

# 计算机应用研究

1990

5

APPLICATION RESEARCH OF COMPUTERS 《计算机应用研究》杂志社





北京四通新技术产业股份有限公司

计算机事业部

## 隆重推出

### ● 万能编程测试卡

无所不能  
开发必备  
十位一体

### ● MICE-II 微机在线仿真器

硬件侦错  
系统调试  
逻辑分析



**MICE-II**

1. EPROM

3. 8751 系列

5. Z86 系列

7. 逻辑电路测试

9. PALS GAL 编程

10. PEEL / PLD 编程

2. EEPROM

4. 8748 系列

6. BPROM(PROM)

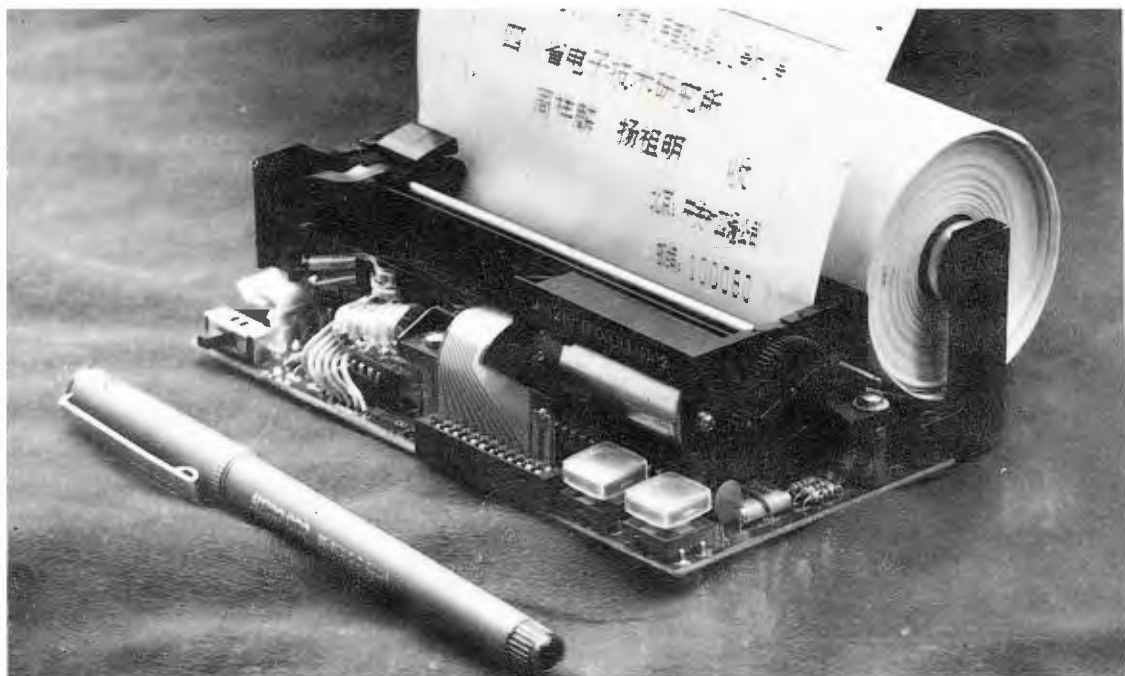
8. RAM 测试

仿真CPU: 8085 8086

8088 Z80 等

是开发调试微电脑的必要  
工具。





## 四川八达电子流量技术研究所 新近推出 MP1221A 中西文智能微型热敏打印机

- \* 采用日本 EPSON 公司微型打印机芯。
- \* 中文、西文、曲线、图表均可打印,切换灵活。
- \* 带有并行通讯接口,可与任何系列的微型机、单板机、单片机联机通讯。
- \* 打印机自带 8031CPU 和包含全部 ASCII 字符及内贮 1837 个常用汉字的硬字库,具备造字功能,随心所欲,简易方便。
- \* 每行正常方式下可打印 32 个汉字;扩展方式下可打印 16 个汉字;西文方式下可打印 40 个 ASCII 字符;3 种字体供君选用。
- \* 噪音低、寿命长、无需色带和墨汁。
- \* 备有可拆卸打印纸支架,可装可卸,得心应手。
- \* 体积小,重量轻,操作简易,耗电省,打印字迹隽永清晰。
- \* 打印机尺寸:宽×高×深=138×38×68mm(不带打印纸)  
宽×高×深=138×65×105mm(带打印纸)

MP1221A 是智能仪器和仪表设计者的好帮手,可广泛应用于各类智能仪器和仪表,为其扩展功能锦上添花! 每部单价:792 元

### CLX-4 车载流量仪

CLX-4 车载流量仪可广泛应用于各种装有蜗轮流量变送器的液体罐车及液体管道,进行流量计量,该仪器采用单片机控制和计算,采用 1 英寸液晶显示屏显示,最大计量可达 9999 升。该流量仪可对不同系数的流量变送器自动修正和精确计量,具有仪器自检功能和系数修正拨盘检测功能,操作简单,可靠性高,防震性好,结构合理,体积小,重量轻,可在恶劣环境条件下准确可靠地计量。 外型尺寸为:210×125×65mm

单 位:四川八达电子流量技术研究所

通讯地址:成都人民南路四段 11 号附 1 号

邮政编码:610015 电话号码:553352

电报挂号:成都 5938

开户银行:成都工商银行跳伞塔分理处

帐 号:67011155

联 系 人:周祥麟 杨祖明

## 《计算机应用研究》杂志办刊单位

四川省电子计算机应用研究中心

贵州省科委计算中心

安徽省计算中心

新疆电子计算中心

吉林省计算中心

青海省测试计算中心

甘肃省计算中心

四川省电子学会

## 《计算机应用研究》杂志编辑委员会

主任委员：张执谦

副主任委员：李泽民

委 员：余 凯 张国栋 贾洪钧

曾光初 王小华 朱景生

## 《计算机应用研究》杂志社董事会

董事长：周赛渝

董 事：唐 珍 郑国基 陆慰椿

秦小竹 龚宇清 黎瑰常

---

计算机应用研究 (双月刊)

(公开发刊)

一九九〇年

第七卷 第五期(总第 37 期)

主 编：张执谦

副 主 编：李泽民

本期责任编辑：张 钢

编 辑 出 版：《计算机应用研究》杂志社

内 文 印 刷：西南冶金地质印刷厂

封 面 印 刷：四川省印刷制版中心

出 版 日 期：1990 年 9 月 30 日

本刊通讯地址：成都市人民南路 4 段 11 号附 1 号

本刊邮政编码：610015

订 阅 处：全国各地邮局

总 发 行：成都市邮政局

---

每册定价：1.30 元

国内统一刊号：CN51-1196

本刊邮发代号：62-68

广告经营许可证：川蓉工商广字 005 号

# 计算机应用研究 第7卷 第5期 (总第37期)

## 目 录

### 软 件 篇

用PC FORTRAN实现的下拉式菜单.....	黄心潮 ( 1 )
Turbo Pascal 4.0下图形驱动程序和字体文件的连接技术 .....	李志伟 ( 3 )
IBM-PC C语言覆盖式程序设计方法.....	徐涛 宋成刚 ( 6 )
“停电宝”——上机过程记录程序.....	杨 宁 ( 8 )
一个用于汉字字幕的图形编辑程序.....	张 昶 ( 10 )
最少操作通用报表系统.....	宽正科 ( 13 )
IBM Token-Ring 网的结构特点.....	殷巍 李雄飞 陈思国 ( 16 )
透明环境的特征.....	陈庆章 ( 18 )
微电脑人体心肌梗塞预测方法.....	张鸿鸣 田卫星 刘翠华 胡庆贤 ( 20 )
计算机在粒度分析数据处理中的应用.....	马晓东 张亚玲 ( 22 )
IBM-PC上的程序流程图编辑/打印软件.....	肖俊良 ( 26 )
PROLOG语言简介.....	李景贵 ( 27 )
数据处理语言的数据流模型.....	叶高英 ( 29 )

### 硬 件 篇

IBM-PC机总线扩展器的设计.....	王砚方 庞文宁 ( 32 )
Mc68488和Intel8048实现 IEEE488接口.....	吴剑勇 ( 36 )
高抗干扰精密V/F或A/D转换芯片与APPLE II微机接口技术.....	杨兆庆 ( 37 )

### 系 统 篇

DRNEST专家系统构造工具的设计与实现.....	杨润生 张小衡 ( 39 )
单片机中断源和中断优先级的扩展.....	张友生 ( 43 )
长途自动交换机微机计费系统.....	郑国基 徐世坤 姚学英 ( 46 )
热水瓶温度检验过程中的自动控制.....	张晓云 王水清 ( 48 )
高速数字信号单片处理机——TMS320C25 硬软件及应用.....	汪亚南 ( 52 )
微机“五防”系统.....	陈骏林 王 涌 李林琪 ( 56 )
甘肃省省级科技项目与科技成果管理系统.....	吕兴寿 吕林霞 雷志军 ( 58 )

### 维 修 篇

IBM-PC/XT及其兼容机系统板维修方法探讨.....	杨剑波 郭永莉 ( 60 )
IBM-PC/XT 微机死机故障维修一例.....	邵发森 ( 63 )
Super-CH显示器扫描电路的代换与维修.....	刘亮生 ( 64 )
2200VS系统死锁故障之一的排除.....	张晓云 ( 66 )
硬盘维护的良好工具—File-Cleaner.....	刘里鹏 ( 67 )

### 信 息 篇

MASP-P电能计量追溯处理系统达到国际先进水平.....	( 19 )
介绍一种高级工具软件.....	( 7 )
信息四则.....	( 5, 15, 31, 45 )

# APPLICATION RESEARCH OF COMPUTERS

Vol.7 No.5 (Total 37)

## CONTENTS

### SOFTWARE

- Realizing Push-down Menu by PC FORTRAN ..... Huang Xinyuan ( 1 )
- Linking Technic between Graph Driving Program and Font Files  
in Turbo Pascal 4.0..... Li Zhiwei ( 3 )
- The Method of Overlaying Programming by C Language in IBM-PC  
..... Xu Tao and Others ( 6 )
- "The Soft UPS" of Computer--Working Record Program  
..... Yang Ning ( 8 )
- A Graph Editor Program for Chinese ..... Zhang Chang ( 10 )
- The General Report System with Minimum Operating  
..... Weng Zhengke ( 13 )
- The Structure and Characteristic of IBM Token-Ring Network  
..... Yin Wei and Others ( 16 )
- The Characteristics of A Transparent Environment  
..... Chen Qingzhang ( 18 )
- The Forecasting Method for Myocardial Infarction of Human Body  
by Microcomputer..... Zhang Hongming and Others ( 20 )
- Applying for Datas Processing of Size Analysis by Computer  
..... Ma Xiaodong and Others ( 22 )
- The Editing/Printing Software of The Program Flow Diagram  
on IBM-PC..... Xiao Junliang ( 26 )
- Brief Introduction for PROLOG Language ..... Li Jinggui ( 27 )
- The Model for Data Stream of Data Processing Language  
..... Ye Gaoyung ( 29 )

### HARDWARE

- Designing for Bus Expander on IBM-PC..... Wang Yanfang and Others ( 32 )
- Realizing of IEEE488 Interface on MC68488 and Intel 80486  
..... Wu Jianyong ( 36 )
- The Interface Techique between APPLE II Microcomputer and A/D  
Conversion Chip..... Yang Zhaoqing ( 37 )

### SYSTEM

- Designing and Realizing for Structural Tools of The DRNEST  
Specialist System..... Yang Runsheng and Others ( 39 )
- Expanding with Interrupt Source and Interrupt Priority on Single  
Chip Processor..... Zhang Yousheng ( 43 )

The Accounting System by Microcomputer on Automatic Switch-board of Long-distance.....	Zheng Guoji and Others ( 46 )
The Automatic Contral for Testing Temperature of Thermos Bottle.....	Zhang Xiaoyun and Others ( 48 )
The High Speed Single Chip processor of Digital Signal—TMS 320C25 Hardware and Software and It's Applying .....	Wang Yanan ( 52 )
The "Five Guard Against System" of Microcomputer .....	Chen Junlin and Others ( 56 )
A Management System for Project and Fruit of Science and Technology.....	Lu Xingshou and Others ( 58 )

## MAINTENANCE

The Maintenancing Method for System Board on IBM-PC/XT and It's Compatible Computers.....	Yang Jianbo and Others ( 60 )
An Example with Maintenancing Deadlock Bug on IBM-PC/XT .....	Shao Fasen ( 63 )
Maintenancing and Replacing for Scan Circuit of SUPER-CH CRT .....	Liu Liangsheng ( 64 )
Eliminating for One of The System Deadlocks on 2200 VS .....	Zhang Xiaoyun ( 66 )
File-cleaner—The Good Tool of Uphold Hard-disk.....	Liu Lipeng ( 67 )

## INFORMATION

Six Pieces of News.....	(The Cover 5, 7, 15, 19, 31, 45)
-------------------------	----------------------------------

---

# APPLICATION RESEARCH OF COMPUTERS

(Publicly Publishing) (Bimonthly)

September 1990 Volume 7 Number 5 (Total 37)

Main Editor: Zhang Zhi Qian, Assistant Main Editor: Li Ze min  
Responsibility Editor for this Issue: Zhang Gang

Editor/Publfsher: Editorial Department of This Periodical

Address: 11—1 Four Section Ren Ming Nan Lu, Chengdu, P.R.C  
Press: South West metallurgical geology Printing house of  
Ministry of Motallurgical Industry

Subscribe Address: The Post of the All Localities of the Country  
Central Dispatching in Inside the Country: The Post of Chengdu

---

Unified Number in Inside the Country: CN 51—1196

Number of the Post Publishing: 62—68 Post Code: 610015

# 用PC FORTRAN实现的下拉式菜单

北京理工大学三系 黄心渊

**摘要:** 本文首先阐述了下拉式菜单的优点,而后介绍了几个可供FORTRAN调用的绘图子程序。在此基础上给出了一个在IBM PC/XT,AT及其兼容机上用FORTRAN语言实现的下拉式菜单程序。

## 一、PC机常用菜单的形式

PC机常用的菜单有三种。一是使用数字或选择项的第一个字母来选取选择项,我们称这种菜单为老式菜单;二是移动屏幕上的亮光标选取选择项,这种菜单只有一层,无子菜单,我们称之为突现式菜单;三就是所谓的下拉式菜单,它选取选择项的方式与突现式相同,只是它有子菜单。实际上,突现式菜单是下拉式菜单的一个子集。一个屏幕上允许数个下拉式菜单同时作用。

下拉式菜单与老式菜单相比,有以下两个优点:

1. 色彩丰富,鲜艳;
2. 用户不需键入所选的选择项,而可移动亮光标进行选择,很直观。

## 二、介绍几个有关的FORTRAN绘图子程序

FORTRAN语言没有屏幕绘图功能,但它可以同C和汇编语言连接。因此,我们可用这两种语言编写几个简单的绘图子程序,供FORTRAN调用。下面是实现下拉式菜单用到的几个子程序。

### 1. GRFMOD

功能:使屏幕进入图形状态,清除屏幕,字符光标移到屏幕左上角。

图形状态下,屏幕呈兰色背景,屏幕坐标系原点(0,0)在屏幕左下角,横向坐标范围为[0,319],纵向坐标范围为[0,227],字符为25行40列。

### 2. ERSCRN

功能:清除屏幕。

### 3. LOCRC (IX, IY, N)

功能:将字符光标移至指定坐标点并擦除光标后面(含光标)指定个数的字符。

输入参数:

IX, IY——要移向的字符光标坐标值,

类型INTEGER\*2

( $0 \leq IX \leq 24, 0 \leq IY \leq 39$ )

行由下往上,列由左向右。

N——从(IX, IY)开始要擦除的字符个数,类型INTEGER\*2。

### 4. GOTOXY (X, Y, ICLR)

功能:由当前内部图形坐标暗走至指定点的坐标并为后继线段指定颜色。

输入参数:

X, Y——目标点坐标值。类型REAL\*4

ICLR——后继背景颜色代号,类型INTEGER\*2。

ICLR取值为-1至6,其对应的颜色为与背景同色、无色、青兰色、品红色、白色、绿色、红色和棕色。

### 5. DRAWLN (X, Y)

功能:从当前内部图形坐标画直线至指定坐标点。

输入参数:

X, Y——目标点坐标值,类型REAL\*4。

### 6. CURSOR (IX, IY CH)

功能:在指定位置显示屏幕叉丝,返回经操作者调整并确认的叉丝坐标值与确认时键入的字符。

输入参数:

IX, IY——要求叉丝出现的屏幕位置的坐标值。类型INTEGER\*2



返回参数:

IX, IY——叉丝当前的坐标值。类型 INTEGER \*2.

CH——用户键入的字符。类型 CHARACTER \*2.

利用F1, F2, F3, F4键可以分别将叉丝移动的步长调整为1, 10, 50, 100个屏幕单位; 利用↑↓←→及对角线上的其它四个键, 可以分别使叉丝在上、下、左、右及四个斜角方向移动一个步长。

### 三、菜单的实现

下拉式菜单的实现需如下几个步骤:

#### 1. 确定显示区域。

在屏幕上选择一个区域供显示菜单用, 同时也确定下层菜单的显示域。本文实例菜单区为从33列到39列, 下拉式菜单子层覆盖上层。

#### 2. 确定菜单的内容

首先确定主菜单的内容, 将其写在指定区域。本文实例主菜单共有5项, 在屏幕的第33列到39列, 从第5行开始, 隔行写一, 而后用方框将各项框起来, 做为选择的依据。为简单起见, 实例只在主菜单的CHAN项设置了子菜单, 子菜单也有5项, 进入子菜单后完全覆盖掉主菜单。该子菜单的第一项为返回其上层菜单。

#### 3. 选择菜单

我们可用CURSOR子程序提供的移动光标功能, 将光标移到我们选定的菜单后, 按回车键, 菜单程序按返回的坐标位置调用相应的下层菜单或子程序。若返回值说明光标点不在菜单区, 则给出信息 "Invalid meau"。

选定菜后, 菜单程序自动调用画矩形框子程序, 以另一前景颜色画框, 说明该项菜单的内容已被选定。退出该项菜单后, 相应的框自动返回原来的颜色。

### 四、实例

实例程序如下:

```

C      菜单程序
SUBROUTINE MENU
COMMON /GG/IM,LDY(15),RUY(15)
CHARACTER CH*1
REAL LDY

C      进入图形环境并写菜单
CALL GRFMOD
40  CALL LOCRC(3,32,7)
22  WRITE(*,'(a\)' )' MENU'
    CALL WRITEC(5,'DRAW')
    CALL WRITEC(7,'DREA')
    CALL WRITEC(9,'DLIN')
    CALL WRITEC(11,'CHAN')
    CALL WRITEC(13,'QUIT')

C      移动光标点菜单
10  CALL CURSOR(IX,IY,CH)
    X=FLOAT(IX)
    Y=FLOAT(IY)
    CALL LOCRC(23,0,25)
    IF (X.LT.219.OR.X.GT.319.OR.Y.LT.LDY(13)) THEN
    CALL LOCRC(23,0,25)
    WRITE(*,'(a\)' )' Invalid menu'
    GOTO 10
    ELSE IF((Y.LT.RUY(5)).AND.(Y.GT.LDY(5))) THEN
    CALL DRAW
    ELSE IF((Y.LT.RUY(7)).AND.(Y.GT.LDY(7))) THEN
    CALL DREA
    ELSE IF((Y.LT.RUY(9)).AND.(Y.GT.LDY(9))) THEN
    CALL DELIN
    ELSE IF((Y.LT.RUY(11)).AND.(Y.GT.LDY(11))) THEN
    CALL CHAN
    ELSE IF((Y.LT.RUY(13)).AND.(Y.GT.LDY(13))) THEN
    GOTO 50
    ENDIF
    GO TO 40
    RETURN
    END

C      子菜单程序CHAN
SUBROUTINE CHAN
COMMON /GG/IM,LDY(15),RUY(15)
CHARACTER CH*1
REAL LDY

C      写子菜单
40  CALL LOCRC(3,31,8)
    WRITE(*,'(a\)' )' SMENU'
    CALL WRITEC(5,'TQQI')
    CALL WRITEC(7,'ROUT')
    CALL WRITEC(9,'COPY')
    CALL WRITEC(11,'FCOL')
    CALL WRITEC(13,'BCOL')

C      移动光标点菜单
10  CALL CURSOR(IX,IY,CH)
    X=FLOAT(IX)
    Y=FLOAT(IY)
    IF (X.LT.219.OR.X.GT.319.OR.Y.LT.LDY(13)) THEN
    CALL LOCRC(23,0,25)
    WRITE(*,'(a\)' )' Invalid menu'
    GO TO 10
    ELSE IF((Y.LT.RUY(5)).AND.(Y.GT.LDY(5))) THEN
    C      改变选定菜单项矩形框的前景色
    CALL SS(5,6)

C      RETURN
    ELSE IF((Y.LT.RUY(7)).AND.(Y.GT.LDY(7))) THEN
    CALL SS(7,6)
    CALL ROUT
    ELSE IF((Y.LT.RUY(9)).AND.(Y.GT.LDY(9))) THEN
    CALL SS(9,6)
    CALL COPY
    ELSE IF((Y.LT.RUY(11)).AND.(Y.GT.LDY(11))) THEN
    CALL SS(11,6)
    CALL FLCO
    ELSE IF((Y.LT.RUY(13)).AND.(Y.GT.LDY(13))) THEN
    CALL SS(13,6)
    CALL BLGO
    ENDIF
    C      退出选择后,恢复其矩形框前景色
    CALL SS(IM,4)

C      GO TO 10
    END

```

# Turbo Pascal 4.0下图形驱动程序和字体文件的连接技术

空军第一航空技术专科学校计算机教研室 李志伟

## 一、引言

Turbo Pascal 4.0 是一种高度集成化系统软件,它集编辑、编译、连接、排错和窗口环境于一体,提供了一个友善的、多窗口的软件开发环境,因此它一经推出,便风靡全球,为广大PC机用户所欢迎,成为最出类拔萃的高级程序设计语言之一。

Turbo Pascal 4.0 同 3.0 相比,有大幅度的改进,除增加了一些新特色外,还增加了一些新的功能,本文主要介绍Turbo Pascal 4.0 中提供的图形驱动程序和字体文件,以及将这些文件连接到内存和.EXE文件中的方法,利用这些方法,可大大提高程序的开发效率,且方便用户。

## 二、Turbo Pascal 4.0的图形驱动程序和字体文件

Turbo Pascal 4.0 在图形处理方面继承了3.0的优点,同时又提供了新的图形系统 BGI——Borland 图形接口,可以实现更加丰富的图形功能。它能支持多种图形卡,能在系统下画线、圆弧、圆、平面多边形,以及其他形状的图形,并且支持许多填充和线型。对于每种图形卡

都有相应的图形驱动程序 (\*BGI) :

CGA.BGI	——支持CGA、MCGA 图形适配器
EGAVGA.BGI	——支持EGA、VGA图 形适配器
HERC.BGI	——支持Hercules 单色 图形适配器
ATT.BGI	——支持AT&T 6300图 形适配器
PC3270.BGI	——支持3270 PC图形适 配器

此外, BGI 图形系统还包含一个  $8 \times 8$  点阵字体和几种笔划字体,系统将这些字体作为图形字符输出正文,用户可以根据需要将这些字体放大或缩小,也可以将其对齐位置,水平或垂直定位等。

使用点阵字体,当放大因子很大时,字符的分辨率就不高了(对于一般的小字符,点阵字体就足够了),而使用笔划式字体,当画大字符时,因其字体是由向量定义的,所以当字体放大时仍能保持高分辨率 and 高质量。

缺省的  $8 \times 8$  点阵字体内嵌在GRAPH.TPU

```

C 画矩形框子程序
SUBROUTINE SS(I,J)
COMMON /GG/IM,LDY(15),RUY(15)
REAL LDY
LDY(I)=227.*(1.-(I+1.)/24.)
RUY(I)=227.*(1.-(I-1.)/24.)
CALL GOTOXY(240.,RUY(I),J)
CALL DRWLIN(319.,RUY(I))
CALL DRWLIN(319.,LDY(I))
CALL DRWLIN(240.,LDY(I))
CALL DRWLIN(240.,RUY(I))
IF ((J.EQ.5).OR.(J.EQ.6)) IM=1
RETURN
END
C 写菜单子程序
SUBROUTINE WRITEC(II,A)
CHARACTER A*4
CALL LOCRC(II,33,6)
WRITE(*,'(A)') A
CALL SS(II,4)
RETURN
END
  
```

用实例程序产生的菜单如下:

MENU	SMENU
DRAW	TQQI
DREA	ROUT
DLIN	COPY
CHAN	FCOL
QUIT	BCOL

## 五、结束语

本文所给出的实例,是笔者研制的绘图系统菜单的一个子集。用户对该系统的菜单设计表示满意。当然,由于水平所限,本文难免有许多缺点,希望能得到广大同行的批评和帮助。

单元中,所以运行时总可以用,而笔划式字体则分别存放在.CHR文件中:

TRIP.CHR——三倍体字体字符集

LITT.CHR——小字体字符集

SANS.CHR——无衬线字体字符集

GOTH.CHR——哥特字体字符集

由于这些图形驱动程序(\*BGI)和字体文件(\*CHR)分别保存在磁盘的不同文件中,所以用户可以在运行时将其装入内存——当程序检测具体的硬件时,也可以将其嵌在程序中,直接连入.EXE文件。

### 三、将.BGI和.CHR连接到内存

#### (一) 连接方法

一个程序中,有时要用到某一种或某几种图形驱动程序,每当用 InitGraph 对图形系统初始化时,都要调用相应的磁盘文件,这将降低程序的开发效率,为此,可在程序的开始部分将所用到的图形驱动程序预先装入堆空间中,并可通过将堆的内存地址传递给函数 RegisterBGI driver来“寄存”驱动程序,这样,程序运行时将使用寄存的驱动程序,而不再由GRAPH.TPU从磁盘装入。

同样,如果程序中用到某一种或某几种笔划字体,每用 SetTextStyle 设置一次字体,就要从磁盘上调用一次相应的字体文件,为避免此麻烦,也可预先将这些字体文件装入到堆空间中,然后将相应的堆地址传递给函数 Register BGI font 来“寄存”字体,使得程序可直接使用堆中的字体,而不需从磁盘装入。

#### (二) 实例

```
program LoadMemory;
uses Graph;
var
  Driver, Mode : integer;
  DriverF,FontF: file;
  DriverP,FontP: pointer;
begin
  Assign(DriverF, 'CGA.BGI');
  Reset(DriverF, 1);
  GetMem(DriverP, FileSize(DriverF));
  BlockRead(DriverF, DriverP, FileSize(DriverF));
  If RegisterBGIDriver(DriverP) < 0 then
  begin
    WriteLn('Error registering driver: ',
            GraphErrorMsg(GraphResult));
    Halt(1);
  end;
  Assign(FontF, 'TRIP.CHR');
```

```
Reset(FontF, 1);
GetMem(FontP, FileSize(FontF));
BlockRead(FontF, FontP, FileSize(FontF));
If RegisterBGIfont(FontP) < 0 then
begin
  WriteLn('Error registering font: ',
          GraphErrorMsg(GraphResult));
  Halt(1);
end;

ReadLn;
Driver := CGA;
Mode := CGAHI;
InitGraph(Driver, Mode, '');
If GraphResult < 0 then Halt(1);
SetTextStyle(TriplexFont, HorizDir, 4);
OutText('CGA Driver and Triplex font have loaded');
MoveTo(0, TextHeight('a'));
ReadLn;
SetTextStyle(SansSerifFont, HorizDir, 4);
OutText('Your disk should be spinning...');
MoveTo(0, GetY + TextHeight('a'));

ReadLn;
SetTextStyle(TriplexFont, HorizDir, 4);
OutText('Back to Triplex');
end.
```

在Turbo Pascal 系统下运行该程序,并注意观察驱动器指示。开始时驱动器旋转,将磁盘文件CGA.BGI和TRIP.CHR装入内存的堆中,然后按〈Enter〉键,程序执行第二部分,对图形系统进行初始化并设置三倍体字体,将发现驱动器不转,因此时CGA.BGI和TRIP.CHR已装入内存,再按〈Enter〉,程序调用无衬线字体,因该字体不在内存中,要从磁盘装入,故发现驱动器旋转。最后,程序又用到三倍体字体,因该字体已在内存,故再按〈Enter〉键,将发现驱动器不再旋转。

如果在程序中用到多种图形驱动程序和笔划字体,则可在程序的开始部分将相应的文件都装入堆中,这样在每次调用时就不必再由磁盘装入。

### 四、将.BGI和.CHR连接到.EXE文件中

因为Turbo Pascal 4.0中的BGI图形驱动程序和字体分别保存在不同的磁盘文件中,因此每次运行时都要将其动态地装入,尽管用上述方法可将其装入内存,但在每次运行开始时仍要调用相应的磁盘文件,即使运行.EXE文件,也需要这些文件驱动。在有些场合,为了使编制的软件更加集成化和更方便用户,最好将这些文件直接嵌入.EXE文件中,下面介绍这种连接技术。

## (一) 用BINOBJ.EXE程序产生.OBJ文件

BINOBJ.EXE 程序用于将二进制文件转换为.OBJ文件, 它有三个参数:

BINOBJ <源程序[.BIN]> <目标程序[.OBJ]> <公共名>

<源程序>是要转换的二进制文件, <目标程序>是要产生的.OBJ文件, <公共名>是要在 Turbo Pascal 程序中说明的过程名. 利用此程序可将.BGI和.CHR文件转换为.OBJ文件.

例如将CGA.BGI文件转换为CGA.OBJ, 并用CgaDriverProc作为外部过程名在 Turbo Pascal程序中说明, 则在DOS下用下列命令即可转换:

```
binobj Cga.bgi Cga CgaDriverProc
```

这里 Cga 缺省了扩展名 OBJ, 执行时 BINOBJ自动将其加上.

同理, 也可将TRIP.CHR和LITT.CHR等字体文件转换为相应的.OBJ文件:

```
binobj trip.chr trip TriplexFontProc
binobj litt.chr litt SmallFontProc
```

## (二) 编制DRIVERS和 FONTS 单元文件

Turbo Pascal 4.0 提供了单元的功能, 各种 Turbo Pascal 程序可以单元的形式连接起来, 根据这个思想, 可以编制单元, 通过单元建立.OBJ文件与主程序的联系.

若一个程序中用到了CGA.BGI和TRIP.CHR、LITT.CHR文件, 则相应的单元文件分别为:

```
unit Drivers;
interface
procedure CgaDriverProc;
implementation
procedure CgaDriverProc; external;
{$L CGA.OBJ}
end.
```

```
unit Fonts;
interface
procedure TriplexFontProc;
procedure SmallFontProc;
implementation
```

```
procedure TriplexFontProc; external;
{$L TRIP.OBJ}
procedure SmallFontProc; external;
{$L LITT.OBJ}
end.
```

## (三) 将DRIVERS和 FONTS单元连接到 .EXE文件中

一个程序在内存中调试通过后, 将程序开始部分的uses语句中加入Drivers和Fonts两个单元名, 然后在 Turbo Pascal 编译菜单下选择D、C命令, 当程序编译到uses语句时, 编译系统将自动调用Drives.pas和Fonts.pas, 将其编译成相应的.TPU单元, 为uses语句所调用, 这样, 最后产生的.EXE磁盘文件中便已嵌入了.BGI和.CHR文件, 此时的.EXE文件便可单独在DOS下运行, 而不需其它程序的支持.

## (四) 说明

如果程序中用到了其它的.BGI和.CHR文件, 自己也可以按照上述方法产生相应的.OBJ文件, 修改Drivers.pas和Fonts.pas文件, 在集成环境的编译菜单下选择Make命令(或在命令行中加/M选择), 则系统重新编译主程序, 并重新编译Drivers.pas和Fonts.pas单元文件, 产生新的.TPU文件, 为主程序调用, 这样就将新的.BGI和.CHR文件连入到了.EXE文件中.

另外, 自己也可以编制MAKE文件(.Mak), 然后使用MAKE实用程序重新编译主程序, 进而产生.EXE文件.

用这种方法产生的.EXE文件使用非常方便, 不需要.BGI和.CHR文件的支持(它们已嵌入到了程序中)便可单独在DOS下运行, 但用这种方法产生的执行代码比用其它方法产生的执行代码大.

## 最新防病毒软件

SCAN V62系列最新防病毒软件可查解目前国际上发现的所有86种病毒, 需要者可同长春市上海路21—3号201软件信箱李柏或本刊联系.



## IBM-PC C语言覆盖式程序设计方法

攀钢自动化部微机室 徐涛 宋成刚

在用C语言进行应用程序的开发时,往往应用程序需占用很大的内存空间,而我们都希望能在内存空间较小的IBM PC机上运行。这里我们介绍一下用C语言编写程序怎样实现覆盖式程序设计,着重讲述了进程间参数传递方法。

当运行大程序时,如果将整个程序连接,一旦装入运行,将会因内存空间大小的限制而使连接失败,使用C语言覆盖式程序设计方法,可以使程序在运行时分块装入,其它模块驻留在外存上,只有当需要它们时才被调入内存中,就是说覆盖技术可以使程序的几个模块共享同一部分内存空间。

C语言覆盖式程序设计主要是通过对进程的控制来实现的。C语言程序库中提供了两族进程控制例程:exec族例程和spawn族例程。它们各有八个例程。在程序中通过选用不同的例程,可以实现进程调用、传递参数给子进程以及改变或继承父进程的环境。达到覆盖式程序设计的目的。

下面,我们举例说明怎样在PC机上使用C V4.00版的覆盖技术。

```
father.C
#include "stdio.h"
#include "process.h"
main (arg c, argv)
    int arg c;
    Char *argv[];
{
    argc=5;
    argv[5]=NULL;
    spawnv (P-OVERLAY,
        "Child.exe" argv);
}

Child.C
#include "stdio.h"
#include "process.h"
main (argc, argv)
    int argc;
    Char * argv[];
{
    argc=5;
    argv[5]=NULL;
    ...
}
```

在这里, father 作为父进程, child 作为子进程。我们在MS-DOS提示符下键入 "father one two 34" 就能看到父进程father调用了子进程 child, 同时将参数传给了子进程 child。

在father中, 整型变量argc是需要传递参数的最大数目。由于待传递参数预先没有确定, 我们可以在程序中将argc的值赋大些, 然后在MS-DOS操作系统提示符下键入命令 "father one two 3 4", 这样被执行程序就带有4个参数one, two, 3, 4 其中one存入argv[1], two存入argv[2], 3存入argv[3], 4存入argv[4]中。

在子进程child中, 也进行同样的说明, 那么父进程father通过例程

```
spawnv (P-OVERLAY, "child.exe", argv)
```

就能调用子进程 child, 并将参数one, two, 3, 4传入子进程。

在father.C中, 我们看到使用了spawn族例程中的 spawnv 例程, 该例程的作用是实现在参数传递和覆盖式调用。

它的格式为

```
spawnv (modeflag, pathname,
        argv)
```

其中modeflag为调用方式。

spawn族例程中modeflag的值在process.h文件中定义为

P\_WAIT 暂停父进程, 直到子进程执行完毕。

P-OVERLAY 用子进程覆盖父进程, 毁掉父进程。

P-NOWAIT 继续父进程, 子进程也并行执行 (C语言V 4.00 版中还未用)。

路径名参数 pathname 给定的条件被当作子进程来执行。路径名可以是一个完全路径 (从

根目录开始), 或者一个不完全路径(从当前工作目录开始), 或者仅由一个文件名组成。如果文件名的扩展名缺省, 则先查找不带扩展名的文件, 若没有找到, 则假设文件扩展名为 EXE, 再查找。

**argv**是用来作为参数传递的, 传给子进程的参数是由作为参数的一个或多个字符串指针指出, 这些字符串组成子进程的参数表, 其长度不大于128字节。

下面列出**exec**族例程和**spawn**族例程供读者参考。

#### exec族例程

```
execl (pathname, arg0, arg1, ...,
      argN, NULL)
execle (pathname, arg0, arg1,
      ..., argN, NULL, envp)
execlp (pathname, arg0, arg1,
      ..., argN, NULL)
execlpe (pathname, arg0, arg1,
      argN, NULL, envp)
execv (pathname, argv)
execve (pathname, argv, envp)
execvp (pathname, argv)
execvpe (pathname, argv, envp)
spawn族例程
spawnl (modeflag, pathname,
      arg0, arg1, ..., argN,
      NULL)
```

```
spawnle (modeflag, pathname,
      arg0, arg1, ..., argN,
      NULL, envp)
spawnlp (modeflag, pathname,
      arg0, arg1, ..., argN,
      NULL)
spawnlpe (modeflag, pathname,
      arg0, arg1, ..., argN,
      NULL, envp)
spawnv (modeflag, pathname,
      argv)
spawnve (modeflag, pathname,
      argv, envp)
spawnvp (modeflag, pathname,
      argv)
spawnvnp (modeflag, pathname,
      argv, envp)
```

其中, **arg0, arg1, ..., argN** 是参数传递的参数表, **argv** 是变量数组, 在例程中使用参数表是采用分离的参数传递, 使用变量数组是采用指针传递参数。参数**envp** 指定一个新的环境变量来改变子进程的环境, 它是字符串指针数组, 其中每一个元素(除了最末一个元素外)指向一个以串结束符结尾的, 定义一个环境变量的字符串。数组**envp**最末一个元素应为NULL, 当**envp**本身为NULL时, 子进程继承父进程的环境。

## 介绍一种高级工具软件

长春市201信箱 东玉华

**Professional Master Key (PMK)** 是一类类似于著名软件**Norton Utilities**的高级工具软件, 但更加易学易用, 适配于各种软盘、硬盘的 IBM 微机及其兼容机。

软件由六个文件组成, 使用时, 在 DOS 提示符下输入 **PMK-MENU** 即可进入主菜单。此时屏幕上共有四个窗口: 左上角是操作提示; 右上是待选用的程序名, 用四个光标键即可选择; 中间是命令行, 按一下空格键, 即可输入相应格式的命令及其参数; 下面的大窗口介绍程序的功能及用法。

选用 **PMK** 程序时, 下面大窗口的上部介绍它的主要功能。包括检查和修改磁盘或文件的任一扇区, 搜索各种字符串, 恢复被删除的文件, 修改文件的属性, 显示磁盘上的文件位置及空间等。

窗口下部是命令的使用格式、参数及其意义。

输入 **PMK** 即进入 **PMK** 菜单。此菜单中有 11 项选择, 在各项选择中, 还有进一步的操作提示。如选择 (3) 或 (5) 项修改扇区时, 按 **F1** 键会出现一帮助窗口, 列出从 **F1** 到 **F10** 键的功

能及 **Home**、**End**、**Ins** 和 **Del** 等键的妙用, 使你的操作更加方便。

主菜单中其它 4 项是 **PMK** 的辅助工具程序。其中 **Fill Disk** 可将磁盘中所有空白扇区中原有数据彻底涂掉, 填上 (也可不填) 用户选用的信息, 从而防止遗留在磁盘上的旧信息被 **L** 恢复, 造成失密。 **UnFormat** 可用于恢复被意外格式化的硬盘, 防止资料丢失。使用时, 应将此文件装到硬盘中, 并在 **AUTOEXEC.BAT** 文件中加上下面这一行:

**UF/SAVE C:**

这样每次开机时, **UF** 可将 **FAT** 中的信息贮存到一个不会被格式化指令破坏的地方。当发生意外格式化时, 用 **PMK** 盘在 **A** 驱动器中启动机器, 输入 **UF C:** 即可恢复 **C** 盘中的文件。

**ZeroDisk** 可永久性的去除磁盘中所有的信息, 使其不能被恢复。

**ZeroFile** 可永久性地删除文件中的数据, 使其不能被恢复。这一功能也可用 **PMK** 中的删除功能加 **FD** 指令来实现。函索方法见上期封 2。

## “停电宝”——上机过程记录程序

浙江大学 杨宁

**摘要:** 这是一个常驻内存的中断处理程序, 用以记录使用微机时从键盘打入的内容, 避免由于突然停电而来不及存盘所造成的损失。

程序编制人员在上机时最担心的莫过于当程序输入快要结束时还未来得及存盘却突然停电, 或者由于其他原因辛辛苦苦输入的程序或数据未能保存下来, 于是第二次上机又得从头输入程序或数据。使用过VAX机的人大概都知道, 它能保存编辑过程, 以便突然停电后也能把输入的程序找回来。那么在微机上能否实现这样的功能呢? 回答是肯定的, 这就是下面要介绍的“停电宝”——上机过程记录程序。

“停电宝”程序(以下简称TDB)的原理是, 通过截取键盘中断程序INT16, 把上机者打入的每一个字母或数字放在一个缓冲区中, 待达到预定的字符计数时, 自动把缓冲区中的内容存到一个数据文件(系统管理子目录: 硬盘上的SYSTEM.MNG中的TDB.DAT)中, 因此即使突然停电, 也可以在那个数据文件TDB.DAT中找到你输入的还没存盘的程序或数据。你可以使用任何一种编辑程序(EDLIN, WS等)编辑TDB.DAT, 取出你所需要的内容, 然后把TDB.DAT删除。

TDB是你上机过程的忠实记录, 为了避免它保存那些没有必要保存的上机过程, 程序中设置了一个软开关Alt-D组合键(当然, 你也可以规定另一个组合键作为软开关), 当运行过一次TDB后, 按一次Alt-D组合键则“停电宝”失效, 不再把你的上机过程存盘, 再按一次Alt-D组合键则“停电宝”又生效, 因此, Alt-D组合键就象是“停电宝”的开关。那么如何才能知道TDB的开关是开还是关呢? 当你第一次运行TDB时, 开关总是开的, 即TDB是有效的, 而之后运行TDB时, 程序就会告诉你TDB已经运行过, 然后告诉你现在TDB的开关状态。

TDB是硬件中断INT16的中断截取程序, 在运行后常驻内存, 因此不但能记录在DOS提示符下打入的命令, 而且能记录绝大多数编辑软件

的编辑过程, 这样就免除了上机操作者输入程序的后顾之忧。而且, TDB对原来的系统没有影响, 全部保留原有功能。在任何时候, 根据你的需要, 按一下Alt-D就可以打开或关闭TDB的功能。如果在DOS提示符下, 打入TDB运行它, 则程序不但在屏幕上告诉你开关的状态, 而且通过微机的扬声器告诉你TDB是开还是关: 开时扬声器响两声, 关时扬声器响三声。

TDB程序是用DEBUG小汇编写成的, 为了便于修改或扩充功能, 我用编辑程序编写了TDB.ASM(见程序清单), 然后利用DOS的输入输出重定向功能:

```
C>DEBUG<TDB.ASM
```

生成TDB.COM文件(仅1280字节)。使用TDB时, 先要建立子目录C:\SYSTEM.MNG(若C盘原来就有该子目录就不需做此步), 然后运行TDB.COM:

```
C>MD C:\SYSTEM.MNG
```

```
C>TDB.COM
```

之后, TDB就在为你服务了。

TDB.ASM程序由很多小段程序组成, 各小段之间留有空区, 可用以增加功能。下面简介部分单元的用途, CS:0120→0123是TDB是否运行过的标志, CS:012A是生效标志, 为1则生效, 为0则失效。CS:0154→0155是预定存盘字符数, 高位在后, 程序清单中是0100H即256个字符, 当你在键盘上打入256个字符时即自动存盘一次。CS:0160→CS:026F是输入缓冲区。程序中CMP AH, 20 JZ 0380两句是判断Alt-D开关的, 因为组合键Alt-D的扩展ASCII码是20H, 如果你想改用Alt-T作开关, 只要把20改为14即可, 其余类推。如果修改了TDB.ASM, 只要再运行一次DEBUG:

```
C>DEBUG<TDB.ASM
```

就可生成TDB.COM, 不需要运行连接程序

LINK. 注意, TDB.ASM 中最后必须有 Q 命令, 否则, 输入重定向功能失败。

最后, 附 TDB.ASM 程序如下, 在 PC AT 和 XT 机上通过。

## 参 考 文 献

1. 张明: 《计算机世界月刊》, 1989.8
2. 《PC/AT Technical Reference》  
IBM corp.

C>TYPE TDB.ASM

A 100  
JMP 04A0

A 105  
DB 'TDB.COM N.L.Y 1990.5.5'

A 120  
DB 'TDBR'

A 126  
DB 0,0,0,0  
DB 1  
DB 0,0,0,0

A 130  
DB "C:\SYSTEM\MNG\TDB.DAT", 0, 0

A 150  
DB 0,0  
DB 0,0  
DB 00,01

A 160  
DB 0

A 270  
DB 0,0  
DB 0,0  
DB 0,0  
DB 0,0  
DB 0,0  
DB 0,0

A 0290  
CLI  
CS:  
MOV [0270], AX  
ES:  
MOV [0272], DS  
CS:  
MOV [0274], ES  
PUSH CS  
POP DS  
CMP AH, 00  
JNZ 02B0  
JMP 02F0

A 02B0  
MOV ES, [0274]  
MOV DS, [0272]  
CS:  
JMP FAR [0126]

A 02F0  
PUSHF  
CALL FAR [0126]  
CLI  
PUSH CS  
POP DS  
MOV [0270], AX  
MOV SI, 0160  
MOV DX, [012B]  
MOV BYTE PTR [SI+BX], AL  
CMP AL, 0  
JNZ 0320  
JMP 0350

A 0320  
CMP AL, 0D  
JZ 0330  
JMP 03C0

A 0330  
INC WORD PTR [012B]  
MOV BX, [012B]  
MOV BYTE PTR [SI+BX], 0A  
JMP 03C0

A 0350  
MOV AX, [0270]  
INC WORD PTR [012B]  
MOV BX, [012B]  
MOV BYTE PTR [SI+BX], AH  
CMP AH, 20  
JZ 0380  
JMP 03C0

A 0380  
MOV AH, [012A]  
CMP AH, 00  
JNZ 03A0  
MOV BYTE PTR [012A], 1  
JMP 03C0

A 03A0  
MOV BYTE PTR [012A], 0  
JMP 03C0

A 03C0  
INC WORD PTR [012B]  
MOV AX, [012B]  
CMP AX, [0154]  
JGE 03E0  
JMP 0480

A 03E0  
PUSH DS  
PUSH DX  
PUSH BX  
PUSH CS  
POP DS  
MOV AH, [012A]  
CMP AH, 0  
JZ 0460  
STI  
NOF  
CLI  
MOV AX, 3D02  
MOV DX, 0130  
INT 21  
CLI  
PUSH CS  
POP DS  
JNB 0420  
CMP AX, 0002  
JNZ 0460  
MOV AH, 3C  
MOV DX, 0130  
MOV CX, 0020  
INT 21  
CLI  
JMP 0420

A 0420  
CS:  
MOV [0150], AX  
MOV BX, AX  
MOV AX, 4202  
XOR CX, CX  
MOV DX, CX  
INT 21  
CLI  
PUSH CS  
POP DS  
MOV AH, 40  
MOV BX, [0150]  
MOV CX, [012B]  
MOV DX, 0160  
INT 21  
CLI  
PUSH CS  
POP DS  
MOV AH, 3E  
MOV BX, [0150]  
INT 21  
CLI  
JMP 0460

A 0460  
PUSH CS  
POP DS  
MOV WORD PTR [012B], 0000  
INC WORD PTR [0152]  
POP BX  
POP DX  
POP DS  
JMP 0480

A 0480  
PUSH CS  
POP DS  
MOV ES, [0274]  
MOV AX, [0270]  
CS:  
MOV DS, [0272]  
IRET

A 04A0  
PUSH CS  
POP DS  
MOV AX, 3516  
INT 21  
MOV WORD PTR [0126], BX  
MOV WORD PTR [0128], ES  
CMP BX, 0290  
JNZ 0500  
CLD  
PUSH CS  
POP DS  
MOV SI, 0120  
MOV DI, 0120-  
LODSW  
ES:  
CMP AX, [DI]  
JNZ 0500  
LODSW

ES:  
CMP AX, [DI+02]  
JNZ 0500  
MOV AH, 09  
MOV DX, 0500  
INT 21  
MOV DX, 05A0  
ES:  
CMP BYTE PTR [012A], 0  
JZ 04F0  
MOV DX, 05D0  
JMP 04F0

A 04F0  
PUSH CS  
POP DS  
MOV AH, 09  
INT 21  
MOV AX, 4C00  
INT 21

A 0500  
PUSH CS  
POP DS  
MOV DX, 0290  
MOV AX, 2516  
INT 21  
PUSH CS  
POP DS  
MOV AH, 09  
MOV DX, 0530  
INT 21  
PUSH CS  
POP DS  
MOV BYTE PTR [012A], 1  
MOV DX, 0600  
MOV AX, 3100  
INT 21

A 0530  
DB 0D, 0A, ' " Ting Dian Bao " "  
DB 'has been installed !'  
DB 0D, 0A, ' Alt-D : Active, '  
DB 'Not active', 0D, 0A, 24

A 0580  
DB 0D, 0A, ' TDB has been run !'  
DB 07, 0D, 0A, 24

A 05A0  
DB 0D, 0A, ' Now TDB is not active'  
DB ' !', 07, 07, 0D, 0A, 24

A 05D0  
DB 0D, 0A, ' Now TDB is active !'  
DB 07, 0D, 0A, 24

R  
R CX  
0500  
R BX  
0000  
N C.TDB.COM  
W  
Q



# 一个用于汉字字幕的图形编辑程序

四川彭水县公安局 张昶

**摘要:** 本文介绍一个用于汉字字幕的图形编辑程序,其特点是:1.可以用任一种方式输入汉字;2.字型可以显示在屏幕上任一位置;3.可以任意指定字体的纵横放大倍数;4.屏上的图形(包括汉字)可以搬移;5.可以点为单位对图形进行修改;6.可以将屏上任一部分图形存盘、取盘。

## 一、引言:

随着计算机应用的深入,程序设计水平的普遍提高,美观的屏幕显示越来越受欢迎,尤其是多字体高精度的屏幕显示,几乎成了一种时尚。近来各种刊物也介绍了不少这方面的工作[1],对追求屏幕美化的工作起到了推波助澜的作用。

本文介绍的工作,提出了一些新的功能和方法,叫它“显示程序”已经不太恰当了,所以称它“编辑程序”也许更适合些。稍加更改,可以实现一些电视特技,最近,我们甚至用它为我们自编的一个录像片配制了电视字幕,效果很好。

## 二、主要功能及其实现方法:

### 1. 拼音或其它任一输入方法输入汉字。

以往的屏幕显示程序,基本上使用区位码输入汉字,而且是在英文方法下用数值形式读入区位码。这样做的理由主要是避免在屏幕底部出现汉字选择提示行,影响画面美观。而本文采用的方法是允许出现此提示行,在存盘时再用图形方式抹去,实际画面上不会出现该行。

### 2. 以任意位置、任意倍数显示汉字。

在显示汉字前先提问该字的纵横放大倍数及坐标,以方框示之,然后由人工确定该位置合适否,不合适可以重选,合适则方框消失,显出汉字。此功能的提示行在屏顶,存盘时同样抹去。

### 3. 图形的搬移。

在图形编辑好以后,如还有不如人意的地方,可用此功能搬动。搬移使用了图形泵功能,也有方框提示。

### 4. 画面的人工修改。

这部分功能与普遍的造字程序方法差不多,不过是在图形方式下,用的点显示和点删除。

## 三、使用说明。

为加快执行速度,本程序使用 TURBO PASCAL 编写;为显示宋、仿、黑、楷多种字体,使用 CCDOS2.13 的  $24 \times 24$  点阵字库,工作时,这些字库应放入 C:\213 子目录下。若使用其它字库,仅修改计算字库记录号表达式即可。

### 命令简介:

L——取图;W——存图;S——显示宋体字;H——显示黑体字;F——显示仿宋体字;K——显示楷体字;T——显示字符及数字;M——图形搬移;E——进入点修改状态;Q——退出本程序。

E 状态(点修改)子命令: I——点上移; M——点下移; J——点左移; K——点右移; C——删除点; W——写点; P——保持原来点不变; E——退出点修改状态;

(附:程序清单)(见下页)

## 参 考 文 献

- [1] 《高精度多字体显示器画面构造程序》  
——北京市142信箱403分箱—李卫元  
《计算机应用研究》,1990年第1期

## 汉字图形编辑程序清单

```

C>
C)type sfhk.pas
program sfhk;
const l:=16006;
type hz=array [1..72] of byte;
      dot=array [1..24,1..24] of boolean;
      bufS=array [1..L] of byte;
var f:string[13];
    ch:char;
{$I graph.p}
Procedure pointp;
const maxX=319;maxY=175;minY=20;
var x,y,x1,y1:integer;
    stat,ch:char;
procedure move;
begin if stat='W' then colortable(3,3,3,3)
      else if stat='C' then colortable(0,0,0,0)
      else colortable(3,0,3,0);
      plot(x,y,-1);
      colortable(1,0,3,2);
      plot(x1,y1,-1);
      x:=x1;y:=y1
end;
procedure up;
begin
  if y>minY then y1:=y-1 else y1:=maxY;
  move;
end;
procedure down;
begin
  if y<maxY then y1:=y+1 else y1:=minY;
  move;
end;
procedure right;
begin
  if x<maxX then x1:=x+1 else x1:=0;
  move;
end;
procedure left;
begin
  if x>0 then x1:=x-1 else x1:=maxX;
  move;
end;
begin
  x:=1;y:=20;stat:='P';
  colortable(1,0,3,2);plot(x,y,-1);
  gotoxy(1,1);write('I/M/J/K/E/W/C/P');
  repeat
    read(kbd,ch); if ch=#27 then read(kbd,ch);
    ch:=upcase(ch);
    case ch of
      'P':begin plot(x,y,0);exit end;
      'C','W','P':stat:=ch;
      {'#72'}'I':up;
      {'#80'}'M':down;
      {'#75'}'J':left;
      {'#77'}'K':right;
    end
  until ch='P'
end;
Procedure line(a,b,c,d:integer);
begin
  colortable(3,2,1,0);
  draw(a,b,a,d,-1);
  draw(a,b,c,b,-1);
  draw(a,d,c,d,-1);
  draw(c,b,c,d,-1);
end;
Procedure edit;
var x,y,x1,y1,m,n,m1,n1:integer;
    ch:char;
    buf:bufs;
Procedure getsource;
begin
  repeat
    gotoxy(1,1);
    write('原坐标(X,Y,X1,Y1)=');READ(X,Y,X1,Y1);
    line(x,y,x1,y1);
    gotoxy(1,1);write('');
    gotoxy(1,1);write('(Y/N)');
    READ(KBD,CH);
    IF UPcase(CH)='Y' THEN LINE(X,Y,X1,Y1)
  until upcase(ch)='Y';
end;
procedure putnew;
begin
  repeat
    gotoxy(1,1);
    write('新坐标(m,n)=');READ(m,n);
    m1:=m+x1-x;
    n1:=n+y1-y;
    line(m,n,m1,n1);
    gotoxy(1,1);write('');
    gotoxy(1,1);write('(Y/N)');
    read(KBD,ch);
    line(m,n,m1,n1);
  until upcase(ch)='Y';
  LINE(X,Y,X1,Y1);
  COLORTABLE(0,1,2,3);
  getpic(buf,x,y,x1,y1);
  COLORTABLE(0,0,0,0);
  putpic(buf,x,y1);
  COLORTABLE(0,1,2,3);
  putpic(buf,m,n1)
end;
begin getsource;
      putnew
end;
procedure wwite;
var fn:file of BUFS;
    sn:string[13];
    buf:bufs;
begin
  gotoxy(1,1);
  write('写入文件名:');read(sn);
  assign(fn,sn);
  rewrite(fn);
  COLORTABLE(0,1,2,3);
  GRAPHWINDOW(0,175,319,199);

```

```

FILLSCREEN(0);
GRAPHWINDOW(0, 0, 319, 19);
FILLSCREEN(0);
GRAPHWINDOW(0, 0, 319, 19);
getpic(buf, 0, 0, 319, 19);
write(fn, buf);
close(fn);
end;
procedure LOAD;
var fn:file of BPF;
    sn:string[13];
    buf,bufs;
    CH:CHAR;
begin
    COLORTABLE(0, 0, 2, 3);
    gotoxy(1, 1);
    write('读入文件名:'); read(sn);
    assign(fn, sn);
    reset(fn);
    read(fn, buf);
    putpic(buf, 0, 199);
    close(fn);
end;
procedure drawzn(hzdot:dot);
var x,y,X1,Y1,i,j,A,B,m,n:integer;
begin gotoxy(1, 1);
    REPEAT
        GOTOXY(1, 1);
        write('放大倍数A, B:'); READ(A, B);
        GOTOXY(1, 1);
        WRITE('');
        gotoxy(1, 1);
        write('x, y'); read(x, y);
        X1:=x+24*A; Y1:=y+24*B;
        LINE(X, Y, X1, Y1);
        GOTOXY(1, 1);
        WRITE('正确吗?(Y/N)'); READ(KBD, CH);
        CH:=UPCASE(CH);
        LINE(X, Y, X1, Y1);
    UNTIL CH='Y';
    GOTOXY(1, 1);
    WRITE('');
    COLORTABLE(3, 0, 0, 1);
    for i:=1 to 24 do
        for j:=1 to 24 do
            FOR m:=1 to a do
                FOR n:=1 to b do
                    if hzdot[i, j] then
                        plot(x+(i-1)*a+m, y+(j-1)*b+n, -1)
end;
procedure tranzn(hzm:hz; var hzdot:dot);
var mmul, i, j, k:integer;
begin
    for i:=1 to 24 do
        for j:=1 to 3 do
            begin mmul:=$0;
                for k:=1 to 8 do
                    begin if (hzm[(i-1)*3+j] and mmul)<>0 then
                        hzdot[i, (j-1)*8+k]:=true;
                    else hzdot[i, (j-1)*8+k]:=false;
                    end
                mmul:=mmul div 2
            end
        end
    end;
    procedure getzn(var hzm:hz);
    VAR qu,wei,rec,i:INTEGER;
    QU:STRING[2];
    zk:file of hz;
    begin
        GOTOXY(1, 1); WRITE('');
        gotoxy(1, 1); write('汉字(图形)'); READ(QU);
        QU:=ORD(QU[1])-$A0; WEI:=ORD(QU[2])-$A0;
        rec:=(qu-16)*34+wei-1;
        IF QU='?' THEN REC:=REC+15*94;
        assign(zk, f);
        reset(zk);
        seek(zk, rec);
        read(zk, hzm);
        close(zk);
    end;
    procedure zn;
    var hzm:hz; hzdot:dot;
    begin
        getzn(hzm); (取字模)
        tranzn(hzm, hzdot); (字模转换成点阵)
        drawzn(hzdot); (显字模)
    end;
    begin
        clrscr;
        graphicormode;
        graphbackground(blue);
        palette(1);
        repeat
            GOTOXY(1, 1); WRITE('T/S/F/H/K/M/E/L/W/Q');
            read(kbd, ch);
            f:='Q:\213\HZK24';
            case upcase(ch) of
                'L':load;
                'W':write;
                'S':begin f:=f+'S'; zn end;
                'F':begin f:=f+'F'; zn end;
                'H':begin f:=f+'H'; zn end;
                'K':begin f:=f+'K'; zn end;
                'T':begin f:=f+'T'; zn end;
                'M':EDIT;
                'E':POINTP;
                'Q':begin hires;
                    hirescolor(yellow);
                    halt end
            end;
        until false
    end;
end;
end;

```

# 最少操作通用报表系统

新疆电子计算中心 袁正科

**摘要** 本文通过对报表的综合分析,经过系统地抽象,提出一种自动生成最小操作通用报表的思想和方法。

## 一、引言

目前,国内的中、小型管理信息系统80—90%是用dBASE系统开发的,因为dBASE命令简单、直观、有较强的数据处理能力,可使复杂的数据处理工作简单化,又易掌握使用。但使用dBASE开发管理软件时,碰到一个非常棘手问题即报表。人们通常有三种处理手段。第一是使用dBASE的报表系统,由于不具有汉字制表符,所以绝大多数的人不用它输出报表。第二是直接输出命令,如@...SAY,?,??,LIST,DISPLAY等命令编写程序,这种做法,编程工作量大,程序通用性差,难以维护,一旦应用环境发生变化,程序必须作一系列的修改,这不算是一种好方法。第三是通用报表系统。目前,国内有很多报表系统,功能很强,非

常灵活,不仅具有制表功能,还有一系列的统计功能。但是这些报表为什么不被广泛接受呢?最主要原因是用户友好性差,可以发现这些报表大多数是交互式的,换句话说要输出一张报表,需要作一系列的对话,如果对话一旦有错,要重新进行,再有引进了一系列的如标题,表尾说明,上表栏,左表栏,等术语。这些名词是由于要作通用报表才引入的,其实一个不懂计算机的办公人员制作一张报表,他根本就不需要知道这些名词,会在一张纸上迅速的画出来。当然用计算机完全模仿人的操作,程序是很难编写的,所以通用报表系统就做成了交互式的,从而也就降低了用户友好性。前两种方法效果差,第三种方法若能克服对话太多的弊病,就会大大地提高通用报表系统的实用性。这就需要认真地研究报表这一

该程序以滚屏方式显示一至多幅所造图形。

```
C>
C>type xs .pas
program xs;
CONST L=16006;
      ll=80;
TYPE BUFS=ARRAY [1..L] OF BYTE;
var fn:file of BUFS;
    sn:string[13];
    buf:bufs;
    CH:CHAR;
    I,J,k:INTEGER;
    B:BOOLEAN;
{$I graph.p}
begin
if paramcount>0 then
begin
    GRAPHICSMODE;
    PALETTE(1);
    COLORTABLE(0,1,2,3);
    graphbackground(blue);
```

```
FOR I:=1 TO paramcount DO
    BEGIN
    assign(fn,paramstr(i));
    reset(fn);
    read(fn,buf);
    close(fn);
    J:=199*2;
    graphbackground(i);
    repeat
    putpic(buf,0,j); j:=j-1
    until j=199;
    END;
    repeat
    {graphbackground(i);
    if i=6 then i:=0 else i:=i+1;
    delay(700)}
    until keypressed;
    hires
    END
end.
```



对象,事实上只要将那些名词不在报表系统中出现,就一定会降低对话次数,下面进行讨论。

## 二、报表抽象

报表一般说来可以抽象为如图1所示,其中标题(表名)是任何表都必须有的。表头说明可有可无,表尾说明包括页标和其他一些说明等,横表头是各栏的说明,可能是单层或多层的。左表头,可能是一栏或多栏,也可能整个没有。表体是报表的主体,是要输出的内容。经过一级抽象的报表,事实上还存在一个至关重要问题没有解决,报表是什么类型的,也就是说表格的特征是什么?要回答这个问题,必须进行二级抽象如图2所示。在图2中可以看出,表格是有几个特征制表符构成的,一共六个状态,形成八种形式。换句话说表格有八种基本形式,对于报表来说若加上有无左表头这一状态,那么就有十六种基本报表形式。报表经过一级,二级抽象后,可以得出这样一个结论:报表是由表体、表头、表说明和表格所组成,其中表格是由特征制表符所构成,其基本形式有十六种;表体是数据库要输出的信息;表说明是一种文字描述;表头是一种受约束的文字说明,所谓约束,如横表头的最下栏是要受数据宽度的约束。

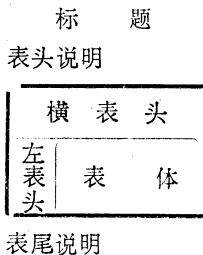


图1 报表一级抽象

制 表 符	状 态	类 型
□ □ □	有 无	封闭型 非封闭型
±   ±	有 无	竖隔线型 非竖隔线型
— — —	有 无	横隔线型 非横隔线型

图2 报表二级抽象

## 三、自动生成通用报表系统

报表经过抽象后得出的结论给自动生成最小操作通用报表系统提供了充分的依据,可以考虑,表说明部分既然是一种文字描述,那么在报表的样板“黑板”上给一块地方书写就可以了。关于标题问题,可以建立一个文件名对照字典,自动复制到相应位置。表头说明除了受约束的一部分需要特殊处理外,其他部分可以象表说明部分一样处理,受约束的部分也不妨建立一个字段名说明字典,这些说明同样可以复制到相应位置。对于表格部分的处理,既然有十六种基本型,不妨事先做成小样板,供用户选择。而这些基本型的表格特征是知道的,那么表格的自动生成就成为很容易的事。一旦有了报表的表格,将表体的数据填在相应位置上也变得很容易。这就是自动生成最小操作通用报表系统的思想。其系统结构如图3所示。

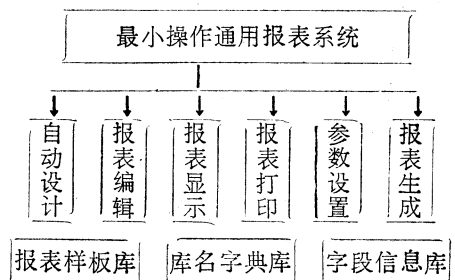


图3 最小操作通用报表系统结构

其中报表样板库中装有二十种报表格式,供使用时选择。库名字典中装有文件名和中文说明。字段信息库中装有字段的中文说明,标识名和字段其它信息。这些内容在dBASE自动生成应用系统中都是已有的东西。下面来简述模块的功能。

1. 自动设计。该模块完成报表的自动设计,由用户从报表样板库中选择一种形式,然后系统就自动生成出所需要的报表格式,并将第一、二个记录装备进去,实现格式设计。

2. 报表编辑。对由基本样板所生成的报表格式若不满意,可以进行交互式编辑,使得满足要求。

3. 报表显示。这部分显示报表输出结果或样板格式, 要输出的报表先不要打印, 在这里显示满意后再输出。由于报表信息比较多, 所以屏幕上采用窗口方式, 分为表头、表体和表尾窗口进行显示信息。

4. 报表打印。该模块完成已装备好的打印库, 并按参数设置的要求打印。

5. 参数设置。该模块完成设置打印机的每页行数, 行距压缩和恢复命令, 列距压缩和恢复命令, 并设置相应的打印驱动程序名称。

6. 报表生成。该模块完成报表的装备任务, 把要输出的库内容按样板表格的格式装备成打印库。数字0自动转换为空, 逻辑型自动装备成“T”或“F”。

该系统用 dBASE III PLUS 编写, 约占 50 KB 磁盘空间。

#### 四、应用举例

该报表系统的最大特点是操作次数少, 最好情况只需要三步操作, 第一进行自动设计, 第二进行报表装备, 第三打印报表。下面举例说明。

例: 教学管理数据库系统(部分)有如下文件:

成绩情况表: SCORE.DBF

结构: 学生编号 课程编号 成绩  
SNO(C, 6) CNO(C, 6) SCORE  
(N, 5, 2)

记录: 870101 C601 77.00  
870104 C604 88.88  
870106 C601 44.44  
870202 C603 66.00  
870101 C602 99.90

使用报表系统操作如下:

1. 自动设计。选择报表样板如下:

标题标题标题标题标题

表头注表头注表头注

页标:

名 称	名 称	...	名 称
内 容	内 容	...	内 容
.....			
内 容	内 容	...	内 容

表尾注表尾注表尾注表尾注表尾注

自动设计之后的装有数据的样板报表如下:

学生成绩表

学 生 编 号	课 程 编 号	成 绩
870101	C601	77.00
870104	C604	88.88

2. 对上表如果需要, 可以用报表编辑功能; 若需要修改, 就执行报表生成, 生成的可打印库如下:

学生成绩表

学 生 编 号	课 程 编 号	成 绩
870101	C601	77.00
870104	C604	88.88
870106	C601	44.44
870202	C603	66.00
870101	C602	99.90

这个结果就可以交去打印, 一张报表的制作过程就结束。

#### 五、结束语

最小操作通用报表系统是对国内众多的报表系统和若干实际应用中提炼出来的, 它最重要的特点是人机对话达到最小, 不需要用户设计表格, 众多的过程通过自动生成进行, 因此输出速度快, 实用性强。它存在的问题是不能覆盖所有的报表类型, 这需要增加一个用户自定义模块来实现这一功能, 不过这很容易做到。

#### 世界上最高速的通用计算机

日本《日经产业新闻》6月7日报道, 日立制作所推出的“M—880”机的处理速度为每秒1亿5千5百万次指令, 是美国IBM公司现在的最高功能的计算机处理速度的1.5倍。

## IBM Token-Ring 网的结构特点

吉林省计算机技术研究所 殷巍

吉林工业大学 李雄飞 陈思国

**摘要:** 本文综述了IBM Token-Ring环形网络的特点。尤其着重介绍了IBM Token-Ring网可与IBM大、中、小、微各类计算机、控制器和其它外设等连接,并实现数据通信和资源共享,以此来展示IBM Token-Ring网的可开发前景。

IBM Token-Ring网是一个基带网,它在局部信息系统中提供高效率、高速度的数据通信。该网完全符合IEEE802.2, 802.5及ECMA89三个标准。

Token-Ring网是环形网,环上循环传输着一个特殊的位串称Token,即“令牌”。令牌在环路上不断产生和循环。获得令牌的工作站有权发送自己的信息。若该站无信息发送,就把令牌传送给下一站。

### 一、Token-Ring网特性

#### A、易连性

Token-Ring网可连接各种IBM PC和工业用机及各种打印机,由此组成一个完整的通信系统,可靠、高效地传输信息。

#### B、可扩充性的连接能力

用户可按需要随时增加、缩减网上设备。所谓连接能力是指加到网上的设备立刻就可共享网上资源。如硬盘文件、高速打印机和各种应用程序。如果是资源设备,则立刻可被共享。

#### C、结构的灵活性

- 单环网可用双绞线连接,最多可连接72个设备。

- 较复杂的网可用屏蔽双绞线连接,最多可接260个设备。

- 使用光纤电缆,网间距离可达两公里,并保证在有噪音及电磁干扰的室外环境中的数据传输。

- 如需连接260个以上的设备,可采用多环结构,两个单环网间的连接需要一个桥接器。

#### D、丰富而先进的软件

IBM Token-Ring网目前面向PC机配置了较丰富的软件,可在DOS3.2或更高版本DOS

的支持下工作。

- APPC/PC (Advanced Program-to-Program Communication) 可以对一个企业范围的通信环境提供完善服务,并为Token-Ring网提供一个SNA LU6.2接口,允许Token-Ring网与SDLC通信链路进行通信。

- NETBIOS (Network Basic Input/Output System) 为Token-Ring网提供一个应用程序编程接口。它的功能是将信息分成便于在网上传输的形式,并在目的结点处重新组合它们。NETBIOS支持PC网络程序,而PC网络程序又支持约40个应用程序。它们都能在Token-Ring环境下,作为服务器的功能程序使用。

>IBM Token-Ring Network Bridge 这个软件可以配合硬件设备,连接多至260个设备的Token-Ring网。

- IBM Token-Ring Network Manager 主要处理临时性的和永久性的两类错误事件。对前一类错误,一般通过重新传递来纠正。有一个功能叫做Soft Error Condition,它可以发现和纠正单个和多个临时性错误。对后一类错误,是立刻将一个强烈的信号送往主控台,因为这类错误比较严重。它对这两类错误均做记载,以供程序员在处理以后发生的严重故障时参考。

先进的管理,及时的处理,可以避免网上资源浪费,最有效地发挥网的优势。

- Series/1-PC Connet Program, 这个程序允许一个或多个网上PC访问Series/1的高密硬盘和高速打印机。它认定Series/1

和Channel-Attached System/370的逻辑连接。

• Asynchronous Communications Server Program, 这个程序可使网上PC机作异步通信服务器。

以上这些程序都可以在网上为用户提高资源共享服务和信息交换服务。

## 二、IBM Token-Ring 网的各种结构

IBM Token-Ring 网是一种星—环形拓扑结构。基于IBM庞大的系列产品, Token-Ring网有着巨大的可开发潜力, 它可连接IBM大、中、小微各类计算机、外部设备和控制器等。整个系统的主要部分包括结点、传输介质、桥接器、网间连接器和环等部分。

### • A、8228多路访问器

在基本单环结构中, 可使用IBM8228多路访问器。一个8228可连接8个设备。8228可自动越过网上没有信号的发生故障的设备, 而重新形成环路, 保证网的正常运行。

### B、桥接器

在环网中, 当数据量增加超过单环网的容量时, 或者结点分布较广, 或结点数太多(超过260), 就要考虑多环网结构。两个单环网用桥接器连接。也可多个环网通过多个桥接器和主干网连接。桥接器是一个高速开关设备, 它类似其它结点一样连到网上。经它传送的信息在传送到另一个环网之前在桥接器中保存。桥接器还具备寻找路径的能力。每个环都有自己的标识和Token机构, 当桥接器或网上其它环有故障时, 可不受影响。为了满足网上较大的数据容量, 可采用多个桥接器互连。

### C、IBM System/36

用一台PCAT作为通信控制器, 可将System/36与Token-Ring网连接。这样可以提高网上设备数据通信的速度。在Token-Ring环境中, 所有Support/36功能都有效, 并可提供IBM PC和System/36间的信息交换和资源共享。而且, Token-Ring网和System/36的通信特性还对共享文件实现安全保护。

Support/36还为PC工作站提供仿真能力, 使IBMPC可仿真IBM5250的显示和打印, 特别还可以使IBM PC作为多重System/36对话期工作站; 作为IBM System/370主机上的3278访问器。

### D、Series/1

Token-Ring 网上的PC/XT或PC/AT可由PC附设提供的高速数据通道连接到Series/1处理器上。

这个PC的附设允许同一网上和不同网上的多个IBM PC工作站物理访问同一个Series/1, 利用它的资源。并且如果这个Series/1连到System/370上还可以实现数据通信。

也就是说, 这种连接也允许网上用户经由Series/1访问System/370主机应用程序。可见Series/1可改进Bisyn Chronous和SNA通信协议。此外, 网上用户也可访问所有与Series/1相连的设备。

### E、3725通信控制器

Token-Ring 网上连接一个3725可建立IBM PC对System/36、System/370应用程序的访问通信。3725通信控制器在System/36、System/370的网络控制程序控制下工作, 作为主机的前端机, 负责处理通信和网络管理, 具体执行有关线路扫描、校验、计算、调制解调器操作的控制、线路交换等任务。它采用存贮—转发工作方式转发网络信息。一个3725再加一个3726扩展器, 可连接多至8个Token-Ring网, 而且也十分有效地增强IBM PC和System/370之间应用程序的通信。

• 3270仿真程序—网上PC机可通过3725通信控制器访问在System/370主机上的3270仿真应用程序。

• System/36可通过3725连到网上, 并可访问与之兼容的APPC应用程序和其它主机应用程序。3725与Token-Ring网的连接是由3725上的TRSS(Token-Ring Subsystem)来实现的。它由连线 and Token-Ring 辅助设备、Token-Ring交换耦合器组成(TICS)。每个Token-Ring辅助设备及其线路构成一个



## 透明环境的特征

商业部安徽商业管理干部学院 陈庆章

透明性 (transparency) 是一个被广泛应用的术语。但究竟什么是透明性? 它有哪些确切的含义? 透明环境的主要特征是什么? 这些问题人们可能并不十分清楚。本文将对此作一浅显的解释。

透明性的最基本含义可以通俗地概括为“单系统印象”, 即不管系统多么复杂, 资源的地理位置如何分布, 设备可能的增加和删除, 表现在用户面前的只是一个单系统。

利用高度的透明性, 用户能吸收一个分布式的和异构的环境的各种优点, 根本不需要对现有软件做任何重要的修改; 在运行某一应用时, 可以使它的一部份在某台机器上运行, 而另一部份在其它机器上运行, 甚至每台机器的结构完全不

一样; 在一个透明的环境里, 用户看到的是哪里有系统资源, 而用不着关心那里的资源是否正在被利用; 用户可以使进程在一个低负载的 CPU 上执行而不必念及它在什么期间被执行; 用户能扩展和减少最基础的计算环境而不会破坏正在进行的处理活动; 用户也能简单地将存储器或处理器相加或重移; ……。

一个透明的环境通常包含以下方面内容:

### 1. 访问透明 (access transparency).

它用于文件、设备、进程和进程之间的互相通讯。它意味着用户能使用同样的系统调用, 而不必关心资源的位置。它支持文件访问同步, “遮掩机器边界”, 提供近似单机的环境。

### 2. 设备透明 (device transparency).

多路转换器, 一个多路转换器可接四个交换耦合器, 16个通信线路的附设。Token—Ring附近可与3725连接; 在带有3726扩展器的3725上最多可连8个Token—Ring附设, 32个通信线路的附设。Token—Ring 交换耦合器提供 Token—Ring网上的附设, 符合IEEE802.5和 ECM A89的标准。以每秒400万 Bit 的速度传送和接收信息。

### F. 3270 SNA Host 网间连接器

当前的网络环境是处在某局域范围内的大量用户互连, 共享资源和进行通信。例如, 一个大型的主干架计算机通过网间连接器就可以给其周围广阔地区内的用户提供访问。网间连接器也是一种结点, 用它处理不同 LNA 间的转换工作, 包括地址、速度、协议等转换。网间连接器还可把环网与各种不同的通信系统连接。例如, 为了利用 SNA 应用环境, 需要 3270 仿真程序和 SDLC 适配器。通过 SDLC 适配器可使 PC 机与 System/370, System/38, System/36 各种计算机以及 Series/1 设备进行通信。运行 PC3270 仿真程序可把网上 PC 机作为 SNA System/370 主干架计算机的仿真终端使用。在局

网环境下, 网上工作站亦可通过 3275 通信控制器访问 System/370 上的 3270 仿真程序, 而实现与 System/370 的通信。

一般来说, 网间连接器也可作为一个网上工作站。至于它能否作为工作站来使用, 这取决于它所负担的工作总量。条件允许, 它还可作为文件服务器和打印服务器。

## 三、结束语

总之, IBM Token—Ring Network 有着巨大的开发潜力。它的数据吞吐能力高于令牌总线和 CSMA/CD 总线。另外, 令牌环易于实现高速度、远距离和数据通信。尤其和 CSM A/CD 总线型相比, 它能连续地经受过载状态, 稳定性极佳。

## 参 考 文 献

[1] IBM 资料 IBM Token—Ring Network A Functional Perspective

[2] 计算机网络 李显济主编 机械工业出版社

[3] 令牌环网现况 小型微型计算机系统

88年 9卷 8期

它是访问透明的一个子集。设备透明有二个含义：第一，设备应该象文件一样有全局的唯一的名字，即一个设备应当通过相同的名字来访问，而不管设备与哪个机器相连以及访问来源于哪个机器。第二，设备的功能应当是均匀地供所有机器利用，而不是专属于某机器或偏重于某机器。均匀的供访问不仅包含一般的访问操作，而且也包含对处理设备的Unix系统调用。

3. 进程透明(process transparency)。它也是访问透明的一个子集。就网络透明来说，进程透明是关键。它的含义是：不管是在同样的CPU上运行，还是在几个CPU上运行，每个进程应当能看到其它进程，所以要求系统能可靠地把信号送出并得到正在其它CPU上执行的进程的状态。双亲进程没有必要了解它的子孙进程是在哪里执行，也不需要任何涉及到远程进程的额外代码。在远程机上所创建的进程必须继承所有同样的在本地机器上所创建的进程的目标。

4. 定位透明(location transparency)。意思是拥有资源的CPU的名字并不是嵌入路径名中的，由此便造成不需要改动资源的名字就能移动资源。资源的名字并没有指出它的位置，嵌入位置以二种方式损害透明性。第一，用户不能透明的对数据重定位；第二，对存贮位置名字的编码使复制资源困难，而使用资源的备份是获得性能改进和提高可利用性的前提。

5. 名字透明(name transparency)。在不同的CPU上使用同样的名字将会访问到同样的资源。换句话说，在一个分布式的环境里，每个资源必须要有一个全局唯一的名字供所有的CPU来访问。

6. 性能透明(performance transparency)。包括对远程资源的访问开销与对本地资源的访问开销相比起来相差无几。

以上六个方面的透明性的结合就构成了网络的透明。

有几项重要的技术的实现对形成透明环境是非常关键的：

1. 进程执行场地的控制。用户能限定进程在哪里执行。用户可以提供程序的外部信息，如打算在哪个机器上执行，执行顺序表，何谓非忙执行场地，靠机器最近的数据存放处等。

2. 在执行期间的进程重定位。用户可能需要在任务执行期间部份地或全部地改变执行场地。如果环境是透明的，用户能将进程从一个处理器移向另一个处理器，并且没有任何感觉仍能完成任务。不仅用户可以改变进程执行场地，进程它自己如果由于某种原因想移动也是可以的，如进程可能想更靠近数据或设备。

3. 在其它CPU上创建进程。用户可根据负载状况，或用户环境提供的线索，或随机地选择一个CPU并在其上创建进程。

4. 不破坏正在进行的工作的资源的增加。系统应当能够自动地增加资源。在正常运行期间，用户能创建文件的拷贝并立即使用它们。系统应当能传播所有备份的任何变化。

5. 没有服务损失的资源迁移。可有计划地或无计划地移动资源。没有损失资源的活动应正常进行。系统应能自动地代换等价的资源。

“单系统印象”可能只是一个强意词，但它确是对透明环境的一个精确的比喻。

## MA SP-P电能计量遥测处理系统达到国际先进水平

〈本刊讯〉一种具有国际先进水平的电能计量遥测处理系统已由贵州省科委计算中心研制成功，申请了国家级技术专利，并于最近通过省级技术鉴定。

本系统是一种通过公共电话交换网对各种用电户用电量的表计值作远程自动采集和实施相应的管理策略的一个计算机—通信—测量仪器三位一体的数据采集处理网络，主要由以下四个独立的部件构成：

1. ZDD-1型电量远动抄表计；2. ZDD-2型自动拨号/接收控制器；3. 中文远程通信处理软件；4. 数据处理软件。

本系统通过公共电话交换网联网，进行多用户、多结点的电度量自动遥测处理，可实现月末零点值的统一抄表，亦可对某一用电户进行随时/定时监测。从根本上解决了人到现场抄表这种既费工又费时的现状，同时使得一些数据，如负荷率、最大需求量、线损等得到准确的检测和统计，为保证有限电力的合理使用提供了可靠、有效的依据。

本系统性能可靠，具有安装简单、操作维护简便、造价低廉、易于推广，并适用于各种累计型表计的自动采集和联网应用。

# 微电脑人体心肌梗塞预测方法

吉林省计算中心 张鸿鸣 田卫星

长春市中医医院 刘翠华 吉林省生物研究所 胡庆贤

近年来,国内外医学专家对早期估计梗塞量十分重视,取得较理想的梗塞量,并以此预测人体心肌梗塞,收到较好的效果。由于人体心肌梗塞是由持久而严重的心肌急性缺血引起的部分心肌坏死,伴有心室功能障碍,临床上产生胸痛和组织坏死的一些全身反应及心电图(和向量图)上反映心肌急性损伤,缺血和坏死的持久与进行性演变,常并发急性循环衰竭及严重心律失常。因此,人体心肌梗塞量的估计对于急性心肌梗塞患者的预后、治疗及疗效评价有着极其重要的意义。

本文提出了应用微电脑预测人体心肌梗塞的方法,同时把对数正态模型和病理模型进行了比较。通过大量病例验证表明,病理模型预测人体心肌梗塞的效果要比对数正态模型的预测效果好得多。

## 一、对数正态模型

设  $t=0$  时发生心肌梗塞  $E(t)$  为时间  $t$  时血清  $cpk$  活性。

$$Y(t) = E(t) - E(0) \quad (1)$$

到  $t=T$  时刻心肌向血液释放的  $cpk$  累积活性为  $ckr$ , 梗塞量  $X$  为:

$$X = \frac{a}{q} \cdot W \cdot ckr \quad (cpk\text{—克} \cdot \text{当量}) \quad (2)$$

其中  $W$  为人体重量;  $—$  为常量。

又经验公式:

$$Y(t) = \frac{b}{t} E^{-\frac{(\ln t - c)^2}{2d^4}} \quad (3)$$

被称为对数正态模型。

这时有

$$ckr = Y(T) + k \cdot \int_0^T Y(t) dt$$

从而得心肌梗塞量为

$$X = \frac{a}{q} \cdot W \cdot ckr$$

这一方法不能早期估计  $k$  值,只有当  $cpk$  活性降至末段才能确定,所以此法早期估计心肌梗塞量不准确。

## 二、病理模型

众所周知,心肌梗塞后不是所有细胞同时坏死和介体,  $cpk$  分子进入血液所需时间  $\tau$  为随机变量,由正态分布可得

$$Y(t) = h \cdot \phi\left(\frac{t-\mu}{\sigma} - \sigma k\right) \cdot E^{-k\mu} \quad (4)$$

$$\text{其中 } h = \frac{qx}{aw} E^{k\mu + \frac{\sigma^2 k^2}{2}} \quad (5)$$

由 (5) 式,可得心肌梗塞量  $X$  为

$$X = \frac{a}{q} \cdot W \cdot h E^{-k\mu - \frac{\sigma^2 k^2}{2}} \quad (6)$$

我们用每隔4小时采样一次所得到四个数据,进行曲线拟合。求出  $h$ 、 $\mu$ 、 $\sigma$  和  $k$  的最小二乘法估计值,故可由 (6) 式直接得出梗塞量的估计值。其中  $k$  值无需单独计算,因此可在早期确定出可靠的心肌梗塞量,预防人体心肌梗塞病的发生。

另外,对梗塞范围扩大的情况(即二次梗塞的发生),对数正态模型法是无能为力的,但我们可用迭代的方法推出病理模型公式:

$$Y(t) = h_1 \phi\left(\frac{t-\mu}{\sigma} - \sigma k\right) E^{-kt} + h_2 \phi\left(\frac{t-t_1-\mu}{\sigma} - \sigma k\right) E^{-k(t-t_1)} \quad (7)$$

其中  $t_1$  为发生第二次的梗塞时间。

因此,可求出总梗塞量  $X_1 + X_2$ , 即为

$$X_1 + X_2 = \frac{a}{q} W \cdot (h_1 + h_2) E^{-k\mu - \frac{\sigma^2 k^2}{2}} \quad (8)$$

### 三、微电脑算法

对  $Y(t)$  的四个参数  $h, \sigma, \mu$  和  $k$  的估计, 我们采用非线性曲线拟合方法, 该算法具有引进一个阻尼因子到法方程组的特点。

令参数分别为  $b_1, b_2, b_3$  和  $b_4$ ; 并设一组初值  $\{b_i^{(0)}\}$  又由四次数据  $(y(t), t)$ , 可以计算出残差平方和  $Q$ ; 还可以求出法方程组的各个系数  $a_{ij}$ , 法方程组为,

$$\begin{cases} (a_{11} + D)\Delta_1 + a_{12}\Delta_2 + a_{13}\Delta_3 + a_{14}\Delta_4 = a_{15} \\ a_{41}\Delta_1 + a_{42}\Delta_2 + a_{43}\Delta_3 + (a_{44} + D)\Delta_4 = a_{45} \end{cases} \quad (9)$$

其中  $D$  为阻尼因子, 初值  $D_0 = 0.01$ .

由法方程组可得一组值  $\{\Delta_i\}$ , 代入迭代方程得,

$$b_i = b_i^{(0)} + w_i \Delta_i \quad (i = 1, 2, 3, 4) \quad (10)$$

若迭代过程发散, 即当  $Q \geq Q_0$ , 只要增大  $D$  值, 可阻止发散趋势。从理论上讲, 对于任何初值, 只要  $D$  趋于充分大, 总可使迭代过程为收敛的, 也就是可有  $Q < Q_0$ , 这样就可得一组迭代值  $\{b_i\}$ 。

若令  $b_i^{(0)} = b_i$ , 又重新开始一次新的迭代过程, 重复进行, 直到当  $|\Delta_i|$  均小于预先给定的阈值为止, 这时得到的  $\{b_i\}$  当前值为参数的估计值。

从而心肌梗塞的估计值为

$$\bar{X} = 9.997 \times 10^{-3} \times w \cdot b_1 E^{-b_3 b_4 - \frac{b_2^2 b_4^2}{2}}$$

该算法的特点在于初值的范围大, 迭代收敛速度快。虽然对于不同患者这些参数可能相差很大, 但我们仍可给出一个统一的初值进行迭代, 而求出所需要的参数值。迭代误差可用均方差来估计迭代优化度。

第二次心肌梗塞情况可以根据相应公式拟合得出, 此时参数为 6 个。

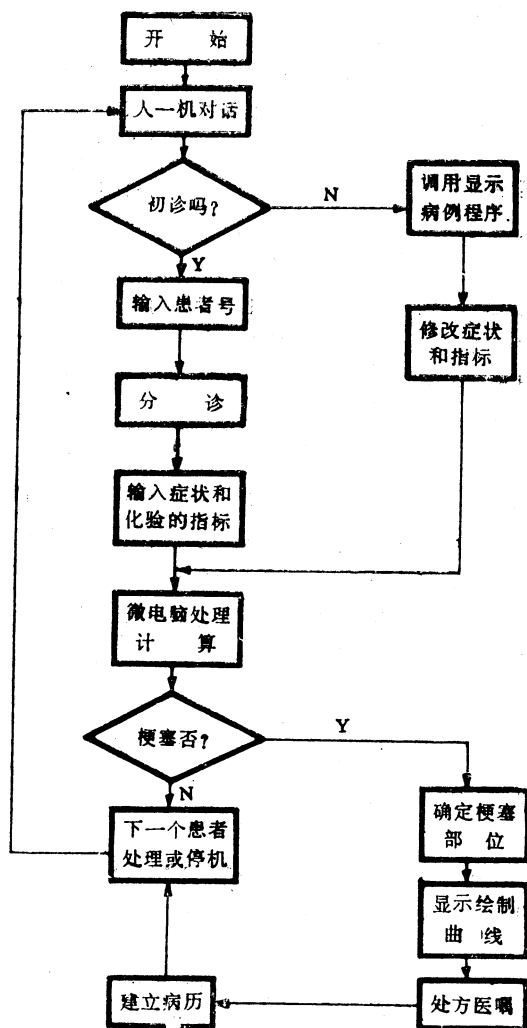
### 四、微电脑程序框图

该系统包括四个子系统:

1. 诊断、病历管理系统。2. 数学模型计算

子系统。3. 图形显示绘制子系统。4. 处方、医嘱子系统。

本系统灵活, 便于修改和维护, 其系统总框图如图:



总框图

### 五、微电脑程序

微电脑预测人体心肌梗塞的程序是用 BASIC 语言编制的, 已在 IBM PC 和 CEC-1 中华机上实现。计算一个患者的病例仅需二十几分钟, 还可以预报 12 小时以后  $cpk$  活性的动态变化, 供医生参考。

该程序还具有图形显示及绘制子程序, 可以显示和给出  $cpk$  动态曲线和病理模型曲线, 并可

# 计算机在粒度分析数据处理中的应用

新疆石油学院 马晓东

新疆电子计算中心 张亚玲

**摘要** 粒度分析数据处理是一项十分繁琐的工作,笔者提出一种用计算机对粒度数据进行处理的方法,并在实际应用中得到满意的效果。

## 一、前言

本世纪60年代以来,国内外学者对粒度分析数据的处理提出了许多有效的方法。这无疑在沉积学的深入研究上起到了积极的推动作用。然而,处理方法至今基本上还是利用手工进行。为达到粒度分析数据处理的简明、快速、可靠的目的,本文提出了一种利用计算机对粒度分析数据进行处理的方法。

## 二、粒度分析数据处理方法

### 1. 粒度数据分布的处理

粒度分布常可用直方图表示其特征。其横坐标表示粒度;纵坐标表示粒级的百分含量。考虑到有利于建立粒度分析标准化,其横坐标采用“伍登—温德华— $\phi$ 值”粒度标准(表2—1)进行标度。

用实验所得的一个样品的数据( $X_i, F_i$ ),  $i=1, 2, \dots, n$ ,在上述坐标系中绘出直方图。该直方图能够快速、准确、直观地表现出实验所

进行直观比较。经临床验证,用12小时的四个数据(甚至前7小时)作的早期心肌梗塞量的估计和全部cpk数据作的梗塞量的计算,其差异不大,可定量分析。此程序还具有用对数正态模型计算梗塞量的子程序,与病理模型法可随时进行比较,诊断准确率可达90%以上。

## 六、两种方法的比较

病理法与对数正态法相比,至少具有以下三个特点:

1.可以在早期确定心肌梗塞范围,进行早期预测。2.可以计算估计心肌梗塞范围扩大后的梗塞量。3.可在cpk活性曲线上升段确定k值。

实践证明,用60至70小时全部cpk数据拟合

表2—1 伍登—温德华— $\phi$ 值粒度标准

沉积物类型	$2^n$	整数及分数形	小数形式(mm)	$\phi$ 值
砾	2 <sup>5</sup>	32		-5
	2 <sup>4</sup>	16		-4
	2 <sup>3</sup>	8		-3
	2 <sup>2</sup>	4		-2
砂	2 <sup>1</sup>	2		-1
	2 <sup>0</sup>	1		0
	2 <sup>-1</sup>	1/2	0.5	1
	2 <sup>-2</sup>	1/4	0.25	2
粉	2 <sup>-3</sup>	1/8	0.125	3
	2 <sup>-4</sup>	1/16	0.0625	4
	2 <sup>-5</sup>	1/32	0.03125	5
	2 <sup>-6</sup>	1/64	0.015625	6
砂	2 <sup>-7</sup>	1/128	0.0078125	7
	2 <sup>-8</sup>	1/256	0.00390625	8
	2 <sup>-9</sup>	1/512	0.001953125	9
	2 <sup>-10</sup>	1/1024	0.0009765625	10

得的样品的粒度变化、各粒级碎屑的百分含量及粒度分布的众值 and 对称性。

### 2. 累积曲线的拟合及粒度参数的确定

累积曲线用以表示粒度分析数据的累积百分

的结果,这两个模型的效果差别不大,但是若只用前12小时内的四个数据拟合,则对数正态模型的拟合优化度显著地低于病理模型;若仅用7个小时的数据,则对数正态模型的理论估计与观测值之间的差距更大,而且表明病理模型的拟合优化度仍然很理想。

总之,由临床应用的事实表明病理模型适用于早期估计人体心肌梗塞范围,应用微电脑预测人体心肌梗塞病的发生,而且该程序还具有通用性,也很容易移植到其它机种上,使用方法简便,计算快,使用效果好。如果与心电图机配合使用,该方法的效果更佳。

比值的变化规律。它的横坐标的标度及意义与直方图相同；而纵坐标表示的则是各粒级的累积百分含量（累积数据是由粗粒级开始计算的）。

累积百分比值的变化规律总是随 $\phi$ 值的增减而变化，“逻辑斯蒂”曲线来表示：

$$Y = \frac{a}{1 + be^{-cX}}$$

（ $a, b, c$ 为待定系数） (1)

累积曲线拟合的关键在于确定（1）式的三个待定系数。由于累积百分比值的增长率最终将稳定于某个水平。因此 $a$ 是已知的，在此

$a \approx 100\%$ 。为确定另两个系数 $b, c$ ，这里采用变量代换。通过变量代换（1）式变为：

$$Y = A + BX$$

利用线性回归可确定出系数 $A, B$ 。并将 $A, B$ 和 $X = \phi$ 回代，则（1）式变为：

$$Y = \frac{1}{1 + e^{A+BX}} \quad (2)$$

利用（2）式，求出 $Y^{-1}(\Phi)$ 。

应用福克（FOLK）和沃德（WARD）公式（表2—2）求出图解粒度参数 $Md, Mz, BP, SK_1$ 和 $K_G$

表 2—2

常用的粒度参数

名 称	福 克 和 沃 德
中 值	$Md = \Phi_{50}$
平均粒度	$Mz = (\Phi_{10} + \Phi_{50} + \Phi_{90})/3$
分 选	$BP = (\Phi_{84} - \Phi_{16})/4 + (\Phi_{95} - \Phi_5)/6.6$
偏 差	$SK_1 = \frac{\Phi_{10} + \Phi_{84} - 2\Phi_{50}}{2(\Phi_{84} - \Phi_{16})} + \frac{\Phi_5 + \Phi_{95} - 2\Phi_{50}}{2(\Phi_{95} - \Phi_5)}$
峰 度	$K_G = \frac{\Phi_{95} - \Phi_5}{2.44(\Phi_{75} - \Phi_{25})}$

### 3. 概率累积曲线的拟合

粒度分析的主要图件——概率累积曲线横坐标同前，而纵坐标改用按单峰正态曲线分布规律刻化的概率百分数坐标。它的一般规律是：由两段、三段或四段斜率 $k$ 有着明显差异的相连直线段组成。针对这种规律，可采用线性回归来拟合此曲线。

通过对维谢尔（visher, 1969）各典型环境的142个样品的概率累积曲线进行对比、统计得出：滚动次总体、跳动次总体和悬浮次总体各段的 $k$ 在一般情况（曲线为普通三段式）时，分别分布在 $(a, b]$ 、 $(b, c]$ 和 $(a, b]$ 三个区间范围内。令筛析所得的一个样品的数据 $(X_i, Y_i)$ ， $i=1, 2, \dots, n$ ，遵从 $X_1 < X_2 < \dots < X_n$ 的顺序，用两点式求其各两点间的斜率，依据上述三个区间，对该样品数据 $(X_i, Y_i)$ 进行分类，得到三组数据： $(X_{1i},$

$Y_{1i}), (X_{2i}, Y_{2i}), (X_{3i}, Y_{3i}), i=1, 2, \dots, j (j < n)$ 。用这三组数据分别求各自的线性回归方程，便得到该样品概率累积曲线的近似公式：

$$Y = \begin{matrix} A_1 X + B_1 & [X_{1i}, X_{2i}] \\ A_2 X + B_2 & [X_{2i}, X_{3i}] \\ A_3 X + B_3 & [X_{3i}, X_j] \end{matrix} \quad (1.)$$

式中： $A_1, B_1, A_2, B_2, A_3, B_3$ 为待定系数。

待定系数可用最小二乘法求得。

方程组： $Y = A_1 X + B_1$

$$Y = A_2 X + B_2$$

的解为粗截点 $T$ 的坐标。

方程组： $Y = A_2 X + B_2$

$$Y = A_3 X + B_3$$

的解为细截点 $s$ 的坐标。

上述是曲线为一般情况时的处理。而在特殊情况（曲线为四段式和三角洲前缘型的三段式）时，分别依据K的分布区间：

$$\begin{array}{ll} (e, f] & [i, j] \\ (f, g] & [k, l] \\ (g, h] & \text{和} [m, n] \\ (e, i] & \end{array}$$

将实验所得样品数据进行分类，再依照前述原理，便可得到样品概率累积曲线的近似公式。

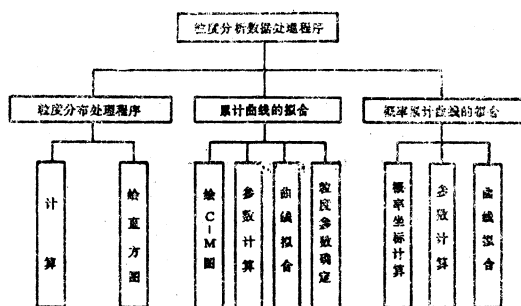
#### 4. C、M 分位值数据的处理

C、M 分位值数据可绘制反映 C、M 的散点图—C—M 图。

C—M 图在解释环境上具有灵敏性强、成功率高等优点。尤其是在区别牵引流和沉积物重力流上更为敏感。因此该图也是沉积环境分析中必不可少的图件。它的横坐标表示 M 值，是非等间距的对数坐标；纵坐标表示 C 值，也是非等间距的对数坐标。

一组样品  $Y_n$  ( $n$  为样品数) 的实验数据  $n$  ( $X_i, Y_i$ ),  $i=1, 2, \dots, m$ , 通过第二节 (2) 式分别求出各样品的  $\Phi_{50}$ 、 $\Phi_{10}$ ；将  $Y$  的  $n$  个 ( $\Phi_{50}$ 、 $\Phi_{10}$ ) 置于前述的双对数坐标中，其结果便是所要绘制的 C—M 图。

### 三、粒度分析数据处理程序框图



### 四、粒度分析数据处理实例

利用上述原理和方法，笔者分别对西北某地， $J_2-s$ 、中原某地  $S_{1/2}$  层位的 42 个样品进行了处理。处理结果如图 4—1, 2, 3, 4 所示。直方图原始数据、累积曲线和概率累积曲线原始数据、C—M 图的原始数据从略。程序清单从略。

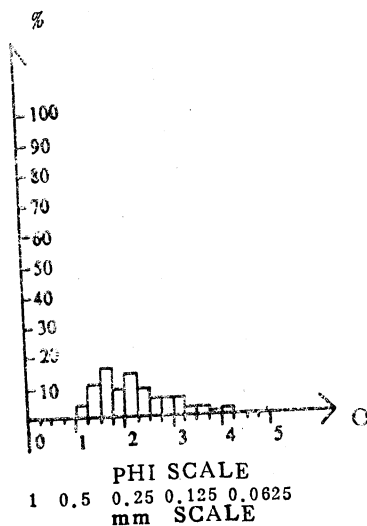


图 4—1 粒度分析直方图

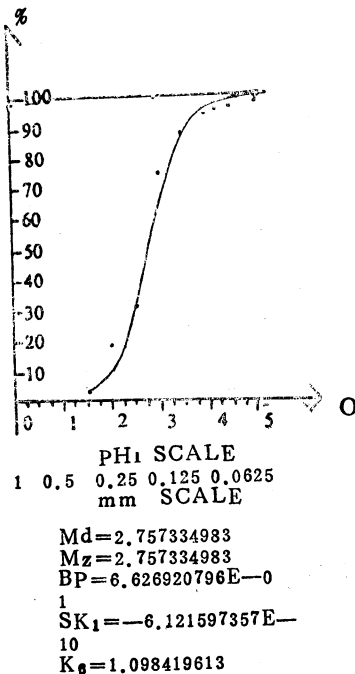


图 4—2 累积曲线及图解粒度参数

### 五、处理结果的几点说明

用上述方法处理了 87 个样品的粒度分析数据，对处理方法及结果有如下几点说明：

#### 1. 薄片分析数据处理结果的校正

前面几节是以筛析数据的处理为例进行介绍



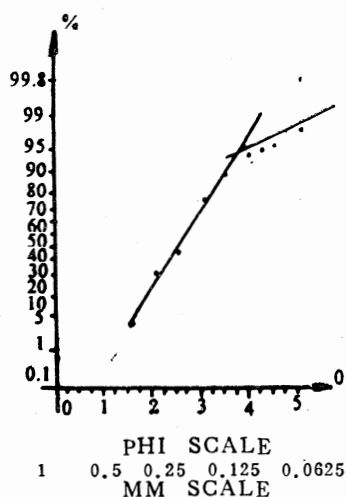


图4-3 概率累积曲线

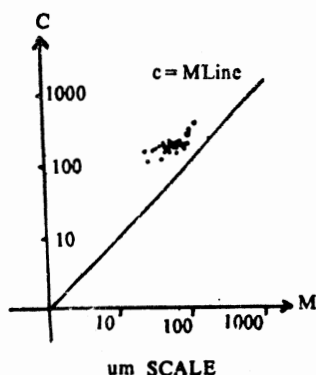


图4-4 C-M图

的。至于沉降粒度分析数据，在精度要求不很高时，也可运用上述方法进行处理。而薄片粒度分析数据由于存在切面视直径与筛析直径之间的偏差（可达 $0.25\phi$ 或更大），因而在任何情况下均不可互用或直接对比。只有采用 Friedman 回归校正方程，将薄片值校正成筛析值后，方能与筛析模式进行对比。具体校正时，只需将薄片分析数据按上述几节方法处理的结果的关键点，根据 Friedman 校正方程进行位移。

为避免无“悬浮次总体”尾端，在切面直径校正到筛析直径前，需进行薄片分析值的杂基校正。考虑到杂砂岩多为重力流沉积体系的产物（其校正值  $\Delta = \text{杂基含量}\% / 2$ ），故对净砂岩

的校正值取  $\Delta = 7/12$  杂基含量%。

## 2. 关于处理结果的准确性、可信性

直方图的绘制基础——粒度分析数据的“分组间距”直接影响该图的形状和粒度分布的对称性。笔者选用 $1/4\phi$ 的分组间距，以此保证该图的精度并兼顾达到易于建立标准化的要求。

运用上述方法处理而得出的结果，是沉积环境研究中几种必须的重要图件和参数。这些图件均采用与目前国际上公认的标准图版（Visser 1969; Passega 1964）一致的比例。因而便于对比。粒度参数是采用当前各油区广泛应用的福克（Folk）和沃德（WARD）的计算公式确定的，因而有利于建立总公司系统的标准化。

## 3. 实用价值

计算机对粒度分析实验数据进行处理，加快了作图和计算速度，同时也提高了正确性和精度（尤其对概率累积曲线）。手工处理效率低且标准化程度差，出错机会多。用计算机进行处理，最多不超过6分钟即可得出一份完整、清晰的结果来。

## 参 考 文 献

- [1] 中国科学院计算中心概率统计组：《概率统计计算》，科学出版社，1982
- [2] 中国科学院数学研究所统计组：《常用数理统计方法》，科学出版社，1973
- [3] 西安矿业学院数学教研室：《数学地质基础与方法》，煤炭工业出版社，1989
- [4] F.P. 阿格特伯格：《地质数学》（中译本），科学出版社，1981
- [5] 张校正等：《袖珍计算机 PC-1500 编写程序语言及外围设备》气象出版社，1984
- [6] 刘宝琛：《沉积岩石学》，地质出版社，1980
- [7] Visser G.S., Grain Size distributions and depositional processes. Jour. sed. petrology, V.39, P.1082-1102, 1969

# IBM PC 上的程序流程图编辑/打印软件

——Flow Charting II + 介绍

广州中山大学计算中心 肖俊良

Flow Charting II 是 Patton & Patton 软件公司1982年推出的一个程序流程图编辑/打印软件。

为了让 **Flow Charting II +** 能发挥它应有的作用, 下面将软件的使用环境和主要特点作一简单介绍, 以让大家对它有所了解。

### 一、Flow Charting II + 的使用环境

使用Flow Charting II + 的最小配置  
为.

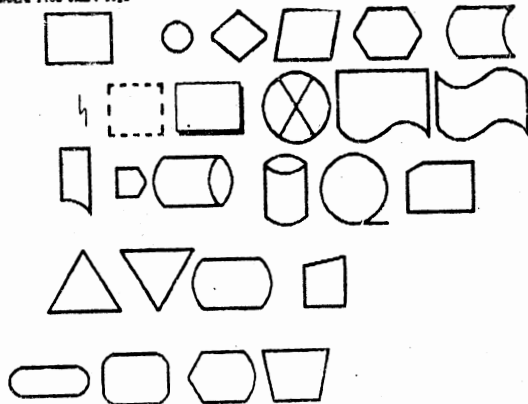
1. IBM PC或其兼容机; 2. 256KB RAM;  
3. 图形卡一块, 与之配套的显示器一台(彩色或单色); 4. 软盘驱动器一台; 5. 点阵打印机一台

## 二、Flow Charting II + 的特点

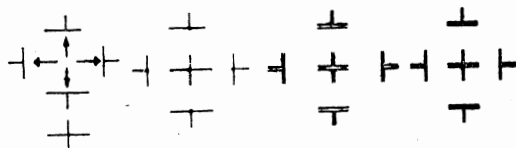
- ### 1. 具有标准流程图符号

系统内部有标准流程图符号可供用户调用，并且各种符号可以用箭头键→←↑↓改变其大小和宽窄等不同形状。

系统具有的标准流程图符号如下:



2. 有适合多种用途的连接符和线型  
具有的连接符如下:



这些连接符的适当使用，可以制作多种图表，下面是其中的一幅：

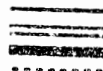
PATTON &amp; PATTON 1985 PURCHASER ORDER LOG

PAGE # DATE:

[illegible]

另外，系统有多种线型可供选择：

### Line Style



- ### 3. 具备多种字根(体)集

可以选择普通、加深、希腊字母、上、下标等多种字体。

系统具备的字根（体）集如下：

**Text Font**

Normal  
Bold  
Italic  
Fat  
High  
Greek  
Superscript  
Subscript

ABC  
ABC  
ABC  
ABC  
ABC  
ABC  
ABC

Micro  
Title

MICRO  
Title



别适合于书写查询和增删库中信息的数据/知识库管理系统。

由于 **prolog** 具有上述特点, 所以, 它将在极其广阔的领域中获得应用, 这包括:

- 1) 计算机辅助设计, 例如建筑、生化、电路等。
- 2) 程序验证和综合合成问题的求解, 定理证明, 公式处理, 人工智能研究。
- 3) 数据库/知识库管理。
- 4) 自然语言处理。
- 5) 专家系统。
- 6) 超高级模拟。

本文将通过 IBM PC机上运行的 **MICRO-prolog** 的几个实例和说明 **prolog** 系统内部的定理证明机制主要采用的两种方法完成解题任务: 模式匹配 (**pattern matching**, 或称一致化——**Unification**) 和回溯 (**Back-tracking**), 以及在操作 **micro-prolog** 时的注意事项。

#### 1) 模式匹配。

在讲述模式匹配以前, 我们来看一下阶乘求解问题。我们知道:

$$1! = 1$$

$$n! = n(n-1)!$$

我们如何用 **prolog** 来求解这个问题呢? 我们的办法是用两个句型作为 **prolog** 程序中的知识库存入计算机。

```
fact(1,1)
fact(x,y):-x_1 is x-1, fact
(x_1,y_1), y is x*y_1
```

以上是一般 **prolog** 语言对阶乘问题的描述。但是, 这种形式对于 **MICRO-prolog** 是不能接受的。其原因之一是对于 **is** 这个内部谓词, **micro-prolog** 中没有定义, 原因之二是减号的处理。在 **micro-prolog** 中包括着两种语法, 一种是 **simple** 语法, 另一种是标准语法 (管理程序的 **micro** 扩充)。无论那种语法它们都不直接支持表达式。只有在装入 **EXPTRAN** 后你才能在程序中使用表达式。

因此程序的正确输入过程如下:

```
> PROLOG LOAD SIMPLE
```

```
& LOAD EXPTRAN
```

这时把大写字母输入方式转为小写。

```
& add (fact (1 1))
```

```
& add (fact (x y) if SUM (x_1 1x)
      &fact
```

```
1 (x_1 y_1) &
```

```
1 TIMES (x y_1 y))
```

这时我们才能得到程序的知识库。这里顺便提一下, 我们使用了命令 **add (<句子>)** 把句型送入数据库。同时使用了两个内部谓词 **SUM** 和 **TIMES**。不难猜出一个是加, 一个是乘法。我们也可用它们做减法和除法。这里就不细说了。

现在, 让我们回到我们的中心问题上去 (模式匹配)。库中的第一个语句说明 1 的阶乘为 1。第二个语句则给出要求 **x** 的阶乘, 必须先求 **x-1** 的阶乘, 然后将所得的解乘以 **x** 的规则。

如果我们提出问题

```
& all (x: fact (2 x))
```

则 **prolog** 系统将给出答案

```
x=2
```

**prolog** 是怎么求出这个答案的呢? 首先, 它将问题的目标与库中的事实和规则逐一进行比较, 把问题与第一事实比较时, 发现虽然 **fact** (相同, 但 1 和 2 不同, 所以匹配失败。当问题与第二事实比较时, **fact** (仍然相同; **x** 和 2 虽然不同, 但因为 **x** 是一个尚未实例化的变量, 所以匹配仍算成功, 而且 **x** 被实例化为 2; 之后又遇到两个未实例化的变量 **Y** 和 **X**, **prolog** 对此采取对策是让两变量共享存贮, 以后, 一旦两变量之一被实例化, 则另一变量也将被实例化为同值; 右) 匹配成功后, **prolog** 发出目标匹配成功信号, 然后问题中的目标 **fact (2, x)** 被分解为子目标继续匹配...

#### 2) 回溯

让我们通过下面的实例来具体说明回溯过程。假定库中有如下事实

```
Copper (brass)
```

含铜的合金 (黄铜)

```
Copper (duralumin)
```

含铜的合金 (硬铜)

# 数据处理语言的数据流模型

核工业西南物理研究院 叶高英

## 一、为什么是数据流模型

在受控核聚变实验中,有大量实验数据需采集和处理。目前,由VAX-11/750计算机上运行的DAS软件系统完成数据采集,同时将生数据(或称原始数据)实时图形显示并存储之。对数据的大量物理处理和变换则只能脱机进行。

由于实时性要求,DAS系统利用全程存储映像方法把所有采集数据存入一个数据文件(称为脉冲放电文件)中。该文件由近百道具有不同物理意义的数据块组成,数据处理时,必须根据

物理需要对其中一个单通道或通道族(目前最大通道族由八个单道组成)数据分别进行处理。这样就出现多个处理进程同时竞争同一数据源的矛盾。此时,若用传统控制台操作方式控制处理进程运行,则多有同时竞争打开文件读取数据失败的情形发生。同样,由于等待输入数据的原因,可导致后续各处理环节不能执行。

进一步考察数据处理的一般过程,本质上系对数据进行接连加工和变换。典型地,从脉冲文件读出数据后,首先进行平滑或数值滤波,排除

Copper (gunmetal)

含铜的合金(炮铜)

Copper (german-silver)

含铜的合金(锌白铜)

tin (solder)

含锡的合金(焊锡)

tin (gunmetal)

含锡的合金(炮锡)

tin (white-metal)

含锡的合金(白合金)

tin (fusible-alloy)

含锡的合金(易熔合金)

欲求所有即含金又含锡的合金,可通过询问:

& all (x : copper(x) & tin(x)) ✓

根据PROLOG搜索策略,系统首先考虑Copper(x),把它作为一个子目标进行匹配。Copper(x)首先与Copper(brass)匹配成功,并使得x实例化为brass然后匹配目标的第二个条件,并将x用brass代替即

tin (brass)

由于库中tin的关系中没有与tin(brass)相匹配的句型,因此tin(brass)匹配失败,其结果导致回溯,使得x成为自由变量,系统重新匹配Copper(x),于是Copper(x)与Copper(duralumin)匹配成功,x又被实例

化为duralumin,同样由于tin(duralumin)不能被匹配,再次产生回溯。第三次匹配Copper(x)的结果使x实例化为gunmetal,产生新的子目标tin(gunmetal)...

因此,一个prolog程序的执行过程,实际上是一个搜索,匹配和回溯的过程。在进行匹配过程中,变量可与任何项进行匹配,并使变量具有该值(利用共享存储),一旦变量在匹配中得到了一个值,则在匹配下一条件时该变量以该值代入,当进行回溯时,该变量又将释放它的值,成为一自由变量。注意,这种变量的传递只能在同一层合取条件层上进行。如要越层传递变量实例化值,只能通过全程变量进行,所谓全程变量是指那些同时出现在一个句子头部和体部的变量。

以上简单的叙述了prolog证明机制。有关更详细的内容以后继续介绍。

## 参 考 文 献

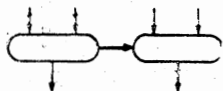
- [1] K.L.Crark, F.G. McCabe,  
《Micro-PROLOG, Programming  
in Logic》
- [2] 《LPA Micro-PROLOG》
- [3] IBM-PC 计算机人工智能语言  
—GCLISP 和 Micro-PROLOG
- [4] 逻辑程序设计语言 PROLOG  
—基础、原理、实现和应用

数据采集时的各种干扰信号,然后利用有关算法进行各种物理数学变换,直至得到要求的结果,并将之以图形曲线、曲面或其它形式输出。

这样一个处理过程,可以用有向图完全表示出来,图的结点为处理站,它对应于上述调用算法进行数据变换。这里所说的算法是一些专门编写的物理变换程序,目前国内外已编写出大量的这类程序模块(常称为 Code),将之存入程序库中,根据需要选择有关 Code,提供必要输入参数,即可实现物理变换的目的。另一方面,图的弧表示数据流,进入每个结点的弧对应于结点输入参数,同样离开某个结点的弧对应于该结点的处理输出,一般地说它流入后续结点。这就是一个完整的数据流模型。

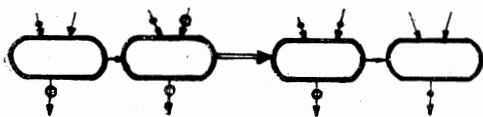
## 二、基本模型选取

数据流模型有两种探讨途径。一种是数据驱动数据流,其工作原理可图示为:



其中带实点的弧表示数据流经该弧到达所指结点。因此,这种模型表示:当一个结点所需输入数据全部计算出来,则该结点处理数据并产生相应输出数据。

另一种命令驱动数据流模型,图示如下:



这里带圆圈的弧表示输出请求(即命令)。由图可见,当某一结点有一输出请求时,若其输入弧中有数据未具备者,则在该弧生成一个请求命令,直到所有数据均到达后,便产生处理输出数据。

从语言文法结构观点看,数据驱动数据流模型系值调用机制,而命令驱动数据流模型则对应于惰性求值机制。虽然后者在语言性能方面综合了值调用与名字调用的优点。除其弊端,然而由于我们的语言专用性强,文法论域简单,因此选

取数据驱动数流模型作为语言基本模型。语言实现系统先行计算数据处理算法过程的全部输入参量,然后进行数据加工处理,输出处理结果。

## 三、语言基本结构

基于前面讨论我们知道语言定义主要由两个部分组成,即对结点及其相关输入输出定义和对数据对象的描述。下面将分别讨论之。

在我们的数据处理语言 SHY 运行环境中,由于有前面提到的程序库支持系统以及功能完备的图形处理程序包存在,使得结点定义处理功能在形式上变得极其简单,仅用一个简单语句:

**Code** <库程序名> (<参数表>)

其形式与通常程序设计语言中的库过程调用语句十分相像。

定义结点输入、输出流分别使用语句:

**take** <变量表> [**from**] **stream** <名字>

**Send** <变量表> [**to**] **stream** <名字>

其中,两个语句中的关键字 **from** 和 **to** 并无语法作用,我们用方括弧围起来表示可省。用户若使用它们,可以增强程序可读性。

很显然 **take** 语句的变量表中应包括从已经存在的流中读取对应结点输入参数的有关数据,而 **Send** 语句则将全部计算结果送入输出流。语言中通过给数据流取名字、按名存储并提供后续结点使用该数据。

除了上述一般处理结点外,SHY 语言用两种特别形式定义两类特殊结点,即数据源和汇结点。在我们的应用中源系指从脉冲文件读取数据,汇系指将最后处理结果数据以硬拷贝或作图形式输出。此时对应语句不再使用关键字 **stream**,而是分别用 **source** 和 **sink** 代换之,而且其后不再需要名字定义,代之以相关特征信息描述。比方对于数据源应指明放电序号、通道名等,对于汇则为图形输出设备及座标选取等有关参量。

由以上讨论,可见 SHY 语言作为数据处理高级操作语言的数据流特性。语言中用来对从流中获取的变量进行进一步计算,以及对使用同一算法过程处理多输入流的重复计算均有相应结构实现,本文不再赘述。

SHY 语言本质上仅处理一种数据即流数据。流是数据种类,而不是指构成流的具体数据的类型,这种类型由生数据形式完全确定,而无需在语言中重新说明。因此 SHY 本质上是一个弱类型语言。但流数据分量的数据结构定义是必须的,在语言中也是允许的。在数据处理应用中使用最频繁的数组便是一例。我们前面讨论的 take 和 send 语句中的变量表中,既可以出现简单变量,也可以是结构变量。例如

```
take A(1—100), B, C(1—200) from  
stream, S1
```

的作用是将 A 及 C 定义为数组,并且从数据流 S1 中顺序读取 100 个数据值给数组 A, 1 个数据值给变量 B, 200 个数据值给数组 B。应该强调指出,这里既定义数据结构(数组)同时又定义了它们的值。而数据流语言的定义性只允许访问已经定义了值的数据结构,而不允许重新定义之,因此上面语言语句中不可出现如传统语言的下标变量赋值那样的用法。

至此,我们已经介绍了语言 SHY 的基本模型及核心部分。一个完整的程序由结点定义序列组成,而每个结点定义可以作为交互应用中一个基本程序段落和操作单位。整个程序结构十分清晰。

#### 四、语言属性

SHY 是一个高级专用数据处理操作语言,选取数据流模型作为其设计基础,使之具有许多优良属性,这里简述如下:

##### 1. 定义性

SHY 基于一种定义性探讨方向,如同函数式语言和关系语言一样,程序给出的不是要执行的命令,而是给出一个定义性说明。完整程序定义一个数据流网,一个程序段落定义其中一个网结点及其相关部分。因此程序本身并不指定执行顺序,不同的语言实现策略可以使同一程序有不同计算顺序。唯一强制的顺序关系是获得输入数据的先后次序,但这是算法本身的特性,而不是语言属性的范畴。

##### 2. 交互性

交互是新一代计算机应用环境为方便用户而

提供的重要计算机使用手段,也是软件系统、特别是应用软件设计所追求的一个重要目标。SHY 语言设计中充分考虑交互操作应用环境,以一个网结点的构造为程序逻辑单位,非常适合在交互环境下完成复杂的数据处理操作。

##### 3. 并行性

SHY 语言具有良好的并行实现和运行特性。一般说来一个结点的多个参数若来自不同输入流则完全可以并行计算这些参数。例如若多个数据源拷贝存在,则可以并行获得源数据,同样一个结点的输出流可以有多个拷贝送往不同后续结点,从而可能启动不同结点并行执行其处理功能。完整数据处理程序的并行执行性是显而易见的。

##### 4. 有限副作用

如同其它数据流语言一样,语言 SHY 具有“作用局部化”属性,语言中不允许重复定义变量的值,所有计算均面向值,具有“引用透明性”,因而语言计身并无副作用。但是结点处理算法过程采用系统库中的库过程及专用物理 Code,这些程序大部分系用 FORTRAN 语言书写,是有副作用的。因此在 SHY 实现时,对这些 FORTRAN 子程序的输入输出参数进行精心安排,将其副作用局部化于这些子程序本身,并不影响 SHY 语言的无副作用属性。

#### 五、结论

通过以上讨论可以清楚地看到,以数据流模型为基础的语言 SHY,克服了传统命令操作语言的诸多弊端,具有新一代说明性语言的许多高级属性。另一方面与交互环境中经常使用的菜单选择方式比较,语言 SHY 直接用作程序设计工具,而菜单方式则只能为完全设计好的程序选择访问路径,因而后者并不具有直接程序设计功能。SHY 语言的确具有较高的学术和实用价值。

《计算机病毒大观》及其《续集》尚有余书可供邮购,每套(两本)14元(含邮费),款到发货。

本刊



# IBM-PC 机总线扩展器的设计

中国科学技术大学 王砚方 庞文宁

**摘要** 本文介绍一种 IBM-PC 机总线扩展器的简单设计方法。此法扩充了 4 至 8 个 PC 机总线 I/O 插件槽口, 用户在做硬件接口调试工作中, 不需经常开关主机电源, 为在 PC 机上做硬件开发工作的用户提供了一个良好的工作环境, 方法简易, 有一定实用意义。

## 一、引言

随着 IBM 个人计算机在我国 的广泛使用, 在许多应用领域中, PC 机的用户需进一步做一些与本学科或具体工作性质有关的硬件开发工作。例如: 微机控制系统, 数据采集, 数字图象信号处理等各种接口板的设计。但是, 我们大家都知道, 凡在 PC 机上做硬件开发工作的用户, 都会遇到一个普遍的问题, 就是在做具体硬件调试工作中, 要经常开关主机电源, 这样给做硬件开发工作的用户带来诸多不便。另外, 由于 PC 机只带有五个系统总线插件槽口, 即使是在最小的磁盘系统中, 也有两个被机器本身占用, 一个用于显示器适配器, 一个用于软磁盘适配器。剩下的三个槽口也根据不同需要被相继占用。例如, 内存扩充, 串行端口等。当然, 某些兼容机或 PC/XT 机的总线 I/O 插件槽口增加到了 8 个, 或者用户使用某些多功能卡也可以节省出 I/O 槽口。但是在计算机内做某些调试工作, 还是很很不方便的。此外, 对于某些需要多通道的用户, 还是显得系统总线插件槽口不足, 需提供另外附加的总线插件槽口, 同时扩展总线插件槽口配有独立的供电电源, 使在调试工作中不需经常开关主机电源。因此, PC 机总线扩展是一定的实用意义, 总线扩展将使在 PC 机上做硬件开发工作的用户免受空间和功率方面的限制, 给调试工作也带来了方便。

PC 机总线扩展设计, 文献 [5] 进行过一些讨论, 文献 [6] 给出了 PC 机系统总线扩展部件的框图。文献 [5—6] 讨论了一般的情况, 其中大部分结果与本文是一致的。

## 二、基本设计思想

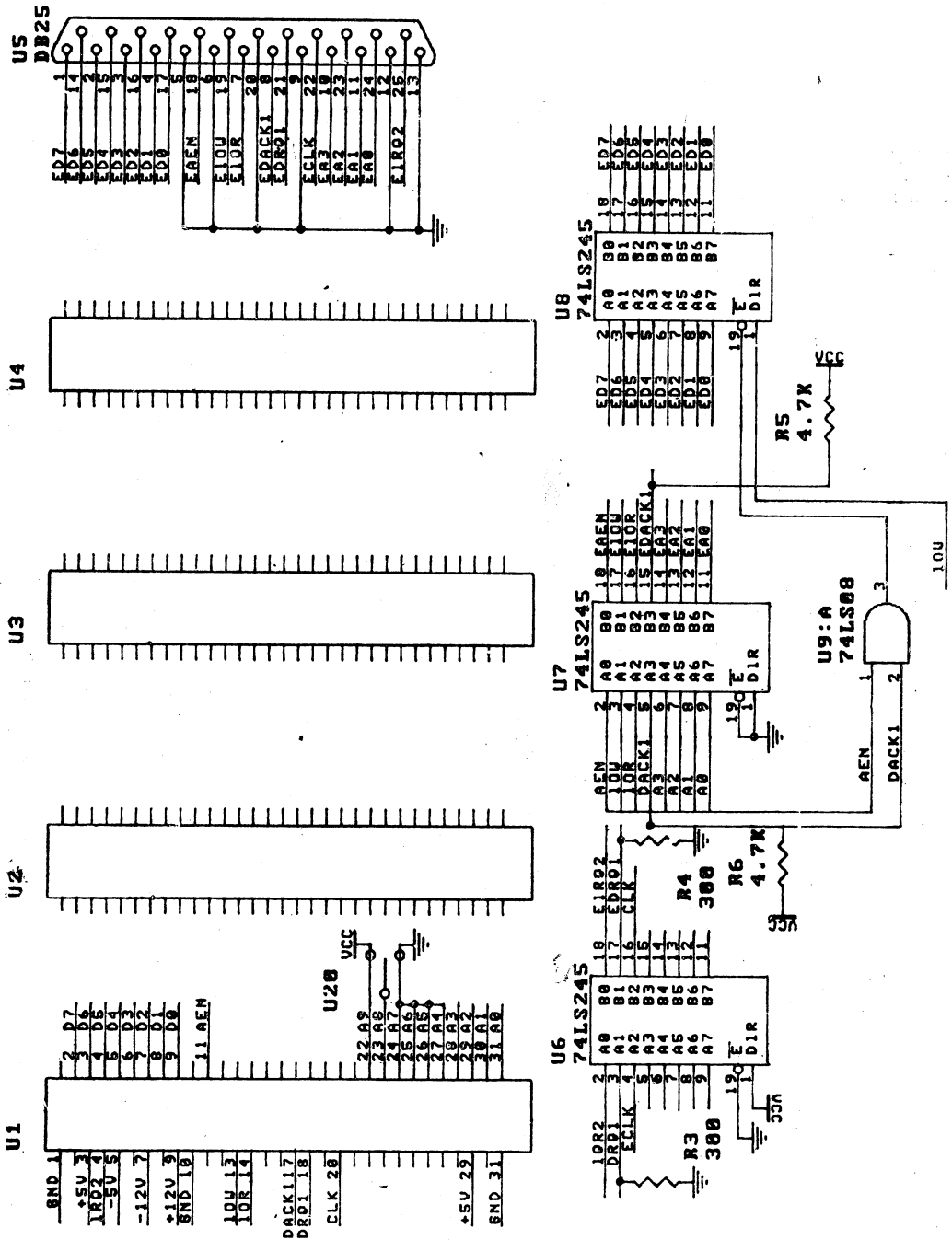
PC 机总线扩展器电路原理图如图 1, 图 2

所示。

PC 机总线扩展, 通过在一个独立的部件上增加扩展槽而加强了 PC 机的性能。该扩展器包括一块插入 PC 机 I/O 槽口插件板, 该板称为扩展板, 主要用来驱动和接收扩展 I/O 槽口的总线信号, 并带有 I/O 端口地址译码器, 一块带有接收器的 4 至 8 个扩展 I/O 槽口电路板。总线信号传输经 25 线带状电缆, 电缆两端均连着 25 脚的 D 型阳插头。

一般来说, PC 机总线上的大多数信号都是比较容易处理的。但是根据传输线理论, 总线传输的信号线上存在反射和信号衰减以及噪声干扰。对时钟频率在  $1 \sim 10\text{MHz}$  范围内通路又被限定在一块印刷板内进行逻辑电路设计, 传输线的影响一般可以忽略。但是信号用总线从一块板传送到另一块板时, 就有显著的传输线效应, 使总线上的信号发生畸变, 产生信号的圆尖峰。因为, 总线信号走线过长会增加额外电容, 使得驱动电路的容性负载随之增加。通常, 电容值大于 200 皮法时, 对总线信号造成的失真将导致整个系统工作不稳定。此外, 出现在板到板的连接中的噪声源之一是串扰。串扰指的是相邻的信号在这条线上所产生的噪声。相邻线中的信号通过电感和电容的作用而相互耦合, 这种耦合随着相邻线长度的增加而增加, 而随着两线的间隔增大而减少。采用有源端负载, 是目前用于减少总线上噪声的主要技术。有源终端就是通过接到公共电源的匹配阻抗来端接所有总线的方法。虽然由于传输线上的分支会在线上引起反射, 但是传播到这条线末端的反射会被有源的终端负载所吸收。通常用于总线系统的驱动器都是具有低、中及高阻抗三种操作方式的三态驱动器。当所有驱动器





图二 接收器电路原理图

断开时, 一个没有终端的总线是处在向上或向下浮动以及仍然停留在上次操作后的电位的自由状态, 这就使得在驱动器将要对这条线充电时, 它必须充到  $V$  伏。用总线传送信号时, 必须把对这条线的充电延迟考虑进去。对有源终端负载来

说, 当所有的驱动器断开时, 总线是处在逻辑阈值上, 从而使得线上的电位变化不会大于  $V/2$ , 这样, 对长线的充电延迟减少了大约一半。

根据以上所述, 我们在设计 PC 机总线扩展器时, 采取了以下两点具体做法。

1. 采用对 PC 机总线局部扩展。这是因为考虑到: 虽然 PC 机总线 I/O 槽口有 62 根线, 但是有部分信号线已让系统本身占用, 用户在做各种接口设计时是无法使用的。例如: 中断信号线  $IRQ_0$  系统定时器输出,  $IRQ_1$  键盘扫描代码中断,  $IRQ_3 \sim 7$  也均被机器系统使用, 提供给用户使用的保留中断源只有  $IRQ_2$ 。同样 DMA 信号除通道 1 是为用户保留的 DMA 通道外, 其余 3 个通道为机器系统使用。所以扩展器的总线传输只用了 19 根信号线。它们是: 数据线  $D_0 \sim D_7$ , 地址线  $A_0 \sim A_3$ , 读写线  $IOR, IOW$ , DMA 请求和响应线  $DRQ_1, DACK_1$ , 中断请求线  $IRQ_2$ , 时钟  $CLK$  和允许寻址线  $AEN$ 。并且在扩展板上将地址信号  $A_0 \sim A_4$  译码, 生成  $IOAEN$  信号, 该信号通过用允许寻址线  $AEN$  代替传输, 这样就无需将  $A_0 \sim A_4$  线传输, 用户可根据自己对 I/O 端口地址范围的需要, 选择扩展 I/O 槽口上的跳线给  $A_0 \sim A_2$  线某一固定电平, 同时选择相应的译码电路。本设计中的译码电路采用一片通用可编程逻辑阵列 GAL 器件。当扩展 I/O 槽口上的  $AEN$  信号为有效时, 则表示选中扩展 I/O 口上某地址范围内的 16 个 I/O 口地址。在我们的设计中, 分别设置在  $200H \sim 20FH$  和  $300H \sim 30FH$  两个地址范围。虽然设计方案只给出了 16 个 I/O 口地址, 但是这 16 个 I/O 口地址在用户设计接口时经译码可生成 32 个端口地址, 在通常大多数情况下, 这些端口地址是足够用的了。这种做法是使被传输的信号线减到最低程度, 以便提高信号传输的可靠性。

2. 对总线传输信号进行一些适当终端匹配和信号驱动。采用把从 PC 机总线来的信号在扩展板上经过 74S24x 系列的器件驱动和匹配, 再送到带状电缆。驱动电路的主要功能是减少传输线电容的影响以及信号反射, 使之不致造成系统单元总线的延迟和信号失真。在扩展 I/O 插件槽口电路板上, 以 74S24x 系列器件为接收器, 接收到的信号再次被增强, 然后去驱动附加的 I/O 槽口。此外, 还需考虑电源去耦, 用 0.01 微法的电容器跨接在 +5V 和地之间以便为总线驱动器

### 三、结论

采用上述介绍的方法, 实现了 PC 机总线插件槽口的扩展。经过长时期使用, 信号传输可靠, 系统工作正常稳定, 取得了满意的效果。该扩展器不仅适用于 PC 机, 也适用于 PC/XT 机和 PC/AT 机。本设计方法在硬件方面, 简便易行, 成本价格低, 一般用户均可自己制做。它与市场出售的扩展机箱相比较, 具有较高的性能价格比。当然, 该扩展器也存在某些局限性。此外, 在扩展板上的译码电路可用一般 TTL 器件组成, 并非一定要用 GAL 器件。

### 参考文献

- [1] Kraus, J.D: "Electromagnetics", McGraw-Hill Book Company, New York 1953
- [2] SKilling, H.H: "Electric Transmission Lines", McGraw Hill Book Company New York, 1951
- [3] "通用逻辑阵列 GAL", 周永钊 编译, 中国科学技术大学出版社, 1989
- [4] "数字与通讯网络中的传输线", 栾毓敏译, 科学出版社, 1982
- [5] "IBM-PC 微型机接口", 孙承鉴译, 北京科学技术出版社, 1989
- [6] 《XT 硬件技术手册》, 康宝祥译, 辽宁省电子计算机学会出版, 1984

\*\*\*\*\*

### 新编笑话一则

远古人阿盘、阿妈夫妇偶然穿过时光隧道, 来到现代。一位记者迎上前去, 询问道: "跨越万年时空, 看透世事因果, 请问要想提前成功, 早年得志, 有无良策可循?" 阿盘沉吟了一下脱口而出: "自古以来只有两法: 一是你爹妈有钱有权, 二是你配偶的爹妈有钱有权。" 阿妈瞅了阿盘一眼: "老头子莫信口开河!" 阿盘拍了一下胸脯继续说道: "这第三法可千真万确: 你善操电脑!"

据《热讯》改编

# MC68488和Intel8048实现IEEE488接口

北京9203信箱14分箱 吴剑勇

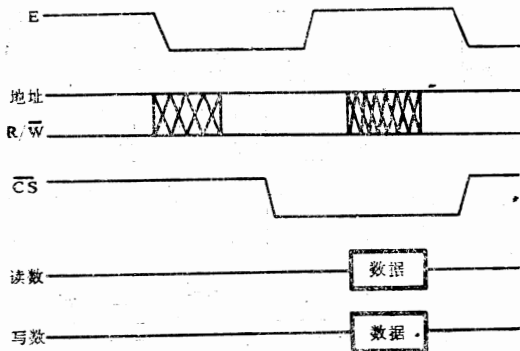
**摘要:** MC68488芯片是一种具有控制功能外九种接口功能的LSI电路,已广泛用作智能仪器的IEEE-488接口,本文介绍了MC68488与Intel8048组成的IEEE-488接口,它具有听、说功能;文中详细介绍了该接口的实现方法及线路图。

## 一、MC68488与Intel8048 硬件线路

由于MC68488是Motorola公司为MC6800系列CPU设计的专用芯片,而Intel8048是Intel公司的产品,因此二者存在着硬件不完全兼容的问题。

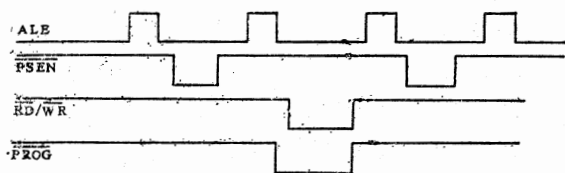
### 1) 时钟

从MC68488与Intel8048的技术手册中可发现这二者硬件兼容的关键在时钟信号E上。MC68488的时钟E是一个外部输入工作时钟,频率小于1MHz。它的时序状态如图一,由图可知 $\overline{CS}$ 片选有效时,时钟E为高电平,即E高电平有效。



图一 MC68488的时序状态图

从8048的技术手册中可以得到,8048的时钟输出信号ALE是晶振频率的1/15。而8048的晶振一般小于8MHz。所以ALE小于1MHz,显然频率范围与MC68488兼容。根据8048的技术手册可得其时序状态图如图二。由图可知,ALE的有效电平为低电平。由此可得,将8048的ALE信号经过非门就完全与MC68488的时钟E兼容了。



图二 8048的时序状态图

### 2. 远控信号 $\overline{REN}$

由MC68488的手册可知,当 $\overline{REN}$ 为低电平时,MC68488处于远控状态。但在实际使用MC68488系列芯片时,发现此时 $\overline{REN}$ 是高电平有效,故电路中 $\overline{REN}$ 接高电平。

图三表示了MC68488与Intel8048组成IEEE-488接口的实际线路图,该线路已经在我所生产的高精度转速校准仪中使用。

## 二、接口的软件框图及注意事项

在MC68488的编程中必须注意,因为地址寄存器 $R_4W$ 的内容不受软件复位的影响,而在硬件复位之后,编程 $R_2W$ (地址方式寄存器)之前, $R_4W$ 内容为00,如果这时写入 $R_3W$ 中的 $b_7=0$ ,结束复位状态,则488接口可能响应地址为00的寻址命令。为避免出现这种情况,应在退出复位状态前把正确的地址写入 $R_4W$ 。

此外在讲功能中数据的结束信号是由软件在发送最后一个字节前控制 $R_3W$ 的强迫结束位 $feoi$ ( $b_5=1$ )来完成,即488处于TACS态时使 $\overline{EOI}$ 输出低电平,然后发出最后一个数据字节,使 $\overline{EOI}$ 线返回高电平。

# 高抗干扰精密 V/F 式 A/D 转换芯片 与 APPLE II 微机接口技术

上海师范大学物理系 杨兆庆

## 一、前言

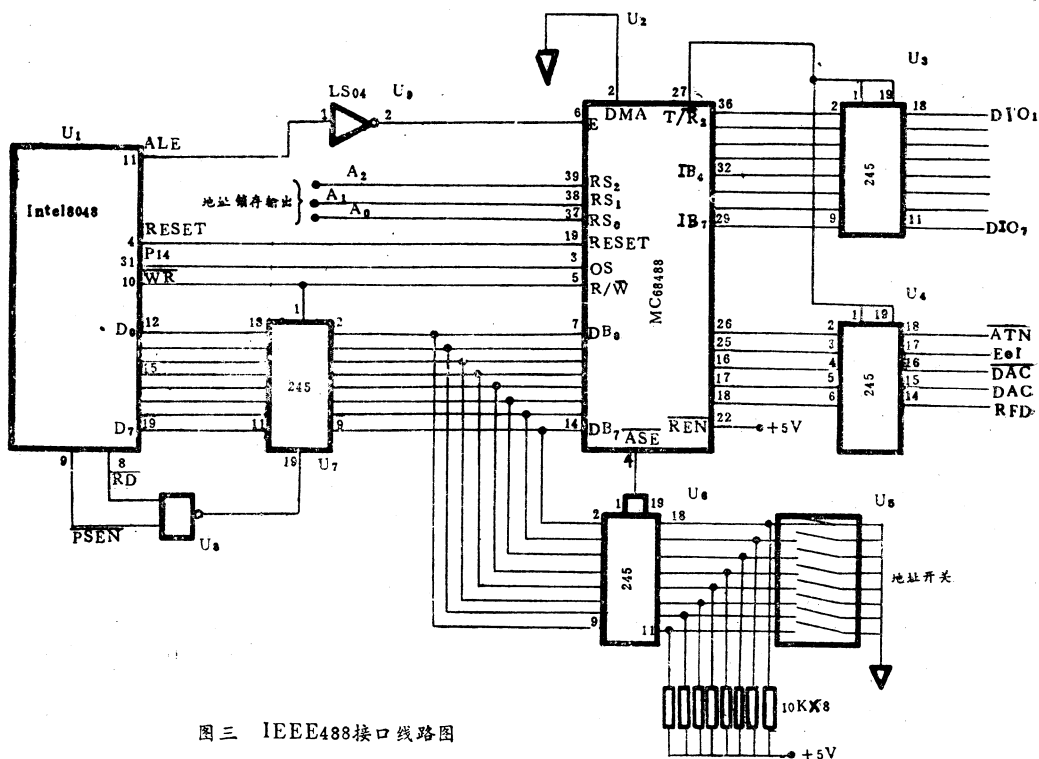
A/D 转换技术是由数字测量的要求而产生和发展起来的,特别是微型计算机的迅速普及更加速了这一技术的进展。现在,人们不仅追求 A/D 转换的速率,而且,也希望它有较高的转换精度。如今能够批量供应市场的 A/D 转换器的最高分辨率已达16位二进制,转换时间已达几十纳秒(例如 ADC1140),但遗憾的是,随着转换器分辨率的提高和转换速率的加快,它的价格也就急剧地增长,一般一片12位的 A/D 转换芯片其价格就高达几百元。大家知道,在许多实际应用过程中,被测的物理量往往变化比较缓慢,因此, A/D 转换的速率就不是主要问题,而主要的是测量要具有较高的精度。在这种情况下

下,如果选用昂贵的转换芯片就显得不太经济。这时,我们就可采用 V/F 式 A/D 转换芯片。一般说, V/F 式转换芯片的转换速率较低,但它却有较高的转换精度和分辨率。例如, V/F 式 A/D 芯片中的 LM331 其分辨率就优于12位逐次比较式 A/D 芯片,并且,它的抗干扰能力也强,但价格仅是后者的十分之一。因此,在速度要求不高的情况下是一种很有发展前途的微机 A/D 转换接口电路。

本文主要介绍一种 V/F 式 A/D 转换芯片 LM331 与 APPLE II 微机的接口方式。

## 二、V/F 式 A/D 转换芯片 LM331

LM331 是美国 National 公司生产的 V/F 式 A/D 转换芯片。它作为一种简单的廉价电路



图三 IEEE488接口线路图

十分适合于用作A/D转换器、精密F/V转换器、长时间积分器、线性频率调制或解调以及许多其它功能。当用作V/F转换器时,其输出电压为频率精确地正比于输入电压的脉冲链,它们之间的关系由下式确定

$$f_{OUT} = \frac{V_{IN}}{2.09V} \cdot \frac{R_S}{R_L} \cdot \frac{1}{C_I R_I}$$

上式中,  $V_{IN}$ 表示输入电压;  $f_{OUT}$ 表示输出频率;  $R_S$ 表示  $R_0 + W_S$  ( $W_S$ 为增益调节用); LM331具备V/F转换技术的所有固有的优点,很容易应用于标准V/F转换器应用场合,而且,它还使用了新的温度补偿能隙基准电路,故在整个工作温度范围内和低至4.0V的电源电压下都有较高的精度。

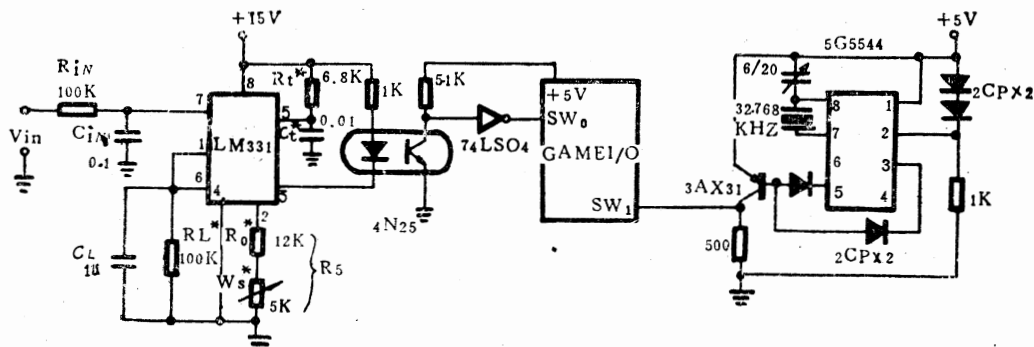
LM331在输入电压  $V_{IN}$  从0~10V变化时,其输出频率将在10Hz~10KHz内变化,其线性度为0.03%,当满度输入时,输出频率为

10KHz,其分辨率高达0.01%,它较三位半A/D芯片MC14433高出5倍。

LM331的主要缺点是转换速度较慢,若想要提高速度,可适当减少检测时间(即缩短采样时间)。

### 三、LM331与APPLE II微机的接口方式

LM331与APPLE II微机的联接可通过APPLE II微机的GAME I/O插座的开关量输入端来实现。在图1的接口电路中,LM331的输出通过光电耦合器4N25送入GAME I/O插座开关量输入端  $SW_0$ 。当一系列与电压成正比的脉冲链输入到  $SW_0$  端时,计算机就应在规定的取样时间内对脉冲链进行计数。图1电路中取样时间为1秒,它是由时钟集成电路5G5544输出的秒脉冲提供。



\* 使用低温度系数的稳定元件  $C_I$  最好采用聚苯乙烯、聚丙烯电容。

图1

众所周知,在许多应用场合下,我们常常要研究诸如温度、压力等物理量与时间的关系,现在,由于计算机的  $SW_1$  端接有时钟集成电路5G5544,因而,我们还可利用秒脉冲来设计一个精密的石英电子钟,这样对于我们研究问题带来了极大的方便。我们只要在LM331的输入端配置相应的传感器和放大电路(例如:LM134作用温度传感器,用超低漂移ICL765作放大器)就能方便地得到被测物理量与时间的关系。

### 四、转换软件框图

图1所示的LM331与APPLE II微机联接

后的A/D转换软件框图如图2所示。在图2的框图中,由于采样时间为1秒,因而计算机所得的脉冲个数除以1000就是与之对应的电压数,这样,LM331不仅满足了精度的要求,而且对于问题的处理也变得十分方便。但需要注意的是,由于计算机在执行程序时总需要一定的时间,因此,在应用GAME I/O作为V/F转换器的接口时,须对LM331输出的脉冲链的频率有一定的限制(即时LM331的输入电压有一范围),否则,将会产生误差。笔者曾作试验,在输入电压为8伏范围内(即脉冲链频率为8KC)



# DRNEST专家系统构造工具的设计与实现

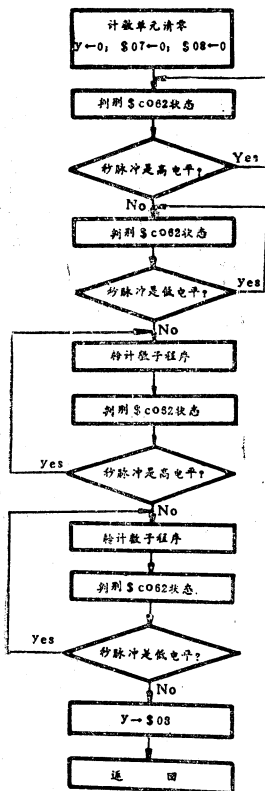
湖南大学 杨润生 张小衡

**摘要:** 本文介绍了作者研制的一个诊断型 DRNEST 专家系统构造工具。该系统由 DOCTOR ES 用户接口、CLEVER 推理机、DKB 知识库、TOOL-BOX 工具箱、DNLPS 自然语言处理子系统和 DKBMS 知识库管理子系统等构成, 能提供纯英文、纯中文和中英文兼容的工作环境。由于设计了良好的用户接口, 提供了解释信息, 使用极为方便。

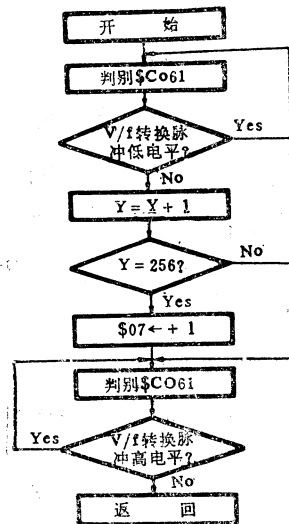
专家系统的研究起始于 60 年代后期。美国斯坦福大学 60 年代的 DENDRAL, 70 年代的 MYCIN 等专家系统的研制成功, 极大地鼓舞了这一研究领域的信心。80 年代以来, 专家系统走出了大学和研究机构, 许多著名的公司已成为开发专家系统的主角。他们积极把专家系统用于

工业项目, 收到了明显的效益。

专家系统是一种既具有高性能又十分复杂的系统。每一个用户, 如果要为自己的应用构造一个特定领域的专家系统, 一般都必须从最基本的模块开始, 他们必定会相互重复地编制功能大同小异的模块。这样不仅会浪费设计人员的时间和



A/D 转换软件框图



计数程序框图

伏, 则可对脉冲链作二分频处理。

V/F 式 A/D 转换芯片 LM331 与 APPLE II 微机的连接, 由于通过了 GAME I/O 的开关量输入端作为输入接口, 因此, 无需任何复杂的硬件接口电路, 并且, 值得一提的是, 如果在 LM331 的输入端接一选择开关或继电器, 利用 GAME I/O 的开关量输出端来进行译码或接通, 那么, 一路 A/D 转换就可起到多路测量的作用。

都可获得较高的转换精度, 这个输入电压范围对于我们来说已经是足够了。如果输入电压超过 8

精力,而且研制周期会拖得很长。这就促使人们去研制开发专家系统的构造工具和环境,以加快这种复杂系统的构造速度。专家系统构造工具就是解决这一问题的理想工具。例如,美国的MYCIN系统是一个医疗诊断系统,后来发展到由MYCIN系统产生了一个EMYCIN系统。抽掉MYCIN系统中具体的医学知识,使它变成一个保留推理系统,知识库和知识库管理系统等一般结构的空框架,就成为EMYCIN系统。它是一个专家系统的母系统(即专家系统构造工具),只要在这个框架中加进不同领域的具体知识,就可以构造出不同领域的专家系统。

### 一、系统设计

DRNEST (Doctor Nest) 是我们用 Prolog 语言在通用的 IBM-PC 机上研制出来的小型诊断类专家系统 (ES) 构造工具。该工具支持专业人员构造自己的专家系统,在 DRNEST 上建立起来的专家系统和 DRNEST 本身可在一切支持 IBM-PC DOS 或者 CCDOS 的机器上运行。DRNEST 已具备现代专家系统构造工具的主要功能,能分别为用户提供纯中文、纯英文和中英文兼容的工作环境,由于有良好的用户接口,使用极为方便。

#### (一) 系统结构

DRNEST 的系统结构如图1所示。它由 DOCTOR ES 用户接口、CLEVER 推理机、DKB 知识库、TOOL-BOX 工具箱、DNLPS 自然语言处理子系统和 DKBMS 知识库管理系统等六部分组成。人类专家或知识工程师通过 DKBMS 提供的“ES构造者用户接口”,可方便地输入所需的知识,建立起相应的知识库。知识库与 ES 用户接口、推理机、自然语言处理子系统和工具箱等模块组成一个整体,便构成了一个既能推理诊断、又能提供解释信息的完整的专家系统。

#### (二) 系统特点

与其它 ES 构造工具相比,DRNEST 具有如下 3 个特点:

1. 引入面向对象的程序设计方法,支持一种称之为“诊断图”的知识结构,适用于整个诊

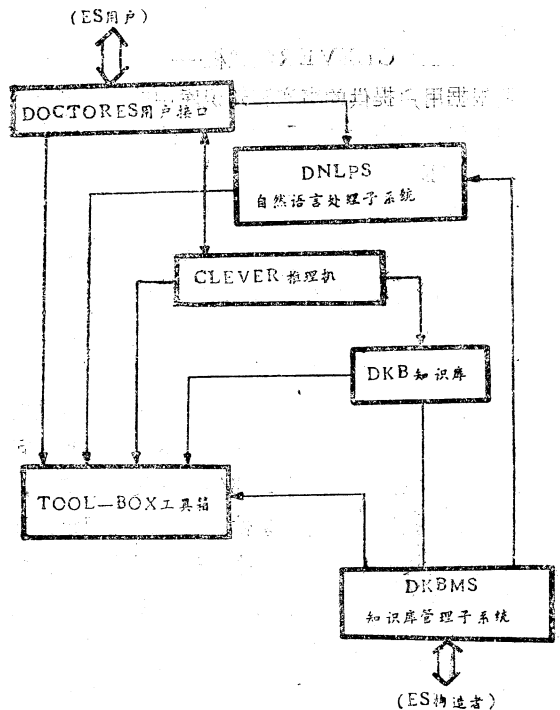


图1 DRNEST的系统结构

断类领域;

2. 系统中的DNLPS具有一定的自然语言理解、自然语言生成和英中机器翻译功能,能为用户提供中文、英文和中英对照的工作环境;

3. 在DRNEST的支持下,用户可采纳或不采纳可信度(CF)技术。如果用户需要此技术,他既可从系统中备有的几种CF计算方法中任选一种,也可用一般的代数表达式自行定义。

### 二、系统实现

(一) DOCTOR ES用户接口——它是每个在DRNES上建立起来的专家系统的统一用户接口,采用人机交互的工作方式。其主要功能是在启始阶段,显示ES的首屏信息(包括ES名称,研制者名称和竣工日期等),登记诊断任务;在运行阶段,ES进入推理分析阶段,该接口负责用户同系统其他组成部分之间的通信和信息转换工作,并能按照用户的要求作“how”或“why”型解释;在结束阶段,该接口负责询问用户对ES工作的意见和评价,建立反映本位诊断情况的“诊断文件”,为专家系统构造者提供

反馈信息等。

(二) CLEVER推理机——它的主要任务是根据用户提供的事实和知识库中的知识进行推理分析,找出被诊断系统的故障位置。在研制 CLEVER时,我们采用了面向对象的设计技术,把被诊断的系统看成是一个由多级部件组成的系统,各部件由相应的“对象”表示,部件对象之间通过收发消息进行通信。

我们采用“诊断图”来表示诊断问题,如图 2(a)所示。诊断图是一个有且只有一个结点无入射边的无环有向图,其结点代表被诊断系统的部件(对象),也可视为 ES 指向该部件时的问题状态。每条边从一个母部件指向某一子部件,蕴含着相应的状态转换条件。初始结点规定为那个无入射边的结点(即整个被诊断系统),目标结点就是那些无出射边的结点。可见,诊断图蕴含着部件间的层次关系。诊断图通常以树的形式出现,称之为诊断树。非树的诊断图可通过复制结点转换为等价的诊断树,如图 2(b)所示。诊断图的初始结点和目标结点分别对应于相应诊断树的根和叶。

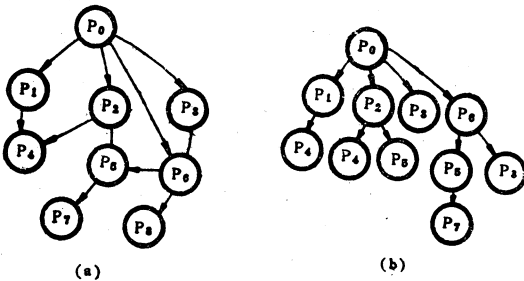


图2 诊断图和对应的诊断树

当 CLEVER 开始工作时,诊断树的根被视为当前怀疑对象,一个“pointed”消息被送往代表根的部件对象。根部件接到此消息后开始推理分析,当它在其子部件中找到一个可疑对象时,便向该可疑对象发送“pointed”消息,要求它完成下一步推理。依次逐步缩小可疑对象。如果某个部件能成功找到下一个可疑子部件,则向自己的母部件发“yes”消息;否则发“no”。母部件接到“no”后,便会在余下的子部件中寻找下一个可疑对象。直至找到故障点。在

DRNEST中,消息的收发是通过规则的匹配与回溯来实现的。

在每一步推理中, CLEVER除了寻找新的可疑对象外,还要完成可信度计算、推理轨迹的记录等工作。每步推理结束时, ES 将当前可疑对象显示在屏幕上,并允许用户有以下选择:

1. 命令专家系统作“why”型解释,此时,系统将显示最后执行的骨干规则中的解释文字。
2. 命令专家系统作“how”型解释,此时,系统将根据推理轨迹产生并显示自然语言解释信息、说明从根部件结点到当前部件结点的整个推理过程。
3. 命令专家系统继续推理,此时,当前部件将开始在自己的范围内寻找新的可疑子部件。
4. 修改事实库,这时 CLEVER 进入事实编辑状态,用户可方便地对原有事实进行增、删、改等编辑操作,专家系统将根据修改后的事实,重新进行推理诊断。
5. 命令专家系统停止推理,此时, CLEVER 将控制交给 DOCTOR 接口,由其作事后处理。

### (三) DKB 知识库

DKB 知识库由规则库、事实库和部件库三个子库组成。

1. 规则库——它是 DKB 知识库中的主要部分,包括人类专家所提供的推理性知识,这些知识由骨干规则、辅助规则和永久性事实表示。骨干规则在推理过程中起主导作用,每条骨干规则对应诊断图中的一条边,代表一步推理,整个骨干规则代表整个诊断图/树。骨干规则的格式为:

“(rule〈规则号〉〈母部件名〉) ( ?  
〈条件1〉〈条件2〉……〈条件n〉) (cf〈CF值〉)  
(reason〈解释文字〉) (pointed〈被指  
子部件名〉)”

其含意是:如果 ES 指向〈母部件名〉,且诸条件得到满足,则指向〈被指子部件名〉,这种因果关系可信度是〈CF值〉,关系存在的原因在于〈解释文字〉。

辅助规则和永久性事实表示诊断图之外的一些关系和事实,在推理中起辅助作用。

2. 事实库——主要用来存放ES用户提供的事实信息,反映受诊对象的各种观测情况,是ES进行推理诊断的事实依据。DKB中的各种事实和基本条件都以“pn型”(即是/否型)事实的形式表示,pn事实的格式是:

“positive/negative (<谓词><主语>)”

事实库还能起到“黑板”的作用,系统生成的一些中间信息都暂存于此,以备后用。

3. 部件库——它是一个由众多部件记录组成的关系数据库,记载着诊断树中各部件结点的情况,整个子库由系统自动产生,部件文件的格式是:

“(part <部件号><部件位置><部件名>)”

其中,<部件位置>表示该部件结点在诊断树中的位置或结点下标。因此,部件库既能简要地描述诊断图/树,又能在规则库和事实库之间起“索引作用”。部件记录中的<部件位置>提供了诊断树中该结点到根结点的路径,可用于解释推理,此外,<部件位置>还描述了部件间的层次关系,有利于采用面向对象的程序设计技术。

(四)DKBMS 知识库管理子系统——它负责知识库的建立与管理,人类专家或知识工程师可通过DKBMS提供的用户接口方便地建立自己的ES。DKBMS由DKBMS接口、DKBMS-1和DKBMS-2三部分组成。DKBMS接口也称ES构造者接口,其功能是:在启动阶段,负责用户使用权的检查、自然语言环境的确立和文件调入等预备性工作;在运行阶段,负责用户与DKBMS-1或DKBMS-2之间的通信和信息转化;在结束阶段,负责完成知识的调整和存贮,新知识的产生等工作。

DKBMS-1是DKBMS的主要部分,其作用是帮助用户完成六项工作:(1)设计ES首屏信息;(2)确定CF的计算方法;(3)建立和编辑骨干规则;(4)初始化知识库;(5)建立诊断对象集(可诊断范围);(6)设计系统钥匙(设定口令)。

DKBMS接口与DKBMS-1一起,便构成了基本型DKBMS。为了进一步加强DKBMS

的功能,我们还建立了DKBMS-2,提供对辅助性规则和永久性事实的处理等功能。基本型DKBMS加上DKBMS-2,便构成了“扩展型”的DKBMS。

(五)DNLPS自然语言处理子系统——它负责简单的自然语言处理工作,由接口、字典和两个功能模块组成,DNLPS的两个主要作用是:

1. 自然语言理解与生成:DNLPS既能将简单的自然语言转换为系统的内部知识;又能根据其它部件给定的规则、事实和记录等产生自然语言,例如,根据CLEVER提供的推理轨迹记录产生解释文字等。这一功能主要是根据自然语言与系统内部知识的对应关系来实现的。

2. 英中机器翻译:ENLPS还能用自己建立起来的字典作英中机器翻译。翻译范围包括DRNEST中所有显示的词句。当碰到无法处理的词句时,DNLPS会向用户请教并随即建立起相应的字典记录,下一次它便会使用这些词句。因此,DNLPS也可作为一个单独的“软件汉化器”使用。由于DRNEST具有英文原版,这种翻译功能使系统能为用户提供中文、英文或中英文兼容的工作环境。

(六)TOOL-BOX工具箱——在研制DRNEST的过程中,我们发现有许多prolog未能提供的基本功能,都是各组成部分共同需要的,为使系统更为紧凑和便于管理,我们为这些功能建立了相应的谓词过程,并集中在TOOL-BOX中以便共享。这些工具谓词功能甚广,包括表编辑,子句移动,带条件读等等。它们不仅在DRNEST中具有通用性,其中有的还可用来建造其它软件系统。

## 参考文献

- [1] Elaine Rich, Artificial Intelligence, McGraw-Hill Company, 1983.
- [2] P.K.Fine and J.C.Lusth, Expert System and Diagnostic Expertise in the Mechanical and Electrical Dormains, IEEE Transactions:

# 单片机中断源和中断优先级的扩展

中国科学院光电技术研究所 张友生

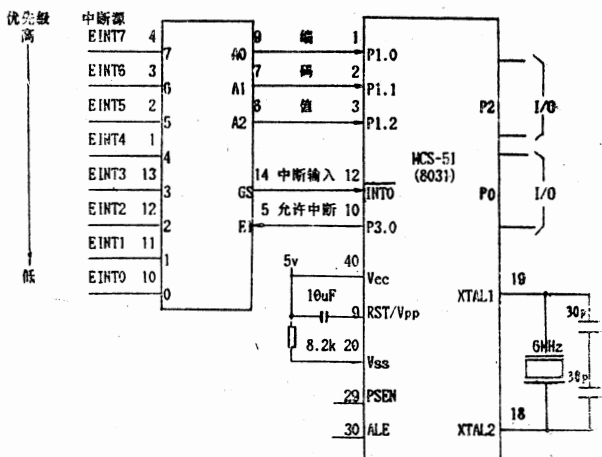
## 一、前言

Intel 公司于一九八零年推出的 MCS-51 系列高档 8 位单片机, 目前在国内已被广泛用于机-电-仪一体化智能控制, 工业过程检测与控制, 计算机网络与通讯技术中。但是, 由于该机在外部中断处理方面, 仅仅提供了两个外部中断输入端(INT0, INT1)和可用软件设置的二级中断优先级, 这在实际应用中受到了极大的限制。因此, 很有必要对单片机外部中断源和中断优先级进行扩展。本文给出了利用单片机现有外部中断输入端, 接受众多的外部中断源申请中断, 并在硬件上对这些中断源进行中断优先级排序, 以及各中断服务子程序的设计方法。

## 二、硬件结构

如果单纯的利用单片机的外部中断输入端来接受众多的中断源输入线, 显然是不够的, 这既要用到中断输入端和部分其他输入口的某些位, 更要如何使这些中断源能被计算机接受, 以及区分它们的中断优先级。假如现在有八个中断源, 八级中断优先级。我们可以用一片“三态输出的 8—3 线优先编码器 (74LS348)”来完成 (图一)。

图中八个中断源按照优先级从低到高 (EINT0, EINT1, …, EINT7) 接到 348 的输入端, 经过优先编码, EINT7 具有最高中



图一 中断源与中断优先级的扩展

断优先权 EINT0 具有最低优先权。384 的输出为二进制编码值, 它与输入的中断源一一对应。每当 348 能有效地接受输入并给出编码值, GS 端就一定伴随着一个电平从高到低的负跳变并输入到单片机的 INT0 端, 向单片机申请中断, 这样, 单片机接到中断申请后可以由 P1 口读取编码值, 并能确切地知道是哪一中断源提出的中断申请, 进而转到执行相应的中断服务子程序。下表较清楚地表明了其工作情况:

on Systems, Man, and Cybernetics, Volume SMC-17, No.3, 1987.

[3] Paul Harmon and David King, Expert System 《Artificial Intelligence in Business》, John Wiley & Sons, INC. 1985.

[4] J. DE KLEER and B. C. Williams, A Theory of Diagnosis from First Principles, Artificial Inte-

ligence Volume 32, No.1, 1987

[5] B. J. Grasz, D. E. Appelt, P. A. Martin and F. C. N. Pereira, Team: An Experiment in the Design of Transtable Natural-Language Interfaces, Artificial Intelligence, Volume 32, No.2, 1987.

中断允许输入 (EI)	中 断 源							编 码 值			单片机 中断申请 (GS)	申请的 中断源 EINT0-7	中断 优先级	
	0	1	2	3	4	5	6	7	A2	A1				A0
H	X	X	X	X	X	X	X	X	Z	Z	Z	H	禁止中断	
L	H	H	H	H	H	H	H	H	Z	Z	Z	H	无中断源 申 请	
L	X	X	X	X	X	X	X	L	0	0	0	L	EINT7	最高
L	X	X	X	X	X	X	L	H	0	0	1	L	EINT6	
L	X	X	X	X	X	L	H	H	0	1	0	L	EINT5	
L	X	X	X	X	L	H	H	H	0	1	1	L	EINT4	
L	X	X	X	L	H	H	H	H	1	0	0	L	EINT3	
L	X	X	L	H	H	H	H	H	1	0	1	L	EINT2	
L	X	L	H	H	H	H	H	H	1	1	0	L	EINT1	
L	L	H	H	H	H	H	H	H	1	1	1	L	EINT0	最低

注：表中“X”表示任意电平；“H”表示高电平；L表示低电平；Z表示高阻态

从工作情况表中看出，“中断输入允许（EI）”是受单片机控制的，当P3.0=1时，348禁止中断输入，其输出为高阻态，向单片机（INT0）提供的中断输入信号也是无效高电平；当单片机允许中断申请时，P3.0输出低电平，即348的EI=L，此时，348允许中断信号输入，并能输出确定的编码值，单片机进入处理时读入这一编码值，便可知道执行哪一个中断服务子程序。至于哪一个中断能获得中断处理，要看它处的中断优先级别，显然，这里的中断源EINT7具有最高优先权，只要348允许，一旦它提出中断申请（低电平），无论其他中断源申请与否（电平为‘X’）74LS348都输出唯一的编码值（000B）。单片机也唯一地执行EINT7申请的中断服务子程序；当中断源EINT7不申请中断时，即为高电平时。中断源EINT6便据有高优先权，一旦它提出中断申请，除中断源EINT7外，其他中断源申请均无效。最后，看看中断源EINT0，它在中断优先级中最低，它要获得中断申请优先权必须在其他中断源均不提出中断时，即：中断源EINT1，EINT2，…，EINT7均为高电平时才行。另外，GS信号在不允许中断或无中断时均输出高电平，对单片机无中断请求，每当出现一次有效的中断申请时，GS信号便从高到低跳变，向单片机输入中断信

号，中断处理工作正常进行。

### 三、编程模式

根据硬件设计及工作原理分析，为简单起见，假设在计算机执行某一个中断源处理子程序时，其他不允许中断它，中断应在主程序中发生，在实际应用中大多属于这类情况。

```

ORG 0000H
AJMP MAIN
; 上电转入主程序
ORG 0003H
; 中断0入口地址
LJMP INTPRT0
; 转中断服务程序
MAIN: ..... ; 主程序
      :      ; 程序初始化
      CLR P3.0
      ; 348允许中断源输入
      .....
      :      ; 执行主程序
INTPRT0: PUSH PSW
      ; 中断服务
      PUSH ACC
      SETB P3.0
      ; 禁止中断源输入348
      MOV A, P1
      ; 取编码值
      ANL A, #07H
      CJNE A, #00H NEXT6
      AJMP EINT7
      ; 转中断源7服务子程序

```

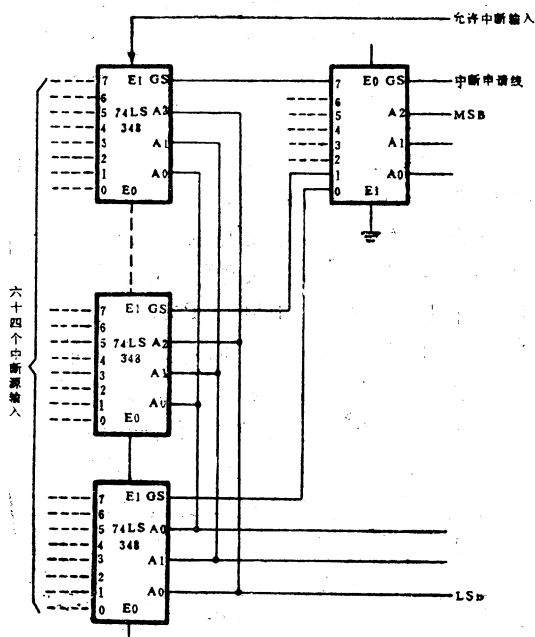
```

NEXT6: CJNE A, #01H, NEXT5
      AJMP EINT6
      ; 转中断源 6 服务子程序
NEXT5: CJNE A, #02H, NEXT4
      AJMP EINT5
      ; 转中断源 5 服务子程序
NEXT4: CJNE A, #03H, NEXT3
      AJMP EINT4
      ; 转中断源 4 服务子程序
NEXT3: CJNE A, #04H, NEXT2
      AJMP EINT3
      ; 转中断源 3 服务子程序
NEXT2: CJNE A, #05H, NEXT1
      AJMP EINT2
      ; 转中断源 2 服务子程序
NEXT1: CJNE A, #06H, NEXT0
      AJMP EINT1
      ; 转中断源 1 服务子程序
NEXT0: AJMP EINT0
      ; 转中断源 0 服务子程序
RETURN; POP ACC
      ; 退出中断处理
      POP PSW
      CLR P3.0
      ; 允许中断源输入
      RETI
EINT7: .....
      ; 中断源 7 服务子程序
      LJMP RETURN
EINT6: .....
      ; 中断源 6 服务子程序
      LJMP RETURN
EINT5: .....
      ; 中断源 5 服务子程序
      LJMP RETURN
EINT4: .....
      ; 中断源 4 服务子程序
      LJMP RETURN
EINT3: .....
      ; 中断源 3 服务子程序
      LJMP RETURN
EINT2: .....
      ; 中断源 2 服务子程序
      LJMP RETURN
EINT1: .....
      ; 中断源 1 服务子程序
      LJMP RETURN
EINT0: .....
      ; 中断源 0 服务子程序
      LJMP RETURN

```

#### 四、结语

利用“三态输出的8—3线优先编码器”解决单片机中断源和中断优先级的扩展问题，是行之有效的方法，如果将优先编码器级联使用，则可以使中断源和中断优先级得到更大规模地扩展。比如，把9片74LS348级联使用可使中断源数目扩展到64个，图二所示。对于单片机来说，用一条中断输入线和一个口（通常为P1口）可以使中断源扩展到高达256个（ $2=256$ ），想必这已足以满足大多数场合的使用了。



图二 六十四中断源输入的优先编码

#### 本刊启事

尊敬的读者：

一九九一年邮局征订工作即将开始，欲续订本刊的读者，请准时去当地邮局办理征订手续，以免脱订。我刊为双月刊，邮发代号62—68，全年每份订价10.80元。



# 长途自动交换机微机计费系统

吉林省计算机技术研究所 郑国基 徐世坤 姚学英

几年前,我们为长春电话设备厂研制了长途自动交换机微机计费系统。该系统在辽宁省铁岭市邮电局通过部级鉴定。各项技术指标均达到要求,得到与会代表好评。厂方陆续生产了二十多台该设备,被安装在十几个省、市运行。用户一致反应该系统工作长期稳定;性能可靠;计费快速、准确;话单完整直观,使用方便;成本低廉。它的应用大大提高了长途电话的使用效率,并摆脱了人工接线,人工计费的繁琐劳动。有着广泛使用价值和经济价值。

## 一、计费系统的主要性能和技术要求

1. 容量:全自动中继器20条、全自动记发器4套。定期用户200个。
2. 机房配一台高速打印机,打印立即、定期和非正常话单。
3. 对远端立即收费用户在营业厅中配两台电传打印机,打印立即收费话单。
4. 计费设备有节假日、夜间半费和人工改动半费起止时间及费率的性能。
5. 计费设备具有对定期收费用户按月累计话费的性能。
6. 计费设备有时钟显示、并能人工校正功能。
7. 计费设备具有自检功能、并打印出自检表格。
8. 向立即用户和定期用户提供二种格式的话单。

## 二、微机计费系统的工作原理

微机构成的计费设备每次通话所要求处理的计费工作都以“中断”方式进行的。在微机RAM中有一个软件编制的时钟。时间数据是在不同的接续时刻从内存的“软时钟”取得。通话距离是由中断器、记发器在第一次向主机发出中断时,同时送出全部被叫码中的“区号”部分来计算。主机处理程序在取出“区号”后,从存贮

在内存中的“区号—费率”对照表中检索出该次通话的基本话价。CPU就有了计算话费的所有原始数据,并能依据送来的主04类别信号中知道该次通话的收费,主04是定期收费用户还是立即收费用户。然后驱使不同的话单打印机将话单输出。

计费系统设有特殊“计费接口”电路,使得从20路中继器和4套记发器发出的请求计费处理呼叫信号和原始数据有秩序地逐一发向主机。计费系统把中继器、记发器看成一个外部的中断源。

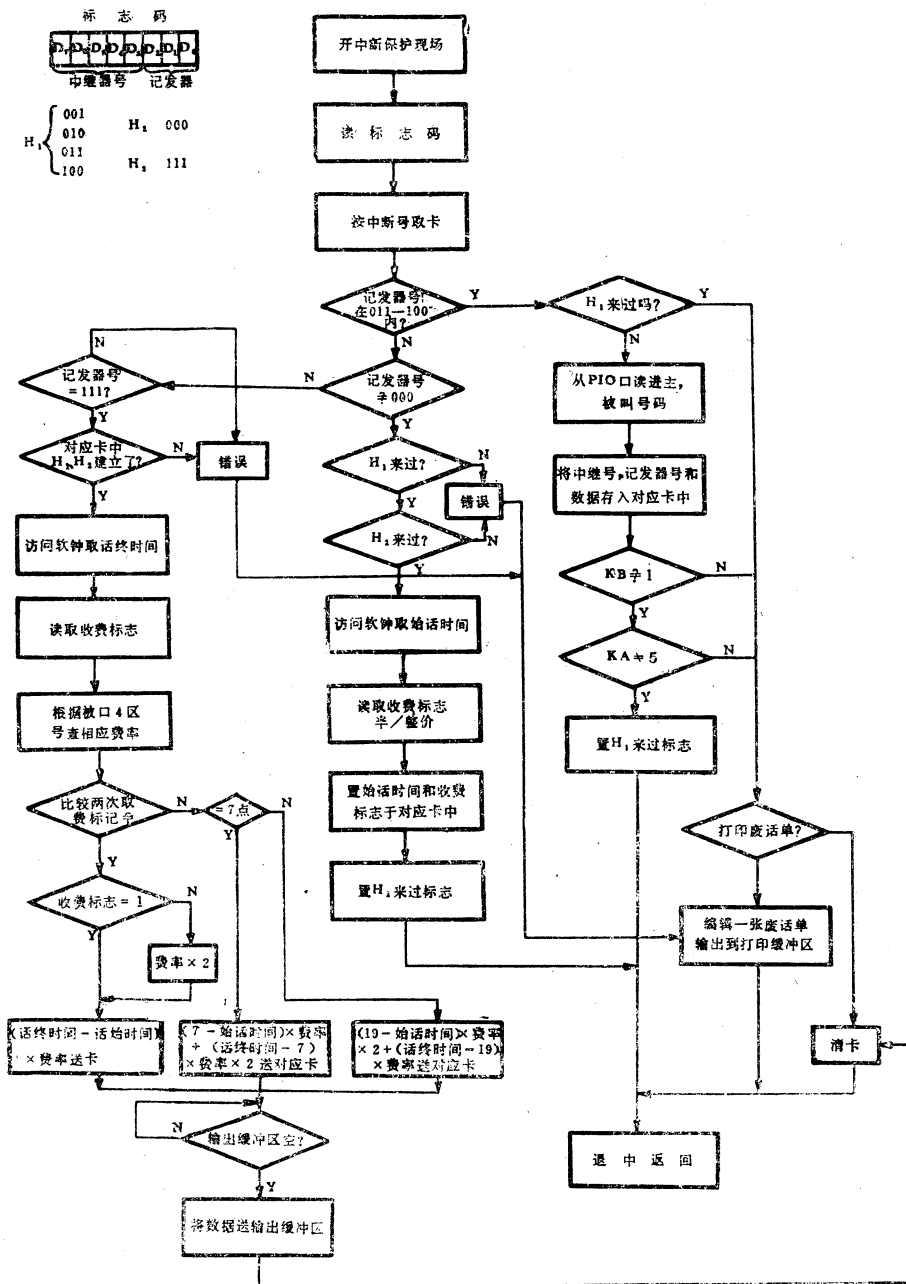
一次完整的接续电话,这个中断源要向CPU发出三次呼叫计费“H”信号,请求CPU以中断方式进行处理:

第一次呼叫 $H_1$ :电话已接续,CPU响应 $H_1$ 中断,即发来一串十个取数脉冲 $Q_M$ ,取出来记发器存放全被主、被叫号码,最后发出 $JW$ 结束信号, $H_1$ 信号撤退。

第二次呼叫 $H_2$ :被叫摘机时从中继电路发出,CPU响应 $H_2$ 时,立即访问内存的“软时钟”,取出实时时刻作为本次通话计费的始话时刻。 $H_2$ 中断程序执行完毕,CPU发出 $JW$ 信号, $H_2$ 信号撤退。

第三次呼叫 $H_3$ :通话终了,中断发出 $H_3$ ,CPU响应后从“软钟”取得终话时刻,计算话费后发出 $JW$ 信号, $H_3$ 信号撤退。

由于20路中继所连接的主叫用户是随机地处于占线、拨号、接通、应答、通话、挂机、折线等情况之中。所以只要有二个以上用户通过计费系统。他们各自的完整过程被反应到CPU时,相互之间的状态就会交叉出现。为了明确某次“H”来自哪路中继,或到哪套记发器取数,在接口电路设有“标志信号”。标志码由八位二进制数组成。 $D_3-D_7$ 是中继器号、 $D_2-D_0$ 为记发器号。三次H信号发出时,只有在 $H_1$ 情况下



记费主程序框图

## 热水瓶温度检验过程中的自动控制

甘肃省计算中心

张晓云

中国科学院近代物理研究所 王水清

**摘要:** 本文介绍了热水瓶生产线上的自动测温控制系统。该系统由一台CAC-3型微电脑, 扩展接口, 温度转换, 数据采集程序等组成。对其他需要自动测温的场合也完全适用。

### 一、测温过程

由于热水瓶内水温的测量范围为 $20\sim 100^{\circ}\text{C}$ ，所以选择2CK10二极管作为测量元件。二极管参数各不相同，在使用前必须经过线性化筛选，以确定每个管子的电压—温度特性曲线。在恒流源供电时（5mA），用7075高精度数字电压表测量到的V—T曲线见图1。得到电压增量式为， $\Delta V/\Delta t = -2.6\text{mV}/^{\circ}\text{C}$ ，线性化可以满足生产需要。转换电路见图2。根据产品指标的要求，在

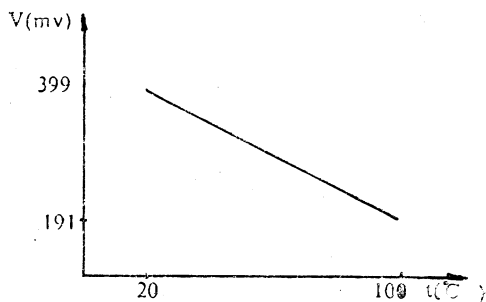


图1 2CKIO的V-T曲线

有记发器和中继器编码, CPU 响应中断, 只要查看  $D_2-D_0$  三位就转到  $H_1$  中断处理程序, CPU 发出十个  $Q_M$  取数脉冲, 把主、被叫号码取来。

H<sub>2</sub>、H<sub>3</sub>发出时，标志码只有中继器码CPU-根据这一点进入相应中断处理程序中。再根据中断码到内存中继码表中查找到本次H<sub>2</sub>或H<sub>3</sub>，应归并到前面哪一个中断H<sub>1</sub>的下面去，填好一张完整的原始话单，供计费处理。

### 三、硬件电路的考虑:

由于电话设备日夜使用、所以做了2个由24伏直流逆变为5伏的稳压电源供微机使用。

考虑到中继器和记发器来的信号是一  $24\text{V} - 0\text{V}$  变化, 并且避免来自电话设备主机干扰, 记发设备与电话设备之间连线均通过光电隔离。

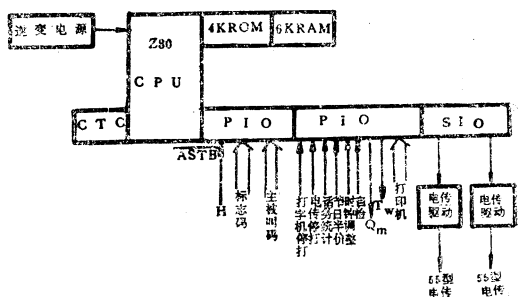
计费设备采用 TP801 单板机、内存又扩充了 6K。并增加了一片 PIO 和一片 SIO，用以驱动一台高速打印机和二台电传机输出话单。

为了操作方便,一些按钮和开关放在面板上。考虑到计费设备出现故障不影响通话和计费,备有另一台计费设备。

#### 四、软件的编制

全部程序由一个循环查询的主程序，五个中断处理程序，以及一些相应功能处理程序和若干子程序组成。共计 4K 在 ROM 中，在 RAM 中还有 20 个“中继卡”，一个“区号—费率表”，200 个有权用户“话费累计表”，能容纳 15 张话单的行打缓冲区和能容纳 5 张话单的电传缓冲区，以及特征字节区组成。行打缓冲区和电传缓冲区留出较多区域是为了在行打和电传由于某种原因暂停时，话单可以保留不致丢失。

中断源的优先级由高到低分配如下：TCC（时钟）→H信号→PIO（行打）→SIO（两个串传）。



### 微机记费系统框图

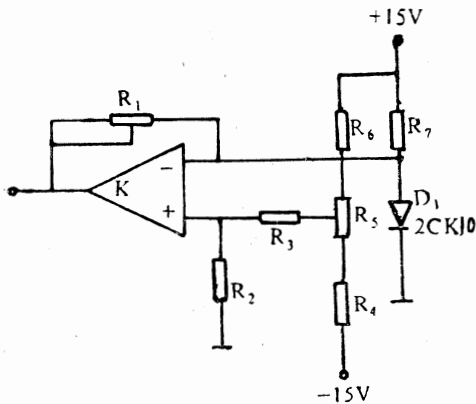


图 2

规定时间内,当水温低于 $85^{\circ}\text{C}$ ,则认为产品不合格。经转换电路整定使检测温度范围为 $70\sim 100^{\circ}\text{C}$ ,对应输出为 $0\sim 5\text{V}$ 的标准电压信号。这样测温元件、转换电路及A/D转换器的对应关系,见表1。进行零点迁移后的8位ADC0809型A/D转换器的转移精度为 $\pm 0.12^{\circ}\text{C}$ ,完全可以满足本系统的要求。

表 1 温度——数字量对照表

温度 ( $^{\circ}\text{C}$ )	$V_D$ (mV)	转换电路 (V)	A/D
70	269	0	00H
85	230	2.5	80H
100	191	5	FFH

探头结构见图3。将2KC10二极管的引线穿过玻璃管后,用CH—31型胶粘剂(工作温度为 $-20^{\circ}\text{C}\sim +130^{\circ}\text{C}$ )将其固定在玻璃一端。另一端也要粘堵封口,这样对于绝热有好处。为防止干扰,引线要用屏蔽线。

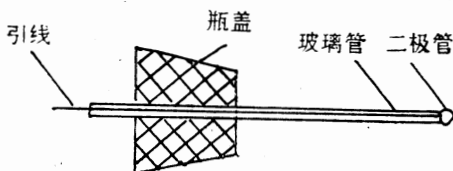


图 3

## 二、系统结构

CAC—2型微电脑的系统见图4。CPU是

Z80芯片。机内有ROM固化的BASIC语言和汇编语言。很适合用于小型工业系统控制。配有16K RAM供用户编程用。微机配有TV和单色monitor接口,音响系统,磁带机接口等。

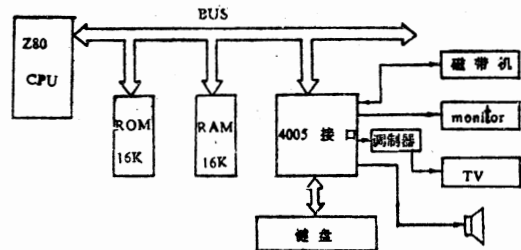


图 4

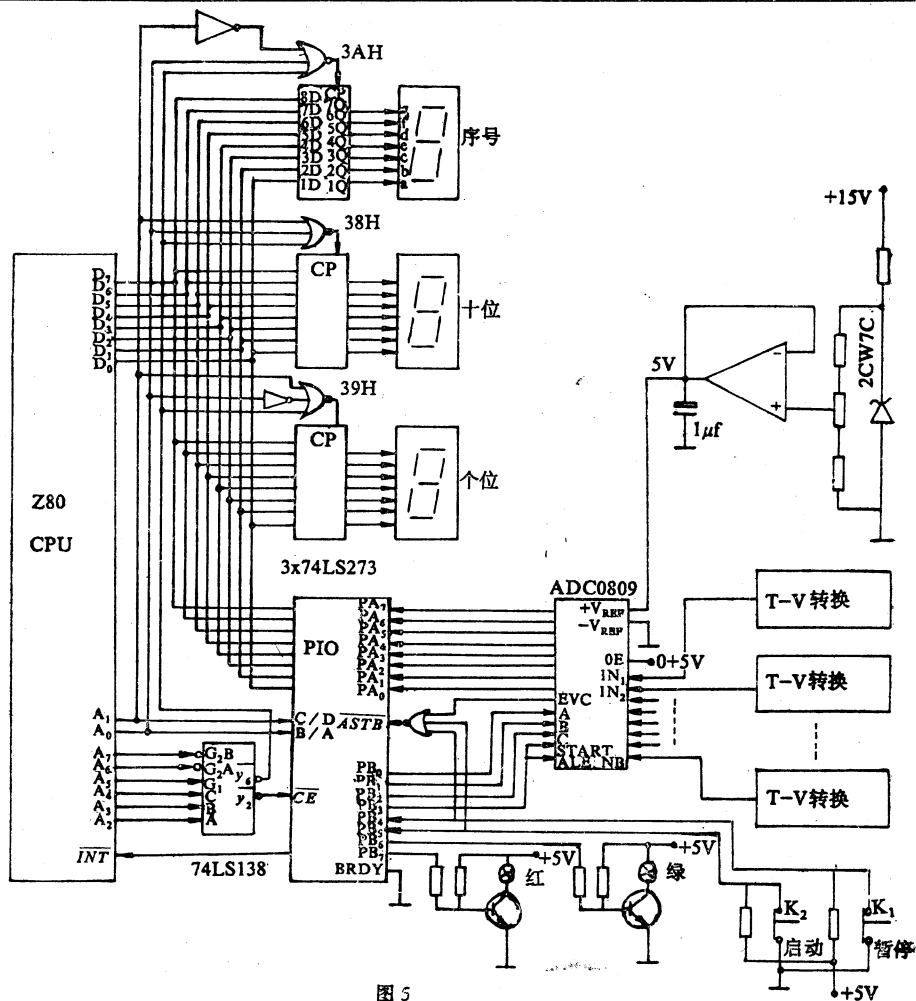
由于本系统价格较低,当配备单显监视器或普通黑白电视机、磁带机和接口电路等构成完整的系统后,其总投资也不超过两千元。考虑到工厂的实际情况。为了降低成本,尽量简化硬件系统,充实软件系统。

图5是控制系统原理图。采用了ADC0809型A/D转换器,对8个热水瓶的温度进行巡回检测。Z80—PIO的地址分配为 $28\text{H}\sim 2\text{BH}$ 。A口设定为输入方式,联接A/D的8位数据线。B口设定为位控方式,作命令和控制用。其中 $\text{PB}_0$ 、 $\text{PB}_1$ 、 $\text{PB}_2$ 用来控制通道号, $\text{PB}_3$ 用来启动A/D转换器, $\text{PB}_4$ 、 $\text{PB}_5$ 是中断源,为输入位。当需要固定显示某一通道温度时,按下暂停巡检按键 $\text{K}_1$ ,这时监视器和七段数码管上显示出该通道序号和温度值。当要恢复巡检时,按下启动巡检按键 $\text{K}_2$ 。 $\text{PB}_6$ 、 $\text{PB}_7$ 是用来发出告警指示灯信号,当测量到某一通道的温度高于 $100^{\circ}\text{C}$ 时,七段数码管的个位和十位显示为“00”,并由 $\text{PB}_6$ 发出绿灯信号,表示测温元件处在蒸气中或元件发生故障,应检查测温元件的安装位置。当水温低于 $85^{\circ}\text{C}$ 时,由 $\text{PB}_7$ 发出红灯信号,表示产品不合格。

三个75LS273锁存器作为七段数码管的接口。地址 $38\text{H}$ 、 $39\text{H}$ 分别显示温度的十位和个位。地址 $3\text{AH}$ 显示通道序号。

## 三、控制系统程序

巡回检测是利用采样开关对输入通道进行逐



个采样。方法是先把第1通道采样4次，然后修改通道号，对第2通道采样4次，直到第8通道。采样周期采用软件延时的方法。在输出通道号后，启动A/D转换。然后延时1ms以后读入数据并且存入内存。对8个通道巡检结束以后，对每个通道的4个数值进行平均运算，平均值就是最后的测量结果。

内存分配见图 6。0~4396H 的内存空间被系统占用,设计控制系统程序时,将汇编子程序安排在 7000H 以上的内存中。这样处理的优点是,汇编子程序不会冲乱 BASIC 程序。控制系统程序采用 BASIC 语言编写。对数据采集程序、显示程序、中断服务程序等子程序用汇编语言编写。在 BASIC 状态可以用 USR 函数调用子程

USR 子程序	7FFFH
BASIC 程 序	7000H
系统变量	4396H
监控和编汇	4000H
BASIC	2000H
	0

图 6

序。汇编子程序要翻译成目标程序（机器码），并用 **POKE** 命令放入内存。各子程序设计成相对独立的模块，可以十分方便地编写和调试。整个控制系统流程图见图 7。

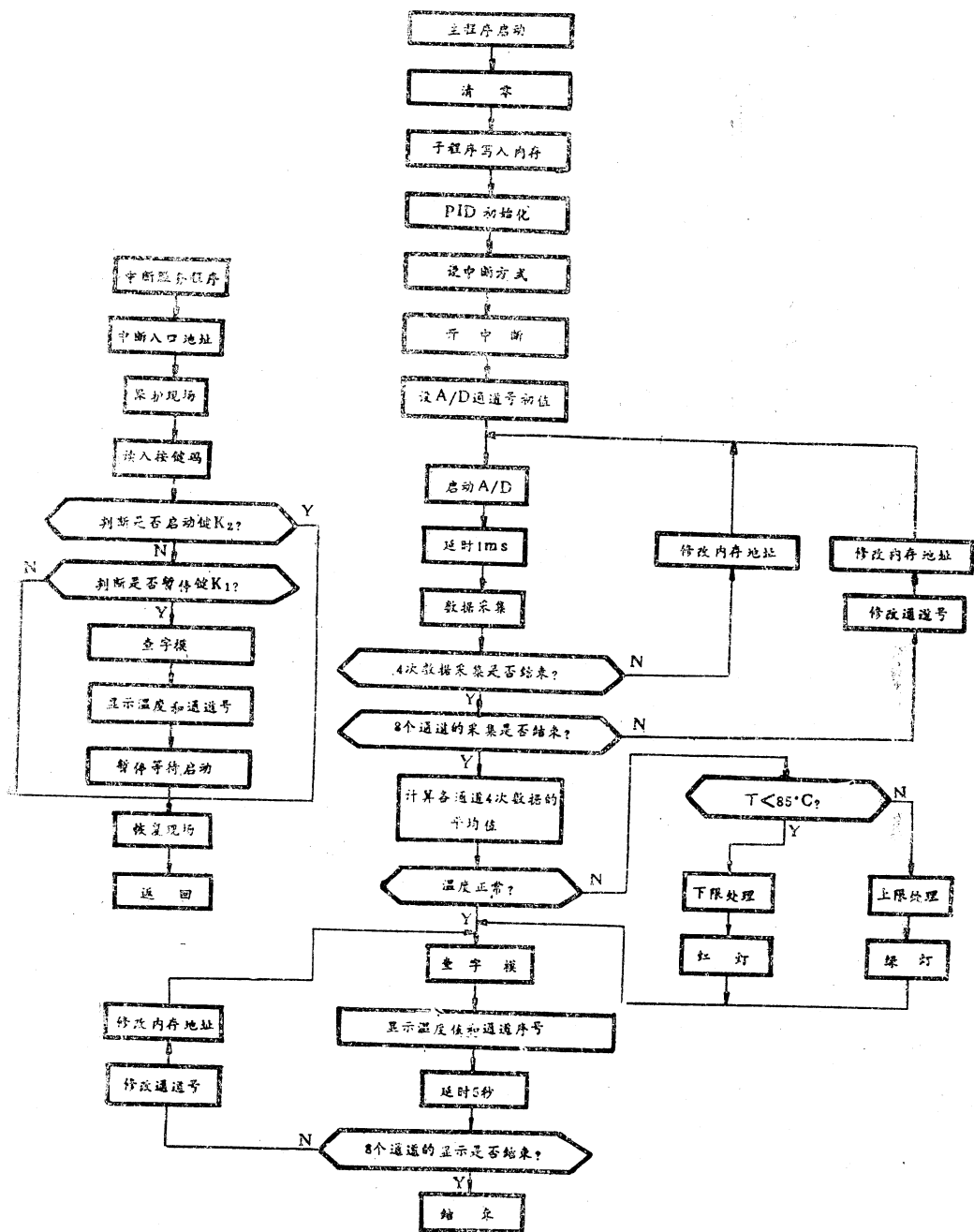


图 7

## 参 考 文 献

- [1] 周明德:《微型计算机硬件、软件及其应用》。
- [2] 《CAC 系列微电脑使用手册》。
- [3] 《PC8300 微型计算机应用手册》。
- [4] 美国德克萨斯仪器公司:《TTL 集成电路特性应用手册》,上海无线电十九厂。

# 高速数字信号单片处理机

## ——TMS320C25 硬软件及应用

电子科技大学 汪亚南

〔摘要〕 本文系介绍TMS320系列数字信号处理单片机的新一代TMS320C25的软硬件特性及其用途,以供读者在实际工作中参考和选用。

**引言** 近代,数字信号处理(DSP)技术已占据广阔的科技领域。随着通讯进一步朝向全数字化网络的方向发展,DSP变得更为有用。数字信号处理技术的特点是用数字方式对信号进行高精度运算和实时处理,并要求数字信号处理系统有足够的灵活性。所谓实时,就是按照预定的DSP算法,以尽快速度处理完毕,其延迟小到难以察觉;系统的灵活性使能适应技术改进与更新,便于对DSP算法加以修正。

DSP技术的重大和最新发展是高速数字信号单片处理机TMS320C25的问世与应用,它是TMS320系列中经过优化设计的新一代成员。

TMS320C25在设计上的主要瞄准目标是全面提高整个系统的速度、通信能力及在处理程序配置上的灵活性。为此,对其软硬件采取了一系列高科技措施。

### 一、硬件结构

TMS320C25是继该系列第一代产品TMS32010基础上发展的、由大规模集成电路构成的一种高速、高位、高性能新型器件。其外形为68引脚的正方形(每边17脚)栅状阵列结构。它的32位ALU(算术逻辑运算部件)和ACC(累加器)可完成各种算术指令和逻辑指令;它具有辅助寄存器堆及大容量的片内RAM和ROM,因而特别适用于数字信号处理系统。其主要特点如下:

- 快速指令周期 100ns。
- 系统吞吐量为TMS32010的2~3倍。
- 对多个处理器具有同步能力。
- 可实现并行DMA操作。
- 具有输入/输出/三态(即I/O/Z)功能。

能。

TMS320C25的结构方框图,如下图所示。

#### 1. 两组独立总线

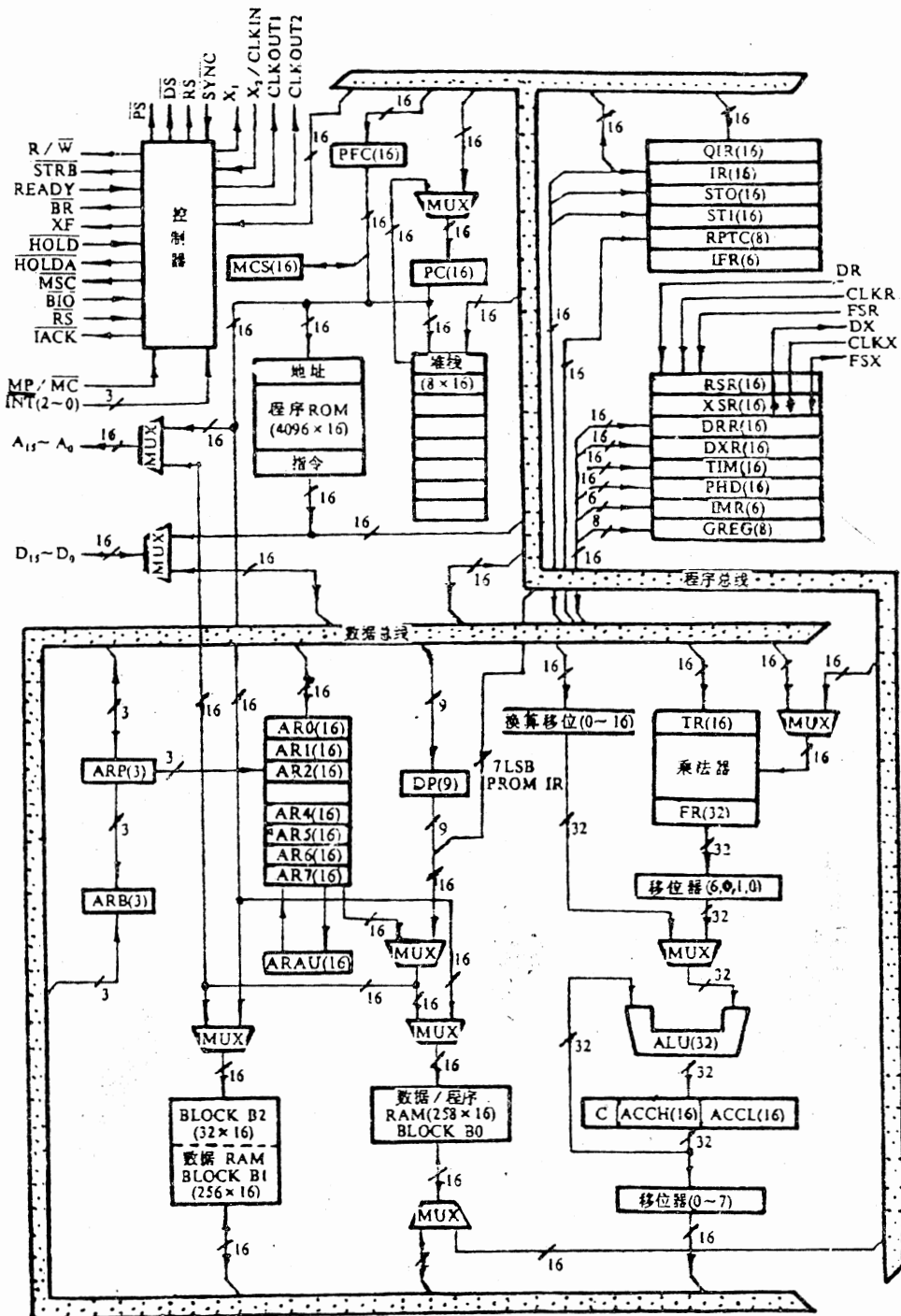
TMS320C25采用了单累加器的哈佛结构,其程序和数据存储器分驻在不同地址空间。因此,取指和运行可完全重叠。在芯片外部,程序和数据存储空间使用同一总线,为的是在器件引脚数最少时,使两个空间的地址范围增加到最大;在芯片内部,TMS320C25提供两组独立总线,即程序总线 and 数据总线。前者为程序存储器所用,后者为数据存储器所用。程序总线上载的是从程序存储器内取出的指令码和立即操作数,数据总线则用于内部各部件的相互连系,将内部组件连到数据RAM。

#### 2. 存储器配置

TMS320C25含有共544(16位)字的片内数据RAM,并分成独立的三块( $B_0$ 、 $B_1$ 、 $B_2$ ),其中 $B_0$ 块有256字,既可作为数据存储器,也可作为程序存储器,使系统增加了灵活性。利用指令(RPTK及BLKP)可将慢速的片外程序存储器中的程序装入 $B_0$ ,然后又通过指令(CNFP)将 $B_0$ 变为程序存储器。用此方法,原来程序存储器中要求快速的程序块就可廉价地存储起来,并得以高速运行。

$B_1$ 和 $B_2$ 共288字,始终作为数据存储器。此544字中,有512字的数据可让TMS320C25处理。

TMS320C25具有片内4K字掩膜ROM。利用MP/ $\overline{MC}$ (微处理器/微计算机方式选择)引脚,TMS320C25将ROM分配到存储器空间以内或以外,这使设计TMS320C25应用系统非常灵活方便。由于片内已有4K字ROM,不仅降低





系统成本,而且也提供了一个真正的单片DSP系统.对于4K以内的程序可以固化到片内掩膜程序ROM中,另有64K字的程序存储器被定为外部空间,供大型程序运行.TMS320C25还提供了直接寻址64K字的片外数据存储器,使DSP算法便于实现.

### 3. I/O接口

TMS320C25能够满足系统接口的各种要求.三个独立的地址空间(程序、数据和I/O)提供了对存储器及I/O的接口.其I/O处理与存储器处理完全相同.TMS320C25利用外部地址和数据总线将I/O设备分配到I/O地址空间;利用READY线实现存储器与各种速度的I/O设备之间的联系;片内串行口用来与串行装置直接通讯;两个串行口寄存器可按8位或16位两种方式工作,串行口设两个缓冲器,可同时进行接收和传送.

### 4. 三级流水线

为了加速指令的执行,实现最大吞吐量,TMS320C25内部采用分层(三级)流水线.指令周期分成三步操作:预取—译码—执行.即当本条指令在执行时,下条指令同时在译码,并且又同时预取再下条指令.这三步操作实际上是直通的(除了被转移指令打断之外),也是相互独立的,因此允许指令重叠运行.TMS320C25内部结构中的PFC(预读取计数器),里面含有要预取的下条指令地址.这样在任何给定的机器周期内,皆有三条不同的指令在做不同的操作.

### 5. 并行DMA操作

TMS320C25支持使用HOLD和HOLDA信号对它的片外程序/数据存储器进行DAM(直接存储器存取)操作.当HOLD置为低电平时,其它处理器可对TMS320C25的外部存储器进行完全控制.使得TMS320C25将其地址线、数据线和控制线置为高阻状态.

TMS320C25有两种通讯方式:一为保持方式,即当HOLD有效时,TMS320C25暂停运行;一为新增的“并行DMA”方式,即在HOLD产生时,TMS320C25工作不停,继续执行,故吞吐量大大提高.

## 6. 同步能力

在多处理器系统中,使用TMS320C25的最大优点是它对多处理器具有同步能力.利用特定的SYNC(同步输入,低有效)引脚,可将多个处理器连接,该SYNC信号常用来使多个TMS320C25的内部时钟同步,由于系统中每个处理器皆同步于自己的内部时钟,故使它们在固定时序下运行.用公共SYNC和外时钟输入可同步多个TMS320C25.

## 二、指令系统

TMS320C25的指令系统是为大力支持数字信号处理、多机处理和高速控制而精心设计的.TMS320C25的绝大多数指令可在单机器周期内完成.TMS32010的源代码与TMS320C25的源代码是向上兼容;TMS320C25的目标码对TMS32020也完全向上兼容,所以TMS32020程序可不修改就在TMS320B25系统上运行.

### 1. 寻址方式

TMS320C25有直接、间接、立即三种寻址方式.

直接寻址:此处不作赘述.

间接寻址:由辅助寄存器堆( $AR_0 \sim AR_7$ )所提供的间接寻址方式,灵活又可靠.间接寻址有7种形式:无变址,加/减1的变址,加/减 $AR_0$ 内容的变址,加/减 $AR_0$ 内容且带反向进位传送的变址.所有形式皆把ARP指向的辅助寄存器内容即 $AR(ARP)$ 用作数据存储器操作数的地址.全部变址操作均在当前辅助寄存器中完成.

间接寻址所用符号为:

$*$ ,  $*+$ ,  $*-$ ,  $*O+$ ,  $*O-$ ,  $*BRO+$ ,  
 $*BRO-$

例:  $ADD *BRO+, 8$

说明:把 $AR(ARP)$ 内容指定的数据存储器中的内容左移8位加至累加器.当前辅助寄存器减去 $ARO$ 的内容.并带反向进位传送.

反向进位变址寻址指的是,在辅助寄存器算术运算部件中,进位传输方向是反向的,并且当前辅助寄存器内容要加/减 $ARO$ 的内容.这种寻址方式要求根据序列大小的一半预置 $ARO$ ,

并把AR(ARP)设置为数据的基(起始)地址。

立即寻址:在指令字中跟在指令操作码后面的字作为立即数。TMS320C25有短(8位及13位常数)立即指令和长(16位常数)立即指令,共含有17种。

## 2. 指令组成

TMS320C25指令集中有133条指令,其中有24条是新增加的,仅为TMS320C25所拥有。这些指令全面提高了器件的功能,可以归纳成下列四组:

- 扩展精度运算指令
- 自适应滤波指令
- 控制状态标志及I/O指令
- 对累加器和寄存器操作指令

## 3. 指令功能

TMS320C25具有强功能的指令,因此,其数值运算和处理功能又有了新的飞跃。

· 单指令循环:在某些高级计算中,要求把相同运算重复若干次。在TMS320C25指令中,含有重复功能,允许一条单指令重复执行。重复计数器(RPTC)为一8位计数器,RPTC可通过RPT(重复)或RPTK(立即重复)指令装入常数0到255,RPTC可一直减到零。因此,对于下一条给定的单指令(不是所有指令都能重复)最多可执行256次。该重复指令非常适合用于乘法/累加(MAC/MACD)、块传送(BLKD/BLKP)、I/O传送(IN/OUT)和表读/表写(TBLR/TBLW)指令。这些通常为多周期指令,当使用重复特性时,所重复的指令只取一次,并且使其为流水线操作方式,从而有效地转变为单周期指令。

### · 完整的逻辑运算和逻辑处理

TMS320C25有一系列逻辑运算和逻辑操作指令。为了对累加器内容进行移位,增设了累加器左移(SFL)或右移(SFR)指令。为了完成多精度算术运算,特设进位位(请见图),并增加将进位位循环(ROL或ROR)指令等。

### · 高级算术运算

TMS320C25含有执行密集型DSP算法的

指令,如MACD(相乘/累加并传送),SQRA(平方加),SQRS(平方减)等。乘法和累加为DSP算法中的基本运算。在数值分析中,要求必须对数值进行平方加/减,这恰是TMS-320C25的特长。

### · 数据传送和存储器块传送

数据传送(DMOV指令)是把指定的数据存储器内容拷贝到下个高地址中,此类传送只在片内数据RAM块B<sub>0</sub>(用作数据存储器)、B<sub>1</sub>和B<sub>2</sub>中进行。存储器块传送(BLKD/BLKP指令)可用来实际连续的存储器字在数据存储器间传送和从程序存储器到数据存储器的块传送。其中,源和目的存储器块不一定都要在片内或都在片外。

这类指令可用以完成DSP应用中必须执行的卷积运算,实现所需的数据传送和数据延迟。这在卷积和数字滤波算法中极为有用。

### · 浮点运算

为了适应动态范围大的计算需要,TMS320C25支持浮点运算,可以处理带有16位尾数及4位阶码的浮点数,也是体现TMS320C25的一项特色。

## 三、TMS320C25的应用

由上已可看出TMS320C25在硬软件设计中的独到之处,精心设计的目的全在于应用。下面简略介绍TMS320C25的几种典型实际应用。

### 1. 数字滤波

数字信号处理所用的数字滤波器,可分为有限脉冲响应(FIR)和无限脉冲响应(IIR)滤波器两类。FIR滤波器是在一个取样数据系统中简单的积之和。滤波器的各阶皆为相乘/累加和传送。TMS320C25用其100ns的指令周期来减少滤波器的执行时间,可以实现多达512阶滤波器。

### 2. 数字通信

在通信网络中,用得上TMS320C25的场合很多。如回波抵消、高速调制/解调、自适应均衡和语音编码等。TMS320C25芯片上的544字片内RAM和4K字片内ROM,可供256阶自适应滤波器(32ms回波抵消)使用,并可用单片

# 微机“五防”系统

华北电力学院 陈骏林 王涌 李林琪

**摘要** 本文首先介绍了微机实现“五防”的机理,然后重点论述了系统的总体结构、功能、软硬件构成技术。

## 1. 引言

电气设备的误操作是电力系统中的频发性事故。据统计,1982年至1984年间全国电气误操作事故达861次<sup>[1]</sup>。电气误操作造成的危害是严重的,轻则造成设备损坏,小范围停电,重则造成人身伤亡,大范围停电,甚至系统瓦解。1985年全国比较严重的大范围停电共13次,其中就有4次是电气误操作引起的。为此,原水电部对误操作问题极为重视,两次召开反误操作经验交流会,制订对策,组织实施。

事实表明,大量的误操作属于运行人员书写操作票或实际操作时的错误。常见的电气误操作有以下五种:(1)带负荷拉、合刀闸;(2)误拉合开关;(3)带地线合刀闸;(4)带电挂地线;(5)误入带电间隔。对应于这五种误操作,水电部提出了所谓“五防”要求。并规定,1986年以后新上的厂站,若达不到“五防”

实现,不需扩展。

语音编码技术,例如全-双工 32kbit/s ADPCM (自适应差分脉码调制)、16kbit/s 副边带编码器以及 LPC (线性预测系数)皆常用于话音传输和存贮。ADPCM 可用全-双工 TMS320C25来实现,TMS320C25为实现上述功能的理想器件。

## 3. 图形/图象和数据处理

对于图形/图象处理应用,重要的是要能够与主机接口。利用TMS320C25的接口灵活性可以实现此种连接并可用于主/从微处理器的各种配置中。

TMS320C25的高吞吐量与多处理能力及数据存储器扩展相结合的特性对数据和数组处理十分有利。使成本降低,并易于用来代替片方

要求,一律不得验收。因此,设计可靠性高的微机五防系统,成了电力生产单位所亟待解决的课题,不言而喻,这一课题的解决具有重要的经济效益和社会效益<sup>[2]</sup>。

## 2. 微机实现“五防”的机理及其特点

目前已有的反误操作装置一般可归结为两类:机械闭锁和电磁闭锁。机械闭锁要求操作人员先查找相应的钥匙,再逐一开锁;但室外机械锁易锈蚀,常在关键时刻打不开锁。电磁闭锁一般需要敷设大量的电缆。另外,这两种装置的共同弱点是无法实现反误拉合开关,因此它们仅能实现“四防”。

本系统实现“五防”的最基本原理是用微机对开关、刀闸的开合状态进行采样,利用软件判断每一步操作是否满足开合逻辑。若满足,则驱动声光引导信号,并自动打开电磁锁以实现操作人员监护。走错间隔或操作不正确时,电磁

式。

## 4. 高速控制应用

利用TMS320G25进行位测试、逻辑处理、定时同步和高速率数据传送( $10^7 \times 16\text{bit}$ 字/S);TMS320C25可用于闭环系统,用来控制信号状态、滤波、高速计算等。磁盘和机器人控制为两类型控制。

## 5. 测试设备

在高级测试设备如谱分析仪和各种高速/高精度仪器、仪表中,常需大存储空间和高性能数字信号处理器。TMS320C25能完成宽度的FFT,并以最少的外接硬件来达到较高的精度。

展望未来,高速数字信号处理单片机TMS-320C25正方兴未艾,前程似锦。

锁在锁的位置,操作人员无法操作。系统组成原理有以下四个特点。

### 2.1 软件实现开合逻辑

电气工程师和现场行动人员都知道,刀闸、开关的开合有其特定的制约关系,这种制约关系即刀闸和开关的开合逻辑<sup>[4]</sup>。经整理归类后即可用布尔表达式描述。这些布尔表达式最终均可编程,构成系统的一个重要组成部分。

### 2.2 防误拉合开关

实现防误拉合开关的关键是寻找合适的测控点。本系统在开关控制回路中串入一对应的继电器,微机可控制其闭合或断开此回路。只有在允许该开关动作的情况下,才构成回路,操作有效;否则操作无效,且系统报警指明误动。开关状态由回路中另一辅助接点采集进入系统。

### 2.3 防误入间隔

如上所述,系统在开锁的同时驱动声光引导信号,这就部分地起到了防误入间隔的作用。此外,在各间隔的网门上同样装有电磁锁。系统根据操作票的内容适时打开相应的电磁锁,间隔内操作完毕时网门重新闭锁。若走错间隔则无法进入网门。

### 2.4 采用分布式测控系统

一般厂站的主控室与现场之间要有上百米距离,若现场所有入/出信号均由电缆拉入主控室,实际上亦不可能。本系统采用分布式微机测控系统,主控室内设一微机主站(master station),现场设若干微机从站(slave station),所有微机均挂在一条通讯线上,现场微机就地测控,信息集中反映在主控室。

## 3. 系统功能

本系统从根本上实现了“五防”要求。在使用和操作时有以下8种功能:

### 3.1 操作票的录入

本系统工作的依据是经值长签署的操作票。因此,系统开始使用时,首先在CRT的汉字提示下由专用小键盘逐条输入操作票,经过对照无误后存入微机。

### 3.2 检查操作票并打印输出

微机对存入的操作票按开合逻辑进行逐条检

查,发现错误则由CRT给出提示,并通过专用小键盘改动,直至通过并打印输出。

### 3.3 执行操作监控和引导

按已通过的操作票一步步引导(声光引导)和监控(电磁闭锁)操作员完成操作,并记录各部操作的时刻。

### 3.4 打印操作票执行结果

按每一步操作的完成时刻打印汉字操作票。

### 3.5 模拟盘动态显示

在主控室自制的模拟盘上动态显示现场状态,每30秒更新一次。

### 3.6 拷贝模拟盘

由绘图机绘出某时刻模拟盘图形(现场状态)。

### 3.7 定时检查电磁锁

每日三次定时检查电磁锁是否完好,锁失灵则报警给出位置。

### 3.8 手动/自动切换

主站可自由切换任一从站为手动或切回自动,亦可全系统切换为手动。系统复位相当于进入自动,系统掉电相当于切换为手动。

## 4. 系统的结构

### 4.1 总体结构

如图1所示,主控室内为GKD\*0工业控制微机,其主要任务是提供一个方便的人机接口,并实现对开关的测控,同时负责整个系统的通讯管理。GKD\*1~GKD\*n微机均安装在现场,实现对刀闸、电磁锁的测控。整个系统可分

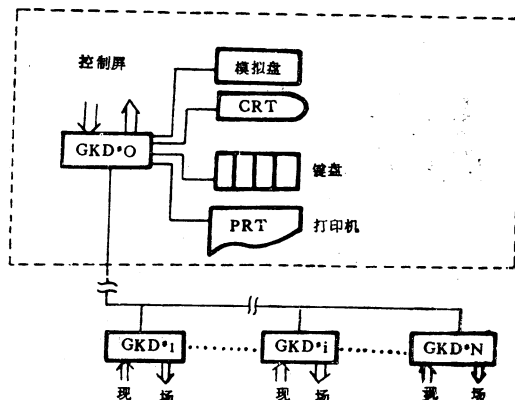


图1 微机“五防”系统总体结构

# 甘肃省省级科技项目与科技成果管理系统

甘肃省计算中心 吕兴寿 吕林霞 雪志军

## 一、前言

科技项目与成果管理是科研管理的核心任务之一。甘肃省科委每年管理的全省重大科技项目达二、三百项之多,其范围涉及数十个学科和技术领域,项目本身又有多种类别和属性,项目承担单位有科研、高校和工厂企业等各个部门,项目

布在 300 平方米的范围内使用,在不加中继器的情况下可控 28 个站。

### 4.2 硬件组成

本系统硬件采用 STD 总线标准、STD—GKD 工业控制微机,以 MCS-51 单片机为 CPU [3]。选用 GKD 微机的 BITBUS 分布式测控系统 [6],并配有:开关量输入板、开关量输

出板,键盘及显示驱动板、打印机接口、CRT 驱动板、实时钟板等。

### 4.3 软件组成

本系统软件全部由 MCS-51 汇编语言编写,也可用 PL/M 高级语言编程。系统可分为主站软件 (GKD\*0) 和从站 (GKD\*1~GKD\*n) 软件,其构成分别由图 2、3 所示。

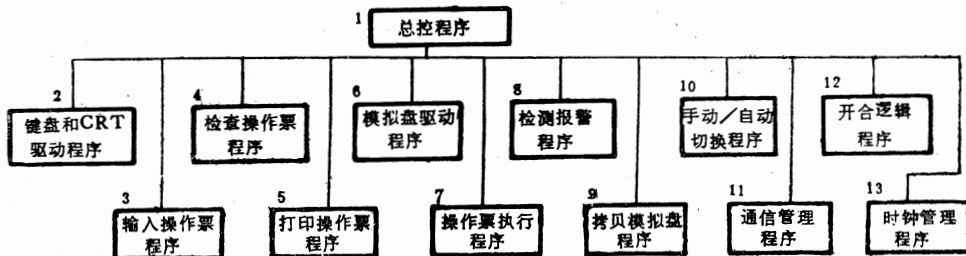


图 2 主站软件结构

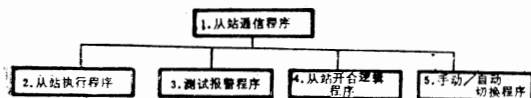


图 3 从站软件结构

## 5. 结束语

本系统运用微机和电磁锁实现了“五防”,并能够方便、彻底地进行手动/自动切换,适用于电力系统。由其机理可见,本系统也同样适用于非电力行业的其它行业,其不同之处仅在于软件中所包含的控制逻辑上差异。因此,本系统不仅具有实用价值,而且也有其一定借鉴意义。

在本系统的设计和实现过程中,得到了我院杨以涵教授的指导和支持以及朱永利、刘长利同

志的帮助,在此表示感谢。同时感谢保定热电厂刘冀田工程师、刘永斌工程师的帮助。

## 参 考 文 献

- [1] “1986年全国继电保护动作及有关故障分析”,《电力技术》,1988年7月
- [2] “日本电网安全自动装置概况”,《电力技术》,1988年7月
- [3] 孙涵芳等编辑:“单片机原理及应用”,北京航空学院出版社,1988年2月
- [4] “隔离开关操作的安全分析与判断”,《华北电力技术》,1987年3月
- [5] Intel “Microcommunication Handbook”, 1987

统地统计分析科研项目结构,调整科研方向和布局,更是意义重大。

## 二、功能设计

根据实际要求,系统主要具备以下五个方面的功能:

1. 信息查询。可按照科技项目数据结构的多种属性内容,进行任意条件的信息检索与输出。

2. 数据统计与报表生成。能按照即定和任意的报表要求,生成各类统计数据码,输出报表50余种,供业务部门直接采用和统计分析之用,以及时发现问题,进行宏观调控。

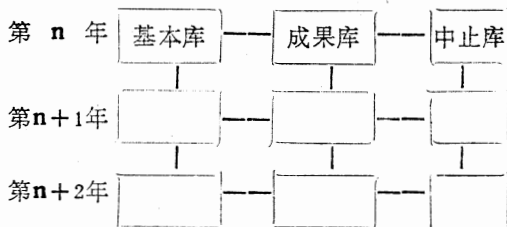
3. 计划管理服务。除完成计划工作所需的特殊查询统计任务外,还结合实际需要,能依据库中未完成的接转科技项目和当年新上项目,编制出年度计划蓝本。

4. 图形输出。按照统计数据码的结果,可生成图形,并输出硬拷贝。图形有近年新采用的带状图,以及传统的条形图和扇形图三种。

5. 数据维护。由于科技项目在实施过程中受外部环境的影响,会出现转项、中止、延期、追加经费等情况变化,要求软件系统能提供方便、完整的数据维护功能。

## 三、数据库设计

科技项目与成果信息包容量大,全部67个字段,且文字型字段多,长度大。由于它们数据内容的唯一性,数据库结构设计本身似无多大难度。但储存信息是在逐年生长。现在库中已输入的,是1978—1987年十年的省立科技项目,以后还会逐年增加。若全部存于一个库中,将会影响到数据检索和运算速度,同时对数据库的维护 and 安全性也不利。因此我们采用逐年分段分块的方法,也即首先按年度分段存放当年的全部数据,然后在该年度再将项目的基本立项情况、成果产出情况及中止失败记录分块存放。各库之间通过项目的唯一标识——项目编号来进行联系,即等效于如下的串、并联方式:



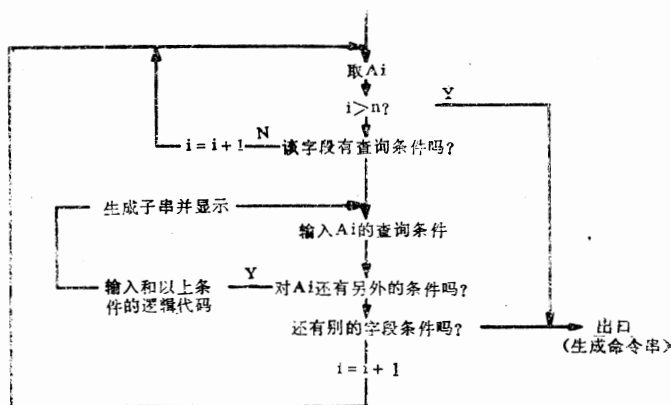
这样,不但给数据维护带来方便,也为不同的信息检索和统计要求提供了灵活性,同时也大大提高了检索、运算速度。

## 四、设计实现

软件实现时,不能仅仅考虑程序与功能的对应实现,而应当与实际环境紧密结合,设计出尽可能简明、完备的面向用户的接口。以信息查询为例,由于它贯穿于系统的各个方面,且记录都较长,为了完全满足用户任意的查询条件输入,同时又能做到简明扼要,本系统中采用了对可查询字段条件的双重循环输入法和查询命令串自动生成法。设库结构字段中可带查询条件的字段名为 $A_1, \dots, A_n$ ,若今有查询条件为

$$A_1 = 'A'.AND.(A_2 = N1.OR.A_2 = N2).AND.A_3 = 'X1'.OR.A_4 = 'X2'$$

其中: $A, X_1, X_2$ 为文字值, $N_1, N_2$ 为数字值,则条件输入和逻辑联结及命令串生成的一般流程为:



程序实现时,为减轻用户负担,除采用详细通俗的人机对话以外,命令串中的逻辑符、括号、圆点、文字引号、数字值的关系符等,全部由机器生成,以最大限度地降低用户的击键次数。除文字输入外,其它几乎只由回车键和数字

# IBM PC/XT及其兼容机系统板维修方法探讨

云南大学计算中心 杨剑波·郭永莉

**摘要** 本文介绍了一种快速处理故障思路及处理故障的方法,它主要以信号的角度叙述了底板故障发生的部分现象及对机器上故障芯片的定位和修理。

## 一、概述

IBM PC/XT及其兼容机,在国内微机用户中占很大比例,自然机器故障率较高,工作量也较大,借助于逻辑分析仪等设备处理,往往许多单位不具备,在维修与处理的实际工作中花费大量时间,收效较微,针对这种情况,我们将近几年在微机维修工作中的经验与理论进行了总结,供大家参考探讨。

在实际维修工作中,人们往往忽略了机器的结构。信号及工作时序的相互关系,凭一些经验去处理,经常故障部位定位不准,造成人为故障扩大,花费了大量时间,而难以实际解决问题。在我们维修处理工作中,通过对机器的整个系统结构进行分析总结,用理论来指导维修,往往可以准确定位,收到事半功倍的效果。

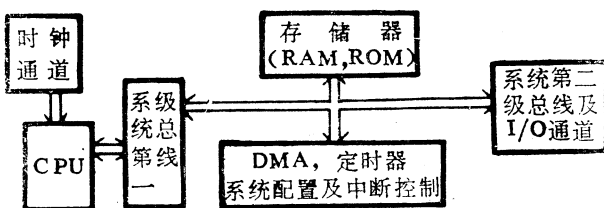
## 二、故障分析方法及处理流程

### 1. 原理简介

在微型计算机的构成中,可分为中央处理器、存储器、输入输出设备等几部分组成,在实际IBM PC/XT中,中央处理器部分又可以看

为由CPU (8088), 时钟发出器 (8284), 总线控制器 (8288) 及其它逻辑控制电路组成,并且在设计中,IBM的I/O通道又是8088处理器的总线的延伸,并增加了8259作为中断控制和8257作为直接存储控制器。借助于信号分离、信号驱动的手段,使得I/O通道处理能力得以提高。

在I/O通道中包括了以下信号:8位双向数据线,20位地址线,6级中断,存储器,I/O的读写控制线,时钟线,定时控制线,三通道DMA控制线,通道校验线及±5伏,±12伏电源和地线,其通道流程如图一。



图一 通道流程图

键即可完成。

为了保证系统的功能、性能,整个研制过程遵守软件工程的开发思想和设计方法,逐步进行。特别是在要求分析阶段,总结现在许多软件系统利用率不高的教训,以实用为基本点同用户反复交流。设计阶段又做了系统模拟,再次听取用户意见,按照他们的希望和要求做了设计修正。

系统的运行环境为CC DOS 2.1D支持下的C-DBASE II,实现的源语言为C-DBASE II和编译BASIC。BASIC主要用于做图,图形数据取自DBASE统计数据库生成的数据文件。

## 五、结语

本管理系统是为甘肃省科委的项目与成果管理而研制的,它原是甘肃科委某软科学研究项目中的一个子课题,完成后即投入使用,它对甘肃科委各业务部门的科研项目立项、管理,科研结构的分析调整等方面提供了有力的辅助手段和完整可靠的数据,受到用户的好评和欢迎。

## 参考文献

- [1] 潘锦平:《软件开发技术》,上海科技文献出版社,1985年出版
- [2] 萨师垲:《数据库系统概论》,高教出版社,1985年出版

根据通道流程, 就可以极其方便地处理故障及定位。

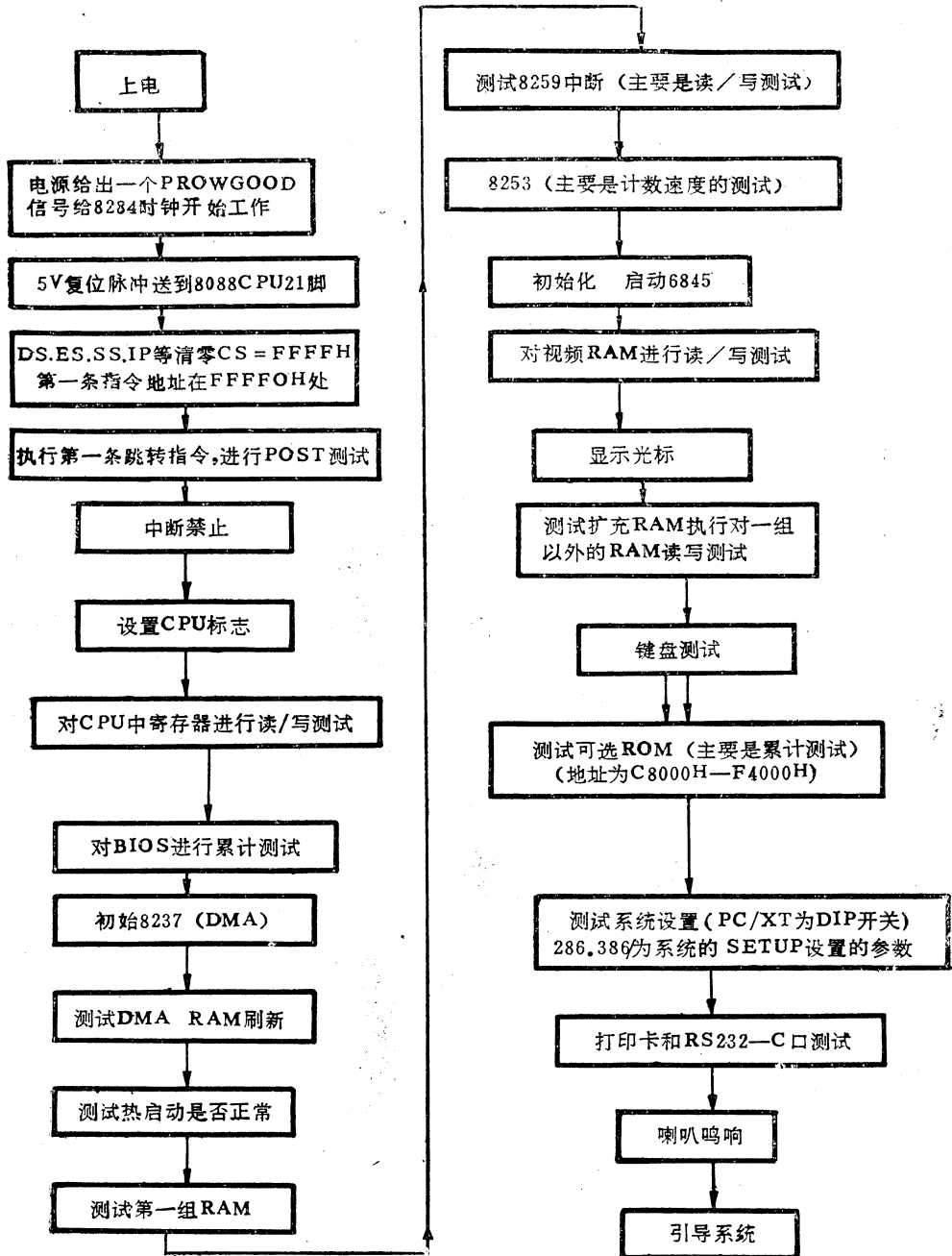
## 2. IBM PC/XT上电启动测试流程

IBM PC/XT上电启动时, 它都要对现系统底板进行一些比较简单但又非常有用的测试, 启动过程如流程图二。

在图二中, 只要某地方有故障, 如果这种故障不是影响整个系统启动, 它将忽视, 继续自检下去, 如果是关键性故障, 则系统将停在该故障点, 不再继续往下工作。

## 3. IBM PC—XT系统板故障分析及处理

根据图一、图二可以看出, 作为IBM PC—



图二 上电启动时执行信号过程



XT 系统板, 故障处理可以先以电源开始。在很多时候往往不注意 POORGOOD 信号, 只有这个信号到来, 8284 才工作, 而时钟又是 CPU 及其它电路工作的基本时序。它不正常, 必导致整个系统不正常, 在基本时钟正常时又来检查第一级总线驱动及总线控制器 (8288 总线控制器, ) 74LS245 数据缓冲器, 74LS373 (A00—A07) 地址缓冲器, 74LS244 地址缓冲器)。在排除了一级总线无问题时, 可检查第二级总线 (74LS245 数据缓冲器 74LS244 地址缓冲器)。因此, 检查流程如下:

开始将所有扩充板拔下

检查  $\pm 5V$ ,  $\pm 12V$  是否对, 不对则修理电源。

检查 8284 时钟电路, PWRGOOD 信号是否正常, 不正常, 可以在 8284 的 11 脚 (RES) 上加个  $33\mu f$  电容与一个  $10K$  电阻串联, 再接到  $+5V$  上, 此时, 检查 8284 的第 8 脚 CLK88) 及第 11 脚 (RESET) 等信号是否正常, 不正常, 一般为 8284 芯片或 74LS04 芯片故障, 此时, 可以检查晶振及 13 脚 (F/C) 是否为低, 若两个脚正常, 可将 8284 第 8 脚与外电路断开, 若此时 8284 上有  $4.77M$  信号, 则是有关 CPU 或总线上故障, 顺着信号走下去, 即可处理。若无  $4.77M$  信号, 则完全可以认为 8284 芯片损坏, 更换即可修复。

以上信号全对时

测量第一级总线, 应先检查 8288 是否正确。正确, 则检查总线驱动器 74LS373 74LS245、74LS244。若发现某一根处于三态, 永远为低、为高。此时该检查其对应芯片是否损坏, 对损坏芯片更换, 即可正常。

以上信号都对

测量第二级总线 74LS244, 74LS245 是否有三态, 线恒为低, 恒为高的线, 若有则检查对应芯片是否损坏, 损坏更换即可正常。

以上信号都对

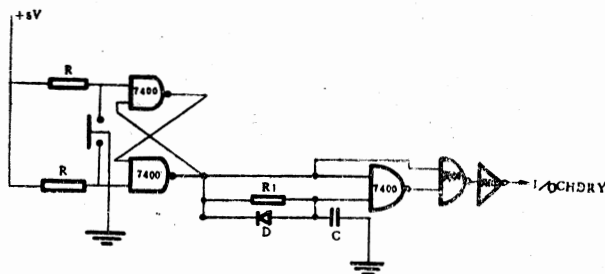
对一、二级总线和 CPU 总线进行测量。测量方法为: 将 I/O CHDRY (即二级总线上的 A10) 接一低电平, 将 A10, b10 短接, 此时上电, 据计算机启动原理可知, 第一条指令为

JWP F000: E05B 地址所在处为 FFFF0H 到 FFFF4H, 如表一:

(表一)

地址	数据
FFFF0H	EA
FFFF1H	5B
FFFF2H	E0
FFFF3H	00
FFFF4H	F0

据计算机工作原理, 我们做了一个 I/O CHDRY 控制器, 使之再复位一次, 产生一个  $400\mu s$  的高电平就能顺利检测出整个系统总线是否正常, 其原理如图三:



图三

注: 图中门电路均使用 7400 元件

只要将计算机电源关了, 将图一电路的输出接到二级总线的 A10 (I/O CHDSY) 上, 就可以上电开始测量, 此时正常的结果为:

开始时

一、二级总线	A19—A0	为	FFFF0H
	D7—D0	为	EAH
CPU 总线	A19—A8	为	FFFH
	AD7—AD0	为	EAH
	S2, S1, S0	为	100

击一次键后

一、二级总线	A19—A0	为	FFFF1H
	D7—D0	为	5BH
CPU 总线	A19—A8	为	FFFH
	AD7—AD0	为	5BH
	S2, S1, S0	为	100

只要继续往下走, 就可以测出 CPU, 及一、二级总线是否有问题; 若有, 对应的位置展开分析就可以找出故障位置。

# IBM PC/XT微机死机故障维修一例

南京向往电脑服务部 邵发森

**故障机器的配置情况：**主板型号“64—256KB SYSTEM BOARD”；主板RAM 256KB，扩展RAM 384KB，总容量为640KB，彩显为CGA，单软盘驱动器360KB，硬盘容量10MB；键盘为83键。

## 一、故障现象

开机自检正常，运行西文DOS正常，但运行中文DOS或者较大的西文文件如“PCTOOLS”就容易死机，不能拷贝，连格式化软盘片都不行，软驱只转键盘命令不能中断。时而屏幕无任何显示，时而显示“PARITY CHECK

1”奇偶校验错，错地址数值也有随机性，有时为00000(S)有时为10000(S)，有时为5000(S)。

## 二、分析检查

既然自检正常，主要的大规模集成电路芯片的故障问题先不予考虑。先检查一些外围的辅助电路。例如：非屏蔽中断电路、8级屏蔽中断电路、奇偶校验电路、以至于动态时的电源电路，检查后均正常。跟踪程序运行检查，发现机器初始化后，在低64KB地址范围内运行正常，一超过64KB就死机。初步认为是高位地址A16—A

测试8255A—5的PA2—PA0的状态器

PA2	PA1	PA0	结果
0	0	1	BIOS累加机错
0	1	0	8253——5错
0	1	1	8237A——5错
1	0	0	第0组RAM错
1	0	1	8259错
1	1	0	CRT适配器错

也是好的，更换电源后故障消失，这儿要说明的是，作为计算机系统板的故障，很少见到大面积故障，常常仅是某一很小故障，只要按原理和信号关系去分析，就可以比较容易处理。

测试8255A—5的状态口测试，可以定位到某一芯片。但有一点必须注意，这儿讲的8253、8237、RAM、8259的错，并不是指这块芯片损坏，而是指包括该芯片在内的这部分逻辑电路，此时，对应原理图，可以非常方便地检查出某一个信号不对，处理后即可恢复。

当发现第一组RAM有问题时，可以将RAS0.CAS0与RAS3.CAS3对调，引导基本系统后，再用高级诊断程序来对第一组RAM的某片损坏芯片定位，即可进行修复。

在实际维修中，往往对电源的正常及电源的负载能力注意不够，造成许多故障无以下手，因此，作为电源负载能力的变化，对整个系统影响比较大，如有一台IBM PC/XT机开机时，硬盘出错，但换到其它机器上又是好的，系统板

又如一台IBM PC/XT自检后显示1701错误，将该硬盘与控制卡换到另一台机器上又是正常的，故障何在？在DMA(8237)的第三通道。原因是在DMA的二通道是为软盘适配器使用，三通道是给硬盘适配器使用。此时，DMA损坏，硬盘自检时写DMA缓冲器失败，因此显示1701错误。机器与外设的连接信号与自检过程，对维修者来讲是非常重要的，有关外设自检过程，本文不多说，请参阅有关资料。

## 三、结束

从上面的分析可以看出，计算机故障的分析处理是需要理论作为指导，由于信号之间相互联系太多，如果用头疼医头的办法处理，往往不可能处理故障，只有在系统的高度上，以信号启动过程自检错误显示等方面，进行全面分析，才能从根本上解决故障。并且，以笔者的分析、处理故障经验来看，绝大部分故障，仅可能出在某一信号的驱动逻辑电路损坏，而不会损坏多块芯片。

由于水平有限，仅起抛砖引玉的作用，不对之处请批评指正。

## Super-CH显示器扫描电路的代换与维修

中国人民解放军军事经济学院计算中心 刘亮生

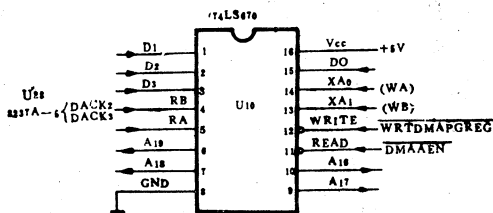
计算机显示器的扫描电路不仅提供系统所需要的低压,还担负着加速极、聚焦极和阴极等中、高压的任务,功耗大、电路复杂、故障率高。如果该部分出了问题,将直接影响显示器的正常工作。在维修扫描电路的过程中,遇到的主要困难是无原配图纸,特别是一些厂方专配的关键零部件,如行输出变压器等的损坏而造成的配制困难。由于查不到有关的技术资料,增加了代换制作的难度,甚至无法修复。

对扫描电路来说,各显示器的原理是一样的,

只是在功能实现,电路设计及工作条件上有着不同的方式和特点。只要熟悉电路,掌握好行输出变压器各脚的功能,在线路上做适当的改进,是完全可以进行代换,修复好此类扫描电路的故障。

**一、故障现象及检查分析过程:** 一台 Super-CH高分辨彩色显示器,出现无光栅故障。经检查电源保险丝完好且无短路现象后,加电开机,发现指示灯及显像管灯丝点亮(亮度稍感暗淡)。初步判断电源电路及行振荡、行激

19有问题,死机又经常处在“DMA”存储器直接访问过程中,即在使用软驱或硬盘的过程中出现故障。这样把故障缩小到“DMA”工作时的低位地址电路。由电路原理图可知。(见附图)



系统地址总线是20位,8237A—5DMA控制器(U28),只能输出16位地址(A0~A15),高4位(A16~A19)则由页面寄存器74LS670

(U10)产生。它具有四组四位寄存器堆,存放四个DMA通道操作的高4位地址A16~A19。它的四位数据输入端接到系统数据总线的低四位D0~D3,当它的第12脚WRITE端接到系统发出的WRTDMAPGREG信号时,(低电平有效)对页面寄存器写入数据总线D0~D3的内容,寄存器组写入选择端WA接地址总线A0,WB接A1,这样,BIOS中按排的页面寄存器写入地址是:

CH1 (用户留用) 83H

CH2 (软驱) 81H

CH3 (硬盘)

82H

四位数据输出端接到系统地址总线A16~A19。页面寄存器读出的控制条件为:第11脚RE AD端接受了系统控制信号DMAEN,(低电平有效)。此时读出寄存器内容的选择端RA接DACK3, RB接DACK2,即控制工作在硬盘通道或是在软驱通道。DACK0通道用于RAM动态刷新,不使用页面寄存器中输出的内容;DACK1通道是留给用户备用的通道,开发后方可使用页面寄存器的输出内容。

经过以上分析,须在线检查U10页面寄存器,用示波器观察74LS670的各个脚的波形发现输出脚6、7、9、10的输出波形异常,脉冲峰值只在1.4V左右,属故障波形,正常波形应在3.9V以上。初步诊断页面寄存器损坏。

### 三、排除故障

用吸锡器将74LS670的16个脚仔细吸干净,取下该集成块,换上一个16个脚的IC插座。用万用表测量有关接线正确后,插上一片好的74LS670,机器恢复正常。

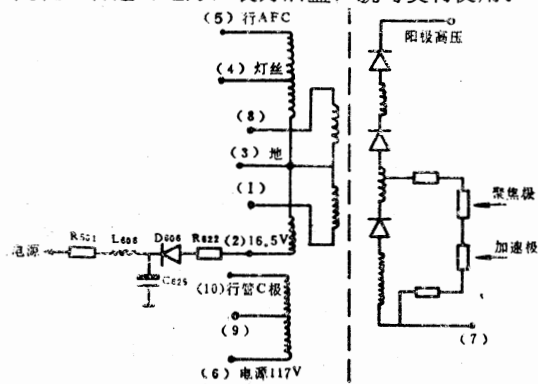
因为主板是多层(四层)电路板,焊接次数多或焊接不当,易造成短路或断路,故采用换坏的IC芯片时,焊上一个IC插座,以利日后维修。

励、行输出极电路正常。开足亮度屏幕无光栅，接着检查显像管电路各供电电压，发现加速极电压与典型电路相比幅值降低了很多。用试电笔探头靠近高压线，氛管不亮，开足亮度几分钟后关机，其瞬间无光点闪烁，表明无阳极高压输出或输出高压不足。再重新检查行扫描输出部分，稳压电源117V通过行输出变压器的6脚送入，由10脚引出接到行输出管C极，提供直流工作电压。经测量该电位已由117V降至90V，拔下行偏转线圈插头后，电压有所上升，说明开关稳压电源是好的，只是由于负载重而引起输出电压偏离正常值。继续测量发现，行输出变压器各绕组提供的电压均按比例减小，其最大可能是它本身有故障，一般为局部短路。这时断开电源，用手摸行输出变压器可感到温升异常。为进一步证实，可将T601行推动变压器初级线圈短路，（使行激励信号为零），这时的稳压电源输出恢复到117V。此时，已可充分肯定问题出在行输出变压器，只是无原配，须另选型代换。

**二、代换与修理方法：**该机扫描集成电路IC芯片是HA11235，显像管是东芝产品，分析有关参数，选用引进东芝技术国产佳丽牌彩电所用的行输出变压器（同系列的也能代换），其主要绕组的相位与电压比及功能脚端与原机基本相同，只是安装尺寸有差异，见附图。2脚提供经整流为16.5V的脉冲电压，3脚为接地端，4脚供显像管的灯丝用，5脚为行AFC自动频率控制电路提供行脉冲，6脚接电源117V，10脚接行输出管的集电极，7脚接ABL电路。分别将2、3、4、5、6、7、10脚用导线与印刷板上对应的点相连接，1、8、9脚不用。再来查看扫描IC电路的工作条件，行扫描电源是12V，由稳压电源输出的46V经R601、R602分压送到IC—11脚。IC—6脚接场扫描电源11V，由行输出变压器的2脚提供。注意到2脚此时输出电压偏高（16.5V），在其输出端串一个 $15 \sim 20 \Omega / 2W$ 的电阻分压，代替原电路的R622，使场扫描电源值达到正常值。仔细检查一遍后，开机观察。调整加速极（SCREEN）电位器和亮度（BRIGHT）电位器，粗调聚焦（FOCUS）电位器，使光栅处

于正常状态。如果光栅幅度有变化，可通过适当改变逆程电容C602来调整行幅，通过调整VR511(D.SIZE)来改变场幅。这些工作做完后，就可将主机视频输出信号接入，一般都能正常工作。然后精调聚焦电位器，使字符显示清晰。如聚焦电压过低，可适当提高阳极电压（减小逆程电容）。如果出现行不同步或相位不对，可调整VR608(H、HOLD)、VR601(H、PHASE)来改善。如仍无效果，则需查主机九针连接器8脚上的行（水平）同步信号，经过Q202、Q205、Q206放大整形送到IC—16脚的波形是否正常。行同步信号同时还加到IC的脉冲检波电路上，在这里能把自行输出电路的行脉冲经过VR601、R609、C609积分而产生的锯齿波电压与行同步脉冲进行相位锁定。行脉冲由行输出变压器第5脚引出，锯齿波电压由IC引脚13加入其内部的行AFC鉴相器。由IC引脚14上输出相应于行同步信号与锯齿波电压之间相位差的直流电压。这个电压又经过由C603、R603、C602组成的低通滤波器，并通过R604耦合到引脚12，施加于行振荡电路的控制电路上，自动调整行频，以实现行振荡与行同步信号相同步。如果集成电路IC扫描芯片正常（可通过换芯片来检查，该芯片带有插座且不易损坏），图象仍不能同步，则需检查振荡频率是否偏离，主要检查引脚12外接的RC定时电路R605、R606、R607、VR608、C607、C608等，适当调换C607//C608（不可过多偏离正常值，以免引起行输出级过压过流）。

通过以上维修调整，一般都能修复扫描电路，使显示器正常工作。最后将行输出变压器固定在一合适的地方，装好后盖，就可交付使用。



## 2200VS系统死锁故障之一的排除

甘肃省计算中心

张晓云

2200VS系统是中小型数据处理机,具有分时多道程序和虚拟存贮功能的操作系统,带有十六台串行工作站及多种外围设备。

由于是多用户系统,在系统运行工作中发生死锁对用户来讲是一个影响很大的问题。当用户把程序和数据输入内存,还未存入磁盘,或是用户正在运行程序,系统发生死锁,临时解决的办法只能重新启动主机,那么用户的程序或数据就会丢失,造成用户方面的损失,并且严重影响整个系统的可靠性,因此,我们对此故障进行如下分析和查找:

### 一、判断是软件故障还是硬件故障

系统发生死锁的原因是多方面的,如用户操作不当,外围设备出故障,特别是硬磁盘驱动器有故障可直接影响系统死锁。还有操作系统本身不完善等都可以造成系统死锁。而在2200VS系统中,操作系统已经稳定地运行了一段时间,故我们初步判断不是操作系统本身的问题。每当死锁出现,我们仔细的询问了用户,也排除了用户操作不当这一原因,所以,我们就从硬件方面考虑排除死锁故障。

### 二、确定故障设备

2200VS系统带有磁带机,宽行打印机,硬磁盘机和软磁盘机,每当死锁出现时,我们注意

观察到磁带机和宽行打印机并没有被使用,故可以排除这两种设备引起的死锁。软磁盘机进行了测试未发现异常也可以排除在外,重点是磁盘驱动器,我们对四台磁盘驱动器逐一进行了测试,又将带有操作系统的盘组分别放在四台磁盘驱动器中,观察了数日,故障仍出现,这时就初步确定为主机(CPU)有故障存在。

### 三、检查主机

主机面板上有各个控制板功能故障显示灯,当主机控制板有异常时,这些显示灯能准确告诉我们故障范围,(到板子级)但当系统死锁时这些显示灯都不亮,这说明主机控制板是好的,剩下的就是主机电源部分。

用电表测量主机供给各个控制板的电压,发现电压均偏低,+5V降低为+4.3V,+12V降低为+11.4V,同时又检出电源电压输出电缆线的根部发热并有烧糊的症状,这种现象说明电源电路中有短路现象,大电流使电缆线发热。

进一步测量电源中各个电器元件,和各点工作电压,果然检查出电源电压输出滤波电容击穿,引起电源输出电压偏低,造成主机电路控制板不能正常工作,形成死锁现象。

换下被击穿的电容,再次测量电压,电缆线再未发热,观察数日后,死锁现象消失,系统运行正常,工作电压稳定。

### 三、HA11235各点参考电压:

(1) 场高次谐波抑制3.0V (2) 场预激励输出0.7V (3) 场线性补偿3.0V (4) 场幅调整3.6V (5) 场RC定时电路6.6V (6) 场扫描电源11.0V (7) 场同步输入4.2V (8) 场同步调整4.2V (9) x射线保护但此处接地0V

(10) 行预激励输出0.4V (11) 行扫描电源12.7V (12) 行RC定时电路6.8V (13) 行AFC比较脉冲输入3.6V (14) 行AFC输出6.8V (15) 连到11脚为12.7V (16) 行同步信号输入1.5V (17) 没用 (18) 接地0V

## 硬盘维护的良好工具——File\_Cleaner

北京理工大学 刘里鹏

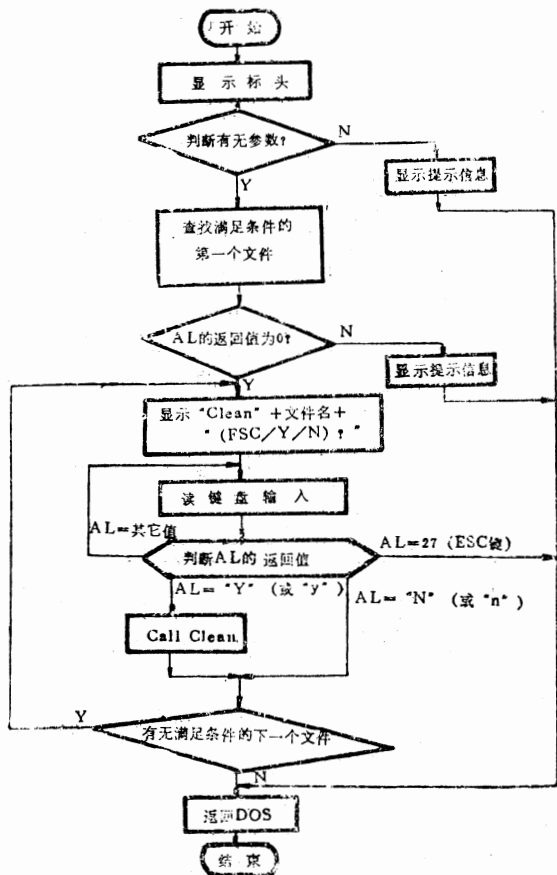
硬盘因存取速度快和容量大等优点而被广泛地使用。但要保证硬盘有足够的可用空间和较高的读写速度,就必须经常对硬盘进行维护,删除那些不再使用的文件。有时,出于保密的考虑,用户还要对硬盘上的某些文件进行彻底地删除,即物理删除,以保证这些文件不被人“非法窃取”。

对于前一工作,通常情况下是使用DOS内部命令中的DEL(或ERASE)命令来实现的。尽管该命令可以通过令程序文件名字符“\*”和“?”来说明文件,实现一批文件的删除,但实用中往往会因为文件命名复杂而导致漏删或错删,极不方便。诸如PCTOOLS一类的工具软件虽然较好地解决了这一问题,但因为读写频繁而使得速度很慢,也不能令人满意。而且它们所进行的删除都是非物理性删除、是可恢复的,无法满足保密的要求。

这里向计算机用户介绍一个小程序FC、ASM(File-cleaner),由它生成的、EXE运行程序能较好地解决以上两个问题。

该程序由汇编语言完成,结构比较简单(其程序框图见图一)。

该程序运行时,首先显示有关的标头信息并判断有无运行参数,若无,则给以提示并退出运行,返回DOS;若有参数,则利用DOS系统调用INT21H中的11H号功能查找满足条件的第一个文件。若AL的返回值不等于0,即无匹配的文件,则显示提示信息并返回DOS;若AL的返回值等于零,即存在同运行参数相匹配的文件,(\*)则由ES:[81H]得到前次功能调用所返回的相应文件名,并询问用户是否要对该文件进行删除操作。若得到认可,则调用删除子程序Clea;若用户响应为“N”或“n”,则跳过子程序调用操作;另外,此处还特意设立了对于ESC键的检查(ASCII码等于27),若用户按动此键,则程序直接返回DOS,这就保证了用户能在程序执行过程中任意中断运行。



图一 程序框图

程序的最后一步是通过DOS功能调用中的12H号功能,查找有无满足条件的下一个文件。若有(AL=0),则转到(\*)处循环执行上述步骤,直到不再有匹配的文件存在为止;若无,(AL≠0),则返回DOS。

由程序框图不难看出,对文件进行物理删除的关键部分是“clea”子程序,它实质上是对相应的文件进行了写零操作,即从一个内容为零,大小为1024字节(此大小用户可自行设定)的缓冲区向文件写入零数据,从而达到清除文件的目的。在运行INT21H中的13H号功能删除文件之前,程序进行了这样几个操作:再次打开

程序清单如下:

```

; Source file
DATA SEGMENT
CR EQU 13
LF EQU 10
ESC EQU 27
MSG0 DB CR,LF,'FC-File Cleaner, Multiple File Utility Version 1.00.'
DB ' All rights reserved.',CR,LF
DB 'By Liu Li Peng,the Computer Department '
DB 'of Beijing Institute of Technology, 1989.',CR,LF,LF,'$'
MSG1 DB CR,LF,'Clean $'
MSG2 DB ' (Esc/Y/N)? $'
MSG3 DB 'No matching Files found !',CR,LF,'$'
MSG4 DB 'File names must be specified !',CR,LF,'$'
HANDLE DW ?
FILESPEC DB 13 DUP(?)
BUF DB 1024 DUP(0)
NUMS DW ?
DATA ENDS
CODE SEGMENT
START PROC FAR
ASSUME CS:CODE,DS:DATA
PUSH DS
XOR AX,AX
PUSH AX
MOV AX,DATA
MOV DS,AX
LEA DX,MSG0
MOV AH,09
INT 21H
MOV SI,80H
MOV AL,ES:[SI]
CMP AL,1
JGE CONT
LEA DX,MSG4
MOV AH,9
INT 21H
JMP QUIT

CONT:
PUSH DS
PUSH ES
POP DS
MOV DX,5CH
MOV AH,11H
INT 21H
POP DS
CMP AL,0
JZ SEARCH
LEA DX,MSG3
MOV AH,9
INT 21H
JMP QUIT

SEARCH: LEA DX,MSG1
MOV AH,9
INT 21H
MOV BX,80H+1
MOV AH,2
MOV CX,0BH

LOP: MOV DL,ES:[BX]
INC BX
INT 21H
LOOP LOP
LEA DX,MSG2
MOV AH,9
INT 21H
GET_KEY: MOV AH,7
INT 21H
CMP AL,ESC
JZ QUIT
CMP AL,'Y'
JZ ERS
CMP AL,'y'
JZ ERS
CMP AL,'N'
JZ NOTERS
CMP AL,'n'
JZ NOTERS
JMP GET_KEY
ERS: MOV DL,AL
MOV AH,2
INT 21H
CALL CLEA
JMP CONT1
NOTERS: MOV DL,AL
MOV AH,2
INT 21H
CONT1: PUSH DS
PUSH ES
POP DS
MOV DX,5CH
MOV AH,12H
INT 21H
POP DS
CMP AL,0
JZ SEARCH
QUIT: MOV AX,4C00H
INT 21H
RET

```

```

CLEA    PROC NEAR
MOV     BX,80H+1
MOV     CX,08H
MOV     SI,0
A1:     CMP BYTE PTR ES:[BX],''
        JE A2
        MOV DL,ES:[BX]
        MOV FILESPEC[SI],DL
        INC SI
        INC BX
        LOOP A1
A2:     MOV BX,80H+3
        MOV CX,3
        MOV DI,SI
        INC SI
A3:     CMP BYTE PTR ES:[BX],''
        JE A4
        MOV DL,ES:[BX]
        MOV BYTE PTR FILESPEC[SI],DL
        INC BX
        INC SI
        LOOP A3
A4:     MOV BX,80H+9
        CMP BYTE PTR ES:[BX],''
        JE A6
        MOV FILESPEC[DI],''
        MOV FILESPEC[SI],0
A6:     LEA DX,FILESPEC
        MOV AH,3DH
        MOV AL,2
        INT 21H
        JC QUIT
        MOV HANDLE,AX
        MOV BX,AX
        MOV AH,42H
        MOV AL,2
        MOV CX,0
        MOV DX,0
        INT 21H
        MOV CX,1024
        DIV CX
        INC AX
        MOV NUMS,AX
        MOV AH,42H
        MOV DX,0
        MOV CX,0
        MOV AL,0
        INT 21H
        LEA DX,BUF
        MOV CX,1024
        MOV BX,HANDLE
        MOV AH,40H
        INT 21H
        DEC NUMS
        CMP NUMS,0
        JA WRT
        MOV BX,HANDLE
        MOV AH,3EH
        INT 21H
        LEA DX,FILESPEC
        MOV AH,3DH
        MOV AL,2
        INT 21H
        LEA DX,BUF
        MOV CX,0
        MOV BX,HANDLE
        MOV AH,40H
        INT 21H
        MOV BX,HANDLE
        MOV AH,3EH
        INT 21H
        PUSH DS
        PUSH ES
        POP  DS
        MOV DX,80H
        MOV AH,13H
        INT 21H
        POP DS
        RET
CLEA    ENDF
CODE    ENDS
        END START

```

文件；调用写文件功能并预置 CX 寄存器值为零；关闭文件。这些操作使得在文件被删除前，文件长度为零。这样，经过删除操作后，即便是利用具有恢复功能（Undelete）的工具软件也无法恢复该文件。当然，如果需要，还可以对本程序进行改进，调用有关的 DOS 功能，对文件名、文件日期、属性等信息也实行清零，从而更加彻底地消除它。

本程序已在 IBM 和长城系列微机上运行通过。经汇编，连接后即可生成 EXE 文件，再经

由实用程序 EXEPACK 压缩，能得到一个非常短小（仅 2K 左右）但实用方便的运行程序。

使用时，可以象 DIR 命令那样调用它。例如：

要对有 BAK 后缀的文件中的一部分进行删除，则键入

```
FC *.BAK /
```

然后进行相应的筛选，即可完成。

程序清单在前面已给出，读者在维护硬盘时不妨一试。