

B 72.



微小型计算机

# 开发与应用

MICRO-MINICOMPUTER  
DEVELOPMENT & APPLICATION



1991



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

发行：天津市邮局

出版：天津市电子计算机研究所

印刷：天津武清县长宏印刷厂

天津市计算机学会

地址：天津市河西区友谊路宾馆南道5号

订购处：全国各地邮局

邮政编码：300061

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 ISSN1001—8786

# 《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 张凤枝  
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 吴锦声  
房家国 张凤枝  
夏纪寅 夏业勋  
袁维本 曹东启  
黄 侃 黄宝良  
章渭臣 梅克定  
童宣明 裴少峰  
薛大中

## 1991年第1期目录

(总第51期)

### 应用实例

- 单片微型机在石油电脱水中的应用.....  
..... 郭云林 蒋利波 邓有训 (1)  
柴油机台架数据采集系统.....  
..... 夏 羽 丁光宇 (4)  
微型计算机在液相色谱梯度淋洗中的应  
用.....齐国良 王贵悦 (10)  
Z80微机在电子计价秤中的应用.....  
.....曹迁邦 (13)  
微机专用时钟在实时控制系统中的应用  
.....吕志民 (15)  
一个开关量信息的大面积采集系统.....  
.....林勇奇 (19)

### 计算机软件

- 模糊聚类实用程序与农业病虫害预测  
.....张国栋 王林 (22)  
51系列单片机多机通信的一种可纠错编  
码方案.....张加胜 (26)  
常见微机第四代工具性能的试验分析  
.....赵国瑞 (29)  
如何编制高质量的驻留内存程序.....  
.....张卫兵 (35)  
正交试验设计的计算机实现方法.....  
.....吴晓禾 赵 俊 (40)

### 经验点滴

- PC机软硬盘故障的排除.....肖伟民 (46)  
在程序中消除和恢复光标的方法.....  
.....姚 琥 宋光爱 (48)

## CONTENTS

### APPLICATION EXAMPLE

Application of Single-Chip Microcomputer in Electro-Dehydrating Process of Crude Oil.....	Guo Yunlin, Jiang Libo, Deng Youxun (1)
Microcomputer Data Acquisition and Processing System for Diesel Engine Test-Bed.....	Xia Yu, Ding Guangyu (4)
Application of Microcomputer in the Gradient Drip Washing of Liquid-Phase Chromatogram.....	Qi Guoliang, Wang Guiyue (10)
Application of Z-80 Microcomputer in Electronic Computing Price Scales .....	Cao Yubang (13)
Application of Microcomputer Special-Purpose Clock in Timing Control Systems.....	Lu Zhimin (15)
A Large Area Acquisition System of Switch Information.....	Lin Yongqi (19)

### SOFTWARE

The Practical Program of Fuzzy Classification and Prediction of Crop Diseases and Insect Pests.....	Zhang Guodong, Wang Lin (22)
An error-Correcting Coding Scheme for Multicomputer Communications of 51 Family Single-Chip Computers.....	Zhang Jiasheng (26)
An Experimental Analysis of the performance of Common Fourth Generation Tools on PCs.....	Zhao Guorui (29)
How to Compile a High-Quality Resident Memory Routine.....	Zhang Weibing (35)
Computational Method to the Design for Orthogonal Test.....	Wu Xiaohe, Zhao Jun (40)

### EXPERIENCE

Removals of Non-Circuit Troubles in PC Series Diskettes and Disks.....	Xiao Weimin (46)
Introduction to an Easy Way of Eliminating Cursors.....	Yao Hu, Song Guangai (48)



# 单片微型机在石油电脱水中的应用

空军雷达学院 郭云林 蒋利波 邓有训

**摘要** 本文介绍MCS-48系列单片微型机在石油电脱水控制设备中的应用,详细描述了硬件与软件的结构;并介绍了系统的特点,所达到的技术指标及解决的主要问题。

## 一、概述

从地下开采出来的石油,必须脱去其中的水份(国家规定含水量小于0.5%),其最后一道脱水处理工序是高压电场脱水。由于原油中化学成份,含水量等因素不稳定,在电脱水过程中电场会不断波动。为保证脱水油的质量和产量,电场需要的电压和电流必须控制在一定范围内,因此需要一种控制设备及时调整电场的电压和电流。可控硅自动调压限流控制柜就是目前广泛用于脱水现场的一种设备。

国内研制、生产和使用电脱水控制设备,已有二十多年的历史。但到目前为止,它们都还是应用模拟电子技术的,与现代化的技术相比已显得陈旧落后,同时也不能满足迅速发展的生产的需要。其主要缺点是:

1. 限流特性差。

2. 过流截止保护响应速度慢,常造成系统的高压硅堆甚至主可控硅过荷烧毁。

3. 由于主可控硅难以严格配对使用,运行中总存在直流成份,造成系统的高压变压器性能下降甚至烧毁。

4. 抗干扰能力差。

5. 故障率高。

6. 操作维护不方便。

根据上述情况和用户的强烈愿望,我们和江汉油田开发公司,武汉市电气控制设备厂密切合作,联合研制了《TZD—J型原油电脱水电脑控制柜》,成功地克服了原有设备存在的问题,并增加了新的功能:

1. 实现了理想的调压限流特性。

2. 过流保护响应速度降到20ms内,完全

避免了瞬时过载对主可控硅和高压硅堆的危害;并在过流截止时设有自动复位程序。一旦过流情况消失,系统便能自动恢复运行,无须人工干预。

3. 实现了主可控硅的不平衡处理,两个主可控硅不必挑选配对便可使用。经电脑调整主回路不平衡电压限制在0.5V以下;当其中一个可控硅开路时,10秒内将自动断电报警,从而有效地保护了高压变压器的正常运行与安全。

4. 设有软启动程序和顺序加电装置,消除了开关机对电网和控制系统的冲击。

5. 在硬件和软件上采取了较完善抗干扰措施,使系统在恶劣环境下也能稳定工作。所有这些都是原模拟控制电路难于实现的。

## 二、主要功能与技术指标

1. 输入电压: 单相50Hz, 380V/220V  
 $\pm 20\%$

2. 输出电压: 0~375V连续可调

3. 输出电流: 0~130A

4. 限流范围: 5~130A连续可调

5. 过流保护: 瞬时峰值200A, 响应速度小于20ms

6. 自动复位(软启动): 周期30秒, 软启动持续时间1分钟。

7. 不平衡处理:

(1) 允许两主可控硅触发时差: 2ms

(2) 调整后不平衡电压:  $< 0.5V$

(3) 报警响应速度: 不大于10秒。

## 三、系统硬件结构

一般石油电脱水成套设备的结构框图如图1所示。

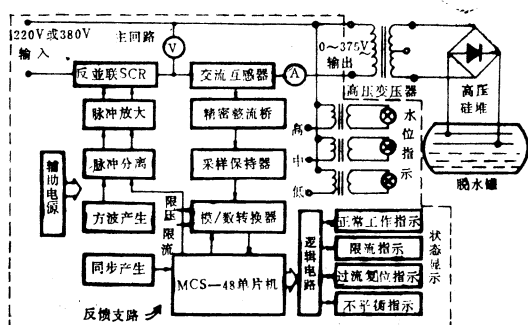


图1 电脱水成套设备框图

《TZD—I型原油电脱水电脑控制柜》的结构框图则如图中虚线框内部分所示。这里只介绍控制柜部分，它主要包括主回路、反馈控制支路，工作状态显示和辅助电源等四部分。其中以MCS—48单片微型机为核心的闭环反馈支路是中心环节，其基本工作原理是利用单片微型机实时地控制主可控硅的导通角，对输出电压电流进行合理调节。这部分的基本工作过程如下：

**1. 信号采样** 交流电流互感器取出主回路电流采样值，经精密整流后送至采样保持器变成离散样本值。

**2. 模—数转换** 将离散信号和限压，限流的给定直流信号输入到A/D转换器的不同通道上，把模拟量转成数字量送入电脑存储器，以待数据处理。

**3. 触发脉冲的产生与放大** 电脑进行适当的数据处理后，发出合适的触发脉冲，与方波相与后，经放大分两路分别控制主回路中可控硅的导通。

#### 4. 功能控制与工作状态显示：

(1) 通过数据处理后，若输出电压不超过限定值，输出电流小于给定值，则“正常工作”指示灯亮。

(2) 如果负载减小，电流增大到限流给定值，则“限流指示”灯亮。说明工作电流已达预定最大值。

(3) 如果负载继续减小，电路电流将

不再增大，电脑控制触发脉冲的相位减小，不断减小可控硅的导通角以降低输出电压，维持电流不变。当移相达到规定的最小导通角时，继续减小相位已无意义，我们规定这种状态为“缓变过流”，电脑立即拆除触发脉冲，并发出声光报警，30秒后，电脑将自动启动系统重新工作。

(4) 在系统工作的全过程中，如发生瞬间过流，电脑立即停发触发脉冲，并发出“过流复位”声光报警。如上述。

(5) 在系统工作过程中，如两个主可控硅有不平衡情况，电脑将自动检测并调节正负半周触发脉冲的相位，使消除直流分量；如果有一个可控硅损坏开路，电脑立即停发触发脉冲，同时发出“不平衡”声光报警以保护高压变压器。

## 四、系统软件结构

为了实现机器的上述功能与技术要求，我们研制了如下程序。

**1. 主程序** 它决定着整个软件系统的工作，其框图如图2所示。主要包括采样子程序，计算子程序，不平衡处理子程序和软启动子程序。同时充分利用微型机的中断功能，定时/计数溢出中断主要用于触发脉冲；外中断则用于复位子程序，并将可控硅的控制时间赋给定时器，然后启动定时器工作。

**2. 采样子程序** 取样电路从主回路提取电流信号后，与限压、限流给定直流信号一并送给模—数转换器，在微型机控制下将模拟信号转变为数字信号。其程序框图如图3所示。

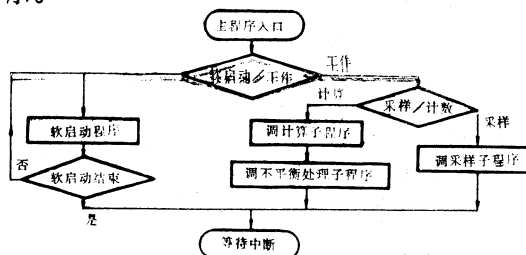


图2 主程序框图

3. 计算子程序 此程序主要用于计算输出电流的平均值, 检测工作电压, 电流是否符合脱水现场规定的指标, 并根据实际情况调整触发脉冲的相位, 框图如图4所示。

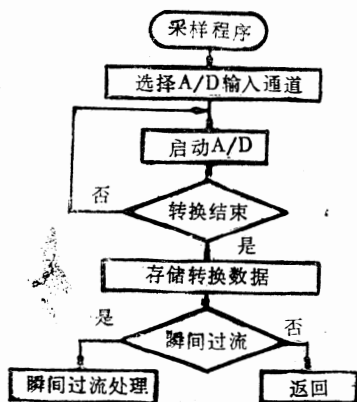


图3 采样子程序框图

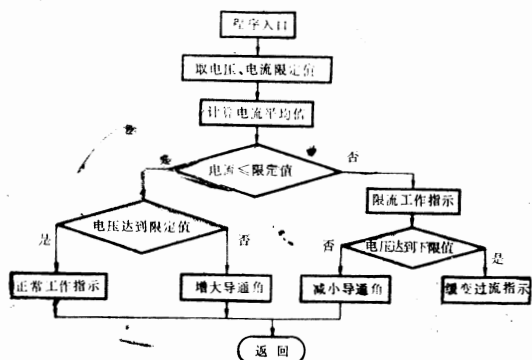


图4 计算子程序框图

4. 不平衡处理子程序 所谓不平衡就是两个可控硅触发时差不一样, 导致输出电压、电流正负半周不对称, 致使主回路存在直流分量。因高压变压器一次侧直流内阻极小 ( $0.03\Omega$  左右)。故即使直流电压不大也可能产生很大的直流电流, 使高压变压器性能下降甚至烧毁。原来设备中要求采用两个完全一样的可控硅, 一起使用。但这给实际应用、维修带来麻烦。过去模拟电路无法解决这一问题。现在我们利用单片机进行不平衡检测, 一旦发现存在不平衡, 即调整某一触发脉冲的相位, 直到平衡为止。这样不仅能确保无直流成份, 而且两个可控硅亦不必严格挑选配对。同时如果其中一个可控硅烧坏开路, 电脑会自动检测出来, 并在10秒内断电报警, 保护了设备。该程序框图如图5所示。

5. 定时/计数溢出中断服务子程序该程序根据触发脉冲的发出时间完成发出触发脉冲的工作, 并确定触发脉冲的宽度(约1ms)并在脉冲发出后进行采样等工作。框图如图6所示。

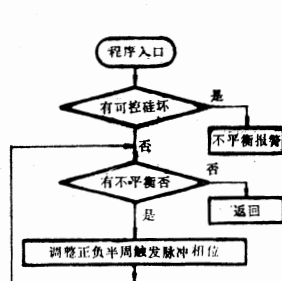


图5 不平衡处理子程序

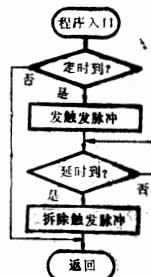


图6 溢出中断子程序

6. 外中断服务子程序 将同步脉冲作为INT中断信号, 使单片机每10ms响应一次外中断, 执行一次中断服务程序后, 修改指针到主程序入口。这样能保障系统程序不致由于外界干扰而失控。另外该程序的作用是将可控硅的导通时间赋给定时/计数器, 然后启动定时/计数器工作。框图如图7所示。

7. 过流/复位处理子程序 用于过流报警和自动复位(重新启动系统工作)。框图如图8所示。

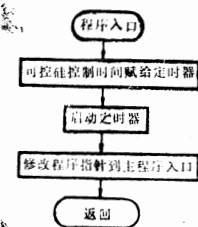


图7 外中断子程序

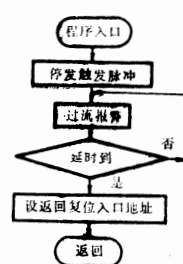


图8 过流/复位子程序

由上可见, 采样子程序解决了各采样信号的模—数转换和瞬时过流问题; 计算子程序解决了正常工作与限流工作状态及缓变过流的处理问题; 不平衡处理子程序解决了可控硅的不平衡调整与报警问题, 过流/复位子程序解决了停机报警和自动复位问题; 两个中断子程序解决了触发脉冲的发出, 宽度及相位控制问题。从而使整个设备的运行完全符合工艺生产的各种情况, 也就是整个工

# 柴油机台架数据采集系统

核工业计算机应用研究所 夏羽 丁光宇

**摘要** 本文介绍的柴油机台架数据采集系统, 是进行柴油机产品试验时所采用的自动数据采集及处理系统。它主要完成对表征柴油机性能特征的重要参数(如扭矩、转速、油耗), 以及反映柴油机工作状态的监测参数(如冷却水温度、润滑油压力)等多种数据的实时采集和处理。它采用上、下位机的分步式结构, 使得数据处理能力及贮存能力大大加强, 不但可输出实时采集数据表, 还可输出试验特性表及试验特性曲线, 并为建立数据库档案打下了基础。文章中将对此系统的功能及技术特点加以介绍, 并重点描绘了系统的软、硬件特性。

## 一、前言

柴油机台架试验是柴油机生产厂对柴油机进行性能检验, 产品定型及各种科研性试验的重要手段, 而现代化的试验室、先进的测试手段和测试设备, 则对柴油机试验的可靠性和准确性起着决定性的作用。以往所采用的常规试验台, 测量精度低, 数据采集和

处理的效率低, 越来越不能适应现代柴油机试验研究的要求。因此, 产品试验的自动化, 已成为柴油机生产厂家十分关注的问题。

我们在济南柴油机厂的配合下, 研制了《柴油机测试台数据采集系统》。系统在设计中, 广泛调查了国内外同类系统的情况, 吸取各家之长, 重点以实用性强、功能丰富、测试精度高、可靠性强为出发点, 得到了用户的信赖, 使该系统很快投入使用。在长

艺生产过程完全在微电脑的严格控制之下。所以该软件系统是一个完善的实际的系统。

地端。通过以上措施可保障系统在恶劣条件下稳定工作。

## 五、抗干扰措施

主要分两个方面考虑:

**1. 软件方面** (1) 采用同步脉冲外中断方式运行程序, 保证程序与控制回路电流相位的严格同步, 使系统程序不致由于外界干扰而失控。(2) 在瞬间过流检测和数据采集时, 采用数字滤波方法, 提高了系统的抗干扰能力。

**2. 硬件方面** 首先对辅助电源采用两级三端稳压块串联稳压, 增加必要的吸收与隔离电路, 增强电源的抗干扰能力。其次, 将微机发出的触发脉冲与一方波逻辑与后分割成独立的两路。可避免主可控硅由于外界干扰产生误触发。另外, 为避免地电流干扰, 使每一功能块均有自己的地线而后接到公共

## 六、结束语

本设备充分利用MCS—48系列单片机处理能力与中断功能, 节省了硬件, 解决了模拟电路难于解决的问题, 实现了电脱水控制系统的更型换代。使系统性能稳定, 功能齐全, 操作简单, 维修方便。目前已投入批量生产并在油田使用, 通过一年多的在线运行, 用户反映良好, 是一个较成功的新产品。

### 参考资料

- 1 《MCS—48系列单片微型计算机原理与应用》北京工业大学编 1987.5.
- 2 《微型计算机控制技术》, 人民邮电出版社 1985.8.



期使用中,特别是在对Z6V190—2型柴油机进行500小时耐久试验中,本系统运行正常,实时采集处理了大量数据,精度均满足要求,收到了良好的经济效益及社会效益。

## 二、系统功能及技术特点

该系统的主要功能有:

1.可同时对柴油机(增压非增压)常规性能试验的59个参数,如:转速、功率、油耗、烟度、漏气、各种温度、各种压力等。及排放试验的5个参数,共64个参数进行自动采集,一次试验,最多可进行13个工况。

2.控制台上配有小键盘及CRT显示器,试验参数、试验条件、试验环境等。可通过人机对话方式键入,并可利用功能键对各种操作进行控制。

3.全部工况数据可在CRT显示器上分帧显示,显示数据每6秒刷新一次,并且对这些数据能够进行定时和随机的打印输出,打印为汉字表格形式。

4.可对油压、油温、水温、转速四个关键监控参数进行越限自动报警。越限值在试验前通过人机对话方式设定,报警方式有声、光报警及屏幕上的该参数频闪。

5.整个试验完成后,可将全部试验数据(最多13个工况数据)传送至上位机(IBM—PC机)。上位机将对全部数据进行处理,并以汉字表格形式打印输出。还可作出本次试验的试验特性表及特性曲线(包括负荷特性、速度特性、推进特性、排放特性)。

## 三、系统构成及工作原理

系统采用上下位机结构,下位机选用国际流行的STD总线结构。由不同功能的模板组成面向总线型的微机数字系统,STD总线专为工业控制设计,具有很强的现场适应能力。总线设计为插槽式,功能电路模板化、整个系统可随意组合,配置灵活、扩展性强、易于调试、维护,下位机配备的外设有:

专用小键盘,SG—1000 12'黑白监视器,M2024打印机。

上位机选用IBM—PC/XT微型计算机或其兼容机。它有丰富的系统软件及数据处理软件支持,可对数据进行优化处理,上位机配有一台M2024(M1724)打印机及一台SR 6602绘图仪。并且配有RS—232C串行接口,以便与下位机进行通讯。

### 1. 下位机硬件配置

硬件配置框图见图1。

下位机为面向总线的开放型系统,整个系统由不同的功能模板组成,它包括:CPU板、内存扩展板、CRT显示接口板、SP IO板、CTC板、A/D板、脉冲信号调理板、电平信号调理板、模拟信号调理板。

#### (1) CPU板及内存扩展板

这是系统工作的核心,CPU板中带有Z80 8位微处理器。工作时钟为2M,内存扩展板则使得系统的存贮空间扩至为64K。

#### (2) CRT显示接口板

它以CRT控制器MC6847为核心,带有6K显示存贮器、可接12'黑白监视器其最大分辨率为 $256 \times 192$ 。

#### (3) SPIO 串行/并行接口板及CTC定时/计数接口板

SPIO板为系统提供了串/并行输入输出通道,板中的2片Z80—PIO提供了32条I/O线。其中12条作为键盘的专用行列扫描线,2条作为报警信号的输出控制线,板内还有一片Z80—SIO,提供RS—232C标准的串行通道。系统用来进行上下位机的通讯,CTC板中共3片Z80—CTC控制器,系统只采用了两片,用来对脉冲输入信号进行计数(详见脉冲信号的调理)。

#### (4) 模拟信号的调理及A/D转换

系统中输入模拟信号(温度、压力等)共56个,它们经过一次仪表后,分别以不同的形式进入系统,如:电流、电压、电阻,并且信号均很微弱,因此必须对它们进行调理,使其成为统一形式的信号。信号调理的

主要任务有：提供多路输入、单路输出的信号通道；滤除电子噪声和浪涌信号；提供固定或可选的信号放大增益；对输入量进行电量归化，转换成统一的一定范围内的电压信号。

模拟量参数经过调理之后，输入给A/D板，在程序的控制下，分别对其进行采样，经实测，模拟量的采集精度可达0.2%。

#### (5) 脉冲信号及电平信号的调理

系统中的部分参数是以脉冲量或电平形式输入的（如转速、油耗、扭矩等）。并且输入信号亦不规范。因此也必须进行相应的调理，调理的主要内容包括：光电隔离，放大、整形等。

光电隔离采用了TIL117光电耦合器、实现了现场与计算机系统的完全电隔离。

#### (6) 报警电路及键盘

声光报警利用两片CA555组成振荡电路，配合驱动电路，分别驱动蜂鸣器及发光二极管，形成声光效应。

键盘的平面图如图2所示。

0	1	2	3	4	5	6	7
8	9	B	C	D	G	N	O
CTL	.	-	←	→	↓	V	Z
RST			打印	清屏	消音	油耗	排放

图2

它的内容是根据人机对话的需要及操作功能设置的。专用性强，操作灵活方便，各键的使用方法将在工作原理中阐述。

键盘阵列的4行8列共12根线，通过提拉电阻接至PIO输入输出线上，用软件对其按键进行识别。

### 2. 系统工作原理

系统的工作进程可分为四个部分：

- 人机对话。
- 数据采集及数据处理。
- 显示及打印等多种输出处理。
- 上位机进行数据处理。

这几部分工作都是通过不同的功能键进

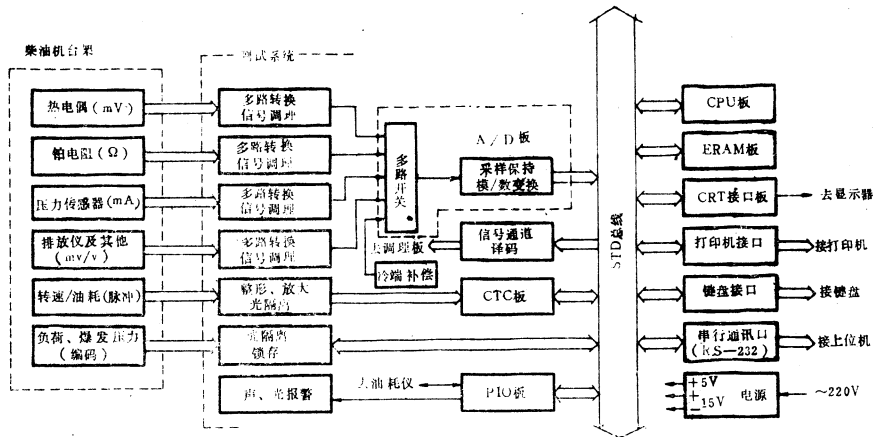


图1 下位机系统硬件框图



行控制的,一次试验可进行多个工况的数据采集及输出处理,最后将所有工况数据送至上位机。在数据采集,采集周期为6秒,并且在一个周期内,系统对所有被检测量进行循环多次(现为6次)采集,去掉最大最小值后,取平均值为有效的采集值,然后再对其进行集中处理(数字滤波,工程量转换等),处理后的数据直接送至显示器显示,整个巡检过程中可通过功能键完成打印、工况变换、油耗测量等功能操作。使操作员能对系统的工作进行控制。

下面介绍两个较特殊量的采集原理:

#### (1) 转速信号的采集

转速信号是速度传感器送出的一组正弦波,其频率与转速成正比。系统中采用了对脉冲信号计数的方法测取频率。具体为:将输入的正弦波进行放大整形,使其成为较规范的脉冲信号。送至CTC板。通过CTC计数器进行计数。CPU在程序设定下,在每一巡检周期的起始处启动计数。1秒钟后读取计数值。经过换算即得到转速值。

#### (2) 油耗测量

油耗参数包括三个量:油耗量、油耗时间及油耗率。系统采用了定油量,测油耗时间,从而计算出油耗率的原则。在测油耗时间中,根据一次仪表油耗仪的工作原理,保留其全部功能。仅从仪器中引出二根信号线,即油耗仪的“T测”及“清零”信号。“T测”作为油耗计时的门控信号“清零”作为测量油耗的启动信号。将这两个信号送至调理板进行相应的处理。即通过对“T测”微分获得它的前沿脉冲及后沿脉冲,做为油耗计时的起动信号和结束信号。对于“清零”信号则配以一些辅助电路后接至PIO,从而可利用键盘上的油耗键,来启动油耗的测量。

另一方面,选择时钟脉冲分频后为计时脉冲,将其送至CTC。在“T测”及“清零”信号控制下,即可完成油耗的测量,由于计时时钟周期为 $32.768\text{ms}$ (CPU时钟为 $2\text{MHz}$ )。计

数误差最多为一个脉冲,即时间测量误差最大为 $33\text{ms}$ ,与油耗时间的实际值( $\geq 30\text{s}$ )相比,其误差 $<0.2\%$ ,满足系统的精度要求。

## 四、系统软件

### 1. 下位机软件

#### (1) 主要设计思想及主流程图

本系统下位机软件均用Z80汇编语言编制。应用程序共约14K,加上程序使用的字库、各类区、表等,共占用了大约40K内存。

整个软件在设计中以表格化、模块化为主导思想,尽量做到逻辑简单,功能强,程序短小精悍。软件按功能分成四大模块,下面分别加以介绍:

#### 1) 键盘—显示管理模块

这部分的功能是利用键盘及显示器进行人机对话,支持实验前的参数键入工作及功能键的操作。

#### 2) 数据采集处理模块

这部分程序的功能是实现在巡检周期内各路检测数据的采集、换算、越限报警及一些简单的数据运算等操作,并以工程量的形式提供一组完整测量数据,供屏幕显示及打印。

#### 3) 外设接口管理模块

这部分内容包括:显示模块、打印模块、发送模块(至上位机),以完成采集数据的输出处理。

#### 4) 中断处理模块

本模块包括:时钟中断服务程序、油耗计时中断服务程序、键盘中断服务程序,以完成系统中三个中断源的中断处理。

系统的主流程图如图3。

#### (2) 软件设计中的几点说明

#### 1) 采样周期的设定

作为实时检测系统,巡检周期是系统设计的一个重要参数。本系统考虑到被采集的量多为变化缓慢的物理量(如温度、压力),经折中考虑,采样周期定为6秒。为了减小误差,每路数据在一个巡检周期内共巡回采

集6次,取其平均值作为该路信号的有效数据(转速与油耗除外)。采样周期和实时时钟保持同步。

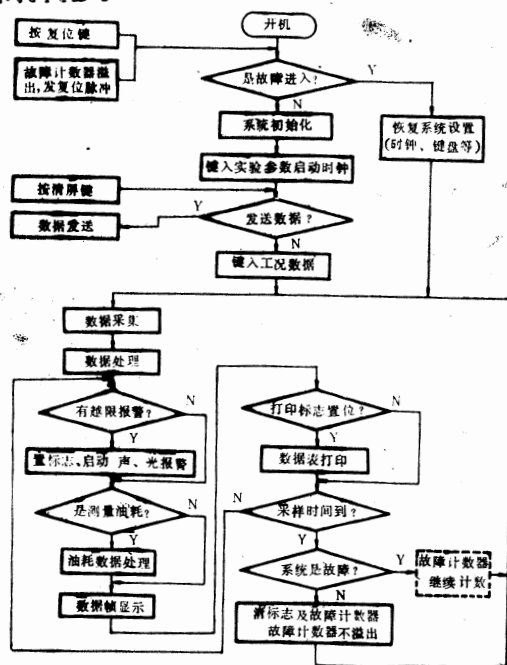


图3 下位机主程序流程图

## 2) 中断响应方式及中断源

实时系统的典型处理方式是中断响应方式,从本系统特点考虑,共设置了三级中断,中断优先级排队依次为:a 时钟。b 键盘c 油耗计时。CPU 对中断源优先级的判别利用了STD总线的链排队功能,根据功能模板在总线杆槽中的排列顺序,就可决定其优先级的高低,在Z80中断方式2的支持下,中断处理过程十分灵活、简便。

## 3) 模拟信号的工程量转换及线性化处理

工程量转换中信号的线性化处理,根据信号类型不同分别按下列公式处理:

① 铂电阻类:  $R_t = R_0(1 + At + Bt^2)$

② 热电偶类:  $T = a_4 E^4 + a_3 E^3 + a_2 E^2 + a_1 E + a_0$

③ 烟度/漏气:  $y = a_2 x^2 + a_1 x + a_0$

(二次曲线拟合)

④ 对于线性信号则按线性值公式处理:

$$Y = (Y_{\max} - Y_{\min})(X - X_{\min}) / (X_{\max} - X_{\min}) + Y_{\min}$$

其中:

Y——测量参数的工程值

$Y_{\max}, Y_{\min}$ ——参数工程值终点、起点。

$X_{\max}, X_{\min}$ ——测量值(A/D输出)的终点、起点。

X——参数的测量值(A/D输出)

## (4) 油耗率

油耗参数的测量在前面已初步介绍,这里着重介绍油耗率的测量。油耗率是根据测得的油量及油耗时间计算而得的。它是柴油机性能试验中的一项重要指标。根据国家及部颁标准,油耗率必须以连续测量的3—4次平均值作为有效值,且必须满足

$$\frac{\text{平均油耗率} - \text{单油耗率}}{\text{平均油耗率}} \leq 0.5\%$$

因此,在软件中,首先连续测量三次油耗参数,经计算后满足以上要求,则其平均值为有效值。若不满足要求,再测第四次,第四次再不满足要求,则判为测量故障。给出故障标志,若满足要求,则打印、显示出测量数据。

## (5) 汉字化表格打印

下位机的打印输出为汉字表格形式,这也是本系统的一大特点。在进行打印软件的编制中,考虑到用Z80汇编语言编制此程序。主要问题就是程序冗长、工作量大。为了节省内存,优化程序,采用了建立打印字符表的方法。其设计思想为:将所有要打印的字符按打印顺序列表,其中加入相应的标志码(包括汉字标志、数据标志、回车标志、缺省处理标志等),而程序只需循环不断取字符表的内容,并作出判断、未进行相应的操作。这样,一个很短的程序就可打印出一张很大的表格,使得10K 多的程序缩短到 1K 多。而字符表只占3K,并且只要修改字符表及程序中的某些参数,就可打印出任意的表格。打印程序流程图见图4。

在进行打印的汉字化处理上,从经济性 & 实用性出发,系统中没有配置汉卡,全部由软件进行处理。方法是:建立专用汉字库(只有打印中要用到的汉字),从IBM—PC机CC



DOS汉字库中调出其标准汉字码,再给所有库中的汉字另行编码(用ASCII码中80H以上的码),与字符码(标准ASCII码)一起编入字符表中。由程序进行判断,并对汉字进行专门的输出处理(配24针打印机)。

实践证明,这样编制的打印软件,投资小,配置灵活,节省内存,打印效果也十分理想。

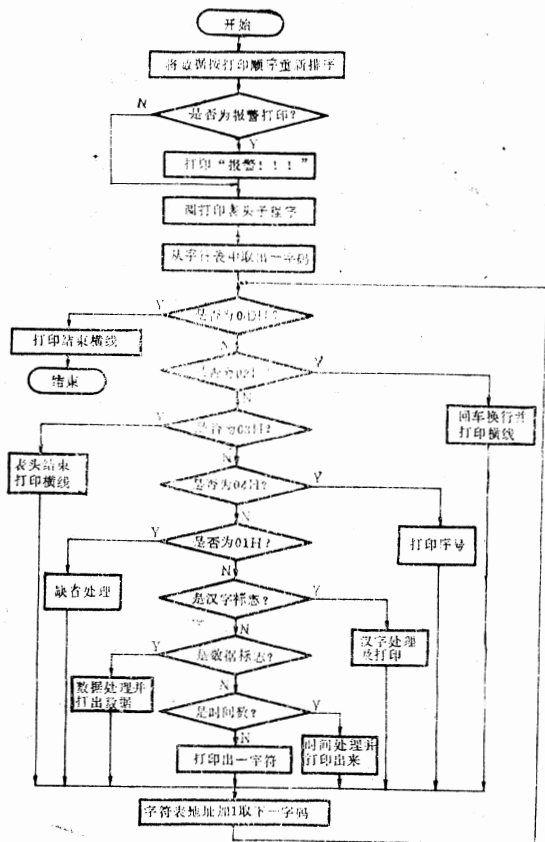


图4 下位机数据表格打印程序框图

## 2. 上位机软件

上位机软件采用模块化结构。主要由四部分组成, a 接收数据; b 数据整理、计算; c 打印全部试验数据表及各类特性表; d 绘制特性曲线。上位机软件框图如图5。

这里主要阐述特性曲线的绘制:

在计算机数据采集和处理领域中,绘制试验曲线的基本方法是采用代数多项式拟合法,但是在内燃机性能试验中,人们发现,用代数多项式拟合法绘制的曲线,常常与实际试验曲线形态不符,在本系统的特性曲线

绘制中,也遇到了同样的问题。

天津大学钱起云和许君风同志提出了可适用于绘制任何类型内燃机性能曲线的方法——几何相似插值法,我们试用了这种方法,取得了较满意的效果。

### (1) 几何相似插值法基本原则

几何相似法绘制试验曲线的最大特点,就是绘制出的曲线与实测数据构成的自然形态是几何相似的,为了保证这个特点,几何相似插值法绘图遵循了如下四个原则:

- 1) 所绘曲线通过所有实测点
- 2) 所绘曲线在各实测点处的切线,由连接各实测点的折线在对应点的平均走向所确定。
- 3) 在两个相邻测点间,用3次式函数进行连接。
- 4) 所绘曲线上各处具有连续的一阶导数。

根据这个原则,可总结出如下四个绘图步骤:

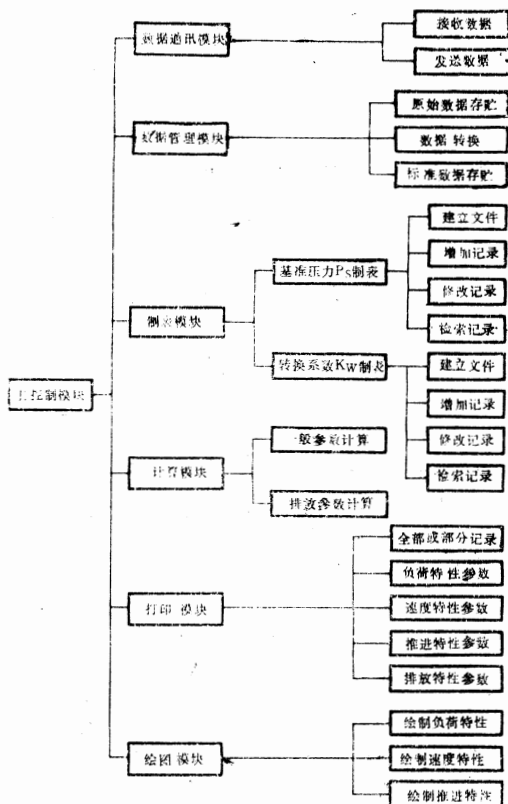


图5 上位机软件框图

# 微型计算机在液相色谱梯度淋洗中的应用

大连市计算技术研究所

齐国良

中国科学院大连化学物理研究所

王贵悦

## 一、问题的提出

目前, 高效液相色谱已成为化学、化工、生物、医学和环境保护等科学领域中重要的分析技术, 是化学家用以解决分析各种课题必不可少的工具。这种方法是利用混合物中各组分在称之为固定相和流动相之间具有不同的分配系数, 使之混合物从固定相的一端加入并随流动相向前移动的过程中各组分在固定相上有不同的保留而被分离。过程是, 溶质加在色谱柱头随流动相一起进入色谱柱 (见图1), 接着在固定相和流动相之间分配, 分配系数大的组分 (如组分C) 不易被固定相滞留, 流出色谱柱较早。若一含有多组分的混合物进入色谱系统, 则混合物中各组分便按其两相间分配系数的不同, 先后流出色谱柱。

但是, 当一种样品混合物的容量因子K范围很宽 (极性范围宽, 分子体积相差较大), 等度冲洗 (即恒定浓度的流动相组成冲洗), 一则时间太长, 二则后面的峰太扁平不便检测。这样就需要梯度冲洗 (即溶剂浓度随时间增长增加)。采用梯度冲洗操作, 需要两台泵, 在冲洗过程中逐步增加混合物溶剂中的极性成份, 以提高单位时间内的峰容量。

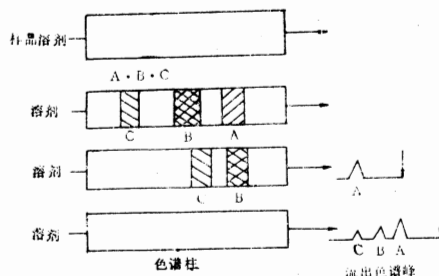


图1 色谱分离过程

①计算二端点外的各测点处的曲线斜率值。

②计算连接除二端点外的各测点之间的三次多项式函数的系数。

③计算与曲线两端点相接的3次多式函数。

④由各测量点间的三次多项式函数计算出各线段上的插值点, 根据以上原则上及绘图步骤, 即可得到较满意的特性曲线。

## 六、结束语

本系统于一九八七年底研制结束, 并通过验收, 经过1年多的试用, 用户反映良好, 于一九八九年十月通过鉴定, 在鉴定会上以其通用性强, 功能丰富, 性能稳定等特点,

得到了专家们的一致好评, 并在核工业部系统中, 获得了部级科技进步三等奖。

在系统研制完成之后, 我们在其基础上又研制了Ⅱ型、Ⅲ型等系列产品, 不仅用于柴油机行业, 而且可用于有类似工作环境的各种工业现场, 这些系统, 在结构上进一步改进, 使其更为合理, 并经济实用, 在功能上, 又增加了下位机显示汉字化处理。显示分辨从原有的 $256 \times 192$ 增至 $640 \times 200$ , 并全部为汉字表格形式, 对不同的参数采集, 不同的打印机输出, 都配上了专门的软件, 方便了用户的使用。

## 参考资料

钱起云, 许君凤《可用于绘制任何类型内燃机性能曲线的新方法——几何相似插值法》。



对于连续梯度冲洗方式，可以用简单的数学公式表示：

线性梯度： $c_g = t/t_G$

凸形梯度： $c_g = (t/t_G)^n$

凹形梯度： $c_g = 1 - (1 - t/t_G)^n$

式中  $c_g$  为溶剂组分的体积分数， $t$  为梯度已执行的时间， $t_G$  为总梯度时间， $n$  为大于0的整数，表示梯度曲线的陡度。曲线如图2所示。

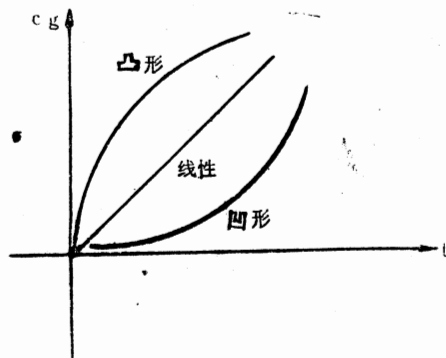


图2 梯度曲线

本文所介绍的就是利用微型计算机 (TP-801A, 按予先设计好的程序在不同的时间内，按不同的溶剂浓度分别对两台泵 (A 泵、B 泵) 进行梯度控制，把两种溶剂打入一只混合器内，混合均匀后再进入色谱柱。从而组成一个液相色谱梯度冲洗微机控制系统，见图3。

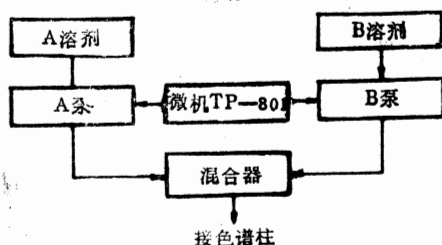


图3 梯度冲洗系统框图

## 二、系统的硬件结构

该系统利用 TP-801A 单板机原有的功能，重编监控程序，增加一片 CTC 接口芯片。改变 CTC 输出脉冲频率，分别控制两台泵的转速，使溶剂强度随时间的增长而增加，实现梯度冲洗目的。

该系统不仅具有梯度冲洗工作方式，同时还具有 A 泵独立工作方式，A 泵、B 泵独立工作方式，A 泵、B 泵固定比例工作方式。共计四种工作方式。并具有流量自动校正功能。

系统的硬件框图如图4所示。TP-801A 单板机原有的六段显示器前四位用来显示时间分、秒。后两位用来显示工作方式。键盘重新进行了定义，把它分为数字键和功能键。功能键又分为参数设定键和运行键。参数设定键用来进行各种参数的设定，当参数设定完后，按下运行键，系统就起动运行。数字键用来根据不同的功能键，完成参数的输入。

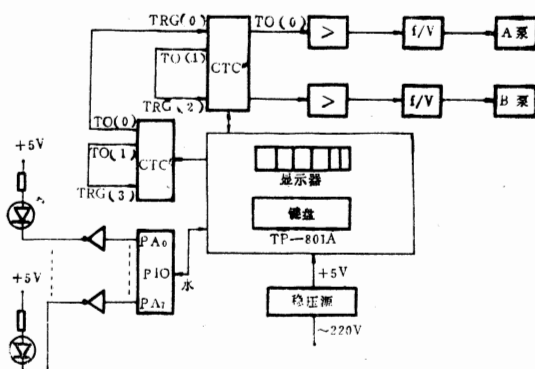


图4 系统硬件框图

PIO 的输出是用来在梯度工作方式下，控制七个发光二极管，指示系统梯度运行阶段时间数 ( $T_1 \dots T_i$ )。

增加一片 CTC 接口芯片，同原有的一片结合起来，输出两路脉冲控制 A 泵，B 泵工作。CTC'0 通道设为定时器，它的输出端接至 CTC''0 通道的输入端 (设为计数器)，输出端用来控制 A 泵。CTC'1 通道设为定时器，它的输出端接至 CTC''2 通道的输入端 (设为计数器)，输出端用来控制 B 泵。CTC'2 通道与 CTC'3 通道用来产生秒中断。

## 三、系统软件结构

在系统硬件已确定的情况下，主要是依靠软件的设计和编程来实现各种参数的设

定,判断,计算、控制和计时等功能。整个软件结构可分为两大部分,1.是主程,2.是中断服务程序。主程主要是由初始化程序;键盘扫描程序;判断程序;计算程序和实时时钟管理等程序组成。占有内存空间4K,用2片2716。

初始化程序主要负责两片 CTC,一片 PIO接口芯片的编程,控制状态设置,中断矢量设定及起动软件时钟。

PIOA通道设为字节输出方式。

键盘扫描程序主要是来输入各种控制参数。并通过判断程序来分析是数字键还是功

能键,并转入相应的处理程序。

计算软件是按照输入的工作方式,对各种参数进行计算并通过计算来改变CTC定时器或计数器的时间常数以改变CTC输出的脉冲频率达到控制A泵或B泵的目的。计算程序还包括浮点四则运算及平方运算程序,BCD码转浮点码程序、浮点码转二进制码程序等,供需要时调用。

时钟管理软件是为A泵或B泵在不同的工作方式下的工作周期而设计的。时钟显示使用单板机的四位显示器,时钟管理采用中断方式。系统程序流程图见图5。

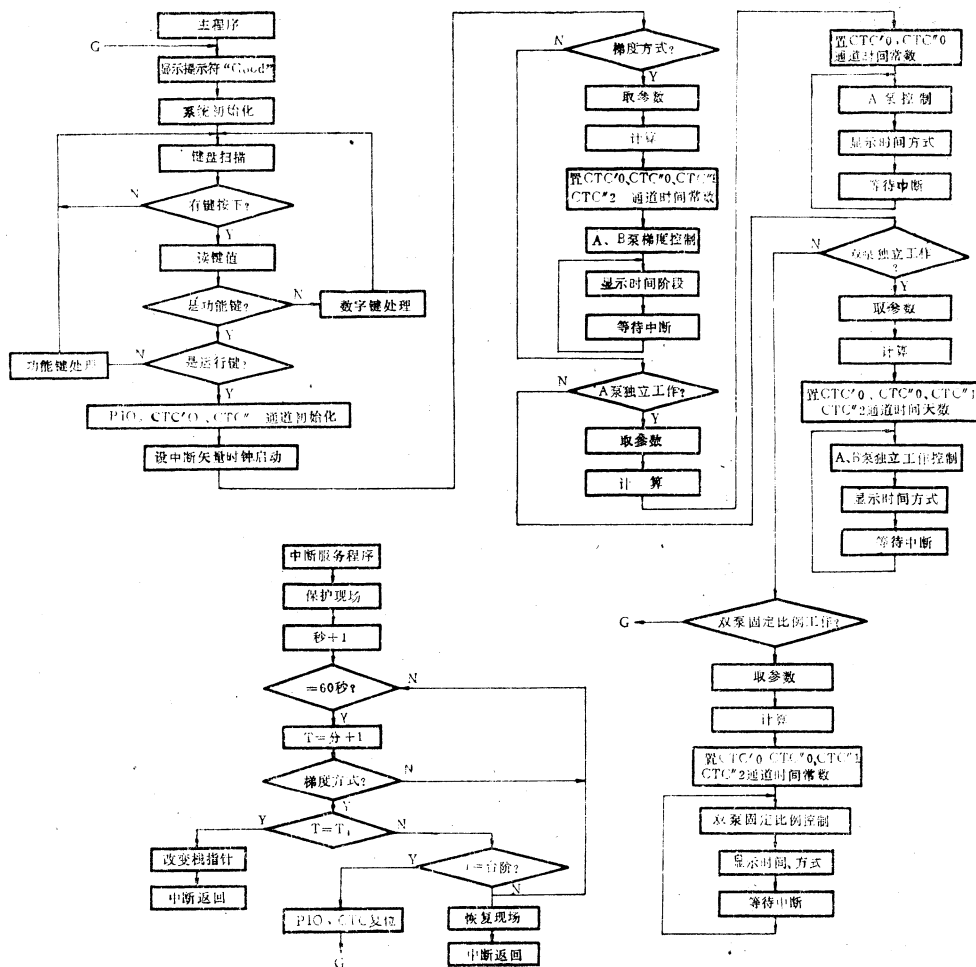


图5 系统程序流程图

# Z80微机在电子计价秤中的应用

杭州商学院 曹迁邦

**摘要** 本文介绍Z80微处理器在国产电子计价秤的使用中提高性能、增加功能、降低成本的设  
计思想。

## 一、引言

笔者利用TP801单板机作为开发工具，现已研究成功了电子计价秤样机。该秤计量指标达到“三级精度”，性能也还稳定；与国内进口组装秤比较，计价功能增强，功能更全；整机单台器件成本（不包括称重传感器和机壳）约530元。

经使用，浙江省计量局实验工厂，杭州南洋传感器厂和海盐传感器厂的称重传感器产品，都能达到整机技术指标。

## 二、Z80专用微机板的组成

Z80专用微机板采用了CPU、PIO、2716各一片，用两片2114构成八位、1K字节内存，用一片138作为地址译码器，也用4MHz晶振。显示和键盘相关电路完全与TP801-线路相同。总共18片集成电路，安装在特制的印刷电路板上，组成一个专用Z80微处理器。八只绿色LED显示器（或16只、双面显示）分两组安装在面板上；左边四只显示重量值，右边四只显示金额值（或单价）。

## 三、Z80微机在称重时的职能

根据电子秤工作原理和计量性能，Z80微处理器应完成如下主要职能：

1. **模数变换。**电子秤中将放大的电压变成数字量，常见有三种方案：选用专用集成电路，如DS-230中采用了IC  $\mu$ PC 646D；用电路组成双积分器，结合微处理器，共同完成A/D转换；笔者采用了V—F变换器，通

过接口电路，在Z80微处理器控制下，实现A/D功能。

2. **零点跟踪。**由于称重传感器、放大器和电压、频率变换器的温漂或失调漂移，都会造成零点漂移，只有依靠软件跟踪，才能达到自动校零。

3. **数据处理。**这包括将接口电路中采集的重量数字分时接收和软件滤波两大任务；然后进行二—十转换及显示重量值。

下面围绕这三个职能作一概述。

硬件电路获得了频率之后，Z80微处理器实质上只要完成测频功能。接口电路如图1所示。图中清零脉冲和门控脉冲由PA<sub>0</sub>和PA<sub>1</sub>输出。

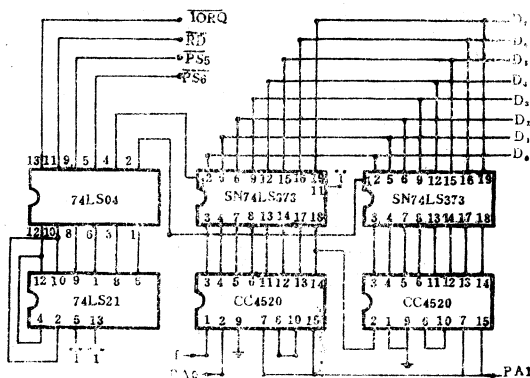


图1 接口电路原理图

近期，我们应用CTC来完成测频功能。这样，可省去接口电路，成本降低（CTC价格明显下降）。

在电子秤软件设计中，采取“净重法”，将系统引入的漂移（温漂和时漂）或皮重作为零漂值处理，不断采集并暂存在内存单元

中,称重时,当前重量采集值减去前时刻零漂暂存值,得到净重,完成软件校零。这种方法还降低了对放大器过高的性能要求。

程序中,重量数据采集两次,然后取平均值。这虽然降低了A/D转换速度,但可以达到消除采重数据的随机误差,提高了数据测量精度。尤其称重速度不高条件下,应当采用软件滤波。

## 四、靠软件,增强功能

电子计价秤除了计量指标之外,另一类则是功能指标。利用软件,不需增加成本,却能完善计价性能,设计特殊计价方式,扩充实用功能,进而满足各种场合的特定要求。

### 1. 利用查表技术

诸如肉类,糖果类等商品价格较稳定,因此笔者设计了代码找单价,完成自动计算功能。很明显,一位代码可对应10种商品单价,二位代码可对应100种商品单价。利用查表法,对不变单价采用人们习惯的十进制数,将它们固化在2716指定的内存单元中,编制软件,寻找对应单价既可以简化程序,又缩短了执行程序时间,还简便了操作。

国产秤按下某数字键后,接着查寻 $PA_2$ 口,当 $PA_2=1$ 时,先显示对应单价,经显示延时然后自动完成金额运算并显示出金额值。该秤设置的“单价复原键”,当按下时,能重显单价,之后又自动完成金额再运算及显示。

查表技术还可以将某些按质论价的商品,找出升降子价系数,转化为查表法。这样,可以不用表格列价,进而加快了计算速度,也减轻了劳动强度。

某县收购棉花,按等级每增减一级,升降子价为1.78%,尾数按二舍八入,三、七作五处理。笔者选择合适等级作基价,然后计算出等级单价,通过查表技术转化为代码找单价。这样直接在机内获得按质论价的价格,并可直接参与计算,显然能提高收购计

价自动化程度。

### 2. 增加功能

由于秤的最大称重范围是有限的,设计具有重量累加和称重次数累加的功能很有必要。笔者将这功能汇编成子程序,规定某功能键给予命令的解释和执行。累加重量值由右边六只显示器显示;最左边二只显示累加称重次数。

### 3. 稳定空秤零点

国外秤的秤盘如有水份,使用者没有复零,则都会出现空秤非显全零。笔者设计成最小称重为5g(6Kg称重秤),各种因素造成空秤漂移只要不超过4g,则让空秤显零。这种以降低最小称重值来换取稳定显示空秤显零的办法,是否可取,供讨论。

## 五、软件支持硬件,简化电路设计

国产新秤要进入批量生产阶段,降低成本是至关重要的,为此,该文也讨论这个问题,提出降低成本,开拓新应用的想法。

### 1. 硬件软件

诸如清零脉冲,门控脉冲、时序波形等全部由软件产生,可以减少电路器件。

### 2. 软件辅助功能

本秤单价不用固定显示器,扩充累加金额最大显示达到六位。这既可类同组装机秤的性能;更方便八位微机的电路设计和程序编制,器件明显减少。借助秤中“单价复原键”,能重显单价供核对用。

### 3. 扩展功能

接口电路设有重量数据锁存和三态门输出,使重量输出的数字量能分时输出给外设,并保证电子计价秤自身的完整性。这样,配置上打印机、与大型计算机联机通讯,构成功能全、处理能力大的粮库收购管理、商场营业管理和自动快速收购系统提供了可能。

### 4. 扩大应用



# 微机专用时钟在定时控制系统中的应用

中山大学 吕志民

**摘要** 本文介绍微机专用时钟MM58167的主要功能,以及在定时控制系统中的应用。它克服了一般可编程实时时钟在市电掉电时停止计时的缺点,使得系统的自动化程度更高。专用时钟能产生多种时间信息和中断控制信号,为系统实现不同时制的自动转换和各种定时控制提供了方便。文章详细介绍了专用时钟和单片微机组成的多路用电控制器,在高校集体宿舍用电管理与控制中的应用情况。

## 一、概述

实时时钟和日历是控制系统的重要设备[1]。是系统的计时依据。目前,多数微机控制系统的计时基准,是利用可编程定时器的定时中断[2],或利用市用交流电每20ms发出的中断信号[3],用编程的方法产生实时时钟。但是,这些方法的缺点是:当掉电时,时钟停止计时。上电后,时钟必须重新校准,系统才能正常工作。这些缺点影响了设备的自动化程度,给用户带来不便。

用微机专用时钟 MM58167作为控制系统的计时基准,能有效地解决上述问题,使系统有较好的控制效果。

专用时钟使用了CMOS工艺,功耗小,在2伏低电压下也能工作[4]。它适用于总线结构的微机系统,地址、数据总线可直接和CPU相接。在计时和控制方面,有如下的功能。

a. 输出八种计时信号:千分秒、百分秒(十分秒)、秒、分、时、星期、日、月,日历在四年内有效。

b. 产生八种可屏蔽中断信号:计数器和

锁存器比较、每十分之一秒、每秒、每分、每时、每天、每星期、每月。

c. 能产生备用中断,为自动切换工作电源提供控制信号。

d. 可判断从计数器中读得的数据是否有效。

e. 用32768HZ石英晶体产生振荡基频,走时准确。

作者使用专用时钟和 MCS—48系列单片微机,研制成“全自动多路用电控制器”,在高校集体宿舍微机管理和控制用电中,收到了良好的效果,使宿舍用电量减少近50%,根本上杜绝了违章用电和超指标用电现象。

## 二、地址代码和功能

专用时钟内部有多个可寻址存贮单元,每个有效长度为8位。有些单元,其有效长度不足8位。如星期几计数和锁存单元,有效长度为3位,贮存相应十进数的1~7,其余位称为无效位。无效位在写操作中不予承认,在读操作中被置成逻辑“0”这些存贮单元的地址代码和功能如表1所示。

表中,几项特殊功能分述如下:

本秤软件采用模块结构,诸如四则运算、数制变换、显示程序等设计成子程序。这样,设计不同称重范围的电子秤、电子计数秤等,可以共享通用性子程序,为工厂尽快

制造系列化产品、承接特殊计算功能的电子秤,能缩短软件编程工作量,加快试制进程,进而尽快地满足各种应用。

表1

地 址					代 码 (十六进制)	功 能
A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>		
0	0	0	0	0	0 0	千分秒计数器
0	0	0	0	1	0 1	百分秒/十分秒计数器
0	0	0	1	0	0 2	秒计数器
0	0	0	1	1	0 3	分计数器
0	0	1	0	0	0 4	时计数器
0	0	1	0	1	0 5	星期计数器
0	0	1	1	0	0 6	天计数器
0	0	1	1	1	0 7	月计数器
0	1	0	0	0	0 8	千分秒锁存器
0	1	0	0	1	0 9	百分秒/十分秒锁存器
0	1	0	1	0	0 A	秒锁存器
0	1	0	1	1	0 B	分锁存器
0	1	1	0	0	0 C	时锁存器
0	1	1	0	1	0 D	星期锁存器
0	1	1	1	0	0 E	天锁存器
0	1	1	1	1	0 F	月锁存器
1	0	0	0	0	1 0	中断状态寄存器
1	0	0	0	1	1 1	中断控制寄存器
1	0	0	1	0	1 2	计数器复位
1	0	0	1	1	1 3	锁存器复位
1	0	1	0	0	1 4	状态位
1	0	1	0	1	1 5	"GO" 命令
1	0	1	1	0	1 6	备用中断

### 1. 计数器、锁存器复位

对某个计数器复位, 必须对“计数器复位”单元进行写操作。复位的具体对象, 由写入的数据决定。复位对象和相应数据如表2所示。锁存器复位与此类似。

### 2. 状态位

微机对计数器进行读操作时, 若计数器内容正发生变化, 所读得数据是错误的。此现象在千分秒、百分秒/十分秒、秒等高频率变化存储单元里, 经常发生。此时, 状态位

会被置成01状态。检查“状态位”存储单元的内容是00还是01, 可知读得的数据是否有效。

表2

数 据 总 线								复位数据 (BCD)	复位对象
D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>		
0	0	0	0	0	0	0	1	0 1	千分秒
0	0	0	0	0	0	1	0	0 2	百分秒/十分秒
0	0	0	0	0	1	0	0	0 4	秒
0	0	0	0	1	0	0	0	0 8	分
0	0	0	1	0	0	0	0	1 0	时
0	0	1	0	0	0	0	0	2 0	星期
0	1	0	0	0	0	0	0	4 0	天
1	0	0	0	0	0	0	0	8 0	月

### 3. "GO"命令

对"GO"命令存储单元进行写操作, 可使千分秒、百分秒/十分秒、秒三个计数器单元复位, 而其它的低频计数单元内容不受影响。把低频计数单元内容预置好后, 到达预定的时间, 执行"GO"命令的写操作, 能使专用时钟和实际时间同步。

## 三、中断控制

专用时钟有 8 种可屏蔽中断控制, 结构如图1所示。

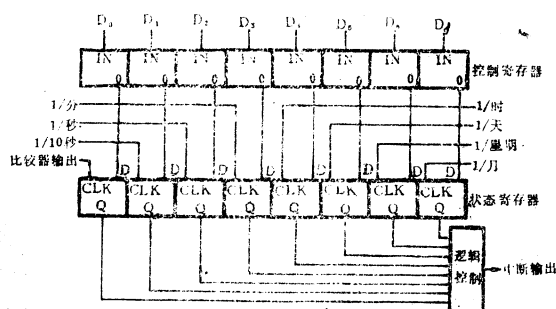


图1 中断控制结构框图

中断控制寄存器是屏蔽寄存器，用于控制8种中断的输出。给控制寄存器某位写入逻辑“1”，表示允许相应的中断输出，若写入逻辑“0”，则禁止相应的中断发生。控制寄存器是只写寄存器，不能读出其内部状态。

中断状态寄存器由8只D触发器构成，其时钟脉冲输入端和8种中断源相接，中断条件得到满足时，输入端的正跳变，使D触发器Q端电平等于D端电平。若D端被控制寄存器置成逻辑“1”，则中断输出端跳变至高电平，发出中断请求信号。状态寄存器只能读，不能写。

8种中断源中，比较器输出是指8个计数器和8个锁存器内容对应相同，比较器正跳变触发中断。其余的中断分别表示每1/10秒、每秒中断一次等等。

锁存器最高两位写入逻辑“1”，则高位字节成为不需关注的字节。在和相应计数器进行比较时，不需关注字节不参与比较。例如“时”锁存器写入C5H，其高位字节成为不需关注字节，在和“时”计数器比较时，只有低位字节参与比较。因此不管计数器内容是5还是15，比较结果都视为相同。利用此技巧，可使时钟在5时和15时都能产生中断。不需关注字节，也可以出现在低位字节上。

中断请求被响应后，对状态寄存器进行读操作，能使状态寄存器和中断输出复位。

#### 四、备用中断

备用中断是芯片在低压供电状态下产生的中断。在图2的电路中，当P.D.端(Power Down)变成低电平时，芯片进入低压供电状态，由备用电源供电。此时，所有数据线都处于高阻状态，8种可屏蔽中断全部禁止，但内部振荡电路仍在工作，时钟继续计时。在这种状态下，唯有备用中断能够产生。

备用中断产生的条件是：

1. 对“备用中断”存储单元进行写操作，第0位要写入“1”。

2. 8个计数器和相应的锁存器逐一比较，内容相同，比较器发生跳变。

由于写操作只能在正常状态下进行，所以条件1要在进入低压供电状态之前完成，而条件2则留在低压供电之中执行。

若对“备用中断”存储单元第0位写入数据“0”，则备用中断禁止。

备用中断为工作电源的切换提供控制信号。在低压供电时，若备用中断已经产生，备用中断输出端变成低电平。当主电源恢复正常供电时，晶体管进入饱和状态，专用时钟改由主电源供电。

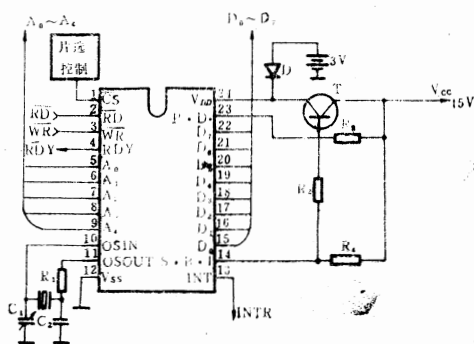


图2 微机专用时钟典型应用

### 五、专用时钟在定时控制系统中应用

在一些高等院校里，由于用电管理手段落后，集体宿舍浪费用电和违章用电屡禁不止。不但浪潮了能源，加重学校经费开支，而且违章用电对人身也不安全。最近作者使用单片微机和专用时钟，研制了“全自动多路用电控制器”，有效地解决了集体宿舍超标用电和违章用电的问题。

多路用电控制器具有如下功能：

a. 自动化程度高。可以实现无人管理。能定时通断房间和公用场所的电源。午夜后，公用场所用110伏串压供电。

b. 限制各房间使用电气设备的功率，超





# 一个开关量信息的大面积采集系统

广西南宁铁路电务段微机工区 林勇奇

**摘要** 本文介绍一个以简单的8行×8列基本矩阵为基础,采用分组扫描方式,用少量硬件实现开关量信息大面积采集的数据采集系统,该系统的特点是硬件电路省,扩展方便。文章以采用 Z80 PIO为输入/输出接口,采集量以1024个对象为例,介绍了这一数据采集系统的设计思想及电路的工作原理。

## 一、引言

对开关量信息的采集,通常采用一对一的采集方式,即一个点对应着一个采集对象。这种采集方式采集对象越多,硬件电路就越庞大。在实际运用中,开关量信息的采集,许多是利用继电器和控制按钮的空闲接点或采用继电器来实现的,在这种情况下,可以用少量的硬件实现大面积的信息采集。下面以采用并行接口片 Z80PIO,信息采集量为1024个对象的采集电路为例,说明这个电路的设计思想。

## 二、硬件电路及其工作原理

硬件电路见图1,采用矩阵采集方式,分组扫描。电路中使用两个并行输入/输出接口芯片Z80PIO,其中PIO(I)用于8×8基本矩阵,A口作为信息采集的行输入口,B口用作列扫描信号的输出口。PIO(II)的A、B两口用作分组选择驱动的输出口,将1024个采集对象分成16组,每组64个采集对象。每组取一个采集对象复连到8×8基本矩阵中的一个点,使得基本矩阵中的一个点对应着16个采集对象,通过对分组选择驱动的

### 3. 时制的转换

近几年,我国开始使用两种时制;北京时制和北京夏令时制。许多单位在不同时制里执行不同的作息制度。为了使控制系统的功能更加完善,应该加入时制自动转换的功能。专用时钟提供多种的时间信息,为编写时制转换程序提供了方便。时制转换规则是:每年四月中旬第一个星期天凌晨2时进入北京夏令时制,时钟由2时调为3时,开始执行夏令时作息时间。九月中旬第一个星期天凌晨2时回复北京时制,时钟由2时调回1时,开始执行北京时作息时间。

时制的自动转换,使系统自动化程度更高,保证在无人看管情况下,能长时间工作。

微机专用时钟以其先进生产工艺和多种功能,为微机实时控制系统提供了准确的连续时间信息和多种定时控制手段。它的应用,对提高微机控制水平和完善控制功能起到积极作用。

### 参考资料

1 P.B.Deshpande & R.H.Ash, Elements of Computer Process Control, Instrument Society of America, 1981.

2 吕志民,微型计算机实时时钟的自动校时,《自动化技术》,1987.3.

3 吕志民,65系列实时控制器及其在生态试验中的应用,《电子与自动化》1988.5.

4 MOS/LSI Databook for Data Sheets, National Semiconductor Corporation.

## 五、结束语

控制, 分时选中其中的一个。

采集信息时, 计算机选中PIO ( I ) 的 16 根分组选择驱动线中的一根 ( 低电平有效 ), 使得挂在这一组上的64个信息能进入采集电路, 而别的组上的信号被隔离。选择PIO ( I ) 的B 口的一根列选择线 ( 高电平有效 ), 这时未被选中的列选择线输出低电平使对应矩阵节点上的二极管截止而被隔离。这时, 当采集对象的接点闭合时, 矩阵电路中所对应的矩阵节点的电位被分组驱动线上的低电平拉低, 相应的行扫描电路上的三极管基极电位为低, 三极管截止输出高电平由PIO ( I ) 的A 口接收。反之, 采集对象的接点断开时, 对应的矩阵节点电位被列选择线上的高电平抬高, 相应行扫描三极管基极被加上高电平, 使三极管饱和导通, 输出低电平由PIO ( I ) 的A 口接收。每一时刻, 只有所选中的某一组、某一列上的 8 行信息被计算机采入。

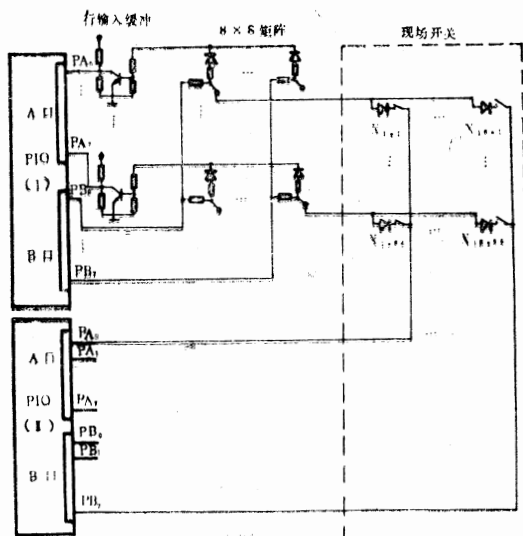


图1 硬件电路原理图

### 三、信息采集程序

设DATABUF为信息存放缓冲区首址, PIO ( I ) 、( I ) 各口的地址分配如表1:

表1

PIO ( I )	PIO ( I )
□A数据寄存器 80H	□A数据寄存器 94H
□A控制寄存器 82H	□A控制寄存器 96H
□B数据寄存器 81H	□B数据寄存器 95H
□B控制寄存器 83H	□B控制寄存器 97H

信息的采集采用中断采集方式, 则中断服务程序如下:

```

ZD: LD A, 0FH          ; PIO初始化
    OUT (83H), A        ; 将PIO (I) 的B口、
                        ; PIO ( I )
                        ; 的A口、B口为输出方
                        ; 式
    OUT (96H), A        ; 将PIO (I) 的A口置
                        ; 为输入方式
    OUT (97H), A
    LD A, 4FH          ; 置数据缓冲区首址
    OUT (82H), A        ; 置分组驱动初始值
    LD HL, DATABUF-1
    LD DE, 01H
FZ: LD A, E
    CPL A
    OUT (94H), A        ; 输出分组驱动信号
    LD A, D
    CPL A
    OUT (95H), A        ; 置列扫描次数
    LD B, 08H          ; 置列扫描初值
    LD A, 01H
CZ: PUSH AF
    OUT (81H), A        ; 输出列扫描信号
    INC HL              ; 调整缓冲区指针
    IN A, (80H)         ; 输入信息
    LD (HL), A          ; 将采入的数据存入数
                        ; 据缓冲区
    POP AF
    RLC A               ; 调整列扫描信号数值
    DJNZ CZ             ; 未采完一组信号返
                        ; 回, 否则转下一组
FZ1: LD A, E           ; 调整分组驱动信号数
                        ; 值

```

```

CP A, 80H
JR Z, FZ2
RLC A
LD E, A
LD A, D

```



# 模糊聚类实用程序与农业病虫害预测

安徽省计算中心

张国栋

六安地区植保站

王 林

**摘要** 本文介绍的模糊聚类分析程序在PC/XT机上开发,用FORTRAN—77语言完成,源程序5000行,最大执行模块140K字节。该程序已自1986~1989年成功地应用于六安地区三化螟测报实例,实现了农业病虫害预测预报超前准确量化。

## 一、模糊聚类预测模型

设一个预测目标 $y$ 的一个发生年份为一个样品,影响该目标的主导因子组 $(X_1, X_2, \dots, X_m)$ 为表征样品的指标。设有 $N$ 年资料( $N-1$ 年的历史资料,1年预测年资料),算法对 $N$ 个发生年作模糊等价划分。通过把预测年划入某一相似历史年组,达到预测目的。

不同的样品有不同的指标值,定义指标间的差异程度为样品距离,对距离作反义变换,得相似系数,定义域 $[0, 1]$ 。构造模糊相似矩阵

$$R = \begin{pmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ r_{21} & r_{22} & \dots & r_{2n} \\ \dots & \dots & \dots & \dots \\ r_{n1} & r_{n2} & \dots & r_{nn} \end{pmatrix}$$

其中 $r_{ij}$ 为第 $i$ 年与第 $j$ 年的相似系数。

显然 $R$ 满足:①自反性  $r_{ii}=1 \quad \forall i$ ;

②对称性  $r_{ij}=r_{ji}, \quad \forall i, \forall j$ 且 $r_{ij} \in [0, 1]$ 。

定义模糊相似矩阵乘法规则:类似于普通矩阵乘法,但元素相乘为两者取小者,元素相加为两者取大者。如 $C=A \cdot B$ ,则

$$C_{ij} = \max_{k=1}^n [\min(a_{ik}, b_{kj})]$$

各类文献已经证明,对 $R$ 作有限次自乘

$$\text{后得 } R^*, R^* = \underbrace{R \cdot R \cdot \dots \cdot R}_1$$

且 $R^* = R^* \cdot R = R^* \cdot R \cdot R = \dots$

$R^*$ 满足自反性、对称性,且满足

$$\textcircled{3} \text{传递性 } \min(r_{ik}, r_{kj}) \leq r_{ij}$$

称为模糊等价矩阵,根据任意指定的阈值 $\lambda (0 \leq \lambda \leq 1)$ ,将 $R^*$ 截为普通等价矩阵 $R'$ ,

$$r'_{ij} = \begin{cases} 1 & r_{ij} \geq \lambda \\ 0 & r_{ij} < \lambda \end{cases}$$

$R'$ 唯一地确定了样品(年份)空间上的一个分类。其直观意义是: $\lambda$ 作为相似尺度,凡模糊相似系数达到此水平,认为两年份相似,划入一组,否则认为不相似。 $\lambda$ 取值越大,“相似”的意义越严格。由于传递性质,可能多个年份互为相似。任一发生年必划入且只划入一个相似组。

## 二、原始数据

六安地区水稻三化螟主要危害在八月下旬,当年五月发预报。

预测目标 $Y_4$ :全年主害代发生量级别(1—5级)

前兆因子 $X_{13}$ :当年元月份平均气温( $^{\circ}\text{C}$ )

$X_{16}$ :上年第三代卵孵高峰日(7月31日为第0天)

$X_{18}$ :冬后加权平均活虫密度(头/亩)

$X_{22}$ :油菜占虫源田面积百分比(%)

$X_{26}$ :上年单晚占水稻面积百分比(%)



$X_{33}$ : 稳定通过16℃至4月25日平均相对湿度(%)

从64年至86年共23年资料,  $X_{13}$ 、 $X_{16}$ 、

$X_{26}$ 分别缺68、69、77年(—10为缺项标志), 程序判断后实际计算取20年。见附表1。

附表1 原始数据

A.D.	Y 4.	X13	X16	X18	X22	X26	X33
1964	5.00	2.80	16.50	125.00	4.46	20.91	82.00
1965	5.00	3.80	11.20	149.00	4.15	12.31	73.00
1966	5.00	3.50	15.60	330.00	3.45	11.41	83.00
1967	4.00	1.00	12.50	86.00	3.53	11.00	78.00
1968	5.00	1.90	10.00	169.00	3.83	-10.00	80.00
1969	5.00	.10	-10.00	54.10	4.44	12.45	87.00
1970	4.00	1.00	17.00	54.00	3.51	12.44	76.00
1971	5.00	2.30	17.70	31.00	3.99	13.44	69.00
1972	5.00	1.90	11.80	194.00	6.25	19.49	87.00
1973	5.00	2.70	16.20	22.00	9.46	19.04	83.00
1974	5.00	1.10	16.80	140.00	7.40	17.81	74.00
1975	5.00	3.90	13.90	365.00	8.75	11.97	78.00
1976	5.00	2.50	17.00	206.00	7.97	13.55	68.00
1977	3.00	-10.00	17.70	128.00	8.99	11.28	76.00
1978	5.00	3.10	18.80	78.00	10.33	10.96	69.00
1979	5.00	3.00	12.20	230.00	7.06	8.93	78.00
1980	4.00	2.50	15.30	105.00	12.57	15.85	60.00
1981	4.00	.90	23.80	27.00	25.72	9.75	67.00
1982	3.00	2.70	17.10	36.00	29.66	7.57	73.00
1983	3.00	2.30	18.20	48.00	27.91	5.36	67.00
1984	3.00	.30	22.00	43.00	27.02	4.65	75.00
1985	4.00	1.80	17.90	18.00	34.05	4.60	67.00
1986	3.00	3.20	17.80	19.00	32.25	3.94	72.00
N =	23	22	22	23	23	22	23
E =	4.35	2.20	16.23	115.53	12.47	11.76	74.87
S =	.81	1.06	3.26	95.36	10.46	4.73	6.84
CV =	.19	.48	.20	.83	.84	.40	.09
Xmax =	5.00[64]	3.90[75]	23.80[81]	365.00[75]	34.05[85]	20.91[64]	87.00[69]
Xmin =	3.00[77]	.10[69]	10.00[68]	18.00[85]	3.45[66]	3.94[86]	60.00[80]

### 三、模糊相似矩阵

为消除不同的量纲对计算结果的影响, 在定义相似系数前, 对原始数据作标准化处理。

$$\text{令 } X'_i = \frac{X_i - \bar{X}}{S} \text{ 其中 } \bar{X} \text{ 为原始数据的均}$$

值,  $S$  为标准差, 以后在不致混淆时,  $X'_i$  仍写作  $X_i$ 。

程序提供 9 种相似系数定义, 本文选用加权欧氏距离系数。

定义距离系数  $S_{ij} = \sqrt{\sum [W_k(X_{ik} - X_{jk})^2]}$

其中  $X_{ik}$  为第  $k$  个因子第  $i$  年数据,  $W_k$  为  $k$  个因子的权系数,  $S_{ij}$  为第  $i$  年与第  $j$  年的距离。

定义相似系数  $r_{ij} = 1 - \frac{S_{ij}}{S_{max}}$

其中  $S_{max} = \max_{i,j} \{S_{ij}\}$

权系数的意义是明确的, 在距离系数中, 相对重要的因子通过权系数放大贡献。权系数由用户选取, 程序负责对键入的权系数作为归一化处理, 即满足  $\sum_{k=1}^m W_k = 1$

在本例,  $W = (0.191, 0.132, 0.152, 0.276, 0.222, 0.027)$

模糊相似矩阵见附表 2.A。

#### 四、模糊等价矩阵与聚类结果

$R$  经 4 次自乘得模糊等价矩阵  $R^*$ , 见附表 2.B。

在  $R^*$  中, 除了对角线元素, 最大相似系数为 0.86, 最小 0.58, 平均 0.65。

取阈值  $\lambda = 0.71$  时, 分成 6 个相似年组, 每组给出发生年公元号及该年  $Y$  的实际发生值 (采自数据文件而非计算结果)。1986 年为被预测年, 无相应的  $Y$  值, 记为 -10, 划入第 6 组, 认为与 1982.1983.1985 年相似, 取该组  $y$  的平均值, 3.33 为予测值, 事后验证, 与实发级别一致, 预报准确。(此后 1987.1988.1989 连续三年应用模糊聚类实用程序预测结果均与实发级别吻合)

取  $\lambda = 0.73$  时, 分成 7 组, 原第二组又被一分为二, 聚类结果见附表 3。

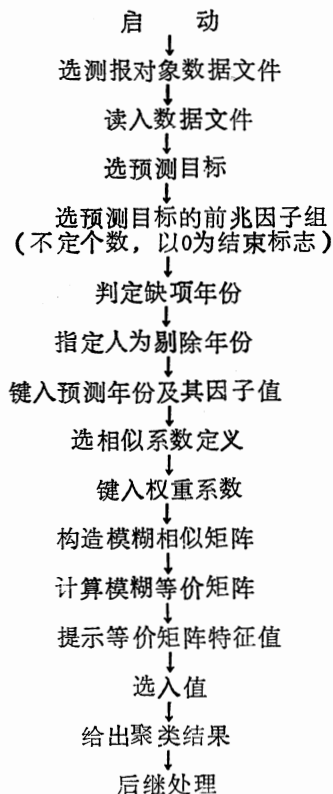
#### 五、程序简介

本程序在 PC/XT 机上用 FORTRAN

—77 语言完成, 源程序 5000 行, 最大执行模块 140K 字节。一个测报对象 (含若干个预测目标) 的有关历史资料组建成 3 个数据文件: 年份、 $Y$ 、 $X$ 。允许最多处理 100 年资料、25 个预测目标和 150 个前兆因子。

程序在中文菜单驱动下执行, 也可以单个功能程序独立执行, 设计了良好的用户界面, 对用户的键盘输入作语法检查和局部语义检查, 对错误报警并自动消除其影响后继续运行。聚类算法运行时间为 10~20 分钟。

程序执行逻辑:



后继处理有 4 种选择: ①重选  $\lambda$  值, 给出另一组聚类结果。②重选相似系数定义, 进行新的模糊相似矩阵计算。③重选预测目标及其前兆因子组, 重新开始计算。④结束。

附表2.A 模糊相似矩阵R

1964	1965	1966	1967	1970	1971	1972	1973	1974	1975	1976	1978	1979	1980	1981	1982	1983	1984	1985	1986
1.000	.542	.500	.468	.530	.639	.697	.798	.664	.438	.651	.560	.469	.658	.208	.274	.218	.074	.103	.130
1965	.542	.631	.491	.418	.553	.539	.533	.415	.628	.620	.583	.755	.605	.086	.348	.300	.027	.168	.258
1966	.500	.631	1.000	.378	.456	.495	.410	.416	.833	.670	.532	.728	.485	.108	.275	.254	.058	.115	.191
1967	.468	.491	.400	1.000	.775	.631	.588	.511	.649	.320	.581	.501	.570	.298	.352	.359	.301	.279	.219
1970	.530	.418	.378	.775	1.000	.532	.532	.742	.276	.625	.588	.478	.606	.380	.380	.399	.393	.310	.243
1971	.639	.456	.456	.631	.754	1.000	.517	.669	.367	.713	.772	.532	.732	.415	.446	.440	.306	.318	.322
1972	.697	.339	.495	.588	.532	.517	1.000	.626	.456	.601	.426	.541	.595	.140	.241	.191	.071	.099	.086
1973	.798	.533	.410	.511	.583	.626	.626	1.000	.367	.355	.611	.456	.700	.308	.402	.335	.177	.233	.256
1974	.664	.415	.416	.649	.742	.669	.678	.647	1.000	.682	.535	.458	.679	.334	.333	.323	.293	.237	.171
1975	.438	.826	.833	.320	.276	.367	.456	.355	.341	1.000	.604	.468	.700	.052	.275	.240	.000	.108	.197
1976	.651	.620	.670	.581	.625	.713	.601	.611	.604	1.000	.736	.681	.770	.368	.449	.440	.275	.310	.321
1978	.560	.583	.532	.501	.588	.772	.426	.623	.536	.468	1.000	.587	.709	.450	.581	.569	.342	.421	.485
1979	.469	.755	.728	.571	.478	.532	.541	.456	.458	.700	.681	1.000	.576	.176	.412	.391	.163	.265	.323
1980	.658	.605	.485	.570	.606	.732	.595	.700	.464	.770	.709	.576	1.000	.385	.503	.469	.264	.358	.356
1981	.208	.086	.108	.298	.443	.415	.140	.308	.394	.052	.368	.450	.385	1.000	.536	.599	.748	.591	.443
1982	.274	.348	.275	.352	.380	.446	.241	.402	.333	.275	.449	.581	.412	.503	1.000	.959	.504	.775	.823
1983	.218	.300	.254	.359	.399	.440	.191	.335	.323	.240	.440	.569	.391	.469	.589	1.000	.597	.841	.800
1984	.074	.027	.058	.301	.393	.306	.071	.177	.293	.000	.275	.342	.163	.264	.748	.504	1.000	.635	.442
1985	.103	.168	.115	.279	.310	.318	.099	.233	.237	.108	.310	.421	.265	.358	.591	.775	.841	1.000	.746
1986	.130	.258	.191	.219	.243	.322	.086	.256	.171	.321	.485	.323	.358	.443	.823	.800	.442	.746	1.000

附表2.B 模糊等价矩阵R\*

1964	1965	1966	1967	1970	1971	1972	1973	1974	1975	1976	1978	1979	1980	1981	1982	1983	1984	1985	1986
1.000	.681	.618	.706	.706	.706	.697	.798	.706	.681	.706	.706	.681	.706	.581	.581	.581	.581	.581	.581
1965	.681	.728	.681	.681	.681	.681	.681	.681	.728	.681	.681	.755	.681	.581	.581	.581	.581	.581	.581
1966	.681	1.000	.681	.681	.681	.681	.681	.681	.833	.681	.681	.728	.681	.581	.581	.581	.581	.581	.581
1967	.706	.681	1.000	.775	.754	.697	.706	.742	.681	.736	.754	.681	.736	.581	.581	.581	.581	.581	.581
1970	.706	.681	.775	1.000	.754	.697	.706	.742	.681	.736	.772	.681	.736	.581	.581	.581	.581	.581	.581
1971	.706	.681	.754	.754	1.000	.697	.706	.742	.681	.697	.697	.681	.697	.581	.581	.581	.581	.581	.581
1972	.697	.681	.697	.697	.697	1.000	.697	.697	.681	.697	.697	.681	.697	.581	.581	.581	.581	.581	.581
1973	.798	.681	.706	.706	.706	.697	1.000	.706	.681	.706	.706	.681	.706	.581	.581	.581	.581	.581	.581
1974	.706	.681	.742	.742	.742	.742	.706	1.000	.681	.736	.742	.681	.736	.581	.581	.581	.581	.581	.581
1975	.681	.728	.681	.681	.681	.681	.681	.681	1.000	.681	.681	.728	.681	.581	.581	.581	.581	.581	.581
1976	.706	.681	.736	.736	.736	.697	.706	.736	.681	1.000	.736	.681	.770	.581	.581	.581	.581	.581	.581
1978	.706	.681	.754	.754	.772	.697	.706	.742	.681	.736	1.000	.681	.736	.581	.581	.581	.581	.581	.581
1979	.581	.728	.681	.681	.681	.681	.681	.681	.728	.681	.681	1.000	.681	.581	.581	.581	.581	.581	.581
1980	.706	.681	.736	.736	.736	.697	.706	.736	.681	.770	.736	.681	1.000	.581	.581	.581	.581	.581	.581
1981	.581	.581	.581	.581	.581	.581	.581	.581	.581	.581	.581	.581	.581	1.000	.635	.635	.748	.635	.635
1982	.581	.581	.581	.581	.581	.581	.581	.581	.581	.581	.581	.581	.581	.635	1.000	.859	.635	.841	.823
1983	.581	.581	.581	.581	.581	.581	.581	.581	.581	.581	.581	.581	.581	.635	.859	1.000	.635	.841	.823
1984	.581	.581	.581	.581	.581	.581	.581	.581	.581	.581	.581	.581	.581	.748	.635	.635	1.000	.635	.635
1985	.581	.581	.581	.581	.581	.581	.581	.581	.581	.581	.581	.581	.581	.635	.841	.841	.635	1.000	.832
1986	.581	.581	.581	.581	.581	.581	.581	.581	.581	.581	.581	.581	.581	.635	.823	.823	.635	.823	1.000

# 51系列单片机多机通信的

## 一种可纠错编码方案

石油大学

张加胜

**摘要** 本文提出了一种在远程数据通信中可实现自动纠错的通信编码, 该码采用 (7, 4) 汉明码的修正方案, 适合于由51系列单片机等构成的分布式多微机测控系统。

### 一、引言

在分布式多微机测控系统中, 由51系列单片机构成其前沿从机子系统是一种较理想的选择。一方面, 51单片机特别适合于智能化测量和实时控制, 另一个很重要的方面是,

其本身具有一个功能较强的异步串行通信口, 尤其适合于多机通信。

该串行口的典型帧格式如图1所示。其中的D<sub>8</sub>位非同一般, 双机通信时可用作奇偶校验, 而多机通信时作为地址字节标识位。具体地讲, 主机在与各前沿机, 即51单片机, 进行通信之前, 先发一个欲要寻址的前沿机

附表3 聚类结果

模糊等价矩阵各特征值

平均值 Rave=.65      标准差 Rdev=.07  
最大元 Rmax=.86      最小元 Rmin=.58

Lambda=.7100

1----	(1964=	5.00),	(1973=	5.00),				
2----	(1965=	5.00),	(1966=	5.00),	(1975=	5.00),	(1979=	5.00),
3----	(1967=	4.00),	(1970=	4.00),	(1971=	5.00),	(1974=	5.00),
	(1976=	5.00),	(1978=	5.00),	(1980=	4.00),		
4----	(1972=	5.00),						
5----	(1981=	4.00),	(1984=	3.00),				
6----	(1982=	3.00),	(1983=	3.00),	(1985=	4.00),	(1986=	-10.00),
1986=C(6)	Yave=	3.33	Ydev=	.58	Ymax=	4.00	Ymin=	3.00

Lambda=.7300

1----	(1964=	5.00),	(1973=	5.00),				
2----	(1965=	5.00),	(1979=	5.00),				
3----	(1966=	5.00),	(1975=	5.00),				
4----	(1967=	4.00),	(1970=	4.00),	(1971=	5.00),	(1974=	5.00),
	(1976=	5.00),	(1978=	5.00),	(1980=	4.00),		
5----	(1972=	5.00),						
6----	(1981=	4.00),	(1984=	3.00),				
7----	(1982=	3.00),	(1983=	3.00),	(1985=	4.00),	(1986=	-10.00),
1986=C(7)	Yave=	3.33	Ydev=	.58	Ymax=	4.00	Ymin=	3.00



地址字节,发此字节时 $D_8$ 位置“1”,而发后续其它信息时 $D_8$ 位清“0”。由于51单片机串行口设有适合于这种多机通信的专门硬件及专门的中断接收方式,主机借助于地址字节标识位,即可以选通所要寻址的前沿机,并实现彼此通信。

可见,51单片机串行口实现多机通信的特殊功能是以牺牲奇偶校验为代价的。然而,对于分布式工业测控系统而言,前沿从机与主机往往相隔较远,更易引入干扰,差错校验一般必不可少,以保证可靠通信。考虑到若采用检错重发的应答方式,应答过程繁琐,降低了数传效率,也增加了通信软件的复杂程度。出于上述两方面考虑,我们提出了采用可纠错编码的通信方案。

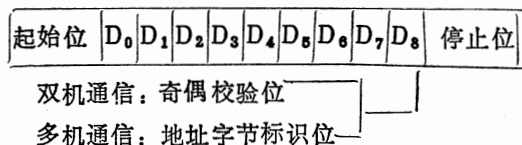


图1 51单片机串行口典型帧格式

## 二、(7, 4) 汉明码及其修正

为实现自动纠错,在传送数据信息时,将每个字节拆成两个四位,分别进行可纠错编码后予以传送。当初我们考虑用(7,4)汉明码的编码方案,其码字生成矩阵为:

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

设 $m = [m_3, m_2, m_1, m_0]$ 为向量表示的待编信息码组, $C = [C_7, C_6, \dots, C_1, C_0]$ 为向量表示的编码后的码字。据公式 $C = mG$ 可将任四位二进制数经编码得到相应的(7,4)汉明码码字,如表1所示。该码最小距离 $d_{m, in} = 3$ ,纠错能力为1,即可纠正任何1位随机错误。

从前面对单片机串行口的帧格式的分析看到,主机要对多个前沿机进行寻址通信时,须先发一个欲要寻址的从机址码,并且相应

地将对应的 $D_8$ 位置“1”当下发其它信息字节时须将该位清“0”这就对主机的串行口提出了一定要求,即主机串行口的帧格式应与前沿机的相匹配。实现这一匹配并不是普通的串行口所能做到的,实际上除51单片机外,大多数可编程串行接口芯片都难以胜任。为解决这一匹配问题,我们采用了(7,4)汉明码的修正方案,即将(7,4)汉明码的每个码字再增加一个附加元,对应于附加元的两种取值可以得到两组码字,如表2所示,一组我们称作(8,4)汉明偶码,另一组称作(8,4)汉明奇码。应用中我们将主机的串行口经编程设置为奇校验,主机欲要对从机寻址时,先发一个偶码作为选通地址码,而在下发后续其它信息时均一律采用奇码。由于采用奇校验,保证了主机在发选通地址码时,奇偶校验位(即 $D_8$ 位)自动置“1”,而发其它信息时恒为“0”。

采用这种修正码方案,一方面实现了主从机间通信格式的匹配,另一方面,经修正后的奇偶码,在纠错能力上也比原来有相当程度的提高。

## 三、奇偶码的纠错译码问题

经修正之后的奇偶两种码,均已不属于正规的线性分组码,有关线性分组码的纠错能力的分析方法已不再适用,但最大似然译码法总是适合的。从这种纠错译码的原理出发,为弄清奇偶码纠错能力怎样,笔者对每一个码字,当8位码元中任发生一位和两位错误的情形进行了分析和研究。在此仅以奇码为例加以简要说明。

(8,4)奇码共有16个奇码字,我们以 $C_7, C_6, \dots, C_1, C_0$ 来表示其每个码字的8位码元,分析结果表明,对每一个码字,在传送过程中若随机地发生a)任何一位错;b) $C_7, C_6, \dots, C_1, C_0$ 这8位中的相邻两位错( $C_6, C_5$ 相邻的情况除外);或c) $C_3, C_0$ 两位错,

均可采用最大似然译码原则并配合以软件查表技术予以纠正。比如从表2中看到, 2H的奇编码为2CH, 此奇码字发生上述三种错误的情况如表3所示。(表2、表3见下页)

正象表3所示的那样, 每个码字对应于上述三种错误, 都错成有15个彼此异样的码值, 我们之所以说全部奇码对上述三种错误能实现纠错译码, 原因就在于对所有16个奇码字来说, 当发生这些错误时所错成的全部  $15 \times 16 = 240$  个码值中, 无任何重叠的码值, 加上16个奇码字本身, 共256个码值, 正好对应于8位二进制数的256种不同组合。将其中每一种组合都视作某个奇码字由于传输出错而错成的码值(16个奇码字除外), 从而可以唯一地将其进行纠错译码。比如表3中所列15个码值, 均由码字2CH错成, 且与2CH距离最小, 故它们连同码字2CH本身均依照最大似然译码原则被译作2H。对于其它15个码字亦如法炮制。至于偶码, 以相同的分析方法并可以得到同样的结果, 不再赘述。

#### 四、编码和译码的软件实现问题

奇偶码的编码和纠错译码, 仅通过软件即可实现, 无需专设任何硬件。可以利用伪指令“DB”来构造奇、偶编码表和译码表, 共四张。比如构造奇编码表只需将16个奇码字顺序排列在以某标号为首址的16个内存单元中即可。以编码表首址为基址, 以待编信息码为偏移量, 通过查表方式即可检索出相应的编码。再如奇译码表的构造方法是, 将00H~FFH这256个码值各自对应的纠错译码值, 顺序排列在以某标号为首址的内存单元中。译码采用与编码过程相同的查表方式。

这种编、译码方法, 不仅汇编语言十分简单和节省硬件, 而且速度快, 尤其是51单片机的汇编语言, 特别便于查表检索。一般在各前沿机上需配置三张表, 即奇编码、奇译码和偶译码表, 而在其上位机需配置奇、

偶编码和奇译码三张表。好在这些码表并占用不了多少内存单元。

#### 五、结语

本文所介绍的可纠错编码修正方案已在一个分布式多微机系统中得以试用, 调试简便, 效果满意。从纠错能力上讲, 能对全部一位错及近乎90%的相邻两位错均可予以自动纠正, 这比奇偶校验方式有明显的优越性。

表1 (7,4)汉明码

信息码组 (H)	编码后的码字		码 重
	信息元	监督元	
0	0000	000	0
1	0001	011	3
2	0010	110	3
3	0011	101	4
4	0100	111	4
5	0101	100	3
6	0110	001	3
7	0111	010	4
8	1000	101	3
9	1001	110	4
A	1010	011	4
B	1011	000	3
C	1100	010	3
D	1101	001	4
E	1110	100	4
F	1111	111	7

#### 参考文献

- 1 张鸣瑞, 编码原理, 北京航空航天大学201教研室, 1987.
- 2 孙涵芳、徐爱卿, MCS-51、96系列单片机原理及应用, 北航出版社, 1988.
- 3 T.豪斯莱·数据通信和远程处理系统, 人民邮电出版社, 1985.

# 常见微机第四代工具性能的试验分析

天津大学计算中心 赵国瑞

**摘要** 微机上的大多数DBMS都被认为是第四代语言(4GL)系统。本文介绍了一种基准测试程序,并给出了对常见微机4GL系统测试的结果。

第四代语言(4GL)已经成了各种计算机上解决问题的重要工具。4GL系统使得非专门人员也很容易描述、检索、维护和显示数据;利用4GL系统提供的良好用户界面,从菜单上就可以选择要执行的命令或功能;使用4GL,可以显著的提高软件的生产率,缩短软件的设计周期。

当前微机上大多数 DBMS 都被认为是 4GL 系统。它们有的是从中小型机上移植过

来的,如FOCUS,ORACLE,INFORMIX;有的是专为微机开发的,如dBASE,RBASE,CONDOR, PARADOX。本文的目的,就是要研究这些系统在一组简单但普遍使用的数据处理任务上的相对性能,探讨4GL当前版本的改进方向。

## 一、试验步骤

表2 (7, 4) 汉明码的修正

原信息码(H)	(7,4) 汉明码		奇码附加元C0	奇码字(H)	偶码附加元C0	偶码字(H)
	信息元C7~C4	监督元C3~C1				
0	0000	000	1	01	0	00
1	0001	011	0	16	1	17
2	0010	110	0	2C	1	2D
3	0011	101	1	3B	0	3A
4	0100	111	1	4F	0	4E
5	0101	100	0	58	1	59
6	0110	001	0	62	1	63
7	0111	010	1	73	0	74
8	1000	101	0	80	1	8B
9	1001	110	1	90	0	9C
A	1010	011	1	R7	0	A6
B	1011	000	0	B0	1	D4
C	1100	010	0	C4	1	C5
D	1101	001	1	D3	0	B2
E	1110	100	1	E9	0	E8
F	1111	111	0	FE	1	FF

表3 奇码字2CH发生各种错误后的情况

原 信 息 码		0010
奇 编 码		00 10 1100
发生一位 错误后错 成的码值 (H)	C7错	AC
	C6错	6C
	C5错	0C
	C4错	3C
	C3错	24
	C2错	28
	C1错	2E
	C0错	2D
相邻两位 出错后错 成的码值 (H)	C7 C6错	EC
	C5 C4错	1C
	C4 C3错	34
	C3 C2错	20
	C2 C1错	2A
	C1 C0错	2F
非相邻两位出错后 错成的码值(H)		C3 C0错 25

试验由以下步骤构成：  
建立试验用样本数据库；  
书写测试程序；  
运行测试程序；  
分析运行结果。

下面对前三个步骤分别加以说明。

### 1. 建立试验用样本数据库

样本数据库由文件（或称为表、关系）SUPPLIER、PART和STOCK组成（详见表1）。数据库中的记录是随机生成的。记录的个数从1百到几千（详见表2）。

表1 样本数据库的结构

SUPPLIER(供应者)		PART(零件)		STOCK(库存)	
SUP #	char(5)	PART #	char(5)	SUP #	char(5)
SUPNAME	char(15)	PARTNAME	char(15)	PART #	char(5)
SUPCITY	char(15)	PARTWEIGHT	integer	QUANTITY	integer
SUPSTATUS	integer	PARTCOLOR	char(6)		
		PARTCITY	char(15)		

表2 各文件的样本大小

样本号	SUPPLIER 的记录数	PART的 记录数	STOCK 的记录数
1	100	100	236
2	200	200	486
3	300	300	725
4	500	500	1208
5	1000	1000	2417
6	2000	2000	4848
7	5000	5000	12154

按照 E.F.Codd 提出的关系完整性概念，能够模拟表的投影、选择、连接、并、差这五种操作的任何一种数据库语言都具有相同的代数表达能力（这里用更加通用的连接操作代替了笛卡儿积操作）。这些操作是构造关系查询的最小集合。并且在同一类关系中只包含唯一的元组而不允许有完全相同的记录。

被试验的4GL都支持投影和选择操作，甚至它们所用的句法结构也很类似。例如，

含有1万和2万个零件记录的大文件也曾试过，但由于某些4GL系统执行某些查询的时间超过一个工作日，故未把它们包含在运行结果中。

### 2. 书写测试程序

用于测试的程序有4类：（1）关系查询，（2）文件操作，（3）记录检索，（4）数据维护（详见表3）。

下面对各类操作给出简要说明。

#### （1）关系查询

查询Q2可以用SQL语言（ORACLE、INFORMIX）表示为：

```
SELECT Part#
FROM PART
WHERE PartWeight <= 700
```

而用dBASE III则可表示为：

```
USE PART
LIST Part* FOR PartWeight <= 700
```

但是，并非所有被试验的4GL系统都支持“REMOVE DUPLICATES RECORDS”（清除重复记录）的功能。对于这些系统（dBASE、CONDOR）我们应用了如下消重方法：

a) 将结果分类；b) 顺序地查看每个记录，如果该记录等于它的下一个记录，则对它加删除标记；c) 将不带删除标记的记录拷贝到一个新结果文件中。

对于并操作和差操作，被试验的系统都未直接提供支持。对于提供SQL语言的系统，用下面的语句实现SUPPLIER关系本身的并操作：

```

INSERT INTO temporal (Sup#, Supstatus, Supcity, Supname)
SELECT * FROM SUPPLIER;
INSERT INTO temporal (Sup#, Supstatus, Supcity, Supname)
SELECT * FROM SUPPLIER;
INSERT INTO result (Sup#, Supstatus, Supcity, Supname)
SELECT DISLINCT * FROM temporal;

```

表3 测试程序表

类型	操 作
关 系 查 询	Q1. 投影 从SUPPLIER中取出所有不同的STATUS和CITY的对偶
	Q2. 选择 从PART中选出不超过700磅重的兰色零件信息
	Q3. 并 合并SUPPLIER本身, 并在结果中不保留重复的记录
	Q4. 差 从PART中消去包含在另一个PART文件中的记录
	Q5. 连接 组合STOCK与PART中有相同PART#的记录
文件 操作	Q6 分类 按PART#的升序将PART中的记录分类
	Q7. 建立索引 对PART文件在PART#上建立索引
记 录 检 索	Q8A. 顺序扫描 计算PART文件中全体零件的重量之和
	Q8B. 索引顺序扫描 利用Q7在PART#上建立的索引实现前一功能
	Q9. 使用辅索引选择 利用予先在PARTCOLOR上建立的索引, 从PART文件中选出不超过700 磅重的兰色零件信息 (与Q2类似)
数 据 维 护	Q10. 修改 把与TEMPPART文件中的PART#相同的PART文件中的全体记录的PARTCOLOR 值改为“RED”
	Q11. 删除 从PART文件中删掉与TEMPPART文件中的PART#有相同值的全体记录
	Q12. 插入 把TEMPPART文件中的全体记录插入到PART文件中
	(注: TEMPPART文件中存放着从PART文件中随机选出的10%记录)

而对其它系统是用APPEND命令合并, 然后再用前面说明的消重方法得到并操作结果的。

对于差操作, 基于SQL的系统 (ORACLE、INFORMIX) 是用下面的SQL语句实现查询Q4的:

```

INSERT INTO temporal (Part#,
PartWeight, Partcolor, PartDesc)
SELECT * FROM PART;
SELECT *

```

FROM PART

```

WHERE Part# NOT IN (SELECT
Part# FROM temporal);

```

而对其它系统是用如下方法: 首先用APPEND命令将PART 文件中的全体记录填到空文件temporal中, 然后对PART文件中的Part#不在 temporal 中的记录加标记, 最后把PART中加标记的全体记录拷贝到结果文件中。

对于连接操作, 被试验的4GL系统都支



持它,虽然有些仅支持等值连接。

## (2) 文件操作

分类和检索是两个普通的文件操作。对于纯关系数据库系统来说,由于记录查询是按照关键字而不是相对位置进行的,因此分类操作似乎不太重要。但是为了给其它系统或报告书写程序产生输出文件,分类却是很有用的。除了ORACLE之外,其它系统都允许按属性(组)的升降序分类。

建立索引可明显地加速检索操作。在关系系统中,建立和维护索引是一个关键因素。被试验的4GL系统都支持建立主关键字(唯一标识)索引的操作,而且,除了PC-FOCUS之外,它们还允许在非主关键字上建立辅助索引。

## (3) 记录检索

Q8A用于检测顺序扫描,Q8B用于检测索引顺序扫描,Q9用于检测使用辅助索引的检索。

## (4) 数据维护

关系数据库模型并未明确定义维护操作。显然,修改已有的数据是在数据库文件上的主要方面。这里的数据维护,包括修改、删除、插入三种常规操作。

## 3. 运行测试程序

试验是在IBM PS/2 80型微机上进行的。主要技术指标为:主频 16 M,主存2M字节(实际仅用 640K),70 兆硬盘,平均查寻时间为29毫秒,操作系统 DOS 3.3。

每个操作都写入一个独立的批处理文件。执行时记录运行开始时刻和执行结束时刻的时间。执行时间中含有回答每个操作并将结果写入硬盘文件所消耗的时间。

## 二、运行结果及分析

对于各种尺寸的样本数据库的每种操作所花费的时间如表4所示。表4中(a)表示PC-FOCUS不直接支持辅助索引;(b)表示ORACLE在对5000个PART记录和12154

个STOCK记录进行连接时,花费的时间超过24小时即86400秒;(c)表示 ORACLE不直接支持文件的分类操作。

从表4可以看出,各4GL 系统之间对同一操作所花费的时间是很不相同的。表5 给出了按小计数量执行最快系统与执行最慢系统所花费时间的比率。从表5可以看到,Q8A操作相差最小,比率为1: 3.3,而Q3和Q5操作相差最大,竟为1000多倍。

从表4和表5可以看出,在数据库中的记录数增加情况下,PARADOX和CONDOR的性能依然是稳定的,它们完成全部操作所花费的时间最少;而有的系统则不然,随着记录个数增加,性能急剧下降,如 RBASE、dBASE和ORACLE 对连接操作来说,最小尺寸的数据库(100个记录)与最大尺寸的数据库(5000个记录)情况下所花费的时间(按单个记录计算)竟相差40多倍。

应该指出,虽然各4GL系统之间的性能相差较大,但是,没有任何系统对所有操作都超过其它系统。不同的系统适于某种应用的环境。例如,PARADOX和CONDOR运行很快,但CONDOR缺少一个可编程的环境,因而不能把关系操作嵌套在传统的高级程序语言中,PARADOX 则不能在小型机以上的系统中使用。ORACLE虽然运行较慢,特别是对大尺寸数据文件的连接操作,但它不仅可以在 PC 机上运行,而且也可以在中小型机上运行。INFORMIX 运行虽然不太慢,也能在PC 机和中小型机上运行,但它只给用户提供一个很复杂的与 C 语言的接口。RBASE和dBASE 是运行较慢的系统,而且仅能在PC机上使用,但是容易学习掌握。

## 三、结论

由13个简单操作组成的 4GL 系统性能测试程序可以比较容易地实现,并且可以测试出被测各4GL系统在不同操作下的相对性能,从而给出了某个4GL工具对于某类应用的

适应性信息。但是,测试程序不是对某个4GL产品的评价程序。对4GL产品的评价需要从更大的范围去考虑。

从上面给出的测试结果可以看出,目前使用的PC机上的4GL系统都还需要在某个或某些方面加以改进或改善,以适应用户的应用要求。

说明:上面4GL系统的各版本如下:

1. dBASE Plus, 1985
2. RBASE System V1.1, 1986
3. PC-FOCUS 2.0
4. PARADOX 1.1, 1986
5. CONDOR 3, 1986
6. ORACLE 4.0, 1984
7. INFORMIX-4GL 1.00, 1986

表4 各样本数据库下查询运行时间表

单位:秒

样    本	操        作													
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8A	Q8B	Q9	Q10	Q11	Q12	
dBASE	1	4	2	21	18	17	2	3	2	4	3	2	5	15
	2	7	3	72	13	60	3	3	2	8	3	2	14	30
	3	10	2	337	18	292	3	4	2	12	6	3	24	49
	4	14	3	882	31	698	4	5	2	25	9	5	46	91
	5	28	5	3348	58	2924	10	8	4	51	17	8	105	186
	6	53	7	13003	116	11515	19	16	7	118	30	15	238	408
	7	133	14	80050	298	71558	48	45	15	359	79	36	695	1149
	小计	249	36	97713	542	87064	89	84	34	577	147	71	1091	1928
RBASE	1	8	3	103	30	25	4	5	2	10	2	3	3	8
	2	15	3	249	107	93	9	5	3	32	2	4	3	10
	3	30	3	414	233	209	17	6	3	58	2	4	3	11
	4	40	4	879	612	542	30	7	4	144	2	4	3	15
	5	82	4	2850	2397	2155	66	11	4	540	2	6	3	17
	6	210	5	10145	9378	8676	157	18	6	2072	2	10	3	31
	7	626	9	58932	56991	54119	533	39	12	12679	2	21	3	63
	小计	1011	31	73572	69748	65819	816	91	34	15535	14	52	21	155
FOCUS	1	11	10	142	99	14	11	9	8	9	(a)	14	9	9
	2	12	10	277	190	28	12	11	8	9		14	9	9
	3	12	11	407	281	40	12	12	9	9		15	9	8
	4	13	10	71	43	66	24	16	8	9		13	8	8
	5	27	10	142	79	125	37	26	8	30		15	8	8
	6	37	12	656	153	337	59	47	10	74		26	8	9
	7	71	15	1733	370	1289	236	113	12	234		61	9	9
	小计	183	78	3428	1215	1899	391	234	63	374		158	59	59
PARADOX	1	2	1	4	2	2	2	1	1	2	1	4	2	4
	2	2	1	5	4	2	2	2	2	1	1	4	4	3
	3	2	1	6	4	3	2	2	2	2	1	4	4	3
	4	3	2	11	6	4	3	2	2	2	1	5	5	4
	5	6	2	27	16	9	3	4	4	3	1	7	7	6
	6	11	3	59	53	15	8	8	8	11	5	22	22	13
	7	29	8	169	182	73	19	32	22	27	12	38	38	15
	小计	55	18	281	267	108	39	51	41	48	22	84	82	48

CONDOR	1	12	4	5	15	6	2	2	1	0	4	2	4	1
	2	13	6	5	18	14	3	2	1	1	4	4	5	2
	3	14	6	6	22	16	4	3	2	1	4	5	7	2
	4	16	8	8	31	31	6	4	2	2	4	7	10	2
	5	22	20	11	56	68	11	12	2	2	4	14	25	2
	6	41	28	18	123	163	24	54	4	4	5	33	71	3
	7	127	117	44	240	641	92	212	7	7	8	117	236	6
小结		245	189	97	504	939	142	289	19	17	33	164	358	18
ORACLE	1	6	1	4	12	55	(c)	3	1	2	1	2	2	1
	2	6	2	70	20	205		6	2	3	3	8	7	6
	3	6	3	51	63	505		9	2	3	3	15	15	6
	4	6	5	420	65	1342		13	3	4	5	28	29	12
	5	7	5	1483	120	5055		14	4	4	7	41	43	15
	6	14	34	7035	783	21856		96	7	8	50	440	321	174
	7	26	35	36993	545	(b)		112	17	17	57	585	456	197
小计		71	85	46056	1608	119018		253	36	41	126	1119	873	411
INFORMIX	1	4	2	3	11	74	9	2	1	2	2	3	5	3
	2	5	2	4	18	269	37	5	1	2	3	5	10	5
	3	6	3	6	25	560	63	7	1	3	5	8	14	6
	4	6	4	9	41	1467	109	23	2	5	9	15	23	8
	5	11	7	15	89	5656	251	49	4	8	18	41	57	31
	6	19	14	30	169	21790	471	105	7	16	37	91	139	69
	7	43	36	78	820	48044	1289	324	17	42	112	333	570	183
小计		92	68	145	1173	77860	2229	515	33	78	186	486	818	305

表5 最快系统与最慢系统及时间开销比率

操作	最快的系统	最慢的系统	最快系统时间开销与最慢系统时间开销之比
Q1	PARADOX	RBASE	1:18.4
Q2	PARADOX	CONDOR	1:10.5
Q3	CONDOR	dBASE	1:1007.4
Q4	PARADOX	RBASE	1:261.2
Q5	PARADOX	ORACLE	1:1102.0
Q6	PARADOX	INFORMIX	1:57.2
Q7	PARADOX	INFORMIX	1:10.1
Q8A	CONDOR	FOCUS	1:3.3
Q8B	CONDOR	RBASE	1:913.8
Q9	RBASE	INFORMIX	1:13.3
Q10	RBASE	ORACLE	1:21.5
Q11	RBASE	dBASE	1:52
Q12	CONDOR	dBASE	1:107

# 如何编制高质量的驻留内存程序

武汉第二炮兵指挥学院 张卫兵

**摘要** 本文通过具体的实例详细地介绍了实现程序驻留内存的原理和方法,以及编制驻留程序的各种技巧。并且深入地讨论了激活驻留程序的方法和各种可能出现的问题。

## 一、概述

在PC-DOS下大多数程序都要经历这样一个过程:被调入存储器执行、结束并返回到COMMAND.COM。象PASCAL、BASIC这样的高级语言程序通常只有这一种过程方式,但汇编语言却不仅如此。在汇编语言程序结束时,如果调用INT 27H中断,当返回COMMAND.COM后,程序的全部或局部仍然留在内存中,并且被保留的部分不会被后续运行的程序覆盖,除非再次装入DOS系统。这样就使我们对操作系统或ROM BIOS的功能进行适当的修改和加强成为可能。例如装入我们自己编制的可以恒起作用的中断处理程序。

## 二、驻留内存的方法

DOS为程序结束处理提供了多种方法,其中最常用的是调用中断INT 20H、INT 27H及系统功能调用INT 21H子功能00H、31H或4CH。使用DOS专用中断INT 22H、INT 23H作为程序结束同样是可行的。

当用上述之一完成DOS的结束操作后均转到INT 22H,该中断将实施返回DOS的最后操作:

包括: a 程序返回到COMMAND.COM的常驻区,出现DOS提示符C>。

b 若COMMAND.COM的暂驻区被覆盖,则执行暂驻区的装入程序,将暂驻部分装入到内存高端,再返回常驻区,出现提示符C>。

c 若程序返回到批文件处理程序,则接着解释批处理文件的下一行批命令并执行

之,直到处理完毕,显示提示符C>。

其中调用INT 20H及INT 21H的子功能00H、4CH退出时均通知DOS结束并释放所占的内存,以供下次再装入其它程序时使用,无法使程序保留在内存中不受覆盖。但使用INT 27H和INT 21H的子功能31H退出后,结果就完全不同了。

### 1. 在程序结束时调用INT 27H退出

INT 27H被调用后,终止当前程序的执行,但保留部分或全部的程序于内存,并不被覆盖,DOS随后做如下动作:

a 从程序PSP:000AH恢复INT 22H的向量;

b 从程序PSP:000EH恢复INT 23H Ctr1—C的向量;

c 从程序PSP:0012H恢复INT 24H严重错误处理向量;

d 转到结束地址。

此后,程序返回到COMMAND.COM的驻留部分,如果正在执行的是批处理文件则取下一行的批命令并解释执行,否则显示C>。

调用方式

入口参数:

DX=要保护程序的最后一字节加1的位移,其中包括程序段前缀PSP

CX=程序段前缀PSP段 \*

出口参数:无

INT 27H一般允许用于写驱动程序或中断处理的COM或EXE程序,使它们驻留内存,但INT 27H最多只能保留内存64k字节。如果待驻留的程序超过64k则无法用INT 27H实现驻留,而只能用INT 21H的

子功能31H。后者允许我们保留任意大小的内存。此外，用INT 27H 驻留程序不能以/HIGH开链接，即该程序不得驻留在内存的高端，因为这部分由COMMAND.COM的暂存部分使用，否则COMMAND.COM不能装入内存，系统将失败。

下面给出了COM和EXE两种文件的例子。

COM文件例：

BEGIN:JMP SET-UP

ZD PROC FAR

PUSH DS; 保存寄存器

:

中断例程

:

POP DS, 恢复寄存器

IRET , 中断返回

FEND EQU \$; 标示例程结束

ZD ENDP

; 建立中断向量

SET-UP:

MOV DX, OFFSET ZD; 例程偏置值→  
DX

MOV AL, 70H

MOV AH, 25H

INT 21H

LEA DX, FEND; 设置常驻程序的偏值

INT 27H; 退出例程驻留

EXE文件例：

对 EXE文件偏置是从 PSP之后第一个字节算起，故需加上100H 才能得到 PSP底部开始算起的偏置值。

BEGIN:JMP SET-UP

ZD PROC FAR

PUSH DS

:

中断例程

:

POP DS

IRET

FEND EQU \$; 标示例程结束

ZD ENDP

; 建立中断向量

SET-UP:

MOV DX, OFFSET ZD; 例程偏置值→  
DX

MOV AX, SEG ZD

MOV DS, AX; 例程段值→DS

MOV AL, 70H

MOV AH, 25H

INT 21H

MOV DX, FEND+100H; 设置常驻例程末尾  
的偏值

MOV BYTE PTR ES:1, 27H; 将INT27H  
写入PSP

RET ; 退出时执行INT27H驻留例程

2. 在程序结束时调用 INT21H的子功能31H退出。

该功能被调用后DOS所做的动作和调用INT 27H退出后相同，但结果却不甚相同。INT21H的子功能31H，能普遍地用于驻留驱动程序或中断子程序库，调用时只要将驻留程序的长度送入DX即可。驻留程序长度不受限制，且允许传送一返回码，故优于INT 27H的驻留。

调用方式

AH=31H

AL=返回码

DX=保留的内存大小(字节数)

INT 21H

例子、带返回码1结束，但驻留内存，保留从程序段前缀开始的16k内存

MOV AH, 31H

MOV AL, 1

MOV DX, 0400H

INT 21H

### 三、驻留程序再获控制权的方法及问题

驻留的程序不同于传统的普通程序，必须考虑它再获控制权的问题，否则它只是白白地占用内存而毫无用处。

在 IBMPC/XT-AT的 RAM中从绝对地址0开始的1k字节里，存放着一张中断向量



表, 我们能够通过这个公用的向量表来替换中断子程序, 由驻留程序来监测硬件事件和软件服务请求。当发出这些硬中断或软中断调用时, 即可进入驻留的对应中断处理程序, 使得驻留程序获得控制权。

另一种方式是采用“热键”启动的方法, 用户按下指定的组合键立即中断当前运行的应用程序而激活驻留的程序, 从而再获控制权。

### 1. 中断方式

中断方式激活驻留程序分两个情况。

一是利用某些特殊的硬件中断, 这些中断可自动地被计算机所调用, 例如: BIOS 包含一特殊的伪中断 (INT1CH), 这一中断在给它提供子程序之前不起任何作用。启动时中断矢量指向 IRET 指令。调用中断程序只是简单地返回。INT1CH 的特别之处在于它是在 BIOS 定时器中断修正日历计数之后由该中断调用的, 也就是说这是每秒 18.2 次自动发生的硬件中断。我们可以改变它的中断矢量使之指向自己程序。这样, 这个程序就会被每秒钟调用 18.2 次。

另一种情况是替代软中断。修改一些未被 DOS 使用的软中断向量, 使之指向我们的驻留程序, 当用户程序中执行一条对应的 INT 命令时, 就可以激活对应的驻留程序。

下面的例子说明如何使中断向量指向自己的驻留程序。INT 21H 的子功能 25H 将中断向量设置到指定的地址。该地址长为两个字, 高位字为段值 (CS), 低位字为偏置值 (IP)。只需按下例的顺序将例程的段值存入 DS, 偏置值存入 DX, 就可使中断向量指向你的例程。注意任何中断例程必须以 IRET 结尾, 而不用通常的 RET 指令。因为 IRET 将三个字托出堆栈包括标志寄存器, 而 RET 指令只推出两个字。如果用普通例程 RET 结尾, 机器内部系统堆栈就会混乱而导致系统死锁。另外, 对任何硬件中断都应在末尾加上这两行程序:

```
MOV AL, 20H
```

```
OUT 20H, AL
```

若硬件中断不以这两行结尾, 则管理硬件中断的 8259 芯片就不会清其当前服务寄存器, 这样就会导致中断在较低优先级上重复调用。最好在修改硬件中断向量时关中断修改完后再打开。

例:

; 建立中断

CLI ; 关中断

PUSH DS; 存 DS

MOV DX, OFFSET ZD; 中断例程偏置  
值→DX

MOV DS, SEG ZD; 中断例程的段→DS 中

MOV AH, 25H

MOV AL, 1CH 或 60H; 向量号 (INT1CH 或  
INT60H)

INT 21H

POP DS

STI ; 开中断

; 中断例程

ZD PROC FAR

PUSH AX; 保存所有已修改的寄存器或  
数据区

:

POP AX; 恢复所有已修改的寄存器和  
数据区

MOV AL, 20H; 只有这两行与硬件中断有关

OUT 20H, AL

IRET

ZD ENDP

这样, 驻留的 ZD 过程就每秒被调用 18.2 次, 或者当其它用户程序调用 INT60H 时, 即可激活 ZD 过程。

### 2. 热键

实现热键激活驻留程序有两个途径。

一是修改已有的 ROM BIOS 中的 INT 16H 中断键盘驱动程序。在正常的 INT 16H 中断程序前, 由我们的常驻程序接过中断 INT 16H 的控制权, 以检测是否按下“热键”。若是, 则激活驻留程序; 否则转入正常的 INT 16H 中断程序。

二是直接修改硬中断INT 9H。常驻程序可以使用正常的ROM BIOS INT9H所不承认的组合键或死键来作为热键。应该注意的是INT 9H是个硬件中断,因此在编写INT 9H的处理程序时,应保证在每次中断后应重新允许8259中断控制器工作。

无论使用那一种方法实现热键激活驻留程序,都应该考虑避免热键的冲突问题。一是两个或多个驻留程序选用了同一个热键;二是可能某个前台应用程序所用的控制键正是后台某个驻留程序的热键。解决的办法是各驻留程序要求使用不同的热键,并且尽量选择一个很少用的默认组合键,以避免应用程序的控制键。

## 四、显示屏幕的保存及恢复

驻留程序的激活是个随机事件。在任意时刻一旦驻留程序激活,则前台应用程序被中断。因此驻留程序退出后必须保证恢复被中断的前台应用程序并使之正确地继续执行。为此在中断当前程序而驻留程序被激活时还必须保存中断前应用程序的机器状态。该状态包括三个方面:寄存器、视频显示窗口以及DOS参数。

### 1. 保存寄存器

保存驻留程序所用的全部寄存器是显而易见的。广义地讲必须强制性地全部的寄存器保存到自己的数据区中并在退出时恢复。唯一例外的情况只有指令寄存器、码段寄存器CS以及标志寄存器,因为这些寄存器已由机器指令RET和IRET自动保存并恢复。

### 2. 保存视频显示窗口

驻留程序被激活后必定向用户推出新的显示窗口,以便显示它的提示信息和功能菜单。当结束驻留程序返回到用户程序时就必

须恢复中断前用户程序的屏幕显示状态。这也包含三个方面:视频显示方式、光标参数(大小位置)、屏幕数据。

#### (1) 保存视频方式及显示页

当前视频方式及显示页可由INT10H功能15H得到。在驻留程序终止前,可由INT10H功能0H来恢复视频方式,由功能5H恢复显页。

#### (2) 保存光标参数

使用INT 10H功能3H可获得当前指定页的光标的类型和位置。功能1H用于设置光标类型,功能2H用于设置光标位置。

#### (3) 保存屏幕数据

如果被中断程序的显示方式是在文本方式,而且驻留程序显示的也是一些文字信息,那么保存和恢复屏幕数据就没有什么问题。有两种方法可供选择:

第一种方法是使用INT 10H的子功能8H读当前光标处的字符及属性,只要把屏上每个位置读一遍并保存即可。恢复的方法则是调用INT 10H的子功能9H,将保存在暂存区中的显示屏数据再写一遍即可。不过这种方法速度较慢。

第二种方法是直接采取保存显示缓冲区,对显示缓冲区直接读写。此方法优点是速度较快。

如果原视频显示方式是图形方式,那么情况就比较复杂了。简单地使用文本方式的保存方法来保存和恢复屏幕数据显然是无法实现的。因为文本方式使用的视频内存区仅由4000个字节组成,而对于CGA卡图形方式的显示缓冲区内存就有32kB,EGA则有多达256kB的内存显示缓冲区。

我们提供两种方法实现:一是牺牲内存保存当前显示屏的全部显示缓冲区,然后调用INT10H的子功能0H切换显示方式。退出时先用INT10H的子功能0H切换回原图形方式再恢复图形。这种方法较容易实现,但需要以牺牲宝贵的内存为代价,这对于驻留程

序来说不是个理想的方法。

第二种方法是仅保存实际被驻留程序冲掉的显示缓冲区,这样需要驻留程序直接通过对显示控制器6845芯片编程实现显示方式的切换,而不能调用INT 10H。因为使用INT 10H的子功能0H切换显示方式时,将清除掉显示缓冲区的全部数据。而直接对6845芯片编程只需要保存常驻程序所使用的4k字节。这种方法解决了第一种方法的缺陷,即驻留程序占用的内存减少,但编程实现较为复杂,而且硬件兼容性较差。

### 3. DOS参数的保护

当DOS装入一程序时,它将维护两个变量:一个是程序段前缀PSP段地址,另一个

是磁盘传送区DAT的段偏量地址。

PSP用于维护与当前过程有关的信息。当驻留程序中断前台程序时,DOS将继续使用属于前台程序的PSP,而不是驻留程序的PSP。只要驻留程序尽可能少打开文件并且在结束时关闭所有打开的文件,则无需为驻留程序重新开辟新的PSP。

磁盘传送区DAT是DOS用于为FCB功能传送文件数据的内存缓冲区。如果驻留程序中调用了FCB功能,则DAT中的数据将被改变。为了能使被中断的前台程序正确地继续执行,驻留程序必须保存DAT的值,并在退出前恢复。我们可以很容易地通过DOS功能调用2FH和1AH来获得并设置DAT。

## 征稿启事

一、《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。重点介绍国内外微、小型计算机总体结构,系统配置,软、硬件和网络设计及应用方面的专题,辟有“应用实例”、“计算机网络”、“典型程序介绍”、“经验点滴”等栏目,选登国内外优秀的应用程序,介绍软、硬件设计心得和创见。向读者提供国内外计算机发展动态和最新的计算机科技知识,适合计算机界广大科技人员、管理人员、教学人员,大专院校师生及各行各业计算机用户阅读。欢迎广大读者踊跃投稿。

### 二、征稿内容

1. 学术论文; 2. 科研成果; 3. 应用实例; 4. 经验体会与创见; 5. 国内外动态介绍与分析。

### 三、来稿须知

1. 来稿请一律用钢笔写在单面稿纸上,(打印稿请注明页数、字数)文章力求主题突出、内容充实、数据可靠、字迹工整、一般不要超过6千字。

2. 文章前面必须要有内容摘要,并附英文标题及姓名,如有程序清单必须用计算机打印,扞图按制图要求画清,在文中所占位置用方框标出并注明图号及图注。

3. 每一标点符号占一格,文中需用黑体字时请在字下加波纹线。

4. 参考文献写于文章最后,一般为书名、作者、页数、出版社与出版年月。

5. 来稿请写真实姓名(发表可用笔名),工作单位、职务、详细通讯地址、邮政编码。

四、来稿如经采用,本编辑部在收到稿件后3~6个月内通知作者。作者请自留底稿,不采用稿件不予退回。稿件发表后,按规定付稿酬。

五、来稿请寄:天津市河西区友谊路宾馆南道五号、天津市电子计算机研究所《微小型计算机开发与应用》编辑部。邮政编码:300061

# 正交试验设计的计算机实现方法

浙江慈溪市公安局电脑室 吴晓禾

浙江台州师专数学系 赵俊

**摘要** 本文介绍一种正交试验设计从正交表的自动构造、试验方案的安排到最后得出分析结果这一全过程在微机上实现的方法。本方法用函数和矩阵运算来构造正交表,用回溯策略来模拟确定试验方案时的人工处理过程,在方差分析中使用了较为别致的数据结构来组织数据。此外考虑到算法与数学模型的关系,文中还介绍了一些数学依据,供同行参考。

## 一、正交表的构造

我们构造的是 $L_t^u(t^q)$ 型正交表,其中水平数 $t$ 限定为素数或素数幂;基本列数 $u$ 可以是任意正整数;给定 $t$ 和 $u$ 两参数后,试验次数即为 $t^u$ ,而表的列数 $q = (t^u - 1) / (t - 1)$ 。构造这样一张正交表可分三步进行。

### 1. 建立有限域运算表

我们知道 $L_t^u(t^q)$ 型表的构造是在一个 $t$ 元有限域 $Z_t$ 上进行的,因此首先必须定义 $Z_t$ 上的加法和乘法运算。当 $t=p$ 为素数时, $Z_t$ 的加法和乘法运算用模 $p$ 同余来定义;当 $t=p^m$  ( $m$ 为大于1的整数)为素数幂时, $Z_t$ 的加法和乘法用 $p$ 域上的不可约多项式同余来定义。前者的有限域运算表只需根据模 $p$ 的同余规则,可以很容易地建立。以下主要介绍当 $t=p^m$  ( $m>1$ )时,相应的有限域运算表的建立方法。

根据多项式理论,对于 $t=p^m$ 时,一定存在一个 $p$ 域上的 $m$ 次不可约多项式 $g(x)$ ,对于集合 $\{ \text{一切 } a_0 \cdot x^{m-1} + a_1 \cdot x^{m-2} + \dots + a_{m-1} \mid a_i \in Z_p, i: 0 \dots m-1 \}$ 定义两个运算:以 $g(x)$ 为模的加法和乘法,即加或乘的结果都用以 $g(x)$ 除后所得的余式来代替。可以证明这个多项式集合对定义的运算作成是一个 $p^m$ 元的有限域[8]。再令这 $p^m$ 个多项式用 $p^m$ 个代号表示,且代号间的运算用相应的多项式运算

得到,则所得的便是所求的有限域 $Z_t$ 及其相应的运算表。不过在实际建立运算表的过程中求多项式的余式是一个极大量的工作,为此我们设计了一个较为快速而简单的算法,称之为移位除法。这个除法的特点是通过移位实现相除。

设被除多项式为 $a_0 \cdot x^n + a_1 \cdot x^{n-1} + \dots + a_n$  ( $c \leq n \leq 2(m-1)$ ),不可约多项式 $g(x) = b_0 \cdot x^m + b_1 \cdot x^{m-1} + \dots + b_m$ ,且分别用数组 $[a_0, a_1, \dots, a_n]$ 与 $[b_0, b_1, \dots, b_m]$ 表示。则移位算法为:

- (1) 如果 $n < m$ 则结束,否则执行(2)至(6);
- (2)  $a_0 / b_0 \Rightarrow ST$ ;
- (3)  $i$ 从1开始到 $n$ 做(4)、(5)循环( $b_i$ 不足时用零补), $i$ 大于 $n$ 时执行(6);
- (4)  $a_i - b_i * ST \Rightarrow a_{i-1}$ ;
- (5)  $a_i \bmod p \Rightarrow a_{i-1}$ ;
- (6)  $a_n$ 置零,  $n-1 \Rightarrow n$ , 返回(1)。

本算法中(4), (5)循环执行一遍,表示被除多项式移位一次,即实行一次相除; (1)中当 $n < m$ 时,表示不能再除,此时数组 $[a_0, a_1, \dots, a_n]$ 便是余式。

### 2. 建立 $Z_t$ 向量空间矩阵

我们是采用矩阵方法来构造 $L_t^u(t^q)$ 表的,为此必须先建立一个 $Z_t$ 上的 $u$ 维向量空间矩阵。记 $Z_t$ 的元为 $0, 1, 2, \dots, r$  (其中 $r =$

$t-1$ ), 由  $Z_t$  元组成的  $u$  维向量  $(a_1, a_2, \dots, a_u)$  ( $a_i \in Z_t, i: 1 \dots u$ ) 全体构成  $Z_t$  上的  $u$  维向量空间  $Z_t^u$ 。显然  $Z_t^u$  中元的个数为  $t^u$ 。依字典序把  $Z_t^u$  中的全体向量排成  $t^u \times u$  阶的矩阵

$$A_0 = \begin{pmatrix} 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & \dots & 1 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ r & r & \dots & r & r \end{pmatrix}$$

据字典序规则, 矩阵  $A_0$  的元素可利用一个函数关系来得到。设  $a_{ij}$  ( $1 \leq i \leq t^u, 1 \leq j \leq u$ ) 为  $A_0$  中的任一元素, 则  $a_{ij} = \text{int}(((t-1) - t_0 \cdot \text{int}((i-1)/t_0))/t_1 + \varepsilon)$  其中  $t_1 = \text{int}(t^{(u-j)})$ ;  $t_0 = t_1 \cdot t$ ;  $\text{int}$  是取整函数;  $\varepsilon$  是一个较小的数, 用于修正因除法所产生的误差。

### 3. 求正交表

按照  $L_{t^u}(t^q)$  的矩阵构造方法, 在建立了  $A_0$  矩阵以后需做以下几步:

(1) 从  $A_0$  中去掉那些第一个非零元素不为 1 的行 (包括全零的第一行), 得  $q \times u$  阶矩阵

$$B_0 = \begin{pmatrix} 0 & \dots & 0 & 1 \\ 0 & \dots & 1 & 0 \\ 0 & \dots & 1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & \dots & r & r \end{pmatrix}$$

(2) 把  $B_0$  的列的顺序颠倒过来, 得矩阵

$$B = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ 1 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ r & r & \dots & 1 \end{pmatrix}$$

(3) 根据  $Z_t$  上的有限域运算表求出矩阵  $A_0 * B'$  ( $B'$  表示  $B$  的转置矩阵), 它是  $t^u \times q$  阶的。最后对  $A_0 * B'$  矩阵依整数加法, 在其每一元素上加 1 就得到了通常见到的  $L_{t^u}(t^q)$  型正交表。

## 二、试验方案的确定

确定试验方案的关键在于: 根据实际问

题选择适当的正交表, 并把各因素按一定的顺序上列。设需要考虑的交互作用个数为  $\alpha$ , 因素个数为  $\beta$  ( $\alpha, \beta$  为非负整数), 水平数为  $t$ , 则选表规则是: 先求出满足不等式  $\alpha(t-1) + \beta < (t^u - 1)/(t-1)$  的最小  $u$  值; 再由  $u$  和  $t$  值确定应采用的正交表。因素上列的规则是: 从第一列开始, 先安排有交互作用的因素, 随后是其交互作用列; 再安排无交互作用的因素; 最后必须留有空列以满足方差分析的需要 (当没有空列时  $u$  应增值)。

然而遗憾的是这一上列规则不能表示为一固定模式供计算机使用。因为常常会出现这种情况: 当某个因素上列后而要进一步安置有关的交互作用列时, 会发现其交互列位置已被其它因素或交互列占用的情况, 此时该因素的位置安排是失败的, 应重新选择。人工上列是采用不断尝试的方法来进行的, 为此我们在算法设计时, 采用了一个非常有效的回溯策略来模拟人工的尝试过程。下面分两部分来介绍试验方案的确定方法。

### 1. 搜索交互列

在要考虑交互作用的问题中, 首先要找出某两个因素的交互作用列在正交表中的位置。  $L_{t^u}(t^q)$  表中任两列都有  $t-1$  个交互作用列。设  $(a_1, a_2, \dots, a_n)'$  和  $(b_1, b_2, \dots, b_n)'$  是表中任意两列 (其中  $a_i, b_i \in Z_t, i: 1 \dots n; n=t^q$ ), 则它们所对应的  $t-1$  个交互作用列按下式确定:

$$\begin{pmatrix} c_1^{(j)} \\ c_2^{(j)} \\ \vdots \\ c_n^{(j)} \end{pmatrix} =_j \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} \quad (j: 1 \dots t-1) \quad (1)$$

其中加乘运算是  $Z_t$  域上的运算。不过这样得到的  $(c_1^{(j)}, c_2^{(j)}, \dots, c_n^{(j)})'$  并不一定是表中的某一列。据正交表定义, 表中同一列中的水平可以置换, 即如果一列经水平间的置换后与另一列相同, 则这两列是等价的。所以我们搜索交互作用列的位置, 也就是搜



素  $(c_1^{(j)}, c_2^{(j)}, \dots, c_n^{(j)})'$  的等价列在表中的位置。

实现后者任务的关键是找出一组水平间的置换关系。假设表中某一列是  $(c_1^{(j)}, c_2^{(j)}, \dots, c_n^{(j)})'$  的等价列, 则一定在一个置换  $\Pi$ , 使  $(c_1^{(j)}, c_2^{(j)}, \dots, c_n^{(j)})'$  经  $\Pi$  置换后与该列相同, 而且置换  $\Pi$  是唯一的。据正交表构造规则, 对某一列的置换  $\Pi$  与该列同一水平的连续个数有关, 设  $sp$  表示该列中同一水平的连续个数, 则置换为:

$$\Pi: \begin{pmatrix} c_{k_1}^{(j)} & c_{k_2}^{(j)} & \dots & c_{k_r}^{(j)} \\ 1 & 2 & \dots & r \end{pmatrix}$$

其中  $r=t-1$ ;  $c_{k_1}^{(j)}, c_{k_2}^{(j)}, \dots, c_{k_r}^{(j)}$  是  $1, 2, \dots, t-1$  的一个排列, 下标  $k_i (i: 1 \cdot \cdot r)$  由下式确定:

$$k_i = (i-1) \cdot SP + 1 \quad (i: 1 \cdot \cdot r) \quad (2)$$

$(c_1^{(j)}, c_2^{(j)}, \dots, c_n^{(j)})'$  经  $\Pi$  置换后得到其等价列  $(c_{1*}^{(j)}, c_{2*}^{(j)}, \dots, c_{n*}^{(j)})'$ , 如果该列与表中假设的等价列相同, 则说明已找到了一个交互列, 否则应重新假设其它列为等价列并进行相应的处理。

搜索某两列的所有交互列的算法为:

(1)  $j$  从 1 到  $t-1$  做 (2) 至 (6) 循环, 否则结束;

(2) 据 (式 1) 得到  $(c_1^{(j)}, c_2^{(j)}, \dots, c_n^{(j)})'$ ;

(3)  $k$  从 1 到  $q$  列 (除被考虑交互作用的两个因素所在的列外) 做 (4) 至 (6) 循环;

(4) 按 (式 2) 规则得置换  $\Pi$ ;

(5) 按  $\Pi$  规则对  $(c_1^{(j)}, c_2^{(j)}, \dots, c_n^{(j)})'$  进行置换, 得到  $(c_{1*}^{(j)}, c_{2*}^{(j)}, \dots, c_{n*}^{(j)})'$ ;

(6) 如果  $(c_{1*}^{(j)}, c_{2*}^{(j)}, \dots, c_{n*}^{(j)})'$  与  $k$  列相同, 则  $k$  列便是一个交互列, 并返回 (2) 做下一  $j$ ; 否则返回 (4) 做下一  $k$ 。

该过程结束时, 两个因素的  $t-1$  个交互作用列也就确定了。

## 2. 因素安排的回溯策略

选择正交表的原则是尽量地小, 以便使试验次数尽量地少, 但这样就不可能把所有因素都安排在正交表的基本列位置上, 因而也就有可能出现当某个因素上列后, 有关它的一些交互作用列的位置已被占用的情况。这一现象当被考虑的交互作用所涉及的因素个数较多时尤为明显。显然, 这是一种不合理的现象, 解决问题的关键在于给各个因素安排一个合理的位置。然而, 确定因素的一个合理位置并不能用某一具体的规则来实现, 它必须通过多次地试探、纠错才能被确定。每当一个因素上列后就应立即分析被考虑的有关交互作用列在表中的占用情况, 当发现有一列被占用时马上取消该因素的安排位置, 并重新选择一个位置及进行相应的试探。这一过程直至对有关的交互列安置都不发生冲突为止。

上述回溯过程的算法设计我们借助于一张交互列记录表 (简称 AR 表) 来完成。AR 表是一张动态建立的交互列记录表, 每当一个因素上列后它就登记被考虑的有关交互列列号。(交互作用的另一因素必须是已经安置在正交表中的) AR 表的作用是存放试探过程中的交互列列号, 一当因素安置失败就取消该表, 进而根据因素的重新安置再建立相应的 AR 表, 直至成功为止。

如: 有一水平数  $t=3$  的问题, A、B 两个因素及有关的交互作用已安置在表中, 现安置因素 C 在第 5 列, 被考虑的交互作用为  $A \cdot C$ 、 $B \cdot C$ , 则所建立的 AR 表为表 1:

表 1

$A \cdot C$	$B \cdot C$
1	1
6	8
7	9

表中第一行的取值为 1 或 0, 分别表示所考虑的交互作用有无, 第二列  $t$  行表示各交互作用所对应的  $t-1$  个交互作用列列号。

因素安排的回溯算法为：

(1) 确定某因素与前面各因素被考虑交互作用的情况,即设置AR表中的第一行;

(2) 把该因素安置在表中列号最小的空列中;

(3) 对AR表中被考虑交互作用的每列做(4)、(5)循环;

(4) 据上节的算法,找出这两个因素的 $t-1$ 个交互列位置,并填入AR表;

(5) 分析AR表中各交互列位置被占用的情况:如果有一列位置已被占用,则取消该因素的安排位置,且置该位置为占用状态,并返回(2);否则返回(4)做下一个被考虑的交互作用;

(6) 把AR表中所有的交互列列号安置到正交表中。(因为此时所有交互列都不会发生安置冲突。)

安排好因素后,把因素的各水平在表中对号入座,并列出试验条件(表中的每一行代表要试验的一种条件)。每种条件试验一次,这就获得了试验方案。

### 三、试验结果分析

当试验结果输入后,我们采用方差分析方法对试验结果进行分析。在方差分析中首先把试验结果数据的总平方和分解到 $L_t$ 、

( $q^t$ )正交表的各列上去,然后计算各个因素、交互作用和误差项的平方和。为计算方便起见我们采用链方式来提供求和项的线索。建链的原则是:把同一交互作用的 $t-1$ 个交互列用一条链连接起来,所有空白列用一条链连接起来,每个因素的链就是因素所在列本身。这些链的建立可大大提高搜索效率,从而加快平方和计算的速度。

#### 1. LINK表的建立

建链工作我们借助于一张LINK表来完成。LINK表有三行组成:第一行就是各因素及其交互作用在正交表中的安排,第二行由0、1、2、3组成,其中1表示因素的链头,2表示各交互作用链的链头,3表示空列链链头,0表示非链头;第三行由0到 $q$ 的数字组成,其中0表示链尾,其它数字表示该链中下一元素的所在列号。

例如:某一问题中水平数 $t=3$ ,因素四个为A、B、C、D,要考虑的交互作用为 $A*C$ 、 $B*C$ ,则它的LINK表为表2:

表2

列号:	1	2	3	4	5	6	7	8	9	10	11	12	13
A	B			C	$A*C$	$A*C$	$B*C$	$B*C$	D				
1	1	3	0	1	2	0	2	0	1	0	0	0	
0	0	4	11	0	7	0	9	0	0	12	13	0	

LINK表的建表算法为:

(1) 置第二行均为-1;

(2) 从最后列开始向前扫描,对第二行为-1的列做(3)循环;

(3) 如果是空列,则加入到空列链中;

如果是因素列,则建立只有该因素的一条链;

如果是交互列,则搜索出其它所有 $t-2$ 个同交互作用的列,并建立一条链。

#### 2. 方差分析

方差分析过程分几步完成。

(1) 计算各列各水平所对应的试验数据和:

设 $y_i (i: 1 \cdots n)$ 表示第 $i$ 次试验的结果数据,用 $T_{jk}$ 表示第 $k$ 列中第 $j$ 号水平所对应的试验数据之和( $j: 1 \cdots t, k: 1 \cdots q$ )。所有 $T_{jk}$ 数据构成一个 $t*q$ 矩阵,该矩阵有时在分析最优水平时要用到。

(2) 计算平方和与相应的自由度:

令  $G = \sum_{i=1}^n y_i$ , 用  $S_k$  表示第  $k$  列上的平方和,

$$S_k = (t * \sum_{j=1}^t T_{jk}^2 - G^2) / t^2$$

(k: 1...q)

且  $S_k$  的自由度为  $f_k = t - 1$  (k: 1...q)。

据定义, 单个因素的平方和与相应的自由度是指该因素所在列的平方和与自由度; 交互作用的平方和是指相应的  $t-1$  个交互作用列的平方和之和, 其自由度是相应的  $t-1$  个交互作用列的自由度之和, 即  $(t-1)^2$ ; 误差项的平方和是指所有空白列的平方和之和, 其自由度为各空白列的自由度之和。

平方和的计算我们可按照 LINK 表中的链来进行。在计算中一当某一链的平方和计算完毕后, 该链就不再需要, 为提高存储空间利用率及便于搜索, 我们把该链的平方和存放在该链头所在列的第三行中。这样在以后的计算中, 某一项目的平方和可按第二行的链头内容找到相应的平方和。

(3) 计算均方和与 F 比值:

某一项目的均方和是指该项的平方和与其自由度之比。这一计算也可在 LINK 表中完成。据 2 的处理, 所有项目的平方和均在 LINK 表中第二行不为 0 列 (即链头) 的第三行上, 而且有: 链头为 1 的自由度等于  $t-1$ 、链头为 2 的自由度等于  $(t-1)^2$ 、链头为 3 的自由度等于  $c(t-1)$  (其中  $C$  为空列个数)。求出各个项目的平方和与其自由度之比便得到相应的均方和, 再把均方和存入原平方和的位置 (平方和已不再需要)。

F 比值是该项的均方和与误差项的均方和之比 (误差项没有 F 比值)。这一工作也可在 LINK 表中完成, 而且可再次把各项的 F 比值 (除误差项外) 存入原均方和的位置。注意当 F 比值小于 1 时, 需对换其分子分母以保证 F 比值不小于 1, 这样能简化临界值的查找工作, 同时也应对换其自由度。

(4) 显著性判断:

把 F 比值与临界值相比, 超过临界值者为显著, 否则就不显著。我们用符号 “\*\*\*”、“\*\*”、“\*” 分别表示在信度  $1-\alpha_0=0.99$ 、0.95、0.90 之下是显著的, 在  $1-\alpha_0=0.90$  之下仍不显著时, 我们就认为该因素或交互作用对考核指标的影响是不显著的。

(5) 选择最优水平:

方差分析观点认为: 只需对显著的因素或交互作用选出最优水平即可, 而不显著的原则上可选在试验范围内的任一水平或由其它指标去确定。对于显著的因素求出相应的  $\max \{T_{jk}\}$  或  $\min \{T_{jk}\}$ , 当指标越

大 (越小) 越好时, 取  $\max \{T_{jk}\}$   
 $1 \leq j \leq t$

( $\min \{T_{jk}\}$ ) 所对应的水平为最优。对于  $1 \leq j \leq t$

显著的交互作用, 则应先把相关的两个因素的每一种水平组合所对应的全部试验结果数据相加, 得到  $t^2$  个数据, 然后对它们进行比较, 当指标越大 (越小) 越好时, 取这两个因素的最优水平能使这  $t^2$  个数据中达到最大 (最小) 所对应的水平组合。当单个因素及其交互作用都显著时, 最优水平的选择要服从交互作用。最后根据上述分析, 结合其它各方面的因素, 得出一组最优水平组合。

## 四、结论与应用实例

根据上述设计思想设计的计算机软件已在 IBM PC 机上用 BASIC 语言实现, 它的最大特点是从正交表的构造、试验方案的安排到试验结果的分析均由计算机完成, 速度快、精确度高, 特别适用于多因素多水平多交互作用的正交试验设计, 在应用上有着广泛的前景。本系统经过多次实际使用, 证明性能良好、使用方便, 很易于推广。下面以一个应用实例来介绍本系统的使用方法。

浙江临海市丝厂的煮茧工艺是整个生产过程的关键环节, 它由多道加热工序组成。为提高丝的产量, 希望通过正交试验设计找出最优温度配合条件。提高丝产量的标志是

提高蚕茧的解舒率，故选用解舒率为考核指标。根据专业知识和生产经验，选定的因素及水平如表3：

表3

因素	A	B	C	D
后动摇温度 (°F)		中水温度 (°F)	静煮温度 (°F)	低渗温度 (°F)
水平				
1	204	198	185	135
2	207	202	190	140
3	210	206	195	145

要求考虑两个交互作用：A\*B与A\*C。据此得交互作用个数 $\alpha=2$ ，因素个数 $\beta=4$ ，水平数 $t=3$ 。程序执行时，据提示依次输入 $t$ 、 $\beta$ 、 $\alpha$ 的值，就可得到试验方案如表4。

Table of Experimental Schemes

表4

NO.	A	B	C	D	NO.				
1	1	1	1	1	15	2	2	3	1
2	1	1	2	2	16	2	3	1	3
3	1	1	3	3	17	2	3	2	1
4	1	2	1	2	18	2	3	3	2
5	1	2	2	3	19	3	1	1	1
6	1	2	3	1	20	3	1	2	2
7	1	3	1	3	21	3	1	3	3
8	1	3	2	1	22	3	2	1	2
9	1	3	3	2	23	3	2	2	3
10	2	1	1	1	24	3	2	3	1
11	2	1	2	2	25	3	3	1	3
12	2	1	3	3	26	3	3	2	1
13	2	2	1	2	27	3	3	3	2
14	2	2	2	3					

经过试验测得数据如表5：

表5

试验号 i	结果数据 yi	试验号 i	结果数据 yi	试验号 i	结果数据 yi
1	60.60	10	60.50	19	61.04
2	60.01	11	64.86	20	60.27
3	59.30	12	66.34	21	63.05
4	60.75	13	63.31	22	61.39
5	60.63	14	62.22	23	60.26
6	64.13	15	62.79	24	66.35
7	67.62	16	60.92	25	59.84
8	68.90	17	65.19	26	60.82
9	67.37	18	65.64	27	59.56

依次输入这些数据，得到各列各水平的数据和为表6：

表6

level	A	B	A*B	A*B	C	A*C	A*C	D
I 1	569.31	555.97	559.66	548.45	555.96	565.08	570.19	570.32
I 2	571.77	561.83	557.43	561.62	563.15	563.23	566.57	563.15
I 3	552.58	575.86	576.57	583.58	574.52	565.33	556.87	560.17

按提示输入各临界值：

## PC机软硬盘故障的排除

随着IBM-PC系列机的大量推广运行,硬件故障都不可避免的经常发生,而多数软驱动器故障属非电路故障,是用户可以自行排除的,不必送到维修部门,笔者总结如下:

### 一、硬盘驱动器

用户使用的硬盘驱动器,大多是温彻斯特驱动器,它头盘密封,且接触起停,使用寿命一般可达五年以上。

#### 故障现象一:

1.用户在调用某个或某几个文件时,系统易死机。运行其它文件正常。

故障原因:1.记录该文件的磁盘扇区有坏磁道,因而磁头在读写时出错,造成死机。2.该文件被感染上了“犹太”(Jerusalem耶路撒冷)病毒,从而造成死机。

排除方法:1.停止该文件的使用,将其命名为其它文件,再拷入正确的文件,如在使用:dbase.exe文件时经常死机:C>rename dbase.exe、dbase.old

• C>copy a: dbase.exe

2.调用解病毒程序SCAN: A>SCAN/D C: 一经查出某文件被感染了 Jerusalem病毒,即删除该文件,再重新拷入新的原始文件。

#### 故障现象二:

系统不能从硬盘中引导,但从软盘中引导系统后,可转C:盘工作。

故障原因:1.command.com 文件被破坏,如被删除,或拷入了不同版本的command.com文件。2.磁盘引导扇区文件被感染了“大麻”病毒,遭到破坏。

排除方法:1.恢复原C盘中相同版本的系统文件,或是用高版本的操作系统,采用版本升级的方法解决。如用DOS3.3,从

软盘中引导DOS3.3版系统后:

(1) A>sys C: ↵

(2) A>Replace a: \ \* \* C:\S/R ↵

(3) A>Replace a: \ \* \* C:/a ↵

2.确诊引导扇区被感染了“大麻”病毒,用DMC消除“大麻”之后,C盘仍不能引导,只有彻底格式化,方法见故障现象三的格式化方法、若“大麻”病毒只是感染了分区表(partition table),消除“大麻”后,仍可继续使用,对于其它病毒,消除后仍能正常运行。

#### 故障现象三:

系统不能从硬盘中引导,通过A盘引导,系统后,也不能转入C盘或D盘。

故障原因:1.对于AT机、286、386等机可能是set-up参数设置错误。2.磁盘标识文件被破坏,或是有坏磁道(包括00磁道)。

#### 排除方法:

1.对于第一种故障,可通过软盘引导完系统后,运行set-up,设置正确的磁盘类型参数即可。若有的设置不正确,重新引导后,系统仍会提示让设置正确的参数。

2.若设置也正确,仍不能使用硬盘,一盘只有彻底格式化来解决,即完成初始化→分区→系统格式化。

#### 格式化方法:

1.直接调用专用格式化程序,如运行DM程序,并正确回答完程序的提问,或引导0520C-H机20MB专用格式化程序。DM程序比较适用286、AT机、386等机,或有标准硬盘的XT及其兼容机。

2.通过debug程序调用磁盘适配器板上的初始化程序(只适用于XT及其兼容机):

即: a>debug

-g=C800:5

并回答提问。对大于32MB的磁盘,应将

其划分成两个盘，否则格式化后，只能使用32MB的空间，一般DOS只支持32MB的系统盘，初始化完后，再完成：

a>fdisk

创建DOS分区后，引导系统，再完成：

a>format c: /s

a>format D: (如分C、D两盘)

3.调用高级诊断程序，或DOS版本下的初始化程序，再完成FDISK，FORMAT程序，再装入系统文件。

通过彻底格式化之后，一般故障均可排除，但有用户可能遇到格式化后，系统运行一段时间就又不能使用硬盘，再查，又是病毒感染。原因是软盘系统就有病毒，或是用户程序有病毒未彻底消除重新感染硬盘系统造成的。有的微机或在初始化或格式化时报出大量的出错信息，这种故障一般出在主机系统板或磁盘适配器板上，这些板上的可拔插的处理芯片、接口芯片、ROM芯片等，引脚形成氧化层，使接触不可靠，用户只需将这些可拔插芯片拔出，去除氧化层，再用纯酒精清擦，正确装回，再重新完成格式化，这种故障一般便可以排除。

## 二、软盘驱动器

在PC系列机上大多使用360KB和1.2MB两种容量的软驱，读写时，磁头紧贴盘片。

故障现象一：

文件在读写时报出错信息，或不能从软驱中引导系统文件，或时好时坏。

故障原因：1.对于这种故障，有的是软盘片磁道上有明显划伤，文件被破坏，因而不能读写。2.在盘片正常的情况下，仍不能读写，则是由于磁头太脏的原故造成。

排除方法：1.用新的盘片记录文件。或对划伤的盘片在DOS3.0以上版本下重新格式化可跳过坏的磁道。2.用清洗盘沾上磁头清洁剂清洁，并多次使用dir A:或B:命令，即可排除故障。

故障现象二：

只能用一个磁头，即格式化也只能用一面。

故障现象三：

磁头小车不走动，整个软驱都不能使用，或A、B两个软驱都不能使用。

故障原因：上述两种故障现象用户可能怀疑有一个磁头已烧坏，或怀疑带动磁头小车的步进马达烧坏，因而送去修理，但多数情况下并非如此，其原因是：由于目前使用的软驱、盘片类型较多，盘片质量即磁粉的涂敷工艺有好有坏，由于磁头与盘片紧贴着运转，质量差的盘片，或经常调用的盘片在一段时间之后，盘片划伤，磁粉脱落，堵塞在驱动器的磁头缝隙之中，磁头一堵，就会减弱磁头的磁场强度，使磁头不能使用。1号磁头（上面磁头）堵塞较多。因此出现只能用一面盘片的现象，或有的文件能读，有的不能读。若0号磁头（下面磁头）一堵，00磁道信号、索引文件均无法读出，整个磁头小车就不能走动。

排除方法：准备一张格式化过的新盘，一张清洗盘及清洁剂，高级诊断程序以及DOS系统（最好是DOS3.0以上版本），若A、B两软驱均不能使用，则需准备一个好的软驱。①从好A盘中引导系统（如C盘能使用则可引导C盘中的系统或C盘中高级诊断程序），用清洗盘清洁磁头，并依靠多次使用DIR B:命令来完成。②再调用高级诊断程序至主菜单下，进行1、2两项测试，即做FORMAT Diskettes和Copy两项（此时应将格式化过的新盘插入被测软驱）。③运行至软盘诊断级，完成1→4的各项测试。若在1、2两项测试中，报为双面（Double side）软盘，则表明两磁头已能读出信息，若仍为单面（single side）软盘，则重复①→③。④在DOS系统下格式化软盘，有的报出较多的坏磁道，则还需清洁磁头，并进行多次格式化即可，最后坏磁道消失，软驱正常。

值得注意的是：A、B两软驱的区分，有的微机是直接由扁状线上的箭头来区分，有



## 在程序中消除和恢复光标的方法

在程序运行时，为了加快显示输入的速度，往往直接对显示缓冲区进行操作，这就要暂时去掉光标。这里我们介绍几种在PC机上消除和恢复光标的方法。

第一种方法是利用6845CRT控制器来实现的,6845CRT控制器是用来控制屏幕输入的部件,光标是由它的10号、11号两个寄存器的内容决定的,其中10号(光标起始地址座标)寄存器的五、六两位决定了光标的有无,两个寄存器的剩余位用来设置光标的形状。可见要消除光标只要将10号寄存器的五、六两位置成1、0即可。如果要恢复光标只要将正常光标的参数送到上述两个寄存器即可实现。

消除光标的程序	OUT DX, AL
MOV DX, 3B4H	MOV DX, 3B5H
MOV AL, 10	MOV AL, 4BH
OUT DX, AL	OUT DX, AL
MOV DX, 3B5H	MOV DX, 3B4H
MOV AL, 20H	MOV AL, 11
OUT DX, AL	OUT DX, AL
恢复光标的程序	MOV DX, 3B5H
MOV DX, 3B4H	MOV AL, 0CH
MOV AL, 10	OUT DX, AL

第二种方法是利用 BIOS 中断实现,在 BIOS 中并没有直接提供消除光标的功能,但我们可以灵活运用 BIOS 中某些功能来实现,具体的实现方法有两种:

1. 通过INT 10的一号功能来实现, 这个

功能是用来设置光标形状的,调用此功能前,光标的起始座标和终止座标分别送到CH和CL寄存器中,在正常操作中,起始座标应小于终止座标。如果我们使起始座标大于终止座标即可消除光标,如要恢复光标只要将使用正常参数调用此功能即可。

2.利用INT 10的五号功能来实现的,这个功能是用来设置光标的位置,如果将光标设置在屏幕以外,同样可以达到消除光标的目的。

具体程序如下:

## 利用一号功能 消除光标

```
MOV AH, 01H
MOV CX, 0201H
INT 10H
```

## 恢复光标

```
MOV AH, 01H
MOV CX, 0B0CH
INT 10H
```

## 利用五号功能

### 消除光标

```
MOV AH, 05H
MOV CX, FFFFH
INT 10H
```

## 恢复光标

```
MOV AH, 05H
MOV CX, 0
INT 10H
```

武汉大学计算机系      姚琥      宋光爱



































的是用软驱上的DS0.1.2.3来区分。如长城0520就是跨接DS0来表示A盘，DS1跨接来表示B盘，用户用好盘来做A盘时，则需跨接DS0，所需修的盘跨接DS1。另外，对于1.2MB的软驱最好也用一个好的1.2MB的软驱来做A盘，且将A、B盘均设置为

1.2MB软驱，以利于诊断测试中的COPY和VERIFY两项测试。若用户坚持每半个月对软驱进行一次清洗磁头，则可防止以上故障的出现，对已擦伤的盘片最好停止使用。

四川绵阳89959部队计算机室 肖伟民