

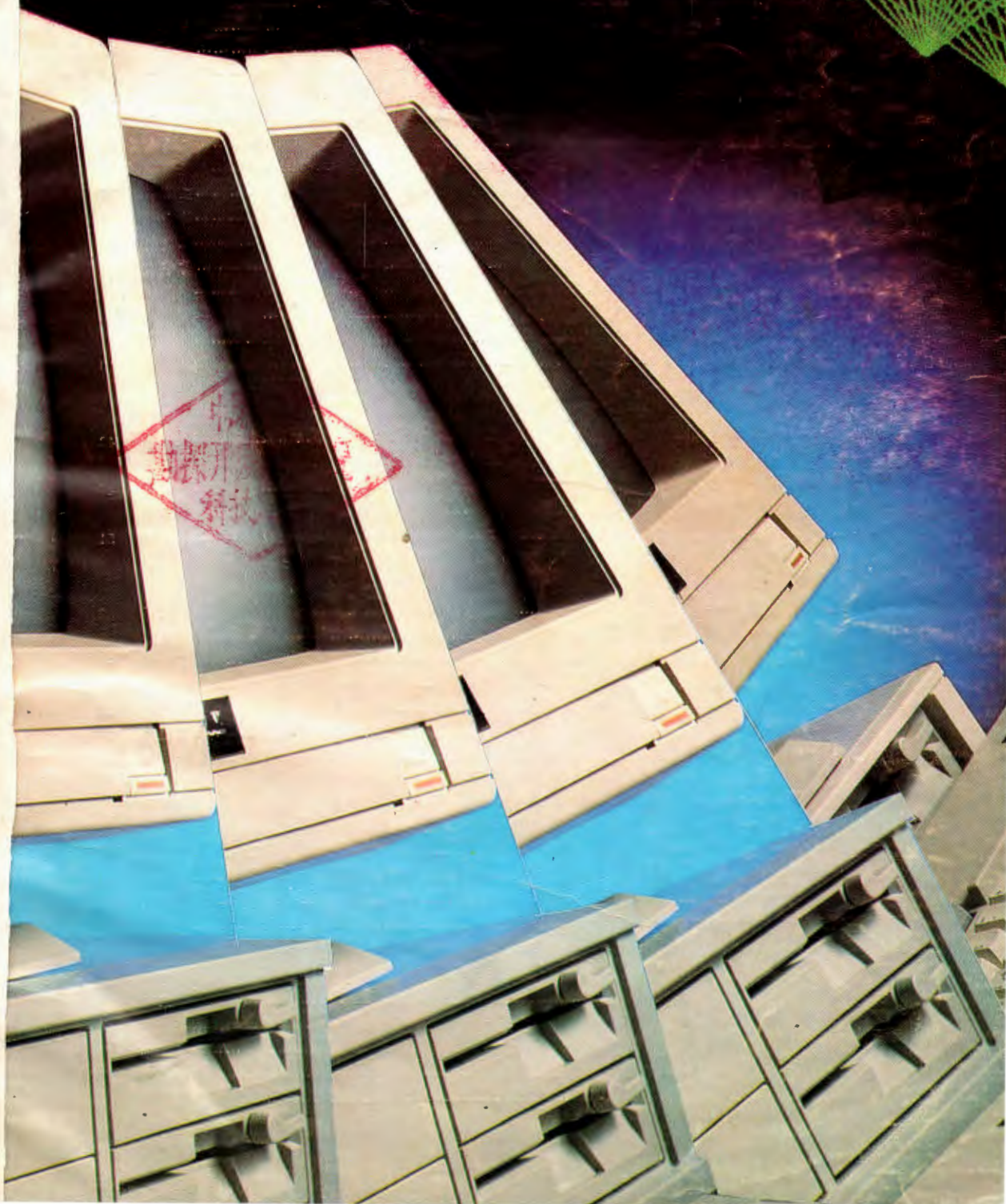
# 計算機應用研究

1990

B61

1

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





### 《計算機應用研究》雜誌辦刊單位

四川省電子計算機應用研究中心 新疆電子計算中心  
貴州省科學技術電子計算機中心 青海省測試計算中心  
安徽省計算中心 甘肅省計算中心  
吉林省計算中心 四川省電子學會

### 《計算機應用研究》雜誌社董事會

董事長：周寶渝  
董 事：唐 珍 鄭國基 陸慰椿  
秦小竹 龔宇清 黎曉常

### 《計算機應用研究》雜誌編輯委員會

主 任 委 員：張執謙  
副主任委員：李澤民  
委 員：余 凱 張國棟 賈洪鈞  
曾光初 王小華 朱景生

1990年第1期（總第33期）

出版日期：1990年1月 本期責任編輯：鄧嘉澍

計算機應用研究（雙月刊）

（公開發行）

國內統一刊號：CN51—1196

主 編：張執謙

副主編：李澤民

編輯出版：《計算機應用研究》雜誌社

通訊地址：成都市人民南路4段11號附1號

郵政編碼：610015

印 刷：西南冶金地質印刷廠

訂 閱 處：全國各地郵局

總 發 行：成都市郵政局

郵發代號：62—68



# 《计算机应用研究》第7卷第1期 (总第33期)

## 目 录

### 软 件 篇

C程序调用中的参量传递.....	贾洪钧 ( 1 )
用BASIC语言直接读取DBASE—Ⅲ数据库数据的方法.....	唐珍等 ( 4 )
负荷传感器的动态测试.....	郑国基等 ( 6 )
单回路智能仪表应用程序设计.....	周立恒等 ( 8 )
单纯形加速法与计算机控制系统的PID最优化仿真设计.....	李天健 ( 10 )
传输控制及网络管理.....	张鸿鸣等 ( 14 )
微机医疗专家系统.....	张鸿鸣等 ( 16 )
高精度多字体显示器画面构造程序.....	李卫元 ( 18 )
一个简易画空、实心框图程序.....	龙兵生 ( 21 )
能够在微机上快速实现CRC校验的一种方法.....	袁兴志等 ( 23 )
中心提升——从数据流图到模块结构图的直接转换方法.....	王治宇 ( 26 )
一个微机“病毒”的感染、激活原理及其防治办法.....	马小宏 ( 32 )
计算机圆点“病毒”的诊断及排除.....	王庆庆 ( 34 )
字库缩减及造字.....	雷学武 ( 36 )
浅谈利用高级字库3KTZ打印特大汉字.....	龙兵生 ( 37 )
关于《利用BASIC产生屏幕美术字的编程技巧》之我见.....	李辉宏 ( 38 )
丰富3070的打印机实线打印.....	魏旭 ( 39 )
SAP6软件顾问系统概述.....	林丕等 ( 40 )

### 硬 件 篇

单回路智能仪表硬件系统设计.....	周立恒等 ( 42 )
APPLEⅡe微机体感诱发发电位检测系统硬件接口设计.....	彭利安等 ( 45 )
一种实时中断键盘接口电路.....	何杰 ( 46 )
IBM—PC机并行口简易扩展器.....	韩知恩等 ( 49 )
计算器应用一例.....	华金保 ( 51 )

### 维 修 篇

外设常见故障与排除.....	陈兰英 ( 52 )
使用FTU调整磁头的方法.....	陈兰英 ( 53 )
0520CH—Ⅱ计算机电源故障一例.....	华金保 ( 54 )
GW0520—CHⅡ硬盘维修一例.....	叶志斌 ( 55 )
3070打印机打印针磨损的处理方法.....	周维寿 ( 56 )
3070打印机断针的简易修复.....	苏亚华 ( 56 )
用注射法使pp40打印机笔更新.....	沈友敏 ( 57 )

### 系 统 篇

APPLEⅡ内外存储扩充技术和提高运行速度方法研究及实例分析.....	沈欣等 ( 58 )
多路数据采集器与微机的软硬件接口.....	陈启录 ( 61 )
微机在小氮肥厂中的应用综述.....	彭献彬 ( 64 )
E <sup>2</sup> PROM2864的使用体会.....	张向春 ( 69 )
实现MC—51单片机单步操作的简易方法.....	周鸣争 ( 70 )

### 信 息 篇

人民银行卫星通信网络加紧施工.....	桑小军等 ( 71 )
我国用微机控制电力系统取得突破——南京地调自动化系统通过部级验收.....	徐洁 ( 72 )
仓库基础数据管理系统通过部级鉴定.....	阮高华 ( 72 )
安徽省计算中心部分科研成果简介.....	顾福良 ( 72 )
信息五则.....	(9, 44, 50, 封三)

## 致读者·作者

——本刊编辑部

尊敬的读者、作者，本刊自创刊至今一致得到你的爱护和帮助，以及你的单位的热情支持，使《计算机应用研究》这本学术技术性刊物得以越办越好、有了很好的起色，为广大读者所喜爱。我们，作为你们桥梁的编辑部怎能不感到由衷的高兴呢。值此新春即将到来之际，我们特留此一角之地，向你和你的单位表示衷心的感谢和敬意。并祝愿在新的一年里取得更大的成就，为振兴中华做出卓越贡献。

同时也高兴的告诉你，本刊为实现广大读者和作者的心愿，从1990年第一期始封面改为彩色封面，封面设计与本刊的学术技术性质交相溶汇，给人以迅速的直觉感，相信你一定会喜欢。另外也请你转告你的单位。本刊早就获准经营广告业务，封面彩印的水平和效果将同样使你们的成果一展风采，使你们的产品在本刊公开发行的各个角落大展宏途。

### 一片变两片——软磁盘打孔器

目前我国许多型号的微机（如紫金Ⅱ、苹果Ⅱ、中华学习机、LASER310等）所使用的都是单驱动器，而双面软盘片现在正越来越多地投入计算机市场，这种软盘在单面驱动器上只能使用一面，这样就造成了相当大的浪费。为了解决这一问题，有不少同志用剪刀等器物在双面软盘的另一侧开另一个写保护孔，以便双面软盘能反过来使用，从而扩充软盘容量，这种开孔方法的问题是：开的孔有毛边，定位难掌握，甚至会损伤磁盘。

我部生产的‘磁盘打孔器’能很方便地为您解决这一问题，使您手中的双面双密度磁盘在苹果机上完全发挥出它的性能，可以轻而易举地使一片软盘正反使用：从而达到‘一片变两片，容量扩大一倍’的功效。

磁盘打孔器的使用非常简单，它带有限位片，因此您无需担心打孔的位置不准确，定好位后只需稍稍用力一按，便在您的磁盘上打出一个没有毛边、位置合适、大小正好写保护孔，一秒钟时间都不到，您手中的磁盘就可以正反两面使用了，经过长时间使用证明：打孔后的另一面磁盘性能完全与正面相同，一片变两片，容量扩大一倍，为您节省开支，诸君不妨一试，保你一试便灵。

软磁盘打孔器售价：9.8元/只 邮购另加邮杂费1.30元

100只以上优惠价：8.8元/只

1000只以上优惠价：7.8元/只

欲购者请按如下地址汇款：南京市向往电脑服务部

开户银行：市农营水西门分理处 帐号：7701501032

本部地址：南京市升州路151号 电话：625263 联系人：邵发森 黄 健

# APPLICATION RESEARCH OF COMPUTERS

Vol.7 No.1 (Total 33)

## CONTENTS

### THE ARTICLES ON SOFTWARE

The Parametric Transmission in C Program Calls.....	Jia Hongjun ( 1 )
The Mehthod for Directly Readed Data by BASIC Language in DBASE—Ⅱ Data Bank.....	Tang Zhen and others ( 4 )
Dynamic Measurement of Carrier Sensor.....	Zheng Guoji and others ( 6 )
Designing About Application Program of Single—Loop Intelligent Instrument.....	Yan Liheng and others ( 8 )
Speedding Method of Plain Figure and Optimal Emulating Design of PID Parameter in the System by Computer Control.....	Li Tanjan ( 10 )
Transmission Control and Network Management .....	Zhang Hongming and others ( 14 )
The Therapeutist System of Microcomputer .....	Zhang HongMing and others ( 16 )
The Program for Producing Display Picture of High—Precision and Multile—Character Style .....	Li weiyuan ( 18 )
A Simple Program of Drawing Flow—Chart Graph.....	Long Binsheng ( 21 )
The Method for Rapidly Realized CRC Verification by Microcomputer .....	Yuan Xingzhi and others ( 23 )
Center Lifting the Method of Direct Conversion from Data—Flow Diagram to Module Structure Diagram.....	Wang Zhlyu ( 26 )
Infection and Activation Principle of Microcomputer Virus and It's Method of Prevention.....	Ma Xiaohong ( 32 )
Computer Virus—Diagnosing and Debugging.....	Wang Qingqing ( 34 )
Reduce Chinese LibRARY and Made—Words.....	Lai Xuewu ( 36 )
Simple Introducing about Print of Super giant Chinese by High Word— Bank 3 KTZ.....	Long Bingsheng ( 37 )
Discussing to Editing Skill of Producing Artistic Words on Screen with BASIC.....	Li Huihong ( 38 )
Expansion of Continuous Line Print on 3070 Printer.....	Wei Xu ( 39 )
An Outline of Consulting System for SAP6 Software.....	Lin Pi Ren Hai ( 40 )

### THE ARTICLES ON HARDWARE

Designing about Hardware System of Single—Loop Intelligent Instrument.....	Yan Liheng and others ( 42 )
---	------------------------------

## Designing about Hardware Interface of Detected System of Body Effect

- Evoked Potential on APPLE—IIe .....Peng Lian and others ( 45 )  
A Circuit of Keyboard Interface with Real-time Interrupt.....He Jie ( 46 )  
A Parallel Interface of Expansive Circuit for Model IBM—PC  
Microcomputer.....Ha n Zhien and others ( 49 )  
One of the Methods Used Calculator.....Hua Jinbao ( 51 )

## THE ARTICLES ON MAINTENANCE

- The Usual Malfunction and It's Removing on Peripheral Equipment  
.....Chen Lanying ( 52 )  
The Method for Adjusted Magnetic Head by FTU.....Chen Lanying ( 53 )  
An Example about Maintenance of Power Supply for 0520 CH—I  
.....Hua Jinbao ( 54 )  
An Example about Maintenance of the Fixed Disk Malfunction on  
GW0520—CH.....Ye Zhibing ( 55 )  
The Resolved Method of Pin Wearing on Printer of Model 3070  
.....Zhou Weishou ( 56 )  
A Simple Method of Reparation for Break Wire on the Printer of  
Model 3070.....Su Yahua ( 56 )  
Renewing the Marking Pen for PP40 by Injector.....Shen Youmin ( 57 )

## THE ARTICLES ON SYSTEM

- Research for the Expand Technique of Internal Storage Capacity and  
the Method of Increase Operation Speed and Analysis of Living  
Example.....Shen Xin and others ( 58 )  
The Multiway Data Collector and Interface between software and  
Hardware.....Chen Qilu ( 61 )  
Application Overview of Microcomputer in Nitrogenous Fertilizer  
Factory.....Pen Xianbing ( 64 )  
Application Experience of E<sup>2</sup> PROM2864.....Zhang Xiangchun ( 69 )  
A Simply Method for MC—51 Single—Chip Microcomputer Single—Step  
Operation.....Zhou Mingzheng ( 70 )

## THE ARTICLES ON INFORMATIONS

- Satellite Communication Network of People's Bank is Building Rapidly  
.....Sang Xiaojin and others ( 71 )  
A Breakthrough for Electric Power Controlled System by Microcomputer  
in our Country—Automatic Dispatching System of Power Network  
Adopted Ministerial Level Check and Accept in Nanjing Region  
.....Xu Jie ( 72 )  
Base Data Management System of Repertory Adopted Ministerial  
Level Appraisal.....Ruan Gaohua ( 72 )  
Simple Introducing about Partial Product in Scientific Research to  
Computer Centre of Anhui.....Gu Fuliang ( 72 )  
Five Pieces of News..... Our's Reporter ( 9, 44, 50, Cover Third)

# C程序调用中的参量传递

贵州省科委计算中心 贾洪钧

**摘要** 本文阐述了C语言程序调用的一般方法,在分析调用时的程序参量传递基础上,利用C例行程序库所提供的管道函数及数据共享管理函数,成功地实现了施调程序与被调程序间参量的相互传递。本文所探讨和提供的方法,对进行C程序设计具有实际使用价值和参考价值。

一般地说,C语言程序可以调用相同支撑环境下的其它可执行文件,这对于提高用C语言编程的效率,增强C程序结构的灵活性等都是极为有效的。为方便说明问题,本文所讨论问题的举例均采用C语言源程序进行,至于被调用的程序是其它源语言(如PASCAL, FORTRAN, ASSEMBLER等)的可执行文件的场合,本文的结论和方法可以推而广之地使用。

## 一、C调用序列

C编译器在处理C程序调用其他程序的语句时,会自动地在该调用前、后嵌入一个C调用序列,其主要功能是:

1. 在转入执行被调程序前,以连续的入栈操作将那些决定施调程序当前状态的寄存器的内容送入系统堆栈保护起来,称之为C调用进入序列。若在286 XENIX SystemV中,C调用进入序列用286宏汇编语句可以写作:

```
push bp
mov bp, sp
push si
push di
```

2. 在返回继续执行施调程序前,以连续的出栈操作将系统堆栈中所保存的当前施调程序的状态信息回送到相应的寄存器中,称之为C调用返回序列。同样可将其写作:

```
pop di
pop si
mov sp, bp
pop bp
```

对于程序所定义使用的参量,C调用序列是不理睬的,因此,它是不处理施调程序和被调程序间参量的相互传递的。要实现这种参量的相互传递,必须采用特殊的处理。

## 二、C程序调用方式及参量传递特点

C程序调用其他程序时,通常采用下述两种

调用方式:

### 1. 函数调用方式

函数调用方式要求被调程序必须而且能够定义为C函数文件,它必须以源文件的形式与施调程序源文件连续作编译,或以目标文件的形式经编译预处理指令指示嵌入施调程序内进行编译。而调用它前,施调程序还必须对它作先行申报。

若被调函数(程序)名为: **funcname** (**formarglist**),则调用语句为: **funcname** (**realarglist**);

函数调用发生时,系统首先执行C调用进入序列,并将调用语句的**realarglist**所指出的施调程序相应参量的当前值(称实际参数)复制到函数定义的**formanglist**所指出的被调程序相应参量的寄存器中(称形式参量)提交被调程序使用,然后才转入执行被调程序。一待被调程序运行完毕,系统立即执行C调用返回序列,转回施调程序继续执行。

由于被调程序使用的是所需的被调程序参量当前值的复制品而非这些参量本身,即使被调程序运行中对这些复制品作了改动,也不会影响原件。可见,函数调用方式中程序参量只满足从施调至被调程序的单方向传递。除非采用特殊技巧,灵活使用变量指针,才能达到参量双向传递的目的。

实例1:

```

↑ # include<stdio.h>
施 int a, b;
调 int rut (int*, int*);
程 main ( )
序 {
  a=5;
  b=7;
  printf( "1-- %d, %d\n,"a,b),
  rut (&a, &b);
  printf( "4-- %d,%d\n," a,b);
  ↓ }

```

```

↑  rut (x, y)
被  int *x, *y;
调  {
函  printf ("2--%d,%d\n" *x, *y);
数  *x=12;
    *y=3;
    printf ("3--%d,%d\n" *x, *y);
↓  }

```

执行结果, 面板显示:

```

1-- 5, 7
2-- 5, 7
3-- 12, 3
4-- 12, 3

```

由于函数调用方式必须将施调程序与被调程序连接成一个整体, 这就降低了程序模块的分散式存储结构, 而当被调程序数较多时, 会急剧增加程序对存储空间的要求, 甚至造成编译失败。

## 2. 进程创建方式

“进程”可被看作是一个正在执行的程序。如同在操作系统shell提示符下打入可执行程序名来激活一个程序运行(称创建一个进程)一样, 也可以在C程序中调用C库的进程控制函数来创建其他进程从而达到调用其他程序的目的。在相同支撑环境下的任何一个可执行程序都可以被创建成进程, 至于这个可执行程序本身是用哪种源语言实现, 倒无关紧要。只要在创建该进程时(即调用该程序时), 确切地指明该可执行文件的完全路径名即可。例如 /usr/filename等。

常用来创建进程的C库函数有:

### 1. system函数 (仅限XENIX环境下)

其调用格式为: `system ("/usr/filename")`;

由system函数调用的程序在其执行完毕后, 系统将返回施调程序继续执行。

### 2. exec函数

其调用格式为: `exec("/usr/filename", filename)`;

或 `execv("/usr/filename")`;

由exec函数调用的程序在其执行完后, 不管施调程序是否执行完, 控制都返回系统。

由于进程创建方式是通过调用进程创建函数——其实际参数是欲调用程序的可执行文件的完

全路径名——来实现对其它程序的调用的, 施调程序与被调程序间形式参量与实际参量的概念在此已失去意义。实际上, 施调程序与被调程序间是不发生参量传递的。

然而在许多场合下, 施调程序与被调程序都必须根据相互间传递的参量来决定程序流向, 二者之间参量双向传递的问题就显得十分重要了。

## 三、参量传递方法的探讨

可以断言, C程序调用中是可以设法实现施调程序与被调程序间参量的相互传递的。下面对我们在进行C程序设计中所采用的几种办法加以说明, 所举的例子均采用 XENIX system V 的C语言编制, 以供参考。

### 1. 磁盘文件法

利用标准C库的文件管理函数, 在磁盘上建立一个用来存放要传递的参量的文件。在施调程序和被调程序中适当地安排对该文件的访问, 即可有效地实现数个程序的参量的双向传递。这种方法简单易行, 但数据传递速度较慢。

### 2. 管道文件法

使用标准C库所提供的管道函数, 在内存中开辟一块用于参量传递的区域(管道), 适当地安排施调程序和被调程序对该管道的访问, 即可实现程序参量的双向传递。

实例2:

```

↑  # include <stdio.h>
   main ( )
   {
   施 FILE *pstrm;
   调 char chr [5];
   程 strcpy (chr, "hello", 5);
   序 printf ("0 %s\n", chr)
   pop pstrm=popen ("/pop1", "w");
   printf (pstrm, "%s", chr);
   pclose (pstrm);
   printf ("2 %s\n", chr);
   pstrm=popen ("/pop2", "r");
   fscanf(pstrm, "%s", chr);
   pclose (pstrm);
   printf ("3 %s\n", chr);
   ↓ }

```



```

↑ 被调程序
main ( )
{
char cs [5];
get (cs);
pop1 printf ("1%s\n", cs);
↓ }

↑ 被调程序
main ( )
{
char chr [5];
strcpy (chr, "yours");
pop2 printf ("%s\n", chr);
↓ }

```

执行结果:

```

0 hello
1 hello
2 hello
3 yours

```

建立程序间通讯管道也是调用程序的方法之一。一旦施调程序打开了通往某个被调程序的管道(如例1中语句`pstrm=popen("/pop1", "w");`),系统将立即装入并执行该被调程序,直到其执行完毕后才返回施调程序继续执行。这一事实说明施调程序无须再采用其他办法即可调用所需程序。但对于被调程序而言,管道实际上是一个高速的标准 I/O 文件(相应于打开管道时的 w/r 方式),因此,被调程序在使用标准 I/O 操作时应特别小心,以免引起误传输或 I/O 操作失灵。

### 3. 共享数据区法

使用扩展 C 库的共享数据管理文件 `<sd.h>`, 由施调程序在内存中开辟一个专用区域来存放欲传递的各个参量,适当地安排对该区各字节的访问,就可以达到在程序间实现参量双向传递的目的。

实例3:

```

↑ 施调程序
# include <sys/sd.h>
# include <ctype.h>
char *as;
main ( )
{
char *as, *a, *b;
*a='5';
as=sdget ("/tmp/share",

```

```

SD_CREAT | SD_WRITE | SD_
UNLOCK, IL, 0777);
sdenter (as);
strcpy (as, a);
strcpy (b, as);
sdleave (as);
printf ("b=%c\n", ) b;
system ("/ calledprog");
sdenter (as);
strcpy (b, as);
sdleave (as);
printf ("finish b=%c\n", b);
↓ }

```

```

↑ 被调程序
# include <sys/sd, h>
# include <ctype, h>
extern *as;
main ( )
{
char *as, *c, d;
*d='8';
as=sdget ("/tmp/share", SD+
WRITE | SD_UNCLOCK);
sdenter (as);
strcpy (c, as);
strcpy (as, d);
sdleave (as);
printf ("middle c=%c\n", c);
return;
}
↓

```

执行结果:

```

b=5
middle c=5
finish b=8

```

采用共享数据区法的优点在于程序对它的访问虽如同访问一个磁盘文件,但访问速度极快而不影响任何其它操作。当被调程序结束后,该区则保留了最后一次访问操作时的最终结果以供给其它程序包括施调程序或其它被调程序使用。

### 4. 系统堆栈法

前面提到, C 调用序列仅通过系统堆栈操作实现对施调程序状态的保护与恢复。由于 C 语言留有同汇编语言的界面。因而在欲传递的参量数

# 用 BASIC 语言直接读取 dbase—Ⅲ 数据库数据的方法

甘肃省计算中心

唐 珍 胡诗秀

甘肃省膜科学研究所

唐润寰

## 一、引言

通常人们在微机上对数据进行建库、录入和维护时,大多应用 dbase—Ⅲ。但要对数据进行编辑处理或利用该数据进行绘图等工作时,由于就 dbase—Ⅲ 本身来说,这些功能不强或没有,所以,一般是用其内部的 COPY 命令将数据拷贝成 .TXT 后缀的文本文件,然后在系统环境下用其它语言(如BASIC, FORTRAN)调用 .TXT 文件来读取所需的数据。这种方法给用户造成系统时间和磁盘空间的大量浪费,并给使用者带来不便。

## 二、综合分析

笔者通过分析 dbase—Ⅲ 数据库的结构,得出一种比较简单的用BASIC语言读取 dbase—Ⅲ 数据的方法。本方法克服了以前诸多读取方法的缺点。首先,以前有的按随机方式打开文件来读取 dbase—Ⅲ 数据库的方法,但该方法只对满足  $(\text{字段个数} + 1) \times 32 + 2$  个字节(或 + 3 个字节)能被  $(\text{字段长度之和} + 1)$  整除条件的数据库才比较简单。若不满足上述条件,则要对读出来的数据进行分割和拼接处理,这样一来,编制出的程序则较长且较复杂。其次,以前有的

按顺序方式打开文件的方法,然后用 INPUT (记录长度,文件号)来读取定长字符串,把标记库文件结构的部分读掉,再将需要的记录部分,按在 dbase—Ⅲ 状态下获得的记录长度一一读出。上述两种方法的共同缺点是都只能针对每一个具体的数据库来编制程序,即每读取一个数据库就需要一个专门的读取程序,因此通用性差。

本方法直接按顺序方式打开文件,并且不采用通常的 LINE INPUT,而改用定长的 INPUT (32, \*2),一次读取 32 个字节,避免了对数据库中超过 254 个字符的行在读取时发生错误。同时我们采用直接从库结构中获取信息的方法,直接获取记录长度,从而使程序成为通用的。另外,在程序中加入错误捕获语句,使在读取库结构部分时消除可能出现的文件结束符 (1A),以便使读取工作正常进行下去。

## 三、库结构分析

dbase—Ⅲ 数据文件存储的格式可分为三大部分。

第一部分:文件标识信息部分

这部分固定占用前 32 个字节。其中部分字节如下:(均为 16 进制数,下同)

目不超过可供使用的通用寄存器数目的情况下,可以将这些参量定义为 register 型,并紧随调用进入序列之后用连续的压栈操作将它们送入系统堆栈。进入被调程序后,可采用汇编语句按其存放地址和参量所需长度来访问堆栈地址(注意不要拨动堆栈指针)。在调用退出序列执行前再依次进行弹栈操作将其送回原来定义的寄存器内来达到程序间参量双向传递的目的。但因这种方法需要考虑好几个界面,特别是当被调程

序是非汇编语言源程序的可执行文件时,实现起来比较困难,所以很少采用。

综上所述,C 程序调用中施调程序与被调程序间程序参量的双向传递是完全可以进行的。尽管实现此种传递的方法各有千秋,难易不一,但都可以达到目的。作为探讨,所列举的方法不一定全面,缺陷在所难免,但只要对之稍作改动以满足特定传输要求,即可用于其它实际的调用参量传递的场合。欢迎有意者于此者不吝赐正。

第一字节: 03H或83H表示是dbase—Ⅱ数据库文件。

第二字节: 文件建立或最后修改的年份。

第三字节: 文件建立或最后修改的月份。

第四字节: 文件建立或最后修改的日期。

第五至十字节: 以二进制数值表示的文件中包含的记录个数, 其中: 低位字节在前, 高位字节在后。

第十一至十二字节: 以二进制数值表示的文件中每条记录的长度。

#### 第二部分: 字段描述信息部分

从第33个字节开始, 每一个字段的描述信息, 固定地占用32个字节, 各字段按顺序存储, 在最后一个描述信息后, 紧接着一个(0DH)回车符作为第二部分结束。

每个字段的的部分字节如下:

第一至十字节: 字段名

第十二字节: 字段数据类型

第十七字节: 字段宽度

第十八字节: 字段的小数位数

#### 第三部分: 数据记录存储部分

从第二部分结束字节(0DH), 再加一个空字节(00H)开始, 以记录长度格式顺序存放数据库的全部数据记录。每条记录起始第一字节是留给删除标记用的, 然后是记录的内容, 记录之间没有任何分隔符, 最后一条记录存放完后, 是一个文件结束标志(1AH)。

### 四、程序结构及清单

知道了dbase—Ⅱ的主要结构, 就可以读取用户所需求的数据库文件。下面的BASIC程序, 只需输入dbase—Ⅱ数据库文件的名字即可读取其数据。程序分为三大部分: 第一部分150—240行为读取各字段的长度信息。第二部分270—300行读取dbase—Ⅱ的文件标识信息部分和字段描述信息部分, 使读取指针指向数据记录的开始。第三部分310—360行为把各字段的数据读出后按行输出。程序在IBMPc/XT上CCDOS环境下通过。

```
100 DIM A%(128),B$(128)
110 INPUT "DBASE NAME [ .dbf] : ";F$
120 F$=F$+".dbf" : V=1
130 ON ERROR GOTO 420
140 OPEN F$ FOR INPUT AS #2
150 FOR K=1 TO 128
160 C$=INPUT$(32,#2)
170 D=ASC(MID$(C$,1,1))
180 IF D = 13 THEN 250
190 E=ASC(MID$(C$,17,1))
200 IF K=1 THEN 240
210 AX(V)=E
220 X=V
230 V=V+1
240 NEXT K
250 CLOSE #2
260 OPEN F$ FOR INPUT AS #2
270 FOR I=1 TO X
280 W$=INPUT$(32,#2)
290 NEXT I
300 W$=INPUT$(35,#2)
310 FOR J=1 TO 5000
320 FOR I=1 TO X
330 B$(I)=INPUT$(AX(I),#2)
340 PRINT B$(I);
350 NEXT I
360 PRINT
370 IF EOF(2) THEN 400
380 W$=INPUT$(1,#2)
390 NEXT J
400 CLOSE #2
410 END
420 IF ERR=62 THEN RESUME NEXT
430 PRINT "File Name Error !"
440 BEEP : GOTO 110
```

### 参考资料

1. 魏万明:《BASIC语言直接调用dbaseⅡ和Ⅲ的数据》,《微型计算机》1987.6.
2. 刘维平:《用BASIC程序处理非ASCII文件的一种方法》,《计算机世界》,1987.10.
3. 辛德强:《用BASIC语言直接读写dbaseⅡ,dbaseⅢ数据库文件》,《微计算机应用》,1988.1
4. 罗文彬、卢玉广:《dbaseⅢDBF文件中数据的安排》,《微型机与应用》,1988.2.



# 负荷传感器的动态测试

吉林省计算机技术研究所  
长春衡器工业公司

郑国基 徐世坤  
郑明杨

## 一、负荷传感器的一般测试:

负荷传感器的线性误差、重复性误差及滞后误差是表征传感器质量的三个主要参数。传感器的弹性体上贴有电阻应变片,将这些应变片组成惠斯顿电桥,并在电桥的对角端加上恒定电压。另外,两端接入数字电压表,当传感器受力时,电桥失去平衡,接数字电压表的两端就有电压输出。负荷传感器测试的一般方法是将被测传感器装在标准测力机上进行加载和卸载。在零载和满载之间等分成若干测试点,在每个测试点保持载荷,读取传感器的输出值。重复三次加、卸载,可测得三十多个数据。最后对这些数据进行计算和处理,先算出平均值、理论值,最后求得线性误差,重复性误差和滞后误差。

这样重复测量量、繁琐计算,一支传感器的测试一般需要一小时以上。而且还要拿到有标准测力机单位进行。这对于大量生产传感器的厂家就急需研制出造价低廉并能快速、自动测试负荷传感器的专用系统。

在美国和西德的某些厂家已采用动态测试系统对传感器进行测试。但在我国还没有采用这种方法,有些单位只是用静态加、卸载对所取的数据进行计算机处理。

## 二、负荷传感器的动态测试基本原理:

几年来,吉林省计算机研究所与长春衡器总厂共同研制了“SB—50型负荷传感器动态测试系统”,并于1987年2月安装调试完毕,已运行五百余次,完成贰百多次,30T、50T负荷传感器的测试任务,试运行表明:该系统性能稳定,使用情况良好,1988年4月27日由吉林省科委主持召开了该系统鉴定会。鉴定结论指出:SB—50型负荷传感器动态测试系统可以对30T、50T负荷传感器的线性、重复性、滞后三项指标进行

测试,整个系统测试精度不低于0.02%FS,已达到预期的要求,该系统由于采用快速仪表和计算机技术,所以测试速度快精度高,并能及时对数据进行处理和打印。

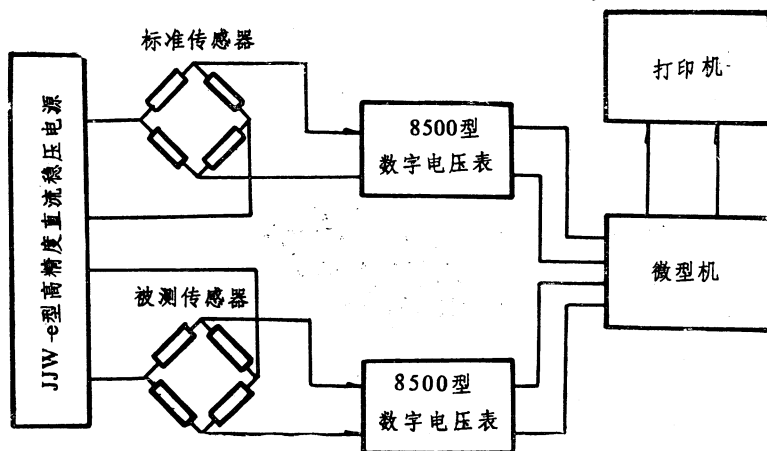
系统设计合理、实用、适合生产传感器厂家作质量控制使用,具有明显的经济效益。

该系统的研制成功,在负荷传感器动态测试方面填补国家一项空白,居国家领先地位。

该系统的详本原理:

在一台50T的液压加力机上,对串联的标准传感器与被测传感器平稳连续加力,标准传感器是西德HBM公司生产的C3H3型高精度传感器。它在国家基准测力机上标定过,两个传感器由高稳定度,直流电源供桥压,其输出各接一块美国FLUKE公司的8500型数字万用表。这是具有六位半的快速采样可编程数字万用表。预先把标准传感器各标准点的值存储在微型机中,两个表的输出通过并行接口和微型机相连,两个传感器在加、卸载过程中的电桥输出通过8500型表高速采样,传到微型机中,CPU把标准传感器的值和预置的第一标准点值比较,如果为加载过程,那么小于标准点值的数据 $V_1$ 留下,再把第二个传来的值 $V_2$ 和标准点的值比较。如果小于标准值,则 $V_1$ 丢掉 $V_2$ 变为 $V_1$ ,再将新传来的 $V_2$ 与标准值比较,如此继续下去,在同时被测传感器通过另一块8500数字电压表快速采样,把数据通过并行接口传送给微型机。CPU把相邻的两个数再放入前面两个数所在的内存相邻位置,如此继续下去。

当标准传感器传入微型机的数大于或等于标准点T时,CPU便把被测传感器在T前传入和T时后传入的两个数求平均值作为被测传感器的值保留下来,标准传感器再传入的数,CPU便把它同第二个标准点相比较,如此继续下去,就可以得到被测传感器的各点数据。



SB-50型负荷传感器动态测试系统原理图

连续三次加、卸载后，便可以得到被测传感器的各点数据。微型机把采集的数据进行处理和计算，最后通过并行口传送给打印机，按要求的格式打印输出。一支负荷传感器的测量和打印输出完成仅需10分钟左右。

### 三、系统的误差分析：

1. 标准传感器选用的是西德HBM公司C<sub>3</sub>H3型传感器，在国家基准测力机上用静加载标出标准值，其重复性误差小于0.01%。

2. FLUKE公司的8500表应用100mv的量值，采样用异步连续形式，速度在2.2ms/次。当从零开始平稳地加载到满载计一分钟时间，8500表可以采数27000次以上。这样由两块表采样所产生的不同步误差可以忽略不计。两块8500表在两年时间曾三次送北京FLUKE公司维修站标定，校准其精度在0.002%。

3. 高精度稳压电源。选用的是国家计量局，北京计量仪器厂生产的JJW-e型高精度直流稳压电源。两个传感器由同一台电源的两路供给。其稳压系数在电网电压 $220\text{V} \pm 10\%$ 范围内 $S \leq 3 \times 10^{-6}$ 。时间漂移，当温度为 $20^\circ\text{C} \pm 0.5^\circ\text{C}$ 时， $S \leq 3 \times 10^{-6}$ 分钟。

4. 系统零点漂移的校正：在每次测试传感器时先对标准传感器及桥压和电压表的零漂进行检测。微型机把漂移因素考虑进去，修改标准传感器的各标准点的值，这就消除了由于系统零漂给测试所造成的误差。

5. 由于标准传感器与被测传感器受力状态为串联，在同一时刻，两个传感器所受力相等。

6. 力源系统。采用双向油缸结构，装有蓄能器，消除了油泵出来的油产生的脉动，使加力过程非常平稳，大大提高了整个系统的精度。

### 四、软件设计

整个软件由下列几个模块组成。

1. 对两个数字电压表（8500）进行初始化的程序。
2. 传感器‘0’点数据采集程序。
3. 加、卸载过程对两个传感器数据采集程序。
4. 系统‘0’点漂移的校正程序。
5. 输出打印程序。
6. 数据处理计算程序。
7. 主程序。

# 单回路智能仪表应用程序设计

吉林省计算机技术研究所

阎立恒 刘晓光 孟刚

**前言：**单回路智能仪表应用范围很广，应用软件的设计和编写，只限于现场应用过的情况进行的，其它各种应用软件和数据处理有待遇到实际问题时再一一进行。

DZY—1型智能仪表应用在金属热加工中有：淬火炉、退火炉、盐炉和玻璃工业中槽沉炉、回火炉的控制。有关检测、自动控制方面应用软件总体设计和数据处理的方法，作一简单介绍，供读者研究和参考。

## 一、应用软件设计

淬火炉、退火炉、盐炉、槽沉炉等均为中温 $500^{\circ}\text{C}$ — $1000^{\circ}\text{C}$ 用电加热的设备。为了生产各种产品，要求电炉的温度满足各种产品的加工工艺。我们就是以电炉加工的产品工艺要求为基础，设计了通用的应用软件和数据处理方法。仅适用于上述炉型和中温的工艺要求，在此条件下，仅仅变动几个参数，就适合于要求的加工工艺，免去那些重复劳动。经过实际现场应用，设计的应用软件和数据处理方法，其效果满足了上述炉型、加工几十种产品的工艺要求，均达到了满意的效果。

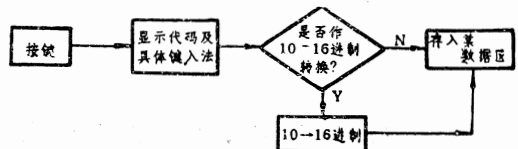
应用软件程序包括：用电设备的温度控制有：升温段、恒温段、降温段和数据处理方法。其中控制电炉温度是共同的目的，设计编写了主程序，而各种电炉加工产品不同，有了特殊要求，其中把那些特殊要求归纳到灵活处理参数，可单独键立，以便达到各自的要求。“第三部分”输入代码及对应参数表的使用，就是实现一种电炉生产同一种产品，工艺要求是相同的，那么可键入一次参数；如果一种电炉（或不同种）生产不同种产品，工艺要求不相同，那么可键入不同的参数。实现了在满足电炉加工产品条件下，不再设计编写重复程序，而只要键入十六个参数，便可实现各自不同要求，达到预想目的。

控制电炉温度的主程序包括：升温、恒温、

降温。软件程序和数据处理方法，在单回路智能仪表中得到了实现，用智能仪表控制的电炉作到了既能保证产品质量、节约用电、延长炉龄，又能改善劳动环境、减轻劳动强度，这都是先进的现代化智能仪表优异性所在。

## 二、人一机对话输入参数数据处理

关于人一机对话部分参数的输入，采用十进制整数/小数和十六进制整数混合输入方式，共输入15个参数，如下图所示：



开始设计时，试用全部参数均采用10进制数的输入方式，后经实践发现十进制数方式，对于现场需要长时间恒温的控制是达不到要求的，为了克服这一不足，而采用十六进制。选用的单位是十分钟。若用十进制输入，最大值为9999，即 $9999 \times 10\text{分} = 99990\text{分} = 1666.5\text{小时} = 69.4\text{天}$ ；若用十六进制输入，最大值为FFFFH，即 $65535 \times 10\text{分} = 655350\text{分} = 10922.5\text{小时} = 404.7\text{天}$ 。可见表示时间如此之长，可以满足各种用电设备的恒温要求，故采用后种方式，其它参数也有类似的情况。

## 三、输入代码及对应参数表如下：

（见下页）

以上这些参数可以一个接着一个的输入，也可为了方便以任意次序输入，但不可不输入，对于有相同种工艺产品的电炉，参数输入一次便可以。且可以反复输入，以最后一次输入为准。

## 四、模拟量的输入

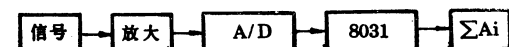
对于模拟量的输入数据，必不可少的要进行数字滤波（软件实现），我们采用的方式是求算



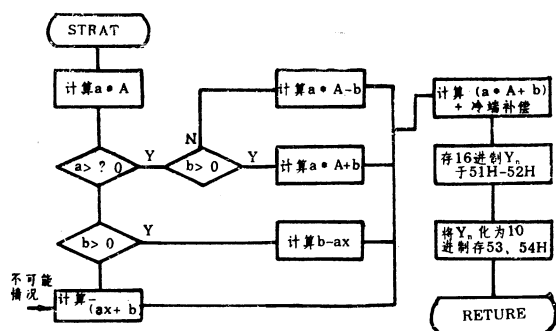
按键	参 数	按键	参 数
0	PID调节周期	A	工程量计算 $y_n = ax + b$ 的a参数
1	标定值R	B	工程量计算 $y_n = ax + b$ 的b参数
2	PID算式中的P参数	C	计算 $y = ax^2 + bx + c$ 留扩充时用
3	PID算式中的I参数	D	冷端补偿
4	PID算式中的D参数	E	升降温速率
5	升温周期	F	键入值结果
6	升温步长		
7	降温周期		
8	降温步长		
9	恒温时间		

术平均值,用均值作为一次输入的输入模拟数字量。

$A = \text{模拟数字量均值} = (\sum Ai) / \text{采样次数}$



$\sum Ai$ 为双字节,A为单字节(8位),由模拟量到工程量转换采用线性公式 $y = a \cdot A + b$  + 冷端补偿。计算框图如下所示:



在升温段以给定升温速率(即一次升几度),按给定的升温步长,增/不增加模拟量的输出值,依据 $R_n - y_n$ 的符号来决定置/清开关量的输出位,对于降温段也是按此法处理。

## 五、恒温段的控制算法

在恒温段采用PID算法和较为流行的开关式控制规律。PID算法是最佳的一种常规控制

算法,依据PID增量算式 $\Delta P_n$ 的符号增/减模出值,再依据 $R - y_n$ 的符号对开关量输出位的置/清。增量算式为 $\Delta P_n = P[(y_{n-1} - y_n) + I(R - y_n) + D(2y_{n-1} - y_n - y_{n-2})]$ 。

根据问题的情况也可选用PID—PD选择算法,大偏差时投入PD算法,因积分分量用于消除稳态误差,在动态过程中大的积分分量却可能使系统的稳定性变坏。改善系统的动态品质,小偏差时再投入PID算法,保证系统的稳态品质。

## 六、输出量:

输出量采用模拟量和开关量同时输出。关于开关量入/出,只用到开关量出值作为控制或报警用。在整个过程中控制用的开关量输出位地址为C口(地址为BO17H)的D<sub>2</sub>位。报警采用开关量输出,由C口(地址为BO17H)的D<sub>1</sub>位(位地址为79H)输出,考虑使用中有全速升/降温的实际要求,故将报警安排正个炉温的升温、恒温、降温、其准则是连续OAH次升/降仍达不到预定的温度时报警。

## 七、主程序框图(略)

## 好消息

本刊将于1990年第一季度出版发行《1989年全国计算机应用研究学术交流会议论文集》和《APPLE II微型机维修实用技术》两种资料。《论文集》主要内容是探讨计算机软件维护及硬件维修在计算机推广应用中的战略地位;交流软件维护和硬件维修的实践经验;展望软件维护和硬件维修的发展前景,同时涉猎软件纠错、加密、解密、系统再开发、计算机通讯、图形文字处理技术、工业控制、办公自动化、新算法、计算机病毒的诊断与解毒、编程技巧等等,内容丰富,见解新颖,方法实用,不乏观点独特的经验之篇。

《APPLE II微型机维修实用技术》系实用性技术资料,在结合讲述APPLE II微机各种背景及各部分工作原理基础上,详尽介绍了该机的电源、主机板、磁盘系统、键盘、显示器、打印机等各个部位的常见故障故障分析及排除方法,可供计算机维修人员参考,亦可作为该种机型的维修培训教材或自学微型机维修技术的实用书其所述方法,同样适用于中华学习机。

需上述资料者,请于90年2月底前向读者服务部唐大利同志联系,以便按需供书;未作预订者不予保证。预订者不必预付订款,只需告知所需资料名称、数量、邮政编码、通讯地址,联系人。

本刊编辑部

# 单纯形加速法与计算机控制系统 的PID最优化仿真设计

甘肃省计算中心 李天健

**摘要:** 本文介绍了一种计算机仿真的方法, 可对计算机控制系统的PID参数进行最优化调整, 从而使系统最优化。文章给出了具体的设计实例、图表及程序框图。读者可直接把这些方法运用于自己的设计之中。

## 一、问题的提出:

在控制系统中, 采用计算机进行控制的系统正在与日俱增。其中采用PID算法的系统占有很大的比重。但PID控制的特点是必须明确了解系统的数学模型, 以便确定PID参数。而许多系统是难以辨识的, 因此往往要经过反复的试验才能获得较满意的参数。有的系统在找到一组PID参数后, 由于系统扰动等因素, 使系统的传递函数发生变化, 从而无法正常调节。因此有必要对PID控制参数进行最优化仿真设计。

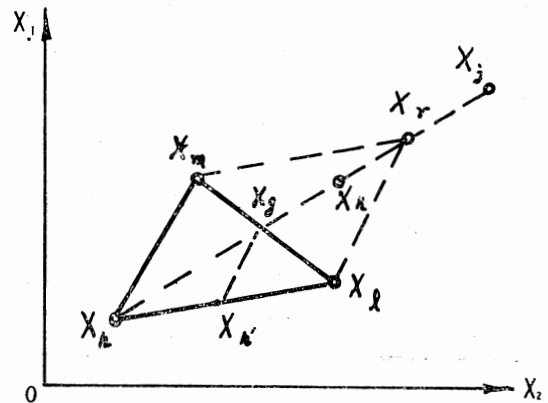
多变量参数的寻优方法有多种, 此处采用单纯形加速法来对PID参数寻优。此法收敛速度较快, 易于在计算机上实现, 在一般的微机上面就可以计算出最优化参数。笔者对此进行了尝试, 结果是令人满意的。

## 二、单纯形加速法寻优的基本方法简介:

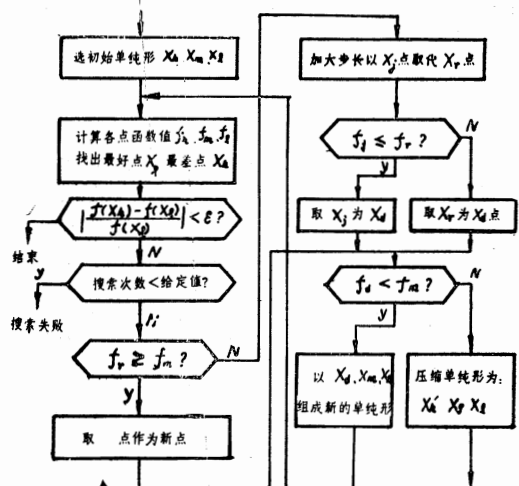
单纯形的定义及基本思想请参考后面列出的参考文献, 限于篇幅, 此处只简略介绍。图1是二维平面上单纯形寻优示意图。

基本形为三角形  $X_h, X_m, X_l$ 。  $X_h$  为最差点,  $X_l$  为最好点, 下一个试探点为  $X_r$ 。其搜索过程如图2。图中  $S=f(x)$  是目标函数。

在  $n$  维空间的单纯形加速法寻优过程与二维时类似。此时三角形变成了  $n$  面体, 其  $n+1$  个顶点组成坐标矩阵  $[X_0, X_1, \dots, X_n]$ , 每个点的坐标为  $x_{ij}$ ,  $i$  表示维,  $j$  表示点, 则此矩阵可表为:



(图 1)



注: 取  $X_{K'}$  点作为新点  $X_d$  (图 2)

$$\begin{pmatrix} X_{10} & X_{11} & \dots & X_{1n} \\ X_{20} & \dots & \dots & \dots \\ \vdots & \vdots & \ddots & \vdots \\ X_{n0} & \dots & \dots & X_{nn} \end{pmatrix} \quad (1)$$

在二维平面中 $\bar{X}_m \bar{X}_l$ 的中点 $X_g$ 在此处变成了除最差点 $X_h$ 外的 $n$ 个点的质量中心,其坐标为:

$$X_{ig} = \frac{x_{ih} + x_{ir}}{2} = \frac{1}{n} \left( \sum_{j=0}^n x_{ij} - x_{ih} \right) \quad (2)$$

( $i=1, 2, 3, \dots, n$ )

其对称点 $x_r$ 的坐标为:

$$x_{ir} = \frac{2}{n} \left( \sum_{j=0}^n x_{ij} - x_{ih} \right) - x_{ih} \quad (3)$$

( $i=1, 2, 3, \dots, n$ )

其寻优是否成功的判断式为:

$$\sum_{i=0}^n [f(X_i) - f(X_l)]^2 < \varepsilon_1 \quad (4)$$

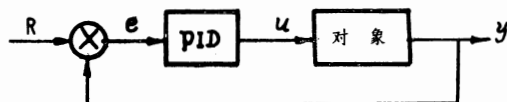
$$\text{或} \left| \frac{f(X_h) - f(X_l)}{f(X_l)} \right| < \varepsilon_2 \quad (5)$$

### 三、将单纯形加速法用于 PID 参数的最优化设计过程

这个过程可以用一个仿真程序来完成,其程序框图如图5。但在编写这个程序之前必须先求出目标函数 $S$ 。

1. 最优化的目标函数;此处采用ITAE准则,即 $S = \int_0^{\infty} t |e(t)| dt$ 。由于无法找到目标函数与PID参数之间直接的数学表达式,因此只能采用系统仿真的办法来解决。

2. 控制系统的框图可以表达为图3。



(图 3)

3. PID调节器数字化后以增量形式给出:

$$\Delta U(n) = kp[e(n) - e(n-1) + \frac{T_s}{T_I} \cdot e(n) + \frac{T_d}{T_s}(e(n) - 2e(n-1) + e(n-2))] \quad (6)$$

4. 控制对象可令其为:

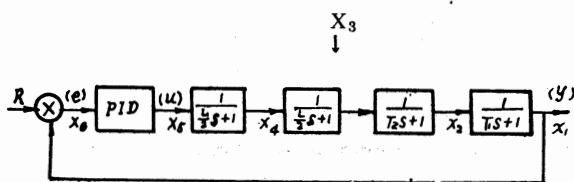
$$\frac{y(s)}{U(s)} = \frac{k \cdot e^{-LS}}{(T_1 S + 1)(T_2 S + 1)} \quad (7)$$

用幂级数可以分解 $e^{-LS}$ 函数为:

$$e^{-LS} = \frac{1}{[1 + (L/2)S]^2} \quad (8)$$

$$\therefore \frac{y(s)}{U(s)} = \frac{1}{(T_1 \cdot S + 1)(T_2 \cdot S + 1)(1 + L/2 \cdot S)^2} \quad (9)$$

原框图(图3)可以重新表达为图4:



(图 4)

5. 离散化表达式的程序化:

(1) PID表达式;式(6)可表达为:

$$x_5 = x_5 + k_1 [x_5 - y_5 + k_2 x_5 + k_3 (x_5 - 2y_5 - z_5)] \quad (10)$$

式中  $x_5 = e(n)$ ,  $y_5 = e(n-1)$ ,

$$z_5 = e(n-2)$$

$$K_1 = K_P, K_2 = T_s/T_I, K_3 = T_D/T_s$$

(2) 一阶延迟网络表达式:

一阶惯性环节的一般拉普拉斯表达式为:

$$\frac{D}{X} = \frac{1}{as+1}$$

用差分反演法可以得到:

$$D(n) = D(n-1) + \left( \frac{T_s}{T_s + a} \right) (x(n) - D(n-1)) \quad (11)$$

由(11)式可以得出后四个传递函数的离散表达式:

$$x_4 = y_4 + k_4 (x_5 - y_5) \quad (12)$$

$$x_3 = y_3 + k_4 (x_4 - y_3) \quad (13)$$

$$x_2 = y_2 + k_5 (x_3 - y_2) \quad (14)$$

$$x_1 = y_1 + k_6 (x_2 - y_1) \quad (15)$$



在以上各式中:  $x_4 = x_4(n)$ ;  $y_4 = x_4(n-1)$  依此

类推.  $K_4 = \frac{T_s}{T_s + L/2}$ ,

$$K_5 = \frac{T_s}{T_s + T_2}; K_6 = \frac{T_s}{T_s + T_1}$$

$$(3) \text{ 目标函数: } S = \int_0^t |e| dt$$

$$S = S + |x_6| \cdot t \cdot \Delta t = S + |x_6| \cdot T_7 \cdot L_1 (16)$$

此处积分采用累加求和方式.  $L_1$  为两次累加之间的时间间隔 (即连续计算的步距),  $T_7$  为计算时的时刻.  $T_s$  为采样周期,  $L_1$  是采样周期的  $1/L_2$ , 所以  $T_s = L_1 \cdot L_2$ .

(4) 计算的顺序:

由  $K_1, K_2, K_3, x_6, y_6, z_6$  求出  $x_5$  (或 (10))  $\rightarrow x_4$  [式 (12)]  $\rightarrow x_3$  [式 (13)]  $\rightarrow x_2$  [式 (14)]  $\rightarrow x_1$  [式 (15)]. 由  $x_6 = R - x_1$  可求出  $x_6$ , 再由式 (16) 可以求出目标函数  $S$

(5) 寻优程序的目标是:

找出PID的三个参数  $K_P, T_I, T_D$  的最佳

值, 使目标函数  $S = \int_0^\infty |x_6| dt$  取极小值.

(此处的积分时限实际上是  $T_7$ , 此时  $e \rightarrow 0$  可以正确地表达  $S$  的原义). 由于上面采用了一系列仿真措施, 使这三个参数与  $S$  之间建立了实用的数学递进关系. 其寻优过程如图 5

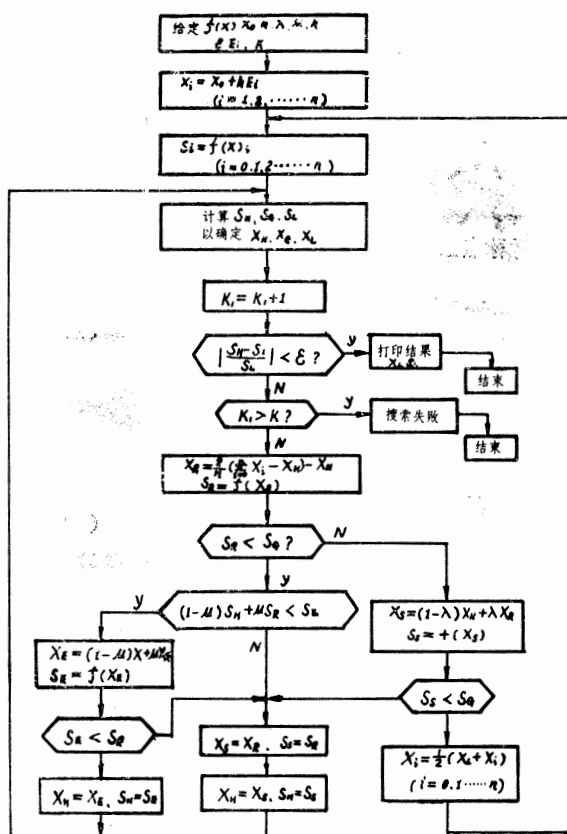
#### 四、差分反演法与状态空间法

以上是采用差分反演法直接将连续系统离散化, 其方法简单, 概念清楚, 但建立的模型误差较大, 有时仿真的情况不够理想, 采用状态空间法推出的模型更好些. 连续系统的状态方程一般可表示为:

$$\begin{aligned} \dot{X} &= A \cdot X + B \cdot U \\ y &= C \cdot X \end{aligned} \quad (17)$$

其中  $\dot{X}, X, B, U, C$  为向量,  $A$  为矩阵.  $U$  为输入量,  $y$  为输出量,  $x$  为中间变量.

方程 (17) 的解可以表述为:



(图 5)

$$X(t) = e^{At} \cdot X(0) + \int_0^t e^{A(t-\tau)} B \cdot U(\tau) d\tau \quad (18)$$

将其离散化后的表达式为:

$$X[(k+1)T] = e^{AT} \cdot X(KT) + \left[ \int_0^T e^{A(t-\tau)} \cdot B \cdot d\tau \right] \cdot U(KT) \quad (19)$$

对一阶惯性环节  $\frac{K_1}{TS+1}$  来说其状态方程为:

$$\dot{x} = -ax + KU \quad (20)$$

式中:  $K = K_1/T$ ,  $a = 1/T$

通过一系列推导后可以得出一个可供计算机仿真的公式:

$$x(n) = M \cdot x(n-1) + N \cdot U(n-1) + P \cdot U(n-2) \quad (21)$$

式中,  $U(n-2) = \frac{U(n) \cdot U(n-1)}{L_1}$

$$M = e^{-\frac{L_1}{T_2}}; N = 1 - M; P = T_2 (A - 1) + L_1$$

用幂级数公式(取前两项)可得:

$$M = e^{-\frac{L_1}{T_2}} \approx 1 - \frac{L_1}{T_2}; N = 1 - A \approx \frac{L_1}{T_2};$$

$$P = T_2 (1 - \frac{L_1}{T_2} - 1) + L_1 \approx 0$$

式(21)变为:

$$x(n) = x(n-1) + \frac{L_1}{T_2} [U(n-1) - x(n-1)] \quad (23)$$

与式(11)比较, 只有第二项的系数不同。(23)

式是  $\frac{L_1}{T_2}$ , 而(11)式是  $\frac{T_s}{T_s + a}$ . 而:

$$\frac{T_s}{T_s + a} = \frac{L_1 \cdot L_2}{L_1 L_2 + a} = \frac{L_1}{L_1 + a/L_2} \approx \frac{L_1}{T_2}$$

此处, 令  $a/L_2 = T_2$ , 而  $L_1 \ll T_2$ , 故上式是成立的。因此(11)式与(23)式是近似相同的。这样就证明了这两种方法的一致性。

此外, 在式(8)中我们采用惯性环节来逼近纯延迟环节  $e^{-LS}$ , 其实计算机可以更准确地仿真纯延迟环节。这就是利用内存位移来实现, 每经过一个采样周期, 将需延迟的量向前移动一个存储单元, 经过  $L/T_s$  个周期后, 即可达到延迟的目的。

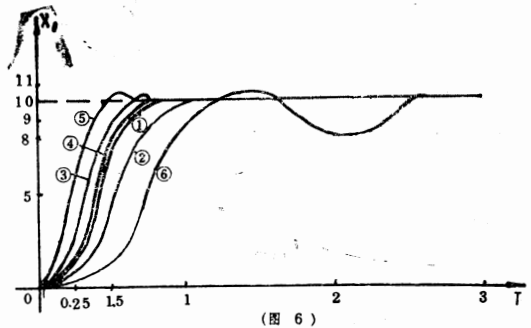
## 五、实例:

笔者对六个完全不同的PID控制系统进行了最优化仿真, 有五个获得了较理想的结果。将这些仿真出来的参数用于实际系统时, 和以下得到的阶跃响应曲线基本吻合, 说明了此方法的可靠性。表一包括了这六个不同系统的惯性参数  $T_1$ ,  $T_2$ , 和纯延迟参数  $T_s$ , 以及最优化仿真后获得的目标函数  $S$  与PID参数  $K_P$ ,  $T_I$ ,  $T_D$  (注意: 表中给出的是标准值, 实际应用时要还原)

这六个被仿真系统的阶跃响应的模拟曲线如图6所示:

表一

系统号	被仿真系统参数			目标函数	最优化PID参数			搜索
	$T_1$	$T_2$	$T_s$	$S$	$K_P$	$K_I$	$K_D$	次数
1	0.44	0.44	0.12	1.189	1.71	0.78	0.23	35
2	0.44	0.44	0.20	2.024	1.22	0.71	0.24	50
3	0.31	0.62	0.07	0.481	2.81	0.80	0.20	69
4	0.45	0.45	0.1	1.201	1.72	0.79	0.24	34
5	0.16	0.8	0.44	0.384	3.35	0.86	0.14	16
6	0.08	0.4	0.52	8.912	1.0	1.195	0.20	43



由图6可以看出: 曲线①, ②, ④具有无超调调节的特点, 曲线③有微量超调, 曲线⑤有5%的超调, 这些系统在一定的要求下都是可用的。曲线⑥有两次超调, 且幅度  $> 20\%$  以上, 调节时间亦相当长。其原因是明显的。由于曲线⑥反应的是一个纯延迟环节系统, 一般的PID调节已不能满足其性能要求。此时应修改算法, 采用Smith算法或Dahlin算法。这说明, 仿真的结果与控制理论的结论是一致的。

## 参考文献:

- [1]: R. J. Bebbiero Meorocessor in Instramen and control, Vol. I, John weley, New York, Inc. 1977.
- [2]: Paul, katz, Digital Control Systems, I, U.S.A Holt Rinehart and wenston, Inc. 1980.
- [3]: J. P. Ignizio著, 闵中求等译, 单目标和多目标系统线性规划, 上海, 同济大学出版社, 1986, 12.
- [4]: 熊光楞, 控制系统的数字仿真, 北京, 清华大学出版社, 1982, 2.
- [5]: 范鸣玉等, 最优化技术基础, 北京, 清华大学出版社, 1982, 12.
- [6]: 张 钨, 微计算机在自动控制中的应用, 北京, 国防工业出版社, 19864.

# 传输控制及网络管理

吉林省计算机技术研究所 张鸿鸣 朱智昕

随着电子计算机及通信技术的发展,数据通信已广泛地被应用于各个领域,如办公自动化、银行支付现金、情报检索及民航订票等系统。

数据通信系统就是把中央电子计算机与分散在各地的许多终端设备或电子计算机用通信线路连接起来的系统。各地产生的数据、问询和其他信息,从终端设备输入,再经过通信线路传送到中央计算机处理。处理后,把结果再由通信线路送回原来的终端设备或其它被指定的终端设备。

为了提高数据通信线路的可靠性,把信息正确地传给对方,除了要对所传送的信息进行差错控制外,还需要各种附加控制,例如需要在信息发送之前,约定发收双方之间的数据通信线路,保持信息发收双方之间的数据通信线路,保持信息发收之间的同步,传送结束后拆除数据通信线路等等的控制。传输控制就是包括差错控制在内的上述控制的总称。进行这些传输控制的一系列规则和步骤,称为传输控制规程。

## 一、差错控制。

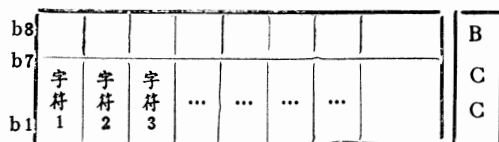
### 1. 奇偶校验。

最简单的控制规程是用于古老的电传打印机 Teletype或Teletype兼容的设备——奇偶校验。在这个方法中,不需要规约就可随时传送数据。计算机不知道所发送的数据是否被接收及正确程度。在每个7比特的代码字符上,都加上一个校验比特,组成单数的1(奇校验)或双数1(偶校验)。如在接收的设备中安设适当的部件,就可以检测每个含有单比特错误字符。这种规程不能检出含有双数比特错误字符。当发现误码后,接收设备也不能通知发送终端。此方法被广泛应用在电传打印机及异步终端机内。

### 2. 双坐标奇偶校验。

双坐标奇偶校验是上面提到的奇偶校验方法的伸延。在这个方法中,每一个数据块的后面会被加上一个块校验字符,如图一所示。每个数据字符会有二个奇偶校验。块校验字符的第一个比特

是对应校验数据块内所有字符的第一个比特,块校验字符第二个比特对应数据块内的第二个比特,以此类推。每个字符本身的奇偶校验字符校验比特称为垂直校验比特。



图一、双坐标奇偶校验

### 3. 循环冗余校验。

随着计算机硬件的发展,循环冗余校验已被广泛应用于差错控制方面。在这一方法中,先把输入数据设想为一个高阶多项式  $P$ , 然后用预定的生成多项式  $G$  的最高次项乘  $P$ , 得到  $P'$ 。将  $G$  除  $P'$  获得商数多项式  $Q$  及余数多项式  $P''$ 。  $P''$  现被作为 CRC, 并将  $P''$  与  $P'$  相加后传输。

在接收端利用与发送端相同的  $G$  去除所接收到的  $P'' + P'$ 。若能被除尽,表示无差错,否则表示有差错。

从理论上所知, CRC 校验可检测所有包含单数比特错误的误码及所有短于校验字符的误码模式。据统计,在一条典型的电话线路上,利用 16 比特 CRC 方法所得误码率比较没有校验系统所得的误码率改进大约 50,000 倍。

## 二、自动请求重发。

当接收端判定传输中有差错发生,便会通知发送端重发信息,直到接收端认为正确接收为止。

自动请求重发工作方式有两种。其中一种是发送端发完一组信息后,等待接收端送回的判定信号,然后再决定是继续发下一组信息,还是重发刚才发过的信息。在另一种系统中,收和发是双工工作,把判定信号插在双方发送的信息中来传送。发送端连续的发送信息,当收到差错信号



时才采取行动。这个系统需要将每个数据块加上号码,差错信号亦需要指示出错的数据块。一般有两个纠错方法,一是把出错的数据块重新发送。另一个是发送端回到出错的数据块,重新发送出错的及所有跟随的数据块。

### 三、规程的标准化。

到目前为止,规程的标准化进展仍较缓慢。差不多每个计算机制造厂家都拥有一套规程。某些企业(例如钢铁公司),分组交换网络及很多其他网络都用不同的规程。因为每个规程各有不同之处,终端用户经常觉得非常不方便。他们的终端机只可以依附同种的计算机。无论终端设备价格如何便宜,用户仍需要向制造商购买。

实际上建立一种国际通用的规程非常困难。但很多国际机构如ISO及CCITT都很努力地在这方面进行工作。

CCITT已通过一系列有关规程的建议,最显著的就是给公共数据网的X.25。它提供了有关建立连接、呼唤,交换数据及通信后拆除线路的一系列步骤。为了更好地利用这些数据网络,所有的设备制造厂家都遵守这个国际上承认的X.25标准。

X.25并不完全包括有关现有的终端设备,但CCITT制定一些建议标准(X.3、X.28及X.29),利用一种称为组装配/拆卸设备把普通的终端连接到分组交换网上。

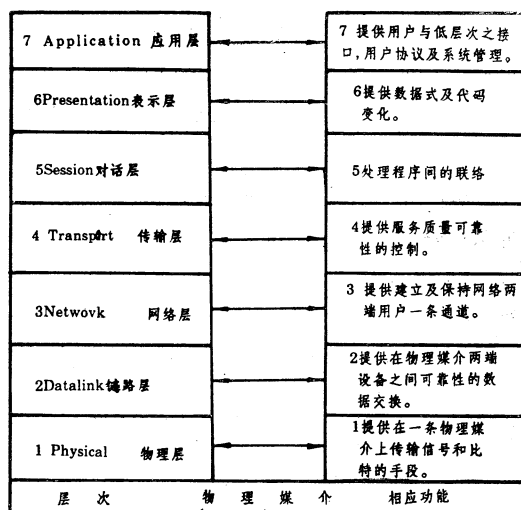
### 四、ISO—层模式。

ISO在尝试制定规程标准化及网络相联建议的时候,提出一个有关标准的模式,这个模式叫ISO开放系统互连。它定义七个规程层次,所以一般称为7—层模式。图二解译了每层的功能。

在开放系统互连模式中,交互工作在相应的层次中进行,即第七层与第七层、第六层与第六层……,以此类推。

把规程层次化有一个好处,只要层次之间接口标准仍被遵守,可以单独修改每一层的内容而不影响整个工作系统。

到现在为止,有关高层次(五、六及七)的进展还很少。X.25标准只是依据第一、二层及三层所制定,现在进行的是传输层,有些规程草



图二. 开放系统互连

案已陆续发表。

### 五、测试仪及规程分析仪。

测试仪可分为模拟测试仪及数字测试仪两种。用于测试传输线路的模拟参数有频率发生器、电平表、可变阻抗衰减器和群时延测试仪等模拟测试仪。用于测试数据传输线路的数字测试仪有误码仪及试验码发生仪等。数据通信用户一般需拥有数字测试仪作为基本数据通信维修手段。

规程分析仪结构比较复杂,它可以模拟、仿真及分析某些计算机的通信规程,也可以从荧光屏上显示传输线路上的数据字符,作为现场分析。它还包括打印及存储设备,把数据存储起来,以便将来再次参考及分析。在大型网络上通常将网络管理系统加上规程分析仪,使能在中央作监察及遥远测试,省去需要在每个节点有专门的网络工程人员。

### 六、网络管理系统。

数据通信网络的监控管理设备是为监督网络使用及确保现行网络设计指标所提供的各种设施组成的。

监控管理设计一般配合硬件及软件来实现。硬件包括线路及系统性能监视器。它们既执行连续的网络监视,也实现了行政管理与会计检查及

# 微机医疗专家系统

吉林省计算机技术研究所 张鸿鸣 朱智昕

当前,许多名老医学专家年事已高,总结、继承与发扬他们的宝贵理论及丰富的临床经验,是一项十分重要又极为迫切的任务。微机医疗专家系统由于它所独有的特点,如能学习、掌握专业知识、理解与运用思维推理规律,可以修改和补充,以及理解限定的自然语言等,就自然地成为完成这一任务的最佳手段之一。

## 一、系统的生成。

系统是基于生成规则,采用了多级分层推理网络的人工智能医疗专家系统。在诊断推理过程中,依据患者的症候(症状)、舌诊、脉象、体征、检验及病理指标等各种有关信息,研制成一个辨证施治医疗专家咨询系统。

微机医疗专家系统的广泛应用,一方面将会普遍提高临床医疗水平,从而也大大造福于人类;另一方面为广大医务工作者向名老医学专家学习创造了良好的机会与条件,这些都具有非常重要的现实意义。

系统着重于医学专家的逻辑推理,辨证论治思维规律的模拟与仿真。它采用了许多近年迅速发展起来的模式匹配,知识表示,生成规则,控制策略及语义网络等人工智能技术与理论。另外系统实现了多对象方式的汉字解释。因为系统是模仿思维的,所以才能对系统的推理过程进行解释,这种解释才会被名老医学专家本人和医务人

员以及患者所理解概念建立及推理过程的仿真。它具有较复杂的复诊处理,解释与教学以及修改与学习等三个功能,并在CRT上用汉字显示,同时输出汉字病历和处方,系统和外界的信息交换全部采用汉字形式,使用汉字对微机医疗专家系统是十分重要的。

## 二、系统的描述。

### 1. I/O功能。

系统的汉字处理分终端CRT显示和打印输出两种形式。能对患者病历(包括病情描述、诊断结果;辨证、立法及方药)作各种形式的解释,程序运行过程中所有人机对话都在CRT上用汉字显示。同时还可由打印机输出打印汉字病历及药方。I/O模块还具有建立病历文件、修改和显示病历文件、建立复诊文件等多种功能。

### 2. 初诊功能。

根据医学专家辨证施治的理论及临床经验,应用人工智能理论和技术,建立了以产生式规则为基础的多层推理网络模型,利用这个模型完成诊断和治疗。网络分为五层,各个层次的划分和构造反映了辨证施治过程的不同阶段。

多层次推理网络的链在程序中采用产生式规则来实现的。在初诊模块里还采用了冗余技术来提高系统的可靠性。由于诊疗软件直接关系到患者的健康与安全,所以采用这种技术有特别重要

分析。软件包与处理机相联系的通用操作系统,或与通信处理机相联系的特定通信软件。监控管理设计的另一方面也提供故障诊断的功能。

对数据通信网路监控管理设计的各项性能的实际定位和控制是网络总体结构的一个重要环节。可以监控的一些局部参数包括:信号质量、误码率、响应时间,以及设备使用情况。当地的操作人员可以确定通信设备的性能,并能在必要时请求适当的维修。管理系统的中央控制中心可

以遥远监察整个通信网络,并能在需要的时候将备用设备和有问题的设备互相转换,提高网络工作效率。

总之,为了得到最高效率,可靠性高及易于诊断出错误的系统,系统设计工程人员需要十分了解所有合用的数据通信技巧及手段。他需要了解如何叫数条数据信道共享一条电话线路及当线路出现故障时,如何叫网络在最短时间恢复工作,因此,研究传输控制及网络管理十分必要。

意义。初诊模块是系统的核心和基础。

### 3. 复诊功能。

在多层推理网络中涉及的诊断, 仅仅是针对一次一次独立的诊断或者初次诊断而言。而实际临床处理中的非初次诊断, 就是复诊, 比初诊还要复杂。复诊不仅要涉及到初诊推理网络中所有内容, 而且要求能把当前病情和以前的病情进行比较, 以确定病情变化。它还必须能对以前诸次的诊断、疗效等信息进行分析和总结, 并结合当前的病情做出判断。为实现上述功能, 在复诊模块中设计了病历存贮与检索、诊断分析与综合等子块。

复诊时, 系统首先要确定病情是好转、不变还是加重三种不同情况, 然后针对不同情况再分别处理。

### 4. 多层次推理与复诊处理功能。

对于医学理论与临床经验的分析以及对具体患者诊疗过程复杂性的了解, 系统的诊断推理采用了多层次, 并具有复诊处理的较复杂功能。

当考虑第K次诊断时, 除了收集患者当前的症状外, 还必须把握第一次到第K-1次诊断及其治疗效果的全面情况, 因此系统必须具有存贮、检索、修改及综合分析病历的能力, 判断一次诊疗效果的能力。

### 5. 系统的解释与教学功能。

对患者进行诊断的过程中, 系统构造了一个推理网络。当医务人员或患者对系统作出的诊断提出问题时, 系统能够展示其推理过程。回溯诊断过程中所使用的诸生成规则。提问可以采用不同的方式, 系统能按照提问者的要求作步进式地回推或前推网络上的一个节点, 也可展示全过程, 这些功能一方面使系统具有教学功能, 另一方面也为检查系统提供了一种有效的手段。

### 6. 系统的修改与学习功能。

系统一方面能自己发现与存贮系统自身与医学专家的诊断不相符合的生成规律, 从而便于医学专家对这样的生成规则进行修改, 为医务人员学习医学专家经验提供了方便。由于系统进行人机对话, 使系统的一些较复杂的智能行为能够实现。

复诊模块能自动发现一些问题, 这是学习的

一个起点。学习功能就是要发现问题并按照医学专家意见自动修改系统存在的问题, 丰富系统的内容, 不断获得新知识。学习模块除本身的功能外, 主要是控制、调用有关子模块, 并与复诊模块、解释模块有机结合, 共同完成系统的学习功能。

除上述功能外, 系统还具有下述一些功能及特点。

系统在诊断过程中, 进行完全有针对性的人机对话, 是通过系统所采用的选择提问控制总调度器来实现的。也就是系统按照其自身推理过程中的实际需要, 完全有目的地提出问题, 从而提高了提问效率。

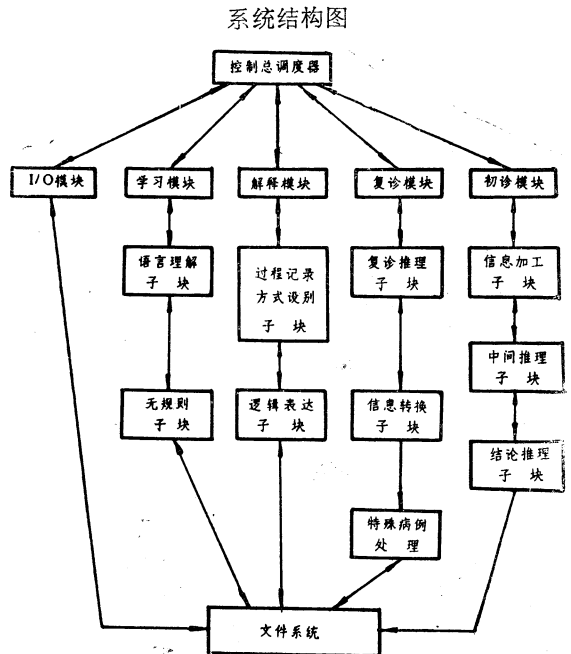
当药房缺少某药方中的某味中药时, 系统能够用性质与功效相近的中药去代替。当药房缺少某成药时, 系统能用功效相近的方药代换之。

对于一些患者, 系统能对病程作出一些预测。

总之, 系统是一个采用了较多人工智能技术的功能较强的微机医疗专家系统。

## 三、系统的结构。

系统采用了按功能划分为若干个大的模块, 每个大模块中又分成若干个子模块, 各个模块之间相对独立, 使得程序结构清晰, 十分便于阅读和维护, 具体如图中所表示一样。



## 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

在我们开发软件的过程中,往往需要在程序的开始给出诸如开发单位、主要开发人员以及版本号等信息,就象电视或电影片头的片名一样,我们希望能做到美观醒目。但目前我国流行的汉字操作系统绝大多数都是以 $16 \times 16$ 点阵的简易字库来显示的,汉字的点阵精度很低,其颜色和大小又不能变化,难以达到我们的要求。于是有人就利用在图形方式下(在CCDOS工作方式下),直接向视频终端存储区写入图形信息数据的方法来实现。但这需要预先在纸上把所需要的高精度汉字及图形用 $320 \times 200$ (中分辨率)即6万多个点“画”出来,再转换成16000个16进制数字,然后写入从B8000H开始的4000H个存储单元中,工作量相当大,修改起来很不方便,而且效果难以达到最佳。

近年来的CCDOS汉字系统在打印输出方面又有了不少改进,比如吴晓军研制的CCDOS2.13(A),能够用四种字体、多种字号打出比较美观的汉字和字符。利用该系统带有的四种字体的高质量24点阵字库,本文给出一个构造含高精度、多字体(宋、仿、黑、楷)、任意大小汉字的屏幕画面的程序,用起来能达到比较满意的效果。希望它能使您的应用软件锦上添花。

为使各种层次的读者均能很容易地读懂,本

程序用BASIC语言写成,程序的输出为一个二进制制图形文件以提供给您的应用程序,该图形文件既可为您的BASIC语言所调用,也可为宏汇编MASM或别的高级语言如Pascal、C语言所调用。

只要您按提示输入,顷刻之间,一幅绚丽多彩的画面就出现在您的显示器上。如您还有不满意处,可逐点修改,尤其是可以删除或画上您任意指定的任意条边的多边形内部所有的点,操作简便、轻松,还可以读入您用本程序或其它软件构造的画面,进行修改,然后存入磁盘,再用您自己的程序给它加上诸如边框、辐射线等修饰(如果您愿意的话),以备您下一步的软件开发使用。如您想要一份硬拷贝,则有两种方式可以选择:按PrtSc,或按P,后者是由程序用汇编语言调用系统中断实现的,二者输出结果是不同的。

**硬件环境:** IBM-PC/XT/AT及其兼容机,内存 $\geq 640$ K,CGA或EGA卡及显示器。需要特别指出的是:由于长城0520对视频控制方式作了改动,所以在该机上无法运行本程序。

**软件环境:** CCDOS2.00及以上版本, BASICA Ver A2.10及以上版本,硬盘上配有 $24 \times 24$ 点阵打印字库CLIB24(1~87区)或HZK

### 四、系统的效果。

根据医学专家诊治的400个患者的病例验证,辨证施治完全符合率为97.5%,基本符合率为0.75%,不符合率为1.75%(其中,完全符合指诊断、辨证、立法及基本方药与医学专家完全一致;基本符合指诊断和立法相同,辨证有程度上的不同,不符合指辨证和医学专家不符合)。

系统基本反映了医学专家的学术思想和临床经验,完正地总结和继承了医学专家的诊治理论和经验,为后人留下了宝贵的医学财富。这个系统投入临床使用,使更多的患者享受到名医水平

验证站果 附表A

验证方式	数量	完全符合	基本符合	不符合
回顾验证	130	127	0	3
双盲验证	70	69	1	0
临床验证	200	194	2	4
合计	400	390	3	7
百分率(%)	100	97.5	0.75	1.75

的治疗,减轻患者的痛苦,提高人民的健康水平。

24T (1≈15区)及HZK24S、HZK24K、HZK24H、HZK24K (16≈87区)之一。当然,若后五种字库都有则可实现字体的多样化。如果您的24点阵字库(非压缩存储)名称与本文不同,只要修改一下程序开头有关的几句即可,不过应注意字库的起始区号与字库逻辑记录之间的对应关系,并且字库必须为纵排字模(一般的24点阵打印字库均为纵排)。另外,字库文件的属性必须是一般属性(Normal),否则,您可以用DEBUG或PC-TOOLS对其属性进行修改。

为提高程序的运行速度,建议读者使用编译BASIC(Microsoft-BASIC(A) Compiler Ver 1.0或以上版本)。假设在C盘中有下列文件:

——BASCOM.COM BASIC编译器

——BASCOM.LIB BASIC程序库

——LINK.EXE LINKER连接器

则其编译连接过程为:

C>BASCOM MAKER; /O <Enter> (不带运行包)

C>LINK MAKER; <Enter>

下面介绍一下程序的使用方法,所有功能均为单键操作:

#### 光标控制:

1. 字母键E、X、S、D 分别控制光标点上、下、左、右移动。这里充分利用了中分辨率时的四种颜色,使您控制的光标点与周围的前景或背景颜色都不相同,同时在它本身为前景与背景时也不相同。这就使您既不会找不到光标点又不至于搞不清光标处的属性(是背景还是前景)。

2. 数字键1≈9用来控制上述EXSD四个命令每次移动的点数。用在EXSD命令之前。省缺值为1。

#### 修改操作:

1. 字母键T写点;Q擦点;G反相(既光标处的属性前景变背景,背景变前景)。

2. 字母键R写块 如您要在某一个五角形区域内全部写上点(不管这里原来有没有点),则把光标移到第一个顶角上按R,移到第二个角

上再按R(这时,两点之间会用一条与周围不同颜色的线连起来),……移到第五个角上按R,这时五边形还没有封闭,再按一下R则封闭起来,然后光标移到五边形的内部任一点,按R后则此多边形及内部各点就充满了前景的颜色。

3. 字母键W擦块与R操作相仿。

#### 其它:

1. 存盘 按K,屏幕上方提示:“文件名”,回答后回车。

2. 读盘 按A,屏幕上方提示:“文件名:”,回答后回车。注意:A操作会把原来屏幕上的内容全部冲掉。对A和K操作提示时直接回车可取消该操作。

3. 拷屏 按P键即可。注意:这时与按Prt-Sc不同:后者结果为颜色号02的图形(简易拷贝);前者结果为颜色03的图形,颜色要深得多(精密拷贝),但代价是所化的时间也要更长。

下面介绍一下程序结构和各功能块所在位置以备读者二次开发时参考:

1000≈1170程序初始化

1180≈1290从字库中取打印字模

1300≈1390为中分辨率显示横扩字模

1400≈1560转化为显示字模并得出初始画面

1570≈1740接受键盘命令

1750≈1910子程序:横扩字模时调用

1920≈1990子程序:E上移光标点

2000≈2070子程序:X下移光标点

2080≈2150子程序:S左移光标点

2160≈2230子程序:D右移光标点

2240≈2250子程序:T写点

2260≈2270子程序:Q擦点

2280≈2330子程序:G反相

2340≈2460子程序:K存盘

2470≈2520子程序:A读盘

2530≈2630子程序:P拷屏

2640≈2830子程序:R写块

2840≈2030子程序:W擦块

3040≈3100子程序:写块子程序调用

3110≈3170子程序:擦块子程序调用

3180≈3250子程序:拷屏子程序调用

3260≈3330子程序:拷屏子程序调用

本程序约定的字数为3个,每字点阵为96×96(横纵均放大四倍),您可以任意选定字数和横纵放大倍数,这时您只要修改第1010句的IP



数组的维数, 1020行的IZZS (总字数)以及1480行LINE, 1490行GET和1520行PUT的相应参数 (程序列表中带下划线表示) 即可。其中, IP数组的容量 (对于整型数组为维数 $\times 2$ )应满足 $\geq 4 + \text{INT}(X * 2 + 7) / 8 * Y$ , 式中: X、Y分别为横纵放大倍数。

最后说明一点: 用本程序构造的精密汉字、图形不只是可以在中分辨率方式下使用, 在高分辨率时除了颜色以外和中分时完全一样。

程序内容列表如下:

```

1000 REM 高精度多字体屏幕画面构造软件 MAKER.BAS
1010 DEFINT I-N: DIM Z$(72), X$(72), Y$(144), W$(72),
H$(24, 24), D$(72), DD$(3, 24), EE$(3, 24), IP(9),
KX(10), KY(10): DEF SEG
1020 SCREEN 1: COLOR 9, 0: KEY OFF: TIMES = 0: IZZS = 3
: IZS = 1
1030 LINE(219, 15) - (307, 36), 2, BF: LOCATE 2, 30
: PRINT "第"; IZS; "个字"; LOCATE 2, 30
1040 LINE(61, 15) - (307, 36), 2, B
1050 LOCATE 2, 14: PRINT "(T/S/W/H/K/C)";
1060 LOCATE 2, 9: PRINT "字库"; ZK$ = INPUT$(1)
1070 IF ZK$ = "T" OR ZK$ = "t" THEN ZK$ = "T"
1080 IF ZK$ = "S" OR ZK$ = "s" THEN ZK$ = "S"
1090 IF ZK$ = "F" OR ZK$ = "f" THEN ZK$ = "F"
1100 IF ZK$ = "H" OR ZK$ = "h" THEN ZK$ = "H"
1110 IF ZK$ = "K" OR ZK$ = "k" THEN ZK$ = "K"
1120 IF ZK$ = "C" OR ZK$ = "c" THEN ZK$ = "C"
1130 IF ZK$ = "T" OR ZK$ = "S" OR ZK$ = "F"
OR ZK$ = "H" OR ZK$ = "K" OR ZK$ = "C"
THEN 1150
1140 BEEP: GOTO 1060
1150 ZK$ = "C:HZK24" + ZK$
1160 IF ZK$ = "C:HZK24C" THEN ZK$ = "C:CLIB24"
1170 PRINT " "; ZK$;
1180 LOCATE 2, 9: PRINT "区码"; QU = VAL(INPUT$(2))
: IF QU = 0 THEN 1030
1190 LOCATE 2, 9: PRINT "位码"; WEI = VAL(INPUT$(2))
: IF WEI = 0 THEN 1030
1200 OPEN ZK$ AS #1 LEN = 72
1210 FOR I = 1 TO 72: FIELD #1, (I - 1) AS ZZ$, 1 AS Z$(I)
: NEXT I
1220 IF ZK$ = "C:HZK24T" OR ZK$ = "C:CLIB24" THEN
QW = 15 * 94 ELSE QW = 0
1230 QW = QW + (QU - 16) * 94 + WEI
1240 GET #1, QW: KEY ON: KEY OFF: TIMES = TIMES + 1
1250 FAAA = VARPTR(#1): DAAA = FAAA + 188
1260 FOR I = 1 TO 72
1270 X$(I) = HEX$(PEEK(DAAA + I - 1)): IF LEN(X$(I)) = 1
THEN X$(I) = "0" + X$(I)
1280 NEXT I
1285 CLOSE #1
1290 REM 以上取字模
1300 FOR I = 1 TO 72
1310 XL$ = MID$(X$(I), 1, 1): XR$ = MID$(X$(I), 2, 1)
1320 A$ = XL$
1330 GOSUB 1750
1340 Y$(2 * I - 1) = B$
1350 A$ = XR$
1360 GOSUB 1750
1370 Y$(2 * I) = B$
1380 NEXT I
1390 REM 已上横扩字模
1400 DEF SEG = &HB800
1410 FOR I = 1 TO 24
1420 FOR J = 1 TO 6
1430 ZH$ = Y$(I - 1) * 6 + J
1440 IJ = (I - 1) * 80 + J - 1: JI = IJ + &H2000: PH = VAL("&H" + ZH$)
1450 POKE IJ, PH: POKE JI, PH
1460 NEXT J, I
1470 DEF SEG: CLOSE
1480 LINE (24, 0) - (27, 47), 2, BF: KCZ = 0
1490 GET (24, 0) - (27, 31), IP: KPCC = 0
1500 FOR J = 0 TO 23
1510 FOR I = 0 TO 23
1520 IF POINT(I, 2 * J) < 0 THEN PUT(16 + (IZS - 1) * 24 * 4
+ J * 4, 52 + 1 * 4), IP, PSET
1530 NEXT I, J
1540 IZS = IZS + 1
1550 IF IZS > IZZS THEN 1570
1560 GOTO 1030
1570 IX1 = 16: IX2 = 303: IY1 = 52: IY2 = 147: IX = IX1: IY = IY1
: IIP = POINT(IX, IY): PSET(IX, IY), IIP + 1
1580 JJ = 1: LOCATE 2, 9: PRINT "OK!"; FOR I = 1 TO 1000
: KY$ = INKEY$: IF KY$ < ">" THEN 1590: NEXT
1590 LINE(0, 0) - (319, 47), 0, BF: KY$ = INPUT$(1)
: IF KY$ = CHR$(27) THEN END
1600 LOCATE 1, 1: PRINT KY$: IF VAL(KY$) = 0 THEN 1620
1610 JJ = VAL(KY$): GOTO 1590: JJ为控制光标移动的点数
1620 IF KY$ = "E" OR KY$ = "e" THEN GOSUB 1920: GOTO 1690
1630 IF KY$ = "X" OR KY$ = "x" THEN GOSUB 2000: GOTO 1590
1640 IF KY$ = "S" OR KY$ = "s" THEN GOSUB 2080: GOTO 1590
1650 IF KY$ = "D" OR KY$ = "d" THEN GOSUB 2160: GOTO 1590
1660 IF KY$ = "T" OR KY$ = "t" THEN GOSUB 2240: GOTO 1590
1670 IF KY$ = "Q" OR KY$ = "q" THEN GOSUB 2260: GOTO 1590
1680 IF KY$ = "G" OR KY$ = "g" THEN GOSUB 2280: GOTO 1590
1690 IF KY$ = "K" OR KY$ = "k" THEN GOSUB 2340: GOTO 1590
1700 IF KY$ = "A" OR KY$ = "a" THEN GOSUB 2470: GOTO 1590
1710 IF KY$ = "P" OR KY$ = "p" THEN GOSUB 2530: GOTO 1590
1720 IF KY$ = "R" OR KY$ = "r" THEN GOSUB 2640: GOTO 1590
1730 IF KY$ = "W" OR KY$ = "w" THEN GOSUB 2840: GOTO 1590
1740 IF KY$ = CHR$(27) THEN END: ELSE GOTO 1590
1750 IF A$ = "0" OR A$ = " " THEN B$ = "00"
1760 IF A$ = "1" THEN B$ = "03"
1770 IF A$ = "2" THEN B$ = "0C"
1780 IF A$ = "3" THEN B$ = "0P"
1790 IF A$ = "4" THEN B$ = "30"
1800 IF A$ = "5" THEN B$ = "33"
1810 IF A$ = "6" THEN B$ = "3C"
1820 IF A$ = "7" THEN B$ = "3F"
1830 IF A$ = "8" THEN B$ = "CQ"
1840 IF A$ = "9" THEN B$ = "C3"
1850 IF A$ = "A" OR A$ = "a" THEN B$ = "CC"
1860 IF A$ = "B" OR A$ = "b" THEN B$ = "CF"
1870 IF A$ = "C" OR A$ = "c" THEN B$ = "FO"
1880 IF A$ = "D" OR A$ = "d" THEN B$ = "F3"
1890 IF A$ = "E" OR A$ = "e" THEN B$ = "FC"
1900 IF A$ = "F" OR A$ = "f" THEN B$ = "FF"
1910 RETURN
1920 IF KCZ < -1 THEN 1960
1930 IY = IY - JJ: IF IY < IY1 THEN IY = IY2
1940 IIP = POINT(IX, IY): PSET(IX, IY), IIP
1950 GOTO 1990
1960 IIP = POINT(IX, IY): PSET(IX, IY), IIP - 1
1970 IY = IY - JJ: IF IY < IY1 THEN IY = IY2
1980 IIP = POINT(IX, IY): PSET(IX, IY), IIP + 1
1990 RETURN
2000 IF KCZ < -1 THEN 2040
2010 IY = IY + JJ: IF IY > IY2 THEN IY = IY1
2020 IIP = POINT(IX, IY): PSET(IX, IY), IIP
2030 GOTO 2070
2040 IIP = POINT(IX, IY): PSET(IX, IY), IIP - 1
2050 IY = IY + JJ: IF IY > IY2 THEN IY = IY1
2060 IIP = POINT(IX, IY): PSET(IX, IY), IIP + 1
2070 RETURN
2080 IF KCZ < -1 THEN 2120
2090 IX = IX - JJ: IF IX < IX1 THEN IX = IX2
2100 IIP = POINT(IX, IY): PSET(IX, IY), IIP
2110 GOTO 2150
2120 IIP = POINT(IX, IY): PSET(IX, IY), IIP - 1
2130 IX = IX - JJ: IF IX < IX1 THEN IX = IX2
2140 IIP = POINT(IX, IY): PSET(IX, IY), IIP + 1
2150 RETURN
2160 IF KCZ < -1 THEN 2200
2170 IX = IX + JJ: IF IX > IX2 THEN IX = IX1
2180 IIP = POINT(IX, IY): PSET(IX, IY), IIP
2190 GOTO 2230
2200 IIP = POINT(IX, IY): PSET(IX, IY), IIP - 1
2210 IX = IX + JJ: IF IX > IX2 THEN IX = IX1
2220 IIP = POINT(IX, IY): PSET(IX, IY), IIP + 1
2230 RETURN
2240 PSET(IX, IY), 3
2250 RETURN
2260 PSET(IX, IY), 1
2270 RETURN
2280 IIP = POINT(IX, IY)
2290 IF IIP = 1 OR IIP = 0 THEN IIP = 3
2300 GOTO 2320
2310 IF IIP = 3 OR IIP = 2 THEN IIP = 1
2320 PSET(IX, IY), IIP
2330 RETURN
2340 DEF SEG = &HB800
2350 LOCATE 1, 1: PRINT "文件名: "; FILE$ = ""
2360 KY$ = INPUT$(1)
2370 IF KY$ = CHR$(13) THEN 2410
2380 PRINT KY$;
2390 FILE$ = FILE$ + KY$
2400 GOTO 2360
2410 LINE(0, 0) - (127, 47), 0, BF
2420 IF FILE$ = "" THEN 2460

```

```

2430 IIP=POINT(IX,IY):PSET(IX,IY),IIP-1
2440 BSAVE FILE$,0,&H4000
2450 DEF SEG
2460 RETURN
2470 DEF SEG=&HB800
2480 LOCATE 1,1:PRINT "文件名: ";INPUT " ".FILE$
2490 IF FILE$="" THEN 2520
2500 BLOAD FILE$,0
2510 DEF SEG
2520 RETURN
2530 KPCS=KPCS+1:DEF SEG=&H7000:OFFSETSEG=0
2540 IF KPCS<>1 THEN 2580
2550 POKE 0,&HCD:POKE 1,&H5:POKE 2,&HCB
2560 LOCATE 1,1:PRINT " ":PSET(0,0),C
2570 GOSUB 3180
2580 CALL ABSOLUTE(OFFSETSEG)
2590 GOSUB 3260
2600 IIP=POINT(IX,IY):PSET(IX,IY),IIP+1
2610 DEF SEG
2620 IIP=POINT(IX,IY):IF IIP=0 THEN IIP=1
2630 RETURN
2640 IF KCZ=-1 THEN 2790
2650 KCZ=KCZ+1:KY(KCZ)=IX:KY(KCZ)=IY
2660 IF KCZ>1 THEN 2700
2670 PSET(KX(KCZ),KY(KCZ)),3
2680 IX1M=IX:IX2M=IX:IY1M=IY:IY2M=IY
2690 GOTO 2830
2700 IF KX(KCZ)=KX(KCZ-1) AND KY(KCZ)=KY(KCZ-1)
    THEN 2770
2710 IF IX>IX2M THEN IX2M=IX
2720 IF IX<IX1M THEN IX1M=IX
2730 IF IY<IY1M THEN IY1M=IY
2740 IF IY>IY2M THEN IY2M=IY
2750 LINE(KX(KCZ-1),KY(KCZ-1))-(KX(KCZ),KY(KCZ)),3
2760 GOTO 2830
2770 LINE(KX(KCZ),KY(KCZ))-(KX(1),KY(1)),3
2780 KCZ=-1:RETURN
2790 PAINT(IX,IY),3,3
2800 GOSUB 3040
2810 IIP=POINT(IX,IY):PSET(IX,IY),IIP+1
2820 KCZ=0
2830 RETURN
2840 IF KCZ=-1 THEN 2990
2850 KCZ=KCZ+1:KX(KCZ)=IX:KY(KCZ)=IY
2860 IF KCZ>1 THEN 2900
2870 PSET(KX(KCZ),KY(KCZ)),1
2880 IX1M=IX:IX2M=IX:IY1M=IY:IY2M=IY
2890 GOTO 3030
2900 IF KX(KCZ)=KX(KCZ-1) AND KY(KCZ)=KY(KCZ-1)
    THEN 2970
2910 IF IX>IX2M THEN IX2M=IX
2920 IF IX<IX1M THEN IX1M=IX
2930 IF IY<IY1M THEN IY1M=IY
2940 IF IY>IY2M THEN IY2M=IY
2950 LINE(KX(KCZ-1),KY(KCZ-1))-(KX(KCZ),KY(KCZ)),1
2960 GOTO 3030
2970 LINE(KX(KCZ),KY(KCZ))-(KX(1),KY(1)),1
2980 KCZ=-1:RETURN
2990 PAINT(IX,IY),1,1
3000 GOSUB 3110
3010 IIP=POINT(IX,IY):PSET(IX,IY),IIP+1
3020 KCZ=0
3030 RETURN
3040 FOR I=IY2M TO IY1M STEP -1
3050 FOR J=IX2M TO IX1M STEP -1
3060 IIP=POINT(J,I)
3070 IF IIP=3 THEN IIP=2
3080 PSET(J,I),IIP
3090 NEXT J,I
3100 RETURN
3110 FOR I=IY2M TO IY1M STEP -1
3120 FOR J=IX2M TO IX1M STEP -1
3130 IIP=POINT(J,I)
3140 IF IIP=1 THEN IIP=0
3150 PSET(J,I),IIP
3160 NEXT J,I
3170 RETURN
3180 IIP=POINT(IX,IY):PSET(IX,IY),IIP+1
3190 FOR I=IY2 TO IY1 STEP -1
3200 FOR J=IX2 TO IX1 STEP -1
3210 IIP=POINT(J,I)
3220 IF IIP=2 THEN IIP=3 ELSE 3240
3230 PSET(J,I),IIP
3240 NEXT J,I
3250 RETURN
3260 IIP=POINT(IX,IY):PSET(IX,IY),IIP+1
3270 FOR I=IY2 TO IY1 STEP -1
3280 FOR J=IX2 TO IX1 STEP -1
3290 IIP=POINT(J,I)
3300 IF IIP=3 THEN IIP=2 ELSE 3320
3310 PSET(J,I),IIP
3320 NEXT J,I
3330 RETURN

```

## 一个简易画空、实心框图程序

重庆制药七厂计算机室

龙兵生

### 一、引言

为满足一些图形功能较弱的语言(例如DBASE-III)在开发编制应用程序时画一些空、实心框图,增强人机对话的友好界面,本人根据计算机处理图形点阵信息的有关原理,使用8088汇编语言编制了一个名为HK.COM的简易画框图程序。HK.COM可直接在DOS下运行,也可作为外部命令由其它程序调用。例如DBASE-III可使用RUN命令来调用HK.COM

### 二、程序生成

考虑到速度方面的要求,为使调用程序能够

快速调用画框图程序进行画框图作业,以COM文件的形式生成画框图程序可实现快速调用需求。

首先使用DEBUG调试程序的A命令将程序一输入内存

—A100	回车	输程序一
—NHK.COM	回车	
—RCX	回车	
CX 0000		
: 01CA	回车	
—W	回车	
—Q	回车	

```

-U100 1ca
SE3D:0100 B88200 MOV BX,0082
SE3D:0103 8A07 MOV AL,IBX1
SE3D:0105 2C30 SUB AL,30
SE3D:0107 B164 MOV CL,64
SE3D:0109 F6E1 MUL CL
SE3D:010B 89C2 MOV DX,AX
SE3D:010D B400 MOV AH,00
SE3D:010F 43 INC BX
SE3D:0110 8A07 MOV AL,IBX1
SE3D:0112 2C30 SUB AL,30
SE3D:0114 B10A MOV CL,0A
SE3D:0116 F6E1 MUL CL
SE3D:0118 01C2 ADD DX,AX
SE3D:011A 43 INC BX
SE3D:011B 8A07 MOV AL,IBX1
SE3D:011D 2C30 SUB AL,30
SE3D:011F 01C2 ADD DX,AX
SE3D:0121 89D0 MOV AX,DX
SE3D:0123 89DA MOV DX,BX
SE3D:0125 83EB02 SUB BX,+02
SE3D:0128 8907 MOV IBX1,AX
SE3D:012A 89D3 MOV BX,DX
SE3D:012C 43 INC BX
SE3D:012D 81FB8E00 CMP BX,008E
SE3D:0131 7CD0 JL 0103
SE3D:0133 A08E00 MOV AL,1008E1
SE3D:0136 2C30 SUB AL,30
SE3D:0138 3C03 CMP AL,03
SE3D:013A 7F5D JG 0199
SE3D:013C A28E00 MOV 1008E1,AL
SE3D:013F 8B1E8800 MOV BX,100881
SE3D:0143 8B168500 MOV DX,100851
SE3D:0147 8B0E8200 MOV CX,100821
SE3D:014B A08E00 MOV AL,1008E1
SE3D:014E B40C MOV AH,0C
SE3D:0150 CD10 INT 10
SE3D:0152 41 INC CX
SE3D:0153 39D9 CMP CX,BX
SE3D:0155 7CF4 JL 014B
SE3D:0157 8B168B00 MOV DX,1008B1
SE3D:015B 8B0E8200 MOV CX,100821
SE3D:015F A08E00 MOV AL,1008E1
SE3D:0162 B40C MOV AH,0C
SE3D:0164 CD10 INT 10
SE3D:0166 41 INC CX
SE3D:0167 39D9 CMP CX,BX
SE3D:0169 7CF4 JL 015F
SE3D:016B 8B1E8B00 MOV BX,1008B1
SE3D:016F 8B0E8200 MOV CX,100821
SE3D:0173 8B168500 MOV DX,100851
SE3D:0177 A08E00 MOV AL,1008E1
SE3D:017A B40C MOV AH,0C
SE3D:017C CD10 INT 10
SE3D:017E 42 INC DX
SE3D:017F 39DA CMP DX,BX
SE3D:0181 7CF4 JL 0177
SE3D:0183 8B168500 MOV DX,100851
SE3D:0187 8B0E8800 MOV CX,100881
SE3D:018B A08E00 MOV AL,1008E1
SE3D:018E B40C MOV AH,0C
SE3D:0190 CD10 INT 10
SE3D:0192 42 INC DX
SE3D:0193 39DA CMP DX,BX
SE3D:0195 7CF4 JL 018B
SE3D:0197 EB30 JMP 01C9
SE3D:0199 2C04 SUB AL,04
SE3D:019B B88400 MOV BX,0084
SE3D:019E 8807 MOV IBX1,AL
SE3D:01A0 A18500 MOV AX,100851
SE3D:01A3 B88D00 MOV BX,008D
SE3D:01A6 8907 MOV IBX1,AX
SE3D:01A8 8B168B00 MOV DX,1008B1
SE3D:01AC 8B0E8200 MOV CX,100821
SE3D:01B0 8B1E8800 MOV BX,100881
SE3D:01B4 A08400 MOV AL,100841
SE3D:01B7 B40C MOV AH,0C
SE3D:01B9 CD10 INT 10
SE3D:01BB 41 INC CX
SE3D:01BC 39D9 CMP CX,BX
SE3D:01BE 7CF4 JL 01B4
SE3D:01C0 4A DEC DX
SE3D:01C1 8B1E8D00 MOV BX,1008D1
SE3D:01C5 39DA CMP DX,BX
SE3D:01C7 7FE3 JG 01AC
SE3D:01C9 CD20 INT 20

```

### 三、使用方法

使用时键入程序名HK后空一格输入13个参数回车即可。

参数与框图结构的对应关系如下：

前6个参数表示所画框图的左上角座标，前3位是横座标，后3位是纵座标。

u后6个参数表示所画框图的右下角座标，各参数表示意义同上。

第13个参数表示所画框图的颜色和空、实心选择。

参数取值范围

A 中分状态 (320X200) 下，横座标为0—319，纵座标为0—199，颜色值取0、1、2、3表示以4种不同色彩画空心框图，取4、5、6、7、表示以4种不同色彩画实心框图。

B 高分状态 (640X200) 下，横座标为0—639，纵座标为0—199，颜色值取0、1表示以2种不同色彩画空心框图，取4、5表示以2种不同色彩画实心框图。

### 四、应用举例

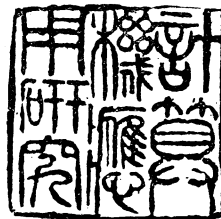
1 在中分状态 (320X200) 下以前景色1在屏幕 (20, 10) 处画一个20X100大小实心框图。

A>HK 0200100391095 回车

2 在高分状态 (640X200) 下以前景色1在屏幕 (120, 109) 处画一个206X60大小空心框图。

A>HK 20103251681 回车

注意：若横座标或纵座标值不足3位时以0补足3位。例一框图左上角横座标为20和10，使用HK.COM时应写成020010形式。



计算机应用研究

# 能够在微型机上快速实现CRC校验的一种方法

四川省地震局计算中心

袁兴志 辛 华

计算机在通讯网上通信时,往往由于通讯系统电气设备的质量问题、气候、磁场等因素的影响,经常存在着使被传送的信息发生失真的可能性。为了确保传送信息的正确性,一般都要对所传送的信息进行检错处理。检错处理的方法,目前,在国内外较常用的是Cyclic Redundancy Check,即循环冗余码检验,简称为CRC校验。

可是,CRC校验,不论一次要传送的信息(bit)量是多少,都有相当多的计算步骤,我们要计算机在很短的时间里完成必要的计算步骤,实现ARQ方式的通讯,这样的工作由PC XT/AT机或较高档一点的计算机来完成是比较容易实现的事[1]。若是按照这种思想设计一个通讯网,如果网上结点(收、发点)太多,每个结点配置一台(PC XT/AT或兼容机)计算机,其经费投资很大;并且,通讯台点不便于流动。为使所建立的通讯系统,即经济又可流动,当然,希望用PC-1500这类计算机来完成通讯任务。要把PC-1500这样一类计算机用于通讯,首先,就要碰到运算速度慢(不能快速算出CRC值,致使应答时间较长。)和所用语言有限等问题。

这些问题怎么解决呢?

## 一、CRC校验的原理[2]

CRC校验的原理是基于二进制信息多项式的表示原理。假设现有N比特信息 $a_1 \dots a_n$ ,则可用多项式 $P(x)$ 表示成如下式子:

$$P(x) = a_1 + a_2x + \dots + a_nx^{n-1}$$

式中 $a_n$ 为二进制信息0或1。

令 $P(x)$ 服从下列代数运算规律:

·加法以2为模的加法。也就是说,若 $a_Kx^K + b_Kx^K = C_Kx^K$ ,则 $a_K, b_K, C_K$ 的关系是逻辑运算中异或(V)关系,其真值如表1所示。

表1.  $a_Kx^K + b_Kx^K = C_Kx^K$ 的真值表

$C_K$	$a_K$	$b_K$
0	0	0
1	0	1
1	1	0
0	1	1

·减法与加相同。即 $-a_Kx^K = a_Kx^K$ 。

·乘除法服从一般的代数规律。

再令多项式 $G(x)$ 完全能表征循环码多项式,并命名为生成多项式。循环码的错码检查和纠正能力,取决于如何规定生成多项式 $G(x)$ 。这里,我们设编码前的信息多项式 $P(x)$ 为:

$$P(x) = a_1 + a_2x + \dots + a_{n-1}x^{n-1}$$

生成多项式 $G(x)$ 为:

$$G(x) = 1 + b_1x + \dots + b_{K-1}x^{K-1} + x^K$$

设 $x^K$ 与信息多项式 $P(x)$ 相乘再除以 $G(x)$ 后的余数多项式为 $R(x)$ ,则有:

$$x^K \cdot P(x) = Q(x) \cdot G(x) + R(x)$$

式中: $R(x) = C_1 + C_2x + \dots + C_Kx^{K-1}$ 为余数多项式。此余数多项式的系数 $C_1C_2 \dots C_K$ 为校验比特。

循环编码信息是将校验比特加到原信息比特而得到的。其表达式为:

$$V = [a_1a_2 \dots a_n c_1 \dots c_K]$$

## 二、实现CRC校验的方法

实现CRC校验,首先就应选定一个收、发双方所共用的一个生成多项式。生成多项式的确定应根据各自通讯要传送的信息而定。目前,常用的循环码生成多项式种类较多,因为,我们的通讯,主要用于传送5个字符为一组的大量地震数据,所以,我们选择 $G(x) = 1 + x^5 + x^{12} + x^{16}$ 为生成多项式; $P(x) = a_1 + a_2x + \dots + a_nx^{n-1}$ 为信息多项式。

有了生成多项式和信息多项式,我们就可以根据已有的信息作长除运算求出CRC值。但是,长除运算的工作,由PC-1500完成,就不能满足通讯速度的要求,为解决这一问题,我们采用了并联处理字符的办法:

### 1. 构造CRC校验的真值表

先列出8个字符,10000000,01000000,00100000,00010000,00001000,00000100,00000010,00000001,再用生成多项式 $G(x)=1+x^5+x^{12}+x^{16}$ 对字符进行长除后的余式分别为 $R(x)_8, R(x)_7, \dots, R(x)_1$ ,并将余式的系数填入表2。例如,对于字符10000000,其信息多项式 $P(x)=x^7$ ,则 $(x^{16} \cdot x^7) \div (x^{16}+x^{12}+x^5+1) = (x^7+x^8) \dots (x^{15}+x^{12}+x^8+x^7+x^8)$ ,即可在表2中第一行( $R_8$ )的对应位( $x^{15}, x^{12}, x^8, x^7, x^8$ )处置1,其余 $R_i$ 值的求法与 $R_8$ 的求法相同,这样就把8个字符的真值表作好了。

### 2. 真值表的使用

当有了真值表2,我们对任一字符,如01010001,就可从表2中查得 $R_7=0100100011000100$ , $R_8=0001001000110001$ , $R_1=0001000000100001$ ,再作 $R_7 \vee R_8 \vee R_1$ 运算,即可得到CRC值(0100101011010100)。由于通信中被传送的信息(bit)串,总是由字符组成,所以,我们就可以根据要发送的信息按字符各(bit)位的情况,在表2中查出相应的值,最后组成被传送的信息式为:

$$V(x) = [a_1 \dots a_n c_1 \dots c_K].$$

接收端,将接收到的信息串称之为 $V'(x)$ ,对 $V'(x)$ 进行解码(将 $V'(x)$ 中的 $a_1' \dots a_n'$ 和 $c_1' \dots c_K'$ 分离出来)后,用与发送端相同的真值表,查出 $a_1' \dots a_n'$ 的CRC值,令为 $C''_1 \dots C''_K$ 。如果 $C''_1 \dots C''_K = C'_1 \dots C'_K$ ,就判定接收无错,否则,就判定接收有错。由此,也就能控制收、发进行的顺序,实现了ARQ方式的通讯。

根据上面的思想,我们对真值表2再进行了加工,把 $R_i$ 各自所对应的值变为十六进制的固放于指定的内存单元中,以备通讯时查用。下面介绍用软件方法实现查表求CRC值的方法。我们

在软件编写中,用BASIC语言处理有关人、机界面事务,而大量的检错处理工作,是用汇编指令来完成的。因限于文章的篇幅,在此只能给出PC-1500计算机发送时求CRC值的程序:

; 程序清单

```

10 DIM A$(20)*64, B$(20)*6
20 I=1
30 INPUT "A$(I) = "; A$(I)
40 S=LEN(A$(I))
50 IF S=0 THEN 105
60 T=64-S
70 FOR J=1 TO T
80 A$(I)=A$(I)+" "
85 NEXT J
90 I=I+1
100 GOTO 30
105 I=I-1
110 I1=I
120 POKE 32765, I
130 G=16399
140 I=I1
150 FOR J=1 TO I
160 FOR K=1 TO 64
170 C$=MID$(A$(J), K, 1)
180 H=ASC(C$)
190 G=G+1
200 POKE G, H
210 NEXT K
220 G=G+2
230 NEXT J
235 POKE &4000, &10, &21, &20,
&42, &40, &84, &81, &08, &12, &31, &24,
&62, &48, &C4, &88
240 CALL TCOO
250 Z=16399
260 I=I1
270 FOR J=1 TO I
280 Z=Z+64
290 B$(J)=" "
300 FOR K=1 TO 2
310 Z=Z+1
320 Y=PEEK Z
330 B$(J)=B$(J)+STR$(Y)
340 NEXT K
350 A$(J)=A$(J)+B$(J)
360 NEXT J

```

执行发报程序

```

7C00 OD EOR(x)
7C01 FD 48 LDX S
7C03 AA 7F FF LDI S, 7FFF H
7C06 FD 88 PSH X
7C08 48 40 LDI XH, 40H
7C0A 4A 10 LDI XL, 10H
7C0C AA 7F FC LDI S, 7FFCH
7C0F B5 3F LDI A, 3FH
7C11 FD C8 PSH A
7C13 68 00 LDI UH, 00H
7C15 6A 00 LDI UL, 00H
7C17 58 08 LDI YH, 08H

```



```

7C19 5A 00      LDI YL, 00H
7C1B 50          INC YL
7C1C 05          LDA(x)
7C1D D5          SHR
7C1E 0E          STA(x)
7C1F B1 03      BCR +03H 7C24H
7C21 BE 7D 00   SJP 7D00H
7C24 FD 52      DEC YH
7C26 93 0D      BCS-ODH 7C1BH
7C28 44          INC X
7C29 FD 8A      POP A
7C2B DF          DEC A
7C2C FD C8      PSH A
7C2E 93 19      BCS-19H 7C17H
7C30 A4          LDA UH
7C31 41          SIN X
7C32 24          LDA UL
7C33 41          SIN X
7C34 AA 7F FC   LDI S, 7FFCH
7C37 FD 8A      POP A
7C39 DF          DEC A
7C3A FD C8      PSH A
7C3C 93 2F      BCS-2FH 7COFH
7C3E AA 7F FD   LDI S, 7FFDH
7C41 FD 0A      POP X
7C43 FD 4E      STX S
7C45 9A          RTN

```

```

7D00 FD 88      PSH X
7D02 14          LDA YL
7D03 D9          SHL
7D04 FB          SEC
7D05 B1 02      SHI A, 02H
7D08 B3 00      ADI A, 00H
7D0A 0A          STA XL
7D0B B5 40      LDI A, 40H
7D0D B3 00      ADI A, 00H
7D0F 0B          STA XH
7D10 A4          LDA UH
7D11 0D          EOR (x)

```

```

7D12 28          STA UH
7D13 44          INC X
7D14 24          LDA UL
7D15 D0          EOR (x)
7D16 2A          STR UL
7D17 FD 0A      POP X
7D19 9A          RTN

```

上面的程序就是查表求CRC值的具体操作，用类似方法，就可编写出PC—1500计算机收报时求CRC值的程序。

### 三、查表结果与长除结果的对应关系

上面给出的程序是查表法的应用一例，这种查表法求出的CRC值是否与长除法算出的CRC值相等呢？下面，我们用两个例子来回答这个问题。

设有任意两个字符：01010001, 11111111

1. 01010001字符的CRC值？

1.1. 用查表法求01010001字符的CRC值  
从表2上查得：

$$R_7 = 0100100011000100$$

$$R_5 = 0001001000110001$$

$$R_1 = 0001000000100001$$

那么， $R_7 \oplus R_5 \oplus R_1 = 0100101011010100$

1.2. 用长除法求01010001字符的CRC值

对于01010001字符，我们可以看成是有8 bit 的一个信息串，由此可得：

$$P(x) = 1 + x^4 + x^6$$

$$G(x) = 1 + x^5 + x^{12} + x^{16}$$

$$\begin{array}{r}
 x^6 + x^4 + x^2 \\
 x^{16} + x^{12} + x^5 + 1 \overline{) \quad} x^{22} + x^{20} + x^{16} \\
 \underline{x^{22} + x^{18} + x^{11} + x^6} \phantom{0000000000000000} \\
 x^{20} + x^{18} + x^{16} + x^{11} + x^6 \phantom{0000000000000000} \\
 \underline{x^{20} + x^{16} + x^6 + x^4} \phantom{0000000000000000} \\
 x^{18} + x^{11} + x^6 + x^4 \phantom{0000000000000000} \\
 \underline{x^{18} + x^{14} + x^7 + x^2} \phantom{0000000000000000} \\
 x^{14} + x^{11} + x^6 + x^7 + x^6 + x^4 + x^2
 \end{array}$$

余式： $R(x) = x^{14} + x^{11} + x^6 + x^7 + x^6 + x^4 + x^2$

余式 $R(x)$ 中没有的项( $x^{15}$ 、 $x^{13}$ 、 $x^{12}$ 、 $x^{10}$ 、 $x^8$ 、 $x^5$ 、 $x^3$ 、 $x^1$ 、 $x^0$ 均补为零，这样就求出了01010001字符的CRC值为0100101011010100。

2. 11111111字符的CRC值？

2.1. 用查表法求11111111字符的CRC值  
从表2上查得：

$$R_8 = 1001000110001000$$

$$R_7 = 0100100011000100$$

$$R_6 = 0010010001100010$$

$$R_5 = 0001001000110001$$

$$R_4 = 1000000100001000$$

$$R_3 = 0100000010000100$$

$$R_2 = 0010000001000010$$

$$R_1 = 0001000000100001$$

$R_8 \oplus R_7 \oplus R_6 \oplus R_5 \oplus R_4 \oplus R_3 \oplus R_2 \oplus R_1 = 000111011110000$

2.2. 用长除法求11111111字符的CRC值计算方法与1.2节相同。

$$P(x) = 1 + x + x^2 + x^3 + x^4 + x^5 + x^6 + x^7$$

$$G(x) = 1 + x^5 + x^{12} + x^{16}$$

$$x^{16} \cdot P(x) \div G(x) = x^7 + x^6 + x^5 + x^4 \dots x^{12} + x^{11} + x^{10} + x^9 + x^7 + x^6 + x^5 + x^4,$$

余式 $R(x) = x^{12} + x^{11} + x^{10} + x^9 + x^7 + x^6 + x^5 + x^4$ , CRC值为0001111011110000。

从以上两例可以看出,对于任意字符的CRC值,用查表法求得结果与长除法算出的结果完全一样。

**结束语:** 通过上述比较计算过程,我们可以确信查表法是可行的。用这种方法,使PC-1500这类微型计算机的通讯功能得到较大地扩充,也就很好地解决了通信速度慢的问题,在实际通

信中,就能代替PCXT机而完成通讯任务。我们用这种查表法成功地研制出了“计算机远程数据通讯”网络软件,在该网络软件中,不论是主台(中心台)或子台都是用查表法作收、发时的CRC校验。该网络软件从1987年6月通过国家地震局鉴定后,陆续在全国十多个省地震局投入运行至今,还没有用户反应在差错控制方面的问题,并且,获得1989年四川省优秀软件二等奖。在此还需说明的事,在资料成文的过程中,李谊瑞老师给予了多方面的指导,特致谢意。

### 参考资料

- [1] 一秋明《一种快速产生CRC的算法》计算机世界月刊1988年第11期
- [2] 一王英彬编《计算机网络原理结构及协议》科学院科海培训中心

表2 8字符对应于 $G(x) = 1 + x^5 + x^{12} + x^{16}$ 的真值表

	$x^{16}$	$x^{14}$	$x^{13}$	$x^{12}$	$x^{11}$	$x^{10}$	$x^9$	$x^8$	$x^7$	$x^6$	$x^5$	$x^4$	$x^3$	$x^2$	$x^1$	$x^0$
$R_8$	1	0	0	1	0	0	0	1	1	0	0	0	1	0	0	0
$R_7$	0	1	0	0	1	0	0	0	1	1	0	0	0	1	0	0
$R_6$	0	0	1	0	0	1	0	0	0	1	1	0	0	0	1	0
$R_5$	0	0	0	1	0	0	1	0	0	0	1	1	0	0	0	1
$R_4$	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0
$R_3$	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0
$R_2$	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0
$R_1$	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1

## 中心提升——从数据流图到模块结构图的直接转换方法

山东经济学院 王治宇

**摘要** 本文在对数据流图与模块结构图相互关系讨论的基础上,提出了从数据流图到模块结构图的直接转换方法——中心提升法,提出了转换原则和实施步骤,进而在计算机上实现了这一转换。这一方法对软件需求分析阶段与软件设计阶段的自动连接将是有益的。

按照生命周期法进行软件开发时,在需求分析阶段要画出数据流图(DFD: Data Flow Diagrams),在软件设计阶段要画出模块结

构图(MSC: Module Structure Chart),两图结构不同、描述对象各异、又往往由不同的设计者完成。传统的设计方法也是从DFD转换

产生 MSC, 但转换不够直接, 转换过程往往与人的水平、经验乃至习惯等主观因素有关, 给转换带来一定困难。如能找到两图的内在联系, 寻求出从DFD到MSC的直接转换方法, 将有助于实现需求分析阶段到设计阶段的自动转换, 本文试对上述问题作一点粗浅的讨论。

## 一、DFD与MSC内在联系分析

### 1. 两图结构上对应关系

我们知道, DFD 是描述系统内数据的加工与存贮的。DFD 中的一个圆圈表示一个加工(或称处理), 圆圈内标有加工的名字。DFD 以箭头表示数据流, 箭头所指为数据流的方向, 箭杆上标有数据流的名字, 一个系统的 DFD 是由多层数据流图组成的立体构造。由于某些加工在某一层 DFD 上已经成为基本加工而被停止分解, 因而在以下各层 DFD 中将不再出现这些加工, 且以下各层数据流图都只是一些互不关联的碎片。一般地说, 较高层次(如顶层或0层)的DFD是完整的, 它们是系统全部功能的描述。我们称这些层次的DFD为完整层。而较低层次的DFD多出现碎片, 它们不能描述系统的全部功能, 我们称这些层次的DFD为不完整层。为了实现DFD到MSC的直接转换, 我们需要把用以转换的不完整层DFD转化为完整层, 这一转

化过程称DFD的完整化处理。其基本方法是在父图中拷贝碎片来填补子图, 使子图成为完整层DFD。具体步骤如下:

①将指定层次的DFD碎片按其逻辑顺序排列, 参照父图与子图的平衡(匹配)关系, 对可连接的碎片进行连接。

②对残缺部分, 到本层的父图中拷贝, 并再作连接。

③若不能在父图中取得正确的碎片, 可逐层向上寻找, 直至将本层DFD填补成完整层DFD为止。

图1(a)中, 第0层和第一层是完整层, 第二层为不完整层。因为在第一层中, “输出结果”加工已经是基本加工, 不再分解。因此, 第二层中因缺少此加工而成为不完整层。若将第一层DFD中“输出结果”加工拷贝来填补第二层DFD可使其完整化为完整层。

模块结构图(MSC)是用以描述系统的结构及其调用关系的。在MSC中, 用方框表示模块, 框内标明模块名, 用箭头表示控制流和数据流。系统的MSC是一个树型结构, 由于某层次上一些模块已经是基本模块而被停止分解。因此, MSC是一个非等长枝树。某些层次上模块的集合可实现系统的全部功能(完整层), 某些层次上模块的集合不能实现系统的全部功能(非

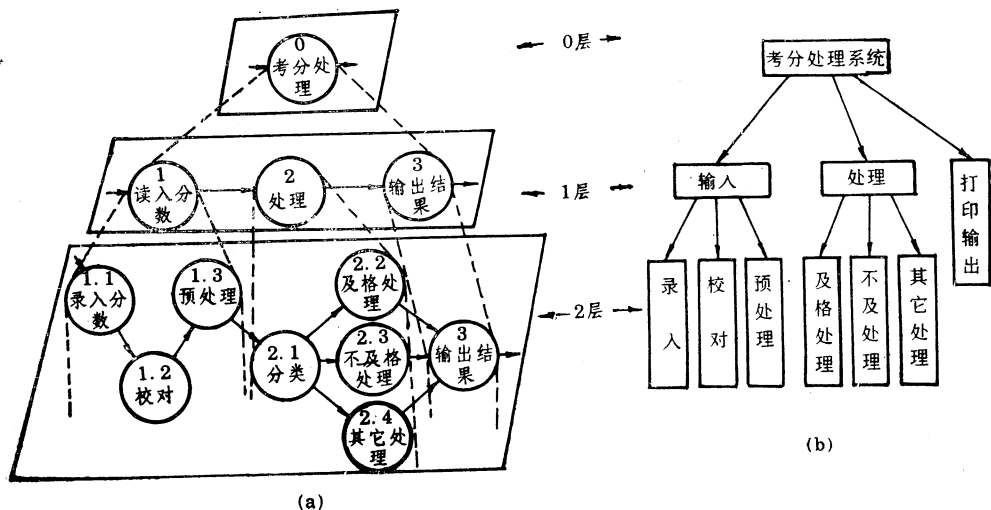


图 1

完整层)。类似于完整化 DFD 的办法可使非完整层 MSC 完整化。

DFD 说明“做什么”，MSC 说明“怎样做”。但两图都是对系统功能的描述。图 1 (a) 和图 1 (b) 是一个《分数处理系统》的 DFD 与 MSC。

对照图 1 (a) 和图 1 (b) 可见，一个系统的 DFD 与 MSC 有着必然的内在联系，在结构上有着良好的对应关系：

① DFD 中每一较高层次上的加工与 MSC 中相应层次上的管理模块相对应。这是因为，DFD 中每一个父加工的功能总是由其所属的几个子加工共同完成的。在 MSC 中，某一管理模块的功能也总是由其所属若干子模块共同实现的。

② DFD 中每一基本加工与 MSC 中相应层次上的某一基本模块相对应。

③ 完整层 DFD 与相应的完整层 MSC 相对应，且它们都是系统全部功能的描述。因为，同一层次上（完整层）所有加工的完成便是系统功能的实现。同理，同一层次（完整层）上所有模块功能的实现也是系统功能的实现。

2. DFD 中数据流方向与 MSC 中控制流方向的关系

系统的输入部分，在 DFD 中某一数据流从前一加工指向后一加工，其意义是数据流经前一加工处理后进入后一加工做后续处理。一般地说，前一加工总是较初级加工，而后一加工则是较深度的加工，这一关系在 MSC 中体现为：一个较高层或较综合的处理总是由几个较低层或较简单的处理实现的，则这几个较低层的处理模块就受控于那个较高层的处理模块，控制流的方向由控制模块指向受控模块。因此，我们说，在系统的输入部分，DFD 中数据流方向与 MSC 中相应控制流方向相反。

经类似的分析，我们说：在系统的处理部分和系统的输出部分，DFD 中数据流方向与 MSC 中相应控制流方向相同。

## 二、中心提升——从 DFD 到 MSC 的直接转换方法

原则上讲，任意一层（完整层）DFD 都

可以转换得到 MSC。但是，由于显然的原因，一般都是用 DFD 的最低层进行转换（这一层上的加工都是基本加工）。中心提升法的实施步骤是：

1. 确定用以转换的 DFD 层，并对其进行完整化处理。
2. 复审并精化 DFD。
3. 确认 DFD 的类型，并实施转换得到 MSC。
4. 对 MSC 做优化处理。

对非完整层 DFD 做完整化处理的方法前面已经谈到。复审与精化处理可参照常规方法进行。确认 DFD 类型（变换型或事务型）的方法也与常规方法相同。这里，我们把非单一类型的 DFD 称复合型 DFD。

下面，就变换型 DFD、事务型 DFD 及复合型 DFD 的转换，分别说明中心提升法的应用。为使问题简化，我们假定所用的 DFD 均已做过完整化处理，并经过复审和精化。

### 1. 由变换型 DFD 转换成 MSC

变换型 (Transform) DFD 具有较明显的输入、变换中心和输出界面，如图 1 (a) 所示。把这种 DFD 转换成 MSC 的关键是找出 DFD 的变换中心（也称中心加工或主加工）。转换步骤为：

① 选定某层 DFD 为转换对象，并对其做完整化、复审和精化处理。

② 去掉所有局部文件，找出逻辑输入加工、变换中心和逻辑输出加工。

③ 画出顶层模块，提升逻辑输入、变换中心和逻辑输出至顶层模块下，做为第二层模块。

④ 其它加工以连线 (DFD 中数据流方向线) 为据自然下垂，做为下级模块。

⑤ 将圆圈改为方框，将输入部分箭头反画，标明模块名、控制流名和数据流名。

⑥ 优化 MSC。

例 1. 将图 2 (a) 所示 DFD 转换成 MSC。

这是一个变换型 DFD，找出逻辑输入加工、变换中心和逻辑输出加工，设计顶层模块 M，按上述步骤直接得到 MSC 如图 2 (b)。

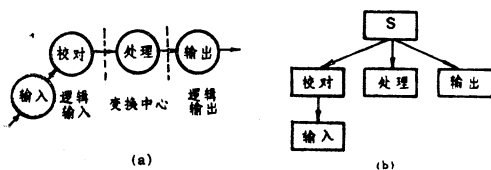


图2

有的系统可能没有变换中心,其逻辑输入加工就是逻辑输出加工,转换时,只需把逻辑输入加工、逻辑输出加工提升到顶层模块下即可。

对于有多输入或输出的DFD,应逐个找出它们的逻辑输入、逻辑输出,并确定变换中心(变换中心也可能由多个加工组成),然后用同样方法转换。

图2. 试将图3(a)所示DFD转换成MSC。

这是一个多输入多输出的DFD,找出各支路的逻辑输入、逻辑输出和变换中心即可得到MSC,如图3(b)所示。

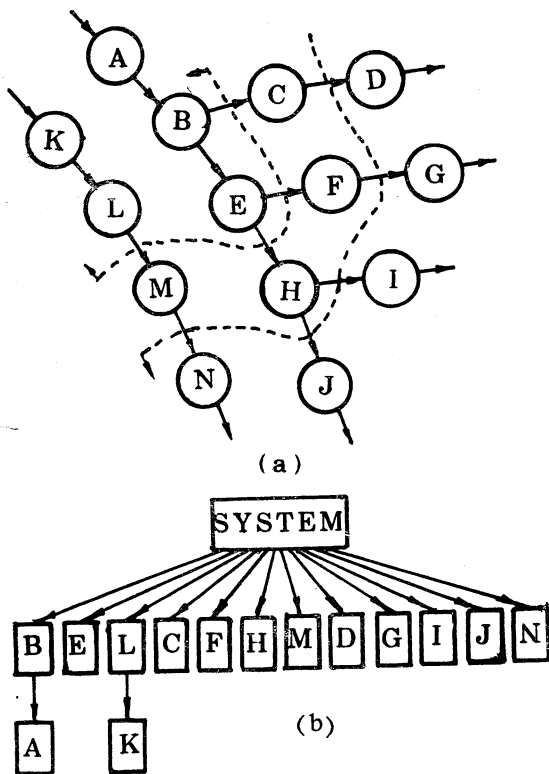


图3

优化:为减轻顶层模块的负担,可适当增设中间层次的管理模块。如输入管理模块、处理管理模块和输出管理模块等。

应当指出,逻辑输入、逻辑输出乃至中心加工的指定带有一定的经验性,这将导致产生不同的MSC。但只要指定不出现错误,转换得到的MSC至少是可用的。

## 2. 由事务型DFD转换成MSC

事务型(Transaction)DFD的特点是:图中某一加工将其输入的数据流分离成一束发散的数据流,形成多条活动路径。这一加工称作前事务中心,若诸活动路径(或大多数活动路径)又汇集入某一加工,则此加工称后事务中心。图4(a)是一典型的事务型DFD。

转换事务型DFD时,只需将前事务中心提升为管理模块,所有支路做为二级模块即可。若有后事务中心存在,则将其与各支路并列为二级模块。

转换步骤为:

- ①选定某层DFD为转换对象。对其做完整化、复审和精化处理。
- ②去掉所有局部文件。
- ③找出前事务中心(若有后事务中心也一并找出)。
- ④提升前事务中心为管理模块,各支路(若有后事务中心也包括在内)作为二级模块。其它加工以连续(数据流方向线)为据自然下垂,作为下级模块。
- ⑤将圆圈改为方框做为模块,输入部分箭头反画,标注模块名、控制流名和数据流名。
- ⑥优化MSC。

例3. 将图4(a)所示DFD转换为MSC。

在图4(a)中,加工①为前事务中心,加工⑦为后事务中心。按上述转换步骤立即可得MSC如图4(b)所示。

至此我们看到,将DFD转换为MSC是轻而易举的。现在打个比喻来说明中心提升法。

把一个DFD看做是一些用线串起来的圆环,使所有圆环处于同一垂平面内。转换时先找到中心(变换中心或前事务中心),并将其提



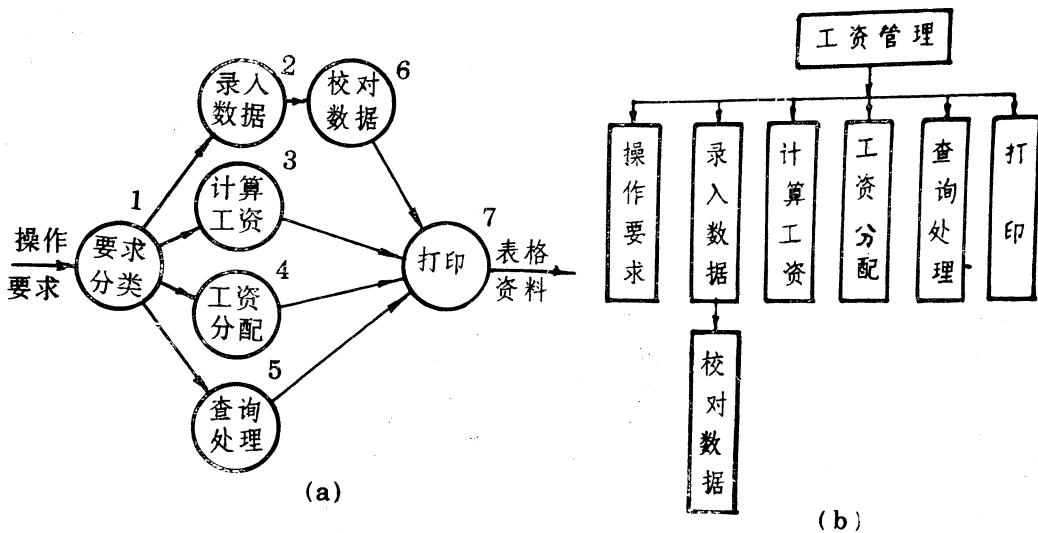


图4

升。若是变换型，需将中心与逻辑输入、逻辑输出一起提升；若是事务型，则只提升前事务中心。让其它圆环自然下垂。再将圆环改成方框并标注箭头、名字即成。

### 3. 由复合型DFD转换成MSC.

一个稍大一点系统的DFD，往往不是单一类型的。通常两种结构兼有，甚至互相嵌套形成一个复杂的结构。我们称这种DFD为复合型DFD。当一个DFD中又含有某一类型的DFD时，我们称这个DFD含有变异分枝。

复定型DFD从总体上看总可以唯一地确定其类型。由于问题的复杂性，DFD的变异分枝可能还含有变异分枝。复合型DFD的转换步骤为：

①选定转换层，作完整化、复审和精化处理。

②去掉所有局部文件。

③以各分枝的逻辑输入、逻辑输出为断点去掉全部变异分枝，使DFD成为单一类型结构(得到母体DFD)。并在去掉分枝处标以分枝编号。

④确定母体DFD的类型，并实施转换的母体MSC。

⑤对不再含有变异分枝的变异分枝，确定其类型，予以转换，并按编号恢复到母体MSC上。

⑥对仍含有变异分枝的分枝，重复③、④、⑤各步，直至全部转换完成。

⑦优化MSC。

例4 用中心提升法转换图(5)a所示DFD。

这是一个复合型DFD，从总体上看它是一个变换型DFD。而输入部分和变换中心是两个

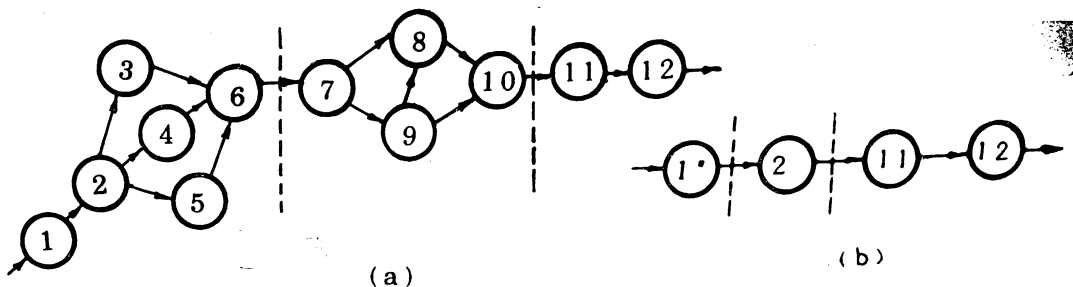


图5

变异分枝, 去掉变异分枝得到母体 DFD 如图5(b), 对其实施转换得到母体 MSC 如图6(a)。

两个变异分枝都是事务型的, 且不再含有变异分枝。分别转换得图6(b)和图6(c)。将图6(b)和图6(c)装配到图6(a)上, 得图6(d)所示 MSC。

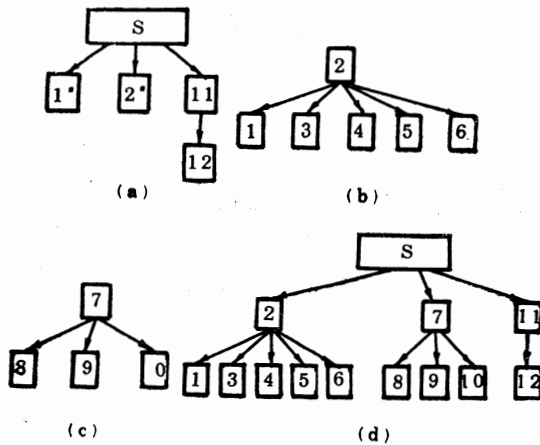


图 6

例5. 图7(a)是一个物资销售部门有关销售业务的DFD, 试将其转换成MSC。

去掉局部文件, 图7(a)分离成三个独立部分, 如图7(b)、图7(c)和图7(d)。

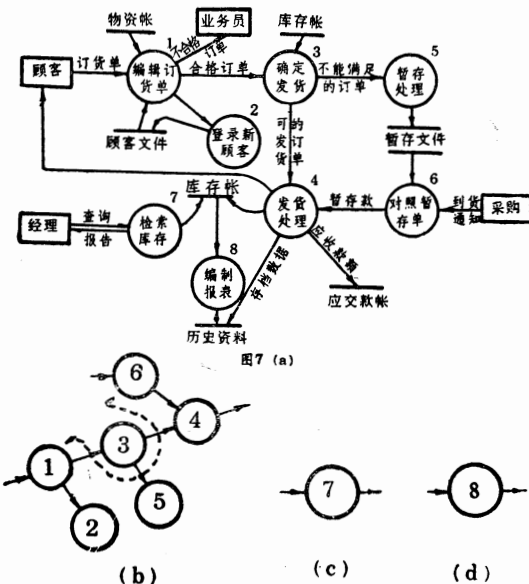


图 7

第一部分(图7(b))是事务型 DFD, 加工③为前事务中心。对其实施转换得到图8(a)所示 MSC。再将其两部分(图7(c)和图7(d))予以转换, 并装配到图8(a)上得图8(b), 此即为所求 MSC。

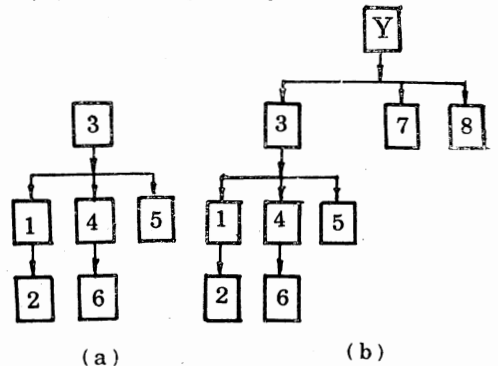


图 8

### 三、从 DFD 到 MSC 转换的计算机实现

显然, DFD (用以转换的) 和 MSC 都是有向图结构, 把 DFD 转换成 MSC 就是将前一图转换成后一图。中心提升法提供了转换的原则和途径。本人在微机上用 dBASE III 实现了这一转换。

转换程序运行时, 先接收 DFD 的描述信息入数据库: DFD. DBF 中, 然后要求户指出每个 DFD (母体的和分枝的) 的类型后, 程序将转换得到的 MSC 的描述信息装入数据库: MSC. BDF 中。

图9(a)的数据是图5(a)的描述数据, 图9(b)是转换后 MSC 的描述数据。

图9(a)和图9(b)中符号“^”表示“空”。图9(b)中“父块”栏中, 符号“S”表示 MSC 的顶层模块。可见, 根据 MSC. DBF 中的数据很容易画出所求之 MSC。

本程序的缺点在其自身还不能判断 DFD 的类型, 不能判断变异分枝的断点。这再经过努力后是可以作到的。

本程序对 DFD 的描述需人工录入。若软件需求分析阶段已经在计算机上实现 (DFD 描述已在计算机中), 则转换自动化程度将进一步提高。

编号	左链	右链
1	^	2
2	1	3 4 5
3	2	6
4	2	6
5	2	6
6	3 4 5	7
7	6	8 9
8	7	10
9	7	10
10	8 9	11
11	10	12
12	11	^

图 9 (a)

编号	父块	子块
1	2	^
2	S	1 3 4 5 6
3	2	^
4	2	^
5	2	^
6	2	^
7	S	8 9 10
8	7	^
9	7	^
10	7	^
11	S	12
12	11	^

图 9 (b)

为了把中心提升法与常规方法作以比较, 这里给出图5 (a) DFD 的常规转换结果如图 10。该图请参见《软件开发技术》(上海科学技术出版社) 第140页图3.36。

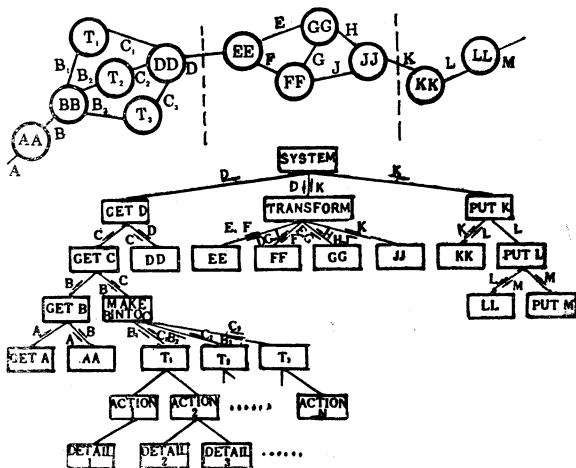


图 10

## 一个微机“病毒”的感染、激活原理及其防治办法

湖南省电子研究所

马小宏

目前, 一个微机病毒(下称病毒)在国内微机机迅速蔓延开来, 降低了微机使用效率, 产生了一些不良影响。该病毒的表现形式是: 在英文方式下, 屏幕上出现一个类似台球运动的园点; 在中文方式下, 屏幕滚动反应极慢。打印24点阵汉字时, 导致机器死锁。FORMAT/S命令格式化的系统盘不能启动, 3+网络系统软件失效…。对此, 国内报刊已作了一些报道。

本文分析该病毒的感染、激活原理, 探讨防治办法。

### 一、病毒的病理机制、感染激活条件

病毒由病毒装入模块, 病原体控制模块、感染模块和激活模块组成。物理上, 病毒装入模块位于带病盘的逻辑第0扇段, 而其它三个模块位于带病盘数据区, 这些模块的逻辑关系见图1。

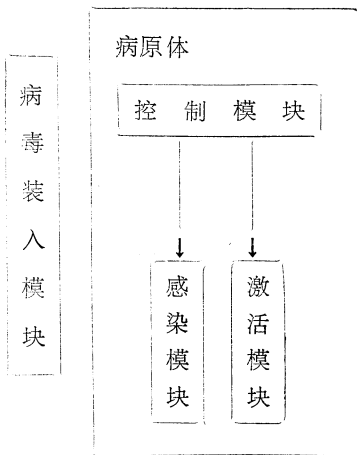


图 1

上述模块的功能分别是：病毒装入模块负责将病原体装入内存，且将INT 13软中断（磁盘I/O中断）改向病原体的控制模块。控制模块的任务是判别病毒感染和激活条件，调用感染、激活模块。感染模块的功能是：向所有进行读盘操作的磁盘感染病毒，而激活模块则是在屏幕上显示园点，降低系统效率。这些模块的逻辑流图见图2。

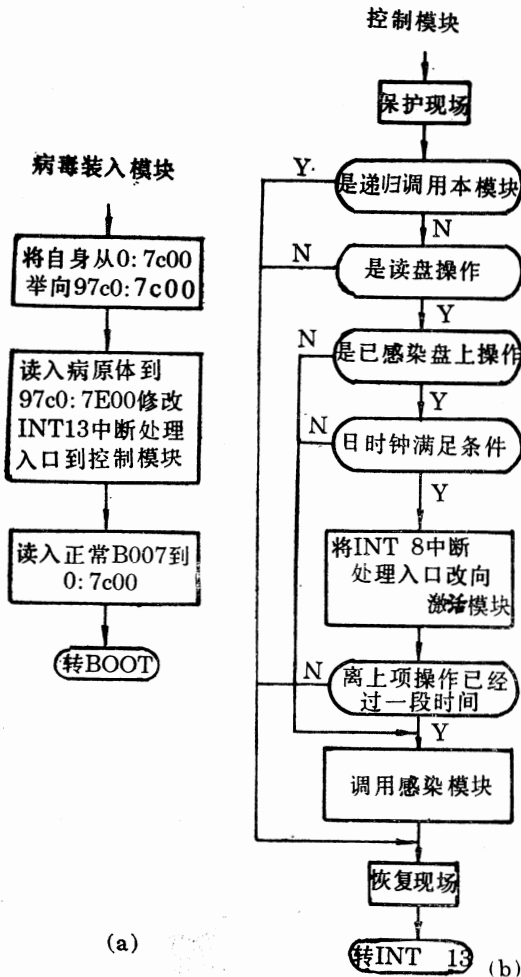


图2

正常情况下，系统自举过程是：开机后，由BIOS对系统基本设备进行检测，然后读入系统盘逻辑第0扇段（BOOT）到0000:7C00，且转入BOOT，BOOT判断磁盘是否系统盘，读入BIO，转入BIO。而带病盘逻辑第0扇已不是正常的BOOT而是病毒装入模块，因而BIOS

将其装入0:7C00后即转入病毒装入模块，病毒装入模块装入病原体，改变INT 13入口到病原体控制模块后，再装入正常BOOT执行之，见图3。

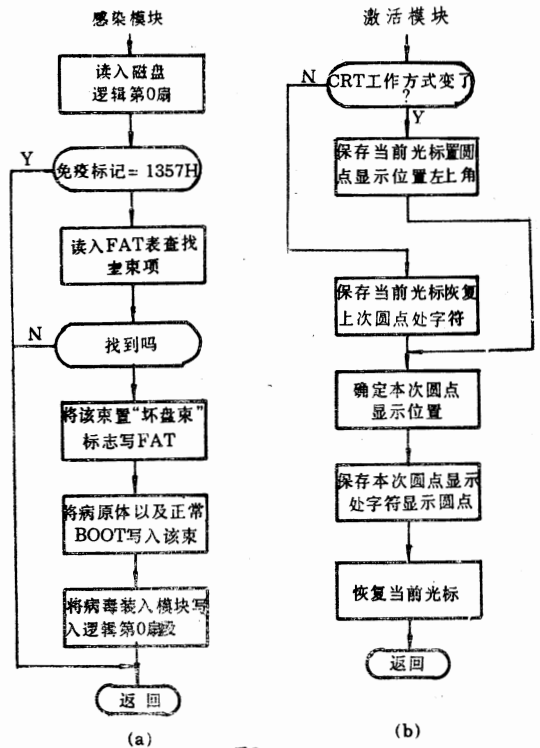


图3

由上述分析可知：病毒是系统自举时侵入系统的，由于INT 13中断处理权已由病原体控制模块接管，因此，凡在被病毒侵入的系统上进行读盘操作的磁盘均可能感染病毒；而由正常盘启动的系统，即使是对带病盘进行操作，也不可能感染、激活病毒。病毒的激活是病原体控制模块将INT 8中断（硬时钟中断）处理入口改向激活模块时发生的，以时钟作为随机函数，只有当该函数值满足一定条件时才可能激活病毒。磁盘被感染后将损失一个盘束的磁盘空间，但不会损失任何数据（因此该病毒称为良性病毒）。病毒侵入的系统还损失2KB内存空间。

## 二、病毒的防治办法

防止病毒感染有如下办法：（一）是保证系统不被病毒侵入，要做到这点关键是做到用正常

## 计算机圆点“病毒”的诊断及排除

南京化工动力专科学校

王庆庆

**摘要:** 本文简述计算机“病毒”程序的工作原理, 重点介绍了“病毒”的诊断与排除方法。

计算机圆点“病毒”是驻存在计算机内部, 用以干扰计算机正常运行的程序。它可以使得计算机工作异常、降低运行速度、消耗计算机资源, 在任何一种应用程序运行时都有可能诱发。该“病毒”具体表现为, 西文方式和中文高分辨方式下, 屏幕上显示类似台球运动的轨迹点; 中文中分辨方式下, 屏幕上下滚动, 且有小球在屏幕上移动, 而且反映速度极慢。计算机系统一旦染上这种“病毒”, 不但干扰机器自身的工作, 而且还会污染其它DOS系统, 其传播速度极快。

### 一、计算机“病毒”的工作原理

计算机“病毒”程序一般可分为三部分: 自举部分、传播部分和表现部分。

“自举部分”首先将“病毒”程序装入内存, 而后转入DOS系统引导部分的入口执行正常的启动程序。

“传播部分”当计算机执行读盘操作时, 如果该盘是未被污染的, 那么“病毒”传播程序首

先在该盘中找一个空闲簇, 将“病毒”程序的传播部分和表现部分写入该簇的第一扇区, 而后将该盘引导扇区的内容搬家到该簇的第二扇区, 并将该簇对应FAT表中的单元作上坏簇标记(FF7H), 最后将“病毒”的引导部分写入盘的引导扇区。这样整个传播工作就完成了。

“表现部分”当“病毒”程序认为时机成熟时, 将启动表现程序, 使屏幕出现台球运动轨迹。

### 二、“病毒”的诊断

了解了计算机“病毒”的工作原理, 就可以对计算机进行诊断, 观察是否染上“病毒”。具体诊断方法可以使用诊断程序(见图1)。这段程序可以在DEBUG下用“A”命令将其输入, 然后存入磁盘。运行诊断程序后, 如果计算机带有“病毒”屏幕立即就会有台球运动轨迹; 如果计算机并未染上“病毒”, 那么计算机将会出现异常或死机。

盘启动系统。(二)是在磁盘上置上免疫标记, 由感染模块的逻辑流程图可知: 如盘上已置有免疫标记, 则该盘不可能被感染。免疫标记字在盘上逻辑第0扇区的01FCH偏移处, 将该字[01FCH] (“w”表示内存字, “b”表示内存字节)置为标记字1357H, 即可免疫。当然, 某些系统也可能将[01FD] b字节作为系统盘号标志, 这时要做适当处理, 最好将该标志改在[0002] b字节。(三)是修改系统FORMAT软件, 使其具有免疫能力, FORMAT中有一个BOOT原型, 只需按(二)的方式置上免疫标记即可。

对于带病盘可用如下步骤“治疗”: ①恢复原正常BOOT, 该BOOT已被感染模块拷入某

一“坏盘束”中。将其读出, 写到盘上逻辑第0扇。②置上免疫标记。③复原“坏盘束”“坏盘束”对应的第一个逻辑扇段号已在病毒装入模块的01F9偏移处。将[01F9H] W取出加1后即为正常BOOT所在的逻辑扇段号。

带病盘很易判别: 我们知道, PSP表的0002H偏移处[PSP, 0002H] W应为系统可用内存高端地址(段单位), 而用带病盘启动的系统中, [PSP, 0002H] W的值将比正常情况时小2K, 这是因为病原体占据了内存高端空间。用DEBUG的“Dφφφφ”命令看出的[nnnn, 0002] W的值即为内存可用高端地址。



```

XOR     AX,AX
MOV     DS,AX
MOV     AX,[0413]
MOV     CL,06
SHL     AX,CL
SUB     AX,07C0
MOV     DS,AX
PUSH    AX
MOV     AX,FEA5
MOV     [7FC9],AX
MOV     AX,F000
MOV     [7FCB],AX
CLI
XOR     AX,AX
MOV     DS,AX
MOV     AX,7EDF
MOV     [0020],AX
POP     AX
MOV     [0020],AX
STI
INT     20

```

图1 “病毒” 诊断程序

### 三、排除“病毒”

当确诊计算机染上“病毒”后,就必须立即进行排除。排除“病毒”可以使用 DEBUG 和 PCTOOLS 这两个工具,也可以仅使用 DEBUG。具体步骤如下:

1. 用未被污染的系统盘启动系统;
2. 确定“病毒”所在的扇区号。

这里我们并不是直接寻找“病毒”所在的扇区号,而是去寻找被“病毒”程序搬了家的原引导扇区的内容所在扇区号。寻找的方法是:用 PCTOOLS 中的“E”命令检查第0扇区的内容,可以看见这样的字符串“IBM<sub>□□</sub>X.X”,因为引导扇区从第3个字节开始的以后8个字节是用于存放厂商名和 DOS 版本号的,这部分内容“病毒”是污染不了的,即使“病毒”侵占了引导扇区,但它并未破坏这个字符串,这样被“病毒”程序搬了家的引导扇区内容中也包含了这个字符串,整个盘上就包含了两个这样的字符串。因此我们把这个字符串作为寻找原引导扇区所在位置的标记,用 PCTOOLS 的“F”命令来寻找该字符串,找到第一个后,打入“G”命令继续向后寻找,找到第二个后记下扇区号,并转换为十六进制数,退出 PCTOOLS。

#### 3. 修改引导扇区

进入 DEBUG, 将被搬了家的引导扇区的内容调入内存, 然后打入命令:

—Lcs:  $\phi ds1$ ↵

其中, d 表示驱动器号, s 表示字符串所在扇区号。

—Wcs:  $\phi d \phi 1$ ↵

将引导扇区内容重新写回引导扇区(第0扇)。

#### 4. 修改FAT表

前述当“病毒”写入盘后, 它将其所在的簇对应FAT表的单元作上了坏簇标记“FF7”, 如果不将这个坏簇标记排除, 那么该簇将无法使用, 这无疑给磁盘空间造成浪费, 所以必须找到该簇对应FAT表的位置, 将“FF7”改写为“ $\phi \phi \phi$ ”(空闲簇标记)。具体寻找可通过公式计算获得。对于不同容量的磁盘和不同版本的DOS其计算公式也不同, 表一给出了常用磁盘的计算公式和一些主要参数。

表一

磁盘类型	FAT表			簇号C与扇区号S的关系 $C = \frac{(S-1)}{5}$	DOS版本
	每个占用扇区	首扇区号	单元长度(字节)		
360KB软盘	2	1,3	1,5	$C = \frac{(S-8)}{2} \times 1.5$	V2.0
10MB硬盘	8	1,9	1,5	$C = \frac{(S-33)}{8} \times 1.5$	V2.0
20MB硬盘	8	1,9	1,5	$C = \frac{(S-49)}{16} \times 1.5$	V2.0
20MB硬盘	41	1,42	2	$C = \frac{(S-107)}{4} \times 2$	V3.0

由公式计算出簇号C的值, 并转换成十六进制数, 将最低位改为 $\phi$ , 例如,  $C = 1F8H$  应改为  $C = 1F\phi H$ 。然后进入 DEBUG, 修改表中的坏簇标记。

—LCS:  $\phi db1$ ↵

d表示驱动器号; b表示FAT表的起始扇区号; l表示FAT表的长度。

—DCS: C↵

找到“F7 $\phi$ F” (对于FAT表单元长度为2的应为F7FF) 所在的位置, 用“E”命令将其改写为“ $\phi \phi \phi \phi$ ”, 最后写回磁盘。

因为每个磁盘有两个FAT表, 所以必须用上述方法对第二个表进行修改。

至此, 整个排除“病毒”的工作就全部完成。

# 字库缩减及造字

延安大学数学系：雷学武

**摘要：**本程序能缩减汉字库，释放一些自由空间，且能造字。

由于主机内存空间的不够大，常使一些大型程序 (PROGRAM) 无法运行，或对一些没有扩充内存的主机，汉字系统根本调不进去，即使采用小型汉字库，虽能进入，而用于某一汉化语言或数据库时 (例 CCBASICA 或汉字 DBASE)，却因其文本本身庞大无法进入某一工作状态，当然更不用说用其完成简单任务。解决的办法，或是扩充主机内存，或是缩减字库。常用的方法之一是不需增添任何硬设备的字库缩减。

按照国家标准 GB2312-80，CCDOS 有 87 区，每区 94 位，具有汉字及其它字符 6763 个，其中一级字库 3755 个，二级 3008 个，二级字库的使用频率较低，对于 1—7 区的其它字符来说，使用频率也不高 (相对一级字库，其实常用的置表符也不过 11 个，可适当处理)，因而，缩减字库可用去除二级库，要求更甚者，还可采用覆盖 1—7 区一般字符区的办法来实现。例对 256K 内存来说，采用一级字库 (去掉二级) 可进入 DOS，可用 DOS 命令如 EDLIN 行编辑编程，但若用

编译或链接程序将其转换成可执行文件，内存却不足；或是想用汉字 BASIC 编辑程序却进不去，因而可通过去除二级字库，并移动一级字库后面若干汉字到 1—7 区以缩减字库来实现 (移前的汉字不应超过  $7 \times 94$  个，且该部分汉字只能通过区位码输入实现，当然若放入它区又另当别论，若希望缩减更多汉字或 1—7 区字符更有效，则一部分汉字必需真正删除)。

由于内存空间及运行速度的限制，字库中的汉字并不完全满足我们的要求，例“囍”及二、三重积分符“𠂉”“𠂊”字库中没有，但某些时候却极为需要，我们可通过造字的办法来实现，CCDOS 常带有造字程序，事实上不用之也可通过本程序之造字子程序来实现，且方法比较直观。CCDOS 汉字点阵为  $16 \times 16$ ，每字占 32 字节。点阵数据在字库中存储的顺序是按照汉字在区位码表的出现顺序来存储的 (8—15 区与 16—23 区的汉字一样，在字库中只出现一次)，每八点对应一个 ASCII 字符。本程序的造字是先

## 四、一个排除“病毒”的实例

该实例是一个排除被“病毒”污染的磁盘系统 (DOS 版本为 3.1)。

1. 用未污染的系统软盘启动系统；
2. 用 PCTOOLS 寻找第二个“IBM□□3.1”字符串所在的扇区号， $S=784=31\phi H$ 。
3. 退出 PCTOOLS，进入 DEBUG。  
-LCS:  $\phi 231\phi 1$ ✓  
-WCS:  $\phi 2\phi 1$ ✓
4. 计算簇号，修改 FAT 表，  
 $C = [(784-1\phi 7)/47] * 2 = 34\phi$   
 $= 158H$  改写最低位为  $\phi$ ，亦即  $C=15\phi H$   
-LCS:  $\phi 212$ ✓

-D15  $\phi 15F$ ✓  
CS:  $15\phi 65\phi\phi FFFF\phi\phi\phi\phi 7\phi\phi -$

⋮  
F7 FF  
⋮

$\phi\phi\phi\phi\phi\phi\phi\phi\phi\phi\phi\phi$

-E158✓  
CS:  $158F7.\phi\phi FF.\phi\phi$   
-WCS:  $\phi 122$ ✓

修改第二个 FAT 表：

-LCS:  $\phi 242 2$ ✓  
-E158  
CS:  $158 F7.\phi\phi FF.\phi\phi$   
-WCS:  $\phi 2 42 2$ ✓

整个排除“病毒”工作完成，重新启动系统便可以正常工作了。

## 浅谈利用高级字库3KTZ打印特大汉字

重庆制药七厂计算机室 龙兵生

### 一 引言

打印大汉字的方法,报刊杂志有些介绍,大都是通过修改相应的打印驱动程序来将汉字库中的汉字点阵信息取出,进行点阵的横向和纵向放大打印来实现大汉字打印的。当放大倍数较大时,打印出来的汉字锯齿形状较为严重,不够光滑和美观,并且打印速度较慢。本文以中国计算机技术服务公司开发的高级打印字库之一3KTZ(48X48点阵,二宋字体)为例,介绍使用GWBASIC语言读取高点阵字库汉字信息实现特大汉字打印的一种简易方法,满足用户对特大汉字的需求。

### 二 字库结构

弄清字库的内部结构,是正确读取汉字点阵信息,实现特大汉字打印的一个关键。通过对高级字库3KTZ进行分析、实验发现,该字库名为48X48点阵,实际上是36X48点阵。3KTZ只收集了从16区到37区的汉字,16区以前的内容没有收集进来。每一个汉字点阵信息在字库文件中占270个字节,前135个字节是汉字的上半部份内容,后135个字节是汉字的下半部份内容,每半个汉字信息内容按照从上到下顺序进行排列,每列24个点,共排36列。

### 三 位置计算

根据上述字库结构分析,考虑到GWBASIC语言的记录长度受一定限制(一字节长)以及处理上的方便,故设定每一个记录长度是135个字节,那么一个汉字点阵信息内容需两个记录来描述,第一个记录反映汉字上半部分内容,第二个记录反映汉字下半部份内容。可用下述公式

来计算一个汉字在字库中的第一个记录位置Z。

$$\text{公式: } Z = 188XZ_1 + Z_2 - 33409$$

其中:  $Z_1$  是汉字的高位内码

$Z_2$  是汉字的低位内码

例如 求出汉字“龙”在字库中的记录位置,因有:

$$Z_1 = \text{ASC}(\text{LEFT} \# ("龙", 1)) = 193$$

$$Z_2 = \text{ASC}(\text{RIGHT} \# ("龙", 1)) = 250$$

故有

$$Z = 188X193 + 250 - 33409 = 3125$$

即“龙”字在字库中上半部份内容在第3125记录,下半部份内容在第3126记录。

### 四 点阵信息读取转换及打印特大汉字

汉字在字库中的位置确定以后,可在程序中设置相应的语句将指定的两个记录分别读出,赋给指定的字符串变量,然后将字符串变量由从左到右的顺序逐个取出一个字节,将此字节转换成8个2进制位字符串的形式(例如166可转换成串“10100110”),对转换后的字符串依照从左到右顺序逐位取出字符串内容并将其转换为数值数据,保存到数组M(36, 48)中,两个记录的内容如此处理完毕后,汉字的点阵信息全部保存在数组M(36, 48)中了,此时即可按照汉字点阵信息在数组M(36, 48)中存放规律逐行逐列读出其内容进行分析判断,若是1,在打印纸上打印字符“■”,若是0,在打印纸上打印空格“(2个西文字符长)”,如此打印下去,打印完36行48列后,一个特打汉字出现在你的面前。

将点阵数据放入数组中,调用点阵分析子程序,若绘出的字符满足我们的要求(否则修改DATA语句),则输入要存入位置的区位码(1—7区,否则要覆盖别的汉字,当然应该根据自己的需要

灵活掌握),显示原字符确定后存新字符入库,重新启动汉字系统即可用之。(需程序清单者,请同本刊联系)

## 五 结束语

使用本文提供的程序利用高级字库打印的特大汉字按点阵计算已经是1728X2304大小,比24X24点阵字库的字大了6912倍,一般情况下已能满足人们对特大汉字的需求,如需再放大,可修改程序一、增加横向和纵向的打印次数即可。本文使用的大字库3KTZ长度为1839872字节。附录一是用程序一(程序一已是对汉字进行横向和纵向放大两倍打印)打印出来的内容。使用程序一时,应先调用相应的打印机驱动程序,并保证高点阵字库3KTZ与程序一在同一目录下。程序一运行以后,首先请用户输入需打印的汉字,然后询问是否旋转90度打印,最后请用户输入组成汉字的点阵字,例如可输入“口”、“○”、“●”等等。若直接回车则程序选择字符“■”来组成汉字点阵。为提高打印速度,可将程序一编译后运行。

10句 进行初始化工作,打开字库文件3KTZ,记录长度设为135字节

20句 输一些汉字,并进行检查和设置

60句 根据所输汉字,求出其在字库中的相应位置

70句 根据求出的位置分别读出汉字点阵信息内容,并转化为8个2进制位串的形式保存在C¥中(1列信息量)

120句 打印读出的点阵信息

180句 将一字节转化为8个2进制串子程序

```
10 DEFINT A-Y:DEFDBL Z:DIM H(36,48):KEY OFF:CLOSE:OPEN "R",3,"3KTZ",100:FIELD 3,
100 AS R1$:SCREEN 2:CLS:LOCATE 2,30:PRINT"打印特大汉字程序":PRINT STRING$(
80,"*"):LOCATE 7,1:PRINT STRING$(80,"*")
20 LOCATE 4,10:INPUT "请输入一串汉字":DS:JE=LEN(DS)/2:IF JE=0 THEN END
30 LOCATE 5,10:INPUT "旋转90度打印吗?":DES:10P=0:IF DES="Y" OR DES="Y" THEN 10P
=1
40 LOCATE 6,10:INPUT "请输入打印点阵字":LBUI$
50 FOR JC=1 TO JE
60 J=2*JC-1:SS=HDS(DS,J,2):Z1=100:Z2=ASC(LEFT$(SS,1)):Z3=2*ASC(RIGHT$(SS,1)):Z4
=334091:Z=Z1*Z2+Z3-Z4:NN=INT(Z)
70 GET 3,NN:AAS=RIS:GET 3,NN+1:RAIS=RIS:FOR JK=0 TO 35:K1=JK*3+1:C$="":AS=AAS:FO
R IO=0 TO 2:K2=K1+10:GOSUB 100:C$=C$+BS:NEXT IO:AS=RAIS:FOR IO=0 TO 2:K2=K1+10:G
OSUB 100:C$=C$+BS:NEXT IO
80 FOR J=1 TO 48:BS=HDS(C$,J,1):H(JK,J)=VAL(BS):NEXT J:NEXT JK
90 LBUI$="":IF LBUI$="" THEN LBUI$="■"
100 WIDTH LPRINT 255:LPRINT CHR$(27)+"000":IF 10P=1 THEN 140
110 FOR JD=1 TO 48:FOR NG=1 TO 2:LPRINT TAB(10):FOR JS=1 TO 36
120 IF N(JS,JD)=0 THEN LPRINT LBUI$:LBUI$; ELSE LPRINT LBUI$:LBUI$;
130 NEXT JS:NEXT NG:NEXT JD:GOTO 170
140 FOR JD=1 TO 36:FOR NG=1 TO 2:LPRINT TAB(3):FOR JS=1 TO 48
150 IF N(JD,49-JS)=0 THEN LPRINT LBUI$:LBUI$; ELSE LPRINT LBUI$:LBUI$;
160 NEXT JS:NEXT NG:NEXT JD:GOTO 170
170 NEXT JC:LINE(168,48)-(639,113),0,BF:GOTO 20
180 BS=HDS(AS,K2,1):B=ASC(BS):BS="":FOR N=7 TO 0 STEP -1:Z=B-2*N:IF Z=0 THEN B
=Z:BS=BS+"1" ELSE BS=BS+"0"
190 NEXT:RETURN
```

# 关于《利用 BASIC 产生屏幕美术字的 编程技巧》之我见

开封市第一职业中专 李辉宏

读了贵刊89年第四期中刊登的国营西北机器厂计算机中心翟明德同志撰写的《利用 BASIC 产生屏幕美术字的编程技巧》一文,对其中的 BASIC 程序的编写有以下浅薄的意见:

1. 程序第10语句说明语句,但说明语句若没有指令 REM,那么电脑将中断程序的运行;

2. 第50语句, L=LEN(P\$)+ " ",其中 LEN(P\$) 为数字型,而空格却是字符,

它们之间是不能相加的。如同人数加金额无意义一样。

3. 通过运行他的程序后,发现:大立体、斜宋体、正宋体的显示是做不到的。原因:程序中这三种字体的显示是:“先定义一个数组,然后将要显示的汉字点阵信息,读进这个数组,最后对此数组中的汉字点阵信息进行加工处理来显示美术字”,而程序中60语句中有 LOCATE 1,10的

命令, 它的列为10, 而后面的点阵列变量Y却是从0~15循环, 这样, 上述三种字将被“清”字所代替。

4. 80语句, 利用了INKEY\$函数, 我认为它与70语句后的LOCATE 1, 1: PRINT “ ”的存在是多余的, 它只能浪费内存和运行时间, 而这两点却正是衡量程序好坏的标准。但单纯地去掉80语句, 也会影响程序的运行, 因为程序中还存在有“GOTO 80”, 所以随之程序也要作相应的变更。

5. 在110~130三条语句中, 原文将判断ZT是否等于4和判断是否等于5这两种特殊情况放在了循环体中, 程序每循环一次, 也就要多做两次不必要的判断, 因此, 将判断ZT是否等于4和5移交循环体外, 会提高运行速度, 也是对程序优化了一步。

另外笔者在180, 240两条语句中增加了转子程序, 这样, 处理后的字将在屏幕上一直保留, 直到按下B键为止。

以上是我对此程序的一点小小的见解, 仅起

抛砖引玉的作用, 如有谬误, 万望读者批评指正。

修改后的程序打印件附后

```
10 REM 这是显示美术字的程序 88.7.28 李辉宏 89.8.20
20 SCREEN 2: KEY OFF: DIM H(24,24,18): CLS
30 INPUT "请输入要显示的汉字: ", P$
40 IF LEN(P$)>12 THEN P$= LEFT$(P$,12)
50 IF LEN(P$)/2 < INT(LEN(P$)/2) THEN L=LEN(P$)+1 ELSE L=LEN(P$): CLS
60 FOR K=1 TO L/2: LOCATE 1,1: PRINT MID$(P$,K*2-1,2): FOR X=0 TO 15
70 FOR Y=0 TO 15: H(X,Y,K)= POINT(X,Y): NEXT Y,X,K
80 SCREEN 2: CLS: LOCATE 10,1
100 INPUT "请选择显示方式: 1-大立体 2-斜宋体 3-正宋体 4-大虚体 5-结束", ZT: CLS
110 IF ZT=5 THEN END ELSE IF ZT=4 THEN 190
120 FOR K=1 TO L/2: FOR X=0 TO 15: FOR Y=0 TO 15: IF H(X,Y,K)=0 THEN 180
130 IF ZT=2 THEN 160 ELSE IF ZT=3 THEN 170
140 A=K*108-118: X=6: B=30: Y=5: CIRCLE(A,B),3: PAINT(A,B)
150 A=K*108-108: X=6: B=28: Y=5: CIRCLE(A,B),3: GOTO 180
160 A=(K+1)*50+15: X=3: Y=B=100: Y=1: LINE(A-1,B)-(A,B),1,B: GOTO 180
170 A=(K+1)*50+15: X=3: B=100: Y=1: LINE(A-1,B)-(A,B),1,B
180 NEXT Y,X,K: GOSUB 250: GOTO 80
190 SCREEN 1: CLS: PRINT P$
200 DEF FNC=C*2+30: DEF FND=50+D*2
210 FOR C=0 TO 130: FOR D=0 TO 16
220 DOT = POINT(C,D): IF DOT=0 THEN 240
230 PSET(FNC,FND),5
240 NEXT D,C: GOSUB 250: GOTO 80
250 LMS= INKEY$: IF LMS="B" OR LMS="b" THEN RETURN ELSE GOTO 250
```

## 丰富3070的打印机实线打印

重庆建工学院计算中心 魏 旭

我们知道3070打印机可以有8种汉字字型, 其中A—D字型用的是24×24字库, 而E—H字型用的是16×16字库。有些文章曾介绍过改变3.COM使3070打印机打出实线, 这是针对使用24×24字库情况下的。在一些情况下, 需要用16×16的字库打印表格。如象打印超宽度表格时, 就需要用16×16字库。在用24×24字库打印实线时, 打印头每次走纸24针, 在用16×16字库打印头每次走纸也是24针这样就空了几针, 所以打印出的表格为虚线。要使打印机打印实线就要修改3.COM程序来改变行距, 即打印头每次走纸不能走满24针, 只能走纸16针。这样打印出来的表格为实线。这样的修改只能是动态的修改, 当使用16×16字库时就走16针当使用24×24字库时就走24针。

改变3.COM程序的步骤为

C) DEUG 3.COM

用DEBUG的U命令找到判断几个字型的程序入口为CS: 88B0, 当为A—D字型时改变

JMP 8952为JMP 9110, 当为E—H字型时改变JMP 8952为JMP 9130

找到发ESC W, N1, N2, N3, N4命令的入口地址为CS: 8E07

—R CX

—CX: 9100

—A 9110

1E6E: 9110 PUSH SI

1E6E: 9114 MOV BYTE PTR[SI],

36

1E6E: 9117 POP SI

## SAP6软件顾问系统概述

四川省电子计算机应用研究中心 林丕 任海

### 一、引言

几乎在所有的科学研究和工程技术领域中,有限元软件(以有限元法和矩阵结构分析为主体的科学研究或工程结构分析应用软件)的应用已转化为社会生产力,使之可以迅速地解决各种各样的工程实际问题,并发挥了巨大的经济效益。目前,世界上拥有的300多个有限元软件中,具有前、后处理能力并和专业范围内的CAD系统融合在一起的仅仅只有几个,SAP6软件便是其中之一。同时,应注意到:尽管开发了一些有效的前后处理系统,但是要掌握这些软件的使用方法,并正确无误地决定分析策略和对计算结果做出正确判断,要求用户有工程设计经验和比较扎实的力学、数学知识,仅仅要求用户阅读的软件资料就有几百页,多则上千页。要读懂掌握它们是一件相当头痛的事。随着人工智能和专家系统方法的出现,完善和丰富了有限元软件在其应用与软件技术开发的前景:

1. 有限元分析、设计与CAD软件
2. 有限元分析、设计与CAD+顾问系统
3. 有限元分析、设计与CAD+完善的专家系统

本文着重论述了有限元软件SAP6顾问系统的体系结构和构造及在M-240D机上的研制。在此我们主要向读者提供一个了解有限元软件在其应用与软件技术发展中的第二阶段轮廓,而有意忽略了许多具体程序问题。

### 二、顾问系统

顾问系统是一个具有积累经验和认识能力且分解使用极为复杂的专用软件系统。系统结构主要由数据库和推理机组成,如图1所示。

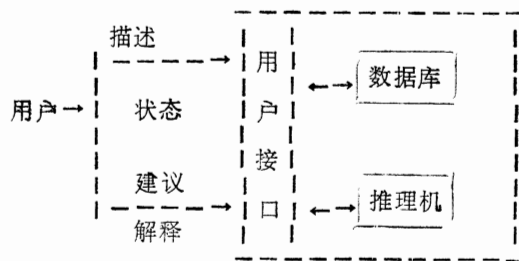


图1 《顾问系统结构》

该系统的核心内容就是将现有的用户接口和分析对策等有关知识编辑在一起,形成一个数据库,通过人机交互式有选择地进行咨询和编辑输出。这种数据库不是将现有的用户接口简单地堆

```
1E6E: 9118 JMP 8925
```

把CX寄存器的内容变为9100用A命令加一段程序,如果是A—D字型转该程序运行。

在CS: 9130处用A命令加一段程序为

```
-A
1E6E: 9130 PUSH SI
1E6E: 9131 MOV SI, 8E01
1E6E: 9134 BYTE PTR [SI], 31
1E6E: 9137 POP SI
1E6E: 9138 JMP 8925
```

这两段程序分别改变ESC W命令后面的参

数,当前面判断字型为A—D时转CS: 9110执行,当判断为E—H字型时就转CS: 9130处执行。

```
-A 8FEA
POP DX
MOV DX, 0930
MOV AX, 3100
INT 21
-W
```

这段程序使加到后面的两小段程序驻留在内存。



集在一起,而是以前提——问题——答案的形式编辑在一起;其分析决策则是由问题——结论形式的一批定则组成。所形成的数据库是有关有限元软件的知识 and 经验的总结,如图2所示。推理机具备推理的能力,它能理解问题并把知识和问题相互对应起来,进行分析决策,最后解决问题。

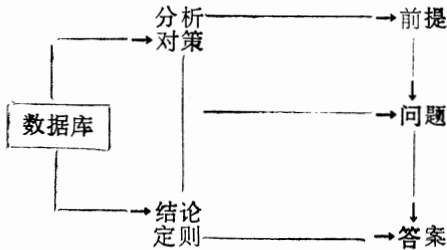


图2 《顾问系统数据库构成》

### 三、SAP6顾问系统体系

SAP6程序是一个大型、通过于线弹性结构的有限元静、动力分析程序。该程序在机械、航空、土建、地质、造船、水利、动力、核能等方面有着广泛的实用价值。同时, SAP6程序有与自己配套的前处理程序 MODEL, 后处理程序 POST及热传导程序 TAP6。这四个程序即可分别独立的使用, 也可配合起来组成一个系统。实为我国目前规模最大、功能最强、效率最高的线弹性结构分析系统。为此, 我们根据各个程序的功能和特点, 采用了三级顾问配套体系, 如图3所示。

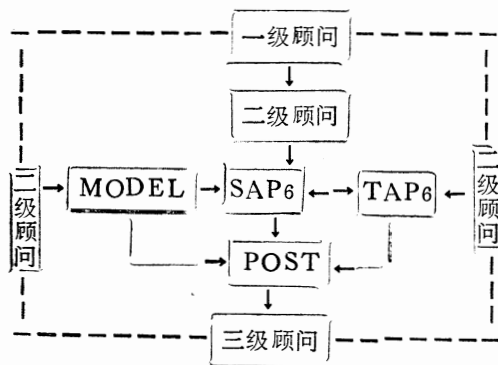


图3 《SAP6程序系统顾问体系》

#### 1. 一级顾问

将现有的SAP6程序系统(包括前处理 MODEL程序、后处理POST程序、热分析 TAP6程序)的用户接口, 以前提——问题——答案的形式编辑在一起。并由前提——结论形式组成的一批有关SAP6程序、MODEL程序、TAP6程序所能应用的范围及分析问题的类型、规模和采用的求解方式、手段的定则综合在一起, 从而构成一个数据库。当进行咨询时, 通过人机对话的方式, 将已建立起的数据库和用户提出问题的性质, 分析目的, 求解规模及其它要求结合在一起, 形成一个推理机构, 以权威性的知识与用户对话和与“数据库”对话, 进行分析决策。同时拥有描述、解释、分析、举例等辅助手段, 以帮助用户进行基本的分析判断和选择决策, 来保证对工程结构设计的合理性和计算的准确性。

#### 2. 二级顾问

对 MODEL、SAP6、TAP6 程序分别建立二级顾问系统, 以帮助用户在选择用某一个程序进行结构分析及建立有限元模型时, 所需要建立和定义的结构、材料、边界约束, 载荷条件及有关数据。该体系通过提问和用户的回答, 要求用户根据自己的问题提供数据, 逐步定义一个结构的有限元网格图形, 材料特性, 几何特征, 受荷条件及边界约束等, 从而帮助用户建立起有限元分析模型以及生成程序求解时所需要的数据。这种提问和答复的方式由浅入深, 由简到繁, 逐步深入, 每次提出的问题对用户来说都十分简单, 故不使用户感到难答。例如, 我们对SAP6程序的49组“数据输入指令组”按其每一组的功能和内容分成5类, 分别形成5个与一级顾问有关联的子数据库, 即结构形状、材料特性、几何特征、载荷条件、边界约束和若干个描述子系统, 即用户接口处理系统。使其能够完全准确地定义所有的问题和表达用户的要求, 并承担用户接口的加工、解释, 即把用户的问题说明书翻译成内部数据文件, 同时还要形成用户问题特征和要求的集合。顾问系统将根据这些数据和用户的要求, 逐步地进行选择决策, 直到完美地完成用户本次分析的任务和要求达到的目的。

# 单回路智能仪表硬件系统设计

吉林省计算机技术研究所 阎立恒 刘晓光 孟刚

## 一、前言

Dzy-1型单回路智能仪表,属于现场工作的数据采集和控制系统,大多是实时控制系统,就是对外界变化现场实时地进行监视、管理和控制。对于不同的技术指标,硬件和软件组成各不相同,但它们有共同的核心组件,依此基础作调整,可应用于很多场合。能于各种传感器配套使用如:温度、压力、流量、液位、PH值等毫伏和毫安信号直接连接。经过放大器,把现场采集来的模拟信号进入单片机,经过编程、变换、运算...等,调节控制输出0—10毫安模拟量信号;还可以输入、输出开关量,与标准执行机构配套使用,进行实现对各种生产过程的监视、检测和自动控制。

**二、智能仪表系统设计过程中,实用和可靠性的考虑贯穿于每一个环节,采用的方法有:**

1. 总体设计时有对智能仪表核心部件及外电路自动检测功能,当智能仪表运转正常时,定时的对系统各模块进行自动诊断。对外界出现的异常情况作出应变处理,若出现的情况无法解

决,应及时报警。当智能仪表系统出现故障时,报警系统进行自动报警。

2. Dzy-1型智能仪表,具有MCS-51单片机通讯功能,把一个控制系统划分为多个相对独立的子系统,每一个子系统由一个智能仪表检测和控制,各个子系统智能仪表中8031的串行口和主机通讯,接受主机的监督管理,是分散控制集中理用的一种下位机。

### 3. 减少系统的故障率

①提高电源的质量,电源干扰是系统不能稳定的主要原因,用防止干扰电源,采用屏蔽、隔离的电源变压器。稳压电源电路采取抗电网干扰和电网电压不稳定的措施,备有完善电源的高低频滤波电路。

②防止电磁干扰,通常对Dzy-1型智能仪表的执行机构及信号传输线采用屏蔽线,对于开关量的输入输出信号采用光电隔离措施。并合理地处理系统地、数字地、模拟地和大地之间的关系。

③合理选择元器件,设计时对元器件的负载,工作环境、速度、功耗等参数都有一定的余

### 3. 三级顾问

三级顾问是直接建立在后处理程序POST上的。主要帮助用户在一、二级顾问的基础上,对其所分析的工程结构的建模及分析后的结果,采用人机对话的方式及图形输出手段,进行检查、判断,以验证工程结构设计的合理性和计算的准确性。

SAP6顾问系统各部分之间的关系说明和问题求解过程如图4所示:

### 参考资料:

崔俊芝:《有限元软件述评》

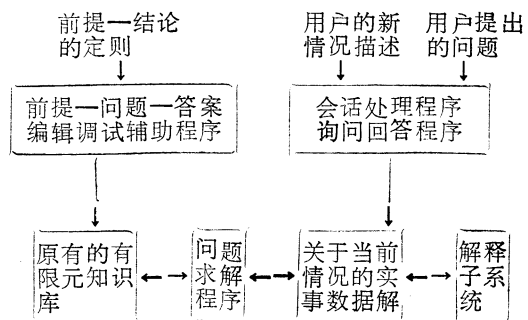


图4 《SAP6顾问系统各部分之间的关系说明和问题求解过程》

量,对元器件进行筛选。

#### 4. 操作和维修方便

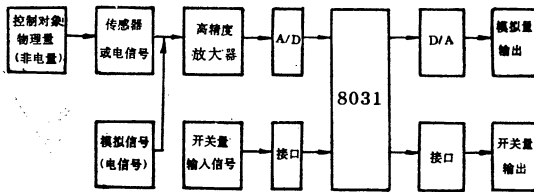
在Dzy-1型智能仪表硬件和软件设计时,考虑了操作和维修方便,尽量降低对操作和维修人员专业知识要求,符合实际便于推广。

#### 5. 性能/价格比高

智能仪表能否被广泛应用,关键在于有很高的性能价格比。采用先进技术,硬件软化是提高性能价格比的基本方法,设计时满足性能指标的前提下尽可能减少硬件成本,能用软件实现的功能均用软件实现。

### 三、单回路智能仪表系统的构成

Dzy-1型智能仪表系统组成如图一所示:



图一. 智能仪表组成框图

### 四、主要性能指标及功能

单片微处理器8031: 内部包含一个8位微处理器, 128个字节RAM, 21个特殊功能寄存器, 4个8位并行口, 一个全双工串行口, 二个十六位定时器。

主频: 6MZH

模拟量输入: 8路, 输入为毫伏信号。

模拟量输出: 1路输出0—10毫安。

开关量输入: 8路

开关量输出: 8路

串行通讯口: 1路

主要芯片: A/D, D/A, RAM2K, 印ROM8K

功能: 六位数码管显示, 调节和控制, 输出精度为5%, 人机对话及自动报警。

### 五、智能仪表系统的配置选择

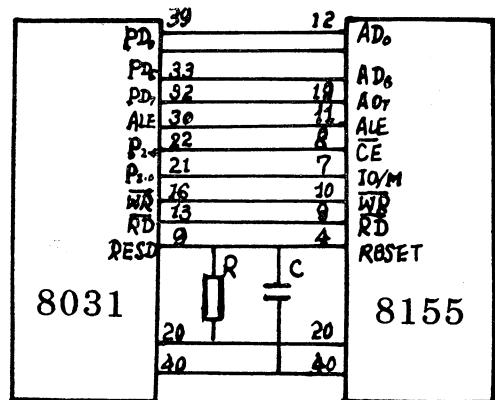
1. 存贮器: 8031内部没有程序存贮器ROM, 对于一般工业仪表系统, 初始化程序所需最大空间约1K字节, 监控程序所需最大空间约2K字节, 主程序所需最大空间为3K字节, 中

断程序所需最大空间约为2K字节。所以外接8K EPROM2764作为程序存贮器, 是可以满足要求的。

#### 2. 8155为接口扩展器

单片8155包含有256个字节的RAM存贮器, 2个可编程的8位I/O口(口A和口B), 一个6位输入/输出口(口C), 一个14位减法计数器。8155功能强, 接口方便, 适用于单片机。所以选用8155作为数码显示和按键的接口电路。

8031和8155接口电路如图二所示。

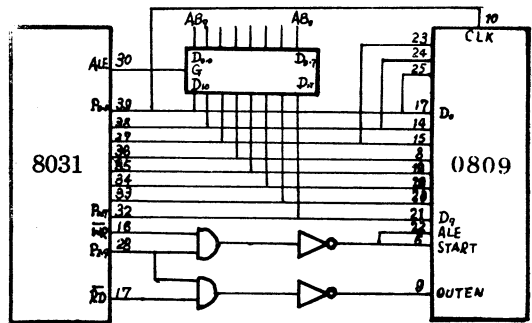


图二. 8155和8031的接口逻辑

3. 8255可以和8031直接接口, 接口逻辑同8155和8031接口逻辑方法相同。可作为开关量输入和输出口, 也可连接打印机。

#### 4. 0809和8031的接口方法

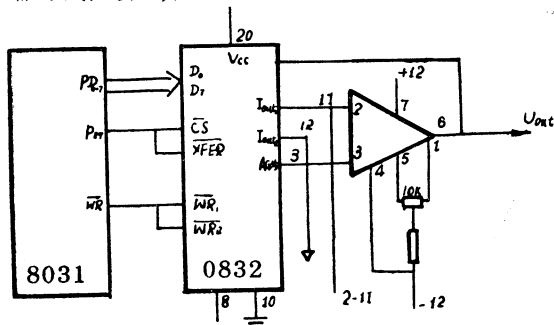
0809是8路8位的A/D转换器, 0809和8031可以直接接口, 其中一种直接接口逻辑图如图三所示:



图三. 0809和8031接口逻辑

5. 0832是八位数模转换电路DAC。

0832由八位输入锁存器, 八位DAC寄存器, 以及八位的D/A转换电路所构成。Dzy-1型智能仪系统中只有一路模拟量输出, 则采用单缓冲器方式, 接法如图四所示:



图四. 0832和8031接口逻辑

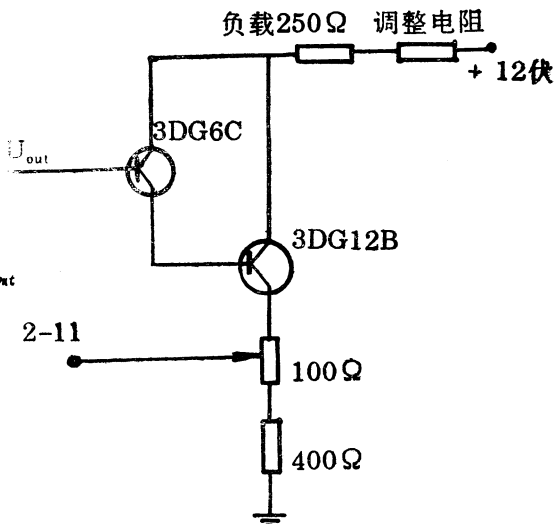
此电路输出为0—5伏直流电压, 为了与各类执行机构相配合使用, 采用如下电路使输出信号转换成电流信号, 即 0—10毫安或 4—20毫安, 其电路图如图五所示:

#### 6. 高精度放大器

传感器输出的大部分是毫伏信号, 不能和A/D转换电路相匹配, 只有将微小的毫伏转换成优级的信号, 才能将采集的信息输入给智能仪表, 我们采用ZF系列放大为1000倍的高精度放大器, 性能可靠, 漂移较小, 工作稳定可靠。

### 六、结论

经过现场应用, 单回路智能仪表系统运行是稳定可靠的, 控制思想的实现方便灵活, 技术上先进。



图五. 电压与电流转换电路

单回路智能仪表, 采用MCS-51系列8031单片机为核心新兴计算技术, 具有体积小、成本低、检测和控制功能强, 存储容量大(8K)、运行稳定、抗干扰能力强以及便于应用的特点。可以按着各个生产部门的需要如: 节能、节约资源、提高生产效率, 改善劳动条件和保护环境、提高自动化水平、开发研制出来的最新智能仪表。它具有丰富的运算、检测、控制及通讯功能。可用于单回路到数百个回路的控制, 系统构成方便灵活, 能满足新建、扩建和技术改造的需要, 是使生产现代化, 提高自动化水平的先进工具——单回路智能仪表, 是新开发研制成功的智能仪表。

## 1989年全国计算机应用研究学术交流会在蓉城举办

为了探讨计算机软件维护及硬件维修在计算机推广应用中的战略地位; 交流软件维护、硬件维修的实践经验; 展望软件维护、硬件维修的发展前景”。《计算机应用研究》杂志编辑部于1989年11月6日在成都四川省电子计算机应用研究中心举办了89年度全国计算机应用研究学术交流会。

参加此届学术交流会的246名代表及论文作者, 遍及全国24个省市自治区, 近200个大专院校、部队、科研机构和企事业单位。

由于此届学术交流会自始至终贯彻了“探讨学术, 交流经验, 沟通信息, 增进友谊, 切磋技艺, 取长补短, 提高效益, 振兴我国信息产业”的宗旨, 使会议开成了一个圆满成功盛会。

本刊编辑部

# APPLE IIe 微机体感诱发电位检测系统硬件接口设计

第三军医大学野战外科研究所

彭利安 刘连生 陈恒胜 谭正中 杨诗球

人体脊髓、皮质体感诱发电位(SPEP、CS-EP), 能够比较直接而客观地反映中枢神经、外周神经的功能状态, 并对神经系统损伤病变的程度部位作出判断。目前国内有条件起步研究或初步应用的单位, 多数是从国外引进设备, 研究与应用起来, 也难以得心应手, 均感难度大, 再因价格昂贵, 一般用户不敢问津, 为能适应国情, 普及基层, 一般能力和条件的医院科室买得起、用得上, 我们为APPLE IIe 微机设计了简单、经济、实用的硬件接口, 构成了一个功能较全的体感诱发电位检测系统(见图1), 用于临床监护和早期诊断, 收到了良好效果。

之。

3. 缩短信号传递环节, 少用电子线路, 多用软件程序代替实现, 设计能够新旧交换的公用接口, 尽可能提高接口的互换性和兼容性。

4. 设计多种方案, 实践中权衡比较, 选择造价低、方法简单、技术新颖的硬件接口, 反复进行调试和考验, 充分保证系统的稳定性和可靠性。

5. 采用组合式结构, 可分可合。分开各自为办公室自动化设备, 医学统计分析的工具, 多处可用的前置放大器, 信号发生器等, 合拢则为医疗诊断仪器, 多机一体, 一体多机, 多机多用, 各取所需。

## 二、接口框图

## 三、工作原理

刺激器(信号源)产生输出两路信号: 一路经过电平转换进入输入输出接口PIO, 作为系统同步起始信号, 命令全机同步工作; 一路经过隔离放大, 通过电报刺激生物体。记录电报接收到SPEP, 先由前置放大, 再用二次放大, 同时抑制伪迹, 提取信号, 先做硬件处理。ABC将SPEP转换成数字信号, 送主机分析处理, 所得数据结果、文字图形, 分别由显示器、打印机、驱动器各接口卡输出, 显示, 打印或存盘若需模拟输出, 可加D/A变换。

## 四、几点说明

1. SPEP信号甚微, 一级前置放大仍不理想, 还需根据调试结果进行二次放大。随着放大倍数逐级增加, 伪迹噪声随之抬高, 超出有用信号几百万倍, 直接送入ADC, 势将烧坏后级芯片, 必须采取合理的抑制和保护措施。

2. 曾有感兴趣作者, 在引进设备上试作SPEP, 均感难度大, 本机硬件调试的实践证

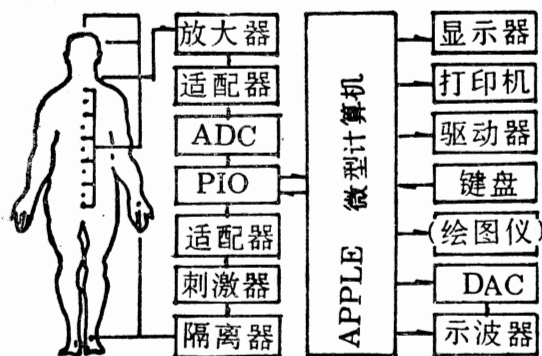


图1. SPEP检查硬件框图

## 一、设计思想

1. 选择普及面广, 价格便宜的APPLE IIe为主机; 以利部分有机单位能够原机共用, 旧机新用, 充分发挥其效益, 无机单位买得起, 有用处, 有利普及SEPP检查和微机在临床中的应用。

2. 充分利用用户原有的常用电生理仪器(刺激器、放大器等)作为主机的辅助装置, 新老结合, 物尽其用, 减低成本, 同时另行设计体积小, 价格低的更新部件, 供用户选择或取而代之。

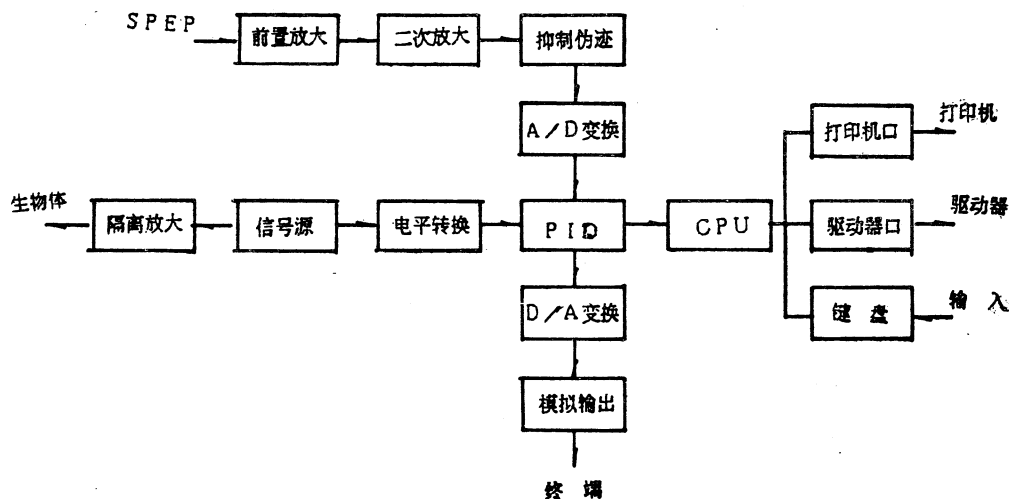


图2. 硬件接口框图

明, 电路基本放大倍数过大, 全是噪声, 过小, 信号仍在淹没之中, 提示了调试和设置合适的放大倍数是提高全机性能的重要保证。

3. 屏幕一次性地自动展示每次的物理波形, 还可打印机硬拷贝, 为节约成本出发, 可以省去DAC部分。

4. 点阵打印机为较理想的输出终端, 绘图

仪可根据用户的条件、兴趣和需要而代替。

5. 同步方式既可外部“通知”内部, 也可主机“指挥”全部, 前者比后者简单。

6. 在实践经验尚不丰富的情况下, 为有助于每次检测成功, 可予先试测皮肤阻抗“有效值”, 或用示波器观察前置放大器, 发现问题, 及时纠正, 置疑时反复试行。

## 一种实时中断键盘接口电路

南京3801信箱423分箱

何杰

**摘要:** 本电路介绍的是一种实时中断键盘接口, 它以大规模集成电路8279为核心, 采用中断的方式, 键数最多能扩充到64个键, 它可以与所有采用多总线结构的单板机相连接, 具有二键封锁或N键巡回扫描的功能, 可自行消除键盘输入的抖动。

### 一、8279简介:

#### 1. 封装形式

8279是40脚双列直插式器件, 它的引脚图排列如图一所示:

#### 2. 工作原理

##### a. I/O控制和数据缓冲器

I/O控制部分用 $\overline{CS}$ 、 $A_0$ 、 $\overline{RD}$ 和 $\overline{WR}$ 四条线控制数据流进出内部的各个寄存器和缓冲器。进出8279的所有数据流都由 $\overline{CS}$ 开启。CPU给出或所

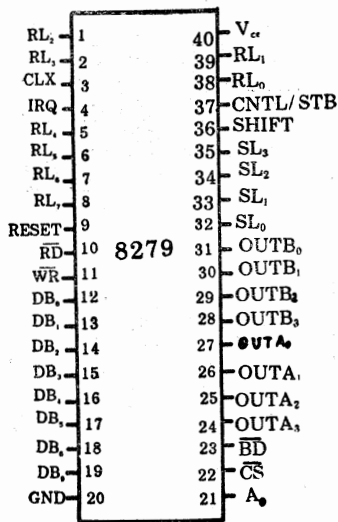
需的信息特征由 $A_0$ 表示,  $A_0$ 为逻辑1表示该信息是命令或状态;  $A_0$ 为逻辑0表示该信息是数据。

$\overline{RD}$ 和 $\overline{WR}$ 决定通过数据缓冲器的数据流的方向, 数据缓冲器是双向的, 连接外部总线和内部总线。该器件未被选中( $\overline{CS}=1$ )时, 就处于高阻状态。在 $\overline{WR} \cdot \overline{CS}$ 时, 对8279进行写操作, 在 $\overline{RD} \cdot \overline{CS}$ 时, 对8279进行读操作。

##### b. 控制和时序寄存器以及时序控制

这些寄存器存放由CPU编程的键盘工作方





图一. 8279的引脚图

式和工作状态信息。工作方式的设定过程是在  $A_0 = 1$  时将所需的命令放到数据总线上，然后发出一个  $\overline{WR}$  信号，在  $\overline{WR}$  的上升沿命令被锁存起来。然后由 8279 对命令进行译码，并执行相应的功能。时序控制部分包括基本的定时计数器链，第一个计数器是一个 N 分频预定标器，可通过编程使内部时序与 CPU 的周期相匹配。其它的计数器对内部基频进行分频，提供键扫描，行扫描时间。

## 二、硬件设计

根据工作的需要，我们设计的这种实时中断键盘是配在专用设备上使用的，所使用的单板机为 86/30，因此下面的叙述均与 86/30 单板机有关。众所周知，86/30 单板机是功能很强的十六位单板机，它采用多总线的结构形式，CPU 为 8086，主时钟可选 5MC 或 8MC，数据总线为 16 位，可进行字或字节操作，地址总线为 20 位，最大寻址空间为 1 兆字节，有多重 8 级中断请求和中断允许信号，读写分为存储器读写和 I/O 读写两种。

我们在设计这种键盘的时候，充分利用了多总线所提供的各种信息，把键盘设计为具有实时处理能力。也就是说，一旦有键按下，8279 就向 CPU 申请中断，中断允许以后，CPU 暂定原来的工作。响应中断处理，处理完成以后 CPU 继

续原来的工作。这样可以节约 CPU 的处理时间。

图二为键盘接口原理图。以图中我们可以看到 8279 的状态口和数据口的口地址分别 FFEAH 和 FFE8H，8279 的八根数据线通过双向总线收发器 8287 与多总线的低八低数据总线相连接，在允许输出的情况下，当  $T=1$  时，数据从 8279 传送到单板机，当  $T=0$  时，数据由多总线传送到 8279，即对 8279 进行初始化。8279 的时钟由多总线提供的时钟经两次分频后得到。8279 的中断请求线经反相以后直接连到单板机八级中数的  $\overline{INT}_0$  端，即键盘中断具有最高优先级。8279 的读写信号分别连接多总线的 I/O 读写信号，8279 的  $A_0$  接至多总线的  $A_1$  端。

我们设计的键盘为专用小键盘形式，它包括 0~9, A~F 十六个数字键，. , , , 和六个命令键。因此我们使用 8279 的  $SL_0$ 、 $SL_1$  和  $SL_2$  三根扫描线经译码后与  $RL_0 \sim RL_7$  八根回扫线相组合形成键盘的接线，见图二。键码的编码形式如下：

$RL_2$	$RL_1$	$RL_0$	$A_2$	$A_1$	$A_0$
0	0	0	x	x	x
0	0	1	x	x	x
0	1	0	x	x	x
0	1	1	x	x	x

其中  $A_2A_1A_0$  三位由  $RL_0 \sim RL_7$  八根回扫线决定。当  $A_2A_1A_0 = 000$  时，与  $RL_0RL_1RL_2$  组合后形成的键码为 00H, 08H, 10H, 18H 等；当  $A_2A_1A_0 = 001$  时，与  $RL_0RL_1RL_2$  组合，形成的键码为 01H, 09H, 11H, 19H 等，其余类推。

使用这种方法设计的键盘，键数最多可达 64 个，且能自动地实现二键封锁或 N 键巡回方式消除键的抖动，而无须另加去抖动电路，能很好地满足各种专用系统要求的操作简单的特点。

## 三、软件设计

在软件设计中，包括对 8259 可编程中断控制器的编程，可编程键盘控制器 8279 的编程和中断程序三个部分。

### 1. 中断控制器 8259 的编程

根据 86/30 单板提供的硬件资料，8259 中断控制器初始化命令字 ICW 和操作命令字 OCW 的口地



## IBM-PC 机并行口简易扩展器

山东纺织工学院微机室 韩知恩 赵学惠

**摘要:** 本文介绍一种微型计算机并行口的扩展电路, 利用此电路可以使一台微机在并行口上同时与打印机、绘图仪或其它并行设备连接, 大大方便了计算机用户。

利用计算机绘图, 是计算机应用的一个主要组成部分, 但在只有一个并行接口的微机系统中, 打印机和绘图仪只能有一种设备与主机连接, 这就给我们的工作带来了诸多的不便, 例如: 我们在使用 IBM PC/XT 机时, 若想绘图, 只有先把绘图仪连接到主机上, 用绘图仪绘图; 若想打印程序清单, 只好关掉电源, 把绘图仪从主机接口上拔掉, 然后再把打印机连接到主机上。这样频繁的拔插, 不仅工作效率太低, 而且对设备也有损伤。是不是还有更好的办法呢? 这里我们介绍一种十分简单的转换电路, 使用此电路, 若想选择使用打印机与绘图仪时, 就不必再拔插接口电缆, 只需轻轻拨动一只开关, 就可选中你所要使用的那个外设了。此扩展器的具体电

路如图。所示 (见下页)

我们仅以打印机和绘图仪为例, 来介绍此电路的基本原理。图中的 K 是由用户操纵的控制开关。A1、A2、C 和 B1、B2、C (均为 74LS367) 组成了打印机与主机和绘图仪与主机的电子开关。哪路电子开关导通, 取决于 K 的位置, 当 K 置于“1”时, 主机与打印机接通, 当 K 置于“2”时, 主机与绘图仪接通。由此原理可知, 欲增加外设的数目, 只需增加电子开关的路数和 K 的刀数。

此扩展器价格低廉, 制做简单, 使用方便, 既可发挥计算机的多功能作用, 又可提高您的工作效率, 请君不妨一试。

```

MOV DX, 0FFEBH
IN AL, DX
LEA BX, CS:DATA
MOV AH, 00
ADD BX, AX
MOV AL, CS:[BX]
MOV BUFFER, AL
MOV DX, 00C0H
MOV AL, 020H      ;OCW2
OUT DX, AL
POPF
POP DI
POP SI
POP BX
POP DX
POP CX
POP AX
POP DS
POP ES
STI
IRET
DATA
DB 30H, 31H, 32H, 33H, 34H, 35H, 36H, 37H
DB 38H, 39H, 53H, 2CH, 3AH, 52H, 57H, 0DH
DB 41H, 42H, 43H, 44H, 45H, 46H, 47H, 50H
DB ?
END
  
```

复位, 实现新的同步, 因此取代码为 D3H, 其初始化程序见清单。

### 3. 中断服务程序

对 8259 和 8279 初始化以后, 系统就可以接受键盘中断。中断服务程序的流程为: 关中断, 保护现场, 读键盘并把键码转换成相应的 ASCII 代码, 使 8259 复位, 恢复现场, 开中断。其服务程序见清单。

经反复试验, 该接口电路操作简单, 调试方便, 可靠性高。

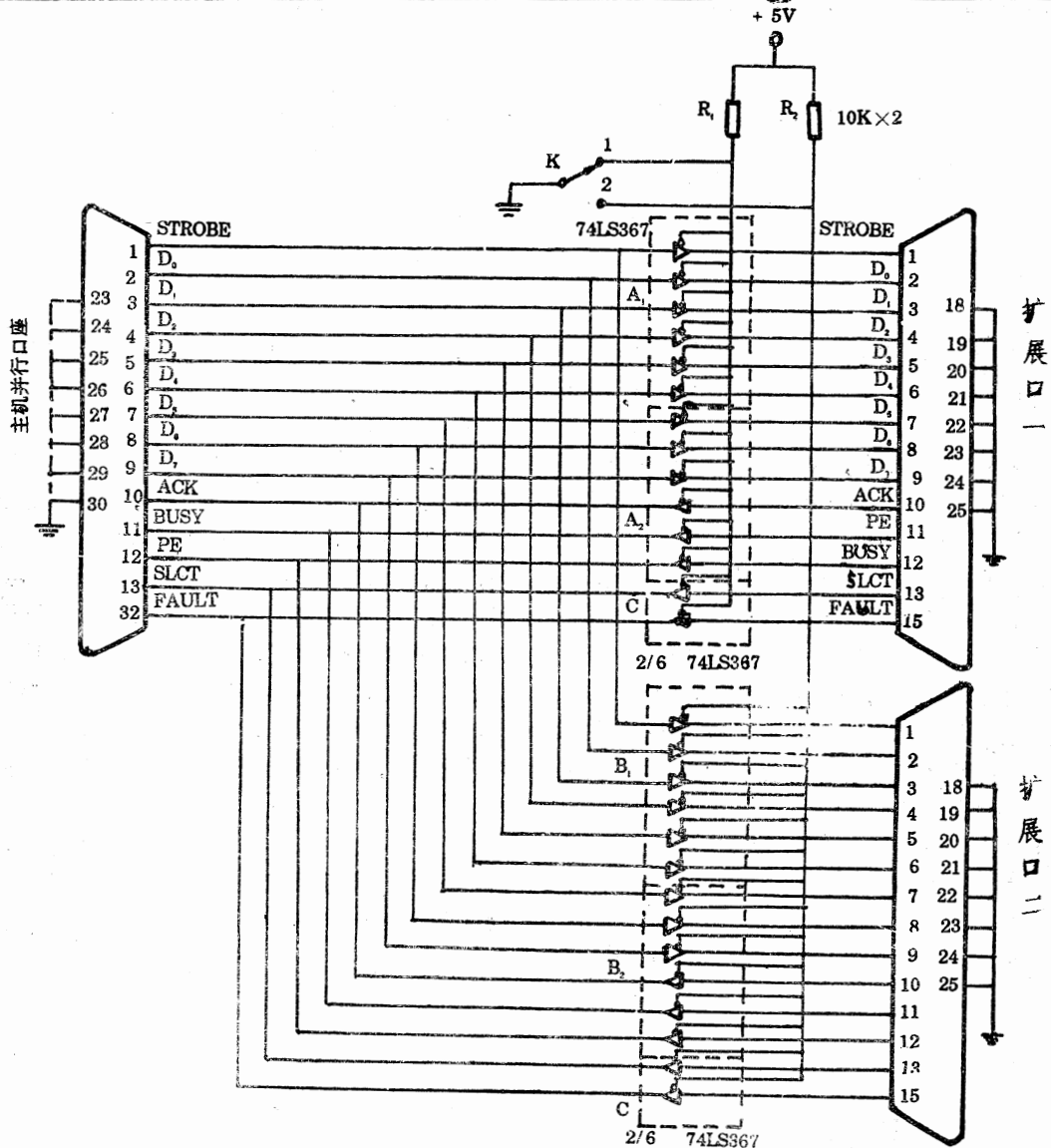


图 PC机并行口简易扩展器

## 好消息

四川省电子计算机应用研究中心引入国外技术开发的下列产品颇受青睐：

一、ICT—55A 智能通用数字集成电路测试仪——可检测 54/74 和 CMOS 4000 系列及其兼容的同类芯片以及部分静态 RAM，同时可自动分析上述芯片的型号、功能。批发价：3200 元，零售价：4300 元。

二、超级通用中华学习机频果机读写卡——可读写 2716~27512 芯片，Vpp 程控自动选择。批发价：400 元（10 块以上）单价：550 元。

三、超级通用 PC 机读写卡——可读写 2716~27512 芯片 Vpp 程控自动选择。批发：650 元（10 块以上）单价：850 元。

联系人：本刊读者服务部唐大利

本刊编辑部

# 计算器应用一例

甘肃省计算中心 华金保

## 一 引言

目前在各个领域的行业中都要用到计数器,如制瓶行业、纺织行业等等。在这些地方使用的计数器大部分是两类,机械式和数码管式的。尤其是在旧机器上使用的基本上都是机械式的。这两类计数器在使用中都有一定的缺陷。首先,机械式的计数器使用寿命较短,需经常更换。其次,数码管式的成本太高且需电源。一旦断电,计数值将丢失,造成一定的损失。为避免上述缺点,本文提供一种利用普通计算器来代替计数器工作的一种简单可行的办法。大家知道在计算器中做加法运算时,是指下列顺序按键:被加数、+、加数、=。在按了“=”号键后,就可在液晶显示屏上得到结果。而屏上原显示的加数消失。其实这时“加数”和“+”操作功能仍被计算器保存。如果我们再次按“=”号键,计算器就会把加数又一次累加到结果上。仅复按“=”号键就可连续累加。用计算器实现计数就是利用了计算器这一功能。

## 二 计数器实现

将普通计算器打开后盖,在“=”号键两端焊接出两根导线即可。这两根导线用来连接到传感器上,进行计数触发。计数器的传感器也很容易制做,在不同的场合可以采用不同的方式;有源的或无源的。如:微动开关、行程开关、干簧管、光耦合器等。只要连接好,计数器就可正确计数。具体步骤如下:

- 1: 接通计算器电源。
- 2: 预置起始初值。
- 3: 按“+”号键。

### 4: 置计数值步长。

通过上述四步,计数器就可以工作了。每当传感器发来一个信号,计数器就会在预置初值上累加计数值步长。如需重新设置,只要给计算器清零,重复上述四步即可。

由于当前使用的大部分计算器都有自动关闭电源的功能(一般是五分钟左右)。所以,在一些频率很低的计数场合,就会发生问题。在这种情况下可以采用定时触发计数器中不影响正常计数的键来解决。例如:可采用时基电路芯片555制做一定时电路,通过光耦合器连接到计算器“MC”键的两端,用来定时触发“MC”键,使计算器不到五分钟就有一次触发而不会自动关机。这个键对正常计数没有影响。

## 三 结束语

用计算器做计数器简单可行,成本低廉,可靠性高。一般普通计算器只有十多元,比数码管计数器的价钱要低得多。计算器的位数最少有八位。所以,其位数一般都能满足使用要求。在使用中用计算器做的计数器可以灵活应用。例如:在纺织机械上计数布长度的地方,每一信号布长是0.5米,我们只要在置计数值时把步长置成0.5米后,就可随时直接从计数器上读到已织布的实际长度,而无需再进行转换。这是一般计数器无法做到的。将第二步的“+”号键换成“-”号键,这样,计数器就变成了一个减法计数器。所以对这样一种简易可行的计数器在那些只要频率不太高(一般不要高于每秒三次)的场合都会大有用武之地。

## 外设常见故障与排除

地质矿产部北京计算中心 陈兰英

外部设备做为计算机系统的重要组成部份,它的正确使用与维护直接影响到能否使计算机系统充分发挥其效能。由于外设大部份为机电、光电、磁电设备。精密复杂,故障率一般高于主机。其故障可分为机械、电路部份、传感部件几类。而机械类故障往往又多于其它故障。现将我们在日常工作中对 WONG CO-11 九轨磁带机、2230 宽行打印机、M-200 卡片阅读机实际维护所遇到的一些常见故障及排除规纳如下,供有关同志参考。

### 一、九轨磁带机

#### ·按LOAD键后不能正常加载

①首先检查鼓风系统是否工作正常。若鼓风机是工作的,只是带不能正常进入真空积带箱,可调节通风软管上的气门和上环形带调节其在积带箱内的上下位置即可恢复正常。

②保险丝 $F_6$ 烧断或LOAD电阻 $R_{228}$ 也烧坏。查电源板电机绕组保护二极管 $CR_8$ ,一般因 $CR_8$ 击穿造成负载短路而致,更换元器件

③收带盘无负压输出,更换伺服板 $U_6$ 元件。

④进入真空积带箱入口处的导轮卡子损坏所致。

#### ·加电后送带轮倒转

若按RESET和RETURN开关后能使送带轮停住。原因是反绕开关问题(未按下时,接地不好所致),更换反绕开关。

#### ·带不能加载、送带轮不转

调节气门不起作用。打开光电传感暗盒查5V灯泡,一般是灯泡坏所致。

·面板加电后,鼓风马达就运转(未按LOAD键应不转)。一般是LOAD与REWIND开关工作状态不对所致,需更换这两个开关。

·不能加载、鼓风系统不工作

①鼓风马达轴承损坏引起,需更换解决

②能直观看到送带盘和鼓风马达前的风扇转动,同时能听到继电器工作声和嗡嗡响声,但马达不转。检查鼓风马达电源如正常、拆掉通风软管及电源接线片等,取出鼓风机总成。如查电机2组线圈均是好的,查 $K_6$ 部件,一般是该器件故障,更换即可。

③LOAD开关内部弹簧故障所致,须修复解决。

#### ·加载与反绕后带均不停在BOT位置

用示波器监视TB7测点,调节伺服板 $R_{181}$ ,使其在BOT处为12V,不在BOT位置为0V $\pm 0.1V$

#### ·读带到末尾不认EOT标记

原因是EOT/BOT检测器的EOT识别电路问题,或与其相联的信号线断。

#### ·整盘带将要读完时,磁带机自己卸载

原因是带在积带箱中的位置不合适,须调整伺服板电位器 $R_{100}$ 和 $R_{153}$ 。

### 二宽行打印机

#### ·打印机不能联机(红灯亮)

一般有三种故障可能性①电源故障②字鼓不转③电机转速不对。如电源部份正常,首先可查带动字鼓转动的电机部份,若启动电容是好的,但电机发烫或手动转动电机轴不自如,一般情况是电机轴承故障所致。

#### ·走纸故障

如带动走纸的轴杆倒转,使纸慢慢往回卷,堆积在字锤组上方。原因①码盘位置处的插头接触问题②码盘盒内的红外二极管板问题或其接触问题。

#### ·色带故障不能联机

色带故障灯始终亮着不能被清除、色带马达不工作。查色带驱动伺服部份 $Q_{303}-Q_{306}$ ,



U<sub>303</sub>—U<sub>305</sub>，一般是这部份的问题。

- 最后一行打印杂乱无章字符且显示纸故障。

原因是有一根钢丝绳应压在纸上面，常在换纸时被忽略而放到纸下面去了。

- 色带反行故障

色带正走时正常，当反走时就卡在那不动了。同时可听到一声摩擦打滑的响声。原因是带动色带的上方齿轮坏了。需将其外罩卡黄片等取掉更换新的齿轮。一般这种齿轮应多备份些。

### 三、卡片阅读机

- 无5V电源

多为电源部分内的 MC 1469R 件烧坏，更换器件。

- 加电正常但不能做读卡动作

控制吸卡和移动卡片进入读站的继电器上面的三层金属连接部件层与层之间脱开，不能一致动作所致。需更换此部件。

- 过载保护、不能读卡

现象：capstone开关下落。查主动轮马达旁类似继电器作用的触点开关，长期工作打火，生成一层氧化层，致使开关不能起到良好

接通作用，用酒精清洁触点后即可。

- 不能连机、不能读卡

加电后绿色高位灯始终亮（无卡片时不应亮），这时放卡连机操作，卡片又读不进，TTY上打出“脱机”信息。一般原因：机内与前操作小面板相连的插头有接触故障，须打开上顶盖排除。

- 正确卡片被读错

原因①长期工作后，读站缝隙内积累灰尘太多所致。需打开外壳、吸去灰尘。②光源或光敏管外壳破碎或某个被烧坏，需更换读站部件。③光源与光敏管两方对应位置发生移动，需加以调整。

- 将空白卡误读为“全1”卡

原因：与接口相连的DIO电缆故障。

- 读卡动作不正规

读卡时出现一次连续读进两张，且被读进后PICK灯亮。再按RESET按钮后，又可进两张，现象又如同前面一样。排除：需多次调整读站上下左右位置，关键是左右位置。将最终正常位置用小刀刻上两条痕迹，以便将来再调整时参照。

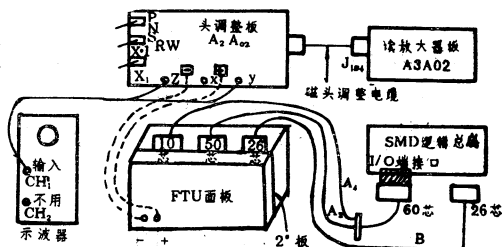
## 使用FTU调整磁头的方法

地质矿产部北京计算中心 陈兰英

FTU 是手提式现场测试设备 (FIELD TEST UNIT)，可对多种磁盘机驱动器进行各种脱机测试。使用 FTU 进行磁头调整的优点是整个调整过程中不占用主机，FTU 直接与驱动器相连，通过 FTU 选择驱动。进行磁头调整的方式有多种，这里介绍的是使用 FTU 调整 SMD 盘机磁头的方法。

### 一 检测与调整的准备

- ①关闭盘机，安装CE盘组，予热60分钟。
- ②连接FTU和驱动。参阅连接示意图。
- ③FTU面板PROCESSOR开关分别放在RESET、HALT、START、RUN位置，其余开关置相应处。



连接示意图

- ④加电。使-5V和+5V指示灯亮，并确保 ON CYL RDY WTPROT、SELD、SEEK-ERR FUT、BUSY RUN灯亮。

## 二 FTU面板有关操作

### 1. 选择驱动

将开关放 DEVICE TYPE, 置入设备类型码; 按 LOAD 键将显示数码传输进入第一排灯中, 同时也装进 FTU 内存; 按下 SELDRV 键, 若 SELLED 点亮, 且第一排灯呈现驱动逻辑地址, 说明选择成功, 否则为不成功。

### 2. 直接连续寻找

开关分别置于 ACCESSONLY、DIRECTSEEK 和 DESTINATION 位置, 置进柱地址数 (十进制) 后, 依次按 LOAD、GO、STOP 键。

将 ACCESS SELECT 开关放 DIRECTCUNT 位置, 置入第二个柱地址, 按 LOAD GO 键, 驱动开始执行测试。

按 CLR 和 RTE 键, FTU 则准备好做下个测试。

### 3. 直接寻找 (已完成驱动选择)

开关位置同前, 用十进制置入柱地址数, 按 LOAD、GO 键, 驱动开始动作。按 STOP 和 CLR 键可停止寻找。

### 4. 置磁头号

开关置 ALIGNMENT、HEAD、MAN 位置, 按 STOP 和 CLR 键。置入要调节的头号。依次按 LOAD、DEL 和 GO 键, 即可在第二排灯中以十六进制显示出 offset 值, 符号位不亮为正, 亮为负。

## 三 磁头检测与调整

### 1. 伺服头检测

①将磁头调整板开关置 S 和 X·1 位置。命令连续寻找 240—245 柱 30 秒。直接寻找 004 柱。

②将 P/N 开关分别置于 P 和 N 位置, 记录第

二排灯中读值, 计算并记载以便与调整后的值对比。例如:  $P = +20$

$$N = -15$$

$$(P) - (N) = (20) - (-15) = +35$$

③命令直接寻找 005 柱, 重复上述观察波形, 读 mv 值, 计算  $(P) - (N)$  过程, 其值均应在  $\pm 60\text{mv}$  内。

④将 004 和 005 柱分别计算的值相加, 总和应在  $\pm 75\text{mv}$  内。

⑤命令直接寻找 245 柱重复上述过程, 值应在  $\pm 60\text{mv}$  内, 以上各计算值若超出规定则需进行伺服头校准。

### 2. 读写头检测

①开关置 RW 和 X·1 位置。直接寻找 004 桶, 分别置入各磁头号, 观察波形, 将 P/N 开关分别置 P、N 记下读值、计算值。

②直接寻找 245 柱, 分别置入各头号, 记下读值、计算值。

③结果值如不在  $\pm 150\text{mv}$  内, 则需对读写头校准。

### 3. 磁头校准

选择 S/RW 开关位置、置头号、将磁头定位在 004 柱上, 松开磁头臂紧固螺钉, 用 4 磅力英寸扭矩紧固, 用调整工具使波形接近要求, 读值达最小 (理想值是接近零)。分别改变 P/N 位置, 连续调整, 使  $|P - N| < 60\text{mv}$ , 用 6 磅力—英寸扭矩旋紧对应螺钉。

将磁头定位 245 柱上, 重复上述步骤, 用 12 磅力—英寸扭矩上紧螺钉, 取出长臂调整工具。

连续寻找 240—245 柱 30 秒, 卸载、加载磁头两次后, 直接寻找 245 柱, 对所有磁头重新检测一遍。合格为止。

## 0520CH—II 计算机电源故障一例

甘肃省计算中心

华金保

**故障现象:** 开机后, 整个系统无反应, 系统不启动, 屏幕不显示, 喇叭不响, 磁盘驱动器不动。

**故障部件:** +5V 电压整流二极管击穿。

**故障分析:** 开机后如果出现上述故障现象, 一般来讲是机器电源部分的故障, 所以要首

先检查一下电源是否正常,有无直流电压输出。先检查外部,即220V交流电源是否正常,计算机电源插头是否插好,交流保险丝是否完好,在确定这些都正常之后,再打开机盖用万用表在主电路板上的+5V和+12V测试点上进行测试,看一看电压是否正常,如果没有电压,则是电源箱内有故障。可对电源箱进行检查修复。

我们知道现在微机中一般都是使用的开关型稳压电源。此类电源都有很强的自保护功能,4组电压+5V、+12V、-5V、-12V中只要+5V和+12V电压一旦出问题,就会使4组电压都无输出,从而实现自保护。笔者这次碰到的故障就是由于+5V电压出故障而引起4组电压都无输出。打开电源箱后,从表面看无任何烧焦痕迹,直流保

险管也是好的。用万用表测量各组电压的输出阻抗,发现+5V电压的输出阻抗很小,所以可以确定故障就出在这一组上。从此组电压输出端往前检查,断开高频整流滤波电感L4,再测+5V电压的输出阻抗,阻抗正常。则证明问题出在脉冲变压器+5V绕组上或整流二极管上。一般来讲,变压器不太容易损坏,所以要先检查整流二极管。测量二极管时要把一端从印制板上断开。0520CH—Ⅱ的开关型稳压电源的各组电压都采用图1的全波整流方式。由于+5V电压的负载最重,其输出电流很大2.3~15(A)之间。所以在+5V的整流中是采用了两个二极管并联在一起整流。如图2。

在这样一种情况下,如果并联的二极管其中

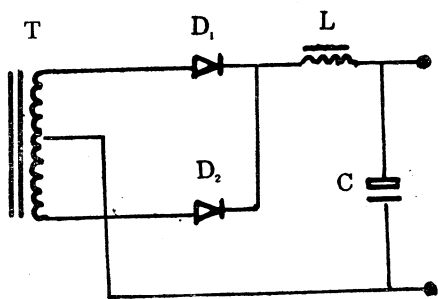


图1

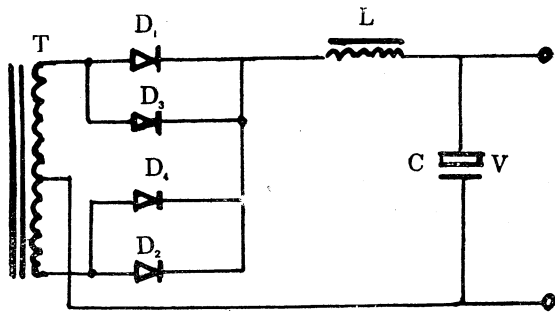


图2

之一性能不太好就很容易损坏。一旦损坏,+5V电压就不正常,整个稳压电源便不能工作。笔者这次遇到的情况就是其中一只二极管击穿短路造成。如果开路,那么另一二极管也会马上烧坏。

把坏的一只二极管拆下,在空载的情况下,通电试验,电源恢复正常,找一同型号二极管换上,加载试验,整机恢复正常。

## GW0520—CHⅡ 硬盘维修一例

黄河大学计算机系

叶志斌

### 1. 故障现象:

硬盘不能自举(自检通过后死机),软盘启动后可执行dir C:↵,但执行C盘文件时显示:

Seek error reading drive C:

Abort, Retry, Ignort?

(驱动器读定位错,中止、重来、忽略?)

### 2. 故障诊断:

运行诊断程序(参考diagnostic的使用说明)给出1704号错,此为硬盘0道错误代码。

### 3. 故障恢复:

#### a. 用一般的方法即重新格式化的方法

Format C: /S

显示Formating....., 此时听到较长时间的吱吱异常声响, 然后显示0道格式化错并中止Format的执行返回系统。

#### b. 用Lowform恢复。

Lowform是硬盘初始格式化程序, 进入后要回答有关硬盘的配置信息, 若应答有误, 则格式化后显示:

Invalid configuration information  
(无效配置信息)

Please run debug at system disk

-g=C800:5

此时, 从A盘进入 debug 后打 g=C800:

5. 程序运行时提示你重新进行硬盘配置。

硬盘配置根据其规格、容量不同而有不同类型, 在0~12的一项选择中CH II 机应选0。

Lowform运行成功后, 应依次运行Fdisk (硬盘划分程序, 用法参阅DOS手册) 和硬盘格式化程序Format C: /S对硬盘再次格式化。

以上两项工作完成后硬盘恢复成功。

## 3070打印机打印针磨损的处理方法

湖北省计委计算中心 周维寿

3070打印机长期使用后, 经常出现打印字迹模糊的现象, 有时甚至看不见打印字迹。这种故障多数是由于打印机长期使用后打印针磨损所致 (不包括断针)。用户如果重新更换打印针或打印头, 又很不经济。这里介绍二种简单实用的方法。

**方法一:** 卸松打印头固定螺钉, 用适量的力推动打印头靠近滚筒, 然后拧紧打印针固定螺钉。加电开机测试, 如果不符合要求, 则重复调整。这种方法的缺点是由于打印头离滚筒太近, 容易造成卡纸, 卡字车等现象。并且, 如果打印针磨损得特别严重, 这种方法还不能解决问题。

相比之下, 下面的方法更为可靠。

**方法二:** 卸下打印头固定螺钉, 拔下打印头电缆, 取出打印头。卸下打印头制动板上的两个螺钉, 取下制动板和润滑橡胶圈, 在洁净的薄纸上剪下直径比橡胶圈小0.5mm的圆, 把此圆纸片放入润滑密封圈里, 整平。再套上橡胶圈, 固定制动板, 装上打印头, 正确连接打印头电缆。加电开机测试, 不合要求的话, 则按上述方法再加入一圆纸片, 重复测试, 直至达到要求。

这两种方法都是减短打印针出针路径, 从而达到抵消打印针磨损的量。方法二更为实用, 适应性更强。请君不妨一试。

## 3070打印机断针的简易修复

福建宁德师专计算机室 苏亚华

引起打印机断针的原因是多方面的。其中最常见的是由于打印针出口孔被油墨污垢堵塞, 使打印针击出后不能及时回缩, 从而被移动的色带拖断。被折断的针比正常的针要短一些, 因此打

印时不能击到纸上而在该针位置出现一行空白点。除此之外, 由于长期使用, 打印针头不断磨损 (尤其是使用频率特别高的针如位于制表符横线处的第十二针), 也会导致打印针长度不够而

缺点,近年来我们为不少用户解决了一些打印机的问题,在我们所遇到打印机故障中,上述情况占绝大多数。如果更换新的打印针或送厂家维修,不仅价格昂贵(一根打印针价格在100元左右),而且低碳高强度打印针不易购到,维修周期长且又麻烦。实际上,这类故障完全可以自行修复,方法十分简单,效果很好,不妨一试。

下面以3070打印机为例介绍修复方法,这个方法也适用于打印头结构类似的其它型号的打印机。

3070打印机针的根部是用银将针与针体焊接起来的,一般在针的尾端部都留有一截多余的针长(约一毫米),而针尖被折断的长度大都不超过一毫米。因此我们的想法是:利用针尾部的余量,把针往前移动一小段距离,使针头达到需要的长度。具体的方法如下:

首先,卸下打印头,打开后盖。轻接打印针

根部,观察打印头前面,可以看到因针尖折断或磨损而变短的针。从后面用镊子小心抽出该针,然后将针根部的银焊点熔开,(注意要使用银焊枪。如果手头上没有,可以找市面上任何一家金银首饰加工店代熔一下,一般的金银店都有银焊枪。)紫金3070打印机的打印针根部还有一个小圆柱套在针上,也要熔开。焊点熔开后,将针向前移动一点,移动的距离视针尖被折断的长度而定,要比折断的部分略长一些。仍用银焊枪将针根部与针体焊牢。

最后,将焊好的打印针按原位置装入打印头,盖上后盖。由于修过的这根打印针比其它针略长一些,故还要用一块油石,浸入机油后取出,将打印头垂直在油石上来回轻轻地磨,直至该针尖与其它针尖平齐为止。

这样,针尖折断的打印针就修好了 使用起来同换了一根新针一样。

## 用注射法使PP40 打印机笔更新

安徽省冶金工业学校 沈友敏

LASER PP40打印机是一种功能很强的打印机,可以接在LASER系列机、Apple系列机以及Dragon系列机上。它具有价格便宜、结构小巧、四色打印、线条清晰等特点,既是一台打印机,又是一台精巧的绘图仪。但在实际使用过程中,由于四色绘图打印笔的笔胆容量较小,使用不久墨水使用完,必须更换新笔。所以四色绘图打印笔的使用寿命不长。如果这种打印笔能象钢笔一样,墨水用完后可以注入墨水再次使用,就无异于给打印机笔注入新的生命,极大地延长了它的使用寿命。

墨水注入打印机笔胆中最简单的方法是用尖咀钳拔出笔头,将吸有同样颜色墨水的钢笔沿着笔头的孔壁慢慢滴注墨水。四色绘图打印笔有黑、红、蓝、绿四种颜色,不可滴注颜色不同的墨水,以免污蚀微孔笔芯。待注好墨水,须对照笔头上的线痕仔细将笔头插入到原位。这种方法

的缺点是笔头插入会有偏差,长度不符可能损伤打印机。

不拔出笔头灌注墨水的方法是用细铜丝插进笔头与笔胆塑料连接部的细槽中,沿细铜丝滴上墨水,让墨水渗进细槽慢慢进入笔胆。这种方法的缺点是墨水往往不容易多量进入笔胆而流到外面。

行之有效的方法是借用医院或单位医务室注射用的旧针头和针筒,在针筒里吸入适量墨水。再用一根与针头粗细相当的缝衣针放在火上把针尖烧红,随即用它在打印机笔的塑料端部钻一个小孔。将针头插入小孔,就可以把针筒里的墨水从小孔注射进笔胆。注射好墨水后,用蜡或胶泥将小孔封堵住,打印笔便可重新使用。用完墨水的四色绘图打印笔可以用注射法反复注射同色墨水更新使用,十分方便。

# APPLE II 内外存储扩充技术和提高运行速度方法研究及实例分析

上海科技大学分部 沈欣 赵正德

**摘要:** 本文从软、硬件出发,深入剖析了APPLE II 的内外存储和分配技术,提出了节省内外存储空间、加快系统的运行速度的一系列措施,以达到相对扩充存储量和提高工作效率的目的。

## 一、引言

APPLE II 的鼎盛局面业已为 IBM PC 所取代。虽然APPLE的典型配置可有 64KB,但除去APPLESOFT (INTBASIC)、ROM、图型保留区,输入/输出区,实际留给用户使用的至多不超过60KB。其中40KB用于运行BASIC,系统主频仅1MHZ,这客观上决定了APPLE的存储量和运行速度远不及于IBM PC。尤其给设计大型程序和处理批量数据带来困难,甚至无法实施。这就要求我们更有效地利用现有存储空间,笔者在参阅了国内外有关资料的基础上,结合亲身实践,深刻地剖析了存储器内部的分配技术,并以大量实例加以说明,以利于APPLE的开发和应用。

## 二、扩充内存及磁盘的存储空间。

### 1. 内存的分配和重新设置用户区域。

系统建立时,分配的用户区有两块:\$300~\$3FFF,用来写一些短小的机器语言子程序和存放某些重要数据,这块用户区比较安全,一般系统不会访问它。但存储量仅200字节。另一块是\$0800~\$95FF,但若配置了汉字系统后,\$4000~\$5FFF存放汉字显示区,\$9000~\$9600作为汉卡组字区,这样,这块用户区分割为不连续的两部分:\$800~\$3FFF共14KB和\$6000~\$8FFF共12KB。当用户程序从\$800开始存放,长度超过14KB时,或者程序执行中占用了大量变量和数组时,就会侵占到汉字显示区,引起内存使用不当,造成显示混乱。解决方法:在程序开头设定LOMEM: 24576, LIMEM: 36864, 规定在\$800~\$3FFF专门存放程序,\$6000~

\$8FFF专门存放变量和数组。

### 2. 存储结构和程序优化。

(1) 采用分隔符: , 在一行程序中多写几个语句。因为程序区的存储格式是链式结构,每增加一程序行至少要占用5个字节。其中2个用于链指针指向下一程序行,2个用于行号,1个用于表示行结束。

(2) 尽量用INPUT,少用READ。因为INPUT直接输入到变量中,不设置数据区,占用空间仅占READ的一半。

(3) 尽量用整型数组,少用实型数组。因为实型数组的每个元素占用5个字节,比整型多3个。

(4) 减少对字符串变量的反复赋值,因为当改变一个字符串变量时,旧值仍然保留在内存中,可用FRE(0)来清理字符串。

(5) 尽量重复使用那些以前用过,现在不再使用的变量名。

(6) 重复使用的程序段,用GOSUB调用子程序。

(7) 少用REM语句。因为它把其后的字符均看作一个ASCII码。

(8) 用POKE, PEEK来存取公共变量数据。当变量值为0~255时直接使用;若大于255的正值,例A=1000,化为A1=INT(A/255) A2=A-255\*A1依次存取。

### 3. 程序的动态链接。

在设计大程序时,常把它分割成若干独立小程序,存在磁盘中,运行时动态链接,这样做既便于程序的模块化设计,又可避免由于程序过长

而超越了用户程序区。根据链接是否可以传送参变量,又可分成两种:拼凑式(程序一)和调用式(程序二)。它能使多个程序段被拼成一个完整的程序或连续执行多个程序。

#### 4. 扩大二进制文件的存储长度。

二进制文件是用来存放机器语言程序,高分辨率图形位映象等二进制资料,DOS3.3严格规定了B类文件的最大长度L\$~\$7FFF(32KB)。执行POKE43364, 255使B文件扩大到64KB的高容量。

#### 5. 有效扩充磁盘的存储量。

〈1〉将磁盘格式化为40个磁道。一般的格式化方法使\$00~\$02的3个磁道存入DOS系统程序,\$11磁道记录文件目录,因此一张磁盘实际留给用户使用的仅31个磁道,124KB。但在17磁道0扇区的VOTC表上从C4~FF并未使用,只是在格式化时被置为00,且在34上指明磁道数为35,可以通过IOB表调用DOS中的RWTS子程序,修改VTOC表的有关位置。(程序三)

〈2〉制作无DOS盘。若磁盘仅用作数据盘,无须引导DOS,则可通过修改VOTC表把\$01~\$02的2磁道释放出来,增加12KB。(程序四),若需要可以恢复DOS(程序五)。也可结合〈1〉〈2〉使用。

#### 6. 分离小汇编程序。

APPLE II在装入INTBASIC时,提供了小汇编程序MINI-ASSEMBLER,可以写入简单的汇编语言源程序,但同时也不得不被INTBASIC占用了一部分空间,因此把INTBASIC和MINI-ASSEMBLER分离出来,单独存入磁盘,可节省内存开支(程序六)。

#### 7. 把DOS移入16KB RAM卡上。

大多数的APPLE II都配置了16KB RAM扩充卡,将DOS移到用户程序所不能使用的这块卡上,可相对扩大用户区12KB。(由于篇幅有限,程序清单略)

#### 8. 检测可用空间大小的方法。

〈1〉检查内存大小。执行PRINT FRE(0),当可用空间大于32KB,得到一个负值,加上65536就是剩余空间大小。

〈2〉检查磁盘大小,有助于我们更有效地使

用和管理磁盘,程序七可以结合在CATALOG中同时显示。

### 三、提高运行速度。

#### 1. 编译源程序为目标程序。

对于侧重于运算的BASIC程序,若借助TASC程序把它编译成目标程序,一般可提高运行速度6~7倍,其编译过程分为PASS1和PASS2编译后生成一带后缀名.OBJ的B文件。有关操作见程序八。

#### 2. 精炼算法,优化程序。

〈1〉选择一个最佳算法。这往往是提高运算速度的关键。程序九和程序十,虽然完成的是同一项工作,但执行的速度相差甚远。

〈2〉重复使用的常数,表达式,应在程序中尽早赋给变量,因为处理常数比处理变量更花费时间。

〈3〉在计算表达式时,尽可能把乘除运算,化为减加计算,避免使用乘方。

〈4〉尽量避免重复找下标。在引用下标时,用简单变量归并公共因子或直接替代,因为找下标变量比简单变量慢。

〈5〉省去NEXT后的变量名。在正确使用循环语句的前提下,可以节省因检查FOR—NEXT中的循环变量名是否匹配所花费的时间。

#### 3. 扩展键盘功能,使它具备功能键。

所谓功能键,就是在按下ESC键后,再按一字母键,即完成了一个完整的指令的输入。这对于提高键盘输入速度是十分有用的,由程序十一可完成功能键的设置,它可以放在HELLO中,这样引导后就能使键盘具有功能键,按CTRL—RESET退出功能键状态,用CALL 24576恢复键盘功能键。

#### 4. 加快DOS存取速度。

DOS3.3在每次存取磁盘后,总要检验是否有误,由于APPLE的驱动器性能较可靠,所以可以执行:POKE46732, 76省略这一项工作。

#### 5. 缩短驱动器马达起动时间。

DOS3.3在开始读写操作之前,花半秒钟的起动时间,等待马达转速稳定,但这个时间估计



## 程序一

```

LOAD I
LIST
10 FOR I=1 TO 100
20 PRINT I*I
30 NEXT I
40 END
1 D$=CHR$(4)
2 PRINT D$;"OPEN B"
3 PRINT D$;"WRITE B"
4 LIST 10-30
5 PRINT D$;"CLOSE"
6 END

```

## LOAD S

```

LIST
100 FOR J=1 TO 100
110 S=S+J
120 NEXT J
130 PRINT S,I
140 END

```

## EXEC B

```

LIST
10 FOR I=1 TO 100
20 PRINT I*I
30 NEXT I
100 FOR I=1 TO 100
110 S=S+J
120 NEXT J
130 PRINT S,I
140 END

```

## 程序二

```

LOAD I
10 FOR I=1 TO 100
20 PRINT I*I
30 NEXT I
40 END
40 PRINT CHR$(4);"RUN S"

```

## 程序三

```

10 FOR I=768 TO 787:READ X:
POKE I,X:NEXT I:POKE 2,2
32:POKE 3,183
20 FOR J=47080 TO 47085:REA
D X:POKE J,X:NEXT J:POKE
47088,0:POKE 47089,96
30 POKE 47092,1:CALL 768:HO
ME
40 FOR K=24772 TO 24830:POK
E K,255:POKE K+1,255:K=K
+1:NEXT K
50 POKE 47092,2:CALL 768
60 DATA 165,3,164,2,32,217,
3,176,2,169,0,160,0,145,
2,169,0,133,40,96,1,96,1
,254,17,0

```

## 程序四

```

10 POKE 42344,76
20 POKE 44723,4
30 POKE 46922,96

```

## 程序五

```

10 POKE 42344,32
20 POKE 44723,12
30 POKE 46922,173

```

## 程序六

```

CALL -151
*2425<F425.F7FFM
*2436:44
*243D:44
*246A:44
*246D:44
*2476:44
*248E:44
*2493:44
*2497:44
*249C:44
*24B4:44
*24B9:44
*2537:45
*255B:45
*25FE:46
*25DD:46
*25E7:46
*2633:45
*263F:45
*2668:45
*2694:46
*2697:46
*26B5:46

```

## #26C1:46

```

*2735:47
*273C:47
*274A:47
*2751:47
*2754:47
*275B:47
*275E:47
*2765:47
*2768:47
*276D:47
*277E:47
*27F4:47
*27FC:46

```

```

*BSAVE MINI-ASS.OBJ A$2425,
L$03DB
*2666G

```

## 程序七

```

10 HOME
20 FOR I=896 TO 960:READ X:
POKE I,X:NEXT I
30 K=149-(157-PEEK(978)):PO
KE 905,K:POKE 938,K:CALL
919
40 R=PEEK(917)+256*PEEK(918
):PRINT"FREE SPACE:";R
50 DATA 1,96,1,0,17,0,145,3
,0,149,0,0,1,0,0,96,1,0,
1,239,216,0,0,169,3,160,
128,32,217
60 DATA 3,162,0,142,149,3,1
42,150,3,160,56,185,0,14
9,162,8,10,144,8,238,149
70 DATA 3,208,3,238,150,3,2
02,208,242,200,192,196,1
44,232,96

```

## 程序八

```

RUN TASC
SOURCE FILE ?
OBJECT CODE FILE ?
MEMORY USAGE :
DEFAULT CONFIGURATION ?
TYPE CNTRL-C TO ABORT,AND
THER KEY TO CONTINUE :
CODE GENERATION COMPLETE.
COMPILATION COMPLETE
BRUN XXX.OBJ

```

## 程序九

```

10 S=0
20 FOR K=1 TO 100
30 FOR I=1 TO K
40 S=S+I
50 NEXT I
60 NEXT K
70 PRINT "S=";S
80 END

```

## 程序十

```

10 S=0:X=0
20 FOR K=1 TO 100:X=X+K:
S=S+X:NEXT
30 PRINT "S=";S:END

```

## 程序十一

```

*6000,61C2
6000- A9 06 85 38 A9 93 85 39
6008- 4C EA 03 A4 24 B1 28 48
6010- 29 3F 09 40 91 28 68 60
6018- E6 4E D0 E2 B6 4F 2C 00
6020- C0 10 F5 91 28 60 20 F8
6028- 92 8E 98 93 AD 96 93 D0
6030- 63 AD 00 C0 2C 10 C0 C9
6038- 9B F0 01 60 A9 A0 20 EB
6040- 92 20 FB 92 AD 00 C0 C9
6048- BE 90 11 C9 AC 90 04 C9
6050- DB 90 2A 2C 10 C0 20 EB
6058- 92 4C 06 93 C9 85 B0 C0
6060- 2C 10 C0 69 40 38 20 2C
6068- FC 4C 1C 93 C9 89 90 E3
6070- C9 8C F0 DF 1B 69 40 20
6078- 9B FB 4C 1C 93 E9 AA 8D
6080- 97 93 A2 FF A9 C0 EB DD
6088- 59 93 D0 FA CE 97 93 D0
6090- F5 8E 96 93 AE 96 93 E8
6098- D0 99 93 8E 96 93 AE 98
60A0- 93 C0 C0 D0 10 A9 00 8D
60A8- 96 93 A9 A0 2C 10 C0 20
60B0- EB 92 4C 06 93 60 00 00
60B8- 09 C0 C3 C8 D2 A4 A8 C0

```

```

60C0- D2 D5 CE C0 D3 D4 D2 A4
60C8- A8 C0 00 CC 09 D3 D4 C0
60D0- C3 C1 D4 C1 CC CF C7 8D
60D8- C0 CC CF C1 C4 C0 D3 C1
60E0- D6 C5 C0 C4 A4 BD A2 84
60E8- A2 C0 D0 D2 C9 CE D4 C4
60F0- A4 BB A2 C0 C4 C5 CC C5
60F8- D4 C5 C0 CE CF D2 CD C1
6100- CC C0 C9 CE D6 C5 D2 D3
6108- C5 C0 06 CC C1 D3 C8 C0
6110- C0 D2 C5 C7 C8 D4 A4 A8
6118- C0 C0 C0 D0 D2 C9 CE
6120- D4 C0 C0 C1 D3 C3 A4 A8
6128- C0 C4 C5 06 CE C0 C3
6130- C1 CC C0 C4 C1 D4 C1
6138- C0 D0 C5 C5 C8 C0 C6 CF
6140- D2 C9 BD B1 D4 CF C0 C7
6148- CF D4 CF C0 C3 C1 CC CC
6150- AD B9 B3 B6 C0 C9 CE D0
6158- D5 D4 C0 D2 C5 C1 C4 C0
6160- D4 C1 C2 A8 C0 CC C5 06
6168- D4 A4 A8 C0 CD C9 C4 A4
6170- A8 C0 CE C5 D8 D4 C0 D0
6178- CF C8 C5 C0 D0 C4 CC A8
6180- C0 D2 C5 D3 D4 CF D2 C5
6188- BA C0 D2 C5 D5 D2 D2 CE
6190- C0 C7 CF D3 D5 C2 C0 D4
6198- C8 C5 CE C0 D5 D3 D2 C8
61A0- C0 D6 D4 C1 C2 C0 C3 CF
61A8- CC CF D2 BD C0 D8 C4 D2
61B0- C1 D7 C0 C4 D2 C1 D7 C0
61B8- CF CE C5 D2 D2 C7 CF D4
61C0- CF C0 A0

```

```

BSAVE APPLE-KEY.OBJ A$6000,
L$01C2
CALL 24576

```

```

1-CATALOG 2-LOAD 3-SAVE
4-D$= 5-PRINT D$; 6-DELETE
7-NORMAL 8-INVERSE 9-FLASH
Q-LIST --RUN A-ASC$(
B-DEF FN C-CALL D-DATA
E-PEEK F-FOR I=1 TO G-GOTO
H-CALL-936 1-INPUT J-READ
K-TAB( L-LEFT( M-MID$(
N-NEXT O-POKE P-PDL(
Q-RESTORE R-RETURN S-GOSUB
T-THEN U-USE( V-VTAB
W-COLOR= X-XDRAW Y-DRAW
;-RIGHT$( , -CHR$( , -STR$(

```

## 程序十二

```

CALL -151
*BEAF,BF9B

```

```

*BEAF- A9
BEBO- 00 4C 43 BE AD C9 B5 80
BE8B- 6C BF 8D 75 BF AD CA B5
BECC- 8D 6D BF 8D 76 BF AD CB
BECS- B5 8D FB BE AD CC B5 8D
BEDD- F9 BE EE BD B5 38 AD C3
BED8- B5 ED BD B5 85 42 AD C4
BEE0- B5 E9 00 65 43 AC BD B5
BE88- AE C1 B5 8A D0 08 AD C2
BEFO- B5 F0 53 CE C2 B5 CA B9
BEF8- 00 96 91 42 C8 D0 EC BC
BF00- EB B7 BE C1 B5 A9 OE BD
BF08- 06 B5 B5 43 AE C2 B5 F8
BF10- 1A A5 42 8D BF B5 A5 43
BF18- 8D C0 B5 20 49 BF EE C0
BF20- B5 CE C2 B5 D0 F5 AD C0
BF28- B5 85 43 AD CB B5 8D BF
BF30- B5 AD CC B5 8D C0 B5 AE
BF38- C1 B5 F0 FA 20 49 BF AE
BF40- C1 B5 A0 00 F0 A5 4C EA
BF48- A2 AC C5 B5 D0 OD AD C5
BF50- B5 AE CA B5 A0 01 20 65
BF58- BF A0 C0 AD BF B5 AE C0
BF60- B5 20 65 BF 60 8D F0 B7
BF68- BE F1 B7 B9 00 97 F0 18
BF70- BD EC B7 C8 B9 00 97 BD
BF78- ED B7 C8 C6 B5 A9 B7
BF80- A0 E8 20 B5 B7 B0 08 60
BF88- 20 EA A2 A2 05 D0 05 20
BF90- EA A2 A2 08 68 68 68 68
BF98- 8A 4C D2 A6

```

```

BSAVE FAST-LOAD.OBJ A$BEAF,L$
O0ED

```

```

10 PRINT CHR$(4);"BLOAD FAST
-LOAD.OBJ";POKE 42103,76;
POKE 42104,180;POKE 42104
,190

```

## 多路数据采集器与微机的软硬件接口

成都市421信箱

陈启录

本文着重介绍多路数据采集器与微机的软硬件适配方法。该采集器与接口在我们为用户研制的控制检测系统中得已使用,效果很好。本采集器与接口主要由多路模拟开关电路,高精度A/D转换器,单稳整形电路,双向可编程接口芯片等组成。见图四。

### 一、MC14433芯片的主要性能及工作原理

A/D转换器我们选用美国 Motorola 公司生产的  $3\frac{1}{2}$  位双积分式 A/D (即MC14433),它相当于二进制十一位 A/D, MC14433 具有精度高,价格低、功耗小,抗干扰能力强,输入阻抗高,结构简单,外接元件少,调试方便等优点。但其转换速度较慢,为3~12次/秒。但用于低速,抗干扰要求高的工业数据采集检测和自动控制系统中是可行的。

1. 主要性能: 精度: 读数的  $\pm 0.05\% \pm 1$  字  
量程:  $\pm 1.999V$  和  $199.9mV$   
转换速度: 3~12次/秒可调  
输入阻抗:  $\geq 1000M\Omega$   
电源电压范围:  $\pm 4.5V \sim \pm 8V$

得比较保守,执行POKE47102,236可以缩短,对于进行频繁读写的数据统计等操作,可以大大提高工作效率。

#### 6. 提高DOS装入程序的速度。

原来的DOS读盘操作是由RNTS子程序完成的,所以总是先将一个扇区的信息读入文件缓冲区,而不是直接取到用户程序区,因此要花费一段时间。如果先将文件中的第一个扇区信息读入文件缓冲区,并登记文件的长度,随后就将所有扇区直接读入程序区,仅在剩下字节数少于256时,才间接通过文件缓冲区,这样就可提高装入速度,程序十二就是一个快速装入程序。值得注意的是它改变了INIT的功能。

具有过量程和欠量程标志,可用于自动量程转换控制。

片内自动极性转换和调零功能。

片内提供时钟振荡电路,也可以外部输入时钟信号(50KHZ~200KHZ)。

数据输出为动态扫描BCD码,分别由DS<sub>1</sub>~DS<sub>4</sub>分时选通输出。

主要引脚功能:

DU: 为实时输出控制端,若DU输入一正脉冲,则使转换结果送至结果寄存器。

EOC: 为一次转换结束标志,前次转换结束时,输出一正脉冲,其宽度为 $\frac{1}{2}$ 个时钟周期。若EOC和DU短接,则每次转换结果都输出到输出锁存器。

DS<sub>1</sub>~DS<sub>4</sub>: 为千、百、十、个位数字输出选通信号。

Q<sub>1</sub>~Q<sub>4</sub>: 为BCD码输出端。

#### 2. 工作原理:

在本次转换结束时,EOCK—正脉冲(其宽度为 $\frac{1}{2}$ 个时钟周期)。结果经多路调制器变成BCD码,由数字选通脉冲DS<sub>1</sub>~DS<sub>4</sub>扫描输出

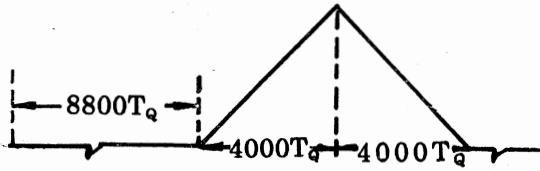
### 四、结束语

应该指出:在实际应用中,我们往往牺牲空间来换取工作速度,反之亦然,因此应根据需要情况,灵活掌握。

### 参考文献

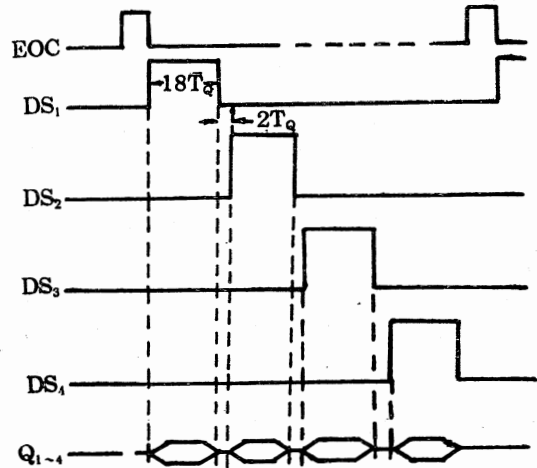
- [1] 朗·普尔: APPLE—II 微型计算机用户指南,北京出版社,1985
- [2] 吴志洪: 程序设计实用技巧,浙江大学出版社,1986
- [3] 林卓然: APPLE II 微电脑使用方法与技巧,中山大学出版社,1986
- [4] 张世英: 苹果—II BASIC 程序设计,北京师大出版社 1985

各DS选通信号占用18个时钟周期,因此,四位数共用80个时钟周期,所以扫描速率=时钟频率 $\div 80$  MC14433每转换一次需约16800个时钟周期,如图一所示。前8800个时钟周期用于内部放大器校零,后约8000个用于双积分。



图一. 积分波形

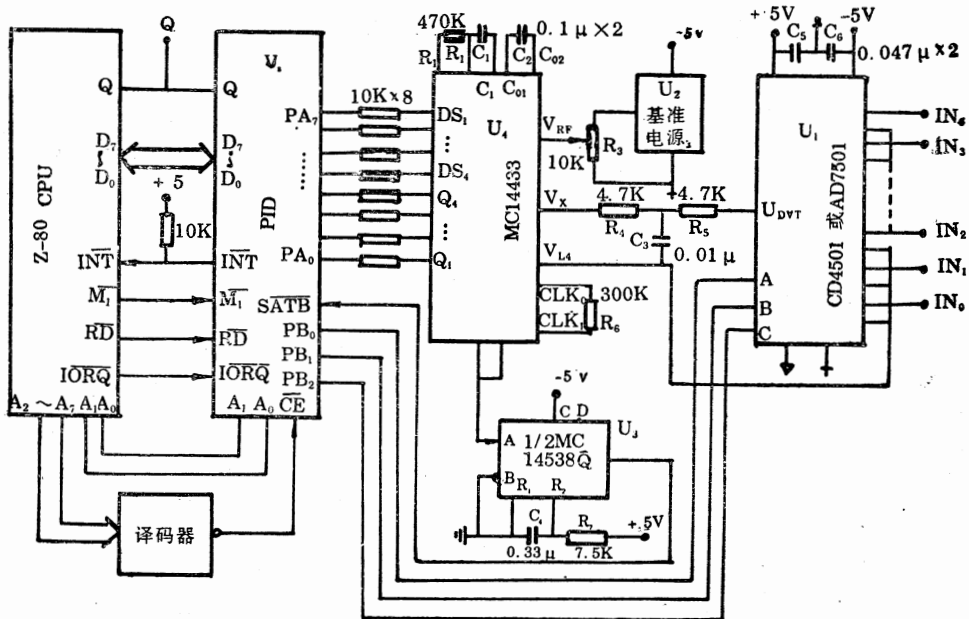
数据输出: MC14433的数据输出是以串行BCD码形式通过多路开关依照DS选通信号依次顺序扫描输出。在DS<sub>2</sub>~DS<sub>4</sub>期间, Q~Q<sub>4</sub>输出三位全位BCD码,即0~9十个数,但在DS<sub>1</sub>期间, Q<sub>1</sub>~Q<sub>4</sub>输出的千位数不是真正对应V<sub>in</sub>转换的千位数,而是极性、状态、标志、超量程,欠量程等的混合代码(在调试过程中要特别注意这一点)。DS<sub>1</sub>每扫描一次Q<sub>1</sub>~Q<sub>4</sub>输出是何种代码由表1的各状态位确定。位驱动与数字的时序图如图二。



图二. 位驱动时序图

## 二、与微机通讯接口及采样方法:

通讯接口如图三所示,从图中可知: U<sub>1</sub>为多路模拟开关电路,该器件选用CD4501或AD7501集成块。它的输入正端接前置放大器输出,负端连在一起与U<sub>3</sub>(A/D)的模拟地V<sub>AG</sub>相连,选择端A、B、C与接口片PIO的口端相连。由微机控制通道选择。U<sub>2</sub>为基准调压器,由它提供A/D的基准电压(2V或200mV)。U<sub>3</sub>为A/D



图三. 通讯接口原理图



## 微机在小氮肥厂中的应用综述

四川省电子技术研究所 彭献彬

**摘要:** 本文综述了我国小氮肥厂发展概况;造气顺序控制,入炉蒸汽流量控制,变换控制,氢氮比控制,合成塔触煤层温度控制,合成塔入口氨含量控制,弛放气控制,选用机型,主要仪表,典型经济效果,控制算法;列出了部分参考资料。

小氮肥厂从1958年诞生以来,全国已有一千多家,其化肥产量已占全国总产量的50%以上。小氮肥厂在支援农业,解决人们吃穿方面起着重要作用。

小氮肥三十年来的发展,取得了长足的技术进步。生产规模从年产氨800~5000吨,扩展到今天的1~2.5万吨。管理上总结了四管一算(原料管理,工艺管理,设备管理,全面质量管理,和经济核算)的基本管理方法,闯过了原料关,技术关、和经济关。小氮肥厂已占稳脚根,在逐步形成适合我国国情的工业体系[1]。

微机在小氮肥的应用是80年代初开始的,也取得了可喜的进展。本文试图就这个领域的微机应用做一综述,总结经验,以求推进发展。由于时间短,查阅的资料和调查的工厂有限,定有不妥之处。

### 一、造气部分的控制

全国小氮肥厂多以煤为原料,用间歇法生产合成氨的原料气一半水煤气。由于煤气发生炉手工操作劳动强度大,易发生事故,能耗高,环境恶劣,迫切要求机械化、自动化操作。

在30年代,机械式自动机就开始用来控制造气炉阀门的自动启闭,至今仍有不少厂在使用着。但它有制气循环时间不准确,又难以调整,体积大,占空间大等缺点。随着技术的进步各种定时与调整准确的灵巧自动机不断问世。

山西大同氮肥厂设计了气动程序自动机,结构简单,造价低,工作可靠[2]。

电气分离元件构成的自动机。

CMOS集成电路做成的数字式自动机,显示直观,工作可靠,效果好[3]。

微处理器为核心的可编程造气控制机[4],

性能十分优异。

上叙各种造气控制自动机虽然结构不同,但其基本功能都是一样的,都自动完成造气炉操作各阶段阀门的启闭。不过以微处理器为核心的造气自动机易修改造气循环周期和各阶段的分配比例,显示各炉运行阶段和阀门启闭状态,进行入炉蒸汽和氢氮比自调,联锁报警,甚至一机控制多台造气炉,并协调炉间关系等等。

以微机为基础的造气自动机,由于可编程其灵活性和适应性较硬逻辑型造气自动机好得多。在硬件不变或少变的情况下,通过修改部分程序或重新编程,自动机的功能就修改,甚至整个控制方案都被变动,这种优越性是硬逻辑型造气自动机无法比拟的。

按造气程序进行自动操作的自动机,在稳定造气操作,避免事故发生,减轻劳动强度等方面起了巨大的作用。但造气各阶段是否达到理想的炉况,只以时间为条件造气顺序控制自动机是没顾及的。

要解决带炉况条件的顺序控制问题,得解决炉况的检测。反映炉况的最重要参数之一是造气炉的炉温。由于造气炉这种对象用探头长期放在其内连续测温难以实现,必须找寻别的办法。采用水管间接测取气化层温度,应用证明可行,其误差小于 $50^{\circ}\text{C}$ [5]。近有用光纤温度传感器测温。

采用间歇、多点、数字显示测温,也行之有效[6]。测温装置由测量探头(包括测温套管和热偶丝),气缸,和数字温度仪组成。其缺点是测一次温,需较长时间。

炉温实时检测解决了,微机就可以根据炉温条件进行控制[7]。吹风阶段,微机根据气化层温度是否达到灰渣的熔点温度来决定是否结束吹

风阶段,转入制气阶段。其余各阶段时间比例不变,但它们的总时间等于一个周期时间减去吹风阶段用去的时间。这一措施保证了制气的开始炉温和合理的吹风时间长度,较只依固定的吹风时间为条件为优。

为了在尽量短的时间内升到制气温度,微机可以协调各造气炉之间的关系,使它们之间操作周期错开来,如有两台炉,彼此错开二分之一周期,三台炉则错开三分之一周期,以增大吹风量。

到目前为止,还未看到制气阶段怎样根据炉况来调整制气时间和方向的报导。

半水煤气生产中,入炉蒸汽流量调节也是研究得比较多的。上下吹制气随着时间的延续,炉温下降,需要入炉的蒸汽量也应随之减少,其减少规律,有的厂已做成曲线,微机按曲线进行控制,提高了蒸汽的分解率,减少了蒸汽耗量[8]。

资料[7]根据气化层温度、煤种、蒸汽分解率,得出蒸汽在炉中停留时间 $t$ ,由 $t$ 按下式计算入炉蒸汽流量 $G$ :

$$G = C \cdot \frac{1}{t}$$

式中:  $G$ —入炉蒸汽流量,  $M^3/S$ ;

$t$ —停留时间,  $S$ ;

$C$ —系数,取决于炉床截面积和蒸汽在气化层通过的距离的乘积。

入炉蒸汽流量调节回路按 $G$ 值改变蒸汽流量。

造气部分还要控制氢氮比。通过控制吹净时间(回收时间)来达到:若 $(H_2 + CO)/N_2 > 3:1$ ,加长回收时间,增加半水煤气中 $N_2$ 含量;反之则缩短回收时间,减少煤气中 $N_2$ 含量(上下吹加氮量一般固定不做调节)。详见氢氮比控制部分。

造气部分选用的微机,主要有 TP801 单板机[7],一位微处理器做成的造气控制机 WGK—C[4],紫金—Ⅱ微机[9]。执行机构采用油压的较水压的为优,阀门为闸板阀。入炉蒸汽用连续调节伐。

资料[7]报导的微机控制造气过程取得的

效果数据是:单炉发气量 $1683HM^3/h$ (手动1100),平均炉渣含炭量5.97%(手动20.89%),平均蒸汽分解率55.88%(手动38.84%),半水煤气成份:

$CO_2$	$O_2$	$CO$	$H_2$	$N_2$	$CH_4$
8.0	0.4	32.1	41.7	15.9	1.9

## 二、变换部分的控制

变换部分主要控制水蒸汽流量,触煤层温度,和变换气中 $CO$ 含量。通过调节水蒸汽流量来控制 $H_2O/CO$ ,触煤层温度,保证变换气中 $CO$ 含量在允许的范围内,同时还要少用水蒸汽。

有的厂用变换气中 $CO$ 含量与水蒸汽/半水煤气串级系统:

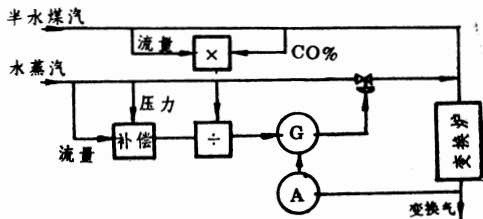


图2-1  $CO$ 与水蒸汽/半水煤气串级系统

水蒸汽流量经压力补偿除半水煤气中的 $CO$ 流量作为付参数,变换气中 $CO\%$ 作为主参数组成串级系统。还有考虑饱和塔回收蒸汽的。

和变换炉触煤温度控制系统,用调节喷水量或炉入气温度来维持。

有的用可编程控制器和 DDZ—Ⅱ 调节器共同来控制变换部分[10]。可编程控制器控制框图见图2-2。本系统是一个前馈加反馈控制系统。一段炉温作为主调参数( $PV_1$ ),蒸汽流量( $X_2$ )经压力( $X_3$ )补偿作为付调参数( $PV_2$ ),构成串级调节;半水煤气中 $O_2\%$ ( $X_5$ )静态前

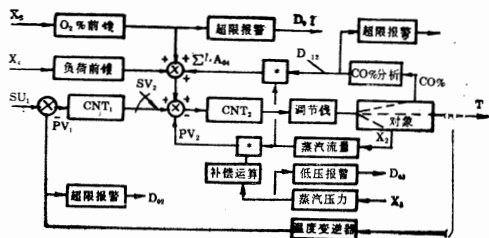


图2-2可编程控制器变换的应用框图

馈; 半水煤气流量 ( $X_4$ ) 动态前馈; 变换气  $\text{CO}\%$  超给定补偿;  $\text{O}_2\%$ 、蒸汽压力、炉温超限报警。CNT1和CNT2为主付调节器,  $\text{SV}_1$ 和 $\text{SV}_2$ 为外给定。

此系统达到吨氨蒸汽耗400kg; 温度波动; 一段炉温 $<\pm 10^\circ\text{C}$ , 二、三段 $<\pm 5^\circ\text{C}$ ; 变换气中 $\text{CO}\%$ 稳定; 触媒寿命延长。

### 三、氨合成部分控制

这部分有合成塔触媒层温度控制, 氢氮比控制, 合成塔进口氨含量控制, 和弛放气控制。

#### 1. 合成塔触媒层温度控制 [11]

此系统一般都是以触媒热点温度作为主参数, 上层敏点温度作为付参数的串级带分程调节的系统。精度在 $\pm 3^\circ\text{C}$ 之内。

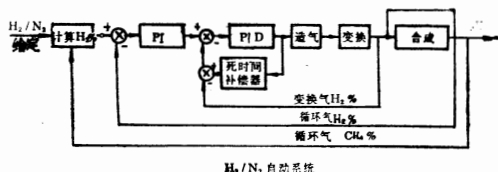
#### 2. 氢氮比控制

欲使氨合成反应速度一直处于最大, 工艺研究表明,  $\text{H}_2/\text{N}_2$ 应从反应始的1.5:1递增到平衡时的3.0:1。工艺实现困难, 至今仍按3:1控制。

本系统由于干扰因素多, 纯滞后时间长, 和无自衡等特点, 未见有用常规仪表成功的报导。

采用微机进行氢氮比控制, 在好些省都取得较为满意的结果,  $\text{H}_2/\text{N}_2$ 合格率达90%以上 [12][13][14][15]。

氢氮比自调系统, 各省尽管局部有所不同, 但可能都因我国测氢仪表较成熟在小氮肥厂用得普遍, 都采用测氢调氮的办法。系统的输入参数有: 半水煤气 $\text{CO}+\text{H}_2$ 含量, 或变换气 $\text{H}_2$ 含量, 或补充气 $\text{H}_2$ 含量, 作为付参数; 循环气 $\text{H}_2$ 含量, 作为主参数, 组成串级调节; 循环气中 $\text{CH}_4$ 含量  $\text{H}_2/\text{N}_2$ 给定值, 作为计算主环给定值  $\text{H}_2\%$ 。系统方块图如下。



主环给定计算式如下:

$$\text{H}_2\% = (97 - 1.3X_{\text{CH}_4}) \frac{r}{1+r}$$

式中:

$X_{\text{CH}_4}$ —循环气 $\text{CH}_4\%$ ,  $r$ —氢氮比。此式按进口氨含量为3%和循环气中  $A=0.3\text{CH}_4$  导出, 否则得修正。

采用 smith 死时间补偿器来补偿死时间的影响。死时间补偿器的传递函数为  $w_p(s)(1-e^{-LS})$ , 其中  $w_p(s)e^{-LS}$  是被调对象的传递函数。

采用区间控制。当循环  $\text{H}_2$  在给定值的某一范围, 主环不计算输出, 仍维持上次值, 从而提高系统稳定性。

采用间断控制, “看看—调调—等等”。采样数据控制系统, 控制本来就是不连续的, 对于死时间长的对象, 笔者认为等待间隔至少要等于对象的死时间, 会使系统更稳定。

采用积分分离算法。积分作用只有在一定偏差范围内才起作用, 小于或大于此范围无作用。

微机多选用TP801单板机, 也有用紫金—Ⅰ微机的。软件为固化程序(前者)。  $\text{H}_2\%$ 用热导式分析器RD—04型,  $\text{CH}_4$ 和  $\text{CO}$ 用红外分析器HW—001型, FQ—C型, 为南京和佛山分析器厂的产品, 也有用上分和川分的。

#### 3. 合成塔进口氨浓度调节系统

合成塔进口气氨浓度与出口混和气中气氨冷凝分离时的压力和温度有关系:

$$\log X_A = 4.1856 + \frac{5.9879}{\sqrt{P}} - \frac{1099.5}{T}$$

式中:

$X_A$ —液氨上气体的平衡氨浓度, %;

$P$ —混和气的压力, 大气压;

$T$ —混和气的绝对温度。

通过调节出氨蒸发器循环气的温度来调节循环气中的氨浓度  $X_A$ 。

#### 4. 弛放气的控制 [16]

工业生产中, 通过把一部分循环气放空来维持循环气的最适宜惰性气体含量。微机控制仿效人的操作, 测循环气 $\text{CH}_4$ 含量, 调放空量, 同时还加上逻辑功能: 合成系统压力超限时, 放空阀全开, 以保证工艺设备安全; 压力正常时, 按 $\text{CH}_4\%$ 自调。

#### 四、小氮肥厂微机控制采用的几种算法

小氮肥厂微机应用中用到的几种算法补充在这里,供参考。它们在本文附的参考资料中不一定找得到。

##### 1. PID算法

PID算法使用得十分广泛,工业控制问题中的绝大多数,使用它都得到了成功的解决。数字PID算法有下面种种。

常规调节器模拟算法离散化的等效数字算法:

$$V_n = V_0 + K_p \left[ e_n + \frac{T_s}{T_I} \sum_{i=0}^n e_i + \frac{T_D}{T_s} (e_n - e_{n-1}) \right] \quad (4-1)$$

式中:

$V_n$ —第 $n$ 次采样时刻控制器的输出;

$V_0$ —上次算法的稳态输出,它使偏差 $e$ 为零;

$K_p$ —比例增益;

$T_s$ —采样间隔,或叫采样周期;

$T_I$ —积分时间常数;

$T_D$ —微分时间常数;

$e_n$ —第 $n$ 次采样时刻的偏差值。 $e_n = SP_n - PV_n$

$SP_n$ —第 $n$ 次采样时刻的给定值;

$PV_n$ —第 $n$ 次采样时刻的过程变量测量值;

$e_{n-1}$ —第 $n-1$ 次采样时刻的偏差值。

(4-1)叫位置形式,PID的速度形式为:

$$V_n = V_{n-1} + K_p \left[ (e_n - e_{n-1}) + \frac{T_s}{T_I} e_n + \frac{T_D}{T_s} (e_n - 2e_{n-1} + e_{n-2}) \right] \quad (4-2)$$

(4-1)和(4-2)式为非理想形式。理想PID算法也有位置和速度两种形式,位置形式为:

$$V_n = V_0 + K_p e_n + \frac{T_s}{T_I} \sum_{i=0}^n e_i + \frac{T_D}{T_s} (e_n - e_{n-1}) \quad (4-3)$$

速度形式为

$$V_n = V_{n-1} + K_p (e_n - e_{n-1}) + \frac{T_s}{T_I} e_n + \frac{T_D}{T_s} (e_n - 2e_{n-1} + e_{n-2}) \quad (4-4)$$

速度形式算法消去了 $V_0$ 和积分加和,理想PID算法P、I、D三项分开,在实际使用中带来方便。

(4-1)~(4-4)式中,取 $T_I = \infty$ ,得到PD算法;取 $T_D = 0$ ,得PI算法;取 $T_I = \infty$ ,和 $T_D = 0$ ,得P算法。若 $|e_n| < E$ (隙限常数)时,不计算新的输出,就得PID隙限算法。又若 $|e_n| > L$ (常数),用PD算法,否则用PID算法,就得积分分离算法。

##### 2. smith死时间补偿算法

大多数工业对象都能用带纯滞后的一阶惯性环节来描述:

$$\omega_p(S) = \frac{Ae^{-LS}}{TS + 1}$$

式中, $\omega_p(S)$ —被控对象的传递函数;

$A$ —放大倍数;

$L$ —纯滞后时间;

$T$ —时间常数。

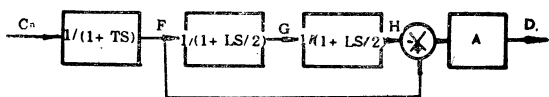
Smith死时间补偿器的传递函数为

$$\omega_{sp}(S) = \frac{A}{TS + 1} (1 - e^{-LS})$$

将 $e^{-LS}$ 展开成幂级数,取前三项代入上式得:

$$\omega_{sp}(S) = \frac{A}{TS + 1} \left[ 1 - \frac{LS}{1 + \frac{LS}{2}} \right]$$

把上式画成方块图:



把各方块离散化,得Smith死时间补偿算法

$$F_n = F_{n-1} + \left( \frac{T_s}{T_s + T} \right) (C_n - F_{n-1}) \quad \dots\dots\dots (4-5)$$

$$G_n = G_{n-1} + \left( \frac{T_s}{T_s + \frac{L}{2}} \right) (F_n - G_{n-1}) \quad \dots\dots\dots (4-6)$$



$$H_n = H_{n-1} + \left( \frac{T_s}{T_s + \frac{L}{2}} \right) (G_n - H_{n-1}) \quad \dots\dots (4-7)$$

$$D_n = A(F_n - H_n) \quad \dots\dots (4-8)$$

### 3. 前馈控制算法

假设干扰通道和调节通道的传递函数分别为

$$G_L(S) = \frac{K_L}{T_L S + 1} e^{-\theta_L S}$$

$$G_P(S) = \frac{K_P}{T_P S + 1} e^{-\theta_P S}$$

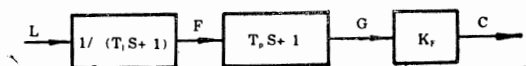
前馈调节器要全部补偿干扰其传递函数应为

$$G_F(S) = -\frac{G_L(S)}{G_P(S)} = -\frac{K_L}{K_P} \cdot \frac{T_P S + 1}{T_L S + 1} \cdot e^{(\theta_P - \theta_L)}$$

设  $K_F = -K_L/K_P$ ,  $\theta_P - \theta_L = 0$ , 上式变成

$$G_F(S) = K_F \frac{T_P S + 1}{T_L S + 1}$$

把上式化成方块图



把各方块离散化, 就得前馈控制算法:

$$F_n = F_{n-1} + \frac{T_s}{T_s + T_L} (L_n - F_{n-1}) \quad (4-9)$$

$$G_n = \frac{T_P}{T_s} (F_n - F_{n-1}) + F_n \quad (4-10)$$

$$C_n = K_F G_n \quad (4-11)$$

若损去超前滞后的动态项, 就得静态前馈算法:

$$C_n = K_F \quad (4-12)$$

因为前馈调节器的几个参数  $K_F$ ,  $T_P$ ,  $T_L$ , 难于准确得到, 且  $\theta_P = \theta_L$  也不一定符合实际, 所以很少单独采用前馈控制方案, 而与反馈控制一齐使用组成前馈加反馈系统。

### 参考资料

- [1] 庆迪益,  
“河北省小氮肥的经济规模”  
《河北化工》 88年 No.2
- [2] 刘四海, 山西环县化肥厂  
“气动程序自控机的应用”  
《小氮肥设计技术》 87年 No.2
- [3] 王文光, 浙江衢州化肥厂  
“煤气发生炉全集成数字控制仪”  
《化工自动化及仪表》 85年 No.6
- [4] 李东林, 西南化工研究院  
“合成氨造气工程微机控制系统”  
《化工自动化及仪表》 86年 No.1
- [5] 秦鸿钧, 湖南大学  
“固定床造气炉气层温度检测和炉况”  
《湖南化工》 87年 No.2
- [6] 蒋俊甫, 江苏省太原化肥厂等:  
“我省小氮肥煤气炉测温技术”  
《化工自动化及仪表》 89年 No.2
- [7] 易武, 湖南大学  
“固定床造气新型微机自控系统”  
《计算技术与自动化》 87年, No.1
- [8] 李建设, 山东寿光化肥厂  
“造气炉蒸汽的微机控制”  
《小氮肥设计技术》 87年 No.5
- [9] 上海化工研究院自动化室  
“STD-1分散型计算机控制系统及其他小化肥厂中的应用”  
《化工自动化及仪表》 89年 No.2
- [10] 顾鼎华, 西南化工研究院刘顺民, 四川梁平化肥厂, 刘顺民  
“SLPC在合成氨变换工程的应用”  
《化工自动化及仪表》 89年 No.2
- [11] 李小军  
“合成塔触媒温度自动调节微机系统”  
《化工自动化及仪表》 86年 No.6
- [12] 蓝怀科, 广西降安氮肥厂  
“微机控制氢氮比系统在我厂的应用”  
《小氮肥设计技术》 88年 No.2
- [13] 单传梅等, 山东肥城化肥厂:  
“氢氮比微机控制系统”  
《化工自动化及仪表》 86年 No.2
- [14] 史恒生, 河北平山县化肥厂:  
“造气工程的十项小改革”  
《河北化工》 89年 No.1
- [15] 张继降, 陕西魏城氮肥厂  
“氢氮比微机控制的应用与效益”  
《小氮肥设计技术》 87年 No.5
- [16] 陈昌荣, 云南天然气化工厂:  
“合成氨弛放气控制系”  
《化工自动化及仪表》: 84年, No.2

# E<sup>2</sup>PROM2864的使用体会

温州市业余科技大学 张向春

电擦除可编程只读存储器E<sup>2</sup>PROM不但具有ROM的非易失性,而且具有RAM的随机读写性。它可应用于数据保护、参数修改、程序调试等许多方面。2864是8KB的E<sup>2</sup>PROM,其引脚排列与SRAM6264/EPROM2764兼容,故在使用6264/2764的地方,不需要修改电路即可取代之。目前看到的资料几乎都是介绍Intel公司的2864A [1], Intel 2864A的维持方式和读出方式与SRAM/EPROM相同,写入方式分为字节写入和页面写入两种,编程时,有数据查询功能。然而,国内市场上见到的芯片许多为SEEQ公司的2864。笔者在使用中发现,SEEQ2864(以下简称2864)与Intel2864A的性能有不同之处:

1. 只有字节写入方式,没有页面写入方式。
2. 编程时,没有数据查询功能。

## 一、单板机连接2864

TP801单板机与2864的连接如图1所示,其地址区间为4000H~5FFFH。象RAM一样,通过单板机的MEM键可对2864存储单元进行检查和修改。但要注意的是,对存储单元进行修改时,键入两位数字后,数码管显示器低两位显示的并非这两位数字,而是FFH(尽管内容已正确写入)。这是因为单板机监控程序是针对RAM编制的,其数字键处理程序中有下列两条指令:

```
LD (HL), A
LD A, (HL)
```

由于RAM的存取速度快,写入后紧接着读出的内容即为刚写入的数据。而E<sup>2</sup>PROM的字节写周期时间一般为几个毫秒,写入后内部编程,数据线呈高阻状态,随即读出则为伪数据FFH。为此,笔者在监控程序中增加了三条指令,即修改为

```
LD (HL), A
PUSH HL
CALL D20MS
```

```
POP HL
LD A, (HL)
```

修改后,对E<sup>2</sup>PROM的读写操作步骤,就象对RAM的操作一样,非常方便。

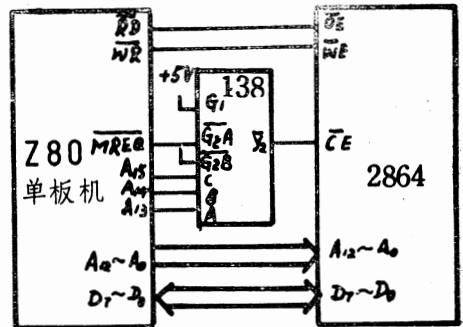


图1 单板机连接2864原理图

## 二、单片机连接2864

8031单片机与2864的连接如图2所示,其地址区间为0000H~1FFFH。在前面,已提到2864单字节写入所遇到的问题及解决的方法,下面则介绍数据块传送的方法。

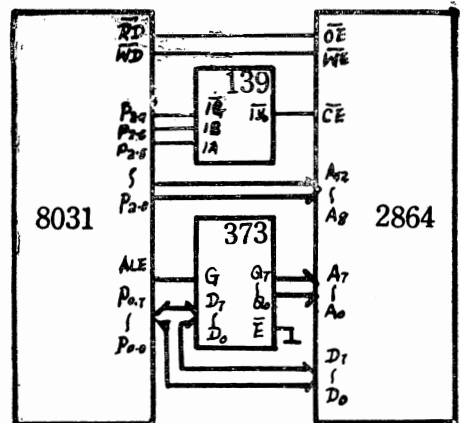


图2 单片机连接2864原理图

Intel 2864A的数据块传送通常采用查询法: CPU向2864写入一个字节后,不断地查询读取该写入单元的内容,并与原来装入的数据比较,相同后再传送下一个字节。然而使用SEEQ 2864时,采用查询法传送数据块则会遇到问题:当传送的数据中有FFH时,其后续数据无法写入。因为当写入FFH时,读取比较即相符(数据线处于三态),则程序会立即转入传送下一个字节。由于前字节编程未结束,则该字节写入无效,因此,该字节的读出比较将不可能相同,使程序进入死锁。为此,笔者采用延时法[2]进行数据块传送,其8031数据块传送子程序如下:

#### 程序清单

```

E2MOV:  MOV    DPTR, # SOUR
        MOV    R0, # DAL
        MOV    P2, # DAH
        MOV    R1, # BYTES
WR:      MOVX   A, @DPTR
        MOVX   @R0, A
        ACALL  DL2
        INC    DPTR
        INC    R0

```

```

        CJNE   R0, # 0, NEXT
        INC    P2
NEXT:    DJNZ   R1, WR
        RET
DL2:     MOV    R7, # 4
DL:       MOV    R6, # 0FAH
DLA:     DJNZ   R6, DLA
        DJNZ   R7, DL
        RET

```

程序中, SOUR为数据块源地址, DAL为目的地址低位, DAH为目的地址高位, BYTES为字节数, DL2为延时2毫秒子程序, 由于2864的字节写周期时间经实际测试约为1、2毫秒, 所以程序中数据写入后延时2毫秒, 以保证能正确地写入。读者在使用2864时, 应根据芯片的实际性能确定合适的延时时间。

#### 参考文献

- [1] 赵超: “新型E<sup>2</sup>PROM 2864A”, 《电子技术应用》 1987年第11期。
- [2] 张向春: “查询输入/输出死锁问题的解决”, 《计算机应用研究》 1989年第2期。

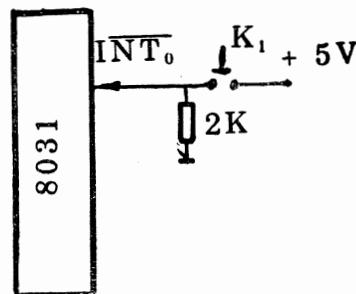
## 实现MC—51单片机单步操作的简易方法

安徽机电学院电气系 周鸣争

**摘要:** 本文针对mc—51系列单片机中断系统的结构特点, 介绍了一种利用中断技术实现对单片机的控制程序进行单步操作的简易方法, 以满足单片机应用系统设计和软件调试及维修的需要。该方法硬件简单, 使用方便, 特别适用于一些缺少单片机开发工具的用户。具有一定的实用价值。

近年来, 单片机以它较高的性能/价格比在工业过度控制、仪器仪表、家用电器等领域越来越得到广泛的应用。通常, 在单片机的应用系统的设计和调试过程中, 往往需要对系统控制程序进行单步执行。而MC—51系列单片机与MC—48系列单片机不同, 在硬件结构中不具有这种功能。为了满足这一实际要求, 本文笔者针对MC—51系列单片机的中断结构特点, 提出一种基于中断技术实现控制程序单步操作的简易方法, 以弥补MC—51单片机功能的不足。这种方法硬件结构简单, 使用方便, 由于采用中断方式而实时性强, 特别适用于一些缺少单片机开发工具的用户使用, 具有一定的实用价值。其硬件结构如图一所示。

户使用, 具有一定的实用价值。其硬件结构如图一所示。



图一

在图中,  $\overline{INT}_0$  为 MC—51 单片机的外部中断, 它的中断请求是否会得到响应, 受到 mc—51 单片机内部允许中断寄存器 IE 中第 0 位的控制, 其优先级则由中断优先寄存器 IP 的第 0 位确定, 当中断源提出中断请求后, 作为应答, CPU 首先使相应的“优先级激活”触发器置位, 以阻断同级和低级的中断。CPU 对  $\overline{INT}_0$  的激活方式常分为两种, 一种是电平激活, 另一种是边沿激活。这两种方式可以靠 TCON 寄存器中的中断方式位  $IT_0$  来控制。若  $IT_0$  为 0, 则采用电平激活方式, 在  $\overline{INT}_0$  引脚上检测到低电平, 将触发外部中断。若  $IT_0$  为 1 则采用边沿激活方式, 在相继的二个周期中, 对  $\overline{INT}_0$  引脚进行连续二次采样, 若第一次采样值为高, 第二次为低, 则 TCON 寄存器中的中断请求标志  $IE_0$  被置 1, 以请求中断, 图中采用电平激活方式, 在这种方式中, 由于在硬件上, CPU 对  $\overline{INT}_0$  引脚的信号完全没有控制 (即在专用寄存器中, 没有相应的中断请求标志), 也不象某些微处理机那样, 响应中断后会自动发出一个应答信号, 用以消去其请求信号。CPU 响应外部中断请求后, 在中断返回前, 该中断请求应撤除, 否则会引起另一次中断。从 MC—51 单片机的中断结构我们知道当一个中断还在进行时, 另一个同级或低级的中断请求是不会得到响应的, 而且 CPU 在中断返回之后至少还要执行一条其它指令, 这个中断请求才会得到响应。因此, 一旦进入到某中断服务子程序, 在中断返回并且再执行一条被中断程序的指令之前, 该中断服务程序是不能重入的。利用 MC—

51 单片机这一中断特点, 若将图二中所示的几条指令作为  $\overline{INT}_0$  的中断服务程序, 当需进行单步

```
JNB P3.2, $; 在  $\overline{INT}_0$  变高前, 原地等待;
JB P3.2, $; 在  $\overline{INT}_0$  变低前, 原地等待;
RETJ
```

图二

执行程序时, 通过软件将 IE 中的第 0 位置 1, 允许  $\overline{INT}_0$  中断, 同时, 将需单步执行的系统应用程序作为主程序。图一中  $\overline{INT}_0$  (3.2) 引脚通常保持低电平, CPU 一直工作于外部  $\overline{INT}_0$  中断服务程序中, 从图二中可知, 在单步按钮  $K_1$  没有按下时, CPU 停止在 JNB 处, 原地等待, 当单步按钮按下后,  $\overline{INT}_0$  端出现一个正跳变, 此时  $\overline{INT}_0$  输入端为高电平, 则中断服务程序停在 JB 指令处等待, 当  $\overline{INT}_0$  又由高电平变为低电平时 (即  $K_1$  断开), 程序往下执行 RETI, 返回应用程序 (即主程序), 执行完一条指令后, 由于这时  $\overline{INT}_0$  端为低电平, CPU 又再次执行中断服务程序, 等待  $K_1$  的再一次闭合。因此,  $K_1$  每闭合一次, CPU 则执行一条主程序中的指令, 从而实现了对系统应用程序单步执行的目的。但需要注意的是,  $K_1$  单步执行键按下的时间 (即  $\overline{INT}_0$  端保持为高电平的时间) 要大于 3 个机器周期 (即 3 $\mu$ s), 以确保 CPU 能采集到高电平值。实际上一般按键的动作均能满足这一要求。

### 参考文献

《Intel 单片机手册》上海市微电脑应用协会

## 人民银行卫星通信网络加紧施工

国务院批准建设的中国人民银行卫星通信计算机网络经过调研、论证、签约, 现已进入设计、安装、调试阶段。

由于缺乏必要的物质技术手段, 人民银行自行使中央银行职能以来, 没有建立起完善有效的资金清算汇划系统。银行之间资金的异地往来一直是通过邮政通信手段, 单据在途传递时间较长, 资金拖欠严重, 全国汇兑在途资金因此高达

400~500 亿元。各专业银行系统内汇兑资金和信贷资金不分普遍存在, 办理跨系统转汇业务时占用其它银行资金现象时有发生, 利用汇差资金扩大信贷范围已成为控制信贷规模的一个漏洞。由于资金清算汇划系统运转不畅, 还使得中央银行难以有效地发挥监测和调控宏观经济, 维护金融运转正常秩序的作用。

为了强化完善中央银行的宏观调控手段, 加

快结算速度,加速资金周转,国务院决定分配一个国产卫星转发器建立人民银行资金清算汇划网络系统,这是我国目前通信手段最为先进的大型计算机网络。按照工程设计,卫星通信主站设在北京,在全国323个一、二级分行所在城市设小型卫星站,组建全国范围的微波通信计算机网络,

系统于1990年全部建成后,将实现全部联行资金往来随发随收,当时核对,全国和各地资金流向及数量变化情况当日可知,产生极大的社会经济效益。

(桑小军 邓京明)

## 我国用微机控制电力系统取得突破

### 南京地调自动化系统通过部级验收

南京供电局研制成功的南京地区供电网调度自动化系统,最近在南京通过了能源部验收,成为全国第一个达到部颁实用化标准的地区级电网

调度自动化系统,标志着我国电力调度自动化已进入实用化阶段。该系统已被确定在全国范围内逐步推广。

(徐 洁)

## 仓库基础数据管理系统通过部级鉴定

全国第一个中型计算机应用软件《仓库基础数据管理系统》日前(8月5日)在武汉通过了由总后勤部组织的技术鉴定。

这套由解放军总后基地指挥部自动化站在中型机上研制成功的《仓库基础数据管理系统》是一个汉字交互式信息管理系统。它由基本情况管理、通信设施管理,营房情况管理等八个应用程序子系统组成,共有十六个数据文件。约五百多个数据项目,一百余项功能。系统采取汉字菜单驱动,提示信息帮助、屏幕设计新颖,人机界面友好。系统除具有通常的检索查询、统计分析,数据维护、汉字表格输出等功能外,还具有模块组装与拆除,应用系统生成、功能路径直接寻地,记录键值修改,多窗口屏幕交互控制,汉字

信息与代码自动转换等技术特点。来自总后勤部、国家科委、国防科技大学等十六个单位的二十二位专家、教授一致认为:《仓库基础数据管理系统》设计合理,技术先进,在国内同类机型的汉字交互式信息管理系统开发中处于领先地位。

该系统的研制成功,为目前微机上现有的汉字交互式应用软件向中型机移植,充分利用中型机存贮容量大,运行速度快的优势,以及微机通过仿真接口,在近程或远程网络上共享中型机的资源,作出了积极的贡献;系统的汉字交互式开发方式和所采取的一些独到的方法与手段,在中型机的应用领域具有很好的推广价值。

(阮高华)

## 安徽省计算中心部分科研成果简介

安徽省计算中心(又名安徽省信息技术研究所),建于一九七八年,是安徽省科委直属的科研单位,从事计算机开发、应用和技术服务。

十年来在国家科委的关怀和省科委的直接领导下,安徽省计算中心不断发展壮大,先后完成研究课题五十多项,其中七项获安徽省科技进步

奖,发表科研论文近百篇,现将部分科研成果介绍给大家,愿与各界朋友共同研讨,热忱地为社会各界服务。

### 农业病虫害测报程序系统

农业病虫害测报程序系统,是安徽省科委

1985年下达的指令性科研项目,由安徽省计算中心、农业病虫科研和植保部门联合研制,于1987年12月通过了安徽省科委主持的技术鉴定。该系统横跨农业昆虫测报原理、数学、计算机软件等多门专业领域,是农业病虫测报部门中长期数值预报工作的微机通用软件。1988年曾荣获安徽省科技进步二等奖,安徽省发明二等奖,并选送北京国际发明奖展览会及马来西亚国际博览会展览。

该系统功能完备,算法先进,运行可靠,测报对象多,测报水平高,在实用性、通用性、完备性、可靠性等方面属于国内领先水平;在系统构成、算法内核、用户接口等关键性技术上有重大创新,属国内首创,其中应用模糊聚类、系统聚类、模糊决策等方法具有八十年代国际先进水平。

该系统运行环境是IBM-PC/XT, CCDO S2.1版, FORTRAN-77算法语言编程, BASIC菜单程序驱动。系统由预测模型子系统、文件管理子系统、数据预处理子系统、预报实践子系统、联系求助子系统组成。以菜单驱动执行,汉字屏幕显示,汉字打印及磁盘文件存贮信息,对键盘输入作语法、语义检查,共158种查错功能。对任一测报对象,允许一次性处理25个预测目标和150个相关因子的100年历史资料。系统将数据库管理、预测模型处理,联机学习等功能融为一体,便于农业技术工作者掌握使用。

本系统历时三年,大规模应用于15个地、市、县植保站,用于水稻三化螟、小麦粘虫测报;正式发布虫情预报45期,189次,准确率达89.5%,最长时效108天。三年累计经济效益1834多万元,在减轻农药对环境污染、促进生态平衡方面也取得明显效果,该项目为今后开发其它病虫种类或其它的电算测报奠定了良好基础,也为我国对农业病虫逐步实现模型化测报积累了经验。

### 织机计算机监测系统

织机计算机监测系统,是1982年国家科委、省科委下达的科研任务,于1985年元月通过安徽省科委主持的技术鉴定。该系统采用具有分布特

点的两级微机处理系统。主要用于纺织企业的布机车间的织机监测,系统可对288×6台织机进行连续监测。1986年荣获安徽省科学技术进步三等奖和省首届微机技术应用二等奖。

系统采用微机—单板机分布系统方案,信号传输采用电流驱动矩阵、译码方式,软件采用模块结构内存覆盖技术。本织机监测系统是一个完整的实时数据收集、存贮处理、检索报告系统。能实时采集五种基本工艺参数,即:布长、经停、纬停、计划停、总停,并可根据用户的需要随时屏幕显示或打印出十三种生产统计报表,及时地显示出车位、布长及档车工的生产产量。

淮北第一棉纺厂全面使用本系统后,经济效益明显,织机效率提高4%以上,台机利用率由原来的75%提高到90%,同时质量亦明显提高。

### HZJ—01微机复式计费系统

HZJ—01微机复式计费系统于85年8月由安徽计算中心,安徽省邮电科研所、青阳县邮电局共同研制,87年9月通过安徽省邮电管理局技术鉴定。88年荣获安徽省科技进步四等奖。

该系统功能和技术指标完全符合邮电部电总(1985)电交字122号文件规定,并在计费准确率方面超过部颁标准,为我省首创,在同类产品中找到国内先进水平。

系统功能:

1. 能对HJ905、906交换机每个用户的每次通话进行复式计费,最大容量800门,月底累加,按农、市话分别打印出各用户的收费清单;
2. 查看、修改所有用户当月、当日累计值;
3. 随时增删电话用户号码,修改用户类别、编号、杆距等;
4. 在需要时,可在约定时间内对某一用户通话情况进行跟踪(即可打印主叫、被叫号码、用户类别、通话时间、次数等);
5. 能提供四种方式、二种类型的清单打印功能,可供操作者根据不同情况选择;
6. 具有故障报警和超时报警功能,并能显示故障类型;
7. 系统采用主、备机方式,主备机间有手

动和自动倒换功能;

8. 具有外存贮器,方便地将内存中的数据转存到外部存贮上,并能在需要时把数据装入内存。

该系统已在安徽省青阳县邮电局使用,完全符合邮电部标,其复式计费设备准确率 $\geq 99\%$ ;市、农话网复式计费准确率 $\geq 97\%$ ,帐务处理准确率 $\geq 99.9\%$ ;复式计费时间精度高于6秒。对促使用户负担合理,提高市话机械利用率和电话接通率将起到明显的作用,对提高邮电通信的社会效益和企业自身效益显著,可在装有HJ905,906型交换机的局、所推广使用。

### 微机档案管理系统

微机档案管理系统是省科委下达的“六五”科技攻关项目——“安徽省科技信息汉字网络系统”中的子系统,于八六年十一月通过安徽省科委技术鉴定会,荣获八七年度省科技进步四等奖及八七年国家档案局科技进步四等奖。

该系统在国家优选高档16位ALTOS—586/40多用户微机上生成,在XENIX操作系统及Skell支持下工作,以及INFORMIX数据库管理系统,ACE报表生成语言组成;系统采用了模块化结构设计技术,功能齐全,有主控六种,检索二十三种,编目六种,统计四种以及打印报表等;具有中文提示,三级菜单方式的友好接口,操作简便,易于掌握,符合国家档案标准和现行档案各项规范,有良好的实用性和通用性,系统建立在多用户机上,可由不同处(科)室多个终端进行操作使用,适用于档案的多用户资源共享,便于推广。在省内属于首创,某些方面具有国内先进水平。

该系统投入运行后,稳定可靠,正确无误,为档案现代化管理提供了新的途径。查全率和查准率均达100%,速度快:查询一次档案卡片最短时间约2秒,效率高:打印一张档案编目表约三分钟,比手工提高十几倍。先后有江苏、上海、浙江等省以及本省马鞍山市、省林业厅有关局共四十多单位,一百五十余位档案管理干部和计算机专业人员前来参观学习,还先后接待了安徽大学、合肥联合大学的多批学生实习。

### WZK烟叶发酵微机过程控制系统

WZK烟叶发酵微机过程控制系统,于一九八六年十一月通过安徽省科委的技术鉴定。本系统主要用于烟叶发酵过程自动控制,采用了预置工艺曲线,实时检测参数,应用逐点比较法实现温、湿度参数按工艺曲线要求变化,系统应用于烟叶发酵过程,该程序控制系统已达到国内先进水平,是省内首创,八七年荣获安徽省科技进步三等奖。

控制系统由SC—801单板机、DDZ—Ⅰ型仪表和热电阻传感器等组成闭合过程控制,该系统具有下列功能:

1. 在给定的参数范围内,通过键盘可任意修改温、湿度,发酵时间及升温速率。

2. 系统对温度、相对湿度、发酵时间及包心温度平均值等参数,能实现数字显示并自动记录曲线,供工艺分析。

3. 在不增加设备情况下,系统有三种工作方式:(1)计算机过程控制;(2)计算机监测人工点动控制;(3)仪表监测人工点动控制。

4. 系统控制精度:(1)温度范围:0~80°C精度:升温 $\pm 2^\circ\text{C}$ ;恒温 $\pm 1^\circ\text{C}$ 。(2)湿度范围:20%~80%;精度:升温 $\pm 5\%$ ;恒温 $\pm 2\%$ 。

(3)升温速率:40°C/天。

5. 自动诊断和报警:(1)硬件故障诊断:对检测线路的短路和开路,系统能自动诊断并报警。(2)超限报警设定:当温度值 $\geq 60^\circ\text{C}$ ,超温报警;湿度值 $\geq 80\%$ ,超湿报警。

系统投入运行后实践证明,烟叶的色、香、味比人工控制提高两个等级,发酵室利用率提高20%,发酵时间缩短1/3,并大大地改善了工人劳动条件。年均增收24万元,经济效益显著,开发前景美好。

(顾福良)