





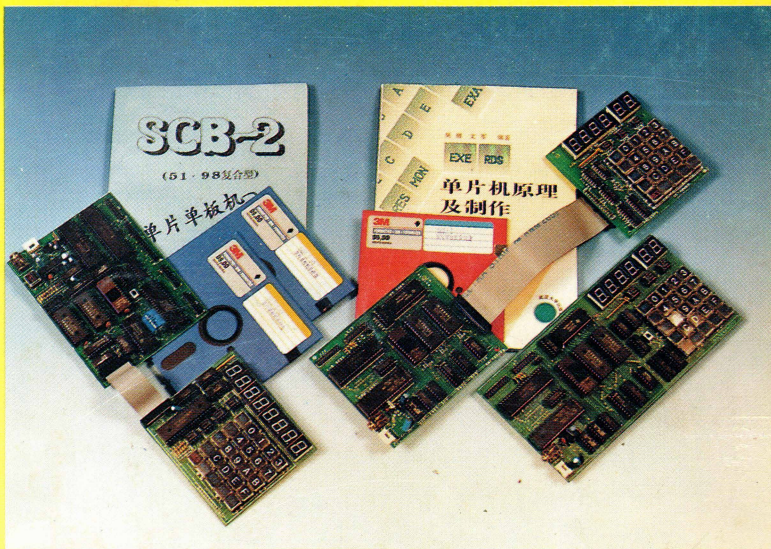


# 武汉尚吉电子研究所

## SCB 系列单片单板机

应用:教学自学 仿真开发 工控仪表(51. 98 两用)

随机资料:系统软盘  
2 张,反通讯,汇编,反汇  
编,debug 调试等文件和监  
控源程序,《单片机原理及  
制作》——SCB-1 软硬件  
详解,《单片单板机原理与  
应用》——SCB-2 软硬件  
详解



地址:武昌珞瑜路  
邮码:430070  
联系人:单部  
电话:716138 716692  
图文传真:(027)716138  
电报挂号:4561  
开户行:工商银行洪山区办  
帐号:2604-67-016072001



# 电子工业出版社广州科技公司

## 邮购书目

书名	定价	邮挂费	出版单位
微计算机的下一场革命—多媒体技术	12.00元	1.80元	电子工业出版社
MS-DOS 结构分析教材之一: BIO 结构分析教程	21.50元	3.20元	电子工业出版社
高频开关稳压电源	9.80元	1.50元	广东科技出版社
开关电源集成电路手册	18.00元	2.70元	中国电子器材深圳公司 (英文资料)
MICROSOFT C6.0 大全	34.00元	5.10元	中国科技大学出版社
亚洲“四小龙”经济与教育	13.80元	2.00元	电子工业出版社
NOVELL 网络安装·使用·开发指南(V2.15~V3.11)	15.50元	2.30元	电子工业出版社
NOVELL 网络及其互联技术	13.00元	2.00元	电子工业出版社
NOVELL 网络故障诊断与维修技术	6.50元	1.30元	电子工业出版社
NOVELL 实用通信编程技术	26.00元	4.00元	新地文化事业出版
Netware 网络操作指南	22.00元	3.00元	新地文化事业出版
Netware 网络应用指南	22.00元	3.00元	新地文化事业版
MODEM 调制解调器技术与应用	14.00元	2.00元	时代科技信息出版

### 本期新书介绍

#### · NOVELL 网络安装·使用·开发指南 (V2.15~V3.11)

汤岳清 康奇林编 李智审校

开本:16开 276页 定价:15.50元 邮挂费:2.30元

该书作者为从事网络软件开发人员,他们在多年实践经验的基础上,参考大量 NOVELL 最新资料编写而成。为网络购买者、安装者、管理员,使用者和应用程序开发者提供最新、最实用的信息资料。全书共分为五大部分。

第一部分是 Novell Netware 产品综述:分别介绍 Novell 集成计算体系结构以及各种产品;Novell 通信产品;TCP/IP 产品和 Netware 开发工具。

第二部分是 Novell Netware 安装指南:分别介绍 NetwareSFT2.15,Netware386 V3.11 的规划,安装方法,网络工作站和打印服务器的安装以及网络设置方法,对 Netware Lite 给出了较详细的介绍。

第三部分是 Novell Netware 用户使用指南,对 Netware386 命令都以实例详加说明,对 Netware 的网络打印作了详细介绍。第四部分是 Novell Netware 网络管理员指南:介绍了 Netware 文件服务器控制台命令及用法等。

第五部分是 Novell Netware 网络应用系统开发指南:介绍 Netware MHs,Netware RPC 以及 Netware 系统接口。C Interface for DOS,C Netware compiler/386 和 Netware SQL/XQL 的具体用法,以及如何在 Novell 网上实现汉字信息处理。

#### · MODEM 调制解调器技术与应用

汤岳清 蒋月勤 郭启纯 编

开本:16 98页 定价:14.00元 邮费:2.00元

介绍 MODEM 分类及标准;如何选购 MODEM;MODEM

安装及测试;MODEM 在电脑通信中的应用;自己编程控制 MODEM 的方法等。

#### · MCS-51、8098 单片单板机软件硬件设计

罗维国 吴微 马国敏 编著 武汉大学出版社

16开 192页 定价:9.80元 邮挂费:1.50元

本书以 CYSCB-2 单片单板机为实例,详细介绍其软、硬件设计和具体实现。资料全部公开,便于读者相互借鉴、资料共享。

全书共分七章,内容包括:单板机应用系统与开发系统概述;CYSCB-2 单片单板机硬件分析和监控程序分析,包括与高位机的通信界面。简要介绍了全屏编辑和宏汇编软件,最后给出了典型应用程序。

结合本书可提供 CYSCB-2 MCS-51、8098 单片单板机(售价:880.00元,邮挂费:20.00元)

#### · 2.13 系列汉字系统用户手册

吴晓军 编著

开本:16 定价:16元 邮挂费:2.4元 晓军电脑工程部出版

本书以 2.13 系列汉字系统最高版本 2.13H 为基础,详细地介绍了 2.13H 汉字系统的安装启动过程、键盘管理模块、汉字显示模块、汉字打印机模块、实用程序等几个主要方面。最后,还介绍了十种可摘挂到 2.13H 汉字系统上的汉字输出方法。

邮购地址:广州石牌华南师大北区一号 203

邮编:510631 联系人:王丽端

汇款请注明收件人详细地址,姓名,邮编,以便能准确及时地将书籍寄上,需要开发票的请注明;零售请到广州五山路华师大科技服务楼 215。谢谢合作。



# 衷 心 奉 献

## ●DP851

(DP851 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. 开发系统板:主板+键盘板(或主板+PC 机)可构成单板型单片开发器。主板上 有 16KB 的监控程序(键盘和 PC 机各占 8KB),具有置入、修改、调试、运行等多种开发功能。

2. 教学实验板:开发器+实验板可构成教学实验板,该板从教学的实际需要出发,具备了实验内容多、覆盖范围宽、结构灵活、性能可靠、价格低廉等优点,是一种大众型的教学实验设备。

3. 目标应用板:开发器+A/D、D/A 等扩展板可构成用户的目标系统(详细情况陆续介绍)。

## ●BJ10 系列

适用于仪器仪表智能化,数据采集,可编程控制等项目的研制开发及应用性计算机产品中的做为嵌入式主机板使用(详细情况陆续介绍)。

## ●BSY 系列

BSY01—07 分别具有 I/O 扩展;键盘及显示控制;A/D、D/A 变换等多种功能。用户可根据不同用途自行选择组合(详细情况陆续介绍)。

## ●TDS—3000 系列

TDS—3000 系列数字显示控制仪表:本系列仪表采用微机技术,实现电热函数的转换,彻底介绍了温度传感器的热电非线性问题,在整个测量范围内不存在非线性误差。微机定时控制的自动稳零系统,保证零点不会漂移。

本系列仪表还能配各种标准电压,标准电流输入,可与任何线性的传感器、变送器配套。还具有断偶保护及报警灯光显示,采样灯光显示等功能。

## ●SW 系列

本厂数字仪表共分三大类十个方面的一百多个品种,该仪表与不同传感器配合使用,可测量温度、压力、电流、电压等多种物理量,并能实现指示调节和报警。

各种仪表均可按用户要求带有 0—5V, 1—5V, 0—10mA, 4—20mA 等一组标准信号输出。

本仪表测量精度高,显示直观,抗震性能好,使用寿命长(详细情况陆续介绍)。

## ●为了便利用户,本厂愿为用户邮购各种电子元器件,各类芯片等。

地 址:北京西城区月坛南街 83 号

开户行:工商银行百万庄分理处

邮 编:100045

帐 号:014—047014—96

电 话:8512947

联系人:张文奇 庞文姬



# 电子与电脑

一九九三年  
总期第 105 期

## 目 录

- 综述 •
  - 令人翘首企盼的 POWER PC ..... 吴腾奇(2)
  - 计算机硬加密原理及技术 ..... 李晓中 张景生(4)
- PC 用户 •
  - INT28H 与常驻内存的程序设计 ..... 罗颖(6)
  - Super-Star 图文排版系统与高级语言接口程序设计 ..... 张辉 张敬民 袁艺(8)
  - 新世纪/XqR 病毒的检测与清除 ..... 车光宏(10)
  - DOS 5.0 的 CONFIG.SYS 的设置 ..... 郭文典(13)
  - C-WordStar 使用基础问与答 ..... 朱大公(14)
  - 计算机病毒的检测、消除和预防 ..... 苏民生(16)
- 学习机之友 •
  - 把 & 和 CALL 语句“嫁接”到赋值语句中去 ..... 邹习新(18)
  - ProDOS 系统内部结构剖析(续) ..... 廖凯(20)
- 语言讲座 •
  - 第十二讲 D1.0 版本补遗 ..... 丁志伟(22)
  - 第十三讲 FORTH-83 标准摘要 ..... 丁志伟(25)
- 学用单片机 •
  - 固体录音机的制作 ..... 罗明宽 车金相(28)
- 电脑巧开发 •
  - 实验教学计算机的研究与开发 ..... 王勤民(31)
  - 半导体存储器在自动打铃器中的应用一例 ..... 刘长茂 邱杰(33)
  - 新型实时日历时钟 MSM6242 ... 徐民 富歌(35)
- 维修经验谈 •
  - AST286 系统板总线故障的分析与排除 ... 梁杰熙(32)
  - 中环 CT100 终端常见故障维修 ..... 徐俊(38)
  - 菊花遥控彩电“锁不住”故障分析 ..... 李桂洁(39)
- 电脑游戏机 •
  - 第三章 6527CPU 的音响系统 ..... 于春(40)
- 电脑通信 •
  - 基于图形的联想多媒体工具 GBH ..... 白成林(43)
- 传真机专题讲座 •
  - 微机——传真机系统 ..... 刘保元 张景生(44)
  - 大连华联商厦的计算机网络 ..... 廖涛(46)
- IC 电路应用 •
  - COP840C 芯片在洗碗机自动控制中应用 ..... 王求是 季行健(48)
  - 乘法器 IC 及应用 ..... 李兰友(50)
- 读者联谊 •
  - 经验数则 ..... 曹彤(51)
  - 消毒软件 CPAV 的局限性 ..... 张志(51)
  - 1993 年总目录 ..... (52)

机械电子工业部电子工业出版社主办

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

印刷:北京三二〇九厂

国内总发行:北京报刊发行局

国内统一刊号:CN11-2199

邮发代号:2-888

国外代号:M924

出版日期:每月 23 日

主编:王惠民 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



# 令人翘首企盼的 POWER PC

广州华南师大物理系(510631) 吴腾奇

IBM 与苹果(Apple)这两个电脑世界的巨无霸,经过十年的市场激烈竞争之后,于一九九一年七月突然戏剧性地宣布彼此联合,再加上另一家半导体公司——摩托罗拉(MOTOROLA)共同签署了一份技术合作协议,三架马车联手发展新一代的个人电脑系统——Power PC。

消息传出,难免引起业界的震惊,深为市场将被垄断而担忧。然而,时至今日却只见有联合声明,未见有实际行动。这又使人们对商场竞争中的真真假假、虚虚实实,感到迷茫了。但无论如何,即使他们的计划流产了,从 IBM 与苹果的互相竞争到联合发展的过程中,多少可以看到电脑产品发展的趋势,了解一下这两霸经营策略的变化,就不是毫无意义了。我国八十年代初期主要大量进口苹果机,到了九十年代初期八十年代末期,随着 IBM PC 及其兼容机价格的下降及其产品的先进性能的提高,又大量进口 IBM PC 机,形成我国目前微型电脑应用的主要潮流。随着我国微电脑生产的发展,现在主要采用 Intel 的 80286、386、486 微处理器用作微型计算机的中央处理单元。我国目前年产量 10 万部的计算机就主要停留在这个水平上,基本上选择 IBM PC 机的发展水平。

## 为什么会出戏剧性的联合?

先从势成水火的竞争说起。一九七七年,由两名辍学的大学生创办的苹果电脑公司,在短短的两三年时间内,竟然创造出年营业额达十亿美元的奇迹,引起了业界人士的瞩目,也给当时已处于电脑业“大哥大”地位的 IBM 带来沉重的压力。根据有关资料显示:一九八一年,苹果公司出品的世界第一部个人电脑——苹果 I 型,已经以其十五万部的销售量而占据私人电脑市场的领导地位。这时 IBM 产品的销量还不到两万部。悬殊的销量令 IBM 对电脑市场的潜力重新估量,面对苹果这位后来居上,来势逼人的电脑新星,绝不能掉以轻心。IBM 所作的快速反应是:为争分夺秒收复失地,在设计上采用了一种“开放”政策,尽量采用市场上已有的设备和技术,并依赖其它公司的研究成果,针锋相对推出一种新型的个人电脑 IBM PC。

IBM PC 面世,立时受到用户和应用软件商的支持和捧场,市场占有率一下子由零跳升至接近四成,可以说是将原属于苹果的个人电脑领导地位夺过来了。从此,也就开始了在电脑市场中 IBM 与苹果势成水火的交恶。

一九八四年苹果为对付 IBM PC,推出了麦金多(MacIntosh)个人电脑系列,取得了较佳的利润。麦金多在技术上是专利垄断的,避免了由于竞争者多而摊薄利润,这是取得较佳盈利的主要原因。但也正由于它的作业软件系统未得开放而很难成为主流电脑,这样,苹果也就一时间还难成为个人电脑的领主。

在 IBM 与苹果的斗争中,IBM 赢得的只是声誉,远没有获得应有的利润。这也许就是当它们展开竞争时,就埋下了联合的契机了吧。

IBM 万万没有想到:它在设计上采取的“开放”政策,除了吸引用户外,亦吸引了大量厂家争相制造 IBM PC 兼容机。当市场上挤满了 PC 兼容机的时候,残酷的个人电脑市场价格战也就开始了。IBM 从零跃升至百分之四十的毛利率优胜纪录并没有维持很久,在以后的日子里出现的是利润不断下降,到九一年,市场占有率降至接近两成的水平,利润大不如前。

藉 IBM 与苹果之争而养肥了的,是 IBM 所依赖的两家公司——Microsoft 和 Intel。前者是一家软件研究公司,它为 IBM PC 编写操作系统,后来又把这个系统以昂贵的价格卖给其他电脑软件发展商,牟取暴利;后者是 IBM PC 机微处理器的供应厂商。IBM 虽然说是依赖其它公司的研究成果,其实是 Microsoft 和 Intel 无形中得 IBM 之助,使它们分别掌握的技术——系统软件及微处理器——沾了名牌之光而成为这领域的霸主,在无其它竞争者可加入的情况下,独享成果。

IBM 推行的个人电脑策略的结果是肥了别人瘦了自己,这是不是决策的错误?追究这个问题已经没有意义。但从这事件中得出的一条经验却是非常宝贵的,那就是:要在私人电脑市场上掌握主动权,就必须掌握半导体和系统软件的尖端技术。

IBM 的经验对苹果来说也同样重要的。

苹果自一九七七年创立以来,即自成一格,不与他人分享系统软件,这种自我垄断的局面,虽然可使苹果产品的销量高企,但也造成它的负面损失,那就是苹果的产品始终难成为市场的潮流。十五年来苹果共推出四种主要的产品,其中苹果 I 及麦金多个人电脑,销量可观,但丽莎(LISA)和苹果 II 就令人失望了。可见从产品差异化来获取高利润的策略,并不是任何时候和任何条件下都行得通的。产品差异化令自己的技术保守,成长也就减缓下来。以电脑市场成长最快速的膝上型个人电脑来说,苹果九一年推出的笔记型个人电脑



虽然一炮而红,但比其它型号的先行者已晚了两年。

IBM PC 兼容机充斥市场,竞相降价所造成的烂市,对任何牌子的电脑的销量,或多或少都会造成影响。在销量难有增长的情况下,苹果的董事长史卡里(Sculley)改变策略的考虑已经成熟——这个新策略就是寻找一个主流厂商合作,推出新一代的操作系统,而且,这个主流厂商已经属意 IBM。

IBM 呢,当它明白在原有 PC 系列上,已难取得增长和利润时,也早有另起炉灶的打算。它正需要一个改写个人电脑系统的得力拍档,苹果成了它的首选。

自古商场如战场,没有永远的敌人,也没有永远的朋友。当 IBM 与苹果都有肥了他人瘦了自己的共同感受时,原先势成水火的对立,似乎已经由于市场的无情而消弥了交恶。于是,苹果高层与 IBM 接触,双方心有灵犀一点通,一拍即合了。

可以看出,IBM 与苹果的联合,是一场互相利用的结盟。彼此都得到于己有利的好处,这就是 IBM 和苹果为什么从势成水火的交恶戏剧性地转为合作伙伴的基本原因。具体地说就是:

1. 苹果与 IBM 联合可以减低投资风险,而且可以避开重蹈一味的“苹果”自行开发所产生的差异化的覆辙。

2. 联合后,苹果可以得到快速的精简指令集计算(RISC)技术,这将有助于麦金多电脑增添许多功能,如阅读潦草手写文字之类。

3. 苹果公司派出大批工程师和 IBM 共同工作,可以吸取 IBM 的技术知识,因为苹果公司也想制造伺服器、中型电脑,作为联系桌上型电脑的中继站。

4. IBM 与苹果及摩托罗拉合作,可以减轻对 Microsoft 及 Intel 两家公司的依赖。在联合产品上掌握半导体和系统软件技术上的控制。

5. IBM 与苹果联合,可以得到苹果专长的图形界面技术的转让,这样的转让能令 IBM 的软件获得像麦金多上的软件一样美观的图形界面。

#### 新一代电脑“新”在哪里?

一九九一年十月二日,IBM、苹果及摩托罗拉宣布三家公司合作的主要内容是:开发一种新的电脑系统,以针对现时的个人电脑市场。新产品暂名为 POWER PC(Performance Optimization With Enhanced Risc 的字母缩写),它将承袭现时颇为成功的 IBM 工作站系统——RS-6000 的设计。IBM 与苹果将为新产品编写操作系统及界定有关之界面标准。摩托罗拉则负责提供新产品系统的微处理器硅片。

那么,这三家合作出品的 POWER PC 新在哪里呢?为什么它要承袭 RS-6000 系统的设计呢?

简单的说,POWER PC 的新特点,就在于它采用 RISC 快速技术,又保留操作系统的兼容性。

#### 1. 采用 RISC 快速技术

RISC(Reduced Instruction Set Computing 的缩写,中文全名可译为“精简指令集运算”)技术的采用,集中表现在微处理器上,它的最大优点是运算的速度

快。

今天的个人电脑,无论是 IBM 的兼容机还是苹果的麦金多,所使用的微处理器都是 CISC 系统(中文全名可译为“复杂指令集运算”),这是六十至七十年代的产物。这种微处理器接收指令以后要通过“只读存储器”ROM 的翻译,电脑才能执行,如果是复杂的程序,所花时间就比较长了。

RISC 微处理器,不需要 ROM 上解码指令,解码的任务交由随机存取存储器 RAM 即可。这样,POWER PC 所需要的硅片便大为减少,同时,由于改良了浮点运算功能,RISC 可同步执行五条指令,比一部运作速度 33MHz 的 386PC 速度快 4 至 5 倍。

RISC 微处理器的发展,技术上主要集中于快速的运作上,其潜力是不可限量的。因为目前 RISC 微处理器只运用于昂贵的工作站或超级电脑上,而 IBM 与苹果及摩托罗拉发展的 POWER PC 只是中价位(主要针对“膝上型”电脑和低档电脑用户)。计划一九九二年面世的 POWER PC,是先发展运算速度可达 50MHz 的微处理器,但即使如此,POWER PC 在人工智能、语言及书写字体的辨认方面,已有长足的进展,根本不需要输入许多指令,而是直接口述命令,要电脑从远方的数据库读取资料,也可根据不同的资料来源,展示全方位的影线。

#### 2. 操作系统的兼容性

前面提过,POWER PC 将承袭 IBM 工作站系统——RS-6000 的设计,所以谈到 POWER PC 新系统的兼容性特点时,就要从这里谈起。

RS-6000 的硬件中心部分结构是围绕着一个由五块硅片组成的核心而建立;POWER PC 的新系统亦会采用类似结构,但核心部分则由摩托罗拉公司采用高科技把五块硅片的功能,全部压缩在一块硅片内,这块硅片只有指头般大小,却安置六百万个晶体管,而精细程度达到 0.8 至 0.5 MICRON,这是 POWER PC 在微处理器上的一项革命性的技术突破。

另一方面,RS-6000 系统是采用 IBM 的 AIX 为操作系统的,AIX 又是 IBM 自己开发的 UNIX 版本(UNIX 为业界承认之多用户操作系统标准);因此,POWER PC 承袭 RS-6000 设计系统,编写应用软件就可以兼容 IBM 的 UNIX 以及苹果公司的 UNIX 版本——A/UX,而吸引不少国家。这也是出于争取用户,扩展销路的考虑。

#### POWER PC 的前景

POWER PC 是电脑巨无霸的产儿,在技术上具有先天的优越性,作为中价档的电脑新一代来说,技术上无疑也是一大突破,而且按照 IBM 与苹果的计划,到一九九五年,在技术上还将有更大的突破,将发展一片比现时 RS-6000 的运算快得多的微处理器,其运算速度最高可达 500-SPECMARKS,运作速度则达 100MHz,这都是令用家感到鼓舞的前景。

然而,一九九三年已经过去一半多了,中价位的 POWER PC 是否能如期推出呢?目前似乎还没见到迹



象,因此令人对三架马车合作的成功性产生怀疑。问题不是不信任这三家公司的技术力量,而是在于大公司的联盟往往是大有大的难处。他们不会因为联合而淹没了自己。他们一方面联合(互相利用)出新产品,同时自己也在继续出产自己的传统或创新产品,这个关系处理不当,就或多或少会影响到合作的关系。

就电脑市场的形势来说,新产品的推出层出不穷,时间是一个重要的因素。各种力量联合推出新产品的酝酿,并非 IBM 与苹果及摩托罗拉所独有,谁能将产品抢先推出,喝得“头啖汤”谁就是迈出成功的一大步。时间越拖得长,任何现在看来是尖端的计划产品,到真正面世时,可能已经远远地落后了。

# 计算机硬加密原理及技术

李晓中 张景生

## 1. 激光加密

激光加密方法是近几年国内外出现的新的加密手段。它的基本原理就是在专用的设备上利用激光的能量破坏软磁盘上特定的扇区的磁介质。由于激光束很细,由此产生的孔肉眼是无法看到的。由于被穿孔的扇区出现了坏字节区,因此,在读这个扇区时,CRC 检验值总是错误的,从而在软盘中可以此作为识别原盘的标志,而且这个标志是无法拷贝的,用同样的手段也难以与原盘做到同一字节区形成坏字节区。

对于不同的软盘用激光破坏不同扇区的某几个(任意)字节时,每一张盘就有了自己的标志,相当于磁盘有了“指纹”。而这种指纹在很大范围内具有独一性。

美国 VAULT 公司近几年推出的激光磁盘及相应的 PROLOK 和 FILELOK 软件就是以软盘指纹作为加密密钥进行处理的,现以 PROLOK 软件为例叙述保密原理。在已经建立“指纹”的软盘上,存有用户程序,一旦用户程序被 PROLOK.EXE 程序加密,以后每当运行加密后的用户程序时,它首先查寻软件上的指纹并将其与加密程序内记载的原始指纹比较,如相同,则运行程序;否则退回系统。用户也可以在用户程序内任何地方输入检查指纹的代码,但由于程序的执行总是脱离不了软盘上的物理指纹,而物理指纹又不能随文件的拷贝而复制,因此达到了保密目的。

## 2. 使用 KEPROM 实现系统的硬加密

INTEL 公司利用它生产 LSI 电路的优势,在 1984 年底推出了“上锁”EPROM (KEPROM) 27916。使用它可实现永久性的软件加密。此芯片对用户是透明的,性能是可靠的。对于非法复制由它写入储存器的程序和数据,系统将无法运转。

27916 KEPROM 的基本性能及加密原理:INTEL 公司生产的 KEPROM 27916 是与 27128 兼容的 16KB 的 EPROM。27916 最大的特点是写入它的信息将进行保密,任何复制的 EPROM 芯片都无法在系统中使用。它的加密原理是,凡是进入一个 KEPROM 系统的用户,必须有一个含有 KEPROM 的设备,而且该设备

的 KEPROM 中锁匙密码一定与所进入系统中 KE-PROM 设定的锁匙密码一致。换句话说,在使用含有 KEPROM 系统时,至少要用两片事先处理过的 KE-PROM 芯片(已经写入相同密码锁 XX 的二片 KE-PROM)。将一片安置在主机内,另一片安在用户终端内,开机时系统自动地由两片 KEPROM 进行类似的双向鉴别信息交换。当双方鉴别结果符合原来设定的密码时,终端上的用户才能使用系统,否则主机拒绝用户使用系统。

27916 KEPROM 锁匙密码是存放在一个长 64 位的“只写”存储器内,且该“只写”存储器内是一旦对 27916 配置完成后(写入锁匙密码,出厂时写入)写入的密码内容就不能再读出。它最大可以有  $18 \times 10^{18}$  种不同的锁匙密码,如此大的密码,对要窃取密码的非法入侵者来说是极其困难的。而锁匙密码对一个已有使用权的用户来说,既不要破密更不用记住,因为锁匙密码用户写入设备的 KEPROM 中,对用户是透明而又省事的,一开机便可进入。而非法复制的复写品,不具有密码锁匙的,它无法进入系统。

## 3. 硬盘引导区的“上锁”与“开锁”

### (1) “上锁”与“开锁”的基本原理

近年来,我们使用长城 0520CH 及长城 286B 等机器中,由于上机人员杂、外来磁盘管理不善,经常发生硬盘系统被“磁盘病毒”(诸如“大麻病毒”)感染,而无法再用硬盘启动的故障。一般的病毒免疫程序,只能防止病毒的进一步感染,而无法使硬盘再自行启动。要解决这一问题,往往需要对硬盘的主引导区(BOOT)和 DOS 的引导区重新进行格式化,而格式化又要破坏原来硬盘的数据,因此,给工作和使用用户带来了极大的麻烦。我们在实用中,采用了一种硬盘“上锁”的办法,使得一般用户(未经允许或未检查所带磁盘病毒情况)不能直接用硬盘启动,从而避免了磁盘病毒对硬盘引导区的感染(磁盘病毒一般必须经启动引导区才会感染)。当“上锁”者使用时,首先用“开锁”软盘启动(绝对无病毒感染的系统磁盘),并自动对硬盘进行开锁。开



锁后,系统再进行一次热/冷启动(抽出软盘,由硬盘自行启动),从而进入硬盘系统管理之下(DOS 管理之下)。这时,用户可以使用系统提供的一切功能及运行自己的用户软件。关机时,再使用一次“上锁”软件,对硬盘引导区进行“上锁”以防止其它用户启动硬盘,从而实现了硬盘加密的目的。

## (2) 实现方法

实用中,首先将硬盘的主引导区的标准数据,在 DEBUG 状态下,写到预先格式化好的一张空软盘的第 39 道的第一扇区上,作为将来恢复硬盘引导区的备份,然后将硬盘主引导扇区的数据(已被 DEBUG 读入内存缓冲区)进行无规律的“破坏”,也就是所谓的加密。当然,加密的方法是多种多样的,以不使其它非法用户破译为原则。最后将加密后的缓冲区数据写到备份软盘的第 38 道的第 1 扇区上,作为“上锁”硬盘引导区的密码数据。

我们自己编写一段加密硬盘引导区和解密硬盘引导区的程序,并放在备份软盘上(使用汇编语言,作为普通用户软件加密保护)。下面我们分析以下加密和解密软件,大家很容易看清它的实现方法。

用下述方法对硬盘加锁之后,即使非法用户用自己的软盘启动系统后,试图进入硬磁盘时(如键入 C:回车),屏幕则显示:

```
INVALID DRIVE SPECIFICATION
```

(无效的驱动器说明)

无法进入硬盘,硬盘自然不会被病毒所感染,从而保护了硬盘。

加密上锁程序:

```
STACK SEGMENT PARA STACK 'STACK'
      DB 64 DUP(0)
STACK ENDS
DATA SEGMENT PARA 'DATA'
DAT1 DB 512 DUP(0)
DATA ENDS
CODE SEGMENT PARA 'CODE'
BEGIN PROC FAR
      ASSUME CS, CODE, SS, STACK, DS, DATA,
            ES, DATA
      PUSH DS
      MOV AX, 0
      PUSH AX
      MOV AX, DATA
      MOV DS, AX
      MOV ES, AX
      MOV DX, 0000H
      MOV BX, OFFSET DAT1
      MOV CX, 2601H
      MOV AX, 0201H
      INT 13H
```

```
      NOP
      MOV DX, 0080H
      MOV BX, OFFSET DAT1
      MOV CX, 0001H
      MOV AX, 0301H
      INT 13H
      RET
```

```
BEGIN ENDP
CODE ENDS
      END BEGIN
```

不难看出,它的功能就是将软盘 A 驱的第 38 道第一个扇区的数据写入硬盘的第一扇区,作为硬盘的主引导区数据信息。

解密解锁程序:

```
STACK SEGMENT PARA STACK 'STACK'
      DB 64 DUP(0)
STACK ENDS
DATA SEGMENT PARA 'DATA'
DAT1 DB 512 DUP(0)
DATA ENDS
CODE SEGMENT PARA 'CODE'
BEGIN PROC FAR
      ASSUME CS, CODE, SS, STACK, DS, DATA,
            ES, DATA
      PUSH DS
      MOV AX, 0
      PUSH AX
      MOV AX, DATA
      MOV DS, AX
      MOV ES, AX
      MOV DX, 0000H
      MOV BX, OFFSET DAT1
      MOV CX, 2701H ;39道,1扇区
      MOV AX, 0201H
      INT 13H
      NOP
      MOV DX, 0080H
      MOV BX, OFFSET DAT1
      MOV CX, 0001H
      MOV AX, 0301H
      INT 13H
      RET
BEGIN ENDP
CODE ENDS
      END BEGIN
```

此程序是将 A 驱的第 39 道一扇区的的信息传递到硬盘的第一扇区上,作为主引导信息。39 道一扇的信息是硬盘主引导区的标准信息,用它启动硬盘,成功是自不必赘述的。



# INT 28H 与常驻内存的程序设计

华中理工大学计算机系(430074) 罗颖

在大量的 DOS 中断调用中,人们对 INT 28H 似乎很少问津,甚至头脑中对它没有多少概念。其实这也不足为奇,因为 INT 28H 是一个未曾公开发表的 DOS 特性。然而,尽管 INT 28H 在人们面前是这么模糊,但这一点却是千真万确的事实:即当 DOS 在提示符下等待按键时,或处于安全状态时,它要不断地调用中断 28H。笔者在编写常驻程序的时候,正是应用了这一特性,使常驻程序安全可靠,获得了很好的效果。现把它的特性逐一介绍。

## 一、神秘的 INT 28H 及其生存背景

我们知道,内存驻留程序的设计是很有难度的。其根本原因要归咎于 MS-DOS 的独特性质了。因为 MS-DOS 每一时刻只能运行一个程序,即单任务操作系统;另一方面,MS-DOS 又提供了程序终止但又可以驻留的例程,简称 TSR,TSR 形式上象多任务的工作方式,它允许一个程序在另一个程序运行时仍在内存中保持着活跃状态。

基于第一个原因(单任务),当编程者编写的应用程序要常驻内存时,原则上不得使用 DOS 调用,对 BIOS 的调用也要受到一定的制约。因为是单任务,自然就不允许 DOS 的再入了。所谓“再入”,也就是不能在同时刻运行两个 DOS 服务。

INT 28H 这个鲜为人知的中断调用,可以协助我们解决这个问题,使常驻程序能获得更多的机会和更安全的利用系统资源,包括使用 DOS 调用。

当然,如果常驻程序仅涉及控制台的 I/O 操作,也就另当别论了。

大家知道,在 DOS 的运行过程中,有些程序片断是不允许被打断的。我们称 DOS 的这些程序片断为“关键片断”,把“关键片段”的其它状态称为“安全状态”。需要指出的是:当 DOS 处于“安全状态”或 DOS 提示符下时,DOS 将不断调用中断 INT 28H,这种特性正是我们要编制常驻程序时研究它的价值所在!这一点,DOS 技术参考书中是没有正式公布的。

## 二、INT 28H 应用技术分析

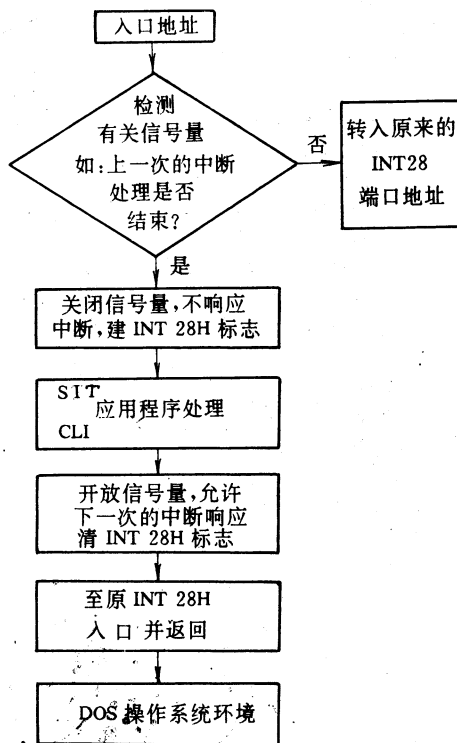
内存驻留程序有两个阶段:暂驻阶段和 TSR 阶段。TSR 的形成必须由暂驻程序通过调用 INT 27H 或者 INT 31H 之一来实现。如果暂驻程序调用 DOS 的过程中 TSR 也调用 DOS,其结果将是灾难性的,会导致整个系统的崩溃。另外,当常驻的应用程序牵涉到磁盘文件的 I/O 端口地址的 I/O 时,也要考虑其安全问题,否则也会引发混乱甚至系统崩溃。由于 DOS 只有处于安全状态时才去调用 INT 28H,因此我们可以这样处理:安排一个新的 INT 处理程序段,把程序的

激活入口地址放在新设置的 INT 28H 入口处,当应用程序返回后,才恢复原来的 INT 28H 功能并返回。这样一来,就保证了只有在用户选择了激活功能键而且 DOS 又处于“安全状态”时才去激活常驻内存!

## 三、编程人手

如果你使用过 PRINT.COM 程序,你就会很清楚,它在 DOS 环境下提供了一个假脱机打印功能。用户把待打印文件(FCB 形式)送至打印队伍排列后,可继续进行其它操作。PRINT.COM 程序利用后台操作时间片,逐个将文件从盘中读出,送至打印机输出。

PRINT.COM 驻留程序以 INT 1CH 和 INT 28H 的中断处理程序调用后台打印输出子模块。这里的 INT 1CH 和 INT 28H 就是后台打印进程的两个激活源。PRINT.COM 程序及几乎所有的商业性常驻程序其编程的结构框图如下所示:





有了这个基本形式,编制成功的常驻程序就不难了。当然,随着应用程序的复杂度提高,这种结构并不是不能改变的,但其基本思想却描述了常驻程序的编程原理。

其入口地址的建立可以通过调用 DOS INT 21H (AH=25H)来实现。为了加深读者对上述内容的理解,下面提供一个简短的程序,运行后,屏幕上不断显示“\*”字符。这个小程序没有实际价值,旨在帮助读者更好地体会本文的编程思想。

```
cseg      segment
          assume cs:cseg,ds:cseg
          org 100h

start:
          jmp initialize

string    db "Hello you are welcome!",13,10,"$"
t         db "0"
old_int28 dd ?
new_int28 proc far
          sti
          mov al,t
          or al,00h
          jz tt
          jmp old_int28
tt:       mov t,1
          call old_int28
          mov ah,2
          mov bh,0
```

```
mov dh,20
mov dl,25
int 10h
mov ah,9
mov al,"*"
mov bh,0
mov bl,7
mov cx,1
int 10h
iret

new_int28 endp

initialize:
          assume cs:cseg,ds:cseg
          mov bx,cs
          mov ds,bx
          mov al,28h
          mov ah,35h
          int 21h
          mov word ptr old_int28,bx
          mov word ptr old_int28[2],es
          mov dx,offset new_int28
          mov al,28h
          mov ah,25h
          int 21h
          mov dx,offset initialize+1
          int 27h

cseg      ends
end start
```

(接第 13 页)

Stacks 是用以设定 DOS 的内部堆栈区的大小。在一般的机器上,大可设定 Stacks=0,0 以增加近 2KB 的可用常规内存;但在高速的机器(如 386DX)上,此举可能会降低系统的速度。另外,少数程序在设定 Stacks=0,0 后将不能正确运行。若用户运行 Windows,建议设置 Stacks=9,256。

Shell, DOS 的默认命令处理器为 COMMAND.COM,如果用户觉得不合适,可用 shell 命令指定其它命令处理器来代替它,这些命令处理器通都是增强型的,例如 4DOS.COM、NDOS.COM、DOSSHELL.COM 等等。须留心的是对这些命令处理器所必需的文件(如 NDOS.OVL、DOSSHELL.EXE)都必须放在根目录下,否则将可能引起启动失败。Shell 命令还可用来改变环境的大小。例:Shell=NDOS.COM。

Switches=1k 将迫使 DOS 放弃使用扩展键盘功能(尽管已经安装),它常用以解决 MS-DOS 5.0 与一些中文系统的冲突。

下面给出两个实例,第一个用在 286 上(一兆内存),第二个用在 386 上(4 兆内存):(附注解)

```
①Device=c:\DOS\HIMEM.SYS;安装 XMS
DOS=High          ;将 DOS 放在 HMA
Stacks=0,0
Files=25
```

```
Buffers=30,4
Device=c:\DOS\SETVER.EXE
Device=c:\DOS\RAMDRIVE.SYS320 512 64/E;安装
虚拟盘
Switches=1k
②Device=c:\DOS\HIMEM.SYS/INT15=360;预留
360KB
Device=c:\DOS\EMM386.EXE NOEMS;没有 EMS
DOS=High,UMB      ;使用 UMB
Files=30
Buffers=10,0
DeviceHigh=C:\DOS\SETVER.EXE
DeviceHigh=D:\MOUSE\GMOUSE.SYS
DeviceHigh=C:\NU\NCACHE.EXE/INSTALL;安装
高速缓冲
Shell=c:\NU\NDOS.COM;用 NDOS 代 COMMAND
Switches=1k
用上述的 CONFIG.SYS,一般都能达到近 610KB
的可用常规内存。
```

MS-DOS 6.0 提供了全新的 Multi-Config 功能,可在同一个 CONFIG.SYS 中设定多种不同的配置供用户在开机时选择(以菜单形式),这使得用户非常容易地改变配置而无需每次都要重新编辑 CONFIG.SYS。



# Super-Star 图文排版系统与高级语言接口程序设计

沈阳东北大学机械工程学院(110006) 张辉 张敬民 袁艺

Super-Star(简称 SPT)是香港金山公司开发的图文排版系统。它与 SPDOS 和 WPS 软件配套使用,可实现图形和文字的混排。随着字处理软件 WPS 的广泛使用,SPT 也逐渐得到流行,成为科技工作者写作科技文章的得力工具。使用 SPT 可以在计算机的屏幕上直接编辑和绘制插图,完成科技文章中的插图制作和图文混排,还可以通过图形扫描器对已画出的图形进行扫描输入。但是在使用 SPT 的过程中,我们发现它的绘图功能虽然很强,却没有给出与高级语言联接的接口,致使一些由高级语言程序计算得到的以及通过测试仪器得到的数据曲线等复杂结果不能直接使用 SPT 来处理,而必须首先将其绘出后方能通过扫描器输入,不但输入效果差而且费工费时,对于那些没有配备扫描器的用户只好望图兴叹。这给许多科技文章的图文混排造成了极大的不便。为了解决这个问题,我们以 SPT 的 V1.10 版本为例,分析了 SPT 非压缩格式(E 格式)图形文件的结构,开发了 SPT 与高级语言接口程序的设计方法。通过这一接口程序可直接生成 SPT 的图形文件,供其编排处理,使得 SPT 更加尽善尽美,锦上添花。图 1 为应用本文介绍的方法形成 SPT 图形文件后,与 SPT 的样板文件拼接后的结果。



图 1 接口实例

## 一、SPT 图形文件的格式及形成方法

SPT 输出的图形文件以象素点的形式存放图形,是一个二进制文件。它有一个 64 字节长的文件头,其中第 1 至第 15 字节为 SPT 图形文件的标识符,其内容为“Super-Star File”,第 16 字节为标识符结尾标志,为 16 进制的 1A(十进制的 26),在第 34 和 35 字节为以象素点为单位的图形宽度(必须为 8 的整数倍),低位字节在前,高位字节在后,第 36 和 37 字节为图形的高度;因此 SPT 图形的最大尺寸为 65528(宽)×65535(高);文件头的其余字节可填入 0。在文件头之后为 SPT 的图形数据。由于 SPT 形成的图形为黑白两色图形,所以图形数据中的 1 位(bit)对应图形的一个象素点,当某一位为 1 时,这个象素点为白色,否则为黑色,这样 1 个字节(byte)图形数据可存放 8 个象素点,图形数据按行存放,其最高有效位(bit 7)对应于连续 8 个象素点的最左端象素,最低有效位(bit 0)对应于连续 8 个象素点的最右端象素。假设图形的宽和高分别为 Width 和 height,则存放一幅图形所需的图形数据字节数为:

$$\text{total.byte} = (\text{width}/8) \times \text{height}$$

这样我们可以在高级语言程序中开辟一个长度为 total.byte 字节的一维数组,在这个数组中按图形算法填入相应的象素点信息后,将文件头和图形数据按二进制格式写入一个文件中即形成 SPT 的非压缩格式的图形文件,图形的坐标原点为左上角。

从以上分析可以看出,SPT 图形的基本单位为象素点。而形成图形的基本图素为点和直线,其它图素可由这两个基本图素构成。为此,我们接下来详细介绍这两种基本图素在上述的一维数组中的形成方法。

## 二、点的画法

如前所述,在一个字节的图形数据中含有 8 个象素点的信息。所以在画一个点时必须找出其所在的数据字节在一维数组中的位置及其在该图形数据中的所在的位(bit)号,然后才能在相应的位上进行操作,填入相应的象素点信息。

假设所要画的点的坐标为(x,y),则该点所在数据在数组中的位置为:

$$\text{byte} = y \times (\text{width}/8) + (x/8) + 1$$

该点在图形数据的位号为:

$$\text{bit} = 7 - (x \bmod 8)$$

式中(x mod 8)为取 x 除以 8 的余数。

由于一个字节中含有 8 个象素点的信息,所以在写一个点时必须保证其余 7 个点的信息不变。解决这



个问题的最好办法是使用位屏蔽,除了与将要修改的那点所对应的位以外,屏蔽字节中其余各位都置1,以这个屏蔽字节与原来数据字节相逻辑“与”(AND),逻辑“与”以后的值再与相应的点设置字节相逻辑“或”(OR),或者相加即完成该点的画点操作,当要画的点为白色时,对应的点设置字节及其位屏蔽字节如表1所示。

表1 位屏蔽与点设置字节

位(bit)	位屏蔽字节	点设置字节
0	254	1
1	253	2
2	251	4
3	247	8
4	239	16
5	223	32
6	191	64
7	127	128

用一个二维数组 bitm(8,2)将这个表放入内存中。并假定存放 SPT 图形数据的一维数组名为 SPT,则写点的算法为:

$$\text{spt}(\text{byte}) = (\text{spt}(\text{byte}) \text{ and } \text{bitm}(\text{bit}, 1)) + \text{bitm}(\text{bit}, 2)$$

### 三、直线的画法

有了画点的方法后,我们就可以通过画点来进行直线的绘制。目前,在计算机图形学中,最好的画线算法是 Bresenham 算法,它不需要实数运算,也不需要使用乘法,取而代之的是通过一系列加法和减法隐含地处理 x 和 y 方向距离改变的比率,算法简单迅速。它的基本出发点是记录下每个点理想位置与实际位置之间的误差大小  $\text{err}_x$  和  $\text{err}_y$  (x 方向和 y 方向的误差),在画线循环的每一次叠代过程中,变量  $\text{err}_x$  和  $\text{err}_y$  分别增加在 x 和 y 方向的总变化量。当某一误差值达到预先设置的限制时,相应坐标加 1,同时重新设置这个误差值,这一过程一直持续到画出整条线。详细框图如图 2 所示。

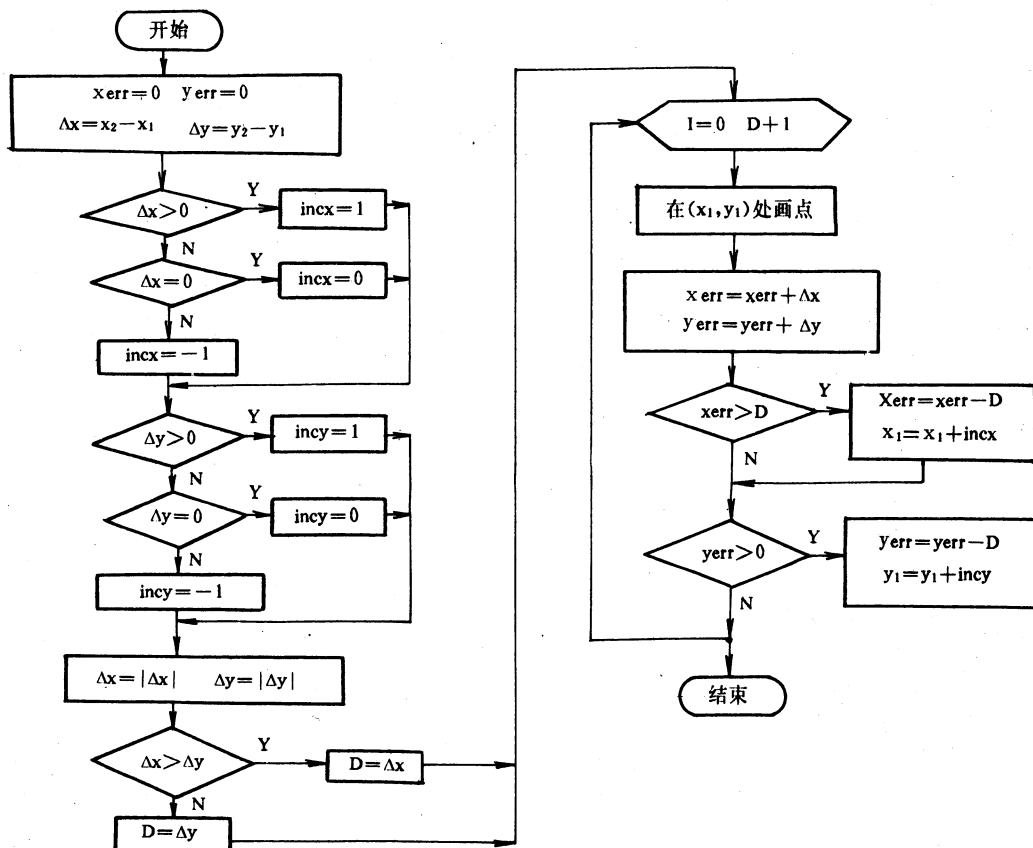


图2 Bresenham 画线算法框图



#### 四、程序设计实例

按上述的 SPT 图形文件生成方法及点和线的画法,就可以选用任一种高级语言来编制接口程序,绘制计算得到或测试得到的数据曲线,生成供 SPT 调用、编辑的图形。按文中介绍的方法绘制的图形为白色,背景为黑色,调入 SPT 后可采用 SPT 的整版反视功能将其转换为白色背景。为便于理解和使用本文所述的方法,我们用 C 语言编制了绘制点和直线的 C 函数,并给出了调用这两个函数在 640×400(宽×高)绘制范围内画一条两端点坐标为(0,0),(639,399)的直线的演示程序,供参考和使用,程序清单如下:

```
/* Super-Star 与高级语言 Turbo C 接口程序 */
#include<stdio.h>
char spt[32000];
int byte_in_line;
main () {
int height=400,width=640;
int total_byte;
char z=0x1a;
FILE * fp;
int i;
int x1,y1,x2,y2;
char title[]="Super-Star File";
byte_in_line=width/8;
total_byte=byte_in_line * height;
fp=fopen("demo.spt","wb");
fprintf(fp,"%s",title);
fprintf(fp,"%c",z);
for(i=0;i<18;i++) fprintf(fp,"%c",0);
fwrite(&width,2,1,fp);
fwrite(&height,2,1,fp);
for(i=0;i<26;i++) fprintf(fp,"%c",0);
drawline(0,0,639,399);
for(i=0;i<total_byte;i++) fprintf(fp,"%c",spt[i]);
fclose(fp);
}
/* 写点函数 */
setpoint (int x,int y)
{
char bitm[8][2]={254,1,253,2,251,4,247,8,
                239,16,223,32,191,64,127,128}
```

```
int byte,bit;
/* 全局变量:char spt[],int byte_in_line; */
byte=y * byte_in_line+x/8;
bit=7-x%8;
spt[byte]=(spt[byte]&bitm[bit][0])+bitm[bit][1];
return;
}
/* 画线函数 */
drawline(startx,starty,endx,endy)
int startx,starty,endx,endy;
{
register int t,distance;
int xerr=0,yerr=0,delta_x,delta_y;
int incx,incy;
delta_x=endx-startx;
delta_y=endy-starty;
if(delta_x>0) incx=1;
else if (delta_x==0) incx=0;
else incx=-1;
if (delta_y>0) incy=1;
else if (delta_y==0) incy=0;
else incy=-1;
delta_x=abs(delta_x);
delta_y=abs(delta_y);
if(delta_x>delta_y) distance=delta_x;
else distance=delta_y;
for(t=0;t<=distance;t++)
{
setpoint(startx,starty);
xerr+=delta_x;
yerr+=delta_y;
if(xerr>distance) {
xerr-=distance;
startx+=incx;
}
if (yerr>distance) {
yerr-=distance;
starty+=incy;
}
}
return;
}
```

## 新世纪/XqR 病毒的检测与清除

安徽财贸学院(233041) 车光宏

最近,被 SCAN85 命名为“Azusa”的 XqR 病毒(又称新世纪病毒)象幽灵似地经常出来捣乱,屡杀不绝。为了弄清原因,笔者对其进行解剖,搞清了此病毒的运行机理,对症下药,编出清除此病毒的汇编程序。

### 一、XqR 的特征

XqR 为一恶性病毒。据此病毒中的信息来看,可能是 91 年 5 月从长沙出笼的。CPAV、TNT 等清病毒软件均不能发现。SCAN85 虽可以查出并清除驻硬盘自举记录(分区表)中的 XqR,但不能发现文件中的 XqR。因此更具危险性——人们会误以为此病毒已被

清除而放松警惕,招致损失。

XqR 兼有文件型病毒和引导型病毒的特征,但主要表现为文件型的。对文件,它感染通过 DOS 中断的 4BH 号功能装入并执行的 EXE 和 COM 文件(不重复感染)。对硬盘,若 DOS 有 6 个以上的隐藏扇区,则它就感染自举记录,(这时此病毒表现为引导型特征)。对于软盘或隐藏的扇区数小于 6 的硬盘,它不感染自举记录。

感染上 XqR 的 COM 文件,其长度均增加 3K (3072),EXE 文件的长度增加 3073 到 3105 不等。被感染文件的日期和时间均不改变。系统染上 XqR 后,若在星期日(系统日期)工作,则常会出现 Ctrl、Capslock 和 Numlock 三键同时有效(如果此时正在对打开的文件操作,将导致丢失数据)。重新启动系统后,一切恢复正常,但不久又会出现上述情况。若在 5 月 4 日(系统日期)工作,XqR 删除试图在 DOS 下运行的任何文件,并在红色背景上显示如下信息:

XqR:

Wherever,I love you Forever and ever!

The beautiful memory for ours in that summer time has been recorded in the computer history.

Bon voyage,My dear XqR!

Yours 05121991 in our Home

此时如果打的是批处理命令,则该批处理文件中所列的各文件均将被删除!损失惨重。若系统日期不满足上述条件,则 XqR 只伺机传染,不出来捣乱和破坏。

## 二、XqR 的运行机理

此病毒的炮制者声称“No one can beyond me”(没有人能超过我),不免有点“夜郎”。但就此病毒程序的自我保护机制来说倒也确有其特点。

XqR 进驻内存的方式有二:其一是在系统从硬盘启动时,驻硬盘自举记录的病毒引导程序将整个病毒程序安装于内存高端。其二是在运行带毒文件时,XqR 通过 27H 号中断将自己驻留内存。

XqR 驻留内存后,通过拦截 BIOS 的 13H 中断,08H 号中断和 DOS 的 21H 中断来激活病毒程序的三个相对独立的程序段之一。13H 中断激活的程序段,其任务是保护自己不被别的程序覆盖及将自举记录送给需要的程序。08H 中断激活的是另一段自我保护程序,该段程序的任务是保证使 DOS 中断向量指向 XqR 的传染、捣乱与破坏程序。21H 中断激活的是 XqR 的传染、捣乱与破坏模块。

上述三程序段,如果分开来看,都没什么高招,但若联系起来看,此病毒确实很精,精就精在各段相互相对独立,但彼此照应,相互保护(保护各段程序的激活条件不被破坏),环环相扣。在此病毒活动的情况下,确实很难攻破。这大概就是其炮制者自感得意之处。

## 三、XqR 的检测与清除

### 1. XqR 的检测

当系统出现第一段所描述的征兆时,即可基本肯定是染上了 XqR 病毒。为了确诊,可进一步检查硬盘自举记录,若在偏移量 00EBH 处的两字节内容为 1905,则说明硬盘自举记录实为 XqR 的引导程序。对于文件,被感染的 COM 文件,其开始的第 4、5 两字节的内容为 6454;EXE 文件开始的第 1AH,1BH 两字节的内容为 1205。病毒就是根据这些值来确定是否感染文件或有 6 个以上隐藏扇区的硬盘自举记录的。据此可对硬盘自举记录和文件进行免疫。

### 2. XqR 病毒的清除

清除 XqR 病毒的步骤如下:

用干净的系统盘启动系统。将程序 1 和程序 2 分别汇编、连接并转换成 COM 文件。执行一次程序 1 可清除硬盘自举记录中的 XqR 引导程序。在根目录和各子目录下分别执行一下程序 2,以清除 COM 和 EXE 文件中的 XqR。(注意:程序 2 不能清除隐藏的 COM 和 EXE 文件中的病毒,如某子目录下有这类文件,应先将其属性改为一般文件,然后再消毒。)

程序 1:

```
;Kill XqR virus in Partition table
code segment
    assume cs:code,ds:code,es:
        code
        org 100h
begin: jmp start
buf db 200h dup(0)
msg1 db 'Not found XqR virus in Parti-
        tion table.',0ah,0dh,'$'
msg2 db 'Found XqR virus in
        Partition table!',0ah,0dh
        db 'Virus removed.',0ah,
        0dh,'$'
start: mov bx,offset buf
        mov ax,201h
        mov cx,1h
```

```
mov dx,80h
int 13h
mov di,offset buf+00ebh
mov ax,[di]
cmp ax,0519h
jnz L1
mov ax,201h
mov cx,2h
int 13h
mov ax,301h
mov cx,1h
int 13h
mov dx,offset msg2
jmp L2
L1: mov dx,offset msg1
L2: mov ah,09h
int 21h
```

```
int 20h
code ends
END begin
```

程序 2:

```
;Kill XqR virus in file
code segment
    assume cs:code,ds:code,es:
        code
        org 100h
begin: jmp start
id_file dw ?
id_virus dw ?
findfile db '*.*',00
dispfile db 0fh dup(0)
buf db 100h dup(0)
dta db 2bh dup(0)
newline db 0ah,0dh,'$'
```



```

msg1 db 'Open ERROR ! $' int 21h ERR1: mov dx,offset msg7
msg2 db 'Read ERROR ! $' jnc L7 jmp ERR
msg3 db 'Not found XqR virus $' mov dx,offset msg1 LL1: cmp ax,0100h
msg4 db 'Found XqR virus ! $' mov ah,09h jnz ERR1
msg5 db 'ERROR! Cannot remove XqR virus ! $' int 21h mov ax,4200h
msg6 db '.....Removed ! $' comexe: mov ah,09h xor cx,cx
msg7 db 'Read/Write/Lseek ERROR! $' mov dx,offset newline xor dx,dx
start: mov dx,offset dta int 21h int 21h
mov ah,1ah L7: mov bx,ax int 21h
mov dx,offset findfile mov dx,offset buf
mov ah,4eh mov ah,3fh
mov cx,20h
mov cx,0000h int 21h
findnxt: int 21h jnc L8
jnc L1 mov dx,offset msg2
int 20h mov ah,09h
int 21h
L1: mov si,offset dta+1eh tonext: mov ah,3eh
mov di,offset dispfile int 21h
L2: lods mov ah,3eh
mov [di],al jmp comexe
inc di L8: mov cx,id-virus
or al,al mov ax,id-file
jnz L2 add ax,dx
mov al,'$' mov di,ax
mov [di],al mov ax,[di]
mov dx,offset dispfile cmp ax,cx
mov ah,09h jz L9
int 21h mov dx,offset msg3
mov di,offset dispfile mov ah,09h
mov al,'.' int 21h
L3: mov cl,[di] jmp tonext
cmp al,cl L9: mov dx,offset msg4
jz L4 mov ah,09h
inc di int 21h
jmp L3 mov si,offset dta+1ah
L4: mov al,[di+1h] mov dx,[si]
cmp al,43h,'C' mov cx,[si+02h]
jnz L5 sub dx,0c00h
mov ax,[di+2h] sbb cx,0000h
cmp ax,4d4fh,'MO' mov [si],dx
jnz comexe mov [si+02h],cx
mov id-virus,5464h mov ax,4200h
mov id-file,0003h int 21h
jmp L6 jnc L10
L5: cmp al,45h,'E' mov dx,offset msg5
jnz n-comexe ERR: mov ah,09h
mov ax,[di+2h] int 21h
cmp ax,4558h,'EX' jmp tonext
jnz comexe L10: mov dx,offset buf
mov id-file,001ah mov cx,0100h
mov id-virus,0512h mov ah,3fh
L6: mov dx,offset dispfile int 21h
mov ax,3d02h jnc LL1
code
ERR1: mov dx,offset msg7
jmp ERR
LL1: cmp ax,0100h
jnz ERR1
mov ax,4200h
xor cx,cx
xor dx,dx
int 21h
jc ERR1
mo ax,id-file
cmp ax,0003h
jnz exefile
mov dx,offset buf+86h
mov cx,05h
jmp L11
exefile: mov di,offset buf+40h
mov ax,0514h
mov [di+1ah],ax
mov ax,[di+04h]
sub ax,06h
mov [di+04h],ax
mov ax,[di+28h]
mov [di+0eh],ax
mov ax,[di+2ah]
mov [di+10h],ax
mov ax,[di+26h]
mov [di+14h],ax
mov ax,[di+24h]
mov [di+16h],ax
mov dx,di
mov cx,1eh
L11: mov ah,40h
int 21h
jc ERR1
mov dx,[si]
mov cx,[si+02h]
mov ax,4200h
int 21h
jc ERR1
xor cx,cx
mov ah,40h
int 21h
jc ERR1
mov dx,[si-02h]
mov cx,[si-04]
mov ax,5701h
int 21h
jc ERR1
mov dx,offset msg6
mov ah,09h
int 21h
jmp comexe
ends
END begin

```

# DOS 5.0 的 CONFIG.SYS 的设置

深圳大学电子系 91 计算机 郭文典

MS-DOS 中除起动必需的三个系统文件外, CONFIG.SYS 和 AUTOEXEC.BAT 亦是操作系统的两个重要文件,它们可由用户根据自己的需求直接编写。把它们设置得当,在对系统的工作效率有着明显提高的同时,亦为用户提供符合实际需要的操作环境。

本文以 MS-DOS 5.0 为例对 CONFIG.SYS 的设置作一些讨论,其它版本的 DOS 均有许多相同之处,可作参考。MS-DOS 5.0 提供的配置命令共 15 条,由 IO.SYS 进行解释,其中的 Device、DeviceHigh、Dos、Files、Buffers、Break、Stacks、Shell、Switches 命令是较常用的几个。

Device/DeviceHigh,两者均用以指定可安装设备驱动程序的名称及其参数,所不同的是 DeviceHigh 是 MS-DOS 5.0 以上版本提供的命令,使用前,系统必须存在 UMB(上端内存块,由 EMM 进行访问),且 UMB 足够大,还须在 CONFIG.SYS 中声明 DOS=UMB,则 DOS 会将此设备的代码置于 UMB,以节省有限的常规内存空间。若试图在没有装入 HIMEM.SYS 和 EMM386.EXE 设备(或其它等效设备如 QEMM386)前使用 DeviceHigh 命令,其功效等同于 Device 命令。下面是几个常用的 DOS 设备:

① HIMEM.SYS,为 XMS(eXtended Memory Specification)内存管理程序,版本号为 2.0,只可用于 286 以上机型。它可将常规内存以上的 Extended memory(以下简称 EXT)变为 XMS,并可启用 A20 控制线,使 DOS 能访问 HMA(高端内存块,1024~1088KB 之间),打破了 1 兆位的限制。XMS 是一种较新的内存规范,其速度较仿真 EMS 快,已有很多的软件能够支持,但一些较老的软件使用 INT 15 来申请 EXT,它们并不支持 XMS(如金山汉字 5.0),往往需要设定/INT15=n 来规定可用 INT15 申请的 EXT 总量,让这些软件亦能使用 EXT,以增大可用常规内存的数目。例如:Device=HIMEM.SYS/INT15=360,这样,金山 5.0 可将其字库放进 EXT,从而节省了 200 多 KB 的常规内存。

② EMM386.EXE 是仿真 EMS 的设备,符合 EMS 4.0,仅能在 386 以上机型使用,它可完成两种功能:一是仿真 EMS;二是使用 UMB。它必须后于 HIMEM.SYS 装入。许多软件都支持 EMS3.2 或 4.0,使用仿真 EMS 能提供更多的内存给它们,从而提高其效率。EMM386 的第二种功能—使用 UMB,能使程序使用 640KB 到 1MB 间的上端内存块(就如使用常规内存一样),这对于设备驱动程序及驻留程序来说是非常有益的,用户可不必担心这些程序占用过多的常规内存而致使另外的软件不能正常动作。正确设置 X

和 I 参数(参看 DOS 手册)能增加 UMB。RAM 参数指定 EMM386 能对 EMS 和 UMB 进行访问;而 NOEMS 参数则指定对 EMS 的访问无效,只能访问 UMB,虽然后者将失去对 EMS 的支持,但 UMB 的数目将增至 159~192KB,相当于可用内存为 799~832KB;用户可从中选择,本文建议参数:Device=EMM386.EXE NOEMS。

③ SETVER.EXE,是 MS-DOS 5.0 以上版本为提高其兼容性而开发的设备驱动程序。某些软件虽然实质上与 MS-DOS 5.0 并无冲突,但它们在运行时首先检验 DOS 的版本号,若非特定值(如 3.30)则退出。用户此时可在提示符下键入“SETVER<程序全名><版本号>”,并在 CONFIG.SYS 中加上“Device(High)=SETVER.EXE”,然后重新启动便可解决这类问题。当然,若软件与 MS-DOS 5.0 存在冲突则此法行不通。例:SETVER GAMETOOL.EXE 3.30。

④ 在 CONFIG.SYS 中设置一个虚拟盘(RAM-DRIVE.SYS)或高速缓冲(SMARTDRV.SYS)都是不错的主意,这里就不细论。

DOS=High | LOW 及 DOS=UMB | NOUMB,前者指示 DOS 是否将其大部分代码(约 50KB)放于 HMA,这样可节省常规内存,但必须先安装好 XMS 驱动程序。由于 HMA 只能由一个程序独占,其它程序将无法再使用 HMA。后者指定 DOS 是否能使用 UMB(必须先安装好 EMM)。例:DOS=High,UMB。

Buffers=m,n,该命令用以指定缓冲区及提前读缓冲区的数目。如果系统正在使用高速缓冲软件(如 SMARTDRV 或 NCACHE 等),则应申明 Buffers 数目,如 Buffers=5,0;若没有使用高速缓冲则应根据硬盘的大小来设置 Buffers 数目:40MB 硬盘的 m 为 20、80MB 的 m 为 30、120MB 的 m 为 40 等。Buffers 太小会降低系统速度,太大则不但不会提高系统速度,而且会浪费内存。如已设定 DOS=High,则 DOS 会将部分的缓冲区置于 HMA 中。例:Buffers=30,8。

Files=m,指定同时打开指针文件的最大数目。若同时打开的文件数目超过 m,DOS 会报告错误。在一些综合的软件包(如 CAD、数据库和文字处理等)中,需同时处理较多的文件,则 m 的数目应设得较大,以减少程序因频繁打开及关闭文件而浪费的时间。例:Files=30。

Break,一般情况下,没必要设置 Break=ON,此举会增加 DOS 检查 Ctrl-Break 所开销的时间。默认值为 Break=OFF。

(转第 7 页)



# C—WordStar 使用基础问与答

新潮计算机集团公司(610051) 朱大公

## 34. 怎样使文本右缘对齐

文本右缘排列不整齐,主要有下述几种情况。首先,右缘有半个汉字(一个字符)的空位没有填满;其次,英汉混排时,紧跟在汉字后面的英文字符不能在右边界处换行;第三,应该处于行尾的标点符号,由于汉字刚好处于右边界处而使其移到了下一行。这种情况文本右缘虽然是整齐的,但不符合汉语书写的排列习惯,必须重新处理;第四,在极少数情况下可能出现右缘有多余半个汉字的空格。对于第一种情况,可以在本行中某一标点符号后多留一空格;对于第二种情况,可以在文本右边界处重新构造段落。按照英语的书写习惯,同一单词被分成两部份分别写在一行末和下一行首时,应按音节分行,并在行末加连字符;对于第三种情况,先按照合并段落的方法把标点符号移至上一行行尾,然后以标点符号为该行最后一个字符重新构造段落,并对新构造的段落用`^B`重新排版。此时我们看到标点符号超越了右边界,但这在正式出版物中也是允许的。如果不允许这样作,可以压缩一行上的空格,使标点符号不超越右边界便行了,此时右缘便是完全对齐的了。第四种情况大多是由于使用`^B`命令不当造成的,对`^B`命令的使用在此就不再加以说明了。

使用上述方法使文本右缘对齐,效果很好,但建议操作者注意以下几点:

首先,文中的标点符号原则上以使用键盘上的标点符号为好。使用汉字字库中的标点符号,在屏幕上看上去好像标点符号只占用半个汉字的位置,但实际上标点符号后的空格是绝对不允许删去的,如果删去,显示的文本立即出错。要知道,我们常借助于删除标点符号后的空格来实现文本右缘对齐。我们推荐,凡英语和汉语中都使用的标点符号,如逗号、感叹号等一律使用键盘上相应的符号,汉语中使用而英文中没有的标点符号,如句号、顿号、书名号等,使用汉字字库中的标点符号。这样就为压缩行中的空格留下了余地。

第二,右缘已经对齐的段落不要再使用`^B`命令。

第三,推荐在输入文本时凡遇自然段开始先输入4个空格,这样会为今后对齐右边界带来不少方便。

最后,设置边界时,保证左、右边界字符数之差为偶数;输入文本时,每一标点符号占两个字符的位置。

从上面的介绍我们体会到,为使文本右缘对齐,主要涉及增减空格、分割段落、合并段落、重排段落等操作。初学者可多作这方面的练习,以提高综合应用C—WordStar各种命令的能力。

## 35. 何谓点命令? 使用点命令应注意些什么?

点命令是以句点“.”打头,后跟两个字符,再跟一

参数的命令。句点“.”后所跟的两个字符应符合C—WordStar的规定,不是任意什么字符都行。参数比较灵活,经常是一整数或一串字符。点命令的作用主要是用于设计打印版面,间或也用于注释。

使用点命令应注意以下几点:

第一,点命令总是放在文本中,并占据一文本行,形成一个段落。也就是说,点命令所在行的上一行以硬回车结束,点命令所在的行也以硬回车结束。

第二,点命令必须从一行的最左列开始存放。就是说,句点“.”必须占据文本行第一列。

第三,当使用`^OL`命令设置文本的左边界不在第一列时,C—WordStar自动将键入的点命令放在第一列,但使用`^B`命令后,点命令将偏离第一列,此时必须将其放回原位,否则点命令不但不起作用,反而会当成文本打印出来。

另外,还要说明几点:

第一,点命令虽然出现在文本中并占据一文本行,但它并不会打印在打印纸上。

第二,如果点命令使用错误,在标志列会看到一个问号“?”。这种错误是指点命令不完善,或无效的点命令。

第三,状态行显示的行号是点命令下面一行的行号,而列号是点命令所处的实际列号。

## 36. 何谓顶部边界? 何谓页首标题边界? 怎样规定和使用这两种边界?

顶部边界是指一页打印纸的顶部距打印文本第一行间的间隔,以行数为单位计算,约定值为3行。这里所说的打印纸顶部实际上是指打印头停留的当前位置,并不是打印纸的自然撕纸线。规定顶部边界使用点命令。`MTn`, $n$ 是用户想要留出的顶部边界行数。

页首标题边界是指页首标题到打印文本第一行的间隔,以行数为单位计算,约定值为2行。要注意的是,页首标题所在的行是计算在顶部边界内的。规定页首标题边界使用点命令。`HMn`, $n$ 为用户想要留出的页首标题边界行数。如果要使用页首标题,则应配合使用`HE`命令,页首标题的内容以字符串的形式紧跟在`HE`命令后。如不使用`HE`命令,则约定页首标题所在的行为空白。由此可见,。`HE`命令后的字符串是页首标题内容,。`HMn`命令中的 $n$ 决定了页首标题放在顶部边界内的具体位置。值得说明的是,一旦使用了带字符串的`HE`命令,则每一页上都会打印出页首标题,想取消页首标题可使用一次不带字符串的`HE`命令。

使用`MTn`和`HMn`命令后,屏幕上的文本并无什么变化,打印输出时便可以清楚地看出这两个命令

的作用。例如，我们使用了.MT5和.HM1两个命令，则在文本打印输出时，打印机首先进纸3个空行（以打印头的当前位置为起点计算），然后打印页首标题（如果使用.HE命令指定了页首标题），接着再进纸一空行，最后开始打印编辑好的文本。有人会提出这样的问题，为什么要让打印机开始进纸三个空行呢？这是为了和规定底部边界的命令.MBn、规定页长的命令.PLn配合使用以调整打印输出文本在一页打印纸上占据的实际长度。

### 37. 何谓底部边界？何谓页末标题边界？怎样规定和使用这两种边界？

底部边界是指打印文本的最后一行至打印纸底部间的间隔，以行数为单位计算，约定值为8行。这里所说的打印纸底部是指打印完一页后打印头停留的实际位置，而不是打印纸的自然撕纸线。规定底部边界使用的点命令是.MBn，其中n为所希望的底部边界行数。

页末标题边界是指打印文本的最后一行与页末标题之间的间隔，以行数为单位计算，约定值为2行。注意，页末标题所在的行是计算在底部边界行数内的。规定页末标题边界的点命令是.FMn，其中n为页末标题边界行数。与页首标题的约定状态不同，如果不使用指定页末标题的点命令，则在.FMn命令指定的行上居中打印页编号，页编号从1开始，每页递增1。如果想改变页末标题的内容可以使用.FO命令，页末标题内容以字符串的形式紧跟在.FO命令后。如果使用不带字符串的.FO命令，表示取消原来定义的页末标题，恢复页末标题为居中打印页编号的约定状态。

### 38. PLn命令是控制页长的点命令，它为什么能控制一页中打印文本的行数呢？

首先应搞清楚何谓页长，所谓页长是指一页打印纸能打印的文本行数。那么，什么又是一页打印纸呢？初学者往往以打印纸的自然撕纸线为标准来定义页，这种定义只在特定条件下才成立。实际上，页应该按.PLn中的n来定义。例如，使用了.PL10命令，那么10行就是一页，而页长也就是10行。仅当顶部边界、底部边界和页长都取约定值，且不改变行间距的情况下，才可以用自然撕纸线来定义页。另外要注意的一个概念是，如果页长为10行，打印输出时并不一定在一页上打印10行文本，因为页长是由三部份组成的，一部份为顶部边界行数，一部份为一页中实际打印的文本行数，再一部份为底部边界的行数，三部份之和才是页长。例如，我们使用了三个点命令.MTn1、MBn2和.PLn，那么在一页中实际打印出来的文本行数为 $(n-1-n2)$ 。如果只使用.PLn命令而不使用.MT和.MB命令，那么n1和n2便取约定值，一页中可打印的文本行数应为 $(n-11)$ 。由此可见，.PLn命令仅仅是决定一页中实际打印的文本行数的因素之一。

使用.MTn、.MBn和.PLn三个点命令要注意两点。一、如果要使用.MTn和.MBn命令，则一定要放在.PLn命令之前，如果把它们放在.PLn命令之后，它们所设定的值不起作用而取约定值。二、.MTn和.MBn

二者之间的先后位置可以任意，但任何一个均应放在.PLn前。二、任何时候都应保证页长大于顶部边界和底部边界之和。一个特例是顶部边界和底部边界均取约定值，则页长最小取值为12。

### 39. 下面三页在C-WordStar下打印出来的文本，请问轮流居于左、右的页编号是怎样打印出来的？

```

XXXXXXXXXX  XXXXXXXXXXXX
XXXXXXXXXX
XX.....  XXX.....  XX.....
:
XXX.    XXX.    XXXX
.1.      .2.      .3.

```

首先应该明确，页编号.1、.2和.3等属于页末标题。当把确定页末标题内容的.FO命令和^PK命令联合使用时便可以得到题中所给出的结果。具体的作法是，在已编辑好的文本首部加入点命令.FO^PK: #.”。其中，.FO表示打印页末标题；#号表示页末标题的内容为顺序编号页；#号两边的句点“.”表明它是加在页编号两边的字符串；^PK命令表示页末标题轮流出现在每页的左边和右边；^PK和.#.间留出适当的空格表示.1.出现在页末右边，.2.出现在页末左边，余类推，如果省去^PK和.#.间的空格，则.1.出现在页末左边，.2.出现在页末右边，余类推。如果^PK和.HE联合使用，则可以出现轮流居于左右的页首标题。要想控制.1..2.等页编号和最后一行打印文本的间距可使用.FMn命令。

### 40. Y命令的作用是删除文件，为什么还要设置另一个删除文件的命令^KJ？类似的命令还有哪些？

任何一种软件如果想挤进优秀软件之列，就应该具备在自身的运行环境下完成若干常用的文件管理功能。删除文件是文件管理中必不可少的功能之一。C-WordStar在其起始菜单下安排了Y命令，不必返回DOS便可以完成文件删除的操作。用户希望在编辑状态下删除某些不需要的文件以让出更多的磁盘空间为当前文件使用，最方便的方法就是在编辑状态下也设置一个命令来完成删除文件的操作。如果不作这样的安排而要用户返回起始菜单显然是不明智的。正是基于这一考虑C-WordStar安排了在编辑状态下使用的文件删除命令^KJ。

与Y命令和^KJ命令类似，C-WordStar还设置了复制文件的O命令和^KO命令、文件更名的E命令和^KO命令。在这三对命令中均是单键命令在起始菜单下使用，三键命令在编辑状态下使用。

#### 小结：

C-WordStar问与答共精选40个题目，并给出了解答。这些问题是笔者教授“怎样使用C-WordStar”课程时学员提问最多、操作错误最多的一些问题。笔者感到单给学员讲清楚怎样操作是不够的，还应该尽可能地给大家讲清楚为什么要这样操作，只有这样才能

(续第27页)



# 计算机病毒的检测、消除和预防

苏民生

## 六、病毒的消除方法

消除病毒,是指复原被病毒感染的磁盘或文件。可以使用通用软件工具,更广泛使用的则是专门检测、消除病毒的软件。

### 1. 消除引导区病毒

对于侵占磁盘引导区和硬盘主引导区的系统型病毒,不但易于检测,而且易于消除。并不一定非要了解病毒的具体情况,只要有必要的备份文件,对于“未知病毒”同样可以消除。备份文件是指存放硬盘主引导区(MBR)内容的文件,存放硬盘各分区引导区(BOOT)内容的文件,以及存放各种软盘引导区(BOOT)内容的文件。

由于硬盘有多种类型,而对同一类型的硬盘又可以有不同的分区方法,因此,每个计算机系统的硬盘都有各自的分区表和各分区不同的参数表,它们的主引导扇区和引导扇区往往不能交换使用。每个系统都应有自己的备份,可以在系统正常情况下用 NU、DE-BUG 等工具软件读出这些扇区内容,再将它们写到备份软盘上作为数据文件保存,并对备份软盘加写保护。一旦发现硬盘引导区或主引导区被改变,就及时从备份盘上将原来的内容写回相应扇区中,使系统恢复正常。

软盘的类型不多,同类软盘的参数表相同,可以通用,因此,只要为每类软盘留一个有 BOOT 区内容的备份文件就够了。现在常用的有 360K 和 1.2M 软盘,是 5 1/4 英寸的,以及 720K 和 1.44M 软盘,是 3 1/2 英寸的,只需备份 4 个 BOOT 文件。也可以直接读入好的软盘的 BOOT 区内容,写到可疑软盘上,直接消除病毒。例如,目前对感染 GenB 的软盘尚无可消除病毒的专门软件,但可以很容易地用 DEBUG 消除,步骤如下:

第一步 在无病毒环境下进入 DEBUG

第二步 插入好软盘,通常是带写保护的系统盘,插入 A 驱动器

第三步 读入好盘 BOOT 区,命令为 L 100 0 0 1

第四步 将有病毒软盘插入 A 驱动器

第五步 将好的 BOOT 区内容写至可疑软盘 BOOT 区,命令是 W 100 0 0 1

如果有多个软盘需要恢复,只要重复执行第四、五两步就可完成。

切记所用“好盘”必须和要修复的软盘类型相同,因为不同类软盘的参数表不同。

### 2. 消除病毒的环境

检测和消除病毒,都要求在无病毒环境下进行。使用带写保护的系统盘启动,能保证这一点。当然,这张软盘应当未被任何病毒感染。这是一个最基本的要求,而且不难做到。但是,不少人忽略了这一要求,使得消除病毒的工作失效。在病毒环境下,对磁盘和文件进行的修复工作,或者由于病毒程序的作用无法实现,或者在修复之后由于病毒程序的作用又立即感染病毒。因此,特别要注意消除病毒时系统处于好环境中。

另一项值得特别注意之处是必须毫无遗漏地对每张软盘都进行检测和消除。在消除病毒时,硬盘是首当其冲的对象,而软盘则由于数量较多,放置分散,易于遗漏。一旦漏掉一张带病毒的软盘或文件,它就成为隐患,会在某一天由于使用它而使病毒卷土重来。这种反复在管理不严格的情况下经常发生,应设法避免。

### 3. 其它病毒的消除方法

要修复被文件型病毒感染的文件,首先必须查清楚它所感染的是那一种病毒,然后针对这种病毒对文件的感染方式进行复原。因此,对于感染文件的病毒应当有一定的了解,掌握其基本特征。对于“未知”病毒,尽管已经有监控手段可以在它出现时发觉它甚至不让病毒进入系统,但是,要恢复被“未知”病毒感染的文件则难以办到。因为还不知道病毒的最基本的特征,复原工作尚无从下手。从这个意义上讲,修复被病毒感染的文件,要比修复引导区被病毒感染的磁盘,要求的条件更多一些,做起来也更为复杂。

为修复文件所需了解的病毒特征,通常有以下几方面:病毒程序长度是多少?它接在原来文件之后还是之前?对于 COM 文件,病毒是否改变前几个字节内容?换成了什么内容?原来的内容被“藏”到哪里去了?对于 EXE 文件,病毒对于文件头做了什么改变?原来的有关参数被移到何处或如何变成新参数?病毒程序是否加密?如果是加密的,病毒程序在执行过程中如何解密?有了这些了解,才能修复文件。

对于文件型病毒,使用 DEBUG 等通用软件工具修复文件,是一件十分细致而又不烦的工作,人们更希望有专门的消除病毒软件可供使用。除了由公安部提供的软件 SCAN/KILL 之外,经常见到的还有 SCAN/CLEAN、CPAV、TNTVIRUS、HUNTUVIR 等。这些软件也在不断提供新版本。现时又出现将已有检测、消除病毒程序固化而制做的防病毒卡,可以在一定时间内起一定作用,而且使用方便;而当病毒又有新发展时,防病毒卡也会有发展的。

由于总是先有病毒然后才有检测、消除方法和工

具,检测消除工作通常要比病毒的发生、发展落后一步,因此,人们不得不经常面对一些“未知”病毒。有时虽然发现一些异常,推测存在病毒,但缺乏更多的了解,无法确定病毒种类,更无法修复文件。在这种情况下,只能借助于通用软件工具进行分析研究。常用的方法之一是选定或制做一个内容不复杂的 COM 文件,留下备份,然后放到怀疑有病毒的环境中进行磁盘读/写操作或运行这个 COM 文件,再恢复为无病毒环境,对使用后的 COM 文件进行检查,与早先留下的备份相对照,看文件有什么变化,如有变化,就可以找出附在原来文件之后或之前的病毒程序。可以采用相同办法获取感染病毒的 EXE 文件标本。在此基础上对于病毒程序可进行更详细、深入的分析,从而找到检测和消除病毒的方法,并进而编写专用程序。

对于其它病毒,如 D2 病毒,以及混合型病毒,修复被感染文件的原理和方法,和对于文件型病毒是同样的。只要病毒存在、出现,它就一定有表现,因此,也就一定能发现它,并找出检测、消除的方法,从这个意义上说,在计算机病毒面前不必过分恐慌。

#### 七、加强管理,减少病毒危害

在 IBM PC 系列微机(及兼容机)上,由于 DOS 本身的特点,很容易受到病毒攻击。在目前病毒流行、新病毒很快出现而检测消除的方法和工具总是落后一步的情况下,做好计算机系统的管理,对于能否安全地使用系统,至关重要。如果管理不善,系统就很容易受病毒感染,不但影响系统自身的正常运行,而且又成为向

外界传播病毒的传染源。反之,如果采取了必要的严格的管理,就可以有效地防止病毒侵入,对于“已知”病毒可以及时发现和消除,对于新出现的病毒,也可以较早地检测出来并采取必要措施将病毒造成的损失限制在最小范围。

通常要注意几个方面:

第一,常用软件。重要文件及工作成果,一定要有备份软盘,并且加写保护。正在进行中的工作,也应将工作结果及时复制到软盘上,以免发生事故时丢失。

第二,对于外来软盘,包括属于自己但在其它系统使用过的软盘,都必须先检查后使用,在确认它是“干净的”之后,才能放心大胆使用。对于由电脑厂商来的软件,同样需要进行检查。这是一件很烦琐而且使一些人厌烦的事,但为了确保安全,却是必须做的。

第三,定期检查系统。使用较多的系统,往往每天都需要检测。如发现可疑情况,应随时检查。

第四,采取必要的防范和监控措施,使用软件(如 CPAV 等)或防病毒卡。

第五,建立必要的管理制度,设立专人负责管理。现在还难以做到一部机器只有一个人固定使用,而对许多人都用的机器,如不规定管理人员和管理办法,则一切防范措施等同于虚设。

在必要的管理之下,病毒的传播会受到相当大的遏止,其危害也可以压制到最小,使得系统能正常工作,发挥其应有的作用。

## 中国地质大学(北京)1994年成人学历教育招生

### 电子计算机应用专业

**报考条件:**具有高中毕业文化程度或同等学力者均可报名。

**报名与考试:**三月初到当地成人高等教育招生办公室登记报名,参加全国统一的成人高考。

**考试时间:**5月份的第二个周六、周日两天。

**考试科目:**数学、物理、化学、语文、政治。

**录取:**按国家教委规定的录取原则及所在省确定的最低控制线录取。具有大学专科以上学历者,可申请

免试入学。

**学习形式:**函授及夜大学;学制三年,全国各地招生。

**培养目标:**达到程序员级水平的计算机应用高级人才。

**主要课程:**电子技术基础,计算机基础知识,操作系统的使用,高级语言程序设计,数据库应用,数据结构,网络使用,软件工程等。

**通讯地址:**北京市海淀区学院路29号中国地质大学(北京)成人教育部 电话:2014874 邮编:100083



# 把 & 和 CALL 语句“嫁接”到赋值语句中去

重庆市九龙坡区重庆铁路中学(630053) 邹习新

在 BASIC 程序中用到 & 和 CALL 语句时,往往因其不能直接带参数(特别是字符型参数)而深感不便。虽有经验介绍利用存于零页的 CHRGET 子程序来接收处理字符串,也往往限于字符串常数,而对字符串变量、函数、表达式还是一筹莫展。另一方面,赋值语句虽然具有极强的处理表达式的能力,可以将包括字符串在内的表达式的值赋予某个变量,但要在机器语言程序中去找到这个变量也是颇费功夫的。

于是,我有了一个大胆的想法:把 & 和 CALL 语句“嫁接”到赋值语句中去,利用 BASIC 解释程序的现成资源来解决这个问题,使用形如

&(参数表达式)

或

CALL(入口地址表达式)(参数表达式)

的语句,让赋值语句不把表达式的值赋给某个变量而直接存入我们指定的地址。

为此,我解读了有关的程序。原来 BASIC 解释执行赋值语句时,首先要查找赋值号左边的变量,找到后即将其地址置入零页的 \$85 和 \$86 两单元,将变量类型标志置于零页的 \$11 和 \$12 两单元(具体内容见附表),然后进入 \$DA52 进行处理。处理完毕返回时,存放结果的字节内容因变量类型而异,即:对于字符串共 3 字节,其第一字节为字符串长度,第二、三字节为存放字符串的首地址(低位在前);对于整型数共 2 字节,直接存放数值(高位在前);而对于实型数则需 5 字节,其第一字节为价码,其余 4 个字节为尾数。

因此,我们只要在机器语言程序中根据需要,按附表要求将表达式数型标志置入 \$11、\$12 两单元,将欲存放结果的首地址置入 \$85、\$86 两单元,就可以用 JSR \$DA52 来求得表达式的结果了。当然,若把上述内容重复几次,就可求得几个表达式的值了。不过要注意一点,当语句中有两个以上的表达式又没有合适的分隔符号时,须防止出现语法错误,此时可用一对对圆括号将前面的表达式括起来。

作为例子,下面给出在中文状态屏幕第十一行循环显示字符串的程序。机器语言部分(功能为在光标处显示一串字符,每行显示多达 35 个西文字符):

```
0300- A9 FF LDA # $FF
0302- 85 11 STA $11
0304- A9 1A LDA # $1A
0306- 85 85 STA $85
0308- A9 00 LDA # $00
030A- 85 86 STA $86
```

```
030C- 85 12 STA $12
030E- 20 52 DA JSR $DA52
0311- A9 FF LDA # $FF
0313- 85 09 STA $09
0315- A5 1A LDA $1A
0317- D0 01 BNE $031A
0319- 60 RTS
031A- A0 00 LDY # $00
031C- B1 1B LDA ($1B),Y
031E- E6 1B INC $1B
0320- D0 02 BNE $0324
0322- E6 1C INC $1C
0324- C6 1A DEC $1A
0326- 20 AB C3 JSR $C3AB
0329- C9 7F CMP # $7F
032B- D0 06 BNE $0333
032D- E6 09 INC $09
032F- E6 09 INC $09
0331- D0 09 BNE $033C
0333- 24 09 BIT $09
0335- 30 05 BMI $033C
0337- C6 09 DEC $09
0339- 20 7D EF JSR $EF7D
033C- 20 0D EE JSR $EE0D
033F- 20 B9 C3 JSR $C3B9
0342- 18 CLC
0343- 90 D0 BCC $0315
```

BASIC 程序部分:

```
10 PRINT:PRINT CHR$(4);"PR#3":PRINT:HOME;
LOMEM:24576:PRINT CHR$(18):C%=768
20 A$=""在汉字符幕的第11行进行中西文混合显示,按
任意键一暂停,按(ESC)键一结束。
25 B$=""可以在屏幕的第":C$=""行 显示35个西文字
符"
30 FOR H=34 TO 2 STEP-1:GOSUB 60:NEXT
40 H=1:GOSUB 60:X=1:IF ASC(A$)=127 THEN H
=0:GOSUB 60:X=3
50 A$=RIGHT$(A$,LEN(A$)-X)+LEFT$(A$,
X):L=L-X:LX=LX-(X+1)/2:GOTO 40
60 IF LX<35-H THEN X=(MID$(A$,L+1,1)=
CHR$(127))*2+1:L=L+X:LX=LX+(X+1)/2
70 POKE 36,H:VTAB 11:CALL C% LEFT$(A$,L):
HTAB 1:CALL C%" "
80 IF V=0 THEN HOME:VTAB Y+1:HTAB 1:CALL
C% B$+MID$(" ",1,Y<9)+STR$(Y+1)+C
$:Y=(Y+1)*(Y<9)
```

```

90 V = (V + 1) * (V < 9); VTAB 7; X = PEEK(49152) -
    128; POKE 49168, 0; IF X < 0 THEN RETURN
100 IF X <> 27 THEN GET X $; IF X $ <> CHR $ (27)
    THEN RETURN
110 PRINT; PRINT CHR $ (4); "PR # 3"; PRINT; END

```

注: 20 行中 A \$ 末尾的空格为每次循环显示之间的间隔, 可任意确定; 70 行在屏幕 11 行显示字符, 为字符串函数及常数的应用; 80 行在屏幕的各行显示字符, 为字符串表达式的应用; 另外若在程序的开头部分加入了 POKE 1014, 0 和

POKE 1015, 3 的语句, 则所有的 CALL C% 均可用 & 代替。(附表 表中数据均为十六进制)。

表达式类型	\$ 11 内容	\$ 12 内容
字符串	\$ 80 ~ \$ FF	\$ 00 ~ \$ 7F
整型数	\$ 00 ~ \$ 7F	\$ 80 ~ \$ FF
实型数	\$ 00 ~ \$ 7F	\$ 00 ~ \$ 7F

## 电子工业出版社迁址启事

我社于一九九三年十月二十日迁至新址办公。

地 址: 北京市海淀区普惠北里 12 号  
 邮政信箱: 北京 173 信箱      部 门 电 话  
 邮 编: 100036                  传达室: 8231376  
 传 真: 8214062                  办公室: 8214062  
 电 挂: 北京 3101                总编办: 8210832  
 交 通: 乘 335 路公共汽      行政处: 8222466  
 车在罗道庄站下车, 南行约      出 版: 8235245  
 200 米向东; 乘 323 路公共      发 行: 8214070  
 汽车在八一湖站下车, 下桥  
 西行。





# ProDOS 系统内部结构剖析(续)

北京铁路局卫生防疫站(100038) 廖 凯

注意:Apple II 监控程序依赖于一个中断发生时被改写的零页单元 \$ 45。监控命令使用这单元作为临时存储器,若此单元在执行中被改变将会发生不可预测的事情。在你使用监控程序期间,应该禁止中断,此外,系统使用 \$ 7F8 单元存储中断发生前所使用的 I/O 槽口的位置,用户不可使用此单元。

## 1. 在 MLI 调用期间发生中断

此部分将讨论在一个中断发生在 MLI 调用执行期间程序应做什么,和你的中断处理例程自身如何调用 MLI。

简单的作法是,程序应该允许 MLI 在启动一个新调用之前完成当前的调用。其机理是修改全局变量,使 MLI 完成它的调用后返回到你的例程,而不是最初调用它的例程。你的程序可以按需要使用 MLI。当完成时,它必须恢复 6502 寄存器为原状态,然后返回到原例程内相应的地址。

要做到这点,中断处理例程应该首先检查 MLI 的状态。若标志 MLIACTV(\$ BF9B)的高位被设置,则 MLI 是在一个调用当中,你的程序应该这样做:

①保存原调用的返回地址(CMDADR, \$ BF9C),而以当前的调用完成时,MLI 应该返回的地址代替它。

②关闭硬件中断,并清除进位标志。

③RTS

MLI 的中断处理程序认为中断已被处理,于是它结束当前的 MLI 调用,并返回到 CMDADR 指定的你的程序的地址。你的程序现在应这样做:

④对于刚结束调用的例程,保存 A, X, Y 和 P 寄存器作为返回状态。

⑤按需要使用 MLI。

⑥恢复 A, X, Y 和 P 寄存器。

⑦转到最初的 CMDADR。

这原程序只看到它的调用已成功地完成,并继续执行。

## 2. 中断处理程序实例

这是一个采样子程序,它从一个 Thunderclock 卡上读取日期,并在屏幕的右上角以秒为单位显示它,它假定卡在槽口 2 上。

```

ORG $ 300
WTTCP EQU $ C20B ;时钟写入口(SLOT2)
RDTCP EQU $ C208 ;时钟读入口(SLOT2)
TCICR EQU $ C080 ;中断控制寄存器(SLOT2)
TCMR EQU $ C088 ;中断复位寄存器(SLOT2)
    
```

```

IN EQU $ 200 ;时钟存储时间单元
UPRIGHT EQU $ 412 ;屏幕右上角
INTON1 EQU $ 47A ;使中断接通(SLOT2)
INTON2 EQU $ 7FA ;使中断接通(SLOT2)
MLI EQU $ BF00 ;MLI 入口地址
* 调用中断
JSR ALLOC.INT ;MLI 安装中断程序
RTS ;结束
SHOWTIME EQU *
CLD
PHP
SEI ;禁止中断。
LDY # $ 20 ;对于 SLOT2
LDA TCICR, Y ;取得中断控制寄存器的值
AND # $ 20 ;检查位 5,是否从时钟卡中断
BEQ NOTCLK ;位 5 为零则非从时钟卡中断
LDA TCMR, Y ;清除中断复位寄存器
LDA TCICR, Y ;在硬件上清除中断
DEC COUNTER ;按每秒显示时间
BNE EXITCLK ;要显示时间尚水到
LDX # 39 ;保存输入缓冲区,因为读
DOIN LDA IN, X ;时钟卡时它将
STA INBUF, X ;被重写。
DEX
BPL DOIN
LDA # $ A5 ;设置 APPLESOFT 串输入
JSR WTTCP ;模式发送它到卡
JSR RDTCP ;读时间到输入缓冲区内
LDX # 21
GETNEXT LDA IN+1, X ;在屏幕上显示时间
STA UPRIGHT, X ;输入缓冲区的 0-22 字符
DEX
BPL GETNEXT
SETCNTR LDA # 64 ;为下一次调用设置
STA COUNTER ;计数器
LDX # 39 ;恢复输入
DOIN2 LDA INBUF, X ;缓冲区
STA IN, X
DEX
BPL DOIN2
EXITCLK PLP
CLC ;通知 MLI 中
RTS ;断已处理完
NOTCLK PLP
SEC ;通知 MLI 不是
RTS ;时钟卡中断
COUNTER DS 1, 0
    
```

```

AIPARMS  DFB 2,0      ;放置中断参数
          DW SHOWTIME ;在这儿
DIPARMS  DFB ,10     ;使两个例程可以使用它们
INBUF    DS 40,0     ;保留 40 字节为输入缓冲
          ;缓冲区保存/恢复之用

```

注意这个程序的重要特征:

①以一个 CLD 指令开始。

②查看是否由时钟卡引起的 IRQ 中断,若不是则设置进位返回。

③当它完成中断处理时,清除进位返回。

下面的程序使用 ALLOC\_INTERRUPT 调用添加中断例程。处理完后它在时钟卡上启动中断。

```

ALLOC_INT JSR MLI      ;调用 MLI 分配中断
          DFB $40
          DW AIPARMS
          BNE OOPS     ;错误时停止
          LDY # $20
          LDA # $AC ;写一个“,”到时钟卡
          JSR WTTCP ;设置 64Hz 中断频率
          LDA # $40 ;现在启动软件
          STA INTON1 ;并通知它在读后
          STA INTON2 ;不禁止中断
          STA TCICR,Y
          LDA #1      ;立即显示时间,以
          STA COUNTS ;每秒为单位
          CLI         ;允许 6502 接受中断
          RTS

```

```

OOPS      BRK         ;错误时停止

```

下面的程序在用一个 DEALLOC\_INTERRUPT 调用从 ProDOS 取消中断例程之前,先禁止时钟卡中断。

```

DEALLOC_INT LDA #0 ;禁止时钟卡中断
          STA INTON1
          STA INTON2
          LDY # $21
          STA TCICR,Y
          LDA AIPARMS+1 ;取得中断号
          STA DIPARMS+1 ;以取消中断分配
          JSR MLI      ;调用 MLI 到取消
          DFB $41     ;中断例程
          DW DIPARMS
          BNE OOPS2   ;错误时终止
          RTS         ;结束
OOPS2 BRK           ;错误时终止

```

### 三、磁盘驱动器子程序

若其它厂家提供的磁盘驱动器是用 ProDOS 工作,则它必须与 Apple 公司提供的磁盘驱动器一样的工作性能。它的引导 ROM 必须在指定的单元有特定的内容,并且它的驱动例程必须使用指定的零页单元存放它的调用参数。

#### 1. ROM 代码规范

当自启动 ROM 从启动的槽口查找一个驱动器时,它在下面单元内寻找这些值(N 是当前正被检测的槽口号):

```

CN01 = $20  这些值必须是在对应于包含磁盘
CN03 = $00  控制卡槽口的 ROM 内
CN05 = $03
CN07 = $3C

```

在自启动 ROM 已成功地找到包含这些字节的一个槽口时,它 JMP 到 \$CN00 单元。这些功能由后面描述的 ROM 代码所完成。

\$CNFC 和 \$CNFD 单元必须包含在驱动器内所使用的卷的总块数,低字节在前。

\$CNFE 单元是一个状态字节,它定义设备的特性,它的位是:

```

BIT 7: 记录媒体是可移动的。
BIT 6: 设备是可中断的。
BIT5-4: 在设备(0-2)上卷的数量。
BIT 3: 设备支持格式化。
BIT 2: 设备可以被写入。
BIT 1: 设备可以被读取。
BIT 0: 设备的状态可以被读取。

```

在某个位被设置时,相应特性有效。

\$CNFF 单元应该包含 ROM 的入口地址的低字节。例如,若 \$CNFF 包含 xx,对于那个设备的入口地址将是 \$CNxx。这地址在启动时由引导程序放置到系统整体页面内的设备驱动程序向量内。

#### 2. 调用参数

在 ProDOS 调用磁盘驱动例程之前,它用调用的参数设置 6 个零页单元,这些单元和含义是:

```

$42: 指令  0=请求状态
          1=请求读
          2=请求写
          3=请求格式化

```

```

$43: 设备号  7  6  5  4  3  2  1  0

```

Dr	Slot	未	用
----	------	---	---

\$44—\$45: 缓冲区指针:指明要传送数据的一 512 字节内存缓冲区的首址。

\$46—\$47: 块号:指出要传送数据的磁盘上的块。

当调用时,驱动例程必须完成由这些参数指定的操作。若调用完成,驱动程序会返回并清除进位。若有一个错误,进位会被设置,错误码被放在累加器内。

设备驱动程序执行时会出现的错误是:

```

$03: 无设备连接
$04: 写保护
$08: I/O 错误

```

(待续)



## 第十二讲 D1.0 版本补遗

北京 FORTH 应用研究会(100034) 丁志伟

这个讲座主要对 D1.0 版本做了介绍,到这里应该告一个段落了。由于讲座的性质,前几讲之中出现的单词很有限。读者可能想对这个版本有一个全面的了解。FORTH 语言最主要部分是词典,把这些词介绍出来,更能使读者看清这个版本的全貌。

下面就先介绍讲座中没有涉及的部分单词,然后介绍这个版本其它方面的情况。

### 逻辑运算词

这几个逻辑运算词,都是按位运算的。

OR (n1 n2 --n3)  
或运算,逻辑和  $n3 = n1 \vee n2$

AND (n1 n2 --n3)  
与运算,逻辑乘  $n3 = n1 \wedge n2$

EOR (n1 n2 --n3)  
异或运算,按位加  $n3 = n1 \oplus n2$

NOT (n1 --n2)  
逻辑反,  $n2 = \bar{n1}$

### 堆栈词

? DUP (n -- n n) 栈顶数不为 0 时,复制栈顶数  
或(0 -- 0) 栈顶数为 0 时不变;

SP? ( -- ) 查看参数栈顶位置,显示栈顶所在地址,栈空时是 90FD。堆栈是向下增长的,地址越小说明栈中数越多。

### 标志置位和复位

下边两个词主要是置标志用。

ON (a -- ) 置位,把 a 中数置成 1

OFF (a -- ) 复位,把 a 中数置成 0

### 变量和常数

下面是词典中的变量。

DADR 其长度是 6 字节,存放磁盘 I/O 矢量。

DLD DLOAD 词用到。

BASE 当前进制数基。

CHARS 16 字节长,存放 0~F 的 ASCII 码。系统输入输出数字用。

HC 多用途临时变量。可供用户使用。

下面是系统词典中的一些常数。

HT 24 光标横坐标

VT 25 光标纵坐标

AREG 6E A 累加器的保护单元

XREG 6F X 寄存器的保护单元

YREG 70 Y 寄存器的保护单元

PREG 71 P 寄存器的保护单元

以上常数指向内容都是一字节,后四个是调子程序时用的。

### 定义词

与定义有关的词有如下这些:

VAR VARIABLE CON : ; IMMD  
;I [C] NAME ;; , C,  
FLUSH (( ))

没有介绍过的有定义方法有:

: name V ; IMMD  
: name V ;I

后者是前者的简化形式。这样定义出的词,称为立即词,它即使在定义中,也会立即执行,而不会被编译进词典,是辅助编译用的,因此又称为编译词。

: name V1 ;; V2 ;

这里用了一个;;结果是定义出一个新的数据类型,具有二次定义能力。V1 是编译时执行的功能,V2 是执行时所做的工作。

例如定义一个数组定义词 ARRAY

: ARRAY 2 \* DTOP +! ;; SWAP 2 \* + ;  
就定义了名为 ARRAY 的定义词。其功能是定义一个象 VAR 那样具有定义功能的新词。

下一步可以定义数组

3 ARRAY DATA  
定义时做左部分的 2 \* DTOP +!  
定义出三个元素的数组 DATA。

执行 DATA 时,可以

2 DATA ?

DATA 先把它地址放在堆栈上,再执行

SWAP 2 \* +

这样就取出 DATA 中第 2 项内容(项数从 0 数起),然后由“?”显示此数。

下面有几个与定义过程有关的词:

[C] ( -- ) 强行编译,在冒号定义中使用它,能把立即型词编入单词中。

NAME (a -- ) 在词典中产生一个词名,同时显示这个词名,冒号定义、VAR 和 CONST 都用到它。

, (n -- ) 逗号词,把栈中数存入词典,经常在定义变量时使用。

C, (e-- ) 把栈中单字节数存入词典。  
 FLUSH ( -- ) 定义过程之后用到,让系统承认定义时对词典所做的改变。冒号定义、常数和变量定义都用到它。  
 “(”和“)”在编译过程中调用解释程序,再回到编译状态。用于临时对系统做某些改变,如改变进制制。

### 输入/输出词

LIN ( -- ) 从键盘输入一行信息,以回车结束。输入的内容存放在 200H 起始的地址之中。其起始和终址存放在 INPUT 和 INTOP 之中。  
 RCH ( --c ) 从输入流中读一个字符,如果输入流空,则调用 LIN  
 ADV ( -- ) 使 INPUT 加 1  
 ZIN ( -- ) 清除输入行

### 反汇编、反编译等系统分析工具和词典有关的词

以下这些词中,有几个曾经介绍过。但它们是这个版本中的重要工具,又能体现出 FORTH 语言的特色。为了完整,将其一起介绍。

ID. ( a-- ) 显示一个词的名字,a 是这个词的首址  
 CFA ( a1--a2 ) 从一个词的首址得出它的执行地址  
 CF. ( a-- ) 显一个词的名字,a 是其执行地址  
 LL ( a-- ) 调用监控反汇编子程序  
 FHY ( -- ) 反汇编一条指令,其地址在 3A、3B,与监控反汇编格式相比,它不显指令的机器码。反汇编一条后,3A、3B 中地址指向下条。  
 FHWD ( -- ) 反汇编一条指令。只显示指令本身,不显示指令机器码和指令所在地址。地址在 3A、3B 中。  
 DFHB ( a1 a2 -- ) 反汇编一段,地址从 a2 到 a1,以 FHY 格式显示。  
 FHY、FHWD、DFHB 这三条是作者设计高级反汇编系统时的副产品,也给反编译程序用。使用上,比监控反汇编命令更灵活,看起来更清楚些。  
 DUMP ( a1 a2 -- ) 显示内存代码。从 a1 到 a2,分成三栏,左边是地址,中间是内存代码,用 16 进制显示,右边是相应的 ASCII 字符,一行显示八个字节内容。  
 DONGCIM ( -- ) 显示一个词的全部机器码。格式:DONGCIM name 按照 DUMP 格式显示。  
 FMCHG ( a--n ) 计算单词定义中一个 FORTH 码的长度,供反编译用。a 是一个定义中存某代码的地址。n 是这个词的代码长度。

下面介绍的 FBY 是系统中最大的词。它使系统更加透明,是分析、改造系统不可缺少的工具。笔者从一开始接触 FORTH 就悉心研究反编译,并且不断进行

改进、完善。它还可以作为编程辅助工具,编程时,如果记不清一个词的功能和用法,利用反编译,大多数情况下不必再去查使用说明书,笔者就是这样的。另外,系统中的词都是精心设计的,有些还经多次修改,初学者可以把它们作为编程范例。

FBY ( -- ) 反编译。格式:FBY name  
 对于用冒号定义的词,按定义格式显示来;对于由汇编定义的词,则反汇编出来。这个词使用了 240~2FF 作表格存储区,因此这段地址不要存放其他数据。

VD ( -- ) 另一种格式反编译。只能处理冒号定义的词。格式:VD name  
 显示内容共三排,左边是代码所在地址,中间是词中的代码,也就是各词的执行地址,右边是相应的词名。

以上两个词加上 DONGCIM,能在不同层次上反映出这个词的编码情况。

### 系统用词

以下这些词,一般是给系统用,如果使用不当,可能造成负作用,请初学者注意。这些词的使用情况,用反编译可以查到。

ENTER	冒号定义词中的第一个词
EXIT	冒号定义词中的最后一个词
DOHD	DO 的运行码
DOEND	LOOP 的运行码
+DOEND	+LOOP 的运行码
VAR@	VAR 的运行码
CON@	CONST 的运行码
LIT	数字进栈用
'	词执行地址进栈用
STR@	字符串用
SLIT	" 的运行码
FND	GETVERB 用
JIFX	条件转移。结构词用
JMPX	无条件转移。结构词用
;;DEF	;;用
;;ENT	;;用
;;EXT	;;用

### 其他词

这些词不便归类,一并在此介绍。

^ C 恢复程序运行。这是不可显字符。对于 CEC-I,可用 QUIT 键。在程序运行当中,若想中断运行,可以按一下 ^ C,程序即可中断,回到解释状态。如果还想恢复运行,在不改变两个堆栈状态的情况下,可以使用这个词,^ C 程序将继续运行。注意:中断时,键入的 ^ C 是由系统判别的,而不是 ^ C 这个词起作用。  
 MON 进入监控,再按 Reset 键或 800G 能回到系

统。

- ABORT 中断程序运行,堆栈复位。
- EXEC (a--) 执行一个词,其执行地址在堆栈里。
- MOVE (n a1 a2 --) 把a1起始的n个字节移到a2起始的内存区,从低位字节开始移动。
- MOVEUP (n a1 a2 --) 把a1-1起始地址向前数的n个字节移到a2-1开始向前的内存区。从高地址开始移动。
- MAIN (--) 主程序。进入系统时由此处开始运行。
- SYSEERR 发生错误时显示错误信息,中断程序运行,清除堆栈。
- DEFERR 遇到一个词无定义时用,它调用SYSEERR,响铃,显示XXX-UNDEF。这里XXX代表输入中那个无定义字符串,它在词典中不存在。
- NSTERR 结构错误。当IF...THEN这类结构前后不匹配时就会执行NSTERR,它调用SYSEERR,响铃,显示XXX-NESTING,清除堆栈。
- ZL0 (a--c) 间接取一字节。其功能可定义为: ZL0 @ c@ ;
- NOP (--) 空操作。可以作调试和延时用。其定义: NOP ;
- SAVE 把编辑缓冲区内内容存入磁盘,形成一个T类文件。格式: SAVE name ✓ 这里name是文件名
- S= (n1 a1 n2 a2 --f) 比较两串是否相等,相等则f=1。

### 系统使用的某些问题

#### 子程序调用

FORTH是一个开放系统,可以用调用外部子程序的方法定义词。这样定义的词在功能和所调用的子程序是一样的。词典中的LL、VTAB等词就都调用了监控中子程序。

CALL (a--) 调用a中子程序

有些子程序需要设置寄存器,或将返回结果存放于寄存器。系统定义了四个寄存器保护单元,AREG、XREG、YREG、PREG,分别代表A、X、Y、P寄存器。这些都是一字节常量指针,作变量用。存取应该用C@和C!。这两个单字节存取词。例如VTAB词的定义是

: VTAB VT C! FC22 CALL ;

#### 错误信息

系统中的错误信息有两种,一是XXX-UNDEF

这里XXX代表一个词典中不存在的字符串。输入错误、10进制状态下使用16进制数或引用还未定义的词,都会显示此信息。所有词都必须先定义后使用。

另一种是

XXX-NESTING

表示结构错误。结构词不匹配就会显示这个信息,这时有必要检查程序中结构词的使用是否有误。

#### ASCII码

系统输入输出多是用ASCII码进行的,每个字符会占一个字节,有效位是低七位,最高位在这个系统中是1,例如空格码是A0,回车码是8D。想了解一个字符的ASCII码,可用ASC这个词。

### 建立自己的系统

盘中有一个文件名为SAVEITX,用来建立新的系统。当你把系统改造之后,检查无误,就可以

CLR DAPP SAVEITX ✓ 再

LOAD ✓

屏幕上会显示出SAVE FILE NAME?

问你给新系统起个什么名字,这时打入给系统起的文件名,再回车,系统就会把新系统作为B类文件存在盘中,然后自动启动新系统,这时就进入你的系统了。

### 盘中文件介绍

盘上提供了一些文件,有些是演示用的,有些是实用工具。其中一部分,是原系统提供的。

FORTH][D1.0	系统主文件
ASSEM	汇编器
SAVEITX	建立新系统用
SHASSEM	学习汇编的参考文件
RANDOM	随机数发生器
DIM	数组词
STRING	字符串词
MRSIC	音乐演示程序
LORES	低分辨率作图词,给游戏用
BREAKOUT	游戏程序
DEFLECTION	游戏程序
XUNHUAN	循环除法示例
FN!	阶乘,可作1000!
BN!	阶乘,用编译BASIC编,B类文件
AN!	除乘,BASCI编,A类文件。
NBL	逆波兰

其中NBL用来把中缀法表示的关系式变成后缀法表示的式子。这是一个小程序,演示用。只能用单个数字表示一个数,其余符号是+、-、×、/和左右括号。用法

DAPP NBL ✓

LOAD ✓

此时词典中已有NBL,用NBL后跟一个表达式就可显示逆波兰式子。

FN!、AN!、BN!三个阶乘文件可以用来比较FORTH和BASIC两种语言的速度。



# 第十三讲 FORTH—83标准摘要

丁志伟

这一讲介绍 FORTH 语言的现行标准: FORTH—83标准。一般情况下, 程序设计语言讲座是不必介绍标准的, 但由于 FORTH 语言易于改造和移植, 灵活多变, 是否有一个标准, 对于学习、交流就很重要了。FORTH—83标准是一个成熟的标准, 影响很大, 每一个 FORTH 爱好者都应该对它有所了解。

一般读者没有见过程序设计语言的标准文本, 通过这一讲可以具体地了解到这类标准是什么样的。当然, 其他语言的标准文本的写法会有所不同。

FORTH—83标准文本的内容较多, 限于篇幅, 以下将以 FORTH—83标准文本的顺序为线索, 以引用原文为主, 摘要对其介绍。

这个标准是由 FORTH 标准组在1983年8月出版的。FORTH 标准组是由一些对 FORTH 有很丰富经验和技术的专家所组成, 这些专家包括用户和开发者。标准的基本内容如下:

1. 前言
2. 目的
3. 范围
4. 协定
5. 术语的定义
6. 参考文献
7. 要求
8. 标号
9. 用法
10. 错误条件
11. 词汇表的表示方法
12. 要求词组
13. 双字长数扩充词组
14. 汇编程序扩充词组
15. 系统扩充词组
16. 控制参考词

以及附录:

- A. FORTH 标准组成员情况
- B. 非控制参考词
- C. 实验建议
- D. 标准组的章程
- E. 建议和意见表格和说明

## 第1部分: 前言

FORTH 语言标准的存在, 并不反对用户(不论是否赞成这个标准)购买或使用与本标准不一致的产品、程序和方法。FORTH 标准是经受定期审查的, 请用户留意采用最新版本。

在70年代中期, 欧洲的 FORTH 用户小组(EFUG)开始对 FORTH 进行标准化工作, 以便提高

FORTH 语言应用软件的可移植性。由此产生了 FORTH—77标准。之后, FORTH 标准组出版了暂定 FORTH—78标准。根据1979年的标准组会议, 于1980年又出版了 FORTH—79标准。

FORTH 语言的特点, 允许将该语言扩充以适应特殊需要及不同的硬件系统。一位程序设计员或软件销售者可以选择严格符合标准的 FORTH 系统, 也可以选择不同于标准的系统。如果本标准对某方面没有明确地提出要求或限制, 则一个系统或应用软件可以在这方面作任何处理, 当然, 在其他方面还应遵从标准。

(注: D1.0版本是不符合这个标准的。)

## 第2部分: 目的

本标准的目的是: 在不同的 FORTH—83标准系统上的以源码形式出现的 FORTH—83标准程序具有可移植性。也就是说, 一个标准程序应在所有的标准系统上等效地执行。

## 第3部分: 适用范围

本标准适用于任何 FORTH—83标准系统上所执行的任何 FORTH—83标准程序, 条件是有足够的计算机资源(内存和外存)。

## 第4部分: 约定

当发生选择冲突时, 标准组遵从如下原则:

1. 功能的正确性—已知界限, 确切的;
2. 可移植性—当程序在标准系统间移植时, 其结果可以重复;
3. 简洁性;
4. 命名的明晰性—  
用说明性名称, 而不用过程名称。例如:  
采用[COMPILE], 而不用'C  
采用 ALLOT 而不用 DP+!
5. 普遍性;
6. 执行速度;
7. 节省内存;
8. 编译速度;
9. 历史上的延续性;
10. 便于发音;
11. 便于理解;

第5部分, 是术语定义。介绍了标准文本中所使用的术语的定义, 共64个

## 第7部分: 要求

7.1.1 每个标准系统应附有下列资源要求说明:

1. 系统使用的词典空间(用字节数表示);
2. 留给用户的词典空间(用字节数表示);
3. 数据堆栈空间(用字节数表示);

4. 返回堆栈空间(用字节数表示);
5. 系统使用的大容量存储块的范围;
6. 给用户的大容量存储块的范围;
7. 操作员可利用的终端设备;
8. 系统对每个一般错误条件和特殊错误条件所采取的动作。

7.1.2 每个标准程序应附加下列最低要求的说明:

1. 词典空间(用字节数表示);
2. 数据堆栈空间(用字节数表示);
3. 返回堆栈空间(用字节数表示);
4. 大容量存储块范围;
5. 操作员用的终端设备。

72 检验要求。即检验本标准的最小环境。

1. 应用词典具有2K字节存储容量;
2. 64字节的数据堆栈;
3. 48字节的返回堆栈;
4. 32块的大容量存储器容量,编号为0到31;
5. 作为操作者终端的 ASCII 码输入/输出设备。

**第8部分是“标准标识”,指进入系统后的提示信息:**

如果一个系统包含本标准的全部要求词,并遵照本标准的文本,则该系统可标出:

FORTH-83 Standard(即:FORTH-83标准)

**第9部分是“应用”。介绍对 FORTH 系统的某些要求。**

**这里只列出小标题**

- 9.1 词的名称和定义
- 9.2 存储寻址
- 9.3 返回堆栈
- 9.4 编译
- 9.5 终端输入和输出
- 9.6 标准系统间的程序移植
- 9.7 多道程序设计
- 9.8 数
- 9.9 程序控制结构

**第10部分:错误条件**

10.1 当一个错误条件发生时,标准系统可能采取下列的一个或几个动作:

1. 忽略错误并继续运行;
2. 显示错误信息;
3. 执行一个特定词;
4. 置于解释态并解释一个操作;
5. 置于解释态并开始解释操作;
6. 其它的系统相关动作。

10.2 一般错误条件

下列错误条件用于许多情况。这些错误条件列表如下,这是针对不同的状态和不同的词而言的。

1. 在碰到一个要求的〈名称〉或定界字符前,输入流已耗尽;
2. 在文本解释或编译过程中,堆栈空间不够或堆栈中数据不足;
3. 在文本解释或编译过程中,没有找到需要的词或找到无效的数;

4. 编译时遇到错误的嵌套控制结构;
5. 在非编译状态下和不在编译一个冒号定义时,执行只能用于编译的词;
6. 删去一个正确执行所要求的词;
7. 词典中剩下的空间不够;
8. 堆栈参数超出范围,例如在词汇表中规定为正16位数,却使用了负数;
9. 不能作大容量存储的正确读或写。

**第11部分是词汇表说明,主要内容是本标准中单词及说明,按 ASCII 字符排序。**

在本标准内,词名称都使用大写字母。

11.4 属性

大写符号表示定义词的属性:

C 仅在冒号定义编译时使用的词。

I 它表示该词是 IMMEDIATE(立即执行的)。在正常情况下,编译时这个词将会被执行。

M 这个词在多道程序设计时可能会发生冲突。

U 用户变量。

11.6 发音

词的名称用自然语言标注,用双引号(”)括上。

11.9 引用其它词和定义

词汇表中的定义可能要参考其它词定义或术语定义。用短语“参见:”表示这种引用。

**第12部分:要求词组**

12.1 要求词组的层次

把要求词组的词按类似的特征分类。但不应从这种分类法中推论出任何具体功能。

核心层(Nucleus layer):

!	*	*/	*/MOD	+
+1	-	/	/MOD	0<
0=	0>	1+	1-	2+
2-	2/'	<	=	>
>R	?DUP	@	ABS	AND
C!	C@	CMOVE	CMOVE>	COUNT
D+	D<	DEPTH	DNEGATE	DROP
DUP	EXECUTE	EXIT	FILL	!
J	MAX	MIN	MOD	NEGATE
NOT	OR	OVER	PICK	R>
R@	ROLL	ROT	SWAP	U<
M*	UM/MOD	XOR		

设备层(Device layer):

BLOCK	BUFFER	CR	EMIT	EXPECT
FLUSH	KEY	SAVE-BUFFERS	SPACE	
SPACES	TYPE	UPDATE		

解释程序层(Interpreter layer):

#	#>	#S	#TIB	'
(	-TRAILING	.	.	(
<#	>BODY	>IN	ABORT	BASE
BLK	CONVERT	DECIMAL	DEFINITIONS	
FIND	FORGET	FORTH	FORTH-83	HERE
HOLD	LOAD	PAD	QUIT	SIGN
SPAN	TIB	U.	WORD	

编译程序层(Compiler layer):

+LOOP	,	"	:	;
ABORT	ALLOT	BEGIN	COMPILE	CONSTANT

```

CREATE      DO      DOES>  ELSE    IF
IMMEDIATE  LEAVE  LITERAL LOOP
REPEAT     STATE  THEN    UNTIL   VARIABLE
VOCABULARY WHILE  [      [' ]
[COMPILE] ]

```

12.2 要求词组词汇表。

(这里列出第一个和最后一个)

```

! 16b addr  -- "store" 79
16位数被存储在地址 addr 中。
] -- "right-bracket" 79

```

置编译态。依次编译来自输入流的文本。典型的使用法参见 LITERAL。参见：[

### 第13部分：双字长数扩充词组

#### 13.1 双字长数扩充词组的层次

核心层

```

2!          2@          2DROP  2DUP          2OVER
           2SWAP D+      D-          D0=
D2/         D<         D=        DABS         DMAX
DMIN        DNEGATEDU<

```

设备层 无

解释程序层 D. D. R

编译程序层 2CONSTANT 2VARIABLE

#### 13.2 双字长数扩充词组的词典(部分举例)

```

2! 32b addr -- "two-store" 79

```

把32位的数存储在地址 addr。参见：“数”

```
D< d1 d2 -- flag
```

参见要求词组中的完整定义。

### 第14部分：汇编程序扩充词组

#### 14.1 汇编程序扩充词组的层次

核心层 无

设备层 无

解释程序层 ASSEMBLER

编译程序层 ;CODE CODE END-CODE

#### 14.2 汇编程序扩充词组的使用方法

由于机器语言程序设计的系统依赖性，因此一个标准程序不能使用 CODE 或 ;CODE。

### 第15部分：系统扩充词组

#### 15.1 系统扩充词组的层次

核心层 BRANCH ?BRANCH

设备层 无

解释程序层 CONTEXT CURRENT

编译程序层 <MARK <RESOLVE  
>MARK >RESOLVE

### 第16部分：控制的参考词组

控制参考词是一种词定义，它们不是要求词。在标准系统的 FORTH 词汇支中不允许存在一个非标准的定义，这些词具有现在的用法，也可能是下一个标准的候选词。(部分举例)

```
--> -- "next-block" I,M,79
```

--(编译时)

连续解释下一个屏面块，可以用于穿过一个块边界的冒号定义中。

```
U.R u +n -- "u-dot-r" M,83
```

用 BASE 基值转换 u，然后从右对齐的每个字段为 +n 个字符宽的格式，显示此无符号数。如果需要显示 u 的字符数大于 +n，则产生错误条件。参见：“数的转换”

讲座到此告一段落。由于不同的读者会有不同的要求，因此一个讲座难以满足每个人的要求。如果您有什么问题、建议，或需要软件，欢迎与我们联系。

(接第15页)

使学员加深对软件功能的认识，达到举一反三的效果。我想，这是学习任何一种应用软件都应遵循的原则。40 个问题没有，也不可能包括 C-WordStar 使用中的所有问题，挂一漏万在所难免。例如，关于非文本编辑就没有论及，字型变换也没有提到。重要的是学习应用软件的使用时要勤上机，上机遇到问题解决不了，不要绕道走，而应千方百计找出变通的处理方法。只要这样作了，对这种应用软件的使用就会得心应手。

由于本人水平有限，文中观点不一定正确，仅供大家参考，同时恭请同行专家和读者不吝赐教。

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

邮编：430070

电话：027-700564

FAX：027-703396



# 固体录音机的制作

北京宣武区科技青少年馆(100053) 罗明宽 车金相

《固体录音机》并非指家庭所用的录音机,而是一种不需要磁带的新型录音机,这种录音机体积小、寿命长、使用方便,可随录随放,虽说录放时间较短,但它的应用领域越来越广泛,已引起了科技界高度重视。

本刊第十期刊登了由 T6668 语言处理器和其它电路组成的人工控制《固体录音机》电路,其电路原理、T6668 语言处理器基本知识、动态 DRAM 和音频功率放大器 LM386 等知识都有所介绍,本期不再赘述。

为了帮助读者制作和对 T6668 语言处理器的开发和应用,我们为读者制备了一批《固体录音机》的印制板图,供广大读者选用,同时本期向读者介绍固体录音机的制作。

## 一、固体录音机印制板图

为了方便读者对照电路原理图进行组装,调试,本期仍给出其电路原理图,如图1。

它的印制线路板图共两块,分别为图2和图3。印板尺寸分别为  $60 \times 60 \text{mm}^2$ ,  $50 \times 50 \text{mm}^2$ 。其中图2是放大1.5倍的印制板元件面图,标有元件名称或符号,称为印制板的正面,图3是图2的背面,称为焊接面(焊点)。

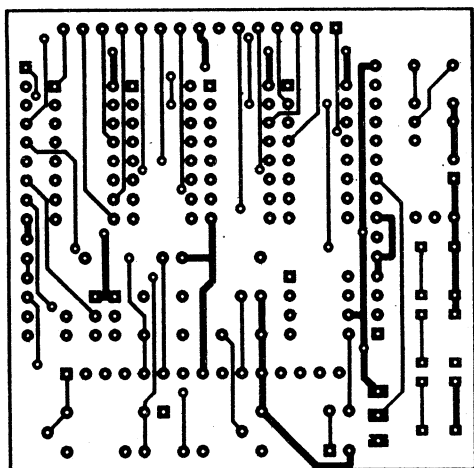


图3

将 T6668 的60个管脚焊接在图2的面板上。便组成双层结构,小巧灵活的固体录音机。

## 二、制作与调试

读者在购到或自制印刷线路板(自制注意尺寸)后,建议读者应与电路原理图一一对照,一能进一步熟悉其工作原理,二能了解每个元器件在板图中的具体位置,为准确焊接提供方便,印制板图上所标明的符号含意如下:

1.  $X_1 \sim X_9$  是指单排插针(或插座)焊接位置。

其中  $X_1 \sim X_4$  为四排插针(座),共60个,每排15个,通过此插针(座)可与 T6668 的60个管脚接到此板上,完成两块印制板的连接。 $X_5$  为喇叭接线端, $X_6$  为驻极话筒的接线端, $X_7$ 、 $X_8$  为选择动态 DRAM 片数的控制端,可用双排三个插针焊接在指定的焊点上。

2.  $K_1 \sim K_4$  是指按键和开关的焊接位置。

其中  $K_1 \sim K_3$  是小型按键, $K_4$  是拨动开关。

3.  $W_1$  和  $W_2$  是可调电位器。其中  $W_1$  是调节录音质量的。 $W_2$  是放音时控制音量的。

4.  $U_1 \sim U_5$  是指集成块(41256、LM386)焊接位置。

其它符号是电阻、电容、发光二极管等,这里不再说明,在组装焊接过程中希望读者注意以下几点:

1. 在  $U_1 \sim U_5$  的位置上不要直接将 DRAM 41256 和 LM386 芯片焊接在板上。最好焊接管座。为拔下插上更换芯片提供方便,焊接管座时,管座的缺口方向应与印制板图标明的方向一致。在管座插入芯片时,芯片的缺口方向应与管座缺口方向一致,否则会烧坏芯片。

2. 焊接电解电容时,注意极性。

3. 驻极体话筒要用屏蔽线引出,起屏蔽作用的金属网接地。

4. 单排插针的座和插针一律用窄形的。以保证两块板(图3和图2)准确进行连接。

5. 电源的接线端注意极性,图2中已标出。

6.  $W_1$  和  $W_2$  最好用导线引出,在板外与电位器相连,便于调试

7.  $C_1$  电容的选择最好不使用瓷片电容,否则杂音较大。

全部焊接完毕后,经认真检测,无误可通电进行调试。一般来说,只要器件完好,电路便能工作,只不过需要细调  $W_1$  的阻值,图3提高录音的质量。

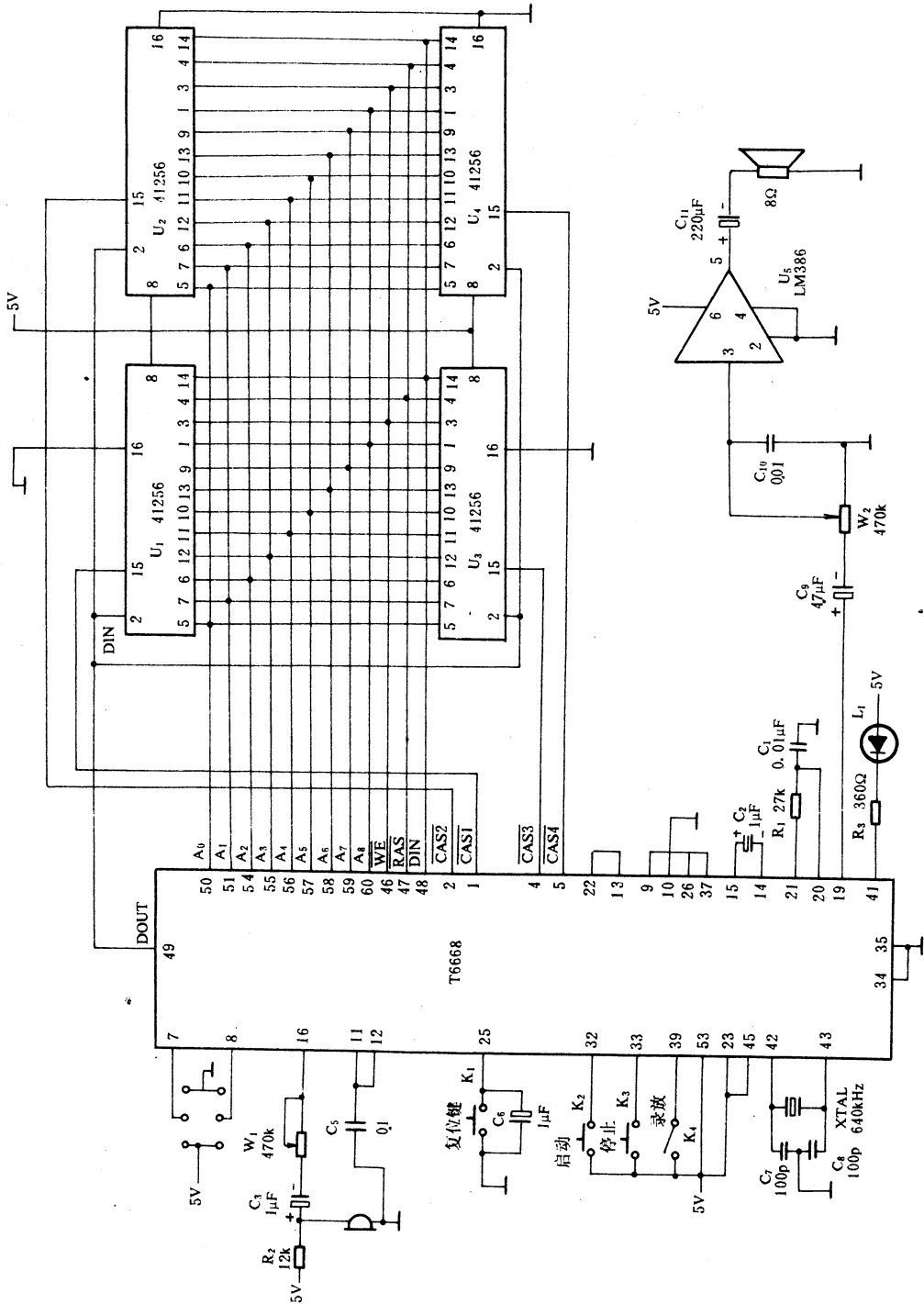


图1

进行录音之前,首先选择 DRAM 的片数,可通过 X<sub>7</sub>、X<sub>8</sub> 的中间插针的高低电平获得。如选择一片,可用两个短路帽将各排插针最右边两个插针连接起来,若选其它片数,可参考第十期有关说明。

#### 录音调试:

首先按一下复位键, (K<sub>1</sub>) 将录放开关 (K<sub>4</sub>) 打向录音状态 (向左拨动), 再按一下启动键 (K<sub>2</sub>) 发光二极管发光, 进入录音状态, 适当调节音量电位器 W<sub>2</sub>, 使音量大小适中, 可对驻极话筒讲话, 同时调节音质电位器 W<sub>1</sub> 的阻值, 当喇叭发出干脆、响亮的声音, 说明 W<sub>1</sub> 阻值合适。录的质量到底如何? 放音来检测。放音时, 将录放开关打向放音状态 (向右拨动) 适当调节音量电位器, 按一下启动按键, 开始放音。如果不满意, 再进入录音状态, 再一次调整 W<sub>1</sub> 的阻值, 通过几次录放调整, 使录音达到最佳效果取下 W<sub>1</sub>, 找一与 W<sub>1</sub> (调整后的阻值) 相近的电阻焊接在 W<sub>1</sub> 左右两边的焊点上。

#### 放音调整:

放音调整, 主要对 C<sub>10</sub> 容量调整, 容量过大, 放出的声音低沉, 容量过小, 声音清脆, 但杂音增大, 读者需反复调整, 达到满意效果。

由于选用一片 DRAM, 录放的时间较短, 时间一

到, 发光二极管由亮变灭, 说明电路录放工作结束。在这种情况下, 若想再进行录或放必须先按一下复位, 电路才能工作, 这点读者必须注意。如果电路处于录放状态, 按一下停止键 (K<sub>3</sub>), 发光二极管由亮到灭, 电路进入守候状态, 即若为录音状态, 保持录音功能, 若原为放音状态, 保持放音功能。

全部调好以后, 可放在一个小盒里面, 电源可用 3 节 1.5V 电池。固体录音机制作完毕, 为了减少杂音, 在录音时可将音量电位器旋至最小放音时, 再调整容量电位器。

### 三、元件选择

元器件质量的高低, 直接影响录音效果。DRAM 选用 41256, 音频功率放大器选用 LM386。T6668 的 42 脚和 43 脚之间跨接的陶瓷振子, 频率可在 650KHz—640KHz 之间选取, 也可用同频率的晶体代用或者频率相近的陶瓷滤波器代用。驻极话筒选用动态范围较大、中灵敏度为宜。C<sub>1</sub>、C<sub>5</sub>、C<sub>10</sub> 不要使用瓷片电容, C<sub>6</sub>、C<sub>2</sub> 选用矮型的为好, 喇叭 8Ω 为好, 其它元器件无特殊要求。

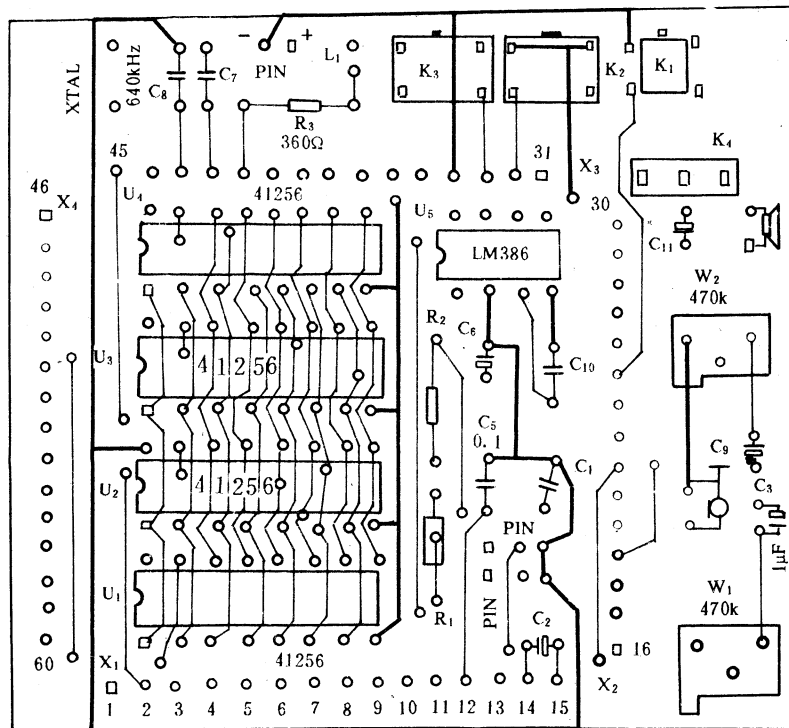


图2



# 实验教学计算机的研究与开发

天津师范大学计算机系(300074) 王勤民

众所周知,计算机科学是一门实践性和技术性很强的新兴学科,故在培养计算机专业人才时,既要求学生掌握较系统的专业理论知识,更要求学生具有较强的实践和动手能力,因此,实践环节普遍受到重视。而在硬件实验方面,实验手段的优劣又尤为计算机教育工作者所关注。但由于种种原因,目前大多数大专院校所使用的硬件实验手段都不很理想,其中使用通用单板机的居多。因为这种单板机是为一般用途而设计的,是一种通用系统,用其进行计算机实验教学,无论是在教学的直观性、可实验性、灵活性和效率等方面,都是不适宜的和难以胜任的。为此,我们研究开发了 LC-10LABCOMPUTER 实验教学计算机。根据实验教学的特点,我们在结构设计、电路设计及系统软件设计方面,进行了多方面的探索和尝试。

## 一、结构设计

为了使 学生通过基础理论学习和实验课学习,了解微机总体结构的全貌及各主要部件的工作原理,并获得应用微机的基本知识和技能,同时考虑到实验课的直观性和示范性,所以在整体结构上采用了“全开放——封闭”型设计方案。所谓“开放”,即以主机板代替面板,让计算机电路的所有元器件都展示在面板上,使学生对计算机的电路结构一目了然;所谓“封闭”,即将电源部分封闭在机箱内,一则保证安全,二则不需外接电源,使电源和计算机成为一个完整的系统,使用方便。整个机器的平面分布图如图1所示。

机器上方设计了五个50线的总线插槽,用于各种扩展功能,使学生一开始就对总线接口及功能扩展具有较深的感性认识;机器下方装有一块面包板和一排提供各总线信号的插座,使学生可根据实验要求,自己选择芯片,设计电路,不仅能加深学生对课堂内容的理解,且有利于提高学生的学习积极性,培养学生的动手能力。

## 二、电路设计

主机板电路框图如图2所示:中央处理器采用 Z-80芯片,1MHz 时钟,4通道计数器/定时器 Z-80CTC; 2KB RAM,4KB ROM; 30键小键盘,6位 LED 显示。在电路布局上,力图以教材中通用的硬件框图为模式,将电路分为三部分:CPU、存储器和输入/输出端口。每部分均为一块独立的电路板,且本身就是一个实验。三块板的实验做完以后,用三总线将它们连接起来就构成了主机板。30键小键盘及其接口电路和 LED 显示器及其接口电路也是两块独立的电路板,分为两个实验,将

它们用总线连到主机板上,即成为主机的输入和输出设备。另外,面包板和总线信号插座为一块独立的板。

根据教学要求,系统配置了七块实验接口卡:多用实验卡、RAM 扩充卡、PIO 卡、SIO 卡、A/D 转换卡、D/A 转换卡和 PIO 调节卡。这些卡是本着由浅入深、循序渐进的原则设计的,其特点是演示功能强,注意知识性与趣味性相结合,力求使实验教学生动活泼。

电源部分在机箱后面设置了+12V 电源开关,当不使用该电源时,可将其关断,以免因误操作或疏忽而损坏主机板电路。在整个系统电路中,还设置了限压保护,用来防止高电压的冲击。

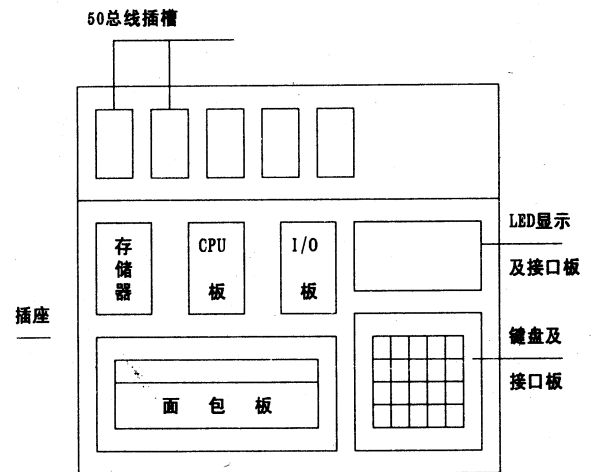


图1

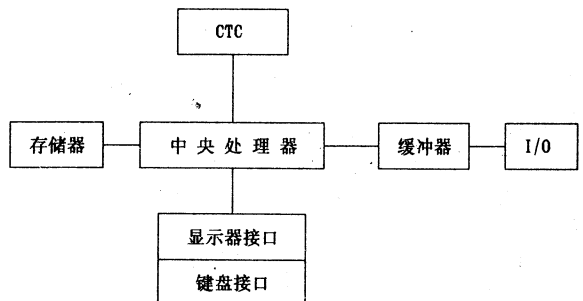


图2

### 三、系统软件设计

软件设计本着调用灵活方便,知识性与趣味性相结合的原则。监控程序将众多的实用子程序有机地连接起来,用户使用时可直接调用。

为便于学习,提高学习效率,还编写了大量的应用子程序:各种算术运算符程序(如加法、减法、双字节乘法、除法,求平均值、求最大值等)、码制转换子程序(如

双字节二进制码转换成BCD码子程序、BCD码转换成二进制码子程序等)和测试及演示子程序(如RAM检查、数字钟、PIO中断测试、PIO演示、SIO测试,三角波发生、电机开环及闭环调速等子程序)。

该实验教学机可复盖计算机组成原理、微机原理及应用和微机接口技术等课程的基本实验。两年来经多所兄弟院校的使用,效果良好。

## AST286系统板总线故障的分析与排除

北京中国人民大学信息中心(100087) 梁杰熙

AST286系统板总线部分出现故障时,对排除其故障有一定的难度。因为AST286的总线结构在PC286总线结构的基础上进行了一些改动,增加了快速总线扩展槽与CPU直接相联。在控制器总线部分(X总线)将I/O串并适配器不进行隔离直接联在X总线上,这样的改动从维修角度讲增加了总线故障的多发性,再加上AST286系统技术资料的限制,所以在排除AST286系统板总线故障时,也就增加了一些难度。下面就维修中遇到的AST286系统板总线故障作一下实例分析。

在AST286系统板总线部分发生故障时,一般用逻辑笔在I/O槽上测试地址总线、数据总线,会发现电平态出现,在维修中遇到的几个AST286总线有故障的系统板均是这种现象。那么与总线故障有关的芯片除CPU本身外,主要与ROM BIOS(U<sub>10U</sub> 12), 82A204(U<sub>6</sub>)有关。82A204是地址总线驱动,82A205是数据总线驱动。在CPU与ROM BIOS的操作过程中其地址的发送、数据的传输分别要通过82A204, 82A205来完成。也就是说在AST286总线部分出故障时,首先要以82A204和82A205为核心来进行分析诊断。82A204和82A205芯片的引脚功能说明一般通过资料都可以查到。

例一:AST286微机中通过用逻辑笔针对其I/O槽进行总线状态的测试,发现地址总线的状态有电平出现,那么首先要以82A204为开端来进行查找故障点。82A204的地址输入是联接来自CPU的A<sub>1</sub>—A<sub>16</sub>的地址位。再通过其控制决定是SA总线操作还是XA总线操作还是MA总线操作,那么在I/O槽上的状态,是来自SA总线,但其取决于82A204的输入端A<sub>1</sub>—A<sub>16</sub>的状态,如果两个状态不对应,故障有可能就在82A204,如果两个状态对应但是错误的电平态,则

说明故障可能在82A204的外围芯片上即造成了还没有输入到82A204以前就已经出错。在这一例中对82A204的输入端进行测试发现其有电平态出现,那么在一般的PC286微机中就要对CPU的工作条件进行检查了,但在AST286中由于其增加了快速扩展槽与CPU直接相联,所以在这里就要考虑与快速扩展槽有关的控制器。在AST286的原配置中其快速扩展槽上联接的是RAM随机存储器控制卡,所以要对RAM随机存储器控制卡进行检查。结果发现是RAM随机存储器控制卡上的地址译码芯片损坏是造成这一故障的根本原因,排除RAM随机存储器控制卡的故障,此总线故障也就排除了。

例二:同样首先用逻辑笔对I/O槽进行测试,发现其数据总线有电平态出现。那么此时就要以82A205为开端进行检查。在检查82A205时发现其数据的双向输入输出状态不对应,此时对一般的PC286微机来讲就有可能是82A205本身有故障,但是在AST286微机中由于控制器总线部分(X总线)与I/O串并适配器直接相联,如果故障是在串并适配器就有可能影响到X总线的正常工作,而X总线中的XA是作为82A205的数据驱动控制。所以此时就要对串并适配器进行检查,但串并适配器上采用了AST286微机专用的控制芯片106000—36B(U<sub>29</sub>),由于无资料对其无法检查。所以在遇到与U<sub>29</sub>有关联的故障时。只有把U<sub>29</sub>取下才能判断故障点的位置。在这一例中采用了这种方法。发现取下U<sub>29</sub>后总线的工作状态就恢复正常了,也就是说,是U<sub>29</sub>造成了总线不能正常工作。但由于AST286微机的串并适配器控制部分是和系统板做在一体的,以及106000—036B芯片无法购到,所以在取下U<sub>29</sub>后还需购买一个串并适配器卡插在系统板的I/O槽上采取代系统板上的串并适配器控制部分。

## 全国计算机软件资格和水平 考试函授辅导班(第六期)招生

为配合国家人事部、电子部计算机司每年进行的全国计算机软件专业技术资格和水平考试,由中国软件行业协会考试指导委员会主办,清华大学计算机系、北京大学计算机系等校专家、教授主讲。

招生对象:各行业从事或有志从事计算机软件工作的人员。

辅导内容及方式:初级程序员(相当技术员)、程序员(相当助工)、高级程序员(相当工程师)、系统分析员(相当高工)四个级别。按考试大纲的要求(详细内容见简章),发放统编教材及有关资料,通过《函授通讯》进行教学指导、布置作业、解答疑难。在学员集中的省市设立面授站。备有全套辅导音像教材,供各地集中培训使用。

学习时间:1994年3月开学,至9月份结业,进行模拟考试。

报名办法:即日起开始报名,填写报名表1份(简章及报名表函索即寄)。

学习费用:报名费10元,学费(含教材、资料、《函授通讯》、邮寄费用等):初级程序员级120元,程序员级160元,高级程序员级170元,系统分析员级180元。联系电话:2561144-2373。

电挂:1331。联系人:张岱,邮汇:北京清华大学计算机系转中国软件行业协会培训学校。邮码:100084。银行汇款:北京清华园城市信用社,帐号:07252-13。收款:中国软件行业协会软件考试培训学校。

## 半导体存储器在自动打铃器中的应用一例

烟台海军航空工程学院(264001)

刘长茂 邱杰

自动打铃器是学校、机关、工厂通知上、下课(班)的一种仪器。它由时钟电路、打铃时刻记忆电路和控制驱动电路组成。过去,记忆电路一般通过机械方法或二

极管矩阵电路构成,存在记忆时刻较少,置定、改换记忆时刻不方便等缺点。近年来,随着电子工业的发展和半导体器件价格的降低,已经出现了由半导体随机存储器(RAM),取代二极管矩阵作为打铃器记忆电路的新一代产品。其基本做法是用数字式时钟电路产生的时刻信号作为地址(以1分钟为单位),把非打铃时刻对应的所有存储单元置“0”,打铃时刻对应的存储单元置“1”。当时钟走到打铃时刻,存储器输出高电平触发一延时电路,在延时期间接通电铃电源,实现自动打铃。

实现这一设计的主要难点在于如何有效地利用存储器的地址资源。时钟电路反映上午、下午、小时、分钟的七段笔划信号线共有22条。这22条线上电平变化对应的二进制数码共有1440个(每分钟1个),而这些数码又是不连续的,也不是单调增加或减小的。如果把这22条线直接作为存储器的地址,则要求存储器有22条地址线,地址空间为4M,这会造成存储器存储空间的极大浪费。

为了解决这一问题,设计采用了时间分段。列位存储和地址复用三条措施。

所谓时间分段,就是把一昼夜24小时分成上午0:00~9:59、上午10:00~下午12:59、下午1:00~9:59和晚上10:00~11:59四段。

列位存储,就是在每一时间段内,只对存储器存储空间的某一列进行读写操作,而对应于一个打铃时刻,只占用一个存储单元的一位。

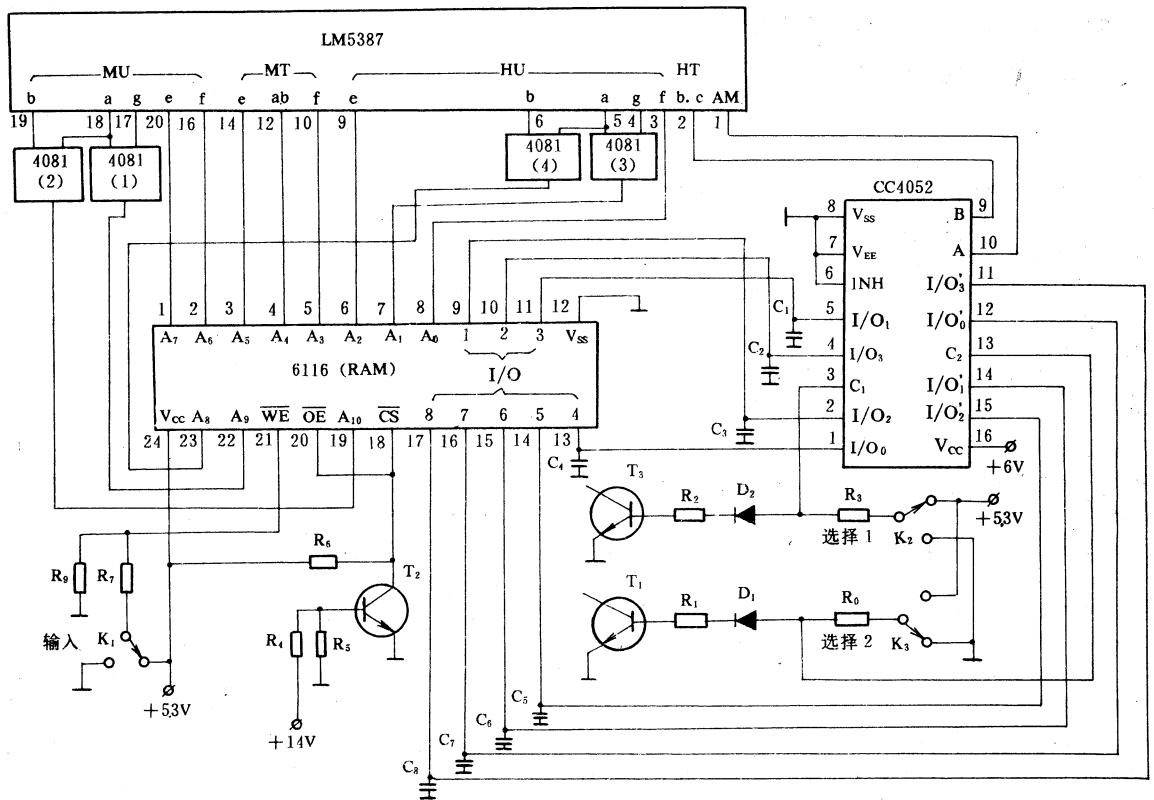
这样,对于同一地址的存储单元,可以在不同的时间段进行读写,而在四段时间的每一段,采用相同的地址系统,这就是地址复用。

采用这三条措施以后,只使用一块RAM 6116(11根地址线,2K字节存储空间),就能完成1440个时刻的存储任务。而且这1440个时刻只占用了6116存储空间的一半(低四位),另一半(高四位)则用于存储另一路电器(如收录机、扩音机等)的预置开、关时刻。这样,不仅有效地提高了存储空间的利用率,而且又增加了一路自动控制电路,实现了一机两用。

存储器与其它电路的连接如附图所示。6116的八根I/O数据线与CC4052的八根I/O数据线相连。CC4052作为I/O控制电路,它是一个双四路模拟开关。它由电平位移电路、四选一译码器和八个双向模拟开关组成。八个开关分成两组,一组中四个开关的一端分别与I/O<sub>0</sub>~I/O<sub>3</sub>相连,另一端连在一起接C<sub>1</sub>;另一组中四个开关一端分别与I/O'<sub>0</sub>~I/O'<sub>3</sub>相连,另一端连在一起接C<sub>2</sub>。两组开关受同一个地址译码器控制,A、B作为地址输入,当BA的电平组合为“00”时,C<sub>1</sub>与I/O<sub>0</sub>接通,C<sub>2</sub>与I/O'<sub>0</sub>接通;当BA为“11”时,C<sub>1</sub>与I/O<sub>3</sub>接通,C<sub>2</sub>与I/O'<sub>3</sub>接通。

由图可见,CC4052的A接时钟产生电路的AM(上午),B接HTb.c(小时的十位),这样在上午的十点钟以前和十点钟以后,以及下午的十点钟以前和以后这四个不同的时间段中,两组开关中各有一个开关接通,分别把6116数据线的低四位和高四位的一位与外





图

部控制电路相连。

例如在下午1:00~9:59这一时间段内,61116的 I/O4通过 CC4052的 I/O0-C<sub>1</sub>连到二极管 D<sub>2</sub>正极;6116的 I/O7通过 CC4052的 I/O'0-C<sub>2</sub>连到二极管 D<sub>1</sub>正极。这样,在这一段时间内的某一时刻,使6116的  $\overline{WE}$  为“0”时,可以通过选择开关 K<sub>2</sub>和 K<sub>3</sub>对6116相应时刻的某一地址单元的第4位和第7位置“0”或置“1”。而当  $\overline{WE}$  为“1”时,存在任一单元第4位和第7位的信息“1”,当相应的时刻到达时,便通过 I/O0-C<sub>1</sub>或 I/O'0-C<sub>2</sub>传递到 D<sub>1</sub>或 D<sub>2</sub>,使 T<sub>3</sub>或 T<sub>1</sub>导通,接通延时或驱动电路,完成打铃或电器开、关动作。这就是如上所说的时间分段和列位存储。

由上可见,在上午或下午的同一时间,外电路是通过不同的路径对存储器的相应单元的不同的位进行操作的。因而允许使用同一地址,这样,只要在0:00~9:59这一时间段内不出现重复地址,就可以满足设计要求。因此,只要用小时的个位、分钟的个位和十位这些七段笔画线形成地址就可以了。但是这些线还有20条,如果从中任选11条作为6116的地址线,在0:00~9:59这一时间段内必然会出现重复地址。计算表明,要保证这一时间段内地址不重复,小时的个位和分钟的个位分别要用5根笔画线,分钟的十位要用4根才行,即总共

要用14根线作为地址线,这仍然和6116的11根地址线相矛盾。

为此,采用了一个含有4个两输入端与门的4081逻辑电路,它把小时和分钟的个位的 a、b 和 a、g 两条信号线相与以后再输出,这就使小时和分钟个位分别用5条信号线形成4位地址线,而分钟的十位的 a、b(已在时钟电路内部相与为一条线)和 e、f 形成三位地址线,结果用14根7段笔画信号线形成了11位地址,满足了6116的要求。

电路中其它元件的作用如下

T<sub>2</sub>起开关作用,其集电极接片选  $\overline{CS}$  和输出允许  $\overline{OE}$ 。当有交流电源供应时,整流滤波电路输出的+14V电压加到 T<sub>2</sub>基极,使 T<sub>2</sub>饱和,  $\overline{OE}$ 、 $\overline{CS}$  为“0”,RAM 处于可读写状态,当交流供电停止时,14V电压消失, T<sub>2</sub>截止,  $\overline{OE}$ 、 $\overline{CS}$  为“1”,RAM 处于不可访问的静止状态,这时接到6116V<sub>cc</sub>的5.3V电压由备用电池供给,以保证存储的信息不丢失。

K<sub>1</sub>是输入开关,可对  $\overline{WE}$  置“0”或置“1”,控制6116处于写或读的状态。

C<sub>1</sub>~C<sub>8</sub>的作用是当6116处于写状态,对某一单元的某一位(或两位)置“0”或置“1”时,保持其余位的电平状态不变。

# 新型实时日历时钟 MSM6242

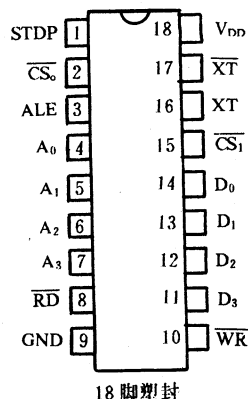
锦州三二二研究所  
(121000)徐民 富歌

随着微机控制技术的广泛应用,为控制系统的采样,定时等操作提供时间基准的实时日历时钟显得越来越重要。而以往采用的时钟芯片,如:MSM5832、MSM58321、DS1216、MC146818等都未能以最佳方式取得最满意的效果。而新型实时日历时钟 MSM6242 以其丰富的内部功能,通过最简洁的硬件连接和软件编程,为用户方便准确地提供了实时日历时钟信息。

## 一、基本功能和特点

MSM6242是一种不需任何接口电路就可以同CPU直接相连进行读写操作的CMOS实时日历时钟。它采用32.768kHz的振荡频率,为用户提供秒、分、时、星期、日、月、年信息,同时还可定期向CPU发中断请求,中断周期由CPU初始化时在芯片内部置好,共表一

有四种:1/64秒、秒、分、时。中断请求信号有两种方式:电平方式(需软件复位)和脉冲方式。CPU根据需要选择12/24小时两种计时格式中的一种。MSM6242还有自动闰年和日期调整功能,例如:写入1985年2月29日或11月31日时,它能自动转变为1985年3月1日或12月1日。并由此开始计时。这一功能为用户使用芯片提供极大方便。此外,MSM6242的功耗很低,在5V供电时,只有150 $\mu$ W,而在备电2V保持时仅20 $\mu$ W。



18脚塑封

图1

地址输入	地址输入				寄存器名称	数据				数据范围	备注
	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>		D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>		
0	0	0	0	0	S <sub>1</sub>	S <sub>8</sub>	S <sub>4</sub>	S <sub>2</sub>	S <sub>1</sub>	0~9	秒寄存器
1	0	0	0	1	S <sub>10</sub>	*	S <sub>40</sub>	S <sub>20</sub>	S <sub>10</sub>	0~5	十位秒寄存器
2	0	0	1	0	MI <sub>1</sub>	mi <sub>g</sub>	mi <sub>4</sub>	mi <sub>2</sub>	mi <sub>1</sub>	0~9	分寄存器
3	0	0	1	1	MI <sub>10</sub>	*	mi <sub>40</sub>	mi <sub>20</sub>	mi <sub>10</sub>	0~5	十位分寄存器
4	0	1	0	0	H <sub>1</sub>	hg	h <sub>4</sub>	h <sub>2</sub>	h <sub>1</sub>	0~9	小时寄存器
5	0	1	0	1	H <sub>10</sub>	*	PM/AM	h <sub>20</sub>	h <sub>10</sub>	0~2 或0~1	0-AM, 1-PM
6	0	1	1	0	D <sub>1</sub>	d <sub>8</sub>	d <sub>4</sub>	d <sub>2</sub>	d <sub>1</sub>	0~9	日寄存器
7	0	1	1	1	D <sub>10</sub>	*	*	d <sub>20</sub>	d <sub>10</sub>	0~3	十位日寄存器
8	1	0	0	0	mo <sub>1</sub>	mo <sub>8</sub>	mo <sub>4</sub>	mo <sub>2</sub>	mo <sub>1</sub>	0~9	月寄存器
9	1	0	0	1	mo <sub>10</sub>	*	*	*	mo <sub>10</sub>	0~1	十位月寄存器
A	1	0	1	0	Y <sub>1</sub>	y <sub>8</sub>	y <sub>4</sub>	y <sub>2</sub>	y <sub>1</sub>	0~9	年寄存器
B	1	0	1	1	Y <sub>10</sub>	y <sub>80</sub>	y <sub>40</sub>	y <sub>20</sub>	y <sub>10</sub>	0~9	十位年寄存器
C	1	1	0	0	W	*	W <sub>4</sub>	W <sub>2</sub>	W <sub>1</sub>	0~6	星期寄存器
D	1	1	0	1	C <sub>D</sub>	30 Sec ADJ	IRQ FLAG	BUSY	HOLD	-	控制寄存器D
E	1	1	1	0	C <sub>E</sub>	t <sub>1</sub>	t <sub>0</sub>	ITRPT /STND	MASK	-	控制寄存器E
F	1	1	1	1	C <sub>F</sub>	TEST	24/12	STOP	REST	-	控制寄存器F

## 二、数据寄存器和控制寄存器

MSM6242内部有16个寄存器(表一),其中13个数据寄存器,3个控制寄存器,寄存器中的数据形式为BCD码。写入寄存器的数据不应超出寄存器允许的数据范围。

13个数据寄存器中存储着用户需要读取的各个信息。选择24小时模式时,H<sub>10</sub>中的PM/AM位无效,读值为0。选择12小时模式时,h<sub>20</sub>被置位。W寄存器数据范围0~6,如表二。

表二

W <sub>4</sub>	W <sub>2</sub>	W <sub>1</sub>	星 期
0	0	0	星期日
0	0	1	星期一
0	1	0	星期二
0	1	1	星期三
1	0	0	星期四
1	0	1	星期五
1	1	0	星期六

MSM6242中有三个相当重要的控制寄存器,芯片中的许多功能都是它们控制完成的。详细介绍如下:

CD 控制寄存器

a: Hold(D<sub>0</sub>)——该位置“1”时,秒时钟信号被禁止进入秒计数器 S<sub>1</sub>。这时, Busy 位可读,当 Busy=0时,可对寄存器 S<sub>1</sub>——W 读写。Hold=1期间,如有进位产生,在 Hold 为“0”之后,秒计数器加1(需要 Hold=0时间不超过1秒)。CS<sub>1</sub>=0时, Hold=0。

b: Busy(D<sub>1</sub>)——一般 Busy=0时,对 S<sub>1</sub>——W 的读写是准确的。

c: IRQ FLAG(D<sub>2</sub>)——该状态位对应于 STD. P 输出。STD. P=0时, IRQ=1, STD. P=1时 IRQ=0。IRQ=1表示对 CPU 有中断请求, MASK(CE 中 D<sub>0</sub>)=0时 STD. P 的输出根据 CE 中 D<sub>2</sub>, D<sub>3</sub> 置定的时间变化。CE 中的 D<sub>1</sub> (ITRPT/STND)=1时, STD. P 输出为“0”,直到 IRQ 被置“0”。IRQ=1时,新中断无效。ITRPT/STND=0时, STD. P 的中断请求输出为宽度 7.8125ms 的负脉冲,此期间如果置 IRQ=0,输出恢复“1”状态。对 CD 中的 Hold 和 ±30ADJ 等操作应对 IRQ 置“1”。

d: ±30ADJ(D<sub>3</sub>)——±30秒调整位。“1”有效。D<sub>3</sub>=1时内部数据寄存器 S<sub>1</sub>——W 不能进行读写操作, 125μs 之后 D<sub>3</sub> 自动变为“0”。

CE 控制寄存器

a: MASK(D<sub>0</sub>)——控制 STD. P 输出。MASK=1时, STD. P=1; MASK=0时, STD. P 可以按两种模式输出。

b: INTRPT/STND(D<sub>1</sub>)——用于切换 STD. P 的两种模式输出。D<sub>1</sub>=0时, STD. P 中断输出时产生一个宽度 7.812ms 的固定负脉冲。另一模式为低电平状态输出,直到 IRQ 被置0。这两种模式输出都需要 MASK=0。中断时刻由 CE 中的 T<sub>0</sub>(D<sub>2</sub>)和 T<sub>1</sub>(D<sub>3</sub>)决定。

c: T<sub>0</sub>(D<sub>2</sub>), T<sub>1</sub>(D<sub>3</sub>)——决定了 STD. P 中断输出的周期,如表三所示。

表三

T <sub>1</sub>	T <sub>0</sub>	周期	占空比
0	0	1/64秒	1/2
0	1	秒	1/128
1	0	分	1/2680
1	1	时	1/560800

D<sub>1</sub>=0时,负脉冲宽度为 7.8125ms 与 T<sub>0</sub>(D<sub>2</sub>)和 T<sub>1</sub>(D<sub>3</sub>)无关。

利用 STD. P 的波形输出可调整基准振荡频率。S<sub>1</sub>——H<sub>1</sub>进行写操作时 STD. P 输出不变化。

CF 控制寄存器

a: REST(D<sub>0</sub>)——用于清零小于1秒的内部计数器, REST=1时,计数器持续复位。REST 被写入0时,计数器被启动。CS<sub>0</sub>=0时, REST 自动变“0”。

b: STOP(D<sub>1</sub>)——改变该标志,停止和启动时产生 122μs 延时, 1=停止。0=运行。

c: 24/12(D<sub>2</sub>)——方式选择。(D<sub>2</sub>)=1选择24小时模式, (D<sub>2</sub>)=0时选择12小时模式,写入(D<sub>2</sub>)位时, REST 必须为1。

d: TEST(D<sub>3</sub>)——(D<sub>3</sub>)=1时,芯片处于测试状态,秒计数器记录的脉冲频率为 5.4163kHz,不再是秒信号。这时 STDP 和 REST 标志不能禁止内部计数器。

三、硬件的基本结构和连接

芯片引脚如图1。四位地址线 A<sub>0</sub>——A<sub>3</sub>,用于对片内16个寄存器的寻址。四位数据线用于与 CPU 的数据传输。XT, X̄T 外接 32.768kHz 晶振。当外接振荡时钟脉冲时,无论 CMOS 还是 TTL 输出都连至 X̄T,同时 XT 悬空。STD. P 脚一般做中断输出,用法参见寄存器功能。CS<sub>1</sub>, CS<sub>0</sub> 片选线, RD 读线, WR 写线, ALE 控制线,这五条线分别有两种方式供 CPU 使用,对 MSM6242 进行读写操作,时序图见图2、图3。

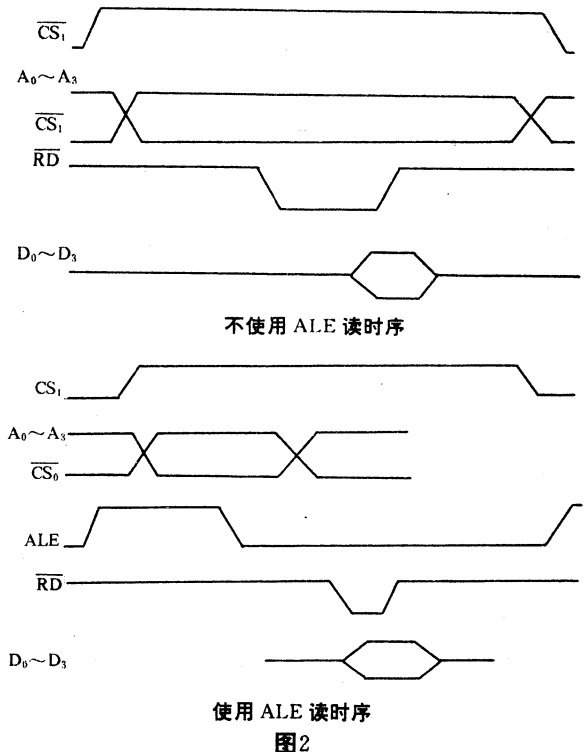
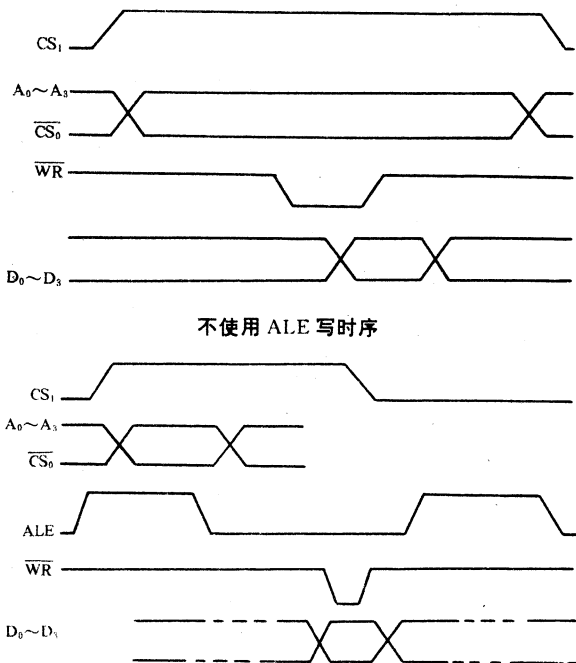


图2



使用 ALE 写时序

图3

MSM6242与CPU的连接相当简单,不需任何接口器件。一般有程序查询和中断请求两种方式。图4以单片8031为例,表示MSM6242与CPU的中断方式的连接。CPU把MSM6242当作一般外接RAM进行读写。

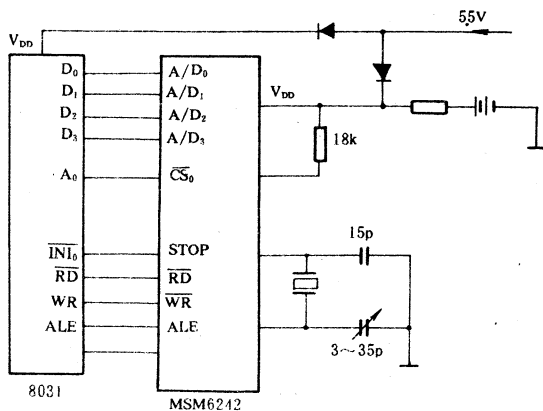


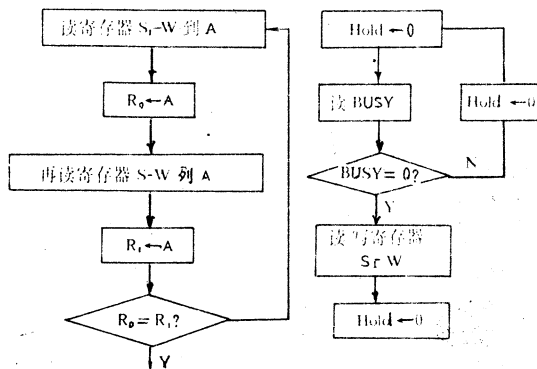
图4

#### 四、MAM6242的一些基本操作

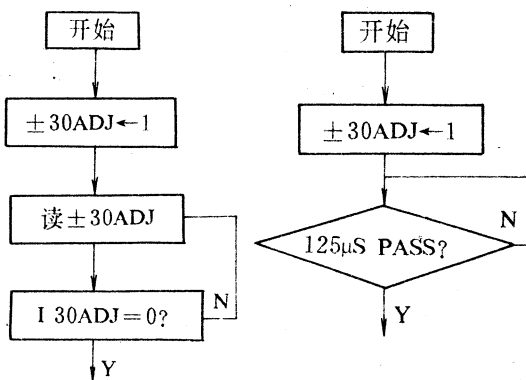
虽然MSM6242与CPU的硬件连接和软件编程

都很简单,但为了能使CPU准确可靠的获得信息,还要遵守一些基本的操作流程。

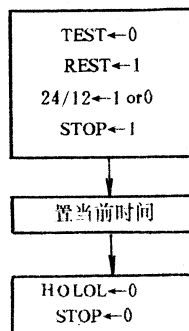
#### 1. 读写流程(使用和不使用 Hold 线两种方式)。



#### 2. 对芯片进行±30秒调整(两种方法)。



#### 3. 初始化。





# 中环 CT100 终端常见故障维修

浙江绍兴市工商银行(312000) 徐俊

## 例一

**故障现象:**终端开机自检能通过,响“嘟”一声,但显示的字符左右晃动,不易认清,十几分钟后满屏显示光斑,喇叭“嘟”、“嘟”乱响,最后屏幕上只反复显示一条扭曲的光带。

**分析与维修:**通过交换显示器排除了显示器故障的可能性,打开终端,量电源板+5V 输出端的电压,发现只有3.7V,调整板上的电压微调电位器 WY1,使+5V 输出端的电压恢复到5.0V,终端发出一“嘟”声,屏

幕显示恢复正常,故障排除。

中环 CT100 终端的电源板因晶体管  $\beta$  值改变等因素,很容易引起输出电压下降,出现包括无图无声在内的各种故障,查故障先从电源电压查起,可以少走许多弯路。

## 例二

**故障现象:**终端自检正常,但当发出联机打印命令后,连在终端上的打印机打出的却是一些错乱的字符,并边打印边胡乱走纸。

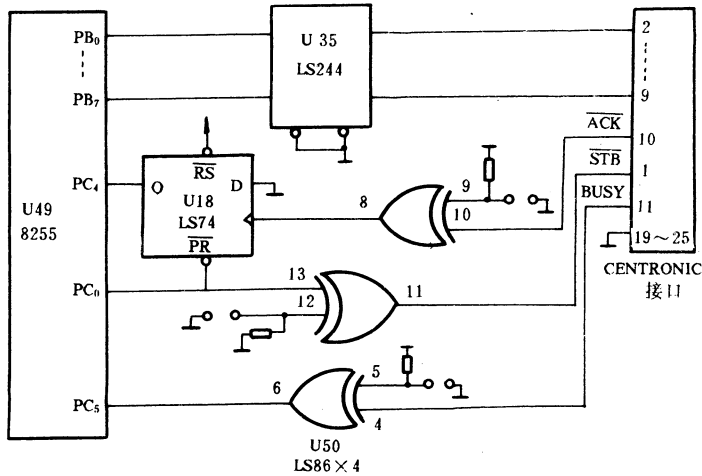


图1

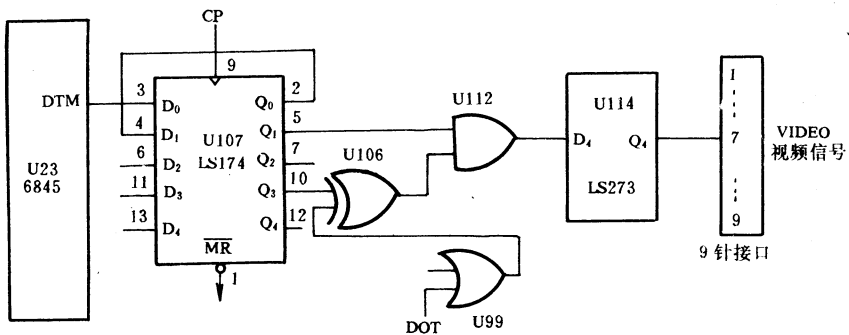


图2

**分析与维修:**通过更换打印机和打印电缆确定故障在终端内,先用交换法排除了并行接口芯片8255故

障的可能,由于终端自检能通过,则故障只可能在8255和 Centronics 接口之间的 U35 (74LS244)、U18 (74LS74)和 U50(74LS86)中,按状态键进入终端诊断测验画面,测试打印机接口,打印机在正确地打出一排“测试测试……测试”中文后,进入了打印零碎字符和胡乱走纸状态。根据测试结果分析,U35能正确发送点阵信息,出现丢数据和乱走纸的故障,问题很可能出在打印机和终端的应答线 ACK,STB 和 BUSY 上,特别是 BUSY 信号上。用脉冲信号笔在 U50(74LS86)4脚输入一脉冲信号,用逻辑笔观察6脚输出,发现输出电平恒为低,证实 U50的4,5,6间的异或门已损坏,更换 U50后故障排除。

本例充分利用了终端的自检功能,并对自检结果进行了细致的分析、推断,缩小了故障的范围,减少了维修的时间。

### 例三

故障现象:开机终端自检能通过,但正反显一律变为反显,光标消失,回扫线明显,屏幕纵向略有压缩,屏

幕右端出现部分和左端相同的内容。

分析与维修:进入终端自检画面,检测终端屏幕特性,发现光标、上划线、下划线、隐秘等功能全失效。先用交换法排除显示控制器6845损坏的可能,故障只可能出在视频输出部分。由于6845送出的允许显示信号 DTM 及光标、反显、隐秘诸信号都需经 U107 (74LS174)锁存,而后再和点信号合并输出,让终端输出全屏白画面,用示波器测6845送入 U107 3脚 D0的允许显示信号 DTM 波形,为一占空比为1:3的方波,但 U107的2脚 Q0端输出的波形却为稍有毛刺的直线,观察送入 U107 9脚的时钟信号正常,根据示波器波形分析故障系 U107(74LS174)损坏所致。更换 U107后故障排除。

本例故障现象众多,故障查找范围较广,由于在维修过程中采用了先从所有信号的公共通道查起的办法,并结合终端自检功能,最终起到了事半功倍的效果。

## 菊花遥控彩电“锁不住”故障分析

大连港香炉礁港务公司(116011) 李桂洁

菊花 FS532遥控彩色电视机采用大规模集成电路作为 CPU,自动化程度极高,其搜台操作非常简单。按一下 AUTO.PRESET 即可完成全频段的自动搜索。在调谐过程中,屏幕右上角显示>>>,同时伴有预选器号码的自动累进显示,搜索完毕,稳定在第一预选器节目信号。完成搜台后,选出的节目按先后顺序储入系统中,以待用户随时选择自己喜欢的信号。重新搜索时,新的信号陆续刷新系统中的旧信号。

故障现象:虽能完成自动搜索,但屏幕右上角的预选器数码总显示“1”,且系统不能记忆,当然不能接受任何电视信号(若原系统中曾储有信号,亦有可能接受部分电视节目)。

讨论分析:此现象俗称“锁不住”。这时的调谐电压不断地在各频段搜台,但搜到节目后却“锁不住”,屏幕上呈现的图象信号仅能稳定1~2秒。

本机采用日本三菱公司 M34300N4-584SP 为中央处理单元,其锁定电视信号依赖于

- 1、在其第⑨脚有行同步脉冲信号 H. SYNC;
- 2、在其第⑩脚有自动频率调整信号 AFT。

检修时,可先查踪 IC501(TAT698AP)第⑩脚合同步信号,一直追踪至 CPU 第⑨脚,包括晶体管 QA13(2SC1815Y)及相关电路。若外围电路无故障且无同步信号,则更换 IC501,信号正常时则更换 CPU。

上述检测正常,我们需检测 IC101(TA7680AP)⑬脚至 CPU 第⑩脚各处信号情况(包括晶体管 QA12相关电路),处理方法与上述类似。

实例:用万用表测得 IC101(TA7680)及 QA12相关电压如下

IC101管脚电压(V)

	正常值	本例值
⑬	5.0	2.8
⑭	8.2	12

QA12管脚电压(V)

	正常值	本例值
b	5.0	2.9
c	11.8	12.0
e	4.5	3.9

经查外围电路正常,断定 TA7680AP 故障,更换后,故障排除。

日立2408彩电的 CPU 亦采用 M34300N4,其检修过程可按本文思路循规分析。

## 第三章 6527 CPU 的音响系统

山东苍山机械电子化学工业局(277700) 于春

任天堂游戏中美妙动听的音响来自6527 CPU 内部的可编程音响发声器 PSG (Programmable Sound Generator)。其主要特点是:

- 五个发声声部;三个主声部,两个副声部。
- 15级音量调节
- 15种音色选择
- 15种音形包络
- 8种包络周期
- 4096种音调

以上六大特点,表明了6527CPU 音响系统的发音功能是十分强大的。只要对发声软开关设置恰当的数值,就可以模拟出各种各样的音响。大家都知道《魂斗罗》二代游戏专门设置了音乐欣赏的密码开关,只要选中密码,便可欣赏游戏中的全部音响。实际上这些效果音响我们都可以在电脑游戏机中通过设置发声软开关模仿发出。

强大的发音功能、简单的发音操作是6527CPU 的又一显著特点。

### 一. 发声软开关简介

6527CPU 的音响发声软开关位于 CPU I/O 区的 \$4000~\$4013 共20个单元,组成五个发音声部,每个声部占用四个单元。其分配顺序如下:

- 第一主声部: \$4000~\$4003
- 第二主声部: \$4004~\$4007
- 第三主声部: \$4008~\$400B
- 第一副声部: \$400C~\$400F
- 第二副声部: \$4010~\$4013

每个声部的四个软开关功能各不相同,下面以第一主声部为例介绍它们的作用。

#### (1) 音色音量控制软开关 \$4000

\$4000 是第一主声部的音色、音量控制软开关,各位功能如图12所示。

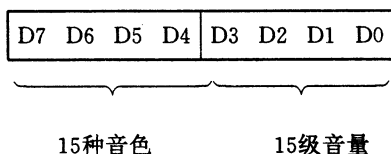


图12

- D0~D3: 为15级音量控制软开关。其值为零时关死音量;其值为1时音量最小;其值为15时音量最大。
- D4~D7: 为音色控制软开关。其中 D4=1 时发声时间延长;D5=1 时可发出各段衰减音;D6=1 时缩短发声时间。

#### (2) 包络控制软开关 \$4001

\$4001 是第一主声部音形包络控制软开关。各位功能如图13所示。

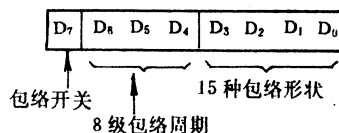


图13

- D7: 为包络开关位。D7=1 时包络有效。
- D6~D4: 为八级包络周期控制开关。调整其取值,可模拟声差极大的效果音。它们与 \$4000 的 D5 位配合可发出多级衰减音。
- D3~D0: 为15种包络形状控制开关,取值范围0~15。取值等于零和等于8时发平直音;取值等于1~7时,发声频率从高到低变化;当取值等于9~15时,发声频率从低到高;当取值等于1、9时发声时间最短;取值等于7、15时发声时间最长。

#### (3) 音调细调控制软开关 \$4002

\$4002 是第一主声部音调细调控制软开关。使用它可以细调发声音调,取值范围为0~255,取值越大发声频率越低。

#### (4) 音调粗调控制软开关 \$4003

\$4003 是第一主声部音调粗调开关。使用它可以粗调发声音调,取值范围0~15,取值越大发声频率越低。

#### (5) 发声开关组 \$4015

\$4015 是启动发声控制开关组,写入不同的数据,可选择不同的发声声部发声。各位控制功能见图14。

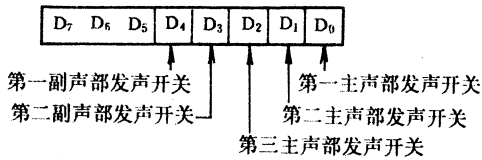


图14

根据图14,向不同位置入数值可方便地控制各声部发声。置入“1”接通开关,置入“0”关闭发声开关。其它四个声部四个单元的分工与第一主声部大体相同,读者可逐一置入数据仔细体会各单元各位的功能。

## 二. 发声方法及控制程序

### 1. 发声方法

6527CPU 的发声方法就是对不同的声部置入不同的数值。置数顺序一般为:

(1)接通发声开关,做好发声准备。一般对 \$ 4015 单元置入 # \$ FF, 接通五个声部的发声开关。若不需要哪一声部发声时,可把其对应位置入零。

(2)对包络开关单元、音调细调开关单元、音调粗调开关单元置入数据,以决定发声音调。

(3)最后对音色、音量控制开关单元置入音色音量数据,就可启动发声。这时,程序可转向,以处理其它事务,只要不对 \$ 4015 写入零或对发声声部的音量单元写入零,发声可一直持续下去。

### 2. 发声控制程序

#### (1)控制发声的原理程序

控制发声的机器语言程序见 No. 3-1

```

No. 3-1 SOUND1
LDA # $ FF
STA $ 4015 ; 打开发声开关,准备发声。
LDA # $ b1
STA $ 4001 ; 第一主声部包络数据。
LDA # $ c1
STA $ 4002 ; 第一主声部音调细调值。
LDA # $ d1
STA $ 4003 ; 第一主声部音调粗调值。
LDA # $ a1
STA $ 4000 ; 第一主声部音色、音量数据。
LDA # $ b2
STA $ 4005
LDA # $ c2
STA $ 4006
LDA # $ d2
STA $ 4007
LDA # $ a2
STA $ 4004
LDA # $ b3
..... ; 以后各声部数据
RTS

```

#### (2)F BASIC 发声程序

仅以第一主声部为例介绍在 F BASIC 状态下的

发声程序,见 No. 3-2。

```

5 REM No. 3-2 SOUND2
10 POKE &H4015,255
20 POKE &H4000,a
30 POKE &H4001,b
40 POKE &H4002,c
50 POKE &H4003,d
60 A $ =INKEY $ (0):GOTO 10

```

输入程序 No. 3-2,按表九填入 a、b、c、d、四个发声参数,就可模拟出不同的效果音。

表九 第一主声部效果音发声数据

效果音响	a	b	c	d	效果音响	a	b	c	d
被枪击中	1F	81	EF	08	八级升调 衰减音	2F	DF	EF	08
飞机降落	1F	96	EF	08	十级升调 衰减音	2F	EF	EF	08
子弹射到 金属上	1F	89	EF	08	十一级升调 衰减音	2F	FF	EF	08
射击声	1F	8B	EF	08	长升调音	3F	BF	EF	08
七宝奇谋 进小屋	1F	FA	EF	08	三级降调 衰减音	AF	F4	EF	08
警报声	FF	FF	FF	FF	五级降调 衰减音	AF	F5	EF	08
铃声	AF	C0	EF	08	十级降调 衰减音	AF	F6	EF	08
四级升调 衰减音	2F	BE	EF	08	二十级降调 衰减音	AF	F7	EF	08
六级升调 衰减音	2F	BF	EF	08	合卡节目 箭头移动声	1F	99	EF	08
七级升调 衰减音	2F	CF	EF	08	合卡节目 换页声	1F	AA	EF	08

从表九可见,仅仅更改一个发声单元的数据就可发出截然不同的效果音。因此,6527CPU 的发声控制是十分奇妙的。读者可试向各发声单元置值,定能找出更为奇妙的效果音。

#### (3)火车运行声音的模拟

使用第一副声部可以极容易地模拟出火车运行的音响。其中有火车上坡、下坡时的加速、减速声,火车钻入山洞的回声,火车过桥梁时的隆隆声,还有火车走远的渐弱声和火车开近的渐强声等,模拟得维纱维肖、十分逼真,程序见 No. 3-3。

```

5 REM No. 3-3
10 B=33
20 POKE &H4015,8

```

```

30 POKE &H400F,&H80
40 POKE &H400C,B
50 POKE &H400D,&HFF
60 POKE &H400E,A
70 PAUSE RND(100)+150:A=A+1
80 IF A=250 THEN A=RND(30)+10
90 IF A>200 THEN B=B+1
100 IF B>47 THEN B=RND(5)+33
110 GOTO 30

```

运行程序,闭目细听,我们犹如乘上列车,飞驰于云、贵铁路干线。如果再使用一个声部加入列车鸣笛的效果音,将会更进一步增强身临其境的感觉。通过 No. 3-3 可明显体会到,6527CPU 的发声系统是多么奇妙,仅用了寥寥数行语句就达到了如此神妙的效果。这在中华学习机中,其实现程序恐怕要多于数倍。

#### (4) 音乐旋律的演奏

以上仅介绍了各种效果音的发声方法,下面讨论音乐旋律的演奏。

6527CPU 可发出6个八度音域共72个音名(包括半音),因此几乎可以演奏常见的各种音乐旋律。

旋律演奏时,当音色选定后,各音符的变化控制就是改变各声部音调粗、细调单元的数据。音符与音调数据的对应关系见表十。

关于音符时值的控制(发声音长),我们不必像苹果机、中华学习机那样采用延时的方法进行控制(那样不仅降低 CPU 的工作效率,而且也无法做到声、像同步控制),而是通过设置计数器的方法实现的,一般放在中断程序中处理。由于 6527CPU 的中断频率为 50Hz,中断周期为 0.02 秒,因此,使用中断次数控制音长,可准确地控制音符的时值。在实际演奏中,音长、音量、音调各数值都是通过查表法获得的。这些内容我们将在以后章节中详细介绍。

#### (5) 各声部的特点及应用

发声系统的五个声部各有特点,使用方法也不尽相同。一般使用第一主声部作为主音演奏,使用第二主声部作为和声伴奏,或两个声部交换位置。也就是说第一、二主声部具有近似相同的特性。第三主声部的发声音调低,(同样的数据在第三主声部要比第一、二主声部低一个八度音)音符时值短,因此常用于演奏节奏、拍节,如大提琴等。两个副声部主要用于模仿各种噪声,如鼓声、锣声、钟声、脚步声、蝉鸣声等效果音。火车运行的声响就是使用了第一副声部实现的。另外各声部之间还有着密切的内在联系,只要合理搭配,就能够发出千奇百怪的奇妙音响。有兴趣的读者不妨耐心的多试几次。

以上讨论,我们大体上解释了引言中关于音响的问题。具体的音乐演奏程序和背景音乐伴奏的插入方法,将在编程特技一章介绍。

表十 音符与音调数据对照表

音符	音调数据		音符	音调数据		音符	音调数据	
	细调	粗调		细调	粗调		细调	粗调
1(7#) : : : 1# :	AE	0E	1(7#) . .	AB	09	1(7#) : : : 1#	69	08
2 : : : 2# :	F3	0D	2 . .	7C	09	2 : : : 2#	5E	08
3 : : : 4(3#) : : : 4# : : : 5 : : : 5# : : : 6 : : : 6# : : : 7 : : : 1(7#) : : : 1# : : : 2 : : : 2# : : : 3 : : : 4(3#) : : : 4# : : : 5 : : : 5# : : : 6 : : : 6# : : : 7 : : : 1(7#) : : : 1# : : : 2 : : : 2# : : : 3 : : : 4(3#) : : : 4# : : : 5 : : : 5# : : : 6 : : : 6# : : : 7 : : : 7#	0D	3 . .	52	09	3 : : : 4(3#) : : : 4# : : : 5 : : : 5# : : : 6 : : : 6# : : : 7 : : : 1(7#) : : : 1# : : : 2 : : : 2# : : : 3 : : : 4(3#) : : : 4# : : : 5 : : : 5# : : : 6 : : : 6# : : : 7 : : : 7#	09	53	08
4# : : : 5 : : : 5# : : : 6 : : : 6# : : : 7 : : : 7#	B9	0C	4# . .	2D	09	4# : : : 5 : : : 5# : : : 6 : : : 6# : : : 7 : : : 7#	4A	08
5# : : : 6 : : : 6# : : : 7 : : : 7#	35	0C	5# . .	0C	09	5# : : : 6 : : : 6# : : : 7 : : : 7#	42	08
6 : : : 6# : : : 7 : : : 7#	F8	0B	6 . .	FD	08	6 : : : 6# : : : 7 : : : 7#	3E	08
6# : : : 7 : : : 7#	BF	0B	6# . .	EE	08	6# : : : 7 : : : 7#	3A	08
7 : : : 7#	89	0B	7 . .	E1	08	7 : : : 7#	37	08
1(7#) : : : 1# : : : 2 : : : 2# : : : 3 : : : 4(3#) : : : 4# : : : 5 : : : 5# : : : 6 : : : 6# : : : 7 : : : 7#	57	0B	1(7#) . . . 1# : : : 2 : : : 2# : : : 3 : : : 4(3#) : : : 4# : : : 5 : : : 5# : : : 6 : : : 6# : : : 7 : : : 7#	D4	08	1(7#) : : : 1# : : : 2 : : : 2# : : : 3 : : : 4(3#) : : : 4# : : : 5 : : : 5# : : : 6 : : : 6# : : : 7 : : : 7#	34	08
2 : : : 2# : : : 3 : : : 4(3#) : : : 4# : : : 5 : : : 5# : : : 6 : : : 6# : : : 7 : : : 7#	F9	0A	2 . .	BD	08	2 : : : 2# : : : 3 : : : 4(3#) : : : 4# : : : 5 : : : 5# : : : 6 : : : 6# : : : 7 : : : 7#	2E	08
3 : : : 4(3#) : : : 4# : : : 5 : : : 5# : : : 6 : : : 6# : : : 7 : : : 7#	A6	0A	3 . .	A8	08	3 : : : 4(3#) : : : 4# : : : 5 : : : 5# : : : 6 : : : 6# : : : 7 : : : 7#	29	08
4(3#) : : : 4# : : : 5 : : : 5# : : : 6 : : : 6# : : : 7 : : : 7#	80	0A	4(3#) . . . 4# : : : 5 : : : 5# : : : 6 : : : 6# : : : 7 : : : 7#	9F	08	4(3#) : : : 4# : : : 5 : : : 5# : : : 6 : : : 6# : : : 7 : : : 7#	27	08
4# : : : 5 : : : 5# : : : 6 : : : 6# : : : 7 : : : 7#	5C	0A	4# . .	96	08	4# : : : 5 : : : 5# : : : 6 : : : 6# : : : 7 : : : 7#	24	08
5 : : : 5# : : : 6 : : : 6# : : : 7 : : : 7#	3A	0A	5 . .	8D	08	5 : : : 5# : : : 6 : : : 6# : : : 7 : : : 7#	22	08
5# : : : 6 : : : 6# : : : 7 : : : 7#	1A	0A	5# . .	85	08	5# : : : 6 : : : 6# : : : 7 : : : 7#	20	08
6 : : : 6# : : : 7 : : : 7#	FC	09	6 . .	7E	08	6 : : : 6# : : : 7 : : : 7#	1E	08
6# : : : 7 : : : 7#	DF	09	6# . .	76	08	6# : : : 7 : : : 7#	1C	08
7 : : : 7#	C4	09	7 . .	70	08	7 : : : 7#	1B	08



# 基于图形的联想多媒体工具 GBH

山东聊城师范学院光通信研究所(252059) 白成林

## 一、概述

联想多媒体技术是近年来世界上流行的新技术。它按人脑的联想思维方式非线性(非顺序)地存储和管理文字、图形、图象、声音等各种媒体信息。它的基本组成是节点和链,节点用于存储信息,链描述节点之间的关系。使用联想多媒体系统,用户可以按自己的观点组织和浏览系统的信息,而且控制系统的运行,自由、灵活。

GBH 则是为加速开发联想多媒体系统而研制的一套工具软件,它以图形为基础,可非线性地存储、管理、浏览图形、图象、声音、文字、可执行模块、表格。该软件在北京已通过了来自中国科技大学研究生院、北京大学、中国科学院、清华大学、中国人民大学等单位的专家鉴定,认为 GBH 项目的开发和研制是成功的,它的技术研究和软件实施居国内领先水平,进入了世界 Hypermedia 技术研究与开发的先进行列。任何用户使用它,无需编程,只需用这个工具采集、编辑各种信息,说明信息间关系(包括用图形描述),即可建成实用的系统。用户可按需要随时修改、扩充应用系统。系统中的信息可以最大程度地保留它的自然面目,GBH 对它们的格式、体积等等都没有规定。系统建立后,可随意浏览,没有操作上的限制。

对于 GBH,每个应用系统的基础是两种图:网状图和分布图。网状图用来描述应用领域中各种关系。分布图用来描述各种分布情况。而网状图的节点和分布图的标志还可以又是网状图,又是分布图,以此类推,形成图形网。

这些分布图和网状图表示应用系统概貌、概念性信息以及信息之间的关系,而图上的节点/标志的予成分:文字说明、图形、图象、声音、可执行模块、表格及表格项相关的信息,则给出细节描述。在 GBH 中,用户可同时浏览概貌、概念及细节,既见树木,也见森林。此外,概貌、细节都有图形表示,内容丰富,一目了然,有助于理解、学习和记忆。

## 二、GBH 的运行环境和主要特点:

GBH 是在微型计算机的 Windows 3.0 系统下实施的,它以面向对象的风格组织命令按钮。在 GBH 中,90%的操作只需按鼠标按钮。其硬件运行环境为:

- 386、486、PS/2 及兼容机或工作站
- 4M 内存
- 80287/80387 协处理器
- 60M 硬盘或 CD-ROM 光盘
- 图形卡(CGA、EGA、VGA 等)

• 鼠标(Microsoft、Mouse 等)

主要特点:

- 1) 非线性存储及管理信息,用户可以按自己的思维方式组织或浏览信息。
- 2) GBH 管理的信息丰富,包括文字、图形、图象、声音,可执行程序、表格。
- 3) GBH 以网状图和分布图为基础,组织连接各种相关信息,直观、信息量大。
- 4) 用户界面友好,操作简便。

## 三、GBH 的主要功能

GBH 是一套系列软件,它提供了十大类编辑、管理、浏览功能:

- 1) 网状图编辑 图形节点的增、删、移动,用户自定义,链的增、删、注解、标签和图形节点名的编辑、图的移动和缩微、画布的放大缩小。其中画节点和移动节点的过程是可见的,并有网格帮助定位。
- 2) 分布图编辑 标志的增、删、命名、属性显示/编辑、用户自定义;分布图的装入、存储、叠加、删除、激活等。
- 3) 关系编辑器 建立、解除、查看图形节点/标志和它们的成分的关系。
- 4) 背景及图片编辑 彩色、多灰度、单色图扫描;手工绘图(点、线、圆、矩形、曲线……);图片显示、放大/缩小、旋转、剪裁(上、下、左、右)、拼图、区域清除、区域存储、填色。
- 5) 文字编辑 文字插入、删除、修改、拷贝。
- 6) 声音管理 录音、放声、列清单。
- 7) 可执行模块管理 建立、删除、查看图形节点/标志和可执行模块的关系,运行与图形节点/标志有关的程序。
- 8) 表格结构及表格的编辑 表格结构的建立、编辑,表格数据的输入、编辑、分页浏览(其中表格数据的类型可以是字符、数值、图象、声音);相关图象的显示及相关声音的播放;表格条件查询;根据表格数据生成直方图、扇形图;由表格生成表格。
- 9) 浏览 网状图、分布图、图形节点、标志、背景、各种予成分的浏览。对每种对象分别提供若干种浏览方式。
- 10) 颜色管理 画布、系统各窗口前(背)景颜色以及节点、标志颜色的自由选择。
- 11) 查询 除表格查询外,在任何状态下均可根据某个予字符串,查询整个系统的有关图、标志或节点,列出清单,进而查看其中的图、节点或标志。

#### 4. GBH 的应用领域:

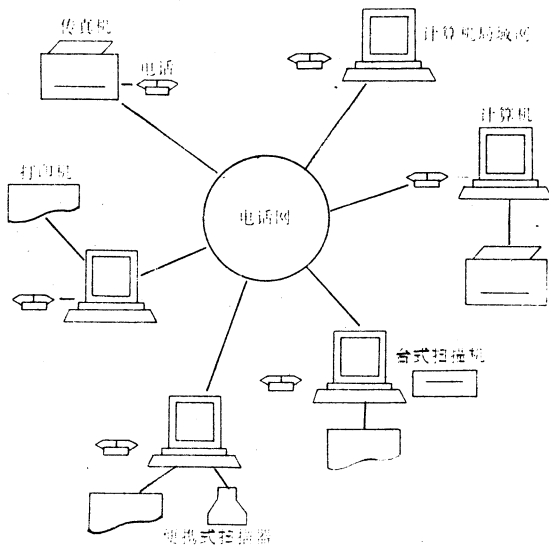
各种分类信息管理(如学科分类、组织机构等);各种分布信息管理(如动植物分布、民族分布、机构分布等);新闻、报刊、娱乐业(电子旅游指南、电子展览馆);商业(各类修理指南、电子广告系统等);百科全书,电

子写作系统,图形专家系统,软件工程工具(软件开发过程及其各种图形、文档程序管理);数据库系统如档案(文字、图片、声音等)分类管理;CAD 系统管理教学系统(如音像教学等);办公系统,制造业等等。

## 微机——传真机系统

刘保元 张景生

微机——传真机系统,又简称为 PC-FAX,是现代化的综合性自动化办公系统。它通常由传真机、微型计算机、打印机、扫描仪(器)、电话网等构成,其信息网络结构如下图所示:



下面举三个典型的实例:

#### 一、TURBO-FAX 系统

TURBO-FAX 是美国 VACOM 公司与机械电子工业部经济信息中心合作推出的产品。在国际众多类似产品中, TURBO-FAX 以其优越的性能名列前茅。

TURBO-FAX 是一个能够支持个人计算机进行数字和图像传真通信的硬件卡和一组软件。利用现有的电话支持,把您的 IBM PC/XT、AT、286、386 及其兼容机等计算机连接为具有数据通信、图像传真功能的信息系统。

目前国内传真机系统的应用已相当广泛,但只用于公文、图表等图像传真。而需要收集数据并用于计算机处理的部门往往是采用将图表等数据用图像传真方

式接收过来,然后进行数据再录入后用于计算机处理,即使应用调制解调器的系统,由于其在国内线路条件下传输速度的限制,也往往无法达到实用水平。如果应用 TURBO-FAX 传真系统则既可完全代替传真机系统并使传真电脑化,又可代替调制解调器进行数据通信,并且在同样线路条件下,其传输速度、传送质量和可靠性大大高于调制解调器,同时为节省传送时间,能将传送数据压缩最高达 80%,一张 360K 软盘的数据量传输时间只需几分钟。TURBO-FAX 系统可广泛用于远程排版印刷系统、远程数据收集统计和处理系统、远程软件程序的传递等领域。

#### 1. TURBO-FAX 的主要特点

- 后台操作 在等待传真时,允许转向 DOS 系统,进行其他工作。
- 电话簿 可储存 16000 个电话号码(20 个电话簿 × 800 个号码)。
- 自动定时传送 可设立 20 个邮箱,每个邮箱可含 800 个电话号码及相应待发送文件名,每个邮箱给一个开启发送时间。
- 多文件传送可自动对某一站点同时发送多个文件。
- 文本编辑 具有全屏编辑功能并可立即传送屏幕内容,用户可根据需要选用外部编辑软件。
- 一些自动功能 具有自动应答、自动为单位名称打印、为日期和时间打印、自动拨号、自动延迟传送、自动进行文件转换、自动存入、自动登记、自动打印、自动重复拨号等功能。
- 自动电源开关 信息实现自动无人值守接收。传真到来时,自动电源开关能够自动将关闭的计算机启动并接收传真,之后自动关闭计算机。
- 支持 PC-传真机 在与计算机一起应用时,能够把传真机当作扫描器及输出拷贝机使用。
- 快速拨号 可以对 400 个常用的电话号进行快速拨号(共 20 个电话簿,每个电话簿可选 20 个号码)。
- 本地区代码和时间表 查询国际传真的地区代码和当地时间。
- 文本行宽度压缩 能够传真最多可达每行 200 个字符的文本。
- 任意文件传送 能够把 DOS 环境下的任何文件(数据、目标码文件、字符文件、图像文件等)进行快速

传送。

- 文件的压缩和加密传送 可将传送的文件压缩最高达80%，大大节省传送时间。并提供了带口令数据加密。接收方可以方便地解开压缩的数据。
- 探询发送与接收 接收方能够向发送方发出探询呼叫，通知其把准备好的文件传送过来。发送方也能够确认来自接收方的呼叫，然后自动将准备好的文件发送给对方。
- 通话发送与接收 发送与接收方如果采用自动拨号方式通信，信息传送结束后，双方可以利用电话进行直接通话。
- 扫描并传送 可配台式扫描器，实现扫描后立即传送。
- 字符与图像文件合并 能够将字符文件与图像文件合并，字符文件由系统自动转换为图像方式。

### 2. TURBO-FAX 支持

- 兼容性：支持所有的 CCITT 的 G III 传真机。
- 自动调整传输速度：自动操作速度为 9600/7200/4800/2400bps。
- 支持 Canon, HP, Microtek, DFI, Logitech 等扫描器。
- 支持 ASCII 格式文件，包括 Lotus 1-2-3, dBASE III Plus, 个人编辑程序和文字处理程序。
- 支持多种格式的图形文件。
- 支持多种9针, 24针及激光打印机。

### 3. TURBO-FAX 的操作环境

- 适用 IBM PC/XT/286/386。
- 适用 CGA, EGA, VGA, 屏幕或 Hercules 图形适配器。
- 适用标准模拟(旋转式或按键式)电话线路。
- 英文版本适用 DOS 2.0或2.0以上版本。
- 中文版本适用 DOS 3.0以上版本。

### 二、POLY FAX“数据——传真通信系统”简介

POLY FAX(“保利传真”)是由中日传真机维修站开发的“数据——传真通信系统”。本系统立足于国情，凡是装有 PC 或 0520 系列机的单位，花上几千元的成本就可实现数万元的传真机的所有功能，并且在报文存档、文件管理等方面，实现传真机所不能实现的功能。

#### 1. 发送功能

- (1)人工拨号发送；
- (2)立即自动拨号发送；
- (3)定时自动拨号发送(按用户编制的“时刻表”发送)；
- (4)同文件，多地址自动发送；
- (5)自动重新拨号发送。

#### 2. 接收功能

- (1)人工接收(存盘、打印)；
- (2)自动接收(存盘、打印)；
- (3)后台接收(存盘)。

3. 数据文件远程传输功能：采用 HDLC 规程，在两台微机之间实现数据文件的远程拷贝，速率高达

9600bps。

#### 4. 传真文件的显示、打印功能。

5. 收、发日志和报文管理功能：可对已收、发的文件进行查询、显示、打印、删除等操作，这是任何传真机都不能实现的功能。

6. 扫描功能：可将图片、文件、手迹、签名、印章等扫描入计算机，实现真迹传真。

7. 中文文本到传真图像的转换功能：可将汉字 WordStar 产生的文本文件转换成能进行传真通信的 MH 码的图像文件。

8. POLY FAX 与国内类似产品比较有如下特点。

#### (1)全汉化

(2)高分辨率图像(450×640, 350×640, 400×640 彩色)，可直接从屏幕上读报文。

(3)多种打印机驱动程序，可面向 P7, 1724, 3070, FX-100, 8320 等打印机打印图像报文，也可配带激光打印机快速输出报文。

(4)将由键盘输入的中文文件转换成传真图像文件，省去了目前多数单位将中文文件用打印机打印出报文再用传真机发送的环节，节省了时间，提高了传输质量和效率。

#### 三、银斯达微机传真系统

这套系统适应任何传真机，与 IBM PC/XT/AT 及兼容机相配合实现了由微机、传真机、各种扫描器组成的管理、传真办公系统。任何人都可自己安装好是它特色之一。

#### 需要的硬件设备：

1. IBM PC/XT/AT 或其兼容机一台
2. 传真机一台
3. 打印机一台(不使用打印机打印传真文件时，可以不选)

#### 4. 银斯达微机传真系统(包括接口卡及软件)

#### 功能介绍：

#### (一)本地功能：

1. 传真机作为微机的精密打印机使用，微机的文件可在传真机上输出。

2. 传真机作为微机的文件、图象扫描仪使用。

#### (二)远程功能：

装上这个系统后，传真机的所有通信功能都可在微机上完成，而且不影响微机的正常操作。这些功能主要是：自动收/发、存储及打印；定时收/发；多点轮询；电话/传真自动转换；按电话簿自动拨号；图象质量级选择；通信管理报告打印等。

1. 可以实现我方和对方的微机与微机、微机与传真机、传真机与传真机之间的任意通信，计算机文件、手写文件、图象图画都可以互相收/发。

2. 可以实现一个文件多点发送，可以由微机自动把文件逐个发给几百个单位。也可以实现不同文件给不同的单位。

3. 上一条的一文件多发及多文件多发都可以设定

定时发送,可以在晚上进行发送,避开电话线路高峰期。

4. 接收的对方传真或数据可以存到磁盘上,方便打印及编辑。

5. 编辑功能很强,可以对图象裁剪、合并、加文字注释、翻转存盘、打印等。

6. 后台操作功能,当微机运行传真功能的同时可以运行其它软件;当有传真进来时,微机会提示。

7. 具有打印截获功能,输出到打印机上的文件可直接改为传真文件发送,支持 CTRL-P,全后台操作。

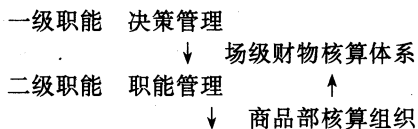
8. 软件内配有与各种排版系统相沟通的程序,便于软硬件的相容。能使用各种扫描仪进行工作。

## 大连华联商厦的计算机网络

大连东升电子有限公司 廖 涛

大连华联商厦是一个大型现代化商业企业,经营商品种类达五万余种,年销售额在人民币三亿元以上,交易频繁,销售手段灵活。

整个商场实行三级管理两级核算的管理体系,如下图:



三级职能 部级管理

经营方式主要有经销、联销、代销。

售货方式有自选、布点式集中收款、传统柜台收款(即一手钱,一手货)。

结算方式有人民币、长城卡、牡丹卡、支票、外汇卷、内转支票、外地汇款等。

由于整个商场收款台点散且多,70多个收款台点分布在八层的2万多平方米的营业楼层中,并且由于商业活动的季节性变化,整个前台收款终端的规模与位置在一定范围内波动。同时,要求商品编码、调价行为不受限制,首先是商品编码达到数万种,这对流通周期短的商店是很必要的。例如金行,它进货种类较多,并且销货周期短,这就要求随时可以增加数以千计的商品编码,并保证实时传送到各个销售点。调价、编码行为都不得影响与中断前台销售,因此系统对网络结构的设计以及网络的选型提出了特定的要求。

针对前台销售手段灵活(有十种以上的异常销售方式,如布头折让、拆零、折旧、优惠、附加税以及各种非营业性收支)、数据采集面广等特点,我们采用了POS终端机(80286 CPU)来作为前台收款设备,利用POS机本身较强的处理能力来实现上述要求,并且通过直接上网的方式,实现高度的数据共享与真正的实时性。

整个大厦的计算机管理系统从功能上分成前台销

售、物价管理、财务、合同、仓储、统计、劳资人事、辅助决策、代码维护等十个子系统。这十个子系统几乎覆盖了一个商业企业的全部业务,形成一个从计划、采购、销售、统计到计划检查的闭环的动态计算机管理系统。

其中前台销售子系统采集每日商场产生的零售业务的数据,它是整个管理系统的重点与基础。它所采集的数据的范围与准确性、实时性等将直接决定整个系统的性能好坏。其它各子系统均使用到它所采集的原始数据。

而商品帐子系统则是全系统的核心,各种数据、凭证在这里汇总、加工,再进行传递与交换,在整个POS网络管理系统中具有承上启下的枢纽作用。

辅助决策子系统则是整个POS系统的目的,它提供各种分析报表与参考数据,供决策者参考。

考虑到整个大厦的数据流量较大,计算机任务繁重,不宜采用传统的集中处理,即主机——终端方式。因为当终端数目较多时(并且在一定范围波动),首先是通信成本急剧增加,此外,主机不但要负责处理每个终端提出的任务,而且要管理与它们的通信工作,随着终端数目的增加,通信控制之类的杂务必将占用大量的主机时间,使之无法集中较多的时间来进行数据处理,因而考虑采用计算机微机局域网,利用网络的高速传输能力(可达2MB~10MB),将部分计算任务转移到各个POS终端机中,充分利用POS终端机本身的计算能力与资源,以均衡整个网络系统的负荷,实现一种分布式的数据处理环境,使整个网络的效率得以提高,同时由于利用了POS终端机的本身资源,在一定范围内增加POS终端机的数目,不会给整个网络带来大的负担。

整个网络系统选用微机局域网,采用树状网络结构。网络分为二级即商品部子网与商场主网。

商品部子网有七套树状网构成,每套子网设有一个文件服务器,用来收集处理各个POS终端机采集的

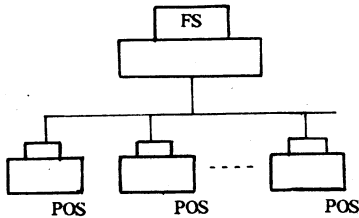
数据,同时也为各个二级职能系统提供服务,POS机与服务器之间通过HUB相连。这种方式有如下特点:

①由于商场环境复杂,一旦一个POS机出现故障,其它POS机及网络服务器将不会受到影响,使网络的可靠性在拓扑结构上得到保证。

②网络扩充能力强,每个HUB有多个端口,可方便地增减或移动收款台点。

商场主网,设有一台主服务器,负责收集处理各个商品部子网的汇总数据,同时,还为二、三级职能系统提供服务。

主网与子网间通过桥连接(如图)。



这样建立起来的网络系统,将一个庞大复杂的数据处理系统均衡地分布到8套既独立又相差的网络之中,每套子网不直接与主网发生联系,但又保留了可联结性,因而使各个子网的效率得到提高,同时也杜绝了网络间的越权查询。

我们可以看到这样的网络设计体现了三级管理两级核算的管理方式。

(上接49页)

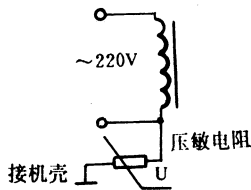
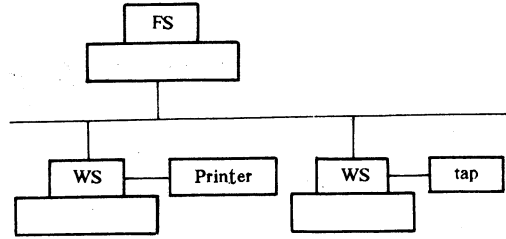


图5 静电放电保护原理图

继电器触点也要在吸、放时产生电火花,会干扰COP840C的正常工作。因此在触点两端并接一个阻容回路,电阻值为 $100\Omega$ ,电容器的容量为 $0.1\mu\text{F}$ 、耐压 $630\text{V}$ ,以形成一个电火花释放回路。

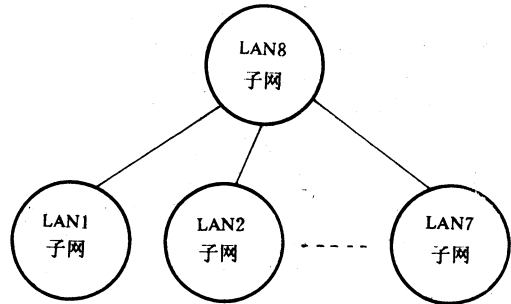
采取以上一系列抗干扰措施以后,大大加强了本洗碗机的工作稳定性和可靠性。

本设计在全自动洗碗机上实际使用证明,整机技术指标及性能均达到设计和使用要求。用户反映可选功能丰富、表面指示直观清晰、操作简便、硬件的保护能力强,适合于电网波动大及潮、热等恶劣环境下工作。这都是由于采用了NS公司的高效能、低价格COP800C系列芯片而得。这说明COP800系列芯片确实适合家电产品自动控制的应用。



网络今后与外界如银行、分店、厂家、客户等的连网可随时进行。

华联商厦的这套网络系统运行已近两年,性能较稳定,整个大厦基本上已全部采用计算机收款,数据采集齐而且准,并可随时打印出各类报表、统计报表以及辅助信息报表,并为二、三级职能系统提供了实时的数据来源。



(上接51页)

接着发现该文件所在子目录中有一个名为“CHK-LIST.CPS”的文件,因而怀疑该目录中的文件被CPAV免疫过。运行CPAV撤除对该文件的免疫,再用公安部KILL V62.00查,发现该文件被“新世纪”病毒(NEW CENTURY)感染。消毒后,再运行该文件,VSAFE不再报警。

因此,有必要提醒电脑用户,不要迷信“免疫”。其次,要采用“安全免疫法”,其步骤如下:

- 1)用新版CPAV撤除要加免疫的文件的免疫功能,(如果不知道该文件是否已被旧版本的CAPV免疫过的话)
- 2)分别用公安部KILL(62.00以上版本),美国SCAN(95以上版本),CPAV(1.2版以上版本)查毒。
- 3)再用新版CPAV加免疫(注意,若被免疫文件不能运行则需撤去免疫)。

另外,CPAV提供的VSAFE.COM和BOOTS SAFE.EXE对防止病毒入侵很有用,建议用户在自己的AUTOEXEC.BAT文件中加上如下语句

PATH C:\KILL(假设上述两文件在C盘KILL子目录下)

VSAFE /8+  
BOOTS SAFE C;



# COP840C 芯片在洗碗机自动控制中应用

王求是 季行健

## (1) 概述

家用全自动洗碗机由装在洗碗机内的 NS 公司产品 COP840C 组成的自动控制器,实现对洗碗机的加热管、清洗泵、排水泵、洗涤剂添加阀、进水阀以及电源开关(包括门开关)等机电部件的自动控制,来完成预定的洗碗过程的不同洗涤方式、烘干、报警等一系列功能。具体功能如下:

- 全自动洗碗机有5种洗涤方式:强力洗、普通洗、经济洗、漂洗、漂清。每种洗涤方式都有其自己固定的流程,用户可根据情况任选一种。

- 有4种异常报警功能:异常水位报警、排水故障报警、进水故障报警、超温报警。

- 有7种烘干时间:30分钟、45分钟、60分钟、75分钟、90分钟、105分钟、135分钟。

## (2) 全自动洗碗机的工作过程

接通电源开关时,机内控制电路约定的状态为:洗涤方式按普通洗,烘干时间为30分钟。此时用户可通过按键 AN2选择改变约定的洗涤方式和通过 AN3按键选择改变约定的烘干时间,然后按启动键 AN1,洗碗机就按设置的洗涤流程进行洗涤和烘干。当全流程结束时,蜂鸣器会提醒用户,并自动切断电源。在执行全流程过程中,洗碗机内水位出现异常时(超过正常水位),控制就进入异常水位报警程序;当洗碗机不能在规定的时间内排净洗涤水时,控制就进入排水故障报警

程序;在规定时间内洗碗机中进水不能达到正常的工作水位时,控制就进入故障报警程序;当洗碗机失控时,机内温度不断上升,要导致洗碗机损坏的危险时,控制就会自动切断电源。由于采取了一系列保护措施,因此本洗碗机使用安全可靠。

## 2 COP840C 芯片控制器原理

图1为芯片控制全自动洗碗机的控制原理图。它是由 COP840C 芯片为主控器件加上键盘显示、检测、控制等部分组成。COP840C 有4个输入口(I口)引脚、4个输出口(D口)引脚、8个可编程输入输出(L口)引脚、8个复用功能口(G口)引脚。G0、G1、G2、及 D0、D1、D2为9个 LED 提供位扫描及段码信号,以点亮相对应的一个 LED。G0、G1、G2及 I3为 CPU 提供键盘信号。图中所示检测输入信号共有4个:G3、G5分别为水位异常、水位检测信号输入端。当水位异常或进水到位时,传感器的触点开关动作,使 G3、G5引脚产生电平跳变,被 CPU 接收。G4为门开关输入信号端,当用户在洗涤中途打开洗碗机门时,G4口由高电平变成低电平,洗碗机立即停止运行。I0、I1、I2为三档温度比较值的输入口,机内温度经过热敏电阻感温,变成电信号,然后与三个不同量值电压的比较器(LM339)比较,在 I0、I1、I2三个端口上产生不同的电平,便于 CPU 控制温度变化及超温报警。

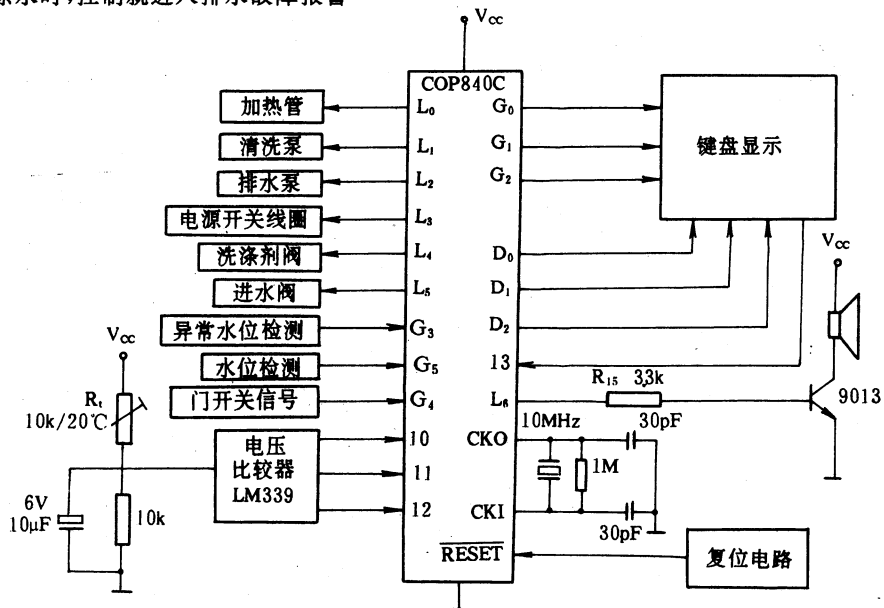


图1 全自动洗碗机芯片控制器原理图

输出控制量共7个,L1~L5分别通过可控硅控制清洗泵电机、排水泵电机、电源开关线圈、洗涤剂供料阀、进水阀。L0通过三极管9013控制继电器通断,使加热管增温或减温,L6通过三极管9013控制蜂鸣器的声响。

### 3 应用软件设计

软件程序由下列程序组成:

- 初始化程序(略)
- 主程序(流程图见图2)
- 清洗子程序(见图3)
- 烘干子程序(略)

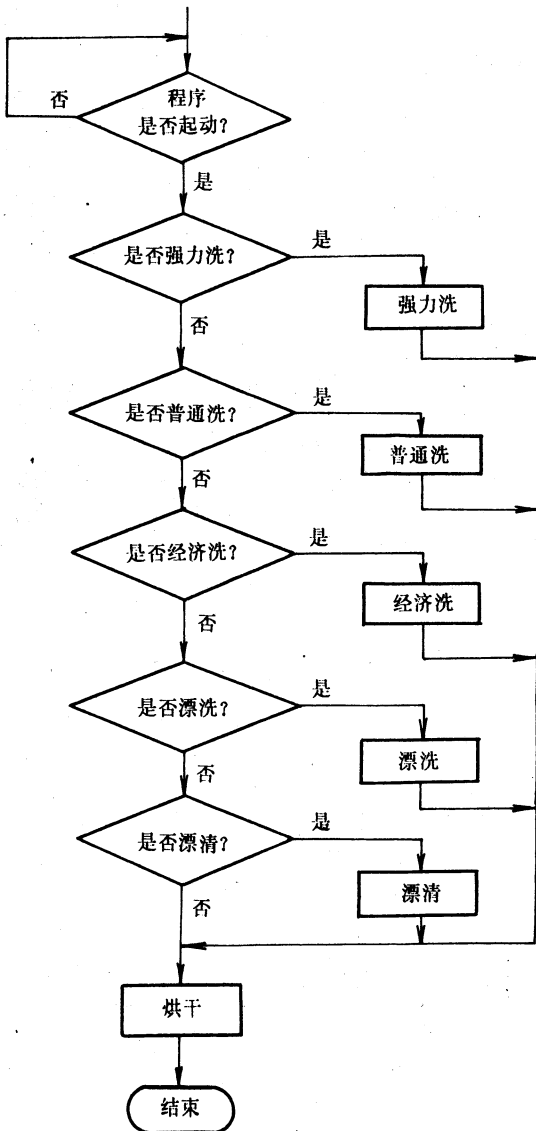


图2 主程序流程图框图

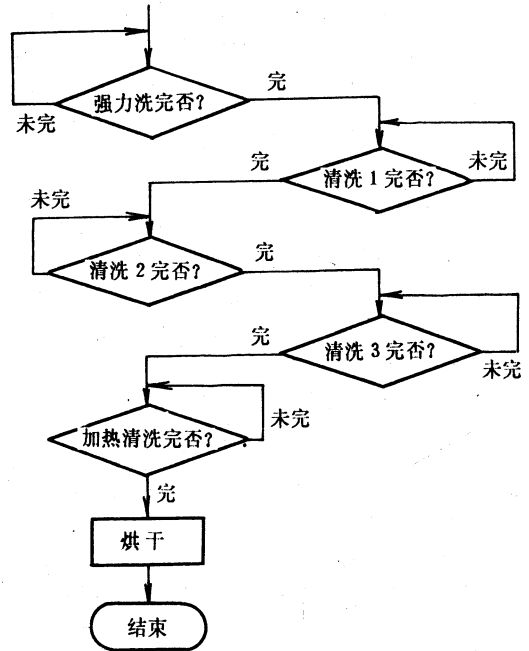


图3

### 4 应用中的其他问题

应用中的其他问题主要是抗干扰问题。由于洗碗机是在加热、潮湿、水流旋转摩擦产生高压静电的恶劣环境下工作,因此对芯片控制板的稳定可靠有更高的要求,尤其要对输入信号端采取抗干扰保护措施。G3、G5两口分别为水位异常、水位检测信号输入口,其触点开关易产生干扰信号,因此接入吸收回路——如图4的C<sub>1</sub>、D<sub>1</sub>和D<sub>3</sub>,使干扰信号被充分吸收。

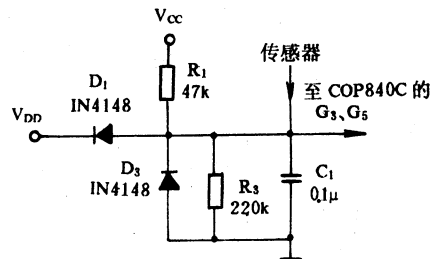


图4 抗干扰电路

由于水与金属内壁的摩擦产生上万伏的静电高压,易击穿芯片COP840C,因此,在外壳与220V电源一端接一只1800V的压敏电阻,形成一个静电释放回路(见图5)。(下转47页)

# 乘法器 IC 及应用

李兰友

(接上期)

## 3 100MHz 带宽的有效值—直流转换器

RMS—DC 转换器可以用专用集成电路来实现,但专用集成电路的频带宽度大多在10KHz~1MHz 左右,更高频率的有效值—直流转换器,例如在100MHz 以上时,则可使用乘法器 AD834实现之。

使用 AD834构成的有效值—直流转换器如图7所示。

图中,AD834接成平方运算方式。输入电压  $V_{IN}$  加到 AD834的输入端 X2和 Y1,同时,输出电压反馈到输入端 X1和 Y2。X1的输入为  $V_{OUT}$ , Y2的输入为  $-V_{OUT}$ 。因此,AD834对  $(V_{IN}+V_{OUT})(V_{IN}-V_{OUT})$  进行计算,输出电压由 IC2进行比较,当 AD834输出电压为零时则平衡,这时。

$$(V_{IN}+V_{OUT})(V_{IN}-V_{OUT})=0$$

$$V_{OUT} = \sqrt{V_{IN}^2}$$

输出为有效值电压。

图中,SMA 端子输入电压  $V'_{IN} = 1V_{rms}$ ,经  $R_1$ 和  $R_2$ 分压,AD834的输入为  $0.5V_{rms}$ ,同时,  $V_{OUT}$ 经  $R_5$ 、 $R_6$ 分压,使输入电压和输出电压一一对应,即  $V'_{IN} = 1V_{rms}$ 时,  $V_{OUT} = 1V_{DC}$ 。

电位器  $VR_2$ 用做零点调整,  $VR_1$ 调整 IC2增益,  $D_3$ 为 AD834的保护二极管。

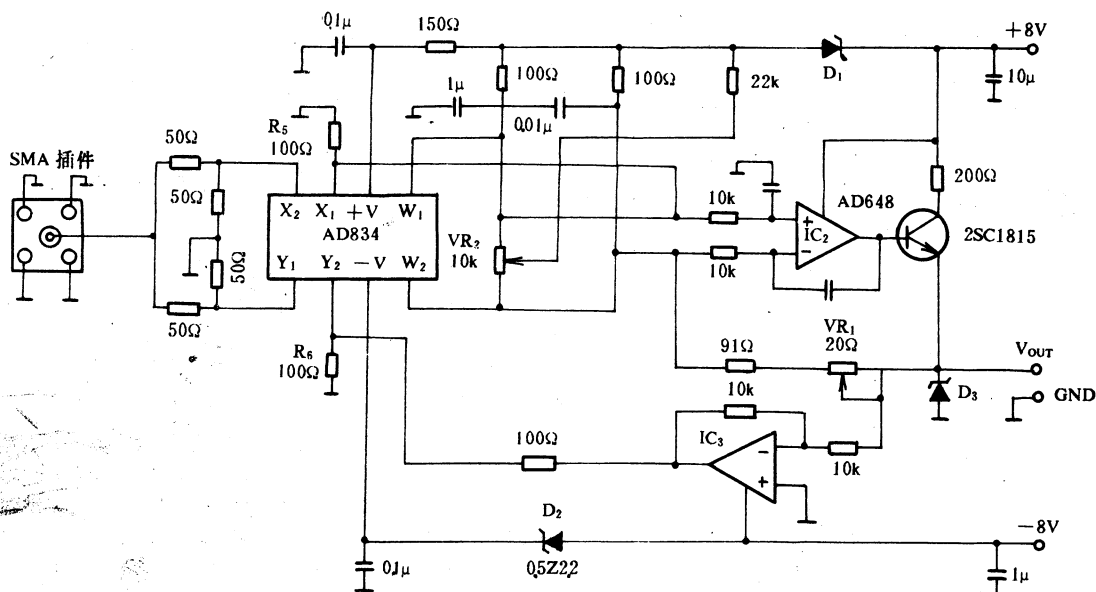


图7 RMS—DC 变换电路

图8为该电路的实测频率特性。本电路可工作到200MHz~300MHz。转换特性如图9所示

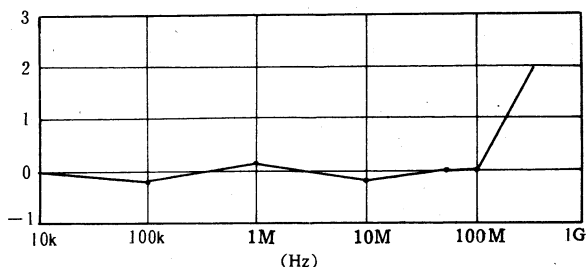


图8 实测频率特性

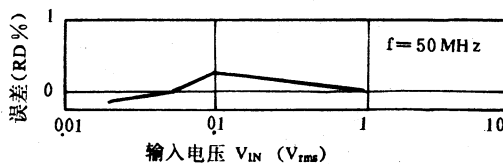


图9 转换特性

#### 4. 热电偶线性补偿

利用 AD538 的平方特性进行热电偶线性补偿的电路如图 10 所示。

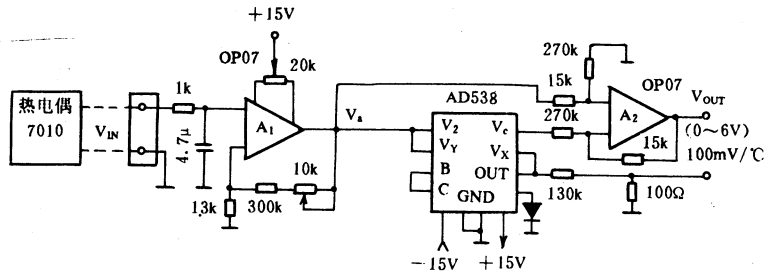


图 10 AD538 的热电偶线性补偿

本电路测温范围为 0~600℃, 输出电压  $V_{out}$  范围为 0~6V。热电偶采用 K 型热电偶, 运算放大器  $A_2$  (ADOP07) 和 AD538 组成线性补偿电路, 运算放大器  $A_1$  输出电压  $V_a = 249.952V_{IN}$ , 调节电位器 VR1 可改变其放大倍数。输出电压  $V_{out}$  可表示为:

$$V_{OUT} = -7.76 + (249.952V_{IN}) - 5.56 \times 10^{-6} (249.952V_{IN})^2$$

$$= -7.76 + V_a - 5.56 \times 10^{-6} V_a^2$$

AD538 的传递函数为

$$V_o = V_i^2 / 10000mV$$

这样,

$$V_{OUT} = -7.76 + V_a - 0.0556V_a^2 (mV)$$

平方项系数由  $R_2$  和  $R_4$  决定,  $R_4/R_2 = 0.0556$ , 选择  $R_4 = 15K\Omega$ ,  $R_2 = 270K\Omega$ 。一次项系数取 1, 即

$$(1 + R_4/R_2)R_3 / (R_1 + R_3) = 1$$

可选择  $R_1 = 15K\Omega$ ,  $R_3 = 270K\Omega$ 。

## 经验数则

兰州军区总医院器械科(730050) 曹彤

### 一、在任何路径下均可使用 WS

建立 WS 子目, 设置 PATH C:\WS, 由于执行尚需要部分 OVL 文件, PATH 仅使 DOS 寻找扩展名为 COM、EXE、BAT 的文件, 因此, WS 找不到 OVL 文件, 会出错, 在 3.2 以上版本的 DOS 中包含一个 APPEND 的外部命令, 执行 APPEND C:\WS 后, 在任何路径下均可使用 WS 了。

### 二、加速 WPS

WPS 软件应配汉卡运行, 在无汉卡时, 每次调用汉字均会到硬盘读字库, 由于频繁读硬盘, 而且硬盘与内存之间的传输率较低, 导致 WSP 换屏速度缓慢。

《电子与电脑》93 年第 7 期刊登了使 WPS 加速的方法, 其实, 这种加速方法与 WINDOW 无关, 完全是由于在 CONFIG.SYS 文件中利用 SMARTDRV.SYS 设置了磁盘高速缓存的缘故。

磁盘高速缓存是在 RAM 中储存了最近访问的磁盘扇区的信息, 每次需读盘时, 系统总是首先检查磁盘高速缓存中是否有需要信息, 如果有, 即免去读盘操作, 从高速缓存中将所需数据传至内存, 缓存满时, 最长时间未用到的数据被替换, 因此经常访问的数据被保存在高速缓存中, 由于减少了访问硬盘数据的时间, 内存间数据传送速度快, 提高了机器性能。

在 CONFIG.SYS 文件中使用 SMARTDRV.SYS 或 ASTCACHE.SYS 等软件在扩展或扩充内存中建立高速缓存, 或在 WPS.BAT 文件中利用 PC Tools 软件包中提供的 PC-CACHE.EXE 软件在运行 WPS 时建立磁盘高速缓存。启动 WPS 后, 系统将字库读至缓存中, 由于不需要频繁读盘, 使 WPS 加速。

### 三、DIR-I 病毒清除方法

《电子与电脑》93 年第 3 期介绍了一种清除 DIR-I 病毒的方法, 本人通过另一种简易方法也将病毒清除。即在带毒环境下, 拷贝所有的可执行文件为扩展名不为 EXE 或 COM 文件, 再删除所有可执行文件, 用干净系统启动后, 将文件名改回成原文件名即可。

## 清毒软件 CPAV 的局限性

北京 张志

CPAV 是著名的消毒软件, 并且还有独特的对可执行文件进行免疫 (Immunize)。但通过实际应用, 发现两个问题。

一、有些可执行文件加免疫后不能运行, 例如 PC-SHELL 7.0 中的某些可执行文件, 加免疫后再运行该文件则会死机, 撤除免疫后恢复正常。

二、若文件被病毒感染后没被 CPAV 查出, 并被 CPAV 免疫, 则该病毒会被保护起来。

笔者就碰到过这一情况。在执行一个可执行文件时, 驻机的 VSAFE 程序报警, 指出该程序要写硬盘引导区。选择 STOP 项后, 重新启动机器, 用 CPAV 和公安部 KILL V62.00 查毒, 没有任何结果。

(下转 47 页)

# 1993年总目录

## ·综述·

	页	期
金鸡报春早 催人再加鞭 .....	本编辑部	2 1
美国超级电脑市场 .....	吕问黎	3 1
浅谈计算机网络 .....	倪永仁	2 2
从三种通用的八位单片机硬、软件比较		
来看 Z8系列单片机的优越性 .....	董伯明	2 3
又闻又见——电视电话 .....	晓亮	2 4
商业管理软件:〈流水账〉的写作经验		
.....	蔡培强	2 5
对公用数据网的几点建议 .....	薛兴华	3 5
人工智能型电脑的发展概况		
.....	万欣 高增锁 万红	2 6
祝贺“电子与电脑”第100期 .....	吴鸿适	2 7
计算机与信息研究综述 .....	贾洪卫	2 7
电子图书的发展及中国市场的开拓		
.....	严华	2 8
如何选购个人电脑 .....	廖佳	3 81
关于中小学计算机装配的探讨 .....	徐忆梅等	2 9
家用微机及其市场开拓 .....	周宏	2 10
PC机加解密软件及技术 .....	陈伟	2 11
令人翘首企盼的 POWER PC .....	吴腾奇	2 12
计算机硬加密原理及技术 .....	李晓中 张景生	4 12

## ·PC 用户·

为 dBASE III “创建”汇编语言接口 .....	涂振宇	5 1
内存自由空间检测程序 .....	潘伟康	6 1
IBM-PC BASIC 语言调用鼠标的程序		
设计方法 .....	陈惠生	7 1
PC 机软件加解密技术剖析(续) .....	李文亮	9 1
普及型计算机汉字使用问答 .....	张济生	4 2
普及型 PC 个人微机如何应用 2.13E 汉字操作系统		
.....	陈卫平	5 2
扩展四通打字机功能的方法 .....	王冈	6 2
引导型病毒与磁盘逻辑坏簇 .....	崔来堂	10 2
DIR-I 病毒的自我快速解法 .....	刘亚兵	11 2
磁盘管理软件 DM .....	李仲先 巢慎冰	12 2
如何用 C 语言编制数据采集软件		
.....	冯品如 张新民	4 3
谈谈 MS-DOS 5.0 新增命令 DOSKEY		
.....	唐银红	7 3
用 SET 命令解决 COMMAND.COM		
的重新调入 .....	叶斌	11 3
任意盘任意子目录下调用 WS 的方法		
.....	何捷	11 3
C-dBASE III 数据文件结构及其与其它		
语言的共享 .....	卢小平	12 3
“中国青蛙”病毒 .....	罗亚	13 3
以毒攻毒——消除 DIR-2 病毒的简便方法		
.....	刘山水	14 3

## IBM PC 微型机 FORTRAN 语言绘图

子程序设计 .....	张辉	4 4
扩展 DOS 内部命令 DIR .....	戴福华	7 4
C 语言汉字显示技术 .....	颜飞	12 4
灵活实用的打印程序 .....	郝显峰	14 4
巧用数组和 GATHER FROM 命令		
.....	何钧军	16 4
西文图形状态下显示汉字的 Turbo C 程序		
.....	张辉 张敬民 范俭	4 5
PC Tools 若干高级实用技巧 .....	李日林	6 5
DOS 下图象文件的打印输出 .....	杨晓慧	7 5
如何使 PC 机具有实时时钟 .....	李维宪	9 5
再谈运行 FoxBASE+ 内存不足的问题		
.....	冯建军	10 5
利用电脑预测儿童青少年成人时的身高		
.....	黎彦才 米绍平 徐奎	11 5
五笔字型系统用压缩字库 .....	徐庆恩	13 5
TRAVELLERC 病毒的发现与清除		
.....	桂宁	15 5
CPAV——使您的电脑百毒不侵		
.....	郭文典	16 5
用“锁匙软盘”保护硬盘 .....	崔来堂	17 5
数据库系统下求三角、反三角函数的命		
令文件方法 .....	凌莉 葛晓光	3 6
在不同的目录结构中使用 RESTORE 命令		
.....	蔡廷武	5 6
Epson LQ1600 对屏幕彩色图形的硬拷贝		
.....	周晓	6 6
C 编程中易犯的错误 .....	苑丽霞	8 6
C-WordStar 使用基础问与答 .....	朱大公	9 6
在 TANGO 软件中建立汉字元件库 .....	肖安顺	11 6
病毒一例 .....	谭洪勇	12 6
“3.6”一种超级恶性病毒 .....	李建俊	12 6
弹出式菜单中屏幕滚动的具体实现		
.....	董立平 李钢得	4 7
普及型微机汉字系统及 2.13H 打印驱		
动程序的改进 .....	曲峰	6 7
DOS 内部命令的修改和扩充 .....	姜金友	7 7
Super PC/XT 附加存储器的使用 .....	袁卫国	8 7
快速筛法求证素数的程序设计 .....	王存	11 7
文件目录的快速查找 .....	刘炳文	13 7
巧用长城 0520EM 运行汉字 dBASE III		
.....	辛启亮	14 7
C-WordStar 使用基础问与答 .....	朱大公	15 7
CCDOS 4.0 解密的方法 .....	范恒钦	16 7
1741 病毒的检测及其清除 .....	邹肇辉	17 7
PCX 格式图像文件的应用技巧 .....	周光辉	5 8
一程序多报表的自动建立、修改与输出		
——dBASE 与 CCED 的结合使用 .....	刘士杰	8 8



西文文本方式下怎样利用 VGA 及其兼	
容显示器显示汉字 .....	樊启柏 9 8
直接读/写 VGA 的图形像素 .....	王绍鑫 12 8
2.13F 系统在单显环境下使用所遇几	
个问题的解决办法 .....	张彦超 13 8
巧用 INT 16 .....	李珊红 14 8
安全可靠的扩充功能键高级编程技术	
.....	侯培玉 15 8
计算机病毒的检测、消除和预防 .....	苏民生 16 8
文件型六四病毒的简析与防治 .....	李超勇 17 8
磁盘树形目录结构及其分析方法 .....	崔来堂 4 9
模拟雪花生成的单元计算法 .....	苏翼凯 6 9
实现功能键重新定义的一种全新方法	
.....	黄庆程 7 9
增加软盘容量的简便方法 .....	李齐 8 9
给 MS 及 WPS 的五笔字型加上自定义词组	
.....	汪建军 9 9
字幕机怎样利用电脑的图象与文字处理功能	
.....	邢秀起 11 9
如何用 C 语言显示空心汉字 .....	李辉 12 9
“将错就错”在 FoxBASE+ 中的应用 .....	陈实 13 9
防病毒软件检测能力的提高 .....	陈乃强 14 9
C-WordStar 使用基础问与答 .....	朱大公 15 9
计算机病毒的检测、消除和预防 .....	苏民生 17 9
磁盘软格式化方法及其实现 .....	高洗 4 10
巧用 DOS 中的 I/O 重定向命令 .....	缪晓明 6 10
如何将 WPS.dBASE III 的帮助信息打印出来	
.....	李政 8 10
WPS 文件阅读器 .....	瞿新国 9 10
不同规格双软配置大容量软盘间文件复制	
.....	彭禾 10 10
用 DEBUG 的 L,W 命令读写硬盘隐含扇区	
.....	王化丰 11 10
C-WordStar 使用基础问与答 .....	朱大公 11 10
计算机病毒的检测、消除和预防 .....	苏民生 13 10
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 10
解决 TSR 程序一次性驻留内存的根本方法	
.....	宋立波 7 11
FoxBASE 下一个新型的通用菜单控制程序	
.....	周丝溪 胡武海 11 11
在批处理文件中巧用 DOS 环境变量 .....	黄庆程 14 11
DOS 文件的 FCB 管理机制分析 .....	崔来堂 16 11
CMOS 中参数的保存及重置方法 .....	金林樾 19 11
获得西山 DOS 五笔字型词组 .....	丁梅 20 11
最小汉字笔画库的设计和使用 .....	黄焕如 21 11
背景音乐的原理及其实现 .....	丁塔 23 11
C-WorStar 使用基础问与答 .....	朱大公 24 11
计算机病毒的检测、消除和预防 .....	苏民生 28 11
INT28H 与常驻内存的程序设计 .....	罗颖 6 12
Super-Star 图文排版系统与高级语言	
.....	张辉 张敬民 袁艺 8 12

新世纪/XqR 病毒的检测与清除 .....	车光宏 10 12
DOS 5.0 的 CONFIG.SYS 的设置 .....	郭文典 13 12
C-WORDS TAR 使用基础问与答 .....	朱大公 14 12
计算机病毒的检测、消除和预防 .....	苏民生 16 12

### ·学习机之友·

实用程序三则 .....	庞建新 13 1
现场评分实用程序 .....	曾庆平 15 1
ProDOS 系统内部结构剖析 .....	廖凯 19 1
数据指针的定位 .....	毛贵明 14 1
给 APPLESOFT 系统增加 CIRCLE 功能	
.....	陈建明 15 2
中华学习机拉幕式显示程序 .....	唐治亚 16 2
任意位小数、整数的连续加减精确运算程序	
.....	张新莲 16 2
ProDOS 系统内部结构剖析(续) .....	廖凯 19 2
XMF-BASIC 全功能扩充 .....	石永琳 17 3
实用程序二则 .....	李成功 19 3
任意数的高精度乘法 .....	张新莲 20 3
ProDOS 系统内部结构剖析(续) .....	廖凯 22 4
CEC-I 中华学习机系统子程序 .....	王志超 17 4
ProDOS 系统内部结构剖析(续) .....	廖凯 22 4
启动 DOS 的技巧 .....	邵爱柏 18 5
中华学习机 DOS 使用技巧 .....	张本灵 19 5
学习机游戏软件“攻关秘诀” .....	王海涛 20 5
任意数的高精度除法运算程序 .....	张新莲 20 5
ProDOS 系统内部结构剖析(续) .....	廖凯 22 5
任意数任意次整数幂的高精度运算程序	
.....	张新莲 13 6
高精度数值算法的改进 .....	邓阳春 15 6
ProDOS 系统内部结构剖析(续) .....	廖凯 17 6
谈 6502 软中断功能的开发应用 .....	苏华 19 7
汉字文稿打印程序 .....	冯惠民 21 7
ProDOS 系统内部结构剖析(续) .....	廖凯 22 7
改进的 CEC-I 实用小程序 .....	薛嘉兴 18 8
ProDOS 系统内部结构剖析(续) .....	廖凯 21 8
6502 反汇编程序 .....	刘民 39 9
按姓氏笔划排序 .....	翁晓毅 46 9
ProDOS 系统内部结构剖析(续) .....	廖凯 18 9
ProDOS 系统内部结构剖析(续) .....	廖凯 19 10
CEC-I 中华学习机汉字系统简介 .....	林永春 30 11
多功能标准化学成绩统计程序 .....	汤永进 31 11
ProDOS 系统内部结构剖析(续) .....	廖凯 33 11
把 & 和 CALL 语句“嫁接”到赋值语句中去	
.....	邹习新 18 12
ProDOS 系统内部结构剖析(续) .....	廖凯 20 12

### ·FORTH 语言讲座·

第二讲 冒号定义 .....	丁志伟 21 1
第三讲 堆栈 .....	丁志伟 21 2
第四讲 FORTH 中的数 .....	丁志伟 23 3
第五讲 编辑及有关操作 .....	丁志伟 24 4
第六讲 程序控制结构上 .....	丁志伟 24 5

第七讲 程序控制结构下	丁志伟	18	6
第八讲 内部结构	丁志伟	24	7
第九讲 输入与输出(上)	丁志伟	21	8
第十讲 输入与输出(下)	丁志伟	22	10
第十一讲 编辑实例	丁志伟	23	10
第十二讲 D1.0版本补遗	丁志伟	22	12
第十三讲 FORTH-83标准摘要	丁志伟	25	12

•初、中级程序员软件水平考试辅导•

计算机硬件基础知识	顾育麒	24	1
计算机硬件基础知识(续)	顾育麒	25	2
基本算法一	李宁	26	3
C语言试题与分析	赵国瑞	28	4
C语言试题与分析(续)	赵国瑞	28	5
数据结构	夏晓东	22	6
C语言中的字符串处理及指针变量在字符串处理中的应用	李宁	29	7
第四届全国计算机软件人员竞赛将在京举行初级程序员级		25	8
一九九三年计算机初级程序员有奖竞赛试题答卷		25	8
一九九三年计算机初级程序员有奖竞赛试题		26	8
一九九三年计算机初级软件人员竞赛试题解答与分析(参考答案)		25	9
链式存储及树形结构	夏晓东	28	10

•新书与软件•

电子工业出版社部分优秀图书		29	1
AutoMovie 2.0版高级工业设计造型动画系统		30	1
东方会计DFKJ通用财务软件简介		30	1
CYSCB-2 MCS-51.8098监控软件设计	吴微 罗维国	32	1
全国计算机软件技术连锁服务中心经营部		24	2
电子工业出版社软件部《新版软件介绍》		30	2
电子工业出版社软件部征稿启事		29	2
Turbo PASCAL V6.0介绍	欧阳慎	29	3
邮购消息		27	4
PCT V2.0写作系统	张连永 杨峰	23	8

•学用单片机•

BJS-98单片机多功能实验系统	袁涛 魏峰	35	1
单片机多机通信中1488芯片的应用	江琪	38	1
开发单片机的结构化高级语言 PL/M-96	袁涛 魏峰	31	2
MCS-51单片机单板机监控程序分析(上)	江琪 刘葳	34	2
BJS-98电路原理与CDW多窗口调试器			

MCS-51单片单板机监控程序分析(中)	袁涛 魏峰	31	3
BJS-98硬件、软件典型实验	袁涛 魏峰	31	4
MCS-51单片单板机监控程序分析(下)	江琪 刘葳	34	4
BJS-51教学实验板的普及型新品“DP-851单片机普及板”技术讲座	王玉民	31	5
单片机普及函授班招生简章		32	5
SCB系列单片单板机介绍		32	5
从三种通用的八位单片机硬、软件比较来看Z8系列单片机的优越性(续)	董伯明	33	5
单片机的学习与应用	罗明宽 车金相	32	6
80C31单片机防掉电和抗干扰电源的设计	马祝阳 姜凤怡	34	6
从三种通用的八位单片机硬、软件比较来看Z8系列单片机的优越性	董伯明	36	6
DP-851监控程序简介	罗明宽 车金相	39	7
实用汉字库芯片的制作	陈超波	43	7
51单片机学习及应用装置	裴巍	45	7
监控程序阅读器	郝健 罗明宽	30	8
EPROM编程卡和逻辑笔	罗明宽	30	9
对《单片机旅馆客房门卫系统》中P2口使用的探讨	吴哲	32	9
单片单板机组成与应用		33	9
T6668语音处理器手动实验	罗明宽 车金相	31	10
MCS-51系列单片机中的加密措施	张戟	34	10
SCB-I型单片单板机详介		35	10
单片机软件抗干扰新设计	李凯里	35	11
一种简单实用的集散控制系统	王济浩	37	11
固体录音机的制作	罗明宽 车金相	28	12

•电脑巧开发•

为APPLE-I增加智能示波器功能	杨悦民	39	1
8031程序写入卡的分析与使用	车金相 罗明宽	37	2
简易计算机辅助教学系统	杨悦民	38	2
PC电力拖动控制	李文兵	36	3
IBM-PC与彩色电视机接口	蒋海明 王辉	37	4
与CEC-I微机接口的EPROM仿真卡	王正英	34	5
PC机并行口的扩展	孙胜春 邢昌风 肖英	37	5
单片机主从式通信系统中的广播发送	江琪	39	6
给单片机加装V/F/F/V转换器	岳海军	40	6
一种防止单片机程序运行失常的软件对策	黄稳山	41	6
病毒克星——华思中文防病毒卡		28	6

微机打印机共享器原理与设计	朱明程等	34	7
IBM-PC 总线扩展技术和开放工作台	朱世鸿	47	7
微型计算机遥控键盘	顾正平	50	7
运用“看门狗”技术提高在线式智能仪器的可靠性	汤忠庆	33	8
如何将苹果机 AD/DA 卡改制为 PC 机 AD/DA 卡	王晰	34	8
UN-1A 型电压模拟数字采集器	刘北顺等	36	8
给 PC 机配装射频调制器	郭荣军	39	8
用黑白电视机改装计算机单显	邓洪儒	36	9
单片机开发器与 PC 机的通信接口	张卫东	37	9
CEC-I EPROM 编程电路	齐向东	38	10
一种具有自动唤醒功能的掉电接口电路	陈敦洪	40	10
LED 智能显示屏的结构及驱动、显示电路	张艺	38	11
实验教学计算机的研究与开发	王勤民	31	12
半导体存储器在自动打铃器中的应用一例	刘长茂 邱杰	33	12
新型实时日历时钟 MSM6242	徐民 富歌	35	12

• 电脑游戏机 •

第四讲 游戏程序的设计过程(下)	于春	42	1
小天才游戏机剖析	王哲	44	1
第五讲 游戏程序的编程特技(上)	于春	43	2
用 F BASIC 编写的“黑白棋”游戏程序	于涛	46	2
第五讲 游戏程序的编程特技(中)	于春	44	3
第五讲 游戏程序的编程特技(下)	于春	43	4
实用程序——彩色电视信号发生器	王长宾	46	4
第六讲 游戏机程序实例分析(上)	于春	41	5
家用电子游戏机故障维修一例	杨林	43	5
第六讲 游戏程序实例分析(下)	于春	43	5
F BASIC 语言编程特技的深入研究	于春	53	7
第二章 6527CPU 的显示系统(上)	于春	42	8
第二章 6527CPU 的显示系统(中)	于春	43	9
第二章 6527CPU 的显示系统(下)	于春	42	11
第三章 6527CPU 音响系统	于春	40	12

• 维修经验谈 •

PC XT/286 检修一例	井然	46	1
中华学习机特殊故障修理一例	高赣湘	46	1
CR3240 打印机与 AR3240 打印机并非完全兼容	董赜 郭瑞芳	47	1
BF PC-BOY 家用电脑的维护	北方电脑公司	47	1
巧修 AST/286 主机系统板	范广宇	40	2
微机运行状态显示灯板维修	张爱国	40	2
微机性能测试与诊断	黄文	41	2
苹果机电源负载能力差改进一例	施元忠	42	2

彩色显示器工作原理简述及故障诊断上	胡野红	42	3
彩色显示器工作原理简述及故障诊断中	胡野红	40	4
电源引起的特殊故障一例	王颀	41	4
《对单片机旅馆客房门卫系统中 P2 口使用的异议》	秦化渤	42	4
彩色显示器工作原理简述和故障诊断下	胡野红	38	5
NKP-824G 打印机一故障维修方法	蒋国权 张宏杰	42	6
M2024 打印机双向打印时错位的解决办法	陆钦俭	43	6
Super-AT286 微机硬盘软故障排除一例	费毅	43	6
谈谈 LQ1600K 打印头的维护及短(断)针利用	刘斌	40	8
后备式中小型 UPS 不间断电源的改进	谭巨高	41	8
高速非击打式打印机市场动态及技术综述	魏梓栋	37	9
彩色显示器 ASTECD1 型开关电源的检修方法	秦伟奇	40	9
四通 MS-2401 打字机维修五例	蔡长安	41	9
巧用 PC Tools 清洗磁头	唐银红	42	9
简易的苹果机故障检修方法	王惠民	41	10
AR-3240 打印机“联机”灯闪烁故障检修一例	张全悦 张书琪	42	10
Super AT286 运行金山 CCDOS 时频频不同步的解决办法	刘良团	43	10
全国电子报刊联合征订简明目录表		44	10
出版软件简介		45	10
激光印字机的维护与故障排除	崔晨荣	41	11
AST286 系统板总线故障的分析与排除	梁杰黑	32	12
中环 CT100 终端常见故障维修	徐俊	38	12
菊花遥控彩电“锁不住”故障分析	李桂洁	39	12

• 电脑与通信 •

小型办公通信系统	张宗耀	49	1
电脑的电话机—Modem	经远	52	1
电脑化更上一层楼		47	2
为关心价格的企业提供一种易用的网络			
Net Ware Lite V1.0	新地	48	2
让微电脑之间相互对话		49	2
计算机多媒体技术的新信息	文波	51	3
基于图形的联想多媒体工具 GBH	白成林	43	12

• 传真机专题讲座 •

第一讲 传真概述	张景生 张建军	53	3
PC 机之间最简单的通信		55	3
二传真机原理(上)——传真图象数据的			

编码和解码	张建军 后俊堂 张景生	47	4
PC 间能传送文件的程序		51	4
应用程序接口 API 时代的开端	文海	50	5
传真机原理下——调制和解调	张建军 孙晓春 张景生	51	5
TCP/IP 网络简介	杨军	52	6
Transputer 与并行处理	师军	53	6
如何选用 MODEM 卡——PC-PC 远程通信(一)		54	6
传真机传输控制规程	张建军 张景生	55	6
用电流环路实现远程快速数据传输	袁学文	55	7
传真机的安装和使用	张建军 张景生	57	7
MODEM 卡的使用——PC-PC 远程通信(二)		59	7
传真机的日常维护	张建军 张景生 李晓中	47	8
通信软件的使用		49	8
NOVELL 网络的保密技术	王杰民	50	8
CEC 与 PC 机通信两例	姚克义	51	9
多媒体技术——计算机发展的一场革命	李恒嘉	52	9
试谈 HELLO918 系统	唐舜	53	9
传真机简易故障的应急修理	张建军、张景生	54	9
PC 机与 8031 串行通信的波特率选择	江琪	47	10
传真通信的通路	张景生 张建军 辛亚西	48	10
我国专用计算机通信网的组网方式	薛兴华	52	10
ORACLE 关系数据库	杨 军	53	10
微机通信网络的发展状况及趋向	薛兴华	50	11
如何升级你的网络工作站	薛强	54	11
微机——传真机系统	刘保元 张景生	44	12
大连华联商厦的计算机网络	廖涛	46	12

### ·IC 电路应用·

指数运算集成电路 AD538 及应用	王俊省	54	1
高频开关稳压电源及其 IC	李秀华 张占松 梁全稳	50	2
美国 IC 公司的专业产品简介		52	21
集成电路滤波器及使用方法(一)	李兰友	53	2
高频开关稳压电源 IC 讲座			
第二讲 开关电源基本组态和企业标准	李秀华 张占松 梁全稳	47	3
集成电路滤波器及使用方法(一)(续)	李兰友	50	3
开关电源用集成控制器	李秀华	53	41
RS232/TTL 电平转换芯片 MAX232 的原理及应用	罗维国 吴微	46	5
单片功率开关电源 IC	李秀华 吴剑棠	48	5
集成滤波电路及用法(二)	李兰友	47	6
高频变压器的设计与选择	戴惟荫 张占松	50	6

模拟乘法器 IC 及使用方法(一)	李兰友	61	7
模拟乘法器 IC 及使用方法(一)(续)	李兰友	53	8
节能的 IC 小电路	向心	54	8
UC3852 控制 IC 及其应用	毛兴武、祝大卫、马存艾	47	9
模拟乘法器 IC 及使用方法(二)	李兰友	49	9
模拟乘法器 IC 及使用方法(二)	李兰友	54	10
PLD 器件的新军——GAL	倪征宇	55	10
计算机中的新技术——Flash Memory	张戟	45	11
模拟乘法器 IC 及使用方法(三)	李兰友	48	11
经验二则	王万春	49	11
COP840C 芯片在洗碗机自动控制中应用	王求是 季行健	48	12
乘法器 IC 及应用	李兰友	50	12

### ·读者联谊·

电脑进入家庭	沙有威	56	1
长安系列微机配置		56	1
我国中小学计算机教育软件的现状与发展	王相东	56	2
PC 机最新教学、游戏软件介绍:华容道		49	3
回音壁		52	4
咨询		55	4
双频卡使用问答	程伟	55	5
两处疏漏 一点建议	何崇乐	56	5
EXPANDED MEMORY 与 EXTENDED MEMORY 译名管见	彭禾	52	6
DOS 应用技巧集锦	李际	63	7
加速 WPS 的实现	邓海、白奉礼	56	7
对《最近出现的几种病毒》的几点异议	何崇乐	52	8
PC/XT 及其兼容机内存的扩充与使用	李晓中	55	8
让文本阅读器显示汉字	许勇	52	8
无绳电话干扰电视节目	钱雁群	56	8
回音壁			
单片软光盘 21MB 系统问世			
电子计算机新一代存储设备		56	8
使用 WPS 字处理软件的小窍门	银明超	56	9
在计算机上输入汉字要规范化	桂英	56	10
字符串译码编码器		56	10
修改 BIOS 程序实现对 PC 机的加密	徐文军 李晓中	54	11
修改软金山系统,适应有缺陷的 386 主板	文森	56	11
CCED 使用次数的修改	朱融	56	11
消毒软件 CPAV 的局限性	张志	51	12
经验数则	曹 彤	51	12

