

E&C
1994
 ●一九九四年 ●总期第 112 期
7

電子

ISSN 1000-1077



電腦



● **ELECTRONICS AND COMPUTERS** ●

银奖产品·配套出口

SBW系列全自动补偿式大功率电力稳压器 DBW系列全自动补偿式大功率电力稳压器 SBWDT 电梯专用补偿式电力稳压器


高效率、低损耗、波形畸变小、应变时间短、运行安全可靠、范围宽精度高、适用各用电单位

规格: SBW(三相): 3~2000kVA • DBW(单相): 3~300kVA • SBWDT: A型50kVA, B型30kVA



JingDa

- 稳压器行业标准在我厂制定
- 邮电电信总局优选推荐产品
- 获国家电力电子产品合格证
- 获91年北京国际博览会银奖
- 参加香港中国得奖产品展览

 要稳胜 找精达!

上海精达电力稳压器制造公司

地址: 上海市新闻路1331号 邮编: 200040 电话: 2794464 2794540 2477585 电挂: 6548 传真: (021)2794088 厂址: 朱行路17号

全国联销·全国联保

经销单位

广西南宁市健宇电气公司	电话: 217256	联系人: 卓仕平	无锡市电梯厂维修工程部	电话: 607559	联系人: 何荣欣
大连汇英电梯安装维修公司	电话: 2633400	联系人: 王振有	上海三菱电梯新疆服务部	电话: 221551	联系人: 付沁芳
威海市中外电梯工程公	电话: 225575	联系人: 毕兆文	维吾尔自治区建筑机械公司		
无锡机电设备公司电仪仪表分公司	电话: 200353	联系人: 侯中峰	电梯公司	电话: 413617	联系人: 骆大明
武汉铁路电梯销售安装维修公司	电话: 741183	联系人: 李奂群	南昌市电梯安装公司	电话: 214645	联系人: 胡康全

● 国内代号: 2-888 定价: 2.40 元

过去的让它过去

兴归电子

现实的祝君实现



图一、A:BM-555 带遥控大功率多功能放大器,该器采用数字电路技术,全轻触式功能按键,带显示 12 级连续音量控制及双声道自动平衡电路,具环绕声回响卡拉 OK 功能,大 LED 显示屏,三路麦克风输入,五功能带显示输入选择,可遥控或手动操作,每台 580 元,邮资 20 元;B:高级豪华型超大液晶频谱自动均衡器,该器采用零失真的专用集成电路及全轻触式电脑数码控制线路,液晶频谱双十四段具五种显示功能,每台 630 元,邮资 20 元。C:CH-938 带遥控激光唱机/立体声卡座,具激光唱机与录音卡座一体化结构,同步自动信号转换,红外线遥控可随意点唱任一首曲目,具录音电平调节及磁

带选择功能,适合不同的磁带使用,每台 980 元,邮资 30 元。

图二、最新推出高级无线传声接收系统,本系统由高级无线麦克风及高灵敏度接收机组成。可直接配接任何卡拉 OK 混音器及扩音机的有线输入插孔。如带 OK 插孔的录像机、镭射机及音响等。以解除有线话筒的各种不便,使卡拉 OK 更灵活,方便和乐趣。该系统采用先进的电子电路及高集成度 IC,具有超静噪,频响宽,音质好等特点,广泛适用于卡拉 OK 厅、舞厅、会议厅及家庭等用。使用电压,接收机 220V、话筒 6V (5# 电池四只),使用距离:≥40 米,单路每套 285 元,邮资 10 元;二路每套 580 元,邮资 20 元。

(单位:元)

型 号 及 名 称	性 能 简 介	单价	邮资
东声 2213HR 镭射唱机 东大尼索 HCD-988 镭射唱机	索尼三光束激光唱头组装,红外遥控,全功能液晶显示,直接 16 首电脑选曲,具随意放送、重复放送、搜索等功能,机身尺寸:430×250×90mm	870 830	30 30
超大液晶屏双十段均衡器	纯集成组装 TL084 均衡线路,日产 9000 型荧光显示巨胆,数字电路转换功能,频谱双十四段四种显示方式,具多种信号输入,尺寸:430×240×70mm	460	20
高级双十段液晶均衡器	采用进口原件组装,5532 作运放。NEC 荧光显示,全轻触转换。尺寸:430×240×60mm	290	20
K 皇牌 K-33 多功能大功率放大器	具剧院式环回立体声效果,四路喇叭输出,双变压器大功率管组装,多种信号输入转换,大型 LED 电平显示,带 BBD 残响,二话筒输入,功率前音箱 2×250W 后音箱 2×50W,尺寸 430×300×110mm	425	20
K 皇牌 K-55 型带遥控全功能扩音机	6 路输入可驳接各种影视讯源,按键式 CPU 电脑智能控制系统,19 键全功能红外线遥控,二路话筒输入,卡拉 OK 混响时间可调,功率 2×400W,尺寸:430×300×120mm	830	40
天朗 928 双卡录放卡座	具双卡连续翻转放音,电脑自动选曲,歌声隐没卡拉 OK 伴奏系统,杜比 B 录放降噪等功能。尺寸:430×220×130mm	560	20
2×100W 倒立自动放扩座	采用进口脑机芯,具中途转向及倒带连续放音功能,带环绕声 BBD 混响,二路话筒输入卡拉 OK,大型 LED 屏电平显示系统,尺寸:430×240×130mm	340	20
2×100W 倒立放扩座	进口机芯组装,外观功能同上,少自动倒带功能,尺寸:430×240×130mm	280	20
朗玛-925 双卡收录音机	具短波、中波、FM 波三段收音,卡 A 卡 B 连续放音,高速同步复制录音等功能交直流两用带卡拉 OK 系统,尺寸:480×141×105mm	230	20
2×120W 功放板	采用 STK4191 集成组装,前置用 5532 作运放,设有集成保护电路,四电位置调节,已组装调试。	140	5
2×250W 大功率功放板	采用大功率达林顿对管组装,前置用 MC1458 三片作运放,专用散热器,六电位置调节,已组装调试。	110	5
2×80W 大功率功放板	采用大功率管组装,前置用 MC1458 二片作运放,五电位置调节,已组装调试。	55	5
卡拉 OK 混响前置板	中心混响采用 3102、3207 四路话筒输入、七电位置调节,已组装调试。	55	5
高敏度超微型无线话筒	使用频率 FM 波 88-108MHz,用 AG13 电池供电,有效发射距离 50 米,尺寸:4×1.2×0.8cm	27	3

说明:以上产品均配有详细资料和相片,索取时请付资料费 2 元,并注明需要名称。索取价目表时请付足回邮邮资。邮购方式:邮局汇款请寄:广东省潮阳市棉城镇兴归电子器材厂邮购部,地址:市中山中路北 11 栋之四,银行汇款请汇:工商银行潮阳市城西办,帐号:24502,所需名称、数量写于汇款附言内。电话:(07644)810758,电挂:0141,邮政编码:515100,联系人:杨江。



单片可编程系统器件

Part Number	Bus Width	DPLD		I/O Port Pins	EPROM Kbit	SRAM Kbit
		Inputs	Product Terms			
PSD301C1	x8 / x16	12	40	19	256	
PSD311C1	x8	12	40	19	256	
PSD302C1	x8 / x16	16	40	19	512	
PSD312C1	x8	16	40	19	512	
PSD303C1	x8 / x16	16	40	19	1024	
PSD313C1	x8	16	40	19	1024	
PSD301	x8 / x16	12	40	19	256	16
PSD311	x8	12	40	19	256	16
PSD302	x8 / x16	16	40	19	512	16
PSD312	x8	16	40	19	512	16
PSD303	x8 / x16	16	40	19	1024	16
PSD313	x8	16	40	19	1024	16

Part Number	Bus Width	DPLD + GPLD			I/O Port Pins	PMU	EPROM Kbit	SRAM Kbit
		Inputs	Product Terms	Registered Macrocells				
PSD401A1	x8 / x16	37	113	8	40	Yes	256	16
PSD411A1	x8	37	113	8	40	Yes	256	16
PSD402A1	x8 / x16	37	113	8	40	Yes	512	16
PSD412A1	x8	37	113	8	40	Yes	512	16
PSD403A1	x8 / x16	37	113	8	40	Yes	1024	16
PSD413A1	x8	37	113	8	40	Yes	1024	16
PSD401A2	x8 / x16	59	126	24	40	Yes	256	16
PSD411A2	x8	59	126	24	40	Yes	256	16
PSD402A2	x8 / x16	59	126	24	40	Yes	512	16
PSD412A2	x8	59	126	24	40	Yes	512	16
PSD403A2	x8 / x16	59	126	24	40	Yes	1024	16
PSD413A2	x8	59	126	24	40	Yes	1024	16

Part Number	Bus Width	DPLD + GPLD + PPLD			I/O Port Pins	PMU	EPROM Kbit	SRAM Kbit	Counter/ Timer	Interrupt Controller
		Inputs	Product Terms	Registered Macrocells						
PSD501B1	x8 / x16	61	140	30	40	Yes	256	16	4-16 bit	8-Level
PSD511B1	x8	61	140	30	40	Yes	256	16	4-16 bit	8-Level
PSD502B1	x8 / x16	61	140	30	40	Yes	512	16	4-16 bit	8-Level
PSD512B1	x8	61	140	30	40	Yes	512	16	4-16 bit	8-Level
PSD503B1	x8 / x16	61	140	30	40	Yes	1024	16	4-16 bit	8-Level
PSD513B1	x8	61	140	30	40	Yes	1024	16	4-16 bit	8-Level

电子与电脑

一九九四年 总期第 112 期

目 录

· 综述 ·

汉字形的自然内码与汉字教学

..... 蒋辅文 张连永(2)

· PC 用户 ·

指定范围 VGA 图像彩色拷贝 方 震(4)

一个实用的彩色图符编辑器程序..... 吕学山(6)

充分利用 DOS 的扩展内存来运行大型程序

..... 张晓莉(9)

在 MS-DOS 高版本下安装汉字系统

..... 李晓中 宋 涛(10)

CCED 数据计算中的出错处理

..... 刘士杰 陈依平(12)

也谈对几种新病毒的正确认识与清除

..... 王江民(13)

· 学习机之友 ·

用 F BASIC 编制的 6502 汇编与反汇编程序

..... 郭 杰(15)

· DOS 操作系统分析讲座 ·

第四讲 汉字信息处理 崔来堂(18)

· 学用单片机 ·

MCS-51 单片机控制的 DTMF 信号发送/接收电路

..... 廖天河 李亚平(25)

片内存储器 RAM 胥筱汀(27)

· 电脑巧开发 ·

计算机网络管理系统在旅行社的设计及实现

..... 崔晨荣 方勤(31)

巧用 PC 机并行口 徐令元(33)

多用手写字符输入板 林根远(34)

· 维修经验谈 ·

打印机讲座

第三讲 针式打印机的基本工作原理 赵继文(36)

· 电脑游戏机 ·

第六章 背景画面绘制的编程技巧(续)

..... 于 春(40)

· 初学者园地 ·

工具软件 PC Tools 的应用 王路敬(44)

CCED 编辑软件中的灵活运用 任富坦(42)

· 多媒体世界 ·

工程多媒体专家系统的技术展望 刘 箴(47)

基于声霸卡开发音响制品管理系统的研究

Microsoft Windows 3.1 应用基础——Windows 进阶

..... 刘维亮 葛仁伟(49)

· Windows 技术讲座 ·

..... 徐鹏力 赵海航(43)

· 电脑通信 ·

PC 机与 MCS-51 单片机的主从式数据通信

..... 康赐荣(51)

· IC 电路应用 ·

时钟芯片 MSM5832RS 梁 治(53)

· 读者联谊 ·

谈 WPS 文件阅读器 陈志鹏(56)

DM(V3.01)使用之我见 袁志新(42)

查找备份文件恢复路径点滴体会 彭 禾(55)

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:施玉新

订购处:全国各地邮电局

国外总发行:中国国际图书贸易总公司

(北京 399 信箱 邮政编码 100044)

广告经营许可证:京海工商广字 0030 号

定价:2.40 元

汉字形的自然内码与汉字教学

蒋辅文 首都师范大学计算中心

张连永 北京教育软件研究开发中心

一、汉字形码的主流路线

科学家预言,二十一世纪是汉字的世纪。人们相信这个预言,是因为他高瞻了世界科技文化发展的流向,洞见了作为中国第五大发明的汉字本身的科学性及其可能被世界接受的媚力。

汉字原本是一套庞大的系统工程,只因被它自身的字数多,字形复杂之烟云笼罩着,不易被人们认识其真面目。虽经十多年编码运动的孜孜求索,还没公认出一种科学的排序方法,如外国人所说:“汉语好学,汉字难写,尤其难查。”其原因在于无论从形的方面还是从音的方面均未解析出它固有的,体现本质特征的自然内码。本文以计算机模拟实验结果为依据,从形的方面阐释汉字的自然内码,抛砖引玉,以期专家学者们斧正。

国内外的汉字形码,主要有两条路线:

1. 以形定位的路线。即把笔划、笔划组合和字元定义到键盘的座标位上,用数字和非读音的字母作代码,如五笔字型、表形码、二维三码、仓颉码以及四角号码、无理序号输入法等,它们都是建立在笔划基础之上的,往往采取拓扑法拆取字元,字被拆得很零散,无法对汉字排出易于辨读的,有唯一地址的顺序来,尽管某些方法获得了较高的输入速度,但因其规则繁多,记忆量大,难以被广大用户迅速掌握。

2. 以音统元的路线。即以键盘字母的音代表字元的音,如见字一识码,五十字元等,而以本文叙述的元音码为终结正果。

元音码以汉字编码专家支秉彝教授的“汉字结构音元化”理论来解释汉字的结构成分,使每个汉字呈现出自己的自然内码并直接转换成字母串:

□□□
□□□=○○○○○○○○○ I
□□□

终于使数以万计的汉字可以用 26 个字母解释其结构成分并进而获得符合国际标准的顺序,方块形的汉字有了与拼音文字等效的文字层次结构模式:

英文词(音)=词素(音)=字母(音) II

中文词(形)=汉字(形)=音元(形/音) III

从方程式 III 可以看出,由于汉字音元有着形和音的双重性格,中文字词之编码、输入和检索已获得统一处理的工具,并为汉字之教学提供一种有效的新方法。

汉字结构音元化理论和元音码为汉字形码开劈了一条主流路线。

二、揭示汉字形的自然内码

1984 年,支秉彝教授在《中文信息的计算机处理》一书序言中指出:“汉字结构是以笔划为基础的字元有规则的组合,很类似拼音文字的字母。如果人们对汉字结构较有认识,就不难理解汉字是有规律的自然信息。”这段话的历史意义在于首次从理论上指明了汉字形的自然内码之客观存在。所谓自然内码,即一种文字有着固定顺序的基本单元如英文的字母,它们组成的单字和词组亦有着固定的自然顺序,无论手工或机器都可以进行一次到位的检索。

纵观汉字的发展史,汉字本来都有其结构成分的,因而本来就存在着自己的自然内码。在古老的甲骨文和金文中,“唯”由口隹两部分左右构成,“周”由用口两部分上下构成。我国最早的文字学家东汉的许慎在约两千年前成书的《说文解字》中指出:会意字“史”,从手持中;“休”,人倚于木;小土为“尘”;三人为“众”;形声字中“河”,我鸟读“鹅”。古代非正统文学著作《笑林广记》记述仙人张果老,吕洞宾和“白吃”饮酒拆字,把繁体“圣”“贤”“愁”拆分为耳口王,臣又贝,禾火心。时至今日,在人们的语言生活和汉字教学中亦常以木子代“李”,丘八指“兵”,八刀喻“分”,禾口王为“程”。汉字的结构成分包括笔划和笔划组合都可以读音或表音。金木水火土,是字,可读;、丿、㇏、㇀、㇁,也都有表音功能,俗称点,撇,草字头,三点水,衣字旁。它们统称为音元,所以从结构上看,汉字是由可读音的和可表音的结构成分—音元按一定的结构模式组成的,可以称作音元文字。音元就是汉字的自然内码,因为它解析了全部汉字的全部结构成分,并使之获得固定的自然顺序。

但是在此之前人们对汉字只解析到“独体为文,合体为字”的程度,合体字一般只分解部首、偏旁两部分,故凡识字者都会说日月为“明”,木子为“李”。至于独体字和合体字内部,则直接解析到笔划,并不解析其结构成分,因而不可能把全部汉字的全部结构成分解析出来。即使解析出来,如果不能采取科学的排序方法并最终排出各有唯一地址顺序,仍然不能说汉字有了自然内码。

汉字自从软书写工具毛笔取代原始的硬书写工具以来,汉字的结构成分—音元长期被笔划掩盖着,所谓“汉字由笔划构成”,“整字分解为字根,字根分解为笔划”。具有二千年历史的部首法,部首就是由笔划数和笔划形态双因素排序的,部首以外的字身也由笔划数排序,汉字的笔划数多少不一,最少者一划,最多者 30—50 多划,简体字平均约 10 划,繁体字平均约 14 划;

每个部首下同笔划数的字又相当多,如<现代汉语词典>中亻部有660多个字,而8、9、10划都有60个之多。故在同部首下同笔划数的字身往往是随机的而无固定的顺序。现时的大多数形码也以笔划为基础,即使能勉强处理国标两级汉字,而在处理大字符集时就显得捉襟见肘,在全字符集面前便寸步难行。

汉字元音码对字的结构解释为“汉字由音元构成”,笔划也是音元,如模式1。音元即汉字的自然内码,由于其音位已经与英文字母键位重合,故可对全部汉字进行不编码,无记忆量的输入和检索,实现了易学性与高效性的统一,其技术数据:码元26个英文字母,基本音元178个,一级字静态平均码长2.908,字词综合动态平均码长<2.0,一级字最大码长为4时静态同码率1%左右,对国标汉字6763个,中日韩统一编码大字符集20902个和全字符集6万个进行检索,最大码长为5时同码率为0。

三、元音法无同码检索的技术保证

科学意义的检索是指一次到位的成功查找。由于英文的语言和文字是同一的,故可实现不编码、一次到位的检索。中文字形的部检索法是部首加笔划数的双因素检索,除了麻烦性不计外,由于①部首有在上下左右的多种可能性,②部首和字身都需要查数笔划数,且不易查得准确,所以不能实现一次到位的准确检索。这犹如分析一部高级的电子设备,不拆取其部件并给予编码,却要从它的每一个零件查起。一切基于笔划或部首法的汉字编码难以实施对汉字的检索,道理就在于此。

元音法实施以音统元的原则,以26个字母摄汉语的420个音节,以420个音节统摄数万汉字,纲举而目张。借鉴于历史源流,以系统工程的理论为导向,在计算机上进行充分的模拟实验,元音法实现了国标汉字、大字符集和全字符集的同码检索,其技术保证是:

1. 对无明显分隔沟的字,采取从薄弱处以平面切割方式,而不是以立体叠架的拓扑方式拆取字元,使割取的音元具有直观可读性。“丈”“史”切割为十乚,中乚,符合<说文解字>关于这两字“从十从手”,“从手持中”的造字理论;“未”“末”切分为土小,土小,解决了迄今为止字典检索和形码输入土土不分的问题。

2. 把繁杂的汉字结构类型简化到只有四类十种:①独体字②二元上下交接型结构,二元上下分离型结构,二元左右分离型结构,③三元品字型结构,三元上下型结构,三元左右型结构,④多元混合型结构,多元上下型结构,多元左右型结构,从而简化了编码方法,分化了同码。

3. 确立各个结构类型字的顺序,使同码字在字典检索中不加区别码就获得固定顺序,在电脑输入和检索中只加固定的区别码。它们的升序是:

(i)二元字上下交接型<上下分离型<左右分离型:未<思,全<旺。

(ii)三元字品字型<上下型<左右型:茄<菩。

(iii)元字混合型<上下型<左右型:棹<罩。

4. 根据每个字的属性采取多因素编码,从而保证了任何同码字必因某项因素之相异而获得区别。元音法解析字的属性有:①字的级别分四级:通用字6763个,基本字13050个,大字符集20902个,全字符集约6—8万个;②字的音元个数,③音元的音位(首字母),④音元的首末笔划音位⑤字的结构类型,⑥字的使用频度。

四、汉字教学的崭新方法

我国当前中小学汉字教学和对外汉字教学都是以笔划作为字的结构单元的,需要学生对每个汉字从整体和细节上进行全面地掌握小学语文课本12册共3272个生字。每年平均学539.9个,每天平均学1.478个。由于采用的是分析教学法,对每个字都要求齐头并进地会认会写、会讲、会用,所以效率相当低,比之于古代师塾的集中识字法亦相差甚远。师塾法以具有相当人文知识的<三字文><百家姓><千字文>为识字教材,再配合一些精粹的诗文,要求在一年左右突破识字关,每天识字30个上下,写字不要求全会,只作为识字的辅助手段,事实上师塾对汉字之认、写、讲、用是采取分阶段进行的。

作为图形符号的汉字在开发人的智力方面有得天独厚的优势。它具有的逻辑结构与现代科学的逻辑结构如出一辙。清代文字学家王筠曾提出拆字法对汉字进行教学,他说:“识字之难在于不能分字,苟分一字为数字,则能执简驭繁,易记而难忘矣。”至于<笑林广记>叙述拆分“圣”“贤”“愁”的传说,可能是古代一种寓教于乐的汉字教学方式的遗存。而<三国演义>记载杨修把“绝妙好辞”暗射为“黄娟幼妇,外孙齏臼”,则可能是这种拆字法的滥觞。我国著名科学家钱伟长教授在概括汉字的特征时指出:“汉字的形态结构特征是以联想联系起来的……人们也是通过联想认字的。”它既有形,又有音,能使人起双重联想,现在元音法把汉字的自然内码解析出来,如果进一步采用新的系统教学法,对于认读汉字和手工书写或在计算机上书写汉字,必将产生突飞猛进的效果。

元音法对小学汉字教学的构想(对外汉字教学的方法略同)是:

1. 对幼儿园大班首先教200个左右的基本音元和简码,会认,并在电脑上会写,使他们具备拆分所学汉字的基本能力。当他们遇到生字“钢笔”二字时,就可以辨认出是由专门乚,竹毛两组音元组成的,然后让他们到计算机上换成JMC,ZM,就写出了这两个字。

2. 一年级在按正常课程学好汉语拼音或拼音中文的同时,学会用笔书写200个音元,学会切割所学的汉字。这样他们在纸面上写字也比传统的写字方法快了,因为传统的写字方法对字的结构成分并不明确,多数是一笔划,一笔划地教,学生很难掌握每个字中的笔划关系。于是他们写字不再念叨竖撇点折,而是念叨左边金字旁—右边门框形——里面一个义,竹字头—下面一个毛,快速念就是金门框义,竹毛,显然这种用分

子构造物质的方法比用原子构造物质高效得多了。

3. 三年级在按正常课程学习拼音查字法和部首查字法的时候,早已因二年级的熏陶,会用电脑或用手工检索元音法词典。现在只需进一步从理论上认识元音法的系统工程性质,以坚定日后用元音法写作和处

理汉字信息的信念。

可以预见,当古老而有生命力的汉字与现代的信息载体英文字母结合起来并形成系统工程之后,既保持了丰富的音形义信息内容,又获得易于认记、书写和传输的新功能,将如虎添翼,加速奔向新世纪的进程。

现货可供美国 BOURNS 公司网络电阻

宁波国际工贸实业有限公司,现货可供各种规格美国 BOURNS 公司生产的各种型号,各种阻值的单排直插(SIP),双排直插(DIP),表面贴装(SMD)网络电阻,本公司可竭诚为用户提供周到的服务,批发、零售、邮购,任君选择。

规 格	零 售	批 发
单排直插(SIP)	8 分/每脚	4.5 分/每脚
双排直插(DIP)	16 分/每脚	12 分/每脚
表面贴装(SMD): 001,002 系列	6 元/每只	4 元/每只
003 系列	9 元/每只	6 元/每只

联系人:蒋佩勇先生 赵端先生

邮编:315000

联系地址:宁波市大沙泥街 61 号 207 房间

电话:7302490 传真:(0574)7302759

·指定范围 VGA 图像彩色拷贝

方 震 复旦大学计算机系 9024[#](200433)

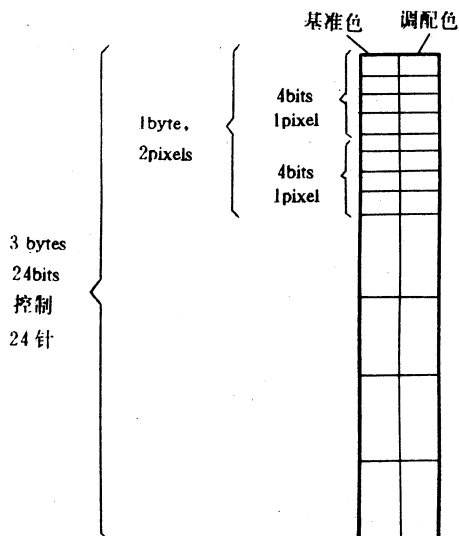
目前彩色打印机和使用彩色色带的行式打印机已比较普及,用户常希望能够将屏幕上精美的彩色画面拷贝下来。现有市售软件多不能尽人意,主要有三个缺点:1. 不支持 VGA 中高分辨率的彩色打印;2. 打印稿的色彩逼真性、清晰度等欠佳;3. 只能全屏拷贝,不能由用户指定打印范围。

笔者针对上述问题编写了一个“指定范围 VGA 图像彩色拷贝”程序,在富士通打印机(兼容 LQ-1600K)上用四色色带调试通过,效果良好。本文着重探讨将内存图像数据向打印机控制数据转换的方法,因而具有普遍意义。所附源程序加有详细注释,读者可以方便地根据自己的打印机控制码格式及 VGA 其他显示模式作出修改。由于版面篇幅限制,只给出程序的核心部分,其他部分如输入打印范围、打印机状态检测等子程序,只给出必要的出入参数说明。本文假设读者已具有 VGA 显示原理方面的基本知识。

彩色屏幕图像的打印拷贝关键有两点:一、长宽比例不能明显变形;二、要利用有限的色带颜色调配出各种屏幕颜色,使色彩逼真。下面按照 VGA 模式 12H 和比较普及的 24 针打印机配 4 色色带讨论这两个问题。

24 针打印机的针头纵向排成一列,分别受控制码 3 个字节的 24 位控制。选定合适的打印密度,即针头

列间距(这选择主要考虑图稿尺寸大小、打印速度、打印成本等因素),剩下的问题就是确定每个屏幕像素点用纵向的多少针绘制。例如,本程序用的是 4 针,这样,每一打印机行对应 $24/4=6$ 个像素行。按照本程序所



选打印密度,并且采用下面将叙述的配色方法,能够很好地满足长宽比例的要求,打印稿的尺寸也适中。

目前市售彩色打印色带多为红、黄、蓝、黑 4 色,通过重叠打印可再得到橙、绿、紫 3 色,我们要使用这 7 种颜色(本质上是红黄蓝 3 种颜色),基本达到 VGA 16 色的视觉效果。笔者采用的方法,是先打印基准色,再平行地打印一系列所谓“调配色”,见附图。比如,品红色由一系列紫色和一系列红色模拟实现,浅蓝色则是蓝色加一系列空白。具体的实现,是以 VGA 颜色值为索引,分别查基准色表和调配色表得到控制打印机的基准色和调配色数据。

下面对源程序作一些说明。基本流程是从视频存储区读出一组(6 行像素)数据,第一遍扫描记录下该组中需用哪几种颜色,然后依次打印这些颜色。子程序 read_scope 读入用户欲拷贝的屏幕范围;posi-cal 根据这范围计算出 left、right、upper、bottom、n1、n2 等重要参数;prn-prepare 完成打印机检测和初始化。程序修改 5 号中断并驻留内存,按下〈Printscreen〉键将启动打印。

这个程序把重点全部放在图像拷贝上,因而并不完善。读者可自己改善用户界面,改永久占用内存为随使用随释放,等等。

```

        endm
-code segment byte public 'code'
        assume cs:_code,ds:_code
        org 100h
        .286
-start: jmp     _set-int
pix-group dw 0 ; 0..79
row-base dw 0 ; upper..bottom, 步长 6
; 6 * 640 = 3840 屏幕像素的基准色和调配色数据
save-pix db 3840 dup(?)
; 记录 6 行像素出现的颜色
blend-save db 3840 dup(?)
fact-colors db 8 dup(?)
line-no dw 0 ; 0..5
ah-save db ?
plane-no db 0 ; VRAM 位平面号 0..3
by3 db 3 dup(?) ; 基准色打印数据
bl3 db 3 dup(?) ; 调配色打印数据
blend-byte db 0
current-color db ? ; 0..6
; 基准色表
color-table db 0,2,6,2,1,3,5,7
              db 0,2,6,2,1,3,4,7
; 调配色表
blend-table db 0,2,6,7,1,1,5,7
              db 0,2,7,7,1,1,4,7
density db 38
n1 db 0
n2 db 5
n db 25
byte-per-row dw 80
upper dw ? ; 0..bottom
bottom dw ? ; upper..479-6
left dw ? ; 0..right
right dw ? ; left..639
right dw ?
pix-start dw ?

send-1 macro xx ; 送一字符至打印机
        mov ah,0
        mov al,xx
        int 17h
        endm

send-2 macro xx,yy
        xor dx,dx
        send-1 xx
        send-1 yy
        endm

send-3 macro xx,yy,zz
        xor dx,dx
        send-1 xx
        send-1 yy
        send-1 zz
        endm

make-3840 proc ; 读入 6 行像素点的
                VGA 颜色值
        mov bx,right
put-zero: mov save-pix[bx],0
        dec bx
        cmp bx,left
        jge put-zero
        mov dx,3ceh
        mov al,4
        out dx,al
        mov di,left
        mov dx,pix-start ; 0 <= x/8
                <= 79
        mov pix-group,dx
        mov ah-save,80h ; 屏蔽器
nxt-col: mov line-no,0 ; 0..5
        xor ax,ax
nxt-line: add ax,row-base ; line-no +
                row-base
        xor dx,dx
        mul byte-per-row
        add ax,pix-group
        mov si,ax ; si=y*80+x/8
        mov dx,3cfh
        mov plane-no,0 ; 0..3
nxt-plane: mov al,plane-no
        out dx,al
        mov al,es:[si] ; 从 VRAM 读
                一字节
        and al,ah-save ; 得到所需位

        cmp al,0
        jz bit-fixed
        mov al,1
        mov cl,plane-no
        shl al,cl
        or save-pix[di],al
        bit-fixed: inc plane-no
        cmp plane-no,3
        jle nxt-plane
        inc di
        inc line-no
        mov ax,line-no
        cmp ax,5
        jle nxt-line
        mov ah,ah-save
        shr ah,1
        jnz mask-not-0
        mov ah,80h
        inc pix-group ; inc x/8
mask-not 0: cmp di,right
        jg make-end
        mov ah-save,ah
        jmp nxt-col
make-end: ret
make-3840 endp

convert-3840 proc
        xor ah,ah
        mov bx,left
convert-on: mov al,save-pix[bx]
        mov si,ax ; 查表用的索引
        mov al,color-table[si]
        mov save-pix[bx],al ; 基准色
        mov al,blend-table[si]
        mov blend-save[bx],al ; 调配色
        inc bx
        cmp bx,right
        jle convert-on
convert-end: ret
convert-3840 endp

make-fact-color proc ; 记录下实际需要
                哪几种颜色
        mov di,7

```

```

init-factcolor: mov fact-colors[si], '$'
                dec di
                jge init-factcolor
                mov di, 0
                mov ah, 0 ; 由黑色开始
see-through:   mov si, left
search-on:     mov al, save-pix[si]
                cmp ah, al
                jz fine-a-color ; 有此颜色
                mov al, blend-save[si]
                cmp ah, al
                jz find-a-color
                inc si
                cmp si, right
                jg inc-color
                jmp search-on
find-a-color:  mov fact-colors[di], ah
                inc di
inc-color:     inc ah ; 下一种颜色
                cmp ah, 7
                je back-out
                jmp see-through ; 再一遍扫描
back-out:     ret
make-fact-color endp

deal-3840 proc
    mov si, left
    mov dl, 0fh, 0000 1111
    xor bx, bx; bx: 0..2
next-byte:     mov ch, 0 ; ch: 基准色
                mov blend-byte, 0; blend-byte: 调配色
next-pix:     not dl
                mov al, save-pix[si]
                cmp al, current-color
                jnz not-base-color
                or ch, dl
not-base-color: mov al, blend-save[si]
                inc si
                cmp al, current-color
                jnz not-blendcolor
                or blend-byte, dl
not-blendcolor: cmp dl, 0fh
                jne next-pix
                mov by3[bx], ch
                mov ch, blend-byte
                mov bl3[bx], ch
                inc bx
                cmp bx, 2
                jle next-byte
                push dx
                xor bx, bx
send-3 by3[bx], by3[bx+1], by3[bx+2]; 打印基准色
send-3 bl3[bx], bl3[bx+1], bl3[bx+2]; 打印调配色
                pop dx
                cmp si, right
                jg deal-end ; si, left..right
                jmp next-byte
deal-end:     ret
deal-3840 endp

print-group proc; 打印机的一行, 屏幕上的六行像素
                call make-3840; 得到6行像素数据
                call convert-3840; 转换成打印机颜色数据
                call make-fact-color; 记下出现的颜色
                mov si, 0
next-color:   mov cl, fact-colors[si]; 取下一个颜色
                cmp cl, '$'
                je a-grp-printed
                mov current-color, cl
                send-3 27, 114, cl; 设定颜色
                send-3 27, 42, density; 打印密度
                send-2 n1, n2; 字节数 = n1 + n2 * 256
                push si
                call deal-3840
                mov dx, 0
                send-1 13 ; 回车但不换行
                pop si
                inc si
                jmp next-color
a-grp-printed: mov ds, ax
                call read-scope; 读入拷贝范围(略)
                call posi-cal; 计算参数(略)
                mov dx, offset scr-copy; 自编中断服务程序
                mov al, 05h
                mov ah, 25h
                int 21h ; 修改 int 05h
                mov dx, offset last-byte
                mov cl, 4
                shr dx, cl
                add dx, 10h
                mov ax, 3100h
                int 21h ; 驻留内存
                code ends
                end -start
scr-copy proc
    sti
    pusha
    mov ax, cs
    mov ds, ax
    call prn-prepare; 打印机初始化(略)
    mov ax, 0a000h
    mov es, ax; 视频缓冲区(VRAM)段地址
    mov ax, upper
    mov row-base, ax ; 0..474
nxt-row-grp:  call print-group
                send-3 27, 74, n; n/180英寸换行
                mov ax, row-base
                add ax, 6
                mov row-base, ax
                cmp ax, bottom
                jle nxt-row-grp; 下一组(6行像素)
                popa
                iret
scr-copy endp
last-byte:
    set int; mov ax, cs
    mov ds, ax
    call read-scope; 读入拷贝范围(略)
    call posi-cal; 计算参数(略)
    mov dx, offset scr-copy; 自编中断服务程序
    mov al, 05h
    mov ah, 25h
    int 21h ; 修改 int 05h
    mov dx, offset last-byte
    mov cl, 4
    shr dx, cl
    add dx, 10h
    mov ax, 3100h
    int 21h ; 驻留内存
    code ends
    end -start

```

一个实用的彩色图符编辑器程序

吕学山 中国金融学院经济信息系(100029)

图符(icon)式界面技术的使用日益广泛,它是用一个个形象化的小图符来表达系统的功能。图符界面是否友好,关键在于所绘制的图符能否形象化地表达

该选项所对应的功能,因此图符的绘制非常重要。

下面给出的彩色图符编辑器程序为程序开发人员提供了良好的图符绘制环境。它具有绘制、保存、调出

和删除图符的功能,操作简便、直观、所见即所得,所以它是编程人员开发图符菜单的得力工具。

该程序用到的主要函数及功能分述如下:

1. 初始化图符函数 `init-icon()`,将保存图符信息的二维数组的值,初始化为背景颜色(灰色)值。

2. 显示图符函数 `display-icon()`,显示一个背景为灰色的图符基座,为绘制图符提供一个背景环境。

3. 编辑图符函数 `edit-icon()`,通过八个方位的光标操作,在图符基座上绘制出所需的彩色图符。

4. 选择色笔函数 `switch-color()`,在屏幕下方形成一条彩色带,为选择不同色笔提供一个友好界面。

5. 保存图符函数 `save-icon()`,将用户绘制的图符保存到图符文件中。

6. 调出图符函数 `load-icon()`,将图符文件中的图符调出显示在图符基座上。

源程序 EDICON.C 经过 Turbo C 2.0 编译、连接,生成 EDICON.EXE 文件即可使用。本程序在长城 286 机,CEGA 显示卡上运行通过。

```
#include <stdio.h>
#include <dos.h>
#include "bios.h"
#include "conio.h"
#include "string.h"
#define IX 300
#define IY 164
#define XDIM 40
#define YDIM 32
#define BCK_GND 8
#define FORE_GND 15

void mode(),edit-icon();
void display-icon(),display-operate-key();
void init-icon(),putpoint();
int switch-color(),save-icon(),load-icon();
char getkey();
struct icon_type{
    unsigned char image[XDIM][YDIM];
    void (*func)();
    int x,y;
}icon;
char pen,pe1,far *egabase;

int main(){
    mode(16);
    egabase=(char far *)MK_FP(0xa000,0x0000);
    directvideo=0;textattr(0x6e);
    gotoxy(24,6);
    cprintf("彩色图符编辑器");
    init-icon();
    edit-icon();
    mode(3);
    return 0;}

void edit-icon()
{ register int x,y;
```

```
char ch;
x=IX;y=IY;
pen=BCK_GND;pe1=BCK_GND;
display-operate-key();
display-icon(300,IY);
do{ icon.image[x-IX][y-IY]=pen;
    putpoint(x,y,pe1,0);
    ch=getkey();
    switch(ch){
        case 75:
            if(pe1==14)putpoint(x,y,pen,0);
            if(pe1==14)x++;break;
        case 77:
            if(pe1==14)putpoint(x,y,pen,0);
            if(pe1==14)x++;break;
        case 72:
            if(pe1==14)putpoint(x,y,pen,0);
            if(pe1==14)y--;break;
        case 80:
            if(pe1==14)putpoint(x,y,pen,0);
            if(pe1==14)y++;break;
        case 71:
            if(pe1==14)putpoint(x,y,pen,0);
            if(pe1==14){y--;x--;}
            if(x<IX)y++;if(y<IY)x++;
            break;
        case 73:
            if(pe1==14)putpoint(x,y,pen,0);
            if(pe1==14){y--;x++;}
            if(x>IX+XDIM-1)y++;if(y<IY)x--;
            break;
        case 79:
            if(pe1==14)putpoint(x,y,pen,0);
            if(pe1==14){y++;x--;}
            if(x<IX)y--;if(y>IY+YDIM-1)x++;
            break;
        case 81:
            if(pe1==14)putpoint(x,y,pen,0);
            if(pe1==14){y++;x++;}
            if(x>IX+XDIM-1)y--;if(y>IY+YDIM-1)x--;
            break;
        case 82:
            pe1=14;pen=8;break;
        case 83:
            pe1=8;pen=8;break;
        case 46:
            switch-color();break;
        case 31:
            save-icon();break;
        case 38:
            if(!load-icon())
                {getch();
                 gotoxy(1,22);clrscr();
                 gotoxy(1,23);clrscr();}
            display-icon(300,IY);break;
        case 32:
```

```

init_icon();
display_icon(300,IY);break;}
if(x<IX)x++;if(x>IX+XDIM-1)x--;
if(y<IY)y++;if(y>IY+YDIM-1)y--;
}while(ch!=1);
}

void display_operate_key()
{textattr(0x10);colour(22,1,24,79,0x00);
gotoxy(1,22);clreol();textattr(0x1f);
cprintf("<HOME><↑><PGUP>打开笔盒<INS>保存图符<S>删除图符<D>")
gotoxy(1,23);
cprintf("<←>      <→>      退    出<ESC>")
gotoxy(1,24);
cprintf("<END><↓><PGDN>关闭笔盒<DEL>调出图符<L>选择色笔<C>")
}

```

```

void display_icon(int startx,int starty)
{register int x,y;
for(y=starty;y<starty+YDIM;y++)
for (x=startx;x<startx+XDIM;x++)
putpoint(x,y,icon.image[x-startx][y-starty],0);
}

```

```

void init_icon()
{register int x,y;
for(x=0;x<XDIM;x++)
for(y=0;y<YDIM;y++)
icon.image[x][y]=BCK-GND;}

```

```

int save_icon()
{FILE *fp;
char fname[80];int result;
textattr(0x0f);colour(21,0,23,79,0x00);
gotoxy(28,22);cprintf("保存图符到文件:");
gets(fname);
if((fp=fopen(fname,"wb"))==NULL){
colour(21,0,23,79,0x00);
gotoxy(28,22);
cprintf("不能打开文件,按任意键继续!");getch();
display_operate_key();
return 1;}
fwrite(&icon,sizeof(icon),1,fp);
if(ferror(fp))result=0;
else result=1;
fclose(fp);
display_operate_key();
return result;}

```

```

int load_icon()
{FILE *fp;
char fname[80];int result;
textattr(0x0f);colour(21,0,23,79,0x00);
gotoxy(28,22);cprintf("调出图符文件从:");
gets(fname);

```

```

if((fp=fopen(fname,"rb"))==NULL){
colour(21,0,23,79,0x00);
gotoxy(28,22);
cprintf("不能打开文件,按任意键继续!");getch();
display_operate_key();
return 1;}
fread (&icon,sizeof(icon),1,fp);
if(ferror(fp))result=0;
else result=1;
fclose(fp);
display_operate_key();
return result;}

```

```

char getkey()
{union key{
int i;
char ch[2];
}k;
k.i=bioskey(0);
return k.ch[1];}

```

```

void mode(int mode-code)
{union REGS r;
r.h.al=mode-code;r.h.ah=0;
int86(0x10,&r,&r);}
void putpoint(int x,int y,int color,int row)
{union REGS r;
if(how==0x18)color=color|128;
r.h.ah=12;r.x.dx=y;r.x.cx=x;
r.h.bh=0;r.h.al=color;
int86(0x10,&r,&r);}

```

```

int colour(int x1,int x2,int y1,int y2,int z)
{union REGS r;
r.h.ah=6;r.h.al=0;
r.h.ch=x1;r.h.cl=x2;
r.h.dh=y1;r.h.dl=y2;r.h.bh=z;
int86(0x10,&r,&r);}

```

```

int switch_color()
{char ch;int i,c=1;
textattr(0x0f);colour(21,0,23,79,0x00);gotoxy(1,22);
for(i=3;i<79;i+=8)
{gotoxy(i,22);
cprintf("<F%d>",c);c++;}
colour(22,0,22,7,0x00);colour(22,8,22,15,0x11);
colour(22,16,22,23,0x22);colour(22,24,22,31,0x33);
colour(22,32,22,29,0x44);colour(22,40,22,47,0x55);
colour(22,48,22,55,0x66);colour(22,56,22,63,0x77);
colour(22,64,22,71,0xee);colour(22,72,22,79,0xff);
gotoxy(36,24);cprintf("请选择色笔:");
do{
ch=getkey();
switch(ch){
case 59:pen=0;break;
case 60:pen=9;break;

```



```

case 61;pen=10;break;
case 62;pen=11;break;
case 63;pen=12;break;
case 64;pen=13;break;
case 65;pen=6;break;
case 66;pen=7;break;

```

```

case 67;pen=14;break;
case 68;pen=15;break;}
while(ch!=59&&ch!=60&&ch!=61&&ch!=62&&ch!=
63&&ch!=64&&ch!=65&&ch!=66&&ch!=67&&ch!=
68);
display_operate_key();

```

充分利用 DOS 的扩展内存来运行大型程序

张晓莉 河北大学(071002)

目前,DOS 3.x 操作系统仍被许多微机用户所使用。尽管常用的 286、386 等中、高档微机的内存有 1M 以上,但 DOS 所管理的基本内存只有 640K,加上 DOS 启动后本身占据一部分,如果用户程序还要用其它的辅助功能如脱机打印、屏幕拷贝、语音系统等,这些应用程序也要驻留内存,为用户提供运行程序的空间就更少了。因此若用户的应用程序过大或数组过多,则 DOS 提示出现“Program too big in memory”或“out of string space”等基本内存已不能满足程序运行的信息。如果程序运行在内存临界状态,虽然不出现以上提示信息,但也许后果更为严重,往往出现死机现象。为了解决这一问题,常规的做法是调整算法,尽量压缩程序的长度,压缩数组,模块之间使用 COMMON 语句共享内存数据区等方法,但这些方法往往难以奏效。下面介绍的是利用 DOS 的扩展内存建立虚拟磁盘(虚盘)的方法来解决这一问题。

一、利用扩展内存创建虚盘

640K 基本内存以外的内存称为扩展内存。我们利用 DOS 文件 VDISK.SYS 设备驱动程序来安装虚盘,方法是在 DOS 的配置文件 CONFIG.SYS 中增添一行 DEVICE 命令,即:

```
DEVICE=C:\DOS\VDISK.SYS 1024 256/E
```

当有此设置之后,重新启动计算机则 DOS 系统自动在扩展内存建立了一个容量为 1024K 字节,扇区为 256 字节可包含 64 个文件的虚盘。

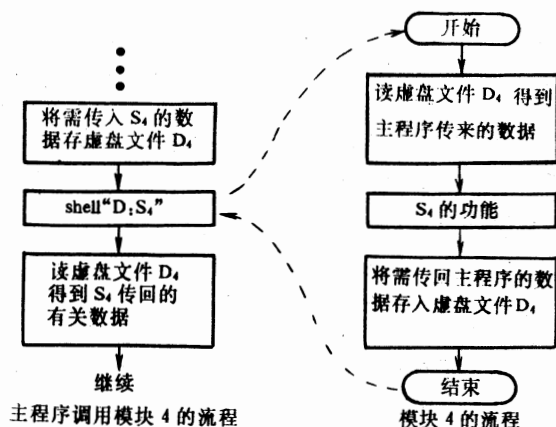
需要注意的是虚盘是在扩展内存的低字节开始的,因此若用户程序需安装汉字系统时,需将字库放置在扩展内存中虚盘的后部,否则将发生冲突,影响汉字的使用而产生错误。

读写虚盘的速度实际上是读写内存的速度,比硬盘快得多。虚盘的盘号是根据当前系统的盘号按字母顺序排下来的,比如现在最后一个硬盘号为 C,则刚建立的虚盘号为 D。

二、把部分程序存入虚盘

在实际编程中,总是把整个问题按功能分为不同的模块,单独编译,然后链接,最后形成一个“.EXE”文件。按模块编程是为了提高程序的质量,并不会减少程

序的体积,因为最终形成的一个“.EXE”文件都要调入内存才能运行。但为了解决内存不够的问题就要改变上面的一些方式,不形成一个“.EXE”文件,而是根据具体情况形成几个“.EXE”文件,例如有一个应用程序包含 MAIN,S1,...S6 七个模块,主模块 MAIN 起主导作用,它下面有六个子模块 S1—S6,其中,S1、S2、S3 实时性要求较强,而模块 S4、S5、S6 实时要求差一些,这样我们把主模块和子模块 S1、S2、S3 链接在一起形成一个“.EXE”文件,长期驻留内存,以便及时调用,而把子模块 S4、S5、S6 单独编译、链接形成独立的 3 个“.EXE”文件即 S4.EXE、S5.EXE、S6.EXE,在应用程序开始运行时先调用一个初始化过程,把 S4.EXE、S5.EXE、S6.EXE 拷贝到虚盘,当需要运行某一模块如模块 4 时,主程序通过执行 shell 命令可以暂回 DOS 下,运行 S4.EXE,因此主程序在调模块 4 时命令为 shell“D:m4”(这里假设虚盘为 D 盘),而不是用 CALL 命令来实现模块的调用,这样在不退出内存程序的情况下暂回 DOS 运行 S4.EXE,运行完毕又回到了主程序调用处继续运行,而刚才 S4.EXE 运行所占的内存又归 DOS 所有了。以同样的方法实现对模块 5 和模块 6 的调用,这样只要 S4、S5、S6 不同时运行,那么留给它们的内存只需满足三者中最大者就可以了,因此这个一共七



个模块的程序某时刻只有5个模块在内存,这样就大大减少了程序在内存中的占有量。因为模块4、5、6是独立的三个“.EXE”文件,因此主程序和它们的数据交流不能象模块1、2、3那样靠参数传递了,只能通过文件来传送,为了提高速度,把这种文件也建在虚盘上。以主程序调用模块4为例给出上述调用过程如图所示。

三、大量的初始数据也存储在虚盘,大大节约了数据空间

DOS 提供给用户的数据区只有64K,对于大的应用程序来讲,这点数据空间是不够的。例如某实时监测系统运行时需要变电站的128个遥测量的名称、系数、类别、上限值、下限值,而这样的变电站可有若干个。比如现有20个,而这些数据若常驻内存,仅此一项数组体积要定义为 $20 \times 5 \times 128$,按实型数计则要占内存约

50K之多,但由于每时刻只需一个站的数据,因此在初始化时把所有站的相同性质的数据存入一个随机文件放置虚盘,如把各个站的系数按站号、遥测号的顺序按每个记录16个数据的形式存在虚盘上,因为有了这样的规律,可以很方便的得到某个站某个遥测的系数,而内存数组只需定义一个站大小,这样数据空间减少了19/20。例如按上述规律想取出X站Y遥测的系数值,只需读虚盘上的系数文件,这个数据的记录号为 $(X-1) \times 8 + \text{INT}((Y-1)/16) + 1$,这个记录中的第 $(Y-1) \bmod 16 + 1$ 个数据就是该系数值,由此可见,这种方法既节省了数据空间又不影响运行的速度。

通过本人具体的实践,证明这种方法是一个很好的、很有效的方法,既解决了内存不足的问题,又对运行速度没有很大影响。

在 MS-DOS 高版本下安装汉字系统

李晓中 宋 涛 北京市海淀区闵南里9号(100080)

近二年在不同刊物上发表了不少有关在 DOS 5.0 或 6.0 下如何安装汉字 2.13 或 UC DOS 系统的文章,为读者提供了许多灵活、实用的方法。看后很受启发,但也有局限性。如果按其中某一篇文章去安装汉字系统总是不能完全通过。笔者在收集众多技巧和亲身体验后,认为要安全、可靠、正确地使用 2.13H 或 UC DOS 汉字系统必须从以下两个方面加以考虑:

1. 设备驱动程序 ANSI.SYS 必须使用低版本的,不能使用高版本(DOS 4.0 以上)的。ANSI.SYS 用来代替标准 BIOS 输入/输出支持软件,它具有如下功能:

- ①重新定义键盘上的某些键
- ②控制光标
- ③控制显示属性,改变图形功能
- ④支持用“MODE”命令改变屏幕上的行数。

CCDOS 2.13, UC DOS 2.0 所用的设备驱动程序 ANSI.SYS 当时是基于 DOS 3.30 以下版本的,而 DOS 5.0 及 DOS 6.0 配备的 ANSI.SYS 与之相比在长度上成数倍的增长(DOS 3.30 为 1647B, DOS 5.0 为 9029B)并增强了许多功能。如:键盘重定义功能、定义灰键功能、小键盘功能。显示功能也增强了许多,如在图形方式(方式6)下,显示页增加到八页,显示方式设置也增加了 EGA/VGA 的许多新方式。而 2.13H 或 UC DOS 2.0 的汉字系统不具备支持这些扩充功能的模块。最终导致了高版本 DOS 下启动汉字系统的“死机”现象。有些文章指出系统配置中加入 ANSI.SYS/K, SWITCHES=/K 命令便可解决死机现象,事实上这只解决了封锁 DOS 5.0 扩展键盘的功能,但并未解决显示方式的不兼容问题。为了适应汉字系统而修改高版本的设备驱动程序 ANSI.SYS 是不合算的。笔者

建议使用汉字系统的用户,最好在自己的启动盘系统配置文件中,使用低版本的 ANSI.SYS 设备驱动程序。

2. 关于汉字系统在虚拟盘(扩充内存区)的起始位置,为了节省常规内存, DOS 5.0 以上将 DOS 的核心部分装入高内存区(HMA),而汉字库将放入扩充存储区,即虚拟盘 E(或 C:)中。CCDOS 2.13H 使用 FILE-3.COM 来确定汉字库虚拟盘在高内存区中的位置。如何才能正确地调出字库的启动位置,笔者经多次实践体会,认为根据系统配置的情况应按以下方法计算:

字库(虚拟盘)在内存中的首地址由三部分组成即:

$$A+B+C$$

①A=1MB 即系统的常规内存区和上端内存块(UMB)

$$640K+384K=1MB$$

②B 由 DEVICE=HIMEM.SYS 命令设置,只要建立扩充内存管理程序,则无论使用 DOS=HIGH 与否,系统将为 DOS 自动留有 64KB 的高内存区(HMA),而并不像有些文章所说的那样,由 DOS=High 将占有 64K 高内存区;DOS=HIGH, UMB 将占有 512KB 的高内存区。当然,要建立扩充区的虚拟盘,必须使用扩充内存管理程序 HIMEM.SYS。那么这 64KB 必然要为 DOS 保留。如果使用 VDISK.SYS 命令建立虚拟盘的话也就不存在 64KB 的 HMA。

③C 由 CONFIG.SYS 中的 RAMDRIVE 语句之前的 SMARTDRV.SYS 或 EMM386.EXE 等管理分配内存的命令来确定,如果 RAMDRIVE 虚拟盘建立命令放到其它内存管理命令之前,就不存在 C 部分。

经上所述,一般的系统配置不外乎三种情况。

①在 CONFIS.SYS 中不用 HIMEM 建立扩充虚拟盘,使用 VDISK.SYS 则虚拟盘从 1MB 开始,无需修改 FILE3.COM 即可。

②使用了 HIMEM.SYS 管理扩充内存区,并用 RAMDRIVE.SYS 建立了虚拟盘,在 RAMDRIVE 命令之前无其它内存管理命令,则虚拟地址为:

$$A+B=(1\text{MB}+64\text{KB})/64\text{KB}=11\text{H}$$

③在 RAMDRIVE 之前增加了 EMM386、SMARTDRV 等命令,则要看实际使用内存来计算。

为了方便读者理解,不妨列举一例:

在 3.6MB 内存的 386 微机上配置出较好的 CONFIG.SYS 和针对 FILE3.COM 的修改方法。

CONFIG.SYS

DEVICE=A:\SYS\SETVER.EXE

DEVICE=A:\SYS\ANSI.SYS

DEVICE=A:\SYS\HIMEM.SYS

DEVICE=A:\SYS\EMM386.EXE 512 FRAME=

D000 RAM

DOS=HIGH,UMB

DEVICE=A:\SYS\SMARTDRV.SYS 1536 1536

DEVICE=A:\SYS\RAMDRIVE.SYS 320 512 64/E

FILES=35

BUFFERS=20

在以上配置中:

$$A=1\text{MB} \quad B=64\text{KB} \quad C=(512\text{KB}+128\text{KB})+$$

$$1536\text{KB}=2176\text{KB}$$

$$\text{开始虚拟地址为: } 1024\text{KB}+64\text{KB}+2176\text{KB}=3264\text{KB}$$

$$3136\text{KB}/64\text{KB}=51 \quad \text{即 } 33\text{H}$$

改动 FILE3.COM

DEBUG A:\213\FILE3.COM ↓

-E 027A ↓

XXXX:027A 10. 33 ↓

--W ↓

--Q ↓

做完上述操作后,重新用 A 盘启动,就可以在自己的 386 微机上使用 2.13H 汉字系统了。

说明:

C 中的 512KB 为 EMM386 建立的仿真扩展内存区;128KB 为 RAM 开关管理的高内存块,

FRAME=D000 EMM386.EXE 可使用的 UMB 地址范围:D0000—EFFFFH 128K

FRAME=C000 EMM386.EXE 可使用的 UMB 地址范围:C0000—C7FFFH 32KB

D0000—EFFFFH 128KB

共计 B128K+32K=160KB

FRAME=E000 EMM386.EXE 可使用的 UMB 地址范围:E0000—EFFFFH 64KB

(一般 AMI 主板的 PC 机可用的 UMB 最大空间为 160K,如果带有 ROM BIOS SETUP 配置的机器,将减少 64K 的可用空间)

UMB 地址范围不足 64KB 时(如 32KB),不能用上述方法修改 FILE3.COM 的 027AH 单元,027AH 处指令为 ADC DX,10H,DL 用来间址虚拟盘的高 8 位;并将结果存放在 0122H 处,而低 16 位地址用 AX 来间址,并存放在 0124H 处。详细的修改方法为:

原 FILE3.COM 计算虚拟盘首地址程序段为:

XXXX:0216 MOV AX,[SI+1AH],取 HZK16 的首族号

SUB AX,2,减去两个保留族号=实际族号

MOV DL,DS:[30DH],每族扇区数

XOR DH,DH

MUL DX,DX=从数据区开始的相对扇区号

MOV DX,200H,每扇区长度=512 字节

MUL DX,DX:AX=起始字节长度

ADD WORD PTR DS:[124H],AX,计算 HZK16 存放的首地址的低 16 位,数据区首址已保存在 0124H 中,在此仅需将计算的低 16 位加到 0124H 中。

XXXX:027A ADC DX,10H,加基数 1MB,若低 16 位有进位,

MOV BYTE PTR DS:[122H],DL 则高 8 位加 1,并存放在 0122H 处,DL,AX 组成 24 位,因此 DL=10H 即表示 1MB。

...

...

修改后的 FILE3.COM 计算虚拟盘首地址程序段为:

XXXX:0261 MOV AX,[SI+1AH],

SUB AX,2,

MOV DL,DS:[30DH],

XOR DH,DH,

MUL DX,

MOV DX,200H,

MUL DX,

ADD WORD PTR DS:[124H],AX,

XXXX:027A ADC DX,10H,

MOV AX,8000H,8000H 为 32K,用户根据实际情况来确定

ADD WORD PTR DS:[0124H],AX,32K 加到低 16 位 0122H 处

ADC DX,00H,若低位有进位,高位加 1

MOV BYTE PTR DS:[122H],DL,高位存到 0122H 处

...

...

详细描述请参阅 EMM386 内存管理资料。

CCED 数据计算中的出错处理

刘士杰 陈依平 北京燕郊有色金属管理干部学院(101601)

中文字表编辑软件 CCED 的主要特点之一,是将文字处理,表格处理与数据计算融为一体。其数据计算简单、方便、容易掌握,但在计算过程中由于操作不当,可能出现某些错误,使计算不能正常进行。由于 CCED 对计算中的出错没有出错提示,因此用户在出现错误时不知错在何处,应该如何处理。为使用户能在数据计算中正确地查错、排错,顺利完成计算任务,下面介绍 CCED 计算中经常出现的一些错误及其解决方法。

一、表列求和计算的出错处理

1. 用 Ctrl+S 对光标所在表列数据求和时,虽操作正确,但数据没有求和。

这种错误的原因一般是光标所在表列的宽度不够,以至存放求和结果的表格单元放不下求和结果。

解决的方法是,将光标置于该表列内,按 F6 键加宽该表列至能放下求和结果为止。加宽后再次使用求和命令,即可完成求和任务。

2. 求和结果小于该表列所有实际数据之和。

此时应检查两点:

(1) 定义行块时,是否将该表列所有要求和的数据包括在内,若没全包括在内,应用 F8 键重新定义行块,再进行求和计算。

(2) 若行块定义正确,应检查光标是否在该表列所有数据的左侧(至少在最左边数字所在列上),否则应将光标移至存放结果表格的最左边(至少在最左边数字所在列上),再按 Ctrl+S 键重新计算。

二、同一行内数据按公式计算的出错处理

1. 写好公式给出计算命令后,计算不出结果。

此时,应从下面四点考虑:

(1) 查看公式书写是否有误,如是否插入了不必要的空格。将圆括号写成了方括号、是否在全角下输入(机器只识别在半角下输入的公式)等。

(2) 是否将计算的数据定义为行块。

(3) 求和行块的首行是否是第一行要求和的数据。如果首行不在该数据行上,而是在第一行数据上面的表横线上或表横线上面的表头内,则机器不予计算。此时,应用 F8 键重新定义行块,再进行计算。

(4) 按 Ctrl+C 时,光标是否在公式右侧。

2. 机器给予计算,但结果不为所求。

出现这种情况一般是计算公式写错,应检查:

(1) 各表列定义是否正确,即是否从表格的第一表列开始依次将其定义为 C1, C2, …, Cn。

(2) 公式是否按要计算的要求给出。

检查修改无误后,再进行计算。

3. 执行计算命令后,在存放结果的表列中全部或某些表格内存放的不是数据,而是一串“*”号。

出现这种错误的原因是存放结果的表格宽度不够,放不下计算的结果。解决的方法是将光标移到出现“*”号的表列内按 F6 键,加宽该表列,然后再计算。

4. 计算时,预计存放结果的表格完全可以放下计算结果,但还是出现“*”号。此时应检查是否给出了应保留的小数位数。因为在计算时,机器将按指定的小数位数保留小数。如果公式中出现了除法运算,但没有指定计算后的小数位数,当运算结果超过表格单元宽度(如出现无限循环小数)时,放此结果的单元就会以一串“*”号代之。这时,只须在计算公式中填加要保留的小数位后再计算。

三、同一表列数据按公式计算的出错处理

1. 写出公式后,计算不出结果

出错原因主要有以下几种:

(1) 计算公式书写不符合要求。

(2) 计算公式未放在存放结果的行上。

(3) 没定义矩形块。

依次检查以上各项并修改后,再进行计算。

2. 在存放结果的行内,只有部分表格单元出现结果,而某些单元内没计算出结果。

出现这种情况一般是矩形块定义有错,定义时没将所有要计算的数据包括在矩形块中。这时,只须查看要计算的数据范围,重新定义矩形块,然后再计算。

3. 存放结果处给出一串“*”号”

出现这一问题的原因与同一行数据计算时一样,一般是由于存放结果的表格宽度不够。用 F6 键依次加宽出现“*”号的表列,再进行计算即可。此外,也可能由于算式中有除法,没有指定保留的小数位数,造成存放结果的表格宽度不够,这时只须填加要保留的小数位数,然后再次计算。

也谈对几种新病毒的正确认识与清除

王江民 山东烟台市(264001)

本刊1992年11期苏民生先生与93年8期何崇乐先生提到了对几种新病毒的认识,二位先生文中对Genp/GenB的解释并无大出入,但还易使人混淆。何崇乐先生文中提出GenF是指感染文件分配表的病毒,这一点是不正确的。上述文章都没有指出用什么方法和工具能彻底清除这些新病毒,在这里再进行一些补充。

美国MacAfee公司的SCAN与CLEAN是非常优秀的查解病毒软件之一,对目前众多的新病毒和变种病毒不断涌现,SCAN软件取其病毒广谱特征码进行查验,例如发现新病毒后显示出如下信息:

“Found the Generic Boot [Gend] Virus boot sector”

这句话的意思是“在DOS引导扇区(BOOT)发现普通的DOS引导区病毒!”。其中的“[GenB]”是为了便于配套的消毒软件CLEAN能确认后只去清除DOS引导区中的病毒,并非称“GenB病毒”。再如:“Found the Generic MBR [GenP]” Virus in partition table”

这句话的意思是“在分区表区域发现普通的主引导记录病毒!”其中的“[GenP]”是为了便于消毒软件CLEAN能确认后只去清除硬盘分区表区域中的主引导记录病毒,并非称“GenP病毒”。其中的“MBR”是英文“Master Boot Record”的缩写,即“主引导记录”。

目前,国内流行的一些新病毒如苏民生文中提到的Ctrl+Break(中断就破坏)病毒、Cmos Destroyer(CMOS设置破坏者)病毒、Disk Killer(新磁盘杀手)病毒、Mask(假面具)病毒等,这些病毒不仅感染硬盘主引导记录,同样也能感染软盘DOS引导区。SCAN软件在硬盘中查出这些病毒后,以“[GenP]”的缩写形式让操作者带进消毒软件CLEAN的参数中,也就是说告诉CLEAN软件该病毒是何种类型与潜藏的区域。另外,SCAN软件在软盘中查出这类病毒后,就用“[GenB]”的缩写形式报告CLEAN软件该病毒是何种类型与潜藏的区域。所以,同一类病毒,感染硬盘与软盘的区域不同,SCAN就有两种表示方式,目的就是让配套的消毒软件CLEAN能够正确去清除主引导区或DOS引导区中的病毒。

再如,Pretty Girl(漂亮女孩)病毒感染硬盘DOS

引导区(BOOT)或软盘DOS引导区(BOOT),所以SCAN查出后都称为普通的DOS引导区(BOOT)病毒,并以“[GenB]”缩写的方式让配套的CLEAN消毒软件能够正确识别。

以上用SCAN软件查出引导区新病毒后,可用配套的CLEAN软件来清除病毒,其格式如下:

CLEAN d:[GenB]

或 CLEAN d:[GenP]

苏民生先生提到过,有时CLEAN用“[GenB]”的参数来解除软盘中的病毒失败。的确如此,对于软盘来说,如果软盘在感染病毒前不是在IBM-DOS系统下用FORMAT命令格式化的,CLEAN消毒就失败,因此,必须用本文后面提到的工具来清除病毒。

用8.4B89版的SCAN软件扫描带有旅行者病毒的文件,出现如下信息:

“Found Generic Virus [GenF]”

而V94以后的SCAN软件扫描同样的文件,出现如下信息:

“Found the Generic Famr [FR] Virus”

这两句话的意思是“发现普通的文件型病毒”,而并非是何崇乐先生一文中指出的“感染文件分配表的病毒”。对这一类病毒,CLEAN不能从文件中清除,只能连文件一起删除。

另外,用SCAN对目前流行的汉字213H和213K系列文件中的PRTA.COM和WS.COM扫描后,也报告有“FamR [FR]”病毒。这是一种误查,我们只要查看该文件字节数分别为4240与22784左右,那么就无病毒。所以SCAN查普通文件型病毒而取得这类病毒广谱特征码,非常糟糕。

以上是用SCAN软件查病毒后,对出现一些提示的理解,便于读者以后正确使用。但该软件也有其不足,比如,它的智能广谱查病毒系统就查不出“火炬”病毒等,这一点就不如我国开发的智能广谱可扩充查解病毒软件KV20。KV20的智能广谱查病毒系统优于SCAN,能查出和清除所有引导区新病毒,并且是汉英两种文字提示。另外,它由用户自己就可很简单方便的扩充到查出千万种新病毒,并且可以查变型病毒和备份硬盘主引导扇区。如果你用CLEAN清除不了病毒,不妨可试用KV20来清除病毒。



高速 CMOS 微处理器系列

零件型号	描述
精简指令集计算机 CMOS 微处理器	
IDT79R4000/4400	高性能、高度综合 64 位中央处理器、全二进兼容容连 R3000A、结合中央处理器、浮点及 16/32KB 高速缓冲存储器，可达 50VAX mips 持续表现
IDT79R3051/52	RISC 控制器 6KB 或 10KB 片上高速缓冲存储器、R3000A 中央处理器中心、4 深度读/写缓冲器、低成本 84 引脚封装
IDT79R3081	RISC 控制器、20KB 片上高速缓冲存储器、R3000 中央处理器、R3010A 浮点加速器、4 深度读/写缓冲器、接脚兼容连 R3051/52/41

高速 CMOS 及 BiCMOS 静态随机存取存储器

零件型号	描述
256/288K 静态随机存取存储器	
IDT71256S	32Kx 8, 标准功率
IDT71256L	32Kx 8, 低功耗
IDT71259	32Kx 9 BiCMOS
3.3V 快速静态随机存取存储器	
IDT713256	32Kx 8, 3.3V
IDT713024	128Kx 8, 3.3V
特殊静态随机存取存储器	
IDT71874	8Kx 8 BiCMOS、高速缓冲存储器标识符连地址比较器异步清除
IDT71580SA	32Kx 9、突发性方式 486

先进先出存储器 (FIFOs)

零件型号	描述
同步 (时钟) 单向先进先出存储器	
IDT72420	64 x 8
IDT72200	256 x 8
IDT72210	512 x 8
IDT72220	1K x 8
IDT72230	2K x 8
IDT72240	4K x 8
IDT72421	64 x 9
IDT72201	256 x 9
IDT72211	512 x 9
IDT72221	1K x 9
IDT72231	2K x 9
IDT72241	4K x 9
IDT72841	双 4K x 9 (可配置)
IDT72205LB	256 x 18 (深度可扩展)
IDT72215LB	512 x 18 (深度可扩展)
IDT72225LB	1K x 18 (深度可扩展)
IDT72235LB	2K x 18 (深度可扩展)
IDT72245LB	4K x 18 (深度可扩展)

高速 CMOS 逻辑产品

3.3V 双密度系列	
IDT54/74FCT163244/A	3.3V CMOS 16 位缓冲器/线驱动器
IDT54/74FCT163245/A	3.3V CMOS 16 位双向收发器
IDT54/74FCT164245T	CMOS 16 位双向 3.3V 至 5V 转换器
IDT54/74FCT163373/A	3.3V CMOS 16 位双向锁存器
IDT54/74FCT163374/A	3.3V CMOS 16 位寄存器
IDT54/74FCT163646/A	3.3V CMOS 16 位总线收发器/寄存器
IDT54/74FCT163501/A	3.3V 18 位寄存器总线收发器

高速 CMOS/BiCMOS 多端口随机存取存储器

零件型号	描述
双端口随机存取存储器	
IDT7130	8K (1K x 8) 主控：业内最常用的双端口 11SRAM
IDT7140	8K (1K x 8) 从属具 IDT7130 功能以提供 16 位或更宽的字：接脚与 IDT7130 兼容
IDT7132	16K (2K x 8) 主控具业内最快速的标准产品、多源
IDT7142	16K (2K x 8) 从属具 IDT7132 功能以提供 16 位或更宽的字：接脚与 IDT7132 兼容
IDT71321	16K (2K x 8) 主控具中断输出的高速双端口
IDT71421	16K (2K x 8) 从属具 IDT71321 功能以提供 16 位或更宽的字：接脚与 IDT71321 兼容
IDT70121	18K (2K x 9) 主控高速双端口连忙碌及中断
IDT70125	18K (2K x 9) 从属具 IDT70121 功能以提供 18 位或更宽的字
IDT7133	32K (2K x 16) 主控高速双端口连忙碌
IDT7143	32K (2K x 16) 从属具 IDT7133 功能以提供 32 位或更宽的字
IDT7134	32K (4K x 8) 在无需件单片仲裁的系统中高速操作
IDT7005	64K (8K x 8) 连忙碌、中断、信号标及主控/从属选择
IDT7025	128K (8K x 16) 业内最大的单片双端口 RAM 连忙碌、中断、信号标及主控/从属选择
IDT7006	128K (16K x 8) 连总线、中断、信号标及主控/从属选择

* 索要详细资料请与晓泥公司联系：北京市白石桥路 3 号友谊宾馆 20469 号 电话：8499430 8498888—20469、邮编 100873

用 F BASIC 编制的 6502 汇编与反汇编程序

郭 杰 唐山市开滦砚土矿

我们知道,与游戏机配套的电脑键盘,其学习卡中的系统程序是用 6502 机器语言编写的,它为用户提供了一种 F BASIC 语言。但是由于说明书的简陋和技术资料的缺乏,以及厂家没有提供汇编与反汇编功能,给用户带来诸多不便。

为了能够在廉价的电脑游戏机上学习和运行 6502 汇编语言程序,剖析和利用学习卡中的系统程序,让电脑游戏机在家庭和学校中发挥更大的作用,笔者用 F BASIC 语言编制了这一集多种功能于一体的实用程序,扩大了电脑游戏机的应用范围,为学习和应用 6502 汇编语言以及了解和掌握学习卡的系统程序提供了一个有力的工具。

一、主要功能与运行环境

1. 主要功能

(1) 可以输入 6502 指令系统规定的汇编语言程序。本程序将其直接翻译为机器语言程序存储在内存中,并可对其进行修改和运行。同时,还提供了一种录入磁带的简易方法。

(2) 可以使用符号地址。按本程序规定使用的符号地址可以用在标号字段和所有三字节指令的操作数字段中,还可用等值命令为其赋值。在条件转移指令中,既可用符号地址,又可用转移到目标处的绝对地址,还可用转移步长。

(3) 可以对内存单元直接写入机器指令码或 16 进制、10 进制或字符型数据。

(4) 可以对 64K 内存单元内容进行反汇编。当遇到数据区或不符合 6502 指令系统规定的操作码,将自动显示 16 进制数据和对应的 ASCII 码字符。

(5) 本程序设置了八条命令,可实现对用户程序进行修改、移动、增删、检查和运行等多种实用功能。

(6) 程序中做了大量检错处理,容错能力较强。

2. 运行环境

能否在自己的电脑游戏机上运行本程序,关键是两点:一、至少要有 8K 内存容量,二、要有本程序中用到的 BASIC 语句。

本程序是在天马公司的 TM5510 键盘上调试成功的,其 F BASIC 语言学习卡内含有两个版本,一个是由人机对话进入 F BASIC,版本号是 BS. 2A,只有 4030 字节,不能运行本程序(如果舍去汇编程序段,可以进行反汇编)。另一个是直接进入 F BASIC,版本号是 V3. 10,有 8182 字节,本程序正是在这个版本中调试成功的,运行后,占内存近 5.5K,另留有 2.5K 内存供用户输入汇编语言程序。这个版本的特点是:1. 可用 F1 至 F4 功能键调出四个游戏程序运行;2. 用户内存地址范围在 \$ 6000 ~ \$ 7FFF。凡具备上述两个特点或版本相同,均可不加修改的正常输入运行本程序。如果用户内存不在 \$ 6000 ~ \$ 7FFF,可对有关语句稍做修改即可。

在输入本程序时请注意一点,就是在 GOTO、GOSUB 语句与其后的行号之间不要留空格,以防运行后的程序超过 5.5K,从而会影响汇编语言程序的输入。

二、本程序的使用方法

本程序规定:用来存放用户目标程序的范围设定在

\$ 7600 ~ \$ 7FFF。程序运行后,屏上显示:

7600: ■

\$ 7600 是输入汇编语言源程序的首地址,其后的提示符——冒号和一个闪烁的光标,是表示等待键盘输入。在这种等待输入状态下,既可以输入汇编语言程序,也可以输入本程序设置的各种命令。

如果输入的是汇编语言程序,则被翻译为机器指令,按屏上自动显示的内存地址,顺序存在相应的内存单元中。同时,根据指令字节的长短,自动在屏上显示下一个内存地址,逐条输入即可。

如果输入的是各种命令,除了置数命令外,都与自动显示的内存地址无关,也就是说,在自动显示的任意单元地址的后面输入这些命令,都会被正确执行。

现将程序和各种命令的输入格式及方法分述如下:

1. 汇编语言源程序的输入格式及方法

本程序规定:汇编语言程序的语句最多可由标号、操作码和操作数三个字段组成,各字段之间用一个空格分隔。具体规定如下:

(1) 标号是用来定义符号地址的,它将自动显示的 16 进制内存地址赋值给符号地址。本程序规定:符号地址由三位字符组成,必须以“&”打头,后两位是任意的 16 进制数,如 &05、&AB、&F0 等均可。标号要在冒号后面紧接着打入。如果某行语句不使用标号,而直接输入操作码或各种命令时(等值命令除外),屏上会自动留出四个空格。

(2) 操作码字段按 6502 指令系统的三个字母助记符输入。本程序未设伪指令,一些功能如 EQU 和 DFB 可用等值命令和置数命令实现。

(3) 操作数字段可以使用符号地址,要用在三字节指令和条件转移指令中。如:“LDA &A2”;“STA &BA, X”;“JMP (&3F)”;“BNE &06”。在条件转移指令中,除了可用符号地址外,还可用绝对地址和转移步长。其它二字节指令不能用符号地址。

除了上述规定外,均可按 6502 的寻址方式输入,但是,16 进制表示符“\$”省略不用。凡是两字节指令的操作数,一律用两位 16 进制数,三字节指令的操作数,一律用四位 16 进制数,位数不足时,要在前面补零。

(4) 与符号地址有关的数据是存放在二维数组中的。由于二维数组的限制,在一个汇编语言程序中,最多可以使用 31 个不同的符号地址。它的计算方法是:如果在标号字段定义了一个符号地址,或定义之前在操作数字段只使用了一次,那么,在其后不管使用了多少次这个相同的符号,均按一个符号地址计算。如果在标号字段对一个符号地址定义之前,操作数字段多次使用了相同的符号,就要按使用的次数计算。

为便于随时了解数组的使用情况,本程序设置了一个检测操作,在等待输入状态时直接回车,屏上将显示“&XX=n”,n 是符号地址数。当 n 值已为 31 时,再使用新的符号地址将使数组溢出,使本程序中断运行。当重新运行本程序时,虽然已

输入到内存中的机器语言程序不会被清除,但数组中有关符号地址的数据被清除,在有些时候会带来不便。

为保留数组中的数据要避免使本程序中断。然而在输入一个新的汇编程序之前,可利用中断和重新运行来删除数组中原有数据,避免前后两个毫无关联的汇编程序使用相同符号地址而造成莫名其妙的错误。

(5)本程序用 CALL 将用户程序做为一个子程序来调用运行。用户程序中的最后一条指令要用返回指令 RTS,以便能顺利返回本程序的等待输入状态。最后一条指令也可用 BRK,但运行完后将退出本程序,而回到系统热启动状态,但本程序和用户程序均不会被清除。如不用上述指令,可能出现意想不到的后果。

(6)在输入完一行程序或命令而没有回车时,发现有误,可用各种编辑键进行修改。如果回车后,发现有误,或者需要增删、修改等,可选用后面将要介绍的有关命令进行操作。

当输入的符号指令与 6502 指令系统不符,或者输入格式不符合本程序规定时,回车后,将鸣一声喇叭,屏上光标退回到冒号后面的初始位置闪烁,同时在本行标号字段的位置显示“ERR”,提示输入有误。

2. 各种命令的输入格式及方法

本程序共设置八条命令,可以在等待输入状态下随时打入。当打入第一个字符时,屏上会自动空出四个空格(等值命令除外),然后才显示刚输入的字符。

在下面介绍的命令格式中,除去检查命令和反汇编命令外,内存地址一律用四位 16 进制数表示。

(1)置数命令

格式: {数值} {标点} [{数值} {标点}] ...

此命令可以将 16 进制数、10 进制数或字符的 ASCII 码直接写入内存,用于初始置数、输入机器指令和修改内存单元,常与变址命令配合使用。数值的类型用其后面的标点表示。16 进制数后面的标点使用逗号;10 进制数后面用分号;字符后面用单引号。

例1

7700: 41,66,C';从 \$ 7700 单元起置数

7703: ;等待键盘输入

在例1中,7700 是 16 进制地址,冒号后面的四个空格,是在输入第一个数字 4 的时候自动留出来的。回车后,这三个数据被顺序写入起始地址为 \$ 7700 ~ \$ 7702 的三个单元中,并自动显示下一单元地址等待输入。在这里,41 是 16 进制数,66 是 10 进制数,C 被写入内存时是它的 ASCII 码。

每行数据不得超过 31 个字符,包括自显部分及空格。一行写不下,回车后可随自显地址继续输入。另外,在标号字段——数据前面的空格处,可以使用符号地址。

(2)变址命令

格式: {内存地址}

此命令可从改变自动显示的内存地址。为提高屏幕利用率,只改变本行的自显地址,随着后面程序或命令的输入,本命令显示的内容将被覆盖。如将例1中的三个数据存入 \$ 7700 ~ 7702 单元,而初始自显地址为 \$ 7600,即可用本命令变址,操作及屏显如下:

7600: 7700;回车前的屏显

回车后,仍在本行显示如下:

7700: OK-7700;这也是一种等待输入状态

OK 表示本命令已执行完毕。在这种等待输入状态,就可以在本行随着光标输入预定的三个数据。随着数据的输入,光标后面的字符被覆盖删除,如下:

7700: 41,66,C';回车前的屏显

以上三行实际上是在屏上一行显示完成的。后面的示例将把具有这种显示特点的命令用括号括起来,括号中的内容,一般仅取回车前的显示状态。如上述显示及完整的操作过程用下面形式表达:

例2

(7600: 7700);变自显地址为 \$ 7700

7700: 41,66,C';将三个数据写入 \$ 7700 ~ \$ 7702 中

(7703: 7600);变址为 \$ 7600

7600: OK-7600;等待输入

上述书面表达,在屏上实际只占两行,有括号的内容被后面的输入覆盖。随着光标从首址 \$ 7600 开始输入汇编程序,光标后面的字符也将被覆盖。

具有这种显示特点的还有运行命令、删除命令和移动命令,均以例2中带括号的形式进行书面表达。

(3)等值命令

格式: {符号地址} = {内存地址}

此命令的功能是将指定的内存地址赋给符号地址。一般情况要写在程序的开始处。

例3

7600: &F1=7606; &F1 被赋以 \$ 7606

7600: LDA #0F;取 \$ F 至 A

7602: STA &F1;送至 \$ 7606 单元

7605: RTS;结束

7606: ;等待输入

(4)运行命令

格式: G

执行此命令,将运行内存中首址为 \$ 7600 始的机器语言程序,运行完毕后,在屏上本命令行显示 OK,屏显特点与变址命令相同。

(5)检查命令

格式: [内存地址] M

检查内存单元内容。如果省略内存地址,只打入 M,则固定从 \$ 7600 单元开始显示各单元内容。

输入命令回车后,屏上首先显示命令中指定的或隐含指定的内存地址及“)”符,随后顺序显示一行八个单元的 16 进制数据,并且在数据下面对应显示 ASCII 码字符,没有对应的字符则填以空格,八个单元数据显示完后将自动暂停显示。如果想继续查看后面内容,请按任意键(回车键除外),则按上述形式连续显示,直到显满一屏才会再次自动暂停。

在显示过程中,如果按空格键,可随时使显示暂停。如果按回车键,则鸣一声喇叭,以示中断,随即返回到等待键盘输入状态。下面将例3示范程序重新输入,并运行和检查如下:

例4

7600: &F1=7606

7600: LDA #0F

7602: STA-&F1

7605: RTS

(7606: G);运行已被翻译成机器语言的目标程序

7606: M;检查从 \$ 7600 始的内存单元内容

7600> A9 0F 8D 06 76 60 0F;回车后返回

7606: ;等待输入

可以看到 \$ F 已被存在 \$ 7606 单元中。利用检查命令可同时显示 ASCII 字符的功能来检查系统数据区,可发现 F BASIC 的所有保留字和与其对应的机内代码:

例5

```
7600:┌┐┐┐CCABM┐;查看从$CCAB始的内容
CCAB>80 47 4F 54 4F 81 47 4F
      G O T O G O
CCB3>53 55 42 82 52 55 4E 83
      S U B R U N
CCBB>.....
```

由于版本或生产厂家不同,起始地址可能有出入。以上显示的\$80是GOTO的代码,\$81是GOSUB的代码,余类推。了解这些机内码有助于分析F BASIC解释程序。

(6)反汇编命令

格式1:[内存地址]L
格式2:[内存地址]K

两格式的差别仅仅是屏幕显示的格式不同。L为紧凑输出格式的显示,K为空行显示,以便于在屏上直接研读或抄写。如果省略内存地址,则固定从\$7600单元开始反汇编。在显示中如果遇到RTS、JMP、RTI指令,将做额外的空行处理,以使屏显的程序段落清晰易辨。

在反汇编过程中,暂停和返回等待输入状态的各种键盘操作与检查命令相同。如果遇到内存中的数据区或没有对应的符号指令,将鸣喇叭做提示,然后自动转换为检查命令的输出格式,显示其后的数据。

反汇编在屏上显示的格式为:中间部分是内存地址,左边为机器指令码,右边是翻译后的汇编语言程序。如将例3程序进行反汇编,操作及显示如下:

例6

```
7606:┌┐┐┐L┐;从$7600始以紧凑格式显示反汇编内容
A9 0F <7600>LDA #0F
8D 06 76 <7602>STA 7600
60 <7605>RTS┐;回车返回
```

7606:■;等待输入

再举一个查找系统程序的入口地址并进行反汇编的例子。机器上电后,6502首先从指定的\$FFFC(低位)和\$FFFD(高位)取出入口地址,然后就从这个地址开始运行系统程序。操作及显示如下:

例7

```
7600:┌┐┐┐FFFCM┐;检查入口地址
FFFC>ED 00.....┐;入口地址为$00ED
7600: EDL┐;从$ED开始反汇编
4C BA 80 <00ED> JMP 80BA┐;中断反汇编
7600: 80BAK;用空行格式反汇编
A9 00 <80BA> LDA #00
8D 00 20 <80BC> STA 2000
8D 01 20 <80BF> STA 2001
20 2B B4 <80C2> JSR B42B
      :
```

以上是V3.10版系统程序初始化开头的几条指令。

(7)删除命令

格式:[首地址]-{尾地址}

此命令可以删除指定的内存地址范围中的内容,而全部写入\$FF。如果省略首地址,则首地址被内定为\$7600。屏显特点与变址命令相同。

例8

```
(7600:┌┐┐┐7610-761F┐);删除这16个单元内容,
写入$FF
```

7600:■OK,7610-761F;等待输入

使用此命令要慎重,不要将\$0000~\$7500内存单元的内容随意清除掉,以免造成意想不到的后果。

(8)移动命令

格式1:{新首址}<{现首址},{现尾址}

格式2:{新首址}<{现首址}-{现尾址}

两种格式命令的作用,均是將一段内存程序或数据按顺序移到新的内存位置。其功能区别是,格式2不仅具备格式1的移动功能,而且在移动后,还可删除那些已被移动到新位置而无用的目标程序。

例9

```
7600:┌┐┐┐LDA┐#00┐
7602: STA 7620┐
7605: STA F0┐
7607: RTS┐
7608:■
```

现在想把7602地址行的指令删除,并使后面的程序往前移。先用格式1移动命令操作,然后用反汇编命令检查,续写如下:

例10

```
(7608:┌┐┐┐7602<7605.7607┐)
7608: L┐
A9 00 <7600> LDA #00
85 F0 <7602> STA F0
60 <7604> RTS
85 F0 <7605> STA F0
60 <7607> RTS
```

可见,预定删除的一句已被向前移动的一段程序覆盖,可是\$7605~\$7607单元的无用程序仍在内存中。如果用格式2的移动命令,就可以同时将这段无用程序删除,写入\$FF。限于篇幅,此练习留给读者。

上面提到的是将程序向前移动时,使用格式2命令会把后面的无用程序删除,如果将程序向后移动,格式2命令又会把前面的无用程序删除。

仍以例9为例。现在想在\$7602和\$7605地址行之间插入一条二字节指令“LDA #04”,因此,首先要将\$7605~\$7607程序段向后移动两个字节,然后再从\$7605单元开始插入这条指令。接例9操作续写如下:

例11

```
(7608:┌┐┐┐7607<7605-7607┐);使用格式2命令
7608: L┐
A9 00 <7600> LDA #00
8D 20 76 <7602> STA 7620;后面自动显示数据
7605>FF FF 85 F0 60.....┐;无用程序被删除,空出两字节
```

(7608: 7605┐);变址命令

7605:┌┐┐┐LDA┐#04┐;插入预定的指令

7607: L┐;再次用反汇编检查

A9 00 <7600> LDA #00

8D 20 76 <7602> STA 7620

A9 04 <7605> LDA #04;此行指令已插进程序中

85 F0 <7607> STA F0

60 <7609> RTS

使用格式2命令后,在反汇编中可清晰地识别与核对要插入的字节数。如果用格式1命令,原7605行的指令仍保留在要插入的地址单元中,虽不影响后面的插入操作,但给识别

造成一定困难。

由于格式2的移动命令兼有删除功能,因此使用时要慎重。另外,对使用符号地址的汇编程序进行移动修改时,要注意对可能出现的跳转错位进行及时修改。修改程序的方法可灵活掌握。如例11的插入,也可用置数命令直接将机器指令码写入内存。

下面再介绍一个将各种命令综合运用的示例:

例12:

有一列不带符号的二进制数(用16进制表示):05,92,AB,0,EA,7F。其中5是数据的个数,要求找出后面5个数中的最大值。

我们将这列数据存在\$7616~\$761B六个单元中,最大值在\$7615单元中。通过编程、运行和用反汇编检查,存在\$7615单元中的值应为\$EA。操作及屏显如下:

(7600:↙);对数组进行检测

(7600:■OK &XX=0);表明二维数组已清零

(7600:▯▯▯▯-7620↙);用删除命令清内存

(7600: 7616↙);对自显地址进行变址

7616:&DA 5,92,AB,0,EA,7F,↙;置数命令

(761C: 7600↙);变址命令

7600:&FF=7615↙;等值命令

7600:▯▯▯▯LDX &DA;计数器置初值为数据个数

7603: LDA #00↙;设最大值为零:A=0

7605:&A1 CMP &DA,X↙;内存中数据与A比较

7608: BCS &A2↙;不大于A,A中最大值不变

760A: LDA &DA,X↙;大于A,用该数替换最大值

760D:&A2 DEX↙;计数器减一

760E: BNE &A1↙;继续循环比较至完毕

7610: STA &FF↙;存最大值

7613:▯▯▯▯RTS↙;结束

(7614:G↙);运行

7614:L↙;反汇编

AE 16 76 <7600> LDX 7616

A9 00 <7603> LDA #00

DD 16 76 <7605> CMP 7616,X

B0 03 <7608> BCS 760D

BD 16 76 <760A> LDA 7616,X

CA <760D> DEX

D0 F5 <760E> BNE 7605

8D 15 76 <7610> STA 7615

60 <7613> RTS

7614>FF EA 05 92 AB 00 EA 7F

可见,\$EA已存在\$7615单元中。在实际编程和操作中,有些步骤可简化或省略。(等续)

DOS 操作系统分析讲座

第四讲 汉字信息处理

崔来堂 石家庄铁道学院计算机教研室(050043)

工作和生活中的数据分为两大类:数值型和非数值型。可以实施四则运算的数据称为数值型数据,否则称为非数值型数据。在多数领域中,非数值型数据的量通常远大于数值型数据的量。计算机由主要用于数值计算,发展到以非数值型数据处理为主,大大拓宽了应用范围,是计算机开发应用的一个重大突破。随着微机的迅速普及,这方面的应用将越来越占更大的比例。

我国的通用文字是汉字,因此,汉字信息处理是非数值数据处理的主要成份和关键问题。自83年至今的十年中,汉字处理技术从无到有,并得以极迅速的发展,目前已达到很高水平。

一、汉字代码

计算机对信息的处理,不是直接处理字符,而是处理字符的代码。代码是按照某种规则对字符的编码。通常说的代码,指的是字符在计算机内部进行存储和处理时所用的编码,即机内码(简称内码)。微机系统中,采用美国信息交换标准码——ASCII码作为字符内码,每个字符的ASCII码占用存储器的一个字节,这些字符包括大小写英文字母、阿拉伯数码、某些控制符,以及其它有关符号等。

1. 汉字代码的种类和标准

汉字的输入、显示和打印,是汉字信息处理的主要功能,具有这种功能的系统,称为汉字信息处理系统(简称汉字系统)。汉字系统可以对汉字代码进行处理。

为了实现汉字处理,汉字系统引入了多种代码。这些代码

包括:由输入设备(通常是键盘)产生的汉字输入码(亦称外码);用于计算机内部对汉字进行存储和处理的汉字机内码(即内码);用于汉字显示的汉字显示字模点阵码;用于汉字打印的汉字打印字模点阵码;用于与网络或其它系统进行通信的汉字交换码;以及用于在汉字库中查找汉字字模的汉字地址码等。

由于上述汉字代码直接影响着汉字处理系统软硬件的研制,需要使之标准化。我国于1981年颁布了GB2312-80《信息交换用汉字编码字符集——基本集》国家标准,这是汉字信息处理系统的基本标准。目前,汉字字模库的设计,汉字输入码与内码的转换等,都广泛采用了这个标准。

上述汉字诸码中,与用户和系统结构关系较大的是输入码和内码。输入码将在后面讲述,下面先介绍内码。

2. 汉字内码

汉字内码是汉字在机器内部进行存储和处理时采用的代码。汉字输入时,通过计算或查询输入字典,将输入码转换成内码;输出时,根据内码在汉字库中查找相应汉字的点阵字模,送去显示或打印。

汉字内码是汉字系统结构设计的基础,因此,应尽量简短,以节省存储空间,并便于处理。但因汉字字型点阵较复杂,字模库也比西文系统大得多,用一个字节的内码已无法进行描述。国标GB2312-80规定,一个汉字内码用两个字节的机内码符来表示。由于ASCII字符机内码(一个字节)的最高位为0,为了不产生二义性,将两个字节汉字机内码符的国标码

之最高位都置1,这样形成的机内码称变形国标码。CC-DOS即采用变形国标码作为它的汉字内码。

用户获取汉字内码的简单方法是:在某汉字系统中,用编辑程序将待查汉字或汉字序列录入一文件中(设文件名为AA),然后再用汉化 DEBUG 或汉化 PC Tools 加载该文件,进行显示观测。具体操作方法如下:

C>DEBUG AA

-D100

或者利用 PC Tools 文件功能中的编辑 E 命令。两种方法中,都是在屏幕的右边显示待查的汉字或汉字序列,左边相应位置给出它们的内码。

二、汉字库

(一)汉字库及其分类

建立一个汉字信息处理系统的关键,是解决汉字的输入、输出和建立汉字库。

汉字以输入码输入计算机,在计算机内部以内码形式存储,而在显示和打印输出时,以汉字字型信息表示。因此,汉字系统必须建立一个用于产生或直接存储汉字字型及必要的图形符号信息的存储系统,称为汉字字模库或汉字字型发生器(简称汉字库)。汉字库由一批汉字字型码或字型压缩码,以及其它有关图形符号,按照一定顺序排列而成,它为输出设备(如显示器或打印机)提供汉字的字型信息。随着汉字系统的广泛应用,字库已成为保证汉字输出,特别是打印输出质量的关键因素。目前,字型信息一般采用点阵方式存储。因汉字笔划较复杂,故需较高的点阵来表示(至少16*16点阵)。对于字型要求更高的环境,特别是印刷系统,必须配备多种字体的汉字库,字数也更多,有的还既需要简体汉字,又需要繁体汉字。

字库的大小不等,所包含的汉字字数可以根据不同的要求来确定,但通常均应包含国标 GB2312-80中规定的一级和二级汉字,以及非汉字图形字符。一级汉字是常用字,一般都知道其读音,故按汉语拼音字母顺序排列;多音字取其常用发音;同音调的字则以起笔的笔形——横、竖、撇、点、折排列;起笔相同者则按第二笔;依次类推。二级汉字大部分较生僻,不容易掌握读音,故按部首排列;部首的顺序按笔划数排列;变形部首排在正常部首之后;同部首字按除去部首之外的笔划数排列;同笔划数的字也以起笔的笔形顺序排列。

汉字国标码基本集 GB2312-80的结构如图4-1所示。

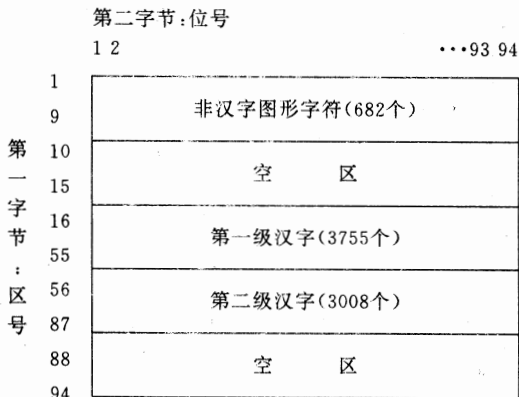


图4-1 汉字国标码基本集 GB2312-80 结构图

为了遵循标准化原则,基本汉字字模库(16*16点阵库,即汉字显示字库)的结构应与上图相似;但因汉字库常驻内存,为了尽量节省内存资源,故略去了上图中的空区以及6、8区(不常用字母和符号区);并将上图中的第9区作为汉字库的

第6区。汉字库与 GB2312-80 国标码基本集的对应关系和区别如图4-2。

汉字库的区号	GB2312-80的区号	内 容
1-5	1-5	符号
6	9	表格符
7	7	大、小写俄文字母
8-79	16-94	一、二级汉字
无	6	大、小写希腊字母
无	8	拼音符号和字母
无	10-15	空区
无	88-94	空区

图4-2 16*16点阵汉字库与 GB2312-80的结构对照表

该汉字库共支持 GB2312-80中的571个图形符号和全部6763个汉字,约占256KB。根据信息存储方式的不同,汉字库分为不压缩型和压缩型。不压缩型库中存储汉字的全点阵信息,故亦称全点阵型。其优点是字型质量高和读取速度快;缺点是系统开销大,以32*32字型点阵为例,一个汉字就要占128个字节,包含国际一、二级共6763个汉字和一些图形符号的汉字库,其容量约896KB。压缩型字库中存放的是字型压缩码,以节省存储空间,但在读取时需要对字型进行还原,速度慢,还原时也容易失真。

根据字型质量,汉字库又可分为普通型和精密型。普通型字库存放的是低点阵汉字字型码,字的质量比较低,如16*16点阵字型码。精密型字库采用高点阵的汉字字型码,可反映出不同的字体、字号及笔锋,能满足书报印刷系统对文字质量的要求,可用于印刷输出。

(二)点阵字库与轮廓字库

就字型信息的构成而言,字库又分为点阵字库和轮廓字库两种类型。

1. 点阵字库

当前,大多数汉字库中的字型信息是以点阵形式来表示的。

这里所谓点阵,就是把一个方块汉字分解为若干小方格,每个小方格的中心点为一个点,有笔划通过的小方格视为“黑”点,无笔划通过的视为“白”点,这样,所有黑点就描绘出相应的汉字字型,即点阵字型(简称点阵)。点阵中的每一个点,用二进制的一位表示,“黑”为1,“白”为0,一个字节存储单元可表示点阵中的8个点,整个点阵就可以由二进制数据形式表示出来。

所谓点阵字库,是由排列成行列的点阵,组成的汉字和字符字型的有序集合。汉字字型点阵通常是16*16,24*24,32*32,40*40等8的倍数,点阵排列可以是横向字型信息,也可以是纵向字型信息。由于显示器是按横向扫描的,供显示用的汉字字型点阵应是横向型的;而打印机的打印针是纵向排列的,因此,送到打印机的必须是纵向点阵字型信息。汉字的这种存储方法亦称整字存储法。打印时,只要把汉字的点阵数据传送给打印机即可。为了加快打印速度,某些打印机本身固化有这种点阵字库(如 LQ1600K 等),这样,只需向打印机传送两个字节的汉字内码,就可打印出相应的汉字。

由于汉字字型复杂,有多种书写体和印刷体,如黑体、楷

体、宋体、仿宋体等,因此,需用不同规格的字型点阵。点阵的大小直接影响了汉字字型的质量,点阵越大,字型质量越高,但占用的存储空间也越大。故应根据不同场合,酌情选用。图4-3给出了有关数据。

点阵字型	每个汉字 占用字节	字 库 字数举例	字库容量 (字节数)
16 * 16	32	4000(7000)	125K(219K)
24 * 24	72	4000(7000)	281K(492K)
32 * 32	128	4000(7000)	500K(875K)
40 * 40	200	4000(7000)	781K(1.37M)
48 * 48	288	4000(7000)	1.13M(1.97M)
64 * 64	512	4000(7000)	2.00M(3.50M)
128 * 128	2048	4000(7000)	8.00M(14.00M)

图4-3 字型点阵及有关数据

16 * 16点阵是一种简易型汉字字型点阵,字型不够美观,仅供显示用,由它组成的字库称为显示字库,其点阵是按横向方式设计的。16 * 16点阵汉字库的字模结构如图4-4

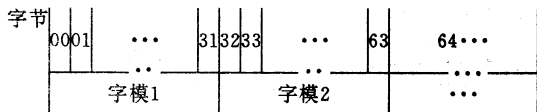


图4-4 16 * 16点阵汉字库的字模结构

24 * 24点阵的字型比较美观,主要用于打印,可以打印出笔锋。它需要的存储容量较大,因此,只当在显示器分辨率比较高的考究场合,才考虑选用24 * 24点阵进行显示。由于宋体字型标准,结构端正,在书籍、报刊上应用广泛,故24 * 24点阵多为宋体。

32 * 32点阵字型可以更好地体现字体风格,并在一定程度上表现笔锋,但其字库容量显著增大。

40 * 40及其以上点阵字型,称高级点阵或精密点阵,它们可以更好地描绘黑体、楷体、宋体、仿宋体等多种风格的汉字,适用于微机 and 页式激光打印机组成的激光照排系统。这种精密型的汉字点阵能满足书刊、报纸出版,以及某些有特殊要求文件印制的要求。

点阵字库比较简单,使用方便,运行速度也较快,在通常应用中也可以保证字型质量。但它的汉字笔划容易产生锯齿、比例失调等现象,特别当点阵数较少和打印较大字型时,汉字打印质量很难保证。常用的2.13低版本软汉字系统,以及王码480,方正 Super I、II、III(原PUC卡)汉卡等使用的都是点阵字库。

2. 轮廓字库

所谓轮廓字库,是运用数学的方法,将每个字符的外形轮廓通过数学公式,用直线或曲线描述。而形成的字型信息的有序集合。轮廓字库中存储的是汉字字型的压缩信息,因此,亦称为汉字的压缩存储法。打印时,必须通过特定的运算,将压

缩的轮廓信息转化为打印机可以识别的点阵信息。这些运算大多是在主机上进行的,因此打印速度与主机的主频有关,有些打印机内部配有微处理器,可完成这种转换工作,使打印速度大大提高,但这种打印机的价格也较高,通常只有激光打印机有此功能。

轮廓字库又有矢量字库和 PostScript 字库之分。矢量字型技术是北大王选教授发明的,它把汉字看成由许多直线段组成,撇、捺则由小段直线逼近而成,这样将汉字描述成封闭的曲线,而只存储各直线段的端点座标矢量,因此,汉字的轮廓可以用矢量来表示。这是一种比较简单的轮廓汉字,但它的字型效果比点阵字库明显改观,只当打印特大字时才有折线感,能满足通常的排版要求。目前采用这种字库的有:华光系统、213K 汉卡、M-6403汉卡和联想集团的矢量字库等。

PostScript 字库是美国 Adobe 系统公司推出,经多次改进而形成的一种电子照排印刷工业标准。它采用三次曲线拟合汉字轮廓,可以很好地保留汉字原有的字型,因此,无论打印多大的字都不会变形。PostScript 字库有三种类型:Type1、Type3和 Type0。Type1用来描述西文字体,Type3是用户自定义类型,Type0是一种复合字库。中文字库 PostScript Level I 解释器利用 Type0字库来支持汉字。目前国内产品只有北大方正字库、方正 Super V、VI型汉卡采用这种汉字库。

加拿大顶尖科技有限公司研制出以单字节为机内码的中文 PostScript 字库,把对汉字字库的处理方法,变成与处理英文字库完全一样,使 Windows 下的英文排版软件及其它西文软件不需任何改造,即可直接处理中文。这是汉字库技术的重要突破,使中西文软件完全兼容,并使汉字库的操作效率明显提高。

字库的种类直接影响着输出汉字的质量,因此,应注意各种系统所配字库的类型,进行认真选择。从汉字库的未来趋势看,轮廓字库将是今后发展的方向,特别是 PostScript 字库。

按照存储介质的不同,汉字库又分为软字库和硬字库。因这种分类方法与用户对汉字系统的使用直接有关,故较详细介绍如下。

(三)软字库

软字库亦称 RAM 字库,它是将字库的全点阵字型码存储在软盘或硬盘上,系统启动时调入内存 RAM 中并驻留。由于字库容量大,调入时花费时间较多,占用内存空间也较大,但读取时速度快。

软字库又分为分级字库和磁盘字库。

1. 分级字库

分级字库是把常用字库(如国标一级字库)在系统启动时装入内存,而使非常用字库(如国标二级字库)常驻外存(磁盘),以解决速度和内存容量的矛盾。

据统计,一般的书刊文章中,国标一级汉字(3755个)出现的概率在99%以上,二级汉字(3008个)出现的概率则不到1%。因此,利用分级方式,使一级字库驻留内存,二级字库常驻硬盘,通常情况下,汉字输出速度与字库全部驻留内存时基本一样;只当连续出现多个二级汉字时,才略有影响,但这种概率很小。故整个系统的汉字输出速度令人满意,而占用的内存空间,仅约为全部字库驻留内存时的一半。

采用分级字库,在读取字模之前,需先把硬盘上的二级字库文件打开,定义其属性为只读,并定义二级字库文件的记录长度,判断该字模是在内存还是在硬盘?然后进行相应的读操作。以16 * 16点阵字库为例,其记录长度应定义为32个字节,一个字模点阵的存储长度),它的读字模的操作流程如图4-5所示。

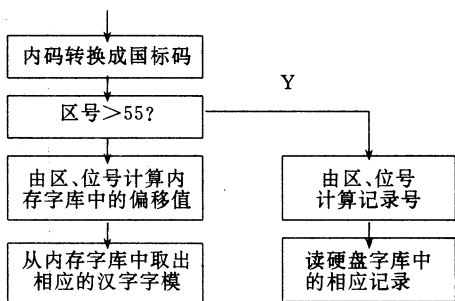


图4-5 在16×16点阵分级字库中读字模的流程图

2. 磁盘字库

高点阵汉字库由于信息量过大,常全部驻留在磁盘(通常是硬盘)上,形成磁盘字库。如CC-DOS中用于打印的24×24纵向点阵字库,以及编辑排版、打印系统中,各种字体的32×32点阵以上的字库。磁盘式字库的优点是价廉,不占用内存空间,但速度慢。为提高读取速度,字库的结构应便于查找,读盘次数要少。为此,磁盘字库中的字型,通常都按国标码的顺序连续存储;又为了使一次读盘就能得到一个汉字字型码,将字库文件的记录长度,定义为该字库中一个汉字点阵字型码的字节长度(如24×24点阵,其记录长度定义为72个字节)。这样,每个记录的内容即为一个汉字的字型码(即点阵信息),各条记录又按国标码顺序排列。当一个汉字输入时,输入码经内码、国标码,很容易转换成字型码在汉字库中对应的记录号,按该记录号读出,就得到了输入汉字的字型码。

软字库的成本低廉,广泛应用于微机汉字系统中,但因关机后RAM中的信息将消失,因此,每次开机都需从盘上向内存装载。

(四) 硬字库

由于软字库的操作速度慢,且大量占用RAM空间,因此,对要求较高的系统,有配制硬字库的必要。

硬字库亦称固化字库,是将汉字全点阵信息存入ROM或EPROM芯片中,读取速度快,但价格较贵。随着大规模集成电路价格的降低,这种字库会越来越普遍地被采用。

硬字库通称汉卡,它是汉字系统中重要的一类,准确地说应该叫做卡式汉字系统,或硬汉字系统。

1. 汉卡实现机制的分类

从显示机制和原理来看,汉卡又分为硬汉卡和软汉卡两大类。

(1) 硬汉卡

硬汉卡是用“纯”硬件实现汉字显示的部件。在西文显示逻辑电路中,增加一个双字节字符集DBCS(Double Byte Character Set)控制器,并在原ASCII字符发生器基础上,增加一个汉字字符发生器,便构成既可显示西文,又可显示汉字的硬汉卡。DBCS控制器从代码流中分离出汉字代码,经处理后送往汉字字符发生器;而ASCII字符则被送往ASCII字符发生器。二者的字模点阵均进入视频控制电路,送显示器显示。

硬汉卡可以与西文显示卡并用,当工作在文本态时,将信息切换到汉卡上,当处于图形态时,又切回到原西文显示卡上。这种配制称为双卡单屏。若将处理双字节代码的硬件模块DBCS设计到VGA芯片中,并在原VGA卡的BIOS中增加能处理汉字的程序代码,这种汉卡既保持了原VGA卡的性能,又具备处理中西文的丰富功能,而不再需要原西文卡。这种配制称为单卡单屏。

硬汉卡上有256KB—4MB的内存空间,资源丰富。系统软件占用的基本内存小,开销省,为应用程序的运行留有充分余地。再者,硬汉卡中的很多处理不需要软件干预,占用CPU的时间少,使整机速度高。

(2) 软汉卡

软汉卡是利用西文显示卡的图形显示功能,由软件干预,采用程序作图方法,模仿西文文本态来显示汉字的部件。

在软汉卡中,原西文文本的显示机构不适合存放汉字代码和属性,必须重新申请一个存储空间,称之为逻辑显示存储器。通过转换程序,将逻辑显示存储器中的代码,转换成汉字字符发生器中的地址,取出字模点阵后经加工,再写入西文显示机构的图形显示存储器,汉字作为一个子图形,按文本态格式显示到屏幕上。我们称这种方式为虚拟文本态。软汉卡实际上就是硬字库卡。

2. 实用汉卡的种类

硬汉字系统通常由视频汉卡和CC-ROMBIOS构成。CC-ROMBIOS扩充了与汉字处理有关的中断调用,使系统在ROM硬件一层支持汉字处理,因此速度很快;视频汉卡则支持文本方式的汉字处理。目前我国应用较多的视频汉卡主要有长城和联想两大类。

长城汉卡出现较早,最初是在0520CH上实现的。它为系统提供了一种特殊的中西文共存而又互不干扰的环境。它将汉字库放入视频汉卡的字符发生器中,实现了字库不占内存,并以648×504的高分辨率支持16种彩色、每屏28行的文本汉字显示,使汉字屏幕界面得到很大改观。这种汉卡的显示方式较为特殊,它不同于PC-DOS字符方式显示,又不同于CC-DOS中采用的图形方式。使用时,如直接向屏幕缓冲区写ASCII码,则与原西文方式一致。当输出汉字时,可采用调用INT 10H中断,或经过特殊换算,将显示字符转换为一个15位的显示器放入缓冲区中,由硬件完成汉字显示。

长城汉卡工作在文本方式,软件汉化时不需要转换显示模式,所以要比在CC-BIOS环境汉化简单一些,但因显示汉字要做特殊转换,因此,汉化还是要做大量工作。长城汉卡设计是以高质高效显示汉字为出发点的,对兼容性考虑较少。其典型产品是014卡、CEGA卡和CVGA卡等。

联想汉卡舍弃了西文显示卡的文本显示功能,重新设计一套中西文兼容的文本显示电路,在原西文字符发生器内放入字模,并使字符发生器能区分汉字内码与ASCII码,相当于配备了中西文字符发生器,而显示卡接口和西文显示卡文本方式完全相同。联想汉卡与西文字符方式保持了较好的兼容,我们显示汉字时只需将汉字内码和属性写入文本缓冲区,即可由硬件完成汉字显示。这样汉字系统完全模仿了西文文本显示方式,所以理所当然支持中西文直接写屏与非直接写屏。

这种方式使计算机在硬件上支持汉字显示,是速度最快最彻底的中文环境。此方式计算机工作于文本模式,汉字显示又不需做特殊转换,因此汉化工作很简单。其典型产品有联想CSVGA卡、联想ASCII汉字芯片。但由于计算机的飞速发展,使这种技术受到很大限制,这是因为必须在剖析欲汉化的适配卡硬件构成的基础上,再增加新内容,故研制周期长。再者需额外增加硬件投资,费用也较高,所以这种技术一般只有计算机研制厂家采用。

此外,还有汉卡将软方法汉化的CCBIOS和汉字库固化,也能达到少占用内存空间的目的。

3. 汉卡概念的扩展

通常,汉卡只提供汉字的输入、显示和打印控制程序(即汉卡系统软件),但随着汉卡研制技术的不断提高,以及OA

(办公自动化)市场的逐步升温,汉卡已向 OA 卡和排版卡等方向转化。这样的汉卡中,有的既提供了系统软件,也提供了用于文字处理、排版印刷、表格制作及文档管理等应用软件,有的干脆只提供应用软件,而丢掉了汉卡本身的含义。

三、汉字的键入和显示输出

(一)汉字的键盘输入

1. 汉字编码

汉字输入是汉字系统的重要组成部分,其方法有键盘输入和自动输入两大类,目前广泛应用的仍是键盘输入。

键盘输入是在西文键盘上,借助于汉字编码技术输入汉字。汉字编码技术是将汉字用键盘字符串表示的方法或规则。汉字编码方案的优劣,直接影响汉字输入的效率和数量,因此,编码规则应简单易记,操作方便,码长要短,输入速度高,而且编码容量大,重码率低,在计算机上也容易实现。

汉字输入系统分为两个层次,即编码层次和软件层次。汉字以输入码的形式输入,这是编码层次;将输入码转成汉字内码,则是软件层次的工作。从发展趋向来看,编码层次应该是越来越简单,如果形成重码,则用智能软件加以识别。编码越简单,越会受到用户欢迎,越能更好地普及应用。

汉字编码技术是汉字系统中最为活跃的领域。目前,国内已研制出数百种编码方案。这些方案大体分为两大类:一类是无理码,如电报码、国标区位码等,它们是硬性规定的,只能靠死记硬背;另一类是汉字特征信息编码,现在绝大多数方案都属于这一类。特征编码又可分为如下几类:按照汉字发音特征的编码,称为音码;按照汉字字形特征的编码,称为形码;音形结合的编码,称为音形码;以及少部分既按照音形,又按照字义编码的,称为音形义码。

这些方案,经优选并在 IBM-PC 机上实现的有十余种,其中较典型,被经常使用的有三四种。例如:音码中的压缩拼音方案和双拼双音方案,前者需要记忆的量小,一学即会,常被初步使用计算机的人员,以及汉字处理工作量不大的广大技术人员所采用;后者编码短,操作方便,汉字输入效率较高,是目前唯一解决了单字无重码输入,并能作到高速盲打的音码方案,因而颇受欢迎。形码方案中的佼佼者,是当前呼声很大的五笔字形方案,它的输入效率甚高,但需要的记忆量大,特别适合专职录入人员采用。已推广开的典型音形码,是自然码编码方案,该方案音形结合、字词结合,输入效率高,又易学易用,也是一种很受欢迎的编码方案。

从用户角度看,汉字输入不外乎三种情况:一是“看打”,如打字员看着文稿打;二是“听打”,如做会议记录;三是“想打”,如教师、作家、记者、编辑等写文章、搞创作。这三种情况下汉字的信息来源是不一样的,“看打”是字形,“听打”是字音,“想打”也是字“音”,因为人写作时是用无声语言思考的,所以对大多数人来说,音码比形码更为合适,它不会像形码那样因拆字而打断思维。

各种汉字编码方案都面临着两个参数与重码之间的矛盾。这两个参数,一是编码长度:码长,重码少,但输入速度慢,码短,重码又会增多;二是编码规则:规则单一化,重码就多,规则复杂,重码就少,但难度又大。简而言之,就是好学的不好用,好用的不好学。用户需根据具体情况酌情选用。

汉字输入系统的潜在用户,不是打字员,而是千千万万的大众,随着计算机大量涌进办公室和家庭,一般干部、学生和广大非专业人员,将成为汉字输入队伍的主流。因此,以前的专业型编码(如五笔字型等)已无法适应新的需要,急需非常易学的普及型汉字编码。诸如集简易性与高效性于一身的“太极码”,以及最新推出的、所谓的“一看就会”编码——“五分钟

码”等,将会越来越受到广泛欢迎。

鉴于上述各方案在有关教材或手册中都有详细讲述,这里不再赘言。

2. 多种汉字编码方案

目前,微机的汉字处理系统都配备多种汉字编码输入方案,建立面向广大用户的、较完善的汉字输入体系,用户可以根据自己的需要与喜爱,选择合适的输入方法,以提高整个系统的使用效率和适应能力。

实现多种汉字输入方案,可以采用不同的方法。一种是在系统启动时,将每一种输入方案的处理程序,和相应的对照表,全部由磁盘加载到内存,这样,在输入方案转换时,不需要访问磁盘,转换速度高,但要占用大量内存空间。考虑到用户在输入汉字时,某一段时间只用一种输入方法,因此,实现多种输入方案的另一种方法,是把多种汉字输入处理程序及对照表,分别以文件形式存入磁盘,由键盘的功能键进行选择,任一时刻只有一种方案被加载到内存;方案转换时,采用 RAM 空间覆盖技术,从盘上加载新的处理程序和对照表,这样可以显著节省内存开销。

再者,汉字输入技术正在迅速发展,要求汉字系统具有一定的适应能力,这通常是设计一个通用软件接口,以便添加新的汉字输入编码方案。

下面以 IBM-PC 为例,介绍多种汉字编码输入方案的实现方法。

3. 汉字键盘输入的实现

整个汉字输入过程,可以分为代码识别和代码转换两部分。

(1)代码识别

从键盘缓冲区读入的字符,由代码识别程序进行识别,并分别进行处理。输入的可能是西文字符、汉字编码字符,还可以是一些功能符及方式转换符。

如果是功能符,代码识别程序完成该功能符定义的功能,各功能符的处理都有相应的处理程序。如果是方式转换符,代码识别程序则对输入方式进行相应的转换,包括清除原提示行内容,在提示行行首显示新输入方式的名称,结束原输入方式下输入的编码字符,开始新工作方式下的输入编码等。如果读入的是汉字输入码,则根据不同的编码方案,转至相应处理程序,把输入码转换成内码,并返回给 DOS。如果读入的是西文字符,则向 DOS 返回 ASCII 码。

(2)代码转换

实现输入码到内码的转换,通常有如下两种方法:计算法和查表法。

1) 计算法

若一种编码方案的输入码和内码之间存在函数关系,就可以采用计算法进行转换。

汉字内码和国标码之间通常有简明的对应关系,因此以国标码或区位码输入汉字时,只需通过简单的计算,就可以实现输入码到内码的转换。

汉字以区位码输入时,先把区位码转换成国标码,然后再把国标码转换成内码。区位码转换成国标码过程如下:取输入码缓冲区中的第1、2个字节(区位码的区号)压缩成十进制BCD码作为区号;再从缓冲区中取第3、4字节(区位码的位号),压缩成BCD码作为位号;然后,再将区号和位号由十进制转换成十六进制。由于区位码是从1区1位计起,而国标码是从2121H开始的,因此,十六进制的区号和位号,均应加上20H才能变成国标码;国标码两个字节最高位再加1形成内码,即完成输入码到内码的转换。

2)查表法

实际应用中,大多数编码方案的输入码和内码之间并无计算规则可循。此时,需要建立输入码和内码对照表(亦称输入码表或扫描表),用查表法实现转换。通常,输入码表的结构如图4-6所示:

输入码1	汉字内码1
输入码2	汉字内码2
⋮	⋮
输入码 n	汉字内码 n

图4-6 汉字输入码表

其中,每个汉字对应一个表项,每个表项分为两栏,一栏为输入码,另一栏为对应的汉字内码。

汉字输入时,系统根据输入的汉字编码,在对照表的输入码一栏中进行扫描,找出输入码的对应表项,并把该表项的内码取出,完成输入码到内码的转换。

查找这种对照表,常用顺序法和对分法。

①顺序法

顺序法是先从表头的第一个表项开始,把输入的汉字编码与表项的输入码逐个比较,相同时,取出该表项的内码,输入码到内码的转换即告完成。

通常,各种汉字编码的输入码之间不存在直接关系,每种编码方案都有一张自己的对照表。为了提高汉字输入速度,对照表须常驻内存,因此,在设计对照表时,既要考虑查找速度,又要注意尽可能减小其规模。例如,由于内码和国标码之间有简单的换算规则,所以,只要把输入码表的表项按国标码的次序排列,那么根据查表所得到的表项序号,就很容易算出相应的国标码,从而即可得到汉字内码。这样,就可以省去上述对照表中汉字内码一栏的内容,节省了内存开销。

当两种编码方案之间有某种依赖关系时,也可以进行简化。

例如 CC-DOS 采用的首尾码,是对汉字字形进行简化后规定的编码,它将汉字的左上部笔画作为首码,右下部笔画作为尾码。用这种方法表示的汉字有重码,因此,在首尾码后可再添加一个“首音码”,即该汉字拼音的第一个字母,以减少重码率。它的输入码最多为三码。CC-DOS 的拼音码也定为三码输入,将一些声母和韵母加以简化,使拼音字母多于3个的汉字也能以三码输入;此外,为了克服该方案中重码汉字多的缺点,也允许拼音码和首尾码联合使用,即前三码为拼音码,最后一码为其首尾码中的首码。由此,在一个表项中,首尾码的第3个码可以和拼音码的第1个码台用,利用这种关系,即可将两种输入码的对照表合为一张,以减少所占用的内存空间。

②对分法

按照输入码表逐次对分进行查找的方法,称为对分法。

对分法查找,要求对照表中的输入码一栏,按照编码的递增或递减顺序排列。查找时,把输入的汉字编码先与对照表中中间表项的输入码进行比较,以确定该编码处于对照表的前半部还是后半部。若在前半部,再在前半部进行对分比较;若在后半部,则在后半部再对分比较。依次类推,直到查到与该汉字编码相同的输入码,则把输入码所处表项中的内码取出。对分法的转换速度比顺序法要快,但它要求对照表中的输入码

必须是一个有序排列。

(二)汉字的显示输出

1. 实现机制

实现汉字显示的过程,是把汉字机内码转换成屏幕上的汉字字型点阵的过程,图4-7给出了 CC-DOS 实现该过程的软硬件组成框图。

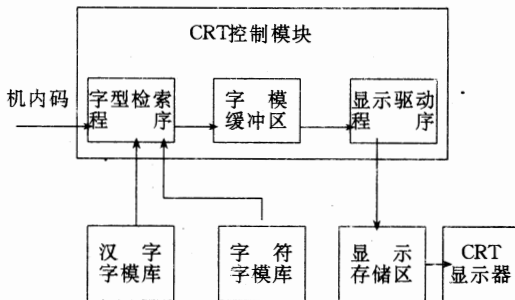


图4-7 CC-DOS 实现汉字显示输出的软硬件组成框图

上图中,送入 CRT 控制模块的是汉字机内码;字型检索程序把机内码转换成相应汉字字模在汉字库中的地址;根据此地址,从汉字库中取出汉字字模,经字模缓冲区加工处理后,形成汉字字形码,送往显示驱动程序;显示驱动程序再把汉字字形码分解后,分别送往显示存储区的偶数线扫描区和奇数线扫描区,在 CRT 控制器的控制下,屏幕上显示出汉字字形。

其中,需要说明的是,此处字符字模库中存放的是 ASCII 字符的点阵信息,供显示器在图形方式下显示字符使用。显示器在字符方式时,显示的字符由字符发生器形成;在图形方式时,字符发生器不起作用,故必须配用字符字模库。

显示存储区用于存放显示信息,CRT 控制器定时将该区内容读出,每秒钟重复60次,于是就可以在显示器上稳定地显示这些内容。

字模缓冲区是内存中对显示字模进行加工的场所,系统把从汉字字模库或字符字模库中取出的字模信息,在这里排列成符合显示器要求的格式,然后再送去显示。在 CC-DOS 中,字模缓冲区共36个字节,前32个字节存放汉字字模信息(字符字模只用前面的一半),两个字节存放一个汉字字模,偶数、奇数字节分别存放汉字左半和右半字模;后面4个字节用于存放“空白”,以形成显示的行间距。

机器启动时,系统要对 CRT 进行初始化,包括确定 CRT 的工作方式(这由显示中断调用 INT 10H 的0号子功能完成),并设置显示存储区、字模缓冲区以及一系列工作区的初始状态等。

2. 实现方式

如前述,汉字系统分为软汉字系统和硬汉字系统,二者汉字显示的实现方式是不同的。

(1)软汉字系统中的实现方式

软汉字系统中,显示分为“非直接写屏”和“直接写屏”两种方式。

“非直接写屏”方式即通常的屏幕显示方式,利用屏幕显示中断 INT 10H,将显示器设置为图形模式,并用其相应子功能进行显示。系统对 ROMBIOS 的 INT 10H 进行了改造,使之可以支持汉字。

汉字显示过程中,系统还采用了虚拟显示缓冲区的概念。所谓虚拟显示缓冲区,是汉字系统在 DOS 内存区另行开辟的

一个缓冲区,其格式完全模拟西文文本方式显示缓冲区的格式,以便与西文软件兼容。当通过 ROMBIOS 中断调用进行屏幕显示时,汉字系统所仿真的 BIOS 中断,截获发来的数据(汉字内码或 ASCII 字符及它们的属性),先将它们存放在虚拟显示缓冲区中,通过字模管理模块提取点阵信息,再将字型点阵及属性写入实际的显示缓冲区(西文的图形显示存储器)中,实现图形方式下汉字及其它信息的显示。非直接写屏广泛应用于早期的汉字系统中,由于这种方法相对简单,至今一些汉字系统依然采用。

现今市场上的软汉字系统中,“直接写屏”者占有很大优势。它们既支持通过正常的 BIOS 调用方式进行的屏幕输出,也支持通过直接写屏方式进行的屏幕输出,性能显著优越。

所谓直接写屏,即不通过屏幕中断 INT 10H,而是用传送命令将待显内容直接送显示缓冲区进行显示,因而显示速度显著加快。

在汉字状态下的直接写屏系统中,系统初始化时,将显示模式隐含设置为西文图形方式,并另行开辟一个缓冲区,大小与西文文本显示缓冲区相同,称显示比较缓冲区。待显示的汉字内码或字符的 ASCII 码及属性,直接写到西文文本显示缓冲区中。汉字系统通过时钟中断 INT 1CH,定期地对西文文本显示缓冲区和汉字系统的显示比较缓冲区进行扫描,比较两个缓冲区有无不同,如有,则更新汉字显示比较缓冲区的内容,并通过字模管理模块,提取被更新的汉字内码或 ASCII 字符的点阵信息,写入西文图形显示缓冲区中,通过西文显示卡的处理,进行显示。

在直接写屏方式下工作,不仅显示速度快,而且汉字与西文的兼容性也好,因此受到用户的欢迎。

支持直接写屏的汉字系统,很多还支持西文制表符的智能判别。西文制表符是一个字节的扩展 ASCII 码,字节的最高位也为1,这样,西文制表符同汉字似乎无法区别开来。但是,通过分析西文制表符和汉字的排布规律发现,在软件中,西文制表符的应用有明显的排布规律,而汉字通常只是为了表达某种语意,汉字与汉字之间的排布情况多是随机的,并无明显规律可循。因此可以认为,在西文制表符的排列规律下,汉字按相似规律排布的概率几乎为零。根据这种特征,汉字系统即可将绝大部分西文制表符与汉字区分开来。

(2) 硬汉字系统中的实现方式

硬汉字系统中采用文本方式显示汉字。但它不采用原西文显示卡的文本显示机制,而是重新设计了一套中西文兼容的文本显示控制线路,这套线路有一个可以区分汉字内码和西文 ASCII 码,并能产生中西文字模的字符发生器。硬汉字系统完全模拟西文文本显示功能,既支持“非直接写屏”,也支持“直接写屏”。

四、汉字操作系统

汉字系统必须结合于操作系统,才能运行。包含汉字系统的操作系统,即称为汉字操作系统。鉴于普及应用的趋势和需要,下面就较典型的微机汉字操作系统进行概要分析介绍。

A. 汉字操作系统分类

目前的微机汉字操作系统,作为一种支撑环境,分为 DOS 类和 Windows 类两大类。

(一) DOS 环境汉字操作系统

西文操作系统 PC-DOS 是 PC 系列微机的主操作系统。DOS 环境下的汉字系统,是指附加在原西文 DOS 环境的一层汉字“外壳”。这层“外壳”由纯软件或软件和硬件构成,通过修改 ROM BIOS 中有关输入输出的中断调用,来截获西文环

境下的输入输出,并加入汉字 I/O 功能,使之实现汉字的输入、显示及打印,具备汉字处理功能。因此,

汉字系统=西文系统+汉字处理功能

这样,汉字系统就把西文 DOS 转化为可以处理汉字的操作系统,即汉字操作系统。

汉字输入、汉字显示及汉字打印,是汉字信息处理系统的三项主要内容,它们都必须由汉字操作系统来实现,或受汉字操作系统的支持。

上述的汉字系统“外壳”,把 BIOS 改造为 CC-BIOS,这样形成的汉字系统,称为 CC-BIOS 汉字系统。它实现汉字处理功能的方法是,使计算机工作于图形模式,模仿西文文本缓冲区,在 DOS 内存缓冲区中开辟一块虚拟缓冲区,以保持与西文软件的兼容。当计算机通过 BIOS 显示输出时,CC-BIOS 截获此信息,将其放入虚拟缓冲区,再据此读取相应的汉字或西文点阵,送往图形方式下的真正显示缓冲区进行显示。这样,尽管真正的屏幕处于图形方式,而用户程序面对的却是文本方式的虚拟缓冲区,就可以用文本方式处理汉字,使汉化软件的兼容性得到一定保证。这种系统实现起来较简单,且不需额外的硬件投资,故被广泛采用,此类典型的系统有:CC-DOS 2.0、CC-BIOS 2.13(A-H)、UCDOS(1-2.2)、WMDOS(1.0-5.0)等。

另一类汉字操作系统,不是采用 CC-BIOS 的单纯软件改造方式,而是对有关的软硬件进行重新设计,使系统的汉字功能更为丰富合理,但通常需增加额外的硬件开销,如上述长城、联想汉卡硬汉字系统等。

前述的支持直接写屏汉字系统,称为中西文兼容系统。在这种系统环境中,西文软件大多不需汉化即可使用,显示速度快,中西文兼容性好,一出现就颇受欢迎。其典型产品有:《中国龙》、《天汇》、《启明星》等。该系统是 DOS 汉字操作系统的主流和方向。

(待续)

广东省潮阳市陈店环东电子有限公司

电子元器件

陈店——汕头特区的所辖镇,它以其天时、地利、人和,而成为全国最大的电子市场之一,陈店环东电子有限公司正是拥有这有利条件,而使她的业务范围遍布全国各地,经营各类进口电子元件品种达到几千种型号,一直以来,都在北京的《家和电器》《电子世界》《家电维修》《北京电子报》上海《现代通信》《实用无线电》成都《电子报》广东《广东电子》等杂志、报刊上刊登邮购广告,和全国各地许多客户、厂家公司建立了长期友好的业务往来,因此,如果你想得到该公司的最详细报价目表和简介,那么,请你来函付资贰元索取。热情欢迎来函、来电、咨询、洽谈合作、邮购。

地址:广东省潮阳市陈店镇东方一路88号

电话:公司(07644)481478 邮购部:484573

传真:485649 邮编:515152

经理:蔡镇荣 手提:(0754)9019654

MCS—51单片机控制的DTMF

信号发送/接收电路

廖天河 李亚平 郑州大学物理系(450052)

本文介绍的双音多频(DTMF)信号发送/接收器,是利用单片机8031控制集成电路MV5087和MT8870完成的。该发送/接收器电路体积小,工作稳定,抗干扰性强,可广泛应用于有线、无线通信设备和控制系统中。

一、DTMF 发送电路

双音多频信号DTMF的发送电路,在单片机8031的控制下,以MV5087为核心,外接标准晶振(3.58MHz)和用8中取2标准键盘构成的4×4键盘矩阵等组成,如图1。每按下一个数字或功能键,则行与列的对应输入端为低电平。这样,就形成一种双音频组合,以产生相应的双音多频信号。电源电压采用+5V,音频无效端悬空,单音禁止端接V_{ss},当有任一键按下时,静噪输出端(10脚)为高电平,当计算机检测到高电平后,通过I/O口对发送系统进行控制,这样DTMF信号从16脚输出经发送电路发送出去。

二、DTMF 信号接收电路

双音多频信号DTMF的接收电路是在单片机8031的控制下以大规模集成电路MT8870芯片为核心,外接标准晶振(3.58MHz)和有关电子元件组成。电源电压为+5V,电路如图1。

DTMF信号经过C₁、R₁输入到IN₋端,DTMF的接收增益由R₂决定。IN₊与V_{REF}短接,给IN₊端提供V_{DD}/2的参考电压作偏置电压,电压增益A_v=R₂/R₁。C₂、R₃组成外部定时电路,以保证芯片EST和S₁/GT正常工作。Q1~Q4输出的二进制编码信号由使能端TOE控制,TOE为高电平时,Q1~Q4输出与当前输

入的DTMF信号相对应的二进制编码,当TOE为低电平时,Q1~Q4端呈高阻状态。DTMF信号与二进制编码信号的对应关系如表1。

输入的DTMF信号经8870解调后,在S₁D端(15脚)产生一控制输出信号,该信号输出端与单片机8031的外部中断输入端INT₁相连,当该信号发生由1到0的跳变时,中断标志IE₁置1(中断安排为下降沿触发方式),8031响应中断后,产生一个RD信号,信号经反相

表1

Flow	F _{high}	NO	TOE	Q4	Q3	Q2	Q1
697	1209	1	H	0	0	0	1
697	1336	2	H	0	0	1	0
697	1477	3	H	0	0	1	1
770	1209	4	H	0	1	0	0
770	1336	5	H	0	1	0	1
770	1477	6	H	0	1	1	0
852	1209	7	H	0	1	1	1
852	1336	8	H	1	0	0	0
852	1477	9	H	1	0	0	1
941	1336	0	H	1	0	1	0
941	1209	*	H	1	0	1	1
941	1477	#	H	1	1	0	0
697	1633	A	H	1	1	0	1
770	1633	B	H	1	1	1	0
852	1633	C	H	1	1	1	1
941	1633	D	H	0	0	0	0

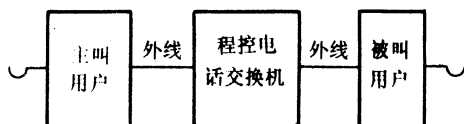


图2 DTMF 用于电话拨号

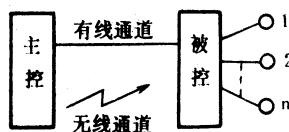


图3 DTMF 控制系统

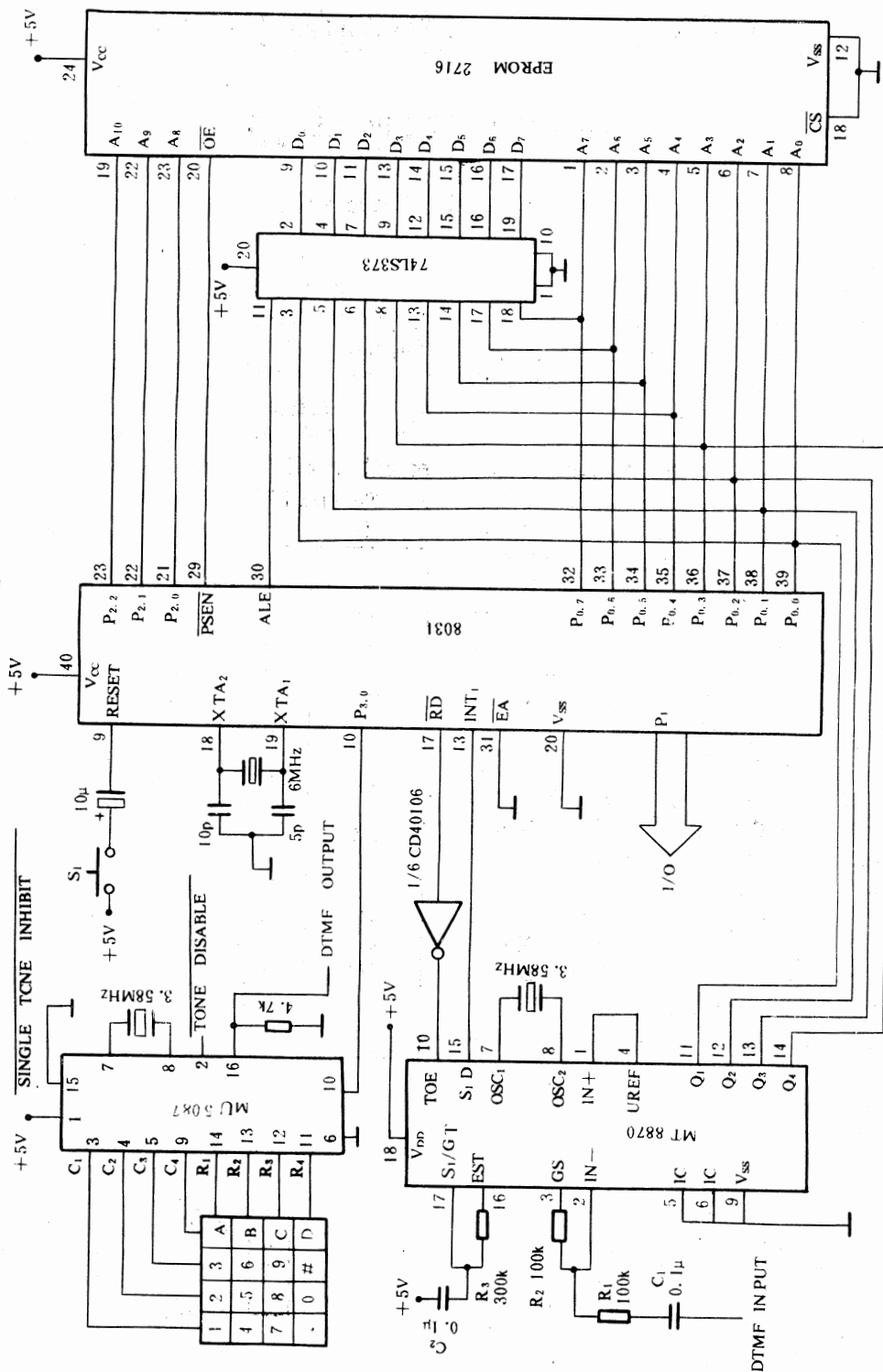


图1 单片机控制的DTMF发送/接收电路

后使 MT8870 的 TOE 端产生一个高电平脉冲信号,该信号使 MT8870 的数据输出端 Q1~Q4 由原来的高阻状态变为与当前输入的双音频信号相对应的二进制编码信号,8031 通过 P₀ 口将该二进制数读入并保存在数据 RAM 区,将读入的数据处理后通过 I/O 口对系统进行控制。

三、DTMF 发送电路与接收电路在通信与控制方面的应用

1. DTMF 电话交换系统

把 DTMF 发送/接收电路与电话机电路相结合,就构成了 DTMF 电话机,用户就可以利用电话机上的数字与功能按键实现 DTMF 电话拨号,如图 2 所示。主

叫用户摘机按键拨号后,电话号码所对应的 DTMF 信号通过电话线传到程控交换机中的 DTMF 接收电路,交换机中的微机识别被叫电话号码后,接通主被叫用户实现双方通话。采用 DTMF 拨号,在电话线上传输的是双音频信号而不是传统的脉冲信号。因此传输距离远,抗干扰能力强,可靠性高。

2. 用于自动控制系统

如果把 DTMF 的发送电路用于主控系统,接收电路用于被控系统,就可以方便地组成有线或无线通信系统,如图 3 所示。其通道数量视需要而定,16 通道以内每通道只需编一位号码即可,若需要更多通道,则可象电话编号一样编为两位或两位以上的号码。

片内存储器 RAM

胥筱汀 北京西城区青少年科技馆(1000

8031 单片机没有片内程序存储区(ROM),有 256 字节的数据存储区(RAM)。其结构如图 1 和表 1。

访问内部数据区,用 MOV 指令。

8031 的内部数据区分成两块,00~7F 和 80~FF。在 00~7F 范围内又能区分:

00~1F 的 32 个字节分为 4 组,称为工作寄存器。在每组内均以 R₀~R₇ 的名称出现。选择哪一组,由状态寄存器中的 RS0 和 RS1 这两位来选择。它的主要功能是用作循环计数器,各组的 R₀ 和 R₇ 可用作内部数据和

外部数据存储器间接寻址的指针。当它们不作为工作寄存器使用时,可作为一般数据缓冲器使用。它们的特点是只需指令操作码中的三位地址即可访问,因此,指令短,存取速度快。经常用于存放一些常用数据或中间结果。

20~2F 的 16 个字节除了可作一般存储器使用外,它们的特点是可以按位寻址。它们每位的地址号为 00~7F,见图 1 右侧,访问它们时用位操作指令。

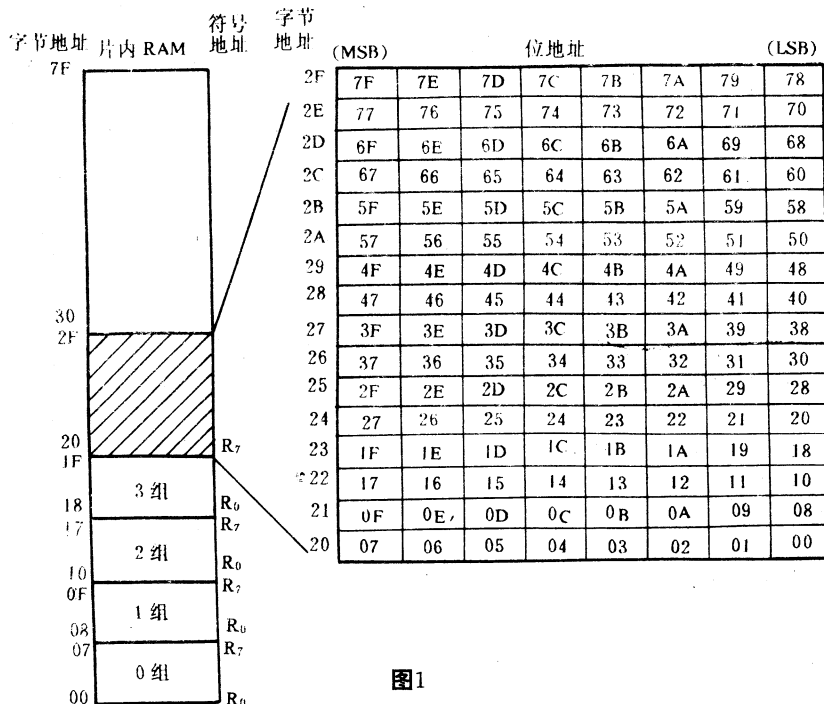


图1

30~7F 范围内除作为一般存储器使用外,经常用它一部分字节作堆栈使用。

80~FF 称特殊功能寄存器区,共128字节,对于8031来说,只有21个字节有定义。有一些是可以进行位操作的,见表1,其中有些寄存器就是8031内部的有关硬件,访问这些寄存器,就是对硬件进行操作,故称为专用寄存器。

8031在 DP-888中的连接。

8031在 DP-888中的连接如图2。

图中8031的 RST 端接主板的复位按钮。按下复位按钮,RST 端被接到+5V,变成高电平。若接通维持两个机器周期以上的时间,即可使单片机复位。RST 端接一个电阻到地,不按复位按钮时,保持低电平。在 Vcc 和 RST 端之间跨接一个电容,便可在开机时,自动使8031复位。这是因为电容两端电压不能跳变。开机前电容器两端电压为零,开机后,5V 电压立即加到 RST 端。电容充电,两端电压逐渐升高。RST 端电压从5V 慢慢下降。只要电容足够大,充电时间满足复位时间的要求,即可实现开机自动复位。

复位,是芯片的初始化操作。它将一部分专用寄存器置成如下状态。

专用寄存器名称	状态
PC	0000
ACC	00
PSW	00
SP	07H
DPTR	0000
P0~P3	FF
IP	XX000000B
IE	0X000000B
TCON	00
TL0	00
TH0	00
TL1	00
TH1	00
SCON	00
SBUF	00
PCON	0XXX0000B
TMOD	00

复位不影响片内 RAM 低128字节的内容。它使 ALE 和 PSEN 两个控制信号变为高电平。

当复位信号 RST 恢复为低电平后,单片机开始执行程序。由于复位后程序计数 PC 的内容为0000H。因此单片机从0000H 地址执行程序。因此当程序运行有错误使机器锁死,可按复位键重新启动。

8031的EA端固定接地,表示它只访问外部程序存储器。XTAL₁ 和 XTAL₂ 端外接一个6MHz 的石英晶体,以示8031的时钟频率为6MHz。INT0是外部中断0允许端。它接于单步/连续开关。当INT0接地,允许外部中断,每按一下“STEP”键,8031执行一条指令,这就是单步运行。当INT0接高电位时,8031全速连续运行

程序。

8031的 P0口为地址/数据复用口,它先送出地址的低8位,用 ALE 信号将其锁存在74LS373中,然后才传送数据。

8031的PSEN和RD信号经过74LS08的与门,产生一个综合信号,送到6264芯片的OE端。因此6264芯片既可作为程序开发区,也可作片外程序存储区两个功能。

8031的 TXD 和 RXD 端连接的是串行口的接口电路。通过它可实现与 PC 机的通信。

8031的 P1口和 P3口的部分引脚,只是引到8031总线插座上,在此图中未画出。

存储器扩展

单片机内部的存储器容量较小,往往不够用。这时需要外接存储器。对于8031单片机来说,片内没有程序存储器,必须外接。DP-888有关存储器扩展的部分电路见图3。

27128是可改写程序的只读存储器,容量16K 字节。其内部存储了两个监控程序:PC 监控和键盘监控(由 K₂进行转换)。6264是随机读写存储器。其容量是8K 字节。

74LS138是三一八译码器。其输入端 A、B、C 连接在 P_{2.5}~P_{2.7}上。此时译码器将64K 字节的存储器地址

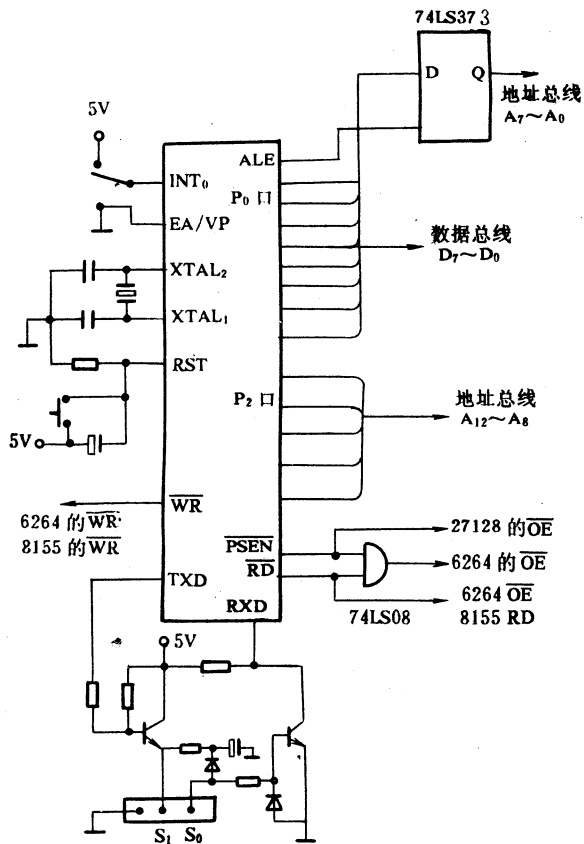


图2

分成了8组。每组为8K字节的空间,DP-888只用了Y₀和Y₄两组。其地址范围是:

Y₀:0000~1FFF

Y₄:8000~9FFF

将Y₀作为27128的片选信号;Y₄作为6264的片选信号。当计算机访问00~1FF范围地址时,就选中了27128,当访问8000~9FFF范围地址时,就选中了6264。

单片机访问程序存储器和访问数据存储器的指令是不同的。访问程序存储器时,CPU产生PSEN \bar 信号,它联到27128的OE \bar 端。此时可以读出程序。访问数据存储器时,CPU产生RD \bar 或WR \bar 信号,对数据存储器进行读出或写入。为了兼顾开发和应用两个方面的需要,加了一个与门将PSEN \bar 和RD \bar 相与之后,送到了6264的OE \bar 端。这就使6264芯片具备了开发程序和运行程序的功能。

I/O 扩展

8031单片机采用外部存储器的方案,P0口和P2口都作为数据总线和地址总线用了。实际上只有P1口作为输入和输出的口线,选用8155芯片实现I/O口的扩展。图4是其引脚图。

8155内部由下列几部分组成:256字节的静态存储器;两个8位和一个6位的I/O口;一个14位减法计数器。

引脚功能如下:

RESET(复位):复位信号输入端,高电平有效。

8155复位后,将三个输入/输出口置成输入方式。

AD0~7(地址/数据总线):与8031的P0口相连。在ALE的后沿将低8位地址锁存在锁存器中。该地址可供输入/输出口或存储器使用,取决于IO/ \bar M的电平。传送数据时,其方向由WR \bar 和RD \bar 决定。

CE(片选):低电平有效。

RD(存储器读):低电平有效。当CE=0,且RD=0时,存储器或I/O口的内容将读至AD线上。

WR(存储器写):低电平有效。当CE=0,且WR=0时,AD线上的数据或写入存储器(IO/ \bar M=0),或写入I/O口(IO/ \bar M=1)。

ALE(地址锁存允许):这个信号的后沿把AD0~7线上的地址和IO/M信号锁存在器中。

IO/ \bar M(输入输出/存储器选择线):该线处于低电平选择存储器;处于高电平,则选择输入或输出。

PA0~7:口A输入/输出线。8位双向。

PB0~7:口B输入/输出线。8位双向。

PC0~5:控制信号线。这6条线可以作输入/输出线,也可以作为口A和口B的控制线。

Timer in:定时器/计数器输入端。

Timer out:定时器输出端=输出方波还是输出脉冲,取决于计数器的工作方式。

8155的工作原理:

8155的输入/输出部分由四个寄存器组成:命令/状态寄存器(C/S),PA寄存器,PB寄存器和PC寄存

器。8155内还有一个14位定时器。这些寄存器或定时器占用的I/O地址如下:

地址(最低三位)	选中的寄存器
000	内部命令/状态寄存器
001	PA口
010	PB口
011	PC口
100	定时器的低8位
101	定时器的高6位和计数器方式

注:IO/ \bar M=1时,以上地址才能选中适当的寄存器。

命令/状态寄存器(C/S)作为命令时各位的定义:

位:	D7	06	D5	D4	D3	D2	D1	D0
功能:	TM ₂	TM ₁	IEB	IEA	PC ₂	PC ₁	PB	PA

PA:定义PA口,该位为0,定义PA口为输入。该位为1,则定义PA口为输出。

PB:定义PB口,该位为0,定义PB口为输入。该位为1,则定义PB口为输出。

PC₁和PC₂两位共同定义PC口,四种情况:

1. 两位都为0,为ALT1,定义PC口为输入。
2. 两位都为1,为ALT2,定义PC口为输出。
3. D3为0,D2为1,为ALT3;

PC₃~₅为输出。PC₀为PA口中断申请线,PC₁为PA口缓冲器满信号,PC₂为PA口中断开放信号。此时三条线为PA口与外设的联络线。

4. D3为1,D2为0,为ALT4;

PC₀~₂为PA口与外部设备的联络信号,与ALT3相同。PC₃为PB口中断申请线,PC₄为PB口缓冲器满信号,PC₅为PB口中断开放信号。

IEA定义为PA口中断允许。

IEB定义为PB口中断允许。

TM₂和TM₁两位是对计数器的命令:

00为空操作,对计数器工作无影响。

01为停止命令。如果计数器还在工作。则停止计数,如果计数器没有启动,则无操作。

10为TC后停止,在完成现行计时器周期后,立即停止。

11为启动命令。装入工作方式和计数长度,并立即启动。如果计时器还在运行,则在完成现行计时周期后,立即启动新的状态和计数长度。

命令/状态寄存器(C/S),作为状态寄存器使用时,其状态格式如下:

D7	D6	D5	D4	D3	D2	D1	D0
—	Timer	INTEB	BBF	INTRB	INTEA	ABF	WTRA

INTR为中断申请,若D0=1,则为PA口申请中断,D0=0,则PA口未申请中断。

BF为缓冲器,D1=1,则为PA口缓冲器满,D1=0,则为PA口缓冲器空。

计算机网络管理系统 在旅行社的设计及实现

广西师范大学 崔晨荣 方 勤

桂林市以其独特的自然景观(岩溶地貌)和人文风情(少数民族地区)吸引着世界各地的旅游观光团及海内外游客。每年接待国内外游客近百万人次,由于游客每年的递增,先后应运而生出众多独资和合资旅行社。桂林现有各类旅行社三十多家。目前,大多数旅行社仍使用手工管理、手工统计和结算,一些旅行社虽有微机,也仅限于处理工资报表和办公文印。并且各类旅行社的管理特点是实时性强、随机性大、信息量散、统计量大、报表多及通讯频繁等。若采用传统的手工操作流程或单用户计算机管理系统已不能适应实际要求,因此,在旅行社采用计算机网络管理系统将成为各旅行社今后所必备的工作界面。基于此类旅行社的管理特点,我师大计测中心与桂林阳朔旅行社共同设计开发了旅行社计算机局域网管理系统。

一、系统设计要求

工作介面全部汉化、菜单化、系统易于维护。

建网后留有扩充余地,易方便挂接新增工作站,能充分利用有盘工作站(单机运行)。

采用防病毒系统,加强网络系统的安全性。

各职能部门管理需设置保密字,防止侵权行为,保证系统的可靠性。

系统的管理方式和结算方式分别按国际旅行社、中国旅行社和福中旅的标准进行处理。

财务管理除具有本行业自身的特点外,系统设计要考虑与通用财务软件的接口。

财务部和业务部对每个团(队)进行并行核算处理。即两部分别结算。最后汇总核查,以确保数据的可靠性和准确性。

业务处理模块在制定团(队)行程计划过程中,需保留所有更改过的记录并存档,以备核查。

二、系统的可行性分析及确定

目前,国内旅行社还没有完善通用的旅行社计算机信息管理系统,亦由于各类旅行社采用不同的标准,管理方式也各异(即没有统一的管理模式),采用单用户微机系统已不能完全替换传统的手工管理、数据统计和结算。为此,我们经实地调研和收集大量的第一手资料,经过深入细致地可行性分析,我们认为要摆脱陈旧的手工操作、提高工作效率,并适应国际上旅游业的发展趋势,只有选择计算机网络管理系统,才能取代繁

锁的传统管理模式,并做到相关部门对数据进行并行处理和资源共享等。鉴于上述分析,网络系统采用已成熟且在国际上已广为流行的 NOVELL 网,软件开发平台采用 SSBMIS 信息管理系统进行全过程软件开发。系统开发采用搭积木(模块独立)方式逐步达到系统设计目标。

三、网络系统配置

硬件组成:

网络服务器是整个计算机网络的核心,其性能对整个网络系统的影响至关重要,为确保网络服务器的可靠性,经考核,采用美国 USALINK486/33(内存 4MB,硬盘 210MB,1.2MB+1.44MB 软驱 VGA 彩显)作为网络服务器。(服务器可采用镜像双工或是双服务器来保证系统的安全、可靠和正常运行)考虑旅行社的具体需要及硬件系统的可靠性,同时在国外业务部配置一台 USALINK386/33(硬盘及其它配置同服务器);另外在经理部,财务部,国外业务部及国内业务部,分别配置同机型 USALINK286/16 作为无盘工作站。以提高硬件系统的互换性。

与之配套的网卡采用 32 位高速网卡安装在网络服务器与有盘工作站,以提高网络软件的运行速度,在无盘工作站配备与之匹配的兼容网卡。

传媒介质采用 50 欧姆同轴细电缆及其配套使用的 T 型头, BNC 接头和终端器等。

考虑到桂林地区供电不稳因素,为整个网络系统配置了磁共振稳压电源以及不间断电源 UPS,以提高整个网络的安全性和可靠性。

软件组成:

通用软件使用 DOS3.3 版本。

网络操作系统为 Netware 386 V3.11。

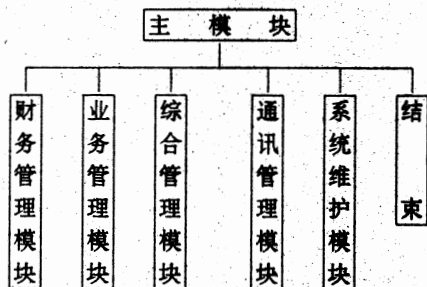
网络汉字软件选用国内首推的晓军 2.13K 网络汉字系统。

开发软件使用 Mfoxbase+V2.10。

为避免计算机病毒的侵袭,采用国际上先进的网络防病毒系统 VSALANProtect 及 PC-cillin(用于网络服务器和有盘工作站)。

四、系统设计

本网络系统采用模块化结构设计,系统功能模块如下图所示:



各功能模块设计如下

1) 财务管理模块

包括国外团(队)财务、国内团(队)财务、系列(长线)团(队)财务各子模块。其它子模块用于显示、查询旅行社组团(队)情况,以及团(队)的住房、用餐、游览等情况,各类报表的产生,数据检索、统计等。

2) 业务管理模块

由于与财务部结算作并行处理,对应的有国外、国内、长线团(队)业务模块,主要用于制定各种计划(制定旅行团行程总计划,用车、用房、用船、订票计划),产生导游接待表和总计划行程表。

3) 综合管理模块

包括财务部和业务部各种报表的显示和打印。信息的检索、列表、阅读、删除、存档;社领导可通过网络通讯向下属发送指令以及日常行政事务处理等。

4) 通讯管理模块

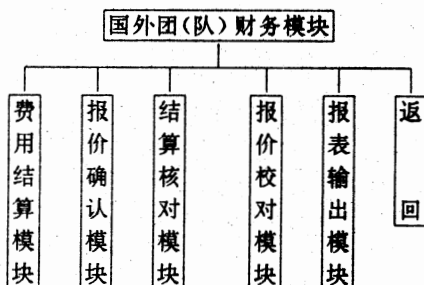
系统设计中允许旅行社各部门之间进行通讯,其中包括:信息的发送和接收,以及对信息的各种处理(检索、列表、阅读、删除、打印等)。

5) 系统维护模块

包括系统数据、程序的硬软盘备份和恢复,某数据文件的添加,系统数据库整理及执行操作系统命令等功能。

五、系统实现

主要介绍财务管理模块中国外团(队)财务,该子模块流程见下图。国内团(队)和系列(长线)团(队)财务模块类同国外团(队)财务模块。



1) 费用结算

该子模块对每一个团(队)的情况(诸如:抵离时

间、航班次、人数、年龄……)、住店情况、用餐情况、游览情况以及用车、船、购票等数据进行录入建库,具备全屏编辑功能(增、删、改等),在确认录入的同时系统自行完成相应的费用结算(按三种标准之一:国旅、中旅和福中旅),并可随时显示或查询任一团(队)的各项费用结算情况或统计数据(按日、月、年统计),可以根据实际需要设定新增项目,为系统提供结果数据。

2) 报价确认

该子模块主要完成旅行社制定预接待团(队)各项开支的核算,并产生报价确认单,通过传真或邮递方式交由对方有关组团社进行核实。

3) 结算、报价校对

根据财务管理制度,经费和收支必须履行出纳经手,会计核实的流程。为此,系统实现中用两个相对独立的模块(业务和财务)并行处理费用结算和报价确认,构成了平行处理核算系统。也就是说,财务和业务两部门分别结算,之后相互比较、核查。而这两个子模块可以实现上述功能的自行完成。并且财务部具有随时查询业务部信息和比较、核对业务部处理数据的优先权。

4) 报表输出

该子模块在报表产生前,先对各项结算费用数据进行统计,根据不同关键字检索打印出单团结算收支表,收入及成本、支出表;核算报表及汇总表;费用汇总表;分项应付和费用拨款结算表;应收结算明细表以及相应的各种日、月、年报表。

六、系统特点

在系统开发研制过程中,采用了比较成熟的先进技术,从多方面保证了系统的设计要求。在设计上综合了国旅、中旅和福中旅等几大管理方式,以国外团为主,具有综合性强的特点;财务管理具有本行业的特点,并且配有通用财务系统接口。该系统选用友好的用户介面,采用全屏幕操作,介面清晰直观,上推、下拉式多级菜单,操作简便;报表输出项全部采用自定义函数和数组,修改方便;对较长的字符字段数据的输入和报表打印输出采用了字段分割和字段合并等技术,为方便用户操作,数据输入全部采用表格化输入技术,尽量减少汉字输入而代之以代码。系统建有公共数据库,减少了数据的冗余和系统开销。根据不同权限的用户设置相应的保密字,杜绝越权行为。

本计算机网络管理系统在阳朔旅行社试运行过程中,已充分显示出高效率的管理特色,但如何将其开发成为一个各类旅行社通用的计算机网络管理系统并使之推广应用,还有待进一步完善。

该计算机网络管理系统在旅行社的设计与实现,为桂林的旅游业适应世界旅游业的现代化管理迈出了可喜的一步,同时也为旅游业的发展起到了积极的推动作用。

巧用 PC 机并行口

徐令元 武汉金融高等专科学校(430061)

在本刊93年第5期《PC 机并行口的扩展》一文中,作者指出 PC 机的并行口已接成打印机使用状态,故要用作其它 I/O 是比较困难的,并介绍了自制并行口扩展板电路的方法。

本文介绍另一种 PC 机并行口的扩展方法,这种方法十分简单,不需要制作或增加任何硬件电路,只需对原有的并行打印口作一处改动,即可使之成为通用的并行 I/O 口。

一、并行打印口的局限性

并行打印口的电路图在许多 PC 机的书籍中都有介绍,这里仅画出其中的数据输入缓冲器(U3, LS244),数据输出锁存器(U4, LS374)及控制输出锁存器(U7, LS174)。

由图可见,数据输入缓冲器 U3 的输入端与数据输出锁存器 U4 的输出端都与 25 芯插座的 2—9 端相连,由于 U4 的 1 脚(\overline{OE})始终接地,其输出端不能处于高阻状态,这样便影响了通过 25 芯插座 2—9 端输入数据,所以 PC 机并行打印口只能用作输出口而不能用作通用的输入口。

二、改进方法

要把并行打印口用作通用的 I/O 口,必须能按需要改变 U4 的 1 脚(\overline{OE})的电平:作输出口时,保持低电平;作输入口时,处于高电平,这时 U4 的输出端处于高阻状态,对经 25 芯插座 2—9 端输入的数据没有影响。

从图中还可看出,控制输出锁存器 U7 是一个六 D 锁存器,有六个输出端,而并行打印口仅用了其中的五个(用于送出控制信号),我们正好把剩余的一个输出端(15脚,6Q)用于控制 U4 的 \overline{OE} 端。

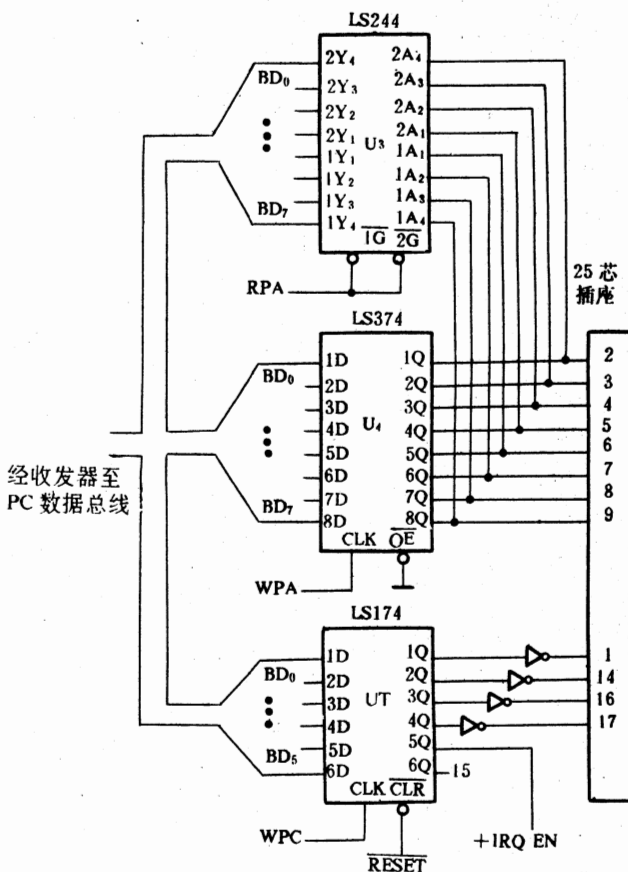
可见,只需对原有的并行打印口作一处改动,即把 U4 的 1 脚(\overline{OE})与地的连线断开,并接到 U7 的 15 脚(6Q),即可使 PC 机的并行打印口用作通用的 I/O 口。

PC 机启动后, U7 清 0, 其 6Q 端为低电平, U4 的 \overline{OE} 端亦为低,并行打印口的工作与未作改进时完全一样。要作输入口使用时,只需用输出指令向控制输出锁存器 U7(其口地址为 37AH)的 6D 端写入“1”,使其 6Q 端变为高电平, U4 的 \overline{OE} 端亦为高,其输出端处于高阻状态,此时 25 芯插座 2—9 端的 8 位并行数据便可通过 U3

(其口地址为 378H)读入, PC 机并行打印口就成了通用的输入口。而并行打印口的输入缓冲器 U2 和输出锁存器 U7 仍可用于读入外设的状态信号和送出所需的控制信号。

三、结束语

上述方法已使用了数年,经与纸带机(包括穿孔机和光电输入机)等多种并行外设连接工作,证明该方法简单易行,工作稳定可靠。现介绍给大家,希望能有所助益。



多用手写字符输入板

林根远 北京耀辉电子通信公司(100054)

目前,家用微机一般均以键盘作为主要的输入工具。本文介绍一个多用手写字符输入板。它可以用在中华学习机或 PC 机中,也可以同单片机相连。该板的用途有:1. 可以用手指直接在输入板上书写所需的字符,字符的点阵坐标被输入计算机,经计算机计算、识别、转化后变为该输入字符的 ASCII 码,并存入内存中,

同时把该字符的标准形式显示在屏幕上。2. 本输入板可以作为小型的绘图板。它能绘制 16×16 点阵的图形或汉字。3. 本输入板还可作为光标控制器。用手指在板上, 上下左右移动, 计算机屏幕上的光标也随之移动。本用途的作用类似于 PC 机上的跟踪球。

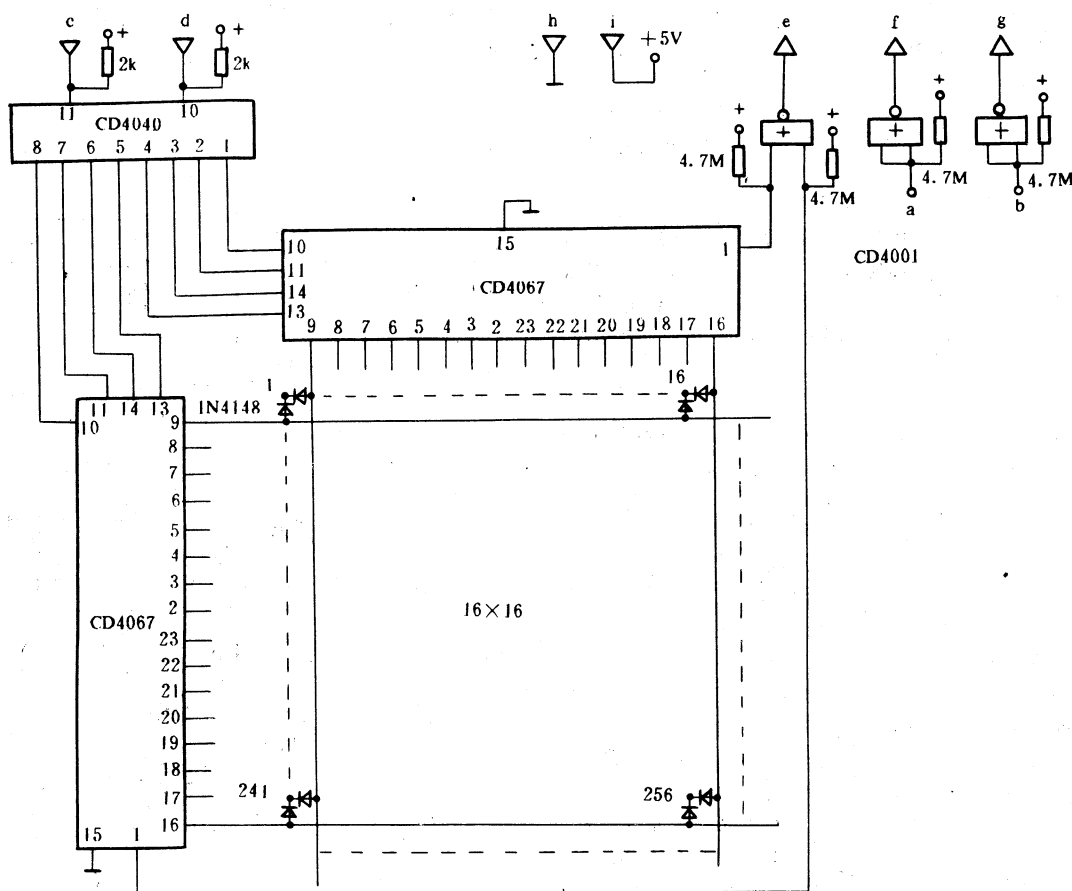


图1

电路原理:

输入板的电路见图1。图中2片 CD4067(单16通道模拟开关)组成 16×16 点阵,由1片 CD4040(12级二进制计数器)驱动其从第1点扫描到第256点。2片 CD4067的公共端接在或非门的2个输入端。当扫描到点阵的某一点时,如果该点没有被接入低电平,则2片 CD4067的公共端无输出,或非门的输入端被2个4.7M 电阻拉成高电平,其输出为低电平。如果该点被接入低电平,则2

片 CD4067 的公共端均输出低电平,或非门输出为高电平。低电平的接入方法用了人体导电的原理。操作者右手手腕上戴一个自制的导电环,导电环与电路的地相连。当右手手指触摸到点阵的某一点,而 CD4067 又扫描到该点时,操作者手腕到手指之间的电阻与 $4.7\text{M}\Omega$ 的电阻形成分压电路,由于人体的电阻远小于 $4.7\text{M}\Omega$,所以或非门的输入为低电平,输出为高电平。点阵中的二极管是起隔离作用的。图中另外两个或非门各组

成一个触摸钮,是当作功能键使用的。本电路共有4根输入线,3根输出线与计算机相连。输入线中2根是电源正负级,1根是清零线,1根是扫描时钟输入线。输出线中1根是点阵输出线,2根是功能键输出线。本输入板与计算机之间需要一个简易的输入输出卡,图2为与中华学习机相连时用的输入输出卡的电路。

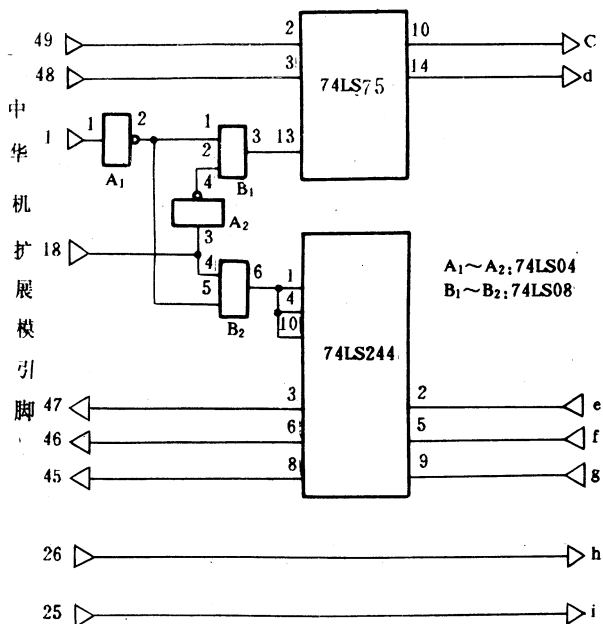


图2

制作与调试:

本电路的关键部分是触摸点阵。点阵中每个点是由2只1N4148二极管,1只螺帽为 $\phi 5$ 的平头铜螺钉和2只铜螺母组成。点阵用双面敷铜板,正面制成16条横

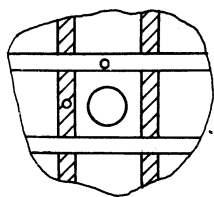


图3

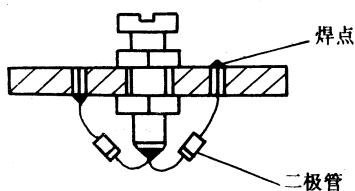


图4

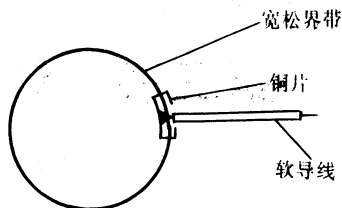


图5

线,背面制成16条纵线,线的宽度是1.2mm左右,线与线之间距离是7mm。在每1个纵横线相交点附近钻2个 $\phi 0.8$ 和1个 $\phi 3$ 的孔,孔的位置见图3。2个小孔是焊二极管用的,1个大孔是安装铜螺钉用的,具体安装方法见图4。每个螺钉的螺帽就是点阵中的一个触摸点。在整个点阵的右面,还要安装2只螺帽为 $\phi 10$ 的平头铜螺钉,螺钉分别与电路中的a,b相连,用作功能键的输入端。在电路的使用中还需要1个导电环,其制作方法见图5。

电路的调试方法是:按图6的接法,在电路中接上电源,2只开关和1只发光管。带上导电环,先按一下K1使电路清零,再用手手指触摸一遍所有的点,应该只有按在第1点时发光管才亮。按一下K2,再用手手指触摸一遍

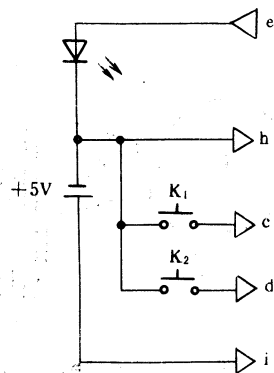


图6

所有的点,应该只有按在第2点时发光管才亮。重复上面的方法,直到测完所有的点。测完后就可以与计算机相连了。

程序的设计:

本电路使用的程序比较长,下面介绍一下程序的设计思路。程序分两部分,第一部分是扫描部分:先使C线输出1个脉冲,让电路清零,然后测试e线,把e线的状态记在内存中,使d线输出1个脉冲,再测试e线,把e线的状态记在内存中,再使d线输出一个脉冲,再测试e线,直到d线发出255个脉冲为止。这样就完成了1次扫描。

第二部分按该输入板的用途共分为3种程序。下面

(下转41页)

第三讲 针式打印机的基本工作原理

赵继文 陕西省计量测试研究所(710048)

一、针式打印基本原理

针式打印机在联机状态下,通过接口接收主机发送的打印控制命令,字符打印命令或图形打印命令,通过打印机的 CPU 处理后,从字库中找到与该字符或图形相对应的图象编码首列地址(正向打印时)或未列地址(反相打印时),然后按顺序一列一列地找出字符或图形的编码,送往打印头控制与驱动电路,激励打印头出针打印。其印字原理如图1所示。打印头是由纵向排成单列(如9针)或交叉排成双列(如24针)的打印针及相应的电磁线圈构成的。当电磁线圈通电激励后,相应的打印针就出针,通过击打色带,在打印纸上印出所需的字符(汉字)/图形。

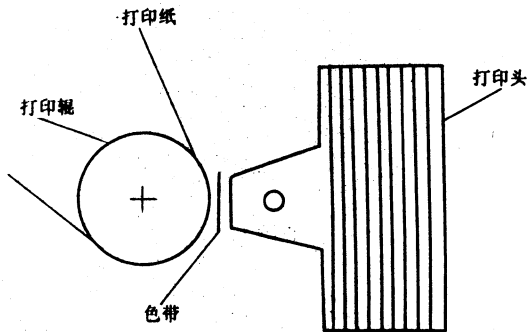


图1 印字原理图

字符(汉字)/图形的打印基本步骤如下:

1. 启动字车;
2. 检查打印头是否进入打印区;
3. 执行打印初始化;
4. 按照字符(汉字)或图形编码驱动打印头打印一列;
5. 产生列间距;
6. 产生字间距;
7. 一行打印完毕后,启动走纸电机,驱动打印辊和打印纸走纸一行;
8. 换行(若是单向打印,则回车),为打印下一行做好准备。

针式打印机基本上是依照上述8个步骤编制监控程序,由监控程序控制打印机完成打印过程。

图2给出9×9点阵字符图象和相应的字形编码。它

是以字符的图象首址作为基本地址,将图象编码依次取出,控制9根打印针出针操作,每打印一次,字车及打印头右移一列,打印一列字形后,列针数加1,修改图象编码地址,共分9次打印出一个完整的字符,每个字符或数字的字形编码占内存的9个字节。

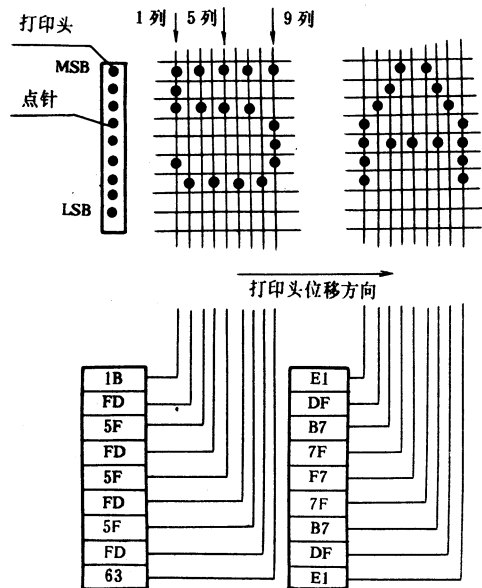


图2 9×9点阵字符图象和相应的字形编码

24×24点阵构成的汉字由72个字节组成,每列3个字节对应24根针,分24次打印完一个完整的汉字。

二、字符库

字符/图形的结构通常由 $m \times n$ 点阵组成,其中 m 为打印针数, n 为打印的列数。由字符/图形点阵的疏密程度决定印字质量。9针字符打印机采用 7×5 、 7×7 、 7×9 、 9×7 和 9×9 等多种点阵格式;24针汉字打印机采用 18×18 、 22×15 和 24×12 等加密型点阵格式。

为了适应不同的打印要求,同一种类字符/图形都有三种点阵结构方式:即高速打印字符,如有 18×12 或 22×9 等格式;高密打印字符,如有 18×18 或 22×15 或 24×12 等格式;上标/下标打印字符,如有 12×8 或 14×9 等格式。

由 $m \times n$ 点阵组成的字符/图形,都用二进制数表示,有点者为“1”,无点者为“0”。因此每个字符/图形都对应一个二进制码(为了方便,一般写为16进制数)。例如一个 18×18 点阵字符/图形,每列由18个二进制码组成,由上而下分为三个字节,每个字符由18列组成,共 $3 \times 18 = 54$ 个字节,每个字节占用内存中一个地址单元。所有的字符/图形编码,都按照〈信息处理交换用七位编码字符集〉中的规定存放在 EPROM 芯片中,分配好地址,以备打印时取用,这就是字符库。

三、汉字库

目前,我国汉字点阵结构分为四类: 16×16 点阵组成简易型汉字; 24×24 点阵组成普及型; 32×32 点阵组成提高型; 48×48 点阵组成精密型。其中 24×24 点阵是目前 24 针汉字打印机普遍使用的格式。用576个点的不同组合,构成6763个汉字的一级和二级汉字库(GB2312-80)或7454个汉字的汉字库(GB5007-85)。通常把 24×24 点阵图象称为全角汉字, 24×12 点阵图象称为半角汉字。每个 24×24 点阵图象组成的汉字占用72个地址单元,编好地址存储在打印机内置的 EPROM 芯片中,这就是打印机的汉字库。

图3给出“正”字的 24×24 点阵图象,以作示例。通常不管汉字的繁简和笔划多少,点阵图象只占22列,左右两边各留一列作为字间距。

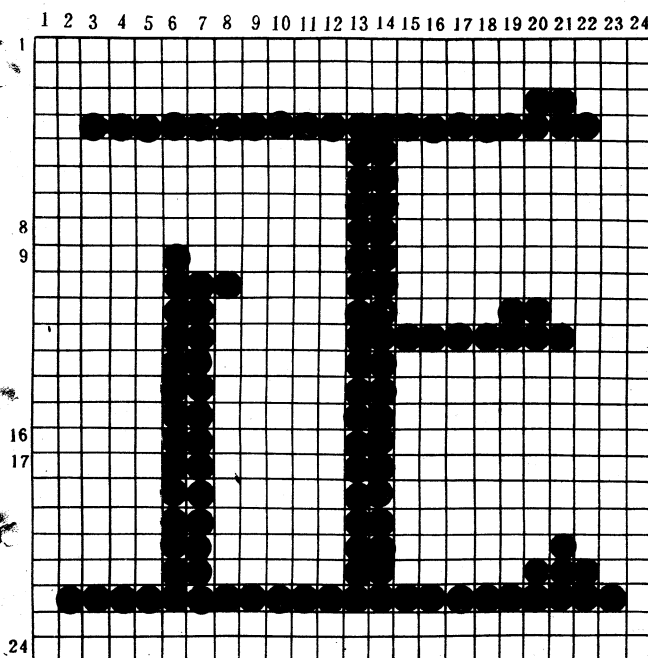


图3 24×24 点阵汉字示例

第四讲 针式打印机 电源电路分析

一、电源种类及用途

针式打印机的电源主要有两组:一组是+5V,主要供逻辑电路、操作面板和走纸电机锁定用。对于采用“高压驱动、低压锁定”字车电机的打印机来说,+5V也供给字车电机,作为锁定电源。另一组是高压电源,用于驱动打印头、字车电机和走纸电机。由于各种型号的打印机所采用的步进电机不同,驱动电路设计也有所不同,故高压电源的规格不统一。现将常见针式打印机高压电源列于下表中。除上述两组电源外,有的针式打印机还有+12V或±12V电源,主要用于RS-232C串行接口板等。

常见针式打印机高压电源

+16V	TH-3070 紫金3070
+24V	LQ-1500 LX-800 TX-800 MX-80/100 RX-80/100 FX-80/100 FX-80+/100+
+35V	CR-3240 CR-3200 LQ-1600K LQ-2500 LQ-800/1000 M-1724 OKI-8320C OKI-5320
+36V	M-2024 M-2024L S-1570(CYD-902)
+31V~+40V +31V~+38V	AR-2463(供打印头) AR-2463(供电机)
+30V~+45V +32V~+38V	AR-3240(供打印头) AR-3240(供电机)
+44V	NM-9400

二、电源电路分析

针式打印机电源一般包括交流滤波电路、电源变压器和直流稳压电路三部分。但近期推出的针式打印机大多采用体积小、效率高的无电源变压器型开关式电源。例如CR-3240、CR-3200、LQ-800/1000、LQ-2500、LQ-1500和LQ-1600K等打印机都采用这种电源,它直接把经过交流滤波器之后的输入电压供给开关式直流压电源。

1. 交流滤波电路

交流滤波电路用于阻止交流电源的噪声进入打印机,同时抑制打印机本身产生的噪声馈入交流电源线。图4给出 AR-3240、AR-2463、TH-3070(紫金3070)、RX-80/100、FX-80/100和 FX-80⁺/100⁺打

印机的交流滤波电路图。图5给出 CR-3240、LQ-800/1000、LQ-2500、LQ-1500、LQ-1600K 和 NM-9400打印机的交流滤波电路图。该电路和电源保险丝 F 都安装在一个电路板上。

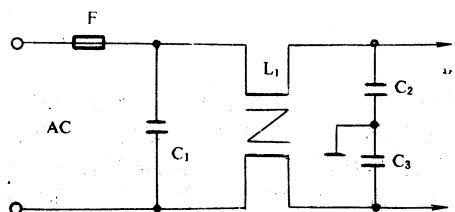


图4 交流滤波电路(1)

2. 电源变压器

电源变压器用于将消除噪声后的输入电压变换为低压,供给后级直流稳压电源。有些打印机的电源变压器,如 M-1724、M-2024、AR-3240和 AR-2463等打印机,在电源变压器的初级绕组中串联了一只热保护器。当绕组中流入电流过大时,热保护器就会立即熔断,以防止绕组因温度升高而烧坏变压器。

3. 直流稳压电源

直流稳压电源分为整流滤波电路和稳压电路两部分。稳压电路主要有三端集成稳压器型、开关型集成稳压电路和无电源变压器的开关型稳压电源等几种类

图5 交流滤波电路(2)

型。前两种电路比较简单。这里仅以 LQ-1600K 打印机的电源为例对无电源变压器型稳压电源的主要部分进行分析。

LQ-1600K 打印机电源方框图如图6所示。提供 +5V、±12V 和 +35V 四组电源。它采用全波整流和电容滤波电路,为了防止开机瞬间,滤波电容器上流过的浪涌电流,在该电容的回路中串联了两只电阻,且在电阻上并联了可控硅。在开机瞬间由两只电阻起抑制作用,待电源工作正常后,可控硅触发导通,将电阻短路不再抑制电流。图7给出主开关电路图。打印机加电后启动电流 I_s 流向晶体管 Q1 的基极,使 Q1 导通,集

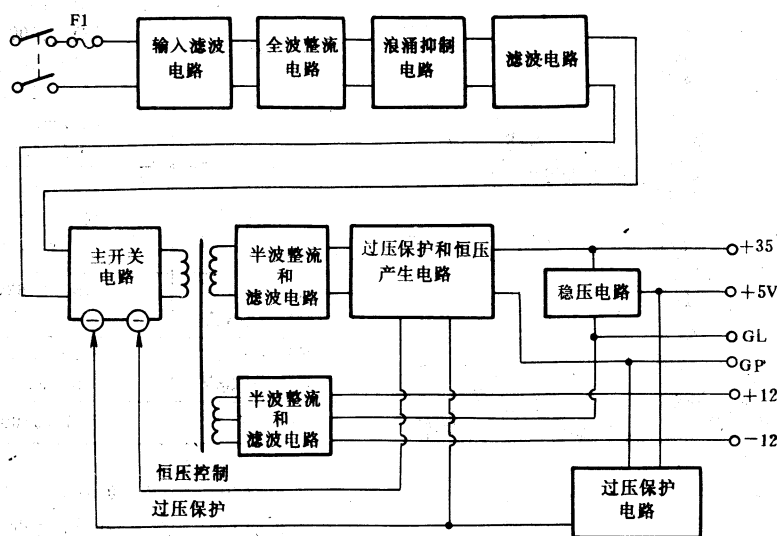


图6 LQ-1600K 电源方框图

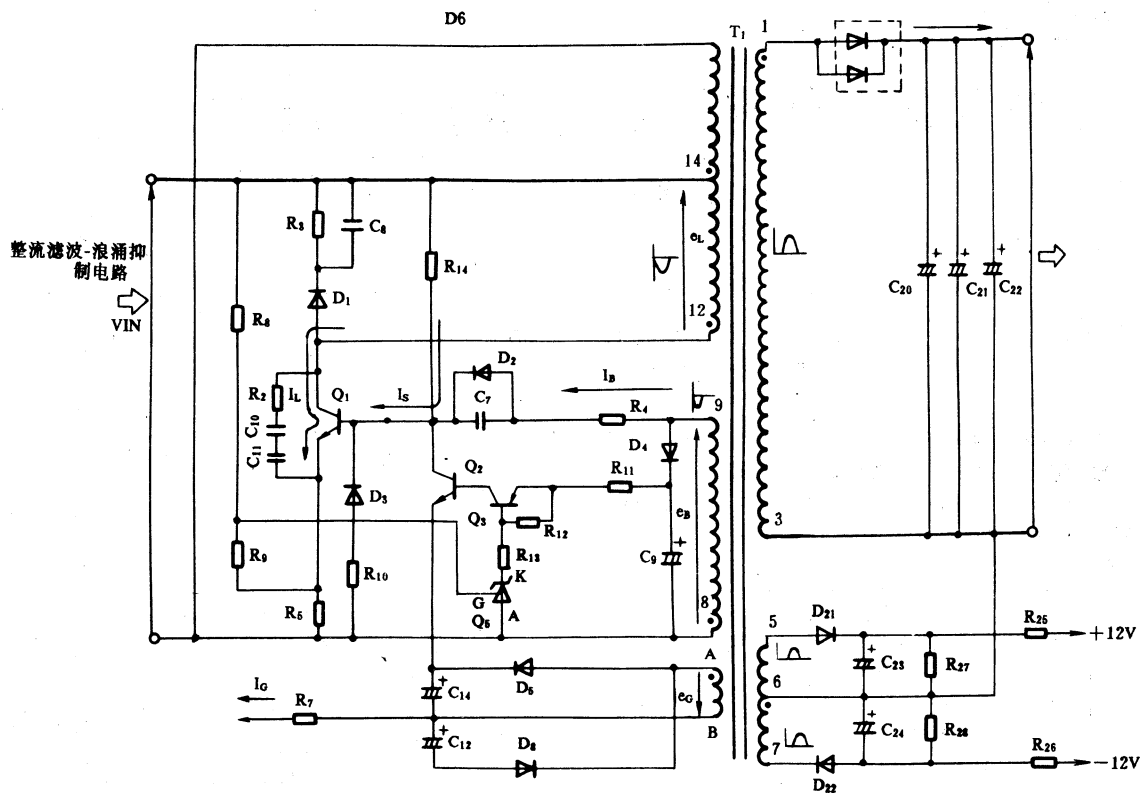


图7 LQ-1600K 电源主开关电路

电极电流 I_c 即为变压器绕组 T/14-12 中的电流, 于是在 T/14-12 上产生了 14 端为正、12 端为负的感应电势 e_L , 绕组 T/9-8 上也感应了 9 端为正、8 端为负的电势 e_B , 这就抬高了 Q1 基极电位, 使 Q1 进一步导通, 产生下列正反馈工作过程:

产生 $e_B \rightarrow Q1$ 的 $I_b \uparrow \rightarrow Q1$ 的 $I_c \uparrow \rightarrow e_L \uparrow \rightarrow e_B \uparrow$

当 Q1 的 I_c 趋于饱和时, I_c 增加率变小, T/14-12 上即感应出方向相反的 $-e_L$, T/9-8 上也感应出 $-e_B$ 。 $-e_B$ 通过二极管 D_3 , 使 Q1 的 b-e 之间受到反偏电压的作用, 使 Q1 趋于截止, 产生下列负反馈工作过程:

产生 $-e_B \rightarrow Q1$ 的 $I_c \downarrow \rightarrow -e_L \uparrow \rightarrow e_B \uparrow \rightarrow I_b \downarrow$

最后导致 Q1 迅速截止。Q1 截止后, Q1 又在 I_s 的启动下产生上述工作过程。如此周而复始的导通、截止, 这就是主开关的工作过程。主开关电路工作以后, 绕组 T/1-3 上产生感应电压, 经半波整流滤波后输出 +35V 电压。+35V 电压分支提供给混合型稳压器 STR20005, 由它稳压后输出 +5V 电压。±12V 电压由绕组 T/5-6-7 上感应的电压经半波整流滤波后输出。该电源具有过压和过流保护功能。

(上接30页)

与 DP-851 不同, 键盘和显示器与主板是可以脱离的。作开发器使用时, 将键盘组件接在 PA 口和 PC 口的低四位上; 将显示器组件接在 PA 口和 PB 口上, 即实现 DP-851 的全部功能。若摘掉主板上 DP-851 的监控芯片, 换上自己开发的程序, 摘掉键盘和显示器, 搬动几个开关。DP-888 就成为一台 8KRAM, 8KROM, 30 余条 I/O 口线的计算机。这对用户是极为有用的。

8155 的 \overline{CE} 接在 74LS138 的 Y_3 上, 其地址为:

6000~60FF: 8155 片内 RAM 256 字节

6100 : 8155C/S 寄存器

6101 : 8155PA 口

6102 : 8155PB 口

6103 : 8155PC 口

6104 : 计数器低8位

6105 : 计数器高8位

第六章 背景画面绘制的编程技巧(续)

于 春 北京裕兴机械电子研究所(100035)

2. 优化查表法

《魂斗罗》游戏已为广大朋友所熟悉、所喜爱。该游戏的编程中有许多优化的方法可供我们学习和借鉴。以下两例供参考。

①武器选项画面的绘制

修改版的《魂斗罗》游戏开始有一个武器选项画面(见图6-4)。在该画面,游戏者可通过选择键(SELECT)在四种武器中任选其一。该画面的绘制程序见No. 6-11。

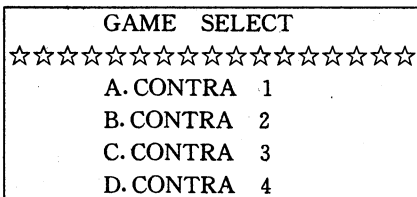


图6-4 《魂斗罗》武器选项画面

No. 6-11 《魂斗罗》武器选项画面的绘制程序

```

F80D A94E LDA # $4 ;置绘图数据区首
F80F 8500 STA $00 址于$00、$01单元
F811 A9F8 LDA # $F8
F813 8501 STA $01
F815 A000 LDY # $00
F817 B100 LDA ($00),Y
F819 8D0620 STA $2006 ;置PPU地址高位
F81C C8 INY
F81D B100 LDA ($00),Y
F81F 8D0620 STA $2006 ;置PPU地址低位
F822 C8 INY
F823 B100 LDA ($00),Y
F825 AA TAX
F826 3011 BMI $F839 ;送数指南大于
F828 C8 INY 7FH转$F839
F829 B100 LDA ($00),Y ;送数指南小于
F82B 8D0720 STA $2007 80H的送数处理,
F82E CA DEX ;将其后的N个数
F82F D0F7 BNE $F828 送入PPU
F831 C8 INY
F832 B100 LDA ($00),Y
F834 C980 CMP # $80 ;不是80H则改变
F836 D0DF BNE $F817 PPU地址,继续送数
F838 60 RTS ;如是80H则结束送数
F839 8A TXA
F83A 297F AND # $7F
    
```

```

F83C AA TAX ;以该数的D6~D0
F83D C8 INY 位为计数器,将其后
F83E B100 LDA ($00),Y 的一数连送N次
F840 8D0720 STA $2007
F843 CA DEX
F844 D0FA BNE $F840
F846 F0E9 BEQ $F831
    
```

数据区如下:

```

F84E 3F 00
PPU 配
F850 08 0F 20 20 20 0F 27 27 27 3F 10 04 0F 05 0F 17
色,送$3F00~$3F07。送PPU$3F10~$3F13
F860 23 D0 88 50 20 E4 17 20 20 20 20 20 20 20 20 20
背景零页配色,$23D0~$23D7
F870 20 20 20 20 20 20 20 20 20 20 20 20 21 2A
F880 0B 47 41 4D 45 00 53 45 4C 45 43 54 21 66 93 3F
    
```

G A M E S E L E C T

```

F890 21 AA 0B 41 40 00 43 4F 4E 54 52 41 00 31 21 EA
A. C O N T R A 1
F8A0 0B 42 40 00 43 4F 4E 54 52 41 00 32 22 2A 0B 43
B. C O N T R A 2 C
F8B0 40 00 43 4F 4E 54 52 41 00 33 22 6A 0B 44 40 00
C O N T R A 3 D.
F8C0 43 4F 4E 54 52 41 00 34 00 00
C O N T R A 4
    
```

图6-4的数据区仅使用了124个单元就完成了PPU配色、背景页配色和画面显示,虽然四行都是相同的内容(CONTRA),但数据区中已留出了相应的空间,即使送显其它内容,数据区也不会增加。

程序No. 6-11的结构比较简单,读者可自己完成程序的转换,不多赘述。

②标题画面的绘制

我们知道,《魂斗罗》的标题画面是由画写的“CONTRA”和两个主人翁的上身组成的。画面既简单抽象又富有力的感召,是正义和美的有机结合与升华。《魂斗罗》仅凭一幅标题画面就勾起了人们的无限遐思与向往,这怎能不令人钦佩和惊叹呢!现在让我们看看这一标题画面是如何绘制的。

标题画面的绘制程序在\$C9CF~\$CA60,其数据区在第二体的\$9097~\$9251,现抄录如下,见程序No. 6-12。

No. 6-12《魂斗罗》标题画面绘制程序

C9CF	A001	LDY	# \$01	该子程序调用前 已向 \$00、\$01单 元置入数据区	CA10	A901	LDA	# \$01	
					CA12	18	CLC		
C9D1	B100	LDA	(\$00),Y	首址 \$9097	CA13	6808	ADC	\$02	
C9D3	8D0620	STA	\$2006	置 PPU 地址高位	CA15	A800	LDX	# \$00	
C9D6	88	DEY			CA17	2098CA	JSR	\$C892	调整取数地址
C9D7	B100	LDA	(\$00),Y		CA1A	4CE8C9	JMP	\$C9E8	
C9D9	8D0620	STA	\$2006	置 PPU 地址低位	CA1D	A001	LDY	# \$01	当取数指南 > 7FH 时,把后面的一个数据 连送 N 次的处理。
C9DC	A902	LDA	# \$02						
C9DE	A604	LDX	\$04		CA1F	8502	TA	\$02	
C9E0	1001	BPL	\$C9E3		CA21	B100	LDA	(\$00),Y	
C9E2	0A	ASL		若 \$04 > 7F 则 A =4	CA23	A402	LDY	\$02	
					CA25	A604	LDX	\$04	
C9E3	A200	LDX	# \$00		CA27	1003	BPL	\$CA2C	
C9E5	2092C8	JSR	\$C892	调整取数地址。	CA29	2036CA	JSR	\$CA36	
C9E8	A000	LDY	# \$00		CA2C	8D0720	STA	\$2007	
C9EA	B100	LDA	(\$00),Y		CA2F	88	DEY		
C9EC	C9FF	CMP	# \$FF	若 A=FF 则结束 送数	CA30	D0FA	BNE	\$CA2C	
C9EE	F06B	BEQ	\$CA5B		CA32	A902	LDA	# \$02	
C9F0	C97F	CMP	# \$7F	A=7F 改变 PPU 地址	CA34	D0DF	BNE	\$CA15	一次送数结束后,调整 取数地址
									以下是延时子程序
C9F2	F05D	BEQ	\$CA51		CA36	8503	STA	\$03	
C9F4	A8	TAY			CA38	0603	ASL	\$03	
C9F5	1026	BPL	\$CA1D		CA3A	6A	ROR		
C9F7	297F	AND	# \$7F		CA3B	0603	ASL	\$03	
C9F9	8502	STA	\$02	若取数指南小于 7FH 则以 \$02 为 计	CA3D	6A	ROR		
C9FB	A001	LDY	# \$01	数器,送后面的 N 个数。	CA3E	0603	ASL	\$03	
C9FD	B100	LDA	(\$00),Y		CA40	6A	ROR		
					CA41	0603	ASL	\$03	
C9FF	A604	LDX	\$04		CA43	6A	ROR		
CA01	1003	BPL	\$CA06		CA44	0603	ASL	\$03	
CA03	2036CA	JSR	\$CA36	若 \$04 > 7F 则延时送 数	CA46	6A	ROR		
					CA47	0603	ASL	\$03	
CA06	8D0720	STA	\$2007		CA49	6A	ROR		
CA09	C402	CPY	\$02	达到送数数量则转	CA4A	0603	ASL	\$03	
CA0B	F003	BEQ	\$CA10	\$CA10 调整地址	CA4C	6A	ROR		
CA0D	C8	INY			CA4D	0603	ASL	\$03	
CA0E	D0ED	BNE	\$C9FD	未达到送数数量则转 \$C9FD 继续送数	CA4F	6A	ROR		
					CA50	60	RTS		

(上接35页)

只介绍一下识别字符程序的思路。先判断输入字符的笔画数,即输入时手指离开输入板的次数减1。然后把字符的每个笔画,通过计算变为8个方向。8个方向是"向上"、"向下"、"向左"、"向右"、"左上"、"左下"、"右上"、"右下"。举2个例子,1."A"字符是3画,第1画是"左下",第2画是"右下"第3画是"向右"。2."B"字符是2画,第1画是"向下"第2画是"右下""左下""右下""

左下"。在把笔画变为方向后就可以与字符库中的各字符数据相比较了。例如:如果输入的字符是3画,第1画是"右下",第2画是"左下",第3画是"向下"。把它与字库中字符比较,发现与"Y"字符数据相同,就说明输入的就是"Y"字符。

以上只是我编程的思路,大家也可以按其它更好的思路编写程序。

CCED 编辑软件 中的灵活运用

任富垠 北京(101601)

在一般的编辑软件中,只提供了一种字块的操作功能,而 CCED 提供了两种字块的操作功能,同时这两种字块又可以互相包容,即一个行块中同时又包括一个矩形块,而矩形块又可以吧整行包括进去。用户可以根据自己的需要方便地进行行间插入(用行块转移或复制的办法)和列间插入(用矩形块转移或复制的办法)。如果在一个行块中,矩形块的宽度定义得合适,还可以一块多用,减少操作次数,提高输入速度。例如,下面的一首诗:

白日依山尽,黄河入海流,欲穷千里目,更上一层楼。

首先要求把整行复制到后文中去,再把中间的两句复制到后文的某行中间,方法如下:把光标移入“黄”处,按 F8,再把光标移到“目”处按 F8,然后把光标移到插入行的上行,按 F7;再把光标移到某行中要插入两句诗的字符后,按 Ctrl+Z 即可,不必重新定义字块。

CCED 不仅有很强的文字编辑功能,而且可以对数字表格直接进行数据统计和公式计算。按照常规,如果进行列求和或者列间运算,应先将表格用行块定义好计算范围;如果进行行间计算,则要用矩形块定义计算范围。这样,要对数字表格同时进行行、列间的运算,不仅要反复建立行块、矩形块,而且什么时候需要建立行块,什么时候需要建立矩形块,往往容易搞混,特别是对于初学者。笔者根据行块矩形块互相包容的原理摸索出一套简单的办法,只要把数字表格的数字区定义成矩形块,即可方便的进行数字表格的行、列间的计算,具体方法如下:

1. 把光标移到数字区的第一行、第一列(不包括表格线),按 F8,再把光标移到数区的最后一行、最后一列(不包括表格的边框线)按 F8,即建好了数字区的矩形块。

2. 用 Ctrl+S 进行一列数据求和;用 Ctrl+C 进行列间和行间的各种运算。

请看下边的实例:

	一车间	二车间	三车间	季总产量	
一季度量	256.8	456.0	680.5	1393.3	~ @L1
二季度量	297.3	546.4	673.6	1517.3	~ @L2
三季度量	355.0	524.2	730.6	1609.8	~ @L3
四季度量	265.4	435.01	724.4	1424.8	~ @L4
季平均产量	293.6	490.4	702.3	1486.3	@L5=(
全年产量	1174.5	1961.6	2809.1	5945.2	L1+L2
					+L3+
					L4)/4,1

~ @C5=C2+C3+C4,1

~ @C3=C3,1

第二、三、四列的全年产量就是用列求和所得;第五列和第五行的数据是用上边相对应的公式,再按 Ctrl+C 所得,并对第三列数据进行了靠右放置的操作(可和第二列、第四列比较)。由此可以看出:这样做不仅简单易学,而且方便实用。如果一个文件中不只一个数据表格需要计算,只要注意行号别重即可。(笔者用的是 CCED 3.3 版本)。

DM(V3.01)使用之我见

袁志新 章丘市成人教育中心250200

众所周知,DM 是一种比较优秀的硬盘管理软件,成为专治硬盘软故障“不治之症”的有力工具。它有如下几大功能:硬盘初始化(Initialzition);缺陷磁道管理(Defect-list management);检验磁盘体(Verify surface);硬盘分区(Partitioning);选择驱动器号(Select Drive);系统配置(Configuration)。目前,DM 软件有几种版本,但本人在使用 3.01 版 DM 时,发现有两个问题不敢与有的杂志上介绍的 DM 使用苟同,在此列出,供探讨。

1. 对硬盘分区时,系统引导扇区不一定非在 0—10 磁道,可以在其它磁道。

本人曾遇到这样一台机器,硬盘 0 磁道遭到破坏,经软盘启动后,运行 DM(A>提示符下键入 DM/M),进入主菜单,选 I 后再键入 D,进入缺陷磁道管理,发现 0 磁道及后面许多不连续的磁道都遭到破坏,到了 200 柱面后才正常。返回主菜单,进入分区菜单,抛开 0 磁道,从 10 磁道开始为 DOS 分区,再为 DOS 分区格式化,最后将一个 WPS 汉字字处理软件拷入硬盘,启动 WPS 时,发现汉字显示及打印输出均不正常。我又重新运行 DM,分区时,把起始磁道定在 200 柱面,后拷入 WPS,再运行,没想到奇迹出现,虽然机器启动有所缓慢,但汉字显示及打印输出均已正常。

2. 虽然 DM(V3.01)硬盘管理范围只能到 33M,但可以借助 Fdisk.com 管理更大容量的硬盘。

一般情况,机器出厂时已做过初始化及分区等工作。对于大容量硬盘(如 120M),可能由于误操作,清除了原有的分区表,而用 DM 硬盘分区时,DOS 分区只能在 0—770 柱面之间,即只能管理到 33M,剩余部分无法管理。怎样才能利用全部的磁盘空间?方法是:利用 DM,进入分区菜单,先打入 N 清除分区表(若存在),然后返回主菜单,初始化硬盘,这时可将全部硬盘初始化。否则,只能初始化原有分区范围内的空间。这时不再利用 DM 的分区菜单,而退出 DM 返回到 DOS 状态下,然后再利用 Fdisk.com 进行分区,激活活动分区,再进行 Format,这样就达到了使用全部硬盘的目的。

Microsoft Windows 3.1

应用基础——Windows 进阶

徐鹏力 赵海舰

前面我们介绍了 Windows 3.1 的基本使用方法, 下面将讨论 Windows 应用程序的集成与 Windows 优化策略。

一、Windows 应用程序的集成

Windows 3.1 提供了三种服务机制: Clipboard(剪贴板)、DDE(动态数据交换)及 OLE(对象链接与嵌入)来实现应用程序间的信息共享。剪贴板作为一种被动的传送装置可将用户指定的数据从一个地方搬到另一个地方。实际操作中, 常先用 Copy(拷贝)或者 Cut(剪切)命令将数据(文本、图形、图像等)送到剪贴板, 再用 Paste(贴粘)命令把它放到所需的地方。

由于所有 Windows 应用程序使用剪贴板的过程实质都是一致的, 下面仅以 Write 及 Paintbrush 应用程序举例说明剪贴板的操作过程。

1° 在 Paintbrush 应用程序中选择欲 Cut 或 Copy 的信息(图1);

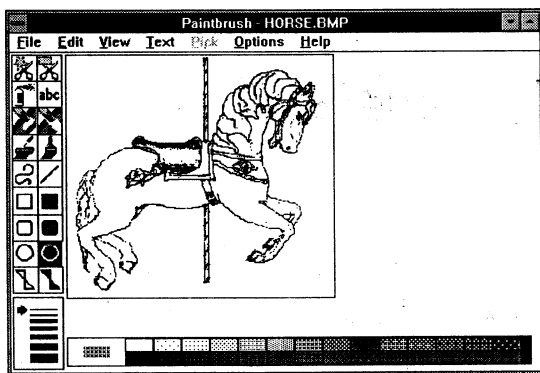


图1 Paintbrush 应用程序

2° 从 Paintbrush Edit 菜单中选择 Cut(剪切)或 Copy(拷贝)命令, 选中的信息将剪切或拷贝到 Clipboard;
3° 打开 Write 应用程序, 将光标定位至欲插入图形的位置(图2);

4° 从 Write Edit 菜单中选择 Paste(粘贴)命令。

此时用户所见的将是一篇图文并茂的文章(图3), 这就是剪贴板的一个简单应用。

但是, 通过剪贴板传递的信息完全是静态的, 有时当我们在 Windows 环境中, 例如追踪股票价格涨落信息时需要不断从股票交易所得到实时动态的数据变化显示, Windows 中的这种技术为 DDE(动态数据交换)。对于大多数应用程序, 建立 DDE 最简单的方法是: 将信息从应用程序 A 拷贝至 Clipboard, 激活应用

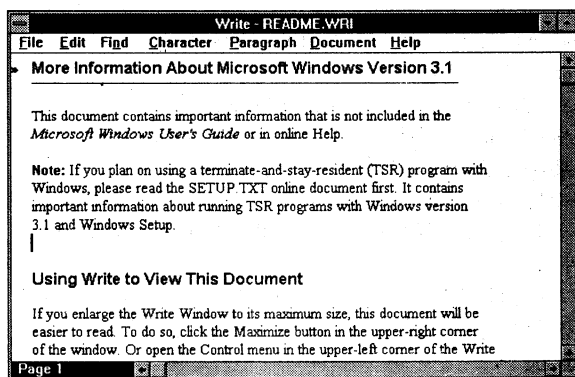


图2 Write 应用程序

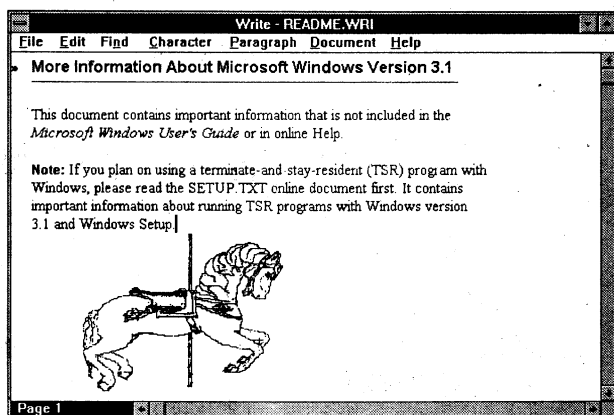


图3 Clipboard 应用实例

程序 B 并选择信息欲放置的地方, 使用 Paste Link 命令。以下我们仍以以上例阐述 DDE 技术。

1° 在 Paintbrush 应用程序中将欲 Copy 的信息存盘;

2° 在 Paintbrush 应用程序中选择欲 Copy 的信息;

3° 从 Paintbrush Edit 菜单中选择 Copy 命令;

4° 激活 Write 应用程序, 光标定位至信息欲放置的位置;

5° 从 Write Edit 菜单中选择 Paste Link 命令。

尽管用户此时所见与上例并无差异, 但当我们编辑 Write 中的图画时, 两者差异显而易见。双击图画进入 Paintbrush 执行修改任务后, DDE 技术使图画的改变立即反映在 Write 中, 而剪贴板技术则需在 Paintbrush File 中选择 Update 命令后才可反映在 Write 中。另外, DDE 与剪贴板技术在使用方法上也有

一些差异,在 DDE 技术中一般都需先将数据存盘后再用 Copy 命令实现应用程序间的数据链接。

对象链接与嵌入 (OLE) 实际是上述剪贴板及 DDE 技术的改进,操作中我们经常混合使用链接与嵌入,限于篇幅,在此不再重复介绍。

二、Windows 优化

Windows 优化也就是提高系统速度和容量的过程。以下是优化 Windows 的一些主要方法:

优化的基本方法:

- 在条件允许的情况下,给计算机安装尽可能多的扩展 (extended) 内存。
- 将 Windows 装入以前所占用的内存减至最小。
- 保证 Windows 环境下有足够的内存空间。
- 要经常清理硬盘上的空间碎片。
- 使用 Smartdrive (高速缓冲工具)。
- 若是在 386 增强模式下,应使用一个永久性的交换文件。
- 系统运行在标准模式下时,应将所使用的 MS-DOS 应用程序交换文件放置到最快速的和空间最富余的磁盘上。
- 运行 Windows 理论上讲只需 2M 内存 (640K 常规内存,其余为扩展内存),但是 2M 内存在使用时经常有不尽人意的情况发生,想要有效地运行 Windows,实际上至少需要 4M 内存。

在 Windows 启动以前,去掉所有不必要的内存占

用,把内存尽量留给 Windows。这里包括不要安装任何不是绝对需要的设备驱动程序或内存驻留程序 (内存驻留程序通常也称作 TSR 或“弹出式程序”)。设备驱动程序和内存驻留程序通常是由 CONFIG.SYS 或 AUTOEXEC.BAT 文件中的命令放入内存的。因此只需读一下 CONFIG.SYS 和 AUTOEXEC.BAT 文件中的内容,即可发现在系统中有那些项使用了内存。CONFIG.SYS 和 AUTOEXEC.BAT 文件里哪些文件是必需的可查阅 MS-DOS 手册。

优化的另一种方法是保持有足够的磁盘空间,可有三种实现方法。

第一,采用文件压缩实用程序;

第二,删掉多余的文件;

第三,提供更多的磁盘空间。

为了提供更多的内存可将 Windows 应用程序的临时文件保存到 RAM 盘上。首先要创建一个 RAM 盘,其次要修改 MS-DOS 环境里的 TEMP 变量。

以上我们初步探讨了有关 Windows 优化的一些问题,有关的详细内容参考 Windows 使用手册。

前面几讲,我们对 Windows 3.1 使用作了基本介绍。应当指出,目前计算机界 Windows 热潮日益高涨,诸如 Word for Windows 字处理,Excel for Windows 电子报表,Powerpoint 图表创作等 Windows 应用软件已引起广大兴趣,而学习这些应用软件的基础正是 Windows 3.1。

PC Tools 文件服务功能(续)

王路敬

三、磁盘服务功能

(一) 磁盘服务功能主菜单

在进入 PC Tools 后按 F3 键,可直接进入磁盘服务功能,也可以在文件服务功能主菜单下键入 F3 转换到磁盘服务功能。

进入磁盘服务功能,屏幕将显示磁盘及特殊功能主菜单;在下面的提示行中显示:

```
DISK:SERVICES:Copy cOmpare Find Rename Veri-
fy view/Edit Map Locate iNitalize
SPECIAL SERVICES:Directory maint Undelete sys-
tem Info Park Help
F3=file srv F10=F3+chg drv ESC=exit PC Tools
```

(二) 磁盘及特殊服务功能的实现

1. 有关功能键的使用

F3:转换到文件服务功能主菜单

F10:转换到文件服务功能主菜单并改变指定的

驱动器

ESC:退出 PC Tools

2. 磁盘服务功能的实现

(1) Copy (磁盘拷贝)

Copy 功能可以进行整个软盘的复制,拷贝的过程是先从源盘读入信息,然后写入目标盘。在复制过程中包括对目标盘的格式化。操作时,用户可在屏幕上看到有关操作,并通过此操作看出源盘的类型是 360K 的还是 1.2M 的,单面软盘还是双面软盘。

[例]将 A 盘中的文件复制到 B 盘

操作步骤如下:

①在磁盘服务功能主菜单下键入“C”,屏幕显示如下窗口信息:

```
Enter SOURCE Drive ID -[A]
```

```
Valid letters are A thru F.
```

```
Press ESC to return
```

②在出现第一个窗口后(此时 PC Tools 让用户输入源盘盘符,默认盘符 C)键入源盘驱动器号“A”并回车,屏幕又弹出第二个窗口,显示如下:

```
Enter TARGET Drive ID- [A]
Valid letters are A thru F.
Press ESC to return
```

③弹出第二个窗口,用户输入目标盘盘符,此时键入“B”并回车。

④用户根据提示信息,将源盘插入 A 驱动器,目标盘插入 B 驱动器,在准备好之后按任意键开始工作。

屏幕显示如下信息:

```

          1          2          3          3
Track 0123456789012345678901234567890123456789
Side 0 RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
Side 1 RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
```

Track 后是磁道号,它因源盘的格式不同而异:360KB 软盘40道,1.2MB、720MB、1.44MB 软盘80道。当读入某一道时相应磁道号下显示“R”。插入目标盘后,则先对其进行格式化,这时相应磁道号下用“F”复盖“R”,接着进行写操作,即用“W”复盖“F”。对每一磁道拷贝成功后,显示一个“.”,复制操作时,如出现错误,则相应磁道号下显示“E”并闪烁。拷贝完成后返回磁盘功能主菜单。

注意:①磁盘服务功能的 COPY 操作只能对软盘进行,若对硬盘操作 PC Tools 将拒绝执行,并显示无效的驱动器标识。

②对于只有一个软盘驱动器的微机,源盘符与目标盘盘符相同,在操作过程中会随时提醒用户换盘。

③尽管拷贝过程中提示“只能拷贝由标准的 DOS 格式化的磁盘,C 功能还可拷贝某些源盘是加密的或由其他操作系统格式化的,但不能拷贝激光加密盘和某些游戏盘。”

(2)Compare(磁盘比较)

用来比较两张软盘文件的内容。

键入“O”后,PC Tools 先后弹出两个窗口,让用户分别输入第一张盘所在驱动器的盘符及第二张盘所在盘符,回车后即开始比较。每一次读盘时显示“R”,比较时用“C”代替“R”,比较成功后以“.”表示,在执行过程中,若发现两张盘内容不同,则显示出两张盘的逻辑扇区号和偏移量及不同处的信息。

值得指出的二点:

①磁盘比较功能不能对硬盘进行操作。

②不管 Copy 还是 Compare 执行时在相应的磁道若出现闪烁的“E”,并不意味着整个磁道坏,因为 PC

Tools 标记磁盘错的最小单位是扇区。

(3)Find(查找)

磁盘服务的 Find 功能与文件服务的 Find 功能相似,只是磁盘服务的 Find 可以在整张盘上查找字符串或十六进制数串,提示信息使用户了解更详细情况。

当键入“F”后,屏幕弹出一窗口,供用户输入指定的盘符。输入后屏幕显示出两个空白行让用户输入要查找的字符串或16进制数串。操作方法与文件服务中的 Find 功能相同。

与文件服务功能不同的是,磁盘服务功能还将提示出被查找内容所在的相对扇区号及偏移量。

(4)Rename(修改卷标)

用于修改磁盘的卷标。

R 功能操作极为简单,键入 R 之后,屏幕显示如下信息:

```
Drive A
Current volume label=None
Enter the new volume label[ ]
Press ESC to Exit
```

其含义是:

驱动器 A

当前卷标=无

输入新的驱动器卷标[]

按 ESC 退出

[例]将 A 盘的卷标改为 PCTOOLS。

操作步骤如下:

①键入“R”并回车

②屏幕弹出一窗口,让用户输入要修改的盘的盘符。这时键入“A”并回车

③根据屏幕有关提示,输入新卷标 PCTOOLS 并回车。

注意:卷标最多只能输入11个字符,如果直接回车将不改变原来的卷标。

(5)Verify(磁盘校验)

用来验证磁盘的可读性。

该功能与文件服务中 Verify 的功能相似,不同的是它可验证整个磁盘诸如卷标、目录、文件等可读信息,而不管磁盘中是否有文件。

操作过程如下:

键入 V 并选择驱动器 A 后,屏幕显示如下信息:

```
Drive A is about to be verified,
```

```
Press any key to continue
```

```
Press ESC to Exit
```

其含义是:

驱动器 A 将被校验

按任意键继续

按 ESC 退出

按任意键则开始校验,显示信息的含义是:

驱动器 A 正在被校验

读逻辑扇区 XXXXXX 到 XXXXXX

按 ESC 键退出

如发现错误,则显示的中文意义是:

逻辑扇区 XXXXXX(驱动器 A 或 B)有一个错误
用 View/Edit 尝试恢复

按任意键继续

按 ESC 退出

如校验后没有发生错误,则显示信息的中文含义是:

没发现错误

驱动器(A)或(B)已被校验过

按任意键返回

如果没发现校验错误,但发现了坏块,则显示:

XXX Byte marked bad in X cluster(s)

[例]校验 A 盘

操作步骤如下:

①键入“V”,弹出一窗口供用户指定要校验的盘所在的驱动器

②键入“A”

③按任意键开始校验,若校验完毕没有发现错误,用户按任意键即可返回磁盘服务功能主菜单,否则将指出错误所在的扇区号。

(6)view/Edit(查阅/编辑)

view/Edit 功能查阅/编辑磁盘上的文件。

它与文件服务中 view/Edit 的操作相似,但它直接进入第二层菜单,并增加了一个功能键 F4,用以确定当前显示的信息所在的区域或文件的名称。当按下 F2 键,则弹出一个窗口,指出要显示的扇区。窗口信息是:

Indicate which sector to display

“B”=BOOT sector

“F”=First FAT sector

“R”=First ROOT directory sector

“D”=First DATA sector

“C”=Enter a cluster # (2-00355)

nnnnn=sector number

窗口的中文含义是:

“B”=BOOT 区,即引导扇区

“F”=第一个 FAT 扇区

“R”=第一个 ROOT(根)目录扇区

“D”=第一个数据扇区

“C”=输入簇号 # (2-00355)(随软盘类型而异)

nnnnn=扇区编号

用户只需根据自己的操作,按下相应的键即可。

(7)Map(磁盘映射)

Map 功能用来映射一个或多个文件或整个磁盘。以“地图”方式让用户看到磁盘空间的分配情况,盘的格式不同,映射“地图”也不同。以 360K 软盘为例,说明 M 功能的使用。

[例]查看 A 驱动器软盘的内存映像

操作步骤如下:

①键入“M”回车后输入指定的驱动器“A”后,屏幕显示:

PCTOOLS Deluxe R4. 21

Vol Label=None

—————Disk Mapping Service—————

Path=A:*.*

```
Entire disk mapped          52% free space
      Track   1   1   2   2   3   3   3
      0   5   0   5   0   5   0   5   9
Double sided
B.....*****
F.....*****
Side 0
F.....*****
D.....*****
---D.....*****
D.....*****
Side 1
.....*****
.....*****
.....*****
```

Explanation of Codes

* Available

. Allocated

B Boot record

h hidden

F File Alloc Table

r Read Only

D Directory

x Bad Cluster

“F” to map files. ESC to return.

②屏幕显示磁盘的内存映像图。图中的各个符号的含义如下:

* (自由)..... 其扇区为自由空间。

B(引导扇区).....该扇区包含一个引导记录,凡格式化过的软盘都有一个引导记录在磁盘的 0 面 0 道 1 扇区。

F(文件分配表).....该扇区(簇)为文件分配表。文件分配表记录着磁盘上每个扇区(簇)的占用情况。

D(目录).....该扇区为文件目录区,主要包括文件名、扩展名、文件长度及生成日期及时间。

.hr(数据区)..... 该扇区为文件数据区。

.(一般文件)..... 一般类型文件。

h(隐含文件)..... 隐含文件。

r(只读文件)..... 只读文件。

x(坏块)..... 坏簇。

③键入“F”后,屏幕显示出 A 盘所有文件的文件目录。在出现提示菜单中,按“G”键将映射被选择的文件。

④在进行单个文件的映射时,使用“←”或“→”键可映射上一个或下一个文件。

⑤键入“D”键返回。

(8)Locate(定位)

用来确定某一文件或某些文件在哪个目录下(路径),并指出文件长度及生成时间。

[例]查找 A 盘上所有扩展名为 .EXE 文件所在的目录

操作步骤如下:

①键入“L”键,并输入“A”表示从 A 盘查找

②根据屏幕提示在 .EXT(扩展名)后的方括号内输入 EXE 并回车确认,PC Tools 将自动扫描全盘。

此时屏幕分类显示出所有以 .EXE 为扩展名的文件所在的路径,文件名和文件长度及日期。

(9)initialize 格式化)

格式化多种类型的软盘。

[例]将 A 驱动器中 360K 软盘格式化为系统盘

操作步骤如下:

①键入“N”并输入“A”,回车后屏幕显示:

Drive A is about to be initialized (formatted).

Choose the formatting desired
and press ENTER to begin.

360K <---> Double-sided, 9 sectors per
track

320K

180K

160K

1.2M

其中含义是:

驱动器 A 将被初始化(格式化)

选择需要的格式

按回车键可开始操作

360K <---> 双面,每道 9 扇区

320K

180K

160K

1.2M

②根据屏幕提示选择软盘类型,并打回车键确定
确认无误后,按任意键开始

③格式化完成后,根据屏幕显示输入卷标名

④回车后,根据屏幕显示键入“Y”(YES),它将执行 SYS.COM(DOS 系统传送命令)到这张磁盘,回到主菜单后,再将 COMMAND.COM 文件(由 DOS 系统盘)拷入到这个刚被格式化的磁盘,该盘即成为 DOS 系统盘。

注意:此项功能不能对 C 盘进行格式化。

工程多媒体专家系统的技术展望

刘 箴 辽宁省铁岭市建筑设计院(11200)

1. 引言

自 80 年代以来,人们已在工程领域建造了大量的专家系统(ES)。这一方面源于技术积累的日益成熟;另一方面,推广技术的紧迫性也刺激了专家系统的研制开发。实践已证明,专家系统在工程领域具有巨大的开发潜力和经济效益。

由于专家系统本质上也是信息系统,其信息也以一定的媒介(媒体)传播,综观已建成的工程专家系统(EES),不难发现,它们都不同程度地存在着信息媒体的单调性。事实上,从信息媒体的角度,工程信息媒体极为丰富,除文字、图表还有图像、影像,声音等。早期的 EES 大多用 Lisp, Micro-Prolog 等人工智能语言或工具编程,故缺少图形支持;近期的 EES 开发则用高级语言编程(包括混合语言的编程),虽注意了图形接口设计,但很少涉及图像、影像以及声音处理。总而言之,以往的 EES 只注重专家系统的符号推理,忽视了图形、图象、影像及声音的知识表达和推理, I/O 信息表达单调,缺少智能性。

近几年来,多媒体这种新兴的信息处理技术正渗透到计算机研究的各个领域,它同样会给 EES 带来一

场革命, EES 和多媒体相结合将产生工程多媒体专家系统(EMES),这将是未来 EES 的发展趋势。本文着重讨论工程多媒体专家系统的基本特征,并针对我国国情,提出了基于 PC 机为开发平台的 EMES 近期应实现的目标。

2. EMES 的开发环境

根据我国的目前国情,直接采用多媒体计算机费用较大,以利用现有计算机为前提的原则,在现有的 PC 机上增加多媒体套件,以下提出 EMES 最基本的开发环境:

主机: 80386 SX/20MHz, 4MB 或 8MB 的 RAM, VGA 彩显, 80M 硬盘, 且具有数学协处理器。

附件: 两键鼠标, 彩色打印机。

套件: 语音卡、话筒, 扫描仪(如手持扫描仪)

软件: MS-DOS 5.0(6.0)操作系统, 或采用语音汉字系统, 声卡驱动程序, 编程采用 C, C++ 或汇编语言, 或采用其它高级语言和汇编语言结合的形式。

如条件许可, 还可增加立体声音响, 录像机, 触摸屏, 绘图仪等套件。

3. EMES 的几个基本特征

本文仅就 EMES 所共有的基本特征进行讨论,与 EES 相比,EMES 具有如下几个特征:

1)采用最自然的人机对话方式:

EES 的用户一般通过键盘或鼠标与系统交互,EMES 将以多种方式与用户交互,它将实现逼真的人机对话。例如,EMES 可以由话筒发出自然语言的指令,由程序转化为系统可识别的驱动命令,有些系统则是通过图像扫描器将图像作为指令输入系统中。EES 一般在输入数据之前要求用户填写问题预答表,在输入时按菜单提示进行,EMES 则省去了这个过程。

2)多媒体信息显示的同步控制:

EMES 可以实现在单机上的多任务调用,它在建立知识映射和管理的同时,也将知识媒体送到显示缓冲区,而且使得视频数据流和语音数据流同步控制。由此,必须用程序语言解决 PC 机的各种中断调用,最大限度地发掘 PC 机的硬件资源。

3)集多种知识表达于一体:

在 EMES 中,知识媒体不仅有文字、图形,还包括图像(静态的)、影像(动态的)及声音。这意味着 EMES 的知识库结构较 EES 远为复杂,为此以下几方面研究需向实用化迈进:

A)图形、图像中的知识提取;B)图像的模式识别技术;C)神经网络快速收敛算法;D)图像、影像库和文字知识库的匹配;E)语音的数据压缩处理;F)自然语言语句理解。

由于在不同的工程领域,多媒体信息的差异性,将导致各种新型的知识表达方法问世,面向对象的知识表达技术将进一步得到补充。

4)系统内部具有较高的透明度:

EMES 将以类似人类专家的方式来传播信息,它能和用户深入沟通,用户可以向系统寻求解释、咨询、谈话、浏览、游戏等,在此过程中信息均以多媒体的形式表现,故系统具有良好的透明性。

5)方便的知识获取:

在 EES 中,知识获取一直是最困难的,因为知识必须事先进行整理,以一定的方式存入计算机才能使用。与此不同,在 EMES 中,知识获取将是一件十分方便的事情。例如,采用图像扫描器,利用 EMES 的知识获取模块,可直接将书本知识转化为系统的内部知识表示;再如,可以由专家利用话筒直接向系统传授知识,甚至可以将专家们的辩论进行录音。

4. EMES 近期要实现的目标

显然,实现具有如上所述特征的 EMES 是很困难的,为此提出近期实现的目标:

1)在已有系统的完善或新系统的开发中,应加入图形或图像甚至声音处理模块,建议在 EES 的开发中,必须要求 EES 具有图形处理的能力,否则这种 EES 的开发商品化意义不大。为此,某些不支持图形的编程语言或专家系统工具需要增加图形接口,编程语言提倡用 C++,另外,还要使系统具有读取由扫描

器输入的静态画面的功能。关于声音的处理,甚至不用语音卡也能达到目的,由于各种语音汉字系统的问世,EES 纳入声音没有大的技术困难。

2)研究各种高效的图形、图象数据压缩算法:

由于图形、图像的数据量较大,而 PC 硬盘空间相比太小,故研究出高效的数据压缩算法在 EMES 开发中举足轻重。

3)基于书本知识的知识获取系统:

由于书本知识来源方便,如果将书本知识直接纳入专家系统,将大大加快 EMES 的知识库和数据库的建造。目前,这方面研究已开始进行,近期将会得到初步应用。

4)图形象素的向量化处理:

图形象素的向量化处理在工程中经常遇到,图形实体(图片、照片等)在由扫描器扫描形成图像文件后,可在此图象中提取字符及基本的图形元素(直线、圆等),目前此项技术已逐步实用化。

5. 结束语

多媒体技术的出现,将给工程专家系统的开发带来一场革命,人们将不仅重视文字知识的整理和提炼,也将注重图形、图像、影像、声音等信息的表达,各种新型的智能处理技术将问世,可以预计,工程多媒体专家系统的开发将是未来几年较为热门的研究课题之一。

《电子产品维修与制作》 创刊第二届电子技术 与维修大专函授班招生

《电子产品维修与制作》杂志是中国电子工业发展规划研究院主办的实用电子技术科普月刊,是国内首家专门刊载电子产品维修与制作知识的杂志。该刊遵循实用性、先进性、系统性、知识性、普及性和资料性的原则,辟有众多栏目,可满足各行各业初中级技术人员、设备维修人员、家电维修人员、电子爱好者等多层次的需要,并可对众多读者、作者进行全方位的服务。该刊16开本、48页,正文胶印,4封彩印,公开发行,今年9月创刊。9—12期自办发行,明年邮局征订发行,单价2.40元。1994年4期定价9.60元,1995年全年定价28.80元。欲订者请汇款至北京市海淀区永定路123号《电子产品维修与制作》杂志编辑部。邮编:100039

杂志社与天津广播电视大学、北京现代家用电器协会联合举办的“电子技术与维修大专函授班”设办公、工业和家用电器设备原理与维修三个专业,发“双结业”证书,凡具有中等文化水平均可报名学习。第二届招生工作业已开始,简章来函备索,联系地址同上。

基于声霸卡开发音响制品管理系统的研究

刘维亮 葛仁伟 上海工业大学电子工程系

一、引言

当今随着超大规模集成电路(VLSI)和信息技术的飞速发展,多媒体已成为九十年代计算机工业界的热门话题。所谓多媒体就是指能够同时抓取、处理、编辑、存储、展示不同类型信息的技术。这些信息媒体包括:文字、图形、图象、动画、音乐、语音等。

目前世界上许多公司和厂家纷纷展示各自的多媒体软硬件产品,多媒体市场愈来愈大,销售量与日俱增。仔细研究不难发现,大多数多媒体产品特别是为数众多的图形图像处理产品价格昂贵,且对工作平台要求较高。结合国情,要在我国普及多媒体技术还有相当漫长的道路。另一方面,音响制品,诸如密纹唱片、盒式磁带,CD等均已走入千家万户,音响产品如 Walk-

man、高档音响,直至“发烧级”产品随处可见。同时,大量价格便宜的286,386兼容机正在企事业单位、学校及家庭中迅速普及。正是基于以上的事实,选用合适的音频卡结合普及率高的音响产品与价廉的微机,开发对音响制品的管理、销售、编辑等系统成为可能。

二、音频卡及其产品分析

微机处理声音是多媒体技术的重要组成部分。由于其总体技术较图形图像处理简单,所以发展较快,产品相对成熟,目前市售的声音卡种类繁多,各俱特色。(见表一)

各种音频卡的基本原理从根本上说是一致的。它们和微机构成的主宿结构工作框图如框图1。

表1. 市售流行音频卡性能一览

产品名称	生产厂家	bit 位数	KHz 采样频率	立体声/单声道	SCSI	从 PC 兼容	从 IDI 接口
Ultrasound	Advanced Gravis	16	44.1	立	无	否	有
Audio Port	Antex Electronics	12	22.05	立	无	否	无
Stereo-F/x	ATI Technologies	8	44	立	无	否	有
Audiopoint	Media Vision	8	22.05	单	无	否	无
Pro Audio Spectrum 16		16	44.1	立	有	是	有
Pro Audio Spectrum+		8	44.1	立	有	是	有
Thunder		8	22.05	单	无	否	无
Windows Sound System	Microsoft	16	44.1	立	无	否	无
Audio Master	Omni Labs	12	44.1	立	无	是	无
Audio Canvas XA-16	Promedia Tech	16	48	立	无	否	无
Sound Blaster	Creative Labs	8	22.05	单	无	否	是
Sound Blaster/16		16	22.05	立	无	是	是
Sound Blaster Pro		8	44.1	立	无	是	是

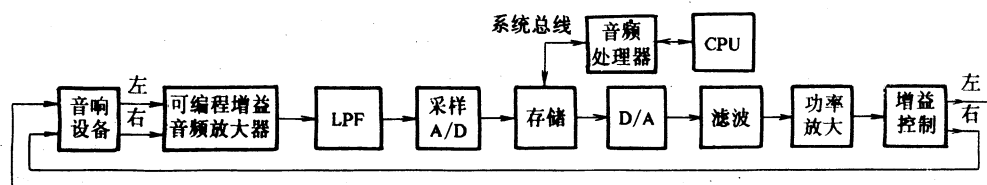


图1. 音频处理系统的构成框图

1. 音响设备:音频信号来源与音响设备,它可以是各类 CD、卡座的线入(line-in)也可以是麦克风一类的拾音器,它通常提供左右双声道立体声信号源。

2. 可编程增益音频放大器:左右声道的信号源输入音频放大器,通过软件可对每个通道分别控制放大器增益。

3. LPF(低通滤波器):设输入音频信号频率 Ω , $0 \leq \Omega \leq \Omega_h$, Ω_h 为 Nyquist 频率, Ω_s 为采样频率。信号频谱在高频 Ω_h 处迅速下降,但并不意味着频带结束,故在采样后易造成频谱重叠称为“混叠现象”,严重影响声音质量,所以利用前置低通滤波器防止频谱混叠。取截止频率为 $\Omega_s/2$,则送去采样的最高频率 $\Omega_h \leq \Omega_s/2$,则 $\Omega_s \geq 2\Omega_h$ 符合 Shannon 采样定理。实际应用中常选 $\Omega_s = (2.5 \sim 3)\Omega_h$ 。

4. 采样和 A/D:模拟的音频信号通过采样和 A/D 后成为数字信号,为降低量化噪声,常提高采样比特率,但又兼顾数据量,故音频信号常用有 22.5KHz(相当于 FM 广播质量)和 44.1KHz(相当于 CD 质量)的采样频率。

5. 存储与处理:数字音频信号存储在 RAM 或外设之中。主机 CPU 与专用音频处理单元通过系统总线对其进行各种处理如合成、波形分析、剪辑、淡入淡出等。

6. 输出:处理后的数字信号经 D/A 转为模拟信号,又经改善输出语音信号质量的滤波器后送入功放与左右声道分别程控的增益控制器再回送入音响设备。

三、Sound Blaster Pro I 性能分析

1. 一九九一年十一月 Microsoft、Philips 等十四家厂商组成的多媒体市场协会应运而生,同时制定了 MPC (Multimedia Personal Computer) 系统,如下:

a. 硬件平台

80386 SX 以上处理器

—2MB RAM(建议4MB)

—30MB 硬盘(建议越大越好)

VGA(16色,建议256色)

CD-ROM

—CD 数字音响输出能力

—传输速度不低于150k/s

—占用 CPU 开销 < 40%

—平均寻址时间 < 1秒

音频卡

—数字化录音(AD 转换)

—8位精度11.025、22.05采样频率

—录音与放音之间时间误差 < 2%

—内置音乐合成器

—内置调音台

—占用 CPU 开销 < 20%

鼠标器

101 键盘

串并口

MIDI I/O 口

b. 软件平台

Microsoft Windows 3.1 (或 Windows Graphical Environment version 3.0 with Multimedia Extensions Version 1.0)

MS-DOS 3.1 以上(建议5.0以上)

MS-DOS CD-ROM 扩展版(MSCDEX 2.20 以上)

2. Sound Blaster 声霸卡系列音频卡目前已成为国际上 MPC 的标准声响设备,在声卡市场上有60%的占有率。全球约五千家以上的软件厂商支持 Sound Blaster 卡。Microsoft 的 Windows 3.1 就是以 Sound Blaster PRO 的音响为蓝图。使用 Sound Blaster PRO 自然就能执行所有符合 MPC 规格的软件。

Sound Blaster PRO I 的主要软硬件如下:

a. 硬件

• 8 bit 立体声 D/A 转换,硬件自适应差分编码 (ADPCM) 压缩技术(2:1~4:1)

• 通道0、1、3DMA 传送,由跳线来实现选择通道

• 4K~44.1KHz 采样频率

• 软件控制的淡入淡出,声像,音量,左/右调节和麦克风混合

• 四声部雅马哈 OPL3 立体声合成器,同时发音数 20 的 FM 合成声音用八种不同波形和四种算法进行四声部 FM 合成

• MIDI 适配器和标准 MIDI 插座电缆 64 bit FIFO

• 高速 CD-ROM 专用接口,占用 CPU 开销远小于 40%

• 内置功放每通道 4W 输出,可和任何扬声器,耳机相连,带 AGC 放大器的麦克风输入孔

b. 软件

• 交互式多媒体演示系统 HSC Interactive SE

• 自动唱机 Juke box For Win 3.1

• FM 智能电子琴 FM Intelligent Organ

• 会说话鹦鹉 Talking Parrot

• 会说话日程表 Creative Talking Scheduler

• 波形工作站 Creative Wavestudio

• 多媒体简易开发演示系统 MMplay

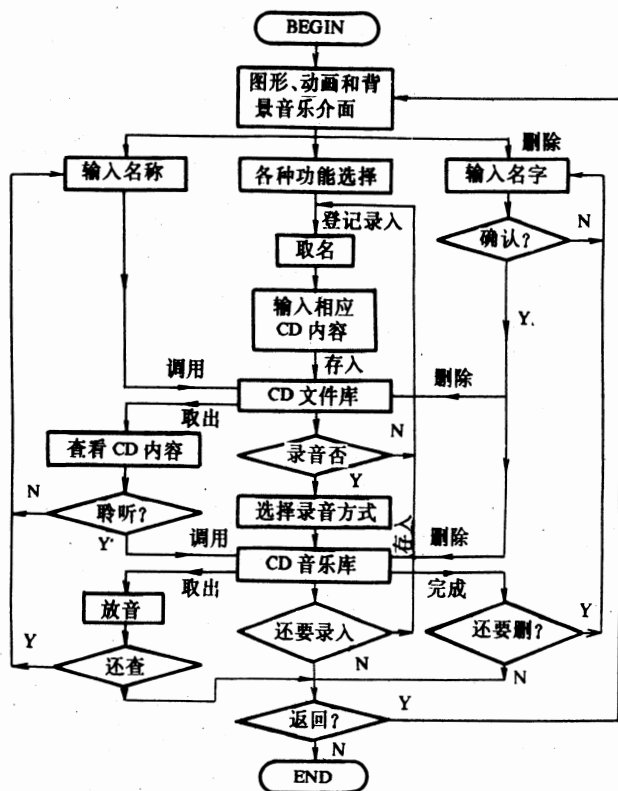
• MIDI 文件驱动程序 SBMIDI

• 声霸标准接口模块 SBSIN

四、应用实例

作者采用了 Sound Blaster PRO I 卡与 386DX 兼容机为上海工业大学文化艺术指导中心开发了一个 CD 唱片管理系统,在此特介绍原理和工作框图。

1. 基本原理:此系统对上百张 CD 进行统一方便的管理,包括登记、录入、查询、打印、聆听、修改、删除等功能。对音乐信号的处理依赖于 Sound Blaster PRO I 卡上使用 MMplay 和 C 语言进行简单的二次开发。各种管理和数据库系统由 C 语言处理,两者通过 DOS 接口紧密联合。这个系统的开发为音响制品管理带来



了极大方便,也使查询各种制品十分生动,查询者不仅了解 CD 内容,更能听取精彩片段,使用价值相当高。

2. 工作框图(见图2):为简化框图只列出了登记、录入、查询三个功能流程。

五、结论

多媒体技术的开发应用在我国起步较晚,整体技术还较落后,综合考虑各方面的因素,在现有条件下选择合适的道路进行多媒体技术的开发应用是十分有必要的。本文提出的方法及构成的实例以使用方便、价廉物美得到了较广泛的应用和一致好评。

总之,由于引入多媒体技术符合当前应用需求的潮流,必然会成为计算机发展的一大趋势。可以大胆预言,多媒体计算机技术在不远的将来很可能会占领所有拥有计算机、电视机、录像机、音响设备的场所,其市场可能是计算机和音像系统的总和,并占领一切文化艺术和娱乐场所。

作者通信地址:上海茂名南路169弄8号(200020)

PC 机与 MCS—51 单片机的主从式数据通信

康赐荣 福建泉州华侨大学电子工程系(泉州362011)

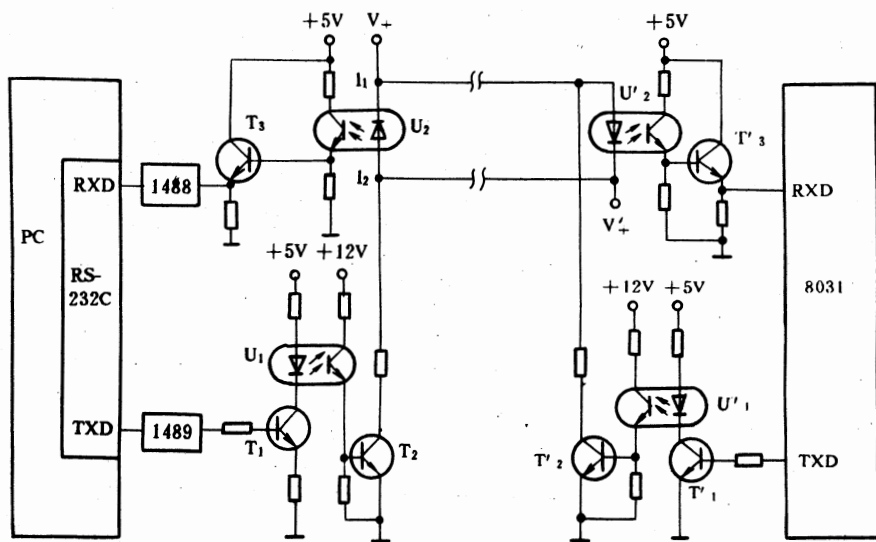


图1 串行通信电路图

1 前言

在食油罐群($8 \times 1000\text{m}^3$)微机二级测控与管理系统中,下位机采用 MCS-51 单片机完成油罐上、中、下层油温、油位、食油瞬时流量、累计流量的检测及进、出油管理阀门的控制等功能。上位机采用 286 微机对下位机采集的数据进行存盘、越限报警、数据处理及对阀门进行遥控,显示罐群工艺流程图、棒图,并完成班报、日报和月报等工作。本文择其中上位机与各下位机的主从式

数据通信加以介绍。

2 串行通信硬件电路

下位机与上位机之间最大距离不超过1000m,每次传输的数据量不大,因此通信速率不成问题,而关键是要保证在工业环境中确保数据传输的可靠性。为此我们选择基带传输的数据通信,采用光电耦合的电流环传输,波特率2400波特,最大距离不小于1000m。电路如图1所示。其中MC 1488和MC1489完成TTL与EIA的电平转换。当上位机输出高电平时, T_1 、 U_1 、 T_2 导通,恒流沿 $V_+ - I_1 - U_2' - I_2 - T_2$ 流通, U_2' 、 T_3' 导通,下位机的RXD得到高电平;当上位机输出低电平时, T_1 、 U_1 、 T_2 截止,无环流, U_2' 、 T_3' 截止,下位机RXD得到低电平;下位机与上位机传送数据时情况类似,不赘述。

采用光电耦合及电流环传输可提高通信系统抗外界电磁干扰能力,避免共地及通信线对微机工作的干扰,提高整个系统工作的可靠性。

3 异步串行通信原理

3.1 8250的初始化

本数据通信系统的异步串行通信协议规定为:一个信息字符由1位起始位“0”、8位数据位、1位奇偶校验位(或地址/数据标识位)、1位停止位“1”组成(图2)。奇偶校验位为1表示上位机发送地址帧,奇偶校验位为0表示发送的是命令帧或数据帧。

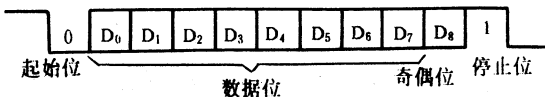


图2 帧格式

PC机异步通信接口采用可编程的8250芯片作为异步接收/发送器。它含有10个寄存器(表1)。

在通信之前要对8250进行初始化,主要包括:

表1 8250的内部寄存器

I/O 端口	LCR 的 D_7	IN/OUT	寄存器名称
3F8H	0	OUT	发送器保持寄存器 THR
3F8H	0	IN	接收器数据寄存器 RDR
3F8H	1	OUT	波特率因子寄存器(低位) BRL
3F9H	1	OUT	波特率因子寄存器(高位) BRH
3F9H	0	OUT	中断控制寄存器 ICR
3FAH	X	IN	中断标识寄存器 IIR
3FBH	X	OUT	线路控制寄存器 LCR
3FCH	X	OUT	MODEM 控制寄存器 MCR
3FDH	X	IN	线路状态寄存器 LSR
3FEH	X	IN	MODEM 状态寄存器 MSR

对 LCR 编程为0BH,它规定了数据帧格式为8位数据位,1位停止位,1位奇偶校验位和1位起始位共11位。不

进行奇偶校验,奇偶位作为地址和数据的标识位用。设置波特率因子为0030H,即波特率因子 $=1.8432 \times 10^6 / (\text{波特率} \times 16) = 48 = 30H$

在 LCR 的 $D_7=1$ 下,00H 送3F9H,30H 送3F8H;MCR 设置为03H;ICR 设置为00H,即屏蔽所有四种中断。

3.2 单片机串行口的初始化及串行通信原理

MCS-51单片机用定时器 T_1 工作于方式2作为波特率发生器,设定时器计数初值为 TH_1 ,则

$$TH_1 = 2^8 - \frac{f_{osc}(SMOD+1)}{384 \times \text{波特率}}$$

PCON 的 SMOD 设置为0, $f_{osc}=11.0592\text{MHz}$,波特率为2400波特,可求得 $TH_1=244=F4H$ 。

单片机串行口的初始化包括串行控制寄存器 SCON 的设置,定时器方式控制寄存器 TMOD 及 T_1 的定时初值的设置,专用寄存器 PCON 的 SMOD 位的设置及中断允许寄存器 IE 的设置。SCON 设置为 F0H,即选择串行口工作方式3,11位 UART,帧格式同图2。REN 置1允许串行接收。SM2为单片机工作方式2或3的多机通信控制位。当 SM2为1时,只有接收的数据第9位 $RB8=1$,才激活接收中断标志 RI,向 CPU 申请中断,数据才能读入累加器 A;当 SM2=0时,则不管 $RB8$ 为0还是1,都激活 RI。根据这个功能可实现多机通信。上位机要同某下位机通信时,发数据为该机地址,且第9位为1表示地址帧,由于各下位机的 SM2=1, $RB8=1$,各下位机均接收该地址帧,与本机地址比较,只有被寻址的下位机地址符合,置 SM2=0。当上位机向被寻址的下位机发命令或数据时,第9位为0表示命令帧或数据帧,这样只有被寻址的下位机才会接收,实现通信。

4 异步串行通信的编程

上位机通信子程序包括上位机向下位机发送要求下位机发送检测数据的命令和接收检测数据,以及对阀门进行遥控的命令。通信子程序供需要时调用,它用80286汇编语言编写,经汇编产生 Obj 文件,然后与 C 语言编写的主程序链接产生 exe 可执行文件。下位机主程序中进行串行口、定时器的初始化,设置波特率,设置接收及发送地址指针,开中断等初始化工作。下位机接收上位机发送数据命令,把检测数据发送给上位机。下位机还接收上位机的遥控命令,完成规定的测控动作或其它操作。为提高数据通信的可靠性,每次数据传送都校验累加和,错误重发和错误重收功能,错误重发及错误重收最大次数为3次,超过3次作线路故障处理。

5 结束语

本通信系统采用光电耦合和电流环实现数据传送,抗干扰能力强。通信系统采用汇编语言编写,运行速度快,占内存少,并采取一些提高数据传送可靠性的措施,使本通信系统能在工业环境下可靠地工作。

参加本项工作的还有黄永福同志,在此谨致深切的谢意!

时钟芯片 MSM5832RS

梁 冶 辽宁抚顺电子技术研究所(113008)

一、引言

在很多场合下,都要用到时钟芯片。特别是能够与微机直接接口的时钟芯片,在计算机技术迅速发展的今天,更受人们的普遍欢迎。下面,本文将要介绍的MSM5832RS 时钟芯片,就是能够与微机接口,并具有掉电保持功能的时钟芯片。

该芯片计时准确、功能齐全,经我们多次使用,效果良好。

二、MSM5832的功能原理

1. 引脚排列及说明(见图1)

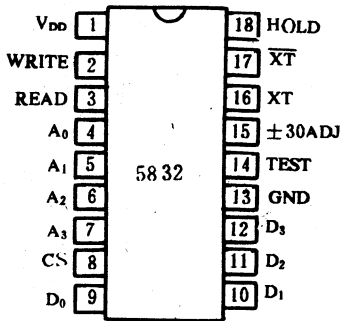


图1

V_{DD} : +5V 电源。

GND: 负电源。

$A_0 \sim A_3$: 地址输入端。

$D_0 \sim D_3$: 数据 I/O 端。

WRITE: 写使能端。

READ: 读使能端。

HOLD: 计数保持端。

CS: 片选端。

TEST: 测试输入端。

$\pm 30ADJ$: ± 30 秒校时输入端。

XT, \overline{XT} : 晶振输入端。

2. 寄存器表

5832内部设有秒、分、时、星期、日、月、年等寄存器。见下:

地址输入 $A_3 A_2 A_1 A_0$	存储 内容	数据 I/O $D_3 D_2 D_1 D_0$	数据 范围	注 释
0 0 0 0	秒个位	* * * *	0~9	地址选择 S_1, S_{10}
1 0 0 0	秒十位	* * *	0~5	写指令完成后,无论 $D_0 \sim D_3$ 为何值全部复 位到零。

0 1 0 0	分个位	* * * *	0~9	
1 1 0 0	分十位	* * *	0~5	
0 0 1 0	时个位	* * * *	0~9	
1 0 1 0	小时 十位	* * * *	0~1 0~2	$D_2=1$ 下午; $D_3=1$ 24小时制 $D_2=0$ 上午; $D_3=0$ 12小时制
0 1 1 0	星期	* * *	0~6	
1 1 1 0	日个位	* * * *	0~9	
0 0 0 1	日十位	* * *	0~3	$D_2=1$,2月份为29天; $D_2=0$,2月份为28天。
1 0 0 1	月个位	* * * *	0~9	
0 1 0 1	月十位	*	0~1	
1 1 0 1	年个位	* * * *	0~9	
0 0 1 1	年十位	* * * *	0~9	

在上面的寄存器表中,有二处需要说明:一个是小时十位,另一个是日十位。小时十位有24小时制式和12小时制式两种,由 D_3 位控制。写入时,上下午由 D_2 位控制。在日十位中,2月份为28天还是29天由 D_2 位控制。

3. 写入方式

在使用5832做时钟芯片时,首先要对其写入正确时间。我们在写入操作时,除了知道寄存器的地址和特殊要求外,还要了解写入方式的时序波形图。

写入方式的波形图如下所示:

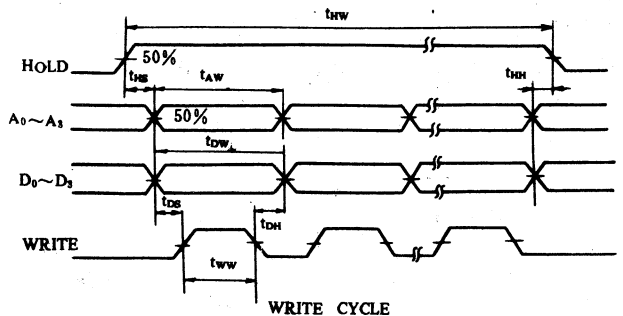


图2

$THS_{MIN}=150\mu S$; $THH_{MIN}=0\mu S$
 $THW_{MAX}=10^6\mu S$; $TAW_{MIN}=1.7\mu S$
 $TDW_{MIN}=1.7\mu S$; $TDS_{MIN}=0.5\mu S$
 $TDH_{MIN}=0.2\mu S$; $TWW_{MIN}=1.0\mu S$

4. 读出方式

对5832一旦写入时间数据后,时钟即开始运行。我们只要按着正确的方式从芯片的指定单元中读出相应的时间数据并加以显示出来,就可以做为时钟应用了。因此,我们还需要了解读出方式的波形图。

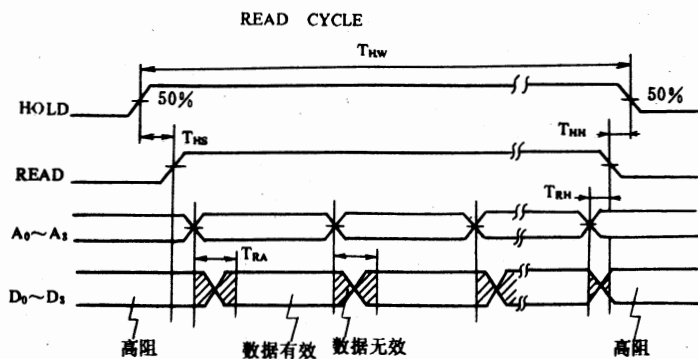


图3

其中: $THS_{MIN}=150\mu S$;
 $THH_{MIN}=0\mu S$; $THW_{MAX}=10^6\mu S$;
 $TRH_{MIN}=0\mu S$;
 $TRA_{MAX}=6\mu S$;

三、MSM5832的实际应用

1. 接口线路

我们以 Z-80PIO 为例,接口线路如图4所示。

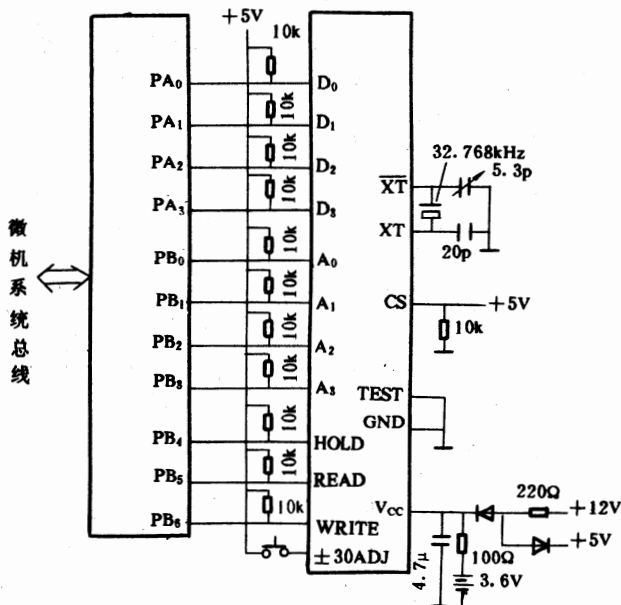


图4

图中,测试端接地,测试禁止;片选端接高电平,使该片始终选中;±30秒校时输入端接一开关,做校时对称用;每一位端口都接有上拉电阻;3.6V 电池做掉电保持用;晶振的频率为32.768KHZ。

2. 程序设计

① 写入程序

根据接口线路图和写入波形图,我们就可以编制出写入程序。现在我们以写分个位为例。程序如下:

ORG WRITE	注 释
WRITE; LDA, 10H	
OUT (PIOB), A	HOLD=1
CALL DELAY1	延时 THS
LD A, 12H	分个位地址
OUT (PIOB), A	输入地址
LD A, 09	
OUT (PIOA), A	送09分
CALL DELAY2	延时 TDS
LD A, 52H	HOLD=1 WRITE=1
OUT (PIOB), A	写入数据
CALL DELAY3	延时 TDH
LD A, 12H	HOLD=1
OUT (PIOB), A	A0~A3=2
END	结束

以上是写分个位程序。同理,也可以编写出写其它时间位数的程序。不过,要改变寄存器的地址。

为了简化写入程序,我们可以将写入地址做为变量编出循环程序或者以子程序方式多次调用。

2. 读出程序

读出程序与写入程序基本类似。只不过在读出程序中,写信号(WRITE)无效,读信号(READ)有效。其余相同。我们仍以读分个位为例。程序如下:

ORG READ	注 释
READ; LD A, 10H	
OUT (PIOB), A	HOLD=1
CALL DELAY1	延时 THS
LD A, 32H	读有效
OUT (PIOB), A	读分个位
CALL DELAY2	延时 TRA
IN (PIOA), A	输入数据
AND 0F	屏蔽高位
LD (DATA), A	保存数据
LD A, 10H	
OUT (PIOB), A	HOLD=1
END	结束

如果我们想要读其它数据,只要改变相应的地址就可以了。此不赘述。

查找备份文件恢复路径点滴体会

彭 禾 四川省涪陵地区卫生局(648000)

历受备份文件无文档,恢复路径不明而难从使用的苦恼,经过探索,渐予解开。1991年问世的 DOS V5.0 功能加强,其 RESTORE. EXE 可直接显示恢复路径,‘难题’成为历史。

DOS V3. 20 及早期版本运行 BACKUP 命令,先在目标盘建立卷名及标志文件——BACKUP. @@@,再逐一给待备份文件的正文之前增添128字节路径之类信息。执行恢复命令时,备份文件前128字节被滤除,以文件原貌装入目标盘。

由此可见,备份文件恢复用路径信息不难查得,DEBUG. COM 完全胜任,不过,微机用户较为熟悉的 PC Tools 方便又直观,其4. 22版的操作如下:

1. PCTOOLS
2. F10 (进入文件功能模式)
3. d; (选驱动器)
4. ↓[*****]CR[*****] (指定 XXX. COM .EXE .PRG#)
5. E (取编辑功能)
6. 查位移 005H ASC I *(从第六个字符起为路径等)
7. [ESC**→*****] (查下一个备份文件)
8. ESC**ESC**Y (退 DOS 下)
选可执行文件,勿取 XXX. SYS. DBF 之类
* 必要时按 F1 转换显示状态

DOS V3. 30 BACKUP. COM 的功能明显加强,参数增至八项。它相应地在目标盘根目录只建立两个文件:

1. 备份文件 BACKUP. nnn, 编号从001起,含链接状态的备份文件,既提高读写速度,又节省磁盘空间。
2. 管理文件 CONTROL. nnn, 编号对应,文件长度与备份文件数量成正比,保存公用的路径名,文件名(表),以及其它管理信息。

此外,使用/F 参数若未指定文件名则在源盘根目录建立 BACKUP. LOG,含备份日期、时间、编号、路径及文件名,打印件部分如下:

```
1993-1-25 16.17.25
001 \PT6\PCSHELL. EXE
001 \PT6\PCSHELL. OVL
:
```

该文件在备份过程中不断增长,运行结束后若能容纳尽量拷贝至末号备份盘或/及用 TYPE BACKUP. LOG>PRN 给出打印件。在存档之前宜查阅备份文件是否齐全,若数目不足,大都是文档位未置‘1’,即目录项 0B—b5 为‘0’(PC Tools 属性显示‘.’及‘0xH’),可用 B>ATTRIB +A d:[PATH]*.* 预

处理——文档位置‘1’(相应为‘A’及‘2xH’)。

欲查看恢复用路径,可调 PC Tools ‘编辑’CONTROL. 001,屏幕上份空白区中显示‘F’之后的字符为路径,如:FSTA、FDOS、FGFYL 等相应的子目录名为 STA、DOS、GFYL,若仅现‘F’即后无字符则表示根目录;下级子目录名均在子目录文件名(表)的末尾,上例为 xxxxFSTA\LT,空行之后列 LT 子目录文件名(表),以此类推。

DOS V5. 0 RESTORE. EXE 增加/D 参数,用以显示路径名,可谓一目了然,其命令格式:

RESTORE drive1: drive2: [path[filename]]/D

即使硬盘安装低版本 DOS,也可经含 RESTORE. EXE 的 DOS V5. 0 软盘启动后,查阅备份盘上的路径名,堪称简便。使用/D 参数时,drive2: 不可缺(出错信息为 no target drive specified),虽然它并不恢复文件至目标盘。

单用此参数仅显示由根目录备份的文件名:

*** Files were backed up 1993-08-01 ***

*** Listing files on drive A: ***

```
Diskette:01
\COMMAND. COM
\CONFIG. SYS
\AUTOEXEC. BAT
```

若缺相应的信息,譬如备份盘上仅含从源盘目录中备份的文件,显示如下:

WARNING! No files were found to restore

因此,通常它与/S 参数伍用,同期显示路径及文件名,一举两得。

实践展示,IBM PC/XT 286 微机 (DOS V3. 30 设 C 盘 7MB 余置逻辑盘),以 C>ATTRIB +A C:STA*.* /S 及 C>BACKUP C:STA A:/M/S 将 C 盘 STA 子目录内容包括下级子目录文件完整地备份至 A 盘,恢复时无需在目标盘先建 STA 子目录及 CD\STA 等,执行 B>RESTORE A: C:\ /s,即自动圆满地恢复到 C 盘,包括其下级子目录 STA\LT 的全部内容。另取以 STA 为当前目录的备份文件,执行同样的恢复命令——B>RESTORE A: C:\ /s,效应一致。经查阅 CONTROL. 001 中的路径名均为 FSTA,调用 COMP. COM 分别比较上述两级从不同当前目录备份的 CONTROL. 001 及 BACKUP. 001,均显示 File compare OK,两者等同。

又作实验,先后用 DOS V2. 10 及 DOS V3. 20 软盘

启动微机,对用早期版本备份的文件执行 A>RESTORE A: C:\ /S,也获满意结果。(A 驱 360 KB)

由此可见,RESTORE 命令向下兼容性良好,并不受两者备份格式迥然不同的影响。DOS V5.0/6.0 的 RESTORE.EXE 效应更属上乘。

或可认为:RESTORE d: d:\ /S 具有广泛适用性,条件是目标盘不存在与备份盘上同名一级子目录

的可能性。在缺 BACKUP.LOG 或 DOS V5.0/6.0 的条件下,若无十分把握,可作实验,举例:微机 A 驱 1.44MB, B 驱 1.2MB(或 A: 1.2MB+B: 360KB),不论备份盘用何版本 DOS 制作,简便的方法是用 DOS V3.30 或高版本 DOS 软盘启动,执行 d>RESTORE B: A:\ /s, #01 备份盘运转结束即中止执行,列 A 盘目录,查明路径,酌情处理。

谈 WPS 文件阅读器

陈志鹏 河北省黄骅市职高(061100)

《电子与电脑》94.2 刊登了一篇名为《也谈 WPS 文件阅读器》的文章,作者确实解决了 WPS 加密文件的显示问题,但作为严谨和完整的文件阅读器,尚存在以下欠妥之处:

首先:WPS 设置密码即加密过程:把密码加密后写入密码区,以密码区的 8 字节与文件内容循环异或操作,写盘。取消密码即解密过程:读入并验证密码,正确时取密码区 8 字节与文件内容循环异或,写盘,再将密码区首位写 00H,标注该文件为非加密文件,其它密码位不变。因此,原文件在显示加密后又解密且先前密码多于一个的文件时,显示混乱。

第二,原文件没有过滤掉非 WPS 文书格式类文件,当阅读到此类文件时,显示混乱。

第三,原文件没有处理软回车换行符及打印控制符,显示格式不正确,偶尔显示混乱,

为此,本人用 Turbo C 2.0 编制 READ 程序,并在 386、286 机上调试通过

READ.C 程序清单

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
main(int argc, char * argv[])
{
    FILE * fp;
    int i, num, t;
    unsigned char password[8], file[8], fname[20];
    if(strcmp(argv[1], "/?") == 0)
    {
        printf("使用格式为 READ 文件名.扩展名 文件必须为 WPS 编辑的文书文件\n");
        return 0;
    }
    if(argc < 2)
    {
        printf("\n 你忘记输入查看的文件名,请现在输入:");
        gets(fname);
```

```
    }
    else
        strcpy(fname, argv[1]);
    if((fp = fopen(fname, "rb")) == NULL)
    {
        puts("\n 文件名或路径可能错,文件不能打开!\007\007请检查\n");
        exit(1);
    }
    if((getc(fp) != 0x01) || (getc(fp) != 0xff))
    {
        printf("\n\007文件【%S】非 WPS 格式文书文件", argv[1]);
        exit(2);
    }
    fseek(fp, 0x2dd, 0);
    fread(password, 1, 8, fp);
    fseek(fp, 0x400, 0);
    t = 1;
    for(num = fread(file, 1, 8, fp); num = fread(file, 1, 8, fp))
    {
        for(i = 0; i < num; i++)
        {
            if(password[0] != 0)
                file[i] = file[i] ^ password[i];
            if((file[i] == 0x8d) || (file[i] == 0x0d)) {t = -t; continue;}
            if((file[i] == 0x0a) || (file[i] == 0x8a) && (t < 0)) {t = -t; printf("\n"); continue;}
            if((file[i] == 0x0a) || (file[i] == 0x8a)) {t = -t; continue;}
            if((file[i] < 0x80) || (file[i] > 0xa0))
                printf("%c", file[i]);
        }
        if(num < 8) break;
    }
    fclose(fp);
    return 3;
}
```