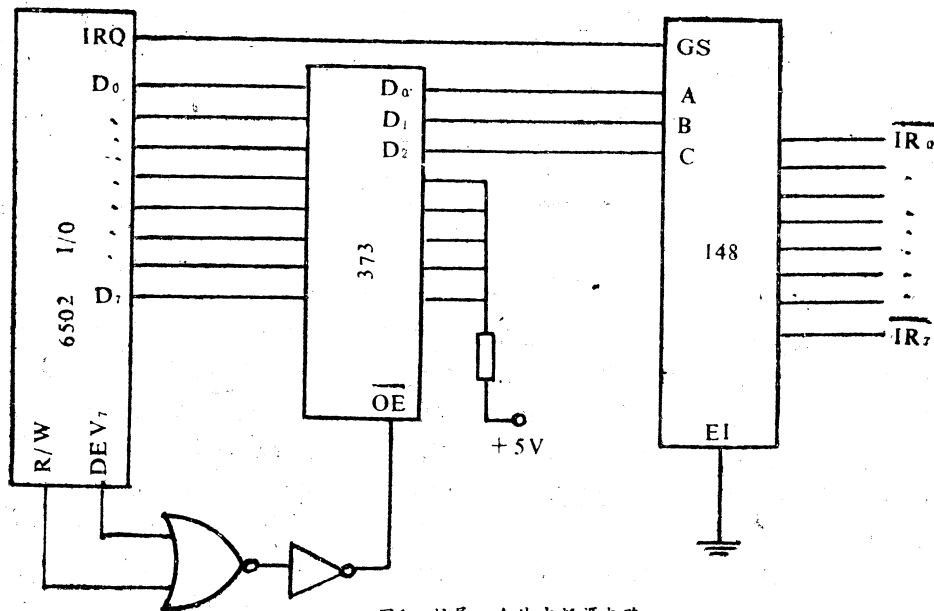


图1 扩展八个外中断源电路

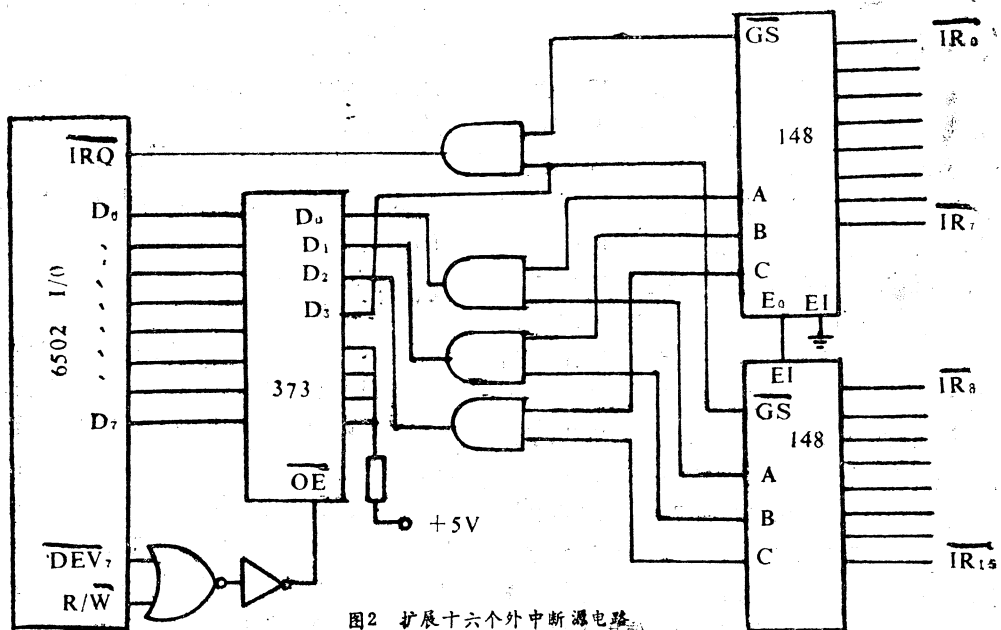


图2 扩展十六个外中断源电路

D₂, 编码群输出端GS和6502CPU外中断源 $\overline{\text{IRQ}}$ 相连。74LS148的0~7个输入端, 输入端7具有

最高优先权, 而输入端0的优先权最低。它们所对应的编码由表1决定。

表1 74LS148 真值表

输 入									输 出				
EI	0	1	2	3	4	5	6	7	C	B	A	GS	FO
H	X	X	X	X	X	X	X	X	H	H	H	H	H
L	H	H	H	H	H	H	H	L	H	H	H	L	L
L	X	X	X	X	X	X	X	L	L	L	L	L	H
L	X	X	X	X	X	X	L	H	L	L	H	L	H
L	X	X	X	X	X	L	H	H	L	H	L	L	H
L	X	X	X	X	L	H	H	H	L	H	H	L	H
L	X	X	X	L	H	H	H	H	H	L	L	L	H
L	X	X	L	H	H	H	H	H	H	L	H	L	H
L	X	L	H	H	H	H	H	H	H	H	L	L	H
L	L	H	H	H	H	H	H	H	H	H	H	L	H

由74LS148的真值表可知, 在该集成电路的输入使能端EI为低电平的前提下, 只要其8个输入端中的任意一个输入为低电平, 将使6502CPU的 $\overline{\text{IRQ}}$ 引线变低; 同时74LS148的输出端将会有相应的三位二进制编码输出。当同时有多个中断源提出中断申请时, 74LS148将按其优先权大的那个输入而决定其代码。这时, 若6502的 $\overline{\text{IRQ}}$ 中断开放, 就可以响应中断源所提出的中断申请。

三、软件编制

众所周知, APPLE II机的6502CPU中断服务程序的入口地址放在\$FFFE及\$FFFF两个单元中。而在APPLE II机上, 这个区域是监控ROM, 已填好入口地址为\$FA40, \$FA40的中断子程序的功能则为判断中断是软件中断(BRK指令)还是外设硬件中断, 如果是硬件中断, 就转到用户中断服务程序入口, 该入口地址存放\$3FE和\$3FF单元中(前者放低八位, 后步放高八位)。用户写好了自己的中断服址子程序后, 在这两个单元中填入入口地址就可以了。但是, 由于6502CPU指令系统的限制, 为了使程序转向各中断源的中断服务程序, 必须在6502CPU的中断服务程序中编写一段引导程序。可参见图3。该引导程序的指导思想: 先将优先权编码表放入0页的几个可供用户使用的单元,

该程序执行时, 先从端口\$CFOFO读入外中断源的编码, 与0页存放的编码表逐一比较, 确定该中断源并转入相应的入口, 执行中断服务子程序。另外, 由于APPLE II监控程序存在一个问题: 即APPLE II在响应中断请求 $\overline{\text{IRQ}}$ 后, 先执行系统监控程序中的 $\overline{\text{IRQ}}$ 子程序, 虽然 $\overline{\text{IRQ}}$ 子程序首先把中断发生时A寄存器的内容存放在\$45单元, 但是在跳出 $\overline{\text{IRQ}}$ 子程序时, 并没有恢复A寄存器在中断发生时的真实内容。故在使用APPLE II的 $\overline{\text{IRQ}}$ 中断方式时, 应对此加以特别注意, 否则很可能出现一些无法预料的错误, 导致程序运行失败。

表3 引导程序清单(见下页)

由于篇幅关系, 有关八级中断嵌套程序清单省略。

四、结束语

本文所讨论的方法以电路简单, 价格低廉为特点。但不足之处在于中断响应时间较长, 由于APPLE II机的6502CPU在真正为所扩展的中断源服务之前需执行一段引导程序来确定为某一中断源服务。虽然在过程控制中使用APPLE II的 $\overline{\text{IRQ}}$ 中断方式时略感不便, 但只要采取相应的措施, 还是能够在过程控制中正确地使用APPLE II的 $\overline{\text{IRQ}}$ 中断方式。

TP801-A单板机EPROM编程器的简单制作

江汉石油学院微机教研室 李华贵

随看计算机技术的迅速发展,大中型控制设备以及智能仪器一般采用2K~16K字节的EPROM存放系统程序,拷贝系统程序作备件是保证维修的关键问题,对于一些引进设备上EPROM的备用更显得重要。鉴于微机以及微机写入器价格昂贵,而且拷贝EPROM不方便,目前流行的DVCC—51—I开发型单片单板机也不可能固化EPROM27128组件,因此,选用非常普遍的TP801-A单板机制作EPROM写入器,整个设备成本低,制作简单,使用灵活,效果良好。

1. 方法简介

将单板机上原有编程电路稍加改动,在布线区增加一片74LS139、一片74LS10、24芯与28芯插座各两个等元件,把编程程序预先固化在一片2716中,且插入原单板机的PROM1处,便组成了通用EPROM编程器。

2. 功能

①在原PROM2处仍然可以对EROM2716进行固化,可以对2732、2764、27128拷贝。

②可以将单板机上任一内存区域(RAM或PROM)内所期望的内容写入指定的EPROM

```
SOURCE FILE: P41
----- NEXT OBJECT FILE NAME IS P41.OBJO
0300:          1      ORG      $300
0300:A5 45      2 INTO     LDA      $45
0302:48        3          PHA
0303:8A        4          TXA
0304:48        5          PHA
0305:98        6          TYA
0306:48        7          PHA
0307:AD F0 C0   8          LDA      $C0F0
030A:C5 09      9          CMP      $09
030C:F0 21     10         BEQ      IR0
030E:90 0D     11         BCC      INT1
0310:C5 1A     12         CMP      $1A
0312:F0 21     13         BEQ      IRF
0314:90 1C     14         BCC      IRE
0316:C5 1C     15         CMP      $1C
0318:F0 21     16         BEQ      IRH
031A:4C 38 03   17         JMP      IR0
031D:C5 07     18 INT1     CMP      $07
031F:F0 08     19         BEQ      IRB
0321:90 03     20         BCC      IRA
0323:4C 2C 03   21         JMP      IRC
0326:4C 3E 03   22 IRA     JMP      INT2
0329:4C 3E 03   23 IRB     JMP      INT2
032C:4C 3E 03   24 IRC     JMP      INT2
032F:4C 3E 03   25 IRD     JMP      INT2
0332:4C 3E 03   26 IRE     JMP      INT2
0335:4C 3E 03   27 IRF     JMP      INT2
0338:4C 3E 03   28 IRG     JMP      INT2
033B:4C 3E 03   29 IRH     JMP      INT2
033E:68        30 INT2     PLA
033F:A8        31          TAY
0340:68        32          PLA
0341:AA        33          TAX
0342:68        34          PLA
0343:40        35          RTI
```

*** SUCCESSFUL ASSEMBLY: NO ERRORS

参 考 文 献

1. 《微处理器接口技术》, 国防工业出版社

广告征集启事

本刊为公开发行的技术类双月刊。为使科技、市场、资金、资源、产品相互配套,有机结合,产生效益,本刊愿为社会各界各行各业牵线搭桥。为此,本刊特辟一定版面刊登计算机软、硬件产品及各类电子产品、科技成果、书讯、培训等广告信息,并开展有关服务。

本刊拥有经营广告业务的正式营业执照,新的广告经营许可证号码为川蓉工商广字005号。热忱欢迎社会各界来函来人来电联系广告信息业务,本刊愿提供优惠优质服务。

《计算机应用研究》杂志社

图3 引导程序清单

中, 字节长度可以预置。

⑧根据不同EPROM 组件的编程时间, 预置定时时间常数, 保证可靠写入的前提下尽量减少写入时间。

④可以事先检查EPROM 是否擦除干净。

⑤在固化过程中自动检查并指出可能产生的错误。

二、硬件制作

1. 片选信号的产生

电路如图1所示, 用一片74LS139 双二—四线译码器译出4个16K地址范围的片选信号($\overline{1Y_0}$ 、 $\overline{1Y_1}$ 、 $\overline{1Y_2}$ 、 $\overline{1Y_3}$), 其中, $\overline{1Y_0}$ (0000H~3FFFH) 为原单板机所占用, 在此不能用, $\overline{1Y_1}$ (4000H~7FFFH) 作源区27128或2764的片选信号, $\overline{1Y_2}$ (8000H~BFFFH) 作目的区27128或2764的片选信号, 对 $\overline{1Y_3}$ 再次译码, 产生4个4K地址范围的片选信号, 其中, $\overline{2Y_0}$ (C000H~CFFFH) 作源区2732或2716的片选信号, $\overline{2Y_1}$ (D000H~DFFFH) 作目的区2732的片选信号, $\overline{2Y_2}$ 和 $\overline{2Y_3}$ 备用。

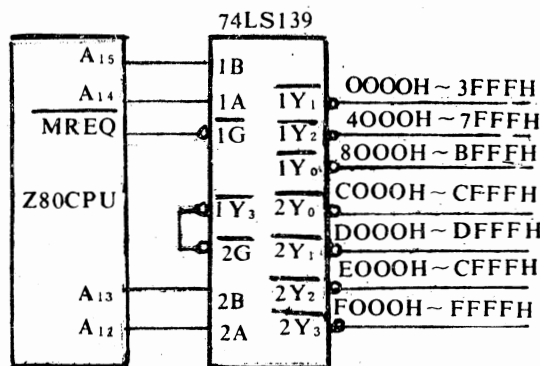


图1. 产生片选的信号电路

表1

引脚方式	片选端 CE(20)	输出允许端 OE(22)	编程端 PGM(27)	A ₉ (24)	V _{pp} (1)	V _{cc} (28)	输出
读	L	L	H	X	V _{cc}	V _{cc}	数据输出
保持	H	X	X	X	V _{cc}	V _{cc}	高阻
编程	L	H	L	X	V _{pp}	V _{cc}	数据输入

2. EPROM27128的介绍

在四种EPROM 组件中, 27128 控制端较多, 以其为例, 分析改制电路的原理与方法, 其它几种的联接, 就不赘述。EPROM27128的28根引脚如图2所示。

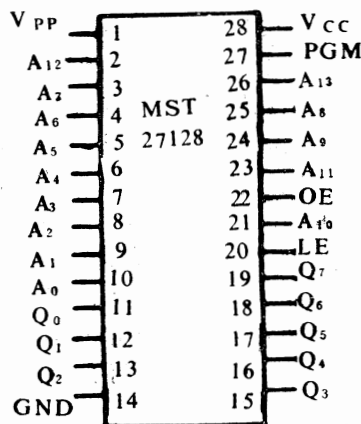


图2. EPROM27128引脚

工作方式选择7种, 根据实用情况选其中三种列于表1中, 从表一中可以看出, 当片选端 \overline{CE} =H时, 输出为高阻; 当其为L时, 是编程或是数据输出, 则取决于 \overline{OE} 、PGM、V_{pp}三个控制端。

3. 具体接法

一是源区EPROM27128 的接法, 二是目区27128的接法, 不管那一片, 其地址线与数据线当然并接至Z80CPU的地址总线与数据总线上, 控制则有区别。

①源区27128的联接

\overline{CE} 与 \overline{OE} 一起接至 $\overline{1Y_1}$, PGM 通过一个2K Ω 电阻接至+5V, V_{pp}、V_{cc}均直接接+5V, 便构成了只读存储器。

②目区27128的联接

仍然利用原TP801—A单板机编程的控制电路,在不破坏原来能对PROM2写入的前提下稍加改动。图3是原PROM2周围的控制电路及增减线路,其中 U_9 、 U_{29} 、 U_{33} 、 U_{34} 、 U_{28} 是原有器件,增加与门74LS11。

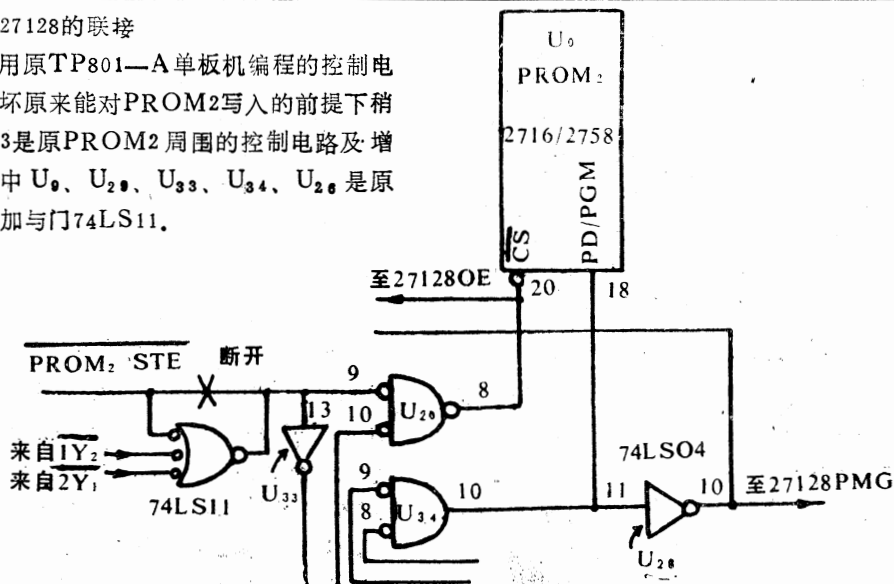


图3 原控制电路与增减线路

将原PROM2 SET线在 U_{33} 与 U_{29} 汇合处断开,改接至74LS11的一个输入端,与门的另两个输入端分别接至 $1\bar{Y}_2$ (目区27128或2764的片选信号)和 $2\bar{Y}_1$ (目区2732的片选信号),对照分析27128与2716工作方式选择表,不难得出如下接法,原 U_{29} (2716)的 \overline{CS} 接至27128的 \overline{OE} ,满足编程时二者均需高电平,读出时二者均需低电平。原2716编程时,PD/PGM为高电平, U_{28} 的输出为低电平,所以将 U_{28} 的输出接至27128的PMG端,满足编时为低电平,读出时为高电平。27128的片选端 \overline{CE} 接至片选信号 $1\bar{Y}_2$,只要27128不被选中,输出为高阻状态,27128的 V_{pp} 与 U_9 的 V_{pp} 相连,受单板机上 S_4 开关控制。

三、软件编程

在所编的程序中,编程程序是主要的,由于没有破坏已有的控制电路,那么编程程序与监控程序中的编程程序大体相同,不同的是:增加了获取预知参数的程序段,即非全片拷贝的情况以及容量不同的EPROM拷贝需要。首先预置源区与目区首地址以及字节长度,还预置定时时间常数,均放入某些内存中或寄存器中,由编程程序首先去获得这些参数,其流程图如图4所示。

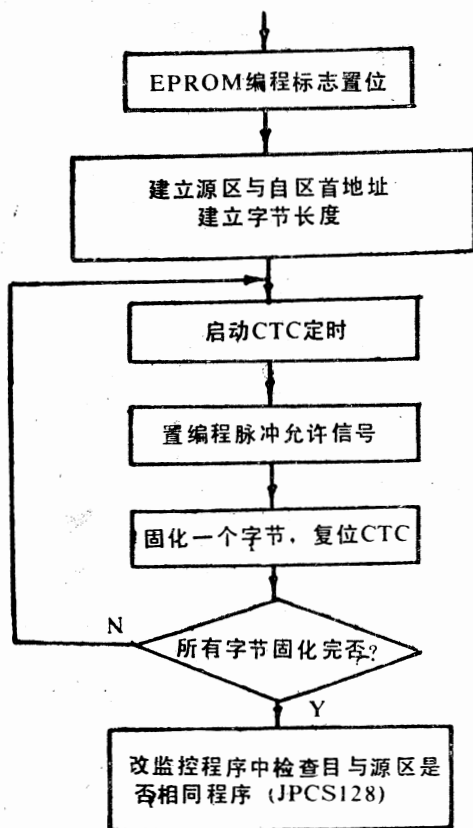


图4 编程程序的流程图

四、结束语

我们根据一大型引进控制设备的需要,作了这项工作,拷贝一片 M5L27128 只需1分钟左右时间,为可靠起见,可设在3分钟左右。当然,不同型号 EPROM 的编程时间不一样,对于16K字节容量的 EPROM 组件,编程时间可能拉得太长,因此要根据产品说明书计算出合适的时间常数。不同 EPROM 组件的 V_{pp} 电压也会有差异,一般调整在14V~20V之间较合适,最好用有调压旋钮的直流电源,逐渐升高 V_{pp} ,以免突然加 V_{pp} 电压烧坏 EPROM。擦洗 EPROM 组件的时间不易太长,7分钟左右的时间就够了。

对不同容量 EPROM 的拷贝可分别编几个独立的程序,且都固化在同一 2716 中,插入原 PROM1 处,使得拷贝不同容量的 EPROM 变成只须运行不同首地址的程序即可。

附程序清单

```

; 拷贝 EPROM 27128 初始化程序
ORG 0800H
START: LD A, 40H
        LD (2F87H), A      ; 被编程的数据源地址的高位字节 → (2F87H)
        LD A, 00H
        LD (2F88H), A      ; 被编程的数据源地址的低字节 → (2F88H)
        LD A, 80H
        LD (2F89H), A      ; 被编程的数据目地址的高字节 → (2F89H)
        LD A, 00H
        LD (2F8AH), A      ; 被编程的数据目地址的低字节 → (2F8AH)
        LD H, 40H          ; 被编程示节数的高八位 → H
        LD L, 00H          ; 被编程示节数的低八位 → L 共计16K个字节
        LD A, 26H
        LD (2F8BH), A      ; 定时时常数 → (2F8BH)
        JP WRITE           ; 转编程子程序
; 编程子程序
OKG 0F90H
WRITE: LD A, 01H
        LD (PRFLG), A      ; 置 EPROM 标志
        LD A, 2FH
        PUSH HL
        POP BC             ; 被编程字节数 → BC
        PUSH HL            ; 被编程字节数存入堆栈区
        LD A, (2F87H)
        LD H, A
        LD A, (2F88H)
        LD L, A            ; 源区首地址 → HL
        LD A, (2F89H)
        LD D, A
        LD A, (2F8AH)
        LD E, A            ; 目区首地址 → DE

```

参考文献

- (1) 一九八五年,《世界集成电路大全》
- (2) “微处理器”,上海交通大学微机研究所、南洋电脑开发总公司,1985年10月
- (3) “TP801 Z80单板计算机使用手册”,北京工业大学电子厂,1987年4月30日

在运行编程子程序之前,必须通过程序或者键盘预置如下参数:被编程的数据区的源地址和目标地址,被编程的字节数以及定时时常数。以 EPROM 27128 为例,如果拷贝 27128 存储器,直接运行起始地址为 0800H 的程序即可;如果将某存储区域数据固化到 27128 的部分区域,则通过键盘预置上述要求的参数后,然后,直接运行起始地址为 WRITE 的编程子程序。至于 2764、2732、2716 三种 EPROM 的固化,与上述类似。

```

ABC:  LD A, 25H
      OUT (86H), A      ; CTC定时方式,无中断
      LD A, (2F8BH)     ; 定时时常数→A
      OUT (86H), A      ; 时间常数
      LD A, 80H
      OUT (8CH), A      ; 清除显示, (PGM PnLSE ENABLE=1)
      LDI               ; 传送一个字节, WAIT时间由CTC定时决定
      LD A, 00H
      OUT (8CH), A      ; 写完一字节, (PGM PULSE ENABLE=0)
      LD A, 03H         ; 复位CTC通道2
      OUT (86H), A
      JP PE, ABC        ; 如果没固化完, 则转
      POP BC            ; 已固化完, 字节数→BC, 准备检查
      LD A, (2F87H)
      LD H, A
      LD A, (2F88H)
      LD L, A           ; 源区首地址→HL
      LD A, (2F89H)
      LD D, A
      LD A, (2F8AH)
      LD E, A          ; 目区首地址→DE
      JP CCS12B        ; 转监控程序中6040H为起始地址的程序, 检查源区与目区数据是否相同。

```

稿 约

- 1、本刊宗旨系“应用促进研究, 研究指导应用”, 故来稿要求“新、实、短、精”, 贵在新颖, 重在实用, 字数宜在3500字以下。
- 2、恕不受理复印、复写稿, 请用钢笔正楷书写于标明字数的方格稿纸上; 若用计算机打印, 每页均要求标明字数。
- 3、文稿中的插图和程序清单均要求一式两份, 编注号码, 并注明在文稿中的位置。图纸要求正规描绘, 程序清单要求黑白反差大, 图纸和程序的宽度为7cm或14cm或21cm, 便于编辑、印刷。
- 4、来稿要求附中英文标题及其摘要。
- 5、请作者自留底稿。人手所限, 恕不退稿。
- 9、要求作者一稿一投。稿件刊出, 赠阅样本, 酌致稿酬。稿件寄出半年后若未见录或未收到录用通知, 作者可函询或自行处理。

本刊编辑部

请订阅《计算机应用研究》双月刊

(邮发代号: 62—68)

本刊是由国家科委系统所属七家省级计算中心及四川省电子学会共同联办并公开发行的科技刊物, 博采、精选国内外计算机应用新技术, 新成果、新信息、新经验。文稿内容突出实用性、工具性、新颖性、通俗性、知识性、趣味性, 系各行各业广大读者学习、推广、应用、普及计算机的园地、窗口和益友, 在全国各地邮局均可订阅, 请勿错失良机。万一脱订, 请同本刊编辑部联系。

1989年和1990年全年每份订价均为: 7.8元

编辑部通讯地址: 成都人民南路4段11号附1号

好消息

广大读者久已期待的实用性技术资料——《静电复印机维修指南》, 现已问世, 每册定价10.00元(不含邮费)。该资料涉猎60余种机型, 融知识性、工具性、资料性、教材性于一体, 图文并茂, 通俗易懂, 实用性强。要想掌握静电复印机的工作原理和操作、维修技能, 并进而成为维修复印机的行家里手, 请订阅《静电复印机维修指南》。该资料由夏则智、吴显俊共同编撰, 由本社编印。需要此资料者, 请同本社联系。

本社编辑部

新刊信息

中国科学技术期刊编辑学会主办的《编辑学报》已经正式出版。

《编辑学报》是有关编辑学的综合性学术期刊, 报道国内外有关编辑学, 主要是科技期刊编辑理论研究成果, 交流编辑实践经验, 为培养编辑人才, 提高期刊质量, 促进科技交流服务。本刊设有理论研究、专题报告、编辑工程、期刊管理、出版知识、科技文章写作、海外信息、书刊评介等。

读者对象, 主要是科技编辑人员, 撰写各类科技文章的科技人员, 大专院校编辑专业的师生等。

《编辑学报》为季刊, 国内定价每本2.00元, 全年4期, 共计8.00元, 本会团体和个人会员9折优惠。订阅者请邮局汇款至“100081”北京海淀区学院南路86号716室中国科学技术期刊编辑学会发行组。

微机流量积算仪研制成功

东南大学研制成功的微机流量积算仪, 目前通过鉴定, 这种热功率计采用单片微机, 软件上采用非线性补偿法实现测量、计算自动化、多功能化, 可精确计算、表示蒸气质量流量和热量流量, 比普通蒸气质量流量计功能多、精度高, 达到八十年代国际水平, 并具有可靠性高、成本低、体积小的优点, 在石油、化工、电力、轻工、冶金等系统中有着广泛的用途。

(李相彬)