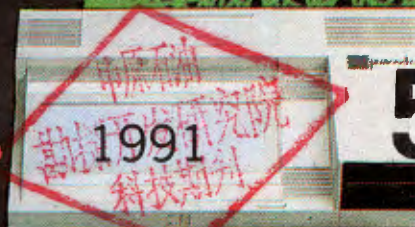
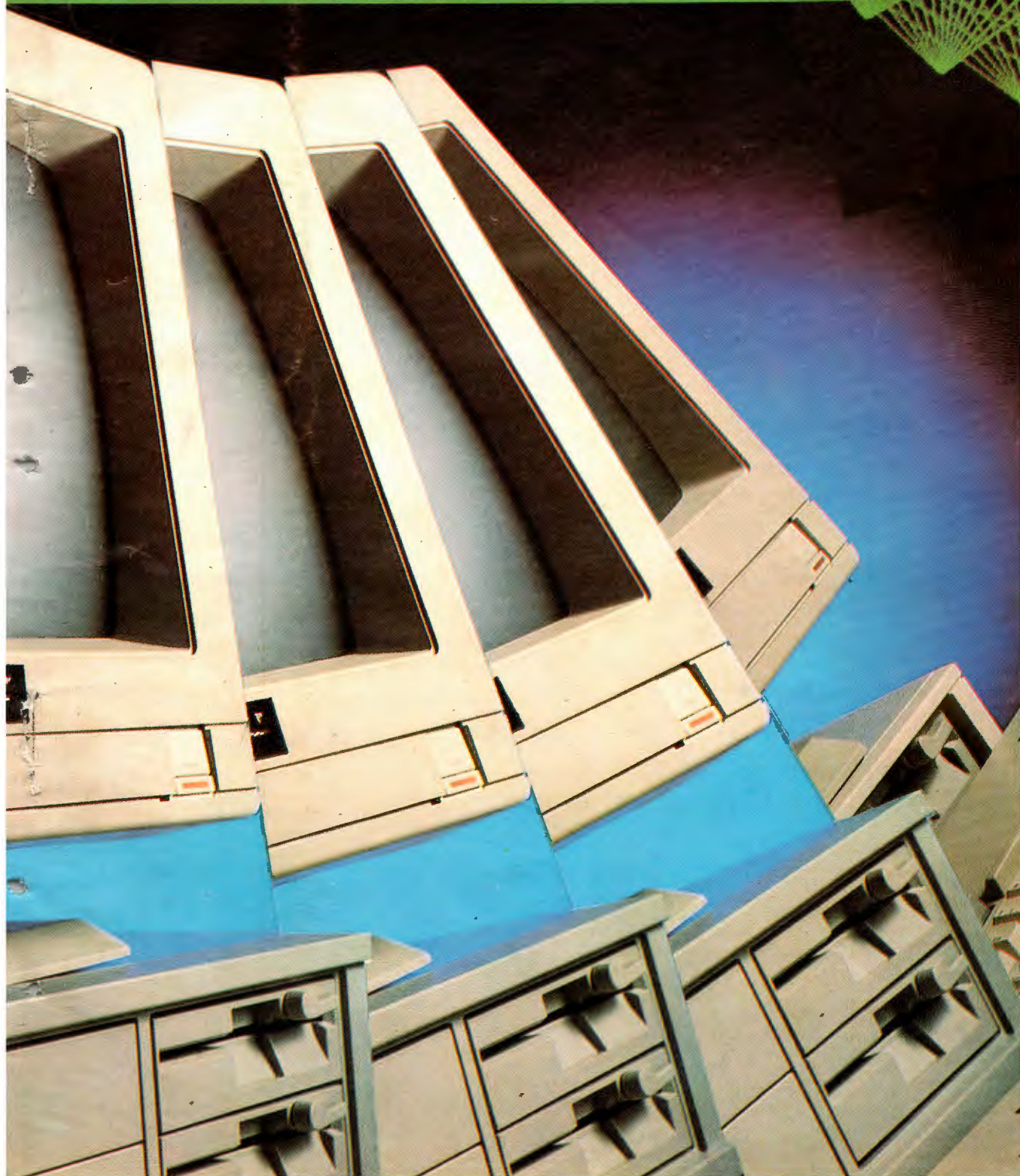


ISSN 1001-3695

# 計算機應用研究



APPLICATION RESEARCH OF COMPUTERS 《計算機應用研究》雜誌





### 《計算機應用研究》雜誌辦刊單位

四川省電子計算機應用研究中心	新疆電子計算中心
貴州省科委計算中心	甘肅省計算中心
安徽省計算中心	廣西計算中心
吉林省計算中心	青海省測試計算中心
河南省計算中心	四川省電子學會

### 《計算機應用研究》雜誌社董事會

董事長：周賁渝

董 事：唐 珍

鄭國基

陸慰梧

李小竹

龔宇清

黎曉常

### 《計算機應用研究》雜誌編輯委員會

主 任 委 員：張執謙

副主任委員：李澤民

委 員：賈洪鈞 曾光初 龔宇清

張國棟 羅海鵬 劉鐵軍

崔振遠 楊劍波 余 凱

1991年第5期（總第43期）

出版日期：1991年9月 本期責任編輯：齊墨之

計算機應用研究（雙月刊）

（公開發行）

國內統一刊號：CN51—1196

主 編：張執謙

副主編：李澤民

編輯出版：《計算機應用研究》雜誌社

通訊地址：成都市人民南路4段11號附1號

郵政編碼：610015

印 刷：新都一中印刷廠

訂 閱 處：全國各地郵局

總 發 行：成都市郵政局

郵發代號：62—68



# 计算机应用研究 第8卷 第5期(总第43期)

## 目 录

### 软件篇

AutoCAD 图形交换文件初探 .....	向南平(1)
冶金过程测温定氧仪系统软件设计与实现 .....	卢正义(6)
用函数定义表示数据结构 .....	叶高英(11)
GWBIOS 图形显示缓冲区的卸出与写入技术 .....	王晓武 毛长风(13)
一种新的 XENIX 界面程序的设计 .....	高 潮(15)
程序设计语言计算机题库系统的研究 .....	陈碧人 龙廷河(17)
开环数控系统消除积累误差的软件设计方法 .....	李天健 翟忠信(19)
也论多种程序语言的综合应用 .....	黄绍东 陈宜金(22)
实用高分辨率彩色图形制作工具 .....	余跃仁(24)
计算机图象仿真技术及应用 .....	杜平安 丁厚本(27)
一个复原空文件的 shell 程序 .....	彭起顺(30)
FOXBASE 中下拉式菜单的简单实现 .....	彭光前(31)
利用 IBM5550 实现表格的自动生成 .....	孙利国 钟立红(33)

### 硬件篇

全自动打印机共享器的研制 .....	唐民山 华晓丽(46)
MCS-51 单片机系统中动态 RAM 的刷新技巧 .....	林 化(46)
MCS-51 单片机与 IBM-PC 微机的串行通讯 .....	肖时江(48)

### 系统篇

微机在发动机生产装配线上的应用 .....	闫立恒 邢洪发(53)
用 8031 控制多屏彩色显示的实现方法 .....	王让定 王效先(56)

### 维修篇

Disk Manager 在磁盘故障诊断中的应用 .....	范立果(59)
--------------------------------	---------

### 信息篇

微机数据通信及其发展前景 .....	张鸿鸣 邢洪发(61)
--------------------	-------------

## 《计算机应用研究》杂志社董事会

董事长:周赛渝

董 事:唐 珍 孙传江 陆慰椿  
吴地兴 郑国基 陈建岭  
黎 蓉 黎瑰常

董事会名单以此为淮

# APPLICATION RESEARCH OF COMPUTERS

Vol. 8 No. 5 (Total 43)

## CONTENTS

### SOFTWARE

- The First Research for Drawing Interchange File of Auto CAD ..... Xiang Nan—ping(1)  
Designing and Realizing for System Software of Metallurgical Temperature and Oxygen Measure .....  
..... Lu Zheng—yi(6)  
Expressing Data Structure by the Function Definition ..... Ye Gao—ying (11)  
The Unloading and Writing Technology of GWBIOS's Graphics Display Buffers .....  
..... Wang Xiao—wu Mao Chang—feng(13)  
The Design for a New shell Program in XENIX ..... Gao Chao(15)  
Research of the Library System on Setting the Examination by Program Designing language on Computer  
..... Chen Bi—ren Long Ting—he(17)  
The Devise on Software Erasing Accumulative Error in the Open—loop Number System .....  
..... Li Tian—jian Zhai Zhong—xin (19)  
Discussing Synthetical Application of Varied Program Languages .....  
..... Huang Shao—dong Chen Yi—jin (22)  
A Usefull Drawing Tool for High—resolution Colour Graphs ..... Yu Yue—ren(24)  
The Emulation Technique of Image on Computers and It's Application .....  
..... Du Ping—an Ding Hou—ben(27)  
A Shell Program of Recoverable Null File ..... Peng Qi—shun(30)  
Simply Realizing for Push—down Menu in FOXBASE ..... Peng Guang—qian(31)  
Auto Tabulating in IBM 5550 Computer ..... Sun Li—quo Zhong Li—hong(33)

### HARDWARE

- Research and Manufacture of the Fully Automatic Printer Sharer .....  
..... Tang Min—shan Hua Xiao—li(44)  
The Refurbished Technique about Dynamic RAM in the System of MCS—51 Single Chip Processor ...  
..... Lin Hua(46)  
Serial Communication of MCS—51 Single Chip Processor and Microcomputer of IBM—PC .....  
..... Xiao Shi—jiang(48)

### SYSTEM

- Application of Microcomputer on Assembly Line for Producing Engines .....  
..... Yan Li—heng Xing Hong—fa(53)  
A Method of Realizing for Controlled Multi—Screens Colour Display by 8031 .....  
..... Wang Rang—ding Wang Xiao—Xian(56)

### MAINTENANCE

- Application in Diagnosing Fault of Disk by Disk Manager ..... Fan Li—guo(59)

### INFORMATION

- Data Communication on Microcomputers and It's Development Prospects .....  
..... Zhang Hong—min Xing Hong—fa(61)

# AutoCAD 图形交换文件初探

中南工业大学资源开发系 向南平

## 一、绪论

近十来,美国 AutoDESK 公司研制的 AutoCAD 一类微机 CAD 系统在国内十分流行,由于它们把以前只有在大中型计算机上才能实现的高级技术奉献给了广大的微机用户,从而促进了 CAD 技术的广泛应用。在短短十多年时间里,CAD 技术即以惊人的速度向前发展,其版本的更新速度更是令人目不暇接。国内也相继出现了各种汉化 AutoCAD 版本,汉化 CAD 的出现,也更加速了 CAD 技术在我国普及和推广应用。目前,在我国各个工程技术部门都有成功的应用 CAD 技术的例子。

AutoCAD 本身是一个完整的图形编辑系统,它维持着一个十分紧凑的图形数据库,其中存储着图形的几何和拓扑信息。但是,用户程序很难直接利用图形数据库中的数据。

但是,经常的情况是,对于一个图形,我们不仅需要利用 AutoCAD 的命令来绘制、修改和输出图形,同时也需要利用图形数据生成各种形式的报表或进行各种性能的分析。另一方面,在某些应用中,我们希望用其他高级语言程序来检查由 AutoCAD 生成的图形或者甚至是直接利用高级语言生成图形再送 AutoCAD 进行修改并输出图形。

为了满足用户的这些要求,AutoCAD 提供了一个 DXF 文件,即图形交换文件(Drawing Interchange File)。它是一个具有严格格式的 ASCII 码文件。以 DXF 文件为媒介,即可实现 AutoCAD 与其他高级语言程序之间的数据交换。要利用 DXF 文件,就必须了解 DXF 文件的结构,只有这样,才有可能在高水平上对 AutoCAD 进行二次开发。

\*\*\*\*\*  
(上接 58 页)64000 个象素点,占用 8031 的 8000H~0BFFFH 的地址空间,象素点按显示位置的不同而存入不同的显示缓冲区空间,偶数行 0、2、4、...、198 的象素点信息存入 8000H(即 0B800H)开始的前 8K 单元,奇数行 1、3、5、...、199 存入 0A000H(即 0BA000H)开始的后 8K 单元,屏上每个象素点[x,y]( $0 \leq x \leq 319, 0 \leq y \leq 199$ )对应与 8031 地址空间中的字节地址如下:

①如果显示行是偶行,则象素点对应的字节地址为:

由于 DXF 文件格式及其组码组值的意义在有关的 AutoCAD 使用手册中都较详尽的说明,因此,本文不想在这方面过多的赘述,只是通过一些典型图素的 DXF 文件实例,说明 DXF 文件的基本构成形式,并介绍几个用高级语言生成 DXF 文件的程序例子,最后给出一个用 True-BASIC 语言构成的 AutoCAD 图形交换文件标准子程序库。

## 二、DXF 文件的总体结构

DXF 文件由顺序出现的四大节(Section)组成,即标题节(Header Section)、表节(Tables Section)、块节(Blocks Section)和实体节(Entities Section)构成。在某一图形的 DXF 文件的标题节中记录着与图形有关的变量的设置;在其表节中记录着四种类型的表,它们依次是线型表、层表、字体表和视图表;在其块节中记录着图形中所有块的定义信息;在其实体节中记录着组成图形的所有实体。

DXF 文件全部是以组(GROUP)组成的,每一个组占两行,第一行为组码,第二行为组值。编写一个构造完整的 DXF 文件的高级语言程序是很困难的,因为必须保持其一致性以便让 AutoCAD 能找到其所需要的一切信息。但实际上,AutoCAD 可以省略 DXF 文件中的很多项,而仍能获得一个可用的图形。如:当不需要设置任何标题变量时,整个标题节即可省略;当不需要设置线型、层、字体和视图时,整个表节也可省略;当没有块定义时,整个块节也可以不要。事实上,整个 DXF 文件只有实体节是必不可少的。因此,用高级语言编写一个生成 AutoCAD 可读图形的 DXF 文件,只需着重了解其实体节的构造情况就可以了。

$$\text{ADDRESS} = 8000\text{H} + 80 * Y/2 + \text{INT}(X/4)$$

②如果显示行是奇行,则对应字节地址为:

$$\text{ADDRESS} = 0A000\text{H} + 80 * \text{INT}(Y/2) + \text{INT}(X/4)$$

字节地址中的相应象素点地址 GRAPH 为:

$$\text{GRAPH} = \text{MOD}(X/4)$$

四、结束语:我们利用现有设备,实现了一机驱动下的多屏显示,对于高分辨率彩显的一机多屏显示扩充亦可利用本文所介绍的方法加以实现。

## 三、DXF 文件的实体节构成

DXF 文件的实体节构成如下:

```

0
SECTION
2
ENTITIES
;
绘图图案: 所有实体的描述
;
0
ENDSEC

```

] 实体开始

] 实体节结束

实体节中可包含多个实体,每个实体以组码为 0、其值为实体名的组开始,实体描述没有明显的结束标志,遇到下一个实体的开始,即意味着本实体的结束。每个实体中除了表示实体所驻留的层名组 8 及其组值是必须的外,其它有关实体的标高(Elevation)、厚度(Thickness)、线型(Linetype)和颜色(Color)的组(分别由组码 38、39、6 和 62 表示),除非需要特殊指定,均可用系统缺省值 0.0、0.0、“BYLTYPE”和“BYLAYER”,当采用系统缺省值时,其相应的组可不出现在该实体的描述中。下面以几种典型实体的 DXF 文件说明其组成情况:

## 1. 绘直线 LINE 实体:

```

0
SECTION
2
ENTITIES
0
LINE
8
0
10
3.0
20
4.0
1.1
7.0
21
8.0
0
ENDSEC
0
EOF

```

] 实体开始

] 实体名为“LINE”

] 实体驻留的层名为“0”层

] 直线的起点坐标, X1=3.0, Y1=4.0

] 直线的终点坐标, X2=7.0, Y2=8.0

] 实体节结束

] DXF 文件结束

该 DXF 文件将在 AutoCAD 图形屏幕的 0 层上绘一条起点为(3.0, 4.0),终点在(7.0, 8.0)的直线。

## 2. 绘园——CIRCLE 实体

```

0
SECTION
2
ENTITIES
0
CORCLE

```

] 实体节开始

] 实体名为“CIRCLE”



8	]	实体驻留的层名为“B”层
B		
10	]	圆心的坐标, X=3.0, Y=4.0
3.0		
20		
4.0		
40	]	圆的半径为, R=3.0
5.0		
0	]	实体节结束
ENDSEC		
0	]	DXF 文件结束
EOF		

该 DXF 文件将在 AutoCAD 图形屏幕的 B 层上, 以点(3.0, 4.0)为圆心, 以半径 R=3.0 画一个圆。

### 3. 写正文——TEXT 实体

0	]	实体节开始
SECTION		
2	]	实体名为“TEXT”
ENTITIES		
0		
TEXT		
8	]	实体驻留的层名为“D”层
D		
10	]	正文左下角坐标为 X=20, Y=3.0
2.0		
20		
3.0		
40	]	正文高度为 0.5
0.5		
1	]	正文内容为“汉字 AutoCAD”
汉字 AutoCAD		
7	]	所采用的字体为“HZ”
HZ		
0	]	实体节结束
ENDSEC		
0	]	DXF 文件结束
EOF		

该 DXF 文件将在 AutoCAD 图形屏幕的 D 层上, 从坐标(2.0, 3.0)开始, 写一个正文字符串“汉字 AutoCAD”, 字符高度为 0.5。

以上给出了三种典型实体的 DXF 文件形式, 读者可以用 EDLIN 文本编辑系统按此格式直接生成几个简单的 DXF 文件, 只需给文件一个扩展名“.DXF”, 然后在 AutoCAD 系统下用“DXFIN”命令调用相应文件的名字, 即可在 CAD 屏幕上显示相应的图形。

AutoCAD 可支持的实体, 根据其版本的高低还有一些, 如: 轨迹(TRACE)、实心区(SOLID)、圆弧(ARC)、三维线(3DLINE)和三维平面(3DFACE)等, 在此不再赘述。

### 四、用高级语言编写生成 DXF 文件的程序

掌握了前述关于图素实体节的构造后,就可以根据其要求来编写相应的生成 DXF 文件的高级语言程序。

由于 DXF 文件是一个 ASCII 码文件,所以只要是能形成 ASCII 码文件的高级语言都是可用的。如,BASIC、FORTRAN、True-BASIC、PASCAL 语言等均可。以下给出两个生成 DXF 图形交换文件实体的 True-Basic 语言子程序例:

#### 1、写 DXF 文件开始及结束标志子程序:

```
SUB BEGIN END( #99,N)
    IF N=1 THEN          ! 当 N=1 时,向 DXF 文件写实体节开始标志
        PRINT #99: 0
        PRINT #99: "SECTION"
        PRINT #99: 2
        PRINT #99: "SNTITIES"
    ELSE                  ! 否则,写实体节结束标志及 DXF 文件结束标志
        PRINT #99: 0
        PRINT #99: "EnDsEc"      ! 实体节结束标志
        PRINT #99: 0
        PRINT #99: "ELF"        ! DXF 文件结束标志
    END IF
END SUB
```

该子程序的作用是向已建立的 DXF 磁盘文件写开始或结束标志;

#### 2、写正文子程序:

```
SUB TEXT( #99,LAYER $ ,TEXT $ ,HIGHT,ROTATE,SCALE,X,Y)
    PRINT #99: 0
    PRINT #99: "TEXT"          ! 实体名
    PRINT #99: 8
    PRINT #99: LAYER $         ! 实体所在的层名
    PRINT #99: 10
    PRINT #99: X                ! 正文左下角 X 坐标
    PRINT #99: 20
    PRINT #99: Y                ! 正文左下角 Y 坐标
    PRINT #99: 40
    PRINT #99: HIGHT            ! 正文高度
    PRINT #99: 1
    PRINT #99: TEXT $           ! $ 正文内容
    PRINT #99: 7
    PRINT #99: STYLE $          ! 正文字体
    PRINT #99: 50
    PRINT #99: ROTATE            ! 正文倾角(整行文字)
    PRINT #99: 41
    PRINT #99: SCALE            ! 正文字体宽度因子
END SUB
```

该子程序在 LAYER \$ 层上,以点 P(X,Y)为正文字符串的左下角,文字高度为 HIGHT、倾角为 ROTATE、字体宽度因子为 SCALE,写正文字符串 TEXT \$。

建立如上所述的一系列子程序后,在主程序中只需调用这些子程序并赋予相应的调用参数,就可生成一个完整的 DXF 磁盘文件。



### 五、图形交换文件标准子程序库

根据 AutoCAD 对 DXF 文件的格式要求,作者用 True-BASIC 语言开发了一个图形交换文件的标准子程序库——CADDXF. LIB,库中每个子程序与 DXF 的一个绘图实体相对应,各个子程序的简要说明如下:

#### 1、建立磁盘文件及文件通道号子程序:OPEN-DXF(#99,DXF-FILENAME\$)

其中通道参数 \$99 为磁盘文件通道号,DXF-FILENAME 为欲建立的 DXF 文件名:

#### 2、写 DXF 文件开始及结束标志子程序:BEGIN-END(#99,N)

当 N=1 时,向磁盘文件写 DXF 文件实体节开始标志;

当 N=2 时,向磁盘文件写 DXF 文件实体节结束标志及 DXF 文件结束标志;

#### 3、绘直线子程序:LTNE(#99,LAYER\$,X1,X2,Y2)

该子程序生成一段在 LAYER\$ 所定义的图层上绘制一条从点 P1(X1,Y1)到点 P2(X2,Y2)直线的 DXF 文件的实体;

#### 4、绘点子程序:POINT(#99,LAYER\$,X,Y)

该子程序在 LAYER\$ 图层上绘一个点 P(X,Y)

#### 5、绘矩形框子程序:BOX(#99,LAYER\$,X1,Y1,X2,Y2)

该子程序在 LAYER\$ 图层上绘一个以点 P1(X1,Y1)为左下角、点 P2(X2,Y2)为右上角的矩形框;

#### 6、绘圆子程序:CIRCLE(#99,LAYER\$,S,Y,R)

该子程序在 LAYER\$ 图层上绘一个圆点在点 P(X,Y),半径为 R 的圆;

#### 7、绘圆弧子程序:ARC(#99,LAYER\$,X,Y,R,A1A2)

该子程序在 LAYER\$ 图层上绘一个圆心在 P(X,Y),半径为 R,起始圆心角为 A1,终止圆心角为 A2 的圆弧;

#### 8、绘折线子程序:POLYLINE(#99,LAYER\$,X,Y,K)

该子程序在 LAYER\$ 图层上绘一第连续的折线,相当于 AutoCAD 内部的多义线命令,点 P(X,Y)为折线(即多义线)顶点,参数 K 取值为 0、1、2。

当 K=0 时,表示该点为折线的起点,写多义线实体开始标志及多义线子实体;

当 K=1 时,表示该点为折线的中间点,只写多义线子实体;

当 K=2 时,表示该为折线的终点,写多义线子实体及多义线实体结束标志。

该子程序的均在循环语句中给折线顶点(X,Y0)循环赋值进行调用。

#### 9、绘实心区域子程序:SOLID(#99,LAYER\$,X1,Y1,X2,Y2,X3,Y3)

该子程序在 LAYER\$ 层上绘一个以点 P1(X1,Y1)、P2(X2,Y2)、P3(X3,Y3)为顶点的三角形实心区域;

#### 10、绘圆环子程序:DONUT(#99,LAYER\$,X,Y,R1,R2)

该子程序在 LAYER\$ 层上,绘一个圆心在点 P(X,Y)、内圆半径为 R1、外圆半径为 R2 的圆环;

#### 11、绘有宽直线:PLINE(#99,LAYER\$,X1,Y1,X2,Y2,WIDTH)

该子程序在 LAYER\$ 层上,绘一条起点在 P1(X1,Y1)、终点在 P2(X2,Y2)、线宽为 WIDTH 的有宽直线;

#### 12、写正文子程序:TEXT(#99,LAYER\$,TEXT\$,HIGHT,ROTATE,SCALE,X,Y)

该子程序在 TAYER\$ 层上,以点 P(X,Y)为正文字符串的左下角,文字高度为 HIGHT、倾角为 ROTATE、文字宽度因子为 SCALE,写正文字符串 TEXT\$;

#### 13、绘正多边形子程序:POLYGON(#99,LAYER\$,X,Y,R,K)

该子程序在 LAYER\$ 层上,以点 P(X,Y)为多边形外接圆圆心、外接圆半径为 R,多边形边数为 K,绘一个封闭的正多边形;

#### 14、绘椭圆子程序:ELLIPSE(#99#LAYER\$,A,B,RROTATE)

该子程序在 #LAYER\$ 层上,以点 P(X,Y)为椭圆中心,以 A、B 为椭圆长短轴、椭圆绕中心旋转 ROTATE 角度,绘一个椭圆;

有了上述子程序,即可绘制任意平面图形。本图形交换子程序库还包括两个三维图形子程序:

# 冶金过程测温定氧仪系统软件设计与实现

辽宁大学计算机系 卢正义

**摘要:**本文概述了冶金过程测温定氧仪多功能特点,着重介绍该系统软件设计与实现,以此展示其应用前景。

## 一、概述

随着电化学法直测技术的应用与深入发展,各种类型液态金属测温定氧仪应运而生,尽管功能各异、屡见更新,但它们集中问题是:功能专一、适用面宽,这是由于它们都是根据一、二个液态金属直测算式而研制的。然而其算式随诸因素变化而不断增加,显然它们就不能适应这些直测因素的变换,多功能冶金过程测温定氧仪适应这方面的变化。本系统通过触摸键盘可以选择各项功能,可用于各种液态金属直接定氧,预报钢液中酸溶铝、碳含量或液态炉渣氧化性,选配各种热偶测定不同介质温度。经二次开发,它还可以用于直接定

氢、定氮、定硫以及钢液结晶定碳等。因此,它是应用、开发固体电解质直接定氧(氢、氮、硫)技术以及选配各种热偶测定不同介质温度的理想仪表。

## 二、系统结构与功能

本系统采用微处理器 Z84COO, 8KRAM, 16KEPROM。A/D 转换器使用三位半双积分型芯片 MC14433, 测温能分辨 1℃ 以下, 采样速度每秒约 15 次。为加强抗干扰能力, 采取光电耦合、低频有源滤波、防脉冲干扰数学字滤波、滑动窗口法自寻最佳采样电势措施。同时通过远距离串行通讯实现大屏幕显示, 其系统结构如图 1 所示。

### 15、绘三维直线子程序: 3DLIN(#99,LAYER\$,X1,Y1,Z1,X2,Y2,Z2)

该子程序在 LAYER\$ 层上绘一条起点为 P1(X1,Y1,Z1)、终点为 P2(X2,Y2,Z2)的三维直线;

### 16、绘三维平面子程序: 3DFACE(#99,X1,Y1,Z1,X2,Y2,Z2,X3,Y3,Z3)

该子程序在 LAYER\$ 图层上绘一个由点 P1(X1,Y1,Z1)

、P2(X2,Y2,Z2)和 P3(X3,Y3,Z3)

该子程序在 LAYER\$ 层上绘一第起点为 P1(X1,Y1,Z1)

、P2(X2,Y2,Z2)和 P3(X3,Y3,Z3)所定义的三维平面。

上述子程序在均已 IBM 机上调试通过,有了这个子程序库,就可方便的用高级语言直接生成诸如地形图、地籍图、地质图及各种工程图的 DXF 文件。当然,根据实际情况,还可修改、增加或删除部分子程序,以满足具体工程技术要求。

## 六、结束语

AUTOCAD 之所以发展如此之快,并广受用户喜爱,其充分开放的结构,是其成功的主要原因。它不仅为用户提供了丰富的绘图和编辑功能,而且提供了形式多样的与高级语言接口的方式。DXF 图形交换文件还只是其一。除此这外,还有一个结构更紧凑的二进制图形交换文件 DXB 和一个 IGES 格式的图形交换文件。这几种形式的图形交换文件各有其优缺点,DXF 文件具有直观、编辑修改容易、功能比较齐全的特点,但格式较复杂,生成的 DXF 文件占用外存较多;而 DXB 图形交换文件则占用外存少,但其文件是以二进制形式写成的,所以不易对其进行直观的分析 and 修改;IGES 文件目前用的不多,所掌握的资料也不够。三种图形交换文件各有千秋,究竟采用何种形式,应视用户的实际情况及个人爱好。

有关 DXB 图形交换文件的格式及如何由高级语言生成 DXB 文件的一些问题,将另文介绍。

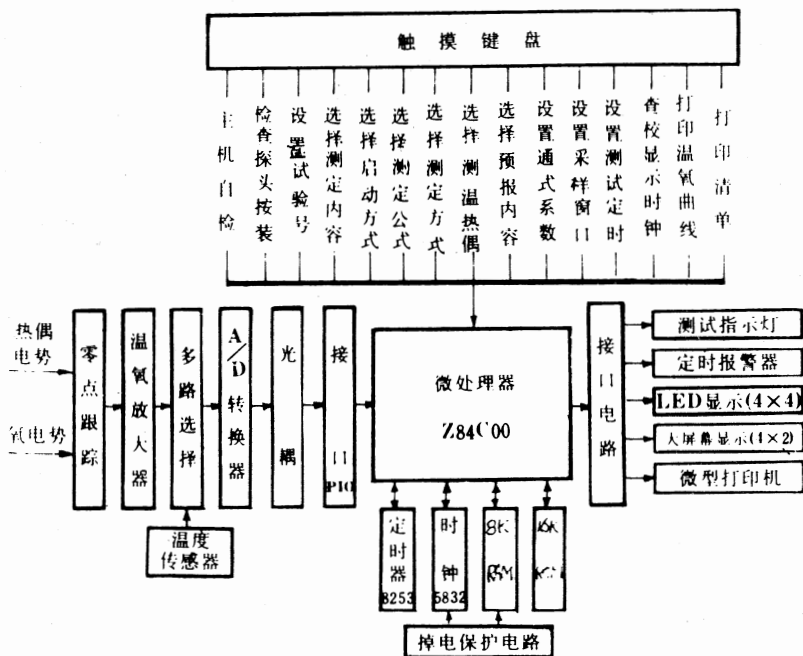


图1

### 系统功能:

- 适应范围宽:可配用国内外各种液态金属定氧探头。任选测温热偶,可瞬间或连续测温定氧。
- 系统自检:置:“自检”键,自动检测系统,显示自检结果。
- 配置数据硬开关:打开它,按功能编码写入各种数据;锁上它,只读不能写入,以保护已存贮的数据。
- 拔偶、定时报警:采样结束报警提示拔偶;8253定时器定时报警。
- 掉电保护:采取可靠性掉电保护措施,以保护RAM与时钟芯片内数据。
- 时钟自动计时:采用5832时钟芯片,显示、打印时间。
- LED显示器:同时显示氧含量、氧电势、温度、[Al]/[C]预报值;大屏幕显示氧含量、温度。
- 打印输出:打印号、采样时间、温度、温度电势、氧含量、氧电势[Al]/[C]预报值以及温度、氧电势曲线。

### 三、系统软件

系统软件包括主程序、子程序库和11个功能模块即:键盘监控、自检、温度补偿、检查探头安装、零点跟

踪、自寻最佳采样电势、时钟、列清单、测温、定氧和[Al]/[C]预报值等程序约16K字节。由于采用模块化程序设计法,程序结构紧凑、清晰、控制灵活可靠。

#### 1、主程序

主程序实际上是系统管理程序,根据用户设置的各种状态调用某功能模块以实现其功能,其流程图如图2。主程序完成系统初始化后判断数据硬开关,当软开关为非工作状态时,无论它处于什么状态,主程序都进入键盘监控程序。

LED显示器分上、下两排数码管各八位。上排左两位显示功能编码,下排右六位显示输入数据。当显示提示符“—”时,提示输入功能编码和相应的各种控制字、工作方式或通式系数。从图2看出:由于程序连续扫描键盘,随时可以显示输入数据。每次键盘输入完毕,按“回车”键,自动转向处理程序,根据功能编码作相应处理然后,再次扫描键盘。例:编码为10时,显示时间;为20时,列清单即把输入数据按功能编码顺序依次全部打印出来以供参考。显示“—”时,直接按回车键,此时处理程序置软开关为工作状态,这样主程序脱离键盘监控程序,显示“HELLO———”以后自动消失,从而开始下面的采集数据工作。



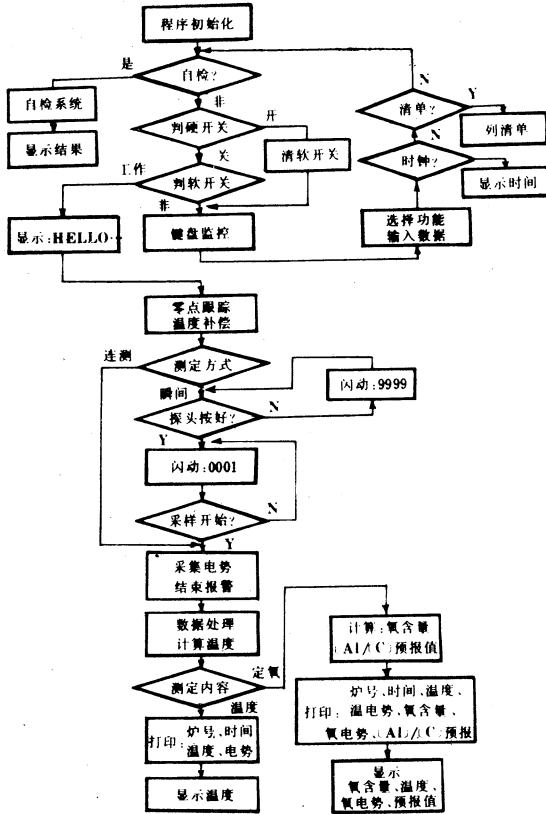


图 2

## 2、键盘与显示器

本系统由于键盘与显示器相互配合,实现系统多功能的特点,其显示格式见图 3。某项采样失败,则该项显示“——”,其它各项显示采样成功的结果。键盘输入时,显示格式见图 4。显示“——”时,用户只需依次键入功能编码、输入数据,每次都在相应位置显示其内容。

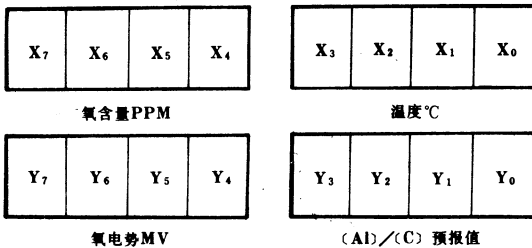


图 3

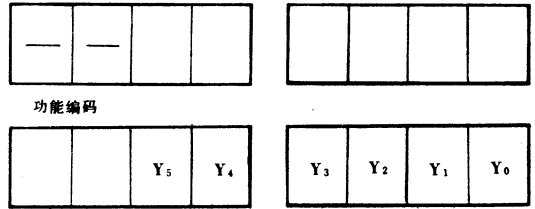


图 4 输入数据

软件采用扫描显示法,可以显示 0—9 和不同符号。显示前应把上排  $x_i$  数据送到首址 2800H 的  $y_i$  缓冲区,然后把下排  $y_i$  数据送到首址 2810H 的  $y_i$  缓冲区再调用扫描显示程序,显示其内容。键盘有数字键 0—9、A、B、C、0、\*, 回车键(\*)处理程序采取查询方式对键盘输入的控制字、功能选择、通式系数分别进行处理保存在 RAM 中待用。数据处理后自动显示“——”,允许再次输入数据。

## 3、采集子程序

系统 A/D 转换器采用 MC14433 芯片,采取中断方式 2 采集数据。其采集数据经光耦送到 Z84C20 PIO 的 A 口输入,其 B 口作为输出控制选择采集温度热电势、氧电势、温度补偿、零点跟踪等各电路以及测量、成功、失败指示灯的显示。采集子程序实现连续快速采集数据,而中断服务程序完成一次数据采集过程,入口是 0790H。从采集子程序看到,采集一路所有数据均可以存放在以 ADDR H 为首址的数据缓冲区内,次数 B 为 0 时,此路采集工作结束。因此它是通用子程序,使用时只需修改一下数据区首址 ADDR H 和采集次数 B,其流程图略。

### 采集子程序

ORG 0800H

ZJZ; LD HL, ADDR H; 数据区首址

LD B, NRH ; 采集次数

CALL ZDZ ; 采集数据

RET

ORG 1000H

ZDZ, LD E, 20H ;软件延时

LOOP, DEC E

JR NZ, LOOP

LD A, 83H; 允许中断

OUT (82H), A

IM2; 中断方式 2

EI; 开中断、转中断服务程序

HALT

DJNZ ZDZ; 次数非零、继续采集

RET

#### 4、自寻最佳采样电势子程序

MC14433 采集到温度电势和氧电势混有许多干扰信号,有效采样电势应是曲线中较平直一段 MN 即平台电势如图 5。软件可采取防脉冲干扰数字滤波滤掉干扰信号,同时设置测量窗口,采用滑动窗口法自动找到平台电势。而窗口内平台电势的算术平均值为有效采样电势,然后去掉零点跟踪线路得到放大器零点漂移电势,最后得到最佳采样电势值。测量窗口高度 H 与长度 L 决定采样电势的精度。

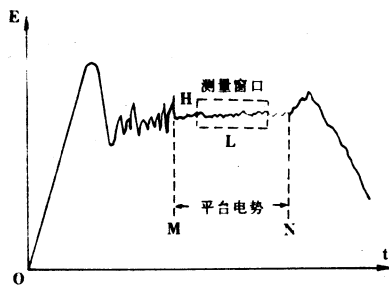


图 5

#### 5、测温子程序

实际应用、研制探头经常要连续测量液态金属,要求测温范围大。本系统采用公式法计算温度值,共值与查表值完全一致。测温范围:

单铂铑偶(Pt-PtRh); 0-1768℃

双铂铑偶(Pt-PtRh); 0-1820℃

热电偶测温时,其热电势与温度值对应关系为多项式即:

$$\text{双铂铑热电偶, } Et = \sum_{i=0}^8 a_i \cdot t^i$$

单铂铑热电偶:

$$-50^\circ\text{C} - 630.74^\circ\text{C}; Et = \sum_{i=0}^8 a_i \cdot t^i$$

$$630.74^\circ\text{C} - 1064.43^\circ\text{C}; Et = \sum_{i=0}^2 b_i \cdot t^i$$

$$1064.43^\circ\text{C} - 1665^\circ\text{C}; Et = \sum_{i=0}^3 c_i \cdot [(t - 1365)/300]^i$$

$$1665^\circ\text{C} - 1767.6^\circ\text{C}; Et = \sum_{i=0}^3 d_i \cdot [(t - 1715)/50]^i$$

其中: Et 为电势(μV), t 为温度(℃);

$a_i, b_i, c_i, d_i$  均为已知常数。

为加快计算速度,采用快速弦载法计算温度值 t。

对快速迭代公式:  $t_2 = t_1 - \frac{E(t_1)}{E(t_1) - E(t_0)} \cdot (t_1 - t_0)$  每次迭代均计算  $E(t_0), E(t_1)$  值, 仅以双铂铑偶为例分析计算  $E(t)$  算法:

$$\begin{aligned} E(t) &= \sum_{i=0}^8 a_i \cdot t^i \\ &= a_8 t^8 + a_7 t^7 + a_6 t^6 + a_5 t^5 + a_4 t^4 + a_3 t^3 + a_2 t^2 + a_1 t^1 + a_0 t^0 + \\ &= (a_8 t + a_7) t + (a_6 t + a_5) t + (a_4 t + a_3) t + (a_2 t + a_1) t \end{aligned}$$

其中:  $a_0 = 0$ , 故  $a_0 t^0 = 0$ 。

因此计算  $E(t)$  可采取多重循环程序, 循环体基本公式是  $(A+Q)t$ , 其流程图略。

#### 6、定氧子程序

液态金属活度是根据固体电解质氧电池原理测定的, 我们采用液态金属直接定氧通式, 复盖各种定氧公式即:

$$a[O] = 10^{(a_0 + b_0/T)}$$

$$\left[ \frac{10^{(a_1 b_1/T)/4} + 10^{(a_2 b_2/T)/4}}{10^{5.041(K+43-b_3)/T}} - 10^{(a_1 b_1/T)/4} \right]^2 \cdot 10^4$$

其中,  $a[0]$ : 待测液态金属中氧活度 (PPM)。

$E$ : 氧电池电势 (MV)。

$T$ : 待测介质温度 (K)。

$t$ : 待测介质温度 (°C)。

$a_0, b_0$ : 氧深解系数。

$a_1, b_1$ : 固体电解质系数。

$a_2, b_2$ : 参比材料系数。

$a_3, b_3$ : 电析引线系数。

通式系数  $a_0b_0 - a_3b_3$  分别概括液态金属种类、固体电解质电子导电性能、参比材料种类及电极引线材料的变化, 而  $a_0b_0, a_2b_2$  两组系数还概括热力学数据出处变化。它们均可以经过计算、查表或自测试而确定。

因此用户可以通过键盘输入不同  $a, b$  通式系数长期保存在 RAM 工作区中, 使用情况变化只需修改  $a, b$  值, 因此拓宽仪表适用范围。

#### 四、结束语

本仪表性能价格比高, 通用性强, 实现多功能一体化, 经过半年多时间在鞍钢测试使用, 取得满意效果。系统增加测量功能, 主机硬件不动, 仅是软件的开发, 显然它有着可观有应用前景。

#### 参考文献

[1] 王丹霞编, 微型计算机汇编语言程序设计, 电子工业出版社, 1985 年。

[2] 何立民编, 单片机应用系统设计, 北京航空航天大学出版社, 1990 年。

## 河南省平原制药厂建成微机局网

### 管理信息系统(MIS)

#### ——河南省计算中心又一重大技术成果通过省级鉴定

河南省平原制药厂是生产抗生素的大型企业, 是全国医药行业的重点企业, 有职工三千多人, 其医药产品行销全国, 部分产品已进入国际市场。为提高管理水平, 增强企业活力, 河南省科委对该厂的计算机应用给予重点扶持。安排河南省计算中心与平原制药厂联合开发了该厂的微机管理系统, 并于七月二十七日通过了省级科委主持的鉴定。

研制组一贯把探索一条适合我国国情的企业管理信息系统和分析、设计、开发的有效方法作为课题目标之一。在研制过程中, 认真按照软件工程的理论和方法, 总结出“五图四表一报告”这么一套简明实用的软件工具。也研制出一个高效实用的信息系统。该系统采用模块化的方法设计, 便于维护和扩充。系统支撑环境是 3+ 网络, 有九个工作站, 并能通过调制解调器与省信息中心完成远程通讯。网内实现设备共享和软件共享。各工作站之间

实现数据库共用和电子信息传递, 从而实现了全厂生产经营信息的共享, 保证这些数据的准确性、一致性和及时性。为系统的后续开发进而实现生产过程自动控制和管理一体化, 打下了可靠的基础。

目前系统已完成财务、仓库、销售、劳资、供应等九大子系统。涉及面广、实时性强, 是一个重大的软件工程。系统使用后, 为该厂中、上层管理决策提供了及时可靠的依据, 使管理和计划建立在科学基础上, 对提高工作效率和企业管理水平起了积极的促进作用, 有明显的经济效益和社会效益。除节约人力、节约时间外, 据该厂有关人员估计, 仅从降低定额储备资金、加速资金周转、控制费用支出以及减少库存资金占用等几个方面估计, 每年可产生直接效益 225 万元, 很具应用推广价值。

(河南省计算中心 崔振远)



## 用函数定义表示数据结构

核工业西南物理研究院 叶高英

在函数式程序设计中, McCarthy 精心设计和开创的使用  $s$ -表达或记法将一个函数定义或一个完整的程序表示为语言中(LISP)的合式数据结构即表数据结构, 这一属性已为大家所熟知, 亦为几乎所有非数值运算语言所沿袭使用。但反过来, 能否用函数定义来表示数据结构呢? 在绝大数程序设计语言中是不可能的, 在函数式程序设计中则有可能, 而且用此方法书写的程序较之用数据结构表示法设计的同一程序其运行效率要高。本文以集合及其上定义的特定运算为例说明这种程序设计方法的可行性。

我们选择的例题要求在任意一个给定的表中, 检索并记录仅出现过一次的原子集。

假定读者熟悉表、原子这些概念, 我们将用类似 LISP 语言先写出一个较容易理解的程序, 然后再引入用函数表示数据结构表的算法程序。

如果把计算结果看作是集合, 则其中元素出现的顺序便无关紧要。书写这一程序可作如下分析:

假定输入表为  $S$ , 输出结果叫做 Singleton, 则按照惯常表运算方式, 我们可做如下简单分析: 若  $S$  是原子则结果 Singleton= $S$ , 否则假定 Singleton( $hd\ s$ )= $A$ , Singleton( $tl\ s$ )= $B$ , 这里  $hd, tl$  为得到表首部与尾部的选择运算。让我们看一看最后结果 Singleton( $s$ )应该是什么? 它等于  $A$  与  $B$  的 集吗? 显然不是, 因为  $A$  与  $B$  可能有公共元素, 该公共元素在  $S$  中出现两次, 那么它是否等于除公共元素之外的其它元素的 集呢? 也不是。因为一个原子在  $hds$  中仅出现一次则它属于  $A$ , 但该原子在  $tl\ s$  中出现两次以上故它不属于  $B$ , 可见该原子仍不能出现在最后结果中。由此可见无法仅从  $A$  与  $B$  中直接得到结果, 需要找出那些在  $A$  中出现且不在  $tl\ s$  中出现以及在  $B$  中出现又不在  $hd\ s$  中出现的原子一起构成要求的原子集。据此可以写出如下程序:

```
Singleton(s)≡tl(sing(s,(),()));
Sing(s,a,b)≡{ if atom(s) then
if member (s,a) then a && b
else if member(s,b) then(s && a)&& remove(s,b)
else a &&(s && b)
endio
else sing (tls,hd p,tlp)
endio
where p≡osing(hds,a,b)
};
remove(x,y)≡if x=hd y then tl y
else hdy && remove (x,tly)
endif;
member(x,y)≡ if y=() then F
elseif x=hd y then T
else member (x,tly)
endif;
```

其中:  $()$  表示空表。

$\&\&$  表示表的 接或在表上增加新元素的运算。

现在让我们看一下  $sing(s,a,b)$  的返回值。若用偶对  $(c,d)$  记返回的两个集合  $c$  和  $d$ , 则  $c$  包含:

- 1、所有  $a$  的元素;
- 2、所有  $s$  中多次出现的原子;

3、所有  $s$  中一次出现且包含在  $b$  中的原子。  $d$  包含：

- 1、所有  $s$  中一次出现且既不在  $a$  中又不在  $b$  中的原子；
- 2、所有不出现在  $s$  中的  $b$  的元素。

由此可见，调用  $\text{Sing}(s, \langle \rangle, \langle \rangle)$  返回的偶对中， $c$  包含  $s$  中多次出现的原子， $d$  包含  $s$  中仅出现一次的原子。因此上述 Singleton 函数的定义是正确的。

从函数定义  $\text{sing}(s, a, b)$  中集合  $a$  与  $b$  的使用情况来看，它们都是用 member 函数进行检索，用置运算进行扩展的，只是对  $b$  集合有时还需用 remove 函数。而程序最终结果只有  $b$  有关， $a$  仅为内部运算的工作集。因此下面我将只讨论用定义函数来表示集合  $a$ ，这不会影响程序的使用。

先前我们在用表  $s$  来表示集合的前提下，可用函数

$f(x) = \text{member}(x, s)$

来表示集合，并且若想在  $s$  中加入一个新成员  $x$ ，则用  $x \&\& s$  运算来完成。现在我想用一个函数定义完全表示集合，则必须构造一个新的函数值使之对扩充后的集合亦具有 member 函数的功能和性质。因此函数的定义表达式应是：

$\lambda(y)$  计  $y = x$  then  $T$  else  $f(y)$  endif;

这个  $\lambda$ -表达式当作用自变量  $y$  时结果如何呢？显然如果  $y$  等于新加进集合中的元素  $x$  或者作用于加入前的集合  $f(y)$  为真时均返回值  $T$ ，因此它确实表示加入  $x$  后所得到的集合。由定义可知空集可以表示为：

$\lambda(y)F$

现在让我们用集  $a$  的这种函数表示来修改先前的  $\text{sing}(s, a, b)$  定义如下：

$\text{singleton}(s) \equiv \text{tl}(\text{sing}(s, \lambda(y)F, \langle \rangle))$

$\text{sing}(s, a, b) \equiv \{ \text{ if } \text{atom}(s) \text{ then}$

$\text{ if } a(s) \text{ then } a \&\& b$

$\text{ else if } \text{member}(s, b) \text{ then}$

$(\lambda(y) \text{ if } y = s \text{ then } T \text{ else } a(y) \text{ endif } \&\& \text{ remove}(s, b)$

$\text{ else } a \&\& (s \&\& b) \text{ endif}$

$\text{ else } \text{sing}(\text{tls}, \text{hdp}, \text{tlp})$

$\text{ endif}$

$\text{ where } p \equiv \text{sing}(\text{hds}, a, b)$

$\} ;$

只有三个地作了改动：参数部分的空表改用  $\lambda(y)F$ 。调用  $\text{member}(s, a)$  变为调用  $a(s)$ ；置运算  $s \&\& a$  用  $\lambda(y) \text{ if } y = s \text{ then } T \text{ else } a(y) \text{ endif}$  代替。在  $\text{sing}$  的返回值偶对中，第一个成员变成了函数。

为了清楚起见，让我们再看一下函数表示集合的实际例证。假定初始时空集，然后逐个加入元素  $A, B, C$ 。则用来表示集合的函数值序列是：

$a_1 = \lambda(y)F$

$a_2 = \lambda(y) \text{ if } y = A \text{ then } T \text{ else } a_1(y)$

$a_3 = \lambda(y) \text{ if } y = B \text{ then } T \text{ else } a_2(y)$

$a_4 = \lambda(y) \text{ if } y = C \text{ then } T \text{ else } a_3(y)$

由此可以看出，若调用  $a_4(A)$  则生成调用  $a_3(A)$ ，后者又生成调用  $a_2(A)$  并返回结果  $T$ 。类似地如果调用  $a_4(D)$  则依次生成  $a_3(D)$ ， $a_2(D)$ ， $a_1(D)$  的调用序列并最后返回结果为  $F$ 。可见  $a_4$  的确是我们所期望的表示包含元素  $A, B, C$  的集合表示函数。

应该指出这里介绍的方法并非对所有集合给出一个一般性的定义。但是函数的这种特殊用法较之使表结构和 member 函数执行同一操作其运算速度要快，效率高。而且这种在数据结构中嵌入函数，从而替代数据结构的用法是函数式程序设计非常强的功能属性之一。

## GWB IOS 图形显示缓冲区的卸出与写入技术

武汉总后基地自动化指挥站王晓武 毛长风

国产长城系列计算机(包括:0520CH, DH, 286EX, 286B 及 386 等)具备很强的图形处理功能,这些图形功能来源于系统的高性能硬件配置和完备的汉字操作系统的支持。长城系列机(包括东海,浪潮,京粤等国产机)配置了高分辨率彩色显示适配器。在 GWBIOS3.0 以上汉字操作系统支持下,能够方便灵活地开发高质量的图形软件。本文讨论的问题并不是如何开发图形软件,而是如何将已有的高质量的显示图形作为文件存贮到磁盘上,在需要的时候再将这个图形文件直接显示到屏幕上,可大大提高图形的输出速度,即所谓的图形显示缓冲区的卸出与写入技术。

GWBIOS 的显示器中断服务程序(INT10H)中含有两个相对应的功能,即保存图象(AH=34H)和输出图象(AH=36H)功能。它们的调用方法分别为:

功能号:AH=34H——保存图象

输入参数:X1,Y1——矩形区域左上角绝对座标

X2,Y2——矩形区域右下角绝对座标

OFS——保存图象缓冲区的地址偏移量

SEG——保存图象缓冲区段地址

参数传递地址:[BP]:[BX]

参数单位:每个参数占两个字节(高位在前,低位在后)

功能号:AH=36H——输出图象

输入参数:X,Y——矩形图象写入屏幕左上角的 X,Y 座标

OFS——缓冲区地址偏移量

SEG——缓冲区段地址

A——操作方式字(1、2、3、4、5、分别为或、与、直接传递、取反、异或等操作)

参数传递地址和参数单位同功能 34H。

GWBIOS 的上述系统功能,为图形显示缓冲区的卸出与写入提供了基本接口,但是,上述 INT 10H 功能只是与内存单元作交换操作,要把图形信息转换成磁盘文件,必须在应用上述功能的基础上做进一步的处理。

为了方便用户使用,笔者编写了一个图形显示缓

冲区卸出与写入通用程度 GRAP,该程序通过命令行赋予不同的选择参数(GRAP 0 或 1),分别完成将屏幕上的图形信息作为文件写入磁盘或从磁盘上将图形文件显示到屏幕上的功能。该程序的处理方法是在调用 GWBIOS INT 10H 的 34H,36H 功能的基础上,将内存缓冲区的图形信息转换成磁盘文件,或将磁盘文件转换成内存图形信息。该程序可作为一个 DOS 的外部命令使用,也可在各种高级语言应用程序中调用。该程序将显示器上的一屏图形划分成两部分,分别写入 G.001 和 G.002 这两个磁盘文件上。

该通用程序的使用方法是:先运行你自己的图形软件,生成你所需要的图形屏幕,紧接着运行 GRAP 0 (回车),这样就将该显示图形写到 G.001 和 G.002 这两个文件中。以后,在你需要时,随时可以运行 GRAP 1 (回车),将图形快速地显示出来。若用户在 Dbase 或 BASIC 程序中调用该程序,可用"RUN GRAP 0 或 1"或"SHELL GRAP 0 或 1"语句,方便灵活地使用。

GRAP. ASM 源程序如下:

```
stack segment para stack
db 64 dup(0)
stack ends
data segment para public
dss db 54004 dup(0)
file1 db "G.001",0
file2 db "G.002",0
mode db 5 dup(0)
errmsg db "input error!",0ah,0dh,24h
;输入参数赋值
para dw 0,0,639,224,offset dss,seg dss ;(X1,
Y1,X2,Y2,OFS,SEG)
para2 dw 0,225,639,449,offset dss,seg dss;
para3 dw 0,0,offset dss,seg dss,3 ;
(x,y,OFS, SEG, A)
para4 dw 0,225,offset dss,seg dss,3 ;
data ends
code segment para public
assume cs:code,ds:data,ss:stack
```



```

start    proc far
    push ds
    mov ax, 0
    push ax
    mov ax, data
    mov es, ax
    assume es: data
    mov si, 80h      ; 从 PSP 中获取命令行参数
    mov di, offset mode
    mov cx, 4
    cld
    rep movsb
    mov ds, ax
    cmp byte ptr [mode+2], '0'
    jz txsc          ; 转保存图象处理
    cmp byte ptr [mode+2], '1'
    jnz err
    jmp txxs         ; 转输出图象处理
err:     mov ah, 09h; 命令行尾参数出错处理
    mov dx, offset errm
    int 21h
    ret
txsc:    ; 图形信息写入磁盘文件处理
    mov bx, offset paral
    call readb
    mov dx, offset file1; Write File1——G. 001
    call writf
    mov bx, offset para2
    call readb
    mov dx, offset file2; Write File2——G. 002
    call writf
    ret
txxs:    ; 磁盘图形文件送屏幕显示处理
    mov dx, offset file1; Read File1——G. 001
    call readf
    mov bx, offset para3
    call writb
    mov dx, offset file2; Read File2——G. 002
    call readf
    mov bx, offset para4
    call writb
    ret
readb    proc near      ; Read Graphic Display Buffer
    mov bp, data
    mov ah, 34h
    int 10h
    ret
    readb endp
writb    proc near      ; Write Graphic Display Buffer
    mov bp, data
    mov ah, 36h
    int 10h
    ret
    writb endp
writf    proc near      ; Write Disk File
    mov cx, 00
    mov ah, 3ch
    int 21h
    mov bx, ax
    mov cx, 0d2f4h
    mov dx, offset dss
    mov ah, 40h
    int 21h
    mov ah, 3eh
    int 21h
    ret
    writf endp
readf    proc near      ; Read Disk File
    mov cx, 00
    mov ax, 3d00h
    int 21h
    mov bx, ax
    mov cx, 0d2f4h
    mov dx, offset dss
    mov ah, 3fh
    int 21h
    mov ah, 3eh
    int 21h
    ret
    readf endp
start    endp
code     ends
end start

```

# 一种新的 XENIX 界面程序的设计

浙江省衢州化学工业公司计算机中心 高 潮

**摘要:**本文详细介绍了一种新的 XENIX 界面程序的设计,这种新的界面与原标准 SHELL 完全兼容,它增加了以往命令的记忆、恢复和即时显示用户当前工作目录等功能,提供了比原标准 SHELL 更加良好的用户界面,更便于 XENIX 的使用开发。

## 一、前言

目前使用 XENIX 的用户越来越多,其所用的界面为标准 shell。标准 shell 没有以往输入命令的记忆和恢复功能,如果用户输完一个命令行后发现其中有字符有错,只能从头重新输;在开发程序过程中,一些命令可能会循环地使用,它们只能一次次地输入;用户工作目录不能即时显示。这些使用使用起来不甚方便,为此,笔者开发了 XENIX 下一种新的界面程序,取名为 pcsh,它支持原标准 shell 的所有功能,并弥补了标准 shell 的上述不足之处,它能即时在提示符中显示用户当前目录,能记忆、恢复和编辑用户以往输入的 24 条命令。经使用,效果很好。

## 二、程序设计

XENIX 操作系统结构分为系统内核和核外程序,也称操作系统和实用程序两部分。其内核完成操作系统的主要功能,如存储管理、进程管理、设备管理及文件管理等等,内核向核外程序提供了有力的支持;核外各种实用程序通过 shell 在内核的支持下实现各种功能。shell 是用户与操作系统的接口,与一般的操作系统不同,XENIX 的 shell 不属于操作系统内核,而是属于核外部分,是以/bin/sh 文件名方式存在的核外实用程序,是在核外以用户态方式运行的一般程序。

shell 与一般的实用程序不同,它是在用户登录后首先被系统调用的,是用户所有进程组首,它接收用户输入的命令后在模块中找到相应程序并运行之,成为 XENIX 与用户的界面,shell 还提供了其它一些重要特性,如输入输出的重新定向、管道命令及后台操作等性能。

XENIX 的进程管理功能允许多进程的并行处理,父进程可以建立子进程执行核外的各种实用程序,/

bin/sh 也可在子进程中被调用。

根据 XENIX 的这些特点,新界面程序设计成命令输入、命令执行两部分组成,其逻辑结构如下:

```
while (EOF not typed) {  
    get command from user  
    execute it by calling /bin/sh  
}
```

(1)命令输入函数 user—input()

此部分程序功能为接收并记忆用户的输入,实现用“↑”、“↓”键来恢复以往命令。

用户是通过终端与操作系统或应用程序发生交互作用的。XENIX 操作系统中的终端用特殊文件来表示,它也可象普通文件的操作方式一样来进行读写。本程序用 open()、read()和 write()等系统调用来实现程序与终端进行信息交换。

由于不同的终端口有不同的设备文件名(如/dev/ttyoa、/dev/ttyob等),所以本程序读写的终端为控制终端(/dev/tty),控制终端指向启动、控制某用户进程的用户终端,而不需知道它的确切设备名。

终端有多种工作模式,在标准 shell 中,终端被设置成正则模式。此时,若某程序读终端的输入,是先由操作系统内核的终端驱动程序接收终端键盘输入的字符存储在缓冲区里并向终端作回送操作,内核还根据键盘输入完成相应的动作(如左删字符等),在接收到用户的回车键时才将用户输入的一串字符传送给读该终端输入的程序。

user \_\_input()中用系统调用 ioctl()将终端改设成原始工作模式,开关团 ECHO。这样,每当用户按下一个键,内核的驱动程序能立即把键值传送给本程序,本程序根据用户按下的键作相应的处理;若是“↑”、

“↓”，则恢复已存在的以往命令；若是“←”、“→”，就在当前命令缓冲区中左、右移动光标；若是“Bs”，则删除光标前的字符；若接收到有字符，则将此字符插入在当前光标位置上（当前光标位置由变量 `pointer` 指示）；若接收到回车键，就将用户的输入形成字符串并返回；若接收到“Ctrl+D”，则返回“EOF”。

退出 `user_input()` 之前，终端被恢复成原正则模式，以保证原 shell 支持的各种实用程序能在本界面程序中正常使用。

本程序在 `in_buf[][]` 二维数组来保存以往输入的命令，其数组的大小决定了能被保存的最大命令行数，其中 `in_buf[o][o]` 为当前输入缓冲区。每当用户输入一个命令后，就依次将二维数组往后移，以保存用户的输入；定义变量 `now` 为指向被恢复命令的指针，当用户输入“↑”、“↓”键时，则相应增减 `now` 的值并把 `in_buf[now][]` 的字符串移至 `in_buf[o][]`，作为用户的当前命令。

#### (2) 命令运行程序 `run_command()`

此程序对 `in_buf[o][]` 中的命令行进行分析并执行。由于 XENIX 允许在一行命令中有多条命令，命令之间用“;”分开，故本程序定义 `now_cmd` 和 `next_cmd` 为当前命令和下次命令的指针，并循环地对用户命令行进行分析，以修正 `cow_cmd` 的值，再运行当前命令，如此往复，直至 `next_cmd=NULL` 时退出。

本程序不直接调用相应模块中的程序来运行用户的命令，而是调用 `/bin/sh` 程序来执行用户命令，这样可保证本界面程序与原标准 shell 的完全兼容，并且原 shell 的一些优越特性（如管道、后台命令等）也能方便地得到继承。

具体方法是：用功能调用 `fork()` 建立一个新的子进程，随后本进程执行 `wait()` 操作以等待新建子进程的运行结束；在子进程中执行：

```
exec 1("/bin/sh", "sh", "-c", now_cmd, (char
```

```
* )0);
```

调用 `/bin/sh` 来执行 `now_cmd` 指向的命令。当命令执行完后，新建的子进程结束，父进程被唤醒转去继续分析用户命令并再次运行，直至 `next_cmd=NULL` 时退出 `run_command()`。

“cd”命令不能用上述办法来执行。因为当前工作目录是进程的一个属性，若 `cd` 命令在子进程中被执行后，其父进程（即用户界面）的工作目录并不会随之改变。因此对“cd”命令，本程序用 `chdir()` 功能调用来完成。

#### (3)、中断信号处理

XENIX 中，用户可用中断键来中止某程序的运行。用户按下中断键后，系统内核接收到中断符，便向发来中断符的终端所控制的所有进程发出 `SIGINT` 中断信号，收到此信号的进程便做相应的处理。可见，用户按中断键后，本界面程序也同时收到 `SIGINT` 信号，但界面程序不应被此信号中断，故本程序初始化时用 `signal(SIGINT, SIG_IGN)` 系统调用来忽略中断信号作用，用户命令应允许被中断，故 `run_command()` 在生成的子进程中调用 `/bin/sh` 之前，用 `signal(SIGINT, SIG_DFL)` 来恢复子进程中对中断信号的默认处理。

#### 三、新界面程序的启动

本程序编译连接后以 `/bin/qccsh` 名存放，它可按下述方法启动：

① 不做任何改变，登录后输 `qccsh` 可进入本界面程序。

② 修改用户的 `.profile` 文件，在文件尾增加 `qccsh` 一行命令，这样用户一登录便能自动启本程序。

③ 修改口令文件 `/etc/passwd` 中对每一用户描述行中的最后字段，将原 `/bin/sh` 改为 `/bin/qccsh`，登录后系统便启动本程序作为用户界面。

需此程序清单者，请同本刊联系。



## 程序设计语言计算机题库系统的研究

成都科技大学计算中心 陈碧人

成都科技大学电力工程系 龙廷河

**摘要** 本文根据命题科学化、规范化的原则,研制出符合实际考试要求的计算机程序语言题库管理系统。该系统由数据库、过程文件及系统程序三大部分组成。组合试卷时,采用人机对话方式设计出各类题数后,便自动组合成考卷。该系统可在没有专门的激光排版之类的软件前提下,输入输出各种数学符号和数学式子。

程序设计语言是计算机的主要基础课之一。要迅速提高我国本科生计算机科学知识的水平,增强程序设计及计算机使用的能力,必须把握各个教学环节,全面规划,搞好课程建设,才能不断提高教学水平。考试是一个重要的教学环节,也是课程建设的组成部分。根据命题科学化及标准化的原则,我们力图研制出符合实际考试要求的程序设计语言的题库管理系统。题库管理系统是在计算机上建立题目库(简称题库)。然后根据一定的要求从题库中选取题目构成一份试卷。建立计算机题库管理系统,是实现考试命题科学化及标准化的重要手段,利用题库管理系统,可以实现教学质量评估,并且可大大减轻教师命题工作中的工作量。

题库虽是存放在计算机存贮介质中的习题集合,但却不同于习题集。因为在计算机上操作时,各操作程序可使习题集变“活”,成为教师搞好教学的得力助手。

计算机程序设计语言题库管理系统由数据库、过程文件及系统程序三大部分组成。题目的存贮和组合试卷是题库管理系统的关键。

### 一、数据库

试题库是试卷生成系统的基础。试题的构成对学生的学学习有很强的指导作用和反馈效果。根据 FORTRAN 程序设计语言教学大纲的要求,我们把多年来积累和收集到的试题,经过分类排队,精心筛选,设计了一个能满足不同层次考试要求的试题库。为了使试题能较全面的反映教学大纲的要求,我们将试题按程序设计语言传统分为:填空题、读程题、改错题、编程题等四种题类。为了使试卷难易程度适当,也为了满足对不同层次的学生出不同难度的题目的要求,特地设置了难度参数,难度参数反映题目的高、中、低级水平,适当控制此参数,使试卷难易适当,并能满足不同层次的要求。

为了使学生全面掌握好教材中各重要章节的内容,对于编程题专门加设章节参数进行控制。章节分选择结构、循环结构,数组和辅程序等四部分。通过合理

控制章节的参数,可使每份试卷中的试题合理分布,覆盖面广。

相关性参数的控制。题库中的试题内容当然是各不相同的,但某些类型的试题虽然内容不同,但却都属于考试范围的一种题型。在一份试卷中,让其相同的题型最多只能出现一道题,这样,就会使试卷中的题具有普遍性和代表性。为了达到这一目的,题库设置一个相关性参数,相关的试题设置相同的相关数值,不相关的试题赋有不同的相关数值。这样,通过程序的控制,避免题型相同或相似的试题出现在同一份试卷中。

### 二、过程文件

因为程序设计试题中,有读程序题(即事先编好一段程序,让学生阅读程序后,写出程序运行结果。),这种读程序的题,所含语句的多少差异很大,少的可能只有几条语句,(输出在纸上时,只占用几行)多的可能有 50 多条语句,(打印在纸上,需占用 50 多行)。如果将所有这些试题都存放在数据库中,则要设置很多字段,才能满足这种“长题”的需要,但是,这对于那些“短题”来说,不免造成存贮空间的极大浪费。又因为程序的书写格式要求严格,每行最多只能书写一条语句,所以,又不便于用记忆型字段来存放试题内容。经过实践摸索,我们采用数据库存放短题,对于长题,则用数据库与过程文件两者相结合的办法,即用数据库存放试题各种参数,而试题的具体内容则作为过程文件的一个过程。选题组合试卷时,只需对数据库中参数进行控制和选择,再通过程序的控制,自动将过程文件中的题组合成试卷。这样,每道长题所占用的存贮空间是由这道题的具体字答数所决定的,既能满足每道题长短的不同需要,又能节约存贮空间,同时又解决了 FORTRAN 语言每行只能书写一条语句的格式要求。所以,打印试卷时,也不需要专门的控制,就能满足 FORTRAN 试卷的各种格式要求。

### 三、系统程序

题库管理系统程序主要由组合试卷,浏览选择、题

库管理,打印输出和试题输入等五大部分组成。

### 1. 组合试卷程序

组合试卷程序是系统程序的中心环节。

指导思想是:试卷具有良好的结构,试题分布均衡合理,试题的难度适当且具有明显的坡度。

组合试卷一开始,屏幕上自动显示一个画面,左半部显示题库中各种没有选用过的题类的数量,右半部提示和引导用户选择各种题的数量,然后,程序根据用户所选择的各种类型的题数,自动在各数据库及过程文件中,选择符合要求的试题,而且自动将其组合在试卷中。因为在这一屏信息中,题类、难度、章节,题数都已进行了设计,所以,程序运行结束,便自动按要求组合成了试卷。

组合成的试卷,对于读程序题和编程题,可以保证选择的试题在五年内不重复,当题库中收集的题更多时,可以保证在更长的时间内不出现重复,这是因为数据库和程序中,用“选否”参数进行控制,当某道题一旦选中了,则它的“选否”字段值将变为 Y,以后将不再被选用,只有当各参数重新初始化后,选否参数的值变为 N,才有可能被选用。

### 2. 浏览组合试卷

如果因为某种特殊的需要,也可以进行浏览选题。这时,将要选择的题类按顺序一道题一道题的显示在屏幕上,如果选中了某题,只需用人机对话方式回答字母 y 即可。

### 3. 题库管理

题库管理程序主要功能有:

- (1) 对各题库中的参数进行初始化;
- (2) 计算各题库中各种题类的数量,以便用户设计各种题类的数量时,心中有数;

### (3) 建立各种索引文件

### (4) 对题库的增、删、派生题等进行操作。

### 4. 打印输出

系统提供两种打印方式,一种是试卷打印,另一种是试题打印。

(1) 试卷打印在组合和编辑好试卷后,要打印输出试卷。输出时,在试题的前面,自动加上一个试卷头,试卷头包括试卷名、考试时间以及各道题的记分表格,接着再按填空、读程序、改错及编程序的顺序输出试卷。

(2) 试题打印 将题库中的试题按存放顺序从头

到尾的输出,可以只输出试题的内容,也可以在输出试题前先输出每道题的各种参数,再输出试题。

### 5. 试题输入部分

此程序模块主要输入和修改题库及过程文件中的试题。

## 四、系统的特点

1. 系统采用屏幕菜单的方式,引导用户使用系统,所以,即算不了解该系统的人,也能命题,组合打印试卷。

2. 在没有任何专门软件的前题下,能完成各种数学符号,数学式子的输入输出。

程序设计语言的一个重要应用方面,是完成数值计算,因此,程序设计语言的试题,肯定要涉及到许多数学式子和数学符号。用激光排版等专门软件,当然可以毫无疑问的输出各式各样的数学符号,数学式子。但是,这不但要花费一笔款子购买专门的软件,更重要的是,必须要在配有硬盘的计算机上,才能进行操作和运行,这就使没有配硬盘的计算机单位,不可能建立和使用题库管理系统,这不能说不是—种莫大的遗憾。笔者也正是在这种情况下,通过不断摸索,终于在—没有硬盘的 IBM PC 机输入输出了各种数学符号和数学式子。

具体分为以下三个步骤:

第一,使用 CCDOS4.0 版本的造字程序,造 27 个特殊符号,如  $\pi$ 、 $\alpha$ 、 $\beta$ 、 $\geq$ 、 $\leq$ 、 $\{$ 、 $\}$ 、 $[$ 、 $]$ 、还有开方符  $\sqrt{\quad}$ 、积分符  $\int$ 、 $\sum$ 、下画线——、乘号·、分数线——等,存放在 CCDOS4.0 字库中。这种造字,当然只是一次性的劳动。

第二,在需要输入特殊数学符号时,用区位码将这些数学符号选出。值得强调的是,在描述数学式子时,有时需要几个符号组合起来进行描述,如数学式  $\sqrt{a}$ ,它分别由下画线——,符号  $\sqrt{\quad}$ ,和  $a$  三部分组成,且下画线必须在上一行的适当位置上, $\sqrt{\quad}$ 和  $a$  在下一行上,通过输入时的位置控制,将这三个符号组合在一起,即描述了  $\sqrt{a}$ 。正因为采用这种组合方法,才使我们在仅造 27 个符号的前题下,就能完成任何数学式子的描述。

第三,输出时,用程序控制纵行的距离,使输出效果与数学式子相同。需此软件者,同本刊联系。

# 开环数控系统消除积累误差的软件设计方法

甘肃省计算中心 李天健 兰州大学数学系 翟忠信

**摘要:** 开环系统由于没有反馈,每点的坐标取决于计算精度,所以往往形成积累误差。本文从软件设计的角度考虑如何解决这一问题,为类似数控软件的设计提供参考。

## 一、问题的提出

开环系统由于其实现容易,价格较低,所以在数控系统中占有一定的比例。但开环系统无一例外都会遇到积累误差的问题,尤其是在加工一些复杂曲线时,须严格按照其解析式进行加工,在计算过程中必然会碰到更为严重的积累误差的问题。

现举例说明。如脉冲当量的设计值为:  $\Delta h = 0.005\text{mm/步}$ ;拟采用4字节浮点运算,其字长可达32位,除符号位与整数部分所点位外,取小数点后位数为23位,所以其精度可达  $2^{-23} = 1.19 \times 10^{-7}$  即每步精度可达  $1.19 \times 10^{-7}\text{mm}$ ;如果其最大加工长度为300mm;最大步数则应为  $300/0.005 = 6 \times 10^4$  步;最大误差为  $1.19 \times 10^{-7} \times 6 \times 10^4 = 7.14 \times 10^{-3}\text{mm}$ 。

可见,在没有计算误差时的总积累误差很小,一般不必考虑。但考虑到每次加工的步数必须是整数,现在来看实际加工时的计算方法。

设已知曲线的解析式为:

$$y = f(x) \quad (1)$$

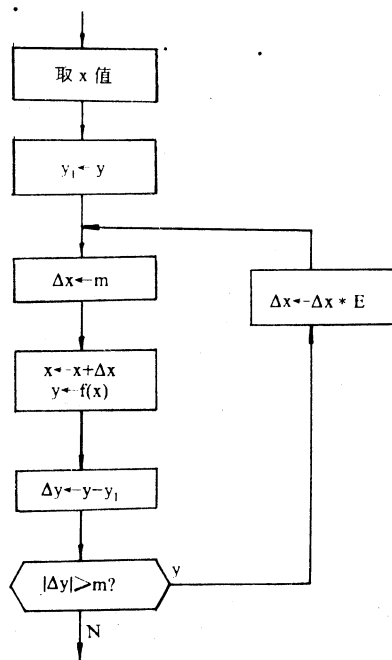
或由参量方程

$$\begin{cases} y = \varphi(t) \\ x = \xi(t) \end{cases} \quad (2)$$

来表示。设每次加工长度不超过  $m$ , 即  $\Delta x = m$ ; 由式(1)可求出  $\Delta y$  ( $\Delta x, \Delta y$  为每次  $x, y$  轴向的加工量)。若求出的  $\Delta y > m$ , 则应调整  $\Delta x$  直到  $(\Delta x \text{ and } \Delta y) \leq m$  为止。其程序框图可用图一表示。

因为采用 CNC 技术,做到这一点并不难。但每次计算出来的  $\Delta x, \Delta y$  存在误差,其值为  $3 \times 1.19 \times 10^{-7}\text{mm}$  ( $\Delta x, \Delta y$  的求得要经过三次运算,每次运算的精度为  $1.19 \times 10^{-7}\text{mm}$ )。此外,每次运算的步数不可能是非整数,所以每次运算的最大误差是  $1/2$  步,即  $0.0025\text{mm}$ ;若经过 100 次运算,其积累误差可达  $100 \times$

$(0.0025 + 3.57) \approx 0.25\text{mm}$ ;这在数控系统中已是不可忽视的大误差了,所以必须采用消除积累误差的方法。

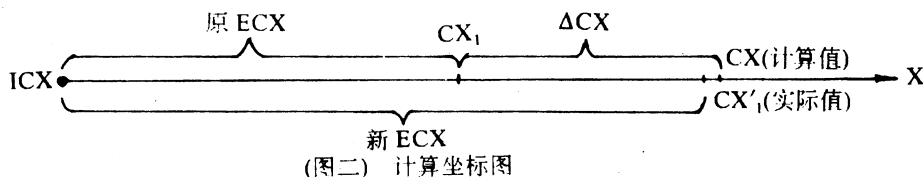


图一. 进刀量调整框图

从以上计算分析中我们还可以看到,脉冲当量越粗,计算步骤越复杂,其后果越严重。

## 二、解决的方法

解决积累误差的基本思想是每次计算下一个加工点(简称新点)的坐标值时不采用上一个加工点(简称旧点)的数据为基础,而是采用起始点为参考点。我们可采用一个坐标图来说明这个思想(见图二)



图二, 计算坐标图

图中：

ICX 加工起始点坐标，

$CX_1$  旧点坐标，

$CX$  新点坐标计算值，

$CX'$  新点坐标精确值，

$\Delta CX$  本次加工位移计算值，

ECX 起始点与实际加工到达点之间位移的精确

值。

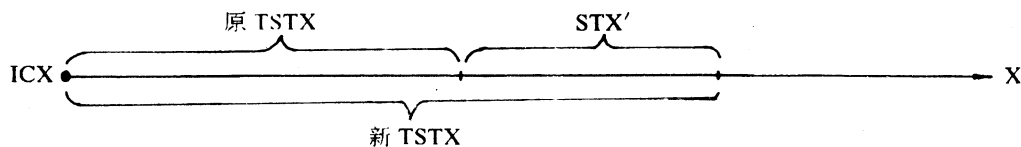
TSTX 当前点到起始点之间已走过的总步数，

STX' 旧点到新点精确值处所走的步数。

$\Delta h$  x 方向的脉冲当量。

(在下文将要引用的 y 方向的符号含义与 x 方向基本一改，其脉冲当量为  $\Delta v$ )

为了使步数与坐标间的关系表达得更清楚，我们以如下步数图加以描述(见图三)。



图三, 步数图

计算的步骤如下：

由计算获得本次加工位移量  $\Delta CX$  后得：

$$STX' = \Delta CX / \Delta h \quad (3)$$

$$\text{新 TSTX} = \text{原 TSTX} + STX' \quad (4)$$

$$\text{新 ECX} = \text{新 TSTX} \cdot \Delta h \quad (5)$$

$$CX_1' = ICX + \text{新 ECX} \quad (6)$$

显然这种计算方法的优点是明显的，它每次加工后的新坐标值不是通过直接计算公式

$$CX = CX_1 + \Delta x \quad (7)$$

来进行计算，而是由公式(6)来完成即：

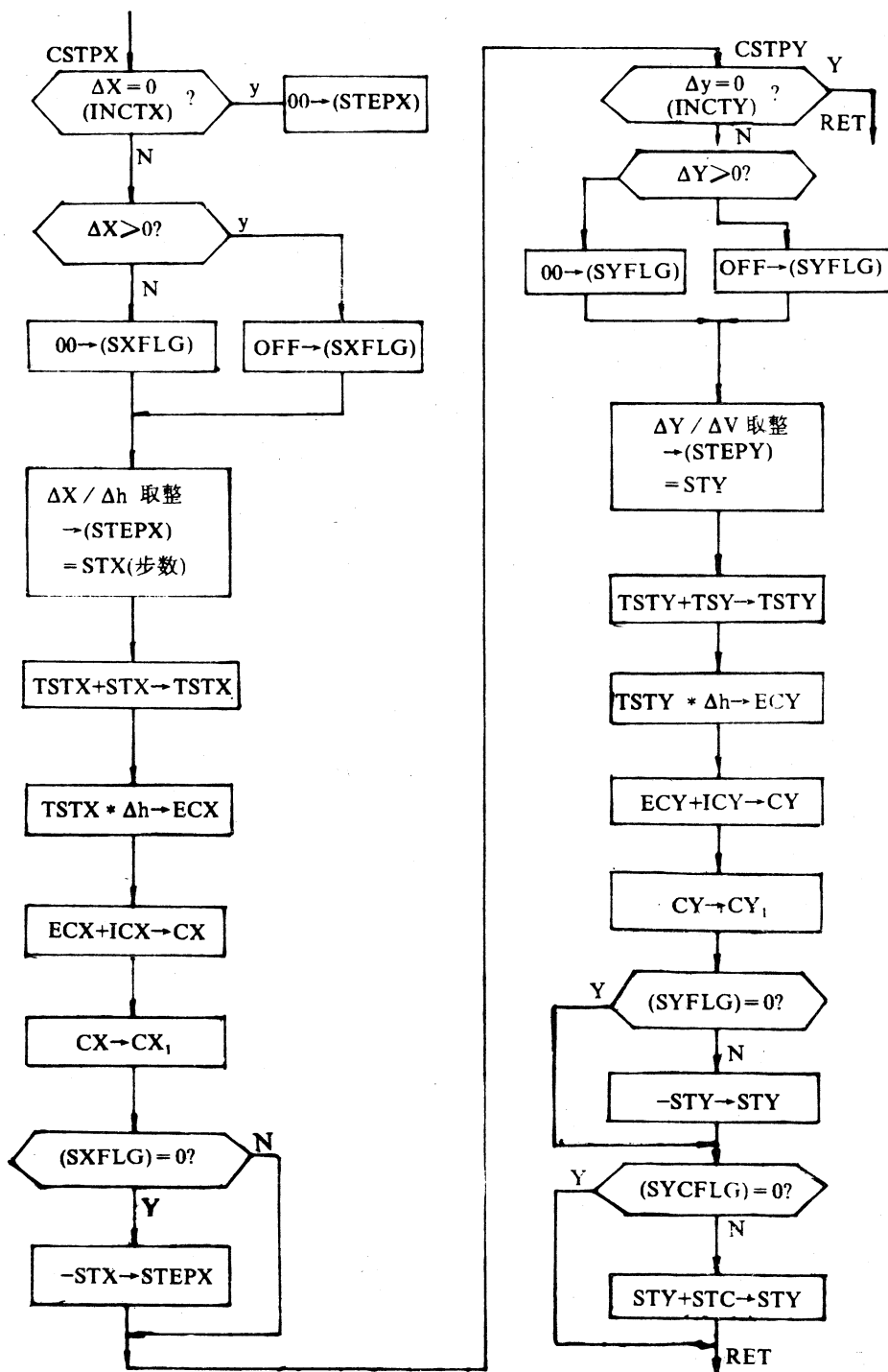
$$\begin{aligned} CX_1' &= ICX + \text{新 ECX} \\ &= ICX + (\text{原 TSTX} + STX') \cdot \Delta h \end{aligned} \quad (8)$$

(8)式中的(原 TSTX + STX')是一个精确数，所以新坐

标  $CX_1'$  中没有上次计算所留下的积累误差，如果有误差的话，也仅仅是  $\Delta h$  本身的误差，这一点前面已讨论过了。

这样做还有一个极重要的优点，即  $CX, CY$  只与走过的总步数及起始点坐标相关；这样，在加工过程中  $CX, CY$  始终与走过的总步数 TSTX 相对应，无论中途如何改变  $CX, CY$  的设定值，都不会改变  $CX, CY$  的绝对坐标。(注意：ICX 一经设定不能随意变动。)这为定位系统软件设计提供了强有力的支持。

精度保持的关键是计步子程序，上面的图四给出了这个子程序的框图，以阐明上述思想的软件实现方法。



图四,计步子程序框图



## 也论多种程序语言的综合应用

中国矿业大学北京研究生部 黄绍东 陈宜金

贵刊 1991 年第二期刊登的“试论多种程序语言的综合应用”一文中,作者钟廷姣同志提供的在批处理程序中采用“虚拟文件”式的接口技巧,在多环境条件下的结构化程序设计中,对于实现程序控制和不同运行环境的切换,以实现功能模块的连续,是一种行之有效的实用方法。

众所周知,在结构化程序设计中,如果所开发的系统规模较小,只有简单的数据处理和科学运算,则只需选择一种高级语言或数据库语言在单一的环境中即可达到预期目的。这时只需通过模块调用和参数传递,便可实现模块连接。反之,对于功能比较齐全的复杂系统,则需根据各功能模块的不同特点和要求,选择不同的程序设计语言方能开发出高质量的软件系统。在这种多语种的混合编程中,如果各自的运行环境不同,采用(钟文)提供的“虚拟文件”接口技术实现环境切换和模块连接确实是一种很实用的方法。

但是很遗憾,(钟文)提供的方法中,回避了一个最为重要的核心问题,即当第二层批文件(子批文件)执行结束后是否需要返回主批文件?以及如何返回主批文件?因而(钟文)提供的方法和示范程序并不能起到切换环境和连接模块的作用。因为这种方法的基本特点是由批文件串连起来的程序文件可以是另一个批处理文件,即其实质是利用批文件的嵌套调用来实现系统的调度和环境切换的,因此只有当下一级批文件执

行结束后必须能够返回上一级批文件方能达到预期目的。否则将导致系统执行过程中的中途返回 DOS,而不能起到程序控制、环境切换和连接模块的作用,因而也就使得(钟文)立论的本来目的失去意义。那么,在批文件的嵌套调用中,如何实现返回呢?可以采用下列两种方法之一达到这个目的:

1. 利用 DOS 的 COMMAND.COM 命令加载子批文件:

大家都知道,COMMAND.COM 是 DOS 的主要组成部分之一,由于它也是一个.COM 文件,因此它也可以像其它.COM 文件一样,在批文件中作为一条外部命令来使用。这样就可以在高一级的批文件中把下一级批文件作为第二份 COMMAND.COM 的命令行参数进行加载,从而实现下一级批文件执行结束后,自动返回上一级批文件的目的,其格式如下:

COMMAND /C BATCH

式中的“/C”为实现返回上一级批文件而设置的开关参数,BATCH 为被调用的下一级批文件名。

2. 利用 CALL 命令调用下一级批文件:

如果使用 3.2 以上版本的 DOS(不含 3.2 版本),可以在上一级批文件中直接使用 CALL 命令调用下一级批文件,即:

CALL BATCH

另外,(钟文)采用了解释型的 BASIC 程序作为生

图中以两维平面加工为例,X 表示左右加工方向,y 表示上下加工方向。其中:

CSTPX 左右加工计步程序入口,

CSTPY 上下加工计步程序入口,

INCTX  $\Delta x$  的内存单元,

INCTY  $\Delta y$  的内存单元,

STEPX 左,右方向标志,

STEPEY 上,下方向标志,

STC 上下方向间隙补偿量,

SYCFLG 上下方向反向判别标志。

由于上下运动存在机械复位误差,所以设置了间隙补偿量 STC 和反向判别标志。其它符号与前面介绍

的一致。

### 三、结论

这套软件在理论上可使计算中的积累误差为零。在将其实施到数控系统后,观察到的情况是令人满意的。虽然在存在机械传动误差的情况下,我们无法证实其计算积累误差为零,但通过对比检测长位移加工与短位移加工的总误差值,发现两者非常接近,所以可以肯定其计算积累误差是接近为零的。

### 参考文献

1. B、C、KVO 等著,王宗培等译,步进电机及其控制系统,哈尔滨工业大学出版社,1984. 9

2. John D. lenk, Handbook of Microcomputer — Based instrumentation and controis, Prentice—Hall. Inc. 1984

成“虚拟文件”的总控文件,根据我们在实践中的体会,总控文件不宜采用高级语言来编写。由于在运行由解释型的高级语言生成的总控文件时,需要解释系统的支持,因此不仅运行速度慢,而且占用磁盘空间也很大。即使使用编译型高级语言进行编写,生成的文件长度也将达到 30KB 字节左右。所以最好采用汇编语言进行编写,这样既精练(文件长度一般都小于 1KB 字节),执行速度也快得多。

本文所附之程序(一),是对<钟文>所附之程序经修改后的示范程序(LDXT. BAT),它是一个具有嵌套调用关系的批处理文件。当下一层批文件执行结束之后,自动返回并继续执行上一级批处理文件,真正起到程序控制、切换环境和连接模块的作用。程序(二)是按 .COM 格式用汇编语言编写的总控块(也称主菜单文件),经编译、链接,生成 .EXE 文件后,再用 DOS 的 EXE2BIN. EXE 命令转换为 .COM 格式的可执行文件。对于低一级的子菜单文件,由于其格式和主菜单文件完全相同,所以只需对主菜单文件中的菜单项加以修改,即可方便的得到子菜单文件。

程度(一)中的 0、1、2、3、4 为“虚拟文件”的名称,它是在总控程序的运行中,以会话方式从键盘上输入的菜单号(DOS 系统功能调用 21H 的 0AH 子功能),然后以文件的方式写入磁盘的(21H 的 3CH 子功能)。

程序(一): LDXT. BAT

```
ECHO OFF
CLS
: LOOP
ZK. COM
IF EXIST 1 GOTO L1
IF EXIST 2 GOTO L2
IF EXIST 3 GOTO L3
IF EXIST 4 GOTO L4
IF EXIST 0 GOTO L0
: L1
ERASE 1
DBASE KCZ
GOTO LOOP
: L2
ERASE 2
COMMAND /C FXXT(采用 DOS 3.3 时,可为 CALL FXXT)
GOTO LOOP
: L3
ERASE 3
COMMAND /C HXXT(采用 DOS 3.3 时,可为 CALL HXXT)
```

GOTO LOOP

: L4

ERASE 4

BASICA TUXT

GOTO LOOP

: L0

ERASE 0

ECHO ON

程序(二): ZK. ASM

CODE SEGMENT

ORG 100H

ASSUME CS: CODE, DS: CODE

BEGIN PROC NEAR

MOV AX, 0006H

INT 10H

MOV DX, OFFSET MENU

MOV AH, 9

INT 21H

MOV DX, OFFSET SELECT

MOV AH, 9

INT 21H

MOV DX, OFFSET BUFFER

MOV AH, 0AH

INT 21H

MOV [BUFFER+3], 0

MOV AH, 3CH

MOV CX, 0

MOV DX, OFFSET BUFFER+2

INT 21H

MOV AH, 4CH

INT 21H

BEGIN ENDP

MENU 0DH, 0AH, '系统主菜单'

DB 0DH, 0AH, '-----'

DB 0DH, 0AH, ' 1、数据管理'

DB 0DH, 0AH, ' 2、绘平面图'

DB 0DH, 0AH, ' 3、绘等值线'

DB 0DH, 0AH, ' 4、数据处理'

DB 0DH, 0AH, ' 0、返回 DOS'

DB 0DH, 0AH, '-----'

SELECT 0DH, 0AH, '请选择功能菜单: \$'

BUFFER

DB ?

DB 3 DUP (0)

CODE ENDS

END BEGIN

# 实用高分辨率彩色图形制作工具

四川省达县地区信息中心 余跃仁

## 摘要:

本文为使用高分辨率彩色显示器的应用程序设计人员提供了一种快速、方便、有效的交互式图形制作工具。该工具软件绘制的图形丰富多采,并能配上特大汉字显示,使应用软件拥有友好的用户介面;交互性强,操作方便,大大减少了软件开发的时间和费用。

## 设计原因及目的

一种软件要想受到广大用户的青睐并拓宽其使用面,最基本的要求是该软件必须具有友好的用户介面。由于人们所获取的信息 70% 以上来自视觉,为此,高分辨率彩色显示器应运而生,它为我们奠定了达到上述要求的硬件基础。但实际上我们在使用高分辨率彩色显示器时常常未能充分发挥其功能,原因是没有较实用的彩色图形制作工具软件。少数用户借助于一些价格昂贵的硬设备,但大多数人便只能把一条条命令拼凑起来,而为了达到满意的视觉效果,需要反复调试多次,真是浪费人力物力。

因此,笔者用 BASIC 语言结合机器语言编制了一组实用的高分辨率彩色图形制作程序,它能快速方便地用来全屏幕编辑诸如软件封面、地图、数据图形、人物肖像等丰富多采的高分辨率彩色图形和进行特大汉字显示,并将图形存盘,以备应用软件调用。该程序包括以下三个部分:

## 一、制图及存图程序 YYRSAVE. BAS

该程序运行后屏幕中心出现一个光标,实际上是一个小白点,当按下某个颜色键时,(屏幕上已显示出颜色与按键的对应关系如程序的 10、20 语句),光标将改变颜色,共有 16 种颜色。按下方向键时(屏幕已显示出方向与按键的对应关系如程序的 30 语句),光标将移动并画出图形的边框,共有 8 个方向。当图形的封闭边框画好后,把光标移动到边框内任何一点(一般选择与边框邻接的内点),按“X”键后再按一颜色键便可在边框内填上所选颜色。需要画圆时,把光标移动到圆心处,按“C”键后再输入半径就完成了。当你按“Z”键退出该程序时,屏幕图形仍保持原样,只是图形没有存盘,再次运行该程序又可继续作图。当然,有些大面积的矩形可退出该程序直接用 LINE 语句完成也行。如果需要特大汉字显示,可调用特大汉字显示程序。当整

个画面完成后,按“&”键把图形存入 C: 盘上的 PICTURE. PIC 文件中。

## 二、特大汉字显示程序 YYRHZANZI. BAS

该程序运行后首先提示:“输入汉字”,这时可输入你需要放大显示的汉字串。然后依次输入字号、颜色、显示起始行、显示起始列,该程序就自动提取显示用字符点阵进行放大后以图形方式显示在屏幕所定位置上。注意:该程序中有一段机器语言子程序,作为提取字符点阵用,反汇编后如程序四。

## 三、调用图形程序 YYRLOAD. BAS

该程序用于快速地将文件 PICTURE. PIC 中的图形映射到屏幕上,实际上就是应用软件的一段子程序。如果应用程序是用 BASIC 语言编写的,只要将该程序加入应用程序即可;如果是其它语言编写的应用程序,可建立一个批处理文件,并在其中加入 BASIC YYRLOAD 来实现。

## 5 REM 制图及存图程序 YYRSAVE. BAS

```
10 PRINT " 0=黑 1=蓝 2=绿 3=青 4=红 5
=洋红 6=棕 7=白 8=灰 Q=淡蓝 W=淡绿"
```

```
20 PRINT " E=淡青 R=淡红 T=淡洋红 Y=
黄 9=高强度白 &=存盘退出 Z=不存盘退出"
```

```
30 PRINT " X=填色 J=左 U=左上 L=右 0
=右上 I=上 >=右下 <=下 M=左下 C=画圆"
```

```
40 M=320: N=220: COL=15: T=0
```

```
50 IF T=1 THEN T=0: COL1=COL: COL=CO
: GOTO 130
```

```
60 PSET (M,N),COL
```

```
70 A$=INKEY$: IF A$="" THEN 50
```

```
80 IF A$="Z" THEN END
```

```
90 IF A$="&" THEN 410
```

```
100 IF A$ "<>" "X" THEN 150
```

```
110 A$=INKEY$: IF A$="" THEN 110
```

```

120 CO=COL : T=1 : GOTO 150
130 PSET (M,N),0 : PAINT (M,N),COL1,COL
140 GOTO 50
150 IF A$="0" THEN COL=0 : GOTO 50
160 IF A$="1" THEN COL=1 : GOTO 50
170 IF A$="2" THEN COL=2 : GOTO 50
180 IF A$="3" THEN COL=3 : GOTO 50
190 IF A$="4" THEN COL=4 : GOTO 50
200 IF A$="5" THEN COL=5 : GOTO 50
210 IF A$="6" THEN COL=6 : GOTO 50
220 IF A$="7" THEN COL=7 : GOTO 50
230 IF A$="q" THEN COL=9 : GOTO 50
240 IF A$="w" THEN COL=10 : GOTO 50
250 IF A$="e" THEN COL=11 : GOTO 50
260 IF A$="r" THEN COL=12 : GOTO 50
270 IF A$="t" THEN COL=13 : GOTO 50
280 IF A$="y" THEN COL=14 : GOTO 50
290 IF A$="9" THEN COL=15 : GOTO 50
300 IF A$="8" THEN COL=8 : GOTO 50
310 IF A$="C" THEN INPUT C : CIRCLE(M,
N),C,COL,,,1 : GOTO 50
320 IF A$ <> "j" AND A$ <> "1" AND A$ <
> "i" AND A$ <> "," AND A$ <> "u" AND A$ <
> "o" AND A$ <> "." AND A$ <> "m" THEN 70
330 IF A$="j" THEN M=M-1 : GOTO 50
340 IF A$="1" THEN M=M+1 : GOTO 50
350 IF A$="i" THEN N=N-1 : GOTO 50
360 IF A$="," THEN N=N+1 : GOTO 50
370 IF A$="u" THEN M=M-1 : N=N-1 :
GOTO 50
380 IF A$="o" THEN M=M+1 : N=N-1 :
GOTO 50
390 IF A$="." THEN M=M+1 : N=N+1 :
GOTO 50
400 IF A$="m" THEN M=M-1 : N=N+1 :
GOTO 50
410 PRINT "正在存盘,请稍候....."
420 OPEN "grp" FOR OUTPUT AS #1
430 WRITE #1,"g",0,0,640,240,0,&H4000
440 CLOSE 1
450 DEF SEG=&H4000
460 BSAVE "c : picture1. pic",0,&HE200
470 OPEN "grp" FOR OUTPUT AS #1
480 WRITE #1,"g",0,241,640,480,0,&H5000

```

```

490 CLOSE 1
500 DEF SEG=&H5000
510 BSAVE "c : picture2. pic",0,&HE200
520 END
5 REM 特大汉字显示程序 YYRHZANZI. BAS
10 DEFINT A-Z
20 DEF SEG=&H8000
40 BLOAD "a : thz",&H100
50 SUBRT=100
55 CALL SUBRT
500 DIM A$(15)
510 OPEN "abcde" AS #1 LEN=1
520 FIELD #1,1 AS A$
530 CLS
540 INPUT "请输入字号(1-9) : ";L
550 INPUT "请输入颜色(1-9) : ";M
555 INPUT "请输入起始行(0-450) : ";U
556 INPUT "请输入起始列(0-640) : ";V
560 K=1
580 CLS
590 Y=U
600 FOR J=32 * K-31 TO 32 * K-16
610 GET #1,J
620 B=ASC(A$)
630 IF EOF(1) THEN 900
640 GET #1,J+16
650 C=ASC(A$)
660 FOR I=7 TO 0 STEP-1
670 IF C>=2^I THEN 700
680 A$(I)=" "
690 GOTO 720
700 A$(I)="*"
710 C=C-2^I
720 IF B>=2^I THEN 750
730 A$(I+8)=" "
740 GOTO 770
750 A$(I+8)="*"
760 B=B-2^I
770 NEXT I
780 X=16 * (K * L-L)+V
790 FOR I=15 TO 0 STEP-1
800 IF A$(I)="*" THEN 820
810 GOTO 830
820 LINE(X,Y)-(X+L,Y+L),M,BF

```

```

830 X=X+L
840 NEXT I
850 Y=Y+L
860 NEXT J
870 K=K+1
880 GOTO 580
890 CLOSE #1
900 END
5 REM 调用图形程序 YRLOAD.BAS
10 SCREEN 0
20 DEF SEG=&H4000
30 BLOAD"c: picture1. pic",0
40 OPEN"grp"FOR OUTPUT AS #1
50 WRITE #1,"u",0,0,0,&H4000,3
60 CLOSE 1
70 DEF SEG=&H5000
80 BLOAD"c: picture2. pic",0
90 OPEN"grp" FOR OUTPUT AS #1
100 WRITE #1,"u",0,241,0,&H5000,3
110 CLOSE 1
120 END
提取汉字点阵子程序 THZ.BAS
2803: 0107 50  PUSH  AX
2803: 0108 53  PUSH  BX
2803: 0109 51  PUSH  CX
2803: 010A 52  PUSH  DX
2803: 010B 54  PUSH  SP
2803: 010C 55  PUSH  BP
2803: 010D 56  PUSH  SI
2803: 010E 57  PUSH  DI
2803: 010F 1E  PUSH  DS
2803: 0110 06  PUSH  ES
2803: 0111 16  PUSH  SS
2803: 0112 0E  PUSH  CS
2803: 0113 1F  POF  DS
2803: 0114 0E  PUSH  CS
2803: 0115 07  POP  ES
2803: 0116 90  NOP
2803: 0117 90  NOP
2803: 0118 B409 MOV  AH,09
2803: 011A BA8001 MOV  DX,0180
2803: 011D CD21  INT  21
2803: 011F B40A MOV  AH,0A
2803: 0121 BA9001 MOV  DX,0190
2803: 0124 CD21  INT  21
2803: 0126 8CCD MOV  BP,CS
2803: 0128 BBA001 MOV  BX,01A0
2803: 012B B89001 MOV  AX,0190
2803: 012E 53  PUSH  BX
2803: 012F 50  PUSH  AX
2803: 0130 5B  POP  BX
2803: 0131 58  POP  AX
2803: 0132 83C302 ADD  BX,+02
2803: 0135 052000 ADD  AX,+0020
2803: 0138 50  PUSH  AX
2803: 0139 53  PUSH  BX
2803: 013A 8B1F MOV  BX,[BX]
2803: 013C 80FF0D CMP  BH,0D
2803: 013F 7411  JZ   0152
2803: 0141 80FB0D CMP  BL,0D
2803: 0144 740C  JZ   0152
2803: 0146 88FA MOV  DL,BH
2803: 0148 88DE MOV  DH,BL
2803: 014A 89C3 MOV  BX,AX
2803: 014C B416 MOV  AH,16
2803: 014E CD10  INT  10
2803: 0150 E2DE LOOP 0130
2803: 0152 B43C MOV  AH,3C
2803: 0154 BAB001 MOV  DX,01B0
2803: 0157 B90000 MOV  CX,0000
2803: 015A CD21  INT  21
2803: 015C 5B  POP  BX
2803: 015D 59  POP  CX
2803: 015E 81E9C001 SUB  CX,01C0
2803: 0162 89C3 MOV  BX,AX
2803: 0164 B440 MOV  AH,40
2803: 0166 BAC001 MOV  DX,01C0
2803: 0169 CD21  INT  21
2803: 016B B43E MOV  AH,3E
2803: 016D CD21  INT  21
2803: 016F EB08 JMP  0179
2803: 0171 0000 ADD  [BX+SI],AI
2803: 0173 0000 ADD  [BX+SI],AL
2803: 0175 0000 ADD  [BX+SI],AL
2803: 0177 0000 ADD  [BX+SI],AL
2803: 0179 17  POP  SS
2803: 017A 07  POP  ES
2803: 017B 1F  POP  DS

```



# 计算机图象仿真技术及应用

重庆大学

杜平安

中国工程物理研究院 丁厚本

## 一、前言

计算机图象仿真(image simulation)即在计算机屏幕上生成具有高度真实感的三维图象。六十年代后期,出现了以高速随机存储器 and 高速处理器为硬件基础的光栅式扫描图形显示器。由于它能控制屏幕每个象素的色彩和亮度,为计算机能够生成具有连续色调的浓淡图象奠定了硬件基础。七十年代初,Bui-Tvong, Phong 首先提出了计算机图象的光照模型,建立了图象仿真的算法基础。随着显示器空间及灰度分辨率的提高,以及各种仿真算法的出现和改进,现在计算机已能生成具有高光、阴影、反射、透明等效果的高度真实感图象。这种技术已用于建筑设计、机电产品外观造型设计、家俱和服装设计等多个领域,在国民经济中发挥着一定的作用。

## 二、光栅式扫描图形显示器

光栅显示器主要由图 1 所示的几个部分组成。

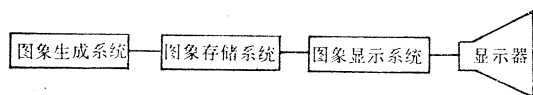


图 1

图象生成系统的作用是把要生成的图素经扫描变换存入刷新缓存中,图象显示系统每次从刷新缓存中

读取一行数据驱动光栅扫描,刷新缓存则存储要显示的图象。在图象仿真技术中,影响图象质量的主要是刷新缓存的容量。

刷新缓存也叫做帧缓存、位图。它是一大块连续的计算机存储区域,通常一个存储单元对应于屏幕上的一个象素,如图 2 所示。

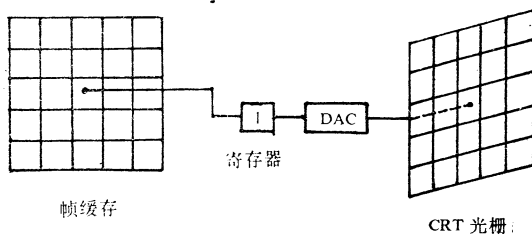


图 2

这些存储单元组成一个位平面。对于象图 2 所示的由一个位平面组成的帧缓存,由于屏幕上每一个象素只对应一个二进制位(只有 0、1 两种状态),因此只有黑白两种显示。

影响图象真实度主要是空间分辨率和灰度分辨率,它们由帧缓存容量决定。空间分辨率影响图象细节的表示,一般要求 900×900 以上,通常采用 1024×1024。空间分辨率低了,图象轮边会有明显的锯齿。灰度分辨率影响图象浓淡的连续性,一般要求 64 级以

```

2803: 017C 5F POP DI
2803: 017D 5E POP SI
2803: 017E 5D POP BP
2803: 017F 5C POP SP
2803: 0180 5A POP DX
2803: 0181 59 POP CX
2803: 0182 5B POP BX
2803: 0183 58 POP AX
2803: 0184 CB RETF
-d180
2803: 0180 5A 59 5B 58 CB 00 00 CA-E4 C8 EB BA
BA D7 D6 3A ZYIX.....:
2803: 0190 20 24 00 00 00 00 00 00 1E-06 C8 FD C8
  
```

```

FD C8 FD 0D $.
2803: 01A0 00 00 00 00 00 00 00 00-00 00 00 00
00 00 00 00 .....
2803: 01B0 00 00 00 00 00 00 00 61-62 63 64 65
00 00 00 00 .....abcde...
2803: 01C0 00 00 00 00 00 00 00 1A-88 DE 89 C3
B4 16 CD 10 .....
2803: 01D0 E2 DE B4 3C BA B0 01 B9-00 00 CD 21
5B 59 81 E9 ...<.....! [Y..
2803: 01E0 C0 01 89 C3 B4 40 BA C0-01 CD 21 B4
3E CD 21 EB .....@....!.>..!
2803: 01F0 08 00 00 00 00 00 00 00-00 17 07 1F
5F 5E 5D 5C .....^ \
  
```

上,256 级效果更好。灰度分辨率低了,图象表面会有明显的亮度差别,显得不光滑。要提高灰度分辨率,必须增加位平面数。如对于八位面的帧缓存,灰度分辨率为 256( $2^8$ )级,如图 3 所示。

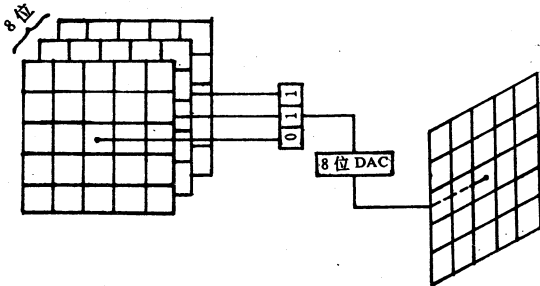


图 3

对于彩色光栅显示器,由于有红、绿、兰三个电子枪,因此每个电子枪都需要一组位平面驱动。设显示器空间分辨率为  $M \times N$ ,红绿兰三基色的灰度分辨率(色调)为  $L$ ,则帧缓存容量为:

$$V = 3 \times M \times N \times \log_2 L$$

如对于具有 24 位面,1024×1024 的显示器,内存空量为 3MB。

帧缓存容量越大,对计算机数据存取、处理和速度要求也越高,价格也会大大增加。目前人们采用了中间色调技术和反走样技术,从软件上来弥补硬件空间及灰度分辨率的不足。

### 三、图象仿真中的有关技术问题

自然界中的各种客观物体,是因其形、光、色、质的表示形式通过人们视觉反映在大脑中的。要使计算机生成具有真实感的仿真图象,就必须遵循、模拟物体表现形式的各种客观规律。

#### 1. 图象形状的绘制

在自然界中观察客观事物时,离视点远的部分显得较小,近的部分则看起来大些,这就是所谓“透视现象”。显示器二维屏幕要真实地反映出深度效果,就应该采用透视图。各类轴测图反映深度信息的效果都不及透视图。

透视图的绘制通过透视变换得到,在计算机内实现并不难,一般几何造型系统都有这种功能。

#### 2. 光照模型

人们之所以能够看见客观物体,是因为物体表面反射和各种光线进入人的眼内。光照模型即是模拟客观物体表面反射光线的数学表达式。图象的真实度在很大程度上取决于光照模型与客观规律的符合程度。

自然界中的光源有环境光和点光源(面光源可看

作由有限点光源组成)。

对于环境光,物体表面只有漫反射。漫反射光强  $I_d = K_d I_e$  ( $K_d$ —漫反射系数,  $I_e$ —环境光光强)

对于点光源,物体表面则有漫反射和镜面反射两种反射。漫反射即是向各个方面均匀的反射,镜面反射则是集中在理想反射方向附近的反射(如图 4 所示)。前者表现为从各个方向观察时表面都具有相同的亮度,后者则表现为表面某一部位具有“高光”。

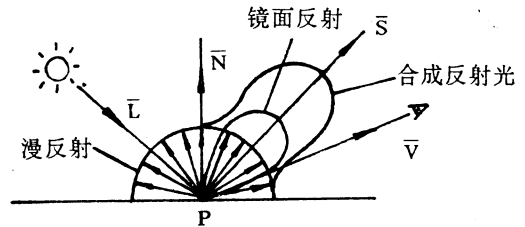


图 4

$$\text{漫反射光} \quad I_d = I_e K_d \cos \theta / (d + k)$$

$$0 \leq \theta \leq \frac{\pi}{2}$$

$$\text{镜面反射光} \quad I_{es} = I_e K_s \cos^n \delta / (d + k)$$

$$0 \leq \delta \leq \frac{\pi}{2}$$

式中:  $I_e$ ——点光源强度

$K_s, K_d$ ——镜面及漫反射系数

$\theta, \delta$ —— $\vec{L}$  与  $\vec{N}$ 、 $\vec{S}$  与  $\vec{V}$  的夹角

$d$ ——P 点到视点的距离

$n, k$ ——常数。

因此,表面某点 P 总的反光量为

$$I_p = \begin{cases} \frac{I_e K_d + I_e (\cos \theta \cdot k_d + \cos^n \delta \cdot k_s)}{(d + k)} & 0 \leq \theta \leq \frac{\pi}{2} \\ I_e K_d & \theta > \frac{\pi}{2} \end{cases} \quad \dots (1)$$

这里,控制表面发光情况的有以下几个因素:

(1)光源强度  $I_e, I_e$ 。如附图 2 所示的台灯,由于环境光值  $I_e$  取得较小,因此台灯座显得比较暗,好象是在夜晚的时候。

(2) $n$  值。 $n$  值大小可以模拟表面光洁及质地情况。如附图 1 所示,左边红球取的  $n$  值较大,因此高光集中而明亮,表面显得光滑和坚硬。中间红球取的  $n$  值较小,因此高光分散而灰暗,表面显得粗糙和松软。

(3) $d$  值。 $d$  值大小可以模拟光线在传播过程中能量衰减的现象。即离光源越远的物体越暗。附图 1 光

源的位置设在屏幕右前方,因此里面的地板显得较暗,它反映了地板的深度效果。

### 3. 仿真算法

目前的仿真算法较多,常用的有扫描线算法和光线跟踪算法。扫描线算法以多边形为计算对象,并结合 Gourand 和 Phong 两种明暗处理方法,能够生成比较真实的图象。附图 2 的台灯就是用这种方法生成的。它的特点是计算量与多边形数量有关,一般情况下生成速度较快。光线跟踪算法以屏幕像素为计算对象,它模拟自然界物体之间相互映照的规律,能够生成具有阴影、反射、透明等效果的高度真实感图象,是目前效果很好的一种较为成熟的算法。附图 1 就是利用这种方法生成的具有阴影、反射和透明效果的图象。左边红球由于光洁度较高,因此能够较清晰地反射地板图案,地板上有球的阴影,通过透明的黄色玻璃可以看到后面的绿球。这里着重介绍光线跟踪算法。

光线跟踪算法的基本思想是,从观察点出发,反向跟踪到达计算点  $P_1$  的反射光线  $R_1$  和透射光线  $T_1$ ,若  $R_1, T_1$  遇到景物中某物体点  $P_2, P_3$ ,再对到达  $P_2, P_3$  的反射和透射光线进行跟踪。依次下去,直到光线射出景外或光强值小于某一设定值。这个跟踪过程可用图 5 所示的二叉树表示。树结点表示光线与物体表面的交点,两枝分别代表反射和透射光线。

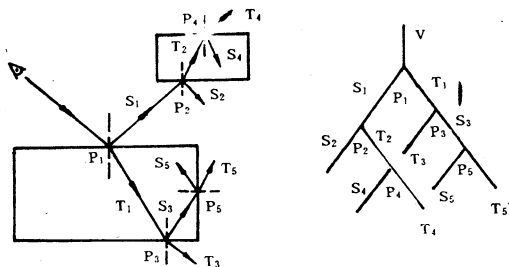


图 5

计算表面某点的光强,需按后序遍历相应的光线跟踪树。这时光照模型应在(1)式基础上加上反射光和透射光两项,即:

$$I_p = I_a K_a + I_i (\cos \theta - K_a + \cos^2 \delta \cdot K_r) / (d + k) + K_r I_r + K_t I_t \quad (2)$$

这种方法的缺点是计算量较大,计算时间与下面一些因素有关,图象复杂程度(即所占像素多少);有无阴影、反射,一般有比没有要多一倍。光源、个数,在有阴影的情况下,  $N$  个光源时的计算时间是没有阴影的  $N+1$  倍;反射次数;是否采用反走样等等。作者在 SUN4 工作站上生成附图 1 的图象花了近半个小时,而

附图 2 还不到 1 分钟。

### 4. 表面纹理

表面纹理即指表面细节。一种是指图象表面的花纹,一种是指表面的光洁状况。在计算机上模拟表面纹理时,前者可采用映象函数,后者则采用摄动函数。

### 四、图象仿真的应用

#### 1. 家俱、服装、鞋子等生活用品的设计

随着人民生活水平的提高,人们对家俱服装等这些必需品不仅仅是过去那种经久耐用的要求,而且有着越来越高的式样、色彩等外在美的追求。市场上这些用品流行的款式、色调更新十分迅速,为适应潮流的变化,生产设计单位总是在设计开发新产品投放市场。利用图象仿真技术,设计者可用计算机快速地设计出很多种样式和色彩的产品方案,利用硬拷贝或摄影得到有真实感的产品照片。这样,刚完成的设计无须制造实物就可进行广告宣传,生产厂家再根据用户订货要求制造实物,能够避免因市场预测不当而造成产品积压,节省大量制造及材料费用。

#### 2. 建筑设计

建筑物色彩、外型在建筑设计中是很重要的。设计者常常设计许多方案,供有关人员选择,之后再制作实体模型,以观察实际效果。这种从方案设计到模型制造,一要大量时间,二受时间、费用限制又不可能设计太多的方案。利用图象仿真技术,设计周期会大大缩短,设计方案能尽量多,并可模拟各种光源,从各个方向比较真实地考察建筑物,避免了模型制作。这在投标设计中是很重要的。

一个建筑与周围环境的协调性也是很重要的,设计者可将周围环境拍摄下来并硬拷贝到计算机内,再将设计的建筑物移动到待建的地址上,就能对设计效果进行较为如实的评价。

设计者还可以“进入”室建筑物内,观察室内的采光效果,进行房间布置等等。

#### 3. 机电产品外观造型设计

过去的机电产品比较强调内在质量,而忽视外观造型,设计的产品常有“傻、大、黑、粗”之感。随着商品经济的搞活,国内外市场产品竞争日益增强,人们逐渐重视了产品的造型设计,特别是象冰箱、彩电这类家用电器。图象仿真技术的应用,使计算机快速计算、绘图能力和人的美学创造思维相结合,使设计周期大大缩短,成本显著下降,对设计者也无绘画技能的要求。

#### 4. 广告装潢设计。

5. 自然景物的模拟。如山、水、树木、云雾以及人体面部、动物等的模拟。

## 一个复原空文件的 shell 程序

山西长治市人民银行 彭起顺

在 XENIX 系统中,有许多空文件。所谓空文件,就是仅有其文件名而无内容,即长度为零的文件,这些文件,在 XENIX 系统刚安装时,是名符其实的空文件,但是,随着以后系统的运行,这些文件的内容将由零逐渐增加,有的甚至是增加很快的。如系统控制台信息数据文件 /usr/adm/messages,系统每启动一次,它均要以追加方式在其内记录某些有关的信息数据,对于某种配置的 GW386 来说,该文件将以每次 1K 多字节的长度增加,若机器启动 100 次,它将增加 100 多 K 字节,占去许多不必要的磁盘空间,这就需要定期地使这些空文件复原,删掉其内容,释放其占用的磁盘空间,使其真正成为空文件。

虽然我们可以编制某些程序,针对不同类型的空文件逐个地处理,但是这样做效率太低,又不能做到对所有的空文件进行处理(因为用户不一定能知道在本系统内究竟有多少个空文件),为此,我们可以设计一个带有普遍性的 shell 程序,只要运行一次该程序,就可以把与空文件名不符实(即空文件长度不为零)的空文件全部清空复原,该程序名为 empfi,其内容如下:

```
1 PATH=/bin:/usr/bin:/etc
2 FILE1=tmp/file1. ¥¥
3 FILE2=/tmp/file2. ¥¥
```

```
4 FILE=
5 emptyfile(){
6     while read FILE
7     do
8         rm ¥FILE
9     done
10 }
11 cd/
12 fixperm -s /etc/perms/* 2> ¥FILE1
13 grep "not an empty file" ¥FILE : |awk '{print
    ¥6}' >¥FILE2
14 emptyfile < ¥FILE2
15 Fixperm -c /etc/perms/*
16 rm -f ¥FILE! ¥FILE2
```

在解释该程序之前,我们有必要先来了解一下 XENIX 系统中的一些说明文件,在 /etc/perms 目录下,有这样一些说明文件,其文件名为:dsmd,ext, rts 等,在这些说明文件中,包括了全部 XENIX 系统文件的概要说明,并按包为类型列在一起,其中一行做为一个文

## 请订阅《计算技术与自动化》季刊

自动化离不开计算技术      计算技术为自动化添翼  
计算技术与自动化联姻      理论研究同应用相结合

该刊特点:寓新颖性、实用性、学术性、通用性于一体,荟当代计算技术与自动化技术之精萃。

读者对象:从事计算技术与自动化专业的广大科技人员、管理人员、情报人员、院校师生。

订阅办法:请直接同该刊编辑部联系。

编辑部地址:长沙岳麓山湖南省计算技术研究所

全年每份订价:10 元(含邮费)。

邮政编码:410012

件的说明,一个文件说明包括6项,依次分别为:包标识符,权限说明,属主和属说明,链接数,文件名,卷号。如在说明文件/etc/perms/rts中有如下一行:

```
RTS e644 bin/bin 1 .usr/adm/messages B01
```

这一行的意思是讲,文件/usr/adm/messages的包为RTS,权限为644,属主和属组都是bin,链接数为1,卷号为B01。这里,需要注意的是在权限说明处有一个前缀字符“e”(即e644处),它是一个文件类型,这里的字符“e”表示/usr/adm/messages文件为空文件(还有其它字符,如X:可执行文件;d:目录文件等),上面给出的shell程序emptyfi所关心的就是此字符“e”。在目录/etc/perms下的各个说明文件中,凡是在该处有字符“e”的文件,均是XENIX系统中的空文件。因此,我们只要搜索所有说明文件(在/etc/perms下)中的该种“e”字符的文件说明,就可以找出XENIX系统中的全部空文件。值得庆幸的是,在XENIX系统中,有一个命令fixperm,它的如下格式:

```
fixperm -s /etc/perms/ *
```

将会以出错形式在屏幕上列出长度不为零的空文件名(当然还会列出其它文件名)。虽然我们可以用fixperm命令的其它格式来达到上述目的,但我们选中的这一格式将参照/etc/perms下的说明文件内容对磁盘上的有关文件进行修正,并同时找出有“e”字符说明的说明项在磁盘上找出与此说明相符的文件,检查其长度是否为零?是,则不予理睬;否,则以出错信息输出到标准错误输出上。

现在,我们来看程序emptyfi(程序中的行号是为了说明方便而加上的)。

程序中,第1行为路径说明,第2-4行分别定义了三个参数,其中前两个参数被定义为两个暂时存放有关结果的临时文件,若当前emptyfi运行的进程号为106,

则其后缀为“.106”。这样做的目的是为了尽量使这两个临时文件不要与其它临时文件同名而产生复盖现象。第5-10行则定义了一个删除文件函数。第11行是使当前目录为根,这是为了满足fixperm命令和emptyfile函数的要求。第12行是我们已讨论过的fixperm命令的格式,但是,这里,我们将其出错输出重定向到了文件%FILE1(即:/tmp/file1.106)中,若假定当前系统中仅存在两个长度不为零的空文件:/usr/adm/messages和/etc/wtmp,则此时/tmp/file1.106的内容为:

```
fixperm: not an empty file:./usr/adm/messages
fixperm: not an empty file:./etc/wtmp
fixperm: bad format executable file:./usr/bin/cref
:
:
```

第13行是从文件/tmp/file1.106中把空文件筛选出来而放在%file2(即/tmp/file2.106)中,则此时/tmp/file2.106的内容为:

```
./usr/adm/messages
./etc/wtmp
```

第14行,就利用自定义的删除文件函数emptyfile从文件/tmp/file2.106中逐一读取文件名而删除之。第15行,是利用fixperm的选择项“-C”去建立空文件(因其在此前已被emptyfile删除掉了)。第16行是删除掉建立的在临时目录/tmp中的两个临时文件。

该程序对于释放一些不必要的被占用的磁盘空间非常有用。当然,我们还可以对其进行完善。如规定空文件长度超过给定值时再将其清除,或者把长度不为零的空文件显示出来以提醒用户这些文件该清空了,清空与否让用户去决定等等。笔者在这里所列出的程序是仅起一抛砖引玉的作用,质量更高的程序,相信读者是会编制出来的。

## FOXBASE 中下拉式菜单的简单实现

江西财经学院信息系 彭光前

“下拉式”菜单由于具有简单明了、新颖直观的特点,受到软件开发者的普遍欢迎。其实现方法在不少报刊中都有介绍,且大多采用INKEY()函数取得光标键值来编程。其实,对拥有FOXBASE的用户,可用FOXBASE中的四条在一些书籍中未介绍的命令,可以方便地编写“下拉式”菜单功能选择程序。这四条命令

是:

1. @(行,列) PROMPT <exp C>

使命令在屏幕指定的位置上显示exp C的值,即菜单项目。

2. SET MESSAGE TO [char-str]

当SET STATUS ON/off命令为ON时,SET MES-



SAGE TO <char-str>命令将用户在该命令中定义的字符串(一般操作提示)在屏幕的最后一行显示。字符串必须用方括号、或单引号、或双引号括起来。字符串的最大长度为79个字符。

### 3. MENU TO <mem var>

该命令在当前定义的菜单上产生光条。光条位置可由光标键(↑、↓、←、→、PgDn、PgUp)控制。

### 4. @ <行,列> BOX [(exp C)]

该命令在屏幕指定位置上为各级下拉菜单开口。

设有一个档案管理软件,其主要功能如下面程序所示,其“下拉式”菜单程序的实现见后面的程序。

这里以较短的程序段实现了各程序功能模块之间的灵活调用,其过程一目了然,体现了结构化程序设计思想,这里要注意的是,尽管在各菜单前面未给出编号,但它们实际隐含了顺序号,故在CASE或IF语句中仍按1至N(菜单项数)进行条件判断。当光条落在所需的菜单位置上,只要敲入回车键则该菜单被选中。在层层退出的菜单设计中,每一级菜单都需要设有一个退出项(对应编号为N),否则需按ESC键退出。

#### \* 下拉菜单程序示例

```
SET TALK OFF
```

```
SET COLO TO +R/B, W/G+
```

```
SET MESSAGE TO' ↑ 向上 ↓ 向下 ← 向左 → 向右  
PgUp 首菜单 PgDn 尾菜单 ↙ 选定'
```

```
CLEA
```

```
DO WHILE .T.
```

```
SET COLO TO +W/R
```

```
@1,20 SAY '★★★★档案管理系统★★★★'
```

```
@3,7 PROM '数据录入'
```

```
@3,19 PROM '查 询'
```

```
@3,31 PROM '组 卷'
```

```
@3,43 PROM '报表打印'
```

```
@3,67 PROM '退出系统'
```

```
@5,4 SAY REPL ('※',36)
```

```
MENU TO SR
```

```
SET COLO TO +W/
```

```
@6,0 CLEA
```

```
DO CASE
```

```
CASE SR=5
```

```
CLEA
```

```
CLOS DATA
```

```
@8,35 SAY '再 见!'
```

```
SET COLO TO W/
```

```
RETU
```

```
CASE SR=1
```

```
DO WHILE .T.
```

```
SET COLO TO +R/B
```

```
@7,20,14,36 BOX '##'
```

```
@9,23 PROM' 收文数据录入'
```

```
@11,23 PROM' 发文数据录入'
```

```
@13,23 PROM' 退 出'
```

```
MENU TO SR1
```

```
DO CASE
```

```
CASE SR1=1
```

```
DO WJ11
```

```
CASE SR1=2
```

```
DO WJ12
```

```
CAWE SR1=3
```

```
EXIT
```

```
ENDC
```

```
ENDD
```

```
CASE SR=2
```

```
DO WHILE .T.
```

```
SET COLO TO +W/B
```

```
@7,20,14,36 BOX '##'
```

```
@9,23 PROM' 档案内容查询'
```

```
@11,23 PROM' 利用率 查询'
```

```
@13,23 PROM' 退 出'
```

```
MENU TO SR2
```

```
IF SR2=3
```

```
EXIT
```

```
ENDI
```

```
SS=STR(SR2,1)
```

```
DO WJ2&SS
```

```
ENDD
```

```
CASE SR=3
```

```
DO WHILE .T.
```

```
SET COLO TO +R/B
```

```
@7,20,14,36 BOX '##'
```

```
@9,23 PROM' 收文组卷'
```

```
@11,23 PROM' 发文组卷'
```

```
@13,23 PROM' 退 出'
```

```
MENU TO SR3
```

```
IF SR3=3
```

## 利用 IBM5550 实现表格的自动生成

中计公司吉林分公司 孙利国 钟立红

**提要：**利用 BASIC 编制了表格自动生成软件，根据输入参数，可生成需要的表格格式及规格，并可随时修改表内数据，无需任何命令，具有操作简单、使用灵活等特点，可替代一些电子报表的部分功能。

### 一、引言

IBM5550 是 IBM 公司推出的多功能个人计算机，具有较强的中文处理能力。其配置的 BAsic 解释程序是由 Microsoft 公司的 MBAsic 改进而成，某些语句具有一些特殊功能。其中的 Color 语句具有如下格式：Color n, m, l

其中 n、m、l 均为整数，具体含义说明如下：n：取值 0~31；表示成二进制形式： $a_4a_3a_2a_1a_0$

各位含义是： $a_0$ ：1 代表各行要加下线。

0 代表不加下线。

$a_1$ ：1 为高亮度显示。

0 为一般亮度显示。

$a_2$ ：1 代表闪烁。

0 代表不闪烁。

其余二位无意义。

m 取值 0~1；0 为正常显示；1 为反像显示。

l 取值 0~4；0——般显示

- 1—各行左侧加竖线；
- 2—各行上方加横线；
- 3—各行上方及左侧加横线；
- 4—保持原有直线及横线。

我们利用上述语句的特点，编制了小型制表软件包。本软件可根据不同需要生成不同形式及不同规格的表格。使用时仅需按照显示器上的提示，输入与表格形式等有关的参数，便可自动生成各科统计报表。操作时无需命令，仅需按下相应的功能键，并可随时对表内数据及标题栏更改，最后结果可存盘。需要时只要将表格数输入，便可重现最后一次表格及内容，使用灵活，方便。

### 二、软件构成

本软件由 BASIC 制表主程序 Bill、BAS，批处理文件 MKBILL.BAT，及数据文件 Bill<sub>0~2</sub>.DAT, Val<sub>1~3</sub>.DAT 等三部分组成。在批处理文件中放置两条语句指明 Bill、BAS 目录及执行语句，则在 DOS 下键入 MKBILL 文件名，便进入制表软件，带有扩展名 .DAT 的数据文件用来存放输入参数及软件自身例 3 中的有关数据。制表主程序 Bill.BAS 由若干模块组成，模块中又含有若干子模块，大体结构如图(1)。

EXIT

ENDI

SS=STR(SR3,1)

DO WJ3&SS

ENDD

CASE SR=4

DO WHIL.T.

SET COLO TO +w/B

@7,20,14,36 BOX'#'

@8,23 PROM' 卷内文件目录'

@10,23 PROM' 案卷 目录'

@12,23 PROM' 全引目录'

@13,23 PROM' 退出'

MENU TO SR4

IF SR4=4

EXIT

ENDI

SR44=STR(SR4,1)

DO WJ4&SR44

ENDD

ENDC

ENDD

RETU

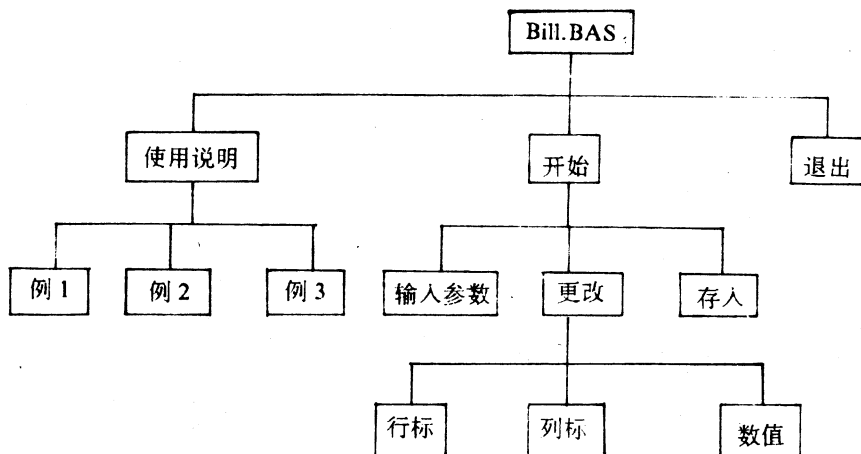


图 1、制表主程序结构

下面结合执行过程,说明各模块的作用。

开机后,在 DOS 状态下,键入文件名 MKBill. 则显示表头菜单,表头下提示行为:

PF<sub>1</sub>:开始 PF<sub>2</sub>:使用说明 ESC:退出

(1)按 ESC 键,结束软件,显示“谢谢,欢迎再次使用”,返回 DOS。

(2)按 F<sub>2</sub> 键,进入使用说明模块,显示简要说明内容及软件中的有关规定,下方显示:

“你想看一下例子吗? 想按 Y, 否则按 N。”

键入 N(不必回车),返回表头菜单;键入 Y 后,则屏幕上方显示出本软件所能生成的表格的三种基本形式:

①第一种采用一般的行宽,其中第一列(称行标题栏)宽度由用户输入,其余列具有相等的宽度,只需根据需要输入总列数。

②第二种采用宽的列标题栏格式,以适应特殊需要,且每列宽度,总列数,总行数均可由键盘输入确定。

③第三种全部采用宽行,生成宽表格,且每列宽度,总列数及总行数均可改变。

屏幕下端显示“请输入你所需表格格式(1-3)”,你只需按其中的某个数字键,便会给出相对应报表的例子,且提示行显示:

改初列标题按 C,改行标题按 L,改数值按 V,返回按 ESC。

你可根据需要,进行有关操作,这部分将结合 PF<sub>1</sub>, 功能介绍。若按 ESC 则回到表头菜单。

<3>在表头菜单下,按下 PF<sub>1</sub> 键,则进入开始制表模块,此时屏幕将提示你输入表格标题名称,表格类型

(1-3)及表格列宽,总列数,总行数等有关参数,必要时可加小标题。若输入参数不合要求,则鸣叫,提示重新输入。为保证表格设置合理,软件对宽度剩余量进行计算,每次输入一个参数后均有显示。

以上参数输入无误后,将生成设想格式的空白表格,并出现提示行:

PF<sub>3</sub>:输入数据 PF<sub>4</sub>:更改 PF<sub>5</sub>:存入 ESC:返回  
选择某一键,便进入相应的子模块处理程序。

①按 PF<sub>3</sub> 后,根据生成的空表格规格,屏幕提示输入列标题,行标题,按列输入数据,输入完毕,自动填入表格,同时再现提示行“PF<sub>3</sub>,……”若输入参数不符合要求,则鸣叫,提示重新输入。

②若想改动表格参数,按 PF<sub>4</sub> 键,则出现提示行:  
改动列标题 C 改动行标题 L,改动数值 V,返回 ESC。

根据需要按相应键,根据提示输入,需改动值所在行号、列号,则提示行下端显示“××”改为? 其中××为原来值,输入改动后的值 YY,回车,则 YY 填入相应位置。

你也可以利用 PF<sub>5</sub> 键输入列、行标题及数值。按 ESC 则回到上一屏。

③若按 PF<sub>5</sub> 则完成表格及内容的存盘。提示行下端显示:“正在存入……”片刻之后显示:“数据已经存入”,些时将你所输入的表格参数及内容存入 .DAT 文件中。下次使用时,只需输入与本次相同的表格参数(总行数:列宽、总列数),便可重现存储的报表。若输入参数不符,则生成新表。

④按 ESC 回到表头菜单下。

本软件的特点是每执行一步,屏幕均有提示,用户可根据提示进行操作,无需任何命令,程序较简单,且变化灵活,使用十分方便。克服了一般报表表格生成麻烦或改变困难及命令繁琐等问题,对软件做进一步改进,可实现更多功能,从而部分代替一些电子报表的功能。

### 三、程序清单(略去了部分结束装饰语句):

```
10 REM IBM 5550 多功能造表程序
20 REM Version 1.00 () Copyright CS$S-JL Corp 1991
30 DIM NMCOL$(50), NML$(16), LM$(16,50), WC(50)
50 DIRECT=0
70 SCREEN 0,0:KEY OFF:COLOR 0,0,0:CLS:LOCATE 5,33,0:PRINT "IBM 5550"
80 LOCATE 7,33:PRINT "个人计算机"
90 LOCATE 9,23:PRINT CHR$(334)+STRING$(17,334)+CHR$(334)
100 LOCATE 10,23:PRINT CHR$(334)+STRING$(34,32)+CHR$(334)
110 LOCATE 11,23:PRINT CHR$(334)+"多功能造表程序"+CHR$(334)
112 LOCATE 12,23:PRINT CHR$(334)+" "+CHR$(334)
120 LOCATE 13,23:PRINT CHR$(334)+"设计者:孙利国、钟立红"+CHR$(334)
122 LOCATE 14,23:PRINT CHR$(334)+STRING$(34,32)+CHR$(334)
130 LOCATE 15,23:PRINT CHR$(334)+STRING$(17,334)+CHR$(334)
140 LOCATE 17,26:PRINT "(C) CS&S-JL 1991.5"
142 LOCATE 19,32:PRINT " * 欢迎使用本表格 * "
150 COLOR ,1:LOCATE 22,25:PRINT "PF1:开始 PF2:使用说明 ESC:退出":COLOR ,0,0
180 K$=INKEY$:DEF SEG:ON KEY(1)GOSUB 5000
190 ON KEY(2)GOSUB 1000
200 IF K$=CHR$(27) THEN GOTO 300
210 KEY(1)ON:KEY(2)ON:KEY(3)ON:GOTO 180
300 SYSTEM
1000 '使用说明'
1010 KEY(1)OFF:KEY(3)OFF:SCREEN 0,0:COLOR ,1,0:LOCATE 4,35:PRINT"使用说明"
1020 COLOR ,0:LOCATE 7,14:PRINT"本造表程序可根据需要生成不同格式,不同规格的表格"。
1030 LOCATE 8,14:PRINT "使用时只需按要求输入有关参数,然后即可输入数据,并可随时更换,使用十分方便"。
1040 LOCATE 9,14:PRINT
1042 LOCATE 10,14:PRINT "本造表程序中从左到右定义为列(0列开始),从上到下定义为行(1行开始)。"
1043 LOCATE 11,14:PRINT
1044 LOCATE 12,14:PRINT "本程序生成的表格有三种基本格式(1-3)。"
1050 LOCATE 20,22:COLOR,1:PRINT "你想看一下例子吗? 请按 Y,否则按 N":COLOR ,0,0
1060 KC$=INKEY$
1070 IF KC$="y" GOTO 4000
1072 IF KC$="Y" GOTO 4000
1080 IF KC$="N" GOTO 50
1082 IF KC$="n" GOTO 50
1090 IF KC$=" " GOTO 1060
1100 GOTO 1060
```

```

1110 RETURN
2000 '制作表程序'
2010 COLOR 0,0,0 : SCREEN 0,0 : CLS : COLOR 9,0,0 : TL=(76-KLEN(TITLE$))/2
2020 LOCQTE 1,TL : PRINT TITLE$ ;
2030 FOR I=1 TO L : LOCATE I+3,1 : COLOR 0,0,3 : PRINT " " ; COLOR ,,2 : PRINT SPC(WC(0)) ;
      FOR J=1 TO M-1 : COLOR ,,1 : PRINT " " : NEXT I
2040 COLOR ,,2 : LOCATE ,1 : PRINT SPC(INT((78-WC(0)-M+1)/(M-1) * (M-1)+WC(0)+M) :
      COLOR ,,0
2050 RETURN
2100 ' 表 2
2110 SCREEN 0,0 : COLOR 9,0,0 : TL=KLEN(TITLE$) : LOCATE 1,(76-TL)/2 : PRINT TITLE$
2120 IF STITLE=1 GOTO 2140
2130 LOCATE 3,INT((76=KLEN(STITLE$))/2) : PRINT STITLE$
2132 SUM=0
2134 LOCATE 4,1 : FOR I=0 TO M-1 : COLOR 0,,3 : PRINT " " ; COLOR ,,2 : PRINT SPC(WC(I)) ;
      NEXT I : COLOR ,,1 : PRINT " "
3136 LOCATE 5,1 : FOR I=0 TO M-1 : COLOR ,,1 : PRINT " " ; COLOR ,,0 : PRINT SPC(WC(I)) ;
      NEXT I : COLOR ,,1 : PRINT " "
2140 FOR J=2 TO L : LOCATE 4+J,1 : FOR I=0 TO M-1 : COLOR 0,,3 : PRINT " " ; COLOR ,,2 :
      PRINT SPC(WC(I)) ; NEXT I : COLOR ,,1 : PRINT " " : NEXT J
2142 FOR K=0 TO M-1 : SUM=SUM+WC(K) : NEXT K
2150 COLOR ,,2 : LOCATE 5+L,1 : PRINT " " ; PRINT SPC(SUM+M-1) : COLOR ,,0
2160 RETURN
2200 ' 表 3'
2210 SCREEN 0,0 : COLOR 9,0,0 : TL=KLEN(TITLE$) : LOCATE 1,(76-TL)/2 : PRINT TITLE$
2220 IF STITLE=1 GOTO 2240
2230 LOCATE 3,INT((76-KLEN(STITLE$))/2) : PRINT STITLE$
2240 SUM=0 : FOR J=1 TO L
2250 LOCATE 3+J*2,1 : FOR I=0 TO M-1 : COLOR 0,,3 : PRINT " " ; COLOR ,,2 : PRINT SPC (WC
      (I)) ; NEXT I : COLOR ,,1 : PRINT " " : NEXT J
2280 FOR K=0 TO M-1 : SUM=SUM+WC(K) : NEXT K
2290 COLOR ,,2 : LOCATE 5+L*2,1 : PRINT " " ; PRINT SPC(SUM+M-1) : COLOR ,,0
2300 RETURN
3000 ' 例子程序,表 1'
3002 DIRECT=1
3010 TITLE$="出勤表"
3020 F=1 : WC(0)=10 : M=7 : L=16 : GOTO 6100 : GOTO 5090
3025 RETURN
3100 ' 表 2
3110 DIRECT=1 : F=2 : M=6 : L=16 : TITLE$="自然情况一览表"
3120 WC(0)=10 : WC(1)=16 : WC(2)=4 : WC(3)=8 : WC(4)=28 : WC(5)=7
3130 GOSUB 7500 : GOTO 5090
3140 RETURN
3200 ' 表 3

```



```

3210 DIRECT=1 : F=3 : M=6 : L=8 : TITLE$="名"
3220 WC(0)=10 : WC(1)=10 : WC(2)=14 : WC(3)=18 : WC(4)=7 : WC(5)=12
3230 GOSUB 8500 : GOTO 5090
3240 RETURN
4000 ' 例子处理程序
4010 SCREEN 0,0 : COLOR 9,,0 : LOCATE ,10 : PRINT "标题 1" : LOCATE 1,38 : PRINT "标题 2" : LOCATE 1,64 : PRINT "标题 3"
4020 M=8 : FOR J=0 TO 11 : COLOR 0,0,3 : LOCATE 4+J,1 : PRINT " " : COLOR ,,2 : PRINT SPC(M) : FOR I=0 TO 4 : COLOR ,,3 : PRINT " " : COLOR ,,2 : PRINT SPC(INT((25-M-5)/5)) : NEXT I : COLOR ,,1 : PRINT " " : NEXT J
4030 COLOR 0,0,2 : LOCATE 16,1 : PRINT SPC(INT((25-M-5)/5) * 5+M+6) : COLOR ,,0
4040 COLOR 9,0,0 : LOCATE 2,38 : PRINT "小标题"
4050 m=7 : color 0,0,3 : LOCATE 4,28 : PRINT " " : COLOR ,,2 : PRINT SPC(M) : COLOR ,,3 : PRINT " " : COLOR ,,2 : PRINT SPC(4) : COLOR ,,3 : PRINT " " : COLOR ,,2 : PRINT SPC(5) : COLOR ,,1 : PRINT " "
4060 COLOR ,,1 : LOCATE 5,28 : PRINT " " : COLOR ,,0 : PRINT SPC(M) : COLOR ,,1 : PRINT " " : COLOR ,,0 : PRINT SPC(5) : COLOR ,,1 : PRINT " " : COLOR ,,0 : PRINT SPC(4) : COLOR ,,1 : PRINT " " : COLOR ,,0 : PRINT SPC(5) : COLOR ,,1 : PRINT " "
4070 FOR J=1 TO 10 : COLOR ,,3 : LOCATE 5+J,28 : PRINT : PRINT " " : COLOR ,,2 : PRINT SPC(7) : COLOR ,,3 : PRINT " " : COLOR ,,2 : PRINT SPC(5) : COLOR ,,3 : PRINT " " : COLOR ,,2 : PRINT SPC(4) : COLOR ,,3 : PRINT " " : COLOR ,,2 : PRINT SPC(5) : COLOR ,,1 : PRINT " " : NEXT J
4080 COLOR ,,2 : LOCATE 16,28 : PRINT SPC(25) : COLOR ,,0
4082 FOR I1=0 TO 5
4090 LOCATE 4+I1*2,55 : COLOR ,,3 : PRINT " " : COLOR ,,2 : PRINT SPC(6) : FOR I2=0 TO 2 : COLOR ,,3 : PRINT " " : COLOR ,,2 : PRINT SPC(4) : NEXT I2 : COLOR ,,1 : PRINT " "
4100 COLOR ,,1 : LOCATE 2*I1+5,55 : PRINT " " : COLOR ,,0 : PRINT SPC(6) : FOR I3=0 TO 2 : COLOR ,,1 : PRINT " " : COLOR ,,0 : PRINT SPC(4) : NEXT I3 : COLOR ,,1 : PRINT " "
4102 NEXT I1
4110 COLOR ,,2 : LOCATE 16,55 : PRINT SPC(22) : COLOR ,,0
4120 LOCATE 17,4 : PRINT "<列宽相等,个数可调>" : PRINT SPC(6) + "<列宽可变,个数可调>" : PRINT SPC(6) + "<列宽可变,个数可调>"
4122 LOCATE 19,24 : PRINT "以上说明是指除左侧标题栏以外的列"
4130 LOCATE 21,18 : COLOR ,1 : PRINT "请选择表格格式,输入 1—3。" : PRINT "返回主菜单,按 ESC" : COLOR 0,0,0
4140 KA$=INKEY$
4150 IF KA$="1" GOTO 3000
4160 IF KA$="2" GOTO 3100
4170 IF KA$="3" GOTO 3200
4180 IF KA$=CHR$(27) GOTO 4210
4190 IF KA$="" GOTO 4140
4200 GOGT 4140
4210 RRTURN
5000 ' 使用

```

```

5002 DIRECT=0
5010 KEY(2) OFF : CLS : PRINT "请输入你选择的表格格式(1-3)" : INPUT F
5012 IF F<1 OR F>3 THEN BEEP : GOTO 5010
5020 PRINT "请输入表格标题：" : INPUT TITLE$ : PRINT "是否需要小标题? (Y/N)" : KS$ =INKEY$
5022 KS$ =INKEY$ : IF KS$ ="Y" GOTO 5032
5024 IF KS$ ="y" GOTO 5032
5026 IF KS$ ="n" GOTO 5040
5028 IF KS$ ="N" GOTO 5040
5030 GOTO 5022
5032 PRINT "请输入小标题：" : INPUT STITLE$
5040 ON F GOSUB 6200, 7000, 7200
5090 LOCATE 22,18 : COLOR 0,1 : PRINT "PF3 : 输入数据 PF4 : 更改 PF5 : 存入 ESC : 退出" : COL-
    OR ,0,0
5100 KEY OFF : K$ =INKEY$
5110 ON KEY(3) GOSUB 5200
5120 ON KEY(4) GOSUB 5400
5122 ON KEY(5) GOSUB 6000
5130 IF K$ =CHR$(27) THEN RETURN 50
5140 KEY(3) ON : KEY(4) ON : KEY(5) ON : GOTO 5090
5150 RETURN
5200 CLS : PRINT"请输入每列标题(从左到右顺序,共" : PRINT M : PRINT "列)："
5201 FOR I=0 TO M-1
5202 PRINT "第" : PRINT I : PRINT "列(输入长度不大于" : IF I=0 THEN PRINT WC(0) : PRINT")"
    ELSE PRINT INT((78-WC(0)-m+1)/(M-1))-1 : PRINT ")" : GOTO 5206
5204 INPUT NMCOL$(I) : IF KIEN(NMCOL$(I))>(WC(0)-1) THEN NEEP : GOTO 5202 : GOTO 5208
    5205 NEXT I : GOTO 5210
5206 INPUT NMCOL$(I) : IF KLEN(NMCOL$(I))>(INT(78-WC(0)-M+1)/(M-1))-1 THEN BEEP
    : GOTO 5202
5208 GOTO 5205
5210 PRINT "请输入行标题(从上到下顺序,宽度小于" : PRINT WC(0) : PRINT"共" : PRINT L-1 :
    PRINT "行)：" : FOR J=2 TO L : PRINT "第" : PRINT J : PRINT "行" : INPUT NML$(J) : NEXT J
5220 PRINT "请输入数据(按列输入,每列从上到下顺序)：" : FOR I=1 TO M-1 : FOR J=2 TO L : PRINT
    "第" : PRINT I : PRINT"列(输入长度不大于" : PRINT INT((78-WC(0)-M+1)/(M-1))-1 :
    PRINT ")" : INPUT LM$(J,1) : NEXT J : NEXT I
5230 ON F GOSUB 2000,2100,2200
5232 ON F GOTO 5240,7240,8200
5240 COLOR ,, 2 : LOCATE 4,3 : PPRINT NMCOL$(0) : FOR I=0 TO M-2 : LOCATE 4,WC(0)+3+I*
    (1+INT((78-WC(0)-M+1)/(M-1))) : PRINT NMCOL$(I+1) : NEXT I
5250 FOR I=2 TO L : LOCATE 3+I,3 : PRINT NML$(I) : NEXT I
5260 FOR I=0 TO M-2 : FOR J=2 TO L : LOCATE 3+J,WC(0)+3+I*(1+INT((78-WC(0)-M+1)/
    (M-1))) : PRINT LM$(J,I+1) : NEXT J : NEXT I
5280 RETURN
5300 LOCATE 22,1 : PRINT" " : RETURN
5400 ' 更改

```

```

5410 COLOR ,1: LOCATE 22,14: PRINT "改动列标题请按 C; 改动行标题请按 L; 改动数值按 V; 返回按
      ESC": COLOR ,0,0
5420 KM$=INKEY$: IF KM$=CHR$(27) GOTO 5300
5430 IF KM$="C" OR KM$="c" GOTO 5460
5440 IF KM$="I" OR KM$="i" GOTO 5490
5450 IF KM$="V" OR KM$="v" GOTO 5520
5452 GOTO 5410: RETURN
5460 LOCATE 23,1: PRINT "
5462 LOCATE 23,1: PRINT "请输入需改动列标题号(0-"; PRINT M-1; PRINT "):"; INPUT I; IF I>M
      THEN BEEP: GOTO 5462
5464 ON F GOTO 5470,7300,7600
5470 LOCATE 23,1: PRINT "
5472 LOCATE 23,1: PRINT " "; PRINT NMCOL$(I); T$(I)=NMCOL$(I); PRINT "改为"; INPUT NM-
      COL$(I); LOCATE 23,20: PRINT "("; PRINT "确定吗? Y/N"; PRINT ")"
5473 KS$=INKEY$: IF KS$="Y" GOTO 5478
5474 IF KS$="y" GOTO 5478
5475 IF KS$="n" THEN NMCOL$(I)=T$(I); GOTO 5460
5477 GOTO 5473
5478 IF I=0 GOTO 5482
5480 I I=WC(0)+3+(I-1)*(1+INT((78-WC(0)-M+1)/(M-1))); COLOR ,2: LOCATE 4,I I: PRINT "
      "; PRINT SPC(INT((78-WC(0)-M+1)/(M-1))-5); LOCATE 4,I I: PRINT NMCOL$(I); COLOR ,0,0
5481 LOCATE 23,1: PRINT " "; GOTO 5410
5482 COLOR ,2: LOCATE 4,3: PRINT " "; PRINT SPC(WC(0)-5); LOCATE 4,3: PRINT NMCOL$(I);
      COLOR ,0,0: GOTO 5481
5490 LOCATE 23,1: PRINT "请输入需改动的行标题号:(1-"; PRINT L-1; PRINT "):"; INPUT J; IF J
      <1 OR J>L THEN BEEP: GOTO 5490
5492 LOCATE 23,1: PRINT "
5500 LOCATE 23,1: PRINT " "; PRINT NML$(J); PRINT "改为"; INPUT NML$(J); LOCATE 23,20:
      PRINT "("; PRINT "确定吗? Y/N"; PRINT ")"
5502 KS$=INKEY$: IF KS$="N" GOTO 5410
5503 IF KS$="n" GOTO 5410
5504 IF KS$="Y" GOTO 5509
5505 IF KS$="y" GOTO 5509
5506 GOTO 5502
5509 ON F GOTO 5510,7340,7640
5510 COLOR ,2: LOCATE 3+J,2: PRINT " "; PRINT SPC(WC(0)-4); LOCATE 3+J,2: PRINT NML
      $(J); COLOR ,0,0
5512 LOCATE 23,1: PRINT " "; GOTO 5410
5520 LOCATE 23,1: PRINT "请输入需改动值所在行号(1-"; PRINT M-1; PRINT "):"; INPUT I; IF I>M
      THEN BEEP: GOTO 5420
5521 LOCATE 23,1: PRINT " "; GOTO 5410
5530 LOCATE 23,1: PRINT "请输入需改动值所在行号:(2-"; PRINT L; PRINT "):"; INPUT J; IF J<2
      OR J>L THEN BEEP: GOTO 5530
5532 LOCATE 23,1: PRINT " "; GOTO 5410

```

```

5539 LOCATE 23,1;PRINT"                ";GOTO 5410
5540 LOCATE 23,1;PRINT " ";PRINT LM$(J,I);T$(J,I)=LM$(J,I);PRINT "改为";INPUT LM
      $(J,I);LOCATE 23,20;PRINT "(";PRINT"确定吗? Y/N";PRINT")"
5541 KS$=INKEY$ : IF KS$="N" GOTO 5552
5542 IF KS$="Y" GOTO 5552
5543 IF KS$="n" THEN LM$(I)=T$(I);GOTO 5539
5546 IF KS$="Y" .HEN LM$(I)=T$(I);GOTO 5539
5547 GOTO 5541
5552 LOCATE 23,1;PRINT"                "
5554 ON F GOTO 5560,7340,7660
5560 I I=WC(0)+3+(I-1)*(1+INT((78-WC(0)-M+1)/(M-1)))
5561 COLOR ,,2;LOCQTE 3+J,I I:PRINT SPC(INT((78-WC(0)-M+1)/(M-1))-4:LOCATE 3+J,I I:
      PRINT LM$(J,I);COLOR ,0,0
5562 LOCATE 23,1;PRINT"                "
5580 RETURN
6000 '入库
6002 ON F GOTO 6010 ,7400,8400
6010 SWC(0)=WC(0):SM=M:SL=L:PRINT"正在存入....."
6012 IF DIRECT=1 GOTO 6022
6020 OPEN "BILL.DAT" FOR OUTPUT AS #1:GOTO 6030
6022 OPEN "vall.DAT" FOR OUTPUT AS #1
6030 PRINT #1,SWC(0),SM,SL
6040 FOR I=0 TO M:PRINT #1,NMCOL$(I):NEXT I
6050 FOR J=2 TO L:PRINT #1,NML$(J):NEXT J
6060 FOR I=1 TO M:FOR J=2 TO L:PRINT #1,LM$(J,I):NEXT J:NEXT I:CLOSE #1
6070 LOCATE 23,1:PRINT"数据已经存入"
6080 RETURN
6100 ON F GOTO 6101,7500,8500
6101 IF DIRECT=1 THEN GOTO 6104
6102 OPEN "BILL.DAT" FOR INPUT AS#1:GOTO 6110
6104 OPEN "vall.DAT" FOR INPUT AS#1
6110 INPUT #1,SWC(0),SM,SL
6120 FOR I=0 TO M:INPUT #1,NMCOL$(I):NEXT I
6130 FOR J=2 TO L:INPUT #1,NML$(J):NEXT J
6140 FOR I=1 TO M:FOR J=2 TO L:INPUT #1,LM$(J,I):NEXT J:NEXT I
6150 COLSE #1
6160 GOTO 5230
6170 RETURN
6200 PRINT"请输入表格第列宽度(4-40)":INPUT WC(0)
6210 IF WC(0)>40 OR WC(0)<4 THEN BEEP:GOTO 6200
6220 PRINT"其余列的数目(应小于";PRINT INT((78-WC(0))/3);PRINT")":INPUT M:M=M+1
6230 IF M>INT((78-WC(0))/3) THEN BEEP:GOTO 6220
6240 PRINT"总行数(从上到下为行)(2-16)":INPUT L
6245 IF L>16 OR L<2 THEN BEEP:GOTO 6240

```

```

6250 GOSUB 2000
6255 IF DIRECT=1 GOTO 5075
6260 OPEN "BILL. DAT" FOR INPUT AS#1 : GOTO 6270
6265 OPEN "vall. DAT" FOR INPUT AS#1 : GOTO 6100
6270 INPUT #1,SWC(0),SM,SL : CLOSE#1
6275 IF WC(0)=SWC(0) AND L=SL AND M=SM THEN GOTO 6100
6285 RETURN
7000 PRINT "请输入表格总列数(2-20,最载侧为第0列)" : INPUT M
7010 IF M>20 OR M<2 THEN BEEP : GOTO 7000
7012 ON F-1 GOTO 7020,7022
7020 PRINT "请输入表格总行数(2-16,最上侧为第1行)" : INPUT L : GOTO 7030
7022 PRINT "请输入表格总行数(2-8,最上侧为第1行)" : INPUT L
7024 IF L>8 OR L<2 THEN BEEP : GOTO 7012
7024 IF L>16 OR L<2 THEN BEEP : GOTO 7012
7040 SUM=0
7050 FOR I=0 TO M-1
7060 PRINT "请输入第"; : PRINT I; : PRINT "列宽度(每列宽度应大于4,总宽度"; : PRINT 78-M; :
      PRINT ", 还剩"; : PRINT 78-SUM-M; : PRINT ")" : INPUT WC(I) : IF WC(I)<4 OR WC(I)>(78-
      -SUM-M) THEN BEEP : GOTO 7060
7070 IF (78-SUM-M)<4 THEN BEEP : PRINT "宽度太大,或列数过多,重新输入" : SUM=0 : GOTO
      7050
7080 SUM=SUM+WC(I) : NEXT I
7160 ON F-1 GOSUB 2100,2200
7162 ON F-1 GOTO 7170,8000
7170 IF DIRECT=1 GOTO 7174
7172 OPEN "BILL1. DAT" FOR INPUT AS#2 : GOTO 7180
7174 OPEN "val2. DAT" FOR INPUT AS#2
7180 INPUT #2,SM,SL
7182 FOR I=0 TO M-1 : INPUT #2,SWC(I) : NEXT I : CLOSE#2
7190 IF WC(I)=SWC(I) AND L=SL AND M=SM THEN GOTO 6100
7200 RETURN
7240 SUM=0 : COLOR ,,0 : FOR I=0 TO M-1 : LOCATE 5,2+SUM+I : PRINT NMCOL$(I) : SUM=
      SUM+WC(I) : NEXT I
7250 COLOR ,,2 : FOR I=2 TO L : LOCATE 4+I, 2 : PRINT NML$(I) : NEXT I
7260 SUM=WC(0) : FOR I=1 TO M-1 : FOR J=2 TO L : LOCATE 4+J,2+SUM+I : PRINT LM$(J,I)
      : NEXT J : SUM=SUM+WC(I) : NEXT I
7270 GOTO 5090
7280 RETURN
7300 LOCATE 23,1 : PRINT " "
7310 LOCATE 23,1 : PRINT"" : PRINT NMCOL$(I) : PRINT"改为" : INPUT NMCOL$(I)
7320 SUM=0 : FOR I=0 TO M-1 : COLOR ,,0 : LOCATE 5,2+SUM+I : PRINT" " : PRINT SPC
      (WC(I)-2) : LOCATE 5,2+SUM+I : PRINT NMCOL$(I) : SUM=SUM+WC(I) : NEXT I : GOTO
      5481
7340 COLOR ,,2 : LOCATE 4+J,2 : PRINT " " : PRINT SPC(WC(0)-2) : LOCATE 4+J,2 : PRINT

```

```

NML$(J):COLOR,,0
7350 LOCATE 23,1:PRINT " " :GOTO 5410
7400 FOR I=0 TO M-1:SWC(I)=WC(I):NEXT I:SM=M:SL=L:PRINT "正在存入....."
7402 IF DIRECT=1 GOTO 7412
7410 OPEN "BILL.DAT" FOR OUTPUT AS#2 GOTO 7420
7412 OPEN "val2.DAT" FOR OUTPUT AS #2
7420 PRINT#2,SM,SL
7422 FOR I=0 TO M-1:PRINT#2,SWC(I):NEXT I
7430 FOR I=0 TO M-1:PRINT#2,NMCOL$(I):NEXT I
7440 FOR J=2 TO L:PRINT#2,NML$(J):NEXT J
7450 FOR I=1 TO M-1:FOR J=2 TO L:PRINT#2,LM$(J,I):NEXT J:NEXT I:CLOSE #2
7460 LOCATE 23,1:PRINT "数据已经存入"
7470 RETURN
7500 IF DIRECT GOTO 7504
7502 OPEN "BILL1.DAT" FOR INPUT AS#2:GOTO 7510
7504 OPEN "val2.DAT" FOR INPUT AS#2
7510 INPUT#2,SM,SL
7512 FOR I=0 TO M-1:INPUT#2,SWC(I):NEXT I
7520 FOR I=0 TO M-1:INPUT#2,NMCOL$(I):NEXT I
7530 FOR J=2 TO L:INPUT#2,NML$(J):NEXT J
7540 FOR I=1 TO M-1:FOR J=2 TO L:INPUT#2,LM$(J,I):NEXT J:NEXT I
7550 CLOSE #2
7560 GOTO 5230
7570 RETURN
7600 LOCATE 23,1:PRINT " "
7610 LOCATE 23,1:PRINT " ";PRINT NMCOL$(I);PRINT ""改为";:INPUT NMCOL$(I)
7620 SUM=0:FOR I=0 TO M-1:COLOR,,0:LOCATE 6,2+SUM+I:PRINT " ";PRINT SPC(WC
(I)-2):LOCATE 6,2+SUM+I:PRINT NMCOL$(I):SUM=SUM+WC(I):NEXT I
7630 LOCATE 23,1:PRINT " " :GOTO 5410
7640 LOCATE 4+2 * J,2:PRINT " ";PRINT SPC(WC(0)-2):LOCATE 4+2 * J,2:PRINT NML
$(J)
7650 LOCATE 23,1:PRINT " " :GOTO 5410
7660 SUM=WC(0):FOR I=1 TO M-1:LOCATE 4+2 * J,2+SUM+I:PRINT " ";PRINT SPC(WC
(I)-2):LOCATE 4+2 * J,2+SUM+I:PRINT LM$(J,I):SUM=SUM+WC(I):NEXT I:COLOR
,,0
7670 LOCATE 23,1:PRINT " " ,GOTO 5410
8000 IF DIRECT=1 GOTO 8004
8002 OPEN "BILL2.DAT" FOR INPUT AS#3:GOTO 8010
8004 OPEN "val3.DAT" FOR INPUT AS#3
8010 INPUT#3,SM,SL
8020 FOR I=0 TO M-1:INPUT#3,SWC(I):NEXT I:CLOSE#3
8030 IF WC(I)=SWC(I) AND L=SL AND M=SM THEN GOTO 6100
8040 GOTO 5090
8200 SUM=0

```

```
8210 COLOR ,,0: FOR I=0 TO M-1: LOCATE 6,2+SUM+I: PRINT NMCOL$(I): SUM=SUM+WC(I)
    : NEXT I
8220 FOR J=2 TO L: LOCATE 4+J*2,2: PRINT NML$(J): NEXT J
8230 FOR K=0 TO M-1: SUM=SUM+WC(K): NEXT K
8240 SUM=WC(0): FOR I=1 TO M-1: FOR J=2 TO L: LOCATE 4+2*J,2+SUM+I: PRINT LM$(J,
    I): NEXT J: SUM=SUM+WC(I): NEXT I
8250 GOTO 5090
8260 RETURN
8400 FOR I=0 TO M-1: SWC(I)=WC(I): NEXT I: SM=M: SL=L: PRINT "正在存入……"
8402 IF DIRECT=1 GOTO 8412
8410 OPEN "BILL2.DAT" FOR OUTPUT AS #3: GOTO 8420
8412 OPEN "val3.DAT" FOR OUTPUT AS #3
8420 PRINT #3,SM,SL
8422 FOR I=0 TO M-1: PRINT #3,SWC(I): NEXT I
8430 FOR I=0 TO M-1: PRINT #3,NMCOL$(I): NEXT I
8440 FOR J=2 TO L: PRINT #3,NML$(J): NEXT J
8450 FOR I=1 TO M-1: FOR J=2 TO L: PRINT #3,LM$(J,I): NEXT J: NEXT I: CLOSE #3
8460 LOCATE 23,1: PRINT "数据已经存入"
8470 RETURN
8500 IF DIRECT GOTO 8504
8502 OPEN "BILL2.DAT" FOR INPUT AS #3: GOTO 8510
8504 OPEN "val3.DAT" FOR INPUT AS #3
8510 INPUT #3,SM,SL
8512 FOR I=0 TO M-1: INPUT #3,SWC(I): NEXT I
8520 FOR I=0 TO M-1: INPUT #3,NMCOL$(I): NEXT I
8530 FOR J=2 TO L: INPUT #3,NML$(J): NEXT J
8540 FOR I=1 TO M-1: FOR J=2 TO L: INPUT #3,LM$(J,I): NEXT J: NEXT I
8550 CLOSE #3
8560 GOTO 5230
8570 RETURN
```

#### 参考文献

[1] 翟英编:《IBM5550 多功能电脑使用手册》,海洋出版社。

## 献锦囊妙计 展灿烂前程

### ——联合办刊年会圆满成功

<本刊讯>为了群策群力同心同德办好《计算机应用研究》杂志,本刊于今年9月中旬在广西召开了联合办刊会,获得圆满成功。

会上,杂志社主编和副主编对办刊现状、办刊宗旨、刊物演变及办刊方向等进行了较为详尽的介绍,通报了第四届全国计算机应用研究学术交流会概况;与会代表以满腔热忱的态度对刊物给予了充分肯定,以认真负责的精神对今后如何进一步搞好联合办刊,加强横向联合等进行了广泛探讨,发表了切实可行的有益意见,通过了联合办刊年会会议纪要。

本届联办年会受到广西、贵州等省区计算中心负责同志的大力支持与热情关照,本刊代表各联办单位顺致热忱!



## 全自动打印机共享器的研制

长沙铁道学院电子工程系 唐民山 华晓丽

**摘要:**一台 PC 机为获得硬拷贝输出,它必须配置一台打印机,但在多台 PC 机集中使用的地方,为节省昂贵的打印机投资,几台 PC 机共用一台打印机是可行的,我们已研制成功全自动打印机共享器,本文讨论它的电路工作原理,技术特性和使用价值。

### 一、概述

通常,一台 PC 机需要配置一台打印机,这不仅投资大,而且在场地比较紧缺的部门和单位也难以实现。一般情况下,打印机在工作中的利用率是较低的。那么,如何提高打印机的利用率,使在少量的投资下,实现打印机共享呢?RJS 型打印机共享器就是为了解决这一问题而研制的。

本打印机共享器属全自动型,即不需人工拨动开关来选择打印机为哪台 PC 机服务,而是完全采用数字逻辑电路的组合实现自动转换,它有四个输入端,一个输出端、四个输入端分别与四台 PC 机的标准并行接口相连,输出端与打印机相连,可以实现一至四台 PC 机共用一台打印机,在任一时刻只为一台已申请打印的

PC 机服务,而互不干扰。对用户来说,跟独占一台打印机一样。

### 二、电路工作原理

RJS 型打印机共享器是专门为具有标准并行打印机接口的四台 PC 机共用一台打印机而设计的,它由单稳态触发电路、译码电路、数据选择电路、互锁电路、驱动电路等组成。逻辑框图如图 1,电路原理如图 2。

静态时,四台 PC 机均无打印输出。各路选通信号  $\overline{\text{strobe}}$  均为高电平,单稳态触发电路均处于稳态,使互锁电路的输出对单稳态触发电路的输入无影响,即不封锁任一路选通信号,译码电路和选通门处于静态,从而使数据选择电路处于关断状态,无数据送往打印机。

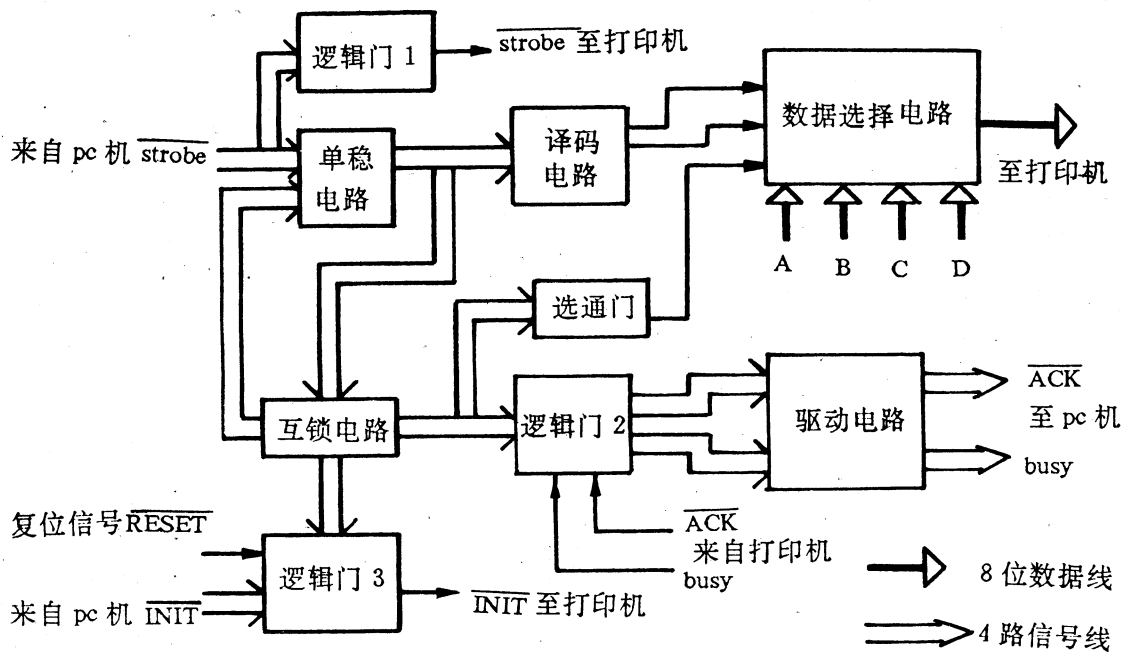


图1 逻辑框图

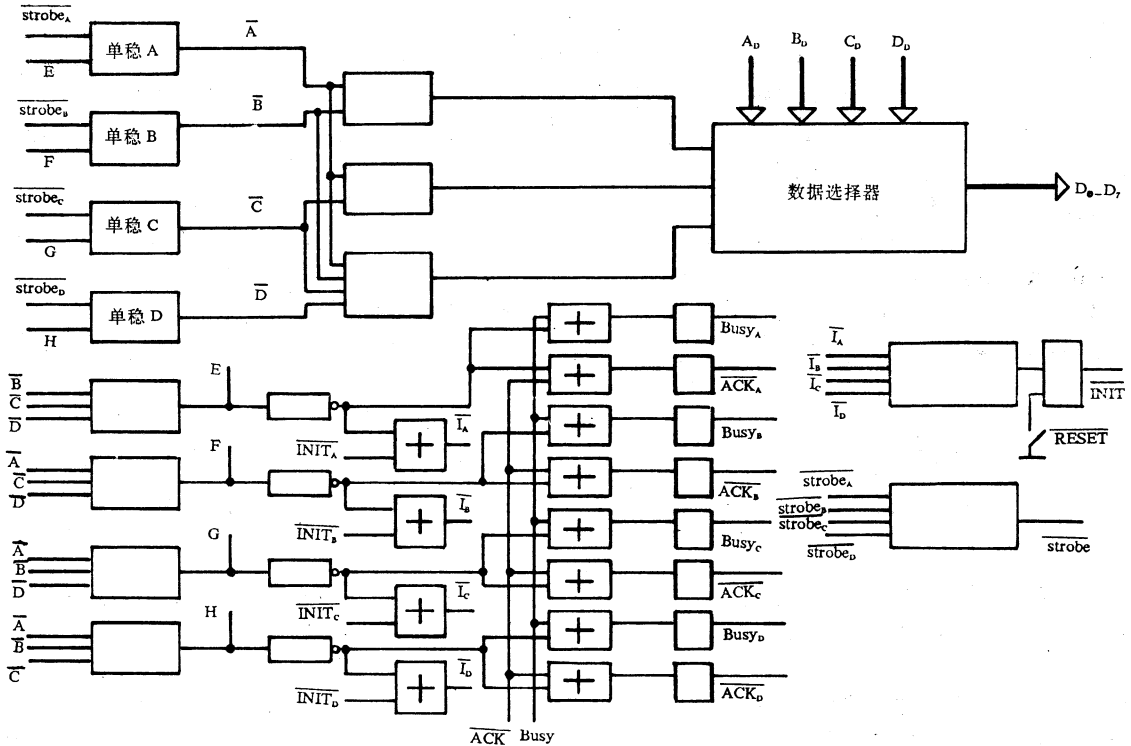


图2 电路原理图

当一台打印机发了打印命令时,则该路选通信号为一序列负脉冲,触发该路单稳态电路,使其输出在整个打印期间都保持高电平,而通过互锁电路封锁了其余三路的选通信号进入各自的单稳电路输入端,使这三路单稳电路不被触发而仍保持稳态。通过译码电路,选中该路的数据,通过选通门接通数据选择电路,从而把该路的数据送往打印机,该路选通信号通过逻辑门1再送往打印机。来自打印机的 $\overline{\text{ACK}}$ 信号及 $\text{BUSY}$ 信号也通过互锁电路后的逻辑门2再送至该路PC机,从而封锁了对其余三路的传送,这样就保证了工作的可靠性,避免了四台PC机可能出现的互相干扰,保证在此期间打印机只为该台PC机服务,直至该文件打印结束,整个电路恢复静态,等待接受下一个打印命令。

另外对初始化信号 $\overline{\text{INIT}}$ 的处理也作了考虑,因为打印机正在为一台PC机服务时,若另一台PC机启动进入DOS状态,就有可能导致打印机被再度初始化,从而破坏了当时的打印模式,致使打印数据丢失,打印输出

不正常,为了避免这种情况发生,在电路中利用了互锁电路的输出。通过逻辑门3,在静态时,四台PC机的初始化信号 $\overline{\text{INIT}}$ 通路都是畅通的,而在打印机为一台PC机服务时,则只有该路的初始化信号能通过,其余三路的初始化信号均被封锁,复位信号 $\overline{\text{RESET}}$ 是一个手动按钮产生的,在有需要时,可人为地对打印机进行初始化复位。

### 三、技术特性

1、四台PC机均可平等地独占打印机,同时申请打印服务时,则最先申请的先打,然后分别打印,其间有数秒钟的时间间隔,绝不会混打在一起。

2、四台PC机的工作与各单独接打印机一样,可自由开、关机,而不影响打印,也不致互相干扰。

3、PC机和打印机可以是任何型号的,只要具有标准并行接口就行。

4、工作环境条件与PC机及打印机相同。

5、电源:AC 220V  $\pm$  5%

# MCS-51 单片机系统中动态

## RAM 的刷新技巧

四川省电子计算机应用研究中心 林 化

MCS-51 单片机应用系统中,有时需要大容量的 RAM 存储器,如果仍采用静态 RAM,将大大增加成本,以至影响应用系统的生命力;并且大容量的静态 RAM 在国内市场也难于购买。用大容量的动态 RAM 取代静态 RAM,则可大大降低成成本,如  $4 \times 64K$ ,  $4 \times 256K$  以及  $4 \times 1024K$  的动态 RAM 价格远低于静态 RAM,在国内市场随处可见。这样的代换,性能并不受多大影响,又使应用系统成本降至人们完全可以承受。

51 单片机应用系统中引入大容量动态 RAM,当然还需对应应用系统进行适当修改;其一是设计动态 RAM 的地址译码和行列地址选择电路;其二是设计刷新电路,前者属常规设计,在此不多讲,而刷新则是应用动态 RAM 的难点,无论采用纯硬件还是纯软件完成刷新,都相当困难。充分利用 51 单片机特性,我们发现一种软、硬件结合的刷新技巧能够大大简化刷新电路的设计,硬件部分见示意图 1。

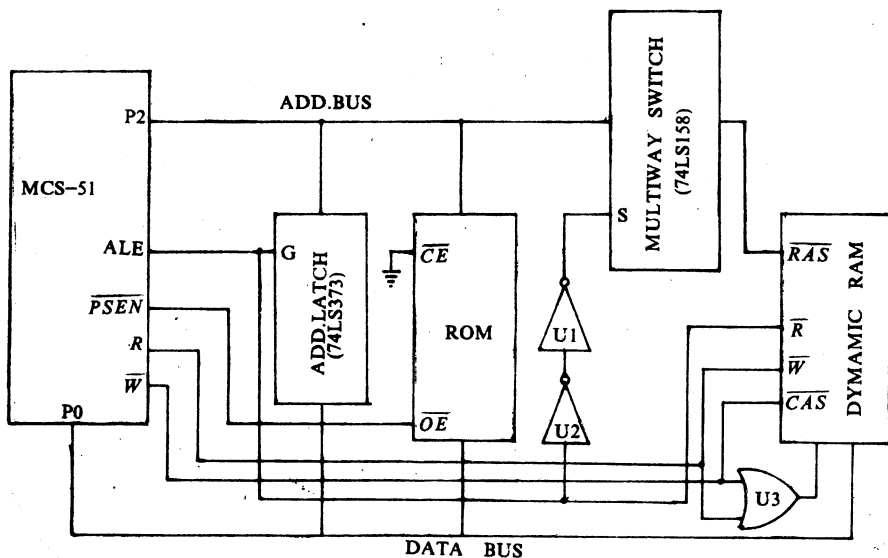


图 1 硬件框图

频率 50HZ ± 5%

功耗 2W

6、外型尺寸 245×169×70mm

7、重量 1.5kg

### 四、使用价值

目前我国广泛使用的 PC 机是一种个人计算机,在国外主要是供个人和家庭使用的,即使是单位和学校使用,由于他们的经济条件,每台 PC 机配一台打印机不成问题,既无共享打印机的要求,也就没有研制打印机共享器的必要,故国外未见不类似产品的问世,而在我国由于国情不同,国力所限,目前主要还是在一些

单位和学校集中使用 PC 机,但高档的打印机价格昂贵,一台 PC 机配置一台打印机即不可能,也无必要,因此实现打印机共享就有其很大的经济价值和现实意义。

RJS 型打印机共享器供四台 PC 机用户共用一台打印机,它完全采用硬件实现全自动工作,而无需人工干预,它体积小,重量轻,结构简单,工作可靠,用途广泛,实用性强,价格低廉,可以与具有标准并行接口的各种 PC 机跟各种打印机和绘图仪或接口相同的其它计算机外部设备配接,因而它特别适用于学校和科研单位。

图示应用系统中,低位地址为行地址,这样,刷新只需在软件中,让行地址(4464的 $A_0 \sim A_7$ , 44256的 $A_0 \sim A_8$ )在规定的刷新时间,刷新周期内翻转一次。51单片机提供了两个定时器中断,只需用一个中断来控制刷新(以下称之为刷新中断),在应用系统中,将该定时器定为短于规定的刷新间隔(4164为2ms, 4464同44256均为4ms),且设该中断优先级最高,技巧在于在该中断相应处理的程序中,什么工作也不做,只安排相应数量的空指令(NOP),使程序地址逐一增加(即依次翻转),硬件上已选择地址低位为行地址,所以这组空指令的执行,使动态RAM行地址得以翻转。如要求行地址 $A_0 \sim A_7$ 翻转,只需安排256条空指令,要求行地址 $A_0 \sim A_8$ 翻转,则只需要安排512条空指令,以此类推。这类空指令也可分几次执行,例如分两次执行,第一次执行一半数NOP,第二次执行另一半NOP,那么两次中断之间时间必须定为短于规定刷新时间的一半,且第二次NOP指令组始地址低位只能比第一次NOP指令组末址大一,同理可分三次,四次等执行NOP指令组。

图示应用系统中,执行刷新中断处理程序中NOP指令组时,51单片机只读ROM中指令,没有其它操作,所以读、写线( $\bar{R}, \bar{W}$ )始终为高电平,当然 $\overline{CAS}$ 也就为高了。 $ALE$ 信号平时为低,51单片机将低八位地址打入地址锁存器时,它变为高,经两级门延时,多路选择器选择端 $S$ 为高, $S$ 为高电平选择地址总线上的低位,即NOP指令组所在地址低位。之后 $ALE$ 回复到低电平状态, $\overline{RAS}$ 信号也就为低,此时动态RAM上行地址为NOP指令所在ROM中地址的低位,参见刷新时序图,尽管经两级门延迟后,多路选择器 $S$ 为低,但此时已完成指

令地址低位所指行的刷新。随指令地址逐一增加和NOP指令组的执行,也就完成了 $\overline{RAS}$ 为低时,动态RAM所有行地址的翻转,达到了动态RAM所有行的刷新。

对动态RAM的存取又是怎样完成的呢?因存取外部存储器,51单片机的 $\overline{PSEN}$ 信号为高禁止ROM输出, $ALE$ 信号为高将低八位地址打入地址锁存器, $ALE$ 信号经两级门延迟使多路选择器 $S$ 为高,则选择存取单元行地址。之后 $ALE$ 变低, $\overline{RAS}$ 为低,选择存取单元行地址,经两级门延时, $S$ 变为低,选存取单元列地址,存取操作必然有 $\bar{R}$ (读)或 $\bar{W}$ (写)信号有效,在 $S$ 变低后,读或写信号到达,使 $\overline{CAS}$ 信号有效,使列地址加到RAM上,这样数据可以从RAM读出或将数据写入动态RAM。

51单片机的寻址范围为 $0 \sim 0FFFFH$ ,只有64K寻址空间,所以当采用大容量动态RAM时,必须对地址译码,当然,刷新电路也必须做相应修改, $\overline{RAS}$ 、 $\overline{CAS}$ 、 $\bar{W}$ 、 $\bar{R}$ 信号等的关系可参照以上原理,刷新中断程序与上述雷同。

刷新时序:

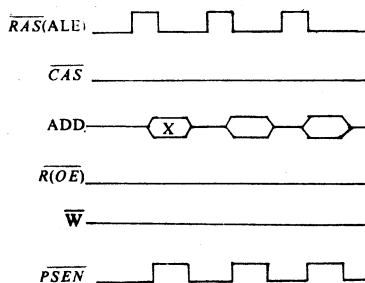


图2 刷新时序

交流学术的讲坛 传播信息的窗口  
展示成果的殿堂 推销产品的市场

## 第四届全国计算机应用研究学术交流暨新产品新成果展示会在蓉城胜利团幕

<本刊讯> 本刊与西南交通大学联合举办的第四届全国计算机应用研究学术交流暨新产品、新成果展示会于今年9月上旬在蓉城胜利召开,圆满成功。

此届学术交流暨产品成果展示会的议程有专题学术报告、产品成果展示、学术论文宣读、经验交流、评选优秀论文、颁奖活动等,内容丰富多彩。来自全国24个省市自治区的123名学术界代表和参展单位代表宣读交流了121篇具有新颖性、实用性、知识性、学术性、通用性的学术论文,其内容涉及系统管理、软件维护、硬件维护、人工智能、专家系统、图形图像、办公自动化、计算机网络、通讯、工业实时控制等等从一个侧面反映了我国当前各行各业推广应用计算机中所取得的丰硕成果,展示了我国信息产业的灿烂前景。会上经同行评议,专家评审,评出了优秀论文一等奖两名、二等奖六名、三等奖十名。

会后将出版学术论文集,向全国发行。

# MCS-51 单片机与 IBM-PC 微机的串行通讯

中国科学院光电技术研究所 肖时江

**摘要:**本文对 MCS-51 单片机与 IBM-PC 微机串行通讯中的一些关键技术问题,硬件连接、波特率选择、数据同步、以及多机通讯等做了较详细的分析,给出了多种具体实现方案,对广大的单片机用户具有较强的实用价值。

## 一、前言

MCS-51 和 IBM-PC 是目前我国应用最广的两种微型计算机。作为单片机,MCS-51 不但功能强、体积小、价格便宜,而且使用方便,特别市场上提供的多种 51 开发系统,为开发应用 MCS-51 单片机提供了非常方便的手段。因此,MCS-51 在许多领域,特别是工业控制和智能仪器已得到广泛的应用。有迹象表明,MCS-51 不久将取代 Z80CPU 而成为 8 位微机之首。作为系统机,IBM-PC 的优点已是众所周知的,可以说,IBM-PC 微机的出现为微机领域带来了一场革命。在我国,由于汉字操作系统(CCDOS)的引入,使得 IBM-PC 在各个领域得到迅速的推广使用。

串行通讯是计算机进行数据通讯的主要方式之一。由于其连线少,成本低,再加上有调制/解调功能,因而特别适合于距离较远且通讯点较多的场合,如各种计算机网络和分布式系统等。RS-232C 是最常用的一种串行通讯标准。在 IBM-PC 微机中,一般都有 1~2 个标准 RS-232C 串行口,简称 COM1 和 COM2。利用这两个串行口,IBM-PC 可以与其它数字设备(计

算机)进行一般的数据传送,或构成局部网络、多用系统和分布式控制系统等。

MCS-51 单片机片内即含有一个全双工的串行口,并具有多机通讯功能,可以方便地构成多机控制系统。但是,由于 MCS-51 的串行口并非标准的 RS-232C,加上某波特率是采用系统时钟由内部通用定时器产生的,误差大。而其数据处理速度也比系统机慢得多。因此当它与 IBM-PC 微机进行串行数据通讯,特别是构成分布式多机控制系统时,将会遇到许多意想不到的技术问题,诸如硬件连接、波特率选择、数据同步以及多机通讯控制等等。笔者根据多年的实践经验,对这些问题提出了一些具体解决办法,供广大同行参考。

## 二、硬件连接

RS-232C 采用负逻辑,其电平范围通常为<sup>[1]</sup>:逻辑“1”=-3V~-15V,逻辑“0”=+3V~+15V。由于 MCS-51 的串行口是 TTL 电平,故必须进行电平转换,一般都采用 MC1488(输出)和 MC1489(输入)来完成。其连接方法如图 1 所示。

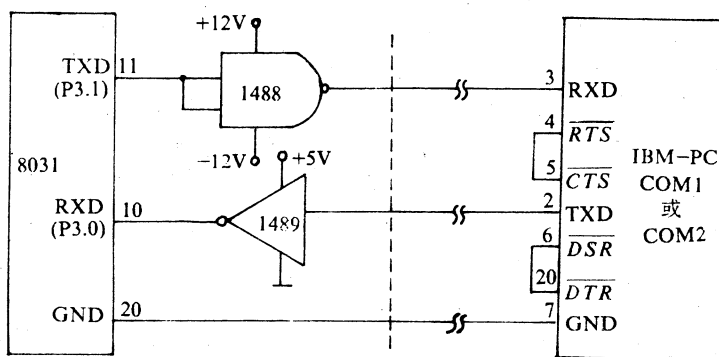


图 1 采用标准电平转换器的接口电路

图中,MC1488 采用  $\pm 12V$  电源,MC1489 采用  $+5V$  电源。IBM-PC 串行口中的第 4 脚 RTS(请求发送,输出)与第 5 脚 CTS(清除发送,输入),以及第 6 脚 DSR(数据

装置准备好,输入)与第 20 脚 DTR(数据终端准备好,输出)分别连接一起,主要是因为 PC 机 BIOS 中的 INT14H(串行通讯程序)在接收和发送数据时均要判

别CTS和DSR是否有效。如果用户自己编制通讯程序,采用判断收/发缓冲区是否满/空的办法来完成数据输入/输出,则可不必要短接。

顺便指出,IBM-PC及其兼容机串行口中的电平转换器通常有两种:老式原装机和长城0520机采用75150/75154,而一般的兼容机均用75188(MC1488)/75189(MC1489),1489中的每个接收门均有一个阈值电平控制端,当该脚开路或接一电容C至地时,其阈值电平 $V_T=1.1\sim 1.25V$ ,即当 $V_i\leq 1.1V$ 时, $V_o="1"$ , $V_i\geq 1.25V$ 时, $V_o="0"$ 。不难看出,这种电平与TTL电平具有兼容性。因此,图1中1488和1489均可用普通TTL反相器代换,从而省去了 $\pm 12V$ 电源!但要注意的是,替换1488的反相器最好用OC门,以便扩大电平范围和增加驱动电流。而替换1489的反相器输入端应加双向二极管电平钳位。此外,这种接法传送距离不宜太长,一般应在3m以内(标准RS-232C电平传送距离可达15m以上),距离过长,因两端地电位不一致以及噪声干扰导致电平范围出错。

上述做法虽然省去了 $\pm 12V$ 电源,但只适合于使用MC1488/1489的PC机,且传送距离近。为了与标准RS-232C电平兼容,又不用 $\pm 12V$ 电源,可采用图2

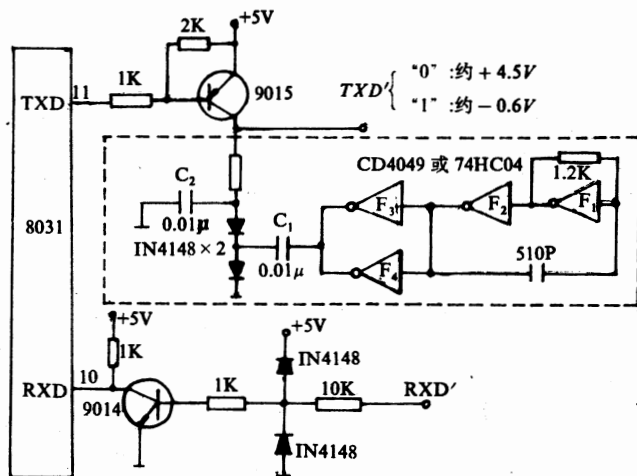


图2 准RS-232C电平转换电路

所示的准RS-232C电平转换电路。图中虚线框内电路产生约 $-7.0V\sim -9.0V$ 的负电压。其工作原理是,由CMOS反相器 $F_1$ 和 $F_2$ 构成的多谐振荡器产生约 $f=0.7MHz$ 的方波,经 $F_3$ 和 $F_4$ 双门驱动后,高电平时对 $C_1$ 充电,低电平时, $C_1$ 反过来又对 $C_2$ 充电,从而在 $C_2$ 上产生一定的负电压。这里采用CMOS反相器主要是为了减少其自身的功耗以及提高高电平电压。值得一提的是,如果8031的ALE脚负载不重,亦可直接利用它作为时钟源(1MHz)<sup>[1]</sup>,这时由 $F_1$ 、 $F_2$ 构成的多谐振荡器就可省去了,不过 $C_1$ 、 $C_2$ 要适当加大些。产生负电源还有一种既简单又实用的方法,如图3所示。如果将RTS引出,亦可产生 $+V$ 而代替图2中的 $+5V$ ,这样可进一步扩大电平范围。但要注意的是,图3中 $\pm V$ 是从信号中提取的,其驱动能力差,因此不宜带过重的负载。

从图2中可以看出,这种电平转换电路既省去了 $\pm 12V$ 电源,而且仅用二只廉价的二极管取代了价格较贵的1488/1489,因而大大降低了接口电路的成本,是一种非常实用的方法,特别是对于象前端控制机、单片机开发系统等提供 $\pm 12V$ 电源有困难的系统。

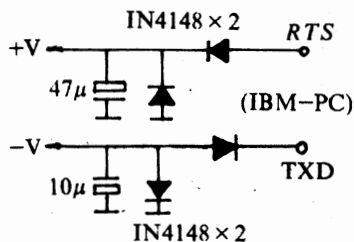


图3 产生 $\pm V$ 电源的简单方法

### 三、波特率的选择

波特率是串行通讯中的一个关键参数,通讯双方波特率的一致性直接影响数据传输的正确性,当误差太大时,甚至无法进行正常的传输。文[2]中指出,对于11位的串行帧(1位起始+8位数据+1位标志+1位停止),所允许的最大波特率误差应不超过4.5%。

RS-232C 通讯标准中,标称波特率有:50、110、150、300、600、1200、2400、4800、9600、19200等。其中以1200~9600最为常用。IBM-PC的串行通讯控制器采用的是Intel8250,其波特率是采用1.8432MHZ专用时钟由16位的除法器产生,可以精确地定出各种标称波特率。而MCS-51的波特率是采用系统时钟由定时器T<sub>1</sub>产生的,由于系统时钟一般为6.0MHZ或12.0MHZ,有效位数少,而T<sub>1</sub>又是用的8位自动装入定时方式。因此,很难精确地定出各种标称波特率。当两台MCS-51单片机之间进行串行通讯时,只要两机的系统时钟和定时常数一样,就能保证数据正确传输。这时的波特率只是反映传送速度的快慢,而不是非要达到多少精度。因为两机的波特率即使有误差,也只是系统时钟(晶体)上的误差,这种误差通常是很小的,可忽略不计。但当MCS-51与IBM-PC进行串行通讯时,这种误差就大了,我们以4800波特为例:

对于PC机,除法器控制常数为:

$$X = \frac{f}{16 \times B} = \frac{1.8432 \times 10^6}{16 \times 4800} = 24(0018H), \text{无误差。}$$

对于MCS-51,当系统时钟 $f_{osc} = 6\text{MHZ}$ 时,定时常数为:

$$n = 256 - \frac{f_{osc}}{32 \times 12 \times B} = 256 - \frac{6.0 \times 10^6}{32 \times 12 \times 4800} \approx 252.75$$

因 $n$ 必须为整数,四舍五入后 $n = 253(\text{FDH})$ ,反过来,因取整得出的实际波特率为:

$$B = \frac{f_{osc}}{32 \times 12(256 - n)} = \frac{6.0 \times 10^6}{32 \times 12 \times 3} \approx 5208$$

波特率误差:

$$\Delta B = \frac{5208 - 4800}{4800} = 8.5\%$$

比所允许的误差大,当然数据也就无法传输正确了。同理,不难证明,在1200~9600的标称波特率中,只有1200是唯一可用的!其 $n = 243(\text{F3H})$ , $x = 96(0060H)$ 。显然,对于那些想用高波特率传送数据的用户来说,这无形之中给出了一个难题。那么是否有办法解决这一难题呢?办法是有的,只是不能局限于几个标称波特率上。其算法如下:

①假定MCS-51机上的一个定时常数 $n$ ,一般 $n$ 取值为F4H~FEH(波特率在1200~9600之间)。

②根据公式 $B_0 = \frac{f_{osc}}{32 \times 12 \times (256 - n)}$ 求出波特率并取整(因只取整,故误差肯定 $< 0.1\%$ )。

③根据公式 $x = \frac{1.8432 \times 10^6}{16 \times B_0}$ 并取整求出PC机除法器控制常数。

④由于 $x$ 取整(四舍五入),导致波特率误差 $\Delta B = \frac{|B_1 - B_0|}{B_0} \times 100\%$ ,其中 $B_1$ 为实际波特率; $B_1 = \frac{1.8432 \times 10^6}{16 \times x}$ ,如果 $\Delta B \leq 4.5\%$ ,则波特率 $B_0$ 及参数 $n$ 、 $x$ 可用,否则不可用。增加 $n$ 值重复上述①~④步求出下一个可用波特率。

根据上述算法,不难求出1200~9600之间的所有可用波特率(假定 $f_{osc} = 6.0\text{MHZ}$ ),如表1所示。注意,如果使用表1中的非标称波特率来进行通讯,PC机程序就不能调用INT14H的功能0来初始化串行口波特率了,而只能采用下述子程序(以COM1为例),即直接往除法器送控制常数。

```
INIT PROC NEAR
    MOV DX, 3FBH
    MOV AL, 80H 指向除法器
    OUT DX, AL
    MOV DX, 3F8H
    MOV AL, XL
    OUT DX, AL 设置波特率常数
    INC DX XL为低8位
    MOV AL, XH XH为高8位
    OUT DX, AL
    RET
```

INIT ENDP

B <sub>0</sub>	n	x	ΔB	B <sub>0</sub>	n	x	ΔB
1302	F4H	0058H	0.5%	2604	FAH	002CH	0.5%
1420	F5H	0051H	0.15%	3125	FBH	0025H	0.4%
1562	F6H	004AH	0.3%	3906	FCH	001DH	1.7%
1736	F7H	0042H	0.5%	5206	FDH	0016H	0.5%
1953	F8H	003CH	1.6%	7812	FEH	000FH	1.7%
2232	F9H	0034H	0.7%	1200	F3H	0060H	0.1%

表1 可供选择的非标称波特率

### 四、数据同步

串行通讯分同步和异步两种方式,尽管MCS-51的串行口既可进行同步通讯(方式0)又可进行异步通讯(方式1~3),但IBM-PC机的串行口都只有异步方



式。在异步串行通讯中,波特率的一致性数据位同步的保证,而帧同步只能靠 $\overline{DSR}/\overline{DTR}$ 、 $\overline{RTS}/\overline{CTS}$ 等异步通讯控制信号了。遗憾的是 MCS-51 串行口没有这些控制信号,因此,当 MCS-51 与 IBM-PC 进行串行通讯时,虽然波特率选择一致,能保证单个字节传输正确,但都无法保证大块数据传输正确。例如,当 PC 机往单片机每次单独传送一个字节时, MCS-51 都接收正确,但连续传送一批数据时,由于单片机比 PC 机的指令处理速度慢, PC 机发送一个字节后,单片机可能还没取走, PC 机就发出下一个字节了,这样就会导致大量数据被丢失。由此可见,在 PC 机与 MCS-51 的串行通讯中,数据帧(1 帧=10 位或 11 位)同步亦是一个关键问题,下面介绍两种同步方法:

#### 1. 硬件方法

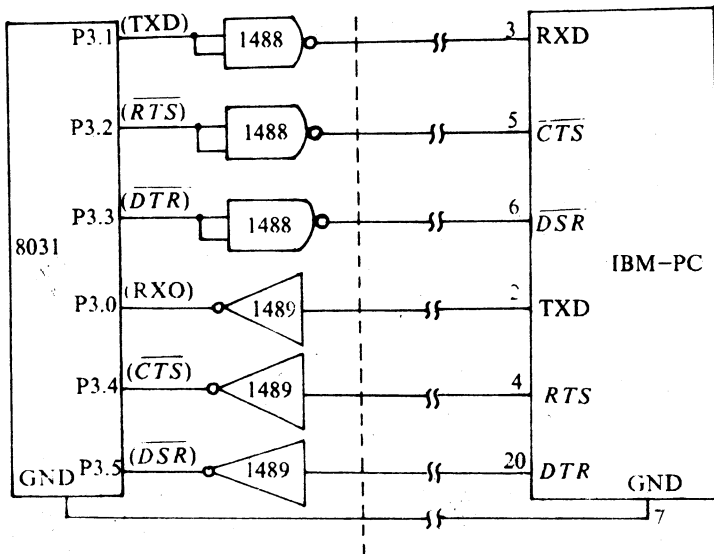


图4 异步通讯连线图

及时将刚收到的字节取走并处理完。这种做法的困难在于延时常数不好确定。一种改进的办法就是一方面发送方采取一定的延时,另一方面在发送一组(可以是一个字节,也可是若干字节)数据后,接收方应回答一个数据信号(不是控制信号!),该信号既可表示接收到的数据正确与否,又可起同步作用。

#### 五、多机通讯

MCS-51 单片机的串行口在方式 2(波特率固定)和方式 3(波特率可变)时具有多机通讯功能。这为组成多机控制系统提供了方便。IBM-PC 微机的软硬件资源相当丰富,特别是它的汉字图形功能,为用户提供

在图 1 的基础上,借助于 8031P3 口的其它位来模拟各异步通讯控制信号,其连接方法如图 4 所示。这样利用 $\overline{RTS}/\overline{CTS}$ 、 $\overline{DSR}/\overline{DTR}$ 就可方便地完成异步双向串行数据传输了。如果 PC 机程序不是通过调用 INT14H 而是自己设计的一个子程序来完成收/发数据的话,只要使用两组信号中任意一组即可完成异步传输,这样又可省去两根连线。

通过 8031 外部扩展一个异步通讯控制器 8251 也是一种行之有效的方法<sup>[1]</sup>,只是那样做以后,硬件成本增加,软件控制也变得复杂了。

#### 2. 软件方法

一种很容易想到的软件方法就是“延时法”。例如,当 PC 机要往单片机发送一批数据时,每发一个字节,就延时一段时间,以便单片机能

了一个非常“友好”的介面。因此,目前大多数分布式控制系统<sup>[3]</sup>均采用 MCS-51 作前端控制机, IBM-PC 作后台主机。这种控制系统对通讯接口提出了新的要求:首先,1 台 PC 机要带 n 台 MCS-51 控制机,故其发送数据总线(TXD)必须具有较大的负载能力;其次, n 台单片机要往 PC 机传输数据,必须解决数据总线(RXD)竞争问题;最后,这种控制系统的通讯距离一般比较远,环境也比较差。因此通讯接口必须具有较强的抗干扰能力。显然,标准的 RS-232C 接口难以满足这些要求。为此,必须采用一种新的串行通讯标准——RS-422A,在这种标准中,所有信号均采用平衡式差

分传输。MC3486/3487 是最常用的一种 RS-422A 电平转换器<sup>[4]</sup>,利用它构成的典型多机控制系统如图 5 所示。图中,线接收器 MC3486 的所有控制端均接 +

5V,即一直使能。单片机端线驱动器 MC3487 的控制端按 P1.0,平时应为低,只有在与 PC 机联系上需传送数据时才置高。如果 PC 机是巡查各单片

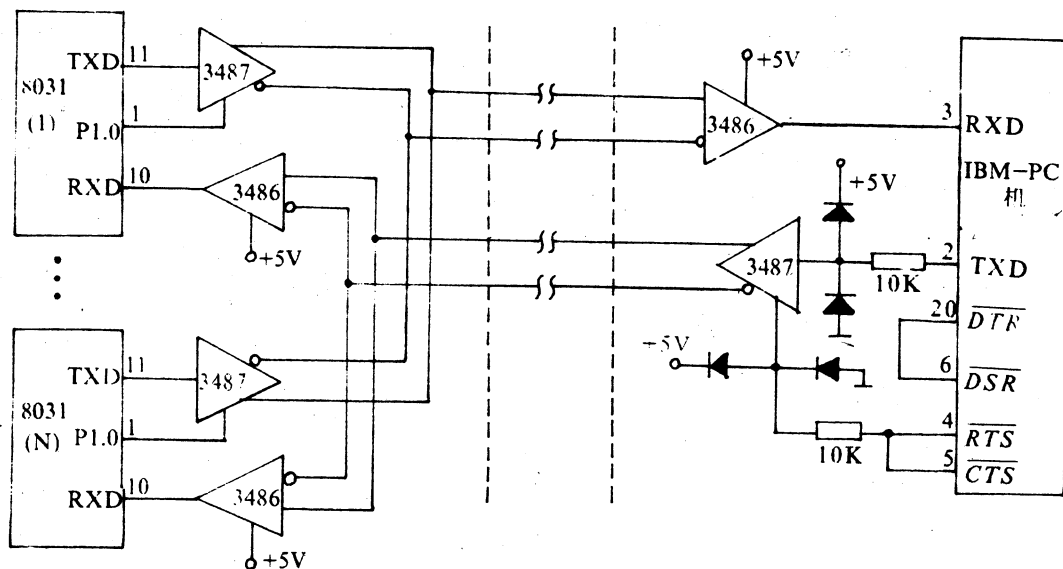


图 5 典型的多机控制系统

机,这样做就可保证数据总线不发生竞争。PC 机方的 3487 控制端是接在 RTS,平时为低,一旦发送数据就为高。注意,MC3486/3487 的输入/输出是 TTL 电平的,因此图中加了 4 只二极管钳位。此外,如果 PC 机 RS-232C 卡上用的不是 1489 而是 75154,那么 3487 的输出信号必须经过一个 RS-232C 电平转换器才能接到 PC 机的 RXD 端。这种接口电路在波特率不太高时,传输距离可达几公里,而且由于两机地线不连在一起,因此有一定的隔离作用。

在 MCS-51 的串行口控制寄存器 (SCON) 中,有一个“地址监听”控制位 SM2,当置 sm2=1 时,只有当接收到的串行数据的第九位 RB<sub>8</sub>=1 时,才激发“接收数据有效”标志 RI。而 PC 机发送数据时,可通过程序设置第九位为奇校验(1)或偶校验(0)。利用这两点,对于图 5 所示的多机系统,PC 机就可方便地与 N 台单片机进行数据通讯。假设 PC 机要从所有控制单片机巡回取一批数据,这时 MCS-51 和 PC 机的程序框图如图 6 所示。图中,PC 机必须将串行数据格式设置成 8 个数据位+1 个停止位,当附加奇校验位时,其控制常数(送 3BFH 寄存器)为 2BH,偶校验则为 3BH。发送数

据时如不调用 INT14H,则要注意将 RTS 置成“1”,以便打开 3487。由于是单片机往 PC 机传送大块数据,后者的处理速度快得多,因此不必采取帧同步措施,但为避免数据传输出错,可增加检错、纠错功能。

## 六、结束语

本文中介绍的大多数方法均已在我们的研制的“CHC905 长途电话台记帐号查询系统”,“LCC89 分布式粮仓温湿度监测系统”,“TOP-51 单片机开发系统”和“LDD422 长距离驱动器”得到了实际应用,其效果良好,值得推广。

## 参考文献

[1] 赵依军等编著,单片机接口技术,人民邮电出版社,1989 年。

[2] 于达仁,在 MCS51 单片机的串行通讯中实现波特率的自动整定,微计算机应用,1990 年第 4 期。

[3] 彭才浩等,微型机—单片机分布式数据采集系统,自动化与仪表,1987 年第 2 期。

[4] 李济芳,传输线接收器 MC3486 和驱动器 MC3487 及其在长线传输中的应用,集成电路及其应用,1989 年第 9 期。

# 微机在发动机生产装配线上的应用

吉林省计算中心 闫立恒 邢洪发

## 一、前言

某厂发动机生产装配线,有六台主要加工设备由国外引进,是用编程器实现控制功能的,200 左右个开关量和模拟量检测点分布在生产线的各处;关键点设置应答装置;并有对发动机缸体号检测用的读码器……等,要对这些进行实时数据采集,并把现场采集到的各种数据,送到上位机,以便对生产装配线的各种状

况进行监视和随机管理。

通过对现场实际情况的了解和分析,我们选用了 TP STD 总线微机作为现场实时数据采集的下位机, IBM-PC/XT 微机为上位机,组成了发动机生产装配线实时数据采集系统,经过设计、安装、调试运行使该系统达到了预想设计目的,效果良好。

该系统具有投资少,组成方便,扩展灵活,抗干扰

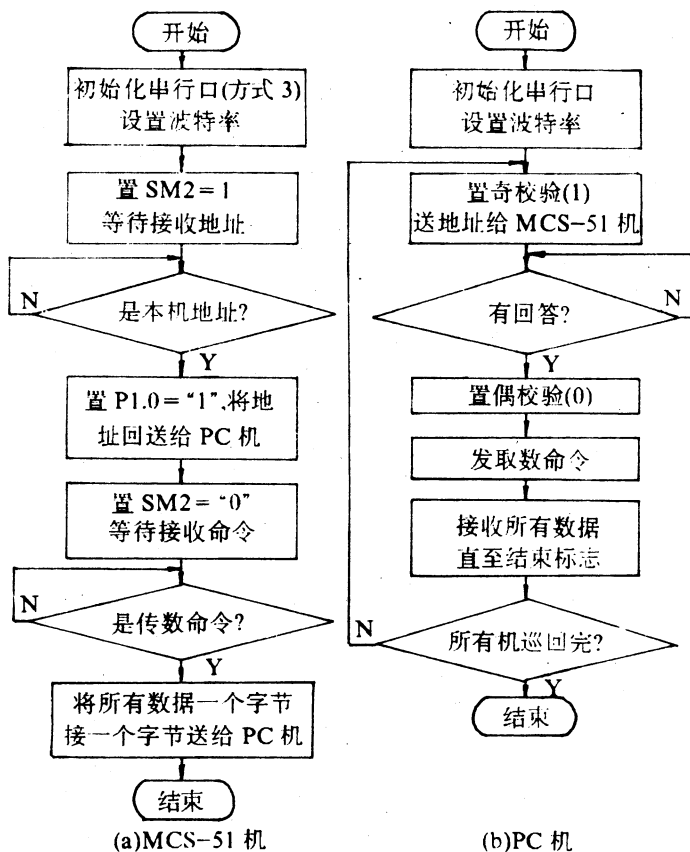
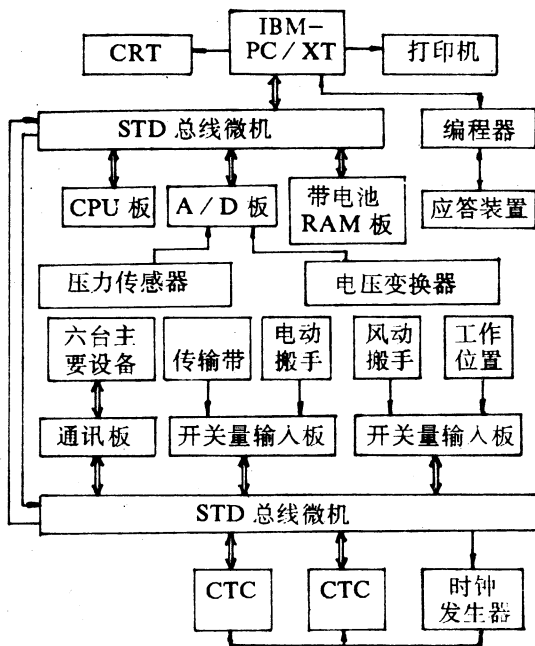


图 6 巡回取数程序框图

能力强,运行稳定可靠的特点。提高了发动机生产装配的自动化水平,提高了发动机生产装配的自动化水平,提高了发动机产品的质量和数量,使国产发动机生产和装配采用了微机技术,是改造旧的落后发动机生产装配线的发展方向。

## 二、系统的组成和作用

该系统有检测部分,通讯部分和应答部分组成。TPSTD总线微机作为现场采集用的下位机;打印机、显示器打印显示随机参数,统一由IBM-PC/XT管理,作为二级管理的上位机;这样该系统组成的总体框图如下:



检测部分:发动机生产装配线电源系统,包括:380伏、220伏、36伏、24伏、±12伏、5伏……等;送风系统:气动夹具,风动搬手等;送油系统:各种润滑油、冷却液送到工作位置;传输系统:有生产装配用的总传输线,各部件传送带把各种件送到工位;通风系统把有些工作位置处的通风点装置检测;各种工作位置的电动搬手……等190个模拟量和开关量点进行检测。

通讯部分:有六台对发动机缸体进行加工和检测

设备,是由国外引进的,组成发动机缸体加工生产和检测的生产线,有独立的编程器检测和控制,并提供了和系统能联机的RS-232接口。

应答部分:既整个生产加工装配线,重要岗位处,设有应答装置,以便使整个生产线实现现代化、自动化的调度和管理。这样组成的发动机装配线,在国内也是属于一流的,具有技术先进自动化水平较高的生产装配线。

1、模拟量输入部分,使用MS-1216A接口板,采用单一的+5伏电源供电。可采用十六路模拟量双端或单端输入方式,经过数据放大器将毫伏级信号放大成标准信号。

主要技术参数:

- (1)A/D:AD574,12位,单极性或双极性。
- (2)放大器增益:1—200倍
- (3)输入阻抗 $\geq 10M\Omega$
- (4)A/D转换时间 $35\mu s$
- (5)转换精度 $\leq 0.25\%$
- (6)通道数:单端32路,双端16路。

### 2、TP STD 88(C)04 多功能CPU板

TP STD88C04板是多功能CPU板,采用CMOS/TTL电路兼容设计,当使用CMOS芯片,具有可靠性高,功耗低,功能强,抗干扰性能好,工作温度范围宽的特点。两个串行通讯通道符合RS-232C接口电平标准。并提供了必要的握手交换信号,可以用于数据终端设备或数据通讯设备相接,板上提供的8位并行输出通道,既可工作在查询方式,也可以工作在中断方式。

### 3、存储板

TPSTD8701带后备电池RAM板,可以访问64KB存贮单元。使用6264RAM芯片,可对存贮在RAM区域中的数据实行断电保护,保存数据的功能在一个月以上。8701板可以全部使用CMOS芯片,同时具有与TTL板兼容、抗干扰强,功耗低,使用温度范围宽等特点。更为重要的是固化了的程序EPROM27C64可插在该板上运行源程序。

### 4、CTC控制板

TP STD8605CTC控制板,提供了三组共12个独立的可编程计数器/定时器接口电路,可对外部触发脉冲进行计数,也可以输出定时脉冲信号。占用16个连续的I/O地址,可在256个口地址浮动。

### 5、开关量输入板

本系统采用TP STD8526 32路光电隔离宽电压范围开关量输入板,是一种光电隔离开关量输入卡。使用TP STD8526板将受控开关输入量和计算机连接起

来,并能过 CPU 板访问 8526 板上四个字节(32 路通道)通道中的任一个,读取受控现场的开关量状态信息或数字量信息。受控现场直接相连的开关量输入接口线路与 STD 总线微机实现了电隔离。去掉了彼此间的公共地线和一切电气联系,从而免除了因有公共地线所带来的各种干扰和地转问题,使受控现场产生的各种具有破坏性的暂态过程与 STD 总线微机系统完全隔离了,因此可以保证 STD 总线微机在生产线上的应用系统,工作在平稳无干扰的环境之中。

#### 6. 通讯板

我们在设计时对该系统的考虑,这一部份尤其重要,是选用一台多路控制器,还是选用多块通讯控制板,经过多方面研究对比,从硬件技术、软件编程、成本上考虑的结果,我们选用了一块 TP STD8308, TP STD8308LINK 板是 TP STD 总线微机与 IBM PC 微机连接起来,能够进行串行通讯一块单通道串行接口板,用这块板实现上位机与下位机的通讯。另外我们又选用了多块 TP STD8307 板,由于接口电路采用 RS-423A 标准,与 RS-232C 完全兼容,并比 RS-232C 有更好的传输性能,用这样板实现生产线上主要 6 台设备进行通讯,统一在 TP STD 总线微机控制和管理。实践证明选用的配置是合理的。

#### 7. 报警

一条发动机生产装配线的实时数据采集和管理系统,由上位机和下位机组成,实现的监控和管理,一旦在生产和装配过程中,出现于设计时不相符的情况,报警系统出现声响报警,并随机显示和打印所发生的故障情况。选用 TP STD8511 光电隔离开关量并行输入输出控制板报警显示。

#### 三、软件程序设计与分析

数据采集处理与监控是在全部工艺过程中实现的,由编程器、STD 总线微机、及 IBM PC/XT 构成网络实现的,除了 6 台主要设备由编程器单独实现对缸体的检测和加工控制外,其余全部由 STD 总线微机实现的,其 6 台主要设备也与 STD 交换信息,这些统一由 IBM PC/XT 实现管理,并随机显示和打印记录。软件采用模块化程序设计规则,分别对各种模块进行单独设计。主要程序模块有:

1. 微机统一管理模块:主要完成年、月、日、时、分、秒的计时,数据采集处理、打印、显示和通讯功能。

2. 通讯模块有:IBM PC/XT 微机和 TP STD 总线微机的通讯功能,六台主要设备的编程器 RS-232C 口和 TP STD 总线微机的 TP STD8307 板的通讯功能模块。

3. 数据采集:对于 200 个点的采集,其中模拟量信号 21 个,开关量信号 173 个,以及 6 个数据口,对这些点进行循环的数据采集,并有时对某一模拟量点和数据口发生中断处理响应。

4. 对发动机缸体号,由条形码阅读器采集,配有 BHS-2000 系列 CCD 扫描器,能自动识别码制,并与串行口处输入,有独立的码制识别模块。

5. 应答模块:由编程器单独构成的应答系统,由编程器单独处理声响转换成码制,或码制转换成声响功能模块,由 IBM PC/XT 直接接受应答信息,并进入统一管理功能模块。

5. 输入输出模块;

6. 存贮模块;

7. 显示和打印模块;

8. 数字滤波程序模块:采用数字滤波方法,可有效的消除采集不真实情况,通过某种运算削去干扰信号,求得有用的信号比重大,这种滤波方法对低频和高频脉冲干扰较为有效。

$$Y_n = \frac{1}{K} \sum_{i=0}^{K-1} X_{n-i}$$

程序中,采样五次中取三求平均值法,作为一次采集的值。就是对某一点连续采集五次,相隔时间和采集完成一次的时间,要满足总体设计要求。去头舍尾,把所要部分三次求和,除了求得平均值,把最后结果送到寄存器中去,作为采集一次的值。

四、系统采用的抗干扰措施有:

#### 1. 提高电源抗干扰能力的措施:

- ①装设低通滤波器和隔离变压器,
- ②采用高抗干扰稳压电源,
- ③装设供电 ups。

#### 2. 防止外部干扰的措施:

- ①防止电容耦合干扰,
- ②防止电磁交链耦合干扰,
- ③采用光电隔离技术,
- ④防止地电流干扰,
- ⑤防止系统电源进线的干扰。

#### 3. 防止线间和线路干扰的措施:

- ①输入采集线要尽量短,
- ②把输入的采集线结合,绞合的双绞线降低共模干扰,

③输入采集线避免和动力线接近,并使用屏蔽线作输入线,只一端接地,

④模拟量采集线与开关量(数据线)及脉冲线不能接近或捆扎在一起。

#### 4. 滤波

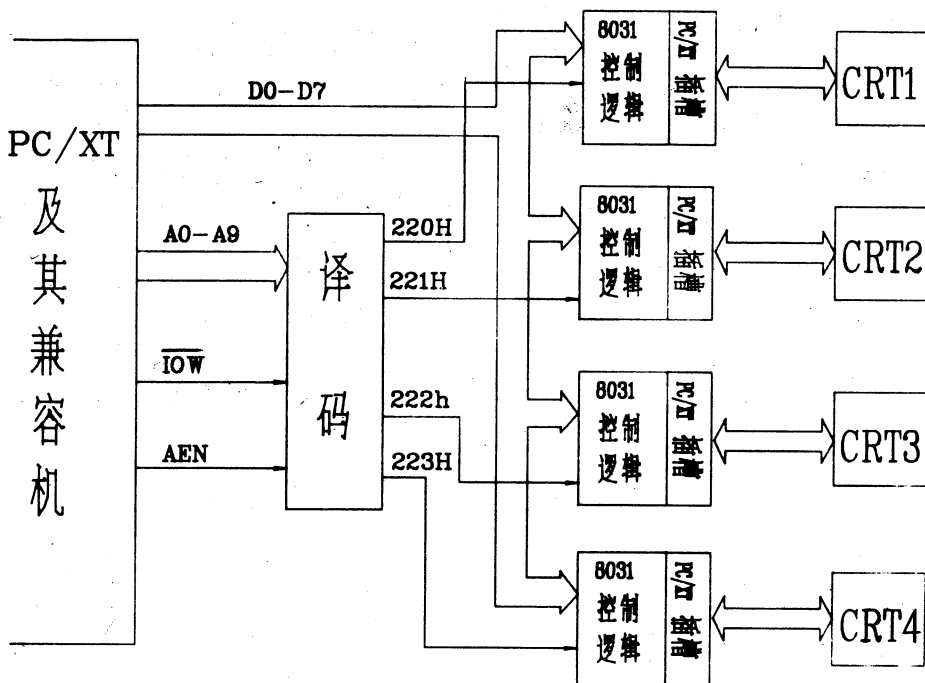
(下转 59 页)

# 用 8031 控制多屏彩色显示的实现方法

西北师范大学计算机系 王让定 王效先

一、引言:在多任务实时测控场合,往往需要多台 CRT 同时显示反映不同情况的信息,以便让操作人员及时了解生产及控制情况。因市场上所售微机一般只带一台 CRT,要实现多屏显示,最简单的办法是再购置几套微机系统,但此法投资大,资源浪费严重。在实践中,我们用 8031 单片机解决了在 PC/XT 及其兼容机

上实现一台主机驱动多台 CRT 的方法。此法仅用少量的硬件,组成适于彩色图形卡的插槽信号(标准 PC/XT 总线信号),彩色图形卡就插在该槽口上。图一给出了在 PC/XT 机上用 8031 扩充 4 台彩色显示器的原理框图。



图一 8031 扩充 4 台彩色显示器原理图

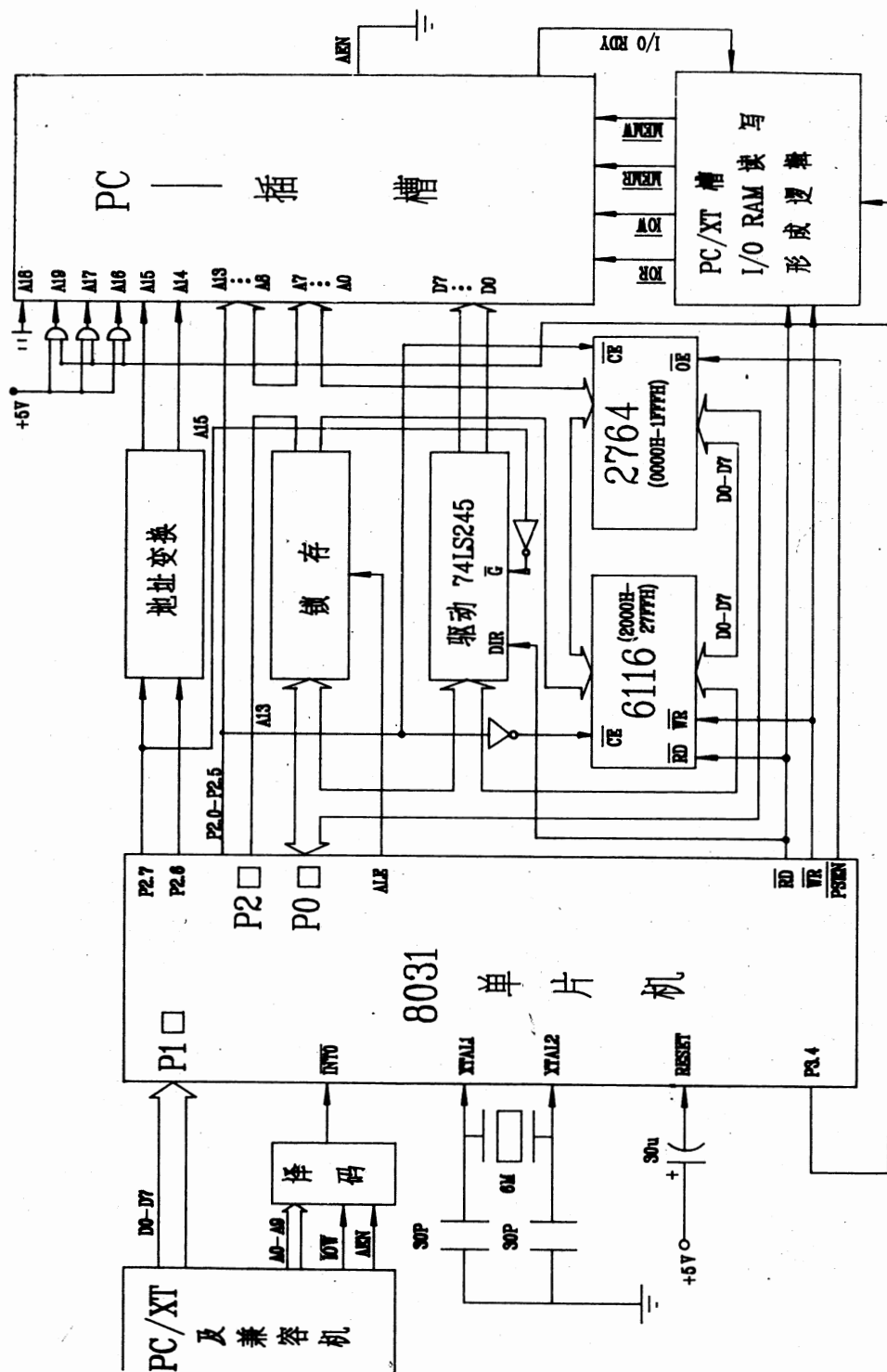
二、控制逻辑硬件组成,控制逻辑硬件如图二所示。

控制逻辑主要解决的是 8031 单片机如何控制产生 PC/XT 及兼容机插槽的地址信号、存储器读写信号、I/O 的读写信号及与主机的接口逻辑。

## 1. 地址变换

在 PC/XT 及兼容机上,彩色图形卡上的显示缓冲区位于 0B800H~0BBFFFH,占用 16K 的内存空间。在 80×25 的文本方式下,可存贮 4 屏显示信息,而在图形方式下,可存贮一整屏的 640×200 的单色图形或 320×200 的彩色图形。为了让 8031 控制彩卡,进行显示信息的传送,从便于硬件及软件设计角度考虑,彩卡

显示缓冲区在 8031 地址空间中定位于 8000H~0BFFFH。同样彩卡的 I/O 口地址在 PC/XT 及兼容机中是 3D0H~3DFH,而 8031 访问的 I/O 口地址定位于 83D0H~83DFH。为使 8031 控制彩卡,须将其 16 位的地址空间转换成 PC/XT 插槽的 20 位地址空间。为实现这个转换,所设计的 PC/XT 插槽的高四位地址信号 A19A18A17A16 接高低电平信号即可,即 A19=A17=A16=1, A18=0。插槽的其它 16 位地址信号 A15...A0 和 8031 的 A15...A0 对应相接。这样,8031 访问 8000H~0BFFFH 实际上访问的是 0B800H~0BBFFFH 的彩卡缓冲区。



图二 8031 控制逻辑组成图

## 2. RAM 及 I/O 读写控制

对 8031 来讲, RAM 和 I/O 是统一编址, 但在 PC/XT 机中, RAM 和 I/O 是单独编址。为此在图二中用 8031 的 P3.4 口线加以区别。具体讲, 当 P3.4 输出为高电平时访问内存(彩色缓冲区), 当 P3.4 输出为低电平时, 访问 I/O 口。亦即, 在 8031 访问显示缓冲区时, 先执行 SET P3.4 指令, 置 P3.4 为“1”, 在访问 I/O 口时, 先执行 CLR P3.4 指令, 置 P3.4 为“0”。

另外, 用 8031 的  $\overline{WR}$ 、 $\overline{RD}$ 、P3.4 及彩卡 I/ORDY 信号的逻辑组合产生插槽的 MEMR、MEMW、 $\overline{IOR}$ 、 $\overline{IOW}$ 。

## 3. 8031 控制逻辑与主机接口

主机通过并行口向 8031 发送数据, 8031 利用口址译码产生的选通信号接至单片机  $\overline{INTO}$  产生中断请求, 并从 P1 口读入信息。图一中的 4 个 8031 控制逻辑与主机的口址为 220H、221H、222H、223H。只要主机执行下面三条命令, 就可将信息送到对应口址的 8031 进行处理。这三条命令为:

```
MOV AL, nn      ; nn 为发送数据
MOV DX, ADRS    ; ADRS 为口址
OUT DX, AL
```

三、控制逻辑软件设计简介: 软件由主程序和中断服务程序两部分组成。程序流程见图三所示。

在所设计的 8031 控制程序中, 彩卡的工作方式限为文本方式或 320×200 的彩色图形方式, 至于具体设为何种方式, 取决于主机所发命令。8031 将主机发送来的命令和信息进行相应处理, 然后将处理完的数据组合成显示信息送往彩卡显示缓冲区。

在文本方式下, 一个显示信息由双字节代码组成, 其中一字节代表显示字符的 ASC II 码, 另一字节是属性码。所有显示信息代码都邻接地存入彩卡缓冲区内, 以便在显示屏相应位置上显示该字符。若要将某个字符显示在屏的某行、某列时, 可由下列关系求出显示信息在显示缓冲区中的存放地址。

$$x = m \cdot 8000H + (160 \cdot (l-1) + 2(p-1))$$

$$y = m \cdot 8001H + (160 \cdot (l-1) + 2(p-1))$$

其中: x—显示字符的 ASC II 码地址;

y—显示字符的属性码地址;

m—页(屏);

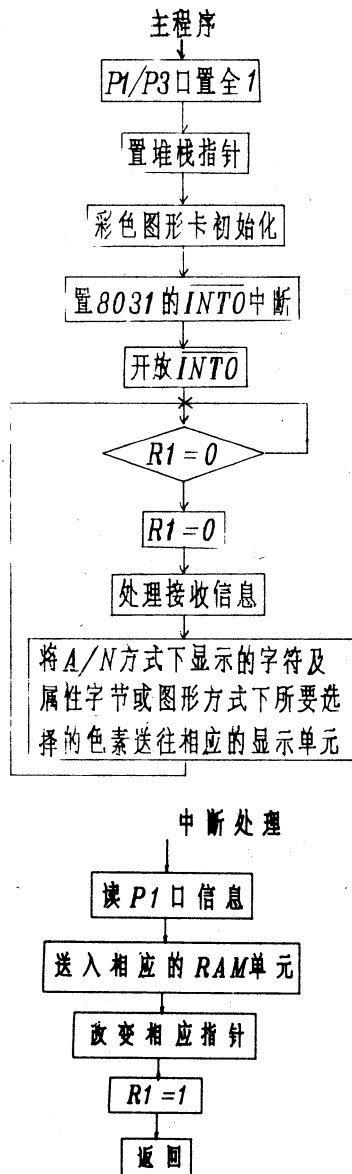
l—显示行数;

p—显示列数。

m、l、p 的取值范围是:

$$1 \leq m \leq 4;$$

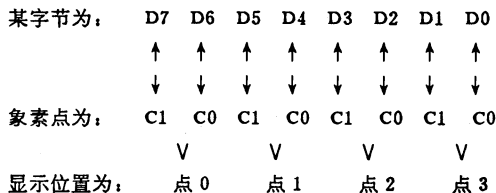
$$1 \leq l \leq 25;$$



图三 控制逻辑软件流程

$$1 \leq p \leq 80.$$

320×200 彩色图形的显示信息由象素点代码组成, 每一字节对应四个象素点。组成格式如下所示:



某点色形的选择依靠 C0、C1 及彩卡 3D9H 口的组合而定。

320×200 的一整屏图形方式共有(下转 1 页)



## Disk Manager 在硬磁盘故障诊断中的应用

长春市 81021 部队自动化工作站 范立果

微机硬盘在使用过程中出现故障,一般分为两大类:一是即系统硬件故障(控制卡、磁盘本身、及系统主板控制部分的芯片等故障);二是所谓的软故障,即硬盘不能自举、不识别硬盘、进入固化 BASIC, DOS 的内部与外部命令不能对硬盘进行读/写、校验操作,甚至一些常用的检测及初始化程序不能正确运行。在此为大家介绍一下 Disk manager 软件在硬盘故障诊断中的应用。该软件集检测、诊断与维护及初始化于一体,并可对硬盘控制(卡)测试、磁盘驱动器测试、还具有对检测到的错误进行清除和孤立的能力,以使用户确定故障的位置等许多实用的功能,并在不损坏硬盘现有信息的情况下对硬磁盘进行信息传送能力的检测、读/写信息测试及硬盘的初始化、重新定义硬盘分区等,赢得了许多微机用户的青睐。下面就一般检查硬盘方法从检测和初始化两个方面介绍该软件的使用方法和一些特殊功能。

一、当确认硬盘发生故障时请不要急于利用各类初始化软件对硬进行信息改写等操作,以免对硬盘上数据信息破坏,首先应利用 disk manager(磁盘管理,以下简称 DM)系统磁盘上的 diag. exe 文件进行硬盘的硬件进行综合测试诊断。具体方法是:

在当前系统盘所在的提示符下如 A: dig 回车(单

显)或 A: diag/c 回车(彩显);如果你的计算机是 AT&T 6300plus 计算机或使用 XT 控制卡的 AT 机请使用: A: diag/p 回车

1、系统提示请选择对一个还是二个驱动器进行测试:

select drive (1 or 2) 1 回车

2、显示 diag 软件识别的硬磁盘驱动器的配置:(以 10 兆硬磁盘为例),并请你认证(Y/N)?

current default size.....10.6 megabytes 306 cylinders  
4 heads do you wish to use these parameters? (y/n) y 回车

3、进入主菜单,以下操作均使用 F1——F10 的类型键进行:

F1. RUN ALL TESTS (运行全部测试)

F2. RUN INDIVIDUAL TESTS (单项测试)

F3. RUN REPEAT THE TESTS (重复测试)

F4. ALLOW C. E CYLINDER WRITES (C. E 柱面写测试)

F5. STOP ON ERROR (错误时停止开关)

F6. SELECT DRIVE TWO (选择两硬磁盘子系统测试)

F7. FORMAT C. E CYLINDER (格式化 C. E 柱面)

F8. DISPLAY ALL DISK PARAMETERS (显示全

(上接 55 页) (1)直流电源滤波

(2)RC 线路滤波

(3)数字滤波

①程序限幅方法

②求算术平均值法

③中值法

④采样时间的选取,对于模拟量采样时间最好选取工频周期(例如 20ms)的整数倍。

5、接地防止干扰

五、主拖动系统

所谓主拖动系统是控制生产线运转的拖动装置。过去大多数采用交流电动机拖动,但是需要多级齿轮变速,才能使生产线在低速下运转,变速机构装置庞大。或需要庞大的液压装置才能实现生产装配线在低速下运转。上述两种情况其速度是一定的,要想对生产线运转速度进行调正,工作量很大又麻烦。

现在我们采用编程器输出脉冲,控制可控硅导通角,使可控硅供电控制生产线拖动电机的电源,实现生产线低速运转的需要,而且生产线的运转速度调正是比较方便的,不需要耗费较大资金,只要变更编程程序的某些参数,改变编程器输出脉冲个数,便可以改变可控硅导通角,改变可控硅输出电压,改变拖动生产线电动机(交流机、直流机)转速,使生产装配线在低速下运转。可见用这种方案去控制主拖动系统,成本低、使用方便、维修简单,是改造旧生产线主拖动系统的措施,尤其对于经常变化的生产线,更为合适。

六、结束语

TPSTD 总线微机在生产装配线上的应用,经过设计,安装、调试运行效果很好。这样的生产线组成另活、投资少、使用方便,见效快,是旧生产线拖动装置的改造和新生产的设计,可选用的一种最佳方案。

部硬盘参数)

#### F9. COLOR ON/OFF (彩色显示开/关)

(1)、选择 F1, 程序执行全部的测试功能, 即进入下一级具体测试菜单, 且该菜单所提示的三种测试均一步一步自动进行测试, 测试完成后返回主菜单, 无须进行干预, 显示如下:

----->F1 对硬盘控制卡进行的穷举控制测试, 这时硬盘可以是断开或连接着的, 这样以便确定故障是控制卡或其它单元上的位置。

F2 对硬盘进入故障查找测试, 这是对硬盘驱动器进行的任意查找测试, 也是对所怀疑的故障进行孤立的一种方法。

F3 任意测试, 是一个任意对硬盘的信息进行全面测试。

注: -----> 为正在进行的当前测试项

(2)、选择 F2 后提示你从测试菜单里选择一个测试项目进行测试:

这个测试菜单中的 F1——F3 同主测试菜单同主菜单上的 F1(运行全部测试)中的 F1——F3 功能一样, 只是供选择的使用其中一项进行测试。

F4 读/写测试, 这是对保留给用户的 C、E 柱面进行读/写/比较操作, 这个测试必须是先已对 C、E 柱面进行了格式化的, 然后进行读写操作, 主要用于对怀疑硬盘有写故障的时候使用。

F5 校验读/写测试, 这是对硬盘 C、E 柱面进行强制性的错误校验, 用于验证写入数据的准确性。运用它的前提必须是主菜单上的 F4 与 F7 必须是已经进行了选择。

F6 校验磁盘信息传送测试, 如果需要对硬盘信息传送测试, 可以使用这个选项, 这是一个没有破坏性的检验, 它不向磁盘表面上写任何信息。这个测试是不执行主菜单上的 F1 功能的(运行全部测试)。

F7 返回主菜单。

(3)、重复测试, 这个选择循环一个测试项接一个测试项进行测试, 用这个选择可以帮助发现周期性故障, 也可以选择一个单独项进行循环测试, 即选择 F3 后接着选择 F2——任意单项测试。或选择 F3 后选择 F1——选择几种方式进行组合循环测试。

(4)、选择主菜单上的 F4 功能键, 这个选择实际上允许诊断程序对一个标准 DOS 不使用的柱面 C、E 柱面进行写测试, 以验证硬盘系统在最初安装、设置时的读写能力。

(5)、F5 这个选择提供一个能在诊断一些错误时能从诊断循环中停下来, 以便分析故障原因。它是一个

复位式开关, 高亮度状态为错误停止状态有效。

(6)、F6 可以使用户在已经进入 DIAG 诊断主菜单后再提供一个对二个硬盘子系统进行选择的机会。

(7)、选择 F7 即对硬盘上 DOS 保留给用户的 C、E 测试区进行格式化, 以为对该柱面写入信息和读出信息做准备, 供校验硬盘的读/写能力时使用。

(8)、主菜单 F8 这个选择, 为在执行诊断硬盘的用户在屏幕上提供一个全面的当前硬盘参数, 便于诊断、维修时参考。(见附表)

(9)、F9 即在主菜单下选择单色/彩色显示器方式。

二、采用上述各项测试基本会找出故障的所在, 如果存在硬件故障, 多数不会破坏磁盘表面信息, 更换芯片后一般可重新启动, 否则存在的硬盘软故障, 即使使用系统磁盘上的 DM. EXE 程序(常说的 DM 法)进行恢复或初始化, 方法如下:

1、在当前盘状态下键入 DM 回车即自动定义分区方式。

在菜单的首行显示 DM 软件所识别的当前硬盘状态信息, 然后请你确认(Y/N)? 即, only drive detected is this correct? (Y/N) y 回车

程序进入硬盘类型选择项即提示, defaulting to drive?

enter model clde for drive (return if stoned ard);

如果这时回车 DM 软件会自动识别进行是否对坏磁道进行登记然后进入主菜单: (G)et, (W)rite the defect——map file

(C)lear, (A)dd to,

(D)elete, from the defect——list

(R)eturn to initialization menu

select an option (R); R 回车

G 为获取磁盘列表文件; W 为写磁盘列表文件; C 为清除; A 为加上柱面号; D 为删除柱面标号。这时选 R 进入初始化菜单:

这时提示一些是否建立磁盘列表文件等信息, 最后提示:

do you wish to proceed with the installation as described above? (Y/N) Y 回车

按 Y 后开始格式化, 格式化完成后系统提示插入 DOS 盘安装引导系统, 最后拷贝 DM 文件, 返回主菜单。

2、当前盘状态下键入 DM/M 回车即为人工定义分区方式。进入主菜单后显示: 当前硬盘开始、结束柱

## 微机数据通信及其发展前景

吉林省计算机技术研究所 张鸿鸣 邢洪发

随着微型计算机的发展,数据通信也已广泛地应用到各个领域。因此,研究微机数据通信及其发展前景十分必要。

### 一、微机数据通信

微机数据通信的实现是通过微机组成的数据通信系统来完成的,所谓微机组成的数据通信系统也就是把中央微机和分散在各地的很多终端设备或微机用通信线路连接起来的系统。各地产生的数据和其他信息从终端机输入,经过通信线路传送到中央微机进行处理。之后,再把结果送回原来的终端机或其他被指定的终端设备。

研究微机数据通信,就有必要首先掌握数据码、传

输方式、通信线路及通信方式、通信接口标准等数据通信技术。研究微机数据通信发展前景,才能充满信心地推广微机数据通信技术,使其更好地为社会主义建设服务。

#### 1. 数据码。

数据是用二进制数表示的数字构成的,用二进制数字“1”和“0”组合成的代码来表示字符,再由若干个字符构成数据。把一位二进制数字称为一个比特,因此,每个字符都由数个比特组成。

国际标准化组织已提出了关于国际标准代码的建议,这种代码是由7个信息比特和一个差错控制的比特来表示。采用这种代码可以表示  $2^7=128$  种组合,其

面、类型(DOS)、是否引导、DOS名字、版本。然后是 MAIN MENU;

(1)initialization menu, (p)artitioning menu,

(S)elect drive, (C)onfiguration menu, (R)eturn to DOS

Select an option(R);

在主菜单下键入:I为初始化硬盘;P进入定义DOS分区菜单;S为选择驱动器;C为配置结构菜单;R返回DOS。

选择I为系统重新生成这时硬盘中数据会全部损失,即:

主菜单下键入I回车显示:

Does the above PARTITION TABLE require modification (Y/N);Y回车

系统进入配置菜单:

(A)llocate, (D)elete, (S)elect boot, (C)hange type, any partition

(N)ew partition table, (R)eturn to preparation menu

Select an option(R);

A 硬盘格式化时以柱面划分分区大小;D为删除已存在的DOS分区;S为选择系统引导;C为改变DOS分区的类型。N为新的分区表;R为进入分区菜单,提示对硬盘写分区数据等信息,逐次回答、认证后即可修复、改写分区,及系统配置的D盘进入重新定义分区,以避免破坏C盘或其它数据。

在选择主菜单下的I或P后对硬盘初始化或分区后DM管理系统均提示插入DOS系统盘,进行传送系统文件和DM文件,然后按提示返回主菜单下,即可完成对硬盘软故障的测试和恢复。

以上只是在使用中的几种常用功能及使用方法,还有不少实用的功能,在这里不细述,还有不到之处请指正。

附表:

例,10兆硬磁盘,两硬盘状态的参数如下:

PARAMETERS	DRIVE	DRIVE
	ONE	TWO
SETUP TYPE	0	0
MEGABYTES	10.6	10.6
CYLINDERS	306	306
HEADS	4	4
REDUCED WRITE CURRENT	306	306
WRITE PRE-COMP	0	0
ECC CORRECTION SPAN	11	11
CONTROL BYTE	5	5
STANDARD TIME OUT	12	12
FORMAT DRIVE TIME OUT	180	180
CHECK DRIVE TIME OUT	40	40
LANDING ZONE	0	0
SECTORS PER TRACK	17	17

中 32 种组合被利用于控制功能,这种代码被称为 CCITT5 号码。CCITT5 号码相当于美国称为美国信息交换标准代码 ASCII; ASCII 和 CCITT5 号码差别不大,大部分数据终端设计时都采用这种代码。

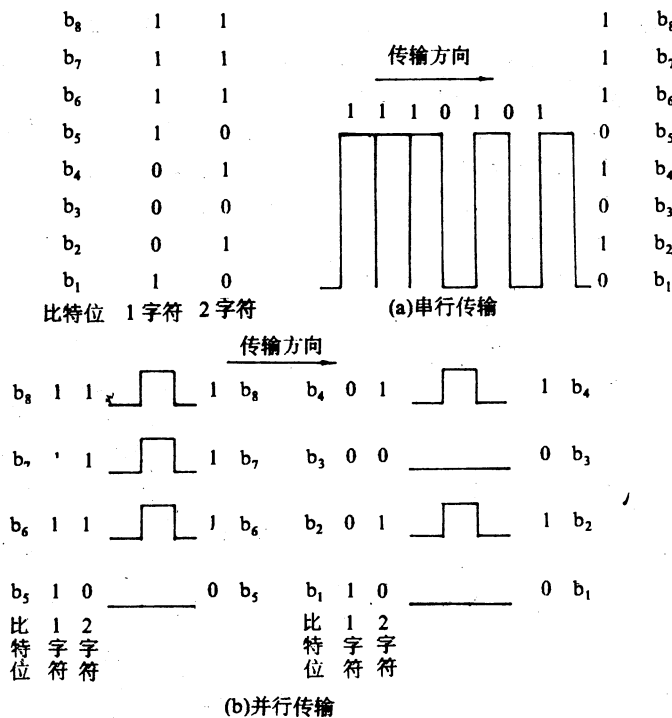
在电传机上使用的代码被称为博多码或 CCITT 号码,它每个字符 5 个比特,所以可组成不同的可用的字符有  $2^5=32$  个。利用一个专用的换档键,可把字符的数目增加。IBM 终端机上使用的代码同 CCITT5 号码不同,它被称为扩充的二——十进制交换码,每个代码

由 8 个信息比特所组成,因此,共有  $2^8=256$  种组合。

## 2. 传输方式。

(1) 串行及并行传输。在通信线路上传送给对方的方式有两种,即为串行和并行传输。串行传输是把每个字符包括的比特按顺序从头至尾一个接一个传出去;并行传输是把每个字符包括的比特同时传出去。

因为并行传输的技术要求比较高,有关硬件及通信线路的价格比较高(线路数量比较多),目前的数据传输,几乎都采用串行传输。



图一 串行和并行传输

(2) 异步和同步传输。异步传输也可称为起止式同步传输,这种方式是在每个被传输字符前附加一个“0”比特作为起止比特,在字符后附加一个或二个“1”比特作为停止比特,收信端以停止比特和起始比特间的转换来识别各个字符范围。在起止式同步中,每传输一个字符都要发送最少两个多余的比特,因此,便于降低传输的效率,这种传输方式目前多用于低速传播。

为了提高传输效率,最理想的情况是所传输的比特全是数据比特减去多余的起止比特,为达到这个要

求而采用的传输方法被称为同步传输。在传输开始时,用同步字符(通常是 SYN 字符)使收发双方终端进入同步,此后就从传输数据中抽出同步信息,修正同步性,由此保持同步。传输时,时钟信息也含在数据中。

ISO/CCITT 规定 SYN 字符专用码为 0010110,在接收端只要检测字符的首位便能取得同步。当没有需要发送数据时,而是发送 SYN 字符,所以能不间断地继续传输数据。由于这个原因,同步方式适用于高速数据传输。

### 3. 通信线路及通信方式。

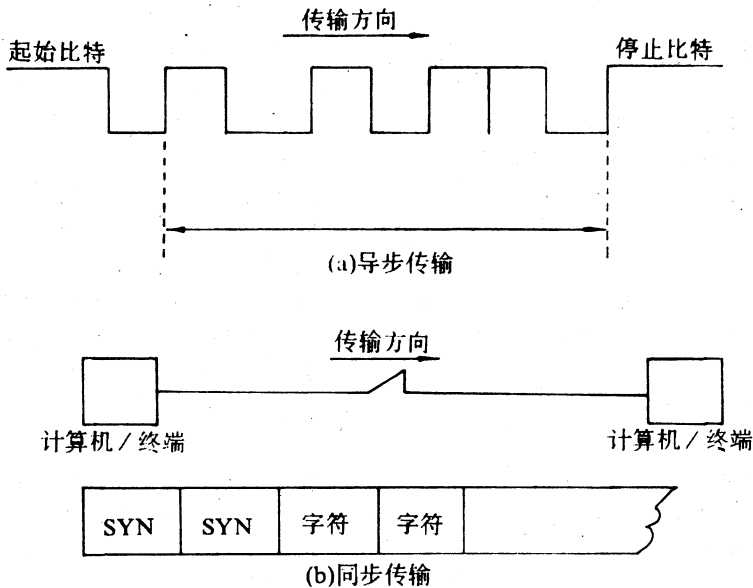
(1)通信线路。发送终端和接收终端若用一对线路连接起来进行数据传输,这种方式叫二线制;若用两对线连接起来进行数据传输,则叫四线制。

通信线路亦可分为用户专线、租用专用和拨号线。用户专线主要是用户自己架设的线路;租用专线是用户向电信公司租用的线路;拨号线是同打电话一样,每

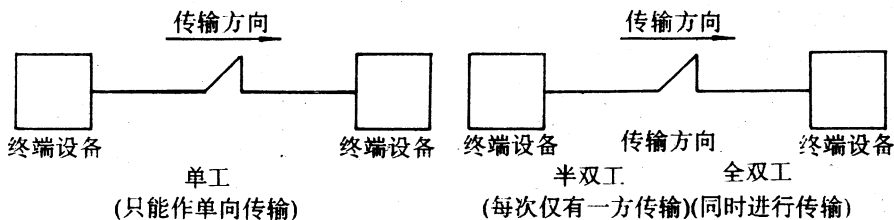
次通信前都需要拨通电话,然后把话音模式转换成数据模式才开始通信。

#### (2)通信方式。

数据只能作单向传输方式称为单工通信;数据可作双向传输,但在同一时间内只能作单向传输的方式称为半双工通信;数据可同时作双向传输的方式称为全双工通信,如图三所示。



图二 异步和同步传输



图三 单工、半双工及全双工传输

### 4. 通信接口标准。

接口一般指两台设备之间的接续条件。美国电子

工业协会发展了一个使用串行二进制方式进行交换的数据电路终接设备和数据终端之间的接口标准,该标

准称为 RS-232 标准。RS-422 和 RS-423 标准也已出现,这些标准提出了在较高传输速度上的要求。但是,标准 RS-232C 仍旧是当前设计调制解调器接口的主要标准。

EIA 标准包括电信号特性,接口的机械特性、接口电路的功能描述和对用于特定通信系统指定群的特定接口电路的标准子集表。另一方面,CCITT 也相应提出了一些建议,有关数据通信接口的是 V. 24 和 V. 28。V. 24 是关于数据终端设备和数据电路终接设备之间的接口电路定义表;V. 28 是有关电路的电气特性。除 V. 24 和 V. 28 之外,CCITT 还新制定了 X 系列接口,X 系列主要适用于公共数据网的室内电路终接设备和数据终端设备之间的接口。

不仅要了解微机数据通信,更重要的是要了解微机数据通信的发展前景。这样才能深刻地理解微机数据通信的功能及重要作用,并能有的放矢地发展微机数据通信技术,使其造福于人类。

目前,在我国以太网、3+网、Token-Ring 环形网和 Novell-NeeWare 网等等都已在开发使用,尤其 Novell-NeeWare 网在北京亚运会上大显身手,为亚运会增添了光彩。

## 二、微机数据通信发展前景。

### 1. 概况。

现在的微机数据通信,主要是通过建立多年的电话交换网来完成的。模拟信号在这些线路上传输,会产生一些频率失真和相位失真,也会引入一些噪音。这种失真和混入的噪音一旦附加在原来的电信号上,就很难消除。传输线路越长,这些失真及噪音累积越大,影响传输质量。另一方面,因线路最高传输速率与线路带宽有关,利用现成的电话网作长距离传输,最高传输速率只可达 16,800 比特/秒。

为了解决上述问题,微机数据通信现在逐渐从模拟通信过渡成数字通信。数字通信和传统的模拟通信不同,在传输线路上传输的不是连续变化的模拟信号,而是离散的脉冲信号,此信号通常是二进编码的脉冲信号。由于接收端需要判别的只是脉冲的有无,所以即使干扰稍大些也能得到高通信质量。另外,电话和图象等模拟信号又可以变成数字信号,因而便于组成统一的综合业务通信网,实现多种信息的自动转换。还有,数字通信设备易于集成化,可靠性高,并且造价较低。

随着光导纤维的发展,微机数据通信传输速率大大提高。目前,世界上许多国家都在大力研究高传输速率和大容量的数字传输系统,这已成为电信技术的发展趋势。

### 2. 脉冲编码调制(PCM)。

在数字通信的发展过程中,主要使用脉冲编码调制来实现模数转换。PCM 基本通过五个阶段:

(1)取样。取样是对连续变化着的输入信号的振幅按一定的时间间隔进行读取,读得的瞬值称为取样值。

(2)量化。量化就是把取样值加以数量化,量化时进行四舍五入,化成整数。

(3)编码。编码是把经过量化的取样值变成数字符号的阶段,变成多少比特的符号是与量化的精度有关。当前,PCM 的实用话频电路中,采用 8 比特的编码( $2^8 = 256$  值量比)b

(4)解码。在接收端把接收的数字信息照原样恢复成取样脉冲序列,被称为解码。

(5)滤波。通过解码得出的取样脉冲序列通过低通滤波器就能把峰脉冲峰值连接起来,得到原来的输入信号。因为在量化过程中部分信号被舍去,所得到的输入信号并不是百分之百原输入信号,不过已能充分满足实际要求。

### 3. 光导纤维。

目前,光导纤维以飞快的速度闯入通信技术的领域,与一般传输媒介如铜线和微波等相比,光纤有下列优点:

(1)频带宽阔——可达数十千兆赫。

(2)抗干扰性强。

(3)在恶劣环境下仍可正常工作——温度可达 1,000℃。

(4)体积小,重量轻——有部分光纤直径只有数微米。

(5)通信保密性好。

(6)传输质量高——误码率可达  $10^{-9}$ 。

(7)价格便宜。

总之,当今已差不多有 750,000 英里的光纤在使用,根据美国电报电话公司 1986 年的统计,已有 150,000 英里的光纤在投入使用。可以预见,不久的将来将从模拟通信系统逐步过渡到数字通信系统,进一步实现通信网络数字化的美好前景即将到来。