

E&C
1992
●一九九二年 ●总期第83期
2

電子

ISSN 1000-1077

與 電腦



• ELECTRONICS AND COMPUTERS •

SCB—1 单片单板机!

武汉市尚吉电子研究所研制开发



武汉市尚吉电子研究所

联系地址: 武汉武昌珞瑜路37—3号

电话: 716138

邮政编码: 430070

电报挂号: 4561

开户银行: 工商银行洪山区办

帐号: 2604—67—016072001

● 国内代号: 2—888 定价: 0.95元

普及型PC机用户委员会常用文录软件清单

软件名称	版本号	适用机型	盘数	软件名称	版本号	适用机型	盘数	软件名称	版本号	适用机型	盘数
CCDOS	V2.0	A	1	GCLISP (汉化)			8	汉化 QUICK BASIC	V4.0		3
CCDOS	V2.1	A	1	TURBO PROLOG (系统盘)			1	编译 BASIC			1
GW DOS	V3.0	C	1	TURBO PASCAL	V5.0		3	DLLTIL		C	7
PCDOS	V3.3		1	编译 PASCAL			2	INFORMIX		C	4
CCDOS	V4.0		5	TURBO C	V2.0		7	DOFMAT		C	1
CCBIOS2.13	A-H		7-8	C 语言			4	CH87			1
LLDOS(双音双拼)			1	MSC	V5.0		9	IBM 检测程序			1
UDOS	V1.0		4	MSC	V6.0		7HD	诊断程序		A	1
MSDOS	V4.01		1	COBOL II	V2.0		2	磁盘管理软件 (DM)			1
CLIB 24	588K		2	FORTRAN			5	DMA (磁盘管理恢复文件)			1
	607K		2	FORTRAN	V5.0		2	汉化 DEBUG			1
CLIB 24*24			11	汉化 DBASE III			5	PCTOOLS (汉化)	V1.10		1
CLIB 32*32			12	DBASE III PLUS			1	PCTOOLS (长城机专用)	V4.12		1
CLIB 48*48			25	编译 DBASE III		A	6	PCTOOLS	V4.11		1
各种打印驱动程序			1	DBASE III 辅助系统			2	PCTOOLS	V5.5		1
高级打印驱动程序 (PS)			1	FOXBASE 绘图软件			4	PCSHL (PCTOOLS)	V6.0		4
9 针打印机驱动程序			1	FOXBASE+	V2.00		1	PCSHL (PCTOOLS)	V7.0		6
LP 24			1	FOXBASE+	V2.10		2	BOX (工具箱)			5
对折打印			1	R:BASE 4000 (库管理系统)	V1.11	C	12	COPYWRITE	V5.01		1
NORTON			4	MS-FORTRAN			2	检测 77 种, 清 12 种病毒			1
五笔字型			2	PE II 编辑软件			1	检测 78 种, 清 13 种病毒			1
		A	1	WORD PERFECT 编辑软件			4	检测 79 种, 清 17 种病毒			1
	V5.0	C	2	WPS (文字处理系统)	V2.0		4HD	检测 84 种, 清 24 种病毒			1
			2	WINDOWS (窗口软件)	V3.0		1	检测、清除、免疫 155 种			
五笔字型学习盘			2	MAKER (报表汉化)			3	.EXE 文件感染病毒程序			4
		单显	1	汉化 BCT 制表			1	日常事务处理及生物统计		DOS3.0	
长城笔形码输入方法		单显	1	CCED 制表编辑软件	V2.2		1	STATPAK 统计	V3.11		1
自然码		CEGA	1	高级智能制表软件			10	STATISTICS 多功能统计			2
表形码学习软件			2	AUTO HCAD (汉化计算机辅助设计)	V2.17		5	HMA STRAK	V3.11		1
CU (词语输入方法)			1	AUTO CAD			8	科研管理软件			2
HU (字处理)		A	2	ACAD (汉化计算机辅助设计)			1	日常管理软件			2
US			3	AUTO CAD			2HD	通用工资管理软件			1
XE			1	PCB			2	财务报表管理软件			1
LOTUS		A	1	OFFICE			1HD	卫生统计程序软件包			1
LOTUS (图形管理系统)			1	OFFICE			5	人事软件 (说明书 2.50 元)			2
解密软件 (专用)			1	CUICH SINTER		CEGA	2	中文干部管理档案			1
CU 激光威力加密			1				1HD	通用财会记帐、银行对帐			2
硬盘加密			1								
加密、拷贝			1								

联系地址: 中国农科院计算中心培训中心

联系人: 王路敬 胡海燕

联系电话: 89-6531-241 831-6583

附表:新一代超集成度 Z80 系列微处理器性能一览表

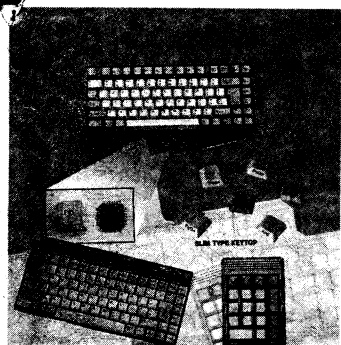
Z80 [®] 框图									
<div>84C00 CPU Power Down</div> <div>0 S C</div>	<div>84C01</div> <div>2K Bytes SRAM WSG</div>	<div>CTC SIO OSC</div> <div>PIA</div>	<div>CTC CGC</div> <div>SIO WDT</div> <div>Z80 CPU</div>	<div>PIO CGC</div> <div>SIO WDT CTC</div> <div>Z80 CPU</div>	<div>40 I/O</div> <div>CTC WDT</div> <div>Z80 CPU</div>	<div>Z80 CPU MMU</div> <div>2 DMA 2 UART 2 C/T C/Ser OSC</div>	<div>16-Bit Z80 CPU</div> <div>Z80.7.8us I/F_{acc}</div> <div>4 DMA UART 3 C/T WSG</div>	<div>CTC 16 I/O SCC/2</div> <div>Z180</div>	
型号	Z84C01	Z84C50	Z84C90	Z84013/C13	Z84015/C15	Z84011/C11	Z80180	Z80280	Z80181
功 能	Z80CPU+OSC + 电源下降方式	Z80/84C01+ 2KSRAM	多用途 I/O (KIO)	智能外设控制器	智能外设控制器	并行 I/O 控制器	高性能 Z80CPU 和 外设控制器	16 位指令与原 Z80 指令系统全部兼容 外设控制器	随机控制器
特 性	Z80CPU 2KB SRAM, 零等待状态发 生器,片内振荡 器与 Z80C00 的 DIP PLCC 封装 管脚全部兼容 EV 方式。	Z80CPU, SIO,PIO,CTC, +8 I/O	SIO,PIO,CTC, +8 I/O	Z80CPU,SIO,CTC, WDT,CGC,WSG, 开机复位,二个 附加片选控制。 EV 方式。	Z80CPU,SIO,WSG CTC,PIO,CGC, 单片完成 Z80 标准 系列功能。开机复 位,二个附加片选 控制。32 位 循环冗余检查。 EV 方式。	Z80CPU,40 I/O, CTC,WDT。 开机复位。 EV 方式。	增强型 Z80CPU, MMU 1MB, 2 个 DMA, 2 个 UART。 定时串行 I/O 端 口片内振荡器。	16 位指令,存储器 管理可达 16MB, 256B 的指令与数据 快速缓冲 CACHE, 外设控制器有四路 DMA, UART,3 个 16 位 C/T,WSG, 有与 Z80Z-BUS 的总线接口。	Z180+串行通 信控制器+16 个 I/O+4 个定 时器/计数器。 EV 方式。
速 度	6MHz, 10MHz	10MHz	8MHz, 10MHz	6MHz, 10MHz	6MHz, 10MHz	6MHz, 10MHz	6MHz, 8MHz, 10MHz	10MHz, 12MHz	10MHz
封 装	44-脚 PLCC 40-脚 QFP	40-脚 DIP 44-脚 PLCC 44-脚 QFP	84-脚 PLCC	84-脚 PLCC	100-脚 QFP	100-脚 QFP	64-脚 DIP 68-脚 PLCC 80-脚 QFP	68-脚 PLCC	100-脚 QFP
应 用	所有应用 Z80 的 场合	同左	同左	智能数据通信 控制器 智能外设控制器	智能外设控制器 调制/解调	智能并行 I/O 控制器 工业显示终端	嵌入式控制器	嵌入式控制器终端 打印机	智能外设,可 控打印机, 传真机,调制 /解调。

* 允许用 Z80 开发系统进行仿真



一九九二年

总期第83期



• ELECTRONICS AND COMPUTERS •

電子與電腦

目 录

• 综述 •

我国单片机应用技术发展趋势及展望(续) 何立民(2)

• PC 用户 •

彩色汉字通用下拉式窗口菜单的设计与实现
..... 王宏(4)

dBASE II 或 Foxbase 菜单式自由组合查询
..... 毛维平(9)

单显 PC 机的应用技巧 袁津生(10)

BASIC 程序用 P 存盘的解密方法 褚协贤(11)

谈谈编译 BASIC 在长城机上的运用 卢祥江(11)

也谈解决运行 FoxBASE 内存不够的问题
..... 周立宇(12)

• 学习机之友 •

虚拟 DOS 的改进及其在通讯网中的应用
..... 石永琳(13)

磁盘数据的压缩存储技巧 陈庆祥(14)

为 Apple DOS 创造三条实用命令 王志超(17)

也谈 CEC-I 全功能造字 覃 敏(18)

ProDOS 磁盘操作系统入门(续) 廖 凯(18)

• 6502 机器语言讲座 •

第二章 微处理器、存储器、寄存器简介
..... 朱国江(20)

• 初级程序员级软件水平考试辅导 •

计算机硬件基础知识(续) 顾育麒(24)

• 学用单片机 •

低功耗单片机最小系统 张培仁 刘振安(29)
新一代超高集成度 Z80 微处理器系列 董伯明(31)

• 学装微电脑 •

打印数字温度计 易齐干(32)

• 电脑巧开发 •

单板机 RAM 分段与程序保护的实现 王新明(37)

• 电脑游戏机 •

第三章 F BASIC 的画面控制语句 于 春(41)

• 维修经验谈 •

PC-1500 计算机的维修 何 静(44)

导电橡胶按钮被磨损的修理方法 梁绍建(44)

IBM-PC 机故障维修一例 欧阳波(44)

• 新书与软件 •

DATA BASE IV 关系数据库 张冰毅(45)

• 读者联谊 •

普及型 PC 个人用户软件交流联谊活动问题解答(二)
..... 王路敬(46)

显示器·显示卡·显示系统 孙梅英(48)

封面 键盘

封二 普及型 PC 机用户联谊会常用交流软件清单

封三 Z80 系列微处理器性能表

封底 SCB-1 单片单板机

机械电子工业部电子工业出版社主办

编辑、出版:《电子与电脑》编辑部

(北京 173 信箱 邮政编码:100036)

印刷:北京三二〇九厂

国内总发行:北京报刊发行局

国内统一刊号:CN11-2199

邮发代号:2-888

国外代号:M924

出版日期:每月 23 日

主编:王惠民 副主编:王昌铭

责任编辑:张 丽

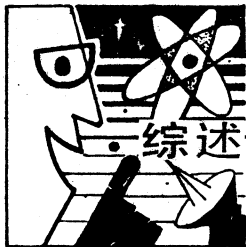
订购处:全国各地邮电局

国外总发行:中国国际图书贸易总公司

(北京 399 信箱 邮政编码 100044)

广告经营许可证:京海工商广字 147 号

定价:0.95 元



我国单片机应用技术 发展趋势及展望(续)

北京航空航天大学 706 教研室(100083) 何立民

3. 外围器件

单片机应用系统的应用开发难度会逐渐转向外部接口,因此对外部接口的新型器件的新发展、市场供货状况要密切注视。

在前向通道中,小信号、高精度、长距离传送一直是通道接口设计的难点。模拟器件价格居高不下,致使一些非快速系统转而求助于纯数字系统,相应地推动了数字化传感器、V/F 转换接口的发展。在不得不使用模拟电路的前向通道接口芯片供货状态已大为改善,市面上已有较多的小信号放大、隔离放大、可编程增益放大、电压电流转换芯片可供选择。但从根本上解决模拟小信号采集时接口通道的高精度,抗干扰只能求助于集成调理器模块。例如可用作高精度、自带激励电压源电流源的小信号放大模块 2B30/2B31。

人机通道接口中语音接口芯片引人注目,其器件价格及供货状态已能供大面积推广使用。典型的语音芯片有两类:一类兼有语音分析与语音合成功能的 UM5100/5101、T6668;另一类只有语音合成功能,但能以很低比特率合成高质量语音的 TMS5220。这些语音芯片的价格都在百元以下。

在 RS232C 接口中用 MAX232(或 ICL232)芯片替代 1488/1489 或简化接口电路,而且可省去±12V 供电电源,MAX232 芯片中有两对 RS232C 收/发电平转换接口。

反向通道中,国产带控制端的 8 路驱动芯片 BIC87 系列正在大面积推广;高压大电流 TWH8751 已成功地构成了步进电机驱动电路;国产交直流固态继电器已进入实用阶段,由于它和 TTL、HTL、CMOS 电路的兼容性,在单片机应用系统中广泛采用;现代电力电子器件如 VMOS 功率器件的引入为后向通道直接驱动大功率电路开辟了良好的前景。

为了抑制电源投切、瞬时短路、压降以及电网串入干扰脉冲造成单片机系统的失灵,在一些系统中使用了电源监视用芯片 TL7705/7700。7705 为 5V 电压监视,在电源上电时能产生可靠的复位信号,能在 500ns 内检测出电源电压降低到 4.5~4.6V 的故障现象,TL7700 的监视电压可由用户设定,故还可对温度、湿度、风量、压力等参量电压进行监测。

在某些应用系统中需要高性能实时时钟时可采用 MSM5832RS(5V 供电)或 ICM7170(3V 电池供电)。

三、单片机应用的开发环境

上海复旦大学推出了 SICE 系列开发装置已形成

我国单片机开发装置的主流机型,对我国单片机开发应用起了重要的推动作用。但制造厂家过多,质量差异较大;用户无法了解监控程序文本,造成使用中的不便。随着用户应用水平的提高,要求介入监控程序的呼声越来越高,在经过了六个年头开发历史的今天,用户对开发装置监控程序仍然一无所知,必然会影响应用开发水平的提高,也不利于监控程序的完善与提高。因此,一些有识之士已开始独立研制开发了一些单片单板机、仿真器产品,其中部分产品的软件文本、硬件电路向用户公布。较早的有武汉大学无线电信息工程系研制的 SCB-1 型单片单板机,中国科技大学计算中心研制的 KDC-III 型开发应用系统,均在《电子与电脑》杂志上公布了监控程序文本和硬件电路。

新器件的出现有利于改善开发装置的硬件电路,如大容量存储器的使用使开发装置少占用户资源,换体技术易实现零地址编程;可编程逻辑器件的使用可简化电路并实现一些特殊逻辑功能;E²PROM 及 SRAM 的可靠掉电保护的使用大大方便了单键盘调试。为了提高开发模板与系统主机的联机通信的可靠性,开始废除用分立元件构成的准 RS232C 接口,代之以 1488、1489 的标准 RS232 接口,甚至采用单 5V 供电的标准 RS232C 接口芯片 MAX232(或 ICL232)。

在软件配置上,目前国内廉价的开发装置普遍都配备有主机用的组合软件,一些较高档的开发装置已配备有 C51/96、PLM51/96 开发软件。

四、单片机应用系统设计的硬件技术

单片机应用系统硬件设计时应充分调查新器件新技术的进展,把一些行之有效的电路技术和器件及时反映到系统中。

在系统的存储器部分由于大容量芯片出现而普遍趋向单片系统,数据存储器采用可靠的掉电保护,早期用夹持电池方法已淘汰而改用焊接腿可充电电池,并加有防改写电路,较先进的结构形式则采用掉电保护 SRAM 集成模块。

抗干扰设计成为系统设计中不可缺少的部分,如电源滤波、电源监测,可靠地系统复位设置、Watchdog 电路、使用 CHMOS 芯片等。使用 CHMOS 单片机不仅本身具有较好的抗干扰能力,而且它的低功耗运行方式为软件中利用“睡眠抗干扰”创造了条件。

语音接口将会在单片机应用系统中广泛使用,具有语音合成、语音分析功能的语音芯片要求存储容量大,并有语言录入部分,故接口电路较复杂,如 T6668

语音接口。但是在许多智能仪表、监测系统、机电一体化系统中,语音词条有限,不要求接口有语音数据库功能,这时采用仅有语音合成功能的芯片如 TMS5220 较好,该芯片的语音数据压缩技术可以大量节约内存容量,可做到一个语音音节只占用 100 个字节左右(约为 T6688 接口的十分之一),在有限词条语音接口中可不单独设存储器,而采用加大单片机数据存储器容量来放置语音数据的办法。这种语音接口十分简单,但要求有现成的语音数据。目前,四川大学计算中心已有供 TMS5220 使用的语音数据库。

硬件电路的标准化、系列化、模块化是人们关心的议题。但由于新器件的发展、应用对象的复杂性,这些工作尚未提到日程上来。可以预计,其发展过程是首先优化单片机系统电路,如存储器配置、掉电保护、系统时钟、复位等;随后优化外围接口并不断筛选、定型、逐渐将难度向外扩展,先提供一部分成熟电路,不必进行试验即可一步制成印刷电路板,部分电路的 ASIC 化有利于系统电路的标准化进程。

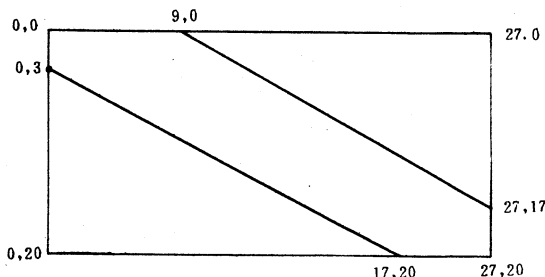
五、单片机应用程序设计技术

目前我国单片机应用系统研究人员大多数是非计算机专业人员,对软件设计比硬件系统设计更为陌生,对应用程序设计的要求、内容、作用认识较为粗浅。因此我国单片机应用程序设计尚处于初级阶段,表现在

(上接 41 页)

```
30 F. X=0 TO 27 ST. 4
40 Y=Y+1
50 DE. SP. Y, (0,1,0,0,0)=CH. (X+28)+CH.
(X+29)+CH. (X+30)+CH. (X+31)
60 SP. Y, Y*25+40, Y*20+40
70 N.
80 LOC. 0,22
```

RUN 后,返回 BG 画面。这时可按 BG GRAPHIC 方法选定一种或几种背景图形画一方框,本例以最外边



(上接第 12 页)

```
8683520      bytes available on disk
655360       bytes total memory
525456       bytes free
C:\>cd\fox
C:\fox>mfoxplus
. | \DOS\CHKDSK C:
```

以功能设计为主要内容,以指令系统为基础的随意性逻辑设计方法。虽然不少应用程序设计中引入了抗干扰设计内容,也未有系统的论述。因此,认真研究单片机应用程序设计方法,正确拟定应用程序设计内容,对提高单片机应用技术水平有极其重要的作用。

应用程序设计所选择的语言与系统的应用对象、开发环境、软件包的容量有关。单片机的具体应用对象及应用环境决定了汇编语言仍是单片机应用程序设计的基本语言;对程序量较大的监控程序可采用 C51/96、PLM51/96 以提高效率,减少编程错误。

要不断总结应用程序设计方法。单片机应用系统的多样性不可能攀求一种全通用的方法;但目前可以逐步推行一些行之有效的设计方法。如实时多任务操作系统设计方法、智能仪表中监控程序的状态分析法和图解设计法。

要正确拟定应用程序设计的内容与要求,改变以功能实现为唯一目的的设计思想,明确应用程序设计中功能实现、容错设计、抗干扰设计是三位一体不可缺少的内容。要保证应用系统可靠地运行,应用程序完成功能实现只是开始,只有具备抗干扰能力、容错能力的应用程序才有可能在实际环境中使用。

(编者按:本文摘自“1991 年全国单片机学术交流会议论文集”,本刊略有增删。)

缘画框,并在框内加两条斜线,画面如下:

然后键入 ESC 和 STOP 返回 BASIC 后,把 10 行中的 BGTOOL 改为 VIEW, RUN 后,则显示丽莎在方框斜线包围的六边形中。

3. BG、SP 画面的配合

读者已经发现, BG 画面的 X 取值为 0~27, Y 取值为 0~20, 而 SP 画面的 X 取值一般为 0~240, Y 取值为 0~220。两个画面中的值域对应关系见随机手册附图。由于取值区域的差异,两个画面的座标必须通过换算才能正确配合。其座标换算方程。

$$\begin{cases} x = X * 8 + 16 \\ y = Y * 8 + 24 \end{cases}$$

式中大写 X, Y 为 BG 画面座标,在 BG 画面左下角显示。小写字母 x, y 为 SP 画面座标。使用方程(1)可在设计 SP 显示座标时进行计算。有时已知 SP 的座标,要根据 SP 的座标设计背景图案,则可使用方程(2)进行换算。

$$\begin{cases} X = x/8 - 2 \\ Y = y/8 - 3 \end{cases} \quad (2) \quad (\text{待续})$$

```
21309440     bytes total disk space
55296        bytes in 2 hidden files
43008        bytes in 10 directories
12527616     bytes in 994 user files
8683520      bytes available on disk
655360       bytes total memory
150704       bytes free
```



PC 用户

彩色汉字通用下拉式窗口菜单的设计与实现

邮电部成都电缆厂科研所(610042) 王宏

下拉式窗口菜单是时下流行的一种用户接口方式。许多西文软件都采用了多彩色的窗口菜单技术,提供集成化用户环境,使用户操作非常方便。它们具有新的风格,如选择菜单功能时采用光棒定位,以回车键或数字(或菜单内容首字母)选择相结合的方式。这种方法使用户感到直观、明快、不易产生误操作。多层次菜单的进入和退出也由原来逐层清屏、显示下层菜单的方法,变为保留主菜单画面,子菜单采用下拉式结构,层次分明,用户不致于因进入太多而如入迷宫。

本文以汉化 Clipper Summer'87 为工具,在 25 行汉字系统下如何实现彩色汉字下拉式菜单窗口问题加以阐述并提供完整的源程序。它不但可以通用于任何管理系统,而且兼管整个管理系统的所有菜单分支及分支选择处理。具有良好的通用性和可维护性。

1. 窗口菜单设计的基本原理

窗口一般是在屏幕上开辟一个矩形,它可以根据命令打开和关闭。当一个窗口打开时,它隐藏了将被覆盖的屏幕内容,包括与其它窗口重叠的部分,正文或图画输出只对窗口内部有效,对窗口外的任何输出都是无效的。窗口关闭时,则要清除它在屏幕上的信息。窗口菜单是窗口用于菜单的一种具体形式。实现窗口菜单流程如图 1 所示。它反映了窗口菜单建立的基本思想和主要步骤。

2. 彩色汉字下拉式窗口菜单的实现

①窗口菜单状态数据的数据结构

一般的菜单都是多层次的。最先出现的菜单往往最后关闭,而且每一时刻,处于重叠式窗口最上面的最新开辟的窗口为活跃窗,菜单选择只能在活跃窗内进行。为了实现这样一种先进后出的存储信息要求,本文采用了菜单库存放有关窗口位置、色彩等状态数据,使程序中不存有菜单内容,而把所有的菜单内容连同功能模块等参数事先存入一个反映各功能模块间关系的菜单库中。菜单库描述信息见图 2。

这样,建立系统窗口菜单状态等参数工作,实际上变成了让用户向菜单库中输入菜单内容等参数(可用程序自动实现),然后按一定规律将这些菜单内容连接起来,显示在屏幕相应位置即可。这样菜单的显示及选择处理过程就是遍历树的过程,其程序框图见图 3。

②窗口菜单的覆盖部分屏幕信息的保存与恢复

Clipper Summer'87 提供了两个函数,可将屏幕信息保存到内存变量或从内存变量中恢复,这样使窗口

菜单的覆盖部分的屏幕区域信息的保存与恢复变得轻而易举了。

储存屏幕函数为 Savescreen(),其格式为:

Savescreen(<expN1>,<expN2>,<expN3>,<expN4>)

其中:<expN1...expN4>是所要储存屏幕区域的坐标位置。系统返回一个字符串用于保存所指定的屏幕区域,最多保存 4000 个字节。

恢复屏幕区域函数是 Restscreen(),其格式:

Restscreen (<expN1>,<expN2>,<expN3>,<expN4>,<expC>)

它用于将事先用 Savescreen()函数保存屏幕区域的信息从内存变量中恢复到指定屏幕区域。恢复的屏幕位置可与原先储存的原始位置相同或不同。若是指定一个新的屏幕位置恢复时,应确保此位置与原储存位置有相同的大小,否则将会得到不明确的结果。而且不要使用 Restore screen 命令恢复用 Savescreen()储存的屏幕区域,否则也会得到不明确的结果。另外要注意,Savescreen()与 Restscreen()均不支持使用 ANSI·OBJ 终端机支援的电脑或应用程序。也就是说,在 DOS 的 CONFIG. SYS 文件中不应有语句:

DEVICE=ANSI·SYS

③窗口菜单的显示及选择

在进行用户界面设计时,一般都是将主菜单放在屏幕最上行或最下行,这样使整个屏幕画面空间集中。主菜单采用固定方式显示在屏幕上,用左右键头移动光棒定位,按回车键或数字键(或首字母)选择。子菜单的选择则依赖菜单库中描述参数,根据主菜单选择的某一功能,从菜单库中提取相应参数,显示出下一级子菜单。菜单布局采用下拉式显示(或重叠子菜单,依菜单库中描述参数而定),见图 4,图中位于主菜单块一的下面有一矩形框,它是当前第一个菜单内容被选中后出现的下一级菜单,当功能完成返回本菜单时,这时再按 Esc 键返回上一级菜单,同时该子菜单窗口消失。每选中一个菜单(若没到叶结点),就在其下方出现下一级菜单的窗口,它就是所谓“下拉式菜单”。

由于子菜单窗口是随主菜单中各功能的位置不同而出现在屏幕的不同地方,且各自的下一级菜单内容条目都不相同,而控制光棒移动却与行列起始位置及范围密切相关,故对每一个不同位置的下拉式菜单,均要编写不同的程序来实现。本文采用了统一的显示位置,即不管主菜单选择某功能,下拉菜单均从屏幕最左

边显示开始,并根据菜单库中描述参数,采用程序递归设计技术解决上述因位置和条目变化引起的困难。这是个非常方便实用的菜单选择程序。

为了使菜单显得醒目,光棒所在位置采用了色彩显示,光棒通过↑、↓键移动,且可实现自动翻转功能(本文介绍的程序未做,用户可根据需要用 Clipper 提供的函数 Scroll()实现),按 Esc 键则返回。由于 Clipper 不支持类似 dBASE III plus 中的 Return to Master 指令,本程序采用了将键值填入键盘缓冲区的方法来模拟 Return to Master,使用户也可从最下级菜单直接返回主菜单画面。菜单选择后,其状态保留,便于操作人员记忆操作流程。整个菜单选择给人一种不疲劳之感觉。窗口宽度随菜单项中最大长度项而变,使得菜单窗口实际上是个可自动伸缩的窗口。

以上详细介绍了利用汉化 Clipper 实现的彩色窗口菜单方法。该程序具有良好的通用性和可维护性。对

任意层次结构的应用管理系统,只要填好菜单库中描述参数,主菜单程序中提示作下修改,系统菜单内容,功能模块扩充、删除、修改等均不需对程序作任何改动,只改动菜单库内容等参数即可。在系统维护模块中,也可加上对菜单库的修改功能。笔者用本文方法实现的彩色窗口下拉式菜单程序已经在“全国职业卫生信息管理系统”中运用,经在配有 EGA 卡的 AST386 上使用效果良好。本程序稍作改动也可在 dBASE III plus、FoxBASE+ 上实现。本程序用 Clipper 提供的编译、连接程序生成了可执行文件,在 DOS 下即可执行。

数据库结构:		C:\CLIPPER\MENU.DBF		
数据记录数:		589		
最新更改日期:		07/31/91		
字段	字段名	类型	宽度	注释
1	MENU_NAME	C	20	菜单项
2	MENU_MARK	C	8	菜单标记
3	MENU_SUB	C	8	下级菜单标记或功能程序名
4	NUMBER	C	2	同层菜单序号
5	UP_MENU	C	8	上级菜单标记
6	XROW	N	2	下拉菜单行坐标
7	YCOL	N	2	下拉菜单列坐标
8	COLOR	C	1	下拉菜单色彩
9	NOTE	C	40	下拉菜单条目注释

图 2

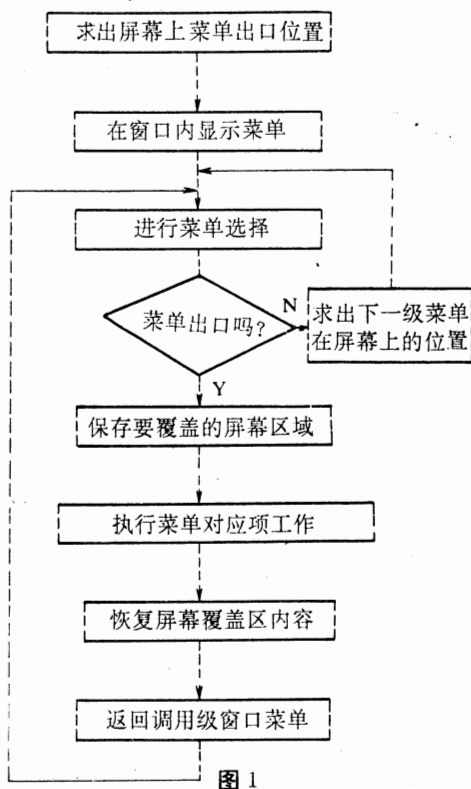


图 1

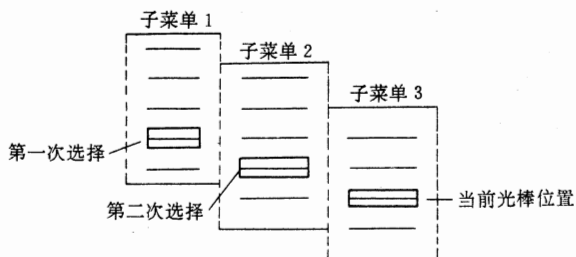


图 4

示例库

记录号	MENU_NAME	MENU_MARK	MENU_SUB	NUMBER	UP_MENU	XROW	YCOL	COLOR	NOTE
1	图形查询与录入	29		1	1	5	1	2	图形查询与录入
2	图形生成	29		2					图形生成
3	简史及一般情况	29		3					厂(矿)简史及一般情况查询
4	毒害种类及构成	29		4					厂(矿)毒害种类及构成查询
5	构成及分布	29		5					厂(矿)部门构成及毒害分布查询
6	个人档案	29	35	6					职工职业卫生个人档案查询
7	测定情况	29		7					厂(矿)毒害测定情况查询
8	体检情况	29		8					厂(矿)职工体检情况查询
9	历史资料	29		9					职工职业卫生历史资料查询

10	一般情况及各种史	35	96	1	29	6	18	3	职工一般情况及各种史查询
11	事故及各种疾病	35		2					事故及各种疾病查询
12	姓名	36	MODEL1	1	96	8	64	6	按姓名查询
13	年龄	36	MODEL2	2					按年龄查询
14	性别	36	MODEL3	3					按性别查询
15	民族	36	MODEL4	4					按民族查询
16	技术职称	36	MODEL5	5					按技术职称查询
17	任意条件	36	MODEL6	6					按任意条件查询
18	浏览	36	MODEL7	7					浏览全部
19	一般情况及嗜好史	96	36	1	9	7	46	5	一般情况及个人嗜好史查询
20	既往病史	96		2					既往病史查询
21	配偶史	96		3					配偶史查询
22	家族疾病史	96		4					家族疾病史查询
23	月经生育史	96		5					月经生育史查询
24	职业史	96		6					职业史查询

* MAIN_MENU.PRG

* 功能:主菜单程序

* 调用文件:SUB_MENU.PRG

CLEAR ALL

SET KEY-1 TO K_BOARD && 设定按 F2 键后执行的程序

SET SAFE OFF

SET DELI TO ' '

SET DELE ON

SET DECI TO 4

SET DEVI TO SCRE

SET SCOR OFF

SET STAT OFF

SET WRAP ON

SET CURSOR OFF

CLOSE DATA

SET CONS ON

PUBLIC LEVEL,V_N,W_RETU

SET COLO TO G+/N

CLEAR

V_FU1='1. 录入'

V_FU2='2. 查询'

V_FU3='3. 统计汇总'

V_FU4='4. 图表输出'

V_FU5='5. 维护通讯'

V_FU6='6. 综合评价'

V_FU7='7. 辅助功能'

DIS1='系统数据库的数据输入'

DIS2='查询系统数据库中数据'

DIS3='系统报表的预处理'

DIS4='报表打印及屏幕上图示显示'

DIS5='系统维护及与其它系统的接口'

DIS6='厂(矿)工业卫生的综合评价'

DIS7='做系统辅助工作'

TS1='再见! 欢迎下次使用!'

TS2='注意:'

TS3='关机顺序:先关主机,后关外设'

TS4='开机顺序同关机相反.'

TS5='→和←键:移动光带,ESC 键:退出系统,ENTER 键或数字:选中,其它键无效'

TS6='↑或↓键:移动光带,ENTER 键:选中,HOME 或 END 键:第一个或最后一个光带,按 F2 键回主菜单'

SET COLO TO W+/N

CLEAR

SELE1

* 打开菜单库

USE MENU

* 保存屏幕区域信息

WINBUFF=SAVESCREEN(0,0,24,79)

DO WHILE .T.

V_N=1 && 设定菜单层数

W_RETU=.T. && 判定用户按 F2 键否

SET COLO TO GR+/B

@ 0,0,2,79 BOX '+-+|+-+|'

@ 2,0,4,79 BOX '+-+|+-+|'

@ 4,0,21,79 BOX '+-+|+-+|'

@ 21,0,23,79 BOX '+-+|+-+|'

@ 5,1 CLEA TO 20,78

SET COLO TO G+/B,N+/R,BG

@ 1,1 CLEA TO 1,78

@ 1,CENTER(TS5) SAY TS5

SET COLO TO R+/B

@ 22,5 SAY '工业卫生微机管理系统'

@ 22,50 SAY '研制:中华人民共和国卫生部'

SET COLO TO G+/B

@ 22,29 SAY '|| 版权所有 ||'

SET COLO TO GR+/N

@ 3,1 CLEA TO 3,78

&& 菜单条目注释显示在第 24 行且居屏幕中央

SET MESS TO 24 CENTER

@ 3,1 PROM V_FU1 MESS DIS1

@ 3,10 PROM V_FU2 MESS DIS2

@ 3,20 PROM V_FU3 MESS DIS3

@ 3,33 PROM V_FU4 MESS DIS4

@ 3,45 PROM V_FU5 MESS DIS5

```

@ 3,57 PROM V_FU6 MESS DIS6
@ 3,69 PROM V_FU7 MESS DIS7
&& 设定画面执行目前 PROMPT 的光棒,将选择结果
存入 MENUCHOICE
MENU TO MENUCHOICE
DO CASE
CASE MENUCHOICE=1
LEVEL='2'
CASE MENUCHOICE=2
LEVEL='29'
CASE MENUCHOICE=3
LEVEL='13'
CASE MENUCHOICE=4
LEVEL='78'
CASE MENUCHOICE=5
LEVEL='83'
CASE MENUCHOICE=6
LEVEL='123'
CASE MENUCHOICE=7
LEVEL='124'
CASE MENUCHOICE=0
SET COLO TO BG+/G
@ 7,30 CLEA TO 9,50
V_RE='N'
@ 8,35 SAY '退出否? (Y/N)'GET V_RE PICT'
X'
READ
IF UPPE(V_RE)='Y'
CLEAR
SET COLO TO
SET COLO TO GR+/G
@ 5,CENTER(TS1) SAY TS1
@ 8,CENTER(TS2) SAY TS2
@ 11,CENTER(TS3) SAY TS3
@ 14,CENTER(TS4) SAY TS4
CLOSE ALL
QUIT
ELSE && 恢复屏幕区域信息并循环
RESTSCREEN(0,0,24,79,WINBUFF)
LOOP
ENDI
OTHE
RESTSCREEN(0,0,24,79,WINBUFF)
LOOP
ENDC
SET MESS TO
@ 1,1 CLEA TO 1,78
IF LEN(TRIM(LEVEL))<>0
DO SUB_MENU
ENDIF
RESTSCREEN(0,0,24,79,WINBUFF)
ENDDO
RETU

```

```

* 文件名:SUB_MENU.PRG
* 功能: 子菜单程序
* 调用文件:SUB_MENU.PRG
PROC SUB_MENU
PRIV MEN_N
MEN_N=LEVEL
SET COLO TO GR+/B
DO WHILE .T.
GO TOP
LOCA ALL FOR MENU_MARK=MEN_N
IF EOF()
SELE 10
WINBUFF1=SAVESCREEN(5,2,20,78)
DO &LEVEL && 调用功能程序执行
@ 5,1 CLEAR TO 20,78
SELE 1
MEN_N=CA_NAME
RESTSCREEN(5,2,20,78,M-)WINBUFF1)
V_N=V_N+1
V_M=STR(V_N,1)
K=K&V_M
LOOP
ELSE
V_ROW=XROW && 下拉菜单屏幕行坐标
V_COL=YCOL && 下拉菜单屏幕列坐标
V_TEMP=V_ROW
V_LEN=0 && 初始化同层菜单项最大项长度
MENU_NUM=1 && 初始化同层菜单数目
CO=COLOR && 下拉菜单色彩
UP_NAME=UP_MENU && 保存上层菜单标记
DO WHILE MENU_MARK=MEN_N
V_LEN=MAX(LEN(TRIM(MENU_NAME)),V_LEN)
IF MENU_NUM<0
M=STR(MENU_NUM,1)
ELSE
M=STR(MENU_NUM,2)
ENDIF
M_NAME&M=TRIM(MENU_NAME)
MENU_NUM=MENU_NUM+1
INFO&M=TRIM(NOTE)
SKIP
ENDDO
MENU_NUM=MENU_NUM-1
SET COLO TO GR+/&CO
@ V_ROW,V_COL CLEA TO V_ROW+MENU_NUM+1,V_COL+V_LEN+1
@ V_ROW,V_COL,V_ROW+MENU_NUM+

```



```

1,V_COL+V_LEN+1 BOX '++-+|+-+|'
SET MESS TO 24
@ 24,0 CLEA TO 24,79
N=1
DO WHILE N<=MENU_NUM    && 显示菜单
    IF N<10
        M=STR(N,1)
    ELSE
        M=STR(N,2)
    ENDIF
    @ V_ROW+1,V_COL+1 PROM M _
    NAME&M MESS INFO&M
    N=N+1
    V_ROW=V_ROW+1
ENDDO
@ 1,CENTER(TS6) SAY TS6
@ 24,63 SAY '[ESC]返回上级菜单'
MENU TO K
GO TOP
LOCA FOR MENU_MARK=MEN_N.AND.NUM-
BER=LTRIM(STR(K))
LEVEL=MENU_SUB    && 下级菜单标记或功能
程序名
IF LEVEL+' '<>' '.AND.K<>0    && 判
断叶结

```

```

点。
CA_NAME=MENU_MARK    && 保存本层菜
单标记
V_M=STR(V_N,1)
K&V_M=K    && 保存本层菜单选择光棒位置
RELEASE K
V_N=V_N+1
DO SUB_MENU    && 递归调用
ENDIF
MEN_N=UP_NAME
IF MEN_N='1'    && 若第一级下拉子
菜单返主画面
    @ 24,0 CLEA TO 24,79
    RETU
ELSE    && 返上级菜单
    V_N=V_N-1
    V_M=STR(V_N,1)
    K=K&V_M
    SET COLO TO G+/B,N+/R,BG
    @ V_TEMP,V_COL CLEA TO V_ROW+1,V
    _LEN+1
    IF W_RETU=.F..AND.V_N< >0
        KEYBOARD CHR(27)
    
```

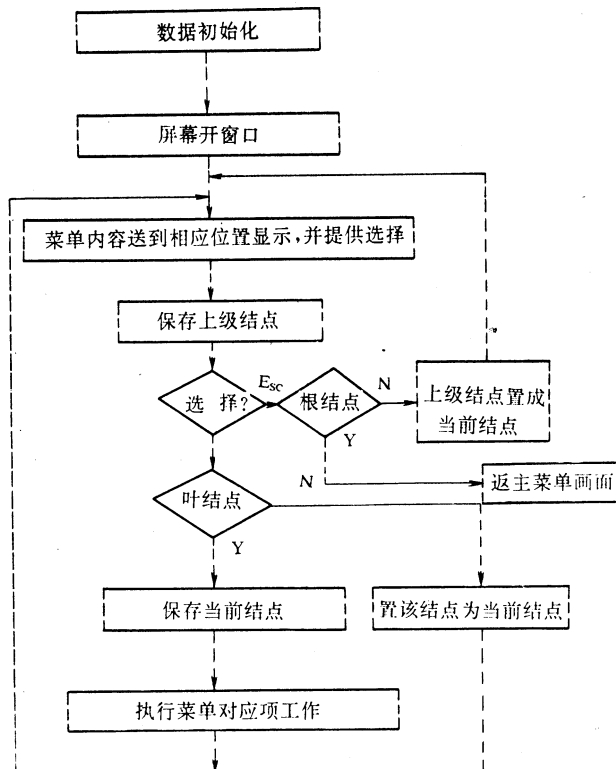


图 3

(下转第 15 页)


```

clear
use
cancel
case i=121
p=val(p)
do case
case y=39
p=p+4
case y=56
p=p+8
endcase
p=ltrim(trim(str(p)))
cx='a'+ '&p'
x1=&cx
hh=hh+x1
s1=ltrim(trim(&cx))
@13,s say s1
s=s+len(s1)
x=6
p='1'
case i=27
@13, s get s2
read
@13,s say s2
s2=trim(s2)
s=s+len(s2)
hh=hh+s2
s2=' '
case i=99
s=12
x=6
y=22
hh=' '
p='1'
s2=' '
clear
loop
case i=13
use abc
locate all for &hh
if eof()
@18,20 say '无此条件的记录'
wait
else
clear
disp all for &hh
wait
hh=' '
s2=' '
x=6
y=22
p='1'
s=12
clear
endif

```

```

endcase
enddo

```

* 出错处理 filename is ERROR. PRG

```

clear
i=' '
@13,20 say '语法或逻辑组合或字符未加引号'
@15,28 say '按任意键重来' get i
clear typeahead
read
do typ

```

单显 PC 机的 应用技巧

河北廊坊 103 信箱 9 分箱(102849)袁津生

自引进 PC 机以来,配有单色图形卡的 PC 机及其兼容机在我国已为数不少。这种机器具有价格比高、字形优美及在单色汉字下显示行数多等优点,很快占领了市场,其中一部分已进入家庭。但有几点不足:

1. 为这种机器而写的游戏软件很少。
2. 在 BASICA.COM 软件中没有图形的功能。
3. 不能运用中分辨率的 CCDOS。
4. 在单色汉字下,不具有词组的功能。

基于上述原因使这种机器的应用受到限制。因此对于单位及家庭来说希望能更多地开发出这种机器的功能。下面介绍一些改进措施。

一、仿真 CGA 的功能

目前还没有一种方法可解决所有单色图形卡仿真 CGA 的图形功能,笔者在多次实验中发现了这一方法。在一些游戏软件中有一个用于单色图形显示的程序 MONO.COM,可将此程序稍加修改即可使单色图形卡具有仿真 CGA 的功能。改动如下:

```

A>DEBUG MONO.COM<CR>
-E 197
4E0D:0197 02.06<CR>
-W <CR>
Writing 026F bytes
-Q <CR>

```

其中<CR>表示回车。若将此程序加在自动批文件 AUTOEXEC.BAT 中,开机后即可完成对 CGA 的仿真。也可在 PC-DOS 操作系统下真键入 MONO<CR>这样在仿真 CGA 下可运行游戏软件、可使 BASICA.COM 软件具有作图功能、可进入 CCDOS2.0 等中分辨率汉字操作系统,并具有图形美观、字形优美等特点。

二、给单色汉字系统加上词组功能

CCDOS 2.13H 以其优越的性能受到广大用户的

欢迎,在此系统中有一外部词组功能,可用字处理软件(WS 或 EDLIN)编辑外部词组,然后用 CZP.EXE 来建立词组文件,如 CZ2.COM。我们可将此文件拿来加入单色汉字启动的批命令中:

```
A>TYPE C.BAT
ECHO OFF
CLS
ECHO ccdos is loading.....
HGC FULL
FILE1
CCCC
CZ2
```

在启动单色汉字的批命令后,将词组调入内存。调用时先键入词组的编码,再按分号键即可。

BASIC 程序用 P 存盘的解密方法

广东恩平县供电局(529400)禡协贤

BASIC 程序编辑完成之后,如果存盘命令选用 SAVE“<文件名>”,P,程序将按加密二进制码格式保存到磁盘中去,这就是程序加密保护。经过加密的程序,用户将无法用 LIST 命令显示程序清单,也无法对其进行修改。本文介绍一种非常简单的 BASIC 程序的解密方法,本解密程序内容只有两个字节,编制、使用方便,任何可以上机操作的人都能轻易完成。

一、解密程序的编辑

解密程序是在 DOS 状态下,按二个步骤编辑。

首先建立一个名为“P.BAS”的空文件,操作如下:

```
A>COPY CON:P.BAS<CR>      建立文件名
- <CR>                      空一行
- ^ Z 或[F6]<CR>           存盘
```

以上操作可以在任意当前工作盘上进行。到此,若用 DIR 命令查当前工作盘,可发现有一个名为 P.BAS 的文件,长度为 2 字节。

第二步用 DEBUG 命令修改上述文件,操作如下:

```
A>DEBUG P.BAS<CR>      装入 P.BAS 程序。
                        如 DEBUG 文件不在 A
                        盘上,DEBUG 命令前要
                        加上所在的盘符。
```

```
-D CS:100 101<CR>      显示偏移地址 100~
                        101 的内容
```

```
239C:0100 0D 0A      (内容为 0D 0A)
-E100<CR>            修改偏移地址为 100 的内容
239C:0100 0D FF<CR>   将 0D 改为 FF
-E101<CR>            修改偏移地址为 101 的内容
```

```
239C:0101 0A 1A<CR>   将 0A 改为 1A
-W<CR>                写盘
```

```
Writing 0002 bytes
```

```
-Q<CR>                退出修改
```

解密程序已编辑完成,其内容就是:FF 1A.

二、使用方法

系统进入 BASIC 状态后,第一步先用 LOAD“<文件名>”装入 P 加密程序到内存。(此时,如用 LIST 命令显示程序,会出错:“Illegal function call”).

第二步再用 LOAD “P”命令,将解密程序调入内存,经过第二步聚后,原先装入内存中的被加密程序已被解开,用 LIST 命令即可以显示该程序的全部内容,并可以进行任意修改编辑。

编辑完后,若再用不带 P 参数的 SAVE 命令重新存盘,即可得到一个未加密的 BASIC 源程序文件。

谈谈编译 BASIC 在 长城机上的运用

江苏南京军区工程兵部(210016)卢祥江

用过的同志都知道,编译 BASIC 是一种非常受欢迎的语言。经过编译的 BASIC 程序有运行速度快、节省存储空间、保密性强、通用性好、其它语言调用方便等优点。但目前的编译 BASIC 版本也有其缺点,那就是在长城系列机上使用时(其它系列机上有),有的 BASIC 命令编译后无法执行,主要是画图语句,如 LINE,PAINT,CIRCLE 等。为了解决这个问题,我们用长城机上配套的 GRD.SYS 系统图形驱动文件中的同类命令来代替,取得了圆满成功。譬如用:

```
OPEN "GRP" FOR OUTPUT AS #1
WRITE #1,"L",C,X1,Y1,X2,Y2
WRITE #1,"B",C,A,X1,Y1,X2,Y2
WRITE #1,"C",C,X0,Y0,R
WRITE #1,"P",X1,Y1,C1,C2
CLOSE #1
分别代替 BASIC 语言中的:
LINE (X1,Y1)-(X2,Y2),C
LINE (X1,Y1)-(X2,Y2),C,B
CIRCLE (X0,Y0),R,C,,1
PAINT (X1,Y1),C1,C2
```

但在使用时要注意:

1. BASIC 程序中有有的画图语句用的是相对坐标,用 GRD.SYS 时要换算成绝对坐标。

2. 使用 GRD.SYS 时,必须要用有 GRD.SYS 和 CONFIG.SYS 文件的硬盘或软盘启动微机,并且 CON-

FIG. SYS 中必须要有 DEVICE=GRD. SYS 这条命令。

3. 在程序中使用, 每屏显示要先打开文件, 即有:

OPEN "GRP" FOR OUTPUT AS #1

这样一条命令, 每屏结束时要有 CLOSE #1 关闭文件命令。请注意: 每屏显示结束时一定要有 CLOSE #1 语句, 否则将会出现最后的几句命令暂不执行的现象, 即最后的几笔不画, 清屏后再画出来。

4. 把 BASIC 源程序编译成 EXE 文件有两种方法, 并各有其特点。

第一种是在编译时使用 BASCOM. LIB 库文件, 编译成的 EXE 文件不要其它文件的支持就可独立运行, 但编译后的 EXE 文件较长, 单个文件占用磁盘空间较大。如: 1K 左右的源程序编译成 EXE 文件约有 19K。

第二种是在编译时使用 BASRUN. LIB 库文件, 编译后的 EXE 文件需要 BASRUN. EXE 的支持才能运行, 即在运行编译成的 EXE 文件的当前盘中必须要有 BASRUN. EXE 文件。但这样编译成的 EXE 文件较短。如: 1K 左右的源程序编译成 EXE 文件约有 2K。

如从运行方便和节约内存考虑可采用第一种方法; 如从节约磁盘空间角度考虑可采用第二种方法。

5. 在用高分辨率显示时, 屏幕设置语句解释 BASIC 与编译 BASIC 有所不同, 用解释 BASIC 调试好的程序经过编译后功能会发生变化。如清屏功能, 用解释 BASIC 清屏很正常的程序, 经过编译后有可能清不掉。经过我们试用, 一般开头要用 SCREEN 0, 清屏或者结束退出时用 CLS; SCREEN 2; CLS; SCREEN 0。如不加 SCREEN 2; CLS 有可能会使字符清除了而画的图没清除。也可以利用该功能, 退出系统, 仍保留图形为其它语言提供使用。譬如: 用该功能在屏幕上画出并保留表格线, 而用其它语言(如 dBASE III 等)填写字符, 就形成了表格线, 其优点是表格线不占字符位置, 且用 dBASE III 语言中的清屏命令也不会清除掉, 可以反复使用, 待用完后再用 BASIC 的清屏程序清除掉。该方法适合中国人的使用习惯, 可以大大方便用户, 可以用这种方法编制出实用性很强的软件。我们有一个软件用了这种方法, 受到了用户的普遍欢迎和很高评价。

6. 请使用者注意, 解释 BASIC 的源程序在要编译时, 源程序必须要用 ASCII 码的形式存盘(即用: SAVE "文件名", A 命令存盘), 否则, 编译时会提示错误信息 "0 SEVERE ERROR(S)"。

注释: 1. #1 可以是 #2、#3……

2. X1, Y1 为起点坐标, X2, Y2 为终点坐标, X0, Y0 为圆心坐标, C, C1, C2 为颜色, R 为半径, A 为实心或空心。

也谈解决运行 FoxBASE 内存不够的问题

江苏徐州邮电局(221000)周立宇

本刊 91 年第二期发表的“如何解决运行 FoxBASE 内存不够的问题”, 介绍了取消二级字库的办法来减少汉字驻留 RAM 空间, 以解决 FoxBASE 运行空间不足的问题。本人经过实践, 发现有更为简单有效的解决方法, 其主导思想是: 不把汉字装入 640K 内存, 以达到节省空间的目的, 而不必删除二级字库, 以方便使用。

目前, 市面上经常看到的 CCDOS4.0 和 CCDOS2.13(A~H)都提供了把汉字安装在硬盘上的处理方法。这不仅使得有足够的内存运行 FoxBASE, 还可有足够内存运行其它有用的软件, 如 Watchman 等病毒检测程序和 PC-Cache 等辅助提高处理速度的软件, 可谓一举多得。

以 CCDOS2.13H 为例处理步骤如下:

1. 把 CCDOS2.13H 的第七号盘装入 A 驱

A>ZK

按屏幕提示分别将 CCDOS2.13H 的内容装入系统。

2. 建立新的批处理 AUTOEXEC. BAT

C>\dos\edlin autoexe. bat

New file

* i

1: * Echo off

2: * cls

3: * cd \213

4: * file0a82

5: * cccc

6: * cc11(或 cc25)

7: * int10f

8: * yx1

9: * prt

10: * file16b

11: * file24a 1sfhk

12: * zf24 3

13: * path c:\; c:\213;

14: * key

15: * cd \

16: * prompt ¥p¥g

17: * ^ Z

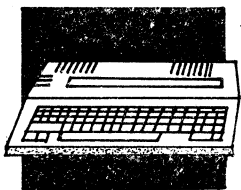
* e

3. 重新开机启动

C:\>\DOS\CHKDSK C:

21309440	bytes total disk space
55296	bytes in 2 hidden files
43008	bytes in 10 directories
12527616	bytes in 994 user files

(下转第 3 页)



学习机之友

虚拟 DOS 的改进及其在通信网中的应用

甘肃临洮农校(730500) 石永琳

本刊 1991 年 7 期刊登的《为无驱动器系统设置虚拟磁盘》一文,为中华学习机的无驱动器用户提供了一个相当于 DOS3.3 的运行环境,减轻了频繁操作录音机的麻烦。此外,虚拟磁盘操作系统还能解决在无驱动器配置的机器上,进行 DOS3.3 操作系统的教学和实习问题。鉴于该文提供的程序有较大的实用价值,笔者现就其中的不完善之处作些补充。

原文程序中有一处排版错误需要更正:\$BDC8 单元应为 \$BD。

一、虚拟盘数据存储、恢复方法的改进

原文给出的数据备份和恢复方法对“小蜜蜂—I(XMF—I)中华机是正确的,只是操作有点烦琐,备份数据时需要分两步完成:

(1)键入 CALL 48557/

(2)用→键扫描屏幕显示的 BSAVE “XNP”,A \$3FFD,L\$4057 命令,回车执行。

恢复数据时执行 BRUN“XNP”命令从磁带读数据。

由于 CEC—I 等苹果兼容机 BASIC 语言无 BSAVE、BRUN 命令,只能在监控状态下存储和恢复数据,为这些机型修改的程序执行时将显示一条监控命令 3FFD.8054W,不进入监控就不能执行。恢复数据时也要经过 CALL—151,3FFD·8054R,3FFDG 等步骤。

针对原文存储、恢复数据操作烦琐的问题,笔者对原程序进行了分析,找到了各种机型通用的数据备份和恢复方法,并通过以下两方面实现。

一是重新编制了数据备份和恢复程序,安排在原程序中 \$BDF7~\$BE99 这一区域,程序附后。

二是修改了两条 DOS 命令,使之适宜于在虚拟盘系统中使用。具体修改过程是在监控状态下键入 A8EF:49┐CE┐4F┐55┐D4┐N 9D4A;19┐BE┐AC┐BD。现在,DOS 就有了两条新命令——OUT 和 IN。其中 OUT 命令用来备份数据,把“盘”中数据写入磁带,IN 命令从磁带读回数据,原 DOS 命令 FP、INT 取消。

改进后的程序,键入 OUT 命令,机器显示 Press any key to SAVE !! 按下录音机录音键,依照提示敲任一键即开始存储数据,约 2 分钟时间全“盘”数据存入磁带。存储开始和结束时机器响一声“嘟”。

键入 IN 命令时,机器显示 Loading... 字样,表示正在等待读入数据,按录音机放音键,读带完毕即可恢复“盘”中原来数据。

二、在简易通信网上运行虚拟磁盘操作系统

把中华学习机的录音机输入输出 EAR、MIC 用屏蔽线交叉连接,就构成能实现机器间通信的简易网络。下面介绍利用简易网运行虚拟磁盘操作系统的方法。这将使您的简易网实现比原来更高一级的资源共享。要求简易网中至少有一台机器配置驱动器。

1. 建立系统程序

在配有驱动器的机器上照原文及本文的改进办法建立系统程序。CALL—151 进入监控,键入 6CFD<9CFD.BFFFM,把程序搬出 DOS 区,退出监控,用 PR#6 命令重新启动 DOS 3.3,键入 BSAVE RAMDOS,A \$6CFD,L\$2303。至此,磁盘上已建立了名为 RAMDOS 的系统程序。

2. 通过简易网为无驱动器机器设置虚拟盘

为了叙述方便起见,把配置驱动器的一台机器称为“主机”,网上其它机器为“分机”。用主机为分机设置虚拟盘的步骤如下:

(1)主机:在 DOS 下键入 BLOAD RAMDOS,将系统程序从磁盘上调入主机。

(2)分机:监控下键入 9CFD.BFFF R 读程序;

主机:监控下键入 6CFD.8FFFW 发送程序。

(3)分机:从首址 \$9CFD 或 \$BD00 处执行程序,机器的 16K 卡即被设置为虚拟盘,可用 DOS3.3 命令对其进行管理了。

可见,这种方法要比从磁带读入程序设置虚拟盘方便可靠。

3. 虚拟盘与主机磁盘的通信

分机“盘”中数据在主机磁盘上建立备份:

(1)主机:监控下执行 5000.8FFFR;

分机:键入 OUT 命令,之后按任一键。

(2)主机:退出监控返回 DOS 系统。键入 BSAVE RAMDISK n,A \$5000,L\$4000,即将虚拟盘中的数据存入磁盘。n 是为了防止文件重名造成覆盖对文件的编号。

恢复虚拟盘中的数据:

(1)主机:BLOAD RAMDISK n 读数据,注意文件编号与存入时一致。

(2)分机:键入 IN 命令

主机:监控下执行 5000.8FFFW 传送数据至分机。

由于条件限制,中小学校计算机室大部分没有为每台机器配置磁盘驱动器。利用简易通信网设置虚拟盘的办法,不仅能满足学习 DOS 3.3 的需要,也将使每台机器都享受到磁盘驱动器的便利。

事实上,即使配置有驱动器,把 16K 卡设置为虚拟盘也是有意义的,这样可以减少对磁盘的访问,有利于保护机器节省时间。

BDF7— 2C
BDF8— 10 C0 2C 00 C0 10 FB 20
BE00— 3A FF 20 09 BE 20 CD FE
BE08— 60 A9 00 85 3C A9 40 85
BE10— 3D A9 FF 85 3E A9 7F 85
BE18— 3F 60 A2 00 BD 8E BE 20
BE20— F0 FD E8 E0 0B 90 F5 20
BE28— 09 BE 20 FD FE 8D 81 C0
BE30— 8D 81 C0 A9 00 85 42 85
BE38— 3C A9 D0 85 43 A9 40 85

BE40— 3D A9 FF 85 3E A9 6F 85
BE48— 3F A0 00 20 2C FE 8D 89
BE50— C0 8D 89 C0 A9 00 85 42
BE58— 85 3C A9 D0 85 43 A9 70
BE60— 85 3D A9 FF 85 3E A9 7F
BE68— 85 3F A0 00 20 2C FE 8D
BE70— 82 C0 60 00 00 8D D0 F2
BE78— E5 F3 F3 A0 E1 EE F9 A0
BE80— EB E5 F9 A0 F4 EF A0 D3
BE88— C1 D6 C5 A0 A1 A1 8D CC
BE90— EF E1 E4 E9 EE E7 AE AE
BE98— AE 00

磁盘数据的压缩存储技巧

四川南充一中(637000) 陈庆祥

众所周知,在利用磁盘存储数据信息时,应尽可能地将性质结构相同的记录存放在同一文件中,以便于进行各种处理。这一点对于只配备了单驱的系统显得更为重要。例如,现在要将某地初中升学考试的万名考生记录以随机文件的形式存盘(目前,百万人口的县市每年初中毕业生都在万名左右)。每个考生的姓名、性别、类别和 7 科考分共 10 个数据项组成一个占 45 个字节的记录(前三项共 6 个汉字,占 18 个字节,后 7 项占 15 个字节,加上 10 个分隔符占 10 个字节),那么,一张单面单密度软盘即使格式化到 40 磁道且不存入 DOS,至多也只能容纳 3000 左右的考生记录。万名考生记录就不得不存放在四张盘内,致使查询记录时必须频繁地换盘(特别是随机查询时),不仅增加许多麻烦,也明显地降低了处理效率。

那么,能否在现有的硬件条件下,用一张软盘来存放更多的数据信息呢?比如,将上述的万名考生记录存入一张盘内。经笔者的反复探索,证明是可行的,其出路就在于对数据进行巧妙的压缩处理。

根据考生记录各数据项的类型与特征,分三种情况进行压缩处理:对于姓名这几个汉字,只存入其内码中的区码与位码,而不必存入汉字的鉴别码 \$7F(中华机硬汉字以 \$7F 为鉴别码);对于考生性别与类别分别用代码表示(如 1—男,2—女),并将两个代码合为一个二位整数,再用 CHR\$ 函数转换为对应的一个 ASCII 字符;对于 7 门考分皆为 0~100 的整数,当然可用上面类似方法转换为 7 个 ASCII 字符,不过由于 CHR\$(4) 为 DOS 命令引导符,会影响数据的正常存取,故可将各考分加上 5 再转换。所以本来占用 45 个字节的考生记录可压缩为 15 个(由于各 ASCII 字符均表示一个相对独立的信息,相互间不必再用分隔符,仅在记录尾部以一个回车符用来区别不同记录),整个文件长度仅为原来的三分之一,一张磁盘就完全可以容

纳一万多名的考生记录。

很明显,数据压缩存储的结果不仅大大提高了磁盘的利用率,方便了大批量的数据处理,而且由于对同一批数据的磁盘存储量大为减少,文件存取时间也相应缩短(虽然数据的转换压缩要耗费时间,但由于 CPU 的处理比磁盘动作快千倍以上,故文件存取仍比原先快得多),真可谓一举数得。

以上数据压缩存储的方法适用于所有长度变化范围较小的数据(较长数据可分段处理),既适用于数值,也适用于字符(如整篇中文文章),既适用于顺序文件,也适用于随机文件,而且不限定任何一种机型。

本文末尾提供了考生记录压缩存储与随机查询的两个程序实例(在中华学习机上通过)。其中程序 1 在每次运行前打入不同的 DATA 语句(用 DATA 语句提供数据便于检查、修改),可将万名考生记录分任意次存入同一随机文件内,(文件长度不受限制,直至数据充满整张磁盘,程序后提供了两次运行的压缩存储例子),程序 2 能随机地查询由程序 1 建立的随机文件内任何一名考生的记录,并且用通常方式输出。

程序 1

```
10 REM 考生记录的压缩存储,以单个 ASC 字符代替
   1—3 位数
20 D$ = CHR$(4);H$ = CHR$(127);S = 0
30 INPUT "是该文件数据的首次输入吗? (Y/N)";SC$
40 PRINT "请稍候! 正在将数据进行转换及存入磁
   盘.";PRINT D$;"OPEN HK,L15"
50 IF SC$ <> "Y" THEN PRINT D$;"READ HK,R0":
   INPUT N
60 READ H; IF H = 0 THEN N=N+S: GOTO 150
70 READ N$,XB,SL: FOR I = 1 TO 7: READ M: M
   $(I)=CHR$(M+5): NEXT
80 PRINT D$;"WRITE HK,R";H
90 FOR I = 1 TO LEN(N$):P$ = MID$(N$,I,1)
```

```

100 IF P$ < > H$ THEN PRINT P$;
110 NEXT
120 PRINT CHR$(10 * XB + SL);
130 FOR I = 1 TO 7: PRINT M$(I);
140 NEXT: PRINT: S = S + 1: GOTO 60
150 PRINT D$; "WRITE HK,R0": PRINT N: PRINT
    D$; "CLOSE"
160 PRINT "OK! 本次存入了"; S; "个考生记录, 文件内
    共有"; N; "个记录, 下次从 200 行起重新打入数
    据。": END
170 PRINT D$: GOTO 40
200 DATA 00001, 王建华, 2, 1, 85, 73, 70, 66, 74, 67, 82
210 DATA 00003, 周 敏, 2, 1, 70, 84, 65, 86, 90, 79, 70
220 DATA 00002, 刘 宏, 1, 1, 86, 90, 92, 77, 86, 79, 92
230 DATA 00004, 肖 军, 1, 2, 88, 94, 75, 82, 90, 84, 96
240 DATA 00000

```

]RUN

是该文件数据的首次输入吗? (Y/N) Y

请稍候! 正在将数据进行转换及存入磁盘。

OK! 本次存入了 4 个考生记录, 文件内共有 4 个记录, 下次从 200 行起重新打入数据。

]LIST 200—

```

200 DATA 00006, 何 锐, 1, 1, 88, 94, 85, 82, 90, 99, 94
210 DATA 00005, 张中明, 1, 2, 80, 64, 70, 56, 64, 69, 74
220 DATA 00000

```

]RUN

是该文件数据的首次输入吗? (Y/N) N

请稍候! 正在将数据进行转换及存入磁盘。

OK! 本次存入了 2 个考生记录, 文件内共有 6 个记录。下次从 200 行起重新打入数据

程序 2

10 REM 随机查询文件中的考生记录

20 H\$ = CHR\$(127): K\$ = CHR\$(32): XB

\$ (1) = "男": XB\$(2) = "女": SL\$(1) = "应
届": SL\$(2) = "往届"

30 D\$ = CHR\$(4): PRINT D\$; "OPEN HK,L15"

40 INPUT "请输入欲查询的考生考号: "; H: N\$ = "

50 IF H = 0 THEN PRINT D\$ "CLOSE": END

55 PRINT D\$ "READ HK,R"H

60 FOR I = 1 TO 3: GET A\$: GET B\$

70 IF A\$ = K\$ AND B\$ = K\$ THEN N\$ = N\$ +
K\$ + K\$: GOTO 90

80 N\$ = N\$ + H\$ + A\$ + B\$

90 NEXT: M = 0: GET XL\$: XL = ASC(XL\$)

100 XB = INT(XL / 10): SL = XL - 10 * XB

110 FOR I = 1 TO 7: GET M\$: M(I) = ASC(M\$)
- 5: M = M + M(I): NEXT

120 PRINT D\$: PRINT "第"H"号考生记录如下:"

125 PRINT "考 号 姓 名 性别 类别 政 语 数
英 物 化 生 总分"

130 PRINT RIGHT\$("0000" + STR\$(H), 5) * " ;

140 PRINT N\$ " " XB\$(XB) " " SL\$(SL) " " ;

150 FOR I = 1 TO 7: PRINT LEFT\$(STR\$(M(I)) +
" ", 3);

160 NEXT: PRINT M: PRINT

170 PRINT D\$: GOTO 40

]RUN

请输入欲查询的考生考号: 6

第 6 号考生记录如下:

考号	姓名	性别	类别	政	语	数	英	物	化	生	总分
00006	何锐	男	应届	88	94	85	82	90	99	94	632

请输入欲查询的考生考号: 1

第 1 号考生记录如下:

考号	姓名	性别	类别	政	语	数	英	物	化	生	总分
00001	王建华	女	应届	85	73	70	66	74	67	82	517

请输入欲查询的考生考号: 0

(上接第 8 页)

ENDIF

ENDIF

ENDIF

ENDDO

RETU

* K_BOARD.PRG

* 功能: 模拟 dBASE II Plus 中的 Return to Master

PARAMETERS P1,P2,P3

KEYBOARD CHR(27)

W_RETU=.F.

RETU

* 自定义函数 CENTER

FUNCTION CENTER

PARAMETERS STRING

RETURN INT(80-LEN(STRING))/2

为 Apple Dos 创造三条实用命令

湖南株洲硬质合金厂(412000)王志超

使用 DOS 3.3 的苹果机或中华学习机用户都会感到:①经常要为 10—16 进制的数据转换大伤脑筋;②复制文件不得不调用专用的拷贝程序,颇为不便;③要了解软盘某个文件的大致内容而不破坏内存的现行程序也很不方便。那么,能否编制几条格式简单的 DOS 命令,来解决上述问题呢?笔者经过一番探索,终于达到了目的,现在推荐给大家,希望能为各位带来方便。

一. 数制转换命令—ZH

1. 本命令用于立即执行方式。格式:

① 10→16 进制转换:ZHAX (X 为 0——65535 的 10 进制数据)

② 16→10 进制转换:ZHAX (X 为 0——FFFF 的 16 进制数据)

以上的格式中,A 为键语,不可缺少。

2. 原理:本命令巧妙地利用了 DOS 本身的几段子程序。首先调用 \$A1A4 来检查命令中是否有“\$”(16 进制标志),接着调用 \$A1B9 来转换数据,转换好的数据放在 \$44、\$45 单元,这时再一次判断输入数据是否 16 进制,是则调用 \$DE24 将转换好的 10 进制数输出到屏幕,否则调用 \$F940 以 4 位 16 进制的形式输出。

二. 文件拷贝命令—COPY

1. 本命令在立即方式或延迟方式均有效。格式:

COPY 文件名 1, 文件名 2

格式中的文件名 1 为源盘的文件名,文件 2 为目标盘的新文件名。

2. 本命令可拷贝 A、B、I 类型的文件。它综合利用了“BLOAD”和“BSAVE”的部分功能。命令执行时,首先读源盘,当文件 1 全部读出后,光标会停在命令的末尾,待用户换上目标盘并按任意键后,就以文件名 2 为名拷至目标盘。拷贝过程中的被拷文件暂存区为 \$3000\$7FFF,可拷贝的文件最大长度为 20K。如果在中文状态下使用,立即方式下可先进入“西文”再执行命令,延迟方式下可在程序中添加以下语句:

(行号)HOME: TEXT: PRINT D\$“COPY 文件名 1, 文件名 2”: PRINT D\$“PR#3”: PRINT

三. 查看文件命令—TYPE

1. 本命令在立即方式或延迟方式均有效。

2. 说明:类似的命令曾见于其它资料,但均存在以下弊病:①命令执行后再执行别的 DOS 命令时常出现“缓冲区不够”的错误;②在中文状态使用时很容易转入“西文”,且常进入反相显示。笔者编制的 TYPE 进行了改进,即:增加清屏功能;调用 \$A316 以释放缓冲区;调用 \$F273 恢复正相显示;屏蔽“西文”键等,这样,在中、西文状态下都能正常显示出指定文件的内

容。另外,CTRL—S 暂停、CTRL—C 中止命令的执行。

四. 命令修改步骤

1. 输入后附的机器码程序,其中,\$9500—\$9534 为 TYPE,\$9535——\$9598 为 COPY,\$9599—\$95BC 为 ZH。

2. 修改有关单元:

① 改命令名。首先将原 \$A898—\$A8E3 的机器码顺序向前移一个字节,这只要执行 A897 < A898.A8E3M 命令即可。然后:

A893;54—59—50—C5

A8E3;43—4F—50—D9

A8EF;5A—C8

② 改键语表。

A931;71—79—

A935;00—01

③ 改入口地址。

9D26;FF—94

9D46;34—95

9D4A;98—95

3. 调出 HELLO 程序,在最前面加一句:

HIMEM;38144;PRINT CHR\$(4)“BLOAD COMMAND”(如果是中文 HELLO,则可去掉 HIMEM;38144 语句)

4. 格式化一个新盘,并将机器码程序以“COMMAND”为名存盘。

5. 启动新盘。现在,你已经可以得心应手地使用这三条新的 DOS 命令了。

最后请注意:原 DOS 命令 CHAIN、IN#、FP 失效。

9500—20	7E	D2	20	A8	A2	20	8C
9508—A6	09	80	C9	91	F0	03	20
9510—ED	FD	AD	00	C0	2C	10	C0
9518—C9	93	D0	08	AD	00	C0	10
9520—FB	2C	10	C0	C9	83	F0	07
9528—AD	C5	B5	C9	05	D0	D7	20
9530—16	A3	4C	73	F2	A9	00	8D
9538—72	AA	A9	30	8D	73	AA	20
9540—A8	A2	AD	C2	B5	29	7F	85
9548—08	20	D5	A3	A5	08	C9	02
9550—F0	08	C9	01	F0	07	20	7A
9558—A4	85	06	84	07	20	82	A3
9560—20	0C	FD	A0	1D	B9	93	AA
9568—99	75	AA	88	10	F7	A5	08
9570—20	D5	A3	AD	60	AA	8D	6C
9578—AA	AD	61	AA	8D	6D	AA	A5
9580—08	C9	02	F0	0E	C9	01	F0
9588—0A	A4	06	A5	07	20	48	A3
9590—4C	51	A8	20	4B	A3	4C	51
9598—A8	20	8E	FD	A2	03	8E	5D
95A0—AA	20	A4	A1	48	CE	5D	AA
95A8—20	B9	A1	A8	68	C9	A4	F0
95B0—08	A9	A4	20	ED	FD	4C	40
95B8—F9	98	4C	24	ED			

ProDOS 磁盘操作系统入门(续)

廖 凯

五、屏幕编辑

ProDOS 允许光标既可在 40 列方式下又可在 80 列方式下的屏幕范围内移动,如果计算机是处在 80 列方式下,那么在光标内会出现一个小+(加号),以区别于 40 列方式,按下 ESC 键即进入屏幕编辑状态。它有如下编辑键:

ESC A	光标向右移一位,退出编辑状态
ESC B	光标向左移一位,退出编辑状态
ESC C	光标向下移一位,退出编辑状态
ESC D	光标向上移一位,退出编辑状态
ESC K	光标向右移一位,保持编辑状态
ESC J	光标向左移一位,保持编辑状态
ESC M	光标向下移一位,保持编辑状态
ESC I	光标向上移一位,保持编辑状态
ESC E	由光标处删至行尾
ESC F	由光标处删至页末
ESC @	同时按下 ESC 和 @ (SHIFT-2) 键,功能同 HOME 指令
CTRL-X	删除当前打入的行
CTRL-K	光标上移一行
CTRL-J	光标下移一行
→键	光标右移一位,将光标下的字符输入内存
←键	光标左移一位,将光标下的字符从内存中删去
↑键	同 CTRL-K。只有 APPLE IIe、IIc 及中华学习机才有此键
↓键	同 CTRL-J。只有 APPLE IIe、IIc 及中华学习机才有此键
DELETE 键	删除行中最后一个字符并回送该字符。只有 APPLE IIe 和 IIc 才有此键

六、程序除错

ProDOS 有几种除错的方法。TRACE 命令被用来显示当前正在执行的行号。即使你从磁盘发送或接收信息,TRACE 也将同 ProDOS 一起正确地工作。如果在 TRACE 有效时而输出同时被显示在屏幕上,那么会有混乱的结果,终止 TRACE 的命令是 NOTRACE。

另一个除错的方法是使用 STOP 和 CONT 命令,从你的程序内部将 STOP 放置在重要的位置上,当程序遇到 STOP 时将停止,你可以由立即方式打印出变量的当前值,要继续执行输入 CONT 即可。这过程是分析程序的一个最简单和有效的方法。

七、错误中断

错误中断是开发一个程序的一个很重要的部分。每当程序发生错误时,它会纠正错误并重新运行程序。在一个程序运行时,发生许多不同类型的错误是不罕

见的,在一个错误发生及中断它时,要懂得处理错误的技巧。

处理错误最常用的命令是 ONERR GOTO 及 RESUME。若错误发生,将出错标志单元 216 的最高位置 1,并将错误类型码放在 222 单元内。POKE 216,0 用于清除错误标记。当一个错误发生时,内部控制堆栈指针被破坏,控制堆栈记录着 GOSUB/RETURN 的位置,错误发生时,错误代码被加到堆栈上,因此在 GOSUB 子程序内发生的任何错误都将破坏 GOSUB/RETURN 指针。ONERR GOTO 语句是用于程序出错时转到指定的错误处理子程序,RESUME 是错误处理子程序的返回语句,它将自动地消除堆栈上无用的字符而保持控制堆栈的完整性。

控制堆栈问题可以通过下面的程序来减轻,此程序通过消除堆栈上无用的字符,来清除错误并保持控制堆栈的完整。

```
10 FOR I=768 TO 777
20 READ J
30 POKE I,J:NEXT
40 DATA 104,168,104,166,223,154,72,152,72,96
```

中断错误之后,用 CALL 768 来执行错误处理子程序,然后用一个 GOTO 语句返回到程序。

在中断错误时,认清错误发生的类型是很重要的,它给使用者一个信息,以指示使用者下一步做什么。例如使用者正发送输出到一个没被连接的打印机,使用者可用下面的方法截取和显示错误:

```
10 ONERR GOTO 1000
.....
1000 IF PEEK(222)=3 THEN PRINT"The printer is not connected or is not ON,Please check the condition and try again."
```

八、程序设计结构与优化

1. 程序结构

程序主要由三个部分组成:初始部分、主体和子程序,首先给程序置初值,设置变量和定义所有数组,然后开发主体,包括菜单和树形分枝,最后是子程序,主体将充分利用这些子程序。使用子程序将使你的程序更灵活、更易读和更容易除错。

2. 变量的规定

APPLESOFT 使用变量的三种类型是:整型、实型和串变量。整型变量用加在变量名后的%表示,其值在-32767 至 32767 之间。整型变量变量名占用 2 字节,值占用 2 字节。实型变量是以最多 9 位十进制数字的精确度来表示其值。实型变量变量名占用 2 字节,值占用 5 字节。串变量表示一个字符串,串变量的最大长度为 239 个字符。串变量变量名占用 2 字节,长度占用 1 字节,指针占用 2 字节,字符串内每一字符占 1 字节。

3. 程序优化

一个好的程序应当具备以下一些条件:

- ①执行效率高,节省运算时间,节省内存空间。
- ②程序条理清楚,用比较简单易懂的方法表示。
- ③考虑到运行时可能发生的各种情况,应设立必要的错误处理程序。

④有适当的说明(用 REM 语句),使程序的使用者能了解程序中各部分的作用。

要想设计好的程序,建议按照以下几方面进行设计:

- ①需选择好的计算方法,这是最重要的。
- ②最好采用结构化的程序设计方法,尽量少用 GOTO 语句,而用 IF-THEN 形式,使复杂的程序也能一目了然。
- ③重复使用同一部分语句时尽量用子程序。
- ④节省使用数组,这是节约内存的关键,不要任意指定过大的数组,尽量重复使用前面已用过而不再用的数组或变量以节省内存。
- ⑤尽量在一行中写几个语句,这样可省内存。
- ⑥重复出现的表达式应事先计算好赋予一个变量保存,以减少重复计算。
- ⑦整型变量的计算比较准确且运算速度快。如内存不够用时,应尽量用整型变量和数组代替单精度变量和数组。整型变量占二个字节,而单精度变量占四个字节。
- ⑧内存不够时,可减少不必要的 REM 语句。
- ⑨在写程序时应考虑到调试和日后使用维护的需要。

下面是一些语句占用内存的情况:

用 DEF FN 语句定义一个函数时占用 6 字节,各保留字占用 1 字节,一个 FOR/NEXT 循环占用 16 字节,一个正被执行的子程序占用 6 字节,在表达式内的各对括号占用 4 字节,在表达式内各个计算的临时结果占用 12 字节。

九、文本文件

ProDOS 的最大特点之一是文件的快速存取,它比 DOS3.3 快 8 倍,在对大型文件进行操作时尤其明显。ProDOS 文件的大小在磁盘上不限于 140K 字节,在使用硬盘时,你可以建一个具有 16M 字节的文件。ProDOS 取消了 MAXFILES 命令,它可同时使用 8 个文件,对于每个打开的文件,ProDOS 将自动分配 1K 字节的缓冲区。

ProDOS 允许用“-”命令来运行文本文件,如果文件带有行号,那么 EXEC 命令不执行这些程序行,而是按行号次序将它们装入内存。如果文件不带有行号,那么 ProDOS 将执行它们。

ProDOS 的文本文件有两种类型:顺序存取文件和随机存取文件。

(一)顺序文件

顺序文件是由若干个连接在一起的数据域组成,数据域由若干字节的字符组成并以回车字符(CR)为

结束标志,一个数据域最多可包含 230 个字符。

1. 写顺序文件

WRITE 命令在 ProDOS 内被改进,在顺序文件内 WRITE 命令可以使用 F# 参数,它是在写操作之前把文件位置指针向前移动若干个数据域作为当前的数据域,这参数取代了 POSITION 命令,在顺序文件内使用 F# 参数的形式是:

10 PRINT D\$;"WRITE/卷目录/文件名,F#"

下面的例子是向 EXAMPLE 文件写入一些数据。

```
10 REM Sequential Text File
100 HOME
110 D$ = CHR$(4)
120 DIM N$(100), C$(100), P$(100)
130 A = A + 1
140 INPUT "Name: "; N$(A)
150 INPUT "Company: "; C$(A)
160 INPUT "Phone: "; P$(A)
170 INPUT "Enter another name?"; Q$
180 IF Q$ = "Y" OR Q$ = "y" THEN HOME: GOTO 130
190 IF Q$ = "N" OR Q$ = "n" THEN 210
200 GOTO 170
210 PRINT D$; "OPEN/DATA. DISK/EXAMPLE"; PRINT
D$; "CLOSE"; PRINT D$; "DELETE/DATA. DISK/EXAMPLE"; PRINT D$; "OPEN/DATA. DISK/EXAMPLE"
220 PRINT D$; "WRITE/DATA. DISK/EXAMPLE"
230 FOR I = 1 TO A
240 PRINT N$(I); PRINT C$(I); PRINT P$(I)
250 NEXT
260 PRINT D$; "CLOSE"
270 END
```

磁盘命令有通过路径来指定的,路径包含卷目录和文件名,文件的打开、关闭、删除和再打开是用以清除原有的同名文件,以防文件被重写。

注意 CLOSE 命令没有路径,当你只用 CLOSE 命令时,所有打开的文件都将被关闭,如果需要,你可以关闭一个指定的文件。

如前所述,ProDOS 是以 512 字节/块为单位来写盘的,并且 CLOSE 或 FLUSH 命令会强使所有剩余的字符被写入文件,FLUSH 命令不能将文件关闭,但会写盘并清除缓冲区。当你确信要立即将缓冲区内的数据写盘时,你就要用 FLUSH 命令。

注意每次打开文件时会占用很多的时间,故最好减少打开文件的次数。最有效和省时的方法是一次将所有数据输入完。

PRINT 语句的用法:在写盘时,你可以用 PRINT 命令来改变数据域的输出格式。例如:

①PRINT "ProDOS":将字符串 ProDOS 写入文件,随后紧跟一个回车字符。

②PRINT "Pro";PRINT "DOS":将字符串 Pro 写入文件并将下一个 PRINT 语句加到此数据域的末尾。

③PRINT "Pro", "DOS":将字符串 ProDOS 写入文件,这逗号无效。 (未完待续)

第二章 微处理器、存储器、寄存器简介

南京大学大气科学系(210008) 朱国江

一、计算机的主要功能

计算机最主要的功能是处理信息。所谓信息,是指客观事物发出的消息、情报、指令、数据和信号等。这些信息可以是文字、图形、声音等等。这种功能可以简单概括成一句话:输入——处理——输出。这就是说,向计算机输入待处理的各種信息,计算机接收了这些信息,对它进行处理、计算、变换和加工,最后再由计算机向外部设备输出加工好的信息。因此,计算机就其本质来说,可以称为信息处理机。计算机在处理信息的过程中,可以不需要人的干预而能自动进行,并且处理速度非常快。自动化和高速度是计算机处理信息的两个最突出的特点。

二、计算机的硬件组成

一台微型计算机通常由四部份组成:存储器、运算器、控制器和输入输出设备。由于运算器和控制器是计算机进行信息处理的中心,通常将它们合在一起称为计算机的中央处理单元(Central Processing Unit,常简称CPU),输入计算机的信息都是在这里处理的。在微型计算机中,中央处理单元CPU是做在一块集成电路芯片内,因而叫做微处理器(Microprocessor Unit,简称MPU)。CPU与存储器、输入/输出接口(常简称为I/O口)一起组成计算机的主机,而把输入/输出设备统称为计算机的外部设备。在外部设备中键盘是输入设备,荧光屏显示器、打印机等是输出设备,而把磁盘、磁带等称为外部存储器。

(1) 输入输出设备

输入设备的功能是输入数据和程序,它与计算机之间的连接部分称为输入接口。输出设备的作用是输出计算机处理的结果,它与计算机的连接部分称为输出接口。因此,接口就是计算机与外部设备之间的连接部分,或者说是计算机与外部设备之间交换信息的通道。一般情况下,外部设备和计算机之间信息流通的接口应包括三个部分:暂存输入/输出数据的数据口(数据缓冲寄存器),用以存放外设忙、闲状态信息的状态口。控制外设启动和停止的控制口等。

计算机访问外设时,首先由CPU向该外设对应的接口发出要访问外设的地址和控制信号,检查该外设接口中的外设是否准备好。若准备好,则CPU通过数据总线与接口中的数据口交换数据,接口再通过它与

外设之间的I/O数据线与外设交换数据,从而完成了计算机对外设的一次访问。若未准备好,则等准备好后,再完成上述交换。所以,计算机对外设的控制,实际上是对接口的控制,外设的地址也是赋予其接口的。

(2) 存储器

存储器是计算机中的重要部件,它的用途是存储数据和程序,既能把信息保存起来,又可把存入的信息取出来而又不破坏原存储内容,还可以重新记录新的信息,把原存储内容消去。向存储器存入数据,叫做写入。从存储器取出数据,叫做读出,对存储器进行读或写操作,叫做访问存储器。

存储器分内存和外存。微型计算机的外存储器主要是磁盘或磁带,而内存储器又分为ROM(只读存储器)和RAM(随机存取存储器)。ROM只能读出数据,不能向它写入数据,主要用来存放计算机的系统程序和常数;RAM既可以对它写入数据,又可以从中读出数据,所以又称读/写存储器,它用来存放用户程序、数据及结果等。

内存储器ROM或RAM内部含有很多存储单元,每一个存储单元都有唯一确定的编号,叫做地址。存储器是按地址存取信息的。当计算机要把一个数据存入某个单元中,或从某个单元中取出数据时,都要首先给出该存储单元的地址,然后查找此存储单元,查到后再根据指令进行数据的存取。地址好比一个教室有一个唯一确定的教室号码,到大楼里找人,要按照教室的号码去寻找一样。所不同的是一个单元能存储八位二进制数,又称一个字节数(字长为8)。对16位机或32位机来说,存储单元的字长则是16位或32位。

微型计算机的内存储器是用大规模集成电路来实现的。最基本的单元电路是触发器,它有“0或1”两个不同状态,代表一位二进制数,因而一个存储单元(存放八位二进制数的)就需要8个触发器。由此构成的存储器称为半导体静态存储器。此外还有利用集成电路电容上有无电荷分别代表“0和1”状态的。因此构成的是半导体动态存储器,它必须周期性地自动将存储的数据重写,以维持电容器上的电荷,这种操作称为存储器的刷新。例如CEC—I的RAM是使用的HM50464动态存储芯片,每片存储容量为64K×4位,两片并联用,组成64K×8位的读/写存储器。读写存储器由存储体、读/写控制部分、译码驱动器三部分组成,CPU通

过控制总线、地址总线、数据总线来控制对存储器的读和写。

(3) 运算器

运算器是对数据进行算术运算(加、减、乘、除)和逻辑运算(与、或、异或)的部件。复杂的运算则是这几种基本运算的组合。

算术逻辑运算单元(ALU)是运算器的核心部分。ALU的两个操作数一个来自累加寄存器(简称累加器A),另一个来自某一个暂存寄存器。运算结果一方面通过CPU内部总线送往累加器A,同时将运算结果的标志送往标志状态寄存器P。由此可以看出累加器A在运算中的地位十分重要,它既是数据的来源地,又是算出结果所要送到的目的地。

运算器的取数操作和运算操作,都是在控制器的控制下进行的。

(4) 控制器

控制器是用来控制整个计算机进行计算的核心部件,它用来指挥和协调计算机各部分的工作。控制器的主要工作有:发出允许输入命令将程序和数据通过输入设备送入存储器中保存起来;发出取数命令从存储器中取数送入运算器;发出运算命令通知运算器完成计算工作;发出送数命令把数据或计算结果送入存储器;从存储器中取出计算结果,并发出输出命令控制输出设备将结果打印在纸上或显示在荧光屏上。总之,计算机能够自动地处理信息,按照一定的顺序工作,按照指令规定的内容执行相应的操作,全都离不开控制器的指挥和协同动作。

控制器由指令寄存器(IR),指令译码器(ID)和操作控制部件三部分组成。运行程序时,CPU从存储器取出指令,首先送入指令寄存器,然后再送入指令译码器,经过译码分析指令的内容,再根据这些内容,送入操作控制部件,产生一系列的控制信号,并在节拍发生器控制下,按一定时间顺序向计算机各有关部件发出执行命令,控制计算机按一定时序有条不紊地进行工作。

三、计算机信息流动和总线结构

前面指出,以微处理器MPU为核心,加上大规模集成电路的存储器、输入/输出接口等组成的微型计算机,再配以所需要的外部设备就构成了微型计算机硬件系统,硬件系统加上必须的系统软件(操作系统、语言处理程序、数据库管理软件、服务性软件等)和应用软件(用户程序)就构成了一个微型计算机系统。

那么软件和硬件之间,各逻辑部件之间,信息是如何流动的?这通常都是采用一束公共的信息传输线——系统总线来进行的。这就是说CPU通过系统总线实现与存储器、输入/输出接口之间的相互连接来交换信息。根据系统总线上传输信息的性质,系统总线又分为地址总线、数据总线和控制总线。

微处理器通过地址总线指定它要访问的内存存储单元的地址码,或者输入输出口的地址码;通过数据总线

与存储器、输入/输出接口交换数据;通过控制总线发出控制信息,控制存储器和输入/输出接口工作。

因此,在计算机内部有两股信息在流动:

一是数据流,它包括各种原始数据,程序,现场信息等。它们由输入设备输入到计算机并存储于存储器中,在运算处理过程中,数据又从存储器读入运算器进行运算,运算结果再存入存储器或由输出设备输出。

二是控制信息流,它包括各种命令(或程序)。这些控制信息以数据的形式输入到存储器中,运行时再由存储器读入控制器,由控制器经译码后变为各种控制信号,来控制运算器进行各种运算和处理,控制存储器的读和写,控制输入/输出设备的启动和停止等。

四、微处理器的内部结构

6502MPU的内部结构可简单分为两大部分:寄存器部分和控制器部分。寄存器部分包括数据总线缓冲器、数据输入锁存器、程序计数器(PCL,PCH)、累加器(A)、算术逻辑运算单元、堆栈指示器(S)、变址寄存器(X,Y)、地址总线锁存器、处理器状态标志寄存器(P)等。寄存器的主要功能是完成对数据的传送、运算和处理。控制器部分包括时钟发生器、指令寄存器、指令译码器、中断逻辑电路等。控制器的主要功能是对读入微处理器的指令进行译码,确定执行该指令的周期数,提供各寄存器工作过程的控制信号,指定寄存器执行该指令的操作,以及改变当前指令的执行次序等。

五、微处理器内部寄存器

由上面分析可知,微处理器的实际处理工作是由寄存器部分完成的,为了更好地学习6502指令系统,我们必须对微处理器内部寄存器有一个全面的了解,特别是对6502中可供编程的几个寄存器加以关注。此外,微处理机中状态标志寄存器的标志位,是实现条件转移的依据,在程序设计中有很重要的地位,学习中更应仔细和小心。

1. 累加寄存器A(Accumulator)

累加寄存器是一个8位寄存器(可存储8位二进制数),用大写字母A作标志。它是计算机执行指令过程中的重要逻辑部件。从外部数据总线输入的数据,一般都暂时存放于A中,而在运算时才从A中取出,送到算术逻辑单元参与运算,其运算的结果仍送回A中。

由于寄存器在CPU内,取用它们的信息比取用CPU外部存储器的信息方便,速度也快,所以现代计算机都在运算器内设置这样的寄存器,用来存放参加运算的原始数据及中间结果,访问它和访问存储器一样,并可在它们之间进行数据传送和各种操作,所以又简称累加器。

与累加器A有关的指令有:实现内存与寄存器之间的数据传送,如LDA,STA;实现寄存器与寄存器之间的数据传送,如TAX,TXA,TAY,TYA;进行算术和逻辑运算的有ADC,SBC,AND,ORA,EOR;进行比较运

算的有 CMP;进行堆栈操作的有 PHA,PLA 等等。这些指令以后还要详细介绍。

2. 程序计数器 PC(Program Counter)

程序计数器用 PC 表示,它由两个 8 位计数器组成(可存放 16 位二进制数),一个是高位地址程序计数器 PCH,用以表示地址的高位部分;另一个是低位地址程序计数器 PCL,用以表示地址的低位部分。它们是 6502 中唯一的一个 16 位寄存器。

PC 中存放的是内存地址,它将到内存的相应地址中取出指令执行,每取出一个指令后,为取下一个指令字节作好准备,PC 中的内容要发生改变,通常是指向下一个指令首字节的地址。这样,就可以使程序按顺序运行。

当程序需要转移到新的地址(如执行转移指令、调用子程序或中断)时,PC 中将被放进要转移的目标地址值。由此可知,PC 控制着程序的执行或转向,因而也可称为指令地址指示器。

例如,指令 JMP (\$ 4000),其指令代码为 6C 00 40,表示重新设置 PC 值。假设内存单元 \$ 4000 的内容是 00,\$ 4001 的内容是 60,则执行 JMP (\$ 4000)指令后,PC 的值为 \$ 6000。

3. 变址寄存器(Index Register)

变址寄存器共有两个,每个可表示 8 位二进制数,分别用 X 和 Y 作标志。它们都是跟地址操作有关的寄存器,主要用于变址寻址方式中存放地址偏移量,也常在编程中当做一个计数器来用,它们可以用指令控制而被置成一个常数,也可以方便地用于加 1、减 1、比较、暂存、恢复数据等简单操作。

与变址寄存器有关的指令:由内存单元取数至变址寄存器的有 LDX,LDY;将变址寄存器中的数存入内存的有 STX,STY;累加器中内容与变址寄存器相互传递的有 TAX,TAY,TXA,TYA;堆栈指示器的内容与变址寄存器互换的有 TSX,TXS;变址寄存器内容进行加 1、减 1 操作的有 INX,INY,DEX,DEY;变址寄存器与存储器内容比较的有 CPX,CPY 等等。

4. 堆栈指针寄存器 S(Stack Pointer)

堆栈指针寄存器,用 S 作标志,它是用来指示堆栈栈顶位置的寄存器,即用于存放栈顶地址,以便进行堆栈操作。由于 6502 规定堆栈设置在第 1 页存储器中,所以堆栈指针 S 也是 8 位寄存器,只用来指出堆栈地址的低 8 位地址。当数据进栈时,堆栈指针 S 自动减 1;而在数据出栈时,栈指针 S 自动加 1。

和堆栈指针 S 有关的指令有:

- TSX, TXS;堆栈指示器内容与变址寄存器内容互换。

- PHA, PLA;对累加器 A 进行进、出栈处理。

- PHP, PLP;对标志寄存器 P 进行进、出栈处理。

5. 标志位寄存器 P(Processor Status)

标志位寄存器,用 P 作标志,这也是一个 8 位寄存器,其中第 5 位规定不用。

在程序设计中,每条指令在执行以后,往往发生进

位、溢出、以及结果为全零,或者结果为正数、负数的情况。指令执行后常常需要保留这些情况,以作为条件分支的根据,标志寄存器 P 就是为了适应这些情况而设置的。

这七个标志位是:

- C—进位标志(Carry),表明指令执行后最高位的进位状态。

C=1,表示有进位或无借位;

C=0,表示无进位或有借位。

- Z—零标志(Zero),表明刚才指令的执行结果是否为零。

Z=1,结果为零;

Z=0,结果非零。

- I—中断屏蔽或中断禁止标志(Interrupt disable),表明是否允许中断。

I=1,禁止中断;

I=0,允许中断。

- D—十进制运算标志(Decimal mode),表明采用 10 进制运算还是二进制运算。

D=1,算术指令作十进制运算;

D=0,算术指令作二进制运算。

D 的状态只对后续的加法或减法指令有作用,它不表示运算的结果,只是在运算前对数作出的规定。

- B—软件中断指令标志(Break flag),表示中断状态。

B=1,中断;

B=0,未中断。

- V—溢出标志(Overflow),表明溢出状态。

V=1,算术运算结果发生溢出;

V=0,无溢出。

- N—负数标志(Negative flag),表明一个数是否为负(一个带符号的 8 位二进制数,最高位为 1 时为负,否则为正)。

N=1,为负数;

N=0,非负数。

对标志位的学习应注意几点:

- 各条指令对标志位的影响不尽相同。

- 寄存器 P 中各标志位的状态一般是指当前指令执行后的状态,所以,标志位常常用在执行条件转移时作为条件判断的依据。

- 6502MPU 还可用专门指令对某些标志位进行置位或复位的操作。如 CLC 指令,其作用是清进位位,即对 P 寄存器中的 C 标志位置 0,而不影响其它各位;SEC 指令则使 C 标志位置 1,对其余各标志位均无影响。

六、内存地址分配

6502MPU 有 16 条地址线可以形成,可访问 64K 字节内存空间。但对中华学习机 CEC—I 来说,由于采用了存储体切换技术(存储空间的映射技术),故实际可寻址 96K 字节,其中 ROM 占 32K 字节,RAM 占 64K 字节。

1. CEC—I 系统的存储空间分配

CEC—I 机的内存系统共有 64K 存储空间,其内存地址分配为三部分:读写存储器 RAM(\$0000—\$BFFF,共 48K)、外部设备码输入/输出空间(\$C000—\$CFFF,共 4K)、存储体切换空间(\$D000—\$FFFF,共 12K),此空间可映射到 RAM,也可映射到 ROM)。

2. RAM 空间分配

CEC—I 机具有 64K 字节的 RAM 空间, $64K=2^8 \times 2^8=256 \times 256$,故常将 64K 空间划分为 256 页(0—255),每页有 256 个单元。其中高八位为 0 的页称为零页(地址范围是 \$0000—\$00FF);高八位为 1 的页称为第 1 页(地址范围为 \$0100—\$01FF),其余类推。RAM 的空间分配情况如下:

- \$0000—\$00FF,零页,是系统 RAM 区,用于存放监控程序、BASIC 解释程序和 DOS 系统数据。

由于 6502 的指令系统中采用零页寻址方式的指令很多,监控程序、Applesoft、整数 BASIC 和 DOS 都有使用零页的情况,因而只有很少的 10 余个单元未被系统占用,因此,在您编制汇编程序时,若要使用到零页地址,必须特别小心,千万不要破坏当前使用的系统程序占用的单元。若要安全地使用零页地址,可先将零页中要用的单元的内容存放到别处去,在使用这些零页单元后,回转系统程序之前,再设法恢复这些单元原先的内容。

- \$0100—\$01FF,第 1 页,是系统堆栈区,MPU 通常只访问堆栈的栈顶单元。由于该区只有 256 个单元,所以堆栈内最多只能存放 256 个字节的内容。

- \$0200—\$02FF,第 2 页,是键盘的一个输入行的缓冲区,也是 256 个单元。所谓输入行的缓冲区,是指当你从键盘打入程序、命令等时,在未按回车键之前,这些输入的字符都以 ASCII 码的形式存放在这个缓冲区内。

- \$0300—\$03FF,第 3 页,在这一区中,监控程序和 DOS 都使用了少数高地址单元作为向量地址,BASIC 程序中有些目标子程序也使用了该区的部分低地址单元。如果用户的机器语言程序不太长,可以放在第 3 页中,但应注意不能存放在 \$3D0 以后的存储单元,这是因为 \$3D0 以后的单元已被系统程序占用。

- \$0400—\$07FF,共 1K 字节存储空间,它是文本显示和低分辨率图形方式第一页的显示缓冲区。这 1024 个字节不能存放用户的程序和数据。

- \$0800—\$0BFF,也是 1K 字节存储空间,它是文本和低分辨率图形方式第 2 页的显示缓冲区。但该空间通常是用户 BASIC 程序和数据的存放区。

- \$0C00—\$1FFF,用户区,存放用户程序和数据。

- \$2000—\$3FFF,共 8K 内存,是高分辨率图形第 1 页显示缓冲区。如果不用高分辨图形显示时,该空间可以供用户程序使用。

- \$4000—\$5FFF,共 8K 内存,高分辨率图形第 2 页显示缓冲区,同样,在不用高分辨率图形显示时,该空间也可存放程序和数据。

应该说明的是,无论低分辨率图形或是高分辨率图形都安排了两页显示缓冲区,这主要是为了方便用户,可以利用它们进行画面的分页显示和动画显示。

- \$6000—\$BFFF,是用户区。但在调用磁盘时,这个区域还要存放磁盘操作系统。例如,使用 DOS3.3 操作系统时,存储单元 \$9600—\$BFFF 区域就被用来存放操作系统以及作为文件输入/输出缓冲区。

- \$C000—\$CFFF,是输入/输出设备占用的地址码区域。输入/输出大致分为两种:一是内核输入输出,二是外围扩展输入输出。

- \$D000—\$FFFF,是存储体切换空间。在这个地址范围内的 ROM 区,用于存放 BASIC 解释程序和监控程序,在这个地址范围内的 RAM 区用于存放整数 BASIC 或 PASCAL 操作系统。

3. 存储体空间的切换

把不同的存储模块映射到同一地址空间称为存储体空间的切换。在 CEC—I 机中,把 12K 地址空间 \$D000—\$FFFF 进行两层切换。一层是在这 12K 空间进行 ROM 和 RAM 之间的切换;另一层是在 \$D000—\$DFFF 这 4K 空间又进行两个 4K RAM 之间的切换。

12KROM 用于存放监控程序和 BASIC 解释程序,16KRAM 用来存放 BASIC 或 PASCAL 操作系统的一部分。

存储体的切换是通过改变软开关的状态来实现的。这可以实现三种选择:一是将 \$D000—\$FFFF 这 12K 空间切换(映射)到 ROM 还是 RAM;二是允许写 RAM 还是禁止写 RAM;三是将 \$D000—\$DFFF 这 4K 空间切换到第一个 4KRAM 还是第 2 个 4KRAM。软开关的地址为 \$C080—\$C08F。如果软开关地址最低位为“1”,则为写 RAM;为“0”时禁止写 RAM。如果软开关地址的 A_3 位为“1”时,则选 RAM 第一存储体中的 4K;若为“0”,则选 RAM 第二存储体中的 4K 等等。

设置软开关时应注意:

- 在 \$D000—\$DFFF 空间的两个 RAM 体,不能选一个读,另一个写,只能选择读、写同一个 RAM 体。

- 对 \$D000—\$FFFF 空间,不能选择读其中一部分空间的 ROM,而读剩余部分的 RAM。

- CEC—I 在加电或复位时,\$D000—\$FFFF 的初始状态被设置为允许读 ROM,允许写 RAM,4KRAM 空间在第二体。

例如,读两次 \$C083,可以接通 1#、3#、4# 3 个 4K 体,并置为可读/写状态(可用 LDX \$C083);用两次 LDX \$C08B 指令接通 2#、3#、4# 3 个 4K 体,并置这 12K 空间为可读/写状态。在选通上述 RAM 时,ROM 将不起作用。而在切断 RAM,恢复 ROM 时,可用 LDA \$C081 指令处理。

初级程序员级水平考试辅导问答

计算机硬件基础知识(续)

电子部十五所(100083) 顾育麒

[第4题] 一个用补码表示的二进制数,最高位为符号位,当符号位为“0”(即正数)时,符号位后面的数值是真实的值;当符号位为“1”(即负数),符号位后面的数值不是真实值,而是补码值,必须对它按位取反,然后在最低位加1,才得到真实的值,而且一定是一个负数。

$$[X]_{\text{补}} = 01110010$$

$$X = +(1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3) \\ = +(64 + 32 + 16 + 2) = +114$$

$$[Y]_{\text{补}} = 10110101$$

符号位 按位取反

$$1001010 + 1 = 1001011$$

$$Y = -1001011 = -(1 \times 2^6 + 1 \times 2^3 + 1 \times 2^1 + 1 \times 2^0) \\ = -(64 + 8 + 2 + 1) = -75$$

[第5题] 对于二进制数的代码 10101101,若把它理解为无符号整数时,八个二进制数字都是数值部分,利用二进制数转换成十进制数的方法,能得到其对应的十进制数:

$$10101101 = 1 \times 2^7 + 1 \times 2^5 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^0 \\ = 128 + 32 + 8 + 4 + 1 = (173)_{10}$$

若把它理解为用补码表示的有符号整数时,最高位是符号位,这个二进制数的代码的最高位是“1”,所以它是一个负数。这个8位的二进制数代码是一个负数的补码表示法,现在要求出其对应的十进制数。

$$[X]_{\text{补}} = 10101101$$

符号位 按位取反

$$1010010 + 1 = 1010011$$

$$X = -1010011 = -(1 \times 2^6 + 1 \times 2^4 + 1 \times 2^1 + 1 \times 2^0) \\ = -(64 + 16 + 2 + 1) = -(83)_{10}$$

[第6题] 当X采用原码表示时,8位定点二进制整数所能表示的最大正数、最小负数:

$$\text{最大正数 } [X]_{\text{原}} = 01111111$$

(符号位)

$$X = (1111111)_{10} = (127)_{10}$$

$$\text{最小负数 } [X]_{\text{原}} = 11111111$$

(符号位)

$$X = (-1111111)_{10} = (-127)_{10}$$

当X采用补码表示时,8位定点二进制整数所能表示的最大正数、最小负数:

$$\text{最大正数 } [X]_{\text{补}} = 01111111$$

(符号位)

$$X = (1111111)_{10} = (127)_{10}$$

$$\text{最小负数 } [X]_{\text{补}} = 10000000$$

(符号位)

按位取反

$$1111111 + 1 = 10000000$$

$$X = (-10000000)_{10} = (-128)_{10}$$

在思考这个问题时,可能错误地认为用补码表示的最小负数是 $[X]_{\text{补}} = 11111111$,如果将补码转换成真值X

$$[X]_{\text{补}} = 11111111$$

(符号位)

按位取反

$$0000000 + 1 = 0000001$$

$$X = (-0000001)_{10} = (-1)_{10}$$

显然,在8位定点二进制整数范围内,还能表示比 $(-1)_{10}$ 更小的负数,这个数不是最小负数。

[第7题] 如果逻辑表达式复杂,则电路设计时使用的器件就多,电路复杂;如果逻辑表达式简单,则电路设计时使用的器件就少,电路简单,因此在电路设计时要将逻辑表达式化简。

逻辑表达式化简到什么程度,可以认为是“最简”的逻辑表达式呢?

首先,乘积项的个数应该是最少的;

其次,在满足乘积项个数最少的条件下,要求每一个乘积项中的变量最少。

常用的化简方法是公式化简法。

下面列出逻辑代数的基本公式:

1. 关于变量和常量关系的等式:

$$[\text{公式 } 1] A + 0 = A \quad [\text{公式 } 1'] A \cdot 1 = A$$

$$[\text{公式 } 2] A + 1 = 1 \quad [\text{公式 } 2'] A \cdot 0 = 0$$

$$[\text{公式 } 3] A + \bar{A} = 1 \quad [\text{公式 } 3'] A \cdot \bar{A} = 0$$

2. 满足交换律、结合律、分配律

$$[\text{公式 } 4] A + B = B + A \quad (\text{交换律})$$

$$[\text{公式 } 4'] A \cdot B = B \cdot A \quad (\text{交换律})$$

$$[\text{公式 } 5] (A + B) + C = A + (B + C) \quad (\text{结合律})$$

$$[\text{公式 } 5'] (A \cdot B) \cdot C = A \cdot (B \cdot C) \quad (\text{结合律})$$

$$[\text{公式 } 6] A \cdot (B + C) = A \cdot B + A \cdot C \quad (\text{乘对加的分配律})$$

$$[\text{公式 } 6'] A + B \cdot C = (A + B) \cdot (A + C) \quad (\text{加对乘的分配律})$$

3. 逻辑代数的一些特殊规律

$$[\text{公式 } 7] A + A = A \quad (\text{重叠律})$$

[公式 7'] $A \cdot A = A$ (重叠律)

[公式 8] $A + \bar{A} = \bar{A} \cdot \bar{B}$ (反演律)

[公式 8'] $A \cdot \bar{B} = \bar{A} + \bar{B}$ (反演律)

[公式 9] $\bar{\bar{A}} = A$

当我们初步掌握了这些基本概念和基本公式以后,就可以做这道题了。

$$\begin{aligned} F &= A \bar{B}C + A \bar{B} \bar{C} + ABC + A \bar{B} \\ &= AC(\bar{B} + B) + A(\bar{B} + \bar{C}) + A \bar{B} \\ &= AC + A \bar{B} + A \bar{C} + A \bar{B} \\ &= AC + A \bar{B} + A \bar{C} \\ &= A(C + \bar{C}) + A \bar{B} \\ &= A + A \bar{B} \\ &= A(1 + \bar{B}) \\ &= A \end{aligned}$$

在化简的第二步,应用了反演律。

$\bar{A} \cdot \bar{B} = \bar{A} + \bar{B}$ [公式 8]

$A + \bar{B} = \bar{A} \cdot \bar{B}$ [公式 8']

使得逻辑表达式的第二步

$$A \bar{B}C = A(\bar{B} + \bar{C}) = A \bar{B} + A \bar{C}$$

为下一步的化简工作,提供了有利条件。

反演律是逻辑代数中的一组重要公式,又称为摩根定理,在电路设计、化简逻辑表达式时经常要用到这组公式。为了便于读者理解,并能灵活运用它,这里用列出真值表的方法,证明它的正确性,见表 1

表 1 真值表

A	B	$A \cdot B$	$\bar{A} \cdot \bar{B}$	\bar{A}	\bar{B}	$\bar{A} + \bar{B}$
0	0	0	1	1	1	1
0	1	0	1	1	0	1
1	0	0	1	0	1	1
1	1	1	0	0	0	0

从表 1 中可以看到 $\bar{A} \cdot \bar{B}$ 和 $\bar{A} + \bar{B}$ 在变量 A、B 的任何组合状态下都有相同的取值,这就证明了 [公式 8] $\bar{A} \cdot \bar{B} = \bar{A} + \bar{B}$ 的正确性。

用同样的方法列表 2。

表 2 真值表

A	B	$A + B$	$\bar{A} \cdot \bar{B}$	\bar{A}	\bar{B}	$\bar{A} \cdot \bar{B}$
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

从表 2 中可以看到 $A + B$ 和 $\bar{A} \cdot \bar{B}$ 在变量 A、B 的任何组合状态下都有相同的取值,这就证明了 [公式 8'] $A + B = \bar{A} \cdot \bar{B}$ 的正确性。

反演律可以推广到多个变量的场合,例如:

$$A \cdot B \cdot C \cdot \dots \cdot M = \bar{A} + \bar{B} + \bar{C} + \dots + \bar{M}$$

$$\bar{A} + \bar{B} + \bar{C} + \dots + \bar{M} = \bar{A} \cdot \bar{B} \cdot \bar{C} \cdot \dots \cdot \bar{M}$$

在 1989 年和 1990 年的全国初级程序员水平考试的试题中,都有化简逻辑表达式的试题,难度适中,因此,在复习时要注意掌握逻辑代数的基本公式和逻辑代数的一些特殊规律,同时要做一些练习题,以帮助理解基本公式的应用方法。

[第 8 题] 运用逻辑代数的基本公式进行化简:

$$AB + \bar{A}C + BC = AB + \bar{A}C + (A + \bar{A})BC$$

这里采用了公式化简法中的“配项法”,在第三项配以因子 $(A + \bar{A})$,由 [公式 3] $A + \bar{A} = 1$,所以逻辑表达式的含义不变。

$$\begin{aligned} AB + \bar{A}C + BC &= AB + \bar{A}C + (A + \bar{A})BC \\ &= AB + \bar{A}C + ABC + \bar{A}BC \\ &= (AB + ABC) + (\bar{A}C + \bar{A}BC) \\ &= AB(1 + C) + \bar{A}C(1 + B) \\ &= AB + \bar{A}C \end{aligned}$$

这道题的答案应选择 (2)

[第 9 题] 中央处理器(CPU)是计算机的重要组成部分,它由运算器、控制器组成,能执行规定的计算机基本操作指令。

控制器是计算机的指挥中心,它的基本功能是取指令,分析指令格式,按指令的要求产生相应的控制电平,指挥协调计算机内各部件工作,完成指令规定的操作。

指令寄存器的作用是保存当前正在执行的一条指令。这条指令是新从存储器取出来的。

程序计数器的作用是用来存放后继指令地址。也就是说,它存放一个地址。所谓后继指令是指后面要继续执行的指令,按程序计数器提供的地址,从相应的存储单元中取出来的内容,就是要执行的指令和要操作的数据。一般指令是顺序存放在存储器中的,所以,程序计数器的内容自动增加,以指向下一条指令的地址。程序计数器也叫做指令地址计数器。

控制器根据人们预先编写好的程序,依次从存储器中取出指令,放在指令寄存器中,通过指令译码器进行译码分析,确定应该进行什么操作,通过微操作控制线路产生相应的微操作控制信号,在确定的时间,往确定的部件,发出确定的信号,使运算器和存储器等部件自动而协调地完成该指令所规定的操作。当这条指令完成以后,再按程序计数器中的地址从存储器中取出下一条指令,并照此同样地分析与执行该指令,如此重复,直到完成所有的指令为止。

这道题的第一个填空应选择答案 (2),第二个填空应选择答案 (1)。

[第 10 题] 指令是计算机用以控制各个部件协调动作的命令。每种机型都有自己的一套指令,称为该机型的指令系统。指令由两部分组成:操作码和地址码。操作码指明这条指令应执行的操作种类,地址码指明这条指令参与运算的数或数所在的单元地址。

查找指令操作数的方法就是寻址方式。最简单的寻址方式是:指令中的地址码就是操作数在主存储器

中存放的有效地址。允许访问主存储器的最大空间决定于指令中地址码的位数。例如地址码 16 位表示可直接访问的内存最大容量为 2^{16} , 即 64K。对于指令字较短的指令, 能够访问的操作数是有限的。为了扩大指令允许访问的内存空间而又不增加指令字的位数, 提供编制程序的灵活性, 在计算机指令系统中采用多种不同的访问内存存储器的方式, 对指令的地址码, 进行变换, 求得操作数的有效地址, 然后才去访问内存存储器。

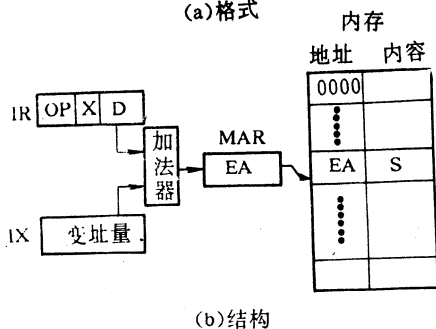
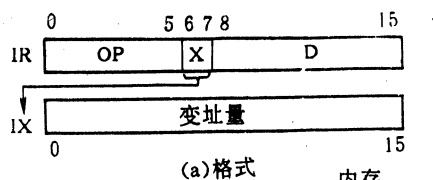
因此, 这道题的答案是(2), 指令系统中采用多种不同的寻址方式的主要目的是缩短指令长度, 扩大寻址空间, 提高编程的灵活性。

一个指令系统中有若干种寻址方法, 例如: 直接寻址、变址寻址、间址寻址等等。某一条指令使用哪一种寻址方式, 通常要在地址码字段中专门设置一个寻址方式字段。由该字段中的二进制数码指定具体的寻址方式。

[第 11 题] 为了扩大可访问的主存空间, 希望增加有效地址的位数。硬件上又不允许随意增加指令字的长度和形式地址的位数。因此, 提出一种新的寻址方式—变址寻址。

这种寻址方式的实质是进行一次地址加法, 使形式地址(位数较少)与某个指定的寄存器内容相加, 求得有效地址。这个被指定的寄存器叫做变址寄存器, 变址寄存器的内容叫变址量。

变址寄存器的位数较多, 与形式地址相加所得到的有效地址位数也较多, 因此可访问的地址空间就扩大了。例如, 8 位形式地址与 16 位变址量相加, 得到 16 位的有效地址, 则指令可访问的主存空间为 $2^{16}=64K$ 。



IR—指令寄存器
IX—变址寄存器
MAR—内存地址寄存器
EA—有效地址
OP—操作码
X—变址寻址特征
D—形式地址
S—操作数

图 1

变址寻址方式的示意图如图 1 所示。这道题的答案应选择(3)

[第 12 题] 当指令地址码指出的主存单元的内容不是操作数, 而是存放操作数的地址时, 则称由指令地址码所决定的地址为间接地址。

要寻找操作数的地址, 必须按照指令给出的间接地址去访问主存, 然后把这个主存单元的内容当作操作数的存放地址。这种寻址方式叫做间接寻址方式。

指令中必须给出间接寻址的标志, 以便与直接寻址相区别。

直接寻址时, 由指令地址码所决定的地址, 就是操作数的地址, 这个地址可以用直接寻址、变址寻址、相对寻址等方法来决定; 间接寻址时, 由指令地址码决定的只是某个操作数地址的地址。这个地址也可以用直接、变址等方法来确定。

间接寻址指令应包括如下几部分:

OP	I	X	D
----	---	---	---

OP 操作码
I 间接寻址特征
X 变址寻址特征
D 形式地址

当 $I=0$ 表示直接寻址; 当 $I=1$ 表示间接寻址。

间接地址 $IA=(IX)+D$

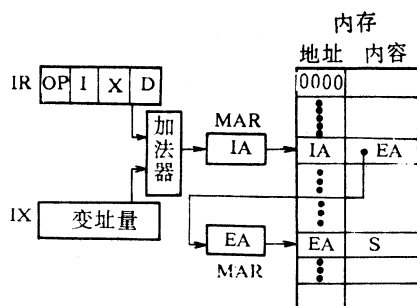
其中 (IX) 表示变址寄存器的内容。当不变址时, $(IX)=0$

操作数的有效地址 $EA=(IA)$

间接寻址的过程如图 2 所示。

这道题的答案应选择(2), 在间接寻址方式中, 操作数在主存单元中。

[第 13 题] 对于微型计算机来说, 空气中的相对湿度不能超过 80%, 若超过这个值则会发生结露现象



IR—指令寄存器
IX—变址寄存器
IA—间接地址
MAR—内存地址寄存器
S—操作数
EA—有效地址

图 2

象,使元器件受潮变质,甚至发生短路,损坏机器;若空气中的相对湿度小于 20%,则因过于干燥,使微机系统容易产生静电干扰,引起机器的错误动作。因此,机房的相对湿度应控制在 20%~80%之间。这道题的答案应选择(2)。

[第 14 题] MIPS(Million Instructions Per Second)值是人们用于表示计算机性能的一种方法。若其值为 1MIPS,表示计算机每秒钟能执行 100 万条指令。

不同的计算机制造厂,在设计计算机时,都有一定的针对性,有的侧重于科学计算;有的侧重于商业管理,有的侧重于图形处理。所以制造厂在选择测试 MIPS 值的基准程序时,各有不同。因此不同厂家制造的计算机,MIPS 值的含义不同,不一定能说明性能的高低。因此,在选购机器时,不仅要看 MIPS 值,还要了解计算机的其它性能,综合比较,才能区别性能高低。

同一个厂家生产的机器,用相同的基准程序测试得到的 MIPS 值,具有可比性。MIPS 值越大,说明机器的速度越快。

表 3 列出 DEC 公司的 VAX 系列计算机的 MIPS 值。

表 3 列出 VAX 11/780 计算机的 MIPS 值为 1.06,它表示每秒能执行 106 万条指令。DEC 公司的 VAX 系统均以 VAX 11/780 机的速度作为衡量该公司其它机器的性能单位,用 VAX 性能单位 VUP(VAX Unit of Performance)表示。例如:VAX 9000 计算机单个中央处理机的性能等于 40 倍 VUP;VAX 9000—440 型计算机为 150 倍 VUP;VAX 6000 型计算机系统性能范围可以从 2.6 倍 VUP 增加到 36 倍 VUP。

表 3 DEC 公司计算机的 MIPS 值

系统 性能	VAX 11/780	VAX 8300	VAX 8700	VAX 6000	VAX 9000
MIPS	1.06	1.70	6.36	3.0~ 38.2	42.4 ~ 159.0
VUP	1.0	1.6	6.0	2.8~ 36	40 ~ 150

这道题的答案应选择(2)

[第 15 题] MTBF(Mean Time Between Failure)是计算机的可靠性指标,称为平均无故障工作时间。有的资料上称为平均故障间隔时间,其实两种说法的含义是一样的。

可靠性是指在规定的时间内,规定的条件下完成规定功能的能力。这是一个定性的要求。能力究竟有多大?要有定量的指标来描述。

可靠度是指在规定的时间内、规定的条件下完成规定功能的成功概率,成功概率是个定量指标,它是一个统计量,可以度量和比较。对于计算机系统来说,可靠度定义中的“完成规定功能”是指计算机系统的技术性能,在生产实践中,用计算机系统的平均无故障工作

时间 MTBF 来表示。

因此,这道题的答案应选择(2)。

[第 16 题] 计算机的内存储器按性能和用途分类,可以分成两大类:只读存储器 ROM 和随机存取存储器 RAM。

只读存储器 ROM 的特点是:它存放在其中的信息是固定的,即使由于关机或停电,切断电源后,它所存储的信息不会丢失。再次接通电源后,它所存储的信息立即可以读出使用。要求在通电后立即可以取用的指令和数据都预先存放在 ROM 中。例如计算机内存存储器中常驻的监控程序、引导程序、加载程序、管理程序、常用的数据表格等。

随机存取存储器 RAM 的特点是:既可以“读出”其中存放的信息,又可以将新的信息“写入”其中。一旦断电,则所存储的信息全部丢失。再次接通电源后,其中已没有有用的信息。若开机后不断电,计算机处于正常工作状态,则“写入”的信息就能保持。

只读存储器 ROM 所存储的信息,是事先写入的,根据写入信息的方式不同,又可以分为下列几种:

1. ROM:不可编程的只读存储器。这种 ROM 所存储的信息,是在芯片生产的最后工序时由生产工厂“写入”的,芯片封装后,无法更换信息,这种 ROM 使用可靠,成本低。

2. PROM:可编程的只读存储器。它所存储的信息可以在芯片封装完成后,再编程写入。常用的方法是“烧入法”,用编程用的强电流脉冲,将芯片内需要烧断的熔丝烧断,熔断的位为逻辑 0,未熔断的位为逻辑 1。信息一旦被写入,熔丝已烧断,再想改写也不可能了,写入的信息就永久固定下来。

3. EPROM:可擦除改写的只读存储器。用户不但可以“写入”信息,还可以用紫外光通过封装上的玻璃窗口照射芯片,使各位全部恢复为 1,亦即将“写入”的信息清除掉,以便重新写入信息。

4. EEROM 或 E² ROM:可电擦除只读存储器。

这道题的第一个填空应选择答案(2);第二个填空应选择答案(1)。如果题型改变,在供选择的答案中有 PROM、EPROM 等,那么也属于第一个填空可选择的答案。

[第 17 题] 目前在微型计算机上配置的软盘驱动器规格如表 4 所示。

表 4 磁盘驱动器的品种和规格

直径(英寸)	说 明	容 量
5.25	单面	160KB/180KB
5.25	双面	320KB/360KB
5.25	高密度	1.2MB
3.5	双面	720KB
3.5	双面	1.44MB

微型计算机上使用的软盘片品种和规格如表 5 所示。

表 5 软盘片的品种和规格

直径 (英寸)	说 明	存储容量	磁道 数	磁头 数	每道 扇区数	每个扇 区字节 数
5.25	单面,倍密度	160KB/180KB	40	1	8/9	512
5.25	双面,倍密度	320KB/360KB	40	2	8/9	512
5.25	双面,高密度	1.2MB	80	2	15	512
3.5	双面	720KB	80	2	9	512
3.5	双面	1.44MB	80	2	18	512

用 DOS 命令到软盘片中去读写时,必须考虑软盘片和软盘驱动器的兼容性。具体规定如下:

1. 160KB/180KB 单面 5.25 英寸软盘驱动器。

可以在这种软盘驱动器上读写的软盘片,只有 160KB/180KB 单面,倍密度 5.25 英寸软盘片一种。

2. 320KB/360KB 双面 5.25 英寸软盘驱动器。

可以在这种软盘驱动器上读写的软盘片有: 160KB/180KB 单面,倍密度 5.25 英寸软盘片; 320KB/360KB 双面,倍密度 5.25 英寸软盘片。

3. 1.2MB 高密度 5.25 英寸软盘驱动器。

可以在这种软盘驱动器上读写的软盘片有: 160KB/180KB 单面,倍密度 5.25 英寸软盘片; 320KB/360KB 双面,倍密度 5.25 英寸软盘片;

1.2MB 高密度 5.25 英寸软盘片。

4. 720KB 双面 3.5 英寸软盘驱动器。

可以在这种软盘驱动器上读写的软盘片只有 720KB 双面 3.5 英寸软盘片一种。

5. 1.44MB 双面 3.5 英寸软盘驱动器。

可以在这种软盘驱动器上读写的软盘片有:

720KB 双面 3.5 英寸软盘片;

1.44MB 双面 3.5 英寸软盘片。

这道题的答案应选择(2)、(3)。

(上接第 29 页)

程序、显示、前置放大器、A/D 等器件的功耗,才能真正达到高性能、低功耗的效果。

这里主要是有两方面要注意:

(1) 改变键盘扫描方式

一般智能仪器中 CPU 处理数据占的整个时间并不多,而扫描键盘、显示是经常不断进行的。这种用程序查询的方式扫描键盘和显示器很不经济。这时 CPU 永远处于忙的状态,不可能进入空闲状态。为了减少功耗,让 CPU 在平时处于空闲方式,并开放外部中断。一般将键盘设计成中断方式,利用外部中断 0 控制键盘,只要有键按下,向 CPU 请求中断,使 CPU 退出空闲状态,进入正常状态。CPU 响应中断后,为了防止重新按键,首先禁止外部中断 0,然后进行键盘扫描,根据按

[第 18 题] 磁带机的数据传送速度=读写磁带的走带速度×磁带记录密度。

磁带机的数据传送速度用每秒传送的比特数 BPS (Bits Per Second) 表示;读写磁带的走带速度用每秒走带的英寸数 IPS 表示;磁带记录密度用每英寸存储的比特数 BPI (Bits Per Inch) 表示。

这台磁带机的数据传送速度为:

$$45 \times 800 = 36000 \text{ BPS}$$

这道题的答案应选择(2)。

[第 19 题] 在输出设备中,打印输出设备的应用十分广泛,几乎所有的计算机系统都配置有打印机。

打印输出设备按其打印方法的不同,可分为击打式和非击打式两种。

击打式打印机是利用打印针击打色带和打印纸,而打印出由点阵构成的字符、汉字、图形。点阵针式打印机就是这种类型。它的打印头由若干根针组成。常用的有 9 针、24 针,通过打印驱动程序控制各个不同位置的针动作或不动作,打印出各种字符或图形。

非击打式打印机是采用物理或化学方法印出字符。它具有高速、无噪声印字等优点。激光印字机、热敏印字机、喷黑印字机等都是这种类型。常用的激光印字机是将激光扫描技术和电子照相技术相结合的非击打式打印输出设备。

这道题的第一个填空应选择答案(3);第二、三个填空应选择答案(1)(2)。

[第 20 题] 这道题的答案应选择(3)。

“计算机辅助”的概念是指依靠计算机技术来得到比人工方法更好、更快、更精确的设计、教学、管理、制造过程。

CAD (Computer Aided Design) 是计算机辅助设计的简称。它的含义是使用计算机系统来辅助一项设计的形成、修改、分析和优化。它的应用范围很广,例如:机械 CAD、建筑 CAD、电气设计 CAD 等。

下的相应键去处理相应的功能。处理完后返回,并开放中断 0 和置 PCON. 0=1,使 CPU 重新进入空闲方式。这样可以大大降低系统的功耗。硬件一般用多个二极管组成或门(对逻辑 0 来说)或者用多输入端的与门解决。

(2) 使用液晶显示

用 LED 作为数字、地址的显示时,一般耗电 200mA—300mA。而要使整个系统降低功耗,一般使用液晶显示。液晶分静态和动态两种,两种驱动电路都比较复杂,具体电路将在另一节讲。

总之我们认真做好各个环节节电,使整个 80C31BH 应用系统总耗电在 10mA 以下是完全可行,也是不困难的。



低功耗单片机最小系统

合肥中国科技大学计算中心(230026) 张培仁 刘振安

CMOS 电路具有低功耗,低速度,高抗干扰能力。它一般比 TTL 电路功耗小 100 倍,速度慢 10 倍。近几年来出现既具有 CMOS 电路优点又兼有 TTL 的速度的 CHMOS 电路。相应出现了如 80C31BH,80C51BH 等单片机,也越来越受人们的重视。特别象野外,井下,无人值守监测站等非常需要低功耗的单片机系统。

对于片内无程序存储器的低功耗单片机 80C31BH 必须配上片外程序存储器。这时要求配置外围芯片也应该是低功耗芯片。这种单片机最小系统由三片组成即 80C31,27C64,74HC373。电路如图 1 所示。

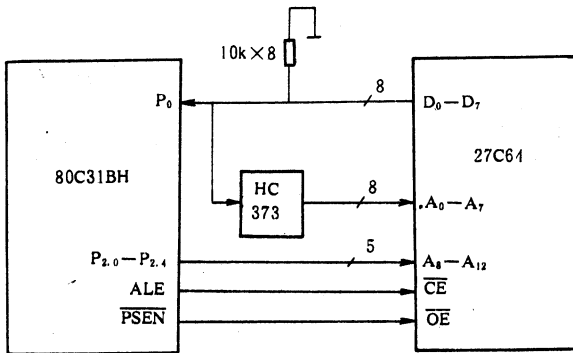


图 1

单片机在节电状态运行功能,工作方式如何设定,以及在低功耗系统中应注意什么,我们对这些问题作一个简单介绍。

1. 单片机节电运行功能

80C51BH,80C31BH 具有正常,空闲、掉电三种工作状态。它们速度快,功耗是 8051,8031 的 1/10。80C51BH,80C31BH 的 CPU 不是静态而是动态,但 80C51,80C31 片内的 RAM 是静态的。故时钟频率不能太低。

几种工作方式耗电量对比表如表一所示。

表一

时钟频率	工作方式	8031	80C31BH
12MHz	正常	125mA, +5V	24mA, +5V
12MHz	空闲	没有	3mA, +5V
	掉电	没有	50μA, 2V

空闲和掉电二种节电方式均可通过软件来选择运行,仅需要正常工作时,才进入正常运行,其它时间均保持在空闲或掉电方式下。这样大大减少单片机的功耗。

空闲方式下,CPU 停止工作,而 RAM,定时器/计

算器,串行口和中断系统继续工作。

2. CHMOS 单片机的节电工作方式

CHMOS 单片机提供三种工作状态是由图 2 的硬件电路实现的。

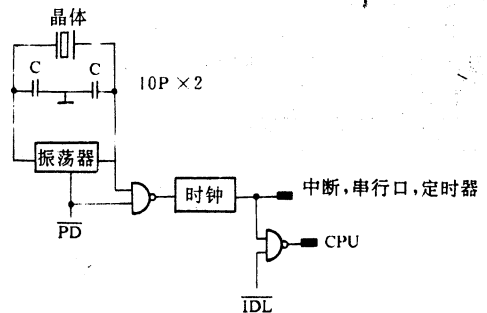


图 2

由图可见,当 $\overline{IDL}=0$ 时,时钟仍继续工作,中断系统,串行口,定时/计数器电路继续由时钟驱动,但此时时钟信号不送往 CPU,即 CPU 处等待状态,称之为空闲工作方式。当 $\overline{PD}=0$ 时振荡器停止工作,只有片内 RAM 和 SFR 的内容被保存,这就是掉电工作方式。这两种工作方式都是由 SFR 中的电源控制寄存器 PCON 设定的。

MSB

LSB

SMOD	-	-	-	GF1	GF0	PD	IDL
------	---	---	---	-----	-----	----	-----

PCON 是电源控制寄存器,地址为 87H。各位定义如下:

SMOD(PCON. 7): 串行口控制用的波特率倍增位。

GF1,GF0: 通用标志位。

PD(PCON. 1): 掉电方式位。当 $\overline{PD}=1$ 时,进入掉电方式。

IDL(PCON. 0): 空闲方式位。当 $\overline{IDL}=1$ 时,进入空闲工作方式。

若 PD 和 IDL 同时为 1,则先进入掉电方式。

复位时,各控制位全为 0。

PCON 不能位寻址,只能进行字节寻址。

空闲工作方式

空闲工作方式下,CPU 进入等待状态。此时仍继续驱动中断系统,定时/计数器,串行口。CPU 内部的状态,堆栈指针(SP),程序计数器 PC,程序状态字(PSW),累加器,片内 RAM 的状态完整保留下来。这时也称之为冻结工作方式。

单片机处于空闲状态时, $\overline{\text{ALE}}$, $\overline{\text{PSEN}}$ 电平为“1”当我们外部取指时, 也就象图 1 所示 HCMOS 的三片最小系统时, 进入空闲状态 P0 口呈高阻状态, P2 口状态是进入空闲前 PC 高八位地址的状态 (PCH), P1, P3 口是进入空闲状态之前的状态。CPU 从空闲状态回到正常工作方式的条件是用中断或硬件复位。

掉电方式

在掉电工作方式 ($\overline{\text{PD}}=0$) 下, 由于 $\overline{\text{PD}}=0$ 的作用, 片内振荡器停止工作, 因此单片机所有运行状态都停止, 只有片内 RAM 中的数据保存起来。

处于掉电时 $\overline{\text{ALE}}$, $\overline{\text{PSEN}}$ 均为低电平 (逻辑 0)。I/O 口引脚状态见表二。

退出掉电工作方式, 只能用硬件复位, 复位操作时将重新定义所有 SFR, 但不改变片内 RAM 的内容。

表二

引脚	空闲	掉电
ALE	1	0
$\overline{\text{PSEN}}$	1	0
P0	高阻	高阻
P1	SFR 数据	SFR 数据
P2	PCH	SFR 数据
P3	SFR 数据	SFR 数据

(有片外 EPROM, 是外部取指)

3. 低功耗单片机系统要注意的问题

80C51BH, 80C31BH 有很多优点。但在实际的应用中尚要注意以下几个问题:

(1) 可控硅导通现象

由于 CMOS, CHMOS 电路半导体工艺容易产生 pnpn 寄生结构。这种结构正好构成可控硅, 一旦产生闭锁, CMOS 的电流可大于 100mA。这种短路现象无法用软件恢复, 只有断电后才可恢复。但 80C31 由于工艺改进, 已经大大减少了这种现象。一般在电源中串接一个小于 50 Ω 的电阻, 以使由于工艺制作原因产生的可控硅触发电流达不到触发水平 (即达不到产生维持可控硅现象的触发电流), 从而避免了可控硅失控现象。

(2) 未使用管脚的处理。

P1, P2, P3 口因片内都有上拉电阻, 所以在未使用时可不作任何处理。但 P0 在片内无上拉电阻, 必须在片外接上拉或下拉电阻。这是因为当复位信号过后, 由于 P0 口用作数据/地址总线, 其引脚上的电平总是有定义的。然而当 80C31BH 处于空闲或掉电方式时, 由于它不执行取指令操作, 故当不外加上拉电阻或下拉电阻时, P0 口引脚悬空, 一旦恢复正常工作方式, CPU 要从 P0 口取指令可能会出错。

上拉和下拉电阻一般在 2K Ω ~50K Ω 之间, 一般可选 10K Ω 。电阻太小时, 对输出低电平影响较大, 功耗也大些; 电阻太大对 P0 口的高电平影响比较大。一般接下拉电阻比上拉电阻可减少功耗效果是一样的。

(3) P1, P2 和 P3 口的驱动能力

80C31BH 中 P1, P2, P3 口都接了三个上拉电阻 (片内接的), 电阻分强, 中, 弱。强、中都是由脉冲控制接通的, 弱是一直接上的 (大小一般为 20K Ω)。这三个电阻组成端口的驱动能力。驱动电流最大在 650 μ A 左右。

(4) 主振频率对功耗的影响

功耗随着主振频率的提高而增加。频率越高, CMOS 管处于放大区的时间占一个周期的整个时间越多, 功耗也随之增加。理论上在 500KHz 时功耗最小, 频率太低不行, 因为 80C31 中 CPU 是动态的, 数据会丢失。

(5) 在 80C31BH 的最小应用系统中对 27C64 的 $\overline{\text{CE}}$ 片选信号处理。

在空闲状态时, 27C64 的片选信号 $\overline{\text{CE}}$ 不接地, 而与 80C31BH 的 ALE 端相连, 以便在 80C31BH 进入等待工作方式时, ALE 变为高电平, 使 27C64 进入低功耗备用状态, 如图 1 所示。

在掉电方式时, 如系统的备用电源仅给 80C31BH 供电, 那么掉电时, $\overline{\text{ALE}}$, $\overline{\text{PSEN}}$ 应为低电平。但如果在电源掉电期间仍给 EPROM 供电, 则需禁止 27C64。在正常工作时 P2.7 输出为 0, 选中 27C64。掉电时 P2.7 = 1 不选中 27C64。如果既有掉电方式也有空闲方式时 27C64 的 $\overline{\text{CE}}$ 端接法如图 3。

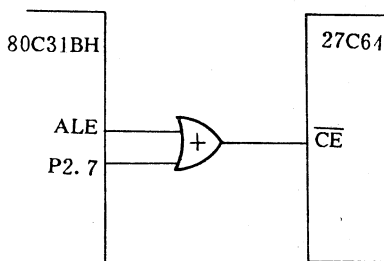


图 3

(6) 空闲前的准备

进入空闲状态前要把准备允许哪个中断源激活 CPU 确定下来, 空闲前要用软件置这些中断源为允许, 当然总中断源也一定开放。

空闲后要端口处于什么状态, 也要在启动空闲前写入端口。原则有两个: 一是维持整个系统正常工作, 二是尽可能降低功耗。例如 P1 口中有几位在应用系统中是控制某些继电器的, 则要置 1。进入空闲时, 这些还是要保持 1, 而其它一些空闲时可能没用, 可以置 0。

空闲时一般不选中 27C64, PC 值在空闲时一般不写入 0000H。

(7) 掉电前的准备和掉电后恢复时的复位时间。

掉电前的准备工作基本和空闲前准备工作相似。掉电后再复位, 信号必须有足够时间使主振荡器重新振荡并稳定下来, 一般需要 10ms 左右。

对于一个应用系统来讲, 只考虑以 80C31 的最小系统的功耗还是不够的, 还应考虑整个系统如键分析

(下转第 27 页)

新一代超高集成度 Z80 微处理器系列

ZILOG 中国代理

中国电子器材公司特区公司 董伯明

Z80 是世界上和我国流行最广的一种八位微处理器。我国广大的工程技术和科研人员几乎都学过和应用过该系列产品,时至今日,我国大专院校有关计算机的教材中,还是以 Z80 系列产品为例进行教学。许多工业控制、尖端电子产品、各种智能终端等等场合仍然大量应用 Z80 系列产品。

近年来,随着 VLSI 工艺水平的不断提高,Zilog 公司采用 Assps(特殊应用标准产品)的先进设计手段,推出了软件与原 Z80 完全兼容的超高集成度 Z80 处理器系列。该系列产品具有高度集成,高度灵活,高速,高性能等特点,又具有很高的性能/价格比,故而在世界上已广泛地应用在各种控制与通信的领域。目前,我国刚开始接触,为了加速这一系列的先进产品在我国推广应用,我们准备从今年开始,继续在“电子与电脑”的“学装微电脑”专栏上介绍这一系列产品的性能及应用技术。相信原有通用型 Z80 的广大用户很快就会步入与掌握新一代超高集成度 Z80 微处理系列。附表介绍新一代超高集成度 Z80 微处理器性能(见封三)。

该系列的产品共有九个品种,采用 NMOS 和 CMOS 二种工艺生产,凡产品型号中有 C 的均是 CMOS 工艺生产的,而没有 C 的是采用 NMOS 工艺。其中:

Z84C01 是该系列产品中最基本的一种产品,它由 Z80 CPU 与 CGC(时钟发生,控制器)组成。

Z84C50 是在 Z84C01 的基础上再加 1KB 的静态 RAM 和等待状态发生器二个模块。

Z84C90 是 Z80KIO(Killer I/O),称为通用的或专用型的多用途 Z80 I/O 电路,意味着只要有了这片电路后,可以取代各种 I/O 电路,它内含二个独立的同步或异步的串行通道,三个 8 位的并行端口,四个独立的定时器/计数器,一个并行接口适配器和片内振荡器,时钟驱动电路等。

Z84011/C11 是有 NMOS 和 CMOS 二种工艺生产的产品,它内含 Z80 CPU,时钟发生、控制器,四个独立的定时器/计数器,五个独立的 8 位并行端口,尚有 WDT(Watch Dog Timer)控制定时器模块,该模块启动后,对于由电源或其它硬件而造成的软件出错,具有缩短出错周期的作用,换言之具有抗干扰的功能。

Z84013/C13 亦采用 NMOS 与 CMOS 二种工艺生产,称之为智能化外设控制和增强型智能化外设控制器。该芯片内含 Z80 CPU,四个独立的计数器/定时器,二个独立的 SIO,一个时钟产生器,控制器,一个监时定时器。

Z84015/C15 亦是有 NMOS 和 CMOS 二种工艺生

产的智能外设控制器,但比 Z84013/C13 增加了二个独立的 8 位并行的 I/O 端口。

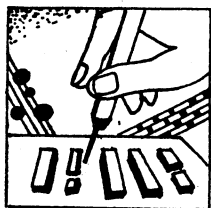
Z80180 是一块高性能的 Z80CPU 与外设接口的组合,称之为 MPU(微处理器单元)。该芯片内含一个增强型的 Z80 CPU 工作频率可高达 12.5MHz,有一个可达 1MB 的存储器管理单元 MMU,二个独立的直接存储器存取 DMA 通道,二个通用的异步发送器/接收器,二个 16 位的计数器/定时器,一个定时的串行 I/O 端口,片内振荡器,等待状态发生器,中断控制器等,实际上原来通用型 Z80 芯片组成的单板机的功能就由这一块芯片完成了,是用 CMOS 工艺制造的。

Z80280 是又一块高性能的微处理器单元 CPU,用 CMOS 工艺制造,它有三态的指令流水线,采用 16 位的 CPU 结构形式,机器码与通用型 Z80 微处理器完全兼容,它有页面存储器管理的单元 MMU,寻址能力可达 16MB,有 256B 的片内快速指令数据缓冲器 CACHE;有一个高性能的 16 位 Z-BUS 的总线接口或 8 位的与 Z80 CPU 兼容的总线接口;三个 16 位的计数器/定时器,四个直接存储器存取通道 DMA;一个全双工的通用型异步接收器/发送器;动态 RAM 的刷新控制器;片内振荡器控制器或外部时钟输入选择器;振荡时钟频率可高达 25MHz;亦有等待状态发生器模块,适用于各种智能外设、打印控制、智能终端等。

该系列的最后一块芯片是 Z80181,称为灵活随机控制器 SAC。该片内部有一个 Z80180 的大模块;另有二个 8 位通用型的端口,每位均可以软件来进行设定。一个串行的通信控制器;一个时钟发生器控制器,四个定时器/计数器;外接振荡器频率可高达 25MHz,用 CMOS 工艺制造,与通用型 Z80 的软件是 100% 的兼容,该芯片可以灵活地应用于各种智能型的外设、打印机控制器、传真机、调制/解调器、各种智能终端等等场合。

不难看出,上述各种产品使 Z80 进入一个崭新的时代,形成了一套超高集成度的新一代 Z80 系列,已在国外竞争激烈的商品经济市场中大量地应用。近来也被我国很多有关工程技术人员和大专院校所关注,相信不久的将来 Z80 超高集成度微处理器一定会应用到原来 Z80 应用的各种个方面,并不断开拓新的应用领域。

(编者按,有关咨询等事宜,请与深圳市振华路电子大厦,中国电子器材公司特区公司联系,邮编:518031,电话:384214,电挂:8410)。



学装微电脑

打印数字温度计

易齐干

本文介绍用热敏电阻检测温度、 $\mu P80$ 进行数据处理、简易打印机打印温度三部份组成的打印数字温度计。

1. 打印数字温度计组成

打印数字温度计系统框图如图 1 所示。热敏电阻检测温度，微电脑 1 将温度变换为数据，以简易 Cen-

tronics 方式把数据送给微电脑 2，微电脑 2 把数据变换为驱动打印机的信息，打印机打印出数字。微电脑 2 与打印机一般为一体，统称为打印机。

2. 微电脑将室温数值化

$\mu P80$ 微电脑将室温数值化可称简单的数字温度计。它的构成与工作原理请参阅“初级微电脑图解教程”。原理图如图 2 所示。

程”。原理图如图 2 所示。

3. 微电脑之间传送数据

微电脑之间传送信息常使用 Centronics 方式。图 3 所表示的微电脑 1 和微电脑 2 的连结是简易 Centronics 方式。

图中， $D_7 \sim D_0$ 是 8 位数据线。 \overline{STROBE} 是微电脑 1 向微电脑 2 传送数据的选通信号线。 $BUSY$ 是微电脑 2 通知微电脑 1 可以接收数据的信号线。微电脑 2 驱动打印机需要时间，打印机打印同时，不接收微电脑 1 传送的数据，如果微电脑 1 不分青红皂白地传送数据，微电脑 2 就发生混乱。 $BUSY$ 信号线的作用是避免出现混乱。

例如，微电脑 1 向微电脑 2 传送日文片假名“テ”和“.”的数据时序如图 4 所示。

(1) 首先，微电脑 1 检查微电脑 2 的 $BUSY$ 的电平。如果为低电平，微电脑 2 才可以接收数据。

(2) 微电脑 1 用 $D_0 \sim D_7$ 数据线要向微电脑 2 传送数据。

(3) \overline{STROBE} 为低

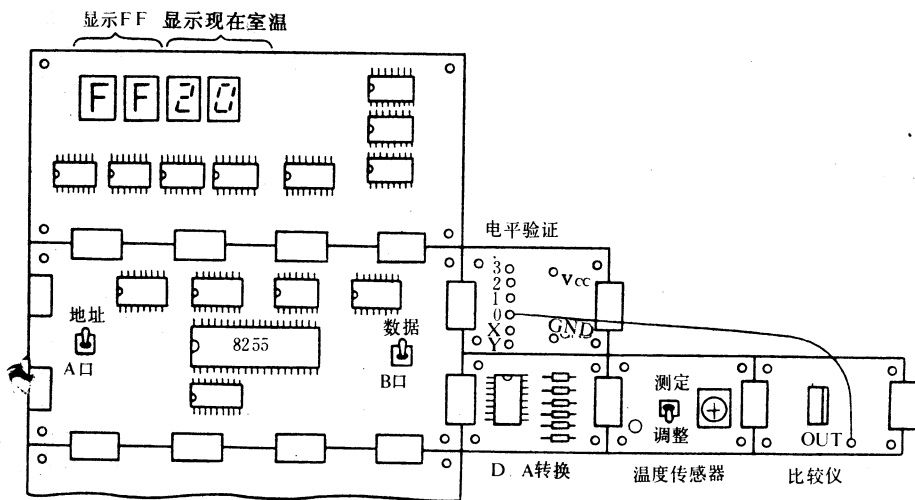


图 1

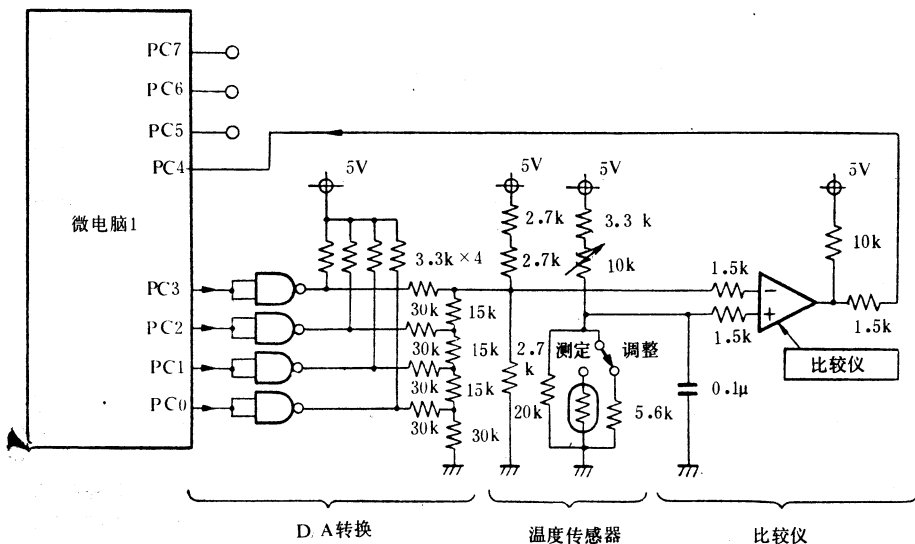


图 2

电平,微电脑 1 输出数据。

(4)STROBE 为低电平的同时,微电脑 2 向微电脑 1 发出 BUSY 为高电平信息,表示接收到微电脑 1 输出的数据,并正在内部进行处理。此时微电脑 1 不能向微电脑 2 输送下一个数据。如果微电脑 2 处理前个数据

完毕,则将 BUSY 变为低电平,通知微电脑 1 重复上述过程。

4. JIS 码

微电脑所认识的打印机能打印出的文字由 JIS 标准来决定。称为 JIS 码。如表 1 所示。

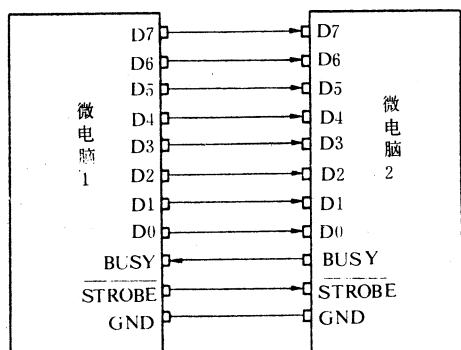


图 3

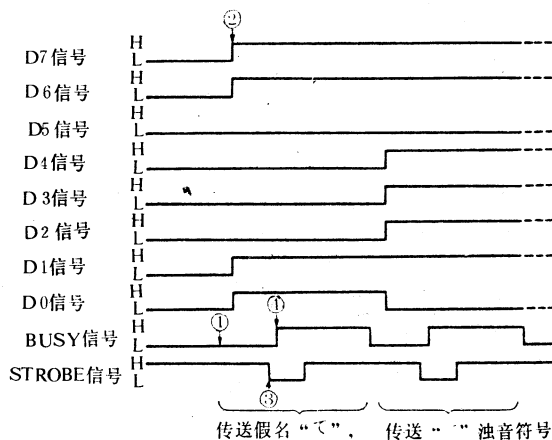


图 4

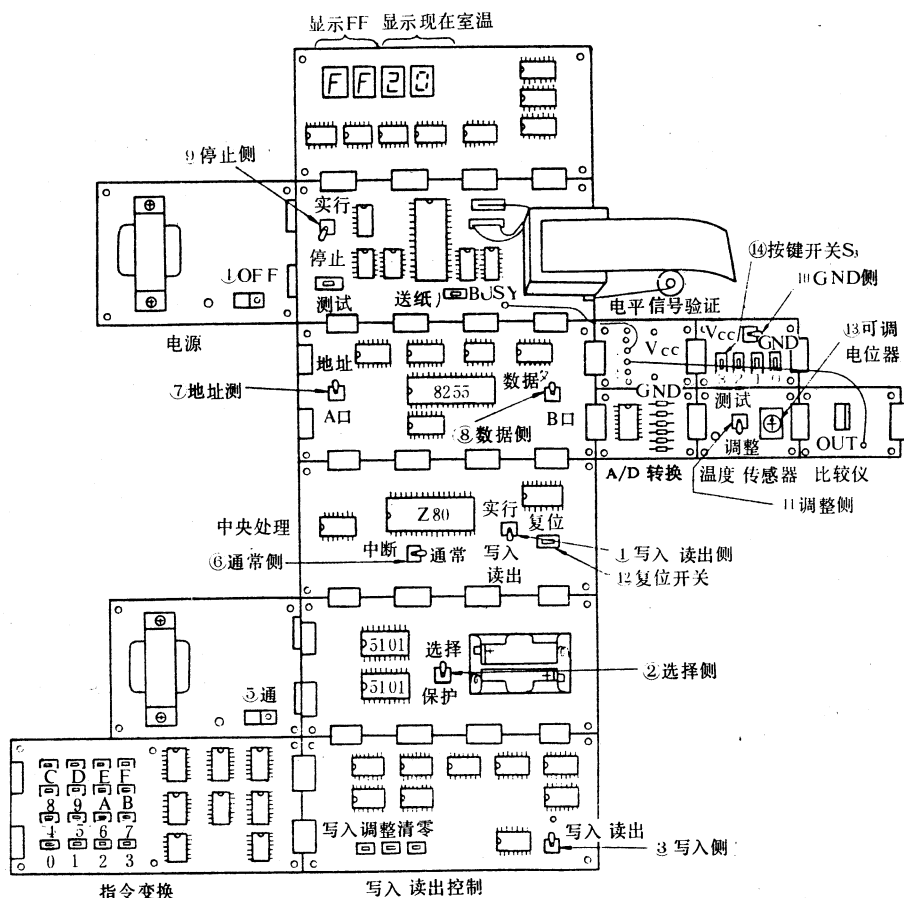


图 5

表 3

端口名	输入/输出	出脚名	任务
端口A	输出	出脚0	向打印机部件输出 STROBE 信号
端口B	输出	出脚0~7	向打印机部件输出 D ₀ ~D ₇ 信号
端口C	输出	出脚0~3	向 D/A 转换部件输出 D/A 转换数据信号
	输入	出脚4	输入比较部件的输出信号
	输入	出脚5	输入打印机部件的 BUSY 信号
	输入	出脚7	输入按键开关 S ₁ 的输入信号

表 2

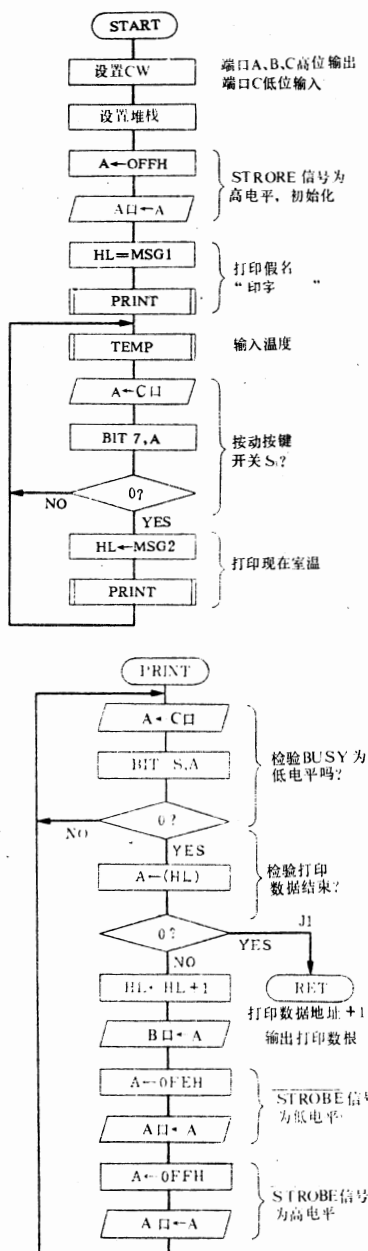
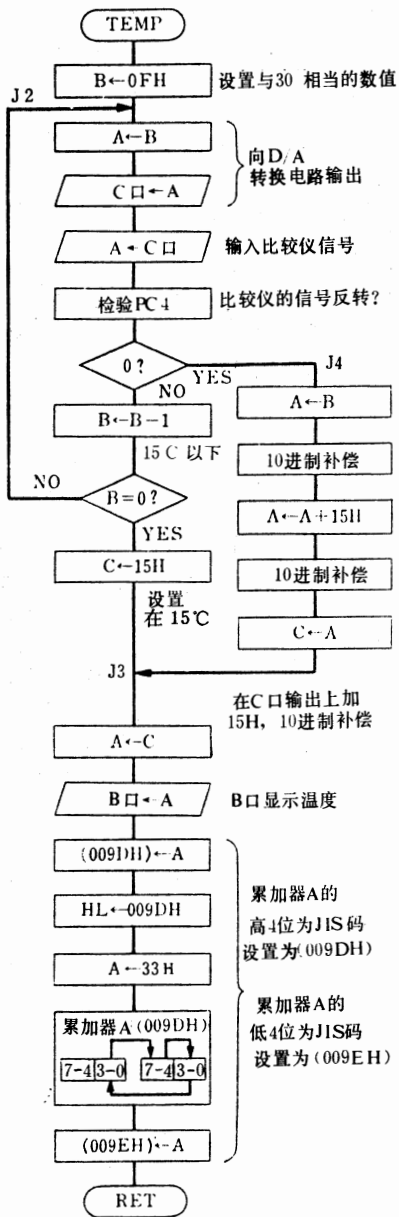


表 1

	0	1	2	3	4	5	6	7
0	0000	0001	0010	0011	0100	0101	0110	0111
1	0000	0001	0010	0011	0100	0101	0110	0111
2	0000	0001	0010	0011	0100	0101	0110	0111
3	0000	0001	0010	0011	0100	0101	0110	0111
4	0000	0001	0010	0011	0100	0101	0110	0111
5	0000	0001	0010	0011	0100	0101	0110	0111
6	0000	0001	0010	0011	0100	0101	0110	0111
7	0000	0001	0010	0011	0100	0101	0110	0111
8	0000	0001	0010	0011	0100	0101	0110	0111
9	0000	0001	0010	0011	0100	0101	0110	0111
A	0000	0001	0010	0011	0100	0101	0110	0111
B	0000	0001	0010	0011	0100	0101	0110	0111
C	0000	0001	0010	0011	0100	0101	0110	0111
D	0000	0001	0010	0011	0100	0101	0110	0111
E	0000	0001	0010	0011	0100	0101	0110	0111
F	0000	0001	0010	0011	0100	0101	0110	0111

标号	助记符	地址	机器语	注释
START	LD A,88H	00	3E 88	A、B、C口高位输出 C口低位输入
	OUT (03H),A	02	D3 03	
	LD SP,100H	04	31 00 01	设置SP堆栈
	LD A,0FFH	07	3E FF	A口初始化
	OUT (00H),A	09	D3 00	
	LD HL,MSG1	0B	21 80 00	设置标号MSG1的地址
	CALL PRINT	0E	CD 30 00	打印标题
MAIN	CALL TEMP	11	CD 50 00	输入温度
	IN A,(02H)	14	DB 02	C口输入
	BIT 7,A	16	CB 7F	若按键开关“通”
	JP NZ,MAIN	18	C2 11 00	
	LD HL,MSG2	1B	21 90 00	现在室温
	CALL PRINT	1E	CD 30 00	打印
	JF MAIN	21	C3 11 00	跳入标号MAIN
PRINT	IN A,(02H)	30	DB 02	检验BUSY是否 为低电平
	BIT 5,A	32	CB 6F	
	JP NZ,PRINT	34	C2 30 00	
	LD A,(HL)	37	7E	检验打印数据
	AND A	38	A7	结束?
	JP Z,J1	39	CA 4A 00	
	INC HL	3C	23	打印数据地址 + 1
	OUT (01H),A	3D	D3 01	输出打印数据
	LD A,0FFH	3F	3E FE	输出STROBE 信号
	OUT (00H),A	41	D3 00	
	LD A,0FFH	43	3E FF	
	OUT (00H),A	45	D3 00	
	JP PRINT	47	C3 30 00	跳入标号PRINT
J1	RET	4A	C9	返回主程序

表 2



标号	助记符	地址	机器语	注释
TEMP	LD B,0FH	50	06 0F	设置D A转换的初始值
J2	LD A,B	52	78	输出 D A 转换
	OUT (02H),A	53	D3 02	
	IN A,(02H)	55	D3 02	比较仪反转输出?
	BIT 4,A	57	CB 67	
	JP Z,J4	59	CA 70 00	若反转跳入标号J4
	DEC B	5C	05	数据 -1
	JP NZ,J2	5D	C2 52 00	若B≠0,跳入标号J2
	LD C,15H	60	0F 15	温度定为15℃
J3	LD A,C	62	79	B口输出温度
	OUT (01H),A	63	D3 01	温度数据变换 为JIS码
	LD (ONDO),A	65	32 9D 00	
	LD HL,ONDO	68	2F 9D 00	
	LD A,33H	6B	3F 33	
	RRD	6D	ED 67	
	LD (ONDO+1),A	6F	32 9F 00	
	RET	72	C9	返回主程序
J4	LD A,B	73	78	温度数据变为 10进制数
	ADD A,00H	74	C6 00	
	DAA	76	27	加上15℃变为 10进制数
	ADD A,15H	77	C6 15	
	DAA	79	27	
	LD C,A	7A	4F	跳入标记J3
	JP J3	7B	CA 62 00	
MSG 1	DB C3H	80	C3	チ
	DB DEH	81	DE	*
	DB BCH	82	BC	シ
	DB DEH	83	DE	*
	DB C0H	84	C0	タ
	DB D9H	85	D9	ル
	DB B5H	86	B5	オ
	DB D1H	87	DD	ン
	DB C4H	88	C4	ト
	DB DEH	89	DE	*
	DB B9H	8A	B9	ケ
	DB B2H	8B	B2	イ
	DB 01H	8C	0D	CR
	DB 00H	8D	00	エンドマーク
MSG 2	DB C0H	90	C0	タ
	DB C0H	91	C0	タ
	DB DEH	92	DE	*
	DB B2H	93	B2	イ
	DB C7H	94	C7	マ
	DB C9H	95	C9	ノ
	DB 20H	96	20	ブランク
	DB B5H	97	B5	オ
	DB DDH	98	DD	ン
	DB C4H	99	C4	ト
	DB DEH	9A	DE	*
	DB CAH	9B	CA	ハ
	DB 20H	9C	20	ブランク
ONDO	DB 20H	9E	20	ブランク
	DB 20H	9E	20	ブランク
	DB DEH	9F	DE	*
	DB 43H	A0	43	C
	DB 00H	A1	00	エンドマーク

例如,微电脑 1 向微电脑 2 输出的信息表示 26°C 时,按表 1 将它数码化,如下所示:

$$\begin{aligned} 2 &= 32H \\ 6 &= 36H \end{aligned}$$

(下转第 38 页)



单板机 RAM 分段与程序保护的实现

福州市福建师范大学物理系微机室(350007) 王新明

在单板机上开发应用程序时,由于程序局部存在错误或者操作失误,甚至环境干扰,都极易使程序运行混乱,造成内存(RAM)程序和数据遭破坏。特别对较长的程序调试是一种潜在威胁。开发简单实用、旨在保护随机存储器(RAM)中程序的硬软件,对解决单板机程序保护、加快程序开发应用是很有意义的。

一、TP801 RAM 分段保护原理

TP801 单板机上配置了 2K 的 RAM,空间分布为 2000H—2FFFH。除高端几个单元被系统使用外,都是用户区。我们可以把 RAM 分成两段。一段为保护区;另一段为非保护区。

两段以 $2 \times \times \times H$ 为分段地址。如图 1(保护区与非保护区位置可以交换)。①把待调试的程序和只供程序读的数据放在保护区内,其他数据区、堆栈区等设置在非保护区内。在程序运行时,把保护区 RAM 转变成 ROM 性质的存储器,只提供 CPU 读取,一旦发现 CPU 向保护区作写操作时,强迫程序中斷来达到保护目的。

②只把程序放在相应空间的保护区内,追踪 CPU 的取指地址,一旦发现 CPU 向非程序区取指就强迫中斷程序运行,来达到保护目的。

设保护区在 RAM 低段,选分界地址为 2520H,则

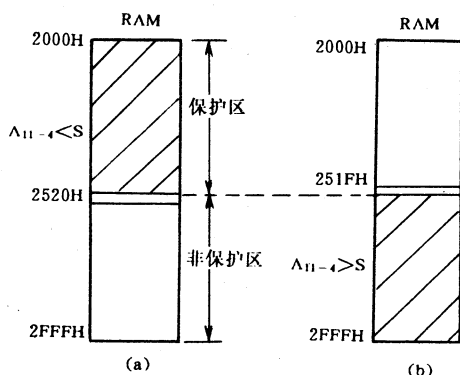


图 2 RAM 的两种分段
(a) $S=52H$; (b) $S=51H$

保护区地址 2000H—251FH; 非保护区 2520H—2FFFH。由于 RAM 地址高四位总是 0010,当分界地址选在最低四位为 0000 处,同时分段判断用小于、则只要取 2520H 中间的 52H 即地址 $A_{11}-A_4$ 就能在 2000H—2FFFH 把 RAM 分成两段如图 2(a)。其中 $A_{11}-A_4$ 表示地址的 $A_{11}-A_4$ 位,分界地址(2520H)的 $A_{11}-A_4$ 用 S 表示不妨称为段地址。当分段判断用大于,则分界地址应选在低四位为 1111 (0FH)处如图 2(b)。

二、硬件设计

电路如图 3。 U_1 、 U_4 为 74LS273 八位锁存器。1 脚是清零端,低电平有效;11 脚为锁存允许端,低电平有效。 U_1 用于锁存段地址 S,由 PS_5 (端口地址 94H)和 IOW 选通 U_1 的 11 脚,在 TP801 单板机上由 94H+PORT(键)将 S 送入 U_1 。S 作为数据,输出给 U_2 比较器的 B 输入端,即把 S 作为比较器的一个数值输入。 U_4 用于在发生 CPU 向保护区作写操作时,锁存被改写单元地址的 $A_{11}-A_4$ (电路在一次 CPU 向保护区写后,转入非屏蔽中斷服务)以提供调试人员在中斷错误程序后,快速找到被破坏的那个单元。读出 U_4 的内容,由 PS_5 和 IOR 选通 U_3 八位缓冲器,将 U_4 内容送数据总线 DB。 U_1 和 U_4 使用共同的端口地址,所以由 94H 口读出(PORT)并在显示器上以两位十六进制数显示的是 U_4 中的保护区被改写单元地址的 $A_{11}-A_4$ (显示成 $\times \times$)。这样我们只要在保护区 $2 \times \times 0H-2 \times \times FH$ 16 个单元查找,就能很快找到被改写的单元。 U_2 是用二片 7485 四数值比较器按图 4 级联成 8 位数值比较器。A 输入

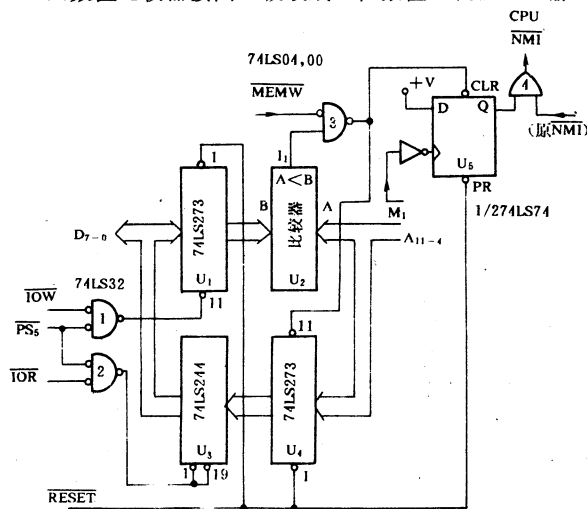


图 3 保护电路

端接地址线的 A_{11-4} 因此,电路工作时,比较器不断

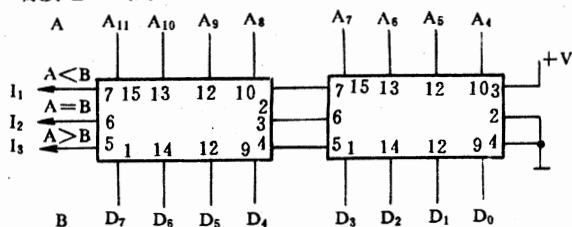


图4 8位数值比较器

比较段地址 $S(B$ 端) 和当前地址的 A_{11-4} 。 I_1, I_2 和 I_3 是比较器 $A < B, A = B$ 和 $A > B$ 的结果输出端, 它们的关系见表1。

电路用小于判断 (I_1) 分辨保护区, 所以分界地址选择第一种方法 (即选在最低四位为全 0000 处。如 2520H)。信号 (原 NMI) 是单板机原系统非屏蔽中断申

表1

A, B	I_1	I_2	I_3
$A < B$	1	0	0
$A = B$	0	1	0
$A > B$	0	0	1

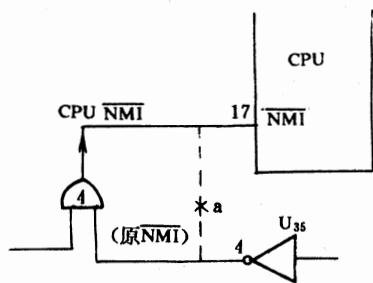


图5

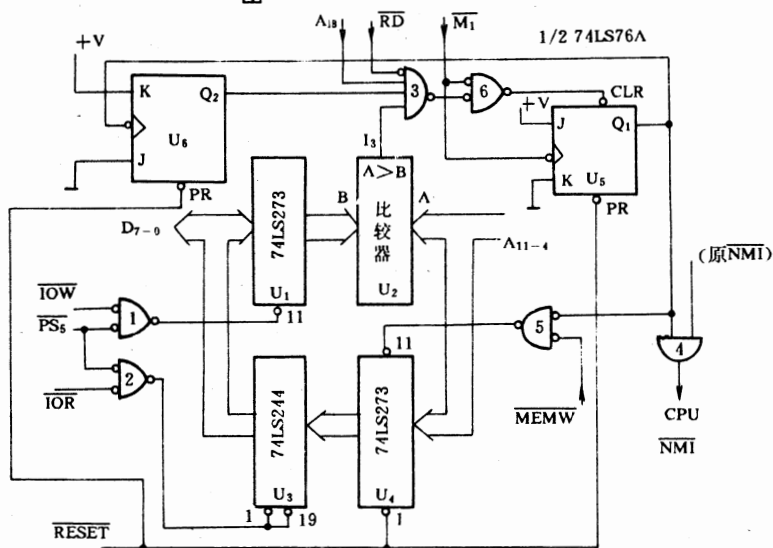


图7

请线。如图5虚线。原电路在 a 处切断, $U_{35} 4$ 脚接保护电路门4的输入端为 (原 NMI) 信号, 门4输出接到CPU的 NMI 输入端 (17脚) 如实线所示。

U_5 为中断请求触发器。由信号 I_1 和 \overline{MEMW} 清零 ($Q=0$) 产生中断请求信号 ($NMI=0$)。由 $\overline{M_1}$ 信号的下降沿置1来清除中断请求。 $I_1, \overline{MEMW}, \overline{CLR}$ 及 NMI 关系见表2。

表2:

\overline{MEMW}	I_1	\overline{CLR}	NMI	说明
0	0	1	1	非保护区写
0	1	0	0	保护区写, 中断请求
1	0	1	1	非保护区读或 I/O 操作
1	1	1	1	保护区读

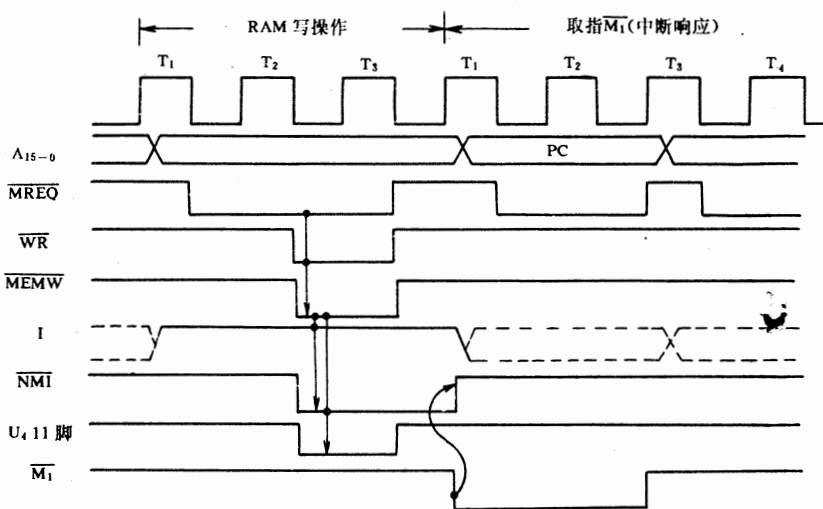


图6

当 $\overline{MEMW}=0$ 且 $I_1=1 \rightarrow \overline{CLR}=0 \rightarrow Q=0 \rightarrow NMI=0$, 导致电路向CPU发出中断请求, 这就是当CPU作RAM写 ($\overline{MEMW}=0$) 且RAM地址在保护区内 ($I_1=1$) 时, 电路清零中断请求触发器 U_5 , 发出中断请求。 Q 的低电平维持到取指周期到来, 由 $\overline{M_1}$ 下降沿自动清除中断请求状态。 $U_4 11$ 脚接 \overline{CLR} , 所以, 当CPU向保护区一个存储单元作写操作的同时, 该单元地址的 A_{11-4} 被锁入 U_4 。CPU作保护区写操作及中断响应周期中各有关信号时序见图6。

\overline{RESET} 可利用原机的复位信号, 电路自带复位电路使用更方便。

电路复位 ($\overline{RESET}=0$) 使 U_1, U_4 复位, U_5 置位。电路处于初始状态: U_1 内 $S=00$, 等效分界地址为 2000H, 保护区长度为 0, RAM全是保护区。在这种状态下, 可以

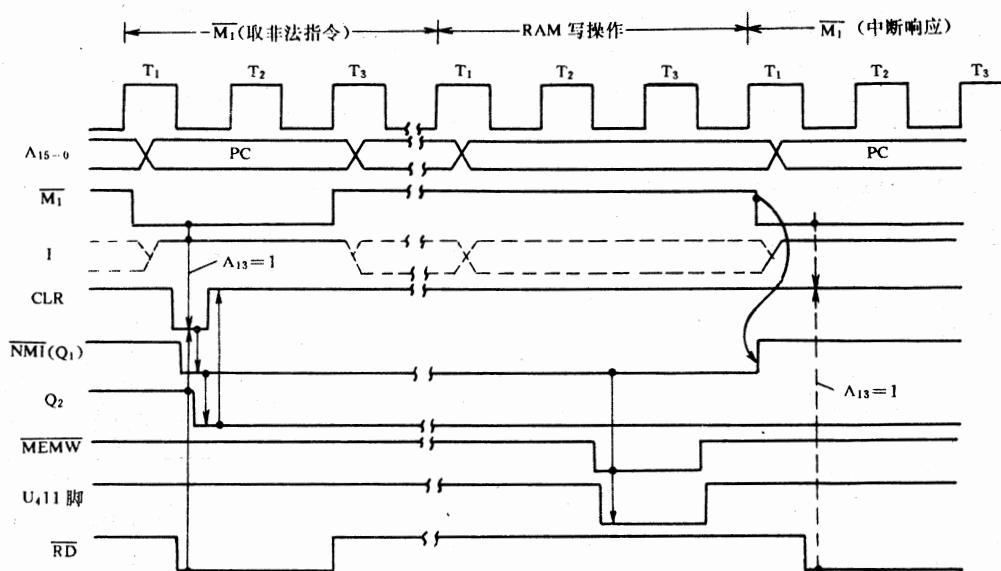


图 8

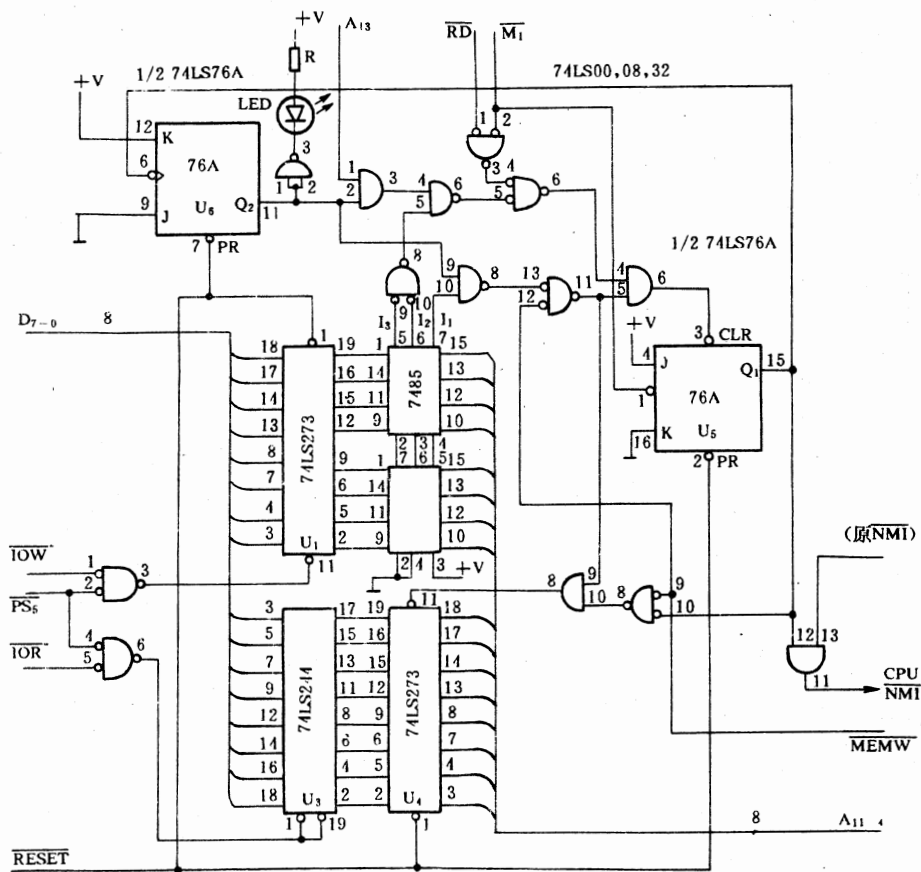


图 9

进行程序、数据输入及 RAM 内容修改等原机操作功能。电路在将段地址 S 置入 U_1 后的程序运行中对所设置的保护区起保护作用。但对非保护区无任何作用。

3. 取指检测保护

这种保护取保护区长度稍大于或等于程序长度,区内多余的空单元,填上 76H(暂停指令码),以防不必要错误产生。电路如图 7。分界判断用大于(I_3),这样分界地址选择第二种方法,即选在低四位为 1111 处(如 251FH,这时段地址 $S=51H$)。

$U_1 \sim U_5$ 的作用同前。 U_6 为中断屏蔽触发器,由 \overline{RESET} 置位呈允许中断状态($Q_2=1$)。 U_6 由 U_5 的 Q_1 下降沿触发电使 $Q_2=0$,这时,由 U_6 的作用,禁止电路产生新的中断请求信号。屏蔽状态维持到电路重新初始化($\overline{RESET}=0$)。

U_5 的 CLR 端状态可由下式表示:

$$CLR = \overline{M_1} + \overline{I} + \overline{Q_2} + \overline{A_{13}} + \overline{RD}$$

电路初始化后, $Q_2=1$ 。因此,必须在其余四个信号同时为: $A_{13}=1$ (地址线上的地址应为 RAM 地址 $2 \times \times \times H$); $I_3=1$ (地址的 $A_{11-4} > S$); $\overline{M_1}=0$, $\overline{RD}=0$ (在取指令周期内)时, $CLR=0 \rightarrow Q_1=0 \rightarrow \overline{NM_1}=0$,产生中断请求。同时 Q_1 下跳使 $Q_2=0$,从而禁止了电路可能的新的中断申请(图 8 中断响应时序中虚线所示)。中断请求信号维持到 $\overline{M_1}$ 到来时,由 $\overline{M_1}$ 下降沿使 U_5 的 $Q_1=1$,自动清除申请中断状态。门 5 的作用是在中断请求期间($Q_1=0$),执行非法指令若作 RAM 写($\overline{MEMW}=0$),则由门 5 选通 U_4 ,将被写的单元地址的 A_{11-4} 锁存下来, U_6 的禁止中断状态也保证了 U_4 锁存地址的唯一性。CPU 取非法指令及 RAM 写、中断响应的各信号时序如图 8。

电路在 CPU 向保护区以外的非程序区取指时起保护作用,执行非法指令至多破坏一、二个 RAM 单元; A_{13} 的加入也防止 CPU 取监控区(0000H—07FFH)指令时引起误动作。电路允许向保护区作写操作,这对一些变量在保护区内的程序是有利的。但电路对取指错误发生在保护区内(如转移指令转移到指令操作数上及之后发生在保护区内的错误运行)没有保护作用。

4. 复合保护

电路如图 9,是上述两电路的合成。它在两种情况之一出现时起保护作用:①在 CPU 取指周期内,指令地址在保护区外;②在执行周期内,CPU 向保护区作写操作时。不仅对保护区有保护作用,而且在多数出错情况下对非保护区也有保护作用。这样,在出错而发生中断后,不仅保存了断点各寄存器状态,还保留了 RAM 数据区、堆栈等的当前值,保证 RAM 程序、变量的安全,有利在发生中断后,供调试人员参考。中断保护发生后,单板机转入非屏蔽中断处理,且显示程序首地址(监控 TPBUG-A 版本)。

电路中, I_1, I_2 和 I_3 的组合构成保护区以小于判断,所以分界地址以第一种方式。 Q_2 处的发光二极管用来指示当前电路工作状态(保护或非保护态)。另外, U_6 的 PR 可由 \overline{RESET} 和另加的开关来共同置位 U_6 ,这样,可在不影响 U_1 的 S 情况下用开关单独使 U_6 从屏蔽态中恢复。

电路可装在一片插板上,利用原机 S-100 总线插座,接插方便,原机硬件几乎不变。方便实用,特别适合于实验开发及学生实验用机。

(上接第 34 页)

$^{\circ} = DFH$

$C = 43H$

这样将数字或文字变换为 8 位数码。按图 4 所示时序形式,微电脑之间传递信息。

5. 微电脑与打印机联接

微电脑 2 与打印机装在一块印刷线路板上。通过 8 线接插件与图 2 所示的数字温度计相连。如图 5 所示。

6. 软件

热敏电阻检测室温经过数据转换,由打印机打印出来。程序清单如表 2 所示。

执行该程序首先打印出“数字温度计”的日文片假

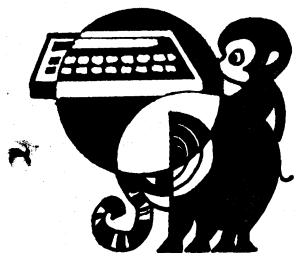
名“デジタルオンドケイ”(程序标记 MSG1 处)。按键开关 S_3 为“通”,打印出“现在室温 $\times \times ^{\circ}C$ ”的日文片假名“タダイマノオンドハ $^{\circ}C$ ”(程序标记 MSG2 处)。测试现在室温用 TEMP 子程序。现在室温数字放在程序标记 ONDO 处,即 9DH 和 9EH 地址中,执行 MSG2 打印指令,可自动调入室温数据。

输入输出口的设置如表 3 所示。

7. 软件改进

上述程序与数字时钟程序相结合,可以实现隔一定时间自动记录温度。

也可以考虑应用于实验室的计测仪器中,或者是自制的测试设备中。有兴趣的读者可进行尝试。



第三章 F BASIC 的画面控制语句

山东苍山县机械电子化学工业局(277700) 于 春

电脑游戏机

F BASIC 语言有着独特的背景画面设计和卡通画面控制功能。它独特的画面控制语句,是任何 BASIC 语言难以比拟的。使用该部分语句,可以巧妙地设计背景画面(BG GRAPHIC)、灵活地控制卡通图案(Sprite)的动作,从而设计出妙趣横生的游戏程序。

BG GRAPHIC 是英文 BACK GROUND GRAPHIC 的缩写,意为利用背景绘图。SPRITE 在英文中是幽灵,小精灵的意思,在此表示卡通图案。为叙述方便,以后称背景图形面为 BG 面,卡通图案面为 SP 面。

由于 F BASIC 语言画面控制语句的独有性,所以,对新老程序人员来说都较陌生,尤其对初学者,接受更感困难。因此,本文以较多的示例,循序渐进,分层介绍该部分语句的功能和使用。并对随机手册中的错误予以更正,遗漏予以补充,并增添了几条新语句。读者只要按介绍的顺序,逐句领会,勤加练习,定能熟练掌握。

一、画面控制语句

1. 设定光标语句(LOCATE)

LOCATE 简写 LOC.

LOCATE 语句的功能是设定光标的显示位置,其格式为

LOC. X,Y

式中:X—水平方向座标。取值 0~27

Y—垂直方向座标。取值 0~23

LOCATE 语句设定光标显示位置的座标为(X,Y)。

如:10 LOC. 0,10

RUN 光标在座标为(0,10)处闪烁。

注意——使用 LOCATE 语句设定闪烁光标时,X 取值恒为 0,Y 取值可以为 0~23。若 X 取其它值,计算机执行时,仍将它当做 0 处理。这是因为闪烁光标为输入语句的起点,而每行的输入起点必然在 X=0 的位置。所以 LOC. 0,15 与 LOC. 10,15 效果是一样的。

例 1.

```
10 F.I=0 TO 20
20 LOC. I,I : P. "#"
30 N.
```

例 2.

```
10 F. A=0 TO 5
20 X=0 : Y=0
30 X=X+1 : Y=Y+1+A
40 IF X>27 OR Y>23 T. 60
50 LOC. X,Y : P. "*" : G. 30
60 N.
```

2. 清屏语句(CLS)

CLS 语句的功能是清除屏幕的文字、画面等,使屏幕变为空白。如例一、例二中最后加一句

PAU. 100 : CLS

RUN 后,打印的图象显示一段时间,然后清屏,仅在屏幕左上角显示 OK 和光标。

3. 测试光标语句(POS 和 CSRLIN)

在程序的运行中,有时要了解程序的执行情况和光标在目前画面上所在位置的座标,使用测试光标语句可以达到这一目的。

POS(0)给出光标在目前画面上的水平座标。

CSRLIN 给出光标在目前画面上的垂直座标。

CSRLIN 简写为 CSR.

例 3.

```
10 F. X=0 TO 25
20 LOC. X,0
30 P. POS(0) : PAU. 10
40 N.
```

RUN 后,打印光标的水平座标。

例 4.

```
10 F. I=0 TO 20
20 LOC. 0,I
30 P. CSR. : PAU. 20
40 N.
```

RUN 后,打印光标的垂直座标。

例 5.

```
10 X=Y=0
20 X=X+1 : Y=Y+5
30 IF X=20 T. E.
40 IF Y>23 T. Y=0
50 LOC. X,Y
60 P. POS(0);", "; CSR. : PAU. 10 : CLS
70 G. 20
```

RUN 闪烁打印出光标的各组座标。

4. 字符—ASCII 码转换语句(ASC, CHR \$)

ASC 语句、CHR \$ 语句是一对相互逆操作语句。ASC 语句实现把字符转换为 ASCII 码。CHR \$ 语句(简写 CH.)实现把 ASCII 码转换为相应的字符。语句格式为:

ASC(字符串)

CH. (X) X 取值范围 0~255

例 6.

```
10 I. A $
20 P. A $ "="ASC(A $)
```


30 G. 10

RUN 后每输入一个字符,则显示该字符对应的 ASCII 码。

例 7.

10 I. A

20 P. A="CH. (A)

30 G. 10

RUN 后每输入一个数字(0~25)则显示该数字对应的字符或图案。

例 8.

10 F. I=0 TO 255

20 P. I="CH. (I)

30 N.

RUN 打印 0~255 所对应的字符和背景图案。

注一在随机手册中仅刊印了背景图形库的 104 种图案,但没有给出各图案所对应的 ASCII 码,这给调用背景图形带来了困难。下面给出所对应的 ASCII 码。读者可把数字标在背景图形库上以便使用。在背景图形库中

第一行的一~八列对应的 ASCII 码为 0~7

第二行的一~八列对应的 ASCII 码为 8~15

第三行的一~八列对应的 ASCII 码为 16~23

第四行的一~八列对应的 ASCII 码为 24~32

第五行的一~八列对应的 ASCII 码为 184~191

第六行的一~八列对应的 ASCII 码为 192~199

第七行的一~八列对应的 ASCII 码为 200~207

第八行的一~八列对应的 ASCII 码为 208~215

第九行的一~八列对应的 ASCII 码为 216~223

第十行的一~八列对应的 ASCII 码为 224~231

第十一行的一~八列对应的 ASCII 码为 232~239

第十二行的一~八列对应的 ASCII 码为 240~247

第十三行的一~八列对应的 ASCII 码为 248~255

说明:在第五行至十三行的 72 个图形可用 CHR \$ 语句直接调用,而第一行至第四行的 32 个图形则必须用以后将要介绍的 CGEN 及 SPRITE 语句才能调出来。

5. BG、SP 面符号配置语句(CGEN)

CGEN 语句的功能是决定背景(BG)面和卡通(SP)图案面上的符号配置。规定哪些内容显示于 BG 面,哪些内容显示于 SP 面。CGEN 语句的格式为

CGEN(n)

式中 n 的取值为 0~3,它的分配情况如下表

n	背景(BG)面	卡通(SP)面
0	显 BG	显 SP
1	显 BG	无 SP, 显符号、字母
2	无 BG	显 SP
3	无 BG	无 SP, 显符号、字母

由于 CGEN 语句较难懂,所以补充了上表,读者可配合随机手册中指令分配表,再结合后文例 11 的演

示,领会该语句的变化和作用。

二、卡通显示语句

该部分共有四个语句:DEF SPRITE、SPRITE、SPRITE ON、和 SPRITE OFF,主要用于卡通图案的显示或消除。随机手册中称这四条语句为立体分解画面用语句,我觉得称为卡通显示语句较贴切、也较易理解。

1. 定义 SP 语句(DEF SPRITE)

DEF SPRITE 简写 DE. SP.

该语句的功能是定义显示于 SP 面上的卡通图案。

在 F BASIC 系统中固化了十六类卡通图案 58 种姿态,每个姿态由一组 4 个数字进行定义,如玛丽哥哥跳为(12,13,14,15),丽莎小姐滑倒为(44,45,46,47)等。其语句格式为

DE. SP. n, (A,B,C,D,E)=字符表达式

式中<1>n 是 SP 的编号,0~7。F BASIC 允许在同一画面中显示八个卡通图案,每个卡通编一个号码,以便识别。

(2)<A>为配色号码,0~3,以后介绍

(3)为显示图案的大小。取值 0~1,当 B=0,图案为一个文字大小(8×8点),取值为 1 时,显示图案有四个文字大小(16×16点)。一般取值为 1。

(4)<C>表示显示的方式。即 SP 是显示于背景前,还是背景后。取值 0~1。C=0,显示于前;C=1,显示于后。

(5)<D>、<E>为显示图案反转指示、各取值为 0~1,为 0 时不翻转,为 1 时翻转。D 定义 X 轴方向,E 定义 Y 轴方向。

(6)字符表达式:一般由 CHR \$(X) 一组四个组成,也可由字符列或字符变量组成。

2. 显示 SP 语句和消除 SP 语句 (SPRITE ON, SPRITE OFF)。

显示 SP 语句 SPRITE ON 简写 SP. O. 功能是打开显示开关,显示 DEF SPRITE 语句定义的卡通图案。

消除 SP 语句 SPRITE OFF 简写 SP. OF. 功能是关闭显示开关,消除已显示的卡通图案。

在进行程序设计时,必须先使用显示语句 SPRITE ON,否则卡通图案不能显示。消除语句一般不用。

3. SP 显示位置语句 (SPRITE)

SPRITE 语句的功能是定义卡通图案显示或消失的位置座标。

SPRITE 简写 SP.

语句格式为

SP. n, X, Y

式中 n 是 SPRITE 编号 0~7, F BASIC 允许在同一水平方向上同时显示 4 个卡通图案,若超过 4 个,以后的 SP 不能显示;在同一画面上允许显示 8 个卡通图案,若超过 8 个,以后的 SP 也不能显示。X 是 SP 显示的水平座标,Y 是垂直座标,取值均为 0~255。实际上由于显示器画面的有效范围限制。一般 X 取值 0~240, Y 取值 0~220。

另外当 SPRITE 语句后省略 X,Y 时,其功能为消除 DEF SPRITE 的作用。

例9. 在(100,100)处显示玛丽哥哥跳,在(150,150)处显示丽莎小姐跳。

```
10 SP. O.  
20 DE. SP. 0, (0,1,0,0,0)=CH. (12)+CH. (13)  
+CH. (14)+CH. (15)  
30 DE. SP. 1, (0,1,0,0,0)=CH. (40)+CH. (41)  
+CH. (42)+CH. (43)  
40 SP. 0,100,100  
50 SP. 1,150,150
```

例10试编一程序,显示玛丽的七种动作。

```
10 SP. O. :Y=-1  
20 F. X=0 TO 27 ST. 4  
30 Y=Y+1  
40 DE. SP. Y, (0,1,0,0,0)=CH. (X)+CH. (X+1)  
+CH. (X+2)+CH. (X+3)  
50 SP. Y, Y*25+50, Y*25+50  
70 N.
```

RUN 显示玛丽的七种姿态。若欲显示其它卡通只须改50行中 CHR\$ 括号中的数字即可。也可以改30行的循环变量初、终值。

例11. 编一程序显示 CGEN 的四种状态

```
10 CLS:SP. O.  
20 F. I=191 TO 255  
30 P. CH. (1);:N.  
40 DE. SP. 0, (0,1,0,0,0)=CH. (0)+CH. (1)+  
CH. (2)+CH. (3)  
50 SP. 0,100,150  
60 F. A=0 TO 3:PAU. 300  
70 CGEN A.  
80 LOC. 0,23:P. "CGEN";A:N.
```

可反复 RUN, 仔细观察 CGEN 语句的变化和功能。

通过上例的演示,细心的读者可以发现,在 CGEN 等于1和3时,玛丽哥哥变为背景图形,它们正对应于背景图形库中第一行1~4列的四个图案。因此,可以编一程序调出图形库中前32个图形。

例12. 调出背景图形库前32个图形。每八个一组,依次显示。

```
10 CLS:SP. O. :CGEN 3  
20 F. A=0 TO 7  
30 DE. SP. A, (0,1,0,0,0)=CH. (A)  
40 SP. A,50,A*30:N.  
50 PAU. 100:SP. OF. :SP. O.  
60 F. I=8 TO 15  
70 DE. SP. I-8, (1,1,0,0,0)=CH. (I)  
80 SP. I-8,100,(I-8)*30:N.  
90 PAU. 100:CLS  
100 F. I=16 TO 23  
110 DE. SP. I-16, (2,1,0,0,0)=CH. (I)
```

```
120 SP. I-16,150,(I-16)*30:N.
```

```
130 PAU. 100:CLS
```

```
140 F. I=24 TO 31
```

```
150 DE. SP. I-24, (3,1,0,0,0)=CH. (I)
```

```
160 SP. I-24,200,(I-24)*30:N.
```

RUN 将依次显示32个背景图形,并每八个图形变换一种颜色。程序50行使用了 SPRITE OFF 语句,消除前一组图形,然后用 SPRITE ON 语句再打开显示开关。也可以使用 CLS 语句达到同一目的。如90行和130行。读者可以用 GOTO 语句反复演示。

上例程序重复部分较多,我们可以再加一重循环,以简化程序。双重循环程序如下:

```
10 CLS: SP. O. :CGEN 3:X=-1  
20 F. A=0 TO 31 ST. 8  
30 X=X+1  
40 F. I=0 TO 7  
50 DE. SP. I, (X,1,0,0,0)=CH. (A+I)  
60 SP. I,50+50*X,I*30:N.  
70 PAU. 100:CLS:N.
```

RUN 与上例程序效果相同。

以上程序若再加进 CGEN 语句使从0~3变化,形成三重循环,则可观察 CGEN 语句四种状态的变化,读者可自己练习。

三、BG、SP 画面结合语句(BGTOOL、VIEW)

1. 返回 BG 画面语句 BGTOOL

BGTOOL 简写 BG.

BGTOOL 语句可作为直接指令使用。其功能是在直接进入 BASIC 状态进行程序设计中,当需要进行背景画面设计时,返回 BG 画面。使用方法:不键行号,直接键入 BGTOOL,回车后,则进入 BG 画面。也可以将它编在程序语句中,在程序运行中返回 BG 画面。在按 BG GRAPHIC 方法设计背景图案完毕后,键入 ESC 和 STOP 可返回 BASIC 状态,继续程序设计或运行。

该语句随机手册中没有介绍,而介绍的从 BASIC 状态返回 BG 画面语句 SYSTEM 在直接进入 BASIC 状态中也无效。SYSTEM 语句只有在对话进入 BASIC 状态中才有效。由于对话进入 BASIC 可供用户使用的内存比直调 BASIC 要少一倍,一般使用直调 BASIC,所以 BGTOOL 语句比 SYSTEM 更有用。读者可在你的随机手册中添上,以利使用。

2. 调 BG 画面语句 VIEW

VIEW 简写 V.

语句功能是在 BASIC 程序执行中调入已设计好的背景画面。其格式为 VIEW。使用方法是在程序的适当地方添上该语句。当程序运行到 VIEW 语句,则调进 BG 画面。

例13. 画一方框,使丽莎的七种姿态在方框中依次显示。

```
10 BG. :SP. O.  
20 Y=-1
```

(下转第3页)



PC-1500计算机的维修

四川雅安地区气象局(625000) 何 静

日本夏普公司生产的 PC-1500袖珍计算机以它功能强,体积小,运算精度高独特优点,被广泛应用在我国气象水文、地质勘探、测量规化等部门的业务中。伴着它适应性强,普及面广,使用环境不限的长处,也随之带来了比较容易出故障的缺点。根据近几年的使用情况看,其故障大多出在外部设备上。CE-150四色绘图打印机的故障率为百分之四十;主机故障率为百分之二十;录音机的故障率为百分之二十;接口电路和其它故障率为百分之二十。因售后服务不太完善,即使一个细小的零件坏了,也不易配制。这给维修工作带来一些麻烦,也在一定程度上影响了 PC-1500计算机的使用效率。以下是笔者从事 PC-1500计算机维修的笔记整理。希望对读者在维修和使用 PC-1500计算机上有所帮助。

一、故障现象:开机就显示“CHECK:6”,加上充电器充电达5小时以上也是一样。做常规检查时,敲 TEST 后打印机不动作。

故障分析排除:机器显示“CHECK:6”,即是系统警告用户停机检查。其故障是由包括电源在内的多方面原因引起的。按照一般维修规则,首先应从电源电路上检查,因为主机有显示,打印机不初始化,故针对本故障现象应从打印电源入手检查。打印电源如图 A。整个电路由充电回路,可充电镉镍电池组和电力检测电路组成。充电回路由 R_1 、 D_1 组成; $D_3 \sim D_5$ 对直流稳压电源 EA-150 输出的 9V 电压降压后送给打印机的各部分; D_2 和 D_6 是将电池组电压分别送给主机和打印机;整个电路的后半部分是电力检测电路,其作用是检查电力是否足够,将检测结果送 CPU 告诉用户。为了检查打印机电源的充电回路是否有故障,将电池组的正极引线如图 B 处断开,并接上 EA-150 充电器向打印机电源充电,然后用电压表测量印刷电路板上连接电池的正负极的两个焊点(如图 C、D)上的电压。此时量得电压为 0.03(正常时应为 9V 左右)。用电阻沿充电回路测试发现二极管 D_1 内部开路。

经过更换 D_1 后显示屏上再没出现“CHECK:6”提示。

二、故障现象:主机显示“ERROR 80”,而且屏上字符暗淡、模糊。打印机不初始化。

故障分析及排除:主机显示 80 类错误,即是系统提示电力不足。但经过用电压表检查电池组电压是充足的。说明电源的前半部分是正常的,要从电力检测电路(如图 A 后半部分)去查找故障原因。电力检测电路由稳压管 HZ4CLL,电位器 W_1 ,三极管 2SC2021,电阻 R_2

组成。当在前的电池组输出电压大于 6V 时,稳压

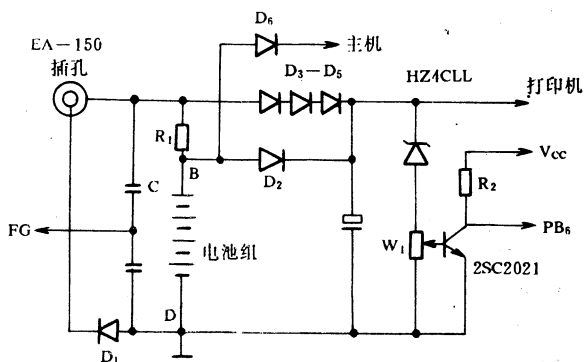


图1
CE-150四色绘图打印机电源电路

二极管反向击穿导通,电位器 W_1 上将有 2.3V 左右的压降,使三极管饱和导通。发射极与集电极间压降为 0V,由集电极向 I/O 端口的 PB_6 输送低电平,CPU 给显示屏发出电力充足的信号。反之,当电池电压低于 6V 时,三极管不能饱和导通。处于放大或截止状态,三极管对地电位为 1.4V 左右,输送给 I/O 端口的 PB_6 为高电平。CPU 发出电力不足的显示。本故障是一直提示电力不足,说明 PB_6 端口未得到低电平,而一直处于高电平状态。用电压表测量三极管集电极电压为 1.41V。经逐步检查发现是由于稳压管开路使三极管一直未导通。因为手头无现成的稳压管替换,故只好采取应急措施,将三极管集电极接地,人为地使 PB_6 得到了低电平。这只能是暂时的,这是将 PB_6 的电位强行拉为永久的低电平,是对机器的一种“欺骗”。其后果将是:当电池电力真正不足时,系统也不会提示警告用户。

机器经过以上处理后,基本上能工作。但显示屏上字符还是模糊不清。显示暗淡,一般是主机电源电压不够或虽电源够但某个元件坏使电源电压降低引起。显示屏电源 V_A 、 V_B 、 V_M 、 V_{DISP} 都是由主机电源供给的。主机电源厚模电路 MA1066 的外接电位器 W_2 ,主要控制调整这四个电压。

经过反复调整电位器 W_2 ,屏上字符清晰可见,本机完全恢复正常工作状态。

三、故障现象:加电后按进纸键有时进纸,有时不进纸。初始化常规检查时,敲 TEST 后打印机未印出标准的四色方框。有的画成长方形,有的画成一条粗线。

故障分析及排除:根据有时进纸,有时不进纸的故

障现象,初步分析故障属于 CE—150四色绘图打印机的机械故障,并且故障在打印机 Y 方向(即垂直方向)的动力传动上。拆开打印机的外壳,外观检查 Y 方向动力传动部分,看不出什么可疑的地方。但当从打印机的正面向后顺时针轻轻转动 Y 方向的过轮时,手感觉偶尔有卡住现象。为此,取下 Y 方向的过轮,轻轻转动

Y 齿轮,手感转动轻松;再用手转动 Y 电机齿轮时,手感异常,再进一步借助放大镜观察,结果发现 Y 电机齿轮上有一条与轴向平行的裂缝。

经过更换 Y 电机齿轮后,CE—150四色绘图找印机的故障消除。

导电橡胶按钮被磨损的修理方法

梁绍建

在现代电子产品中,很多方面都应用导电橡胶按钮技术。例如:电子计算机键盘、电子手表、电视遥控器,计算器键盘等。用导电橡胶做按钮有很多优点,它防水防尘性能极佳,可以在恶劣的环境下工作。用导电橡胶做的键盘决无大头针等微小物品掉进去造成事故的麻烦。

导电橡胶按钮一般是在橡胶按钮上和印刷电路板接触的键面上涂复一层导电胶而成。在使用中由于键面和印刷电路板之间的磨擦而产生磨擦,导电键面磨损后导致按钮灵敏度下降,最后完全失效。这种现象一般发生在常用的一个或几个按钮上。如果拆下按钮,用万用表的电阻档测一下那个失效的按钮键面,正常值应为几百欧姆至一千欧姆左右。如果阻值很大,就可以

判定为导电橡胶失效,就需要更换修理。但在一般情况下很难购到配件,我们可以自己用下面的办法修复。拆下键盘,用细砂纸把键面进行粗化处理,轻轻磨去表面的亮光即可,然后用酒精棉擦净。再找一点石墨粉,或用细砂纸磨下一点铅笔芯的粉末代用也可,最好用软些的3B 铅笔。把石墨粉均匀摊在一块蜡纸上,(蜡纸最好用不干胶标签的衬纸)石墨粉的厚度大约在0.5mm左右,用502胶水在处理后的键面上薄薄地涂上一层。立即把键面按在石墨粉上不动,十几秒后把蜡纸移走。由于蜡光纸表面非常光滑,可以在键面上获得一个平滑的导电表面。最后用酒精棉将键面上的浮粉擦去。这个键面的即告修复。用同样的方法修复其它按钮,这种方法简便易行,效果极佳。

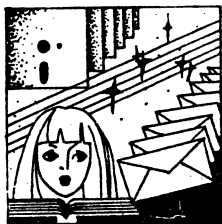
IBM—PC 机故障维修一例

武汉江岸车辆工厂(430012) 欧阳波

故障现象:平时硬盘起动正常。突然有一次,一开机,硬盘不断循环自检,不能系统自举。用软盘起动,但不能进入硬盘,如强行进入,则提示:INVALID DRIVE SPECIFICATION(非法驱动器标识符)如用非低级格式化来格式化硬盘则同样提示非法驱动器标识符。

用磁盘医生 NDD 软件对硬盘进行诊断,结果为:系统自举区已破坏,且提示是属硬件上的破坏,低级格

式化是唯一的办法,针对这种故障,把74LS244芯片用酒精擦洗并压紧,再用 NDD 诊断,则没有硬件上的故障,这样避免了破坏硬盘上已有的数据,同时排除了故障。这是由于芯片74LS244的管脚接触不良和表面灰尘影响所致。如有上述这种故障,擦洗和压紧74LS244芯片后,仍未排除故障,则换上一个好的74LS244芯片,故障即排除。这种故障多半是74LS244芯片所致。



DATA BASE IV 关系数据库

鸡西矿务局计算中心(158100) 张冰毅

新书与软件

DATA BASE IV (以下简称 DBIV) 关系数据库管理系统是1984年10月由 IKE 计算机有限公司推出的, 1985年初国内对它进行了汉化。DBIV 可以运行于 IBM-PC、PC/XT、PC/AT、IBM5550 和长城 0520 系列机等及其兼容机上, DBIV 对系统要求: 内存 256KB, 2.0 版以上 PC-DOS 和由 DOS 支持的一个 360KB 软盘或硬盘; 现有汉字 DOS 均支持 DB IV 软件系统。

时过七年, dBASE III 各版本和 FoxBASE 汉化版已广泛应用, DBASE IV 汉化版已推出的今天, 为什么又重提 DB IV 呢?

近两年中华学习机已逐步进入中、小学校和家庭, 与 IBM-PC 兼容的主要机种有 CEC-PC 和 BF PC-BOY, 主机内均配有 16×16 点阵汉字库, 基本配置均为内存 256KB, 一个 360KB 软盘, 提供与电视机的接口, 可将黑白或彩色电视机作为监视器用。在这种基本配置下, 无论 dBASE III, 还是 FoxBASE 和 DBASE IV 均无法运行, 而 DB IV 却可以, 并能够开发一些小型数据库信息管理系统, 这对只具有基本配置的用户不能不说是一件好事。

DB IV 最多可同时处理 10 个文件, 每个文件最多可容许记录数达 65535 条, 每条记录最多可容纳 4000 个字节, 每条记录最多可设 64 个属性; 观察(索引)命令中关键字最大长度为 80 个字符。数据有效位数对标准 BCD 码达 14 位, 对专用版本使用双精度可达 16 位, 而通常算术运算精度也可达到 6 位。

DB IV 可用两种操作方式工作, 一是直接执行 DB IV 命令的直接方式, 二是使用 DB IV 命令集的程序工作方式。DB IV 设有五种类型文件, 命令文件、观察文件、报告文件与文本文件, DB IV 对其记录采取分级处理方式。

DB IV 设置 44 种命令、四类共 35 种函数; 尤其是该系统设置了一种编辑器 (DB Editor), 用它可方便地建立新的文本文件, 编辑一个已有的文本文件, 或是开发应用程序等较强的编辑功能。此外, 它还设有一种可方便地建立输入/输出屏幕的 DB 屏软件。

使用过 PC 机的人基本上都熟悉 PC-BASIC 语言

和 dBASE III 关系数据库; 而 DB IV 有约四分之一的命令与 BASIC 命令名字相同, 功能接近, 有约二分之一的命令与 dBASE III 命令名字相同、功能接近。例如: 显示磁盘文件目录用 FILES 命令, 打开一个数据文件用 OPEN 命令, 显示与数据文件有关的信息用 DISPLAY 命令, 对数据文件中数字属性求和用 SUM 命令。在 DB IV 中, 与 BASIC 语言相同的是内存变量的数目仅受硬件设备的限制; 字符串变量要在变量名后加上“\$”号, 数字值变量则不加, 在整型数字值变量后加上“%”号; STR \$ 函数是将数字值变为字符表达式, 而 VAL 函数则是将字符表达式变为数字值。DBIV 可以象 BASIC 语言那样用 GOSUB-RETURN 命令调用子程序, 这对提高编程速度和质量是很有用的。若内存够用, DB IV 可用 // 命令执行 DOS 命令及语言命令 (如 BASIC、Multiplan 等), 这对编制较复杂的应用软件提供了一个非常方便的条件。

DB IV 本身提供了把数据文件转换成文本文件的命令, 可以用高级语言对文本文件进行处理; 另外经过对 DB IV 的关系数据文件的结构和记录进行研究, 找出文件结构和记录之间的数学关系式, 给出了一种用高级语言直接处理 DB IV 关系数据文件的方法, 这就提供了 DB IV 与高级语言的直接接口, 采用这种直接接口方式编制应用软件, 既可以互相取长补短, 又不多占用外存空间, 具有较大的实用价值。

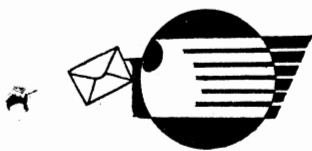
DB IV 主系统文件 DB4.EXE 只有 113KB, 且无覆盖文件, 这样在一张 360KB 软盘上给用户至少提供 200KB 外存空间; 若单独使用一张数据程序盘, 可先把 DB IV 主系统文件 DB4.EXE 调入内存后再抽出系统盘, 而后把数据程序盘插入 A 驱动器, 这样整个 360KB 软盘外存空间全部留给用户了。采用这种方法对建立小型数据库信息管理系统用一张软盘就够用了, 这就为 CEC-PC 等机型用户使用关系数据库提供了方便条件。

总之, DB IV 功能较强, 使用方便, 易学易懂, 对在一定范围内推广使用具有较大的意义。

(上接第 47 页)

普通显示器的扫描频率是固定的, 同一种显示器只能与一种对应的显示卡相匹配。如果要更换显示卡, 则必须同时更换显示器。

多同步显示器的扫描频率是可变的, 既可以配置 EGA 显示卡, 又可以配置 VGA 显示卡, 当需要提高显示分辨率时, 只要更换显示卡, 不需要更换显示器。



普及型 PC 个人用户软件交流联谊活动

问题解答(二)

读者联谊

北京中国农科院计算中心(100081) 王路敬

2. PC 机的系统板包括哪些主要功能部分?

PC 机的主机采用大板结构。系统板水平地固定于机箱底部。系统板的主要部分分成五个主要功能区。它们是微机处理器子系统和只读存储器(ROM)子系统,读写(R/W)存储器子系统,各种 I/O 适配器和 I/O 通道的支持部件。

系统板的核心是 Intel 8088 微处理器。它是一种 16 位微处理器,它的内部结构是 16 位的,而对外的数据总线是 8 位。它与 Intel 8086 在软件上是完全兼容的,其指令系统和汇编语言是相同的。因此,支持包括乘和除的 16 位操作。它的内部有 8 个 16 位的通用寄存器,可以存放操作数。可以实现寄存器间接寻址、基址寻址、变址寻址、以及基址加变址等多种寻址方式,使指令更加灵活,能适应简单变量、下标变量、矩阵等方面的运算。可实现 16 位算术运算和逻辑运算,实现 16 位数的移位和循环,且能指定任意的移位次数,可实现多种 16 位数的串操作。

在 Intel 8088 内部有 9 个标志位,可反映 CPU 操作的状态,实现各种条件转移和循环,重复控制。可实现 16 位数(或 8 位数)的输入输出,采用间接寻址方式,I/O 端口地址可以扩展到 64K 个。

Intel 8088 具有 20 条地址引线,可直接寻址 1M 字节内存空间。8088 可实现 256 个矢量中断,它具有软件中断,非屏蔽中断请求(NMI),屏蔽中断请求(INTR)和追踪等中断方式。利用软件中断可以很方便地调用操作系统中的大量子程序,大大简化了程序的编制。若把 8088 接成最大组态,可以很方便扩充浮点运算协处理器 Intel 8087。加上了 8087 可以使浮点运算的速度大大提高。

Intel 8088 可寻址的内存地址范围 0000—FFFFH。给定一个 20 位的地址,就可以从 1M 字节中取出所需要的指令或操作数。在 8088 系统中,存储器的访问,在不改变段寄存器值的情况下,寻址的最大范围是 64KB。所以,若有一个任务,它的程序长度,堆栈长度,以及数据区长度都不超过 64KB,则可在程序开始时,分别给 DS、SS、ES 置值,然后在程序开始就可以不再考虑这些段寄存器,程序就可在各自的区域中正常地进行工作。若某一任务所需的总的存储长度包括程序长度,堆栈长度和数据长度等不超过 64KB,则可在程序开始时使 CS、SS、DS 相等,程序也能正常地工作。如果对于一个程序中要用的数据区超过 64KB,或要求从两个或多个不同区域中去存取操作数,也是十分方便的,只要在取操作数之前,用指令给数据寄存器重新赋值就可以了。

在系统主板上可以安装 8K—48K 的 ROM 或 EPROM。它是 PC 机系统中不可缺少的一部分。由于只读存储器的芯片具有永久存储信息和只准读出不准写入的功能,因此,常把 PC 机系统的磁盘操作系统的引导程序、系统自检测试程序、I/O 驱动程序、BASIC 解释程序等固化在 ROM 中。

PC 机的 I/O 通道是 8088 CPU 总线扩展,它对信号进行隔离、增强驱动能力并由附加中断和直接存储器存取(DMA)来提高功能。I/O 通道一般包括:一个 8 位双向数据总线、20 根地址线、6 级中断、存储器和 I/O 读写的控制线、时钟和计时信号线、DMA 控制器,其中三个通道用于 I/O 设备与存储器之间的高速数据传输,一个通道用于对动态存储器进行刷新。

为了在硬件上对 IBM-PC 系统进行扩展,在系统板上安放了几个 I/O 插槽。对于 PC 机作为基本系统来说,一个槽要用来插 5.25 英寸软盘驱动器适配器,用以带 1—2 个软盘驱动器。一个槽用来插 IBM 单色显示器和一个并行打印机。也可以用两个插槽,一个用来插彩色并行打印机适配器,以带彩色显示器;另一个用来插并行打印机适配器,以带并行打印机。剩下的插槽可以用来扩展系统的内存 RAM,可以用来扩展 I/O 接口,并行或串行接口,可以作网络接口板,以形成局部网络,也可以用来插 A/D 或 D/A 转换板等等。

3. PC 机主机板上开关如何设置?

各类 PC 主机板上一般有两个开关 S1 和 S2(PC/XT PC/AT 无 S2)。这两个开关的设置用来确定:PC 机有无 8087 协处理器,内存存储器的总容量,显示器类型以及软盘驱动器的数量等。随着微机技术的发展,有的 PC 机已改为一个开关,或用软件系统配置方法。开关设置正确,或系统配置参数正确,主机能正常工作,否则自检给出错误信息。

IBM-PC 主机板上 S1 和 S2 开关,对 PC 机系统进行配置,故称系统配置开关。开关 S1 设置及其对应的功能如下:

位 置	功 能
1—7—8 ON ON ON OFF ON ON OFF OFF ON	决定所装 5.25 寸软盘驱动器个数 无驱动器 一个驱动器 两个驱动器
2 OFF ON	为 8087 协处理器留,不用协处理器置 ON 带协处理器 不带协处理器

3—4	设置系统板上的 RAM 存储容量
ON ON	16KB
OFF ON	32KB
ON OFF	48KB
OFF OFF	64KB
5—6	显示器类型
ON ON	无
OFF ON	40×25彩色
ON OFF	80×25彩色
OFF OFF	单色显示器

开关 S2 设置与功能如下:

位 置	功 能
1—2—3—4	决定存储器选件的容量
OFF ON ON ON	选96KB
ON OFF ON ON	选128KB
OFF OFF ON ON	选160KB
ON ON OFF ON	选192KB
OFF ON OFF ON	选224KB
ON OFF OFF ON	选256KB
OFF OFF OFF ON	选288KB
ON ON ON OFF	选320KB
OFF ON ON OFF	选352KB
ON OFF ON OFF	选384KB
OFF OFF ON OFF	选416KB
ON ON OFF OFF	选448KB
OFF ON OFF OFF	选480KB
ON OFF OFF OFF	选512KB
OFF OFF OFF OFF	选544KB
5—6—7—8	均置于 OFF 状态

PC 机用户不少是 IBM-PC/XT 机,其系统板上 S1 为系统配置开关,该开关的位置与其对应的功能如下:

位 置	功 能
1	常规操作为 OFF 状态,ON 时通道循环返回
2	使用协处理器时置成 OFF,否则置成 ON
3—4	系统板存储容量
OFF ON	128KB
ON ON	192KB
OFF OFF	256KB
5—6	正在使用的显示器适配器类型
ON ON	无
OFF ON	40×25彩色/图形
ON OFF	80×25彩色/图形
OFF OFF	单色显示器
7—8	5.25寸软盘驱动器个数
ON ON	带一个软盘驱动器
OFF ON	带二个软盘驱动器

4. PC 机系统主板的 I/O 插槽是如何工作的?

PC 机系统主板上装有的 I/O 插槽,有的系统板上装有5个,有的系统板上装有8个。所剩余的空插槽它们是为 PC 机配加选件方便而设置的,每一个 I/O 插槽有62根接线,这62根线可以把各种信号提供加到扩充设备上。62根线并列操作,因此任何扩充板都可以插入到 I/O 插槽。所有扩充板均采用一种公用的62根线连接结构,因此,PC 机的功能扩充是比较方便的。

I/O 插槽62根线的具体用法分成四类:

(1)8条线用来给扩充板提供各种不同电压的电源;

(2)8条线用来传送数据总线的8位数据。所有数据,其中包括输入存储器和输入/输出设备的数据都要经过这8条线。

(3)20条用于寻址。当数据要输入或输出到存储器或输入/输出设备时,需要指定一个地址,以标明该存储器地址单元或设备号。用于存储时,所有20根都用来指明1024K 个存储器中是哪一个单元正在被寻址。当用于输入/输出设备时,用9根地址线对512种不同设备进行寻址。

(4)26根线用来传送控制信号。例如读存储器命令,写存储器命令,输入/输出设备的读命令和写命令。

连接到 I/O 插槽的每一件设备都在不断地注视着输入/输出通道上的信号。例如,要发出一个输入/输出命令,它将通过在输入/输出该命令线上出现一个信号表示出来。而当该命令发出以后,所有输入/输出设备都注意地址线,而所有存储器电路都不管这些,如果发出一个存储器的写命令,就会产生与上述情况相反的情况,所有输入/输出设备将不管地址线,而所有的存储器电路都注视着那个给出的地址线。由于输入/输出的命令是请示性的,所以每个输入/输出设备都要注视地址线。一旦某个设备的地址出现在它的面前,它就立即采取行动,否则,它就等候下一次出现它的地址线。

5. PC 机的端口是干什么用的?

PC 机的端口是指微处理器8088用来简化和统一与外界进行联系的一种机构,它是微处理器与存储器以外的设备传送数据的唯一通路。

PC 机的微处理器与外围设备取得联系的设备例如键盘、软盘驱动器、硬盘驱动器等,都配备一个端口供它使用,一个端口就是一条假想的数据传送通路,给它分配一个端口号,它可以根据处理器的命令接受或发送数据。当8088需要把数据发到某个端口时,它利用 OUT 指令来指定端口号,以及要被发送的数据,该数据通常为一两个字节。当从端口接收数据时,它利用 IN 指令来指定端口号,以及要接收的数据。8088可以同任何一个端口联系,而不管它是否在工作。

PC 机微处理器8088端口是用16位二进制数来规定的,因此可最多达64K 个端口可供使用,而在实际应用上只分配了很少几个端口,因此,有大量的端口可供扩充。

但是要注意 PC 机实际上是以端口作为地址单

元,来区分不同的外设。因为一个外围设备不仅有数据寄存器,还有状态寄存器和控制命令寄存器,它们各需要一个端口才能区分,故一个外围设备往往需要数个端口地址。

问题解答

显示器·显示卡·显示系统

机电部第15研究所(100083) 孙梅英

1. 显示器与显示卡的关系?显示卡在实际显示系统中起什么作用?

微型计算机的显示系统由显示器和显示卡组成。

显示器是一种显示设备。它的功能是接收视频信号,在屏幕上显示字符、汉字、图形和图象。

显示卡又称为显示控制接口板、显示适配器。它是主机与显示器之间的接口。它直接插在主机的总线插槽内。它的作用是:接收 CPU 和主存储器发送的信息,输出视频信号和同步信号到显示器。

由此可见,显示器与显示卡的关系是非常密切的,是微型计算机显示系统中不可分割的两个组成部分。

2. 单色显示器与彩色显示器的区别?

阴极射线管显示器简称为 CRT 显示器,有时直接称为 CRT。它分单色和彩色两种。

阴极射线管(Cathode Ray Tube 简称 CRT)由电子枪、偏转装置和荧光屏组成。电子枪是阴极射线管的主要组成部分,包括灯丝、阴极、栅极、加速阳极和聚焦极。

单色显示器的工作原理:电子枪发射的电子束(阴极射线)通过聚焦和偏转系统撞击荧光屏,所发射的电子数取决于施加在电子枪上的电压。在控制电压的作用下,聚焦和偏转系统产生的电磁场将电子束聚焦于荧光屏上的某一点。当电子束轰击荧光粉涂层时,受轰击的部位便发出光来。光点的亮度取决于电子束中的电子数目。若我们控制电子束使其射向荧光屏上不同的点,便可显示出图形。

彩色显示器的工作原理与单色显示器基本相同。但是彩色显示器要有三个电子枪,分别对应红、绿、蓝三基色的信号强弱。在荧光屏内壁涂有彩色荧光粉,按三基色迭加原理形成彩色图象。

3. 显示器与显示卡怎样有机组合才是最合理的? CGA、EGA、VGA 之间是否有联系?同一个显示器中能否配备多种显示卡?

显示器是显示设备;显示卡是显示控制接口,两者组合在一起则成为显示系统。在配置显示系统时,要根据显示的技术要求(颜色、分辨率、显示器的扫描频率……等),选择适当的显示器和显示卡,组合成合理的显示系统,这样才能获得最佳的显示效果。

这里介绍几种显示卡的显示标准和特点:

1. MDA

MDA 是单色字符显示系统的显示控制接口板。它的特点是字符显示质量高,采用 9×14 点阵的字符窗口(指每个字符在屏幕上所占的点数,它包括字符显示点阵和字符间隔),满屏幕可以显示 80×25 行字符,对应的分辨率为 720×350 个像素。MDA 不能兼容图形方式。

2. CGA

CGA 是彩色图形/字符显示系统的显示控制接口板。它的特点是可以兼容字符与图形两种显示方式。在字符方式下的字符窗口为 8×8 点阵,因而字符显示质量不如 MDA,但是字符和背景可以选择颜色。在图形方式下,可以显示分辨率为 640×200 ,2种颜色的彩色图形或分辨率为 320×200 ,4种颜色的彩色图形。

3. EGA

EGA 集中了 MDA 和 CGA 两种显示标准的优点,功能进一步增强。字符显示窗口为 8×14 点阵,字符显示的质量优于 CGA 而接近于 MDA。图形方式的分辨率为 640×350 ,16种颜色,显示彩色图形的性能明显优于 CGA,而且兼容原 CGA 和 MDA 的各种显示方式。改进型 EGA(EGA+)或超级 EGA(Super EGA)的图形分辨率可以达到 640×480 或者 800×600 个像素。

4. VGA

VGA 是 IBM PS/2 系统的显示标准,在字符方式下的字符窗口为 9×16 点阵。在图形方式下,可以显示分辨率为 640×480 ,16种颜色的彩色图形或分辨率为 320×200 ,256种颜色的彩色图形。按 VGA 标准设计的显示控制接口板已经用于 IBM PC/AT 和 386 超级微型计算机系统。改进型 VGA(VGA+)或超级 VGA(Super VGA)显示控制接口板的图形分辨率可以达到 800×600 , 960×720 和 1024×768 个像素。

决定显示系统性能的另一个因素是显示器。各种不同型号的显示器,主要区别是扫描频率不同。显示卡的分辨率越高,输出视频信号的频率也越高,与它配套使用的显示器,扫描频率也应该越高。

对于固定频率的显示器来说,扫描频率是固定不变的。显示器的扫描频率必须与显示卡输出的视频信号频率相同,这样,才能得到显示系统设计时所要求的显示效果。

下面列出几种常用的显示卡所要求的显示器扫描频率。

显示卡	MDA	CGA	EGA	VGA
显示器水平扫描频率(KHZ)	18.432	15.75	21.85	31.75

随着显示技术的发展,1985年研制出一种新的显示器,它的扫描频率是可变的。它利用自动跟踪技术,使显示器自动与显示卡输出的视频信号相同步,同步的频率范围在 $15\text{KHz} \sim 35\text{KHz}$ 之间。这种显示器称为多同步显示器。

(下转第45页)

中软总公司(电脑大厦)

向您提供下列库存积压产品:

序号	名 称	单位	销售价
1	XIDEX 5" 软盘	片	2.00
2	5" 单/双燕牌	片	2.00
3	5" 单/双3M	片	2.00
4	5" GW	片	2.00
5	JANUS 5" 加密盘	片	5.00
6	8" 单/双	片	7.00
7	8" 单/双3M	片	7.00
8	8" 双/双3M	片	13.00
9	8" 磁盘盒	个	10.00
10	8" 空盘盒50466	个	10.00
11	5" 清洗盘 JANUS	盘	15.00
12	5" 清洗盘3M	盘	20.00
13	8" 清洗盘	盘	10.00
14	框带:3070	只	20.00
15	框带:9400	只	20.00
16	电脑电话	部	100.00
17	起拔器	个	2.00
18	涂抹器	个	150.00
19	鼠标控制器	个	400.00
20	游戏控制板	块	100.00
21	LQ 彩打接口板	个	200.00
22	扩展板5550	块	300.00

序号	名 称	单位	销售价
23	ROM 板 16 * 16	块	300.00
24	内存扩充板 17-1	块	350.00
25	时钟板	块	50.00
26	串行口板 RS-232	块	100.00
27	多 功 能 板 - 17 0520CE	块	350.00
28	HED 汉卡 AT	块	700.00
29	学习机打印卡	块	60.00
30	五笔字型卡	块	180.00
31	显示卡	块	180.00
32	数 字 化 仪 WT - 4000	台	3500.00
33	绘图仪 SPL-400	台	3500.00
34	绘图仪 LP-3700	台	20000.00
35	绘图仪 MP-1000	台	1500.00
36	打印机9250A	台	3500.00
37	TX-104清洗剂	瓶	20.00
38	TX-248清洗剂	套	20.00
39	TX-300清洗剂	套	20.00
40	TX-308/309A 清洗 剂	瓶	20.00
41	TX-800清洗剂	包	20.00
42	扁平电缆压接工具 MT-PSSEOIB	台	500.00
43	扁平电缆压接工具 MT-PS-DIP	台	500.00
44	74、75等系列各种元 器件		

我公司一次性的处理库存积压产品共有3000多种,包括各种 PC 机、286、386各档次不同机型,价格特优,欢迎选购。

地址:北京海淀区学院南路55号电脑大厦,邮编:100081,乘16路汽车电脑大厦下车。

联系人:温友良(经营计划处副处长) 开户行:中国工商银行海淀分理处,帐号:461187-79

电话:831.7722转1302