



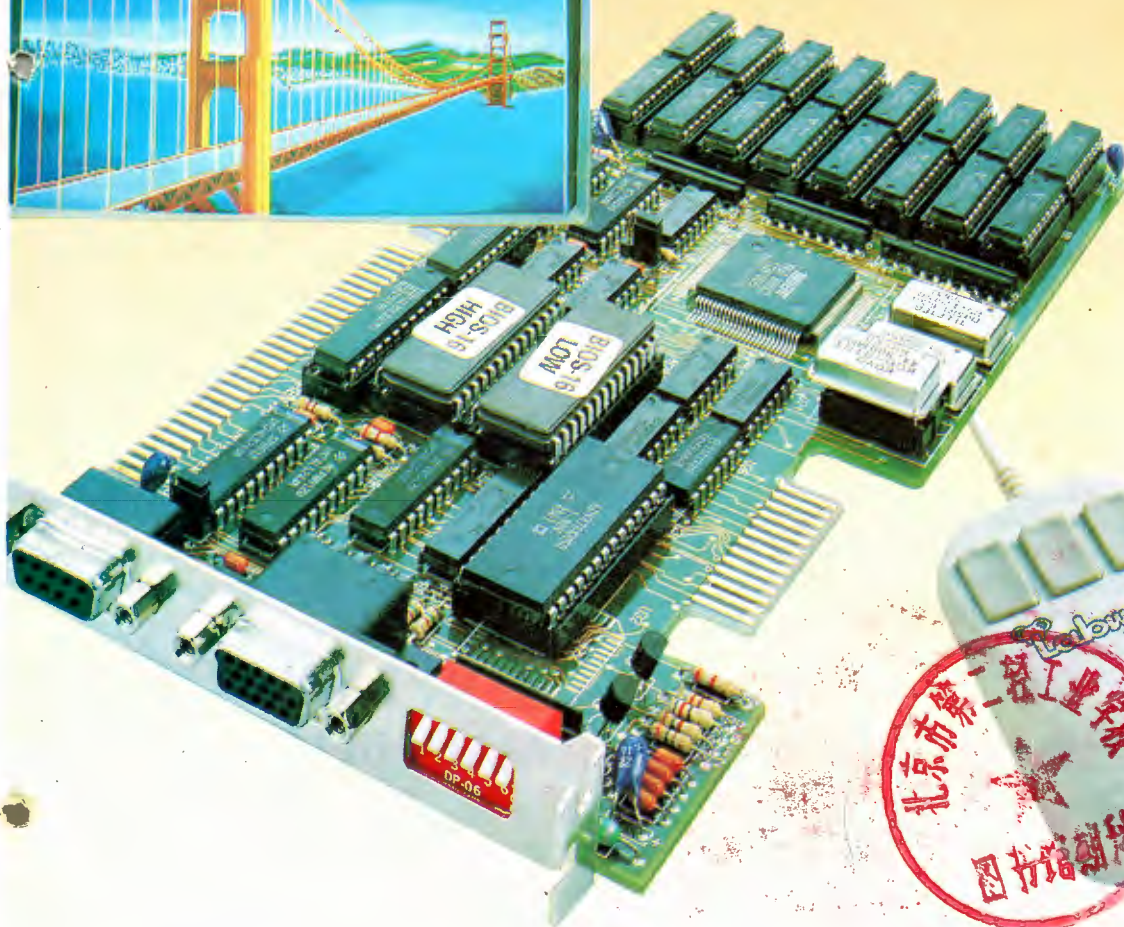
● 一九九一年 ● 总期第74期

5

電子

744

與 電腦



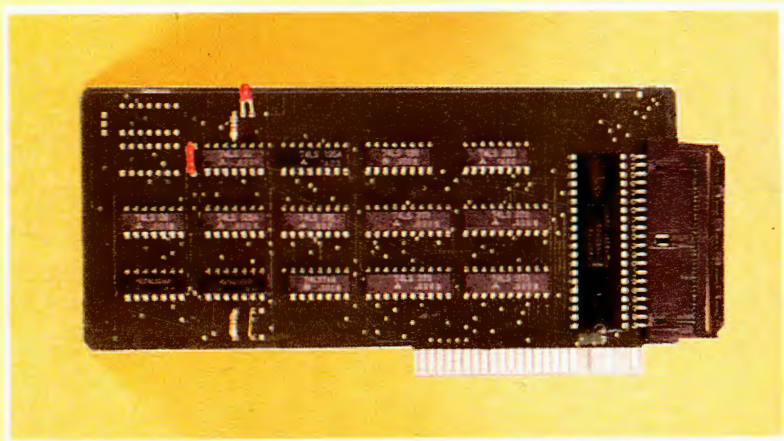
• ELECTRONICS AND COMPUTERS •

本刊向CEC-I及Apple II 广大用户推荐

● AP 51-I 型万能接口卡

● Z 80/PRT 卡

《电子与电脑》编辑部监制·经销



AP 51-I 万能接口卡

AP 51-I 型单片微机学习、开发卡 (万能接口卡)

AP 51-I 型单片微机学习、开发卡(万能接口卡)是一种配合 CEC-I 型中华学习机及 APPLE II 进行学习 and 开发 MCS-51 系列单片微机的多功能智能卡。在不用于单片微机学习和开发时,插在中华学习机或 APPLE II 上可做为一个串行通信接口卡,可用于中华学习机,APPLE II 微机与 IBM PC 及其兼容机之间通信、联网;或做为一个打印机接口卡,可用于全脱机方式打印;或做为一个时钟卡,可显示时间,控制主机定时动作;或做为并行接口卡;或做为 PC 磁盘驱动器兼容卡……。只要配上相应的软件,就可方便地用于再开发、灵活地完成各种功能。“万能接口卡”上的 8031 单片机与主机 6502 CPU 并行工作。8031 可以寻址主机的所有存储空间和 I/O 空间,完成如控制、显示、发声、打印、键盘输入、管理 DOS 等功能,以至接管主机的整个系统,使主机成为一台以 8031 为 CPU 的中华学习机。与该卡配套使用管理软件一片,说明书一本。

中华学习机、APPLE II 机 Z 80/PRT 卡

Z 80/PRT 卡是专为 CEC-I 型中华学习机、APPLE II 设计的一种 Z 80 及打印卡。插上该卡后,可运行 CP/M 操作系统下的 COBOL 程序、FORTRAN 程序和 dBASE II 数据库管理程序。尤其适用中华学习机的硬汉字,配上该卡您会感到如虎添翼。Z 80/PRT 卡上不仅有 Z 80 CPU 芯片及其外围电路,同时还做上了打印机接口电路,具有驱动打印机的功能。

购买办法:

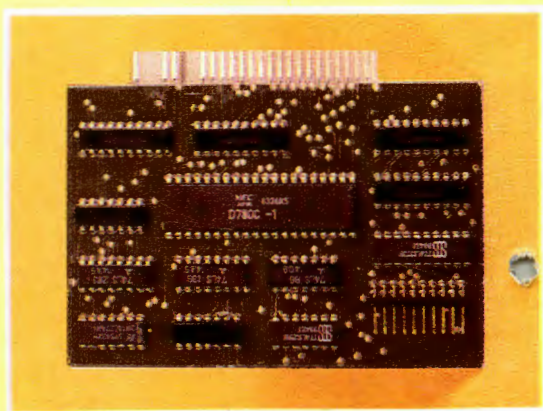
AP 51-I 万能卡(含软件一盘,说明书一本)
售价 330 元

Z 80/PRT 卡 售价 160 元

银行汇款:北京工商银行翠微路分理处 891238—72 电子工业出版社《电子与电脑》编辑部

邮局汇款:北京 173 信箱《电子与电脑》编辑部
邮编 100036

包装邮寄费另加 10 元



Z 80/PRT 卡

中国电子器材公司特区公司

本公司是机械电子工业部派驻特区的电子产品管理和经营机构,本公司立足特区,全部产品直接货源,价格性能比均优,来人来函订购均可,竭诚为广大用户提供资料咨询,引进新产品代办手续,欢迎比较和建立长期合作关系。

以下选介美国 MOTOROLA 公司部分集成电路现货价目表(参考)

1991 年 5 月 1 日

型 号	名 称	单 价(元)				
		1-24	25-49	50-99	100-999	1000 以上
MC1403U	电平驱动(同 LB1403)	2.28	2.09	2.00	1.90	1.80
MC1411P	周边七段驱动阵列	5.52	5.06	4.83	4.60	4.37
MC1412P	周边七段驱动阵列	5.04	4.62	4.41	4.20	3.99
MC1416P	周边七段驱动阵列	4.68	4.29	4.10	3.90	3.70
MC1436P	高压单运放	10.20	9.40	8.90	8.50	8.00
MC1458CP	通用双运放	2.64	2.42	2.31	2.20	2.09
MC1488P	RS232 用线驱动器	2.50	2.30	2.20	2.10	2.00
MC1496P	平衡调制解调器	2.30	2.10	2.00	1.90	1.80
MC1558U	双 MC1741 运放	11.40	10.30	9.80	9.30	8.80
MC1568L	双 $\pm 15V$ 跟从调节器(军品)	54.00	49.50	47.30	45.00	42.80
MC1596L	平衡调制解调器(陶瓷封装)	20.20	18.50	17.60	16.80	16.00
MC1596G	平衡调制解调器(金属圆壳)	19.10	17.50	16.70	15.90	15.10
MC3359P	窄带中频放大器 FM	4.80	4.40	4.20	4.00	3.80
MC3362P	窄带中频放大器 FM	10.56	9.68	9.24	8.80	8.36
MC3401P	若尔顿输入器运放	5.16	4.73	4.52	4.30	4.08
MC3403P	四运放	4.32	3.90	3.78	3.60	3.42
MC3450P	四 75107 上拉有效	11.40	10.45	9.98	9.50	9.30
MC3452P	四 75108 集电极开路	28.80	26.40	25.20	24.00	22.80
MC3470P	软盘读出放大系统	10.68	9.79	9.35	8.90	8.46
MC3485P	IBM360/370 总线驱动器	32.40	29.70	28.40	27.00	25.70
MC3486P	RS422/423 总线接收器	8.30	7.60	7.20	6.90	6.60
MC3487P	RS422/423 总线驱动器	7.20	6.60	6.30	6.00	5.70
MC142100	4X4X1 交叉点开关	4.68	4.29	4.10	3.90	3.70
MC14409P	脉冲拨号(附再拨号机能)	17.90	16.40	15.70	14.90	14.20
MC14419P	20F8 按键至二进制编码器	8.40	7.70	7.35	7.00	6.65
MC14495P1	BCD-7 段十六进制用锁/译码/驱动	10.80	10.00	9.50	9.00	8.60
MC145026P	遥控编码器	6.08	5.58	5.32	5.07	4.82
MC145027P	译码器	7.61	6.97	6.66	6.34	6.02
MC145028P	译码器	7.61	6.97	6.66	6.34	6.02
145030P	遥控编译码器	15.24	13.97	13.34	12.70	12.07
MC145100P	4X4X1 交叉点开关	15.00	13.80	13.10	12.50	11.90
MC145146P	四线串行锁相频率合成器	27.60	25.30	24.20	23.00	21.90
MC145152P	四线串行锁相频率合成器	34.60	31.70	30.30	28.80	27.40
MC145503	单片脉冲调制译码/滤波器	55.20	50.60	48.30	46.00	43.70
MC34063P1	DC-DC 转换器	9.96	9.13	8.72	8.30	7.89

公司地址:广东省深圳市振华路(华发北路口)电子器材公司

电话:354214 354345 图文传真:350876 邮编:518031

电挂:8410 开户行:深圳工商银行华强分理处 帐号:21-006500031

Apple IIe 主板芯片管脚信号测试表(使用逻辑笔)

Apple IIe

IC 名称(主板左至右)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40		
LS125	P	P	P	L	P	P	L	H	P	H	P	P	L	H	P	P	P	P	P	P	H																					
PLA	P	P	P	P	P	P	H	P	H	L	P	P	P	P	P	P	P	P	P	P	H																					
FF	H	H	L	P	H	P	P	L	P	P	H	P	P	P	H																											
LS244	L	P	P	P	P	P	P	P	P	L	P	P	P	P	P	P	P	P	P	P	L	H																				
LS245	L	P	P	P	P	P	P	P	P	L	P	P	P	P	P	P	P	P	P	P	H																					
LS374	P	P	P	P	P	P	P	P	P	L	P	P	P	P	P	P	P	P	P	P	H																					
LS244	L	P	P	P	P	P	P	L	P	L	P	P	P	P	L	P	L	P	L	H																						
视频	P	P	P	P	P	L	P	P	L	P	P	H																														
ROM存储器	L	P	P	P	P	P	P	P	P	P	P	P	P	P	H	H	P	H	P	P																						
管理单元	L	H	P	H	O	H	P	H	P	P	P	P	P	P	P	P	P	P	P	P	L																					
6502	P	P	P	P	P	P	P	L	H	P	P	P	P	P	P	H																										
I/O	L	L	P	P	P	H	L	L	P	L	L	H	H	P	H	O	P	P	P	P	P																					
管理单元	P	P	P	H	H	P	L	H	L	H	H	H	H	P	H																											
LS10	P	P	P	P	P	P	P	H	L	H	H	H	H	P	H																											
3-8译码器	H	P	P	P	P	P	P	H	P	P	P	P	P	P	L																											
RAMs	H	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	L																					
CD-ROM	H	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	L																					
LS02	L	H	L	P	P	L	L	P	P	P	H	P	L	H																												
EF-ROM	H	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	L																					
LS154	P	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	P	P																				
键盘ROM	L	L	L	H	L	L	H	H	P	P	P	L	P	P	P	H	L	P	P	L																						
LS251	O	L	L	H	P	O	H	L	P	P	P	H	H	H	H																											
RC741	P	L	L	P	P	P	H	O																																		
NE555	H	O	H	H	H	H	O	H	H	O	H	L	H	H	O	H																										
译码器	P	P	P	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L																					

在“空格键”
按后的状态

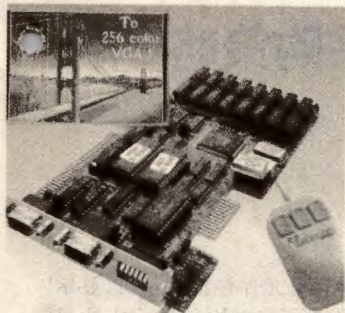
逻辑笔指示

P = 脉冲

H = 高

L = 低

O = 无显示



一九九一年

总期第74期

電子與電腦

• ELECTRONICS AND COMPUTERS •

目 录

• 综述 •

- 90年代的电脑技术和采购策略(下) 殷广济(2)
- CGA, EGA, VGA 余 红(3)
- 计算机技术的产物——条形码技术 于江苇(4)

• PC 用户 •

- 通用打印机状态检测程序的改进 夏伟文(5)
- 通用制表程序 赵 欣(6)
- 也谈 Cdbase Ⅲ 的纵向送数方式 王传信(9)
- PCTOOLS 的使用技巧 刘建明(11)
- 程序常驻内存(TSR)的实现方法 李 华(12)
- 防止非法使用 DBASE 程序的一种方法 黄世刚(14)
- 也谈 IBM 机图形存取 焦小友(14)
- 侦错核对程序 张永辉(16)

• 学习机之友 •

- 让 CEC-BASEC 也有自动行号功能 沈东山(17)
- 图开屏幕坐标的对应存储单元 葛海东(18)
- 神奇的 \$COBO 单元 李 峰(18)
- 怎样的读 CEC-I 学习机的汉字系统 刘文勤(18)
- 用机器语言解八皇后问题及其推广 卢祖兴 刘超宏(19)
- 也谈汉字平移显示 李茂学(19)

录

- 前台显示后台绘图的秘密 谢吉华(20)
- 演示抛体运动 白梅兰(21)
- 多单元电视接收天线设计程序 史文忠(22)
- 一个实用的造型表编辑程序 左震宇(23)

• C 语言初阶讲座 •

- 第三讲 程序控制结构(一) 李文兵 王玉华 蓝智斌(25)

• 电脑巧开发 •

- 语言报时钟的原理与实现 蔡莲红(28)

• 学用单片机 •

- 简易单片机电子琴 张培仁 张新宇(30)
- MCS-51 单片机在听力播放、发射系统中的应用 沈乃才 宋振环(32)

• 学装微电脑 •

- 用微电脑检查印刷电路板 易齐干(37)

• 初级程序员级水平考试辅导讲座 •

- BASIC 语言程序设计问答 李志刚 吕品(44)

- 封二、APPLE IIe 维修资料 香港余健康
- 封三、中国电子器材公司特区公司集成电路价目表

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:北京市邮政局

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:徐云鹏

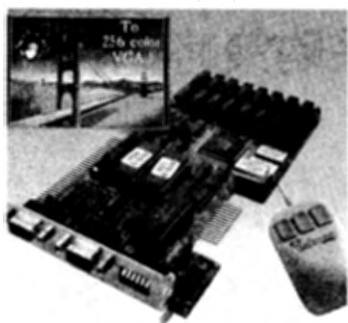
订购处:全国各地邮电局

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



一九九一年

总期第74期

电子与电脑

• ELECTRONICS AND COMPUTERS •

目

• 综述 •

- 90年代的电脑技术和采购策略(下) 殷广济(2)
CGA, EGA, VGA 余红(3)
计算机技术的产物——条形码技术 于江苇(4)

• PC 用户 •

- 通用打印机状态检测程序的改进 夏伟文(5)
通用制表程序 赵欣(6)
也谈 Cdbase II 的纵向送数方式 王传信(9)
PCTOOLS 的使用技巧 刘建明(11)
程序常驻内存(TSR)的实现方法 李华(12)
防止非法使用 DBASE 程序的一种方法 黄世刚(14)
也谈 IBM 机图形存取 焦小友(14)
纠错核对程序 张永辉(16)

• 学习机之友 •

- 让 CEC-BASEC 也有自动行号功能 沈东山(17)
图开屏幕坐标的对应存储单元 葛海东(18)
神奇的 \$COBO 单元 李峰(18)
怎样的读 CEC-I 学习机的汉字系统 刘文勤(18)
用机器语言解八皇后问题及其推广
..... 卢祖兴 刘超宏(19)
也谈汉字平移显示 李茂学(19)

录

- 前台显示后台绘图的秘密 谢吉华(20)
演示抛体运动 白梅兰(21)
多单元电视接收天线设计程序 史文忠(22)
一个实用的造型表编辑程序 左震宇(23)

• C 语言初阶讲座 •

- 第三讲 程序控制结构(一)
..... 李文兵 王玉华 蓝智斌(25)

• 电脑巧开发 •

- 语言报时钟的原理与实现 蔡莲红(28)

• 学用单片机 •

- 简易单片机电子琴 张培仁 张新宇(30)
MCS-51 单片机在听力播放、发射系统中的应用
..... 沈乃才 宋振环(32)

• 学装微电脑 •

- 用微电脑检查印刷电路板 易齐干(37)

• 初级程序员级水平考试辅导讲座 •

- BASIC 语言程序设计问答 李志刚 吕品(44)

- 封二、APPLE IIe 维修资料 香港余健康
封三、中国电子器材公司特区公司集成电路价目表

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:北京市邮政局

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:徐云鹏

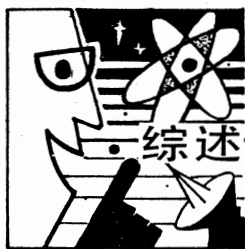
订购处:全国各地邮电局

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



90 年代的电脑技术和采购策略(下)

北方电脑公司 殷广济

80486 微处理器由一个高度优化的和 80386 处理器兼容的处理器和 80387 兼容的数学协处理器及 8KB 的数据与程序同一的 cache 合为一体,共有 120 万个晶体管(80386 为 25 万个),采用 1 微米的 CHMOS 1μ 工艺制作。具有 32 位的内部体系结构以及 32 位的地址总线 and 数据总线。在单个周期常用的指令,如存取指令的执行,使用流水线作业及 RISC 技术,采用一种新的脉冲串数据传送机制,由存储器顺序阅读 4 个 32 位字,以保持芯片上数据及指令高速缓存经常装满,由于晶体管之间讯号传送距离大为缩短,80486 的运算速度比 80386 芯片快 5 倍,工作于 33MHz 时,达到 20VAXMIPS,可充分满足如 AUTO CAD 和 Lotus1-2-3 等软件的数学运算中所需的浮点运算功能,其性能足以与 IBM3090 主机媲美。

80486 芯片是与 80X86 系列产品高度兼容,换言之,它可以执行为 IBM PC 机编写的,价值估计高达 150 多亿美元的众多通用软件及工程应用软件,并为 DOS, Windows/386, OS/2, UNIX Systemv/386 提供高水平的性能。与 DOS 兼容是 Intel 公司为 80 X86 系列微处理器发展重心,所以,80486 的基本设计要求会有效地应用于 IBM PC 系列及兼容机中。

据称,Intel 公司花在 80486 芯片上的投资超过 3 亿美元和 4 年时间,虽然并不是每个人都看好这颗芯片,认为它必能在众多的微处理器芯片中脱颖而出,成为小型机和工程工作站的中流砥柱,但无论如何,大部分专家都同意,80486 将会是半导体发展历史上的一个精心杰作,而且将会赋予明天的台式电脑真正易于使用的运行环境。你要买高档的 32 位电脑时,应认真考虑 80486 系统。

总之,微处理器技术不断朝着集成度高,速度快、功能多、体积小的高档领域迈进。

由于世界交通发达,人员往来增多,人们企望有一种比台式电脑小的、能携带,可使用直流电源、能进行文字及事务处理的电脑。80 年代开发膝上型(Laptop)及书本型(notebook)袖珍型(handhold)等便携式电脑并开始普及使用,正以爆炸速度赶上台式电脑,也许会成为电脑世界的骄子。

便携式电脑要求体积小,主板尺寸也必须相对缩小,故对芯片集成度要求较高,另一个更重要的要求在电力消耗方面,因此,此类芯片组都必须具备电源管

理功能。

芯片组必须对电脑的耗电元件包括磁盘机、显示器、CPU 以及一些机械动作进行完善的电源管理,电脑系统操作时,系统内部的各元(组)件,并非时时都处于运行状态,如果对这些不在运行状态的组件进行供电控制,就可延长整个系统的操作时间,此类电脑是从各元、组件一点一滴地把电力节约下来的。

液晶显示器模块所消耗的电力几乎是整个系统的一半,通常以外围芯片来侦测其使用状态,进而控制对液晶显示器的供电。

CPU 是另一个耗电元件,通常以切断对其供电及降低频率两种方法来达到省电目的。将 CPU 的电源切断,CPU 内的资料必须映射到存储器,以便在对 CPU 加电后继续执行工作,降低 CPU 频率是另一种省电方法,且内部资料亦能保存下来。

便携式电脑的有些特性是台式电脑所没有的,故标准的 BIOS 并不完全适用于便携式电脑,必须加入标准 BIOS 的欠缺的部分,便携式电脑具有如下特性:

CPU 档次从 8088、80286 到 80386 SX,内存 1MB 左右并可扩充,采用 3.5 英寸软磁盘和 2 英寸硬磁盘,其容量在 20MB 以上,通常为 40MB,液晶显示,EGA/VGA 标准,亦有采用气体/等离子技术,电源大多采用高效镍镉电池,充电后使用寿命均在 4-8 小时,重量在 3 公斤左右,膝上型稍重些。目前国内销售价格在 8-9000 元到 2000 多元之间,视其品种 PC,XT,286,386 不同而异。

中国情况有些特殊,也许不会有那么多人拿着便携式电脑转来转去,但在地质勘探,远航交通,办公室自动化及军队方面可能会有一定用途和相当大的驰骋空间。联想到近几十年来电脑工业的飞速发展及集成电路的更大规模集成,购买便携式电脑也许 5-10 年的期限内,不会为你使用的电脑的优越功能失去而感欣慰。

由于人工智能,特别是计算机辅助设计(CAD),计算机辅助工程(CAE)的兴起和软件开发的需要,要求有一种作为大型计算机在网络中的节点机,它具有良好的图形处理功能以便设计人员使用,这就是工作站。工作站的产品可依照其处理机的性能,存储器容量,显色能力,解析度,网络输入输出速度等分为初级、传统、超级工作站和绘图超级工作站等四大类。参见下表:

性能类别	初级工作站	传统工作站	超级工作站	绘图工作站
CPU 性能(MIPS)	2—12	4—15	8—20	20—40
FPU 性能(Mflops)	0.5—1.5	0.5—2.0	2.0—16	16—50
存储器(MB)	4—16	8—64	8—128	16—1024
硬盘容量(MB)	40—600	80—600	600—1300	1024 以上
显示屏尺寸(英寸)	12—19	12—19	15—19	19
显色能力	16—256	256—4096	256—16700	256—16700
解析度	1024*768 —1280*1024	1100*900 —1280*1024	1280*1024	1280*1024
网络速度 mbits/sec	10	10	10—125	10—1000

工程工作站主要应用领域如下:

应用化学:分子模型、量子化学、计量模型、程序工程、化学设计

工业自动化:离散程序控制、连续程序控制、自动测试装置

设计自动化:机械、电子/电机、建筑、工程

地球资源:地震分析、油田探测、地图

实时系统/资料收集与控制:建筑物自动控制、交通管制、大气监控、水质和污水控制、领空控制

绘图/动画/影像:电视/影片/动画/影像处理、电视传输

软件开发:电脑辅助软件工程(CASE),人工智能及其它软件开发

电脑辅助排版:电脑辅助技术排版、电脑辅助电子排版系统

医药:身体扫描、病人监控

从用户角度看,工程工作站比起大型、中型、小型电脑系统来价格低廉,并可以通过网络共用系统资源,减轻大、中、小型电脑系统的负荷,所以愿意购买工作站。

目前工作站市场主导者是厂家还不是用户。主要的供应商有 SUN, HP/APDLLO, DEC, Inter Graphic 和 Silicon Graphic。然而未来超大规模集成电路技术,开放性系统,电脑辅助软件工程以及面向目标处理技术的发展将会改变目前市场的势态,高性能工作站功能会更强,中级工作站价格会下降,以 80486 为基础的 PC 系统对目前的工作站在价格上会取得优势,所以廉价的 PC 是否也会成为

工作站系统是很值得人们关注的,其关键在于某些特殊工程用工作站的价格能否大幅度下降与高性能 PC 竞争,或是以 PC 为基础的廉价系统的功能能否和工作站一样令人满意,人们期待 PC 将成为明日工作站的开始。未来充满挑战和期待。

可以预见未来工作站的特点如下:(1)MIPS 将转换成 BIPS(每秒执行数十亿指令);(2)存储容量由 MB 转换成 GB(数十亿字节);(3)外存装置将由 GB 转成 TB(数百亿字节);(4)用户使用软件时与硬设备无关;(5)工作站将有标准的用户界面;(6)工作站与多媒体及视听装置相结合;(7)UNIX 系统将成为工作站的套装软件之一。

用户不仅会考虑优异的性能/价格比,也会对应应用软件的多寡,升级特性,可扩充性及厂家的可信度,服务及是否采用标准作为采购准则。

CGA, EGA, VGA

余 红

随着第一台微型计算机的诞生,市场对显示系统的需求永不满足,其性能在计算机科技人才的“勤奋+科学+应用”创新精神的努力下,一代代翻新,永无止境。目前,众多的电脑公司,纷纷推出高性能的显示卡。例如:

最早的 IBM PC 仅提供两种显示卡,即“单色显示卡”(简称 MDA)和 CGA 卡。MDA 只为单色显示器设计的,可显示 25×80 的文(数)字。分辨率为:720×350,而 CGA 可显示分离的红、蓝、绿信号,也可显示混合的视频信号,还可以显示以点绘制的图形以及

ASCII 码字符,其分辨率为 640×200。由于 MDA 只能显示文(数)字,而许多程序必须在屏幕上显示色彩,所以使用者选择了有彩色绘图功能但文字的清晰度较低的 CGA 卡。随着高层次微型计算机的推出,IBM 又发展出另一种提供较高文字及绘图分辨率的 EGA 卡,可以模拟 MDA 和 CGA,还可以在单色显示器上显示以点绘制的图形,可产生 16 色文(数)或 640×350 图形图象,其分辨率比 CGA 要高,更重要的是 EGA 同时显示的文字和图形远比 CGA 清晰,能使许多高级软件发挥其功能。以后又推出了“显示图形阵列”即 VGA 卡,

VGA 的程序设计界面和 EGA 相似,许多对 EGA 写的程序不经修改即可在 VGA 上运行,其分辨率比 EGA 高,文本方式下为 720×400 ,图形方式为 640×480 ,它可以从调色板中同时产生 256 种颜色。其他 $800 \times 600/16$ 色, $800 \times 600/256$ 色, $1024 \times 768/16$ 色的 VGA 卡花样翻新,图形功能强,使微计算机可显示的彩色画面,色彩缤纷。

由于 IBM PC 显示卡不支持汉字显示,国产长城计算机根据 IBM 的几种显示卡,研制出适合汉字显示的长城显示卡。如早期的长城微型计算机采用 IBM 的 CGA 卡和 014 汉字显示方式,原长城 014 板的显示方式有两种:字符方式,分辨率为 640×504 , 80×28 字符,16 种颜色。图形方式:分辨率为 640×480 ,8 种颜色。它主要配置于 GW0520CH 上,显示 25 行汉字外加三行汉字提示行。由于 014 方式与国际上流行的 CGA 及 EGA 方式有较大差别,不兼容。长城又开发出了另一种 CEGA 显示卡,它继承了 014 方式的汉字字符方式显示,并且与国际上流行的显示标准 CGA,EGA,VGA 高度兼容。其分辨率为 640×350 ,16 色图形显示方式,可以共享更多的软件,它还可以实现中西文图文叠加。CEGA 板也包括两种工作方式:方式 1 和方式 2。其中:

方式 1:兼容原长城高分辨率彩色图形适配器(014 板)的汉字,图形显示以及 IBM EGA 的中低分辨率的图形显示。

方式 2:兼容 IBM EGA 的所有显示方式,并增加显示图形汉字的功能。

CEGA 卡配置于长城 286,386 机型,通过 CEGA 板上

的开关以及功能调用等两种途径,可以灵活地切换这两种工作方式,

近一年长城新近推出的 CVGA 卡,全兼容 IBM VGA,并具有图形 $640 \times 480/256$ 色, $800 \times 600/16$ 色, $800 \times 600/256$ 色, $1024 \times 768/16$ 色等增强功能。全兼容长城 CEGA 汉字显示标准:支持字符/图形迭加功能,支持 16×16 点阵/ 24×24 点阵汉字字库的软切换。具有新型的单区 4 字节汉字方式,使汉化工作变得更简便,特别适于 DOS4.0, WINDOW 等高级多窗口软件,其配置于长城 386 以上的机型。

综上所述:

IBM PC 常见的显示卡有:

CGA 卡:彩色图形卡,英文名:Color Graphice Array.

EGA 卡:加强型图形卡,英文名:Enhanced Graphics Adapte.

VGA 卡:显示图形阵列,英文名:Video Graphics Array.

长城微型计算机中对应的显示卡有:

014 板:长城高分辨率彩色图形适配器

CEGA 卡:中文增强型图形卡,Chinese Enhanced graphics Adapter.

CVGA 卡:中文显示图形阵列,Chinese Video Graphics Array

显示卡作为计算机的“视觉神经”,将会越来越丰富发达。

(封面上给出的是外国公司有关 VGA 卡的图片资料)

计算机技术的产物——条形码技术

山东省科学院情报所 于江苇

许多人可能已注意到,商店的货架上有许多商品上印有条形码标志。这种标志的作用是什么?为什么要使用这种标签呢?我想这个问题应该从计算机的数据输入方法谈起。

大家知道,计算机的运行速度相当可观,现已达到每秒钟几亿次;另一方面,由于计算机数据输入技术发展还不尽人意,因此人们一直在寻找一种快速、准确的自动输入技术,为了适应这种需要,条形码技术就从此诞生了。

一、条形码符号的基本结构及种类

一个完整的条形码符号是由空白区、起始位、信息位、校验位(可选)、终止位和空白区组成。如图所示:

空白区	起始位	信息位	校验位	终止位	空白区
-----	-----	-----	-----	-----	-----

空白区:顾名思义即没有任何印刷符号和信息的区域。它用来提示扫描器准备扫描条形码。

起始位:在扫描阅读时,当判定有起始位存在时,扫描器才会继续处理起始位以后的一系列信息。起始位的使用,给条形码增加了保险。

终止位:使用终止符,可避免不完整信息的输入。

校验位:它是校对读入的信息是否有效所采取的一种手段。

以上我们所谈到的是各种条形码制所具有的共性,而具体到每一种码制本身的具体特性来说,比如其密度、长度、连续性、宽条的个数、奇偶性等都有其不同之处了。条形码的种类在近四十年内逐渐增加起来,但就其本身的特性及实用性来讲,常见的有:

1. UPC 码(Universal Product Code):它是一种只包含数字的商品零售代码,它被广泛应用于各种(下转 43 页)



PC 用户

通用打印机状态检测程序的改进

夏伟文

本刊在一九九一年第二期上发表了《通用打印机状态检测程序》一文,对于不了解程序原理的人,在实际应用时很容易把参数搞错。为了解决这个问题,作者又把程序作了进一步的修改。

程序的原理在上面提到的文章中写得很详细,这里就不介绍了。新程序的改进之处是应用了数据文件,简化了判断用户打印机参数的过程,具体方法如下。

(一)用下面的程序 1 来判定用户所使用的打印机的状态参数,也就是运行该程序时根据提示将打印机分别置于正常、断电、缺纸和未联机状态下即可将不同的状态状态寄存器的数值(十进制)存入数据文件 PRCH.DAT。

(二)将 PRCH.DAT 拷入当前目录中,就可以使用下述的程序 2 对打印机状态随时测试。

程序 1:

```
10 OPEN"CHPR.DAT"FOR OUTPUT AS#1'打开数据文件
20 FOR I=1 TO 4
30 READ A$
40 CLS:LOCATE 10,18:PRINT A$;"然后按任意键..."
50 IF INKEY$=" "THEN 50
60 REM 测定参数
70 DEF SEG=&H40 '指向 BIOS 数据区
80 B=PEEK(9)*256+PEEK(8)+1'求状态寄存器的地址
90 A=INB(B) '求状态参数
100 WRITE #1,A '将参数写入数据文件
110 NEXT I
120 CLOSE #1 '关闭数据文件
130 SYSTEM
140 DATA"请将打印机置于正常状态,"
150 DATA"请将打印机置于未联机状态,"
160 DATA"请将打印机置于断电状态,"
170 DATA"请将打印机置于缺纸状态,"
```

程序 2:(以 TH 3070 打印机为例)

```
5 REM 文件名 CHPR.BAS
10 SCREEN 0,0,0:WIDTH 80:CLS:KEY OFF
20 OPEN"CHPR.DAT"FOR INPUT AS#1'打开数据文件
30 FOR I=1 TO 4:INPUT #1,A(I):NEXT I'从数据文件读入参数
```

40 CLOSE #1'关闭数据文件

50 DEF SEG=&H40'指向 BIOS 数据区

60 B=PEEK(9)*256+PEEK(8)+1'求状态寄存器的地址

70 IF INP(B)=A(1) THEN 130'如果打印机准备好则转到 110 句退出 BASIC

80 BEEP:COLOR 7,K=1'否则发出警报并在屏幕上显示出提示

90 IF INT(B)=A(2) THEN COLOR 7,0:CLS:LOCATE 10,21:COLOR 20,1:PRINT"注意";:

COLOR 2,0:PRINT"请按打印机面板上的 SELECT(联机)键!";GOSUB 150:BEEP:GOTO 90

100 IF INP(B)=A(3) THEN COLOR 7,0:CLS:LOCATE 10,21:COLOR 20,3:PRINT"注意";:

COLOR 4,0:PRINT"请打开打印机电源开关!";GOSUB 150:BEEP:GOTO 100

110 IF INP(B)=A(4) THEN GOSUB 140:GOTO 110

120 IF INP(B)<>A(1) THEN COLOR 7,0:CLS:LOCATE 10,25:COLOR 20,1:PRINT"注意";:

COLOR 15,0:PRINT"打印机尚未准备好...";GOSUB 150:K=K+1;

IF K>7 THEN 70 ELSE 120

130 CLS:COLOR 7:SYSTEM

140 COLOR 7,0:CLS:LOCATE 10,27:COLOR 20,2:PRINT"注意";:COLOR 14,0;

PRINT"请给打印机装好纸!";PRINT:PRINT TAB(12);:

PRINT"然后看面板 SELECT(联机)和 PAPER END(缺纸)指示灯是否正常!";GOSUB 150;

BEEP:RETURN

150 FOR I=1 TO 3200:NEXT I:RETURN

该程序的应用主要是面向于那些对计算机原理一无所知的录入员和初级操作员,他们在使用应用软件时,如果打印操作有误,操作系统会给出一些英文指示,这时往往令操作者们手足无措。有了这个程序,可以用汉字信息帮助他们改正错误,以免由于误操作而造成死机或丢失信息:尤其是在应用广泛的 dBase 系统中开发应用软件时,调用该程序可确保打印功能的正常实现。

上述程序已在 GW-386/20、GW0520-CHII 型机上调试通过。

通用制表程序

济南 山东省机械工业学校 赵欣

一个管理程序要具有实用性,应使用户享有充分的“自主权”。例如打印数据库中哪些字段,打印符合什么条件的记录,应由用户决定。我在为某单位编制管理程序时,编制了一个具有上述功能的通用制表程序,它可以对任意以汉字作字段名的数据库进行制表打印,数据库的结构改变时,仍能按改变后的库进行打印,而制表程序本身无需修改。现将该程序介绍大家,愿能起到抛砖引玉的作用。

一、程序的功能及设计思想

首先介绍几个数据库:1区是需制表打印的数据库,其库名由用户键入,在程序中为 &na(下简称原库)。2区是打印库 PRDB2. PRG,该库是全 C 型字段,各记录值与原库完全相同,但字段宽度有所改变。3区是打印库的结构库 STRU3. DBF,其中 FIELD__NAME 字段存放的是打印库(也是原库的)字段名, FIELD__LEN 字段是打印库各字段的宽度。4区是表线库 LINE4. DBF,字段名与打印库完全相同,各字段值都是一段表线,库中只有 3 个记录,分别用来产生上表线 la、中表线 lb 和下表线 lc。5区是条件库 COND5. DBF,只有一个字段名为“条件”的 C 型字段,存放各种打印条件。

1. 生成打印库和表线库

这一功能是由过程 PR1 来实现的。它依据原库生成其结构库,再对结构库做三点改造。第一:描述类型的 FIELD__TYDE 字段值全改为 C;第二:描述字段宽度的 FIELD__LEN 字段值改为值宽度和栏目名(即原库字段名)宽度中较大的。例如:性别字段,字段值“男”或“女”只占 2 字符宽,而栏名“性别”占 4 字符宽,因此相应的 FIELD__LEN 值取 4;第三:由于制表符占 2 字符宽度,所以还要把 FIELD__LEN 的值都改为偶数。这样修改后的结构库就成了打印库所需的结构库。在此基础上又生成了表线库。为了条件打印的需要,还将原库和打印库建立了关联。

2. 产生打印条件

调用过程 PCOND 可产生打印条件,也可以无条件打印。用户可以自己键入打印条件,也可用条件库中已有的条件。在设计程序时一旦用户提出增加或修改打印条件,只需修改条件库就可以了。

3. 分页制表

该程序可分页制表,使用宽或窄打印纸。每页行数、起始页号、页标题及制表时间全由用户决定。

4. 用户任选打印字段

主程序第 31 至 44 行将原库的所有字段(也是打印库的字段)均显示出来,供用户按项号选择需打印的字段。(该项号也是 STRU 库的记录号)并且可以随时

显示当前的打印宽度,当用户不想再选项号或已超过纸的宽度时会自动停止选项,开始打印。

二、几点说明

1. 关于打印库的形成

一般使用 APPEND FROM 语句要求当前库的字段名和类型与源数据库要一致,但实验证明,用该语句可以生成打印库,不管原库中相应字段是 N、C、L、D 都不会丢失数据。

2. 非汉字字段名的处理

由于本制表程序是用原库中的字段名作为表中的各栏目名,若是非汉字字段名可以修改 STRU3. DBF,在该库中再增加一个 C 型字段存放所有原库字段的栏目名,再对程序稍加修改就可以了。

3. 打印内容的取中

为了使各项打印内容能打印在本栏的中间位置,还可再修改过程 PR1,由于各栏的宽度均存在 STRU3. DBF 库的 FIELD__LEN 字段中,而打印库各字段实际宽度又可用 TRIM 和 LEN 函数配合求出,只要根据这两个宽度就可将打印库中的各字段值取在中间位置。

4. 打印条件 &COND

主程序中 LOCATE FOR 语句中的打印条件是通过宏代换产生的,但有些版本的 CDBASE-III 不能对这种宏代换产生的逻辑表达式进行正确解释。这时可按下面的思路修改程序。第一步:先建一个语句库,其中只有一个 C 型字段,库中的第一条记录的值是“LOCATE FOR”+COND,这式中 COND 是由过程 PCOND 产生的打印条件。第二条记录值是“RETURN”。第二步:由语句 COPY TO C:DTOP. PRG SDF 就可以产生一个外部过程。这种用数据库文件生成一个程序文件的方法,经实验证明完全可行。第三步:在主程序适当处调用 C:DTOP. PRG 就可以了。

1: * * 用户任选字段、条件的通用制表程序 c.prna. prg

2: clear

3: clear all

4: set talk off

5: set device to screen

6: set print off

7: set procedure to c:proc

8: do prli

9: p="y"

10: do while p="y" . or. p="Y"

11: clear

12: @2,5 say "请开打印机,针头对准纸的第一行"

13: a=1

14: w=170

```

15:pn=1
16:pa=66
17:@3,5 say "1 宽纸 2 窄纸;请输入 1 或 2" get a pict "99"
18:@4,5 say "请输入分页行数(默认每页 66 行)" get pa
19:@5,5 say "起始页号(默认值 1)" get pn
20:read
21:if a=2
22:w=80
23:endif
24:set margin to 0
25:bt=space(60)
26:ti=space(11)
27:@6,1 say "请输入制表时间" get ti
28:@7,1 say "请输入报表的标题" get ti
29:read
30:bt=trim(bt)
31:sele 3
32:r=0
33:clear
34:go 1
35:do while .not. eof()
36:c=0
37:do while (.not. eof()) .and. ((c+len(trim(FIELD__
NAME)))<=76)
38:s=str(recno(),2)+trim(FIELD__NAME)+" "
39:@r,c say s
40:c=col()
41:skip
42:enddo
43:r=r+1
44:enddo
45:@ 7,0 clear
46:@ 8,1 say "请按项号选择需打印的项目(输入 0 结束)"
47:z="|"
48:p="|"
49:s="z+"
50:sd="|"
51:sc=0
52:la" r"
53:lb=" |"
54:lc=" L"
55:do while .t.
56:h=" "
57:@ 9,10 say "输入需打印的项号" get h pict "999"
58:read
59:if h=" 0".or.h="0".or.h=" 0".or.h="000"
60:@ 7,0 clear
61:@ 8,10 say "请停止输入项号"
62:exit
63:else
64:go val(h)
65:f=FIELD__LEN
66:f1=len(trim(FIELD__NAME))
67:st=trim(FIELD__NAME)+" +q+"
68:if f=f1

```

```

69:sbt=trim(FIELD__NAME)+"|"
70:else
71:f=(f-f1)/2
72:sbt=space(f)+trim(FIELD__NAME)+space(f)+"|"
73:endif
74:sc=sc+FIELD__LEN+2
75:l2=len(sb+sbt)
76:if l2>=w
77:@ 7,0 clear
78:@ 8,10 say "已超过打印纸的宽度,停止输入项号"
79:exit
80:else
81:s=s+st
82:sb=sb+sbt
83:fn=trim(FIELD__NAME)
84:sele 4
85:go 1
86:la=la+&fn
87:go 2
88:lb=lb+&fn
89:go 3
90:lc=lc+&fn
91:sele 3
92:@ 7,0 clear
93:@ 7,1 say sbt
94:@ 8,1 say "目前的打印宽度是:"+str(sc+2,4)
95:endif
96:endif
97:enddo
98:ls=len(s)-3
99:s=substr(s,1,ls)+" +z"
100:l2=len(sb)-2
101:sb=substr(sb,1,l2)+" +|"
102:j=len(la)-2
103:la=substr(la,1,j)+" +r"
104:lb=substr(lb,1,j)+" +|"
105:lc=substr(lc,1,j)+" +L"
106:cond=" "
107:do pcond
108:set print on
109:set device to print
110:L=len(la)
111:cb=(1-2*len(bt))/2
112:sele 1
113:locate for &cond
114:do while .not. eof()
115:eject
116:@ prow()+1,cb say chr(27)+"ID"+bt
117:?? chr(27)+"IA"
118:@ prow()+1,L/2 say "第"+str(pn,3)+"页"
119:@ prow(),L-15 say ti
120:@ prow()+1,0 say la
121:@ prow()+1,0 say sb
122:do while (.not. eof()) .and. (prow())<=pa-5)
123:sele 2

```



```

124:@ prow()+1,0 say lb
125:@ prow()+1,0 say &s
126:sele 1
127:continue
128:endo
129:@ prow()+1,0 say lc
130:@ prow()+1,0 say " "
131:* 下一句是为不致漏打最后一行而设
132:?? chr(13)+chr(10)
133:pn=pn+1
134:endo
135:if prow()<pa-1
136:@ pa-1,0 say " "
137:?? chr(13)+chr(10)
138:endif
139:set print off
140:set device to screen
141:@ 7,0 clear
142:@ 9,10 say "继续打印吗? Y/N"
143:set console off
144:wait to p
145:set console on
146:endo
147:close proce
148:clear all
149:clear
150:set talk on
151:return

```

```

1:* * 过程文件 c:proc.prg
2:procedure pcond
3:* 生成打印条件
4:sele 5
5:use c:cond5
6:store " " to c,cl
7:m="y"
8:do while m$ "Yy"
9:clear
10:@ 3,10 say "1. 条件打印 2. 无条件打印 请输入 1 或 2"
get c
11:pict "9"
12:read
13:* 无条件打印时 cond="recno()>0"
14:cond="recno()>0"
15:if c="1"
16:@ 5,10 say "1. 自己输入条件 2. 使用已固化的条件" get
c1 pict
17:"9"
18:read
19:if c1="2"
20:sele 5
21:disp all
22:input "请输入您选中条件的记录号" to rno
23:go rno
24:cond=条件

```

```

25:else
26:cond=space(50)
27:@ 6,10 say "请输入打印条件" get cond
28:read
29:endif
30:cond=trim(cond)
31:endif
32:wait "以上各项输入需修改吗 Y/N?" to m
33:endo
34:return
35:procedure prli
36:* 生成打印库 c:prdb2.prg 和表线库 c;line4.dbf
37:* 原库与打印库建立关联
38:set safe off
39:clear
40:sele 1
41:accept "请输入需打印的数据库名" to na
42:use &na
43:@ 4,15 say "正在生成打印库和表线库,请耐心等待"
44:copy to c:stru3 structure exte
45:use
46:sele 2
47:use c:stru3
48:repl all FIELD__ TYPE with "C"
49:repl all FIELD__ LEN with len(trim(FIELD__ NAME))for
50:FIELD__ LEN<len(trim(FIELD__ NAME))
51:repl all FIELD__ LEN with FIELD__ LEN+1 for
52:int(FIELD__ LEN/2)<>FIELD__ LEN/2
53:creat c:prdb2 from c:stru3
54:append from &na
55:sele 3
56:use c:stru3
57:copy to c:stru6
58:sele 4
59:use c:stru6
60:repl all FIELD__ LEN with FIELD__ LEN+2
61:creat c;line4 from c:stru6
62:append blank
63:append blank
64:append blank
65:sele 6
66:use c:stru6
67:index on FIELD__ LEN to c:st6
68:go bott
69:* 生成两条最长的粗、细实线 s1,s2
70:k=(FIELD__ LEN-2)/2
71:store " " to s1,s2
72:i=1
73:do while i<=k
74:s1=s1+"—"
75:s2=s2+"—"
76:i=i+1
77:endo
78:set index to
(下转第 15 页)

```

也谈 CdBASE III 的纵向送数方式

王传信

“电子与电脑”90年11期所载“dBASE III 中常规送数方式的改造”一文(以下简称“改”文),提出了一个纵向送数的课题。这个课题提得好,带普遍性。只可惜,那段程序还有毛病,故在 PC 机上无法运行。下面就三个方面的看法,共同探讨一下:

一、“改”文程序的毛病

该程序有两种错误(语法错误与逻辑错误),故在 PC 机上无法运行。此外,还有一些不足之处,这当然是不影响运行的。这里分九点,简要地谈一下。

1. 前三个 STORE 语句的毛病

(1)变量 CONDITION 是字符型的,故 T 就要写成“T”才行。若用逻辑型的,则 T 要写成 .T. 的。

(2)FIELDNAME、A1 等九个字符型变量,用空串“”(或一个字符‘ ’)来定义,语法上是对的,但不实用。因为 GET 句遇到空串,就一闪而过,不让操作者送数。即使是一个字符的串,我们也无法送入多字符的数据。

2. DO WHILE .CONDITION. 与 @1,10 SAY ‘继续吗(Y/N)?’ GET CONDITON 不配套。其中 Y 与 T 不一致,故程序总是中断停运。这样,每输一个学科成绩,就得重新 DO... 一下。

3. DATABASE1. DBF 文件名的长度 > 8 个字符,故在命名上有毛病,但还可在机上凑合用。

4. REPLACE FIELDNAME WITH A&P 中的 FIELDNAME 不是字段名,故要加 & 宏代换符号。

5. USE DATABASE1 的位置不当,故每输入一个学科名,就要重新开一次库。此外,在程序结束前,没有 USE 的关库语句,不安全。

6. 屏幕显示,新老数据混杂在一起,不清晰,看上去非常伤神。

7. DATABASE1 库的库结构,最好事先交待一下,以利阅读。

8. 库内各学科成绩,宜用数字型字段。若用字符型而又要让数字排齐的话,那未送数的要求就高,白白浪费时间。

9. 用功能键直接输入学科字段名,做得对。但要注意, F1 是不准再定义的。

PROG1. PRG 是修改后的程序,可在 PC 机上运行。

二、关于充分利用 dBASE III 本身的功能问题

谈到横向与纵向送数, dBASE III 本身就有此功能,只是人们很少注意。BROWSE 就是既可用直接命令方式又可用程序方式进行全屏幕的横向与纵向输入的命令语句。BROWSE 纵向送数的操作方法如下:

1. 打开 DATABASE 库,键入 BROWSE 命令。屏幕

横向显示全部库字段。纵向显示记录。

2. 按 CTRL-HOME 键,屏幕顶行就显示下列五种功能的提示(光标停在 BOTTOM 处):

BOTTOM TOP LOCK RECORD# FREEZE

3. 把光标移到 FREEZE 处,回车,屏幕顶行就提问 FREEZE 的字段名。

4. 键入学科字段名(可用已定义的 F2-F10 功能键),回车,屏幕光标就停在该字段的第一个记录处。

5. 现在就可以纵向送数了。光标只会上下移动而不横走。待输送完毕后(若有必要,还可增加新的记录),按 CTRL-W 键,然后关库。

假如学科太多, BROWSE 后,要送数的学科却躲在屏幕右侧的外端时,则可以用 LOCK 功能把学生姓名字段锁住,然后用 CTRL-→ 把该学科字段移入屏内,再进行 FREEZE 等操作。

PROG2. PRG,就是用 BROWSE 操作的程序(比上述直接命令操作法,还要简单)。

三、关于纵向间接方式送数程序的编写

“改”文程序的特点是采用了间接方式送数(用 A1-A8 内存变量间接送数,一次 8 个),比直接开库送数要好一点,但屏幕的利用率太低。另外,当第 8 个数据输错时,有可能无法当场纠正。 PROG3. PRG 采用了逐个显示,逐个送数,逐个检查,逐个置换,每屏可容纳 7×5 即 35 个学生的成绩的间接送数方式。同时,程序还避免了新送数据混杂的毛病,屏幕显示清晰。估计它是可以满足运行的要求的。

上述三个方面的看法,不一定全对,仅供商讨之用。

附上两种数据库的库结构与三种程序的清单。

数据库结构 : A:DATABASE. dbf

记录个数 : 18

最后更新日期: 01/14/91

字段	字段名	类型	宽度	小数
1	学生姓名	Character	8	
2	S1	Numeric	6	2
3	S2	Numeric	6	2
4	S3	Numeric	6	2
5	S4	Numeric	6	2
6	S5	Numeric	6	2
7	S6	Numeric	6	2
8	S7	Numeric	6	2
9	S8	Numeric	6	2
10	S9	Numeric	6	2
总计			63	

数据库结构 : A:DATAS1. dbf

记录个数 : 18

最后更新日期: 01/14/91

字段	字段名	类型	宽度	小数
1	学生姓名	Character	8	
2	S1	Character	8	
3	S2	Character	8	
4	S3	Character	8	
5	S4	Character	8	
6	S5	Character	8	
7	S6	Character	8	
8	S7	Character	8	
9	S8	Character	8	
10	S9	Character	8	
总计			81	

```

* PROG1. PRG *
SET TALK OFF
STOR . T. TO CONDITION
STOR ' ' TO FIELDNAME
USE DATABASE
DO WHILE CONDITION
CLEA
@ 1,10 SAY '请输入学科名:' GET FIELDNAME
READ
GO TOP
do while . NOT. EOF()
@ 2,0 CLEA
STOR ' ' TO A1,A2,A3,A4,A5,A6,A7,A8
X=0
DO WHILE X<8 . AND. (. NOT. EOF())
X=X+1
P=STR(X,1)
@ X+1,17 SAY 学生姓名 GET A&P
SKIP
ENDD
READ
SKIP -1 * X
X=0
DO WHILE X<8 . AND. (. NOT. EOF())
X=X+1
P=STR(X,1)
REPL &FIELDNAME WITH VAL(A&P)
SKIP
ENDD
endd
@ 1,10 SAY '继续吗(. T. /. F.)?' GET CONDITION
READ
ENDD . T.
USE
RETU

```

```

* PROG2. PRG *
SET TALK OFF
USE DATABASE

```

```

BROW
USE
RETU

* PROG3. PRG *
SET TALK OFF
DY = 'Y'
SUBJNAME=SPAC(8)
USE DATABASE
DO WHILE UPPE(DY) = 'Y'
CLEA
@1,30 SAY '请输入学科名:' GET SUBJNAME
READ
CLEA
COL=0
do while COL<65
@1,COL SAY '学生姓名'+subjname
COL=COL+16
endd
COL=0
ROW=2
GO TOP
do while . NOT. EOF()
DX = 'N'
A=SPAC(6)
DO WHILE UPPE(DX) # 'Y'
@ ROW,COL SAY 学生姓名 GET A
@9,COL SAY '对吗(Y/N)?' GET DX
READ
ENDD
REPL &SUBJNAME WITH VAL(A)
ROW=ROW+1
IF ROW>8
ROW=2
@9,0 CLEA
if COL<64
COL=COL+16
else
COL=0
@2,0 CLEA
endi
ENDI
SKIP
endd
CLEA
? CHR(7)
@5,30 SAY '继续吗(Y/N)?' GET DY
READ
ENDD
CLEA
USE
RETU

```

PCTOOLS 的使用技巧

刘建明

PCTOOLS 是 PC 机最为流行的磁盘工具软件之一,它为广大用户提供诸多的 DOS 功能。若能熟练地使用 PCTOOLS 无疑会给您的工作带来莫大的方便,本文是笔者的一些体会。

1. PCTOOLS 常驻内存

PCTOOLS 为用户提供了常驻内存的功能,使用方法是:键入

PCTOOLS /R128 (Enter)

此时屏幕提示用户 PCTOOLS 已常驻内存等信息。此后,用户可以随时启动 PCTOOLS,只要键入(Ctrl)+(Esc)即可,即先按(Ctrl)不放,再按一下(Esc)键。其中的/R128 选项说明了 PCTOOLS 常驻内存时将占用 128K 的空间。另外 PCTOOLS 还有两个选项/BW 和/Fn。/BW 选项说明 PCTOOLS 将在黑白显示状态下工作,/Fn 选项中的 n 为 1-9 的数字,它定义了一个功能键来代替上述的(Esc)键,如:键入

PCTOOLS/R128/F4/BW (Enter)

后,用户可以随时按下(Ctrl)+(F4)键启动黑白显示的 PCTOOLS。

2. 两个窗口

为了便于下面的叙述,首先介绍一下 PCTOOLS 的两个窗口,它们是针对文件操作的 File Function 窗口和针对磁盘操作及特殊操作的 Disk and Special Function 窗口。除了早期的版本,绝大多数 PCTOOLS 都是基于这两个窗口进行操作的。在窗口的下方,有一个命令提示区,用户只要键入命令中的大写字母,就可以执行该命令。两个窗口可以用(F3)键方便地切换。在文件操作窗口中,用户可以用上移键、下移键和(Enter)键选择需要的文件,文件被选择后由 PCTOOLS 给出入选序号,并由特殊色显示其文件名和入选序号。

3. 修改文件的属性

文件的属性有只读(Read Only)、隐含(Hidden)、系统文件(System)、文档文件(Archive)等。如果文件属性被修改为只读,则用 DEL 命令不能将其删除。如果文件属性被修改为隐含,则用 DIR 命令无法列出文件名。虽然许多 DOS 提供了修改文件属性的外部命令,但用 PCTOOLS 修改文件属性更方便。方法是进入文件操作窗口,用户选择好需要修改的文件后,键入“A”,执行“Attrib”命令,屏幕给出相应的窗口提示,根据提示,用上移、下移和(Enter)键将相应属性的开关由 OFF 改为 ON,最后键入“U”将其写入磁盘。一个文件修改好后,可用左移和右移键继续修改其它入选文件的属性。

4. 隐含子目录

许多 DOS 都为用户提供了修改文件属性的功能,

却没有提供修改子目录属性的功能。而用 PCTOOLS,我们可以在任何一级子目录上完成这项工作。首先,记住需要修改属性的子目录名,然后进入磁盘和特殊操作窗口,键入“F”,执行“Find”命令,PCTOOLS 提示您输入要查找的字符,此时,键入子目录名,然后回车,则 PCTOOLS 帮您查找该子目录名在磁盘上的位置。查到以后,键入“E”,进入编辑状态,光标停留在目录名的第 0 个字节上,移动光标至同一行的第 11 个字节上,将其值修改为 17,再按下(F5),就将其写入磁盘。最后,按(Esc)键,退回磁盘和特殊操作窗口,修改就完成了。

由于 PCTOOLS 是在整个磁盘上查找子目录名,因此子目录名不能和磁盘上的其它文件及子目录同名,否则将会引起意想不到的错误。

5. 进入加密的子目录

子目录名中含有不可显示的 ASCII 码或半个汉字时,或者子目录被隐含时,其他人想进入该子目录是很困难的。用 PCTOOLS 就可以不受这些限制。每当按下功能键(F10)时,PCTOOLS 等待用户输入一个驱动器号并回车以后,PCTOOLS 就开始查询磁盘,一旦发现磁盘上存在子目录,就显示一个包括各个子目录的树形表,用户可以方便地移动光标进入该子目录,而不需要键入该子目录名。进入子目录后,就可以对其中的文件进行 COPY 等操作。

6. 文件目录的排序

在大型软件的开发过程中,往往要对大批量的文件进行存储、删除、修改。文件在目录中的位置不固定,令人眼花缭乱,甚至找不到文件。PCTOOLS 能将文件目录按照某种规律排序,让同一类文件排在一起,这样就不会显得零乱了。

进入 PCTOOLS 的 File Function 窗口,键入“S”,执行“Sort”命令,PCTOOLS 提示:(F7)按文件名排序,(F8)按扩展名排序,(F9)按文件所占空间大小排序,(F10)按文件生成时间排序,“A”正向排序,“D”倒排序,“U”排序结果存盘等信息。按照提示,您可以根据需要排序。

7. 恢复被删除的文件

如果因误操作而删除了没有备份的文件,将是非常遗憾的。用 PCTOOLS 可以快速、方便地恢复这些被删除的文件。进入 PCTOOLS 的磁盘和特殊操作窗口,键入“U”,执行“Undelete system”命令,屏幕提示用户选择一个子目录。用户选择子目录后,屏幕上列出该目录中所有被删除文件的文件名。从屏幕上可以看出这些文件名的第一个字符已被删除标志所覆盖,用户移动光标选择好需要恢复的文件。键入“G”,PCTOOLS 开始

恢复文件。首先,PCTOOLS 提示用户恢复文件名的第一个字符,再按下(F1)键,由 PCTOOLS 自动恢复文件的簇等信息。照此,依次恢复其他被删除的文件。

除此之外,PCTOOLS 还有许多功能和用法,而且

随着 PCTOOLS 版本更新,功能不断增强,完善。由此可见,熟练地使用 PCTOOLS 对使用 PC 机的专业人员和非专业人员都是非常有益的,本文仅仅是抛砖引玉,希望大家都能来讨论 PCTOOLS 的使用。

程序常驻内存(TSR)的实现方法

北京工业大学计算机科学系 8771 班 李华

常驻内存(Terminate and Stay Resident),以下简称 TSR,意即程序虽然结束,但其内容还保留在 RAM 中。当重新按下触发键时,即可马上启动程序,无需从磁盘再把程序装到 RAM。Borland 公司的软件产品 SK (Sidekick),就是一个极为有名的 TSR 实用程序。本文是以 Turbo C 为背景,简单介绍如何实现 TSR 的设计。

大部分 TSR 都透过某一个中断启动。常见的中断有时钟、键盘及打印屏幕(prtsc)。Intel 8086 系列微处理器可提供 256 个向量驱动中断,中断服务程序 ISR 从中断向量表所指定的地址开始执行。向量表中的 256 个条目,每个四字节长,其中的地址指向处理特定中断类型的中断程序。向量表从 0000:0000 开始,中断 0 的服务程序地址在 0000:0000~0000:0003,中断 1 在 0000:0004~0000:0007,...依次类推。

TSR 程序设计包括两部分。第一部分用来启动,然后透过 TSR 的 DOS 功能呼叫跳出。此过程必须包括把 TSR 程序进入点的地址,置于向量表的正确位置内。第二部分是实际应用的浮现。在大多数软件产品中,这一部分采用视窗技术来进行操作应用。

TSR 启动部分:这里我们采用中断 9,即按键中断来启动。第一步把目前向量表中中断 9 的地址移往 DOS 所不用的一个中断,在此我们使用中断 60,第二步把 TSR 的进入点地址置入向量表中中断 9 的位置。当启动时,首先透过中断呼叫 TSR,然后检查读入字符是否为热键。如果是则执行 TSR 应用部分,否则关闭 TSR。使用按键中断的优点为(1)没有损失已有的功能。(2)允许多个 TSR 应用同时存在,由不同的热键启动。

在给出正式的程序前,我们要了解 BIOS 输入按键的方法。标准 DOS 设有 15 个字符长的缓冲区,以容纳从键盘敲入的字符。这种方式许可“打在前头”(type-ahead)等待处理。每当按下一键时,即产生中断 9。按键输入的中断服务程序 ISR 从键盘读入字符并置于缓冲区内,当呼叫 BIOS 的键盘字符输入功能时,只需查看缓冲区的内容。按键缓冲区位于 0000:041E(十进制 1054);每一个按键产生 16 位的扫描码,15 个字符加上在末尾被自动加上的回车键扫描码,共 32 字节。

缓冲区如同一个环形队列,由一个首指针与一个尾指针控制其读写。首指针指向最近一次敲入的字符,尾指针指向将为 DOS 或 BIOS 的请求而送回的下一个字符。首指针存于 0000:041A(十进制 1050)位置。尾

指针存于 0000:041C(十进制 1052)。当队列为空时,首尾指针之值相等。了解了这些内容,对于下面应用部分中判断是否为热键的程序就容易看明白了。

启动部分程序如下。此程序以 F2,F3 为热键,F2 启动 notepad(记事卡),F3 启动视窗 window,在此对这两个过程的完整程序就不写出了,有兴趣的同学可以自己添加完全。

set __vid__ mem()为屏幕模式函数,使公用指针 vid__mem 指向屏幕 RAM 的地址,以便视窗程序块中读写的应用。

tsr()过程调用中断 21H 使程序留驻内存。参数 size 用来告诉 DOS 保留多少空间给 TSR 程序,其单位为段,每一段为 16 字节。一般把程序的 .EXE 文件大小除以 16,再乘以 2,作为 size 的大小将是安全的。

main()执行后,程序仍留驻 RAM,可“随时待命”。

```
/* Terminate and stay resident using  
keyboard interrupt */
```

```
#include "dos.h"  
#include "stdlib.h"  
#include "ctype.h"
```

```
void interrupt tsr__ap();
```

```
char far *vid__mem;
```

```
/* busy is set to 1 when program is active,  
0 otherwise */  
char busy=0;
```

```
main()
```

```
{ struct address  
{ char far *p;  
} temp;  
/* pointer to interrupt 9's address */  
struct address far *addr=  
(struct address far *) 36;  
/* pointer to interrupt 60's address */  
struct address far *int9=  
(struct address far *) 240;
```

```
/* Move the keyboard interrupt routine's address
```



```

        to int 60. If int 60 and int 61 contain the same
        address, then the TSR program has not been
        installed */
    if (int9->p==(int9+1)->p)
    { int9->p=addr->p;
    addr->p=(char far *) tsr__ap;
    printf("tsr installed -f2 for notepad,");
    printf("f3 for window");
    }
    else { printf("tsr already initialized\n");
    exit(1);
    }

    set__vid__mem();
    tsr(1600);
}

set__vid__mem()
{ int vmode;

    vmode=video__mode();
    if ((vmode!=2) && (vmode!=3) && (vmode!=
7))
    { printf("video must be in 80 column text mode");
    exit(1);
    }

    /* set proper address of video ram */
    if (vmode==7)
        vid__mem=(char far *) 0xb0000000;
    else
        vid__mem=(char far *) 0xb8000000;
}

/* terminate but keep resident */
tsr(size)
    unsigned size;
    {union REGS r;
    r.h.ah=49; /* terminate and stay resident */
    r.h.al=0; /* return code */
    r.x.dx=size; /* size of program/16 */
    int86(0x21,&r,&r);
    }

/* return the current video mode */
video__mode()
{ union REGS r;
    r.h.ah=15; /* get the video mode */
    return int86(0x10,&r,&r) & 255;
}

TSR 应用部分:

/* this is the entry point into the TSR application code */
void interrupt tsr__ap()
{ char far * t=(char far *) 1050; /* address of head pointer

```

```

*/
    geninterrupt(60); /* read the character */
    if (*t!=*(t+2)) /* if not empty */
    { t+=*t-30+5; /* advance to the character position */
    if (*t==60 || *t==61)
    { bioskey(0); /* clear the f2/f3 key */
    if (! busy)
    { busy=! busy;
    application(*t);

    busy=! busy;
    }
    }
}

application(w)
int w;
{ switch(w)
    { case 60: notepad();
    break;
    case 61: window();
    break;
    }
}

```

interrupt 为 Turbo C 的保留字, 为中断函数定义符, 允许把 C 函数当作 ISR 使用。函数 geninterrupt() 也是 Turbo C 具有的。它是以要产生的中断编号来呼叫的。中断 60 的功能为仅由标准输入读取按键。(main() 中已将中断 9 的地址放在中断 60 的地址上)。从中断 60 返回后, 检查是否首指针所指内容为热键(F2/F3), 二者位置码为 G0 及 G1。若按下二热键, 则必须把它读进来, 以免被应用读到。公用参数 busy 用来阻止两个应用同时启动。应用过程 application() 中的 notepad() 与 window() 函数, 在此就不写出了。

通过以上介绍, 读者可对常驻内存程序的实现方法有一个大概了解。但由于 TSR 程序要破坏原系统正常的 DOS, 所以使用与设计时, 要特别小心。

(上接第 20 页)

功能: 清除“后台”为第一页的图形。

② POKE 230,64; CALL—3086;RETURN

功能: 清除“后台”为第二页的图形。

③ POKE—16299,0; POKE 230,32;RETURN

功能: 第二页做“前台显示”, 第一页做“后台绘图”。

④ POKE—16300,0; POKE 230,64;RETURN

功能: 第一页做“前台显示”, 第二页做“后台绘图”。

防止非法使用 dBASE 程序的一种方法

贵州水城钢铁公司 机修厂 黄世刚

程序员通常在 dBASE III 程序中设置口令来防止非法人员启动程序。但非法人员只要查看源程序或者修改口令部分就能运行保密程序。我厂的标准化管理系统采用了一块引导软磁盘(保密盘),使得无此软磁盘的人员无法运行程序。全部程序在 IBM PC/XT 型机上通过。下面将保密方法简单介绍一下。

1. 首先建立一个引导盘,该盘仅供允许人员使用,在引导盘上建立一个引导文件:

文件名 BZH. BAT

ECHO OFF

A:

COPY A: BZH. PRG C: BZH. PRG

C:

ECHO ON

DBASE BZH

A:

ECHO OFF

DEL C: BZH. PRG

ECHO ON

2. 在该引导盘上建立一个 dBASE III 程序的引导文件,这个引导文件也是主程序:

文件名 BZH. PRG

SET TALK OFF

CLOSE DATA

CLEAR

? " "

? "机修厂标准化管理系统"

? "1. 标准资料数据库"

? "2. 厂标准体系表"

? "3. 标准化工作文件"

? "4. 标准化管理机构"

? "5. 退出"

WAIT "请选择:" TO AC

DO CASE

CASE AC="1"

DO BZZL

CASE AC="2"

:

RETU

ENDCASE

ENDDO

3. 每次使用时在 DOS 状态下插入引导盘,打入 BZH 后 DOS 就将主程序拷贝到 C 盘,并自动调用 C

盘上的 dBASE III 及主程序 BZH. PRG,程序运行结束后引导程序自动删除 C 盘上的主程序。这样无引导盘的非法人员就无法运行 C 盘上的 dBASE III 程序了。

为了进一步提高程序的保密度,还可以将引导盘上的 BZH. BAT 和 BZH. PRG 隐藏起来。

也谈 IBM 机图形存取

上海工程技术大学 焦小友

本刊 89 年第 6 期刊登的“IBM 微机图形的另一种存取方法”中,作者采用的方法是将屏幕上图形信息经“GET”存入数组,再打开一个文件,将该数组读入文件。经本人多次试用,发觉有三个缺点。

一、调用文件后,图形并不能直接“PUT”在屏幕上,必须在调用前“GET”划取一块与存图时同样大小的数组。通常程序编制者并不能记住过去某个时候曾用过多大的数组,更不用说在屏幕上精确地划取了。若每次都记录图形文件“GET”了多大尺寸及范围,既烦琐又不可靠。

二、存取速度太慢。笔者作了一个测试,结果如下:(在 PC/XT 机上,用该文的范例程序)

整形数组大小	存入软盘时间
3000	26.929 秒
5000	44.99 秒
8002(全屏)	1 分 17.78 秒

由上表可见,存不到半屏图形将近半分钟,存全屏图形竟达 1 分 17 秒多,这段时间计算机可以从一累加到溢出!

三、文件所占的字节数太大,经测试如下:

整形数组大小	图形文件大小(字节)
3000	9001
5000	15001
8002(全屏)	24007

一幅全屏图形比用常规“BSAVE”存入的要大 8K!

综上所述,这种图形存取方法至少在 BASIC 领域内是一种不宜采用的方法。

下面向 BASIC 用户介绍一种方法,它既避免了上述缺点,又比用“BSAVE”灵活,笔者开发的一个绘图软件《ORIONARTS》均采用此方法存取图形。范例如下:(用 QUICK BASIC)

```
SCREEN 2: LINE (10,10)-(100,100),,B '随意作一图形
DIM P%(16004) '定义数组
GET (0,0)-(639,199),P% '划取全屏
DEF SEG = VARSEG(P%(0)) '定义段
```

BSAVE "pic",VARPTR(P%(0)),16004 '存全屏

DEF SEG

这段程序把全屏存盘,各测试数据如下:

整形数组大小	存盘时间	文件大小
16004(全屏)	8.23 秒	16019 字节

全屏存盘仅用 8 秒,文件所占字节比"BSAVE"存屏(16K)还少。以下是调图的例子:

DIM L%(16019) ;数组大小即为文件大小

DEF SEG = VARSEG(L%(0))

BLOAD "pic",VARPTR(L%(0))

DEF SEG

PUT (0,0),L%

调图时,须看一下图形文件的字节数,并据此定义数组的大小。调出后随时可"PUT"在屏幕上,由于是"PUT"到屏幕上的,所以也消除了以往使用常规调图时由于奇偶线原因造成的"二次成像"缺陷。

如果使用的是 BASICA 或 GWBASIC,只须把上述二程序加上行号并删去 DEF SEG = VARSEG(数组(0))"这句即可。

(上接第 8 页)

```

79:go 1
80:do while . not. eof()
81:fn=trim(FIELD__NAME)
82:fw=FIELD__LEN-2
83:sele 4
84:go 1
85:repl &fn with substr(s1,1,fw)+"+"
86:go 2
87:repl &fn with substr(s1,1,fw)+"+"
88:go 3
89:repl &fn with substr(s1,1,fw)+"+"
90:sele 6
91:skip
92:enddo
93:use
94:sele 1
95:use &na
96:set relation to recno() into b
97:set safe on
98:return

.use c:stru3
.list to print

```

记录号	FIELD	NAME	FIELD	TYPE	FIELD	LEN	FIELD	DEC
1	分类号	C	20			0		
2	书名	C	34			0		
3	作译者	C	10			0		
4	出版社	C	12			0		
5	出版时间	C	8			0		
6	版次	C	4			0		
7	检索号	C	6			0		
8	总册数	C	6			0		
9	现存数	C	6			0		
10	页数	C	6			0		
11	单价	C	6			2		

图书库全部记录

第 2 页

1990.12.5

分 类 号	书 名	作 译 者
73.87053 2589	计算机应用教学大纲	中华教育部
73.87054 2308	电子计算机讲义	王玉龙
73.87055 2819	计算机通考试题	日本研究所
73.87051 2650	电子计算机实验	白中英

北京安华科技市场西城经营部

——主要经营:电子计算机、电子元器件

各种微机:

- 中华学习机: CEC-1 ¥900.00 小蜜蜂-I ¥920.00
- 家庭 PC 机: ¥3150.00

主频:4.77/12 兆 RAM:640KB 软驱:360KB×2,101 键盘
14"双频单显
Smper286:¥6950.00
主频:10/21 兆 RAM:1MB 可扩 5MB 5¼软驱:1.2MB. 360KB 各 1

硬盘 3.5"40MB. 101 键盘 14"双频单显 数显机箱
选配:14"CGA 彩显加 1500 元 14"1024×768 VGA 彩显加 3800 元

各种:APPLE II、CEC-I、PC 机接口板;九针、24 针打印 机

还承接各种微机的修理业务

对教育部门、个人购机实行优惠

地址:北京西四北大街 120 号 电话:65.6342 邮码:100034

联系人:孙义强 宋函华 BB 机:126 呼 45198 42616

侦错核对程序

西安供电局 张永辉

在港刊杂志上常见有程序中附有侦错核对码,对用户的键入程序是很有帮助的,其形式是每一程序行或监控下每显示一行数据都显示一个核对码,如键入此程序,比较一下核对码便可得知有无错处。笔者试着编写了一个较为简单的程序 EXBM(见程序二),对监控还有所加强。

EXBM 运行后程序放在辅存,CPU 的控制权也在辅存。为区分是否在 EXBM 控制下,BASIC 的提示符改为“>”,监控的提示符为“?”。如果要退出 EXBM 状态,可按 CTRL—RESET;在辅存没被破坏的前提下键入 POKE49283,0 或监控状态下键入 C083 可再行进入。有两点需要说明:一是因 CEC 汉字系统自动置主存方式,禁止辅存,所以退出 EXBM;二是起动 EXBM 前系统内不能有使用辅存的程序。

在 BASIC 下,输入程序的常见错误有多字、少字、字符输入错误,其它还有夹带看不见的控制字符、行号错误等。当出现字符输入错误时,计算出的核对码和正常的不同,但不影响下一行的核对码,如出现多字少字现象,不但影响该行的核对码,还使下一行的核对码出现错误。当误键入了控制字符时,除光标控制符外均发出响声,如同按了 CTRL-G。LIST 时同样发出响声。使用者应掌握这些规律,以便迅速查出错误所在。

使用 EXBM 时先键入一行程序,回车后将显示一个核对码,与正确的核对码比较,无误后可继续键入,否则改正错误。为方便查看,LIST 后程序左起上下对齐,空出核对码和行号。核对码是两位十六进制数,在每个程序行的行首,用一个底横线分隔行号(见程序一)。BASIC 语句的显示方式和通常不一样,在 EXBM 状态下如果保留字在行尾,有可能保留字的后几个字符显示在下一行,而通常是不能分开显示的。EXBM 运行后一些 BASIC 语句将被禁止使用,特别是 RUN、GOTO、CONT 等命令的使用有可能破坏系统,最好不要使用这些命令。EXBM 只是为了侦错除错。DOS 的一般读写命令是允许执行的。

显示“?”符是进入监控的标志,在 EXBM 状态下可使用小汇编,对监控也无不良影响,控制符的处理同 BASIC 状态。在使用数据(块)显示指令时,每八个(或八个以下,视起始地址而言)数据有一个核对码,在数据输入时最好也采用显示时的统一地址,以便得到正确的核对码。因核对码的计算和数据的地址有关系,如果把数据输入到其它地址,即便是数据无误也得不到正确的核对码。数据输入完一行并回车后,将显示一个核对码,以供比较。在显示核对码的同时还将显示下一个要输入数据的地址,方便于数据块的输入。另外在数据块的显示中除按 CTRL-S 可暂停外,还可用空格键

暂停,按 EXC 退出或按任一键继续显示。

EXBM 占用零页的 \$ 19——1F 单元,此时用户不得不得使用这些单元。

一行程序或数据按其不同排列关系、不同大小而得出的核对码是大大超过 256 种状态的,就是说一个核对码可为多个程序行得出,这也是 EXBM 的不足之处。它是针对常见的、一行中只有少数错误,它的盲区是一行中出现多处错误,并且这些错误对核对码的影响相互抵消为零。不过这是极个别的情况。

程序一

```
19__10 HOME
00-20 HTAB (10);VTAB (7);PRINT "EXBM1.0 SYSTEM"
AC__30 END
```

6000-	A2 00 8D 81 C0 8D 81 C0	\$ 45
6008-	BD 00 D0 9D 00 D0 E8 D0	\$ A5
6010-	F7 EE 0A 60 EE 0D 60 D0	\$ 04
6018-	EF 8D 83 C0 8D 83 C0 A9	\$ 8F
6020-	BE 8D 40 D4 A9 13 8D 0D	\$ 2A
6028-	D5 A9 D9 8D 0E D5 A9 5C	\$ 85
6030-	8D FA D6 A9 D9 8D FB D6	\$ 5F
6038-	A2 FF E8 BD 00 61 9D 12	\$ 67
6040-	D9 E0 F5 F0 0C BD A3 60	\$ B1
6048-	E0 5D B0 EE 9D 66 D7 90	\$ 0D
6050-	E9 A9 66 8D 03 D7 8D 53	\$ A2
6058-	D7 8D 5B D7 8D 60 D7 A9	\$ BC
6060-	D7 8D 54 D7 8D 04 D7 8D	\$ 60
6068-	5C D7 8D 61 D7 A9 51 8D	\$ B5
6070-	08 D7 A9 60 8D DB FE 2E	\$ 61
6078-	DB FB A9 20 8D 69 FF A9	\$ 82
6080-	78 8D 8C FD A9 D9 8D 8D	\$ 94
6088-	FD A9 EA 8D 6C FF A9 80	\$ 30
6090-	8D 6A FF A9 D9 8D 6B FF	\$ 64
6098-	A9 BE 8D BE FD A9 D9 8D	\$ B0
60A0-	BF FD 60 A6 24 D0 0B C6	\$ FE
60A8-	25 48 20 8E FD 68 A6 1A	\$ 15
60B0-	86 24 E4 1A D0 04 C9 20	\$ ED
60B8-	F0 03 20 5C DB 60 AD 00	\$ 4E
60C0-	C0 C9 A0 D0 18 2C 10 C0	\$ 62
60C8-	AD 00 C0 10 FB C9 9B D0	\$ 86
60D0-	09 A9 87 20 ED FD 68 68	\$ 22
60D8-	68 68 8D 10 C0 60 B1 1C	\$ 8A
60E0-	29 0F A8 B9 F5 D9 48 65	\$ 82
60E8-	1C 29 0F A8 68 18 79 F5	\$ 75
60F0-	D9 A0 00 18 71 1C 45 1C	\$ 4F
60F8-	45 1D 18 65 1F 85 1F 60	\$ 8B
6100-	60 20 1C D9 20 4B D9 4C	\$ C4
6108-	3C D4 A5 9B 85 1C A5 9C	\$ 9A

6110- 85 1D 20 3D D9 20 3D D9	\$ 58
6118- A0 00 84 1F 20 39 D9 20	\$ FE
6120- 39 D9 B1 1C D0 F9 60 18	\$ 33
6128- 20 A1 D7 E6 1C D0 02 E6	\$ 26
6130- 1D 60 20 9C FC A9 1D 85	\$ 2C
6138- 24 20 48 F9 A9 A4 20 5C	\$ 79
6140- DB A5 1F 20 DA FD 20 9C	\$ F6
6148- FC 60 48 8A 48 20 1C D9	\$ 21
6150- A5 1F 20 DA FD A9 DF 20	\$ F6
6158- 5C DB 68 AA 68 20 24 ED	\$ 52
6160- A4 24 C8 84 1A 60 A5 24	\$ 8C
6168- 85 1E 20 9C FC 60 A5 31	\$ A5
6170- C9 BA D0 33 A5 3C 85 1C	\$ 4A
6178- A5 3D 85 1D A9 A0 20 ED	\$ C4
6180- FD A5 41 20 DA FD A5 40	\$ 18
6188- 20 DA FD A9 BA 20 ED FD	\$ 14
6190- 20 20 FC A5 1E 85 24 A0	\$ BB
6198- 00 84 1F 20 39 D9 A5 1C	\$ 9E
61A0- C5 40 D0 F7 20 44 D9 A9	\$ 1C
61A8- BF 85 33 60 20 DA FD A5	\$ D4
61B0- 3C 85 1C C5 40 D0 08 A5	\$ E2
61B8- 3D C5 41 D0 02 84 1F A5	\$ E0
61C0- 3D 85 1D 20 39 D9 A5 3C	\$ 62
61C8- C5 3E D0 06 A5 3D C5 3F	\$ 26
61D0- F0 06 A5 1C 29 07 D0 0A	\$ 3A
61D8- 20 44 D9 A0 00 84 1F 20	\$ C1
61E0- 81 D7 60 1A F9 48 C6 14	\$ 52
61E8- 99 5A CC 09 EE 71 39 B9	\$ 5B
61F0- 30 2E 96 B1 2E B0 60	\$ DF

让 CEC—BASIC 也 有自动行号功能

云南地矿局物质供应处 沈东山

中华学习机的 BASIC 没有自动行号的功能。《苹果机工具箱》一书中介绍了一个自动行号程序 AUTO LINE。该程序从 100 开始产生行号。如在 BASIC 程序区中已有一个 BASIC 程序,并且有大于 100 的行号时,就必须用 POKE 命令在指定内存单元预置行号,否则就有可能使内存中原有的 BASIC 程序被破坏,使用起来不太方便。

本人经过对 AUTO LINE 进行分析,设计了一个 AUTO 程序,该程序占用 \$ 9000—\$ 9161 内存单元,具有自动寻找程序尾行,并将尾行号取整为整 10 后给出行号,解决了 AUTO LINE 需人工预置行号的不便之处,在内存中没有 BASIC 程序时从 10 给出行号。

使用方法:在监控下将程序输入内存,然后用 BSAVE AUTO,A \$ 9000,L \$ 161 命令存盘备用。使用时只需 BRUN AUTO 命令便可实现自动行号功能。按

F5 键或 Ctrl—F 键便可退出自动行号功能。再次进入时可用 CALL 36864 调用。在运行中如果不想正在输入的当前行,可按 Ctrl—X 键即可清除当前行,重新进行输入。如果将 \$ 90F4 单元的内容改为 \$ 0B,可在行号后自动生成 DATA 语句。这样可以很方便地在程序尾部加入程序所需要的数据。

9000- A9 90 85 74 A9 00 8D 60
9008- 91 8D 61 91 A5 67 85 69
9010- A5 68 85 6A A0 00 B1 69
9018- D0 05 C8 B1 69 F0 1C A0
9020- 02 B1 69 8D 60 91 C8 B1
9028- 69 8D 61 91 A0 00 B1 69
9030- AA C8 B1 69 86 69 85 6A
9038- 4C 14 90 20 3F 90 60 A9
9040- 00 8D 4E 91 8D 4F 91 8D
9048- 50 91 8D 51 91 8D 52 91
9050- A9 F0 8D 53 91 A9 14 8D
9058- 54 91 A9 11 8D 55 91 A9
9060- 24 8D 56 91 A9 11 8D 57
9068- 91 A9 F0 8D 58 91 AC 61
9070- 91 D0 15 4C 79 90 88 D0
9078- 0F AD 60 91 F0 35 C9 0A
9080- B0 1E D0 2F 4C E0 90 60
9088- A2 04 A9 06 20 C7 90 A9
9090- 05 A2 03 20 C7 90 A9 02
9098- A2 02 20 C7 90 4C 76 90
90A0- A2 03 A9 01 20 C7 90 AD
90A8- 60 91 38 E9 0A 8D 60 91
90B0- 4C 79 90 A2 04 20 C7 90
90B8- A9 00 8D 52 91 A2 03 A9
90C0- 01 20 C7 90 4C 84 90 18
90C8- 7D 4E 91 9D 4E 91 C9 0A
90D0- 10 01 60 E9 0A 8D 4E 91
90D8- 18 CA A9 01 4C C8 90 00
90E0- 68 AA 68 48 A9 F1 85 38
90E8- A9 90 85 39 20 EA 03 68
90F0- 60 48 8A C9 06 10 0A 68
90F8- 91 28 BD 4E 91 18 69 B0
9100- 60 68 2C 00 C0 10 FB 91
9108- 28 AD 90 C0 2C 10 C0 48
9110- C9 86 D0 11 68 A9 98 48
9118- A9 1B 85 38 A9 FD 85 39
9120- 20 EA 03 68 60 C9 8D F0
9128- 02 68 60 8A 48 A2 03 FE
9130- 4E 91 BD 4E 91 C9 0A 30
9138- 08 A9 00 9D 4E 91 CA 10
9140- EE 90 05 A9 01 9D 4E 91
9148- 68 AA 68 60

图形屏幕坐标的 对应存储单元

葛海东

在 APPLE II 和 CEC—I 的高分辨率图形模式下,直接用机器语言中的送存指令作图,具有速度快的优点。但是图形点地址与我们所熟悉的直角坐标并不一致。本文提供一个能把直角坐标转换为图形点地址的程序。

执行前,要把纵坐标放入 \$1E,把横坐标放入 \$1B,若横坐标超过了 \$FF(255)那就要置 \$1A 为 \$01,反之置为 \$00。在图形第一页时置 \$303 为 \$08,第二页时则置为 \$10。程序执行完后,\$FE、\$FF 中已按低在前高在后的顺序放入了图形点所在单元的地址。这样便于用间接变址指令如:STA(\$FE),Y 等来存取。\$CF 单元的数表示此点在存储单元的七个连续点中的位置。例如:若 \$CF 中的值为 \$03,那么它就位于七个点中的 d₃ 位(最小位元为 d₀),右移或循环右移三次就可用来改变此点的颜色,从而达到画点的目的。

程序中使用了 \$FF、\$FE、\$CF、\$E8、\$1B、\$1E、\$1A 七个零页单元,并改变了寄存器 A 的内容,其它均不受影响。由于采用相对寻址,因而可以把它放到内存的任何位置执行。

```
300- 84 18 A9 10 85 FF A5 1E
308- A0 A0 0A B0 08 A0 50 0A
310- B0 04 A0 00 4A 0A 0A 84
318- FE 26 FF 0A 26 FF 0A 66
320- FE 4A 4A 4A 65 FF 85 FF
328- A4 1A A5 1B 38 85 CF E9
330- 07 E6 FE B0 F8 88 F0 F4
338- C6 FE A4 18 60 00 00 00
```

神奇的 \$C0B0 单元

福建 南平师范学校 李锋

CEC—I 的新用户时常对字库的自检感到疑惑不解,机器怎么一而再,再而三地出现同一批地址 \$4000~\$7FFF。原来,汉字的字库容量为 256K,由于 CEC—I 的中央处理器 6502 仅有 64K 的寻址能力,因此在汉字处理电路中,采取了存储器切换技术。存储体的切换,是通过 \$C0B0 单元置值来实现。单元 \$C0B0 中各位的功能情况如下:

D₇D₆D₅D₄D₃D₂D₁D₀ 功 能

××××0000	选择空间 \$4000~\$7FFF 第 1 体
××××0001	选择空间 \$4000~\$7FFF 第 2 体
××××0010	选择空间 \$4000~\$7FFF 第 3 体
××××0011	选择空间 \$4000~\$7FFF 第 4 体
××××0100	选择空间 \$4000~\$7FFF 第 5 体
××××0101	选择空间 \$4000~\$7FFF 第 6 体
××××0110	选择空间 \$4000~\$7FFF 第 7 体
××××0111	选择空间 \$4000~\$7FFF 第 8 体
××××1000	选择空间 \$4000~\$7FFF 第 9 体
××××1001	选择空间 \$4000~\$7FFF 第 10 体
××××1010	选择空间 \$4000~\$7FFF 第 11 体
××××1011	选择空间 \$4000~\$7FFF 第 12 体
××××1100	选择空间 \$4000~\$7FFF 第 13 体
××××1101	选择空间 \$4000~\$7FFF 第 14 体
××××1110	选择空间 \$4000~\$7FFF 第 15 体
××××1111	选择空间 \$4000~\$7FFF 第 16 体
×××0××××	禁止彩色显示
×××1××××	允许彩色显示
××0×××××	选择空间 \$8000~\$BFFF 第 1 体
××1×××××	选择空间 \$8000~\$BFFF 第 2 体

地址空间 \$4000~\$7FFF 的每一体的容量为 16K 字节,汉字库共 16 体,因此汉字库的容量为 16×16=256K。自检时,是一体一体地自检,故相应地址重复 16 次,约需一分钟!

怎样解读 CEC—I 学习机的汉字系统

刘文勤

CEC—I 型中华学习机有较完善的汉字处理能力,许多拥有中华学习机的爱好者在二次开发应用中都希望详细了解该机汉字系统对汉字和 ASCII 字符的具体处理过程。但由于汉字系统固化在辅存中,要想研究它也要费点脑筋。笔者为此编了一个小程序,可以将辅存中 D000~FFFF 之间的 12K 字节汉字处理程序传送到内存 1000~3FFF 区间,然后返回监控系统。现在您可以对汉字系统的任意段落进行反汇编,仔细分析研究了,不过要记住,这段程序现在在内存中的高位地址,要加上 \$C0 才是该机汉字系统的真实地址。

```
0800- A9 00 AA A8 85 06 85 08
0808- A9 D0 85 07 A9 10 55 09
0810- C0 8D 09 C0 B1 06
0818- 91 08 C8 D0 F9 E6 07 E6
0820- 09 E8 E0 30 D0 F0 8D 08
0828- C0 8D 82 C0 4C 69 FF
```


用机器语言解八皇后问题及其推广

广西柳州 龙城中学 卢祖兴 刘超宏

八皇后问题是指:在 8×8 的方棋盘上放置 8 个国际象棋的皇后,使之彼此不同行,不同列,不同对角线,以免互相吃掉。这是高斯在 1850 年首先提出来的数学趣题。由于计算量很大,以致这位著名数学家在有生之年也未能找出问题的全部解答。

自从计算机出现之后,八皇后问题自然也成了计算机爱好者讨论的热门话题。由于通常都是在 8 位机上用 BASIC 语言来求解,仍然费时较多。由于算法不同,有的竟长达 8、9 小时,快的也要近 30 分钟,若用汇编语言来解,就能极大地节省时间。

我们用 6502 机器语言编写的程序一,就能在 22 秒钟给出八皇后问题的 92 组解和解法布局图。方法是,在监控状态下,准确地输入程序一,然后 300 G 或在 BASIC 状态下,CALL 768 即可。若事先接通打印机,还可把解法及布局图印在纸上。(0 表示空白,* 表示皇后)。为推广到 N 皇后的情形,输入程序二,就可以很方便地设定 N 值($3 < N \leq 16$),用以解 N 皇后问题。

当 N 值较大时,可打印解法,而省略布局图,以节省时间和打印纸。在监控下输入程序一后,运行程序二,输入 N 值及选择解法(B)或布局图(F)即得出结果。

程序一

```
0300- A9 01 85 E0 A9 00 85 09
0308- 85 0A 85 18 E6 18 A6 18
0310- A9 01 95 E0 A6 18 CA 86
0318- 19 A6 19 B5 E0 85 1A A6
0320- 18 B5 E0 38 E5 1A F0 5D
0328- 10 04 49 FF 69 01 85 1A
0330- 8A 38 E5 19 C5 1A F0 4F
0338- C6 19 10 DD A5 18 C9 07
0340- 90 CA 85 08 E6 09 D0 02
0348- E6 0A A0 08 24 0B D0 18
0350- A6 08 98 D5 E0 08 A9 AA
0358- 28 F0 02 A9 B0 20 ED FD
0360- A9 A0 20 ED FD 88 D0 E8
0368- A6 08 B5 E0 20 E3 FD 24
0370- 0B F0 05 20 48 F9 F0 03
0378- 20 8E FD C6 08 10 CB 20
0380- 8E FD 20 8E FD A6 18 B5
0388- E0 C9 08 B0 04 F6 E0 D0
0390- 83 C6 18 10 F0 60
```

程序二

```
5 POKE 11,0
10 INPUT "N=" ;N; POKE 831,N-1; POKE 843,N; POKE
```

906,N

```
15 INPUT "Full of Brief? (F/B)"; A$
17 IF A$ = "B" THEN POKE 11,1
18 CALL 768
20 PRINT "TOTAL="; PEEK (10) * 256 + PEEK (9)
30 POKE 831,7;POKE 843,8;POKE 906,8
]RUN
N = 8
```

Full or Brief? (F/B)F

```
0 0 0 0 * 0 0 0 4
0 0 0 0 0 0 * 0 2
0 * 0 0 0 0 0 0 7
0 0 0 0 0 * 0 0 3
0 0 * 0 0 0 0 0 6
* 0 0 0 0 0 0 0 8
0 0 0 * 0 0 0 0 5
0 0 0 0 0 0 0 * 1
```

```
0 0 0 * 0 0 0 0 5
0 0 0 0 0 0 * 0 2
0 0 0 0 * 0 0 0 4
0 * 0 0 0 0 0 0 7
0 0 0 0 0 * 0 0 3
* 0 0 0 0 0 0 0 8
0 0 * 0 0 0 0 0 6
0 0 0 0 0 0 0 * 1
```

]RUN

N = 8

Full of Brief? (F/B) B

```
4 2 7 3 6 8 5 1
5 2 4 7 3 8 6 1
3 5 2 8 6 4 7 1
```

也谈汉字平移显示

四川铜梁县中学 李茂学

贵刊九〇年第八期上介绍汪五一同志的“CAI 课件制作辅助工具——文字平移显示”,对汉字的平移作了较清楚的论述,读了深受启发。本人根据该文提供的思想方法,在中华学习机上重新设计了文字平移显示制作程序。本文的 BASIC 和机器语言程序都大大短于汪文的程序,操作简单,使用方便,在此呈献给大家供参考。下面介绍使用方法:

1. 输入机器语言程序 2 和程序 3,分别以 LMX · SET 和 LMX · PY 为文件名存盘。
2. 运行 BASIC 程序 1。输入平移汉字和行数(指刚输入的汉字显示在屏幕上有几行)。
- 按提示逐次回答后输入存盘(文件名如:PYHZ)
3. 在程序中或键盘上打入如下命令:

BLOAD PYHZ

BRUN LMX · PY

就会看到输入的汉字在指定的行上向左平移显示。

本程序最大限度可移动显示 400 个汉字,也能在 APPLE 机上运行。机器语言程序全是浮动地址,可放在内存中的适当地方。

程序一

```
1 REM 汉字平移点阵信息提取
2 POKE 24578,8; POKE 24580,8; POKE 24579,96; POKE
24581,96
5 PRINT CHR$(4);"BLOAD LMX.SET"
10 INPUT "输入平移汉字:";A$;PRINT A$
15 INPUT "有几行?";B;HGR2
20 VTB=1;PRINT A$
30 FOR I=1 TO B:VTB=9;CALL 8192
40 PRINT CHR$(13)
50 NEXT I
60 INPUT "显示在第几页(1/2)?";P;IF P>2 OR P<1
THEN 60
65 INPUT "在第几行平移(1-12)?";N;INPUT "存入文
件名";M$
70 A=PEEK(24579)*256+PEEK(24578);B=PEEK
(24581)*256+PEEK(24580)
80 L=B-A+1;B=8192*(1+(P=2))
85 N=N-1;A=B+40*INT(N/4)+256*(N-INT
(N/4)*4);B=A+128
90 POKE 24577,INT(A/256);POKE 24576,A-256*
PEEK(24577);POKE 24583,INT(B/256);POKE 24582,B
-256*PEEK(24583)
100 PRINT CHR$(4);"BASVE";M$;"A$ 6000,L";L
```

程序 2

```
2000- AD 04 60 85 FE AD 05 60
2008- 85 FF A0 00 A9 00 85 FA
2010- A9 80 85 FC A9 40 85 FB
2018- 85 FD A2 00 B1 FA 84 06
2020- A0 00 91 FE 18 A9 04 65
2028- FB 85 FB E6 FE D0 02 E6
2030- FF A4 06 E8 E0 08 D0 E4
2038- A2 00 B1 FC 84 06 A0 00
2040- 91 FE 18 A9 04 65 FD 85
2048- FD E6 FE D0 32 E6 FF A4
2050- 06 E8 E0 08 D0 E4 C8 C0
2058- 28 D0 B1 A5 FE 8D 04 60
2060- A5 FF 8D 05 60 60
```

程序 3

```
3000- 2C 10 C0 AD 02 60 85 FE
3008- AD 03 60 85 FF 20 38 30
3010- 18 A5 FE 69 10 85 FE A5
3018- FF 69 00 85 FF AD 00 C0
3020- 10 01 60 A5 FF CD 05 60
3028- 30 E3 CD 05 60 D0 D4 A5
3030- FE CD 04 60 10 CD D0 D5
3038- A0 00 B1 FE 48 98 AA 68
```

```
3040- 95 50 C8 C0 10 D0 F3 A9
3048- 07 85 06 AD 00 60 85 FA
3050- AD 01 60 85 FB AD 06 60
3058- 85 FC AD 07 60 85 FD A2
3060- 00 B5 50 6A 08 95 50 A0
3068- 27 B1 FA 0A 28 6A 6A 08
3070- 0A 38 6A 91 FA 88 10 F1
3078- 28 18 A9 04 65 FB 85 FB
3080- E8 E0 08 D0 DC A2 00 B5
3088- 58 6A 08 95 58 A0 27 B1
3090- FC 0A 28 6A 6A 08 0A 38
3098- 6A 91 FC 88 10 F1 28 18
30A0- A9 04 65 FD 85 FD E8 E0
30A8- 08 D0 DC C6 06 D0 9C 60
```

“前台显示、后台绘图” 的秘密

谢吉华

许多游戏软件运行时,能快速地更换画面且不露出任何绘图过程,令人赞叹不已。初看起来,似乎 CEC—I 机有“前台”、“后台”两部分。那么,“前台显示,后台绘图”的功能是如何实现的呢?

CEC—I 机有两页高分辨图形区,计算机执行 H PLOT 语句时,零页存储单元 \$E6 为高分辨率作图的页面指针,当 \$E6(十进制 230)的值为 \$20(十进制 32)时,H PLOT 语句的执行结果画在高分辨第一页上;\$E6 单元的值为 \$40(十进制 64)时,H PLOT 的执行结果画在高分辨第二页上。因此,我们在观看高分辨第一页图形时,可用命令 POKE230,64 使高分辨第二页图形区作为“后台”,接着运行的绘图程序就会将图形画在第二页上,这样,“前台显示、后台绘图”的功能就实现了。第二页图形画好后,只要用 POKE—16299,0 就可以观看画好的第二页。这幅在“后台”画好的图形就在“前台”上显示出来了,而刚才为“前台”的第一页只要执行 POKE230,32 后又可转为“后台绘图”了。

细心的读者会发现,当第二页图形作为“前台”,而作为“后台”的第一页需要清除画面时该怎么办呢?如果使用 HGR 命令不但会使第一页转到“前台”而且还会让大家看到清除画面的动作,这样“后台绘图”就徒有虚名了。笔者发现 HGR、HGR2 有一段公用程序可供借用,它的入口是 \$F3F2(十进制—3086),所以要清除“后台”为第一页的图形,可执行如下命令:

POKE 230,32;CALL—3086

为了便于大家掌握“前台显示,后台绘图”的技巧,现将几个关键的子程序及功能说明如下:

① POKE 230,32;CALL—3086;RETURN

(下转第 13 页)

演示抛体运动

北方工业大学 白梅兰

应用微机的作图功能模拟物理实验是经济、方便又准确的现代化手段。本文介绍一个有趣的抛体运动问题的微机模拟演示。

1. 问题

设从某一点 O 以同样速率 V_0 沿各个不同方向, 同时抛出 N 个物体(视为质点), 验证:

- (1) 任意时刻 t, 各质点总是落在某一圆周上;
- (2) 园心自由下落, 半径与时间 t 成正比增大。

2. 问题分析(采用国际单位)

设抛射点 O 为坐标原点, 水平向右为 X 轴正方向, 竖直向上为 y 轴正方向, 抛射体的初速率为 V_0 。任意一个质点 $i(i=0, 1, \dots, N-1)$ 的抛射角为 θ_i (抛射初速度与 X 轴正方向的夹角), 又设抛射时刻为计时起点。若不计空气阻力, 质点的运动方程为

$$X = V_0 \cos \theta_i \cdot t \quad (1)$$

$$Y = V_0 \sin \theta_i \cdot t - \frac{1}{2} g t^2 \quad (2)$$

式中 g 为重力加速度, 其值取为 9.8, t 是抛射体运动的时间。为使屏幕显示效果较好, 本程序设计初速度 V_0 为 16。由①②两式消去角度 θ_i 得到

$$X^2 + (Y + \frac{1}{2} g t^2)^2 = (V_0 t)^2 \quad (3)$$

由方程③可以看出:

(1) 在某一时刻, t 值一定, ③式为一园方程, 其园心坐标为 $(0, -\frac{1}{2} g t^2)$, 半径为 $V_0 t$ 。因此, 在任意时刻 t, 从同一点 O, 以同样的初速率 V_0 , 不同的抛射角 θ_i 抛出的 N 个质点总是散落在以 $V_0 t$ 为半径, 以 $(0, -\frac{1}{2} g t^2)$ 为园心的同一园周上。

(2) 园心 $(0, -\frac{1}{2} g t^2)$ 从抛射点 O 自由下落, 园的半径 $V_0 t$ 与时间 t 成正比增大。

以上两点, 通过微机模拟可以得到验证。

3. 模拟程序简介

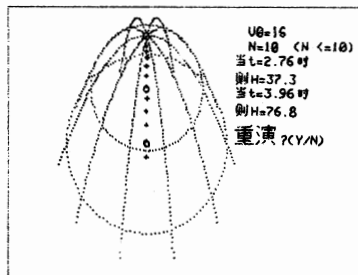
本程序显示的汉字、图形和字符, 都是用造型表建立的。通过 XDRAW M AT X, Y 语句将编号为 M 的图形显示在坐标为 X, Y 的位置。键入程序后, 再在监控状态下键入造型表代码便可运行。

4. 模拟过程说明

程序首先在显示屏上方印出坐标原点和初速。依照提示, 键入抛体个数 N (本程序设计 $N \leq 10$) 再按回车键, 电脑便从 O 处以同样的初速率 V_0 , 不同的抛射角 $\theta_i (\theta_i = \frac{2\pi}{N} i)$, 同时抛出 N 个质点。随后各个质点分别作抛体运动。

在抛体运动过程中, 只要按一下 T 键, 电脑就画园。可以看到, 任一时刻各质点确实处在同一园周上,

画完园后, 电脑又适时地给出抛体运动的时间 t, 园



心下落距离 h 和半径 R 之值。屏幕上每一瞬时显示的小园表示园心的瞬时位置, “+”表示园心运动的等时间隔轨迹, 很容易看出园心是作自由落体运动的。

相继按两次 T 键, 从先后画两次园, 给出两次 t、R 之值, 均能表明半径 R 与时间 t 成正比增大。

演示到最后, 电脑提示“重演? (Y/N)”。若要重演则键入 Y 重新开始; 键入 N 则退出。

```
100 POKE 232,0 :POKE 233,96
300 HGR2 : HCOLOR= 3: ROT = 0: SCALE=-1
310 HPLLOT 0,0 TO 279,0 TO 279,191 TO 0,191 TO 0,0
320 LET W=2*3.14159265:X0=110:Y0=20:G=9.8:
V0=16:C=190:XDRAW 7 AT X0-9,Y0+7
325 HW=-2:HT=C:VT=20:ST$="V0=16":GO-
SUB 10000
330 HT=C:VT=VT+10:ST$="N=? (N<=10)":
GOSUB 10000
335 INPUT N
336 IF N<0 OR N>10 THEN HT=C:ST$="N=? (N
<=10)":GOSUB 10000:VT=VT-10:GOTO 330
338 XDRAW 31 AT C+12,VT
339 HT=C+12:VT=VT:ST$=STR$(N):GOSUB
10000
360 AX=V0*COS(W/N)
370 AY=V0*SIN(W/N)
380 J=.06:FOR K=1 TO 72:T=K*S
400 FOR I=1 TO N
410 X(I)=X0+V0*COS(I*W/N)*T
415 Y(I)=Y0-V0*SIN(I*W/N)*T+4.9*T*T
T
419 NEXT I
420 LET RX=AX*T:RY=AY*T:R=SQR(RX*RX
+RY*RY):H=4.9*T*T
423 IF INT(K/5)=K/5 THEN XDRAW 6 AT X0-2,Y0+
1+G*T*T/2
425 FOR I=1 TO N:HPLLOT X(I),Y(I):NEXT
```



```

427 IF PEEK (- 16384) = 212 THEN POKE - 16368, 0;
GOSUB 510
490 NEXT K
501 VW = 1; HT = C - 10; VT = VT + 20; XDRAW 1 AT
HT, VT; XDRAW 4 AT HT + 17, VT
502 HT = HT + 35
503 ST$ = "? (Y/N)"; GOSUB 10000
504 GET A$
507 IF A$ = "N" THEN TEXT: END
508 IF A$ < > "Y" THEN 504
509 GOTO 300
510 XDRAW 3 AT C - 10, VT + 10; HT = C; VT = VT +
10; ST$ = "@=" + STR$ (T); GOSUB 10000
511 XDRAW 5 AT C + 40, VT + 2
512 DRAW 7 AT X0 - 9, Y0 + 7 + G * T * T / 2

```

```

513 FOR P = 0 TO W STEP 3.1415926 * 3 / 180
514 HPLOT X0 + R * COS (P), Y0 + H - R * SIN (P)
516 NEXT P; HW = - 2
520 XDRAW 2 AT C - 10, VT + 13; HT = C + 2; VT = VT
+ 13; ST$ = "H=" + STR$ (INT (H * 10 + .5) / 10);
GOSUB 10000
530 RETURN
10000 FOR I = 1 TO LEN (ST$)
10020 CH = ASC (MID$ (ST$, I, 1)) - 32
10030 HH = 8 + HW; IF CH = 0 THEN 10060
10050 XDRAW CH AT HT, VT
10060 HT = HT + HH
10070 NEXT I
10090 RETURN

```

```

6000-39 00 74 00 AE 00 CD 00
6018-61 01 61 01 72 02 62 01
6030-D8 01 E7 01 61 01 61 01
6048-61 01 61 01 61 01 61 01
6060-61 01 61 01 4D 02 61 01
6078-2D 2D 2D C4 3B 3F 3F E6
6090-36 66 2D 2D 26 70 2D 25
60A8-40 2D 2D 05 20 00 01 40
60C0-0D 40 03 24 24 44 31 36
60D8-25 C0 23 FC 73 4D 71 40
60F0-24 2D 2D 25 0E 2D 2D 36
6108-B4 32 2D 2D 74 2D 25 96
6120-9D 6F 40 18 08 2D E5 16
6138-00 28 E5 16 04 00 49 49
6150-04 00 49 1C 1C 24 0C 0C
6168-00 01 20 24 64 2D 15 36
6180-22 00 01 40 18 40 03 0C
6198-2D 1E 1E 0E 15 F6 3F 07
61B0-46 2D 2D 03 38 3F 24 2D
61C8-04 00 01 40 18 4D 1B 2D
61E0-40 03 64 2D 15 26 00 01
61F8-C9 07 38 38 60 0C 0C 04
6210-15 F6 BF 16 04 00 49 09
6228-3F 4E 31 26 00 21 24 24
6240-6C 09 36 36 36 DF 40 18
6258-04 00 01 40 18 40 18 36
6270-04 00 49 04 00

```

```

6008-E3 00 19 01 39 01 3E 01
6020-69 01 78 01 82 01 92 01
6038-F7 01 01 02 61 01 09 02
6050-21 02 61 01 61 01 61 01
6068-61 01 61 01 5A 02 61 01
6080-3F 3F 04 28 2D 2D 66 2D
6098-40 18 49 3F 3F 3F 37 1C
60B0-03 24 24 64 2D 15 36 36
60C8-36 36 1E 04 00 29 2D 2D
60E0-03 04 00 05 28 20 40 18
60F8-1F 3F 3F 3F 96 28 2D 74
6110-73 0E DF DB DB 2B 28 20
6128-36 F6 07 20 B0 40 03 96
6140-09 40 18 40 18 40 18 BF
6158-04 00 09 05 28 20 E4 1C
6170-36 1E 3F 04 60 0C 04 00
6188-2D 15 F6 C6 1E 1E 2E 2D
61A0-20 00 49 21 24 24 3C 1E
61B8-2D 04 00 09 2D 05 20 1C
61D0-2D A6 1A 1E 1E 36 04 00
61E8-40 18 20 0C 2D 15 B6 38
6200-00 01 28 2D 25 38 3F 27
6218-B8 3F 20 E4 2D E5 23 04
6230-6C 09 36 36 36 DF 40 18
6248-78 11 0E 04 00 21 24 24
6260-36 0E 15 05 28 20 24 24

```

```

6010-52 01 5A 01 61 01 61 01
6028-A2 01 AE 01 BB 01 CA 01
6040-16 02 61 01 61 01 61 01
6058-61 01 61 01 3D 02 61 01
6070-61 01 69 02 2D 2D 2D 2D
6088-2D 24 24 3F 3F 3F 3F 3F
60A0-3F 3F 3F 44 2D 2D 2D 36
60B8-FE 24 24 96 D2 1E 4D 70
60D0-25 24 3C 3F 3F B7 29 2D
60E8-1C 1C 04 40 18 A8 0E 4E
6100-2D 35 3E 3F 27 1E 3F 27
6118-00 40 18 24 24 2D 36 77
6130-70 F5 07 20 40 03 0E 04
6148-76 73 2D 0E 40 03 24 1C
6160-04 00 C9 0C 0C 0C 0C 04
6178-09 25 24 24 EC 13 8D 92
6190-25 00 01 40 18 40 18 2D
61A8-1E 36 2D 0D 04 00 01 A8
61C0-3F 37 26 40 03 0C 0C 2D
61D8-09 2D 05 20 1C 3F 17 26
61F0-3F 4D F2 1E 3F 04 00 49
6208-00 01 40 18 40 03 0C 2D
6220-00 21 24 24 6C 09 36 3E
6238-78 11 0E 04 00 21 24 24
6250-2C 2D 15 F6 3F 0E 0E 0E
6268-00 49 24 E4 1C 6C 09 F6

```

多单元电视接 收天线设计程序

新疆石河子农垦中专 史文忠

电视天线的种类很多,本程序是多单元定向天线的 VHF 和 UHF 频带的工作设计。(1—60 频道)
定向天线制做简单,增益高,是一种常用天线。此

程序设计的天线可对单一频道接收,也可接收同一方向的多频道电视节目。

使用方法:①当程序运行后,输入所需接收的最低频道值,回车后,再输入所需接收的最高电视频道值,回车后,屏幕立即显出天线的各种数据。如需接收单一频道,两次的输入为同一频道值即可。②数组表中数据不能变更③天线的引向振子长度取相等,引向振子间距也取相等。

10 HOME (CLS)

20 INPUT "接收最低频道值(D)";M

(下转第 24 页)

一个实用的造型表编辑程序

北京一〇一中学 初三(1)班 左震宇

造型表是 APPLE 机的一个良好的绘图工具。但是造型表编制步骤繁琐,虽然已有不少制作造型表的辅助软件,但这些程序总不能使人十分满意。

经过本人多次实践,编制成功了一个实用的造型表编辑程序。本程序制表速度快,精度高,功能齐全,输入方便。程序由输入和扫描两部分组成。程序开始运行,首先输入造型数和存入内存的首址(必须在 \$6000 以上)。接着在屏幕左侧有一个 47×47 的点阵,这是作图区。在右边有一个方框,框内可直接映射出作图区中的图形,以便直观地看到屏幕的效果。在点阵中央有一条短横线做为光标,用 I、J、K、L 键控制光标上下左右移动。对于点的画和擦,本程序采取了两种输入方法:第一种是用 U 键画上或擦去一个点;另一种是用 I、J、K、L 直接擦点或画点,C 键用来切换擦或画两种模式。这两种绘图方式由 P 键切换。为了更好的利用作图区和让作图更准确,程序提供了座标输入。接 T 键后,输入横座标、纵座标,便可将光标移至对应处。若按 Q 键可退出编辑,进入扫描部分。扫描部分就是由用户将作图区中的造型扫描一遍(用 I、J、K、L 键移动光标)。如果想修改造型,返回编辑状态,可按 B 键。如果扫描完毕,想退出编辑状态,可按 Q 键。使用过其它造型表编辑程序的用户,可能有这样的发现,有时一个造型表打入时没有什么地方可挑剔,可打出来与原造型大不相同,这是由于 APPLESOFT 对于 \$01—\$07 的造型编码运行有误。本程序对于这个问题做了完善的处理。使打出的造型完全正确。当你退出编辑状态后,计算机询问你是否要把这幅图拷贝到下一幅图中。若是键入 Y 则在下一个造型编辑时,作图区中仍然保留上一幅图,便于稍作修改,造出相差不多的第二幅图,这样可提高制表效率。若键入 N 则在编辑下一个造型时,先将作图区清除干净。在所有造型都编辑完之后,计算机将询问是否存盘。若键入 Y,则在键入文件名后可将造型表存盘。

```
10 GOTO 80
20 X2=X;Y2=Y;X=X+(A=76 AND X<185)*4-(A=74 AND X>3)*4;Y=Y+(A=75 AND Y<184)*4-(A=73 AND Y>3)*4;RETURN
30 POKE -16304,0;POKE -16297,0;POKE -16302,0;POKE -16299,0;RETURN
40 POKE N2,S(S);N2=N2+1
50 IF S=0 OR S=4 THEN POKE N2,8;N2=N2+1
60 RETURN
70 POKE 232,B1;POKE 233,B2;DRAW I/2 AT X3,Y3;POKE 232,0;POKE 233,3;RETURN
80 FOR I=768 TO 825;READ A;POKE I,A;NEXT;POKE 232,0;POKE 233,0;DATA 2,0,6,0,10,0,62,36,53,0,47,
```

```
0
85 POKE 49238,0
90 DATA 24,165,107,105,9,133,6,133,8,165,108,133,7,144,2,230,7,105,18,133,9,160,0,169,0,145,6,230,6,208,2,230,7,165,6,197,8,208,240,165,7,197,9,208,234,96
100 DIM A%(47,47);SCALE=1;ROT=0
110 FOR I=0 TO 7;READ S(I);NEXT ;DATA 24,8,16,24,28,12,20,28
120 HOME;VTAB 1;INPUT "How many shapes?";N;VTAB 5;INPUT "Start address in RAM?";M
130 N2=M+2+2*N;D=N2-M+1;B2=INT(M/256);B1=M-B2*256;POKE M,N;POKE M+1,0
140 FOR I=2 TO 2*N STEP 2
160 POKE M+I+1,INT((N2-M)/256);POKE M+I,N2-PEEK(M+I+1)*256-M;POKE N2,0;POKE N2+1,0
170 IF CPY=1 THEN GOSUB 30;GOTO 190
180 HGR2;HCOLOR=1;FOR P=0 TO 191 STEP 4;HPLOT 0,P TO 188 ,P;HPLOT P+1,0 TO P+1,188;NEXT ;CALL 780
190 X=95;Y=94;X1=24;Y1=24
200 HCOLOR=3;HPLOT 203,69 TO 255,69 TO 255,119 TO 203,119 TO 203,69
210 HCOLOR=3-A%(X1,Y1);DRAW 2 AT X,Y
220 GET A$
230 HCOLOR=A%(X1,Y1);DRAW 2 AT X,Y;HCOLOR=3
240 A=ASC(A$);GOSUB 20
260 X1=(X+1)/4;Y1=(Y+2)/4
270 IF A=85 THEN A%(X1,Y1)=3-A%(X1,Y1)
280 IF PUT=1 THEN A%(X1,Y1)=C2
290 HCOLOR=A%(X1,Y1);DRAW 1 AT X,Y
300 IF A=81 THEN X3=205+X1;Y3=140+Y1;GOTO 380
310 IF A=80 THEN PUT=ABS(PUT-1)
320 IF A=67 THEN C2=ABS(C2-3)
330 HPLOT 205+X1,70+Y1
340 IF A=84 THEN TEXT;VTAB 20;INPUT "INPUT X,Y";X3,Y3;X1=X3;Y1=Y3;X=X1*4-1;Y=Y3*4-2;GOSUB 30
370 GOTO 210
380 POKE -16368,0;S=0
390 FOR J=0 TO 3 STEP 3
400 POKE N2,0
410 HCOLOR=3-A%(X1,Y1);DRAW 2 AT X,Y
420 GET A$
430 HCOLOR=A%(X1,Y1);DRAW 2 AT X,Y;HCOLOR=3
440 A=ASC(A$);GOSUB 20
460 IF X=X2 AND Y=Y2 AND A<>66 AND A<>81 THEN 410
```

```

470 IF A=81 THEN GOTO 590
480 COLOUR=A%(X1,Y1)*4/3
490 X1=(X+1)/4;Y1=(Y+2)/4
500 S=((Y<Y2)*0+(X>X2)+(Y>Y2)*2+(X<
X2)*3)+CO)*INT(2^J)+S
510 IF A=66 THEN N2=M+D;HCOLOR=0;GOSUB 70;
GOTO 190
520 HCOLOR=3;GOSUB 70
530 NEXT
540 POKE N2,S
550 IF S>7 THEN N2=N2+1;GOTO 380
560 GOSUB 40;GOTO 380

```

(上接第 22 页)

```

30 INPUT "接收最高频道值(G)";N
40 FOR PS = 1 TO 60
50 READ PL
60 IF M = PS THEN PD = PL
70 IF N = PS THEN PG = PL
80 NEXT PS
100 PO = SQR (PD * PG)
110 DE = INT (0.3 * PG + 0.5)
120 DI = INT (0.6 * DE)
130 DR = INT (0.19 * PD - 0.5)
140 IE = INT (0.42 * PG)
150 IR = INT (0.5 * PD + 0.5)
160 EI = INT (0.95 * PG / 2)
170 B = INT (0.003 * PO + 0.5)
180 HOME : PRINT "      ";M;"频道至";N;"频道的天
线数据"
200 PRINT "引向振子的间距      =" ;DE;"cm"
210 PRINT "引向振子与馈电振子间距      =" ;DI;"cm"
220 PRINT "反射振子与馈电振子间距      =" ;DR;"cm"
230 PRINT "引向振子长度      =" ;IE;"cm"
240 PRINT "反射振子长度      =" ;IR;"cm"
250 PRINT "馈电振子长度      =" ;EI;"cm"
260 PRINT "馈电振子宽度      =" ;B;"cm"
270 PRINT "馈电振子开口距      =" ;INT(0.027 *
      PO + 0.5);"cm"

```

```

300 DATA 571,496,438,375,340,175,168,160,154,148,
142,137,63,3,62,2
310 DATA 61,2,60,2,59,3,58,4,57,5,56,6,55,8,55,54,
2,53,4,49,2,48,5
320 DATA 48,47,3,46,7,46,2,45,6,45,1,44,5,44,43,5,
43,42,5,42,41,6
330 DATA 41,1,40,7,40,2,39,8,39,4,39,38,6,38,2,37,
8,37,4,36,66
340 DATA 36,32,35,97,35,63,35,29,34,97,34,64,34,32,
34,01,33,71

```

(上接第 31 页)

```

114      RET
115 HIGH:  MOV R0,#02H
116      RET

```

```

570 POKE N2,S;IF S>7 THEN N2= N2+1;GOTO 590
590 POKE N2,0;N2=N2+1
600 IF I=2*N THEN 630
610 TEXT;VTAB 9;PRINT "Copy this shape to next one?";
GET A$;IF A$="Y" THEN CPY= 1;GOTO 630
620 CPY = 0
630 N1=N1+2;D=N2-M
640 HCOLOR=0;GOSUB 70;NEXT I
650 TEXT;VTAB 12;PRINT "Save to disk? (Y/N)";;GET A
$;IF A$="Y" THEN VTAB 16;INPUT "File name?";N$;
PRINT CHR$(4);"BSAVE";N$;"",A";M";",L";D
660 END

```

```

117  FREQ:  DB 0FCH,43H,0FCH,0ACH,0FDH,82H,
          0FDH,0C7H;音频表
118          DB 0FDH,09H,0FDH,33H,0FEH,05H
119          DB 0F8H,8BH,0F9H,05BH,0FBH,04H,
          0FBH,8FH
120          DB 0FAH,14H,0FAH,06BH,0FCH,0BH
121          DB 0FEH,21H,0FEH,56H,0FEH,0C1H,
          0FEH,0E3H
122          DB 0FEH,84H,0FEH,99H,0FFH,02H
123  KEYBL:  DB 41H,11H,12H,21H;键值表,代表:
          0 1 2 3
124          DB 22H,42H,51H,52H;
          4 5 6 7
125          DB 32H,62H;低音键
126          DB 31H,61H;高音键

```

以上程序是在 KDC-Ⅲ 开发机上调试成功的。这时 2764(EPROM)地址 0000H,8155 的命令/状态寄存器地址为 4100H,输入控制字 0DH 时,设置功能是 C 口(地址为 4103H)为输出,B 口(地址为 4102H)为输入。

此程序较好解决了音频和音长的统一,比一般方法压缩了一半数据。

在硬件也可以脱离 KDC-Ⅲ 单片开发机,单独地成为一个应用系统。可以用 P1.0~P1.6(8031 P1 口)来完成扫描键盘和输入键值,用 P1.7 驱动喇叭。整个系统可由 8031,2764,373 三片组成的最小应用系统组成。这时 2764 片选信号接地。从而真正构成一个独立电子琴。也可有 2764(8K)存储近 50 首乐谱的数据,使其具有自动演奏的功能。

当然要逼真地模拟音乐,还仍有一些问题要解决。音乐包括四要素:音频,音长,音色和音强。音频和音长用单片机比较好解决。音强可以用 D/A 转换控制,而音色的模拟将还有一定的问题。尽管如此,用单片机产生音乐将是十分有意义的工作。它将是一个无任何转动的机构的数字信号产生音乐系统,正向各类游戏机一样,音乐是存在 EPROM 中的,对于需要设计带有自动音响功能的电子产品,采用单片机音乐系统,不失为一种可行的技术方案。

第五讲 程序控制结构(一)

李文兵 王玉华 蓝智斌

本讲介绍 C 程序控制结构。在 C 程序中,控制程序流程,使用控制结构。控制结构是由控制语句构成的。控制结构大致分为三种,即:

- 条件分支结构
- 循环控制结构
- 中断与转移结构

使用这三种控制结构,就能够编制出结构化程序。本讲的最后,介绍递归函数,这是一种特殊的程序结构形式。

1 分支结构

(1)if 结构 其结构形式如下所示:

if(条件表达式) 语句;

例如,求 a,b 两个数中大者的程序,就可以使用 if 结构编写,如练习 5.1 所示。

(练习 5.1)

A>TYPE EXP5 __ 1. C

```
main()
{ int a,b,x;
  a=5;b=3;
  x=b;
  if (a>b) x=a;
  printf("max=%d\n",x);
}
```

A>EXP5 __ 1

max=5

(2)if~else 结构 其结构形式如下:

if(条件表达式) 语句 1;

else 语句 2;

(练习 5.2)

A>TYPE EXP5 __ 2. C

```
main()
{ int a=5,b=3,x;
  if (a>b) x=a;
  else x=b;
  printf("max=%d\n",x);
}
```

A>EXP5 __ 2

max=5

(3)嵌套 if~else 结构 if~else 结构中还可以放进 if 结构或 if~else 结构,这叫做嵌套结构:

```
if(条件表达式 1)
{
  if(条件表达式 2)
  {
    if(条件表达式 3) 语句 3;
    else 语句 4;
  }
  else 语句 2;
}
else 语句 1;
```

在这种嵌套结构中,为使嵌套层次关系清晰,我们提倡程序员把程序写成如上所示的齿状模样。

(练习 5.3)

A>TYPE EXP5 __ 3. C

```
main()
{ int number=55;
  /* nested if statement 1 */
  if (number>0){
    if (number%2==1)
      printf("%d positive odd\n",number);
    else
      printf("%d negative\n",number);
  }
  /* nested if statement 2 */
  number=-number;
  if (number>0){
    if (number%2==1)
      printf("%d positive odd\n",number);
    else
      printf("%d positive even\n",number);
  }
  else{
    if (number%2==0)
      printf("%d negative even\n",number);
    else
      printf("%d negative odd\n",number);
  }
}
```

A>EXP5 __ 3

55 positive odd

-55 negative odd

(4)多分支 if~else if 结构 其结构形式如下:

if(条件表达式 1) 语句 1;

else if(条件表达式 2) 语句 2;

else if(条件表达式 3) 语句 3;

:

:

else if(条件表达式 n) 语句 n

else 语句 n+1;

在这种结构形式中,程序只执行满足条件的语句。如果哪个条件都不成立,就执行语句 n+1。

(练习 5.4)

A>TYPE EXP5__ 4. C

```
main()
{ unsigned char c;
  printf("enter char:");
  scanf("%c",&c);
  if (c>=0x00 && c<0x20)
    printf("%x,control code",c);
  else if (c>=33 && c<47)
    printf("%x [%c],special char(group 1)\n",c,c);
  else if (c>=48 && c<=57)
    printf("%x [%c],digits\n",c,c);
  else if (c>=58 && c<=64)
    printf("%x [%c],special char(group 2)\n",c,c);
  else if (c>=65 && c<=90)
    printf("%x [%c],uppercase letter\n",c,c);
  else if (c>=97 && c<=122)
    printf("%x [%c],lowercase letter\n",c,c);
  else if (c>=91 && c<=96)
    printf("%x [%c],special char(group 3)\n",c,c);
  else if (c>=123 && c<=126)
    printf("%x [%c],special char(group 4)\n",c,c);
  else
    printf("%x [%c],non ascii char\n",c,c); }
```

A>EXP5__ 4

enter char: l
31 [l],digits

A>EXP5__ 4

enter char: \$
24 [\$],special char(group 1)

A>EXP5__ 4.

enter char: &
26 [&],special char(group 1)

(5)Switch 结构 Switch 语句可实现多分支结构:

switch(变量)

```
{ case 常量表达式 1:
    语句 1;
    break;
  case 常量表达式 2:
    语句 2;
    break;
    :
  case 常量表达式 n:
    语句 n;
    break;
  default:
    语句 n+1;
}
```

当 switch 括号中的变量值与某 case 中常量表达式的值相同时,就执行该 case 中的语句。若该语句后跟有 break 语句,则跳出 switch 结构。若无 break 语句,则

顺序执行下一个 case 中的语句。若变量的值与任何 case 的常量表达式的值都不相同,则执行 default 中的语句。使用 switch 结构需要注意以下几点:

(1)switch(变量)中的变量只限字符型(char)、整型(int)、枚举型(enum)。

(2)switch 语句也可嵌套使用。

(3)若干个 case 可公用一个语句,如练习 5.5 所示。

(练习 5.5)

A>TYPE EXP5__ 5. C

```
main()
{ int i;
  printf("enter i=");
  scanf("%d",&i);
  switch(i)
  { case 1:
    printf("case 1\n");
    break;
    case 2:
    printf("case 2\n");
    break;
    case 3:
    case 4:
    case 5:
    printf("case 3-5\n");
    break;
    default:
    printf("default");
  }
}
```

A>EXP5__ 5

enter i=1
case 1

A>EXP5-5

enter i=9
default

2 循环结构

用下面的循环控制语句,便可实现循环结构。

- for
- while
- do-while

(1)for 结构 for 结构的形式如下所示:

for(表达式 1;表达式 2;表达式 3) 语句;

其中表达式 1 一般用来初始化循环控制变量;表达式 2 为条件表达式,以此控制循环次数;表达式 3 用来修改循环控制变量。例如:

```
int i;
for (i=0;i<100;i++)
  printf("%d",i);
```

和

```
int i;
for (i=100;i>0;i--)
```

```
printf("%d,i);
```

此外,可以使用两个循环控制变量,如:

```
int i,j;
for (i=0,j=100;i<j;i++,j--) 语句;
(练习 5.6)
```

```
A>type exp5__ 6.c
```

```
main()
{ int i,j=0;
  /* for statement 1 */
  for (i=0;i<100;i++) j+=i;
  printf("summation=%d\n",j);
}
```

```
A>exp5__ 6
```

```
summation=4950
```

(2)无限循环的 for 结构 for 语句的三个表达式中任何一个都可以缺省。当表达式 2 缺省时,C 编译系统则认为表达式 2 恒为真,故 for 结构就变成无限循环结构,其特例如下所示:

```
for (;) 语句;
```

为使无限循环的 for 语句在某条件下终止,可加入 break 语句,如练习 5.7 所示。

(练习 5.7)

```
A>TYPE EXP5__ 7.C
```

```
main()
{ int i,j;
  i=0,j=1;
  for (;;)
  { int k;
    k=j,j=i+j,i=k;
    if (k>100) break;
    printf("%d ",i);
  }
}
```

```
A>EXP5__ 7
```

```
1 1 2 3 5 8 13 21 34 55 89
```

(3)while 结构 其结构形式如下所示:

```
while(条件表达式) 语句;
```

功能是:只要条件表达式成立,就执行条件表达式后紧跟的语句;该语句可以是空语句、简单语句、或复合语句。例如:

```
int len=10;
while (len>0)
{ printf(" * ");len--;
```

只要 len 为正,该程序就打印 * 号。

while 的条件表达式中可引进字符输入库函数,如:

```
while ((ch=getchar())!= '\n');
```

其中 getchar()为字符输入库函数。注意:该 while 语句的执行语句就是空语句。这样,只要从键盘上输入的字符不是\n,输入操作就可以连续进行下去。

(练习 5.8)

```
A>TYPE exp5__ 8.c
#include <stdio.h>
```

```
main()
```

```
{ char ch;
  int len=10;
  char s[80];
  printf("while statement test 1\n");
  while (len>0)
  { printf(" * ");
    len--;
  }
  printf("\nwhile statement test 2\n");
  while (scanf("%d",&len)!= '\n')
    printf(" :len=%d\n",len);
}
```

```
A>exp5__ 8
```

```
while statement test 1
*****
while statement test 2
2 3 4 5 (仿效返回)
: len=2
: len=3
: len=4
: len=5
^ C
```

执行结果中注有仿效返回的行,是为使操作者及时了解输入情况而设的操作系统功能。紧跟在仿效返回下面的,才真正是程序执行结果。

(4)无限循环 while 结构 其结构形式为:

```
while(1){....}
```

只要圆括号中的值非 0,就是无限循环 while 结构。

(练习 5.9)

```
A>TYPE EXP5__ 9.C
```

```
main()
{ int len;
  printf("while statement test\n");
  len=10;
  while(1)
  { printf("%d ",len--);
    if (len<=0)
      break;
  }
}
```

```
A>EXP5__ 9
```

```
while statement test
10 9 8 7 6 5 4 3 2 1
```

(5)do-while 结构 其一般结构形式是:

```
do{ 语句;}
while (条件表达式);
```

若条件表达式成立,则执行语句。与 while 结构不同的是,不管条件成立与否,首先要执行一次 do 后面的语句。如练习 5.10 所示

```
A>type exp5__ 10.c
#include <stdio.h>
main()
```

```

{ char ch;
int len;
printf("do-while statement test 1\n");
len=10
do { printf(" * ");
    } while((len--)>=0);
printf("\ndo-while statement test 3\n");
len=50;
do { printf("%d ",len--=5);

```

```

if (len<=0) break;
} while(1);
}
A>exp5__10
do-while statement test 1
*****
do-while statement test 3
45 40 35 30 25 20 15 10 5 0

```

(待续)



电脑巧开发

语音报时钟的原理和实现

清华大学 蔡莲红

石英电子表走时准确,使用方便,它使钟表工业向前跨了一大步。近年来,语音合成技术的进步,使语音闹时、报时的愿望得以实现。比如,时钟到了设定的时间,它用语音告诉你:“时间到了。”有的时钟能随时向您报告准确时间。本文将简单介绍一下语音报时的基本原理和实现。

一、语音报时钟简介

图1表示了语音报时钟的原理框图。它是在电

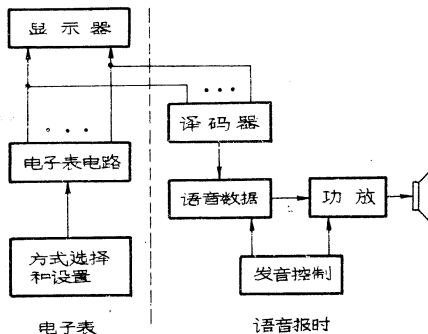


图1 语音报时钟原理框图

子表的基础上,增加了语音报时功能。市场上已有大量的电子表集成块,本文不再讨论。语音报时钟的方式选择和设置跟一般电子表相同。如应具有设置月、日、时、分,设置闹时的时、分功能。因而它应有一个方式选择开关——选择闹时、时间、日期。还应有两个按钮:时(HR)、分(MIN)。它们组合起来实现方式选择和设置功能,如表1。

语音报时部分由译码器、语音数据、发音控制和功放组成。语音数据是数字化了的语音波形。若以语音报时,首先要知道当时的时间。在不改动电子表集成电路的情况下,以连到显示器的连线中截获信号,并进行译码。译码器输出的是待发声的语音数据的地址。然后读出数据,报出时间。发音控制的作用是实现声音强/弱、报时开/关、报时按钮的控制。

表1 电子表方式选择和设置功能

		方式选择器位置			所用键	功 能
		闹时	时间	日期		
闹 时	小时	↑			HR	按HR键,小时加1
	分	↑			MIN	按MIN键,分钟加1
设置时间	小时		↑		HR	按HR键,小时加1
	分		↑		MIN	按MIN键,分钟加1
设置日期	小时			↑	HR	按HR键,小时加1
	分			↑	MIN	按MIN键,分钟加1

二、语音报时的基本原理

1. 语音数字化表示

语音信号处理技术的发展,把模拟语音信号变成数字量存储起来,需要时再恢复的应用日益广泛。事实上,处理数字化语音更为经济、其放音也易于控制。好语音圣诞卡、语音玩具、语音闹时和报时就是最简单的应用。

数字化语音的获取和重放原理如图2所示。

从话筒输出的模拟语音信号,经放大和滤波后,送模数转换器(A/D)。A/D按一定时间间隔(按一定的抽样率),将模拟语音信号变成数字量。这些有序排列的数据。其数值大小表示语音抽样的幅度,就是数字化的语音。A/D每完成一次转换,都把数据存入存储器中。需要放音时,先从存储器中取出数据,经数模转换变成模拟量,再经功率放大和滤波,即可放音了。在上述过程中应解决好以下几个问题:

(1)滤波器的截止频率和语音抽样率:首先根据系统对放音质量的要求,确定语音抽样率。在一些简单的应用系统中,语音抽样率为6K—10KHz。根据采样定理,相应滤波器的截止频率为3K—5KHz。

(2)语音抽样的码位:在数字表示中,采用二进制编码是方便的。这里码位指二进制编码的码位,也就是指A/D转换的位数。一般取8—12位。

语音的数字化过程,通常在计算机或专用开发系统中完成。

2. 数字化语音的放音:

如果已经获得语音数据,可将其写入EPROM中,

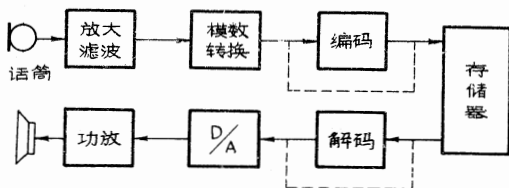


图2 语音数字化和放音

利用图3的电路放音。图中,三片74LS161是EPROM的地址计数器。开关K未接通时,地址计数器输出0地址。开关K接通一次,单稳电路74LS123输出一正脉冲,地址计数器在555供给的时钟作用下,开始计数。1408是数模转换器,LM386是功率放大器。值得指出的是:555输出的时钟频率应与原A/D转换的频率相等。

3. 语音的编码和解码

按照上述的原理实现语音数字化和放音,线路简单,但数据量大。如果语音抽样率为8KHz,每个抽样

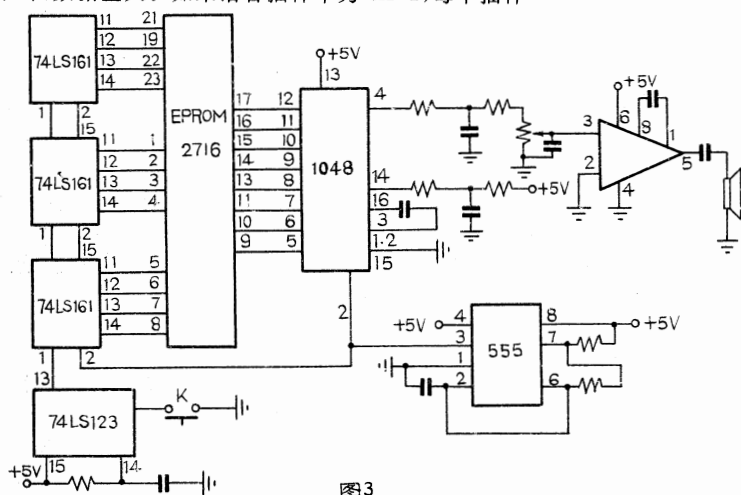


图3

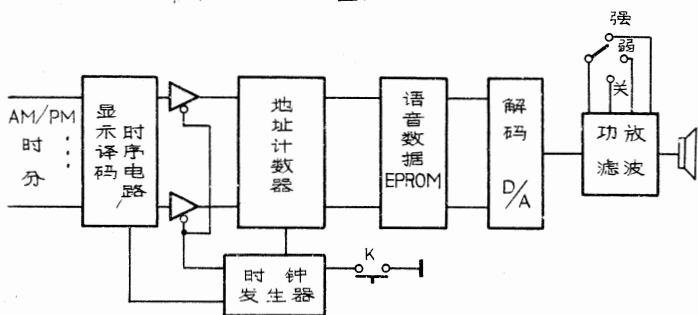


图4 语音报时的实现框图

用8位二进制表示,那么每秒钟得到8KB(64K位)语音数据。若想讲“早上五点五十八分”,大约需3秒钟,也就是说需24KB存储器。若要能报任何时和分,需要的存储器就更大了。研究发现,可以对语音数据进行压缩编码,而基本不影响发音质量。带来的益处是减少了数据量。因此在实用系统中,几乎都采用了压缩编码,或由硬件实现,或由软件实现。放音时,再进行解码。这就是图2中编码和解码的作用。

在语音的编码/解码技术中,最常用的是自适应差值脉冲编码(Adaptive Delta Pulse Code Modulation 简称ADPCM)。市场上已有多种ADPCM编解码的集成电路,下面仅列出几种:OKI公司的:MSM5205、MSM5218、MSM5248、MSM6295, TI公司的TSP50C41(42/43/44), Toshiba的T6831, Harris公司的HC55536、HC55564和SamSung的KS5911和KS5912等。采用ADPCM编码,可将原抽样的8位数据,压缩到4、3、2或1位,大大减少了数据量。其放音电路的组成与图3类似,只是将数模转换集成电路换成专用ADPCM集成电路。

三、语音报时的实现

要用语音报时,先确定要报的音,如:月、日、点、分、1—59、早上、上午、下午和晚上等。然后建立标准数字音库(写好EPROM)、各发音的首地址索引,最后按

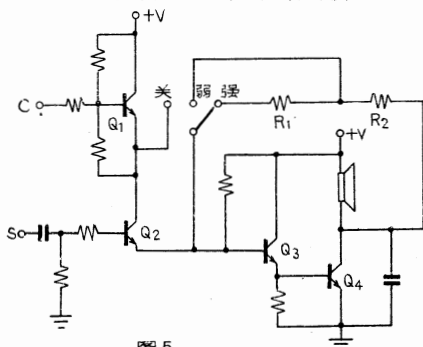


图5

图4的框图设计具体电路。其中显示译码的作用是把送显示器的AM/PM、时、分、月、日信号取来,译出待发音的EPROM首地址。由于每次报音需由几个短音组合而成,因此还需设计一时序电路,顺序定时送出EPROM地址。这部分电路可由TTL集成电路或门阵电路实现。图4中其它部分都已熟悉。这里仅介绍一简单廉价的功放滤波电路,如图5。图中S是语音信号输入端。C是控制端。一般情况下(报时),C为高, Q1导通,关掉报时开关时,C为低, Q1截止, Q2不能传送语音信号。 Q3和Q4构成反相功率放大器。

一个三掷开关,选择发音强、弱和无音。实际上声音强弱的控制是靠改变功放反馈电阻实现的。反馈电阻增大为 $R1+R2$ 时,声音强,为 $R2$ 时,声音弱。当开关拨到“无音”时,线路中将语音信号短路,故无声音输出。

图4中,除功放滤波控制开关和扬声器外,均可做成一块中规模集成电路,若再配一块电子表电路和显示器,就可形成一个语音报时钟。



简 易 单 片 机 电 子 琴

张培仁 张新宇

在本刊 91 年第二、三期上,向大家介绍了 KDC-Ⅲ 的软硬件设计。在第四期上介绍了自动奏乐系统。

本期我们介绍一个让单片机变成电子琴的程序。硬件装置仍采用 KDC-Ⅲ 开发机,利用其按键进行演奏。该电子琴框图如图 1 所示。

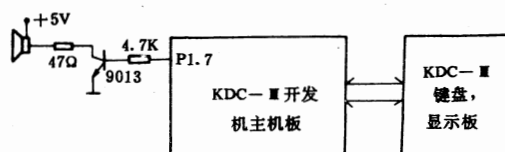


图 1

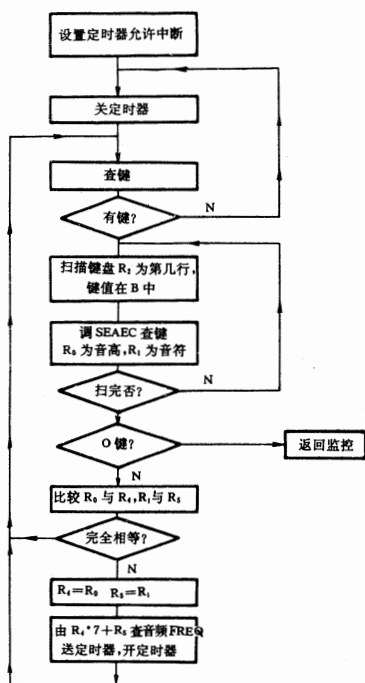


图 2

键盘译码表和功能表如下:

0 C3B2	1 C0B3	2 C0B4	3 C3B4
4 C4B3	5 C1B3	6 C1B4	7 C4B4
8 C5B3	9 C2B3	A C2B4	B C5B4

键盘上 1~7 代表音符 1~7。同时按下 8(或 9)和 1~7 键表示高音。同时按下 A(或 B)和 1~7 键表示低音。按键 0 则退出本程序。

电子琴程序设计思路是:查键盘,根据键值表的频率送定时器,并由 8031 的 P1.7 口输出去驱动喇叭发声。程序流程图如图 2 所示。

首先检查键盘。如有键,先延时 1ms,逐行扫描,并调用 SEARC 子程序将键值转换成音符键(1~7)或者高键(低、中、高)。然后判断所得键值与原先键值是否相等。相等即按键不变,继续查键盘。不相等,查音频表 FREQ,设置新的定时常数,产生新的声音。继续查键盘直至按下 0 键。

电子琴程序清单:

```

1      ORG 2000H
2      MOV IE, # 82H      ;允许中断
3      ANL 89H, # 0F0H    ;置定时器为工
                           ;作方式 1
4      ORL 89H, # 01H
5      AJMP START
6      ORG 200BH          ;中断服务程序
7      CLR TR0
8      CPL P1.7           ;取反,产生方
                           ;波
9      MOV TH0, R6
10     MOV TL0, R7
11     SETB TR0
12     RETI
13     START: CLR TR0
14     KEYBD: MOV DPTR, #  ;置 8155 为:
15              4100H      ;C 口输出
16              MOV A, # 0DH ;C 口输出
17              MOVX @DPTR, A ;B 口输出
18              MOV A, # 1BH ;查键盘
19              MOV DPTR, #
20              4103H
21              MOVX @DPTR, A
22              MOV DPTR, #
23              4102H
24              MOVX A, @DPTR
25              CPL A
26              ANL A, # 18H
27              JZ START    ;无键转 START
                           ;有键,延时 1ms
DE-   MOV R5, # 0EH
LAY:   ACALL DIMS
       DJNZ R5, DELAY
    
```


28	MOV R3, #01H		72	MOV B, #07H	
29	MOV R0, #00h		73	MUL AB	
30	KSCAN: MOV A, R3	;逐行扫描键盘	74	ADD A, R5	
31	MOV DPTR, #4103H		75	DEC A	
32	MOVX @DPTR, A		76	RL A	
33	MOV DPTR, #4102H		77	MOV DPTR, #FREQ	
34	MOVX A, @DPTR		78	MOV R2, A	
35	CPL A		79	MOVC A, @A + DPTR	
36	ANL A, #18H		80	MOV R6, A	
37	JZ NOKEY	;扫描行无键	81	MOV A, R2	
38	RL A	;有键, 键值在 B	82	INC A	
39	SWAP A		83	MOVC A, @A + DPTR	
40	MOV B, A		84	MOV R7, A	
41	MOV A, R3		85	MOV TH0, R6	
42	MOV R2, #01H		86	MOV TL0, R7	
43	DD: CLR C		87	SETB TR0	
44	RRC A		88	AJMP KEYBD	;继续扫键直至 0 键
45	JZ NPP		89	DIMS: MOV R2, #64H	;延时子程序
46	INC R2		90	DLAY: NOP	
47	AJMP DD		91	NOP	
48	NPP: MOV A, R2		92	NOP	
49	SWAP A		93	DJNZ R2, DLAY	
50	ADD A, B		94	RET	
51	MOV R2, A		95	SEARC: MOV B, #00H	;查键值子程序,
52	ACALL SEARC	;调用 SEARC 查键	96	MOV DPTR, #KEY-BL	;R1 为音符 (1~7)
53	NOKEY: MOV A, R3	;扫描行无键处理,	97	SEA: MOV A, #00H	;R0 为音高,
54	RL A		98	MOVC A, @A + DPTR	;R0=0,1,2
55	MOV R3, A		99	XRL A, R2	;表示: 中音, 低音, 高音
56	CJNE R3, #40H, KSCAN	;未完扫下行	100	JZ YES	
57	MOV A, R1		101	INC B	
58	JZ EXIT	;0 键转 STT	102	INC DPTR	
59	XCH A, R5	;与原键比较	103	AJMP SEA	
60	XRL A, R5		104	YES: CLR C	
61	JZ UNCH1	;即比较: R5-R1	105	MOV A, B	
62	MOV A, R0	; R4-R0	106	SUBB A, #08H	
63	MOV R4, A		107	JNC LOW	
64	AJMP CHAN	;变转 CHAN	108	MOV R1, B	
65	EXIT: LJMP 0000H	;0 键退出程序	109	RET	
66	UNCH: AJMP KEYBD		110	LOW: MOV A, B	
67	UNCH1: MOV A, R0		111	SUBB A, #0AH	
68	XCH A, R4		112	JNC HIGH	
69	XRL A, R4		113	MOV R0, #01H	
70	JZ UNCH	;键未变继续查键		(转至 24 页)	
71	CHAN: MOV A, R4	;键改变重置时间常数			

天津纺织工学院 沈乃才 宋振环

一、系统构成

整个系统以 8031 为 CPU, 系统硬件如图 1 所示。配置 2KB 程序存储器(2716)。系统还配置了 6 位 LED 数字显示器用以显示时间值, 12 个键的键盘。共用一

二、软件

1. 主程序

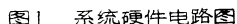
方式、启动定时器 0、显示时间并判定预定时间是否到?若时间到,则发出相应的控制信号,启动被控对象进行工作。

2. 中断服务程序

定时器 T0 工作在方式 1,每隔 100ms 请求中断一次,每执行一次中断服务程序,溢出次数计数器单元初值减 1,判其减 1 后是否为零,如不为零,恢复现场,从中断返回。减足 10 次后为零。秒的个位值增 1。如此反复,并按秒、分、时的顺序逐位判别,按正常时间规律进行进位或清零,然后中断返回。

3. 键入子程序

为了在开机时,使其显示的时间与标准时间相吻合,必须进行标



准时间值的输入。本系统采用程控键扫描方式。键输入子程序流程图如图 4 所示。程序首先对 8155 初始化,继而判别键盘有无键闭合,利用二次调用显示子程序延迟 12ms 消除键抖动。为了保证键的一次闭合只作一次处理,必须等待输入键释放后才对输入键进行处理。最后将键号值送往 A 中。

4. 显示子程序

显示子程序流程图如图 5 所示。显示数据缓冲区

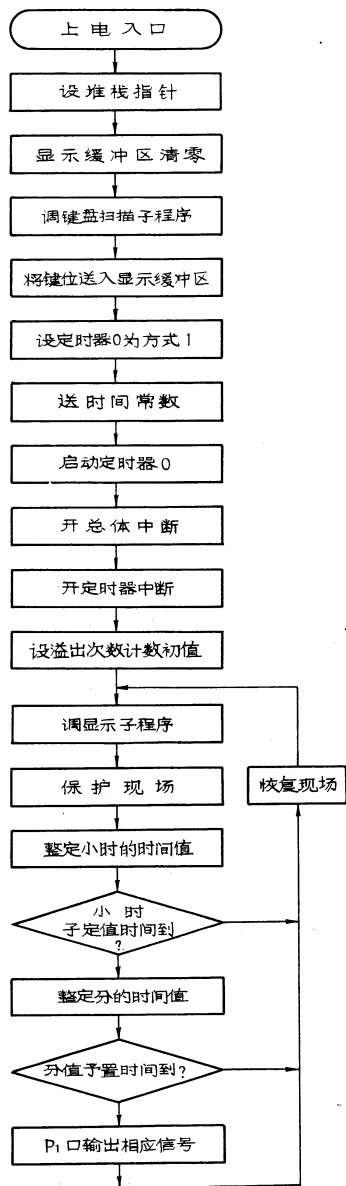


图 2

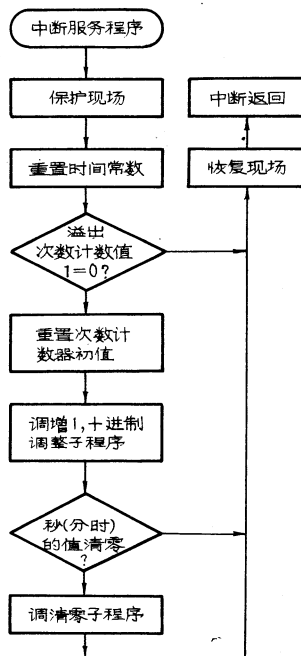


图 3

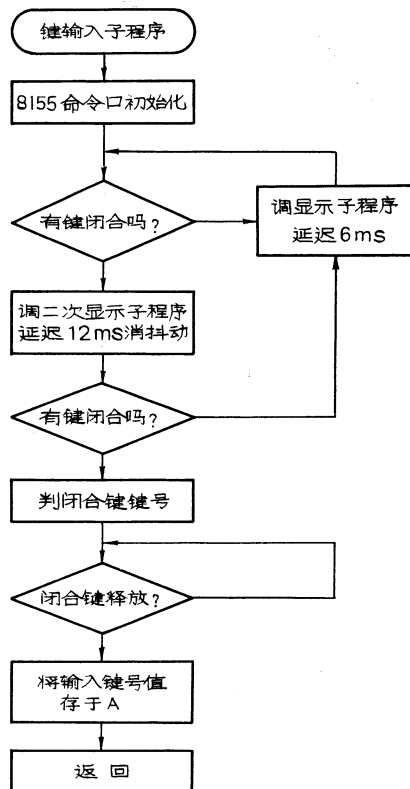


图 4

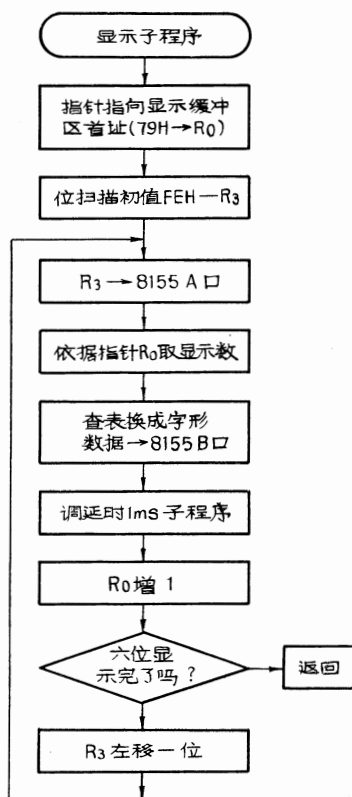


图5 显示子程序流程图

为 79H~7EH。主程序对 8155 的初始化之后,将欲显示的数据写入 79H~7EH。循环调用此显示子程序,显示器将时间值显示出来。直到各位都扫描一次,返回主程序。

该系统在我院已运行三年之久。实践证明,运行效果良好,工作可靠。不仅保证了学生有充分的课外听力练习时间,而且还节省了外语教师的大量课时。本系统于 1988 年底也通过了有关专家的技术鉴定。

本系统亦可在原有基础上稍加修改,成为各种实时控制器使用。

程序清单:

A>type ss.asm

```

ORG 8000H
START:  AJMP MAIN
        ORG 800BH
        AJMP PIT0
        ORG 8010H
MAIN:    MOV SP, #60H
        MOV R0, #79H
        MOV R7, #06H
ML1:    MOV @R0, #00H
        INC R0
        DJNZ R7, ML1
  
```

```

MOV R0, #7EH
MOV R7, #06H
ML2:    PUSH PSW
        SETB PSW.3
        LCALL 078DH
        POP PSW
        MOV @R0, A
        DEC R0
        DJNZ R7, ML2
        MOV TMOD, #01H
        MOV TL0, #B8H
        MOV TH0, #3CH
        SETB 8CH
        SETB AFH
        SETB 0ASH
        MOV 30H, #0AH
ML0:    LCALL 0745H
        PUSH PSW
        PUSH ACC
        MOV A, 7EH
        SWAP A
        ORL A, 7DH
        CJNE A, #06, NEXT1
        SJMP F1
NEXT1:  CJNE A, #07, NEXT2
        SJMP F2
NEXT2:  CJNE A, #15, NEXT3
        LJMP F3
NEXT3:  CJNE A, #16, NEXT4
        LJMP F4
NEXT4:  CJNE A, #18, NEXT5
        LJMP F5
NEXT5:  CJNE A, #19, NEXT6
        LJMP F6
NEXT6:  CJNE A, #20, NEXT7
        LJMP F7
NEXT7:  CJNE A, #21, HFXC
        LJMP F8
HFXC:   POP ACC
        POP PSW
        LJMP ML0
F1:     MOV A, 7CH
        SWAP A
        ORL A, 7BH
        MOV R5, #7EH
        CJNE A, #44, NEXTA1
        LJMP WORK
NEXTA1: MOV R5, #0FCH
        CJNE A, #45, NEXTA2
        LJMP WORK
NEXTA2: MOV R5, #FEH
        CJNE A, #46, NEXTA3
        LJMP WORK
NEXTA3: LJMP HFXC
  
```

```

F2:    MOV A,7CH
        SWAP A
        ORL A,7BH
        MOV R5,#FAH
        CJNE A,#17,NEXTB1
        LJMP WORK
NEXTB1: MOV R5,#FCH
        CJNE A,#18,NEXTB2
        LJMP WORK
NEXTB2: MOV R5,#FEH
        CJNE A,#19,NEXTB3
        LJMP WORK
NEXTB3: MOV R5,#FAH
        CJNE A,#50,NEXTB4
        LJMP WORK
NEXTB4: MOV R5,#FFH
        CJNE A,#51,NEXTB5
        LJMP WORK
NEXTB5: LJMP HFXC
F3:    MOV A,7CH
        MOV A,7CH
        SWAP A
        ORL A,7BH
        MOV R5,#FEH
        CJNE A,#49,NEXTC1
        LJMP WORK
NEXTC1: MOV R5,#FCH
        CJNE A,#50,NEXTC2
        LJMP WORK
NEXTC2: MOV R5,#FEH
        CJNE A,#51,NEXTC3
        LJMP WORK
NEXTC3: LJMP HFXC
F4:    MOV A,7CH
        SWAP A
        ORL A,7BH
        MOV R5,#FAH
        CJNE A,#22,NEXTD1
        LJMP WORK
NEXTD1: MOV R5,#FCH
        CJNE A,#23,NEXTD2
        LJMP WORK
NEXTD2: MOV R5,#FEH
        CJNE A,#24,NEXTD3
        LJMP WORK
NEXTD3: MOV R5,#FAH
        CJNE A,#55,NEXTD4
        LJMP WORK
NEXTD4: MOV R5,#FFH
        CJNE A,#56,NEXTD5
        LJMP WORK
NEXTD5: LJMP HEXC

```

```

F5:    MOV A,7CH
        SWAP A
        ORL A,7BH
        MOV R5,#FEH
        CJNE A,#09,NEXTE1
        LJMP WORK
NEXTE1: MOV R5,#FCH
        CJNE A,#10,NEXTE2
        LJMP WORK
NEXTE2: MOV R5,#FEH
        CJNE A,#11,NEXTE3
        LJMP WORK
NEXTE3: MOV R5,#FAH
        CJNE A,#42,NEXTE4
        LJMP WORK
NEXTE4: MOV R5,#FCH
        CJNE A,#43,NEXTE5
        LJMP WORK
NEXTE5: MOV R5,#FEH
        CJNE A,#44,NEXTE6
        LJMP WORK
NEXTE6: LJMP HFXC
F6:    MOV A,7CH
        SWAP A
        ORL A,7BH
        MOV R5,#FAH
        CJNE A,#15,NEXTF1
        LJMP WORK
NEXTF1: MOV R5,#FFH
        CJNE A,#16,NEXTF2
        LJMP WORK
NEXTF2: LJMP HFXC
F7:    MOV A,7CH
        SWAP A
        ORL A,7BH
        MOV R5,#FEH
        CJNE A,#00,NEXTG1
        LJMP WORK
NEXTG1: MOV R5,#FCH
        CJNE A,#01,NEXTG2
        LJMP WORK
NEXTG2: MOV R5,#FEH
        CJNE A,#02,NEXTG3
        LJMP WORK
NEXTG3: MOV R5,#FAH
        CJNE A,#33,NEXTG4
        LJMP WORK
NEXTG4: MOV R5,#FCH
        CJNE A,#34,NEXTG5
        LJMP WORK
NEXTG5: MOV R5,#FEH
        CJNE A,#35,NEXTG6
        LJMP WORK
NEXTG6: LJMP HFXC

```


F8:	MOV A,7CH	JNZ RET0
	SWAP A	ACALL CLR0
	ORL A,7BH	MOV R0,#7EH
	MOV R5,#FAH	ACALL DAAD1
	CJNE A,#06,NEXTH1	MOV A,R2
	LJMP WORK	XRL A,#24H
NEXTH1:	MOV R5,#FFH	JNZ RET0
	CJNE A,#07,NEXTH2	ACALL CLR0
	LJMP WORK	RET0:
NEXTH2:	LJMP HFXC	POP ACC
WORK:	MOV 90H,R5	POP PSW
	LJMP HFXC	RETI
PIT0:	PUSH PSW	DAAD1:
	PUSH ACC	MOV A,@R0
	SETB PSW,3	DEC R0
	MOV TH0,#3CH	SWAP A
	MOV TL0,#B8H	ORL A,@R0
	MOV A,30H	ADD A,#01H
	DEC A	DA A
	MOV 30H,A	MOV R2,A
	JNZ RET0	ANL A,#0FH
	MOV 30H,#0AH	MOV @R0,A
	MOV R0,#7AH	MOV A,R2
	ACALL DAAD1	INC R0
	MOV A,R2	ANL A,#F0H
	XRL A,#60H	SWAP A
	JNZ RET0	MOV @R0,A
	ACALL CLR0	RET
	MOV R0,#7CH	CLR0:
	ACALL DAAD1	CLR A
	MOV A,R2	MOV @R0,A
	XRL A,#60H	DEC R0
		MOV @R0,A
		RET

(上接 47 页)

```

10 FOR I=1 TO 5
20 GOSUB 100
30 PRINT S
40 NEXT I
100 FOR I=1 TO 20
110 S=0
120 INPUT B(I)
130 S=S+B(I)
140 NEXT I
150 RETURN

```

答:(1)缺少数组说明语句,5 DIM B(20)

(2)主程序和子程序使用了同一个变量,应把 10 行和 40 行变量 I 改成 K。

(3)第 10 行和第 100 行的循环终值应调换。

(4)语句 S=0 应放在循环外,即改为:

```

100 S=0
110 FOR I=1 TO 5
正确的程序如下:
5 DIM B(20)
10 FOR K=1 TO 20
20 GOSUB 100
30 PRINT S
40 NEXT K
50 END
100 S=0
110 FOR I=1 TO 5
120 INPUT B(I)
130 S=S+B(I)
140 NEXT I
150 RETURN

```



用微电脑检查印刷电路板

易齐干

怎样使用微电脑自动检查电子线路板的焊接情况,是保证电路板质量的重要手段。现在,使用 $\mu\text{p}-80$ 教育部件,介绍检查诊断印刷线路工作状态是否正常的基本方法。被检的电子线路板是以基础篇中的反相器与锁存器板为例。

1. 检查反相器板

(1) 反相器板的印刷线路图及电源理如图 1 所示。

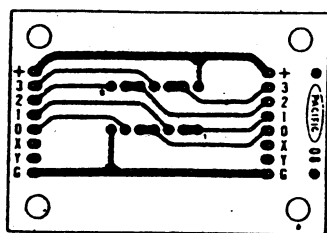
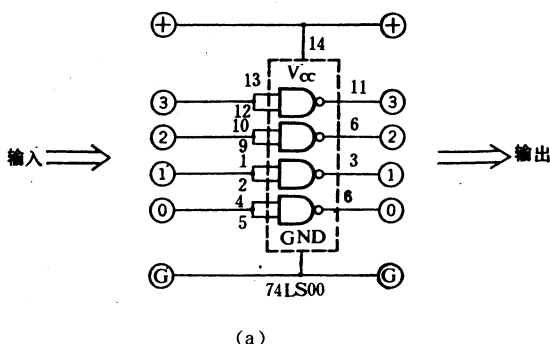


图 1

使用 $\mu\text{p}-80$ 教育部件检查的反相器 74LS00 板,如图 2 所示。

74LS00 为“四与门”电路,因将每个与门输入端合二为一,故实为四反相器功能。

(2) $\mu\text{p}-80$ 输入输出板上各口的分配如图 3 所示。

端口 A、B 与 C 高位为输出口,端口 C 的低位为输入口。

① 端口 C 的高位向反相器输出检查数据。

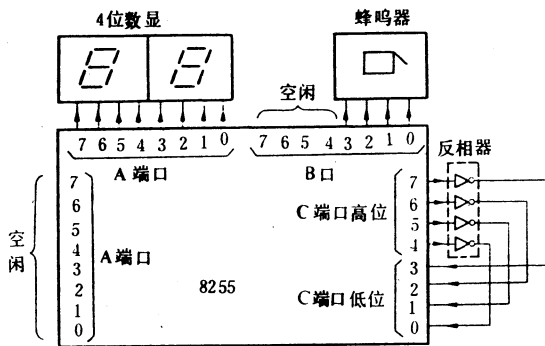


图 3

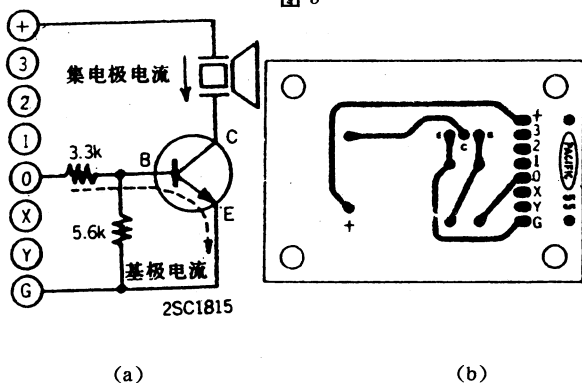


图 4

② 反相器反转后的数据输入给端口 C 的低位。

③ 端口 A 输出检查结果进行 2 位数字显示。

当显示 FF, 检查结果为正常;当显示其它数据, 检查结果为不正常。与端口 B 相连的蜂鸣器发出蜂鸣, 压电蜂鸣器的印刷板图及原理图如图 4 所示

(3) 检查反相器板的流程图如图 5 所示。

由 00E0H 地址开始, 顺序地设置并存储用于检查的输出数据和验证数据。如表 1 和图 6 所示。

① 反相器由第 0 位到第 3 位依次输出 1 (高电平) 为输出数据。(即: 10, 20, 40, 80)。

② 预先设置的验证数据与反相器反转后的数据进行比较, 看其是否一致。

同样, 可以求出第 1 位到第 3 位相对于输出数据的验证数据。另外, 00E8H 地址数据 FFH 是检查结束数据。

参考上述内容, 编制成的反相器部件检查程序如

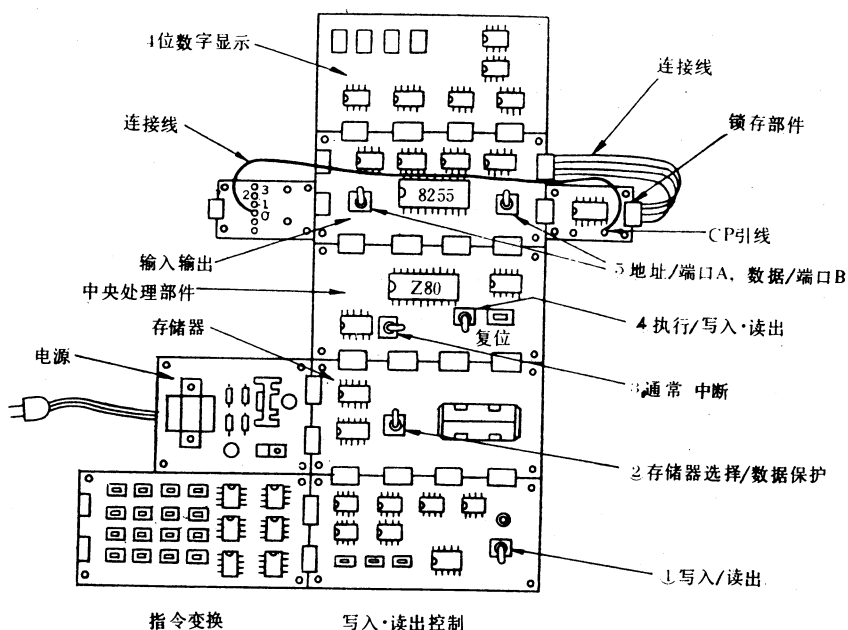


图 2

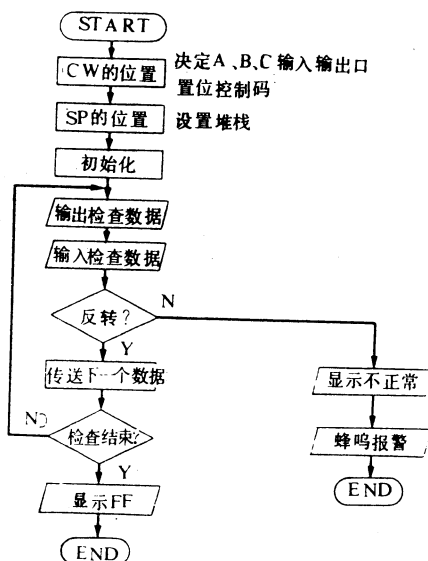


图 5

表 1

地址	数据	注释
E 0	10	第0位的输出数据
E 1	1E	第0位的验证数据
E 2	20	第1位的输出数据
E 3	2D	第1位的验证数据
E 4	40	第2位的输出数据
E 5	4B	第2位的验证数据
E 6	80	第3位的输出数据
E 7	87	第3位的验证数据
E 8	FF	存入FF为检查结束数据

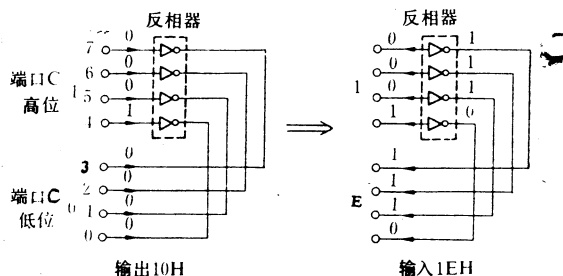


图 6

表 3

部件名称	开关No	开关方向
写入·读出控制	1	读出侧
存储器	2	存储器选择侧
中央处理	3	通常侧
输入输出	4	写入·读出侧
	5	端口A、端口B侧

表 4

显示	原因
10	第0位断线
20	第1位断线
40	第2位断线
80	第3位断线

表 2 所示。

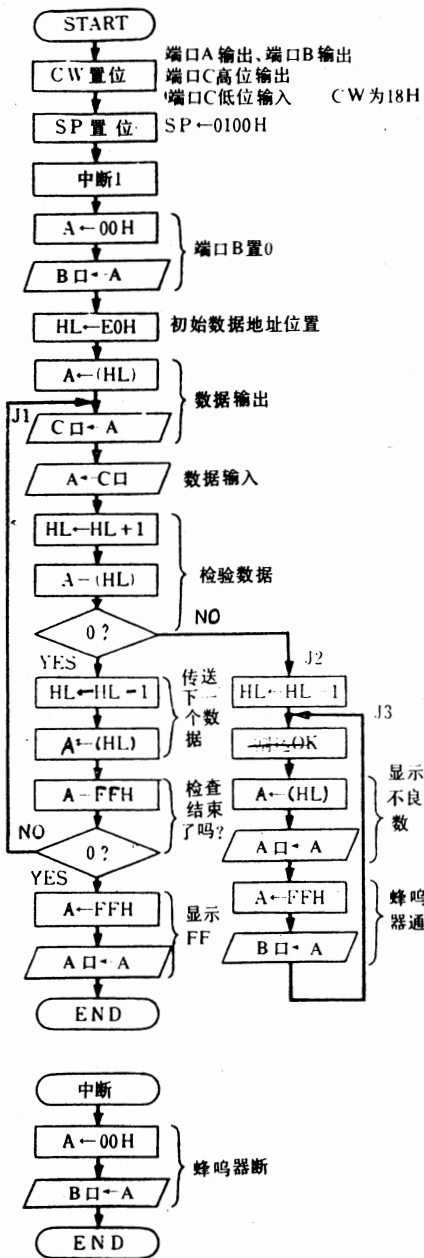
(4) 检查方法

① 表 2 所示的检查程序写入完毕, 并进行检验之后, 照图 2 连接各部件。

② 各开关的位置与表 3 一致。

③ 按动中央处理部件的复位开关, 其时间为 1 秒

表 2



标号	助记符	地址	机器语言	注释
START	LD A, 81H	00	3E 81	CW置位
	OUT(03H), A	02	D3 03	
	LD SP, 0100H	04	31 00 01	SP置位
	IMI	07	ED 56	中断
	LD A, 00H	09	3E 00	B口置于0
	OUT(01H), A	0B	D3 01	
	LD HL, E0H	0D	21 E0 00	初始数据地址置位
	LD A, (HL)	10	7E	数据输出
J 1	OUT(02), A	11	D3 02	
	IN A, (02H)	13	DB 02	数据输入
	INC HL	15	23	
	CP (HL)	16	BE	检验数据
	JP NZ, J2	17	C2 26 00	
	INC HL	1A	23	传送下一数据
	LD A, (HL)	1B	7E	
	CP FFH	1C	FE FF	检查结束了吗?
	JP NZ, J1	1E	C2 11 00	
	LD A, FFH	21	3E FF	显示FF
	OUT(00H), A	23	D3 00	
	HALT	25	76	
J 2	DEC HL	26	2B	
J 3	EI	27	FB	可以中断
	LD A, (HL)	28	7E	显示不良数
	OUT(00H), A	29	D3 00	
	LD A, FFH	2B	3E FF	蜂鸣器通
	OUT(01H), A	2D	D3 01	
	JP J3	2F	C3 27 00	
中断	LD A, 00H	38	3E 00	
	OUT(01H), A	3A	D3 01	用中断, 蜂鸣器断
	HALT	3C	76	
DATA		E0	10	检验第0位
		E1	1E	
		E2	20	检验第1位
		E3	2D	
		E4	40	检验第2位
		E5	4B	
		E6	80	检验第3位
		E7	87	
		E8	FF	检验结束

钟左右,将处于写入读出侧的开关④搬至实行侧,于是,4位数字显示部件的右2位显示FF或其它数据。

若显示FF,表示反相器合格。若为其它数据,说明不合格。此时,与端口B连接的蜂鸣器发出蜂鸣。若参照表4的检验表,则很容易查出故障所在。 $\mu\text{P}-80$ 中央处理部件的开关③搬向中断侧,则中止蜂鸣。

2. 检查锁存器板

(1)锁存器电路如图7所示。

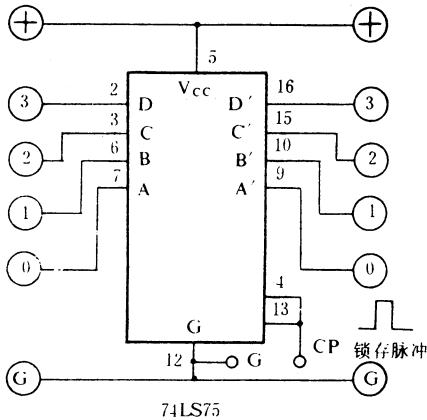


图7

当锁存器板输入信息时,按动某个按键锁存住这一变化的信息,并使之不变。 $74\text{LS}75$ 具有这种锁存功能,见图7。

实际上是将 Ω 型锁存脉冲加在CP脚上,(Ω 为高电

平时) $74\text{LS}75$ 的输入数据将在输出端保持不变。当CP为低电平时,即使输入数据变化,输出数据也不变化。

(2)使用 $\mu\text{P}-80$ 教育部件检查锁存器板的连接图如图8所示。

$\mu\text{P}-80$ 中输入输出的分配如图9所示。端口A、B、C的低位为输出口。端口C的高位为输入口。

①端口C的低位向锁存器板输出检查数据。

②向端口C的高位输入锁存的数据。

③由端口B向显示4位数字部件输出检查结果。当显示FF,检查结果为正常。显示其它数据,检查结果不正常。

④由端口A向锁存部件的CP输出锁存数据,并由显示4位数字部件显示该数据。

(3)检查锁存器部件的板的流程图如图10所示。简要说明如下。

①由端口C低位输出锁存的数据。同时将该数据保存在CPU的C寄存器中。

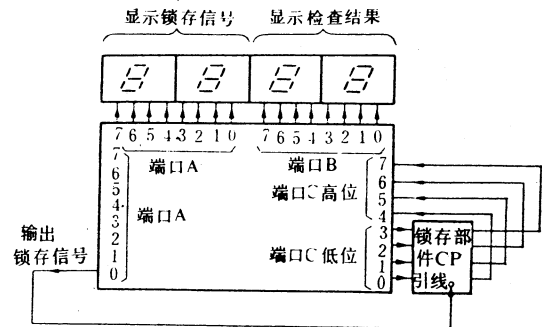


图9

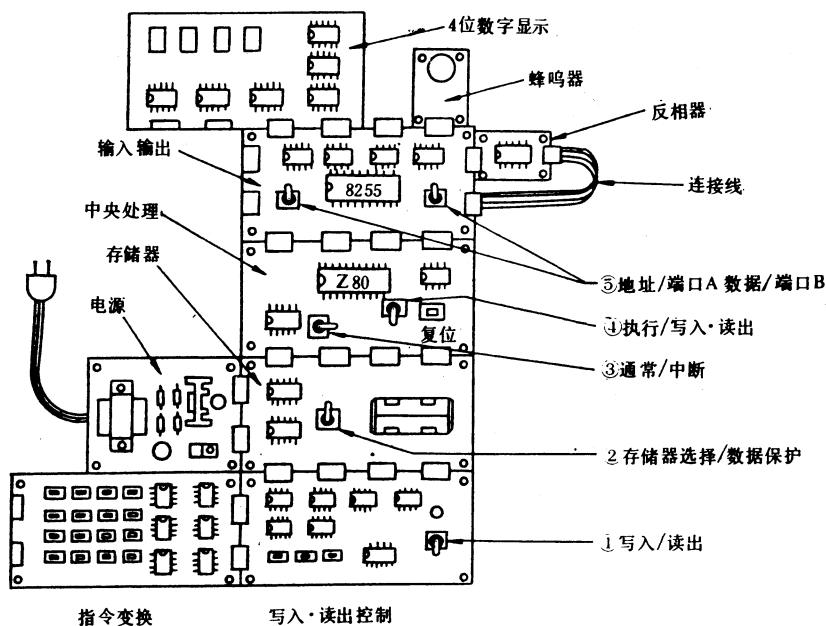


图8

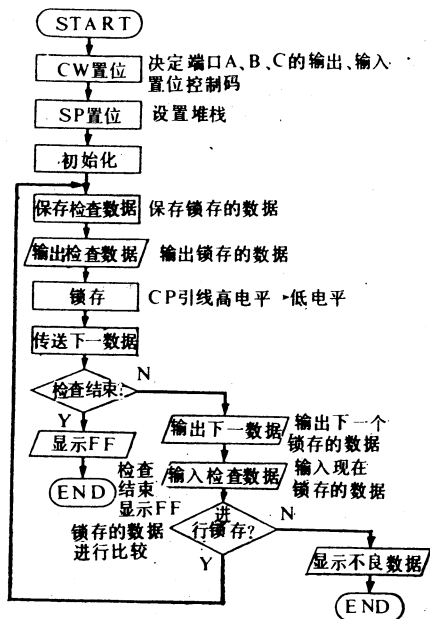


图 10

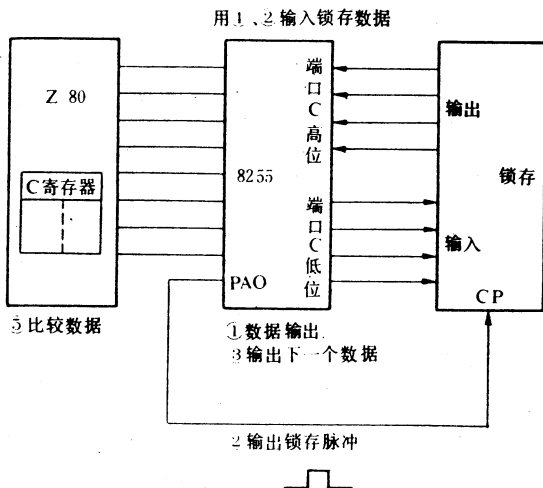


图 11

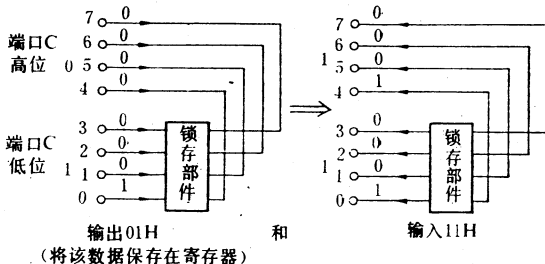


图 12

②端口 A 第 0 位的输出由高电平变为低电平时, 控制 LS75 锁存。

③由端口 C 的低位输出下一个锁存的数据。

④将②锁存的数据, 从端口 C 的高位输入。

⑤ $\mu p-80$ 比较由④输入的数据和由①保持在 C 寄存器中的数据。此时, 若锁存, ①输出的数据, 则与 C 寄存器的数据一致。若没锁存, 则与 C 寄存器的数据不一致。其原因是原封不动地输入③所输出的数据, 或是通过焊锡点的情况变化输入了其它的数据。上述关系如图 11 所示。

⑥数据比较结果若一致, 则进行下一位; 若不一致, 仍显示这一位。

由端口 C 输入的数据与 C 寄存器中原来的数据进行比较, 如图 10 所示。当数据不一致时, 可按下述方法用移位法变换输入数据, 再和 C 寄存器的数据进行比较。

	端口 C 高位	端口 C 低位
①端口 C 输入数据 (11H)	0 0 0 1	0 0 0 1
②AND F0H	X X X X	X X X X
③向右移位 4 次	① 0 0 0 0	1 0 0 0
	② 0 0 0 0	0 1 0 0
	③ 0 0 0 0	0 0 1 0
	④ 0 0 0 0	0 0 0 0
	0 0 0 0	0 0 0 1

④与 C 寄存器的数据 (01H) 进行比较

同样方法由第 1 位到第 3 位变换端口 C 输入数据, 与 C 寄存器数据进行比较为合适。

参考上述内容编制的锁存部件检查程序, 如表 5 所示。

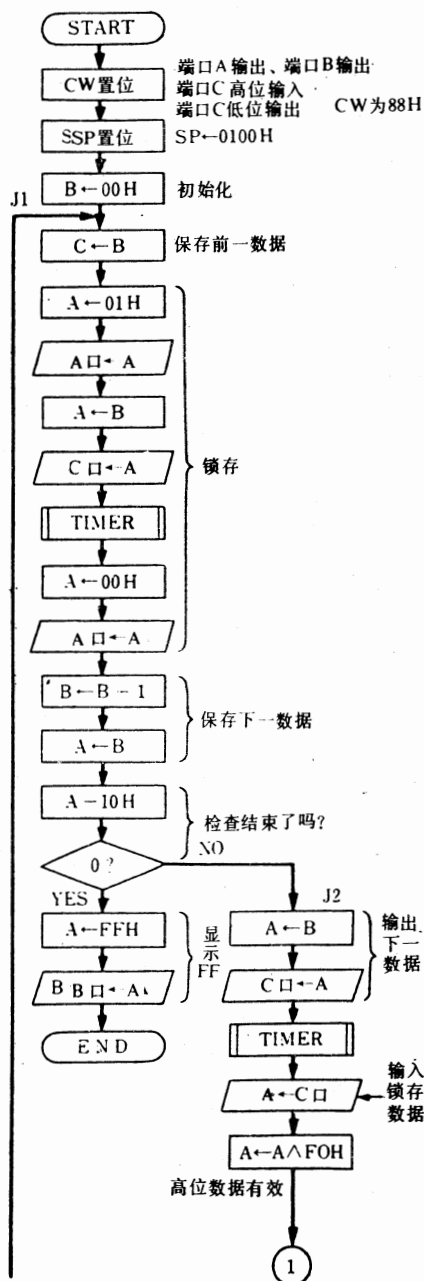
表 6

部件名称	开关 N。	开关方向
写入·读出控制	1	读出侧
存储器	2	存储器选择侧
中央处理	3	通常侧
	4	写入·读出侧
输入输出	5	端口 A 端口 B 侧

表 7

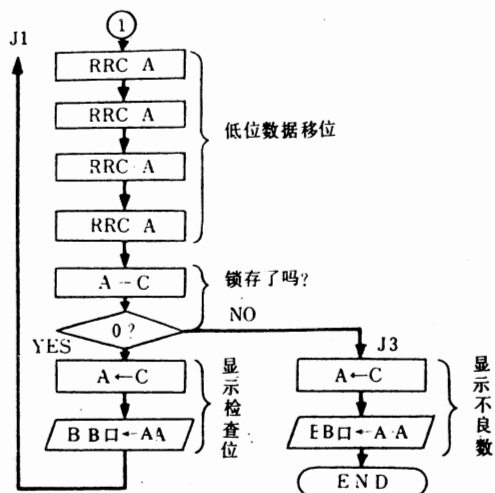
显示	原因
1 0	第 0 位断线,
2 0	第 1 位断线,
4 0	第 2 位断线,
8 0	第 3 位断线,

表 5



标号	助记符	地址	机器语言	注释
START	LD A, 88H	00	3E 88	CW置位
	OUT(03H), A	02	D3 03	
	LD SP, 0100H	04	31 00 01	设置堆栈SP
	LD B, 00H	07	06 00	初始化
J 1	LD C, B	09	48	保存前一数据
	LD A, 01H	0A	3E 01	
	OUT(00H), A	0C	D3 00	
	LD A, B	0E	78	进行锁存
	OUT(02H), A	0F	D3 02	
	CALL TIMER	11	CD 44 00	
	LD A, 00H	14	3E 00	
	OUT(00H), A	16	D3 00	
	INC B	18	04	保存下一数据
	LD A, B	19	78	
	CP 10H	1A	FE 10	检查结束了吗?
	JP NZ, J2	1C	C2 24 00	
	LD A, FFH	1F	3E FF	检查结束
	OUT(01H), A	21	D3 01	显示FF
	HALT	23	76	
J 2	LD A, B	24	78	输出下一数据
	OUT(02H), A	25	D3 02	
	CALL TIMER	27	CD 44 00	
	IN A, (02H)	2A	DB 02	输入锁存数据
	AND FOH	2C	E6 F0	高位数据有效
	RRC A	2E	CB 0F	数据在低位
	RRC A	30	CB 0F	移位
	RRC A	32	CB 0F	
	RRC A	34	CB 0F	
	CP C	36	B9	锁存了吗?
	JP NZ, J3	37	C2 40 00	
	LD A, C	3A	79	显示检查位
	OUT(01H), A	3B	D3 01	
	JP J1	3D	C3 09 00	
J 3	LD A, C	40	79	显示不良位
	OUT(01H), A	41	D3 01	
	HALT	43	76	
TIMER	LD D, H	44	16 50	
J 4	LD E, H	46	1E F0	
J 5	DEC E	48	1D	
	JP NZ, J5	49	C2 48 00	
	DEC D	4C	15	
	JP NZ, J4	4D	C2 46 00	
	RET	50	C9	

接表 5

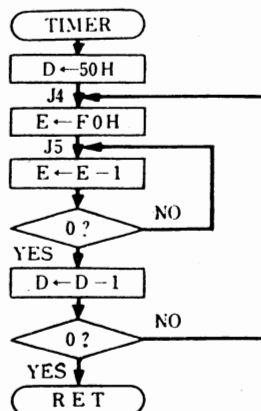


⑦ 检查方法

(1)照图 11 连接各部件。写入表 5 所示的检查程序,并进行检验。

(2)各部件的开关位置与表 6 所示一致。

(3)按动中央处理部件的复位开关,时间为 1 秒左右,将写入、读出一侧的开关④搬至实行侧。于是,显示 4 位数字部件的右边 2 位由 00 依次变化到 0F。最后,若显示 FF,说明锁存器板合格;若显示其它数据,则说明不合格。参考表 7 检验图表,能较容易找出故障原因。



因。

该程序是,用延时方法延长各位的检查时间,如果将程序中“CALL TIMER”语句处,换成执行 3 次“NOP”(00H)指令时,能加快检查速度。

以上讨论了反相器板和锁存器板的检查系统,这仅是用微电脑自动检查的基本事例。用微电脑检查复杂电路板的原理是一样的,最关键的问题是要理解被检查部件的功能、工作原理、并编制出程序。

(上接第 4 页)品,诸如食品、音像磁带、出版物、金属制品以及其它许多物品。它还通常用于包装、销售计账和数据处理等方面。

EAN 码(European Artical Number):它是一种世界各国广泛采用的通用物品条形码。它的符号表示法和用途与 UPC 相似,且相互兼容。

3. 39 码:它是一种包含字母、数字和专用符号的通用条形码。主要用于文献图书、工业生产、仓储、货运、票证、邮政及医院等领域。

4. 25 码:是一种只包含数字的通用条形码。主要用于各种类型的文件、货运及工业生产。

二、条形码系统的工作原理

条形码系统的基本组成部分包括:条形码扫描器、译码器、微型计算机这三大部分。

下面我们以经常使用的光笔为例,来介绍一下条形码系统的工作原理。

光笔作为扫描器的一种,是由发光二极管来产生光源的,经一球型兰宝石透镜聚焦在条形码表面上,其

反射光线强弱与条形码中条和空的颜色深浅、宽窄大小成对应关系。反射光线被光笔中的光敏元件接收并转换成模拟电信号,经放大电路放大后,送至比较器,由比较器产生相应的窄脉冲输出,比窄脉冲信号送到译码器。而译码器是由微处理器及其它控制芯片组成,其主要功能是将条形码读入设备送来的条形码电信号进行译码,变成计算机可识别的 ASCII 码,这样就可将 ASCII 码直接通过计算机键盘的输出口送至计算机,利用已编好的计算机应用程序,完成对数据的录入、统计、管理、控制等功能。必要时,计算机将发出相应的控制信号,经数模转换→放大驱动后,控制机械手操作相应的机械,从而达到生产过程自动控制的目的。

综上所述,条形码技术是在计算机应用的实践中应运而生的一种非常适用的数据自动采集技术。近年来,它在实践中不断得到完善和发展。为了加速我国的对外经济贸易、促进我国经济的繁荣和发展,我们要大力推广使用条形码技术。

初级程序员级水平考试辅导讲座

BASIC 语言程序设计问答

李志刚 吕品

问题 1 什么是 BASIC 语言?

答:1965 年,由美国 Dartmouth 学院的 John G. Kemeny 和 Thomas E. Kurtz 两位教授以 FORTRAN 语言为蓝本,设计出适合于计算机程序编制人员使用的一种高级程序设计语言,这就是 BASIC 语言。它是 Beginner's ALL-purpose Symbolic Instruction Code(初学者通用符号指令代码)的缩写。在我国,BASIC 语言已是各类学习机、PC 机乃至高档微机上的宿主语言。由于 BASIC 语言具有易学易用的特点,已受到广大计算机用户的重视,并用它编制出涉及各个领域里大量的优秀应用软件,在计算机应用中发挥了重要作用。

问题 2 BASIC 语言有什么主要特点?

答: BASIC 语言是一种通用的面向算法的高级程序设计语言。它是一种适合于初学者使用而又实用的语言。BASIC 语言有以下几个主要特征:

1. BASIC 语言好学易懂。基本 BASIC 一共只有 17 种语句。基本 BASIC 命令和语句中使用的词以及运算符号与英语中使用的词以及数学中的符号差不多,因此比较直观、易于理解和记忆。

2. BASIC 语言是会话式语言。当 BASIC 命令和程序输入计算机后,计算机立即进行语法检查,并在出错时给出提示。用户可在计算机上边运行边修改,直至程序正确为止。

3. 一般 BASIC 语言有立即执行方式。即从键盘输入语句后,立即给出结果,给检查和调试程序带来很大方便。

4. BASIC 语言系统虽小但功能齐全。除科学计算外有一定的文字、图形和音响的处理能力。可用来编制中小规模的应用程序和进行各种辅助设计、辅助教学以及游戏软件的编制等。

5. BASIC 语言是小型计算机语言,它允许的数的范围、变量个数、数据类型、子程序调用等都有一定限制。大部分 BASIC 是解释型的,程序运行速度较慢。

问题 3 什么是 BASIC 程序? BASIC 程序的逻辑行与物理行是指什么?

答:一个 BASIC 程序是由若干个程序行组成。每个程序行由行号、语句组、行结束符三部分组成。BASIC 规定每个程序行中的最大字符个数(含空格) ≤ 255 。

行号是程序行的标志,写在每个程序行之首。BASIC 程序是按行号升序次序在内存存储器中顺序存放,

并按此顺序执行程序。行号不仅指示了程序的执行次序,同时也可由行号改变执行顺序。PC BASIC 中的行号允许使用 0~65529 之间的整数。在程序中行号可以不连续(间隔 10 为好),在输入程序时行号次序不限,删改某行时,其它行不受影响。

语句组是程序行的核心,它规定了计算机在执行该程序行时应做的操作。当语句组中只含一条语句时该行程序称为简单程序行,否则称为复合语句程序行。复合语句行中语句组内各语句之间用冒号分隔。每个语句由语句定义符和语句体两部分组成。语句定义符是指定该语句执行何种功能的动词,又称为 BASIC 的系统保留字,它位于语句体之前。语句体是该语句所执行操作的对象。语句定义符和语句体的书写格式由语言本身做出规定,BASIC 语言允许空语句体存在。

行结束符是通过打一个回车键来产生的。输入编辑某一程序行结束时必须按回车键。在书写程序时一般可用“/”表示回车键,通常可省略不写。

一个完整的程序一般由输入数据部分,计算处理部分,输出结果部分和结束部分组成。

物理行是指显示屏上的一行或打印机上的一行。逻辑行是指 BASIC 程序的一个完整程序行,它是由行号、语句组和行结束符构成。由于每个程序行中字符个数最大可达 255 个,因此一个逻辑行就可能对应一个或多个物理行。

问题 4 有几种提供数据的语句?它们的主要用途各是什么?

BASIC 语言中主要有三种提供数据的语句,即 LET、INPUT、和 READ 语句。

LET 语句主要用于表达式的计算及少数变量的赋值。

INPUT 语句主要用来在程序运行中从键盘给少量需改变值的参变量赋值。

READ 语句主要用于对较多变量的已知数据进行赋值。

问题 5 BASIC 语言中常用函数有几类?用户怎样引用?

答:BASIC 语言中的函数分为标准库函数和用自定义函数两类。标准函数主要分为数值函数、字符串函数、输入输出函数及其它辅助函数。标准函数库是 BASIC 语言系统为完成特定的功能而提供的一组标准子程序。它们包含在 BASIC 语言中,并由该语言本身

规定每个标准函数的名字,用户在使用时只需写出它的函数名和自变量名就可以直接引用。在表达式中它们可象变量一样参与运算,但不能对其赋值。如要将 $X+Y$ 的平方根赋值给变量 A,只要写出 $LET A=SQR(X+Y)$ 即可。

自定义函数是由用户为解决自己的特定问题而定义的非标准函数。自定义函数在程序中一经定义后,就同标准函数一样可在表达式中引用。

上述两类函数都可被程序多次调用,不同处在于:标准库函数独立于用户程序。自定义函数则包含在用户程序中,由用户自己编写和定义。

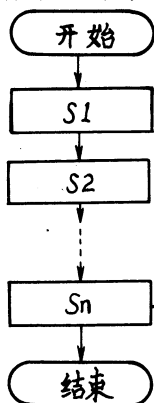
问题 6 BASIC 程序的基本控制结构是什么? 由哪些语句实现?

答: BASIC 程序的基本控制结构有三种: 顺序结构、选择结构和循环结构。

1. 顺序结构

顺序结构是程序的基本结构。也是最简单的结构。在此结构中,程序按行号排列的先后次序执行,其流程图如图一所示。

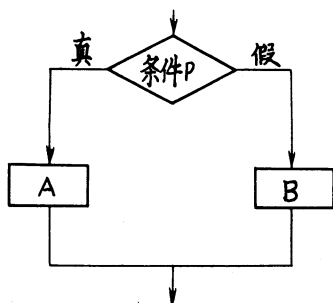
其中 S_1, S_2, \dots, S_n 为 n 个程序行。只要确定好各程序行执行的先后顺序就可较容易地完成程序设计。



图一、顺序结构

2. 选择结构

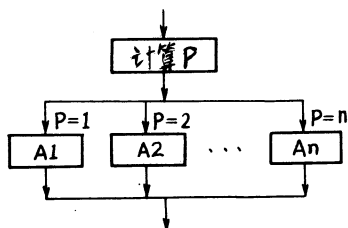
选择结构又称分支结构,它是按一定的条件在多个可能的计算或处理步骤中选取其中之一去执行。最简单的选择结构如图二所示。当条件 P 成立时(为真),执行 A 框,否则执行 B 框。一般情况下,选择结构可以嵌套或为多路选择,如图三所示。



图二、选择结构

3. 循环结构

循环结构又称重复结构。在程序设计中需要连续重复执行某一过程时,大多采用循环结构。常见的循环结构有三种类型:

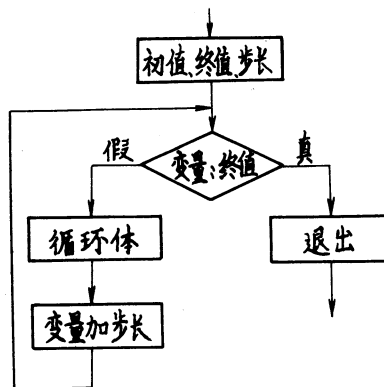


图三、多路选择

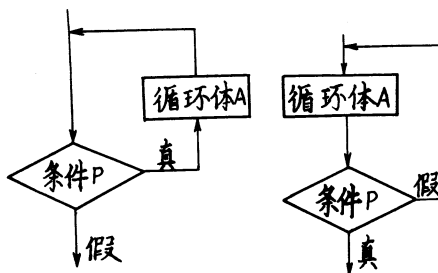
a. 计数循环: 可用 FOR/NEXT 语句实现,一般用于已知循环的次数、循环的初值、终值和步长的情形,其工作流程图如图四所示。

b. 当型循环结构: 可用 WHILE/WEND 语句实现,一般用于不知道循环次数的情况,只有在某个条件不满足时退出循环。其工作流程图如图五所示。

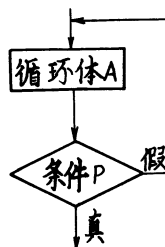
c. 直到型循环结构: 可用 IF/THEN 语句实现,一般也是用于不知道循环次数的情况,在条件满足时退出循环。其工作流程图如图六所示。



图四 计数循环



图五 当型循环



图六 直到型循环

从流程图可看出,当型循环与直到型循环都是在一定条件下重复执行循环体 A,所不同的是,当型循环判断条件满足时,执行循环体 A,然后再次判断条件是否满足,以决定是继续执行循环体 A,还是退出循环。而直到型循环先执行一次循环体 A 再判断条件是否满足,以决定是重复执行循环体 A,还是退出循环。

通常这三种循环结构之间是可以互相转换的,例如用三种结构所编的求 $\sum_{n=1}^{100} \frac{1}{n(n+1)}$ BASIC 程序,其繁简程度不一。在程序设计中采用哪种循环结构,要视题目要求及已知条件而定,不能一概而论。

(1) 计数循环

```
10 S=0
20 FOR K=1 TO 100
30 S=S+1/(K×K+K)
40 NEXT K
50 PRINT "S=";S
60 END
```

(2) 当型循环

```
10 S=0;K=1
20 WHILE K<=100
30 S=S+1/(K×K+K)
40 K=K+1
50 WEND
60 PRINT "S=";S
70 END
```

(3) 直到型循环

```
10 S=0;K=1
20 S=S+1/(K×K+K)
30 K=K+1
40 IF K<=100 THEN 20
50 PRINT "S=";S
60 END
```

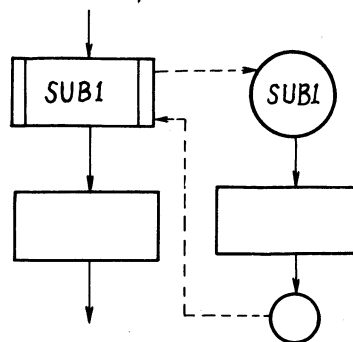
在 BASIC 中,由于上述三种结构都只有一个入口和一个出口,结构易于掌握,同时用它们又可以组成其它各种各样的程序结构,因此,熟练地掌握这三种基本结构,才能用 BASIC 语言编制出质量较高的程序。

4. 子程序结构

BASIC 还为程序设计者提供了子程序这一重要的程序结构,如图七所示。在程序设计中可将某些常用语句序列组合成一段程序,作一种特定的处理。该段程序就称为子程序段。如图中 SUB1 是子程序段。在主程序中需要用该特定处理功能时,就由主程序来调用,完成处理后又回到主程序调用处的下一条语句继续往下执行。从而大大地提高了程序编制的效率,减少了多余的程序段,提高了程序的可读性和可移植性。

问题 7 FOR/NEXT 与 WHILE/WEND 有什么区别?

答:FOR/NEXT 语句通常称为计数循环,而 WHILE/WEND 称为条件循环,也称当型循环。



图七 子程序结构

1. 按给定的次数重复执行一系列语句,这类循环称为计数循环。一般用于循环次数、循环控制变量的初值、终值和步长已知的情形。在 BASIC 语言中,计数循环一般是用 FOR/NEXT 语句来实现的。它的格式如下:

```
FOR 控制变量=X TO Y STEP Z
循环体
NEXT 控制变量
```

其中 X 是一个算术表达式,它产生循环控制变量的初值。Y 是算术表达式,是循环控制变量的终值。Z 是做控制变量增量的算术表达式,也叫步长。

这种递增(减)计数循环,通常是达到某一个值时终止。如果 Z 的值是正(负)的,每执行一次循环体,控制变量值增(减)一个步长,直至大于(小于)终值为止。

有两类计数循环;一类是执行循环体之前先检测循环变量,条件不满足则退出循环。这叫做先判定后执行。另一类是先执行循环体后检测循环控制变量。叫做后续判定结构,此种情况循环体至少要执行一次。究竟是哪一类取决于 BASIC 的版本。

2. 多数情况下,有些问题的求解,只知道控制条件,事先又无法知道重复循环的次数的,此时可使用条件控制循环的当型结构 WHILE/WEND。它的格式如下:

```
WHILE 条件表达式
循环体
WEND
```

是先判定后执行类型。先检测条件表达式是否成立(真),如果成立则执行一次循环体,否则循环终止。

计数循环和条件循环两种方法在设计循环结构时均可替代使用。条件循环适用范围较广,是一种常用的循环结构。

问题 8 IF/THEN 语句与 FOR/NEXT 语句嵌套使用时,有什么规则?

答:1. 在 IF 语句中嵌套了 FOR/NEXT 语句。

IF 语句的格式是:IF 表达式 THEN 语句组 1 [ELSE 语句组 2]。必须注意,IF 语句只能出现在一个逻辑程序行中,不能分开写在一个以上的程序行中。如


```
110 IF A>B THEN PRINT A
120 ELSE PRINT B
```

是非法的,应该写成

```
110 IF A>B THEN PRINT A ELSE PRINT B
```

在语句组 1 和语句组 2 中可出现任意 BASIC 语句。当 FOR/NEXT 嵌套在 IF 语句中时,要注意不能将 IF 语句分割成几个程序行。必须保证在嵌入 FOR/NEXT 结构后,原 IF 结构仍在一个行号下。如

```
100 IF A>100 THEN FOR I=1 TO 20:A=A+I×
I:NEXT I
```

如果嵌入后程序行超长,可采用调子程序的方法解决。

2. 在 FOR/NEXT 中嵌套 IF/THEN 语句。

FOR/NEXT 结构是由 FOR 和 NEXT 语句为语句括号构成一个语句组或程序段。因此,一般对出现在 FOR/NEXT 结构中的 IF/THEN 语句没有特殊的限制。

问题 9 使用子程序有什么好处?

答: BASIC 语言提供了用 GOSUB/RETURN 语句调用程序内部子程序的方法。

内部子程序的编写方法是将程序中多处需要调用的一般程序提出,编写成一个公用子程序段。在主程序内的任何地方都可对它重复调用。大量使用子程序后会系统开销(进/出栈操作增加,程序执行速度下降)略有增加。然而这很小的代价换来了较大的好处,比如程序的结构化程序提高了。源程序代码量的减少也为程序换来了较大的工作空间。同时,程序结构清晰,不仅编程效率提高了,也改善了程序的可读性,为程序的维护和移植带来了很大的方便。

问题 10 下列程序段各完成什么功能?

(1) 10 INPUT "N=";N

20 IF N<=0 OR N>50 THEN 10.

(2) 10 S=0

20 FOR I=1 TO 10

30 READ A;S=S+A

40 NEXT I

50 DATA 5,7,8,3,1,6,12,23,35,22

(3) 10 FOR I=1 TO 4

20 FOR J=1 TO I

30 PRINT TAB(4*J+1);I*J;

40 NEXT J

50 PRINT

60 NEXT I

(4) 10 P=1

20 FOR I=1 TO 20

30 P=P*I

40 NEXT I

(5) 10 INPUT A\$

20 L=LEN(A\$);DIM C\$(L)

30 FOR I=1 TO L

40 C\$(I)=MID\$(A\$,I,1)

50 NEXT I

(6) 10 A=RND(1)*100

20 B=INT(RND(1)*90+10)

30 PRINT A,B

(7) 10 INPUT "N=";N

20 IF N<100 OR N>999 OR INT(N)<>N THEN
10

30 A=INT(N/100)

40 B=INT(N/10)-A*10

50 C=N-A*100-B*10

60 PRINT C;B;A

答: (1) 从键盘接收[0,50]之间的实数 N。

(2) 求十个数的和。

(3) 打印三角形数阵:

1

2 4

2 6 9

4 9 12 16

(4) 求 $1 \times 2 \times 3 \times \dots \times 10$ 的积(即求 10!)

(5) 把从键盘输入的字符串分离成单个字符,并存入数组 C\$。

(6) 产生一个(0,100)之间的随机数 A 和一个[10,99]之间的随机整数 B。

(7) 从键盘输入一个三位数,把这三位数字分离出来,并按逆序打印。

问题 11 下列程序中空白处应填入什么内容? 该程序是求下面二维数组中最大元素所在的行和列。

```
[ 1 2 4 5
  3 7 5 4
  2 6 2 3 ]
```

10 DIM _____

20 FOR I=1 TO _____

30 FOR J=1 TO _____

40 READ A(I,J)

50 NEXT _____

60 NEXT _____

70 FOR I=1 TO _____

80 FOR J=1 TO _____

90 IF MAX>A(I,J) THEN _____

100 MAX=_____ :II=I,JJ=J

110 NEXT J

120 NEXT I

130 PRINT II;JJ;MAX

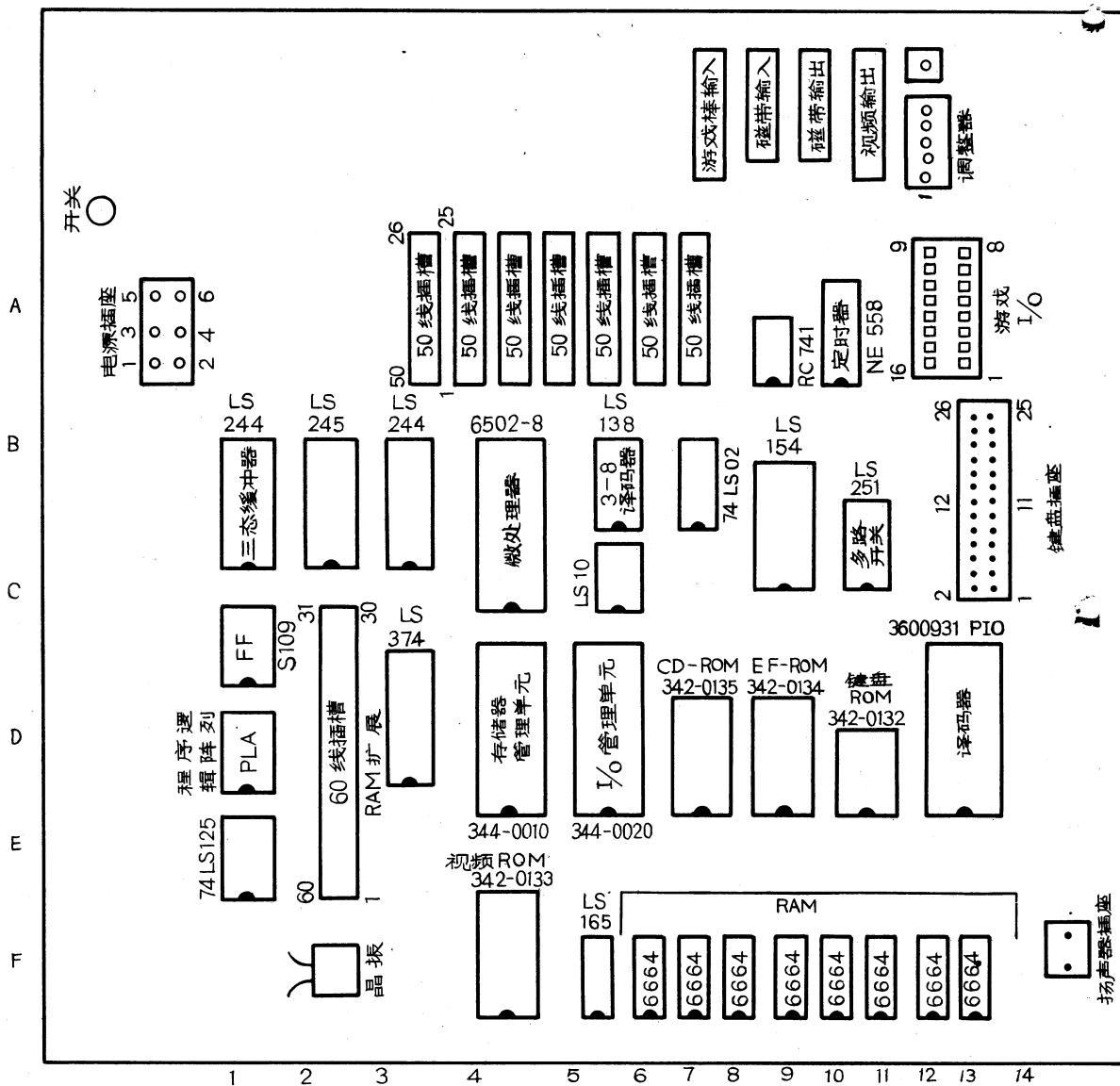
140 END

150 DATA 1,2,4,5,3,7,5,4,2,6,2,3

答: (按行号顺序) A(3,4)、J、NEXT I、4、3、110、A(I,J)

问题 12 下面程序是求 20 名同学每个同学的五门学科考试总成绩,程序有哪些错误? (转至 36 页下)

Apple IIe 主板元件排列图



把发展电子工业放到突出位置

中共中央关于制定国民经济和社会发展规划和“八五”计划的建议

电子工业是促进我国产业结构现代化的带头产业，未来十年要从投资分配、技术开发、设备更新、产业政策和组织管理等各个方面，为电子工业的迅速发展和推广应用创造条件。集中力量开发以大规模集成电路为中心、计算机为主体的投资类电子产品，大力加强微电子技术、计算机与软件、传感器的开

发和在国民经济中的普遍应用，积极利用电子技术改造传统产业，促进新兴产业的成长。努力发展光纤、卫星、微波等通信产品和消费类电子产品，以满足生产发展和人民生活提高的需要。

(转载《人民日报》1991年
1月29日头版头条)