

微小型计算机



开发与应用

MICRO-MINICOMPUTER
DEVELOPMENT & APPLICATION



1992

11

微小型计算机开发与应用编辑部

微小型计算机开发与应用 1992年 第一期（总第57期）

微小型计算机开发与应用（公开发行）

1981年创刊

编辑：《微小型计算机开发与应用》编辑部

发行：天津市邮局

出版：天津市电子计算机研究所

印刷：天津武清县长宏印刷厂

天津市计算机学会

地址：天津市河西区友谊路宾馆南道5号

订购处：全国各地邮局

邮政编码：300061

定价：0.95

邮局代号6—87

津工商广字0146号

国内统一刊号CN12—1122

ISSN 1001-8786

《微小型计算机开发与 应用》编辑委员会

顾问 郭平欣
主编 黄侃
副主编 王治宝 张凤枝
委员 (以下按姓氏笔划排列)

于万源	于清汶
王治宝	王 镭
王士禧	王寿松
付园明	许镇宇
朱植松	曲庭维
李凤祥	刘连棣
陈力为	吴锦声
房家国	张凤枝
夏纪寅	夏业勋
袁维本	曹东启
黄侃	黄宝良
章渭臣	梅克定
童宣明	裴少峰
薛大中	

1992年第1期目次 (总第57期)

应用实例

病房信息系统的实现与应用

.....邱明辉 刘志敏 赵京莉 付长青(1)

压力容器辅助设计及报价系统在产品开发中的

应用.....林 海(4)

单微机控制的多车安全检测系统.....黄钦胜(7)

VAX电子汽车衡联机处理系统设计

.....刘冠福(10)

单片机实现电气传动系统PI调节器

.....王恒震(16)

计算机软件

仪表的微机校验及综合管理

.....张汉涛 马俊杰(20)

C语言直接操作DBF数据库的方法与实现

.....谷德桥 马英忱 冯勋欣(23)

一种以IBM PC/AT来控制多总线图象分析

系统.....叶红 霍强 唐慧明 顾伟康(26)

数组和文件中直接生成图形数据的绘图方法

.....邓长根(29)

一种图象数据压缩存贮技术及其实现

.....孟宪福(35)

提高IBM PC微机输出效率的方法

.....支超有(39)

DDJ动态、多维数据检索算法设计

.....马长生(42)

接口技术

MCS-51单片机配接CE-515P彩色绘图打印机和一

种汉字打印的实现.....戴强(44)

经验点滴

用MIRROR程序巧取硬盘分区表

.....戈 海(47)

长城0520CH多功能板打印机适配器维修一例

.....王映辉(48)

CONTENTS

APPLICATION EXAMPLE

Implementation and Application of Ward Information Management System.....	
.....Qiu Minghui Liu Zhimin Zhao Jingli Fu Changqing (1)	
Application of CAD for Prossure Container & Quotation System in Product Development.....	
.....Lin Hai (4)	
Single Microcomputer Controlled System to Detect Automobiles for Safety.....	
.....Huang Qinsheng (7)	
The Design for VAX Electronic Car scale On-Line Processing System.....	
..... Liu Guanfu (10)	
PI Regulator in Electric Transmission System Implemented by Using Single Chip Microcomputer.....	
.....Wang HengZhen (16)	

SOFTWARE

Calibrating & Synthesising the Meters by Using Microcomputer.....	
.....Zhang Hantao, Ma Junjie (20)	
Approach & Implementation for Directly Operating the DBF Database by C.....	
.....Gu Deqiao Ma Yingchen Feng Xunxin (23)	
A Multi-Bus Image Analysis System Controlled by IBM PC/AT.....	
.....Ye Hong Huo Qiang Tang Huiming Gu Weikang (26)	
Plotting Method of Forming Plot Data into Array and File.....	
..... Deng Changgen (29)	
The Technique for Compressing the Image's Data & Its Implementation.....	
.....Meng Xianfu (35)	
Method for Improving the Output Efficiency of IBM PC.....	
.....Zhi Chaoyou (39)	
Algorism Design for DDJ Dynamic and Multidimensional Data Retrieving.....	
.....Ma Changsheng (42)	

INTERFACE

Connecting between MCS-51 Single Chip Microcomputer and CE-515P Colour Plotter-Printer and Implementation of Chinese Character Printing.....	
.....Dai Qiang (44)	

EXPERIENCE

Ingenuisly Fetching the Hard Disc Partition List by Using MIRROR Program.....	
..... Ge Hai (47)	
An Example for Maintaining the Multi-Function Board Printer Adapter of the Great-Wall 0520CH.....	
..... Wang Yinghui (48)	

病房信息系统的实现与应用

解放军总医院计算机室 邱明辉 刘志敏 赵京莉 付长青

(北京五棵松解放军总医院计算机室 邮编 100853)

摘要 本文介绍我院在小型机上用集中处理方式开发的病房信息系统,详细描述了系统的数据库设计、I/O设计和数据的安全保密等技术问题,并对其推广应用作了简要介绍。

1 前 言

计算机真正用于病房业务管理，目前在我国还处于初期阶段。病房日常的一些工作，如象抄写药疗单、治疗单、膳食通知单、送取检验单、主管护士把医生所开医嘱抄写到医嘱记录单上等一系列繁杂的工作，如果能用计算机代替，那将会把医护人员从繁忙中解脱出来，从而提高工作效率，减少手工引起的一些差错事故。由于录进了病人基本信息和医嘱，在医院管理上还将引出很多其它方面的重要应用，诸如划价、病人流动情况日报的产生以及领导机关对病人情况的查阅等等。

系统的硬件包括: HP3000小型计算机, 中英文终端和中英文打印机。

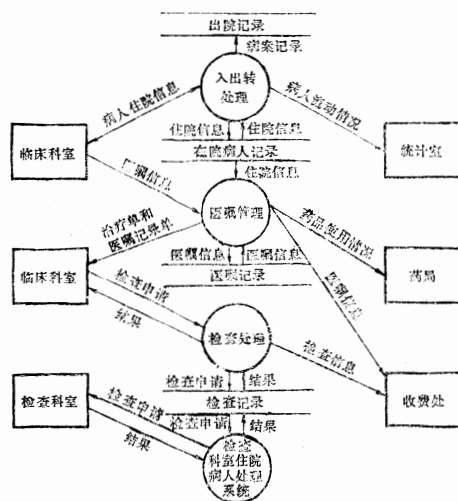


图1 病房信息流程

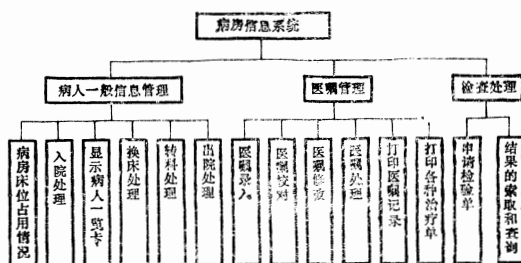


图2 系统功能结构

2 系统信息流程和功能

首先，我们来看一看病房信息的流程，如图1所示。

由信息流程，我们可以看出：病房信息管理主要是对病房床位、病人一般信息、医嘱和检验单的管理。病房床位和病人一般信息管理包括入院、出院、转科和换床；医嘱管理是整个病房信息的核心，它包括医嘱的录入、查询和根据医嘱产生的各种治疗单和医嘱记录单；检验单的管理包括检验单的申请、检验结果的索取和查询。系统的功能结构如图2所示。

3 实现时应该重点解决的几个技术问题

我们医院的病房信息系统在小型机上采用集中式处理方式开发而成,在实现时,我们着重考虑了以下几个问题:

3.1 数据库设计

在我们这个系统中把数据库分为两类：公共数据库和私用数据库。公共数据库存放代码名称字典，例如药品代码字典库，途径代码字典库和用法代码字典库等，供所有病房公用；私用数据库（以下称为“病人库”）存放病人的一般信息、医嘱和检验结果，每个病房一个，供单个病房专用，这样设计有以下几个优点：

a. 检索时速度快；

b. 安全保密，因为专用数据库数据量不会很大，而且每个病房只能访问它自己的数据库。在实现时，我们还设计了几个临时公共库，供所有病房做转科、换床和出院处理时，暂时保存病人信息使用。为了减少磁盘空间的占有量，病人库的存放形式以代码为主。

由于系统使用的是HP3000小型机上的IMAGE网络数据库，所以，在设计病人库时，我们在数据库记录中设置了一些标志数据项，将来对系统进行维护和功能扩充时，可以达到不改变数据库结构就能满足要求的目的，为系统的维护和功能扩充提供了便利条件。

3.2 I/O设计

在事务处理中，大量的操作是输入、输出操作，它直接影响着应用系统的使用。如果I/O问题得不到很好的解决，系统应用起来就存在着很大困难，更难以推广。

3.2.1 数据的录入

在这个系统中，我们采用两种方式进行数据录入，即：间接方式和直接方式。

3.2.1.1 间接方式

这种方式主要用于提取病人的一般信息。我们这里，间接方式是指从主机中提取已有信息供显示观察。

凡是住入我们医院的病人，不管是急诊，还是门诊都得经过住院处办住院手续，这样病人的一般信息（诸如：姓名、性别、年龄、入院日期、住院号、门诊诊断和工作单

位等）就直接在住院处进行录入，等病人住入病房后，再由病房进行入院处理。病房只需输入一个入院日期和住院号即可将病人一般信息从住院处数据库中间接提取显示出来，然后给病人安排一个床号和主管医生，就将住院病人接收到病房。

3.2.1.2 直接方式

我们这里的直接输入是指通过终端屏幕直接输入信息到主机中。我们这个系统主要采用此方式进行输入。

前面提到医嘱管理是整个病房信息管理的核心，而医嘱录入又是医嘱管理的基础。由于应用计算机管理病房的日常工作，所有由手工抄写的各种治疗单和医嘱记录单都将全部由计算机产生，因此医嘱录入的难易、快慢程度直接影响着病房正常工作的进行。为此，我们对医嘱进行了分类，大类分为：长期医嘱和临时医嘱；小类共分六类，即：护理常规、分级护理、膳食、病危/重、药物和其他医嘱。其他医嘱是指前面五类不能包括的所有医嘱，诸如：记入出量、输血、中药等。这六类医嘱，我们分别进行了编码，采用编码的形式进行医嘱录入。

病人在住院期间，药物医嘱相对于其他几类医嘱要多些，因此，为了方便用户录入，我们在药品字典库中，使用了两种代码，即按新编药物学手册分类的编码和汉语拼音缩写编码。对于用户，大部分都采用汉语拼音缩写代码进行录入，因为它简单，几乎不需要任何培训，只要粗会汉语拼音即可，且录入速度快。汉语拼音代码的编码原则是：取药品名中每个汉字的汉语拼音字头，然后将它们合在一起。这样，即使汉语拼音不太好，也不妨碍我们录入医嘱。例如，要输入“六神丸”这种药，只需要输入“LSW”即可把它送入病房病人专用库。当然，有时会出现不同药，却有相同的汉语拼音缩写代码，这时，用户应该象在微机上使用汉语拼音输入汉字一样，从中挑选出所需药品。由于其它几类医嘱相对要少得多，因此，我们

采用普通编码的形式进行录入。在输入过程中,对于字典库中没有的医嘱(当然这种情况很少),我们让用户采用正文的方式输入。这样,用户在录入医嘱时,采用编码和正文并用的方式,以确保病人完整医嘱的录入。

3.2.2 结果的输出

病人的医嘱通过终端进入主机后,相应地应该产生一系列的工作单据,例如各种治疗单和医嘱记录单等等,所有这些单据,都在各病房本地打印机上输出产生。

3.2.2.1 生成各种治疗单

治疗单是根据医嘱途径进行分类产生,我们把它分为以下六类:药疗单、注射单、治疗单、小治疗单、点眼小治疗单和中药单。

3.2.2.2 产生医嘱记录单

病人的医嘱,不管是临时的还是长期的医嘱都在终端联机打印机上打印相应的医嘱记录单。我们采用续打的方式打印医嘱记录单,即每次打印医嘱时都由计算机控制自动接着上一次打印的位置继续打印。其实现算法描述如下:

假设医生今天为某床病人开了 m 条医嘱,并且已经输入到计算机中,在我们打印该病人的医嘱记录时,首先查找以前已经打印了多少条医嘱记录(已打印过的医嘱记录,我们在数据库中有标记标识),不妨设为 n 条,并假设一页纸可以打印 k 条医嘱,一条医嘱占一行纸,在打印今天的 m 条医嘱之前,打印机应该空走 $\text{mod}(n/k)$ 行纸(mod 是求余数之意),然后再接着打印今天的 m 条医嘱。

当然,要实现计算机自动寻找打印位置这一功能,还必须配置较好的打印机才行。具体讲,打印机至少应该具备两个功能:a.单页纸打印功能;b.自动送纸功能,比如,LQ—1600K, OKI—5320SC等等。我们医院病房配置的是LQ—1600K打印机。

3.3 数据的安全保密

3.3.1 责任问题

病房数据,特别是医嘱的录入,不是固定专门操作人员,而是由病房当班护士录入,这样就有必要由计算机自动记录数据录入者姓名。当医护人员发现数据有错时,则要进行修改。修改人的姓名、修改时间等信息也自动记录到了主机中。采用此措施既可敦促操作人员认真负责地录入数据,又可防止恶意地破坏数据,使病人数据安全可靠。

3.3.2 数据的安全保密

前面我们提到把数据库分为公用和私用两类,对前者,只允许使用我们这个系统的用户对它进行读操作,数据库的维护由专人负责;对于后者,每个病房只能访问属于它自己的数据库。用户访问某个病房的私有数据库,必须具备两个条件:a.在系统保密字库中有他的保密字记录;b.用户属于该病房的人员。这样,各病房的数据只限于本病房工作人员操作,从而达到安全保密的目的。

4 系统的推广与应用

4.1 病房数据的其它应用

住院病人一般信息和医嘱以及检查项目的录入,还将为医院其他方面的计算机管理提供原始数据。例如,病人的医嘱和检查项目为收费部门划价提供准确数据,由于人为因素少收或不收费的现象可以得到有效控制。又如,病房数据为病人流动情况(诸如入院、转科、出院和死亡)日报的产生,以及领导机关对病房床位使用情况和病重/危病人情况的查阅提供及时而准确的数据等等。

4.2 系统的推广与应用情况

病房信息系统在我们医院部分高干病房推广应用有半年多,与其他别的应用系统相比,它的推广与使用要冒一定的风险,以前手工抄写的各种治疗单和医嘱记录单都被计算机替代管理,万一出什么差错,其后果不堪设想。因此,刚开始试用时,我们采用人

压力容器辅助设计及报价系统

在产品开发中的应用

兰州石油化工机器厂计算办 林海

(邮编 730050)

摘要 本文详细介绍了兰州石油化工机器厂压力容器辅助设计及报价系统的设计结构, 系统流程的设计思想和在产品开发中的应用, 它对缩短产品开发周期是非常有益的。

1 概述

为了使我厂炼化产品尽快适应国内外工程投标的速度和质量, 也为了适应当前国内外产品的更新速度, 缩短产品的设计周期和提高产品的设计质量便显得非常重要, 为此我们开发成功了压力容器辅助设计及报价系统。该系统是在IBM PC/XT机器上开发成功的, 使用的语言是C86、FORTRAN77和dBASE III 数据库, 该系统可以对三种压力容器(球罐、卧罐、换热器)进行强度计算, 并通过强度计算的结果进行产品报价

(目前只对国内价格进行报价), 也可通过用户提供的图纸直接进行报价, 该系统不仅可以提高产品的开发速度和质量, 而且可以为设计人员提供产品的最终销售价格, 使设计人员对自己设计出的产品可行性有个大致的了解。

2 系统功能

该系统主要包括三大部分, 即产品强度计算模块, 报价计算模块和各类型各材质的材料单价库的维护模块, 详细系统结构见图1。

工和计算机并行管理的方式进行, 等使用一段时间后, 医护人员发现计算机管理的那套信息与人工管理完全相同, 而且计算机管理规范、整洁、清楚、效率又高, 这样, 病房就逐步开始使用计算机管理了。

病房使用本系统以来, 医护人员普遍认为, 只要一次正确录入医嘱后, 随时都可以产生病人的治疗单和医嘱记录单, 不但避免了重复性的劳动, 而且减小了差错事故的概率, 从而提高了工作效率。

5 结束语

我们医院的病房信息系统, 是在HP3000/48小型机上开发, 采用集中式处理方式实

现的。目前, 受经费限制, 每个病房只安装了一台终端, 主要供护士使用, 如果经费和机器允许, 医生办公室也安放一台终端, 这样医生开医嘱可以不用医嘱本, 直接在终端上开医嘱, 从而可以避免重复性的劳动。如果更多的检查科室和临床辅助科室能够联机, 临床医护人员可以从计算机终端传递膳食申请、检查申请及结果, 不但可为诊断治疗提供及时方便的服务, 还可减少手工传递抄写的麻烦和不便, 提高工作效率。

本文撰写完后, 任连仲高级工程师对此稿进行了审阅, 并提出很多宝贵意见, 在此深表谢意。

2.1 产品强度计算模块 该模块主要是对换热器, 球罐, 卧罐产品的各零件各部位进行强度和应力计算, 为设计人员提供可靠的参考, 最终产生一强度计算书, 如果对产品进行报价那么同时也产生一个数据文件, 该文件中有产品各零部件的材质, 类别, 型号及重量。

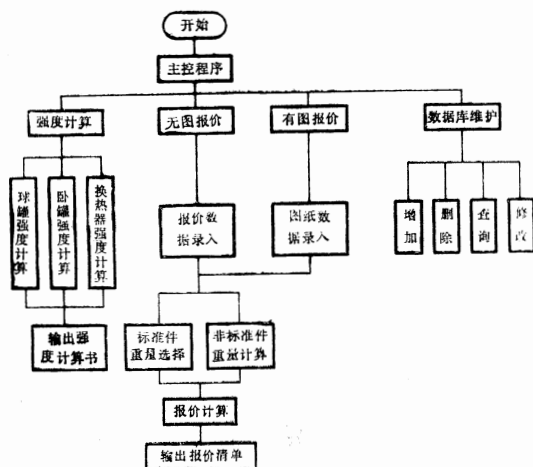


图 1

2.2 报价计算模块 该模块主要是对各材料的金额, 工费, 车间成本, 工厂成本及单台价格进行计算, 在该模块中可录入各材料的利用率及各类参数等(如企业管理费率, 工费率等), 最终产生一张报价清单。

2.3 单价库的维护模块 该模块就是对该料单价库进行维护。由于最近几年材料价格改动比较大, 并经常有新型材料替代老型号的材料, 所以材料价格及型号需要经常修改, 每次需要报价计算时, 首先要对单价库中的材料型号及价格进行核查, 使价格库中的各类数据符合最近材料价格的变动情况。该模块可以对数据进行查询, 修改, 录入和删除。

3 系统的基本原理

系统的原理图如图2。

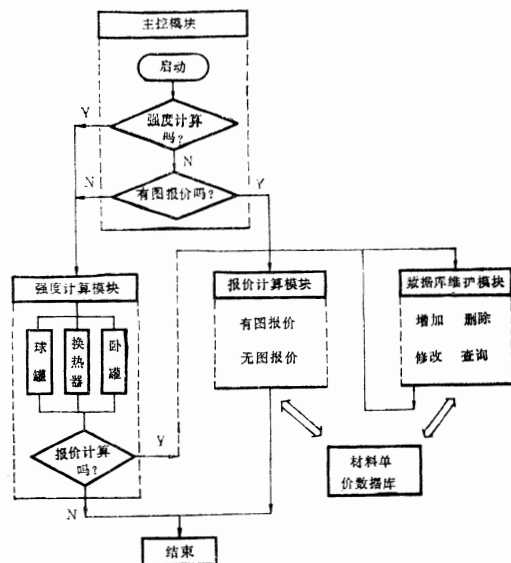


图 2

系统采用模块结构, 各部分是独立的, 又是一个整体, 如果用户选用了无图报价, 那么在强度计算完成后, 生成一个 .TXT 文件, 因为无图报价子模块是 dBASE III 数据库编写的, 它可将 .TXT 文件直接转化成 dBASE III 数据库文件, 文件中有被设计产品各主要零部件的材质, 型号, 规格及重量, 无图报价子模块将利用数据库中的数据进行价格换算, 以此来达到设计结果与无图报价之间传递数据的作用。

在有图、无图报价子模块中有对产品的标准件、非标准件及附加件进行数据录入的功能标准件、附加件和非标准件结构图如图3、图4、图5, 以此来达到对强度计算中没有涉及到的部件进行数据处理, 使报出的价格更加准确, 当数据准备完成后, 就调用了报价计算子模块, 在子模块中有对报价所需要的一些基本数据(例如: 企业管理费率、工费率、利润率、税率, 平均吨工时及

4.3 有图报价 当定货用户提供了图纸时，那么可直接将图纸上各零部件的有关数据进行录入，直到对产品重量有影响的所有数据都录入完成后，便可进入报价计算，报价计算完成后，产生一张报价清单，用户可将报价清单表打印出来做为参考。

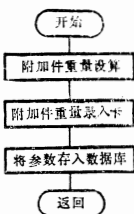


图 4

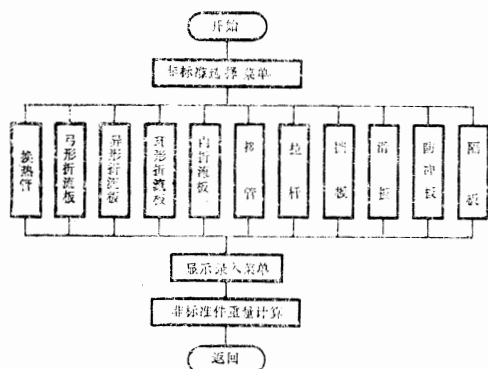


图 5

4 系统的总体流程

总体流程见图6，其工作过程如下：

4.1 强度计算 强度计算目前可对换热器、球罐和卧罐进行强度计算，其中可对产品进行单台设计、系列设计和校核计算，并根据“压力容器设计规定”GB150 进行程序设计，强度计算完成后如果要打印强度计算书，就调用打印程序进行打印，如果需要报价，就转入无图报价模块进行报价数据处理并计算，最后返回到主控程序。

4.2 无图报价 当定货用户没有提供图纸时,就要进入无图报价子模块,在无图报价

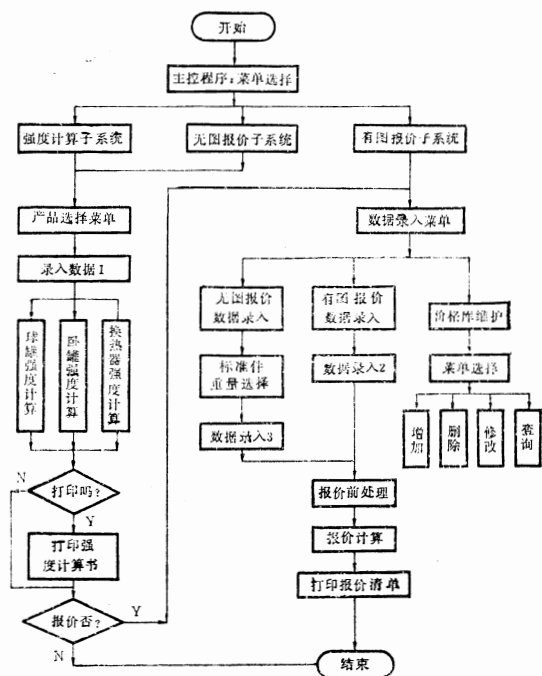


图 6

图注:

数据1：强度计算所要求的数据

数据2：原图中各部件的材质，重量、及锻件级别等。

数据3: 非强度计算的各部件材质和重量.

5 结论

单微机控制的多车安全检测系统

华南理工大学计算机系 黄钦胜

(邮编 510641)

摘要 本文介绍了单台微机并行控制安全检测线上二辆汽车的指挥、检测、打印报表和检测结果存档的检测系统。该系统主要采用了陷阱技术。通过键陷阱程序的执行,达到并行检测二辆车的目的。

近几年来,随着机动车辆的迅速增加,为了确保汽车的安全性和防止公害,全国各地先后建起了机动车辆安全检测线,这些检测线大致可分为两大类,即手动检测线和计算机控制的检测线。

手动检测线其检测结果受人为因素的影响大。

计算机控制的检测线,其实施方案大致有以下几种:第一种检测线允许在线进行一部车的检测,计算机检测系统实现对车辆各项数据的采集处理、打印报表和结果存档的操作,而车辆行驶的指挥和检验设备的启停均由检测员控制。另一种实施方案是分级型的计算机检测系统。这种系统具有并行检测性高的特点。第三种实施方案是本文要介绍的单微机控制的多车并行检测系统。

1、二车并行检测系统的

硬件组成

为便于说明,以单微机指挥在线二辆车的行驶,控制检验设备的启停,采集车辆各

项检测数据并加以处理,最后打印报表和将检测结果及车辆有关参数存档等操作为例进行讨论。检测系统由微机基本系统及外围设备等组成,参见图1。

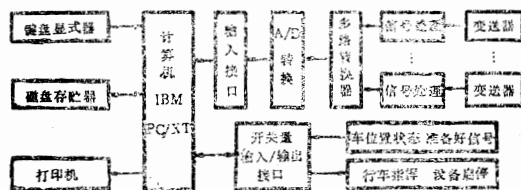


图1 检测系统的硬件组成框图

微机基本系统包括主机、磁盘存储器、键盘和打印机;外围设备主要包括A/D转换器和开关量的输入输出。由变送器将汽车的各种物理量如制动力、车速、前照灯光强度、光轴偏斜量、侧滑量和有害废气含量等变换为电压模拟量,经过放大器把模拟量放大并变换为单极性0~5V的电信号,再由多路转换器轮流切换到某一通道的模拟量进行A/D转换,由输入接口将采集到的数据送入计算机,采用数字滤波技术将所采集的数据加以处理,最后求出相应的物理量。开关量输入输出接口用于接收车辆位置状态和检测

创造出一个CAD软件是要应用,应用才能得到社会的承认,因此必须满足先进性、实用性和可推广性,本系统的强度计算和报价计算结果是先进的、准确的、新颖的一种方法,输出的强度计算书和报价清单表与本厂现用的计算方法得出的结果是完全一致的,经过一年的使用,受到设计人员和经

销人员的欢迎。本系统除了产品的强度计算是用C语言和FORTRAN77语言编写的以外,其它部分都是用dBASEⅡ数据库语言编写的,并将其编译成EXE文件,所以本系统即不影响运行速度,又易于掌握,非常适合于普及推广应用。

工位准备好信号以及以开关量表示车辆的外观检测结果；控制箱信号灯和检验设备的动作，达到指挥司机行车和配合检验设备操作的目的。

根据车辆检测的实际情况，把七个检测项目依次分为二个检测段：第一检测段包括废气、制动力和车速三个项目的检测；第二检测段包括前照灯、噪声、侧滑和外观检查四个项目的检测。在线并行检测二辆车即允许第一、二检测段各有一车的检测操作并行进行。

2 二车并行检测系统

程序模块的设计

检测系统的程序具有自动检测、输入脱机检测 results 和查询车辆检测参数等三大功能。以菜单形式提供用户选择。

为实现单台微机控制二辆车的并行检测操作，设置了键陷阱程序，把第二检测段的前照灯、噪声和侧滑等项目的检测用键自陷中断进行处理。每当这些检测项目准备就绪时，产生键中断信号，处理机便自动执行陷阱程序，采集变送器（经放大）的模拟信号进行A/D转换，并将所采集的数据加以保存，待适当时候再进行处理。为产生键信号，需设置一开关，当检测到前照灯项目和噪声喇叭时由检测员按下开关，将此开关信号与检测侧滑量由红外线产生的开关信号并联后，连接到可被陷阱键（包括功能键F₁~F₁₀和光标移动键）的任一按键的触点上（设用F₁键）即可。

键陷阱程序流程图参见图2。在陷阱程序中，根据车辆在第二检测段所要检测的项目完成数据采集和建立下一检测状态，指挥车辆行驶等操作。

单台微机控制在线二辆车并行检测的关键在于引用了键陷阱中断技术。第二检测段中的前照灯、噪声和侧滑的检测是通过键F₁的陷阱程序进行的，而外观检查项目则

是在适当的时候采用程序查询的方法进行的。当查询到外观检查完成后即读入外检参数并打印报表和数据存档；当查询的外检操作尚未完成则继续运行程序完成其它操作。仅当在第一检测段的车已完成检测第一段的项目（或无第一段检测项目）时才等待外检的完成，这就使得程序直接控制的输入输出得到灵活的应用。

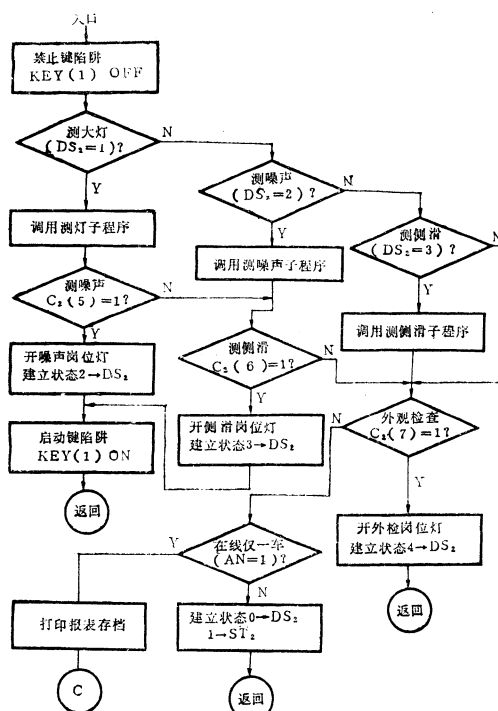


图2 键陷阱程序流程图

利用键陷阱中断技术检测前照灯、噪声和侧滑等参数的过程应不影响第一段检测项目的检测。我们采用了中速的A/D转换器，用BASIC调用汇编程序的采集速度约为0.1毫秒，则采集10个数的时间只需1毫秒时间，陷阱程序的执行不大于几十毫秒，这对于第一检测段的废气、制动力和车速的测量过程没有什么影响，因为这些量的变化过程相对于几十毫秒来说，变化是缓慢的，而且这种影响只是当正在测量第一检测段的参数时掉进陷阱程序时才发生。

为了保证第二检测段前照灯、噪声和侧滑量测量的实时响应性能，主控程序的编制

应防止一个程序行过长的执行时间。例如

```
10 FOR I=1 TO 4500:NEXT I
延迟约2秒时间, 应将其分为二行编写。
```

```
10 FOR I=1 TO 4500
20 NEXT I
```

此外, 为了保证按键 F_1 的误动作不引起陷阱程序的执行, 只在进入前照灯或噪声或侧滑项目时才用 $KEY(1)$ ON驱动 F_1 键的陷阱, 一旦进入陷阱, 立即用 $KEY(1)$ OFF撤消陷阱, 避免因 F_1 误动作再次掉入陷阱。并行检测二车程序, 其主流程在于第一检测段检测项目上。主程序流程见图3。

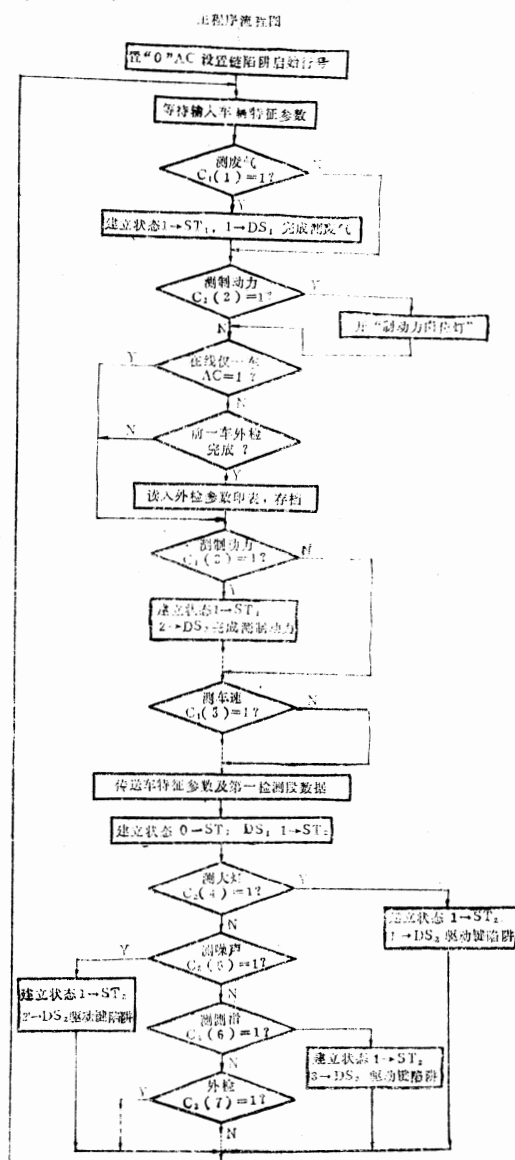


图 3

从图3可见, 当第一检测段某项目检测完后, 根据下一检测项目建立相应的状态指挥车辆行驶, 在此期间, 若有车辆完成要检的全部项目, 则由打印机将结果印出并存档, 使印表和车辆进入下一检测工位并行进行, 从而提高检测速度。

另外, 为了使操作人员输入被检车特征参数 (如车主单位, 车牌号码等) 期间不影响检测速度, 使用一台微机作为输入车特征参数, 用RS—232C异步通讯口将特征参数送给检测线上的计算机, 通讯过程只需1秒钟左右。

3 总结

采用以上方案控制在线二辆车的并行检测系统在倍宜县汽车检测线正常运行已有半年时间, 由于系统设计较充分考虑了车辆行驶、检测和打印等操作的并行性, 使检车速度得到满足。检测车辆的速度约每六、七分钟一辆, 这一速度可满足年检近二万台车的地区。值得指出的是, 若要进一步提高检车速度, 则只要使在线并行检测的车辆数增加。如把检测项目分为三个检测段, 设置多个键的陷阱, 对自动检测程序作少量的修改, 可使在线并行检测三辆车的要求得到满足。

单微机控制的多车检测系统具有性能价格比较高、便于用户使用和维修等特点, 符合国情, 有推广使用的实际意义。

参考文献

- 1 谢剑英. 微型计算机控制技术. 国防工业出版社, 1985
- 2 张如洲. 微型计算机数据采集与处理. 北京工业学院出版社, 1987
- 3 吴浩珪等. 机动车安全检测计算机控制系统. 华南理工大学, 1989

VAX电子汽车衡联机处理系统设计

拱北海关关闸口办事处技术科 刘冠福

(邮 码 519030)

摘要 为扩大计算机的应用范围,往往要把非计算机外围设备联入计算机系统,本文介绍了把电子汽车衡联入VAX计算机系统的两个方案,并详细介绍了一种最优的实现方法。

1 引言

电子汽车衡被广泛地应用在铁路、港口、工厂等部门,完成称重计量工作。随着微电子技术的发展迅速,它一般都配有标准输入、输出接口的显示仪表,可以直接与计算机连接,仪表数据能为计算机应用软件直接接收,但是目前国内电子汽车衡与计算机联机使用,一般都以IBM-PC及其兼容的单用户个人计算机为多,把电子汽车衡与多用户的计算机系统联机应用还很少看到。

最近我们把电子汽车衡直接连入VAX计算机网络系统的尝试,使它成为VAX计算机系统内的联机共享设备,并实现了远距离的终端操作,取得了很好的效果。

2 实现方案的方法

把电子汽车衡直接连入VAX多用户计算机系统,虽然电子汽车衡仪表提供了标准的RS-232接口,但其处理并没有单用户的个人计算机那么简单,其主要原因是所有连入VAX计算机系统的外围设备都是由各自的驱动程序控制的,然而系统没有提供非计算机外设通用的设备驱动程序,我们的目的是要想法利用VAX计算机系统现成的设备驱动程序。通过分析,我们发现VAX计算机的终端设备驱动程序可以为我们使用,主要是电子汽车衡仪表输出接口与终端设备的I/O接口在以下两个方面相同:一是接口都是RS-232;二是交换的信息都是标准的ASCII码。

把电子汽车衡仪表的输出接口与计算机终端数据线连接,使电子汽车衡成为VAX计算机系统的一个共享设备,电子汽车衡应用户通过对仪表的读操作读取称重数据。由于我单位目前有VAX6320、VAX/750和MICRO VAX II三种DEC公司的中、小型机,在不同的机上,用户终端联入的方法是不同的,因此把电子汽车衡接到不同计算机上代价是不同的。例如:我单位VAX 6320计算机系统的终端都是通过终端服务器与计算机连接的,这样按终端服务器的售价,每个终端口约合人民币一万元;而VAX/750和MICRO VAX II机上终端都是通过终端接口板与主机相连接的,每个终端口约占人民币三千元。显然,我们把电子汽车衡仪表接入VAX/750或MICRO VAX II所需的投资较少。

3 硬件配置

实现该系统的硬件很简单,只需占VAX/750机终端接口一个,电子汽车衡显示仪表、RS-232接口一个及终端数据线一条。图1是它们的硬件连接图。

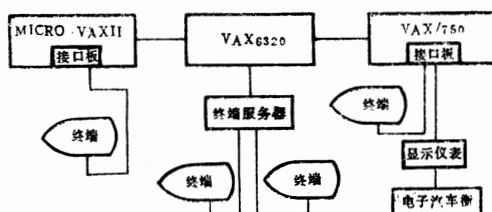


图 1

4 系统分析及软件

实现原理

当我们选择把电子汽车衡接入VAX/750机的方案后,由于电子汽车衡显示仪表是通过终端数据与主机相连,是由终端设备驱动程序控制,并且电子汽车衡仪表又采用数据连续发送方式,这样当仪表电源接通,每一时刻都有数据从输出接口通过终端数据线向终端设备驱动程序发送,这就相当于一个用户向主机系统请求登机,按照 VAX VMS 操作系统内部处理,电子汽车衡及其仪表(相当于一个终端设备)已为一个名为〈login〉的系统进程独占,从而导致电子汽车衡再也不能为其它用户程序使用,读取仪表显示的称重数据。

为了解决上述问题,我们就在系统开机时产生一个分离进程,并让该进程抢先占用电子汽车衡仪表,并使自己处于休眠状态,等待其它进程激活。这样由于仪表已为这个分离进程占用,操作系统就不再允许其它终端用户的登机请求,就不会出现〈login〉进程占用电子汽车衡显示仪表,使其它用户程序不能使用电子汽车衡的情况。与此同时,我们还可在这个分离进程运行的程序中设定当它被其它用户进程激活时的工作是:读取电子汽车衡仪表的称重数据,并把读得的数据送回激活它的进程。这样一来,我们就能够实现用户需要使用电子汽车衡时就去激活该分离进程,读取称重数据的目的。

使用电子汽车衡的用户可以不在连接汽车衡仪表的计算机上,这时用户可以通过计算机网络运行连接汽车衡计算机结点上专门用户的网络任务,由它激活前述分离进程,取得汽车衡仪表输出数据,并把数据回送给要求使用汽车衡的远程用户,达到多个计算机共享电子汽车衡的目的。

为了方便应用系统使用电子汽车衡,我们设计了一个“申请使用电子汽车衡”的软

件接口模块,这一模块与应用系统程序链接,它的工作是运行连接汽车衡结点上的网络任务,接收网络任务送回的称重数据。

实现本系统的另一件重要工作是在连接汽车衡结点计算机开工程序中包括:为连接的终端线口设置适当的参数、设备存取控制信息及保护码、限定词的DCL命令;产生分离进程的DCL命令。

5 主要的DCL命令及 部分源程序清单

5.1 开工过程中用到的DCL命令

```
SET TER/nocho/noeight(仪表7数据位)-  
/speed=仪表波特率/pari=仪表校验方式-  
TTxx(仪表占用的终端口)  
SET PROT=(o:r)/dev/owner_uic=uic  
TTxx
```

(注: uic为分离进程的用户标识码)

```
SET ACL/acl=(id=uic, access=r)-  
/object_type=dev TTxx  
SUBMIT/user=分离进程的用户名-  
/priority=255分离进程运行程序的驻留
```

盘: [驻留目录名]程序名

5.2 部分源程序清单

5.2.1 申请使用汽车衡程序段,本段插入应用系统主程序(用COBOL语言编写)。

• 数据变量部分。

¥

```
01 TXT_BUF IS EXTERNAL.  
03 ACKSTR PIC X(5).  
03 WTSTR2 PIC X(5).  
03 USED FG PIC X(14).  
01 NT_SPEC IS EXTERNAL PIC X(40).  
01 SYS_STATUS PIC S9(9) COMP.  
01 SYS_NORMAL PIC S9(9) COMP  
VALUE IS EXTERNAL SS ¥-NORMAL.  
01 RMTND PIC X(5) VALUE "GB2::".  
01 LCLND PIC X(5).
```

• 过程部分

• RMTND,连接汽车衡仪表的计算机结点名称。

• LCLND,本地计算机结点名称。

• 用户名 口令,被运行网络任务的存取控制

• 信息, 该用户拥有分离程序。

APLWGT.

• 取本地结点名到变量LCLND中程序段。

IF RMTND=LCLND GO TO LCLPRC.

*运行网络任务, 取得重量数据程序段。

MOVE' GB2' 用户名 口令' : : * TASK

— ' =网络任务名' ' TO NT_SPEC.

CALL" NETSND" USING

BY DESCRIPTOR TXT_BUF

BY DESCRIPTOR NT_SPEC

GIVING SYS_STATUS.

GO TO GOTDATA.

LCLPRC.

*激活分离进程, 取得重量数据程序段。

GOTDATA. :

5.2.2收方网络操作程序(FORTRAN语言编写)。

SUBROUTINE NETRCV(TXT_BUF)

IMPLICIT NONE

INTEGER*4 IOSTATUS

CHARACTER TXT_BUF*24

100 FORMAT(A24)

OPEN(UNIT=1, NAME=' SYS NET

1ACCESS=' SEQUENTIAL' ,

RECL=24,

1CARRIAGECONTROL=' NONE' ,

ERR=600,

1FORM=' FORMATTED' ,

TYPE=' OLD' ,

1IOSTAT=IOSTATUS)

300 READ(1,100,ERR=600,END=600,

1IOSTAT=IOSTATUS(TXT_BUF)

GO TO 800

600 TYPE • , ' IOSTATUS=' , IOSTATUS

800 RETURN

END

5.2.3收方网络应答程序(FORTRAN语言编写)。

SUBROUTINE NETACK(TXT_BUF)

IMPLICIT NONE

CHARACTER TXT_BUF • 24

INTEGER • 4 IOSTATUS

5100 FORMT(A24)

WRITE(1,100,ERR=600,

1IOSTAT=IOSTATUS)TXT_BUF

500 CLOSE(1,STATUS=' DELETE')

RETURN

600 TYPE • , ' ERROR CODE=' ,

IOSTATUS

RETURN

END

5.2.4专用网络任务程序(COBOL语言编写)。

IDENTIFICATION DIVISION.

PROGRAM-ID.WGTCOM.

ENVIRONMENT DIVISION.

INPUT-OUTPUT SECTION.

FILE-CONTROL.

SELECT WGT-FILE ASSIGN TO

" 系统盘;

- " [分离进程用户目录名]WGTFLL.DAT"

ORGANIZATION IS RELATIVE

ACCESS MODE IS DYNAMIC

RELATIVE KEY IS FRE-NUMBER.

I-O-CONTROL.

APPLY LOCK-HOLDING ON

WGT-FILE.

DATA DIVISION.

FILE SECTION.

• 分离进程传送重量及同步信息文件。

FD WGT-FILE

LABEL RECORD IS STANDARD

FILE STATUS IS FILE_STATUS.

01 WGT-REC.

03 LCKFLG PIC X.

03 WGTSTR PIC X(5).

03 PRCNAM PIC X(15).

WORKING-STORAGE SECTION.

01 FRE-NUMBERPIC 9(4).

01 FILE_STATUS PIC XX.

01 TXT_BUF IS EXTERNAL.

03 ACKSTR PIC X(5).

03 WTSTR2 PIC X(5).

03 USEDFFG PIC X(14).

01 WGTNUM PIC 9(5).

01 INUSE PIC X(11) VALUE" 地磅有人用!" .

01 PRCNAM PIC X(15).

01 SYS_STATUS PIC S9(9) COMP.

01 SYS_NORMAL PIC S9(9)COMP.

VALUE IS EXTERNAL SS¥_NORMAL.
 01 FRST_TIM PIC X.
 01 J PIC 9(4).
 01 BUFDATA PIC 9(5).
 01 IREAD PIC X(5) VALUE" IREAD."
 PROCEDURE DIVISION.
 BEG SECTION.
 BEGIN.
 MOVE IREAD TO ACKSTR.
 MOVE SPACES TO USED FG ACKSTR
 WTSTR2.
 CALL" NETRCV" USING
 BY DESCRIPTOR TXT_BUF
 GIVING SYS_STATUS.
 IF SYS_STATUS NOT=SYS_NORMAL
 THEN
 MOVE" WGTERR" TO ACKSTR
 GO TO ACKPRG
 END-IF.
 OPEN I-O WGT-FILE ALLOWING
 ALL.
 IF FILE_STATUS NOT=" 00" THEN
 MOVE" WGTERR" TO ACKSTR
 GO TO ACKPRG.
 END-IF.
 MOVE 1 TO FRE-NUMBER J.
 MOVE" Y" TO FRST_TIM.
 READ WGT-FILE REGARDLESS
 INVALID.
 MOVE" WGTERR" TO ACKSTR GO
 TO ACKPRG.
 IF FRST_TIM=" Y" AND LCKFLG =
 " 0" THEN
 PERFORM SETFLG THRU SETFLG-
 EXIT
 ELSE
 *变量LCKFLG等于" 1" 表示地磅有人使用。
 IF FRST_TIM=" Y" AND LCKFG=
 " 1" THEN
 MOVE INUSE TO USED FG
 MOVE" WTUSED" TO ACKSTR
 MOVE ZEROES TO WTSTR2
 GO TO ACKPRG
 END-IF

END-IF.
 *激活分离进程。
 CALL" SYS¥WAKE" USING
 OMITTED
 BY DESCRIPTOR BUFFER1
 GIVING SYS_STATUS.
 IF SYS_STATUS NOT=SYS_NORMAL
 THEN
 MOVE" WGTERR" TO ACKSTR
 GO TO ACKPRG
 END-IF.
 *测试汽车是否在地磅上。
 *连续称重, 检查重量是否稳定。
 .
 *称重成功, 送回重量数据。
 MOVE WGTSTR TO WTSTR2,
 MOVE SPACES TO USED FG.
 ACKPRG.
 CALL" NETACK" USING
 BY DESCRIPTOR TXT_BUF
 GIVING SYS_STATUS.
 *初始重量同步文件。
 MOVE ZEROES TO WGTSTR
 LCKFLG.
 REWRITE WGT-REC ALLOWING
 NO INVALID
 MOVE 0 TO SYS_STATUS.
 CLOSE WGT-FILE.
 STOP RUN.
 SETFLG.
 MOVE" 1" TO LCKFLG.
 REWRITE WGT-REC ALLOWING
 NO INVALID
 MOVE" WGTERR" TO ACKSTR.
 MOVE" N" TO FRST_TIM.
 READ WGT-FILE REGARDLESS
 INVALID
 MOVE 0 TO SYS_STATUS.
 SETFLG-EXIT.
 EXIT.
 *分离进程运行的程序(用COBOL语言编写).
 IDENTIFICATION DIVISION.
 PROGRAM-ID, DTCHPRC.
 ENVIRONMENT DIVISION.

INPUT-OUTPUT SECTION.

FILE-CONTROL.

SELECT WGT-FILE ASSIGN TO

" 系统盘;

- " [分离进程用户目录名]WGTFL. DAT
ORGANIZATION IS RELATIVE
ACCESS MODE IS DYNAMIC
RELATIVE KEY IS FRE-NUMBER.

I-O-CONTROL.

APPLY LOCK-HOLDING ON WGT-
FILE.

DATA DIVISION

FILE SECTION.

FD WGT-FILE

LABEL RECORD IS STANDARD

FILE STATUS IS FILE-STATUS.

01 WGT-REC.

03 LCKFLG PIC X.

03 WGTSTR PIC X(5).

03 PRCNAM PIC X(15).

✱

WORKING-STORAGE SECTION.

01 FRE-NUMBER PIC 9(4).

01 I PIC 9(4).

01 FILE-STATUS PIC XX.

01 WGTSUB-STR.

03 STRT-CHR PIC X(4).

03 WGTDATA PIC X(5).

03 FILLER PIC X(9).

01 SYS-STATUS PIC S9(9) COMP.

01 SYS-NORMAL PIC S9(9) COMP
VALUE IS EXTERNAL SS✱-NORM-
AL.

• 仪表连接终端接口设备名。

01 DEVNAM PIC X(10) VALUE" _TTC
6;".

01 CHAN PIC 9(4) COMP.

01 FUNREAD PIC 9(4) COMP
VALUE IS EXTERNAL 10✱-READV
BLK.

01 IOSB IS EXTERNAL PIC 9(18)COMP.

01 CMDSTR PIC X(10) VALUE" ALL-
TTC6;".

01 BUFLN1 PIC S9(4) COMP.

01 BUFFER1 PIC X(16).

01 ITMCOD2 PIC S9(9) COMP
VALUE IS EXTERNAL JPI✱-
PRCNAM.

01 ITEM-LIST1.

03 BUFLNG1 PIC S9(4) COMP
VALUE 16.

03 ITMCOD1 PIC S9(4) COMP.

03 BUFADR1 POINTER
VALUE REFERENCE BUFFER1.

03 RETLEN1 POINTER
VALUE REFERENCE BUFLN1.

03 FILLER PIC S9(5) COMP VALUE
0.

PROCEDURE DIVISION.

BEGIN.

MOVE SPACES TO BUFFER1.

MOVE ITMCOD2 TO ITMCOD1.

• 取分离进程名。

CALL" SYS✱GETJPIW" USING
OMITTED OMITTED OMITTED
BY REFERENCE ITEM-LIST1
OMITTED OMITTED OMITTED
GIVING SYS-STATUS.

• 分配抢占终端口。

CALL" LIB✱SPAWN" USING
BY DESCRIPTOR CMDSTR
OMITTED OMITTED OMITTED
OMITTED
OMITTED OMITTED OMITTED
OMITTED
GIVING SYS-STATUS.

CALL" SYS✱ASSIGN" USING
BY DESCRIPTOR DEVNAM
BY REFERENCE CHAN
OMITTED OMITTED
GIVING SYS-STATUS
OPEN I-O WGT-FILE ALLOWING
ALL.
MOVE 1 TO FRE-NUMBER.
READ WGT-FILE REGARDLESS

```

INVALID
MOVE 0 TO SYS_STATUS.
• 记下分离进程名给激活它的进程使用.
MOVE ZEROES TO LCKFLG WGT-
STR.
MOVE BUFFER1 TO PRCNAM.
‡
REWRITE WGT-REC ALLOWING
NO INVALID
MOVE 0 TO SYS_STATUS.
GO TO SUSPEND.
GETWT-LOP.
READ WGT-FILE REGARDLESS
INVALID
MOVE 0 TO SYS_STATUS.
MOVE 1 TO I.
ACC-LOP.
CALL "SYS‡QIOW" USING
    OMITTED
    BY VALUE CHAN
    BY VALUE FUNREAD
    BY REFERENCE IOSB
    OMITTED OMITTED
    BY REFERENCE WGTSUB_STR
    BY VALUE 18
    OMITTED OMITTED OMITTED
    OMITTED
GIVING SYS_STATUS.
MOVE WGTDATA TO WGTSTR.
WRT-REC.
REWRITE WGT-REC REGARDLESS
INVALID
MOVE 0 TO SYS_STATUS.
SUSPEND.
CALL "SYS‡HIBER" .
GO TO GETWT-LOP.
CLOSE WGT-FILE.
STOP RUN.

```

5.2.5 发方网络操作程序(用FORTRAN语言编写).SUBROUTINE NETSND(TXT

```

_BUF,NT-SPEC)
IMPLICIT NONE
INTEGER*4 IOSTATUS
CHARACTER*24 TXT_BUF
CHARACTER*40 NT_SPEC
100 FORMAT(A24)
OPEN(UNIT=1,ACCESS=' SEQ-
    UENTIAL' ,
    1NAME=NT_SPEC,FORM=' FO-
    RMATTED' ,
    1CARRIAGECONTROL=' NONE
    ' ,TYPE=' NEW' ,
    1IOSTAT=IOSTATUS,RECL=24,
    ERR=800)
300 WRITE(1,100,IOSTAT=IOSTAT-
    US,
    1ERR=600)(TXT_BUF)
READ(1,100,IOSTAT=IOSTATUS,
    1ERR=600,IOSTAT=IOSTATUS,
    1END=700)(TXT_BUF)
GO TO 800
600 TYPE *, ' IOSTATUS=' ,IOSTA-
    TUS
GO TO 800
700 CLOSE(1,STATUS=' DELETE' )
800 RETURN
END
‡

```

6 结束语

目前本系统已投入使用,效果良好。使用本系统的优点在于把电子汽车衡作为VAX计算机系统的共享设备,既方便了操作,又提高了使用效率;实现无专门操作人员的远程控制,提高了自动化程度,节省了人力;无需另建操作房,额外的终端或微电脑设备,可节省投资近二万元。参考文献略。

单片机实现电气传动系统PI调节器

天津理工学院自动化系 王恒震

(天津市南开区红旗南路221号 邮码 300191)

摘要 本文提出用一片8031单片微型计算机实现比例积分(PI)调节器的硬件软件设计方法。重点讨论了为提高运算精度和运算速度、减少存贮单元以及数字给定所具有的特色。通过运行实验表明,该电路具有良好的运行效果。单片机比应用单板机运算速度快、易于扩展、体积小和价廉的优点。

1 引言

采用MCS-51系列单片微型计算机(单片机)8031实现数字PI调节器,具有比单板机价格低、体积小、运算速度快,硬件电路易于扩展,和软件设计简单的优点。尤其是该机型具有乘除指令,可缩短控制算法的运算时间,另外它与外部设备的数据传送简单。

目前大多数电机控制是以模拟电路为基础，以运算放大器构成控制回路，调整系统时需调整电位计，有些情况下还要改变电阻和电容，元件更换必须人工完成，致使调整费时，此外电路对温度变化敏感产生漂移，运行一段时间需重新调整或更换元件。

数字式传动系统由程序来完成模拟控制回路的功能, 它对干扰、老化和温度的变化不敏感, 软件参数不会漂移, 滞后极小, 还能完成故障诊断、信息显示, 运算记录, 改变参数等功能。

今后交流和直传动系统可全部计算机化, 并可实现多种数字控制算法, 例如: 双环PID、积分分离PID、微分先行PID以及SISO自适应控制算法等。

2 单片机实现PI调节器的硬件电路

本文中的硬件电路主要由 8031、ADC 0809、和 DAC0832 组成，可用于交直流电机的调速控制，它的原理如图 1 所示。

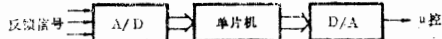


图1 硬件原理图

图2示出了具体硬件线路图。

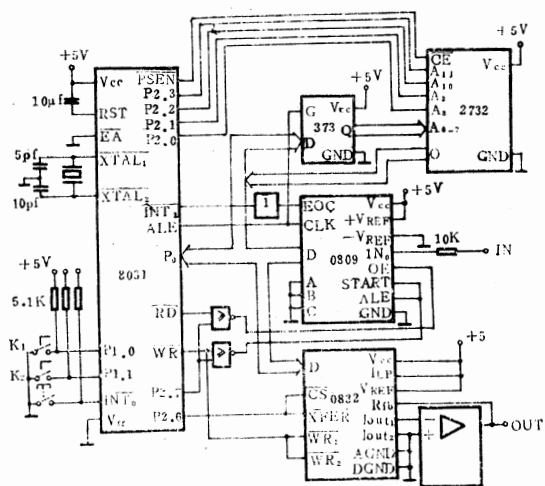


图 2

2.1 8031中 P_0 口为多用口，既作为数据输入、输出口，又和 P_2 口组成地址口。（ P_0 口为双向三态地址/数据总线）。

8031不带程序存贮器，为此扩展了一片EPROM 2732作为程序存贮器。因总线端口分时地出现地址和数据，所以用ALE信号的下降沿把地址信息锁存在地址锁存器74LS373中，随后 \overline{PSEN} 输出低电平，总线读入2732中的指令；地址的高八位出现在 P_2 口上。

P₁口作为给定值读入口, K₁、K₂分别与P_{1.0}、P_{1.1}相接, 共有四种状态作为速度给定, 当K₁、K₂选择某一转速后, 给 $\overline{\text{INT}}$ 。输入低电平, 使8031转入中断服务程序, 从

P_1 读入给定, 由于 P_1 口的许多端未占用, 还可增加调速级别。

2.2 输入与A/D转换, 在应用于单环调速时, 只使用0809的IV₀输入口, 将A、B、C均接地, 若用于多环调速, 可将A、B、C适当地接于地址线上, 即可使0809具有8个输入端。

8031对0809的控制引脚为 $P_{2.7}$, 因此0809口地址为7FFFH。当从0809输入时, 由 $P_{2.7}$ 和 \overline{WR} 信号启动0809; A/D转换结束后, 输出EOC信号启动8031的 \overline{INT}_1 , 使8031转入中断服务程序, 从0809读入数据, 并进行控制算法的运算。

2.3 运算结果的输出由DAC0832完成, 其控制端为 $P_{2.6}$, 即口地址为0BFFFH。当用于不可逆调速系统时可将其接成单极性, 若用于可逆系统可接成双极性。

3 PI基本算法及软件结构

图3为双环直流调速系统结构图, 现将虚线内速度环PI调节器部分用单片机完成, 电流环仍采用连续时间系统实现。

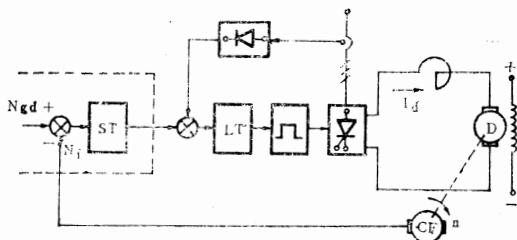


图 3

3.1 基本算法

本系统采用模拟化设计方法, 即按连续系统理论设计速度PI调节器参数, 然后将如下的微分方程转换为差分方程。

速度PI调节器的传递函数

$$G(S) = \frac{U(S)}{E(S)} = K_P \frac{\tau S + 1}{\tau S}$$

其中 K_P ——比例系数; τ ——时间常数

根据系统实测参数, 并按照典型Ⅱ型系统设计方法求得调节器的各参数如下:

$$K_P = 0.921; \tau = 0.092$$

将 $G(S)$ 写成微分方程的形式

$$\tau \frac{du}{dt} = K_P \tau \frac{de}{dt} + K_P e \quad (1)$$

由于采样周期为10ms, 小于系统最小时间常数, 所以可直接用差商代替微商的方法将其离散化,

$$\text{令 } K_I = \frac{K_P}{\tau}$$

式(1)可化为

$$\begin{aligned} \frac{U(K) - U(K-1)}{\Delta T} &= K_P \frac{E(K) - E(K-1)}{\Delta T} \\ &+ K_I E(K) \quad (\Delta T \text{ 为采样周期}) \\ U(K) &= U(K-1) + (K_P + \Delta T K_I) E(K) \\ &- K_P E(K-1) \end{aligned}$$

$$\text{令 } K_1 = K_P; \quad K_2 = K_P + \Delta T \cdot K_I$$

则有差分方程:

$$U(K) = U(K-1) + K_2 \cdot E(K) - K_1 \cdot E(K-1) \quad (2)$$

$$\text{其中 } K_1 = K_P = 0.921$$

$$\begin{aligned} K_2 &= K_P + \Delta T \cdot K_I = K_P + \Delta T \cdot \frac{K_P}{\tau} \\ &= 0.921 + 0.01 \frac{0.921}{0.092} \\ &= 1.021 \end{aligned}$$

将 K_1 和 K_2 化为二进制为

$$K_1 = 0.1110, 1011, 1100B$$

$$K_2 = 1.0000, 0101, 0110B$$

考虑到运算精度, 这里取小数点后12位, 而小数点前取4位, 再将其化为16进制:

$$K_1 = 0.E, BCH$$

$$K_2 = 1.0, 56H$$

可见, K_1 和 K_2 都是16位定点数, 小数点在高4位后。而式中 $E(K)$ 有:

$$E(K) = N_{gd} - N_f$$

数字给定 N_{gd} 和经A/D变换得到速度反

馈值 N_f 均为8位数字量。

在保证运算精度的前提下, 为提高运算速度, 关键在于加快式(2)中两次乘法运算。我们采用16位×6位乘法运算子程序, 由于系数 K_1 和 K_2 均为正数, 采用只判断

$E(K)$ 或 $E(K-1)$ 的符号来确定运算前后是否求补, 由求补子程序进行求补。乘法运算结果 $K_1E(K-1)$ 和 $K_2E(K)$ 分别存放在第一组寄存器和第二组寄存器 (R_4) (R_3) (R_2) 中(共24位)。提高运算速度的方法还有: a. 利用A/D转换的 $100\mu s$ 时间中计算一次乘法; b. 经A/D转换采集的数据和内部计算产生的中间结果均在片内 RAM 区中传送和暂存, 这样执行时间快; c. 在程序中采用了许多寄存器间址, 这类指令为单字节, 也使执行速度较快。

积的符号位存放在 R_5 , 利用 $U(K-1)$ 、 $K_2E(K)$ 和 $-K_1E(K-1)$ 求和得出 $U(K)$, (除符号位外为24位), 由于 DAC0832 为8位数模转换器, 取 $U(K)$ 的整数部分进行限幅处理后, 输出给D/A转换器, D/A转换采用单极性输出方式, 将电压信号作为电流调节器的给定。

我们采用了6MHz晶振, 单个PI运算时间约占 $840\mu s$, 为采用单片机实现双PI运算奠定了程序设计的基础。

数字给定键电路的设计也具有特色, 我们设置了四种给定状态: 0转/分, 150转/分, 750转/分和1500转/分, 经计算分别对应十六进制数的00H, 18H, 78H和EFH, 这些数据存在程序存储器的4个单元中, 由MOVC指令调取。读取给定值的工作由优先级最高的外部中断0服务程序完成。

定时器的定时时间取决于采样周期10ms, 我们采用定时器 T_1 中断服务程序, 用来启动ADC0809, 并重装定时器初值为下次采样作准备。

图4为主程序框图。主程序: 主要完成系统的初始化, 包括设定中断状态, 设定定时方式0及装入定时初值、装入 K_1 及 K_2 , 清工作单元, 踏步等待中断到来并转入中断服务程序, 图5、图6、图7分别为 $\overline{INT0}$ 、 T_1 、 $\overline{INT1}$ 中断服务程序框图。

外部中断1服务程序: 这是本程序的主题, 它完成整个PI算法的计算。当A/D转换

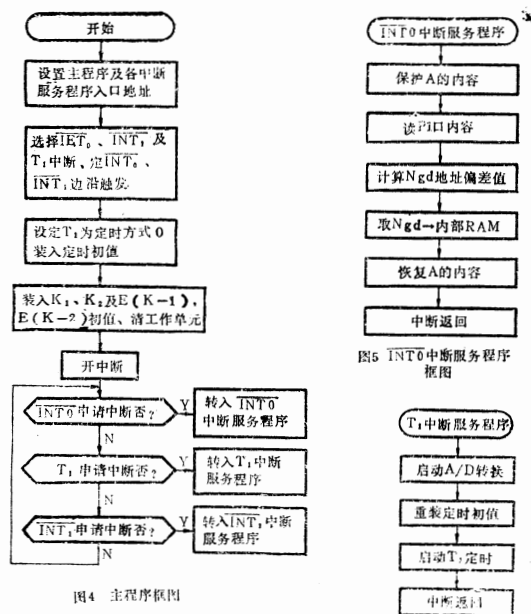


图4 主程序框图

图5 $\overline{INT0}$ 中断服务程序框图

图6 T_1 中断服务程序框图

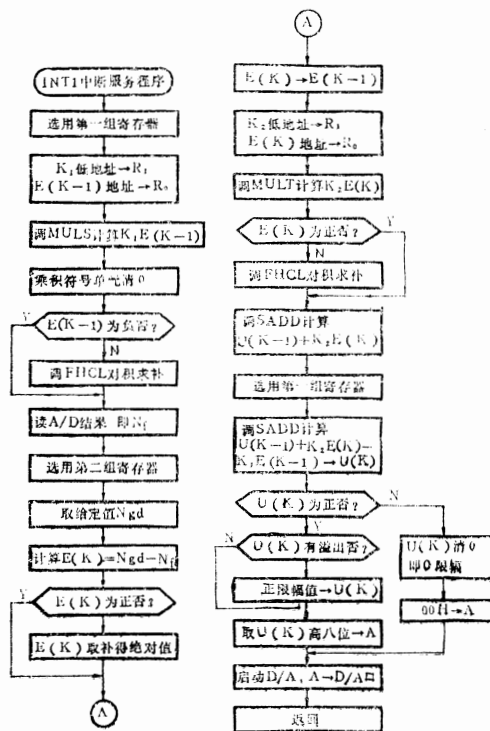


图7 $\overline{INT1}$ 中断服务程序框图

完成后, 0809的EOC产生中断请求, 转入 $\overline{INT1}$ 中断服务程序。

程序先计算 $K_1 \cdot E(K-1)$, 将结果存在第一组寄存器的 (R_2) (R_3) (R_4)

中。根据 $E(K-1)$ 的正负求取补码并将符号位存于 R_5 中, (由于此项前为减号, 乘积的符号应与 $E(K-1)$ 相反)。

其次, 读取A/D值, 计算 $E(K)$ 。为节省内部RAM空间, 将 $E(K)$ 存入原 $E(K-1)$ 单元, 其符号存于位地址00H, 并根据符号计算 $E(K)$ 的绝对值。

第三计算 $K_2 \cdot E(K)$ 将结果存在第二组寄存器的 (R_2) (R_3) (R_4) 中, 同样根据 $E(K)$ 的符号对积求补, 并将符号放在 R_5 的末位。

第四进行两次加法运算, 求取三项之和, 根据结果进行限幅处理, 并通过循环移位得到和的整数部分, 经D/A输出给电流调节器。框图中MULS为乘法子程序, FHCL为求补子程序, SADD为加法子程序。

4 实验结果

4.1 模拟PI调节实验结果

模拟PI调节的启动及干扰波形见图8(a)。

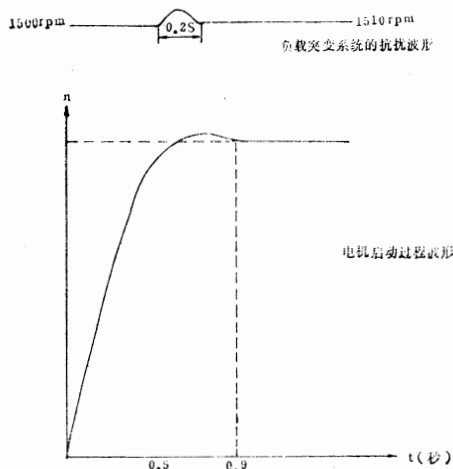


图8(a) 模拟PI调节的启动及抗扰波形
过渡时间 $t_s=0.9s$

超调量 $M_p=2\%$

静态误差 $e=0.67\%$

4.2 数字PI调节所得结果

数字PI调节的启动及干扰波形见图8(b)。

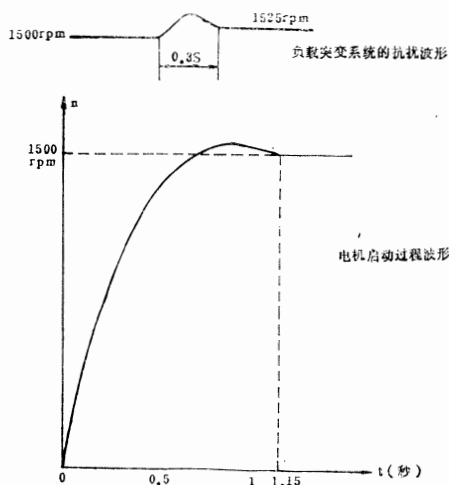


图8(b) 数字PI调节的启动及抗扰波形
过渡时间 $t_s=1.15s$

超调量 $M_p=3\%$

静态误差 $e=1.6\%$

实验结果从图8(a)、(b)的比较, 可以看出:

由于参数和采样周期选取合理, 应用单片机实现PI调节器, 取得了与模拟PI调节器接近的动静态性能指标。

5 结束语

本文所作的工作是基本的, 在此基础上, 可进一步提高控制精度、提高运算速度、增加监控功能, 同时在硬件电路变动较少的情况下实现双PI调节。

参考文献

- 1 陈伯时. 自动控制系统. 机械工业出版社, 1981, 7
- 2 郭锁凤等. 计算机控制系统—设计与实现, 航空工业出版社, 1987, 11
- 3 涂时亮, 张友德 编译. MCS-51 用户手册. 复旦大学出版社, 1989, 9

仪表的微机校验及综合管理

唐山供电公司计算中心

张汉涛 马俊杰

(邮码 063000)

摘要 微机、带GPIB接口的标准表能够组成数据采集系统。在软、硬件的合理设计下,这种系统即可以进行仪表校验,又能同仪表管理相结合,成为一个集数据采集及信息管理为一体的综合系统。本文主要介绍该系统的设计及实现方法。

利用传统手工校验方法,对仪表进行校验时,工作即费时又费力,而且数据处理也很繁琐,常常会出现一定的人为误差。目前带有GPIB接口的标准表发展很快,它能跟控制器联结形成系统。基于这个想法,我们利用微机做控制器,设计了一个仪表综合处理系统。该系统能根据仪表台账参数实时采集仪表的测量数据,并自动处理、判断,给出校验结果。系统取代了仪表的传统校验方法,实现了仪表校验的半自动化,快速,准确,可靠。另外,利用微机还实现了台帐数据、检定数据及历史记录等的综合管理及分析,达到了仪表管理的现代化。形成一个集数据采集及信息管理为一体的综合系统。

1 系统规划图

传统的校验方法是:给定仪表后,检定人员首先查阅档案确定仪表的校验程序,找到全检量程,然后根据仪表的有关等级要求(仪表原始参数)按照有关规程逐点进行观察、计算、判断,并把有关检定过程数据入账建卡,最后给出校验结果。根据业务调查、分析,系统依照如下层次进行设计,如图1。

原始数据输入:主要用来建立仪表的台帐,把仪表的有关原始参数通过键盘输入。

数据采集:由一套以微机为控制器的数据采集系统实现。按照已存入机内数据库中的有关该仪表的原始参数进行测量,把被检表的检定数据采集到计算机里。

综合处理:微机除做控制器外,还用来

对数据进行综合管理。包括两方面的工作:对采集的数据进行计算、分析、判断,给出检定结果;对台帐数据、历史记录数据、检定数据等实行现代化管理。

信息输出:通过显示器、打印机以图、表等形式给出校验结果及其它管理信息。

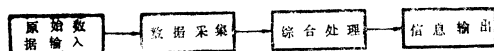


图1

2 系统硬件结构

整个系统按组合式开放结构设计。系统采用标准接口,由多个部件组合而成。微机即可以控制一台标准表,也能通过IEEE488电缆连接多块标准表,大幅度地提高系统的处理能力。另外,为减小系统误差,系统设计不使用常规的A/D卡。系统中各组成部分,即能独立工作,又能在软件的配合下协调运行。

本系统主要有微机,标准表,标准源,IEEE 488接口,测量控制面板等组成。系统联接(以检定交流电压表为例)如图2:

系统配置方案:

微机:IBM PC/XT, 8088 CPU, 640 K RAM, 360K软驱一台, 20M硬盘一台。

标准表:英国输力强公司的7151多用表。

标准源:JX1792单相源。

IEEE 488接口:MEC26001仪器接口

卡。

袖珍测量面板：联接于主机键盘的移动式测量控制开关，由一组按键组成，它同主机键盘的某些键直接相联。该面板上的键等效于主机键盘所联键的功能，并能同主机键盘并行工作。它可以在主机较远的地方使用，灵活、方便。

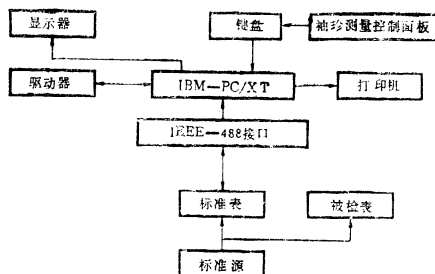


图2

3 软件结构及功能描述

软件主要实现电气仪表的微机校验及综合管理。总体结构如图3。

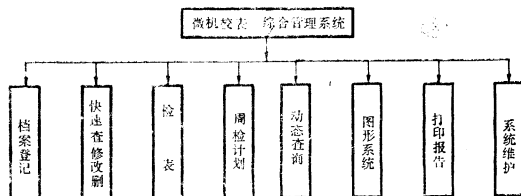


图3

新档登记：输入仪表诸如仪表名称、制造厂家、型号、规格、出厂编号、精度、检定周期、固定资产等有关参数，建立仪表的台帐。

查修改删除：按统一编号对档案、检定记录、历史记录卡、进行查找、修改、删除、打印等。

检表：按部颁规程，对仪表进行校验。

做周检计划：根据仪表的检定周期及使用情况做出年度检表计划。

动态查询：按任意条件对档案、检定记录、历史记录卡，进行查找、修改、删除、打印等。并且打印表格的表栏可以动态组合（即动态制表）。

图形系统：对档案数据、检定记录数据

等做图形分析。

打印表格：打印计量部门需要的固定格式的表格。

系统维护：用来维护系统正确运行，包括索引、数据备份、字典库管理等。

软件使用编译dBASEⅢ，编译BASIC、测试语言（本系统为测试TBASIC）及汇编等语言编码，利用动态信息词典、动态查询、动态制表打印、动态屏幕等先进技术编程。程序设计、文档等要模块化、规范化。

4 数据结构设计

系统从数据规范化、易维护性等出发，根据数据的性质及作用，数据结构设计如图4（以数据库的形式表示）：

档案库：系统的核心数据，包括仪表的原始参数、检定数据、历史记录等三大类基础数据。

说明库：档案库的结构说明。是档案库中数据的各种物理说明，包括：字段名、字段类型、长度、小数位、汉字描述、录入标记、修改标记、词典标记等。该类数据规范，变化少。主要用于提高基础数据的处理效率。

词典库：它存放某些关键字段的标准词汇及联想词汇，主要解决汉字输入的瓶颈问题，另外也保证了输入数据的规范化及一致性。

屏幕库：用来存放运行过程中菜单、提示信息、环境参数等。

表栏库：存放类似打印表格表栏的固定格式数据。

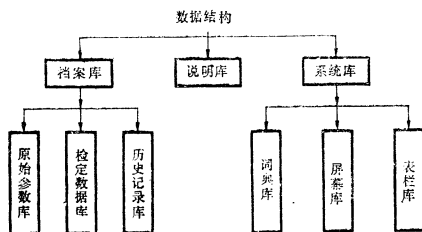


图4

5 计算方法

以0.5级仪表为例

$$5.1 \text{ 引用误差: } R = \frac{A - A_0}{A_m} \times 100\%$$

其中: A——被检表分度值

A₀——标准表数值与分格常数乘积

A_m——被检表满刻度分格数

R——引用误差

$$5.2 \text{ 修正值: } X = -(A_0 - A)$$

5.3 最大基本误差: 取绝对值最大的引用误差

$$5.4 \text{ 变差: } Y = \frac{|Y_1 - Y_2|}{|A_m|} \times 100\%$$

其中: Y₁——所测分度的升值

Y₂——所测分度的降值

Y——该刻度的变差

5.5 最大升降变差: 取最大的变差

5.6 化整: 若在拟舍异的数字中, 左边第一个数字小于5则舍去; 大于5(包括5的右边有效数字)则进一位; 恰等于5时, 若保留数字的末位数是偶数则舍去, 是奇数则进一位。

6 结束语

《微机检表、综合管理系统》是依照上

述方案设计的一个系统, 目前已在我局修试所投入运行。系统主要实现了电气仪表校验的半自动化及仪表管理的现代化。该系统取代了电气仪表传统的手工校验方法, 微机实时采集数据, 并及时对数据进行计算、化整和判断, 快速给出仪表的检定结果, 自动把有关数据分类建档, 实现了仪表检定的半自动化及智能化, 快速, 准确, 可靠。整个过程符合国家计量标准。该系统还实现了台帐数据、检定数据及历史记录等的综合管理及分析。能对台帐任意查询、修改、删除及增加等, 能随意检索仪表的有关历史数据, 进行各种统计、分析, 并能输出计量部门要求的各种报表。

参考文献

- 1 凌连生等编译. 新颖关系数据库管理系统 汉字dBASEⅢ. 中科院计算所, 1987
- 2 7151 COMPUTING MULTIMETER OPERATING MANUAL Solartron Instruments 1984
- 3 张福炎等. IBM PC的原理与应用(续篇). 南京大学出版社, 1985
- 4 夏东涛. Clipper dBASEⅢ语言编译工具. 中科院科海培训中心, 1986
- 5 仪器控制器接口TBASIC语句. 北京微电子技术应用研究所, 1985

新型导光路

日本电气公司最近开发出新型导光路, 为未来研制可以同时处理大量信息的超高速并列电子计算机创造了条件。

导光路与仅在两端有出入口的光导纤维不同, 它具有许多出入口, 单个光信号可以通过它传给多个受光元件, 因此人们称它是传导光信号的“公共汽车”。

该公司开发的导光路比迄今已有的导光路功率高10倍以上, 并且信号的出入口可以大幅度增加。

导光路的原理是在平面玻璃上装有许多凹透镜, 光信号进入凹透镜通过玻璃板从其他凹透镜出来, 传到各演算线路。这样可由并列的电子计算机进行处理。

新型导光路还可以将不同波长的光信号同时向各演算线路传送, 从而可大大提高信息传送和处理能力。

C语言直接操作DBF数据库的方法与实现

石家庄军械工程学院 谷德桥 马英忱 冯勋欣

摘要 本文在对DBASE III数据库进行剖析的基础上,提出了一种用C语言直接对DBF数据进行数据读取和记录处理的方法,并结合实例,说明了C语言编程的步骤和技巧。

1 引言

C语言不仅具有很强的通用程序设计功能,而且还有很强的可移植性。对于图像图形等较复杂的问题, C语言确实都是一种强有力的工具。但它对于大量数据信息的处理,不如DBASE III语言。如将C语言和D-BASE III语言相互调用,取长补短,一个关键的问题就是用C语言对DBF数据文件进行识别和数据处理。因此,建立一套C语言对D-BF数据库直接读取和处理的方法是十分重要的。

2 DBAS III 数据库文件的结构剖析

DBF数据库磁盘文件是以二进制数字和ASCII字符相结合的形式存储的。表1和表2列出了DBF数据库文件结构说明部分的内容。其中,表1包括了文件的标识符, 建库时间, 记录个数, 结构说明部分的总字节数等, 长度共为32字节。表2进一步说明每个字段的有关信息, 包括字段名称、类型、长度和小数位个数等。每个字段占32个字节。在字段描述之后, 有一字节(0DH)表示了文件数据结构说明部分的结束和第一个记录数据的开始。数据按记录和字段依次排列, 中间无分隔符。每个记录的开始字节是记录删除标识。当该字节为(20H)时, 表示不删

除。当该字节为(2AH)时, 表示有删除。数据之后为一个字节的文尾标识(1AH)。

表1 数据库文件结构说明

位置(字节)	长度	说 明
0	1字节	前两位为版本号
1—3	3字节	建库日期(年月日)
4—7	4字节	记录个数
8—9	2字节	结构部分字节数
10—11	2字节	记录字节个数
12—31	20字节	保留
32—(n—1)	每32字节	字段描述部分(表2)
n	1字节	说明部分结束标志 0DH

表2 字段描述

位置(字节)	长度	说 明
1—10	11	字段名, ASCII 字符
11	1	字段类型, ASCII 字符
12—15	4	保留
16	1	二进制表示字段长度
17	1	二进制表示小数位数
18—19	2	保留
20	1	工作区标识ID
21—31	11	保留

从表中可以看出, DBF数据库文件主要包括两部分内容。第一部分是从开始到(32+32* 字段数+1)个字节。第二部分是数据部分, 它是按照第一部分的数据结构说明依次有序排列的。这样可以通过读取第一部分

的结构信息,在第二部分找到所需的数据位置和内容。

3 语言直接读取DBF数据库文件的方法

3.1 打开DBF数据库 由于DBF数据库是以数字和ASCⅡ字符相结合的方式存储文件的,而且需要对数据库进行读写操作,因此应采用以读写二进制方式打开文件,即

```
fp=fopen("filename.dbf",  
"r+b");
```

3.2 读取结构说明信息 首先按表1和表2的结构说明来定义两个结构用以存放数据库结构描述信息。

```
Struct dbf1{  
    char    flag;  
    char    date[3];  
    unsigned long record-num;  
    unsigned    n1-long;  
    unsigned    record-long;  
    char    unse[20];  
};  
Struct dbf2{  
    unsigned char f-name[11];  
    char    f-type;  
    char    unsel[4];  
    char    f-long;  
    char    decimal;  
    char    unse2[2];  
    char    work-area;  
    char    no3[11];  
};
```

利用fread函数,将文件的数据结构说明部分的全部信息读入结构之中。

```
fread(结构1指针, 32, 1, fp);  
fread(结构2指针, 32, 字段个数,  
fp);
```

3.3 读取数据 在读取数据之前,先用fseek函数确定要读取数据指针的位置,再用fread函数将数据读到内存。其中字段个数为:

(数据结构说明部分字节数-32-1)/32

4 C语言对DBF数据库的修改、删除和插入

4.1 修改DBF数据库 对于DBF文件的某记录或记录中的某字段内容,可以采用如下步骤进行修改。首先,用fseek函数确定记录指针位置,然后利用fwrite函数写入新的内容。注意当要写入的内容少于记录所规定的字节长度时,需加空格填满。

4.2 删除DBF数据库的记录 首先定义一块内存空间,把要删除的内容后面的全部数据读入该内存空间,再将指针移到删除的开始位置,将上述的内容写回文件之中,并将文件的记录个数减1。

4.3 插入DBF数据库记录 首先定义一块内存空间,将要插入的位置后面的全部数据放入内存,然后在此位置上写入新的记录内容,再将指针移至下一个记录开始处,把内存的数据写回文件中,最后修改记录数。

完成上述修改、删除和插入等功能需要注意以下几个问题:

- 对数据库进行操作的最后一步,要修改建库日期并注意文件尾(1AH)的位置是否正确。
- 当需要开辟内存空间传递数据时,如数据量过大,可开辟适当的内存空间,分步将数据传递到文件的相应位置。
- 记录是按结构说明的字节长度依次排列的,如某记录中的数据内容比规定的字节数短,文件中是用空格填满的,故在对数据操作时,一定要注意记录的完整性。

5 举例

现用Turbo C 1.5对用Foxbase建立的数据库进行打开和修改操作。首先任意建立一个DBF数据库,然后调用本程序,再用Foxbase语言观察所获得的结果。

可以在此基础之上,增加删除、插入以

及特殊要求的功能, 使该软件系统化。

6 源程序

```
#include <stdio.h>
#include <alloc.h>
#include <string.h>
#define FIELD_MAX 10
struct dbf1 {
    char        flag;
    char        date[3];
    unsigned long record_num;
    unsigned    nl_long;
    unsigned    record_long;
    char        unse[20];
};
struct dbf2 {
    unsigned char f_name[11];
    char        f_type;
    char        unsel[4];
    char        f_long;
    char        decimal;
    char        unse2[2];
    char        work_area;
    char        no3[11];
};
void f_open(char *);
void f_replay(unsigned int, char *);
FILE *fp;
unsigned int field_max;
struct dbf1 no1;
struct dbf2 no2[FIELD_MAX];
void f_open(char *fname)
{
```

```
fp=fopen(fname, "r+b");
if(fp==NULL) {printf( "\ncannot
open dbf-file" );exit(0);}
fread(&no1, 32, 1, fp);
field_max=(no1.n'-long-32-1)/32;
fread(&no2, 32, field_max,fp);
}
void f_replay(unsigned int num,char *
cc)
{int i,char *tt;
if(num>no1.record_num){printf( "\nno
the record, " )exit(0);}
tt=malloc(no1.record_long+1);
strcpy(tt,cc);
for(i=strlen(cc);i<no1.record_long;i+
+) strcat(tt, " ");
tt[i]='\0';
fseek(fp,(num-1)*no1.record_long+
no1.nl_long,0);
fwrite( " ", 1, 1,fp);fwrite(tt,no1.
record_long-1, 1, fp);
free(tt);
}
void main( )
{
f_open( "dw.dbf" );
f_replay(3, "aaaaaaaaaa" );
fclose(fp);
}
```

参 考 文 献

- 1 沈金发, 郑甬京. 新一代高性能关系数据库系统汉字 dBASE IV. 清华大学出版社
- 2 徐金梧等编译. Turbo C使用大全. 北京科海培训中心

一种以IBM PC/AT来控制 多总线图象分析系统

浙江大学图象研究室 叶红 霍强 唐慧明 顾伟康

(邮码 310027)

摘要 本文结合一台图象分析系统,给出了以IBM PC/AT为主控机的Multibus系统的设计,讨论了IBM PC/AT和Multibus之间的时序配合及有关的总线扩展技术。该方案可广泛地应用于以Multibus为总线的系统中。

1 引言

由美国INTEL公司提出的Multibus被推荐为国际标准IEEE—796总线,运用此总线,可极为方便地构成多微处理机系统,实现分布处理,多重处理和并行处理,它与其他总线相比,具有应用面广,功能齐全,布线合理,适应性强等优点。因此,它已成为微处理机系统设计中广泛使用的一种工业标准总线。

我国“七·五”期间定点研制的小型图象分布系统,其控制总线就是采用Multibus总线标准,其中各个功能模块的控制接口都是按Multibus的规范来设计。该系统的总体结构如图1所示[2]。

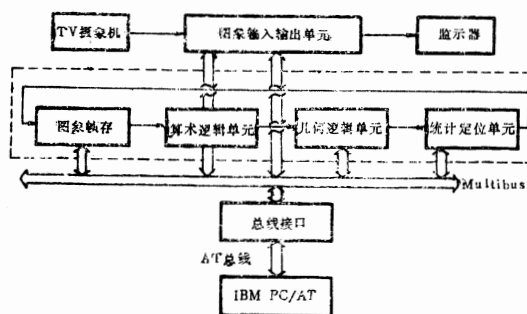


图1

由于IBM PC系列微型机在国内已被广泛使用,为了便于推广图象系统,设计成可以采用IBM PC/AT作为主控制机的形式。

但是由于AT机的总线规范与Multibus的不一样,AT机本身也不具备足够大的驱动能力,所以不能将AT机与图象分析系统直接相连,在其间必须加上一个总线接口。本文将给出这样一个总线接口的设计。

2 系统总线接口概述

IBM PC/AT与Multibus之间的总线接口由三部分组成,总线发送单元,总线接收单元及通信电缆,总线发送单元板插在AT机的I/O扩充槽内,总线接收单元板插在图象系统主计算机的系统插槽内,其间用标准的通信电缆通过连接器连接。结构示意图如图2。

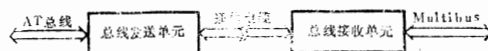


图2 总线接口示意图

这种总线接口主要解决三个问题:

解决AT总线信号与Multibus标准的总线信号间的配合。(包括逻辑电平的相互转换及时序配合)等;消除信号通过电缆传输而引起的干扰;以及为Multibus提供一定的驱动能力。

2.1 总线发送单元

总线发送单元的作用是将AT机输出的地址、数据和控制信息进行缓冲放大,经通信电缆发送到接收单元,同时接收从接收单

元通过电缆发来的控制信号，状态信号及数据。另外总线发送单元的控制逻辑产生一些控制信号，使得AT机能适应Multibus的工作方式及满足其时序要求。图3是总线发送单元框图。

总线发送单元包括这样几个部分：控制总线缓冲和地址总线缓冲部件，控制总线驱动和地址总线驱动部件，数据的传送及接收部件，响应信号接收器及总线控制逻辑部件，还包括MEM和I/O译码器，时序控制，总线方向控制，延时和超时控制等。

从PC/AT的I/O通道送出的地址信号，控制信号不是直接送往通信电缆，而是经过缓冲器后，送往驱动器，然后送到通信电缆，这样做可减少电缆电容的影响和减少信号畸变。

总线控制部件包括了存储器I/O译码电路，此电路用来识别目前的操作是否针对多总线的。延时和超时控制电路，是用于产生等待状态和结束总线信号。总线收发方向和使能控制电路用来产生数据总线是收还是发的控制信号。时序控制逻辑电路，产生Multibus EM信号，以表示目前的操作是针对Multibus总线的。

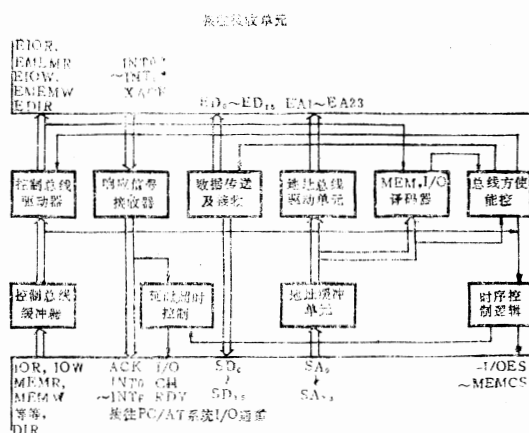


图3 总线发送单元框图

2.2 总线接收单元

总线接收单元接收从通信电缆来的系统控制信号和地址信号，经转换后去驱动Multibus系统通道。总线发送单元将Multibus的响应信号和中断请求信号经转换后去驱动

通信电缆，数据收发器用来管理数据的读写，控制逻辑电路产生此电路所需的逻辑信号，以实现接收单元的正常工作的。图4为总线接收单元框图。

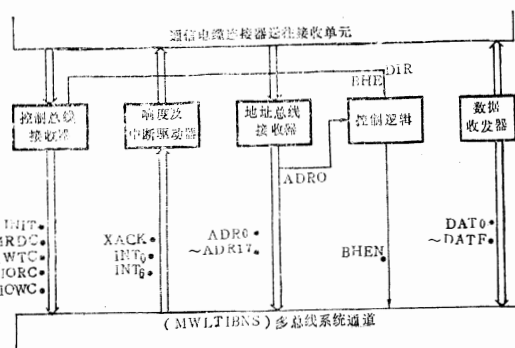


图4 总线接收单元框图

3 Multibus与AT总线的时序配合

3.1 Multibus读写时序及要求

图5a和图5b分别为Multibus的读写信号的波形。从图中可以看出：

3.1.1 地址信号和写数据必须在命令有效之前50ns已经稳定，并在命令结束后至少还要保持50ns以上，这是作为主控器能可靠工作的必要条件。

3.1.2 读数据和BACK信号必须在命令结束后不超过65ns的时间内，变为浮动状态，这是作为受控器能可靠工作的必要条件。详细的说明可见[1]

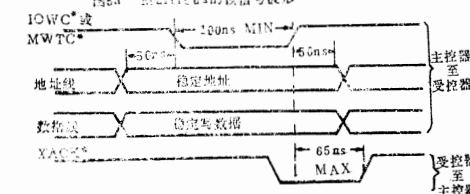
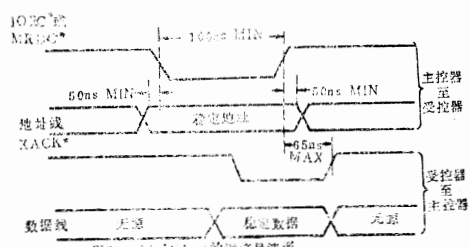


图5b Multibus的写信号波形

3.2 AT总线读写时序及配合

我们详细分析了IBM PC/AT机的硬件电路图,得出了它在I/O通道上的有关读写时序图,其时序非常复杂,分为许多种情况。主要取决于以下信号的使用情况:—MEMCS16, —I/OCS16 OWS, I/O CH RDY及存储器操作的对象。

在我们的设计中,使用—MEMCS16, —I/OCS16,来表明对Multibus的操作处于等待传输状态,同时也禁止了IBM PC/AT内部的切换245,避免了总线冲突。这样当进行16bit传输时,是指16bit同时传输,而不是分成两个8bit传输。然而,总线周期至少要插入一个等待状态,在一个等待状态以后,总线周期是否结束还取决于Multibus*的ZACK信号,由它来影响I/O CH RDY信号。

这种时序是满足IBM PC/AT作为Multibus主控器所要求的。即对于存储器操作,写数据建立时间为 $T_p/2$ (T_p 为AT机的系统时钟周期),所以,AT机的最高时钟频率可达10MHZ,仍能满足Multibus的速度要求,对于I/O操作,写数据建立时间为 T_p ,AT机的最高时钟频率可达20MHZ,仍可满足Multibus的速度要求。本设计中的总线时序图从略。

3.3 延时与超时电路

Multibus总线周期结束必须依靠XACK*信号,IBM PC/AT中利用I/O CH RDY信号可以插入附加的等待状态,延长总线周期,以适应慢速装置的要求。延时电路的作用就是利用XACK*信号去驱动等待状态产生器逻辑,结束一个80286总线周期。但是,如果80286所选的地址对应于一个Multibus上并不存在的设备的话,XACK*信号永远也不会产生,如果没有总线超时保护电路,80286就会无限等待有效的Ready信号而陷入“死机”。因此必须加入超时保护电路。电路图从略。

在IBM PC/AT中规定,I/O CH RDY线为低电平的时间不能超过2.5ms,因此,利用这个时间参数,设计一个单稳多谐振荡电路。定时电阻,电容的选择应满足: $0.7 RC \leq 2.5ms$,来满足超时电路的要求。

4 数据总线收发

4.1 Multibus中的数据传送方式

在Multibus系统中,可以允许16位处理器8位处理器混合使用。为此,在有16位数据总线的主控器,受控器板上安装了交换缓冲器,当只存取数据的高8位时,可通过交换缓冲器,将高8位的数据输出到DAT0—DAT7线上,再借助于DAT0—DAT7进行数据的传输。

16位和8位数据的交换,由ADR0*和BHEN*信号控制,BHEN*表示进行16位存取信号[1]。与AT机中的SBRE的定义不同。表中是BHEN*与AT机系统总线上SA₀与SBRE的关系。

从此表可得出: $BHEN^* = SA_0 + \overline{SBRE}$

SA ₀	\overline{SBRE}	BHEN*	存取形式
L	H	H	存取低8位
H	L	H	存取高8位
L	L	L	存取16位(偶地址)

80286对于奇地址字传送分成两个总线周期进行,第一个周期在D15—D8上传送数据,第二个周期在D7—D0上传送数据,这样两个字节数据的传输是自动发生的,对软件是不透明的。这种工作方式同Multibus的要求也是相符的。

4.2 扩展单元中数据通道的设计

在IBM PC/AT中,I/O通道的所有8位装置与微处理器间的通信使用D0—D7,16位装置将使用D0—D15,为了支持8位装置,在奇

数组和文件中直接生成图形数据的绘图方法

同济大学

邓长根

(上海四平路1239号 邮编 200092)

摘要 本文提出在内存数组和外存文件中直接生成图形数据,再输出到图形打印机打印图形的方法,具有以下优良功能:若无高分辨率图形屏幕但有图形打印机,仍能打印出光滑无锯齿现象的、大小可任意调整的(包括局部图形、超大图形)、横竖比例保真的图形,速度比屏幕拷贝方法快得多。

1 前言

屏幕拷贝是保存屏幕图形资料的一种方法。虽有屏幕拷贝的几种高级用法,可实现部分拷贝、超大图形的拷贝等^[1],但屏幕拷贝仍有几方面需改进。首先,要求计算机配备高分辨率图形屏幕,才可能绘出光滑曲线。其次,屏幕拷贝实质上是对屏幕图素实行扫描,速度较慢,当图素布满屏幕但分布较稀时,屏幕扫描显得很经济。第三,屏幕拷贝出的图形横竖比例可能失调,影响图形质量。第四,屏幕拷贝不便于改变图形大

小;拷贝超大图形要用专门的程序实行分屏显示,逐屏拷贝^[1],甚感不便。

本文提出的在内存数组和外存文件中直接生成图形数据、再输出到图形打印机打印图形的方法改进了上述屏幕拷贝的不足之处。

1.1 根据打印机映象图形指令的要求,直接在内存数组和外存文件中生成图形数据,无需图形屏幕更无需高分辨率图形屏幕作为中间媒介,只要选择打印机高分辨率图形工作状态,就能打印出光滑无锯齿现象的图形。

1.2 直接生成图形数据的方法,比先在屏

地址字节传送时,D8至D15上的数据至8位传到这些装置过程中切换到D0—D7,反之亦然。其内部电路如图6。

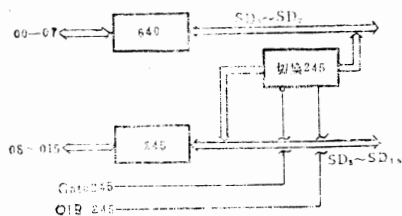


图6 IBM PC/AT内部数据切换电路

在我们的设计中,利用—MEM CS16信号和—I/O CS16信号来禁止掉IBM PC/AT内部的切换245,使得Multibus操作时,在AT系统总线上奇地址字节传送通过SD8—

SD15来进行,同时,这样在进行16bit传输时(偶奇址)不分成两个8bit来传,而是16bit同时传输,缩短了总线周期。据此设计的数据总线收发电路从略。

参考文献

- 1 J.B.Johnson,S.Kassel《多总线设计手册,结构,总体设计与应用》科学出版社,1987
- 2 杨杏生,李惠超等.“图象系统中视频流水线”.“模式识别与人工智能”第二卷第三期,1989.3
- 3 《IBM PC/AT技术手册》
- 4 L.C.Eggebrecht《IBMPC接口技术》.海洋出版社等

幕上绘图再扫描屏幕图形拷屏打印的方法快得多；当图形较大但图素分布较稀时，如绘框架轴线图，本方法只“画”点不扫描，效率更高；当图形不大时，可直接在内存数组中生成图形数据，减少内外存数据传送，速度更快。

1.3 对于打印机的不同图形工作状态，分别设置不同的图形横竖向调整比例系数，保证图形不失真。

1.4 对于打印机的不同图形工作状态，分别设置不同的参数DOTPMM；横向每mm范围内打印点数，因而可以绘出给定宽度和高度的图形，不难产生厘米纸上绘图的效果；输入图形范围，可实现局部图形打印；绘超大图形时，程序自动将图形分块，进行内存数组与外存文件之间的图形数据交换，无需另编专用程序。

本文程序针对图形打印机 EPSON LQ—1000 (K) 用 FORTRAN 语言写成，实际应用时，不难针对其他类型图形打印机，用其他计算机语言实现。

2 方法要点和主程序

2.1 设置打印机图形工作状态参数

在打印机图形工作状态参数设置文件 PRINTER.CFG 中，NPMODE 为打印机图形工作状态种数，接下来是 NPMODE 行数据，每行数据包括：

PLMODE——打印机图形工作状态号，参见有关手册；

DOTPMM——水平（宽度）方向每 mm 范围内打印点数；

RATIO——垂直（高度）方向调整比例，以确保图形不失真。

PRINTER.CFG 例：

```
File, PRINTER.CFG
NPMODE
6
PLMODE DOTPMM RATIO
0      2.369668  1.0000000 :
      8-dot mode & single-density
2      4.725898  0.5026128 :
      8-dot mode & double-density
4      3.154574  0.7529691 :
      8-dot mode & CRT-graphics
32     2.364066  3.0214286 :
      24-dot mode & single density
33     4.716981  1.5248571 :
      24-dot mode & double-density
38     3.546099  2.0142857 :
      24-dot mode & CRT-graphics
```

2.2 在数组和文件中生成图形数据

打印机的映象图形指令包括两部分，第一部分选择图形工作状态，第二部分为图形数据。要打印出所希望的图形，关键在于产生正确的图形数据。本文方法的主要特点就是在内存数组和外存文件中直接生成图形数据。

当图形较小时，可在内存数组中生成图形数据（简称画图）；当图形较大时，内存数组因受内存容量限制不能开很大，无法全部在内存画图。此时，可将图形分块，逐块在内存画图，非当前块中的图形数据存入文件。图块的大小由输入参数确定。

图形数据中的点阵信息存于字符中，字符 ASCII 码的二进制数的 8 位数分别代表 8 个点，数 ‘1’ 对应的点打印，否则不打印（详见手册），因此内存数组为字符型。外存文件为二进制随机无格式文件，存放字符，记录长度由输入确定。

2.3 图数边界参数

画图时，本文方法还分别测定图形的边界，IXL、IXR 分别为宽度方向的左边界、

右边界, IYT、IYB分别为高度方向的上边界、下边界。测定图形边界有以下作用或优点:

2.3.1 打印整个图形时, 打印有效(非空)图形, 避免打印头走空, 提高打印效率;

2.3.2 可输入IXL、IXR、IYT、IYB, 打印局部图形;

2.3.3 计算打印点列数: $IX = IXR - IXL + 1$, 由此计算图形数据的前2个字节: $N1 = MOD(IX, 256)$, $N2 = IX / 256$ 。

2.4 主程序清单及说明

示范主程序 FPDemo 清单如下。第7~11行读入打印机图形状态设置参数。第12~18行选择处理图形数据的方式: MODE=0, 结束运行; MODE=1, 生成图形数据后

立即打印图形; MODE=2, 生成图形数据存入文件; MODE=3, 从文件中读入图形数据打印图形。第19~24行输入图形分块大小 LINECH、打印机图形状态序号 IPMODE、纸宽 PWIDTH 和纸高 PHEIGHT (以 mm 为单位), 计算图形宽度 WIDTH 和高度 HEIGHT (以点数为单位)。第29~43行初始化图形边界、内存数组 FPD 和外存文件 FPDemo.PLT。第44行调用的示范子程序 FDRECT (略) 利用本文方法画宽为 PWIDTH、高为 PHEIGHT 的矩形, 可用其他绘图子程序替换。第27、45行子程序 PLT-PRN 根据 MODE 取值分别完成打印或存盘。根据应用程序的实际情况, 数组 FPD(1200, 200) 可改为 FPD(L1, L2), 要求 $WIDTH \leq L1$, $LINECH \leq L2$ 。

主程序清单

```

1:      PROGRAM FPDemo
2:      CHARACTER *1 FPD
3:      INTEGER PLMODE, WIDTH, HEIGHT
4:      DIMENSION PLMODE(10), DORPMM(10), RATIO(10)
5:      COMMON/FPLTDT/FPD(1200, 200), LINECH, LINEDT, LINEIP, IBL-
      OCK, MBLOCK
6:      COMMON/PLTBND/IXL, IXR, IYT, IYB
7:      OPEN(2, FILE='PRINTER.CFG', STATUS='OLD')
8: 10     READ(2, *, ERR=10) NPMODE
9:      DO 30 I=1, NPMODE
10: 20    READ(2, *, ERR=20) PLMODE(I), DOTPMM(I), RATIO(I)
11: 30    CONTINUE
12: C     MODE=0: Finish running this program
13: C     MODE=1: Print plot just after forming plot data
14: C     MODE=2: Save plot data into plot data file
15: C     MODE=3: Print plot from plot data file
16: 40    WRITE(*, '( ' MODE, ' ' < )' )
17:      READ(*, *) MODE
18:      IF(MODE.EQ.0)GOTO 100
19:      WRITE(*, '( ' LINECH , IPMODE , PWIDTH (mm) , PHEIGHT
      (mm): ' ' < )' )
20:      READ(*, *)LINECH, IPMODE, PWIDTH, PHEIGHT
21:      LINEDT=LINECH*8
22:      MODEIP=PLMODE(IPMODE)
23:      WIDTH=INT(PWIDTH*DOTPMM(IPMODE)+0.5)
24:      HEIGHT=INT(PHEIGHT*DOTPMM(IPMODE)*RATIO(IPMODE)+0.5)

```

```

25:      IF(MODE.EQ.3) THEN
26:          OPEN(3, FILE= ' FPDEMO.PLT' ,STATUS= ' OLD ' ,FORM= '
          BINARY' ,
1      ACCESS= ' DIRECT' , RECL=WIDTH)
27:          CALL PLTPRN(MODE, MODEIP)
28:      ELSE
29:          IXL=1200
30:          IXR=0
31:          IYT=1200
32:          IYB=0
33:          OPEN(3, FILE= ' FPDEMO.PLT' , STATUS= ' NEW' , FORM= '
          BINARY' ,
1      ACCESS= ' DIRECT' , RECL=WIDTH)
34:          DO 50 I=1, LINECH
35:              DO 50 J=1, WIDTH
36: 50          FPD(J, I)=CHAR(0)
37:              DO 60 IBLOCK=2, (HEIGHT-1)/LINEDT+1
38:                  LINEIP=LINEDT*(IBLOCK-1)+1
39:                  DO 60 I=1, LINECH
40: 60          WRITE(3, REC=LINEIP+I) (FPD(J, I), J=1, WIDTH)
41:                  IBLOCK=1
42:                  MBLOCK=IBLOCK
43:                  LINEIP=1
44:                  CALL FDIRECT(WIDTH, HEIGHT)
45:                  CALL PLTPRN(MODE, MODEIP)
46:      ENDIF
47:      GOTO 40
48: 100      STOP
49:      END

```

3 画点子程序

画点子程序FPOINT使用环境由FPDEMO设定，程序清单如下：

```

1:      SUBROUTINE FPOINT(X, Y, COLOR)
2:      CHARACTER*1 FPD
3:      INTEGER X, Y, COLOR
4:      COMMON/FPPLTDT/FPD(1200, 200), LINECH, LINEDT, LINEIP, IBLO-
          CK, MBLOCK
5:      COMMON/PAPBND/IXL, IXR, IYT, IYB
6:      IF(X.LT.IXL) IXL=X
7:      IF(X.GT.IXR) IXR=X
8:      IF(Y.LT.IYT) IYT=Y
9:      IF(Y.GT.IYB) IYB=Y
10:      LBLOCK=(Y-1)/LINEDT+1

```

```

11:      IF(LBLOCK.NE.IBLOCK) THEN
12:          DO 10 I=1, LINECH
13:      10  WRITE( 3, REC=LINEIP+I) (FPD(J, I), J=1, IXR)
14:          IBLOCK=LBLOCK
15:          MBLOCK=IMAX(IBLOCK, MBLOCK)
16:          LINEIP=LINECH*(IBLOCK-1)+1
17:          DO 20 I=1, LINECH
18:      20  READ( 3, REC=LINEIP+I) (FPD(J, I), J=1, IXR)
19:      ENDIF
20:      L1=MOD(Y+7, 8)
21:      IL=2*(7-L1)
22:      I=(Y-LINEIP)/8+1
23:      IC= ICHAR(FPD(X, I))
24:      IF(COLOR.NE.0.AND.MOD(IC/IL, 2).EQ.0) IC=IC+IL
25:      IF(COLOR.EQ.0.AND.MOD(IC/IL, 2).EQ.1) IC=IC-IL
26:      FPD(X, I)=CHAR(IC)
27:      RETURN
28:      END

```

子程序形参中, X、Y为点坐标, COLOR为颜色。图形坐标起点在纸张左上角, $1 \leq X \leq \text{WIDTH}$, $1 \leq Y \leq \text{HEIGHT}$ 。由于打印图形为单色, COLOR=1表示画点, COLOR=0表示删点。

第6~9行测定图形边界。第10~19行判断所画点是否位于当前图块(图形数据存贮于数组中的图形), 若不位于当前图块, 则将当前图块存入文件, 再从文件中读入新的当前图块。第20~26画点或删点。

```

1:      SUBROUTINE PLTPRN(MODE, MODEIP)
2:      CHARACTER*1 FPD
3:      COMMON/FPLTDT/FPD(1200, 200), LINECH, LINEDT, LINEIP, IBLOCK, MBLOCK
4:      COMMON/PAPBND/IXL, IXR, IYT, IYB
5:      WRITE(3, REC=1) IXL, IXR, IYT, IYB
6:      DO 100 I=1, LINECH
7:      100  WRITE(3, REC=LINEIP+I) FPD(J, I), J=1, IXR)
8:      IF(MODE.EQ.2) RETURN
9:      IF(MODE.EQ.3) READ(3, REC=1) IXL, IXR, IYT, IYB
10:      WRITE(*, '( * IXL, IXR, IYT, IYB= * ', 414\)' )IXL, IXR, IYT, IYB
11:      READ(*, *) IXL, IXR, IYT, IYB
12:      IX=IXR-IXL+1

```

画直线、圆等标准子程序中的画点子程序FPOINT后, 就可在内存数组和外存文件中画直线, 圆等。

4 图形数据处理子程序

子程序PLTPRN处理图形数据, 具有以下功能: MODE=1, 生成图形数据后立即打印; MODE=2, 图形数据存入文件; MODE=3, 从文件中读入图形数据打印图形。程序清单如下:


```

13,      N2=IX/256
14,      N1=MOD(IX, 256)
15,      OPEN(1, FILE=' PRN' , STATUS=' OLD' , FORM=' BINARY' , AC-
        CESS=' SEQUENTIAL' )
16,      WRITE(1) CHAR(27), CHAR(51), CHAR(24)
17,      IF(MODEIP.GT.30) THEN
18,          DO 200 J=IYT, IYB, 24
19,              L=(J+7)/8
20,              READ(3, REC=L+1) (FPD(I, 1), I=1, IXR)
21,              READ(3, REC=L+2) (FPD(I, 2), I=1, IXR)
22,              READ(3, REC=L+3) (FPD(I, 3), I=1, IXR)
23,              WRITE(1) CHAR(27), CHAR(42), CHAR(MODEIP), CHAR(N1), CHAR
        (N2), 1
        (FPD(I, 1), FPD(I, 2), FPD(I, 3), I=IXL, IXR), CHAR(13), CHAR
        (10)
24, 200    CONTINUE
25,      ELSE
26,          DO 300 J=IYT, IYB, 8
27,              L=(J+7)/8
28,              READ(3, REC=L+1) (FPD(I, 1), I=1, IXR)
29,              WRITE(1) CHAR(27), CHAR(42), CHAR(MODEIP), CHAR(N1) CHAR
        (N2)1 (FPD(I, 1), I=IXL, IXR), CHAR(13), CHAR(10)
30, 300    CONTINUE
31,      ENDIF
32,      CLOSE(1)
33,      RETURN
34,      END

```

第5、9行分别在图形文件第1个纪录写、读边界参数，第6、7行将最后当前图块存盘、第10、11行输入需打印的图形范围。第16行选择打印机走纸，使得图形连贯但不重叠。第18~24为24针图形工作状态打印图形，第26~30行为8针图形工作状态打印图形。前者分辨率比后者高，但速度比后者慢。第23、29行尾表示回车、换行。

5 结束语

本文方法改进了屏幕拷贝图形的不足之处，具有以下优点：

- a. 若无高分辨率图形屏幕但有图形打印机，仍能打印出光滑无锯齿现象的图形；
- b. 速度比屏幕拷贝快得多；
- c. 图形不失真；
- d. 图形大小可任意调整，可打印局部图形和超大图形；
- e. 可为FORTRAN 计算程序增加图形处理功能。

参考文献

李卫东. 屏幕拷贝的几种高级用法. 电脑与微电子技术, 1989.3

一种图象数据压缩存贮技术 及其实现

大连理工大学计算机系 孟宪福

(辽宁省大连市 邮编300072)

摘要 本文描述了一种二值图象数据压缩存贮技术,即二维图象数据压缩技术,同时给出了这种压缩技术的软件实现过程。

1 概述

在信息处理领域中,图象处理不同于规则图形处理,规则图形可通过相应的参数进行描述,而图象很难通过参数来进行描述。在一般的计算机图象处理系统中,图象是通过光学扫描器读入的,对二值图象处理来讲,其读入的是图象的点阵信息(即0、1信息),要处理一个屏幕大小(解像度为 640×400)的图象,大约在计算机中需要存贮30K字节左右的图象点阵信息,如果要处理更大的图象或者同时处理多个图象,则将需要更大量的存贮空间。虽然在图象处理过程中,我们可以采用内、外存数据交换的技术来处理大量的图象数据,但这种做法将导致大大降低图象处理速度。因此,为了能在微机中对图象进行有效地、快速地处理,就必须设法对图象数据进行压缩处理,以减少需要存储的数据量。

目前,对图象数据所采用的压缩技术一般可分为两类:一类是所谓“变形”的压缩技术,使用这种技术压缩的数据,经恢复算法恢复后,与原始数据有所差异,对于图象来讲,就是“变形”;另一类是所谓“不变形”的压缩技术,使用这种技术压缩的数据,经恢复算法恢复后,与原始数据完全一样,这对于图象来讲,就是“不变形”。由于受一般光学扫描器解像度的限制,通过光学扫描器读入的图象数据本来就与源图象有

所差异,如果这时再采用“变形”的图象数据压缩技术,则恢复后的结果图象将会发生更大变形,这也就失去了图象处理的意义。鉴于上述原因,我们设计并通过软件实现了一种“不变形”的图象数据压缩技术,即“二维图象数据压缩技术”,并把这种技术成功地应用于我们在NEC PC-9801机上开发的图象、图形、文字混合处理系统中。由于“二维图象数据压缩技术”是在“一维图象数据压缩技术”的基础上实现的,所以,我们先来介绍一下“一维图象数据压缩技术”。

2 一维图象数据压缩技术

一般来讲,图象数据压缩与汉字字模压缩是有区别的,其区别主要有两点:一是汉字字模数据是定长的,也就是说,对于不同的汉字,其字模数据的个数是相等的(当然是对同一点阵的汉字字模而言的),而图象数据是不定长的,图象的大小不同,则其点阵数据个数也不同,有时甚至差异很大;二是,一般来讲,汉字字模是在“小范围”内黑白相间的,也就是说,由于汉字字模数据本身并不很多,同时由于汉字字型本身的特点,使得汉字字模本身不会出现局部“一片黑”或“一片白”的情况,而对于图象数据来讲,可认为是在“大范围”内黑白相间的,由于图象本身的特点,使得多数图象会出现某个局部“一片黑”(即全“1”)和

“一片白”（即全“0”）的情况。

根据二值图象数据的上述特点，在进行图象数据的压缩处理过程中，只要设法减少图象中空白区域中数据的存储个数（也就是减少“0”的存储个数），而不改变非空白区域中（即含“1”区域）的数据存储个数，就可以实现图象数据的压缩，由于图象数据中含“1”的数据个数没有任何变化，所以这种方法为“不变形”的图象数据压缩技术。

下面是一维图象数据的压缩过程。

把一行图象数据每8位（一个字节）进行一次分割，当8位全是“0”时，就把所设置的压缩标志的相应位置“0”后，跳过此8位数字（即不把此8位数字传送给结果数据区，而仅保留一压缩标志位）；当8位数字不全为“0”时，把压缩标志的相应位置“1”后，把此8位数字传送给结果数据区。这样，一个字节的压缩标志可以代表8个字节的图象数据。在一个字节的压缩标志的各位都按同样的方式设置完后，再把一字节的压缩标志传送到结果数据区中它所代表的压缩数据的前面，对于一行中的所有图象数据都按同样的方式进行压缩，同时，对要处理的所有行的图象数据都采用相同的压缩技术进行逐行压缩，这样得到的压缩结果称为一级压缩结果，在一级压缩结果的基础上，再利用同样的压缩技术进行二级和三级压缩，对于一般的图象处理系统来讲，三级压缩结果可以作为最后的压缩结果而存储起来，以便以后进一步处理。由于这种压缩技术是以一行图象数据为压缩对象的，而没有考虑每行图象数据之间的联系，所以，我们把这种压缩技术称为一维图象数据压缩技术。图1给出了一维图象数据压缩的详细流程图。

由上述压缩过程可知，图象数据的压缩结果为压缩标志与数据的混合体，因此，在对压缩数据进行恢复时，可按顺序对相应的压缩标志进行处理，也就是当压缩标志为“1”时，把压缩数据区中的相应数据，传送给已恢复的结果数据区中；当压缩标志为

“0”时，把一字节的“0”数据传送给已恢复的结果数据区中，这样，逐级进行恢复。由此可见，对于使用这种技术压缩的数据，是很容易实现其恢复算法的。

从一维图象数据的压缩过程可以看出，在极端情况下，当需要处理的图象为“一片黑”时（即图象数据为全“1”的情况），则其压缩的结果数据个数将超过原始图象的数据个数（增加了压缩标志字节数），因此，如何有效地解决与此类似的问题，将是二维图象数据压缩技术要考虑的主要问题。

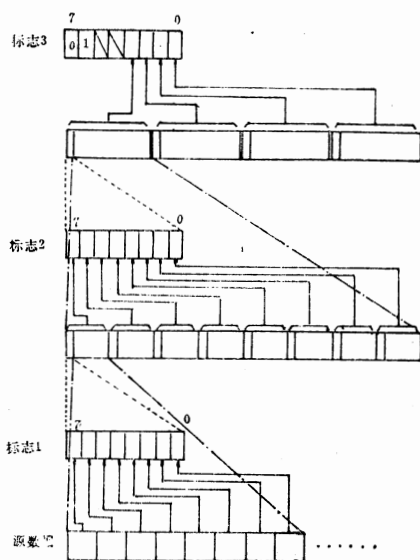


图1 一维图象数据压缩过程

3 二维图象数据压缩技术

从一维图象数据压缩技术可以看出，图象中的空白区域（也就是“0”）越多，则图象数据的压缩率越高。因此，要想提高图象数据的压缩率，其关键是要设法使得图象中的空白区域增多，当然，同时还要考虑相应的恢复算法。如前所述，图象可以被看作为“大范围”内黑白相间的，也就是说，其黑色区域或白色区域一般来讲是成“片”连续

的。我们知道,“异或”逻辑操作有下述性质,即:

- a. $x \text{ XOR } 1 = \text{NOT}(x)$
- b. $x \text{ XOR } 0 = x$
- c. $(x \text{ XOR } y) \text{ XOR } y = x$

由上述性质a和b可知,在图象数据中,当相邻两行的对应位相等时(实际上,即使有少量不等也无妨),将两行相邻的图象数据的对应位进行“异或”操作后,将大大增加图象数据中“0”的个数,同时,上述性质c又给出了经过“异或”操作后,数据的恢复算法。这样,我们就可以在进行一维图象数据压缩之前,先将邻近行之间的图象数据的对应位进行“异或”操作,见图2,然后,再把逐行进行“异或”操作之后的结果数据作为一维图象数据压缩过程的输入数据进行压缩处理。这样,由于增加了要压缩的图象数据中“0”字节的个数,所以将大大提高压缩率。

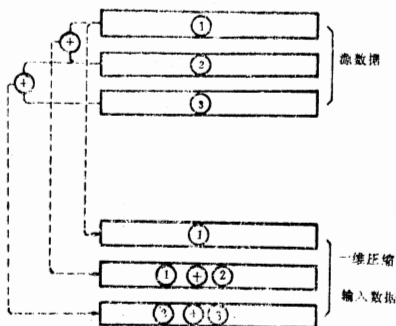


图2 二维图象数据压缩过程

由上述“异或”操作的性质C可以看出,利用这种技术对被压缩的数据进行恢复时,被恢复成的结果数据,再经过相应行之间的对应位的“异或”操作后,即可得到原始的图象数据。由于这种压缩技术不仅考虑了需要压缩的单行图象数据,而且也考虑了图象数据中行与行之间的联系,所以,我们把这种图象数据压缩技术称为“二维图象数据压缩技术”。

下面,我们给出二维图象数据压缩技术的C语言实现过程Comp_data(),每压缩一行图象数据调用一次此过程。该过程虽然

是图象数据的压缩过程,但根据相同的设计思想,就可以实现对已压缩的图象数据的恢复过程。下面对该过程中使用的几个主要变量说明一下。

length: 存放要压缩的每一行图象数据的长度(字节数)。

buff_3: 存放一行未压缩的原始图象数据。

buff_2: 用于临时保留上一行的图象数据,以便进行行之间的“异或”操作。

buff_1: 输出变量,用于存放已压缩的一行图象数据。

除了上述几个变量之外,其它变量的意义通过程序是不难明白的,这里就不再赘述了。

```
char buff_1[294],buff_2[80],buff_3[128];
int length=80,buff1_cnt,buff3_cnt;
void comp_data( ) /*compress the
                    image's data*/
```

```
{
    int i,j,tem3_cnt,temcnt;
    static char t_buff[80];
    unsigned k;
    unsigned char d_flag,s_flag,c_flag;
    buff1_cnt=buff3_cnt=0;
    tement=buff1_cnt++;
    s_flag=1;
    if(length==64)
        s_flag=2;
    c_flag=0;
    for(i=0;i<length;i++)
    {
        t_buff[i]=buff_3[i];
        buff_3[i] =buff_2[i];
        buff_2[i]=t_buff[i];
    }
    k=0xc8;
    for(j=0;j<s_flag;j++)
    {
        d_flag=0;
        tem3_cnt=buff3_cnt;
        for(i=0;i<64;i++)
        {
```

```

        d_flag |= buff_3[buf3_cnt];
        buf3_cnt++;
    }
    if(d_flag==0)
        k=k>>1;
    else
    {
        buf3_cnt=tem3_cnt;
        c_flag |= k;
        k=k>>1;
        comp_data1( );
    }
}
buff_1[temcnt]=c_flag;
}
void comp_data1( )
{
    unsigned char c_flag,d_flag;
    unsigned k;
    int i,j,temcnt,tem3_cnt;
    k=0x80;
    c_flag=0;
    temcnt=buf1_cnt++;
    for(i=0;i<8;i++)
    {
        d_flag=0;
        tem3_cnt=buf3_cnt;
        for(j=0;j<8;j++)
            d_flag |= buff_3[buf3_cnt++];
        if(d_flag==0)
            k=k>>1;
        else
        {
            buf3_cnt=tem3_cnt;
            c_flag |= k;
            k=k>>1;
            comp_data2( )
        }
    }
}

```

```

        buff_1[temcnt]=c_flag;
    }
    void comp_data2( )
    {
        unsigned char c_flag;
        unsigned k;
        int i,temcnt,tem3_cnt;
        temcnt=buf1_cnt++;
        k=0x80;
        c_flag=0;
        for(i=0;i<8;i++)
        {
            tem3_cnt=buf3_cnt;
            if(buff_3[buf3_cnt++]==0)
                k=k>>1;
            else
            {
                buf3_cnt=tem3_cnt;
                c_flag |= k;
                k=k>>1;
                buff_1[buf1_cnt++]=buff_3
                [buf3_cnt++];
            }
        }
        buff_1[temcnt]=c_flag;
    }
}

```

4 结束语

上述二维图象数据压缩技术被成功地应用于我们在 NEC PC—9801 微机上开发的图象、图形、文字混合处理系统中。实践证明，这种压缩技术对图象数据压缩率高，且算法简单。在上面介绍图象数据压缩技术过程中，我们仅仅考虑了图象边界为矩形的情况（即每一行的图象数据长度相等），实际上，只要我们为每一行图象数据设置相应的坐标信息和长度信息，是不难把上述图象数据压缩技术应用于图象边界不为矩形的情况的。

提高IBM PC微机输出效率的方法

航空航天部第六〇三所 支超有

(西安72信箱309分箱 邮码 710089)

摘要 本文介绍了在IBM PC系列微机上将用户的输出预先写入磁盘,再由键盘热键启动驻留于内存的中断服务程序,实时分时地读取磁盘数据并向输出端口发送。讨论了截取并完善现有中断的内存驻留程序的设计及其应注意的问题,由于内存驻留程序与操作系统成前后台的工作方式,从而提高了用户在高档微机上输出数据的效率和计算机的利用率,并方便用户的使用。

1 问题的提出

试验室环境或工程计算中,使用IBM 286或386微机的用户,完成大批量的数据处理工作后,往往要将数据结果打印输出或格式化数据,以绘图仪绘出图形曲线。由于微机性能的提高,与打印机、绘图仪相比,使得用户整个工作的“瓶颈”问题转移到输出。对于连接有多台外设的微机,如何提高其输出效率成为用户首先关心的问题。那么,能不能在微机DOS磁盘操作系统之上,开发微机并行输出功能,从而使微机具有大型或中型机那样的假脱机输出功能。

在用户程序开发中,通过切换输出端口打印、绘图输出数据,不失为一种方法,但它要求用户熟悉输出操作技术,实现过程比较复杂,不具有通用性。

由于目前的微机都配置有大容量的硬盘,切实可行的方法就是在磁盘上开辟一输出缓冲区,并对应于某一输出设备。在用户程序中将输出改向写入磁盘中对应于该设备的缓冲区,待用户输出操作完成后,启动一中断服务程序,由该例程将输出缓冲区数据发送给输出设备——绘图仪或打印机。

考虑到一般微机上都配置有打印机并行接口及RS-232C串行接口,这里我们将串

口的输出改向磁盘,其缓冲区命名为一数据文件。而中断服务程序则截取实时时钟中断例程,增加对通讯口状态检查功能,并完成读取磁盘数据及向串口发送工作。这样的中断例程由系统硬件计数器的情况所调用,用户完成低速的输入或输出(如编辑程序、打印输出)时,启动中断服务例程,就可将用户预先写入磁盘的输出从串口输出,增加了系统的并行处理能力。

2 系统中断的扩充 与内存驻留

中断服务程序的设计是截取了系统原有中断,并对其扩充完善以满足用户新的要求,在系统启动时驻留于内存并结束退出(称TSR)。

扩充并完善已有的中断时,首先要保存原有的中断向量,这是由调用DOS系统功能软中断INT 21H实现的。其入口寄存器值为:

AH=35H, 功能号;

AL=中断号;

返回时:

ES:BX=中断向量的段:偏移地址。

重新设置中断向量地址,使其指向新的中断服务程序时,入口寄存器的设置为:

AH=25H, 功能号;

AL=新的中断号;

DS: DX=指向新的中断服务程序。

设计好的中断服务程序驻留于内存时, 必须考虑兼容性。即它必须与其它 TSR 程序、操作系统、中断了的前台应用程序、BIOS 磁盘活动, 以及与中断处理程序“和平共处”。

由于新的 TSR 中断例程是截取了原有中断, 在新的例程一开始应调用原有的中断, 这样不至于破坏 DOS 系统原有的功能, 方法是使用下面两条语句:

PUSHF

CALL OLD-VECTOR

考虑到新设计的 TSR 例程必须与 DOS 操作系统兼容, 从而使内存驻留程序可以自由地使用 DOS 中断服务程序, 这就要求 DOS 正在执行期间, 不能激活 TSR, 一旦得知 DOS 已不执行可激活 TSR。如何才能确定 DOS 当前是否正在执行? 可使用中断 INT 21H 的功能 34H, 此中断返回一个指向由 DOS 维护其标志的指针, 用以表明 DOS 代码当前是否在活动, 标志 INDOS 实际上是一个计数器, 表示已进入 DOS 代码的递归次数。如果 INDOS 为 0, 表示 DOS 无活动; 如果它大于 0, 表示 DOS 正在活动且其代码已进入了一次以上, 功能 34H 调用规则如下:

进入, AH=34H

返回, ES=INDOS 段地址;

BX=INDOS 偏移地址

中断例程执行前, 必须保存原来的机器状态, 包括有各寄存器的内容, 视频显示及 DOS 参数; 结束退出前恢复所保存的内容。磁盘传送区 DTA 是 DOS 用于为文件控制块功能传送文件的内存缓冲区, DOS 维护当前 DTA 段: 偏移地址的一条内部记录。TSR 使用 DTA 操作时, 当前 DTA 被临时地改变, 我们可通过 DOS 功能调用 2FH 和 1AH 来获得原有 DTA 及设置新的 DTA。

内存驻留程序中断现有例程的执行应满

足一系列条件, 激活 TSR 的时机应在 BIOS 磁盘服务不进行时。所有 BIOS 磁盘服务均是通过中断 INT 13H 访问的, 为了防止在中断 INT 13H 进行过程中激活 TSR, 简单的方法是捕获此中断并当 BIOS 忙时, 设置标志 IN-BIOS 为“1”。

另外, 由于 TSR 是由计算机硬件情况所调用, 必须考虑重新进入问题, 在 TSR 程序中设置一 BUSY 标志, 只要代码当前在活动, 该标志置位。激活 TSR 的中断处理程序首先必须检查此标志, 如果置位, 则程序立即返回, 有效地防止了递归地激活; 如果此标志未置位, 则继续测试以确定是否激活 TSR。

3 微机假脱机输出的实现

微机假脱机输出是由一系列中断服务程序完成的, 这些中断例程驻留于内存中, 当用户按下指定的组合键 HOTKEY (称为“热键”), 激活并启动 TSR 工作, 读取用户预先写入磁盘的数据文件并向串口发送数据。该 TSR 与操作系统成前后台方式工作, 该功能类似于 DOS 系统屏幕硬拷贝功能。

驻留内存的中断例程完成初始化工作包括对 RS-232C 通讯端口通讯格式参数的设置, 其端口各寄存器设置为:

3FBH=80H;

3F8H=0CH;

3F9H=00H;

3FBH=07H。

对应的通讯格式为: 速率 9600 波特; 7 个数据位; 1 个停止位。另外从程序段前缀 PSP 处得到数据文件名, 打开数据文件, 准备以后对其读写。第三, 检测目前工作的 DOS 版本号, 因为 DOS 2.0 以前的版本不支持一些必要的功能。再完成对系统 09H、13H、1CH、28H 原中断向量地址的保存, 及设置分别对应于新的中断例程的向量地址。

由于 09H 号中断对应于硬件键盘中断,

28H号对应于DOS空闲中断,新的中断例程增加了对预先定义的HOTKEY的检测功能,它定义为同时按下ALT及LEFT-SHIFT两键,对应的键盘编码为000AH。键值是这样获得的,读地址0000H:0417H处的一个字,查看BIOS SHIFT键标志,以确定HOTKEY组合键当前是否被按下。如果HOTKEY被触发,到设置标志PER-ACT为“1”,允许读文件并输出。

ICH号计时时钟中断以每秒18.2次的速率被调用,能实时分时地与系统并行工作。读磁盘数据并向端口发送的例程截取了该中断。它进入前,要执行一系列的测试以确定是否被激活,如果测试中任一项失效,则立即发出IRET指令以返回。实行测试有:

测试PER-ACT标志,查看用户是否按了热键HOTKEY;

测试BUSY标志,确定该TSR是否正在活动;

如果IN-BIOS标志等于1,则BIOS磁盘服务中断13H当前正在活动,返回。

如果通过双字指针INDOS-PTR访问的INDOS标志大于0,则DOS不应被中断,该处理程序简单地返回。

上述四项测试通过后,执行中断服务,它首先检查通讯口是否空闲,如果空闲,则从磁盘读取字符并向通讯端口发送,直到通讯接收缓冲区满退出。在读盘过程中遇到文件结束则结束,并设置标志PER-ACT为“0”。根据以上方法,我们已编制了读出磁盘数据文件,向连接绘图仪的串口发送数据,完成绘图输出的程序(清单略)。

参考文献

- 1 夏东涛编译.MS-DOS高水平程序设计.电子工业出版社
- 2 [美]Ray.Duncan著,贺志强,李旨译.DOS磁盘操作系统高级程序员指南.1987.1

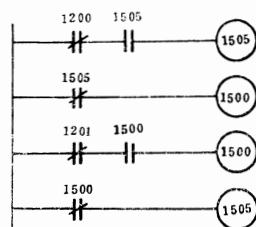
编者按

本刊91年第四期刊登了常斗南同志的文章“PC在电梯控制系统中的应用”后,读者来信对此文提出了一些不同看法。对于读者这种严谨治学、对杂志负责的一切做法我们深表欢迎,并及时同作者联系,作者对原文进行了一些修改,现发表于下。

更正

第四期发表的PC在电梯控制系统中的应用一文中的图3所示梯形图,有遗漏应改正如下:第16个继电器线圈1009的下面应补充的记忆内容如图所示。

与输出继电器Q500相连接的常闭接点0505应改为1505,常开接点1003与常闭接点1009串联的这一行中应增加常闭接点1006与之串联。



与输出继电器Q505相连接的常闭接点0500应改为1500,常开接点1000与常闭接点1006串联的这一行中应增加常闭接点1009与之串联。

倒数第三行的常开接点0500应改为常闭接点,倒数第二行的常开接点0505应改为常闭接点。两者应该并联,即一端联母线,另一端两者应短接。其余联接正确。

常斗南

DDJ动态、多维数据检索算法设计

天津海关计算中心 马长生

(邮 码 300012)

摘要 几年来,我们对各种类型的数据检索进行了剖析,找出了规律,研制了动态多维检索算法,简称DDJ算法。该算法是在VAX计算机上,用PASCAL语言开发的,灵活地应用了指针变量,集合变量和可变长字符串变量。把复杂的检索过程,简化成几个小的算法模块,用一个共用程序可满足千变万化的检索要求,从而减轻了编程劳动,加快了检索速度,节省了存储空间,提高了系统效率。

1 问题提出

目前,计算机的发展速度越来越快,应用范围也越来越广。然而,数据检索仍然是计算机应用的一个重要方面。特别是对计算机中已存储了大量数据资源的单位来说,经常要为上级机关和社会提供各种数据资料,进行各种各样的检索。如果不同的检索需要不同的程序,那么要给软件的设计和维护带来很多麻烦。如何加强对数据的综合分析能力,如何提高计算机的查找速度,以及如何用一个程序满足各种各样的检索要求,是计算机软件设计人员的一项重要课题。

所谓数据检索,就是要在文件或数据库中找出指定的记录。计算机中的数据是存储在数据库或文件中的,文件或数据库是大量记录的组合,而记录是由多个数据项组成的,少则几项,多则十几项,甚至于百余项。最简单的检索是给出一个数据项中的一个指定值,复杂的检索不仅给出多个相关数据项,而且每个数据项中又可给出多个指定值。再加上每次检索的数据项和给定值又是变化的,这种变化的范围又很大,因此,各种检索组合的种类很多。比如:海关贸易统计中有十八个数据项。数据项的各种组合数可以用公式

$$\sum_{n=1}^{18} C_{18}^n \text{ 来计算。其中 } C_{18}^n = \frac{P}{N!},$$

$$P_{18}^n = 18 \times (18-1) \times \dots \times (18-N+1);$$

$$N! = N \times (N-1) \times \dots \times 1.$$

$$\text{那么 } \sum_{n=1}^{18} C_{18}^n = C_{18}^1 + C_{18}^2 + \dots + C_{18}^{18} \\ = 262143.$$

一个数据项下面又可以给多个可能值,比如贸易国别216个,其各种组合数为:

$$\sum_{n=1}^{216} C_{216}^n = 1.05 \times 10^{65}$$

商品代码近4000个,其各种组合数为:

$$\sum_{n=1}^{4000} C_{4000}^n = 1.318 \times 10^{1204}$$

如果把十八种数据项的各种组合都加起来,那么,很难想象这个数字有多大,这么大的数是不能用千、万、亿来表示的,而只能用天文数字来描述了,面对千变万化的数据检索。如何设计一种算法,能覆盖上述各种检索组合,如何设计一个固定的程序,就可以满足可多可少,可大可小,可长可短的各种检索要求,当用户提出一个新的检索条件时,不用重新编写程序,也不用重新编译,链接,就可实现动态,多维数据检索。这就是DDJ算法需要解决的问题。

2 摸清规律

为了解决上述问题,必须对数据检索进行详细的分析,对一个文件检索的逻辑关系可归纳成九种类型,即:

- a. 一个数据项中一个检索值。
- b. 一个数据项中多个检索值,且之间为“或”关系。
- c. 一个数据项中多个检索值,且之间为“与”关系。
- d. 多个数据项之间为“或”关系,每个数据项中一个检索值。
- e. 多个数据项之间为“与”关系,每个数据项中一个检索值。
- f. 多个数据项之间为“与”关系,每个数据项中多个检索值,且之间为“或”关系。
- g. 多个数据项之间为“与”关系,每个数据项中多个检索值,且之间为“与”关系。
- h. 多个数据项之间为“或”关系,每个数据项中多个检索值,且之间为“或”关系。
- i. 多个数据项之间为“或”关系,每个数据项中多个检索值,且之间为“与”关系。

通过对上述检索类型的分析,我们认为只有1, 2, 5, 6四种类型的检索是有用的,被称为相关检索,其它类型被称为无关检索。再对四种检索进行分析,可以把1, 2, 5作为第6种类型的特殊情况。从而找出这样一个规律,即凡是有用的检索,一般具有两个特点:第一,给定需要检索的数据项与数据项之间,必须是“与”的逻辑关系;第二,每个数据项中各给定值之间,必须是“或”的逻辑关系。比如:在海关贸易统计数据中,要查找贸易国别是“日本”、“香港”和“美国”,而且商品名称是“三十二位以上的计算机”、“彩色电视机”和“八吨以上卡车”的全部记录。这里的贸易国别和商品名称为给定的数据项,它们之间的关系是“与”的关系,而每种数据项中的给定值之间则是“或”的关系。也就是要查找那

些从“日本”或“香港”或“美国”进出口的,同时(“与”)商品名称必须是“三十二位以上计算机”或“彩色电视机”或“八吨以上卡车”的全部记录。

通过对数据检索的类型分析,我们可以找出检索的特点,总结出检索的规律。把对一个文件的检索归纳成一个模式,简化了算法设计,为编写一个程序满足各种各样的检索要求打下了基础。

3 算法设计

根据上述分析,在数据检索过程中,应该记住两种情况,一是要记住检索的数据项;二是要记住每个数据项中的各给定值。一般说来数据项的个数是比较少的,有限的。而每个数据项的给定值则是比较多的,无法确定的。因此,我们采用集合变量来表示需要检索的数据项,而用二叉树结构来表示每个数据项中的各种给定值。用集合变量来表示多维,用二叉树结构来实现动态。具体步骤如下:

3.1 定义给定项集合SO

这个集合变量中可以包括所有的数据项的编号或其中一部分。最简单的情况是含有一个数据项编号,也可以是两项、三项或多项。集合中有多少个元素,就是多少维的数据检索。在海关贸易统计中共有十八个数据项,因此在集合中最多包括十八个元素。开始时应设集合SO为空,这个变量是靠用户通过终端根据用户的选择输入而建立的,在SO中应包括用户需要检索的数据项。比如:前面提到的需要查找的数据项只有贸易国别和商品名称,假设贸易国别数据项的代码为“5”,而商品名称数据项的代码为“7”,那么给定集合变量SO应该被设置为 $SO = [5, 7]$,SO中元素的多少是由用户提出检索数据项的多少而决定的。

3.2 建立条件树

MCS-51单片机配接CE-515P彩色绘图 打印机和一种汉字打印的实现

重庆大学自动化系 戴强

(重庆市沙坪坝区重庆大学12舍26号 邮编630044)

摘要 本文给出了用CE-515P彩色绘图打印机作为微机测控系统输出设备的方法,并给出了一种在绘图打印机上实现汉字打印的方法。

1 概述

在使用的点阵式微型打印机中,适合于作为单片机输出设备的打印机,一般都是些价格不高的中低档微型打印机。它们在要求以曲线形式和表格形式记录数据时就显得十

分不便。而CE-515P彩色绘图打印机是一种内带微处理器的输出设备,其价格适中、体积小、功能强,是一种比较理想的输出设备。

2 CE-515P简介

在条件树中应包括所有的检索条件,程序设计中应考虑到各种可能性。因此,有多少个数据项就应该允许建多少棵条件树,树中的条件数据也是根据用户的要求从终端输入而得到的,在海关贸易统计中最多可建十八棵树。如果只检索贸易国别和商品名称两个数据项就建立两棵树,一棵是贸易国别树,一棵是商品名称树,其它树都为空。在贸易国别树里包括了用户提出的全部检索国别,在商品名称树中包括了用户提供的所有检索商品名称。树的结构一般为二叉树,以便提高查找速度。

3.3 读取一条记录和遍历二叉树

从数据库或文件中取出一条记录,首先判断哪些数据项的编号在给定集合SO中,然后再用相应的数据到相应的二叉树中去遍历。一个数据项命中以后,再处理下一个检索数据。所有检索条件都满足才算命中记录,否则取下一条记录,一直到把所有满足检索条件的全部记录都取出来为止。

3.4 对满足条件的记录进行处理

根据用户要求(这些要求也是从终端输入的),该合并的合并,该累加的累加,选

择输出数据等,最后显示或打印出用户要求的数据。

4 应用效果

现在,DDJ算法,经不断整理,不断完善,已基本成型,并在海关贸易统计、信息管理、法规管理、文档管理以及报关自动化系统的程序设计中得到了应用。实践证明,这种算法不仅查找速度快、使用灵活、思路简明,而且覆盖面大、适应性强、应用广泛。多次为有关部门提供了重要信息,发挥了较好的效益,大大提高了计算机的综合分析能力。比如:利用这种算法很方便地从12.96亿个数据中查出了天津口岸一九八八年进出口同类商品有“贵进贱出”的问题,引起了有关部门的重视。发挥了计算机在宏观监控中的作用,显示了计算机的优越性。今后,我们还将与其它综合性大型项目中推广应用,以使现有的计算机设备及计算机中存储的大量数据发挥更大的作用。

CE-515P是日本SHARP公司生产的一种彩色绘图打印机。可以通过并行口或串行口与系统相连。该机有四支彩笔可供选择,在程序的控制下可进行字符打印或进行绘图。

该机提供多种幅宽的纸供用户选择,字符大小和线型通过命令可任意设置,以方便用户编制不同的输出格式。

3 8031与CE-515P 的接口

8031与CE-515P的接口是很简单的,通过CE-515P提供的并行口与8031的P1口和P3口直接相连,其信号的连接如图1所示。

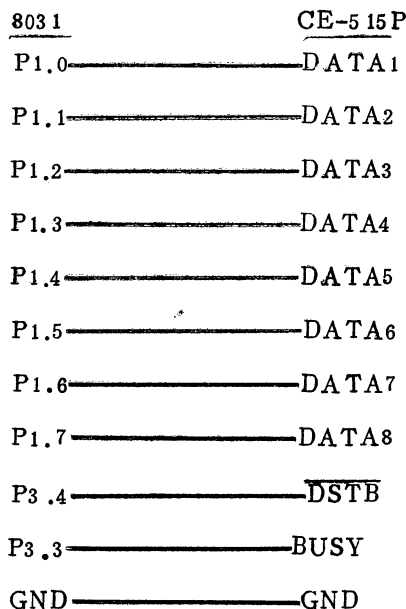


图1

P3.4为输出,接“选通”信号 $\overline{\text{DSTB}}$,当CE-515P接收到 $\overline{\text{DSTB}}$ 信号为低时,数据(DATA1~8)被接收。

P3.3为输入,接“忙”信号BUSY,当单片机接收到BUSY为高电平时,CE-515P不能接收数据。

CE-515P接口信号的时序如图2所示。

根据上面的连线及时序,可写出打印机的基本驱动程序。注意,这里8031的时钟频率为8MHZ。

DRIVE: JB P3.3, DRIVE

```
MOV P1, A
CLR P3.4
SETB P3.4
DRIVE: JNB P3.3, DRIVE
RET
```

CE-515P从单片机中接收打印命令以及待打印的字符信息,命令和字符全部用ASCII码传送给CE-515P。只要将命令或字符的ASCII码放入累加器A中,调用DRIVE子程序就可驱动打印机按命令所要求的方式工作。

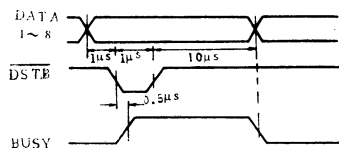


图2

4 汉字打印的实现

在CE-515P中,笔走一步距离为0.2mm,只要落笔走一步就可在纸上打出一个点,根据汉字点阵来控制笔的抬落笔运动就可描述出一个完整的汉字图形,从而实现汉字的打印。

汉字以24×24的点阵来描述,每个汉字点阵占24行,每行占3个字节,这样,每个汉字由72个字节的点阵数据来描述。

在表1中作为例子给出了“年、月、日”三个汉字的编码,对应汉字的点阵数据存放在以CROM为起始地址的程序存储器中。用户在具体的系统中可自行编制所需的汉字点阵数据库。应注意汉字的编码顺序应和点阵数据存放顺序相对应。

例如:“年”的编码为81H。

“月”的编码为82H。

“日”的编码为83H。

则三个汉字点阵数据的存贮顺序如图3。

图4是字符和汉字打印程序的流程,待打印的字符和汉字编码存放在以CROM为起

始地址的数据缓冲区中。

表1

高4位

	0	1	2	3	4	5	6	7	8	9	A	...	F
0			SP	0	@	P	c	p	~				
1			!	1	A	Q	a	q	年				
2			..	2	B	R	b	r	月				
3			#	3	C	S	c	s	日				
4			\$	4	D	T	d	t					
5			%	5	E	U	e	u					
6			&	6	F	V	f	v					
7			,	7	G	W	g	w					
8	BS		(8	H	X	h	x					
9)	9	I	Y	i	y					
ALF			*	:	J	Z	j	z					
BLU	ESC		+	:	K	[k	{					
C			,	<	L	¥	l	:					
DCR			-	=	M]	m	}					
ESO			.	>	N	^	n	_					
FSI			/	?	O	...	o	☒					

低4位

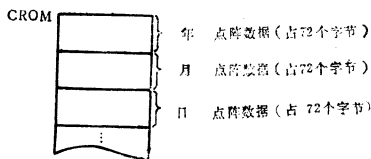


图3

在图4中, R7为汉字点阵行数(24), R6为点阵一行的字节数(3), R5为一个字节的位数(8)。

利用该程序我们可打印字符和汉字, 例如在CRAM的数据缓冲区中存放数据 31H, 39H, 39H, 31H, 81H, 33H, 82H, 31H, 38H, 83H, FFH。则调用该程序可打印出: 1991年3月18日

关于CE-515P的有关绘图命令, 可参阅其使用说明书, 在此不一一列举。

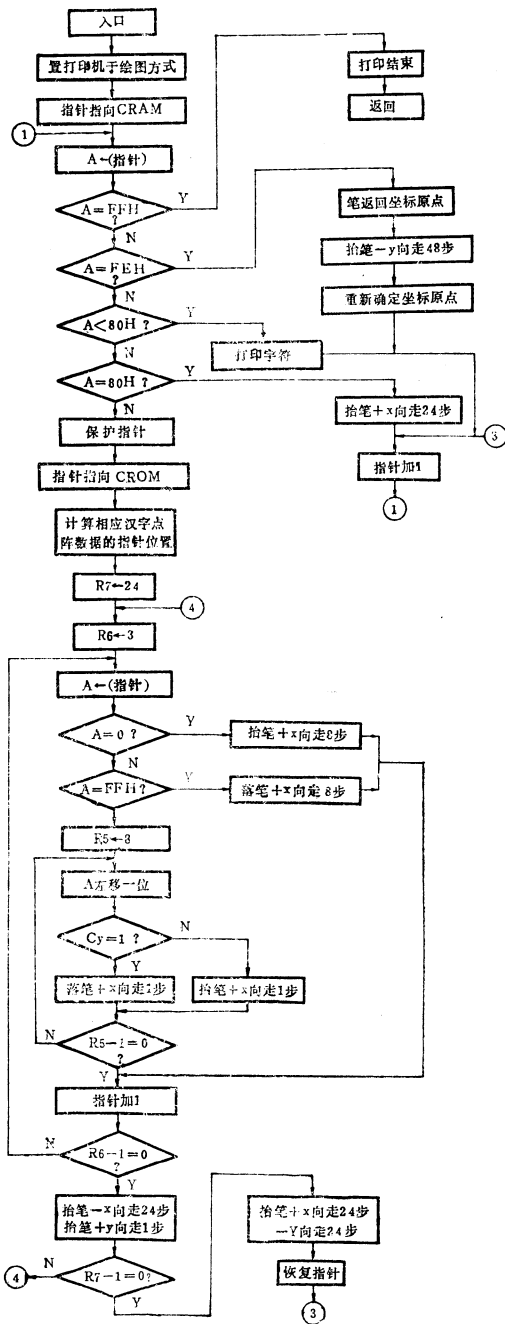


图4

5 结论

使用CE-515P彩色绘图打印机作测控系统的输出设备是可行的, 我们已成功地将该打印机用于“单片微机烘炉测控系统”中,

用MIRROR程序巧取硬盘分区表

苏州市物资信息中心 戈 海

(苏州市道前街170号 邮编 215002)

每台经过格式化后的硬盘都至少拥有一个分区,应用中也可根据需要使用多个分区,有关硬盘驱动器分区信息存贮在硬盘的一张特殊的分区表中。如果这种分区表信息丢失或破坏,DOS将无法“识别”你的硬盘,并且每当你试图访问这台硬盘驱动器时都将显示一条“Invalid drive Specification(无效的驱动器说明)”出错信息。

当出现这种情况时,用户可能会采取几种不同的选择:

a 硬修复:对硬盘低级格式化,重新分区。

b 软修复:用其它方法读取并修改硬盘分区表,使之恢复正常。

大多数用户都因为硬盘上存贮着大量有价值的信息而愿意采用第二种选择,但分区表信息存贮在硬盘0头0柱1扇区上,并且标志为隐藏扇区,用普通调试程序(DEBUG)无法观察到。因此,往往不知从何着手。

读取硬盘分区表的一种方法是先修改BOOT区(引导区)的第1C、1D两位,使其为0(即将硬盘隐藏扇区数置为0)再用普通的方法加以观察。

读取硬盘分区表的另一种方法是调用中断13。

上面两种方法都较复杂,还有一种可以

借助现成工具,无需了解DOS内部机制的简便有效的方法,那就是运行带有/PARTN参数的MIRROR程序。

众所周知,不带/PARTN参数的MIRROR程序可以存贮根目录(ROOT)、文件分配表(FAT)和自举记录(BOOT)的备份,这通常足以恢复你的数据除非你的分区表已被破坏。

使用带有/PARTN参数的MIRROR程序可以把硬盘分区表信息存贮到一个名为PARTNSAV.FIL文件中。每当你增加或者修改任何格式化信息的时候(使用FDISK.COM),或者改变逻辑驱动器的时候(例如,首先进入然后再删除D),需要重新运行MIRROR/PARTN以便存贮新的分区表信息。当完成这一操作后,将带有这一文件的软盘存放到安全的地方。

一旦分区表意外损坏,可以运行REBUILD/PARTN使你能够恢复分区表信息以便你可以再次通过运行REBUILD程序来恢复计算机的目录和文件分配表。

REBUILD程序可以检查存贮在PARTNSAV.FIL文件中的驱动器参数是否匹配实际的驱动器参数。两者必须完全匹配才行,否则REBUILD程序将拒绝恢复这些信息。

另外,REBUILD/PARTN/L将显示

所有输出数据以表格形式输出,表头用汉字提示。

对于一般测控系统中使用汉字数目不多,本文所给出的方法可提供120多个汉字,一般可以满足实际需要,用户还可以根据需要编制汉字点阵来满足特殊要求。

参考文献

- 1 宋宏远,杨天怡.单片微型计算机原理及应用.重庆大学出版社,1990年6月
- 2 张福炎.微型计算机IBM PC原理与应用(续篇).南京大学出版社,1990年4月
- 3 杨大全.TP801单板机配接ASG-1100微型绘图打印机.TP通讯,第十期

长城0520CH多功能板打印机

适配器维修一例

长庆油田宁夏测井站 王映辉

(宁夏灵武县马家滩 邮码751408)

1 故障现象

a. 当用〈CTRL〉+P或〈CTRL〉+Prt Sc键连接(选通)打印机后,用TYPE命令打印某一文本文件或用DIR命令列目录时,屏幕显示:

Write fault error Writing device PRN
Abort, Retry, Ignore?

且打印机在每行的开头(第一列)顺次打印W,r,i,...,P,R,N紧接在下一行打印出:

Abort, Retry, Ignore?

之后,打印机暂停。

b. 当用SHIFT+Prtsc键进行屏幕硬拷贝时,打印正常。

c. 用WORDSTAR进行文本打印时也能正常进行;但用dBASEⅢ或其它高级语言控制打印时则在屏幕上显示:

Device fault

且打印机不动作或将打印纸推进一行。

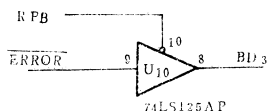
2 判断分析

2.1 利用DIAGSTAR(随机诊断程序)进行打印机适配器检查(插上相应的短接插头)时,无故障信息显示。

2.2 当用BASIC的OUT和INP命令对打印机的三个端口(378H, 379H, 37AH)进行读写时,发现状态寄存器(通过379H口)D₃位(即ERROR)始终为“0”,即一直处于出错状态。

2.3 根据打印机适配器原理,跟踪ERROR去向,由图1(注:与IBMPc/XT的U₁₀脚

码不同)可知:打印机出错时发出ERROR信号(低电平有效)到U₁₀的第9脚,当U₁₀的第10脚(三态门的控制端)为低电平时,通过U₁₀的第8脚输出到内部数据总线(即BD₃)。由于计算机其它方面及打印机自检正常,计算机与打印机之间的电缆线经检查也完好,所以初步判断U₁₀(74LS125AP)不正常。



↑
打印机适配器插座第15孔

图1

2.4 进一步用示波器检查,发现当U₁₀-10脚接地(低)时,无论U₁₀-9脚是高电平还是低电平,U₁₀-8始终输出为低电平,说明此处有故障。

2.5 将U₁₀-8脚与座板上的连线切断(悬空)时,则打印完全恢复正常。

通过以上各步,可完全确定U₁₀(74LS125AP)芯片已坏,更换之即可。

由以上分析看出,此种故障,由于屏幕硬拷贝正常;WORDSTAR可驱动打印;加之用DIAGSTAR(随机诊断程序)进行检查时无故障显示(利用短接插头),因而易使人误认为是软件问题。

本文目的仅在于尽量多的告诉大家,当发生此故障时可能出现的几种现象。由于笔者水平有限,难免有漏洞之处,欢迎批评指导。

出当前的驱动器分区表。分区表的长度是以兆字节单位表示的,且基于标准长度的扇区(512字节)。如果你硬盘上的扇区长度是其它数值,那么所显示出的分区长度可能是

不正确的,分区长度是以扇区个数乘以512而得到的。

为了把分区表信息输出到打印机上,可以增加一个可选的参数/P。