

计算机应用研究

ATION RESEARCH OF COMPUTERS

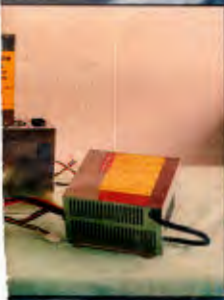
1994 1

计算机处理系统
 动播音系统
 用系统
 软件
 微机
 计算机

新潮电脑

时代新潮

秤



1001-3695



四川新潮计算机产业集团公司
 中国四川成都新鸿路四号 邮编: 610051
 电话: 4448100、4444670 传真: 4444115

“安”和“稳”的故事……



“凯联”追求稳定的电压和自身的稳定!

我厂专业生产稳压电源。SBW、DBW系列三相、单相全自动大功率补偿式稳压器，是专为电网电压波动而设计的，是国内目前最为先进的节能型稳压电源。

SBW、DBW系列具有：功率大、损耗低、稳压精度高、输入范围宽、能承受瞬间超负载……。它广泛应用于邮电、工业、医院、交通、科研、宾馆……等一切需要电压稳定的场所。也是所有进口设备的必配电源。

主要技术指标及规格：

指标：稳压精度：额定电压 $\pm 1\sim 5\%$ (可调)
稳压范围：额定电压 $\pm 20\%$ 损耗：1.5% (平均)
波形畸变：0.1% 应变速率： $\leq 0.1s$ 有过压、故障自动保护
规格：三相 10, 20, 30, 50, 100, 180, 225, 300, 400, 500, 600, 800, 1000 (KVA)
单相 3, 5, 10, 20, 30, 50, 100, 180, 225, 300 (KVA)

“凯联”服务宗旨：上门调试 代办托运 来询即答 终身服务

SBW、DBW——系列 全自动大功率补偿式稳压器

凯联



稳压



上海凯联稳压设备厂

地址：上海市常德路379号 电话：2537960 2553368 传真：021-2553368 电挂：1454 邮编：200040

经 销 商：
• 山东省机电设备总公司 地址：济南市经五路225号 电话：733723
• 陕西省机电设备总公司 地址：西安市西五路22号 电话：775249
• 湖南省机电设备总公司 地址：长沙市五一中路6号 电话：420645
• 南京市机电设备总公司 地址：南京市太平南路514号 电话：2215084
• 浙江省机电设备公司 地址：杭州市延安路217号 电话：555646
• 闽东电机(集团)公司 地址：福州市五一一路144号 电话：524452

为软件技术产品标个价

——计算机软件技术产品价值评估系统面世

甘肃省计算中心的曾光初等研究人员,在广泛开展社会调研、大量查阅国内外文献资料、充分汲取有关单位有益经验、认真听取专家意见、反复进行课题论证基础上,经过两年多潜心研究与刻苦攻关,于最近完成了一项软件学研究计划项目——计算机软件技术产品价值评估系统。

此系统的研究工作是在深化改革、进一步开放和软件产业、软件市场迅猛发展之形势下进行的。研究人员针对如何准确描述计算机软件技术产品的经济价值和社会价值,如何准确描述软件技术产品的共性和个性,如何建立定量评估的数学模型和程序系统,如何建立适应我国目前软件市场状况的估算模型和系统,如何正确坚持劳动价值与使用价值相统一、价格与功能相统一、研制开发与市场需求相统一的观点等一系列复杂问题,建立起了一套较为完整、科学、规范的软件技术产品价值、价格评估体系,是一项关系到国民经济建设和科技进步的重大研究项目,具有相当重要的学术价值及现实意义。

课题研究人员对软件技术产品的各种内在和外在属性进行了深入探讨,明确提出了“软件技术产品价值评估的核心是其本身的价值问题”这一重要概念,提出了软件价值评估应以“价值确定价格”、“价格体现价值”的设计思想,并以软件技术产品的文档资料齐备性、软件性能良好度和其他综合因素的优良度等特性作为价值因子,从而对其软件价值高低进行评估;该系统首次提出将多迪(Doty)评估模型与国内应用较为成功的模型联合使用,从而建立了一套接近于我国国情的软件成本估算模型;首次提出将价值因子的定量评估直接作用在成本模型上,从而建立了由软件产品价值高低确定并体现软件产品寿命的软件估价模型。该系统通过评估模型所设计的软件价值评估系统,对多例软件成果分析和评估结果证实,此项课题提出的评估思想正确,估价模型合理,评估系统新颖实用,评估结果与我国当前软件市场现状相吻合。

计算机软件技术产品价值评估系统的问世,对于软件产业化和建立完善的软件市场具有明显的推动和催化作用,对于软件产业的科学管理、正确决策和防止软件侵权、裁决侵权纠纷等,具有一定借鉴参考作用,是适应当前改革开放、社会经济发展的应时之需,对于加强软件产品开发、销售、使用、保护等各个环节的科学化、规范化管理,解决软件产品商品化过程中价值与价格相统一的难题,为处理软件产品销售的定价与软件侵权中经济仲裁提供了令人可信的估价依据,因而具有明显的学术价值和实用推广价值。

《计算机应用研究》杂志办刊单位

四川省电子计算机应用研究中心
贵州省科委计算中心
安徽省计算中心
吉林省计算中心
内蒙古电子计算中心
云南省电子计算中心
四川省电子学会

新疆电子计算中心
甘肃省计算中心
广西计算中心
山东省计算中心
河南省计算中心
青海省测试计算中心

《计算机应用研究》杂志社董事会

董事长:周赛渝

董 事:管小宝 陆慰椿 闫长荣 王升亮
李天健 郑国基 叶大卫 乔中南
冯德成 朱 华 王俊杰

《计算机应用研究》杂志编辑委员会

主 任 委 员:张执谦

副主任委员:李泽民

委 员:贾洪钧 曾光初 龚宇清 罗海鹏
张湘金 张国栋 范德元 李文华
刘铁军 崔振远 刘启茂

1994 年 1 期(总第 57 期)

出版日期:1994 年 1 月

责任编辑:张 钢

计算机应用研究(双月刊)

J I S U A N J I Y I N G Y O N G Y A N J I U

(公开发刊)

刊 号:CN51-1196/TP(国内)
ISSN 1001-3695(国际)

邮发代号:62-68(国内)
BM4408(国外)

主 编:张执谦

副 主 编:李泽民

编 辑 出 版:《计算机应用研究》杂志社

通 讯 地 址:成都市人民南路 4 段 11 号附 1 号

邮 政 编 码:610041

电 话:(028)—5582666 转 2055

印 刷:成都市新都华兴印务有限公司

订 阅 处:全国各地邮政局

国内总发行:成都市邮政局

国外总发行:中国国际图书贸易总公司

计算机应用研究 第 11 卷 第 1 期(总第 57 期)

目 次

综述评论

Pentium 系统概述	王义章 朱华(1)
计算机中的新技术——Flash Memory	张戟(4)

研究探讨

手写体汉字识别的神经网络算法	刘健勤(6)
RAID 的性能评价与比较	骆新国 张江陵(8)
关于《计算机在数论中应用》的几个问题	刘忠(11)

应用实践

实用可靠的 Foxbase 应用系统加密工具 Foxlock	王晓武(12)
在工作站上用 C+XWindow 进行图形程序设计	袁道华 李炳法 黄金姬(13)
VGA 卡的高效模块化图形程序设计	蒋日东(15)
DBMS 数据库设计说明书的集成写法	傅春英 刘杰(18)
以彩色画面为背景的三层菜单生成程序	午锁平(21)
美术字及复杂图形的制作及在程序图形界面中的应用技术	余滨(23)
谈在汉字系统环境下单字节边框与汉字同屏显示的实现	殷霞(27)
计算机访存冲突的对策与实现	罗建明(29)
计算机升级从 CGA 显示方式到 EGA 显示方式	明星 杨平(31)
设计用户界面应注意的问题	罗海鹏 麦翠荣 黎家珍(33)
关于《TVGA256 色图像截取技术》中的 DOS 重入问题及其解决方法	熊文龙(35)
软件温度计的设计	翁正科(40)
一种工业控制机中使用的通用型智能显示终端	蒋大林(41)
智能型电炉电源供电装置	闫立恒 王延平 吕文 闫彦(43)
一种电梯微机控制系统的设计	张昆(46)
歼击机空——空格斗导弹攻击模拟系统的过程控制	陈德金 陈德煌 曹永清(49)
饲料生产中的微机控制系统	赵文安(53)

维护维修

计算机耐故障技术的探讨	王玉昆 杨国才(56)
M1742 打印机故障维修六例	陈永红 穆大明(59)
IBM4224 打印机故障排除三例	董正平(61)

动态简讯

山东省计算中心简介	(3)
本刊启事及简讯	(7、20、28、52、61)
新技术产品简介	(42)
上海凯联稳压设备厂产品介绍	(62)

APPLICATION RESEARCH OF COMPUTERS

Vol. 11 No. 1 (Total 57)

CONTENTS

SURVEY

- The View of Pentium System Wang Yi Zhang et al. (1)
A New Technology on Computer—Flash Memory Zhang Ji(4)

RESEARCH

- Algorithm of Neural Network on Distinguishing Script Chinese Characters Liu Jian qin(6)
A Complete Evaluation and Comparison for RAID Luo Xin guo et al. (8)
A Few Questions about《the Applications of Computers in Number Theory》 Liu Zhong(11)

APPLICATION

- Practical and Reliable Foxbase Applied Systems Encryptic TOOL of Foxlock Wang Xiao wu(12)
Designing Graphic Routine on Working Station by C+X Window Yuan Daohua et al. (13)
Hight Efficient Modular Graphic Program Design of VGA Jiang Ri dong(15)
Integrated Writing Method for Designing Manual by DBMS Database Fu Chun ying et al. (18)
Threé Layers of Menu Generator on the Background of Colour Image Wu Suo ping(21)
Making Artistic Calligraphy and Complicated Graphic and Application Technology in Program Graphic Interface Yu Bin(23)
Discussing Realization of Displaying Single Byte Frame together With Chinese Characters in Chinese System Yin Xia(27)
Countermeasures of Computer Access Memory Contention and Its Implementation Luo Jian ming(29)
Computers Going up from CGA Display Modes to EGA Ming Xing et al. (31)
Should Pay Attention to Problem for Designing User's Interface Luo Hai peng et al. (33)
Question about Inputting DOS Again in 《Cutting And Drawing Technique of TVGA 256Colours Image》 and Its Resolving Method Xiong Wen long(35)
Designing Thermograph of Software Weng Zheng ke(40)
A Common Smart Display Terminal in Industrial Controller Jiang Da lin(41)
Intelligent Supply Device of Electric Furnace Source Yan Li heng et al. (43)
A Design of Controlling System of Lift by Microcomputer Zhang Kun(46)
Process Control of Air—to—air Wrestle Missile Attack Simulator System by Fighters Chen De jin et al. (49)
Controlling System in Fodder Production by Microcomputer Zhao Wen an(53)

MAINTENANCE

- Inquiry of Computer's Fault Tolerance Technology Wang Yu kun et al. (56)
Six Examples about Maintenancing Faults of M1724 Printers Chen Yonghong et al. (59)
Three Examples about Fixing Breakdown on IBM4224 Printers Dong Zheng Ping(61)

INFORMATION

- Brief Introduction about Compute Centre of Shandong Province (3)
Notice of our Magazine and Information (7,20,28,52,61)
Brief Introduction of Advanced Technical Products (42)
Introducing Products of Kailian Regulated Equipment Factory of Shanghai (62)

Pentium 系统概述

贵州省科委计算中心

王义章

朱 华

(贵阳 550002)

摘要 本文就第五代芯片——Pentium 的问世及 Pentium 系统(586)的研究动向作一扼要概述,并对 Pentium 系统结构设计方面的考虑作简要介绍。

关键词 电脑芯片 体系结构 Pentium

自 Intel 公司今年 3 月 22 日在全球同时宣布:第五代电脑芯片——Pentium 正式批量投产以来,已把个人电脑迅速推向新一代,有人称 Pentium 正在使个人电脑脱胎换骨,Pentium 即俗称的 586 微处理器,已经引起各方面极大的关注,本文就有关情况作一扼要综述。

Pentium 是 64 位微处理器,有 60MHz 和 66MHz 两种档次。集成度达 310 万个晶体管,差不多是 Intel486 处理器的 3 倍。Pentium 功能强大,被认为是 Intel 芯片发展过程中又一具有里程碑意义的产品。单就运算速度而言,也就超越某些大型电脑的中央处理器,同一应用软件,在 Pentium 强力浮点运算单元上运行,速度比在 33MHz 的 Intel486DX 上运行快 5~10 倍。Pentium 处理器既可提供上述水平的性能,又可和以往的微处理器全面兼容,但是,如果只是简单的将 Pentium 植入原先的 8086/8088 芯片和 8 位 I/O 数据线所设计的体系结构上,即将 Pentium 屈就于传统的 PC 设计上,无异于“好马拉破车”。不言而喻 I/O 瓶颈等更为突出。因此,Pentium 的问世,也将促进微电脑体系结构上的变化、发展。

为抢先上市,各厂商都在力争推出自己的 Pentium 系统,目前首批上市的 Pentium 系统中,绝大多数都是采用可升级的设计,并未完全发挥这套 CPU 的功能,正如转型自 386 的 486 系统一样,较快速的处理器采用的仍然是旧有设计,只是将处理器与存储器组合可以配合 Pentium 芯片达到最佳状态。但是,就系统整体而言,仍应属于 386 或 486 的应用范围。正如惠普公司开发专家 Jim. Mathios 所说:“第一代的 486 系统是采用补丁式的方案,也就是将 486CPU 直接放到最合适的 386CPU 使用的机壳内,同样的情形也发生在 Pentium 上。目前,所有的厂商都是就原有的 486 设计,焊接出差强人意的 Pentium 电脑系统。”虽然如此,首批推出 Pentium 系统的厂商却认为,第一批购买者对成本的考虑较少,因此,系统都拥有非常丰富的配置,位于加州 Fremont 的系统主板制造公司 Computer Systems 的产品经销经理 TonyTong 认为,Intel 公司提供的

Pentium 系统的参考设计是采用高性能、高成本的元件,例如 Pentium 系统可能提供以 VRAM 为主的图形加速器,内建的 SCSI-2 接口系统,用于连接外设和主机系统,基于局部总线的 SCSI-2 驱动器,光盘机,或者通过 DSPs (Digital Signal Processors——数字信号处理器)来支持多媒体配备,或提供专门的声音/影像芯片。当 Pentium 系统较为普及之后,这些豪华的功能是否能大量的销售市场,目前尚无定论,有人认为厂商在这个时候将这些特色纳入系统中,可以提高性价比,从而提高销售量。视讯芯片制造公司 S₃ 的 NealMargulis 认为:“这些功能显然已是 Pentium 系统的标准配备,将来不会有太大的改变”。“Pentium 系统少了这些功能客户是不会买的”。另外则有人认为,从目前人们对 PC 机的价格承认来看,将来市场势必出现功能削减的 Pentium 系统。

一般认为,Pentium 系统将会在服务市场首先获得使用者的青睐。Intel 公司目前控制着世界 PC 机的芯片市场,通过 Pentium 的开发生产,公司将进入高档服务器市场,与基于高档 RISC 的工作站,甚至与传统的大型机开展角逐。据称,采用 2 或 4 个处理器的 Pentium 系统,其计算能力足以与最大的 ES/9000 相匹敌。因此,有人预测,至少在近期,Pentium 会在服务器领域独领风骚。位于 Plymouth, MN 的 TricordSystems 公司已计划采用 Pentium 芯片,作为其 EC5000PowerFrame 服务器的中央处理单元,NEC 技术公司宣布采用 60MHz Pentium 处理器的服务器。但是 TricordSystems 公司付总裁 MarkGraver 同时指出,除非同时改善 I/O 的执行效率,否则 Pentium 服务器仍然无法大幅度提高网络系统的传输速度。如果将它屈就于传统的 PC 设计上,所开发出来的系统依旧无法成为企业型网络系统上强韧稳定且具有高效能的服务器。为了提高关键所在的 I/O 速度,人们正在采用新的数据总线设计和外存系统。同时还通过“重复”,“错误更正”和“预测性诊断”一类技术的应用,实现稳定和易管理特性。另一方面,主从环境中所采用的数据库及其它应用范围的服务器,目前已经能够受惠于较快速度的 CPU 及存储系统。位于加州 ParallanComputer 公司的付总裁 DavisFields 认为:

“Pentium 对那些使用 SQL 服务器和 OS/2, Notes and Oracle 的用户所造成的影响, 要比那些使用 Netware 的深远得多”。

在软件方面, 随着 Pentium 的出现, 软件开发环境也将引起极大的变化。Intel 公司也与各编译程序, 操作系统及应用软件的开发公司一起努力, 以确保所有软件都能充分利用 Pentium 处理器所提供的优点, 增强软件在 Intel 处理器上的性能, 不但现有软件无需修改即可在 Pentium 上运行, 而且经过重新编译, 能将进一步利用 Pentium 的优越性, 软件性能更可大幅度提高。目前, 已有超过半打的 32 位机操作系统准备要在以 Pentium 为核心的个人电脑系统上开展竞争。Pentium 销售主任 Curt Nichols 指出: Santa Cruz Operation 公司的 SCO UNIX、Microsoft 的 Windows、NT Next 的 NextStep 以及 IBM 的 OS/2 2.1 等操作系统都可在 Pentium 上获得优化。

尽管目前上市的 Pentium 系统存在这样或那样的问题, 但是, Pentium 的出现, 无疑将个人电脑推上更重要的新地位。在 Pentium 处理器的开发期间, Intel 自始至终一直与软、硬件开发公司一起合作, 以便开发一套完整的系统构件, 包括芯片组, 超高速缓存和时钟驱动器等。Intel 公司也同时宣布了外设部件接口 (Peripheral Component Interface) 芯片组, 使用户能直接连接硬盘、图形终端和其它外设, 从而提高 Pentium 系统的性能。人们正在积极开展下一代 PC 电脑的开发工作。预计, 九三年底到九四年初, 将会看到“正式的”Pentium 系统——为 Pentium 从头设计的机器。新一代的产品将在九四年——出现, 并急速增加, 它将大大提高电脑系统及其应用的科技水平, 这种 Pentium 系统就是人们所期望的 586 系统。

该如何定义“真正的”Pentium 系统目前没有一定的标准, 规格之一可能是完全的 64 位机结构, 然而目前从技术经济的观点都不太可能, 例如, 目前还缺乏 64 位 ISA/EISA 总线的理论基础, 如果某些广泛使用的附加设备如数据机等, 不能利用这种结构, 为此而提高复杂度和成本就大可不必。可能性较高的情形是, Pentium 的标准将结合不同规格的总线, 快速接口, 以及整体性比 486 强的主板。系统将包括图形加速功能, 立体音响以及其它多媒体特色。服务器则可能提高稳定性, 诊断功能和系统管理 (图 1)。

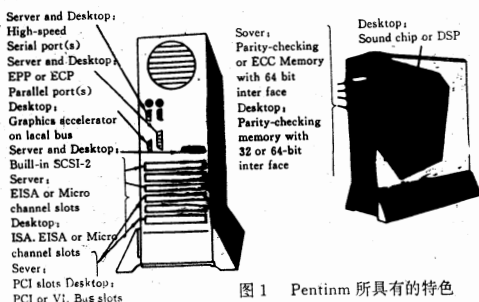


图 1 Pentium 所具有的特色

正如本文指出, Pentium 的问世将促进微电脑体系结构的变化、发展。近年来, 微电脑本身体系结构已经在下述三方面发生了不少变化: 1. 广泛采用高速缓冲存储器 (Cache)。Cache 在单处理机中, 可以填补主存同处理器间的速度差异, 在多处理机中它可以减轻互连网络中信息传递的负担。根据现有 VLSI 技术水平, 发展了多级 Cache 技术。例如, 80486 处理器芯片上有 8K 字节的 Cache, 而在 486 芯片外面加装第二级 Cache, 把容量扩大到 128K 字节; 2. 广泛采用多层次总线结构。高速 CPU 与高速 Cache 的配合, 又要求有高速的总线来传输数据, 这是目前标准的系统总线所远远不能满足的。因此, 现代高档电脑系统都另设内部总线, 这种内部总线有时具有二个层次, 最高速度的内部总线提高高速 CPU 与 Cache 之间的传输通路, 次高速的内部总线为存储管理部件与大容量 DRAM 提供传输通路。另外有两种与外围设备相连的外部总线, 一种较高速的外部总线与图形卡、DMA、局部网络卡以及 SCSI 相连接; 另一种 8 位的较低速的外部总线连接软盘、键盘、串行接口、ISDN 以及音频端口。这种多层次总线结构随着 CPU 芯片速度的提高以及多媒体环境的外围设备多样化而显得更加重要; 3. 多处理器系统结构, 采用高档的通用微处理器构成可变的 (Scalable) 多处理器系统, 一般采用 4 到 8 个微处理器, 使用各种版本的多机 UNIX 操作系统。在上述微电脑体系结构方面的新动向下, Pentium 系统在结构上将有何发展呢?

行家指出: 使得 Pentium 系统有别于上一代电脑产品的特征有二: 其一是 Pentium 系统的存储系统; 其二是局部总线 (Localbus) 和中介总线 (Mezzanine bus) 的采用。

Pentium 芯片的 64 位存储器接口, 目前的操作速度是 60MHz 和 66MHz。在这样的速度下, DRAM 需要通过外部快速存储器控制单元的协助, 才能持续不断地提供程序和数据给处理器。大多数的 Pentium 系统会使用 256KB 或更大的 Cache——可能由同步形态的 VRAM 构成, 以达到 90% 以上的命中率, 有了如此高的命中率之后, 由 Cache 通往主存的路径也许没有必要比 32 位宽, 根据惠普公司的估计, 使用 64 位的 DRAM 后, 所递增的速率可能只有 5%, 其原因是, 只有当快速动作未成功时, 较宽的总线才会显现其作用。不过, 主板制造公司 Micronics 的开发经理 Tom Benoit 则持不同看法, 他认为: “任何采用 32 位存储界面的系统都不是真正的 Pentium 系统”。他表示, Micronics 公司所开发的 Pentium 主板已经采用 64 位的存储界面。因为对该公司而言, 此举并未增加成本, 更何况提高 5% 的效率, 对高级使用者而言, 也是很大的差别。关于 RAM 结构的未来趋势, 厂商都采取观望的态度, 就目前而言, 多级 Cache 设计已能弥补处理器与存储器之间的速度差异, 成本也比较低。但是电脑系统终

究需要较快速的 DRAM,以补偿 Cache 的不足。

Pentium 系统的另一决定要素是局部总线(local bus)或中介总线(Mezzanine bus)的采用。所谓局部总线是指来自处理器的延伸线路,并与处理器同步操作,就像 486 系统所采用的 VESA(VideoElectronicsStandardsAssociation)总线一样。而中介总线在概念上是介于局部总线和 I/O 总线之间,它的线路属于处理器的缓冲区,而它的操作速度则与 CPU 的速度成一定比例(二分之一或三分之一)。Intel 公司正在推动厂商采用它的外设部件互连 PCI(Peripheral Component Interconnect)标准和晶片组,来开发 Pentium 系统的中介总线,有效地解决快速 CPU 和慢速 I/O 之间的不匹配。PCI 目前的规格是定在 32 位,操作速度是 Pentium 外部时钟的二分之一。新的 PCI2.0 版增加了 64 位的定址功能以及 3.3V 的应用,并首次定义连接附加卡的 PCI 接头插座。行家估计,九三年底 PCI 在系统制造厂商之间将会掀起一阵热潮。PCI 的替代方案是已广为 486 开发厂商所采用的 VESA VL-BUS。OPTI 公司推出一组能够同时支援 Pentium 和 VL-Bus 的芯片。可望成为成本比 PCI 低的可行方案,原因在于市面上已经出现许多便宜的 VL-Bus 控制卡。VL-Bus 的规格也是设定在 32 位,但目前尚未定案的 1.1 版中将包含对 64 位接口的支持。当然 VL-Bus 也有不足,由于它是为 40MHz 或以下的操作速度而设计的,当速度超过 50MHz 时,就显得不足。有专家指出“对于局部总线所需要的定址功能,PCI 做得比较好”。然而,对于那些低速的功能像鼠标和数据机使用的串口 I/O 而言,低成本的 ISA 总线就绰绰有余了。所以在扩充卡方面,ISA 提供的选择范围比其它新开发的总线来得广。另一方面,设计师也可以只采用 VL-Bus 或 PCI,再附加新的串口 I/O 供速度较慢的设备使用,而完全不需要采用 ISA 总线。PCI2.0 版的规格也包含一个 ISA 或 PCI 扩充卡共用的插槽。因此,与八十年代中期的 PC 机结构相比较,当前 PC 机设计的趋势是将一般用途的 ISA 总线上的功能逐渐转移到更宽、功能特殊、并且与 CPU 速度同步的总线上。ISA 本身 8 位的界面和 8MHz 的速度已成为所有外设功能的瓶颈。反之,今天的电脑系

统在 CPU 与存储器之间采用的是 32 位或 64 位的通道,而在和外存以及视讯设备等方面采用快速的 32 位或 64 位总线。当然为了配合速度较慢的 I/O 设备和保持与以前产品的兼容性,今天的系统仍然保留着 ISA 总线。

EISA 的情况不一样,Micronics 的 Benait 指出,Pentium 服务器一定会提供 EISA 扩充槽,过去用户曾因价格因素对 EISA 裹足不前,而在 Pentium 系统中,它的价格相对地就便宜多了,但是也有异议,超级服务器制造商 NetFrameSystems 公司的董事长兼技术总裁 CarlAmdahl 在提到 EISA 缺乏扩充性和对位功能时表示,该结构“不适合”身负重任的服务器,另一方面他也提出,NetFrame 公司在他们的系统中,可以无限制地使用具有对位功能,25-MBPS 的 MPSP(MultiprocessorParallelServer Architecture)总线。TricordSystems 的看法与 Amdahl 相同,也采用他们自己设计的总线来改善系统的性能,CPU、存储器以及驱动器以高速的总线互相连接,象网络卡等附加卡则安装在 EISA 结构的扩充槽上。与此类似,Parallan 公司也为其系统的 CPU、存储器,以及 SCSI 驱动器设计一组高速的内部总线,同时提供双重的 MicroChannel 扩充槽,供网络卡,WAN(Wide-area network)连接卡,传真数据卡和磁带机使用(图 2)。

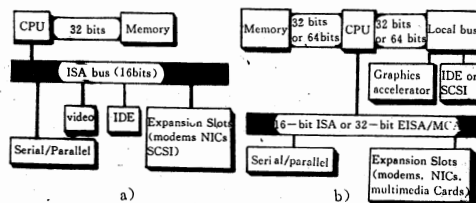


图 2 a): 1986 年间的 PC 机结构, b) 1993 年间的 PC 机结构

综上所述,从 Pentium 的问世,到 Pentium 系统的完善,还有一个过程,在这个过程中充满了激烈的竞争。核心是体系结构上的技术赶上 Pentium 处理器的技术,真正发挥 Pentium 的作用。另一方面就是高性价比。在这个竞争中,我们将会看到对微电脑的科技水平及应用方面都会有很大的促进,受益的将是人类社会。

山东省计算中心简介

山东省计算中心建于一九七六年,是山东省成立最早的计算机应用科学研究机构,隶属于山东省科学院。多年来,山东省计算中心以面向社会、面向四化,为国民经济建设服务为其存在宗旨。

山东省计算中心拥有较强的技术实力,现有高中级技术人员五十余人,初级技术人员五十余人,硕士研究生三十余人,具有一支从事计算机工程、科学计算、计算机网络、计算机软件、无线电通讯、数据库技术、智能模拟、CAD、系统工程、检测维修等专业技术工作的队伍。

山东省计算中心设备资源丰富,配有 IBM4341 大型计算机,CALCOMP 大型绘图仪、TEKTRONIX-618 图形处理系统,并引进 REDUCE、ADINA、SAP5、SAP6 等应用软件,还配有 APOLLO 和 SUN 工作站等各种类型的高档微型计算机;同时还有各种高级语言和丰富的应用软件。

地址:山东省济南市经十路东首科学院路

邮编:250014

电话:6615102-246

计算机中的新技术——Flash Memory

西安交通大学

张 戟

(710049)

摘要 本文从工作机理、性能特点及应用领域等几方面详细介绍 90 年代初在计算机领域中出现的新产物——Flash Memory。

关键词 电擦除 Flash Memory

快速擦写存储器(Flash Memory)是 Intel 公司于 80 年代末 90 年代初推出的一种新型存储器。由于它众多的优点而深受用户的青睐。Flash Memory 的两个主要特点是可以整体电擦除和按字节重新编程,是完全非易失的。由于 Flash Memory 的突出性能,使它问世后获得了迅速的发展。Intel 公司已决定由 Flash Memory 来完全替代 E²PROM,不再生产 E²PROM 产品。虽然 E²PROM 具有可以按字节进行电擦除的特点,但只有很小一部分应用领域有这种特殊要求。下面将从三个方面来介绍 Flash Memory。

1 Flash Memory 的工作机理

Intel 公司的 ETOX-I (EPROM Tunnel Oxide)快速擦写存储器的工艺是由标准的 CMOS EPROM 工艺发展而来的,它的单元结构与 EPROM 的结构一样。两者的主要差别是栅极氧化层厚度不同,EPROM 的氧化层厚度通常为 325 Å,而 ETOX-I 的氧化层较薄,为 100-120 Å,正是这一点使后者具有电擦除功能。

在编程方式下,Flash(以下都这样简称)的工作机理和普通的 EPROM 完全一样。控制栅上施加编程电压 V_{pp} (12V),漏极上施加比 V_{pp} 稍低的电压(7V),源极接地。在漏源极之间的电场作用下,热电子穿越沟道,但在控制栅上的高电压吸引下,这些自由电子越过氧化层进入浮置栅,当浮置栅获得足够的自由电子后,就在源漏极间造成一个导电沟道。

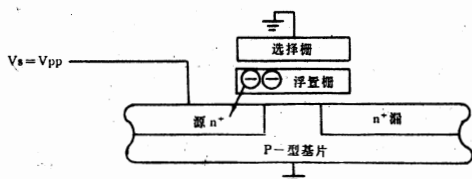


图 1 擦除时的 Flash

Flash 的擦除机理与 EPROM 的机理完全不

同。在 EPROM 中,靠紫外光来中和浮置栅上的电荷以达到擦除目的。对 Flash,高电压 V_{pp} 施加在源极上,控制栅接地,在此电场作用下,浮置栅上的电子就越过氧化层进入源区,被外加电源中和掉,如图 1 所示。

2 Flash Memory 的主要性能特点

(1)高速芯片整体电擦除——芯片整体擦除时间系 1S,而一般 EPROM 需花 15~20min 进行擦除。

(2)高速编程——采用快速脉冲编程算法,对于 28F256 芯片,每个字节的编程花费 100μs,对 28F512、28F010 和 28F020,每个字节的编程仅花 10μs。对于上述 4 种芯片,整个芯片编程时间分别为 4s、1s、2s 和 4s。对于 28F256A 芯片,整个编程时间为 0.5s。

(3)最少 10,000 个擦除/编程周期,通常可达到 100,000 个周期。

(4)采用 $12V \pm 5\%$ 的编程电压。

(5)高速度的存储器访问——最大的读取时间不超过 135ns。

(6)CMOS 的低功耗——最大工作电流为 30mA,备用状态下的最大电流为 100μA。

(7)内部的命令寄存器结构可用于微处理器/微控制器(单片机)兼容的写入接口。

(8)抗噪声特征—— $\pm 10\%$ 的 V_{cc} 允差,通过 Intel 的独特 EPI 工艺过程实现了最高程度的死锁保护,在地址和数据线上承受 100mA 电流、-1V~ $V_{cc}+1V$ 的电压仍能防止死锁。

(9)与 E²PROM 相比,Flash 具有密度大、价格低、可靠性高的明显优势。下表列出了两者的密度和可靠性方面的比较数据:

性能比较 \ 型号	Intel ETox™ Flash	E ² PROM
存储体单元晶体管数	1	2
单元尺寸(1μ 特征尺寸)	15μ ²	38μ ²
循环周期故障	0.1%	5%

Intel 公司 1991 年提供的 DIP 封装的 28F256—200 (200 μ s 的 256k 位的 Flash) 批量报价约为 4.4 美元/片,与 EPROM 相比,价格相当低廉。

3 Flash Memory 的应用领域

20 年来,计算机的发展突飞猛进,CPU 的性能几乎每两年提高一倍。而作为 I/O 的软盘和硬盘驱动器的性能却要 10 年才提高一倍。磁盘至少在 3 个方面严重影响了计算机的发展:

(1) 磁盘驱动器是机电装置,它是计算机中最易磨损、寿命最短的部件。

(2) 由于访问磁盘是低速的操作过程,每字节约需 15ms,大大低于 CPU 访问内存的速度(每字节 80~120ns),因而磁盘操作形成了系统的数据“瓶颈”,是影响系统速度的关键因素。

(3) 磁盘驱动器的功耗大,体积大。不利于计算机的小型化。

多年来,计算机生产商一直在探索取代磁盘的途径,出现了所谓的“固态盘”(SolidStateDisk)。在一些计算机系统中采用的 RAM DISK 就是一种固态盘,它实际上是带电池供电的 SRAM 卡。但这种 SRAM 的密度不高(1 个存储单元含 4~6 只晶体管),采用电池来保存数据也不是十分可靠,且这种卡的体积仍较大,价格也贵,因而很难取代软盘和硬盘。但是随着 Flash 的出现,就使得固态盘真正有可能在很多领域内取代软、硬盘。这种替代是极具革新意义的,也是 Flash 的一个重要应用领域。它完全屏弃了机械装置,具有高达 1 百万小时的平均故障间隔时间,使计算机的寿命大为延长。Flash 的读取时间为 120~250ns,因而消除了机电式磁盘驱动器所造成的数据“瓶颈”,Flash 是高密度的 CMOS 芯片,由它做成的固态盘功耗低、体积小,特别适用于便携式计算机(如笔记本式计算机和掌上型计算机)。Intel 公司推出的一种 4M 字节的 Flash 存储器卡,典型工作电流为 40mA,典型备用电流为 800 μ A,功耗远比磁盘驱动器低;其体积为 54 \times 85.6 \times 3.3mm³,也比最小的软盘驱动器(2.5 英寸盘)的体积小很多。这种固态盘已逐渐形成了一种被国际个人计算机存储器卡协会(PCMCIA)所认可的标准,称之为“PC 卡”。这种卡使用起来很方便,在便携式计算机中已获得广泛应用。目前这种卡的主要缺点是价格比较贵,但这不会成为其推广应用的主要障碍。

除了上述的 PC 卡之外,在 PC 机中采用 Flash 作为其嵌入式(Embedded)的存储器,可以从多方面提高计算机性能。若采用 Flash 来贮存计算机的 BIOS(基本 I/O 系统)代码,可以使计算机厂商灵活

地改变 BIOS,以适应微计算机技术迅速发展的趋势。微机系统厂商实行开放性的政策,更需要有灵活可变的 BIOS。品种繁多的硬件卡和成百上千种流行软件包,使得任何一种 BIOS 都难以兼容所有这些硬件和软件。如果 BIOS 是存贮在 Flash 中的,则软、硬件生产商可以在提供新的硬件卡或软件包时,同时提供一张含有适用的 BIOS 的盘(软盘或固态硬盘),把新的 BIOS 重新装入机内 Flash 存储器中,以运行新的软硬件。

由于 Flash 具有整体电擦除和按字节重新编程的功能。所以 Flash 很适合于数据采集系统。采集的数据可以周期性地由芯片取出,进行分析。擦除后,成为空白芯片,可反复使用至少 10,000 次。也可以用多个芯片构成数据采集的“滚动窗口”。HP 公司的 HP54504A 双源数字存储示波器的采集系统中就大量使用了 Flash 芯片。

对于需周期性地修改被贮存的代码和数据表的应用场合,Flash 是十分理想的器件。因为若采用 EPROM,修改一次代码,需花 15~20 分钟,而用 Flash 只要花 1 秒钟。而且擦除和重新编程可以在同一个系统中或同一个编辑器插座中进行。在组装或装配阶段,为了对硬件进行诊断,常需要根据情况编写各阶段的硬件调试程序。若用 Flash 替代 EPROM,则芯片可直接焊在电路板上,既提高了系统的可靠性,也增加了调试的灵活性。

在系统售后服务中,有时也要修改程序,如用 EPROM 的系统修改就会很繁、有时干脆要把整块电路取下送回厂里返修,这种售后服务费用很昂贵。若采用 Flash 就可以通过串行通信口对电路上的芯片直接进行改写,大大降低了费用支出。

最后我们谈谈采用 Flash 来固化操作系统。

长期以来,操作系统是装到软盘或硬盘中的,很少采用固化的措施,这是因为系统设计者认识到操作系统是在不断更新和改进的,固化反而会带来很多麻烦。凡是用过计算机的用户都会深刻体会到,开机后,等待磁盘驱动器稳定转动起来,然后把操作系统从软盘或者硬盘通过低速的 I/O“瓶颈”装载到系统内存中,是一个相当“漫长”的过程。如果采用 Flash 来固化操作系统,将会显著地缩短开机后用户等待的时间,而同时又不影响方便地更新操作系统。

总之,Flash Memory 是一种具有诱人的应用前景的新型存储器,从单片机应用系统、个人计算机系统到各类计算机系统,都有它的用武之地。

手写体汉字识别的神经网络算法

西安交通大学博士 303 信箱 刘健勤 (710049)

摘要 本文提出了一种用于手写体汉字识别的神经网络算法,该算法可以模拟人类认识过程在特征提取和分类方面的联想记忆特性,可将其非监督/监督学习机制的自适应动力学属性应用于系统的稳定和优化。该算法同现有算法的区别主要体现在两点上:(1)神经网络的联想记忆机制将传统模式识别中的两个独立过程——特征提取与模式分类有机地加以综合;(2)自组织映射(Kohonen Som)网络与 BP 学习机制相综合,以有效地进行汉字识别和极大地提高收敛速度。该算法已应用于手写体汉字识别,并且对并行分布式仿真环境和体系结构来说,它是可并行化和实用的。

关键词 神经网络 模式识别 汉字识别 并行处理

1 引言

近年来,人们提出了许多实现汉字识别的方法和技术,由于汉字在手写体形式中的特点,现有手写体汉字识别系统需要进一步改进以提高其效率,它们缺乏足够的灵活性和并行处理能力。特别是在光照条件差、噪声污染严重的情况下系统就变得不够实用和有效。重新兴起的神经网络和并行计算技术给人们以启示:神经网络可实现传统(精确)逻辑推理所不能完成的模糊语义逻辑推理;相应的并行计算结构可提高信息处理速度以使手写体汉字识别系统成为办公室自动化中的实用工具。为此,本文提出了一种神经网络算法以实现手写体汉字识别的逻辑分析功能。

2 算法介绍

该算法将二值化图像阵列的非监督、非线性特征提取与贝叶斯统计决策的非监督分类、反馈学习机制相综合。在传统方法中,特征提取和目标识别是两个独立的过程,它们之间的联系未得到充分的利用。但从人类认知心理学角度来看,人脑的生物神经网络系统是以集成(integrated)方式工作的,视觉信息输入的特征被视网膜神经元阵列所感知,并被并行分布式结构映射为时变参数集,本文有别于传统算法,它以自组织、自学习和自适应机制模拟人类联想功能和认知映射,相应认知(识别决策)结果由神经网络算法给出,其中自组织分类算法由 Kohonen 网络构成,该分类算法可处理原始图像阵列,并包括以联想记忆、分布式存贮方式实现的特征隐式提取。而 Kohonen 网络的知识学习与权值更新是由多层 BP 网络实现的,这也可认为是 Kohonen(传统)网络的一种改进。该算法的结构框图如图 1 所示。

示。

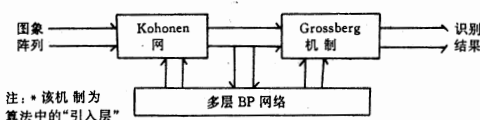


图 1 算法结构框图

象 SOM(Kohonen)这样的自组织神经网络,可自动地通过外界环境进行学习,它是与人类感知过程一致的。该神经网络的非监督学习特性可被认为是来源于其自组织机制而象 BP 这样的神经网络是有教师监督的学习能力很强的自适应系统,它适合于样本学习和特定类识别,该算法是通过 BP 学习算法来增强其非线性能力的。

3 手写体汉字识别算法

该算法的一般化数学分析作者另文论述,在这里,作者要着重强调的是该算法在手写体汉字识别中的应用,并对应用该算法的过程进行了相应分析。

3.1 神经网络算法介绍

根据图 1 所示的网络结构,神经网络算法描述如下:

输入层和输出层

原始像素阵列被用作神经网络的输入,该层的每一个结点被组织成多维向量形式。模式集合被映射成超球面上的点集。识别结果由输出层给出,它具有贝叶斯意义下的最佳值。

中间层

Kohonen 层(结点)

Kohonen 网输出结点 \vec{Y} 的量化反映了输入模式向量 \vec{X} 的匹配程度。

最佳输出 $\vec{Y}_{opt} (\in \{\vec{Y}_i(t) | i=0,1,\dots,N\})$ 应满足输入模式和权向量之间的最优匹配。根据该规则,权向量调整如下:

$$\frac{d}{dt}(W_{ij}(t)) = d(t) * (X_i(t) - W_{ij}(t)), i \in N_c(t)$$

其离散形式为:

$$W_{ij}(t+1) = W_{ij}(t) + \alpha(t) * (X_i(t) - W_{ij}(t)), \\ i \in N_{Ej}(t), 0 \leq \alpha \leq 1$$

其中 $\alpha(t)$ 为增益系数, $N_c(t)$ 和 N_{Ej} 分别表示公式中结点相应的领域。

该网络所应用的 Kohonen 结构得到七十年代以来视觉感知方面生物神经网络研究的实验支持^[1]。

BP(层结点)

在多层 BP 网结构中,映射被用以调整和训练 Kohonen 结点集(权值集):

$$F: R^n \rightarrow R^m \quad \bar{W}_{BP} = F(\bar{Y})$$

其中 \bar{W}_{BP} 为 $\bar{W}_{Kohonen}$ 矩阵的最优逼近。

“引入层”

Grossberg 学习原理引导 Kohonen 结点对最终识别的决策过程, Kohonen 结点的最终权值调整是由“引入层”的宏结点完成的:

$$W_{Kohonen}^{new} = W_{Kohonen}^{old} + (d * W_{BP}^{old} - e * Y)$$

其中 e, d 分别表示小于 1 的常数, 并随训练时间 t 的增加而减小。

3.2 识别过程

预处理过程包括图像输入、汉字分割、标准化($n \times n$ 阵列, n 由用户选定)和骨架化(Skeletonizing 即二值化过程), 然后表征原始汉字信息的像素阵列被用作该算法的输入。由于每个汉字的点阵尺寸可固定为若干常数, 所以该算法的通常规格为 16×16 、 32×32 等(该规格是由标准化过程所决定的)。识别过程的第一步是对随机得到的初步识别结果进行迭代推理, 直至获得最佳值集合为止。

传统方法是从图像阵列中提取特征向量, 但是通常情况下, 特征向量并不能有效、准确和充分地表达汉字属性, 特别是在手写体汉字识别系统中, 断笔和连笔就是实用系统中需要解决的比较困难的问题。到目前为止, 像句法模式识别这样的基于模型的方法仍不能完善地解决该问题。其根本原因就是传统人工智能技术缺乏容错能力和有关模式信息不完

全情况下的模糊推理功能。而人工神经网络恰恰在该方面具有很强的优势。从本质上讲, Kohonen 结构实现了模式信息的自适应压缩, 并使汉字骨架特征的非形式化表示得到增强, 以并行分布化的网络结点形式记忆其本质属性。同时, BP 网络通过非线性测度空间映射和逼近, 减少了模式分类系统中冗余的信息。由 BP 网和 SOM 网综合而构成的联想记忆表征机制是该文中所提出的新算法与现有算法的一个重要区别。

3.3 系统的硬件支持与软件仿真环境

该神经网络的实现需要具有并行分布式处理的硬件支持, 可用于软件环境仿真的系统体系结构可提供功能强大的仿真外壳(Shell)和并行运算功能, 它是使系统实用化的重要环节。

4 结束语

文中所提出的算法与现有手写体汉字识别方法的区别主要在于: (1) 将传统模式识别中孤立的特征提取与分类识别两个环节综合为基于神经网络的隐式学习与联想记忆机制; (2) 将 Kohonen 网与 BP 网有机地综合, 在 Grossberg 学习规律引导下, 有效地实现了汉字识别并极大地提高了网络的收敛速度。该算法在并行分布式仿真环境下是可并行化和实用的。进一步的研究工作将主要集中在知识自动获取工具和多个知识源综合系统的开发研制上。

5 参考文献

- [1] 焦李成著.《神经网络系统理论》, 西安电子科技大学出版社, 1990 年。
- [2] T. Kohonen《Self-organizing Map, Proc. of IEEE》Vol. 78, No. 9, Sept. 1990. PP1464-1480.
- [3] 刘健勤等, “非监督图像分割的神经网络方法的研究”, 《西安交通大学学报》, 第 27 卷第 3 期, 1993 年 6 月, PP. 91-98.
- [4] Jian-qin Liu, A New Neural Network Model for Perceptual Image Segmentation, Summaries Accepted Communications of International Conference AMSE MSC' 92 (Hefei, China, Oct, 1992), PP. 249.

欢迎订阅《电子天府》

《电子天府》(双月刊), 是一本实用性的中级电子技术杂志, 国内统一刊号: CN51-1190。1994 年将迎来《电子天府》创刊 10 周年。9 年来, 共出版了 57 期, 发行近 500 万册, 深受广大读者欢迎。1994 年全年一次性订阅价 30 元; 1993 年《电子天府》合订本每册邮购价 30 元。可直接汇款, 也可来信索取订单。

《电子天府》1993 年合订本主要内容有: 系统讲解大屏幕彩电(含画王)、组合音响、各种型号的彩电加装红外遥控器、影碟机、最新松下录像机、夏普 B78 录像机、图文传真机、大型电子游戏机、开关电源、有线电视(CATV)、东芝 C4 卫星电视接收机等最新流行的中、高档电子电器的原理、电路、检修技术、实用 IC 数据资料、维修实例及介绍海外电子科技动态, 国内电子四新科技成果。《电子天府》合订本是奉献给电子工程技术人员、维修人员和电子爱好者的一本最新案头工具书。(610017/成都市育婴堂街 20 号《电子天府》发行部)

RAID 的性能评价与比较

华中理工大学计算机系 骆新国 张江陵 (武汉 430074)

摘要 冗余磁盘阵列 RAID (RAID—Redundant Array of Inexpensive Disks), 是利用资源重复的并行性系统。我们认为它的特性可以从可靠性、容量冗余度和时间冗余度三方面来全面刻画。本文给出了表征时间冗余度的参数计算方法, 并用三个综合指标对几种 RAID 进行了分析比较。结果表明, RAID2 是一种综合性能指标较理想的阵列结构。

关键词 RAID 时间冗余度 容量冗余度 可靠性

1 引言

RAID 是由多台磁盘存储器组成的具有容错功能的结构化快速大容量外存贮系统。关于它的特性描述, 过去已引入过诸如同步度、异步度、磁盘利用率(数据磁盘数与总磁盘数之比)、冗余度(指冗余磁盘数与数据磁盘数之比)等概念, 文献^[1]为讨论大型磁盘阵列中的故障容错问题还引入了校正代价与组大小两个指标。它们分别与校验矩阵 H 中的列和行中“1”的个数相对应。校正代价 P 是指对某一磁盘写入时需同时写入的校验磁盘数, 表示对某一数据磁盘的改写带来的冗余写操作数。组大小 G 表示替换故障磁盘时重构或恢复其数据必须访问的磁盘数量。这两个参量一定程度上对冗余度进行了描述。本文为了全面直观地刻画系统特性, 引入了容量冗余度、误码率和时间冗余度三个综合指标。

所谓容量冗余度(Cr)是 RAID 为提高可靠性而引入的冗余磁盘容量与数据磁盘容量之比。当采用同型号磁盘构造 RAID 时, 它等于冗余磁盘数 r 与数据磁盘数 n-r 之比, 即: $Cr = \frac{r}{n-r}$ 。参数 Cr 的大小反映了 RAID 为提高可靠性而付出的容量代价程度, 对系统的建造成本有直接影响。通常以百分数表示。

所谓误码率(Pr)是 RAID 与主机相联的读出数据通道上平均每位数据的错误概率。它反映了系统的数据可用性程度。前一篇文章中我们已作过详尽分析, 当 RAID 设有准确设备故障诊断与及时替换恢复功能, 误码率是反映系统可靠性的根本指标。它与系统中磁盘单元的可靠性参数、检查纠错方案与出差控制方式等有关。这个参数是人们最为关注的可靠性指标。一般要求它远远小于系统中总磁盘容量的倒数。

所谓时间冗余度是指在一定的分割方式(即分

割的数据元长度 L 和每组数据元数 m 一定的情况)下, 正确地执行主机的写入或读出数据请求相对于理想的并行多磁盘存取所付出的额外响应时间与有效响应时间之比。这里的理想并行多磁盘存取是指与主机的界面是一个无冗余磁盘的绝对可靠磁盘阵列。

2 冗余磁盘读写的额外时间开销

假设 RAID 中各磁盘的存取通道并行, n 台磁盘(其中 r 台冗余磁盘, n-r 台数据磁盘)同时开始寻道且同一组内的各存贮信元在各盘上有相同的物理位置(即同磁道号和同扇区号), 如 n 台磁盘参数相同, 那么相对于同步并行多磁盘系统来说, 同步所需的额外时间主要是由各磁盘在相应磁道上的旋转等待时间随机性带来的。我们依概率理论来计算。

设 T_i 为第 i 台旋转等待时间随机变量。为便于度量比较, 以读写一个扇区时间长度为单位。假设每道扇区数为 S, 则 T_i 服从参数为 S 的均匀分布, 其概率密度 $f_i(t)$ 及分布函数 $F_i(t)$ 分别为:

$$f_i(t) = \begin{cases} \frac{1}{S} & 0 \leq t \leq S \\ 0 & \text{其它} \end{cases}$$

$$F_i(t) = \int_0^t f_i(t) dt = \begin{cases} \frac{t}{S} & 0 \leq t \leq S \\ 1 & t > S \end{cases}$$

$$E[T_i] = \int_0^S t f_i(t) dt = 0.5S$$

RAID 中 n 台磁盘独立定位, $T_i (i=1 \sim n)$ 之间相互独立, RAID 的等待时间随机变量 $T_a = \max_{1 \leq i \leq n} T_i$, 其分布函数为:

$$F_a(t) = \prod_{i=1}^n F_i(t) = \begin{cases} \frac{t^n}{S^n} & 0 \leq t \leq S \\ 1 & t > S \end{cases}$$

$$E[T_a] = \int_0^S t f_a(t) dt = \int_0^S \frac{t \cdot dF_a(t)}{dt} dt = \int_0^S \frac{nt^n}{S^n} dt$$

$$= \frac{n}{n+1} S$$

n 台磁盘同步所需的额外时间开销为:

$$\Delta T(n) = E(T_a) - E(T_i)$$

$$= \frac{n}{n+1} S - 0.5S = (0.5 - \frac{1}{n+1}) S \dots\dots(1)$$

同理, $n-r$ 台磁盘同步所需的额外时间开销为:

$$\Delta T(n-r) = (0.5 - \frac{1}{n-r+1}) \cdot s \dots\dots(2)$$

由于引入了冗余结构,每次读写平均所需额外时间开销为:

$$Trw = \Delta T(n) - \Delta T(n-r)$$

$$= (\frac{1}{n-r+1} - \frac{1}{n+1}) \cdot S = \frac{r \cdot S}{(n+1)(n-r+1)} \dots\dots(3)$$

可见 Trw 随着 r 增大而显著增大。

3 容错冗余时间

容错冗余时间 Trf 由两部分组成,即排除设备故障时的冗余时间 $Trf1$ 和纠正错误码引起的冗余时间 $Trf2$ 。下面分别计算。

先计算一次排除故障重构的开销时间 Tel , 设需访问 G (G —组大小) 个磁盘来恢复故障磁盘的数据。而且有 T 磁道数据需恢复, 假定磁盘平均寻道时间为 Q (以读写一扇区的时间为单位)。恢复一磁道数据需要同时读 G 个磁盘并写一次磁盘, 所花时间为:

$$Tel' = (Q + \Delta T(G) + \frac{1}{2} S + S) + (Q + \frac{1}{2} S + S) = 2Q + (\frac{1}{2} - \frac{1}{G+1}) S + 3S = 2Q + (3.5 - \frac{1}{G+1}) S$$

每次需恢复的平均磁道数为 T , 那么排除一次故障所花时间为:

$$Tel = T \cdot Tel' = 2QT + (3.5 - \frac{1}{G+1}) TS$$

再来计算 RAID 出现一故障磁盘的概率。设磁盘的平均无故障时间为 $MTBF$, 平均修复时间为 $MTTR$, 单个磁盘的出故障可能性通常以 $Pf = \frac{MTTR}{MTBF + MTTR}$ 表示。 n 台磁盘中出现一台故障的概率为 $(1 - Pf)(1 - Pf)^{n-1} = nPf(1 - Pf)^{n-1}$

因而, 故障容错的冗余时间为:

$$\begin{aligned} Trf1 &= nPf(1 - Pf)^{n-1} \cdot Tel \\ &= nPf(1 - Pf)^{n-1} \cdot [2QT + (3.5 - \frac{1}{G+1}) TS] \\ &= \frac{n \cdot MTTR \cdot MTBF^{n-1}}{(MTBF + MTTR)^n} [2QT + (3.5 - \frac{1}{G+1}) TS] \end{aligned} \dots\dots(4)$$

与之分析类似, 假设误码以单位出错为主, 即每位错需一次纠错写操作, 可求得纠正误码的冗余时间为:

$$Trf2 = n \cdot L \cdot B(Pe - Pr) \cdot (Q + \frac{1}{2} S + 1) \dots\dots(5)$$

其中, Pe, Pr 分别为磁盘和 RAID 的平均误码率。

L 为分割长度 (扇区数), B 为每扇区长度 (位数)。

由 (4)、(5) 式得

$$\begin{aligned} Trf &= Trf1 + Trf2 = n \cdot L \cdot B(Pe - Pr) (Q + \frac{S}{2} + 1) + \frac{n \cdot MTTR \cdot MTBF^{n-1}}{(MTBF + MTTR)^n} [2QT + (3.5 - \frac{1}{G+1}) TS] \dots\dots(6) \end{aligned}$$

由 (6) 式可知容错冗余时间与磁盘设备的可靠性参数 $Pe, MTTR$ 和 $MTBF$ 有关, 同时与 RAID 的可靠性参数 Pr 有关, 此外与 n 、组大小 G 、平均恢复磁道数 T 、每道扇区数 S 、每扇区位数 B 、分割长度 L 及平均找道时间 Q 有关。

4 时间冗余度 R_t 的计算

由 (1)、(6) 式得, 总的冗余时间开销为:

$$\begin{aligned} Tr &= Trw + Trf = (\frac{1}{n-r+1} - \frac{1}{n+1}) S + n \cdot L \cdot B(Pe - Pr) \cdot (Q + \frac{1}{2} S + 1) + \frac{n \cdot MTTR \cdot MTBF^{n-1}}{(MTBF + MTTR)^n} [2QT + (3.5 - \frac{1}{G+1}) TS] \dots\dots(7) \end{aligned}$$

我们计算在给定分割方式下, 分割成 k ($k = n - r$) 个 L 长度的数据存贮元组存取所需的有效响应时间。根据以上分析时的参数约定, 有效时间应为:

$$\begin{aligned} Te &= Q + (0.5S + \Delta T(n-r)) + L \\ &= Q + \frac{n-r}{n-r+1} \cdot S + L \dots\dots(8) \end{aligned}$$

依据时间冗余度定义有:

$$\begin{aligned} R_t &= Tr/Te = \{ (\frac{1}{n-r+1} - \frac{1}{n+1}) S + n \cdot L \cdot B(Pe - Pr) \cdot (Q + \frac{1}{2} S + 1) + \frac{n \cdot MTTR \cdot MTBF^{n-1}}{(MTBF + MTTR)^n} [2QT + (3.5 - \frac{1}{G+1}) TS] \} \cdot [Q + \frac{n-r}{n-r+1} \cdot S + L]^{-1} \end{aligned}$$

当 $Q = S, L = S/3$ (接近通常情况) 时, 我们有:

$$\begin{aligned} R_t &= [\frac{r}{(n+1)(n-r+1)} + n \cdot B \cdot (Pe - Pr) \cdot (1.5S + 1)/3 + \frac{n \cdot MTTR \cdot MTBF^{n-1}}{(MTBF + MTTR)^n} \cdot (5.5T - \frac{T}{G+1})] \cdot (\frac{n-r}{n-r+1} + 4/3)^{-1} \dots\dots(9) \end{aligned}$$

以 5 台磁盘组成的 RAID5 为例, $n = G = 5, r = 1, B = 512, Pe = 10^{-9}, Pr = 10^{-16}, S = 40, T = 300, MTTR = 1, MTBF = 100000$, 则

$$R_t \approx (5 \times 10^{-2} + 0.5 \times 10^{-4} + 7.5 \times 10^{-2}) / 2.13 \approx 5.86\%$$

计算结果表明, 在 RAID5 中相对于有效响应时

间有 5.86% 的冗余时间开销。这种分析计算对估计系统的频带是很有意义的,同时可以用于比较不同系统的速度特性。

5 分析比较与讨论

目前已提出六种结构 RAID1~6^[2],其中 RAID1 是镜像磁盘结构,它是 Hamming 码 RAID2 的一个特例,RAID3~5 都是带奇偶校验的磁盘阵列,不同的是各磁盘(或磁盘单元)间的同步异步存取和磁盘调度方式不同。在阵列系统规模较大时,以 RAID5 较为实用。RAID6 采用较高可靠性的二维奇偶校验。我们比较一下 RAID2、RAID5 和 RAID6 三种具有代表性的磁盘阵列。

5.1 容量冗余度比较

RAID2 是由 r 台冗余磁盘和 $2^r - r - 1$ 台数据磁盘组成 $n = 2^r - 1$ 阵列系统,其 Rc 为

$$Rc2 = \frac{r}{2^r - r - 1} \approx \frac{\log^n}{n - \log^n} \dots\dots (10)$$

RAID5 是由 1 台冗余磁盘和 $n - 1$ 台数据磁盘组成的,其中 Rc 为:

$$Rc5 = \frac{1}{n+1} \dots\dots (11)$$

RAID6 是由 $2G$ 台冗余磁盘和 G^2 台数据磁盘组成 $n = G^2 + 2G$ 阵列系统,其中 $Rc6$ 为:

$$Rc6 = \frac{2G}{G^2} = \frac{2}{G} \approx \frac{2}{\sqrt{n}} \dots\dots (12)$$

比较(10)、(11)、(12)式可知

RAID5 容量冗余度最小,RAID2 次之,RAID6 较大。

5.2 数据可用性比较

RAID5 是奇偶校验冗余磁盘阵列,能检出 n 台磁盘单位和奇数位出错,而纠错要通过磁盘的 ECC 码出错指示来完成。传统的 ECC 码是 32 位,只能检出一个扇区中 11 位错,最新的智能控制器采用 170 位 ECC 可纠 80 位错^[3]。如某一扇区出错超过上述位数,ECC 码不能检出,因而无法纠错。再则,当检出 3 位或 3 位以上奇数位错时,如果其中两位错出现在不同的磁盘上,如 D1 磁盘的第 i 位和 D2 磁盘的第 j 位出错,它与 D1 磁盘的第 j 位和 D2 磁盘的第 i 位出错的检错结果相同,因而无法纠正出错。可见 RAID5 只能纠正 n 个扇区(每台磁盘上一个扇区)中的单位出错和某些奇数位错或某一磁盘中的少于或等于 8 位出错。

RAID6 是将磁盘排成矩阵形式,行和列各设置奇偶校验磁盘。通过校验信息可纠正任何单错,对于在一行或一列中的双错依靠 ECC 指示正确定位。对于不在同一行或同一列中的任意两个错误,第 1 行第 i 列和第 2 行第 j 列的出错校验指示与第 1 行第

列和第 2 行第 i 列出错校验指示相同,校验码本身无法准确定位,将依赖于磁盘 ECC 码准确定位,这种情况下的纠错能力同样受到 ECC 码的检错能力限制。对于三位出错,当有两位在同一行或同一列时,相应的行或列奇偶校验无法检出。可见它只能纠正任意单位错、某些双位错和不在同一行或同一列中的奇数位错,对某一磁盘的多位错也将受到 ECC 能力的限制。

RAID2 是采用 Hamming 码原理构成,其最小码距为 3,它可以立即纠正一位错或正确检查出两位错来。结合磁盘 ECC 出错指示纠错,其误码率与 RAID6 相当。

由此分析可知,RAID5 误码率较大,RAID2 和 RAID6 都较小。

5.3 时间冗余度比较

从(9)式来看关于误码纠错时间冗余度项,因 Pr 的不同 RAID5 较小,关于故障容错时间冗余度项,RAID6 的 G 最小,因而它的这项值也最小,RAID2 的 G 相对稍大,它的这项值较大,而对于 RAID5,其分组 G 即为 n ,因而它的这项值最大。总的来看,RAID6 时间冗余度最小,RAID2 次之,RAID5 较大。

表 1. RAID 的特性比较

	RAID2	RAID5	RAID6
读码率(Pr)	较小	大	较小
容量冗余度	较小	小	较小
时间冗余度	较小	大	小

将上述分析比较结果列于表 1 中。由此我们可以看出 RAID2 是三项综合指标都较理想的阵列结构。

6 参考文献

- [1] Garth A. Gibson, et al, "Failure Correction Techniques for Large Disk Arrays",《Third International Conference on Architectural Support for Programming Languages and Operating System》April, 1989, p123~132.
 - [2] Randy H. Katz, et al, "Disk System Architectures for High Performance Computing",《Proceedings of the IEEE》Vol. 77, No. 12, Dec. 1989, P1842~1858.
 - [3] 张报昌, "智能磁盘控制器的发展和结构设计的特点",《计算机研究与发展》, Vol. 23, No. 10, 1986.
- * 本研究受国家博士学科点科研基金资助。
* 本研究同时受国家自然科学基金重点项目资助。

关于《计算机在数论中的应用》的几个问题

西安六三一所 刘忠(710068)

贵刊1993年第2期上刊载了英国约克大学数学系颜松远先生的文章《计算机在数论中的应用》很有广度和深度,引起了我的兴趣,通过认真阅读和分析,发现几个问题,愿与颜松远先生共同探讨。

在文章中,作者叙述了费尔马小定理:

定理3(费尔马小定理):如果P为素数,且 $\gcd(a, p) = 1$, 则 $a^{p-1} \equiv 1 \pmod{p}$

接着作者测定1387,叙述如下:

$2^{1387} \not\equiv 1 \pmod{1387}$, 因此根据费尔马小定理,这不是一个合数而是一个素数。

首先,根据上下文的主要思想,我怀疑作者原文中应为:“ $2^{1387} \equiv 1 \pmod{1387}$ ”,其中的“ $\not\equiv$ ”应为“ \equiv ”,所以我用C语言编程验证如下:

```
main()
{
    const Con=1387;
    int i=Value=1;
    for (i=1;i<Con;i++)
        if((Value=Value*2)>Con)Value-=Con;
    printf("2 * * %d = %d(mod %d)",Con-1,Value,
```

lue,Con);

}

运行结果如下:

$2 * * 1386 = 1 \pmod{1387}$

其次,费尔马小定理只表明素数具有的一个特征,并非所有的合数均无此特征,只是充分条件而非必要条件,故仅根据费尔马小定理只能判定一个数不是素数而不能判定一个数是素数。这样,文章中与此有关的结论失去了依据。

作者在文章中引出了一个定理:

定理6 如果N为合数,那么其素因子必定都小于或等于根号N。

显然此定理有误,比如14为合数,其素因子为2和7,而 $\sqrt{14} < 7$,不满足结论要求。此定理可修改为:

定理6 如果N为合数,必有素因子小于或等于根号N。

因此,根据此定理得出的算法应增加一个出口。

以上几点看法,愿与颜松远先生及有兴趣者共同切磋。

收稿日期:1993年10月3日

(上接14页)char *argv[];

```
{
    Canvas canvas;
    Panel panel;
    xv-init(XV-INIT-ARGC-PTR-ARGV,
    &argc,argv,NULL);
    frame=(Frame)xv-create(NULL,FRAME,
    FRAME-LABEL,"SAMPLE PROGRAM",
    NULL);
    canvas=(Canvas) xv-create(frame,CAN-
    VAS,CANVAS-REPAINT-PROC,
    canvas-repaint-proc,
    CANVAS-X-PAINT-WINDOW,FALSE,
    CANVAS-AUTO-SHRINK,TRUE,
    CANVAS-AUTO-EXPAND,TRUE,
    CANVAS-WIDTH,640,
    CANVAS-HEIGHT,480,
    (NULL);
```

```
window=canvas-paint-window(canvas);
panel=(panel)xv-create(frame,PANEL,
NULL);
(void)xv-create(panel,PANEL-BUTTON,
PANEL-LABEL-STRING,"Large",
PANEL-NOTIFY-PROC,Large,NULL);
(void)xv-create(panel,PANEL-BUTTON,
PANEL-LABEL-STRING,"Small",
PANEL-NOTIFY-PROC,small,NULL);
(void)xv-create(panel,PANEL-BUTTON,
PANEL-LABEL-STRING,"Quit",
PANEL-NOTIFY-PROC,quit,NULL);
window-fit(panel);
window-fit(canvas);
window-fit(frame);
xv-main-loop(frame);
exit(0);
}
```


实用可靠的 foxbase 应用系统加密工具 foxlock

总后司令部 王晓武 (北京 100842)

foxlock 是一套专门为 foxbase 应用系统而开发的加密工具软件,该工具软件针对 foxbase 应用系统难于实现安全保护的缺陷,吸收各类运行程序加密技术的优点,并结合 foxbase 应用系统的特点,实现了 .fox 伪编译文件和 .dbf 数据文件的加密保护。被加密的伪编译文件和数据文件不仅能够防止 unfox 对 .fox 文件的反编译和对 .dbf 文件的非法访问,而且还能防止应用系统软盘和安装在硬盘上的应用系统被非法复制。

foxlock 较好的解决了 foxbase 应用系统安全保护的难题,该系统已被很多软件开发和销售单位采用,例如国家药典委员会出版的中国药典软件系统版、北京市药品检验所推广的中国药品检验数据库管理系统等,均使用了 foxlock,其安全保护效果受到数据库用户和销售单位的好评。

1 foxlock 的组成

foxlock 采用 MS-C 语言编程,可在任何档次的 PC 机上使用,它由五个独立的程序模块组成:

- foxfmt.exe — 应用系统软盘加密程序
- lockfile.exe — .fox 和 .dbf 文件加密程序
- install.exe — 应用系统安装程序
- uninst.exe — 应用系统回收程序
- run.com — 应用系统运行程序

上述五个程序模块的功能详见图 A、B、C、D、E 流程图所示。

2 foxlock 的功能

上述五个独立的程序模块组成一套完整的 foxlock 工具,各个独立的模块分别实现不同的功能:foxfmt.exe 程序实现对 foxbase 应用系统软盘

的加密功能,经 foxfmt 程序处理的装有应用系统程序的软盘,能够有效的防止非法复制。

lockfile.exe 程序实现对 .fox 和 .dbf 文件的加密功能,经 lockfile 程序处理的 .fox 伪编译文件,使用任何版本的 unfox 工具也不能反编译成 .prg 文件,因为此时的 .fox 文件已不再是标准的伪编译文件;经 lockfile 程序处理的 .dbf 文件,在通常的 foxbase 解释状态下不能访问其结构和数据,此时的 .dbf 文件也不再是常规的数据库文件。

install.exe 程序实现对被加密应用系统从软盘到硬盘指定子目录的安装,它遵循用户输入的子目录名,自动将应用系统全套文件安装到指定硬盘子目录下,该程序按照以下的步骤安装应用系统:

①检测应用系统安装盘上的密码,密码正确继续安装,否则删除当前目录文件。

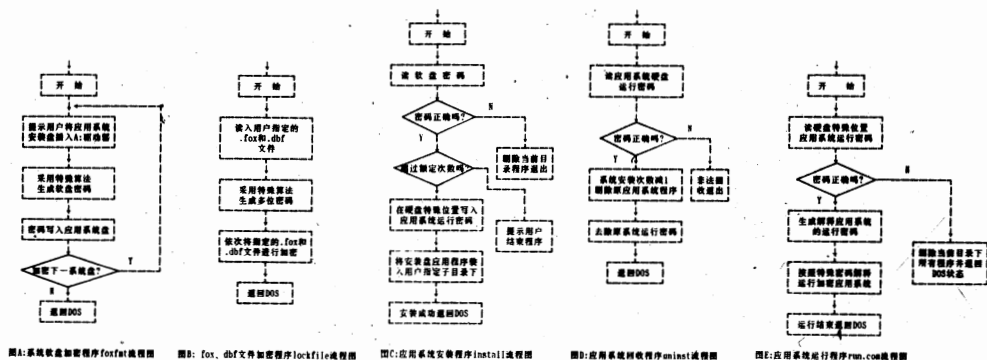
②检查用户安装本系统的次数,若已经超过额定次数,则停止安装。

③将系统运行密码写入硬盘特殊扇区,防止从硬盘上非法复制应用系统。

④将应用系统全套软件复制到用户指定的硬盘子目录下,正常返回 DOS。

uninst.exe 程序实现将应用系统从硬盘回收回到原应用系统软盘的功能。该功能使得用户在额定安装次数内,灵活方便地装卸应用系统。

run.com 程序实现加密后的 foxbase 应用系统的安全运行功能,经上述过程加密处理的 foxbase 应用系统,按照常规的运行方法是不能运行的,run.com 运行程序按照以下步骤运行加密后的 foxbase 应用系统:



在工作站上用C+XWindow进行图形程序设计

四川大学计算机系 袁道华 李炳法 黄金姬 (成都 610064)

摘要 本文概述了XWindow窗口系统的特点及在工作站上用C+XWindow进行图形程序设计的方法,最后给出了一个用XView和XLib的编程实例。

1 引言

近年来,工程工作站以其高性能得到了广泛的使用,C语言以其精巧的设计和卓越的功能成为当今计算机程序设计的首选语言。工作站上的C语言没有提供图形库,使得用户的图形程序设计变得非常困难.XWindow是网络图形工作站和高档微机上的一个理想的分布式、网络透明和独立于设备的多任务窗口和图形系统,使用XWindow系统的图形库,你可以方便地完成所需要的各种画窗和绘图功能。所以C+XWindow系统的联合编程为在工作站上开发用户界面友好的图形程序提供了一个理想的工具。本文以SUN工作站为运行环境进行讨论。

2 XWindow窗口系统

目前窗口已成为一种为人们广泛接受的计算机用户接口方式,如微机上的Microsoft Windows和工作站上的XWindow System(X窗口系统)。由美国MIT、IBM、DEC、AT&T、SUN等联合开发的X窗口系统,以其极方便的用户接口,强大的编程功能和高质量的代码,改变了整个工作站的世界,现正在工作站领域得到普遍采用,并逐步使其成为一个ANSI标准.XWindow是一种事件驱动的窗口系统,提供了在网络上传送图形信息的标准协议。

X窗口系统使用越来越为人熟悉的顾客/服务员模型,顾客和服务员都是软件程序,顾客是应用程序,它要求服务员画窗、文字和其它物体,服务员程序在各个工作站上运行,将所需物体画在显示器上.X窗口系统提供的XLib库含有两百多个过程,其中的许多类似于任何图形库中都有的过程,例如作图过程有:XDRAWLINE(画线),XDRAWRECTANGLE(画矩形),XDRAWARC(画弧)等;画窗的过程有:XCREATEWINDOW(生成窗口),XMAPWINDOW(映射窗口)等;另外还含有一些与控制事件、队列、字体处理、键盘、鼠标和颜色有关的过程。

为方便用户编程,在XWindow核心系统(X协议)的基础上,提供了XLib、XView、Xt等应用环境工具。

3 XView开发工具

XView是在XLib基础上开发的一个简便、实用、高效的应用开发工具,用它它可以方便地开发出窗口化的图形用户界面。一般的应用完全可以只用XView来进行编程,若在应用程序中需要作图,则可以结合XLib库中的画图例程来进行编程。

XView是一种面向对象的窗口化编程工具,程序中可以用XV—create()语句来建立各种XView对象,其中包括:框架(FRAME)、面板(PANEL)、油布(CANVAS)、文本窗口(TEXTSW)、服务员图像(SERVER—IMAGE)、菜单(MENU)、服务员(SERVER)、滚动条(SCROLLBAR)等,这些对象的建立满足一定的层次关系。

(1) 框架

框架是XView程序中的一种基本对象,每个应用程序都对应一个基本框架(主窗口)。在基本框架上可以建立子框架以及面板、油布等子窗口。

(2) 面板

面板是XView程序中的一类重要对象,它建立在框架上,通过面板可以得到非常好的应用程序界面。在面板上可以安装面板按钮(PANEL—BUTTON)、面板文本项(PANEL—TEXT,通过它可以完成参数输入)、面板消息项PANEL—MESSAGE,包括图像消息)等。

(3) 油布

油布作为XView程序中的一类重要对象,它也是建立在框架上。油布提供XView程序的绘图窗口,在油布上可以进行各种绘图。

(4) 服务员

服务员对象可以用来建立与运行在网络中任何工作站上的X服务员的连接,从而实现XWindow的网络和图形信息传送功能.XV—init()可以用来初始化与某个服务员的连接,XV—Create(),则可以用来创建与多个X服务员的连接。在建立了与远程服务员的连接后,则可以在远程机器上开窗、绘图。

XView编程基本步骤是:

①初始化与服务员程序的连接(用XV—init()命令);

②创建应用程序的基本框架及其它子框架;

③在基本框架上建立面板、油布等子窗口；

④创建其它 XView 对象；

⑤进入该应用的基本框架事件处理主循环(用 XV—main—loop() 命令)。

4 移植微机上的 C 语言程序到工作站上

对于微机上一个用标准 C 写的程序可以直接移植到工作站上(可能会有一些细微的改动)。下面我们主要讨论微机上的一个用 C 语言写的图形程序如何移植到工作站上。

前面我们已经提到,工作站上的 C 语言没有作图能力,所以必须对原来的 C 语言程序作较大的修改,用工作站上的 C 语言和 X 窗口系统来实现原来微机上的 C 语言程序。其中主要任务就是用 X 窗口系统完成原来 C 程序中的绘图功能。为简便起见,我们选用 XView 和 XLib 来进行编程。

在用 XView 编程时,应在油布窗口的重绘过程中进行绘图。下面给出在移植过程中所要作的修改步骤:

(1)包含头文件 XView.h、XLib.h 等;说明全局变量 Display * dpy, Window xwin, GC gc。

(2)将原来程序中的主函数改为一个油布重绘过程 canvas—repaint—proc (canvas, paint—window, xrects); 初始化变量 dpy, xwin, gc;

dpy = (Display *) xv—get (paint—window, XV—DISPLAY);

xwin = (Window) xv—get (paint—window, XV—XID);

gc = DefaultGC (dpy, DefaultScreen (dpy));

(3)建立一个新的 main() 主函数,其中初始化与服务器程序的连接,创建该应用程序的基本框架,在该基本框架上创建一个油布(canvas),指定油布的重绘过程为 canvas—repaint—proc。

(4)将原来微机程序中的绘图命令用 XLib 的作图过程来替换。

此方法适用于对任意复杂的微机 C 语言图形程序的移植。

5 程序实例

这里我们给出一个在 SUN 工作站上用 C 语言、XView、XLib 的编程实例。该程序在基本框架上建立了一个油布子窗口和一个面板子窗口,在面板窗口上安装了三个面板按钮 Large、small 和 Quit。当程序启动时,在油布窗口画出一个圆,每按一次按钮 Large,将此圆放大一定的比例。每按一次按钮 Small,则将此圆缩小一定的比例,当按下按钮 Quit 时,退出该应用程序框架。

下面是该实例的 C 语言源程序:

```
#include <stdio.h>
#include <X11/Xlib.h>
#include <xview/xview.h>
#include <xview/canvas.h>
#include <xview/panel.h>
#include <xview/xv—xrect.h>

int r=10,x=300,y=220;
Frame frame;
Xv—Window window;
Display * dpy;
Window xwin;
GC gc;

void canvas—repaint—proc (canvas, paint—
window, xrects)
Canvas canvas;
Xv—Window paint—window;
Xv—xrectlist * xrects;
{
    dpy = (Display *) xv—get (paint—window,
XV—DISPLAY);
    xwin = (Window) xv—get (paint—window, XV
—XID);
    gc = DefaultGC (dpy, DefaultScreen (dpy));
    XClearWindow (dpy, xwin);
    XDrawArc (dpy, xwin, gc, x—r, y—r, 2 * r, 2 *
r, 0, 360 * 64);
}

void large()
{
    if (r<200){r=r+10;
    canvas—repaint—proc ((Canvas) NULL, win
dow, (Xv—xrectlist *) NULL);}
}

void small()
{
    if(r>10){r=r—10;
    canvas—repaint—proc ((Canvas) NULL, win
dow, (Xv—xrectlist *) NULL);}
}

void quit()
{
    xv—destroy—safe(frame);
}

main(argc, argv)
int argc;
```

(下转 11 页)

VGA卡的高效模块化图形程序设计

华中理工大学 蒋日东 (武汉 430074)

摘要 本文介绍了VGA卡的硬件特性及图形控制器的编程方法,并结合具体编程实例,阐述了模块化图形程序设计方法与风格。最后给出了几个实用的高速图形子程序。

关键词 VGA卡 显示内存 模块化

目前,许多微型计算机都配置有VGA显示系统,并且有多种高级语言支持其卡上的丰富的图形功能。但由于考虑到程序的通用性,许多高级语言的图形函数都要先作一些测试与判断,或者通过BIOS调用来实现其功能。显然,这种方法处理速度慢,功能差,远不能满足高速图形程序设计的需要。

针对高级语言图形函数的不足,本文将介绍如何利用VGA卡的显示硬件,开发一些图形程序设计中常用的高速图形子程序。程序的模块化结构将使图形程序十分简洁,编写与调试亦很方便。

1 VGA卡的图形工作模式

VGA卡的结构复杂,功能强大,既保持对CGA、EGA显示卡的兼容性,又提供了几种更高级的显示模式。在图形模式下,由于视频DAC与颜色查找表的引入,使显示系统具有丰富的彩色色调,可在 $262144(2^{18})$ 种彩色色调中实现256种颜色的全屏显示。其内部各种控制寄存器为图形程序设计提供了丰富的硬件资源。

在图形模式下,屏幕被分割成许多个像素点,显示缓冲区的地址范围为A000:0000~A000:FFFFH。一个像素点与视频缓冲区中一个字节或一个字节的若干位相对应。色彩存储多是采用面向设备的彩色页面法,同时也支持使用压缩像素的仿真模式(如13H模式)。

VGA显示系统中,图形控制器及定序器内都含有丰富的控制寄存器。用户可通过端口操作实现对其编程,内部图形控制器是处理器与显示内存之间的数据通道,对写入显示内存的逻辑数据执行与、或、异或及移位等逻辑操作,利用这些功能可协助硬件简化图形操作。图形控制器具有两个端口地址,一个是索引端口3CEH,用于选择图形控制器中各种控制寄存器;另一个是数据端口3CFH,用于读写所选控制寄存器数据。图形控制器中各控制寄存器的索引号见图1。下面介绍几个重要的控制寄存器:

序号(3CE)	图形控制寄存器(3CF)
00	设置/重置寄存器
01	设置/重置允许寄存器
02	颜色比较寄存器
03	数据移位与功能选择寄存器
04	读出页面选择寄存器
05	模式寄存器
06	混合寄存器
07	颜色忽略寄存器
08	位屏蔽寄存器

图1. 图形控制寄存器

1.1 设置/重置寄存器

用于定义显示存储写入操作时填入显示存储的颜色,其高四位保留,低四位分别对应页面0、页面1、页面2及页面3,该寄存器缺省设置为00H。

1.2 设置/重置允许寄存器

该寄存器将决定是否允许用设置/重置寄存器中的数据填写页面,为1时允许,为0时禁止。禁止的页面将用处理器数据写入,缺省设置为00H。

1.3 数据移位与功能选择寄存器

具有两个独立的功能,即对写数据移位和写入数据进行逻辑运算,其各位定义为:

$D_7 \sim D_5$ 保留(0)

$D_4 \sim D_3$ 功能选择

$D_2 \sim D_0$ 移位计数

第3位和第4位决定写模式时的操作,00时不修改,01时与背景进行AND运算,10则为OR运算,11为XOR运算,缺省设置为00。

1.4 位屏蔽寄存器

用于在读出一修改一写入周期屏蔽某些位使其不作修改。该寄存器某一位为0时表示处理器用处理器锁存器内容写该位,为1时则表示用处理器数据写。因此要执行位屏蔽操作,须先执行读操作装入处理器锁存器,后执行写操作。

2 模块化图形程序设计

为使程序简洁、书写方便、调试容易,可以子函数形式编写各功能子模块。如打开一个文件可编写一个子模FILE*STREAMNO()(见程序一)。通过

对该程序的调用,便可用一条语句完成打开文件,读取数据等操作,如:

```
fread(ASCII—WORD,1,ASCII—NUM * 16,
STREAMNO("ascii.dot","rb"));
```

并且还具有文件不能打开的错误提示及相应出口。对于图形程序,同样可编写一些功能子模块使程序编写简单,调试容易。若利用上述图形控制器的各控制寄存器,还可使图形程序具有非常高的显示速度。作者已经开发了写点、矩形填充、光标设置、汉字与 ASCII 字符快速显示等功能子模块,这些子模块的速度均高于直接使用 TurboC 内部提供的函数所编写的子模块速度,并且使用上更加灵活,如所有子模块均提供写点颜色(有的还包括背景色)、逻辑操作(OR、XOR 及 AND 写方式)的接口。为进一步提高速度,在各子模块结束之后,均未对各控制寄存器复位,复位将在最后关闭图形系统时进行。为保持模块化风格,图形系统设置及关闭均以子模块形式提供(分别见程序二及程序三)。

对于矩形填充、汉字及 ASCII 字符的显示,由于显示信息已存储在数据字节内,利用图形控制器其速度效果将更加明显。因为数据字节可直接送显内存,避免了对数据字节内各位的测试判断及显示。当水平方向的坐标不是 8 的整数倍时,数据字节不能一一对应显示内存,其各位而是分跨在两个显示内存地址上,这时只需要将该字节按内存地址对应的坐标进行一次分解即可。如图 2 所示。

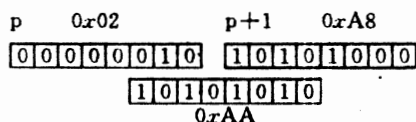


图 2. 数据字节的分解

设 x 除以 8 的余数 $\text{mod} = 6$, 数据字节为 $0xAA$, 则在显示内存地址 p 及 $p+1$ 中的内容应分别为 $0x02$ 、 $0xA8$, 这可通过数据字节分别右移 mod 位、左移 $8 - \text{mod}$ 位而得到。经实测,在字形库加载到内存的情况下,作一汉字与 ASCII 字符显示的动画试验,用上面所介绍的方法比其它两种方法均快数倍以上,具体时间与比值如图 3 所示。

		时间(秒)	比值
本文方法	ASCII	3.74	1.00
	汉字	2.53	1.00
用 point 函数实现	ASCII	17.58	4.70
	汉字	11.10	4.39
用 putpixel 函数实现	ASCII	31.14	8.33
	汉字	19.05	7.53

图 3. 汉字与 ASCII 字符显示速度比较

3 结论

模块化程序设计方法将使图形程序简洁,编写及调试均很方便。利用 VGA 卡的图形控制器,还可使图形程序具有更高的显示速度。根据实际需要可将所开发各功能模块建立一个库,或直接加入到图形库中,我们就可以象调用 C 语言的内部函数一样调用它,并且还节省了子模块的编译时间,应用起来十分方便。

```
#include<graphics.h>
#include<stdlib.h>
#include<stdio.h>
#include<conio.h>
#include<dos.h>
#define CHINESE—NUM 100
#define ASCII—NUM 128
unsigned char CHINESE—WORD[CHINESE
—NUM][32];
unsigned char ASCII—WORD[ASCII—NUM]
[16];
#define INDEXREG 0x3ce
#define VALREG 0x3cf
#define OUTP(index, val) { outportb(IN-
DEXREG, index); \outportb(VALREG, val); }
char far *base=(char far *)0xa0000000;
void pointreset(void);
```

/* 程序一:filename —— 文件名, r—w —— 读写方式 */

```
FILE *STREAMNO(filename, r—w)
char *filename, *r—w;
{FILE *stream; while ((stream = fopen (file-
name, r—w)) == NULL)
{
printf("\n Can't open file %s!\n", filename);
getch();
exit(1);
}
return stream;
}
```

/* 程序二, 设置 VGA 高分辨率模式 */

```
void VGAmodeSET(void)
{
int gdriver=DETECT, gmode, errorcode;
initgraph(&gdriver, &gmode, "c:\\tc\\bgi");
errorcode=graphresult();
if (errorcode != grok)
```

```

{
    printf("Graphics error: %s\n", grapherrormsg
(errorcode));
    printf("Press any key to halt:");
    getch();
    exit(1);
}
}
/* 程序三: 关闭并复位图形系统 */
void VGAmodeCLOSE(void)
{
    pointreset();
    closegraph();
}
/* 程序四: 写点
(x,y)——点坐标, color——颜色,
orxor——逻辑操作 */
void point(int x, int y, int color, int orxor)
{
    unsigned char register mask=0x80;
    char far *p;
    p=base+y*80+x/8;
    mask>>=x%80;
    OUTP(0,color);
    OUTP(1,0xff);
    OUTP(3,orxor);
    OUTP(8,mask);
    *p &= 0xff;
    /* 程序五: 控制寄存器复位 */
    void pointreset(void)
    {
        OUTP(0,0)
        OUTP(1,0)
        OUTP(3,0)
        OUTP(8,0xff);
    }
    /* 程序六: ASCII 字符串显示
string——ASCII 字符串, (x,y)——起始点座标,
color——颜色, orxor——逻辑操作 */
    void asciixy(unsigned char *string, int x, int
y, int color, int orxor)
    {
        register i,j,k,l,m;
        char far *p;
        OUTP(1, 0x0f); OUTP(3, orxor); OUTP(0,
color);
        m=x/8;
        k=x%8;
        while(*string){

```

```

for(i=y,j=0;i<y+16;i++,j++){
    p=base+i*80+m;
    l=ASCII-WORD[*string][j];
    OUTP(8,1>>k); *p&=0xff;
    OUTP(8,1<<(8-k)); *(p+1) &=0xff;

    m++; string++;
}
}
/* 程序七: 汉字显示
string——汉字编码, (x,y)——起始点座标,
dx——字间间隔距离 color——颜色, orxor——
逻辑操作 */
void chinesexy(unsigned char *string, int x,
int y, int dx, int color, int orxor)
{
    register i,j,k,l,m;
    char far *p;
    OUTP(1, 0x0f); OUTP(3, orxor); OUTP(0,
color);
    while(*string){
        for(i=y,j=0;i<y+16;i++,j+=2){
            p=base+i*80+x/8;
            k=x%8;
            l=CHINESE-WORD[*string-1][j];
            OUTP(8,1>>k); *p&=0xff;
            OUTP(8,1<<(8-k)); (*(p+1) &=0xff;
            l=CHINESE-WORD[*string-1][j+1];
            OUTP(8,1>>k); *(p+1)&=0xff;
            OUTP(8,1<<(8-k)); *(p+2)&=0xff;

            x += dx;    string++;
        }
    }
    /* 程序八: 主程序 */
    void main(void)
    {
        int i;
        fread(ASCII-WORD, 1, ASCII-NUM*16,
STREAMNO("ascii.dot", "rb"));
        fread(CHINESE-WORD, 1, CHINESE-
NUM*32, STREAMNO("word.dot", "rb"));
        VGAmodeSET();
        for(i=100; i<300; i++)
            point(i, i, 15, 0x18);
        asciixy("abcd1234EFGH", 120, 120, 14, 0x18);
        chinesexy("\x01\x02\x03\x04", 200, 200, 20,
14, 0x18);
        getch();
        VGAmodeCLOSE();
    }
}

```

DBMS 数据库设计说明书的集成写法

山东省计算中心 傅春英 刘 杰 (济南 250014)

摘要 DBMS 数据库设计说明书的集成写法,是笔者在计算机软件开发实践中,总结出的一种行之有效的写法。这不仅对数据库设计说明书的写法提出了革新,对数据库的设计技术也提出了新的方法和策略。

关键词 集成写法 设计约定 数据库逻辑结构图

1 引言

编制数据库设计说明书是为了描述该软件系统所使用的数据库的设计过程,以便具体地建立、调试、使用、维护这些数据库。

关于数据库设计说明书的写法,不少软件工程方面的书刊都有理论阐述。国家标准局发布的《计算机软件产品开发文件编制指南》GB8567-88 中,对数据库设计说明书作出了编制规范,要求对软件系统所使用的每一数据库写一份数据库设计说明书。这种要求写法的优点不容置疑。而一个 DBMS 所启用的数据库少则数十,多则数百。如果按规范要求去写,总觉得重复太多,篇幅太长,写起来或查阅起来不可忍受。笔者在 DBMS 软件开发实践中试用了“分层”、“集成”写法,使数据库设计说明书可写可读。

所谓“分层”、“集成”写法,就是按照软件划分子系统的方法,把 DBMS 所启用的数据库分为主层和子层。主层和每一子层分别写一份设计说明书。每一设计说明书,将本层启用的所有数据库的数量、名称、状态,数据库间的交换关系及交换方法,调用数据库的程序及状态,以及每一数据库的结构、特性都集成在一起,通过三表一图的形式表示出来。

笔者使用该方法写出的通用帐务处理及大型企业综合生产统计等软件系统数据库设计说明书,深受用户及专家的好评。

2 集成写法建议

本文所建议的集成写法,原则上适用于任一 DBMS 数据库设计说明书。为了说明这一写法的实用性,以下以通用帐务处理软件数据库设计说明书为背景,对数据库设计说明书每一章节的内容及写法进行描述,供大家评价。

第一章 引言

本章的编写内容基本同国标要求相同,其区别在于集成写法是站在本层的角度对其编写目的、编写背景、参考资料,专门术语等进行说明。

第二章 外部设计

2.1 设计约定。本节对本系统所启用的所有数据做出设计规范要求,这是数据库设计的第一个里程碑。

2.1.1 关于数据库名称的设计约定

为便于记忆、维护和存取,应对本层启用的数据库名称设立约定。可按数据库的用途或存贮数据的类别来定义数据库名称。

在通用帐务处理软件数据库名称时约定如下:

- (1) 凡用于凭证处理的数据库名称以 PZ 打头;
- (2) 用于明细帐的数据库名称以 MX 打头;
- (3) 用于总帐的数据库名称以 ZZ 打头;
- (4) 用于汇总的数据库名称以 HZ 打头;
- (5) 用于过程处理的数据库名称以 CX 打头。

以上五种数据库,如果每类启用的数据多于一个时,以 A、B、C……Z26 个英文字母加以区别。

2.1.2 数据库字段名称的约定

同数据库名称的设计约定一样,由于数据处理的需求,数据库字段之间要进行数据交换、数据计算、数据查询、打印输出等处理。为便于记忆、编程、测试、维护,对数据库字段名称也应设立约定。

通用帐务处理软件,要求所有数据库字段名称都以 Z 打头,并以阿拉伯数字为序号加以区分。并且同一意义的字段名称,不管出现在本层的任一数据库,其字段名称都相同。比如:记帐日期字段,无论它是在凭证库,还是在明细帐库或总帐库,其字段名都为 Z1。

2.1.3 关于编写数据库设计说明书的约定

2.1.3.1 关于本设计说明书的层次说明

在此说明本设计说明书在整个软件系统数据库设计说明书中所处的位置。是主层还是子层。如果为子层,说明其子层名称;如果为主层,则在此还应说明本软件数据库设计说明书是怎样分层的,列出各层的名称与状态。如果本软件系统只有一层,在此说明本系统不分层或只有一层即可。

2.1.3.2 数据库设计说明书图表设计约定

集成写法的实现,靠的是三表一图的设计,在此

HZZ. PRG	存放本期(资金占用)	ZW22. PRG	ZW32. PRG
	凭证汇总数据	ZW41. PRG	ZW51. PRG
HZL. PRG	存放本期(资金来源)	ZW22. PRG	ZW32. PRG
	凭证汇总数据	ZW41. PRG	ZW51. PRG
CXA. PRG	临时存放查询、 打印结果	ZW21. PRG	ZW23. PRG
		ZW24. PRG	ZW31. PRG
		ZW33. PRG	ZW34. PRG
		ZW35. PRG	ZW41. PRG
		ZW42. PRG	
CXB. PRG	临时存放查询、 打印结果	ZW41. PRG	ZW42. PRG
		ZW33. PRG	ZW34. PRG
		ZW35. PRG	

附表三:数据库属性及状态对照表

(1)PZKA. DBF(其属性及状态相同的数据库有:PZK-B. DBF,PZKC. DBF,PZKN. DBF)

序号	字段名	状态	类型	长度	小数位	备 注
1	Z1	日期	C	10	0	
2	Z2	编码	C	7		
3	Z3	凭证号	C	3		
4	Z4	摘要	C	14		第一位:A-收
5	Z5	一级科目	C	14		B-付 C-转
6	Z6	二级明细	C	18		
7	Z7	借方金额	N	10	2	第二、三位:序号
8	Z8	贷方金额	N	10	2	
9	Z9	余额	N	13	2	
10	Z10	复核标记	C	1		
11	Z11	凭证数	N	2		
12	ZC	查询标记	C	1		
13	ZI	汇总标记	C	3		
14	Z61	三级明细	C	16		

(2)HZZ. DBF(其属性及状态相同的数据库有,HZL. DBF)

序号	字段名	状态	类型	长度	小数位	备注
1	Z1	日期	C	10		
2	Z2	编码	C	7		
3	Z3	一级科目	C	16		
4	Z4	借方金额	N	13	2	
5	Z5	贷方金额	N	13	2	
6	Z6	余额	N	13	2	
7	ZC	查询标记	C	1		
8	ZI	汇总标记	C	3		
9	Z61	上期余额	N	13	2	

3.2 物理结构设计

3.2.1 说明数据在内存中的安排

3.2.2 说明使用的外部设备及外存空间要求

3.3 运用设计

3.3.1 将数据库设计过程中涉及到的图表、文字说明加以整理,编排,写出目录,建立起数据字典。

3.3.2 说明在数据库设计中,将如何通过区分不同的访问者、不同的访问类型和不同的数据对象,进行分别对待而获得数据安全保密的设计考虑。

3 参考文献

[1]国家标准局 1988 年 1 月发布的《计算机软件产品开发文件编制指南》GB8567-88,p34-35。

[2]General Electric Compang 著,杨绍祺等翻译·《软件工程手册》,科海培训中心出版,1989.8,p198-203。

[3]朱三元等编译·《软件工程指南》,上海翻译出版公司,1985.9,p42-61。

子云电子公司经营产品简介

电子仪器

日立:VC-6023 VC-6024 VC-6025 VC6045 VC-6155 VC-6075 VC-3130 VC-3120
V-1085 V-1065 V-1060 V-222 V-212

泰克:TEK-2465A TEK-2465B TEK-2245A TEK-2245B TEK-2211

固纬:GOS-3310 GOS-622 GOS-623 GOS-625 GOS-643 GOS-6100 GOS-645
GFG-813 GFG-8016

惠普:HP-54501A HP-54502A HP-54503A HP-54504A HP-1651B HP-1654B

建伍:CS-4025 CS-4035 CS-8010 DS-8200

及各类数字万用表、高级智能繁用表、频率计、频谱仪以及其它专用仪器仪表。

OA 办公自动化设备

一体化机:日本理想 RC4500 RC5800 RC5600. 英国 5500 5305 日本理光 VT-2100

复印机佳能:NP-270 NP-1215 NP-2020 NP-3050 NP-3525 NP-3825

施乐:1027S 1027 5026 2520 理光:4065 4085 4418 4470 4490

速印机:首霸 450 超霸 550 堀井 355 365 基士得耶 200 3030 3130 4130

打字机:四通 MS-1300 MS-2401 MS-3401H MS-2406A MS-2406F MS-2411(新型)

桌面办公系统:909 王码 480 巨人 6403 OS-91

计算机:兼容机 286 386SX 386DX COMPAQ386 486 HP 各类机。

碎纸机:905 903 切纸机、装订机、分页器等。

医疗仪器:B超、X机、心电图机、心脏监护仪、胃镜、呼吸机、麻醉机及其它生化分析仪器设备。

制冷设备、各类空调、超低温冰箱。

以上产品价格优惠,函索即寄。

成都市武侯大街 162 号 联系人:徐宇光 电话:5551721

以彩色画面为背景的三层菜单生成程序

甘肃省计算中心 午锁平 (兰州 730030)

摘要 本文利用 windows 下后扩展名以 TIF 图像数据(非压缩型)或者用其他方式录入到计算机的图像数据(如用扫描仪、摄像仪等一系列图像输入工具)用作软件封面背景,再利用汉字放大技术点明软件题目,配上可移动式的活动三层菜单合成一幅既漂亮又直观实用的用户界面,给人清新的感觉。例如制作人物、风景、建筑物、纺织品等不同格调的软件封面。本软件用 Microsoft C6.0 编写(也可以用 TURBO-C、PASCAL 等编写)

关键词 软件封面 图像背景 三层菜单

1 封面背景图像的提取

(1)可直接利用扫描仪扫描图。一般扫描 640×480 个像素(即 VGA 满屏),若是彩色扫描仪扫出 R、G、B 三色图像数据则要在普通微机显示器使用时,可仅取其中一色即可。)

(2)利用 CCD 等摄像提取图像数据,方法同前。(若使用 C540 摄像时所取图像数据仅为 512×512 像元,且头文件为 821 字节。这时在 640×480 屏幕显示器上可将汉字题目写在 510-640 列的空间部分,以弥补屏幕不满之差,另外多出的高可减去,仅显示到 480 行即可。)

(3)可直接取用 windows 下的 TIF 或别的可转为以一个字节为一像素的数据文件。但要弄清楚所取图像的宽和高的像素数。例如 TIF 文件中 63、64 字节是图的宽、第 75、76 是图的高,第 135、136 字节是图的头文件长度。其中:宽(即列)=第二字节×256+第一字节,高和头文件字节数的求法同样。

(4)可利用语言中的绘图命令自己画一幅图形等。

2 图像背景的显示、贮存与恢复

图像文件在第一次恢复出原显示时,一般采用写点方式(即用 setpixel(x,y)函数)但速度较慢。第一次显示后可贮存起来以后,再恢复时速度就提高了。显示、贮存与恢复程序块如下:

(1)显示图像程序

```
.....
if((fp=fopen(namef,"rb"))==NULL)exit(0);
/* 读无此文件时退出系统 */
fseek(fp,headbyte,SEEK-CUR);
/* headbyte 为头文件字节数 */
for(y=1;y<=Row;y++)
/* Row 为行(即图的高) */
{fread(temps,sizeof(char),colx,fp);
/* colx 为行的宽度 */
i=0
for(x=1;x<=colx;x++)
```

```
{xl=temps(i);
color=xl/16
/* VGA 为 16 色 */
setcolor(color);
setpixel(x,y);
i++;
}
fclose(sp);
(2)存贮图像
Void Savescr(char *fname)
{
FILE *fp
int i;
register long j;
char far *base=(char far *)0XA0000000L;
if(access(fname,0)==0) return;
/* 如果同名文件存在,则不存盘 */
fp=fopen(fname,"wb");
for(j=0;j<38400;j++)
/* 38400 即 640×480/8 */
{for(i=0;i<4;i++)
{outp(0x3CE,4);outp(0x3CF,i);
putc(*base,fp);
}
base++;
}
fclose(fp);
outp(0x3CE,0);
}
(3)图像再现
Void Loadscr(*fname)
{
FILE *fp;int i,l=1;
register long j;
char far *base=(char far *)0XA0000000L;
if((fp=fopen(fname,"rb"))==NULL)exit(0);
/* 文件不存在则退出系统 */
for(j=0;j<38400;j++)
l=1;
for(i=0;i<4;i++)
```

```

{outp(0x3C4,2)
outp(0x3C5,1);
*base=getc(fp);
l=l*2;
}
base++;
}
fclose(fp);
outp(0x3C4,0XF0);
}

```

3 三层菜单生成程序的设计思想及实现方法

大量的软件开发工作都相应要使用菜单,而各不同的软件,菜单内容、数量也不同,其工作繁而杂,为此设计一个仅填写参数就可以生成的公用菜单,则为软件开发提供方便。参数文件可用多种手段建立。例如用 FOXBASE、dBASE、WS、EDLIN 等均可。但数据库文件由要求转换为 *TXT 文件。

数据结构为:

```

CH1      Number  2
CH2      Number  2
CH3      Number  2
B-H2     Char   16
B-GN     Char   12

```

其中 ch 为一、二、三层编号;B-H2 为菜单汉字提示字,最长为 8 个汉字;B-GN 为此菜单涉及的功能选择程序块(即程序文件名加“.”和扩展名。)

在菜单生成程序中建四个数组来存放这些变量。

即第一层菜单名 MENU1[1],第二层菜单名 MENU2[1][J]和第三层菜单名 MENU3[J][K]

功能程序名 MENUGN[J][K]

以下是生成程序段:

.....

```

if((fp=fopen("MENUSCR.TXT","rb"))==
=NULL) exit(0);
fscanf(fp,"%s",floor1);
valuel=atoi(floor1);
for(i=0;i<=valuel;i++)
{
fscanf(fp,"%s",floor2[i]);
value2[i]=atoi(floor2[i]);
for(j=0;j<=value2[i];j++)
{
fscanf(fp,"%s",floor3[i][j]);
value3[i][j]=atoi(floor3[i][j]);
}/* 读取菜单个数控制变量 */
for(i=0;i<valuel;i++)
fscanf(fp,"%s",MENU1[i]);
for(i=0;i<=valuel;i++)
for(j=0;j<=value2[i];j++)
{fscan(fp,"%s",MENU2[i][j]);

```

```

for(i=0;i<=value2[i];i++)
for(j=0;j<=value3[i][j];j++)
{fscanf(fp,"%s",MENU3[i][j]);
fscanf(fp,"%s",MENUGN[i][j]);
/* 读取菜单汉字和功能程序名 */
}

```

fclose(sp);

形式的布局格式;

例(4, 4, 4, 6 4, 2, 7 4, 3, 2, 6 5, 3, 2, 4, 5, 3)

其中第一层为 4 个功能所对应所生三个子功能块(有一个退出块),而每个功能块又可对应出各自的孙功能块。即第二层为第一子块为 4 项,第二子块为 4 项,第三子块为 6 项。依次第一子块的 4 项,第一项的子子项为 4 个功能,第二项的子子项为 2 个功能……。

4 菜单的使用

一次在屏幕可显示出第一层全部菜单内容,第二层第一块子菜单,第一子菜单的孙菜单。然而进入选择操作时可以按“→”,“←”箭头来进行第一层菜单选择,而每移动一个箭头将可以拉出一个子、孙菜单,且抹去前一个菜单块并恢复原图屏幕。具体过程可用图的局部存取,以及一 getimage 和一 putimage 函数实现。

菜单的移动子程序块为:

Case RightArrow“→”控制第一层向右移,第二、三层自动移动且擦去前一菜单块并恢复出原图像屏幕。

Case LeftArrow“←”控制第一层向左移,同上。

Case DownArrow“↓”控制第二、三层的向下选择功能。

Case UpArrow“↑”控制第二、三层的向上选择功能。

Case Home“Home”控制第三层回到第二层选择。

Case Cr“ENTER”按一次控制第二层到第三层选择,按第二次进入第三层选择中的功能模块调用。

System(MENUGN[I][J])函数可直接运行功能模块程序(此程序要求为 *EXE 或 *COM 文件)。

5 几点问题

此软件由于菜单的个数是变化的,建立太大太多势必影响内存的占用,调用较大程序时会发生内存不够的问题,其次使用 FOXBASE 或 dBASE 则又要进入其环境,再者可使用 oracle 数据库则可省去存放菜单的内存空间,但要在 oracle 下使用且只能用 Microsoft C 语言,至于更好的办法有待于寻找。

6 参考文献

- [1]《Microsoft C 6.0 技术丛书》,中国科学院希望高级电脑技术公司,1991,12
- [2]午锁平,“为软件配置良好的用户界面”,《计算机应用研究》,NO3. 1993。

美术字和复杂图形的制作及在程序图形界面中的应用技术

西南交通大学 余 滨 (成都 610031)

摘要 本文就怎样利用已有的软件制做精美的图形作了详细的叙述,并给出了在程序中使用这些图形的两种方法。

关键字 图形制作 图形文件格式 程序设计 图形界面

使用过 Windows 的用户对其图形界面一定有很深刻的印象,Windows 的全部界面都是基于图形方式的。用户面对的不再是枯燥的字符命令,取而代之的是令人赏心悦目、易学易用的各式各样的图标。从这一典型的软件,我们可以看到软件的图形界面水平对用户是多么重要。但对于目前大量的 DOS 环境下的编程,没有 WINDOWS 环境提供的各种资源工具,因此图形界面的制做大相径庭,所以每个软件的界面水平有很大的差异。笔者在几年编程工作的基础上,总结出如下一些技术供大家参考。

1 精美图形界面的制做

软件要具有精美的图形界面,首先是要制作这些图形。图形的制作是相当复杂的一个过程。为解决这一问题,许多软件公司已经花了大量人力物力。现在已经有相当多的很成熟的图形编辑软件。但每种软件都有各自的优缺点。笔者通过分析国内较流行的各种图形编辑软件发现:WPS 中的 SPT,鼠标器配的 IMAGE72,WINDOWS V3.X 配的 PAINTBRUSH 这三个几乎每位电脑都有的图形编辑软件如能配合使用,完全可以满足制作各类精美的图形的工作。下面介绍如下:

1.1 优秀的美术字编辑软件——SPT

WPS 金山桌面印刷系统赢得了广大计算机工作者的喜爱,特别是 WPS 中的 SPT(Super Star)图文编辑系统更让人爱不释手。在美术字的编辑方面它可以说是无可比拟的,它有宋体,仿宋体,楷体,黑体四种基本汉字字体及 10 种英文字体,最新版还可以支持十几种矢量字体。有标准、长形、扁形 0 号~7 号 8 种统一字号,也可由用户自己定义大小,还可以选择空心、虚体、旋转、左右倾斜、上下画线等修饰及网点、网格、反视等背景。所以凡是图形中要美术字,在 SPT 中用户可以在屏幕上所见即所得地直接编辑图形中的各种美术字。

SPT 编辑的图形是以独特的 .SPT 或 .TIF 图形文件形式存储的。将 .SPT 文件转换为 PAINTBRUSH 可以处理的 .BMP 形式文件已经解决,见《计算机世界月刊》1992 年第 3 期 20 页,转换后的 .BMP 文件是单色的,但在 WINDOWS 的

PAINTBRUSH 下可以转换成彩色的 .BMP,再经过上色处理即可得到彩色的美术字图形。

1.2 优秀的图形编辑软件——IMAGE72

IMAGE72 是一个非常实用而且功能十分齐全的图形软件,它除具有自由画线、画直线、画圆或椭圆,画方框,橡皮擦,喷墨,区域填充,输入正文,点编辑,查看整个画板,反像,编辑填充图案和调色板,打印,调节线宽等功能外,还有一些非常实用和独特的功能:

● 剪辑(SCISSORS):当用户用鼠标选择这肖像时,屏幕将显示包括所有剪辑功能的弹出菜单,这些功能是 IMAGE72 软件的最令人兴奋之处。用户利用这些功能甚至可以将效果本来并不好的图形处理得令人刮目相看,这些功能是:

* 存储(STORE)——该功能允许用户切割一个小于 IMAGE 板的图像并存储到缓冲区中。

* 复制(COPY)——该功能允许复制一块指定区域的图像。

* 移动(MOVE)——该功能允许用户切割一块指定区域的图像并移动到屏幕的其它位置上。

* 取轮廓(OUTLINE)该功能允许用户对一块指定区域的图像取轮廓。

* 旋转(ROUTE)——该功能允许用户对一块指定区域的图像任意角度旋转。

* 增强(ENHANCE)——该功能允许用户对一块指定区域的图像进行增强处理。

* 减弱(FADE)——该功能允许用户对一块指定区域的图像进行减弱处理。

* 反像(INVERSE)——该功能允许用户对一块指定区域的图像进行反像处理。

* 倾斜(SLANT)——该功能允许用户对一块指定区域的图像进行倾斜处理。

* 取镜像(MIRROR)——该功能允许用户对一块指定区域的图像取镜像。

* 倒置(FLIP)——该功能允许用户将一块指定区域的图像倒置过来。

* 阴影(SHADOW)——该功能允许用户对一块指定区域的图像取阴影。

● 粘接(PASTE):以下面 5 种粘接方式将 *.PUT 或 *.CUT 文件粘接到图像上。

PSET——覆盖;OVERLAP——或方式;

XOR——异或方式;AND——与方式
PRESET——反像覆盖。

IMAGE72 可以读写多种格式的文件。彩色方式下,可读入下列格式的图像文件:PUT、CUT、PCX、PMT、PAT、PAL,用户既可将 IMAGE72 的图形文件转换成上述形式,也可以将上述形式的图形转换为 IMAGE72 可以处理的任何形式。因此 IMAGE72 处理的图形很容易为其它图形编辑软件使用。它与 WINDOWS 的 PAINTBRUSH 的接口可以用 .PCX 形式。

另外,由于 IMAGE72 是专门为 A4SCAN 手握式扫描器而设计的,所以用户可以扫描和处理大小为 A4 的彩色图像或大于 A4 的单色图像。

尤其值得指出的是,IMAGE72 图形软件是 100% 用汇编语言编写的,因此它处理图像快速和高效,是同类软件中速度最快的一种。

从上面的介绍可以看到 IMAGE72 图形软件操作简单、易学,而且功能齐全。与 CoreDraw 等软件相比,虽然功能略弱,但 IMAGE72 非常小,软件的所有执行文件都是 COM 文件,它只占一张 360K 的低密软盘,内存空间也只需 640K。所以各类机器都可以使用,因此用该软件来进行图像编辑是很容易实现的。

1.3 PAINTBRUSH 功能及特点介绍

Windows V3. X 是目前较为流行的软件,其本身是一种基于图形方式的多任务操作环境,关于它的开创性工作已经有大量的文章及著作论述,这里仅介绍它带的一个优秀图形应用软件 PAINTBRUSH,这是一个全屏图形绘制软件,它提供了 18 种绘图工具和 28 种彩色(16 种纯色、12 种组合色),用这些工具和彩色,可以绘制各种色彩丰富的图形。该软件在色彩方面可以说是比较出色的。因此你可以把 SPT、IMAGE72 编辑的图形用 PAINTBRUSH 来进行色彩方面的处理。

同时 PAINTBRUSH 还提供了图形的剪贴、复制、放大、缩小、细画等图形编辑功能。其细画功能很强,可以对全屏幕进行细画处理。在西文字符方面比 IMAGE72 略强。

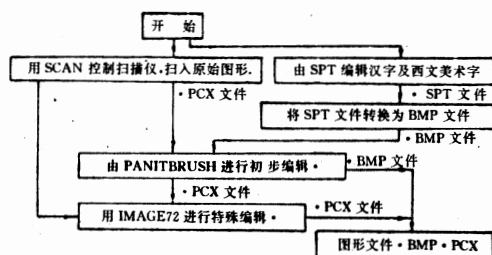
PAINTBRUSH 软件还有一优势是它为 WINDOWS V3. X 的应用程序,能在 WINDOWS V3. X 环境下与其它 WINDOWS 程序一起运行。而 WINDOWS 带有可以控制各类扫描仪或其它图形输入设备的软件,这样就可以在 WINDOWS 环境下将原始图形用这些图形输入进计算机中,再直接用 PAINTBRUSH 编辑该图形。

该软件可以处理 .BMP、.PCX 和 .MSP 文件。可以通过这些接口与 SPT、IMAGE72 交换图形文件。

1.4 综述

以上软件各具特色,但都有缺陷。IMAGE72 和

PAINTBRUSH 不具备 SPT 的汉字美术字的编辑功能,SPT 和 PAINTBRUSH 不具备 IMAGE72 那样强的图形编辑能力。IMAGE72 不具备 PAINTBRUSH 对色彩的控制能力,下面给出一复杂图形制作的流程,就可以充分利用这三个软件各自的特点对同一图形交替编辑,以达到理想的果。



经过上述流程处理,可以充分利用上述三种软件的特点,形成精美的彩色图形,这些图的正确应用将使软件的界面水平得到很大的提高。下面就此问题介绍几种方法。

2 在 C 语言程序中调用图形的技术

要将精美的一幅幅图像用在程序中仍需要解决一些具体问题,你必须了解各类图形文件的格式以及怎样将这些文件显示出来。由于图形文件格式繁杂,处理起来比较复杂。笔者在几年的工作中发现,目前软件一般是在 VGA 640×480 16 色的图形方式下工作,所以这里介绍怎样利用在上述的制做过程中形成的 .BMP 或 .PCX 文件在该显示模式下快速地显示图形。

2.1 整幅图形显示

将复杂的 .PCX 或 .BMP 图形文件转换成简单的 VGA 的四页面映射文件(本文中定义为 PIC 文件),然后在程序中用一个简单的 LOAD 函数显示图形。此方法调用图形的函数简单并且速度快。下面的程序 SAVEPIC EXE 用 BMP 文件形成 PIC 文件(关于 BMP 文件的格式及显示方法已有文章介绍,见参考文献(2))

程序如下:

```

#include <conio.h>
#include <process.h>
#include <stdio.h>
#include <graphics.h>
void save-pic();
main()
{
    int driver, mode, errorcode;
    register int i;
    int result;
    driver = DETECT;
    initgraph(&driver, &mode, "");
    errorcode = graphresult();
    if (errorcode != grOk) /* an error occurred */
    { printf("Graphics error: %s\n", grapherrormsg(errorcode));
      printf("press any key to halt.");
    }
  }

```

```

getch();
exit(1); /* terminate with an error code */
}
/* 显示 BMP 图形文件 */
result = spawnl(p-WAIT, "BMP-DISP EXE", NULL);
if (result == -1)
{ perror("Error from spawnl");
  exit(1);
}
/* 存 VGA 四个页面形成 PIC 文件 */
save-pic("pic.dat");
getch();
return 0;
}

void save-pic(char * fname)
{ FILE * fp;
  int i;
  register long j;
  char far * ptr;
  fp = fopen(fname, "wb"); /* 打开 PIC 文件 */
  if (fp == NULL)
  { printf("error on fopen pic file");
    exit(1);
  }
  for (i = 0; i < 4; i++)
  { outportb(0x3ce, 4); /* 设为读方式 */
    outportb(0x3cf, i); /* 控制页面 */
    ptr = (char far *) 0xa000000L; /* 页面地址 */
    for (j = 0; j < 38500L; j++) /* 存一页 */
    { putc(*ptr, fp);
      ptr++;
    }
    fclose(fp); /* 关闭 PIC 文件 */
    outportb(0x3cf, 0)
  }
}

在程序中调用图形的函数如下:
void load-pic(char * fname)
{ FILE * fp;
  int k = 1;
  register long i, j;
  char far * ptr;
  fp = fopen(fname, "rb");
  for (i = 0; i < 4; i++)
  {
    outportb(0x3c4, 2); /* 设为写方式 */
    outportb(0x3c5, k); /* 设定页面 */
    ptr = (char far *) 0xa000000L; /* 页面地址 */
    for (j = 0; j < 38500L; j++) /* 显示一页 */
    { *ptr = getc(fp);
      ptr++;
    }
    k++;
  }
  fclose(fp); /* 关闭 PIC 文件 */
  outportb(0x3c5, 0xf);
}

```

下面是调图形的示范程序。

```

#include <stdio.h>
#include <graphics.h>
void load-pic(char * fname);
main()
{
  int driver, mode, errorcode;
  register int i;
  driver = DETECT;
  initgraph(&driver, &mode);
  /* read result of initialization */
  errorcode = graphresult();
  if (errorcode != grOk) /* an error occurred */
  {
    printf("Graphics error: %s \n", grapherrormsg(
      errorcode));
    printf("press any key to halt:");
    getch();
    exit(1); /* terminate with an error code */
  }
  load-pic("train.pic");
  getch();
  cleardevice();
  return 0;
}

2.2 部份图形的显示
将复杂的 PCX 或 BMP 图形文件中需要的部份以 TURBO C 中 getimage 函数的格式保存(本文定义为 IMG 文件)然后用 putimage 函数在任一位置显示它。下面为保存部份图形为 IMG 文件的程序。
#include <conio.h>
#include <process.h>
#include <stdio.h>
#include <graphics.h>
void save-img(char * fname, int x1, int y1, int x2, int y2);
main()
{ int driver, mode, errorcode;
  register int i;
  int result;
  driver = DETECT;
  initgraph(&driver, &mode, "")
  errorcode = graphresult();
  if (errorcode != grOk) /* an error occurred */
  {
    printf("Graphics error: %s \n", grapherrormsg(
      errorcode));
    printf("press any key to halt:");
    getch();
    exit(1); /* terminate with an error code */
  }
  /* 显示 BMP 图形文件 */
  result = spawnl(p-WAIT, "BMP-DISP EXE",
    NULL);
  if (result == -1)
  { perror("Error from spawnl");
    exit(1);
  }
  /* 存部份图形为 IMG 文件 */
}

```

```

save_img("img.dat",10,10,100,100);
getch();
return 0;
}

void save_img(char * fname,int x1,int y1,
int x2,int y2)
{
    FILE *fp;
    char far * ptr;
    fp=fopen(fname,"wb");/* 打开 IMG 文件 */
    if (fp==NULL)
        printf("error on fopen pic file");
    exit(1);
}

size=imagesize(x1,y1,x2,y2);
ptr=(char *)malloc(size);
getimage(x1,y1,x2,y2,ptr);
fwrite(ptr,sizeof(char),size,fp);
fclose(fp);
free(ptr);
}

```

在程序中调用图形的函数如下:

```

void load_img(char * fname,int x1,int y1)
{
    FILE * fp;
    char far * ptr;
    long int x,y;
    int x1,x2,y1,y2;
    fp=fopen(fname,"rb");
    /* 计算图形的大小 */
    x1=fgetc(fp);
    x2=fgetc(fp);
    y1=fgetc(fp);
    y2=fgetc(fp);
    x=x2*256+x1;
    y=y2*256+y1;
    size=x*y;
    ptr=(char *)malloc(size);
    fread(ptr,sizeof(char),size,fp);
    fclose(fp);
    putimage(x1,y1,ptr,COPY_PUT);
    free(ptr);
}

```

下面是调图形的示范程序。

```

#include<stdio.h>
#include<graphics.h>

void load_img(char * fname,int x1,int y1);
main()
{
    int driver,mode,errorcode;
    register int i;

    driver=DETECT;
    initgraph(&driver,&mode,"");
    /* read result of initialization */
    errorcode=graphresult();
    if (errorcode!=grOk)/* an error occurred */
    {
        printf("Graphics error: %s\n",grapherrormsg
(errorcode));
        printf("Press any key to halt.");
        getch();
        exit(1);/* terminate with an error code */
    }

    load_img("train.pic",10,10);/* 将图形显示
在(10,10)开始的地方 */
    getch();
    cleardevice();
    return 0;
}

```

上述方法运用得当,可以非常得心应手地编辑各种程序所需的图形,并且在软件中适当地使用它们,便可使你的程序的界面水平得到很大提高。

注:本文中.BMP文件的显示程序BMP-DIS-
P.EXE因篇幅较大,且已经有文章介绍,所以这里
没有给出,读者有意请与笔者联系。

3 参考文献

- [1]“SPT与Paintbrush交换图形数据”,《计算机世界月刊》,1992.3,第20页。
- [2]黎峰,“Microsoft Windows/3.0位图文件格式分析”,《计算机世界月刊》,1992.2,第28页。
- [3]《VGA编程指南》,北京希望电脑公司出版。

(上接12页)①检测硬盘上本系统的运行密码,正确继续运行,否则删除当前目录所有文件。

②按照特定密钥解释运行.fox文件和操作.dbf文件。

3 foxlock 使用方法

foxlock 工具软件配套发行的用户手册详细介

绍了其使用方法,在此不再赘述,需要说明的是,被加密的foxbase应用系统,虽然按照特殊密钥解释运行,但是其运行效率与常规运行方法没有差异,这一点实际运行中能够得到证实。另外,foxlock全套软件采用C语言编制,并采取了严格的反跟踪技术,试图破密者也只能是大海里捞针而已。

谈在汉字系统环境下单字节边框与 汉字同屏显示的实现

内蒙古电子计算机中心 殷霞 (呼和浩特 010010)

摘要 本文谈及了用倒显的技术在汉字系统环境下实现单字节边框(制表符)与汉字的同屏显示,并摘录出窗口系统中的关键程序段,详细说明了这种技术的具体实现。本文可以为广大的界面开发者们提供一点参考。

关键词 窗口 单字节制表符 扩展 ASCII 码

1 概述

目前,越来越多的软件开发人员已意识到是否具有美观、友好的用户面对于一个软件的重要性。因为他们知道上帝们是否喜闻乐见他们的软件成果,用户界面起了多大的作用。因此软件开发者们不惜花费大量的精力设计界面,而其中的关键技术在于如何改善支持用户语言的技术。作为最流行的一种——窗口技术,已被广泛地应用于各种软件中。窗口是计算机显示屏的一部分(或全部),用以显示特定的内容。窗口的显示内容与屏幕的其它部分是分开的。窗口通常有可见的边界。但在窗口系统的开发过程中,却经常会发现在西文状态下很美观的窗口,当运行于某些汉字系统环境中时,原有的边框(制表符)却变成一些奇怪组合了的汉字,如 UC DOS 下原先单线上边界被显示成哪馁馁馁……等。这是由于汉字和制表符的机内码发生冲突引起的。我们知道,通常汉字的机内码被定义成高位均为 1 的两字节,而单字节符其内码是大于 7F 的扩展 ASCII 码,最高位也是 1。这样当两个制表符要相邻显示时,由于许多汉字系统尚不具备自动识别的能力,就可能将这连续最高位均为 1 的两字节看作是汉字的机内码而显示出汉字来,使得汉字和单字节边框不能正确地同屏显示。不少程序员选择了汉字而不得不忍痛放弃边框,但同时也为失去原有的美观效果而惋惜。为了使汉字和边框共存,有些程序员尝试着将单字节边框转换成汉字系统中对应的双字节制表符后再将其显示出来,但这样做比较繁琐,能否使汉字与单字节制表符正确地同屏显示呢?经过多次试验和思考,笔者用较巧妙的方法,实现了这个目的。

2 设计思想

思想很简单,只需要把显示的内容按正常顺序的倒序显示,再把有汉字的地方用正序重新显示一遍即可。

3 实现技术

鉴于篇幅所限,以下仅给出几个关键程序段供读者参考。为便于理解,这些程序段中涉及到的其它程序或函数均以文档形式做出解释。值得一提的是正是用了这些程序段代替原有的相应程序段,使得我们原有的窗口系统在汉字系统中也能很美观地显示。

程序段如下:

```
/* HSWND 系统 窗口结构、函数程序 */
/* —HSWND.C 关键程序段— */
/* 各类边框结构的字符集— */
Struct {
    int nw, ne, se, sw, side, line;
} wcs[] = {
    {218, 191, 217, 192, 179, 196}, {201, 187, 188, 200,
                                     186, 205},
    {214, 183, 189, 211, 186, 196}, {213, 184, 190, 212,
                                     179, 205},
    {194, 194, 217, 192, 179, 196};
};
/* ——在屏幕上显示一个窗口 —— */
void display—window(WINDOW * wnd, char ** strarr)
{
    char ** mp;
    int tx;
    char * s = WTITLE;
    if (verify—wnd(&wnd) && ! VISIBLE){
        VISIBLE = 1;
        vswap(wnd);
        /* * * * * * 重新显示窗口标题 * * * * */
        tx = ((WIDTH - 2 - strlen(s)) / 2);
        while (* s && tx < WIDTH - 1 displ(wnd, tx++, 0, * s +
        +, WTITLEC);
        /* * * * * * 重新显示窗口中有关的字符串 * * * * */
        mp = strarr;
        if (* strarr != NULL){
            WCURS = 0;
            SCROLL = 0;
            while (* mp){
                wprintf(wnd, "%s", * mp++);
                if (* mp) wputchar(wnd, '\n');
```



```

    }
    }
}

/* -- 绘制窗口边框 -- */
static wframe (WINDOW * wnd)
{
    register int xl, yl;
    if (! verify-wnd(&wnd)) return;
    /* ----- 显示窗口标题 ----- */
    displ(wnd, WIDTH-1, 0, NE, WBORDER);
    dtitle(wnd);
    displ(wnd, 0, 0, NW, WBORDER);
    /* ----- 显示窗口边框的两侧边界 ----- */
    for (yl = 1; yl < HEIGHT-1; yl++) {
        displ(wnd, 0, yl, SIDE, WBORDER);
        displ(wnd, WIDTH-1, yl, SIDE, WBORDER);
    }
    /* ----- 显示窗口边框的底边界 ----- */
    displ(wnd, WIDTH-1, yl, SE, WBORDER);
    for (xl = WIDTH-2; xl > 0; xl--)
        displ(wnd, xl, yl, LINE, WBORDER);
    displ(wnd, 0, yl, sw, WBORDER);
    {
        /* ----- 显示窗口标题 ----- */
        static dtitle (WINDOW * wnd)
        {
            int xl, i, ln;
            char * s = WTITLE;
            int llen;
            int titp;
            if (! verify-wnd(&wnd)) return;
            if (s) {
                ln = strlen(s);
                if (ln > WIDTH-2) i = 0;
                else i = ((WIDTH-2-ln)/2);
                if (i > 0) {
                    llen = i;
                    while (i--) displ(wnd, i, 0, LINE, WBORDER);
                }
                titp = llen;
                while (* s && titp < WIDTH-1) displ(wnd, titp +
                    +, 0, * s++, WTITLEC);
                xl = WIDTH-1;
                while (xl > titp) displ(wnd, --xl, 0, LINE, WBOR-
                    DER);
            }
        }
    }
    /* -- 在窗口的特定位置以给定属性显示一字符 -- */
    void displ (WINDOW * wnd, int x, int y, int ch, int at)
    {
        int * vp;

```

```

        int vch = (ch & 255) | (at < 8);
        if ((vp = waddr(wnd, x, y)) != NULL) * vp = vch;
        else vpoke(x + COL, y + ROW, vad(x + COL, y + ROW),
            vch);
    }
    /* -- 从指定位置处获取字符及属性 -- */
    static int dget (WINDOW * wnd, int x, int y)
    {
        int * vp
        if ((vp = waddr(wnd, x, y)) != NULL) return * vp;
        return vpeek(vad(x + COL, y + ROW));
    }
    /* -- 交换视频图像和保留缓冲区中的内容 -- */
    static vswap (WINDOW * wnd)
    {
        int x, y, chat;
        int * bf = (int *) SAV;
        for (y = HEIGHT-1; y >= 0; y--)
            for (x = WIDTH-1; x >= 0; x--) bf++;
        for (y = HEIGHT-1; y >= 0; y--) {
            for (x = WIDTH-1; x >= 0; x--) {
                chat = * (--bf);
                * bf = dget(wnd, x, y);
                displ(wnd, x, y, chat & 255, (chat > 8) & 255);
            }
        }
        /* ----- 有关函数的解释 ----- */
        int * waddr (WINDOW * wnd, int x, int y);
        ★此函数返回指定坐标在对应窗口的位置, 如果窗口不可见, 函数返回根据 x 和 y 坐标计算出来的缓冲区地址。若可见, 则扫描链表, 寻找比被访窗口更近的窗口, 如果有一更近的窗口盖住被访字符位置, 则返回该窗口存储地址的相对地址。若没有更近的窗口盖住被访问字符位置, 则函数返回 NULL 指针, 通知调用者访问视频 RAM。
        vpeek(vad(x + COL, y + ROW));
        vpoke(x + COL, y + ROW, vad(x + COL, y + ROW), vch);
        ★这两个函数实现直接视频读和写。
        int verify-wnd (WINDOW * *);
        ★此函数在链表中搜寻给定的 WINDOW 的结构地址。并返回一真假值反映表中是否存在该 WINDOW 结构。
        void wputchar (WINDOW * wnd, int);
        ★此函数在窗口中当前位置上显示一字符。
        void wprintf (WINDOW * wnd, char * ,
            .....);
        ★此函数从窗口中当前位置开始显示一串字符。

```

计算机访存冲突的对策与实现

湖北省电力中心调度所 罗建明 (武汉 430077)

摘要 计算机系统中,多个内存用户访存之间,内存用户访存与内存刷新之间相互冲突是一个很重要的技术问题。本文以 Classic—7835 计算机为例,论述了计算机访存冲突的产生机理,阐明了解决此问题的技术途径,并对该机内存扩容中出现的问题提出了具体解决方法。

关键词 内存用户 访存冲突 内存刷新

1 前言

随着大规模集成电路和计算技术的发展,存储系统结构的研究和应用也有了很大进展。内存容量从 KB 级发展到 MB 级,甚至还有向更大方向发展的趋势。内存容量的增大,必然给内存的分配、管理带来了许多新问题。例如:把内存分为多少个内存模块较合理,多个内存用户访存之间的相互冲突以及内存用户访存与内存刷新的矛盾如何解决等等。

Classic—7835 计算机一九八二年初作为湖北电网监控系统控制机正式投运。当时,内存容量 512KB。随着电力生产的不断发展,原有容量已不能满足要求。因此,提出把内存扩充到 2MB 的方案。扩容中,由于外商提供资料不全遇到上述问题,使扩容工作一度受挫,现叙述如下。

2 访存冲突的产生

Classic—7835 计算机有 3 个内存用户:CPU 操作数用户, CPU 指令用户, IOP 用户。整个内存分为 8 个内存模块。其逻辑框图示于图 1。内存用户按交叉访问方式工作。交叉访问相关模式如图 2 所示。每个内存用户可访问任一内存模块,任一模块某时刻只能被一个内存用户访问。如果某时刻有多个内存用户要求访存,则出现访存请求冲突。

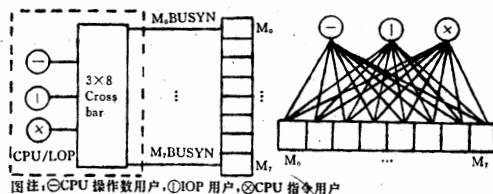


图 1

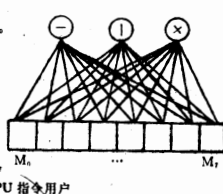


图 2

由于 CPU 按流水线方式工作,每个时钟周期都有可能出现新的内存用户访存请求,而内存模块存储周期要占用一定的时钟周期,因此,当前一内存用

户有某内存模块而此模块处于忙状态时,又有新的内存用户请求访问此模块,则将引起模块冲突。另外,当某模块自身进行刷新时,如果某用户请求访问此模块,也将引起访存冲突。

3 访存请求的处理

访存请求是多用户访存的首要条件。在 Classic—7835 计算机中,由于 3 个内存用户都在一块 CPU/IOP 板上,故仅有一根用户访存请求信号 USRQ_N。其逻辑电路如图 3 所示。

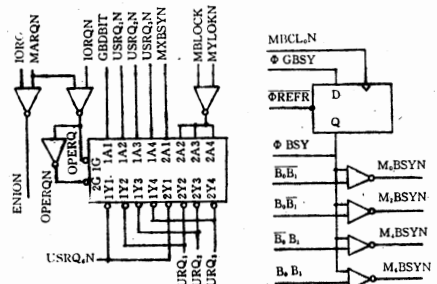


图 3

图 4

在图 3 中,当 MARQ 低时,表示 CPU 操作数用户欲访存进行操作数存取;当 MARQ 高时,表示 CPU 指令用户欲访存取指;当 IORQ 低时,表示 IOP 用户请求进行 I/O 操作。

如果 MARQ 低,则 OPERQ 低,ENION 高。OPERQ 低,选通 2G,1G 高阻态,若请求的模块不忙(MXBSYN 高),使 USRQ_N 低,发用户访存请求。而 ENION 高禁止 IOP 进行 I/O 操作。

如果 MARQ 高,但有 IOP 请求,即 IORQ 低,则 ENION 低,OPERQ 低。ENION 低允许 IOP 访存进行 I/O 操作,OPERQ 低与前一种情况一样,选通 2G,若请求模块不忙(MXBSYN 高),使 USRQ_N 低。这里,操作数用户访存请求与 IOP 用户访请求都是 USRQ_N。这是 CPU/IOP 一个突出特点。应该清楚,虽然操作数用户访存请求和 IOP 用户访存请求都是 USRQ_N,但不会由此引起混

乱。因为即使 MARQN、IORQN 同时为低使 OPERQN 低,而 MARQN 低使 ENION 高,禁止了 IOP 操作。

如果 MARQN 高,而无 IOP 请求,IORQN 也高,则 OPERQ 低,选通了 1G, 2G 为高阻态,由于 | A | 接地,使 | Y | 脚恒为高,即 USRQ₀N 为高,这样,CPU/IOP 实际不发指令用户访存请求。

从以上分析可知,CPU 操作数用户访存请求优先级最高,IOP 用户次之,CPU 指令用户最低。必须清楚,如果没有优先级更高的访存请求,指令用户自然取得访存优先权。另外,扩容前,内存用户访存请求只是用户间访存约束条件,而主存储器对此并无要求,因此,指令用户不发访存请求并不影响它访存。

4 访存模块冲突

扩容后使用的内存板每块含有 0#~3#4 个物理模块,0#、2#物理模块可设定为 0#、2#、4#、6#逻辑模块,1#、3#物理模块可设定为 1#、3#、5#、7#逻辑模块。以 0#物理模块为例说明。

逻辑方程(1)为 0#物理模块“开始忙”逻辑关系表达式。

$$\begin{aligned} \overline{\phi GBSY} = & M_0BSY \cdot \overline{B_0} \cdot \overline{B_1} \cdot \overline{\phi BSY} + M_2BSY \cdot B_0 \cdot \overline{B_1} \cdot \overline{\phi BSY} + M_4BSY \cdot \overline{B_0} \cdot B_1 \cdot \overline{\phi BSY} + \\ & M_6BSY \cdot B_0 \cdot B_1 \cdot \overline{\phi BSY} \end{aligned}$$

B₀、B₁为内存板开关设定值,当 B₀、B₁为 00 时,0#物理模块(ϕBSY)对应 0#逻辑模块(M₀BSY),其余类推; $\phi GBSY$ 为 1 时,说明 0#物理模块开始忙。

由(1)可知:如果模块不忙($\overline{\phi BSY}$ 高),且有某内存用户请求访问此模块(M₀BSY、M₂BSY、M₄BSY、M₆BSY 任一为高),这时 $\phi GBSY$ 为高,0#物理模块从不忙开始忙。图 4 是内存模块忙电路。在图 4 中,若 $\phi GBSY$ 高,总线时钟 MBCL₀N 触发 0#物理模块忙触发器 ϕBSY 高。 ϕBSY 在时钟作用下,产生内存地址、数据及读写命令选通,同时,向内存用户回送相应模块忙回答信号 MXBSYN。

判别及处理访存模块冲突如下:

某内存用户访问某模块时,先发模块忙信号 MXBSYN,将该模块忙触发器置为“1”状态,并保留到该模块存储周期结束。此状态三个内存用户共用。

每个内存用户访存的逻辑模块号分别与对应的物理模块号的模块忙状态进行比较,以确定对该模块的访问是否会发生模块忙冲突。

当内存用户访存遇到模块忙冲突时,该访存请求处于等待状态,直到被访问的模块的存储周期结束,才允许对该模块的访问。

5 内存用户访存与内存刷新冲突

每块内存板有 2K 刷新队列,每个物理模块刷

新量为 512。0#物理模块刷新请求逻辑方程为:

$$\phi REFQ = RFSTART \cdot \phi SV \cdot \phi BSY \cdot \overline{B_0E} \quad (2)$$

其中:RFSTART 刷新队列不为 1; ϕSV 0#物理模块不是正在刷新为 1; ϕBSY 0#物理模块不忙为 1; B_0E 0#物理模块允许为 1;逻辑方程(2)表示:如果刷新队列不为空,且不是正在刷新,物理模块不忙并允许,则提出物理模块刷新请求。

0#物理模块开始刷新逻辑方程:

$$\begin{aligned} \phi REFR = & \phi BSY \cdot \overline{USRQ} \cdot \phi REFQ \cdot M_0BSY \cdot \overline{B_0} \cdot \overline{B_1} \\ & + \phi BSY \cdot \overline{USRQ} \cdot \phi REFQ \cdot M_2BSY \cdot B_0 \cdot \overline{B_1} \\ & + \phi BSY \cdot \overline{USRQ} \cdot \phi REFQ \cdot M_4BSY \cdot \overline{B_0} \cdot B_1 \\ & + \phi BSY \cdot \overline{USRQ} \cdot \phi REFQ \cdot M_6BSY \cdot B_0 \cdot B_1 \quad (3) \end{aligned}$$

这里 B₀、B₁与逻辑方程(1)定义一样。 \overline{USRQ} 为 1 表示没有内存用户访存请求,由逻辑方程(3)可知:如果模块不忙(ϕBSY 为 1),也没有内存用户访问此模块($\overline{M_0BSY}$ 、 $\overline{M_2BSY}$ 、 $\overline{M_4BSY}$ 、 $\overline{M_6BSY}$ 某一项为 1),且没有内存用户访存请求(\overline{USRQ} 为 1),此时,若有模块刷新请求($\phi REFQ$ 为 1),则 $\phi REFR$ 为 1,物理模块开始刷新。且由图 4 可知: $\phi REFR$ 低强置模块忙触发器 ϕBSY 为 1,将相应逻辑模块忙 MXBSY 信号发向内存接口,禁止内存用户访问此模块。

内存用户访存与内存刷新之间的处理策略如下:若有内存用户访存请求,就禁止内存刷新;如果用户访存很忙而使内存刷新队列增加到总刷新量的一半时,内存发总线互锁信号 MBLOCKN,禁止访存而进行内存刷新,直到刷新队列小于总刷新量的一半时,再释放 MBLOCKN 信号。

内存刷新前,先提出刷新请求,若刷新条件满足,则进行内存刷新,刷新时禁止内存用户访存。

综上所述:由于扩容前 CPU/IOP 不发指令用户访存请求,当 CPU 取指时,内存不知道有用户请求访存而进行内存刷新,因此,指令用户被禁止。为了解决这个问题,必须在 CPU/IOP 中增加指令用户访存请求信号 USRQ₀N。

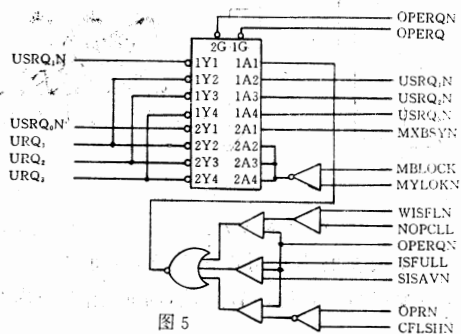


图 5

计算机从 CGA 显示方式升级到 EGA 显示方式

贵州省科委计算中心 明星 杨平 (贵阳 550002)

1 问题的提出

随着 PC 环境的图形在各个领域越来越重要, IBM 最初的彩色图形适配器(CGA)的局限性也越来越明显,许多软件要求在 EGA 或 VGA 显示方式上才能运行。如:中科院希望公司推出的超级组合汉字系统 UCDOS 软件,它要求的显示方式为 EGA、VGA 或 COROL400,在这种环境下方可运行。然而许多 IBM PC/XT 机配置的是 CGA 适配器以及相适应的 CGA 显示器,为了充分地利用原有设备,从经济的角度出发,提出了在原有的 IBM PC/XT 机上将 CGA 适配器换为 EGA 适配器,不换显示器,实现 EGA 显示方式。

问题:

- ①、IBMPC/XT 机上安装 EGA 适配器;
- ②、解决 EGA 适配器输出的信号与显示器同步。

2 解决 IBMPC/XT 机上安装 EGA 适配器

EGA 适配器采用的仍然是 8 位数据线和 20 根地址线的 PC 总线设计,这样它便能直接插入 PC 总线插槽。在 PC 机主板上有一个 DIP 配置开关,其中 5.6 两位是用作选择适配器类型的,将 5.6 两位置为 ON 状态,即无显示器适配器状态,系统开机时即可将 EGA 视频 BIOS 调入系统内。

3 解决 EGA 适配器与 CGA 显示器同步

3.1 EGA 视频模式

模式	类型	分辨率	彩色数
0.1	彩色文本	40×25(320×200, 8×8 字符)(CGA)16	
0.1	彩色文本	40×25(320×350, 8×14 字符)(CGA)16	
2.3	彩色文本	80×25(640×200, 8×8 字符)(CGA)16	
2.3	彩色文本	80×25(640×350, 8×14 字符)(EGA)16	
4.5	彩色图形	320×200(CGA)	4
6	彩色图形	640×200(CGA)	2
7(注 1)	单色文本	80×25(720×350, 8×16 字符)	单色
(注 1:单色适配器兼容)			
0D	彩色图形	320×200(CGA)	16
0E	彩色图形	640×200(CGA)	16
0F	彩色图形	640×350(EGA)	单色
10	彩色图形	640×350(EGA)	16

从模式上看 EGA 模式保持了与 CGA 模式的兼容。

3.2 CRT 控制(6845)寄存器

CRTC(阴极射线管控制器)寄存器构成了 EGA 最大的寄存器。顾名思义这些寄存器控监视器(CRT)的定时及同步化功能这些寄存器复制了大多数的 CGA6845 控制器的功能下表总结出了不同之处:

索引值	CGA	EGA
2h	水平同步位置	水平空白起始
3h	水平同步宽度	水平空白结束
4h	垂直同扫总数	垂直同扫起始
5h	垂直总数调节	垂直同扫结束
6h	垂直显示数	垂直扫描总数
7h	垂直同步	显示位置溢出
8h	隔行模式	预置扫描

3.3 显示器

在水平显示期间,电子束打开和关断的频率(点时钟频率)决定了适配器水平分辨率。增加点时钟的频率,可以提高水平分辨率。显示器还必须要有足够小的磷光点使之能够区分不同的象素,一般说来点间隔越小,显示器质量越好。完成水平扫描后,电子束在水平回扫期向下移到下一行。这一过程持续下去直到最后一行,在最后一行结束处,开始垂直回归。垂直回扫和水平回扫类似。垂直分辨率取决于两个频率——水平频率和垂直频率。水平扫描频率越高,在一个垂直周期内允许显示的水平线越多。同样垂直周期越长(即垂直频率越低),则允许显示的水平线越多。但是,如果垂直频率变得太低,显示将出现闪烁。当频率低于 60Hz 时,大多数人能够感到闪烁,所以,CGA 和 EGA 显示器的场频率都用 60Hz 的垂直。

综上所述得出如下算式:

算式一:水平频率(行频)=点时钟频率/水平总数/字符宽

注:水平总数=水平总数寄存器值+2

字符宽为 8

算式二:场频=行频/一个垂直周期内允许显示线数

CGA 示器的行频为 15.75KHz,场频为 60Hz

EGA 显示器的行为 15.75KHz 和 21.8KHz,场频 60Hz。

3.4 解决 CGA 显示器的行频同步

从 CGA 显示器的内部电路分析得知,适当地调行场振荡电路可以在显示器性能允许范围内,使行频达到 21.8KHz,即改变 RC 振荡电路,调整 R 值使之产生 21.8KHz 的行同步频率。这样便实现了 EGA 适配器输出的行频同步。

然而,由于 CGA 显示器本身设计上的局限,点间的间隔过大,单纯改多行同步频率,还是不能真正地实现 EGA 方式。

3.5 增加点时钟频率减小点间隔:

增加点时钟频率可以提高水平分辨率,然而点时钟频率的增加,从算式一可知水平频率(行频)也要相应地增大。在一行内适当增大水平总数,就可以使水平频率不变,同样也就使得一行内的点数增多,相应得到的结果是点间间隔也就减小。

3.6 修改 CRT 控制器寄存器

为了调整水平总数,必须修改 CRT 控制器寄存器参数值。笔者对各个模式如下的寄存器值作了修改:

索引值	寄存器
0h	水平总数
1h	水平显示结束
2h	水平空白起始
3h	水平空白结束
4h	水平回扫起始
5h	水平回扫结束
6h	垂直总数
10h	垂直回扫起始
11h	垂直回扫结束

由于水平总数的增加,而一行内允许显示的点数为 80 个,所以这就涉及到水平显示结束、水平空白起始、水平空白结束、水平回扫起始和水平回扫结束,这些控制水平显示的寄存器值的修改,达到只允许显示 80 个点。

3.7 修改后的 EGA 适配器向下兼容

在原 EGA 适配器中有七个 CGA 兼容模式,EGA 适配器在这些模式下输出行频为 15.75KHz,而改进的 CGA 显示器它只具备 21.8KHz 这一个行频值,它不能实现双频显示方式,所以为了与这些 CGA 模式兼容,必须修改 EGA 适配器外部寄存器的混合输寄存器和 CRT 控制寄存器,来实现其中的 CGA 兼容模式。

A、外部寄存器的混合输寄存器的修改:

模式	0	1	2	3	4	5	6	D	E
----	---	---	---	---	---	---	---	---	---

(上接 30 页)

CPU/IOP 改线如图 5 所示(修改部份用粗线标出)。在下列四种情况下,不发指令用户访存请求:

- (1)操作数用户禁止指令用户;
- (2)指令堆栈满;
- (3)送指令地址到指令地址寄存器;
- (4)指令堆栈更新。

改线后,Classic-7835 计算机内存扩容成功,运行一直正常。

6 结束语

本文针对 Classic-7835 计算机内存扩容中出现的問題,从分析具体逻辑电路入手,论述了计算机访存冲突及处理这个存储器设计中带有普遍性的关键问题,具有一定的理论价值和实际意义,可供有关

原: 寄存器值 23 23 23 23 23 23 23 23 23

改: 寄存器值 27 27 27 27 27 27 27 27 27

其中:在混合寄存器中

	2—3 位选择时钟:	
00B	14MHz	(来自总线)
	25MHz	(VGA)
01B	16MHz	(来自 EGA)
	28MHz	
	6—7 位选择扫描速率:	
00B	200 行	
01B	350 行	

B、CRT 控制器寄存器的修改:

由算式二得知,一个垂直周期内允许显示线数为 364 线,为了适应 EGA 适配器中 CGA 兼容方式,所以还需要对 CRT 控制器寄存器的如下寄存器的参数值进行修改:

索引值	寄存器
6h	垂直总数
10h	垂直回扫起始
11h	垂直回扫结束

如模式 2、3,它的分辨率为 640×200 ,即原 EGA 适配器一个垂直周期允许内显示线数为 200 线,垂直总数寄存器值为 106h(由算式二得出: $15750/60=262.5$)

为了在修改后的 EGA 适配器上得到模式 2、3,垂直总数寄存器值应改为 16ch(由算式二得出: $21800/60=364$)(注:寄存器的高位在溢出寄存器中)。

综上所述,通过对 CGA 显示器、RC 振荡电路的修改和对 EGA 适配器的 CRT 控制器寄存器和外部寄存器的混合输寄存器参数的修改,实现了 EGA 适配器在原 CGA 显示器上的使用。

4 结束语

采用这种方法用做 IBM PC/XT 升级为兼容 286 机也很有用,只要在升级 286 机上安装一块升级 EGA 卡,即可用这种升级方法实现 EGA 显示方式,使原来的显示器得以充分地利用。现在已制出实验样机。

同志参考。

在这一工作中,曾得到裘旭时同志许多帮助,在此鸣谢。

7 参考文献

- [1]. Modcomp Classic Series Model 7835 Central Processor Unit Technical Manual Part 1~3
- [2]. Modcomp Classic Series Model 3693 Technical Manual
- [3]. Modcomp Classic Series Model 3696 Technical Manual
- [4]. 陈致明·“向量计算机多端存储器若干技术问题的研究”,《计算机研究与发展》VOL. 26,NO. 4, 1989

设计用户界面应注意的问题

广西科学院

罗海鹏

(南宁 530022)

海蓝电脑公司

麦翠荣

黎家珍

(南宁 530022)

参加过许多计算机应用系统的测试、鉴定,碰到了一些程序设计经验还不太多的技术人员设计的程序,它们常常有一些共同的问题,在这里我们把这些问题的指出来,并提出一些建议,供编写应用程序的技术人员参考。

我们这里讲的应用程序是小规模的,因此它们的设计者基本上是个体劳动者,就象欧洲大工业革命之前的手工作坊一样,个人的工艺、技巧充分体现了每个人的风格。因此,我们所提出的问题,在有些人看来,也许不认为是什么问题,这就象公说公有理,婆说婆有理,萝卜、咸菜各有所爱一样。我们下面所提出的看法,确实不应当被看成是一个唯一的尺度。

1 菜单设计

1.1 列出的菜单实际上不允许任意选择。

1.1.1 同一个菜单上的各模块不在同一个层次上

菜单上列出了若干个模块名,但实际上需要在执行了其中的某一个模块或某几个模块后,才能跟着执行其余的模块。即在菜单的总体设计、模块划分上有问题。程序设计者本人常常不容易觉察到这一类的问题,但这一个逻辑上的、原则性的错误。

1.1.2 同一个菜单上的各模块的使用权限不同

列在同一个菜单上的模块,可能有的只能供录入人员使用,有的只能供计划统计人员使用,有的只能供经理查询分析决策,有的只能供系统维护人员使用。如果把这些模块一并列出,任何一个人可以运行任何一个模块,则一旦冲乱了数据,搅乱了程序,就不知找谁来负责了。

有些程序设计人员分清了权限,设计了不同的保密级别,但在菜单上未分开。处在这种情况下被测试时,程序设计人员往往在一旁解释,菜单中的某某模块由于权限不同,在这里是不能使用的。但是,不让使用的模块就不要在菜单中列出来,用不着摆花架子。不同的操作人员应当配备不同的专用盘。

1.1.3 菜单中的某些项没有对应的程序

这可能是由于:

a 程序设计人员模仿、照搬其他人的菜单设计,但别人菜单中的某些功能自己不需要,而忘记修改菜单了;

b. 程序并未全部完成,就匆匆拿出来想通过测试、鉴定;

c 表里不一,菜单是为了表演用的,虚张声势,华而不实。

1.2 同一个软件系统中的各菜单,包装截然不同,象一盘散沙。

1.2.1 几个人合作一个系统,编写不同的菜单,个体劳动,风格各异,凑到一起就很不协调。

1.2.2 部分菜单从别人的程序挪用来,而其他的自己编,因此两者形成了较大的反差。

1.2.3 虎头蛇尾。前面的菜单色彩缤纷、纷繁复呈,越往后越简陋,到最后甚至取消了所有的容错功能、色彩和提示,到这时只有程序设计者本人知道该怎样往下操作了。

在总体设计时,参加程序设计的几个人应充分协商,把菜单的外表形式也大致确定下来。

1.3 进一步谈菜单包装

1.3.1 不同的软件系统不必强求统一的规格,可以有不同的风格,可以是华丽型的,也可以是简洁型的。但同一个软件系统应该有相对统一的规格。

1.3.2 同一级的菜单,应在格式上、色彩上是一样的;不同级的菜单,在格式上、色彩上最好有一定的变化,操作时由于格式、色彩的变化,自然而然地提醒用户进入下一级的菜单了。

●最好不要经常使用大片的红色,这样使用对保护操作者的眼睛不利。

●有一种可供选择的方式是,采用层号的号数来做菜单花边的字符,这样操作者随时都知道自己处于第几级的菜单上。还可以在菜单的某一个角落上注明上一级或上面每一级的菜单名,使程序的来龙去脉更清晰,使操作者随时都知道自己处在什么位置。

1.3.3 屏幕菜单最好居中放置,不要偏左、偏右、偏上、偏下。

1.3.4 操作的提示在屏幕上的位置应相对固定,不要一会儿放在上方,一会儿放在下方。习惯上是

放在屏幕下方,这样容易引起操作者的注意。

1.3.5 有些程序设计者为了强调屏幕上出现的某些内容的重要性,采用了在着重点闪烁的方式。这时要注意,这个闪烁不能太快,否则太刺眼。可以采取很慢地闪烁的方式,既强调了这部分内容的重要性,光线变化又比较柔和。

1.3.6 很多非原版买进,而是非法复制的工具软件,用它们编写应用程序时,在屏幕上常常有不正常的光斑不能消除。

2 初始的操作

2.1 (是与否)的选择按键的一致性

许多人在同一段程序里给出的选择按键,某一次用(Y/N),另一次又用(0/1),再有一次又可能用(Y/任意键)等等。最好能够把选择按键统一起来。

2.2 选择按键的容错性

2.2.1 如果选择按键是(Y/N),则输入了“Y”、“N”以外的任何字符后,程序不应顺着任何一条路向下执行,应该停在当前的菜单上。

2.2.2 按错键时,应给出一个提示按错键的声响,光标停在原来的位置,等待输入正确的字符。

2.2.3 按错键后,有些程序虽然设计了不往下执行,但屏幕菜单重新显示了一遍,这是不必要的。

2.3 菜单操作的提示

2.3.1 菜单序号个数和操作提示数码个数不一致

例如,菜单序号是:1、2、3、4、5、6,而操作提示数码是:1、2、3、4。产生这样的问题的原因是,扩充了功能模块,跟着菜单也扩展了,但是忘记改操作提示。

2.3.2 提示数码的格式不统一

例如,可按键是“0、1、2、3、4、5、6”,那么在一个菜单上用的提示是上面这样,在另一个菜单上用的可能是“1、2、3、4、5、6、0”,还可能是“0...6”,“1...0”、“0、1、...6”,“1、...6、0”,“1、2、...、6、0”等等。应当使这个格式统一起来,不要一会儿是这样的,一会儿又是那样的。选用的表达方式不应当产生模糊的概念,例如“1...0”就不知道到底指哪几个数码。

2.4 减少击键操作次数

程序设计人员在设计(Y/N)选择的操作时,可以设计成按两个键:Y或N,回车;也可以设计成按一键:Y或N。按一个键的执行效率当然高些,但这样做也就失去缓冲的余地了。

3 进一步的操作

3.1 删除不必要的交互回答

3.1.1 有些程序设计人员偏爱频繁的交互问答,人机充分交流,但未考虑每一次的询问是否是绝对必要的。计算机的运行速度是很快的,但人的操作

速度相对是非常慢的,人机的多次交互问答,极大地降低了计算机的运行效率。应大刀阔斧地精简操作步骤。

3.1.2 问过的问题还要问。这是程序设计人员的错误。原因是这个程序设计人员逻辑思维这时产生了混乱,也有时是一个程序改来改去,改出了毛病。

3.2 日期输入

3.2.1 应有日期输入格式的提示

年份是2位数还是4位数?顺序是年、月、日还是日、月、年还是月、日、年?月份是输入2位数还是1位、2位都可以?日期是输入2位数还是1位、2位都可以?对这些问题要给出明确的提示,这样日期输入才不容易出错。

3.2.2 容错功能

日期输入的容错方面要做比较细致地设计。有些防不胜防的错误,例如,月份不小心输入成了13,日期不小心输入成了35,输进了一个2月30日,要求计算机给出6月份到4月份的统计报表等等。所有这些可能的错误,应尽可能地考虑到,给出解决办法。

另一种观点是认为这样的输入错误是不可能全部预计到的,不必要把程序设计的精力过份集中到这里。这些同志认为,实际使用时,这样的录入错误是极少发生的。

3.3 数据录入

3.3.1 有不必要的数据重复录入

由于总体设计上未考虑周全,从不同的模块录入了相同的数据。

3.3.2 适当的数据重复录入是允许的

这些带有重复性质的数据可用于校验录入数据的正确性。

3.3.3 如果可能,应用代码代替中文录入

这可以使许多人望而生畏的中文录入问题得以解决,并可大大提高录入效率。当录入的中文仅属于一个比较小的集合时,建立代码库,并在屏幕上给出操作提示,是可行的、比较有效的。

3.3.4 数据库的数据冗余

录入的数据尽量不要在数据库中的一个以上的地方多次存放。否则增加了存储空间,可能会降低执行效率。若漫不经心地让数据冗余任其泛滥下去,可以使整个数据库变成一个垃圾箱。

在存储容量允许的情况下,经慎重考虑后,也可以有选择地保留一些数据冗余,目的是使应用程序执行的效率更高。

数据库中一般是保留原始数据,不保留计算结果。但使用频率特别高的计算结果,计算费时又较多,如果存储容量允许,也可在数据库中保留。

(下转 39 页)

关于《TVGA256 色图像截取技术》中的 DOS 重入问题及其解决方法

武汉水运工程学院 熊文龙 (430063)

摘要 本文对《TVGA256 色图像截取技术》中的 DOS 重入问题进行了分析,提出了一种解决方法。本方法采用 DOS 数据交换区技术建立了一组解决 DOS 重入的宏程序库,把该程序插入源程序之中,即可解决 DOS 重入问题。该方法具有通用性。

关键词 内存驻留程序 DOS 重入 SDA 屏幕图像 宏指令

1 问题的提出

本刊在 1993 年第 2 期上刊登了《TVGA256 色图像截取技术》一文,提出了直接存储和重显屏幕图像的方法。该方法以 $640 \times 480 \times 256$ 色的 5DH 模式为例,编写了实际运行的程序,解决了 TVGA 系统存储和重显该类图像的问题,有很大的实用价值。

该文中的程序 1 采用 TSR(内存驻留程序)编程技术,在运行中发现经常会出现死锁。并且在有些情况下必然会出现死锁。例如,只要在屏幕的显示过程中,按下了该程序使用的热键“Print Screen”,该程序必然会发生死锁。其他许多情况也会“随机”地出现死锁,大大降低了该程序的使用效果。为此,笔者对该程序进行了分析,发现了该程序中存在的问题主要是在 INT 5H 中断程序中调用了 DOS 功能而未作任何相关检查。笔者对该程序进行了修改,使其能够在任何时候都可以通过热键保存当前的屏幕,然后返回前台程序继续处理。

2 DOS 重入的基础

MS-DOS 是一个单用户操作系统,在执行 DOS 功能调用 INT 21H 代码时是不允许中断它的程序再调用 INT 21H 功能的,这种情况称为 DOS 的不可重入性。由于 MS-DOS 内部只有三个堆栈:I/O 栈、磁盘栈和辅助栈,执行 DOS 功能调用时,就会使用这三个栈。当前台程序处在 INT 21H 中时,启动了 TSR 程序,而在 TSR 程序中又调用了 MS-DOS 功能,就会使前台程序保存在栈中的数据被 TSR 的数据覆盖。

不同的 DOS 功能调用将使用不同的 DOS 内部栈,在 INT 21H 调用中,0CH 以下的功能使用 I/O 栈,0DH 以上的功能调用使用磁盘栈,严重错误中断使用辅助栈[1]。因此,如果前台程序正在使用 0CH 以下的功能,而 TSR 程序调用 0DH 以上的 DOS 功能,就不会有问题。有几个 INT 21H 功能(33H、50H、51H、62H 和 64H)由于非常简单,他们

使用的是用户栈,因此在任何情况下都是可能重入的[1]。

现在我们知道,DOS 不能重入的根本原因是由于在重入过程中破坏了系统堆栈中必须保留的 DOS 关键数据。如果我们采取措施,在进入 DOS 功能调用之前,预先把必要的关键 DOS 数据保留下来,在调用 DOS 功能之后,再恢复原有的 DOS 数据,继续原来的处理,便不会出现问题了。

解决 DOS 重入所必须保存的 DOS 关键数据区称为 DOS 数据交换区(简称 SDA,Swapping Data Area),对于各种 DOS 版本,它在内存中的地址是固定的。有几个未公开的 DOS 功能调用可以得到 SDA 的地址。对于 DOS V3. XX 和 V5. 0 版本,INT 21H 的功能 5D06H 可以返回 SDA 的地址和长度;对于 DOS V4. XX 版本,使用的是 INT 21H 的功能 5D0BH。由于 DOS V4. XX 版不流行,而且该版本不可以使用多个 SDA,与其它版本不兼容,这里不作介绍。下面给出 INT 21H 的 5D 06H 功能调用的返回数据格式:

DS:SI——指向 SDA 的指针

CX——处于 DOS 时(即 INDOS 大于 0),需要交换的数据长度(字节数)

DX——最少(即 INDOS 等于 0 时)需要交换的数据长度(字节数),DOS 数据交换区的长度,不同的 DOS 版本可能不相同;同一 DOS 版本,在不同的 DOS 状态下,需要保留的数据长度也不相同。对于 DOS V3. 30 版本,当 INDOS 标志为 0 时,只有 18H 字节(DOS V5. 0 为 1AH 字节);当 INDOS 标志大于 0 时,有 73CH 字节(DOS V5. 0 为 78CH 字节)。由此可见,目前 SDA 的最大长度不超过 2KB(即 800H)字节。

采用 DOS 数据交换区技术重入 DOS,必须与未公开的 DOS 功能 INT 2AH 配合使用,以确定 DOS 是否处于关键部分。当 DOS 处于关键部分时,不允许交换 SDA,用户程序可以等待 DOS 退出关键部分时,再调用 DOS 功能。INT 2AH 是由 DOS 调用的,用户程序必须截获 INT 2AH 中断,以判断

DOS 是否处于关键部分。DOS 退出关键部分时,要调用 INT 2AH 的功能 81H 或 82H,用户的 INT 2AH 中断程序可据此设置标志[1]。

采用 SDA 技术重入 DOS,除了上述几个功能外,还必须设置自己的磁盘数据传送区 DTA 和程序段前缀 PSP。这可以使用 DOS 中断 INT 21H 的功能 1AH、50H,来设置用户程序的 DTA 和 PSP [1]。

根据实际使用情况发现,除了根据 DOS 是否处于关键区来决定重入 DOS 外,还有两个情况也不能进入 DOS 功能调用,即当中断了 INT 13H 或 INT 10H 而激活 TSR 程序时,不能在该 TSR 程序中调用 DOS 功能。因此,TSR 程序应该截获 INT 13H 和 INT 10H 中断,使其进入时设置标志,退出时取消标志。当激活 TSR 程序时 DOS 正处于关键区,或者激活 TSR 程序时中断了 INT 13H 和/或 INT 10H, TSR 程序不能调用 DOS 功能。这种情况下,常常不能完成 TSR 程序需要的操作,可以退出 TSR 程序,等待时机成熟再启动 TSR 程序完成所需要的操作。

有两种方法可以使 TSR 程序重新激活。一是通过 DOS 的 INT 28H 中断调用,另一是通过 INT 1CH(或者 INT 08H)。

INT 28H 是 DOS 空闲中断,当 DOS 处于等待键盘输入时,不断调用 INT 28H,TSR 程序通过截获 INT 28H 中断来获得控制,实现 DOS 功能调用,完成 TSR 应有的操作。

INT 1CH 是实时时钟中断,它以每秒 $18 \cdot 2$ 次的速度获得控制权。因此,TSR 程序可以通过截获 INT 1CH 中断来检查是否满足激活条件,达到随时激活的目的。

INT 28H 只有在等待键盘输入时才能获得控制权,由它激活 TSR 程序非常可靠,但时间上滞后较大,特别是在屏幕显示连续变化时,INT 28H 得不到控制权,无法保存变化过程中的某个屏幕。而采用 INT 1CH 激活 TSR 程序将能更及时地捕捉到动态屏幕图形,因此,本系统采用 INT 1CH 来重新激活 INT 5H,截取屏幕图像。

上面介绍了解决 DOS 重入所使用的 SDA 技术,也可以采用别的技术。但使用 SDA 技术的程序简洁、通用性好。

3 实现 DOS 重入的方法

采用上述 SDA 技术,开发 TSR 程序时,只要在普通的编程基础上加上三段通用程序,以宏调用的方式嵌入用户程序中,即可以在任何时候实现 DOS 重入。下面以 TSR 程序为例,说明三个程序段的调用方法:

```

. . . . .
NEW_INT_XXH ; TSR 程序使用的中断服务程序(该程序需要调用 DOS)
. . . . .

```

```
SAVE_DOS_SWAP_M ; 保存系统的
```

SDA(第一段程序)

```
SET_DTA MY_DTA ; 如果使用 DOS
磁盘功能调用,设置 DTA
```

```
SET_PSP MY_PSP ; 设置 PSP
```

```
. . . . .
```

```
(任意多个 DOS 功能调用)
```

```
. . . . .
```

```
RESTORE_DOS_SWAP_M ; 恢复系统的
SDA(第二段程序)
```

```
. . . . .
```

```
IRET
```

```
. . . . .
```

```
; 初始化程序部分
```

```
START:
```

```
. . . . .
```

```
INIT_DOS_SWAP_M ; 采用 SDA 技术初始
化程序部分(第三段程序)
```

```
. . . . .
```

```
END START
```

4 《TVGA256 色图像截取技术》中 DOS 重入问题的解决方法

在《TVGA256 色图像截取技术》一文提供的程序 1 中,按上述方法,进行以下修改,就可以实现在任何情况下截取屏幕图像而不至于造成机器死锁。

4.1 在初始化程序段中增加以下三行,取得程序段前缀 PSP 和 DOS 数据交换区参数,如下所示:

```
start:
MOV     AX,ES
MOV     CS:[MY_PSP],AX ; 为 TSR 程序保存 PSP
```

```
INIT_DOS_SWAP_M ; 解决 DOS 重入需要的初始化程序段(宏指令)
```

4.2 在 INT 5H 中断程序的入口处,增加判别重入条件、设置标志及保存 DOS 数据交换区等操作指令,如下所示:

```
stt:
STI
push    ax
CMP     BYTE PTR CS:DOS_KEY_ID,0; 检查 DOS 是否在关键区
```

```
JZ      STT0 ; 不是,继续
```

```
JMP     EXIT_ON_INT5_WANTED_POP_UP ; 是则退出存图操作
```

```
STT0:
```

```
CMP     BYTE PTR CS:IN_INT_13H,0; 检查是否在 INT 13H 中断处理之中
```

```
JZ      STT01 ; 不是则继续
```

```
JMP     EXIT_ON_INT5_WANTED_POP_UP ; 是则退出
```

```
STT01:
```

```

CMP    BYTE PTR CS:IN_INT_10H,0;检查
        是否中断了 INT 10H 操作
JZ     STT1_1;不是则继续
JMP    EXIT_ON_INT5_WANTED_POP_UP;
        是则退出
EXIT_ON_INT5_WANTED_POP_UP:
CMP    BYTE PTR CS:INT_1CH_WANT_POP_UP,0;检查是否 INT 1CH 调用
JZ     NOT_INT_1CH_2;不是
JMP    EXIT_ON_INT5;是则退出
NOT_INT_1CH_2:
INC    CS:INT5_WANTED_POP_UP;应该
        弹出的次数加一
JMP    EXIT_ON_INT5
STT1_1:;满足弹出条件
CMP    CS:IN_INT_05H,0;继续检查是否已
        经处于弹出之中
JZ     STT1_0;不是,继续
JMP    EXIT_ON_INT5;是则退出
STT1_0:
CMP    BYTE PTR CS:INT_1CH_WANT_POP_UP,0;是 INT 1CH 弹出?
JZ     STT1_1;不是
DEC    CS:INT5_WANTED_POP_UP;是则
        弹出次数减一
STT1_1:
MOV    CS:IN_INT_05H,1;设置弹出标志
MOV    CS:INT5_SAVE_AX,AX;设置局部堆栈
        CLI
        MOV    CS:INT5_SAVE_SS,SS
        MOV    CS:INT5_SAVE_SP,SP;
        MOV    AX,CS
        MOV    SS,AX
        MOV    SP,OFFSET INT5_STACK_E
        sti
        MOV    AX,CS:INT5_SAVE_AX
        push    bx;寄存器进栈保护
        push    cx
        push    dx
        push    ds
        push    bp
        push    es
        PUSH    DI
        PUSH    SI
        mov     ax,cs
        mov     ds,ax
        mov     es,ax
        mov     ah,0fh;判断显示方式
int     10h

```

```

MOV    CS:[CUR_MODE],AX;保存当前的屏
        幕方式

```

```

nl:

```

```

INT0_SWAP:

```

```

SAVE_DOS_SWAP_M;保存 DOS 数据交换区

```

```

SET_DTA MY_DTA;设置磁盘传送区 DTA

```

```

SET_PSP MY_PSP;设置程序段前缀 PSP

```

```

lea    dx,fn

```

4.3 在 INT 5H 程序的末尾,增加恢复 DOS 数据交换区的程序段,并设置相应的操作标志,如下所示:

```

mov     ah,3eh;关闭文件
int     21h
OUT_SWAP:
        RESTORE_DOS_SWAP_M;恢复 DOS 数据交换区
        cmp     byte ptr fn,'Z';建立下一个文件名
        jz      al
        inc     byte ptr fn
        jmp     c
al:     inc     byte ptr fn[1]
        mov     byte ptr fn,'A'
c:      MOV     AX,0E07H;响铃,以示成功
        int     10h
cl:
        POP     SI
        POP     DI
        pop     es
        pop     bp
        pop     ds
        pop     dx
        pop     cx
        pop     bx
        CLI
        MOV     SS,CS:INT5_SAVE_SS
        MOV     SP,CS:INT5_SAVE_SP
        sti
EXIT_ON_INT5:
        mov     al,20h
        out     20h,al
        pop     ax
        MOV     BYTE PTR CS:INT_1CH_WANT_POP_UP,0
        MOV     CS:IN_INT_05H,0
        irect
int5     endp

```

5 SDA 技术宏指令库程序清单

以下列出了采用 SDA 技术实现 DOS 重入所必须的全部宏指令。由于篇幅所限,有些子程序没有列出,有兴趣的读者可以直接联系。

***** 第一部分 *****

采用 DOS 数据交换区技术解决 DOS 重入问题的宏程序库 DOS_SWAP.INC SWAP_BUFFER_SIZE=800H;SDA 的最大长度(字节数),重入 DOS 前必须执行的程序(第一段,驻留内存,保存 SDA)

```

SAVE_DOS_SWAP_M MACRO

```



```

LOCAL    WAIT_EXIT_DOS_KEY,NOT_INDOS
        CLI
        MOV    CS,[SAVE_AX],AX
        MOV    CS,[SAVE_SS],SS;保存系统堆栈
        MOV    CS,[SAVE_SP],SP
        MOV    AX,CS;设置用户堆栈
        MOV    SS,AX
        MOV    SP,OFFSET END_STK
        STI
        PUSH    DS
        PUSH    ES
        PUSH    CX
        PUSH    DI
        PUSH    SI
        PUSHF
WAIT_EXIT_DOS_KEY:
        CMP    BYTE PTR CS,[DOS_KEY_ID],0;测试
            DOS 是否处于关键区
        JNZ    WAIT_EXIT_DOS_KEY;是处于关键区,
            则等待
        LDS    SI,DWORD PTR CS,[SWAP_PTR];取
            SDA 的指针
        MOV    CX,CS,[SWAP_SIZE_ALWAYS];取最
            小 SDA 的长度
        CMP    BYTE PTR DS,[SI+1],0;测试 indos 标志
        JZ     NOT_INDOS
        MOV    CX,CS,[SWAP_SIZE_INDOS];取最大
            SDA 长度
NOT_INDOS,MOV    CS:[CUR_SWAP_SIZE],CX
        PUSH    CS
        POP     ES
        CLD
        LEA    DI,SWAP_BUFFER
        REP    MOVSB;保存 SDA
        POPF
        POP    SI
        POP    DI
        POP    CX
        POP    ES
        POP    DS
        MOV    AX,CS,[SAVE_AX]
        ENDM
;完成 DOS 功能调用后必须执行的程序(第二段,驻留内
;存,恢复 SDA)
RESTORE_DOS_SWAP_M MACRO
    LOCAL    WAIT_EXIT_DOS_KEY
    PUSH    DS
    PUSH    ES
    PUSH    CX
    PUSH    DI
    PUSH    SI
    PUSHF
WAIT_EXIT_DOS_KEY:
    CMP    BYTE PTR CS,[DOS_KEY_ID],0;测试
            DOS 是否处于关键区
    JNZ    WAIT_EXIT_DOS_KEY;是处于关键区,
        则等待
    LES    DI,DWORD PTR CS,[SWAP_PTR];取系
        统的 SDA 指针
    MOV    CX,CS,[CUR_SWAP_SIZE];取本次使
        用的 SDA 的长度
    PUSH    CS
    POP     DS
    LEA    SI,SWAP_BUFFER
    CLD
    REP    MOVSB;恢复系统的 SDA
    POPF
    POP    SI
    POP    DI
    POP    CX
    POP    ES
    POP    DS
    MOV    SS,CS,[SAVE_SS];恢复系统堆栈
    MOV    SP,CS,[SAVE_SP]
    STI
    ENDM
;初始化程序段(第三段,不驻留内存)
INIT_DOS_SWAP_M MACRO
    LOCAL    GET_SWAP_OK,SWAP_ERROR
    PUSH    DS
    PUSH    ES
    MOV    AX,5D06H;取得 DOS 数据交换区指针
    INT     21H
    JNC    GET_SWAP_OK
    SWAP_ERROR
    POP     ES
    POP     DS
    MOV    DX,OFFSET DOS_SWAP_ERR;取指针
        出错,退出
    @DispStr DX,CS
    @Exit
    GET_SWAP_OK: MOV WORD PTR CS:[SWAP_
        PTR],SI;保存 SDA 的指针
    MOV    AX,DS
    MOV    WORD PTR CS:[SWAP_PTR+2],AX
    CMP    CX,SWAP_BUFFER_SIZE;测试 SDA 的
        最大长度
    JG     SWAP_ERROR;超过缓冲区长度则出错
    MOV    CS:[SWAP_SIZE_INDOS],CX;保存最大
        交换长度
    MOV    CS:[SWAP_SIZE_ALWAYS],DX;保存最
        小交换长度
    MOV    AX,352AH;取得系统的 INT 2AH 矢量
    INT     21H
    LEA    SI,OLD_INT_2AH
    PUSH    CS
    POP     DS

```

```

MOV     WORD PTR [SI],BX;保存系统的 INT
        2AH 矢量
MOV     WORD PTR [SI+2],ES
MOV     AX,252AH;设置 TSR 程序的 INT 2AH 矢量
LEA     DX,NEW_INT_2AH
INT     21H
MOV     AX,3510H
INT     21H
LEA     SI,OLD_INT_10H
PUSH    CS
POP     DS
MOV     WORD PTR [SI],BX
MOV     WORD PTR [SI+2],ES
MOV     AX,2510H
LEA     DX,NEW_INT_10H
INT     21H
MOV     AX,3513H
INT     21H
LEA     SI,OLD_INT_13H
PUSH    CS
POP     DS
MOV     WORD PTR [SI],BX
MOV     WORD PTR [SI+2],ES
MOV     AX,2513H
LEA     DX,NEW_INT_13H
INT     21H
MOV     AX,351CH
INT     21H
LEA     SI,OLD_INT_1CH
PUSH    CS
POP     DS
MOV     WORD PTR [SI],BX
MOV     WORD PTR [SI+2],ES
MOV     AX,251CH
LEA     DX,NEW_INT_1CH
INT     21H
MOV     WORD PTR CS:[NEW_INT_05H_PTR

```

```

+2],DS
MOV     WORD PTR CS:[NEW_INT_05H_
        PTR],OFFSET INT5
POP     ES
POP     DS
ENDM
SET_PSP MACRO SET_ADDR;设置新的程序段前缀
        (PSP)
PUSH    BX
PUSH    AX
MOV     BX,SET_ADDR
MOV     AH,50H
INT     21H
POP     AX
POP     BX
ENDM
SET_DTA MACRO SET_ADDR;设置新的磁盘传送缓
        冲区
PUSH    DS;其长度应不小于 128 字节
PUSH    DX
PUSH    AX
MOV     DX,SEG SET_ADDR
MOV     DS,DX
MOV     DX,OFFSET SET_ADDR
MOV     AH,1AH
INT     21H
POP     AX
POP     DX
POP     DS
ENDM

```

6 参考资料

- [1]吴双编译《尚未公开的 DOS 秘密》海洋出版社 1991 年 5 月 175 页。
- [2]求伯君主编《深入 DOS 编程》北京大学出版社 1993 年 1 月。

(上接 34 页)

4 其他

(1) 应随时删除屏幕垃圾

在问答过程中,可能产生很多无用的屏幕垃圾,应随时在屏幕上删除它们。

经慎重考虑,确实需要保留的操作痕迹,才保留下来。

(2) 屏幕设计应注重实用性,不摆花架子

在大多数情况下,绝大多数操作人员在用熟了该软件后,常常根本连屏幕都不看。你的屏幕设计得再豪华,也不过是对牛弹琴。

(3) 帮助功能在不同的模块里应该有针对性的不同的帮助

不应当发生在任何一个模块里调“Help”,得到的都是同一个内容的帮助的情况。

(4) 对于一个好的系统来说,一定要配备一份详尽的用户使用手册。

(5) 屏幕在操作过程中不要出现错别字,不要出现不通的语句。

例如,“请稍候”的“候”字常常被写成“侯”字。

软件温度计的设计

新疆电子计算中心 龔正科 (乌鲁木齐 830011)

国外的许多软件,在系统安装,系统生成,文件拷贝等操作过程中,为了形象地描述其进行的过程,都设计了一个叫做温度计的程序,它模拟过程进展情况,象一个温度计一样,在软件执行过程中,不断地提高“温度”,当温度达到 100%,表示程序运行结束。

下面给出这个程序,它是用 FOXPRO 2.0 编写的。

* 程序名称:TH. PRG
* 功能说明:温度计演示程序

```
PUBLIC m.g_thermwidth
SET TALK OFF
DO acttherm WITH "软件温度计..." && 激活温度计
DO updtherm WITH 2 && 初始温度计值
FOR i=1 TO 10000
do case
case i=1000
DO updtherm WITH 10
case i=2000
DO updtherm WITH 20
case i=3000
DO updtherm WITH 30
case i=4000
DO updtherm WITH 40
case i=5000
DO updtherm WITH 50
case i=6000
DO updtherm WITH 60
case i=7000
DO updtherm WITH 70
case i=8000
DO updtherm WITH 80
case i=9000
DO updtherm WITH 90
case i=10000
DO updtherm WITH 100
endcase
NEXT i
DO updtherm WITH 100 && 显示 100%
```

DO deactthermo && 关闭温度计
RETURN

* ACTTHERM((text))—Activate thermometer.
* 功能说明:激活温度计
* 过程名称:ACTTHERM

```
PROCEDURE acttherm
PARAMETER m.text
PRIVATE m.therm_x1,m.therm_y1,m.therm_x2,m.therm_y2
m.therm_x1 =INT((SROW()-6)/2)
&& 根据屏幕的行数,确定 X1 坐标
m.therm_y1 =INT((SCOL()-57)/2)
&& 根据屏幕的列数,确定 Y1 坐标
m.therm_x2 =m.therm_x1+6
&& 确定右下角坐标
m.therm_y2 =m.therm_y1+57
DEFINE WINDOW thermometer
FROM m.therm_x1,m.therm_y1 TO m.therm_x2,
m.therm_y2;
FLOAT DOUBLE COLOR SCHEME 5
```

&& 定义窗口
ACTIVATE WINDOW thermometer && 激活温度计
@0,3 SAY m.text && 在窗口的 0 行显示信息
m.g_thermwidth=54
@2,1 TO4 , m.g_thermwidth
&& 在窗口中画一个(2,1)~(4,54)框
RETURN

* UPDTHERM((percent))_Update thermometer.
* 过程名称:UPDTHERM
* 功能说明:修改温度计指针

```
PROCEDURE updtherm
PARAMETER m.percent
PRIVATE m.nblocks,m.percent
ACTIVATE WINDOW thermometer && 激活温度窗口
m.nblocks=(m.percent/100)*(m.g_thermwidth-4)
&& 计算百分数
@3,3 SAY REPLICATE(" ",m.nblocks) && 显示温度计
RETURN
```

* DEACTTHERMO—Deactivate and Release thermometer window.
* 过程名称:DEACTTHERMO
* 功能说明:关闭温度计窗口并释放

```
PROCEDURE deactthermo
RELEASE WINDOW thermometer && 释放窗口
RETURN
```

一种工业控制机中使用的通用型智能显示终端

北京工业大学基础部电工教研室 蒋大林 (100022)

摘 要 本文介绍了一种适用于 STD 工业控制机中使用的通用型智能显示终端的设计情况。文章中给出了该终端的功能指标及软硬件的组成。

1 引 言

随着计算机技术的普及和发展,STD 工业控制机以其优良的性能被越来越多的用户所采用。但是由于使用的场合不同,要求显示的信息和画面也不相同,因此使用者往往需要根据自己的要求,去设计编制复杂的屏幕显示程序。

目前虽然市场上销售的 STD 总线结构的 CRT 显示卡的种类较多,但所有 CRT 显示卡都是非智能型的。即无论显示卡中采用何种型号的显示控制器芯片(CRTC),显示卡都是通过 STD 总线与主机 CPU 板相连接。由主机板上的 CPU 通过运行显示管理与控制程序来对显示卡上的 CRTC 进行控制,在屏幕上显示出字符或图形。显然使用这种非智能型的 CRT 显示卡,在设计屏幕显示程序时,需要详细地掌握卡内使用的 CRTC 的性能,这对于大多数用户来说无疑是一件十分复杂的工作。

为了克服非智能型 CRT 显示卡的不足,我们设计了一种适用于 STD 总线结构的工业控制机中使用的智能型 CRT 显示终端。该显示终端采用新颖的设计方案和大规模集成电路技术,将 CPU、CRTC、通信接口以及存储器等电路集成在一块标准的 STD 总线板上,同时配备了丰富的显示管理和控制程序,使其具有与普通行式打印机兼容的命令集,这样对使用者来说,该智能型显示终端就象一台行式打印机,使用起来十分方便。

2 智能显示终端的性能

该智能显示终端的主要性能如下:

(1)具有图形工作方式和字符工作方式,能够显示全部 ASCII 字符及汉字,具有多种图形命令。

(2)采用 UPD-7220 芯片作显示控制器(CRTC),具有较高的显示分辨率,可以配接多种制式的 CRT 显示器,并且可以用软件命令来设定显示器的分辨率。

(3)采用标准的打印机并行接口与主机进行通信,通信速度快。并且具有和普通行式打印机相同的

控制代码命令集,程序设计简单。

(4)采用标准的 STD 总线结构,占用两个 I/O 口地址。并且可以多块卡共同使用,使控制系统可以同时配接多个显示器,显示不同的内容。

(5)配有实时监控程序,采用多任务调度方法对显示终端进行实时操作管理,响应速度快,实时性好。

3 硬件组成及特点

智能显示终端由 STD 显示控制卡和 CRT 显示器组成。显示器的尺寸及分辨率可根据需要选定,显示器与显示控制卡采用标准九针视频接口相连接。显示控制卡采用 STD 总线结构,由 CPU、CRTC、通信接口、存储器及视频合成电路等几部分组成。它相当于把普通 STD 总线结构的 CPU 板、CRT 板和 I/O 扩展板的功能组合在一起,因而具有较高的性能价格比。智能显示终端的显示控制板的硬件框图见图 1。

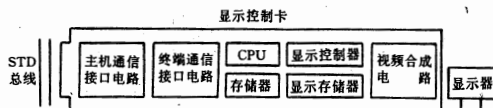


图 1 智能显示终端显示控制卡的硬件框图

该智能显示终端显示卡的最大特点是把主机与显示器之间的通信接口电路排在一起。这样对控制系统的主 CPU 来说,智能显示终端仅相当于一台行式打印机。主 CPU 通过 STD 总线对显示控制卡上的主机通信接口进行操作,从而控制屏幕显示的内容。由于通信接口排在一起,减少了连线过长造成的干扰,有助于提高数据传送的速度,同时这种结构使得该智能显示终端的显示卡可与多种型号的 STD 总线结构的 CPU 板相联接。

4 软件设计

为了提高智能显示终端的处理能力,在软件设计中采用了实时多任务调度管理程序,该程序包括初始化程序、多任务调度程度、显示处理程序、通信处理程序和实时钟管理程序等几部分。智能显示终

端的软件全部采用汇编语言设计,其软件整体框图见图 2。

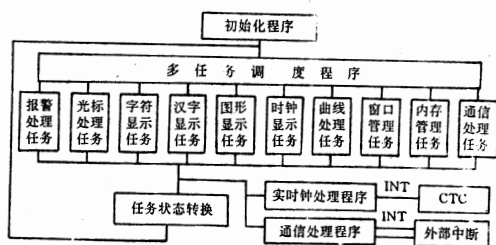


图 2 智能显示终端的软件整体框图

(1) 初始化程序负责对智能显示终端的显示控制卡中使用的芯片进行设置和自检,以保证智能显示终端开机以后可以立即投入正常的工作。

(2) 多任务调度程序负责智能显示终端的实时管理工作。在程序设计中采用了优先数算法,即优先级高的先于优先级低的,同一优先数的按先进先出的原则进行调度,也就是先进入就绪队列的任务首先变成执行状态。所有的任务控制块(PCB)排成一个就绪队列,优先级高的排在队首,优先级低的排在队尾。

为了提高智能显示终端的实时响应指标,应尽量减少多任务调度程序占用 CPU 的时间。为此在多任务调度程序中采用了多堆栈的办法,为各个任务分配了不同的堆栈指针,当中断处理时,把现场的信息存入各个任务本身的堆栈区内。

(3) 显示处理程序负责智能显示终端的显示处理任务。它由多个命令组成,可以完成字符显示及多种作图功能。在多任务调度程序的管理下,智能显示

终端可以同时执行多种显示处理任务。

(4) 通信处理程序负责智能显示终端与主机之间的数据交换和命令识别工作。为了提高智能显示终端的通用性,在软件设计中采用了普通的行式打印机中使用的命令格式,使显示命令的代码及格式与普通的行式打印机中使用的命令格式完全相同,这样用户在设计屏幕显示内容时,就如同用打印机拷贝一样方便,并且不必再单独设计打印程序,具有事半功倍的效果。

(5) 实时钟管理程序提供了一个定时信号来完成时间计时和各任务状态的转换。当 CPU 响应实时钟的中断申请后,直接在实时钟的中断处理程序中完成任务状态的转换及定时操作,然后返回调度程序。这样可加快智能显示终端的处理速度。

5 结束语

智能显示终端为工业控制机增加了一个新的外部设备,它采用新颖的设计和实时监控程序,大大减轻了用户的设计工作,具有较高的实用性和性价比。

6 参考文献

- [1] INTEL 《Microsystem Components Handbook》. 1988
- [2] 郝忠孝 《微型机实时软件设计》. 机械工业出版社. 1989
- [3] 熊桂喜 《PC 机各类显示系统的编程及应用》. 清华大学出版社. 1992
- [4] 蒋大林 《TP801 单板机显示打印接口板的设计》,《电子科学技术》第 16 卷第 173 期 P9~12

新技术产品简介

一机多屏 显示分配器

此种分配器由专用集成电路实现,可多级长距离驱动,传送距离可达 200 米,图像清晰,效果逼真,无拖尾,无毛刺,性能稳定,质量可靠,适合于 EGA、CGA、VGA、双频单显等显示分配,一部主机输出,通过显示分配器可实现一机五屏、一机十屏显示,广泛应用于证券、期货、教学、车站、机场、码头、广

告等各行各业及公共场所的实时多屏显示。

VGA——PAL 彩色电视卡

可代替 CGA、EGA、VGA 显示器,电脑输出的图文彩色数据信息转换为彩色 TV PAL 制式,效果良好,图像清晰逼真,使彩色电视机变为电脑显示屏。

以上产品有现货供应,质量保证并负责售后维修。需要者,请同本刊编辑部联系。

智能型电炉电源供电装置

吉林省计算机技术研究所 闫立恒 王延平 吕文 (长春 130012)
吉林省电力职工大学 闫彦 (长春 130010)

1 前言

(1)智能型电炉电源供电和温度控制的结合,属于现场数据采集和控制供电电源系统,该系统是由单片机和可控硅为核心部件组成。智能型电炉的硬件和软件组成各不相同,但它们有共同的核心部件,在此基础上进行调整,可适用于各种类型的电炉,满足电炉在各种场合下的应用。把现场传感器采集来的信号,经过放大器进入单片机,经数据处理、变换、计算、编程序等。调节控制输出脉冲信号,直接驱动可控硅触发极,使可控硅导电,把电炉由交流供电方式,变换成直流供电方式。因为脉动直流供电电源电压大小,容易控制,能满足电炉热加工产品的工艺要求,是旧电炉改造新电炉生产选用的新产品。

(2)机械加工件质量使用寿命短,主要是热处理这道工序,无论是淬火、退火、渗炭……等,质量上很难达到要求的。主要原因是电炉电源供电和温度不容易控制,如果说用交流供电方式算作控制的话,那么误差是相当大的,因而导致出加工的产品质量问题。用智能型电炉电源供电,就容易实现对电炉温度的控制,准确而且精度高,也就是说电炉生产技术问题得到解决,那么也就解决了电炉热加工生产的产品质量。

(3)节约投入资金,特殊种类电炉例如:推杆式电炉、紫外线烘干电炉、喷涂烘干电炉、冶金用电炉、食品烘干机……等,温度要求低,为此采用一台大型调压设备,改变电源电压给电炉供电,才能实现温度的特殊控制要求。智能型电炉电源供电,不采用其它专用设备,就能满足供电和控温要求,仅此项节约投资费用就相当可观了。

(4)节约资源、能源、提高产品合格率,减少返修率。92年一用户统计,一台H100千瓦电炉,仅减少返修率一项,每年节约电能1.85万度,合人民币叁万捌仟元。节约了资源、能源和劳动生产力,用智能型电炉电源供电装置,可节约用电30%左右。

(5)应用先进的电子、计算机技术和电力技术相结合,使电炉生产技术现代化,同时也提高了自动化水平,节约了劳动生产力。

(6)智能型电炉供电电源控制装置,缩短电炉的预热时间,提前预热可实现无人操作,同时延长电炉

炉窑的使用周期。

总之,可直接把现场传感器信号(热电偶……等)的毫安和毫安信号,这种模拟信号量经过放大器,进入单片机,经过变换处理、运算、编程等。一方面将当前炉温值输出到数码管,显示电炉当前的温度值;另一方面输出脉冲信号,直接触发可控硅导通角,使可控硅导通,从而可控硅把交流电源变成脉动直流电源,脉动直流电源送往电炉,使电炉加热,进行各种各样的热加工生产。

2 智能型电炉供电电源,实用性和可靠性,贯穿每一个环节,采用的方法有:

(1)主回路,电炉加热、升温、恒温、保温、降温,均采用无触点自动进行控制。

(2)总体设计时对智能电炉供电和温度控制的核心部件,设有自动检测功能。当电炉正常运转时,定时的对系统各模块进行自动诊断。对外部出现异常情况时能作出应变处理,若出现的情况无法解决时,及时报警,并能自动断开主回路的自动控制断路器。

(3)减少系统的故障率

①提高电源的质量,电源干扰是控制系统不能稳定的主要原因,因此采用防止干扰电源,用屏蔽、隔离的电源变压器,中间层接地。使用完善的高低通滤波器电源。例如脉冲触发使用光电隔离触发电路,防止可控硅导通时回馈干扰控制系统……等。

②防止电磁干扰、高频干扰,合理地处理系统地、数字地、模拟地和大地之间的关系。

③元器件的选择,电炉工作环境条件不好,在设计时对元器件的负载、速度、功耗、过载……等,都留有一定的余量,并对元器件进行筛选。

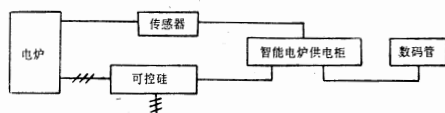
④性能/价格比高。

智能电炉供电电源,能否广泛的应用,关键有两个方面:一方面采用先进的技术,硬件软化是提高性能/价格比的基本方法,在满足功能指标的前提下尽量用软件实现,减少硬件成本。另一方面是提高人的素质,加强自动化和现代化控制观念的修养,同时也要从长远观点出发,节约资源,节约能源,广泛使用智能型供电电源。

3 供电电源系统框图及主回路连接方法

(1)智能型电炉供电电源和控温系统装置如下图,从系统装置框图看出,单片机一方面输出电炉的

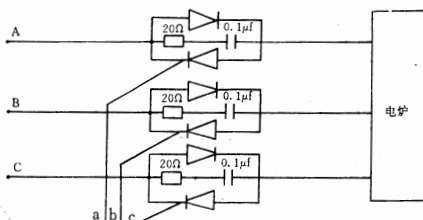
当前温度。另一方面单片机输出脉冲控制可控硅,给电炉供电形成一个闭环调节控制系统。



本系统只要改变可控硅容量的大小,可适用于大、中、小等电炉,具有投资少、见效快、简单易行的特点。是旧电炉改造时必须采用的最佳方案,电炉生产厂家要充分应用的,现代化新技术。经过长期运行该系统稳定可靠,是电炉改造更新换代的产品。

(2) 供电电源和电阻炉的连接方法

主回路原理图如下:



电阻炉内部电阻丝接法为:

①星形接法。②角形接法。③一字形接法。

4 主要技术指标

(1)输入信号 0~10 毫安 输入电阻 0~5kΩ

(2)输出信号: 三相移相脉冲,每相输出正脉冲。

20Ω 负载时,脉冲幅值 > 4v, 脉冲宽度 35μs,, 最大导通角 150°。

(3)电源电压: 交流 380 伏 ± 10%, 50 周波

(4)主频 6 兆赫 (5)工作环境

温度 0~50°范围内, 相对湿度不大于 85%。

(6)输出三相脉冲相位差 120° ± 5

(7)配用可控硅

KP-50A, KP-100A, KP-200A, KP-600A/1000 伏

ZP-50A, ZP-100A, ZP-200A, ZP-600A/1000 伏

(8)串型通讯接口一路

5 智能型电炉供电电源控温系统配置的选择

(1)存贮器 8031 单片机内部没有存贮器 ROM,对于一般工业仪表系统,初始化程序所需最大空间为 2K 字节,监控程序所需最大空间约为 3K 字节,主程序所需最大空间约为 10K 字节,中断程序所需最大空间 5K 字节,所以外存贮器选用 32K27C256 作为程序存贮器用,是完全可以满足要

求的。又选用 2864A 存贮器,电炉在运行过程中,其它原因停电,不久电炉又恢复运行,那么电炉恢复运行后,可按着电炉停止运行前参数运行,因为停电时的运行状态参数,已被存入 2864A 存贮器,工艺参数也存入。

(2)串行异步通讯接口 这一接口电路的设置,提供现代化管理与控制思想。电炉可与任何微机 RS-232C 接口通讯,便于群控和分散式控制提供一个先决条件。

(3)高精度放大器

传感器输出的大部分信号是毫伏或毫安值,不能直接进入单片机的 A/D 电路,只有将这一信号放大成伏级信号,才能进入单片机,我们选用放大倍数为 0~1000 倍的高精度放大器,精度高、漂移较小,工作稳定可靠。

6 应用软件设计

在金属热加工中、有淬火炉、退火炉、渗炭炉等;玻璃工业中:熔化炉、槽沉炉、退火炉;大部分属于中温用电加热电炉。紫外线烘干炉、食品电炉、油漆烘干炉等为低温用电加热电阻炉。各类电炉都有一些工艺参数要求,我们采用输入值的办法,存入到固定存贮器,不丢失原始数据。电炉的升温速率、恒温温度、降温速率、恒温时间……等,均可作为人机输入。有特殊要求的电炉,每开炉之前单独输入,没有机器认为一般处理,自动采用一般参数,作为软件方法处理。为了生产各种热加工产品,要求智能电炉供电和温度满足各种产品加工工艺,软件方法设计时实现电源供电的调节,保持电炉温度。软件程序设计和编写是以电炉加工金属产品工艺要求为基础,设计了通用软件程序和数据处理方法,可用于电炉生产各种产品,有很强的代表性。可以说适合于各种电炉的各种工艺要求,在此种条件下,仅仅变更几个工艺参数,程序就能满足各种类型电炉生产各种产品了,免去了重复设计编写应用程序的劳动。经过多年现场的应用,实际应用中的改进,应用软件程序设计和编写是符合现实的。达到节约资源、能源等目的,可以说使用方便、操作简单。

应用软件程序包括:用电加热电炉的电源和控温是一致的,调节供电电源电压,使电流流过电阻丝加热器多和少(即电流大和小),电炉供电电源加以控制,保证电炉温度为目的,是自适应的闭环调节控制系统。温度控制有:升温段、恒温段、降温段和数据处理方法。其中控制电炉供电电源满足电炉温各段温度要求为目的。由于电炉加工产品不同,其中有的电炉工艺要求不一样,把那些要求特殊的条件,归纳为灵活处理参数,灵活处理参数可单独键入,以便达

到各自的生产要求。输入代码对应参数表的使用,就是实现一种电炉生产同一种产品,工艺要求相同,那么可键入一次工艺参数;如果一台电炉生产不同种类产品,工艺要求不相同,那么可键入不同种类工艺参数;在满足电炉加工产品的条件下,不再重复编写应用程序,而只键入十六个参数之中的几个参数,便可实现各自的不同要求,达到各自的预期目的。

软件程序和数据处理方法,在单片机中得到实现,用智能型电炉供电电源和控温方法,控制电炉作到了即能保证产品质量,节约资源和能源,延长炉龄,又能改善劳动环境,减轻劳动强度。

人机——对话,关于人机对话部分参数的输入,采用十进制整数/小数和十六进制整数混合输入方式,共键入十五个参数。

设计时,似全部参数均采用十进制数的输入方式,后来经实践时发现十进制方式,对于现场控制是达不到要求的,为了克服这一不足,而采用了十六进制。恒温控制选用的时间是十分钟,若用十进制输入,最大值为 9999,即 $9999 \times 10 = 99990$ 分钟 = 1666.5 小时 = 69.4 天;若用十六进制输入,最大值为 $\text{FFFFH} = 65535 \times 10 = 655350$ 分钟 = 10922.5 小时 = 404.7 天,可见表示时间如此之长,可以满足各种电炉设备的恒温时间要求,故此采用了后种方式,其它的参数也有类似的情况发生。人机对话参数键入到 2864A 存储器。

恒温段的控制算法,在恒温段采用 PID 算法和较为普遍的开关式控制算法,依据 PID 增量 ΔP_n 的符号增/减模出值,再依据 $R - Y_n$ 的符号对开关量输出位的置/清,增量算式为 $\Delta P_n = P[(Y_{n-1} - Y_n) + I(R - Y_n) + D(2Y_{n-1} - Y_n - Y_{n-2})]$ 。

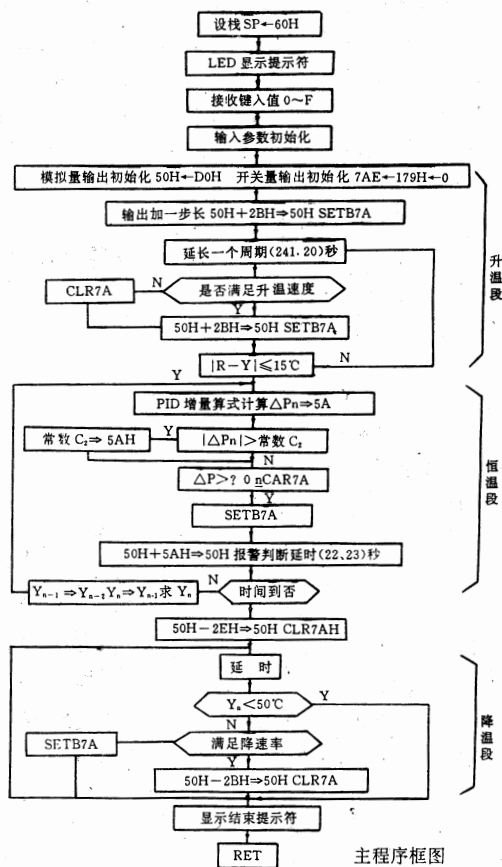
根据问题的情况也可选用 PID—PD 选择算法,大偏差时投入 PD 算法,因积分分量用于消除稳态误差,在动态过程中大的积分分量可能使系统的稳定性变坏。改善系统的动态品质,小偏差时投入 PID 算法,保证了系统的稳态品质。

模拟量输入时加入冷端补偿,由模拟量到工程量转换时,采用线性公式 $y = a \cdot A + b$ + 冷端补偿,冷端补偿是用软件实现的。输入代码对应参数表如下:

按键	参 数	按键	参 数
0	PID 调节周期	A	工程量计算 $Y_n = ax + b$ 的 a 参数
1	标定值 R	B	工程量计算 $Y_n = ax + b$ 的 b 参数
2	PID 算式中的 P 参数	C	计算 $Y_n = ax^2 + bx + c$ 留扩充时用
3	PID 算式中的 I 参数	D	冷端补偿

4	PID 算式中的 D 参数	E	升降温速率
5	升温周期	F	键入值结果
6	升温步长		
7	降温周期		
8	降温步长		
9	恒温时间		

7 主程序框图



8 结 论

智能型电炉电源供电柜,采用 MCS—51 系列单片机和可控硅为主要部件的新兴技术,具有成本低、体积小、测量与控制功能强,运行稳定抗扰能力强,无触点控制以及便于应用的特点。可按着各种领域各种部门的需要,如节约资源、节约能源、改善劳动环境、增加生产效益,提高自动化水平,研制出来的最新智能型电源供电柜。该智能型电源供电柜经过多年来不同领域的运行应用,稳定可靠,达到了预期的效果,受到应用单位的好评。电源供电柜控制思想实现方便灵活,技术先进,能满足旧的电炉设备技术改造的需要,新建和扩建电炉的需要,是使电炉生产现代化,提高自动化水平的新产品。

一种电梯微机控制系统的设计

山东工业大学计算机系 张 昆 (济南 250014)

1 前言

微机用于控制电梯和继电器接触逻辑控制相比,具有许多优越性,它成本低,体积小,可靠性高,使用寿命长,简化了安装调试工作,更突出的优点是微型计算机具有算术运算功能和灵活的逻辑运算功能,因此可以实现更完善的自动控制,例如对于电梯平层可以实现自适应控制,使平层情况达到最佳状态。

本文将提供一种电梯微机控制设计方案,并对有关原理作出扼要论述。

设计实现的是载人电梯的自动控制,主要功能有:外呼、内选的自动登记,并在登记后给呼叫者以回答,避免重复呼叫。电梯可按登记情况自动运行和停靠。能自动开门、关门。在关门过程中门完全关闭之前,若遇到关门障碍可以重新打开,当障碍消除又自动关闭。当载荷超重时能发出报警,并拒绝关门。能自动改进平层情况。

该系统的两大特点:

(1) 采用双机系统:采用两个 8031 型单片机,分别构成主机和从机。主机置于机房内,主要用于控制轿厢运行,呼叫登记,厢外显示,向从机发送状态信息和一些命令。从机置于轿厢内,主要任务是控制轿厢门的运动,内选登记和轿厢内的各种显示,以及接收和处理轿厢上的其它信息。并将必要信息发送到主机。两机间采用串行方式通信,发送和接收信息。采用双机控制、串行通信的目的在于减少轿厢与外界的连接导线。这部分导线不单长度大,而且其一端是随轿厢运动的,易于折损,减少它们的数量具有重要意义。

(2) 采用光栅作为反映轿厢运动的传感器,对于光栅盘转动产生的电脉冲计数并通过计算可以确定轿厢位置。位置计算分别用于确定轿厢速度切换时机、发出停靠命令时机,以及检查平层情况等几个方面。传统上,在井道中装设开关式传感器,作为减速、加速位置的标志,使用了光栅盘则可省去这部分传感器,大大减轻了安装和维护的工作量。

2 硬件系统结构和工作原理

系统构成框图如图 1 所示。图中箭头表示数据传送方向。

主机核心由一片单片机 8031 构成,程序用一片

ROM 2764 存贮,随机数据存贮器用一片 6116 构成。8031 的 P₀ 口作为高 8 位地址输出。由 ALE 信号打入锁存器, P₂ 口作为高 8 位地址输出。P₀ 口还分时兼作数据输出用。输入信号按阵列形式连接。阵列的各列经与门接于 8031 的 INT₀ 引脚,当有输入信号时引起中断,然后 8031 的 P₀ 口送出扫描各行的输出序列,由 P₁ 口取回扫描结果,由此可判断出输入信号产生的位置,分别转入相应的处理程序。

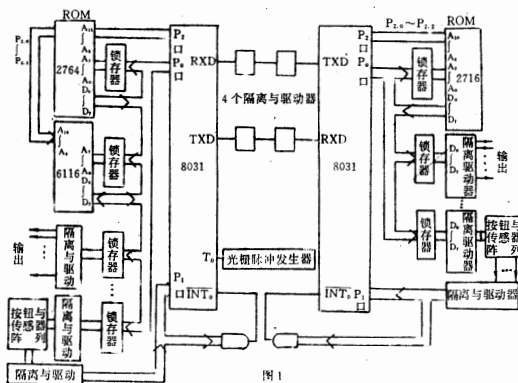


图 1

输入信号由以下几部分组成:①外呼信号:由设在楼道墙壁上的按钮产生,中间楼层每层各设上行和下行按钮共 2 个,底层设上行按钮一个,顶层设下行按钮一个。②平层信号:每层一个,安装在井道中楼层地板处。③控制台按键信号,有数字键 10 个,功能键 6 个。用于系统调试。

输出信号,由 P₀ 口送入外部寄存器。寄存器的选择由地址译码决定。输出信号有以下三类:①牵引电机控制信号:包括正反转控制、速度切换控制以及分级切除起动电阻控制,共 6 个信号。②登记显示信号,每个呼叫按钮配用一个。登记后指示灯亮,电梯到达该站后,指示灯灭。③电梯运行状态显示:方向和楼层。④控制台显示:用 7 个数码管显示。

由光栅盘产生的脉冲信号,经 T₀(P_{3.4})引脚输入单片机。

所有输入、输出与单片机的连接均通过隔离驱动器,以避免外界干扰信号引入计算机。

从机的连接与主机相似,但由于数据量少而不使用随机存贮器,仅使用单片机内部存贮器即可。

从机输入信号有:内选信号,对应每层一个;直驶信号1个;超载信号1个;门位置信号4个;关门障碍信号1个。

从机输出信号有:登记显示信号,对应每楼层一个;楼层显示信号;运行方向信号;超重警告信号各1个;门电机控制信号5个。

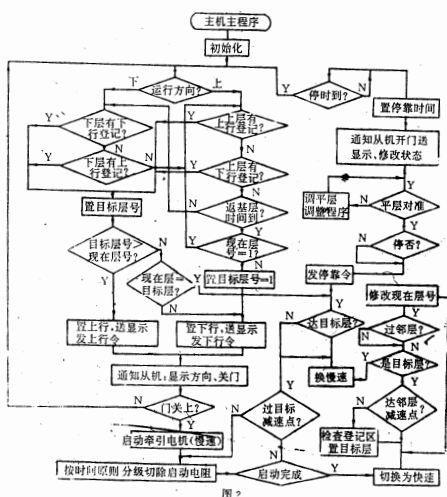
3 软件设计

程序分五大类:·主机监控主程序·主机中断服务程序·从机监控主程序·从机中断服务程序·子程序。

主机中断服务程序包括:·呼叫信号处理·光栅信号计数处理·串行口接收中断处理·控制台按键信号处理。

从机中断服务程序有:A. 内选信号处理 B. 串行口接收中断服务 C. 直驶信号处理。

主机控制主程序框图如图2所示。



现将程序设计中几个主要问题的处理原则叙述如下:

(1) 呼叫登记 通过装在每层楼墙壁上的上行或下行按钮呼叫电梯,称为外呼;轿厢内,通过选择按钮选择要去楼层,称为内呼或内选。外呼和内选通过中断处理程序,综合成电梯运行任务集合,登记在固定的存贮区,每项任务占一个存贮单元。登记区分上行任务、下行任务两个区。外呼按呼叫方向登记在对应登记区的对应单元。内选按选择楼层与现在楼层比较结果,登记在上行区或下行区,若选择层高于现行层则登记在上行区,否则登记在下行区。因此,对于同一存贮单元,有否登记是外呼、内选相“逻辑或”的结果。

辑或”的结果。

(2) 当前运行目标的确定,在主机的主程序中,在主机开始工作,经初始化程序完成工作的准备后,或主机工作过程中当电梯停靠时间到,则主程序进入检查登记确定当前运行目标的程序段。为实现顺向接送乘客原则,检查按一定顺序进行,按照这一特定顺序首先检查到的登记便是电梯运行的当前目标站,将其登入专用存贮单元。检查顺序应是按运行方向和现在层次,从最近的一层开始,顺次检查各层,直至最远层(一层或顶层)的邻层,而后从最远层开始检查反向登记,直至最近层。这过程中,如有登记,则电梯按原方向运行。若无登记,则再检查反向登记区,从最近层次开始,直至最远的端层,然后又进入顺向登记区,由远而近逐层检查,直至最近的一层。在这个过程中,若检查到登记,则电梯改变运行方向。例如现在电梯向上运行,停在四层,检查顺序是先检查上行登记区,从五层开始逐层向上检查直邻顶层,而后检查下行登记区,从顶层开始逐次向下直至五层。这个过程中,检查到登记,则电梯仍上行,若无登记则在下行区从三层开始检查到二层,而后检查上行登记区,从一层检查到三层,若有登记,则电梯下行。若上行、下行都无登记,则电梯停留较长时间后返回一层。

(3) 当前运行目标的改变 电梯启动前,必须根据登记情况,确定目标,以便确定启动方向,但在运行过程中,顺向更为接近的某层有外呼或内选,只要在电梯到达该层的减速点前,都应能得到服务。为此程序中安排了相应的登记项检查,在电梯到达某一层的减速点前都检查该层有无登记,若有,则将其置为目标。轿厢到达该层时停靠。

(4) 直驶功能 轿厢内设直驶按钮,按直驶按钮,可使电梯进入直驶状态,直驶状态下,轿厢可直达某层,而不在中途停靠(即使途经的楼层有登记)。直驶状态用于完成紧急事务。为具有此项功能,直驶按钮设计成软开关形式,在非直驶状态下按一次该按钮,则转换为直驶状态;在直驶状态下,按一次该按钮,则转为非直驶状态。直驶状态的显示为,直驶指示灯亮,直驶登记目标显示灯亮,其它登记目标显示灭。完成直驶任务后自动恢复非直驶状态。直驶功能在程序上的处理是:1. 直驶登记在相应中断中完成,其状态显示亦在此中断程序中完成。存贮器中单独设直驶目标登记区,登记内容为直驶目标楼层号。2. 在主程序中直驶的处理包含在目标登记框中,若为直驶状态则将直驶目标登为目标;其它登记不能登入目标单元。直驶任务完成后,恢复非直驶状态的任务隐含在修改状态的方块中(见图2)

平层调整子程序利用的基本原理是依据电梯停靠的位置偏差。修改停靠命令发出时间。

(5) 事故停车后的处理 非正常情况下(例如突然断电)轿厢停在两楼层中间位置,电源恢复后电梯可重新启动,起动过程中若轿厢已过目标层的减速点,电机将不转入快速,仍以起动时的慢速运行至目标,否则,仍以正常的速度切换程序运行。

(6) 轿厢位置计数校准 为防止计数积累误差,当轿厢到达底层时,计数器清零。此外,每楼层都设停靠位置传感器,一方面用此作为平层的自动调整,另一方面当停靠在准确位置时,计算机可以探查到此信号,此时可用它校准位置计数,当计数与根据位置信号计算出的对应计数值相差过大时,应发出工作出错的信息。这部分工作应放在平层调整程序中完成。

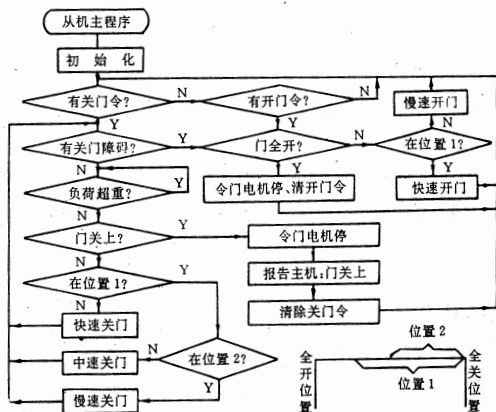


图3 门位置示意图

(7) 控制台按钮 用于对控制系统程序的调试和某些参数的输入、修改。通过特定按钮的中断处理程序,可运行调试程序,在调试程序控制下可设置断点或单步运行,可向某些存储单元输入数据,可显示某些存储单元的数据或程序代码。这样,人们可方便地观察程序运行情况,以及对一些参数作出调整,

(上接 55 页)

配方、混合机混合时间、每斗料额定重量、原料和仓号对应表等。

(10) 能打印每班的生产情况日报表和月报表。

6 结束语

我们所研制和开发的微机控制系统先后在两个饲料厂投入运行。在有大型电机频繁启动以及振动极大的现场,该系统仍能连续、稳定、准确无误地工作。该系统给饲料厂带来了显著的经济效益与社会效益,受到了用户的欢迎。实践证明,该系统具有结

为在实践中排除故障,改进程序或扩充程序功能创造方便条件。

(8) 从机主程序框图如图3所示。从机根据主机命令控制开门和关门动作,根据门的位置确定门运动的速度。当关门时,若检查到有关门障碍或超重则停止关门,且将开至最大位置,以防夹住乘客或物件。

(9) 主从二机间的通信 主、从机都经常准备接收以对方发来的信息,当接收到信息时引起中断,转入相应处理程序。发送信息,由相应程序在适当时机进行,例如从机向主机发送的登记信息,是在每次内选中断处理程序中进行的;又如主机向从机发送的开门命令是在轿厢正常停靠之后紧接着进行的。

4 抗干扰措施

载人电梯运行的安全可靠十分重要,必须从机械和控制两个方面采取措施予以保证。就控制系统而言,增强抗干扰性能,对保证其可靠工作有重要意义。电梯数控系统工作环境存在着较强的干扰:由于主牵引电机功率较大,起动、速度切换频繁,因之电流变化较频繁,变化幅度也大,必然造成电网电压的波动;接触器触头通过电流较大,其频繁的开、关动作,必将一次次的造成冲击电压,以及触头间的放电,前者以电网过电压形式,后者以辐射电磁波形式干扰控制系统。为防止上述以及其它干扰,可采取以下措施:

(1) 控制系统的直流电源应有较高质量,特别是有较好的稳压性能和滤波性能。必要时在直流电源之前加交流稳压器。主、从控制机各用独立的直流稳压电源。

(2) 控制系统有较好的屏蔽罩,以防空间电磁波进入。

(3) 控制系统与外界的接口电路,主、从机的通信电路,采用隔离电路(如光电隔离器等)。

(4) 控制系统不要靠近变压器、电机等安放。否则,要加磁屏蔽。

(5) 从程序上增强可靠性。

合实际、设计合理、运行可靠、维护简单、自动化程度高等特点。该系统可应用于各种类型的饲料厂,具有良好的应用与推广前景。

7 参考文献

- [1] 周明德著,《微型计算机硬件软件及其应用》,清华大学出版社,1988年。
- [2] 安维蓉,莫安民著,《微型计算机及其在控制中的应用》,测绘出版社,1986年。
- [3] [日]荒木庸夫著,赵清译,《电子设备的屏蔽设计》,国防工业出版社,1977年。

歼击机空——空格斗导弹攻击模拟系统的过程控制

空军第一航空学院 陈德金 陈德煌 曹永清 (河南信阳 464000)

摘要 本文概述了空——空格斗导弹攻击模拟系统的总体设计思想,着重介绍了计算机用于导弹攻击模拟系统过程控制的方法。并分析了机载使用计算机的干扰与抗干扰问题。

关键词 模拟系统 过程控制 信号处理 包线数据库

1 概述

歼击机空——空格斗导弹攻击模拟系统是一种用于歼击机战术、技术训练的模拟器材。特别适用于歼击机空——空格斗导弹对空中目标攻击的战术、技术训练。其主要技术性能指标如下:

适用高度:600—10000 米

载机速度:220—1200 公里/小时

攻击角度:全方位

攻击角度误差:7°

攻击距离:满足发射包线决定的最大和最小距离
离轴发射角:0°~35°

离轴发射角误差:0.5°

歼击机空——空格斗导弹攻击模拟系统主要由歼击机 1、PL—8 训练弹 2、高度信号采集器 3、速度信号采集器 4、距离信号采集器 5、航向角信号采集器 6、目标截获信号处理器 7、离轴角信号处理器 8、通信装置 9、模/数转换器 10、接口电路 A11、微计算机 12、接口电路 B13、控制盒 14、射击按钮 15、时钟显示器 16、监视/恢复电路 17、音响装置 B18、拉烟器 19、音响装置 A20 和微型打印机 21 等组成。

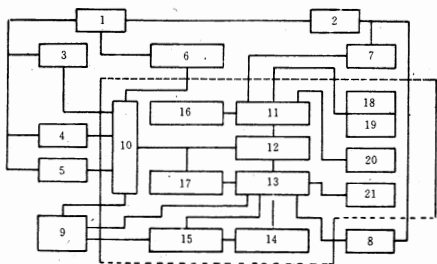


图1 模拟系统原理框图

飞行员驾驶飞机跟踪目标,当导弹截获目标后,按下发射按钮,信号采集系统对攻击机高度、速度、

航向角和目标机航向角及目标距离等信号进行采样;计算机根据采样结果计算得出包线数据库的地址代码;由该地址表取得导弹攻击的最小距离和最大距离,并与目标距离进行比较。当目标距离大于最小距离和小于最大距离,离轴发射角小于30度时,计算机给出“命中”指令,通过通信装置控制目标机产生“中弹”响应。

1.1 高度信号采集

高度信号传感器采用GYQ—4型静压传感器,它用于感应静压的大小。高度信号采集器输出为0~5V的直流信号,输出电压与高度成正比关系。电压为0时对应的高度等于200米,电压为5V时对应的高度等于10400米。

1.2 速度信号采集

速度信号传感器采用GYQ—3型全压传感器感应全压的大小。速度信号采集器输出0~5V的直流信号,电压与速度成正比关系。电压为0时对应的速度等于200公里/小时,电压为5V时对应的速度等于1475公里/小时。

1.3 距离信号采集

目标距离信号从测距雷达引出,经过目标距离信号采集器处理后,输出0~5V的直流信号,电压与目标距离成正比关系。电压为0时对应的距离等于200米,电压为5V时对应的距离等于18050米。

1.4 航向角信号采集

飞机航向角信号从飞机航向指示器引出,经航向角采集器处理后,输出0~5V的直流信号,电压与航向角成正比。当电压为0时对应的航向等于0度,当电压为5V对应的航向等于360度。

1.5 截获信号处理

从PL—8空——空导弹训练弹导引头引出截获音响信号,接入截获信号处理器。当导弹截获到目标时,处理器输出一个衡定的高电平信号。否则,输出低电平信号。

1.6 离轴角信号处理

PL—8空——空导弹的训练弹截获到目标后,

导引头输出截获音响信号的同时,还提供一个离轴角信号,该信号的电压与离轴角成正比。我们将该信号引入离轴角信号处理器,当离轴角小于 30 时,离轴角信号处理器输出为衡定的高电平信号。否则,输出低电平信号。

1.7 信息的无线传送

攻击机与目标机之间的信息传输由专用的通信装置完成,通信装置由专用电台和收/发转换电路组成。主要用于目标机的航向信号向攻击机传输和攻击机的中弹控制指令向目标机传输,采用分时传输方式。

按下射击按钮,收/发转换电路工作,使电台由接收状态转换为发射状态,并将收/发转换指令发射出去,然后恢复到接收状态。目标机电台接收到收/发转换指令后,在收/发转换电路的控制下,转换为发射状态,并持续 1.5 秒钟;在这个时间内,目标机将航向角信号通过电台发射出去。攻击机将目标机的航向角信号接收后,输给模/数转换器。

满足发射条件后,由计算机输出中弹控制指令,通过收/发转换电路,将电台转换为发射状态后,将中弹信号发射出去(一种具有固定振荡频率的正弦波信号),并持续 1.5 秒钟,然后恢复到接收状态。目标机接收到中弹信号后,输给计算机处理,最后产生“中弹”响应。

2 控制装置主要功能

为了真实地模拟导弹发射情况,导弹攻击模拟系统的过程控制应完成以下功能:

(1)能根据导弹发射条件对每次发射导弹的成功与否进行判断;

(2)满足发射条件时,发出中弹指令使目标机音响装置和拉烟器工作。同时记录发射时刻和中弹时刻;

(3)不满足发射条件时,记录发射失败的原因;

(4)能模拟余弹记数,并对余弹灯和射击按钮进行控制。

3 控制装置的构成

控制装置由模/数转换器 10、接口电路 A11、微计算机 12、接口电路 B13、控制盒 14、射击按钮 15、时钟显示器 16、监视/恢复电路 17、音响装置 B18、拉烟器 19、音响装置 A20 和微型打印机 21 等组成(见图 1)。

3.1 微计算机

系统主机由 Z80CPU 中央处理机、4KB 只读存储器 EPROM、2KB 静态读写存储器 RAM、Z80A—PIO 八位可编程并行 I/O 接口、Z80—CTC 可编程计数器等组成。

3.2 接口电路

接口电路由两块接口板 A 和 B 组成,接口板上分别装有稳压电源电路、打印接口电路、波形整型电路、电平变换电路和驱动电路等。

3.3 模/数转换器

模/数转换器采用 8 位的模/数转换器 ADC0808。在本系统中,需要转换的信号有攻击机高度、速度、航向角和目标机航向角及目标距离。我们采用分时采样的方法,对它们进行采样,每 20ms 采样一次。每次采样对象由微计算机选择。

3.4 控制盒

在控制盒上装有复位按钮、时钟校对按钮和充弹按钮及余弹指示灯,它们通过电缆与主机相连。

3.5 显示装置

显示装置由音响装置、拉烟器和时钟显示器及记录仪组成。音响装置为多谐振荡器,输出端接飞行员耳机;拉烟点火电路为单稳态电路;时钟显示器由六位数码管组成。音响装置 A 受微计算机控制,当满足射击条件时,音响装置 A 工作,提示飞行员本次射击成功。音响装置 B 和拉烟装置作为命中效果显示,也受控于微计算机。记录仪采用了 TPUP—40A 微型打印机,它内部带有单片机,打印时不占用主机时间。

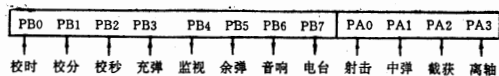


图 2 PIO 接口连接关系

4 控制装置的工作原理

控制装置的工作原理如图 1 所示,机载计算机连续运行时钟程序。按下发射按钮时,若导弹已截获目标,且离轴发射角小于 30 度,微计算机就对本机高度、速度、航向角和目标机的航向角及目标距离等信号进行实时采样,并将目标机航向角和攻击机航向角相减取得进入角;微计算机根据采样结果计算出包线数据库的地址代码;由该地址代码查表取得导弹攻击的最小距离和最大距离,并与目标距离进行比较。当目标距离大于最小距离和小于最大距离和按下射击按钮时间大于 1.5 秒时,计算机给出中弹指令,通过通信电台传输给目标机,同时打印输出本次射击的有关参数,控制音响装置 A 工作,揭示飞行员本次射击成功。

若不满足上述射击条件的一个或多个时,则本次射击失败,微计算机将失败原因通过机载打印机打印记录。经多次射击使余弹数为零时,在微计算机控制下切断余弹灯和射击按钮电源,禁止本机发射

“导弹”。

目标机电台将接收到的中弹信号输给微计算机。经软件滤波确认为真实信号后,微计算机给出指令触发音响装置A和拉烟器工作,同时打印输出本次中弹的有关参数,并禁止本机继续发射“导弹”。

4.1 发射包线数据库的建立

发射包线数据库是该模拟系统的重要组成部分。它是根据空——空格斗导弹的性能及其使用要求给出的典型发射包线建立的[1][2],由导弹的发射包线图可知,飞机在不同的高度、不同的速度和不同的进入角情况下发射导弹,其射击的最大距离和最小距离是不一样的。为此,我们将高度从200米开始取点,每增加680米取一点,共取16个点;速度从200公里/小时开始取点,每增加85公里/小时取一个点,共取16个点;进入角从0度开始取点,每增加11.61度取一点,共取32个点分别进行查图计算,得出各种情况下的最小射击距离和最大射击距离。然后,将所有数据按下式转换成数字量后存入包线数据库。包线数据库的结构如图3所示。

最小距离和最大距离的填表数据:

$S(\text{数字量}) = S(\text{模拟量}) \div 70 \text{ 米}$ 取整数后换算成二进制代码填表。上式中70米为转换成数字量的分辨率。

A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	最小射击距离
0	0	0	0	0	0	0	0	0	0	0	0	0	0	—高度200米,速度200公里/小时,进入角0°
0	0	0	0	0	0	0	0	0	0	1	1	1	1	—高度200米,速度200公里/小时,进入角360°
0	1	0	0	0	0	0	0	0	1	0	0	0	0	—高度200米,速度280公里/小时,进入角0°
0	0	0	0	1	0	0	0	0	0	0	0	0	0	—高度200米,速度200公里/小时,进入角0°
0	1	1	1	1	1	1	1	1	1	1	1	1	1	—高度10400米,速度1475公里/小时,进入角360°
1	0	0	0	0	0	0	0	0	0	0	0	0	0	—高度200米,速度200公里/小时,进入角0°
1	1	1	1	1	1	1	1	1	1	1	1	1	1	—高度10400米,速度1475公里/小时,进入角360°

图3 包线数据库结构

4.2 模/数转换

高度从200米~10400米,对应模拟量0~5V,转换后数字量为0~255,分辨率为40米。速度从200公里/小时~1475公里/小时,对应模拟量0~5V,转换后数字量为0~255,分辨率为5公里/小时。进入角从0度~360度,对应模拟量0~5V,转换后数字量为0~255,分辨率为1.41度。距离从200米~18050米,对应模拟量0~5V,转换后数字量为0~255,分辨率为70米。

根据飞机高度H产生地址的A12、A11、A10和A9的编码,其计算方法如下:

当 $n \times 680 - 340 + 200 < H < n \times 680 + 340 + 200$ 时,A12、A11、A10和A9的二进制代码对应十进制数为n。其中 $n = 0 \ 1 \ 2 \dots 15$ 。

根据飞机速度V产生地址的A8、A7、A6和A5的编码,其计算方法如下:

当 $n \times 85 - 42.5 + 200 < V < n \times 85 + 42.5 + 200$ 时,A8、A7、A6和A5的二进制代码对应十进制数为n。其中 $n = 0 \ 1 \ 2 \dots 15$ 。

根据进入角Q产生地址的A4、A3、A2、A1和A0的编码,其计算方法如下:

当 $n \times 11.61 - 5.8 < Q < n \times 11.61 + 5.8$ 时,A4、A3、A2、A1和A0的二进制代码对应十进制数为n。其中 $n = 0 \ 1 \ 2 \dots 31$ 。

4.3 数字滤波

在按下射击按钮后的1.5秒内,模/数转换器在微计算机的控制下,分别对攻击机高度、速度、航向角和目标机航向角及目标距离进行25次采样,每采样5次进行一次滤波。滤波时,将5个采样值中的最小值和最大值去掉,然后求平均。

4.4 余弹记数

为了在训练时给飞行员提供更多的练习机会,我们在模拟系统中设置了一次充弹3发、9发和255发三种形式,可在地面通过开关选择不同的装弹数。每次射击后的余弹数为射击前的余弹数减1。

5 软件的组成及工作原理

系统软件采用汇编语言编写,并编制成模块式结构。整个程序工作只是一系列子程序的调用,其中每个子程序又进一步调用一系列子程序,而这些子程序中的某些子程序又由另外一些更深的子程序组成。它们主要是主程序模块、实时时钟模块、射击/中弹模块、模数转换模块、数字滤波模块、地址代码模块、攻击条件模块、余弹记数模块、打印模块和代码转换模块等,主要程序流程图如图4、图5和图6所示。

6 系统的干扰与抗干扰

6.1 干扰类型

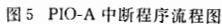
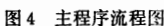
飞机上的许多设备对机载计算机来说都是干扰源,如发动机启动系统、飞机配电系统、电动机、变流机和电伺服控制等设备都会产生干扰信号、计算机在这种恶劣的环境下工作,很容易受到干扰,出现的问题主要是假信号响应和软件混乱。假信号响应是干扰信号从输入端进入系统引起误动作,本系统最容易受干扰的是中弹输入信号。软件混乱通常表现为信息丢失或程序执行中跳到了存储器某个随机的位置,使程序不能正常运行。

6.2 抗干扰措施

① 假信号的滤波

本系统除在硬件上采用回路区极小化、信号滤波和电容屏蔽、电感屏蔽、RF屏蔽等通常的技术措

② 软件混乱的恢复



[2] 王长胤.《微计算机原理及其应用》武汉:科学技术出版社,1984;4

• 电话:5582666 转 2055

饲料生产中的微机控制系统

郑州工学院计算机与自动化系 赵文安 (450002)

摘要 本文介绍了饲料厂微机控制系统软件、硬件的设计思想与实现方法。整个计算机的硬件系统和软件系统均采用模块化设计方法设计。根据笔者调试和运行经验,文中还提出了为保证计算机安全、可靠运行应采取的抗干扰措施。该系统经过两年多的现场运行取得了良好的效果,1993年元月通过了省级鉴定,受到了专家们的好评。

关键词 微机控制 饲料加工

1 前言

混合饲料的一般生产方法是人工控制饲料加工设备,用机械磅秤对各原料进行称量,经过定时混合后包装入库。由于称量过程中物料的变化引起的冲击、振动等干扰使得称量精度难以保证,因而影响了产品质量。为了改变饲料行业这种落后的混合饲料加工方法,我们接受了河南省计经委下达的饲料厂通用的微机控制系统科研项目。经过近两年的研制与开发获得了成功。本系统能够按照各种混合饲料配方的要求,自动、精确地控制各个饲料加工设备的运转,从而生产出完全符合设计要求的、高质量的饲料产品。以下详细介绍该系统硬件、软件的设计思想与实现方法。

2 系统硬件设计

系统硬件组成如图1所示,其中:

(1)原料仓,共12个,各仓上部的进料门和下端的喂料器均由微机控制。各仓上、下喂料器信号均由微机检测。

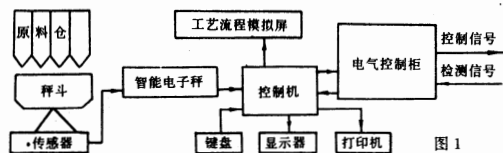


图1

(2)工艺流程模拟屏,通过它动态模拟显示生产工艺流程。

(3)智能电子秤,其量程为100至650公斤。该秤的主要部件有秤斗、传感器、仪用放大器、采样保持器、A/D变换器、STD总线底板、Z80CPU板、Z80PIO板、显示板和电源板。

(4)控制机,选用国际流行的STD总线结构的工业控制微机。该机种为模块化结构,可以根据不同类型饲料厂的需要选用相应的模板组合。在我们设计的系统中选用的主要模板为:STD总线板、Z80CPU板、不断电存储板、光电隔离板、键盘与显

示器接口板、打印机接口板和电源板。为控制机配备的外设有显示器、打印机和键盘。

3 系统软件设计

整个系统软件分为生产控制、配方及参数处理、报表打印三大部分。各部分均采用模块化设计方法设计。所有软件均用Z80汇编语言编写。系统软件的功能模块图如图2所示。各个模块的功能及实现方法叙述如下。

3.1 生产控制软件的设计

在生产控制中,配料、混合及下料为并发过程。为了使控制系统的响应速度满足生产的要求,我们对控制软件采用实时多任务的设计方法,将配料、混合、下料以及参数打印、累计等工作作为各个任务由时钟管理模块进行调度。

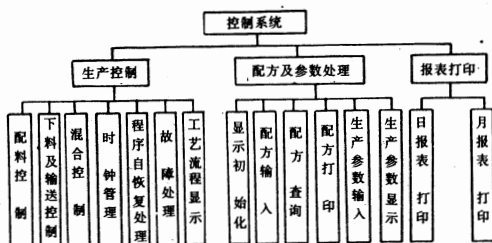


图2

3.1.1 配料控制模块

配料过程是按照配方中给定的各原料的重量控制各原料仓喂料器电机的启、停,使各原料分时进入秤斗,分别进行称量。我们采用的控制算法如下:

对于第m种原料第n秤有:

$$\sum W_p(m) = \sum_{i=1}^n W_p(m, i) \quad \dots\dots\dots (1)$$

$$\sum W_t(m) = \sum_{i=1}^n W_t(m, i) \quad \dots\dots\dots (2)$$

式中: $W_p(m, i)$ 为第m种原料第i秤的实际称重。

$W_t(m, i)$ 为第m种原料第i秤的理论重量。

$\sum W_p(m)$ 为第m种原料实际累计称重。

$\sum W_t(m)$ 为第 m 种原料理论累计重量。

由式(1)和式(2)可得第 m 种原料的实际累计误差 $AE(m)$ 为:

$$AE(m) = \sum W_p(m) - \sum W_t(m) \dots\dots\dots (3)$$

由式(2)和(3)得:

$$W't(m, n+1) = W_t(m, n+1) - K \cdot AE(m) \dots\dots\dots (4)$$

式中:

$W't(m, n+1)$ 为第 $n+1$ 秤, 第 m 种原料校正后的理想称重。

K 为校正系数, 其值由实际运行经验取得。第 m 种原料第 n 秤的落差重量 $Wf(m, n)$ 为:

$$Wf(m, n) = W_p(m, n) - W_s(m, n) \dots\dots\dots (5)$$

式中:

$W_p(m, n)$ 为秤稳定后第 m 种原料第 n 秤称得的实际重量。

$W_s(m, n)$ 为停止喂料时第 m 种原料第 n 秤称得的实际重量。

理想情况下:

$$W_t(m, n) - E \leq W_p(m, n) \leq W_t(m, n) + E \dots\dots\dots (6)$$

即:

$$W_t(m, n) - E \leq W_s(m, n) + Wf(m, n) \leq W_t(m, n) + E \dots\dots\dots (7)$$

式中, E 为允许误差重量, 其值为:

$$E = W_t(m, n) \times 0.2\% \dots\dots\dots (8)$$

式中常数 0.2% 为给定的称量误差百分值即控制精度, 由用户在设计系统前提供。

由于料仓内原料密度、料位以及机械振动等因素对落差重量 $Wf(m, n)$ 的影响是随时间而变的所以使得停止喂料时的称重:

$$W_s(m, n) = W_p(m, n) - Wf(m, n) \dots\dots\dots (9)$$

也为时变函数。为了便于计算, 对于第 m 种原料每次用其前三秤的平均落差重量作为该原料第 $n+1$ 秤的落差重量的估计值。其计算式为:

$$\bar{W}f = (\sum_{i=n-2}^n Wf(m, i)) / 3 \dots\dots\dots (10)$$

计算机停止喂料时的控制重量为:

$$W_s(m, n+1) = W't(m, n+1) - \bar{W}f \dots\dots\dots (11)$$

对于每一种原料我们均用式(11)计算其相应的停止喂料时的控制重量。由此可见, 该控制算法既考虑到了每种原料每秤落差重量的影响, 又考虑到了每种原料以前各秤累计误差的影响。所以这种控制算法可使连续生产时自动寻求补偿重量的最优值并保证各秤的综合指标最优。

为了保证各原料重量达到给定精度要求除按式(11)进行控制外还要将流入秤斗内的稳定后的实际重量进行精度判断, 对欠缺部分用自动点动加料的方法进行微调。

3.1.2 下料及输送控制模块

下料及输送控制为逻辑时序控制。当一秤斗料配制完成后即进入下料输送控制。这时, 首先检查混合机是否完成上一斗料的混合并输送完毕, 也就是要检查混合机是否处于就绪状态。如果混合机处于就绪状态则检查半成品传送绞龙和半成品提升机是否在运行。如果未运行则启动其运行并延长一段时间后开启秤斗门, 待下完料后, 关闭秤斗门即完成下料控制。

3.1.3 混合控制模块

混合控制和配料控制、下料及输送控制为并行控制过程。当一秤料经下料、提升进入混合机后开始为混合机定时, 当达到额定的混合时间后, 首先检查成品仓是否满, 若未满则打开混合机下料门并延时至仓门开到位后继续延迟一段下料时间。放完料后开始关混合机下料门, 关到位后, 混合机即为就绪状态。此时, 混合机为接受下一秤斗料作好了准备。如果检测到成品仓已满或者混合机下料门未到位等情况则报警以提醒操作人员采取相应的处理措施。

3.1.4 时钟管理模块

时钟管理模块主要是为控制系统提供时标和闹钟等调度依据的管理程序。我们将其设计为一个中断服务程序, 利用 CPU 上的一片 Z80CTC 作为定时中断源, 采用中断方式 2 工作方式并将其作为最高中断级。时钟管理模块对它调度的各个任务中的子任务的状态及其时序关系通过总共 21 个定时闹钟进行管理。

3.1.5 程序自恢复处理模块

在生产控制过程中, 微机受到干扰以后, 有时会将正确的源语句误认为语法有错或因对某一控制条件误判而进入等待状态, 即出现所谓的“死机”现象。为了及时地发现这种“死机”现象, 我们利用系统时钟对控制程序进行了有规律的测试。为此, 需要在内存中设置一个地址单元, 存放程序运行是否正常的标志。程序在正常运行时, 每隔一定时间向标志单元送入表示程序正常运行的标志信息, 而在时钟中断程序中, 从该单元内读取标志信息。根据读取的内容, CPU 就可以知道程序运行是否正常。如果程序运行正常, 则时钟中断程序清除标志信息, 然后, 正常返回到断点。如果程序运行不正常可再等几个系统时钟周期, 继续测试几次(例如 5 次)。若出现了程序运行正常的标志, 则可按正常情况处理, 否则, 就转向重新启动入口, 从而实现程序的自动恢复。

3.1.6 故障处理模块

在生产控制过程中, 有时会出现一些异常情况, 如秤斗和混合机的下料门关或开不到位等情况, 这时通过该模块进行报警显示并发出声响以提醒操作人员进行相应的处理。

3.1.7 工艺流程显示模块

该模块用以在显示器上动态、形象地模拟显示整个生产工艺流程以便于操作人员直观地了解与掌

握生产情况。

3.2 配方及参数处理软件的设计

当前,一个工业控制机仅有良好的控制功能是远远不够的,用于饲料生产的工业控制机还应该具有灵活、方便的配方及工作参数的输入、修改、查询和打印等功能。对于用户所需要的各种操作均应由层层汉字菜单引导。各种显示、打印均采用中文形式。这些要求对于经过汉化的通用微机来说是不难作到的,但对于内存较小,无任何汉化软件和汉卡的工业控制机来说就需要花费一定的时间精心设计。这部分软件我们主要设计了系统显示初始化模块、配方输入模块、配方查询模块、配方打印模块、生产参数输入模块、生产参数显示模块。另外,编制了公用子程序模块,自造了一个能容纳 200 多个汉字和字符的小型字库,还建立了一个能容纳 50 个常用配方的固化配方库。各模块的主要功能简介如下:

3.2.1 显示初始化模块

该模块使显示器处于图形工作方式并作些初始化工作用以显示汉字和各种字符。该模块还将屏幕分区用以多窗口显示。

3.2.2 配方输入模块

该模块对用户从键盘上输入的配方中的各原料名汉字代码与重量数字分别处理,然后,存入内存中的不掉电区域以备生产时调用。

3.2.3 配方查询模块

该模块根据用户在键盘上输入的配方号码将配方库中相应的配方以汉字方式显示在屏幕上。可以显示单个配方,也可以连续显示库中所有配方。

3.2.4 配方打印模块

该模块可以把配方库中的所有配方依次打印输出,也可以按用户输入的配方号逐个打印输出。

3.2.5 生产参数输入模块

通过该模块,用户可以从键盘上输入开机时间、混合时间、原料名和仓号对应表、每斗料额定重量等生产参数。计算机将把这些参数存储到内存的不掉电存储区供生产控制时调用。除开机时间外,如果下一班组使用的参数和上一班组相同,就不必重新输入了,生产时计算机可直接调用以前保存的工作参数。

3.2.6 生产参数显示模块

该模块根据用户在键盘上输入的各种参数的代码将相应的参数显示在屏幕上。

3.3 报表打印软件的设计

这部分软件有如下两个功能模块:

3.3.1 日报表打印模块

开机后,该模块首先将配方中各原料名及其对应的理论重量值送到打印机上打印输出。正式生产时,每一秤料配制结束后,该模块便将该秤斗中各原料的实际重量在打印机上用表格的形式打印输出。

每班生产结束时,此模块便将该班的日生产量,各原料的实际消耗量及累计误差重量等以表格形式打印输出。

3.3.2 月报表打印模块

该模块将各个生产班组一个月的总产量、各原料的实际消耗量及各原料的累计误差等进行统计、汇总并以表格的形式打印输出。

4 抗干扰措施

一般来说,工业现场对计算机的干扰是比较强的,有时会使计算机执行错误或“死机”。为了使控制机可靠、稳定地运行,我们采取了如下抗干扰措施。

(1)控制机与现场的所有控制线及信号线均通过光电隔离接口板连接,这样可以大大地减少外界对控制机的干扰。

(2)电网的瞬变过程是经常发生的,其干扰往往是通过传导和幅射两种形式传给主机。为了防止这些干扰,我们在电网和控制机之间串联了交流稳压器、隔离变压器、低通滤波器和噪声滤波器。

(3)为了防止空间电磁干扰,我们将装有控制机的金属机箱置于有屏蔽作用的生产工艺流程模拟屏内,从而使控制机得到了双重屏蔽。另外,还要使控制机接地点悬空以和系统地隔离。

(4)在计算机所控制的每个继电器的线圈上并联储适当的电阻、电容以消除和减少继电器动作时产生的电磁波干扰。

(5)为了防止偶然出现的强干扰造成“死机”而中断生产控制,我们为控制系统设计了一个如前所述的自恢复程序模块。当控制机受到干扰而出现“死机”现象时,该程序能够引导控制机恢复正常运行。

(6)为了防止脉冲信号、工频信号和秤斗的周期性振动信号对电子秤称量的干扰,我们在软件上增加了数字滤波的功能。

由于在软件和硬件方面采用了上述多种抗干扰措施,所以整个控制系统运行稳定、可靠。

5 系统主要性能及指标

(1)配料精度为 0.2%。

(2)具有手动、自动切换功能。

(3)具有掉电保护功能,恢复供电后能保证生产的连续性。

(4)具有汉字提示和较强的人、机对话功能。各种操作均由层层菜单引导,使用简单、直观、方便。

(5)能动态模拟显示生产工艺流程。

(6)具有实时数据处理功能,如定时自动显示、打印、异常情况报警等。

(7)能自动校零、误差修正、流料落差补偿。

(8)能固化 50 个常用配方(根据需要可增加),能随机存储 10 个配方(根据需要可增加)。

(9)生产过程中,用户可根据需要随时改换生产

(下转 48 页)

计算机耐故障技术的探讨

西南农业大学计算中心 王玉昆 杨国才 (四川重庆 630716)

摘要 本文以研究抗故障计算机为题,讨论了计算机系统的耐故障技术,叙述了耐故障技术的基础,从而引入了冗余性概念,同时还提出了耐故障技术的几种方法,并对几种方法进行了分析比较。

关键词 计算机 故障 耐故障技术 冗余性

1 前言

随着科学技术的飞速发展,计算机技术的不断提高,计算机应用系统已渗透到社会的各个领域,如银行的联机系统、铁路和航空部门的交通系统、通讯系统以及服务部门的计算机系统等。因此计算机与人们的日常生活有着千丝万缕的联系,不用说,这些系统的核心部分是计算机。通常人们总认为这些系统是由计算机控制,所以,是不会出错的,是绝对安全的。实际上,这类系统也会出错。如果这类系统一旦发生故障,就会造成巨大的经济损失,甚至会出现人员伤亡的重大事故,也就是说,切不可忘记计算机也会出现故障。例如:日本分别在昭和 36 年和 59 年,某银行的联机系统发生故障,某电话局的交换机发生故障以及电话电缆起火,致使银行联机网络不通。而且,还使证券业务、行政活动、交通管理等方面发生了恐慌,至今还记忆犹新。因此,对于处理价值高的信息处理系统人们尤为关心,要求系统具有尽可能高的可靠性和安全性。

2 系统的可靠性和安全性

人们经常使用故障、错误、障碍等这类相似的词汇来评价一系统的工作状态,其各自的含意可作这样的定义。

故障(fault)是指系统的构成部件处于丧失其功能的状态。

错误(error)是指系统规定的内容或理论的正确状态与计算或测试的结果不一致。

障碍(failure)是指系统失去了规定的功能。

可见,系统因故障存在,出现错误,而引起系统障碍。

评价系统的可靠度 $R(t)$ 时,采用故障率和平均故障寿命(MTTF; Mean Time To Failure)以及平均故障间隔(MTBF; Mean Time Between Failure)。所谓的设备故障率是指,设备在时刻 t 和时刻 $t+\Delta t$ 之间故障发生的机率。例如,观察某设备使用过程中的故障率,其最初阶段的故障多(称为初期故障率),

以后,故障逐渐下降,呈现为稳定期(称为偶发故障期)。若继续使用,因机器磨损使故障率再次上升(称为磨损故障期)。但是,IC 集成电路不会磨损。将以上过程用图表示时,尤如西式浴盆的形状,因此称之为故障率曲线。

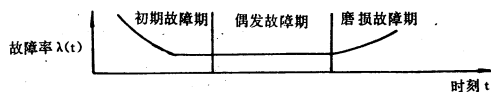


图1 故障率的浴盆式曲线

由上述可见,故障率虽然是时间的函数,但实际使用时,把故障偶发期常作为常数来看待。所以,故障率是表示单位时间里故障发生的次数,特地将每 10^6 小时发生一次故障的机率称作一非特(fit),并将其作为故障率的单位。可见,该值越小其可靠性就越高。

可靠度 $R(t)$ 是时间的函数,当 $t=0$ 时, $R(t)=1$, 当 $t=\infty$ 时, $R(\infty)=0$ 。而故障率 $\lambda(t)$ 是指 $R(t)$ 在一段时间范围到下一个单位时间内出现故障的机率。于是,可用下式予以定义。

$$\lambda(t) = \{-d(R(t))/dt\}/R(t) \quad (1)$$

λ 为常数时,可求得 $R(t)$, 即为

$$R(t) = e^{-\lambda t} \quad (2)$$

MTTF 可用下式求得,即为

$$MTTF = \int_0^{\infty} R(t) dt = 1/\lambda \quad (3)$$

1945 年,世界上最早制造的电子计算机(ENIAC),使用了 18800 个电子管,它仅用 30 秒的时间就完成了子弹在 1 分钟内所飞行的弹道计算。当时其计算速度比子弹还快而震惊世界。但是 ENIAC 的平均故障间隔约为 1 小时。

表 1、电子部件故障率的大概数据

SST, MSI(小、中型集成电路)	50—70 非特
LSI (大规模集成电路)	150—300 非特
128K—256K 的 EPROM	150—200 非特
256K—1M 的 DRAM	100—200 非特
固定电阻	

由(3)求得,MTTF是故障率的倒数。例如:100非特部件的MTTF为 10^7 小时,这表明可用1100年。然而,通常设备使用的部件很多,所以,这样就使MTTF大大变小。比如,对于使用了1000个100非特的部件所构成的设备,若将其中某一部件发生的故障,看成是整个设备发生故障的话,那么,整个设备的故障率便成为 $100 \times 1000 = 10^5$ 非特,MTTF为 10^4 小时(约1.4年)。因此,要使由许多部件组成的计算机系统的故障率变小,重要的是要提高各个部件的可靠性,以及减少部件数量。

3 耐故障技术

计算机系统丧失规定的功能而不能正常工作的原因之一是故障的发生,构成具有高可靠性系统的技术,应包括避免故障发生(fault avoidance)和耐故障技术(fault tolerance)技术。

避免故障的出现是第二次世界大战以后,通过引入质量管理技术而发展起来的一门重要技术。该技术应包括:(1)要充分重视构成系统部件的制造工序的质量管理;(2)尽可能地减少构成部件的数量;(3)要完善系统的工作环境,以尽量避免故障的发生。

耐故障技术指假定构成系统的部件有故障问题。当故障发生时:(1)尽可能不影响系统的工作状态,此外(2)即使影响该系统,但造成的影响很轻。或(3)即使系统发生故障而自动恢复时,应避免向外部发出危险信号。可见,这样的系统必须具有故障检测和恢复的功能。为此,系统应具有冗余性。

冗余性包括硬件冗余性、软件冗余性和信息冗余性的技术,它们之间有着紧密关系。

3.1 硬件的冗余性

3.1.1 多重系统

使用两台计算机可使系统的可靠性和可用性得以提高,当前,已进入实用阶段的有并行和备用系统。而且,将还有计算机的三重化系统。

a、并行系统

如图2所示那样的系统,是采用两台计算机同时进行某一项工作,它是边进行处理边进行对应检查结果的系统。当两台计算机的处理结果一致时输出其结果,否则,就认为其中某一计算机发生了故障,并对故障进行诊断、检查,将出现故障的那一台计算机从系统断开,只让正常工作的那台计算机进行处理。这样,即使有故障出现也不会中断系统运行,以保证了系统的正常工作。

b、备用系统

当不要求并行系统那么高的可靠性时,通常采用如图3所示的联机处理,另一台计算机进行别的

工作(批处理)。这样,即使正在工作的那台计算机出现故障时,便可用备用的那台计算机进行替换,以保证系统的正常工作。

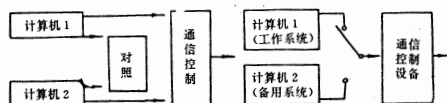


图2 并行系统

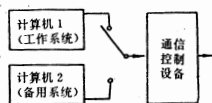


图3 备用系统

该系统与并行系统相比,系统的可靠性变高,而且经济性好,所以,在联机系统中得到了广泛的应用。

如果两台计算机的可靠性R相等,错误检测以及替换设备的可靠度为1时,系统的可靠度 R_D 为:

$$R_D = 2R - R^2 \quad (4)$$

因此,当 $0 \leq R \leq 1$ 时, $R_D \geq R$

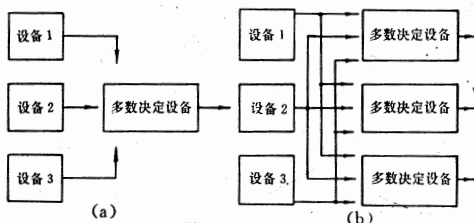


图4 TMR系统

银行的联机系统、列车和飞机的预订票系统等就是采用这种系统的例子。

C、三工结构(三重化)系统

该系统是采用三台功能相同的计算机同时进行工作,并能根据输出的多数决定最终的输出结果。因而,大大地提高了系统的可靠性,即为有名的TMR(Triple Modular Redundancy)系统

现假设如图4(a)所示各设备的可靠度为R,当多数设备所决定的可靠度为1,系统的可靠度为 R_T 可用下式求得。

$$\begin{aligned} R_T &= (\text{所有设备无故障时的概率}) + (\text{任何两台设备无故障时的概率}) \\ &= R^3 + 3R^2(1-R) \\ &= 3R^2 - 2R^3 \end{aligned} \quad (5)$$

因此,如果 $R > 1/2$ 时, $R_T > R$,这样与一台设备的R相比,TMR的 R_T 高。相反,如果将 $R < 1/2$ 时的极差的设备三重化也就毫无意义了。假定设备的可靠度为(2)式的 $R = e^{-\lambda T}$ 的话,那么,TMR的MTTF便为

$$\int_0^{\infty} R_T dt = 5/(6\lambda) \quad (6)$$

若 TMR 系统中的多数设备出现故障时,将输出错误信息。为了防止这种情况的发生,可考虑图 4(b)所示那样的将输出线路三重化的方案。据说对宇宙飞船的控制就是使用 4 台计算机,并采用三对一的多数决定方式。最近,已向市场推出了将整个系统二重化的 24 小时内能连续工作的计算机系统。

3.2 信息冗余性

通常信息冗余性包括奇偶校验码、错误修正码、自动校验、故障诊断和安全装置技术。

奇偶校验码:这一技术已在信息处理中得到广泛的应用,由于它只能检测出发生的错误,并不能反映出正确的信息是什么。因此,需要一个当检测出错误时,将它能修正为正确信息的代码。

错误修正码:奇偶校验码中引入了 1 位的冗余位来对错误进行检测,通常对信息可按照汉明码(一种错误检验码)来修正错误。如检验出错误后,就能将其修正为原来正确的信息,这样的错误修正码在存储器中得到了实际的应用。最近在日本生产的计算机存储器就具有单一错误修正和双重错误检测的功能。

自动校验:使用错误修正码就能对错误进行检测和修正。这时,必须对检测错误线路本身进行检测,这种检测线路自身的线路称为自动校验线路。

故障诊断:制造设备时,有必要检查产品是否符合规格要求,当设备出现故障时,还有必要检查是哪里发生了故障。即采用“在设备输入时,加入检查用的信息来(称为检查输入)检查相应输出信息”的方法。

检查输入的生成是根据被检查的设备有无回路来决定,但二者的难易程度是大不相同的。为简单起见,可把在 N 根输入线的电路叫做组合回路。这样,可能输入的组合有 2^n 种。其检查的集合大小为 2^n , n 愈小就实用。但是,当 $n=10$ 时, $2^{10}=1024$; $n=20$ 时, $2^{20} \approx 1.05 \times 10^6$,如增加一个检查输入时,检查输出所需时间为 1μ 秒,当 $n=20$ 时,检查的时间约为 1 秒,但是,当 $n=30$ 时, $2^{30} \approx 1.07 \times 10^9$,其检查的时间就需要 1071 秒(17 分)。

假定信号线将检查对象的故障固定为 0 或 1 时,就没有必要对所有的 2^n 的输入进行检查。通常这是由计算机预先求出必要的检查输入。不过,近年来由于电路集成化程度的不断提高,使要生成这样的检查输入变得愈加困难。这种方法也用于对存储器的检查工作,但如何解决容量大的存储器以及多部件的检查时间就成为突出问题。最近在日本已研制出将检查线路组装到存储器内部,而成为让存储器自己检查自己的存储器。

安全装置:安全装置与所叙述的提高其工作效率的耐故障技术是不同的。安全装置是指:在系统出现故障时,能保证系统的安全。在日本,正将此项技术应用于新干线、大型客车、原子能发电、机器人控制、AI 等方面。最近,日本正在研制称之为安全 IC 的 IC 系统,并将这种安全 IC 设计为 0,一旦出现故障就必然输出 0 的信息。

3.3 软件冗余性

计算机系统是由硬件和软件组成的,因此,可将软件的冗余性应用到耐故障技术之中,作为软件冗余的典型例是采用副本备份(Back up copy)。因而成为使用时间的冗余性。

3.3.1 有效性检查

对处理过程中,得到的中间结果或最后结果,采用插入其它软件的办法以实现在一定范围内进行检查的目的。这样的办法常在控制系统中采用。

3.3.2 自我检查

通常在处理过程中,运用周期性进行的自我检查软件,以实现对工作状态进行检查。如能设计合适的监视程序,一旦出现有不合适的时钟复位时,就进行中断。状态检查是指通过对所定义的数据进行预定的处理,并将其结果与存储的正确结果进行比较。采用这样的检查,不仅能监视硬件,而且对于程序的失控也能进行监视。

3.3.3 多程序的使用

据统计,在计算机出现故障的原因中,软件原因占 65% 左右。因此,至关重要的是不出现程序的逻辑错误。这一点是与避免故障技术相对应的。然而,通常要判断程序是否出现逻辑错误却是困难的。因此,一些先进国家,目前正在研究解决这类逻辑错误的方法,其中有一种方法是很有名的,即为 N 方案程序设计(N -Version programming)方法。这种方案采用了如下方法,即准备进行相同处理的程序 N 个(N 个人,各自使用不同的算法、语言、编译程序),通过将所得的结果以少数服从多数的形式来决定其最后的结果。

4 结束语

计算机在我国已广泛应用于各行各业,并取得了巨大的社会效益。随着我国各种信息系统的开发与应用,人们都希望计算机系统具有越来越高的安全性和可靠性。为此,本文对计算机系统的耐故障技术进行了探讨,希望能起到抛砖引玉的作用。

5 参考文献

- [1]【日】相沉恒博,王玉昆等编译,《计算机与人工智能》1992 年
- [2]张福炎等编著,《微型计算机 IBM PC 的原理与应用》1986 年
- [3]【美】A. B. 肯泽德著,潘元年等译,《数据系统及其管理》1987 年

M1724 打印机故障维修六例

黑龙江大学

陈永红 穆大明 (哈尔滨 150080)

故障现象 1:

M1724 打印机的皮带传导轮损坏。

故障的分析与排除:

M1724 打印机皮带传导轮,主要是传导皮带带动打印头左右来回运转。在该打印机的左端有两个外径 12MM,高 7MM 的工字型硬礁轮,以轴心间距为 15MM 的位置固定在活动金属托板上。传导轮的内孔与轴中部各有一个凹弧型圆环槽,内套钢珠,机械原理同轴承。由于打印机长时间连续运转,轴轮磨损造成钢珠套损坏,钢珠脱落,即轴轮间距增大,打印头从车只要运转,皮带就从硬礁轮上部跳出来,脱轴或掉带,而使打印机无法使用。由于皮带传导轮不一定能买到,下面给出两种修复皮带传导轮的简便方法:

(1)皮带传导轮很坚硬,而轮的内孔又很光滑,与轴直接接触时转动摩擦力很小,因此旋转的灵活性不太受影响。采用一只弹簧卡环,用钳子夹开,把它整形,压平,然后卡入皮带传导轮轴中部凹弧型圆环槽,再用钳子将弹簧卡夹圆,并将凹弧型圆环槽加点黄油,最后将硬礁轮套入传导轮轴。由于弹簧卡有弹性,硬礁轮套入时,必须歪斜一点,使一点劲,就可使硬礁轮的内孔套上弹簧卡环,这样弹簧卡环代替了原生的钢珠套,硬礁轮不会上下跳动,皮带也不脱落,使用正常。

(2)采用一块 2MM 的铜片(或尺寸同轴一样)将传动轴包住(用电烙铁焊死),铜片成圆形,要比硬礁轮高出 6MM,然后将硬礁轮套入轴中,再将固定死的圆形铜片,用锯条锯两条槽,用钳子将铜片弯过来,使硬礁轮往上不能跳动,致使皮带不会脱落。

以上方法经过长时间的运行,效果很好,但要经常将硬礁轮轴中加点黄油或机油,保持轴轮之间润滑,使传导轮减小摩擦力。旋转更加灵活。

故障现象 2:

M1724 打印机驱动电路外部特性变坏。

故障的分析与排除:

M1724 打印机,打印时挂色带,检查后发现打印针断了 2 根,色带严重破损,更换打印针和色带后进行试打,仍然挂色带。这就排除了是色带破损而引起的挂带故障,而后对针驱动电路进行检查。经查,针驱动电路的各部分逻辑功能正常(逻辑笔

测试)。后断电用万用表测试针驱动电路 #C4 (M5248)各脚对地电阻,发现阻值只有 600 Ω 左右,而正常时对地电阻有 180k 左右,故判断 M5248 芯片损坏。

数字电路的损坏,一般分为逻辑功能损坏和外部特性损坏两种。一般地讲,逻辑功能一旦损坏,该芯片也就报废了。但外部特性变坏的数字电路经过技术处理后,仍可正常使用。像 M5248 这样的芯片在市场上较难买到,因此,对逻辑功能完好,外部特性变坏的针驱动芯片的修复,更有必要。

由针驱动电路和 M5248 内部等效电路图可知:当驱动电路不工作时, #C3 使 HD18、HD20、HD22、HD24 为高电平,而使 #C4(M5248)截止。当 #C3(M60005-0217FP)输出一个低电平使 HD18、HD20、HD22、HD24 有效时,由 +5V 通过 RA8 给 M5248 提供偏流而使其工作,当其外部特性变坏时,将使 M5248 处于临界状态,当 HD18、HD20、HD22、HD24 信号由低变高而还没有变低时, M5248 就开始饱和导通了,反之,当 HD18、HD20、HD22、HD24 信号由低变高而还没有变高时, M5248 也饱和导通了,故该片对应的针 18、20、22、24 一直出在外面,故而挂坏色带和折断打印针。查到故障原因后,将 RA8(10K)电阻排去掉,使 M5248 脱离临界状态,而仅用 #C3PC 口输出的信号使其工作,经开机试打,一切正常。

故障现象 3:

接通电源后,字车不能复位,CSF 灯闪烁,不能自检打印。

故障的分析与排除:

在 M1724 打印机中,开机后首先做回车初始化,其中包括对各个基本电路的测试。若初始化正常,则字车复位后,打印机等待下一步操作。若初始化不正常,则经面板不同指示的闪烁来表示发生故障的局部电路。CSF 灯闪烁,说明是电机驱动电路出了故障。电机驱动电路包括字车电机驱动电路和走纸电机驱动电路。由于字车不能正常复位,故可以认为是字车电机及其驱动电路的故障。M1724 打印机字车电机驱动电路以 2-2 相激励方式驱动字车步进电机。字车电机是 4 相 48 极永磁步进电机。该电机共有 6 条引出线,其中 5-1、5-2、6-3、6-4 引线电阻为 19.4 Ω ;1-2、3-4 引线电阻为 38.8 Ω ;其余各引线电阻应为 ∞ 。CRCOM 是字车电机供电允许信号,由 CPU M50734FP 的 P30 口输出。

当 CRCOM 信号为低电平时,经 #C174LS05 反相, #A2TD62504P 放大,使 QA2 TH3J10 中有关的复合管导通,字车电机的公共供电端 CRMCOMA、CRMCOMB 才能得到 +35V 的驱动电压。CR1—CR4 是字车电机四相的驱动信号,是由 CPU 的 P20—P23 口输出的。在 CRCOM 为低电平时(驱动状态),CR1—CR4 经 #D1 74LS05 反相,激励 QA1 TH3L20 中的各复合管,将驱动脉冲加至字车驱动电机 CRM1—CRM4。

为了减小电源不稳定引起的转矩偏差,字车电机驱动电路采用了由电压比较器 μ PC339C 构成的闭环恒流驱动系统驱动字车电机。 μ PC339C 与反馈电阻 R1、R2 构成闭环恒流控制系统。当字车电机相绕组线圈中的驱动电流增大时,反馈电阻 R1、R2 上的电压升高,使 μ PC339C 的输出端电压降低,经 #A2 放大,使 QA2 中的复合管导通程度有所降低,字车电机的公共供电端 CRMCOMA、CRMCOMB 的输出电压降低,从而减少了字车电机相绕组线圈中的驱动电流,达到恒流控制的目的。

当 CRCOM 信号由低电平变为高电平时,字车电机供电被截止。由于驱动电流的消失,在字车电机的线圈中会形成超过电源电压 3—5 倍的冲击电压,为保护字车电机线圈不被此冲击电压损坏,引入了由 Q2、ZD1、R83 及 D45 构成的保护电路。将冲击电压迅速释放到 +35V 电源电路中。

在电路中,承受高电压、大电流、频繁开关的元件是最容易损坏的。由对字车驱动电路分析可看出,在电路中造成故障的易损元件包括:字车电机、大功率驱动元件、保护电路元件、TTL 数字逻辑集成电路。因此,在查找故障时,首先对上述易损元件逐个检测:

(1)用万用表 $\times 1$ 档测字车电机各相线圈电阻值,可判断其好坏。

(2)测 QA1、QA2 好坏,其测量方法与一般复合三极管相同。

(3)测保护电路中各元件。其中 Q2、ZD1 最易损坏。

(4)测 #C1、#D1 两片六反相器 74LS05。

实际测得是 QA2 损坏。更换 QA2,故障排除。

故障现象 4:

M1724 打印机缺针。

故障的分析与排除:

打印机使用频繁,当长时间的使用后,容易出现缺针的现象。其原因多数是由于打印头信号电缆磨断而造成的。因为,在打印过程中,打印头的运动造成打印头电缆与色带盒摩擦,信号线又特别细,容易引起断裂,影响使用。

将断裂处擦净,用直径 0.1mm 左右的漆包线,用调温电烙铁将断裂的芯线焊好,注意不要影响其它芯线。

预防的办法是:用长条胶带贴在信号电缆上,使胶带保护信号线,并经常检查,胶带磨损后及时更换胶带,这样处理后,延长了使用寿命,效果较好。

故障现象 5:

开机后,不能手动进纸,其余全部正常。

故障的分析与排除:

切断打印机电源后手动进纸正常,且工作正常时,换行、换页均正常,这就排除了进纸电机机械传动部位出的故障,故障肯定出在进纸电机的控制电路部位。M1724 打印机与其它型号的打印机一样,对步进电机采取“高低压驱动”,即步进电机转动时由 35V 电压供电。步进电机不转时也要通电将其锁住,锁住时电压为 2V。按步进电机控制电路图,开机后测 QA27 脚电压,步进电机转动时电压为 35V,正常。锁住时为 24V,异常。由于锁住时电压升高,电流增大,则产生的磁力距也大,造成不能手动进纸。QA2 型号为 TH3J10,内有三个大功率驱动管,其 3、5 脚分别输出 CRMCOMA、CRMCOMB 控制字车步进电机,7 脚输出 LFMCOMA LFMCOMB 控制进纸步进电机。查 #B3(74LS05)9、8 脚输入、输出电平正确, #A2(TD62504P)输入、输出也电平正确。对于 TD62504P,即使不清楚其内部的电路,也可简单看作三个完全一样的门电路,6、7、5 脚为三个输入端,11、10、12 脚为三个输出端。6、11、和 7、10 脚构成的门电路肯定是好的,因为字车控制电路正常,加之外围电路完全一致,所以可以对照逻辑关系判断 5、12 脚组成的门电路性能的好坏。再查 QA2 输入端(相当于三极管 b 极)2、4、6 脚电压正常且相等。而 7 脚输出电压异常,判定 QA2 坏,更换后故障排除。

故障现象 6:

打开打印机电源开关,发现指示灯不亮,检查连线一切正常。过了一会儿,发现开关处机壳温度很高、烫手。立即关闭电源开关。

故障的分析与排除:

检查本机的前级——稳压电源,发现电压表上表头指针已打到最大 250V(实际可能还不止 250V)。经检查发现稳压电源坏,输出电压大大超过了 220V。打开 M1724 打印机机壳,发现高温部件为变压器。拆下变压器,用欧姆表检查电源变压器初级,发现开路,但次级完好。未发现变压器有烧糊烧焦的现象。卸下变压器的紧固安装金属框架。一层层剥去变压器线包外的青壳纸及聚脂薄层,直到两个线包全部露出。可看出线包间夹有绝缘材料。变压器中所有抽头均从这些绝缘层中引出。剥开靠近金属底板(线包下方)的两线包间的绝缘管,该管内包着的是热敏保险丝。上面标有 220V、130℃。用万用表检查发现热敏保险丝断开了,它正好位于变压器的初级。更换热敏保险丝,M1724 打印机变压器的故障排除。

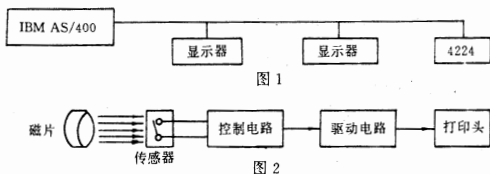
IBM4224 打印机故障排除三例

紫京计算机技术开发公司 董正平 (北京 100081)

IBM4224 打印机是美国 IBM 公司为其 S/36、S/38、AS/400 等中小型计算机系统配置而设计生产的一款打印机,在我国拥有一定数量的用户。

例 1 故障现象

如图 1 所示,一台 AS/400 小型机,连接两台显示终端,一台 4224 打印机,构成一小型机系统,当一打开打印机电源开关时,其中一台终端便立即失去显示信息。



首先将 4224 打印机与主机连接的双芯电缆连接器断开。这时终端立即恢复了显示,说明故障是由打印机引起,然后对 4224 打印机进行 301 自检,方法是:①按 ALternate 键,同时按 Test 键,使面板显示“301”数字为止。②按 Start 键。打印机便开始自检打印。自检打印正常,但打印出的数据表明:这台打印机的地址设置为 0,而通常在小型机系统中,一般是将主控终端的地址设置为 0,其它外设的地址可任意设置,所以怀疑是地址设置有问题。再来检查两台终端的地址设置,发现一台为 0,是主控终端,另一台为 1,作为显示终端,很显然,此故障并非打印机硬件故障,而是由于打印机的地址设置与主控终端的地址重叠造成。由于 4224 打印机的地址是通过面板上的按键来设置的,用户如操作不当,常会引起地址变化,造成此类故障,所以只要重新设置一下打印机地址,将打印机地址设置为 2,故障即可排除。

例 2 故障现象

开机后,Power-on 测试不能完成,面板显示故障码“66”。

4224 打印机正常情况下,打开电源开关后,机器自动完成 Power-on 测试,约需 30 秒左右,在这个过程中,面板显示变化如下:FFF→200→100→AX→00,最后的 00 表示测试通过,基本硬件正常。如果测试没有通过,则显示故障码,本例中的故障码为“66”,查随机用户手册,可知“66”表示双芯电缆适配器卡坏,通常 4224 打印机是通过双芯电缆适配器卡用双芯电缆与主机连接。打开机壳,将位于机器后部的双芯电缆适配器卡拔下,仔细检查发现该卡出厂时共有三处跳线,其中有一条在焊点处断开,用电烙铁将断点焊上,插入主机,开机故障排除。

例 3 故障现象

打印速度变慢。

据用户反映,有时在更换色带或上纸后,打印速度变慢,但用手拍打几下机身,打印速度又恢复正常。从现象上看,象是打印机电路接触不良引起,但对线路板及连接插座等检查和清洁后,故障未能排除,进一步仔细观察后发现,在打印机上盖边缘一侧隐蔽处嵌入了一块永磁式圆形小磁片,而在打印机基座的相应位置上装有一个打印头速度抑制传感器,此传感器实际上是一磁性开关,如图 2 所示,当盖上盖时,小磁片与磁性开关靠得很近,磁力使磁性开关吸合,向控制电路发出信号,使打印机按正常(高速)打印,当打开盖时,由于失去磁力,磁性开关断开,通过控制电路使打印头的速度受到抑制,即打印机进入低速打印状态。打印机在使用久后,小磁片的磁力减弱,往往出现更换色带或上纸后,盖上盖而磁性开关不能可靠地吸合。造成打印速度变慢,通过拍打,借助震动的外力,才能促使磁性开关吸合。这种故障的排除方法很简单,只要找一块永久磁铁,吸附在小圆磁片上进行“充磁”,一般只需几小时,故障就可排除。

收稿日期:1993 年 3 月 27 日

本刊启事

为了更快地推动、促进我国计算机产业的迅猛发展,及时地为计算机科研、开发、生产等部门提供计算机行业的最新发展动态,加速高新尖端技术、最新软硬件的开发、移植、引进,及时地为广大读者及计算机爱好者奉献更多更新的计算机专业技术资料,并使众多作者、译者脱颖而出,《计算机应用研究》杂志社现正在开展优惠出版各类计算机专业技术资料业务,欢迎广大作、译者踊跃投稿。具体出版业务欢迎来函来电来人商议。

联系人:张钢(邮编:成都 610041)

《计算机应用研究》杂志社启

SBW、DBW 系列全自动大功率补偿式稳压器

概 述

SBW、DBW 系列全自动大功率补偿式稳压器是我厂引进西欧最新技术,结合我国用电国情设计的稳压器。本产品在配电电压波动或负载发生变化时,能自动保护输出的供电电压的稳压,确保精密仪器设备的正常运行。

工作原理

SBW 系列三相补偿式电力稳压器的原理框图见图 1。它由调压器 T1、补偿变压器 T2,伺服电动机 MS 与电压采样比较控制装置等组成。现以 A 相为例,说明其稳压原理。从附图中可知: $U_{A0} = U_{A1} - \Delta U_A$ 。

式中, U_{A0} —A 相的输出电压; U_{A1} —A 相的输入电压; ΔU_A —A 相的补偿电压。

当输入电压 U_{A1} 或负载变化引起输出电压 U_{A0} 变化时,电压采样比较控制装置从稳压器输出端采样,采样电压经整流滤波后与基准电压上限值和下限值比较。当采样电压大于基准电压上限值时,电压采样比较控制装置控制继电器动作,使伺服电动机 MS 作一定方向旋转,带动调压器 T1 上的电刷作相对滑动,这时在 A 相调压器上出现一个 U_{A2} ,在补偿变压器 T2 的 A 相一次测相应产生 U_{A3} ,则在二次产生补偿电压 ΔU_A ,它使输出电压 U_{A0} 下降,直至回复到稳压精度的允许范围内。此时,继电器释放,伺服电动机 MS 停止转动。反之,伺服电动机 MS 向反方向旋转,补偿电压 ΔU_A 改变极性,使输出电压 U_{A0} 上升,直至回复到稳压精度的允许范围内,以达到输出电压稳定的目的。

DBW 系列单相补偿式稳压器的原理框图见图 2。它由补偿电路、电压检测电路,伺服电机控制电路及传动机构等组成。

电压检测电路用来检测输出电压的变化,产生随输出电压变化而变化的信号电压。通过与预先设定的上、下限基准电压进行比较、判断未得到的电压是否超出稳压精度所允许的范围。若超出,则控制伺服电机正转或反转,经旋转装置带动接触调压器上的电刷移动,改变调压器的二次电压,从而使补偿电压改变自动保持输出电压的稳定。 $|U_2| = |U_1| \pm |U_c|$

为了使稳压器能正常操作,保护负载设备,SBW、DBW 系列产品装有过压和故障保护电路。

三相规格(KVA):10,20,30,50,100,180,225,300,400,500,600,800,1000,

单相规格(KVA):3,5,10,20,30,50,100,180,225,300

应 用

SBW、DBW 系列交流稳压器是通用型补偿式单、三相稳压电源,用户根据自己的需要选择合适的规格。它的输入电压范围相当宽,弥补了目前交流电网电压波动较大的不足之处。另外,该系列产品可工作于任意负载。因此,深受广大用户欢迎。它广泛应用于工农业、交通、邮电、军事、铁路、科研文化等领域。例如:大型机电设备、大型仪器设备、试验设备、金属加工设备、生产流水线、建筑工程设备、大型医疗设备、宾馆、电梯、照明、广播和电视台的供电以及其它需要电力稳压的场所。

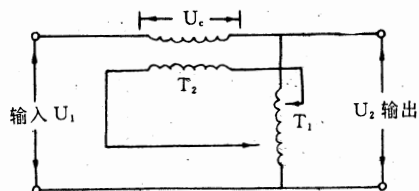


图 1

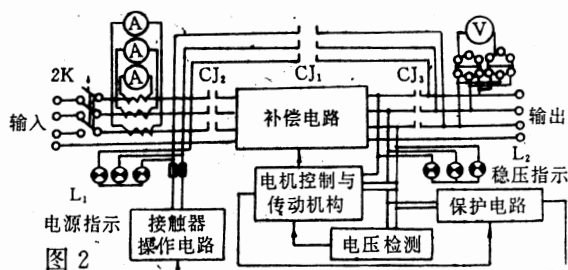


图 2

上海凯联稳压设备厂

地址:上海市常德路 379 号(静安区政府对面)

电话:2553368 电挂:1454 传真:021-2478167 邮编:200040

树立企业形象之最佳窗口 传播技术产品之最佳媒体

欢迎加入《计算机应用研究》杂志协办单位行列

现代社会已进入信息世界。信息是发展经济之重要战略资源,及时沟通信息并利用信息资源已成为当今社会之潮流,也是各厂矿企事业单位,各科研院所,各大专院校通往社会、了解市场、树立自身形象,传播高新技术、高科技产品、跻身于科技之林、占领市场之重要途径。为了广辟信息之源,加强信息意识,开发信息资源,实现信息价值,增强竞争能力,由国家各部委所属四川省电子计算机应用研究中心、云南省电子计算中心、贵州省计算中心、新疆电子计算中心、甘肃省计算中心、广西计算中心、山东省计算中心、河南省计算中心、安徽省计算中心、吉林省计算中心、内蒙古计算中心、青海省计算中心等十二省区计算中心共同联办的《计算机应用研究》杂志为真正发挥横向联合,专业互补,信息共享之优势,热忱欢迎各大中小型企业单位、各科研院所、各大专院校积极加入《计算机应用研究》杂志协办单位行列。

●《计算机应用研究》杂志简介●

《计算机应用研究》杂志创刊于一九八一年元月,系由国家科委所属四川、云南、贵州、新疆、甘肃、广西、山东、安徽、河南、吉林、内蒙古、青海 12 省、区计算中心等单位共同联办的、国内外公开发行的计算机技术双月刊,创刊 12 年以来,曾多次荣获科技期刊优秀奖,并已被国家科委判定为中国科技论文统计源期刊。由于《计算机应用研究》杂志自始至终坚持“以应用促进研究,以研究指导应用”,“传播计算机应用知识、沟通计算机应用信息、推广计算机应用成果、促进计算机学术交流、提高计算机应用水平、拓展计算机应用领域、增加计算机应用效益、加快社会信息化进程”的办刊宗旨和新颖、实用、快速、优质的办刊方向,老订户对她“爱不释手、念念不忘”,新读者“一见钟情,相见恨晚”,共同的感受是“一册在手,受益匪浅”,深受广大读者、作者所喜爱,其发行量更是年年递增,发行范围不仅覆盖各省、自治区、直辖市和各地、市、州、县,还遍及海内外,连赫赫有名的英国大不列颠国家图书馆也订阅并收藏本刊,同时世界各国国家图书馆也均可直接查阅本刊,国外一些著名大公司也纷纷致函本刊编辑部索取《计算机应用研究》杂志。在国内外广大读者关心支持下,《计算机应用研究》杂志现已成为国内计算机领域知名度较高的专业期刊之一。

《计算机应用研究》杂志设置栏目有:“新产品、新技术窗口”、“综述评论”、“应用实践”、“研究探讨”、“维护维修”、“动态简讯”等等,栏目众多,内容新颖,丰富翔实,兼顾高、中、低不同层次。同时本刊为国家工商局登记注册的广告经营单位,承接国内外计算机行业及电子行业的广告设计、发布、宣传,并承接计算机领域最新技术的开发、技术培训、展览、咨询、资料翻译、印刷出版等多项业务。《计算机应用研究》杂志社热忱欢迎各厂矿企事业单位、各科研院所、各大专院校踊跃加入《计算机应用研究》杂志协办单位的行列,以达到信息共享、彼此受益之目的。

●统一刊号: CN51-1196/TP(国内)
ISSN1001-3695(国际)

●邮发代号: 62-68(国内)
BM4408(国外)

●广告经营许可证:工商广字 050 号 ●全国各地邮局均可订阅 ●国外总发行:中国国际图书贸易总公司

●《计算机应用研究》杂志社通讯地址:成都市人民南路四段 11 号附 1 号(邮编 610041)

●联系电话:(028)-5582666 转 2055 ●传真:(028)-5581946

●《计算机应用研究》杂志社为协办单位提供下列服务●

1. 每一年度在《计算机应用研究》杂志上免费为各协办单位提供 6 次内文黑白宣传版面,其每次提供的广告宣传版面的规格尺寸为 16 开整版(即高 23 厘米×宽 15 厘米),以刊载宣传各协办单位所提供的:“高新科技产品、高新技术介绍宣传”“协办单位简介”“专利发明”“技术转让”“实用技术”“科技园有赏信息”“科技推广”“项目合作”“项目包销”“价格动态”“市场预测”“国际合作”“贸易机会”“招标投标”“国内外科技动态”“物资交流”“展销活动”“培训启事”等各类信息。
2. 在《计算机应用研究》杂志及所有出版物的版权栏目中刊登出各协办单位名称,以提高各协办单位在社会上的知名度。
3. 在协办期间,协办单位可以推荐一名计算机专业或电子专业的技术人员担任《计算机应用研究》杂志的编委,并在杂志版权栏目中刊出其编委姓名,同时,协办单位还可推荐 1—2 人担任《计算机应用研究》杂志社的特约记者,负责采访本单位、本地区有关计算机推广应用的新闻稿件及宣传报道。
4. 在协办期间,《计算机应用研究》杂志社向协办单位所推荐的编委,特约记者颁发记者证,以资证明。

- 5. 在稿件质量同等情况下,《计算机应用研究》杂志社优先录用协办单位推荐的技术稿件。
- 6. 免费为各协办单位提供有关技术项目的咨询服务。
- 7. 若协办单位欲在《计算机应用研究》杂志封面或插页上作彩色广告宣传,杂志社仅收取印刷工本费,不再另收广告费。
- 8. 按时赠阅各期《计算机应用研究》杂志。
- 9. 《计算机应用研究》杂志社不定期召开协办单位办刊工作会议,听取各协办单位对办好刊物,提高经济效益之设想、意见、建议。

●参加《计算机应用研究》杂志协办单位手续●

- 1. 各大、中、小型厂矿企事业单位、科研院所、大专院校可随时加入《计算机应用研究》杂志协办单位。欲加入协办的单位每年度需缴纳协办费 5 千元整。
- 2. 请详细填写《计算机应用研究》杂志协办单位成员回执卡,经《计算机应用研究》杂志社董事会审核备案后,确认为协办单位成员(另:回执卡各栏目务必工整填写,切勿潦草)。
- 3. 请将填好后的《计算机应用研究》杂志协办单位成员回执卡寄回《计算机应用研究》杂志社张钢、李泽民同志处存档。
- 4. 各协办单位的协办费请汇至:
 ※户名:四川省电子计算机应用研究中心 开户行:工商行成都跳伞塔分理处 帐号:24808921280
 ※若邮局汇款请汇至:成都市人民南路四段 11 号附 1 号 (邮编 610041) 《计算机应用研究》杂志社
 在收到各协办单位的协办费后,杂志社立即将报销发票寄至各协办单位。
- 5. 为确保双方责、权、利,协办单位可与《计算机应用研究》杂志社签订具有法律效力的协办协议,以期双方共同遵守执行。

热忱欢迎各单位来电、来函、来人咨询参加《计算机应用研究》杂志协办单位事宜。我们期待与您合作,共展宏图!

联系地址:成都市人民南路四段 11 号附 1 号 邮编:610041《计算机应用研究》杂志社

联系人:张钢 李泽民 联系电话:(028)—5582666 转 2055 传真:(028)—5581946

《计算机应用研究》杂志协办单位成员回执卡 编号:

协办单位名称:		联系人:
协办单位通讯地址:		邮政编码:
电话:	电挂:	传真:
首次加入《计算机应用研究》杂志协办单位时间:		
缴纳协办费金额及寄款日期:		
协办单位名称及公章:		

注:回执卡填好后请寄回《计算机应用研究》杂志社张钢、李泽民同志处存档

