

E & C

1993

一九九三年 ● 总期第103期

10

電子

ISSN 1000-1077

與 電腦



• ELECTRONICS AND COMPUTERS •

微计算机的下一场革命

多媒体技术



● 电子工业出版社 ●

微计算机的下一场革命——多媒体技术

内容简介:多媒体技术和多媒体计算机,近一二年来已悄悄进入国内市场,成为电脑界的热门话题。目前,在我国随着计算机性能价格比的不断提高,以及对其使用方面认识的逐步普及,计算机进入家庭的势头有增无减。如果当人们知道,在计算机内加上有关的多媒体卡和软件,还可以同录像机、摄像机、音响设备相互连到一起,可以随心所欲的进行自己的创作或编制出声情并茂的节目时,难道您不想了解多媒体技术吗?

本书较全面的介绍了多媒体的基础知识;各种多媒体卡和软件;多媒体制作和应用技术。

本书由电子工业出版社出版,全书 200 千字,大 32 开,定价:12.00 元 邮挂费:1.50 元
购书地址:(电子工业出版社广州科技公司邮购部)广州市石牌华南师大北区一号 203
邮编:510630 联系人:王丽端

国内代号: 2-888
定价: 1.60 元

DP851 独领风骚

“DP-851 单片机普及板”是一种模块式结构的 51 系列单片机系统。它有三种配置方式以实现不同类型的需要。

1. 单板型自开发器配置

主板+PC 机,可构成联机型单片机自开发器。可以实现汇编语言编程及调试。

主板+薄膜键盘板,可构成单板型单片机自开发器。可以实现机器码的编程及调试。

上述两种配置均具有置入、修改、调试、运行等多种自开发功能。

2. 基本配置(教学实验板)

基本配置是由开发器+实验板构成的教学实验系统。由于实验板上配置有进行实验用的接线柱(用绕褪工具绕接)、LED 显示器、蜂鸣器、DIP 开关、发光二极管,因此可以作 LED 显示、时钟、声音、交通灯、开关输入输出等常用教学实验。还可用主板上的六位 LED 显示器及键盘板上的 32 个键进行更为复杂的教学实验。

3. 扩展配置(目标应用板)

开发器+A/D,D/A 等选件板构成用户的各种目标系统。用户可通过 40 芯电缆连接各种应用板。

附表:邮购套件价格表

1. 单板型自开发器配置		2. 教学实验板配置		3. 扩展配置(选件)	
套件①:主板 +PC 机电缆和盘片		套件②:主板 +薄膜键盘板		套件③:自开发器 +实验板+辅助件	
				选件④:固化板 调试、扩展、A/D 板等	
主线路板 40 元	主线路板 40 元	主线路板 40 元	主线路板 40 元	普及型逻辑笔:20 元	
焊接件 38 元	焊接件 38 元	焊接件 38 元	焊接件 38 元	自升压固化板:190 元	
芯片 148 元	芯片 148 元	芯片 148 元	芯片 148 元	总线驱动调试板:60 元	
PC 联机电缆和盘片 25 元	薄膜键盘板 70 元	薄膜键盘板 70 元	薄膜键盘板 70 元	存储器扩展板:98 元	
说明书 3 元	说明书 3 元	说明书 3 元	说明书 3 元	三位半 A/D 板:120 元	
5V 电源 25 元	5V 电源 25 元	实验板 25 元	实验板 25 元	四位半 A/D 板:240 元	
		焊件和芯片 25 元	焊件和芯片 25 元	积分 12 位 A/D 板:350 元	
		40 芯电缆 5 元	40 芯电缆 5 元	高速 574A/D 板:980 元	
		绕褪工具 8.5 元	绕褪工具 8.5 元	A/D、D/A 板:320 元	
		电源 25 元	电源 25 元		
		包装盒 10 元	包装盒 10 元		
合计:279 元/套	合计:324 元/套	合计:397.5 元/套	合计:397.5 元/套		

注 1:以上价格不包含邮费。用户汇款时请加邮费(按套件×6%付费)

注 2:购买套件①的用户必须有 PC 机联机后才能使用。

注 3:套件①或②组成的自开发器可用 40 芯电缆外接各类扩展和实验板。

▲DP851 整机、散件,本厂独家批发。

DP851 总代理:北京西城电子仪器仪表厂

地址:北京西城区月坛南街 83 号

邮编:100045

电话:8512947

开户行:工商行北京百万庄分理处

帐号:014-047014-96

联系人:张文奇 庞文姬

电子工业出版社新书邮购消息

书代号	书 名	邮购价
B2270	FOX BASE 实用大全	20.70
B2380	AUTO CAD 使用大全	18.20
B2390	Turbo Pascal 大全	18.40
B3000	AUTO CAD 使用手册	25.30
B3001	RISC 技术参考大全	43.10
C0850	国内外流行摩托车电器原理维修图集	13.60
C0871	国内外流行汽车电器原理维修图集(1~6集)	99.50
C0990	实用空调制冷设备维修大全	18.60
C1111	家电维修技术精华大奖赛丛书(1~10册)	115.00
L1280	微机显示技术实用手册	69.00
L1290	英汉信息技术标准语词典	21.30
L1300	中国电子行业单位名片集	55.00
L1331	优质电子元器件实用手册(上、下册)	57.50
L1400	实用光电控制电路精选	17.80
N1140	《电子报》92年合订本	10.20
Z4002	《无线电》92年合订本	17.30

邮购办法: 邮局汇款: 北京万寿路电子工业出版社发行部邮购科

银行汇款: 开户行: 北京工商银行翠微路分理处

帐号: 661036—40 户名: 电子工业出版社发行部

邮编: 100036 电话: 8233693 电挂: 3101

注: 以上图书邮购途中丢失, 向我部索赔, 写清书代号、书名、详细地址、收书人姓名寄给我部。



电子与电脑

一九九三年总第 103 期

目 录

• 综述 •

家用微机及其市场开拓 周 宏(2)

• PC 用户 •

磁盘软格式化方法及其实现 高 洗(4)

巧用 DOS 中的 I/O 重定向命令 缪晓明(6)

如何将 WPS、dBASE III 的帮助信息打印出来 李 政(8)

WPS 文件阅读器 瞿新国(9)

不同规格双软配置大容量软盘间文件复制 彭 禾(10)

用 DEBUG 的 L, W 命令读写硬盘隐含扇区 王化丰(11)

C-WordStar 使用基础问与答 朱大公(11)

计算机病毒的检测、消除和预防 苏民生(13)

怎样对付计算机病毒

——介绍 WIPE 程序和硬盘初始化加密法 张 亭(15)

• 学习机之友 •

ProDOS 系统内部结构剖析(续) 廖 凯(19)

• FORTH 语言讲座 •

第十讲、第十一讲 丁志伟(22)

• 初、中级程序员软件水平考试辅导 •

链式存储及树形结构 夏晓东(28)

• 学用单片机 •

T6668 语音处理器手动实验 罗明克 车金相(31)

MCS-51 系列单片机中的加密措施 张 戟(34)

SCB-I 型单片单板机详介 (35)

• 电脑巧开发 •

CEC-I EPROM 编程电路 齐向东(38)

一种具有自动唤醒功能的掉电接口电路 陈敦琪(40)

• 维修经验谈 •

简易的苹果机故障检修方法 王惠民(41)

AR-3240 打印机“联机”灯闪烁故障检修一例 张全悦 张书琪(42)

Super AT286 运行金山 CCDOS 时帧频

不同步的解决办法 刘良团(43)

全国电子报刊联合征订简明目录表 (44)

出版软件简介 (45)

• 电脑通信 •

PC 机与 8031 串行通信的波特率选择 江 琪(47)

• 传真机专题讲座

传真通信的通路 张景生 张建军 辛亚西(48)

我国专用计算机通信网的组网方式 薛兴华(52)

ORACLE 关系数据库 杨 军(53)

• IC 电路应用 •

模拟乘法器 IC 及使用方法(二) 李兰友(54)

PLD 器件的新军——GAL 倪征宇(55)

• 读者联谊 •

在计算机上输入汉字要规范化 桂 英(56)

字符串译码编码器 (56)

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 特约编审：苏子栋

责任编辑：张 丽

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：1.60 元

家用微机及其市场开拓

重庆工业管理学院管理工程系(630050) 周 宏

随着家庭收入的逐渐增加和1992年微型计算机的大幅度降价,微型计算机在我国已经开始进入家庭。虽然微型计算机还远未成为一般的家用电器,但是其市场前景是极为乐观的。无论是微机厂商还是打算购买家用微机的家庭,都需要对家用微机市场有了解和认识。

一、家用微机及其需求的特点

从本质上讲,家庭微机与企事业单位用于工作事务的微机并不是两个概念。适合于家庭的微机,是微机中的中低档产品,这是由家庭使用的实际需要和家庭的经济能力所决定的。对家用微机的需求,在用途、微机系统的配置、价格等方面都有其特点。

(一)用途

国外的有关统计表明,家用微机大约30%的时间被用作娱乐,例如玩电子游戏等,而其他70%的时间被用于家庭数据管理、子女的培养教育以及成人的学习和工作。在我国,家庭购买微机也主要用于这四个方面:娱乐、家庭数据管理、学习和工作。

1. 娱乐 各种电子游戏深为少年儿童所喜爱,有的成年人也会迷上电子游戏,各种娱乐软件深受欢迎。家用微机可以成为良好的娱乐工具。

2. 家庭数据管理 一些家庭将用微机实现家庭数据的管理,例如家庭消费及有关事务的管理。个别的个体经营户也会用到家用微机。

3. 子女教育培养 对重视文化教育和子女才能培养的家庭,家用微机将成为子女智力开发和技能培养的重要工具。利用家用微机,既可以让子女学习关于计算机的知识,了解现代科技,又可以培养他们利用计算机这种科学工具,更好地学好其他的知识。

4. 成人的学习和工作 家用微机也是成年人学习和工作的有效工具。在现代社会中,掌握计算机的使用技能,会有越来越重要的作用。家用微机使工作人员在家中也能完成许多学习和工作事务,例如某些软件的编制、技术设计、资料整理、文字编辑等等。对于文字工作者,在家中用微机从事写作是极为方便的。我国一些作家在利用微机从事创作之后,极为赞叹微机带来的高效率。

(二)家用微机系统的配置

家用微机系统的适当配置可以从两个方面讨论:

1. 微机的主机 目前在我国流行且适宜于家庭的微机主机主要是两种:8088CPU(中央处理器)类型

和80286CPU类型。前者的内存储器容量为640K字节,后者为1M字节。8位的8088系列微机已属于淘汰产品,但是由于其价格便宜、性能可靠,也还有一定的市场。16位的80286系列微机比8088系列微机在性能上要高一个档次,而价格只稍高一点;286系列机的内存储量大,速度是8088系列机的四倍以上,可扩充性能好,是运行一些新型软件所必须的。除此之外,以32位的80386CPU构成主机的386系列微机,比286系列微机更高一个档次,其性能优越,但同时价格也较高。因此,从我国的国情看,286系列微机在三年内将是适宜于我国家庭的主流机种。再往后,286、386系列微机将共同成为我国家用微机的主流机种。

2. 外部设备 家用微机的最少外部设备配置,是键盘和软磁盘驱动器各一个,另外借助家用电视机作为显示器。这种配置节约了显示器的开支,是最大的优点,因此会受到部分消费者的欢迎。由于是最少的配置,又会产生诸多的不便,例如,微机的使用和电视的观看会发生矛盾,而电视的显示效果也较差。在最少配置的基础之上,还可以选择加单色或彩色显示器、再加一个软磁盘驱动器、加硬磁盘、加打印机等。彩色显示器、硬磁盘、打印机价格均在1200元以上,一般家庭购买力有限,暂时不会选用。所以可以预测在近几年,配置键盘、单色显示器和一至二个软磁盘驱动器的微机,会最受家庭的欢迎。

(三)价格

价格始终是刺激消费的最有力的因素,在人均收入较低的我国尤其是如此。1992年世界性的微机大降价,明显地刺激了世界微机需求的增长。1992年12月,我国将32位以下微机的进口关税下调30%,对缓解我国微机价格随其它物价的上涨而上涨,起了一定的作用。1993年我国如能加入关贸总协定,将是微机价格下降的因素。

配置单色显示器和一个软磁盘驱动器的286系列兼容机,目前的最低价格在2550元左右。如果其价格能降低于一台普通的彩色电视机,则将会有力地刺激消费。配有彩色显示器并加上硬磁盘的286系列兼容机,目前的最低价格在5500元以上,如果其价格居高不下,则难以进入普通家庭。

二、市场需求的潜力

我国有两亿多城市人口,约6千万个家庭。在彩色电视机、电冰箱等高档电器基本普及之后,一些家庭将

把注意力转向微型计算机,从而形成了家用微机市场需求的潜力。假设在6千万个家庭中将有1%的要购买微机,全国就有60万台的需求;如果这种需求平均分10年表现为购买行为,则每年就可以售出6万台家用微机。再假设家用微机每10年更新换代一次,则每年都可以售出6万台微机。其实,以上的估计是很保守的。在一些发达国家,仅大、中学生拥有私人微机的,就在10%以上。我国家用微机市场需求的潜力,极可能达到每年10万台,其数量相当巨大。

三、市场开拓

世界微机行业的一个大趋势,是竞相薄利多销。世界微机的价格还在进一步下降,据专家估计,在1993年内还将降价10—20%。从我国的实际情况看,考虑物价总水平的上升,国内的微机价格在今年内估计会趋于稳定。我国微机厂商有必要尽量降低成本,调整经营策略,遵循薄利多销的原则,结合我国家庭的消费水平,打开家用微机这一很有潜力的市场。

迄今为止,全国各地大多数百货公司都不经营微机产品。微机产品几乎都由一些高新技术企业经营,产品的销售点多数聚集在“科技一条街”,这无疑给微机

披上了神秘的色彩。大多数消费者很少看到关于家用微机的宣传,也很少看到市场上的微机产品,这必然严重影响微机向家庭的渗透。家用微机市场,是家用电器市场中尚未开发的一块沃土。微机进入家庭如同其它家用电器进入家庭一样,是社会发展的必然趋势。最先认识到这一点,并努力开发此市场的厂商,将会获益匪浅。在大力开发家庭用机产品的同时,需要象宣传其它的家用电器一样,大力宣传家用微机产品,普及有关的基础知识,造就社会舆论,同时也需要象对其它的家用户器一样,把微机产品送上百货公司的货架和橱窗,以吸引和争取消费者。在国外,已经在出现“象卖馅饼那样卖计算机”的现象,这种开拓市场的策略对我们是有启发意义的。

要开拓家用微机市场,还需要开发适合家庭应用的微机软件,包括娱乐性的软件、儿童和学生教育及学习的软件、家务管理性质的软件、以及工作性质的软件等等,以提高家用微机的利用价值。此外,还需要做好家用微机的售后服务等工作,在各个方面为微机进入家庭创造良好的条件,以成功地开发我国的家用微机市场。

LASER200—310 的 两种开发卡

一、Z80 开发卡主要用于 Z80 控制设备及简易数控机床的调试、维修及软件复制等。其功能:①DEBUG 系统包括 Z80 指令的汇编、反汇编、单步执行及设置断点等;②自备写入电源的 EPROM 复写功能;③LASER 机、中华机、TP801 单板机的磁带读写功能;④70 多条 BASIC 扩展功能。邮购价:280 元。

二、8751 开发卡:①具备上述全部功能;②8031 指令的汇编、反汇编、单步执行及在程序中设置断点等;③8751 写入;④32K 虚拟磁盘及其操作系统。邮购价:600 元。以上的邮费均 10 元。邮购地址:上海浦东新区龚路乡永利五队邮编 201209 联系人章永得。

把实验室搬进电脑,

红峰示波卡的宗旨!

武汉红峰电子工业公司

(430070)

Tel:027—700564

FAX:027—703396

磁盘软格式化方法及其实现

湖北远安 93 号信箱(444202) 高 洗

本文介绍的磁盘软格式化技术,可以减少对磁盘进行的大量格式化操作,从而避免不必要的磁头与磁盘的磨损,并减少因格式化所占用的宝贵时间。

据经验分析,一张软盘(或者是一个硬盘),在其寿命期内一般要被格式化五次以上,除第一次格式化是必须进行的一个步骤外,其余的基本上都是为了清除磁盘上不应有的数据而进行的。因此如果有一种方法可以去掉这些不应有的数据而又不必格式化整个磁盘的话,那么,后几次格式化就成为毫无必要。事实上,大

多数情况下只需要对 BOOT 区、FAT 区和 FDT 区进行操作即可获得与格式化整个磁盘相同的效果。为了区别于普通格式化操作,我们不妨称这种方法为磁盘软格式化方法。

本文介绍的软格式化程序主要针对软盘设计,考虑的对象为 360K 和 1.2M 软驱,对 1.44M 软盘和硬盘的格式化程序,可参考本文介绍的程序修改形成。特别是硬盘的软格式化一定要严格按照规定的格式编程。

程序清单如下:

```
CODE      SEGMENT
          ASSUME CS:CODE,DS:CODE
          ORG    100H
START:    JMP SOFTFMT
DRI_NUM  DB 0
MES1     DB 0DH,0AH,'SOFTFMT V1.0,BY GAO
          XIAN.,24H
MES2     DB 0DH,0AH,'DRIVER NAME NOT
          FOUND!',24H
MES3     DB 0DH,0AH,'DOS VERSION 3.2 OR
          LATER.'
MES4     DB 0DH,0AH,'READING OR WRIT-
          TING ERROR.',24H
MES5     DB 0DH,0AH,'SOFT-FORMAT OK!',
          24H
MES6     DB 0DH,0AH,'BEGIN FORMATTING,
          ARE YOU SURE?'
SOFTFMT: XOR,AX,AX
          MOV AH,30H
          INT 21H
          CMP AL,3
          JB VERR
          CMP AH,14H
          JAE BEGI
VERR:    MOV BX,1
          MOV AH,40H
          MOV CX,27
          MOV DX,OFFSET MES3
          INT 21H
          MOV AX,4C00H
          INT 21H
ERR:     MOV AX,4C00H
          INT 21H
BEGI:    MOV AH,40H
          MOV BX,1
          MOV CX,33
          MOV DX,OFFSET MES6
          INT 21H
          MOV AH,0
          INT 16H
          MOV AH,14
          INT 10H
          CMP AL,79H
          JE BEGI1
          CMP AL,59H
          JNE ERR
BEGI1:   CALL SUB_1
          MOV AH,9
          MOV DX,OFFSET MES1
          INT 21H
          MOV AL,BYTE PTR DRI_NUM
          MOV BX,OFFSET MAIN
          MOV CX,2
          MOV DX,0
          INT 25H
          JC EXIT1
          PUSH CS
          POP DS
          MOV BX,OFFSET MAIN+512
          CMP BYTE PTR [BX],0FDH
          JNE K1200
          MOV CX,5632
          MOV BX,OFFSET MAIN
          MOV AL,0
          MOV BYTE PTR [BX],AL
          INC BX
          LOOP CL0
          MOV BX,OFFSET MAIN+512
          MOV BYTE PTR [BX],0FDH
          MOV AL,BYTE PTR DRI_NUM
```



```

MOV BX,OFFSET MAIN
MOV CX,12
MOV DX,0
INT 26H
JC EXIT1
JMP EXIT2
K1200:  CMP BYTE PTR [BX],0F9H
JNE EXIT1
MOV CX,15360
MOV BX,OFFSET MAIN
MOV AL,0
CL1:   MOV BYTE PTR [BX],AL
INC BX
LOOP CL1
MOV BX,OFFSET MAIN+512
MOV BYTE PTR [BX],0F9H
MOV BX,OFFSET MAIN
MOV AL,BYTE PTR DRI_NUM
MOV CX,28
MOV DX,0
INT 26H
JC EXIT1
EXIT2:  PUSH CS
POP DS
MOV AH,9
MOV DX,OFFSET MES5
INT 21H
JMP EXIT3
EXIT1:  PUSH CS
POP DS
MOV AH,9
MOV DX,OFFSET MES4
INT 21H
EXIT3:  PUSH CS
POP DS
MOV AX,4C00H
INT 21H

SUB-1  PROC NEAR
CLD
MOV DI,80H
MOV CL,[DI]
SUB CH,CH
INC DI
CMP CX,1
JBE LOC-12
LOC-8: MOV AL,20H
REPNE SCASB
JCXZ LOC-12
MOV AL,[DI]
CMP AL,61H
JB LOC-9
SUB AL,20H
LOC-9:  CMP AL,63H
JAE LOC-12
CMP AL,40H
JBE LOC-12
INC DI
CMP BYTE PTR [DI],3AH
JNE RET-13
SUB AL,41H
MOV BYTE PTR DRI_NUM,AL
JMP RET-13
LOC-12: MOV AH,9
MOV DX,OFFSET MES2
INT 21H
MOV AX,4C00H
INT 21H
RET-13: RETN
SUB-1  ENDP
MAIN   DB ?
CODE   ENDS
END START

```

为了节省篇幅,程序中省略了软格式化后检查BOOT区是否被修改从而判断是否有病毒感染嫌疑的功能和建立系统盘的功能,读者可自行扩展。

SOFTFMT.ASM的用法很简单,编译、连接并转换为SOFTFMT.COM命令后,可作为DOS的普通外部命令使用。格式为:“C>SOFTFMT A.”或者:“C>SOFTFMT B:”。如果不指定驱动器名,则不执行格式化功能,提示后退出;若盘未硬格式化过或格式化信息被破坏,提示后退出。另外,在进入格式化之前还有一

次确认的机会,以免误操作丢失数据。由于SOFTFMT并不真正格式化所有磁道,所以万一误操作也还有可能进行适当的挽救,因为磁盘的文件数据区完好无损。这要牵涉到其它技术,故不在此叙述。

使用SOFTFMT命令,360K软盘的软格式化时间消耗大约是硬格式化的三十五至四十分之一;1.2M软盘的软格式化时间消耗大约是硬格式化的七十五至八十分之一;硬盘的效率就更高了。这样就既节省了时间,又延长了软盘和驱动器的寿命,读者不妨一试。

巧用 DOS 中的 I/O 重定向命令

河南中原油田石油学校(457001) 缪晓明

本文介绍 DOS 手册中一个大多数人未予重视的命令,即 I/O 重定向命令,结合其它 DOS 编程命令给出了四个应用实例,即使用 I/O 重定向命令修改程序,调试、修改汇编语言程序,显示内存容量,备份硬盘结构信息。

PC DOS 版本 2.0 以上都支持一个类似于 Unix 系统的 I/O 重定向功能,如果用户能熟悉它并能结合 DOS 的其它功能加以灵活运用,将会更加方便有效地使用微机,提高应用水平。

I/O 重定向

DOS 的输入输出通常是在标准设备(键盘和显示器)上进行的,但也可重新定义为另外的设备。此设备可以是磁盘文件或设备文件,也即可以改变输入输出的方向。DOS 承认的常见设备文件有:CON(控制台,即键盘/显示器),AUX(异步通信口),PRN(并行打印机),NUL(虚拟设备)等等。

(1)输出改向 输出改向可以在命令中键入一个大于号“>”加上输出文件名或设备名即可。如果该输出文件已经存在,则将清除该文件并进行复写。格式是:

```
>[d:] [path] filename
```

使用两个连续的大于号“>>”,可用来添加一个文件。如果该文件已存在,则输出将附加于原文件之后,不会破坏原文件。若文件不存在,则该命令类似于上述命令。

```
A>dir b.*.exe>b:dirs.txt
```

```
A>dir b.*.com>>b:dirs.txt
```

这两个重定向的列目录命令,可产生一个目录文件 dirs.txt,包括扩展名为 .com 和 .exe 的文件目录。进而用命令:

```
A>type b:dirs.txt >prn
```

```
A>copy b:dirs.txt >lpt1;
```

这两个命令达到的效果是一样的,即重定向输出设备为并行打印机。

(2)输入改向 类似于输出改向,在命令中使用小于号“<”及输入文件名,可进行输入改向。格式是:

```
<[d:] [path] filename
```

使用实例

例 1:利用重定向功能修改程序

在软件移植过程中经常要对原有软件进行修改,特别是软件的 I/O 接口部分。这类修改的特点是:难度不高,工作量大,需要非常细心,又很有规律。我单位有一个用 dBASE III 编写的人事工资管理系统,原运

行环境为 PC/XT 机,汉字 CC DOS 2.10 版,现需改在 286 机的高分辨率环境下工作。由于制表符区位的差异,使 30 个报表程序中的制表符都需要进行修改。如用 EDLIN 程序处理,虽然可以用字符串替换功能,但由于 EDLIN 只提供命令方式而不支持程序方式,若每个程序有 20 种制表符需要修改,则共需 $30 \times 20 = 600$ 次的字符串替换命令。用重定位功能可以实现修改过程的自动化。方法如下:

(1)用 EDLIN 或 COPY CON:AA.BAT 建立一个名为 AA.BAT 的批命令文件,其中插入了以下形式的命令:

```
FOR %%F IN (*.PRG) DO EDLIN %%F<XG
```

其中 XG 是一个任取的文件名,其内容是 EDLIN 程序的命令序列。在本例中如下,可用 EDLIN 自身建立。

```
1,R <原字符串 1> ^ Z<替换字符串 1>
```

```
1,R <原字符串 2> ^ Z<替换字符串 2>
```

```
...
```

```
...
```

```
1,R <原字符串 N> ^ Z<替换字符串 N>
```

```
E
```

(2)现在程序的修改工作只需在 DOS 提示符下运行 AA.BAT 即可一次修改完所有的程序。用类似的方法可以修改各类程序,只要对 XG 文件进行细致的设计,就可以获得满意的结果。

例 2:用 DEBUG 的 U 命令可以很方便地进行反汇编,但是如果没有汇编语言源程序。又需要修改源程序,尤其是插入修改,一般是比较麻烦的。但实际上只要有扩展名为 .COM 和 .EXE 的可执行文件,再结合重定向命令就方便多了。如有一个 T.COM 文件,长度为 200H 字节。键入以下命令:

```
A>DEBUG T.COM>T.ASM
```

```
-u 100 3ff
```

```
-q
```

T.ASM 就是 T.COM 的机器码和反汇编结果。你可以用 EDLIN 等编辑软件调用 T.ASM,进行分析和打印,修改时将其中的机器码去掉,借助于编辑软件进行插入、替换等修改。工作完成后,将第一行的 U 命令改为:

```
f 100 ffff 0
```

```
a 100
```

存盘,然后键入命令:

```
A>debug<t.asm
```

在退出 DEBUG.COM 后,用命令方式进入 DEBUG 环境中,再用 D 命令查看文件长度(最后一个非

零字节的地址减去 100H), 设置寄存器 BX, CX, 再将汇编好的内容以文件名 T.COM 存盘即可。

例 3: 在 AUTOEXEC.BAT 中显示内存容量, 加强对病毒的监视能力。

许多病毒常修改内存总量(减少 1~2K), 使病毒程序常驻内存。PC 机的内存总量参数存放在 0:0143H 和 0:0414H 单元中, 只要知道微机的实际容量, 对比一下就可以判定内存中是否染上病毒。这种方法对于那些操作系统型病毒、感染 COMMAND.COM 的外壳型病毒以及利用修改内存总量隐藏自己的病毒都很有效。一旦发现参数不符可及时采取措施。由于 0:0143H 和 0:0414H 两单元存放的是基本内存参数, 所以对一些扩展了内存的 AT, 286, 386 机同样适用。常见的内存配置和两单元的对应关系如下:

实际内存	0:0413H	0:0414H
640K	80H	02H
512K	00H	02H
384K	80H	01H
256K	00H	01H

在批文件中加入以下命令: DEBUG<DEBUG.DAT

其中 DEBUG.DAT 是 DEBRG 程序的命令序列, 建立方法同例 1, 内容为:

```
D 0:413 L 2
Q
```

例 4: 微机用户都知道象 DOS 系统盘是很重要的系统软件, 需要做好备份。实际上对拥有硬盘的用户来说, 硬盘的结构信息(包括硬盘引导区 BOOT, DOS 分区引导区 DOSBOOT, 文件分配表 FAT, 文件目录 FDT)也是非常重要的系统软件。尤其是现在许多恶性病毒发作时都破坏硬盘的结构信息, 从而达到破坏整个硬盘数据的目的。经常备份硬盘的结构信息可以最大限度地减少病毒造成的数据损失, 对于引导型的病毒也可以起到消毒的作用。

以 DOS 3.3 管理的 43M 硬盘 C: 为例, 假设硬盘结构信息的四部份分别保存到文件 BOOT.DOS, DOSBOOT.DOS, FAT.DOS, FDT.DOS 中。实现保存的批处理文件 BACK.BAT 内容如下:

```
@ECHO OFF
IF NOT EXIST *.DOS GOTO A
ATTRIB -R *.DOS
IF NOT EXIST *.BAK GOTO B
ATTRIB -R *.BAK
DEL *.BAK
:B
RENAME *.DOS *.BAK
:A
DEBUG<BACK.DAT>NUL
ATTRIB +R *.DOS
ATTRIB +R *.BAK
```

ECHO BACKUP SUCCESSFUL!

实现恢复的批文件 REST.BAT 内容如下:

```
@ECHO OFF
IF EXIST *.DOS GOTO B
ECHO DOS INFORMATION NOT FOUND!
GOTO END
:B
DEBUG<REST.DAT>NUL
ECHO RESTORE SUCCESSFUL!
:END
```

其中 BACK.DAT 和 REST.DAT 为 DE-BUG.COM 的命令序列。它们的内容为:

BACK.DAT	REST.DAT
A 100	A 100
MOV AX,0201	MOV AX,0310
MOV BX,1000	MOV BX,1000
MOV CX,0001	MOV CX,0001
MOV DX,0080	MOV DX,0080
INT 13	INT 13
INT 3	INT 3
G=100	N BOOT.DOS
N BOOT.DOS	L 1000
R CX	G=100
200	N DOSBOOT.DOS
R BX	L 100
0	W 100 2 0 1
W 1000	N FAT.DOS
L 100 2 0 1	L 100
N DOSBOOT.DOS	W 100 2 1 55
W 100	W 100 2 56 55
L 100 2 1 55	N FDT.DOS
R CX	L 100
AA00	W 100 2 AB 20
N FAT.DOS	Q
W100	
L 100 2 AB 20	
N FDT.DOS	
R CX	
4000	
W 100	
Q	

注意: 使用时需将文件 DEBUG.COM, ATTRIB.COM, BACK.BAT, REST.BAT, BACK.DAT, REST.DAT 都放在同一软盘上。

用户在做好备份后, 可用 COMP 命令比较 BOOT.DOS 和 BOOT.BAK, 以及 DOSBOOT.DOS 和 DOSBOOT.BAK 是否一致。若原机器无毒, 比较结果两文件不一样, 则说明有病毒侵入。同时也可结合例 3 来判断。发现有病毒后, 解毒方法很简单: 在无毒环境下, 将备份好的硬盘结构信息用 REST.BAT 恢复即可。注意: 这时恢复要用 *.BAK, 因为 BOOT.DOS 和 DOSBOOT.DOS 可能已经包含着病毒的程序。

如何将 WPS、dBASE III 的帮助信息打印出来

四平师范学院计算中心(136000) 李 政

中西文字处理软件 WPS 和中西文关系数据库管理系统 dBASE III 在我国流行十分广泛,深受广大用户的喜爱。这两种软件都配有中文在线帮助功能。工作时只要按帮助键,就可以在屏幕上显示需要的帮助信息。这些帮助信息设计得非常精练而有条理。如果能把它们的全文打印出来,不失为一个精巧的使用手册。

这些帮助信息用通常的方法(如,打印屏幕、Ctrl + P)是打印不出来的,本人通过对这些帮助信息文本存放格式的分析,找到了比较有效的方法。其基本思想是:①确定帮助信息所在的文件;②从该文件中过滤出帮助信息文本,而将其它非文本信息删除或转换,最后形成一个可被一般字处理软件所接受的文本文件;③对以上形成的文本文件,用字处理软件(如 WPS)进行编辑和打印。

下面就这两种软件分别介绍具体实现方法:

一、打印 WPS 帮助信息

1. 确定帮助信息所在文件

WPS 1.2 版有四个文件:WPS.CFG、WPS.EXE、WPS1.OVL 和 WPS2.OVL。为了确定帮助信息在哪个文件中,可用 PC Tools 文件服务的 Find 功能,在上述四个文件中查找帮助信息的特殊关键字(比如:“一、键盘编辑控制”)。

注意,在西文 PC Tools 下查找中文关键字,需要先知道对应的内码,然后再查找内码。具体步骤如下:

①用 EDLIN 建立一个含有特殊关键字的文件

AAA:

EDLIN AAA

New file

* I

1 * 1、键盘编辑控制

2 * ^ Z

* E

②用 DEBUG 显示对应的内码

DEBUG AAA

-D

XXXX:0100 D2 BB A1 A2 BC FC C5 CC B1 E0 BC AD BF

D8 D6 C6

-Q

③进入 PC Tools,选择上述四个文件,用 Find 功能查找内码“D2BBA1A2BCFCC5CCB1E0BCADBFD8D6C6”所在的文件,可以确定帮助信息在文件 WPS1.OVL 中。

2. 取出帮助信息文本

①复制 WPS1.OVL 的副本

在 DOS 下,用

COPY WPS1.OVL PPP

复制 WPS1.OVL 的一个副本 PPP。PPP 经处理后,最终做为帮助信息的文本文件。

②确定帮助信息始址

DEBUG PPP

-S 100 FFFE D2 BB A1 A2 BC FC C5 CC B1 E0 BC AD BF

XXXX:771C (该关键字从偏移地址 771C 处开始)

-RCX

CX 9088

:2000

(规定字节数,略大于实际文本长度)

-W 771C

(从 771C 开始存盘)

-Q

3. 编辑打印文本文件

进入 WPS,编辑文件 PPP。由于 WPS 的帮助信息文本中间不含控制码,而且整个文本集中存放在一起,所以,编辑工作十分简单,只要将开头和结尾处的一些无用信息删除,再按要求进行排版,便可打印输出。

二、打印 dBASE III 帮助信息

与 WPS 相比,dBASE III 具有更强的帮助功能。帮助信息包括六大类:1. 开始;2. 什么是...;3. 怎么做...;4. 建立数据库;5. 使用现有的数据库;6. 命令和函数。这些信息经过整理打印,便是一本实用手册。

1. 确定帮助信息所在文件

dBASE III 系统有四个文件:DBASE.EXE、DBASE.OVL、HELP.DBS、ASSIST.HLP,帮助信息的文本在文件 HELP.DBS 中。

2. 滤出帮助信息文本

dBASE III 帮助信息文本的存放格式不象 WPS 那样简单,整个文本分成几块,每个文本块之中以及各文本块之间都含有许多控制码。因此,需要首先将这些控制码删除或转换,才可滤出有用的文本。

①复制 HELP.DBS 的副本

在 DOS 下,用

COPY HELP.DBS KKK

复制 HELP.DBS 的一个副本 KKK。KKK 做为一个“毛坯”,最后加工成帮助信息的文本文件。

②删除各文本块之间的控制码——“初加工”

这一工作可用 DEBUG 来做,但比较麻烦。如果用 PC Tools 的 Wordp 功能就简单多了,因为 PC Tools 的 Wordp 可以接受非文本代码。

用 PC Tools 的文件服务功能,选择文件 KKK,再用 Wordp 功能,删除 KKK 中的无用信息块(包括控制码),保留帮助信息文本块,操作时,需要用到:

F5(select) 选块

不同规格双软配置大容量软盘间文件复制

四川省涪陵地区卫生局(648000) 彭 禾

国内 286/386/486 微机大都为双软盘驱动器配置,规格以 360KB 及 1.2MB 各一台,1.2MB + 1.44MB 组合最为常见。用户每反映:大容量软盘之间复制文件不直达,借助硬盘(子目录)或虚拟盘中转,既费时又耗机。

众所周知,在 DOS 环境中,各种大容量软驱的正常运行需有不低于相应版本操作系统的支持。DOSV 3.00 启用 1.2MB 软驱;720KB 3.5 英寸软驱需 DOS V3.20;DOS V3.30 开始支持 1.44MB 软驱;2.88MB 软驱则有赖于 DOS V5.0。

目前,配置大容量软驱的微机大都安装 DOS V3.20 以上版本的操作系统。1985 年推出的 DOS V3.20 启用逻辑软驱设备驱动程序——DRIVER.SYS,支持在规格不同双软驱配置中实现单软驱大容量软盘间文件或文件组的复制,犹如‘增加’等容量软驱,从而摆脱中转盘。

DRIVER.SYS 设在 CONFIG.SYS 中,命令格式如下:

```
DEVICE=DRIVER.SYS /D:d [/T:tt] [/S:ss]
[/H:h] [/C] [F/f]
```

开关值随版本升级而增加,开关用字母略有不同。

/F 与/T+/S+/H 组合等效,可两者取其一

/D:d d 为命名或添逻辑名软驱序号

第一软驱——0;第二软驱——1;第三软驱——2

/T:tt tt 为每面磁道数(默认值 80)

360KB 软盘——40;1.2MB、720KB、1.44MB

2.88MB 软盘均为 80

/S:ss ss 为每磁道扇区数(默认值 9)

360KB 软盘——9;1.2MB 软盘——15;

720KB 软盘——9;1.44MB 软盘——18;2.88MB

软盘——36

/H:h h 为磁头数(默认值 2),五种软盘机均为 2

/C 专用于 1.2MB 软驱,表示具有磁盘更换检测功能

/F:f f 为磁驱规格代号(默认值 2)

360KB 软驱——0;1.2MB 软驱——1

720KB 软驱——2;1.44MB 软驱——7;2.88MB 软驱——9

实例

SJ 286 微机第一软驱 1.44MB,第二软驱 1.2MB,硬盘 40MB,含逻辑盘 D,安装 PC DOS V3.30,其 CONFIG.SYS 如后:

```
SHELL=C:STA\COMMAND.COM/P [置入子目录防病毒感染]
```

```
COUNTRY=002,,C:STA\COUNTRY.SYS
[启用中国式日期]
```

```
DEVICE=C:STA\DRIVER.SYS /D:0 /F:7
```

[添第一软驱逻辑名]

```
DEVICE=C:STA\DRIVER.SYS /D:1 /F:1/C
```

[添第二软驱逻辑名]

```
DEVICE=C:STA\VDISK.SYS 384 512 112/E
```

[设置虚拟盘(XMS)]

```
DEVICE=C:STA\ANSI.SYS
```

[支持 CCBIO 2.13H]

```
FILES=34
```

```
(AUTOEXEC.BAT 含 SET COMSPEC=C:
STA\COMMAND.COM)
```

引导后显示:

Loaded External Disk Driver for Driver E

Loaded External Disk Driver for Driver F

VDISK version 3.30 virtual disk G:

Transfer size adjusted

Buffer size: 384 KB

Sector size: 512

Directory entries: 112

Transfer size: 8

其中 E 为第一软驱逻辑名, F 乃第二软驱逻辑名, G 为虚拟盘名。

讨论

一、软驱添逻辑名后,相应容量的软盘间复制文件或文件组等操作即可在单台软驱上顺利进行,如本例执行 A>COPY DUP.EXE E:,将 1.44MB 或 720KB 软盘上 DUP.EXE 复制到另一张 3.5 英寸软盘。

二、添逻辑名软驱执行单驱 DISKCOPY 如常,例: B>[C:\DOS\]DISKCOPY,即执行 B 驱动器中,1.2MB 软盘全盘复制,提示照旧,不涉及逻辑名。

三、DOS V 3.20 CONFIG.SYS 中,DRIVER.SYS 与/D:d,以及开关与开关之间均需有一空格。否则显示出错信息如下:

```
Bad or missing Driver.sys /D:0 /F:f
```

```
Bad or missing Driver.sys /D:1 /F:1 /C
```

四、DOS V3.0 系列 CONFIG.SYS 中 LASTDRIVE 默认值为 E(5),指物理软驱和硬盘总数,它不涉及虚拟盘名及软驱逻辑名的数目。微机外存储器少于超过五台,一般不另列配置命令。

五、DRIVER.SYS 仅适用于软驱,为硬盘(子目录)设逻辑名当用 SUBST 命令,其指定的字母与 LASTDRIVE 设置有关。如指定 H:CONFIG.SYS 中

需含 LASTDRIVE=H,并回避已用的逻辑名,如本例之 D,E,F,G。

六、设置第三软驱的命令是 DEVICE=DRIVER.SYS /D:2 /F:f,此命令行应设在其它 DEVICE=DRIVER.SYS /x:n /x:n 之前,如接着设第二条相同

的命令,可为之添逻辑名。以 PC DOS V3.30 设置第三软驱,尚需在上列设置命令行之前另加专用设备驱动程序 EXDSKBIO.DRV 命令行,AST Premium 系列则另有前置性专用设备驱动程序——ASTDSK.DRV,均详见使用手册。

用 DEBUG 的 L,W 命令读写硬盘隐含扇区

大连理工大学 8 号信箱(116023) 王化丰

熟悉 DEBUG 的人都知道其读写磁盘扇区的命令格式是: -L addr d m n

-W addr d m n

addr——读入内存的首地址;

d——逻辑驱动器号:A=0,B=1,C=2,...

m——首逻辑扇区号;

n——读写的扇区数目。(≤80H)

一般,m 从 0 开始直到所读写盘的最后一个逻辑扇区号,n 从 1 到 80H。这样对于有隐含扇区的硬盘来说,似乎就无法用 L,W 来读写其隐含扇区,而要编写读写盘的小程序了。其实不必,可以用补码的方式读其隐含扇区,只是要一次全部读出才行。具体作法如下:

假设硬盘 C:,则进入 DEBUG 后,在其提示符下,如下操作:

-l 100 2 0 1

-d 100 120

××××,0100 EB 34 90 4D 53 44 4F 53 -33 2E 33 00 02 04 01 00 .4.MSDOS3.3...

××××,0110 02 00 02 3B A2 F8 29 00 -11 00 04 00 11 00 00 00

(此逻辑盘隐含扇区数为:00 11 H,则其逻辑扇区号为-00 11H.)

-h ffff 0011 (求-0011H的反码得 FFEE H)

0010 FFEE(-0011 H 的补码等于反码加 1,即为 FFEE H)

-l 100 2 ffef 11(用-0011H的补码读 11 H 扇区)

-W 100 2 ffef 11(写 11H 扇区)

有的大容量硬盘,其每条磁道上可达 21H 个扇区,对于这样的逻辑盘其隐含扇区一般为 21H 个,则可类似操作:

-l 100 2 0 1

-d 100 120

××××,0100 EB 34 90 4D 53 44 4F 53 -33 2E 33 00 02 04 01 00 .4.HSDOS3.3...

××××,0110 02 00 02 47 F4 F8 3D 00 -21 00 08 00 21 00 00 00 ...G...=...1 ...

-h ffff 0021

0020 FFDE

-l 100 2 ffd1 21

-w 100 2 ffd1 21

-q

此法也适用于对逻辑 D:,E:等盘隐含扇区的读写。使用此法写盘时千万小心,因为是多扇区写盘,一定确保写入磁盘的正确位置以正确的内容,否则可能会破坏原盘主引导区等重要部分的信息。

C-WordStar 使用基础问与答

新潮计算机集团公司(610051) 朱大公

17. 使用 ^KB、^KK 和 ^KH 命令应注意些什么?

首先,本块的起始标志(用 ^KB 定义)和终止标志(用 ^KK 定义)二者的位置不能颠倒,起始标志在前,终止标志在后。其次,起始标志和终止标志二者不可缺一,当然更不能不定义。第三,操作者定义起始标志和终止标志的先后次序可以任意,即先定义起始标志还是先定义终止标志是无关紧要的,只要保证起始标志在前,终止标志在后就行了。第四,在编辑的整个文本中只能定义一个文本块,也就是说只能出现一对 、<K>。如果重复定义,C-wordStar 怎样处理呢?当出现这种情况时,C-wordStar 只承认操作者最

后一次使用 ^KB 和 ^KK 所定义的文本块,原来定义文本块时出现的块标志 、<K> 自动消失。第五,当需要缩小或扩大原来定义的文本块时,如果文本块的起始位置和终止位置二者中有一个不需要改变,则可以只重新定义需要改变的那一个块标志就行了。第六,当完成了所需的块操作后,一般应该使用一次 ^KH 命令隐去块标志。初学者最容易犯的错误之一就是不定义文本块便实施块操作。如果实施块操作没有达到预期的目的,确因没有定义文本块,则屏幕上会给出相应的提示。但情况往往是,上次块操作后没有用 ^KH 隐去块标志,本次操作前又不重新定义文本块,发出块操作命令后,块操作顺利执行完毕,结果却出现了

错误,特别是用 ^ KY 命令误将大段文本删除,损失就大了。因此,一定要养成实施完块操作后立即使用 ^ KH 隐去块标志的良好习惯。最后,当前文本存盘后,盘上的文件不保留块标志,如果重新调入盘上的文件,实施块操作前必须重新设置块标志。

18. 怎样实现内存中的当前文本与磁盘上的文件交换信息?

实现内存中的当前文本与磁盘上的文件交换信息主要靠 ^ KW 和 ^ KR 这一对块操作命令。

^ KW 将正编辑的文本中的文本块写到磁盘文件去,建立一个新的磁盘文件。使用该命令前要先给文本块打上块标志。发出 ^ KW 命令后,屏幕会提示操作者输入一个文件名,然后回车。完成上述操作,当前文本中的文本块便保存到操作者命名的文件中了。如果操作者给出的文件名与磁盘上的某文件同名,C-WordStar 会询问操作者是否重写该文件,回答 Y 表示重写,即用当前文本块的内容覆盖文件中的老文本;回答 N 表示不重写,此时应重新输入一个文件名。最后说明一点,^ KW 命令执行完毕,正在编辑的文本、屏幕状态等均保持使用该命令前的状况。

^ KR 命令把磁盘文件上的文本读到正在编辑的文本中,其具体位置是插入到当前文本的光标所在处前。

下面作几点说明。首先,^ KR 是重写磁盘上的同名文件。因此磁盘上原文件的内容全部被已定义好的文本块的内容所覆盖,不可能实现追加操作。要想实现追加操作,只能采用变通的方法。其次,并不是只有 ^ KW 形成的磁盘文件才能由 ^ KR 读入到当前文本中,其它方式如 ^ KD、^ KS、^ KX 等所生成的磁盘文件同样能用 ^ KR 读入到当前文本。第三,^ KW 和 ^ KR 配合使用,实现 ^ KC 的功能,在某些情况下显得更方便、更安全。例如,当我们正在编辑一篇文章时,该文章中有一大段文字要重复使用多次,我们可以先使用 ^ KW 命令将该段文字写到磁盘上,需要时用 ^ KR 命令将其调入当前文本即可。使用 ^ KC 当然可以实现上述功能,但必须随时注意查看块标志,稍有不慎便可能把不需要的内容拷贝过来,这当然是我们不希望的。第四,使用 ^ KR 命令可以实现多个文件的组合,这在一份文件很长,分配给几位录入员录入时经常采用。具体作法是,首先让每一位录入员将自己录入的文本存盘,形成磁盘文件,然后用 ^ KR 命令依次读入这些磁盘文件,最后用 ^ KD 等存盘命令存盘便可以收到一份包括全部文本的磁盘文件了。与此同理,配合使用 ^ KW 和 ^ KR 命令也可以实现追加操作。最后提醒初学者注意,使用 ^ KW 前一定要先定义文本块或检查块标志。^ KR 虽归入块操作,但它是输入磁盘文件,与是否定义块标志无关。

19. 怎样删除大段文本?

删除大段文本虽然可以采用诸如 ^ G 等删除单个字符的命令,但操作频繁,原则上应使用块操作中的 ^ KY 命令。使用 ^ KY 命令前先将要删除的文本定义为

一文本块,然后发 ^ KY 命令,此时被定义的文本块便全部被删除。该操作具有一定的危险性,只要文本块已定义好,且未用 ^ KH 隐去块标志,一旦发出 ^ KY 命令,系统不提示是否确实要作删除操作便立即执行,如果是误发生 ^ KY 命令,损失便无法挽回,故使用该命令应慎之又慎。

20. 什么叫复制当前文本? 怎样实现复制当前文本?

所谓复制当前文本就是把当前文本的一部份或全部复制到光标所指定的位置处,源地址和目标地址处均出现所复制的文本。要实现当前文本的复制,除了可以配合使用 ^ KW 和 ^ KR 外,常用的方法是使用 ^ KC 命令。使用 ^ KC 命令前要做两件事,一是把待复制的文本定义为一文本块,一是把光标移到目标地址处。做完这两件事后发出 ^ KC 命令,待复制的文本便复制到光标所在处,具体地说是插入到光标之前,原来的文本向右推移。^ KC 是 C-WordStar 设置的正规的文本复制命令,使用频率极高,但同一内容的文本块需要多次复制时,建议大家使用 ^ KW 和 ^ KR 命令。

21. 何谓移动当前文本? 怎样实现移动当前文本?

所谓移动当前文本就是把当前文本中的一部分或全部移动到光标所指定的位置处,仅在目标地址,即光标所在处出现移动过来的文本,源地址处的文本消失。移动当前文本和复制当前文本功能上的主要差别就是源地址处的文本是否消失。实现移动当前文本的方法是使用 ^ KV 命令。^ KY 命令的具体操作方法是,先定义好待移动的文本块,然后将光标移至目标地址,最后发 ^ KY 命令。

22. 文本块太大时,如何实现相应的块操作?

C-WordStar 对文本块的大小没有严格的定义,但在实施块操作时却有可能出现文本块太大的出错提示信息,此时块操作不能正常进行。为了实现相应的块操作,变通的方法还是有的,这就是重新定义文本块,将待操作的文本块分小,一次对一较小的文本块进行操作,经两次或多次同样的操作,便可以实现特定的块操作功能了。

23. 当使用存盘命令保存当前文本时,有时会提示盘空间不够,但经检查发现剩余盘空间仍大于所需保存的文件的长度,这是为什么?

C-WordStar 工作时所需的磁盘空间大致为正在编辑的文本的长度的 3 倍。其中,一部分为当前文本存盘时形成磁盘文件使用;另一部份为生成后备文件。BAK 使用;第三部份为临时文件使用。因此,当盘剩余空间大于当前文本的长度,但又不是够大时(约为当前文本长度的 3 倍)便可能出现盘空间不够的出错提示信息。下面对后备文件和临时文件作一简单说明。后备文件是文件名后缀为 .BAK 的一种文件,其意义和用法与 DOS 下很多其它实用程序生成的 .BAK 文件很类似。当我们在 C-WordStar 下编辑已经存在的文件时,如果该文件的文件名为 FILE.WS,那么编辑完成后存盘时(或尚未编辑完成,中间实施存盘操作时),被

编辑过的文件使用文件名 FILE.WS,而原来盘上的文件则被改名为 FILE.BAK。由此可见,.BAK 文件是当前文件最新版本的前一版本。当操作者认为最新版本还不如前一版本满足使用要求时,则可以放弃最新版本转而使用前一版本,当然对文件名后缀.BAK 要作修改。这就是.BAK 文件的主要用途。临时文件以. \$\$\$ 为文件名的后缀,它是 C-WordStar 工作时产生的。例如,执行 ^QR 命令时,如果文件太长,C-WordStar 要对其进行分解,生成以. \$\$\$ 为后缀的临时文件,^QR 命令执行完毕,. \$\$\$ 文件由系统自动将其删除。由此可见,. \$\$\$ 文件的生成与消失对用户是完全透明的,但它却实实在在地要使用一定的盘空间。

24. 用 C-WordStar 编辑文本时,经常要在已编辑过的文本中插入一些内容,这些内容在当初编辑文本时已考虑到了,但一时又无法确定它们的确切位置,有无一种办法帮助我们在确定插入位置后迅速地找到这一位置呢?

回答是肯定的。由于这里所提到的现象具有普遍性,所以 C-WordStar 设置了一对很有用的命令 ^Kn 和 ^Qn 来解决这类问题。

下面先对 ^Kn 和 ^Qn 两个命令作一些说明。

^Kn 命令的功能是在当前光标处设置位置标志。几是数字 0~9 中的一个。使用该命令后,光标处会出现标记 <n>。由于 n 有 10 个可能的值,所以每一文本最多可设置 10 个位置标记。

^Qn 命令的功能是使光标返回位置标记为 <n> 处。显然,n 的数值范围也是整数 0~9

使用 ^Kn 和命令时需要注意以下几点,首先,只有使用 ^KS 命令存盘,所设置的位置标志才会保留下来,使用其它命令存盘将使其丢失。其次,不要担心会给文本设置上相同的位置标志,因为 ^Kn 命令在 n 相同时只承认最后一次设置,即 n 相同时,多次使用 ^Kn 命令,只有最后一次有效。第三,当光标置于位置标记 <n> 右边首字符的位置时,使用 ^Kn 命令会使标记 <n> 隐去,此后只要没有使用相同的 n 来发 ^kn 命令,则当用相同的 n 发 ^Qn 命令时,该位置标志会重新出现,并且无论光标处于何处,均立即返回该位置标志处。第四,使用 ^Qn 命令时,如果位置标志 <n> 不存在,可用 <Esc> 键返回编辑状态。

在弄清楚了 ^Kn 和 ^Qn 命令的功能后,让我们回到问题本身,看一看该怎样具体操作。在编辑文本过程中,对某段文本尚无法确定放在 A 处好还是放在 B 处好,我们可以先在位置 A 处用 ^K0 命令设置位置标志 <0>,再用 ^K1 命令在位置 B 设置位置标志 <1>。当我们已经确定要在某处,如位置 B 处插入该段文本时,发出 ^Q1 命令光标便会自动移往位置 B,等待用户输入。如果已经记不清具体的位置与哪一个位置标志对应,可以用 ^Q0~^Q9 命令查找具体的位置,即使如此,比起用光标移动命令在整个文本中查找某一特定位置还是要方便得多。

计算机病毒的检测、消除和预防

苏民生

三、文件型病毒及其传染方式

1. 什么是文件型病毒?

文件型病毒是指病毒程序附着于正常文件中的病毒,又简称为文件病毒。

通常微机上的文件病毒,其病毒程序往往附着在原来文件的前面或后面,这类文件型病毒又称为外壳型病毒。

文件型病毒最显著的特点就是增加文件的长度。有些病毒还改变原来文件的生成日期。这些特点可以使人们通过文件长度、日期或代码和有无变化来检测是否感染病毒。

当执行已感染病毒的程序文件(COM 文件、EXE 文件,有时还包括覆盖文件 OVR 等)时,最先执行的是附着在该文件的病毒程序,然后才执行该文件原来的程序。因此,从运行过程上看,执行带病毒程序文件,相当于在执行正常操作之前插入了一段病毒程序的执行过程。

文件型病毒按照病毒程序是否驻留内存,可分为瞬时作用和驻留内存两种。瞬时作用的病毒程序不驻留内存,它仅仅在执行带病毒程序时执行一次,病毒的传染和造成的破坏都仅仅在此期间发生,如果在此期间系统和文件未被破坏,就可以继续工作,执行原来程序。而对于后一种病毒,一旦执行病毒程序,它就首先将自身存放于内存驻留区,同时修改系统参数和设置,包括减少用户可用内存大小,修改中断服务程序入口地址等。文件型病毒经常修改的有 INT21H 的入口地址,此外还常常有 INT8 等的入口地址。这就使得这些中断程序之前都插入了一段病毒程序,当调用这些中断程序时,要在执行中断程序之前先执行病毒的插入部分,从而实现病毒对系统的监控和感染、破坏。这就是说,系统从此处于病毒环境,这种环境下执行的程序文件都可能被传染上病毒,成为带病毒的程序文件,而在这种环境下往往检查不到病毒的存在,也无法实现对病毒文件的复原(即消除病毒)。

文件型病毒对文件的感染可分为主动攻击和执行时感染两种方式。主动攻击方式是当执行病毒程序时,它检查磁盘上其它文件,如发现所检查的文件没有被感染,就立即将它感染上该病毒。瞬时作用的维也纳病毒(648病毒)和驻留内存的1575/1591病毒都属于这一方式。执行时感染方式,是指在病毒环境中每当执行一个程序文件,如果所在磁盘没有写保护,这个程序文件就会感染病毒;反之,在病毒环境下并不去执行的干净的程序文件不会受到感染。无论哪一种方式,首先受到攻击的通常是文件COMMAND.COM,因此,经常检查该文件的长度和日期有无变化,是检测病毒的一种方法。而有些病毒程序为了隐蔽自身,又常常规定不传染COMMAND.COM病毒。

受攻击的程序文件类型,对于不同的病毒是不相同的。有的病毒只感染COM文件,有的病毒对COM文件和EXE文件都感染,还有一些文件型病毒感染其它文件。

和引导区病毒对于磁盘的交叉感染一样,多种文件型病毒对于同一文件的交叉感染也是很复杂的问题,使得检测病毒更不容易,消除病毒更加费力气,甚至有的文件难以复原。

2. 以色列病毒

这是国内出现较早的文件外壳型病毒,又称为耶路撒冷(Jerusalem)病毒、黑色星期五病毒、疯狂拷贝病毒、长方块病毒、PLO病毒等,1987年秋首次出现于耶路撒冷的希伯来大学。该病毒流传很广,变体甚多,危害严重。

病毒程序常驻内存,感染COM文件和EXE文件。对于COM文件,病毒部分1808字节放在原来文件之前,而在原来文件之后附加5字节标志“MsDos”,只感染一次。对于EXE文件,除修改文件标头之外,将1808字节的病毒部分续接在原来文件之后,并且可以多次感染,以致被感染文件在很短时间内快速增大,使所在磁盘无法继续使用。病毒部分第一行(16字节)内容的十六进制代码为:

```
E9 92 00 73 55 4D 73 44-6F 73 00 01 F5 09 00 00
```

相应ASCII字符为

```
i...sUMsDos...u...
```

用PC Tools或DEBUG查看被感染的EXE文件,每看到这样一行,就是一个病毒块的开头。对于长度不是16的整数倍的EXE文件,第一次感染病毒对长度增加比1808字节稍多,以后每次感染都恰好增长1808字节。“疯狂拷贝”即由此得名。

感染以色列病毒的系统,执行用户程序的速度减慢至原来的1/10。病毒进入系统半小时之后,屏幕左下方出现小方块,干扰屏幕显示。最为严重的是在恰好为13日星期五的日子,病毒环境下执行的每个COM文件和EXE文件都将被病毒删除,曾经造成不小的损失。“黑色星期五”即由此得名。

3. 1575/1591病毒

这是文件外壳型病毒,传染COM文件和EXE文

件。对每个文件只传染一次。将原来文件长度稍许增大成为16整数倍(字节数),然后将病毒部分1575字节接在后面,使原来文件增大1575至1591字节。感染病毒的COM文件最后三个字节为43 0C 0A,EXE文件时为45 0C 0A。

执行带病毒程序时,首先执行的是病毒程序;它对于硬盘COMMAND.COM进行感染,并在以后的运行过程中将病毒程序驻留内存,修改INT 21H等中断服务程序入口地址。

在病毒环境下,每当执行DIR、COPY等读写文件的DOS命令时,病毒程序逐一传染给磁盘上的COM文件和EXE文件。它不传染只读文件、隐含文件和系统文件。被感染文件的形成日期被改为当前日期。

基于以上特点,有的文献中称它是一个“通过文件进行传染并传染文件的操作系统型病毒”。

1575/1591病毒的破坏部分在一定条件下激活,此时从屏幕左上角开始向右爬出一条“虫子”,逐行由左向右移动,移动时“吃掉”虫子右边的字符,然后再显示出来,扰乱整个屏幕。因此,又有人称该病毒为“虫子”病毒,“毛毛虫”病毒,“蜈蚣”病毒。

4. Vienna病毒

这是一种不驻留内存的文件外壳型病毒,又称为648病毒,DOS 62病毒,UNESCO(联合国教科文组织)病毒。最初出现于1988年。

Vienna病毒只感染COM文件。病毒程序长648字节,续接在原来文件之后,并将原来文件前三个字节修改为跳转至病毒开头的跳转指令。执行带病毒COM文件时,它首先跳转到病毒部分开头,然后恢复原来程序前三个字节内容。病毒程序随后查找磁盘上未被感染的COM文件,将病毒程序传染过去。最后才转去执行原来的程序。传染病毒方式为主动攻击型。

病毒的表现部分在一定条件下会使系统重新引导,B型变体会偶然删除某些文件。该病毒有一定的破坏性。

5. 1701/1704病毒

1710病毒的破坏部分激活时,屏幕显示内容纷纷降落到屏幕底部。因此,又称为雨点病毒,瀑布病毒(Cascade),落叶病毒(Autumn Leaves),落泪病毒(Falling Tears),感冒病毒(FLU.)。首次出现于1987年,是感染COM文件的文件外壳型病毒,病毒程序驻留内存。

1704病毒与1701病毒只差一条指令,因此视为相同病毒,检测/消除软件Scan/Clean对它们以[17XX]表示。它们都有若干变体。

执行带病毒的COM文件时,首先跳转到续接在原来文件之后的病毒部分,然后才执行原来的程序。病毒部分的主体是加密的,只有其前面34字节的解密程序未加密。执行病毒程序时,先由前34字节指令将其主体解密,然后才转到病毒主体部分,将病毒程序驻留内存并修改有关参数和某些中断服务程序的入口地址。该病毒与众不同之处是对自身加密。

病毒程序驻留内存后,系统成为病毒环境下。此时执行的每个 COM 文件都会感染病毒;对每个 COM 文件只感染一次。感染病毒的 COM 文件,前 3 字节被替换,原来的前 3 字节内容被移至病毒部分并且被加密。

6. 1741 病毒

这是近期出现的文件外壳型病毒,感染 COM 文件和 EXE 文件。病毒程序做了加密处理,采用了对于 DEBUG 的反跟踪措施。在病毒环境下查看被感染文件长度时,用不同方法会看到不同的长度。

病毒程序长 1741 字节,将原来文件适当增大调整为 16 的倍数后,将病毒程序续接在其后。因此,感染病毒后文件增大 1742 至 1757 字节,视原来文件长度而定。

病毒程序驻留内存,通过修改中断服务程序入口

地址,将自身“插入”到 INT 21H 之前。在病毒环境下执行 DOS 命令 DIR 时,插入在 INT 21H 中的病毒程序将病毒传染给尚未感染的 COM 文件或 EXE 文件。对于具有只读、隐含或系统等特殊属性的文件也一样进行传染,但传染后又恢复其原来属性,如不用有效方法查看就不易察觉。文件的形成时间也不产生变化。在病毒环境下用 DIR 命令查看,被感染文件的长度未发生变化,这是一种假象,使人们难以觉察已处于病毒环境。但是,如果已经有文件感染病毒,长度变得比原来大了,而在某个环境中用 DIR 命令查看时,文件仍保持原来大小,就可以断定这是一个 1741 病毒环境。这也可成为检测该病毒的一种办法。

(未完待续)

怎样对付计算机病毒 ——介绍 WIPE 程序和硬盘初始化加密法

北京航空航天大学机械厂(100083) 张 亭

计算机病毒是程序,而程序只有被执行才能起作用,病毒程序也只有当其被执行才能传染,这是计算机病毒传染的一个原理。病毒可以变化类型,变化特征,变化传染机制,唯独不能变化原理,从原理出发去对付病毒是以不变应万变的办法。

基于原理,如果一台健康硬盘的微机不用软盘开机启动,也不执行来自软盘的程序,永远不可能沾染病毒。但一个“世外桃源”式的硬盘不能充分利用已有的软件资源,也难以充分发挥机器的功能。况且,硬盘上的程序基本上都是来自软盘,谁也不可能从系统到应用,什么软件都自己去编。因此主要问题在于把关。

为了把关的需要,笔者设计了一种可称“硬盘初始化加密法”的手段,并用驻留在软盘首扇区的 WIPE 程序方便地实现。加密后的硬盘外在表现上与以往的各种硬盘加密法一样,即:硬盘不能自举,系统软盘启动后找不到硬盘。不同之处:(一)加密时用无写保护的 WIPE 软盘开机启动,在实现硬盘引导的同时硬盘被加密,软盘成为钥匙盘,方便而又简捷;(二)需要使用硬盘时用钥匙盘开机启动,就能实现硬盘引导,比硬盘自举还要快,因没有解密过程,也无须再次加密;(三)无钥匙盘的人即使用 DEBUG 手工操作,获取硬盘信息亦无异大海捞针,因为分区表被转移到钥匙盘上,硬盘主引导扇区已被清零,形同刚刚做完低级格式化。这就是“硬盘初始化加密法”名称的由来。此法同时提供了一种“快速低级格式化”的技术,必要时,这样的硬盘能够有 FDISK 重新分区。

可以想象,一个如同刚出厂尚未分区的硬盘,任何病毒在其面前也是无可奈何的。然而一旦被钥匙盘打开,就完全解除了武装,文件型病毒可长趋直入。于是

责任就落到持钥匙盘者的身上,别人已没有传入病毒的可能。钥匙盘持有者得到了外来软件,可用 DOS 系统软盘启动后,先在一种“无硬盘”状态下,试运行于软盘和虚拟盘,并用程序监视其是否有非授命的写操作,确认无毒方可拷入硬盘。这种把关条件同时也是运行带毒软件而保证硬盘安全的条件,外人用机也只能以这种方式在软盘和虚拟盘运行程序。这就大大限制了文件型病毒的传染。

引导型病毒仍有能力进入加密后的硬盘,虽然进入后因其不被再次执行而没有什么意义,但进入的过程中可能同时发病。笔者注意到,硬盘时代已很少有意去用软盘引导,往往是由于冷启动时忘记打开软驱的门,被引导型病毒乘虚而入。为此,WIPE 程序具备了一种“先下手为强”的功能,就是引导硬盘建立系统后,把使用中的无写保护软盘统统变成 WIPE 软盘,覆盖可能存在的引导型病毒。这一过程很快,对系统速度的影响微不足道,而且不管是哪一个软驱,哪一种格式。区别于病毒传染,这一过程:①完整保留 BPB 表;②不备份首扇区。既可使 WIPE 软盘作为数据盘照常使用,又能对原有用户文件、数据完全没有影响,且不与引导型病毒交叉共存。

上述第②个区别恰恰是各种引导型病毒的一个致命弱点,它们备份首扇区引导记录的目的在于隐蔽自己,也因此造成对软盘文件、数据的威胁。WIPE 则没有任何隐蔽的必要,当它改写硬盘时都会伴有显示和响铃,而且抓住了病毒的这一弱点:若 WIPE 软盘被引导型病毒感染,病毒将在其执行作为备份保留的 WIPE 程序时,被 WIPE 重新覆盖并再次启动,决无进入系统、再次扩散传染的可能。因此,每一张 WIPE 软

盘不仅都能用来加密其他硬盘,也都能成为引导型病毒传染的一个句号——到此为止。众所周知,引导型病毒因其短小而易读易改,变种极多,而 WIPE 程序有能力将其斩尽杀绝。

虽然 WIPE 钥匙盘也象普通 WIPE 软盘一样,可作为数据盘照常使用,不怕被引导型病毒感染,但普通 WIPE 软盘却不能打开加密后的硬盘,钥匙盘只能用 DISKCOPY 等全盘复制方式备份。钥匙盘以双字标记,可变化近四十三亿种不同形式。

第一张 WIPE 软盘的制作可在 DEBUG 中用 A 命令键入程序一,核对无误后执行 G=100。任何赖在硬盘上请不出去的引导型病毒都将在加密时被毫不客气的刷掉,但加密前应保证硬盘上没有任何文件型病毒。程序二是解密程序,万一有所疏漏或把关失误,可在钥匙盘打开硬盘后用来解密,以便干净的 DOS 系统软盘启动后能够找到和进入硬盘,进行杀灭文件型病毒的操作。解密后的硬盘为 WIPE 硬盘,可在病毒清除后用开口 WIPE 软盘再次加密。程序三是 A 盘写保护程序,用以检查外来软件。应存于 DOS 系统软盘,在外来软件试用前执行。执行后系统对 A 盘拒写,并给出响铃警告。

WIPE 程序适用于 DOS 3.3 操作系统,其编制中运用了病毒的编程技巧,运行中采用了病毒的驻留内存方式,“以其毒之道还治其毒之身”。

最后要强调一下,对付计算机病毒应从原理入手,加强责任心,提高系统维护水平。WIPE 程序只是一个抗病毒工具,任何工具都是有局限性的,“一卡在手,高枕无忧”之类的说法不过是商业性的宣传而已。

程序一:

```

100  MOV    AX,201
103  MOV    BX,400
106  MOV    CX,1
109  MOV    DX,0
10C  INT    13
10E  JB     100
110  MOV    SI,403
113  MOV    DI,203
116  MOV    CX,1B
119  REPZ
11A  MOVSB
11B  INC    CL
11D  MOV    BH,2
11F  MOV    AX,301
122  INT    13
124  JB     11F
126  INT    20
200  JMP    2BA
21E  CMP    DL,81
221  JZ     26D
223  PUSH   DS
224  PUSH   SI
225  PUSH   DI
226  PUSH   CX

```

```

227  CLD
228  PUSH   CS
229  POP    DS
22A  MOV    SI,200
22D  CMP    CX,1
230  JNZ    258
232  AND    DH,DH
234  JNZ    258
236  CMP    DL,80
239  JZ     245
23B  CMP    AH,3
23E  JNZ    277
240  ES:
241  CMP    [BX+B],SI
244  JZ     28C
246  MOV    CL,42
248  SUB    SI,CX
24A  MOV    DI,BX
24C  ADD    DI,SI
24E  REPZ
24F  MOVSB
250  CLC
251  POP    CX
252  POP    DI
253  POP    SI
254  POP    DS
255  RETF   2
258  MOV    DI,[SI-4D]
25B  CMP    [DI+2],CX
25E  JNZ    272
260  CMP    [DI],DX
262  JNZ    272
264  CMP    AH,3
267  JZ     250
269  POP    CX
26A  POP    DI
26B  POP    SI
26C  POP    DS
26D  JMP    C800:1C8
272  CMP    DL,80
275  JZ     269
277  CMP    AH,16
27A  JZ     28C
27C  CMP    AH,2
27F  JNZ    269
281  XOR    DI,DI
283  MOV    DS,DI
285  TEST   BY PT[43F],1
28A  JNZ    269
28C  CALL   3A6
28F  PUSHF
290  PUSH   AX
291  PUSH   ES
292  PUSH   BX
293  PUSH   DX
294  XOR    DH,DH

```

296	MOV	AX,201	2FA	MOV	[SI-2],AX
299	MOV	BX,SI	2FD	MOV	ES,AX
29B	CALL	399	2FF	POP	SI
29E	JB	2B3	300	POP	DI
2A0	PUSH	CS	301	PUSH	AX
2A1	POP	DS	302	MOV	CX,109
2A2	CMP	[SI+2B],SI	305	PUSH	CX
2A5	JZ	2B3	306	REPZ	
2A7	MOV	DI,3	307	MOVSW	
2AA	ADD	SI,DI	308	RETF	
2AC	MOV	CL,1B	309	MOV	AX,201
2AE	REPZ		30C	PUSH	AX
2AF	MOVSB		30D	CALL	39C
2B0	CALL	391	310	POP	AX
2B3	POP	DX	311	MOV	SI,BX
2B4	POP	BX	313	MOV	BX,1BE
2B5	POP	ES	316	MOV	DI,BX
2B6	POP	AX	318	MOV	CL,4
2B7	POPF		31A	MOV	BP,CX
2B8	JMP	251	31C	CMP	[BX+SI],DL
2BA	CLD		31E	JZ	347
2BB	XOR	AX,AX	320	ADD	BL,10
2BD	MOV	SS,AX	323	LOOP	31C
2BF	MOV	DX,AX	325	CALL	3AD
2C1	MOV	BX,7C00	328	CMP	WO PT [SI-8],100
2C4	MOV	SP,BX	32D	JNZ	336
2C6	PUSH	CS	32F	CMP	WO PT [SI-6],200
2C7	PUSH	BX	334	JZ	308
2C8	PUSH	AX	336	INT	16
2C9	PUSH	BX	338	JMP	342
2CA	PUSH	DX	33A	MOV	CX,1
2CB	POP	DS	33D	MOV	AX,301
2CC	MOV	DI,413	340	INT	40
2CF	DS:		342	JMP	FFFF:0
2D0	PUSH	DI	347	CS:	
2D1	DEC	CX	348	MOV	[DI-B],BX
2D2	PUSH	AX	34B	ADD	SI,DI
2D3	INC	BP	34D	MOV	CL,20
2D4	CMP	AL,A	34F	PUSH	CS
2D6	OR	AX,3931	350	POP	ES
2D9	CMP	[BP+SI],SI	351	REPZ	
2DB	POP	ES	352	MOVSW	
2DC	OR	AX,A	353	CALL	3AD
2DF	TEST	BY PT[DI],7F	356	XOR	DX,DX
2E2	JNZ	33A	358	DEC	BP
2E4	MOV	SI,4C	359	JZ	377
2E7	MOV	DI,7C6E	35B	CALL	381
2EA	MOVSW		35E	MOV	AX,301
2EB	MOVSW		361	JB	358
2EC	POP	DI	363	PUSH	AX
2ED	MOV	AX,[DI]	364	CALL	3AD
2EF	DEC	AX	367	POP	AX
2F0	STOSW		368	SUB	BX,AX
2F1	MOV	CL,6	36A	CALL	39E
2F3	SHL	AX,CL	36D	PUSH	CS
2F5	MOV	WO PT[SI-4],1E	36E	POP	DS

36F	MOV	SI,CF
372	JMP	0:7D32
377	CMP	[7F95],AX
37B	JZ	308
37D	PUSH	CS
37E	POP	DS
37F	JMP	36A
381	MOV	SI,46C
384	LODSW	
385	MOV	[7DF8],AX
388	CS:	
389	MOV	[12B],AX
38C	LODSW	
38D	MOV	[7DFA],AX
390	CS:	
391	MOV	[132],AX
394	MOV	AX,301
397	XOR	BX,BX
399	PUSH	CS
39A	JMP	3A2
39C	MOV	BH,7E
39E	MOV	DX,80
3A1	PUSH	DS
3A2	POP	ES
3A3	MOV	CX,1
3A6	PUSHF	
3A7	CS:	
3A8	CALL	FAR [6E]
3AC	RET	
3AD	PUSH	DS
3AE	POP	ES
3AF	MOV	BX,7C00
3B2	MOV	DI,D2
3B5	CS:	
3B6	MOV	CX,[DI+2]
3B9	CS:	
3BA	MOV	DX,[DI]
3BC	JMP	3A6
3FE	DB	55 AA

程序二:

100	XOR	BX,BX
102	MOV	DS,BX
104	MOV	AX,[413]
107	MOV	CX,6
10A	SHL	AX,CL
10C	MOV	ES,AX
10E	MOV	CL,1
110	MOV	DX,80
113	MOV	AX,301
116	PUSHF	
117	ES:	
118	CALL	FAR [6E]
11C	JB	113
11E	INT	20

程序三:

0100	JMP	0119
0102	OR	DH,DH
0104	JNZ	0114
0106	CMP	AH,3
0109	JNZ	0114
010B	MOV	AX,E07
010E	INT	10
0110	CLC	
0111	RETF	2
0114	JMP	F000:FFF0
0119	CLD	
011A	PUSH	CS
011B	PUSH	CS
011C	POP	CX
011D	POP	ES
011E	XOR	AX,AX
0120	MOV	DS,AX
0122	MOV	SI,100
0125	MOV	DI,115
0128	MOVSW	
0129	LODSW	
012A	CMP	AX,C000
012D	JB	013E
012F	STOSW	
0130	MOV	AX,102
0133	MOV	[SI-4],AX
0136	MOV	[SI-2],CX
0139	MOV	DX,119
013C	INT	27
013E	MOV	DI,102
0141	MOV	SI,DI
0143	MOV	DS,AX
0145	MOV	CX,9
0148	CMPSW	
0149	LOOPZ	0148
014B	JNZ	014F
014D	INT	20
014F	MOV	SI,16B
0152	PUSH	CS
0153	POP	DS
0154	LODSB	
0155	OR	AL,AL
0157	JZ	0162
0159	MOV	BX,7
015C	MOV	AH,E
015E	INT	10
0160	JMP	0154
0162	XOR	AH,AH
0164	INT	16
0166	JMP	F000:FFF0
016B	DB	'Virus in'
0173	DB	'memory'
017B	DB	'!!!'0A 0D
017F	DB	0

ProDOS 系统内部结构剖析(续)

北京铁路局卫生防疫站(100038) 廖 凯

4. 转换系统程序

每个系统程序必须包含一个转换到另一个系统程序的指令,由于新的系统程序被装到起始地址 \$ 2000 处,转换到新程序内的代码应该被放到低于 \$ 2000 处以免被重写。转换系统程序的方法是:

- ①关闭所有打开的文件。
- ②提示用户给出所需的系统程序的名字。
- ③打开包含指定的系统程序的文件,使用 GET—EOF 调用确定文件的长度。
- ④重新分配由你的系统程序所使用的 RAM。
- ⑤读取整个文件到内存起始地址 \$ 2000 处。
- ⑥关闭文件。
- ⑦放置所需的路径名或部分路径名到 \$ 280 单元。
- ⑧执行一个 JMP 指令到 \$ 2000。

如果在新系统程序读入期间发生了一个 MLI 错误,并且不能恢复你的程序,那么屏幕会被清除并通知用户重新启动系统。

第七步是很重要的。一个使用多个文件的系统程序必须知道它在哪个目录内可存取文件。在 \$ 280 单元使用路径和当前部首(如果路径不是用一个斜线开始的话),一个系统程序可以确定存有其它文件的目录。这样,任何系统程序可以被放在磁盘的任何目录内。

二、管理系统内存

1. 使用堆栈

在 Apple 机内,堆栈占用内存 \$ 100—\$ 1FF,从页的高字节开始向下使用。在一个中断发生时,中断处理程序保存栈的低 16 字节,但只有在栈大于 3/4 满时才进行。

系统程序应该在冷启动进入点设置栈指针为 \$ 1FF。

2. 使用扩展 64K RAM 存储器

Apple II e 的扩展 64K RAM 存储器不能被 MLI 调用存取。对于这扩展存储器,系统在启动时将它设定为 /RAM 虚拟盘,设为 S0,D2。

若想将这区域作为正规数据空间给你的程序用,就要从系统中将这卷(/RAM)删除,按下面的方法进行:

- ①使用现有的磁盘设备数(DEV CNT, \$ BF31),搜寻整个设备表(DEV LST, \$ BF32 到 \$ BF32+DEV CNT),寻找字节的高半部等于 8(表示 S0,D2)的字节。
- ②将表的右半部向左移动一字节来删除这字节。
- ③设备数(DEV CNT)减一。

④如果最后设备存取的 DEV NUM(\$ BF30)等于零,则指定 DEV NUM 为最后设备在 DEV LST 的值。

只有在这表中的设备才被系统使用。

3. 系统整体页面

内存 \$ BF00—\$ BFFF 包含系统的全局(整体)变量,有些变量可以被系统程序设置和使用,例如系统位图和日期及时间单元。有些变量,例如机器识别字节是提供信息的,不能被改变。有些变量只被系统内部使用。

下面的汇编程序的伪指令 DFB 指定一个值到当前的存储单元。DW 伪指令设置一个二字节地址(低字节在前)到当前单元。

系统整体页面

MLI 进入点是你经常调用的整体页面内的唯一地址。

BF00:ENTRY JMP MLIENT1;MLI 调用进入点其它进入点,不要使用这些:

BF03:JSPARE JMP GL.RTS;备用向量
BF06:DATETIME JMP GL.RTS;时钟计时子程序
BF09:SYSERR JMP SYSERR1;错误报告向量
BF0C:SYSDEATH JMP SYSDEATH;系统故障向量
BF0F:SERR DFB \$00;错误代码 0=无错

磁盘设备驱动程序向量

BF10:DEVADR01 DW \$0000 ;如果 64KRAM 存在
BF12:DEVADR11 DW \$0000 ;S1,D1
BF14:DEVADR21 DW \$0000 ;S2,D1
BF16:DEVADR31 DW \$0000 ;S3,D1
BF18:DEVADR41 DW \$0000 ;S4,D1
BF1A:DEVADR51 DW \$0000 ;S5,D1
BF1C:DEVADR61 DW \$0000 ;S6,D1
BF1E:DEVADR71 DW \$0000 ;S7,D1
BF20:DEVADR02 DW \$0000 ;S0 专用
BF22:DEVADR12 DW \$0000 ;S1,D2
BF24:DEVADR22 DW \$0000 ;S2,D2
BF26:DEVADR32 DW \$0000 ;S3,D2
BF28:DEVADR42 DW \$0000 ;S4,D2
BF2A:DEVADR52 DW \$0000 ;S5,D2
BF2C:DEVADR62 DW \$0000 ;S6,D2
BF2E:DEVADR72 DW \$0000 ;S7,D2

通过设备号列印所有现有的磁盘设备,当请求存取一个未识别的卷时,从列表的末端到开端搜寻设备。在 DEV LST 内每个字节的低半部是一个设备识别码:0=Disk II,4=ProFile。

BF30:DEVNUM DFB \$60;使用的最后设备 Slot 和 Drive
BF31:DEV CNT DFB \$01;现有磁盘设备数=1
BF32:DEV LST DFB \$E0\$60;现有磁盘设备的检索

表;在表内的设备可能未连接

DFB 0,0,0,0;这些是用于对
DFB 0,0,0,0;未识别的卷名
DFB 0,0,0,0;搜寻整个设备

BF40: ASC"(C)1983 APPLE COMPUTER"

接下来是低 48K 内存的位图。每一位对应存储区的一页(256 字节)。保护区用一个 1 标记,未保护区用一个 0 标记。ProDOS 不允许向保护区内读取或在保护区内分配输入输出缓冲区。

BF58:BITMAP DFB \$ CF, \$ 00, \$ 00, \$ 00, \$ 00, \$ 00,
DFB \$ 00, \$ 00, \$ 00, \$ 00, \$ 00, \$ 00,
DFB \$ 00, \$ 00, \$ 00, \$ 00, \$ 00, \$ 00,
DFB \$ 00, \$ 00, \$ 00, \$ 00, \$ 00, \$ 01,

BF58 单元的内容 CF 表示第 0、1、4、7 页被保护。

BF6F 单元的内容 \$ 01 表示 \$ BF 页(\$ BF00 ~ \$ BFFF)被保护。

下面表中的地址是用于打开的文件的缓冲区地址。这些只是提供信息,除用 MLI 调用 SEF-BUF 之外,不要改变它们。

BF70:BUFFER1 DW \$ 0000;文件号 1
BF72:BUFFER2 DW \$ 0000;文件号 2
BF74:BUFFER3 DW \$ 0000;文件号 3
BF76:BUFFER4 DW \$ 0000;文件号 4
BF78:BUFFER5 DW \$ 0000;文件号 5
BF7A:BUFFER6 DW \$ 0000;文件号 6
BF7C:BUFFER7 DW \$ 0000;文件号 7
BF7E:BUFFER8 DW \$ 0000;文件号 8

中断向量被存放在下表。此外,这些信息只能被 MLI 调用 ALLOC-INTERRUPT 和 DEALLOC-INTERRUPT 所改变,A,X,Y,栈和状态寄存器的值在最近中断的时候也存在这里。另外,中断的地址也被保护起来。这些可以用于研究和除错,但不能被用户改变。

BF80: INTRUPT1 DW \$ 0000 ;中断子程序 1
BF82: INTRUPT2 DW \$ 0000 ;中断子程序 2
BF84: INTRUPT3 DW \$ 0000 ;中断子程序 3
BF86: INTRUPT4 DW \$ 0000 ;中断子程序 4
BF88: INTAREG DFB \$ 00 ;A 寄存器
BF89: INTXREG DFB \$ 00 ;X 寄存器
BF8A: INTYREG DFB \$ 00 ;Y 寄存器
BF8B: INTSREG DFB \$ 00 ;堆栈寄存器
BF8C: INTREG DFB \$ 00 ;状态寄存器
BF8D: INTBANKID DFB \$ 00 ;ROM,RAM1 或

RAM2, 语言卡的

\$ D000

BF8E: INTADDR DFB \$ 00 ;中断的地址返回地址

下面的任选项可以在 MLI 调用之前被改变。

BF90: DATE DW \$ 0000 ;存放年,月,日
BF92: TIME DW \$ 0000 ;存放小时.分钟
BF94: LEVEL DFB \$ 00 ;文件级
BF95: BUBIT DFB \$ 00 ;控制备用位的使用
BF96: SPARE1 DW \$ 0000 ;备用

下面的只是提供信息。

MACHID 标识系统属性:

Bit 7-6: 00= I ;01= I + ;10= I e ;11= I 仿真

Bit 5-4: 00=未用;01=48K;10=64K;11=128K

Bit 3-2;保留将来定义

Bit 1: 1=80 列卡;0=无

Bit 0: 1=Thunder clock 或兼容的;0=无

SLTBYT 指明包含 ROM 的槽口。

PFIXPTR 指明一个现有的部首,如果非零。

CMDADR 上一次 MLI 调用的参数表的地址。

SAVEX 和 SAVEY 上一次 MLI 被调用时的 X 和 Y 的值。

BF98:MACHID DFB \$ 00 ;机器识别码

BF99:SLTBYT DFB \$ 00 ;A“1”bit 指明 ROM 在槽口“bit#”

BF9A:PFIXPTR DBF \$ 00 ;若=0,无部首存在

BF9B:MLIACTV DFB \$ 00 ;bit 7 为“1”=MLI 有效

BF9C:CMDADR DW \$ 0000 ;返回 MLI 最后调用的地址

BF9E:SAVEX DFB \$ 00 ;X 寄存器

BF9F:SAVEY DFB \$ 00 ;Y 寄存器

下面的空间是保留给语言卡存储区切换子程序的。所有子程序和地址将随着系统的配置而改变。下面只适用于 64K 系统。

BFA0:4D 00 E0 EXIT EOR \$ E000 ;测试 ROM 启用
BFA3:F0 05 BEQ EXIT1 ;若 RAM 启用则转移
BFA5:8D 82 C0 STA ROMIN ;否则启用 ROM 并返回
BFA8:D0 0B BNE EXIT2
BFAA:AD F5 BF EXIT1 LDA BNKBYT2 ;
BFAD:4D 00 00 EOR \$ D000 ;启用
BFB0:F0 03 BEQ EXIT2 ;若无扩展 RAM 则转移
BFB2:AD 83 C0 LDA ALTRAM ;否则启用扩展 \$ D000
BFB5:68 EXIT2 PLA ;恢复返回码
BFB6:40 RTI ;再允许中断并返回
BFB7:38 MLIENT1 SEC
BFB8:6E 9B BF ROR MLIACTV ;指向 MLI 现有的中断子程序
BFB8:AD 00 E0 LDA \$ E000 ;保存语言卡/RAM
BFB8:8D F4 BF STA BNKBYT1 ;
BFC1:AD 00 D0 LDA \$ D000 ;在 MLI 退出时恢复
BFC4:8D F5 BF STA BNKBYT2
BFC7:AD 8B C0 LDA RAMIN ;强使 RAM 卡接通
BFCA:AD 8B C0 LDA RAMIN ;允许写 RAM
BFCD:4C 00 D0 JMP ENTRYMLI

下面是中断退出和进入子程序

BFD0:AD 8D BF IRQXIT LDA INTBANKID ;确定 RAM 卡的状态
BFD3:F0 0D BEQ IRQXIT2 ;如果启动则转移
BFD5:30 08 BEQ IRQXIT1 ;若辅助 \$ D000 启动则转移
BFD7:4A LSR A ;确定有无 RAM 卡
BFD8:90 0D BCC ROMXIT ;若 ROM 仅有系统则转移
BFDA:AD 82 C0 LDA ROMIN ;否则启动第一个 ROM
BFDD:B0 08 BCS ROMXIT ;转移总是进行...

BFDf,AD 83 C0 IRQXIT1 LDA ALTRAM ;启动辅助 \$D000
 BFE2,A9 01 IRQXIT2 LDA #1 ;预置 BANKID 给 ROM
 BFE4,8D 8D BF STA INTBANKID ;(若 RAM 卡中断,则复位)
 BFE7,AD 88 BF ROMXIT LDA INTAREG ;恢复累加器...
 BFEA,40 RTI ;并退出
 BFEB,AD 8B C0 IRQENT LDA RAMIN ;这进入点用在
 BDEE,AD 8B C0 LDA RAMIN ;在中断时执行 ROM
 BFF1,4C 31 D1 JMP IRQRECEV ;A 寄存器被存在 \$ 45
 BFF4:00 BNKBYT1 DFB \$ 00
 BFF5:00 BNKBYT2 DFB \$ 00

BFF6:48 55 53 54 4F 4E DFB \$ 48, \$ 55, \$ 53,
 \$ 54, \$ 4F, \$ 4E

下面 2 字节必须被每个系统程序设置。第 1 字节是可以被系统程序使用的 MLI 最早的版本号。第 2 字节是系统程序版本号。

BFFC:00 IBAKVER DFB \$ 00 ;最早的 MLI 版本
 BFFD:00 IVERSION DFB \$ 00 ;系统程序版本号
 BFFE:00 KBAKVER DFB \$ 00 ;兼容的最小版本
 BFFF:00 KVERSION DFB \$ 00 ;版本号(释放号)

(待续)

(上接第 27 页)

:去污 加洗衣粉 漂洗 ;
 :启动甩干电机 甩干电机 接通 ;
 :停止甩干电机 甩干电机 断开 ;
 :甩干 启动甩干电机 等待 3 分钟
 停止甩干电机 ;
 :响铃 蜂鸣器 接通 ;
 :停止响铃 蜂鸣器 断开 ;
 :洗毕响铃
 5 0
 D0 响铃 等待 5 分钟
 停止响铃 等待 5 秒钟

LOOP ;

: 洗衣 漂洗 去污 漂洗 漂洗 漂洗
 甩干 洗毕响铃 ;

这个例子,程序的算法比较单纯,很多深入问题也还没有涉及到。实际当中各种程序往往比它复杂得多。但做为一个完整的示例,对于这个讲座来说,对于初学者更全面地了解 FORTH 语言来说,还是很必要的。

另外,再强调一下,当硬件条件具备(指洗衣机和接口卡)时,又使用汉化 FORTH 语言版本的情况下,以上的设计是可以实际运行的,而不是汉语写的“伪码”或注释。计算机程序只能用“英文”编写的看法,应该改一改了。

《录像机维修》出版

由电子工业出版社主办的专业性普及技术读物《录像机维修》,94 年 1 月出版。宗旨是:传播录像机、摄像机、摄录机、电视机、监视器、激光视盘等视频技术。推广最新产品,为用户服务,重点为维修人员提供维修方法、经验和实用资料。以专家专稿为主,选登国内外几十种电子刊物的相关文章。设摄录像机、选购常识、使用指导、维修技术、拾零园地、资料图表

六个固定栏目和几个机动栏目,奉献读者所需,揭示维修诀窍,兼顾消费指南,博采众家之长。以期刊形式编写,书的形式自办发行。每月一期,16 开,32 页,定价:2.00 元,全年 24.00 元。凡订不足一年的每期另加 0.50 元邮资。汇款至北京东燕郊交通部电视中专电教室《录像机维修》编辑部,邮码:101601。

第十讲 输入与输出(下)

北京西四北三条 10 号 FORTH 应用研究会(100034) 丁志伟

上一讲介绍了输出功能,这一讲介绍输入功能。

输入的内容也有数字和字符(或字符串)两种情况。这回先来看看单个字符的输入。

D1.0 版本中,有一组用于单个字符输入的词,它们是:

INKEY \$ (--0)或(--C1)检查是否有按键,如果有,则在栈中留下键码及 1;如果没有,则在栈中留下 0。

KEY@ (--C)等待键盘输入一个字符,把该键的 ASCII 码留在栈中。

PAUSE (--)暂停程序运行,等待按键,按键之后,恢复程序运行。

ASC (--)等待键盘按一键,按键之后,把该键的 ASCII 码显示出来。

这几个词是有关联的,其中 INKEY \$ 最基本,其余几个都是在它的基础上建立起来的。请看其定义:

```
: INKEY $ C000 C@ 7F >
  IF C000 C@ 1 C010 OFF
  ELSE 0 THEN ;
```

熟悉苹果机的读者都知道,一旦有按键,就会把这个键的 ASCII 码(其最高位是 1)置入 C000 单元,这是由硬件决定的。检测到这个单元高位为 1,就说明有按键,读出键码之后,对 C010 进行读写,会将 C000 最高位置入 0,以便下一次读取,这也是硬件所完成的。

这里有一个词

OFF (a--)把 a 地址内容置 0

INKEY \$ 的执行过程是,先读取 C000 单元,如果有按键,这个单元的高位应该为 1,也就是大于 7FH,就把其内容(也就是键码)取出放到堆栈上,然后在栈顶放一个做为标志的 1,最高对 C010 进行读写,将 C000 高位置 0,以便下一次正确读入键码;如果没有按键,C000 单元高位为 0,不大于 7F,就在栈顶放进一个 0。

有些读者对于以上介绍的过程,如果一时难以理解,可以不去深究。

INKEY \$ 这个词名是从 PC BASIC 借用过来的,可能用 KEY \$ 更好些。在 PC/FORTH 中,同样功能的词名是 ? TERMINAL。

下面这个小程序,可以看出 INKEY \$ 是怎么使用的。

```
: TEST A0
  BEGIN DUP C. 1+ DUP FF >
  IF DROP A0 THEN
```

INKEY \$

END DROP ;

执行这个小程序,屏幕上会不停地显示各个字符,一旦有按键,就会停止运行。

KEY@这个词是利用 INKEY \$ 定义的。

```
: KEY@ BEGIN INKEY $ END ;
```

它是一个循环,等待按键。没有按键时,INKEY \$ 会在栈中留下一个 0,于是循环继续;一旦有按键,就会退出。END 会消耗掉 INKEY \$ 留下的标志,退出时只在栈中留下一个键码,请看下面“密码”小程序:

```
: PASS ." Please press password"
  BEGIN KEY@ D0 = DUP
  IF ." Yes,you are right"
  ELSE ." No,try again" THEN
  END ;
```

只有按 P 键(ASCII 字符为 D0)才能通过。

在编上边小程序时,需要知道“P”键的键码,为此系统词典提供了一个词 ASC,它的定义是这样的:

```
: ASC KEY@ . ;
ASC ✓
```

等待按一键。按 1,显示出它的键码 B1。

这个词虽然简单,但没有它,查键码有时却是比较麻烦的事情。

有些程序在运行时显示速度太快,或内容太多,希望能暂停一下,为此系统提供了一个词 PAUSE。请看其定义:

```
: PAUSE KEY@ DROP ;
```

执行 KEY@ 读取一键之后,会在栈中留下键码,此处并不需要这个码,因而用 DROP 将其丢掉。

PC/FORTH 中也有类似功能。与 KEY@ 对应的词名为 KEY。

从这一组词的定义方法中,可以看出系统设计的某些方面情况,FORTH 系统的构成之简单灵活,此处可见一斑。

实际使用中,还有大量的字符串输入。输入字符串的功能,可以借助单个字符输入的功能编出来,PC/FORTH 就是这样做的。D1.0 版本则不是这样,它用了监控中键盘输入子程序。

在 D1.0 版本中,有一个单词名为 LIN,它的功能是从键盘输入一行字符串,被输入的内容存放在 200H 起始的键盘缓冲区之中,解释状态下和编辑状态下的键盘输入最终都用到它。当然,输入之后还要有一个处理过程,解释状态下是将键入的内容放在键盘缓冲区,

然后解释执行；编辑状态下要将输入的内容搬到缓冲区，进行各种编辑处理。

在解释状态下输入的内容，解释器会把它们按一个词一个词处理，因而要能区分出一个词。处理时首先要知道被处理内容所在位置。词典中有两个变量，名为 INPUT 和 INTOP，其中存放着两个地址指针，分别指向将被处理内容的起址和终址。这部分内容又称为输入流。一般情况下，输入流位于 200H 起始的位置，也就是键盘输入缓冲区之中。每次键入的内容，就都放在这里。

系统中有一个名为 YICI 的单词，它的功能是从输入流中读取一个词，放到 DTOP 处，以便下一步处理，然后，移动 INPUT 指针，指向下一个词。怎么算做一个词呢？比如打入一行：

```
DEC 256 HEX .
```

查看 10 进制下的 256 在 16 进制下如何表示。

上边这行中，DEC、256、HEX 和点单词分别被 YICI 认做一个词。只要是空格、回车隔开的各部分，都算做词。词与词之间必须要有空格或回车，否则就会认为是一个词；一个词不可分成两部分，否则就会认为是两个词。不遵守这种要求，系统就不能正确地各个单词。对于习惯于苹果机 BASIC 的读者，对这一点请加注意。

被 YICI 读入 DTOP 的词，可以由 PNM 显示出来。比如：

```
: .YICI YICI PNM ;
```

定义之后输入

```
.YICI A-WORD
```

显示出 A-WORD。

下面研究一下 YICI 是如何定义的：

```
: YICI SKIP A0 WORD ;
```

其中有两个词：

SKIP (--) 当输入流起始指针指向的不是一个词起始而是空格或回车时，使这个指针跃过这些回车和空格，指向一个词的起始。

WORD (n--) 把 n 中两个字节做为一个词的终止标志，从输入流读入一个字符串，然后将其存入 DTOP，在 DTOP 处同时给出字符串长度。

对于各个 FORTH 系统，WORD 的功能大同小异。D1.0 可以用两个字节做定界符号，而多数系统只用一个字节做定界符。YICI 中的 A0 WORD，系统会认为回车和空格是定界符。用 A9 (在括号的 ASCII 码) 做定界符，就产生了 . (的定义：

```
: .( SKIP A9 WORD PNM ;
```

输入流中的内容将要进行各种处理。LOAD (编译) 就是其中之一，LOAD 是做编译，请看其定义：

```
: LOAD BTOP @ 1 - INTOP !  
    BUF @ INPUT ! ;
```

它只做一件事，把输入流指针指向编辑缓冲区，使

得编辑缓冲区的内容得以解释执行。此过程可能产生很复杂的处理过程，由于与 I/O 关系较少，不介绍了。严格地讲，LOAD 并不是编译，真正的编译是由冒号、VAR 等定义词完成的。这种设计思想，是很巧妙的。

下面介绍数字的输入。

从操作角度看，数字输入比字符串简单；但从系统处理角度看，数字输入却比较复杂。

数字输入的处理过程要分为两步。首先从键盘接收一个字符串，这个字符串中每个字符都应该是数字字符，如“1”、“3”、“8”等。接收字符串，刚才已经介绍过。然后还要将其转换成系统所能接受的数据形式，存入堆栈。

解释状态下，可以输入数字，前几讲已多次使用，直接打入一个数字即可。比如：

```
53
```

回车之后，53 已经入栈。

这种把数字存入堆栈的功能，是由解释器完成的。前几讲从不同角度介绍过解释器，这回再介绍一下它是如何处理数字输入的。先把它的定义列出来：

```
: MAIN2  
    BEGIN GETVERB  
        IF SWAP DROP CFA EXEC  
        ELSE NUMBER 0=  
        IF DEFERR THEN  
        THEN
```

```
    AGAIN ;
```

介绍其中与数字输入有关的两个词。

GETVERB (--a₁ a₂ 1) 或 (--0)

它调用 YICI，从输入流中读入一个字符串，然后检查是否词典中的词，如果是，则在栈中留下这个词的首址和终址，以及做为标志的 1；如果不是，则只在栈中留下一个 0 标志。不管这哪种情况，都将这个字符串放在栈顶。

NUMBER (--n 1) 或 (--0)

它与 WORD、YICI 或 GETVERB 配合使用。检查由 WORD 读入词典顶部的字符串是否全部由数字字符组成，如果是，则将其转换成双字的数据，存入堆栈，然后在栈中放一个 1 做为标志；如果转换不成功，则只在栈顶放一个 0 标志。

MAIN2 具有解释执行的功能，先不去讨论。这里只关心它的数字转换功能。它从输入流中读入一个字符串，如果不是词典中的词，就由 NUMBER 将其转换成数字。假如不成功，就执行错误处理词 DEFERR。前边的 53，是能正确转换的，于是执行后会把这个数存入堆栈。

用户需要在程序中编入输入数字的功能时，只要使用 YICI NUMBER 即可。请看例子。

```
: TEST ." Please input a number"
```

```
YICI NUMBER
```

```
IF ." Your input is" . CR
```

```
ELSE ." Noc a number" THEN ;
```

系统中有一个词名为 GETNUB,其功能是要输入一个数,如果不是数,就会响一声,显示冒号,直至输入一个数字。

```
: GETNUB 33 C@ BA 33 C!  
  BEGIN YICI NUMBER DUP 0=  
    IF 87 C. THEN  
  END SWAP 33 C! ;
```

这个词功能与 BASIC 中 INPUT 相仿,必须打回车,如果希望只输入一位数字,而不打回车,就要用 KEY@了。在 PC/FORTH 中,与 GETNUB 功能相同的词名为 IN#。

前边介绍了 NUMBER 的功能,与它相近的还有一个单词,名为 ASCB。它的名称含义是 ASC→B,ASCII 字符表示的字符串转换成二进制(BINARY)数码。堆栈中的数可以看做二进制数。这个词与 NUMBER 不同之处在于,被转换的字符串不一定放在词典顶部,而要事先给出它的地址,它的栈注释是:

```
( n1 a--n2 1)当转换成成功时。  
(n1 a--0)当转换不成功时  
n1 a 代表一个字符串,n2 是转换好的数。
```

与 NUMBER 和 ASCB 相反,堆栈中的数据也可以转换成字符串,供输出使用。

```
BASC ( n0 n1 a--n2 a)将有符号数转换成字符串。n1a 为将要存放输出字符串的地方。
```

```
UBASC ( u n1 a--n2 a)将无符号数转换成字符串。
```

这两个词的差别,是前者转换有符号数,后者转换无符号数。它们不会有转换不成功的情况,因此不给出正确与否标志。

举个例子,先建立一个字符串词,用于存放将转换好的字符串。

```
: $N1 " 123456789" ;
```

这个串中内容并不重要,但要有足够的位数。之后打入:

```
357 $N1✓
```

此时栈状态就是(n₀ n₁ a),然后执行

```
BASC✓
```

已将栈中数转换成字符串存放在 \$N1 中了。此时栈状态为(n₂ a)。继而执行

```
S.✓
```

显示出 357。

在 FORTH 中,被显示的内容都是字符和字符串。数字也要先转换为字符串再显示。点单词都经过了转换,请看定义:

```
: . ( n-- )  
  " 0123456789ABCDEF" BASC S. ;  
: U. ( U-- )  
  " 0123456789ABCDEF" UBASC S. ;
```

因为有可能显示 2 进制,所以其中字符串都是 16 位长。其中 1~F,是随便写进的,每次执行点单词时,

其内容都会自动变化,不确定。

看一下点单词的执行过程。

```
: . ( --n0)执行前栈中要有一数  
" 0123456789ABCDEF": (n0--n0 n1 a)  
  给出串地址  
BASC ( n0 n1 a--n2 a1)将数字转换为字符串,长度为 n2  
S. ; (n2 a1-- ) 显示,定义结束。
```

前边介绍了数字和字符串的 I/O 操作。I/O 操作还有一个重要方面是磁盘 I/O。源程序、新生成的系统以及各种数据,都有可能存到磁盘上,如果磁盘功能不完整,一个软件使用起来就很困难。

从原理上,FORTH 是自成系统的,可以包含磁盘存取,不再需要操作系统。然而实际上,各种磁盘存取动作相当复杂,有关程序的编写也就很麻烦。既然已经有可利用的 DOS,就不必再去费力编写,直接调用即可。另一方面,为了能利用 DOS 下其他软件系统的资源,FORTH 系统应该与 DOS 兼容,因此真正自成操作系统的 FORTH 版本并不多。讲座中介绍的 D1.0 和 PC/FORTH 就都借用了 DOS。由于 APPLE 机和 PC 机的 DOS 不一样,两个版本的 FORTH 在磁盘 I/O 方面差别也就很大。这里还是主要介绍 D1.0 版本。

前几讲中介绍过的有:

DAPP 把磁盘文件读入编辑缓冲区

SAVE 把编辑缓冲区中的文件存入磁盘

DAPP 的功能是把磁盘文件从磁盘中读出,加在编辑缓冲区原有内容之后。如果不再需要原有内容,可以先执行 CLR,清除缓冲区。也可以定义一个词:

```
: DREAD CLR DAPP ;
```

DREAD 的功能与 BASIC 中的 LOAD 相同,会刷新缓冲区内容。

假定有一个磁盘源程序文件名为 SS,想将它编译进词典,就可以

```
DREAD SS✓ (读入文件)
```

```
LOAD✓ (将其编译)
```

然后就可以执行 SS 中的词了。但这样可能会有一点麻烦:编辑缓冲区之中可能原来有一部分有用的内容,此时已被 SS 冲掉。为了解决这个问题,系统提供了一个单词,名为 DLOAD,它会将磁盘中源程序直接编译进词典而不通过编辑缓冲区,因此不会影响缓冲区中原有文件。比如还是编译 SS,就这样:

```
DLOAD SS✓
```

除了编译速度比 LOAD 慢些,还是很方便的。这个词还有一个优点,就是它可以包含在其他源文件中,类似于 C 语言中的 INCLUDE。一些汇编词的源程序就用到了它。汇编器平时是不用的,只是在定义汇编词时临时使用。在汇编源程序前端就可以包含一个 DLOAD ASSEMBLER(装入汇编器)然后是编译词的定义。

FORTH 系统还提供了 DOS 命令接口方法。D1.0 版本中有一个词名为 DOS:,在这个词之后可以跟一

行 DOS 命令。比如

```
DOS: CATALOG
```

会将目录列出来,与 DOS 下一样。

列目录和删除文件是两个常用磁盘操作。词典中提供了两个词,不用 DOS;而可以直接执行。请看它们的定义:

```
: CAT STDOS CNT ." CATALOG"  
CR STNRM ;  
: DELETE STDOS CNT GNM ." DELETED"  
NME S. CR STNRM ;
```

定义中 STDOS 和 STNRM 两个词分别是接通和断开 DOS, CNT 是发出一个 control-D 命令,这是 DOS 命令的前导字符。GNM 是键盘接收一个文件名,然后将其存入名为 FNM 的数组变量之中。NME S. 是把 FNM 之中的文件名输出。在这里与 DELETE 合起来组成一个 DOS 命令。

词典中还提供了一组对文本文件操作的词,分别是:

```
OPN  打开文件  
RD   读文件  
WT   写文件  
CLS  关闭文件  
GNM  读取文件  
FNM  变量,存文件名。30 字节长
```

对这些词的使用,要涉及到 DOS 的内容,限于篇幅,无法深入讨论了。

在 PC/FORTH,有一套 DOS 功能调用词和文件存取功能词,比较全面,这里不介绍了。

苹果机和中华学习机上还有一些接口,比如游戏口、录音口等。这些接口的 I/O 操作只是对与之相应口地址的读写就可以进行操作。不过要注意,需要使用字节操作词 C@ 和 C!。比如扬声器的口地址为 C030,对这个地址读一次,电路(其控制输出电路)就翻转一次,按一定频率进行读,扬声器就会发出一定频率的声音来。前边介绍的 INKEY\$ 这个词,也利用了对口地址的读写。对口地址进行操作,在程序上往往是非常简单,但要正确使用,需要对硬件有一定了解才行。另外,在某些情况下,要求速度较高,这样的单词要由 FORTH 中的汇编来编效果才好。用 FORTH 来编这类词,往往是很方便的。

第十一讲 编程实例

在前几讲中,介绍了堆栈、运算、程序结构及 I/O 等 FORTH 语言各方面的基本内容。由于这个讲座是为初学者而办的,一些比较深入的内容,如汇编、数据结构、控制编译及各种高级技巧,就无法介绍了。

有些读者看过前几讲之后,可能会提出这样的问题:完整的 FORTH 程序是什么样的?问题似乎有些奇怪。一方面,对于熟悉 FORTH 语言的来说,不会提出这样的问题;另一方面,对于只熟悉其他语言的初学者,这个问题却是实实在在,需要弄清楚。

为什么这么说呢?先从熟练的 FORTH 使用者的角度看看。

举一个最简单的例子。在 D1.0 版本中,LIST 是列出词典目录,现在想使用 WORDS 这个词名完成这个功能,要这样定义:

```
: WORDS LIST ;
```

尽管简短,它却是一个完整的程序,可以编译和执行。从键盘打入这一行内容,回本之后,这个词已被编译入词典,系统已具备这个词的功能,可以使用 WORDS 了。这就是一个完整的程序。

从严格意义上说,每一个单词的定义,对于 FORTH 来说,都是一个完整的程序。每个词一般都很少,可以一个词、一个词地定义。由此系统的功能逐步增强,直到具备了你的需要的功能为止。

以上的解释,对于初学者来说,并不解决实际问题。一般情况下,应用程序不会象上述 WORDS 那样简单,往往要定义出多个词,综合考虑各种因素才行。这就必然涉及一系列问题。FORTH 语言有其自身性质,需要多方了解,多加练习才能熟练掌握。

在具体举例之前,先讨论一下 FORTH 语言与其他语言不同的设计倾向。

拿输出作例子,比如 BASIC 语言中打印要用 PRINT 语句,可以用来打印字符串、数和空格等,通过不同的格式、不同的函数来确定所打印的不同的内容和方法。而在 FORTH 语言中,显示各种内容要用不同的词,比如:

- (n--) 显示单精度有符号数
- U. (u--) 显示单精度无符号数
- R(n₁n₂--) 右对齐显示单精度有符号数
- D. (d--) 显示双精度数

还有其他输出词,就不一一列举了。在 FORTH 中,每个单词一般比较小,所执行的功能比较单纯,单词是一个个模块,数量较多。在其他语言中,语句较少,每个语句功能比较复杂。这种差别,在使用不同语言编程时,会造成不同的设计倾向。

下面通过一个实例看看一组程序是怎么编出来的。这是一个 FORTH 语言的经典示例。笔者根据讲座的实际情况把它改造了一番,利用自上而下的设计方法,逐步设计出一系列小程序,实际生成时,再按照先定义后使用的原则,自下而上生成。

设想现在要为一台全自动家用洗衣机编出洗衣程序。先介绍一下背景情况。

首先,使用汉语来定义新词,能使用汉语编程的版本目前已有多种。

其次,介绍假设的硬件概况。主机使用 APPLE 机或中华学习机,软件版本使用 D1.0 版本,为了能对硬

件进行控制,假定已经做好一块控制接口卡,插在5号槽口。它的外设接口地址为COD0H~CODFH,共16个地址,比如COD0为控制漂洗电机的口地址,向其中写入1,漂洗电机就旋转,向其中写入0,电机就停止。其他口地址的功能也与此相似。应该说,从硬件的角度看,这种接口卡不难设计。

先定义几个口地址常数。

```

COD0  CONST    漂洗电机
COD1  CONST    甩干电机
COD2  CONST    注水开关
COD3  CONST    排水开关
COD4  CONST    水位上限开关
COD5  CONST    无水开关
COD6  CONST    洗衣粉活门
COD7  CONST    蜂鸣器

```

洗衣机所执行的动作,有漂洗、去污(加洗衣粉漂洗)、甩干、注水和排水等。洗衣结束时还要发声报信。

漂洗和甩干各自使用一台电机,由程序控制其旋转和停止。

注水和排水可以由电磁阀控制,也可以使用水泵。无论用水泵还是电磁阀,在程序控制上都一样,只控制电源通断。注水时,当水位涨到一定高度,会使水位上限开关接通,程序检测到开关已接通,就停止注水。排水时也有一个检测开关,当水排空时,无水开关就会接通,程序检测到这个开关接通,就停止排水。

去污时要加洗衣粉。洗衣粉放在一个小盒子里,下面有一个活门,打开活门就加入一定数量的洗衣粉,然后关闭活门。这也是由程序控制的。

下面就先来看看注水这个过程如何编程。请看程序:

```

: 注水 接通注水开关
BEGIN 水位到上限?
END 断开注水开关 ;

```

这个词执行什么功能,有FORTH知识的人一看便知。它先接通注水开关,开始注水,当水位到上限就关闭注水开关,停止注水。

“注水”这个定义当中有三句话:“接通注水开关”、“水位到上限?”及“关闭注水开关”。实际上它们分别是三个单词。虽然这三个词如何定义不知道,但读者已经能感受到,对于FORTH语言,如果对词名定义得好,可以使程序“明白如话”,这是许多程序员所追求的一种境界。

再来介绍一下“注水”当中的三个词如何定义,在此之前先定义两个词:

```

: 接通 ( a-- ) 1 SWAP C! ;
: 断开 ( a-- ) 0 SWAP C! ;

```

这两个词的功能,分别把1和0写入a地址当中,也就能使电机等设备接通和断开电源。a地址在这里是前边定义的几个常数,如“漂洗电机”、“注水开关”等。接下来:

```

: 接通注水开关 注水开关 接通 ;

```

```

: 断开注水开关 注水开关 断开 ;
这两个词的含义和定义方法不必再解释。再看:
: 水位到上限? ( --f)
水位上限开关 C@ ;

```

这个词放在“注水”的循环中,就不断地检测水位上限开关口地址中的数,水位涨到一定高度,就能使开关接通,执行“水位到上限?”时就会在栈上放上一个1标志,于是退出循环,继而使注水停止。

对于这几个词,了解FORTH的人很容易看懂。它们都不复杂,简单到使人怀疑它们能完成什么象样的功能。然而它们确实能执行预定功能,从词名上就能看出程序能干什么。

“注水”这个词也可以换一种方法定义,在风格上,更象使用其他语言。

```

: 注水 1 注水开关 C!
BEGIN 水位上限开关 C@
END 0 注水开关 C! ;

```

这样定义的好处是,效率高,总体上程序量小;缺点是,当程序规模比较大时,就不那么便于理解了,需要详细注释才行。

好的程序该是什么样的,有许多标准,其一是程序要易于理解,这一般需要注释。有这样的说法:“最好的注释就是程序自身”,当然,前提是,程序自身要能易于理解。请读者认真体会上述两种设计倾向的差别。

现在来看看这台洗衣机在洗衣时做什么工作:

```

: 洗衣 漂洗 去污 漂洗 漂洗 漂洗 甩干 洗毕响铃 ;

```

这是常见的洗衣过程,要做什么,从程序上看一清二楚,不必再解释。如果“漂洗”、“去污”、“甩干”和“洗毕响铃”这几个词都已存在,现在就可以结束设计过程了。实际上还没有,应该事先定义出来,先设计出总体过程,再把它分解,设计出低层部分,直至能够实现为止,这就是自顶向下的设计方法。

接下来就来看看“漂洗”如何定义

```

: 漂洗 注水 转动漂洗电机 5分钟
排水 ;

```

其中“排水”的定义方法与“注水”相似,由于已有定义“注水”的经验,以下几个词就按照生成顺序自下而上定义出来。请看:

```

: 接通排水开关 排水开关 接通 ;
: 断开排水开关 排水开关 断开 ;
: 水已排空? 无水开关 C@ ;
: 排水 接通排水开关
BEGIN 水已排空?
END 断开排水开关 ;

```

至此,注水和排水已经设计过了。再来看看“转动漂洗电机5分钟”如何设计。

```

: 转动漂洗电机 5分钟
启动漂洗电机 等待 5分钟
停止漂洗电机 ;

```

其中包括两个词,使漂洗电机启动与停止。

: 启动漂洗电机 漂洗电机 接通 ;

: 停止漂洗电机 漂洗电机 断开 ;

“等待 5 分钟”还不知如何定义,其实就是简单的延时,由于其他地方还会用到延时,下面一并给出它们的定义。

: 秒等等 (秒数——)

0 D0 600 0

D0 I DROP LOOP
LOOP ;

其中 600 是随便给出的。循环次数到底该是多少,需要经实际测试、调整、调整好之后,输入:

32 53 5 秒等待 + .

加法之前插入了“5 秒等待”,要等 5 秒才给出结果。从外部看,好象这个加法需要做 5 秒钟才能完成。

一分钟等于 60 秒。可以利用“秒等待”定义出“分等待”:

: 分等待 (分钟数——)

0 D0 60 秒等待 LOOP;

洗衣机程序中会有几种不同时间的延时,下面是它们的定义:

: 等待 5 秒钟 5 秒等待 ;

: 等待 3 分钟 3 分等待 ;

: 等待 5 分钟 5 分等待 ;

至此,“漂洗”中各环节都定义过了。接下来,再定义出“去污”:

: 去污 加洗衣粉 漂洗 ;

与“漂洗”不同之处是先要加洗衣粉。洗衣粉装在一个小盒子里,下面有一个活门,打开这个活门,就可以使洗衣粉通过重力漏到洗衣桶之中,于是就可以这样定义:

: 开启活门 活门开关 接通 ;

: 关闭活门 活门开关 断开 ;

: 加洗衣粉 开启活门 等待 5 秒钟 关闭活门 ;

只阅读程序,就能知道其工作过程。

“洗衣”中要用到的“漂洗”和“去污”都已设计过,再来定义“甩干”。

: 启动甩干电机 甩干电机 接通 ;

: 停止甩干电机 甩干电机 断开 ;

: 甩干 启动甩干电机 等待 3 分钟

停止甩干电机 ;

到此为止,只剩下“洗毕响铃”没有定义。

: 响铃 蜂鸣器 接通 ;

: 停止响铃 蜂鸣器 断开 ;

: 洗毕响铃 5 0

D0 响铃 等待 5 秒钟

停止响铃 等待 5 秒钟

LOOP;

“洗毕响铃”会发出 5 响间断的声音,提醒你,衣服已经洗完。

至此,“完整”的洗衣程序已设计完毕。前面介绍的

过程,主要使用了自上而下的设计方法。这种方法在其他语言中也广泛应用,但多数语言的模块化、结构化程序不象 FORTH 这么高,设计上就很受限制。

实际生成单词时,没有被定义过的词,系统是不认识的,系统对每一个单词,一般要求先定义后使用;而且以上这些词在设计时其顺序也有些凌乱,因此需要调整一下顺序才行。下面就按先定义后使用的原则,把这些词重新排列,同时也能使初学者看到一组“完整”的程序是什么样子的。

COD0 CONST 漂洗电机
COD1 CONST 电干电机
COD2 CONST 注水开关
COD3 CONST 排水开关
COD4 CONST 水位上限开关
COD5 CONST 无水开关
COD6 CONST 洗衣粉活门
COD7 CONST 蜂鸣器

: 秒等待 (秒数——)

0 D0 600 0

D0 I DROP LOOP
LOOP ;

: 等待 5 秒钟 5 秒等待 ;

: 分等待 (分钟数——)

0 D0 60 秒等待 LOOP ;

:等待 3 分钟 3 分等待 ;

:等待 5 分钟 5 分等待 ;

:接通 (a——) 1 SWAP C! ;

:断开 (a——) 0 SWAP C! ;

:接通注水开关 注水开关 接通 ;

:断开注水开关 注水开关 断开 ;

:水位到上限? 水位上限开关 C@ ;

:注水 接通注水开关

BEGIN 水位到上限?

END 关闭注水开关 ;

:接通排水开关 排水开关 接通 ;

:断开排水开关 排水开关 断开 ;

:水已排空? 无水开关 C@ ;

:排水 接通排水开关

BEGIN 水已排空?

END 断开排水开关 ;

:开启活门 活门开关 接通 ;

:关闭活门 活门开关 断开 ;

: 加洗衣粉 开启活门 等待 5 秒

关闭活门 ;

:启动漂洗电机 漂洗电机 接通;

:停止漂洗电机 漂洗电机 断开;

:转动漂洗电机 5 分 启动漂洗电机

等待 5 分 停止漂洗电机 ;

:漂洗 注水 转动漂洗电机 5 分钟

排水 ;

(下转第 21 页)

链式存储及树形结构

北京大学 夏晓东

上一讲主要介绍了数据结构的基本概念,认为数据结构包括数据的逻辑关系、存储结构以及数据的运算三个方面。我们还给出了以顺序方式存储的几种简单的数据结构,它们是顺序表、栈、队列,并且引入了抽象数据类型的概念。

这一讲主要讨论链式存储方式以及用链式存储方式组织的树形结构及相应的算法。本讲座只能作为《数据结构》课程的入门,有关其他数据结构以及相应算法、算法分析、应用等方面的内容请读者阅读相应的教材。

一、链式存储方式

上一讲介绍的几种数据结构如顺序表、栈和队列,都是采用顺序方式在内存中存储的,显然其存储密度高;但对顺序表进行插入和删除往往造成大量数据的移动(确保仍然是顺序存储),效率比较低。另外,顺序存储的结构要求占用连续的存储空间,存储分配只适于预先在编译时进行(静态分配)。链式存储方式则可以克服以上的不足,适用于在运行时动态变化较大的数据结构。也就是说,链式存储方式容易实现动态分配,但必须有一个相应的存储管理系统。

以链式存储的线性表,我们可称之为链表。由于有不同的链接方式,又有单链表,双链表,循环链表等等之分。这里,我们只介绍单链表。

因为链式存储的线性表,其相邻结点在存储上并不一定相邻,所以与顺序表不同,单链表中的每个结点都要有一个指针域,以指向相邻结点(通常是后继结点)。如图1所示:

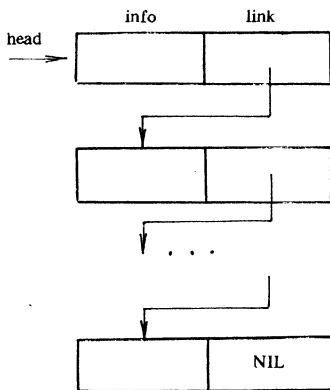


图1 单链表的存储结构

单链表结构可用 PASCAL 语言描述如下:

```
TYPE pointer = ^ node;
node = RECORD
    info: datatype;
    link: pointer
END;
```

VAR

head: pointer;

其加 datatype 是结点中的数据类型,pointer 就是指向结点的指针类型。于是,单链表就可定义成 pointer 类型,如上面的变量 head。单链表的最后一个结点由于没有后继,其指针域为空(NIL)。

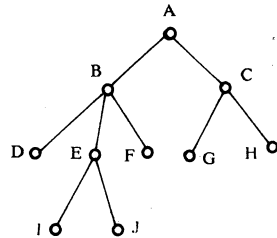


图2 一个树结构

结点 A 是根结点,是结点 B 和 C 的父结点(父母);B 和 C 都是 A 的子结点(子女);D, E, F 都是 B 的子女;...;D, I, J, F, G, H 这些没有子女的结点称为树的叶结点(或终端结点),其他结点为分支结点(非终端结点);B, C 为兄弟结点,D, E, F 也是兄弟结点,...

实际应用中,我们通常用的是树林结构,即零棵或多棵不相交的树的有序集合。如下图所示的具有两棵树的树林。

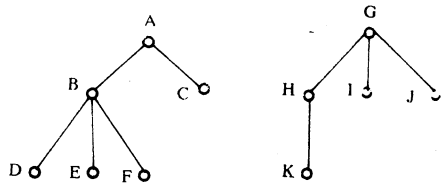


图3 一个树林结构

树结构中,由于每个结点可以有多个子结点,若用链式存储来实现,由于指向子结点的指针个数不等,每

个结点很难统一描述为一个结点类型。然而,任何一个树(林)结构都可以用一个等价的二叉树结构来描述。

二叉树:由结点的有限集合构成,这个有限集合或者为空集,或者由一个根结点及两棵不相交的分别称作这个根的左子树和右子树的二叉树组成(递归定义)。

请读者注意,二叉树和树是两种不同的数据结构,尽管有些概念很相似甚至相同。重要的差别是:二叉树的结点的子树要区分是左子树还是右子树。

显然,二叉树的结点最多只有两个子结点,所以用链式存储很容易实现,即每个结点都看成具有两个指针域(左指针,右指针)。有一种很自然的办法将前述的树(林)结构转换为对应二叉树:凡是兄弟就用线连起来,然后留下父母到第一个子女的连线,去掉父母到其他子女的连线。如将图 3 所示的树林用此法即得到图 4a,使之稍微倾斜就得到图 4b 所示的二叉树。

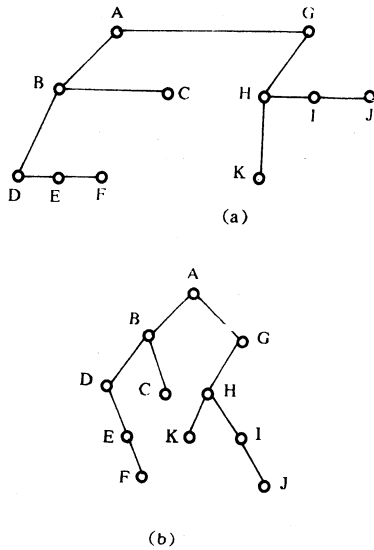


图 4 树林转换为二叉树

二叉树用链接方式表示时,其结点形式为:

left	info	right
------	------	-------

数据域为 info,左指针域 left 指向左子女,右

由于存储结构改变了,上一讲中为顺序表定义的运算也就不适用了。因为结点之间的关系不是通过存储单元的顺序性来表示,而是通过 link 域来实现的,所以插入和删除运算不必进行数据的移动,而只需改变相应的结点的 link 域的值。一个新的问题是:插入新结点时,这个新结点究竟分配在存储区的什么位置?这就给我们带来了存储管理的问题。由于篇幅所限,我们不可能讲解存储管理系统。我们假定已实现了存储管理,可直接使用 Pascal 语言中预定义的两个标准过

程 new 和 dispose。

• 分配——new(p):

假定在需要一个新结点时,只要调用本过程就能分配一个与指针 p 所指向的结点类型大小相同的存储区,其地址放在指针变量 p 中。

• 释放——dispose(p):

与 new(p)相反,当要删除指针 p 所指向的结点时调用 dispose(p)即可释放该结点所占用的空间以利于再分配。

这样,在对单链表进行插入新结点操作时,可调用 new 以分配新结点的存储区;在对单链表进行删除结点的操作时可调用 dispose 以释放该结点所占用的存储区。我们将对单链表可能实施的基本运算用 PASCAL 的 PROCEDURE 或 FUNCTION 头格式罗列如下:

```
PROCEDURE initialize (VAR p:pointer);
    {初始化 p 为一个空表}
FUNCTION length(P:pointer);integer;
    {返回单链表 p 中的结点个数}
PROCEDURE append(VAR p:pointer;x:datatype);
    {在单链表 p 的表尾添加一个内容为 x 的结点}
PROCEDURE insert(VAR p:pointer;i,x:integer);
    {在单链表 p 的第 i 个结点后插入一个内容为 x 的结点}
PROCEDURE delete(VAR p:pointer;i:integer);
    {删除单链表 p 的第 i 个结点}
PROCEDURE getnode(VAR p:pointer;i:integer;
    VAR q:pointer);
    {找到单链表 p 的第 i 个结点 q}
```

请读者自己写出上述对单链表操作的程序。

我们发现,虽然链式存储方式数据的存储密度较低(因指针占用了不少的空间),但这种以存储空间作为代价换取的是操作的灵活性。

二、树形结构

以上所述的数据结构都是线性结构,而树形结构是计算机科学中常用到的一个重要的非线性结构。

所谓树形结构,顾名思义,它类似于自然界中的树;其一,一个结点可以有多个后继结点,一个结点最多只有一个前驱结点;其二,结点之间有着分支和层次的关系。用树形结构能够很自然地表示计算机应用中常用到的嵌套数据。

典型的树形结构有两种,树和二叉树。

树:是这样一种数据结构,(1)有且仅有一个结点没有前驱(此结点称为根结点);(2)除根结点之外,每个结点都有且仅有一个前驱;(3)对任何一个非根结点,都存在一个从根到该结点的结点序列,使得序列中的结点两两相邻。

例如,结点 A,B,C,D,E,F,G,H,I,J 构成以下的树:

指针域 right 指向右子女。这种表示法又称为左右链表示法。

TYPE

```
pointer = ^ node;  
node = RECORD  
  info: datatype;  
  left, right: pointer  
END;
```

VAR

```
t: pointer;
```

于是,可用一个指向根结点的指针(如上面的 t)来表示一个二叉树。

三、树形结构的周游

树形结构的周游是指,按一定次序系统地访问树形结构的所有结点,使每个结点恰被访问一次。由于(树)林可以转换为对应的二叉树,我们只介绍二叉树的周游。

周游二叉树的方式主要有三种:

(1)前序周游

- a. 访问根结点;
- b. 前序周游左子树;
- c. 前序周游右子树;

例如,对图 4b 所示的二叉树进行前序周游所访问结点的顺序为: A B D E F C G H K I J。

(2)后序周游

- a. 后序周游左子树;
- b. 后序周游右子树;
- c. 访问根结点;

例如,对图 4b 所示的二叉树进行后序周游所访问结点的顺序为: F E D C B K J I H G A。

(3)中序周游

- a. 中序周游左子树;
- b. 访问根结点;
- c. 中序周游右子树;

例如,对图 4b 所示的二叉树进行中序周游所访问结点的顺序为: D E F B C A K H I J G。

上述对二叉树的周游方法都可用递归过程来实现。下面给出中序周游二叉树的递归过程,假设结点的数据域为字符型。

TYPE

```
nodeptr = ^ node;  
node = RECORD  
  info: char;  
  left, right: nodeptr  
END;
```

PROCEDURE infix_trav(t: nodeptr);

BEGIN

```
IF t <> NIL THEN
```

```
  BEGIN
```

```
    infix_trav(t^.left);  
    writeln(t^.info);  
    infix_trav(t^.right)
```

```
  END
```

END;

也可用栈的操作来消除递归,如:

PROCEDURE infix_trav(t: nodeptr);

VAR

```
  ok: boolean;
```

BEGIN

```
  ok := false;
```

```
  REPEAT
```

```
    WHILE t <> NIL DO
```

```
      BEGIN
```

```
        push(s,t); t := t^.left;
```

```
      END;
```

```
      IF empty(s) THEN ok := true
```

```
      ELSE
```

```
        BEGIN
```

```
          pop(s,t);
```

```
          writeln(t^.info);
```

```
          t := t^.right
```

```
        END
```

```
      UNTIL ok
```

```
    END;
```

以上用到的栈的操作请参阅上一讲的内容,这里只需将栈的元素类型改为指向 node 的指针类型 nodeptr,并将用 FUNCTION 实现的 pop(s)改为用 PROCEDURE pop(VAR s: stack; VAR t: nodeptr)实现,弹出的元素由指针 t 返回。

用栈实现二叉树的中序周游算法可这样理解:

1. 从根结点出发沿左指针往下走,同时将遇到的结点压入栈中,直至左指针为空,到达“最左下”结点;
2. 如果栈为空,表明当前周游的二叉树是空树,并且整个二叉树周游结束;否则弹出栈中最上面结点,进行访问,然后进入右子树,回到第 1 步。

对二叉树的基本操作除上面周游外至少还应有:

1. 构造单结点的二叉树 maketree

PROCEDURE maketree(x: char; VAR t: nodeptr);

BEGIN

```
  new(t);
```

```
  t^.info := x;
```

```
  t^.left := NIL;
```

```
  t^.right := NIL
```

END;

显然,构造一个二叉树可通过不断调用 maketree 来实现,如

```
  maketree('A', t);
```

```
  maketree('B', t^.left);
```

```
  maketree('C', t^.right);
```

构造了只有三个结点的二叉树 t。

2. 删除一个二叉树 deletetree

PROCEDURE deletetree(VAR t: nodeptr);

BEGIN

(下转第 37 页)

T6668 语音处理器手动实验

北京宣武区科技青少年馆(100053) 罗明宽 车金相

编者按:本期只是向读者介绍将 T6668 接成手动操作模式。本栏目将继续向读者介绍在单片机控制下实现语言库的功能。同时本刊将开展在 T6668 征集和评选手动操作模式下的实用电路的活动。

现代计算机技术的不断发展,促进了语音技术的日臻成熟与完善,应用前景十分广阔。为了使读者了解掌握这一技术,现将日本东芝公司生产的语音合成芯片 T6668 介绍给读者。

T6668 语音处理器是一种专门用于语言信号的记录和再生的超大规模 CMOS 集成电路。该电路采用动态存储器容量可达 1M 位。用 T6668 组成录放系统,可由微机等进行 CPU 控制和外部开关人工控制。因此东芝公司的 T6668 以它较强的电路功能和简单的外围电路而倍受大家欢迎,被广泛应用于广告、智能仪器和自动录音电话机中。

一、T6668 的引脚分布及作用

T6668 的引脚分布如图 1。

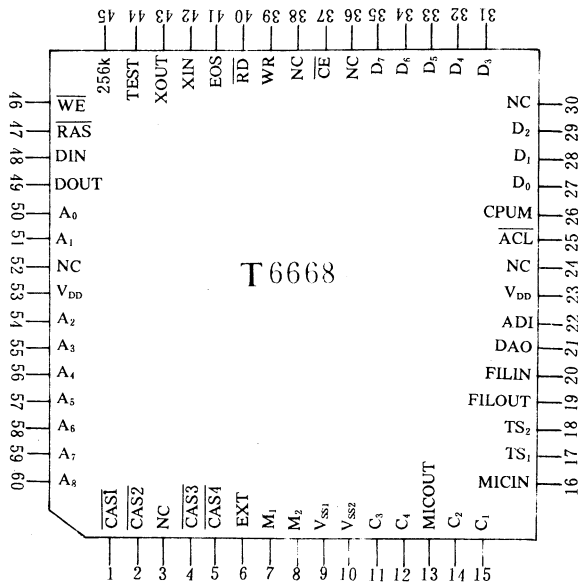


图 1

图 1 中, T6668 各引脚的名称, 序号及功能如下: CAS1(1)、CAS2(2)、CAS3(4)、CAS4(5): 为输出端,

低电平有效,以 $\overline{\text{CAS1}}$ 的顺序控制第一块至第四块动态 DRAM 的列选通。

EXT(6): 为输出端, T6668 内部地址计数器最高位溢出时, 有信号输出, 用于测试电路的输出引脚。

M₁(7)、M₂(8): 为输入端, 用于确立外接 DRAM 芯片的片数: 即:

片	引脚	M ₁	M ₂
1		L	L
2		L	H
3		H	L
4		H	H

L 为低电平
H 为高电平

V_{SS1}(9)、V_{SS2}(10): 为电源端, 用于连接电源的负端。

C₃(11)、C₁(12): 连接耦合电容端。

MICOUT(13): 为内部话筒放大器的输出端, 输出在 1/2V_{DD} 这个中心上下摆动的振荡信号。

C₂(14)、C₁(15): 为内部话筒放大器的耦合电容的连接端。

MICIN(16): 为内部话筒放大器的输入端, 驻极话筒的电信号经过电容由此输入。

TS₁(17)、TS₂(18): 为芯片测试端, 用户可以闲置不用。

FILOUT(19)、FILIN(20): 内部重放带通滤波器的输出及输入端。

DAO(21): 为语音合成电路声音输出端, 录制期间, 可获得输入声音的监听输出, 也是一个(在 1/2V_{DD}) 上下摆动的振荡信号。

ADI(22): ADM 分解(A/D)电路的模拟输入端。

V_{DD}(23、53): 为 +5V 电源接线端。

$\overline{\text{ACL}}$ (25): 为复位信号输入端, 低电平有效。

CPUM(26): 为操作模式转换端, 手动操作, 该脚接低电平, 若为高电平为 CPU 控制下的操作。

D₀(27)、D₁(28)、D₂(29)、D₃(31)、D₄(32)、D₅(33)、D₆(34)、D₇(35): 在 CPU 控制下的操作, D₀~D₇ 作为双向数据总线, 以便使 CPU 和 T6668 的命令和数据传输。在手动操作模式下, 则按下述方法使用:

① D₀~D₃ 输入端用作段选, 通过 BCD 码选择 16 段。若悬空不接, 不分段可连续使用。

② D₄ 为启动输出端, 当输入 20ms 以上的正脉冲

时,便启动录和放。

③D₅ 为停止输入端,当此脚输入 20ms 以上的正脉冲,电路停止录和放。

④D₆、D₇ 为比特率选择输入端,具体放置如下表:

引 比 特 率	D ₇	D ₆
8kbps	L	L
11kbps	L	H
16kbps	H	L
32kbps	H	H

L 为低电平
H 为高电平

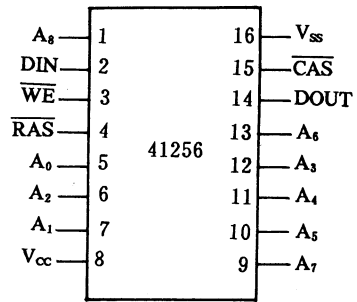


图 3

下面将应用电路作如下说明:

(一)整体电路结构

在录音时,由驻极话筒输入的语音信号,经 T6668 内部 ADM 分析合成后,变成数字序列信号进行量化处理,形成 L 或 H 存储到外接的 DRAM 存储器中去。在放音时,将存储在 DRAM 中语音序列信号经 ADM 解释,经 D/A 转换,重新转换为模拟信号,以便恢复为语音电信号,再通过外接音频功率放大器、扬声器,最终得到语音的再生。

(二)本应用电路是以手动操作模式进行语音录放实验,所以 CPU_M(26)接为低电平。关于 CPU 控制模式今后向读者介绍。

(三)本应用电路画了四片 DRAM 存储芯片,在具体实验时,只用了二片存储芯片。此时只要读者将 M₁ 和 M₂ 两根插针分别接入电源的地线和 +5V 端即可。如果读者想获得更长时间的录放,可按管脚说明将 M₁ 和 M₂ 的电平作相应变换即可。

(四)本电路选择:8kbps(比特率),所以 D₆(34)、D₇(35)都与地线相接。

(五)在手动操作模式下,D₀~D₃ 为段选控制端。本电路 D₀~D₃ 四个管脚全部悬空,为不分段,连续使用。如果读者想实验段功能,可将 D₀~D₃ 的四根插针作相应电平变换。

(六)本电路使用了 256K 位的 DRAM 存储器,故 256K(45)接在 +5V 电源端。

(七)EOS(41)“语音终止”的信号输出端,根据此脚功能利用 LED 发光时表示正在录放中,录放停止 LED 熄灭。

我们在实验 T6668 语音芯片时,参考了有关电路,按这些电路进行实验效果很不理想,杂音很大,经过我们的修改实验效果较为理想。不过希望读者在选用不同的话筒时,认真仔细调整 W₁ 的阻值,使其工作在最佳点上。另外还希望读者认真调整 C₁₀ 的容量,过大声音发沉,音量变小,过小声音变大,杂音也随之增大,在录音时,最好将音量电位器 W₂ 调到最小,以免引起在录音时,经扬声器反馈到话筒引起自激。

三、调试与使用

读者根据电路原理图在线路板上将元器件焊好。

$\overline{\text{CE}}$ (37):在手动操作模式下,该脚用于控制 DAO 有无信号输出,在放音时该引脚接高电平;DAO 无信号输出,反之有信号输出,但在录音时,该脚必须为低电平。

WR(39):CPU 控制时,为写入脉冲输入端。在手动操作模式下,该引脚用于决定是录音还是再生,当该脚接高电平时,进行录音,接低电平为再生。

$\overline{\text{RD}}$ (40):手动时该引脚悬空不用,在 CPU 控制时,为该脉冲输入端。

EOS(41):“语音终止”信号输出端。录放开始变为低电平,结束时输出高电平。

XIN(42)、XOUT(43):为振荡电路的输入和输出端,陶瓷振子可在 640kHz~655kHz 之间选择。

TEST(44):测试电路端,可悬空不用。

256K(45):用来选择 DRAM 容量控制端,当该脚接成低电平,选 64K 位的 DRAM,若接高电平则选择 256K 位的 DRAM。

$\overline{\text{WE}}$ (46):写脉冲输出端,低电平有效,接动态存储器的 $\overline{\text{WE}}$ 端。

$\overline{\text{RAS}}$ (47):行地址选通信号输出端,低电平有效,接 DRAM 的 $\overline{\text{RAS}}$ 端。

DIN(48):为数据输入端,接动态(DRAM)的数据输出端(DOUT)。

DOUT(49):为数据输出端,接 DRAM 的数据输入端。(DIN)

A₀(50)~A₈(60):为地址输出端,接 DRAM 地址输入端,其中 52 脚为空脚,53 脚为 +5V 接线端。

NC(3、24、30、36、38、52):为空脚,在电路中悬空。

二、T6668 语音芯片应用电路

T6668 语音合成芯片构成的语音录制与再生系统,可以选择使用由微处理器管理控制的 CPU 操作工作模式或由外接开关组成一人工手动操作的工作模式。每种模式各有其自身操作规则和要求,读者必须注意。

下面简单的应用电路以手动操作模式,其电路原理图见图 2。

DRAM 动态存储器 41256 管脚图如图 3。

焊好后认真检查焊点是否因过大造成焊点间短路,排除短路故障,经检查焊接无误,可通电进行调试。

(一)调试:①接通电源,将音量电位器 W_2 调到适当位置,此时扬声器发出声响,说明电路基本正常,若无声响,应先查电源是否接通,如电源接通再查复位按键是否接触不好。最后检查扬声器是否完好。若无任何故障,扬声器是会正常出声的。

②将录放开关 K 打开,按一下启动按键,此时 LED 应发光。说明电路处于录音工作。适当调整音量和音质电位器 W_2 和 W_1 ,对准话筒讲话,从扬声器可监听到说话的声音。此时按下停止键,LED 应立即熄灭,说明电路停止录音状态电路录音正常。

③将录好的语音信号存放在 DRAM 中,将录放转换开关 K 闭合,按一下启动键,LED 发光,同时扬声器发出录好的语音,说明电路语音再生正常。如果录放不正常(但 LED 亮灭正常),应检查动态 DRAM 是否管脚方向插反,DRAM 是否有问题。

通过上述调试电路正常后,仔细调整 W_1 ,使录音处于最佳状态,即不失真,此时用万用表测量使用 W_1 的实际阻值,将该阻值焊在 W_1 的位置上,此时电路调整完毕,便可投入使用。

在调试过程中,读者千万注意,集成块取下插上一定在断电的情况下进行。需要焊接时将烙铁脱离电源,利用余热进行焊接。

(二)使用

电路一切正常,便可进行录放。

1. 录音过程:

首先将音量电位器 W_2 调至最小,按一下复位键,将录放开关 K 闭合,按一下启动按键,此时读者可对准话筒进行录音,若想停止录音可按一下停止键,录音便结束,如果没按停止键,到一分钟左右自动停止录音。

2. 语音再生

首先将录放开关 K 打开,实际上已处于放音状态,调解音量电位器,按一下启动键,此时放音开始。

此电路可随时录也可随时放,使用十分方便,如果读者稍动脑筋,很可能研制出有实用价值的产品来。

我们将在今年底开展学装“固体录音机”制作活动,感兴趣者可来信索取报名表,来信请寄:北京西城电子仪器仪表厂三里河西口 18 号楼,邮编:100045,联系人:张文奇。供应套件:125 元/每套,另加邮资费 10 元。

MCS—51 系列单片机中的加密措施

陕西西安交通大学电气系(710049) 张 戟

目前大多数单片机的应用程序没有采取有效的加密措施,致使单片机上的应用程序被非法复制,严重损害了开发者的利益。随着单片机在自动控制、测量、智能化仪器仪表等领域中的广泛应用,单片机上应用程序的加密已成为急需解决的重要问题。本文将介绍几种单片机中的实用加密措施。

单片机中的加密措施一般可以分为两类:硬件加密和软件加密。硬件加密是指用器件来实现的加密,软件加密是指在片内 RAM 或结构化、模块化结构的程序中加入保密字达到加密目的。下面将从这两方面加以阐述。

一、利用在单片机片内 RAM 中加保密字的方法

对于一些具有掉电保护功能的单片机,用户可以通过在单片机片内 RAM 中设置保密字的方法实现加密保护。保密字设置好后,当系统运行时,应用程序通过经常测试(对保密字的算术和逻辑运算)保密字的正确性来控制应用程序的流向,实现加密目的。

下面介绍一段简单的在单片机 RAM 中设置保密字的程序。

```
ORG 2000H
SJMP SET_PASSWORD ;转向设置保密字入口
ORG 2003H
SJMP PROTECT ;转向掉电保护中断服务入口
```

```
ORG 2040H
SET_PASSWORD:
MOV 30, #XXH ;设置保密字(保密字个数和内容
              由用户根据需要决定)
MOV 31H, #XXH
:
MOV IP, #01H ;设置外部 0 中断(掉电保护中
              断)为高优先级中断
SETB EX0 ;允许外部 0 中断
WAIT:NOP
SJMP WAIT ;等待掉电中断
PROTECT:
ORL PCON, #02H ;设置成掉电工作方式
END
```

把设置保密字程序写入 EPROM 中,然后插入单片机系统,使系统运行后再掉电,这样保密字就设置好了。然后拔掉该 EPROM,插入用户应用程序的 EPROM。用户可以在全程序中加入一系列的运算和判断程序来控制程序的流向。

二、利用在模块化程序中加保密字的方法

在单片机的应用程序中多采用结构化和模块化的程序设计方法。因此,通过在程序中加保密字破坏程序的模块化结构,利用循环和跳转指令隐藏程序的正常执行次序,就能达到加密的目的。

1. 在两个程序模块的连接处加入保密字。

```

        ⋮ } BLOCK1
        ⋮
2040  22    RET
2041  75    DB  75H
2042  E0    MOVX A,@DATA } BLOCK 2
2043  F5 74  MOV  74H,A
2045  22    RET
        ⋮
        ⋮

```

在两个 BLOCK 之间加入的保密字为 75H, 用一般的反汇编将得到一段与原程序完全不同的程序。

```

        ⋮ } BLOCK 1
2040  22    RET
2041  75 E0 75 MOV E0H,#F5H } BLOCK 2
2044  74 22    MOV A,#22H
        ⋮

```

由此可见, BLOCK2 的指令内容已面目全非, 这样就达到了防反汇编的加密目的。

2. 利用两个模块共用某些字节达到加密目的。

```

        ⋮ } BLOCK1
2040  11E0 AJMP 20E0H
2041  E0    MOV A,@DATA
2042  F574 MOV  74H,A
2044  22    RET
        ⋮ } BLOCK2

```

由上面程序可以看出, 2041H 单元的内容 E0H 是 BLOCK1 和 BLOCK2 的共享单元数据。

但是, 反汇编后的程序却为:

```

        ⋮ } BLOCK1
2040  11 E0 AJMP 20 E0H
2042  F574  MOV  74H,A
2044  22    RET
        ⋮ } BLOCK2

```

可以看出, BLOCK2 之中的 MOVXA, @DATA 指令丢失了, 从而使非法用户无法看到完整的原程序, 达到加密目的。

三、利用硬件实现对单片机的加密。

在一些单片机中, 如 MCS-51 系列中的 8751H、8751BH、87C51 和 8752BH 中设置有程序存储器 (EPROM) 封锁线路, 该线路阻止了非法用户的侵入。当封锁位被置位后, 单片机就禁止任何外部对 EPROM 的访问。只有将它擦除后才能撤销封锁状态, 恢复器件的原有功能。除了具有封锁线路外, 还有 32 字节的加密陈列, 用户通过对 32 字节的加密陈列的编程进一步对单片机进行加密。非法用户解密时必须知道这 32 字节的正确顺序, 否则无法解密。

在硬件加密中有一种十分重要的器件——可编程逻辑器件 PLD。它的一个非常有用的特点就是具有良好的防仿制性。在任何一个应用系统中或任何一块电路印制板上, 只要装上一块 PLD, 非法用户就很难仿制。这是因为 PLD 的逻辑功能完全是由开发者自己设计的, 其内部逻辑结构图很难根据其外部特性推导出来。

一种新的 PLD 芯片——可编程门阵列 PGA, 用户使用时无需特殊的写入电路, 只要改变内部 RAM 的信息内容, 就可获得一块新的芯片。还有一种被称为可编程门阵列逻辑器件 GAL 芯片, 其破译的概率只有 $1/256!$ 。

随着微电子技术的发展, 今后硬件电路设计中的标准数字 IC 将被专用集成电路 ASIC 所替代。这样不仅可以增加集成度, 降低成本, 更重要的是可以增加硬件电路的保密性。所以, PLD 将会在硬件电路设计方面得到广泛的应用, 也是单片机加密技术的发展方向。

SCB—Ⅱ型单片单板机详介

SCB—Ⅱ型单片单板机是武汉尚吉电子研究所于 1992 年底推出的新一代单片单板机, 它采用 8098 和 8031 选择式双 CPU 结构。上市后即得到数以千计的客户广泛应用, 现就其硬件及结构、监控软件及软盘软件、配套资料及使用方法、应用领域等情况详细介绍如下:

SCB—Ⅱ的硬件是 8031 或 8098 双 CPU 结构, 若使用 8098 作 CPU 时, 系统可提供 4 路 10 位 A/D、1 路 PWM 输出等 8098 所具备的全部功能。此外 SCB—Ⅱ有 8 路 8 位 A/D 即 ADC0809 和 1 路 8 位 D/A 即 DAC0832; 16K EPROM (即 27128) 和 16K 带有充电电池的掉电保护电路的 RAM, 此 RAM 可用 E²PROM 直接替换。有 8255 扩展的 3 个 8 位并行或位控的 I/O 口; 有 BAX232 构成的单 +5V 供电的标准 RS232 全双工

串行通信口; 键盘和显示部分由 5×5 进口微动开关按键 (也可直接接插软性薄膜开关) 和 8 位 LED 数码显示组成, 用 8279 驱动, 不占用 CPU, 且提高了显示亮度。系统还有抗干扰用的“看门狗”电路。SCB—Ⅱ的外型结构设计, 也切实地面向应用, 它采用整分式结构即有整体式和分体式两种, 有双复位电路, 主板部分可以完全脱离显示键盘板而独立地开发应用, 外引出线达 100 芯, 它将 8031、8098 的全部 I/O、A/D、D/A 并行口等全部引出。

SCB—Ⅱ不仅有全面、典型、规范的硬件, 还有堪称完美的软件设计。主要分为监控软件和软盘软件。监控软件共有两部分四组, 即 8031 的本机板监控和 CRT 监控、8098 的本机板监控和 CRT 监控, 全部监控共计 8K, 装在 27128 内。8031 和 8098 两部监控的功能是一模

一样的。除有：单步、断点、键扫描与键分析、显示、内外存储器读写、数据移动、E²PROM 编程等基本开发功能外，还有与 PC 机直接通信程序、2716~27512 多种 EPROM 的读写高速固化程序、A/D、D/A 子程序、打印驱动程序等实用性很强的多种程序。而 CRT 监控部分的主要功能和本机板监控类同，但它主要将操作控制权交给了上位机（即 PC 机或 286 机等），使你可以在上位机上进行各种开发、直接汇编、存盘等。SCB-Ⅰ 的软盘软件也分 8031 和 8098 两部，功能也一样。主要有 8031、8098 的汇编、反汇编、正反向通信、文件编辑、数据格式转换等模块。若用户需要则还可以提供 PL/M 等其他软盘软件。

SCB-Ⅰ 的配套资料达 80 余万字，由我国著名教授王长胤和武汉尚吉电子研究所所长文军共同编写，并由武汉大学出版社于 1993 年 7 月正式出版发行。全书分上、下两册，定名为《单片单板机原理与应用》，共计 9 章，第一、二、三章介绍了 8031、8098 单片机的结构与原理、指令系统等。第四、五、六、七章为 SCB-Ⅰ 单片单板机的硬件组成原理、使用、监控程序详解和如何开发应用，第八章则结合教学需要提供了单片实验及实验程序。第九章提供了全部的监控程序清单。因此该书既是一本学习单片机的入门教材，也是单片单板机的解剖学习和应用手册。

优良的软、硬件配置，再加上全公开方式的详尽资料使 SCB-Ⅰ 得以有广泛的采用。

首先，由于单片单板机深广的应用，使得其在教学中越来越被重视，又由于 SCB-Ⅰ 的硬件配置全面、规范、新颖，软件实用丰富，资料则完全是面向教学而

编写，本身既可做大、中专教材和自学教材，而使其被上千所大、中专院校应用在教学中，其取代 TP801 的教学已是必然趋势。

SCB-Ⅰ 除主机板外，还可以选配两块功能板即 EPROM 读写板和仿真板。EPROM 读写板和 SCB-Ⅰ 联接后可以对 2716~27512 EPROM 进行读写、复制、校检、高速固化等多种操作，而其配套的仿真板则可以使 SCB-Ⅰ 进行 51、98 两用型仿真，借助地址空间选择功能可以实现零地址在线仿真，并可通过选择实现晶振切换，这些功能和 SCB-Ⅰ 本机板一起则可以构成一套完整的开发系统，且既可以开发 51 系列也可以开发 98 系列，因此 SCB-Ⅰ 被诸多客户用作良好的开发工具。

SCB-Ⅰ 无论软硬件配置设计，还是外型结构设计及资料公开性等方面都切实地面向应用，不仅面向教学应用与开发工具应用，还为用于工业控制及仪器仪表而进行了各方面的专门设计，如“抗干扰”电路、仪表软薄膜开关、数据采集与掉电保护、传送、控制、打印的硬件及软件等等，目前已有 2 千 1 百多客户将其应用在工业系统中。

显然，SCB-Ⅰ 单片单板机的应用必将有助于单片技术在我国的应用与发展，我们热切希望能为我国计算机单片机技术应用而做出自己的贡献。

单位：武汉市尚吉电子研究所

联系地址：武汉武昌珞瑜路 37-3 号

电话：716138 716692 邮政编码：430070

电报挂号：4561 图文传真：(027)716138

联系人：单 部

向您提供用户电费微电脑管理系统

我部新近设计研制了一套用户电费微电脑管理系统（Microcomputer Management System for the charges of Using Electricity 简称 M²SC 系统）。

该系统由一台中央处理主机（高档 PC 机）和若干用户数据分机构成。属于当前先进的分布式微计算机数据采集处理系统之一。

中央处理主机安装在用电主管部门，主要完成如下功能：

- 定期地访问各单位数据分机，取得用电数据；
- 计算各单位的用电费，并进行存储备份处理；
- 打印各单位的电费帐单；
- 根据需要向各单位数据分机发送控制指令；（备选）
- 与上级微机系统联网，以便于交换用电数据信息。（备选）

数据分机安装在各用电单位的变电室或电度表附

近，视具体环境而定。分机能完成以下的功能：

- 计量线路用电量（或自动读表），并保存数据（带掉电保护）；
- 与主机可靠通信，传送用电数据；
- 带数据接口，以备显示或打印机使用；（备选）
- 带限电控制接口（例如可对滞纳电费单位的生活用电进行跳闸）。（备选）

其他功能还可根据使用单位的具体需要更改。

上述 M²SC 系统的最大特点是可靠性高，使用维护简便，投资低廉（无须建网费用）。一旦普及推广，将大大改善提高用电管理的质量和水平。

竭诚欢迎有意者与我部联系合作。

邮政编码：246005

地址：烟台大学计算机应用开发中心

联系人：李凯里

电话：(0535)248995-746

开辟单片机世界新天地

各种类型单片机开发系统——ATD 系列产品

* ATK—C196 高档强功能16位全系列单片机开发系统

该开发系统可对 Intel 公司 MCS—96 系列单片机不同封装芯片 8095, 8096, 8097, 8098, 80C196, 80C198 进行仿真开发, 因而具有很高的性能价格比。

* ATD—98D 8098 单片机开发系统

本开发系统有开发方式和应用方式两种工作方式。在开发方式下可实现编辑、汇编、反汇编、调试以及读写 EPROM 等功能; 在应用方式下, 用户可以使用板上各种资源和接口, 如 8098/8155 资源, 键盘显示/微型打印机接口等, 从而可以脱机独立工作。

* DP-851 最新普及型51单片机自开发系统

可提供多种配置, 既有用于学校教学和科研的系列配套实验板, 又有适用于各行各业的控制的高可靠工业级板, 性能优异, 价格低廉, 使用方便。本公司是北京市单片机应用技术协会的著名产品——DP—851 的生产单位, 提供整机, 办理邮购。

优质服务、终身保修、欢迎认购!

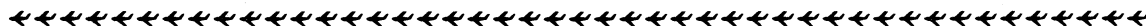
北京弗蓝卡数字系统有限公司是中美合资高科技企业。其合资中方北京市中软计算机研究所是新型的高科技企业, 主要从事信号处理与工业控制领域的高科技新产品的研制开发。经过多年坚持不懈的摸索与拼搏, 已创立著名的 ATD 系列产品: ATD—A100 (3 亿 2 千万次超高速实时处理系统); ATD—C25 (1000 万次高速开发系统); ATD—C30 (TMS320C30 多功能开发系统); ATD—ASP 工作站 (多用 DSP 开发应用平台); ATD—GPS 系列 (车载、船用、手持各类卫星定位系统); ATD—KD 系列微电脑阻焊机控制器。ATD 产品已应用到全国各地国家级“八五”规划项目中, 得到了用户的好评。

北京市中软计算机研究所是国内最早开发、生产单片机开发系统的著名单位, 单片机开发系统使用自己的版权软件。产品达到了工业应用的可靠性等级, 并经北京市新技术产业开发试验区的专家鉴定 (京试鉴字 52 号)。中软所还将不断推出先进的、使用方便功能齐全的单片开发系统, 为广大用户服务。

北京市中软计算机研究所 电话: 203. 3240 254. 1324 传真: 256. 4527

北京弗蓝卡数字系统有限公司 联系人: 高正 李秀莲

地址: 北京海淀区成府路 6 号或北京海淀区五道口东升园 7 条 9 号 邮编: 100083



(上接第 30 页)

```
IF t<>NIL THEN
  BEGIN
    deletetree(t^.left);
    deletetree(t^.right);
    dispose(t);
  END;
```

这是一个后序周游的递归过程, 其中访问结点就是释放结点的存储区。如,

```
deletetree(t);
```

将删除上面构造的二叉树 t。

树形结构在计算机科学中有着广泛的应用, 比如数据检索领域中常用到树形目录, 如平衡二叉树、B+树等等。

CEC-I EPROM 编程电路

湖北黄石高等专科学校实验中心(435003) 齐向东

CEC-I 中华学习机,软件丰富,汇编程序易学,而且显示直观,数据存取方便。用 CEC-I 对 EPROM 编程比用单板机或单片机更具优越性。下面介绍的电路,结构简单,编程安全可靠,可在 CEC-I 中华学习机或 Apple 苹果机上对 2716~27128 等多种 EPROM 芯片进行编程。

原理电路如图所示:电路中应用了一片 8255 可编程接口芯片,在编程软件中对其初始化,可将三个端口 A、B、C 按需要分别设置为数据输入/输出。电路中将 A 口作为数据传送端口与 EPROM 的数据线对应连接;B 口和 C 口中的 6 位用来传送地址并与 EPROM 的地址线对应联接。14 根地址线寻址范围为 16K 字节。C 口中的 PC₆、PC₇ 两位线专门传送控制信息,在软件的作用下,主要用来产生定时脉冲。8255 的 D₀~D₇ 经由双向数据传送电路 74LS245 与主机的数据总线相连接,编程地址、数据及控制信息均由总线传送到 8255 各端口的控制寄存器中。

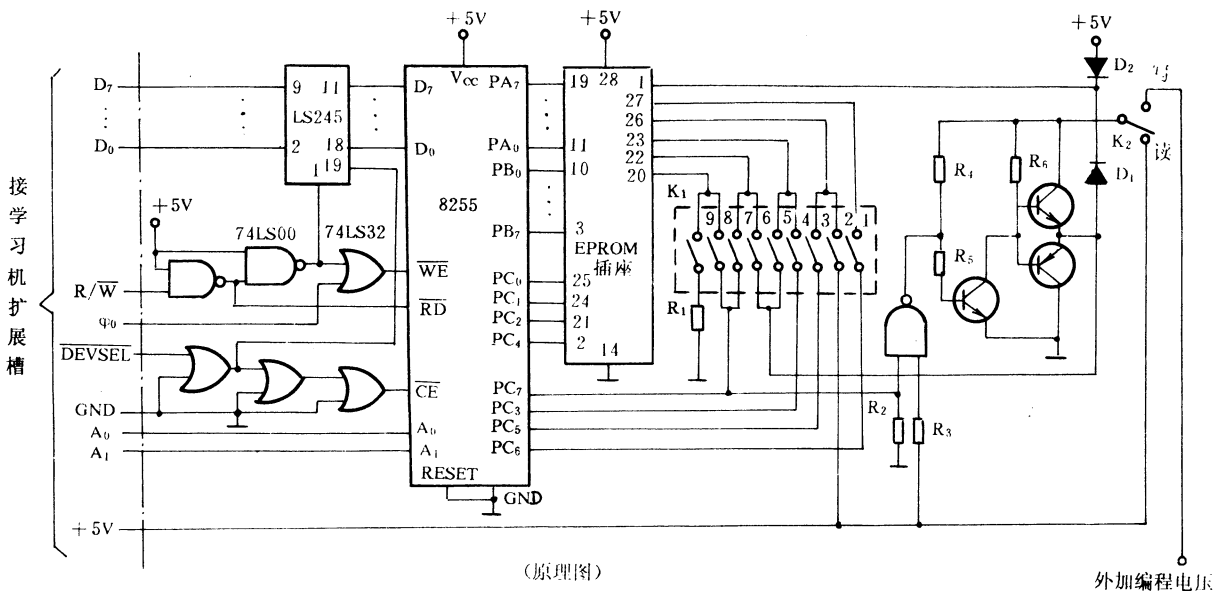
2716~27128 的绝大多数引脚是一一对应的,仅有少数几个引脚的功能不同。另外,2716、2732 有 24 个引脚,2764、27128 有 28 个引脚,用一个 9 位拨码开关通过

改接连线的方式,变换不同 EPROM 芯片的引脚功能,这样处理后,仅调用一组子程序就可以完成对 2716~27128 等多种芯片的读、写和数据比较等操作。

对 EPROM 芯片进行编程操作时,外部编程电压通过由三极管等器件组成的软开关电路加到 EPROM 芯片上,各种芯片均采用脉冲电压方式供电,直接由程序提供的控制信息予以实现。这样处理后,在保证可靠编程的前提下,使编程软件获得了统一。采用软开关电路可以有效地减少浪涌电压对芯片的危害,使编程更加安全可靠。考虑到开关电路的降压作用,外加编程电压应高出正常值 0.5~1 伏。

表(1)、表(2)分别给出 9 位拨码开关在对 EPROM 进行读、写操作时的通断状态。开关接通(ON)为“1”,开关断开(OFF)为“0”。

编程软件由用 BASIC 语言编写的管理程序 GLCX,用汇编语言编写的读 RROM,写 WROM 及数据比较 CPROM 三个子程序所组成。事先应将这些程序存入磁盘。进行读、写操作时,只要运行 GLCX 程序并按菜单选项和正确输入数据即可方便地完成各种操作。需要注意的是:操作中为了便于进入计算机内存查



(原理图)

看数据,地址及读、写字字节数均应输入十六进制数,写操作时的时间常数字要输入十进制数。对2716、2732进行编程时的时间常数字是144;对2764、27128进行编程时的时间常数字是18,这样一来在调用汇编语言子程序时,其中的延时子程序可产生50mS或1mS的字节编程时间。在中文状态下,内存可存放数据的地址范围是\$6000~\$8FFF约8K字节的存储空间,在西文状态下,从\$2000~8FFF的24K字节的的空间可用来存放数据。对EPROM芯片进行编程时,由于不同引脚的芯片共用一个插座,不要将引脚插错。制作的编程卡可直接插在CEC-I的扩展槽中(应保证是1号槽),使用Apple苹果机编程时编程卡插入1号槽。

表(1)

EPROM	编程开关状态
	1 2 3 4 5 6 7 8 9
2716	0 1 0 0 1 0 1 0 1
2732(A)	0 1 0 1 0 0 1 0 1
2764(A)	0 0 0 1 0 0 1 0 1
27128(A)	1 0 1 1 0 0 1 0 1

表(2)

EPROM	编程开关状态	编程电压
	1 2 3 4 5 6 7 8 9	
2716	0 1 0 0 1 0 1 1 0	25V+1V
2732	0 1 0 1 0 1 0 0 1	25V+0.5V
2732(A)	0 1 0 1 0 1 0 0 1	21V+0.5V
2764	1 0 0 1 0 0 1 0 1	21V+1V
2764(A)	1 0 0 1 0 0 1 0 1	12.5V+1V
27128	1 0 1 1 0 0 1 0 1	21V+1V
27128(A)	1 0 1 1 0 0 1 0 1	12.5+1V

下面是编程操作的软件部分:

```

5  REM RWEPROM
10 HOME:PRINT:PRINT
15 PRINT TAB(10);"EPROM 操作选择"
20 PRINT TAB(8);"-----"
25 PRINT TAB(10);"1-----读 EPROM"
30 PRINT TAB(10);"2-----写 EPROM"
35 PRINT TAB(10);"3-----数据核查"
40 PRINT TAB(10);"4-----退出"
45 INPUT "键入选择号:";N
50 ON N GOSUB 350,400,500,100
55 INPUT "还有选择项吗?(Y/N)";G$
60 IF G$="Y" THEN 100

```

```

65 INPUT "是否进入监控?(Y/N)";P$
70 IF P$="N" THEN 100
75 CALL-151
100 END
200 PRINT"-----"
210 PRINT
220 INPUT"数据存放低位地址:";X$;
GOSUB 300:A=Y
230 INPUT"数据存放高位地址:";X$;
GOSUB 300:B=Y
240 INPUT"读(写)字节低位数:";X$;
GOSUB 30:C=Y
250 INPUT"读(写)字节高位数:";X$;
GOSUB 300:D=Y
260 POKE 26,A;POKE 27,B;POKE 6,C;
POKE 7,D
270 RETURN
300 N=LEN(X$):W=1:Y=0
310 FOR I=N TO 1 STEP -1:G$=MID$(X$,I,1):G=VAL(G$)
320 IF ASC(G$)>64 THEN G=ASC(G$)-55
330 Y=Y+W*G;W=W*16:NEXT:RETURN
350 HOME:PRINT
360 PRINT "拨多位开关到读的位置"
370 GOSUB 200
380 PRINT CHR$(4);"BRUN RROM"
390 RETURN
400 HOME:PRINT
410 PRINT "拨多位开关到写的位置"
420 PRINT"加上编程电压":GOSUB 200
430 INPUT"编程首址低位地址:";X$;GOSUB 300;
G=Y
440 INPUT"编程首址高位地址:";X$;GOSUB 300;
H=Y
450 INPUT "输入时间常数字:";F
460 POKE 28,F;POKE 8,G;POKE 9,H;HOME
470 PRINT:PRINT"====="候"====="
480 PRINT CHR$(4);"BRUN WROM"
490 RETURN
500 HOME:PRINT
510 PRINT "拨多位开关到读的位置"
520 GOSUB 200
530 PRINT CHR$(4);"BRUN CPROM"
540 X=PEEK(250)
550 IF X<>0 THEN 590
560 HOME:PRINT
570 PRINT TAB(10)"###核查正确###"
580 GOTO 600

```


掉电工作方式,必须要有一定的硬件环境支持才能实现。许多参考书籍对比介绍甚少,而有的书籍介绍的电路又过于复杂。笔者在开发某一应用系统时,设计了一套简单实用的具有上电自动唤醒功能的掉电接口电路(如附图所示)。该电路由三部份组成,即主电源和备用电源电路,上电检测及复位电路以及掉电检测电路。现将各部份电路的作用分述如下:

一、主电源和备用电源电路

交流电源经整流滤波后,由三端稳压器7805进行稳压,输出稳定的+5V电压,在Vcc1处供系统使用。而80C31则由两处提供电源:一路是在正常工作中Vcc1经隔离二极管,VD₁提供;另一路是在掉电后由备用电池GB经隔离二极管VD₂提供。电路中设置电容器C₁有两个目的,一是上电时Vcc1经二极管VD₁对它充电,待充电电压上升到稳定的4.7V后上电检测电路中LM741输出端才输出高电平。另一目的是系统掉电后,C₁正端电压将保持一定的时间,CPU将在这段时间里执行掉电中断服务程序而进入掉电工作方式。

二、上电检测及复位(自动唤醒)电路

这部份电路由电压比较器LM741及相关元件组成。整流滤波后的电压经2.8kΩ电阻限流,VW稳压后与LM741反相输入端连接,作为比较器的基准电压。而比较器同相输入端则由上分压电阻从主电源电路中的C₁正端引入。现基准电压设为为1.1V,同相端电压

$$V+ = \frac{16 \times 10^3}{(47+16) \times 10^3} \times (5-0.3) = 1.19(V)$$

即当系统上电后,Vcc1经二极管VD₁向C₁进行充电,

待上升到稳定的4.7V时,比较器LM741输出端才输出高平。此高电平经复位电容C₂、电阻R₁的作用而使80C31复位,复位时间

$$t = 10 \times 10^{-6} \times 8.2 \times 10^3 = 0.082(S) = 82(ms)$$

大大超过10ms的要求。此复位操作使已进入掉电工作方式的80C31重新启动,也即将其自动唤醒。

三、掉电检测电路

这部份电路由VD₁、VD₂、VT及一些阻容元件构成。系统正常工作时,交流电正负半周分别经VD₁、VD₂整流,使VT基极总处于高电平而截止,故集电极总为高电平。即80C31的INT0端一直为高电平而不发出中断请求。系统掉电瞬间,三极管VT基极电压下降为0而导通,INT0端接收到一个下降脉冲而启动中断。这样在80C31Vcc电压开始跌落之前,CPU有充裕的时间进入掉电工作方式和执行其它相关的工作。

整流管VD₁及VD₂直接连接到交流输入端可以最快地发出掉电极警信号。

在INT0的中断服务程序里,执行一条ORLPCON,#02H的指令,对特殊功能寄存器PCON.1置位,即可使80C31进入掉电工作方式。

最后还要指出,因80C31P0口的特殊结构,当工作在掉电方式时,由于不执行取指操作,P0口将悬空而导致功耗加大。所以必须在P0口接上提升或下拉电阻(2~50kΩ,一般选用10kΩ),以保证P0口不致处于浮空状态。

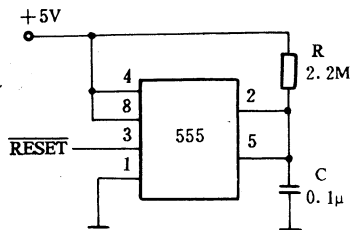
实践证明,以上介绍的电路结构简单,性能稳定,可靠性高。

简易的苹果机故障检修方法

广东汕头第一技工学校(525041) 王惠民

故障现象:开机后没有“哔”声,屏幕出现黑白相间棒状图形,无光标出现。

故障原因分析:此现象表示监控程序没有工作,开机后无法自检,而造成此现象大部分是ROM6或第一排RAM有问题;另外无光标,可能是时基电路(如图)有问题。



检修方法:用一台同型号正常的主机作测试机,简

称W机;故障机简称H机。步骤如下:

1. 将H机上的ROM6拔出插入W机相应位置测试,若正常说明ROM6无问题,将其插回H机,否则更换ROM6。

2. 将H机第一排RAM拔出,插入W机相应的位置上测试,不正常,说明RAM有问题,利用“对半测试法”将故障的RAM片排除、更换。

3. 将正常的RAM片插回H机,开机后有“哔”声,但仍无光标出现。

4. 检查时基电路,用万用表的DCV档测3脚的输出电平。正常工作是:当电源合上,3脚输出为高电平,电源+5V对电容C充电,当C充电到3.3V左右时,3脚输出为低电平,送出触发信号。经检查3脚输出电平一直为+5V(高电平),说明时基电路故障,更换555时基集成电路,故障即排除。若更换555集成电路后,故障仍存在,则是电容C损坏,更换0.1μF电容即可。

AR—3240打印机“联机”灯闪烁故障 检修一例

石家庄市自动化研究所(050031) 张全悦 张书琪

1. 故障现象:

AR—3240打印机加电后,字车能够复位,但“联机”灯闪烁,打印机不能工作。

2. 电路说明:

为了保护打印头,AR—3240打印机设置了打印头温度检测、公共驱动三极管检测、三极管阵列检测三种检测电路,当发生打印头温度过高、公共驱动三极管或三极管阵列烧毁时,打印头停止工作,并且联机灯闪烁不停,以告之使用者。检测电路见附图,下面对该电路作简要说明:

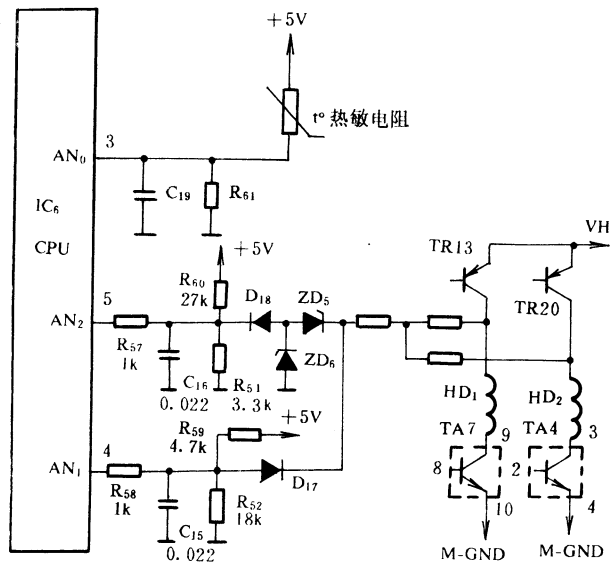
当安装在打印头中的热敏电阻因打印头温度升高而阻值减小时,AN₀点电位升高,当温度升至120℃以上时,打印头停止工作,并且联机灯开始闪烁;

当公共驱动三极管 TR₁₃-TR₂₀中任一管子烧毁后(短路),打印头驱动电源 V_H(30~45V)将 ZD₅、ZD₆击穿,使得 AN₂的电位由正常时的0.5V升高,当 AN₂高于1.5V时,打印头停止工作,联机灯闪烁。(注:电子工业出版社93年2月出版的《常用针式打印机故障检修120例》一书第112页将“高于1.5V”误写为“低于1.5V”)

当三极管阵列 TA₂-TA₇中任一三极管烧毁时,D₁₇负极电位被拉低,随之 AN₁电位也降低,当 AN₁电位低于3.4V时(正常值为3.8V),打印头停止工作,联机灯闪烁。

3. 故障原因分析与检修:

联机灯闪烁,故障必定在上述三电路中,首先将打印头电缆拔掉,故障仍然存在,说明故障不在温度检测



电路。测 AN₂电位为0.5V,正常,故障也不在这里。用万用表测 AN₁电位,表针在2.8~3V间摆动,既不稳定,又低于正常值,说明故障在三极管阵列及其检测电路,为找出故障点,将 D₁₇焊下一条腿,AN₁电位不变,说明故障在检测电路,而不是三极管阵列电路。焊下 C₁₅,AN₂电位恢复正常。测 C₁₅,其严重漏电,漏电电阻约5kΩ,证明 AN₁电位是被漏电的 C₁₅拉低的,由于漏电电阻阻值不稳定,故 AN₁电位也是不稳定的。换上好的 C₁₅,故障排除。

Super AT286运行金山 CCDOS 时帧频不同步的解决办法

广东湛江赤坎北桥路一号之四(524039) 刘良团

我买了一台 Super——AT286 电脑配单色显示器 (Super)。这台电脑在西文 DOS 操作系统下能正常显示,但运行香港金山 CCDOS 时,屏幕就出现场(垂直)不同步现象,能够出现汉字,但同时出现几个画面不停跳动的不同步现象。

开始认为显示卡不行,换了新显示卡,但故障依旧,这就可能是显示卡和显示器不匹配的故障了。拆开显示器后,发现显示器里没有调节场同步的电位器,显示器的场同步电路是由一块集成电路 TDA1170N 和外围电路组成,如图1所示,集成电路的引出脚6和9的外接电路是控制场振荡的频率的,只要调节

TDA1170N 的第6和第9脚外接电路元件的参数就可以改变场振荡的频率,实现场同步,具体改接过程如下:

首先把 TDA1170N 的第6脚外接电阻 R304 的一端焊开,然后再在 DA1170N 第6脚和电阻 R304 焊开端焊接上一个27k 的微调电阻 W1,如图2所示。其它元件不作改动,开机后运行香港金山 CCDOS,等汉字画面出来后,适当调节 W1 使画面不再跳动就可以了,改接后西文 DOS 显示不受影响。

希望我这个修理经验能给别人带来帮助!

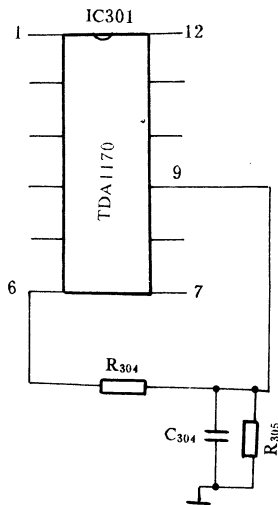


图1

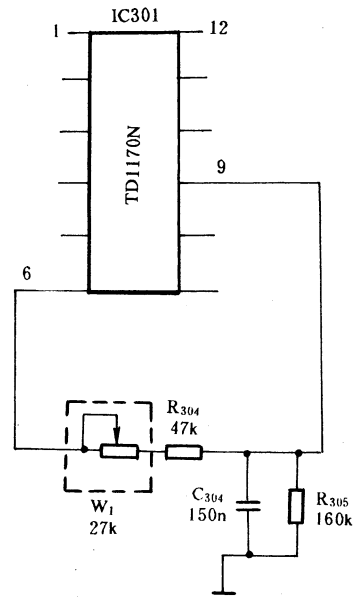


图2

全国电子报刊联合征订简明目录表

订阅代号	报刊名称	刊期	开本页码	单价	全年订价	通讯地址	邮码
1-145	电子商报	周二	对开4版	0.20	20.80	北京石景山路23号	100043
1-28	计算机世界报	周报	4开160版	0.60	30.00	北京市750信箱	100039
1-48	北京电子报	周报	4开8版	0.20	10.44	北京朝阳区东三环北路36号	100026
1-49	国际电子报	周报	4开32版	0.25	12.00	北京市750信箱	100039
2-354	电视技术	月刊	16开100页	2.60	31.20	北京743信箱	100015
2-355	电声技术	月刊	16开48页	1.60	19.20	北京743信箱	100015
2-675	电信技术	月刊	16开60页	2.00	24.00	北京东长安街27号	100740
2-75	无线电	月刊	16开64页	2.40	28.80	北京东长安街27号	100740
2-888	电子与电脑	月刊	16开56页	2.40	28.80	北京万寿路173信箱	100036
2-889	电子技术应用	月刊	16开80页	2.00	24.00	北京927信箱	100083
2-892	电子世界	月刊	16开40页	1.60	19.20	北京市165信箱(万寿路西街11号)	100036
4-141	电子技术	月刊	16开48页	2.20	26.40	上海085-253信箱	200009
4-236	中学科技	月刊	16开48页	1.90	22.80	上海冠生园路393号	200233
4-316	现代通信	月刊	16开32页	1.50	18.00	上海085-253信箱	200009
4-386	无线电与电视	双月刊	16开56页	2.50	15.00	上海市瑞金二路450号	200020
4-489	实用无线电	双月刊	16开56页	2.30	13.80	上海市冠生园路393号	200233
46-115	电脑	月刊	大16开96页	3.00	36.00	广州石牌华南师范大学内	510631
61-74	软件报	周报	4开4版	0.24	12.48	成都市金河街75号	610015
61-75	电子报	周报	4开8版	0.30	15.60	成都市金河街75号	610015
61-8	西部电子信息报	周报	4开4版	0.18	9.36	成都市桂王桥西街66号	610017
61-87	电子文摘报	周报	四开四版	0.18	9.36	成都118信箱	610015
62-175	实用电子文摘	双月刊	16开128页	4.00	24.00	成都市金河街75号	610015
62-189	家庭电子	月刊	16开32页	1.40	16.80	成都118信箱	610015
66-32	机械与电子	双月刊	16开48页	2.00	12.00	贵州贵阳市延安西路67号	550003
77-19	电脑报	周报	4开8版	0.28	14.64	重庆市双钢路三号	630013
81-10	中国电子报	周四	对开四版	0.25	51.50	北京4357信箱(北京石景山路23号)	100043
82-141	国外电子测量技术	季刊	16开48页	3.00	12.00	北京海淀区学院路5号	100083
82-339	计算机世界月刊	月刊	16开120页	2.00	24.00	北京750信箱计算机世界月刊部	100039
82-340	家电维修	月刊	16开32页	1.50	18.00	北京东四十三条32号	100007
82-417	微型机与应用	月刊	16开52页	1.80	21.60	北京927信箱	100083
82-454	信息与电脑	双月刊	大16开64页	2.50	15.00	北京西直门南大街16号	100035
82-512	电脑爱好者	月刊	16开56页	1.80	21.60	北京海淀区中关村南二街五号102室	100080
82-518	今日电子	月刊	大16开96页	6.80	81.60	北京海淀区车道沟一号滨河大厦九层	100081
82-541	电子制作	双月刊	16开48页	1.80	10.80	北京东四十三条32号	100007
自办发行	电子天府	双月刊	16开128页	5.00	30.00	成都桂王桥西街66号	610017
自办发行	电讯技术	双月刊	16开	4.00	24.00	成都市外西茶店子东街48号	610036
自办发行	电子质量	月刊	大16开48页	2.68	32.16	广州1501信箱9分箱	510610
自办发行	广东电子	月刊	大16开48页	2.50	30.00	广州1501信箱9分箱	510610
自办发行	电子产品可靠性与环境试验	双月刊	16开72页	3.00	18.00	广州1501信箱9分箱	510610
自办发行	音响世界	月刊	大16开132页	8.50	102.00	广州市石牌五山路科技街108号	510630

出版软件简介

中国软件行业协会软件出版分会拥有出版社、计算机集团公司和软件研发单位等65个会员单位。本刊将陆续介绍他们出版和开发的计算机软件,供广大读者和用户选购。

EGA/VGA/TVGA 高级编程子程序库 电子工业版

3.5英寸软盘2片/5.25英寸1.2M软盘2片塑夹装
定价:300

本软件既包括文本模式下的各种基本和特殊操作,如屏幕滚动与分割、三套用户自定义字符集的使用等子程序,还包括16色和256色图形模式下常用算法的实现,如点、线、圆、填充、造型填充、平滑滚屏、显示块

的存储和恢复、显示字符以及图形图象压缩与解码,设置非标图形模式等子程序,这些子程序库可供汇编语言、C语言和 Turbo Pascal 调用,也可以直接作为子程序链入用户程序。

本软件与《EGA/VGA/TVGA 开发指南》一书配套发行。

40点阵汉字库到矢量汉字库转换 电子工业版

3.5英寸软盘2片/5.25英寸1.2M软盘2片 塑夹
定价:400元

本软件能把普通汉字系统中供打印用的40点阵汉字库(例如2.13中的HZK40H)转换成高精度的矢量字库。生成的矢量字形漂亮丰满,可以无级缩放。矢量字

库可被任意操作环境下的高级语言(如C或BASIC等)调用,支持在绘图机或激光打印机上输出矢量汉字。

运行环境:286以上各种微机,MS-DOS 3.30以上版本。

反病毒工具箱软件 电子工业版

5.25英寸360K软盘1片 定价:200元

反病毒工具箱软件(V1.01版),是针对寄生于硬盘主引导扇区病毒和硬盘分区表信息易丢失而研制的一种高效、广谱反病毒软件,并具有分区表信息自动修复功能。它可以清除目前已出现的以及今后出现的一切寄生于硬盘主引导扇区的病毒,即使硬盘分区表信

息丢失,该软件也照样能自行恢复,并保证硬盘有效信息不受破坏,使用这些反病毒工具不仅使一般微机操作人员能够动手进行系统维护,同时也为软件开发人员提供了方便的系统维护工具。

运行环境:DOS系统为2.2~5.1版本,IBM PC/XT,AT及兼容机。

通用图文数据库管理系统(软件) 北大版

5英寸高密盘2片 定价:1550元

本软件集图文处理(文字和图像的编辑、显示和输出)和数据库管理(数据浏览、数据添加、数据查询、数据修改)于一体,可广泛应用于公安、交通、海关、军队、旅游、工商、图书情报及其他各类企事业单位的档案、

资料等的管理。

通过使用本软件,可使数据信息(或表格)与图像(照片)同时显示在屏幕上,显示时可选用黑白或彩色方式,并可按照需要放大或缩小图案。也可通过国内常见的24针或激光打印机将图像和文字资料打印出来。

通用字幕设计系统软件 GWDS

5英寸盘 4片 塑夹 估价:390元

HWDS 系统是 ACSH(艺术汉字屏幕处理系统)的高级版本,适用于各类显示器,这通过从汉字库中选取汉字,在显示屏上进行美术设计,实现各种各样的字幕图形制作,用户可以从不同规格的字库(16×16、24×24、48×36点阵)及不同字体的字库(黑体、宋体、楷

体及仿宋体)中选取字模,将需处理的汉字设计为大型普通字、多向斜体字、多向立体字、标语字、空心字及多向雕塑字美术汉字图形。最终产生的图形可按全屏幕或单字方式存贮并可进行多种动画处理。可用于设计软件的彩色封面、彩色字幕、广告汉字等。操作简单,画面美观。

钢筋混凝土辅助教学系列软件 清华版

截交线 CAI 课件定价:350元

动态演示截交线作图过程,可供《工程制图》课程教学使用。

机械设计试题库定价:900元

包含1000余道试题,提供多种选题方式及多种试卷打印格式。

清华 AFB——数据库自动编程系统

定价:1800元

该系统为 FOXBASE + FOXPro 的自动编程系

统,支持多用户环境。

全自动 Fox/clipper/dBASE 源程序生成器 ADG3.0定价:3000元

该软件为全国0500系列优秀软件,可自动生成菜单、查询和报表等各种管理程序。

企业档案微机管理系统 EAMS1.0

定价:2500元

完全按国家标准编制,是企业升级的必备工具,档案管理的有力助手。

绘图型数据库管理系统 GMFP2.0 上海交大版

GMFP 是为 MFOXPLUS 扩展绘图功能的软件。它能绘制各种统计图表和几何图形;不受屏幕字符、列限制,以像素点为定位单位显示多种字体的中西文字符;用键盘或鼠标器直观绘图;利用各种图像扫描仪输入图象;上述方式可相互组合叠加,生成的画面可存

入文件或备注型字段,可随时显示打印,并可将数据直接打印成条形码。使用286、386及其兼容机,配 VGA 卡。适用于统计分析、题库、工艺管理、人事管理、生产过程监视等需要数据与图形功能相结合的广大社会用户。

儿童营养发育电脑咨询软件(7—12岁) 成都科大版

5英寸软盘1片 附说明书 定价:600元

输入儿童身高、体重、胸围、坐高、肩宽等测量指标,计算机打印出的咨询单包含以下内容:该儿童各单项指标的评价结果;综合评价其体型发育、营养发育状况并给出针对性的咨询建议;(根据遗传因素及儿童发育现状预测儿童成人时身高),若有血常规和微量元素检查结果,则可根据结果提出针对性的咨询建议。该系统使用的资料权威、内容丰富、全面、科学、准确、针

对性强、重复率低。系统分个体咨询和集体咨询,并设有数据输入、检查、修改、单项咨询打印等功能,操作极其简单,适用性强,有十分明显的社会效益和经济效益。

运用环境:IBM PC/XT、286、386及其兼容机,CCDOS-2.0以上版本

用户对象:小学及各级保健站、防疫站、儿童保健机构、各级医院儿科

PC 机与8031串行通信的波特率选择

北京电信学校(100021) 江 琪

任何的通信设备进行串行通信时,都有一个相互通道接口的问题。这其中相互之间采用一个统一的波特率是必须的。目前的微型计算机,不管是系统机、单板机,还是单片机都具有可供选择的异步串行通信波特率,而这一般是通过某一特定的寄存器进行设置来完成的。

就 IBM PC 机和8031单片机而言,PC 机是通过异步通信适配器上 INS8250中的除数锁存寄存器的设置而完成的,8031机是通过对其的定时/计数器1中 TH1(方式1、3)的写入而实现的。通常由于不同的系统采用不同的晶体,而造成所设置的波特率与实际波特率之间有误差。当两个系统所产生的误差都在一定的范围内时,通信是可以完成的。误差越大,通信的稳定性也越差。一般由于在设计系统硬件时,晶体都已经选定了,故在设计两系统通信时,对于波特率的选择就要有一个计算。显然较高的通信波特率是大家都希望得到的。

在8250中有一个可编程的波特率发生器,其输入频率是1.8432MHz,它的输出波特率可用下式计算:

$$\text{波特率} = \frac{\text{输入频率}}{\text{除数} \times 16} \quad (1)$$

这里的除数即为装入除数寄存器的16位二进制数。例如,我们将0900H这个数装入除数寄存器,则

$$\text{波特率} = \frac{1.8432\text{M}}{900\text{H} \times 16} = 50$$

又例如,我们将000CH这个数装入除数寄存器,则

$$\text{波特率} = \frac{1.8432\text{M}}{0\text{CH} \times 16} = 9600$$

9600bps 是 INS8250允许的最大波特率,这是由它的硬件所决定的。

对于8031来说,波特率的计算公式如下:

$$\text{波特率} = \frac{f_{osc}}{32 \times 12 \times (256 - \text{TH1})} \times 2^{\text{SMOD}} \quad (2)$$

这里,SMOD 是8031中特殊功能寄存器 PCON 中的最高位,为串行通信波特率的倍增位,可选择为0或1。TH1为8031定时器1的自动重装值。 f_{osc} 是8031采用的外接晶振值,由于 TH1仅为8位,由它决定的值常常在上式中得不到整除,故误差较大。那么如何计算出 PC 机与8031单片机串行通信时的波特率最佳值呢?我们给出下面两种算法。

1. 由8250的除数值推出8031的波特率

由上面可知,通过给出不同的除数,就可以得到不同的波特率。一般来说,这样得到的大多数波特率并不是整数。把这个波特率取整后由(2)式可以得8031的

TH1的值(也不一定是整数),取整后再次由(2)式得到8031的波特率。如果这时8250与8031的波特率差值很小,我们即可选来作为最佳值之一,否则丢弃。下面这个 BASIC 程序给出了当 SMOD=1,8250与8031的波特率差的绝对值 ≤ 2 ,8031的晶体是6MHz时,这种算法1000波特以上的几个最佳值。

```
5  lprint*
10  for i=65535 to 12 step -1
20  b8250=1843200/16/i;b=int(b8250+.5)
30  a=int(256-2*6000000/b/12/32+.5)
40  b8031=6000000*2/12/32/(256-a)
50  if b8031<1000 or abs(b8031-b8250)>2 or b8031=c
    then 100
60  lprint tab(1);"b8031=";int(100*b8031)/100;
70  lprint tab(16);"TH1=";a;TAB(30);"b8250=";
80  lprint int(100*b8250)/100;tab(45);"CHSH=";I
90  c=b8031
100 next i
110 end
```

RUN

```
b8031=1077.58 TH1=227  b8250=1076.63 CHSH=107
b8031=1201.92 TH1=230  b8250=1200  CHSH=96
b8031=1420.45 TH1=234  b8250=1422.22 CHSH=81
b8031=1644.73 TH1=237  b8250=1645.71 CHSH=70
b8031=1953.12 TH1=240  b8250=1952.54 CHSH=59
```

2. 由波特率倒推出8031的 TH1和8250的除数值

这个过程正好与上面的过程相反,即先给出波特率,然后算出8250的除数和8031的 TH1值,取整后再分别利用(1)和(2)式算出8250和8031的波特率,如它们差的绝对值足够小,则可选来作为最佳值之一,否则丢弃。下面给出这种算法的 BASIC 程序(条件同上)及运行结果:

由以上两种算法可以得到相同的最佳值,适当放宽条件,还可以得到更多的值。如果我们将两差值 < 2 改为差值/b8031则可有另外一种算法,也会得到更多更高的值。

```
10  for i=100 to 9600
20  a=1843200/16/i;b=256-6000000*2/i/12/32
30  b8031=6000000*2/12/32/(256-int(b+.5));b8250
    =1843200/16/int(a+.5)
40  if abs(b8031-b8250)>2 then 100
50  if b8031=c then 100
60  lprint tab(1);"b8031=";int(100*b8031)/100;
70  lprint tab(16);"TH1=";int(B+.5);tab(30);"b8250
```

传真通信的通路

张景生 张建军 辛亚西

在传真通信系统中,传输信道的传输质量对传真通信的影响较大。由于我国通信比较落后,还没有建立专门的传真通信网。现在,我国的传真通信大多数在公用电话交换网上进行,也有的租用电话线进行传真通信。GB3382—82上规定,三类传真机可在公用电话交换网和租用线上使用。下面分别讨论这几种传真通信网。

一、利用公用电话交换网的传真业务通信

目前,我国的长途电信网采用省间中心(大行政区)、省中心(省会所在地)、地区中心(行署所在地)和县中心四级汇接制,其接续方式依据交换的方式不同有人工、半自动和全自动接续方式。人工接续方式是用户首先拨通长途记录台登记,然后由接续台话务员根据用户的申请接通被叫用户。半自动方式是用户进行长途呼叫时拨通特定的电话号码(我国目前规定为“173”)到长途台半自动挂号台,主叫用户在摘机状态下等待,话务员根据主叫用户的申请,拨通被叫用户的

电话号码后双方即可通话。全自动方式是用户直接拨被叫用户所在城市的电话号码即可接通通话的方式。

由于当前各国的电话网是最大的公用通信网,而在实现长途自动化的国家,用户可以直接拨号呼叫全国任意一个地方。这样,利用目前的电话网进行传真业务通信,有如下的好处:

- (1)电话网是覆盖范围最广的电信网络,利用它可以简单地组织国内和国际的传真业务。
- (2)用户的传真机可以与电话机共享同一对用户线和交换机门号,不需为投资而增加费用负担。
- (3)传真与话音业务可做到网络资源共享,有利于提高网络资源的利用率。
- (4)对传真业务按电话业务收费,通信费用低廉。

因此,世界各国绝大多数的传真机是在公用电话网中使用的。

图1是利用公用电话交换网电路作为传真通路的通信路由方框图。

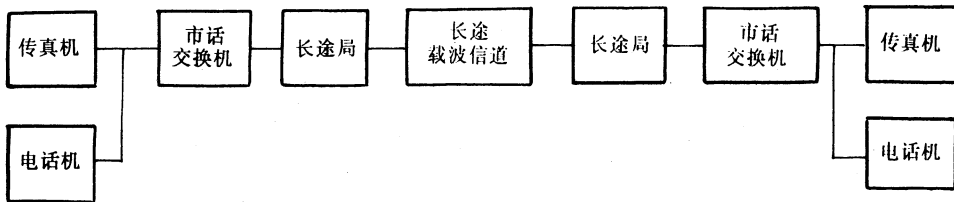


图1 利用长途电话交换网电路作传真通信通路示意图

由上图可以看出,在公用电话交换网中进行传真通信,交换设备及通路对传真质量要产生一定的影响。主要是来自线路及交换设备上的传输衰耗、电路噪音、群时延失真、相位抖动以及电路瞬断等因素。

在公用电话交换网上进行传真通信,不仅可以完成点对点的通信,还可依靠传真机本身或附加设备,实现多址通信和特种业务。

(1)顺序式同文通信

这种方式利用传真机本身的存储器 and 拨号功能,用人工或自动呼叫的方法,通过公用电话网,将同一份

文件按照所设定的拨号顺序依次发出。例如:松下公司生产的 UF—2EXC(H)型传真机及 UF—200型传真机、冲电气公司的 OF—10系列传真机等,均可实现。

(2)轮询方式

在一般的传真通信中,一般由发端作为主叫端,呼叫对方。在轮询通信中,是由收端作为主叫。轮询可以是单家,也可以是多家;可以用人工拨号的方式轮询,也可以由传真机本身的自动拨号器按照存入的电话号码的顺序自动拨号轮询,也可由操作员设定时间定时轮询。为防止文件误传,一般情况下,传真机处

```
="";  
80 lprint int(b8250 * 100)/100;tab(45);"CHSH=";int(a  
+1.5)  
90 c=b8031  
100 next i  
110 end
```

```
b8031=1077.58 TH1=227 b8250=1076.63 CHSH=107  
b8031=1201.92 TH1=230 b8250=1200 CHSH=96  
b8031=1420.45 TH1=234 b8250=1422.22 CHSH=81  
b8031=1644.73 TH1=237 b8250=1645.71 CHSH=70  
b8031=1953.12 TH1=240 b8250=1952.54 CHSH=59
```

于轮询工作方式时,都要设有相应的轮询密码。

(3)一发多收同文通信

这种工作方式必须附加同文通信装置。我们知道,在三类机点对点通信中,传真机首先要进行传输控制规程的交换,以确定传真机的工作种类、扫描线密度、传输速率等,如果是多家同文通信,不附加同文装置,则传输控制规程无法交换,也就不能确定传真机的工作参数。为解决这一问题,可在主叫传真机与多家被叫用户间加入相应的设备,这种设备可将各个被叫用户的传真机工作方式加以综合,确定出最佳工作方式,向主叫传真机报告,开始传输报文。这种工作方式适用于中心站。

(4)存储转发式中继通信

这种工作方式是利用传真机的存储功能,将某一

用户的报文,存储在传真机的存储器中,然后通过人工或自动拨号方式转发给另一用户。这种方式可用于传真通信的转报。

用公用电话交换网实现传真通信有易于实现的优点,但也存在一些问题:如网络无差错控制能力,线路质量差时通信质量无保证;传真信息需先经调制转换为电话网可以传送的模拟信号,其最高通信速率为9600比特/秒,影响传递时间的缩短;公用电话交换网不能提供传真同文通信。因此,日本、美国、香港等发达国家和地区都建立了传真通信网。

二、利用租用电路作为传真通路

如果传真业务量很大的话,也可租用电话电路作为传真通信电路。图2为租用电话电路(专线)作为传真通信的示意图。

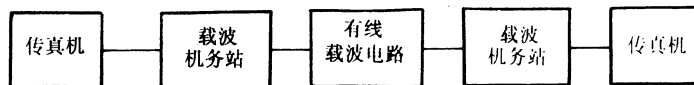


图2 租用电路形式的传真通信示意图

在租用电路(专线)上工作时,免除了交换机的干扰,但是,三类传真机的某些功能就不能使用,如自动拨号顺序发报功能,定时发报功能,定时轮询功能等。应注意的是,三类传真机在租用电路(专线)上工作时,由于线路上不提供直流电压,传真机内摘机检测电路不起作用,所以需对传真机内部开关或参数进行设置,某些传真机需改接内部连线。

三、传真通信网

前面所说的传真通信是建立在公用电话交换网或租用电路上的,这种通信方式有一定的局限性,如同文通信、不同类别的机器互通、传真信箱等不能开展,所以在日本、美国、香港等国家和地区相继建立了传真通信网。

传真通信网提供下列服务:

(1)不同机类间互通

目前,正在使用的传真机中有一类机、二类机、三类机,它们之间的互通可由传真通信网解决。如用户使用的是一类传真机,需要向使用二、三类机的用户发报,一般情况是不可能的,但传真通信网将一类传真机信号接收后,经过压缩编码后发向相关用户,从而解决不同机类间的通报。

(2)同文通信

(3)代收业务

(4)数据库信息检索

(5)传真联机故障诊断。

四、通路质量对传真通信的影响

电话属于听觉通信,最后接收信息是靠人的耳朵,人耳对通信质量的要求是响度和清晰度,而影响响度和清晰度的主要传输损伤因素是传输衰减、电路噪声、衰减频率失真等。传真通信属视觉通信,最后接收信息是用人的眼睛,人眼对传真副本的质量要求是文字可

读度和文字清晰度。电话电路中影响这两个指标的因索是很多的,许多对话音传输影响不大的因素,对于传真通信来说是不能容忍的。表1列出了电话电路中对传真传输损伤的各种因素。这些损伤因素来自传真通路中各个组成部分,以及通路外的某些干扰源。每一种损伤因素对三类机传输信号的影响都是误码,表现在三类机的工作状态上是联系不能建立、传输中断、复制图象中出现字迹拉长等现象。这些表现根据损伤因素出现在传真通信过程的不同阶段而定。

传输损伤因素分为稳态参数和瞬态参数两类。所谓稳态参数是指该类损伤因素固定存在,其值不随时间变化或变化不大,如传输衰耗、频偏、群时延失真等。瞬态参数是该类损伤因素是突发性的,其值随时间变化很大,通常将脉冲噪声、相位突变、电平突变、瞬断列为瞬态参数。在定量测量时,稳态参数基本上只测量一次即可,但瞬态参数要经过多次测量,再从大量测试数据求得统计特性。

表1 传输损伤因素的定义及主要来源

传输损伤因素	定义	主要来源
传输衰耗	信号在传输过程中,各种原因引起的电平降低。通常用dB或N表示	市话线路、传输和交换设备
电路噪声	传输通路或设备中除有用信号外的任何电的干扰。	电路中的热噪声、串音,电磁耦合等
群时延失真	在一定的频率范围内,群时延随频率而变化所产生的失真	通路中带滤波器的相位特性

衰减失真	在一定的频率范围内, 衰减随频率变化而不同所引起的失真	通路滤波器的衰减频率特性和无加感市话电缆的衰减频率特性
非线性失真	当输入信号为正弦波时, 输出信号的波形与输入波形不一致, 产生失真	电路中的非线性
受话回声	接收端所收到的回波信号	线路阻抗失配所引起。
相位抖动	信号过零点交叉点的晃动	20~300Hz 频率成分在电路中产生的寄生相位调制
频率偏差	发送信号载频和接收信号载频之间的差值	由于元件老化、电源变化所引起的载频系统内载频变化
单频干扰	单一频率所产生的干扰	电路中的串音、交调、载频泄漏
脉冲噪声	持续时间小于 0.5ms 的电平突变	步进制交换机件接点接触不良, 拨号脉冲以及强电所引起的干扰
相位突变	信号相位的瞬时变化, 短时间后相位回到原来的值或是变成新值	载波电路中载供倒换、无线多路传播
电平突变	短时间内电平变化 ± 3dB	长途电路净衰耗的快变化
瞬断	短时间内电平下降 6dB 以上, 持续时在 0.5ms~1min 以内	系统转换(电路故障引起)、接触不良、无线衰落

五、三类机对传输信道的要求

利用电话电路进行传真通信时, 为了保证传真信号的正常传输, 我国对话路开通传真通信作了相应的规定。传输标准是从用户对通信质量满意度(用户对通信质量的满意程度)出发, 规定传真传输电路全程的各项传输损伤因素的总容限(传真容忍范围的限值), 保证三类传真机在电话网或租用电路中的传输质量, 是用户和通信主管部门的共同要求。传输标准的作用是作为三类机传真通信系统设计、使用和制订电路维护规程的依据。

1. 三类机传输质量

三类机的传输质量从两方面评价, 一是报文传输建立率, 二是传真副本文字质量。

(1) 报文传输建立率

报文传输建立率即三类机按照规定的传输速率完成报文传输的次数与指定速率训练总次数之比。用公式表示为:

$$\text{报文传输建立率} = \frac{\text{指定速率训练成功的次数}}{\text{指定速率训练的总次数}}$$

传输标准中规定报文传输建立率在 80% 以上, 其含义是指当电路传输损伤因素容限满足 4800 比特/秒(或 9600 比特/秒)传输速率指标时, 报文传输成功的次数应达到 80%, 且其报文质量满足使用要求。

(2) 传输副本文字质量

传真质量在传输副本文字质量中得到充分反映, 对副本文字质量的评价方法可采用客观评定法和主观评定法两种。在实用中, 因客观评定法受到条件的限制, 只用于制造生产厂家或研究单位, 所以多采用主观评定法。

主观评定法是用人们的视觉印象对传真副本进行直接的判读, 按质量很好、好、一般、差、很差五个等级进行主观评分。在评定中, 应采用国际 GB3382-81(传真测试样张)作为发送原稿。《传真测试样张》大致可分为: 文字可读度部分由 270 个 4 号仿宋体汉字、32 个 5 号仿宋体汉字以及汉字手写体和少数民族文字组成; 分辨力测定部分由不同的线条及楔形线组等图形组成。

测试时, 先辨读传真副本上的文字, 将可读出的文字数与样张文字总数相除, 即可得出可读度, 用公式表示为

$$\text{可读度} = \frac{\text{能辨认的字数}}{\text{总字数}}$$

清晰度则是笔划完整无缺的字数与总字数之比, 即

$$\text{清晰度} = \frac{\text{笔划完整无缺的字数}}{\text{总字数}}$$

由于文字笔划的组成有一定的关系, 一个字的笔划的某些部分, 甚至部分笔划遭受损失还可能判定, 所以可读度比清晰度高。

在三类机传输标准中规定: 当全程传输损伤因素在容限范围内时, 副本文字的质量应达到: 当扫描线密度为 7.7 线/毫米时采用 MH 编码方式的副本质量达到 3.5 分以上, 可读度达到 99% 以上; 采用 MR 编码方式时的副本达 3 分以上, 文字可读度达 99%。当扫描线密度为 3.85 线/毫米时, 采用 MH 编码方式的副本文字质量平均评分达 2.2 分以上, 文字可读度达 99%。在这里, 质量很好为 5 分, 好为 4 分, 一般为 3 分, 差为 2 分, 很差为 1 分。

至于分辨力的判定, 要用专门的仪器来判读, 在实用中, 也可用眼睛去分辨。能分辨出的最小细节的倒数, 即为分辨力。由于每人的分辨能力不同, 能判出的细节也就不一。所以分辨力判定只能作为参考。

2. 传真的传输电路

在电话网或租用电路上进行传真通信, 其路由是多种的, 有线或无线, 长途或短途。不论哪种形式, 均可用图 3 表示。

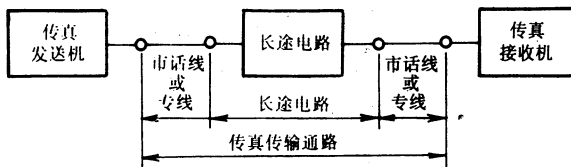


图 3 传真传输电的连接

传真全程传输是从传真发送机的输出点起,到传真接收机的输入点止。为了保证传输质量的稳定,必须对全程损伤因素进行合理的规定。

在三类传真机的传输标准中,对损伤因素的容限作了相应的规定。一般情况下,实际电路指标优于造成质量下降的水平。三类机对传输通道的要求见表2。

表2 三类机传真通信全程传输损伤因素容限

传输损伤因素	容 限			
	4800比特/秒		9600比特/秒	
全程衰减	载频1800Hz的全程衰减不大于32dB 注:频率为800Hz的全程衰减不大于24dB		载频1700Hz的全程衰减不大于22dB 注:频率为800Hz的全程衰减不大于18dB	
群时延失真	频率范围(Hz)	群时延失真(ms)	频率范围(Hz)	群时延失真(ms)
	500~600	6.5	500~600	5.0
	600~800	5.0	600~800	3.0
	800~1000	3.0	800~1000	2.0
	1000~2800	1.5	1000~2800	1.5
	2800~3000	3.0	2800~3000	2.0
衰减失真	频率范围(Hz)	允许相对于800Hz衰减的最大偏离值(dB)	频率范围(Hz)	允许相对于800Hz衰减的最大偏离值(dB)
	500~1300	-1~+3.0	500~1300	-1~+3.0
	1300~1600	-1~+4.5	1300~1600	-1~+4.0
	1600~2000	-1~+8.0	1600~2000	-1~+5.5
	2000~2200	-1~+10.5	2000~2400	-1~+7.0
	2200~2500	-1~+12	2400~3000	-1~+8.0
2500~3000	-1~+13.5			
信号噪声比	S/N(非加权)>24dB		S/N(非加权)>27dB	
受话回声	信号回声比 S/E>27dB		信号回声比 S/E>29dB	
谐波失真	二、三次谐波衰减均>25dB		二、三次谐波衰减均>42dB	
相位抖动	相位抖动不大于10° P-P			
频率偏差	频率偏差 Δf 不大于5Hz			
单频干扰	信号干扰比>30dB			
脉冲噪声	峰值超过-16dBmo的脉冲噪声的计数15分钟内不超过18次		峰值超过-21dBmo的脉冲噪声的计数15分钟内不超过18次	
相位突变	超过测量门限20°的相位突变在15分钟内发生次数不超过10次			
电平突变	超过测量门限3dB的电平突变在15分钟内发生次数不超过8次			
瞬 断	超过测量门限6dB、持续时间不大于3ms的瞬断在15分钟内发生次数不过8次			

我国专用计算机通信网的组网方式

江苏省邮电管理局(210003) 薛兴华

为加速信息资源的开发利用,推动计算机数据库及信息资源的共享,我国近几年专用计算机通信网的发展明显加快。到目前为止,银行、铁路、民航、航天、电力、农业、统计、商业、海关、公安、新华通讯社等部门已基本建成全行业的专用计算机网络系统,在为本部门提供计算机数据业务、联网数据库资源共享及通信方面发挥了积极作用。

建立专用计算机通信网必须从本部门实际需要出发,既应考虑到投资规模、网络覆盖范围,又应考虑到数据业务的服务对象、网络用途等,正确选择组网方式。目前,我国建成投入使用的专用计算机网主要采用如下五种组网方式。

1. 部门行业自行规划、投资兴建专用计算机通信网

这种组网方式建设费用大、技术要求高、建设周期长,大多是在本行业已有较发达的专用电信网基础上建立的。

全国铁路计算机网是国内建立最早、规模最大、速率最高的全行业数据网络。铁路分局以上的网络称为基干网,是一个分布型网络;铁路分局以下的网络称为基层网,是一个集中型的网络。根据信息源点到铁路分局之间的线路负载小、铁路分局至铁路局和铁路局至铁道部之间的线路负载较大的特点,铁路计算机网最终将是一个分布与集中相结合的网络拓扑结构。

目前的铁路基干网有铁道部至12个铁路局、56个铁路分局共69个网络结点,采用美国DEC公司的DNA网络协议和专线连接方式,通信主速率为4800bit/s。北京、广州两铁路局管辖内已采用9600bit/s的速率。铁道部至柳州、兰州、乌鲁木齐三个边远铁路局已使用卫星信道。

基层网,即铁路分局计算中心至管辖内的主要信息源点共1000个,目前正在建设中,主要信息源点指编组站、区段站、主要货运站、局间分界站及车务段、机务段等。京沪、京广、京哈三大铁路干线已有288个基层站段实现了与铁路分局计算中心联网,其中上海铁路局连通了58个信息源点,并实现信息共享和综合处理。此外,上海铁路局还建成了路局(1个)一分局(6个)一主要车站(80个)三级大型计算机远程联网系统,路局对各分局采用话路专线通信方式,通信速率为4800、2400、1200及300bit/s不等。全系统共有200个网络节点,路局、分局配有VAX系列机等。

2. 利用公用电话网组建专用数据网

这种组网方式投资费用较少,建设时间短,是目前较普遍采用的组网方式。

专用数据网的发展离不开现有的电话通信网资源,即使在美国、日本和西欧等经济发达国家,发展数据通信也都是首先利用已建成的四通八达的电话网资源,然后再建专用和公用数据交换网。我国已制定“在电话网上开放数据业务的技术体制”,并在全国开办电话网的数据通信业务。

电话网上的数据通信系统由中心计算机、数据信道和数据终端设备等组成。主要数据通信业务有信息通信、远程数据处理、远程控制等。在电话网中,数据传输的速率一般为300bit/s至9600bit/s。在一般情况下,电话网使用费与通信时间成比例。由于利用电话网进行数据通信的速度较慢,故常被用于通信量较低的计算机网数据通信。

中国民航专用数据通信网是国内大型实时性计算机通信网,是利用公用电话通信网的50多条长途电话专线组建的。该网的拓扑结构为树型结构,今后将向网状结构发展。

该网以北京主机系统为中心向外辐射,由9台中大型前端处理机和18台远程通信集中器构成网络节点。前端机及节点机采用分布式通信处理机DCP和通用通信处理机UCP。结点机之间采用比特高级链路协议(UDLC),与国际民用航空通信网(SITA)之间采用SITA专用的P1024通信规程(UNISCOPE)。全国各城市的用户终端通过8大通信节点(民航北京售票处、首都国际机场、上海、广州、成都、西安、沈阳和乌鲁木齐)联接到北京。

3. 利用公用分组数据交换网(CNPAC)和租用专线组建本部门的系统网

这种组网方式较容易,能够实现各部门对计算机数据通信的需要,是计算机网发展的方向。

分组数据交换网是一种高速度、高质量的公用数字数据交换网,它的最大通信速率可达到64kb/s,容易实现不同类型、不同速率计算机之间、计算机与终端之间、终端与终端之间的数据传送。分组交换网采用流量控制方法传送数据,可以利用现有线路资源,具有很高的可靠性,是目前世界上多数国家采取的方式。

核科学技术情报联机检索服务系统是一个综合的核科技信息收集、加工、存储和检索的服务系统。系统使用超小型机BULLDPS7000/260(内存32MB,外存6000MB),利用MISTRAL情报检索软件和IDS-I网状数据为管理软件,建立了有关核科技的文献型、事实型和数值型数据库,通过CNPAC与公用电话网提供快捷方便的成果查新、科研立项、科学研究、管理决策、技术经济分析等检索服务。

用户进入公用分组数据交换网是计算机网发展的方向,但由于我国目前 CNPAC 节点机少,网络覆盖面小,网络功能不够全,满足不了用户组建专网的要求,因此目前利用 CNPAC 建网的部门行业还较少。但随着今后公用分组网的扩建,覆盖面的扩大、网络功能可靠性的提高,数据库资源的丰富,更多计算机用户将使用 CNPAC。

4. 利用甚小天线卫星地球站(VSAT)系统组建专网

这种组网方式灵活方便、可靠性强,覆盖区域广,是目前数据通信网技术中较先进的一种,发展前景远大,已引起各部门的重视。如新华社卫星主站报版远程传输系统、中国人民银行金融卫星数据通信专网等。

VSAT 卫星通信网综合了分组信息传输、交换技术、K_a 和 C 频段通信技术、微波技术及大规模集成电路、多址协议、频谱扩展等先进技术,具有较大的灵活性,信道误码率低,网中用户可不受地理位置及地面繁杂通信线路的限制。主站为全网运行的枢纽,起着沟通用户首脑机关(主交换机或用户交换机)与全国各地小站间卫星信道的作用,并对他们实施分配管理和监视。

各专用网的主机(用户交换机)以地面中继线路与主站相连并在主站中互相隔离。当业务需要时,主站可以与 PBX/PABX 及数据交换设备相连。

小站分为户外单元、户内单元两部分。户外单元包括小天线、高频系统及支架;户内单元包括调制解调器、编解码器及接口,装在机箱内。小站使用普通单相市电。

VSAT 系统终端可放置在室内,可以不通过公用网进行语音、传真和数据的传输。中国人民银行建成的覆盖全国的金融卫星数据通信专用网(又称电子联行 EIS),采用 VSAT 系统组网方式,是一个专门用作数据传输的星型网。

该专网的中央主机设在北京沙河,236 个小站设在全国各个城市分行;主站与小站可直接通信,小站与小站不能直接通信,须经主站转发。系统中主站通过载波向各小站发送信息,采用时分复用(TDM)体制,每载波信道传输速率为 57.6kb/s,共用 22 个载波信道。各小站通过载波信道向主站发送信息采用预分配单路单载波体制,各小站载波信道的传输速率从 2.4~9.6kb/s 不等,根据业务量大小而定。

ORACLE 关系数据库

杨 军

ORACLE 关系数据库管理系统是由美国 ORACLE 公司开发的,在国际上拥有广泛的市场,在关系数据库的多次评比中一直名列前茅,它具有以下特点:

兼容性。ORACLE 产品完全实现了 SQL——这一工业标准关系数据库语言,因此,在 ORACLE 上开发的应用软件,也能运行在 IBM 的 DB2 和其它基于 SQL 的 DBMS 产品上。

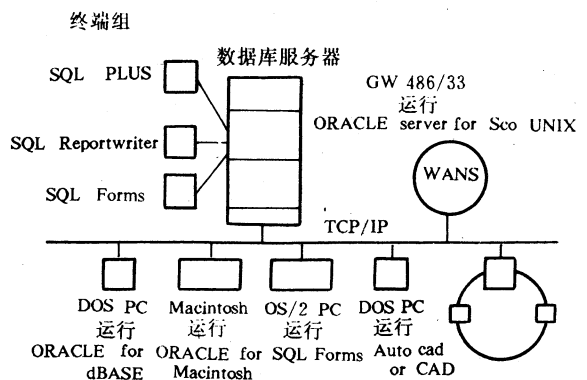
可移植性。ORACLE 可以运行在各种主机、小型机、工作站和 PC 上,并且支持很多种操作系统。

可连接性。ORACLE 的分布式结构使 ORACLE 数据库和应用能够驻留在多台计算机上,并且相互间的通信是透明的,支持多种通信协议,如:TCP/IP、SPX/IPX、DECNET 等,可以把不同的计算机、操作系统和网络连接起来。

多能力性。ORACLE 的优越性体现在其先进的体系结构上,这一先进的体系结构使 ORACLE 具有最大的吞吐量,多用户事务处理,保护数据以免非法存取及系统失败的能力。ORACLE 产品还为各种不同层次的用户提供了一系列工具,这些第四代开发工具帮助你建立强有力的产品应用,并且可以满足最常见的要求——从屏幕格式设计、报表生成到即席查询。

分布式能力。ORACLE 的数据库服务器采用先进的客户/服务器模式,支持分布式处理环境和分布式数

据库,在分布式处理环境中,服务器和客户同存于一个网络中,服务器运行大的、共享存储的和事务的程序,客户运行较小的前端应用。这样既减小网络通信开销,又具有高度并行性,保证数据库的一致性和完整性,ORACLE 服务器的分布式数据库结构,使用户可以动态地访问和更改数据库表,这些数据库表存放在由不同计算机组成的网络中,用户在存取这些表时无需知道它们的物理位置。



模拟乘法器 IC 及使用方法(二)

李兰友

(接上期)

二 AD834

1. 引脚与参数

AD834是可用于0~500MHz高频的模拟乘法器 IC。AD834的引脚排列如图8所示。

引脚说明如表4

表4 AD834引脚说明

引脚序号	符号	功能
1	Y1	} Y 输入
2	Y2	
3	-Vs	负电源
4	W2	} 输出
5	W1	
6	+Vs	正电源
7	X1	} X 输入
8	X2	

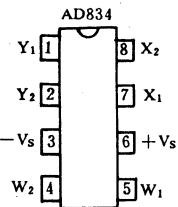


图8 AD834引脚

AD834输入输出特性如表6所示

表6 AD834输入输出特性

	参 数	值
输 入	输入满量程范围	±1V
	限幅电平	±1.3V
	失调电压	0.5mV
	失调漂移	10μV/°C
	输入偏移电流	45μA
输 出	比例电流	4mA±1%
	噪声电压密度	16nV √Hz
	零信号电流	8.5mA
	差动失调电流	±20μA

AD834参数如表5

表5 AD834主要参数

型号	传递函数	总误差 (%)	X 输入非线性 (%)	Y 输入非线性 (%)	小信号带宽 (MHz) (0.1Vrms)	转换速率 (V/μs)	电源电压/电流 (V/mA)
AD834	$\frac{X \cdot Y}{(1V)^2} \times 4mA$	0.5	0.4	0.1	500		±4~±9/ +14 -35

2. 基本工作电路

由 AD834组成的电路如图9所示。AD834是电流输出型 IC, 输出电流 I_{out} 为

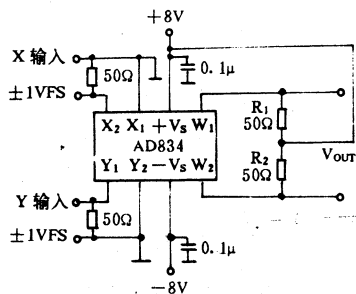


图9 AD834基本电路

$$I_{out} = 4mA \cdot X \cdot Y / (1V)^2$$

当 X、Y 输入各为 1V 时, $I_{out} = 4mA$ 。

将电流转换为电压由电阻 R_1 和 R_2 完成。这时, 有

同相电流 8.5mA 流入 AD834, 取 $R_1 = R_2$ 消除同相电流的影响。

这样, 当 $R_1 = R_2 = 50\Omega$, $X = Y = 1V$ 时, 输入电压 I_{out} 为

$$V_{out} = 4mA (2 \times 50\Omega) = 0.4V$$

实测 AD834 的失真率, 在 100MHz 时 Y 输入侧为 -50dB, X 输入侧为 -44dB, 失真率甚小。同时, AD834 的馈通也很小, 在 100MHz 时达 -50dB 的程度。

3. 使用注意事项

(1) +V 端子电压需低于 W 输出。

为使 AD834 能正常工作, 必须保证 W 输出高于 +V 端子电压。例如, 当使用如图 10 的电路进行实验时, 不管怎样调整, 输出电压总是飘忽不定。其主要原因是 +V 端子直接接正电源, 电压高于 W 输出所致。

一般而言, 要求 W 输出高于 +V 端子电压 1V 左右, 电路可正常工作。降低 +V 端子电压的电路如图 11 所示。

PLD 器件的新军 ——GAL

浙江湖州 (313018) 倪征宇

近十年来,随着微电子技术和计算机技术的发展逻辑器件的新产品层出不穷,而其中之佼佼者则为采用高速电可擦 CMOS(即 E²CMOS)工艺制造的 GAL 器件。目前,世界 IC(集成电路)市场对 ASIC(专用集成电路)的需求日益增加。据有关专家推测,到本世纪末,ASIC 将在 IC 市场占统治地位,而作为 ASIC 的主要部分——GAL 技术,将成为其发展的先导。自 1985 年以来,国外应用和开发 GAL、PAL 技术的势头方兴未艾,许多新产品的的设计都应用了此技术。而国内一些科研单位也竞相研究 GAL 和 PAL 应用技术及开发工具,已有产品运用了此技术进行设计。可以看出,电路设计的器件逐渐由标准数字 IC 向 ASIC 器件过渡。今后,对从事电子电路设计人员来说,不仅要了解电路器件本身结构和进行硬件在线实时调试,而且必须具有计算机软件知识和进行软件编制能力。某学者说:“今天的系统设计工程师不会设计 ASIC(GAL)产品,就好像五、六十年代的系统设计工程师不会设计印刷电路板一样。”

GAL 器件的迅速崛起,是同其优越的性能分不开的。下面就 GAL 的优越性及其完善的开发设备作一简单介绍。

GAL (Generic Array Logic)——通用逻辑阵列,是 1985 年由美国 LATTICE 公司投产的,是在 PAL (Programmable Array Logic)器件的基础上发展起来的崭新的逻辑芯片,其结构直接师承了 PAL 器件的“与”——“或”结构,并有了新的突破,形成了“与”——“输出逻辑宏单元”结构,并和 PAL 器件 100% 的兼容。它和 PAL 同属于 ASIC 中的一个重要分支——PLD (即可编程逻辑器件),是目前最具竞争能力的 ASIC 器件,自投入市场以来,受到了广大用户的宠爱。GAL 作为电路设计的一种理想器件,具有以下功能:

(1) 采用高速电可擦除 CMOS(E²CMOS)工艺,可最大程度地提供可测试功能以及生产工艺质量,另外可随时擦除,最适合于样机的研制。

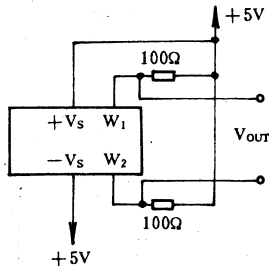


图10 AS834使用中错误的接法

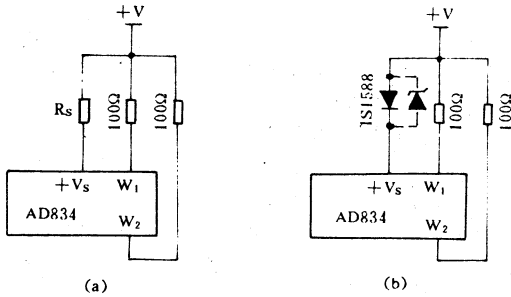


图11 降低+V端子电压的方法

图11(a)为电阻降压法。在正电源和 +V 端子之间接入一电阻 R_s , 则在电阻 R_s 上生成 $R_s \cdot I_{cc}$ (电源电流) 的压降。由于 AD834 的电源电流为 11mA, 当 R_s 选 50Ω 时, 则生成约 0.55V 的压降, 使 +V 端子电压降低。当负载电阻为 100Ω 时, R_s 选择 100Ω 为宜。

图11(b)为使用二极管式稳压管降压的方法。使用二极管时, 其导通压降约为 0.7V 左右。

另外, 还可采用双电源方法, 使 +V 端子电压略低于 W 输出。

(2) 限幅电平

AD834 输入的限幅电平为 ±1.3V, 因而, 输入电压范围最大为 1.3V。当输入电压超过 1.3V 时, 精度将急剧恶化。实测表明, 当输入电压在 1V 以内时, 线性度非常好, 特别是输入电压在 0.7V 及其以下时, 直线性尤高, 误差在 0.1% 以下。

(3) 电源电压的影响

电源电压的变化, 将导致产生增益误差和零点误差, 一般而言, 电源电压变化在 1V 左右时, 这两种误差均在容许范围内, 不会影响正常工作。

(4) 输入偏流的影响

AD834 的输入偏流为 45μA, 因此, 当输入端接入高阻时, 将产生失调电压。例如, 当输入端外接 100Ω 电阻时, 则产生 $100\Omega \times 45\mu A = 4.5mV$ 的失调电压, 与 AD834 的输入失调电压 500μV 相比, 是一个相当大的值。

减小输入偏流影响的电路如图 12 所示。图中取 $R_1' = R_1$, 用以减小偏流的影响。

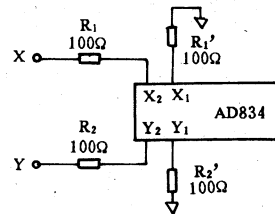


图12 减少偏流影响的接法。

(2)其芯片速度不低于任何其他 TTL 可编程逻辑芯片速度。

(3)具有 CMOS 的低功耗特性。

(4)具有输出逻辑宏单元(OLMC),使用户能按需要对输出组态。

GAL 芯片所具有的以上各项功能,决定了它对系统设计带来的诸多优越性:

(1)可取代 TTL/74HC 组合逻辑电路,取代低密度门阵列和其他可编程逻辑芯片。

(2)功能集成度高,在给定的体积内可集成的逻辑功能的数目多,一般可替代4~12个中小规模集成芯片,有利于降低系统的研制费用,减小了成品体积和功耗。

(3)芯片、电路板数量减少,大大简化了系统设计,提高了系统的可靠性和稳定性。

(4)GAL 器件提供了一个能被编程的保密单元,可用来防止检验和读出芯片中的编程数码,保护了芯片设计专利,防止他人抄袭。

(5)对芯片制造商来说,由单一器件的生产代替了多种器件的生产,大大简化了生产过程,避免了繁重的体力和脑力劳动。

(6)GAL 的另一个优点是减少了原材料的浪费和容易进行对产品质量的跟踪。

GAL 器件的应用受到了先进的开发工具的支持,其开发设备有两大类:

(1)可编程逻辑开发软件包。它由专门研制开发工具的公司提供,运用这些软件,在计算机上可对设计的电路逻辑进行卡诺简化、自动布线和仿真调试。较先进的通用软件包有 Data I/O 公司的 ABEL 软件包和 Assisted Tech 公司的 CUPL 软件包。

(2)编程器。GAL 器件编程方法不需要特殊的电压和定时,故大部分制造厂家的编程器都能支持对 GAL 器件的编程,如用户已有一套标准的 PLD 器件的开发工具,不需要进行复杂、昂贵的改动,即可对 GAL 进行编程。

GAL 芯片就类型来说可分为四个系列:20引脚的 GAL16V8系列、24引脚的 GAL20V8系列、24引脚的 GAL39V8系列和24引脚的 ispGAL16Z8系列。前两个系列为常用的器件。而 ispGAL16Z8系列是目前 GAL 器件中最新、最先进的系列,它不需要特殊的编程硬件,只要“在系统”对器件编程即可。GAL 芯片就其封装外形来说,可分为六种类型,即塑料双列直插式、陶瓷双列直插式、侧面镀铜双列直插式、塑料方形罐式、陶瓷无引线架式和模片式。

通过以上对 GAL 器件的介绍,笔者认为 GAL 器件不愧是一种理想的可编程逻辑器件,它以其高性能、高可靠性、可擦除性及输出逻辑结构可组态的特性,将受到更多用户的喜爱。目前,清华大学微电子研究所已引进了设计新一代 ASIC 产品的 CAD 工作站;复旦大

学计算机系已将 GAL 产品应用于单片机开发系统中;北京四通集团公司研制和开发了四代 GAL 编程器;一些大学还开设了这方面的课程。可以预计,不久的将来,在国内将掀起应用和开发 GAL 器件的高潮。

在计算机上输入汉字要规范化

北京(100088) 桂 英

在计算机上输入汉字,就是在计算机上书写汉字,所以有规范用字的问题。目前,我国有许多汉字输入方法,但设计者有些忽视了在计算机上输入汉字时规范用字的问题,有的汉字编码设计者为了排除汉字编码中的重码,人为的规定了反常的书写汉字顺序,和改变汉字的形状与结构。如××××规定追、哉字末笔是横,进、围字末笔是竖,递、困字末笔是撇,运、国字末笔是点,刀、力、选、园字末笔是折等等;又规定把才字三画写成横、竖、撇,于字写成横、横、竖(于与干字没有区分了);还规定秉、练、炼字末三画写成小字,东字第一、二画、长字第二、三画和低字第六、七画均写成七字等等。这明显的改变了汉字的笔画顺序、笔画形状和结构。这对于人们学习计算机汉字输入时,正确掌握和学习汉字笔画顺序、笔画形状和结构是有害的,会把人们引向不规范用字的错误轨道。

笔者希望汉字编码的专家们和有关部门,要重视和关注计算机汉字输入规范用字的问题,这是利国利民,造福子孙后代的大事。

字符串译码编码器

字符串译码编码器采用硬件方法,实现了字符串的单址并行处理。本技术已向专利局申请专利。

字符串是由一定长度的有序的字符组成,它包含着特定的信息,要对其进行译码或编码就需对其每个字符按次序,按长度地进行记忆或记录,然后发出译码或编码信息。字符串译码编码器就是由一组能够在计数器控制下按次序输入字符的;可对输入字符进行记忆或记录的;完成记忆或记录后可发出相应的译码或编码信号的字符串译码单元组成。

本技术的优点:实现了字符串的单址并行处理,提高了字符串处理的速度,改进了字符串的查找方式等。

本技术可用于:字符串指令的译码及搜索,变量表格管理等。

联系人:杨艳忠

联系地址:河北省迁安县松汀首钢普查队

邮政编码:064404