



微小型计算机

# 开发与应用

MICRO—MINICOMPUTER  
DEVELOPMENT & APPLICATION



1990 5

微小型计算机开发与应用编辑部



微小型计算机开发与应用（公开发行）

---

编辑：《微小型计算机开发与应用》编辑部

出版：天津市电子计算机研究所

天津市电子计算机学会

地址：天津市河西区友谊路爱民道5号

邮政编码：300061

发行：天津市邮局

印刷：天津武清县长宏印刷厂

订购处：全国各地邮局

定价：0.95

---

邮局代号6—87

津工商广字0146号

国内统一刊号CN12—1122



## 《微小型计算机开发与应用》

### 编辑委员会

顾问	郭平欣
主编	黄侃
副主编	王治宝 邹秀凤
委员	(以下按姓氏笔划排列)
	于万源 于清汶
	王治宝 王 镭
	王士禧 王寿松
	付园明 许镇宇
	朱植松 曲庭维
	李凤祥 刘连棣
	陈力为 邹秀凤
	吴锦声 房家国
	夏纪寅 夏业勋
	袁维本 曹东启
	黄侃 黄宝良
	章谓臣 梅克定
	童宣明 裴少峰
	薛大中

## 1990年第5期目录

(总第49期)

### 应用实例

- 冲压用微机控制自动送料系统..... 储鸿庆(1)  
计算机系统仿真及应用实例..... 姚文庆(3)  
利用Apple-II开发MCS-51系列单片机  
..... 陈育华 宋宏远 杨天怡(8)  
8031单片机在智能微波癌症诊断仪上的应用  
..... 金 明 谢瑞和 向勋贤(11)  
简易微型打印机在单片机中的应用... 宋宝海(14)

### 计算机软件

- 在不同的显示卡上建立通用的汉字系统的技术  
..... 戴发金(18)  
新颖统计图形的设计及实现方法 .....  
..... 袁 泉 朱光俊 罗 炼(25)  
采用编码技术压缩数据库文件存储结构  
..... 刘长江(28)  
财务记账账页的压缩打印技术..... 杨云江(31)  
AOS/VS系统记录的使用..... 杨建农(32)  
用24针式打印机打印屏幕图形..... 刘 迅(36)  
IBM PC机除法溢出中断的修改..... 邢建春(38)

### 接口技术

- VAX11-DMF32并行接口的开发应用  
..... 周 鸿 刘志镜(42)  
IBM PC/XT与FWX-4675绘图仪的联接  
..... 梁卫平(44)

### 经验点滴

- 如何将反汇编出的程序写入文件..... 戴水贵(46)  
IBM PC/XT故障一例——ROM 芯片接触不良  
..... 高国明 龚安平(47)  
高速宽行打印机不走纸的检修..... 赵元哲(48)

# CONTENTS

## APPLICATION EXAMPLE

- Automatic Material Feeding System Controlled by Microcomputer in  
Punching.....Chu Hongqing (1)
- Computer System Simulation and Application..... Yao Wenqing (3)
- Developing MCS-51 Single Chip Microcomputer with Apple-II .....  
.....Chen Yuhua Song Hongyuan Yang Tianyi (8)
- Application of 8031 single chip Microcomputer in Intelligent Micr-  
owave Cancer Diagnostic Meter.....  
..... Jin Ming Xie Ruihe Xiang Xunxian (11)
- Application of Simple Microprinter in Single Chip Microcomputer  
.....Song Baohai (14)

## SOFTWARE

- The Technology of Designing the Common Chinese Processing Sy-  
stem in Different Kinds of Display Cards.....Dai Fajin (18)
- Designing and Implementing Method of Novel Statistical Graph.....  
..... Yuan Quan Zhu Guangjun Luo Lian (25)
- Using Coding Technology to Compress the Data File Storage Stru-  
cture.....Liu Changjiang (28)
- Compressed Printing Technology for Property Accounting Page.....  
..... Yang Yunjiang (31)
- AOS/VS System Log Function.....Yang Jiannong (32)
- Printing the Screen Graph Using 24-Pin Printer .....Liu Xun (36)
- Modification of Overflow Interrupt in Division on IBM PC.....  
.....Xing Jianchun (38)

## INTERFACE

- Development and Application of VAX11—DMF 32 Parallel Interface  
..... Zhou Hong Liu Zhijing (42)
- Connection between IBM-PC/XT and FWX-4675 Plotter.....  
..... Liang Weiping (44)

## EXPERIENCE

- How to write the Program Produced by Reverse Assemble into File  
.....Dai Shuigui (46)
- An Example of Fault in IBM PC/XT—The Contact of ROM Chip  
is Bad.....Gao Guoming Gong AnPing (47)
- Examing and Repairing the Feeding Paper Function of High Speed  
Line Printer When the Function Is Wrong... Zhao Yuanzhe (48)



# 冲压用微机控制自动送料系统

天津大学机械系 储鸿庆

**摘要** 本文介绍了一个用于冲床自动化送料的微机控制系统。机械部分采用步进电机驱动辊轮来传送带料。控制部分由TP801单板机及有关硬件组成,通过PIO接口实施输入与输出各种控制信息。所编制的专用控制软件可以保证系统灵活地适应各种不同送料步距要求,同时,能够根据不同料距适时自动调整加速启动和减速停止过程的控制。因此,这是一个通用的简易型自动送料系统,很适于在中小企业应用。

本系统是为了解决校办工厂设备的自动化送料问题而研制的。采用微机控制冲床送料的主要优点是:对各种不同送料步距等参数的变化适应能力强,软件稍加修改即可应用,这就简化了生产不同规格产品时的调试机械系统的工作。

## 一、机械部分动作原理

辊式送料机构动作原理如图1所示。步进电机通过齿轮传动驱动送料辊转动,钢板材料夹在两辊轮中间送进。当滑块下行至即将冲压前的某一位置时,打杆压住提升杆,将上辊抬起,此时模具对板材冲压加工。加工结束后滑块回程,打杆升高,提升杆复原,上辊在弹簧作用下下压,夹住板材。此后,步进电机再次启动,则送料辊又开始下一周期的送料工作。

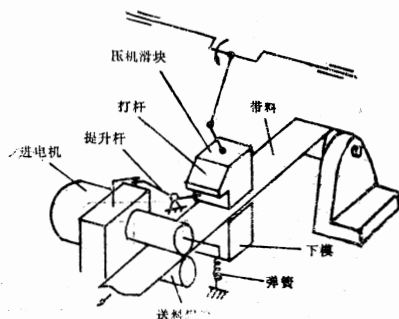


图1 辊式送料机构原理图

用单板机控制送料工作,主要是控制辊子与压机滑块间动作的协调,调节送料步进

电机启停时的加速、减速过程,并实现由用户现场键入当时工序的送料步距,由程序自动决定并控制电机旋转角度及动作频率。

## 二、控制部分的硬件设置

控制部分的硬件主要是TP801B型单板计算机,经其上配置的PIO外设端口与外界系统的输入输出部件相连接的。整个系统硬件配置框图见图2。PIO端口接到光电耦合接口电路后,再经晶体管放大电路及末级功率放大电路后驱动步进电机工作。所用步进电机以三相六拍方式运转,利用软件实现环形脉冲分配器的功能。实际上本系统仅用到了PIO B口的PB<sub>0</sub>, PB<sub>1</sub>和PB<sub>2</sub>分别通过接口与步进电机的A, B, C三相引线端相接, A口的PA<sub>7</sub>端作为压机反馈信息的接收口。

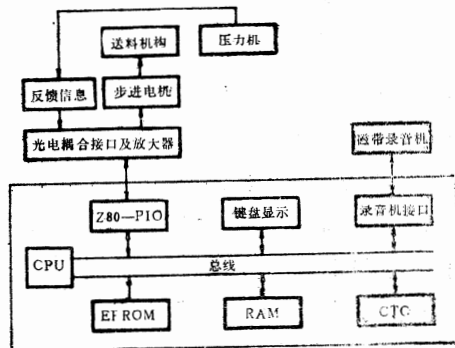


图2 系统硬件组成

## 三、送料速度的调节控制

步进电机启动时若没有逐步加速过程可

能会引起“失步”现象，而在电机停止时若不考虑逐步降速过程则又可能出现“超程”问题。为使本系统在不同送料步距的情况下均能实现“充分”的加速和减速，在软件中设置了送料速度调节控制功能。

对步进电机的起动、停止时速度的控制，实质上就是对控制脉冲频率的控制，本系统设计的电机间歇动作时控制频率曲线如图3。这样的频率变化规律完全由CTC定时器的时间常数控制。改变了CTC的时间常数，就改变了步进电机的转速。本系统的软件应用CTC中断方式2来产生定时脉冲，时间常数 $T_c$ 和脉冲频率 $f$ 的关系如下式：

$$1/f = t_c \times p \times T_c \times 10^{-6} \quad (1/s)$$

式中， $t_c$ 为时钟 $\phi$ 的周期；

$p$ 为定标系数（16或256）。

如果在起动或停止时，在一定范围内连续改变时间常数值，即可实现加速或减速控制。

为了实现速度的自动控制，我们对系统中料距参数与脉冲数 $n$ ，步距与时间常数间的关系作了推导。

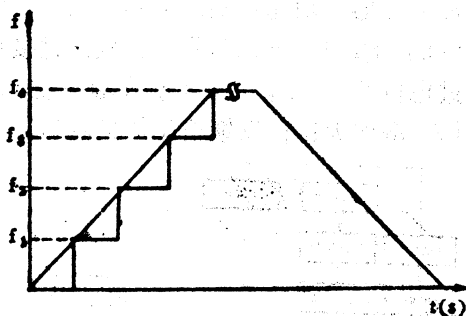


图3 步进电机运行速度曲线

结合实际运行条件，对本系统每次电机动作周期的时间作如下分割：电机启动到正常运行所用时间和从正常运行到停止所用时间相等，各占整个送料周期的2/5，而整个送料动作时间约等于滑块往复周期的1/2，启动和停止阶段各均分为4等分时间，各形成4个速度台阶，其中每一台阶对应着一个特定的脉冲频率和时间常数。每变动一个台阶，就改变一次脉冲频率和时间常数。

由实际设计所定的电机步距角 $\theta$ ，辊子半径 $R$ ，可推得脉冲数 $n = b/\theta \cdot \frac{\pi}{180} \cdot R = b/0.75 \times \frac{\pi}{180} \times 30 = 2.55b$ ，其中 $b$ 为送料步距。

由此，最后推出各台阶脉冲数与步距关系为：

$$n_1 = 0.11b, n_2 = 0.22b,$$

$$n_3 = 0.33b, n_4 = 1.23b.$$

进一步推出时间常数与步距间关系为：

$$T_{c1} = 3679.24/b, T_{c2} = 1839.62/b,$$

$$T_{c3} = 1226.42/b, T_{c4} = 917.65/b,$$

有了这个关系，就可在程序中通过适时设置相应的CTC时间常数来控制电机送进的速度了。

## 四、软件设计

软件工作原理如下：若送料与滑块机构尚未启动，则CPU循环等待滑块下行信号。一旦滑块信号到达，则CPU等待CTC中断请求信号，而后执行中断服务程序，控制电机输出一步。此后CPU返回主程序，并检查脉冲计数器的值。只要脉冲计数器 $C$ 不为零，就继续等待中断信号。一旦 $C$ 为零则重新给 $C$ 赋值，同时，时间常数也被赋给新值。这样循环往复，直至 $B$ 计数器回零后才返回去等待滑块新的信号。图4和图5分别给出主程序框图和中断服务程序框图。

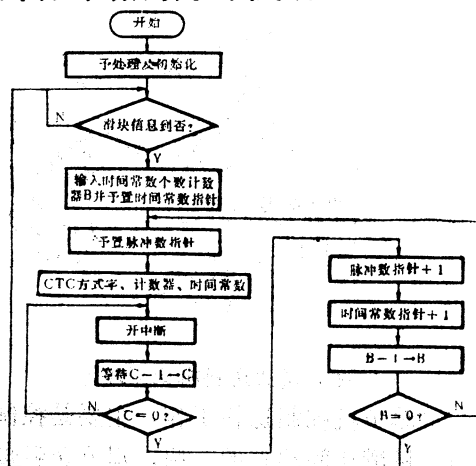


图4 主程序框图

# 计算机系统仿真及应用实例

天津大学计算机系

姚文庆

**摘要** 本文扼要论述了系统仿真计算机、仿真语言及仿真方法。并用GPSS通用系统仿真语言在微型计算机上实现了小型汽车生产装配线的仿真应用,从而获得了该生产装配线的性能评价,人员调度,设备配置及生产效益的优化方案。

## 一、前言

计算机系统仿真是计算机应用的一个十分重要的方面。是一门新兴学科。它在工程技术,企、事业管理,自动控制,社会经济及系统工程等领域中的作用愈来愈重要。这是因为,计算机系统仿真可以评价被仿真系统的性能指标;剖析系统各部分之间的关系和影响;比较各种设计方案,优化设计;对新建系统的理论进行检验;预测被仿真系统的投资、规模、人员调度、设备配置及经济效益等。

计算机系统仿真依据研究之目的及系统的物理活动,可把系统分为连续系统和离散系统两大类。与其对应的则有连续系统仿真和离散系统仿真两种类型。

## 二、仿真计算机与仿真语言

### 1. 仿真计算机

电子数字计算机是能够仿真任何已建或未来建的一切系统的强有力工具。并行硬件如流水线或阵列处理机都是高质量高效率的仿真计算机。因为仿真速度是一项重要因素,所以仿真往往强调并行性。并行性可通过并行机或分布式处理系统来实现。复杂的实时系统仿真可用特殊仿真硬件来完成。

传统的计算机仿真是采用单台大、中型机,一台计算机既要负责仿真计算,又要执行事件集处理。这不仅价格昂贵,而且使用不便,不利于普及应用。代之这种大型机仿真的是多微处理机或微型机组成的紧耦合或松耦合网络分布式处理系统。这是系统仿真

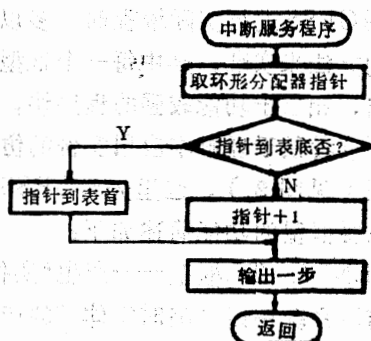


图5 中断服务程序框图

## 五、结论

本系统属简易型微机控制自动送料系统。它具有下述几方面优点:

1. 与原先的手工送料相比,实现自动送料可以极大地提高生产制件的效率,解放了人的双手,提高了安全性。

2. 比起手工操作来自动送料的步距精度高,送进均匀。在大批量生产的情况下,这将节约大量金属材料,提高经济效益。

3. 这个系统能够完成各种不同料距要求的自动送料工作,这就使之具有相当大的通用性,可以适应不同冲件尺寸类型的变化。

4. 所用硬件及其接口十分简单,投资费用小,易掌握,适合中小企业使用。

的重要发展方向。对于大量普及应用，特别是非实时仿真系统，则宜采用小型、微型计算机。这是系统仿真普及推广的重要途径。

## 2. 仿真语言的选择

计算机仿真最广泛使用的是高级语言 FORTRAN, PASCAL 及 PL/1 等；然而这些语言因不是专门用于仿真设计的，所以用它们书写的仿真程序往往很长。为此，相继发明了几十种专门用来编写仿真程序的语言。

用于连续系统仿真的语言主要有 IBM 1130—CSMP, IBMS/360—CSMP (Continuous System Modelling Program) 和 CSSL 等。IBM1130—CSMP 是面向框图的数字仿真程序，它适于小型计算机使用。S/360—CSMP 是面向方程的语言，它用 FORTRAN 的格式书写，但并不要求用户具备 FORTRAN 和数字计算机的广泛知识。CSSL (Continuous System Simulation Language) 是过程语言 FORTRAN—iV 的一种扩充，它的主要用途是对连续系统进行动态仿真。用于离散系统仿真的主要语言有 GPSS, SIMSCRIPT 和 SIMULA 等。其中 GPSS (General Purpose System Simulation) 通用系统仿真语言用得最广泛。用当前流行的 GPSS/360 或 GPSS—V 版本书写的较短程序也能仿真较复杂的系统。从这种意义上看，它的功能是很强的，使用灵活方便。不仅具有自输出能力，而且具有很好的诊断功能。

SIMSCRIPT 和 SIMULA 仿真语言也是广为使用的。它们既能以某些通用程序设计语言为基础进行程序设计，如 SIMSCRIPT 以 FORTRAN 为基础；SIMULA 以 PASCAL 或 ALGOL 为基础，也可以完全独立使用。一般来讲，GPSS 和 SIMSCRIPT 这两种语言，基本上能满足绝大多数用户对系统进行仿真的要求。

本文所研究的实例属于离散事件系统，采用的是 GPSS 通用系统仿真语言。

## 三、仿真方法

对一个系统进行仿真，首先要建立该系统的模型。在建模之前，通常先要定义被仿真系统的几个量：一是不可控变量；二是可控变量或决策变量。建立模型通常包括两项任务，一项是建立系统模型结构，另一项是提供与系统有关的数据信息。数据获得的方式可以是统计法，且要符合系统实体活动中各属性之间的近似关系式或函数表达式。其次要设计仿真程序，这先要提出符合被仿真系统的仿真算法。对于离散事件系统，一般算法流程图如图1所示。

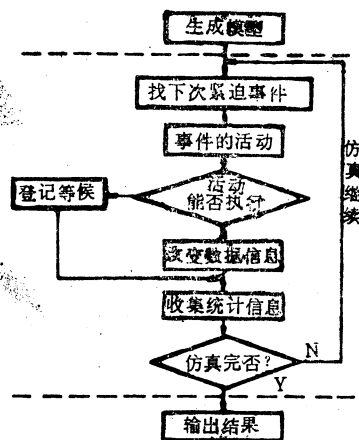


图1 离散系统仿真算法框图

采用 GPSS 进行实际编程时，多以框型流程图的方法来实现，其中每一个框型代表一个语句，相当于功能较强的程序块。

为了读者便于理解本应用实例的仿真程序流程图（见图2），这里先对 GPSS 语言的几个主要框型的功能简述如下：

1. GENERATE 框型——产生“实体”。负责被仿真系统中诸“临时实体”的产生。它有7个场的数字信息域，通过填写它们的参数，可以实现实体到达的模式及有关功能。

2. QUEUE 框型——排队。其功能是将



VE, 它代表实体接受服务完毕而出库。

本应用研究所用有关框型还有: ASSIGN——赋初始值, TABULATE——制表输出, VARIABLE——定义变量, FUNCTION——定义函数, LOGIC——逻辑开关, GATE——选通及 TEST——检测等。GPS S—V可供利用的基本框型有48个, 而微机用 GPSSR/PC可供使用的基本框型有38个。

#### 四、用 GPSS 进行小型汽车装 配线的仿真

我们观察发现，汽车装配线共分七个工位，从第一个工位开始，每当七个工位各自装配结束时，开始走线，汽车部件经一至七个工位后装成汽车。在装配过程中，第一、三、五号工位合用一部天车，其中第一号工位间断地使用三次，第三、五工位各使用一次。

装配线有大线和小线之分。大线包括装配底盘、车桥、车轮等大部件；小线主要安装车楼内部仪器仪表等配件。依据实际生产过程，小线工种必须在大线装配之前已备足一定数量安装好的配件。可以说，小线不会构成整体装配线的效率降低之因素。为此，对该系统仿真，我们只考虑大线，这是符合系统仿真建立模型之原则的。

由于该厂汽车装配主要靠人工操作,所以人的因素不能忽视。采集的有关数据信息只能取其平均值。仿真中对于实体的产生模式采用内扞值随机数产生法。我们把装配速度视为不可控变量;而把天车数量和工人数量视为决策变量,其仿真的目的和指标是如何较小的改变决策变量就能较大的提高装配效率,增加产品产量。

根据上述对系统的分析,我们设计的该汽车装配线的仿真程序流程如图2所示。不难看出,它是用表示为框型的指令来实现的。输入计算机则只给出代表该框型的指令

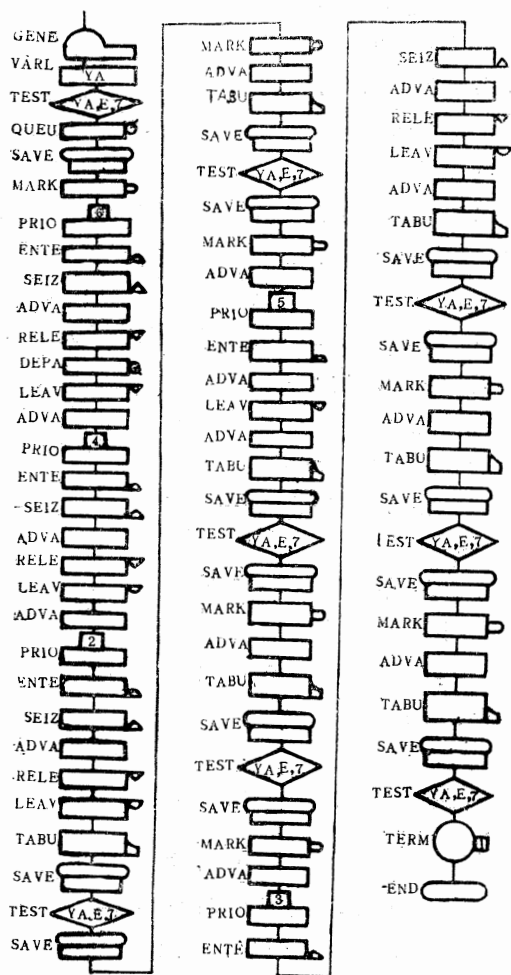


图2 仿真程序流程图

6. ENTER框型——入库。相同的“设备”或服务人数可用“仓库”来表示。它可定义容量。“仓库”有空间，实体可进入；否则，不能进入。与其成对的框型是LEA-

语句, 列出指令语句, 提供参数值后, GPSS处理程序能够把框型的全部信息存储起来。执行是通过解释实现的。当所用指令已写出并编译为有效方式时, 解释操作也就通过框型组合功能从头开始处理。虽然这种方式有碍编码效率的提高, 但其优点也较明显, 它允许框型快速汇编, 排除故障迅速。

在此还应对图2的流程说明以下几点:

1. 本程序流程包含七大块, 每一大块对应一个工位, 这样的结构显得简明清晰。

2. 由于第一、三、五工位使用天车, 故对使用天车的实体赋予不同的优先级别, 以确定其使用次序。

3. 由于现场生产线是先铺线, 即开工时使七个工位同时启动, 为此, 我们应首先产生七个实体作为铺线过程。

如前所述, 在七个工位同时完成后才能走线。这样就必须控制程序待七个工位中装配时间最长者完成后才能使实体前进。这通常可用控制变量进行控制, 然后再通过测试指令语句实现之。在GPSS中, 控制变量可以定义布尔变量, 通过测试布尔变量来实现。然而在本应用实例中, 由于七个实体同时进行, 使用布尔变量控制是不可能的。还因为本语言规定: 只当实体通过逻辑语句 (LOGIC BLOCK) 时, 才能给实体设置布尔量。因此, 其它工位的测试语句也只能对本工位的实体进行测试, 不可能涉及其它六个。面对这种问题, 我们解决的办法是, 用一个“全程”变量来实现控制。这个“全程”变量不受产生实体的约束。我们采用数值量进行控制, 其实现方法同样通过测试语句来实现。

## 五、输出结果及分析

仿真输出结果如表1所示, 还有若干统计数据未列入。

现在有必要就以下几点加以说明:

1. 设备利用率 (FACILITIES UTILIZATION) 我们知道, GPSS对仿真结果采取统计输出。鉴于本例的程序结构是七个实体同时运行, 故表1中给出的设备 HOOK (天车吊钩) 的利用率 (0.1416) 只是针对一个过程而言。由于整个仿真时间包括本程序内的延迟时间与执行此程序所花费的时间, 欲知HOOK的真正利用率, 即是说, 在某时间点上的实体从工位一至七的利用率, 则需进行下列简单计算, 已知每个实体平均使用HOOK 5次, 而从实体产生的达到模式可知, 每  $1080 \pm 120$  个时间单位产生一个实体, 其平均值为  $5 \times 202.15 / 1080 = 0.9359$ , 即利用率实为 93.59%。

2. 仓库 (STORAGES) 仓库用来修改吊车数量, 其目的有二, 一是为了提高装配效率, 二是为了减轻工人劳动强度。定义仓库名为 NUM, 容量为 2。平均容量为各间隔时间之和与仿真时间之比值, 其值是 0.20。由于一个实体从工位一至七需经过 5 次入库, 每进一次为一个实体, 20 个实体的总和为 100, 加上当前值 (等于 1), 总共是 101。间隔时间之和与总框数之比值为 233.68, 间隔时间之和占总仿真时间的比率为 0.102。

3. 队列 (QUEUE): 队列名为 1, 最大含量为 1, 这说明实体没有在队中等候, 平均容量近似为零。

4. JSZ=16: 代表一个作业班生产的汽车数量。

## 六、修改决策变量重新仿真

1. 改变天车数量为 2。我们已定义仓库的容量等于 2, 所以是胜任的。结果天车利用率下降近 50%, 但生产汽车数量仍为 16, 显然此举不可取。

2. 天车数量为 2, 在第一、三、五工位上各增加一名装配工人, 结果产生实体数量增加 (相应地对仓库、实体计数、队列、制表统计都有影响), 汽车产量增加 3 辆, 其

生产率提高18.7%。

3.如果在第一、三、五工位上再各增加一名工人,仿真结果指出,汽车产量并未增加。

由此我们可得出这样的结论,上述的第2项是比较理想的方案。

## 七、结束语

仿真本例的普遍意义在于:

1.微、小型机配上通用系统仿真语言诸

如GPSS等,对于一般工程系统和非工程系统均可适用,其应用范围是相当广泛的。

2.对于类似上述系统,通过仿真均可获得系统的性能,从而提供人员、设备等的优化配置,以期获得较高的经济效益。

3.对于未建系统的优化设计,其意义更加明显。

在本项仿真研究之前,王庆节同志在毕业设计时曾做过不少工作,对本文的完成很有帮助。

表1

FACILITIES							
--AVERAGE UTILIZATION--							
REFERENCE	# OF ENTRIES	AVERAGE TIME/TRAN	TOTAL AVAIL. TIME	UNAVIL. TIME	CURRENT STATUS	PERCENT AVAILABILITY	
HOOK	81	222.15	0.1416	0.1416	0.	A	100.00
STORAGES							
--AVERAGE UTILIZATION--							
REFERENCE	NUM CAPACITY	AVERAGE CONTENTS	ENTRIES	AVERAGE TIME/UNIT	TOTAL AVAIL. TIME	UNAVAIL. TIME	PERCENT AVAILABILITY
	2	0.20	101	233.68	0.102	0.102	0.
QUEUES							
REFERENCE	MAXIMUM CONTENTS	AVERAGE CONTENTS	TOTAL ENTRIES	ZERO ENTRIES	PERCENT ZEROS	TOTAL AVG. TIME/TRAN	
1#	1	0.	21	21	100.00	0.	
HALFWORD SAVEVALUES							
NUMBER-JSZ	CONTENTS	NUMBER-CBU	CONTENTS	NUMBER-CCU	CONTENTS	NUMBER-CDU	CONTENTS
CGU	1						

## 主要参考文献

1 B.P.Zeiglar, Theory of Modelling and Simulation, John Wiley & Sons Inc, 1976,

2 王正中,《系统仿真技术》,科学出版社, 1986.  
3 姚文庆,《计算机系统模拟》,天津大学, 1986.



# 利用Apple- I 开发MCS-51系列单片机

重庆大学

陈育华

宋宏远

杨天怡

**摘要** 本文着重论述如何在Apple-I上,把CP/M状态下汇编生成的MCS-51目标程序或者通过RS-232串行通讯接口传输给简易开发装置,或者直接通过写入卡固化在EPROM中,再插入单片机系统进行运行调试,以提高开发效率。

## 一、前言

在工业测控领域中,单片机系统越来越得到广泛应用。由于单片机不具备编程调试功能,在研制用户系统时,要借助相应的微机开发系统。本文介绍的开发系统如图1所示。

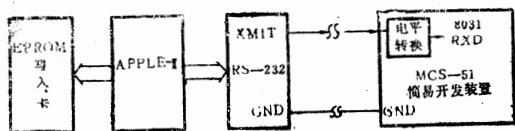


图1

## 二、把目标程序读入内存

Apple-I CP/M操作系统功能强,资源丰富,为对MCS-51汇编语言编写的源程序进行编辑和交叉汇编创造了良好的条件,而且实现起来较为便利。我们借用CP/M的ED.COM编辑程序输入和修改51系列单片机源程序。用8080机器语言研制出了51系列单片机汇编程序MCS51.COM,它能对MCS-51单片机源程序进行自动汇编,生成PRN列表文件和COM可执行的目标文件。源程序可完全按照MCS-51汇编语言的格式编写;有伪指令功能,能使用符号地址;对相对偏移量能自动计算并自动填入指令中;如源程序有语法错误,则全部汇编结束后,输出错误表和出错指针,供用户修改。

经编辑、汇编后生成的COM文件就是要传送到简易开发装置或固化在EPROM中的目标程序。为此,我们采用CP/M操作系

统调用技术先将COM文件从磁盘上全部读入内存。系统调用的公共入口地址为0005H,调用命令的编号送入C寄存器,调用命令所需要的参数送入相应的寄存器。

将目标程序读入指定内存区域的流程如图2(a)所示。图2(b)为写EPROM和数据传输的处理程序。

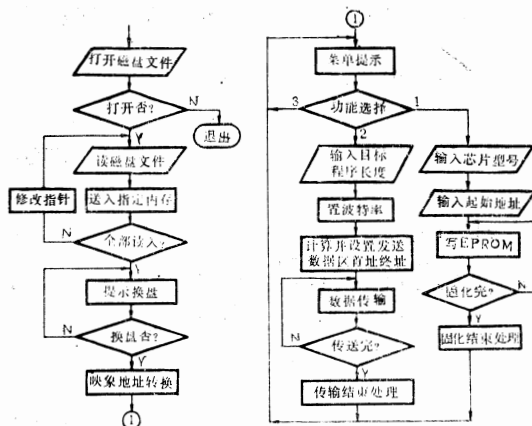


图2(a)

图2(b)

用到的系统调用命令有:

### 1. 打开文件命令15

```
LD C, 15
LD DE, FCB
CALL 5
```

### 2. 读文件命令20

```
LD C, 20
LD DE, FCB
CALL 5
```

文件控制块FCB的首地址装入DE寄存器对,为打开文件或读文件做好准备。文件控制块向系统调用指定被打开或读操作的文件名、扩展名及其他有关参数。打开文件调

用返回后,如A累加器的内容为FFH,说明文件不在磁盘,无法打开。否则可以读文件,每次系统调用只从磁盘上读回一个扇区128个字节数据,存放在0080H起始的内存缓冲区。可用数据块传送指令LDIR将它们移至内存中指定的区域。读盘调用返回时,A累加器的内容一般为零,此时,修改内存数据存放指针,继续读盘。直到A累加器的内容不为零,表明已读完文件的最后一个扇区。至此,把目标文件读入指定内存区域的过程结束,应该转入程序的第二部分。

### 三、Z80映象地址与6502映象地址的转换

用上述方法读入的数据是存放在对应Z80映象地址的内存当中,而专为Apple-Ⅰ的Apple soft设计的RS-232卡和EPROM写入卡只能工作在6502的监控状态。6502状态下的内存地址与Z80的映象地址是不同的。当Z80地址为0000H~AFFFH时,其映象关系是:

$$6502地址 = Z80地址 + 1000H$$

用下述程序段可实现Z80映象区域到6502映象区域的转换,而又不使已存在指定区域的目标程序受到破坏。

```
LD HL, C600H
LD (F3DOH), A
LD HL, (F3DEH)
JP 000BH
```

转换后,系统进入Apple soft状态。因此,有关将目标程序从指定的内存区域中经RS-232传送给简易开发装置的数据传输处理以及进入写EPROM状态的处理程序可用BASIC语言编写,并以文件的形式存入专用磁盘中。流程图如图2(b)所示。

目标程序从磁盘被读入到指定内存后,在终端上提示信息,告诉使用者“插入专用磁盘”,换盘工作完毕后,按提示回答操作

就可实现Z80到6502映象地址的转换。

在读入、转换过程中,利用CP/M系统调用命令1和命令9可实现人机对话。命令1等待用户从终端输入一个字符到A累加器中,它不需要入口参数。而命令9则是从终端输出一个以ASCII字符“\$”结束的字符串,字符串首址应预先放在DE寄存器中。读入、转换程序用Z80语言编写,并命名为RT.COM,它和ED.COM编辑文件及MCS51.COM汇编文件一起存入系统盘中。

### 四、数据传输与固化 EPROM

Apple-Ⅰ在CPU未变的情况下可以配上扩展卡来扩大其功能。用RS-232串行通讯卡与简易开发装置联机以及使用EPROM写入卡,就可利用Apple-Ⅰ原有的系统功能,构成简便高效的开发系统。EPROM写入卡和RS-232通讯卡上均有一片EPROM 2716,固化着相应接口的管理程序,支持板上的逻辑线路。当有关准备工作做好后,呼叫选通相应插槽上的接口卡,控制权就移交给相应接口管理程序。把目标程序或者按照EPROM的工作过程逐个字节地写入EPROM芯片中,或者按照设定的波特率及起止地址等传输给简易开发装置。

51单片机有一个全双工串行口,在简易开发装置上经电平转换后能和RS-232标准接口通讯。图1中,Apple-Ⅰ把目标程序经RS-232卡向简易开发装置的传送是无差错控制的单向数据传输。其信息格式为10位,无奇偶校验,波特率可在110~1200bit/s之间选择。波特率与RS-232卡上组合开关的状态、Apple-Ⅰ通过命令POKE 1144+S, r设置的数(Quantum)以及8031定时器1等参数的关系如表1所示。(其中,S为RS-232卡插入Apple-Ⅰ的插槽号)。

Apple-Ⅰ根据用户从终端输入的目标程

表1

波特率			110	300	600	900	1200													
Apple-I (RS-232)	开关状态	1	ON	ON	/	/	OFF													
		2	ON	OFF	/	/	OFF													
		3	ON	ON	/	/	ON													
	Quantum No.	Hex	BO	40	20	15	10													
		Dec	176	64	32	21	16													
8031 (串行方式1)	foc (MHz)	6	12	6	12	6	12	6	12											
	SMOD	0	1	0	1	0	1	0	1	0	1	0	1							
	定时器1方式2 定时常数TH1	72	/	/	/	CC	98	98	30	E6	CC	CC	98	EF	DE	DE	BC	F3	E6	E6

序的长度,计算出待传送数据区的终止地址,选通RS-232卡进行数据块传输。MCS-51简易开发装置在串行口方式1中断服务程序中,通过读SBUF寄存器的内容,来接收Apple-I发送的数据。有关数据传输的进一步讨论请参阅文献[1]。

## 五、MCS-51单片机的开发

将编辑、汇编、通讯和写EPROM等过程联系起来,MCS-51单片机的开发过程可归纳如下。

假定已完成下述准备工作:

1. Apple-I、RS-232卡、EPROM写入卡与MCS-51简易开发装置已按图1所示连接正确(如,RS-232卡、EPROM卡分别插入2、3号插槽),然后开机。

2. 准备好系统盘和专用盘。系统盘是在CP/M操作系统支持下运行的,包括ED.COM编辑文件、MCS51.COM汇编文件和RT.COM读盘转换文件。其中MCS51.COM和RT.COM文件是自行研制的。系统盘也可用作工作盘。专用盘存放的是按照图2

(b)编写的服务于写EPROM和数据传输的处理程序。

于是,整个开发步骤为:

(1) 在Apple-I上将系统盘插入A驱动器,B驱动器可插入另外一张工作盘。用ED.COM文件建立、编辑、修改源文件file.ASM。文件名file由1~8个可打印的ASCII字符组成,扩展名约定为ASM。

格式 > A ED d; file.ASM  
d为驱动器号

(2) 用MCS51.COM文件对源文件进行汇编,生成PRN文件和COM文件

格式 > A MCS51 file.abc

a、b、c为三个可选参数。a指出源文件所在的驱动器号,b指出生成的COM文件所在驱动器号,如为Z,表明将不生成COM文件,C指出生成的PRN文件所在的驱动器号,如选择X,表示仅在终端上显示出来,而不生成PRN文件。

(3) 用RT.COM文件,将目标文件file.COM读入内存(起始地址约定为1000



# 8031单片机在智能微波癌症诊断仪上的应用

华中理工大学

金 明

谢瑞和

向勋贤

**摘要** 本文介绍了智能微波诊断仪的设计原理与方法。

## 一、概述

目前,国际上用于肿瘤诊断的方法有很多,如X-CT, NMR-CT, 微波热像仪, 红外热像仪等, 华中理工大学在调研国外文献的基础上, 自行设计、研制出“S波段微波诊断仪”, 在全国十余家大医院进行了门诊临床诊断, 并于1988年通过了机电部主持的技术鉴定。某些肿瘤专家与电子学家一致认为本诊断仪在深部肿瘤的无损微波诊断方面

处于世界领先水平。

在“S波段微波诊断仪”的基础上, 我们采用8031单片机为微处理器, 把原来人工阅读亮温谱判据图的工作交给诊断仪自动完成, 通过LED显示出诊断结果, 并用打印机打印出一份病历报告, 实现了微波诊断仪的自动诊断。

## 二、诊断仪的主要特点

### 1. 工作频率为:

H)。读入后, 按照提示“INSERT THE DISK, THEN PRESS 'Y' ”将系统盘取出, 插入专用盘后, 按下“Y”键, 实现从Z80到6502映象地址的转换并进入如下菜单提示选择:

#### 1) WRITTING EPROM

(写EPROM)

#### 2) DATA TRANSMITTING)

(数据传输)

#### 3) QUIT (退出)

(4) 如选择1, 进入写EPROM状态。

按照屏幕提示, 输入EPROM芯片的型号、起始地址(2000H)等信息, 既把目标程序固化在EPROM中, 固化完毕, 如正确则提示“WRITTING OK!!”, 否则提示“WRITTING ERR!!”

(5) 如选择2, 屏幕上出现提示“LENGTH OF OBJ.PROG.L\$”, 等待用户从终端输入目标程序的长度。此时, 应先运行MCS-51简易开发装置上的接收程序(接收程序可固化在ROM中)。例如, 在DVCC-51-Ⅱ开发装置上, 接收程序的机器码为

2050H: 75 8D CA 74 20 F5 89 D2

AF C2AB D2 AC D2 8E 74 50

F5 98 75 82 00 75 83 21 80 FE

2023H: 40 04 D3 C2 98 32 E5 99

F0 A3 C2 98 32

本接收程序的波特率为300bit/s, 存放接收数据的起始地址为2100H。用命令2050 EXEC执行接收程序。

然后, 在Apple-Ⅱ上输入目标程序的长度(十六进制), 按回车键后, 进入通讯状态, 将目标程序传输给简易开发装置指定的内存区域中。当Apple-Ⅱ屏幕上出现“TRANSMITTING OK!”提示时, 传输结束, 用户可以在MCS-51开发装置上运行目标程序。

本文叙述的开发手段, 在本室应用一年多来, 收到了良好的效果, 为高效率开发研制单片机产品创造了条件。实践证明, 利用CP/M操作系统的资源和有关扩展接口卡与简易开发装置联机可构成高效的开发环境, 相应的硬件连接简单、可靠, 软件实施方便, 可移植性强。

$f = 2.25 \sim 2.65 \text{GHz}$ 。

2. 积分时间为  $0.8\text{s}$  时, 温度分辨率为  $\Delta T_{\text{min}} = 0.25\text{K}$ 。

3. 探头采用园极化波微型探头, 使用方便。

4. 探测人体深度达食管、鼻咽、肝脏等处。

5. 诊断仪具有自动诊断和灵敏度校验功能。

6. 通过LED可直观显示电压、 $M_1$ 、 $M_2$ 及最终诊断结果, 并配有微型打印机, 可打印一份完整的病历。

### 三、诊断仪的整机设计与结构

智能微波诊断仪由传感器电路、微波信息处理电路, A/D转换电路、微处理机及接口电路以及电源等五部分组成。如图1所示。

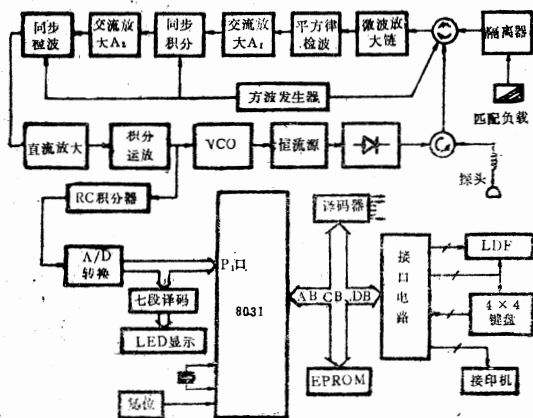


图1 智能微波诊断仪框图

#### 1. 传感器电路

传感器电路由探头、环形器、Dicke开关、匹配负载、微波放大链、微波检波等组成, 完成人体微波热辐射信息的提取功能。由于在微波频段人体的自发辐射能量密度约为  $10^{-10} \text{W} \cdot \text{m}^{-2} \cdot \text{Hz}^{-1} \cdot \text{Sr}^{-1}$  数量级, 接收系统

必须采用远比通讯、雷达设备要求高的低损耗、低噪声、宽频带的微波器件, 所用探头, 也必须满足一定的空间分辨率, 以达到最佳效果。

#### 2. 微波信息处理电路

由传感器电路提取出的人体微波热辐射信息, 经过平方律检波后进入交流放大、同步积分、同步检波, 直流放大、积分运算放大, 即可获得模拟电压信号。将此信号的输出直接送到压控振荡器VCO, 完成V-f变换, 再将所得的频率可调的脉冲信号去通截噪声源的恒流源开关, 完成频率~噪声功率的变换, 反馈到整机闭环的相加点。

同步积分、同步检波、Dicke环形器开关的同步信号均由方波发生器产生。

#### 3. A/D转换电路

模拟电压信号经过A/D转换, 得到数字信号, 然后送到微处理机进行运算、处理。

针对人体体温变化较慢这一特点, 我们选用5G14433做为A/D转换芯片。这是一种双积分式A/D转换芯片, 转换速度为4~5次/秒, 分辨率为2/1000, 上述特点均满足本系统设计要求。信号输出形式为分时输出4组BCD码, 其时序图如图2所示。

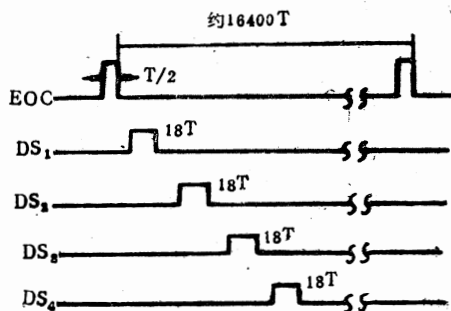


图2 5G14433时序图

为了直观地看到电压信号的变化, 将A/D的输出信号送到BCD-7段译码电路, 利用4只LED显示器, 就可显示其电压信号。

#### 4. 微处理机及接口电路

处理机部分是一个以8031单片机为中心

处理单元的专用微机系统,包括主机、接口电路及附属电路。如图3所示。

Intel公司的MCS-51系列单片机具有丰富的指令功能,并且其接口电路简单,易于开发,所以选用8031单片机为本诊断仪的微处理器。

时钟信号由一片6MHz的晶振产生。外部程序存储器选用4k字节的EPROM2732,接口电路由一片8155构成。8155芯片内部带有256字节的RAM,及A、B、C三个可编程的双向I/O口。显示器由4只LED构成,具有10个数字键、乳腺癌、鼻咽癌、食管癌和肝癌4个诊断功能键,一个置初值键,一个灵敏度校验键。

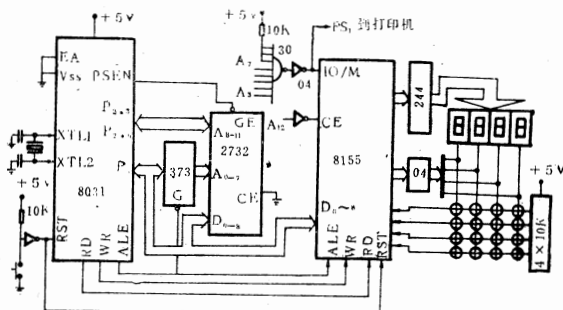


图3 微处理机电路

8155的片选信号由线译码电路产生。2732的片选信号直接接地。诊断仪地址分配如表1所示。

表1

	地 址 分 配
2732	0000H~0FFFH
8155	RAM 7000H~70FFH
	A口 70F9H
	B口 70FAH
	C口 70FBH
	控制口 70F8H
打印机	70FEH~70FFH

由上表可以看出,8155的A、B、C控制口及打印机地址均与8155内部RAM部分重合。这样做,可以简化硬件电路。

打印机的译码信号与8155相同,其地址

为70FEH~70FFH。除了打印机控制信号BUSY、 $\overline{ACK}$ 之外,开关量 $S_1$ (用于控制何时采集A/D转换数据)和DU-EOC信号也通过74LS244输入到8031。打印机选用TP- $\mu$ P16A型微型点阵式打印机。

## 四、软件系统

诊断仪全部软件采用MCS-51汇编语言,约占4k字节,固化在EPROM2732中。

监控管理程序是整个程序的核心,主要完成系统自检、初始化及对键盘、LED的管理。

6个功能键分别对应4个肿瘤诊断程序DIG1、DIG2、DIG3、DIG4,一个输入初值程序SOSU,一个灵敏度检验程序TEST。

整机系统灵敏度定义如下:

对应于校验温度 $T_1=30^\circ\text{C}$ ,采集30次,得到一组电压信号 $V_i$ , (其中 $i=1\cdots 30$ )

直方差

$$\delta = \frac{1}{30} \sum_{i=1}^{30} |V_i - \bar{V}|$$

(1)

通过体模试验测得系统斜率D,所以定义系统灵敏度:  $\Delta T_{min} = \delta \cdot D$  (2)

$\Delta T_{min}$ 愈小,说明系统灵敏度愈高,测量愈准确。

4个诊断程序功能几乎相同,包括A/D转换子程序,数制转换子程序、以及乘法、除法子程序,唯一不同之处就是根据设置不同的软标志,跳转到不同的二维判断子程序。

DIGi程序流程图如图4所示。

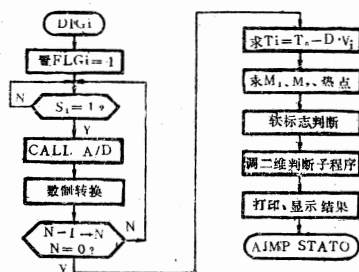


图4 DIGi程序流程



# 简易微型打印机在单片机中的应用

南京机械局职大

宋宝海

**摘要** 单片机配备CASIOHR-8计算器中的显示器和简易打印机后,系统便具有显示和纸带记录功能。本文重点介绍了这个接口的硬件电路和打印程序。

这样通过编制程序就能打印出单片机RAM中的内容。

## 一、打印机硬件接线图

### 1. CASIOHR-8键盘接线图

CASIOHR-8型计算器各按键在计算器内部按图1a所示矩阵排列,如果手动按下数字键“1”则行线 $X_2$ 和列线 $Y_4$ 连接,经计算器处理后,在显示屏上出现数字“1”,如果需要打印记录这个数字就按下P/#键。AC键是清零键。

如在单片机系统中,只需打印数字时,则键盘矩阵可简化成图1b,也就是从接口电路中向键盘矩阵引入四根行线,五根列线。

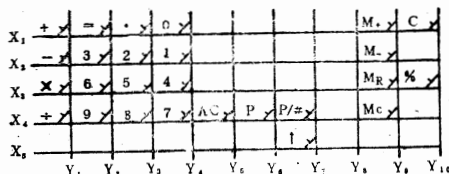


图1a CASIO HR-8 键盘矩阵

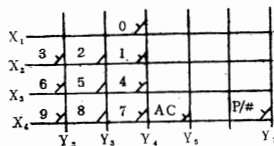


图1b 实用键盘矩阵

A/D子程序中,为了消除A/D转换带来的误差,可采用数字滤波技术,即对于某模拟电压 $V_i$ ,采集10次得到 $V_{i1}, V_{i2}, \dots, V_{i10}$ ,取其平均值得:

$$V_i = \frac{1}{10} \sum_{j=1}^{10} V_{ij} \dots \dots \dots (3)$$

可把 $V_i$ 做为模拟电压信号 $v_i$ 经过A/D转换后的结果。

## 五、结 论

智能微波癌症诊断仪的研究是生物电磁学、微波热辐射计量学、生物医学、生物电子学、计算机科学等多门交叉学科的研究成果,其深部肿瘤的检测性能已达到原发癌的普查与筛选实用水平,并超出国际上关于微波热像仪诊断性能的报导水平。

由于智能微波诊断仪对人体完全无损害,无任何微波辐射,诊断费用低,诊断结果准确,诊断食管癌的阳性符合率(TP值)达86%,阴性符合率(TN值)达81%,是

一种极易推广的用于肿瘤防治和筛选的有效手段,也可用于肿瘤治疗动力学连续监测,为治疗方案提供数字评估。目前,全国已有十余家大医院利用本诊断仪进行临床诊断。

### 参考文献

- 1 A.H.Barrett, P.C.Myers: "Detection of Breast cances by microwave Radiometry". Radio Science.1977.
- 2 P.C.myers, N.L.Sadowsky, A.H.Barrett: "Microwave Thermography: Principles, methods and clinical applications". Journal of microwave Power.14(2), 1979.
- 3 向勋贤,樊文泉等  
"生物热辐射信息提取与应用"  
微波, No.2(总字第9期),1987年
- 4 《MCS-51.96系列单片机原理及应用》  
北航出版社.徐爱卿等



### 三、打印程序框图

在单片机的实际应用中，如应用在压力容器产品检验装置中，检验结果打印为：

#9003140018

#1510161200

第一行表示5项内容分别是：90年3月14日，第001号产品，第8号检验员。第2行表示4项内容分别是：产品保压值为15kgf/cm<sup>2</sup>，保压时间是10分钟，实际压力是16kgf/cm<sup>2</sup>，实际时间是12分钟，00表示合格产品。这样打印以上结果的框图如图3所示。

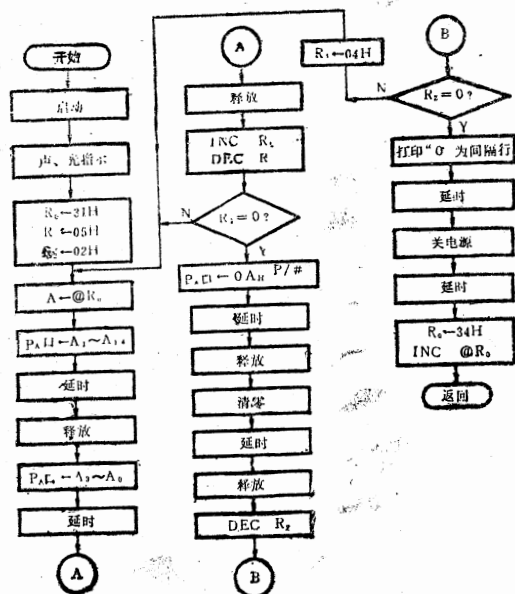


图3 打印程序图

其中：(1) R<sub>0</sub>是存放打印数值的缓冲区单元的首地址。R<sub>1</sub>是存放一行中所打印的项目数。R<sub>2</sub>是存放打印的行数。

(2) 在框图中，P<sub>1</sub>.0 ← 0AH 框的意义是按下打印键P/#，清零的意义是按下AC键。

(3) 单独调此打印程序时，首先要对8255芯片进行初始化，按以上框图编写的程序如下：

```
PRT: CLR A
      MOV DPTR, #2002 H
      MOV X @ DPTR, A
```

```
CLR P1.5
MOV A, 05 H
```

```
K0: CLR P1.2
     CLR P1.3
     LCALL MDELAY
     SETB P1.2
     SETB P1.2
     LCALL M DELAY
     CLR C
     SUBB A, #01
     JZ K1
     AJMP K0
```

```
K1: MOV R0, #31H
     MOV R2, #02H
     MOV R1, #05H
```

打印2行  
第一行有  
5项内容

```
K2: MOV A, @R0
     ANL A, #FOH
     SWAP A
     MOV DPTR, #2000H
           8255A口地址
     MOVX @DPTR, A
     LCALL DELAY 延时
     MOV A, 0CH
     MOVX @DPTR, A } 断开
     LCALL DELAY (释放该键)
```

```
MOV A, @R0
ANL A, 0FH 取低四位
MOVX @DPTR, A
LCALL DELAY
MOV A, #0CH
MOVX @DPTR, A } 断开
LCALL DELAY
INC R0
MOV A, R1
CLR C
SUBB A, #01
MOV R1, A
JZ K3
AJMP K2
```

```
K3: MOV A, #0AH
     MOVX @DPTR, A } 按下打印键
     LCALL MDELAY
     LCALL MDELAY
```

```

MOV    A, #0CH
MOVX   @DPTR, A } 释放该键
LCALL  DELAY

MOV    A, #0BH
MOVX   @DPTR, A } 按下AC键
LCALL  DELAY

MOV    A, #0CH
MOVX   @DPTR, A } 释放AC键
LCALL  DELAY

MOV    A, R2
CLR    C
SUBB   A, #01
MOV    R2, A
JZ     K4

MOV    R1, #04H    第二行
                     四项内容

AJMP   K2
K4    MOV    A, #0AH
        MOVX   @DPTR, A } 按下P/
        LCALL  MDELAY      键
        LCALL  MDELAY

```

```

SETB   P1.5    关打印机电源
LCALL  MDELAY

MOV    R0, #34H } 产品序号加1
INC     @R0
LJMP   STAR    } 返回主程序

MDELAY: MOV    R6, #06H
        LP3: MOV    R5, #FFH
        LP2: MOV    R4, #FFH RET
        LP1: DJNZ   R4, LP1
            DJNZ   R5, LP2
            DJNZ   R0, LP3

```

#### 四、结束语

经调试证明本电路简单、工作可靠、打印数字清晰。本文只应用了CASIOHR-8计算器的输出功能，如能进一步开发计算器的其他功能将使单片机得到更多地应用。

(上接35页)

```

fclose ( fuser );
fclose ( fd );
}

```

清单中的: util; cedors; user\_register.log是用户记录文件名。

以上介绍了如何起动系统记录和对系统记录文件中的各用户使用机器数据的处理。对于用户记录文件，管理员还应该进行适当的管理工作。当一个新的用户注册上机时管理员应将使用者的姓名、工作单位、通讯地址、电话、注册日期、联机用户名以及预交金额等基本情况记录在用户记录文件中，并对该文件进行如：增加，删除，修改，查看，用户结帐等维护。另外，为了计算用户使用机器的费用，还必须建立起一个记费标准文

件。该文件中记录联机时间、CPU时间、打印页数、I/O块数等的收费单价，根据这些单价可在结帐程序中使用一些算法对不同情况的用户进行优惠收费等处理。还可以根据情况对各项数据的单价进行适当修改和调整，这样便可根据单价和每个用户的用机记录数据结算出用户的费用。在用户每次下机时还可运行一个给出用户当天用机情况和累计用机情况的报告，提醒用户的各项开销情况和剩余金额。并对各用户的页-秒数与联机时间之比排出名次，用以考查各用户的用机效率。使用户利用尽可能少的联机时间来解决尽可能多的问题。用户在此监督下会自觉地努力提高自己的用机效率。使机器更有效的运转。



# 在不同的显示卡上建立通用的汉字系统的技术

苏州大学

戴发金

**摘要** 本文通过分析IBM-PC及兼容机上的四种不同的显示卡的异同点,说明了现有的汉字系统在 IBM-PC 兼容机之间不通用的原因,并对此问题提出了一种可行的解决方案,最后,给出了按此思想设计成功的“通用的汉字、图形仿真终端”中的有关程序清单。

## 一、问题的提出

众所周知,近些年来IBM-PC及其兼容机已广泛地应用于我国国民经济建设的各个领域。计算机的应用离不开汉字处理系统,所以,近些年来国内许多科研单位研制成了多种汉字处理系统,成功地为IBM-PC及其兼容机配上了汉字处理系统。但是,其中没有任何一个版本能够在IBM-PC及其多种兼容机之间做到完全通用的。因此,在多种兼容机上要分别开发多个版本,这给汉字系统的开发、维护和用户的使用都带来了一定的困难。究其原因,主要是IBM-PC及多种兼容机上配了多种不相兼容的显示卡,这些显示卡在图形方式下的初始化参数、显示缓冲区的地址,相邻二线、相邻二字符行之间的地址差都不相同,且其变化是无规律性的,而这些特性又与汉字处理系统密切相关,从而造成建立一个通用的汉字系统的困难。对这个问题的解决方法主要有以下两种:1.对每一种显示卡都搞一套汉字系统,这一方案就是本文前面所提到的,这也是目前IBM-PC上的汉字系统的现状。2.通过分析各种显示卡,找出其共性,对不同的都用一个特征变量表示。因此,汉字系统仅需设计一种,对可变的特性实际是访问其特征变量的当前值。在系统自举时由用户输入显示卡或计算机的型号(或由系统自动识别),然后系统将对显示卡的特征参数填入相应的特征变量中,从而实现了汉字系统的通用性。

下面就来介绍方案2的实现技术。

## 二、各种显示卡的异同点

同一型号的个人计算机所配的显示卡一般来说是一样的。因此,本文是以计算机的型号来代表显示卡的型号,这样做会有助于读者记忆。本文主要介绍8088、8086这两个档次的个人计算机上的显示卡,对80286、80386的各种兼容机上的各种显示卡可用同样的方法加以研究。

### 1. IBM-PC/XT, GW0520A, GPS0520

#### (1) 图形方式的初始化

这几种个人计算机的显示卡是兼容的,所有显示卡都可工作在字符、图形两种工作方式之一。汉字系统是以图形方式处理字符的,因此,首先必须将显示卡初始化成图形方式,这是通过将16个参数输出到CRT控制器6845的16个寄存器中来实现的,这16个寄存器是通过3D4H, 3D5H来间接访问的。同时还要对3D8H, 3D9H两寄存器进行初始化。这18个参数是: 38H, 28H, 2DH, 10, 7FH, 6, 64H, 70H, 2, 1, 6, 7, 0, 0, 0, 0, 1EH, 3FH。

(2) 显示缓冲区的地址与显示线的关系: 该显示卡的最高分辨率为640×200, 显示缓冲区的段地址为B800H, 显示缓冲区地址分成100组, 每组分两条线, 如图1所示。每组内的两条线之间的地址差为2000H, 相邻两组之间的地址差为0050H。

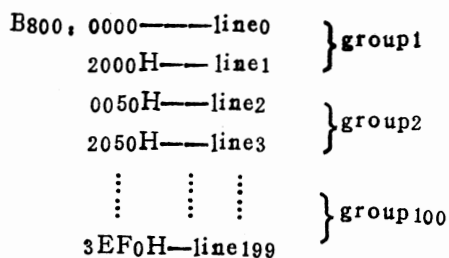


图1

## 2. GW0520C-H

### (1) 图形方式的初始化

GW0520C-H是国产的一个优选机种，出厂时已配有汉字系统，不过那是以字符方式处理汉字的。若要在该机上实现图形汉字系统或进行绘图，仍要将它初始化为图形方式，CRT-6845上的口地址仍为3D4H，3D5H，3D8H，3D9H。18个参数分别为：64H，50H，54H，3，1CH，1FH，1CH，1CH，2，11H，FFH，FFH，0，0，0，0，1EH，3FH。此外还要将3DDH初始化为3。

(2) 显示缓冲区地址与屏幕显示线的关系：

显示卡的最高分辨率为640×450，显示缓冲区段地址是B800H。显示缓冲区地址分为25组，每组含18条显示线，如图2所示。

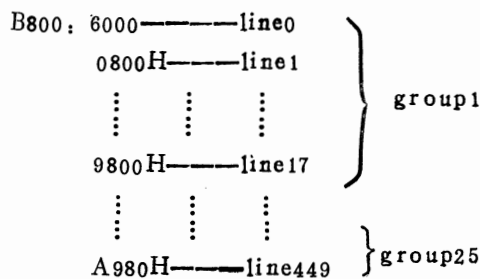


图2

其中每组内的两条相邻线的地址差为0800H，相邻两组之间的地址差为0050H。

## 3. M24

### (1) 图形方式的初始化

M24是加拿大生产的IBM-PC兼容机，它的cpu为8086，它在我国西南地区有广大的用户。CRT-6845的口地址为3D4H，3D5H，3D8H，3D9H。18个参数为：38H，28H，2DH，10，7FH，6，66H，70H，2，

1，6，7，0，0，0，0，1EH，3FH。此外还应将3DEH初始化为41H。

(2) 显示缓冲区地址与屏幕显示线的关系：

图形方式的最高分辨率为640×400，显示缓冲区的段地址为B800H。显示缓冲区地址分为100组，每组含4线，如图3所示。

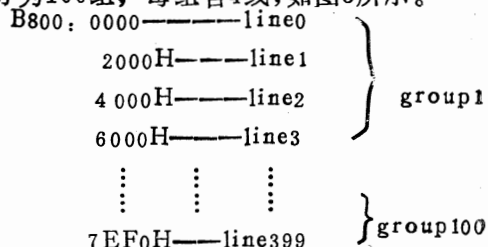


图3

其中每组内的相邻二线之间的地址差为2000H，相邻两组间的地址差为0050H。

## 4. HW-PC

### (1) 图形方式的初始化

HW-PC广泛用于我国的地矿、银行等系统，它的CRT-6845的口地址与M24相同，应将3DFH初始化为2。它的18个参数为：38H，28H，2DH，10，7FH，6，66H，70H，3，2，6，7，0，0，0，0，1EH，3FH。

(2) 显示缓冲区的地址与显示线的关系：

它的最高分辨率为640×400，显示缓冲区的地段地址为B800H，显示缓冲区地址分为100组，每组含4条线，如图4所示。

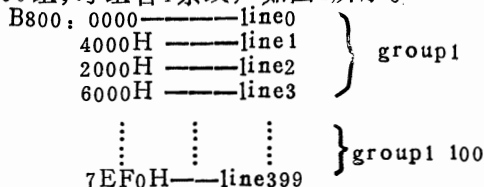


图4

其中，每组内的相邻二线的地址差是不定的。相邻两组间的地址差为0050H。

以上例举了目前国内用得较广的四种机型的4种不同显示卡的特性。从中可以看出，它们在图形方式的初始化参数、显示缓冲区的地址与显示线的对应关系等方面都有明显差别。

### 三、特征变量的选择

#### 1. CRT-R-Addr (字)

CRT-6845中的16个寄存器的间接访问寄存器首址为3D4H, 但80286、80386 机上也有采用3B4H或3C4H的。因此从发展的角度看, 也应为此设立一个特征变量、以贮存此首地址。

#### 2. CRT-R-ADDR1 (字)

除了前18个参数初始化的口地址可由首地址推得外, 以上各卡还要分别对3DEH, 3DFH, 3DDH进行初始化, 因此也要为此不同的口地址建立一个特征变量。

#### 3. LINE-ADDR (1KB线性表)

对汉字系统的通用性影响最大的是屏幕上各显示线与其对应的显示缓冲区地址之间没有一致的对应关系。这样, 汉字系统的显示子模块在将某字符的前一字节的点阵数据写入显示缓冲区后, 要经过多次比较才能得出下一字节点阵数据要写入的显示缓冲区的地址, 这会严重影响汉字系统的显示速度。因为, 显示子模块是汉字系统中执行频度最高的子模块。随着显示卡种类的增多, 显示速度将会变得更缓慢。本人经过一段时间的摸索认为: 对这一问题的较好的解决方法是: 将选定的显示卡的各显示线的地址按线号从小到大的顺序写入一个线性地址表中。显示子模块用显示线号, 就可以迅速地从此线性表中查出该显示线所对应的显示地址。对 $640 \times 450$ 的显示卡来说, 此线性表仅需占 $450 \times 2 = 900\text{kbs}$ 的内存, 这对占内存300余KB的汉字系统来说是微不足道的。但它却可使汉字系统做到很好的通用性。

#### 4. WINDOW-L (字节)

表示一屏所能显示的汉字的行数 ( $640 \times 200$ 的为10行), 这样汉字系统可以根据显示卡的分辨率的高低自动调节每屏所显示的行数。

#### 5. STATUS-S (字)

表示状态行的起始线号。

#### 6. ES-SEG (字)

表示显示缓冲区的段地址。目前市面所用的显示卡的地址有B800H, B000H, A000H, A800H等。

#### 7. ROW-LINE (字节)

每行汉字所占的显示线数。

目前市面上所用的显示卡对汉字系统的通用性有影响的所有特性都可以用以上的特征变量完全概括出来。因此, 对一个显示卡的与以上特征变量相对应的参数值, 可以与汉字系统源代码编辑在一起同时进行汇编编译形成目标代码, 也可以经过汉字系统提供的系统调用直接将参数值提供给汉字系统。这样, 这个汉字系统就可以应用于具有该显示卡的系统中。系统自举时会询问您是使用什么型号的机器 (或显示卡) 的。由此看来, 这样的汉字系统对各种不同显示卡具有很好的可扩充性。当然, 这里对汉字系统的要求是: 在其代码设计中, 凡涉及以上特性的, 都是访问其相应的特征变量的当前值。例如, 显示子模块就是以显示线号为索引, 从线性表 LINE-ADDR 中查找其对应的显示缓冲区的地址。这种技术的可行性及有效性已在本人设计的“通用的汉字、图形仿真终端”中得到了肯定的证实。

### 四、显示线地址的产生

由上述可知, 在汉字系统的装入、初始化过程中, 包括对线性地址表 LINE\_ADDR 的初始化, 也就是将当前选中的显示卡的各显示线所对应的显示缓冲区的地址依序填入该表。对 $640 \times 450$ 的卡来说, 其显示线地址占 $450 \times 2 = 900$ 字节, 若由人工从键盘输入, 并与汉字系统源代码一起编辑、同时汇编成目标码, 不但容易出错, 而且随着系统所包含的各类显示卡数量的增加, 需要占用一批存贮空间。由此看来, 宜在汉字系统初始化时由程序自动产生地址数据填充线性表 LINE\_ADDR。例如中子程序 SET\_LINE\_ADDR 就是为此目的而设计的。为实现这一

目的, 需要增加定义以下的特征变量:

#### 1.ADD\_LINE1(字)

组内偶数线到奇数线地址增量。

#### 2.ADD\_LINE2(字)

组内奇数线到偶数线地址增量。

以上两特征变量主要为具有隔行扫描功能的显示卡如 HW-PC 上的显示卡而定义的。

#### 3.GROUP(字节)

屏幕上显示线所对应的显示缓冲区地址的分组数。

#### 4.SUM\_LINE(字节)

每组内的线数。

### 五、显示卡参数与汉字系统的连接

对每种显示卡, 只要向汉字系统提供以上10个特征变量的16字节数据和CRT\_6845各寄存器的19字节的初始化参数, 该汉字系统就可以应用于使用该显示卡的计算机系统中。以上显示卡的参数与汉字系统的连接方法有二个: (1) 将参数与汉字系统源代码同时编辑、汇编(编译)形成目标代码。

(2) 对已形成了目标代码的汉字系统, 特别是对汉字系统的用户来说, 如何将新的显示卡的参数传递给系统呢? 为此, 系统提供

```
1: CRT_INIT          PROC          NEAR
2:                   ASSUME        CS: CODE, DS: CODE
3:                   JMP           BEGIN
4: TERM_MODE          DB           1; 0=GW_CH 1=PC/XT, GW_A 2=SUN_286 3=HW_
                                     PC 4=M24
5: LINE_ADDR          DB           450*2 DUP(0); STORE LINE ADDRESS
6: ES_SEG             DW           0; GRAPHIC DISPLAY MEMORY SEGMENT
7: CRT_R_ADDR         DW           0; CRT_6845 RIGISTOR INTERFACE ADDRESS
8: CRT_R_ADDR1        DW           0
9: STATUS_S           DW           0; THE IRST ROW OF STATUS LINE
10: ADD_LINE1         DW           0; INCREASMENT FROM EVEN LINE TO ODD LINE
11: ADD_LINE2         DW           0; INCREASMENT FROM ODD LINE TO EVEN LINE
12: WINDOW_L          DB           0; THE LENGTH OF WINDOW
13: ROW_LINE          DB           0; THE ROW NUMBERS PER CHARACTER LINE
14: GROUP             DB           00
15: SUM_LINE          DB           00
```

了一个软中断INT 10H, 入口参数是AH=50 DS:DX 以\$为结尾的显示卡名称的首址, DS:SI 35字节参数的首址。这样系统就可将新的显示卡的名称及其35字节的参数值填入汉字系统中预留的空间里, 并驻留系统盘上。

### 六、实 例

以下的8088汇编子程序CRT\_INIT是本人按此思想设计的“通用的汉字、图形仿真终端”中显示模块的初始化子程序。其中SET\_CRT\_PARA是初始化以上10个特征变量; SET\_LINE\_ADDR是初始化线性地址表 LINE\_ADDR; CRT\_R1\_16 是初始化 CRT\_6845上的 19 个寄存器。以上“通用的汉字、图形仿真终端”的显示模块可用图5表示。

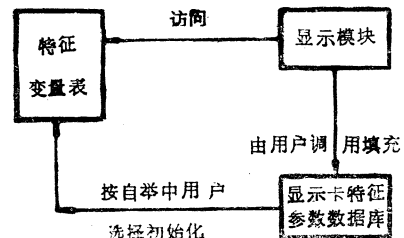


图5

```

16: VAR_NUM          DB      19+16,
17: ; -----GW0520C_H-----
18: CRT_DATA         DW      0B800H, 3D4H, 3DDH, 433, 0800H 0800H
19:                  DB      24, 18, 25, 18
20:                  DB      64H, 50H, 54H, 3, 1CH, 1FH, 1CH, 1CH, 2, 11H
21:                  DB      0FFH, 0FFH, 0, 0, 0, 0, 1EH, 3FH, 3
22: ; -----IBM-PC/XT GW0520A-----
23:                  DW      0B800H, 3D4H, 3DFH, +183, 2000H, 2000H
24:                  DB      10, 18, 100, 2
25:                  DB      38H, 28H, 2DH, 10, 7FH, 6, 64H, 70H, 2, 1, 6, 7, 0, 0
26:                  DB      0, 0, 1EH, 3FH, 0
27: ; -----SUN--286-----
28:                  DW      0B000H, 3B4H, 3EFH, 384, 2000H, 2000H
29:                  DB      24, 16, 100, 4
30:                  DB      35H, 28H, 2DH, 7, 64H, 0DH, 64H, 64H, 2, 3, 0, 1, 0, 0
31:                  DB      0, 0, 0AH, 36H, 0
32: ; -----HW_PC-----
33:                  DW      0B800H, 3D4H, 3DFH, 384, 4000H, 0E001H
34:                  DB      24, 16, 100, 4
35:                  DB      38H, 28H, 2DH, 10, 7FH, 6, 66H, 70H, 3, 2, 6, 7, 0, 0
36:                  DB      0, 0, 1EH, 3FH, 2
37: ; -----M24-----
38:                  DW      0B800H, 3D4H, 3DEH, 384, 2000H, 2000H
39:                  DB      24, 16, 100, 4
40:                  DB      38H, 28H, 2DH, 10, 7FH, 6, 66H, 70H, 2, 1, 6, 7, 0, 0
41:                  DB      0, 0, 1EH, 3FH, 41H
42: PREPARE           DB      35 * 10    DUP(0)
43: BEGIN:            PUSH    AX
44:                   PUSH    BX
45:                   MOV     AL, TERM_MODE
46:                   MOV     BL, VAR_NUM
47:                   MUL     BL
48:                   MOV     BX, AX
49:                   ADD     BX, OFFSET CRT_DATA
50: •                 CALL     SET_CRT_PARA
51:                   CALL     SET_LINE_ADDR
52:                   CALL     CRT_R1_16
53:                   POP     BX
54:                   POP     AX
55:                   RET
56: CRT_INIT           ENDP
57: SET_CRT_PARA       PROC    NEAR
58:                   ASSUME  CS, CODE, DS, CODE
59:                   PUSH    AX
60:                   PUSH    CX
61:                   PUSH    SI
62:                   XOR     SI, SI

```



```

63:      MOV      CL, VAR_NUM
64:      XOR      CH, CH
65:      SUB      CX, +19
66: S_C_P1,  MOV      AL, [BX]
67:      MOV      [OFFSET ES_SEG+SI], AL
68:      INC      BX
69:      INC      SI
70:      LOOP     S_C_P1
71:      POP      SI
72:      POP      CX
73:      POP      AX
74:      RET
75: SET_CRT_PARA  ENDP
76: ; -----
77: CRT_R1_16     PROC      NEAR
78:      ASSUME    CS, CODE, DS, CODE
79:      PUSH     [AX]
80:      PUSH     [CX]
81:      PUSH     [DX]
82:      PUSH     [BP]
83:      XOR      BP, BP
84:      MOV      DX, DS: [BP+OFFSET CRT_R_ADDR]
85:      XOR      AH, AH
86:      MOV      CX, +16
87: C_R1_16_3,    MOV      AL, AH
88:      OUT      DX, AL
89:      INC      [DX]
90:      INC      AH
91:      MOV      AL, [BX]
92:      OUT      DX, AL
93:      INC      BX
94:      DEC      DX
95:      LOOP     C_R1_16_3
96:      ADD      DX, +4
97:      MOV      AL, [BX]
98:      OUT      DX, AL
99:      INC      DX
100:      INC      BX
101:      MOV      AL, [BX]
102:      OUT      DX, AL
103:      INC      BX
104:      MOV      DX, DS: [BP+OFFSET CRT_R_ADDR1]
105:      MOV      AL, [BX]
106:      OUT      DX, AL
107:      POP      BP
108:      POP      DX
109:      POP      CX

```

```

110:          POP          AX
111:          RET
112: CRT_R1_16      ENDP
113: SET_LINE_ADDR  PROC          NEAR
114:          ASSUME        CS: CODE, DS: CODE
115:          PUSH          BX
116:          PUSH          CX
117:          PUSH          DI
118:          XOR            DI, DI
119:          XOR            BX, BX
120:          MOV            INTERCHANGE, 0
121:          MOV            CL, GROUP
122:          XOR            CH, CH
123: S_L_A1:         PUSH          CX
124:          PUSH          DI
125:          MOV            CL, SUM_LINE
126: S_L_A3         MOV            [OFFSET LINE_ADDR+BX], DI
127:          ADD            BX, +2
128:          CALL           ADD_DI
129:          LOOP           S_L_A3
130:          POP            DI
131:          ADD            DI, +80
132:          MOV            INTERCHANGE, 0
133:          POP            CX
134:          LOOP           S_L_A1
135:          POP            DI
136:          POP            CX
137:          POP            BX
138:          RET
139: SET_LINE_ADDR  ENDP
140: ADD_DI         PROC          NEAR
141:          ASSUME        CS: CODE, DS: CODE
142:          XOR            INTERCHANGE, 1
143:          JE             ADD_DI1
144:          ADD            DI, ADD_LINE1
145:          RET
146: ADD_DI1:       ADD            DI, ADD_LINE2
147:          RET
148: •ADD_DI        ENDP

```

# 新颖统计图形的设计 及实现方法

四川省南充棉纺织厂

袁 泉

四川省南充地区计划委员会

朱光俊

四川省南充市新泉计算机研究所

罗 炼

**摘要** 本文介绍一套统计图形软件包的设计及实现方法。此套统计图形软件包包括：直方图，叠加直方图、折线图、园图、甘特图、排列图、散布图、直方图（质量管理使用）、控制图（质量管理使用）等九种图形软件。在分辨率为 $640 \times 200$ 图形方式下作图，显示信息量较多，具有字体放大和四种显示方式供选择。屏幕显示采用活动窗口，可自动或手动控制屏幕移动显示方式。采用动画技术，使屏幕移动速度很快。屏幕拷贝打印可按要求放大打印当前屏幕的内容。为汉字dBASE III提供最方便的数据传递方式。

## 一、前言

随着计算机应用的发展，特别是经济管理中的应用的不断深入，对统计图形应用的要求越来越高。目前，企（事）业计算机管理工作中多采用汉字dBASE III关系数据库，也正因为其具有极强的数据管理和处理能力，适用性强等特点，故而广泛被采用。但是，汉字dBASE III中图形处理能力较差，通常采用的方法如下：

a. 利用其它语言的图形处理能力。

如采用BASIC语言或PASICAL语言的图形语句完成作图工作。其缺点：占用内存

较大，不宜在汉字dBASE III环境中直接运行，当在较小的内存中运行时矛盾更加突出。数据的传递比较麻烦；若采用汉字LOTUS1—2—3等集成软件时，不仅存在上述两种问题，并且汉字LOTUS1—2—3是在分辨率为 $320 \times 200$ 图形方式下完成作图，屏幕显示信息量有限。

b. 利用汉字dBASE III的语句（如poke语句）实现作图时，显示速度太慢，并且显示RAM地址难于计算。

c. 目前所使用的屏幕拷贝打印，均采用DOS所提供的屏幕拷贝打印方式，此方式不能按要求实现纵向和横向的任意放大打印。

## 七、小 结

本文所介绍的方法是在认真分析了GW 0520C-H、IBM-PC/XT、HW-PC、M24等个人计算机上使用的四种不同显示卡后提出的，并在“通用的汉字、图形仿真终端”中得到实际应用。这种方法也完全适用于显示卡性能与以上四种完全不同的SUN-286、MDT-286系统。从目前情况看，这种方法对IBM-PC兼容机具有普遍的适用性。大大节省了汉字系统的重复开发时间，方便了汉字系统的维护。

## 参考文献

- 1 许文信等，UNIX环境汉字信息系统，计算机研究与发展，1987年4期。
- 2 戴发金等，单板多机系统的显示及控制软件的设计，计算机科学技术与应用，1989.3。
- 3 周明德主编，IBM—PC（0520）系统原理及应用，清华大学出版社，1985.9。
- 4 张福炎等编，微型计算机IBM—PC的原理与应用，南京大学出版社，1985.7。

## 二、图形的设计

针对上述问题,笔者设计出一套统计图形软件包。此套软件基于大多数微机上所配备的CGA图形显示卡(即分辨率为 $640 \times 200$ ),同时也为方便汉字dBASE III环境下的使用而设计。在 $640 \times 200$ 的方式下完成作图工作,因此屏幕显示的信息量较多。显示信息以填写数据库内容的方式由用户自己决定。可在屏幕中任意位置按实际要求放大显示,并且有四种字形显示方式可选择。在汉字dBASE III环境中,建立相应的数据文件(\*.DBF)作为数据传递之用,便于数据的传递。数据文件的区段由用户自己定义。此套统计图形采用移动窗口屏幕显示方式,当图形涉及的内容较多时,可根据需要选择自动或手动移动窗口屏幕。屏幕图形采用动画技术等使作图速度变得很快。目前所使用DOS提供的屏幕打印,不能实现纵向、横向放大打印屏幕内容。因此,笔者设计屏幕拷贝打印程序时,考虑到实际应用的需要可设置纵向、横向任意比例放大倍数打印当前屏幕内容。当纵向、横向放大倍数均置为1时[即不放大打印],屏幕拷贝打印的图形比使用DOS提供的屏幕拷贝打印的图形小1/3。此套统计图形软件包可在操作系统下、dBASE III环境下或其它语言中直接调用。

此套统计图形软件包还包括两个可独立使用的程序:字型显示处理程序;图形拷贝打印程序。其中,字型显示处理程序,文件名:ZXXS.EXE。使用时,采用数据库文件作为数据传递方式,数据库的建立与以下“显示信息库”的建立方式一样。独立使用时,操作系统下使用格式为:ZXXS+空格+显示信息库文件名[.dbf],dBASE III环境下使用的格式为:run+空格+ZXXS+空格+显示信息库文件名[.dbf];图形拷贝打印程序的文件名为PRT2463.EXE。操作

系统下使用的格式为:PRT2463+空格+行放大倍数+空格+列放大倍数,dBASE III环境下使用的格式为:RUN+空格+PRT2463+空格+行放大倍数+空格+列放大倍数。

## 三、图形的实现方法

以直方图为例详细介绍图形的实现方法:

### 1.数据库的建立及约定

为了方便数据传递,将数据和显示的信息分为两个数据库文件:数据信息库和显示信息库。数据库的记录由用户按实际需要任意确定。

#### 数据信息库

此数据库中的数据为作图时所需的数据。数据库的名称由用户自定义,扩展名可定义为.dbf,也可由用户自定义。

#### 区段的设定:

第一区段 区段名自定义,区段长度自定义,区段类型为字符型,区段记录内容为图例。

第二区段 区段名为X轴第一标号,区段长度自定义,区段类型为数据类型,区段记录内容为实际数据。

第三区段 区段名为X轴第二标号,区段长度自定义,区段类型为数据类型,区段记录内容为实际数据。

第四区段 区段名为X轴第三标号,区段长度自定义,区段类型为数据类型,区段记录内容为实际数据。

....

....

第N区段 区段名为X轴第N—1标号,区段长度自定义,区段类型

为数据型, 区段记录内容为实际数据。

由于dBASE III中规定设定区段数最大值为255, 故N的最大值为254。用户可按实际需要确定N的大小。

实例:

区段名称	类型	长度	小数位
(1) NAME	Char/text	4	
(2) 一月	Numeric	12	2
(3) 二月	Numeric	12	2
(4) 三月	Numeric	12	2
(5) 四月	Numeric	12	2
(6) 五月	Numeric	12	2
(7) 六月	Numeric	12	2
(8) 七月	Numeric	12	2
(9) 八月	Numeric	12	2
(10) 九月	Numeric	12	2
(11) 十月	Numeric	12	2
(12) 十一月	Numeric	12	2
(13) 十二月	Numeric	12	2

显示信息库

此数据库包括在屏幕显示的信息, 显示的行数、列数, 行放大倍数, 列放大倍数, 以及显示方式的选择。数据库名称由用户自定义, 扩展名可定义为.dbf, 也可用户自定义。

区段的设定:

第一区段 区段名自定义, 区段长度自定义, 区段类型为字符型, 区段记录内容为屏幕显示的信息。

第二区段 区段名自定义, 区段长度自定义, 区段类型为数据型, 区段记录内容为屏幕显示信息的行数。

第三区段 区段名自定义, 区段长度自定义, 区段类型为数据型, 区段记录内容为屏幕显示信息的列数。

第四区段 区段名自定义, 区段长度自定义, 区段类型为数据型, 区段记录内容为屏幕显示信息的行放大倍数。此区段如缺省时, 行放大倍数默认为1。

第五区段 区段名自定义, 区段长度自定义, 区段类型为数据型, 区段记录内容为屏幕显示信息的列放大倍数。此区段如缺省时, 列放大倍数默认为1。

第六区段 区段名自定义, 区段长度自定义, 区段类型为数据型, 区段记录内容为屏幕显示信息的字符显示方式。此区段如缺省时, 字符显示方式默认为0。

(注: 第四区段、第五区段、第六区段可确省。)

实例:

区段名称	类型	长度	小数位
(1) C	Char/text	10	
(2) H	Numeric	2	
(3) L	Numeric	2	
(4) HF	Numeric	2	
(5) LF	Numeric	2	
(6) ZX	Numeric	2	

## 2. 字形显示方式的选择

程序中提供四种字形显示方式

- (1) 正常字形显示方式: 0
- (2) 逆向旋转90°字形显示方式: 1
- (3) 逆向旋转180°字形显示方式: 2
- (4) 逆向旋转270°字形显示方式: 3

## 3. 数据库的使用

数据库的区段按以上约定建立后, 在汉字dBASE III中直接填写实际数据, 也可在程序中按需要填写数据, 因此, 数据传递非常方便。

## 4. 功能键的使用

图形程序中设计八个功能键供选择。

(1) Ctrl\_T: 显示/关闭图例说明。[此功能键是一次性开关]

(2) Ctrl\_N: 显示/关闭当前光标所在位指示Y轴上对应的实际数据。[此功能键是一次性开关]

(3) Ctrl\_Y: 自动向右方向移动光标位置, 如超出屏幕显示时, 屏幕窗口自动连续左移; 当按任意键时, 光标停止右移; 当光



## 采用编码技术压缩数据库文件存储结构

# 新疆独山子乙烯工程指挥部

刘长江

**摘要** 新疆独山子炼油厂计算机信息系统中, 广泛采用编码技术进行各子系统的数据库设计, 该系统荣获石油天然气总公司一九八八年度计算机应用一等奖, 其数据库的设计原则、指导思想和程序设计方式, 受到专家的一致好评。

## 一、问题的提出

数据库管理系统的汉化,对于我国计算机应用起到了极大的促进作用,但由于中文处理的特殊性,又给数据库的设计和应用带来一系列的问题,数据库存贮容量问题越来越引起人们的重视,对于一些数据量很大的数据库如:人事、供应等,采用常规的数据库设计方法,有可能引起磁盘容量不足的问题。以人事数据库为例,如果该库含有工号、姓名、年令、文化程度、爱人情况、工资情况等六十个字段,其中四十个字段含有中文,

平均每个中文字段五个汉字，其余字段为四字节字符或整数则每个记录要占用480个字节，对于我厂这样一个拥有八千多职工的中型企业来说，一个人事库则要占用硬盘空间四十多兆字节，对于一般的微机，这个开销实在太大了，即便对于高档的超级微机，这也是一个不小的数目，何况系统软件，应用软件还要占用一部分存贮空间，如果是一套多用户系统，其它用户在这台机器上的可用空间则所剩无几了，诚然，我们可以采用按小单位划分建库来分散数据，这样能在局部查询中优化查询，但其总存贮空间并没有因

[illegible]

标移动到X轴最后一个直方图时，光标将停止右移，并以蜂鸣声提示。

(4) Ctrl\_Z: 自动向左方向移动光标位置,如超出屏幕显示时,屏幕窗口自动连续右移;当按任意键时,光标停止左移;当光标移动到X轴第一个直方图时,光标将停止左移,并以蜂鸣声提示。

(5)→: 手动向右方向移动光标位置, 如超出屏幕显示时, 屏幕窗口左移; 当光标移动到 X 轴最后一个直方图时, 光标将停止右移, 以蜂鸣声提示。

(6)←：手动向左方向移动光标位置，如超出屏幕显示时，屏幕窗口右移；当光标移动到X轴的第一个直方图时，光标将停止右移，以蜂鸣声提示。

(7) Ctrl\_P: 打印当前屏幕显示内容。打印时,按用户设定的纵、横放大倍数要求进行屏幕拷贝打印。

(8)ESC: 退出。

## 5. 屏幕拷贝打印

当用户设置屏幕拷贝打印功能键(Ctrl\_P)时,将要求设置打印纵向放大倍数、横向放大倍数。纵向放大倍数的最大值为999;横向放大倍数的最大值为999。纵向放大倍数的缺省值为1;横向放大倍数的缺省值为1,均认为不放大。此时,屏幕拷贝打印的图形比使用DOS提供的屏幕拷贝打印的图形小1/3(附图略)。

此套统计图形软件包采用C语言编写。适用于IBM PC—XT、AT 及兼容机，在CCDOS2.1及以上版本的DOS 操作系统中运行。此套软件包系南充棉纺织厂与南充市新泉计算机研究所、南充地区计划委员会信息中心共同研制开发，并在南充棉纺织厂投入运行一年多，反映良好。

数据的分散而减少,反而加大了综合查询和统计分析的系统开销。

在独山子炼油厂计算机信息管理系统中广泛采用了数字编码技术和主语言方式、有效地解决了上述问题,并开发出一个个高效实用的应用子系统。

## 二、采用编码技术压缩数据存贮量

前面提到,中文数据库大大方便了用户,但随之而来的是使数据库的容量更加庞大,甚至到了使人无法忍受的程度,我们在数据库设计中采用编码技术,一般有两种方法,一是数据库全部采用代码方式,另行建立一个数据词典文件,还有一种方法是不用数据文件方式,而采用另行建立一个对照数据库的方式。不论采用那种方法,其实质都是用指定的代码代替汉字,这样相同代码的汉字在数据词典中只出现一次,而代码的长度要远远少于汉字的长度,从而减少了数据库占用的存贮空间。仍以上述人事库为例,60个字段,全部采用编码存贮方式后,数据库中只存序号,则两字节整数可表示32767个不同项目,实际上一般三位数就足够了,这样每个记录占用字节数为 $40 \times 2 + 20 \times 4 = 160$ 个字节,数据量只有原来的30%左右,如果能对不同的字段根据其不同内容个数,分别给以不同的长度,则可以进一步节省空间,如性别一栏,只有两个值,即男和女,只要3位数就可以表示出来(其中高两位表示字段号)。当然这样一来,程序处理上增加了一点难度,这里不再详加叙述。

数据词典结构如下:

1 2 3 4 5(位)

字段号 序号,中文信息

例如有以下三个字段,政治面貌、族别、性别,其数据词典格式如下:

1001,党员

1002,团员

1003,群众

2001,汉族

2002,蒙古族

2003,...

2004,...

3001,男

3002,女

显然,采用编码技术后,可大大节省存贮空间,并且只要系统是汉化操作系统,主语言(又叫嵌套式语言)支持中文,数据库允许使用英文数据库,这是我们开始自己也曾未曾预料到的,这一点对于微机也许意义不是很大,但对于有些没有配备中文数据库的中小型机来说,至少提供了一种有效使用数据库的手段。编码技术的采用有许多优点,但随之而来的是给数据库管理和程序设计带来一系列的新问题,如:数据库不再象全中文时那样直观,某一字段代码超过三位数之后怎么办?如何将代码和中文数据进行转换?操作是否方便?对运行速度是否有影响等等,这些问题不解决,节省再多的存贮空间也是没有意义的,这些问题将在下面进行讨论。

## 三、有关问题的讨论

编码技术的优点是显而易见的,对上面提到的问题我们一一进行讨论。

首先看编码位数,如果是五位码,则每个字段的项目不能超过1000个,当然,这种情况是很少见的,一旦出现这种情况,一种方法是扩充编码位数,很显然为一个或个别几个字段增加一位编码是不值得的,这样会使操作增加许多困难。一般常采用另一种方式,即单独为该字段建立一个数据词典,在程序中加以处理,用户操作起来不会有什么不适应的感觉。

采用编码技术后,一般程序设计采用主语言方式,首先程序在数据库中按指定的条件检索数据,然后根据数据词典将代码转换成中文,对于用户来说,体会不到查出的是

代码还是中文数据,开始我们担心这个转换过程会使用户等待时间加长,但采用了平衡二叉树查询技术之后,对程序运行几乎没有什么影响,从库里检索出数据到转换显示出来几乎没有什么停顿,下面给出二叉树查询的子函数:

```
retrieve(p, d)
/*在以p为根的二叉树中查d串*/
TREE p;
char d[];
{
    if(p != NULL) {
        if(strcmp(d, p->bh) == 0) {
            printf("%S", p->name);
            return(0);
        }
        if(strcmp(d, p->bh) > 0)
            retrieve(p->right, d);
        else retrieve(p->left, d);
    }
    else return(-1);
}
```

限于篇幅,建立平衡二叉树的算法不再给出,在数据结构的有关资料上均可查到,我们这里无需证明二叉排序树的查询效率,对于只有8M主频的 BCM—68000 微机来说,在六千多个记录中查询,回车之后汉字即可显示在屏幕上,速度上是不需担心的。

采用代码技术后,用户界面更加友好了,用户适应后使用十分方便,首先中文输入量大大减少了,原先每个字段的汉字均需输入,现在只需输入一次,以后就可以使用该代码来输入该项目了,如光标停在政治面貌一栏时,用户只要打入1后回车,程序自动将栏号乘以1000后加上输入值1得到1001,然后调用 retrieve函数去进行查询,得到党员的显示,其中建立二叉排序树、查询等子程序均已结构化形成标准函数,在任何程序中都可以任意调用。如果用户忘记了代码,可以用帮助键进行查询相应代码,这就大大方便了用户,对于一个新代码,则要求输入对应的中文信

息,由程序将其插入到排序二叉树上,然后写入数据词典,既保证了本次程序运行中该代码及数据已存在,又保证了该项目已被加入到数据词典中,从系统开销上看,一个六千个记录的数据词典也不过占用几十K的空间,对一般的系统来说这种开销都是可以接受的。

通过上面的介绍我们可以看出,编码技术是一种节省存储空间,改善用户界面有效使用数据库的有效手段,我们采用这种技术在容量和速度都很有限制的BCM—68000微机上开发出了人事、销售、计划、调度等多个应用子系统,在鉴定会上受到与会专家的一致好评。由于该项技术是以主语言为核心,既保持了数据库安全可靠的优点,充分发挥了它强大的检索功能,又有利于提高数据库软件的效率、可移植性和分布处理能力。对于网络系统,采用编码技术,可以大大减少网上的数据传输量,有助于提高网络传输速率和网络的响应时间,另外还可以锻炼软件人员的程序设计能力。

编码技术有很多优点,但远非尽善尽美,例如:数据库中强有力的检索手段—模式识别和匹配,在全是代码的数据库中就无法发挥其作用,我们只能通过其它手段来进行处理,比如将经常要用到模式匹配的字段,我们可以将中文信息直接放入数据库中,将编码技术和常规使用方法结合起来以求得到最佳的使用效果。

我们的实践证明,编码技术是有实用价值的,希望广大计算机工作者都来进行尝试,使这一技术得到更广泛的应用。



# AOS/VS 系统记录的使用

天津市电子计算机研究所 杨建农

**摘要** 本文介绍了MV系列小型计算机操作系统AOS/VS系统记录功能的使用。并通过在“中国电子情报检索系统”中对这一功能的使用，说明了如何使用系统记录对多用户系统进行计时收费管理。如何根据系统记录的数据对用户进行有效的用机指导。

## 一、引言

MV系列小型计算机在我国已有了广泛的使用。其操作系统AOS/VS的系统记录功能记录了所有联机用户的使用信息。使用好这一功能对系统管理员来说是十分重要的，可通过它对联机用户进行管理，也可根据它对联机用户进行计时收费，从而提高系统的使用效率。

## 二、AOS/VS系统记录的功能

AOS/VS的系统记录文件是根目录中的名为SYSLOG 的文件。系统管理员可通过

主进程用CLI的SYSLOG命令启动、停止它的工作。系统在SYSLOG 文件中记录了。

- 与用户用机记帐有关的信息
- XODIAC网络信息
- 软的和硬的错误记录
- 在系统中运行的任一超级用户用LOG-GEVENT命令记入SYSLOG文件中的临时信息等等。

启动系统记录功能的最常用命令是：

SYSLOG/START ddmmyy

此处ddmmyy是上一次启动记录文件的日期如：18DEC89。这样就清楚且唯一地标识出了各记录文件。为了使文件的意义更加清楚，可加上文件名后缀。LOG，如：18DEC89.LOG。这样可以在每天启动系统

一般的三栏式帐页，只需用窄行打印纸，比起宽行打印纸来可节省三分之一的纸张经

费。因此，这种打印格式颇受用户欢迎。

附表1 用一般打印方式打印输出的帐页式样

1990年		摘要	收 方										付 方										收或付	余 额													
月	日		亿	千	百	十	万	千	百	十	元	角	分	亿	千	百	十	万	千	百	十	元		角	分	十	亿	千	百	十	万	千	百	十	元	角	分
		上年结转					4	3	8	0	0	5	1	9											收					4	3	8	0	0	5	1	9
1	23	上级拨款			9	8	7	6	5	4	3	2	1												收		1	0	3	1	4	5	4	8	4	0	
1	30	发元月份工资																						收			1	0	2	6	7	8	4	6	0	0	
		本月合计	1	0	3	1	4	5	4	8	4	0												收		1	0	2	6	7	8	4	6	0	0		
		累计	1	0	3	1	4	5	4	8	4	0												收		1	0	2	6	7	8	4	6	0	0		

附表2 用本打印方式打印输出的帐页式样

1990年		摘要	收 方					付 方					收 或 付	余 额																						
月	日		亿	千	百	十	元	角	分	亿	千	百		十	元	角	分	十	亿	千	百	十	元	角	分											
		上年结转			4	3	8	0	0	5	1	9											收					4	3	8	0	0	5	1	9	
1	23	上级拨款			9	8	7	6	5	4	3	2	1											收		1	0	3	1	4	5	4	8	4	0	
1	30	发元月份工资																						收			1	0	2	6	7	8	4	6	0	0
		本月合计	1	0	3	1	4	5	4	8	4	0												收		1	0	2	6	7	8	4	6	0	0	
		累计	1	0	3	1	4	5	4	8	4	0												收		1	0	2	6	7	8	4	6	0	0	

附表3 新增加的打印图型符

第1组	亿	万	千	百	十	元	角	分		
第2组	0	1	2	3	4	5	6	7	8	9
第3组										



运行之前启动当天的系统记录文件工作，并将前一天的系统记录文件以日期加以区别，并保留起来以备使用。

SYSLOG文件记录的这些信息，并非ASCII码文件，不能直接读取它们。不过用户在他们的程序中使用适当的代码格式，也可以直接读取系统记录的信息。系统提供了一个REPORT实用程序来取得记录文件报告。REPORT程序对系统记录文件做出解释、归类、显示等处理，并可将系统记录的信息写到用户指定的任一设备式文件上。

标准报告由两部分组成，第一页报告用户信息，第二页报告设备错误。它所得到的信息如下：

- 用户名（\*表示超级用户）
- 此用户的联机时间
- 此用户使用磁带机的时间
- 此用户打印输出的页数
- 此用户占用CPU时间（以时：分：秒毫秒表示）
- 此用户读/写的盘区数
- 页-秒数（此数为使用的物理页数与每秒所占CPU秒数的乘积，由此可考查此用户的用机效率）
- 此用户在联机期间建立的进程数
- 错误信息

报告中第一页的基本格式如下表所示：

### 三、实例介绍

表

VSERNAME	CONNECT CONSOLE	TIME UNIT	PAGES PRINTED	CPU TIME	I/O BLOCKS	PAGE SECS	NUMBR PROCS
• OP	2:21	0:00	3	0:01:41.187	4834	10523	25
YANG	0:09	0:00	0	0:00:01.663	176	143	3
WANG	2:29	0:00	15	0:00:32.306	3535	3696	40
ZHANG	1:11	0:00	2	0:08:05.565	7480	61608	28
:	:	:	:	:	:	:	:

笔者在“中国电子情报检索系统”中利用系统记录功能对联机检索用户进行了计时收费管理，在此做一简单介绍。

为了给用户记录下来用机情况，必须在每天开机后启动多用户工作之前启动SYSLOG文件工作。同时将前一天的系统记录文件传送到一个事先建立好的：LOGS目录中，并使用REPORT程序将其报告写到一个临时文件中，然后读取该文件的有关用户用机记帐信息，并将这些数据逐一累加到管理员自己建立的用户记录文件中，待以后用户结帐时查阅。为此可做一宏程序，每天开机后启动此宏程序，宏程序中！NEQUAL为比较用逗号隔开的两个字符串，如果不相同执行下面的命令，否则执行[！ELSE]后面的命令。由于宏程序中使用了PUSH命令使得当前系统环境上推一层，在PUSH命令之前赋给STRING的内容与其之后赋给STRING的内容不相同，所以程序中[！STRING/p]为当天的日期文件名，而[！STRING]为前一天的日期文件名。

宏程序中的PRO\_NAME程序可用任一种高级语言编写。（笔者使用的是C语言）它把由REPORT程序产生的系统记录报告文件USERLOG.LOG进行处理。读出每个用户的联机时间、使用CPU时间，打印页数I/O块数等数据，并找到用户记录文件中各用户的记录，将其中的数据累加到相应的用户记录中去。

宏程序如下页所示：

```

PUSH
STRING [MAKE_DATE[ 'EXPLODE[ 'DATE]]], LOG (将当天日期文件名ddmmmyy.log
[ 'INEQUAL, ([ 'FILENAMES: LAST_LOG_DATE]), ( )] 放入STRING1中)
PUSH
STRING [: LAST_LOG_DATE] (将前一天日期文件名ddmmmyy.log放入STRING2中)
[ 'INEQUAL, [ 'STRING/P], [ 'STRING]] (比较STRING1和STRING2是否相同)
SYSLOG/START[ 'STRING] (启动系统记录文件, 并将前一天的记录文件放入STRIN
PUSH G2所指定的文件中)
SUPERUSER ON
DIR,
ACL [ 'STRING] OP, OWARE+, RE
WRITE Moving logfile [ 'STRING] to directory : LOGS
MOVE : LOGS : [ 'STRING] (将前一天的记录文件传送到, LOGS目录中)
X REPORT/L=USERLOG.LOG[ 'STRING] (将前一天的记录报告放入文件USERLOG.
LOG中)
X PRO_NAME (运行记帐程序)
DELETE USERLOG.LOG
DELETE [ 'STRING]
POP
[ 'ELSE]
SYSLOG/START (启动系统记录文件)
[ 'END]
POP
[ 'ELSE]
SYSLOG/START (启动系统记录文件)
[ 'END]
SUPERUSER ON
DELETE/2=IGNORE : LAST_LOG_DATE
STRING/L= : LAST_LOG_DATE (更新日期文件: LAST_LOG_DATE的日期)
ACL : LAST_LOG_DATE+, R
POP

```

这个程序的简单处理框图如图1:

首先该程序必须根据REPORT 报告文件的格式将文件指针定位在第一个用户名的位置。根据报告文件中不管前面的格式千变万化在第一个用户名之前总是“SECS”和“PROCS”两个字符，由此便可指定文件指针到这两个字符之后，然后读出第一个用户名和该用户的各项数据。并使用这里的用户名找到用户记录文件的该用户记录，将联机时间和CPU时间变换成与用户记录文件单位相同的数据格式，并连同其它有关数据

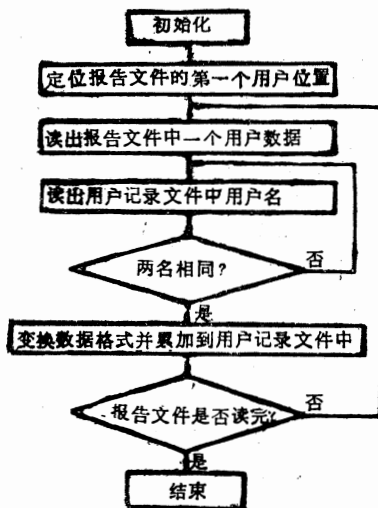


图1

一起累加到该用户记录的各项数据上。这个程序的清单如下：

```
#include<stdio.h>
FILE *fopen( ), *fd, *fuser;
char string[80], syusername[17],
        ccntry[255];
char user[40], con[40], time[40], cpu
        [40], filestr[180];
char syusername[17], cc[4], bj[4];
int pages, iao, pag, num, cpum, conn,
        cpun;
int err, dd, ddl, connl opunl, 1001;
int l;
main( )
{
    fdf=open( ":userlog.log", "rt" );
    fuser=fopen( ":util:cedors:user_reg:
ster.log", "rt" );
    fseek( fuser, 0, 2 );
    dd=ftell( fuser );
    fseek( fd, 0, 2 ); dd1=ftell( fd );
    fseek( fd, 0, 0 );
    do
    {
        fscanf( fd, "%s", time );
        if ( strcmp( time, "SEOS", 4 ) == 0 )
        {
            zero( time, strlen( time ) );
            fscanf( fd, "%s", time );
            if ( strcmp( time, "PROCS", 5 ) =
== 0 ) break;
        }
        zero( time, strlen( time ) );
    }
    while ( ftell( fd ) ( dd1 );
    err=fscanf( fd, "%s%s%s%d%s%d%d
%d", user, con, time, &pages, cpu,
&iao, &pag, &num );
    while ( arrl=-1
    {
        fseek( fuser, 0, 0 );
        do
        {
```

```
if ( ftell( fuser ) == dd ) break;
fgets( bj, 2, fuser );
fseek( fuser, 144, 1 );
fgets( syusername, 17, fuser );
fseek( fuser, 40, 1 );
if ( strcmp( syuse rname1, syuser-
name, 16 ) == 0 ) break;
    }
    while ( ftell( fuser ) < dd );
    fseek( fuser, -20, 1 );
    if ( strcmp( syusername1, syusername1,
16 ) == 0 )
    {
        zero( time, strlen( time ) );
        conn=atoi( con );
        itoa( conn, time );
        conn=conn*60;
        conn=conn+atoi( &oon[strlen( ti-
me )+1 ] );
        zero( time, strlen( time ) );
        cpum=atoi( cpu )
        itoa( cpum, time );
        cpum=cpum*60000;
        cpum=cpum+1000*( atoi( &cpu
[strlen( time )+1 ] ) );
        cpun=cpum+atoi( &cpu[strlen(
time )+4 ] );
        zero( time, strlen( time ) );
        connl=getw( fuser );
        cpunl=getw( fuser );
        iao1=getw( fuser );
        fseek( fuser, -12, 1 );
        conn=conn+connl;
        putw( conn, fuser );
        cpun=cpun+cpunl;
        putw( cpun, fuser );
        iao=iao+iao1;
        putw( iao, fuser );
    }
    err=fscanf( fd, "%s%s%s%d%s%d
%d%d" user, coo, time, &pages, cpu,
&iao, &pag, &num );
    }
}
```

(下转第17页)

# 用24针式打印机打印屏幕图形

机械委北京自动化所 刘迅

在实际工作中,有的用户为了节省资金,充分利用现有设备,希望直接使用针式打印机打印屏幕图形。目前在这方面受到了很多限制。有的只能打印某一种显示模式下的屏幕图形;有的虽能进行屏幕图形硬拷贝,但打印出的图形与屏幕图形大小不成比例(往往比屏幕图形小的多),视觉效果不好;并且当屏幕上出现多种颜色的彩色图形而只希望拷贝出某几种颜色的图形时,就无能为力了。特别是在实际应用中,屏幕上作出的图形周围会有很多提示、说明等并不希望拷贝出的内容,而现有的打印机驱动程序是不能满足这些要求的。针对这一情况,作者对使用针式打印机打印屏幕图形进行了尝试,解决了上述问题,并取得了较好的使用效果。

我们的开发工作是在以PC机(XT或AT)、M1724打印机及COLOR-400彩卡配普林斯顿12"彩显的系统上进行的。当然,这一工作原理对其它配置也是适用的。

使用针式打印机拷贝屏幕图形,就是要编制出一个打印机打印图形的驱动程序,这主要有两大方面的问题需要解决:一是打印机的控制,这包括打印机走纸控制和打印机打印点阵的输出控制;二是屏幕图形数据的采集与整理。下面就分别加以说明。

对于24针式打印机来说,打印图形是利用其打印点阵的功能实现的,即使用ESC操作码序列,这在打印机使用说明中均可查到。着重解决的问题是:要保证打印机打印完一行点阵后走纸到下一行的距离要处理好,以使图形连接正确,否则将出现图形误差,这同样可以靠ESC操作码序列设置打印走纸间隔来实现。对M1724型打印机,这一命令序列为:ESC.J.n1.另外,因打印机针距很小,这样

若将屏幕点阵数据直接打印,就会造成图形太小,视觉效果很差。如M1724型打印机,打印640×400的彩色图形时,拷贝下来的图形只有屏幕图形实际大小的四分之一左右,这是我们所不希望的。这样我们就必须将图形放大,使其接近于屏幕图形的实际大小或比它还大些,以取得满意的视觉效果。而解决这一点就要对影像进行横、纵两个方向上的放大处理,如对于M1724型打印机,要使打印出的图形与屏幕图形大小相近,就要打印使图形影像横纵都扩大一倍,对于横向,我们可以直接使用ESC序列码,即:ESC.=.n1.n2.其中:n1=2,n2=128,即屏幕列数为:n1×256+n2=2×256+128=640列,打印列数为640×2=1280列,而纵向放大影像的实现就要靠程序来完成了。当然,若横向还希望放大的话,也同样要靠程序来实现。再有,当屏幕上有多种颜色的图形时,有一个各彩色页面图形打印的问题,因M1724是黑白打印机,只能打印一种颜色图形,所以,一种方法是将各页面的图形组成一个页面后送给打印机,使图形拷贝一次完成;另一种方法是先取一个页面图形打印,完成后使打印机退回到图形打印起点,再打印下一个页面。作者认为第一种方法较好,它可以避免因回车退纸造成的起始打印误差,并缩短打印时间。

在谈第二个问题的解决方法之前,先要搞清楚,打印机打印点阵数据与屏幕数据在屏幕缓冲区中的排列顺序是不同的。对于打印机来说,打印机的24针数据是按列进行的,即打印它1列24点值后,打印头前进一列位,准备好下一列的打印操作。而对于屏幕数据来说,它在缓冲区中是按行存贮的。所以,在向打印机送点阵数据时,先要将屏幕上以

行存贮的数据,变为适合于打印机打印输出的以列值存贮的数据形式,才能正确地打印出屏幕图形来。

第二个问题的解决,就是要编制出一个打印屏幕图形的驱动程序来。这一程序的框图如图1所示。

下面对程序框图作一些必要说明:

设置打印起始行、结束行、起始列、结束列:这是这一打印驱动程序的四个入口参数,当然还可增加选彩色页面的参数。若总是进行全屏幕拷贝,可在程序中直接置数,但在实际工作中,往往只希望将屏幕图形部分硬拷贝,而屏幕上其它如提示、说明等内容并不需要,故为了使程序适用于各种情况,设置了这四个入口参数。

行计数器大于设置总行数吗?在这一判断模块中,设置总行数是指结束行与开始行之差,并且当它不是12的整数倍时,要将其进位到12的整数倍,这是因为行计数器是按12增值的。这里要说明一下,增值为什么是12,而不是24呢?正如前面所述,这是为了使打印点阵影像在纵向扩大一倍而决定的,以使打印出的图形与屏幕图形大小相近,以便收到满意的视觉效果。

打印屏幕12行子程序调用:这一功能模块是整个驱动程序的核心,它编制的质量如何,将直接影响打印效果的好坏及打印时间的长短。这一模块的程序框图如图2所示。

现就这一模块作一些必要说明。

设置打印一行数据缓冲区:这一缓冲区的设置是放大位影像ESC操作码要求的,它要求一行的打印数据要一次向打印机送出。这一缓冲区的大小是这样确定的:ESC.(=).n1.n2.是位影像操作码,根据输入的打印起始行与结束行数据,即可确定n1、n2值。如全屏幕拷贝时,是从0列至639列,则有:  $n1 \times 256 + n2 = 2 \times 256 + 128 = 640$  列,而一行24针数据要由三个字节组成,故缓冲区大小应为:  $640 \times 3 = 1920$  字节。若不

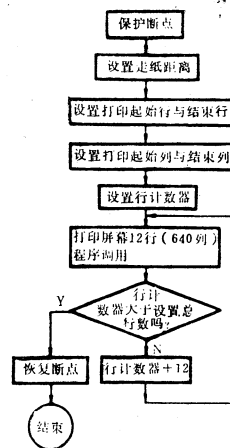


图1

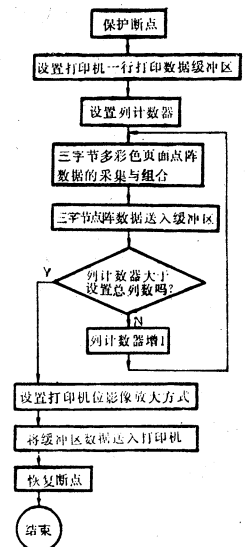


图2

能全屏幕拷贝,则按此原则,同样可求出缓冲区的尺寸。另外有一点要注意的是,这一ESC操作码要求:  $1 \leq N + (n1 \times 256 + n2) \leq 2176$ , 其中N是在这个代码前输入的数据的列数,而2176值是被打印机最大打印列数卡死的。在实际编程中,不要超过此值,否则,ESC操作码将变为无效命令!

三字节多彩色页面点阵数据的采集与组合:这一模块包括,任一页面任一列某四行数据的采集、扩大和多页面数据的组合整理。如前所述,要满足纵向影像的放大,24针数据应为12点数据扩大一倍成24点数据后再送入打印机,24点由3个字节组成,4点值扩大一倍即为一字节8位值。这样,我们就必须每采集纵向4点值就放大一次成8点值,送入缓冲区。这里不使用采集完12点后一次放大成24点值的方法,主要是因为使用字节操作比较灵活且速度较快的原故。多个页面的组合比较容易实现,只要将各页面放大后的位影像值相“或”后,再送入打印机即可实现。

上面是本人就用24针式打印机打印屏幕图形的实现之方法的体会,其中会有不到之处与有待改进之处,望同行们提出宝贵意见,使这一方法进一步完善。



# IBM PC机除法溢出中断的修改

南京工程兵工程学院 邢建春

## 一、问题的提出

IBM PC机中当除法运算的分子大于或等于分母时,产生0号中断(除法溢出中断),给出除法溢出提示。如果溢出中断服务程序能够给出产生溢出的除法指令(DIV或IDIV)的位置,即该除法指令处代码段寄存器的值CS:IP,同时给出执行DIV或IDIV除法指令时寄存器AX、BX、CX、DX、SI、DI的值,这对于确定产生除法溢出的位置,判断产生溢出的原因,进而排除除法溢出错误,加快程序调试的进度,是非常有益的。本文给出了实现这一目的的一种方案,在ST386机器上的应用是成功的。

## 二、解决问题的方法

实现以上要求的程序清单见第三部分。我们重新设计了0号中断服务程序,以代替BIOS给出的0号中断服务程序。这个新的中断程序能够给出上面提出的CS:IP及AX、BX、CX、DX、SI、DI的值。

0号中断服务程序的修改是在主程序中完成的,程序框图见图1。首先将新的中断程序divint的入口地址放入DS:DX中,然后调用DOS的int21H第25H号功能,修改0号中断向量,即将DS:DX之值填入RAM存储器低端开始的4个字节中,替代BIOS给出的0号中断入口地址。主程序随后调用DOS的int21H第31H号功能,将新的0号中断服务程序驻留于内存中,并使程序进程退回DOS。这样,就实现了对0号中断的修改。

新的0号中断服务程序框图如图2所示。首先保存中断点处的现场数据。然后从栈中

取出中断点处CS:IP的值(即DIV或IDIV指令的位置),以及寄存器AX、BX、CX、DX、SI、DI的值,调用二进制——ASCII码转换子程序,将各寄存器的值转换为ASCII码,并存到指定的内存单元中,调用DOS的int21H第9号功能,在屏幕上给出“除法溢出”提示,并将CS:IP、AX、BX、CX、DX、SI、DI的ASCII值从内存中取出显示在屏幕上。最后,调用DOS的int21H第4CH号功能将程序进程返回到DOS下。这时,我们就可以根据0号中断程序给出的中断点位置CS:IP及各寄存器的值,很快的找到产生溢出的位置及原因,并对程序作出适当的修改。

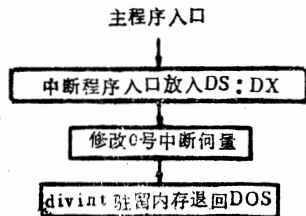


图1 主程序流程图

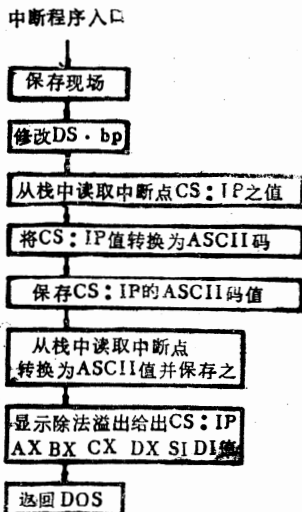


图2 中断程序流程图

### 三、程序清单

这是在ST386微机上用MS-Assembler V5.0汇编通过的程序ZERODIV.ASM的清单。

```
name zerodiv
page 55, 132
title 'ZERODIV.ASM...Divide Overflow Interrupt Handler '
ZERODIV.ASM——修改BIOS除法溢出中断的汇编语言程序
;
; 编译、连接得到可执行文件的方法:
;           C>MASM ZERODIV,
;           C>LINK ZERODIV,
; 将ZERODIV.EXE转换为ZERODIV.COM,
;           C>EXE2BIN ZERODIV ZERODIV.COM,
; 注: 必须执行文件ZERODIV.COM, 执行文件ZERODIV.EXE不能正确修改
; 中断服务程序
cr    equ    0dh                ; 常数定义

lf    equ    0ah

cseg segment
org    100h
assume cs: cseg, ds: cseg, ss: cseg, es: cseg
mainp proc
    mov     ax, cs                ; 取得中断服务程序入口段地址
    mov     ds, ax                ; 及段内偏移地址
    mov     dx, offset divint
    mov     ax, 2500h             ; 修改0号中断入口地址
    int     21h
    mov     dx, ((pgm_len+15)/16) + 10h
    mov     ax, 3100h             ; 退回DOS并将中断程序驻留内存
    int     21h
mainp endp

divint proc
    push     ax                    保存现场
    push     bx
    push     cx
    push     dx
    push     si
    push     di
    push     bp
    push     ds
```

push	es	
sti		
mov	ax, cs	
mov	ds, ax	
mov	bp, sp	
mov	si, offset cs_ip+9	; 取得CS:IP的存储区首地址
mov	ax, ss:[bp+14h]	; 从堆栈中取CS
call	bin_asc	; 将CS的值转换为ASCII码
mov	ds:[si], dx	; 存ASCII码
mov	ds:[si+2], bx	
mov	ax, ss:[bp+12h]	; 从堆栈中取IP
call	bin_asc	; 将IP的值转换为ASCII码
mov	ds:[si+5], dx	; 存ASCII码
mov	ds:[si+7], bx	
mov	di, 3	
add	bp, 10h	
mov	si, offset ax_bx	
reglp: mov	ax, ss:[bp]	; 从堆栈中得到AX, BX, CX, DX, SI, DI
call	bin_asc	; 的二进制值, 调用二进制—ASCII码
mov	ds:[si+5], dx	; 转换子程序, 得到对应的ASCII码,
mov	ds:[si+7], bx	; 并将此ASCII码存到内存指定的位置
dec	bp	; 为在屏幕上显示作好准备
dec	bp	
mov	ax, ss:[bp]	
call	bin_asc	
mov	ds:[si+14], dx	
mov	ds:[si+16], bx	
dec	bp	
dec	bp	
add	si, 18	
dec	di	
jnz	reglp	
mov	dx, offset warn	; 在屏幕上给出除法溢出提示
mov	ah, 9	; 并显示CS:IP及寄存器
int	21h	; AX, BX, CX, DX, SI, DI的值
mov	ax, 4cfffh	; 调INT 21第4C号功能返回DOS
int	21h	
divint	endp	
bin_asc	proc	; 二进制—ASCII码转换子程序
mov	dx, ax	; 入口为AX

```

        mov     al, ah                ; 出口为DX:BX
        call    toasc
        mov     al, dl
        mov     dx, bx
        call    toasc
        ret
bin_asc endp

```

```

toasc proc                ; 二进制—ASCII码转换子程序
        xor     bx, bx                ; 入口为AL
        mov     bl, al                ; 出口为BX
        mov     ch, al
        mov     cl, 4
        shl     bx, cl
        mov     bl, ch
        and     bl, 0fh
        cmp     bh, 0ah
        jnc     asc1
        or      bh, 30h
        jmp     asc2
asc1:   or      bh, 40h
        sub     bh, 09h
asc2:   cmp     bl, 0ah
        jnc     asc3
        or      bl, 30h
        jmp     ascret
asc3:   or      bl, 40h
        sub     bl, 09h
ascret: mov     cl, bl
        mov     bl, bh
        mov     bh, cl
        ret
toasc endp

```

```

warn      db      cr, 1f, 'Divide overflow detected'
cs_ip     db      'at point' , 0, 0, 0, 0, ':', 0, 0, 0, 0, cr, 1f
ax_bx     db      cr, 1f, 'AX=' , 0, 0, 0, 0, 'BX=' , 0, 0, 0, 0
cx_dx     db      'CX=' , 0, 0, 0, 0, 'DX=' , 0, 0, 0, 0
si_di     db      cr, 1f, 'SI=' , 0, 0, 0, 0, , DI=' , 0, 0, 0, 0, cr, 1f
          db      '$'
pgm_len   equ     $-mainp

cseg      ends
end       mainp

```

# VAX11—DMF32并行接口的开发应用

西安市电子科技大学计算中心

周鸿 刘志镜

**摘要** 本文介绍了通过配接专用的接口适配器,编制驱动程序,实现了对原有旧设备 BCK-I 磁带机通过DMF32控制板的并行接口与 VAX-11/780机的DMA传输,为有效地利用设备,提高系统的利用率提供了有益的尝试经验。

## 一、引言

为了对一些旧设备加以利用,通过配接专用的适配器,编制驱动程序,我们进行了一种国产磁带机BCK-I通过DMF32控制板的并行接口与VAX机进行DMA传输的一次尝试。

## 二、系统方案

DMF32是VAX系列机的一种DMA UNIBUS智能控制器,它支持下列I/O设备的一种组合:

- 八路异步线路
- 一路同步线路
- 一个DMA 行式打印机接口或一个增强的DR11-C功能的并行接口

BCK-I 磁带机是一种国产的双驱动式磁带机,具有手动操作和联机自动操作两种功能。在联机情况下,需要由主机发送相应的控制信号并等待数据输入。

将BCK-I 联接入VAX系统,为了提高磁带的I/O速度,同时又不致影响主机对系统中其它设备的处理,我们采用了DMF32的并行接口并以DMA方式进行数据传输。

系统的硬件连接如图1所示。

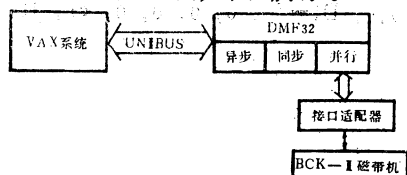


图1

## 三、接口适配器

作为主机与磁带机的接口,一方面它要能将磁带的串行数据转换成并行数据(以字节为单位),然后将其发送给主机;另一方面,它还必须具有将主机的并行数据转换为串行数据发送给磁带机的功能。因为采取DMA方式,每次数据交换为64B,在适配器中设置两个64B的数据井。不论是主机发来的数据,还是从磁带上读向主机的数据,都先存放在这两个数据井中。

DMF32 并行接口与磁带机信号约定如下:

控0	0	1	0	1
控1	0	0	1	1
工作状态	停	启	读带	写带

A请求:表示适配器RAM允许主机存取数据。

B请求:表示适配器向主机发送一次数据已结束。

适配器的结构如图2所示。

图2中RAM由两片静态读写存储器2114 A组成,它使用两段地址,每段为64B。磁带数据地址寄存器由七位组成,前六位可计数。它表示带上数据向RAM的存取地址。主机数据地址寄存器同磁带数据地址寄存器结构相同,它表示主机向RAM存取数据的地址。

适配器的工作过程简述如下:在读带

时，主机发来“启”态、“读带”态，经控制电路，驱动磁带机转动，磁带数据经移位、锁存，而进入RAM的数据段，其地址由磁带数据地址寄存器指定。当其中一段的64B空间满后，向主机请求DMA传送。同时，磁带机继续工作，其数据进入RAM的另一数据段。这样，主机便有填满一个段的时间来响应DMA请求。当下一段满时，再次向主机请求DMA传送。同时，因为上一段的数据已经取空，磁带数据地址又重新指向上一段的开始，继续读带。依次执行，进行读带的操作。

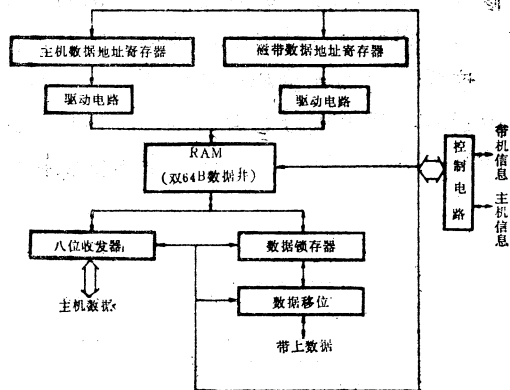


图2

在写带时，与读带相反，当RAM空时，向主机请求DMA传输，一次传输填满适配器RAM的一个段。其地址由主机数据地址寄存器指定。两个段轮换，相互配合，直至写带结束。

从以上的工作过程可以看出，这样的设计能够比较好地协调高速主机与低速外设的矛盾，而又不致影响系统效益，避免了单缓冲时一方面是低速外设，而另一方面因不能及时得到主机的响应而不得不停止工作的情形。

#### 四、DMF32并行接口

##### DMA传送驱动程序

在VAX/VMS中，并未给用户提供了DMF32并行接口的DMA驱动程序，只是有一些

类似的样板程序

可参考。

VAX/VMS

提供一个比较标准的

用与开发，一般驱

驱动程序

驱动程

功能判

FDT例

设备处理

驱动程序的组织与编写，在1的有关资料中有比较详细的描述。因为一个特殊的地方是要进行DMA DMA传输更多地依赖于硬件的支下面仅列出其设备处理例程的简化图3。

驱动程序分派表列出了该驱动程序的入口点。功能判定表列出了对该驱动程序的能代码：指出这些功能是否由该驱动程序处理，并指定子程序来预处理（对应的FDT例程入口）。

括启、停、读带、写带四种功

设备处理例程激活一个设备驱动程序，完成实际的数据交换，即进行的数据传输。

驱动程序的组织与编写，在1的有关资料中有比较详细的描述。因为一个特殊的地方是要进行DMA DMA传输更多地依赖于硬件的支下面仅列出其设备处理例程的简化图3。

如 DR11-C的驱动程，操作系统中，为了给用户提供人机介面，方便机器的使用，驱动程序的组织规定如下：

!序表 (DPT)  
序分派表 (DDT)  
要表 (FDT)



# IBM-PC/XT与FWX-4675绘图仪的联接

中国科学院上海生物化学所 梁卫平

**摘要** 本文介绍通过修改BIOS的办法来解决IBM-PC/XT与FWX-4675绘图仪联接的问题。

FWX-4675是一种六笔平面式绘图仪，并行接口信号名称与其他绘图仪一致。绘图仪直接与IBM-PC/XT打印机接口相联，工作时绘图仪总出错报警，这是由于绘图仪在接收命令过程中丢失数据所致。

其一，绘图仪接收数据时不是靠计算机发来的选通信号（Strobe），而是靠自身的回答信号（ACK）锁存数据。因此，送给绘图仪的数据一直要保持到ACK信号结束，数据保持时间比较长。接口信号时序见图1。其二，IBM-PC/XT的BIOS中打印机驱动部分（INT 17H中打印分支的框图如图2），有一个不理想的地方：先往输出口送数据，再判“忙”状态。由于IBM-PC/XT速度比绘图仪快，如果上一个数据被送到绘图仪接口而又未被ACK信号锁存时，计算机不管绘图仪忙不忙，马上又把下一个数据送往绘图仪。这样，上一个数据就被冲掉了，造成命

令传送失码现象。

针对这个问题，只要将INT17H加以修改，改成先判“忙”，等绘图仪不“忙”后计算机再往绘图仪发数据。具体办法如下：编一段汇编程序；用INT21H中25H功能调用修改INT17H的中断向量，使它指向程序的入口；用INT27H调用，将程序驻留在内存。这样，以后无论哪一级调用INT17H打印中断时，就去执行修改过的INT17H。此后IBM-PC/XT与FWX-4675绘图仪联接就正常了。

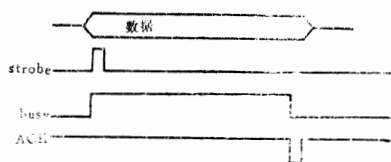


图1 绘图仪接口信号时序

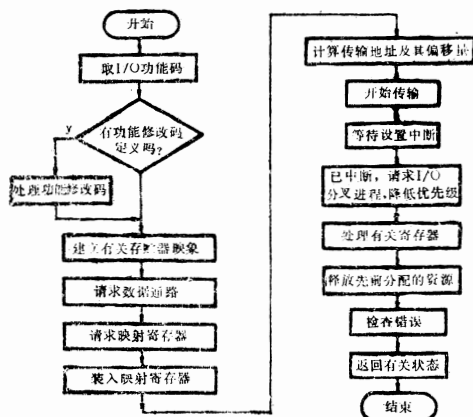


图3

## 五、结束语

对VAX机的进一步开发，使我们有机会作为普通用户深入到VAX/VMS的更深一层。同时，也为我们积累了一些有益的尝试经验。如果全国的广大VAX机用户都能做一些类似的工作，那么无疑将为现有的设备发挥更大的效益做出贡献。

### 参考文献

- 1 DMF32 User Guide
- 2 VAX software handbook
- 3 VAX/VMS device driver Programming Guide

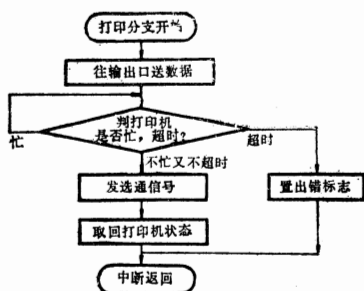


图2 ROM BIOS INT 17H打印分支框图

### 自编程序清单

```

CODE SEGMENT PARA PUBLIC 'CODE'
START PROC FAR
    ASSUME CS:CODE

```

```

A1:  STI
      PUSH    DS
      PUSH    DX
      PUSH    SI
      PUSH    CX
      PUSH    BX
      OR      AH, AH
      JZ      B2
      DEC     AH
      JZ      B8
      DEC     AH
      JZ      B5

```

```

B1:  POP     BX
      POP     CX
      POP     SI
      POP     DX
      POP     DS
      IRET

```

```

B2:  PUSH    AX
      PUSH    AX

```

```

AGAIN: MOV    DX, 3BDH; 取状态口
        IN     AL, DX
        TEST   AL, 80H; 判打印机“忙”
        JZ     AGAIN
        MOV    DX, 3BCH
        POP    AX

```

```

OUT     DX, AL; 将待打印数据送出
NOP
NOP
MOV     AL, 0DH
MOV     DX, 3BEH
OUT     DX, AL    ; 发选通信号
MOV     AL, 0CH
NOP
NOP
OUT     DX, AL
POP     AX

```

```

;
B5:  PUSH    AX
B6:  MOV     DX, 3BDH
      IN     AL, DX
      MOV    AH, AL
      AND    AH, 0F8H

```

```

;
B7:  POP     DX
      MOV    AL, DL
      XOR    AH, 48H
      JMP    B1

```

```

;
B8:  PUSH    AX
      JMP    B6

```

```

;
A2:  PUSH    CS
      POP     DS
      MOV     DX, OFFSET A1; 修改中
                                   断矢量

```

```

      MOV     AX, 2517H
      INT     21H
      MOV     DX, OFFSET A2; 驻留返回
      ADD     DX, 100H
      INT     27H

```

```
START ENDP
```

```
CODE ENDS
```

```
END A2
```

### 参考文献

- 1 《局部网络操作系统DOS高级技术分析》  
国防工业出版社, 张载鸿.
- 2 FWX—4675绘图仪说明书.

## 如何将反汇编出的程序写入文件

在实际工作中，为了破译或修改EXE文件，常需把EXE文件用debug调试程序进行反汇编，并将反汇编出的程序打印出来，以便仔细阅读。由于EXE文件比较长，又有数据段与代码段之分，这就使得用debug调试程序做这项工作显得十分麻烦。

为了一次性快速完成这项工作，我用了一种比较好的方法。今将这一方法介绍给读者。

### 一、如何确定EXE文件

#### 数据段和代码段的长度和位置

为了将EXE文件中的数据段和代码段准确地反汇编出来，我们必须确定数据段和代码段的存放位置和长度。这项任务可用debug调试程序来完成。用debug调试程序将要进行反汇编的EXE文件装入内存后，其存放格式如图1所示。

从图1可知，EXE文件数据段的起始地址为DS:100H，代码段的起始地址为CS:IP，文件的全长由BX: CX寄存器给出。

有了上述信息后，根据图1我们就可以计算出数据段和代码段的长度或终止地址。数据段长度LD的计算公式为：

$$LD = CS : IP - DS : 100H \quad (1)$$

代码段长度LC的计算公式为：

$$LC = BX : CX - LD \quad (2)$$

如我们要对fprint.exe进行反汇编，用debug调试程序的R命令，可看出CS=26，DS=26A5，IP=0000，BX=0000，

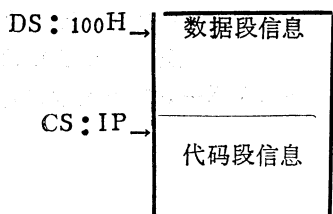


图1 EXE文件存放格式

CX=546。这样，LD和LC的具体计算方法为：

$$\begin{aligned} LD &= 26D3 : 0000H - 26A5 : 100H \\ &= 26D30H - 26B50H \\ &= 1E0H \end{aligned}$$

$$\begin{aligned} LC &= 0000 : 546H - 1E0H \\ &= 546H - 1E0H \\ &= 366H \end{aligned}$$

通过以上计算，我们可得出fprint.exe文件中，各段的长度或起始地址如图2所示：

### 二、利用管道功能将反汇编出的程序写入文件

算出了fprint.exe文件中各段的长度和起始地址后，就可以用edlin编辑程序建立一个给debug使用的命令文件。假设该文件的文件名为abc，根据图2，该文件的内容为：

D DS:100 2DF；用D命令显示数据段信息

U CS:O L 365；用U命令反汇编出代码信息

Q，退出debug 调试程序  
abc文件建立之后，我们就可以用管道功能将反汇编结果写入file文件。操作方法为：

type abc | debug fprint.exe > file

此命令的功能是：将type abc的输出信息输入给debug fprint.exe，再将debug fprint.exe的输出信息写入file文件中。这条命令执行完之后，就将fprint.exe文件的反汇编结果全部写入了file文件。此后，用ws就可以将file中的反汇编程序分页打印出来。

# IBMPC/XT故障一例—ROM

## 芯片接触不良

### 一、故障现象

和往常一样，机器上电自检（640K）OK。装载2.13F汉字系统，进入DBASE III环境。然后，在点“.”状态执行一些DBASE III命令，到此一切正常。然而，当打开数据库执行APPE命令时，屏幕内容突然消失，机器进入自检状态。

这时，自检过程异常：

1.反复自检：检测RAM芯片至160K或32K或576K不等便重新自检。

2.蜂鸣器伴有“嘀，嘀”声：自检或屏幕消失时，喇叭伴有连续不断的“嘀，嘀...”声。

3.偶尔能通过自检，系统引入内存，但不等DBASE III系统软件进入内存，机器便又转入自检状态。

4.冷热起动后，故障现象依然如故。

### 二、处理过程

由于近来病毒程序流行，面对机器反复自检，且伴有怪叫声自然联想到是否为病毒程序所致。观察到冷热起动后，故障现象并不消失，便可排除软件故障，即病毒程序的攻击。因为病毒程序故然可模仿自检过程且发出怪叫声，但在冷热起动之后，不可能立

即发作。

排除软件故障的可能之后，便转入硬件故障的分析。机器运行中断故障，可能是CPU、系统板电路、ROM、RAM等故障引起。RAM扩展卡往往松动造成Parity奇偶检验错，或死机或重新自检。384K扩展板插槽重插之后，故障现象没有变化。

取掉扩展板后，现象依旧。

下一步怀疑的对象是内存RAM工作参数、状态区不稳定，而造成上述故障。换掉系统板上BANK 0处9片RAM，即系统上前64KBRAM。故障仍然没有排除。

更进一步怀疑的对象便是ROM芯片。ROM芯片中的程序有错或不稳定，同样能导致上述故障。将两片ROM芯片重新插接好，重新打开电源，系统运行正常。故障排除，至此判定为ROM芯片使用时间长，接触不良而引起上述故障。

由于我们办公室式微机房环境条件不佳：温度、湿度、灰尘等因素问题以及维护条件不够理想，便容易导致上述故障。

建议同行们在保养机器时，定期用酒精清洗接口板插头处，轻轻按一按带插座式集成块，使其接触良好，避免此类故障导致的系统崩溃。

西南铝加工厂计算机中心

高国明 龚安平

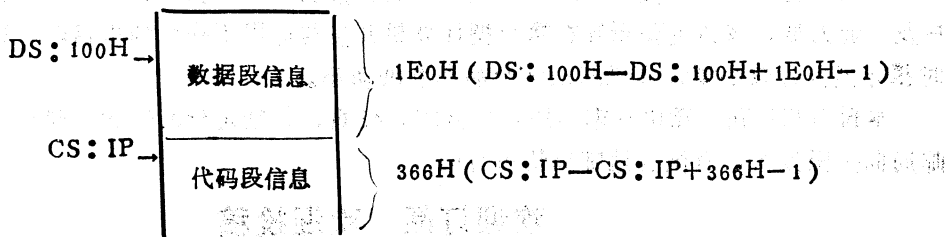


图2

南昌陆军学院计算机室

戴水贵

## 高速宽行打印机不走纸的检修

DEC公司中小型计算机上所配的LXY 22是一台功能很强的高速宽行打印机,为理想的批量拷贝输出工具,它使用并行接口,含有可更换的PROM多种字符集,每分钟可打600行 $\times$ 132个字符,还可作图形输出。我们单位的一台LXY22:在一次突然断电后再启动时,出现不走纸现象,且有糊味出现,操作员立即将其关掉。这时由于打印机本身不能工作,只能作脱机检查。

初步分析可能有这样两个问题:

1.有可能步进电机坏。

2.有可能驱动步进电机的驱动板上有问题。断开步进电机和驱动板(P600 Power Supply),测量四组线圈,发现电机未坏。那么按照《p600 Field Maintenance Manual》第87、89两页上的图55和图56检查流程图可知,驱动板上有问题,发现板子上R95电阻( $4.0\Omega$ , 3w)裂了一条缝,卸下测量,发现该电阻确实被击穿,换上一个 $3.9\Omega$ , 2w的电阻后再加电,不正常走纸两、三行后又将该电阻烧断了,此时我们分析,能烧坏此电阻的强电流肯定也会将与其相连的管子烧坏,我们将与其相连的Q<sub>4</sub>、Q<sub>5</sub>、Q<sub>12</sub>、Q<sub>13</sub>几个管子卸下测量,发现Q<sub>5</sub>和Q<sub>13</sub>已坏,这两个管子分别为TIP127和TIP122。代换管子为BU406和2N6042,前级放大所加的管子为3DG182,代换如图1所示。

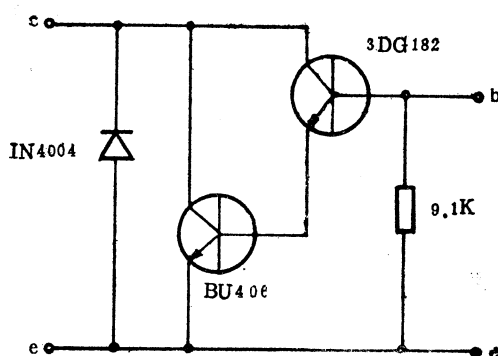


图1

当然,LXY22打印机的维护小问题还不少,如打印的字型模糊,这是因为打印头被油泥、纸屑所堵,只要打开清除头上污物即可,这是一般情况,如果还不行,那就该换打印头;还有,打印色带受阻问题,LXY22行打色带回转控制,是靠色带环上粘的锡片和传导棒的接通与否来控制的,一旦色带走完,锡纸和传导棒一接触,则控制回带。进口带由于其介质硬无受阻问题,但由于价格等原因,我们只能用国产带,但国产带质地稍软,再加上用手工将其缝在色带环上,这样易于卷带(叠带)受阻,一旦出现此问题,将传导棒底下的小槽用透明胶布或其它胶布填平,即可保护色带不受阻——即卷带(叠带)。

西安电子科技大学计算中心 赵元哲

## 征订启事

《微小型计算机开发与应用》是国内外公开发行的中等技术性刊物,面向广大计算机用户及专业人员。该刊主要报导在微小型计算机开发与应用工作中的丰富经验和成果,并将及时报导国内外微小型计算机的最新产品和发展动态。

本刊为双月刊,逢单月末出版,16开本,48页,每期定价0.95元。刊号6-87,由天津市邮局向全国发行,全国各地邮局均可订阅。

## 欢迎订阅 欢迎投稿

本刊地址:天津市河西区友谊路爱民道5号

邮政编码:300061

电话:33.3651