

微小型计算机



开发与应用

MICRO-MINICOMPUTER
DEVELOPMENT & APPLICATION

10015

微小型计算机开发与应用编辑部

微小型计算机开发与应用（公开发行）

编辑：《微小型计算机开发与应用》编辑部

出版：天津市电子计算机研究所

天津市计算机学会

地址：天津市河西区友谊路宾馆南道5号

邮政编码：300061

发行：天津市邮局

印刷：天津武清县长宏印刷厂

订购处：全国各地邮局

定价：0.95

邮局代号6—87

津工商广字0116号

国内统一刊号CN12—1122ISSN1001—8786

《微小型计算机开发与 应用》编辑委员会

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

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

1991年第5期目录 (总第55期)

应用实例

- 病房多用单片机定时器
.....吴振声 王兆月 邢立军 (1)
- 水电站微机监控网络管理
...游大海 沈宗树 万永明 管家宝 (3)
- 微机在自动门百万次寿命试验中的应用
.....王瑞刚 王丽清 (8)
- 单片机在无线遥测标仪中的应用
.....刘子义 (11)
- 微机磁场成型控制
.....马滋宇 郭友民 (14)

计算机软件

- Prime计算机自动记账系统(PCAAS)的设计与实现.....杨振平 张维韬 (16)
- 如何在Turbo Prolog 系统中调用汇编语言定义的数据库.....梁继鸿 胡 涛 (19)
- 用微机辅助微波理论课教学
.....刘 健 李在清 黄记辉 (25)
- “所见即所得”.....朱桂林 (28)
- IBM—5550机与长城、IBM—PC、286、386机之间连网通信中的实时汉码转换
.....谢晓尧 江国章 (32)

接口技术

- IBM-PC/XT与MCS—51 并行长线通讯中的硬件接口和软件设计.....郑相周 (34)
- Turbo prolog和Turbo c的接口技术
.....张川民 (39)

经验点滴

- 彩色显示适配器故障的分析与检测
.....李笑梅 陈致明 (42)

CONTENTS

APPLICATION EXAMPLE

- Multi-purpose Single Chip Computer Timer Used in Ward
.....Wu Zhensheng Wang Zhaoyue Xing Lijun (1)
- Monitoring and Controlling Network Management Using Microcomputer for
Hydraulic Power Station
.....You Dahai Shen Zongshu Wan Yongming Guan Jiabao (3)
- Application of Microcomputer for Million Times Life Test of Auto-Door
.....Wang Ruigang Wang Licong (8)
- An Application of Single Chip Computer to the Telemetry Surveyor's
Beacon.....Liu Ziyi (11)
- Controlling the Magnetic Field Forming Using Microcomputer
.....Ma Ziyu Guo Youn in (14)

SOFTWARE

- Design and Implement of Prime Computer Automatic Accounting System
(PCAAS)Yang Zhenping Zhang Weitao (16)
- How to Call the Database Defined by Using Assembly Language in Turbo
Prolog System.....Liang Jihong Hu Tao (19)
- Computer Aided Instruction in Microwave Theory Courses
.....Liu Jian Li Zaiqing Huang Jihui (25)
- "What You See Is What You Get" Zhu Guilin (28)
- Real Time Chinese Code Transformation in Networking Communications
among IBM-5550, the Greatwall Computer IBM-PC 286, 386
.....Xie Xiaoyao Jiang Guozhang (32)

INTERFACE

- Design of Hardware Interface and Software for IBM-PC/XT and MCS-51
Parallel Long Line Communications.....Zheng Xiangzhou (34)
- Interface between Turbo Prolog and Turbo C.....Zhang Chuanmin (39)

EXPERIENCE

- Analysing and Checking the Fault of Color Display Adapter
.....Li Xiaomei Chen Zhiming (42)

病房多用单片机定时器

天津医学院生物医学工程系 吴振声 王兆月 邢立军

摘要 本文介绍的是一个用 MCS—51 单片机构成的应用系统, 它具有一个走时精度较高的实时钟、同时可供 8 个床位各自独立使用的定时器、声音报警器、电器开关驱动器以及测量体温的显示功能等。可以通过按键操作和用数字显示, 文中对系统的硬件结构原理和软件的设计作了具体说明。

一、概 述

在医院的理疗、针灸等部门或病房当中，经常需要定时器，而且是多床位同时需要不同的定时时间，操作要简便，可见度要高，报警信号要明显。目前多数医院仍沿用插棒式的机械定时钟，虽能定时，但功能单一，铃响以后是谁申请的定时需要自己记忆和辨认。至于价格，也未必便宜。所以，自动化程度高，操作简便，使用灵活，价格便宜的定时器是医院中所需要的通用设备。

基于上述情况,我们研制了单片机多路定时器。本定时器的核心是目前应用极为广泛的51系列单片机,配置了外部设备,构成了一个可编程的计时定时应用系统,具有体积小,可靠性高、功能强等特点,不仅能满足医用要求而且还有很多功能可供开发,有着广泛的应用领域。

二、技术性能

1、可作实时钟 用六位数字显示当前的时分秒，24小时累计误差 ≤ 0.2 秒，可以在任意时刻对表。

2. 定时功能 (1) 同时可供8个床位单独申请不同的定时时间, 通过按键置入床号和定时时间, 范围是1—99分钟, 定时误差 ± 1 秒。(2) 定时时间到有音响报警, 相应的床号闪烁10秒钟。(3) 各床定时时间可以查询、可以修改或取消。

3. 驱动功能

在定时时间内可以给出驱

动信号,使相应的医疗电器处于通电状态,定时时间到自动切断电源,驱动开关的触点可以通过5A交流电流。

4. 体温的测量显示功能 将体温经传感器变成0—5V信号转换为相应的数字显示出来。

上述各功能都不影响实时钟对当前时间的显示。本定时器还有其它功能供开发，如脉搏的测量显示，可编程自动开关等等，不仅可以用来更新医院设备，同时还可应用于其它工矿企业、事业单位及家庭各个领域。

三、硬件的构成和原理

硬件的构成如图1。大体分四个部分：

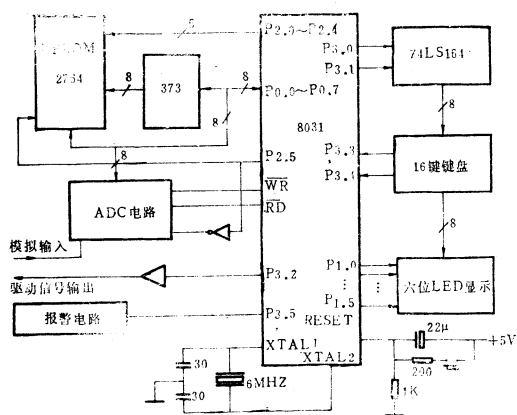


图1

1. 单片机8031 由地址锁存器74LS373和程序存储器2764 (EPROM) 等芯片构成的最小系统,这是控制的核心。8031的P₀口既是低8位地址线,又是数据线,采用分时工作,地址信息通过ALE信号锁存在373的输出端。P₂口用作高位地址线。地址选通以

后,通过PSEN信号访问程序存储器,执行固化在2764中的管理和应用程序。

2. 六位数码显示和16键键盘 使用者可以方便地观察系统的工作状态或给它输入命令。单片机串行 I/O 口工作在方式0情况下,8位段码或键盘扫描信号由口线 $P_{3.0}$ 串行输出, $P_{3.1}$ 给出同步脉冲,移位寄存器LS164将串入的8位信号转换成并行信号输出并锁存,送给显示器或键盘,口线 $P_{3.4}$ 、 $P_{3.5}$ 为键盘反馈的查询线,而 $P_{1.0} \sim P_{1.5}$ 为显示的位选信号,实现动态显示。

3. 模数转换电路 包括ADC0809、或非门74LS02和反相器74LS06三块芯片,8031的 $P_{2.5}$ 和ALE分别作0809的片选和时钟信号,通过 \overline{RD} 和 \overline{WR} 对其操作,反映体温的模拟信号可以从0809的 IN_0 进入本机实现数字显示。

4. 报警和驱动电路 报警信号由 $P_{3.5}$ 控制,声响由一块556芯片形成,经功放后由扬声器发出。控制医疗电器的驱动信号与定时时间同步由 $P_{3.2}$ 输出,再经电子开关控制继电器触点的吸合或断开。

此外还有电源及上电复位等电路。

四、软件系统的介绍

本定时器软件系统的结构如图2所示。主要包括主程序、中断服务程序、校表程序、申请定时程序、查询报警程序和测量体温的显示程序等等。

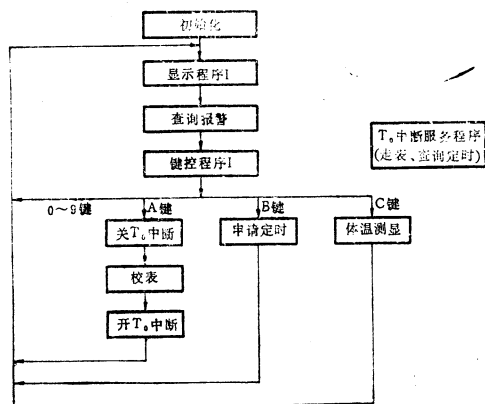


图2

1. 主程序包括：(1) 初始化,设堆栈指针,清数据缓冲区,设置工作单元和I/O口的初值,开中断,定时器编程和启动等。

(2) 查询报警标志单元决定是否转入报警状态。(3) 显示当前时间或系统状态,扫描键盘。无键闭合则在主程序中循环,有键闭合则根据键值转入相应的程序入口。

2. 中断服务程序功能有二：(1) 实现时钟管理, T_0 定时器在主程序中设定每1/8秒申请中断一次,每中断8次秒位加1并显示,且按六十进位改写分和时。(2) 查询定时单元,若有定时时间到的即记录床号、设置报警标志供主程序查询报警。

校表程序是在关中断(即停止走表)情况下进行的,键入当前的准确时间后时钟自动启动。

3. 申请定时的程序其流程图如图3所示。其功能是：(1) 不影响实时钟的运行和显示,所以启动第二套键扣和显示程序。(2) 把合法的床号和定时时间存入相应的地址单元。(3) 可以查询任一床位所剩下的定时时间,并可修改或取消。(4) 返回主程序。

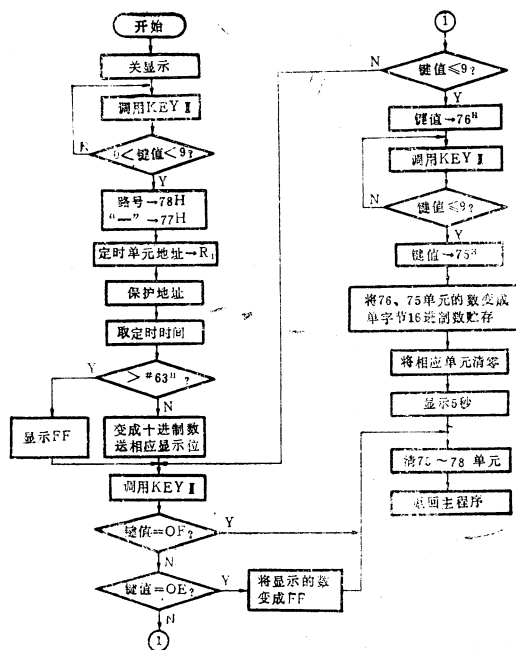


图3

水电站微机监控网络管理

华中理工大学 游大海 沈宗树 万永明 管家宝

摘要 本文针对水电站控制的特点,详细地探讨了微机用于水电站监控时的网络管理。尤其研究了用286微机作为控制管理机的可行性,主要内容包括编程技巧,中断应用、DOS和BIOS的利用和人机对话等内容。文中还介绍了黄坛口水电站微机监控系统。

一、引言

目前水电站计算机控制技术正在迅速发展,基本上还没有标准化成型的产品。因此,在设计某电站的计算机控制系统时,需要进行大量的方案比较和研制工作,这就在某种程度上妨碍了计算机控制在水电站上的广泛使用与推广。造成这种现象的原因主要在于硬件的配置和软件的编制难以做成标准模式。

鉴于上述原因,我们研究了水电站微机网络监控程序的开发、设计、编制和调试,并朝程序的标准化做了一定的工作。

二、水电站监控的特点和所实现的功能

水电站大小有别,在系统中作用地位也不尽相同,但大多数水电站对计算机监控系统的要求是基本相同的。通常监控系统的功能可具体描述如:

1. **数据采集和处理** 主要有模拟数据采集和处理,开关数据的采集和处理,脉冲量的数据采集和处理,数字量的采集和处理。

2. **控制与调节** 主要包括工况转换、线路送停电,断路器断合、刀闸开闭、变压器分接头有载调节等,并对其进行监视和记录;有功、无功的调节。

3. **厂内经济运行和自动发电自动电压控制** 经济运行包括机组优化组合和机组间负荷分配,自动发电控制包括按指令跟踪频率和自动跟踪系统频率。

4. **运行管理自动化** 主要有事故故障记录,操作记录,运行参数记录,各种报表记录。

5. **人机联系** 主要包括人机会话、各种画面和表格显示的召唤。

水电站计算机监控概括起来有如下特点:(1)系统较复杂;(2)可靠性要求高;(3)抗干扰能力强。

三、IBM PC及兼容机的特点

3. **提高可靠性的措施。**(1)降低电源内阻,以防通过电源内阻耦合产生干扰。(2)各芯片电源入口均有退耦电容,减少各部之间的互相影响。(3)继电器触点加吸收电容。(4)在软件上除去保证各程序模块的可靠以外,还在空白的程序存储空间填入00、00……00、0200、00字节,在有干扰的情况下能自动跳回主程序,保证定时功能有效。

参考文献(略)

五、几个问题的解决

1. 为了保证计时和定时的连续和准确,本系统采用二套扫描和显示程序,保证在实时钟显示消隐的情况下,软时钟继续运行,随时可恢复对当前时间的显示。

2. **重叠报警的处理。**如果有两路以上的定时时间,在极其相近的时间以内到时,报警不会重叠或丢失,软件中将各床位的到时均设相应标志,可以依次报警并闪烁显示床号。

IBM PC及其兼容机在微型计算机领域里产生的巨大影响是众所周知的。监控网络管理软件编制用到的PC机主要功能如下:

1. 确定系统资源 (1) 查找并确定磁盘驱动器的数目及类型, 使用INT 21H功能调用1CH能完成此功能; (2) 确定RAM, 使用INT 12H可完成此功能。

2. 中断管理 中断是微机中非常重要的功能之一。其中编制自己的中断又是监控程序的重要任务之一。INT 21H的功能25H可将中断向量设置到特定的地址。

3. 程序管理 (1) 分配/撤消分配内存, INT 21H的48H—4AH号能完成此功能。当监控管理软件较大, 内存较紧张时用到此功能; (2) 使用DOS用户指令, 在监控管理软件中有时可利用微机中现有的DOS用户指令, 如DIR或CHKDSK。这可大大简化编程。INT 21H的功能4BH可完成此功能; (3) 程序覆盖, 当某部分管理程序无需经常使用, 可将其放在磁盘中, 以节省内存。需要再调入内存, INT 21H的功能4BH可完成此功能。

4. 置/读时间 日期、中断INT 21H的功能2CH读取时间, 功能2DH设置时间; 中断INT 21H的功能2AH和2BH用于读取和设置日期。一般情况下它们不能完成实时时钟功能, 即满24小时后不自动将日进一。

5. 置/读时实时钟 当加装实时时钟卡后, 能完成自动日历计时。另外AT机中断1AH的扩展功能也能完成日历自动计时。

6. 延时操作 利用INT 1AH的功能0, 把要延时的脉冲数(18.2次/秒)和日历计数相加, 然后不断取日历计数, 并随时和期望值比较, 相等时延时结束。

7. 发声 在某一特定的时刻需要以声音提示, 此时只要以所需的频率打开和关断8255端口B的位1(与喇叭相连), 就可以从8255外围接口转接器发声。

8. 键盘控制 (1) 清键盘缓冲区可利

用INT 21H的功能0CH, (2) 等待键入字符但不回显, INT 21H的功能7和8完成此功能, 功能8还能检查Ctrl—Break功能; (3) 等待键入字符并回显, INT 21H的功能1等待接收字符并在光标处显示; (4) 立即获取字符, 当程序不能停下等待字符输入时, 可用INT 21H的功能6完成此功能, 但字符不回显; (5) 检查/设置双态键和换档键的状态, 利用INT 16H的功能2可完成此功能。

9. 视频显示 (1) 光标控制, 在人机对话需要移动光标, INT 10H的功能2可以设置光标位置; 同时这个功能也可用来打开/关闭光标; (2) 屏幕字符显示, 操作系统提供了六个‘写’屏例程, 三个属于BIOS, 三个属于DOS, 它们差异在于‘写’完字符之后光标是移动, 是否导致滚屏, 是否设置字符的属性或颜色等; (3) 画点状图形, 这个功能可在屏幕上画线, 画图形或表格。

10. 磁盘文件管理 (1) 确定可用磁盘空间, INT 21H的功能36H能说明多少磁盘空间是空的; (2) 读/更改根目录, INT 21H的功能4EH可寻找第一个文件名。功能4FH可寻找后续文件名; (3) 建立/删除文件INT 21H的功能3CH可建立一个文件; (4) 打开/关闭文件, INT 21H的功能3DH完成此功能; (5) 写顺序文件, 先用3DH打开文件, 然后用INT 21H功能42H完成“附加”数据顺序文件; (6) 读顺序文件, 文件由3DH打开, 然后由INT 21H的3FH功能顺序读出数据; (7) 写/读随机文件, 功能号同(5)、(6)。

11. 串行口通信 主要有下列工作: 串口片初始化, 设置当前通信端口, 监视串行端口状态数据发送, 数据接收。

以上主要综述了以汇编语言编制管理程序时所需要用到的功能, 若管理程序的实时性要求不高, 也可考虑用高级语言编制, 如BASIC。

四、监控网络程序的设计、

编制和调试

一般说来程序设计有下述几种方法,流程图,模块程序设计,结构程序设计以及自上向下设计。笔者以为上述几种方法各具特色。结合水电站监控网络程序的设计,我们以为下面的几个特点在编制程序时应充分注意并加以利用:1.水电站监控网络中各个子系统具有相对的独立性,如监测部分、调控部分、通信部分、离线计算部分等。它们之间的联系主要通过数据传送来实现,因此在程序设计时可以分别按模块程序设计,这时模块之间的划分是明确的。2.在动手编制每个模块前应了解清楚该部分的功能和要求。模块程序的编制可借助于流程图。流程图可清楚地标明这个模块的每个细节。3.由于整个网络管理程序较长,通常都是采用自上向下的设计,即把系统的整体细分为一系列的模块任务。每一个子任务用一个子任务程序来测试和调试,直到最后把整个任务集合起来。

在程序设计过程中应先有一个较粗的流程图,它包括主要电站功能框架,这个框架应设置得比较灵活,以便能容许地增添新的功能和内容。比较大的管理程序的完成不可避免地要经过多次增添、修改、删除等过程,但整个管理程序的逻辑结构要正确。

在管理程序中要多次地用到代码转换(十六进制转十进制,或反之,ASCII码转二进制或反之等)显示、打印、数据通信传输等常用模块应多花些时间完善。

程序的调试是非常重要的环,一般分两步完成:首先是排除语法错误,使编译、连接通过;然后是逻辑功能调试。一般是在各模块调试完毕后进行总体连调,这时的总体程序是相当大的,调试是较麻烦的,即使使用debug程序,如使用不当,也是相当费时的。比如在单步执行时,要确定某条语句的位置,然后单步执行下面一段程序就是如

此,根据经验,我们在所要单步执行的语句前插入下面一段程序可使调试大为方便:
PUSH AX; MOV AH, 0; MOV AL, SCR (SCR为屏幕方式,一般取3); INT10H; POP AX; INT 3。执行了它们后屏幕设置为80×25的文本方式。同时停在所要单步执行的语句前。这个方法也同样可用于模块程序的调试,这时用debug的T指令执行INT3的后一条指令。

水电站监控网络的状态是较多的,现场调试时很难用真实的工况来满足计算机监控设备所要求的全部外部条件,因此很难真实试验计算机监控装置对所有工况的执行、转换是否都是正确的。针对这种情况可以专门设置一段初始化程序,对各种工况进行临时设置,以替代真实工况,减少对现场配合的要求,又不影响程序调试。

总之程序的调试是非常灵活的。这需要对现场的熟悉和经验的积累。

五、模块化问题

随着水电站计算机监控工作的开展,为了提高工作效率,我们把下面部分进行了模块化处理。只要保护好现场,处理好出入口,均可处理为子程序或宏指令。

- 1.彩色模式设置;
- 2.光标定位;
- 3.读光标;
- 4.清屏;
- 5.显示汉字或英文字幕(包括清屏显示和非清屏定位显示);
- 6.画彩色坐标轴;
- 7.画彩色图形块;
- 8.设置中断向量;
- 9.BCD码转换十六进制数;
- 10.十六进制转BCD码或ASCII码;
- 11.BCD码转ASCII码;
- 12.在屏幕上写一个字符;
- 13.获取实时时钟;等。

以上均可处理为宏指令。下面可处理为

子程序;

- 14.有功调整;
- 15.无功调整;
- 16.电压调整;
- 17.频率调整;
- 18.日负荷输入;
- 19.屏幕编辑;
- 20.日负荷经济分配;
- 21.无功分配;
- 22.各种通信;
- 23.读磁盘文件;
- 24.写磁盘文件;
- 25.实时时钟显示;
- 26.状态输入并处理;
- 27.状态输出;
- 28.电量采集;
- 29.非电量采集;
- 30.各种打印;等。

由于上述这些模块的使用频率较高,将它们处理为模块后就使主网络管理程序的编制简明扼要。

六、实例研究

我们以黄坛口水电站为例研究微机监控网络的管理,系统总体结构如图1所示。该电站位于浙江省西部乌溪江上,是乌溪江梯级

水电站的第二级电站。电站设置有四台机组,两台主变、两条出线。电站库容0.79亿米³,水头30米,设计流量16米³/秒,是一个日调节的、以发电为主兼为下游工农业用水的水库。

该电站的微机监控要求有如下功能:

1.最优运行计划。包括:(1)根据给定的日负荷计划,确定次日每小时开机台数,台号和有、无功目标值,(2)可对上述计划进行人工修改;(3)修改当日小时及以后各小时的计划。

2.机组状态自动控制。包括:(1)按日运行计划自动执行开机、停机、发电一调相互转换;(2)键控开机、停机、发电一调相互转换;(3)键控:自动一小时发电一小时调相运行方式。

3.机组参数自动调节、包括:(1)按日运行计划,自动执行有功、无功的增减;(2)电站与系统解列时,自动执行电压调整;(3)键控上述各功能。

4.电气开关量监测;

5.电气模拟量监测;

6.非电量监测;

7.实时显示运行参数表;

8.实时显示运行状态画面;

9.显示记录参数和状态;

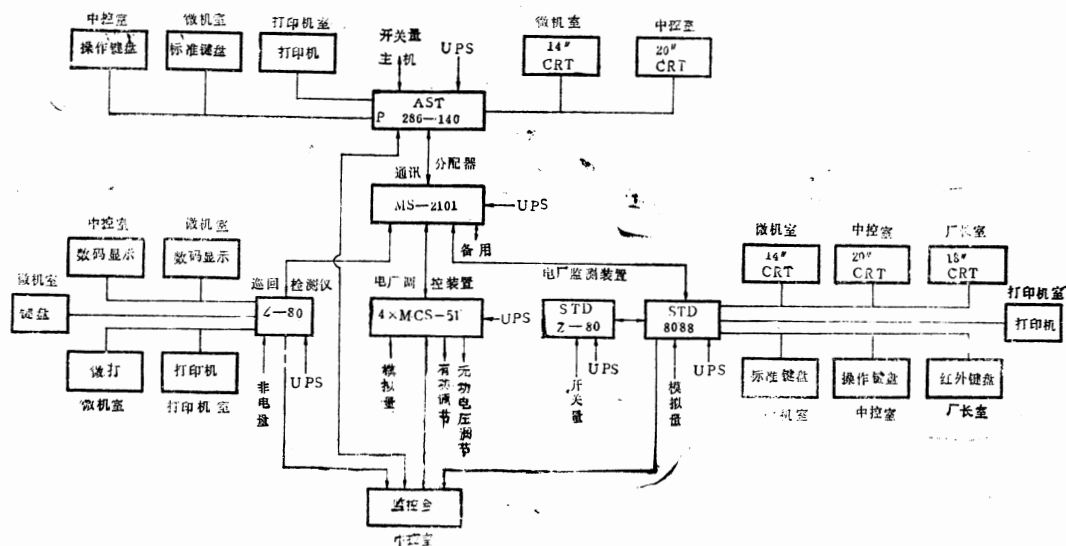


图1

- 根据上述功能要求和硬件配置,采用图2所示总体框图。

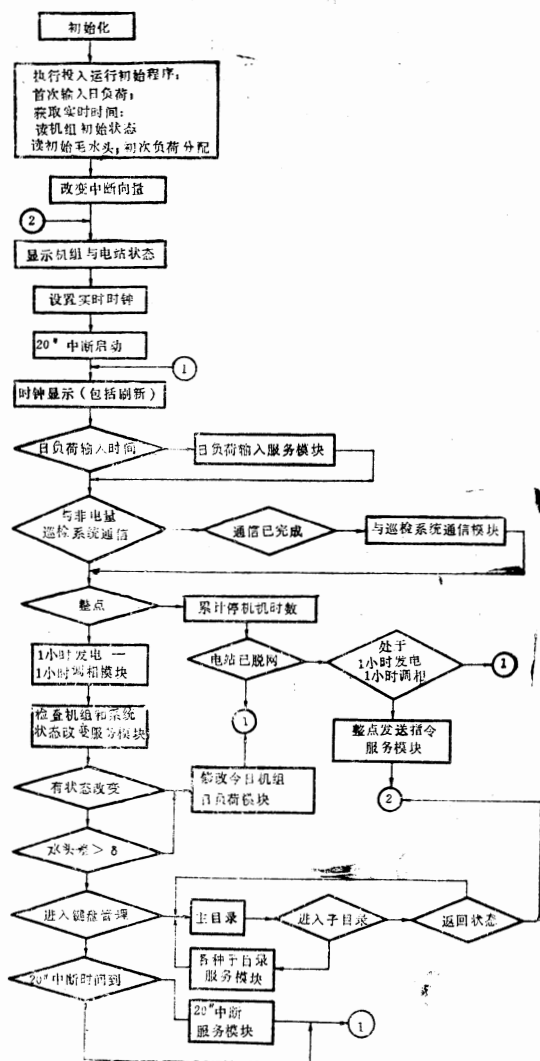


图2 总体框图

图2中各模块功能说明:

日负荷输入服务模块: 1) 通过键盘输入次日24小时的日负荷曲线; 2) 转换成16进制数; 3) 按最优方案分配各机组间的负

一些设计说明,非电量巡检子系统本身自备内存与打印机,故与主机的通信不是太

微机在自动门百万次寿命试验中的应用

沈阳黎明发动机制造公司自动门厂 王瑞刚 王丽清

摘要 本文介绍了以TP801单板机为核心构成的自动门百万次寿命试验控制系统。在自动门进行寿命试验时,该系统是用来对自动门进行实时控制和状态处理的。文中详细介绍了系统的硬件、软件结构,工作原理及主要功能。

一、问题的提出

在对自动门进行寿命试验时,要求长时间连续不间断的进行。在长达上千小时的试验过程中,不可能派人昼夜监测。而对自动门的实时控制和各种运行状态的处理,又远非人工所能胜任。笔者在对自动门进行百万次寿命试验时,采用TP801单板机设计了一套自动控制系统。实际投入运行后,取得了令人满意的效果,现将这一系统作一介绍。

多,每日一次即足够,故这个模块每日调用一次;而STD子系统与主机的联系较密切,主机每小时需向STD传送上、下游水位等,另外STD时间安排较紧,具体何时开始通信,须由STD决定。由主机每20"给STD一次通信机会,以响应STD的通信要求,因此与STD的通信模块放在20"中断服务模块。根据电厂要求设置了3种调控制方式:1)计算机全自动,基本无需人工干预;2)计算机键盘操作,这其中包括成组调节功能;3)计算机全部切除,改由常功部分操作。

由于该监控系统的子系统较多,要使整个系统统一协调工作,通信部分显得极为重要。为此我们在通信软硬件上进行了充分考虑。尤其在软件上进行了容错处理,任何一子系统出现通信故障均不会影响其它子系统正常工作,并在主机屏幕给出通信故障提示和故障性质。一旦故障排除,子系统可马上投入正常工作。

二、系统功能

该系统的主要功能有:

1.能自动定时向自动门控制器发出开、关门信号。开、关门信号的周期为10秒,开、关门时间相等,各5秒。

2.能自动记录自动门的开、关次数和出现的故障次数,并将记录结果自动打印出来。为避免打印机频繁起动,规定门每开、关二千次打印一次,门开、关二万次累计打

本文通过对实际水电站监控系统的研究,归纳了一些水电站微机监控网络程序设计,编制的一些原则、技巧。我们选用主机AST286作为上位机,通过现场考验,长期运行可靠性是很好的。监控网络管理程序的功能较齐全,满足电站需要,操作直观,选择性强,易于扩充功能。该电站监控系统已交付现场使用,工作正常。

参考文献

- 1 曹加恒,曾庆音,林青,宏汇编语言程序设计,武汉大学出版社,1988。
- 2 [美]威廉·默里,白晓笛译,80386/80286汇编语言程序设计,电子工业出版社,1988。
- 3 Larry E.Jordan,张连仲,付贵江译,IBMPC通信与网络,1985。
- 4 华中理工大学、乌溪江水电厂、黄坛口水电站微机经济运行实时监控系统的技术文件,1990。

七、结 论

印一次。以此循环,到一百万次时打印总运行次数,而对门出现的故障次数则随机打印。

3. 具有自动保护和自启动功能。当电机过热或自动门控制系统出现故障时,能自动停机,切断自动门控制系统的电源,而当自动门控制系统故障排除或电机温度降到其额定温升以下时,整个系统自动接通电源,重新工作。

4. 具有状态显示和声报警功能。能显示以下内容:门工作正常,开门、关门故障,电机热保护。并在电机过热和门出现故障时,发出声报警信号。

三、系统硬件结构及工作原理

该系统充分利用TP801单板机的功能,通过必要的接口处理,实现了对自动门的实时控制和各种状态的监测,达到了无人值守的目的。

系统的硬件框图如图1所示。

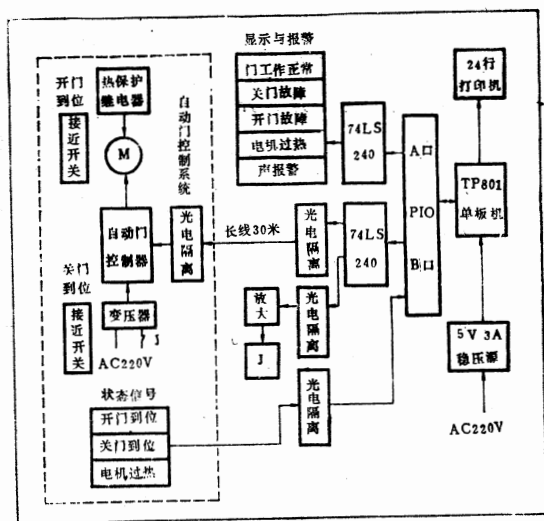


图1

TP801单板机的打印机接口,配置一台24行打印机,用于打印开、关门次数和故障次数,PIO的A口输出信号通过74LS240八缓冲器缓冲放大后,驱动发光二极管和音乐IC,用来显示自动门的各种运行状态,实现显示与声报警功能。PIO的B口输入、输出信

号通过缓冲,光电隔离等来实现对自动门控制器的实时控制和运行状态监测。其中PB₁输出的开、关门控制信号,通过缓冲、光电隔离、光电隔离到自动门控制器,用来控制门的开、关。PB₂输出的电源控制信号,经缓冲,光电隔离,放大。用来驱动继电器接通或切断电源。来自接近开关和电机热保护继电器的状态信号,经光电隔离后,送至PB口的PB₃~PB₅。

四、系统软件结构

系统硬件确定后,主要通过编制完整的软件来实现门的开、关、状态检测、状态显示、数据记录及打印等功能。系统软件可分为三部分:主程序,中断服务子程序,打印字符子程序。

1. 主程序

主程序流程见图2。主要是由初始化程序、计数程序、判断程序等组成。初始化程序主要完成PIO接口芯片的控制状态,中断矢量,中断控制字的设定。即:PIO的A口设定为字节输出方式,PIO的B口为位控方式,允许“或”产生中断。计数、判断程序主要完成门开、关次数的记录和判断,然后

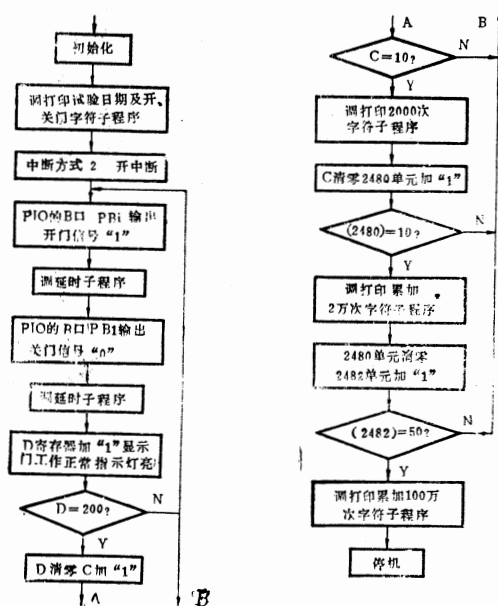


图2 主程序流程图

调用相应的打印字符子程序进行打印。

2. 中断服务子程序

中断服务子程序用来实现门的故障性质判定, 切断、接通自动门控制系统电源, 门的工作状态显示与报警及定时等功能。流程图如图3。

3. 打印字符子程序

打印字符子程序是由初始化程序、打印内容子程序, 打印程序等组成。初始化程序主要完成把将要打印的内容首地址写入2F00、2F01单元、末地址写入2F02、2F03单元, 字符点、行数为10, 存入2F04单元中。打印内容子程序主要负责把将要打印的试验日期、运行次数, 累加次数等内容存入存储器中。打印子程序为TP801“TPMP”中的控打程序。流程图如图4所示。

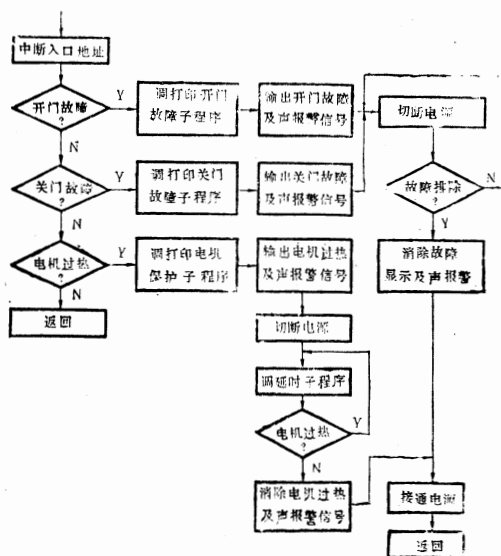


图3 中断服务子程序流程图

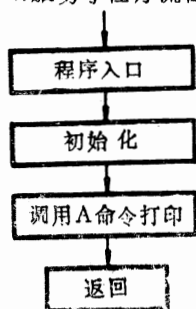


图4 打印子程序流程图

五、抗干扰措施

1. 长线处理

TP801单板机与自动门试验台的距离约30米。如此长的距离, 用于对自动门进行实时控制和状态检测的输入、输出信号是不能直接连到PIO口的, 必须进行抗干扰处理。特别是用于控制自动门控制器的开、关门信号更是如此。因为自动门控制器是以6748单片机做为主处理器的。它的输入信号的可靠性, 关系到整个系统能否正常工作。本系统对输入到PIO口的自动门状态信号, 采用一般的光电隔离方式。而对PIO输出到自动门控制器的开、关门信号则采用图5电路, 很好地解决了长线传输中的抗干扰问题。其中传输电缆采用双绞屏蔽电缆。信号传输为电流驱动方式, 而不是电平信号, 因而该电路有很强的抗干扰能力。

2. 电源处理

本系统开始组装时, TP801单板机电源与自动门控制系统用电源及电机冷却用电源采用同一电源插座板。系统试运行, 常因电风扇、车间冲床的起、停而导致单板机不能正常工作。经研究采用下面的电缆接线方式, 很好地解决了这一问题。单板机、自动门控制系统用电源及电机冷却用电源, 分别从三相电源中的A、B、C三相与零线之间获得320V交流电源, 从而避免了相互干扰。

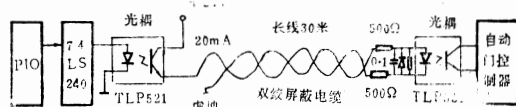


图5 长线传输中的抗干扰电路

六、结束语

本系统在设计完成后, 着手组装运行。于1990年3月正式开始进行自动门百万次寿命试验。在整个试验过程中, 历时近4个月, 实现了无人值守, 长期连续自动工作。

单片机在无线遥测觥标仪中的应用

福州大学计算中心 刘子义

摘要 本文介绍一种基于MCS—51系列单片机控制的WCZ100—C型智能无线遥测觥标仪的工作原理,软硬件设计及仪器的特点。

一、前言

我们与福州古楼无线电8厂联合研制的WCZ100—C型智能无线遥测觥标仪是与经纬仪配合,采用活动觥牌视准线法,通过无线电遥控测定水库、大坝、大型建筑物、桥梁、港口和码头等建筑物的水平位移。是水电站大坝维护的主要检测仪器之一。该仪器采用8031单片机控制并进行数据处理。能连续显示每次测量值并及时打印测量结果。

二、测量原理

要测定被测物(如大坝)的水平位移,首先必须在大坝两端固定山头上通过经纬仪建立一根视准线。测量时,司镜人员通过经纬仪观察固定在大坝待测位置上的觥牌。如偏离视准线,司镜人员操作无线电发射机发出让觥牌正走或反走信号。该信号经接收机接收并解调后,输入到8031单片机用以控制步进电机,使觥牌正走或反走,直至觥牌上的准线与视准线重合。这时,再通过发射机发出让单片机读取觥牌正向或反向移动距离(即被测物位移)的一次测量值。为了测量精确,通常要求重复测量4次或8次,然后取平均值。

三、硬件结构

觥标仪的硬件结构主要由觥牌移动装置,无线电发射机和接收机以及单片机控制系统三部分构成。

1. 觥牌移动装置

觥牌移动装置是由觥牌、丝杆和步进电

机组成。觥牌固定在丝杆上,在步进电机带动下,觥牌可沿丝杆正向或反向移动。步进电机每走一步觥牌移动0.01mm。丝杆的最大工作行程为100毫米。在觥牌移动装置的两端各装一个限位开关。

2. 无线电发射机和接收机

无线电发射机和接收机是实现遥控、遥测的关键部分。发射和接收采用调频方式,载频为150MHZ。用于控制8031单片机让步进电机正走或反走以及取数的信号,采用不同特征的调制信号。

3. 单片机系统

单片机系统的原理图如图1所示。本仪器采用8031单片机,字长8位,在一片40引脚的芯片上集成了CPU, RAM, I/O接口, 定时器等。振荡频率为6MHZ。外部扩展一片2732 4K EPROM存放应用程序。一片8155接口芯片作为并行输入和输出口,初始化时PA口和PC口设定为输出口, PB口设定为输入口。PA₂~PA₄接步进电机驱动电路,控制步进电机正走或反走。PA₀输出宽度固定的矩形脉冲。PB₁接一开关用于选择测量次数为4次或8次, PB₁=1表示测量8次, PB₁=0表示测量4次。PB₀和PB₇分别与觥牌移动装置的始端和终端的限位开关相连。一旦觥牌移动到两个端位,限位开关动作,只允许觥牌向相反方向移动,以保护丝杆不受损坏。

四位发光二极管显示器采用共阳极接法,动态显示,用于显示每次的测量值。8031单片机的P1口为段数据口。8155的PC口为扫描口。

8031 单片机直接连有一台 GP16 微型打印机, 用于打印每次水平位移测量值、平均值和中误差。

单片机的二个外部中断输入 $\overline{INT_0}$ 、 $\overline{INT_1}$ 和定时器外部输入 T_0 与无线电接收机连接。当无线电接收机接收到发射机发出的让步进电机正走或反走信号时, 经解调产生

$\overline{INT_1}$ 或 $\overline{INT_0}$ 中断脉冲, 使 CPU 中断让步进电机正走或反走一步。定时器 T_0 , 亦作外部中断使用。初始化时, T_0 设定为方式 2 (自动恢复常数) 外部计数方式, TH_0 和 TL_0 的初值均为 0FFH, 并允许 T_0 中断。当接收机收到取数信号时, 经解调产生计数脉冲, T_0 计数溢出, 使 CPU 产生一次中断完成取数工作。

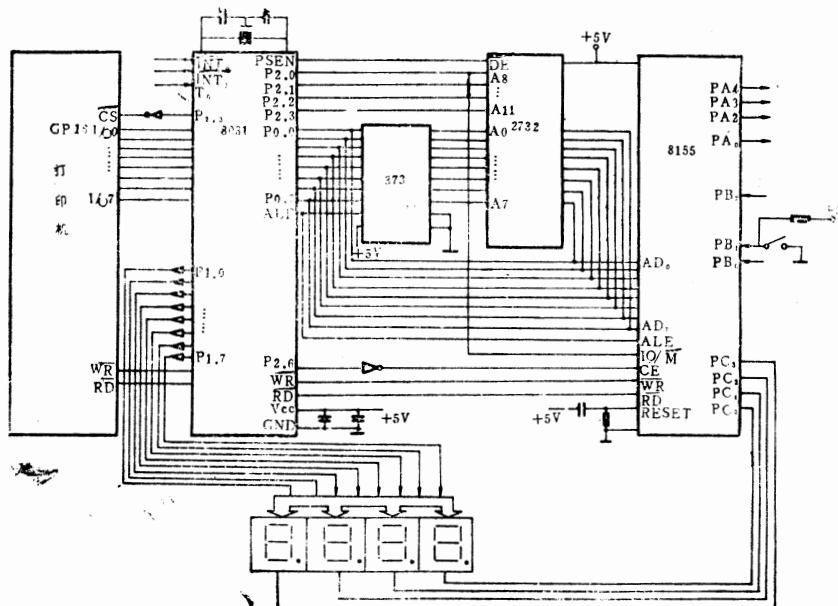


图1

四、软件设计

觥标仪的应用程序由主程序, 中断服务程序以及计算、显示、打印、代码转换等子程序构成。采用模块结构, 固化在 2732 EPROM 中, 8031 单片机片内 128 个字节和 8155 片内 256 个字节 RAM 用于存放测量的数据和计算的中间结果。

1. 主程序

主程序框图如图 2 所示。首先对 8031 和 8155 进行初始化; 清零计数单元; 传送计算参数。接着显示厂标 “F—3 厂” 3 秒后, 由 8155 的 $PB_1 = 0$ 或 1 决定重复测量的次数为 4 次或 8 次。随后循环调用显示子程序, 在这期间开放所有外部中断, 步进电机每走一步 CPU 中断一次, 显示器及时显示位移的计数

值。当 CPU 接收到司镜人员发出的取数中断信号时, 除完成取数工作外, 并把设定的测量次数减 1。直到取完规定的测量次数, 主程序开始转入计算并打印测量结果。

2. 中断服务程序

中断服务程序包括: 步进电机正走一步、反走一步和取数三个程序。

(1) 步进电机正走一步服务程序框图如图 3 所示。执行步进电机正走一步服务程序时, 除让步进电机正走一步外, 还在 8155 的 PA_0 输出周期为 5ms 的矩形脉冲供控制使用。步进电机每正走一步, 位移的计数值加 1。当觥牌移动到终端位时, 终端限位开关动作, 8155 的 $PB_7 = 0$ 。这时禁止外部中断 1 中断, 允许外部中断 0 中断, 不允许步进电机正走, 只允许反走。一旦计数单元的计数值为 9999 而觥牌尚未移动到终端位置, 步进电

处理●

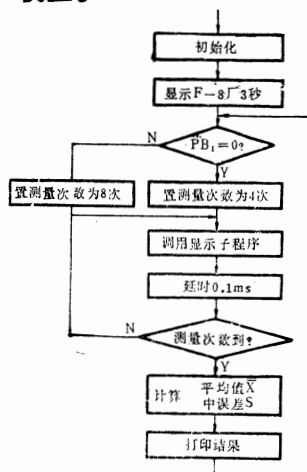


图3

如图4所示。同样，执行步进电机反走一步程序时，除让步进电机反走一步外，还在8155的PA0输出周期为5ms的矩形脉冲供控制使用。步进电机每反走一步，位移的计数值减1。当视牌移动到始端位时，始端限位开关动作，8155的PB0=0，这时禁止外部中断0中断，允许外部中断1中断。不允许步进电机反走，只允许正走。一旦视牌移动到起始位置时，而计数单元的计数值出现负值，就自动清零计数单元进行校正。

3. 计算

设 n 次测量值为 X_1, X_2, \dots, X_n ($n=4$ 或 8)

平均值: $\bar{X} = \sum X_i / n$

中误差: $S = (\sum (\bar{X} - X_1)^2 / n(n-1))^{1/2}$

为了提高精度,减少计算误差,计算程序采用多字节浮点运算并对计算结果作舍入

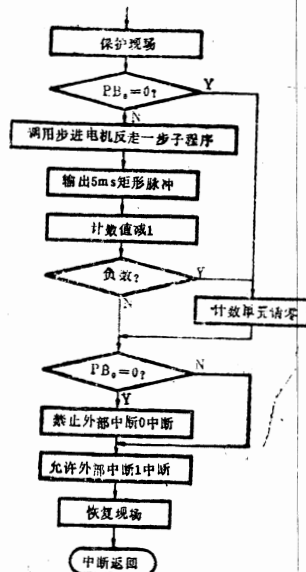
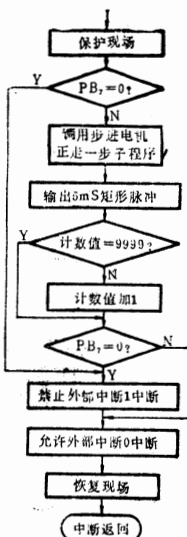


图5

五、结束语

1.采用无线电遥测技术,司镜人员可直接遥控觇牌的移动。提高了照准精度和作业速度。

2.采用单片机控制和数据处理,可现场计算并打印每次的测量值、平均值和中误差。大大减轻人工计算的工作量和减少计算误差。

3.该仪器体积小，结构牢靠，携带方便，性能稳定很适合于野外作业。

4. 主要技术指标:

测量范围: $\pm 50\text{mm}$, 工作行程: 100mm
最小读数: 0.01mm , 综合误差 $< 0.1\text{mm}$,
遥测距离: $\geq 1000\text{m}$

主要参考文献

- 1 徐君毅等, “单片微型计算机原理与应用”, 上海科学技术出版社, 1988。
- 2 复旦大学微机开发应用研究室, “MCS-51单片微机原理与应用”, 1988。
- 3 张友德等, “MCS-51单片微机实用子程序及其应用”, 1988。

微机磁场成型控制

桂林矿产地质研究院

马滋宇

郭友民

摘要 装在模具中的磁粉，在给定的压力下通过强磁场作用，就形成各种规格的磁性材料元件。本文介绍一个用8031单片机构成的磁场成型加工控制系统，着重介绍该系统的功能、硬、软件设计等。

一、系统功能

按生产工艺要求，本系统具有如下功能：

1. 在磁粉工件加压之前短时间给磁场电磁铁通电，以对磁粉工件进行定向处理。

2. 工件在单柱油压机上加上所需压力之后，保持磁场电磁铁正向通电（约50A）充磁5秒钟左右（时间可调）。

3. 正向通电之后，再反向通电（电流大小同上）进行退磁处理，退磁时间0.2—3秒（可调）。

4. 动态显示充、退磁时间。

5. 接收操作人员根据生产要求输入所给定的充、退磁时间。

6. 当给定压力之后，由系统控制锁定保持；工件加工完毕之后，自动解锁复位。

二、硬件设计

本系统硬件由8031, 2716/2732, 74LS373, 8155等主要组件组成。8031, 2716/2732, 74LS373构成最小系统，8155用于扩展并行口。硬件逻辑如图1示。

说明：1. 时钟频率6MHZ；2. EPROM 2716/2732用作扩展外部程序存储器；3. 8155芯片作为显示器及控制磁场电磁铁供电电路的脉冲输出口；4. P1口用作接收充、退磁时间的输入，同时P1口还定时送出脉冲以控制直流电压的正/反向切换及释放压力保持电磁铁锁；5. 采用二位数码管构成时间显示器；6. 电源：+5V——微机系统供电，+12V——为执行电路供电。

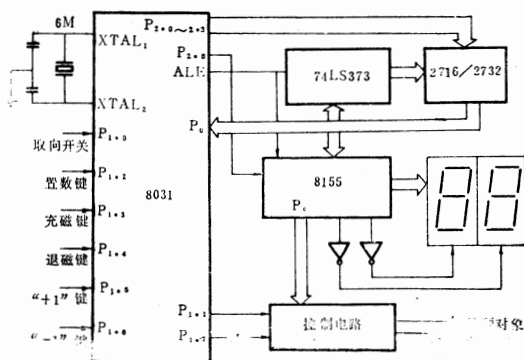


图1 系统硬件框图

三、软件设计

整个软件系统由主控程序（见图2）、显示子程序（见图3）、置数子程序、计数子程序及延时子程序等组成。

1. 主控程序 主控程序完成8031、8155的初始化，充、退磁时间的予置，正向充磁及反向退磁的控制，保持压力电磁铁锁的释放等功能。

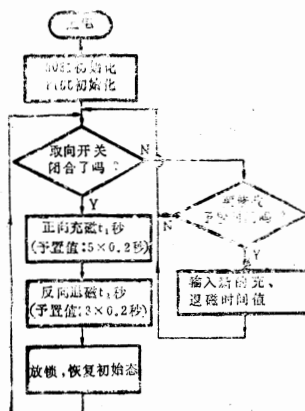


图2 主控程序框图

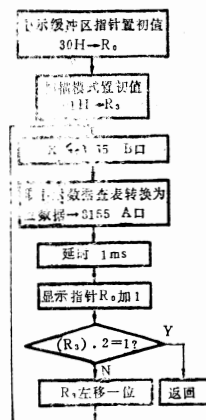


图3 显示子程序框图

2. 显示程序 完成存放在缓冲区的充、退磁时间显示。

3. 置数 (+1、-1) 子程序: 此程序能像调“电子表”的时间那样连续+1或-1, 以方便地实现充、退磁时间数的设置。程序框图见图4。

4. 计数子程序: 该子程序完成两个功能: (1) 定显示时间单位值为200ms; (2) 动态显示充、退磁时间倒数数值(即每次减一个单位值后的时间), 程序框图如图5示。

5. 延时子程序: 按不同延时时间要求, 有1ms、6ms和0.5秒等三种延时子程序。

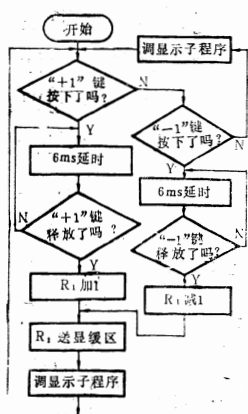


图4 置数程序框图

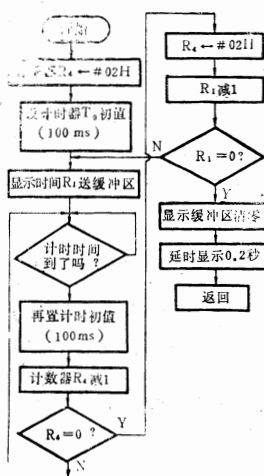


图5 计数子程序框图

四、抗干扰措施

为保证控制系统工作稳定可靠, 采取了

如下抗干扰措施:

1. 对供电电源及其它主要电路环节利用电容器进行高、低频滤波。

2. 对电磁元件采用旁路二极管及RC吸收电路消除干扰影响。

3. 组件中空余的端子均实际电路情况接地或接电源。

4. 对微机系统和执行电路采用独立供电, 以避免公共电源相互干扰。

5. 在软件中加消除抖动措施来提高系统的抗干扰能力。

由于采取了必要的抗干扰措施, 该系统在强磁场干扰的环境中, 一直正常工作。

五、结束语

本系统配合单柱油压机进行一次性给压杆加压(压模下移)操作, 完成对磁场中磁粉工件的取向、充磁及退磁等生产过程的自动控制。把由原来人工操作的九道工序减少为二道, 一次加工的时间由3分钟减到1分钟。这不仅大大减小了工人的劳动强度, 提高了生产效率, 而且由于对工件加工时的充、退磁时间准确控制, 使产品的质量得到了进一步的提高。

参考文献

- 1 单片微机原理与系统设计技术, 复旦大学微机研究室。
- 2 通用单片微机仿真器, SICE复旦大学微机研究室。

日第五代电子计算机今秋试运行

日本最近推出第五代“会思考”的电子计算机“PIM”并行网络系统, 既可用于演算, 又可进行一定逻辑推理。计划于今秋投入试运行。

日本第五代计算机的研制工作是1982年开始实施的, 预计1992年春季告一段落。日本通产省决定1992年后, 采取以PIM并行网络系统的并行程序设计软件为中心的方针, 把第五代电子计算机的硬件成果推向生物工程和半导体开发等高新技术产业的应用前沿。

Prime 计算机自动记账系统 (PCAAS) 的设计与实现

西安交通大学 杨振平
西北建筑工程学院 张维翰

摘要 本文介绍Prime计算机自动记账系统简称PCAAS, 是为Prime系列计算机所编制的, 适合于我国国情的管理软件。文中介绍该系统的总体结构, 设计思想及实现方案, 给出使用的方法和实例。

一、前言

PCAAS (即Prime计算机自动记账系统的简称) 是为Prime系列计算机所编制的, 适合于我国国情的管理软件。PCAAS在多用户操作系统环境 (Primos) 下运行, 整个系统资源共享, 以文件作为处理对象, 较方便地实现了各种数据的建立, 统计, 查询。具有功能齐全, 操作方便, 图表清楚, 安全性强这样一些特点。从管理者的角度看, PCAAS实现了取消预登记这一环节, 减轻了账务人员大量、繁琐的统计计算, 提高了计算机的科学管理水平。从使用者的角度看, PCAAS的灵活性在于以下几点:

1. 使用方便 本系列机安装上PCAAS软件后, 面向用户的终端室是开放的, 用户可以直接上机计算, 一次用机时间没有限制。机房按PCAAS所记录的实用用机时间收取费用。

2. 收费合理 PCAAS统计一个用户的实际用机时间不只是终端时间, 而是记录出用户所占用的终端时间 (记为CRT), CPU时间 (记为CPU), 输入输出时间 (记为I/O)。并按照不同比例收取三者的费用作为整个费用, 改变了过去只按CRT计算费用的不合理作法。

3. 查询方便 PCAAS为用户提供了一些命令如WATCHT, SERCHA等, 允许用户查询记账系统中与自己有关的某种信息。

二、PCAAS设计思想及各子系统功能描述

PCAAS包括四个子系统, 即建账子系统, 记账、累计子系统, 查询子系统, 统计报表子系统, 每个子系统均以文件作为处理对象。根据需要整个PCAAS使用六种文件类型如原始文件、日记录文件、总记录文件、非法用户文件、中间数据文件、报表文件。这些文件通过子系统单访或互访。建账子系统主要产生原始文件; 记账、累计子系统产生中间数据文件、日记录文件, 同时形成总记录文件、非法用户文件; 统计报表子系统产生报表文件; 查询子系统访问有关的文件。图1说明了PCAAS主要的结构及各部分间的关系。

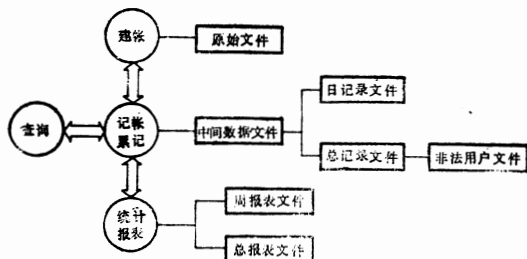


图1: PCAAS结构框图

原始文件: 为管理人员和PCAAS提供一些必须的数据。其中主要包括: 用户姓名, 单位, 登记日期, 预交上机费用 (即金额), 目录名称 (即UFD名), 所属类别等。

中间数据文件：是记账，累计子系统中一个重要的文件，用来记录不同用户在不同终端或同一用户在不同终端上录入到系统的日期，时间。

日记录文件：是记账，累计系统处理的一个结果文件，是整个PCAAS的核心部分。它用来记录，累计用户当天所使用的机器时间，打印后使用的纸张数和其它一些费用。PCAAS保留日记录文件期限为一周，一周后自动消失。

总记录文件：是日记录文件的累计，必要时销毁。

非法用户文件：是根据总记录文件及原始文件所提供的信息，核算用户的资源（即予交的金额）是否耗尽，如果资源已耗尽，称该用户为非法用户并将该用户目录名登入该文件中。

周报文件：是将一星期内日记录文件汇总、同时计算出每个用户一周内费用的总和，并以清晰的表格打印出来。

总报表文件：是输出从×年×月×日开始至处理该报表的时间每个用户使用机器时间，打印的纸张数，总的杂费，及总的费用，还剩余多少费用等项目。

1. 建账子系统

凡需上机的用户，都须经过建账子系统，登记如下表项：用户姓名，工作单位，登记日期予交金额，注册目录名，类别号。其中予交金额包括计算费用，需打印的纸张费用和其它一些杂费（如：文件的保管费，文件的转带费，键盘录入程序费等）。类别号为一序号（如：1，2，3，…），这里是将用户分为几个等级，每个级别对应于一序号。PCAAS收费时，可按照事先安排好的收费标准表进行。

该子系统还具有追加用户资源（即用户需增加的费用）的功能。用户一旦发现自己的资源已耗尽或已接到由记账累计子系统发出的“请追加新的资源”信息后，可带上新的投资由专管账务的人员或获许的管理人员

通过该子系统来完成。

该子系统是PCAAS中非常重要的一部分，为防止非财务人员使用该子系统，在PCAAS的设计中采取多重保护措施。主要表现为：①终端室所有用户没有权利使用该子系统的命令。②该项工作只能在指定的终端设备上进行。③进行操作的财务人员必须带有自己工作证号码的密码。以上三点均满足时，才能进入该子系统工作。否则，系统提示信息“NO, Access Rights”（非访问权）。为防止误操作或修改错误，该子系统对各操作步骤以重显示方式反问，从而可以减少人为的错误。同时，凡在该子系统下操作的人员，无论做什么工作，终端上都要显示操作者的工作证号码，若发现不符，请及时修改密码。

2. 记账、累计子系统

记账、累计子系统是整个PCAAS的核心部分。首先由主机系统的内部LOGIN命令检查用户的UFD名子、口令。若已通过，再启动该子系统访问非法用户文件，如果属于非法用户者，该子系统给用户发出“Your Resources Exhausted, Cannot Accept You”（资源耗尽，不予接受），同时就停止该用户的使用权。直到用户追加了新的投资后，方可使用。如果属于合法用户，这时该子系统访问主机的时钟系统，打开中间数据文件，将用户所占用终端的实际AMLC线号以及当时的日期，时间记入。

在用户打入LOGOUT命令结束本次运行时，该子系统完成如下工作：（1）打开日记录文件，将用户本次上机所使用的CRT, CPU, I/O时间分别以小时：分：秒，分：秒的形式直接写入或累计后写入日记录文件中。（2）打开总记录文件，将上机时间累计写入该文件中，同时按“收费标准表”计算出各项支出的总费用，检查是否超支，若超支则将UFD名登入非法用户文件中，并给出下面的信息：“Your Resources Exhausted, Please Add New Resources”。

该子系统访问的频率较高。为防止用户在处理某个文件期间，另一用户也要访问同一文件时丢失处理信息，产生其它错误，该子系统使用了随机存取文件，有效地节约了在处理文件上所开销的时间，同时使用等待排队子程序技术。

3. 查询子系统

本子系统为管理人员和用户提供了非常方便的查询手段。通过简单的命令，可以访问整个记账系统。本子系统中主要的查询命令有WATCHT, SERCHA, 用户利用它可以查询与自己有关的各种信息。

4. 统计报表子系统

本子系统给用户提供一个表：周报表，总报表。功能见周报表文件，总报表文件。

三、命令的使用方法及举例

1. WATCHT 命令，使用格式如下：

WATCHT { -L[ALL] },
-ALL, -HELP

或 WATCHT {-L, -A} [{MON,
TUE, WED, THU,
FRI, SAT, SUN}]

格式隐用说明（以后均同）

（1）加下画线的命令必打入。

（2）{ } 或 [] 中为任选参数。其中 [] 内的参数可任选；{ } 中的参数必选取且一次只选取一个。

例 1. WATCHT -L

[WATCHT set 1986.11.5]

USER DATE CRT(h:m:s) CPU
(m:s)

YANG 01/05/87 3:57:37 4:58
I/O(m:s) P300 u80

2:50 10 0

此例说明，当前用户YANG，87年1月5日总的计算时间为CRT用了3小时57分37秒，CPU用了4分58秒，I/O用了2分50秒，宽行纸用了10张，窄行打印机没有用。

2. SERCHA 命令，使用格式如下：

SERCHA -L[{ PA, ALL }]

任选参数项说明：

-L——显示当前注册用户总账表。除计算时间，总的打印纸张外，还有杂费项目，核算出的总费用及结余费用等。

-L PA——显示任一用户总账表。

-L ALL——显示所有用户的总账表，并计算出总的收支情况。

例2 说明用户YANG.F，类别号为5，预交现金600元，实际支出各种费用为588.99元，现结余11.01元。

3. CPACCOUNT 命令，用来建立用户账单，追加用户新的投资。只能由管理人员使用。

例3：图2说明CRACCOUNT的使用过程。

例2: SERCHA -L

[SERCHA set 1986.11.5]

USER: YANG. F	INPUT: 600.00	TYPE: 5 01/10/87
OUTPUT: CRT(sec)	CPU(sec)	I/O(sec)
915683	3994	33855
\$ 53415	\$ 26.63	\$ 28.21
P300(p)	U80(p)	FEE(\$)
0	0	0
\$ 0.00	\$ 0.00	\$ 0.00
COST: \$588.99	REMAINS: \$11.01	

如何在Turbo Prolog系统中调用汇编语言定义的数据库

安徽大学

梁继鸿

安徽医科大学

胡涛

摘要 众所周知,用Turbo Prolog语言开发专家系统,知识库的存放容量较有限,存取效率较低。这是因为用Turbo Prolog进行多模块程序设计时,生成了庞大的目标代码的结果。那么能不能在不影响存取效率(甚至可能提高)的同时,压缩代码结构呢?回答是肯定的,那就是用汇编语言建立数据库,再由Turbo Prolog语言调用执行。

一、接口说明

1. 外部谓词说明

为了通知Turbo Prolog系统某个整体谓词是用汇编语言来实现的,则要对整体谓词追加一个规则说明:

global predicates

谓词名(参数类型表)-(流程模式表)

language asm

规则说明使所接口的语言明显化,其目的在于使激活记录和参数格式化,调用和返回约定、段定义、连接和初始化等问题得以简化。

2. 调用约定和参数

8086系列给程序员提供了远近两种调用子程序的选择,Turbo Prolog要求所有的调用和返回都是采用远调用。(FAR·)在下推参数后,它对于外部谓词的调用格式如下:

MOV DX, SEGMENT DATA

MOV DS, DX

CALL FAR PTR 外部谓词名

可以看出,当某个外部谓词正在执行时分配的数据段,仍是在主程序中命名为DATA的段,当交接到一个用汇编语言写的过程时,参数按正常顺序下推进栈;而被调用的过程则负责从堆栈中传递参数,并在返回时修正堆栈指针。

~~~~~

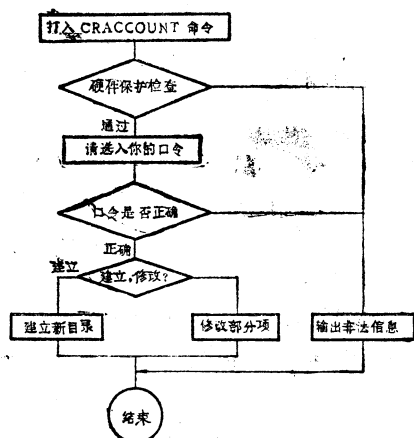


图2

4. STASTA命令,用来统计日记录文件,形成报表,并自动打印出来。(管理人员专用)

5. ADDA命令,用来增加,修改用户的杂费项目。(管理人员专用)。

## 四、结束语

PCAAS整个应用软件是用FORTRAN语言写成。这样凡是PRIME计算机均可安装本系统。同时使用本系统的单位可以根据需要提出新的要求,以增加,补充,完善本系统的功能,更好地为自己服务。

本系统稍加修改,可移植到其它大、中小型计算机上。

本系统在本校PRIME 550计算机上研制,并调试通过。经过半年多的运行实践,证实了该系统的可行性,稳定性,实用性。

8086系列的许多语言编译程序，有一个16位或32位指针的选择，其中16位指针对应于段缺省值。为了要访问所有内存空间，Turbo Prolog一般总是用32位指针。

Turbo Prolog的数据类型用下述方式实现：

|    |              |            |
|----|--------------|------------|
| 整数 | 2字节          |            |
| 实数 | 8字节 (IEEE格式) |            |
| 字符 | 1字节          | 下推进栈时占2字节  |
| 串  | 4字节          | 双字指针指向空终止串 |
| 符号 | 4字节          | 双字指针指向空终止串 |
| 复合 | 4字节          | 双字指针指向记录   |

对于输入参数，该值直接下推，其长度视类型而定；对于输出参数，下推进栈的32位指针指向返回数据所在单元的位置。

### 3. 命名约定

Turbo Prolog的同一谓词可有几个数据类型变种和流程模式变种，对于每一变种的调用都有一个分开的独立过程。为了要调用这些不同的过程，应给每个过程赋予不同的名字。命名约定是：以同一谓词名开头，后接从0开始的编号。例如，给定说明：

```
global predicates
subroutine (integer, integer)-
    (i, i), (i, o) language asm
subroutine (real, real)-
    (i, o), (i, i) language asm
```

谓词subroutine则有以下四个变种：

第一个变种(integer, integer)-  
(i, i)命名为subroutine\_0;  
第二个变种(integer, integer)-  
(i, o)命名为subroutine\_1;  
第三个变种(real, real)-(i, o)命名为subroutine\_2;  
第四个变种(real, real)-(i, i)命名为subroutine\_3.

应注意，对于i模式的不同数量类型变种，按照从左到右的高低权数，先说明整

型，后说明实型，否则不能实现整数类型的变种。

## 二、现场记录说明

### 1. 输入参数是整数、字符的情况

#### (1) 无参

```
double language ASM
[BP+02]  程序继续执行的地址
[BP+00]  开始执行前的地址
汇编程序需增加的约定：
double_0 PROC FAR
public double_0
PUSH BP
MOV BP, SP
:
POP BP
RET
double_0 ENDP
```

#### (2) 有一个参数

```
double(A)--(i) language asm
A的值, 2字节      [PB+06] A值
double之后, 4字节 [BP+02] 程序
                    继续执行的地址
BP, 2字节        [BP+00] 开始
                    执行 BP
```

```
double(A)--(o) language ASM
```

因为输出参量以32位指针下推进栈的，说明，现场记录如下：

```
[BP+10]
A, 4字节      [BP+06]
double之后, 4字节 [BP+02]
BP, 2字节      [BP+00]
```

汇编程序需增加的约定：

```
double_0 proc FAR
public double_0
PUSH BP
MOV BP, SP
:
MOV SP, BP
```



```

POP     BP
POP     AX
POP     BX
ADD     SP, +02 }
           (指输入) ADD
           SP+04 (指输出)
PUSH    BX
PUSH    AX
RET

```

double\_0 ENDP

### (3) 有两个参数

double(A, B) — (i, o) (i, i) (o, o)

language ASM

#### 1) (i, o) 调用约定:

[BP+10] A 的结束值, 占2字节

[BP+6] 存放B的地址, 占4字节

[BP+2] double执行后, 程序继续执行的地址, 4字节

[BP+0] 在开始执行至double之前设置的BP值, 2字节

#### 2) (i, i) 调用约定:

[BP+08] A, 2字节

[BP+06] B, 2字节

[BP+02] double后的地址, 4字节

[BP+00] BP, 2字节

#### 3) (o, o) 调用约定:

[BP+10] A, 4字节

[BP+06] B, 4字节

[BP+02] double之后执行地址, 4字节

[BP+00] BP, 2字节

汇编程序需增加的约定:

a.double\_0 Proc FAR

ADD SP, 6 (i--2, o--4)

double\_0 ENDP

b.double\_1 Proc FAR

ADD SP, 4 (i--2, i--2)

double\_1 ENDP

c.double\_2 Proc FAR

ADD SP, +08 (o--4, i--4)

double\_2 ENDP

### (4) 有三个参数的现场记录:

double(A, B, C) — (i, i, i), (i, i, o), (i, o, o), (o, o, o)

language ASM

a.(i, i, i)

汇编程序需增加的约定:

[BP+10] A, 2字节

[BP+08] B, 2字节

[BP+06] C, 2字节

ADD SP, 6 [BP+02] double之后, 4字节

[BP+00] BP, 2字节

b.(i, i, o)

汇编程序需增加的约定:

[BP+12] A, 2字节

[BP+10] B, 2字节

[BP+06] C, 4字节

ADD SP, 08 [BP+02]

[BP+00]

c.(i, o, o)

汇编程序需增加的约定:

[BP+14] A, 2字节

[BP+10] B, 4字节

[BP+06] C, 4字节

ADD SP, 10

d.(o, o, o)

汇编程序需增加的约定:

[BP+14] A, 4字节

[BP+10] B, 4字节

[BP+06] C, 4字节

ADD SP, 12

## 2. 输入参数是串、符号和复合情况 (占4字节)

### (1) 无参

见“输入参数是整数情况”

### (2) 有一个参数

a.double(A) — (i), (o) language ASM)

[BP+06] A, 4字节

[BP+02] double之后, 4字节

[BP+00] BP, 2字节

汇编程序需增加的约定:

double\_0 proc FAR

ADD SP, +04

double\_0 ENDP

### (3) 有两个参数

```
double(A, B) — (i, o), (i, i),
(o, o) language ASM
```

```
[BP+10] 输入A, 占4字节
[BP+06] 存放B, 占4字节
[BP+02] double之后, 4字节
[BP+00] BP, 2字节
```

汇编程序中增加的约定:

```
double_0 proc FAR
    ADD SP, 8
double_0 ENDP
```

### (4) 有三个参数的现场记录:

```
double(A, B, C) — (i, i, i), (i,
i, o), (i, o, o), (o, o, o)
language ASM
```

```
[BP+14] A, 4字节
[BP+10] B, 4字节
[BP+06] C, 4字节
[BP+02] double之后, 4字节
[BP+00] BP, 2字节
```

汇编程序需增加的约定:

```
double_0 proc FAR
    ADD SP, +12
double_0 ENDP
```

## 三、程序设计实例

通过汇编语言中提供的定义伪操作指令(DB, DW, DD, DQ, DT), 我们能够设计出各式各样的数据模式。

第一个例子是根据编号进行查表, 结构是:

```
STOKTAB DB '05', 'Excavators'
        DB '08', 'Lifters'
        .
        .
        DB '27', 'Pumps'
```

要求编号是由有智能的Prolog系统提供, 调用汇编语言进行查询, 将结果值再返回Prolog系统中, 程序如下:

```
PEXAM6.PRO
```

```
global predicates
search(string, string) —
    (i, o) language asm
goal
    write("This program demonstrates random scanning"), nl,
    write("table and only selective"), nl,
    write("number is: 05, 08, 09, 12, 23, 27"), nl,
    write("Please input certain number: "), readln(A)
    search(A, T), nl,
    write("The index of", A, ": "), write(T), nl.
```

```
AEXAM6.ASM
```

```
TITLE TABSRCH TABLE SEARCH
STACKSG SEGMENT PARA STACK 'STACK'
    DW 32 DUP(?)
STACKSG ENDS
DATASG SEGMENT PARA 'DATA'
ERR DB 'The number is not in the table'
STOKTAB DB '05', 'EXCAVATORS'
        DB '08', 'LIFTERS'
        DB '09', 'PRESSES'
        DB '12', 'VALVES'
        DB '23', 'PROCESSORS'
        DB '27', 'PUMPS'
DESCRN DB 10 DUP(?), '$'
DATASG ENDS
CODESG SEGMENT PARA 'CODE'
search_0 PROC FAR
    public search_0
    ASSUME CS:CODESG, DS:STACKSG, ES:DATASG
    push bp
    mov bp, sp
    MOV AX, DATASG
    MOV DS, AX
```

```

MOV ES, AX
CALL C10SRCH
lds si, dword ptr [bp+06]
lea ax, descrn
mov [si], ax
mov ax, datasg
mov [si+02], ax
mov sp, bp
pop bp
pop ax
pop bx
add sp, 08
push bx
push ax
RET
search_0 ENDP
C10SRCH PROC
MOV CX, 06
push ds
lds si, dword ptr [bp+10]
MOV AX, [si]
pop ds
LEA SI, STOKTAB
C20:
CMP AX, [SI]
JE C30
ADD SI, 12
LOOP C20
CALL R10ERR
RET
C30:
MOV CX, 05
LEA DI, DESCRN
INC SI
INC SI
REP MOVSW
RET
C10SRCH ENDP
R10ERR PROC
LEA DX, ERR
MOV AH, 09
INT 21H
RET

```

```

R10ERR ENDP
CODESG ENDS
END search_0
BEXAM6.BAT
LINK INIT +PEXAM6+AEXAM6+
PEXAM6.SYM, MIX6, ,
PROLOG

```

第二个例子说明了模块化的程序设计。由Prolog提供一个数字月份，通过ASM查询对应的英文单词，并将结果返回Prolog系统中。Prolog再对返回的结果进行处理。下面是两个Turbo Prolog模块与一个ASM模块传递数据的情形。

为了管理模块程序设计，Turbo Prolog引用了工程说明和全局说明两个概念。在编译和连接模块之前，每个模块必须以两个编译指令project和include开头。有且仅有一个模块包含一个目标段。

为掌握模块连接的具体步骤，假设两个模块名为：MAIN.PRO和SUB1.PRO，工程名GLOBDEF.PRO。

第一步。在Setup菜单下用Librarian命令建立一个LIBRARIAN文件。当系统提示“Name of module list?”时，键入：MYPROJ.然后将内容编辑为：

```

MYPROJ.PRJ
MAIN+SUB1+

```

第二步。建立、编辑、保存全局说明文件，使之成为。

```

GLOBDEF.PRO
global domains
name=string
global predicates
welcome(name)-(i)
access(integer, name)-(i, o)
language ASM

```

第三步。建立、编辑、保存主模块中的文件，使之成为：

```

MAIN.PRO
project "myproj"
include "globdef.pro"

```

```

predicates
.test
goal
    test.
clauses
    test, -clearwindow,
        write( "please input current
            month(1—12): " ), readint
            ( A ), nl,
        access( A, B ), nl,
        welcome( B ).

```

第四步.在Options菜单下将编译开关置为:“Compile to OBJ”,然后发布编译命令,以生成文件MAIN.OBJ和MYPROJ.SYM。

第五步.建立、编辑、保存子模块文件:

```

SUB1.PRO
project "MYPROJ"
include "globdef.pro"
clauses
    welcome( Name ), -
        write( "welcome" , Name ),
        write( "Nice to meet you." ),
        sound( 100, 200 ).

```

第六步.置编译开关“Compile to obj”,

第七步.建立汇编文件:

```

TITLE DIRECT TABLE ACCESS
STACKSG SEGMENT PARA STACK
    ' STACK'
        DW      32    DUP( ? )
STACKSG ENDS
DATASG SEGMENT PARA '
    DATA'
THREE DB 3
MONTAB DB ' JAN' , ' FEB' ,
        ' MAR' , ' APR' , ' M-
        AY' , ' JUN'
        DB ' JUL' , ' AUG' ,
        ' SEP' , ' OCT' , ' NO-
        V' , ' DEC'
ALFMON DB ' ? ? ? ' , ' $'

```

```

DATASG ENDS
CODESG SEGMENT PARA ' CO-
    DE'
access_0 PROC FAR
PUBLIC access_0
ASSUME CS,CODESG, DS:DATAS
G,SS:STACKSG:ES,DAT-
    ASG PUSH bp
    mov bp, sp
    MOV AX, DATASG
    MOV DS, AX
    MOV ES, AX
    mov ax, [bp+10]
    CMP AH, 00
    JZ C20
    SUB AH, AH
    ADD AL, 10
C20: CALL D10LOC
    CALL F10DISP
    Lds si,dword ptr [BP+06]
    LEA ax, ALFMON
    MOV [si], ax
    mov ax, datasg
    mov [si+02], ax
    mov sp, bp
    pop bp
    pop ax
    pop bx
    add sp, 06
    push bx
    push ax
    RET
access_0 ENDP
D10LOC PROC
    LEA SI, MONTAB
    DEC AL
    MUL THREE
    ADD SI, AX
    MOV CX, 03
    CLD
    LEA DI, ALFMON
    REP MOVSB
    RET

```



# 用微机辅助微波理论课教学

哈尔滨工业大学 刘健 李在清 黄记辉

**摘要** 本文论述微机在微波理论课教学中的作用,并具体介绍作者开发的Smith圆图辅助教学软件,微波晶体管放大器辅助图解软件及天线阵理论方向图分析软件。

## 一、微机在微波理论课教学中的作用

工科院校非微波专业微波理论课教学分为电磁场理论、微波技术、微波电子线路三门课程。由于微波理论课理论性较强,且较抽象,学生们难于很快理解与掌握,如电磁场理论中天线阵方向图,其形状与阵元类型及阵排布规律有关,而且是一个三维图形,不仅初学者难以想象,就是有一定经验的内行也较难考虑,作者开发的天线阵方向图计算机辅助分析软件,用户只需输入阵参数即可在CRT上得到理论方向图,使用十分方便。Smith圆图是微波技术教学的重要内容之一,由于头绪较多,学生们难于很快掌握,而其辅助教学软件可以让用户在微机上查算Smith圆图,并演示其过程,经过多次这样的训练后,学生们很快掌握了Smith圆图的原理与应用方法。微波晶体管放大器是微波电子线路的重要教学内容,其辅助图解分析软件将设计中稳定性分析、工作点选择、工作参数曲线以及放大器结构等均在CRT上图解出来,用户可通过键盘改变设计参数并立即观察设计结果,既生动又形象,而且

节省大量设计时间,很受学生们欢迎。综上所述,微机在微波理论课教学中可利用其图象显示功能生动形象地反映抽象概念,可以加快学生们应用所学知识设计具体电路的时间。因而在微波课教学中有很大辅助作用。

本文的几个教学软件,均用编译BASIC编成,在IBM-PC及兼容机上使用。

## 二、Smith圆图辅助教学软件

本软件具有如下功能:能根据用户输入参数在屏幕上绘制相应的圆族,如等电阻圆、等电抗圆、等反射系数圆等;能由用户控制一枚游标在圆图上移动,并查算当前位置的圆图参数,如 $|\Gamma|$ ,  $\varphi$ ,  $r$ ,  $x$ 等;能由用户输入参数在圆图上找到相应点;能定义简单传输线结构,并控制一枚游标沿线移动,同时在圆图上显示当前游标所对应的位置;能图示沿传输线电压、电流驻波分布;能将阻抗圆图变换为导纳圆图并演示三支节阻抗匹配过程。本文重点介绍传输线结构定义与图示分析。

将传输线的基本结构单元按表1方式加以定义,则图1a所示传输线可以这样描述

$T(1, 100, 0)$  ✓

$Z(1, 50, 0.25)$  ✓

```
D10LOC ENDP
F10DISP PROC
    LEA DX, ALFMON
    MOV AH, 09
    INT 21H
    RET
F10DISP ENDP
CODESEG ENDS
END access_0
```

第八步. 连接程序:


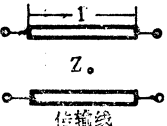


```
link init+subl+main+myproj.sym+
aexam5,myproj,,prolog
```

## 四、结束语

通过Prolog调用在ASM中定义的数据库并将之应用于专家系统的设计中,这种方法是可行的。例如在设计气象预报专家系统的过程中,若能采用ASM定义数据库则可大大地缩短可执行文件代码的长度,加快存取效率,提高系统的响应性能。参考文献略

S1↙  
 Z(2, 50, 0.25)↙  
 T(2, 0, 0)↙  
 S2↙  
 Z(3, 50, 0.5)↙  
 G(100, 50)↙  
 E↙——输入完毕

表1

| 元件                                                                                       | 格式及举例                                                                     |
|------------------------------------------------------------------------------------------|---------------------------------------------------------------------------|
| 终端<br>  | T (1, 50, 30)<br>标识符<br>第一个终端<br>$Z = \epsilon_0 + j30$                   |
| <br>传输线 | Z (3, 50, 0.25)<br>标识符<br>第三根传输线<br>$Z_0 = 50\Omega$<br>$l = 0.25\lambda$ |
| 分支<br>  | S1——第一个分支                                                                 |
| 源<br>  | G (100, 25)<br>标识符<br>$R_g = 100\Omega$<br>$U_a = 25V$                    |

输入完毕后,系统建立 T, Z, G 三个数组分别存放终端、传输线、源的参数,同时将各元件的字头符号取出按顺序建立一个检索卡(图1b)并为其分配一个指针单元。

图示传输线上的点在圆图上的位置关系的程序框图如图2a,显示画面如图2b。

传输线沿线驻波分布图示程序框图如图3a,显示结果如图3b。

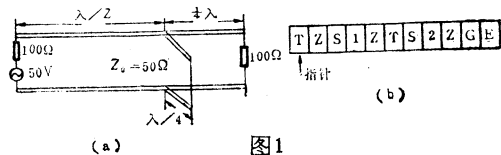


图1

### 三、微波晶体管放大器图解软件

采用传统的图解方法设计微波晶体管放大器,需根据晶体管的S参量经过复杂、烦琐的计算,求得各种曲线位置参数后标画在圆图上十分麻烦,设计周期也很长。

本软件只须输入晶体管参数,就能立即

在屏幕上绘出导纳圆图,并划分出不稳定区域,标出 $\Gamma_{s\text{opt}}$ 及 $G_{\text{max}}$ 的位置,用户可通过键盘输入参数增绘相应的曲线,如等G圆、等F圆等,用户根据这些曲线选择合适工作点,并通过键盘将一枚游标落在该点上,系统即自动计算设计出相应的放大器的结构参数并分析其性能指标。系统程序框图如图4a,一个实例如图4b。

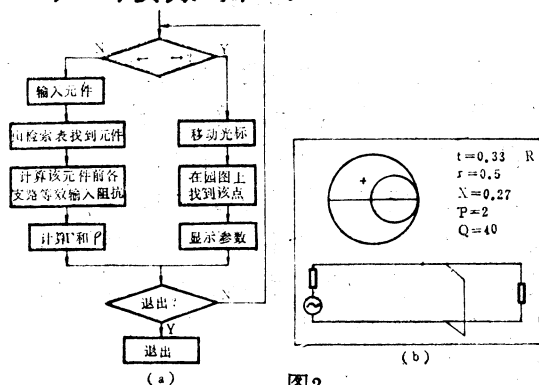


图2

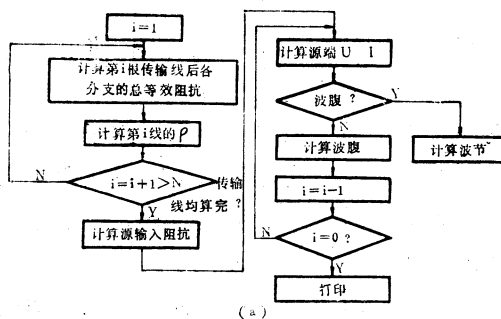


图3

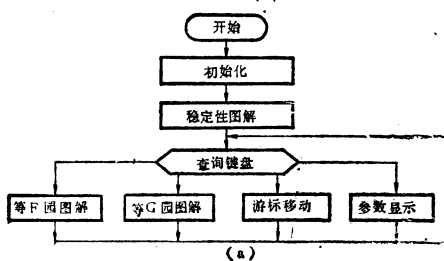


图4

## 四、天线阵理论方向图

### 的计算机辅助图解分析

运行本软件后,用户首先根据屏幕提示输入阵参数(包括阵元类型、工作频率、阵的维数、各维方向上阵元安放个数、间隔及相位关系等)。参数输入完毕,系统即计算天线阵的方向图并以数组形式存贮。计算结束后系统又提供给用户一张菜单,用户可选择显示XOY平面或XOZ平面或YOZ平面的方向图。显示形式是在画面上分别图示阵元方向图、阵因子方向图以及天线阵总体方向图(图5b),用户可放大显示任一幅图或将其变为对数坐标图示方向图与 $\theta$ 或 $\varphi$ 的关系,在这种状态下,用户可键控一枚游标沿曲线运动,并示出当前位置参数。

天线阵的方向性函数可表达如下式:

$$f(\theta, \varphi) = f_s(\theta, \varphi) \times f_1(\theta, \varphi) f_2(\theta, \varphi) f_3(\theta, \varphi) = f_s(\theta, \varphi) \times f_a(\theta, \varphi) \quad (1)$$

其中 $f(\theta, \varphi)$ 是天线阵的总体归一化方向性函数, $f_s(\theta, \varphi)$ 是阵元的方向性函数, $f_a(\theta, \varphi)$ 称为阵因子,由 $f_1(\theta, \varphi)$ ,  $f_2(\theta, \varphi)$ 和 $f_3(\theta, \varphi)$ 三部分乘积组成,它们分别代表三个空间方向上的阵列形式对总方向图的贡献。对(1)式取 $\theta=90^\circ$ 即得到XOY平面上方向图的投影;对(1)式取 $\varphi=0^\circ$ 即得到YOZ平面上方向图的投影;对(1)式取 $\varphi=90^\circ$ 即得到XOZ平面上方向图的投影,于是程序框图如图5a,一个实例如图5b。

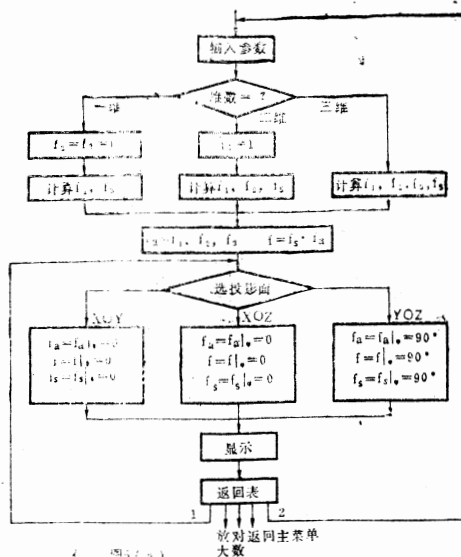


图5 (a)

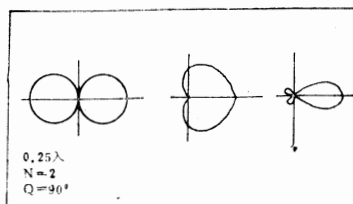


图5 (b) 无线阵方向图图示一例,从左至右依次为阵元方向图,阵因子,总方向图均为XOY平面,二元阵,阵间距0.25入,微电相位90°

图5 a(上) b(下)

## 五、微机在微波课教学中的巨大潜力

作者在利用微机辅助微波课教学中仅作了些初步的工作,这方面有巨大潜力可挖。比如讲授电磁波在各向异性媒质中传播时的特性时,往往较抽象,学生难于理解,讲授波导中的导行电磁波时,其波导内的场分布及壁电流也是难点,利用微机的图形显示及动画功能将其生动演示给学生,一定会收到好的效果。作者希望广大同行积极丰富微波课教学软件,充分发挥微机的辅助作用。

### 参考文献

- 1 孔道礼《微波技术》哈尔滨工业大学出版社,1988年。
- 2 梁昌洪《计算微波》西安电子科技大学出版社,1985年等。

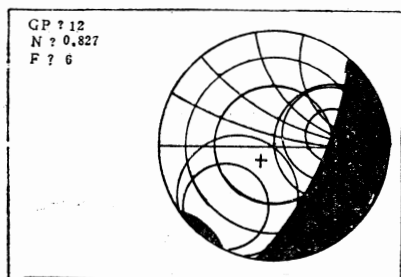


图4 (b) 图示微波晶体谐振器一例

晶体参数:  $S_{11} = 0.47 \angle -55^\circ$   $S_{12} = 0.081 \angle 76^\circ$   
 $S_{21} = 0.4 \angle 62^\circ$   $S_{22} = 0.91 \angle -43^\circ$

阴影区为不稳定区,“+”为工作点

图4

# “所见即所得”

天津海军勤务学院 · 朱桂林

**摘要** “所见即所得”，是排版系统、桌面系统、印刷系统、出版系统的特色之一，它完全地模拟整个文本的打印效果，使您在打印整篇文章之前就可以在屏幕上先观察一下整篇文章的排布情况。本文通过对字库的认识、对汉字、字符点阵分布、存储规律的认识和介绍汉字、字符点阵信息显示方法等而系统地介绍了排版系统、桌面系统、印刷系统、出版系统“所见即所得”技术的实现基础，以使您加深对汉字、字符属性的理解、增强系统开发能力和了解“所见即所得”技术的实质。

“所见即所得”，就是说，屏幕上见到的就是打印所得到的。“所见即所得”技术是近期图文处理新的突破，又称“模拟显示”，即模拟打印效果而显示，它逼真地在屏幕上描述打印信息的点阵分布的综合信息。实际上，将汉字、字符以16点阵、24点阵、32点阵、48点阵大小乃至放大若干倍的大小打印出来已是几年前的贡献了，大家都比较熟悉了；而将打印汉字“模拟显示”却是近几年排版、桌面、印刷系统新的贡献，“北大方正”、“华光”、“小麻雀”、“大文豪”、“909”、“DTP（联想）”、“puc（北大）”、“4S（四通）”、“M-6402（天大）”、“先锋CP910”、“新龙壹号”、“爱华DP-989”、“黎明LM-901”等系统都有此功能。

“所见即所得”，是以将16、24、32、48点阵汉字、字符显示出来为基础，加上对已知点阵信息进行放大、缩小、背景处理、修饰处理等处理而形成各种综合信息用以显示为拓宽，而形成的类打印技术。

本文主要介绍如何将16、24、32、48点阵汉字、字符显示出来。由于点阵的综合处理。显示牵涉到一些算法、原则，且各系统不尽相同，因此本文暂不作深入探讨。下面我们来认识一些典型的汉字、字符库，认识一下点阵的分布、存储情况，及如何将这些点阵信息调出来显示在屏幕上。

## 一、字库

目前字库很多，典型的有以下几种：

### 1. 24×24点阵字库

24点阵字库有CLIB24（607K）、CLIB24（588K，有的操作系统改为HIK24（CCDOS 2.13A），有的改为CLIBS、DOT（金山CCDOS 4.03、5.00，四通CCDOS 5.10）、HZK24S、HZK24F、HZK24K、HZK24H、HZK24E、HZK24T等。主要用于打印。

它们均列式存放汉字点阵，是为打印方便。库中每字占72字节，各库所含内容稍有不同：

CLIB24（607K）含256个西文字符（代码0—255）、1—15区纯中文字符、16区以后的汉字；

CLIB24（588K）含1—15区纯中文字符、16区以后汉字，不含西文字符；

HZK24S、HZK24F、HZK24K、HZK24H、HZK24E均是487K大小，仅含汉字，不包括纯中文字符和西文字符；

HZK24T只含有纯中文字符。

HZK24T+HZK24S=HZK24（588K）

### 2. 16×16点阵字库

16点阵字库，主要用于显示。有CCLIB（237K）、CCLIBJ、DOT（243K）等。



CCLIB (237K) 为 CCDOS 2.10、3.10 所用, CCLIBJ、DOT 为金山 CCDOS 4.03 所配。

它们均以行式 (或横式) 存放汉字、纯中文字符点阵, 是为显示方便组字。库中每字占 32 字节, 各库所含内容亦不同:

CCLIB (237K) 只含 1—7 区纯中文字符和 16 区以后汉字, 有的系统将 9 区线符改在了 6 区;

CCLIBJ、DOT (243K) 只含 1—9 区纯中文字符和 16 区以后汉字。

字卡的情况稍有不同, 含西文字符、1—9 区纯中文字符和 16 区以后汉字。

那么字库的西文字符存于何处呢? 于 BIOS 或 DOS 之中。

### 3. 32×32 点阵字库

32 点阵字库, 主要用于打印。有 CLIB32、ALIB32、BLIB32、DLIB32 等。

它们均以列式存放汉字、纯中文字符和西文字符点阵, 是为打印组字快速。库中每字占 128 个字节, 包含内容亦不同:

CLIB32 含 256 个西文字符、1—15 区中文字符、16 区以后汉字, 同 CLIB24。32 点阵的各种字体共用一种字符, 它就仅存于 CLIB 32 的前部;

ALIB32、BLIB32、DLIB32 均只含有 16 区以后的汉字, 只是字体不同, 每字的信息各异。ALIB32 楷体库, BLIB32 仿宋体库, DLIB32 黑体库, CLIB32 宋体汉字的居所。

### 4. 48×48 点阵字库

48 点阵字库, 主要用于打印。有 CLIB 48、ALIB48、BLIB48、DLIB48、ASC48 等。

48 点阵字有两种字模: 48×48 宋体, 40×40 仿宋、楷、黑体。存放形式有 48×45 宋, 48×36 仿宋、楷、黑这两种形式。字符倾向于 48×36, 48 点阵共用一种字符, 存于 ASC48 中。

48 点阵字库均以列式存放汉字、字符点阵。它们存放的内容是这样的:

CLIB48, 存放 48 点阵宋体字, 每字 270

字节, 只含 16 区后汉字;

ALIB48、BLIB48、DLIB48 分别存放 48 点阵楷、仿、黑体字, 每字 216 字节, 只含 16 区以后汉字;

ASC48, 放 48 点阵西文字符和纯中文字符, 含 ASCII 代码 32—127 (96 个字符) 字符和 1—9 区纯中文字符, ASCII (西文) 字符占 108 字节, 纯中文字符占 216 个字节。

## 二、点阵分布、存储

汉字点阵分布是这样的, 构造汉字时, 以有点为 1, 无点为 0, 则 8 个点的 0.1 就构成 1 个字节, 是数的形式, 数在被存储时是作为字符的 ASCII 码 (0—255) 看待的。如有这么一字节为 01000001 (即 65, 或 41h), 虽然其值是 65, 但以 “A” 存起, 取时将以 “A” 取出, 如显示则显示 “A”, 实际上它的代码才是我们的点阵字节值, 故需还原其代码 (在 BASIC 中要还原用 ASC (x\$) 函数; 在汇编中则无需还原, 已是)。那么若干个点阵字节存放也如此, 只是字节多了, 根据需要还有区别。下面看看区别:

1. 16×16 点阵汉字, 行式存放, 所以每行占 2 字节, 16 行占 32 字节。存放格式如下:

|    |    |
|----|----|
| 1  | 2  |
| 3  | 4  |
| ⋮  | ⋮  |
| 31 | 32 |

存放顺序仍然是: 1, 2, 3, 4, ..., 31, 32。其实, 任何点阵字的存放顺序都是由 1 顺序到 n, 只是组字时参照存放格式生字。

2. 24×24 点阵汉字, 列式存储, 故每列点阵占 3 字节, 24 列共占 72 字节。存储格式如下:

|   |   |     |    |
|---|---|-----|----|
| 1 | 4 | ... | 70 |
| 2 | 5 | ... | 71 |
| 3 | 6 | ... | 72 |

3.  $32 \times 32$ 点阵汉字, 列式存储, 每列点阵占4字节, 32列共占 $4 \times 32 = 128$ 字节; 且是每一列的4个字节存储完后, 再存下一列的4个字节。存储格式如下:

|   |   |     |     |
|---|---|-----|-----|
| 1 | 5 | ... | 125 |
| 2 | 6 | ... | 126 |
| 3 | 7 | ... | 127 |
| 4 | 8 | ... | 128 |

而不是:

|    |    |     |     |
|----|----|-----|-----|
| 1  | 4  | ... | 94  |
| 2  | 5  | ... | 95  |
| 3  | 6  | ... | 96  |
| 97 | 98 | ... | 128 |

CLIB24、CLIB32的西文字符、中文字符占有和汉字一样大的点阵存储空间, 只是西文字符只有效地分布其点阵于左半部。

4.  $48 \times 48$ 点阵汉字, 列式存储, 有两种点阵字模:

(1)  $48 \times 36$ , 每列6个字节, 36列, 共 $36 \times 6 = 216$ 字节。

$48$ 点阵字均是被存完上24点(即上3字节)的所有列后才被存储下24点(即下3字节)的所有列的。故存储格式如下:

|     |     |     |     |
|-----|-----|-----|-----|
| 1   | 4   | ... | 106 |
| 2   | 5   | ... | 107 |
| 3   | 6   | ... | 108 |
| 109 | 112 | ... | 214 |
| 110 | 113 | ... | 215 |
| 111 | 114 | ... | 216 |

这里还需特别说明一下 ASC48 这一  $48 \times 48$ 点阵文件的点阵分布及存储情况。它也是  $48 \times 36$ 的点阵字库, 它包含两方面的内容: 一是1-9区纯中文字符; 二是ASCII代码32-127的西文字符。前者的存储同  $48 \times 36$ 点阵汉字, 而后者则不同。

ASCII西文字符以  $48 \times 18$ 点阵存储, 也是列式存放, 每列6字节, 18列共  $18 \times 6 = 108$ 字节; 且它是每列6字节存放完后才存放下一列6字节的。故存储格式如下:

|   |    |     |     |
|---|----|-----|-----|
| 1 | 7  | ... | 103 |
| 2 | 8  | ... | 104 |
| 3 | 9  | ... | 105 |
| 4 | 10 | ... | 106 |
| 5 | 11 | ... | 107 |
| 6 | 12 | ... | 108 |

(2)  $48 \times 45$ , 每列6字节, 45列, 共 $45 \times 6 = 270$ 字节。存储格式如下:

|     |     |     |     |
|-----|-----|-----|-----|
| 1   | 4   | ... | 133 |
| 2   | 5   | ... | 134 |
| 3   | 6   | ... | 135 |
| 136 | 139 | ... | 268 |
| 137 | 140 | ... | 269 |
| 138 | 141 | ... | 270 |

### 三、高点阵汉字显示

高点阵汉字显示, 一定要参照各库点阵的分布规律行事, 否则即使找到了点阵信息, 也很难显示成字。取点阵就参照以上一、二, 显示点阵又是这样实现的, 这和打印稍有不同, 打印是只管在当前打印位置打点即可, 而显示则是记住应该有点的位置的光标值( $x, y$ ), 然后在该点位( $x, y$ )打点(即画点)。

汉字显示, 以一字节为单位, 一字一字地操作, 每字占多少字节, 则一次取多少字节以进行分析, 而在屏幕上组字。

汉字、字符点的产生和位置控制情况如下:

1. 点的产生: 将一字节数8位(其大小为0-255)从最高位向最低位逐位求出(Basic用除以128( $2^7$ )、64( $2^6$ )、32( $2^5$ )、...、2( $2^1$ )、1( $2^0$ )而得; 汇编可用左移一位或“与”操作等而得), 如是1, 则记住当时的光标位置(Basic用两个数组元素, 汇编可用一对字单元), 这样点的情况就以位的情况记载下来。

2. 位置控制: 如是行式存储, 则分析点时, 横坐标递增(一般递增1, 如倍放, 则相应加倍值), 一行字节分析完, 则纵坐标递增, 横坐标回原位。如是一组字, 则后移一定位置后, 显示下字仍同前。说明: 横坐标的变化反映列的变化, 纵坐标的变化反映行的

变化。

如是列式存储,则分析点时,纵坐标递增,一列分析完,则横坐标递增单位值,纵坐标回原位,一组字的情况同上。

32点阵和48点阵的情况略微特殊。

(1) 32点阵打印需控制两行(对24针打印机而言),显示可同24点阵,因为其一列存完才存第二列,见二之点阵分布表。

(2) 48点阵完全不同,它的纵6字节是先存完所有列的上3字节,再以同样方式存储所有列的下3字节,因此便于打印,显示则需分两步控制:先控制汉字、字符一半字节的上显示,方法同24点阵;再控制汉字、字符另一半半字节的下显示,先使纵坐标+24,再同24点阵控制显示。

ASC48的西文字符仍同24、32点阵,是因其纵6字节(一列)存完才存下一列点阵字节的缘故。

#### 四、数字、字符显示例程序

汉字、字符点阵显示属于图形显示,为简洁,使用了高分辨机提供的GRD、SYS图形处理系统的功能(安装在CONFIG、SYS中),用Basic的PSET语句速度更快。当然用汇编的画点功能是最彻底的了,但道理一样。

本文程序用Basic编制,均在GW286B上通过,主要使用了GRD、SYS的D和M功能,D画点,M关闭现有显示(清屏),具体请参阅GW286B或GW0520C-H《DOS使用手册》。

本文程序在Basic软件中运行时,做一点声明:

由于Basic A规定随机文件的记录大小 $\leq 128$ ,故CCDISP4.bas、CCDISP5.bas、CCDISP6.bas不能在其中运行;

CGWBASIC规定随机文件的记录大小可 $\geq 255$ (甚至可以很大),故所有文件均可在其中运行。但如把CCDISP4.bas的220语句改为:

220 ClIb\$ = "CLIB48" : no = 270也

不能运行,因为字符型变量其内容长度超过了255,这是不允许的,故将CCDISP4.BAS改为了CCDISP5.BAS,实际是360语句和相应内容作了调整,这样CCDISP5.BAS就可以显示所有48点阵汉字了,不同的字体只需改一下220语句。

CLIB48, 每字270字节,宋体;

ALIB48, 每字216字节,楷体;

BLIB48, 每字216字节,仿宋体;

DLIB48, 每字216字节,黑体。

显示24点阵字库 CLIB24汉字及纯中文字符程序、西文字符的程序;显示16点阵字库CCLIB汉字程序,显示48点阵汉字库程序。显示48点阵字符库ASC48的纯中文字符的程序,显示48点阵字符库ASC48西文字符的程序清单略(需要可同作者联系)

#### 五、综合处理、显示简述

综合处理就是放大、缩小、背景显示、修饰显示等。这些无非是在原点阵信息基础上进行加工,产生新的字型信息和综合信息。如放大,无非就是原点阵信息乘倍数,然后填充空隙,而得到放大的字型信息;网格背景显示,无非就是两种信息——原点阵信息和网格信息组合在一起,而得到综合了的信息。显示方法等均可同前。

#### 六、本文的意义

本文旨在提供给计算机专业人员“所见即所得”技术的基础认识,大家共同进行新的开拓,促进我国办公自动化技术的发展;对我们的操作、使用人员则提供一个“所见即所得”的基础印象,帮助理解一些基础知识。

#### 参考文献

1 《长城05 20C—H DOS使用手册》, 中国计算机发展公司长城微型机开发公司, 1987年版。

2 《Great Wall 286B使用手册》。中国计算

# IBM—5550机与长城、IBM—PC、 286、386机之间连网通信中的实时汉码转换

贵州工学院控制技术研究所 谢晓尧

贵州省轻纺工业厅 江国章

**摘要** 本文介绍了异种机即IBM—5550机与长城机之间实现连网通信的关键问题，汉码的实时转换。首先形式描述了5550机和长城机的汉字内码，以及两内码之间的关系。然后以进程方式描述了如何用汇编实现实时转换过程。

## 一、引言

从一九八四年以来，我国陆续进口了大批的IBM—5550微型机。该机的特点是性能较强，汉字内码采用非GB2312—80码，属于我国非主流机种。而长城及其兼容机IBM—PC，汉字内码是采用GB2312—80码。结果是IBM—5550机与长城（包括IBM—PC和兼容的286、386机）并不兼容。使5550机的推广应用受到制约。近年来随着3<sup>+</sup>网，286，386机的逐渐普及，如何将5550机连入3<sup>+</sup>网，实现与长城机的中远程通信，显然是一个提高5550机应用率的强有力措施。鉴于我国汉字的特殊性，解决上述问题的关键是通信中的实时汉码的转换。

本文首先形式描述5550机和长城机的汉字内码，及两内码之间的关系，然后以进程方式描述了如何使用汇编实现实时转换过程。

## 二、IBM—5550和长城 机内码关系的形式化描述

要实现汉字内码的转换，首先必须找出两机种之间汉码的关系。长城机采用的GB2312—80码用两字节表示一汉字。高字节为汉字区码，低字节为汉字位码。为避免与ASCII码发生冲突，其两字节的高位恒置为“1”。IBM—5550机采用的汉字编码也是用两字节表示一汉字，但编码完全不一致，并且字节高位并非恒置为“1”。两种汉码关系的形式描述如下：

长城汉字内码：

If 5550内码低 > 9EH

then begin 长城位码 = 5550内码低 - 9EH;

长城区码 = (5550内码高 - 80H) × 2

end

else begin 长城区码 = (5550内码高 - 80H) × 2 - 1;

if 5550内码低 > 7EH then

长城位码 = 5550内码低 - 40H

else

长城位码 = 5550内码低 - 3FH

end;

长城内码高 = 长城区码 + A0H;

长城内码低 = 长城位码 + A0H;

机发展公司，1988.1版。

3《PS使用说明书》，中国长城计算机集团公司，1989.1。

4《PC—DOS 2.0系统汉化和CCDOS 2.0/2.10分析与使用》，李玉林编，1986.4等。



### 5550汉字内码:

长城区码=长城内码<sub>高</sub>-A0H;

长城位码=长城内码<sub>低</sub>-A0H;

If 长城区码为奇

then begin

5550内码<sub>高</sub>=(长城区码+1)/2+80H;

if 长城位码>3FH then

5550内码<sub>低</sub>=长城位码+40H

else

5550内码<sub>低</sub>=长城位码+3FH

end

else begin

5550内码<sub>高</sub>=(长城区码/2)+80H;

5550内码<sub>低</sub>=长城位码+9EH

end

### 三、IBM—5550和长城机

#### 汉码之间的实时转换

在5550机与长城机之间的连网通信中,汉码的转换过程被看成是一个进程,当某种条件满足时,进程被激活。由形式描述可见,通信中加入5550机中的汉码转换进程与长城机中的汉码转换进程是不一样的。一旦汉码转换进程被激活,被激活的进程从其输入流上得到通信对方传送来的汉字内码,对该内码进行加工处理,转换成适合本方机种的汉字内码,然后送到屏幕上实时显示。进程完成对汉字内码的转换后,就进入死亡状态。等待激活条件满足后,进程可再度被激活。

5550机中的汉码转换进程的激活条件由两部份组成:

长城内码<sub>高</sub>≥80H ∧ 长城内码<sub>低</sub>≥80H

长城机中的汉码转换进程的激活条件仅由一部份组成:

5550内码<sub>高</sub>≥80H

现用8086汇编语言描述转换进程所对应的静态程序如下:

1. 5550机中转换进程所对应的静态程

序:

设转换前Ax寄存器内是长城机汉字内码,转换后是5550机汉字内码。

```

sub ah, 0a0H      (16+6)
sub al, 0a0H      (16+6)
test al, 1        (5)
jz p1             (16)
add al, 1         (4)
shr al, 1         (2)
add al, 80H       (4)
cmp ah, 3fH       (4)
jnbe p2           (16)
add ah, 3fH       (4)
jmp ex            (15)
p2: add ah, 40H    (4)
    jmp ex        (15)
p1: Shr al, 1      (*2)
    add al, 80H    (4)
    add ah, 9eH    (4)
ex:

```

:

2. 长城机转换进程所对应的静态程序:

设转换前Ax寄存器内是5550机汉字内码,转换后是长城机汉字内码。

```

cmp ah,          9eH
jbe p1
sub al,          80H
shl al,          01H
sub ah,          9eH
jmp P2,
P1: sub al,      80H
    shl al,      01H
    sub al,      01H
    cmp al,      7eH
    jbe P3
    sub ah,      40H
    jmp P2
P3: sub ah,      3fh
P2: add al,      0a0H
    add ah,      0a0H

```

:

(下转第48页)

# IBM—PC/XT与MCS—51并行长线 通讯中的硬件接口和软件设计

湖北汽车工业学院 郑相周

**摘要** 本文就工业控制和检测方面最常用的系统机IBM—PC/XT和MCS—51单片机(8031)间的通讯,提出采用并行通讯的方法,并给出了并行长线通讯的硬件接口、简略的软件、及接口和软件中采取的各种抗干扰措施。实践证明,文中提供的原理和方法是切实可行的。

## 一、概述

分布式控制系统及各种实际工程问题而言,使用我国目前拥有量最多的IBM PC型计算机作为中心控制单元,而局部单机系统采用MCS—51系列单片机,无疑很有发展前景。由两者组成的分布式检控系统,不同程度地存在着相互通讯问题。实时测控系统,不仅要求通讯的速率,也对通讯的可靠性提出了较高的要求。

在实际的工业测控系统中,数字信号常常必须传送较长的距离,为满足实时的要求,长线通讯采用并行的方法是很值得考虑的。

本文就IBM-PC/XT和MCS-51(8031)间的长线通讯提出了并行接口和软件设计方法,并为提高通讯的可靠性采取了必要的措施。

## 二、并行通讯原理

一般地,双机间的并行通讯采用握手方式传送数据和状态。上级机和下级机分别设定为I(IBM)机和M(MCS-51)机。

I机有如下的信号线:

- (1) 8位数据/状态线: 输入;
- (2) 8位数据/状态线: 输出;
- (3) DR: 数据准备好, 输出; 高电平有效;

- (4) DA: 数据接收到, 输入; 高电

平有效;

- (5) SR: 状态准备好, 输出; 高电平有效;

- (6) SA: 状态接收到, 输入、高电平有效;

- (7) INTRQ: 中断请求, 输出; 边沿触发方式, 由高跳变为低电平;

- (8) IRQ<sub>3</sub>: 中断请求接收, 输入; 由低到高边沿触发。

在上述8类信号线中,前两类是数据或状态的输入/输出线,DR、DA为数据传送握手信号,SR、SA为状态传送握手信号。INTRQ为中断请求信号,可使用M机的中断功能来传送数据或状态;而IRQ<sub>3</sub>为中断请求接收线,用来在M机向I机申请中断,进行数据或状态的传送。

M机也具有与I机相同种类和数量的信号线,其中IRQ<sub>3</sub>在M机(MCS-51)中为INT0(或INT1)。需注意的是,I机和M机上相对应的信号其输入输出性质相反。

图1所示为由这些信号组成的通讯系统原理图。

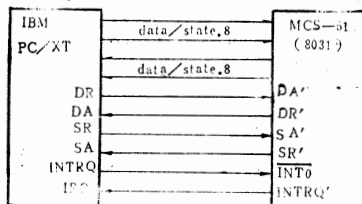


图1 通讯线路原理图

由图可知,在实际通讯中所用到的信号

线数达22个。这无疑增加了系统的干扰因素,在实际应用中,可采用一定措施把单向通讯变为双向通讯,可使信号线数至少减少8根。

数据和状态的传送,除具有正确的通讯线路外,还必须依一定的时序才能正常进行。下面以I机向M机传送数据(或状态)来说明传送数据或状态的时序。

在由I机向M机发送数据(或状态)时,先由I机将数据(状态)置于信号线上,如果采用中断方式传送,则由I机向M机发INTRQ,按约定方式传送,则不发INTRQ。当数据(状态)在信号线上稳定后,I机发DR有效;按照约定或中断,M机检查其DA'是否有效,有效,则读入数据并向I机发DR'有效(即DA有效);I机发DR后,即检查DA是否有效,有效,则发DR无效;M机发送DR'有效后,即检查DA'是否无效,无效,则发送DR'无效;I机在确认DA无效后,才能进一步发送下一个数据。至此,I机向M机完成一个发送周期T。

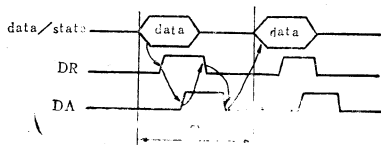


图2 I机发送数据时序图

由于两机所用系统频率不同,指令运行速度有差异,因此在每次发送data/state前,都要确认自己的DA是否处于无效状态,当处于无效状态时,才可发送下一个数据/状态;否则,在连续传送data/state时,会因运行速度的差异而导致通讯进入僵死状态。

由图2可知,若不考虑信号线的延滞,并行通行的最小周期取决于各机执行指令的时间。对于由IBM PC/XT和8031组成的通讯系统,并行通讯的周期约为120μs左右。

### 三、并行通讯中的接口及抗干扰措施

IBM-PC/XT和8031单片机间的通讯接口应满足以下要求:

(1) 接口应满足并行通讯的基本要求,即应具有分别为3位的并行输入输出,用于握手联络的DR、SR、INTRQ输出和DA、SA输入,其中IRQ<sub>3</sub>可使用IBM-PC/XT中的中断请求IRQ<sub>3</sub>,而INT<sub>0</sub>即为8031中的INT<sub>0</sub>。

(2) 接口应满足CPU控制的要求

各接口的CPU应能对I/O操作的各通道进行寻址和控制信号的输入输出;输出的信号能锁存;

(3) 满足抗干扰的要求

为使并行通讯正确地进行,接口应采取必要的措施,以提高系统的抗干扰性能。

显然,在满足并行通讯接口的前两个基本要求中,并行接口芯片INTEL 8255A-PPI是很好的可选元件。本文的接口即由8255A-PPI构成。

IBM-PC/XT的并行通讯接口如图3示。

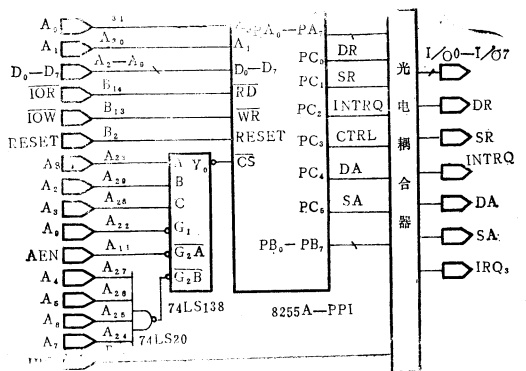


图3 IBM-PC/XT并行通讯接口原理

用IBM-PC/XT内部插槽中的信号对8255A进行译码,其四个端口的译码从地址02F0H开始到02F3H。8255A的A端口用于data/state输入,B端口用于data/state输出,C端口PC0、PC1、PC2分别为DR、SR和INTRQ,PC3用控制将单向传输线合并为双向传输线,PC4、PC5分别用作DA、SA。所有的信号(除CTRL外)都经过光电耦合器引出。图中左侧线上的标号(如

A31) 为IBM—PC/XT插槽的信号序号, 左侧标号为所用的信号名称。

8031单片机的通讯接口如图4示。

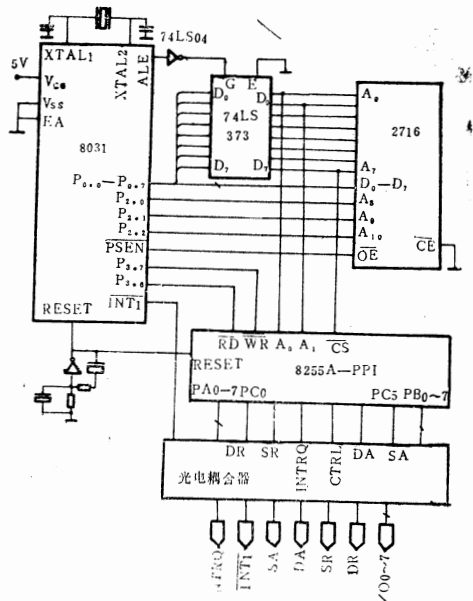


图4 8031通讯接口原理图

8255A的用法与IBM—PC/XT的接口中8255的端口用法相同。

以上的通讯接口可以满足通讯和CPU控制的基本要求。

为了增强双机间通讯的抗干扰性能, 系统采取了一系列抗干扰措施。

### 1. 采用光电隔离并转换为电流环方式

在不带调制解调装置的串行通讯中, 采用电平方式很易受到外界的干扰。

在双机并行通讯接口采用光耦隔离方法增强通讯的抗干扰性能, 同时将驱动电平信号转换为十几mA的电流信号, 在接收端也通过光耦再将电流信号转换为具有TTL电平的电压信号送给接口。如图5所示。

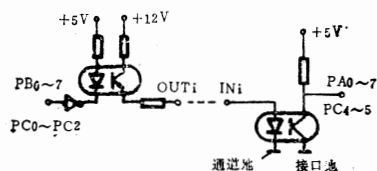


图5 光电隔离与电流环

在图5所示电路中, 通讯通道专用一通道电源(+12V), 且通道地与接口地不相连

通, 通过光耦器件隔离, 可提高抗干扰作用。

在实际应用中, 采用扁平电缆传递信号, 通讯距离高达30m以上。

## 2. DATA/STATE信号线的并行双向通讯

在数据通讯中(如串行通讯), 有单工、半双工、全双工方式, 在实际应用中, 大多数只工作于半双工方式。从信号线串扰的角度看, 通讯中所使用的信号线数量越多, 通讯中的干扰因素也就越多。为解决这一问题, 接口的光电耦合器采用如图6所示的电路把16根的单向通讯线转变成为8根的双向通讯线。

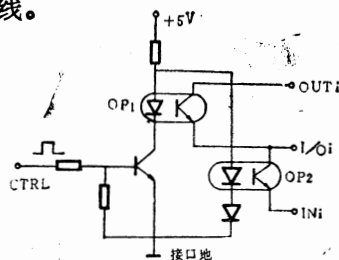


图6 开关电路

接口使用光电耦合器构成的开关电路将图5中同端(或是IBM—PC/XT或是8031接口)的(INi, OUTi)连在此电路, 由PC3施加控制脉冲, 即可实现双向通讯。

如图6示, 在未加控制脉冲时, 晶体三极管截止, OP1不工作, OUTi和I/Oi呈常开; OP2工作, I/Oi和INi呈常闭; 当外来正向控制脉冲时, 晶体管导通, 由于一般晶体管饱和压降小于二极管的 $V_D$ , 当OP1导通时, OP2即截止。

显然, 使单向的16根通讯线变为8根的双向通讯线, 是以硬件成本的增加和通讯速率的减慢为代价的。由于给PC3加载将使通讯周期增加。

### 3. 其它措施

除上述采取的两条主要措施外, 还采取了其它一些措施。如在实际布置通讯线时, 地和信号线间隔排列, 有助于提高抗干扰性能。

## 四、软件设计及抗干扰措施

在编制通讯软件时,首先要通过软件设置8255A的工作方式,然后再按照约定或中断的不同方式编制通讯软件或中断服务程序,两机间的通讯严格按前述的时序进行。

### 1.8255A—PPI的设置

8255A—PPI在两个接口的用法相同。将8255A设置为工作方式0,其中A通道,B通道分别用于输入和输出,C通道中,PC0~PC3用于输出,PC4、PC5用作输入,其各自的用途见于两接口原理图中。

下面分别为8031和IBM—PC/XT对8255A的设置程序清单。

#### IBM—PC/XT设置8255A程序

```
MODE EQU 98H
PORTA EQU 2F0H
PORTB EQU 2F1H
PORTC EQU 2F2H
CNTRL EQU 2F3H
MOV AL, MODE
MOV DX, CNTRL
OUT DX, AL
MOV DX, PORTC
MOV AL, 00110011B
OUT DX, AL
```

#### 8031设置8255A程序

```
MODE EQU 98H
PORTA EQU 7CH
PORTB EQU 7DH
PORTC EQU 7EH
CNTRL EQU 7FH
MOV A, MODE
MOV @R0, CNTRL
MOVX @R0, A
MOV @R0, PORTC
MOV A, #00111011B
MOVX @R0, A
```

### 2.双机通讯程序设计

双机间的通讯应严格按照图2所示的时序编写,下面以8031向IBM—PC/XT发送

数据为例,绘出两者通讯的子程序清单。这两个子程序可供其它需要通讯的程序调用。

#### 8031向IBM—PC/XT发送数据(数据在A中)程序

```
OUTDATA: MOV @R1, A, 保存数据
MOV @R0, PORTC
MOVX A, @R0
ORL A, #00001000B
MOVX @R0, A; CTRL=1, OUT
DETDA0: MOVX A, @R0
ORL A, #11111110B;
DA' = 0;
CPL A
JB0 DETDA0
MOV A, @R1
MOV @R0, PORTB
MOVX @R0, A; OUT DATA
MOV @R0, PORTC
MOVX A, @R0
ANI A, #11111110B
MOVX @R0, A; DR' = 1
DETDA1: MOVX A, @R0
ORL A, #11111110B;
DA' = 1;
JB0 DETDA1
ORL A, #00000001B
MOVX @R0, A; DR' = 0
RETR
```

#### IBM—PC/XT接收8031数据程序

```
INDATA PROC NEAR
MOV DX, PORTC
IN AL, DX
AND AL, 11110111B
OUT DX, AL; CTRL=0, IN
DETDR1: IN AL, DX
TEST AL, 00010000B
DR' = 1;
JNZ DETDR1
MOV DX, PORTA
IN AL, DX; IN DATA
MOV DATA, AL
MOV DX, PORTC
```



```

IN    AL, DX
QR    AL, 00000001B
OUT   DX, AL ; DA'=1
DETDR0: IN    AL, DX
TEST  AL, 00010000B
      ; DR'=0?
JZ     DETDR0
AND    AL, 111'1110B
OUT    DX, AL ; DA'=0
RET
INDATA ENDP

```

从这两个子程序可计算出8031向IBM—PC/XT发送数据所占用的时间，即通讯周期。其最小周期约为120 $\mu$ s（8031的osc为6MHZ，IBM—PC/XT的CLK为4.77MHZ）。

### 3. 按中断方式进行通讯的软件设计

双机间进行通讯时，可以根据不同的需要选用查询方式和中断方式。

查询方式比较简单，通讯中可达到最快的速率，且只要在主程序调用通讯子程序即可；但查询方式要占用较多的主机时间，不利于解决其它问题。

中断方式在需要通讯时，向对方发中断请求，对方在中断服务程序内进行通讯。这种方式可较少地占用主机时间，但由于中断服务程序的引用，将使通讯速率略有降低。

在IBM—PC/XT使用中断时，系统用IRQ<sub>3</sub>信号线。程序预置中，先改写0000：002CH矢量表中IRQ<sub>3</sub>的中断矢量值，然后才能使中断进入到服务程序中去。改写机内中断矢量表的程序见下，INTSI为中断服务程序的程序名。中断服务程序的格式如下，其中的执行语句不可缺少。

#### IBM—PC/XT改写中断矢量程序

```

PUSH    DS
MOV     AX, 0
MOV     DS, AX
MOV     BX, 2CH
MOV     AX, SEG INTSI
      ; INTSI 段值
MOV     [BX+2], AX

```

```

MOV     AX, OFFSET INTSI
      ; INTSI 首址
MOV     [BX], AX
ASSUME DS: DATAREA
POP     DS

```

#### IBM—PC/XT中断服务程序

```

INTSI
INTSI   PROC   NEAR
CLI
PUSH    AX
PUSH    BX
.....
( 中断服务程序内容 )
.....
MOV     AL, 20H
MOV     20H, AL
POP     BX
POP     AX
STI
IRET
INTSI   ENDP

```

8031使用中断方式时，首先在主程序内进行中断预置；由于中断信号采用8031片内的INT1，故应在固定内存0013H处填写转移到中断服务程序的跳转语句。相应的程序清单见下。

#### 8031 中断预置

```

SETB    EA
SETB    IT1
SETB    EX1

```

#### 8031 中断服务程序INTSM

```

ORG     0013H
AJMP    INTSM
.....

```

```

INTSM:  PUSH    A
        PUSH    PSW
        .....
        ( 中断服务程序内容 )
        .....
        POP     PSW
        POP     A
        RETI

```

### 4. 软件中的抗干扰措施

# Turbo prolog和Turbo c的接口技术

陕西师范大学 张川民

**摘要** 本文介绍TURBO PROLOG语言和TURBO C语言之间的接口技术。文章在分析两种语言各自所需不同运行环境的基础上,解决了它们之间在内存分配上的不一致性等问题,完满地实现了予期目标。使得为人工智能领域的研究提供了有利条件,也为TURBO C的应用提供了人工智能能力。

Turbo prolog是一种基于逻辑程序设计的语言,而Turbo c则以其简洁、灵活、高效、可移植性好而成为现代最流行的过程式通用程序设计语言。前者,尤其适合于开发专家系统、动态数据库、自然语言处理程序等,而后者,则被广泛地用于系统程序和应用程序的开发。它们各有自己独特的长处。若能将用这两种语言编写的程序连接起来,则为人工智能领域的研究提供了有利条件,也为turbo c的应用提供了人工智能能力。这样一来,当需要解决比较复杂的问题时,只要给出问题的描述形式,然后让turbo Prolog语言去进行推理。因此 turbo prolog和turbo c的连接,不仅可以使得程序代

码清晰度提高,程序的灵活性增强。而且还可以极大地缩短软件开发周期。

## 一、实现时注意事项

1.turbo prolog程序可以调用turbo c编写的函数,就象调用它的内部谓词(predictes)一样,但是目前turbo c程序还不能调用turbo prolog程序模块。

2.所有调用的turbo c库函数都必须加上下划线作前缀,用户自定义的函数则不需要加下划线前缀。

3.在连接turbo prolog和turbo c时,为了保证软件设计的一致性,与屏幕有关的输

=====

为使通讯更加准确地进行,软件也采取一定的抗干扰措施。

### (1) 自检

开机后,双方将约定的数据分别按约定方式和中断方式传送,以判定通讯线路的故障状态,并由IBM—PC/XT加以输出。

### (2) 跳出“假中断”

由于干扰,很可能使得程序因“假中断”请求而进入中断服务程序,在中断程序中根据握手信号的状态设置开关,根据开关状态可控制跳出假中断。

### (3) 用纠错码降低误码率,

在成批传送数据时,每批数据后发该批数据和的纠错码,接收方根据纠错码的状态确定是否重发。

### (4) 设置“陷阱”

现场的测控系统常因外界的干扰发生程序“飞掉”的现象。在单片机的程序存储器中,子程序之间的卡用单元和主程序的某些单元填入跳转指令,以使程序飞跳后可回到程序,从而减少部分干扰的影响。单字节的NOP指令和跳转指令结合应用效果更好。

## 五、结论

在IBM—PC/XT和386之间,采用上述的设计以及相应的抗干扰措施,不仅可使系统能实现并行双向通讯,提高系统的抗干扰性能,且可以满足一定的实时性要求。经实际应用,采用扁平电缆,传送距离可达30m,传送速率约为8300byte/每秒。

出操作如printf,putc等不再具有原来的功能。如果必须在turbo c程序中向屏幕输出。则可引用prolog.lib库中的wrch函数和zwf函数(wrch函数可以在turbo prolog窗口内写一字符,zwf函数与turbo prolog语言中的writef标准谓词其有相同的功能)。

4.通过turbo prolog调用的turbo c函数不能有返回值,并且它们应当被定义为void类型。参量的流程模式应根据turbo prolog全局谓词说明而确定。

## 二、实现接口技术

### 应具有的文件

1.应有将turbo c源程序编译成目标程序所必须的turbo c系统中的基本文件,如TC.EXE或者TCC.EXE和TURBOC.CFG文件以及必须的库,如CL.LIB、EMU.LIB、MATH.LIB等。

2.应具有将TURBO PROLOG源程序编译成目标程序所必须的TURBO PROLOG系统中的基本文件。如PROLOG.EXE以及必须的库,如PROLOG.LIB等。

3.系统连接程序,如TC系统中的TLINK.EXE程序或DOS操作系统中的LINK.EXE连接程序。

### 三、连接要点

#### 1.对程序模块进行编译

(1)在TC源程序中不能使用main()函数,它必须由TURBO PROLOG源程序中的主模块(包含一个目标(GOAL)的模块)所替代。

(2)被TURBO PROLOG调用的TURBO C函数必须加上\_n作为函数名的后缀(其中n为0,1,2,...等,根据TURBO PROLOG源程序中该全局谓词的模式流的个数确定其数值)。

(3)TURBO PROLOG主模块中必须

包含TURBO C函数,并且这些函数应描述为全局谓词。

(4)为了保证内存分配的一致性,所有的TURBO C源程序模块都必须在大型内存模式下编译,这是TURBO PROLOG系统在编译时使用的唯一内存模式。

(5)如果程序中调用TURBO PROLOG版本号为1.1的库,那么在编译TURBO C源程序模块时必须将寄存器分配选项关闭(-r-)。

(6)编译TURBO C源程序模块时,必须将产生下划线选择项开关关闭(-u-)。

#### 2.对程序模块进行连接:

(1)INIT.OBJ必须作为连接时的第一个目标文件(它是TURBO PROLOG的初始化模块,是TURBO PROLOG系统上的一个目标文件)。

(2)如果程序中用到TURBO C库函数,连接时应加上CL.LIB库。

(3)如果程序中有实数运算,那么还应当连接EMU.LIB和MATNL.LIB库。

#### 3.对程序模块进行连接的命令形式:

可以用下述连接命令形式之一进行连接装配工作:

(1)TLINK INIT <T-P-PROGRAM>  
<T-C-PROGRAM><T-P-PROGRAM.SYM>,  
[EXEFILE], [YOUR-LIBS], PROLOG  
[+EMU][+MATHL]+CL

(2)LINK INIT <T-P-PROGRAM>  
<T-C-PROGRAM><T-P-PROGRAM.SYM>  
[EXEFILE], [YOUR-LIBS], PROLOG  
[+EMU][+MATHL]+CL

(以上各命令必须在单一的命令行上执行)。

## 四、完成接口技术的步骤

1.将TURBO PROLOG源程序编译成\*.OBJ文件及\*.SYM文件的方法:

在DOS下输入PROLOG即进入TUR-

BO PROLOG集成环境, 键入F3将 PROLOG源程序 (PANDC.PRO) 调入内存之后, 按Shift\_F9则可产生(PANDC.OBJ)目标文件及 (PANDC.SYM) 符号文件 (这是连接时所必须的两个文件)。

2. 将TURBO C源程序编译成.OBJ 文件的方法有两种, 根据情况可选用其中任一种来完成:

(1) 采用 TURBO C 集成环境: 在DOS下输入 TC 即进入 TURBO C 集成环境。这时, 根据连接要点中对程序模块进行编译工作的要求, 应对TURBO C 系统进行编译选择项的设置。

按Alt\_O (字母) 从TURBO C菜单上选择满足以下要求的选择项:

Options/Compiler/Model ( Large )

Options/Compiler/Code generation/

Generate Underbars ( off )

Options/Compiler/Optimization/Jump

Optimization ( off )

Options/Compiler/Optimization/Use register variables ( off )

完成上述设置工作后, 进入Options/Store菜单, 将这些设置内容保存在turboc.cfg配置文件中, 供以后使用。(若选择项无需改变时, 则该设置可一直延用下去)。

设置工作完成后, 键入F3将TURBO C源程序 (FACT.C或CSUM.C)调入内存, 然后按Alt\_F9则可产生对应的目标文件。

(2) 采用命令行操作环境

在DOS下输入:

TCC -ml -o -c -r -u- C语言源程序名

则可产生对应的目标文件。

3. 将各独立模块连接成可执行文件

选用“对程序模块进行连接的命令形式”中任一种完成连接工作。如:

Tlink init pandc fact csum pandc.

sym,pandc.exe, ,prolog+emu+math1

+ cl

#### 4. 运行

在DOS下输入: PANDC即可运行。

### 五、程序示例

现以求一整数之阶乘和另一实数相加之和的程序说明设计方法。

PROLOG源程序如下:

PROGRAM PROLOG

```
1. /* PANDC.PRO */
2. global predicates
3. factorial(integer, real)-(i, o)
   language c
4. sum(real, real, real)-(i, i, o)
   language c
5. goal
6. makewindow(1, 40, 31, "The
   result window of Turbo c", 0,
   0, 5, 80),
7. makewindow(2, 37, 28, "The
   window of the turbo prolog pro-
   gram", 5, 0, 5, 80),
8. write("Enter an integer: "),
   readint(Int),
9. write("Enter a real: "),
   readreal(Real),
10. shiftwindow(1),
11. factorial(Int, Result),
12. sum(Result, Real, Sum),
13. shiftwindow(2),
14. write("The result: ", Result,
   "+", Real, "=", Sum).
```

第三行: 说明factorial是一个由turbo c语言编写的仅有一种流程模式的全局谓词。其中, 第一个参数为输入值、第二个参数为返回值。

第四行: 说明sum是一个由turbo c语言编写的仅有一种流程模式的全局谓词。其中, 前二个参数为输入值, 第三个参数为返回值。

第十一、十二两行: 分别调用由factorial和sum两函数。

# 彩色显示适配器故障的分析与检测

军械工程学院 李笑梅 陈致明

**摘要** 彩色显示适配器的故障是微型机的常见故障之一,本文分析了彩色显示适配器的各种常见故障现象,阐述了检测手段,并给出了排除故障的技术途径。

彩色显示适配器是一个相对独立的I/O部件,微型机系统的重要组成部分。大量的工作实践告诉人们,彩色字符/图形显示适配器的故障是微型机的常见故障之一。许多故障现象,如开机无显示,主机出现等待“死锁”,微机工作不正常等,均与此有关。

此类故障现象的发生,既有工艺方面的原因,更有元器件的质量原因。特别是集成

电路芯片,在早期失效之后,即进入了稳定期,在稳定期,器件的失效较小,但随机器使用年代的增加,或由于使用环境条件的恶化,将促使集成电路芯片的失效率成指数上升,这将导致各种故障现象的出现。

本文通过理论分析,并结合自己多年的工作实践,对彩色显示适配器的一些故障进行分析、判断和检测。

第十四行:在TURBO PROLOG窗口打印总体运行结果后返回操作系统。

C语言源程序如下:

PROGRAM C

```
1. /* FACT.C 计算阶乘 */
2. void factorial_0(int top, double
   *result)
3. {
4.   double x;
5.   int i;
6.   zwf("The factor of %d are: ",
   top);
7.   if(top < 1) *result = 0.0;
8.   else for(x = 2.0, *result = 1.0;
   top > 1; top--, x = x + 1.0)
9.     *result = *result * x
10.  zwf("%f\n", *result);
11. }
```

```
1. /* CSUM.C 计算两输入数之和 */
2. extern void zwf(char *format,
   ...);
3. void sum_0(double parm1, double
   parm2, double *res_p)
4. {
5.   *res_p = parm1 + parm2;
```

42

```
6. zwf("The factor %f and the real
   %f are %f\n", parm1, parm2,
   *res_p);
7. }
```

连接命令:

```
tlink init pandc fact csum pandc.
sym, pandc.exe, ,prolog+emu+mathl
+cl
```

运行结果:

运行时,程序在TURBO PROLOG窗口要求使用者输入一整数和一实数,待用户分别输入之后,程序则在TURBO C窗口显示由TURBO C函数计算的阶乘的值和该阶乘与上述实数相加的结果。最后,再返回TURBO PROLOG窗口显示最终运算结果。

(本文中所提到的软件版本,Turbo prolog为2.0, Turbo c为V1.5)

## 参考文献

- 1 陈兆乾等编译, TURBO ROLOGP 程序设计, 南京大学出版社, 1989.
- 2 Khin Maung Yin, Using Turbo P-  
rolog, Que Corporation, USA, 1987.
- 3 张川民, 微电子学与计算机, 第8卷第1期,  
47页, 1991.



## 一、主机等待“死锁”故障

这类故障多出现在开机的时候,其现象往往表现为,主机开机时,屏幕无任何显示,出现了“死锁”等待。

微型机正常开机后,时钟发生器8284A产生复位信号,CPU8088初始化,即进入第一个取指周期,读取ROM BIOS的自检入口地址,读出转移指令JMP F000:E05B,对机器开始自检。若自检正确才能调入操作系统,显示DOS提示符。如果不能形成上述转移指令,机器就不能转入自检,此时将出现机器的“死锁”现象。

IBM-PC/XT的CPU是INTEL8088,其基本总线周期由T<sub>1</sub>、T<sub>2</sub>、T<sub>3</sub>、T<sub>4</sub>四个T状态组成。对于低速的存储器 and 外设可在T<sub>3</sub>、T<sub>4</sub>之间插入T<sub>w</sub>等待状态。

在T<sub>1</sub>状态下时,CPU8088传送20位地址和状态信号到系统总线上,总线控制器8288发出相应的控制命令;在T<sub>2</sub>状态时,数据总线与CPU8088接通,8088开始执行数据传递操作;在T<sub>3</sub>状态的前沿(即时钟的下降沿)采样I/O READY线。如果I/O READY信号为高电平,则把数据总线上的数据读入CPU或写到指定地址单元;若I/O READY信号为低电平,就在T<sub>3</sub>之后插入一个T<sub>w</sub>等待状态。以后在每个T<sub>w</sub>状态的前沿,再对I/O READY信号采样,只要I/O READY信号为低电平,继续插入T<sub>w</sub>等待状态,直至采样到I/O READY信号为高电平为止,才结束等待状态,在这个T<sub>w</sub>周期之后进入T<sub>4</sub>状态。T<sub>4</sub>状态结束,8088就完成了数据传送,结束了总线周期,恢复各信号的初始状态,并准备执行下一个机器周期。

根据工作过程,对上述的故障现象,可判断为I/O READY信号恒为低电平所至。此时测量地址总线A<sub>19</sub>-A<sub>0</sub>=FFFF0(H)和XA<sub>19</sub>-XA<sub>0</sub>=FFFF0(H)、数据总线D<sub>7</sub>-D<sub>0</sub>=EA(H)、XD<sub>7</sub>-XD<sub>0</sub>=EA(H)、

局部总线的数据AD<sub>7</sub>-AD<sub>0</sub>=EA(H)、控制信号MEMR和XMEMR均为低电平,与前边分析相符。这将造成在取指周期只读出了转移指令代码EA(H),没有读出转移地址,所以开机后,没能执行自检。在T<sub>3</sub>状态8088采样时,发现I/O READY信号恒为低电平,插入等待状态,由于I/O READY信号恒为低电平,8088总处于等待状态,即出现了主机“死锁”的现象。

下面结合图1,分析I/O READY信号恒为低电平的产生原因。

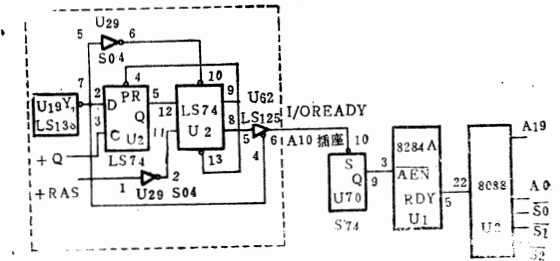


图1 彩色适配器产生I/O READY信号及与系统板联线图

图1虚线括起来的部分在彩色适配器板上,其余部分在系统板上。从图1中可知,CPU MEMSEL是经U19地址译码输出的信号,它是允许CPU访问彩卡上的VRAM的信号。当CPUMEMSEL为低电平时,使触发器U2—8输出高电平,经三态门U6—6输出的I/O READY信号亦为高电平,即I/O READY为高电平,可进入T<sub>4</sub>状态,允许8088去访问I/O设备;若图1中译码器、触发器、倒相器、三态门中的任一芯片逻辑功能错,都将造成I/O READY信号恒为低电平。当然,当A<sub>10</sub>与B<sub>10</sub>插座短路时,由于B<sub>10</sub>插座是接地端也会造成I/O READY信号恒为低电平。无论哪种原因引起I/O READY信号为低电平,都会使系统板上的等待触发器U70置“1”,以致使READY信号为低电平(表示没准备好),并插入T<sub>w</sub>等待状态,使机器处于“死锁”等待。

## 二、字符显示错误故障

由字符显示的基本原理可知,凡需要显

示的字都必须存入显示缓存器 VRAM 中。存储器的每一个存储单元，对应屏幕上的一个字符位置，屏幕上显示的内容，取决于存储单元中存放的数据，即待显示内容的 ASCII 码值。当扫描到屏幕上某一个字符的位置时，VRAM 中对应的存储单元的 ASCII 码就被读出，并送到字符发生器，从字符发生器中选出要显示的字符点阵，经控制转换电路后，就能在显示器上显示出要显示的字符。

出现字符显示错误故障，表明在开机自检时，显示适配器上的 VRAM 或数据输出逻辑部分出错误，自检程序转去执行 F000：F9D8 入口的子程序，使喇叭鸣一长两短，通知用户机器有故障。

### 1. 开机后满屏幕显示“、”的故障

由系统 ROM BIOS 原理可知，开机自检时，需要初始化屏幕显示方式。在初始化时把 16KB VRAM 的所有偶数地址单元，填入 20H（空格符），奇数地址单元填入 07H，即黑底白字的属性代码。所以在正常情况下，刚开机的瞬时，屏幕应呈全黑色，而开机满屏显示“、”，显然是不正常的。

此类故障现象可以采取多种办法确定故障点，现举例说明。

#### （1）调用诊断程序确定故障点：

在诊断程序中，选择彩色字符/图形显示器子程序，并在该子程序中选择彩色图形显示，此时屏幕全为黑底并显示 BLACK 五个字母，当出现此类故障时，黑底就显示“、”符号，但 BLACK 字母不产生变化。我们知道黑底为 ASCII 码 20H，“、”符号为 ASCII 码 60H，而 BLACK 五个字母分别为 42H、4CH、41H、43H、4BH，其 ASCII 码值如下：

“ ” = 00100000 = 20H

“、” = 01100000 = 60H

B = 01000010 = 42H

L = 01001100 = 4CH

A = 01000001 = 41H

C = 01000011 = 43H

K = 01001011 = 4BH

从上可见，20H 变为 60H，显然是 D6 位置“1”所致，而 BLACK 五个字母的 D6 位原来就为“1”，所以其 ASCII 码值没变。

上述分析，可初步确定与 D6 位相对应的 VRAM U51 芯片有故障，为进一步确定故障位，将 U51 输出的负载断开参见图 2。

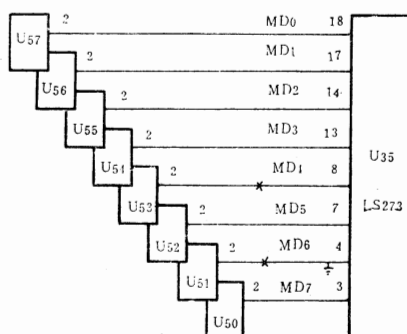


图2 彩色字符/图形显示适配器 VRAM 部分电路图

把 U51—2 脚与印制线刻断，即 U35—4 与 U51—2 开路，此时把 U35—4 接地。重新开机，屏幕全为黑色，即输出 ASCII 码值为 20H，因此可确认 U51 存储芯片冒“1”。另外也可采用交替负载的办法，即把 U53—2 脚印制线刻开（打×处），由 U51—2 飞线与 U35—3 连接，U53—2 飞线与 U35—4 连接（此时去掉 U35—4 接地线），再开机底色为“0”字符，即 ASCII 码 30H。显然故障现象随着 U51—2 连到 U35 的不同引脚，屏幕上的显示也随着变化。根据上述分析和判断确定了 U51 存储芯片为故障芯片。

#### （2）调用 DEBUG 确定故障点：

彩色字符/图形显示缓存器 VRAM 是由 8 片动态 RAM 组成，组件号 U50—U57。这 16KB 容量驻留在 CPU 地址空间的入口地址为 B800H，为了检查是否为 VRAM 的故障，可以使用 DEBUG 调试程序。首先使用 R 命令把有关寄存器的初始值设置如下：  
ES = 0B800H，DS = 0B800H，CX = 2048H，IP = E49DH，CS = F000H。这是调用 BIOS 中的子程序 STGTST—CNT 所需要的

入口条件。其中ES、DS为要测试的存储器段地址，CX等于要测试存储器块的字数，IP、CS的内容是调这个存储器测试程序的入口地址（IP：偏移量，CS：段地址）。

再用debug的G命令从F000：E49D地址开始执行BIOS中的子程序。程序执行结果应在屏幕上显示如下：

×××× ××201

其中×表示数字，中间两位××指出所损坏的存储器芯片的位置，它的对应关系如表1。

表1 VRAM组件与出错代码对应关系

| 出错代码 | 01  | 02  | 04  | 08  | 10  | 20  | 40  | 80  |
|------|-----|-----|-----|-----|-----|-----|-----|-----|
| 组件号  | U57 | U56 | U55 | U54 | U53 | U52 | U51 | U50 |

另外对VRAM的故障查找也可用debug的填充命令F。

A>debug

—F B800：00000 3FFF FF

—D B800：0000

通过显示的内容与事先写入的内容的一致性比较，即可确定故障芯片的位置。

## 2. 显示数字出错的故障

此类故障多为数据传输通路上出现的故障所致。由于总线传输数据不对，开机后喇叭鸣一长两短，同时屏幕显示的数字出错误。显示适配器板的内部输入/输出数据通路如图3所示。

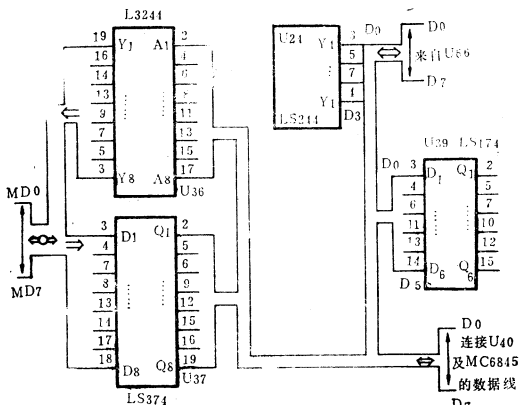


图3 输入/输出数据通路

该板的内部数据总线的输入/输出是经

过收发器U66（74LS245）连到图3的内部数据总线D7—D0上的。同这个内部数据总线直接相连有状态驱动器U24的输入端，彩色选择寄存器U39的输入端和连接方式控制寄存器的U40的输入端，以及MC6845的输入/输出数据线。当CPU8088向VRAM写入数据时，内部数据总线通过U36同VRAM的数据端MD0—MD7相连的，而当CPU8088从VRAM中读出数据时，读出的数据先锁存在U37，然后再送到内部数据总线。

从图3可知，若上述电路中，任一组件，任一位数据传输通路有误，都会造成屏幕显示的字符错。

例如，从键盘输入奇数（1，3，5，7，9）而屏幕显示为偶数（0，2，4，6，8）的故障。

从分析可知，“0”的ASCII码值为30H，而“1”的ASCII码值为31H。显然，由于D0位丢“1”使得31H变为30H，就出现了“1”变为“0”的故障现象。其余类推。根据数据传送输出的逻辑电路，采用适当办法，即可排除此类故障。

## 三、控制电路故障

在开机后由于控制电路出现的故障现象是多样的，如开机后喇叭鸣一长两短、无显示、字符彩色和底色出错，字符不同步及满帧有规律的图形等。由于控制电路较复杂，而故障现象又多种多样，因此，只能具体问题具体分析。

### 1. 不能显示图形的故障

出现此类故障时，开机后喇叭鸣一长二短。对于这类故障的分析，应从显示方式的控制部分入手。显示方式与选择信号的对应关系于表2。

确定各种显示方式的部分控制电路于图4。从表2和图4可见，方式寄存器的各位状态，直接决定各种显示方式。

方式寄存器第C位，即U40—2输出高电平时，产生HRES高电平有效信号，经一系

列控制产生14MHz的点时钟DOTCLK。反之，产生7MHz的点时钟信号，由该位输出高低电平来决定显示80字符，还是40字符。输出高电平为80字符，反之为40字符。

方式寄存器第1位，即U40—5输出高电平时，产生GRPH信号为真，表示是图形显示方式，该信号经U23—11输出MUXB为高电平。当选择信号MUXB信号为高电平时，是图形显示方式，反之为字符显示方式。

**表2** 显示方式与选择信号的对应关系

| 显示方式          | 方式寄存<br>器 状 态 | 两个选择信号 |      |
|---------------|---------------|--------|------|
|               |               | MUXA   | MUXB |
| 40×25黑白字符显示   | 101100        | L      | L    |
| 40×25彩色字符显示   | 101000        | L      | L    |
| 80×25黑白字符显示   | 101101        | L      | L    |
| 80×25彩色字符显示   | 101001        | L      | L    |
| 320×200黑白图形显示 | x01110        | L      | H    |
| 320×200彩色图形显示 | x01010        | L      | H    |
| 640×200黑白图形显示 | x11110        | H      | H    |

注：L：表示低电平，H：表示高电平，X：表示任意值“1”或“0”

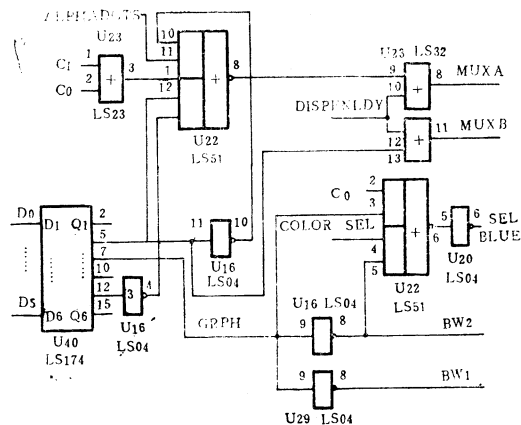


图4 显示方式控制电路图

方式寄存器第2位, 即U40—7输出高电平, 经倒相产生BW1、BW2低电平信号, 送往复合输出电路, 封锁了复合输出信号的彩

色显示。反之，输出彩色显示。

方式寄存器第3位,即U4C—10输出高电平,产生VIDEO ENABLE为允许显示信号。

方式寄存器第4位,即U40—12输出高电平时,使选择信号MUXA为高电平。

方式寄存器第5位,即U40-15输出高电平,产生允许闪烁信号ENABLE BLINK。

由此不难看出,由方式寄存器输出的状态形成了控制选择信号MUXA, MUXB,并进而控制形成各种显示方式。

前述的故障现象是只显示字符，而不能显示图形。对此类故障，我们按上述的分析可知：

显示字符时,方式寄存器第1位为“0”,MUXB为低电平,显然这一类故障是由MUXB信号恒为低电平。通过图4不难查出U40、U23中任一组件逻辑功能失效或与方式寄存器第1位对应的传输通路出错都有可能引起选择信号MUXB变为低电平,造成只能显示字符而不能显示图形的故障现象。

对于其它类似的故障现象，也就不难分析与解决了。

## 2. 字符颜色及底色错误的故障

此类故障多发生在彩色编码器及有关的逻辑电路中，图5为I、R、G、B形成逻辑图。

它由字符/图形锁存器U34、属性/图形锁存器U35、彩色选择寄存器U93、彩色编码器U9、U10、U101、驱动器U67和图形素串/并移位器U7、U8组成。

在字符方式下,从VRAM中读出信息。字符/图形锁存器,存放字符代码CC7-CC0,它作为字符发生器ROM的高位地址A10—A3。属性/图形锁存器,存放属性代码AT7—AT0,它直接送入彩色编码器。在图形方式下,字符/图形锁存器、属性/图形锁存器分别接收偶/奇字节像素,经图形串/并移位形成控制信号C0、C1送入彩色编码器,彩色编码器将字符或图形的像素的前景/背景产生视频三元色(R、G、B)和亮度I,经驱动

器U67送往彩色显示器。

彩色编码器由两个双四选一数据选择器U9、U10(74LS153)和一个六D触发器U101组成。在允许端STR为低电平时,根

据选择信号MUXB、MUXA的状态,从输入信息中选择其中四个数据作为输出。表3示出了各种显示方式下I、R、G、B的内容。

在字符方式下,选择信号MUXB为低电

表3 各种显示方式下I、R、G、B的内容

|                        |               | MUXB | MUXA | STL | I                   | R              | G              | B              |      |
|------------------------|---------------|------|------|-----|---------------------|----------------|----------------|----------------|------|
| 字 符<br>方 式             | ALPHADOTS=1   | 0    | 0    | 0   | AT3                 | AT2            | AT1            | AT0            | 前景色彩 |
|                        | ALPHADOTS=0   | 0    | 1    | 0   | ENABLE<br>BLINK,AT7 | AT6            | AT5            | AT4            | 背景色彩 |
| 320×200<br>彩色图形<br>方 式 | C0=0或<br>C1=0 | 1    | 0    | 0   | BACK<br>GROUND 1    | C1             | C0             | SEL<br>BLUE    | 前景色彩 |
|                        | C0=C1=0       | 1    | 1    | 0   | OVER<br>SCAN 1      | OVER<br>SCAN R | OVER<br>SCAN G | OVER<br>SCAN B | 背景色彩 |

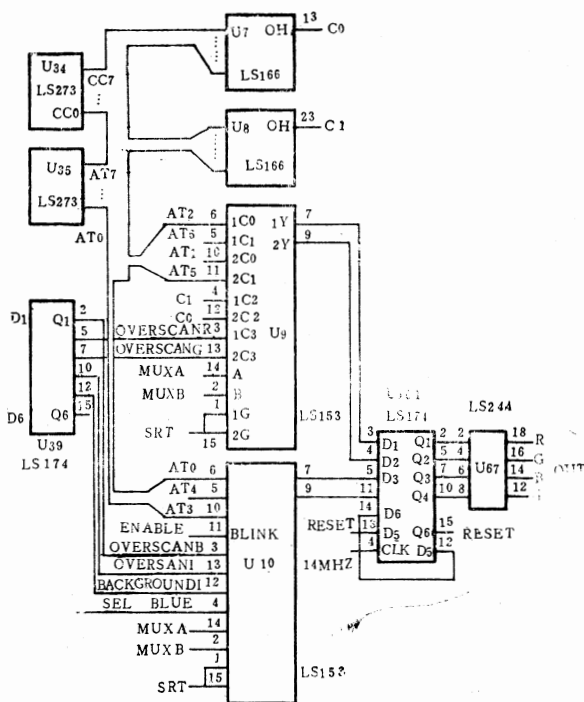


图5 IRGB形成逻辑图

平,若字符串/并移位器有字符代码输出则选择信号MUXA为低电平。此时I、R、G、B由字符属性代码AT3—AT0决定,然后从16种相应的色彩中选择一种颜色,来显示前景色彩(见表4)。

若字符串/并移位器无字符代码输出,则选择信号MUXA为高电平。此时,I、R、G、B由ENABLEBLINK、AT7及AT6—AT4决定,并选择相应颜色作为背景色彩。

在320×200彩色图形方式下,选择信号MUXB为高电平,若C1≠0或C0≠0 MUXA为低电平,双四选一数据选择器1C2、2C2输入信号作为1Y、2Y的输出。即选择BACK GROUND I、C1、C0、SEL BLUE作为I、R、G、B,来显示图形象素的前景色彩。

其中,BACKGROUND I是彩色选择寄存器第4位信息,SEL BLUE由彩色选择寄存器第5位COLOSEL决定(参见图4)。

若C0=C1=0,则MUXA为高电平,双四选一数据选择器选择1C3、2C3作为1Y、2Y输出,即I、R、G、B由彩色选择寄存器第3位—第1位决定,来显示图形的背景色彩。

根据上述的分析,我们可以对前述的故障现象进行分析和判断。

当开机时屏幕背景呈现红色,而字符及前景颜色正常时,显然故障发生在背景色彩的显示部分,且是红色显示出问题。而故障部分是在U9、U10组件之前,而不是之后

(因为前景颜色显示正常)。

通过分析可知,在 $I=0$ 的条件下,如果AT6恒为高电平,从表3和表4可以看出,若AT4亦为高电平时,其背景色彩将变为绛红色,若AT5为高电平时,其背景色彩将变为褐色。显然,故障是由与AT6有关的部位U35、U9—5及相应的传输通路,引起该位

为高电平所至。

其它类似的故障现象,也不难分析和检测。

显示适配器的故障是多种多样的,在这里我们只能挂一漏万的对一些常见故障现象,进行分析和检测。由于作者水平有限,错误之处在所难免,敬请批评指正。

表4 十六种可选择的色彩

| 序号 | I | R | G | B | 色彩 | 序号 | I | R | G | B | 色彩  |
|----|---|---|---|---|----|----|---|---|---|---|-----|
| 0  | 0 | 0 | 0 | 0 | 黑  | 8  | 1 | 0 | 0 | 0 | 灰   |
| 1  | 0 | 0 | 0 | 1 | 兰  | 9  | 1 | 0 | 0 | 1 | 淡兰  |
| 2  | 0 | 0 | 1 | 0 | 绿  | A  | 1 | 0 | 1 | 0 | 淡绿  |
| 3  | 0 | 0 | 1 | 1 | 青  | B  | 1 | 0 | 1 | 1 | 淡青  |
| 4  | 0 | 1 | 0 | 0 | 红  | C  | 1 | 1 | 0 | 0 | 淡红  |
| 5  | 0 | 1 | 0 | 1 | 绛红 | D  | 1 | 1 | 0 | 1 | 淡绛红 |
| 6  | 0 | 1 | 1 | 0 | 褐  | E  | 1 | 1 | 1 | 0 | 黄   |
| 7  | 0 | 1 | 1 | 1 | 白  | F  | 1 | 1 | 1 | 1 | 强白色 |

(上接第33页)

IBM—5550机的CPU为8086,时钟频率5MHZ,周期长度是200ns。以5550机中的转换进程所对应的静态程序为例。执行该段程序转换一个汉字合计需花费143个时钟周期,约合时间28.6 $\mu$ s。如果5550机是利用电话线与长城机进行远距离通信或连入3+网,则最大的传输速率是2400bps,即一秒钟约传送120个汉字,平均传送一个汉字的时间是8,333 $\mu$ s。由此可见这种方式下的接收方除了采用中断方式外,也可以采用查询方式进行数据接收。近距离通信以及长城机的分析雷同,在此不再赘述。

注1:高低——指汉字内码的低字节或高字节。

#### 四、结束语

本文介绍了5550机与长城机的实时通信中的关键问题和解决方法。由于篇幅所限,许多问题无法在这里详细讨论。目前,这一方法已成功地应用在贵州轻工厅异种微机通信网络中。

笔者认为,由于5550机与我国的主流机种的不兼容性,造成我国进口的大批5550机处于闲置状态。这种情况对于我们这样一个国力单薄的国家来说是一种巨大的资源浪费。5550机的实时通信的开发无疑缓解了这一现象。

目前,我国已进口和正在进口大量的各种微机,如何将这互不兼容的机器连成网络实现通信,是一紧迫问题。5550机与长城机的实时通信的成功开发,为这一问题的解决提供了一条捷径和方法。(参考文献略)