

微小型计算机



开发与应用

MICRO—MINICOMPUTER
DEVELOPMENT & APPLICATION



1990 3

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

地址：天津市河西区友谊路爱民道5号

发行：天津市邮局

印刷：天津武清县长宏印刷厂

订购处：全国各地邮局

定价：0.95

邮局代号6—87

津工商广字0146号

国内统一刊号CN12—1152

《微小型计算机开发与应用》

编辑委员会

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

编辑:《微小型计算机开发与应用》编辑部

出版:天津市电子计算机研究所

天津市电子计算机学会

地址:天津市河西区友谊路爱民道5号

(邮政编码:300061)

发行:天津市邮局

印刷:天津市武清县长宏印刷厂

订购处:全国各地邮局

定价:0.95

1990年第3期目录

(总第47期)

应用实例

- GTY—10型弹簧实验机微机数据采集处理系统
.....郑荣良 王勇 秦代克 王德军(1)
- 电机运行参数的微机监测系统
.....周鸣争(4)
- 单片机在水泥生料配料中的应用
.....林振良(6)
- 用TP801实现电视发射机的自动控制
.....陈永春 孔俊宝(9)

计算机软件

- 一种新型数据库—BC数据库
.....刁成嘉(12)
- 施工图预算系统的设计·计算表
.....赵家俊(17)
- VAX/VMS系统的软接口.....赵彦斌(21)
- 一个小型汉字显示程序.....龙安川(25)
- 管理信息系统开发过程中的快速
软件开发工具.....曲廉生(28)
- 基于Turbo prolog的CAI软件
开发技术.....何惠治(33)
- 微机在高校学生综合测评工作中的应用
.....张克善 张品福(38)

接口与通讯

- IBM PC机异步通信适配器(RS—232口)
基本程序及其使用方法
.....苗长云 杨素君(41)
- Apple机与TP801机之间的数据
通讯方法.....杨振江(45)

CONTENTS

APPLICATION EXAMPLE

Microcomputer Data Acquisition and Processing System for GTY-10		
Spring Tester.....	Zheng Rongliang	Wang Yong
	Qin Daike	Wang Dejun (1)
Monitoring System for Motor's Running Parameters Using Microcomputer		
.....	Zhu Mingzheng	(4)
Application of Single Chip Microcomputer in Raw Material Burden of		
Cement.....	Lin Zhenliang	(6)
Implementing Automatic Control of TV Transmitter Using TP 801.....		
.....	Chen Yongchun	Kong Junbao (9)

SOFTWARE

A New Type Database—BC Database.....	Diao Chengjia	(12)
Designing the Budget System of Working Drawing The Computed Entry Table.....	Zhao Majun	(17)
The Soft Interface for VAX/VMS System.....	Zhao Yanbin	(21)
A Small-sized Program for Chinese Character Display.....	Long Anchuan	(25)
A Fast Development Tool for Software in Development Procedure of Management Information System.....	Qu Liansheng	(28)
Development Technology of CAI Software in Turbo Prolog.....	He Huizhi	(33)
Microcomputer Application in Comprehensive Evaluating College Students	Zhang Keshan Zhang Pinfu	(38)

INTERFACE & COMMUNICATIONS

Asynchronous Communication Adapter (RS-232 Port) Support Program
and Application for the IBM PC Microcomputer..... Miao Changyun Yang Sujun (41)

A Data Communication Method between Apple and TP 801
Microcomputers..... Yang Zhenjiang (45)

GTY-10型弹簧试验机微机数据

采集、处理系统

江苏工学院 郑荣良 王 勇

青岛弹簧厂 秦代克 王德军

摘要 本文介绍了微机的接口电路、传感器及软件。

一、问题的提出

GTY-10型弹簧试验机是由济南试验机厂生产的,主要用于测试汽车钢板弹簧和螺旋弹簧的载荷——变形特性。国内很多弹簧生产厂家都正在使用该机。它有三种量程:2T、5T、10T。在测定弹簧的载荷变形特性时,为了能较准确地读出各给定载荷下的变形量,通常这些点的加卸载速度应很慢或为零。因此当测定点较多时尤其是当需要绘制特性曲线时,需要花费很长的时间。另一方面,试验机靠人工操作也很难将压力值精确地稳定在测定点上,因此对测量精度也有一定的影响。

近年来,计算机技术已取得了迅速发展,尤其是微型计算机,由于成本低,体积小,功能强,为计算机应用和推广开辟了广阔的领域。为提高试验机工作效率和测量精度,江苏工学院与青岛弹簧厂合作将微型计算机用于GTY-10型试验机的改造,实现对数据的自动采集、自动记录、自动处理、自动绘图。大大缩短做试验的时间,并提高了精度,节省了人力。

二、硬件设计

1. 设计原则

应保证试验机的位移及力的量程不变;尽量采用原设备的传感器,以降低改造成本;

微机系统一旦有故障,原测试系统可继续使用;计量部门要对试验机定期鉴定调整,微机测试系统的测量精度应不受其影响。

2. 测量系统

符合上述原则的测量系统如图1所示。

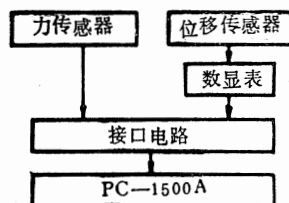


图1

位移传感器选用原试验机的同步感应器,它把位移信号转变为电信号送数显表,数显表显示动态过程位移值(精确到0.1mm),数显表的位移脉冲信号送接口电路,由接口电路处理后送PC-1500A。另一路是力传感器。测量力信号的方法很多。经过分析比较,我们选用了增量式光电脉冲角位移传感器,使传感器与测力盘测力指针同步转动,通过测量指针的转角来测量弹簧的载荷。该传感器每周输出5000个脉冲,分辨率较高。接口电路,对来自力传感器,位移传感器的信号进行电平转换,分别计数,再通过三态输出器与计算机数据总线相连。

计算机选用日本夏普公司生产的PC-1500A型袖珍计算机,它除可进行科学计算,还可进行控制与检测。既可使用高级语言,又可使用汇编语言。操作简单,维修方便。

3. 力传感器的安装

力传感器的安装方式如图2所示。

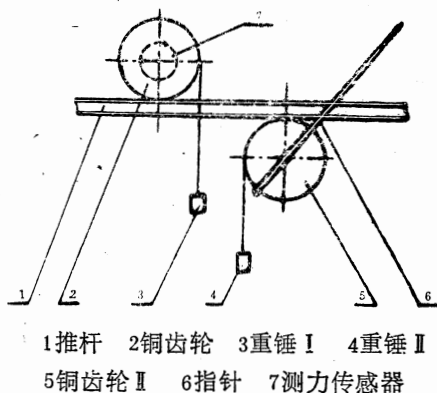


图2

除件2、件3、件7外，其余都是试验机原有件。件2与测力传感器7同轴转动，件3是为了抵消在卸载时由于增加件2而产生的阻力，件1是M12×1的螺杆。件2、件5通过圆周上的小齿与件1啮合，适当调整件1与件2、件5的啮合程度可使件2与件5的转角完全相同，从而也就保证了传感器的输出与表盘指针6的示值一致。在改变量程时也不必调换传感器。

4. 接口电路

接口电路如图3所示，由电平转换器CD4049，加减计数器CD40193，三态输出器D74H244C，双D触发器74HC74，或门CD4071，三八译码器74HC138等16块集成块组成。CD4049的作用是将两传感器输入的脉冲信号电平转换成与PC-1500A兼容的

电平。CD4049的地线与两传感器的地线直接相连。CD40193对来自位移及力传感器的脉冲信号进行计数。计数值通过三态输出器送数据总线。三态输出器为低阻时，计算机可采集计数值，为高阻时，计数信号与数据总线隔离。三态输出器的状态由三八译码器74HC138控制。计数器的清零信号由程序清零和机械清零共同控制，两者为相或的逻辑关系。程序清零信号由双D触发器74HC74产生。机械清零信号由试验机压头上的微动开关控制。

三、软件设计

软件设计时，应考虑程序的通用性。对经常变化的参数，如量程K，最大载荷 P_M 等，采用人机对话方式输入。为使计算机能及时地采集数据，程序采用跟踪查寻法，用汇编语言编制跟踪程序，其框图见图4。

采样各点的载荷值通过子程序1存放在相应存贮单元，供计算机跟踪查寻用。子程序2是跟踪查寻子程序，子程序3是绘图子程序，最后的人机对话是要操作人员输入是否要重复试验同一弹簧。若要重复试验，计算机将转入测试状态，否则结束试验。

1. 子程序1的说明

该子程序把各采样点的载荷值转换成脉冲数，再存放在&5200为首地址的存贮单元中，每采样点分配两个单元。框图见图5。

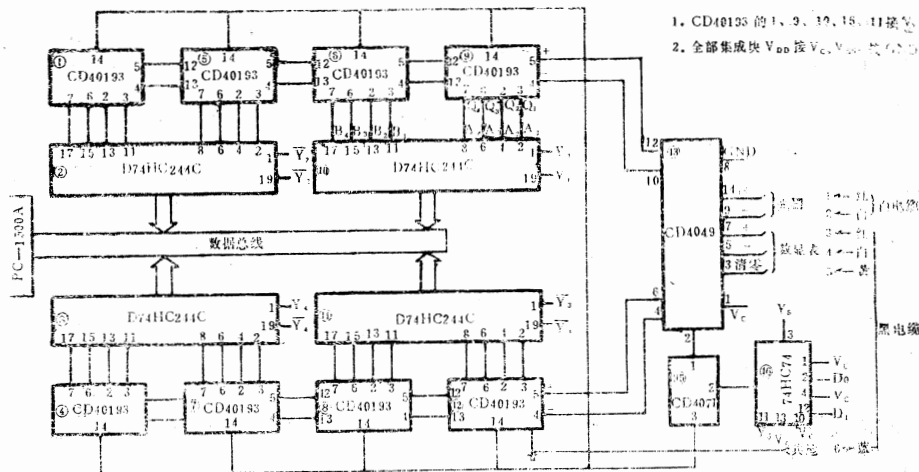


图3

2. 子程序 2

该子程序为跟踪搜寻采样子程序，用汇编语言和BASIC语言混合编成，框图见图6。

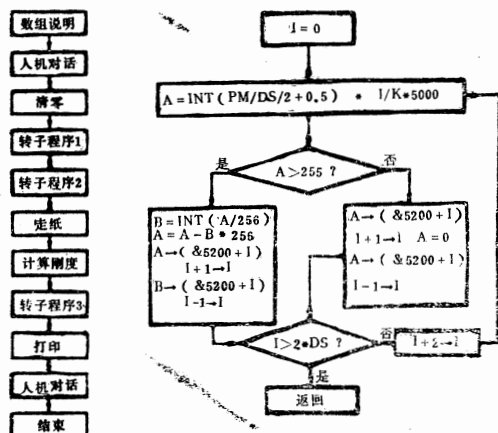


图4 总框图

图5 子程序框图

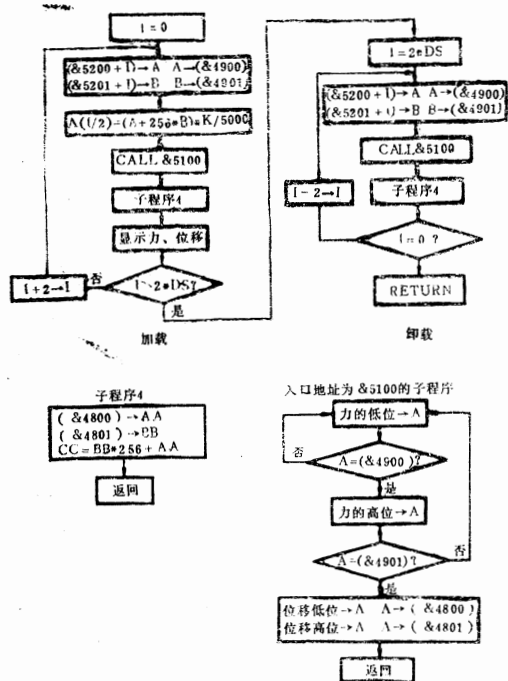


图6 跟踪搜寻子程序

把放在 $(\&5200 + I)$ ， $(\&5201 + I)$ 两单元的数值分别送A、B，再分送到 $(\&4900)$ ， $(\&4901)$ 两单元。调用入口地址为 $\&5100$ 的汇编语言子程序。该子程序把力传感器的脉冲数与 $(\&4900)$ ， $(\&4901)$ 两单元存放的脉冲数进行比较，当相等时，立即采集位移数值，并送 $\&4800$ ， $\&4801$ 两

单元内。再转子程序4，计算位移的大小，并将位移值和载荷值显示出来。若不相等则继续跟踪。当所有的载荷点的位移全部采到后，再按与加载相同的方式采集各点的卸载位移。

3. 子程序 3

打印子程序，打印格式见图7。

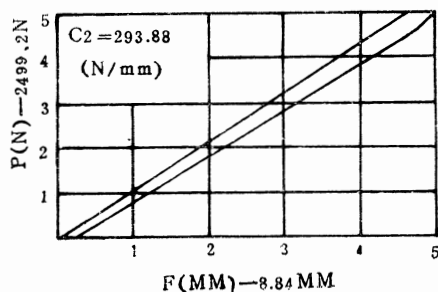


图7

P (N)	f_g (mm)	f_x (mm)
0	0.0	1.4
832	2.8	4.6
1668	5.5	7.6
2500	8.3	10.7
3332	11.0	13.5
4164	13.7	16.7
5000	16.6	19.7
5832	19.4	22.7
6664	22.2	25.7
7496	25.0	28.6
8332	27.7	31.5
9164	30.6	34.6
9996	33.2	37.5
10828	35.9	40.2
11664	38.7	42.7
12496	41.5	44.2

$$F(10) = 29.6 \text{ MM}$$

$$C1 = 281.49 \text{ N/MM}$$

$$C2 = 293.88 \text{ N/MM}$$

其符号意义为：

P —— 载荷，N；

f_g —— 加载时的变量，mm；

f_x —— 卸载时的变量，mm；

电机运行参数的微机监测系统

安徽机电学院 周鸣争

摘要 电机运行参数的微机监视系统,是为芜湖县鲁港排灌站设计的一套电动机组联机巡回检测系统,它分别对十一台电机的运行工况实现在线的集中检测、显示、记录和超限报警。本文对其硬件结构和软件设计作一介绍。

一、前言

在电机运行过程中,其电流、电压、温升等参数是工作状态好坏的一些重要标志。通常为保证电机安全可靠的工作,需要对这些参数进行检测和监视。一般情况下这一工作由值班人员按手工的方式来进行,其检测速度慢,工作量大,有时还会造成一些人为的差错。为了提高电机运行参数检测的精度和可靠性,减轻值班人员的工作量,设计了这套系统。

二、硬件结构

该系统主要由检测电路、多路模拟选择开关矩阵、A/D、Z-80单板机、微型打印机和报警电路几部分组成。其结构如图1所示。

1. 检测电路

由十一个温度变送器、十一个电流变送器、三个电压变送器和一个总功率变送器组

成。完成对十一台电机运行时二十八项参数进行检测,并转换成相应的电压信号,作为A/D接口的输入。

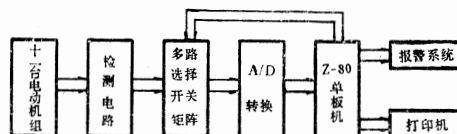


图1 系统结构框图

2. 多路模拟选择开关矩阵

由于在系统中,我们只使用了一路A/D转换器,因此必须要经过选择开关对二十八项参数进行选择。整个开关矩阵由译码器、驱动电路和继电器矩阵组成。其原理如图2所示。

图中我们采用了随机采样方式,使用了两级地址译码(即一次电压译码、二次电流译码)。当微机需要对某一参数进行检测时,将该参数的采样地址送入PIC A口的低七位。该地址通过行译码和列译码便分别得到八根行线和四根列选择线。然后再送往行驱动和列驱动电路。使三十二个继电器中与地

F(10)——额定载荷时的加卸载变形量平均值。

C2——两点刚度, N/mm;

C1——一点刚度, N/mm。

第10点载荷为额定载荷, 第7点和第13点用来计算两点刚度。

在程序设计时,已充分考虑到使操作尽量简单,初中以上文化程度人员,只要略加指点就可根据使用说明书正确使用。1989年3月经青岛市计量局测定,精度符合要求。满量程时,计算机测试值与表盘指针示值之差不到1%,性能稳定可靠。

四、结语

址编码相对应的一个继电器导通,参数的电压信号通过该继电器的导通触点送入A/D接口的输入端。在矩阵中,每一路继电器线圈都串接一个二极管,其目的是为了防止驱动电流的串扰,保证开关矩阵工作可靠。

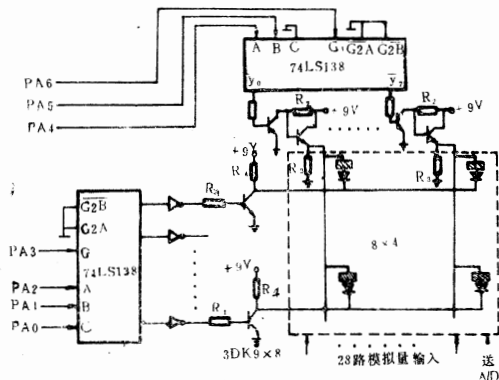


图2

3. A/D接口与参数打印

系统所使用的单板机为安徽无线电厂生产的 DBJ-Z80 单板机。该机具有可编程的 PIO接口和 CTC定时器,并有八路 A/D和一路 D/A转换器。其 A/D转换器的分辨率为八位。转换时间为 $100\mu s$ 。均能满足系统精度要求。

按照系统功能,当某一电机的工作参数超限时,要求系统在进行声光报警的同时进行参数的打印记录,为此特配制了一台 MP-16B 型微型机械式点阵打印机。该打印机在单板机 CPU 的管理下,能打印全部 ASCII 字符和少量的汉字。以作为系统打印输出设备,满足打印记录的要求。

4. 声光报警

在电机运行时,若某一参数超过额定范围,这时系统需通过报警以提醒值班人员的注意。整个声光报警电路由译码器、锁存器、驱动电路、单稳延时电路、声响和发光器件所组成。其结构如图3所示。

当单板机检测到某一参数已超过额定范围后,立即将其报警地址送入 8255 B 口,这时五根输出线经译码后得到三十二路报警信号并使其其中一路有效,该信号与 PA₇ 一起作用,使相应的锁存器输出为“1”,驱动电路中相应的电流开关导通使灯泡发光,进行光报

警。如此同时,PA₇ 为“1”时触发单稳电路工作,产生一个八秒的延迟脉冲来驱动声响元件进行声报警。系统通过声光报警可使值班人员立即知道报警所在的物理位置,以便准确的排除故障,保证电机安全、可靠的运行。

三、系统的主要功能

在系统设计过程中,为了减少硬件工作量,我们尽量避免对原单板机进行修改,只在原机的基础上进行功能的扩充。通过设计后,整个系统具有以下的主要功能:

1. 时间的显示和打印

系统在运行时,在对二十八个参数巡回检测的同时,并在显示器上进行时、分、秒的时间显示。在进行记录打印之前,先将打印时间打印在参数之前。

2. 电度值累加

系统以 1 次/秒的速率对电度值进行采样并进行累加和打印。

3. 各参数的自动检测

系统以 1 次/10 秒的速率对二十八个参数进行巡回采样、判断,并能进行各参数的记录打印和超限报警。

4. 记录打印

记录打印有三种方式

(1) 定时打印:系统每隔一小时对十一台电机所有的二十八个参数的值自动进行记录打印。以作为当班记录使用。

(2) 单机打印:当操作人员只需要对某一台电机所有参数进行查看时,可从键盘上打入单机打印命令,系统就能将该电机的三相电压、电流和电机温度以及当前时间打印输出。完成实时查询打印的功能。

(3) 报警打印:当系统在进行声光报警的同时,亦能自动将报警参数的名称、机号和其值的大小打印在记录纸上,以供检查使用。

四、系统的软件设计

在进行系统软件设计时,考虑到单板机的性能,内存的大小以及维修,使用人员的

单片机在水泥生料配料中的应用

宁波机电工业研究设计院 林振良

摘要 本文叙述了用单片机80C31为主机, 联接外设如A/D接口、D/A接口、CRT显示接口等组成实现生料配料集中控制, 并充分考虑了信号输入、输出中传输隔离及软件结构化的特点。以低成本求高效率, 充分保证了系统的抗干扰能力及系统的正常运行。

国内计算机在水泥生料配料控制环节中, 已得到了广泛应用。其中大多数采用了系统机或单板机。从系统机来讲, 一般采用Apple或IBM-PC 兼容机, 它所具有的特点是功能全、处理能力强, 并可配接许多外设, 以求得水泥厂的工厂一体化控制要求。对于单板机而言, 大多数厂家皆采用Z-80单板机为主机并配接外设构成了计算机控制系统。但是, 由于系统机成本高, 投资大, 对于大型水泥厂比较适用。为了适应目前普遍以立

窑为主的中、小型水泥企业工厂实时控制所存在干扰大的特点, 我们不用单板机, 而采用了80C31单片机为主机并配以外设如A/D、D/A、存储器、CRT显示器及接口、打印机接口等, 实现了计算机集中控制系统。采用CRT 显示, 有效地弥补了单板机等用LED 数码管显示所带来的不直观的缺点, 有效地实现了成本低、抗干扰性能高、硬件通用性及实用性等特点。通过软件的修正还可广泛应用于化工、石油、玻璃等类似行业

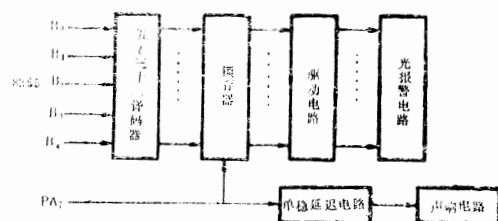


图3

专业技术水平等因素, 力求做到软件使用方便。因此, 我们把很多复杂的计算化为查表技术来解决, 从而使程序简化, 缩短了程序的运行时间。整个系统程序由二部分组成。

1. 主程序

主程序主要完成系统的初始化, 各键盘命令的解释和执行以及定时、报警打印等功能。其流程如图4所示。

2. 中断服务子程序

由于系统要求要有实时性, 因此我们在软件设计中采用了Z-80中断方式。系统在执行初始化时, 将中断向量的高八位地址送入中断页面寄存器, 低八位地址送入 CTC 接

口, 每隔1秒CTC向CPU发出中断请求, 系统转向中断服务子程序执行。以完成对各参数的巡回检测和超限判断以及报警功能。其流程如图5所示。

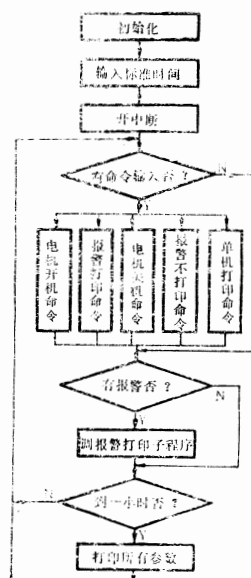


图4

(参考文献略)

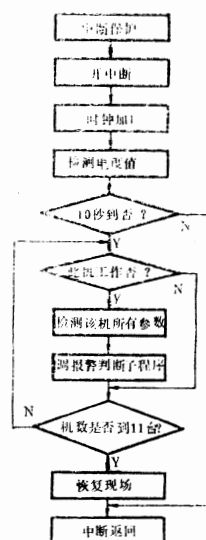


图5

的计算机现场实时控制系统中。

一、系统结构及硬件设计

本系统工作原理是通过传感器所测信号经过放大器放大后送于计算机，计算机通过模数转换开关A/D把模拟信号转换成数字信号，此信号经过折算获知实际瞬时流量，由此计算机每时每刻通过实际瞬时值与实际给定值误差实现实时控制并输出控制数字信号，经过D/A转换成模拟控制量控制触发器的导通角，从而控制电振机的输入电流来改变电振机振幅由此改变实际下料量，使实际流量逼近于给定流量，以至相等，从而控制每种物料的计量精度，以达到生料配料中每种物料的配比精度。

针对本系统工作原理,系统硬件设计主要包括三大部分。

1. 主机部分
2. 输入输出部分
3. 供电部分

硬件结构图如图1。

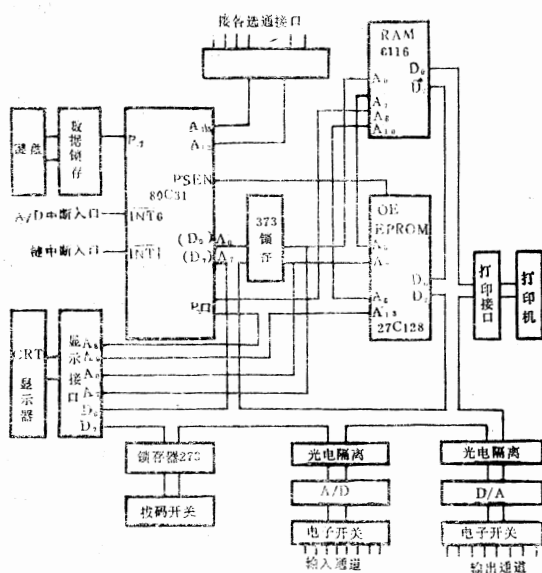


图1 系统框图

主机部分：采用80C31，微机扩展部分

包括6MHZ时钟、16K EPROM、2K RAM、SHARP (夏普) CE-515P打印机、CRT显示器及其接口、外RAM 保护及掉电恢复电路。

输入输出通道：包括输入信号模数转换电路(A/D)、输出信号数模转换电路(D/A)、人一机对话键盘接口及拨码开关输入接口。

供电部分：采用隔离变压器、高频滤波器、整流电路及稳压块。

下面就系统硬件主要部分作一概述。

(1) 存贮器。以充分考虑计算机的内存容量及存贮能力, 保证软件的充分利用以及数据存贮、CRT显示所带来的简易表格处理及打印处理, 在此本系统采用16K EPROM作为程序存贮器, 2K作为数据存贮器, 并且为了防止掉电等意外情况引起外部有效数据的丢失, 本系统采用了干电池数据保护及开关性RAM的强制保护。

(2) 打印机。在现场实时控制中,有必要对现场数据进行记录,以利于今后工厂生产管理的报表及统计,这就要求现场控制系统应具有打印功能。在此本系统选择了打印清晰且体积小、成本低的四色打印机夏普CE-515P打印机,并通过74HC273锁存器作为打印机接口与主机进行数据打印处理。

(3) CRT显示器及接口。在工厂实时控制中,使用者总是要求操作简单、显示直观。以往常常以LED数码管作为显示手段,这样显示处理繁杂且不能一目了然。本系统弥补了这一缺点,采用了显示清晰、字形为 5×7 点阵、分辨率高、逐行扫描的CRT显示器,并相应设计了CRT显示接口,使得本系统人——机对话操作更加形象化、直观化。

(4) A/D及D/A转换器。由于模拟信号是经传感器检测,并经放大器放大而得到。由于现场环境的恶劣性,就避免不了较频繁的瞬时干扰及高频噪声干扰,除了用软件进行数据处理及硬件的RC滤波外,还要求系统A/D片抗干扰性能好。在此本系统采用了

精度高、抗干扰性能好的A/D转换片 ICL 7109, 并采用中断方式, 使CPU在A/D转换时可处理其它事情。由于系统采用集中控制, 各种物料的信号由电子开关CD4051 转换, A/D轮流采样。如果因集中控制引起采样处理不好会产生输入信号混杂, 得不到真实物料数据, 此A/D转换弥补了这种不足, 它具有转换结束后的自动校零功能。这样在本系统对A/D采样中断响应处理A/D采样信号同时, 有足够的时间去切换电子开关以利于一次性得到精确的输入信号, 而且此芯片转换时间快, 转换时间为10ms, 有利于对系统进行有效的控制。对于输出通道, 本系统采用标准8位转换的AD1408作为D/A转换器, 并通过J-FET管高输入阻抗运放TL084相联来有效地控制触发器的导通角。为了防止外部数据信号所带来的干扰影响主机工作, 本系统对输入、输出通道与主机之间采用了光电隔离, 使主机不受从输入输出通道中干扰源所影响, 进一步提高系统的可靠性。

(5) 键盘接口及拨码开关。为了使操作简便, 人一机对话直观。本系统采用类似于系统机键盘标准结构, 设计了键盘接口, 操作标准化、直观化。由于现场最多物料为8种, 有的厂家所用物料不到8种, 为了使本系统实时控制通用性, 通过拨码开关及接口设计, 可以用一台主机进行通用集散控制。

综上所述本系统硬件设计具有以下几个特点:

①采用了CRT显示器及标准的人—机对话功能, 操作简单, 形象化, 直观化。

②硬件设计采用了工业用CMOS芯片, 大大提高了系统对环境的适应能力及抗干扰能力。

③系统中所有输出输入通道全部采用光电隔离, 供电部分采用集成稳压块及相当强的电源滤波能力, 进一步提高系统的抗干扰能力及系统的可靠性。

④通过扩展通讯接口, 便于实现与上位机联网工作, 实现钙铁成份控制及全厂生产

管理工作。

⑤结合硬件与软件设计, 实现系统的自诊, 同时考虑了对外部数据的保护及自恢复电路, 为系统正常运行提供了可靠保证。

⑥通用性强。对于类似的实时控制系统如化工、建材、石油、玻璃等类似系统是一个理想的工业PC机。

二、系统软件设计

系统软件设计采用了模块化结构, 整个系统软件由多个子程序块组成。包括监控、数据采集及处理、控制算法及实时多功能处理程序块。

监控程序块。包括人一机联系、CRT显示、打印处理程序块, 实现系统正常运行所必需的人工操作及状态、数据、参数CRT显示和数据必要的记录。

数据采集及处理。该程序块根据系统实时控制规则“数据采集—控制—输出”, 对现场信号进行采样、滤波、处理, 为实时控制提供准确的第一手资料, 保持系统的正常运行及控制。同时也为系统从量的方面着手提供必要的实际数据; 进行必要处理, 如实际流量、秒累积流量、产量等计算。

控制算法。本系统采用了以偏差为控制目标的PID控制算法, 保证了系统高的计量精度。

实时多功能处理块。实时多功能块包括协调各子程序工作的主程序块以及各种必要的计算及处理功能。如折算系数的自动换算、物料的实际流量计算、每种物料产量计算、总产量计算, 台产量调整以及参变量的修正, 保证系统的正常工作。

总之, 软件设计充分考虑了水泥生料配制的一切功能以及必要的系统自诊, 与硬件相配合, 为系统正常运行提供了可靠保证。

三、结束语

用TP801实现电视发射机的自动控制

南京邮电学院 陈永春 孔俊宝

摘要 本文提出了用 TP801对电视发射机进行自动控制的微机系统。介绍了该系统构成及功能, 硬件配置, 软件功能及抗干扰措施等。本控制系统功能强, 成本低, 控制效果良好, 是当前对省以下电视台的一种理想的控制设备。

一、引言

由于计算机技术的不断发展, 控制自动化这一领域已渗透到各个行业。我国广播电视发射台运用微机实现机房自动控制和管理的工作也已起步, 但尚未取得突破性成果。本文提出用TP801完成10kw电视发射机的自动控制与管理, 实现了电视发射机的开关机自动控制, 自动倒换备机, 数据的自动打印, 故障的自动巡检和报警等, 这对于提高播出质量, 减少停播时间、降低值班人员劳动强度有很大益处。

本文共分三个部分。1.系统硬件, 包括结构及功能, 单板机及其扩展, 微机隔离和发射机接口。2.系统软件, 包括初始化、开关机、自动巡检、报警、计时和打印等功能框图。3.结束语。

二、系统硬件

1.系统结构及功能

根据系统的功能要求和现场环境, 这里采用TP801单板微型计算机作为系统主机。

本系统通过现场近半年多的运行表明, 抗干扰性能强, 可靠性高, 计量精度高, 为物料精确配比提供了高的计量精度。尤其是性能价格比明显高于单板机所构成系统。为以后与上位机联网进行在线成份闭环控制及实时生产管理提供了可靠保证。

参考文献

系统如图1。

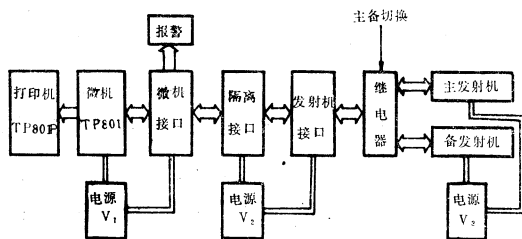


图1 系统的硬件结构框图

整个系统由发射机接口, 接收正在播出发射机的23个模拟量, 以便将反映发射机实时工作状态的电流、电压, 功率等模拟量转换成0—5V的直流电压, 然后通过隔离接口进入微机接口进行A/D转换, 再送入微机进行处理分析, 同时微机根据情况发出控制指令, 通过发射机接口控制发射机。从而完成自动开关机, 自动倒换备机, 自动巡检发射机工作情况, 数据自动打印的功能。电源V₁、V₂、V₃是相互独立的, 以提高系统的稳定性和可靠性。

2.单板机及扩展

本系统采用TP-801单板机所有器件。Z—80单板机有三条总线, 即数据总线DB, 地址总线AB, 控制总线CB。挂在三条总线上

- 1 微型计算机控制系统设计工程—从芯片到系统 清华大学出版社 刘植桢等编
- 2 自动控制系统 机械工业出版社 陈伯时编
- 3 微型计算机外围设备 科学出版社 胡宗藻等编
- 4 MCS—51微机原理及应用(上、下册) 复旦大学微机应用研究室

的器件有Z80—CPU，存储器及Z80—PIO，Z80—CTC等。其中Z80—PIO是通用并行I/O接口芯片，有两个八位口，即口A和口B。本设计根据系统的要求，又扩充了两片Z80—PIO。把原有的Z80—PIO称为PIO I，扩充的两部分称为PIO II、PIO III。将PIO I、PIO II口A规定在位控方式，通过PIO I口A完成主备机电源开关、冷却开关，及自动倒备机等动作，PIO II口A完成主备机电流过荷、天线闭锁自动解除动作。PIO I口B完成备机高压开、天线到位、门开关吸合；主机门开关吸合、天线到位、系统自动/手动等动作。

将PIO II口B，PIO III规定在输出方式，控制告警系统，使发光二极管亮，铃响。告警系统由23个发光二极管和1个蜂鸣器组成。通过亮灯位置，可以确定故障位置。线路连接如图2。3个PIO的口地址见表1。

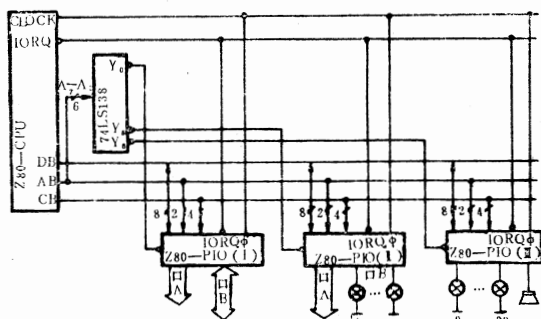


图2 单板机扩展连接图

3. 微机接口 (A/D转换)

A/D作用是将0—5V的模拟电压转换成8位数字信号。A/D采用ADC0809，其时钟频率为10KHZ—1280KHZ，为了保证转换精度，其时钟频率取500KHZ，故时钟频率可由单板机时钟四频后得到。又ADC0809全部数字量引脚电平与TTL兼容，因而可以方便地使用74系列产品。ADC0809的启动由IOW和地址有效共同产生，这样由OUT nn指令，即可启动模数转换，转换结束后用IN nn指令，即可读出地址为nn的模拟量，从而计算机就很容易地用软件来控制模数转换。

因一片ADC 0809只有8个输入，而这里

选用23个模拟量，故采用如图3所示方案。这个电路用三片8通道模拟切换开关CD4051和1片ADC0809相配合，顺序将23个模拟量送入A/D变换。省去了2块A/D片子。为了有效地接通每一路模拟量送去变换，当IO—I RQ有效时，将地址锁存到74LS273之中，通过74LS138对地址译码，则选通口地址00H—17H中的某个口，将此通道的模拟量送向ADC0809。由于A/D时钟周期为 $\frac{1}{500 \times 10^3} = 2\mu s$ ，每个被转换的模拟电压通过比较器s与基准电压进行8次比较，一次比较需要8个时钟周期，因而ADC0809芯片转换所用时间为 $2\mu s \times 8 \times 8 = 128\mu s$ ，又本系统巡检软件采用查询方式，所以在程序中需延时等待128μs以上，在此采用400μs。

表1 PIO 口地址表

	A ₇ —A ₂	译码输出	A ₁ A ₀	口数据寄存器	口地址
Z80-PIO (I)	100000	Y ₀ = PS ₀	00	□A	80H
	"	"	01	□B	81H
	"	"	10	□A	82H
	"	"	11	□B	83H
Z80-PIO (II)	100101	Y ₅ = PS ₅	00	□A	94H
	"	"	01	□B	95H
	"	"	10	□A	96H
	"	"	11	□B	97H
Z80-PIO (III)	100110	Y ₆ = PS ₆	00	□A	98H
	"	"	01	□B	99H
	"	"	10	□A	9AH
	"	"	11	□B	9BH

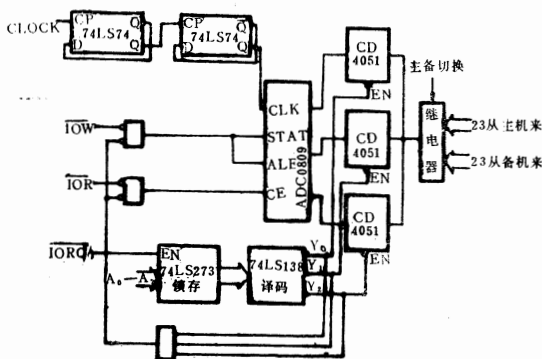


图3 A/D转换电路

4. 隔离接口

本系统采用光电隔离。其最大优点是抗干扰能力强。由于外地线与机内地线是互相绝缘的,所以,外地线上的噪声不会进入机内,光电隔离器的外电源和内电源也是独立的,采用了光电隔离器以后,还可以防止脉冲信号的干扰。

5. 发射机接口的功能

(1) 将反映发射机实时工作状态的电流、电压、功率等模拟量转换成0—5V的直流电压,以便送至微机接口进行A/D转换后,由TP801进行处理和分析。

(2) 给出开关机,自动倒换备机等信号。

要完成上述功能,主要由取样电路,电平转换电路,驱动电路,继电器等组成。本机采用灯丝电压取样。由于发射机图像功放部分的灯丝电压为直流供电,灯丝电压不接地,故采用光电耦合的方法。

三、系统软件

工作过程主要有以下六个部分:

1. 初始化模块 其完成PIO、CTC、A/D等的初始化。

2. 开机模块 其完成发射机的自动开关机。

3. 自动巡检模块 其完成对整机工作状态的巡检,遇到机器不正常情况时,告警并打印。遇到电流三次过荷或声频信号中断,或视频信号中断,可自动倒换备机。

4. 报警模块 其作用为报警,并根据亮灯位置判断故障位置。

5. 计时模块 其作用为记录停播时间,给出开关机时间和打印巡检数据的时间间隔1小时。

6. 打印模块 自动完成发射机工作状态的抄表功能,并在机器出故障时,打印故障参数。

四、结束语

本文提出了一种实用的发射机微机控制方案。控制系统简单,成本低廉,并且给系统的进一步开发提供可能。本系统还可进一步增强功能,尚有扩展的余地。

参考文献

- 1周明德《微型计算机硬件软件及其应用》
- 2肖冬荣《微型计算机实时控制的抗干扰》

光学电脑

由华裔科学家黄庭钰领头的贝尔实验室光学电脑开发小组,研制成功自动光电效应仪器(SEED),这将为光学电脑发展树立了一个新里程碑。在理论上,光学电脑比现在的电脑更具有竞争优势:

1. 光具有物理世界中最快的传输速度,比电子的传输速度快上数十万倍以上,因而,光学电脑的计算速度至少会比现有超级电脑的计算速度还要快上千倍以上。

2. 光子运动时彼此不会产生干扰,也不需要线路的中介传输,因此可以采取平行处理的方式来进行计算,易于模拟人工智能。

3. 可以传递更多的信息,而电子就无法过份压缩。

光学电脑时代的来临,应该是不久的将来。

一种新型数据库—BC数据库

南开大学计算机与系统科学系 刁成嘉

摘要 BC数据库系统(B—二进制数位, C—汉字)采用按二进制数位(Bit)为单位进行数据存贮, 有效地强化了数据的存贮密度, 且具有一般数据库的各种功能。该系统全部采用菜单式会话工作方式, 全部汉化, 并与DBASE—Ⅲ兼容。有效地解决了在微型计算机上处理较大量数据(几万—几十万条数据记录)的问题。

一、前言

微型计算机在非数值数据处理方面已经取得很大进展, 国内外各种数据库系统得到广泛应用, 不少成熟的数据库系统已成为商品出售。但就目前使用最广泛、功能最强的关系数据库系统DBASE—Ⅲ而言, 由于本身设计及硬件配置关系, 受到如下限制:

1. 处理信息量有限

仅以配置有10M字节温氏硬盘的IBM-PC-XT微机而论, 为了对数据库进行快速检索、必须对建成的数据库进行分类和索引, 即按数据文件记录中某些关键字建立一系列的分类文件和索引文件。这样便会使10M字节硬盘的存贮容量, 用掉其中的2/3来存放各种索引文件和分类文件, 剩下1/3强的存贮空间存放其基本数据, 即只有3~4M字节可作为基本数据库使用。

在DBASE—Ⅲ中, 数据按字节为单位进行存贮, 一位数字占一个字节, 一个小数点占一个字节, (如一个人的工资为132.56元, 就要占6个字节)。并且, 数据库中记录要求等长, 即使记录中有的项无数据, 也要留出空间, 冗余量很大。在DBASE—Ⅲ中, 为了减少冗余, 记录长度可变, 即设有细目项(MEMO)。但该项本身就占有十个字节, 用以存放一个指针, 它指向另外一个磁盘文件(备注文件), 在这个文件中, 每个细目项是按512个字节为一数据块进行存放的, 即不管是否有512个字节的内容,

均占512个字节的存贮空间, 其冗余量是相当可观的。

凡此等等, 当数据库中数据记录达到相当数量(如5万条记录)时, 该问题就不可忽视了。(如对于具有5万条记录的数据库, 每条记录增加一个字节, 其存贮空间就要多占50k字节。)这样, 10M硬盘中, 4M字节用来存放基本数据。可存放的记录条数和记录长度受到限制。存贮容量与记录条数和记录长度的关系可表示为:

存贮容量 = 记录条数 × 记录长度

这样, 就使DBASE—Ⅲ中提供的强大功能(如记录总数可达 10^6 条, 每条记录长度可达4k字节等)不能实现。

例如, 天津市卫生局要建立全局约五万职工人事档案信息库, 每个职工一条记录, 每条记录约有180个数据项, 约占1k字节。如果存贮空间为4M字节, 则只能存放约4000条记录, 远远不能满足要求。

2. 存贮容量与检索速度之间的矛盾

为了扩大基本数据库存贮容量, 必须要挤占磁盘存贮空间, 减少相关索引文件, 使检索速度大大放慢, 不能很好地解决检索速度与存贮容量之间的矛盾。

综上所述, 使用DBASE—Ⅲ完成较大容量信息系统的处理很困难, 有必要研制一种可处理相当信息量, 具有较快的检索速度, 适用于多种汉字系统的微机数据系统, 这正是我们研制BC数据库的目的。

二、BC数据库系统简介

“B”、“C”分别是 Bit、Chinese的第一个字母，意思是按位存取全汉字数据库。

为了有效地实现在固定外存容量（10M或20M硬盘）中存放较大数量数据，并且可在相当规模数据中进行快速索引，适用于多种汉字系统，本数据库系统研制过程中，采用了以下几条措施：

1. 提高系统运行效率

为达到此目的，本系统采用了IBM-PC编译高级语言与C语言嵌套形式编程，最终形成全机器码软件系统，大大提高了系统运行效率。

2. 强化数据存贮密度

（1）数据库中数据以Bit为单位进行存贮，可使有效存贮容量成几倍增长。

例如图1所示，一个字节有八个二进制数位（Bit）：

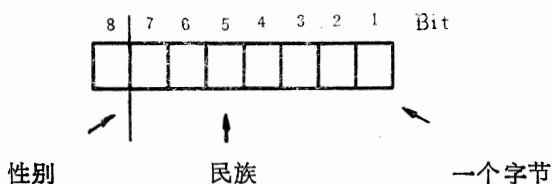


图1

可以分成两部分：第一部分只占一个二进制数位（在最左边第8 Bit），可用来表示性别：0表示女，1表示男。第二部分占七个Bit，可表示0~127共128个状态，我们用第二部分表示民族的代码，我国现有56个民族，再加上外来侨民，128个状态足够用。这样，用1个字节，就可以存放在DBASE-Ⅲ中需用3个字节存放的信息，有效存贮容量得到成倍增加。

当然，按Bit存贮数据的过程，全部由软件系统自动转换完成，用户使用起来还用字节为单位进行存贮一样方便。翻译、转换、显示、打印也由软件系统完成，面对用

户的全是ASCII码数字和汉字。

（2）减少数据库冗余度。

为了有效降低数据库中数据冗余程度，该新数据库系统设立了可变长度记录。与DBASE-Ⅲ中为数据记录提供的细目项（MEMO）一样，本数据库系统也提供了些项功能。但在DBASE-Ⅲ中，细目项本身占十个字节；而本系统只用两个字节即可，可节省4/5的存贮空间。DBASE-Ⅲ中细目项指定的存贮空间中数据是按固定的512个字节为单位进行存贮，冗余量很大；而本数据库系统为每个细目项开辟的存贮空间是由用户自己定义的，可使冗余度大大降低，有效地利用外部存贮空间。

我们采用以上方法，对天津市卫生局的全球人事档案管理系统的记录进行压缩，使得原来要占1k字节的一条记录，只有120个字节就可以存贮，存贮效率提高了近一个数量级。

（3）定期清理数据库。

为了减少数据库中数据冗余程度，本数据库系统中设置了“系统维护”功能块，可根据需要，定期自动清理系统中的冗余数据（如对调出的人员删除记录，调入的插入记录，修改相关索引文件，重新分类排序等）。

3. 提高检索速度

本系统采用了改进B+树结构建立相关索引文件，实现对数据库的快速检索。

4. 适用于多种汉字系统

（1）为适应我国现在微机汉字系统种类繁多，规格不一的局面，本数据库系统的程序代码中不出现一个汉字，可在各种可以使用DBASE-Ⅲ的微型计算机上使用。所有系统需要的汉字（如菜单及各种提示信息等），由用户在启用本系统前，利用自己计算机使用的汉字系统，根据使用说明，首先输入本数据库系统中，并以文件形式存入磁盘，再正式运行本系统。这样，便达到了适用多种汉字系统的目的。

(2) 数据库结构由用户在系统提示下自行建立, 可采用按Bit存贮或按Byte存贮两种形式存贮数据, 由用户根据需要自行挑选。这些结构信息均放在磁盘中相关文件内。

(3) 显示格式和打印格式用户也可在系统提示下自行建立, 结构存入相关磁盘文件, 根据需要随时显示和打印。

5. 与DBASE-Ⅱ兼容

DBASE-Ⅱ是广大用户熟悉的关系数据库管理系统。本数据库系统将在以下几方面与之兼容:

(1) 本系统数据库文件若按字节存贮, 可用常规方法在DBASE-Ⅱ中直接进行处理, 若按Bit存贮, 本系统可根据需要, 先将其转换成ASCII码文件, 然后与DBASE-Ⅱ交互。

(2) 用户在DBASE-Ⅱ中建立的数据库文件, 也可转换成本系统中结构相同的按Bit存贮的文件, 交互使用。

6. 系统结构

本系统的核心文件是BC库, 要处理的基本数据在该库中按位存贮。另外还有用户字典、系统字典、报表文件、单位汇总等几个支持文件。用户字典中存放了用户使用该系统时所需的汉字信息; 系统字典存放的是本系统用于显示菜单、屏幕提示的汉字信息; 把汉字存入专门的文件而不直接出现在程序中, 保证了本系统适用于不同的汉字系统。报表文件规定了打印报表的文件结构; 单位汇总文件存贮了各单位若干项目的汇总参数。另外, 系统还设置了位图文件, 它详尽地描述了BC库的结构。

为方便用户对BC库的操作, 系统又设置了录入数据、数值计算、建立索引、标签打印、查询、系统维护等几项功能。

系统的结构如图2。

系统共设置了五种数据类型, 分别是数值型、代码型、备注型、日期型和字符型(汉字型)。BC主库中的一般基本的数据

项, 多为代码型数据。本来要用几个汉字、占用十几个字节的数据, 现在表示成代码的形式, 有效地压缩了存贮空间。备注型数据, 即细目项, 它指向另一个细目文件。日期型数据也压缩成两个字节表示, 用七位表示年(0~99), 四位表示月(1~12), 五位表示日(1~31)。这也有效地提高了存贮空间的效率。

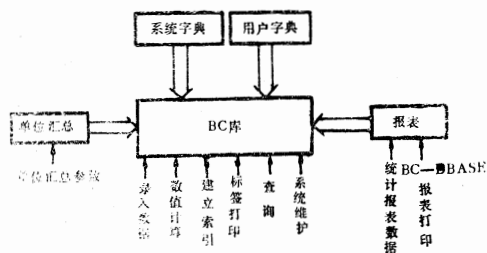


图2

7. 索引文件

利用BC数据库为天津市卫生局建立的人事档案系统中的数据, 经压缩强化后约占4.5M字节。该系统是在带20M硬盘的AT机上实现的。这样, 除了4.5M的基本数据库开销, 及各种表报系统文件约占2M字节外, 剩余的磁盘存贮空间还建立了十二个相关索引文件, 由于有了这些索引文件, 可以方便, 快速地对数据库进行各种访问和操作。

经试验证明, 在十万条记录时, 利用索引文件约一秒钟便可查到要找的记录。这样, 过去需要小型机才能完成的工作, 在微型计算机上就可方便的实现了。

8. 报表、字典和单位汇总参数及标签的打印

报表可根据用户要求, 由用户自己设计。报表, 字典使用几率及各单位汇总参数的收集工作是在系统运行时自动生成, 随时供打印。标签由用户自行设计, 打印。字典使用几率和单位汇总参数形成一个DBASE-Ⅱ文件, 用户可在DBASE-Ⅱ中对其进行处理。

在外围机上执行TELEPHN程序(T-ELEPHN取意为用电话线进行通信),便进入通信状态。首先在系统所提示的主菜单下选择查询功能块。该模块能完成对库文件中数据的查询,修改、删除等功能。执行任何一项功能,一般都要先从库文件中选择记录,即在“查询”功能块的菜单中选择“记录选择”这一功能。中心机接到该命令后,便把已建立的所有索引文件通过通信线路显示到外围机的屏幕上,要求选择。系统将根据选定的索引文件和接收到的索引关键字快速地从库文件中找到所需记录。一般是通过“地址姓名”索引文件、以“单位地址码十姓名码”的“GLZ”为索引关键字来进行记录选择。即系统接收到“地址姓名”索引的选择以后,便提示用户输入“单位地址码”和“姓名码”,根据这两个关键字,从库文件中找到所需记录,经过BC数据到DBASE—Ⅱ数据的转换,把该记录的前九项数据显示到外围机上,并要求确认。得到确认以后,便把指针定位到该记录,否则,重新选择。

选择记录以后,就要进行数据的修改,这包括对数值型数据的增加、减小、替换,对非数值型数据的替换以及对整条记录的删除等。若事先没有进行记录选择,便认为是对当前记录的修改。修改前,先选择修改方式,即进行全项修改还是选择项修改。如果要进行选择项修改,系统便把所有数据项及其代码显示到外围机屏幕上,外围机上的用户通过代码,选择要进行修改的数据项。选择完毕,回车后,系统便把这些数据项及其值显示到外围机屏幕。若进行全项修改,则系统把该记录的所有数据项及其值逐项显示到外围机屏幕,等待修改。无论是全项修改或者选择项修改,系统在接收修改数据以前,都要把该记录的“单位地址码”和“姓名码”各取五位拼成十位字符串,赋值给修改文件的“GLZ”一项。设修改前的数值是 m ,读一次键盘后,得 mm ,若 $mm \neq m$,说明要进行修改。这时,再读键盘,便得到真

正的修改值,把这个修改值送到修改文件中,并在修改标志的相应位设上标志。然后,再进行下一项的显示、修改,直到所有要修改的数据项全部修改(或查询)完毕。这时,便形成了修改文件的一条记录。若还要继续修改下一条记录,便从记录选择开始,重复此过程,否则,把修改文件传送到中心机。中心机对收到的修改文件进行检错,若检测出错误,便生成一个出错文件,送回外围机,要求重新修改;对修改文件中没有出错的数据,根据各项的修改标志和修改值,对BC文件中的数据做出相应修改。

若只想查看一下BC文件中的某些数据,不修改,则在“查询”功能块的菜单中选择“显示数据”功能。和修改数据一样,显示数据也有全项显示和选择项显示两种方式。所不同的是,这项功能只能把数据显示出来,而不进行修改,从而也不生成修改文件。

4. 修改BC库文件

下面简单介绍一下中心机如何根据接收到的修改文件,对BC库中的数据进行修改。

本系统对各种类型数据的录入有各自不同的功能块。在修改模块中,有自动修改功能,只要选择此项功能,该数据库系统就会自动把从各下属机器中传过来的修改文件,依次进行自动鉴别,完成修改任务。当然,同时就会完成报表的更新,单位汇总参数和字典使用几率的统计等各种操作。

在本模块开始工作时,先打开外围计算机通过MODEM传送过来的修改文件,对每条记录进行如下的工作:首先根据该记录的删除标志BF000判断是否要删除该条记录,若 $BF000 = "1"$,则通过“地址姓名”索引文件,以“GLZ”值为索引关键字,从BC库中找到该记录并删除它;若 $BF000 = "0"$,就要对该记录的若干项进行修改。

在本系统中,不管是删除、修改或显示某条记录,对BC库的查询是一个十分重要的问题。就天津市卫生局的这个人事档案来

施工图预算系统的设计·计算表

天津财经学院 赵家俊

摘要 本文提出了在DBASE下施工图预算系统的设计思想和设计步骤；设计了与手工计算基本一致的计算表，借助宏代换“&”的功能实现了对工程量和其它有关数据的计算。这样，工程预算工作中长期存在的一大难题获得解决，并为计算机在有关领域中的应用铺平了道路。

一个应用程序系统，总是针对某一专业解决问题的。专业特点，是选择计算机软件工具的依据。选择得当，将把这两者的特征结合起来，使系统设计逻辑清楚，简单化、条理化。

施工图预算系统数据处理的主要特点是：（1）数据量大；（2）计算简单，其中，工程量计算尤为一大难题；姜云飞同志创造的“符号模式匹配”技术把这项工作向前推进了^[1]；但是，因为他使用的是BASIC语言，比较复杂、不易掌握。当我们把姜氏设计的计算表置于DBASE下，一条新思路就摆在面前，问题也就变得极为简单了。工程预算中的这一大难题在此获得解决，并为计算机在有关领域中的应用铺平了道路。

一、数据组织·处理程序系统

一个完整的施工图预算系统的工程直接

说，共有几万条记录，如果用一般的线性查找技术，那么平均的查找时间约为20~30分钟。为解决这一问题，系统提供了用B+树结构建立索引文件的功能，用户可以根据需要，选用不同的关键字，建立若干索引文件。

“地址姓名”索引就是这样的一个例子。

以上只是介绍了对非细目项数据的修改。若要修改细目项，则在形成修改文件时，同时形成一个该细目项的子修改文件，并且与主修改文件相联。因此，在修改时，必须首先判断要修改的是否细目项，形成两个分支，然后对主文件或子文件进行各数据项的修改。具体的处理方式大致相同。

费计算及其工料分析的流程如图1。计算工程量、计算工程直接费及分析其工料消耗，是施工图预算最重要的三个步骤。其中，工程量计算是第一个步骤，也是最关键的一个步骤。

对图1所示流程说明如下：

1. 对任何数据处理都可以表示为：

常数+初始数据→结果数据

在这里，带箭头的线段表示某种数据处理方法，它编织在程序之中。

在所讨论的范围内不变的数据就是常数。常数的概念是相对的；在施工图预算中，单价表是作为常数来引用；但是，在编制单价表时，它本身就是被测量的对象了。自然，并不是任何数据处理都一定存在和需要相应的常数参加。

初始数据可以是实验数据、或测量数据、或统计数据，是数据处理的主要对象。

四、应用前景

本数据库系统用压缩数据，强化存贮密度的方法来提高磁盘的有效存贮空间，解决了有效信息量与检索速度之间的矛盾。这种用软件的方法代替靠更新设备（配置大容量存贮器以致更换计算机）来提高性能的传统方法，既经济，又可靠。对于目前国内众多的IBM-PC/XT及其兼容机的用户来说，要进行较大规模的信息管理，采用BC数据库是一个极好的选择方案。

在工程预算工作中，无论哪一类数据，数据量都是不少的，我们把它们都存放在外存贮器上（一般是磁盘），这称为建立数据库。三种类型的数据对应着三种类型的库，它们的差别仅在于库中数据追加的时间不同。

类程序有PRYF、PRCLJ、PRGZL。(2) 面向结果数据表。一类结果数据一张表，由一个程序来单独处理。流程图中以竖放的双线框来表示。该类程序总是从若干张表取得数据作为加工处理的对象。并把处理结果送到结果表中。从一张表取得数据常有两种方式：逐个记录依次取得；按条件查寻取得。一般地，从初始库取得数据是前一种方式，从常数库取得数据是后一种方式。这类程序有PR01~PR05。

面向计算表的程序与面向结果表的程序组成处理程序系统。

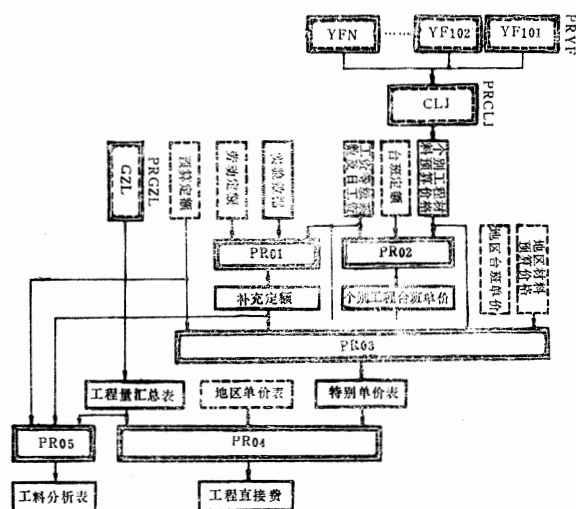
二、表的初态管理系统

总控程序

图1所示流程中的各表，在未启动处理程序系统工作之前，必须处于一确定的初始态：第一，常数表中填好了常数。这些数据具有法令性，除非有上级机关指令，否则不能修改。第二，初始表已经填好了初始数据。若为计算表，还填好了计算表达式。第三，结果表已建立了结构，但无数据记录。

用来确定各表初始态的程序就是初态管理程序。初态管理程序的主要功能是输入或修改初始数据及其计算表达式，建立初始库。常数库的数据具有法令性，应该专门处理，不列入初态管理系统中。结果表的结构一经建立，一般是不需要改动的。但是，初态管理系统的结果表管理程序，必须删除结果表中已经存在的数据记录（上次数据处理的结果）。当然，这个操作也可以放到处理程序系统中，当向结果表第一次追加数据之前来进行。

根据上面的分析可知，施工图预算管理系统的主控程序应该提供对四种功能系统的选择：初态管理系统；处理程序系统；输出表处理系统（主要处理两张输出表：工程预算书及工料分析表）；退出，返回DBASE命令状态。



三、工程量计算表及其处理程序

工程量计算表GZL的结构与内容格式如表1。表为初始态。TW域是为了与图纸核对数据方便而设的。若某一记录的计算表达式为其它记录的计算表达式完整引用,则在该记录的BL域内填一内存变量名。BL域、JSS域左边不留空格。

表处理程序列下:

```
C> type prgzl.prg
use gzl
do while .not. eof ( )
    store trim (jss) to n
    store "&n" to m
    replace sl with m
    store trim (bl) to x
    if x( ) = ""
        store m to &x
    endif
    skip
enddo
return
```

计算表计算表达式的处理之所以能够简单、方便地实施,是得力于宏代换“&”的功能。下面简单地说明这种功能:

• STORE “3+5” TO N

N是一个字符型内存变量,其值是一个带引号的字符串。

• STORE &N TO M

&N将N的值得那个字符串两边的引号拿掉,STORE &N TO M便等价于 STORE 3+5 TO M。M的值是8。M是一个数值型变量。

采用计算表的优点是直观、方便,与手工计算使用的表式一致,不需要经过专门训练就可掌握使用。

四、单位运费计算表

YFR及处理程序

此处, R=101, 102, ..., N。R的最大取值N,就是运费计算表的最大编号。N是一个三位正整数。凡存在运输问题的材料

表1

工程量计算表GZL

DH	TW	SH	BL	JSS	SL	MCH	PJ
46-6		150	Q1	$(5+0.12 \cdot 2) \cdot 2 \cdot 320$			
240-113			Q2	$3 \cdot 18$		J732-1-16	
建筑面积			\$	$(240 \cdot 35) + (18 \cdot 6) - (15 \cdot 3)$			
滑轮			\$	3			15.8
68-2		200	Q3	$3.6 \cdot Q1 + 1.5 \cdot 7$			
44-1			\$	$Q1 + Q2 - Q3 + Q3 \cdot 0.24$			
:	:	:	:	:		:	:

DH—定额编号, C TW—图纸部位, C SH—水泥号, N BL—内存变量名, C
JSS—计算式, C SL—工程量, N MCH—门窗型号, C PJ—补充价, N

就有一张这样的表。在材料预算价格计算表中，有一个域是运费计算表号YFBH，有一个域是单位运费DWYF，是YFBH指出的运费计算表的计算结果。

运费计算表的结构与内容格式如表2。表示为初始态。表的最后一个记录是运费合计数，JE域的值要传送到材料价格计算表CLJ相应记录的DWYF域中。一般说来，材料不同，运费计算表的具体内容就不会完全相同。

表2 运费计算表YFR

XM	QZ	YJ	JS	JE
用 马车运至	交货地点	10.0	0.20 * 10	
起 运 站	至 车 站			
铁路运费	北 京 至	120.0	0.20 * 120	
	天 津			
由终点站用	天津站至	15.0	0.20 * 15	
汽车运工地	工 地			
汽车装卸费			1	
HJ				

XM—项目，C QZ—起止地点，C
YJ—运距/公里，N JS—计算式，C
JE—金额（单位运费，元），N

处理程序列下：

```
C> type pryf.prg
input m
store str (m, 3) to p
do while m< >100
do supryf
store m-1 to m
store str (m, 3) to p
enddo
return
```

从键盘输入m的取值范围是 $101 \leq M \leq 999$ ，取3位整数，是运费计算表的最大编号。

```
C> type supryf.prg
use yf&p
do while xm< > "hj"
store trim (js) to b
store &b to c
replace je with c
skip
enddo
sum all for xm< > "hj" je to d
go bottom
replace je with d
use clj
locate all for yfbh=&p
replace dwyf with d
return
```

五、结束语

本文是作者为讲授“计算机在预算管理中的应用”而撰写的讲义的一部分，经提炼而成。施工图预算是工程预算中比较重要的一环，具有典型性。作者为本课程设计了模型系统，限于学生实习条件，在APPLE II上调试通过。现部分程序已用 DBASE III 改写移植于IBM PC/XT。

参 考 文 献

- 1 姜云飞，“符号模式匹配技术在建筑预算系统中的应用”，1986，4，《微计算机应用》。
- 2 《基本建设预算》，高等财经院校试用教材，基本建设预算编写组 编著，中国财政经济出版社，1984年第1版。

VAX/VMS系统的软接口

57357 部队软件科 赵彦斌

摘要 本文以VAX/VMS为基础, 阐明了通过共享公共变量程序段的方法实现高级语言与汇编语言之间、高级语言之间软接口的理论和技术问题, 通过正确的方式传递参数的实现高级语言和汇编语言与系统过程之间的混和编程。

一、问题的提出

各种程序设计语言, 除了具有共同的特点外, 各自还具有其显著的优点。PASCAL语言有广泛的程序设计结构, 丰富的数据类型和数据结构, 编程清楚、简明; C编译简单, 编写的程序清晰紧凑, 移植性能好; FORTRAN用于计算的性能好; 汇编的目标模块质量好; 能直接汇编成机器语言, 装入ROM中。因应用课题的需要, 程序员往往选用最佳的程序设计语言, 以求编出质量好的程序。为了充分有效地利用各种语言的优点, 我们常常采用混和语言程序设计。

一种语言编写的程序如何调用另一种语言编写的子程序, 参数是如何传递等问题是混和语言程序设计中遇到的主要问题, 这便是如何实现系统软接口的问题。

本篇介绍以VAX/VMS为基础, 高级语言之间、高级语言与汇编语言之间, 以及高级语言和汇编语言与系统过程之间的混和编程。

二、高级语言与汇编语言的衔接

一般说来, 高级语言支持的程序段作为主程序段, 汇编语言支持的程序段作为子程序段。要实现高级语言与汇编语言的衔接, 必须保证参数的正确传递。

在VAX-11中, 连接程序的基本操作是在可执行映象内分配虚存, 解决被连接模块

的符号的引用问题, 并给于浮动的全局符号赋值, 而且, 连接程序所用的程序节允许各模块互相使用对方的变量和函数。例1包含了分别编译的两个C的源程序w1.c和w2.c, 其中, w1.c调用w2.c中的w2()函数, 请参见例1。

```
$ type w1.c
# include stdio
int a;
main ()
{
    w2 ();
    printf ( "%d ", a );
}
$ type w2.c
w2 ()
{
    extern int a;
    a=1;
}
$ r w1
1
```

例1 C外部变量共享内存的源程序清单及运行结果

用下列的DCL命令

```
$ CC w1 ↵ (↵表示回车, 下同)
$ CC w2 ↵
$ link w1, w2 ↵
```

执行上述三条命令后, a落实到同一虚存。因此, 高级语言与汇编语言的接口是通过共享公共变量程序段来实现的, 连接程序解决所有地址之间的转换问题, 重新定位存

贮地址，协调查存分配。例2阐明了FORTRAN是如何调用MACRO的。请参见例2。

```
$ type w33.for
    PROGRAM MAIN
    INTEGER X
    COMMON /A/X
    CALL w2
    TYPE 20, X
20    FORMAT (I4)
    END

$ type w21.mar
    .PSECT A, NOEXE, WRT,
    GBL, LONG, OVR, RD, PIC,
    SHR
A, , .BLKL 1
    .PSECT CODE, NOWRT, EXE
    .ENTRY W2, ^M< >
    MOVL #28, A
    RET
    .END

$r w33
28
```

例2 FORTRAN调用MACRO子程序的源程序清单及运行结果

W33.for是FORTRAN主程序，W21.mar是MACRO子程序。w21.mar将A赋初值28，w33.for将w21.mar传递过来的A通过X打印出来。主程序通过call w2进入子程序入口，实现调用。由于FORTRAN的COMMON说明的公共变量A是块程序段，且具有PIC，OVR，REL，GBL，NOEXE，RD，WRT，LONG属性，因此，在汇编子程序中，对A的说明也必须是具有上述属性的程序段。只有这样，连接程序才能把FORTRAN主程序的公共变量正确地传递给汇编子程序。

C主程序调用MACRO子程序可以用相同的方式实现，只需将C与MACRO的公共变量说明成外部变量即可。利用C与MACRO的混和编程可以将C的结构运算大大简化，例3是MACRO将C的结构ex的成员first,second,third进行运算的一个例子。

请参见例3。

```
$ type cml.c
#include <stdio.h>
#include <sys/types.h>
struct {
    long int first;
    long int second;
    long int third;
} ex;

main ()
{
    val ();
    printf ( "ex.first=%d\n", ex.first );
    printf ( "ex.second=%d\n", ex.second );
    printf ( "ex.third=%d\n", ex.third );
}

$ type mmcc.mar
    .PSECT ex, pic, WRT
    NOEXE, usr, ovr,
    rel, gbl, shr, rd,
    long
first: : .blk1 1
second: : .blk1 1
third: : .blk1 1
    .PSECT CODE,
    NOWRT, EXE
    .ENTRY val, ^M< >
    movl #1, first
    movl #2, second
    movl #3, third
    ret
    .END

$r cml
ex.first=1
ex.second=2
ex.third=3
```

例3 C主程序调用MACRO子程序的源程序清单及运行结果

三、高级语言与高级语言的接口

由于VAX-11连接程序的功能，因此，同高级语言调用汇编语言一样，高级语言之间的调用也是通过共享公共变量程序段实现的。

例4 讲述了C主程序调用 FORTRAN 子程序的方法。请参见例4

```
$ type h1.c
#include math
#include stdio
main ()
{
    extern int x, y, z;
    printf ( "x= " );
    scanf ( "%d ", &x );
    printf ( "%ny= " );
    scanf ( "%d ", &y );
    add ();
    printf ( "z=%d ", z );
}
$ type add1.for
SUBROUTINE ADD
INTEGER A, B, C
COMMON/x/A/y/B/z/C
C=A+B
RETURN
END

$ r h1
x=7
y=9
z=16
```

例4 C主程序调用 FORTRAN 子程序的源程序清单及运行结果

在例4中，h1.c为c主程序，add1.for为FORTRAN子程序。C编译器在编译h1.c时，遇到extern int x, y, z时，便分别为程序段x, y, z分配一个存贮区。C主程序先对外部变量x, y赋初值，通过add()进入FORTRAN子程序段入口，FORTRAN子

程序段入口，FORTRAN子程序ADD对公共块x, y的变量A、B求和，计算出的值赋给公共块Z的变量C，返回后，C主程序将外部变量Z的值打印出来。

C主程序调用FORTRAN子程序时，公共变量在C中用extern说明，在FORTRAN中用COMMON说明。反之，FORTRAN主程序调用C子程序也是这样。例5是FORTRAN主程序调用C子程序的一个例证程序。请见例5。

```
$ type add2.for
PROGRAM SETVAL
INTEGER WWW
COMMON/DF/WWW
WWW=15
CALL PRIN
END

$ type h2.c
#include stdio
prin ()
{
    extern int df;
    printf( ".....%d....."df,); }
$ r add2
.....15.....
```

例5 FORTRAN 主程序调用C子程序的源程序清单及运行结果

四、程序设计语言与系统过程的软接口

VAX-11定义了许多本机态程序设计语言写的系统过程，并且有一套调用过程，允许在过程之间传送变元。只要遵循调用标准的规定，无论何种语言，均可调用系统过程。

C主程序可以通过立即值传送变元，引用传送变元，描述符传送变元，可变长的变元表传送变元调用系统过程。例6说明C通过描述符传送变元的方式调用系统时间库。

```

$ type dcll.c
# include ssdef
# include descrip
main ()
{
    int i;
    $DESCRIPTOR (name, "show
                    time ");
    i=lib$spawn (&name);
}
$ r dcll
1-DEC-1989 16:23:22

```

例6 C调用DCL命令的源程序清单及运行结果

在例6中,\$DESCRIPTOR是予处理宏,完成描述字符串的初始化,建立一个编译时串描述符,并把它传给 lib\$spawn.lib\$spawn 请求调用进程的命令语言解释程序(CLI),为执行DCL命令show time生成一子进程。

在VAX-11中,例6在系统允许后台的情况下,才能运行。

高级语言与系统过程之间变元的传递方式复杂多样,而汇编语言与系统过程之间变元的传递方式则较为简单。在MACRO语言中,只要正确地配置系统过程所要求的参数,便于实现调用。例7是MACRO主程序调用记录管理服务(RMS)的例子。请参见例7。

例7实现了拷贝一个文件的功能。MACRO利用宏指令调用RMS服务例程,实现对文件和记录的操作。\$FAB宏为文件访问块分配存贮区,并对存贮区初始化。\$FAB的参数FNM、MRS、RAT等指定了文件的属性。\$RAB分配一个记录访问块(RAB),通过参数FAB建立于FAB的联系。

```

$ type jjjj.mar
.TITLE COPYFILE
$IODEF
.PSECT DATA, WRT, NOEXE
INFAB: $FAB FNM=< INFILE: >

```

```

INRAB: $RAB FAB=INFAB, -
        UBF=REC_BUFF, -
        USZ=REC_SIZE
OUTFAB: $FAB FNM=< OUTFILE:
        >, -
        MRS=REC_SIZE, -
        RAT=CR, -
        FOP=CTG
OUTRAB: $RAB FAB=OUTFAB, -
        RBF=REC_BUFF
REC_SIZE=132
REC_BUFF:
        .BLKB REC_SIZE
        .PSECT CODE, NOWRT,
        .EXE
        .ENTRY COPYFILE, ~M
        < >
$OPEN FAB=INFAB
BLBC R0, EXIT1
$CONNECT RAB=INRAB
BLBC R0, EXIT1
MOVL INFAB+FAB$L_ALQ, OUTF-
AB+FAB$L_ALQ
$CREATE FAB=OUTFAB
BLBC R0, EXIT1
$CONNECT RAB=OUTRAB
BLBC R0, EXIT1
READ: $GET RAB=INRAB
        BLBS R0, WRITE
        CMPL R0, #RMS$_EOF
        BEQL DONE
EXIT1: BRB EXIT
WRITE: MOVW INRAB+RAB$W_
        RSZ, OUTRAB+RAB$W_RSZ
        $PUT RAB=OUTRAB
        BLBC R0, EXIT
        BRB READ
DONE: $CLOSE FAB=INFAB
        $CLOSE FAB=OUTFAB
EXIT: RET
        .END COPYFILE

```

例7 MACRO调用RMS的源程序清单

一个小型汉字显示程序

西安交通大学 龙安川

我们在编制程序时, 需要进行一些汉字显示, 如汉字提示、汉字标识等, 当然我们可以使用汉字操作系统来完成, 但使用汉字操作系统有如下缺点:

1. 汉字操作系统占用内存较大, 这在计算机内存不大的情况下, 有时无法使用;

2. 汉字操作系统启动速度较慢, 等待时间较长;

3. 进入汉字操作系统后, 显示行数减少了一半, 使得一些用西文或数字进行的显示内容减少。如我们在绘制一个坐标图时, 经常需要在坐标旁标以数字刻度, 在汉字操作系统下就显得不够使用。

针对上述情况, 我编制了一个小型汉字显示程序, 它可以在 80×25 任何字符位置处显示汉字, 其占用内存很小, 使用时只需要和你的主程序链接在一起就可以了。

下面就该程序结构简要介绍一下: (见程序1)

在该程序中有一个汉字内码表, 它包含你程序中用到的所有汉字, 汉字内码表是按照从小到大的顺序排列的。另外还有一个汉字字模库, 其排列与汉字内码表一一对应。字模库最大空间为64KB, 故可存贮2k个汉字。程序包括汉字内码查找和汉字显示两部分。其中查找程序采用折半查找方式。

下面以编译BASIC为例说明一下本程序的使用方法。比如我们需要在第11行11列显示“汉字显示”四个字, 首先我们在汉字操作系统下, 在主程序中编入如下语句:

```
LOCATE 11, 11; S$ = "汉字显示";  
CALL CPRINT(S$)
```

然后把主程序编译后与汉字显示程序链接起来就可以了。运行时无需进入汉字操作系统。

从上面我们可以看出其使用方式与在汉字操作系统下是差不多的, 所以引入该程序后对主程序的编制不会带来很大的麻烦。另外, 尽管这里我们只以编译BASIC为例进行了说明, 实际上它不仅适用于编译BASIC, 还可用于FORTRAN, PASCAL, 以及C语言等, 其使用方式基本相同, 区别仅在于各自的语句格式不同而已。

在该程序建立过程中, 工作量较大的是汉字字模库的建立, 为此我还编制了一个汉字字模库建立程序(见程序2), 只需在汉字操作系统下运行该程序, 并输入你所用到的汉字, 程序会自动建立一个名为CODE.DAT的字模库, 这样在不同主程序中的汉字显示程序只需把汉字内码表及汉字字模库部分进行更换即可。

程序1

```
Para      Struc      ;interface  
savebp    dw 0  
retloff   dw 0  
retseg     dw 0  
string     dw 0  
para      ends  
,  
dtseg      segment para public 'code'
```

参考文献

- 1 王沛民《微型计算机原理及接口技术》
- 2 2000系列《FORTRAN用户手册》
- 3 2000系列《VAX—11记录管理服务参考手册》
- 4 2000系列《VAX—11C语言参考手册》

VAX/VMS的系统软接口其实非常简单, 它是通过两个分别编译的源程序共享公共变量程序段和正确传递参数实现的, 而连接程序起了协调虚存分配的作用。

```

chead dw 0BABA H ,chinese
      dw 0CABEH
      dw 0CFD4 H
      code
ctail dw 0D7D6 H
mptr  dw 0 , model
      pointer
;
dispptr dw 0 ,display pointer
;
cnow  dw 0 ,current code
slen  dw 0 ,string length
sptr  dw 0 ,string pointer
dtseg ends
;
mseg segment para public 'code '
;汉 (BABA)
model DB 020H, 000H, 010H,
      004H, 01BH, 0FEH, 018H, 00CH
      DB 041H, 00CH, 021H,
      00CH, 030H, 098H,
      032H, 098H
      DB 004H, 0D0H, 00CH,
      070H, 058H, 060H,
      038H, 070H
      DB 030H, 0D8H, 039H,
      08EH, 01BH, 007H,
      014H, 002H
;示 (CABE)
      DB 000H, 008H, 00FH,
      0FCH, 000H, 000H,
      000H, 000H
      DB 000H, 002H, 03FH,
      0FFH, 000H, 0C0H,
      004H, 0E0H
      DB 006H, 0D0H, 00CH,
      0D8H, 00CH, 0CCH,
      018H, 0CEH
      DB 018H, 0C7H, 030H,
      0C3H, 043H, 0C2H,
      000H, 080H
;显 (CFD4)
      DB 008H, 008H, 00FH,

```

```

      0FCH, 00CH, 018H,
      00CH, 018H
      DB 00FH, 0F8H, 00CH,
      018H, 00CH, 018H,
      00FH, 0F8H
      DB 00BH, 068H, 023H,
      064H, 013H, 066H,
      01BH, 06CH
      DB 01BH, 078H, 013H,
      070H, 003H, 062H,
      07FH, 0FFH
;字 (D7D6)
      DB 001H, 080H, 010H,
      0C2H, 03FH, 0FFH,
      070H, 006H
      DB 020H, 014H, 007H,
      0F8H, 000H, 030H,
      000H, 060H
      DB 000H, 0C2H, 03FH,
      0FFH, 000H, 0C0H,
      000H, 0C0H
      DB 000H, 0C0H, 000H,
      0C0H, 003H, 0C0H,
      000H, 080H
mseg ends
;
cdseg segment para public 'code '
      assume cs,cdseg, ds,
      dtseg
cprint proc far ,chinese
      print
      public cprint
      push bp
      mov bp, sp
      push ax
      push bx
      push cx
      push dx
      push ds
      mov ax, dtseg
      mov ds, ax
      mov bx, [bp] .string

```

```

mov ax, es, [bx]
mov word ptr slen, ax
mov ax, es, [bx+2]
mov word ptr sptr, ax
mov bh, 0
mov ah, 3
int 10h      ,read cursor
mov al, 160
mul dh
shl ax, 1
mov dh, 0
add ax, dx
mov word ptr dispptr, ax
dispptr=row*320+column
cp100:  cmp word ptr slen, 0
        je cpret
        mov bx, word ptr sptr
        mov ah, es, [bx]
        mov al, es, [bx+1]
        mov word ptr cnow, ax
        sub word ptr slen, 2
        add word ptr sptr, 2
        mov cx, offset chead      ,cx
                                =search head
        mov dx, offset ctail      ,dx
                                =search tail
cp150:  cmp cx, dx
        jg cp100
        mov bx, cx
        add bx, dx
        shr bx, 1      ,bx=int((cx
                                +dx)/2)
        and bl, 1111110b      ,bx=search
                                pointer
        mov ax, [bx]
        cmp ax, word ptr cnow
        jg cp200
        je cp500
        add bx, 2
        mov cx, bx      ,cx=bx+2
        jmp cp150
cp200:  sub bx, 2
        mov dx, bx      ,dx=bx-2

        jmp cp150
cp500:  sub bx, offset chead
        mov ax, 16
        mul bx
        add ax, offset model
        mov word ptr mptr, ax
        ,mptr=16*cptr
        call graph ,display chinese
        add word ptr dispptr, 2
        jmp cp100
cpret:  pop ds
        pop dx
        pop cx
        pop bx
        pop ax
        pop bp
        ret 2
cprint  endp
,
graph  proc near ,display chinese
        push ax
        push cx
        push si      ,ds, si=model
                                pointer
        push di      ,es, di=display
                                buffer pointer
        push ds
        push es
        mov ax, word ptr dispptr
        mov di, ax
        mov ax, word ptr mptr
        mov si, ax
        mov ax, 0b800h
        mov es, ax
        mov ax, mseg
        mov ds, ax
        mov cx, 8
        cld
g100:   lodsw
        mov es [di] , ax
        lodsw
        mov es, [di+2000h] ax
        add di, 80,

```

管理信息系统开发过程中的 快速软件开发工具

天津电子计算机研究所 曲廉生

摘要 本文讨论了管理信息系统开发过程的方法及步骤,介绍了一种方便实用的软件开发工具,最后通过通用菜单程序的设计,分析了这套自动生成器的设计思想和实现方法,并给出了它的程序清单。使用这套开发工具,不仅可缩短软件开发周期,降低软件开发成本,而且还可改善软件质量,提高软件生产率。

一、管理信息系统的开发过程

建立一个MIS(管理信息系统),按软件工程方法实施,通常分为六个阶段:1.系统调查;2.系统分析;3.系统设计;4.程序设计;5.系统调试;6.系统维护。

系统分析阶段比较常用的工具是数据流程图,用它描述设计系统总体逻辑模型,和下属各层的逻辑结构。但它无法详细表达一个系统的全部处理逻辑及数据的详细内容。所以还需使用数据字典和逻辑加工说明两个辅助工具来弥补数据流程图的不足。最低一层的数据流程图中的各个处理逻辑,可采用结构式语言描述,因为结构式语言的可读性和可修改性好,有利于程序设计阶段实现编程自动化。

系统设计阶段常用的工具结构图,它可表达已被分解成由若干模块组成的系统结构,以及这些模块之间的接口。结构系统设计是把一个系统设计成具有层次式的模块化结构,使模块内部的各个组成部分的处理动

作,执行一个而且只执行一个功能,因为这样的功能组合的内部紧凑性最优,与其它模块的联接形式最好,同时模块的可修改性最高。

程序设计阶段的任务是将结构图中的每一模块变成一个程序。采用结构程序设计方法编制的源程序具有很好的可读性和可修改性,有利于系统调试和系统维护阶段程序查错和调试。在程序设计时要尽量采用自动化的MIS开发工具,实现计算机自动或辅助设计,不仅对改善软件质量和提高软件生产率有重要意义,而且还可缩短开发周期,使系统迅速投入运行和尽快地产生经济效益。

二、方便实用的软件开发工具

由于程序处理的对象是数据,为了提高软件的通用性,需要将数据和程序分离,以保持程序与数据的相对独立性,一般可采用两种办法:一是程序生成器,针对某一类问题或某类报表的处理,用程序的方法,为各种功能模块生成相应的功能程序。二是通用程序模块加数据词典法,针对某一大类处理对

```
loop g100
pop es
pop ds
pop di
pop si
pop cx
pop ax
```

```
ret
graph endp
cdseg ends
end
```

程序2:

```
10 DIM C%(2)
20 SCREEN 2
```

下转(第48页)

象编写一套通用程序,使每一功能模块使用一个通用程序,而通用程序处理具体问题的数据形式或报表格式,则由一数据词典的生成子系统生成。由于数据词典的结构是不变的,从而保证了系统的通用性,而其内容是可变的,从而增强了系统的适应性。

根据上述原则,针对一般管理信息系统的常用功能,我们研制了一套软件开发工具,其中包括:1.数据字典自动生成器;2.数据库自动生成器;3.程序结构图自动生成器;4.菜单自动生成器;5.录入、校验、修改自动生成器;6.查询自动生成器;7.任意多层表头报表打印自动生成器;8.统计、汇总自动生成器;9.报表纵向调正、横向调正、纵横调正自动生成器;10.图形自动生成器;11.通用汉字代码对照程序;12.通用复制备份软盘程序;13.通用加密程序;14.通用接口程序。

应用这套辅助编程工具,在系统分析阶段可自动建立数据字典和自动建立数据库。在程序设计阶段可自动生成菜单、数据录入、计算、校验、修改、查询、统计、汇总、调正、制表打印;自动做程序结构图、直方图;查找汉字代码和复制备份文件等而无需编程。需要编程的只是诸如辅助计划、预测决策等较特殊的业务处理模块。这样一来必然会节省编程时间,缩短系统开发周期,促使新系统提早投入运行和启用。

三、自动生成器设计举例

在一个复杂的系统中,往往会有成十上百个菜单,但这些菜单程序的功能基本相同,所以可以编写一个通用程序来显示和选择系统中的所有菜单。

首先利用自动建菜单库程序ZDJCDK·PRG建立一个菜单库,然后利用通用菜单程序ZDJCD·PRG把一个系统的全部菜单功能模块连接起来,显示系统中所有的菜单,当用户需要重新建菜单或修改已建的菜单时,只需新建或维护这个菜单库,而不必修改程

序。程序清单见附录。下面对其设计思想予以简介。

在foxbase+ (或dBASE III+、或foxpro)数据库环境下,用copy structure EXTENDED命令把一个库结构拷贝到一个结构描述库中,于是原数据库的结构便转化成结构库的记录,用APPEND BLANK命令给结构库追加功能名称 GNMC、功能序号 GNXH、执行命令 ZXML和标志BZ等四个记录,随后将结构中转化来的头4个记录删去,通过CREATE FROM命令,用修改后的结构库生成新建的菜单库。此后以12为循环条件自动把功能序号添入菜单库中,最后通过全屏幕窗口编辑命令 BROWSE,键入相应的功能名称、执行命令和标志。

分析菜单结构,若以其第一层为右子树,以其下一层的功能名称或执行程序为左子树,则可以将菜单构成一个二叉树。我们用二叉树结构来建造菜单库:库中的GNXH字段为0时,ZXML存放的是上层菜单的记录指针。GNXH不为0时,若ZXML存放的是下层菜单的记录指针,则继续调用下层菜单的内容;若选取的不再是菜单而是菜单叶子时,ZXML存放选中该项所应执行的命令,以调用指定的功能模块。当ZXML为上层或下层记录指针时,BZ为2;当ZXML为执行命令或0时,BZ为1;当用户选择菜单的路径上设有密码时,BZ为3,此时便可自动询问和检查密码,若键入的口令三次出错,则自动退出系统,以确保系统的安全性。

为了使屏幕上显示的功能序号、功能名称排列的整齐美观,通用菜单程序把它们排列成5行2列。2列的位置分别用表达式 $2X-1$ 、 $2x$ 表示。它们随着行数 x 的变化,在屏幕的相应行和列上顺序显示出来,并利用屏幕划线命令@ TO 给各层菜单划出大小始终不变的周边框线。工作屏幕最后一行始终做功能选择区且颜色保持不变。使用区域置色命令SET COLOR,用不同颜色标记不同层次的菜单,可以达到较理想的屏幕效

果。

菜单的功能可按数字键或字母键选择,也可用光标来控制选择。若利用刚键入的键函数INKEY(),可以使全屏幕功能键让光标上、下、左、右移动,当选中菜单中某一条时,只要将光标停在该条上,按回车键即可执行,这样更加直观、灵便。

四、结束语

我们研制的这套简便易用的管理应用软件,为篇幅所限,在此就不再一一介绍了。实践证明,这些辅助编程工具,既可协助用户实现快速自动编程,又可增进系统对改革和发展的应变能力,很适合我国目前开发MIS的需要。

附录1.通用菜单程序

```
type zdjcd.prg
CLEAR ALL
SET SAFE OFF
SET EXAC OFF
SET TALK OFF
CLEAR
SELE 9
USE CDK
MBZ=BZ
DO WHILE.T.
GO INT (RECNO () /12) * 12+1
CLEAR
STOR 1 TO X, H
STOR 0 TO K
STOR 2 TO L
@ 2, 1 TO 8, 79 DOUBLE
@ X+1, 32 SAY trim (GNMC)
SKIP
DO WHILE x<=5..and.zxm1# '0'
IF H=2*X-1
@ X+2, 6 SAY H PICT '9'
@ X+2, 9 SAY GNMC
h=h+1
ELSE
IF H=2*X
```

```
@ X+2, 42 SAY H PICT '9'
@ X+2, 45 SAY GNMC
h=h+1
x=x+1
ENDIF
ENDIF
SKIP
ENDDO
GNH=0
@ 9, 28 SAY '请选择 (0键退出) '
@ 9, 45 GET GNH PICT '9'
READ
I=0
DO WHILE I<11
IF GNXH=GNH
ZXL=ZXML
MBZ=BZ
IF MBZ='2'
K=K+1
IF K=2
L=0
ELSE
L=2
ENDIF
SET COLOR ON
SET COLOR TO K+2
GO VAL (ZXL)
EXIT
ENDIF
stor 0 to f
IF MBZ='3'
DO WHILE F<4
IF F>3
RETU
ENDIF
CLEA
@3, 10 SAY '请键入口令:'
set console off
acce to 口令
SET CONSOLE ON
IF 口令<'>' 'XG'
? '口令错,按任意键,重来:'
WAIT
```

```

LOOP
ELSE
EXIT
ENDIF
F=F+1
ENDDO
GO VAL (ZXL)
EXIT
ENDIF
IF ZX L= "RETU "
CLEAR
@3, 20 "SAY "本系统运行结束 再见 "
WAIT " "
CLEAR ALL
RETU
ENDIF
JLH=RECNO ()
IF ZXL= "1 "
EXIT
ENDIF
&ZXL
IF .NOT. MBZ= "1 "
&MBZ
ENDIF
SELE 9
USE CDK
GO JLH
EXIT
ENDIF
SKIP-1
I=I+1
ENDDO
SELE 9
SET COLOR TO
ENDDO

```

附录2. 自动建菜单库程序

```

C> type zdjcdk.prg
set talk off
set safe off
clear
stor spac (6) to c
@row () +1, 5 say "输入库名: "
@row () , 15 get c

```

```

read
stor trim (C) + ".dbf" to d
use &d
copy to jgk stru exte
use
use jgk
appe blan
repl field_name with "gnmc"
repl field_type with "C"
repl field_len with 20
appe blan
repl field_name with "gnxh"
repl field_type with "n"
repl field_len with 2
appe blank
repl field_name with "zxml"
repl field_type with "c"
repl field_len with 10
appe blan
repl field_name with "bz"
repl field_type with "c"
repl field_len with 10
go top
dele record 1
dele next 4
pack
crea jcdk from jgk
use
stor space (6) to k
acce "输入新建菜单库名: " to 1
k=trim (1) + ".dbf"
copy file jcdk.dbf to &k
use
use &k
input "输入欲建功能个数" to n
j=0
do while j<=n
j=j+1
i=0
stor 0 to m
do while i<=11
i=i+1
mlen=i

```

```

loca for gnxx=i
  appe blan
  repl gnxx with i
  stor m+1 to m
enddo
skip
  enddo
go top
do while .not.eof ()
  loca for gnxx=11
  repl gnxx with 0
  cont
enddo
go top
  do while .not.eof ()
  loca for gnxx=12
  repl gnxx with 0
skip
enddo
@9, 0
@9, 0 say '键入显示功能名称, 键毕敲
      ^_W '

  appe blan
browse fields gnmcc lock 1 freeze gnmcc
@9, 0
@9, 0 say '键入功能序号, 键毕敲 ^_w '
go top
browse fields gnmcc, gnxx lock 1 free-
      ze gnxx

@9, 0
@9, 0 say '键入执行命令, 键毕敲 ^_W '
go top
browse fields gnmcc, zxml lock 1
      freeze zxml

@9, 0
@9, 0 say '键入标志, 键毕敲 ^_W '

```

```

browse fields gnmcc, bz lock 1 freeze
      bz

@9, 0
use
acce '要打印吗? (Y/N) 'TO DY
USE &k
set print on
if dy = 'y ' .or. dy = 'Y '
list STRU to print
list to print
use
endif
set print off
sele 1
use
sele 2
use
erase jgmcc.dbf
erase zzk.dbf
erase xkl.dbf
close data
return

```

附录3.菜单库的结构和内容

```

.use cdk
.list stru
Structure for database: C:\cdk.dbf
Number of data records:      14
Date of last update      : 01/01/80
Field Field Name Type      width
Dec
  1  GNMC      Character    20
  2  GNXX      Numeric      1
  3  ZXML      Character    10
  4  BZ        Character    10
. . Total . .
.list

```

基于Turbo Prolog的CAI软件 开发技术

连云港职业大学

何惠治

摘要 本文在论述了CAI的特点和国内外发展动向的基础上,着重介绍了适合于高等理工科、师范、大专类学校,基础有机化学桥环烃结构与命名教学软件《YJHT》的充满活力的设计方法和采用Turbo prolog语言实现本系统程序设计的开发技术。

一、引言

随着我国教育的不断深入与发展,计算机辅助教学CAI系统,近年来也正在我国进一步兴起。一个好的CAI系统,完全可以把一位有丰富教学经验的教师的教学思想,充满活力的教学内容和富于启发式的灵活的教学方法和策略;付诸实施,前景广阔。

国际上CAI虽是从五十年代开始起步,但直到七十年代微机出现后才有较迅速的发展。

目前已有充分的数据证明CAI比传统的教学方式优越得多。第一,不受时间和课堂的限制,也可因材施教。学生可以完全根据自己的知识功底和智能情况选择最合适的内容和进度进行学习。第二,CAI可以充分发挥计算机仿真、绘图与音乐等功能,达到最佳的学习效果。第三,CAI可以给出学生最真实的学习成绩的记录,通过对学生的学习

成绩和对全部试题正确率及误差率的分析,获得修改教材和试题相应内容的必要依据,同时还可以实现教材版本的更新。

二、《YJHT》CAI 软件的系统配置

计算机辅助教学CAI,已成必然的发展方向。国际上PC机已被选为开发智能软件的机型。近年来国内各大专院校,企事业单位IBM-PC/XT机及其兼容机的普及率相当迅速,因此《YJHT》CAI软件在IBM-PC/XT机上研制是合适的。

Turbo Prolog语言编译系统是由我国南京大学1987年汉化成功并推出的。它是一种全新的陈述式语言。且语法非常简单,易于实现模块化程序设计,程序的可读性好。特别是它独具风格又彩色斑斓的多窗口功能和简便的绘图与音乐功能,可使CAI软件锦上添花,让学生感到妙趣横生,学而不厌,故本系统采用Turbo Prolog语言编程。

Record#	GNMC	GNXH	ZXML	BZ	Record#	GNMC	GNXH	ZXML	BZ
1	综合计划生产系统	0	retu	1	8	值台帐	8	76	2
2	工业计划	1	12	2	9	进度表	9	82	2
3	商业计划	2	24	2	10	•	0	0	1
4	商业统计	4	36	2	11	•	0	0	1
5	购销登统	5	48	2	12	工业计划	0	1	2
6	在途登统	6	52	2	13	录入	1	do jt01	1
7	量台帐	7	64	2	14	查询	2	do jt02	1

三、《YJHT》CAI 软件的设计技术

《YJHT》CAI软件是为完成高等理工科、师范、大专等类学校基础有机化学桥环烃结构与命名的教学任务而设计的。下面拟就四个方面加以论述。

1. 《YJHT》CAI软件的总体设计目标

(1) 系统给出二环桥环烃命名简则和示范例题后, 相继给出一组10题二环桥环烃的分子结构图形和每题的四个可选择的命名答案。学生只须选按1—4中的数码键来回答问题, 如果按错, 系统提出忠告。当首次即选中正确答案时称为正选, 得10分; 而误选时系统给出必要的提示, 经一次提示后再做对得5分, 其余为0分。每当做完一题后屏幕的底行给出正确答案的说明, 并对不同得分的情况给予不同的评价和不同的激励语, 并发出不同的音乐声以示区别, 同时在屏幕的右上角用鲜明的彩色窗口显示累计得分和累计时间。按任意键后系统把有关数据记入动态数据库中, 然后继续出题, 直到10个题全部完成后, 以'*'标记返回主菜单。

(2) 与1方式相同, 学生可通过多环桥环烃命名简则和例题的学习后, 完成另一组10题多环桥环烃的结构—命名练习。

(3) 系统通过自动回溯从动态数据库中学生的三组数据: 班级代号, 学号和日期以及二组累计得分和累计时间, 检索出来, 并通过成绩等级评定程序得出: 优秀、良好, 中等, 及格和不及格的结论, 然后打印输出一份每题的得分和总和以及综合评定的成绩单。

(4) 系统应在核对教师代号密码后, 允许教师对学生的成绩记录做正选率和误选率的统计分析, 其分析结果打印输出, 给教师修改试题提供依据。

2. 《YJHT》程序结构流程图

《YJHT》程序结构流程图如图1所示。

图中各个部分是按系统所要完成的总体教学目标而设计的。操作十分简便, 图、文、声并茂的教学方式可使学生轻松愉快地完成学习任务。

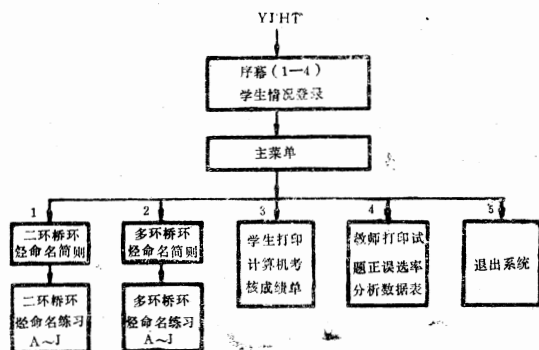


图1 YJHT程序结构流程图

3. Turbo Prolog模块化设计技术

(1) 由于CCDOS和语言的编译系统各约占250KB内存, 因而在CCDOS支持下编写一个大型实用程序, 必需按每次可编译的源程序的大小, 将其合理的划分成若干模块, 每个模块可以分别编辑和编译, 然后将它们连接成一个可执行的程序。有两种方法可实现上述连接: 1) 对于在文本方式下编辑的带汉字的源程序, 需在生成各自的目标模块后退出Turbo Prolog系统, 直接在CCDOS下将它们连接成可执行程序。2) 对于在图形方式下编辑的带汉字的源程序, 可直接在DOS支持下, 利用西文Turbo Prolog自身的〈工程〉模式自动连接成功可执行的程序。本系统就是采用2)的方法实现的。YJHT.EXE程序达220KB以上。

(2) 工程说明和全局说明

Turbo prolog 为了实现模块化程序设计, 引出了工程说明和全局说明二个概念, 分述如下。

一个实用程序的所有模块应统称为同一个工程名, 并必需在每个模块开头用 Project “工程名” 加以说明。为了实现模块的连接还需建立一个包含该工程所有模块名的模块表文件。例如本系统的工程名为 YJHT, 且由十个模块组成, 故模块表 YJHT.P-

rJ文件的具体内容为:

```
general+HT1+HT2+HT3+HT4+HT5+HT6+PRINT+PCMCR+THE-end
```

其中general模块由各模块所共用的子句组成,这样做有利于模块的调试和连接。HT1是主模块,它是整个程序唯一包括目标的模块,程序中所有的动态数据库文件都必需在该模块中用标准谓词Consult“文件名.dba”调入内存。

为了实现模块间有参或无参谓词的交互,最好的一个办法是,把参与模块间交互的谓词称为全局谓词,并把所有的全局谓词和其领域说明以及动态数据库说明,建立一个独立的全局说明文件中。值得一提的是,全局谓词的说明必须指出其参量的模式流。本系统的全局说明文件为global.pro,其文件内容大体如下:

```
global domains
    direction=up, down
    List=integer*
    Cn, sn, date=string
data base
    bb_sum(integer, integer, integer)
    pb_sum(integer, integer, integer)
    student(Cn, Sn, date)
    bba(integer, integer, integer, integer, integer)
    :
    :
global predicates
    main-menu(char, char) - (i, i)
    repeat
    delay
    music(direction, integer) - (i, i)
    interlude(integer, List) - (i, 0)
    show(integer, integer, integer) - (i, i, i)
    test(integer, integer, string, str-
```

```
ing, ) - (i, i, 0, 0)
    :
    :
    pb_brief(integer) - (i)
    print
    pcmcr
    the_end
    :
    :
```

为使该文件能为本工程的所有模块所共享,每个模块的开头必须用编译命令include“global.pro”把全局文件包括进去,才能顺利通过编译。

(3) 模块的程序结构

Turbo Prolog语言为自顶向下逐步求精的程序设计与调试创造了非常有利的条件。关于模块的程序结构问题,准备谈两点:

1) 通常的程序结构

为了提高程序设计和调试的速度, Turbo Prolog允许把一个模块再分成若干个子模块,当第一个子模块调试成功后,可紧接着编辑第二个子模块的领域,谓词说明和子句进行调试等等。也就是说,一道程序,除了目标段是唯一的以外,其他如领域段、谓词段和子句段均可多次重复使用,故一般的程序结构非常灵活。这种积木式的搭接法给程序设计和调试均带来极大的方便。

2) 主模块和其他模块的程序结构

经验表明,如果把模块的程序结构分为调试结构和连接结构,对于模块的调试和连接工作带来方便。

由于Turbo Prolog只允许主模块中包括目标段和调入数据库的事实,故主模块和其他模块的程序结构有所不同。图2和图3分别给出了主模块和其他模块的程序结构实例。

```
include "global.pro"
Predicates
    :
    goal
```

```
Consult( "YJHT.dba" ), Prelude.
include "general.pro"
clauses
prelude: —
:
```

图2-a 主模块的调试结构

```
project "YJHT"
include "global.pro"
pred: cates
:
goal
consult( "YJHT.dba" ), prelude
clauses
prelude: —
```

图2-b 主模块的连接结构

```
include "global.pro"
predicates
:
goal
bb_c( 0, 0, 0 ).
include "general.pro"
clauses
bb_c( SG, SM, SS ): —
:
```

图3-a 一般模块的调试结构

```
project "YJHT"
include "global.pro"
predicates
:
clauses
bb_c( SG, SM, SS ): —
:
```

图3-b 一般模块的连接结构

可以看出, 连接结构中的第一行必须是工程说明。由于本系统的通用子句文件将作为一个独立的模块被连接到执行文件中, 故编译命令: include "general.pro" 从连接结构中删去了。

4. 特殊设计技术的探讨

(1) 恰当利用动态数据库

Turbo Prolog提供了一整套动态数据库操作和管理的内部谓词。经验表明, 恰当而又巧妙地设计动态数据库是完成一个好的程序设计的关键之一。

1) 中间结果数据的处理

例如本系统中, 当学生学习完命名简则后, 开始相继作某一练习题时, 由于回答错, 系统提示学生按任意键返回去再学习命名简则, 然后再重作该题, 此刻系统必须把当前该学生的累计得分和累计时间以及正误选题情况等中间结果, 存贮在动态数据库中, 以备再次学习简则后, 重作该题前将这些中间结果数据从数据库中取出, 使系统能正确运行下去。同时这些中间结果数据, 在完成任务后应及时从动态数据库中删除。

2) 数据库数据项的扩充与完善

如果一个最终应保存在磁盘文件中或打印输出的数据项, 是随着程序的运行在不同模块中所获取的数据项的总和的话, 那么就必须建立多重初始动态数据库, 分别完成数据项的采集, 最后利用Turbo Prolog合一算法把各个数据库中的数据项分别取出来, 并一起存入最终形式的动态数据库中, 完成数据项的扩充和完善。

3) 磁盘数据库的应用

Turbo Prolog把关系数据库描述为一组事实, 且是程序的一个组成部分。如果数据库是基于内存的, 而且数据量很大, 则势必受到内存空间的限制。例如本系统为了给教师提供正误选率分析, 必须保留每个学生作20个题目的正误选情况记录, 若保留30个学生的记录就有600条事实, 尽管这些事实教师在教师进行正误选率分析过后可以从内存中删除, 但分析之前毕竟对内存产生威胁。解决的办法之一是建立文件型数据库, 把数据存入磁盘, 然后根据内存可接受的数量, 进行分批统计分析处理。

(2) 几种溢出问题的解决

调试程序时, 常常会遇到以下三种情况, 分别叙述如下。

1) 如果系统给出代码溢出或代码数组太小的提示后, 一般情况下只要利用编译命令code=新段数值(系统的默认值为1024段)来增加一定段数后, 即可得到解决。应当指出的是, 该编译命令在连接模块中必须插在project“工程名”和include“global.pro”之间。否则出错。

2) 如果系统给出堆栈溢出内存不够或循环不止的提示时, 经查实无死循环存在, 解决的办法之一, 必须把该模块设法变小才能得到解决。但如果该模块不便改小, 那么就需要把汉字改为画图方式下输出, 而模块的调试和连接均要在DOS与西文Turbo prolog支持下进行, 也能妥善解决。

3) 如果系统给出堆栈溢出内存不够或循环不终止的提示时, 应根据程序的具体问题采用不同的解决办法。一般情况下可以首先利用编译命令Check deter m放在模块的开头进行模块的调试, 然后在系统指出不确定子句的地方加上截断元素!来节省堆栈空间。假若某个子句中子目标太多也会引起堆栈溢出, 这就需要增设谓词, 将该子句分成多个子句加以解决。如果某个子句中包含一系列具有递归算法或者具有标准谓词findall以进行对数据库回溯查找的子目标时, 很容易引起堆栈溢出。用变体子句代替这些子目标, 同时每个变体子句后面加上恒失败谓词fail, 使其谓词调用失败, 这样做既可以连续调用其变体子句来完成程序的任务, 且每个子句的失败还可将堆栈存贮释放出来。

4) 要特别注意系统的输出窗口应统筹考虑。每个模块中的窗口当使用完后要及时用removewindow加以去掉。否则很可能虽然

单个模块调试已经成功, 但连接后的执行文件并不能运行。特别遗憾的是, 系统往往无法提供窗口太多的错误信息, 来解除设计者心中的疑点。

(3) 实现上脚字的输出

本系统在多环桥环烃命名练习中, 利用右上脚字标明某桥与主环相连的碳原子位号, 例如: 三环[4.4.1.1^{1'} 5]十三烷或者: 五环[4.2.0.0^{2'} 5.0^{3'} 5.0^{4'} 7]辛烷

解决上脚字输出的一个最方便的办法是, 利用Turbo prolog的作画命令, 编制一个包括逗点在内的数字库, 只要用含有坐标数据参量的数字谓词调用该库即可完成上脚字的输出。

四、结论

采用Turbo Prolog语言编写CAI软件是比较理想的。系统完成后, 经化学专业的教师和学生试用, 认为图、文、声并茂的本系统软件形式非常生动活泼, 实现了使学生可以在轻松愉快中完成学习任务。

在此对编写脚本的连云港教育学院沈开惠副教授和对脚本给予审核的南京大学段康宁教授表示感谢。

参考文献

- 1 纪有奎、等编, PROLOG语言教程、海洋 87
- 2 李卫华、周祥和、刘灼光编译, IBM-PC机编译型PROLOG语言、武大 87
- 3 陈兆乾、潘金贵、谢俊元编译, TURBO-PROLOG程序设计(上、下册)南大 87

微机在高校学生综合测评工作中的应用

天津职业大学 张克善 张品福

摘要 本文介绍了在PC机上如何利用dBASE III数据库管理系统,对高校学生德、智、体综合测评工作所进行的开发与应用。文中进行了系统分析,并阐述使用国际新型的PAD图实现系统主要功能模块的方法。归纳了此项课题研究工作的实际意义。

随着教育的不断深入,在高等院校建立起科学的、规范化的学生德、智、体综合考评制度,已是形势发展的必然趋势。是培养学生成为四化建设人才的一项有力措施。是新时期大学生思想政治工作一条新途径。

一、综合测评目的和方法

综合测评工作,不仅对高校学生的日常管理及在校期间评定三好、党团组织的发展建设、毕业生择优推荐等提供了确切的依据,同时为高教工作各级领导掌握学生综合情况而提供可靠的信息。此外还可以激励学生的进取精神与学习热情,使学生能够认识自我、完善自我,积极地完成自我设计。

二、系统说明和系统建立

综合测评是在对学生德、智、体诸方面素质进行全面考察的基础上,采用定性分析和定量计算相结合的办法来进行的。测评工作以学生班为单位,每学期评定一次。德育素质包括政治思想、道德品质、组织纪律等项,德育素质分占德智体总分的30%。智育素质包括各科学成绩和学习能力两个方面,分别占总分的50%及10%。体育素质为体育成绩、运动成绩、健康状况,其体育素质分占总分的10%。另外德、智、体三方面均可按突出成绩者酌情加分,及严重错误者酌情减分的原则打附加分。

对于专科2~3年学制的学生,综合测评管理系统要反映包括学生自然状况的诸方面

数据几十项,混在一起会给输入数据、后面的查询工作带来许多不便。因此数据库建库时,首先考虑有利于系统的功能实现,尽量分库操作和数据共享,以节省内存及提高速度。

本德智体综合测评管理系统(简称DZ-TCP)分成两类库,即德育、体育素质库(DT.DBF),智育素质库(ZY.DBF)。两个库之间用相同关键字,编号或姓名建立联系,同时打开两个库就能反映一个学生全部情况,为统计和打印报表提供方便。

三、系统实现

1. PAD图特点及原则

使用一般流程图,其主要缺点是把系统处理过程和处理顺序混杂在一起,使得一些较复杂的系统流程图显得繁琐,逻辑关系不清晰。为了直观地表达系统的逻辑关系,近年来国际上使用一种新的流程图——PAD(Problem Analysis Diagram)。它由时间和过程二维结构组成。将处理过程和处理时序作为两个坐标轴,纵坐标是一个时间(时序)轴,横坐标是一个过程(层次)轴。PAD图能准确地描述原程序框图的处理过程和处理时序。

由于它能极其逼真地刻划dBASE III处理问题的过程,对于只采用顺序、判断、循环和过程四种基本结构的模块化程序。PAD图与命令文件存在一一对应的关系。通过对PAD图工程化遍历处理,即对所有框从上到下,从左到右的先后顺序访问一遍,即可得到命令文件。

工程化遍历处理原则:

(1) 每访问一个处理框就把对应语句写下来。

(2) 第一次访问某循环框时, 写上 DO WHILE<条件>, 再次访问就写上 ENDDO。

(3) 第一次访问某简单判断框时, 写上 IF<条件>, 再访问写上 ENDIF。

(4) 第一次访问某选择判断框时, 写上 IF<条件>。第二次访问时写上 ELSE, 第三次访问时写上 ENDIF。

(5) 多重选择判断框写上 DO CASE。

经过以上的遍历处理, PAD图就快速准确地写成命令文件, 为编写和修改程序带来方便。

2. 系统主控模块 (DZTCP.PRG)

德智体综合测评管理系统 (DZTCP.PRG), 包括数据库维护功能模块 (CHANG.PRG)、查询功能模块 (LOCON.PRG)、统计功能模块 (TOTAL.PRG)、及打印报表功能模块 (REPO.PRG)。系统模块结构如图1所示。系统主控模块 PAD图如图2所示, 图中虚线箭头表示 PAD图工程遍历顺序。

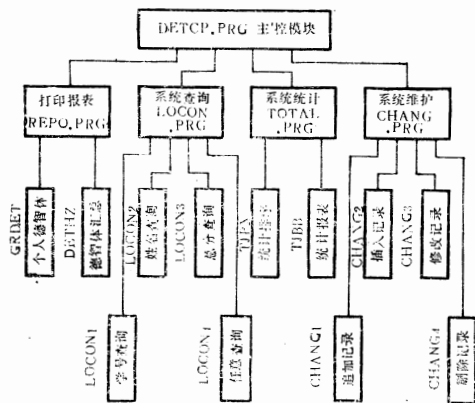


图1. 系统模块结构图

3. 系统主要功能模块

(1) 系统库维护功能模块 (CHANG.PRG)

本系统库维护功能模块中设计了追加输入、中间插入、修改记录及删除记录四种功能, 可以对数据库进行全面的维护操作。本功能模块提供不依赖全屏幕操作, 又能全覆

盖各种维护功能的方法。为了保证数据库使用中的安全保密, 使用了两级口令。将用户分为两个级别, 第一次只允许使用其他三个功能模块, 即只能读取数据库的内容。而在数据库维护模块前设置了第二道口令, 方能完成改写数据库的操作。图3 所示系统库维护模块PAD图。

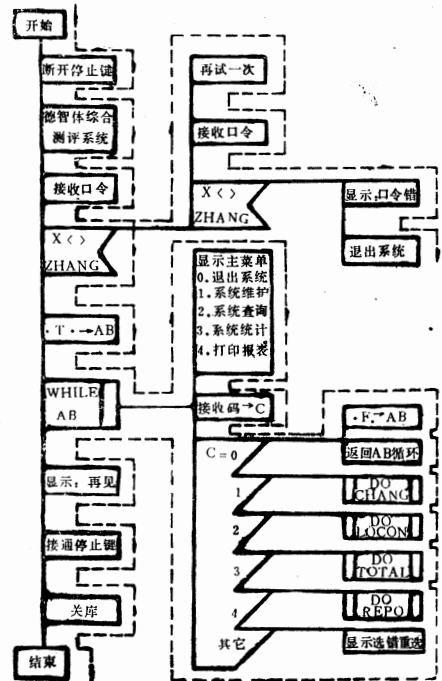


图2. 系统主控模块 (DZTCP.PRG) PAD图

(2) 查询功能模块 (LOCON.PRG)

查询工作是系统管理中一项重要功能, 本查询功能模块向用户提供了三类操作方式。按学号、按姓名的查询、按德智体总分查询及自由指定字段、任意输入条件及随意地选择字段名清单的任意查询方式。图4 所示查询功能模块PAD图。

(3) 统计功能模块 (TOTAL.PRG)

此模块功能可统计各班德智体指标中分数线档次所占人数的比例, 统计最高分数、最低分数及名次的排列情况, 利于班级内的评比。统计功能模块PAD图如图5 所示。

(4) 打印报表功能模块 (REPO.PRG)

打印报表功能模块分为德智体汇总情况

报表(DZTHZ·PRG)及个人德智体报表 (GRDZT·PRG), 按学号打印出每个学生德、智、体分类项数据、总分并计算出各分类项及总分的平均分数, 使用于各专业、班级间评比工作等。图6所示个人德智体报表 PAD图。

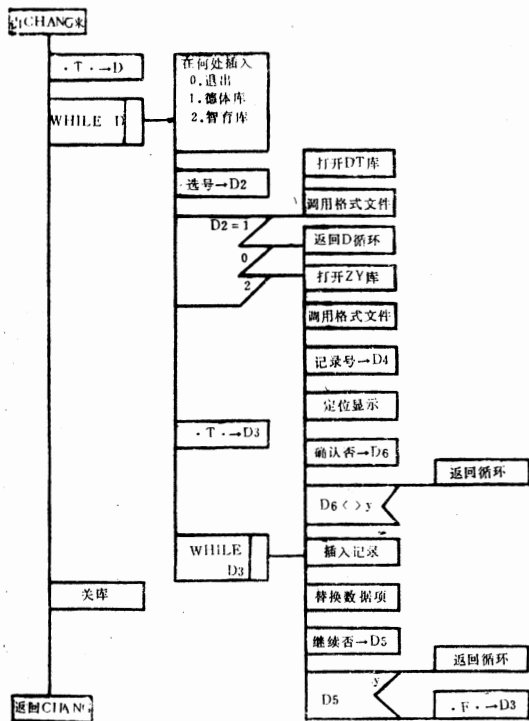


图3 系统库维护模块 (LOCON₂·PRG) PAD图

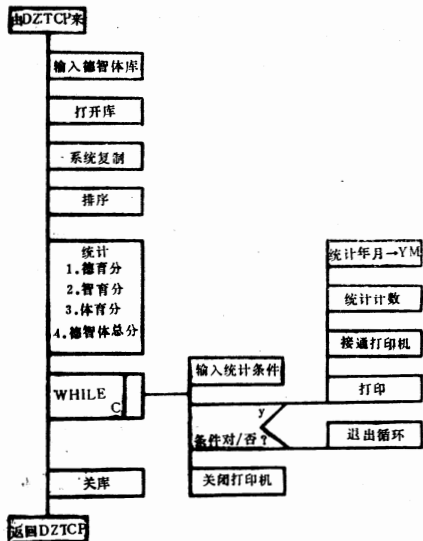


图5 统计模块 (TOTAL·PRG) PAD图

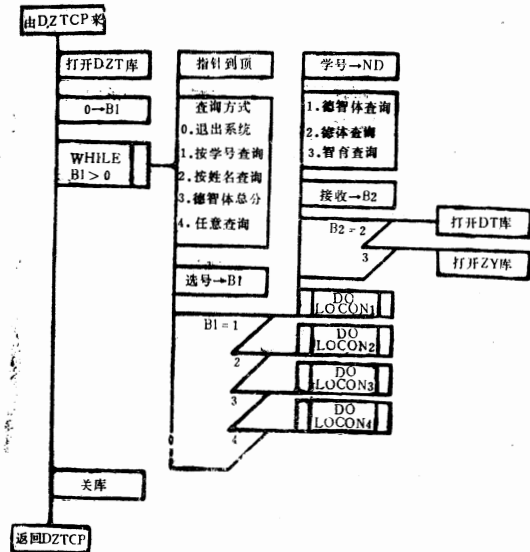


图4 查询模块 (LOCON·PRG) PAD图

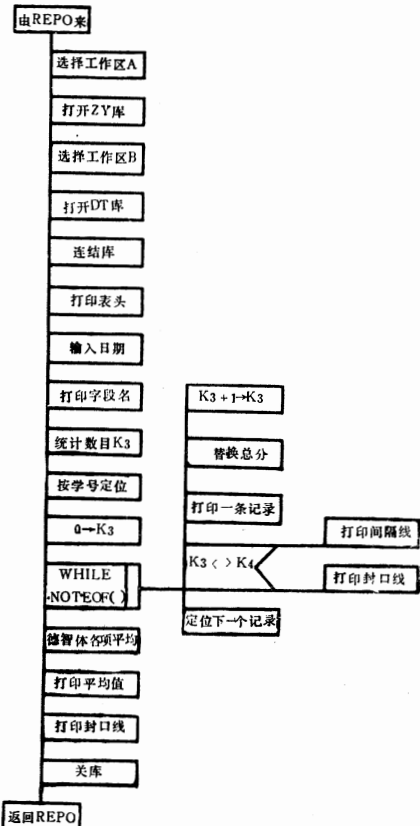


图6 个人德智体模块 (GRDZT·PRG) PAD图

本系统的开发和运行, 大大减轻了各级
40

IBM PC机异步通信适配器(RS—232口)

基本程序及其使用方法

阜新矿业学院

苗长云

阜新矿物局职工大学 杨素君

摘要 本文分析了 IBM PC 及其兼容机的异步通信适配器基本程序, 并阐述了利用该程序编制与外部串行设备通信软件的四个主要部分的方法。

一、引言

由于 IBM PC 及其兼容机(以下简称 PC 机), 不仅结构先进、性能价格比高、有丰富的系统软件和应用软件, 而且还有较强的技术和经营后盾等优点, 因而得到了广泛的应用。在国内, 它已受到广大用户的欢迎, 成为微机的主流。随着 PC 机应用领域的不断扩大, 它与外部串行设备的连接几乎成为不可缺少的内容。异步通信适配器是 PC 机与外部串行设备连接的关键部件, 它是完全可编程的, 通过编程, 可选择不同的通信方式。由于异步通信适配器接口板是现成的, 故实现与外部串行设备通信的主要工作是编程。采用汇编语言直接编程时, 需全面了解接口板, 并应按要求编写, 这种编写方法复杂, 易出错。PC 机 ROM 驻留了 ROM Bios 例行程序, 其中包含异步通信适配器基本程序。应用它编制与外部串行设备的通信软件则简便有效。本文对此进行了全面分析, 并介绍了用它编制通信软件的方法。

二、异步通信适配器的基本程序分析

PC 机的 ROM Bios 中的异步通信适配器基本程序已添入 PC 机的中断向量表, 此程序用软件中断 INT 14H 来调用。该程序框图如图 1 所示。它可使用两块异步通信适配器插件板, 分别为 0 号和 1 号, 并由 DX 寄存器选

择, 若 DX 寄存器为 0, 选择 0 号插件板; DX 寄存器为 1, 选择 1 号插件板。一般, 只安装一块异步通信适配器板, 即 0 号插件板, DX 寄存器为 0。异步通信适配器基本程序有四个主要部分, ①初始化通信口; ②发送数据; ③接收数据; ④测试通信口状态。下面分述其功能和执行过程。

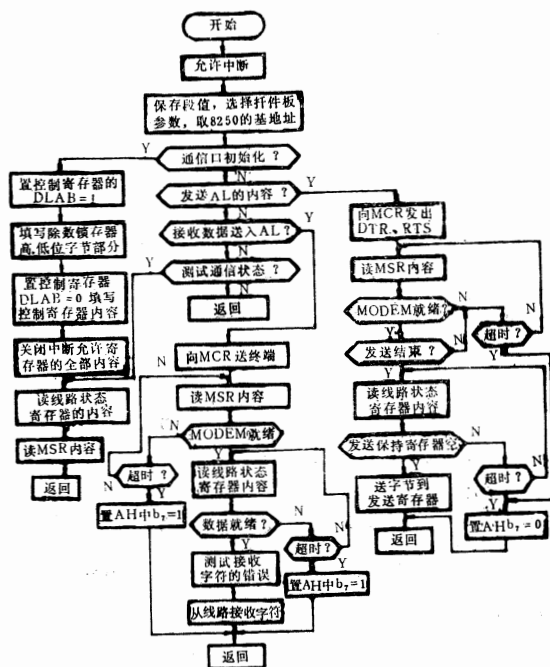


图 1 异步通信适配器基本程序原理框图

1. 初始化通信口

功能: 按所选定的通信参数设定波特率、奇偶性、停止位及字节长度。

执行步骤:

(1)CPU按照选定的波特率参数添写除数锁存器的高位部分和低位部分。

(2)CPU按照选定的参数添写线路控制寄存器(LCR)的内容。

(3)指出线路状态寄存器(LSR)和MODEM状态寄存器(MSR)的内容。

(4)保护断点,返回调用程序。

2. 发送数据

功能: 根据线路状态,向线路发送AL中的字符。

执行步骤:

(1)向MODEM控制寄存器发送数据终端就绪(DTR)和请求MODEM发送数据。

(2)读MSR的内容,判断MODEM就绪及发送结束与否。

(3)读LSR的内容,判断发送保持寄存器的状态。

(4)保护断点,返回调用程序。

3. 接收数据

功能: 从通信线路中接收数据至AL寄存器中。

执行步骤:

(1)CPU向MODEM控制寄存器(MCR)发送数据终端就绪。

(2)读取MSR的内容,判断MODEM是否就绪。

(3)读取LSR的内容,判断数据是否就绪。

(4)从线路接收字符。

(5)保护断点,返回调用程序。

4. 测试通信口状态

功能: 读取LSR和MSR的内容,返回通信口状态。

执行步骤:

(1)读LSR的内容。

(2)读MSR的内容。

(3)保护断点,返回调用程序。

由以上分析可见,异步通信适配器基本程序是对异步通信适配器的编程。它能够通过异步通信适配器实现发送数据和接收数

据、初始化异步通信适配器及读取通信口状态的功能。它为通过异步通信适配器与外部串行设备(如调制解调器MODEM)通信提供了有利条件。

三、异步通信适配器的 基本程序使用

利用异步通信适配器的基本程序编制与外部串行设备通信的软件可分为以下三个过程:

a 设定初始条件

d 调用异步通信适配器基本程序

c 返回

下面分别介绍利用异步通信适配器基本程序四个部分编制PC机与外部串行设备的通信软件。

1. 初始化通信口软件

初始化通信口软件具有实现设置通信口的初始状态,确定通信参数,波特率、停止位等功能。编制方法如下:

(1)设置初始条件

1)把00H送入DH,00H或01H送入DL,即(DH)=00H,(DL)=00H或01H,分别表示0号或1号插件板。

2)把00H送入AH,即(AH)=00H,表示初始化通信口。

3)把初始化参数送入AL中,AL中各位功能如表1所示。

表1

位	b ₇ b ₆ b ₅	b ₄ b ₃	b ₂	b ₁ b ₀
功能	波特率选择	奇偶校验	停止位	字符长度

b₇、b₆、b₅位是用以确定串行口的波特率的选择位,共可设定从110波特到9600波特八种,这三位的编码与波特率关系如表2所示。

b₄、b₃为设定通信数据每一个字节所要

采用的奇偶校验位，它可设置成三种状态，其含义如下：

① b_4 =任意, $b_3=0$ 表示无奇偶校验位。

② $b_4=0$, $b_3=1$ 表示采用奇校验。

③ $b_4=1$, $b_3=1$ 表示采用偶校验。

表2

$b_7 \quad b_6$ 波特率 (波特) b_5	0	0	1	1
	0	1	0	1
0	110	300	1200	4800
1	150	600	2400	9600

b_2 表示异步通信的停止位选择位。它可设置成以下两种状态：

表3

位	7	6	5	4	3	2	1	0
AH	0	发送移位寄存器空	发送保持寄存器空	停顿指示	缺少正确的停止位	奇偶错	数据重送	数据就绪
AL	接收线路信号检测	振铃指示	数传机就绪	清除发送	△接收线路信号检测	振铃后沿指示	△数传机就绪	△消除发送

选定0号通信插件板，波特率为1200 Baud，无奇偶校验，停止位为1，字符长度为8位通信参数时的初始化通信口程序如下：

```

mov DH, 00H; 选择0号插件板
mov DL, 00H;
mov AH, 00H; 通信口初始化
mov AL, 83H; 初始化参数
INT 14H ; 调用异步通信适配器基本程序

```

中断INT14返回后AL和AH中分别装有调制解调器及线路的状态，它们反应通信口的状态。

2. 发送数据程序

发送数据程序能将所要发送的数据通过通信口传送到通信线路。

编制过程如下：

① $b_2=0$ ，表示停止位为1位。

② $b_2=1$ ，表示停止位为2位。

b_1 、 b_0 位为通信的数据字长选择位。有两种形式：

① $b_1=1$, $b_0=0$ ，表示通信字长为7位。

② $b_1=1$, $b_0=1$ ，表示通信字长为8位。

(2)调用异步通信适配器基本程序

利用软件中断INT14H调用。

(3)返回

中断调用返回时，AX寄存器中装有通信口的状态，其中AH中包含线路的状态，AL中含有调制解调器的状态。它们的含义[1]如表3。

(1)设置初始条件：

1)把00H送入DH，00H或01H送入DL中，即(DH)=00H，(DL)=00H或01H，分别表示选择0号或1号插件板。

2)把01H送入AH中，即(AH)=01H，表示为发送数据。

3)把要发送的数据送入AL。

(2)调用异步通信适配器的基本程序。利用软件中断INT14H调用。

(3)返回

中断调用返回后AH中 b_7 表示异步通信适配器基本程序的返回状态，若 $b_7=1$ 表示发送数据未成功。

例：AL中装有被发送的数据，其发送程序如下：

```

mov DH, 00H; 选择0号插件板
mov DL, 00H;

```

mov AH, 01H; 发送程序
INT 14H ; 调用异步通信适配器基本程序。

异步通信适配器的基本程序返回时, 判断AH中 b_7 即可知发送数据是否成功。

3. 接收数据程序

接收数据程序能通过通信口接收来自通信线路的数据。编制过程如下:

(1) 设置初始条件。

1) 把00H送入DH, 00H或01H送入DL, 即(DH)=00H, (DL)=00H或01H, 表示选择0号或1号插件板。

2) 把02H送入AH, 即(AH)=02H, 表示为接收数据。

(2) 调用异步通信适配器基本程序。

利用软件中断INT14调用。

(3) 返回。

从异步通信适配器基本程序返回后有两种状态, 接收成功时把接收字符存放于AL中, 若不成功, AH中 $b_7=1$ 。

利用异步通信适配器基本程序编写的接收字符程序如下:

```
mov AP, 00H; 选择0号插件板
mov DL, 00H;
mov AH, 02H; 接收数据
INT 14H ; 调用异步通信适配器基本程序。
```

中断INT14H返回后, 若成功则把接收字符至AL中, 若不成功则AH中 $b_7=1$ 。

4. 测试通信口状态程序

测试通信口状态程序是完成测试MODEM的状态及线路的状态, 即取得LSR和MSR的内容。该程序编制过程如下:

(1) 设置初始条件

1) 把00H送入DH, 00H或01H送入DL中, 即(DH)=00H, (DL)=00H或01H, 分别表示选择0号或1号插件板。

2) 把03H送入AH中, 表示测试通信口状态。

(2) 调用异步通信适配器基本程序

利用软件中断INT14调用。

(3) 返回

调用异步通信适配器基本程序的返回状态与前述初始化通信口程序的返回状态一样, 都是MSR和LSR的内容。

测试通信口状态程序如下:

```
mov DH, 00H; 选择0号插件板
mov DL, 00H;
mov AH, 03H; 测试通信口状态
INT 14H ; 调用异步通信适配器基本程序
```

从异步通信适配器基本程序返回时, AH和AL中分别装有调制解调器及线路的状态。

以上介绍了通信软件中四个部分程序及编制方法, 每个程序都能返回其执行状态。通信软件根据不同的返回状态及需要进行相应的处理, 这部分内容主要是信息的处理及加工的问题, 它与异步通信适配器接口板没有直接关系。

四、结束语

采用异步通信适配器编制通信软件中的四个主要部分, 能够克服用高级语言编制所带来的功能差, 运行速度慢等缺点, 又能解决直接用汇编语言编制通信软件中的复杂、易出错等方面的问题。实践证明采用此方法编制通信软件既简便又有效。

参 考 文 献

- 1 张福炎等编著, 《微型计算机IBM-PC的原理与应用》, 南京大学出版社, 1987。
- 2 辽宁省电子计算机学会, 《硬件技术手册》1984。

Apple机与TP801机之间的数据通讯方法

西安电子科技大学电子机械系 杨振江

摘要 本文论述了Apple机与TP801机之间的接口电路以及数据通讯方法。这种方法可使单板机利用Apple机进行软件编程、软件开发、数据处理。以Apple机作为输出方式、TP801机作为输入方式列出了通讯程序清单。

一、前言

目前,使用TP801系列单板机进行各种工业控制、数据采集等应用非常之多,但这类单板机(或单片机)在编程与开发时,几乎都存在着几种缺点:其一,不能直接反复使用助记符编程,而是每次都需要人工汇编(通过机器语言二进制码编程),这就给编程者带来了很多不便,甚至难以实现。其二,所开发的程序或采集的数据不能通过软盘长期保留。其三,在数据分析、数据处理方面难以开发。而本文所介绍的通讯方法,是在现有的机器(Apple机与TP801机)基础上,设计一接口电路,就可方便的将Apple机与TP801机连接起来,并通过相应的执行程序就可实现它们之间的相互通讯了。我们知道,Apple机上可直接插上Z80卡和E-PROM写入卡并能使用CP/M操作系统进行Z80程序设计,因此,在TP801机上所使用的程序就可完全在Apple机上编程调试,并能通过该接口传输到TP801机上实际动态调试、运行。在程序完全调试好以后,还可利用EPROM写入卡在Apple机上将执行程序固化,并插到TP801机上运行。而源程序以及从TP801机上采集的数据还可以文件形式存入Apple机软盘中去。

二、实际通讯的接口电路

Apple机与TP801机之间的接口电路非

常简单。图1是该接口的实际线路图。只需一适配器——VIA6522集成芯片就可将二者连接起来。VIA6522的左边是TP801机中的Z80 PIO A口各信号线;VIA6522的右边是Apple机扩展槽口各信号线。在图中,VIA6522的A口各信号线分别与Z80 PIO A口各信号线对应相连;VIA6522的数据线 $D_0 \sim D_7$ 、四条寄存器选择线 $RS_0 \sim RS_3$ 、读写线 R/\overline{W} 、外设中断请求线 \overline{IRQ} 、芯片选择线 $\overline{CS_2}$ 等线与Apple机数据总线 $D_0 \sim D_7$ 、地址线 $A_0 \sim A_3$ 、读写线 R/\overline{W} 、中断请求线 \overline{IRQ} 与外设选择线 \overline{DEV} 等线对应相接。其中,TP801机要与Apple机共地。

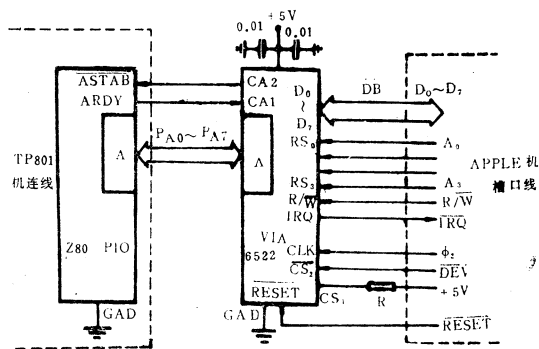


图1 Apple机与TP801机接口

为了确定VIA6522各寄存器的地址以及编程和叙述方便。我们假设接口板是插在Apple机的第七个槽口上(若插在其它槽口也可以,只是 \overline{DEV} 外设线地址区间要相应的改变),此时,Apple机的外设线 \overline{DEV} 所对应的地址是 $C0F0H \sim C0FFH$ 。因此,VIA6522 A口输入和输出数据寄存器、方向

寄存器。外设控制寄存器和中断标志寄存器的地址分别是：C0F1H、C0F3H、C0FCH和C0FDH。TP801机Z80 PIO的A口数据寄存器和控制寄存器地址（由TP801机硬件决定）分别是80H和82H。

三、通讯软件

完成Apple机与TP801机之间的相互通讯，需要各自的传输程序，即：Apple机作为输出，TP801机作为输入和TP801机作为输出，Apple机作为输入的相应控制程序。它们之间的协调工作均受ASTB、CA2、ARDY和CA1信号的联络与控制，且对不同的工作方式，这四条线所起的作用也不相同。

1. ASTB、ARDY、CA1和CA2信号的作用

在TP801机作为输出工作方式时，ASTB信号作为Apple机收到TP801机数据的回答信号，ARDY信号作为TP801机通知Apple机数据已准备好的输出信号；在TP801机作为输入工作方式时，ASTB信号作为Apple机通知TP801机数据已准备好的输入选通信号，ARDY信号作为TP801机已收到Apple机数据的回答信号。在这两种方式中，ASTB信号的上升沿将会引起中断。

在Apple机作为输入工作方式时，CA1信号作为TP801机通知Apple机数据已准备好的联络信号，CA2信号作为Apple机已收

到TP801机数据的回答信号；在Apple机作为输出工作方式时，CA1信号作为TP801机已收到Apple机数据的回答信号，CA2信号作为Apple机通知TP801机数据已准备好的选通信号。在这两种方式中，CA1信号的上升沿将会引起中断。

2. 程序清单

以TP801机作为中断输入方式、Apple机作为查寻输出方式的通讯程序见附录A与附录B。在执行该程序时，应先置数据的起址和末址。对Apple机来说，起址和末址应分别放在\$E0、\$E1、和\$E2、\$E3内存单元中。对TP801机来说，起址和末址应分别放在HL和DE寄存器对中。对于Apple机作为输入方式，TP801机作为输出方式的程序清单，因编程基本与前者相同，即略。

四、结 论

该文所介绍的通讯方法是简单、经济和实用的。使用者在不需要增加太多的硬件电路的基础上，经过执行相应驱动程序就可实现Z80单板机的实际开发。虽然，本文是对Apple机与TP801机而言，但 this 方法是完全可以推广到单片机与Apple机或单片机与IBM—PC机等机型中去。特别是对单片机被广泛应用的今天更是如此。

参考文献

1 中华学习机技术参考手册，清华大学出版社

1987.8

2 TP801机使用手册，1984.8

附录A:

TP801机为输入方式程序清单

SOURCE FILE: Z80_APPLE.PRN

	(0800)	0001	ORG	0800H
	(00AE)	0002	RESTRI:	EQU 00AEH
0800	3E08	0003	START1:	LD A, 08H
0802	ED47	0004		LD I, A
0804	3E30	0005		LD A, 30H
0806	D382	0006	OUT	(82H), A
0808	3E4F	0007	LD	A, 4FH

080A	D382	0008	OUT	(82H) , A
080C	3E87	0009	LD	A, 87H
080E	D382	0010	OUT	(82H) , A
0810	ED5E	0011	IM2	
0812	FB	0012	EI	
0813	E5	0013	PUSH	HL
0814	AF	0014	XOR	A
0815	ED52	0015	SBC	HL, DE
0817	44	0016	LD	B, H
0818	4D	0017	LD	C, L
0819	E1	0018	POP	HL
081A	DB80	0019	IN	A, (80H)
081C	77	0020	LD	(HL) , A
081D	76	0021	LOOP :	HALT
081E	0B	0022	DEC	BC
081F	AF	0023	XOR	A
0820	E5	0024	PUSH	HL
0821	60	0025	LD	H, B
0822	69	0026	LD	L, C
0823	110000	0027	LD	DE, 0000H
0826	ED52	0028	SBC	HL, DE
0828	E1	0029	POP	HL
0829	C21D08	0030	JP	NZ, LOOP
082C	F3	0031	DI	
082D	C3AE00	0032	JP	RESTRI
0830	40008	0033	DEFW	0840H
	(0840)	0034	ORG	0840H
0840	DB80	0035	START2 :	IN
0842	77	0036	LD	(HL) , A
0843	23	0037	INC	HL
0844	FB	0038	EI	
0845	ED4D	0039	RETI	
0847	(0800)	0040	END	START1

附录B:

Apple机为输出方式程序清单

SOURCE FILE: APPLE_Z80.PRN

1000 :	1	ORG	\$ 1000
1000 : A0 00	2	START	LDY # \$ 00
1002 : A9 FF	3	LDA	# \$ FF
1004 : 8D F3 C0	4	STA	\$ C0F3
1007 : A5 E3	5	LOOP2	LDA \$ E3
1009 : C5 E1	6	CMP	\$ E1
100B : F0 26	7	BEQ	LOOP3
100D : A9 0F	8	LOOP4	LDA # \$ 0F

100F: 8D FC C0	9	STA \$C0FC
1012: AD FD C0	10 LOOP1	LDA \$C0FD
1015: 29 02	11	AND # \$02
1017: F0 F9	12	BEQ LOOP1
1019: B1 E0	13	LDA (\$E0), Y
101B: 8D F1 C0	14	STA \$C0F1
101E: A9 0D	15	LDA # \$0D
1020: 8D FC C0	16	STA \$C0FC
1023: 18	17	CLC
1024: A5 E0	18	LDA \$E0
1026: 69 01	19	ADC # 01
1028: 85 E0	20	STA \$E0
102A: A5 E1	21	LDA \$E1
102C: 69 00	22	ADC # 00
102E: 85 E1	23	STA \$E1
1030: 4C 07 10	24	JMP LOOP2
1033: A5 E2	25 LOOP3	LDA \$E2
1035: C5 E0	26	CMP \$E0
1037: D0 D4	27	BNE LOOP4
1039: 00	28	BRK
1040:	29	END

上接 (第28页)

```

30 OPEN "CODE, DAT" FOR
    APPEND AS # 1
40 CLS
50 LOCATE 8, 1: INPUT
    "输入汉字: ", X$
60 LOCATE 1, 1: PRINT X$
70 PRINT # 1, " "; X$; "
    ( " ; HEX$(ASC(MID$(X$,
    1, 1))) );
    HEX$(ASC(MID$(X$,
    2, 1))) ); " ) "
80 FOR I=0 TO 3
90 PRINT # 1, CHR$(9);
    "DB" ;
100 FOR J=0 TO 7
110 N=I*8+J
120 X=(N MOD 2)*8
130 Y=INT(N/2)
140 GET(X, Y)-(X+7,
    Y), C%
```

```

150 IF C%(2)<16 THEN
    180
160 PRINT # 1, "0", HEX$
    (C%(2)); "H" ;
170 GOTO 190
180 PRINT # 1, "0C", HEX$
    (C%(2)); "H" ;
190 IF J=7 THEN 220
200 PRINT # 1, " ";
210 GOTO 230
220 PRINT # 1
230 NEXT J
240 NEXT I
250 LOCATE 9, 1: INPUT "继续输入吗? (Y/N)", X$
260 IF X$="Y" OR X$="y"
    THEN 40
270 CLOSE # 1
280 END
```