



● 一九九一年 ● 总期第77期

8

# 電子

ISSN 1000-1077

744

544

## 與

# 電腦



• ELECTRONICS AND COMPUTERS •



防水、防塵、防靜電、抗磨損，數據永存！



永

存牌電腦軟磁盤係柯達公司產品。因采用杜幫公司專利聚四氟乙稀特殊塗層和專門的防靜電外套，具備了電腦用戶意想不到的功能：

水、咖啡、指印以及塵、渣等一揩即去，數據秋毫無損；  
即便軟驅磁頭老化，亦不會因磨擦造成數據流失；  
拒危險的靜電於門外。

為保存您珍貴的數據，請選用永存牌軟磁盤

總經銷：深圳科技商場電腦外設部  
地 址：深圳市深南中路農墾大廈二層  
電 話：368778—26. 357774 356464  
聯係人：林元莊  
郵 編：518031



● 国内代号：2 888 定价：0.95元



# 家庭电脑教学键盘+BS卡 使游戏机变成了家用电脑

## KB200 电脑教学键盘 学习电脑知识的良友

### 四大功能：

- 可进行 FBASIC 语言程序设计。
- 输入曲谱即可自动演奏的电子琴功能。
- 存储有 104 个动画图案，有丰富的绘图功能。
- 具有计算器功能，可直接进行四则运算。
- 英文打字练习功能。

### 让游戏机变成通用型家庭电脑

#### 本编辑部

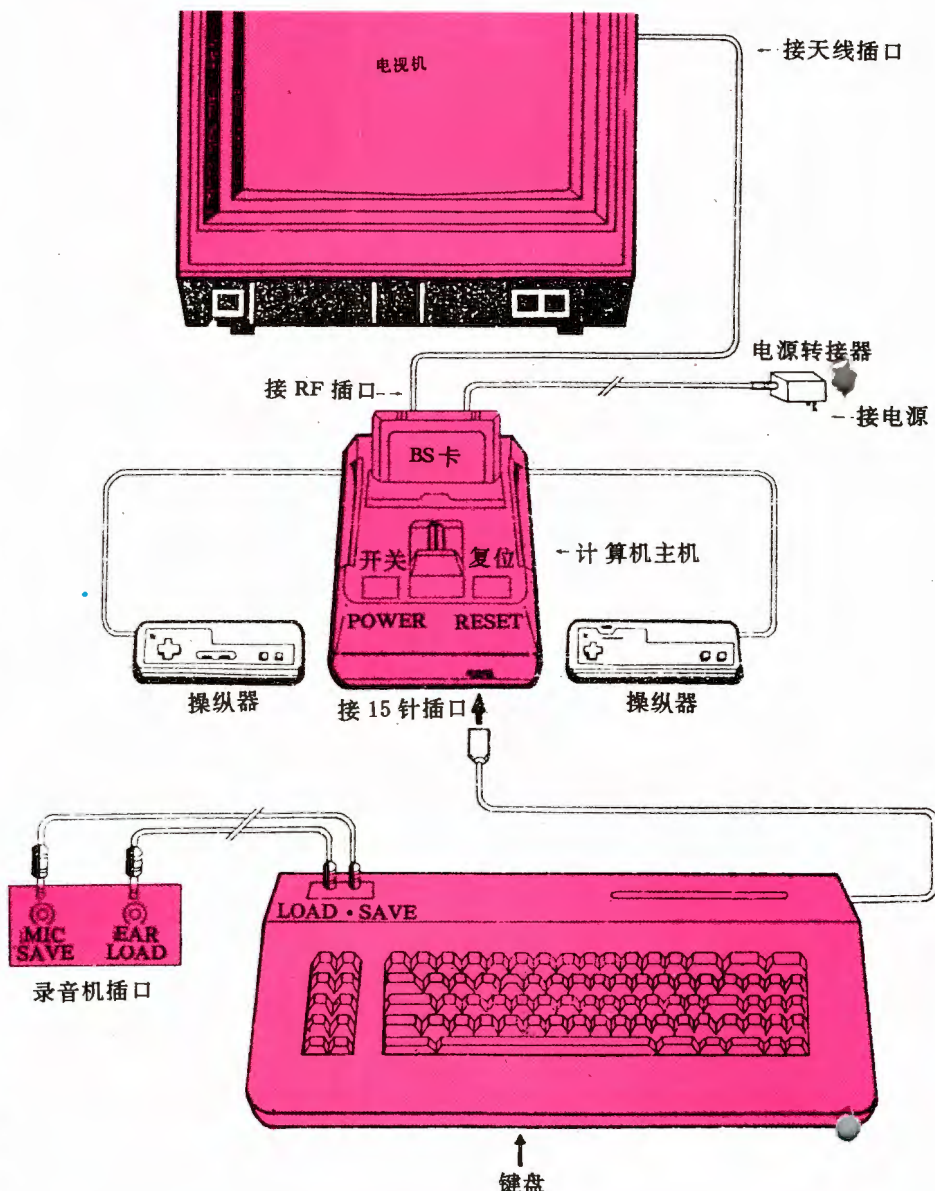
近二三年来，随着游戏机热的兴起，游戏机每年以近百万台的速度，进入普通百姓家。由于游戏卡娱乐性很强，对孩子的吸引力很大，但多数家长又怕孩子贪玩而影响学习，为此，国内许多有识之士认为，我们自己应该发展自己的游戏机工业，开发适合我国国情的游戏卡和游戏机，积极引导消费寓教育于游戏机之中。

机电部第46研究所，成立了天津新星电子公司专门生产SY-700，SY-9100等任天堂型号的游戏机及光线枪，各种教育类及娱乐类型的游戏卡。同时，设计制作了KB200电脑教学键盘。可以插到任何一种游戏机主机上，同时再插上BS卡，就可以学习使用BASIC语言进行程序设计，开发应用程序，这样一来，孩子既可以玩游戏机又可以学习电脑的应用知识，可以设想一下，如果能有上万，几十万甚至上百万个孩子从小能从游戏中学会电脑，这对促进我国科技事业的发展，是一种多么巨大的能量啊！

《电子与电脑》作为电脑方面的普及杂志，我们愿意为使成千上万个游戏机迷，成为电脑迷而效力。

初步计划，年内我们将用一定的篇幅介绍游戏机电脑的专用FBASIC语言及其编程方法，当条件成熟时，和厂家共同组织有奖竞赛活动。

希望大家支持这项活动，多撰稿，为繁荣我国的游戏机事业共同努力！



生产厂家：机电部 46 所新星电子公司

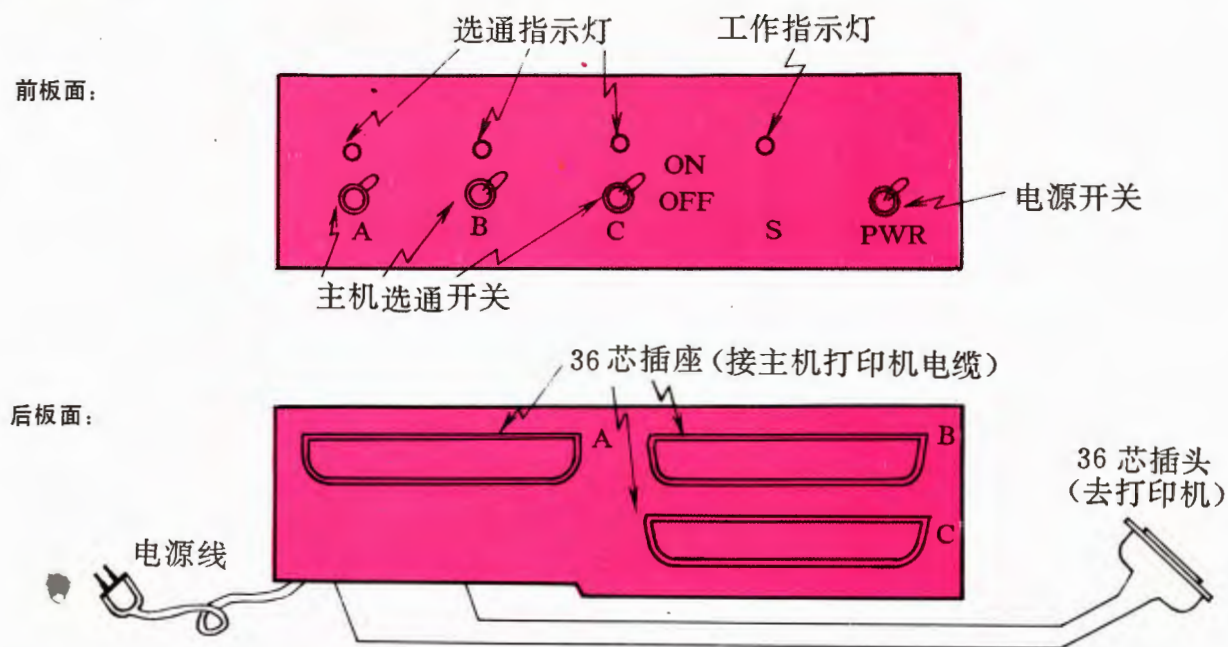
地址：天津新技术产业园区

电话：318342 邮编：300192 天津 55 号信箱

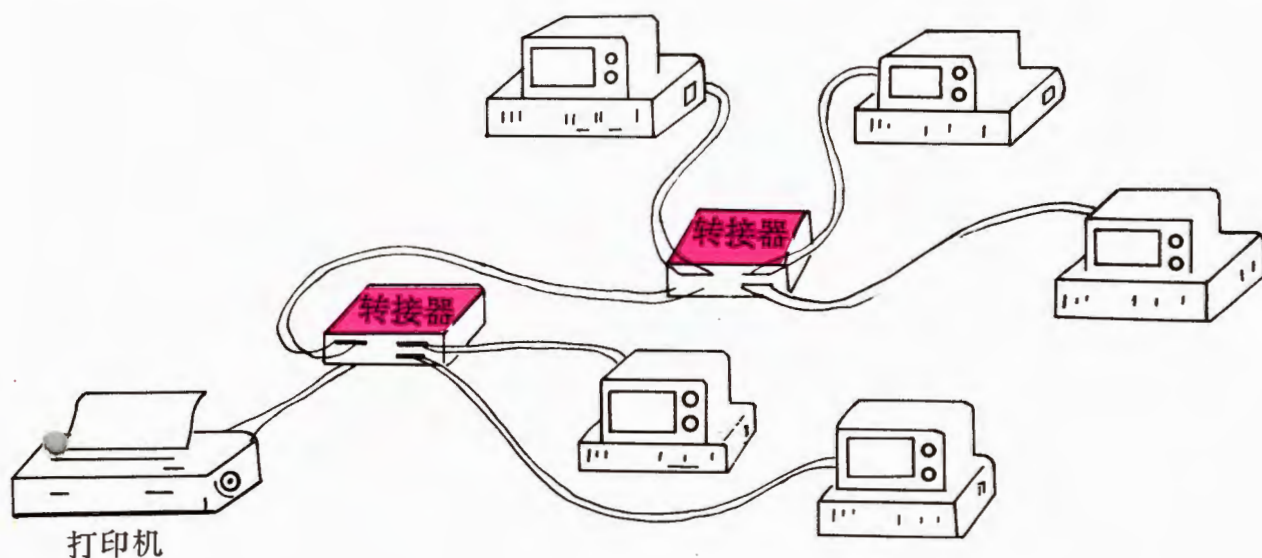
# PUC打印机一并三转接器使用说明

1. 功能：本转接器在单台使用时，可由三台主机共享一台打印机，也可采用二级级联方式；使用两台转接器，可由五台主机共享一台打印机；使用三台转接器，可由七台主机共享一台打印机；使用四台转接器，可由九台主机共享一台打印机。

## 2. 板面示意图：



3. 级联示意图：将去打印机的插头，接至另一台转接器的主机 A 或主机 B 或主机 C 的任意一端即可达到级联目的，以两台转接器级联五台主机共享一台打印机为例，示意如下：







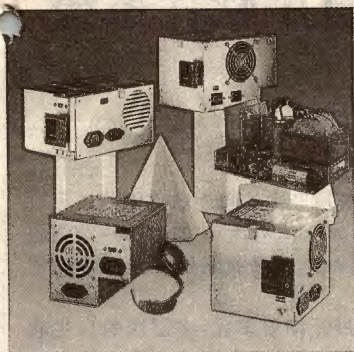
# 電子 與電腦

# 844

一九九一年

总期第77期

# 電子與電腦



ELECTRONICS AND COMPUTERS

## 目 录

### • 综述 •

永不过时的电脑软件—自生软件系统

..... 郑佑(2)

### • PC 用户 •

通用下拉式菜单的实现方法 ..... 张志远(3)

如何改变数据记录在库中的绝对位置 ..... 李海飞(4)

dBASE III 过程文件加密的发现 ..... 刘人伟(5)

对 SET CARRY ON 的一点补充 ..... 李志刚(5)

IBM PC/XT 磁盘扇区读取方法分析 ..... 吴兆熊(6)

使用设备驱动程序 ANSI.SYS 扩充屏幕和键盘

..... 曹长英 李杰(8)

用 PC TOOLS 修改软件的汉字提示 ..... 李开新(10)

巧用批命令定义功能键 ..... 邢观峰(10)

CCDOS 下屏幕死锁的解决 ..... 成刚 杨明秋(11)

VDISK 的妙用 ..... 叶炯(11)

### • 学习机之友 •

谈谈 FORTH ..... 丁志伟(12)

再谈特殊 6502 指令 ..... 李永前(13)

扩展监控系统命令 ..... 蒋建一(14)

MZ-731 机绘制的财务计划执行分析报表

..... 黄浩珍(14)

机器语言程序计算阶乘 ..... 易鹤铃(17)

ProDOS 磁盘操作系统入门(续) ..... 廖凯(18)

用堆栈思想处理任意 N 个元素的排列组合问题

..... 索素文(20)

简单有趣的解密办法 ..... 张亨(22)

### • C 语言初阶讲座 •

第七讲 结构数据(一) ..... 李文兵(22)

### • 学用单片机 •

单片机汉字打印驱动程序的设计思想 ..... 尹征(25)

### • 学装微电脑 •

微电脑控制跨接弯制机 ..... 易齐干(29)

### • 电脑巧开发 •

如何用 EPROM 读写器对 EEPROM 编程

..... 闫怀兵、陈立山(35)

LASER-310 机加接数字音响发生器 ..... 王建峰(36)

### • 维修经验谈 •

任天堂系列游戏机检修实例(续) ..... 林华春(38)

“8088 系统板测试卡”对系统板总线部分的诊断测试

..... 梁杰熙(39)

一九九一年计算机初级程序员有奖竞赛试题 ..... (41)

一九九一年计算机初级程序员有奖竞赛试题答卷

..... (48)

### • 新书与软件 •

给 TANGO 软件增加汉字元件库 ..... 黄健、陈军(47)

机械电子工业部电子工业出版社主办

编辑、出版:《电子与电脑》编辑部

(北京 173 信箱 邮政编码:100036)

印刷:北京三二〇九厂

国内总发行:北京报刊发行局

国内统一刊号:CN11-2199

邮发代号:2-888

国外代号:M924

出版日期:每月 23 日

主编:王惠民 副主编:王昌铭

责任编辑:施玉新

订购处:全国各地邮电局

国外总发行:中国国际图书贸易总公司

(北京 399 信箱 邮政编码 100044)

广告经营许可证:京海工商广字 147 号

定价:0.95 元



電子

ISSN 1000-1077

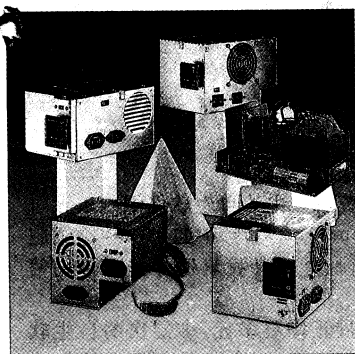
與電腦

844

一九九一年

总期第77期

# 電子與電腦



• ELECTRONICS AND COMPUTERS •

## 目 录

### • 综述 •

- 永不过时的电脑软件—自生软件系统 ..... 郑佑(2)

### • PC 用户 •

- 通用下拉式菜单的实现方法 ..... 张志远(3)  
如何改变数据记录在库中的绝对位置 ..... 李海飞(4)  
dBASE III 过程文件加密的发现 ..... 刘人伟(5)  
对 SET CARRY ON 的一点补充 ..... 李志刚(5)  
IBM PC/XT 磁盘扇区读取方法分析 ..... 吴兆熊(6)  
使用设备驱动程序 ANSI.SYS 扩充屏幕和键盘 ..... 曹长英 李杰(8)  
用 PC TOOLS 修改软件的汉字提示 ..... 李开新(10)  
巧用批命令定义功能键 ..... 邢观锋(10)  
CCDOS 下屏幕死锁的解决 ..... 成刚 杨明秋(11)  
VDISK 的妙用 ..... 叶舸(11)

### • 学习机之友 •

- 谈谈 FORTH ..... 丁志伟(12)  
再谈特殊 6502 指令 ..... 李永前(13)  
扩展监控系统命令 ..... 蒋建一(14)  
MZ-731 机绘制的财务计划执行分析报表 ..... 黄浩珍(14)  
机器语言程序计算阶乘 ..... 易鹤铃(17)  
ProDOS 磁盘操作系统入门(续) ..... 廖凯(18)  
用堆栈思想处理任意 N 个元素的排列组合问题

- ..... 索素文(20)  
简单有趣的解密办法 ..... 张亨(22)

### • C 语言初阶讲座 •

- 第七讲 结构数据(一) ..... 李文兵(22)

### • 学用单片机 •

- 单片机汉字打印驱动程序的设计思想 ..... 尹征(25)

### • 学装微电脑 •

- 微电脑控制跨接线弯制机 ..... 易齐干(29)

### • 电脑巧开发 •

- 如何用 EPROM 读写器对 EEPROM 编程 ..... 闫怀兵、陈立山(35)  
LASER-310 机加接数字音响发生器 ..... 王建锋(36)

### • 维修经验谈 •

- 任天堂系列游戏机检修实例(续) ..... 林华春(38)  
“8088 系统板测试卡”对系统板总线部分的诊断测试 ..... 梁杰熙(39)  
一九九一年计算机初级程序员有奖竞赛试题 ..... (41)  
一九九一年计算机初级程序员有奖竞赛试题答卷 ..... (48)

### • 新书与软件 •

- 给 TANGO 软件增加汉字元件库 ..... 黄健、陈军(47)

机械电子工业部电子工业出版社主办

编辑、出版:《电子与电脑》编辑部  
(北京 173 信箱 邮政编码:100036)

印刷:北京三二〇九厂

国内总发行:北京报刊发行局

国内统一刊号:CN11-2199

邮发代号:2-888

国外代号:M924

出版日期:每月 23 日

主编:王惠民 副主编:王昌铭

责任编辑:施玉新

订购处:全国各地邮电局

国外总发行:中国国际图书贸易总公司

(北京 399 信箱 邮政编码 100044)

广告经营许可证:京海工商广字 147 号

定价:0.95 元



# 永不过时的电脑软件——自生软件系统

郑 佑

(WORLD-WIDE INSTITUTE OF VALUATION, U. S. A.)

一个具有永久生命的软件系统(称为自生软件系统)已在近期研制成功,它是基于一个软体自动化的新概念,其目的在于将目前所有的软件都转化为一种永久的形式——这种形式将使软件永远不会过时或随着电脑科技的新发展而被淘汰。以这种形式写出来的软件一旦完成,就永远不需要重写,因为电脑可以自动将它们修正或改进。

目前,软件是寄生于电脑技术。例如,当电脑硬件系统,操作系统或语言系统改变时,在这些系统环境下写出来的软件系统就会被淘汰。一般来讲,所有现在的软件,迟早都是会随着电脑技术的不断变化而被淘汰,这一点是不容置疑的。

在这种永不过时的软件系统中,所有的软件都是由电脑生成的,都可以在每一次再生成的过程中,自动的由电脑来解释和修正。在技术上,软件的重新生成是由向软件生成器输入一种特殊的描述问题特性的信号文件。此信号文件就是键盘输入的直接记录,软件生成器可以直接认识这种信号文件,因而也可以自动地将它所代表的软件加以修正。软件生成器和信号文件生成器本身也都必需有它们自己的信号文件,因而也可以被重新生成。因此,所有的在这一永不过时的软件系统中的软件都是由电脑生成,它们都有描述它们自己的信号文件。

这种自我生产的软件系统与生物细胞的结构极为相象,为电脑设计的基本原理提供了一个新的概念:以自我生产(简称为自生)细胞科技为基础设计新型电脑。能够自我繁殖的生物细胞同时具备许多优点,自生细胞科技也同时具有相类似的优点:

1. 目前大多数的软件都是由专业程序员直接用电脑语言写出来的,而在自生软体系统中,所有的程序(正如生物细胞中的蛋白质)都是经向电脑中的软件生成器(正如细胞中的蛋白质生成器,如核糖体)输入一种特殊的描述问题特性的信号文件(正如细胞中的脱氧核糖核酸,DNA)而产生的。目前的电脑系统是一种不成熟的粗糙的产品,因为在最初发明时,只想到把产品尽快的造出来,而结果是在这种系统下产生出来的软件,就象是直接制造出的很容易腐化的蛋白质,由于没有生命力而被淘汰。

2. 上述的那种特殊的描述问题特性的信号文件,正如细胞中的脱氧核糖核酸(DNA)有永远生存的特性,它不但可以用于任何电脑系统,而且不受任何将来电脑科技新发展的影响。由于这种信号文件是使用者与电脑用人类语言对话的方式所产生的。因此所有只要受过一般教育的人都能够使用电脑,以上二点使电

脑的使用范围普遍增加。

3. 由自生软件系统所产生的自生软件细胞单元可以代替电子神经网络(neural network)中的电子神经细胞(neuron),这种神经细胞可以自我繁殖并相互传递信息。

4. 人工智能的发展正需要多元,多层次的自生软件细胞系统,人工生物(Artificial life)也需要自生细胞科技的发展。

5. 现今的电脑发展已经到了一个相当的程度,应该有足够的适应性,或者说是复杂性,没有任何理由使电脑与其使用者(包括程序员在内)在脱离人类语言的情况下相互隔绝。自生软件系统将使高级电脑语言全部作废。

6. 目前的电脑结构是一个初期,草率而仓促(从自然创造生物的眼光来看)的设计结果。就连它的发明者之一 John von Neumann,也认为不满意,在他生命的最后几年中,一直致力于研究一个和生物细胞类似的系统。

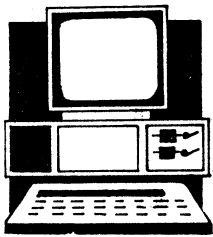
7. 一个可以由最初的软件自我生成同样软件的示范系统已经设计成功。

大多数目前的科技都是由简单部份,结合构成复杂系统。而自生细胞科技的主要概念是,从包含一切抗能和所有基本知识的复杂的细胞元开始。自我生成同样复杂的新细胞,新细胞可以在生成时改变它的性能来适应各种新的环境的需要。

这一再生细胞科技将是自动化的最终形式。它是解决和控制高复杂性问题所必不可少的。同时也是生命的特征。细胞观念使多元化细胞设计成为可能。细胞的再繁殖使程序的复杂性成几何级数增长。正如人类的大脑可增加  $10^{12}$  的复杂性,如果没有自我繁殖的能力,如此巨大的数字是不可能用人工制造出来的。

用科技术语来讲,生物生命也可以被称作是自动化的自生细胞科技。可能正是由于它的过份复杂,使得外来的全面控制成为不可能,而最终的控制就让给此细胞为基础的生命体自己掌握。从技术上来讲,生命是由天文量级复杂程度的程序完成的,是真正最先进的科学科技,尽管至今为止,还没有人能够将此程序破译。现在二十世纪的无生命科技,落后于生命科技至少几千年。不过,第一个自生软件系统的产品正在设计、计划于明年问世。

今后,软件系统将分成二种:一种是永不过时的,如自生软体系统,另一种是迟早会被淘汰的,如目前大多数的软体系统。照理来讲,所有现在非自生的以及将来的软体系统将都得改写成或写成自生软件系统。



PC 用户

# 通用下拉式菜单的实现方法

山东计算机服务公司临沂分公司 张志远

一个设计良好的菜单不仅富有艺术情趣,令人耳目一新,而且能减轻用户记忆负担,减少错误操作,提高效率。通常的菜单设计方法是根据编号或字母选择相应的功能,使用者要全神贯注于屏幕信息,寻找按键的位置,容易出错。近来流行的菜单形式有下拉式、上弹式及开窗口等,较好的解决了上述问题。它使用移动键上下左右移动光棒,光棒所在的位置即要选择的菜单项,按回车后进入该项功能,使用方便,不易出错,且保留主菜单画面,子菜单以下拉对应方式层层进入,层次分明。这种菜单虽然好,但大多是在工具软件中采用汇编语言或其它语言编写,一般用户不熟悉。笔者通过多次实践,在 Foxbase+2.0 和 DBASE III 下,实现了下拉式菜单的设计,现介绍给大家参考。

## 1. 主菜单(程序清单见 CD1·PRG)

主菜单采用在屏幕的顶部一行显示各菜单项,其中有一菜单项的底色与其它菜单项不同,并进行闪烁。这一底色被称作光棒,此光棒可以通过按键左右移动,按回车键即选定了光棒所在的菜单项。调用方法:

主程序将各菜单项的内容赋给变量 G1~Gn;

定义 Z 为全局变量;

调用 DO CD1 WITH 参数 1,参数 2,参数 3,参数 4(参数 1:显示行、参数 2:起始列、参数 3:列距、参数 4:列数);

根据变量 Z 的值进入相应的功能子程序(其中:Z=" "退出,Z="1"菜单项 1,Z="2"菜单项 2,...Z="n"菜单项 n)。

## 2. 子菜单(下拉式菜单 CD2·PRG)

主菜单选定后,便进入相应的子菜单,子菜单的列座标与相应的主菜单项对齐,且采用下拉方式,保留主菜单。光棒可通过按键上下移动或退出,回车即选择。只是 DO 语句中参数的意义不同,参数 1:起始行;参数 2:起始列,参数 3:行距,参数 4:行数。

另外需要说明的一点,此通用菜单程序,根据需要变换行坐标,可变成通用上弹式菜单程序。

\* CD1.PRG 主菜单程序

PARAM A,B,C,D

\* A:显示行,B:起始列,C:列距,D:列数

SET COLO TO W+/R+

@24,0 SAY"提示:左移←右移→选择 Enter 退出 Esc"

SET COLO N/GR+

N=1

L=B

DO WHILE N<=D

X=LTRIM(STR(N,3))

@A,L SAY G&X && 显示各菜单项

N=N+1

L=L+C+LEN(G&X)

ENDD

L1=L-LEN(G&X)-C

X="1"

L=B

DO WHILE .T.

SET COLO TO W+\*/B+

@A,L SAY G&X && 显示被选菜单

X1=X

L1=L

W=INKEY(0) && 返回击入的键值

SET COLO TO W+/B+

DO CASE

CASE W=19 &&←左移

X=IIF(VAL(X)=1,STR(D),STR(VAL(X)-1,3))

X=LTRIM(X)

L=IIF(X=LTRIM(STR(D)),L1,L-LEN(G&X)-C)

CASE W=4 &&→右移

X=IIF(VAL(X)=D,"1",LTRIM(STR(VAL(X)+,3)))

L=IIF(L="1".AND.LEN(X)=1,B,L+LEN(G&X1)+C)

CASE W=13 &&Enter 选择

@A,L SAY G&X

Z=X

EXIT

CASE W=27 &&Esc 退出

Z=" "

EXIT

ENDD

SET COLO TO N/GR+

@A,L1 SAY G&X1 && 恢复上次被选菜单

ENDD

RETU

\* CD2.PRG 子菜单(下拉式)程序

PARAM A,B,C,D

\* A:起始行,B:起始列,C:行距,D:行数

SET COLO TO R+/G+

@24,0 SAY"提示:上移↑下移↓选择 Enter 退出 Esc"

SET COLO TO R+/W+

N=1

DO WHILE N<=D

X=LTRIM(STR(N,2))

@A+(N-1)\*C,B SAY G&X && 显示各菜单项

N=N+1

ENDD

L=A

DO WHILE .T.



```

SET COLO TO GR + * /N
X=LTRIM(STR((L+C-A)/C,2))
@L,B SAY G&X    && 显示被选菜单
X1=X
W=INKEY(0)    && 返回击入的键值
SET COLO TO N/W+
DO CASE
CASE W=5    && ↑ 上移
L=IIF(L=A,(D-1)*C+A,L-C)
CASE W=24    && ↓ 下移
L=IIF(L=(D-1)*C+A,A,L+C)

```

```

CASE W=13    && Enter 选择
Z=X
EXIT
CASE W=27    && Esc 退出
Z=" "
EXIT
ENDC
SET COLO TO R+/W+
@(&X1-1)*C+A,B, SAY G&X1    && 恢复上次被选菜单
ENDD
RETU

```

## 如何改变数据记录在库中的绝对位置

河北涿州市物探局研究院软件所 李海飞

dBASE III 提供的 'COPY TO <FILENAME>STRUCTURE EXTENDED' 命令可以用来把当前库文件的各个域变成 <FILENAME> 文件中的记录。<FILENAME> 包括 FIELD-NAME, FIELD-TYPE, FIELD-LEN, FIELD-DEC 四个域。对我们来说, FIELD-NAME 最为重量。通过对 <FILENAME> 的记录循环可以达到对当前库文件的整个域进行操作的目的。

再利用 dBASE III 的宏替换功能, 可以使本程序具有通用性。在编程过程中还用到了临时库文件, 如 TempFile.dbf, sef.dbf, 用完以后即把它们删除了。

程序如下:

```

set heading off
set talk off
con=.T.
* con 为控制循环的逻辑型变量
do while con
clear
accept '请输入数据库文件名:' to Processfil
FName=Processfil+'.dbf'
if .not. file(&FName)
    ? '数据库文件不存在, 返回'
    return
endif
sele 1
use &FName
* &FName 为需要处理的数据库文件
copy to sef structure extended
sele 2
use sef
* sef.dbf 为 &FName 的结构伸展文件
sele 1
accept '要浏览整个数据库吗(Y/N)?' to Ans
if upper (Ans)='Y'
list all
endif
input '请输入准备转移的记录的记录号:' to SouRec
goto SouRec

```

```

copy next 1 to TempFile
sele 3
use TempFile
* TempFile.dbf 为只有一个记录的临时文件
sele 1
input '请输入转移到的记录号:' to DesRec
goto DesRec
if SouRec<DesRec
    insert blank
* 若源记录号小于目的记录号, 则在目的记录后面插入
else
    insert blank before
* 若源记录号大于等于目的记录号, 则在目的记录前面插入
endif
sele 2
go top
do while .not. eof()
* 通过对结构伸展文件 sef.dbf 的循环来达到对 &FName 文件循环的目的
FN=FIELD-NAME
sele 1
replace &FN with c->&FN
sele 2
skip
enddo
sele 1
if SouRec<DesRec
dele record SouRec
else
dele record SouRec+1
endif
pack
accept '还要继续吗(Y/N)?' to Ans
if upper (Ans)='N'
    con=.F.
endif
use
sele 2

```

```

use
erase sef. dbf
) sele 3
use
erase TempFile. dbf
enddo
set talk on
set heading on
return

```

## dBASE III 过程文件 加密的发现

大港石油管理局炼油厂 刘人伟

我厂现用由电子工业部第六研究所 1988 年 10 月出版的计量管理软件包《CMIS V1.0》，当数据库录入到三千五百以上时，再由此库进行其它方面的管理和应用，结果发现软件包的运行速度越来越慢，且出错率逐渐增加，造成部分数据录不进去或串库，以及数据行缩小等，使得打印出的台帐格式不齐，数据丢失。虽然该软件包已经采用 CCDOS 4.0 系统来提高运行速度，但由于受到 dBASE 1.0B 版本的限制，无法正确完成计量管理的大量工作，使该软件内的功能、速度等均受到很大限制。例如：机内每做一项工作的进入和退出，都需要很长时间方可完成，从而有可能造成不必要的错误。因此要经常对数据库进行备份处理，以保证不因意外而发生错误。这给正常工作带来很多麻烦和困难，及要求用户要了解该软件包中部分数据的作用，否则要整盘拷贝（对所有数据库），对一般用户来说有一定难度。为了提高软件的速度和可靠性，以充分利用我厂现有 AST286 机的特性，我把《CMIS》软件包在 FOXBASE+2.0 上试运行，结果发现该软件包上的过程文件有局部地方加密，而无法利用 FOXBASE+ 的高速，高容量等特性。

在对此 dBASE III 过程文件解密时，我首先试用 EDLIN 对加密文件内容进行修改，但未成功。原因是加密的行语句不能显示而无法修改，若进行重新输入编辑此行，却出现该行语句重合叠加等，而无能为力。又试用 Wordstar 等文本编辑软件解密也无效。虽然调用 dBASE III 的过程文件编辑程序，进行有密语句重写存盘，可以解决。可要把上百个过程文件中的加密文件进行修改，工作量很大很烦琐。但解密是可行的，同时发现此加密方法为行语句尾加密。

针对行语句尾加密，我首先利用 DEBUG 对其中一个有密文件 XB05. PRG 进行反汇编，找出正常语句行尾与有密语句行尾的区别。结果发现正常无密语句行尾标记为 0D0A（回车、换行），而有密语句行尾标记为 0D8A，所以做相应修改即可完成。

经修改后的 XB05. PRG 文件，可用 Wordstar、

EDLIN 进行文本编辑修改，也可以完整地输出程序清单。同时，能顺利地 Foxbase+ 上运行，只要把 Foxbase+ 上的 config. FX 做适当修改即可。

通过上述发现与修改，我决定用 Pctools R1.0 软件，对整个 CMIS 软件的过程文件进行全面解密。使用 Pctools 的 search 功能，对过程文件进行有密处查找修改，找出含有十六进制码 0D8A 处，把 8A 的 8 改为 0，即可完成该语句行的解密工作。此方法简单易行，任何人都能在短时间内，对上百个加密文件进行统一的解密工作。在解密时，发现有个别行语句行尾为 0D73，这些在解完 0D8A 后，再对 0D73 用同样的方法，把它改为 0D0A 即可。

对 CMIS 软件解密后，再在 Foxbase+2.0 上运行，收到良好效果，使 CMIS 软件包运行速度更快，更理想，也解决了由于 dBASE III 所带来的软件不足，从而保证该软件的正常运行。解了密的 CMIS 软件包，还可以在 dBASE III Plus 上运行，运行结果也很理想。

上述加密方法也可用于自编的 dBASE III 过程文件中，以防止他人修改。只要用 DEBUG 或 Pctools 等工具软件把 dBASE III 的过程文件的行语句尾 0D0A 中的 0A 改为其它十六进制数码即可。

若把 0D0A 改为 0D1A，（1A 为文件结束符 ^Z 的代码），该被加密的语句以后内容，再也不能通过 type 和文本编辑（包括 dBASE III 的文本编辑）等方法看到和修改原程序清单，所以此法更能起到保密和保护程序作用。上述方法再结合 DOS 和 dBASE III 的其它加密方法，能达到更理想的加密效果，而不被他人所利用。

## 对 SET CARRY ON 的一点补充

常德科技情报研究所 李志刚

人们在用 dBASE III 编制程序时，为了加快数据输入速度，常使用 SET CARRY ON 命令来使输入当前记录时携带上一记录。然而人们不曾注意到，假如上一记录是做了删除标记“\*”的，那么携带时，删除标记被带上，而且若没有经过 PACK 命令压缩，则以后增加的记录均被带上删除标记。因使用 PACK 命令占时较长，在平时删除时，一般不会即时使用 PACK，而是待有了一定的删除标记记录时，才集中整理数据，所以稍不注意使用 PACK 命令，就有可能引起大量数据的丢失，而让人感到莫名其妙，因此本人提出了注意的方法，供参考。即：在使用 APPEND 命令，并输完当前记录后，紧补上 RECALL 命令，以唤醒当前记录，因不管记录有无删除标记，RECALL 命令均将其变为无删除标记，这样即避免了误删记录的可能性。



# IBM PC/XT 磁盘扇区读取方法分析

重庆渝州大学计算机教研室 吴兆熊

系统结构研究及程序编制人员,常有必要读取某些程序的机器码及反汇编文本。由于保密、覆盖、不常驻内存等因素,要从内存直接获得完整的反汇编文本往往很困难。这种情况下,只好直接从磁盘上按文件的扇区分布,把文件完整地读出。要做到这一点,首先须了解 IBM PC/XT 提供的几种读磁盘扇区方法。本文拟通过对磁盘扇区结构和 ROM BIOS 的分析,来阐明这些读磁盘扇区方法的各自特点。

## 磁盘结构及扇区分布

双面软盘驱动器有两只读写磁头,固定在存取臂的头部,各用于对磁盘的一个面进行数据读写。存取臂可沿磁盘径向移动,数据是在磁盘旋转过程中写入磁盘上。所以,对应于读写磁头所在位置的数据记录部分为一个圆圈,这一圆圈就是可记录数据的,称为磁道的轨迹。一个磁道读写完毕,存取臂带动磁头沿盘面径向移到另一位置。因此,在盘面上形成了多条同心圆的磁道。PC/XT 机用的双面倍密度软盘,每面有 40 个磁道,两面共 80 磁道。

硬盘机的结构与上述类似。在 IBM PC/XT 机中常用的是二块盘片 4 个磁头的 10M、20M 字节容量的硬盘机,其结构如图 1 所示。

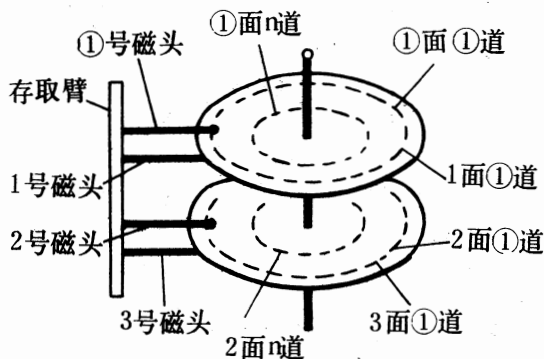


图1 磁盘驱动器结构示意图

标准 10M 硬盘每面有 305 磁道,增强 20M 硬盘每面有 614 磁道。

每一盘面上的磁道均从零开始编号。因为所有磁头均固定在存取臂上,当存取臂停留在某一位置时,所有磁头均驻留在各对应面的同一编号磁道上。如图 1 所示,当 0 号磁头处于 0 面 0 道位置时,其余三个磁头也分别处于对应盘面的 0 道上。我们把各个盘面上的同编号磁道重叠起来,组成一个圆筒,称之为柱面。即

是说,0 号柱面指的是 0~3 面的 4 个 0 号磁道,1 号柱面指的是 0~3 面的 4 个 1 号磁道,等等。这意味着 20M 硬盘具有 614 个柱面。与此类似,双面软盘每一柱面包含 2 个同一编号的磁道,这两磁道组成一柱面,共有 40 个柱面。

双面倍密软盘的每一磁道分为 9 个扇区,每扇区可记录 512 字节数据,一张盘片可记录的数据量为:

磁头数  $\times$  每面磁道数  $\times$  每道扇区数  $\times$  每扇区

记录字节数  $= 2 \times 40 \times 9 \times 0.5K = 360K$

标准 10M 和增强 20M 硬盘,每磁道分为 17 个扇区。每扇区可记录数据量同样为 512 字节。按上述公式,标准 10M 盘共可记录  $4 \times 305 \times 17 \times 0.5K = 10370K$  字节;增强 20M 盘可记录 20876 字节。软盘及硬盘每面的磁道及扇区分布如图 2、3 所示。

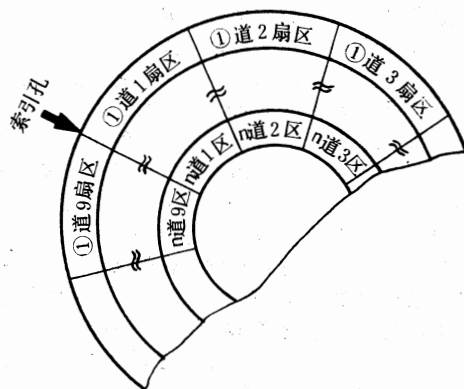


图2 软盘的磁道及扇区分布

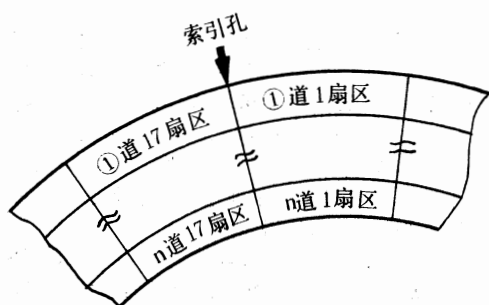


图3 硬盘的磁道及扇区分布

## 磁盘数据读取

在 DOS 系统层,可通过软中断命令 INT 25 读取磁盘扇区数据。在 ROM BIOS 则通过硬盘和软盘各自的软中断 INT 13 读取。三种方法各有什么不同呢?我们先根据前述磁盘扇区结构,来讨论扇区读取方法。

IBM PC/XT 采用两种方法:

### 1. 按顺序扇区号读取。

从磁盘组的 0 面 0 道 1 扇区开始,直到最末一面,末道的最后一个扇区,依序连续编号,按这方式分配给每一扇区的编号,称为逻辑扇区号。要读某一扇区,只须指定该扇区的逻辑扇区号即可。DOS 系统中的软中断 INT25,就是用的这种读取方式。这时,利用 DX 来指出准备读的起始逻辑扇区号,CX 指出须读出的扇区数目。最大可读出扇区数不超过 7EH。执行 INT 25 时。

磁道号	0		1		...	39	
磁头号	0	1	0	1	...	0	1
物理扇区号	1...9	1...9	1...9	1...9	...	1...9	1...9
逻辑扇区号	0...8	9...17	18...26	27...35	...	702...710	711...719

表 1 360K 双面倍密软盘物理与逻辑扇区号对照表

与软盘不同的是,硬盘将 0 面 0 道的 17 个扇区设为隐含扇区(各型硬盘机的隐含扇区数不尽相同,请参阅有关资料)。因此,硬盘是以 1 面 0 道 1 扇区作为 0 号逻辑扇区,按上述软盘编号方法,依序编号直至最后一个磁道第 3 面的 17 扇区。

### 2. 按扇区所在物理位置读取。

这种方法是按磁盘的面号、磁道号及扇区在该磁道中的顺序号读取,即按图 2、图 3 所示扇区位置读取。PC/XT 机的 ROM BIOS 中有 2 个按这种方式读取扇区的子程序,即软盘 DISKETTE-IO 和硬盘 DISK-IO。它们都是靠软中断指令 INT 13 调用。尽管都是 INT 13,但硬盘和软盘的入口参数存在如下差别:

#### a、软盘 INT 13

该子程序入口绝对地址在 ROM 区的 FEC59H。

(AH)=0~6;(AH)=2 时,执行读磁盘扇区功能。

(AL)=1~9;要读的扇区数。

(CL)=1~9;起始扇区数。

(CH)=0~27H;要读扇区所在磁道号

(DH)=0,1;面号。

(DL)=0~3;驱动器号(0=A 盘,1=B 盘.....)

ES;BX;指向输入缓冲首地址。

调用软盘 INT 13 时,每次只能读出指定磁道内的扇区数。那就是说,(AL)和(CL)中的参数是互相制约的。如 CL 指定该磁道的 7 号扇区为始读扇区;那末,可读扇区数最多只能到该道的末扇区——9 扇区。这时(AL)最大值不能超过 3,否则将出错。

#### b、硬盘 INT 13

入口绝对地址在硬盘 ROM 区的 C8256H。

从 AL 指定的驱动器磁盘上读出的数据,被依次存入由 DS;BX 指定的内存缓冲区内。

这里,有必要介绍一下逻辑扇区号的编排原则。因为所有读写磁头均固定在同一存取臂上,磁道的改变要靠存取臂的机械移动来完成。在同一磁道上改换扇区,只须检索该磁道上各扇区的标志即可;所需时间较移动存取臂短得多。因此,记录数据时,希望把每一磁道上的所有扇区都记录满后,才移动存取臂至下一磁道继续记录,即把每一柱面录满后,才改由下一柱面续录。据此,扇区是按柱面(磁道号)的顺序连续编号。以双面软盘为例,其编号顺序是从 0 面 0 道 1 扇区开始,依序编至第 9 扇区;接下来是 1 面 0 道 1 扇区,然后是 0 面 1 道 1 扇区……。逻辑扇区号的起始扇区编号为零,即 0 面 0 道 1 扇区为 0 号逻辑扇区,如表 1 所示。

(AH)=0~14H;(AH)=0~5 与软盘 INT 13 的调用功能相同。

(AL)=1~80H;要读的扇区数(长读为 1~79H)。

(CL)=1~11H;起始扇区号(CL 的 b5~b0 位)。

(CL 的 b7b6 位)+(CH)=0~3FFH;柱面(磁道)号。

(DH)=0~7H;磁头号(面号)

(DL)=80~87H;驱动器号(80=C,81=D,.....)

ES;BS;指向输入缓冲区首地址。

硬盘 INT 13 的最大可读扇区数与面号、道号及起始扇区号无关。

### 几点说明

1、INT 25 不能读出硬盘的隐含扇区(0 面 0 道 1~17 扇区),因隐含扇区未编入逻辑扇区号序列。为了读出隐含扇区的内容,应该使用硬盘 INT 13,按隐含扇区所在物理位置读出。

2、若只知道扇区号,用 INT 25 读出是十分方便的。例如,根据 FAT 查出了文件分布的逻辑扇区号,直接送入 DX 再调用 INT 25 即可。在这种情况下,若采用 INT 13,则需要把逻辑扇区号换算为物理扇区号。这牵扯到磁盘的规格和参数,十分麻烦,且易出错。因此,只有已知欲读扇区的物理位置,或必需读隐含扇区(如清除硬盘病毒)时,才使用 INT 13。

3、INT 25 的输入缓冲区地址是用 DS 寄存器来指出。驱动器 0~1 号为软盘,2 号为硬盘,而 INT 13 是用 ES 寄存器指示缓存区地址。驱动器 0~3 号为软盘,80~87H 为硬盘。这些细微差别都要特别注意。

4、软盘 INT 13 和硬盘 INT 13 尽管调用号都相



同,但在结构和功能上却有很大的差别。使用时,除了要仔细研究前面介绍的参数或有关资料外,还需要了解你使用的是哪一个 INT 13,否则很容易出错。为此,有必要在这里对机器的启动过程,概略地作一介绍。

机器启动后,执行 ROM BIOS;由子程序 CLR-STG (FE18EH)填入中断矢量。这时,在 0000:004C~004F 单元填入的是软盘 INT13 的入口地址 FEC59H。此后,ROM-SCAN 子程序 (FE518H)检查有无硬盘 ROM 存在。若有,则转至硬盘 ROM (C800:0003)。在此,首先把软盘中断的入口地址值 FEC59H,转移到 INT 40 的位置 0000:0100~0103 单元;而用硬盘 INT 13 取代,把入口地址 C800:0256H 填入到 0000:004C~004FH 单元中。往后,软盘 I/O 功能由 INT 40 完成,软中断指令 INT 13 被用于硬盘 I/O 了。

但是,在系统有硬盘 ROM 的情况下,仍然可利用软中断指令 INT 13 来实现软盘 I/O 功能。因为,在硬

盘 INT 13 子程序一开始,就检查 DL 中的盘号。若 (DL)<80H,则属于软盘 INT 13,转而调用 INT 40 指向 FEC59H,执行软盘 I/O 功能。否则,按硬盘 INT 13 处理。那就是说,用户在 DL 中置入的是软盘号,那末调用的就是硬盘 INT 13;置入的是硬盘号,调用的就是硬盘 INT 13。当然,其余参数也应根据置入的盘号,按前面介绍的范围来选用,否则将会出错。

#### 结束语

在 IBM PC/XT 机中,软中断指令的实质是调用一组专用处理子程序。本文仅对其要点作了一些分析。具体使用中,还要考虑该子程序对某些寄存器及堆栈的破坏等因素。为了不发生错误,最好采取一些措施,如另设一用户堆栈,对受影响寄存器进行保护,有关程序用 DEBUG 验证一下等等。

文中仅对软中断读盘功能作了介绍。按理类推,对软中断写盘功能也能够理解,故不再赘述。

## 使用设备驱动程序 ANSI.SYS 扩充屏幕和键盘

解放军兽医大学计算机教研室 曹长英 李杰

PC DOS 装载的标准设备驱动程序支持标准输入、标准输出、标准打印机、软盘以及硬盘设备,支持键盘和屏幕输出。系统提供的这些常规的屏幕和键盘控制功能是由常驻的 'CON' 设备驱动程序实现的,然而它还有如下不足:

(1)每一键符定义是固定的,无法改变。

(2)屏幕显示类似电传打字机的输出,无法改变光标位置及显示的色彩等。

为此,还提供了—个扩展屏幕和键盘控制的驱动程序,命名为 ANSI.SYS 文件,附加在 DOS 系统盘上。DOS 支持这些设备不需要用户任何命令指定,而是在启动 DOS 时自动把它装入,方法是通过在设置文件 CONFIG.SYS 时,在其中加入一个 DEVICE=<命令>这种形式指定的。在此即:DEVICE=ANSI.SYS。

ANSI.SYS 文件除了保持原常驻的 CON 驱动程序的全部功能外,还扩展了如下的功能:

(1)随意指定光标位置。

(2)控制光标的移动。

(3)清除屏幕。

(4)指定屏幕显示规格。

(5)设置显示色彩参数。

(6)随意对任一键符赋予新的定义。

这些功能虽然只是直接控制显示器硬件功能 (ROM BIOS 的 INT)的一部分,但能确保调用上述功能的应用程序可在所有运行 DOS 系统的机器上执行。

为使 ANSI.SYS 加载内存后,取代常驻的 CON 驱动程序,在编制 ANSI.SYS 程序的设备头时,其设备名

字域内仍取 "CON" 名。这样,当 DOS 开工初始化期间,该驱动程序作为可安装的字符设备驱动程序,被连接在常驻的 CON 程序之前,成为设备链中仅次于空设备 NUL 的第一个字符设备驱动程序。当应用程序对键盘或屏幕请求一个 I/O 操作时,DOS 自然在设备链中,首先匹配有扩展功能的 CON 驱动程序。

下面讨论 ANSI.SYS 具备哪些扩展屏幕和键盘控制的功能。

在 DOS 版本 2 和 3 下,当通过 "DEVICE=ANSI.SYS" 命令加载内存后,任意一个应用程序,在调用 INT 21H 的 01H、02H、06H 和 09H 等四个子功能时,可发出特殊的字符序列控制屏幕光标的位置、设置色彩参数以及对任一键符赋予新的含义。

#### 特殊字符序列的格式如下:

ESC [# 字符;---

其中:(1)ESC 是一个字符的代码(十六进制数是 1B,十进制数是 27,)而不是三个字符---"ESC"。

(2)ESC 之后是一字符串,以[为前导字符。

(3)[之后是#,它表示一个数字参数,若是字符则必须用十进制数表示出其 ASCII 码值。

(4)参数之间用;分隔开。

#### 光标控制功能:

ESC #;H 设置光标位置

将光标移到参数所指定的位置。第一个#指定行号,第二个#指定列号。缺省值为 1,即让光标"回家"(左上角)。

ESC [# A 控制光标上移

光标上移#指定的行数,列位置不变。#缺省为1,即上移一行。若光标已在顶行,则不予理睬。

ESC [ #B 控制光标下移。

光标下移#指定的行数,列位置不变。#缺省为1,即下移一行。若光标已在底行,则不予理睬。

ESC [ #C 控制光标前移。

光标前移#指定的列数,行位置不变。#缺省为1,即前移一列。若光标已在最前列,则不予理睬。

ESC [ #D 控制光标后移。

光标后移#指定的列数,行位置不变。#缺省为1,即后移一列。若光标已在最后一列,则不予理睬。

ESC [ #; #f 控制光标位置

光标移到参数指定的位置。第一个#指定行号,第二个#指定列号。缺省值为1,即让光标“回家”(左上角)。

ESC [ #; #R 报告光标位置

将当前光标位置通过标准输入设备报告 DOS,第一个#指定行号,第二个#指定列号。

ESC [ 6n 返回光标位置

ESC [ s 保存光标位置

ESC [ u 恢复光标位置

#### 屏幕清除功能:

ESC [ 2j 清除屏幕,光标“回家”(左上角)

ESC [ k 清除自光标位置开始到行尾的全部信息。

#### 屏幕显示规格:

ESC [ = #h 设置屏幕显示规格 参数#:

0--40×25 黑白;

1--40×25 彩色;

2--80×25 黑白;

3--80×25 彩色;

4--320×200 彩色;

5--320×200 黑白;

6--640×200 黑白;

7--在行尾处键入将产生一新行,不上移。

ESC [ = #l 复位屏幕显示规格 参数#:

0~6 同上

7--行尾不卷绕(丢掉超过行尾的字符)

#### 图形显示控制:

ESC [ #; ...; #m 设备图形显示的特征

在图形方式下,字符显示将具有由下述参数所指定的特征,直到有另一个控制序列改变。

参数# 含 义

0 全部特征关闭(常规的黑底白字)

1 粗体(高亮度)

4 下划线(仅适用单色显示)

5 闪烁

7 反转(白底黑字)

8 消隐(不可见)

30 黑字

31 红字

32 绿字

33 黄字

34 蓝字

35 深红色字

36 蓝绿色字

37 白字

40 黑底色

41 红底色

42 绿底色

43 黄底色

44 蓝底色

45 深红底色

46 蓝绿底色

47 白底色

#### 键符的新定义控制:

ESC [ #; #; #p 赋予参数指定的键符新定义

第一个#指定替换的键符,其后的#定义替换后的新定义。若第一个#为0,则与其后的第二个#共同构成被替换的扩展 ASCII 键符。

例如:

ESC [ 65; 81p 表示 A 新定义为 Q

ESC [ 0; 68; 97p 表示 F10 的新定义为 a

ESC [ #; "STRING"; ... #p 赋予参数#指定的

键符以双引号内的字符串定义

或者"String"与#任意的组合

例如:

ESC [ 0; 68; "dir"; 13P

该序列意味着 F10 定义为 dir 命令(带回车键)。

以上这些功能,用户可以自己编制一个外部命令文件或在批处理中使用,根据需要选用有关命令。在系统设置 CONFIG. SYS 文件中写上这条 DEVICE=ANSI. SYS 命令,并且保证根目录中含有 ANSI. SYS 文件,才能使系统支持一系列以 ESC 字符领头的字元组的特殊功能。

中文操作系统 CCDOS 2.0/2.1 是在 PC DOS 2.0/2.1 的基础上扩充汉字输入和输出功能后形成的,键盘输入和屏幕和输出功能的扩充就是利用 ANSI. SYS 的控制特性。此命令文件非常重要,在实际使用中,我们碰到除了在长城本身所带的操作系统外,如果在 CONFIG. SYS 文件中不加入 ANSI. SYS 命令,如:(1)在 2.13F 系统结构中无此命令,则系统将无法启动,在启动的中途死机。(2)在 286、0520CH 等长城系列机的 2.13D 系统结构下调用长城系统结构下装入的 dBASE 管理系统,在修改程序文件存盘后,会出现屏幕不能向上滚动,数据总是在第 25 行显示的现象,给使用带来不便。相反如果在 CONFIG. SYS 中加入了该命令就很好地解决了这些问题。

我们可以根据需要,在实际中灵活使用设备驱动程序 ANSI. SYS 来扩充屏幕和键盘,弥补 DOS 命令的某些不足。



## 用 PC TOOLS 修改软件的汉字提示

山东济宁市煤气公司调度室 李开新

对于改变软件的提示信息一般用 DEBUG 进行修改,但对于一个不太懂 8088 汇编语言的人来说比较麻烦,且由于 DEBUG 本身的局限,不能修改 \*.EXE 文件(或 \*.HEX 文件),需将后缀改变,才能修改。而用 PC TOOLS 修改则简而易行,而且可以直接修改 \*.EXE 文件(或 \*.HEX 文件),下面以 CDBASE III ver 1.0 为例说明如下:

DBASE 启动后的提示信息存在于 DBASE.OVL 文件中,执行 DBASE.EXE 文件时将其显示在屏幕上,其中有“山东省济宁市计算机实业公司软件部”。设欲将其改为:山东省济宁市煤气公司调度室李开新,修改只对 DBASE.OVL 文件进行即可。具体修改如下:

一、查取“山东省济宁市计算机实业公司软件部”的机内码为:C9BD B6AB CAA1 BCC3 C4FE CAD0 BCC6 CBE3 BBFA CAB5 D2B5 B9AB CBBE C8ED BCFE B2BF,又查得“煤气公司调度室李开新”的机内码为:C3BA C6F8 B9AB CBBE B5F7 B6C8 CAD2 C0EE BFAA D0C2。查取汉字的机内码的方法有多种,可以编辑一文件,只含以上文字,根据汉字编码的高低字节最高位都为“1”的特点,用 PC TOOLS 的“F”命令查取;可以从国标区位码表查得;可以编程序或用其它方法查得。

二、启动 PC TOOLS,对 A 驱动器的软盘(含 DBASE.OVL 文件)操作,进入文件功能后,选择 DBASE.OVL 文件,按“F”键进入查找字符串模块,按“F1”键进入机内码模式,输入要查的汉字的机内码,键入回车后自动查找,查找结束后,按“E”键进入“观察编辑”,则在找到的机内码前显示闪烁的光标,然后用新的汉字机内码输入以覆盖原码。具体修改前后如下:

修改前显示

一.....BCC6(计)CBE3(算)BBFA(机)CAB5(实)  
D2B5(业)B9AB(公)CBBE(司)C8ED(软)BCFE(件)  
B2BF(部).....

修改后显示:

——.....C3BA(煤)C6F8(气)B9AB(公)CBBE(司)  
B5F7(调)B6C8(度)CAD2(室)C0EE(李)BFAA(开)  
D0C2(新).....修改结束后,按“F5”存盘,按“ESC”键退出 PC TOOLS。

三、运行 DBASE.EXE 文件,即可显示新的提示信息。

四、用此方法也可修改其它提示信息,或对其它西文软件的提示信息进行汉化。笔者已在 PC-286 机上对几个中西文软件进行提示的汉化,新用操作系统

为西山 CCDOS-4。PC TOOLS 版本为 DeluxeR4.3。

五、注意:修改时尽量使新的提示信息量少于原有信息量,余下的填“20”(空格),以免误改操作码,引起文件执行错误。

## 巧用批命令定义功能键

福州大学计算机系 12901 信箱 邢观锋

在上机操作实践中,我们常常要用到一些使用频度较高的关键字字符串,若每一次都一个个字符输入,势必增加击键次数,容易发生错误,降低了工作效率。如果能把功能键定义成一些使用频度较高的字串,无疑能提高效率,简化机时。本文介绍的方法是采用 DOS 的批文件功能,可直接在 DOS 提示符之下使用,只要具备磁盘操作系统基本知识的人就可运用。

步骤:先用 EDLIN 或其他文本编辑程序建立批文件 DY1.BAT

A>EDLIN DY1.BAT

1: \* ^ [[0;59;“dir A.”;13p

2: \* ^ [[0;60;“dir B.”;13p;“chkdsk”;13p

然后执行 type dy1.bat,这时屏幕上没有显示任何文本内容,但是功能键 F1 已被定义成 dir A.;F2 已被定义成 dir B.;chkdsk,这时按下 F1 即执行 dir A.功能,按下 F2 先执行 dir B.;紧接着自动执行 chkdsk 命令。建立文本 dy1.bat 时应注意以下几点:

1.“^”字符是用 ^V(即 CTRL-V)键输入;

2.59,60 分别是 F1 和 F2 键的扩展 ASCII 码,引号内是要定义的内容,13 表示回车,p 表示此行结束(p 要小写);

下面提供一段小程序,原理同上,由于采用了批文件中的哑参数,能任意地改变 F1~F9 功能键的定义。

A>EDLIN DY2.BAT

1: \* ECHO ^ [[0;59;“%1”;13p

2: \* ECHO ^ [[0;60;“%2”;13p

3: \* ECHO ^ [[0;61;“%3”;13p

4: \* ECHO ^ [[0;62;“%4”;13p

5: \* ECHO ^ [[0;63;“%5”;13p

6: \* ECHO ^ [[0;64;“%6”;13p

7: \* ECHO ^ [[0;65;“%7”;13p

8: \* ECHO ^ [[0;66;“%8”;13p

9: \* ECHO ^ [[0;67;“%9”;13p

执行时只要键入 dy2,后面按顺序跟上要定义的内容,相互之间至少要一个空格。执行结果屏幕上也是不显示文本内容,但各功能键已被定义,若 dy2 后不跟任何定义内容,即可取消 F1~F9 原有的定义。

(注:以上所述方法在 CONFIG.SYS 中须有 DEVICE=ANSI.SYS 此条命令)。

## CCDOS 下屏幕死锁的解决

桂林工商银行电子中心站 成刚 杨明秋

### 一、现象：

使用 CCDOS (如 CCDOS2.1, CCDOS3.0 等) 时, 有时会出现正常的信息 (指非提示行的信息) 掉进提示行 (第 10 行) 的情形。一旦掉进提示行, 则屏幕的正常内容不再滚动, 如 dir 列目录命令的所有显示都在最低行。每一行的输出都将前一行的内容覆盖掉。这种现象破坏了屏幕的正常显示, 影响了用户的使用, 我们把这种现象称之为屏幕死锁。

### 二、分析：

在 CCDOS 的显示 (CRT) 控制模块中, 采用了实屏和虚屏两个概念。操作系统对实屏中的最末一行 (第 10 行) 进行了特殊的处理, 使它成为提示行。它恒对应虚屏的最后一行 (第 24 行)。系统对实屏内容进行滚动时, 只滚动前面的 10 行 (0~9 行), 而不滚动第 10 行。所以, 一旦信息进入第 10 行, 就会造成屏幕死锁。

屏幕死锁的原因在于系统盘的根目录的系统配置文件 config.sys 中使用了配置命令 device=ansi.sys, 使得 ansi.sys 代替了系统的标准屏幕和键盘程序。

ansi.sys 是一个用来扩充屏幕及键盘功能的设备驱动程序, 它把 25 行看作是屏幕的一页, 即 0~24 行。这样, 当显示内容进入到最后一行 (第 24 行) 时, 就造成了屏幕死锁——前面我们讨论过, CCDOS 实屏的提示行 (第 10 行) 恒对应虚屏的第 24 行。

### 三、解决屏幕死锁的方法

解决屏幕死锁的方法有很多, 下面简单谈几种。

#### ① 第一种方法：

最简单的方法是将文件 config.sys 中的 device=ansi.sys 命令删去, 即可消除屏幕死锁现象。这种方法的缺点是用户不能使用屏幕和键盘的扩充功能。

#### ② 第二种方法：

用 debug 命令对造成屏幕死锁的 ansi.sys 稍作修改。进行如下操作 (下划线处为用户修改处)。

0278 处的指令 CMP BYTE PTR [0102], 19 和 027F 处指令 MOV BYTE PTR [0102], 18 决定了光标定位的范围是第 0~24 (18H) 行, 现在我们把光标定位范围修改为 0~23 (17H) 行, 那么显示的内容就不会进入最后一行 (第 24 行), 从而消除了屏幕死锁现象。

```
A>debug ansi.sys
-e 27c
-xxxx;027c 19.18
-e 283
-xxxx;0283 18.17
-w
```

#### ③ 第三种方法：

用 CCDOS 原配盘上的 ansi.sys 替换掉引起屏幕死锁的 ansi.sys 也可消除屏幕死锁。

实际上, CCDOS 原配置的 ansi.sys 也象第 2 种方法那样作了修改。

方法二和方法三既可消除屏幕死锁, 又可使用户使用屏幕和键盘的扩充功能。

## VDISK 的妙用

北京农业大学 244# 信箱 叶莉

DOS3.0 以上版本提供了一个名为 VDISK.SYS 的虚拟磁盘驱动程序, 从而使用户能自由地按自己需要建立适当大小的 RAM 虚拟盘。使用中发现, 2.X 的 DOS 版本利用 3.0 所提供的驱动程序也可以使用 RAM 虚拟盘。

对双软驱无硬盘系统的用户, 在从应用程序中退出时, 常会碰到 "Insert disk with COMMAND in drive A and strike any key when ready" 这个提示。这时你不得不中断手头繁忙的工作, 从也许是一大堆磁盘中找出系统盘。之后也许还得再将工作盘换入, 这的确是一件让人头痛的事。

随着 DOS 版本的升级, 系统文件在磁盘中所占容量也越来越大。DOS 3.0 已达约 58K, 而 DOS3.3 与 CCDOS 汉字库已不能并存于同一张盘上。因此一般作法是除启动盘外通常将盘格式化为工作盘 (即不含系统文件)。这样, 前面所说情况更容易发生。

本人在上机实践中发现一种利用 RAM 虚拟盘巧妙地解决这一问题的方法, 实现过程如下:

一、确认你的启动盘上有 VDISK.SYS 文件

二、建立或编辑该盘上 CONFIG.SYS 文件, 使之包含如下行:

```
DEVICE=VDISK.SYS,nk
```

其含义是设置一大小为 nk 的虚拟盘, 视自己需要设置 n 大小。若使用本方法, 建议 n 至少为 25K。

三、建立或编辑 AUTOEXEC.BAT 文件, 使之包含如下两行:

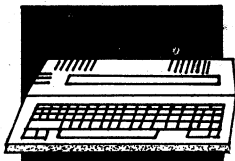
```
COPY COMMAND COM C:(C:虚拟盘盘号)
```

SET COMSPEC=C:COMMAND.COM (设置系统重新装入命令处理程序的路径为 C:) 若所设置虚拟盘盘号不同, 则需相应改动此两处。

用这个盘启动后, 你就再也不会碰到要求插入系统盘的提示, 当你从应用程序退出, 不会再停顿, 而是立即回到系统提示符下。

RAM 虚拟盘最大特点是高速度, 其存取速度即内存存取速度, 比磁盘操作快几十倍。因此, 使用上述方法 (下转 13 页)





## 学习机之友

# 谈 谈 FORTH

北京机工印刷厂 丁志伟

FORTH 语言,从软件设计的角度看,比起 BASIC 来,更适合于中华学习机这样的八位机。鉴于人们普遍对它缺乏了解,下面用问答形式向大家介绍。

甲:你怎么想起要介绍 FORTH 语言来的?

乙:据了解,中华学习机已经生产了十几万台,加上苹果机及兼容机,还有各种以 Z80 为 CPU 的微机,我国八位微机的社会拥有量相当大。近年来 16 位 PC 机越来越便宜,八位机面临被淘汰的境地。这么大一笔社会财富要被浪费,真是太可惜了。我们还没富裕到如此迅速地对微机更新换代的程度,应该充分发挥这些微机的效能,延长其生命周期。

甲:这和你要介绍的 FORTH 语言有什么关系?

乙:我先问你,你使用的是中华学习机,又有一定应用经验,开发应用程序时用什么语言呢?

甲:一般用 BASIC,如果要求速度高,或对某些特殊要求的场合,就用汇编。大多数人都如此,你呢?

乙:过去我也和你一样。BASIC 一般还可以用,但它毕竟是初学者语言,对已经跨过初学门槛的人来说,常会感到不足。比如说速度慢,程序长而变量较多时,编程和排错相当麻烦,维护和修改也较困难。汇编就更麻烦了,设计周期长,对编程者要求也很高。

甲:你说得不错。那 PASCAL、FORTRAN、LOGO 等语言是否会好些?

乙:当然也能用。可惜受机器内存、速度限制,八位机上这些语言的版本都不完善,使用者很少。

甲:不论用什么语言,在八位机上编一些大程序,常有捉襟见肘,顾此失彼之感,总不知出路在哪里。

乙:我过去也有这种感觉,直到找到 FORTH,问题才基本解决了。这是一个值得推荐的语言。

甲:真的吗?你先告诉我,FORTH 是什么意思?

乙:查克·摩尔(Chuk Moore)是美国的一个程序设计师。1969 年,他在美国佛吉尼亚国立无线电天文台工作时发明了这种语言,用来控制巨大的天文望远镜,当时在第三代计算机 IBM1130 上运行。由于这个语言具有一系列不同寻常的特点,效果又非常好,被命名为 FOURTH,也就是英文第四代(计算机语言)的意思。1130 机上规定标识符只能用 5 个字符,于是去掉了 U,就成了 FORTH。顺便说一句,由于这个成功,FORTH 被国际天文学会定为天文台自动化的标准计算机语言。

甲:噢,是这么回事。那这个语言都有哪些不同寻常的特点呢?

乙:FORTH 是面向堆栈的可扩展的语言系统。大多数操作都是通过堆栈进行的,很少或不用变量就能

对数据进行操作。运行时占内存少,速度快,由于通过堆栈进行操作,程序和表达式都是用一种特殊的被称为逆波兰的表达法表示的。表达式不用括号,不必考虑优先级。系统在结构上采用链式结构,又被称为词典,各功能块通过链表被串连在一起,功能块被称为动词,相当于其他语言中的语句、函数或命令。这样的结构非常易于扩展和压缩。

甲:你说的扩展是怎么回事?

乙:不管用什么语言编程,经常会遇到这种情况,所需要的功能,语言系统并没提供,这时你怎么办?

甲:这一般要通过编程解决。有些功能实现起来很麻烦,常要剖析所使用的语言系统,和原有系统协调也不太容易。不过好在《电子与电脑》经常登一些这方面的小程序。当然,有时用一两个扩展功能还行,多用几个就难了,可能会发生冲突。

乙:出现这种情况的原因在于,不论一个语言版本功能多么强,总不可能面面俱到,适合每个用户的各种需要。而且这些语言系统本身结构上比较复杂,多不对用户开放,不提供或很少提供扩展功能,靠打补丁式的扩展方法难免顾此失彼,更何况众多扩展功能的作者互不通气。

甲:那 FORTH 是怎么解决这个难题的呢?

乙:FORTH 遇到不够时,通常也是靠编程来解决。不同的是,FORTH 有一套扩展机制,程序经编译后就成为系统的一部分,是向词典增添一个新动词。

甲:新词和原有系统关系怎么样。能协调一致吗?

乙:只要编程时别出错,这个新词就好像是系统原来就具备的一样,它也能象系统中其他动词一样被调用,地位完全相同。其实系统本身很多动词就是用扩展的方法编出来的,如果对原有系统不满意,也可以重新定义有关动词,我自己就时常这么做。

甲:很有意思。这样的新词能定义多少呢?

乙:只要内存够用,想编多少都行。实际应用中可以看做没有限制。反过来,也可以删除一些不需要的动词,以便使系统保持在适当的规模上,FORTH 系统是能够拼接裁剪的。

甲:是否可以认为,扩展是通过编程进行的?

乙:不错,编程和扩展可看成一回事。通过编程,可以把系统逐步改造成符合自己需要和习惯的样子。这样的系统,主动权在使用者手里,越用就越得心应手。从这个角度看,FORTH 具有一些“学习”能力。

甲:真没想到还有这种编程语言。有这种能力,FORTH 是否很复杂呢?

乙:恰恰相反,FORTH 的功能是建立在简明的基

础之上的,结构简单,也不难理解。

甲:占内存大吗?

乙:不论是系统本身,还是源程序,或编译好的动词都相当短。一般情况下比 BASIC、PASCAL 等语言短得多,甚至经常比汇编还短。这是 FORTH 的特点之一。我用的版本,包括各种基本功能、编辑、磁盘存取等,有 160 多个词,还不到 8K 字节。后来我进行了一些改造,增加了不少基本功能,还增加了虚拟磁盘,反编译等,也只有不到 12K 字节。其中很多动词都是编程辅助工具,不是必需的,还可以删除。

甲:这对内存紧张的八位机来说,是相当理想的。

乙:据介绍,某些应用的目标程序,可以少于 1K 字节,由于占内存少,整个语言中的解释、编译、编辑、磁盘存取、汇编等功能可以同时常驻内存,成为一体。除了源程序,很少用得着磁盘存取。

甲:在编程、调试时一定很方便。

乙:确实,我对此深有体会。

甲:那速度怎么样?

乙:FORTH 速度快是公认的。我用的版本,在中华学习机上比 BASIC 一般要快 2~5 倍。对于要求速度高的场合,还可以用 FORTH 中的汇编来编动词。编译速度也相当快,其他高级语言根本无法与之相比。

甲:用 FORTH 编程是否也要快些?

乙:有人称 FORTH 是天然的结构化语言,比 PASCAL 的结构化水平还高,非常适合于软件工程学倡导的自顶向下的程序设计方法。具体编程时,用系统扩充的方法,每个动词相当于子程序,都可以独立调试,编程和调试都较简单。子程序可以通用,编制程序当然要快得多。我使用 FORTH 一年多,除了编一些小程序外,还对系统做了不少改造、扩充,编了 6502 高级反汇编、FORTH 反编译、虚拟磁盘等较大的程序。用 BASIC 来编,这种进度是不可能的。

甲:真是太迷人了,FORTH 好学吗?

乙:开始有些难。对于初学者,由于 FORTH 用逆波兰表示法,要有一个适应过程。较高水平的应用则要求对硬件系统有所了解和掌握汇编语言。跨过开始了门槛,随着对系统的适应,加上对系统的扩展、改造,就会越来越容易。怎么样,想学吗?

甲:当然想学,我以后有问题还要来麻烦你。

乙:别客气,有问题尽管来问。

甲:谢谢你介绍这么多,再见!

乙:再见!

(上接 12 页)

法还有一优点:当你使用目前一些流行软件均提供的功能 OS Shell(如 BASIC3.0、Turbo 系列)时,也需读入 A 盘上 COMMAND 文件,一方面,若 A 盘中无该文件则出错而使操作失败;另一方面读入至少需 3~5 秒的时间。而在使用本方法后,既能保证正确执行,并且在一瞬间即可完成,非常便利,充分发挥软件的功能。

另外,当程序需进行大量磁盘读写操作时,将磁盘

## 再谈特殊 6502 指令

浙江金华二中 李永前

特殊 6502 指令,笔者最早见于《最新苹果尔机加解密大全》(关永健著),于是一直在寻找 6502 指令的规律。以后在各杂志上相继见到几篇关于特殊 6502 指令的文章,但本刊 91 年第 2 期刊登的 6502 指令表最完整最有规律。对此表笔者再作如下三点补充:

1. 表①②③中现有指令(除两个保留外),均为普通 6502 指令;表④中的指令均为特殊 6502 指令。

2. 在各表中尚有许多空格,能填上的还有:表③中第 000 列 000~011 行及 100 列 000~111 行的十二个空格均为死机。表①中 000 列 100 行;表②中 010 列 100 行;表③中 000 列 100、110、111 行的五个空格均为 NOP。其余空格还有待各位验证后填上。

3. 表①中的 000 列 001 行应为 JSR \$nnnn;表④中的 000 列 110 行应为 CMP(\$nn,x)。

特殊 6502 指令可用于软件加密。一个特殊指令通常包含两个普通指令,而执行时间却与一个普通指令执行时间相等。因此如能很好地加以利用,将能起到缩短程序,提高运行速度的功效。为此笔者编写了一个高分辨第一页快速清屏的程序,以供参考。

0300-	A2 00	LDX	# \$00
0302-	A9 00	LDA	# \$00
0304-	9D 00 3F	STA	# 3F00,X
0307-	E8	INX	
0308-	D0 FA	BNE	\$0304
030A-	A9 1F	LDA	# \$1F
030C-		CF	???
		06	ASL
	030D-	03	
\$ 03		D0	BNE
	030F-	F1	
\$ 0302		60	RTS
	0311-		

030C~030D 为特殊 6502 指令,相当于普通指令:

DEC \$0306

CMP \$0306

本程序比同类程序减少了三个字节,速度加快了 48 个节拍。

读写转向 RAM 盘,可大大提高效率。如本人自编的一个汉化工具软件,需对目标文件进行逐字节读取,以提取英文提示。当目标文件较大时,速度慢得让人难以忍受。改进后方法为先将目标文件拷入虚拟盘(相当于一次完成所有的磁盘操作),再对虚拟盘中文件进行读写。实测表明,在原方法下需 6 分钟读取完成的目标文件在改进方法下仅需 40 多秒,速度提高近十倍。

# 扩展监控系统命令

蚌埠医学院 蒋建一

中华学习机的监控系统,有一个提供给用户的监控扩展命令(CTRL-Y命令)。每当键入该命令,监控程序便会跳转到内存第三页的\$3F8单元去执行,通常\$3F8处存有一条JMP\$FF69指令返回监控系统,若我们重新设置该处的跳转地址,使其转向自己编制的程序入口,便可获得监控系统的控制权。利用这一特点,我编制了几个新的监控命令,以弥补原监控命令的某些不足。新增的监控命令及其功能简介如下:

一、CTRL-Y命令。在监控系统下键入CTRL-Y/可进入小汇编状态;在小汇编状态下键入\$CTRL-Y/便可退回到监控系统。

二、CTRL-YK命令。同时以十六进制数和ASCII字符两种形式显示内存中的内容,格式如下:

1: <地址>CTRL-YK 显示指定地址单元内容。

2: <地址1>·<地址2>CTRL-YK 显示从地址1到地址2间的全部内存单元内容。

3: ·<地址2>CTRL-YK 显示从当前地址到地址2之间的内容。

4: CTRL-YK 显示当前地址到该行行尾内容。

若与1CTRL-P联用,例如1 CTRL-P<地址1>·<地址2>CTRL-YK所显内容可输出到打印机。

三、<地址1>·<地址2>CTRL-YL命令。将内存地址1到地址2中的机器指令反汇编并以双列输出到打

印机。该命令自动联打打印机,并能自动分页打印。按空格键可开始或暂停打印。当打印的程序多于2页,按P键可使其打印完一页后暂停,等待换纸或调整纸等。按ESC键可终止打印。

四、<地址1><地址2>·<地址3>CTRL-YM命令。该命令功能为在搬移内存中机器码程序的同时能够自动修正其绝对转移地址等。

在键入该程序前,须首先做好如下工作:

1. \*AA57:2/设置DOS文件缓冲区个数为2。

2. \*9D00:80 9A/将DOS缓冲区文件名栏地址改到\$9A80。

3. \*A7D4G/重新设定DOS文件缓冲区。

然后再按照程序清单正确键入全部程序,并以MON(\*^YKLM)为名存盘。本程序放在DOS的一号缓冲区中,其首址为A\$9AA6,长度为L\$237。为了方便调用我们可以用一个BASIC程序来设置并装入监控扩展程序。以后只要开机运行一次这个BASIC程序,便会自动设置好其扩展入口,并将程序保护起来。这时我们便可以进入监控,方便地选用各扩展命令了。

```
20 POKE 43607,2;POKE 40192,128;POKE 40193,154
30 CALL 42964
40 PRINT CHR$(4);"BLOAD MON(*^YKLM)"
50 POKE 1017,166;POKE 1018,154
```

```
9AA6- A4 34
9AA8- B9 00 02 C9 CB F0 1E C9
9AB0- CC F0 14 C9 CD F0 6E C9
9AB8- 8D F0 03 4C 65 FF A5 33
9AC0- C9 AA D0 06 4C 50 D3 20
9AC8- 86 9B 4C 56 D3 C0 02 B0
9AD0- 19 A5 3C 85 3A 09 07 85
9AD8- 3E A5 3D 85 3B 85 3F 90
9AE0- 09 A5 3C 29 07 D0 0D 20
9AE8- 0A 9B A2 01 20 78 FE 20
9AF0- 92 FD 84 47 A9 A0 20 ED
9AF8- FD B1 3C 20 DA FD E6 47
9B00- 20 BA FC 90 DC 20 0A 9B
9B08- F0 79 A9 1F 85 24 B1 3A
9B10- 09 80 C9 FF F0 04 C9 A0
9B18- B0 02 A9 AE 20 ED FD C8
9B20- C4 47 D0 EA 60 A0 00 A5
9B28- 42 85 3A A5 43 85 3B 20
9B30- 2C FE 38 A5 40 E5 44 85
9B38- FE A5 41 E5 45 85 FF A2
9B40- 00 20 8C F8 A5 2F C9 02
9B48- D0 28 A0 01 B1 3A C5 3C
9B50- C8 B1 3A E5 3D B0 1B 88
9B58- B1 3A C5 40 C8 B1 3A E5
```

```
9B60- 41 90 0F A0 01 B1 3A E5
9B68- FE 91 3A C8 B1 3A E5 FF
9B70- 91 3A 20 53 F9 85 3A 84
9B78- 3B A5 3A C5 42 A5 3B E5
9B80- 43 90 BC 4C 69 FF A5 3E
9B88- 85 18 A5 3F 85 19 A9 36
9B90- 0A 85 1E A2 FF 86 1B A2
9B98- 4B 86 1C 20 A4 9B A2 00
9BA0- 86 1B 86 1C 09 80 85 1F
9BA8- A5 3C A4 3D 85 3A 84 3B
9BB0- E0 00 D0 03 20 67 9C 85
9BB8- 06 84 07 20 67 9C A5 1A
9BC0- D0 01 60 4A 85 1D 90 02
9BC8- E6 1D A5 06 A4 07 85 3A
9BD0- 84 3B 20 97 9C C6 1D D0
9BD8- F9 85 08 84 09 E6 1B E6
9BE0- 1B 20 DD FB 20 58 FC 20
9BE8- 80 F2 A2 05 BD CD 9C 20
9BF0- 5C DB CA D0 F7 A6 1B A9
9BF8- 00 20 24 ED 20 73 F2 24
9C00- 1F 10 07 29 7F 85 1F 20
9C08- C0 9C A9 01 20 95 FE 20
9C10- 8E FD A6 1C 20 50 F9 A9
9C18- AE 20 ED FD A6 1B A9 00
```

```
9C20- 20 24 ED A9 AE 20 ED FD
9C28- 20 8E FD 20 8E FD A5 06
9C30- A4 07 20 83 9C 85 06 84
9C38- 07 C6 1A F0 1B A2 26 86
9C40- 24 A2 02 20 50 F9 A5 08
9C48- A4 09 20 83 9C 85 08 84
9C50- 09 20 A4 9C C6 1A D0 D3
9C58- A9 8C 20 ED FD 20 93 FE
9C60- A5 3A A4 3B 4C B4 9B A2
9C68- 00 86 1A C4 19 90 08 D0
9C70- 11 C5 18 90 02 D0 0B 20
9C78- 97 9C E6 1A A6 1A E4 1E
9C80- D0 E9 60 20 9F 9C A6 3A
9C88- 20 99 FD 20 48 F9 20 8C
9C90- F8 20 D3 F8 4C 9C 9C A2
9C98- 00 20 8C F8 20 53 F9 85
9CA0- 3A 84 3B 60 AD 00 C0 10
9CA8- 23 2C 10 C0 C9 9B D0 06
9CB0- 20 93 FE 4C 65 FF C9 D0
9CB8- D0 02 85 1F C9 A0 D0 0C
9CC0- AD 00 C0 10 FB 2C 10 C0
9CC8- C9 A0 D0 F4 60 BA C5 C7
9CD0- C1 D0 00
```



# MZ-731 机绘制的财务计划执行分析报表

广东 花县宾馆 黄浩珍

夏普 MZ-731 微机的四色绘图打印机可输出 80 列字符。我用 BASIC 语言编了这个程序,用它来做财务计划执行情况分析报告,提高工作效率很多,对于及时提供分析资料,指导经济活动,有一定的实用价值。

本程序只要在 DATA 语句中置入本单位各个部门本期实际发生的营业额和利润,以及对应的计划数、去年同期数,就会自动完成汇总和求出与财务计划执行分析有关的百分比,并绘制出下列图表:

(一)《本期计划执行报告表》。分别列出各单位本期实现的营业和利润的计划数、完成数、完成计划的百分比和利润率。

(二)《总额比重图》。以园形百分图描出各部门营业额与利润额在总额中所占的比重。

(三)《同期比较图》。以直方图绘出各部门营业额、利润额与去年同期变化的比较。

本程序用模块式的结构编制,每执行完一项功能后,都先回到主菜单程序,所以功能是任选的,既可以只打印某一个表或图;也可以重复打印某一个表或图。如把单位名称改为月份,在数据置入中把 B\$ 改为 1-N 月,就可成为《期中计划执行情况累计表》;同理,如置入 1-12 月份的数据,就可以编制出本单位的《全年分月计划执行情况报告表》。另外,绘图机所绘出的图形比例精确,用放大尺放后,可作为办公室的悬挂图表。

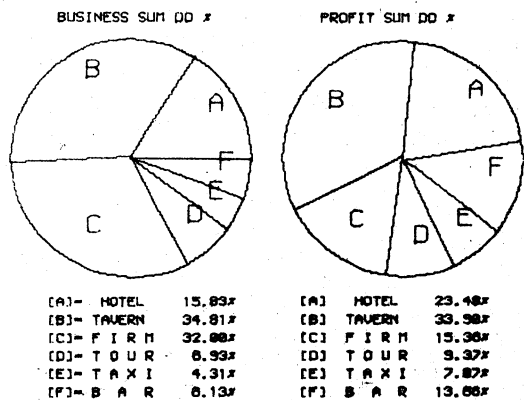
本程序头段为计算程序。(1)为主菜单程序;(2)为打印《计划执行报告表》程序;(3)为绘制《总额比重图》程序;(4)为绘制《同期比较图》程序;(5)为打印星状线条子程序;末段为数据。

为使图面完整合理,某一单位发生亏损时,《总额比重图》的利润部份会自动剔除该单位不计,《同期比较图》会将其置零,但注释部份仍标明其实际数。

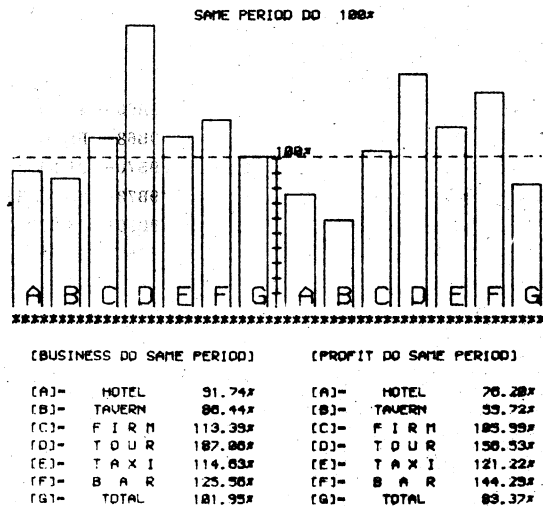
为便于校对及保全资料,数据统一用 READ 读入。主程序(10-1300 行)与数据应分别存带。数据部份统一用 3000 行号开始,以时间作注释及存带文件名称,先在 3010 行号的 DATA 中置入 B\$ (时间)和单位数(N)。从 3020 行号开始,依次置入表列各数。每一部门占用一个行号,校对好后先存带。使用时先 LOAD 载入数据文件,再用“ME.”指令连接主程序运行。

```
10 REM ---PLAN CARRY REPORT---
20 READ B$,N,M=N+1,Z=6;W=1;C$="-----"
30 IF N>=10 THEN Z=4;W=0
40 DIM A$(M),I$(M),A1(M),A2(M),A3(M),A4(M),B1(M),B2(M),B3(M),B4(M),B5(M)
50 FOR I=1 TO N:I$(I)=CHR$(64+I)
60 READ A$(I),A1(I),A2(I),A3,B1(I),B2(I),B3
```

1990.10 SUM SCALE MAP



1990.10 SAME PERIOD COMPARE



```
70 A3(I)=A2(I)/A1(I)*100
80 A4(I)=A2(I)/A3*100
90 B3(I)=B2(I)/B1(I)*100
100 B4(I)=B2(I)/B3*100
110 B5(I)=B2(I)/A2(I)*100
120 A=A+A1(I);B=B+A2(I);C=C+A3
130 D=D+B1(I);E=E+B2(I);F=F+B3
140 NEXT;RESTORE
```

```

150 A$(M)="TOTAL",I$(M)=CHR$(64+M)
160 A1(M)=A,A2(M)=B,A3(M)=B/A*100,A4(M)=B/
C*100
170 B1(M)=D,B2(M)=E,B3(M)=E/D*100,B4(M)=E/F
*100,B5(M)=E/B*100
180' ---[1]---
190 PRINT "MENU PROGRAM";PRINTTAB(10)"MENU PRO-
GRAM";PRINT
200 PRINT,"[1] TOTAL LIST";PRINT
210 PRINT,"[2] SCALE MAP";PRINT
220 PRINT,"[3] SAME PERIOD COMPARE MAP";PRINT
230 PRINT,"[4] END";PRINT
240 PRINT,"PLEASE SELECT KEY";
250 INPUT K,IF K>4 THEN 250
260 ON K GOTO 270,500,920,1280
270' ---[2]---
280 PRINT "THE PRINTER
LIST";PCOLOR 1
290 PRINT/PTAB(7)B$;"PLAN CARRY REPORT";PCOL-
OR 2
300 PRINT/PTAB(8)C$;GOSUB1300;PCOLOR1;PRINT/P
310 PRINT/PTAB(21)"BUSINESS";PRINT/PTAB(51)
"PROFIT";PRINT/PTAB(72)"PROFIT";PRINT/P"
NO NAME"
320 PRINT/PTAB(15)"PLAN FULFILL
DO % PLAN FULFILL DO %";PRINT/
PTAB(72)"RATE%"
330 GOSUB1300;PRINT/P;PCOLOR0
340 FOR I=1 TO M
350 PRINT/P"[I$(I),]";
360 PRINT/PTAB(6)A$(I);
370 PRINT/PTAB(14)A1(I);
380 PRINT/PTAB(25)A2(I);
390 PRINT/PTAB(35)" ";
400 PRINT/PUSING"###.###";A3(I);
410 PRINT/PTAB(45)B1(I);
420 PRINT/PTAB(55)B2(I);
430 PRINT/PTAB(63)" ";
440 PRINT/PUSING"###.###";B3(I);
450 PRINT/PTAB(70)" ";
460 PRINT/PUSING"###.###";B5(I)
470 PRINT/P;NEXT I
480 GOSUB1300;PCOLOR0
490 MODE TN;PRINT/P;PRINT/P;GOTO 180
500' ---[3]---
510 PRINT "THE SUM SCALE MAP";PCOLOR1
520 PRNT,"THE SUM SCALE MAP"
530 PRINT/PTAB(9)B$;"SUM SCALE MAP"
540 PCOLOR3;PRINT/PTAB(8)C$;PRINT/P
550 PRINT/P"BUSINESS SUM DO %";
560 PRINT/PTAB(22)"PROFIT SUM DO %"
570 PCOLOR1;B=0
580 MODE GR;MOVE 120,-100,HSET
590 CIRCLE 0,0,100,0,360,5;PHOME
600 FOR I=1 TO N
610 A=A1(I)/A1(M)*360;B=B+A;C=B-A*.5
620 X=100*COS(B*PAI(1/180))
630 Y=100*SIN(B*PAI(1/180))
640 LINE 0,0,X,Y
650 X1=60*COS(C*PAI(1/180))
660 Y1=60*SIN(C*PAI(1/180))
670 MOVE 0,0,X1,Y1;GPRINT[W,0],I$(I)
680 PHOME;NEXT
690 PCOLOR3;B=0;V=B1(M)
700 FOR I=1 TO N;IF B1(I)<0 THEN V=V-B1(I);NEXT
710 MOVE 240,0,HSET
720 CIRCLE 0,0,100,0,360,5;PHOME
730 FOR I=1 TO N
740 IF B2(I)<0 THEN 820
750 A=B2(I)/V*360;B=B+A;C=B-A*.5
760 X=100*COS(B*PAI(1/180))
770 Y=100*SIN(B*PAI(1/180))
780 LINE 0,0,X,Y
790 X1=60*COS(C*PAI(1/180))
800 Y1=60*SIN(C*PAI(1/180))
810 MOVE 0,0,X1,Y1;GPRINT[W,0],I$(I)
820 PHOME;NEXT
830 MOVE -360,-130;MODE TS;PCOLOR2
840 FOR I=1 TO N;A=A2(I)/A2(M)*100
850 B=B2(I)/V*100;IF B<0 THEN B=0
860 PRINT/PTAB(10)"[I$(I)]=";A$(I);
870 PRINT/PUSING"###.###";A;
880 PRINT/PTAB(50)"[I$(I)";A$(I);
890 PRINT/PUSING"###.###";B;NEXT
900 GOSUB1300
910 PCOLOR0;MODE TN;PRINT/P;GOTO180
920' ---[4]---
930 PRINT "THE SAME PERIOD COMPARE MAP";PCOLOR1
940 PRINTTAB(6)"THE SAME PERIOD COMPARE MAP"
950 PRINT/PTAB(7)B$;"SAME PERIOD COMPARE"
960 PCOLOR2;PRINT/PTAB(8)C$
970 PCOLOR3;MODE TS;PRINT/PTAB(30)"SAME PERIOD
DO 100%"
980 MODE GR;MOVE0,-80,HSET;L=240
990 LINE %5,480,0;PHOME;PCOLOR1
1000 MOVE 0,-100,HSET
1010 FOR I=1 TO M
1020 X=A4(I);Y=L/M-Z;Y1=Y*.5
1030 LINE %1,0,X,Y,X,Y,0
1040 MOVE Y1,0;GPRINT[W,0],I$(I)
1050 MOVE Z+Y,0,HSET;NEXT
1060 PCOLOR3;AXIS0,10,10
1070 GPRINT[0,0],"100%";PHOME
1080 PCOLOR2;MOVE Z,0,HSET
1090 FOR I=1 TO M;S=B4(I)
1100 IF S<0 THEN S=0
1110 IF S>200 THEN S=200
1120 X=S;Y=L/M-Z;Y1=Y*.5
1130 LINE 0,X,Y,X,Y,0;MOVE Y1,0

```

(下转 17 页)

# 机器语言程序计算阶乘

北京大学物理系新生 易鹤龄

最近我在某电脑杂志上看到一个用 BASIC 编写的求阶乘精确值的程序,运行后觉得运算速度太慢。算 100 的阶乘需 36 秒,500 的阶乘用了 19 多分钟。如果算 1000 的阶乘,不等上一、二个钟头是不会有结果的。于是我用机器语言编了一个程序,计算 100 的阶乘约 2 秒,500 的阶乘不过 1 分钟,算 1000 的阶乘也只花了 5 分 8 秒,速度之快是 BASIC 程序望尘莫及的。

此程序输入内存后,在 BASIC 状态下键入 CALL7696 或在监控状态输入 1E10G 后,会出现提示

光标。这时输入一个小于 10000 的数,程序就会算出该数的阶乘。如果输入大于 10000 或在输入中加带了字母,程序会自动排错。程序计算完一次后,会重新出现提示光标,准备下一次输入。如果想退出此程序,只需在出现提示光标时直接按回车键。

程序中 1FA3~1FA4,1FBB~1FC2 是为打印输出整齐设置的。如需打印,请先键入 POKE1657,80 把打印机设置成 80 列,如果只需屏幕显示,将上述地址内容均改为 EA 即可。

1E10— A9 01 C9 8D D0 0B A5 FC	1EA8— 11 85 11 A9 00 65 12 85	1F40— 85 07 20 B6 1E C6 09 A0
1E18— D0 04 A5 FD F0 03 20 FF	1EB0— 12 CA D0 E9 D8 60 A0 00	1F48— FE B1 08 F0 09 E6 08 D0
1E20— 1E A9 00 85 FC 85 FD 20	1EB8— 84 12 84 11 84 10 B1 06	1F50— F8 E6 09 4C 49 1F E6 09
1E28— 6F FD AD 00 02 C9 8D D0	1EC0— 85 16 A9 FC 8D 54 1E 8D	1F58— A0 00 B1 FE D0 06 E6 FE
1E30— 01 60 A2 00 BD 00 02 C9	1EC8— 96 1E 20 53 1E A5 10 91	1F60— D0 02 E6 FF F8 18 A9 99
1E38— BA B0 D7 C9 B0 90 D3 E9	1ED0— 06 A5 11 85 10 A5 12 85	1F68— 65 FC 85 FC A9 99 65 FD
1E40— B0 0A 0A 0A 0A A0 04 0A	1ED8— 11 A9 00 85 12 A5 FD F0	1F70— 85 FD D8 A5 FD D0 04 A5
1E48— 26 FC 26 FD 88 D0 F8 E8	1EE0— 0B A9 FD 8D 54 1E 8D 96	1F78— FC F0 03 4C 3A 1F C6 09
1E50— 4C 34 1E A5 FC 29 F0 F8	1EE8— 1E 20 53 1E E6 06 D0 02	1F80— A9 FD 18 65 08 85 08 A9
1E58— F0 3B 4A 4A 4A 4A AA A9	1EF0— E6 07 A5 09 C5 07 D0 C6	1F88— 00 65 09 85 09 AD 09 1E
1E60— 00 85 15 85 14 18 A5 16	1EF8— A5 08 C5 06 D0 C0 60 A5	1F90— 20 DA FD AD 08 1E 20 DA
1E68— 65 14 85 14 90 06 A9 00	1F00— FD 8D 09 1E A5 FC 8D 08	1F98— FD A9 A1 20 ED FD A9 BD
1E70— 65 15 85 15 CA D0 EF A2	1F08— 1E AD 39 1F 85 07 A0 00	1FA0— 20 ED FD A2 24 A0 00 B1
1E78— 04 06 14 26 15 CA D0 F9	1F10— 84 06 A9 00 91 06 E6 06	1FA8— 08 20 DA FD A5 08 D0 09
1E80— 18 A5 10 65 14 85 10 A5	1F18— D0 FA E6 07 A6 07 E0 70	1FB0— A5 09 CD 39 1F F0 0F C6
1E88— 11 65 15 85 11 90 06 A9	1F20— D0 F2 AD 39 1F 85 07 85	1FB8— 09 C6 08 CA D0 E7 20 8E
1E90— 00 65 12 85 11 90 06 A9	1F28— 09 85 FF A9 00 85 06 85	1FC0— FD A2 27 4C A5 1F 20 8E
1E98— 0F F0 19 AA 18 A5 16 65	1F30— FE A9 03 85 08 A9 01 8D	1FC8— FD 4C 8E FD
1EA0— 10 85 10 90 0C A9 00 65	1F38— 00 20 A5 FE 85 06 A5 FF	

(上接 16 页)

```

1140 GPRINT [W,0],I$(I);MOVE Z+Y,0
1150 HSET,NEXT
1160 GOSUB1300,PCOLOR0,PRINT/P
1170 PRINT/PTAB(7)“[BUSINESS DO SAME PERIOD]”;
1180 PRINT/PTAB(48)“PROFIT DO SAME PERIOD]”;
PCOLOR1,PRINT/P
1190 FOR I=1 TO M
1200 PRINT/PTAB(10)“[”;I$(I);“]= ”;
1210 PRINT/PA$(I);“ ”;
1220 PRINT/PUSING“###.##%”;A4(I);
1230 PRINT/PTAB(50)“[”;I$(I);“]= ”;
1240 PRINT/PA$(I);“ ”;
1250 PRINT/PUSING“###.##%”;B4(I);NEXT
1260 PRINT/P;GOSUB1300,MODE TN;PCOLOR0
1270 PRINT/P;PRINT/P;GOTO 180
1280 CLS;END

```

```

1290 ' ---[5]---
1300 PCOLOR3,MODE TS;PRINT/P
1310 FOR I=1 TO 80;PRINT/P“*”;NEXTI
1320 PRINT/P;RETURN
3000 ' ---90.10 DATA---
3010 DATA “1990.10”,6
3020 DATA “HOTEL”,280000,275405,300194,25200,
23546,30902
3030 DATA “TAVERN”,600000,605489,700456,33000,
34005,56940
3040 DATA “FIRM”,550000,556566,490840,16500,
15409,14538
3050 DATA “TOUR”,120000,120454,64393,7200,9398,
6004
3060 DATA “TAXI”,80000,74936,65373,6400,7096,5854
3070 DATA “BAR”,100000,106539,84848,12000,13705,
9498

```

(上接 16 页)



# ProDOS 磁盘操作系统入门(续)

廖 凯

## 第三章 ProDOS 命令

ProDOS 增加了 7 个新命令,改进了 DOS3.3 的 15 个命令,取消了 DOS3.3 的 6 个命令。

ProDOS 的大部分命令可在立即执行方式下使用,少数命令只能在程序方式下使用。在程序方式下使用 ProDOS 命令的方法与 DOS3.3 一样,其格式为: PRINT CHR\$(4);“ProDOS 命令”。

如果你的计算机配置的是新监控 ROM,可用大小写字母输入所有 ProDOS 命令和 APPLESOFT 语句。

DOS3.3 所使用的槽口号及驱动器号,在 ProDOS 中已被卷名所代替。当你要访问某个磁盘时,只要给出此磁盘卷名,ProDOS 就会自动地在所有的驱动器上寻找此盘。槽口号及驱动器号在 ProDOS 上仍可使用。

在下面的命令格式中,带方括号的参数在使用时可要可不要,根据自己的需要而定。在格式中的参数可以用十进制或十六进制数表示,十六进制数必须以“\$”表示。在下面的命令中,除特别说明外,均可在立即执行方式和程序调用方式下使用。

### 一、ProDOS 新的命令

#### 1. CAT 和 CATALOG

格式:CAT 路径名[,S#][,D#]

CAT/卷名/子目录/子目录

CATALOG 路径名[,S#][,D#]

CATALOG/卷名/子目录/子目录

功能:CAT 以 40 列显示方式简单地显示磁盘目录,CATALOG 是以 80 列显示方式详细地显示磁盘目录。所显示的各项的含义如下:

/××××:卷名(即磁盘名)。

NAME:文件名。在文件名前面的星号表示已上锁。

TYPE:文件类型。ProDOS 主要有以下几种类型:

文件类型	类型符号
APPLESOFT 程序	BAS
APPLESOFT 变量文件	VAR
二进制文件	BIN
目录文件	DIR
ProDOS 系统文件	SYS
可移动代码文件	REL
文本文件	TXT
用户自定义文件	\$F#
其它文件类型也可能存在,有些是针对 APPLEIII	

而言的,有些是由 INTEGER BASIC 或 INTEGER 变量转换而来的(ProDOS 不提供 INTEGER BASIC),INTEGER BASIC 会有一个 INT 文件类型,INTEGER 变量会有一个 IVR 文件类型。

BLOCKS:表示一个文件在磁盘上占用的空间量,单位是块。ProDOS 以逻辑块代替磁道及扇区,一个逻辑块相当于 DOS3.3 的两个扇区,为 512 个字节,ProDOS 将磁盘划分为 280 个块。

MODIFIED 和 CREATED:这些项目表示修改文件的最后日期和时间,以及建立文件的日期和时间。用户可以通过计时卡设置时间或者用系统盘设置时间。

ENDFILE:ProDOS 分配给文件的最大字节数。

SUBTYPE:表示一个文本文件的长度或二进制文件的起始地址。

A:表示二进制文件的起始地址。

R:表示文本文件所占用的记录长度(用字节表示)。如果 R=0,那么这个文件是一个随机存取文件。

BLOCKS FREE:显示磁盘上未用的块数。

BLOCKS USED:显示磁盘上已使用的块数。

TOTAL BLOCKS:显示磁盘上的总块数。

在列子目录时,用户可以先设置部首(PREFIX)再 CAT 或 CATALOG。在列目录时,可用 CTRL-S 键暂停列目录。

#### 2. CREATE

格式:CREATE 路径名[,T 类型][,S#][,D#]

CREATE/卷名/子目录

CREATE/卷名/子目录/文件名,T 类型

功能:此命令用于产生不同的文件类型。ProDOS 允许用户建立一个目录或子目录文件。参数“T 类型”表示要产生的文件类型,“类型”以 3 个字母符号表示。

#### 3. FLUSH

格式:FLUSH [路径名]

FLUSH/卷名/子目录/文本文件名

功能:当资料要写入一个文本文件时,FLUSH 会使程序每次都缓冲区内资料写入磁盘。ProDOS 在资料写入磁盘之前,先将资料存储在 512 字节的缓冲区内。如果程序在未写好这些资料前令系统失去控制,这些资料便会失去。使用 FLUSH 命令可确保你的程序的安全。虽然 WRITE 命令也可以将资料写入磁盘,但这命令会使磁盘存取程序的速度减慢。在使用 FLUSH 而未给文件名时,将使所有打开着的文件被 FLUSH。

#### 4. PREFIX

格式:PREFIX

PREFIX,D#

PREFIX 路径名[,S#][,D#]

PREFIX/卷名/子目录/

功能:此命令是用于定义一个路径名。路径名可以指定为目录或子目录,使它较方便地与一个目录内的不同文件工作。当单独使用 PREFIX(无参数)时,ProDOS 会送回当前的路径名。如果只有 S#和 D#参数,PREFIX 会设置成此驱动器内磁盘的名字。如果用户在定义路径名而在结尾没加上斜线时,ProDOS 会自动加上,使系统保持完整性。

#### 5. STORE 和 RESTORE

格式:STORE 路径名[,S#][,D#]

STORE/卷名/子目录/BASIC 文件名

RESTORE 路径名[,S#][,D#]

RESTORE/卷名/子目录/BASIC 文件名

功能:STORE 命令是将目前的程序所产生的变量值(要产生变量值,必须先执行程序)存入磁盘。STORE 命令将每个变量的值和名称保存在一个 VAR 型文件内。对于存储少量的变量值,STORE 命令执行的速度会比在程序调用下的 WRITE 命令快。而存储大量的变量值,则需要相当长的时间。

RESTORE 命令是将 VAR 型文件内的变量值去代替目前的变量的值。

#### 6. — (短横线)

格式:— 路径名[,S#][,D#]

—/卷名/子目录/文件名

功能:“—”命令可以将一个 BASIC 程序或机器语言程序或可执行的文本文件装入内存并执行。它可代替 RUN,BRUN 和 EXEC 命令。它可运行的文件类型有 BAS,BIN,SYS 和 TXT。

### 二、ProDOS 改进的命令

所有改进的命令可以被用在与 DOS3.3 相同的环境中,在转换 DOS 程序到 ProDOS 时,这些命令不需要改变。

#### 1. APPEND

格式:APPEND 路径名[,L#][,S#][,D#]

APPEND/卷名/子目录/文本文件名

功能:此命令用于增添资料到文本文件的末尾。APPEND 先打开一个文件,将指针定位到文件的末尾,然后写入资料。如果用于随机存取文件,那么记录将被加到最后逻辑记录的后面。此命令只能用在程序方式下。

L# 参数表示在文件内的一个记录的长度,并被用在 APPEND 一个随机文件时。如果记录长度(L#)与文件被建立时一样,那么新文件将连接在旧文件的末尾。当 L# 是一个不同的长度时,ProDOS 会根据记录长度来划分 ENDFILE,剩余部分是当前记录的偏移量。ProDOS 使用如下公式计算 ENDFILE:

$$\text{ENDFILE} = \text{偏移量} + \text{记录长度} + 1$$

#### 2. BLOAD

格式:

BLOAD 路径名[,A#][,B#][,L#/,E#][,T 类型][,S#][,D#]

BLOAD/卷名/子目录/文件名,A#,T 类型

功能:此命令可以用“T 类型”参数从磁盘装入任何类型文件到内存的任意位置。如果没有“T 类型”参数,那么 ProDOS 将装入一个 BIN 类型文件。BLOAD 允许用户装入高分辨率图形到内存的图形页。高分辨率图形第一页的首地址为 \$2000,第二页为 \$4000。

各参数含义如下:

A#:指定文件的首起字节装入内存的起始地址。

B#:是指要从文件内第几个字节开始装入。如果没用这参数,那么就是从文件的首起字节开始装入。

L#/E#:是用于装入全部或部分的文件到内存。L# 指定文件要被装入到内存的字节长度,E# 指定文件要被装入到内存的最后地址。

T 类型:指定文件类型,以 3 个字母符号表示。

#### 3. BRUN

格式:

BRUN 路径名[,A#][,B#][,L#/,E#][,S#][,D#]

BRUN/卷名/子目录/文件名,A#

功能:此命令是先装入一个类型为 BIN 的二进制文件,然后运行这程序。ProDOS 无法区别图形文件与二进制数据,如果一个图形文件被运行,那么会改变 ProDOS 在内存的位置,如果发生这种情况,就需重新启动系统。

其参数含义与 BLOAD 相同,只不过 BRUN 除了要装入内存外还要运行。如果没使用 A# 参数,那么程序将被装入内存的地址与 BSAVE 时的起始地址一致。

#### 4. BSAVE

格式:

BSAVE 路径名,A#,L#/,E#[,B#][,T 类型][,S#][,D#]

BSAVE/卷名/子目录/文件名,A#,L#/,E#

功能:此命令是用于将内存中的二进制数据存放到磁盘上。用户可以将内存任何位置的数据存放到任何类型的文件上,并可从文件内的任何位置开始存放。用户也可以保存高分辨率图形文件,供以后调用。

A#:此参数指明了要存储的数据的首起字节在内存中的起始地址。使用 BSAVE 时,必须使用此参数。

L#/E#:L# 指定要存放到磁盘的字节长度。E# 指定将被存储的数据在内存的最后地址。在使用 BSAVE 时,必须使用两个参数的其中一个。

B#:指出将被存储的数据从文件内的第几个字节开始存放。如果没用此参数,那么数据是从文件内的首起字节开始存放。

〈未完待续〉

# 用堆栈思想处理任意 N 个元素的排列组合问题

吉林市化学工业公司 第一中学 索素文

引题:有 3 个人,从事三种不同工作。问:如何分配可得到最大效益?

这是一个固定 N 个元素的全排列问题,可用三重循环进行模拟,其中循环变量 I, J, K 分别代表三种不同工作;数组元素 A(I), B(J), C(K) 分别代表三个人各自从事每种工作时的效益。程序如下:

```
10 FOR I=1 TO 3:READ A(I),B(I),C(I):NEXT I
20 FOR J=1 TO 3:FOR K=1 TO 3
40 IF I=J THEN 100
50 FOR K=1 TO 3
60 IF K=I OR K=J THEN 90
70 T=A(I)+B(J)+C(K)
80 IF T>T1 THEN I1=I;J1=J;K1=K;T1=T
90 NEXT K
100 NEXT J,I
110 PRINT "A--",I1,"B--",J1,"C--",K1
120 PRINT "T-----",T1
140 DATA 9 个给定数据
```

思考:若将题中的 3 改为 N(N 不固定),由于无法预知是几个元素,就无法用多重循环模拟。下题的引入可以帮助我们找到新方法。

引题 2:对任意自然数 N,打印出这前 N 个自然数的所有不同数字的排列来。

这是一个元素由自然数 1~N 构成的,特殊的 N 个不同元素的全排列问题。我们重新回顾一下循环的意义:多重循环的思想就是计算机中的堆栈思想:最先进入最外层循环,最后进入最内层循环;而在退出时,却是最先退出最内层循环,最后退出最外层循环。对于此题,可直接利用设置栈指针的方法。程序如下:

```
10 INPUT N;DIM A(N);T=1
30 IF T=0 THEN END
40 A(T)=A(T)+1;IF A(T)>N THEN A(T)=0;T=T-1;GOTO 30
50 FOR I=0 TO T-1;IF A(I)=A(T) THEN 40
60 NEXT I
70 IF T<N THEN T=T+1;GOTO 40
80 FOR I=1 TO T:PRINT A(I);";":NEXT I;PRINT;
GOTO 40
```

程序说明:T 指针模拟各循环层的走向;A(T) 指针模拟每一循环层循环变量的取值(T, A(T)=1, 2, …, M)。

20 句,第一次进入最外层循环。

30 句,判断是否已退出最外层循环。

40 句,不断改变当前循环层循环变量的取值,即执行某一层循环体。

50 句,相当于完成题 1 中各循环层间的 IF… THEN… 语句功能。

70 句,从外循环依次进入内循环

80 句,输出。

由解法我们看到:利用 T 和 A(T) 两个指针可以完成 N 重循环。这不仅解决了由于 N 不固定给使用多重循环带来的困难,也克服了不同机器对循环嵌套层数的约束。这样,引题 1 后面提出的问题即可有解。

例 1:N 个人从事 N 种不同工作,如何分配可得到最大效益?

分析:因 N 不固定,不能象引题 1 用三个一维数组表示效益。规定 T——第 T 个人, A(T)——第 A(T) 项工作,引入二维数组 B(N, N), 其中 B(T, A(T)) 表示第 T 个人从事第 A(T) 种工作时的效益。程序:

```
10 INPUT N;DIM A(N),B(N,N),A1(N)
30 FOR I=1 TO N:FOR J=1 TO N:READ B(I,J);
NEXT J,I
40 T=1
50 IF T=0 THEN 140
60 A(T)=A(T)+1;IF A(T)>N THEN A(T)=0;T=
T-1;GOTO 50
70 FOR I=0 TO T-1;IF A(I)=A(T) THEN 60
80 NEXT I
90 IF T<N THEN T=T+1;GOTO 60
95 S=0
100 FOR I=1 TO N;S=S+B(I,A(I));NEXT I
110 IF S<=S1 THEN 130
120 FOR I=1 TO N;A1(I)=A(I);NEXT I
125 S1=S
130 GOTO 60
140 FOR I=1 TO N
150 PRINT I;"--";A1(I)
160 NEXT I
170 PRINT S1
190 DATA N*N 个数据
```

说明:40~100 句产生一种工作分配方案,其中:T——第 T 个人, A(T)——第 T 个人从事第 A(T) 项工作;110~125 句比较保留当前效益和最大时的方案;130 句产生新分配方案;140~170 句输出最佳方案。

例 2:N 皇后问题推广:在一个 M×L 格的棋盘上,摆 N 个皇后,使它们不能两两相吃,找出所有解来,(皇后的吃法是可吃同行、同列、同对角线上的子)。

分析:这是一个平面问题。所谓一个皇后在棋盘上的某一位置,是指它位于棋盘上第 I 行,第 J 列。我们可以分两步作这个题,第一步:在 M 行中任选出 N 行



不动;第二步在 L 列中任选 N 列与第一步所选的 N 行作各种不同的组合(即对所选的 N 列作排列)。之后,按题目要求将满足条件的各点打印出来。这里要搞清两点:1、选排列与全排列的区别;2、组合问题和排列问题的区别。对于 1,区别在于选出元素的个数  $N \leq$  总元素个数 M。修改引题 2:

```
10 INPUT M,N
... .....
40 A(T)=A(T)+1,IF A(T)>M THEN.....
... .....
```

```
70 IF T<N THEN T=T+1,GOTO 40
```

对于 2,区别在于组合无序而排列有序,即同一组元素,不同顺序表示不同排列,而不同顺序与组合无关。根据这一特点修改引题 2 就可变为组合问题:

```
50 FOR I=0 TO T-1,IF A(T)≤A(I) THEN 40
```

由上分析给出皇后程序如下:

```
10 INPUT M,L,N;DIM A(M),B(L)
```

```
15 IF N>M OR N>L THEN 10
```

```
20 T=1
```

```
30 IF T=0 THEN END
```

```
40 A(T)=A(T)+1,IF A(T)>M THEN A(T)=0,T=T
```

```
-1,GOTO 30
```

```
50 FOR I=0 TO T-1
```

```
55 IF A(T)≤A(I) THEN 40
```

```
60 NEXT I
```

```
70 IF T<N THEN T=T+1,GOTO 40
```

```
80 K=1
```

```
90 IF K=0 THEN 40
```

```
100 B(K)=B(K)+1,IF B(K)>L THEN B(K)=0,K=K-1,GOTO 90
```

```
110 FOR I=0 TO K-1
```

```
120 IF I=0 THEN 150
```

```
130 IF B(I)=B(K) OR ABS(A(I)-A(K))=ABS(B(I)-B(K)) THEN 100
```

```
140 GOTO 160
```

```
150 IF B(K)=B(I) THEN 100
```

```
160 NEXT I
```

```
170 IF K<N THEN K=K+1,GOTO 100
```

```
180 P=P+1,PRINT"NO: ";P
```

```
190 FOR I=1 TO N
```

```
200 PRINT A(I);"- ";B(I);" ";
```

```
210 NEXT I,PRINT,GOTO 100
```

程序说明:A(I)——行元素,B(I)——列元素。15 句,保证在棋盘内任选 N 点;20~70 句在 M 行中任取 N 行;80~170 句在 L 列中任取 N 列进行排列,其中 146 句保证皇后在不同列且不同对角线;120 句和 150 句保证皇后在边界时处于不同列;180~210 句输出符合条件的摆法。

由此可见,正确、深刻理解引题 2 对于处理任意 N 个元素的排列、组合问题是非常重要的,同时对于任意重循环问题,也可以利用此法解决。

有兴趣的读者可试解下列两题。

1、有一个表达式  $abcdef/ghi=1988$ ,其中  $a \sim i$  这 9 个字母代表 1~9 中的数字,且这些数字各不相同。

请编程序打印出这 9 个字母分别代表的数字。

2、用 1 与 2 两种数字写 N 位数,打印出其中所有任意相邻的两位不全是 1 的 N 位数。

## 简单有趣的解密办法

唐山白玉瓷厂 张亭

BASIC 程序防止被 LIST 有多种方法,有一种在 REM 或 DATA 语句中埋设 CTRL-D 的手段是十分有效的,它可以使 LIST 产生错误信息而被中断,也可以连上 PR#6 命令重新热启动。甚至能连上 INIT 命令暗伏杀机。迄今尚未见到对这一手段的破解办法。

其实,这种保护手段的弱点是十分明显的。它必须依靠 DOS 才能实现,只要输出不经过 DOS 就会失效。在标准 DOS3.3 中,只要执行一次 CALL40672 就能达到这个目的。这样做使系统脱离了 DOS,LIST 已不会有任何后顾之忧。我们可用 CTRL-RESET 重接 DOS。

这全部操作过程未免繁琐,而且对于不同版本的 DOS,如存放在 16K 扩展 RAM 中的 DOS 就不通用。下面介绍一个更简便,并且十分有趣的办法:

];PR#0;LIST

或者列在打印机上:

];PR#1;LIST(假设打印卡插入 1#槽)

PR# 命令前面加上冒号,是为了区别于 DOS 的 PR# 命令而作为 BASIC 命令来执行。这样一来输出过程就“瞒过”过了 DOS,而 LIST 执行完之后,控制权又自动归还了 DOS,BAISC 对打印机的接通与否 DOS 也概不认帐,因为那都是“背着它干的。”

(上接第 24 页)

```
printf("mean=%d",mean(s));
```

```
}
```

```
mean(a)
```

```
int a[];
```

```
{int i;
```

```
long s=0l;
```

```
for(i=0,i<MAX;i++)
```

```
{s+=a[i];
```

```
}
```

```
s/=MAX;
```

```
return((int)s);
```

```
}
```

```
A>EXP7-10
```

```
mean=16371
```

程序中出现的库函数 srand()和 rand()源型包含在 stdio.h 中,用法和功能分别是:

void srand(unsigned seed) 初始化随机发生器。

int rand(void)产生随机数,返回值为  $0 \sim 2^{15}-1$ ,返回周期为  $2^{32}$ 。(待续)

## 第七讲 结构数据(一)

李文兵

二讲介绍了基本数据类型,这里介绍结构数据。这包括枚举型、数组型、结构型、共体型和指针型。

## 1. 枚举(enum)

枚举型也叫枚举型,其数据结构定义形式如下:

```
enum 枚举类型名 {枚举表} 变量表
```

枚举类型名和变量表是任选项。例子如下所示:

```
enum color {red, green, blue};
```

```
enum color tv;
```

这里 color 是枚举类型名, tv 为该枚举类型的一个变量。对于枚举变量,可以进行如下赋值和比较运算。

```
tv = red;
```

```
if (tv == green) printf("tv color's green\n");
```

这时,若打印变量 tv 的值,即:

```
printf("%d %d %d", red, green, blue);
```

则输出为 0、1、2。可见,枚举型数据为整型常量。其列举表中各个标识符不仅能按 0、1、2...取值,而且能设定为任何整型常量。对于上例,若按如下初始化:

```
enum color {red = 100, green, blue};
```

则 red, green, blue 就分别对应 100、101、102。

枚举型变量可用于控制结构中,如在 for 结构中:

```
main()
```

```
{enum color tv;
```

```
for (tv = red; tv <= blue; tv++)
```

```
{:
```

```
}
```

```
}
```

又如用于 switch 结构中:

```
switch (tv)
```

```
{ case red: 语句 1; break;
```

```
case green: 语句 2; break;
```

```
case blue: 语句 3; break;
```

```
default: 语句 4;
```

```
}
```

巧妙地运用 enum 型结构数据,可使程序易读易懂,如练习 7.1 所示。

```
A>TYPE EXP7-1.C
```

```
main()
```

```
{enum color {red, blue, green} c-char;
```

```
/* enum type test 1 */
```

```
printf("%d %d %d\n", red, blue, green);
```

```
/* enum type test 2 */
```

```
c-char = red;
```

```
if (c-char == red) printf("RED\n");
```

```
/* enum type test 3 */
```

```
for (c-char = red; c-char <= green; c-char++)
```

```
printf("%d ", c-char);
/* enum type test 4 */
for (c-char = red; c-char <= green; c-char++)
{switch (c-char)
{case red:
printf("\nred\n");
break;
case blue:
printf("blue\n");
break;
case green:
printf("green\n");
break;
}
}
printf("program end. \n");
}
```

```
A>EXP7-1
```

```
0 1 2
```

```
RED
```

```
0 1 2
```

```
red
```

```
blue
```

```
green
```

```
program end.
```

## 2 数组

所谓数组在 C 语言中是指同一类型数据的集合。

Turbo C 能定义任何维数的数组。

(1) 一维数组 一维数组的定义格式为:

数据类型 数组名[尺寸];

例如: char name[16];

就定义了一个一维数组,其每个元素都是字符型数据,一共有 16 个这样的元素,变量名为 name。注意,该数组的 16 个元素是从 name[0] 开始,到 name[15] 为止的。实例见[练习 7.2]

```
A>TYPE EXP7-2.C
```

```
main()
```

```
{ char ch[10];
```

```
int i;
```

```
for (i = 0; i < 10; i++)
```

```
{ch[i] = 'A' + i;
```

```
printf("%c ", ch[i]);
```

```
}
```

```
}
```

```
A>EXP7-2
```

```
ABCDEFGHIJ
```

使用 scanf() 函数给字符型数组输入一个字符串的程序, 如练习 7.3 所示。

A>TYPE EXP7-3. C

```
main()
{
    char name[30];
    printf("What is your name:");
    scanf("%s", name);
    printf("Hello, %s\n", name);
}
```

A>EXP7-3

What is your name: WANG

Hello, WANG

注意, 由于 name 为数组名, 其值是数组本身的地址, 因此, 当作 scanf() 函数的参数时, 其前面不需要取址运算符 &。

(2) 二维数组 其定义格式为:

数据类型 数组名[尺寸1][尺寸2];

例如: int mat[5][5];

就定义了一个 2 维整型数组; 其行数为 5, 列数亦为 5, 共有 25 个元素; 每个元素都是整型。如图 7.1。

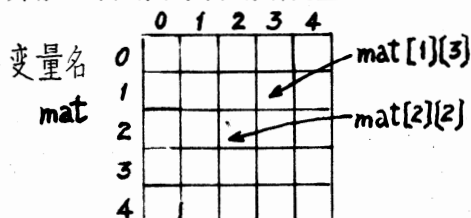


图 7.1 2 维数组的元素

二维数组赋值和输出的程序, 如练习 7.4 所示。

A>TYPE EXP7-4. C

```
main()
{
    int row=5, column=5, i, j;
    int mat[5][5];
    for(i=0; i<row; i++)
        for(j=0; j<column; j++)
        {
            mat[i][j]=i*row+j;
            printf("%2d ", mat[i][j]);
        }
    printf("\n");
}
```

C>EXP7-4

```
0  1  2  3  4
5  6  7  8  9
10 11 12 13 14
15 16 17 18 19
20 21 22 23 24
```

若对练习 7.4 进行修改, 使程序执行结果为:

```
0  1  2  3  4      sum=10      accum=10
5  6  7  8  9      sum=35      accum=45
10 11 12 13 14     sum=60      accum=105
15 16 17 18 19     sum=85      accum=190
20 21 22 23 24     sum=110     accum=300
```

则程序如练习 7.5 所示。

A>TYPE EXP7-5. C

```
main()
{
    int mat[5][5];
    int row=5, column=5, i, j;
    for(i=0; i<row; i++)
        for(j=0; j<column; j++)
        {
            mat[i][j]=i*row+j;
            printf("%2d ", mat[i][j]);
        }
    sum(mat[i]);
}

sum(v)
int v[];
{
    int i, ss=0, column=5;
    static int sss=0;
    for(i=0; i<column; i++)
        ss+=v[i];
    sss+=ss;
    printf("sum=%3d accum=%3d\n", ss, sss);
}
```

该程序中, 语句 sum(mat[i]) 是函数调用, 在函数 sum() 中引进实参 mat[i], 以计算 i 行元素之和和截止到 i 行的累加和。可见, 二维数组的每一行可作为一维数组使用。

(3) 三维数组 其定义格式为:

数据类型 数组名[尺寸1][尺寸2][尺寸3];

例如: int array[3][4][5];

就定义了一个 3 维数组, 它是由  $3 \times 4 \times 5 = 60$  个元素组成的数组, 如图 7.2 所示:

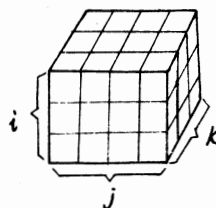


图 7.2 3 维数组 i[j][k] 的元素

3 维数组的程序如练习 7.6 所示。

A>TYPE EXP7-6. C

```
main()
{
    int m=5, i, j, k;
    int array[5][5][5];
    for(i=0; i<m; i++)
        for(j=0; j<m; j++)
            for(k=0; k<m; k++)
            {
                array[i][j][k]=i*m*m+j*m+k;
                printf("%5d ", array[i][j][k]);
            }
    printf("\n");
}
```

A>EXP7-6

```

0      1      2      3      4
5      6      7      8      9
10     11     12     13     14
15     16     17     18     19
20     21     22     23     24
25     26     27     28     29
30     31     32     33     34
35     36     37     38     39
40     41     42     43     44
45     46     47     48     49
50     51     52     53     54
55     56     57     58     59
60     61     62     63     64
65     66     67     68     69
70     71     72     73     74
75     76     77     78     79
80     81     82     83     84
85     86     87     88     89
90     91     92     93     94
95     96     97     98     99
100    101    102    103    104
105    106    107    108    109
110    111    112    113    114
115    116    117    118    119
120    121    122    123    124

```

(4)数组的初始化 前面我们分别介绍了1~3维数组的定义格式,以此类推,可以定义任何维数的数组。这里,我们介绍一下数组的初始化方法。

①1维数组可按如下进行初始化:

```
int array[2]={0,1};
```

②2维数组可按如下进行初始化:

```
int array[2][2]={ {1,0},
                  {0,1} };
```

练习 7.7 所示的初始化方法也是允许的。

A>TYPE EXP7-7.C

```

int max=4;
int array[4][4]={ {1},
                  {1,1},
                  {1,1,1},
                  {1,1,1,1} };

main()
{ int i,j;
  for(i=0;i<max;i++)
    {for(j=0;j<max;j++)
      printf("%2d ",array[i][j]);
    printf("\n");
  }
}

```

A>EXP7-7.C

```

1 0 0 0
• 24 •

```

```

1 1 0 0
1 1 1 0
1 1 1 1

```

在2维数组的定义中,行数是可以缺省的,如练习 7.8 所示。

A>TYPE EXP7-8.C

```

main()
{ int max=5;
  int array[][5]={ {1},
                   {1,1},
                   {1,1,1},
                   {1,1,1,1},
                   {1,1,1,1,1} };
  int i,j;
  for(i=0;i<max;i++)
    {for(j=0;j<max;j++)
      printf("%2d ",array[i][j]);
    printf("\n");
  }
}

```

A<EXP7-8

```

1 0 0 0 0
1 1 0 0 0
1 1 1 0 0
1 1 1 1 0
1 1 1 1 1

```

可以看出,2维数组 array 是按 5 行 5 列处理的。

字符型数组可用双引号引起来的字串进行初始化,如练习 7.9 所示。

A>TYPE EXP7-9.C

```

main()
{ char a1[]="array of character";
  char a2[5]="char";
  printf("%s\n %s",a1,a2);
}

```

A>EXP7-9

```

array of character
char

```

程序中,数组 a1 可初始化为任意长的字符串;而数组 a2 包括字符串结尾符\0,只限 5 个字符。

(5)数组参数 C 语言中,数组可以做函数的参数,可使函数调用多个同类型的数据,如练习 7.10。

A>TYPE EXP7-10.C

```

#include "time.h"
#include <stdio.h>
#include "stdlib.h"
#define MAX 10000

main()
{ int i,s[MAX];
  long n;
  srand(time(&n));
  for(i=0;i<MAX;i++)
    s[i]=rand();
}

```

(下转 21 页)





# 单片机汉字打印驱动程序的设计思想

苏州信息技术研究所 尹 征

本文以苏州信息技术研究所开发的 DHX-I 单片机汉字系统为例,介绍了以 8031 单片机为核心(CPU)的系统的打印机接口和打印驱动程序的设计思想。该方法主要解决了汉字在 9 针打印机上(如:CP-80 打印机)的打印输出,并且实现了汉字与 ASCII 码的同时打印。

## 一、DHX-I 单片机汉字系统

DHX-I 单片机汉字系统是一个只需由用户开发应用软件既可作为用户专用系统的系统,有以下几个部分组成:

1. CPU: INTEL 公司 51 系列 8031 单片机;
2. 存储器: 64KB EPROM 含 16KB 操作系统; 64KB RAM 并可切换成打印字库;
3. 显示器: 5.5 英寸 320 线,文本为 24 \* 8 汉字;
4. CRT 接口板: 8255 并行接口芯片, 2048 个显示

字库;

5. I/O 接口: 打印机接口, 四个并行接口, RS232 串行通信接口;

6. 键盘: 共 20 个按键(包括 12 个数字键, 7 个功能键, 1 个回车键);

7. 软件: 监控程序、打印驱动程序、屏幕编辑程序、数据处理程序、汉字造字程序等;

## 二 CP-80 打印机

CP-80 打印机是一种经济实惠的 9 针打印机,每秒可打印 80 个字符。每个字符由 9 \* 9 点阵组成。当以标准字符尺寸输出时,每行可打印 80 字符(或 132 个紧凑字符)。当以“宽体”字符尺寸输出时,每行可打印 40 字符(或 66 个紧凑字符)。CP-80 打印机还具有图形打印功能,通过软件送出一定的控制字符,可以实现图形和汉字的打印。

不同的打印机都有不同的输出字符集。CP-80 打印机可以打印出 96 个标准的 ASCII 字符和 128 个图形字符,另外它还可以处理和执行若干个控制字符。所谓控制字符是指那些能被打印机识别并使打印机执行某种特定功能的字符,一般它并不被打印出来,所以我们也称之为“不可打印字符”(有关控制字符的作用,在设计打印程序时是非常有用的,因篇幅所限,不再刊出,请参阅有关 CP-80 操作手册)。

## 三、打印机接口的设计

DHX-I 单片机汉字系统打印机接口采用的是标准的 Centronic 并行接口,它可与各种 9 针打印机联接。如 CP-80、CP-135、SP-1000、MX-80、MX-100、FX-100、FX-80 等打印机。接口信号共有 36 条,其中主要使用的信号如下表所示:

编号	信号名	方向	功 能 说 明	
1	STROBE	IN	以 STROBE 脉冲读入数据,在接收端的脉冲宽度必须 $> 0.5\mu s$	(选通)
2	DATA1	IN	这些信号分别表示第 1 位到第 8 位的并行数据信息。且各信号在数据为逻辑“1”时为高电平,逻辑“0”时为低电平。	(数据)
3	DATA2	IN		
4	DATA3	IN		
5	DATA4	IN		
6	DATA5	IN		
7	DATA6	IN		
8	DATA7	IN		
9	DATA8	IN		
10	ACK	OUT	约为 $7.6\mu s$ 的脉冲,低电平表示数据已接收,准备接收新数。	(回答)
11	BUSY	OUT	高电平表示打印机不能接收数据,低电平时才能接收数据。	(“忙”)
12	PE	OUT	高电平信号表示打印机缺纸。	(缺纸)

从上表可以看出接口信号包括数据选通信号 STROBE、打印机回答信号 ACK、打印机“忙”信号 BUSY 和 8 根数据信号 DATA1~DATA8 等。其时序图如图 1 所示:

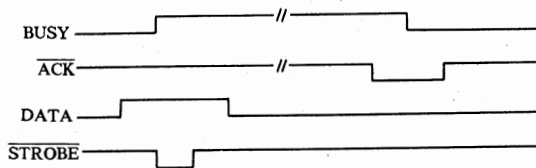


图 1

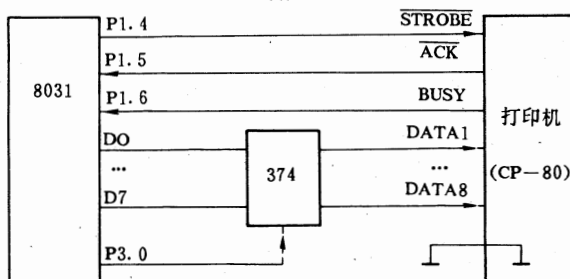


图 2

打印机接口的硬件使用了 8031 的四个口线 P1.4、P1.5、P1.6、P3.0,由软件执行口操作来完成与打印机的同步及数据传送。打印机接口硬件如图 2 所示。

#### 四、打印机驱动程序的设计

DHX-1 单片机汉字系统打印驱动程序的主要功能是完成字符和汉字的打印输出。这是整个系统中除了汉字显示之外关于汉字处理的又一个重要环节。实现汉字打印输出的过程,实际上就是把由汉字分区、代码(二字节)构成的一组汉字机内码的集合转换为打印

纸上的汉字图形的过程。各种 9 针打印机的控制字符可能有些不同,但这个过程总是一致的,其主要有以下五个步骤:

第一,从打印数据缓冲区 0(1600H—17FFH)中逐个取出预先存放在该区域内的需要打印输出的汉字文本的机内码即分区、代码。

第二,根据取出的机内码逐个计算出横向汉字点阵的首址并取出其点阵。

第三,把横向汉字点阵转换为可供打印的纵向汉字点阵并送入打印数据缓冲区 1 和 2。

第四,把打印数据缓冲区 1(1800H—1BFFH)的数据送打印机输出,即打印上半个汉字,然后 7/72"回车换行,以便能够打印下半个汉字。

第五,把打印数据缓冲区 2(1C00H—1FFFH)的数据送打印机输出,即打印下半个汉字,然后回车,以便接着能够打印 ASCII 码。

由于 9 针打印机一次只能打 9 个点,所以对 16 \* 16 汉字点阵的打印是分二次来完成的。汉字打印输出的流程图如图 3。

从以上的流程图中我们已经发现有许多模块都是通过调用不同的子程序来实现的,这些子程序是整个打印驱动程序的核心。下面,就围绕这些子程序对打印驱动程序作进一步的阐述。

##### 1. 输出字符子程序 PIOC

本程序首先打开打印数据端口,把输出字符(数据)送到打印机的数据端口上,这里的输出字符可以是 ASCII 码(如:A、1、%……),也可以是控制字符(如:回车、换行、换码……)。然后循环测试 BUSY(“忙”)信号,如果打印机忙,则继续测试;如果打印机闲,则发出 STROBE(选通)信号,选通打印机,此时数据端口上的数据被打印机接收。循环测试 BUSY(“忙”)信号的时间应根据系统的不同而定。程序如下:

PIOC;SETB P3.0; 打开打印机数据端口。

MOV DPTR, # 7FFFH; 打印机地址 7FFFH → DPTR。

MOVX @DPTR, A; 输出字符 → 打印机。

MOV R7, # 7; 设置超时计数器。

MOV R6, # 0

MOV R5, # 0

PIO1; JNB P1.6, PIO2; 读 P1.6 状态, 判打印机忙?

DJNZ R5, PIO1; 打印机忙, 继续。

DJNZ R6, PIO1

DJNZ R7, PIO1

SETB PERR; PERR 位置“1”。

RET

PIO2; CLR P1.4; 打印机闲, 选通打印机。

SETB P1.4

RET

##### 2. 回车换行与换行子程序 CRLF

本程序先向打印机输出一个回车(0DH)控制字符,打印机一旦接收到该控制字符即输出打印缓冲器中已有的内容,打印后回车。然后把换行(0AH)控制字

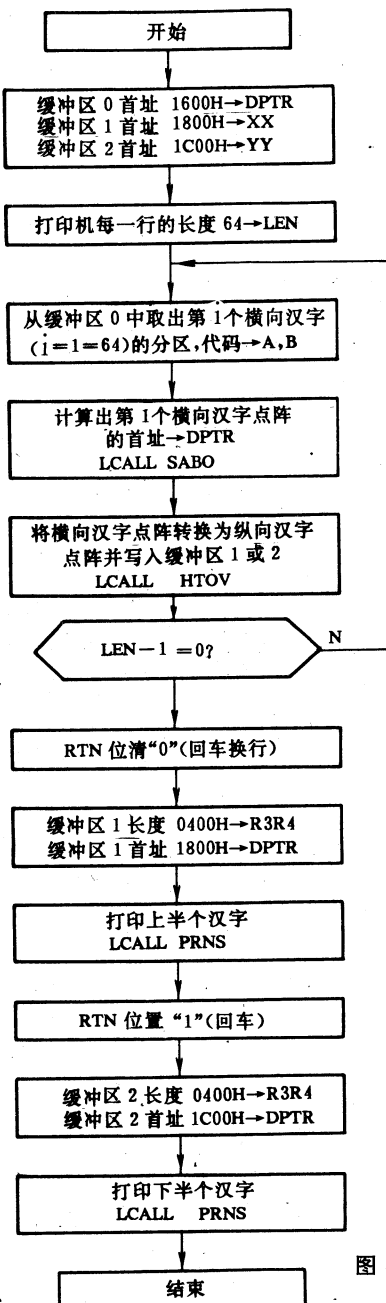


图 3

符送到打印机的数据端口上,循环测试 BUSY(“忙”)信号,如果打印机忙,则继续测试;如果打印机闲,则发出 STROBE(选通)信号,选通打印机,此时打印机换行。程序如下:

```
CRLF; MOV A, #0DH; 0DH(回车)→打印机
      ACALL PIOC
LF:   MOV A, #0AH; 0AH(换行)→打印机
      MOVX @DPTR, A
      MOV R7, #7FH; 设置超时计数器。
      MOV R6, #00H
      MOV R5, #00H
CLF1; JNB P1.6, CLF2; 读 P1.6 状态, 判打印机忙?
      DJNZ R5, CLF1; 打印机忙, 继续。
      DJNZ R6, CLF1
      DJNZ R7, CLF1
      SETB PERR; PERR 位置“1”。
      RET
CLF2; CLR P1.4; 打印机闲, 选通打印机。
      SETB P1.4
      RET
```

### 3. 7/72”回车换行子程序 LF8Z

本程序完成打印输出上(下)半个汉字时的回车换行。当打印输出上半个汉字时,先输出 ESC(1BH)1(31H)控制字符(二个),打印机接收到后即打印输出上半个汉字,打印结束后实现 7/72”的回车,然后调用换行子程序;当打印输出下半个汉字时,仅向打印机输出一个回车(0DH)控制字符,打印机接收到后即打印输出下半个汉字,然后回车。程序如下:

```
LF8Z; JB RTN, NLF8Z; RTN 位=1? yes. 转 NLF8Z
      MOV A, #1BH; NO. ESC 1
      ACALL PIOC
      MOV A, #31H
      ACALL PIOC
      ACALL LF; 0AH(换行)→打印机
      RET
NLF8Z; MOV A, #0DH; 0DH(回车)→打印机
      ACALL PIOC
      RET
```

### 4. 选择图象模式子程序 ESCS

本程序先输出 ESC(1BH) \* (2AH) 控制字符(二个),其次输出 m(图象模式),最后输出 nln2(映象信息即汉字点阵的长度)。此时,打印机即可按照设定的图象模式进行图形和汉字的打印输出。程序如下:

```
ESCS; MOV A, #1BH; ESC
      *
      ACALL PIOC
      MOV A, #2AH
      ACALL PIOC
      MOV A, B; m
      ACALL PIOC
      MOV A, R3; n1
      ACALL PIOC
      MOV A, R4; n2
      ACALL PIOC
```

RET

### 5. 设定横向列表位置子程序 ESCD

本程序先输出 ESC(1BH)D(44H)控制字符(二个),其次输出 n(横向列表的位置),最后输出 NUL(00H)。此时,打印机就从设定的位置(列)处开始打印输出。程序如下:

```
ESCD; MOV A, #1BH; ESC D
      ACALL PIOC
      MOV A, #44H
      ACALL PIOC
      MOV A, R4; n
      ACALL PIOC
      MOV A, #0; NUL
      ACALL PIOC
      RET
```

### 6. 设定多种打印模式子程序 ESCJ

本程序先输出 ESC(1BH)! (21H) 控制字符(二个)。其次输出 n(打印模式)。此时,打印机即可按照设定的打印模式进行 ASCII 码的打印输出。程序如下:

```
ESCJ; MOV A, #1BH; ESC !
      ACALL PIOC
      MOV A, #21H
      ACALL PIOC
      MOV A, B; n
      ACALL PIOC
      RET
```

### 7. 打印上(下)半个汉字子程序 PRNS

本程序首先输出 ESC(1BH) \* (2AH)2(02H)控制字符(三个)即选择 ESC Y 图象模式 9 高速双密度),然后连续输出缓冲区 1(2)映象信息即汉字点阵到打印机,等到缓冲区 1(2)的内容全部输完之后立即进行 7/72”的回车换行。程序如下:

```
PRNS; PUSH DPH; 保存 DPTR。
      PUSH DPL
      MOV B, #2; 选择 ESC Y 模式。
      LCALL ESCS
      POP DPL; 恢复 DPTR。
      POP DPH
      INC R4
PRN2; PUSH DPH; 保存 DPTR。
      PUSH DPL
      MOVX A, @DPTR; 映象信息即汉字点阵→打印机。
      LCALL PIOC
      POP DPL; 恢复 DPTR。
      POP DPH
      INC DPTR
      DJNZ R3, PRN2; 映象信息即汉字点阵取完吗? NO. 转 PRN2
      DJNZ R4, PRN2
      ACALL LF8Z; yes. 7/72”回车换行。
      RET
```

### 8. 计算横向汉字点阵的首址子程序 SAB0

本程序根据汉字文本中汉字的机内码即分区、代

码计算出一个横向汉字点阵的首址。

横向汉字点阵首址的计算公式为：

横向汉字点阵首址 = 2000H × 分区 + 20H × 代码

9. 横向汉字点阵转换为纵向汉字点阵并写入缓冲区(1,2)子程序 HTOV

本程序把一个横向汉字点阵的四块点阵(左上角、左下角、右上角、

第 0 字 节	第 1 字 节	第 2 字 节	.....	第 13 字 节	第 14 字 节	第 15 字 节	上半汉字点阵
第 16 字 节	第 17 字 节	第 18 字 节	.....	第 29 字 节	第 30 字 节	第 31 字 节	下半汉字点阵

(接 36 页)

```

70 A=-32767:B=-32762:C=-32758:D=-32753
80 FOR I=0 TO 35
90 F=130,813*1,0594634#^I:NUMBER=
4000000/16/F
100 A(I)=INT(NUMBER/256):B(I)=INT
(NUMBER-A(I)*256+.5)
110 NEXT
120 POKE A,248:POKE B,254:POKE C,251:
POKE D,15:X=USR(0)
125 POKE A,240:POKE C,244
130 READ X,Y:IF X=-1 THEN POKE A,248:

```

右下角)分别旋转 90°为可供打印输出的纵向汉字点阵,同时写入缓冲区(1,2)。纵向汉字点阵的排列图如下所示:

10. 1/4 个汉字点阵旋转 90°子程序 HTVA

本程序把一个横向汉字点阵中的第 k 块(K=1-4)即 1/4 个汉字点阵的 8 个字节的相同的位(由高到低:b7-b0)组成一个纵向的 8 个字节,即实现了 1/4 个汉字点阵旋转 90°功能。

11. 写入缓冲区 1(20)子程序 HTVC(HTVD)

本程序把纵向汉字点阵的第 i 个字节(i=0-15)写入缓冲区 1,把纵向汉字点阵的第 j 字节(j=16-31)写入缓冲区 2。

12. 数据指针加 8 子程序 DPAD

本程序实现数据指针 DPTR 加 8 的功能。

```

POKE B,255:X=USR(0):END

```

```

140 X1=11
150 IF X>9 THEN X=X/10:X1=23
160 IF X<1 THEN X=X*10:X1=1
170 I=INT(X*1.8-1.3)+X1
180 POKE B,A(I):POKE D,B(I):X=USR(0)
190 FOR T=1 TO 100*Y*SPEED:NEXT
200 GOTO 130
210 DATA.....
1000 DATA -1,-1
OK

```

## 北京安华科技市场西城经营部

主要经营:电子计算机、电子元器件

各种微机:

• 中华学习机:CEC-1

¥900.0 小蜜蜂-I ¥920.00

• 家庭 PC 机:¥3150.00

主频:4.77/12 兆 RAM:640KB 软驱:360KB × 2,101

键盘

14" 双频单显

Super286:¥6950.00

主频:10/21 兆 RAM:1MB 可扩 5MB 5¼ 软驱:1.2MB.

360KB 各 1

硬盘 3.5" 40MB. 101 键盘 14" 双频单显 数显机箱

选配:14" CGA 彩显加 1500 元 14" 1024 × 768 VGA 彩显加 3800 元

各种:APPLE II、CEC-I、PC 机接口板:九针、24 针打印

机

还承接各种微机的修理业务

对教育部门、个人购机实行优惠

地址:北京西四北大街 120 号 电话:65.6342 邮码:100034

联系人:孙义强 宋函华 BB 机:126 呼 45198 42616

## 北京华源技贸商行

经营:中华学习机、苹果机系列产品

中华学习机:750~1150 元(保修三年)

天坛学习机:995 元(保修一年)

软盘驱动器:520~620 元

中华学习机各种软件 800 余种

软盘:(学生盘、ACT、DATA、3M、Maxell janus)

2.5-7.5 元

单包显示器 520-550 元

中华学习机扩充卡 打印卡 Z80 打印卡

驱动卡 双驱动卡等

游戏棒(开关量、模拟量) 39~50

中华学习机各种资料、书籍 30 余种

凡在本部购买学习机者,将得到保修、免费培训的优惠,欢迎来函索取目录,本部代修学习机、驱动器、并可为学校配置中华学习机网络。批发面议。

地址:北京东城沙滩五四大街 52 号 邮政编码:100009

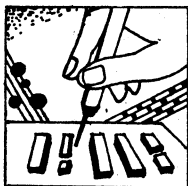
电话:512.7308 开户行:工商银行崇外分理处中都信用

社

帐户:华源技贸商行

帐号:中都 03040

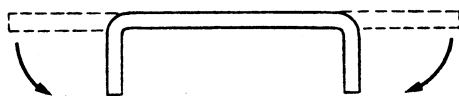
联系人:于民 张小健



学装  $\mu$ P-8

# 微电脑控制跨接线弯制机

组装电子部件时多数印刷线路板上使用图 1 所示的跨接线。跨接线一般是  $\varnothing 0.6$  毫米镀锡线，弯制成“U”形。



弯曲

图 1

## 1. 机械结构

跨接线弯制机大致分为：模具定位、镀锡线送进、切断和弯曲等四部份。

### (1) 模具定位。

模具指弯曲镀锡线时使用的上模具、下模具。如图 2 所示。固定块和移动块组成上模具，圆柱凸轮控制它左右移动，决定镀锡线弯曲长度。下模具是正六角柱体。如图 3 所示。下模具固定在轴上，与轴一起旋转，在圆柱凸轮的控制下，上、下模具的宽度相适应，即可选择跨接线弯曲长度（六种选择：8；10；12；15；18；20mm）。小形 DC 电机通过 3 级减速后，驱动下模具轴低速回转。DC 电机轴上安装孔盘，由光传感器探测穿过的光，检测电机转速，将产生的脉冲输入给微电脑。如图 4 所示，由编码器控制模具移动。

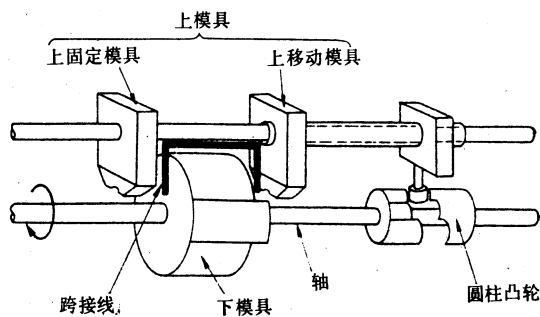


图 2

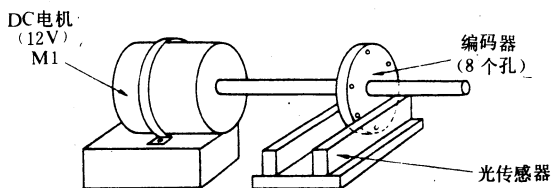


图 4

易齐干

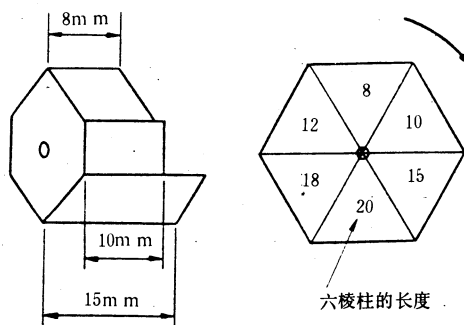


图 3

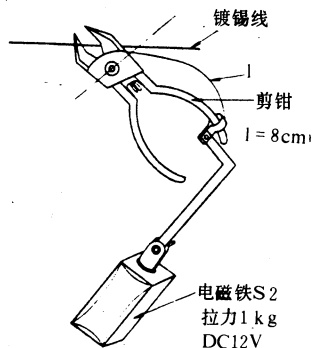


图 6

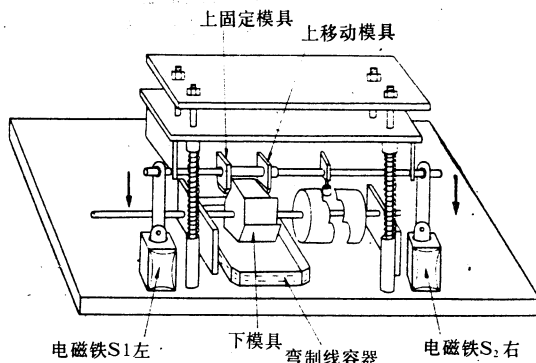


图 7

### (2) 镀锡线的送进

$\varnothing 0.6$ mm 的镀锡线绕在线轴上，进入下模具后，由于卷绕过而成圆环状，不能正确地弯曲。因此，如图 5 所示，通过两对矫直滚将卷绕变形的线矫正过来。使用 DC 电机与减速齿轮驱动送进滚子回转，镀锡线从两个送进滚子之间通过，完成镀锡线的送进。



光传感器检测光穿过编码器的孔数,产生脉冲输入给微电脑,决定镀锡线的进给量。

#### (3)镀锡线切断。

如图 6 所示,按所需要的尺寸,切断镀锡线,使用小形剪钳,切断需要 3kg/cm 力矩,则电磁铁以 1kg 的拉力下拉剪把的端部(柄长 8cm)。

#### (4)镀锡线弯曲

镀锡线的弯曲如图 7 所示。在模具的左右两侧装有电磁铁,将上模具拉向下模具,镀锡线受压弯曲。

#### 2. 接口制作:

微电脑教育部件  $\mu\text{P}-80$  发出的信息变换为适合操纵机械的信息。接口电路图如图 8 所示。 $\mu\text{P}-80$  的输入输出部件端口 C 高位④输出“通”信息,导通晶体管 2SC1815,继电器 R4 为“通”,驱动模具电机 M4 回转。编码器检测到模具应该处于停止位置时,端口 C 高位④输出“断”信息,模具电机 M4 停止回转。接着,从端口 C 高位⑤输出“通”信息,通过晶体管以及继电器 R3,驱动镀锡线送进电机 M3,传动如图 5 送进滚子回转,完成镀锡线的送进。同时,编码器(图 5)检测送进长度,达到既定长度后,M3 电机自动停止转动。然后,由端口 C 高位⑥输出“通”信息,驱动电磁铁 S<sub>2</sub>(图 6)拉动剪钳剪断镀锡线。最后,端口 C 高位⑦输出“通”信息,电磁铁 S<sub>1</sub>(图 7)动作,上、下模具合拢折弯镀锡线。1 秒钟后,上模具返回原处。

这样,结束一个循环。在折弯镀锡线时,一旦 M<sub>4</sub> 电机旋转,下模具转 1 转,如与基准起始位置一致,附着在下模具上的跨接线会落到盛放跨接线的盒内,如图 7 所示。

$\mu\text{P}-80$  与接口连接,如图 9 所示。

当控制信息为 5V 时,晶体管 2SC1815 导通,继电器 R<sub>1</sub>~R<sub>4</sub> 为“通”,步进电机与电磁铁工作。若控制信息为数 0V,晶体管 2SC1815 不导通,步进电机与电磁铁不工作。

#### 3. 软件设计。

##### (1)主流程图

表 1 给出输入输出分配表,跨接线弯制机主流程图如表 2 所示。它将显示、基准、模具位置设置、镀锡线送进、切断、弯曲、根数的计数等各功能子程序化。讨

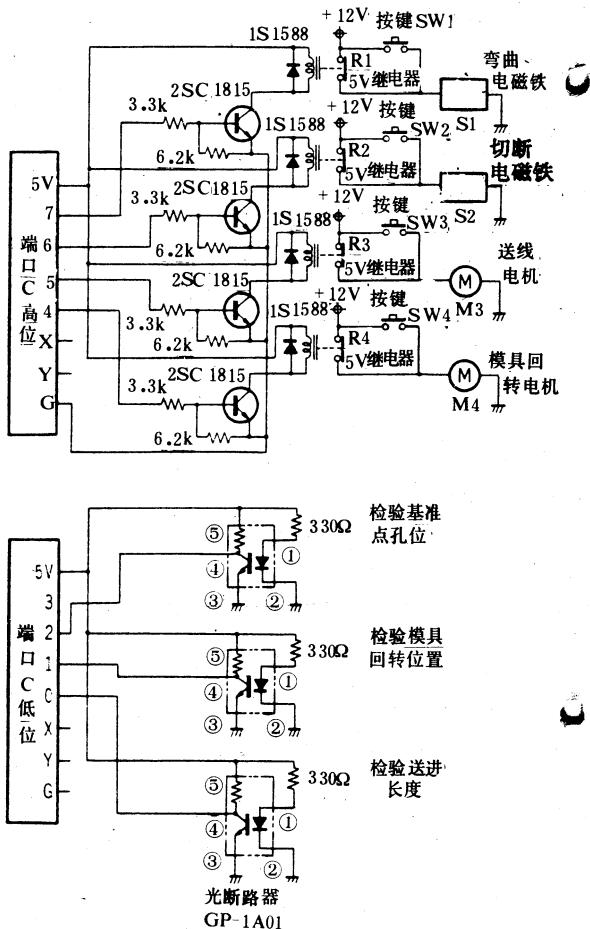


图 8

论如下。

#### (2)显示

使用 4 位数字显示部件,右边两个 7 段 LED 显示加工根数,左边两个 7 段 LED 显示弯曲长度。长度与根数的置位数据分配如表 3 所示。显示子程序流程图与程序如表 4 所示。

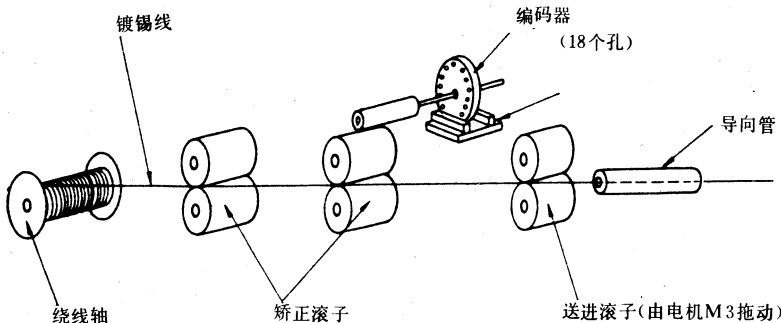


图 5

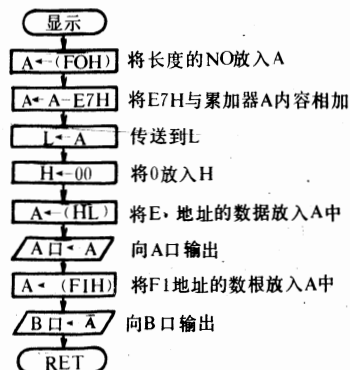
表 1 输入出分配表

端口名	输入输出	出脚名	任 务
端口 A	输出	0~7	显示弯曲宽度
端口 B	输出	0~7	显示加工根数(10进制数)
端口 C	输出	7	弯曲电磁铁
	输出	6	切断电磁铁
	输出	5	送进电机
	输出	4	模具回转电机
	输入	3	—
	输入	2	编码器(线的送进)
	输入	1	编码器(模具的位置)
	输入	0	编码器(基准点)

表 3 长度与根数置位数据分配

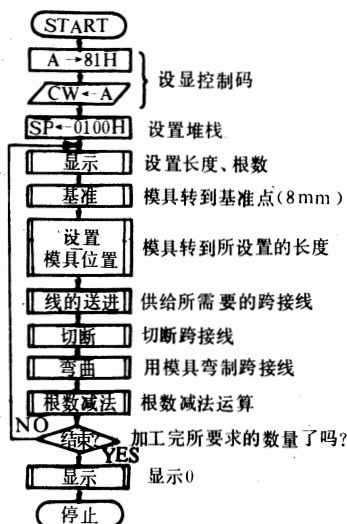
地址	内容
F 0 H	01 H 8 mm
	02 H 10mm
	03 H 15mm
	04 H 20mm
	05 H 18mm
	06 H 12mm
F 1 H	根 数 以 10 进制表示,最大为 99。

表 4 显示子程序流程图与程序



标号	助记符	地址	机器语言	注释
H YOJI	LD A, FOH	23	3A F0 00	将长度数据NO放入A累加器
	ADDA, 0E7H	26	C6 E7	将长度数据传送到HL
	LDL A	28	6F	
	LD H, 00H	29	26 00	
	LD A(HL)	2B	7E	长度数据输入给A
	OUT(00H), A	2C	D3 00	在端口A输出
	LDA(F1H)	2E	3A F1 00	向端口B输出数量
	OUT(01H), A	31	D3 01	
	RET	33	C9	

表 2, 主流程图与程序



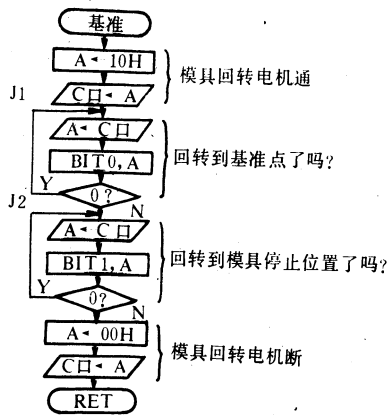
标号	助记符	地址	机器语言	注释
START	LDA, 81H	00	3E 81	设制控制码
	OUT(03H), A	02	D3 03	
	LD SP, 0100H	04	31 00 01	
MAIN	CALL HYOJI	07	CD 23 00	显示长度、数量
	CALL KIJJUN	0A	CD 34 00	模具回转到基准点
	CALL TEIICH	0D	CD 4B 00	模具转到所要求的位置
	CALL OKURI	10	CD 66 00	供给跨接线
	CALL SETSUD	13	CD 8D 00	切断
	CALL MAGE	16	CD 9C 00	弯曲
	CALL GENSAN	19	CD AB 00	根数减法运算
	JP NZMAIN	1C	C2 07 00	加工完了吗?
	CALL HYOJI	1F	CD 23 00	显示0
	HALT	22	76	停机

标号	助记符	地址	机器语言	注释
	DB 08H	E 8	08	用于显示的数据
	DB 10H	E 9	10	
	DB 15H	E A	15	
	DB 20H	E B	20	
	DB 18H	E C	18	
	DB 12H	E D	12	

(3) 确定基准点和模具位置。

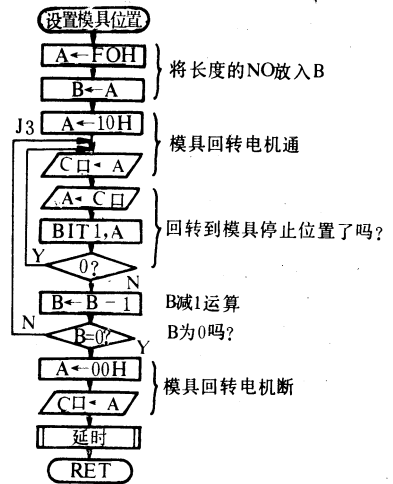
模具起始基准点的流程图与程序如表 5 所示, 弯曲 8mm 跨接线的模具回转时基准点在正上方。为得到不同长度的跨接线, 决定上、下模具位置的流程图与程序如表 6 所示。模具位置设置原理如图 10 所示。步进电机输入两支编码器产生的脉冲, 以不同位置的孔决定电机停止转动。

表 5



标号	助记符	地址	机器语言	注释
KIJUN	LD A, 10H	34	3E 10	模具回转电机通
	OUT(02H), A	36	D3 02	
J1	INA, (02H)	38	DB 02	回到基准点了吗?
	BIT 0, A	3A	CB 47	
	JP Z, J1	3C	CA 38 00	
J2	INA, (02H)	3F	DB 02	回到模具停止位置了吗?
	BIT 1, A	41	CB 4F	
	JP Z, J2	43	CA 3F 00	
	LD A, 00H	46	3E 00	模具回转电机断
	OUT(02H), A	48	D3 02	
	RET	4A	C9	

表 6



标号	助记符	地址	机器语言	注释
TEIICH	LDA(FOH)	4B	3A F0 00	将长度的NO放入B中
	LDB, A	4E	47	
	LDA, 10H	4F	3E 10	模具回转电机通
	OUT(02H), A	51	D3 02	
J3	INA, (02H)	53	DB 02	能转到模具停止位置吗?
	BIT 1, A	55	CB 4F	
	JPZ, J3	57	CA 53 00	
	DEC B	5A	05	
	JPNZ, J3	5B	C2 53 00	
	LDA, 00H	5E	3E 00	模具回转电机断
	OUT(02H), A	60	D3 02	
	CALL TIMER	62	CDB5 00	1秒延时
	RET	65	C9	

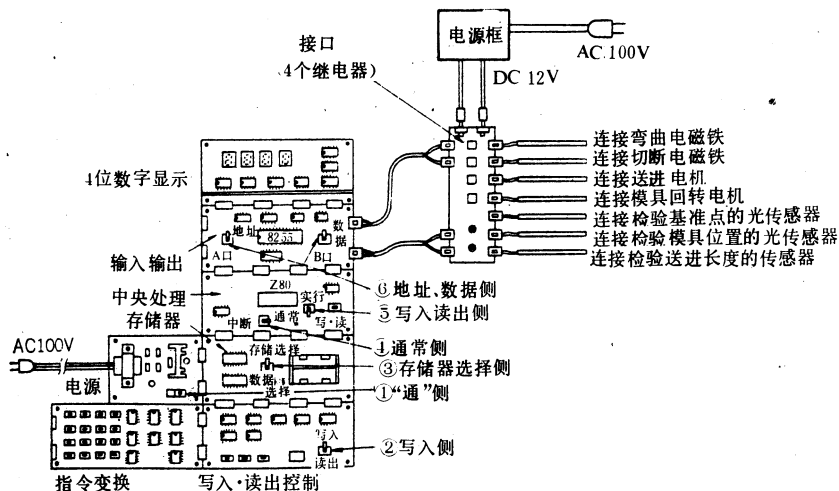


图 9

#### (4) 镀锡线的送进

编码器孔数的不同, 改变光传感器脉冲的产生, 再通过脉冲计数器决定所需长度的镀锡线的送进。编码器孔数为 18 时, 脉冲数与镀锡线长度的比例如表 7 所示。

1 个脉冲线的送进长度计算公式(由表 7)。

(例) 镀锡线弯曲长度 8mm 时,

镀锡线的送进 = 28mm  
每 1 个脉冲送进长度 = 1.12mm

这样, 所需送进脉冲 = 28 / 1.12 = 25

25 为 10 进制数, 如用 16 进制数表示为 19H (19H = 1 × 16 + 9 = 25)。

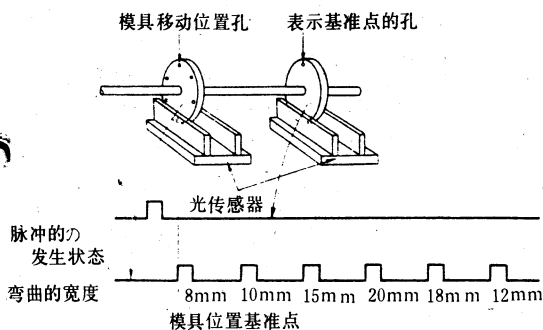


图 10

表 7

镀锡线弯曲宽度	8mm	10mm	15mm	20mm	18mm	12mm
脉冲数	19H	1BH	20H	24H	22H	10H
镀锡线送进长度	28mm	30mm	35mm	40mm	38mm	32mm

镀锡线送进流程图与程序如表 8 所示。

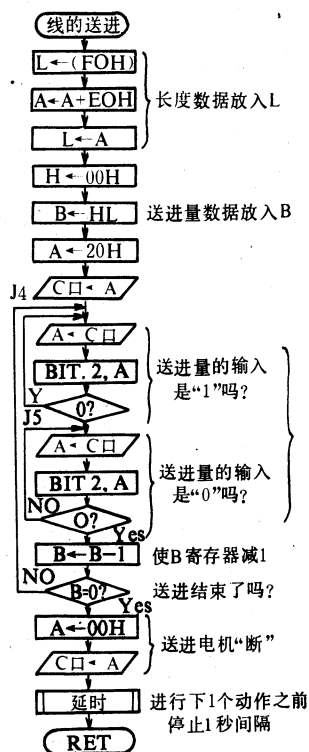
#### (5) 镀锡线的切断

电磁铁  $S_2$  与剪钳连接, 电磁铁  $S_2$  通电, 它拉动剪钳切断镀锡线。流程图与程序如表 9 所示。电磁铁  $S_2$  通电时间延时 1 秒。延时流程图与程序如表 10 所示。

#### (6) 镀锡线弯曲

设置在模具左右两侧电磁铁  $S_1$  通电, 弯曲镀锡线。流程图与程序如表 11 所示。电磁铁通电时间延时 1 秒, 也使用前述的延时程序。

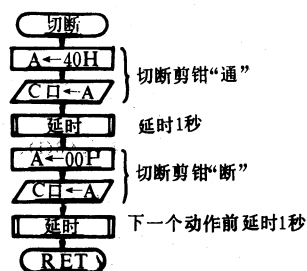
表 8



标号	助记符	地址	机器语言	注释
OKURI	LDA, (FOH)	66	3A F0 00	长度数据装入 L
	ADD A, EOH	69	C6 E0	
	LD L, A	6B	6F	
	LD H, 00H	6C	26 00	
	LD B, (HL)	6E	46	送进量数据放入 B
	LD A, 20H	6F	3E 20	送进电机“通”
	OUT(02H), A	71	D3 02	
J4	IN A, (02H)	73	DB 02	送进量输入是“1”吗?
	BIT 2, A	75	CB 57	
	JPZ, J4	77	CA 73 00	
J5	IN A, (02H)	7A	DB 02	
	BIT 2, A	7C	CB 57	送进量输入是“0”吗?
	JPNZ, J5	7E	C2 7A 00	
	DEC B	81	05	使 B 减 1
	JPNZ, J4	82	C2 73 00	送进结束了吗?
	LDA, 00H	85	3E 00	送进电机“断”
	OUT(02H), A	87	D3 02	
	CALL TIMER	89	CDB5 00	延时 1 秒
	RET	8C	C9	

标号	助记符	地址	机器语言	注释
	DB 19	E1	19	送进量 8cm
	DB 1B	E2	1B	送进量 10cm
	DB 20	E3	20	送进量 15cm
	DB 24	E4	24	送进量 20cm
	DB 22	E5	22	送进量 18cm
	DB 1D	E6	1D	送进量 12cm

表 9

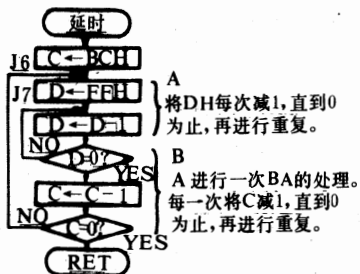


标号	助记符	地址	机器语言	注释
SETSUD	LD A, 40H	8D	3E 40	切断剪钳“通”
	OUT(02H), A	8F	D3 02	
	CALL TIMER	91	CDB5 00	延时 1 秒
	LD A, 00H	94	3E 00	切断剪钳“断”
	OUT(02H), A	96	D3 02	
	CALL TIMER	98	CDB5 00	延时 1 秒
	RET	9B	C9	

#### (7) 根数减法运算

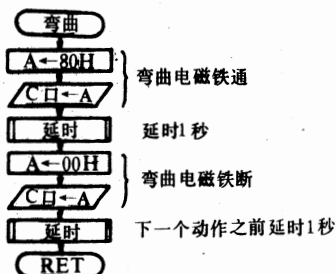
运算只能在 A 累加器内进行, F1H 地址内容为加工根数, 要传送到 A 累加器。每加工 1 根跨接线要进行减 1 运算, 并变为 10 进制数。μP-804 位数字显示部件 7 段 LED 右两位显示余数。根数减法运算流程图与程序如表 12 所示。

表 10



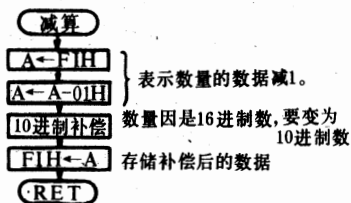
标号	助记符	地址	机器语言	注释
TIMER	LDC ,BCH	B5	0E BC	延时1秒
J6	LDD ,FFH	B7	16 FF	
J7	DEC D	B9	15	
	JP NZ, J7	BA	C2 B9 00	
	DEC C	BD	0D	
	JP NZ J6	BE	C2 B7 00	
	RET	C1	C9	

表 11



标号	助记符	地址	机器语言	注释
MAGE	LDA, 80H	9C	3E 80	弯曲电磁铁“通”
	OUT(02H), A	9E	D3 02	
	CALL TIMER	A0	CD B5 00	延时1秒
	LDA, 00H	A3	3E 00	弯曲电磁铁“断”
	OUT(02H), A	A5	D3 02	
	CALL TIMER	A7	CD B5 00	延时1秒
	RET	AA	C9	

表 12

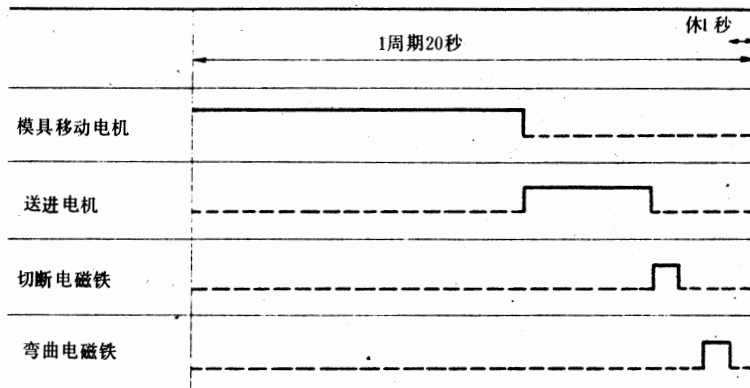


标号	助记符	地址	机器语言	注释
GENSAN	LDA, (F1H)	AB	3AF 100	根数数据减1
	SUB 01H	AE	D6 01	
	DAA	B0	27	10进制补偿
	LD(F1H), A	B1	32 F 100	存储补偿后的数据
	RET	B4	C9	

表 13

开关No	开关方向
①	“通”侧
②	读出侧
③	选择侧
④	通常侧
⑤	实行侧
⑥	端口A、端口B侧

表 14



## (8) 程序写入与检验

照图 9 连接  $\mu p-80$ , 各开关的方向如图中所示。由 00H 地址到 EDH 地址顺序写入程序。写入完毕后, 再由 00H 地址开始, 顺序调出检验程序写入是否正确。查出错误之处, 要进行修改并再写入。FOH 地址内容为长度, F1H 地址内容为加工根数。

## (9) 运转

- ①首先按动接口印刷线路板上的按键开关 SW<sub>3</sub>, 切断镀锡线剪钳运动到最前位置。
- ②图 9 的①~⑥开关方向如表 13 所示。
- ③驱动源电机、电磁铁时电源开关为“通”状态。
- ④按动中央处理部件的复位开关, 时间为 1 秒, 开关⑤在实行侧, 跨接线弯制机则开始运转。加工时序表如表 14 所示。

⑤加工完所需要的数量, 4 位数字显示部件 7 段 LED 右边两位显示 00, 弯制机运转停止。

## (10) 软件改进

本文软件举例中, 每次输入 1 种跨接线的弯曲长度、数量等信息。改进软件程序, 输入不同长度的多种跨接线的信息, 也能进行多品种、大批量生产。1 个品种生产结束后, 采用蜂鸣器、放大器等方法, 通告操作者, 这样, 将会提高生产效率。本文举例中, 精简基本程序, 由多个子程序很方便地组成完整的程序。今后, 在程序中为添加补充多种功能, 采用补充子程序方法也较为方便, 容易得到较为完整的程序。





# 如何用 EPROM 读写器对 EEPROM 编程

闫怀兵 陈立山

EEPROM 电擦除存储器用在控制电路中,可以方便的更新用户的一些控制参数,随写随擦,停电后保存信息,加电后又可改写数据,这种芯片可进行 10000 次以上的擦写工作,但是对于 EEPROM 在初始化的编程与写入时,它与传统上应用的 EPROM(只读存储器)写

入与读出的方法不相同,因而需要专门的 EEPROM 写入器才能实现,笔者在使用电擦除的 EEPROM 时,仅将使用的紫外线擦除的 EPROM 写入器略加改动,就可以很方便的电擦除的 EEPROM 进行编程写入,其方法如下:

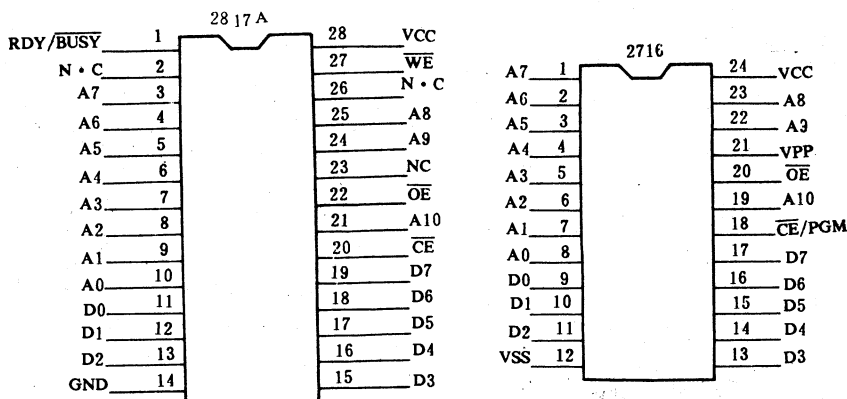


图 1

用一台普通的 PC 机 EPROM 写入器,按照欲写入的 EEPROM 芯片,选择管脚与之相似的 27×× 系列 EPROM 芯片,在电路上略加改动便可用 EPROM 写入器写入,如:电擦除芯片 EEPROM 2817A,它是 28 脚芯片,而 2716 是 24 脚芯片,(参看图 1),仅和 2817A 的上四脚,即 1,2,28,27 四脚不相同。2817A 的 28 脚是 Vcc,2716 的 24 脚是 Vcc,而 2817A 的 26 脚正好是 N.C(空脚),从上所知,若要对 2817A 进行读出时,只要将其插入 2716 的 24 脚管座,然后从第 24 脚引一条线接在 2817A 的第 28 脚上,此时便可以按照读 2716 的方式进行读出,无须改动原 2716 的驱动程序便可。

当进行写入时,从图中得知,与 2716 第 21 脚 VPP 对应的 2817A 的 23 脚为 N.C.第 1 脚是 RDY/BUSY,这是 2817A 的专用写入查寻信号端,当正在写入时 RDY/BUSY 输出低电平,表示忙,当写入结束时 RDY/

BUSY 输出高电平,表示写入结束,通知 CPU 可以写入下一个字节,据此,可利用 PC 机中的 CPU READY 端,当它为低电平时,停止数据传送,处于等待状态,当其变为高电平时,将完成数据传送,由此可见它的逻辑功能正好与 2817ARDY/BUSY 所要的逻辑状态完全相同,而等待信号可由 PC 机 62 芯 I/O 扩展槽上的 A10I/O CHRDY 端控制产生,所以,便可从 A10 上引一条线到 2817A 的第 1 脚,作为查寻写入是否完毕的控制信号。此外,2817A 的第 27 脚 WE 是写入控制信号,当 WE 为低时,写入有效,因此我们完全可以用 PC 机 62 总线 I/O 插槽上的 MEMW 信号控制它,将 B11 连接到 2817A 的第 27 脚,最后运行原 PC 机中的程序一个字节一个字节地自动写入到 2817A 中采用这种方法即可省略一套写入器,又不必改动写入驱动软件,也无须去计算写入的时间控制参数,经我们多次写入,从未失误,简单可靠。

# LASER—310 机加接数字音响发生器

王建峰

SOUND-AY-3-8914 是数字音响发生器,可以与 8 位机配合工作,创造各种音色,丰富微电脑的使用范围。关于它的详细原理可以参见《无线电》86 年第三期,本文只向大家介绍它在 LASER 机上的实际应用方法,有兴趣的读者可以一试(同功能的芯片还有 AY--3-8910 和 AY-3-8912)。

连接线路非常简单,只使用一个非门,为了降低产生的声音频率,还可以在时钟输入端加接分频器。由于所有端口均与 TTL 电平兼容且为三态线,所以使用方

度、音乐频率、噪音频率(或不输出噪音)和采用包络音形,见表一。

幅度寄存器可以存入 0~15(即有 16 档音量)当放入 255 时即为包络发生器控制。频率寄存器高位可放入 0~15,低位 0~255,产生的频率为  $f=f_q/16/(高位 * 256 + 低位)$ 。

另外有用的寄存器还有包络周期寄存器,地址为 243(高位)和 247,均可存入 0~255 的数据;包络形状寄存器,地址 250,可存 0~15;噪音频率地址 249,可

存 0~31;还有一控制寄存器,功能见表二,当某一位为 0 时,可以打开相应通道(反之关闭该通道)。

向这些寄存器写数据采用以下程序,读者可通过实验来了解该芯片的使用。

LDA,x→选取寄存器  
LD(FFFFH),A→锁存地址

LDA,y→需写入之数据

OUT(FDH),A→发出 IORQ 信号,写入数据

RET  
把它作为 BASIC 程序的子程序,使产生各种声音讯号。以下提供一简单程序,用来演奏单音乐曲。使用该

芯片,音质优美,数据编写简易,读者可以分析改编,用于演奏音域更宽,具有和声、节奏和一定动态的乐曲,同样可制造枪炮、海浪等音色,为电子游戏配声。

注:DATA 的编写:用此程序可以演奏 1~7 的 C 大调乐曲。DATA 的形式为一组二个数,前者为声调,可用 1~7 代表音乐中的 1~7,用 0.1~0.7 代表 1~7,

10~70 代表 1~7,1.5、2.5 等表示 #1、#2,其它类推;后者为节拍时值,乐谱中时值为多少就取多大。(如一拍写 1,半拍写 0.5……)

】List

10 INPUT "SPEED(1-100)";SPEED

20 FOR J=-1 TO 7

30 READ X:POKE -32767+J,X:POKE -32758+J,X

40 NEXT

50 DATA 62,0.50,255,255,62,0.211,253

60 POKE -32750,201,POKE 30862,0:POKE 30863,128

(下转 31)

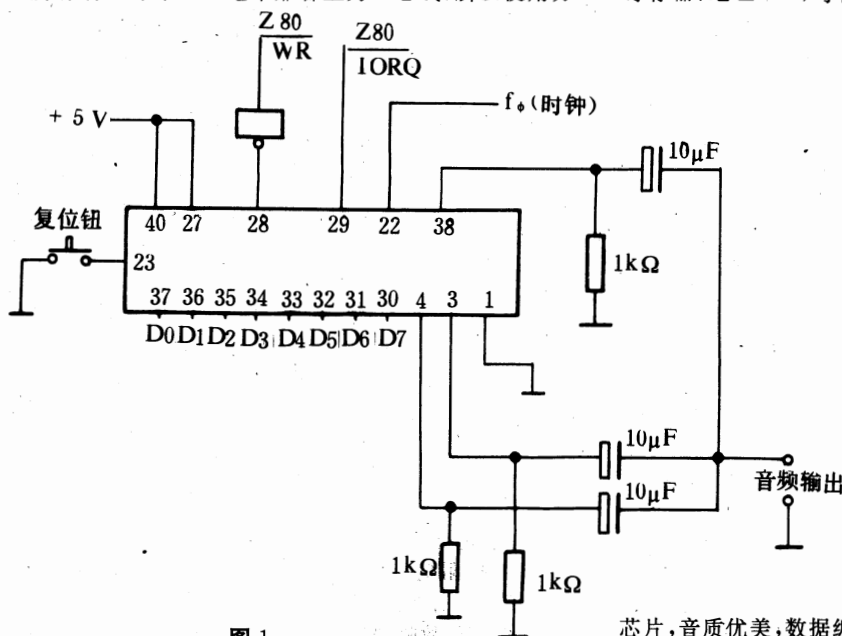


图 1

便,工作可靠。如图 1 所示,实质上是把该芯片作为一个只写外设。通过实验,得出以下有用数据:

表一

通道	幅度	频率	
		高位	低位
A	251	240	244
B	252	241	245
C	253	242	246

(表一:数据为寄存器地址)

表二

音质	/	/	N	N	N	M	M	M
通道	/	/	C	B	A	C	B	A
二进制码位	B7	B6	B5	B4	B3	B2	B1	B0

(表二:M—music 乐音,N—noise 噪音)

该芯片共有三个音响通道,它们可以分别控制幅



## 维修经验谈

# 任天堂系列游戏机检修实例(续)

林华春

**例十、故障现象：**一台红白 LT-900D 游戏机，I 号操纵手柄操作时无规律出现暂停，将 II 号手柄拔下，I 号手柄才能正常使用。

**分析与检修：**首先怀疑 II 号手柄中有断线或碰线，经检测后没有发现断、碰线，根据暂停现象时有时无，确认故障还在 II 号手柄操纵盒中，测量 4021 各脚电位均正常，换一片新 4021 后故障仍存在，经仔细检查后发现 4021 的⑭脚与电源 5V 的联接印刷电路板有开路现象，焊接后故障排除。此现象说明 CMOS 电路不使用的空脚一定不能悬空。

**例十一、故障现象：**一台任天堂 616 游戏机，有图像，有伴音，但图像上下跳动。

**分析与检修：**该游戏机在输出插口处有一兼容变换开关，当遇到不兼容的游戏卡时，可拨动此开关，游戏可正常进行。出现上述故障时先把兼容开关拨向另一边，但图像无变化，然后用逻辑笔按 4011 四与非门的真值表，检测其输出端的逻辑变化关系，发现一路输出无变化，换一片 4011 后故障排除。

**例十二、故障现象：**一台红白游戏机，在同是 PAL-D 制的彩电上使用，有些彩电显示彩色图像，而有些彩电只显示黑白图像，此时伴音正常。

**分析与检修：**出现上述故障的原因有两方面，一是彩电中彩色副载波捕捉频率窄，二是游戏机中晶振频率偏移。对于不同型号的同制式彩电，虽然其彩色副载波的频率同为 4.43MHE，但捕捉频率范围宽的彩电，其彩色还原好，也就是能够显示出彩色图像的彩电，若捕捉频率范围窄的彩电，游戏机晶振频率稍有偏移，即

不能还原彩电图像。因此我们从游戏机方面来排除上述故障，方法是改变与晶体 26.601712MHE 串联的电容，使电容量适当增加或减少，彩色即可还原，经上面方法处理后故障排除。

**例十三、故障现象：**一台红白 737 游戏机，图像正常，无伴音。

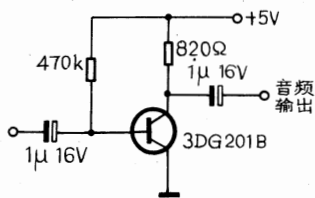
**分析与检修：**从电视荧光屏图像看，画面清晰，因此可以判断故障是由游戏机伴音电路损坏引起。用 2200Ω 高阻抗耳机，听 CPU 的①、②脚，有伴音信号；听 74HC368 的⑬脚也有伴音信号；听 74HC368 的⑭脚，无伴音信号，这说明 74HC368 损坏。由于游戏机其它功能正常，可确定 74HC368 只有伴音一路损坏，所以采用外接放大电路解决，电路如附图，接好后开机，伴音恢复正常。

**例十四、故障现象：**一台红白无型号游戏机，图像正常，伴音失真。

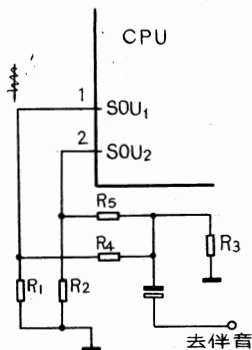
**分析与检修：**首先用小螺丝刀微调 RE 射频调制盒中的伴音振荡线圈，失真现象无变化，因此确定故障不是 RF 射频调制盒有故障。沿伴音电路用高阻抗耳机听伴音信号，听 CPU 的①、②脚有伴音信号；听输出电阻 R<sub>5</sub> 另一端（见附图）却发现缺一路信号，经仔细辨别后认定是 CPU②脚的伴音信号没有送出。用万用表测试 R<sub>5</sub> 阻值，该电阻已开路，用新电阻替代后，伴音失真现象消除。

**例十五、故障现象：**一台红白游戏机，图像正常，伴音极小。

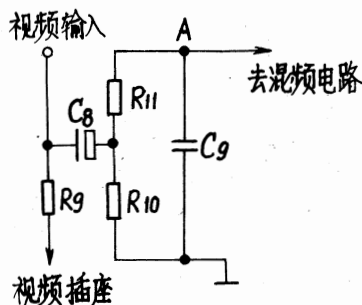
**分析与检修：**首先按例十四的方法听伴音信号，在



例十三



例十四



例十七

RF 射频调制盒以前电路均正常,微调伴音振荡线圈后,伴音信号增大,但图像却出现扭曲,再调图像载波振荡线圈,伴音信号又消失,用万用表检查射频调制电路后发现伴音振荡线圈次级开路,重绕后开机,经过常规调整伴音、图像均恢复正常。

例十六、故障现象:一台灰白 767 游戏机,无图像,无伴音。

分析与检修:按正常方法检修电路板电路均正常,偶然碰一下射频盒,发现伴音线圈很热,启开盖后看到,这种射频调制盒电路采用了三只三极管,从印制板上的位置及电路原理分析,靠近伴音振荡线圈外侧的一只三极管为射频放大管,用万用表测量时,发现其已击穿,换新后开机,有图像,仍无伴音。拆下伴音振荡线圈,发现其内部断线,重绕后再开机,伴音、图像均正常。

例十七、故障现象:一台红白游戏机,有伴音,图像

上有不同步的斜条。

分析与检修:用示波器测 CPU 的②脚,有正常的视频信号,在射频盒视频输入端测量,也有正常的视频信号,启开射频盒,见附图。测  $C_8$  两端均有视频信号,当测试到 A 点时无视频信号,断电后测量,发现  $C_9$  电容漏电严重,换新后伴音、图像恢复正常。

例十八、故障现象:一台红白任天堂 HVC-001 游戏机,开机后图像上有两条水平干扰带,伴音中有严重的交流声。

分析与检修:首先用万用表测量整流器输出的直流电压,为 12V 正常,然后测 +5V 输出电压发现不稳定,随即更换 7805 稳压 IC,开机后仍不正常,焊下电源滤波电容后发现漏电,换新后开机,干扰带去掉,但伴音中仍有交流声,用小螺丝刀调整 6.5MHz 伴音振荡线圈后,伴音恢复正常。

## “8088 系统板测试卡”对系统板总线部分的诊断测试

梁杰熙

“8088 系统板测试卡如果不能以连续方式测试时,说明故障在系统板的总线部分。此时,就要使用测试卡的单步诊断测试部分。单步诊断测试是通过测试卡上的状态来分析判断故障发生在系统总线的哪一点。所以需要对系统板总线的结构及工作状态,有一个简单的了解,才能正确的使用测试卡和判断其故障。

一、8088 微型计算机的总线结构:

1. 总线结构的划分:

以 8088 为 CPU 的微型计算机其总线可划分为地址总线,数据总线,控制总线。地址总线 and 数据总线是通过一级或多级缓冲器进行传输的。IBM PC/XT 及其兼容机和国产 0520 系列机中,地址总线 and 数据总线一般都使用的 74LS373、74LS244 和 74LS245 芯片。控制总线是 8088CPU 用于发送系统控制命令的,在 8088CPU 的微型计算机中由于 8088 是工作于最大模式,其控制命令不是由 8088 本身发送的,而是通过一

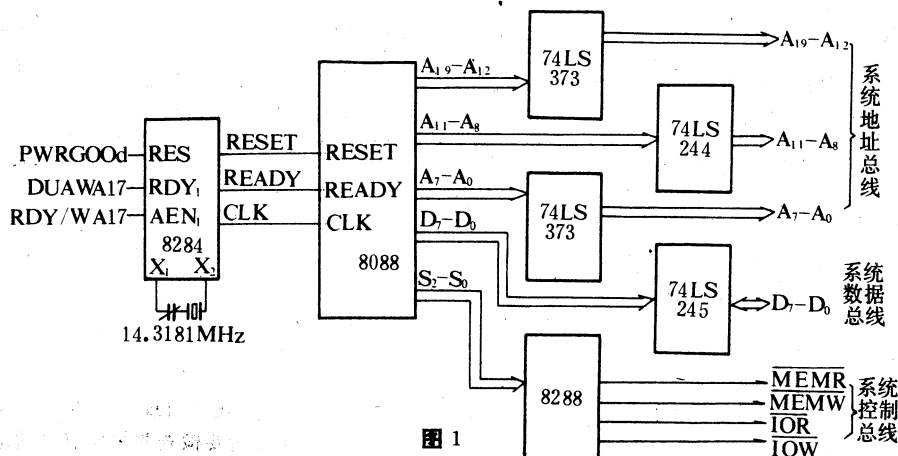


图 1

个总线控制器由 8288 提供发送。下面以 IBM PC/XT 机为例,用图 1 来描述其地址总线,数据总线,控制总线的结构:

## 2. 总线结构下的工作状态:

8088 在工作时主要有三个控制信号,RESET(系统复位),READY(外设准备好),CLK(CPU 时钟)。由图“1”看到 RESET,READY,CLK 都是 8284 时钟发生器提供给 8088 的,在系统加电时,这三个信号的状态如下:RESET 信号:微机电源箱的四种直流电压输出(+5V,+12V,-5V,-12V)正常后,送出 50 $\mu$ s 的低电平 PWRGood(电源好)信号,这个信号送到 8284 的  $\overline{\text{RES}}$  端,经 8284 控制输出高电平的 RESET 信号,其需要保持 50 $\mu$ s 时间才能使 8088 达到稳定的工作状态。READY 信号:8284 的准备好控制输入端 RDY1,AEN1 分别接受来自外围电路的  $\overline{\text{DMAWAIT}}$ , $\overline{\text{RDY}}$ /WAIT 的信号状态,在  $\overline{\text{DMAWAIT}}$  为高电平和  $\overline{\text{RDY}}$ /WAIT 为低电平时,8284 的 READY 输出端为 8088 提供了有效的高电平信号。反之在  $\overline{\text{DMAWAIT}}$  为低电平或  $\overline{\text{RDY}}$ /WAIT 为高电平时 8284 的 READY 输出端为低电平,此时 8088 将插入等待状态,直到 READY 输出端恢复为高电平 8088 才脱离等待状态,进入 8088 工作周期。CLK 信号:8284 的 X<sub>1</sub>,X<sub>2</sub> 引脚连接晶体振荡器,在 8088CPU 的微型计算机中,一般晶振频率选择的都是 14.3181 MHE 的晶体振荡器,但现在有些 8088CPU 的兼容机在系统设计时为了提高其运行速度选择 24MHE,30MHE 等频率的晶体振荡器。但是其连接方式都是将晶体振荡器接至 8284 分频后提供给 8088 的 CLK 时钟端。以上的 RESET,READY,CLK 信号状态满足后,8088 就要通过地址缓冲器,数据缓冲器和总线控制器对微机系统进行,A<sub>0</sub>~A<sub>19</sub> 的地址发送,D<sub>0</sub>~D<sub>7</sub> 的数据传输和 S<sub>0</sub>~S<sub>2</sub> 的总线控制状态的发送。地址总线和数据总线通过缓冲器其输入输出状态是相对应的,控制总线的输入输出状态是通过总线控制器由 S<sub>0</sub>~S<sub>2</sub> 的输入状态来决定  $\overline{\text{MEMR}}$ , $\overline{\text{MEMW}}$ , $\overline{\text{IOR}}$ , $\overline{\text{IOW}}$  等的输出状态,在 8088 进行正常工作时这些输入输出信号都应是在脉冲状态。

## 二、8088 系统板测试卡对总线测试的原理。

8088 微处理器的总线工作周期分成四个时钟周期,T<sub>1</sub> 周期(总线地址发送周期),T<sub>2</sub> 周期(总线数据传输周期),T<sub>3</sub> 周期(8088 采样 READY 信号周期),T<sub>4</sub> 周期(恢复周期准备下一个总线工作周期)。这里要说明的是 8088 采样 READY 信号周期,即 T<sub>3</sub> 周期。在 8088CPU 的微机系统中 I/O 接口电路和一些慢速的存储器其数据存取的响应时间较慢,在整个系统总线工作周期频率速度下,不能获取正确的数据结果,此时需要在 8088 的总线工作周期内插入等待状态,这个等待状态取决于 T<sub>3</sub> 周期开始时的信号下降沿 8088 采样 READY 信号,若 READY 信号为低电平,就在 T<sub>3</sub> 周期结束后插入等待状态周期 Tw,在 Tw 周期开始时继续对 READY 信号进行采样,当 I/O 设备完成数据传输后 READY 信号变为高电平,8088 在采样到 READY

信号为高电平则此时总线工作周期进入 T<sub>4</sub> 周期准备下一个总线工作周期。(见图“2”插入等待状态的总线工作周期时序图:

这种使总线工作周期插入等待状态让其总线工作状态一个周期一个周期的执行。因为系统总线在正常工作时的某一个周期中其信号是处在一定的有效状态下的只有在下一个时钟周期时其状态才发生变化,这样连续起来才呈现的是一个脉冲状态,我们把动态连续的总线状态通过插入等待状态的方法将其分成稳定单步(一个周期)的状态,再通过我们编制的诊断测试程序来测试是哪一点先发生了故障。此如说在插入等待状态一步一个周期的执行,在执行到某一个周期时按诊断测试程序给出的正确数码值某一个信号点的值应为高电平信号状态,此时呈现在测试卡上确为低电平状态,这就说明这个信号点是故障点。所以我们在对系统总线诊断测试时将其称为对系统总线的单步诊断测试。

## 三、使用“8088 系统板测试卡”如何诊断测试总线故障

### 1. 测试卡的单步诊断测试:

测试卡如果不能使用连续方式诊断测试系统板故障时,就要使用测试卡的单步诊断测试部分。测试卡将“L”处短接,改换成“D”处短接。见图“3”“L”和“D”的短接点是测试卡的连续测试和单步测试的选择点。在单步诊断测试时,测试卡的右上角有一个标有 W<sub>1</sub> 的按动开关,在未按下时,一直向 8088 发送  $\overline{\text{READY}}$  低电平信号,让 8088 系统总线保持在第一个初始周期状态下。当按下按动开关 8088 测试卡就向 8088 发送一个高电平 READY 信号后马上恢复成低电平。使 8088 系统总线又保持在第二个周期状态下,当按动一下开关系统总线就向下执行一个周期并稳定在这个周期状态下。来达到走一步观察一次系统总线状态的目的。系统总线的状态在测试卡上是体现在 8088 系统板测试卡上标有 A<sub>0</sub>~A<sub>19</sub>,D<sub>0</sub>~D<sub>7</sub> 的发光二极管上,其状态是用二进制的方法表示,亮时表示这位为“1”,不亮时表示这位为“0”。在与测试卡上诊断测试程序给出的数码值(十六进制)要进行二进制与十六进制的转换后再比较其正确的对应关系,来分析判断其故障点。

### 2. 如何根据测试卡上的状态判断系统总线的故障点:

板测试卡在单步测试总线状态时,第一步呈现在测试卡上的地址总线的状态(A<sub>0</sub>~A<sub>19</sub>)应是 FFFF0 数据总线的状态(D<sub>0</sub>~D<sub>7</sub>)应是 EA。为了说明问题我们列出测试卡上诊断测试程序给出的前 10 步的正确数码表:

第 1 步	FFFF0 EA	第 6 步	FFFF5 00
	FFFF1 77		FF877 FA
	FFFF2 F8		FF878 B8
	FFFF3 00		FF879 AA
第 5 步	FFFF4 F0	第 10 步	FF87A AA

如果第一步状态出现多位不对则说明 8088 根本没有工作,这时需要检查的故障点是满足 8088 正常工



作时的三个信号的状态。即 RESET, READY, CLK, RESET 信号在开机加电时必须有一个由高向低的时间为  $50\mu s$  的跳变状态, READY 信号必须保持在高电平状态, CLK 的时钟频率必须在系统所要求的时钟频率范围内, 比如 IBM PC/XT 机其频率为 4.77MHz。如果这三个信号都满足了说明 8088 芯片本身损坏, 如果这三个信号有一个不满足则说明为 8088 提供这三个信号的 8284 时钟发生器损坏, 或是晶体振荡器损坏。晶体振荡器其两极的对地电压平均值在 2.5V 左右说明晶体振荡器没有故障, 而是 8284 损坏, 如果晶体振荡器两极对地的平均电压达不到 2.5V 左右, 则说明是晶体振荡器损坏, 造成了 8284 无法向 8088 提供 CLK 时钟, 至使 8088 不能工作。如果第一步呈现在 8088 系统板测试卡上的地址状态是正确的与数码表相符, 而数据总线的状态是多位不对与数码表相差甚远, 则说明故障出现在控制总线上。因为在 8088 进入正常工作时, 将其地址发送出去后, 接着在其进行数据传输的同时要发送出总线的控制命令, 即现在的数据传输是进行存储器读操作 (MEMR) 还是进行存储器写操作 (MEMW), 或是进行 I/O 读操作 (IOR) 还是进行 I/O 写操作 (IOW)。如果控制总线送不出这些总线控制命令, 那么传输的数据就无法正确的呈现在测试卡上, 所以必然是控制总线出现了故障。控制总线有故障也就是 8288 总线控制器有故障, 因为 8284 是发送总线控制命令的控制芯片。在第一步的地址总线, 状态和数据总线状态与数码表相符后, 此时也就可以说 8088、

8284, 晶体振荡器, 8288 这些芯片没有问题, 这样就可以按动一下  $W_1$  按动开关一步一步的向下执行。在一步一步向下执行的过程中如果要发现呈现在 8088 系统测试卡上的地址总线状态或数据总线状态与数码表不相符, 此时则说明故障出现在地址总线缓冲器上或是数据总线缓冲器上。如果呈现在 8088 测试卡上是地址总线的状态与数码表不符, 则就要与数码表上的值找出每位的对应关系。比如在执行到第 6 步时出现了地址总线状态与数码表不符, 正确的值应是 FFFF5, 而呈现在 8088 系统板测试上的地址总线状态确是 F0FF5, 此时将这个十六进制的状态数值按二进制展开, 对应到地址总线的每一位上, 即  $F \rightarrow A_{19} \sim A_{16}$ ,  $0 \rightarrow A_{15} \sim A_{12}$ ,  $F \rightarrow A_{11} \sim A_8$ ,  $F \rightarrow A_7 \sim A_4$ ,  $5 \rightarrow A_3 \sim A_0$  则看到  $A_{15} \sim A_{12}$  对应的那一位为“0”。也就是说在单步走到第 6 步时对应地址总线  $A_{15} \sim A_{12}$  的那一位缓冲器芯片出现故障了, 再由图纸或线路板上找到  $A_{15} \sim A_{12}$  对应的那个缓冲器芯片, 将其换下后故障一定排除。同样在单步执行到某一步时如果出现数据总线的状态与数码表不相符, 也将其进行二进制展开来对应数据总线的每一位, 找到其对应的缓冲器芯片将故障排除。总体来说单步测试的第一步是对 8088 8284 晶体振荡 8288 的诊断测试在第一步通过后的测试则是对地址总线缓冲器和数据总线缓冲器的测试, 它把系统总线结构内的所有芯片都测试到了, 判断出故障点所在位置。

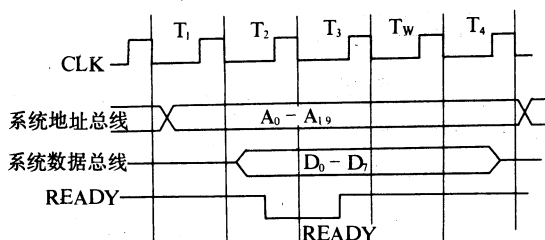


图 2

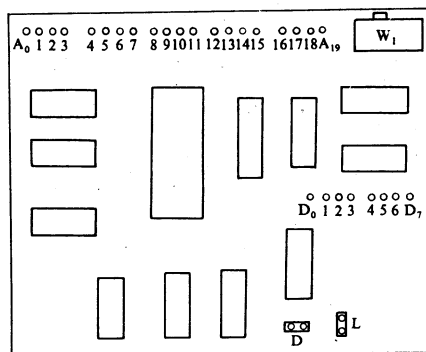


图 3

《电子与电脑》1991 年上半年合订本已经出版, 每本订价 7.8 元, 欲购者请附加 15% 邮费。由于数量有限, 需要者请从速。

邮购办法: 邮局汇款 北京万寿路 173 信箱电子工业出版社《电子与电脑》编辑部

银行汇款 开户行北京市工商银行翠微路分理处帐号 891238-72

# 一九九一年计算机初级程序员有奖竞赛试题

主办单位：中国软件行业协会考试指导中心、中国高档微机协会、《电子与电脑》编辑部

一、从供选择的答案中选出正确答案，把相应的字母写在答卷纸的对应栏内。（每空1分，共40分）

1. 二进制数1101110.01的十进制数表示是\_\_\_\_，十六进制数表示是\_\_\_\_；十六进制数50A.E的十进制数表示是\_\_\_\_，二进制数表示是\_\_\_\_，八进制数表示是\_\_\_\_。

供选择的答案：

- A. 110.25    B. 108.625    C. 1290.85    D. 1280.85    E. 2412.7  
F. 1414.7    G. 3D.25    H. 6E.4    I. 10100001010.111  
J. 101001100.0111

2. 设数 $X=0.1001$   $Y=-0.0101$ ，则 $[X+Y]$ 补=\_\_\_\_， $[X-Y]$ 补=\_\_\_\_。

供选择的答案：

- A. 10.0010    B. 0.0100    C. 1.011    D. 0.1110

3. 在计算机中表示0时\_\_\_\_码是唯一的，且全为0。

供选择的答案：

- A. 国标    B. 原    C. 补    D. 反

4. 浮点数取值的大小由\_\_\_\_决定，而浮点数的精度由\_\_\_\_决定。

供选择的答案：

- A. 阶码的位数    B. 阶码的数值    C. 尾数的位数    D. 尾数的数值

5. 设A, B, C为逻辑变量，则逻辑表达式 $(A+B)(A+\bar{B})$ 可化简为\_\_\_\_， $AB+\bar{A}C+\bar{B}C$ 可化简为\_\_\_\_。

供选择的答案：

- A.  $AB+C$     B.  $AC+B$     C.  $AB$     D.  $A$

6. 对二进制数1011001进行奇校验，其校验位的值为\_\_\_\_。

供选择的答案：

- A. 4    B. 2    C. 0    D. 1

7. 操作系统是对计算机系统的全部资源进行控制与管理的系统软件。系统资源指的是\_\_\_\_。

供选择的答案：

- A. 软件、数据、硬件、存储器    B. 处理机、存储器、输入/输出设备、信息  
C. 程序、数据、输入/输出设备、中央处理器    D. 主机、输入/输出设备、文件、外存储器

8. PC DOS系统的命令处理程序COMMAND.COM的功能是\_\_\_\_。

供选择的答案：

- A. 负责接受和解释用户输入的命令及出错处理，具有全部内部命令的处理程序  
B. 负责出错处理，具有全部内部命令、外部命令的处理程序  
C. 负责接受和解释用户输入的命令，具有全部内部命令、外部命令的处理程序  
D. 负责出错处理，具有全部内部命令、外部命令的处理程序

9. 当系统显示如下错误信息:Bad command or file name时，应从两方面进行检查\_\_\_\_。

供选择的答案：

- A. 磁盘驱动器是否关好小门；磁盘是否贴有写保护  
B. 磁盘是否贴有写保护；命令字及格式拼写是否正确  
C. 查看命令拼写是否正确；查看磁盘上是否有该文件  
D. 内存中是否有该文件；是否为可执行文件

10. 若要把 A 驱动器内DOS系统盘上的文件内容全部拷贝到空白的硬盘上，应采用如下方法\_\_\_\_。

供选择的答案：

- A. 使用 COPY A:\*. \* C: 命令    B. 使用 DISKCOPY A: C: 命令  
C. 先使用 SYS C: 命令，    D. 先使用 FORMAT C:/S 命令，  
再使用 DISKCOPY A: C: 命令    再使用 COPY A:\*. \* C: 命令

11. COPY文件操作意味着\_\_\_\_

供选择的答案：

- A. 在同一个介质内产生一个同名同内容的文件  
B. 产生一个同内容的文件  
C. 传送一个文件，原文件消失  
D. 只能在另一设备介质上产生一个不同名而内容相同的文件

12. AUTOEXEC.BAT 文件是自动批处理文件, 它的主要内容是\_\_\_\_; CONFIG.SYS文件内容是\_\_\_\_\_。

供选择的答案:

- A. 建立或编辑文件的一系列子命令
- B. 设置系统结构的命令清单
- C. 启动后希望立即自动执行的一系列命令
- D. 合并单独产生的目标模块, 控制被测试程序运行的命令

13. 在微机操作过程中磁盘驱动器指示灯亮时不能插取磁盘的原因是\_\_\_\_\_。

供选择的答案:

- A. 会损坏磁盘驱动器
- B. 可能将磁盘上的数据破坏
- C. 影响计算机的使用寿命
- D. 内存中的数据将丢失

14. 文件的组织包括文件的逻辑结构和物理结构两方面。文件的逻辑结构是指\_\_\_\_; 物理结构是指\_\_\_\_\_。

供选择的答案:

- A. 一个文件内部各部分之间的关系
- B. 文件由记录组成或为有序字符的集合
- C. 文件在外存上的存放方式
- D. 采用连续结构或链接结构的文件

15. 使用结构化程序设计的方法, 使得软件的\_\_\_\_, \_\_\_\_加强。

供选择的答案:

- A. 可读性
- B. 可靠性
- C. 可移植性
- D. 可操作性

16. 按照国家标准GB2312-80规定, 一个汉字由\_\_\_\_\_个字节组成, 该码表共分\_\_\_\_\_区, 每个区有\_\_\_\_\_位, 共有\_\_\_\_\_个汉字, 第一个汉字在\_\_\_\_\_区的第一位。

供选择的答案:

- A. 2
- B. 3
- C. 9
- D. 16
- E. 85
- F. 94
- G. 3755
- H. 3008
- I. 6763
- J. 7445
- K. 8836

17. \_\_\_\_\_不是计算机汉字处理过程中采用的数据结构。

供选择的答案:

- A. 字形表示
- B. 内部码
- C. 外部输入码
- D. 打印输出码

18. BNF是一种描述\_\_\_\_\_的语言, 最先被用于\_\_\_\_\_语言。

供选择的答案:

- A. 高级语言
- B. 汇编语言
- C. 语法
- D. 算法
- E. ALGOL
- F. PASCAL
- G. FORTRAN

19. 数据库管理系统的功能不包括\_\_\_\_\_。

供选择的答案:

- A. 数据库构造
- B. 输入/输出调度
- C. 缓冲区管理
- D. 程序的存储分配

20. 数据库中常用的数据模型不包括\_\_\_\_\_。

供选择的答案:

- A. 层次模型
- B. 网络模型
- C. 系统模型
- D. 关系模型

21. 一个过程文件中允许最多包含的过程\_\_\_\_\_。

供选择的答案:

- A. 256
- B. 128
- C. 32
- D. 28

22. 数据库操作语言一般不具有的功能是\_\_\_\_\_。

供选择的答案:

- A. 从数据库中检索数据
- B. 数据收集器和集中操作
- C. 向数据库中添加数据
- D. 用于并发访问控制的操作

23. 用数据描述语言精确地定义数据模型的程序称为模式, 对应于不同的模型也有三个相应的模式, 其中不包括\_\_\_\_\_。

供选择的答案:

- A. 线模式
- B. 外模式
- C. 概念模式
- D. 内模式

24. 以下说法\_\_\_\_\_不正确。

供选择的答案:

- A. 数据库减少了数据冗余
- B. 数据库避免了一切数据重复
- C. 数据库避免数据的不一致
- D. 数据库中的数据可以共享

25. 关系数据库操作功能不包括\_\_\_\_\_。

供选择的答案:

- A. 投影
- B. 选择
- C. 连接
- D. 抽象

二. 填空, 将对应的答案填写在答卷纸的对应栏内。(每空1分, 共30分)

1. CPU是指计算机的\_\_\_\_\_和\_\_\_\_\_部分。

冯·诺依曼是\_\_\_\_\_国人, 他为现代计算机结构奠定了基础, 于1946年他首先提出了\_\_\_\_\_程序的思想, 由其思想制做的计算机的特点是程序和\_\_\_\_\_指令\_\_\_\_\_执行。

3. CPU中的控制寄存器有各自的功能, 如: 程序计数器(PC)常用于存放\_\_\_\_执行指令的地址, 指令寄存器常用于存放\_\_\_\_执行指令的指令码。
4. 在一个指令周期中, CPU要执行操作的依次步骤是读指令、\_\_\_\_、取操作数、执行指令和\_\_\_\_。
5. 指令的地址段指出的是操作数本身称为\_\_\_\_寻址方式, 指令的地址段指出的是操作数的地址称为\_\_\_\_寻址方式, 指令的地址段指出的是操作数地址的地址称为\_\_\_\_寻址方式。
6. DBMS是数据库管理系统的一组软件, 它一般包括数据库管理功能、\_\_\_\_功能、建立维护功能和\_\_\_\_功能。
7. 设某软盘规格如下:  
存储面: 双面, 磁道数/每面: 77, 每个磁道的扇区数: 20, 数据长: 128字节/扇区, 那么该软盘容量是\_\_\_\_KB; 如果一个扇区能存储一张卡片的数据, 那么这张软盘能存储\_\_\_\_张卡片的数据。
8. 打印机是计算机系统的最基本的\_\_\_\_设备, 按印字方式可分为击打式和\_\_\_\_打印, 按工作方式可分为并行式和\_\_\_\_打印。
9. 计算机系统中使用的输出设备CRT的中文意思是\_\_\_\_。
10. 设有一个空栈, 栈顶指针是1FF0H, 现有输入序列2、5、0、7、6, 经过操作push, push, push, pop, push, pop, push后, 栈中存储数据(按地址存放的顺序)是\_\_\_\_, 栈顶指针是\_\_\_\_。
11. 汉字编码方案有多种, 常见的拼音码是\_\_\_\_码, 区位码是\_\_\_\_码, 五笔字型码是\_\_\_\_码; 上述三种编码可在\_\_\_\_种汉字系统上使用。
12. 信息处理是指对信息\_\_\_\_和传播的一系列活动的总和。
13. 对于微型计算机系统用户, 最常见的计算机病毒传染媒介是\_\_\_\_。

三· 填空, 将相应的答案填写在答卷纸的对应栏内。(每空2分, 共12分)

请在A盘上建立如图所示的树型目录结构:



1. 请写出显示A盘上根目录下所有扩展名为.PRG的文件目录的命令。
2. 设当前目录为根目录, 一级子目录BAS已经建立, 请写出建立二级子目录BA1的命令。
3. 设当前目录是BAS, 请写出在二级子目录BA1下建立文件AA.TXT的命令。
4. 设当前目录为根目录, 请写出把子目录BA1下的文件AA.TXT与子目录DB3下的文件AA.BAK联接成一个名为ALL.BAK文件, 存放在子目录BA2下的命令。
5. 设当前目录为BAS, 请写出把子目录BA2下的文件ALL.BAK改名为ALL.TXT的命令。
6. 设当前目录为BAS, 请写出删除子目录BA1下的文件AA.TXT的命令。

四· 设当前驱动器为A, 请在B驱动器内的盘上建立一个名为PC.BAT的批处理文件, 它具有如下功能:(共8分)

1. 清屏幕;
2. 暂停, 用中文显示考生姓名及工作单位;
3. 复制A驱动器内系统盘上的TREE.COM文件到用户盘的根目录下;
4. 在屏幕上显示用户盘的所有目录路径及子目录中的文件名;
5. 在屏幕上显示本批处理文件的内容;
6. 文件结束。

五· (共20分)

已知AAA.DBF数据库的结构和记录如下:

数据库结构: 数据库名: C:AAA.DBF

数据库中的数据记录个数: 8

数据库最后一次更新日期: 01/01/80

序号	字段名	类 型	宽 度	小 数 位
1	姓名	字 符 型	6	
2	性别	字 符 型	2	
3	出生日期	日 期 型	8	
4	民族	字 符 型	4	
5	数学	数 字 型	5	1
6	语文	数 字 型	5	1
7	平均分	数 字 型	6	2

\*\* 总计 \*\* 37

数据库记录:

Record#	姓名	性别	出生日期	民族	数学	语文	平均分
1	王红	男	12/23/78	汉	98.0	78.0	0.00
2	乌冰	女	03/07/65	蒙	98.0	67.0	0.00
3	李夺	女	10/23/77	维	87.0	78.0	0.00
4	买买切	男	02/25/64	维	99.0	94.0	0.00
5	赵充	男	06/25/69	蒙	54.0	87.0	0.00
6	卓仁	女	07/26/45	藏	78.0	58.0	0.00
7	杨林	女	09/07/55	汉	97.5	87.0	0.00
8	李洋	男	06/17/64	维	78.0	63.0	0.00

1. 假设数据库AAA.DBF已经打开, 在答卷纸对应栏内写出完成下列要求的命令: (每个命令1分, 共8分)

- ① 填充数据库AAA.DBF中平均分字段。
- ② 把记录指针移到第一个平均分大于90分的记录上。
- ③ 复制数据库AAA.DBF中全部姓王的同学的记录, 新文件名为BBB.DBF。
- ④ 对数据库AAA.DBF 中的性别和平均分字段进行索引, 新文件名为CCC.NDX。
- ⑤ 显示数据库AAA.DBF中少数民族名单, 且不带记录号。
- ⑥ 将数据库AAA.DBF中姓名、性别、平均分字段的结构拷贝出来, 新文件名为DDD.DBF。
- ⑦ 将65年底以前出生的人员情况显示出来。
- ⑧ 将数据库AAA.DBF 复制成标准格式数据文件, 新文件名为EEE.TXT。

2. 请在答卷纸对应栏内顺序写出执行下面各命令后记录指针指向的位置 (共4分, 每空0.5分)

```
.USE AAA
.2
.SKIP -1
.SKIP
.SKIP 2
.GO TOP
.LIST NEXT 8
.LIST
```

3. 请在答卷纸对应栏内写出下面程序或命令执行后的运行结果 (每空2分, 共8分)

① 程序如下:

```
SET TALK OFF
USE AAA
DO WHILE RECNO() < 6
    IF .NOT. 民族 = '维'
        SKIP
    LOOP
ENDIF
DISPLAY 姓名
SKIP
ENDDO
```

② 程序如下:

```
SET TALK OFF
USE AAA
DO WHILE .NOT. EOF()
    IF RECNO() < 6
        LIST 姓名 FOR 民族 = '维'
    ELSE
```



```

EXIT
ENDIF
ENDDO

```

③命令序列:

```

.USE AAA
.COUNT FOR 性别='男' .AND. 民族='维' .OR. 民族='蒙' TO A2
.?A2

```

④命令序列:

```

.USE AAA
.COUNT FOR 性别='男' .AND. (民族='维' .OR. 民族='蒙') TO A3
.?A3

```

六·阅读下面的BASIC程序,将运行结果填写在答卷纸的对应栏内。(每小题3分,共12分)

①程序:

```

10 M=1:A=1
20 PRINT A;" ";
30 M=M+1:A=A*M
40 IF M>10 THEN 60
50 GOTO 20
60 END

```

②程序:

```

10 INPUT N
20 FOR I= 0 TO 2*N
30 PRINT TAB(20+ABS(N-I)-N);
40 FOR J=N TO ABS(N-I)STEP -1
50 PRINT RIGHT$(STR$(J),1);
60 NEXT J
70 IF J=N-1 THEN 110
80 FOR P=J+2 TO N
90 PRINT RIGHT$(STR$(P),1);
100 NEXT P
110 PRINT
120 NEXT I
130 END

```

RUN  
? 2

③程序:

```

10 DATA -2
20 DEF FNA(X)=X*X-4
30 DEF FNB(X)=5
40 READ G
50 D=FNA(G)*SGN(G)+FNB(G)
60 DEF FNC(X)=FNA(X)-FNB(X)
70 PRINT FNA(D)
80 PRINT FNB(D)+FNC(D)
90 END

```

④程序:

```

10 READ N,P,X
20 GOSUB 50
30 PRINT P*X
40 END
50 IF N=0 THEN 80
60 N=N-2
70 GOSUB 50
80 P=P*X
90 PRINT P;
100 RETURN
110 DATA 6,2,3

```

七·下面是求300以内被6除余2,被7除余3的整数的个数及其累加和的BASIC程序,请填补空缺语句或表达式。(每空1.5分,共6分)

程序:

```
10 S=0:N=0
20 FOR I= 1 TO 300
30 IF (I-2)/6<>_____ THEN 70
40 IF (I-3)/7 <> _____ THEN 70
50 PRINT I,
60 S=_____:N=_____
70 NEXT I
80 PRINT
90 PRINT "个数N=";N,"累加和S=";S
100 END
```

八· (每小题6分, 共12分)

新华商店第一季度销售复印机情况如下:

品名	零售价	销售量 (台)	销售额 (元)
理光FT-5560	68000	15	
理光FT-4085	32800	6	
施乐1027	29850	7	
佳能NP-3825	49000	10	
东芝BD-5511	24800	8	
优美3200	32000	11	

在答卷纸上完成

1. 补充完成下面的BASIC程序, 使其具有统计第一季度总销售额及上交税额 (税额 = 销售额的20%), 并按照销售额从高到低输出复印机销售情况表 (如下):

品名	零售价	销售量	销售额
理光FT-5560	68000	15	1020000
佳能NP-3825	49000	10	490000
优美3200	32000	11	352000
施乐1027	29850	7	208950
东芝BD-5511	24800	8	198400
理光FT-4085	32800	6	196800

第一季度总销售额: 2466150元

上交税额: 493230元

2. 画出相应的程序流程图。

BASIC程序:

```
10 CLS:PRINT:PRINT:PRINT
20 DIM A(6,4),X$(6)
30 FOR I=1 TO 6
40 READ X$(I)
50 FOR J= 2 TO 3
60 READ A(I,J)
70 NEXT J
80 NEXT I
90 FOR I= 1 TO 6
100 A(I,4)=_____
110 S=S+A(I,4)
120 NEXT I
130 T=S*.2
140 FOR I= 1 TO 5
150 FOR J= _____ TO 6
160 IF _____ THEN 200
170 SWAP X$(I),X$(J)
180 FOR K=2 TO 4:SWAP A(I,K),A(____):NEXT K
200 NEXT J
210 NEXT I
215 PRINT "品名","零售价","销售量","销售额"
220 FOR I= 1 TO _____
230 PRINT X$(I),
240 FOR K=2 TO 4:PRINT A(I,K),:NEXT K
```

```

245 PRINT
250 NEXT I
260 PRINT "第一季度总销售额:";S;"元 "
270 PRINT "上交税额:";____;"元"
280 DATA "理光FT-5560",68000,15,"理光FT-4085",32800,6
290 DATA "施乐1027",29850,7,"佳能NP-3825",49000,10
300 DATA "东芝BD-5511",24800,8,"优美3200",32000,11
310 END

```

九·从供选择的答案中选出正确答案，把相应的字母写在答卷纸的对应栏内。（每空1分，共10分）

1. A total computer system includes both hardware and \_\_\_\_.
2. Key elements in a computer system include input, processing, and \_\_\_\_ devices.
3. One frequently used tool is the program \_\_\_\_.
4. BASIC is now by far the most popular high-level \_\_\_\_ used in microcomputer system.
5. Software is a set of \_\_\_\_ that tell the computer what to do.
6. String \_\_\_\_ too complex.
7. WARNING: All \_\_\_\_ on the disk will be lost.
8. The \_\_\_\_ capacity of this machine is equal to 10 MB.
9. UNIX is an \_\_\_\_ system developed at Bell Laboratories.
10. Insert new diskette for drive B and \_\_\_\_ any key when ready.

供选择的答案：

- A. Strike    B. memory    C. data    D. software    E. language  
F. operating    G. formula    H. output    I. flowchart    J. instructions

（考题答卷见 48 页）

## 给 TANGO 软件增加汉字元件库

黄健 陈军

TANGO 是原理图设计、连线网络表生成、印刷线路板自动布线等 CAD 及光绘、钻孔 CAE 于一体的多功能软件。其特点是“方便、易学、实用、快速”，自动布线率在 95% 以上。但美中不足的是 TANGO 软件不能在中文状态下运行。笔者在充分研究了 TANGO 软件的元件库源文件（扩展名·SRC）后，编写了一个 BASIC 实用程序，该程序能将汉字图形库自动地转换成 TANGO 软件的元件库源文件，通过 TANGO 汇编后可直接供 TANGO 绘图时使用。从而使 TANGO 能编辑出各种带有汉字的图形，如程序框图等，附图是笔者在项目开发设计时通过 TANGO 绘制的面板图。

### 一、程序的使用方法

1、进入 BASIC 状态，装入程序并运行。（如果程序能编译的话，则使用起来更方便）。

屏幕提示：filename?

这时应输入欲建立的汉字元件库的文件名。输入文件名后按回车，程序便进行自动转换。程序运行结束

后便得到了元件库源文件。

2、运行 TANGO 的库文件汇编程序 COMPILE.EXE 对刚才建立的汉字元件库源文件进行汇编（注意：元件库源文件的扩展名必须是·SRC），汇编完后便得到汉字元件库文件（扩展名·LIB）。

### 二、程序说明

20 句中的 C1 为欲转换的汉字库的起始区位码，程序中 C1=0101 包含了汉字库中的符号图形，如果不要符号图形，仅转换汉字，可使 C1=1601。

1204 句中的 8794 是汉字库中了最后一个汉字的区位码。值得注意的是 TANGO 的汇编程序一次只能汇编 2000 个元件，读者可将一个汉字库分几次转换。

60~140 句为一子程序，作用是根据区位码获得汉字的点阵信息。

（因篇幅有限，程序清单不再登出，需要者请寄 5 元复印及邮挂费来本刊索取。）

# 一九九一年计算机初级程序员有奖竞赛试题答卷

总分：\_\_\_\_\_

姓名：\_\_\_\_\_ 性别：\_\_\_\_\_ 年龄：\_\_\_\_\_ 文化程度：\_\_\_\_\_ 邮政编码：\_\_\_\_\_

工作单位：\_\_\_\_\_ 通信地址：\_\_\_\_\_

注意事项：

1. 答案必须在十月四日寄出，以邮戳为准(通信地址：北京学院路 31 号中间软件行业协会考试指导中心。邮政编码 100083)。
2. 填写答案时，必须将试题号与对应的答案栏对准，绝勿弄错，若某试题有多个答案时，请顺序给出(共 150 分)。
3. 解答时字迹务必清楚，字迹不清楚时不能得分。
4. 是否愿意参加复赛：是\_\_\_\_，否\_\_\_\_。

题号	小题号与答案										得分			
一	1		2		3		4		5		12	40		
	6		7		8		9		10		5			
	11		12		13		14		15		8			
	16		17		18		19		20		10			
	21		22		23		24		25		5			
二	1				2				3			8	30	
	4				5				6			7		
	7				8				9			6		
	10				11				12			9		
	13									1				
三	1												15	
	2													
	3													
	4													
	5													
	6													
四												8		
五	1	①												20
		②												
		③												
		④												
		⑤												
		⑥												
		⑦												
		⑧												
	2											4		
3	①		②		③		④				8			
六	①											3	12	
	②			③			④			9				
七												6		
八	①											6	12	
	②	请另外附纸画出流程图										6		
九	1		2		3		4		5		5	10		
	6		7		8		9		10		5			