

E&C
1992
 ●一九九二年 ●总期第84期
3

電子

ISSN 1000-1077

與 電腦



• ELECTRONICS AND COMPUTERS •

让 PC 286 微机配用电视机的

—PAL 制式转换卡

功能

- PAL 卡仅适用于带有场频调整旋钮的黑白或彩色电视机
- 图象清晰度取决于电视机的分辨率(低于专用监视器)

安装

- PAL 卡必须与双频卡或 CGA 卡联用
- PAL 卡可插入主机任何一个扩展槽中
- PAL 卡的九针插头转插到双频卡或 CGA 卡的视频输出插座上。再用射频电缆将 PAL 卡的射频输出端和电视机天线输入端连接起来

调整

- 将电视机设定在三频道。开机后,如画面不同,可调整电视机同步使画面稳定
- 如果色彩不正,可调 PAL 卡上的有关微调阻容器件(具体方法在本刊 92 年第 4 期刊出)

邮购

- 北京地区读者,可到北京王府井大街 30 号“景康电脑科技服务部”选购

联系人: 韩家钢

电话: 554613 542082

京外读者,可汇款到“北京市朝阳区三源里邮局七号信柜”

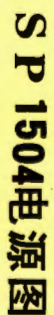
邮编: 100027 联系人: 杨炳林

售价: ¥180.00 元

另加邮挂费: ¥10.00 元

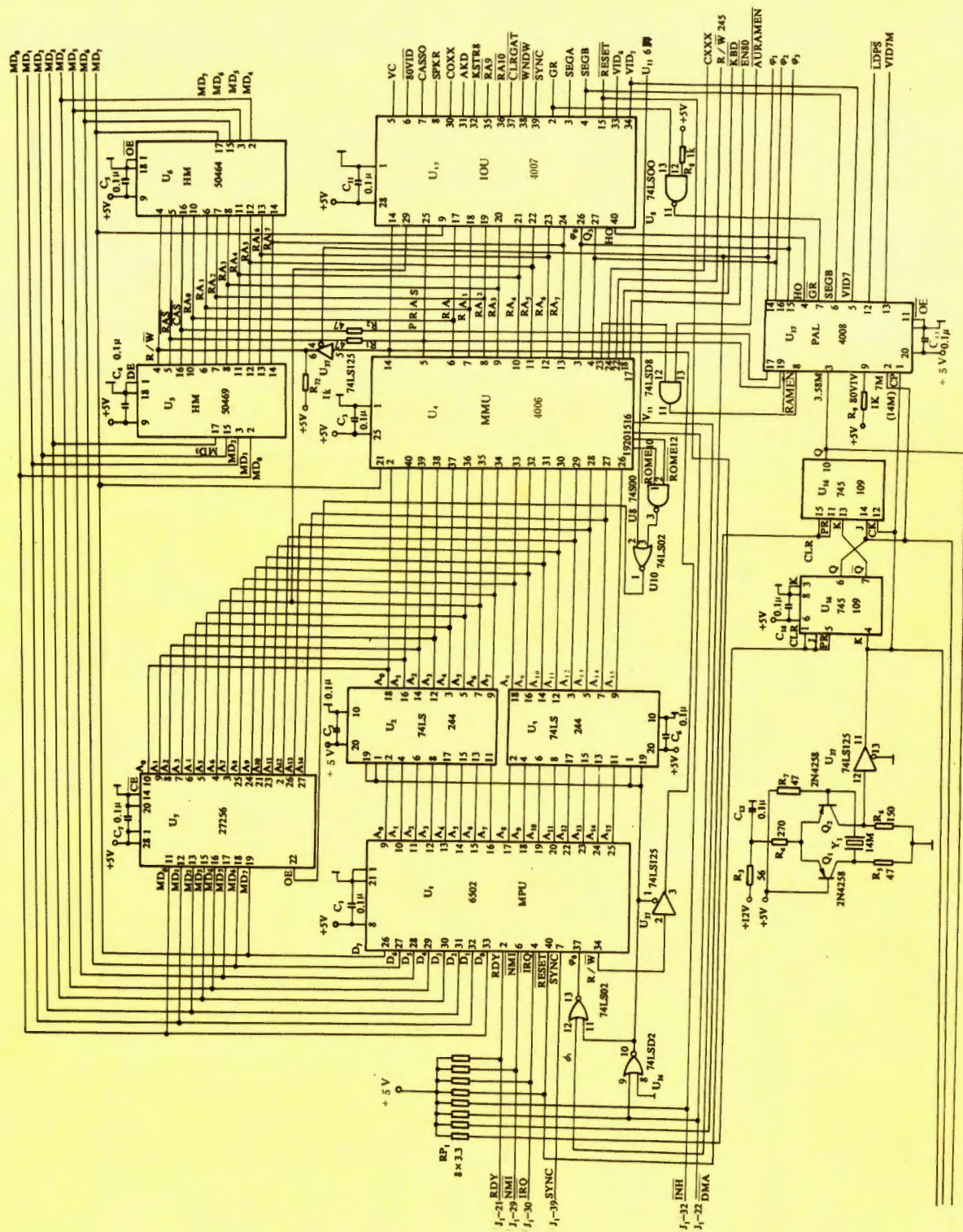
- PAL 卡由北京亚达电子公司制作,本刊监制。
- 图片为 CGA 演示软件的屏幕效果





SP 1504电源图

CEC—I 中华学习机主电路原理图





電子

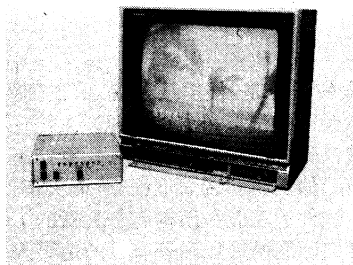
ISSN 1000-1077

電腦

379

一九九二年

总期第84期



• ELECTRONICS AND COMPUTERS •

電子與電腦

目 录

• 综述 •

软件汉化原理 郑茂松(2)

• PC 用户 •

编制能灵活打印二维报表的“报表生成系统”

..... 苏士俊(3)

在管理信息系统中实现计算模型的规则库方法。

..... 孙永芳 吴泉源(6)

dBASE II 修改文件属性的方法 许再由(8)

抗病毒软件

——Turbo Anti-Virus V6.80A 简单 唐银红(8)

新颖的年历打印通用程序 何贤敏(10)

• 学习机之友 •

CEC-I 汉字系统子程序的应用 傅叔平(11)

CATALOG 命令的改进 陈治浩(14)

中华学习机查找汉字区位码 胡瑞辉(14)

监控程序的浮动 赵 旭(15)

中华学习机特殊使用技巧 王 冈(15)

有趣的对比 丁志伟(16)

中华学习机汉化引导程序 张福森(17)

如何在苹果机持续运行一程序 张 骏(18)

ProDOS 磁盘操作系统入门(续) 廖 凯(19)

汽车大赛 马宇昊(21)

• 语言讲座 •

6502 机器语言程序设计

第三章 6502 MPU 的寻址方式 朱国江(22)

• 初级程序员级软件水平考试辅导 •

第三章 数据结构 宋丹颖(26)

• 学用单片机 •

单片机最小应用系统与液晶显示器(LCD)的接口

..... 张培仁(29)

• 学装微电脑 •

电源 ON、OFF 自动运转装置 易齐干(32)

• 电脑巧开发 •

微机视频接口器的研制 韦江维 覃龙生(37)

XMF 学习机(单驱扩展箱)改装为 XMF 及 PC 两用机

..... 杨青海(38)

• 电脑游戏机 •

第三章 F BASIC 的画面控制语句 于 春(39)

• 维修经验谈 •

NP125 型复印机中一个易被忽视的故障

..... 许 鹰(42)

PC-88 键盘分析与故障维修 王耀亭(42)

APPLE II 电源常见故障维修方法 邓满园(44)

IBM PC/XT 机的软磁盘驱动器磁头校准程序

..... 范思尧(45)

• 读者联谊 •

普及型 PC 个人用户软件交流联谊活动问题解答(三)

..... 王路敬(46)

封一:封面说明见 21 页

封二:SP1504 电源图

封三:CEC-I 中华学习机主控电路原理图

封四:PAL 制式转换卡

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

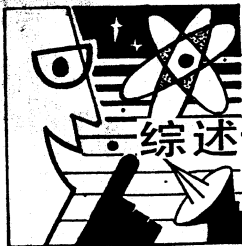
订购处:全国各地邮电局

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



软件汉化原理

中国科学院计算技术研究所(100080) 郑茂松

随着计算机在中国的推广使用,对中文软件的需求日益增多。鉴于软件汉化是中文软件开发的主要途径,提高软件汉化工作的效率和质量迫在眉睫,软件汉化研究势在必行。

中文软件的开发有两条途径:一条途径是重新设计和开发完全中文化的中文软件。这种中文软件的开发方法同世界上迅速发展的软件技术不相适应,不能充分利用国际上通用的丰富的软件资源,其应用也有很大的局限性。因此,这条途径看来很难走通。另一条途径是对西文软件实行汉化,这是目前广泛采用的一种切实可行的中文软件开发方法。

一、软件汉化的原理

软件汉化实质上是将被汉化的西文软件看成是一个缺少汉字处理功能的错误程序。软件汉化就是要测试和调试这个程序,直至它具有汉字处理功能为止。

根据测试和调试的原理和方法,我们把软件汉化的过程分为三个阶段:发现“错误”,查找“错误”、纠正“错误”。这里的“错误”是指“先天不足”、“水土不服”、“言语不通”三方面问题。前一个阶段是测试程序的过程,后两个阶段是调试程序的过程。通过这三个阶段的反复使用,反复迭代,反复试验,来完成软件汉化的全过程。

1. 发现“错误”

发现“错误”阶段的任务是:根据西文软件的功能和用户的要求,对被汉化软件拟定汉字处理功能描述,并据此设计测试用例,通过运行软件,检验测试用例是否符合预先拟定的汉字处理功能描述。若符合,则说明被汉化软件本来就具有这种汉字处理功能,在这一点上无需汉化;否则,说明被汉化软件不具有这种汉字处理功能,因而把不符合汉字处理功能的测试用例用于软件汉化的全过程。至于哪些测试用例符合预先拟定的汉字处理功能描述,取决于软件汉化的层次以及被汉化软件本身。

对于软件汉化这种特殊的软件测试,它所测试的软件一般均是目标代码。软件汉化人员往往只有软件的使用说明书,很少具备软件的技术说明书,面对可读性极差的目标代码,很难了解程序的内部结构和逻辑。因此,软件汉化设计测试用例一般无法采用白箱法,而只能采用黑箱法,而且主要采用黑箱法的等价分类法和边界值分析法。

在发现“错误”过程中,无论是拟定汉字处理功能描述,还是设计测试用例,或是运行软件试用各种汉字处理功能,都需要充分了解被汉化的西文软件的功能及其所处的环境,尤其是与增加汉字处理功能有关的原有功能。

2. 查找“错误”

查找“错误”阶段的任务是:根据从发现“错误”阶段得到的出错症状诊断出错位置和出错原因。

软件汉化把对被汉化软件执行路径的分析看作是一个由“黑箱”到“灰箱”的测试和调试过程。所谓灰箱,是指通过分析对整个软件的功能和结构有个大概的了解。这种方法强调针对问题的蛛丝马迹有目的地寻找和定位与测试用例等价类相对应的目标代码段。这种方法既避免了繁重的全局分析工作,又可有效地确定出错位置。查找“错误”的具体方法如下:

(1) 猜测试探查找法

根据从发现“错误”阶段得到的迹象和线索着手,提出对出错原因的种种猜想,列出发生错误的种种可能原因,并用测试用例去试探可能会引起这些原因的代码,通过调试,逐个证实或排除有关出错原因的猜想,最后把出错范围缩小到最低限度,再经过认真分析和试验,最终找到程序出错位置。

猜测试探查找法没有确定的步骤,在很大程度上是凭经验和直觉来猜测试探查找出错原因和试探程序出错位置。这些经验和直觉取决于软件汉化人员对被汉化软件的了解程度、是否熟悉编译原理、程序设计经验多少、推理和归纳能力强弱等。尽管它似乎是一种“瞎凑”、“蛮干”和“凭运气”的方法,但对于具有丰富的程序设计经验和软件知识的软件汉化人员来说,却是一种常用的行之有效的方法。

(2) 逻辑分析查找法

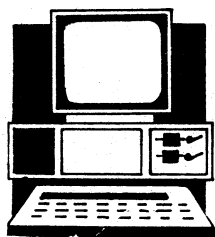
逻辑分析查找法从分析程序的内部逻辑着手查找“错误”。

对于全局逻辑分析,要分析被汉化软件的文件结构及内存映像结构,分析各个文件及其所含子程序的功能和相互调用关系。通过全局逻辑分析,可大致了解整个软件的总体逻辑结构,为进一步进行局部逻辑分析奠定基础。

当我们通过全局逻辑分析确定或基本上确定错误所在文件或子程序时,就可以对其中的各个疑点所在局部程序逐一进行局部逻辑分析,进一步确定出错原因和出错位置。局部逻辑分析又可分为静态分析和动态分析两种。静态分析就是把与错误有关的局部程序打印出来,仔细分析。动态分析就是通过调试工具分析程序的执行情况。静态分析可以帮助我们掌握局部程序的静态结构,动态分析可以帮助我们了解局部程序的动态执行过程。在局部逻辑分析过程中,常常把静态分析和动态分析结合起来使用。

(3) 启发式查找法

(下转第5页)



PC 用户

编制能灵活打印二维报表的“报表生成系统”

北京计算机学院(100044) 苏士俊

在一个具体的应用系统中,会有许多数据库文件,每个数据库又含多个字段,所以在打印报表时,不可能把全部字段都打印出来。常规的报表都有固定的格式和确定好的数据字段,但它们有时满足不了实际工作的需要。比如,经常遇到的随机查询,查询后要把满足条件的记录打印出来,但要求打印的数据项却常常与常规报表不同,或者只需打印一个仅含几个字段的简表。对于这样的系统,可以按下述方法为它设计一个相应的“报表生成子系统”,用来支持该种报表的表头和内容的打印。

由于数据库中的字段名常常用西文或汉语拼音,所以把数据库中的全部字段名和对应的汉字都放于一个“字段名数据库”内,查询记录的指针定位后,在某工作区打开该库,将字段名以菜单形式用汉字显示,请用户任选其中的某些字段。这时再用字符串相加的运算,将相应的中文字段名临时拼凑出一个汉字表头(表头=表头+中文字段名)。

假如用户选择了 NAME SEX, AGE..., 那么用 LISTNAME, SEX, AGE... 命令可以显示这些字段的值,但程序中难以预先写出这个随机的、含逗号的字段名表,普通用户也常常不会使用单条命令对数据库进行操作。为显示选中的字段内容,可使用下面 2 条命令:

```
STRING=NAME+" "+SEX+" "+STR(AGE,2,0)
? &STRING
```

如果把“? &STRING”语句放于 DO WHILE 循环中且移动指针,就可以代替 LIST 命令完成打印功能:

```
SELECT <区号>
USE <数据库文件>
LOCATE ALL FOR <条件>
DO WHILE .NOT. EOF()
? &STRING
CONTINUE
ENDDO
```

问题的关键是如何获得这个具有随机性质的 STRING,现简述如下:

我们从这个内存变量 STRING 后面的表达式结构可以得到启示,它与中文表头的结构相似,只是字段名是西文的,所以在拼接中文表头时,从存有中、西文字段名的“字段名数据库”中同时取出相应的西文字段名,也设法拼接在一起。

STRING 拼接时,适当地加入某些报表表格符号,然后把它放于语句(? &STRING)中,即可将报表符号连同字段的值形成一个完整的数据行,供打印使用。不断地移动满足条件的记录指针,不断地输出拼接好的

记录,就可以完成整个二维报表的灵活打印功能。

若字符串拼接时再增加串长的测试和判断,还可保证报表内容的总长不超过纸宽或屏幕宽度。例如,有一职工档案库(DA1.DBF),其库结构和数据如下:

.LIST STRUCTURE

库文件结构——文件名:C:DA1.dbf

库文件中的记录数量: 3

库文件的最后更新日期:01/01/80

| 字段 | 字段名 | 类型 | 宽度 | 小数 |
|------------|-----|-----|----|----|
| 1 | XM | 字符型 | 6 | |
| 2 | XB | 字符型 | 4 | |
| 3 | MZ | 字符型 | 4 | |
| 4 | ZC | 字符型 | 6 | |
| * * 总计 * * | | | | 21 |

.LIST

| 记录号: | XM | XB | MZ | ZC |
|------|-----|----|----|----|
| 1 | 李 森 | 男 | 汉 | 讲师 |
| 2 | 张丽萍 | 女 | 回 | 助教 |
| 3 | 王 立 | 男 | 汉 | 教授 |

注:为简化示范程序,该库中暂未设工资(GZ)和工作日期(GZRQ)等非字符型字段。

为编制“报表生成子系统”而设立的“字段名提示库”(ZD.TSK.DBF),其库结构和数据如下:

.LIST STRUCTURE

库文件结构——文件名:C:ZD-TSK.dbf

库文件中的记录数量: 4

库文件的最后更新日期:01/01/80

| 字段 | 字段名 | 类型 | 宽度 | 小数 |
|------------|-------|-----|----|----|
| 1 | 西文字段名 | 字符型 | 6 | |
| 2 | 汉字字段名 | 字符型 | 6 | |
| 3 | 字段长度 | 数值型 | 2 | |
| * * 总计 * * | | | | 15 |

.LIST

| 记录号: | 西文字段名 | 汉字字段名 | 字段长度 |
|------|-------|-------|------|
| 1 | XM | 姓 名 | 6 |
| 2 | XB | 性 别 | 4 |
| 2 | MZ | 民 族 | 4 |
| 3 | ZC | 职 称 | 6 |

下面是完成具有灵活性打印功能的简化程序:

* 字段名提示部分:

```
SET TALK OFF
USE ZD-TSK
A=0
```

```

M=0
CLEAR
@6,18 SAY " * * * * 相关项提示 * * * * (括
号内的数字为字段宽度)"
DO WHILE .NOT. EOF()
    M=M+1
    @7+A,20 SAY STR(M,2,0)+"：" +TRIM(汉
    字字段名)
    @7+A,32 SAY "(" +STR(字段长度,2,0)+
    ")"
    SKIP
    A=A+1
ENDDO
* 以下拼汉字表头和 STRING:
总长=80
DO WHILE .T.
    汉字表头="|"
    STRING="'|'"
    线="-"
    剩余宽度=总长-2
    @ 0,2 SAY "表头:"
    DO WHILE .T.
        CCD=0
        @5,20 SAY SPACE(60)
        @4,0 SAY SPACE(80)
        @4,5 SAY "用户自定义显示项",
        @4,21 SAY"请选择显示项的编码:" GET CCD
        PICTURE "9" RANGE 1,M
    READ
    @ 4,50 SAY "稍等....."
    GO CCD
    OLD=剩余宽度
    剩余宽度=剩余宽度-字段长度-1
    IF 剩余宽度>=0
        汉字表头=汉字表头+TRIM(汉字字段名)+
        "|"
        STRING=STRING+"+" +西文字段名+"+"
        +"'|'"
        A=1
        DO WHILE A<=字段长度+1
            线=线+"-"
            A=A+1
        ENDDO
        @ 4,50 SAY SPACE(15)
        @ 1,0 SAY 线
        @ 2,0 SAY 汉字表头
        @ 3,0 SAY 线
        CON="Y"
        @ 5,0 SAY SPACE(80)
        @ 5,5 SAY "屏幕剩余宽度:" +STR(剩余宽
        度,3,0)
        @ 5,20 SAY "继续输入吗(Y/N)?" GET
        CON PICTURE "!"
    READ
    @ 5,20 SAY SPACE(60)

```

```

IF CON="N"
    RK="Y"
    @4,0 SAY SPACE(80)
    @4,5 SAY " 认可吗(Y/N)?"GET
    RK PICTURE"!"
    READ
    IF RK="Y"
        EXIT
    ELSE
        @5,0 SAY SPACE(80)
        @5,5 SAY" 不认可,请重新编辑表
        头内容,按任一键..."
        WAIT " "
        @ 5,0 SAY SPACE(80)
        EXIT
    ENDIF
ENDIF
ELSE
    ?"屏幕超宽!此时的处理已省略."
    RK="Y"
    EXIT
ENDIF
ENDDO
* 以下程序显示表头和报表内容:
IF (RK="Y").AND. (LEN(STRING)>1)
    USE
    SELECT 2
    USE DA1
    ?线
    ?汉字表头
    ?线
    DO WHILE .NOT. EOF()
        ?&STRING
        ?线
        SKIP
    ENDDO
    WAIT
    USE
    RETURN
ELSE
    @1,0 SAY SPACE(80)
    @2,0 SAY SPACE(80)
    @3,0 SAY SPACE(80)
    LOOP
ENDIF
ENDDO
RETURN

```

注:①下面是操作时的屏幕显示情况:

表头:

| | | |
|----|----|----|
| 姓名 | 职称 | 民族 |
|----|----|----|

用户自定义显示项,请选择显示项的编码:

屏幕剩余宽:59

* * * * 相关项提示 * * * * (括号内的数字为字段宽度)

1: 姓 名 (6)

2: 性 别 (4)
3: 民 族 (4)
4: 职 称 (6)

②拼接表头时,若增加表头的超宽处理,可直接用下面的程序段代替 RK="Y"和 EXIT 2条语句:

```

    剩余宽度=OLD
    CN=0
    @ 4,0 SAY SPACE(80)
    @ 4,0 SAY "超出80列!你可以:"
    @ 5,0SAY " 1.反复改最后一栏 2.重新编辑表头
0.不再增加
    @ 5,50 SAY "请选择(0-2):" GET CN PIC-
    TURE "9" RANGE 0,2
    READ
    CN=STR(CN,1,0)
    DO CASE
        CASE CN="0"
            RK="Y"
            EXIT
        CASE CN="1"
            @ 5,0 SAY SPACE(80)
            @ 5,5 SAY "屏幕剩余宽:"+STR(剩余宽度,
            3,0)

```

```

CASE CN="2"
    RK="N"
    @ 5,0 SAY SPACE(80)
EXIT

```

ENDCASE

③本程序只处理了字符型字段的拼接。由于 STRING 中只准拼接字符型的内容,为处理其它类型字段,可在 ZD.TSK.DBF 中改设“变形字段名”字段,其内直接存放 XM、XB、ZC、STR(GZ, 6, 2), DT0C(GZRQ)…。这时:

STRING=STRING+" "+TRIM(变形字段名)+" "+
+"|'".

④下面是运行简化程序形成的 STRING:

```

. ?STRING
    屏幕显示: '|'+XM + '|'+ZC + '|'+MZ
+ '|'+

```

⑤相应的报表内容如下:

| 姓 名 | 职 称 | 民 族 |
|-----|-----|-----|
| 李 森 | 讲 师 | 汉 |
| 张丽萍 | 助 教 | 回 |
| 王 立 | 教 授 | 汉 |

(上接第2页)

有时,我们可以借助一些启发信息来分析出错原因和确定出错位置,这样做有时确能找到解决问题的捷径。启发信息来源很多,例如,通过观察工作环境,查看历史、暂时去掉程序的某些部分等手段,可以从中发现一些启发信息。启发式查找法的关键在于善于发现和善于利用启发信息。

(4)跟踪查找法

在无法采用上述方法找到出错位置的情况下,可采用跟踪程序的方法来查找“错误”。

跟踪查找法一般借助于调试工具采用线路跟踪方式,把单步方式与断点方式结合起来使用,即用断点执行方式以较快的速度通过确认无错的程序段,而出现错误迹象后再采用单步方式逐步跟踪,直至找到出错位置为止。当执行程序出错时,亦可采用回溯跟踪方式,从发现错误征兆的地方开始往回追溯程序代码,把出错前的一段程序的有关线路和赋值情况输出,以供软件汉化人员分析出错原因。

3. 纠正“错误”

纠正“错误”阶段的任务:根据出错原因和出错位置,修改被汉化的程序,并用测试用例反复执行程序,检验是否符合汉字处理功能描述,直到它具有汉字处理功能并且排除因修改程序而引入的新错误为止。

在修改程序时,要斟酌修改后的程序代码,最好修改后的程序代码长度比修改前的程序代码短或一样长,以便覆盖;否则,就需要采用“打补丁”的办法把多余的部分接到空白位置,有时还需要增加文件的驻留长度。

软件汉化常常会产生副作用,这是难免的。产生副

作用的原因是多种多样的,往往是由于考虑不周或未完全弄清被修改程序的含义而引起的。例如,在被汉化软件中,一处的数据可能被多处程序所引用,因而当修改了这一处数据后,虽使一处程序纠正了错误,但却影响到另一处程序。在修改程序的过程中,应尽量避免产生副作用;当出现副作用时,要设法消除副作用。

软件汉化对程序的修改是否正确,是否存在副作用,除了回归测试外,在汉化软件投入使用后,还需要在使用过程中逐渐发现。要及时反馈用户发现的问题,不断维护和完善汉化软件。

二、软件汉化的工具

由于软件汉化是依照测试和调试程序的原理和方法进行的,因此需要借用测试工具和调试工具当作软件汉化工具。目前,测试工具尚很少用于软件汉化,而调试工具则是软件汉化的主要借用工具。

为了提高软件汉化的效率和质量,有必要总结软件汉化的规律,研制适合软件汉化的专用工具。软件汉化的专用工具一方面要汲取软件汉化借用工具的优点,另一方面要力求实现软件汉化的自动化。由于软件汉化是一个复杂的推理、分析、判断、归纳和思维过程,因而软件汉化的自动化很难实现。目前只能研制一些软件汉化辅助工具,例如,提示信息汉化辅助工具。

提示信息汉化辅助工具大多采用并行的提示信息汉化方法,自动地寻找西文提示信息,通过全屏编辑修改提示信息,使提示信息汉化工作并行化、自动化、简单化,从而提高软件汉化的生产力和可靠性。

(编者按:郑茂松同志编著的“中文软件和软件汉化”一书,即将由电子工业出版社出版发行;本文作了适当删节)。

在管理信息系统中 实现计算模型的规则库方法

孙永芳 吴泉源

高级管理信息系统的一个重要特征是系统含有大量的辅助决策模型,并且,模型的求解往往采用先进的人工智能技术和软件重用技术实现,因而系统一般都具有鲜明的智能化和工具化的特征。本文以我们在 IBM PC 机上实现的一个机电设备高级管理信息系统为背景,用基于规则的计算工具,描述这种智能管理信息系统的设计方法和实现流程。

一、嵌套型分段函数计算模型

在我们的管理信息系统中,为有效地评价和管理各种大型机电设备的性能完好状况,经高度抽象,最终以一种通用的嵌套型分段函数建立了相应的辅助决策模型,它的一般形式是:

$$Y = \begin{cases} f_1(x_{11}, \dots, x_{1m_1}) & \text{条件1} \\ f_2(x_{21}, \dots, x_{2m_2}) & \text{条件2} \\ \dots\dots\dots \\ f_n(x_{n1}, \dots, x_{nm_n}) & \text{条件n} \\ f_{n+1}(x_{n+11}, \dots, x_{n+1m_{n+1}}) & \text{其它} \end{cases}$$

这里,变量 x_{ij} ($i=1, 2, \dots, n+1; j=1, 2, \dots, m_i$) 允许是另一分段函数。若某个 f_i 不含变量,或者说某个 $m_i=0$,这个 f_i 即表示一常量。

这类模型的含义是:若条件1成立,则用 f_1 计算 Y 的值;否则,若条件2成立,则用 f_2 计算 Y 的值;……;否则,若条件 n 成立,则用 f_n 计算 Y 的值;否则,用 f_{n+1} 计算 Y 的值。

按照传统的实现技术,这类函数不难直接编写到程序中。但是,这样做不利于模型的修改和扩充。我们知道,管理信息系统的生命力就在于它的易修改性和易扩充性。例如,在我们的系统中,许多设备的性能函数属经验函数,许多计算公式的结构和有关数据需在实际应用中不断完善;在该系统研制期间,不少设备的计算公式甚至还在讨论之中。以后,对于需要加入该系统管理的新设备,其性能函数更要经过多方研究平衡才能确定,并扩充到系统中。凡此种种,那种把大量计算公式编死在程序中的传统做法,既不可取,也行不通。为此,我们运用知识工程的方法,把计算模型依规则形式组织到规则库中,并编制一个规则解释程序,由该程序对规则库中的所有规则进行统一的解释与计算。这样,模型与程序是分离的,对模型的增删改就是对规则库的增删改,正象对数据库的增删改一样,不用改动和新编任何程序。事实上,数据库系统的基本功能

就在于用户可以随意进行系统所规定的各种库操作,而不会对系统程序的正常运行造成丝毫影响。

二、计算规则和计算规则库

所谓计算规则就是表示嵌套分段函数的规则,它的一般形式是:

(规则名, 条件1, 计算公式1,
条件2, 计算公式2,
……
条件 n , 计算公式 n ,
TRUE, 计算公式 $n+1$)

其中,规则名对应于函数名 Y ,条件部分为逻辑表达式,计算公式部分为算术表达式。“TRUE”表示恒真,当条件1至 n 均不成立时起作用,相当于分段函数的最后一个条件“其它”。

把各个分段函数对应的计算规则集中在一起,可以组成一个或多个数据库文件。每个这样的数据库文件就是本系统的一个计算规则库,它们的每一个记录由五个字段组成如下:

| 规则名 | 条件 | 计算公式 | 变量说明 | 注释 |
|-------|-------|-------|-------|-------|
| C(10) | C(50) | C(50) | C(50) | C(40) |

每一记录共200个字符,对应于一个分段函数的一个分支,显然,整个分段函数的计算规则应当由 $n+1$ 个记录表示,且它们的规则名均对应于同一个函数名。“变量说明”用以说明“条件”和“计算公式”中出现的变量;若说明的形式为

[变量名, 数据库名, 检索条件]

则相应变量的值由检索指定数据库文件按指定检索条件获得;若说明的形式为

{变量名, 规则库名}

则相应变量的值由调用指定规则库的指定规则获得,这个规则的规则名就是说明中的变量名。

每一个记录的注释部分用以增强规则的可读性,可以填写计算规则的各种注释,与规则的计算无关。

三、规则解释程序

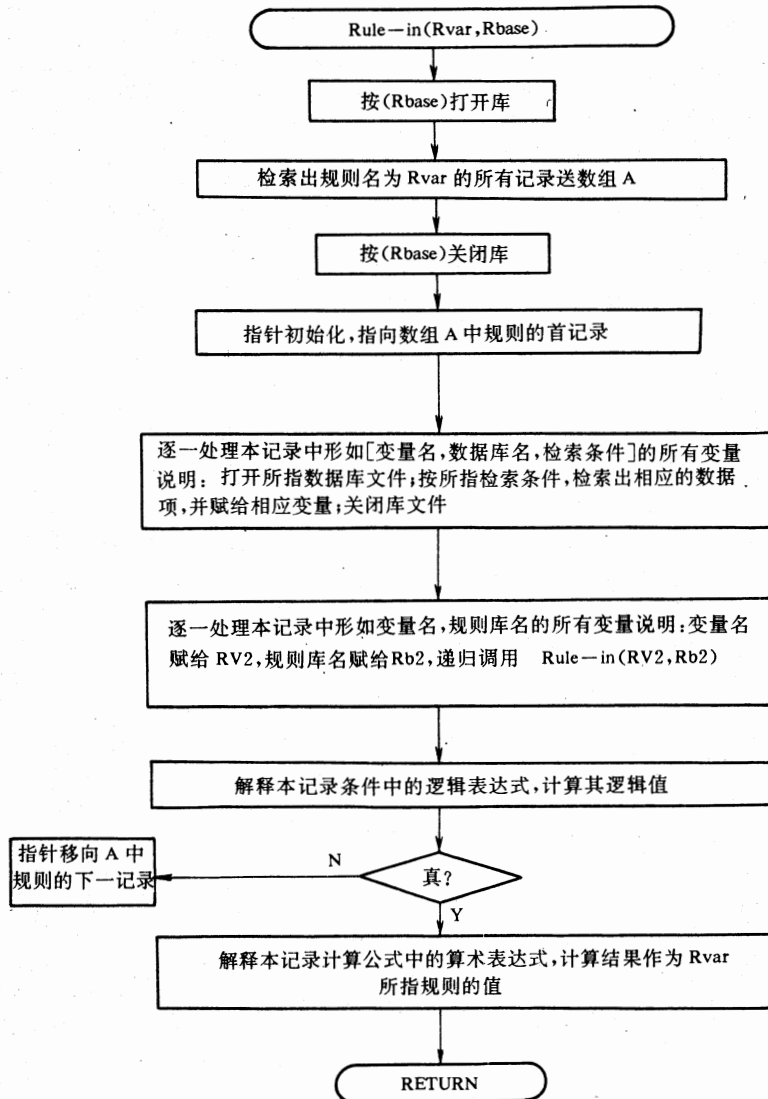
孤立地看,规则库与数据库没有什么区别,只有规则库加上规则解释程序才能完成动态的计算任务。本系统的规则解释程序的任务是,从指定的规则库中对指定的规则进行计算。其主要过程是 Rule.in (Rvar,

Rbase),调用时,变量 Rvar 已赋以指定的规则名,Rbase 已赋以指定的规则库名,要求计算结果回送作为变量 Rvar 所指规则的值。

规则解释过程 Rule.in 的程序流程详见流程图。注意,由于分段函数允许嵌套,该过程成了一个递归过程,即内部出现了递归调用。如果你的机器配有 Oracle

数据库系统,则该过程容易用高级语言 Pascal 嵌入 Oracle 实现。

在计算规则库中,如果把变量说明划分为两部分,每一部分分别对应于规则的条件部分和计算公式部分的变量说明,那么 Rule.in 的程序流程稍加改进可提高它的执行效率。



规则解释过程 Rule.in 的流程图

dBASE III 修改文件属性的方法

湖北江汉油田物探处(433100) 许再由

在 dBASE III 下,倘若把文件改为隐含属性,就不能被 DOS 系统的 DIR、COPY、ERASE 等命令显示、拷贝、删除,从而起到加密文件的作用。本文介绍用 dBASE III 生成小汇编程序的方法,通过 DOS 软中断 INT 21H 的 43H 功能调用,可以修改文件属性。dBASE III 生成汇编程序的关键是如何形成汇编语句的 16 进制机内码。为此,我们先把汇编语句的机内码换算成十进制数,利用 CHR(n) 函数将十进制数变成字符,存入数据库,再用 COPY TO(文件名)DELI WITH BLAN 命令把字符拷贝到后缀为 TXT 的文件中,该文件里的机内码就是所要求的汇编语句,最后换名后缀是 .COM 的文件,用 RUN 命令运行这个文件即可修改文件属性。但由于 dBASE III 不能产生 16 进制 00H,而汇编程序里的 43H 功能调用,规定 00H 为文件名的结束标志。所以我们建一个仅含 00H 机内码的文件如下:

```
C)DEBUG
-R CX
CX 0000
:1
-N BB.COM
-W
-Q
```

本文产生的汇编程序名为 JM.COM,用 COPY JM.COM+BB.COM 命令就能在 JM.COM 中补充一个 00H。源程序用 dBASE III 编写,也适用于 FOXBASE+ 系统。该程序还可修改子目录的属性,具有隐含属性的目录,不能被 DIR、TREE 命令发现。

```
程序名:JM.PRG
CLEA
SET TALK OFF
SET SAFE OFF
P=SPAC(12)
P1=SPAC(1)
@ 2,25 SAY "输入文件名:" GET P
@ 3,25 SAY "文件属性(普通0,只读1,隐含2,系统4)"GET
```

```
P1
READ
USE JM1
ZAP
APPE BLAN
P1=VAL(P1)+32
K=CHR(184)+CHR(01)+CHR(67)+CHR(177)+CHR
(P1)+CHR(186)+CHR(16)
K1=CHR(01)+CHR(205)+CHR(33)+CHR(205)+SPAC
(5)+TRIM(P)
REPL A1 WITH K+K1+CHR(26)
COPY TO JM DELI WITH BLAN
COPY FILE JM. TXT TO JM.COM
RUN COPY JM.COM+BB.COM
RUN JM.COM
RETU
```

数据库结构—数据库 :C:JM1.dbf
数据库中的数据记录个数: 1
数据库的最后更新日期 :12/03/90

| 字段 | 字段名 | 类型 | 宽度 | 小数 |
|------------|-----|-----|----|----|
| 1 | A1 | 字符型 | 28 | |
| * * 总计 * * | | | 29 | |

由 JM.PRG 产生的 JM.COM 汇编程序如下:

```
C)DEBUG JM.COM
9CAF:0100 B80143 MOV AX,4301
9CAF:0103 B120 MOV CL,20
9CAF:0105 BA1001 MOV DX,0110
9CAF:0108 CD21 INT 21
9CAF:010A CD20 INT 20
9CAF:010C 2020 AND [BX+SI],AH
9CAF:010E 2020 AND [BX+SI],AH
9CAF:0110 7177 JNO 0189
9CAF:0112 2E CS:
9CAF:0113 7072 JO 0187
9CAF:0115 67 DB 67
9CAF:0116 001A ADD [BP+SI],BL
```

抗病毒软件 Turbo Anti—Virus v6.80A 简介

湖北远安县财政局 唐银红

Turbo Anti—Virus v6.80A 是 CARMEL 软件公司于一九九〇年六月推出的抗病毒软件包,由于其诊断的病毒种类多达 154 种,诊断和消毒能力强,采用下拉式菜单,用户接口界面良好,易于使用,越用越觉

得方便。特别值得推崇的是,当 Turbo Anti—Virus v6.80A 驻留内存后,它就象一名忠诚的卫士,时刻监视着机器的工作,对各种操作,首先进行病毒检查,如发现病毒,立刻用文字和声音报警,提醒操作员及时消除

病毒。但当它驻留内存后,对机器的运行速度稍有影响。

一、Turbo Anti-Virus v6.80A 的构成

Turbo Anti-Virus v6.80A 全部存放在一张软盘上,共8个文件,因其主文件名称为 TNTVIRUS.EXE,故整个抗病毒软件包常被称为 TNT,在检测和清除病毒的过程中,其全部参数均可用下拉式菜单指定,也可用命令项中的高亮度字母来选择。BOOTS SAFE.EXE 用于检查命令行中参数指定的驱动器的引导记录,如果命令行上无参数,则检查缺省的驱动器。TSAFE.COM 使病毒检查程序驻留内存,用于监测计算机的工作,并可随时用热键激活。INSTALL.EXE 向指定盘和路径上安装整个抗病毒软件包,并在指定盘根目录上产生自启动文件 AUTOEXEC.BAT,使机器在启动时首先检查内存和分区表,并将病毒监测程序驻留内存。README.DOC 文件是软件包的使用说明,包括上述四个文件的使用方法和参数,本软件包能检查的154种病毒的名称和特点等。

二、Turbo Anti Virus v6.80A 的使用

1. TNTVIRUS.EXE 检查和清除病毒软件,如果要检查和清除病毒,只需键入 TNTVIRUS<ENTER>。

执行该程序后,首先检查自身是否被病毒感染,如果检查到 TNTVIRUS.EXE 本身已被病毒感染,则向用户提出警告,告诫用户 TNTVIRUS 本身具有危险性。如程序本身完好,则对内存进行检查,通过后出现主控窗口。

主控窗口可以分成四部分:最上面一行是主命令,可用 ALT 键使光标移到主控窗口的命令行上,用光标键移动光标至所需的命令上,回车键使命令生效。中间又分为两部分,左边显示当前或指定盘上的树型目录结构,检查时则显示被检查子目录中的被检查的文件名称。右边显示有关系统信息,诸如工作盘的类型,子目录的个数,当前目录中还没有检查的文件个数,系统时间、日期,发现的病毒的名称,系统所提供的消息等。最下面的一行是对当前命令的说明,在任何时候,要想退出 TNTVIRUS,按 ALT-X 键返回 DOS。

2. BOOTS SAFE.EXE 检查引导区程序

BOOTS SAFE 检查指定盘的引导扇区和分区表,使用时对内存也进行检查。一旦发现引导扇区或分区表与映象图不同,则警告窗被打开,用声光报警,并显示以下信息:

Boot/Partition sector was modified
Restore Update Continue Stop

移动光标键可选择恢复(Restore)、更新映象图(Update)、继续(Continue)、终止(Stop)。

BOOTS SAFE 命令行上可带以下参数:它们是

D:指定检查的目标盘

M:对被检查盘的引导扇区产生新的映象图

R:从 A 盘读取映象图,并拷贝到指定盘

例如:BOOTS SAFE D:/M 的意义是:检查 D 盘的引导扇区和分区表,并将 D 盘的引导扇区映象图拷贝到

软盘中。

3. TSAFE.COM 驻留内存程序

本程序运行后,使保护程序驻留内存。它提供八个方面的保护功能,其控制窗口用热键 ALT-T 激活,用数字键设定是否保护。这八个功能是:软盘格式化保护、硬盘低级格式化保护、驻留保护、模拟写保护、检查所有文件、检查引导扇区、引导扇区保护、执行文件保护。TSAFE 命令行上的参数可改变热键,如要使 ALT-S 为热键,只需打入

TSAFE/AS

4. INSTALL.EXE 安装程序

当执行 INSTALL 程序后,系统自动将整个 Turbo Anti-Virus 软件包安装到指定盘上。当盘符被指定后,自动产生 TNTVIRUS 的子目录,并将主要文件拷贝到子目录中,同时生成 AUTOEXEC.BAT 文件,文件内容如下:

C:\TNTVIRUS\TSAFE

C:\TNTVIRUS\BOOTS SAFE C:

三、与 VIRUSCAN 2.0 V54 的比较

VIRUSCAN 2.0 V54 是美国 McAfee Associates 于九〇年元月注册发行的,因美国 AST 公司购买其版权并免费向其286用户提供,加上公安部的推荐,在我国拥有众多的用户,因此有必要作一比较。

1. Turbo Anti-Virus 采用下拉式菜单,整个运行过程中,检查的文件和有关运行参数都在主控窗口上显示,人机界面十分友好。

2. Turbo Anti-Virus 的运行参数可在使用过程中指定,参数(如盘符、路径、检查方式等)容易修改。而 Viruscan 的运行参数必须在命令行中给定,如果参数设置有错,则不能进入检查程序。

3. Turbo Anti-Virus 集病毒检查、消毒于一体,如果没有设置清除病毒的功能,程序在运行过程中一旦发现病毒,除用声光报警外,还询问用户是否消毒,而 Viruscan 中的 SCAN 程序只能检查是否染毒,消毒必须用其他软件,即使是新版 Viruscan v 61,查出病毒后,须另外调用 CLEAN-UP 清除文件病毒,如果病毒隐藏在分区表或引导扇区,则须调用 MDISK 程序消毒。

4. Turbo Anti-Virus 能检查154种病毒,而 Viruscan 仅能检查60余种。

5. Turbo Anti-Virus 能发现被检查盘中 BOOT/PARTITION(引导扇区)的任何改变,并能恢复到未修改的状态,也能根据用户的要求,将被检查盘的引导/分区扇区备份至软盘上。

6. 当 Turbo Anti-Virus 的 TSAFE 程序驻留内存后,能对系统资源进行保护,特别是当病毒企图对引导/分区扇区进行写(修改)操作时,它能产生一个假想的写保护,使病毒放弃传染的企图。

7. Turbo Anti-Virus 的消毒能力强,无论病毒是在内存,或是在文件及分区表内,均能有效地杀灭病毒,消毒后不须关机或热启动。

新颖的年历打印通用程序

浙江台州卫生防疫站(317000) 何贤敏

以往的年历打印程序,只能打印阳历,在很大程度上不能满足日常需要,本文提供的程序能同时打印阳历、阴历,打印格式合理,经过修改某些数据,又能起到通用的打印其他年份的作用,效果比较满意。

在程序设计过程中,采用对阳历、阴历分别计数、比较,阳历用普通数数字,阴历号数用国际汉字库中的半角数字,月号用罗马数字,阴、阳历同时打印的方法实现。

程序10~130为设置初值;140~200打印年历头;210~260及430~450是为保证上、下月衔接正确所作的数值计算及判断;270~410为打印年历正文。

若要进行其他年份的打印,只需对20、40、50、70、80、90、150行中的某些变量初值或常数作相应的修改即可。有关变量的含义已在程序中说明。

提供的程序在 CCBIO 2.13H、BASICA 3.30 下调试运行通过。

```
10 WIDTH "LPT1:",150;DIM N1(15),N2(16),N$(15)
20 DATA 31,"一 月",29,"二 月",31,"三 月",30,"四 月",31,"五 月",30,"六 月"
30 DATA 31,"七 月",31,"八 月",30,"九 月",31,"十 月",30,"十一月",31,"十二月"
35 REM 上二个语句行是阳历各月总天数及月名
40 DATA 30,30;REM 阴历上一年延续至阳历今年的各月总天数
50 DATA 29,30,30,29,29,30,29,29,30,29,30,30;REM 阴历今年1~12月各月总天数
60 DATA 30,30
70 KK=3; REM KK 为阳历今年第一天星期数(0~6)
80 J2=27; REM J2 为阴历在阳历今年第一天的号数
90 A=11; REM A 为阴历在阳历今年第一天的月份
100 R=12-A
110 FOR I=1 TO 12;READ N1(I),N$(I);NEXT
120 FOR I=1 TO 16;READ N2(I);NEXT
130 I=1;II=I+1;J=1;JJ2=J2;D$="日 一 二 三 四 五 六"
140 LPRINT "@P";
150 LPRINT TAB(17);"——一九九二——"
160 IF J > 12 THEN GOTO 470
170 LPRINT "@I"
180 LPRINT TAB(19);"~~~~~";N$(J);"~~~~~";
TAB(69);"~~~~~";N$(J+1);"~~~~~";
190 LPRINT "@A";
200 LPRINT TAB(5);D$;TAB(55);D$
210 K=KK;IF K > 6 THEN K=0
```

```
220 KK=(N1(J)+K)-7*INT((N1(J)+K)/7)
230 J2=JJ2;JJ2=N1(J)-N2(I)+JJ2
240 IF N2(I)-J2+1>=N1(J) THEN II=I ELSE IF N2(I)-J2+N2(I+1)+1>=N1(J) THEN II=I+1 ELSE II=I+2
250 M=1;IF N2(I)-J2+1=N1(J) OR N2(I)-J2+N2(I+1)+1=N1(J) THEN M=3
260 IF II>12 THEN II=II-12
270 IF J1<N1(J) THEN J1=J1+1 ELSE GOTO 340
280 IF J1>9 THEN P=6*K+4 ELSE P=6*K+5
290 IF J2<=N2(I) THEN P$=CHR$(172)+CHR$(160+J2);J2=J2+1;GOTO 320
300 II=I-R; IF II<=0 THEN II=II+12
310 P$=CHR$(162)+CHR$(240+II);J2=2;IF MM=3 THEN I=I+1;MM=MM-1
320 LPRINT TAB(P);J1;P$;
330 IF K>=6 THEN K=0 ELSE K=K+1;GOTO 270
340 IF JJ1<N1(J+1) THEN JJ1=JJ1+1 ELSE GOTO 410
350 IF JJ1>9 THEN PP=6*KK+54 ELSE PP=6*KK+55
360 IF JJ2<=N2(II) THEN PP$=CHR$(172)+CHR$(160+JJ2);JJ2=JJ2+1;GOTO 390
370 II=II-R; IF II<=0 THEN II=II+12
380 PP$=CHR$(162)+CHR$(240+II);JJ2=2;IF M=3 THEN II=II+1;M=M-1
390 LPRINT TAB(PP);JJ1;PP$;
400 IF KK>=6 THEN KK=0 ELSE KK=KK+1;GOTO 340
410 IF J1<N1(J) OR JJ1<N1(J+1) THEN GOTO 270
420 J1=0;JJ1=0;J=J+2
430 IF N2(II)-JJ2+1>=N1(J+1) THEN I=II ELSE IF N2(II)-JJ2+N2(II+1)+1>=N1(J+1) THEN I=II+1 ELSE I=II+2
440 MM=1; IF N2(II)-JJ2+1=N1(J+1) OR N2(II)-JJ2+N2(II+1)+1=N1(J+1) THEN MM=3
450 IF I>12 THEN I=I-12
460 GOTO 160
470 LPRINT;LPRINT;END
```

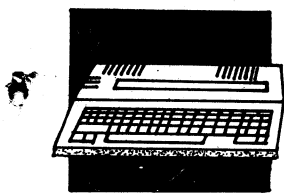
一九九二

| 日 | 一 | 二 | 三 | 四 | 五 | 六 | 日 | 一 | 二 | 三 | 四 | 五 | 六 |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 5 | 6 | 7 | 8 | 9 | 10 | 11 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 12 | 13 | 14 | 15 | 16 | 17 | 18 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 19 | 20 | 21 | 22 | 23 | 24 | 25 | 16 | 17 | 18 | 19 | 20 | 21 | 22 |
| 26 | 27 | 28 | 29 | 30 | 31 | | 23 | 24 | 25 | 26 | 27 | 28 | 29 |

| 日 | 一 | 二 | 三 | 四 | 五 | 六 | 日 | 一 | 二 | 三 | 四 | 五 | 六 |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| 15 | 16 | 17 | 18 | 19 | 20 | 21 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 22 | 23 | 24 | 25 | 26 | 27 | 28 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| 29 | 30 | 31 | | | | | 19 | 20 | 21 | 22 | 23 | 24 | 25 |

| 日 | 一 | 二 | 三 | 四 | 五 | 六 | 日 | 一 | 二 | 三 | 四 | 五 | 六 |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 3 | 4 | 5 | 6 | 7 | 8 | 9 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 16 | 17 | 18 | 19 | 20 | 21 | 22 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 23 | 24 | 25 | 26 | 27 | 28 | 29 | 21 | 22 | 23 | 24 | 25 | 26 | 27 |
| 30 | 31 | | | | | | 28 | 29 | 30 | 31 | | | |

(由于地方有限年历适当删去)



学习机之友

CEC—I 汉字系统子程序的应用

成都市成人教育学院(610017) 傅叔平

中华学习机 CEC—I 上固化的汉字系统是一个丰富的资源,其中有很多有用的子程序。用户程序直接调用它们往往事半功倍,有时还可获得一般情况下难以得到的效果。直接使用系统子程序还可大大缩短程序长度,对提高编程效率,节约内存空间及增强程序的可读性等方面均有积极的意义。

汉字系统的管理程序主要分为接口部分(\$C300~\$C3FF)和主体部分(辅存2的\$EC00~\$FFFF)。对于接口部分的子程序,用户程序可用通常的方式直接调用,而对主体部分的子程序则应以如下方式调用:

JSR \$C3AB;进入辅存2

JSR (子程序入口)

JSR \$C3B9;回到主存2

下面介绍部分子程序应用的例子。

1. 代码转换

CEC—I 一共用了四种汉字内码:区位码、国标码、异形国标码和内码,常涉及到这些代码之间的转换。前三种代码之间有非常简单的数学关系,转换很容易。内码与这些代码之间的转换要复杂一些,转换时可直接调用下面的两个子程序:

\$C389——功能:将异形国标码转换为内码,但转换后最高位为1,入口、出口参数均在累加器A中。

\$C392——功能:将内码转换为国标码,入口、出口参数均在累加器A中。

程序1是将\$300单元中的区位码转换成内码。程序中前三行是先将区位码转换成异形国标码。

【程序1】

```
0310— AD 00 03      LDA      $ 0300
0313— 18            CLC
0314— 69 A0          ADC      # $ 0A0
0316— 20 89 C3       JSR      # C389
0319— 29 7F          AND      # $ 7F
031B— 8D 00 03       STA      $ 0300
031E— 60            RTS
```

2. 自动切换汉字输入方式

汉字输入方式的切换一般是用按功能键F1~F5的方法。但有的程序要求在某些时刻自动切换输入方式,这时,调用如下的子程序可以达到目的:

\$EC18——功能:置汉字输入方式;入口参数:累加器A中是F1~F5的键码,分别为\$01、\$0C、\$17、\$14、\$06)。

程序2可将当前状态转入区位输入方式,第5~6行是恢复执行前面指令时清掉的状态提示字。

【程序2】

```
0310— A9 17          LDA      # $ 17
0312— 20 AB C3       JSR      $ C3AB
0315— 20 18 EC       JSR      $ EC18
0318— 20 B9 C3       JSR      $ C3B9
031B— A9 12          LDA      # $ 12
031D— 20 2B C3       JSR      $ C32B
0320— 60            RTS
```

3. 包括状态行的清屏

\$EC54——功能:全屏幕清屏

HOME 或 PRINT CHR\$(12)在汉字状态下清不了状态行。若想连状态行一起清除,可用如下指令:

JSR \$C3AB

JSR \$EC54

JSR \$C3B9

RTS

此子程序清屏后光标仍在当前行,而不是在屏首。它也可用于清高分辨率图形第二页。

4. 打印机失控和F4/F5键引起死机的解除

\$EC94——功能:初始化打印机工作单元和功能键扩充口。

不少CEC—I的用户曾遇到这样两种情况:一是在汉字状态下工作以后,再用PR#6热启动DOS,以后就会出现打印机失控的现象;二是使用某些要用F4或F5键的程序后又热启动DOS,以后一按F4或F5键就死机。这两种现象都是因为汉字系统用的第三页单元中的值在DOS启动的过程中被破坏所致。这时,用上述初始化程序即可解决问题。可用下列指令:

JSR \$C3AB

JSR \$EC94

JSR \$C3B9

RTS

它们相当于如下BASIC程序(程序3)的功能,

【程序3】

```
10 POKE 1659,0:POKE 1787,1:POKE 1915,1:POKE
2043,40
20 POKE 913,96:POKE 911,96
30 POKE 915,76:POKE 916,27:POKE 917,65:POKE
918,51:POKE 919,1:POKE 920,24
```

也可用下列键盘操作完成同样的初始化工作:

①按复位键(CTRL)←(RESET)进入西文BASIC状态;

②POKE 1147,0

③按(中文)键进入汉字状态。

5. 不显CEC汉字封面

有的软件希望在第一次进入汉字状态时,不显示

CEC 原来的汉字封面而直接显示自己设计的封面,这时也可利用上述初始化子程序 \$EC94(程序4):

【程序4】

```
0300— A9 03      LDA      # $ 03
0302— 8D 7B 04    STA      $ 047B
0305— 20 AB C3    JSR      $ C3AB
0308— 20 94 EC    JSR      $ EC94
030B— 20 B9 C3    JSR      $ C3B9
030E— 60          RTS
```

注意不能只用程序4的前两条指令,否则由于跳过初始化打印机控制单元的步骤,虽然可以不显示 CEC 汉字封面而进入汉字状态,但一旦打印就会紊乱。

6. 获取汉字点阵

\$ECCD——功能:查找汉字点阵;入口参数:国标码高字节置入 \$D6、国标码低字节置入 \$D7,若是 ASCII 字符,则 \$D6置0、ASCII 码放入 \$D7。出口参数:查找成功则状态标志寄存器的进位位 C 为1,否则为0。调用后,若查找成功,则16×16点阵的32个字节存放在 \$94D0开始的32个单元。

该子程序在汉字及西文状态下均可使用。

由于汉字在屏幕上的点与对应内存映射区中字符的各位的对应关系是倒序的,CEC—I 为了提高效率,从汉字库中读出的字节的各位也是倒序的。所以若要得到正确的点阵数据,在调用本子程序后还要将 \$94D0开始的32个字节的各位颠倒过来。

还要指出的是,汉字屏幕上所有要显示的汉字或字符,均要用 \$94D0开始的32个字节作为暂存点阵字模的缓冲区。故调用本子程序获得点阵数据后,应立即将其从 \$94D0~\$94EF 移到安全区域,否则当显示新字或程序运行完后显示系统提示符时,均会冲掉 \$94D0~\$94EF 中有用的点阵数据。另一个办法是在西文状态下调用本子程序,就不会发生上述问题了。

程序5是以16个四位16进制数的形式按行向量输出一个汉字或 ASCII 字符的16×16点阵数据,该程序可作为一个子程序被调用,它有两个入口,入口1为 \$6000,调用前须将汉字的内码高字节放 \$300、低字节放 \$301。入口2为 \$6010,调用前将汉字的国标码高字节放入 \$D6,低字节放入 \$D7。对字符, \$D6置0、\$D7放入 ASCII 码。调用后汉字点阵在 \$310开始的32个字节中,程序5既可在中文状态下运行也可在英文状态下运行。其中地址 \$6029~\$603F 部分是调用监控子程序以四位16进制形式输出点阵数据。

程序5—1是调用程序5的 BASIC 程序,此程序可以高速地获得国标一、二级汉字中的任何汉字或符号的点阵数据。

【程序5】

```
6000— AD 00 03 20 92 C3 85 D6
6008— AD 01 03 20 92 C3 85 D7
6010— 20 AB C3 20 CD EC 20 B9
6018— C3 90 25 A0 1F B9 D0 94
6020— 20 41 60 99 10 03 88 10
6028— F4 A0 00 B9 10 03 C8 BE
```

```
6030— 10 03 20 41 F9 20 48 F9
6038— C8 C0 20 90 EE 20 8E FD
6040— 60 A2 08 0A 6E 02 03 CA
6048— D0 F9 AD 02 03 60
```

【程序5—1】

```
20 PRINT :PRINT CHR $(4);“BLOAD PRO5”
25 CH = 768;CL=769;FC=24576
30 INPUT “汉字?”;A$;IF A$=”” THEN 90
40 POKE CH,ASC(MID$(A$,2,1));POKE CL,ASC
  (RIGHT$(A$,1))
50 CALL FC
60 GOTO 30
90 DEL 20,20
]RUN
汉字?啊
0004 2F7E F904 A904 AA14 AA7C AC54 AA54
AA54 A954 E974 AD54 0A04 0804 0814 080C
汉字?
```

7. 在第11行显示汉字

\$EE0D——功能:显示字符或汉字;入口参数:A 中放字符的 ASCII 码或汉字码。这里的汉字码为三字节,第一个字节为汉字引导符 \$7F,第二、三两个字节为汉字的异形国标码的高字节和低字节,显示一个汉字须分三次调用。

与 CSWA 和 GB·CSWA 不同,此子程序能在屏幕的任何地方显示汉字,包括第11行(提示行),而前者在第11行显示时,和 BASIC 中一样,超过20列时显示的汉字会出现残缺不全的现象。

程序6是在屏幕第11行第33列处显示“A 啊”。

【程序6】

```
0300—A5 24 48 A5 25 48 A9 0A
0308—85 25 A9 20 85 24 20 AB
0310—C3 A9 41 20 0DEE A9 7F
0318—20 0DEE A9 B0 20 0D EE
0320—A9 A1 20 0D EE 20 B9 C3
0328—68 85 25 68 85 24 60
```

8. 造字

\$F0B5——功能:将 \$94D0开始的32个字节表示的16×16点阵显示出来。

利用该子程序可以造任意16×16点阵的汉字或其它符号,这只需在调用前将点阵数据放入上述单元即可,但要注意,点阵数据字节的各位应置为左右相反。另外,该子程序只能用于显示。

程序7是供显示用的造字程序,使用该程序前应将用户造的字库置于 \$7000开始的区域,每32个字节为一个自造字的16×16点阵,各个字依次排队,并将 \$3F5~\$3F7三个单元置为 \$4C、\$00、\$60,调用时用命令“&n”,表示显示用户造的第 n 个字,该程序中已有先颠倒点阵字节各位的功能,故自造点阵中的各字节可直接按正序给出。

其实,不只是造汉字,只要是16×16点阵的任何符号或图形均同样可造,用户只要自己设计所需的点阵即可。还可以利用造图形的方法获得动画效果。程序

7-1是一个简单的例子,它在屏幕上显示一个不断张口、闭口的小精灵。方法是造两个相似的小精灵图形,一个张口,一个闭口,在同一位置交替显示,并适当控制显示时间。程序7-1中第三行是将磁盘上的点阵数据装入内存,在这个例子中用的点阵数据是:

```
7000— 0F F0 3F FC 7F FE FF FF
7008— C3 C3 9B 9B BF BF DF DF
7010— EF FB 63 E3 70 06 30 06
7018— 38 0E 1C 1C 0F 78 07 E0
7020— 0F F0 3F FC 7F FE FF FF
7028— C3 C3 9B 9B BF BF DF DF
7030— FF FF 7F FB 77 F6 3B EE
7038— 3C 1E 1F FC 0F F8 07 E0
```

【程序7】

```
6000— A0 00 B1 B8 C9 30 90 18
6008— C9 3A B0 14 20 7B DD 20
6010— FB E6 8A 18 69 20 85 D7
6018— A9 2A 85 D6 20 23 60 60
6020— 4C C9 DE 20 AB C3 20 30
6028— 60 20 B5 F0 20 B9 C3 60
6030— A5 D6 C9 2A F0 06 20 CD
6038— EC 4C 6A 60 A9 00 85 06
6040— A9 70 85 07 A5 D7 38 E9
6048— 20 AA CA F0 10 A5 06 18
6050— 69 20 85 06 A5 07 69 00
6058— 85 07 4C 4A 60 A0 1F B1
6060— 06 20 6B 60 99 D0 94 88
6068— 10 F5 60 A2 08 0A 6E 78
6070— 60 CA D0 F9 AD 78 60 60
```

【程序7-1】

```
1 B$=CHR$(13)+CHR$(4)+"BLOAD"
2 PRINT B$;"PRO7"
3 PRINT B$;"PRO7-1.LIB,A$7000"
4 POKE 1013,76;POKE 1014,0;POKE 1015,96
5 PRINT CHR$(12)
6 FOR J=1 TO 10
7 GOSUB 100
10 VTAB 4;HTAB 10;& 1
15 GOSUB 100
20 VTAB 4;HTAB 10;& 2
50 NEXT
90 DEL 1,3;END
100 FOR I=1 TO 300;NEXT
120 RETURN
```

9. 清屏幕任一行

\$F388——功能:清任一行(包括提示行);入口参数;\$25单元中置要清的行的行坐标(0~10)。

程序8的作用是清除提示行,若要清其它行,只要将\$307单元的直值为相应的行坐标即可。

【程序8】

```
0300— A5 24 48 A5 25 48 A9 0A
0308— 85 25 20 AB C3 20 88 F3
0310— 20 B9 C3 68 85 25 68 85
```

0318— 24 60

10. 在汉字状态中打印文本字符

\$F656——功能:打印文本字符;入口参数:累加器A中是打印字符的ASCII码。该子程序可在各种状态(中文、西文、高分辨率图形、低分辨率图形)中打印出文本ASCII字符(包括控制字符),而且它不需要连通和断开打印机的指令。

程序9运行时打印出“ABCDE”五个字符,妙就妙在它在中文状态下打印出7×5点阵的文本字符。使用程序9时将要打印字符的ASCII码置于\$314开始的单元,最后跟回车码\$0D和结束标志\$00。

【程序9】

```
0300—A2 00 20 AB C3 BD 14 03
0308—F0 06 20 56 F6 E8 D0 F5
0310—20 B9 C3 60 41 42 43 44
0318—45 0D 00
```

11. 打印机换页

若利用上述子程序\$F656向打印机输出一个换页码\$0C,则可在汉字状态(也可在其它任何状态)下使打印机换页,即用如下指令:

```
LDA # $0C
JSR $C3AB
JSR $F656
JSR $C3B9
RTS
```

汉字系统中可供用户程序直接调用的子程序还有很多,由于篇幅所限就不一一列举了。

新书征订

- | | |
|----------------------------|--------|
| ① 电脑编程技巧 200例 | 9.80元 |
| ② 电脑维修经验 200例 | 6.30元 |
| ③ 微机可编程序控制器原理使用及应用实例 | 5.50元 |
| ④ 实用家电维修与制作集锦 第一集 | 13元 |
| ⑤ 电冰箱、冷藏柜、空调器、电动机维修技术和修理经验 | 8.40元 |
| ⑥ 国内外黑白电视机修理经验300例(续集一) | 15元 |
| ⑦ 业余电子制作详解(初级篇) | 8.30元 |
| ⑧ 业余电子制作电路技术手册 | 8.60元 |
| ⑨ 音箱制作精选 | 16.10元 |
| ⑩ 实用录像机维修技术 | 14元 |
| ⑪ 微型计算机系统原理分析与维修(上、中、下) | 22元 |
| ⑫ IBM-PC和长城系列微型机操作应用实践问答 | 14.30元 |

购书者请汇款至北京农展馆南路十二号(通广大厦)中国电子学会北京爱谊电子服务部 邮编100026。

本服务部还为读者提供电子类书刊100余种,购书者可 直接来函索取书目(自付邮资0.30元)。

CATALOG 命令的改进

广西南宁第二十四中学(530001)陈治浩

本文提出了一种对 APPLE DOS 中 CATALOG 命令的改进方案(见程序一和程序二)。改进后的命令能在列完目录后显示出磁盘上的文件数目和自由空间的扇区数,并且使 CATALOG 命令适应于中文状态。

一、磁盘中文件数和自由扇区数的显示

及时地了解磁盘的使用情况,无疑是很意义的。而利用执行 CATALOG 命令时顺便显示磁盘中自由扇区数的方法也许是最为实用的了,具体实现的方法本刊以前曾有过介绍。但是,将自由扇区数显示在原命令显示磁盘编号的地方,笔者认为是一种缺憾。因为了解磁盘编号对于管理磁盘也是很有益处的,特别是在使用的磁盘比较多时更是如此。本文介绍的方法保留了原命令中显示磁盘编号的功能,当 CATALOG 命令列完目录以后才显示出磁盘中的文件数和自由扇区数,其显示格式与 PC-DOS 中的 DIR 命令一样。如果读者使用的 DOS 已经过修改,CATALOG 命令具有中断列目录功能的话,那么显示出的文件数仅是在中断以前已经被列出的文件数目。

二、CATALOG 命令在中文状态下的使用

APPLE DOS 设定文件名由30个字符(ASCII 码)组成,即使我们的文件名只有一个字符,DOS 也要用空格将其补足30个字符。这种设定不适用于 CEC-I 等微机在中文状态下的使用。CEC-I 等微机在中文状态下每行只能显示34个字符,一个汉字在显示时占两个字符的位置。列目录时,每个文件名前面要用7个字符来显示文件类别、长度等信息,这样就要两行才能列完一个文件名。而实际上一般文件名的长度并没有(也不需要)30个字符,后面全是空格,故列目录时看起来就是空一行列一个文件名了。如果我们在文件名中加入四个以上的汉字,列目录时“空行”就不存在了。这是因为每个汉字在内存中用三个字节表示,而显示时只占两个字符的位置,含四个以上汉字的文件名在显示时的长度不会超过26个字符,故在一行里可以显示完毕,“空行”也就自然消失了。

我们只要在 DOS 中修改一个字节(程序二中的60语句),就可使 CATALOG 命令只显示文件名前面的26个字符。这样,不管文件名中有无汉字,都不会在列目录时产生空行了。这时,如果某一文件名的长度超过26个字符的话,在列目录时其超出的部分将被略去,我们就看不到完整的文件名了,但在磁盘中的文件名仍是完整的。

程序二中的50语句使 CATALOG 命令每页只列八

个文件,若在西文状态下使用可将其删去。

若用 INIT 命令格式化一张新磁盘,载入新磁盘的将是修改后的 DOS,程序一、二均没有必要保留在磁盘上了。

程序一

```
BC56- 20 2F
BC58- AE A9 00 85 06 85 08 85
BC60- 09 60 20 ED FD E6 06 60
BC68- 20 70 FC 20 48 F9 A6 06
BC70- A9 00 20 24 ED A2 0B BD
BC78- B7 BC 20 ED FD CA D0 F7
BC80- A0 A0 B9 F2 B3 F0 0D 0A
BC88- 90 FB 48 E6 08 D0 02 E6
BC90- 09 68 D0 F1 88 D0 EB A6
BC98- 08 A5 09 20 24 ED A2 0D
BCA0- BD AB BC 20 ED FD CA D0
BCA8- F7 4C 7F B3 E5 E5 F2 E6
BCB0- A0 F3 F2 EF F4 E3 E5 F3
BCB8- A0 A0 A0 A9 F3 A8 E5 EC
BCC0- E9 C6 A0
```

]BSAVE DIR.1,A\$BC56,L\$6D

程序二

```
10 POKE 44487,86:POKE 44488,188
20 POKE 44518,98:POKE 44519,188
30 POKE 44589,104:POKE 44590,188
40 PRINT CHR$(4);"BLOAD DIR.1"
50 POKE 44452,12:POKE 44605,8
60 POKE 44567,25:NEW
```

中华学习机(CEC-I) 查找汉字区位码

福建 永定一中(364100)胡瑞辉

随着高考的不断改革,许多省市的高考招生用计算机统计分数和录取志愿。由于计算机处理的需要,考生在报名时应填写考生信息卡,要求考生将自己的姓名用国标区位码来填写。通常用人工对照区位码表来查找汉字区位码是比较麻烦的,而且容易查错。本文的程序可以帮你解决困难。

程序运行后首先进入提示是否要打印输出,按下功能键 F2 后即可用汉语拼音输入所要查找的汉字姓名。40—120 语句和子程序 500—560 语句是实现将所输入的汉字的 ASCII 码转换成区位码,其中字符串数组 AS(I) 的每一个元素存放一个汉字的区位码。130—160 语句是输出所查找的汉字区位码。

```
10 PR# 3: PRINT: HOME
15 VTAB 5: HTAB 8: INPUT "是否要打印?(Y/N)";C$
20 HOME: VTAB 3: HTAB 2: PRINT "按 F2 键后,请输入汉语拼音查找姓名"
```

```

25 IF C$ = "Y" THEN POKE 1659,3
30 VTAB 5: HTAB 10: INPUT "姓名:";A$
40 A = LEN (A$):N = INT (A/3)
50 FOR I = 1 TO N:FOR J = 0 TO 1
70 X$ = MID$ (A$,3*I - 1 + J,1)
80 X = ASC (X$)
90 GOSUB 500
100 A$ (I) = A$ (I) + Y$
110 NEXT J:NEXT I
130 VTAB 6: HTAB 10: PRINT "区位:";
140 FOR I = 1 TO N
150 PRINT A$ (I),"",
155 A$ (I) = ...
160 NEXT I: PRINT : PRINT: POKE 1659,0
170 VTAB 8: HTAB 5: PRINT "按任意键查找另一姓名区
    位"
180 GET B$: GOTO 20
500 IF X < 34 THEN X = X - 28: GOTO 540
510 IF X < 44 THEN X = X - 29: GOTO 540
520 IF X < 58 THEN X = X - 30: GOTO 540
530 X = X - 31
540 Y$ = STR$ (X)
550 IF X < 10 THEN Y$ = "0" + Y$
560 RETURN
]RUN

```

姓名:张小彬
 区位:5337,4801,1782,
 姓名:李虹
 区位:3278,2671,

监控程序的浮动

江苏、苏州、景范中学(215005) 赵旭

中华学习机的监控程序驻在 \$F800~\$FFFF 空间。利用监控命令可以方便地对内存中的信息进行检查、修改、比较、传送等工作。但中华学习机的汉字管理程序及 16K RAM 卡所占空间都与监控所占空间有重叠。当计算机切换到汉字管理程序或 16K RAM 卡时,监控程序就不能运行。也就不能直接利用监控命令来剖析汉字管理程序或已驻在 16K RAM 卡中的程序。更不能利用监控命令将机器程序直接写入 16K RAM 卡。

针对以上情况,笔者采用将监控程序先移到 \$2800—\$2FFF 空间,修改程序中有关绝对转移地址,再调用浮动后的监控程序的方法,解决了以上问题。

以上工作是由一个工具软件完成。具体操作方法:先输入程序一和程序二。再运行程序一,即刻计算机发出“嘟”声,并进入浮动后的监控程序。以后如再要进入

浮动后的监控程序,只要在 BASIC 状态键入 CALL24576(或监控状态键入 6000G)。

应用举例:1. 在监控状态键入 C3AB G,中华机将内存切换到汉字管理程序。此时即可用监控中 L 命令直接阅读汉字管理程序了,也可用 M 命令将汉字管理程序移到其他空间。2. 键入 C3B9 G 退出汉字管理程序后,再键入 C08B C08B,中华机将内存切换到 16K RAM 卡。如果 RAM 卡中已有程序,则可用 L 命令直接阅读。当然也可直接用监控命令修改 RAM 卡中的程序,或直接将机器程序写入 RAM 卡中。

程序一

```

10 FOR I=512 TO 527
15 READ A:POKE I,A
20 NEXT I
25 POKE 1017,16:POKE 1018,96
30 CALL -144
50 DATA 178,184,176,176,188,198,184,176,176,
    174,198,198,198,198,153,141

```

程序二

```

6000- A9 2D 8D 54 AA 8D 56 AA
6008- A9 2E 8D BF 2F 4C 65 2F
6010- 20 75 FE 20 2C FE A2 00
6018- 20 8C F8 A5 2F C9 02 D0
6020- 45 A0 01 38 B1 3A E5 40
6028- CB B1 3A E5 41 90 37 A0
6030- 01 A5 3E F1 3A A5 3F C8
6038- F1 3A 90 2A 38 A0 01 A5
6040- 40 E5 44 85 FE A5 41 E5
6048- 45 85 FF 38 A5 3A E5 FE
6050- 85 00 A5 3B E5 FF 85 01
6058- B1 3A 38 E5 FE 91 00 C8
6060- B1 3A E5 FF 91 00 20 53
6068- F9 85 3A 84 3B 18 A5 3A
6070- E5 3E A5 3B E5 3F D0 9E
6078- 4C 00 60

```

中华学习机特殊使用技巧

北京二十二中(100007) 王冈

1. REM 加密

这是一种较难被识破的 BASIC 程序加密方法,用这种方法加密的程序只能运行,而不能列出清单。这种加密方法的关键在于借用了 DOS 命令的引导字符 CTRL-D,所以不使用磁盘操作系统的机器不能使用此方法加密。

下面介绍具体操作步骤:

(1)在需要加密的程序前加入一行 REM 语句。

例如:0 REM ABCDEFG

(2)CALL-151 进入监控状态

(3)键入以下数据:

806:0D 04 50 52 23 36 0D

(4)3D0G 返回 BASIC 状态

这个程序即加密完毕。它可以运行,也可以存盘,但如果要执行列表命令 LIST,机器不但不能列出程序,反而会去引导 DOS,原来的程序自然也被清除了。

2. 磁带 BASIC 程序的自动运行

使用过磁盘操作系统的用户都知道使用 RUN 命令可以从磁盘上调入一个 BASIC 程序并立即运行,而使用磁带存取程序却没有这种功能。但是,只要按以下方法操作,就能做出可自动运行的磁带 BASIC 程序。

(1)键入一个 BASIC 程序(或从磁盘、磁带上装入)。

(2)POKE 82,213

(3)SAVE 将程序写入磁带

等到写带完毕,一个可自动运行的程序便做好了。以后只要用 LOAD 命令把程序从磁带装入内存,它就会自动运行。

3. 特殊命令 GAME

中华学习机,有一条未公布的命令—GAME(有些厂家生产的中华学习机没有此命令),这条命令的作用是使主机的喇叭中发出一声短促的鸣叫,所以可将它作为一条发声语句写入程序中。但是,由于此命令触动了一些软开关,所以不能在中文状态下使用。

4. LOGO 语言的截取

中华学习机固化了 LOGO 语言子集,使用户用 LOGO 语言编写过程十分方便。但是,由于一些 APPLE 机没有与 CEC—LOGO 兼容的 LOGO 系统盘,所以使用在 APPLE 机上调试 CEC—LOGO 语言过程的愿望成为泡影。但是,可以通过截取中华学习机上的 LOGO 语言系统程序的方法来达到这个目的。具体作法为:

(1) MAXFILES1

(2) CALL-151

(3) C007:00 (将 I/O 空间切换到内部 ROM 空间)

(4) C600G (将 ROM 中的 LOGO 解释程序装入 RAM)

(5) 3D0G

(6) BSAVE LOGO,A\$1BF6,L\$7D8A

(7) 输入一段引导程序:

10 D\$=CHR\$(4)

20 PRINT D\$;"MAXFILES1"

30 PRINT D\$;"BRUN LOGO"

用 SAVE 命令将此程序存盘。

以后,在 APPLE 机上只要运行这张盘上的引导程序,便可以将 CEC—LOGO 解释程序装入内存,输入、调试或运行在中华学习机上编的 LOGO 过程(使用 CEC—LOGO 的 APPLE 机应有 64KB 的内存)。

有趣的对比

北京西北三条10号(100034)

中国 FORTH 应用研究会 丁志伟

91年第8期和第9期分别刊登过两个计算1000的阶乘的程序。一个是用6502机器语言编的,运行时间是5分8秒;一个是C语言编的,在286机上运行时间是15秒。这里再介绍两段阶乘程序,都是在中华学习机 CEC—I 上分别用 FORTH 语言和 BASIC 语言编写,以资比较。

FORTH 程序

```
0 VARIABLE KS 0 VARIABLE ZZH
CODE *1W/MOD (略) ENDCODE
CODE SUAN1W (略) ENDCODE
DEC (DECIMAL)
: XIAN DEC 20480 ZZH @
DO I @ DUP 1000 < IF 0. THEN
    DUP 100 < IF 0. THEN
        DUP 10 < IF 0. THEN
            -2 +LOOP ;
: N! 20480 DUP DUP ZZH ! KS ! ON
1001 1
DO 0 ZZH @ 1+ KS @
    DO I @ J *1W/MOD ROT SUAN1W I !+
    2 +LOOP ?DUP
    IF 2 ZZH +! ZZH @ ! THEN
        KS @ @ 0= IF 2 KS +! THEN
LOOP XIAN ;
```

计算1000!运行时间是13分58秒。其中 SUAN1W 和 *1W/MOD 是用汇编语言方法定义的 FORTH 动词(相当于语句),前者意思是把两数相加再除以10000,取余数和商,后者是把两数相乘再除以10000,取余数和商。

有些人对 FORTH 语言不太了解,其实若有一本合适的教科书做参考,读懂以上程序并不难。以下的 BASIC 程序是仿照上边程序的算法编写的。

```
10 DIM A(3000):A(0)=1:PRINT
20 W=1000000:
30 INPUT N:FOR I=1 TO N
40 FOR J=M TO K:A=I*A(J)+B:B=INT(A/W):
NEXT
50 IF B THEN K=K+1:A(K)=B:B=0
60 IF NOT A(M) THEN M=M+1
70 NEXT
80 PRINT A(K):,IF K=0 THEN RUN
100 FOR I=K-1 TO 0 STEP -1
110 IF A(I)<1000 THEN PRINT 0;
120 IF A(I)<100 THEN PRINT 0;
130 IF A(I)<10 THEN PRINT 0;
```


140 PRINT A(1);,NEXT;RUN

计算到400!耗时8分6秒,推算1000!要用1小时略多。若经过编译,速度加快,但也需时29分多。

下面对 BASIC、FORTH 和汇编做个比较,这都是 APPLE 机(或兼容机)运行的。

速度。汇编语言最快,BASIC 最慢。上边两段程序中 BASIC 循环次数只用 FORTH 循环次数的2/3,应该加快,再经编译也能提高速度,但仍不如 FORTH。FORTH 是一种非常重视效率的语言。

编程效率。汇编语言编写费时、调试麻烦已是人所共知。FORTH 和 BASIC 都是高级语言,编写、调试都方便得多。前边这段 BASIC 程序是在 FORTH 之后编的,编程比 FORTH 快,不过 BASIC 由于执行速度太慢,测试起来相当费时间。据笔者经验,程序规模越大,用 FORTH 编程效率越高。

改进情况。一般说来,一段程序被编出来之后,总有改进余地。汇编语言程序较长,改进起来也会相当麻烦。BASIC 语言编的这段程序是反复改进的结果,看来已无法再完善以提高速度了。FORTH 程序也经过几次改进,每次都使得程序更简单,执行速度更快。其中一种改进方法是把某些高级成分用汇编重新定义,能有效地提高速度。这段程序还有改进余地,由于没有实际用途,就没继续改进。

我们前后发表计算阶乘的文章共三篇,主要是为了比较各种计算机语言的性能(当然不是严格和全面的),也是想借此向读者介绍 FORTH。自91年第8期“谈谈 FORTH”刊出后,引起了广泛兴趣。有关 FORTH 的学习资料及软件咨询,请直接与丁志伟同志联系。本刊将陆续组织 FORTH 学习和应用方面的材料,欢迎投稿。 编者

中华学习机汉化引导程序

北京中国农业电影制片厂(100081)张福森

CEC—I 中华学习机具有汉字系统,用中文作文件名,对于没有英文基础的使用者,将会是很方便的。

仿照 APPLE 机西文自动引导程序中的一种,参考有关资料,我编写了一个用中文作文件名的汉化自动引导程序。

把西文状态的自动引导程序汉化,关键在于屏幕字符的查询。本程序中20行的一段机器语言子程序,经机内入口地址为 \$C389 的 USR.DCOD 子程序,把从 \$07 单元读入的异形国标码转换为学习机内码后,送入 \$08 单元。300行至330行的 BASIC 子程序调用该段机器语言子程序,完成屏幕字符查询。其中300行处理 ASCII 码,310行、320行处理汉字。

针对中文方式下屏幕显示行数少的特点,30行至40行的一段机器语言修改了 DOS 中 CATALOG 命令的显示格式。令其显示适当行数后暂停,按回车键中止列目录,进入自动引导方便使用。程序中设置了两页屏幕显示说明文字。

本程序对 BASIC 文件和一般二进制文件操作效果良好。对于存储和显示与中文方式有冲突的二进制图像文件,应用 FID 程序中单文件拷贝功能将其拷贝到用本程序格式化的磁盘上,再把名字改为中文。切不可在中文方式下直接存盘。否则不能正常运行。此时应把程序中280行作如下改动,即可正常运行。

```
280 PRINT CHR$(14);PRINT CHR$(17);PRINT D$;B$
```

注意每个文件名应使用四个以上汉字,否则在列目录时会出现空行,致使自动定位寻找屏幕字符功能出错。以 CEC—I 中文目录为文件名,用本程序格式化

磁盘,在以后的使用中,你一定会感到它的方便实用。

```
10 HOME;DATA 165,7,32,137,195,133,8,96,32,12,
    253,201,14,1,208,3,76,44,174,76,60,174
20 FOR I=768 TO 775:READ A;POKE I,A;NEXT
30 FOR I=43453 TO 43465:READ A;POKE I,A;NEXT
40 POKE 44452,6;POKE 44601,76;POKE 44602,189;
    POKE 44603,169;POKE 44605,5
50 D$=CHR$(4);C$="CATALOG";PRINT D$;"PR
    #3";PRINT CHR$(18);PRINT CHR$(12)
60 HTAB 5;PRINT " * * * 本磁盘用中文作文件名 *
    * "
70 HTAB 5;PRINT "列目录时,第一幕列出七个文件名暂
    停。按【CR】键即可退出列目录,进入自动引导。按其它
    键继续列目录"
80 HTAB 5;PRINT "按文件名前的代表字符即可运行该文
    件。按数字键后,再按文件名前的代表字符,即可进行以
    下注明的功能操作:"
90 HTAB 22;PRINT "按任一键继续";PRINT CHR$(19);
    PRINT CHR$(12)
100 PRINT "1—LOAD(装载文件)";
    PRINT "2—LOCK(文件加锁)";
    PRINT "3—UNLOCK(文件解锁)";
110 PRINT "4—DELETE(删除文件)";
    PRINT "5—TEXT(退出本程序)"
120 VTAB 8;HTAB 17;PRINT "按任一键开始列目录";
    PRINT CHR$(19);PRINT CHR$(12)
130 PRINT D$;C$;PRINT CHR$(13)
140 T=0;H=4;FOR V=1 TO 7;GOSUB 300;IF F<=64
    AND F<>32 THEN VTAB V;HAB H;PRINT CHR
    $(219)CHR$(T+193)CHR$(221);T=T+1;S=V
```

如何在苹果机持续运行一程序

辽宁大学611#(110036) 张 强

有时需要在计算机上连续不间断运行一程序,例如计算 π 的高精度解,常常需要几天甚至几个月的连续运算。然而操作人员和机器不能长时间不间断地工作,为了解决这个问题,我们可利用外存来帮助。

我们知道,APPLE II及其兼容机上运行BASIC时,解释程序将BASIC产生的数据存放在BASIC程序后的存储器内,通过第零页的几个指针查找。在Applesoft解释程序工作时,第零页起到至关重要的作用。根据以上两点,我们只需将第0页及数据区储存到外存即起到保护BASIC工作环境的作用。具体作法如下:

1. 运行BASIC程序(在DOS环境下)。

2. 连续运算几小时后中断(用Ctrl-C)。

3. CALL-151进入监控,查看变量区首址指针(\$69,\$6A),自由区首址指针(\$6D,\$6E),字符串区首址指针(\$6F,\$70),HIMEM值指针(\$73,\$74);记下这些指针数值,并计算各区长度(注意:当程序无新串产生时,不必查看字符串指针及HIMEM值)。

4. 将零页,数据区分别存入外存,关机。

这样下次开机引导DOS后,将程序、数据区、零页所对应的文件分别调入内存后,打入CONT便继续运行了。

例如有以下测试程序:

```
10 A$="This is ";B$="a test program!";J=10
20 C$=A$+B$
30 FOR I=1 TO 1000;FOR J=1 TO 1000;NEXT J
~~~~~
150 NEXT V
160 B$="RUN";K=PEEK(-16384);IF K<128 THEN
    FOR K=1 TO 5;NEXT K;K=FREE(0);GOTO 160
170 POKE -16368,0;K=K-176;IF K<1 OR K>5 THEN
    250
180 IF K=5 THEN SPEED=255;END
190 IF K=1 THEN B$="LOAD"
200 IF K=2 THEN B$="LOCK"
210 IF K=3 THEN B$="UNLOCK"
220 IF K=4 THEN B$="DELETE";PRINT CHR$(15)
230 A$="哪一个文件?";PRINT B$;A$
240 PRINT CHR$(7);PRINT CHR$(14);GET K$;K=
    ASC(K$)-48
250 IF K<17 OR K>T+16 THEN 160
260 H=2;V==S-T+K-16;GOSUB 300;IF F=66 AND
```

```
40 PRINT I,C$
```

```
50 NEXT I
```

```
]RUN
```

```
1 This is a test program!
```

.....

^C中断,屏幕最后显示为:

```
22 This is a test program!
```

```
]CALL-151
```

*69.74 假设这n个单元分别为:

变量区首址:\$830;自由区首址:\$845

字符串区首址:\$95E0;HIMEM值:\$9600

变量区长度=\$845-\$830=\$15

字符串区长度=\$9600-\$95E0=\$20

(^C返回BASIC状态)

```
]SAVE TEST PROGRAM
```

```
]BSAVE ZERO PAGE,A$0,L$FB
```

```
]BSAVE DATA,A$830,L$16
```

```
]BSAVE CDATA,A$95E0,L$20
```

此时关机,下次开机引导DOS后:

```
]LOAD TEST PROGRAM
```

```
]BLOAD DATA
```

```
]BLOAD CDATA
```

```
]BLOAD ZERO PAGE
```

```
]CONT
```

屏幕出现:

```
23 This is a test program!
```

:

```
(B$="RUN" OR B$="LOAD" THEN B$="B"+B$
270 N=0;FOR H=7 TO 34;GOSUB 300;N$=N$+C$;
    NEXT H;B$=B$+N$;PRINT CHR$(15);PRINT
    LEFT$(B$,34)
280 PRINT CHR$(14);PRINT D$;B$
290 CLEAR;GOTO 50
300 D=37376+2*(H-1)+68*(V-1);E=PEEK(D);F=
    PEEK(D+1);IF E<>127 THEN C$=CHR$(F);
    GOTO 330
310 N=N+1;F=F+128;POKE 7,F;CALL 768;W=PEEK
    (8);IF N=1 THEN C$=CHR$(127)+CHR$(W);
    GOTO 330
320 C$=CHR$(W);N=0
330 RETURN
```

ProDOS 磁盘操作系统入门(续)

廖 凯

2. 增加数据

APPEND 命令可以在一个顺序文件的末尾增加数据而不会破坏前面的数据。APPEND 命令首先打开文件,并将指针指到文件的末尾,然后发出 WRITE 命令。若增加资料到顺序文本文件(见上例)内,就将行号210和220合并为:

```
220 PRINT D$;"APPEND/DATA.DISK/EXAMPLE"
再次运行这程序并输入其它的数据,你在运行下面的程序(Read Sequential File)时,可以看到结果。
```

3. 读顺序文件

READ 命令可以使用 F# 参数,其一般形式是:

```
10 PRINT D$;"READ/卷目录/文件名,F#"
下面举例说明 READ 命令的用法,并将结果打印到屏幕上。
```

```
10 REM Read Sequential File
50 DIM N$(100),C$(100),P$(100)
100 D$=CHR$(4);PRINT D$;"PR#3"
120 PRINT D$;"OPEN/DATA.DISK/EXAMPLE"
130 PRINT D$;"READ/DATA.DISK/EXAMPLE"
140 ONERR GOTO 260
150 PRINT "NAME";TAB(25);"COMPANY";TAB(25);
    "PHONE";PRINT
160 I=I+1
170 INPUT N$(I);INPUT C$(I);INPUT P$(I)
180 X1=LEN(N$(I));X2=LEN(C$(I));X3=LEN(P$(I))
190 IF X1>26 THEN N$(I)=MID$(N$(I),1,25);X1=25
200 IF X2>26 THEN C$(I)=MID$(C$(I),1,25);X2=25
210 IF X3>21 THEN P$(I)=MID$(P$(I),1,20);X3=20
220 PRINT N$(I);TAB(29-X1);C$(I);TAB(32-X2);P$(I)
230 GOTO 160
240 PRINT D$;"CLOSE";END
260 POKE 216,0
270 GOTO 240
```

此程序首先打开文件 EXAMPLE 并准备读取,行号150在屏幕上显示三个标题。行号170的 INPUT 语句将从 EXAMPLE 中读取数据作为输入数据。行号180将各字串的长度存入相应的 X 变量内,行号190至210使各类数据在屏幕上以竖列(栏)的形式显示,并使各列向左对齐。行号140建立错误中断,转至行号260清除错

误并关闭文件。

①INPUT 与 GET 的区别:INPUT 和 GET 语句都可以从文件读取数据,但有所区别。INPUT 一次可以从文件读取若干个数据域。若一串字符中包含逗号或冒号,INPUT 语句将以此符号作为结束符,并放弃剩余的字符,而 GET 一次只从文件中读一个字符。GET 语句将不影响读取逗号和冒号。

②读取任何文件:ProDOS 不支持 MON 和 NOMON 命令。因此,磁盘 I/O 信息不能反映到屏幕上,我们不知道写入文件内的是什么。我们可用下面的程序来解决这个问题。这程序既可在顺序文件内又可在随机文件内读取和列印每个字符。

```
100 REM RETRIEVE TEXT WITHOUT MON
110 D$=CHR$(4);PRINT D$;"PR#3"
120 INPUT "FILE NAME TO READ?";N$
130 PRINT D$;"OPEN"N$
140 PRINT D$;"READ"N$
150 ONERR GOTO 210
160 GET A$;PRINT A$;
180 GOTO 160
190 PRINT D$;"CLOSE";END
210 POKE 216,0
220 PRINT;PRINT"End of File"
230 GOTO 190
```

4. 读取其它文件类型

ProDOS 允许读取任何文件类型。一个有用的例子是读取一个卷目录或各种子目录。在一个程序内,用户会有机会看到各种文件。下面的程序将列印卷目录,并在屏幕上按四列格式列印文件名。另外,如果屏幕空间是有限的,那么你可以增加一个 IF/THEN 语句来终止显示。

```
90 REM Horizontal Catalog
100 V=5;I=0;D$=CHR$(4)
110 PRINT D$;"PR#3"
115 PRINT D$;"PREFIX"
120 INPUT " ";P$
125 PRINT D$;"OPEN"P$";TDIR"
130 PRINT D$;"READ"P$
135 ONERR GOTO 210
140 INPUT A$
145 IF A$="" THEN 140
150 IF MID$(A$,18,1)="D" THEN INVERSE
155 IF MID$(A$,1,12)="BLOCKS FREE;" THEN A$=""
160 IF MID$(A$,1,5)="NAME" THEN A$="";GOTO
```

```

140
165 IF LEFT$(A$,1)="/" THEN B$=MID$(A$,1,
15);GOTO 175
170 B$=MID$(A$,2,16)
175 H=(I*18);GOSUB 215
180 PRINT B$;NORMAL
185 I=I+1;IF I=4 THEN I=0;V=V+1
190 IF LEFT$(A$,1)="/" THEN V=V+1;I=0
195 GOTO 140
200 PRINT D$;"CLOSE";END
210 POKE 216,0;GOTO 200
215 POKE 1403,H;VTAB V;RETURN

```

行号115和120读取和设置当前磁盘的部首,将其值存入P\$内。行号125打开目录(P\$)。由于P\$不是文本文件,故必须指定文件类型(,TDIR)。行号140读取首个数据域或第一个文件名。每个文件名在起始位置保留一个空格,若文件上锁则留给上锁标记(*星号);若使用目录,则为/(斜线)所留。行号150至165检查文件的不同属性。若找到目录文件将以反相显示。行号170至190读取文件名并打印,设置显示窗口,记录要被打印到屏幕上的各个文件。

(二)、随机文件

随机文件的建立和检索方法与顺序文件相似,不同的是随机文件必须指明记录长度和记录号。

在使用随机文件时,要查明当前的部首,以便将数据写到正确的目录内。在文件被打开时,ProDOS 查看文件名是否已存在。若文件名不存在,则建立一个,指定缓冲区空间将文件指针指到起始位置;若卷目录不存在,则发生错误并中断程序。

随机文件是以记录为单位进行数据读写。每个记录可以包含若干数据域。一个数据域最多可包含230个字符。

1. 记录长度

当一个随机文件首次被打开时,必须定义记录长度。在程序内用L参数指定一个记录的长度。如果没指定长度,那么ProDOS将预定长度为1。记录长度的取值范围在1~65535之间。其格式为:

```
10 PRINT D$;"OPEN/卷目录/文件名,L#"
```

不同的随机文件有不同的记录长度,但在同一个随机文件内,记录的长度都是一样的。

2. 写随机文件

在写随机文件时,必须指定记录号:

```
10 PRINT D$;"WRITE/卷目录/文件名,R#"
```

WRITE命令可以使用B#参数或F#参数。B#参数表示从该记录的第几个字节开始执行写操作,而F#参数则表示从当前位置向前移动若干个数据域,从此数据域开始执行写操作。

3. 读随机文件

在使用READ命令时必须指明记录号,象顺序文件一样,随机文件使用INPUT或GET语句从文件中读取数据。其一般格式为:

```
10 PRINT D$;"READ/卷目录/文件名,R#"
```

READ命令也可以使用F#参数或B#参数。当用B#参数时,你必须从0开始计数。

(三)建立一个16M字节的文件

ProDOS可以存取如16M字节一样大的文件。事实上你可以在一个143K字节的磁盘上建立一个16M字节文件。

```

10 D$=CHR$(4)
20 PRINT D$;"OPEN BIG,L256";PRINT D$;"WRITE
BIG,R0"
25 PRINT 0
30 PRINT D$;"WRITE BIG,R65535"
35 PRINT "A"
40 PRINT D$;"WRITE BIG,R0"
50 PRINT 1
60 PRINT D$;"CLOSE"

```

这程序将在5英寸软磁盘上建立一个名为BIG的16M字节随机存取文件。如果你CATALOG此磁盘,则会看到如下显示:

```
/PRODOS
```

```

NAME TYPE BLOCKS ENDFILE SUBTYPE
BIG   TXT    5      16776962 R=256

```

尽管当前只有两个记录(R0和R65535),ProDOS却分配了5块磁盘空间给此文件。只有一块是用于存储数据,其它块全为零,如果你要打印其数据,ProDOS则读取此文件并打印零,直到读取R65535为止。

(四)STORE和RESTORE

STORE和RESTORE命令是新增加的ProDOS命令。STORE命令允许你将存储器内的所有变量保存起来。相反,RESTORE命令是将被STORE保存的所有变量装入存储器内。使用STORE命令的不利因素是所有变量和数组都被保存而不管重要与否。

ProDOS以压缩格式放置变量于存储它们之前。用STORE存储数百个变量要比用顺序或随机文件快得多,但要占用较多的磁盘空间。STORE将变量放入类型为VAR的文件内。在用RESTORE时,所有在内存的变量被清除,并将新的变量装入内存。

STORE和RESTORE可以在程序开始时读取数据,而在程序结束时保存数据,大致与顺序文件一样。你还可以用STORE和RESTORE增加数据。这过程需要程序将数据存入一个数组内。

```

5 REM Store/Restore
10 D$=CHR$(4)
20 DIM N$(100),C$(100),P$(100)
25 HOME
50 PRINE "1. Store, Write to Disk"
60 PRINE "2. Restore, Read from Disk"
65 PRINE "3. END"
70 GET A$;B=VAL(A$);IF B>0 AND B<4 THEN 80
75 GOTO 70
80 ON B GOTO 130,230,300
130 A=A+1
140 INPUT "Name:";N$(A)
150 INPUT "Company";C$(A)

```

汽车大赛

浙江湖州市新风小学 马宇昊

我在暑假里编了一个电脑游戏，取名“汽车大赛”。运行后，出现跑道，你驾驶的汽车在跑道下端。你必须用 J、L 和 K 键控制汽车左行、右行和停下，以逃避迎面奔来的汽车。随着道路变得越来越窄，难度也越来越高。当汽车被撞或驶出跑道，则游戏结束。屏障下端的数字是你的分数，如超过十万分，则说明你是高水平了！

```
10 C=17;D=35
20 GR: COLOR=1
30 W=10; B=20
40 COLOR=3;VLIN 0,39 AT 15
50 COLOR=2
60 VLIN 0,39 AT W: VLIN 0,39 AT B
70 COLOR=10; PLOT C,D
80 E=INT(RND(1)*19+1)
90 IF E=W OR E<W OR E=15 OR E=B OR E>B THEN
80
```

```
100 FOR F=0 TO 39
110 COLOR=15;PLOT E,F
120 P= PEEK(-16384)
130 IF P=204 THEN COLOR=0;PLOT C,D;C=C+1
140 IF C=15 THEN C=16
150 IF C=W THEN 300
160 IF C=B THEN 300
170 IF P=202 THEN COLOR=0; PLOT C,D;C=C-1
180 IF C=15 THEN C=14
190 IF C=W THEN 300
200 COLOR=0;PLOT E,F
210 COLOR=10;PLOT C,D;IF P=202 OR P=204 THEN
COLOR=0; PLOT C,D
220 V=V+100;VTAB 24; HTAB 14;PRINT V
230 IF F=35 THEN 260
240 IF F=39 THEN 280
250 NEXT F
260 IF C=E THEN 300
270 NEXT F
280 IF W=13 OR B=17 THEN GOTO 50
290 COLOR=0;VLIN 0,39 AT W;VLIN 0,39 AT B;W=W+
1;B=B-1; GOTO 50
300 TEXT ;HOME ;VTAB 11;HTAB 19;PRINT "End"
```

电视音像系统的佳音

——CZZ 彩色台标时码发生器

超大规模集成电路的采用使得本彩色台标时码发生器具有：

1. 多种调节功能；
2. 简便的操作方法；
3. 清晰稳定的彩色字符；
4. 成倍压缩的体积和重量；
5. 极高的性能价格比。

它成为电视台、闭路电视系统、音像出版系统、公用天线系统不可缺少的设备之一。它为用户随机的提供各种彩色台标和时码，从而保护您的节目源不受侵犯。

由于本机采用超大规模集成电路及单片机控制，故整机故障率几乎为零。本机售后一年内如发生质量故障，一律采取整机以坏换新的办法，以保障信誉。外地用户来京取货免费赠送3盘录像带。

(上接20页)

```
160 INPUT "Phone";P$(A)
170 INPUT "Enter another Name?";A$
180 IF A$="Y" OR A$="y" THEN HOME;GOTO 130
190 IF A$="N" OR A$="n" THEN 210
200 GOTO 170
210 PRINT D$;"STORE/DATA. DISK/EXAMPLE"
220 GOTO 50
230 PRINT D$;"RESTORE/DATA. DISK/EXAMPLE"
```

本机与国内同类产品比较(外型见封面)

| | CZZ— I | 国内同类产品 |
|------|--------------|--------|
| 显示功能 | 正反两种显示 | 只能正显示 |
| 台标时码 | 位置满屏任意移动 | 不可移动 |
| 彩色锁相 | 移定 | 爬行严重 |
| 时码误差 | 每日<1秒 | <10秒 |
| 集成度 | 大规模集成电路单片机控制 | 分立芯片组合 |

本机每台售价4800/元

还可设计几种台标图案供换用，每增加一种图案，收费500—800元，视图案复杂程度而定。

园恩寺电子实验厂

邮编100009

地址：北京市东城区后园恩寺甲20号

联系人：刘宜秋、王丽燕

电话：4016359 6853592

开户银行：北京工商银行王府井分理处

帐号：402052—60

北京站换乘104、108、113路车交道口下车

```
240 FOR I=1 TO A
250 PRINT N$(I);PRINT C$(I);PRINT P$(I)
260 NEXT
270 PRINT;INPUT"PRESS RETURN TO CONTINUE";A$
280 GOTO 25
300 END
```

(未完待续)

本刊不慎，遗失1121991003号记者证，声明作废。

第三章 6502 MPU 的寻址方式

南京大学大气科学系(210008) 朱国江

一、什么是寻址方式

在第一章中,我们曾指出控制计算机各部件协调动作的命令称为指令。每条指令均由两部分组成:一部分称为操作码,它指明该条指令的操作性质,执行什么样的操作(即指令功能);另一部分是操作数,实际上是操作的对象,它指明参与操作或运算的操作数据本身,或者操作数据所存放的单元地址,或者操作数据所在地址的地址。

数据和指令在存储器中存放的位置,称为地址。地址又分为两种:指令地址,即为存放指令的地址;操作数地址,为存放数据的地址,简称为操作地址。

简单地说,寻址方式就是寻找指令地址或操作数地址的方式。

二、6502 指令系统的特点

6502 微处理器共有 56 条基本指令,13 种寻址方式,其主要特点有以下几点:

- 虽然 6502 内部可供访问的寄存器较少,但它的指令系统包含零页寻址方式,可访问 \$0000—\$00FF256 个地址,零页寻址方式的指令占用字节少,执行时间短,对节约内存、缩短程序执行时间都是有利的。

- 6502 的指令长度为 1 至 3 个字节,第一字节一律是操作码,它决定 6502 完成一种运算或操作,而操作码后面跟着的是操作数或操作数地址,占用一个或二个字节,字节多少视寻址方式而定。

- 6502 和外设交换数据或信息采用存储器映象方式,凡是访问存储器的指令,都可以用于外设,这样访问外设的指令多,给使用带来极大的方便。

三、13 种寻址方式

1. 立即寻址(Immediate Addressing)

立即寻址方式,指的是操作对象部分直接给出了操作数本身,该操作数称为立即数。

例如,指令代码 A9 5A,是一条两字节指令,用汇编语言写出就是 LDA # \$5A,其中 A9 是操作码,表示取一个数装入累加器 A 中;累加器 A 中的值,就是 A9 后面的 5A,这是一个立即数而不是地址码。这条指令的功能是将立即数 \$5A 送累加器 A,即 \$5A→A。

说明:

- 立即寻址方式的指令,常用来设置程序所需用的各种固定初值。

- 立即寻址方式指令,长度均为两个字节。

- # 符号表示后面跟着的是立即数,\$ 是 16 进制数的表示符号。

- 操作数部分不是地址,而是操作数本身。

- 立即数的形式是 # \$nn,其中 nn 是两位 16 进制数,所以立即数不超过 8 位二进制数范围(\$FF)。

- 立即寻址方式指令中,用于取数至累加器 A 的有 LDA # \$nn;取数至变址寄存器的有 LDX # \$nn, LDY # \$nn;累加器 A 与立即数带进位加、减的有 ADC # \$nn, SBC # \$nn;与累加器 A 的值进行逻辑运算的有 AND # \$nn, EOR # \$nn, ORA # \$nn;与寄存器比较的有 CMP # \$nn, CPX # \$nn, CPY # \$nn 等共 11 条指令。

2. 绝对寻址(Absolute Addressing)

绝对寻址方式中,操作数部分是操作数在存储器中的实际地址(称为绝对地址,又称直接地址)。

例如,采用绝对寻址方式的取数指令 LDA \$4000,是把内存中地址为 \$4000 单元的内容装入累加器 A 中,即(\$4000)→A。这里 \$4000 表示绝对地址,(\$4000)表示 \$4000 单元存储的内容(以下表示法同)。

又如,指令:20 DA FD (JSR \$FDDA)通知 CPU 转入 \$FDDA 地址单元取指令执行。

注意 LDA # \$5A 和 LDA \$4000 两条指令的异同点,相同点是它们都是向累加器 A 中置数,所以它们的汇编符号都是 LDA(LOAD A 的缩写);但它们的操作码不同,前者是 A9,后者是 AD。操作码不同表示寻址方式不同,前者是立即数寻址,后者是绝对寻址。这个简单例子可以加深对寻址方式这一概念的理解。

说明:

- 绝对寻址方式的指令,其长度都是三个字节。

- 绝对寻址方式用两个字节存放操作数的地址,它们的寻址范围可以是整个 64K 存储器的任何一个地址,使用方便,缺点是每条指令占据三个字节。

- 凡用两个字节表示一个地址时,总是先给出低位字节,再给出高位字节,即高位在后,低位在前。绝对寻址方式指令表示成机器码时,就属这种情况。

- 绝对寻址指令有:内存单元取数至寄存器的 LDA \$nHnL, LDX \$nHnL, LDY \$nHnL;寄存器中的数存入内存的 STA \$nHnL, STX \$nHnL, STY \$nHnL;无条件转移 JMP \$nHnL;转移到子程序 JSR \$nHnL;存储器内容加、减 1 的 INC \$nHnL, DEC \$nHnL;累加器与存储器带进位加、减的 ADC \$nHnL, SBC \$nHnL;逻辑运算的 AND \$nHnL, EOR \$nHnL, ORA \$nHnL;检测存储器的 BIT \$nHnL;算术左、右移位的 ASL \$nHnL, LSR \$nHnL;循环左、右移位的 ROL \$nHnL, ROR \$nHnL;寄存器与存储器比较的 CMP \$nHnL, CPX \$nHnL, CPY \$nHnL 等共 23

条。

3. 零页寻址 (Zero—page Addressing)

零页寻址方式表达的操作数,实际上是一个操作数地址,并仅限于存储器的零页范围 0000—00FF 内。

例如,取数指令 LDA \$FB,它的意思是吧零页地址 FB 单元中的操作数送到累加器 A。这里的操作数是地址,而且只有一个字节 FB,表示它是零页地址 00FB。其指令的机器码为 A5 FB,功能为 (\$FB)→A。

指令 A5 FB (LDA \$FB)和指令 AD FB 00 (LDA \$00FB)在功能上是等同的。前者系零页寻址,后者则为绝对寻址,但前者比后者少用一个字节。

零页寻址和绝对寻址的操作对象部分都是操作数所在地址,不同的是前者只能在零页的 256 个单元中,而后者可以是整个 64K 存储单元中的任一个地址。

说明:

- 零页寻址均为二字节指令,写成机器码时第一个字节是操作码,第二个字节为操作数的零页地址。

- 零页地址中操作数地址仅限于零页范围 (0000—00FF),故在表示操作数地址码时,不必给出高 8 位地址 (约定为 \$00),因而指令长度可以减少一个字节。

- 零页寻址方式中的操作数表达式是 \$nn,其中 nn 代表两位十六进制数。

- 零页寻址指令占用字节少,执行速度快,在节省内存和缩短时间上,为程序编制提供了方便。

- 零页寻址方式的指令有 20 条,它们是:内存单元取数至寄存器 LDA \$nn,LDX \$nn,LDY \$nn;寄存器中数存入内存 STA \$nn,STX \$nn,STY \$nn;算术运算指令 ADC \$nn,SBC \$nn;增减指令 DEC \$nn,INC \$nn;逻辑运算指令 AND \$nn,BIT \$nn,EOR \$nn,ORA \$nn;移位指令 ASL \$nn,LSR \$nn,ROL \$nn,ROR \$nn;比较指令 CMP \$nn,CPX \$nn,CPY \$nn。

4. 隐含寻址 (Implied Addressing)

所谓隐含寻址,指的是地址已为操作码隐含,不需再指出,即操作数所在地址隐含于操作码之中。

例如,指令 CLC,其机器码为 18,功能是置标志寄存器中的 C 标志位为 0 (清进位标志)。清除进位位显然是对 CPU 中的状态寄存器进位位 C 进行操作,因为操作码 (\$18)本身已包含了操作数的内容,所以,完全没有必要在指令中再用一个字节来表示操作数。

又如,DEX 指令,它执行变址寄存器 X 中的内容减 1 操作,然后再存入 X 中,即 $X-1 \rightarrow X$,操作码为 \$CA。INX 指令则执行变址寄存器 X 加 1 操作,即 $X+1 \rightarrow X$,操作码为 \$E8。这里 \$CA 和 \$E8 的操作码中,分别隐含了对变址寄存器 X 进行操作 (减或加)的信息,同样不需要选用另外字节来存储操作数。

说明:

- 隐含寻址方式的指令,均为单字节指令,因操作数地址隐含在操作码中。

- 隐含寻址的操作数是在寄存器 X、Y、S 及 P 中的各标志位。

- 隐含寻址的指令有:进出栈操作 PLA,PHA,PLP,PHP;各寄存器之间数据传送 TAX,TXA,TAY,TYA,TSX,TXS;中断返回 RTI;子程序返回 RTS;设置进位标志 SEC;设置十进制运算方式 SED;设置中断禁止状态 SEI;清除进位标志 CLC;清除十进制运算标志 CLD;清除中断禁止状态 CLI;清除溢出标志 CLV;强迫中断 BRK;空操作 NOP;变址寄存器加、减操作 DEX,INX,DEY,INY 等共 25 条。

5. 累加器寻址 (Accumulator Addressing)

累加器寻址方式指的是:操作数在累加器 A 中。

例如,循环左移指令 ROL A,其功能为:将 A 中的操作数的各位连同进位 C 在内循环左移一位,最高位移到标志位 C 中。机器码为 2A,助记符为 ROL。

说明:

- 累加器寻址皆为单字节指令。

- 由于操作仅涉及累加器 A,因此,操作数地址不需要单独表示,只要隐含在操作码字节中即可。从这个角度上说,累加器寻址也是一种隐含寻址。

- 累加器寻址主要用于移位,指令只有 4 条,即:ASL A,LSR A,ROL A,ROR A。

6. 绝对 X 变址 (Absolute X—Indexed Addressing)

这是使用 X 寄存器进行变址的一种寻址方式。操作数地址是由绝对地址加上 X 寄存器中的值构成。

例如,指令:BD 00 40 (LDA \$4000,X)是一条取数指令,其操作码为 BD,操作数存放在一个内存单元,其地址由指令的第二字节和第三字节指示的地址 (本例是 \$4000)加上变址寄存器 X 的内容得到。

我们将指令给出的 16 位的绝对地址称为基地址,而寄存器 X 中的内容称为偏移量,则:基地址+偏移量 (X)=操作数实际存放的有效地址。

因此,上述指令的功能就是把 4000+X 单元中的值取出来送到累加器 A 中。如果 X 变址寄存器中的值是 \$20,那么这条指令就是把 \$4020 中的数送入累加器 A 中。

又如取数指令 STA \$0330,X,其功能是把累加器 A 中的内容存放到存储器 \$0330+X 单元中去,即 $A \rightarrow (\$0330+X)$ 。

说明:

- 绝对 X 变址方式的指令皆为三字节指令。

- 同一条绝对 X 变址指令,因 X 中的值不同,而改变了操作数的地址。这就是把 X 寄存器称为变址器的原因。由于基地址加 X 偏移量才是操作数的有效地址,所以,有效地址是经过上述“变址”过程得来的。

- 绝对 X 变址指令共有 15 条,其中取数指令两条: LDA \$nHnL,X, LDY \$nHnL,X; 存数指令一条: STA \$nHnL,X; 算术逻辑运算指令 12 条: ADC \$nHnL,X, DEC \$nHnL,X, INC \$nHnL,X, SBC \$nHnL,X, AND \$nHnL,X, EOR \$nHnL,X, ORA \$nHnL,X, ASL \$nHnL,X, LSR \$nHnL,X, ROL \$nHnL,X, ROR \$nHnL,X, CMP \$nHnL,X。

7. 绝对 Y 变址 (Absolute Y—Indexed Addressing)

它是使用 Y 寄存器进行变址的一种寻址方式。这种寻址方式把指令给出的 16 位地址作为基地址与偏移量(Y 变址寄存器中的值)相加后所得的地址,作为操作数实际存放的有效地址。

例如,传送指令 LDX \$FE30,Y(设 Y 寄存器中的值为 \$20),其功能是把 \$FE30+Y,即 \$FE50 单元中的内容取出来送入寄存器 X 中,即(\$FE30+Y)→X。

说明:

- 绝对 Y 变址方式的指令皆为三个字节。
- 绝对 Y 变址和绝对 X 变址十分相似,其区别前者使用的是 Y 变址器,后者使用的 X 变址器。
- 绝对变址寻址(X 或 Y)与零页变址寻址(X 或 Y)很类似,其区别仅在于零页变址中只给出一个 8 位地址(低 8 位),其高 8 位约定为 \$00。
- 绝对 Y 变址的指令有: LDA \$nHnL, Y, LDX \$nHnL, Y, STA \$nHnL, Y, ADC \$nHnL, Y, SBC \$nHnL, Y, AND \$nHnL, Y, EOR \$nHnL, Y, ORA \$nHnL, Y, CMP \$nHnL, Y 等共 9 条。

8. 零页 X 变址(Zero-page X-Indexed Addressing)

使用 X 变址寄存器的零页变址,简称零页 X 变址。将零页地址作为基地址与 X 寄存器的内容相加,即得操作数存放的实际零页地址。

例如,取数指令 LDA \$A3,X,其功能是把存储器 \$00A3+X 单元中的内容取出,送累加器 A。

又如, LDA \$3F,X 与 LDA \$003F,X,这两条指令表达的含意相同,功能一样。但使用上有点区别:前者操作码为 B5,后者操作码为 BD;前者为零页 X 变址,后者为绝对 X 变址;前者为两个字节,后者为 3 个字节。

说明:

- 零页 X 变址指令皆为两个字节长度。
- 绝对变址寻址(X 或 Y)和零页变址寻址(X 或 Y)含义相近,但寻址方式不同,前者寻址范围广,可以是任意地址;后者仅限于零页地址。
- 零页 X 变址只能产生零页地址,例如,指令 LDA \$A3,X,若 X=\$62,则操作数的有效地址不是 \$105,而是 \$05。这一点在使用时应特别注意。
- 零页 X 变址寻址指令有 16 条: LDA \$nn,X, LDY \$nn,X, STA \$nn,X, STY \$nn,X, ADC \$nn,X, DEC \$nn,X, INC \$nn,X, SBC \$nn,X, AND \$nn,X, EOR \$nn,X, ORA \$nn,X, ASL \$nn,X, LSR \$nn,X, ROL \$nn,X, ROR \$nn,X, CMP \$nn,X。

9. 零页 Y 变址(Zero-page Y-Indexed Addressing)

使用 Y 变址寄存器的零页变址,简称零页 Y 变址。它与零页 X 变址类似,只是变址器值取自 Y 的当前值。

例如,指令 STX \$5F,Y,表示将一个数送至变址寄存器 X,而这个数存放在变址器 Y 的值加上 \$5F 之后所指示的零页存储单元中。

说明:

- 零页 Y 变址指令是二字节的。

• 采用零页 Y 变址的指令只有两条: LDX \$nn,Y 和 STX \$nn,Y。

• 零页 Y 变址的其它意义同零页 X 变址有关说明。

10. 间接寻址(Indirect Addressing)

间接寻址指令的操作对象部分给出的是操作数所在地址的地址(称为间接地址)。这意味着,在操作码后面的两个字节并不是有效地址,而是存放有效地址低 8 位的存储单元(间接地址)。亦即操作数的有效地址必须通过两次寻址才能找到。

例如,无条件转移指令 JMP(\$2000),首先从指令指出的间接地址 \$2000 单元取出操作数有效地址的低 8 位(设为 \$5F),从间接地址加 1 单元(即 \$2001 单元)取出有效地址的高 8 位(设为 \$4D),则 \$4D5F 即为操作数的有效地址;然后将 \$4D5F 送程序计数器 PC,CPU 随即转向 PC 所指示的单元(\$4D5F)去执行指令。

注意 JMP \$2000 和 JMP(\$2000)这两条指令的差别:两者均为无条件转移指令,但操作码不同,前者是 4C,后者是 6C,因而寻址方式各异。前者是绝对寻址,有点像 BASIC 中的 GOTO 2000,后者是间接寻址,像 BASIC 中的 GOTO 2000,而 2000 行语句是 GOTO 4D5F。当然这是形象化的比喻,GOTO 语句后面的行号,相当于我们举例中的内存地址。

间接寻址都是 3 字节指令,而且仅有 1 条: JMP(\$nHnL)。

11. 相对寻址(Relative Addressing)

相对寻址的指令格式是:首字节为操作码,次字节为相对转移地址,又称相对位移量,或偏移量 D,它实际上是条件转移指令的跳转步长。D 值可正可负,符号由最高位指明(1 为负,0 为正)。操作数的实际地址是由取出本指令后,下一条指令的操作码所在的地址加偏移量 D 来得到。即取出本指令后的 PC 值加 D。

为了弄清相对寻址的概念,我们不妨先看一个机器语言程序实例:

```
0300—AD 00 20    LDA $2000
0303—D0 05       BNE $030A
0305—A9 20       LDA # $20
0307—8D 00 40    STA $4000
030A—00         BRK
```

第一条指令采用绝对寻址方式,把 \$2000 单元的内容送累加器 A 中。第三条指令采用立即寻址,把立即数 \$20 送累加器 A。它们都是取数指令。第四条指令采用绝对寻址方式,把累加器 A 的内容送入 \$4000 单元,是存数指令。最后一条指令是单字节指令,显然是隐含寻址方式,产生中断,它把标志位寄存器 P 中的 B 标志位(第 4 位)置 1。第二条指令 BNE,是一条条件转移指令,条件是非 0,其条件成立与否取决于上一条指令执行的结果。当第一条指令从 \$2000 单元取出的数为 0 时(Z=1),条件不满足,不产生转移动作,程序接着执行第三条指令;而当从 \$2000 取出一个非 0 值

(Z=0)时,条件满足,程序转移到第五条指令。

如何判别这种转移?转移地址又是如何计算出来?这就要分析指令计数器 PC 的作用。

上述程序执行时,PC 中首先置入的是首地址 \$0300,于是计算机依 PC 的值执行 \$0300 中的指令,同时将该指令的长度(字节数 3)与 PC 的值相加(\$0300+\$03),再送入 PC,此时 PC 值为 \$0303,正好是第二条指令的首地址。第一条指令执行完后,依 PC 中的值去执行 \$0303 单元开始的第二条指令,同时把第二条指令长度(二字节)加入 PC 中,使 PC 变成 \$0305。第二条指令要求检查第一条指令执行后的 Z 标志位状态(\$2000 中的数为 0 时,Z=1;\$2000 中的数非 0 时,Z=0)。当 Z=1 时,程序不产生转移,按正常顺序执行下一条指令(即 \$0305 开始的第三条指令);当 Z=0 时,非 0 条件满足,程序产生转移,于是将 PC 中的值(\$0305)加上偏移量 D(本例是 \$05)置入 PC 中,变成 \$030A。因此,执行 \$0303 的第二条指令后,直接转向 \$030A 执行第五条指令。

由此可知,条件转移指令 BNE \$030A 的意思是根据上一条指令执行后零标志位 Z 的状态,决定程序是否分支,若 Z=0,则分支转向 \$030A 去执行;反之不产生分支,执行 \$0305 开始的指令。

对本例来说,若产生分支转移,则偏移量 D(即跳转步长)应从 \$0305(本指令 BNE 取出后的 PC 值)算起,到 \$030A 为止,即 $D = \$030A - (PC + 2) = \$030A - (\$0303 + \$02) = \$05$ 。或者由 $\$0303 + \$02 + D = \$030A$ 中算出 D。后一种方法物理意义明确且容易记忆,即:

转移指令的首地址+2(转移指令都是两字节指令)+D(偏移量)=目标地址(操作数的实际地址)

偏移量 D 也可能取负值,请看下列实例:

```
0300—CE 00 20 DEC $2000
0303—AD 00 20 LDA $2000
0306—D0 F8 BNE $0300
0308—A9 20 LDA # $20
030A—8D 00 40 STA $4000
030D—60 RTS
```

本程序,条件转移指令 BNE 的首字节(操作码 D0)存放在 \$0306 单元,偏移量 D 存放在 \$0307 单元。该指令的意思是若 Z=1(非 0 条件不满足),则执行 \$0308 单元处的指令;若 Z=0(非 0 条件满足)跳转至 \$0300 处执行。显然这里偏移量 D 是负值,由上述计算方法有:

$$\$0306 + \$02 + D = \$0300$$

$$\therefore D = \$0300 - \$0308 = -\$08$$

D 为负值,应用补码表示,即 F8。(注:负数的补码可以通过对原码求反加 1 得到)。

说明:

- 相对寻址指令皆为两字节。
- 相对寻址只适用于条件转移指令。
- 操作数部分是条件转移指令的跳转步长。
- 操作数的实际地址是取出本指令后 PC 值(本

指令首地址+2)与偏移量之和。

• 偏移量为负时,要用补码表示。

• 为减少实际编程时的麻烦,书写相对寻址指令的助记符时,用转移后的目标地址代替偏移量。

• 相对寻址指令有: BCC、BCS、BEQ、BMI、BNE、BPL、BVC、BVS 等 8 条。

12. 先变址(X)间接寻址((IND,X))

这种寻址方式实际上是零页 X 变址和间接寻址两种方式的结合。首先,它对零页基地址 IND 进行 X 变址,求得零页地址 $IND + X$;再以 $IND + X$ 作为间接地址,经两次间接寻址找到操作数的有效地址;第一次对 $IND + X$ 间址得到操作数有效地址的低 8 位;第二次对 $IND + X + 1$ 间址得到操作数有效地址的高 8 位。

例如,取数指令 LDA(\$06,X),其机器码为 A1 06。若已知变址寄存器 X 中的值为 02,存储单元(08)=65,(09)=87,(8765)=FF,零页基地址 IND=06,则根据上述定义,先计算 $(IND + X) = (06 + 02) = (08) = 65$,这是该指令操作数第一次间址,得到其有效地址的低 8 位;第二次间址 $(IND + X + 1) = (06 + 02 + 1) = (09) = 87$,得到有效地址的高 8 位,所以操作数实际地址是 \$8765。而本指令的功能则是把 \$8765 地址单元中的数 FF 取出来送到累加器 A 中。本例可以归结为下述程式:

$$LDA(\$06,X) \Rightarrow (06 + X) = (08) = 65 \Rightarrow PC_L \Rightarrow (06 + X + 1) = (09) = 87 \Rightarrow PC_H \Rightarrow PC \Rightarrow 8765 \Rightarrow (8765) = FF \Rightarrow A。$$

说明:

• 先变址(X)间接寻址指令皆为两字节指令。

• 变址寄存器只能用 X。

• 先做 X 零页变址,再两次间接寻址,才能得到操作数的有效地址。

• 指令有:取送数 LDA(\$nn,X)、STA(\$nn,X)、算术运算 ADC(\$nn,X)、SBC(\$nn,X)、逻辑运算 AND(\$nn,X)、EOR(\$nn,X)、ORA(\$nn,X)及比较运算 CMP(\$nn,X)等 8 条。

13. 后变址(Y)间接寻址((IND),Y)

后变址(Y)间接寻址,是先间接寻址,后 Y 变址。它是间接寻址和绝对 Y 变址相结合的一种寻址方式。其步骤是先在零页间接寻址,取得一个 16 位基地址,然后再进行绝对 Y 变址寻址,即以基地址加 Y 变址寄存器的内容得到操作数实际存放的有效地址。所以,这种寻址方式应称为先间接寻址后绝对 Y 变址。

例如,取数指令 LDA(\$06),Y,机器码为 B1 06,设(06)=65,(07)=87,(Y)=02,(8765)=EE。第一步间接寻址,因(06)=65,(07)=87,所以间接寻址后的地址为 \$8765;第二步绝对 Y 变址,因(Y)=02,那么 Y 变址后为 \$8767;第三步将内存单元 \$8767 中的数 EE 取出来送累加器 A。

说明:

• 后变址(Y)间接寻址为两字节指令。

• 变址寄存器只限于 Y 寄存器。

(下转第 31 页)

数据结构

北京电脑天地学校(100051) 宋丹颖

数据结构自测题

一、从供选择的答案中选出同下列叙述关系最密切的字句,把编号写在题目的对应题号旁。

- A. 说明数据元素之间的顺序关系,抽象地反映数据元素的结构。
- B. 各数据元素之间存在着线性关系,具有均匀性和有序性。
- C. 在进行插入或删除操作时,只需改变结点的指针,不必移动表中别的元素。
- D. 经常用于表达式的转换和求值、处理子程序调用等的数据结构。
- E. 通常采用顺序存储结构,每个数据元素由一个值和一组下标组成。
- F. 由一个或多个数据项组成,各数据项的数据类型可不同。
- G. 求出字符串中含有字符个数的操作。
- H. 求出字符串中从指定位置开始且长度为一定值的串。

供选择的答案:

- ①数据结构 ②数据的存储结构 ③数据的逻辑结构
- ④栈 ⑤队列 ⑥线性表
- ⑦链表 ⑧记录 ⑨数据类型
- ⑩数据对象 ⑪数组 ⑫联接
- ⑬求长度 ⑭定位 ⑮求子串

二、从供选择的答案中选出正确答案,将其编号填写在叙述中的横线上。

1. 栈和队列是两种基本的数据结构,栈是具有_____特点的特殊线性表,而队列是具有_____特点的特殊线性表。

供选择的答案:

- A. 只进不出 B. 先进先出
- C. 后进先出 D. 随机进出

2. 使用双向链表存储数据,其优点是可以_____。

供选择的答案:

- A. 提高检索速度 B. 很方便地插入和删除数据
- C. 节约存储空间 D. 很快回收存储空间

3. 设有一个栈,元素进栈的次序为1,2,3,4,5。能得到下列出栈序列中的_____。

供选择的答案:

- A. 1,2,3,4,5 B. 2,3,4,5,1
- C. 5,1,2,3,4 D. 5,4,2,3,1

4. 排序是数据处理中重要的运算,它的功能是将一个数据元素的无序序列按某种指定的顺序调整成为一个

有序序列。各种排序方法可按不同的原则进行分类。冒泡排序和快速排序属于_____。

供选择的答案:

- A. 插入排序 B. 选择排序
- C. 交换排序 D. 归并排序

三、从供选择的答案中选出应填入下列叙述中_____内的正确答案。

1. 在作进栈运算时,应先判别栈是否[A];在作退栈运算时,应先判别栈是否[B]。当栈中元素为n个,作进栈运算时发生上溢,则说明该栈的最大容量为[C]。

为了增加内存空间利用率和减少溢出的可能性,由两个栈共享一片连续的内存空间时,应将两栈的[D]分别设在这片内存空间的两端,这样,当[E]时,才产生上溢。

供选择的答案:

- A, B: ①空 ②满 ③上溢 ④下溢
- C: ①n-1 ②n ③n+1 ④n/2
- D: ①长度 ②深度 ③栈顶 ④栈底

E: ①两个栈的栈顶同时到达栈空间的中心点

②其中一个栈的栈顶到达栈空间的中心点

③两个栈的栈顶在栈空间的某一位置相遇

④两个栈均不空,且一个栈的栈顶到达另一个栈的栈底

2. 线性表中的[A]的主要优点是从表中任一结点出发都能访问到所有结点。使用[B],可根据需要在前后两个方向上方便地移动,而使用[C]则只能顺链循一个方向移动。若有几个线性表同时并存,且在处理过程中备表的长度动态地变化,元素的总数也动态地改变,这种情况下应选用[D]存储结构。供选择的答案:

- A, B, C: ①单向链表 ②可利用空间表 ③双向链表
- ④循环链表

D: ①顺序 ②链接

3. 对大小为n的队列,随着元素的进队和出队,其首、尾指针将随之发生变化。当队列的队首指针值等于队尾指针值时,则表示队列处于[A]状态;当队尾指针=_____时,表示队列处于[B]状态。队列是限定在[C]进行操作的具有[D]性质的特殊线性表。

供选择的答案:

- A, B: ①进队 ②出队 ③队满 ④队空
- C, D: ①一端 ②两端 ③先进先出 ④随机进出

4. 要进行顺序查找, 则线性表[A]。要进行二分法查找, 则线性表[B]。若表中元素个数为 n , 则顺序查找的平均比较次数为[C]。

供选择的答案:

A, B: ①必须以顺序方式存储

②必须以链接方式存储

③必须以顺序方式存储, 且数据元素已按值递增或递减顺序排好

④必须以链接方式存储, 且数据元素已按值递增或递减顺序排好

⑤既可以以顺序方式存储, 也可以以链接方式存储

C: ① n

② $n/2$

③ n^2

5. 有一个二维数组 A, 行下标的范围是 1 到 9, 列下标的范围是 1 到 5, 每个数组元素用相邻的 4 个字节存储。存储器按字节编址。假设存储数组元素 A[1, 1] 的第一个字节的地址是 0。存储数组 A 的最后一个元素的第一个字节的地址是[A]。若按行存储, 则 A[4, 5] 和 A[6, 3] 的第一个字节的地址分别是[B]和[C]。若按列存储, 则 A[8, 1] 和 A[3, 4] 的第一个字节的地址分别是[D]和[E]。

供选择的答案:

A—E: ①28 ②44 ③76 ④92 ⑤108

⑥116 ⑦132 ⑧176 ⑨184 ⑩188

四、从下列叙述中选出正确的叙述, 把编号写在横线内。

1. 顺序存储方式只能用于存储线性结构。

2. 顺序存储方式的优点是存储密度大, 且插入、删除运算效率高。

3. 链表的每个结点中都恰好包含一个指针。

4. 栈和队列的存储方式既可以是顺序方式, 也可以是链接方式。

5. 数组和记录都是只能存储相同类型数据元素序列的数据结构。

6. 栈和队列都是一端固定, 只能在另一端进行存取操作的特殊的线性表。

7. 由于数组中的数据元素的存储位置是下标的线性函数, 所以可对其中的元素进行随机访问。

8. 采用紧缩格式存储字符串, 可以节省存储空间, 但在进行串的运算时需要多花费时间去分离同一个字中的字符。

9. 对以顺序方式存储的栈和队列, 必须预先分配存储空间。而对于以链表形式存储的栈和队列, 则无须预先分配存储空间。

10. 串在存储结构与具体计算机的编址方式没有关系。

自测题分析及参考答案

一、分析

A. 本叙述是指数据的逻辑结构。

B. 线性表中各数据元素之间存在着线性关系。具有均匀性: 各元素具有同样的数据类型; 有序性: 各元素是有序的, 不可交换次序。

C. 线性表有两种存储结构: 采用顺序存储结构的线性表叫“顺序表”。采用链接存储结构的线性表叫“链表”。这种存储方式不必为其分配一块连续的存储空间。每个元素的存储单元叫结点, 每个结点由两部分组成: 数据项和指向下一个数据项的指针, 数据元素之间的关系是用指针来体现的。无论插入还是删除操作, 只要改变结点的指针即可, 无需移动别的元素。

D. 这种数据结构是栈。它的重要特性是“后进先出”。因此, 只要处理数据的顺序与这些数据的输入顺序不同时, 栈都是很有用的。子程序调用时, 由于返回的次序和调用的次序相反, 所以, 一般是设置一个返回地址栈。高级语言编译中的表达式转换和求值也要用到栈。设操作数栈和运算符栈两个栈, 编译程序自左至右扫描表达式, 凡遇操作数一律进栈, 若遇运算符则判其优先数是否大于运算符栈顶元素的优先数。若大, 则进栈; 反之, 则栈顶运算符退栈, 并形成一条只有一次运算的机器指令, 指令中的运算对象为操作数栈顶上的两个元素。

E. 数组是线性表的推广, 作为一种数据结构, 在几乎所有的程序设计语言中都允许使用。它通常采用顺序存储结构。一维数组中的每个数据元素由一个值和一个下标组成; 二维数组中的每个数据元素由一个值和一对下标组成; n 维数组中的每个数据元素由一个值和一组下标组成。

F. 数组是相同类型数据元素的序列; 而记录是各种不同类型成分组合起来构成的数据结构。记录是可作为一个个单位来处理的一组相连的数据, 每个记录由一个或多个数据项组成, 各数据项的数据类型可不同。

G. 字符串是一种字符型的数据结构, 有一系列串的运算。求出字符串中含有字符个数的操作称为求长度, 是串的基本运算之一。很多程序设计语言都提供了求串长度的函数。

H. 简单地说, 字符串就是一个字符序列。而串中任意个连续字符组成的字符子序列称为子串。求串中从指定位置开始且长度为一定值的子串的操作称为求子串运算。

答案: A③ B⑥ C⑦ D④
E① F⑧ G⑬ H⑮

二、分析

1. 栈是一种所有的插入或删除只能在表的一端进行的特殊的线性表。由于只能在一端出入, 所以先进栈的数据“压在下层”, 只能后出来, 而最后进栈的数据元素最先出栈, 它具有“后进先出”的特点。队列是在表的两端进行操作的特殊线性表, 专门用于插入元素的一端叫队尾, 用于删除元素的一端叫队首。它类似于日常生活中的购物排队。先来的人排在前面, 后到的人顺次排在队尾。先来的人买到货物先离开队伍, 是一种“先进先出”的结构。

2. 链表可分为单向链表和双向链表。单向链表是用一个指针循一个方向连接的链表,当给定一个表元素时,只能顺链的方向移动查找,影响存取速度。而双向链表有左、右两个链指针,一个接前趋元素,一个接后继元素。可以在前后两个方向上移动,提高检索速度。

3. 已知元素的进栈次序为 1,2,3,4,5,根据“后进先出”的特性,它只能得到出栈序列中的 B 组答案。操作步骤是:

- | | |
|-----------|-----------|
| (1)push 1 | (6)push 4 |
| (2)push 2 | (7)pop 4 |
| (3)pop 2 | (8)push 5 |
| (4)push 3 | (9)pop 5 |
| (5)pop 3 | (10)pop 1 |

出栈序列为 2,3,4,5,1。

4. 排序方法可按不同的原则进行分类。整个排序过程都在内存进行的排序称为内部排序,常用的内排序法有:插入排序、交换排序、选择排序、归并排序和基数排序等。冒泡排序的基本思想是:每次将待排序序列中的数据项两两比较,凡是逆序则进行交换。将序列照此方法从头至尾处理一遍称为一次冒泡,关键字值小的记录好比水中气泡逐次往上飘浮,而关键字值大的记录好比石块沉入水底,如此反复比较交换,直到排好为止。快速排序是对冒泡排序的改进。它的基本思想是通过一次排序将文件分成两部分,然后分别对这两部分进行排序以达到最后整个文件有序。这两种排序方法都是基于比较交换,因此属于交换排序。

答案:1. C, B 2. A 3. B 4. C

三、分析

1. 栈是一种“后进先出”的数据结构。栈底固定而栈顶浮动,要设一个栈指针 top 指出栈的栈顶元素位置。栈有进栈、退栈、判满、判空四种基本运算。在作退栈运算时,应先判别栈是否为空,若已没有数据元素,再作退栈运算就要发生“下溢”。在作进栈运算时,应先判别栈是否已满,当 $top = \text{Maxsize}$ 时,表示分配给栈的空间已满,这时再有元素进栈就要发生“上溢”。若栈中元素个数为 n 个,此时再作进栈运算则发生上溢,说明该栈的最大容量就是 n 。

对于以顺序方式存储的栈,常采用一块连续的存储区域。在有多个栈的情况下,会遇到两个问题:一是每个栈所需空间大小很难估计;二是每个栈的容量在使用中都是动态变化的。在设置栈时要注意解决其中一个栈发生上溢,而其余各栈尚留有很多空间的问题。当有两个栈共享一片连续的内存空间时,应将两个栈的栈底分别设在这片内存空间的两端,然后,各自向中间延伸。这样,只有当两个栈的栈顶在栈空间的某一位置相遇时才产生上溢。两个栈互补余缺,是一种经常被采用的、比较合理的方案。

2. 线性表中的循环链表,是使单向链表最后一个元素的指针又指向第一个元素构成的。因为是循环结构,所以从表中任一结点出发都能访问到所有结点。使用双向链表,可在前后两个方向上移动。单向链表则只能顺

链沿一个方向移动。顺序存储结构不便插入、删除;对于经常要动态变化的表,应采用链接存储结构。

3. 队列设有队首、队尾两个指针。对大小为 n 的队列,当队首指针值等于队尾指针值时,表明队列为空;当队尾指针 $= n$ 时,表明队列已满。队列是一种限定在表的两端进行操作的具有“先进先出”性质的特殊线性表。

4. 查找就是根据给定的某个值,在表中确定一个关键字等于给定值的记录或数据元素。顺序查找的方法是:用待查的关键字值与线性表中各数据元素逐个进行比较,若找出相等值的数据元素则查找成功,若找遍整个线性表都没有值相等的数据元素,则查找失败。顺序查找的优点是对线性表中数据元素的次序无任何要求,不必按关键字值排序;对线性表的存储结构无特殊要求,顺序存储、链接存储均可。若表中有 n 个元素,则比较次数最少一次,最多 n 次,平均比较次数为 $n/2$ 次。二分法查找又称为折半查找,是一种效率较高的线性表查找方法,它不仅基于关键字值查找,而且要求线性表中数据元素按关键字值排序,并且以顺序方式存储该线性表。

5. 数组通常采用顺序存储结构,它是在内存中连续存放的一组数据。这样就有一个按什么次序把数组元素排在线性序列里的问题。常用的排列次序有行优先顺序和列优先顺序。二维数组行主序下元素排列顺序为:

$a_{11}, a_{12}, \dots, a_{1n}, a_{21}, a_{22}, \dots, a_{2n}, \dots, a_{m1}, a_{m2}, \dots, a_{mn}$

列主序下元素的排列次序为: $a_{11}, a_{21}, \dots, a_{m1}, a_{12}, a_{22}, \dots, a_{m2}, \dots, a_{1n}, a_{2n}, \dots, a_{mn}$

二维数组元素的地址计算公式为:

行主序: $LOC(a_{ij}) = LOC(a_{11}) + ((i-1) * n + (j-1)) * L$

列主序: $LOC(a_{ij}) = LOC(a_{11}) + ((j-1) * m + (i-1)) * L$

其中 a_{11} 为第一个元素存放的起始地址, n 和 m 分别为数组每行和每列的元素个数, L 为每个元素占用的存储单元个数。

根据条件可知:二维数组 A 行下标范围是 1 到 9,列下标范围是 1 到 5,即 A 数组共有 9 行 5 列 45 个元素,每个元素用 4 个字节存储,因此可计算出存储数组 A 共需要 180 个字节,最后一个元素的第一个字节的地址是 176。

行主序下,元素 A[4,5]的第一个字节的地址计算方法是:

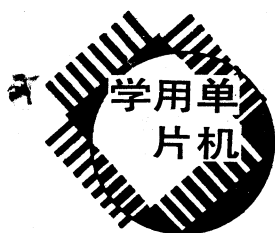
$$\begin{aligned} LOC(A[4,5]) &= LOC(A[1,1]) + ((4-1) * 5 + (5-1)) * 4 \\ &= 0 + (3 * 5 + 4) * 4 = 76 \end{aligned}$$

同理,可计算得 A[6,3]的第一个字节的地址是 108。列主序下, A[8,1]的第一个字节的地址是:

$$\begin{aligned} LOC(A[8,1]) &= LOC(A[1,1]) + ((1-1) * 9 + (8-1)) * 4 \\ &= 0 + (0 + 7) * 4 = 28 \end{aligned}$$

A[2,4]的第一个字节的地址为 116。

(下转第 31 页)



单片机最小应用系统与液晶显示器(LCD)的接口

合肥中国科技大学计算中心(230026) 张培仁

液晶显示器是一种极低功耗显示器件,在袖珍式仪表式低功耗应用系统中广泛使用。

液晶显示器是一种被动式显示器件。它本身不发光而只是调制环境光。它与其他显示器相比有如下特点:

(1)低电压(3V—6V),微功耗($0.3\mu\text{W}$ — $100\mu\text{W}$)。它适于与CMOS电路直接相配,可用于各种自动化检测设备中,尤其适合于电池供电的移动式仪器、仪表、手表、计算器中。

(2)不怕亮光,可在室内外明亮环境下使用,显示清晰度不会随环境亮度增加而减弱,在太阳光下也能清晰显示。

(3)体积小,外形薄为平板型显示。手表液晶显示器厚度1.2~1.6mm,计算器显示器厚度1.6~2.2mm,仪器厚度为2.2~5mm,引出线用导电橡胶引出,使用方便。

(4)显示面积、字形、大小在一定范围内不受限制。

1. LCD 基本的工作原理:

LCD是由两层玻璃之间封入一些液晶材料,当光线射入时,产生偏振光,这些光被液晶材料旋转 90° ,光线被反射回去,呈透明状态;当上下电极加一定电压,液晶分子转成垂直排列,失去旋光性,入射光无法通过下偏振片返回,因而呈黑色。

2. LCD 的驱动方式:

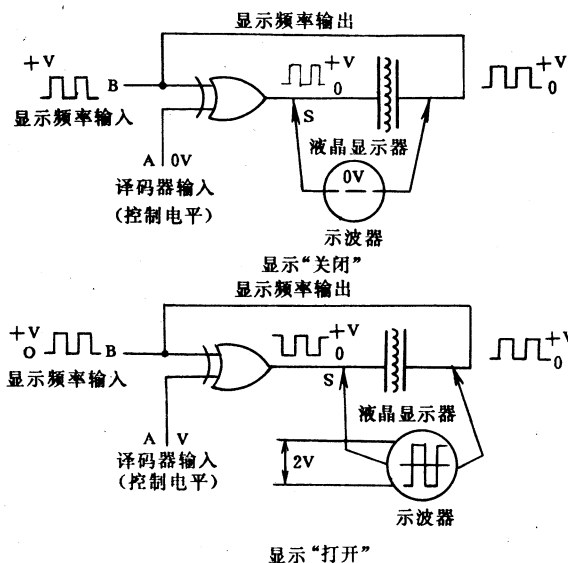


图 1

液晶显示器的驱动方式由电极引线的选择方式确定。当液晶显示器选定后,驱动方式也就确定下来。一般有静态驱动和动态驱动两种。不论哪一种方式驱动都是用方波(交流)来驱动。动态驱动只是分时驱动,也就是每一段片分时地被驱动。静态驱动,广泛用单一寻址由异或门构成的移相电路构成,见图1。

显示“关闭”,A端为“0”电平,S端与B端同相位,这时显示器两端相对电平差“0”V,液晶的片段不显示。显示“打开”,A端的为“1”电平,S端与B端反相位,这时显示器两端构成交流电压,片段显示。这里要说明的是每一段都要有一个异或门来驱动。

液晶显示器交流驱动波形如图2。

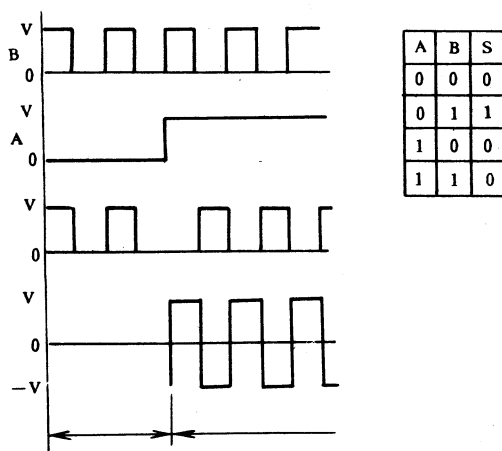


图 2

由于每段都要有一个异或门来驱动,单片机最小应用系统要驱动6个八段液晶显示器,光用8031的P1,P3口是不够,必须扩展2片8255芯片才能胜任。

这样显然用了较多的器件。为了减少器件,也可以用6片4095(小规模芯片)用串行口输出要显示的数据,稳定一段时间后如0.1秒,再从串行口输出上次输出的数据的反码,这样循环下去,同时可以得到稳定的显示,并且接口非常简单,所用芯片比前者大大地减少。因为这时节约2片8255和48个异或门。

3. 单片机和液晶显示器接口,驱动时要注意的问题

首先要保证加到液晶显示器两端交流电压平均值等于零,要注意直流分量越小越好,至少要小于100mV。否则,过大的直流电压使液晶材料迅速分解,大大缩短了显示器的工作寿命(一般直流寿命只有几

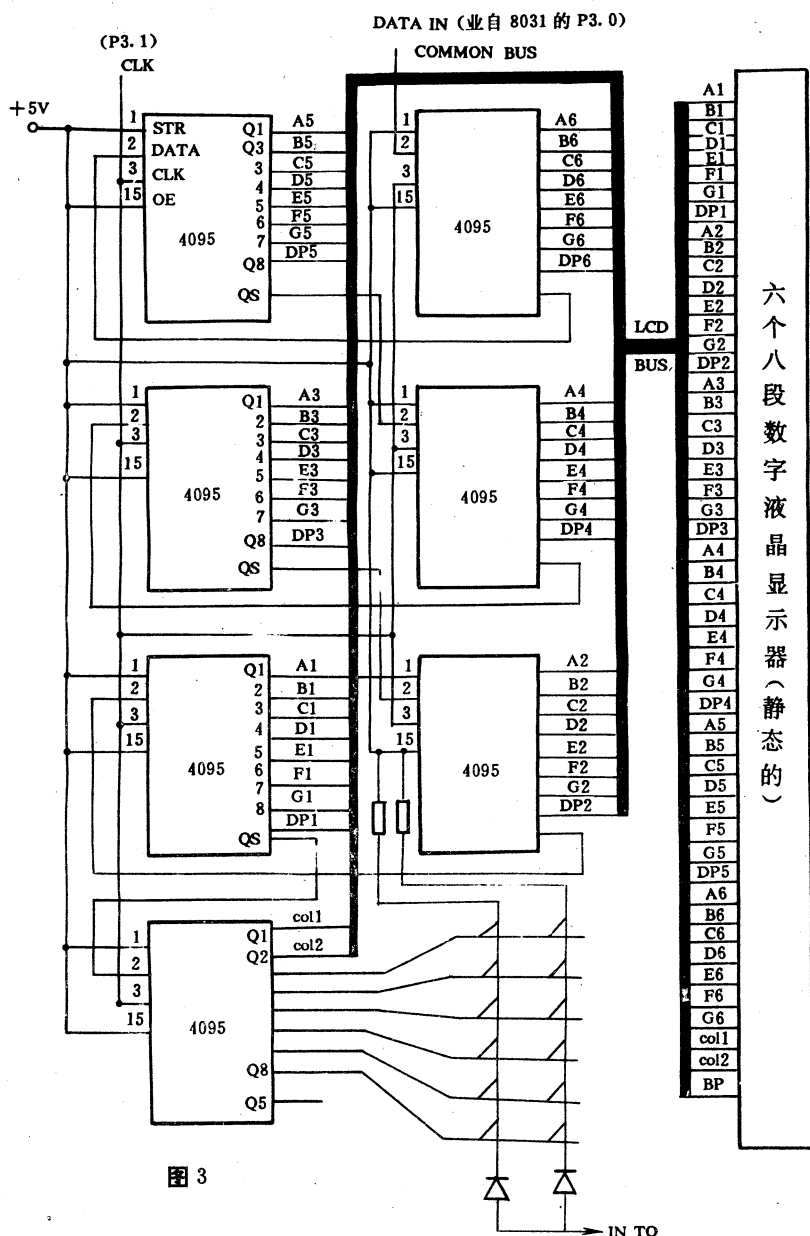


图 3

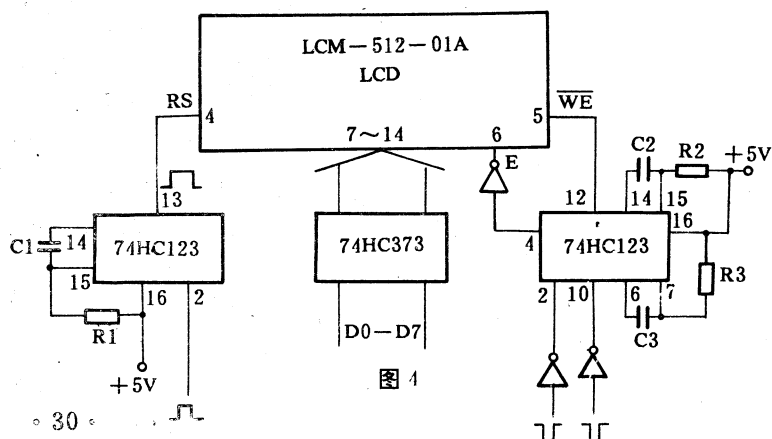


图 4

十小时)。这样就要求显示器的公共电极绝对不能接数字信号地。要加一个显示频率方波,它的相位与段电极(也称前电极)相位相反,两者严格对称,从而保证显示器平均电压等于零,如图 1。在图 3 中从单片机最小系统的 8031 串行口 P3.0, P3.1 分别输出要显示的数据和同步时钟。LCD 的公共电极也接在 4095 一个端上,这样公共极和前极同步求反,保证平均直流电压为 0。图 3 是 6 个八段静态液晶显示器驱动和接口电路。

这个接口所用程序是很简单的。即把串行口设为移位寄存器,每次输出 7 个字节数据(其中包括公共电极的)。经过 0.1 秒原来数据求反再输出,这样重复循环下去。

对于动态液晶显示器和单片机接口电路是比较复杂的。因为动态液晶要分时驱动不同的片段。一般一个 LCD(动态)有 3 个到 4 个 COM 端。如果有 32 根 SEG 线(即段驱动端,公共端有 3 个)这样可以显示 $32 \times 3 = 96$ 个显示段。每段是否显示由每个 SEG 与 COM 的电压来决定。这样使线路复杂化。但是在国外一般已把动态驱动集成到一个芯片上,因此它与单片接口也就不复杂了。

我们在便携式数据采集器采用日本三洋产的点阵式液晶显示组件 LCM-512-01A 作显示器。一次可显示一行 16 个 ASCII 码字符。这种显示屏由于内部有控制电路,字符发生器和显示数据存储,其驱动方式简单,可直接与 8031 的数据总线相连,进行写命令和数据。显示器和 8031 还有两条控制线 E 和 WE,一条命令/数据选择线 RS。其中 E 为选通 LCD 的使能信号,高电平有效。WE 为 LCD 的读/写控制线。WE 为高电平,则 8031 对 LCD 进行读操作,WE 为低电平时 8031 向 LCD 写命令或数据。由于

LCD 是一种低速 I/O 设备,它接受一个命令大约要 1—2 μ s 以上,而执行一个命令则需要 40 μ s—1.6ms 之间,因此 8031 向 LCD 写时要保持 1—2 μ s。对于 8031 的 12MHz 晶体,一般一条指令小于 1 μ s。所以我们为了保证正确的控制,传送用了 74HC123 单稳电路,从而保证速度的匹配,使数据的可靠传送。同时用 74HC373 锁存 8031 的数据线的数据。图 5 中定时电容 C1—C3

(上接第 25 页)

· 寻址方法是先在零页中作间接寻址,后再进行绝对 Y 变址寻址。

· 指令有: LDA(\$nn), Y, STA(\$nn), Y, ADC(\$nn), Y, SBC(\$nn), Y, AND(\$nn), Y, EOR(\$nn), Y, ORA(\$nn), Y, CMP(\$nn), Y 等 8 条。

四、寻址方式小结

1. 寻址方式共 13 种,但不是每一类指令都具有,那一类指令具有哪些寻址方式,可以查 6502 指令表。

2. 学习不同的寻址方法,应注意它们之间的异同点,特别应该掌握它们各自的功能。

3. 编程时可根据具体要求,灵活选用某条指令允许的某一种寻址方式,以达到程序占用内存少,执行速度快的目的。

4. 同一道题目,可以有不同的编程方法。一般来说,同一道题目可以用不同的寻址方式指令处理。

5. 累加器寻址、隐含寻址方式的操作数在 CPU 的

(上接第 28 页)

答案: 1. A② B① C② D④ E③

2. A④ B③ C① D②

3. A④ B③ C② D③

4. A⑤ B③ C②

5. A⑧ B③ C⑤ D① E⑥

四、分析

1. 正确。

2. 错误。顺序存储方式的优点是存储密度大,但由于插入、删除时要引起大量数据元素的移动,所以运算效率低。

3. 错误。双向链表的结点中就包含左、右两个链指针。

4. 正确。

5. 错误。本叙述对数组来说是正确的,而记录可以包含

(上接第 44 页)

四、保护电路和次级整流滤波

如果启动电源时,发出“嗡嗡”或“滴答”声时,首先肯定有一组输出电压短路。测量 R_{21} 电阻值可能比原值增大 7—12 Ω ,使在它的压降足以触发可控硅 Q_5 控制端,使可控硅导通,致使输出短路,电路停止振荡,起到保护电路作用。如果测量 R_{12} 值与原值差别不大,就要检查相应次级整流滤波电路上的有关元件。

为 150P,电阻 R_1 — R_3 为 33K,可满足大于 2 μ s 脉宽。为了简化硬件我们没有采用访问 LCD 的忙标志。这样避免数据双向锁存。

总之液晶显示器与单片机(特别 CHMOS)相结合在很多领域很有用途。特别在携带式移动的仪器和设备更是如此。

内部寄存器中,不必访问存储器,执行速度快。

6. 零页寻址占用内存少,执行速度快,有利于提高编程的效率。

7. 变址寻址方式,常用于处理数据块、表格等问题,方便灵活。

8. 立即数寻址常用于设置初始数据,程序初始化或当计数器选用。

9. 相对寻址专用于条件转移指令,偏移量指出跳转的目标地址。

10. 间接寻址常用来存放专用程序的入口地址,从而使专用程序方便地在存储器中浮动。

11. 先变址(X)间接寻址适用于多个数据块传送,后变址(Y)间接寻址则常用于动态数据处理。

12. 零页 X(Y)变址寻址和先变址(X)间接寻址,仅限于在零页中进行变址,而绝对 X(Y)变址,后变址(Y)间接寻址,可访问 64K 空间的任一地址。

一个或多个数据项,各数据项可以具有不同的数据类型。

6. 错误。队列的两端都不固定,一端专门进行存(插入)操作,另一端专门进行取(删除)操作,两端都在动态变化。

7. 正确。

8. 正确。

9. 正确。

10. 错误。串的顺序存储方式分为紧缩格式和非紧缩格式。采用紧缩格式时,一个存储单元中存放几个字符与机器字的长度有关,所以与计算机的编址方式是有关的。

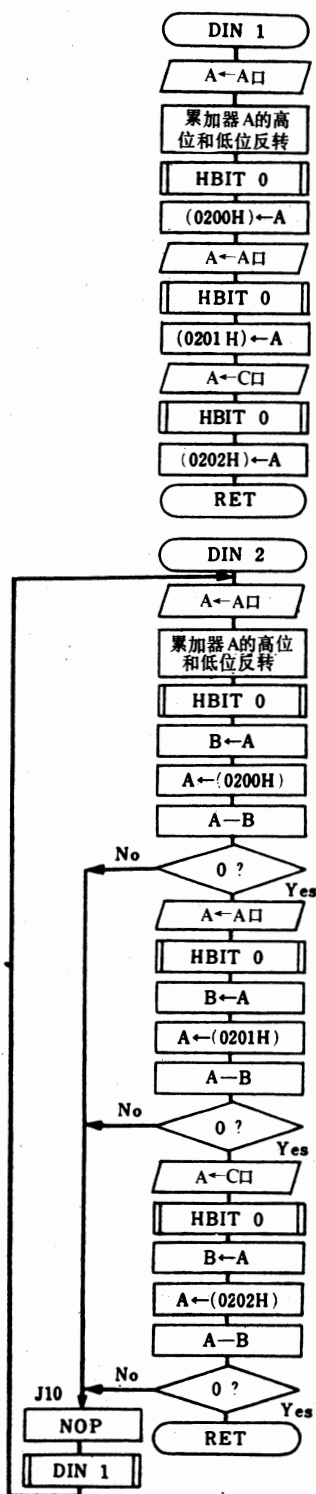
答案: 1. \checkmark 2. \times 3. \times 4. \checkmark 5. \times

6. \times 7. \checkmark 8. \checkmark 9. \checkmark 10. \times

另外发出“嗡嗡”或“滴答”声有另一个原因。测量 +12V 时,电压值超过 +15V。稳压管 CR_{20} 导通,使可控硅 Q_5 导通,也造成 +12V 短路接地而这种过压又不能通过 R_{16} 来调正。出现这个问题的关键是振荡环路上 R_{11} 电阻值变小使其振荡反馈信号增大, Q_5 集电极电压增大,使次级端产生电压过大,适当选择 R_{11} 使次级整流 +12V 电压不能超过 +15V 电压,方可使电源正常工作。

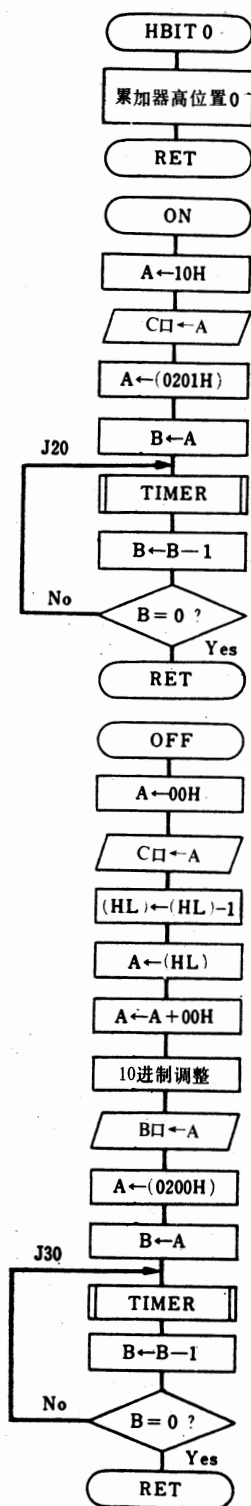


表 2



| 标记 | 助记符 | 地址 | 机 械 语 | 注 释 |
|--------|--------------|------|----------|-------------------------|
| DIN1 | IN A,(00H) | 0050 | DB 00 | 读入“断”时间数据 |
| | RRC A | 0052 | CB 0F | 累加器 A 的高位和低位反转 |
| | RRC A | 0054 | CB 0F | |
| | RRC A | 0056 | CB 0F | |
| | RRC A | 0058 | CB 0F | |
| | CALL HBIT0 | 005A | CD C0 00 | “断”时间数据放入 0200H 地址 |
| | LD(0200H),A | 005D | 32 00 02 | |
| | IN A,(00H) | 0060 | DB 00 | 读入“通”时间数据 |
| | CALL HBIT0 | 0062 | CD C0 00 | “通”的时间数据放入 0201H 地址 |
| | LD(0201H),A | 0065 | 32 01 02 | |
| | IN A,(02H) | 0068 | DB 02 | 读入“通”——“断”时间设定值 |
| | CALL HBIT0 | 006A | CD C0 00 | “通”——“断”次数数据放入 0202 地址 |
| | LD(0202H),A | 006D | 32 02 02 | |
| RETURN | RET | 0070 | C9 | |
| | | | | |
| DIN2 | IN A,(00H) | 0080 | DB 00 | 读入“断”时间数据 |
| | RRC A | 0082 | CB 0F | 反转累加器 A 的高位和低位 |
| | RRC A | 0084 | CB 0F | |
| | RRC A | 0086 | CB 0F | |
| | RRC A | 0088 | CB 0F | |
| | CALL HBIT0 | 008A | CD C0 00 | “断”时间数据放入 B 寄存器 |
| | LD B,A | 008D | 47 | |
| | LD A,(0200H) | 008E | 3A 00 02 | 比较两次读入的数据不相等时跳入 J10 |
| | CP B | 0091 | B8 | |
| | JP NZ,J10 | 0092 | C2 B0 00 | |
| | IN A,(00H) | 0095 | DB 00 | 读入“通”时间数据 |
| | CALL HBIT0 | 0097 | CD C0 00 | “通”时间数据放入 B 寄存器 |
| | LD B,A | 009A | 47 | |
| | LD A,(0201H) | 009B | 3A 01 02 | 比较两次读入的数据不相等时，跳入 J10 |
| | CP B | 009E | B8 | |
| | JP NZ,J10 | 009F | C2 B0 00 | |
| | IN A,(02H) | 00A2 | DB 02 | 读入“通”——“断”时间设定值 |
| | CALL HBIT0 | 00A4 | CD C0 00 | “通”——“断”次数数据放入寄存器 B |
| | LD B,A | 00A7 | 47 | |
| | LD A,(0202H) | 00A8 | 3A 02 02 | 比较两次读入的数据不相等时，跳入 J10 |
| | CP B | 00AB | B8 | |
| | JP NZ,J10 | 00AC | C2 B0 00 | |
| RETURN | RET | 00AF | C9 | |
| J10 | NCP | 00B0 | 00 | 比较结果不相等时，再次读入数据，返回 DIN2 |
| | CALL DIN1 | 00B1 | CD 50 00 | |
| | JP DIN2 | 00B4 | C3 80 00 | |

表 2



| 标记 | 助记符 | 地址 | 机械语 | 注 释 |
|--------|--------------|------|----------|---------------------|
| HBIT 0 | RES 7,A | 00C0 | CB BF | 累加器的高位置0 |
| | RES 6,A | 00C2 | CB B7 | |
| | RES 5,A | 00C4 | CB AF | |
| | RES 4,A | 00C6 | CB A7 | |
| RETURN | RET | 00C8 | C9 | |
| ON | LD A,10H | 00D0 | 3E 10 | 继电器为“通” |
| | OUT(02H),A | 00D2 | D3 02 | |
| | LD A,(0201H) | 00D4 | 3A 01 02 | 仅在设定时间继电器为“通” |
| | LD B,A | 00D7 | 47 | |
| J20 | CALL TIMER | 00D8 | CD D0 00 | |
| | DEC B | 00DB | 05 | |
| | JP NZ,J20 | 00DC | C2 88 00 | |
| RETURN | RET | 00DF | C9 | |
| OFF | LD A,00H | 00E0 | 3E 00 | 继电器“断” |
| | OUT(02H),A | 00E2 | D3 02 | |
| | DEC(HL) | 00E4 | 35 | 端口B显示的“通”-“断”设定次数-1 |
| | LD A,(HL) | 00E5 | 7E | |
| | ADD A,00H | 00E6 | C6 00 | |
| | DAA | 00E8 | 27 | |
| | OUT(01H),A | 00E9 | D3 01 | 仅在设定时间继电器“断” |
| | LD A,(0200H) | 00EB | 3A 00 02 | |
| | LD B,A | 00EE | 47 | |
| J30 | CALL TIMER | 00EF | CD D0 00 | |
| | DEC B | 00F2 | 05 | |
| | JP NZ,J30 | 00F3 | C2 BF 00 | |
| RETURN | RET | 00F6 | C9 | |

表 2

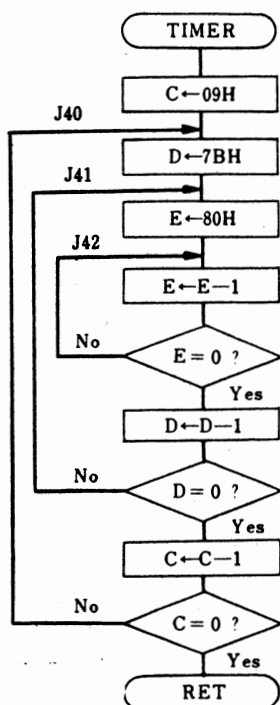
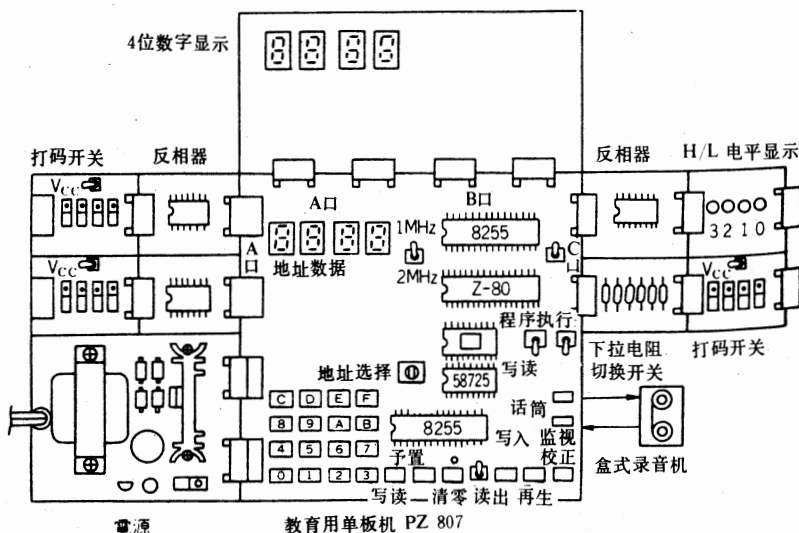
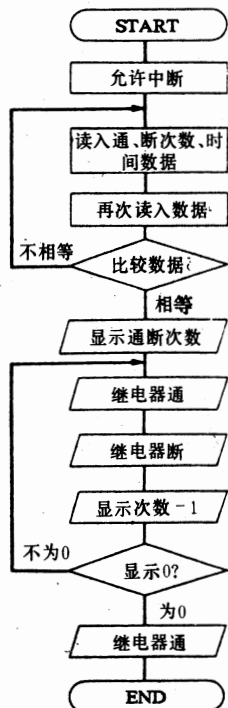
[illegible]

图 4

执行仿真程序,4 位数显部件显示预先设置的 ON、OFF 次数,端口 C 高位电平显示部件的第 0 位 LED 灯亮表示所设置的时间,同时,4 位数显部件出现减 1 显示。

如果仿真程序无误,即可将仿真程序改为工作程序,根据需要考虑是否写入 EPROM 存储器。装入工作用微电脑,对仪器完成电源 ON—OFF 自动运转控制。



微机视频接口器的研制

广西民族学院物理系(530006) 韦江维 覃龙生

目前,很多用户尤其是众多经费不足而又必须开设微机实验的大、中专院校、基础科研及家庭教育、娱乐,都需要一台既便宜又能完成一定任务的电脑。低档PC机已基本能满足这一需要。如能考虑充分利用现有资料又能增强PC机功能,那将给低档PC机增添色彩。为此,我们成功地研制出微机视频接口器,接上家用电视机就可完全取代原来的彩色显示器,使用户根据自己的需要,花不多的钱,就能配置成真正的个人电脑。

在我国,彩电的制式都采用PAL制,而目前几乎所有的微机显示系统都采用NTSC制。另外,在电视发射与接收系统中,为抑制外界的干扰,全电视信号的图像信号采用负极性调制,而由于微机显示系统受外界干扰较小,因此采用正极性调制,其行场同步信号是高电平有效。本视频接口器成功地完成上述的信号转换,即把微机的显示信号(包括R、G、B、I、VSYNC、HSYNC)在新副载波基础上重新组合,形成符合我国电视技术标准的视频信号,原理如图1所示。

本接口器的核心部件采用了MOTOROLA公司生产的MC1377彩色电视信号处理器,又称为PAL/NTSC转换器。它的主要功能就是从基础波段取出红、绿、蓝和同步信号,按PAL(或NTSC)制直接产生复合视频信号。其内部电路主要有:副载波形成电路、同步电路、压控相位调整、解编码器、输出放大调整电路。使用MC1377芯片,使得接口器电路十分简单,易于调制,工作可靠。

电路中,74LS244是视频接口器与彩色显示适配器的接口; $R_1 \sim R_6$ 是分压网络,因为MC1377的R、G、B输入信号电平要求为 $1.0V_{P-P}$ (输入的彩色信号电平为 $5.0V_{P-P}$); C_1 、 C_2 、 C_3 为MC1377的输入电容;同步信号 H_{SYNC} 、 V_{SYNC} 经异或门(74LS86)后形成复合同步信号,再经反相器(74LS04)反相后送至MC1377的复合同步输入端(2脚),另外的两级反相和电感L在一定程度上起到延时同步信号的作用;17、18脚为时钟输入,晶振采用4.43MHz晶体振荡器;20脚为PAL/NTSC转换开关,悬空为PAL制;19脚为(R-Y)与(B

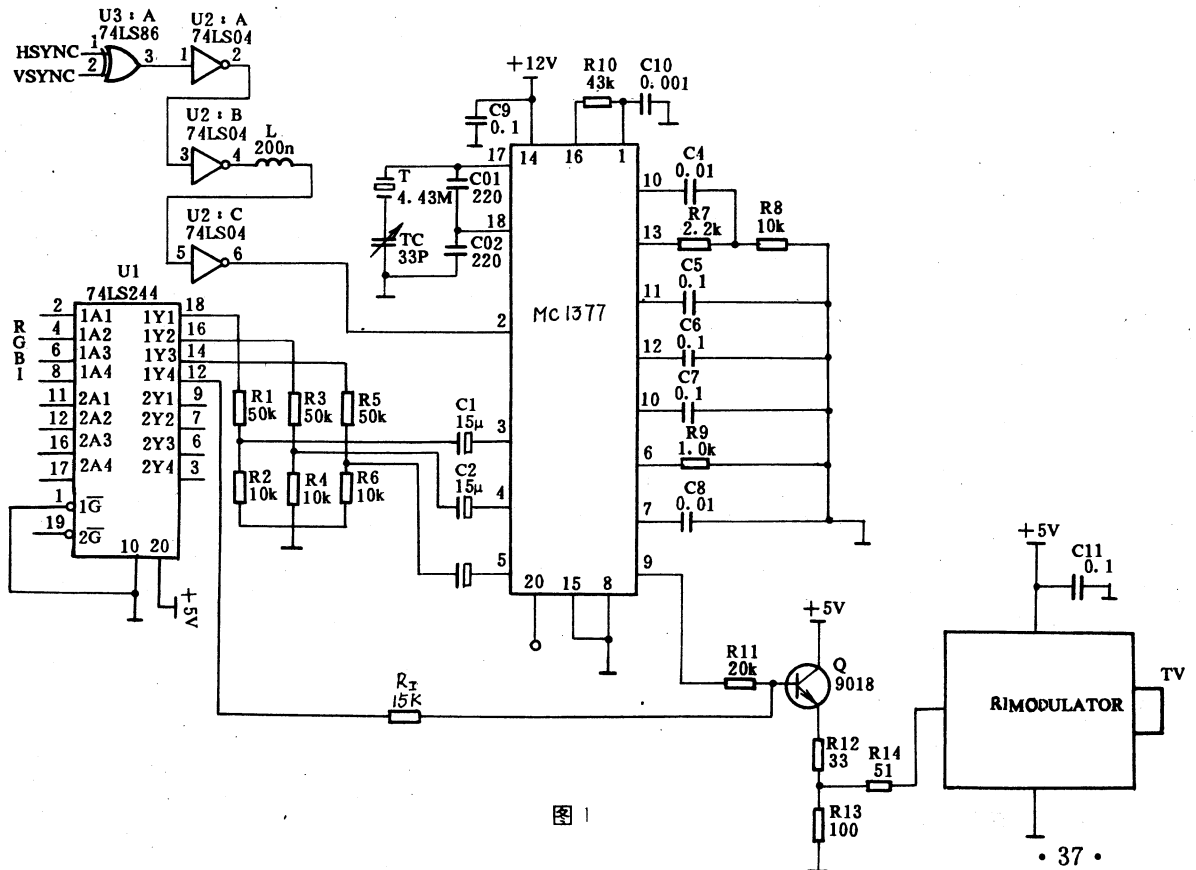


图 1

-Y)信号相位调整;16脚为8.2V参考电压输出;1脚为同步信号调整;10、13脚为色度信号调整;6、8脚为延时调整;9脚为复合视频信号输出,其输出幅度为2.5V_{r.p.p.}。晶体管Q主要起放大、阻抗匹配作用。RF调制器电路包括本机振荡器和混频器,它所输出的高频振荡与视频信号混频、经混频之后可将此信号直接与电视机天线连接。调节电视机频道,电视机就可当彩色显示器使用了。

显然,本接口器输出的只是复合视频信号,要形成全电视信号还得有音频信号。在这里我们就如何才能获得全电视信号作简单讨论:在介绍接口器电路时,曾经提到MC1377 16脚为8.2V参考电压输出,其主要作用是提供MC1374直流工作电压,MC1374是音、视频复合放大输出器,其内部电路主要有:音频调节器、声音载波振荡器、RF振荡器、RF调整器,只要将MC1377输出的复合视频信号和音频信号送至MC1374的相应引脚,就可获得全电视信号。当然,使用MC1374时在技术上有它自己的要求,另外还得需要一些外围电路支持(如4.5MHz陷波电路等),这里就不一一介绍了。

本视频接口器可连接黑白或彩色电视机,值得一提的是,用家用电视机充当显示器,由于分辨率有限,较理想的选择是选取CGA显示模式。

XMF学习机(单驱扩展箱) 改装为XMF及PC两用机

南京903信箱33分箱 杨青海

中华学习机有丰富的学习软件,是中小学教学的主要机种,但在实际工作中所使用的绝大多数为PC系列机,应用软件丰富,使学习机相形见绌。目前PC机价格虽然逐年降低,但皆在3000元左右,对一般家庭而言,这是一笔不小的开支。最近我对XMF机进行改装,投资千元,使XMF机成为XMF及PC两用机,大大提高了性能价格比,较好的解决了学与用的矛盾,效果良好,有兴趣的同志不妨一试。改装原理见图1。

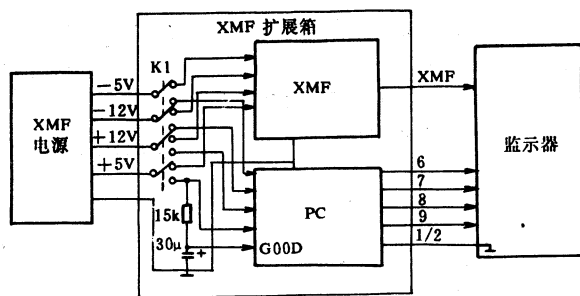
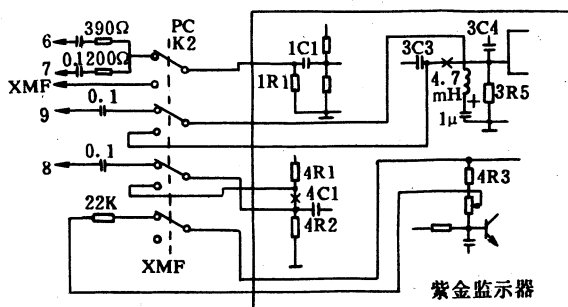


图1

一、电源及其改装:XMF机电源设计容量较大,该机使用手册提供数据为:+5V 4A;+12V 2A。机内电源组件上标注为:+5V 7A;+12V 3A,而PC机在基本配置(单驱、显示卡)状态下,实测+5V电流不超过3A,±12V、-5V电源容量更没问题,这样就为PC机解决了关键的供电问题。PC机所需的电源GOOD信号可通过简单的RC电路解决(见图1)。电源改装比较简单,打开电源组件外壳,把印刷板抽出一小部分,即可看到引出线焊点,用新的引出线(长度约30cm,最好与原线颜色、粗细一致),逐根替换原引线(原电源线及插头留给XMF机用),本人选用小型瓷质8D2W波段开关,其中3刀并联给+5V用,2刀并联给+12V用,-12V及-5V各用一刀,电源地线与XMF、PC机主板地线接在一起不需控制。



注:1、2、6、7、8、9为显卡9针插头针号

图2

二、显示器改装:如果选配具有直接使用电视机及学习机显示器接口的CGA卡则不需对学习机显示器进行改动。如使用单显或双频卡,把学习机显示器改为XMF-PC两用显示器(CG A/MDA),改装方法可参看《软件报》91年第23期。本人使用的显示器为紫金CZX-12型,改装方法略有不同,利用4×2拨动开关分别切换数据输入、行频、帧频调整,线路见图2。帧同步输入端与地之间增加4.7mH、1μF 2只元件组成的串联电路,是为了减少光栅顶部的卷边现象。为求简单,行扫描电路只把行频调整进行改动,所以720×350方式工作时,西文方式只能显示75列字符,但影响不大,中文方式下显示正常。

三、结构安装:XMF机单驱扩展箱中空余一台驱动器的位置,正好可以固定一只5.25英寸半高驱动器。电源组件移开,空出PC主板安装位置。在安装时应根据键盘接口及扩展槽的位置进行调整,以便于安装显示卡和驱动器卡。应注意,XMF机扩展箱内有效高度只有11cm,所以主板与底板之间在保证绝缘的前提下间隙应尽量小,否则显示卡将无法直立安装,有条件时使用90度转接口更为方便。电源组件可固定在机内空余处,但为了提高可靠性,利于散热,建议把电源固定在机箱后部。

改装后,在PC机状态下连续工作4小时以上运行正常。



电脑游戏机

第三章 F BASIC 的画面控制语句

山东苍山县机械电子化学工业局(277700) 于 春

四、SP 动作控制语句

1. 定义 SP 动作语句(DEF MOVE)

DEF MOVE 简写 DE. M.

语句功能是定义卡通图案的种类、运动方向、运动速度、移动量等。其格式为

DE. M. (n)=SP. (F,G,H,I,C,A)

式中(1)n—动作编号 0~7。

(2)F—SP 的种类编号 0~15

(3)G—运动的方向 0~8

(4)H—运动速度 1~255,H=1 最快,为 1 秒移动 60 点;H=255 最慢,255 秒移动 60 点。

(5)I—移动量 1~255,每一单位移动量为 2 点。如 I=200,则移动 400 点。

(6)C—显示的优先度 0~1,C=0 显示于 BG 面前,C=1 显示于 BG 面后。

(7)A—配色号码 0~3。

2. 启动 SP 语句(MOVE)

MOVE 简写 M.

MOVE 语句的功能是启动 DEF MOVE 语句所定义的 SP 开始运动,直到运动完 DEF MOVE 语句规定的移动量才停止。MOVE 语句的格式为

M. n₀,n₁,……,n₇ n₀~n₇ 任选

式中 n 为动作编号。MOVE 语句可同时启动 1~8 个卡通图案开始运动。

例 14 使乌龟向左移动 200 点。

```
10 CLS:SP.O.
```

```
20 DE. M. (0)=SP. (13,7,1,100,0,0)
```

```
30 M. 0
```

RUN 乌龟从屏幕中心(120,120)处开始向左移动,完成移动量后停止。

说明:在 DEF MOVE 语句中,运动方向 G、显示优先度 C 和配色号码 A 可以省略,省略 A 时,计算机自动设定 A=0,并使 SP 从(0,0)处开始运动。省略 G 时,计算机自动选定运动方向为 4。在省略时要注意 A、C、G 可以省略,但分隔符“,”要保留。

上例欲使乌龟反复动作,可加—40 句

```
40 PAU. 300:G. 10
```

RUN 后,乌龟反复移动,可以发现,乌龟每次移动的起点都在屏幕中心。能否人为设定乌龟的运动起点呢?答案是肯定的。POSITION 语句可以达到这一目的。

3. 规定 SP 动作座标语句(POSITION)

POSITION 简写 POS.

语句格式为

POS. n,X,Y

式中 n:动作编号 0~7。X,Y 为动作前的座标(起点座标)。X,Y 取值,一般 X 取 0~240,Y 取 0~220。

例 15 定义太空船向右上方运动,起点座标为(50,50)。

```
10 CLS:SP.O.
```

```
20 DE. M. (0)=SP. (9,2,1,100,0,0)
```

```
30 POS. 0,50,50
```

```
40 M. 0
```

RUN 太空船则从(50,50)起向右上方运动,若使其反复运动,则需加入 50 句

```
50 PAU. 300:G. 10
```

注意:在例 14、15 所加的反反复运动语句中均加了 PAUSE 300,即程序执行暂停 300,引入暂停的作用是给卡通图案留出一段时间完成所定义的运动量。若不加这一语句,则卡通图案只在原地抖动而不能移动。

例 16 试编程序使太空船的运动起点和运动方向随机变化。

```
10 CLS:SP.O.
```

```
20 I=RND(9):IF I=0 T. 20
```

```
30 X=RND(240):Y=RND(220)
```

```
40 DE. M. (0)=SP. (9,I,1,100,0,0)
```

```
50 POS. 0,X,Y:M. 0
```

```
60 PAU. 300:G. 10
```

例 17 编一程序使画面上显示 8 个卡通图案分别向 8 个方向运动,起点座标随机变化。动作反复进行。

```
10 CLS:SP.O.
```

```
20 FOR I=0 TO 7
```

```
30 DE. M. (I)=SP. (I,I+1,I+1,100,0,0)
```

```
40 X=RND(240):Y=RND(220)
```

```
50 POS. I,X,Y
```

```
60 M. I:N.
```

```
70 PAU. 500:G. 10
```

RUN 屏幕上显示 0~7 八种卡通图案从不同的起点,以不同的速度、向八个方向反复运动。

若将上例 30 行中配色号码由 0 换为 1/2,可以使八个卡通图案显示四种色彩。

若要 8~15 八种卡通图案显示、运动。可将 30 行中的 F 由 I 改为 8+I 即可。

从以上几例的程序运行中,读者可以发现,尽管 SP 的运动起点可以设定,但在反复运行时,第二次运行的起点不是第一次运行的终点,即运行路线不连贯。为使运行路线连贯,再介绍一个语句。

4. 测量 SP 的座标语句(XPOS、YPOS)

XPOS 简写 XP.
YPOS 简写 YP.

使用这两个语句,可以测量卡通图案动作前的座标或动作结束时的座标,也可以连续测试卡通图案在运动中的座标。其格式为

XP. (n) : YP. (n)

式中 n 为卡通图案的动作编号,两语句中的 n 必须相同。有了这两个语句,可在卡通图案动作前,测量出所在座标(X,Y),然后把座标值赋给 POSITION 语句,就可以使卡通图案的运行路线连贯。

例 18

```
10 CLS : SP. 0. : X=50 : Y=50
20 F.I=1 TO 8
30 DE. M. (0)=SP. (0,I,1,100,0,0)
40 POS. 0,X,Y
50 M. 0 : PAU. 300
60 X=XP. (0) : Y=YP. (0) : N.
```

RUN 显示玛丽由(50,50)开始连续向八个方向运动。

5. 判断动作完成语句[MOVE(n)]

以上举例中,都要引入 PAUSE 语句,留出一定的时间,等待卡通图案完成运动量后再运行以后程序。这种方法虽然有效,但不精确。尤其在连续运动时,不是时间不够,没有完成移动量就改为下一运动,就是时间多余,上一移动量完成后停一段时间才进行下一运动。引入 MOVE(n)语句可以弥补这一缺陷。

MOVE(n) 简写 M. (n)

MOVE(n)语句用于判断 n 号卡通图案动作是否完成。当动作正在进行中 MOVE(n)=-1,当动作完成时 MOVE(n)=0。

例 19 编一程序,令玛丽哥哥走八边形

```
10 CLS : PAU. 100
20 LOC. 9,9 : P. "Ma Li Ge Ge";
   LOC. 6,10 : P. "Zou Ba Bian Xing"
30 SP. 0. : X=35 : Y=130
40 F.I=0 TO 7
50 DE. M. (0)=SP. (0,I+1,1,30,0,0)
60 POS. 0,X,Y
70 M. 0
80 IF M. (0)=-1 T. 80
90 X=XP. (0) : Y=YP. (0)
100 N.
```

6. 令 SP 停止、消失语句(CUT、ERA、CAN)

(1)令卡通图案停止动作语句(CUT)

CUT 语句可使 MOVE 语句启动的卡通图案在移动中停止,等待下一个 MOVE 语句的启动。再启动后,继续运行上一次未运行完的移动量,直至运行终了才停止。CUT 语句可同时定义八个卡通图案停止运动。其语句格式为:

CUT n₀,n₁,.....n₇ n 为动作编号

(2)令卡通图案消失语句(ERA)

ERA 可使 MOVE 语句启动的卡通图案在移动中消失,直到下一次 MOVE 语句启动后,再显示,继续运行上次未运行完的移动量,直至运行终了才停止。ERA 语句可同时定义八个卡通图案消失。语句格式为

ERA n₀,n₁,.....n₇ n 为动作编号

(3)令卡通图案永远消失语句(CAN)

CAN 语句可使 MOVE 语句启动的卡通图案在移动中永远消失。既使再输入 MOVE 语句也不能启动显示,直到程序运行终了。再重新运行或者再重新用 EDF MOVE 语句定义,才能显示。该语句主要用于游戏程序设计中,当卡通图案超过八个时,使程序在运行中消除部分卡通图案,另定义启动其它的卡通图案。(注:虽然 ERA 语句也能令卡通消失,但它仍占用八个卡通之中的位置,这时,若定义其它卡通,仍不能显示。)

在游戏程序的设计中,CAN 语句是一条较有用的语句,随机手册中没有介绍,读者自己添上。

CAN 语句可同时定义八个卡通图案永远消失。其语句格式为

CAN n₀,n₁,.....n₇

(4)三个语句的不同点

三个语句的不同点在于 CUT 语句只令卡通图案运行停止,但不消失;ERA 语句令卡通图案停止并消失;CAN 语句则令卡通图案永远消失。CUT、ERA 语句作用后,可用 MOVE 语句再启动;CAN 语句作用后,用 MOVE 语句不能启动。实际上,CAN 语句的作用就是解除 DEF MOVE 语句的定义。

例 20,编一程序,演示 CUT、ERA、CAN 语句的功能区别。

```
10 CLS : SP. 0.
20 DE. M. (0)=SP. (0,3,1,100,0,0)
30 DE. M. (1)=SP. (1,3,1,100,0,0)
40 DE. M. (2)=SP. (2,3,1,100,0,0)
50 POS. 0,10,70,
60 POS. 1,10,120
70 POS. 2,10,170
80 M. 0,1,2 : PAU. 100
90 CUT 0 : P. "CUT0" : PAU. 100
100 ERA 1 : P. "ERA1" : PAU. 100
110 CAN 2 : P. "CAN2" : PAU. 100
120 M. 0,1,2:P. "MOVE 0,1,2"
```

RUN 显示 3 个卡通图案,0 号玛丽哥哥,1 号丽莎,2 号飞鸟自屏幕左边向右运动。运动大约 2 秒钟,玛丽停止,屏幕左上角显示 CUT0,随后大约 2 秒钟,丽莎消失,左上角显示 ERA1,又过大约 2 秒钟,飞鸟消失,左上角显示 CAN2,两秒钟后,丽莎出现与玛丽同时运动,左上角显示 MOVE 0,1,2,这时,飞鸟不再出现。

通过上例,读者可以直观体会到这三个语句的不同和各自的特点。

上例运行可反复按 RUN 多观察几次,也可以补入

使程序自动反复运行。

7. 判断卡通相遇语句(CRASH)

CRASH 简写 CR.

在游戏程序的设计中,经常要判断一卡通图案是否与其它卡通相遇,若相遇,相应作某种处理。如采蘑菇游戏中,当玛丽与红蘑菇相遇时,身体要比以前增大一倍。这就要首先判断是否相遇,若相遇转去执行身体长大子程序后再返回,若不相遇,则继续往下运行。在 F BASIC 语言中判断是否相遇使用的是 CRASH 语句,其格式为

CRASH(n) 简写 CR. (n)

式中 n 为被测试卡通图案的动作编号,在卡通图案的运行中,被测卡通与几号卡通相遇,CRASH(n)的值就等于相遇卡通的编号,未相遇 CRASH(n)的值则等于-1。

例 21,令 0~7 类八个卡通图案随机运动,分别测试 8 个卡通是否与其它卡通相遇,打印出各自的 CRASH 值。

```
10 CLS : SP. O.
20 F. I=0 TO 7
30 DE. M. (I)=SP. (I,I+1,I+1,200,0,0)
40 X=RND(170) : Y=RND(150)
50 POS. I,X+I*10,Y+I*10
60 M. I
70 A=CR. (I) : P. I"#"="";A
80 N.
90 PAU. 300 : G. 10
```

RUN 屏幕左上角依次打印出 0#~7# 八个卡通图案的各自的 CRASH 值。

上述程序只能测试一次相遇。若要测试某一个卡通图案在全部运动过程中与其相遇的所有卡通,可将上述程序改动如下:

```
(1)删去 70 行
(2)加 85 行
85 A=CR. (7);P. "7#CRASH="A;
IF M. (7)=-1 T. 85
```

RUN 在屏幕左边垂直打印 7#CRASH=与其相遇的卡通编号或-1。读者可清楚地看到,7#卡通太空堡垒与其它卡通相遇时 7#CRASH=后由-1 变为相遇的卡通编号。

五、操纵器控制语句(STICK、STRIG)

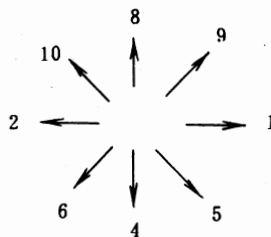
随机手册中仅介绍了两条操纵器语句,没有介绍如何使用操纵器控制卡通图案运动,而且只介绍了 STICK 语句的四个方向控制,遗漏了另外四个斜方向的控制。因此,使大多数使用者对操纵器如何控制卡通束手无策,不知从哪里下手编制程序,本文通过示例介绍使用方法。

1. 操纵器方向控制语句(STICK)

STICK 简写 STL.

STICK 语句可使用操纵器对卡通图案进行八个运

动方向的控制。在八个方向的取值为



2. 操纵器功能控制语句(STRIG)

STRIG 简写 STRI.

随机手册中对 STRIG 的简写 STR. 是错误的,请读者更正。

有关这两个语句的说明,请参阅随机手册,现通过示例,介绍如何使用操纵器控制卡通图案运动。

例 22 用操纵器控制玛丽哥哥运动。

```
10 CLS:SP. O. : X=35 : Y=130 : I=0
20 K=STI. (0) : IF K=0 T. 20
30 IF K=1 T. I=3
40 IF K=2 T. I=7
50 IF K=4 T. I=5
60 IF K=8 T. I=1
70 IF K=9 T. I=2
80 IF K=5 T. I=4
90 IF K=6 T. I=6
100 IF K=10 T. I=8
110 DE. M. (0)=SP. (0,I,1,2,0,0)
120 POS. 0,X,Y
130 M. 0
140 IF M. (0)=-1 T. 140
150 X=XP. (0) : Y=YP. (0) : G. 20
```

RUN 用 1# 操纵器可控制玛丽哥哥向各个方向运动。(待续)

邮 购 信 息

电子工业出版社于一九九二年元月十四日在北京人民大会堂举办首发式,隆重推出全国首次家电维修技术精华大奖赛的丰硕成果——《家电维修技术精华丛书》(1—10)。欢迎广大读者选购。

书名及邮购价(含邮挂费)

①收音机 8.30 元 ②收录机、组合音响 8.70 元 ③黑白电视机 14.40 元 ④彩色电视机(上、下册)30.00 元 ⑤录像机、摄像机 8.50 元 ⑥卫星电视接收机、共用天线 6.20 元 ⑦电子表、电表、万用表 6.20 元 ⑧游戏机、家用微电脑、计算器 6.00 元 ⑨家用制冷、空调设备 17.80 元 ⑩小家电 8.70 元

邮购办法:1. 邮局汇款请寄北京万寿路 173 信箱电子工业出版社发行部邮购科(汇款单附言栏内请注明购书清单)。2. 银行汇款:单位名称:电子工业出版社销售服务部,开户行:北京市翠微路分理处,帐号:661036—40 邮政编码:100036 电话:813693
注:购全套书邮购价 110 元。



维修经验谈

NP125 型复印机中一个易被忽视的故障

常州 102 医院(213003) 许 鹰

NP125 型是广东湛江复印机厂生产的一种性能稳定,质量较好的复印机,其使用寿命长,操作维护简单,在社会上有相当的一批用户。但是,在使用了 7—8 万张以后,常会出现“E4”故障,其现象为:复印机一启动,操作台就出现“E4”故障字样,同时左边依次出现“卡纸”、“缺少墨粉”、“缺纸”等图形显示,机器死锁,无法运行。当把电源插头断开再重新插上或将前侧盖板拉开再合上,机器偶能运行一下,或中途再次停止,重复上述过程,仍然如此。

对此故障,开始感到十分棘手,因为这常常是复印机的交流电路板或直流电路板出了故障。为此,笔者花了许多时间对这二块电路板进行了仔细的检查,发现各电路工作点均正常,集成电路也没什么问题,反复检

查了好几遍,结果均为正常,而故障仍然存在。最后又查看各动态信号,发现复印机的时钟脉冲信号 CPS 幅度很小,有时干脆没有,对此信号顺藤摸瓜,最后终于发现在复印机后侧面中部靠近硒鼓处,有一小块与交流电路板用电线相连的转轮式光电时钟脉冲发生器,由于长时间的使用,光电发射与接收装置上已积满了许多灰尘,光电脉冲无法正常产生,经用干净毛笔掸去灰尘,装好试用,故障立即排除,恢复正常,真是踏破铁鞋无觅处,得来全不费功夫。因此当 NP125 复印机出现“E4”故障时,不妨先检查一下光电时钟脉冲发生器,如果不是该问题,再检查交、直流电路板或送修理部门修理,这样可以少走弯路。

PC—88 键盘分析与故障维修

西安空军工程学院六系计算中心(710038)

王耀亭

编者按:

PC—88 兼容机是香港两个公司分别组装的,习惯上称美国 PC—88 和日本 PC—88。因为各主要部件的来源不同,所以二者不仅在主机、终端、键盘的外形上不同,而且在实现功能的逻辑线路上也不同。它们的相同之处是:都是 16 位个人计算机,并且与 IBM PC 完全兼容。凡是在 PC—88 上运行的软件,均可以在 IBM PC 机上运行。基本配置如下:

128K 内存

打印机并行口

RS—232 串行口

两个 5 $\frac{1}{4}$ 软盘驱动器

ROM BASIC(日本 PC—88 机在不用 MS—DOS 启动时,将自动进入 BASIC 状态。)

有单色和彩色适配器两种

键盘用 83 键

PC—88 机在操作使用上和 IBM PC 机完全相同。

键盘是人—机通信的通道,它通过键盘接口直接和系统交换数据信息。目前微机种类繁多,键盘接口电路也各有不同。IBM—PC/XT 的键盘接口采用 Intel8048 单片微处理器,IBM—PC/AT 的键盘接口采用一片 8 位单片微处理器 Intel8042/8742,PC—88 兼容

机中键盘接口是用一片 8748 单片微处理器来控制的。由于采用的微处理器和其它控制芯片不同,实现功能的逻辑线路有很大的差异,给维修带来很大的困难。我们有一批 84 年配置的 PC—88 兼容机,经过几年的使用,不仅主机、软驱、终端不断出现故障,而且键盘也时有故障出现。下面就 PC—88 兼容机键盘的工作原理和维修分析如下:

1. PC—88 键盘的逻辑线路

该机无随机逻辑线路,因维修需要,我们根据实物描绘线路如图 1 所示,大致可以表明逻辑关系。

2. 工作原理

键盘接口电路采用 8748 微处理器,矩阵译码电路由两块 MSM4051 和一块 MC74HC42 组成。MSM4051 为 8 选 1 模拟开关电路。它同 C511 和 C541 芯片,也称单八通道多路传输器。它有三个输入端 A、B、C,选中输出八种状态中的一种。无论那一路输出,③脚 COMMON 端同时有输出,接到 LM311 的 ③脚。两块 MSM4051 译码产生 16 条行扫描。MC74HC42 为 BCD 十进制译码器,用来产生 8 条列扫描。检测电路采用单电压比较器 LM311,②、③脚输入比较信号,通过放大,⑦脚输出的波形是②脚波形的检测放大,送到 8748 进行检查。8748 根据检测器送来的信号,确定有无键被按下,和同时按下的键数以及按键的位置。另外两块芯片是 7405 和 7408,它们起驱动控制作用。

3. 故障维修

〔例 1〕按任何键主机无响应,也就是说,键盘没有输出信号到主机。

打开键盘,首先用示波器观察键盘接口,看有无+5V、KBDDATA、KBDCLK及KBDRESET信号。只要有开机或复位动作,KBDRESET就应该有由高电位—低电位—高电位的瞬变。KBDDATA和KBDCLK平时为高电位,只要有按键动作,就可以看到负脉冲串。若发现那个信号不正常,则可跟踪查找。

8748的②、③脚输入振荡脉冲,时钟频率为6MC,④脚为复位输入,低电平有效。用来对8748初始化。①脚 T_0 为编程方式控制端,从接口2通过7408的③脚送来。⑨脚 T_1 为测试端,从接口1通过7405的⑧脚送来。②⑦、②⑧、②⑨、②④脚接74HC42的12、13、14、15脚。③⑩、③⑪、③⑫脚接两个4051的11、10、9脚。静态测量74HC42的各输出端均有3.76V,两个4051的输出端均为2.23V电平。按键时,74HC42输出端有负脉冲串。

根据各信号之间的关系,用示波器跟踪检查波形,

就会发现一些异常现象,特别是两块4051的输入输出波形、电平应相似。通过检查,发现静态时两块4051的电平基本相同,但在动态时一块4051输出电平时却叠加了负脉冲,影响了译码电路和检测电路。换了之后,电位和波形正常,键盘工作也正常了。

〔例2〕只有部分按键起作用,部分按键无反应。按Caps Lock和Num Lock键不起作用。

用示波器观察到,LM311的②脚电压为2V,是正常的,但③脚却在2V上叠加有正脉冲,不正常。7408的⑧脚平时高电平,但按Num Lock键后低不下来,灯不亮。断开 Q_3 的④脚,只有8个字符(8、9、I、O、K、L、M)不显示,其它正常。测量④与⑩脚之间的阻值只有110K左右,④脚有+5V电压,说明两脚间有短路。换了 Q_3 后,键盘工作正常了。

APPLE II 电源常见故障维修方法

天津肿瘤医院 邓满国

APPLE II 微机电源是一种高频转换的开关电源。一旦微机不能正常工作或出现信息误差,最大的可能是由于电源故障引起的,现将维修电源中排除故障的方法供读者参考。

见电路原理图。从图中可知 CR_1 、 C_1 、 C_2 、 C_3 是前级整流滤波环节。初级绕组 L_1 、 R_4 、 C_4 、 Q_3 ,绕组 L_2 、 R_{11} 、 CR_6 组成振荡环节。 Q_1 、 Q_2 、 CR_7 、 Q_4 、 AR_1 是反馈控制环节。 CR_{20} 、 R_{27} 、 R_{21} 、 Q_5 是保护电路。以下就每一个电路上可能出现的问题加以讨论。

一、前级整流与滤波环节

若无电压输出,检查保险管已断,这多半是 Q_3 或 CR_3 已坏。如果经检查两管无问题,而换上新保险管开

机后保险管再断,那么应该检查 CR_1 中各P-N结电阻,若其正反向电阻相等或趋于零,则该管已击穿,若 CR_1 完好,则故障出现在前级整流滤波电路。

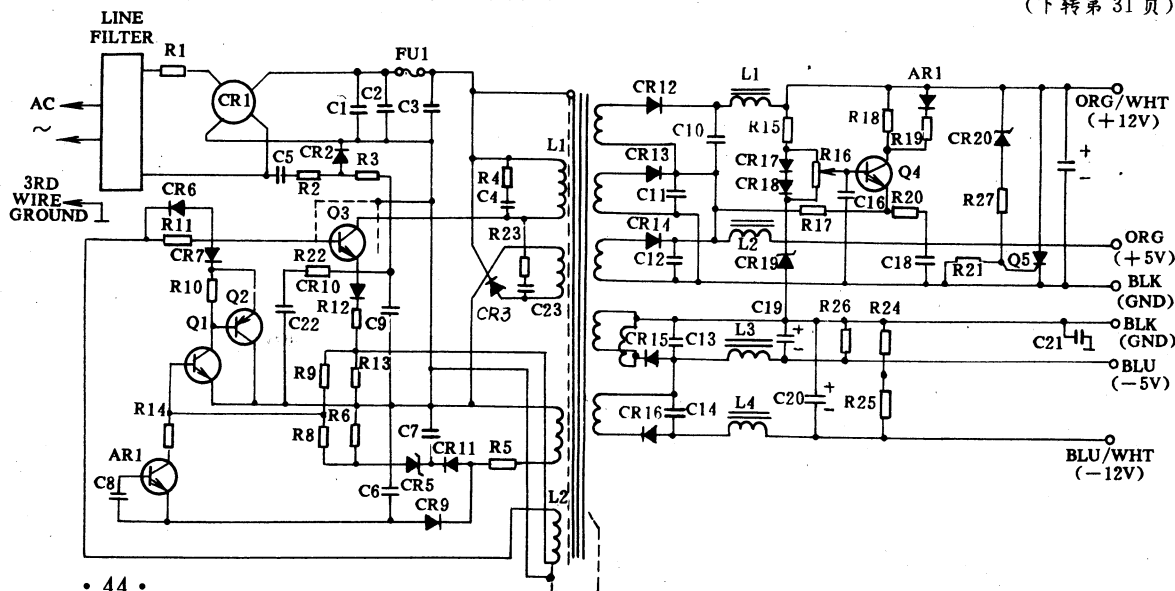
二、振荡环节

电路正常振荡,能在 Q_3 集电极上用示波器测出有600V大约20KHz交流电。若只有300V直流电压,说明振荡环节有问题。检查 Q_3 管各P-N结是否烧断。若检查是好的,再查一查 R_{12} 有可能呈高阻或开路状态,而 R_{12} 只有十几个欧姆,若更换 R_{12} 后,电路还不能起振,就要进一步检查振荡回路上的 R_{11} 和 CR_6 是否同时出现问题。以上这些情况是电路不能起振的原因。

三、反馈控制环节

± 12 、 ± 5 四组电压不准时,首先调正 R_{16} 。如果调正无效,就可能是光电偶合器件 AR_1 出现问题。它有问题就不能使电压过高或低的信号反馈给 Q_1 、 Q_2 来调整振荡管 Q_3 的输出电压幅度,而振荡环路上其它元件的损坏也可能使调整电压的 R_{16} 不起作用。

(下转第31页)



IBMPC/XT 机的软磁盘驱动器磁头校准程序

宁波市农行电脑室(315040) 范思尧

软磁盘驱动器是一个机电结合的精密产品。驱动器的磁头在长期使用中容易产生偏移,使磁头不能正常读写。校准磁头偏移一般应用校准盘和示波器进行测试校准;如没有该设备也可自编一个校准程序进行校准。本人在实际工作中用 DEBUG 的汇编功能自编了一个校准程序,并在 0520 系列机上多次成功地校准了驱动器的磁头。使用时只要对该程序中的有关寄存器值作适当设置(CH=磁道号 0-39,DH=磁头号 0 或 1,DL=驱动器号 0-1)就可以对不同驱动器,不同磁头,不同磁道分别进行校准,为了使用方便可以预先

把校准不同驱动器、磁头、磁道的程序分别命名,以便调用方便。校准时只要把一个完好的软盘放入要校准的驱动器,再调用相应的程序,当磁头偏移时,则屏幕显示 ERROR,当校准后则显示 PASS;调整磁头的一般顺序是先调整 0 面定磁头 39 道和 0 道,再调整 1 面动磁头 39 道和 0 道;当 39 道和 0 道均通过后,其它磁道一般均能通过,不一定再需校准;具体操作时需根据磁头定位机构的不同,采取不同的调整方法,以保证磁头物理位置的准确,以下为程序清单。

(本程序在 0520A,0520DH 等机上通过)

c>debug diskc.com

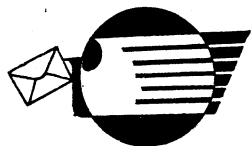
-d0100 01ff

```
5622:0100 52 45 53 45 54 2E 2E 2E-2E 2E 2E 2E 2E 2E 2E RESET.....
5622:0110 2E 2E 2E 52 45 41 44 2E-2E 2E 2E 2E 2E 2E 2E ...READ....
5622:0120 2E 2E 2E 2E 57 52 49 54-45 2E 2E 2E 2E 2E 2E 2E ...WRITE...
5622:0130 2E 2E 2E 2E 2E 2E 56 45-52 46 2E 2E 2E 2E 2E 2E .....VERF..
5622:0140 2E 2E 2E 2E 2E 2E 2E 2E-2E 2E 2E 2E 2E 2E 2E .....
5622:0150 2E 2E 2E 2E 2E 2E 2E 2E-2E 2E 2E 2E 2E 2E 2E .....
5622:0160 2E 2E 2E 2E 2E 2E 2E 2E 50-41 53 53 2E 2E 2E 2E 2E .....PASS.
5622:0170 2E 45 52 52 4F 52 2E 2E-2E 2E 2E 2E 90 90 90 90 .ERROR.....
5622:0180 90 90 90 90 90 90 90 90 90-90 90 90 90 90 90 90 .....
5622:0190 90 90 90 90 90 90 90 90 90-90 90 90 90 90 90 90 .....
5622:01A0 90 90 90 90 90 90 90 90 90-90 90 90 90 90 90 90 .....
5622:01B0 90 90 90 90 90 90 90 90 90-90 90 90 90 90 90 90 .....
5622:01C0 90 90 90 90 90 90 90 90 90-90 90 90 90 90 90 90 .....
5622:01D0 90 90 90 90 90 90 90 90 90-90 90 90 90 90 90 90 .....
5622:01E0 90 90 90 90 90 90 90 90 90-90 90 90 90 90 90 90 .....
5622:01F0 90 90 90 90 90 90 90 90 90-90 90 90 90 90 90 90 .....
```

-u0200 02cb

| | | | | | |
|------------------|------|---------|------------------|------|---------|
| 5622:0200 BB0001 | MOV | BX,0100 | 5622:0227 B80030 | MOV | AX,3000 |
| 5622:0203 E8B900 | CALL | 02BF | 5622:022A 8EC0 | MOV | ES,AX |
| 5622:0206 B201 | MOV | DL,01 | 5622:022C BB0000 | MOV | BX,0000 |
| 5622:0208 B400 | MOV | AH,00 | 5622:022F B402 | MOV | AH,02 |
| 5622:020A CD13 | INT | 13 | 5622:0231 B008 | MOV | AL,08 |
| 5622:020C F6C4FF | TEST | AH,FF | 5622:0233 B527 | MOV | CH,27 |
| 5622:020F 7409 | JZ | 021A | 5622:0235 B101 | MOV | CL,01 |
| 5622:0211 BB7101 | MOV | BX,0171 | 5622:0237 B600 | MOV | DH,00 |
| 5622:0214 E8A800 | CALL | 02BF | 5622:0239 B201 | MOV | DL,01 |
| 5622:0217 EB07 | JMP | 0220 | 5622:023B CD13 | INT | 13 |
| 5622:0219 90 | NOP | | 5622:023D 07 | POP | ES |
| 5622:021A BB6701 | MOV | BX,0167 | 5622:023E F6C4FF | TEST | AH,FF |
| 5622:021D E89F00 | CALL | 02BF | 5622:0241 7409 | JZ | 024C |
| 5622:0220 BB1301 | MOV | BX,0113 | 5622:0243 BB7101 | MOV | BX,0171 |
| 5622:0223 E89900 | CALL | 02BF | 5622:0246 E87600 | CALL | 02BF |
| 5622:0226 06 | PUSH | ES | 5622:0249 90 | NOP | |

| | | | | | | | |
|-----------|--------|------|---------|-----------|--------|--------|---------|
| 5622:024A | EB06 | JMP | 0252 | 5622:0293 | B404 | MOV | AH,04 |
| 5622:024C | BB6701 | MOV | BX,0167 | 5622:0295 | B008 | MOV | AL,08 |
| 5622:024F | E86D00 | CALL | 02BF | 5622:0297 | B527 | MOV | CH,27 |
| 5622:0252 | BB2401 | MOV | BX,0124 | 5622:0299 | B101 | MOV | CL,01 |
| 5622:0255 | E86700 | CALL | 02BF | 5622:029B | B600 | MOV | DH,00 |
| 5622:0258 | 06 | PUSH | ES | 5622:029D | B201 | MOV | DL,01 |
| 5622:0259 | B80040 | MOV | AX,4000 | 5622:029F | CD13 | INT | 13 |
| 5622:025C | 8EC0 | MOV | ES,AX | 5622:02A1 | 07 | POP | ES |
| 5622:025E | BB0000 | MOV | BX,0000 | 5622:02A2 | F6C4FF | TEST | AH,FF |
| 5622:0261 | B403 | MOV | AH,03 | 5622:02A5 | 750C | JNZ | 02B3 |
| 5622:0263 | B008 | MOV | AL,08 | 5622:02A7 | 90 | NOP | |
| 5622:0265 | B527 | MOV | CH,27 | 5622:02A8 | 90 | NOP | |
| 5622:0267 | B101 | MOV | CL,01 | 5622:02A9 | 90 | NOP | |
| 5622:0269 | B600 | MOV | DH,00 | 5622:02AA | BB6701 | MOV | BX,0167 |
| 5622:026B | B201 | MOV | DL,01 | 5622:02AD | E80F00 | CALL | 02BF |
| 5622:026D | CD13 | INT | 13 | 5622:02B0 | EBD2 | JMP | 0284 |
| 5622:026F | 07 | POP | ES | 5622:02B2 | 90 | NOP | |
| 5622:0270 | F6C4FF | TEST | AH,FF | 5622:02B3 | BB7101 | MOV | BX,0171 |
| 5622:0273 | 7409 | JZ | 027E | 5622:02B6 | E80600 | CALL | 02BF |
| 5622:0275 | BB7101 | MOV | BX,0171 | 5622:02B9 | EBC9 | JMP | 0284 |
| 5622:0278 | E84400 | CALL | 02BF | 5622:02BB | 90 | NOP | |
| 5622:027B | EB07 | JMP | 0284 | 5622:02BC | 90 | NOP | |
| 5622:027D | 90 | NOP | | 5622:02BD | 90 | NOP | |
| 5622:027E | BB6701 | MOV | BX,0167 | 5622:02BE | 90 | NOP | |
| 5622:0281 | E83B00 | CALL | 02BF | 5622:02BF | B90800 | MOV | CX,0008 |
| 5622:0284 | BB3601 | MOV | BX,0136 | 5622:02C2 | 8B17 | MOV | DX,[BX] |
| 5622:0287 | E83500 | CALL | 02BF | 5622:02C4 | B402 | MOV | AH,02 |
| 5622:028A | 06 | PUSH | ES | 5622:02C6 | CD21 | INT | 21 |
| 5622:028B | B80020 | MOV | AX,2000 | 5622:02C8 | 43 | INC | BX |
| 5622:028E | 8EC0 | MOV | ES,AX | 5622:02C9 | E0F7 | LOOPNZ | 02C2 |
| 5622:0290 | BB0000 | MOV | BX,0000 | 5622:02CB | C3 | RET | |



普及型 PC 个人用户软件交流联谊 活动问题解答(三)

读者联谊

北京中国农科院计算中心(100081) 王路敬

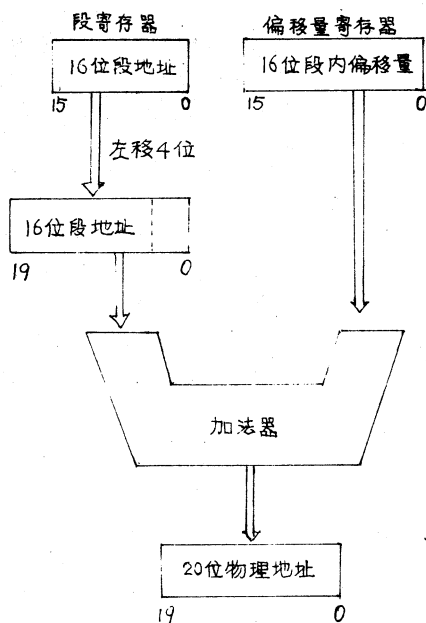
6. PC 机程序中编排的地址和信息在存储器中实际存放地址有何不同?

PC 机的 CPU 8088 具有 1 兆字节的存储寻址空间,需要 20 位地址,而其寄存器都是 16 位的,为此引入了分段的概念。CPU 对内存进行访问时,把内存分成若干段。段可以看成是一个连续的存储区,每一段最多 64K,必须从一个能被 16 整除的绝对地址开始,这 16 整除的地址称之为节,亦就是说,段地址必须以节为边界。

在具有地址变换机构的微机中,允许程序中编排的地址和信息在存储器中实际存放的地址不同,在 8088 系统中每个存储单元认为有两类地址:程序中所涉及到的地址是逻辑地址,一个逻辑地址由两部分组成:段地址和段内偏移量,对于给定逻辑地址的任何一

个存储单元来说,段地址决定了该段的第一个字节的位置,段内偏移量则是这个存储单元相对于该段首字节的距离。逻辑地址表示形式是:××××:××××,前面 4 位十六进制数为段地址,后面 16 位二进制数为段内偏移量。

8088 与存储器之间的所有信息交换却要使用物理地址。由于 8088 具有 20 根地址线,存储器的物理地址是一个 20 位二进制数值(00000H—FFFFFH),所以它与 1M 字节存储空间中和每个存储单元是一一对应的。20 位物理地址描述信息在存储器中实际存放的地址,也叫绝对地址。它的形成是首先将段寄存器内容左移 4 个二进制位,右边补上 4 个“0”形成 20 位段起始地址;再与 16 位段内偏移量相加,从而形成 20 位的存储器物理地址如下图所示:



图

7. PC 机系统中插入 8087 有什么好处? 如何处置?

在 PC 机系统中,为了增强其性能,可插入 8087 协处理器与 8088 一起工作,以提高数值运算的速度。

8087 协处理器有如下五个主要特点:

(1) 8087 与 8086/8088 微处理器一起工作控制它们之间的数据传送。8087 专门执行数值运算,对同样的数值,它的执行速度比 8086/8088 快约 10—100 倍,甚至更多。

(2) 8087 有 8 种数据类型:8 位、16 位、32 位、64 位字长的整数;32 位、64 位、80 位字长的浮点数;18 位十进制数,它的浮点数符合 IEEE 浮点标准。

(3) 具有 6 类数据传送、算术运算、超越函数、常数、比较、处理器控制共 68 条指令。

(4) 具有可寻址的 8 个 80 位专用寄存器堆栈。这 8 个 80 位寄存器,用于保存内部数据。

(5) 具有 6 种内部异常功能处理。8087 捕捉各种各样的运算错误,当出现错误时,8087 产生一个异常条件,这些异常条件是:无效操作、除数为 0、下溢出、未规格化操作数,上溢出、结果不精确。8087 捕捉到异常时,产生中断请求,要求用户程序进行处理。

PC 机运行时,8087 监视 8088 的工作,每当执行 8087 的数值运算指令时,8087 立即接过控制,执行指令规定的操作。操作结束后,交出控制,让 8088 继续运行。

在主机系统板上有一个 8 位系统配置开关 DIP,用来选择和设置系统配置的情况。只要将 DIP 开关的第 2 位开关拨到 OFF 位即可插入 8087 工作。(对长城 0520 把 8087 协处理芯片按缺口方向插入,再将第 1 号开关拨到 OFF 位置即可。)

8. PC 机系统的适配器是指一种什么设备?

PC 机通过各种外部设备与外界通信或交换数据,这种过程称为输入/输出。PC 机的外部设备主要有:键盘、显示器、打印机、软盘驱动器、硬盘驱动器、终端设备及其他专用设备。这些设备有机械的、电子的、机电的,它们传送的信号可能是数字信号,也可能是连续变化的电流或电压信号,而且信号电平有高低,信号传送速度也各不相同,当把这些外围设备接到 PC 机时,就需要相应的接口电路,亦称为适配器。

适配器的任务是把外围设备送往 8088 的数据转换成 8088 适用的格式,或把 8088 送往外围设备的数据转换成与外围设备相容的格式;向 8088 提供外围设备状态信息,如设备“准备好”、忙或闲、缓冲器“满”或“空”等等。协调 8088 与外围设备处理数据速度上的差异。一般说来,每种外部设备都有其特点,必须用专门适配器实现 8088 与外围设备间信号的转换与匹配,并提供适当的时序与控制信号,进行数据缓冲、同步协调,传递外围设备状态信号以及可编程设置工作方式等功能。

9. PC 机 8088 与外围设备间数据传送有哪几种方式?

一般有三种方式:程序控制的数据传送方式;中断传送方式;DMA 方式传送数据。

程序控制下传送数据时,在执行数据传送操作前,8088 先要检测外部设备的状态,如果外部设备正在工作(“忙”),则程序继续检测状态,8088 进行等待。如果设备准备好下一次传送,8088 就执行一次传送操作,如此循环下去,直接把数据传送完。这种传送的缺点是 8088 要不断查询外部设备是否“准备好”;当外部设备准备好时,8088 等待,无法执行其他操作,这就浪费了时间,对于低速外部设备,如键盘,其传送速度很慢,因而降低了 8088 的效率。为了提高计算机的效率,8088 与外部设备间传送数据通常采用中断方式或 DMA 方式。所谓中断是指计算机在正常执行程序的过程中,由于种种原因,使 CPU 暂时停止当前程序的执行,而转去处理临时发生的事件,处理完后,再返回去继续执行被暂停的程序。换言之,在程序执行过程中,插入另外一段程序运行,这就是中断。使用中断技术,使得外部设备与 CPU 不再是串行工作,而是分时操作,从而大大提高了计算机的效率,随着计算机的发展,中断不断赋予新的功能。中断系统已成为现代计算机不可缺少的组成部分。高效率的中断系统,可以最少的响应时间和内部操作处理所有外部设备的服务请求。PC 机利用中断功能,可使 8088 与外部设备并行工作,消除 8088 的等待时间,提高 8088 的利用率,同时 8088 可以同时管理多个外部设备的工作,提高输入输出数据的吞吐量。除此之外,还可利用中断技术进行实时处理,故障处理等。

PC 机 8088 具有简单而多用途的中断系统,每个中断都规定一个中断类型号(代码 0—255),8088 根据中断类型号从中断向量表中取中断服务子程序入口地址,转去执行中断处理子程序。

中断技术解决了高速 8088 与低速外部设备之间的矛盾,但对于高速外部设备,它就无能为力了。为了快速传输大量数据,PC 系统中采用存储器直接访问技术,简称 DMA。用 DMA 方式传送数据时,在存储器和外部设备之间,直接开辟高速传输数据的通道,不需要 CPU 8088 的干预,因而使数据传送速度达到最快。利用 DMA 方式传送数据时,数据的传输过程完全由硬件控制,这种电路叫 DMA 控制器。该控制器向 8088 申请 DMA 传送,在 8088 允许 DMA 工作时,处理总线控制权的转交避免因进入 DMA 工作而影响 8088 的正常活动或引起总线竞争,在 DMA 期间,管理系统总线,控制数据传送,确定数据传送的起始地址和数据长度,修正传送过程中的数据地址,在数据块传送结束时,给出表示 DMA 操作完成的信号。

10. IBM-PC/XT 及其兼容机系统内存存储空间是如何分配的

IBM-PC/XT 及其兼容机系统具有 1MB 存储空间。用十六进制表示的地址范围是 00000H—FFFFFH。这 1MB 存储空间划分为三个区域:RAM 区;保留区和 ROM 区。

存储器空间的低区,0—640K 的存储区域是读写存储器 RAM,对于 640KB 的系统 RAM,一般在系统板上安装 256KB,剩下的部分安装在 I/O 插槽中的存储器扩展板上。系统板上安装的 RAM 容量由系统配置中第 3 位和第 4 位状态来决定。运行 DOS 2.0, RAM 容量至少要有 128KB。接下来的 128KB 存储空间是系统保留用作字符/图形显示缓冲区的区域。对于不同的显示适配器,实际使用的存储器区域各不相同。IBM 单色字符显示适配器使用的缓冲区容量是 4KB 地址,为 B0000H—B0FFFH。IBM-PC/XT 中的彩色字符图形显示适配器使用的缓冲区容量是 16KB,地址为 B8000H—BBFFFH。如果使用高分辨率的显示适配器则显示缓冲区要使用更大的存储区域。

存储空间的最末 256KB 是系统的 ROM 区,这个区里安装的存储器都是只读存储器,其中前 192KB 的区域放系统中的控制 ROM 或 ROM 扩展板。一般高分辨率显示适配器的控制 ROM 安放在 C800H—C7FFFH 的区域内;硬盘驱动器适配器的控制 ROM 安放在 C800 开始的区域内。如果要安装用户固化在 ROM 中的程序,可以使用 ROM 区中尚未使用的地址。系统最后 64K 的存储器区域是基本系统 ROM 区。IBM-PC/XT 系统板上安装了 40KB 的基本 ROM,其中包括基本输入/输出系统 BIOS (8KB) 和 ROM BASIC (32KB), GW0520A 安装了 8KB 的基本 ROM。

计算机中的读写存储器 RAM 中存放的信息在停电关机后就消失了,为了使计算机初始化和进入正常工作状态,在计算机中必须安装不会因停电而丢失的初始化程序。初始化程序放在只读存储器中,加电后自动执行,对计算机进行初始化,再引导各种软件工作。

对于 IBM-PC/XT 系统板上安装的基本 ROM 40KB 占用地址是 F6000H—FFFFFH。其中固化 BASIC

占用地址 F6000H—F6FFFH;基本输入输出系统 BIOS 占用地址 FE000H—FFFFFH。BIOS 的功能对 PC/XT 系统进行初始化,同时也是高层软件和硬件之间的接口,它具有如下的功能:

(1) 系统冷启动和热启动;

(2) 系统自测试;

(3) 基本外部设备的输入输出驱动程序;CRT 显示器(单色和彩色)、键盘、打印机、软磁盘和异步通信接口等,这些驱动程序都要通过中断调用;

(4) 硬件中断管理程序;

(5) 系统配置分析程序;

(6) 字符图形发生器;

(7) 一天的时钟管理程序;

(8) DOS 引导程序。

系统板的 40KB 基本 ROM 信息装在两块 ROM 芯片中。一块是 8KB 的芯片,内装固化 BASIC 的前 8KB;另一块是 32KB 的芯片,内装固化 BASIC 的后 24KB 和 BIOS 8KB。

(待续)

中国微计算机单片微机 学会单片机函授班 (第二期)招生简章

一、宗旨

为适应“八五”计划要求,以高科技促国民经济的发展,加速单片机向社会生产和生活各个领域的渗透。根据学会理事会决议,在总结第一期函授的成功基础上,特在全国范围内举办第二期手脑并用,费用少、见效快的单片机函授班。

二、教学形式

以 MCS-51 单片机为学习内容,通过自学函授教材,阅读指定的辅导材料,要求人手一机,理论联系实际,定期完成指定的作业,习题和实验。学习中学员可随时反馈疑难问题,并定期作书面答疑。

三、函授对象

凡知识更新的各类工程技术人员,科教人员,管理干部,解放军军地两用人才,向高科技进军的技术工人,职业学校的师生,科技兴厂的乡镇企业技术骨干,自谋职业的知青等均中报名参加。

四、招生事项

报名日期:自简章公布之日起即可报名

报名办法:邮寄 2 元报名费,索取报名表、章程等。

报名时务请写明通讯地址、邮编号及联系方式。

开学日期:1992 年 5 月

报名地点:南京市 东南大学计算机系

联系人:孙育才,邮编:210018

五、办学单位

主办:中国微计算机单片微机学会

机电电子工业出版社《电子与电脑》编辑部

承办:南京电脑应用协会单片机专委会