

微小型计算机



# 开发与应用

MICRO-MINICOMPUTER  
DEVELOPMENT & APPLICATION

1990 6

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  
天津市电子计算机学会

**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0146号

国内统一刊号CN12—1122

# 《微小型计算机开发与应用》

# 1990年第6期目录

## 编辑委员会

(总第50期)

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
	章渭臣 梅克定
	童宣明 裴少峰
	薛大中

## 应用实例

电力系统动态模拟微机变送系统……	张红萍 (1)
8031单片机在分光光度计中的应用…	张剑波 (4)
单板机控制的钨极氩弧焊机系统 ……	
………	赵质健 冯常洗 (6)
PLC可编程控制器在造纸低浓盘磨机中的应用…	
………	施能民 (10)
一种基于传感器的机器人反馈控制方案及	
实验研究………	苑静中 (12)

## 计算机软件

计划编制系统的设计与实现 ……	
………	刘辉 赵青玲 吴振庆 (16)
面向问题的自动编程……	朱瑞香 袁晓洁 (21)
全国联行对帐处理系统……	吴锦声 卢文 (26)
一个实用的词组库编译程序……	
………	郑国安 (29)
论MICRO VAX II 的编程方法…	吴锦前 (35)

## 新产品介绍

适于并行处理的处理器——T414	陈荣华 (43)
------------------	----------

# CONTENTS

## APPLICATION EXAMPLE

Microcomputer Transformer System Used in Dynamic Analogue for Power Stations .....	Zhang Hongping ( 1 )
Application of 8031 Single Chip Microcomputer in Spectrophotometer .....	Zhang Jianbo ( 4 )
Tungsten Argon Arc Welding System Controlled by Single Chip Micro-computer .....	Zhao Zhijian Feng Changxi ( 6 )
Application of the Programmable Controller in Paper-making Disk Refiner .....	Shi Nengnia ( 10 )
A Robot Feedback Control Method and Experiment Based on Sensors .....	Yuan Jingzhong ( 12 )

## SOFTWARE

The Designation and Implementation for a Project Making System .....	Liu Hui    Zhao Qingling Wu Zhenqing ( 16 )
Problem-oriented Automatic Programming .....	Zhu Ruixiang Yuan Xiaojie ( 21 )
A Processing System of Checking Account for Joint Operation in Whole China .....	Wu Jinsheng Lu Wen ( 26 )
A Practical Compiler for Phrase Base .....	Zheng Guoan ( 29 )
On Programming Method of MICRO VAX II .....	Wu Jinqian ( 35 )

## NEW PRODUCT

Processor T414 Which Is Suitable for Parallel Processing .....	Chen Ronghua ( 43 )
--	---------------------

# 电力系统动态模拟微机变送系统

华中理工大学 张红萍

**摘要** 本文介绍用Z80—Ⅰ型单板机代替电力系统动态模拟中的电压及电流变送器,并对电力系统的各种运行状态量进行实时检测和处理。该系统具有较高的采集速度和较强的数据处理及自定义显示功能。

## 一、引言

随着微型计算机的迅速发展,电力系统中许多传统的检测、控制和保护装置正在逐渐地被微机代替。现在连一些传统的设备,如变送器也可被微机取代。电力系统中常用的变送器延时惯性较大,测量精度不高,用微机代替它即提高了电量的转换速度又提高了测量精度,使电量的输出数值化,对进一步用微机作实时控制或继电保护提供了充分条件。

## 二、设计思想

电力系统动态模拟及真实的电力系统都需要迅速而准确地检测系统运行中各种状态量,如发电机的电压、电流、有功、无功功率等。为了满足这一需要,设计中考虑采用速度较快且价格又便宜的Z80—Ⅰ型单板计算机作主件,配备一台微机接口装置来完成一台发电机组的检测工作。

### 1. 电量信号采集

由于软件设计上的优化处理,只要采集发电机运行中的电压和电流瞬时值,即可算出电压、电流、有功功率和无功功率的有效值。

电量信号的采样是等间隔的,即 $t=nT_s$   
 $T_s$ 就是各相邻采样的时间间隔。采得的量是离散的时间信号,可用一个时间——数的序列 $X$ 来表达,序列中的第 $n$ 个数的符号为 $X(n)$ ,因此序列 $X$ 可用下式表达:

$X = \{X(n)\} \quad -\infty < n < +\infty$   
这种无限序列可以对应电力系统运行中的各

种电量信号序列。

发电机的电压,电流信号就是周期性的无限序列,一般为正弦波信号。对它们的采集可以周期为单位,如一个周期采样三点、十二点、二十四点等。为了与软件处理相配合,该系统用一个周期采样十二点。这样就有:

$T_s = \frac{T}{12}$  一般情况下发电机频率为50 Hz,所以周期 $T$ 为20ms,故采样间隔 $T_s$ 为1.67ms。

采用Z80-CTC作定时器件采样,有公式如下:

$$T_s = N \times n \times T_j$$

其中:  $N$ 为CTC定标系数,取为256;

$T_j$ 为单板机工作周期为0.5μs;

$T_s$ 为采样间隔为1.67ms。

可算出 $n=13$ ,程序中定CTC时间常数  
为十六进制数0D即可满足采样要求。

### 2. 软件设计基础

软件上采用富氏十二点带滤波的算法,先用采得的发电机电压、电流瞬时值算出电压、电流的虚、实部分量: $U_x$ 、 $U_R$ 、 $I_x$ 、 $I_R$ ,然后再根据公式算出所求的电压、电流、有功功率、无功功率有效值。公式推算如下:

因为:

$$\begin{aligned} S &= U \times I \\ &= (U_R + jU_x)(I_R - jI_x) \\ &= (U_R \cdot I_R + U_x \cdot I_x) + j(U_x \cdot I_R - U_R \cdot I_x) \end{aligned}$$

又因为

$$S = P + jQ$$

所以有功功率

$$P = U_R \cdot I_R + U_x \cdot I_x \quad (1)$$

$$\text{无功功率 } Q = U_X \cdot I_R - U_R \cdot I_X \quad (2)$$

$$\text{电压有效值 } U = \sqrt{U_R^2 + U_X^2} \quad (3)$$

$$\text{电流有效值 } I = \sqrt{I_R^2 + I_X^2} \quad (4)$$

其中带X下标为虚部分量,带R下标为实部分量。

软件设计中安排单板机上的六个自定义键分别为显示电压、电流、有功功率和无功功率及运行程序键。程序设计容许在做电力系统短路试验或其它试验时,提高采样速度;通过数模转换板的D/A口输出采样值给其它设备或上层计算机作进一步处理。

### 三、硬件结构

微机变送系统硬件结构如图1,具体介绍如下。

#### 1. 组成

(1) 主机采用Z80-Ⅱ型单板计算机,主频2MHz,字长8位。

(2) ADA-168模数、数模转换板,16路单端输入,输入信号0~5V;8路输出0~5V,字长8位,转换时可15μs。

(3) 微机外层接口装置,三个交流电压量和四个交流电流量的输入输出口,内设滤波电路。

(4) 直流抬高电路,将输入信号的零点上移2.5V输出。

图1中PT为电压互感器,CT为电流互感器,PT、CT之前为系统设备,PT、CT之后为本文所介绍的装置及电路。下面将着重介绍微机外层接口装置和直流抬高电路。

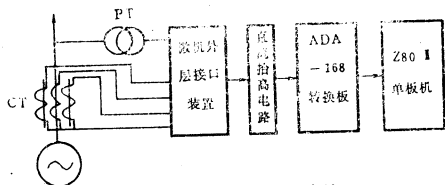


图1 微机变送系统结构图

#### 2. 微机外层接口装置

微机外层接口有三个交流电压输入量和四个交流电流输入量,七个输出均为0~2.5V交流量。它的组成框图如图2所示。

交流电压和电流的输入是从动态模拟的电压互感器PT和电流互感器CT送来的。微

机外层接口的任务有二:其一是将互感器送入的电量进行比例变换,使最大可能输入电压瞬时值和最大可能输入电流瞬时值变换为不超过A/D板规定的电压值;其二是保证动态模拟系统和计算机系统有可靠的屏蔽隔离和有足够的抗干扰作用,并滤除部分高频信号。其中的低通滤波部分是为了满足采样定理而装设的,它由两节RC电路组成,滤波部分的时间常数不超过4ms。

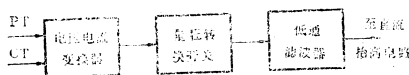


图2 微机外层接口框图

#### 3. 直流抬高电路

由于ADA-168转换板的输入量是单极性的,现采用一个以LN324运算放大器为核心的直流抬高电路,如图3。这样就可将交流输入信号转变为ADA-168板可接收的单向输入了。

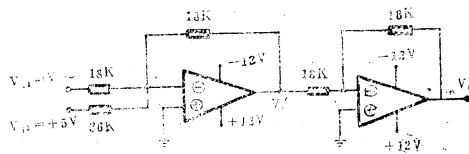


图3 直流抬高电路图

其中 $V_i$ 为来自微机外层接口的电量;+5V为直流偏压量。

此电路可将输入信号抬高2.5V,即零点上移2.5V。微机外层接口的输出为交流0~2.5V经过此电路调整后正好为ADA-168转换板输入端可接收的信号。

### 四、软件系统

#### 1. 算法分析

电力系统在正常运行状态下,电量信号基本上是周期函数,它可被分解为一个函数序列之和或说是一个级数。由于富氏级数是将周期函数分解为正弦函数,最适合于计算机计算其基频分量或倍频分量之用,所以微机用于电力系统最常用到的是富氏算法。

富氏算法是采用某一正交函数组作为样品函数, 将这一正交样品函数组与待分析的时变函数进行相应的积分变换, 求出与样品函数频率相同的分量的实部和虚部系数, 进而可求出待分析的时变函数中该频率的谐波分量的模和相位。

实部分量:

$$U_{Rn} = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} U(t) \cdot \cos n\omega_0 dt$$

虚部分量:

$$U_{Xn} = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} U(t) \cdot \sin n\omega_0 dt$$

用离散值计算时其实部为:

$$U_{Rn} = \frac{2}{N} \sum_{k=1}^N U_k \cdot \cos nk \frac{2\pi}{N}$$

式中N为一个周期T中的采样个数, 取N=12; 由于是计算基频数值, 故取n=1。用采得的12点样值计算 $U_{Rn}$ 有下式:

实部分量:

$$U_R = 2(U_{12} - U_6) + (U_2 - U_8 - U_4 + U_{10}) + \sqrt{3}(U_1 - U_7 - U_5 + U_{11}) \quad (5)$$

同理虚部分量为:

$$U_X = 2(U_3 - U_9) + (U_1 - U_7 + U_5 - U_{11}) + \sqrt{3}(U_2 - U_8 + U_4 - U_{10}) \quad (6)$$

(5)、(6)式即为计算实、虚部分量公式。电流的实、虚部分量计算公式与之相同。有了电压、电流的虚、实部分量, 就可根据(1)、(2)、(3)、(4)式算出所求的电量。

## 2. 程序设计

电力系统中电压、电流均为A、B、C三相, 并且每一相的计算公式相同, 设计中利用变址寄存器IX, 每相计算时可调用同一个计算子程序, 节省内存简化程序。检测计算程序见流程图4。

用Z80-CTC作定时中断采样, ADA-168转换板的六个A/D输入通道, 分别采入三相电压和三相电流的瞬时值。每一通道输入的电量建立一个12点的数据窗口, 程序设计中采用窗口移动式, 保证数据窗中始终为最新的电量输入值。中断采样程序流程见图5。

将Z80-II单板机中的六个自定义键分别

用于显示电压、电流、有功功率、无功功率和一个运行程序键。在显示程序中先将算得的电量值由十六进制数转化为十进制数, 再调用显示子程序。电压、电流量各保留一位小数位显示。自定义任选显示程序流程见图6。

## 3. 子程序库

十六位带符号乘法子程序: MULT16;

三十二位求补子程序: QB32;

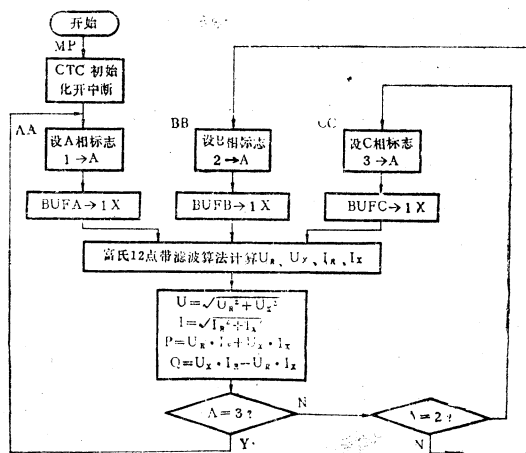


图4

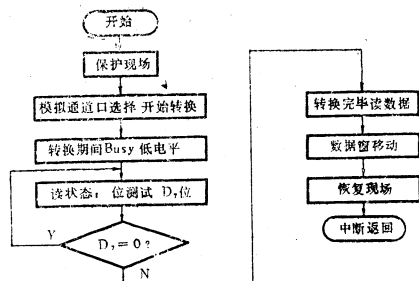


图5

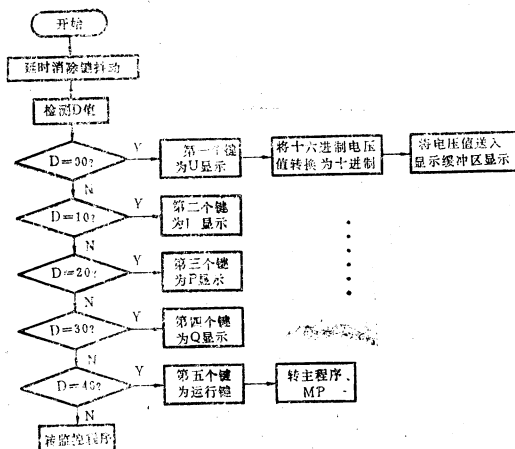


图6

# 8031单片机在分光光度计中的应用

浙江大学计算与信息中心 张剑波

**摘要** 本文介绍8031单片机控制的分光光度计数据处理仪,及其和流动比色管联用实现自动分析的方法。

## 一、引言

紫外可见分光光度计是我国最早投入生产的光谱仪器,也是国内应用最广泛的分析仪器之一。国产72系列、75系列单光束紫外可见分光光度计,虽然其功能没有进口的高档仪器强,但由于它的价格低廉,性能稳定,能满足一般性的定量分析而受到普遍欢迎。但是,这类仪器存在实验数据量大的缺点,操作员在数据处理方面的工作量往往是样品分析的数倍,从而大大影响了仪器的使用效率及范围。我们采用8031单片机为核心,研制成功用于分光光度计的数据处理仪,该处理仪和流动比色管配合,可自动地完成样品的定量分析和报表生成。

## 二、仪器组成

分光光度计与数据处理仪组成的自动分析系统框图如图1所示。当一定波长单色光通过有色溶液时,吸光度与溶液的浓度成正比。透射光经光电转换并放大后,以电位形式输出透光率或吸光度。

模拟式分光光度计(如721, 751等)与计算机进行联机工作时,首先应经过模数转换器把模拟电压转换成数字量,并在计算机控制下,进行数据处理及结果输出。

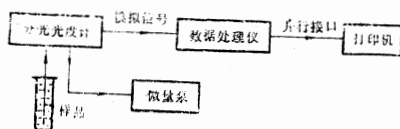


图1 分光光度计自动分析系统框图

**1. 分光光度计** 为了要实现分光光度计的分析自动化,应采用流动比色管代替一般比色管。用微量泵或其它应用虹吸原理的抽液泵,使溶液经过比色管,只要待测液流经比色管的速率和计算机定时测量的时间间隔相匹配,就可自动地完成整个测量过程。

由于分光光度计种类较多,相应的输出电平范围也各不相同,常见的有0~10mv, 0~200mv, 0~2v等几种。因此,为了使它们能和A/D芯片要求的电压输入范围匹配,我们采用放大倍数多档可调的前置放大器。前置放大器还具有低通滤波作用,其截止频率为30Hz,以消除50Hz工频干扰。

**2. 数据处理仪** 如图2所示为数据处理仪的原理框图,该处理仪采用MCS—51系列单片机8031,其中P<sub>0</sub>、P<sub>2</sub>口用作系统的数据总线 and 地址总线,P<sub>1</sub>口用于16键矩阵的接口,两根中断引脚 $\overline{INT_0}$ 、 $\overline{INT_1}$ 分别和ADC0804及键盘的中断请求线相连,内部定时器工作于方式0定时0.1秒作为时间基准,完成样品定时测量,并行接口8255的A、B口用于8位数码管及打印机的数据线,C口用作它们的控

二十四位除法子程序: DEVID24;

三十二位快速开方子程序: KF32;

十六进制转化十进制子程序: ZH10;

十二点富氏算法求实部分量子程序:  
POWER;

十二点富氏算法求虚部分量子程序:  
POWX;

显示子程序: DISPUT.

## 参考文献

- 1 陈德树、尹项根, 计算机继电保护原理与技术, 湖北省电力中心调度所出版, 1987.3.
- 2 动态模拟微型计算机控制系统外层接口装置, 华中工学院, 1983年4月.



制线, EPROM 27128内存放监控程序及各功能子程序,提供用户界面、控制模数转换器ADC0804,并行接口8255,随机存储器6116等芯片,有步骤地完成信号采集、运算、存

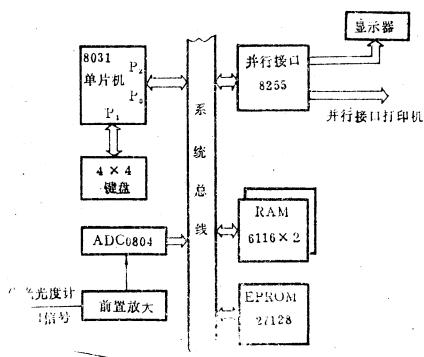


图2 数据处理仪原理框图

储等功能,并通过八段LED及并行接口打印机输出各种中间提示,结果及其报表。

### 三、主要功能及算法

本数据处理仪提供直读吸光度、浓度、求取拟合直线及定时测定吸光度和浓度等功能,它们都在监控程序的管理下完成,现将主要功能的算法介绍如下。

**1. 直读吸光度** 根据定义,溶液的吸光度可表示为空白透射光强和含样品透射光强之比的对数来表示,而光强到A/D采样值的转换是线性的,因此吸光度应等于相应的采样值之比的对数值,即单片机只要对输入进行采样、求商,求对数并显示即可直读吸光度。

**2. 拟合直线的求取** 根据光的吸收定律,在其它条件恒定时,待测物质的浓度应和吸光度成正比。因此,可根据一组已知浓度的标准样品及相应的吸光度,用最小二乘线性拟合方法求出拟合直线的各项系数,

$$A = b_0 + b_1 C$$

A为吸光度, C为标样浓度,

$$b_1 = S_{Ac} / S_{Cc}$$

$$b_0 = \frac{1}{n} \left( \sum_{i=1}^n A_i - b_1 \sum_{i=1}^n C_i \right)$$

拟合直线的优劣可用相关系数和最大偏差来检验。

$$\text{相关系数 } r = S_{Ac} / \sqrt{S_{AA} \cdot S_{Cc}}$$

$$\text{最大偏差 } \delta = \max |A_i - b_1 C_i - b_0|$$

$$\text{其中: } S_{AA} = \sum_{i=1}^n A_i^2 - \frac{\left( \sum_{i=1}^n A_i \right)^2}{n}$$

$$S_{Cc} = \sum_{i=1}^n C_i^2 - \frac{\left( \sum_{i=1}^n C_i \right)^2}{n}$$

$$S_{Ac} = \sum_{i=1}^n \left( A_i \cdot C_i \right) - \frac{\sum_{i=1}^n A_i \cdot \sum_{i=1}^n C_i}{n}$$

$C_i, A_i$ 为第*i*个标准样品的浓度及相应的吸光度,  $n$ 为标样个数。

**3. 定时测量功能** 在进入定时测量以前,用户应先设置定时间隔。定时测量的工作原理如图3所示,首先判断定时间隔是否已到,如时间到,即采用256次平均迭加采集当前透过光度,并和空白透过光强值一起推算出吸光度A,再由拟合直线方程推算出待测物质的浓度,并把时间、吸光度、浓度通过打印机输出,全部过程完成后,定时计数置零,并进入下一次测量。

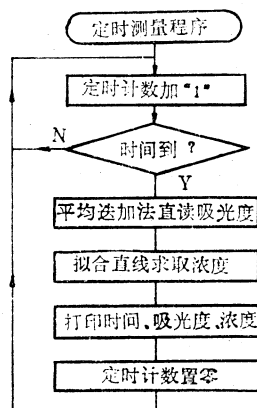


图3 定时测量程序流程图

### 四、结果与讨论

该仪器在研制完成后,已在医院、冶金环保、地质矿产等单位使用,普遍反应良好,

# 单板机控制的钨极氩弧焊机系统

辽宁省警务应用软件研究所

赵质健

黎明航空发动机公司

冯常洗

**摘要** 本文介绍了利用TP-801单板机及其接口元器件,设计钨极氩弧焊机控制系统的方法。该系统可按照焊接工艺的要求,自动控制各个焊接过程,并能自动调节电弧长度。文中给出了输入/输出接口连线图、步进电机的环形分配表及程序流程图。

利用TP-801单板机控制的钨极氩弧焊机系统,能够实现焊接过程的自动控制。对某些焊接参数如焊速、电弧长度进行自动控制。由于采用八位的模/数转换器,可以保证当电弧电压变化大于 $\pm 40$ 毫伏时,自动进行弧长调节。为了保证系统的稳定运行和模拟量、开关量的可靠传送,对环境的干扰和高频引弧所产生的强干扰,采取了必要而有效的措施。

这个系统是针对某圆筒形产品的环形对接焊缝的工艺要求而设计的。为了便于了解系统,现将焊接的工艺要求介绍如下:

焊接工艺要求七个状态,如图1。

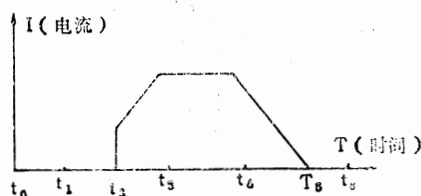


图 1

- $t_0$ —送氩气;  $t_1$ —焊枪通电;
- $t_2$ —引弧及小电流上升;
- $t_3$ —在额定电流下进行弧长自动调节
- $t_4$ —电流衰减;
- $t_5$ —熄弧;  $t_6$ —停氩气。

在 $t_3-t_4$ 阶段,要求电弧电压的变化应小于 $\pm 150$ 毫伏。

## 一、控制方案

焊接过程的控制可以方便地通过单板机的输入输出实现。Z80-PIO是可编程的具有A、B两个端口的输入/输出器件。每个端口有八位数据线与外设相接。CPU与外设之间的全部数据传送均经PIO传送。Z80-PIO有四种可供用户选择的工作方式。我们选用方式3一位控方式。在这种方式下,CPU可以向端口写入数据,也可以从端口读出数据,所以这种方式特别适用于实现状态控制。

单板机控制氩弧机焊的另一项重要任务是进行电弧长度的自动调节。

弧长自动调节的方法是通过八位模/数转换器将电弧电压转换成八位数字量,再与给定的标准电压在CPU中进行减法运算,运算结果的符号位控制带动焊枪拿上下运动的步进电机的运动方向,运算结果的差值控制步进电机应转的步数,从而实现弧长的自动调节。

焊枪的运动速度能否满足最高焊速的要

该仪器已在89年通过浙江大学学术委员会组织的技术鉴定。该仪器和流动比色管联用,在浙江大学医院用于自动测量GPT表明,可大大提高分析及处理速度,一般操作人员可

在10秒内完成1个样品的测量。此外,该仪器还可应用于需用分光光度计进行测定的化工及医药制造等生产过程。

求,是系统能否实现弧长自动调节的关键。按照焊接工艺的要求,最高焊接速度为13毫米/秒。焊枪的上下运动是由步进电机带动滚珠丝杠运动实现的。当步进电机按 $f=500$ 步/秒的频率中速旋转时,其可靠性最好。此时,步进电机每转一步所用时间为2毫秒。我们设计步进电机的步进当量为0.042毫米/步。这样,步进电机每秒转310步,即可满足最高焊速的要求,而且在这个条件下,每秒钟内弧长调节占用时间为2毫秒 $\times 310=620$ 毫秒,其余380毫秒的时间可供CPU执行调节弧长的程序。由上述分析可以得出这样的结论,系统是能够满足焊接工艺最高焊速的要求的。

为了实现自动焊接,还应进行焊缝终点控制。对于环形件焊缝,按照工艺要求,一般应有20毫米的焊缝重叠长度。若环形焊缝长为 $S$ 毫米,则施焊长度为 $(S+20)$ 毫米。我们用Z80-CTC计数/定时器的通道1—通道3,来完成焊缝的终点控制。CTC是具有四个独立通道的可编程的器件,当它作为定时器时,能按照程序的安排定时地发出脉冲,并在发出脉冲的同时向CPU申请中断;当它作为计数器时,能对外界事件进行计数,在达到程序规定的数值时,向CPU申请中断,并可输出一个脉冲。

CTC的接线如图2。

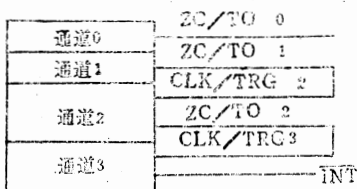


图2

在程序中规定,CTC通道1按定时器方式工作。在开始进行弧长调节时,启动通道1,使其每隔30毫秒由ZC/TO1端输出一个脉冲。工作在计数器方式的通道2的CLK/TRG2端接收到此脉冲后,每隔 $(20\text{毫米/伏})$ 秒便在

ZC/TO2端输出一个脉冲。其脉冲间隔时间即为焊件以焊速 $V$ 运动20毫米所需要的时间。CTC通道3工作在计数器方式,其时间常数为 $T_c = (\frac{S}{20} + 1)$ ,由接线图可以看到,通道2的输出脉冲作为通道3的外触发信号。这样接线的意义为:焊件每走20毫米,通道3的计数器的时间常数减1。(当通道3计数结束后,发出 $\overline{\text{INT}}$ 信号,通知CPU施焊长度已达到 $(S+20)$ 毫米,应该进行下一个焊接动作了。

## 二、系统的构成

### 1. 系统结构如图3。

### 2. 端口分配说明

在设计输入/输出部分时,一个问题要考虑外部设备与微型机的配合,并传送数据、状态与控制命令。另一个问题是如何对输入/输出设备寻址,以便CPU可选出某一特定的输入/输出设备供数据传送用。

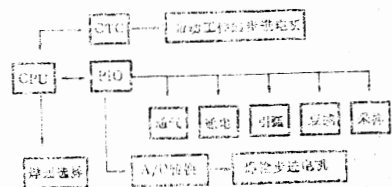


图3

在系统中,微型机与外部设备的接口用了两块Z80-PIO芯片和一块CTC芯片。分别介绍如下:

PIO<sub>1</sub>A口的八位端口数据总线,与模/数转换器的八位数据线相连接,用来把一个八位的数字量送入CPU中。PIO<sub>1</sub>B口和PIO<sub>2</sub>A口共十六位端口数据总线,用于实现自动焊接过程所需要的各种开关量的输入和输出。其中PIO<sub>1</sub>的PB<sub>7</sub>-PB<sub>0</sub>用于控制焊接过程。PA<sub>7</sub>-PA<sub>0</sub>输入各焊接动作执行情况的数据,以提高系统的可靠性。其中PA<sub>0</sub>作为“急停”输入,即当焊机发生意外故障时,



中断CPU的工作, 切断焊接电源, 保护断点及各寄存器内容, 以便故障排除后继续工作。PIO<sub>2</sub>B口的PB<sub>0</sub>-PB<sub>2</sub>经功率放大电路接至焊枪步进电机的三相绕组, 以控制焊枪运动。

输入/输出接口连线见图4。

Z80-CTC芯片共有四个通道, 其中通道1-3的用处已如前述, 通道0用来产生带动工作旋转的步进电机所需的步进脉冲。只要改变通道0的时间常数就可以方便地改变焊速。

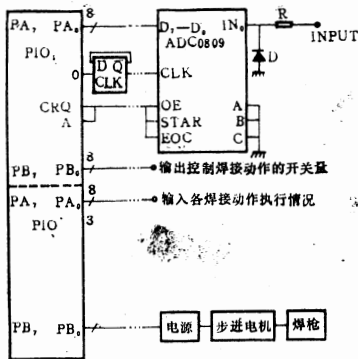


图4 图中的“...”表示光电隔离电路

### 3. 模/数转换

采样得到的实际电弧电压是模拟量。这个模拟量需要在单板机的CPU中与标准的焊接电压相比较。而在单板机中, 所有的信息都是以二进制的数字量形式处理的。这就提出了将模拟量转换成数字量的问题。在本系统中, 采用ADC0809芯片来完成上述模拟量到数字量的转换工作。

ADC0809芯片是一种模/数(A/D)转换片, 其中包括“模拟多路转换器”和“A/D转换器”两大部分。“模拟多路转换器”有8路模拟开关, 它可以选通8路模拟量输入的任何一路单端模拟信号。地址译码与输入选通的关系见表1。

当A、B、C为不同的输入电平时, 分别选中ADC0809的输入通道0-7。

“A/D转换器”包括一个高阻抗削波器, 漂移补偿比较器, 一个带有树状模拟开关的256R分压器和一个逐次近似寄存器。它

将“多路转换器”输出的模拟量转换成八位的数字。

表1

模拟量 通道选择	地址线		
	C	B	A
IN 0	0	0	0
IN 1	0	0	1
IN 2	0	1	0
IN 3	0	1	1
IN 4	1	0	0
IN 5	1	0	1
IN 6	1	1	0
IN 7	1	1	1

在ADC0809芯片中, A/D转换速度取决于A/D转换芯片的时钟频率。ADC0809的时钟频率范围为10千赫-1280千赫。被转换的模拟电压通过“比较器”与基准电压进行八次比较, 每次比较需要八个时钟周期, 因此转换时间与时钟频率的关系为:

$$T_{\text{转换}} = \frac{1}{f} \times 8 \times 8$$

按照上式, 如果A/D转换芯片的时钟频率为1兆赫, 则

$$T_{\text{转换}} = \frac{1}{1\text{兆赫}} \times 8 \times 8 = 1\text{微秒} \times 8 \times 8 =$$

64微秒

ADC0809在工作时, 有两种工作方式。一种是用中断方式完成等待, 即CPU在启动ADC0809后, 仍做其它工作。A/D转换结束后, ADC0809发出EOC信号, 并送入OE, 再送入PIO的ASTB端, 则PIO用中断方式向CPU申请中断, 将转换结果读到CPU中。另一种是由软件延时完成等待。本系统采用了

后一种方式。在程序启动ADC0809后,CPU延时,其时间大于A/D转换所需时间,然后将A/D转换结果读入累加器中。

因为系统只有一路模拟量输入,所以A、B、C三端均接地,也就是选择了通道0作为模拟量输入端。

在A/D转换中,数字量01H对应的模拟量为20毫伏。考虑到采样时的衰减,数字量01H对应的电弧电压为40毫伏。这样,当电弧电压偏离标准电压 $\pm 40$ 毫伏时,CPU即控制步进电机开始进行调节弧长。由此可见,电弧电压的变化远小于 $\pm 150$ 毫伏,即满足了焊接的精度要求。

#### 4. 步进电机的环形分配

由于采用了微处理机,在自动调节系统中对执行部件——步进电机的控制,就出现了更加灵活方便的方法。传统的控制步进电机的方法是步进脉冲经环形分配器分配后,送到步进电机的各绕组。我们通过编制程序让计算机完成环形分配器的工作。这样,即有助于提高系统的效率和可靠性,又降低了费用。

在系统中,步进电机的工作模式选定为三相双三拍,即各相绕组正向通电顺序为AB-BC-CA-AB。现采用移位法实现环形分配,其方法简介如下:

建立一个环形分配表。如表2。

表2

	C	B	A	nn											
11	10	9	8	7	6	5	4	3	2	1					
0	0	1	1	0	1	1	0	0	1	0					
0	1	1	0	0	1	0	1	0	0	1					
0	1	0	1	0	0	1	1	0	1	1					

在环形分配表中,“1”表示对应的步进

电机绕组通电;“0”表示绕组不通电。程序在初始化时,将表中的0—7位放入某一存储单元中,8—11位放在累加器A的D<sub>0</sub>—D<sub>3</sub>位。累加器A的D<sub>0</sub>—D<sub>2</sub>位输出到PIO<sub>2</sub>B口的PB<sub>0</sub>—PB<sub>2</sub>位。如果需步进电机正转,在每送出一累加器中的内容后,环形分配表左移一次(四位),再送出累加器的内容。如需步进电机反转,则将环形分配表右移一次(四位)再送出累加器中的内容。PIO送出的环形分配的内容,经过每次移位后,各不相同,步进电机各绕组按其移位的顺序依次通电,实现步进电机旋转。采用软件完成步进电机脉冲的环形分配工作,大大简化了硬地件接口。

上述用程序设计实现环形分配的方法,只需稍加改动,也适用于五相以下的各种工作模式的步进电机。

有关的程序流程见弧长自动调节程序流程图。

### 三、抗干扰措施

单板机在工作中不可避免地要受到来自各方面的干扰。在同一电源负载上的其它设备的通、断电,要影响单板机的正常运行。氩弧焊机本身的高频引弧脉冲发生器产生的高频、高压引弧脉冲若进入单板机,甚至能使单板机的IC损坏。

为确保系统稳定、可靠运行,必须采取抑制干扰信号导入主机的措施。

1. 在系统中,无论是PIO还是CTC,在其与外部设备的连接中,都设计了光电隔离电路。

2. 能够产生强干扰的设备,与单板机各用不同相电源,可以减少通过电源引入的干扰。试验发现这是一种很简便实用的抑制干扰的措施。

3. 各种通用的屏蔽措施是不可缺少的。在我们设计的这个系统中,由于干扰太强,特为单板机和信号采集系统设计了互相绝缘的双层金属壳,外层接设备地,内层接数字

# PLC可编程控制器在造纸低浓盘磨机中的应用

福州大学

施能民

**摘要** 本文重点介绍了PLC可编程控制器在造纸低浓盘磨机中的应用。由于对低浓盘磨机的电气控制系统用PLC可编程控制器软件实现,大大增加了可靠性,取得了较好的经济效益。

## 一、前言

造纸盘磨机是造纸生产车间的关键设备,它是把从化浆车间送来的松木原纸浆进一步加工磨细;制成更均匀、更纤细的纤维,给纸机提供合格的纸浆。一旦它出了故障,则将引起造纸车间整条生产流水线全部停产。

青州造纸厂造纸车间是拥有一套日产150吨的纸机。其配套的造纸低浓盘磨机是由

一台3000kw的三相同步电动机拖动,它的电气控制系统是采用繁多的继电器和接触器控制系统。由于继电器接触器电气控制线路复杂和联锁触点繁多。常常因为触点接触不可靠,或线路松脱引起整条造纸机生产线停机,造成重大的经济损失。

同时这套低浓盘磨机的拖机电机是三相同步电动机,因此在它启动时需靠一个准确的时间继电器进行延时后,将其转子励磁投入,如延时出现偏差,则将使三相同步电

取得了很好的抑制干扰的效果。

## 四、程序流程图

程序流程图分为两部分。

1.状态流程图见图5。

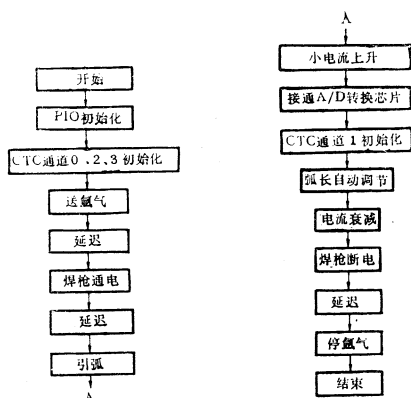


图5

2.弧长自动调节程序流程图见图6

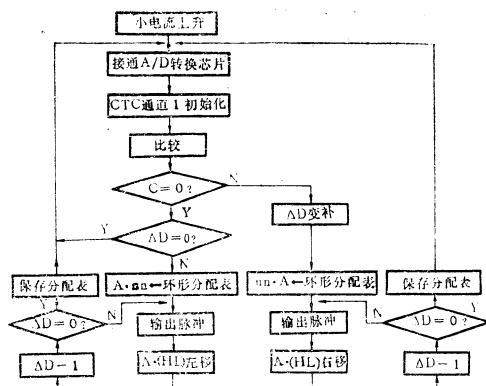


图6

## 五、结束语

实验证明,利用微机可以方便地进行氩弧焊接工艺过程的自动控制和焊接参数控制。由于采取了必要的抗干扰措施,系统工作稳定可靠。该系统现已投入使用,并已取得了较好的效果。

## 参考文献

《微型计算机硬件软件及其应用》。周明德  
清华大学出版社, 1984。



动机失步。

综合上述造纸低浓盘磨机电气控制的工艺要求，采用PLC可编程序控制器能较好地满足其控制要求，而且功能准确可靠、体积重量小，它能把繁锁的、最容易出故障的继电器和接触器触点之间的相关的逻辑关系，用PLC可编程序控制器的软件来实现，而且它可以根据工艺要求的变动，方便地修改电气控制逻辑关系软件，从而达到变动了的工艺要求。

## 二、低浓盘磨机电气控制要求

1. 三相同步电动机定子接电后其转子经过一段准确地延时，将其励磁绕组投入直流电源。

2. 为了确保三相同步电动机的安全可靠运行，其电气控制线路中拥有：主机定子绕组过流和过热保护、转子励磁电路欠压保护、励磁电路过热保护、励磁电路中整流二极管故障保护、失磁保护以及主机润滑油系统的联锁保护等。

3. 三相同步电动机定子绕组靠一台高压油开关控制电源的输入与断开，油开关本身具有合闸线圈动作后自锁的功能，其断开必须接通油开关的跳闸控制线圈。

4. 为了操作人员的工作需要，在操作台上需要提供各种信号灯指示，以便告知操作人员机器设备运行是否处于正常状态。并且可以辨别机器设备出故障的方向和发生原因。

## 三、造纸低浓盘磨机

### 电气控制系统原理图

1. 继电器—接触器控制系统如图1
2. 低浓盘磨同步电动机主电路如图2
3. 低浓盘磨机信号指示电路如图3

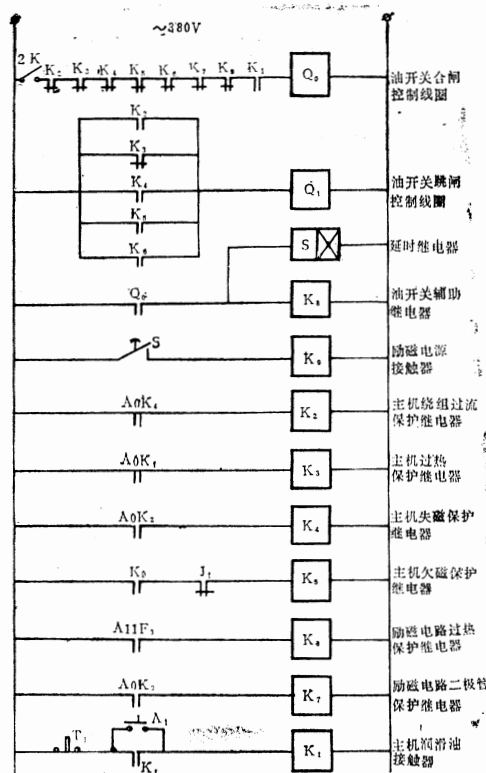


图1 继电器—接触器控制系统图

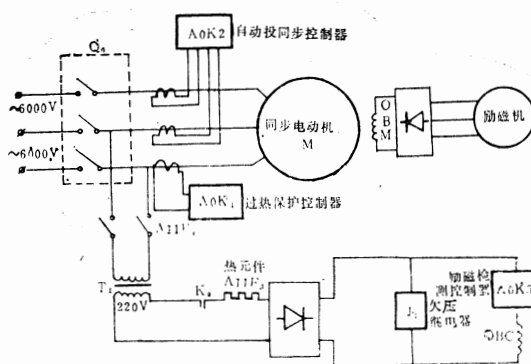


图2 低浓盘磨同步电动机主电路

## 四、低浓盘磨机电气

### 控制PLC梯形图

根据低浓盘磨机的控制要求，设计PLC电气控制系统梯形图如图4所示。信号指示梯形图如图5所示。这样利用PLC可编程序控制器内部的软件系统，实现了对低浓盘磨机的电气控制。

# 一种基于传感器的机器人反馈控制 方案及实验研究

天津市电子计算机研究所 苑静中

**摘要** 本文经过分析比较,提出了对一般工业机器人不作任何改动,利用示教盒通讯口作为外部传感器信号反馈通道的控制方案,并具体介绍了建立这种反馈通道的测试方法和分析结论。作者采用这种控制方案实现了一个基于六维腕力传感器FS6—120A的PUMA机器人力反馈系统。以“零力示教”、“硬笔画线”为例的实验研究表明了上述控制方案的可行性和良好的性能。

## 一、问题的提出

工业机器人向智能化发展的研究基础首先在于配置外部传感器,并把经过处理的传感信息引入控制系统,这里的关键问题是建立合适的反馈信号通道。

例如,为使机器人能完成抛光、研磨、装配等操作,需要在机械手末端安装腕力传感器,以实现力与位置混合控制;又如,使用视觉装置可以测定机器人操作对象的位置和姿态,根据测定结果导引机器人接近目标,但目前一般的工业机器人都没有现成的反馈通道引入力信号或视觉信号,也没有软件支

## 五、结束语

随着微型计算机的不断发展,PLC可编程控制器的功能将不断地改善。目前PLC可编程控制器除了能对开关量控制外已经拥有对模拟量的运算和控制功能。将PLC编程控制器引入造纸机控制系统,必将进一步改善和提高纸机运行的可靠性和运行性能,并且可以进一步节约纸机电气控制系统的投资和

维修费用。对于一些电气控制的改造提供一个良好可行的途径。

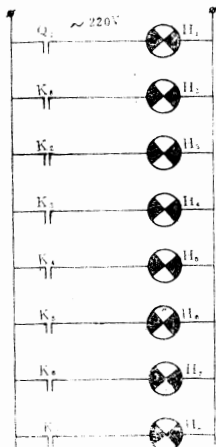


图3

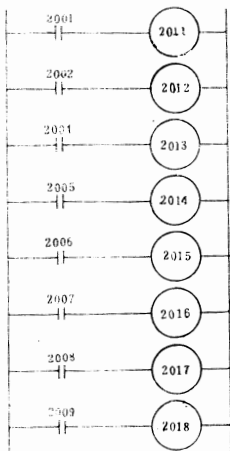


图5

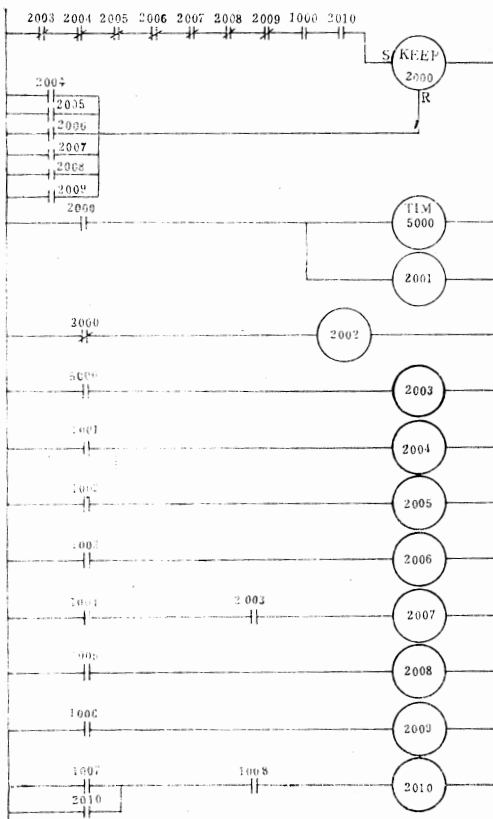


图4

持这些感觉信号的运算处理。

考虑到一般的外部感觉信号最后都需要转换成机器人的位姿控制量,以便与目前机器人的运动学控制方式相符,感觉信号的反馈通道可以从下列三种可能中进行选择。

1. 扩充或改造原有的机器人编程控制系统。以PUMA560机器人为例,把传感器看作机器人系统的外部设备——示教盒、终端、软盘驱动器一样,再为它增加一个串行接口,与主控处理器PDP-11/23相连(见图1),同时,扩充原有VAL软件系统,增加系统的语言结构和控制算法,才能形成通用、规范的命令语句,工作量显然很大。更好一点的方法是将感觉信号进行转换后加入关节伺服级,可使系统的响应速度加快,这样,硬件改造又成为主要工作,难度相当于另外设计关节伺服系统。

2. 利用终端通讯口作为反馈通道。由于终端送往主控处理器的信息是键入的VAL命令语句。因此,首先必须把感觉信号经过另外的处理器进行转换处理,生成速度、定位及运动语句的字符ASCII码,送往终端接口,这种方法虽然简便、直观,但实时性太差。例如,修改一个示教点,共约60个字符,如果终端口波特率为9600,再加上解释转换时间,那么调整机器人末端动作一次就需要100MS以上。

3. 利用示教盒通讯口作为反馈通道。这样引入感觉信号虽然与方法2类似,但由于主控处理器从示教盒接收的信号已是设定的座标系中位置、姿态的控制量,如果传感器的输出经过微机转换处理成同样类型的反馈信号,那么传感器的数据量大大少于VAL命令语句字符的数据量,再加上省去命令的解释过程,实时控制的可能性明显增强。

相比之下,上述方法3对原有机器人控制系统可以不作任何改动,在简便易行与实时性之间作出了合适的折中,我们选择这种方法进行了实验研究工作,并以配置六维腕力传感器的PUMA机器人为例,通过“零力示

教”、“硬笔画线”等典型动作,验证了系统的可行性。

方法3的研究工作,首要的问题是用微机模拟机器人示教盒的运动控制功能。

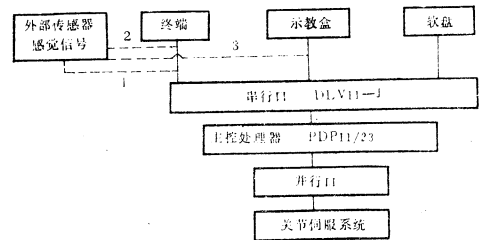


图1 PUMA机器人控制系统结构图

## 二、示教盒功能的模拟

机器人示教盒实际上主要是一个手动控制器,通过按钮设定Joint、Tool、World三种座标系之一,通过Speed旋钮设定速度值,然后扳动六个运动方向开关之一,机器人手臂就可以实时地动作。六个运动方向开关任意组合,同时扳动,机器人末端就能完成复杂的轨迹。运动过程中,速度旋钮可以连续变化,座标系也可以随时切换。

示教盒内部结构主要是一个8748微处理器,接收来自示教盒面板上按钮开关的信号,调用EPROM中的固化程序进行处理,通过串行口向机器人主控制器发送一组运动数据,主控制器则回送机器人运动状态、限位错误等显示数码。

要用另外的微机编程模拟示教盒的运动控制功能,必须分析清楚:a.示教盒向主控制器发送的运动数据的编码格式、编码内容及其物理意义;b.主控制器向示教盒发送的显示信息的编码格式和内容;c.两者相互通讯的同步方式,传送周期。由于目的在于外部功能的模拟,我们只把对示教盒内部硬件、软件的分析作为参考,采用下述两种方法“截获”了示教盒发往主控制器的运动数据信息:

1. 模拟VAL系统中示教盒中断服务程序。即,另写一段中断服务程序,把示教盒



设定一组开关的输出信息另外存入主控器的RAM指定单元,每中断一次,读写一个数据,连续读写,就可以分析读写数据的周期规律,再把这些数据与示教盒开关意义反复比较,最后分析出数据的格式和相应的物理意义。

**2. 实测示教盒发送信息。**在VAL系统的监控状态下,按动示教盒某个开关,键入“DONE”命令,在系统的ODT状态下读出示教盒发送信息原来存放单元的内容,就可以得知对应的运动数据。

通过上述两种方法所得结果的相互验证,表明:

(1) 示教盒发出的运动数据共有9个字节,分别表示速度、座标系和运动方向。

(2) 主控器与示教盒之间的通讯周期即为VAL系统的运动控制周期(28ms)。

(3) 每当示教盒收到主控器发来的数据后,随即发出运动数据,两者以此作为同步信号。

根据分析的结果,我们成功地在IBM-PC/AT上编制了示教盒功能模拟程序,通过键盘操作,不但可以实现示教盒面板上所有开关、按钮、指示灯的功能,而且由于是编程实现,因而可以显示、监测更多的通讯信息,增加了示教盒功能的透明度。

“软件示教盒”的实现,为机器人感觉信号的反馈控制敞开了通道,同时又保留了对机器人手动控制方式。

### 三、基于感觉信号的反馈控制方案

研究机器人智能控制的典型实验是给原有系统加入力觉反馈,我们把美国Barry Wright公司研制的FS-120A六轴腕力传感器

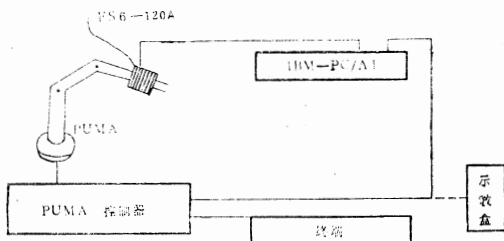


图2 实验系统构成

配置于PUMA机器人手臂末端,按照前述反馈通道方式,构成了一个实验系统(见图2)。

FS-120A腕力传感器集信号转换、数据的模/数、数/模转换及解耦运算等功能为一体,具有模拟量、数字量、字符ASCII码三种形式,能以38400、19200、9600等多种波特率与外界通讯,一旦接收到外界发来的ASCII码命令字符,就可周期性地输出,最短的采样处理周期可达2.08ms。我们选用了通讯波特率为9600、格式为16字节数字量的成组数据输出方式,其中12个字节为传感器直角坐标系中的6个力(力矩)分量,其余4个字节用作同步检测或出错标记。

利用PUMA-560示教盒串行口作为力反馈控制通道,PC-AT机接收用户输入的控制要求及腕力传感器发来的感觉信号,运行力控制算法后将新的位置控制量传给PUMA控制器,实现闭环控制。通过示教盒口发给PUMA控制器的位置控制要求实际上仅仅是速度矢量,速度方向可以相对三种座标系来描述,即定义在机械手末端并随机械手一起移动的Tool座标系;位于机械手底座的World座标系和关节空间。速度矢量允许在两个控制周期之间任意切换。PUMA控制器负责对速度矢量进行逆运动学求解、生成关节空间轨迹、然后由控制器的下位机进行位置伺服。反馈控制系统框图见图3。

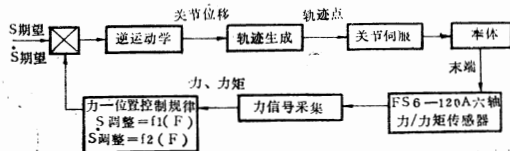


图3 控制系统框图

整个反馈控制系统的通讯时序关系如图4所示。

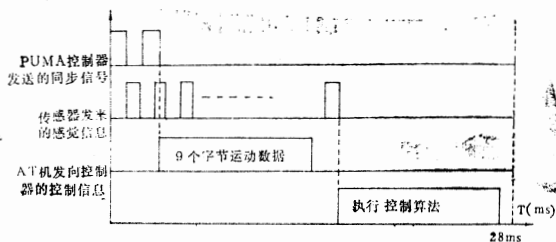


图4 时序关系图

## 四、实验研究

在前述的示教盒功能模拟程序中，我们插入了力（力矩）—运动控制量的转换算法。并对两个比较典型的柔性动作进行了实验。

### 1. 零力示教

所谓“零力示教”是与位置示教相对而言的，我们知道，通常机器人系统都提供一个手动示教盒，扳动示教盒上的开关、按钮、可以控制机器人在工作空间内移动，并记录移动过程中走过的位姿点，以便机器人能够自动重复这种动作。对于配备了腕力传感器的机器人，如果再采用这种方式进行示教就显得太“笨拙”了。因此，采取用外力作用于机械手进行示教。所谓“零力”是指这样一个过程，从机械手的角度讲，它总是试图保持与外界的相互作用力为零。

为了实现这样的控制目标，机械手末端需沿作用力（力矩）方向移动，避开外力的作用。由于示教盒输出运动数据的特点是，在某种坐标系下，末端各方向的运动速度都是同一个设定值。因此，控制算法中必须解决的有两个问题：

（1）保证使受力（力矩）方向与运动方向一致。以二维力为例（图5），当检测到 $x$ 、 $y$ 方向都有力分量时，模拟程序发往机器人控制器的运动信号是使 $x$ 、 $y$ 方向同时等速运动，而结果与要求末端按受力方向运动的期望不一致，即末端受力方向与其运动速度方向不一致，因而，必须在检测到受力分量后求解受力大小的均方根 $\sqrt{F_x^2 + F_y^2}$ ，再确定 $x$ 、 $y$ 方向上不同的运动时间间隔，进行插补计算，以满足一致性。

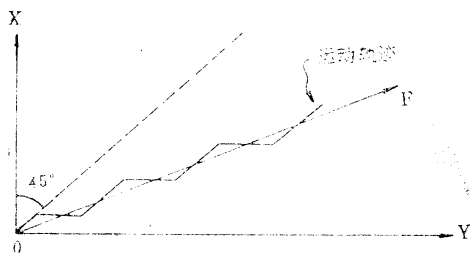


图5 二维折线插补

（2）区别受力大小不同的情况，平滑地调整速度设定值，例如，受力大，运动速度也加快，以满足末端运动的柔性控制。

### 2. 硬笔画线

硬笔画线是指在机械手末端安装一个硬笔，控制它在任意的平面上画出线段来。画线过程与零力示教相比有较大差别，其过程可分为三个阶段：

①接近阶段

②画线阶段

③离开阶段

接近阶段是指机械手从空间某一点移动到与画线平面相接触的位置，这个阶段属于纯位置控制过程，即控制机械手沿Tool座标系的Z轴方向移动，在移动过程中检测是否机械手末端与平面发生接触，一旦与平面接触上，则进入画线阶段。画线阶段是力—位置混合控制过程，除了要保证机械手末端Tool座标系的Z轴方向与过接触点的切平面垂直外，还要沿Z轴与接触点维持一恒定大小的接触力（力矩）。在这种接触状态下，控制机械手沿画线方向移动，在移动过程中，随时对机械手的位姿进行检测，不满足有效接触时进行机械手的位姿调整。第三个阶段比较简单，当画线完成时，将机械手沿Tool座标系的Z轴负方向移动一段距离即可，这也是位置控制过程。画线作业控制流程见图6。

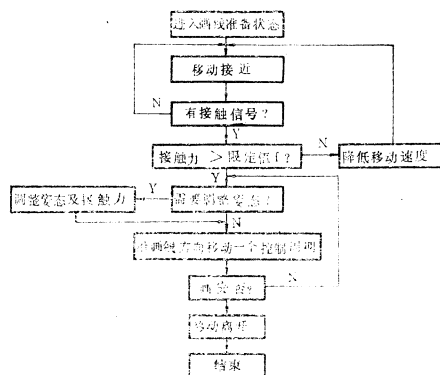


图6 画线作业控制流程图

实验结果表明，在机械手末端（腕力传感器）姿态不变的情况下，末端能平滑迅速下转（第47页）

# 计划编制系统的设计与实现

郑州机械专科学校 刘辉 赵青玲 吴振庆

**摘要** 本文介绍了在PC机上开发的一个运用投入产出法编制产品计划的“计划编制系统”，描述了该系统所采用的数学模型、算法、设计思想、软件结构及功能。

投入产出法又称联系平衡法，由美国经济学家W. Leontief提出，它作为一种比较成熟的经济分析方法，在世界各国得到极其普遍的推广和应用，并在经济分析和计划工作中产生了一定的效果。W. Leontief 因此而荣获诺贝尔经济学奖金。

投入产出模型是经济数学模型的一种，运用投入产出模型进行计划编制，会涉及到大量复杂和繁琐的计算，基于此，我们设计开发了该“计划编制系统”。

## 一、数学模型及算法

### 1. 原始数据的采集

把企业的产品按序编号，再把它们纵横交叉叠加在一起，就构成一个概括企业全部投入产出关系的棋盘式平衡表，表格的横向说明了产品按使用价值在补偿生产消耗和最终产品方面的分配情况，纵向反映了产品的消耗构成。它由计划人员根据已经发生的企业内部各部分生产及产品间的数量联系而绘制。表1为某钢铁联合企业报告期投入产出表。

此时

$$A = \begin{pmatrix} 0 & 2.11059 & 0 & 0 & 0.139996 & 0 & 0 & 0 \\ 0 & 0 & 1.0050 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2.0000 & 0.120596 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.009968 & 0 & 0 & 0 \\ & & & \dots & & & & \\ 0.0081 & 0.0926 & 0.013159 & 0.01067 & 0.01451 & 0.02351 & 0.06266 & 0.0031 \end{pmatrix}$$

### 2. 直接消耗系数矩阵的生成

直接消耗系数分为两部分：一是生产单位产品j对产品i的直接消耗系数 $a'_{ij}$ ，另一是生产单位产品j对i种能源(燃料或电力)的直接消耗系数 $a''_{ij}$ ，它直接反映了产品之间的直接生产联系和消耗结构，计算数学表达式为：

$$a'_{ij} = \frac{x_{ij}}{x_j} \quad (i, j=1, 2, \dots, n) \quad (1)$$

$$a''_{ij} = \frac{h_{ij}}{x_j}$$

$$(i=1, 2, \dots, m, j=1, 2, \dots, n) \quad (2)$$

式中，n为企业生产的产品种数，m为所耗费的能源种数， $x_{ij}$ 为产品之间流量，亦即第j种产品对第i种产品的消耗量， $h_{ij}$ 为第j种产品对第i种能源的消耗量， $x_j$ 为第j种产品的年总产出量。

若用A表示直接消耗系数矩阵，则：

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{t1} & a_{t2} & \dots & a_{tn} \end{pmatrix}$$

$$(t=1, 2, \dots, n+m)$$

如表1所示， $n=8$ ， $m=3$ ，则：

$$a_{11} = 0/2954.7920 = 0$$

$$a_{12} = 2866.3890/1358.4403$$

$$= 2.11059$$

...



表1 企业投入产出表

	铁矿石	铁精石	烧结石	生铁	钢、锭	钢坯	钢材	废钢铁	商品量	总产量
	1	2	3	4	5	6	7	8	9	10
铁矿石	0	2866.3890	0	0	88.4030	0	0	0	0	2954.7920
铁精石	0	0	1358.4403	0	0	0	0	0	0	1358.4403
烧结石	0	0	0	1275.5290	76.1529	0	0	0	0	1351.6819
生铁	0	0	0	0	637.7645	0	0	0	0	637.7645
钢锭	0	0	0	0	0	528.3385	0	3.1343	99.9970	631.4698
钢坯	0	0	0	0	0	0	390.2096	50.0725	0	440.2821
钢材	0	0	0	0	0	0	0	25.1747	300.0000	325.1747
废钢铁	0	4.3870	0	24.1906	89.4765	0	0	0	0	118.0541
自产燃料	0	15.3758	50.4283	382.7171	17.3149	17.6201	57.6535	0	0	522.1097
外购燃料	0	0	79.8257	64.7331	69.7143	2.4308	12.5582	0	0	229.2621
外购电力	23.9338	125.7774	17.7871	6.8049	9.1626	10.3510	20.3754	0.3660	0	214.5582

### 3. 完全消耗系数矩阵的生成

完全消耗系数全面地反映了产品之间的直接和间接消耗关系。根据投入产出原理，完全消耗系数矩阵的计算公式可以从直接消耗系数矩阵A导出，它亦由两部分构成：

第一部分是产品间的完全消耗系数矩阵B'，其算法为：

若用I表示 $n \times n$ 单位方阵，用 $(I-A)^{-1}$ 表示两个矩阵相减后的逆阵，则 $B' = (I -$

$A)^{-1} - I$ ，此时A矩阵的 $i, j$ 取值范围均为： $i, j=1, 2, \dots, n$ 。

另一部分是产品对能源的完全消耗系数矩阵B''，其数字模型为 $B'' = A \cdot (B' + I)$ 。此时A矩阵的 $i, j$ 取值范围为： $i=n+1, n+2, \dots, n+m, j=1, 2, \dots, n$ 。

对于上述已知A矩阵，经过矩阵的减法运算、求逆运算和乘法运算后，求得的完全消耗系数矩阵B为：

$$B = \begin{bmatrix} 0 & 2.12517 & 2.1358 & 4.44911 & 5.55415 & 6.66498 & 7.997978 & 4.679951 \\ 0 & 0.0069 & 1.01198 & 2.105576 & 2.553474 & 3.064169 & 3.677002 & 2.151567 \\ 0 & 0.0069 & 0.0069 & 2.095101 & 2.540771 & 3.048925 & 3.658709 & 2.140864 \\ 0 & 0.00326 & 0.00328 & 0.04187 & 1.19881 & 1.438575 & 1.72629 & 1.010124 \\ & \dots & & & & & & \\ 0.0081 & 0.11074 & 0.1215 & 0.2720634 & 0.352064 & 0.445986 & 0.597843 & 0.3291 \end{bmatrix}$$

对于求逆中行列式的值的计算。本系统采用传统的高斯消去法。

上 $Y_k$ ，然后把所得的每一行的结果相加，便得到计划期该行的总产量。

### 4. 计划期总产量计算表的建立

根据企业在计划期预计的对外销售量即商品量，如假定对外销售序号为K号产品的销售量为 $Y_k$ ，则用 $Y_k$ 去分别乘以矩阵B中第K列的诸元素，第K行、K列的乘积需再加

据本文上面所例举实例，假设该联合企业预定在计划期对外销售序号为4、5、6、7这四种产品，它们的商品量分别为5，105，10，320，那么：

$$5 \times 4.44911 = 22.246$$

$$105 \times 5.55415 = 583.186$$

$$10 \times 6.66498 = 66.65$$

$$320 \times 7.997978 = 2959.353$$

$$22.246 + 583.186 + 66.65 + 2959.353 = 3231.434$$

(本例计算选择精确到小数点后三位) 依此类推，即可建立计划期总产量表。

表2为本系统根据以上数据输出的计划期总产量计算表。

表2 计划期总产量计算表

		生 铁	钢 锭	钢 坯	钢 材	总 产 量
		4	5	6	7	
铁矿石	1	22.246	583.186	66.65	2559.353	3231.435
铁精石	2	10.528	268.115	30.642	1176.641	1485.926
烧结石	3	10.476	266.781	30.489	1170.787	1478.533
生 铁	4	5.224	125.875	14.386	552.413	697.898
钢 锭	5	.222	124.633	14.244	546.96	686.059
钢 坯	6	.179	15.845	11.811	453.538	481.373
钢 材	7	.056	4.969	.568	341.806	347.399
废钢铁	8	.264	23.3	2.663	102.255	128.482
自产燃料	9	3.513	89.707	10.652	465.789	569.661
外购燃料	10	1.177	42.57	4.92	201.301	249.968
外购电力	11	1.36	36.967	4.46	191.31	234.097

## 5. 企业计划期投入产出表的建立

据公式(1)、(2)可导出如下公式:

$$X_{ij} = a_{ij} \cdot X_j \quad (i, j=1, 2, \dots, n)$$

$$H_{ij} = a_{ij} \cdot X_j \quad (i=n+1, n+2, \dots, n+m, j=1, 2, \dots, n)$$

式中,  $X_j$  为计划期第  $j$  种产品的年总产量,  $X_{ij}$  为计划期第  $j$  种产品对第  $i$  种产品的消耗量,  $H_{ij}$  为计划期第  $j$  种产品对第  $i$  种能源的消耗量。

显然, 依据表2计算得出的计划期总产量指标, 与矩阵A中相应位置的数相乘, 即可得计划期投入产出表。

表3为本系统输出的企业计划期投入产出表。该表全面地反映了计划期企业的经济状况, 是物资材料供应部门不可缺少的资料。

表2、表3中的数据单位均与录入的计划期商品量的单位一致。

## 二、系统的设计思想与实现过程

### 1. 系统运行环境

本系统以CCDOS2.1为汉字支持软件, 在IBM PC及其兼容机上开发并实现, 既可运行于硬盘系统, 也可运用于软盘系统, 要求内存在512kb以上, 所用打印机为24针宽行打印机。

### 2. 系统结构

本系统分两大模块: 一是原始数据的管理模块, 另一是最终需求表的计算输出模块。

#### (1) 原始数据的管理

原始数据以文件的形式存放于数据文件中, 本系统共设置了5个数据文件。

$\Lambda_1$ 、 $\Lambda_2$ 数据文件: 用于存贮栏目名称表1和栏目名称表2中的数据。

由于各企业生产的产品数量与名称各不相同, 从而影响到投入产出表的栏目不固定, 因此, 考虑到程序的通用性, 建立了两个栏目名称表, 栏目名称表1存放企业的产品名称, 栏目名称表2存放企业所需用的能源的名称。

$B_1$ 、 $B_2$ 数据文件: 这是两个过渡文件, 分别用于存放字段结构装配表1和字段结构装配表2的数据。字段结构装配表实质上反映的是栏目名称表中每一字段的长度及名称数量。

C数据文件: 用于存放投入产出表中的数据, 即存放投入与产出的数量关系。

以上数据在录入时, 均采用容错技术对误输的数据进行重新修订。

#### (2) 最终需求表的计算输出

运行该模块时, 系统通过对数据文件 $B_1$ 取值, 得到字段结构; 通过对数据文件 $\Lambda_1$ 取值, 得到产品名称; 同样, 通过对数据文件 $B_2$ 取值, 得到字段结构; 通过对数据文件 $\Lambda_2$ 取值, 得到能源名称。

综合所得到的两类字段结构, 此后系统即可访问数据文件C, 从中取出投入产出表的原始数据。依据原始数据及上述算法, 系统逐次计算生成直接消耗系数矩阵和完全消耗系数矩阵, 最后生成并输出计划期总产量计算表(如表2)和计划期投入产出表(如表3)。

在生成完全消耗系数矩阵时涉及到了比较复杂的求逆运算, 本系统以高斯消去法求行列式为一通用子程序, 分别计算各相应行列式的值, 然后依据矩阵求逆的数学计算方法, 求得完全消耗系数矩阵。

在输出计划期总产量计算表和计划期投入产出表时, 本系统可自动调整表格的各列宽度, 以使输出的表格美观、大方。为扩大程序的通用性, 还作以下规定: 如果表格的总列数大于8, 则设置打印机的总宽度为132, 否则为80。

### 3. 系统功能

系统工作流程如图1所示。具有以下功能。

(1) 本系统采用汉字信息提示、功能菜单选择等增强人机对话功能的方法, 引导用户正确使用, 并提供了较强的容错功能。

表 3 企业计划期投入产出表

	铁矿石	铁精石	烧结石	生 铁	钢 锭	钢 坯	钢 材	废钢铁	商品量	总产量
	1	2	3	4	5	6	7	8	9	10
铁矿石	0	3135.391	0	0	96.045	0	0	0	0	3231.436
铁精石	0	0	1485.925	0	0	0	0	0	0	1485.925
烧结石	0	0	0	1395.796	82.736	0	0	0	0	1478.532
生 铁	0	0	0	0	692.898	0	0	0	5	697.898
钢 锭	0	0	0	0	0	577.048	0	3.411	105	686.059
钢 坯	0	0	0	0	0	0	416.879	54.495	10	481.374
钢 材	0	0	0	0	0	0	0	27.389	320	347.398
废钢铁	0	4.799	0	26.471	97.212	0	0	0	0	128.482
自产燃料	0	16.819	55.161	398.011	18.812	19.265	61.594	0	0	569.662
外购燃料	0	0	87.317	70.837	75.741	2.658	13.416	0	0	249.969
外购电力	26.175	137.581	19.456	7.447	9.955	11.317	21.768	.398	0	234.097

# 面向问题的自动编程

南开大学 朱瑞香 袁晓洁

**摘要** 本文介绍了一个面向问题自动编程系统的设计与实现,阐述了由问题描述到目标程序生成的全部过程,提出了实现程序设计自动化的一种方法。整个系统由领域知识库系统、编程知识库系统、问题理解系统和软件生成系统四部分组成,系统用PASCAL语言编成,在IBM PC/AT计算机上实现。

## 一、引言

程序设计自动化(简称AP)早就是程序设计人员所追求的目标。在电子计算机发展的早期,即五十至六十年代,AP追求的主要是提高编程能力和效率,办法是向程序员提供各种高级编程语言、七十年代出现了数据

库查询语言,并在某些较窄的成熟领域出现了全自动的应用程序生成器。八十年代,AP的含义更接近于自然,它所希望达到的目标是:计算机能根据用户的问题描述,自动进行推理,生成所需要的目标程序,并且自动调试、自动排错和优化等,完成人们所期望的工作。

久性保存。存盘时,若是重新录入数据,系统会自动删除盘中原有的数据文件。

(3) 对数据文件访问后,可计算出A B矩阵,最后生成两个最终需求表并可根据用户需要输出若干份。

(4) 系统的通用性较强。交互式技术的使用、数据文件的结构特征、表格输出时列宽比例的自动变化,等等。

(5) 在保证精度的条件下,用户可根据需要选择小数点后的精确位数。

(6) 具有较快的运行速度。本文所举实例仅用七分钟运行,即可完成全部计算、输出功能,远非人工所能比拟。

该系统的程序调试采用了由下向上分层分步的调试技术,运用了许多程序调试方法,诸如强制技术、断点打印、功能模拟等,系统的各项要求均能完全满意地实现。

## 参考资料

- 1 钟契夫、乌家培等编,《投入产出原理及其应用》,中国社会科学出版社,1982年10月出版。
- 2 冯文权主编,《经济预测的原理与方法》,武汉大学出版社,1986年12月出版。

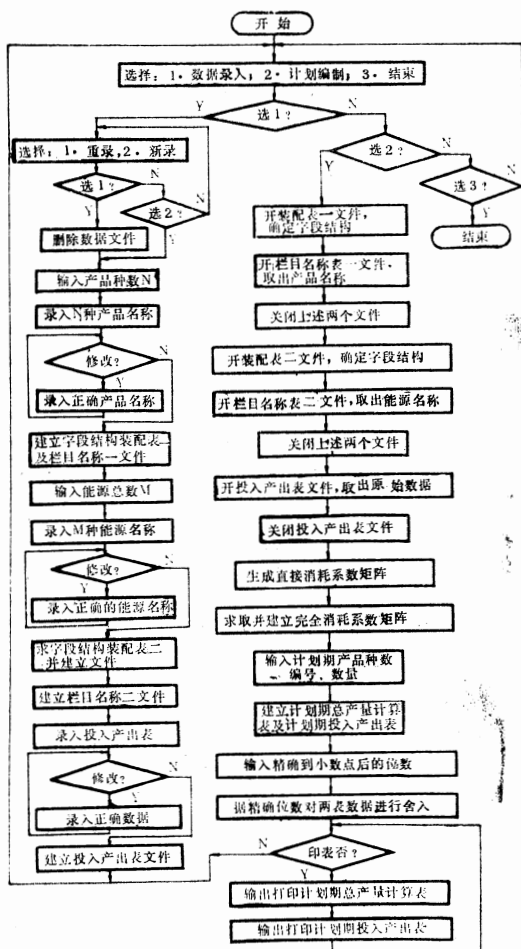


图 1

(2) 原始数据存放于文件中,便于永



但就当前的理论水平和技术水平，完全达到程序生成自动化还相当困难，因为它涉及到丰富而又多种多样的知识和经验，不过人们正逐步向该目标迈进。目前，人们利用人工智能、知识工程、专家系统等技术建造了一些系统<sup>[1][2][3]</sup>，得到了一些重要成果，但离彻底实用还有一段距离，大部分结果还处于有待进一步完善和提高了的实验阶段，并且都是面向具体领域的，其应用范围往往非常窄。

我们在深入学习对他人经验的基础上，提出了一种面向问题的AP方案，并据此开发了一个系统。该系统经在某些具体领域上的实验，证明是可行的，结果还是令人满意的。下面着重介绍一下系统的设计思想与结构。

## 二、系统概述

所谓面向问题的自动编程就是用户使用某种类自然语言给出要解决问题的非形式化描述，系统利用各种知识对问题的描述进行理解、分析、匹配推理，寻找出恰当的算法，最后生成解决问题的程序。在此过程中，如遇到无法解决的问题，就与用户进行交互，寻找解答并学会该种解法。

我们的系统就是基于上述理解设计的，它的目标就是由用户输入简单的问题描述，由系统自动生成相应的程序。整体结构如图1：

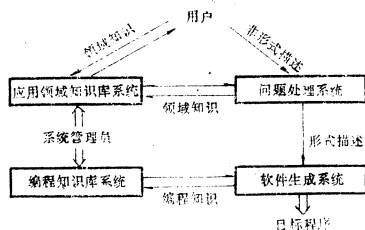


图1 系统结构图

其中各子系统的功能如下：

1.应用领域知识库系统：以词汇和语法结构为基础贮存应用领域的知识，并把这些

知识映射成系统能识别的形式描述。其组织形式采用分块结构，通过扩展块结构系统具有应用的广泛性。

2.问题理解系统：负责理解用户提出的编程要求，对其进行合法性检查，通过对词法、语法、语义处理，提取有关编程信息，形成问题的形式描述。

3.软件生成系统：它从形式描述中得到事实，从编程知识库中找到相应的规则，进行匹配、推理，逐步求精，从而生成最佳软件。

4.编程知识库系统：存放用于生成程序的各种编程知识和软件开发知识，它的建立与维护由系统管理员负责完成。

## 三、问题描述及词法理解

### 1.描述语言

描述语言实际上是系统与用户的接口，因此它要满足来自用户和系统两方面要求。其一，要使用户能方便地使用它，完备地表达各种各样的设计思想和要求；其二，要足够的规范化，使系统能无二义性地接受它、理解它，并在此基础上高效率地生成目标程序。由于目前自然语言的理解还是一个尚未攻关的难关，因此要最大限度地满足用户还有相当的困难，所以我们设计描述语言的基本思想还是先从系统角度出发，首先考虑系统设计要求，综合系统可以识别的结构模型，确定系统可以达到的对语言的理解程度，其次再考虑语言结构尽可能地满足人们的习惯，使用户能够方便地利用它表达出各种信息。基于系统的功能要求，其自然语言由下列五种语句组成：

(1) “(输入)”[〈数词〉]〈主体〉{“,”[〈数词〉]〈主体〉}。“.”

(2) 〈主体操作〉“.”

(3) “如果”〈条件描述〉“那么”〈主体操作块〉[“否则”〈主体操作块〉]“.”

(4) “当”〈条件描述〉“则”〈主

体操作块〉“.”

(5) “重复”〈主体操作块〉“直到”〈条件描述〉“.”! 其中双引号括起来的是终极符, 方括号括起来的是可以缺省的部分, 尖括号括起来的是非终极符。各非终极符的规定如下:

〈主体〉::=〈修饰语〉〈客体〉

〈修饰语〉::=〈修饰词〉“的”|〈修饰词〉“、”〈修饰语〉|〈修饰词〉“的”〈修饰语〉

〈修饰词〉::=〈名词〉|〈形容词〉

〈客体〉::=〈名词〉|〈名词〉“、”〈客体〉

〈数词〉::=〈数〉|〈数〉〈量词〉

〈主体操作〉::=[〈副词〉]“ (“ 〈动词〉 “)” 〈主体〉 | [〈主语〉] “ (“ 〈动词〉 “)” 〈主体〉

〈主语〉::=〈主体〉|〈数词〉

〈主体操作块〉::=〈主体操作〉|〈主体操作〉“、”〈主体操作块〉

〈条件描述〉::=〈主语表达式〉〈比较词〉〈主语表达式〉

〈主语表达式〉::=〈主语〉|〈主语〉〈运算符〉〈主语表达式〉

定义中用到的非终极符号〈名词〉、〈动词〉、〈形容词〉、〈副词〉、〈运算符〉、〈比较词〉等为人们通常理解的常规概念。

例: (输入) 50名学生的姓名, 学生的数学、物理、化学的成绩。(计算) 学生的平均成绩。如果学生的平均成绩 $<60$ 分那么(打印) 学生的姓名、成绩。(求出) 学生的总成绩。从小到大(输出) 学生的总成绩、姓名。

从上述描述语言的规定可以看出, 它基本符合人们书写自然语言的习惯, 用户可以用很方便地使用它描述所要求解的问题。

## 2. 语言词库

为了使系统能理解描述语言, 建有一个庞大的应用领域知识库。由于编程的应用领域相当广, 相应的知识也就较多, 故采用分

块结构形式进行组织, 把公用知识组成核心块, 针对某个领域的建成应用领域块, 这样组织一方面便于扩充, 另一方面当应用于某一领域时, 只需装入核心块与相应的领域块即可, 就可以避免信息的冗余。该应用领域包括语言理解和语法推理两个知识库。语言理解库由名词、动词、形容词、副词四个子库组成, 其中以名词库的规模为最大。通过大量应用领域知识的分析, 我们发现处理语言描述主要是集中理解语言中的各种词汇, 并为其在计算机内配以适当的结构或操作, 使之能够加入到各种运算中, 从而得出目标结果。因此, 词库的建立要体现出这些特征。系统中的各种词库均用框架形式表示, 词汇的总框架形式如下:

词名: (词的名字, 同义词均可列入)

词性: (名词, 形容词, 动词, 副词之一)

结构类: (该词在计算中可充当的角色)

前词词名: (可有的前缀词)

含义: (可对应的样本程序名)

⋮

后词词名: (可有的后缀词)

含义: (可对应的样本程序名)

⋮

### 例 1

词名: 成绩、分数、分

词性: 名词

结构类: real

前词词名: 平均

含义: average (real)

前词词名: 总

含义: sum (real)

### 例 2

词名: 数学、物理、化学...

词性: 名词

结构类: 无

### 例 3

词名: 打印、输出...

词性: 动词

结构类: output

前词词名: 从大到小、从高到低

含义: lsprint

前词词名: 从小到大、从低到高

含义: slprint

### 3. 语言的分析与处理

问题理解系统对输入的非形式描述要进行两部分处理, 一是语言层的分析, 二是形式描述的生成。在此先介绍一下语言层分析所做的工作。

#### (1) 词法分析

1) 切割各个词汇, 建立当前问题词汇表, 为各个词汇配以变量名;

2) 把数据库中的词汇与词库中的词汇匹配, 如词库中没有, 就与用户进行交互, 确定是否有错或是用户新建立的单词, 若是新的单词, 则由用户提供属性槽值, 扩充词库完成新知识的获取。当词库中有该词时, 取出该词的框架, 存入系统建立的当前问题知识库;

3) 把各个词汇的词性和结构填入当前问题词汇表中。

#### (2) 语法检查

查看各词的类别, 检查它们的组合是否符合描述语言的语法规定, 并检查这种搭配是否合理。

## 四、语义理解及形式描述的生成

这部分工作由问题理解系统来完成。所谓语义理解是要把用户输入的非形式描述转变成形式描述。

### 1. 形式描述框架

根据知识的类别, 分为描述型、处理型、表达式型三种形式描述框架。

(1) 数据对象的内部描述形式: (描述型)

```
attribute : object
name : ...
structure, simple #
           complex #
           index n1 to n2 [, m1 to m2] #
type-of-element: integer #
                real #
                int array #
                real array #
deflist n1 : type1 ;
        n2 : type2 ;
        :
        np : typep #
deftype<typename> #
form-of-access : random # | directly # |
                indirectly #
```

#### (2) 语句的内部描述形式: (处理型)

```
attribute : process
name : ...
action : input<var> #
        output<var> or <value> #
        let<var> : <value> #
        select B1, B2, ..., Bn; C1, C2, ...,
        Cm #
        execute B1, B2, ..., Bn #
        circulate B1, B2, ..., Bn #
        repeat B1, B2, ..., Bn #
        name<val1>, <val2>, ..., <valn> #
precondition: <value> # or none
post condition : <value> # or none
```

#### (3) 表达式的内部描述形式: (表达式型)

```
attribute : express
name : ...
operator : + # | - # | < # | ... | < > #
operand : a1; a2 #
```

### 2. 语义理解

#### (1) 语句中〈主体〉的处理

非形式描述语言中的〈主体〉部分, 按数据对象的描述形式生成。前面提到的应用领域知识库中还有一个语法推理知识库, 该库中存贮的是生成形式描述的知识, 知识是以产生式形式组织的。规则的左部一般是各种结构类、词性搭配方式; 右部是形式描述框架中对应的属性槽值。对于每个主体先根

据词汇的分布情况,形成如下树状结构见图2, (其中同一层儿子结点是以“、”划分的,父与子结点是以“的”划分的)

然后,区分合并同类主体,依据语法推理知识库,建立数据对象描述形式、填充各槽的值。如:学生的数学、物理、化学的成绩。如图3所示。

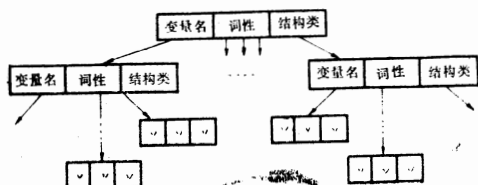


图2

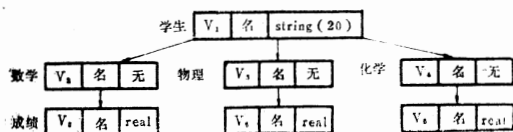


图3

经推理变换形成如下形式:

```
attribute: object
name: ***
structure: complex #
type-of-element: deflist V1: lstring(20);
V2V5, V3V5, V4V5: real #
form-of-access: random #
```

## (2) 语句的处理

由于主体描述已处理完毕,每个主体都有了形式描述的名字,所以语句处理就变得简单多了。语句共有五种形式,均采用自底向上、逐层归类的方法进行形式化。

### 1) 输入类语句

输入对象是主体,因此转化为如下形式:

```
attribute: process
name: ***
action: input<var>#(<var>为主体名表)
```

### 2) 主体操作

根据主语及动词的情况决定采用何种动作,同时若动词后面的主体有特殊含义,还要选择样本程序名及确定参数,通常变换后形式为:

```
attribute: process
name: ***
```

```
action: output<var>or<value>#
let<var>: <value>#
name<val1>, <val2>, ..., <valn>#
(name为样本程序名, <vali>为实参)
```

### 3) 循环、选择语句

先把条件描述形成express 属性的形式描述,然后再把“如果”类,“当”类,“重复”类语句处理成如下形式:

```
attribute: process
name: ***
action: select B1, B2, ..., Bn; C1, C2, ..., Cm#
(如果...那么B1, ..., Bn否则C1, ..., Cm)
execute B1, B2, ..., Bn #
(如果...那么B1, B2, ..., Bn)
circulate R1, B2, ..., Bn #
(当...则B1, B2, ..., Bn)
repeat B1, B2, ..., Bn #
(重复B1, B2, ..., Bn, 直到...)
precondition: <value>#or noue
(如果或当后面的条件)
postcondition: <value>#or noue
(直到后面的条件)
```

总之,经过问题理解系统的上述工作,就把用户输入的非形式描述转换成形式描述。

## 五、目标程序的生成

从形式描述到目标程序的生成是由软件生成系统和编程知识库系统合作完成的。这两个系统的设计与目标语言有密切关系,我们的系统生成的目标程序是PASCAL程序,故系统的知识存贮形式及推理过程都与PASCAL语言直接有关。

### 1. 编程知识库

根据人们的编程经验以及编程处理过程,我们把编程知识总结成三类存于该库中。

(1) 描述型知识: 给出PASCAL语言的数据对象与语句的组成规则,它们是通过分析总结PASCAL语言获得的。采用框架形式表示描述型知识。

(2) 处理型知识: 给出程序的组织规

# 全国联行对帐处理系统

天津市电子计算机研究所 吴锦声 卢文

**摘要** 本文主要介绍天津工商分行计算中心使用的全国联行对帐处理系统, 它能同时完成80个分理处的对帐处理, 功能齐全, 使用效果较好。

## 一、概述

文中对联行对帐系统设计中的主要问题, 系统信息流程, 软件结构都做了简要介绍。

全国联行往来, 是指全国工商行系统内, 经核准有全国联行行号的行处, 在不同省, (市)之间相互资金划拨的帐务往来。它是办理异地结算业务和划拨内部资金的重要工具。目前全国联行往来的基本做法是: 由发报行, 收报行分别核算, 中国人民银行计算中心集中逐笔核算监督、收报行逐笔对帐; 管辖分行督促检查, 最后由总行汇总轧平。

根据以上做法和计算机处理的特点, 以及人工处理的惯例, 确定了对帐要素。即以科目号, 收行号, 发行号, 金额, 报单号为核对项, 发报日期, 报单类别为参考项。又确定了本系统采用的对帐方法。即以对帐表为单位进行初核和复核。输出对帐报告表(分已核, 未核两部分)做为第一阶段。对第一阶段未核报单定期核对, 输出未核销文件已核对报单清单和已冲正清单。为第二阶段。显然第二阶段对帐要反复进行, 直到未核文件报单全部核对为止。

## 二、工作环境

则, 是编辑程序段的依据, 采用产生式形式表示, 左部给出各程序段之间应满足的条件, 右部给出应该怎样进行组织, 是属于控制流类。数据流类还是子过程类等等。

(3) 样本程序知识: 采用PASCAL子程序的形式表示。在子程序中设置一些参数, 这种子程序只能称为模式, 因为它表达了解决某一类相似处理的共同处理步骤, 其中的参数由具体问题才能确定。

### 2. 目标程序的生成

软件生成系统负责把编程要求的形式描述转换成相应的PASCAL程序。由于形式描述与编程知识表示相类似, 所以转换工作比较简单, 采用匹配推理方式实现, 详见文献[4]。

## 六、结束语

为证明系统的可行性, 我们在IBMPC/AT机上编了一个实验程序, 采用PASCAL语言写成。经运行效果良好, 说明系统的设

计思想是合理的。但系统还有待进一步扩充与完善, 如应用领域知识的不完备性和不确定性, 描述语言的抽象程度的不明确等等。总之, 目前AP的研究非常广泛, 我们所做的工作只是一个尝试。

### 参考文献

- 1 D.Barstow, "Domain-Specific automatic programming"  
IEEE trans, Software eng, vol.SE-11, No.11, 1985.
- 2 U.Bartels, w.Olthoff, P.Raulefs, "AP E, An expert System for automatic programming from abstract specification of data types and algorithms." proc.Tth IJCAI, 1980, 1037-1043.
- 3 G.Guida, M.Guida, S.Gusmeroli, M.Somavico, "Design and experimentation of an expert system for programming in-the-large." Proc.AAAI-86, 1155-1163.
- 4 朱瑞香, 王学锋, "编程知识的表达及其应用"  
《知识工程》, 1989年第1期。



## 1. 系统硬件配置 (联行系统运行的最小配置)

主机: M—240H

CPU: H—8840 1台

操作系统: VOS1/ES

内存: 不小于1M

外存: 磁盘H—8595 80  
柱面

外部设备:

软盘机: H—8131—1 1台

磁带机: H—8487 2台

行式打印机: H—3276 1台

外围设备:

通讯控制器 H—8666

调制解调器 HM—240 C/D

FBT银行专用终端, 终端控制器,  
工作站

## 2. 软件环境

VOS1—ES操作系统具有实时控制和批处理的能力, 它提供多用户工作环境, 用户个数在系统生成时指定。

VOS1—ES操作系统提供了完善的文件管理功能。用户可根据自己的需要组织SAM, DAM, ISAM, VSAM四种类型文件。

VOS1—ES操作系统支持多种语言。系统提供作业控制语言做为系统与操作人员的接口, 联行系统选择COBOL语言为编程语言。

## 三、设计目标

全国联行系统最多可同时处理80个分理处的全国联行对帐业务。设计能力为, 年处理对帐表和来帐报单各360万笔。采用批处理方式, 每批可输入来帐报单15000笔, 对帐表报单35000笔。一般每日一批, 也可多批。系统将对存于各文件中的来帐报单和对帐表报单(最多各22万笔)定期进行各种核对, 并输出相应报表。

来帐报单有两种传送方式: 联机传送是分理处通过联机终端, 由联机处理系统分离

后, 将数据直接输入, 存放在专用的SAM文件中。另一种为软盘传送, 初核后来帐报单修改信息也用软盘传送。

对帐表由总行送来, 传送方式有三种: 联机传送。它是通过通讯线路传送数据, 将数据存入专用的SAM文件中。第二种方式是磁带传送。如果是打印的对帐表, 需先手工录入软盘, 然后再送入系统。

## 四、信息流程

该系统为典型的事务性处理系统, 数据量大, 处理周期长, 环节较多。因此设计合理的数据结构和流程是该系统的基础。我们按照联行往来的业务情况将一个业务处理周期分解成几个部分, 每一部分相对独立, 使原来复杂的问题更加简化。再根据这部分的需求选择文件类型和组织方式。本系统使用了三种文件类型SAM文件(顺序), VSAM文件(随机), DAM文件(相对)。例如: 来帐处理分为输入信息检查, 缓存, 对帐。输入信息检查逐笔进行, 选用SAM文件。缓存是批处理选用DAM文件, 来帐对帐变动性大, 选用VSAM文件, 然后制定出合理的数据结构和流程, 每部分的流程合并产生系统的信息流程, 系统信息流程从数据的角度清晰地描述了系统的全貌。为便于系统维护, 对每个文件名, 数据名都作了统一规定。这使得系统资料更加规范化。

信息主要分为三大部分。来帐信息由分理处提供, 经过输入检查, 分离、缓存, 送入来帐对帐文件。对帐表信息由总行计算中心提供, 经过输入检查送入对帐缓存文件。这两部分属于原始信息。对帐缓存文件与来帐对帐文件进行核对, 其中未对上部分转入未核销文件(第一阶段), 未核销文件与来帐对帐文件进行核对(第二阶段), 凡已核对报单在第二阶段即可删除。对帐后产生的有关信息, 属于第三部分(各种报表, 帐目等)。信息流程图如图1所示。

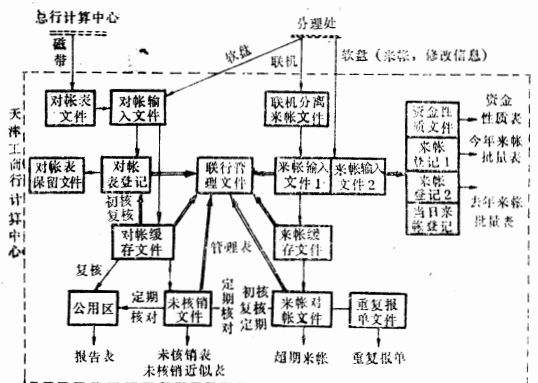


图1 信息流程图

## 五、系统总体结构

全国联行系统模块共分四层，第一层为命令解释和作业流控制层，是操作人员与联行系统的接口，用作业控制语句写成。第二层为控制层，控制处理复杂的操作。如：来帐处理，对帐处理等。第三层为功能层，每个模块具有独立的功能，第四层为公共存取模块。各层自上而下调用。第三层中除复杂的操作由控制层控制外，部分模块可跳过控制层由第一层直接调用。本层模块之间不发生联系，二，三，四层用COBOL语言编写。

第二层控制层在控制处理复杂操作的同时,还负担着管理信息的作用,经过不同阶段处理后的信息反馈到控制层,记录在有关文件中。通过这些文件可及时了解对帐表,来帐报单的输入情况,以及初核,复核情况。及时掌握未对上的对帐表,来帐报单。向用户提供必要的信息,找出原因进行修改,准备再次对帐。对联行系统的处理过程起着监督作用。因此也相当于联行系统线管理层。

系统结构如图 2 所示。

## 六、系统的特点

除对帐算法外,管理程序的设计也是本

系统的关键。它包括管理表,科目帐,来帐汇总表,对帐汇总表等。从管理表中可看出不同阶段的处理结果,来帐和对帐表的输入情况,缓存未核对的来帐情况,对帐情况。有多少对帐表已核对,有多少未核对。还可提供查错信息。四个科目帐相互制约,可监督对帐工作进行顺利。从来帐汇总表中可看出每个分理处每批来帐的输入情况,删改情况,供分理处查对。从对帐汇总表中,可看出每批对帐表的输入情况。

为确保系统正常运行，对输入数据要进行各种检查，及时处理把故障隐患排除在系统之外，对一些数据输入的~~错误~~也提供了修改手段，一旦意外情况发生系统具有保护措施，具有拷贝，恢复功能，故障排除后，不丢失信息，能较快的恢复正常运行。

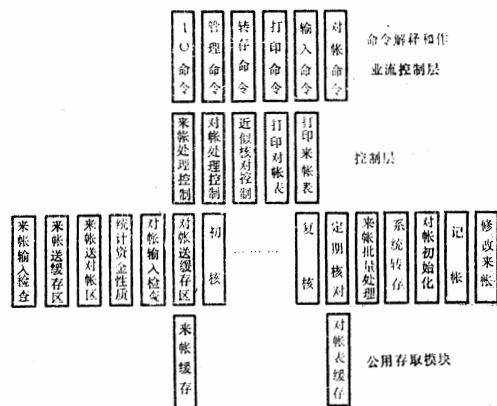


图2 系统结构图

## 七、运行情况

两年多运行情况表明,该系统稳定可靠,操作方便,具有良好的适应性,高峰时五日处理六万笔帐。不仅代替了大量繁杂的手工对帐工作,还有效的防止了原来人工处理全国联行往来中出现的重复来帐现象。这两年中,每年都查出百万元的重复报单。对保证资金的安全起了重要作用,其经济和社会效益都很明显。

# 一个实用的词组库编译程序

中国石油化工总公司镇海石油化工总厂计算机站

郑国安

**摘要** 本文介绍一个实用的词组库编译程序ZBY，这个编译程序能够编译得到达64KB的目标词组库，每个词组的汉字数可达125个。

## 一、前言

汉字的输入一直是汉字信息处理中的一个“瓶颈”。使用词组输入法，是提高汉字输入速度的一种有效方法。这种输入方法，特别适用于一般的业务人员。CC-DOS向用户提供了具有较强功能的词组输入方法，普遍受到用户欢迎。

但是，使用CC-DOS向用户提供的词组库建库工具CZ.EXE来建立词组库是一件既费时又费力的工作，操作步骤复杂，不直观，不能得到词组码本，给用户带来很多不便，影响了词组输入方法的进一步推广使用。

用编译的方法来生成CC-DOS的目标词组库，是一个很好的想法。我们先用通用的字处理程序（如WORDSTAR）或者行编辑程序（如EDLIN），按一定的语法要求编辑建立一个源词组库文本文件。然后使用词组编译程序对其加工，生成符合CC-DOS系统格式的目标词组库。

用这种编译方法来建立、生成目标词组库，有许多的优点，如：

a. 操作方便，直观明了。

b. 因为利用了通用的文本编辑程序来建立源词组库，因此可以使用这些文本编辑程序的各种命令对源词组库进行增加、删除、修改、检索、查阅、并库、分库等各种操作。而且很容易得到词组库码本的硬拷贝。

c. 可以很容易进行各种词组库之间的成批转换。譬如，可以把五笔字形词汇库先通过一个简单的转换程序，转换为文本文件（源词组库），然后再通过词组库编译程序生成可

以在“拼音”状态下使用的目标词组库。笔者就曾经用这种方法把一个含有6000多个常用词组的五笔字形词汇库，转换成了拼音词组库。

本文介绍一个实用的词组库编译程序Z-BY（程序清单附后）。

## 二、建立源词组库

源词组库的语法规则如下：

〈源词组库〉 ::= 〈行〉 { 〈行〉 }

〈行〉 ::= 〈项〉 { , 〈项〉 } 〈回车〉

〈项〉 ::= 〈输入码〉 = 〈词组〉 { = 词组 } }

〈输入码〉 ::= 1个字母/2个字母/3个字母

〈字母〉 ::= A . . Z/a . . z

〈词组〉 ::= 〈汉字〉 { 〈汉字〉 }

譬如，下面就是符合以上语法规则的一个源词组库：

ZH=镇海=总和=组合=最后

HG=化工=合格=回顾=后果

JSJ=计算机，DYJ=打印机

CZ=操纵=操纵=操作系统

等等，指出一点：源词组库文件名必须有后缀。

## 三、词组库编译

请首先按附录的程序清单，建立编译程序的源程序ZBY.ASM。检查无误后用宏汇编MASM通过汇编，然后连接（LINK）生成编译程序的执行代码ZBY.EXE。

键入：ZBY〈源词组库文件名〉

开始对源词组库进行编译，编译采用两

遍扫描, 第一遍扫描检查源词组库的语法错误。可能发现的语法错误有以下六种:

1. 文件名错。
2. 输入码超长。
3. 输入码符非法。
4. 词组组成符非法。
5. 词组超长。
6. 词组库太大。

如发现语法错误, 就不进入第二遍扫描。应按屏幕上的错误提示, 改正以后重新进行编译。

第一遍扫描通过后进入第二遍扫描, 生成目标词组库, 目标词组库的文件名是源词组库的文件名去掉后缀, 由编译程序自动给出。如源词组库的文件名是ZZG.DAT, 则目标词组库文件名就是ZZG。

#### 四、目标词组库装入CC-DOS

编译通过后, 就可以用CC-DOS的CZ-LOAD命令把目标词组库装入内存使用了。我们曾经使用ZBY.EXE编译得到若干个通用词组库和专用词组库, 效果非常好。最大的一个词组库有57KB。

程序清单

```
; NAME ZBY.ASM
; 词组库编译程序
; 编制日期:1989年8月
SSEG SEGMENT PARA STACK 'STACK'
    DW 64 DUP (?)
SSEG ENDS
CSEG SEGMENT
    ASSUME CS:CSEG,DS:CSEG,ES:
    DSEG, SS:SSEG
    START PROC FAR
    PUSH DS
    XOR AX, AX
    PUSH AX
    PUSH DS
    MOV SI, 0080H
    LODSB
    MOV CL, AL
    XOR CH, CH
```

L0: LODSB

```
    CMP AL, 20H
    JA L1
    LOOP L0
LL:  PUSH CS
    POP DS
    MOV DX, OFFSET ERI
    CALL DISPO
    POP DS
    RET
L1:  DEC SI
    MOV CS:SSI, SI
    PUSH CX
    PUSH SI
    INC SI
L2:  LODSB
    CMP AL, " ."
    JZ L3
    LOOP L2
    POP SI
    POP CX
    JMP LL
L3:  DEC SI
    MOV CS:ESI, SI
    POP SI
    POP BX
    MOV BYTE PTR[SI+BX], 00
    MOV DX, SI
    MOV AX, 3D00H
    INT 21H
    JB LL
    PUSH CS
    POP DS
    MOV FILE, AX
    MOV AX, 0006H
    INT 10H
    MOV CL, 0CH
    CALL DISPI
    MOV DX, OFFSET BEGIN0
    CALL DISPO
    MOV CL, 0CH
    CALL DISPI
    MOV DX, OFFSET BEGIN1
    CALL DISPO
    MOV CL, 0CH
    CALL DISPI
    MOV DX, OFFSET BEGIN2
```

	CALL DISPO		JZ L13
	MOV CL, 00		CMP AL, 0DH
	CALL DISPI		JZ L13
L4:	MOV XA, 01		MOV DX, OFFSET ER4
L5:	MOV JIA, 01		CALL DISPERR
	XOR BL, BL		JMP L14
L6:	CALL READ	L13:	TEST BL, 01
	OR AL, AL		JZ L15
	JNZ LP		MOV DX, OFFSET ER4
LT:	JMP LS		CALL DISPERR
LP:	MOV AL, REF	L15:	PUSH AX
	CMP AL, 1AH		MOV AX, CZD
	JZ L6		INC AX
	CMP AL, 0AH		MOV CZD, AX
	JZ L6		SHL AX, 1
	CMP AL, "="		SHL AX, 1
	JZ L8		XOR BH, BH
	INC BL		INC AX
	OR AL, 20H		INC AX
	CMP AL, "a"		ADD AX, BX
	JB L7		CLC
	CMP AL, "z"		ADD AX, ZJD
	JA L7		POP AX
	JMP L6		JC L16
L7:	MOV DX, OFFSET ER3		ADD ZJD, BX
	CALL DISPERR		CMP AL, 0DH
	JMP L6		JNZ L17
L8:	CMP BL, 03H		MOV AX, LN
	JNA L10		CALL AXINC
	MOV DX, OFFSET ER2		MOV LN, AX
	CALL DISPERR		JMP L4
L10:	XOR BL, BL	L17:	CMP AL, " , "
L11:	CALL READ		JNZ L18
	MOV AL, REF		MOV AL, XA
	TEST AL, 80H		CALL AINC
	JZ L12		MOV XA, AL
L14:	INC BL		JMP L5
	CMP BL, 0FH	L18:	MOV AL, JIA
	JNZ L11		CALL AINC
	MOV DX, OFFSET ER5		MOV JIA, AL
	CALL DISPERR		JMP L10
	XOR BH, BH	L16:	MOV DX, OFFSET ER6
	ADD ZJD, BX		CALL DISPERR
	JMP L10	LS:	MOV CL, 0CH
L12:	CMP AL, "="		CALL DISPI
	JZ L13		MOV AL, BZ
	CMP AL, " , "		TEST AL, 0FFH



```

JZ S0
MOV DX, OFFSET RAS2
CALL DISP0
MOV BX, FILE
MOV AH, 3EH
INT 21H
POP DS
RET
S0:  MOV DX, OFFSET RAS1
      CALL DISP0
      XOR CX, CX
      MOV DX, CX
      MOV AX, 4200H
      MOV BX, FILE
      INT 21H
      MOV AX, DSEG
      MOV ES, AX
      MOV AX, CZD
      XOR DI, DI
      STOSW
      MOV SI, DI
      SHL AX, 1
      SHL AX, 1
      INC AX
      INC AX
      MOV ZIP, AX
S1:  MOV SI, OFFSET SR
      MOV AX, 5B5BH
      MOV[SI], AX
      MOV[SI+2], AL
S2:  CALL READ
      OR AL, AL
      JZ E0
      MOV AL, REF
      CMP AL, 1AH
      JZ S2
      CMP AL, 0AH
      JZ S2
      CMP AL, "="
      JZ S3
      OR AL, 20H
      MOV[SI], AL
      INC SI
      JMP S2
S3:  MOV DI, SIP
      MOV SI, OFFSET SRM
      MOVSW

```

```

MOVSB
XOR BL, BL
MOV SI, DI
MOV DI, ZIP
S4:  CALL READ
      MOV AL, REF
      TEST AL, 80H
      JZ S5
      INC BL
      STOSB
      JMP S4
S5:  MOV ZIP, DI
      MOV ES, [SI], BL
      INC SI
      MOV SIP, SI
      CMP AL, "="
      JZ S3
      JMP S1
E0:  MOV BX, FILE
      MOV AH, 3EH
      INT 21H
      MOV SI, ES1
      MOV DX, SSI
      POP DS
      XOR AX, AX
      MOV[SI], AX
      MOV[SI+2], AX
      MOV AH, 3CH
      MOV CX, 0020H
      INT 21H
      PUSH AX
      MOV CX, CS, ZIP
      PUSH ES
      POP DS
      XOR DX, DX
      MOV AH, 40H
      POP BX
      PUSH BX
      INT 21H
      MOV AH, 3EH
      POP BX
      INT 21H
      PUSH CS
      POP DS
      MOV CL, 0CH
      CALL DISP1
      CALL DISP1

```

```

MOV DX, OFFSET GOOD
CALL DISP0
RET
READ PROC
PUSH BX
MOV BX, FILE
MOV CX, 0001H
MOV DX, OFFSET REF
MOV AH, 3FH
INT 21H
POP BX
RET
READ ENDP
DISP0 PROC
MOV AH, 09H
INT 21H
RET
DISP0 ENDP
DISP1 PROC
MOV DL, 00
MOV AL, CTRL
INC AL
CMP AL, 0AH
JB DISP10
MOV AL, 04
PUSH AX
PUSH CX
MOV DH, 24
MOV AH, 02
INT 10H
MOV DX, OFFSET CONT
CALL DISP0
MOV AH, 07
INT 21H
XOR BX, BX
MOV CX, 0030H
MOV AX, 0A20H
MOV DX, 1800H
INT 10H
MOV DL, 0
POP CX
POP AX
DISP10: MOV CTRL, AL
MOV DH, AL
MOV AH, 02
INT 10H
XOR BX, BX

```

```

PUSH CX
MOV CX, 0050H
MOV AX, 0A20H
INT 10H
POP CX
MOV DL, CL
MOV AH, 02H
INT 10H
RET
DISP1 ENDP
DISP2 PROC
PUSH SI
MOV CL, 04
MOV AH, AL
AND AH, 0FH
SHR AL, CL
AND AL, 0FH
ADD AH, 30H
ADD AL, 30H
POP SI
RET
DISP2 ENDP
DISPERR PROC
PUSH BX
PUSH AX
MOV BZ, 0FFH
PUSH DX
MOV CL, 0AH
CALL DISP1
MOV SI, OFFSET DIS
MOV AX, LN
PUSH AX
MOV AL, AH
CALL DISP2
MOV[SI], AL
MOV[SI+1], AH
POP AX
CALL DISP2
MOV[SI+2], AL
MOV[SI+3], AH
MOV AL, XA
CALL DISP2
MOV[SI+6], AL
MOV[SI+7], AH
MOV AL, JIA
CALL DISP2
MOV[SI+10], AL

```

```

MOV[SI+11], AH
MOV DX, SI
CALL DISP0
POP DX
CALL DISP0
POP AX
POP BX
RET
DISPERR ENDP
AINC PROC
    ADD AL, 01
    DAA
    RET
AINC ENDP
AXINC PROC
    CLC
    CALL AINC
    XCHG AL, AH
    ADC AL, 0
    DAA
    XCHG AH, AL
    RET
AXINC ENDP
;
BZ DB 0
SSI DW 0
ESI DW 0
LN DW 1
XA DB 1
JIA DB 1
DIS DB"行项节:", 24H
CONT DB"[看清后打任意键]" , 24H
ER1 DB"文件错或源库名非法!" , 0DH, 0AH, 24H
ER2 DB"输入码 超长 !" , 0DH, 0AH, 24H

```

```

ER3 DB"输入码符非法 !" , 0DH, 0AH, 24H
ER4 DB"词组组成符非法 !" , 0DH, 0AH, 24H
ER5 DB"词组超长 !" , 0DH, 0AH, 24H
ER6 DB"词组库太大 !" , 0DH, 0AH, 24H
RAS1 DB"第一遍扫描通过,进入第二遍扫描 !" , 0DH, 0AH, 24H
RAS2 DB"第一遍扫描有错!不进入第二遍扫描,请改正错误 !" , 0DH, 0DA, 24H
GOOD DB"词组库编译完成!再见 !" , 0DH, 0AH, 24H
BEGIN0 DB 56 DUP ( " " ) , 0DH, 0AH, 24H
BEGIN2 DB 18 DUP ( " " ) " , <1988年8月> " , 18 DUP ( " " ) , 0DH, 0AH, 24H
BEGIN1 DB"※石化总公司镇海石油化工总厂词组库编译软件V1.00※" , 0DH, 0AH, 24H
CZD DW 0
ZJD DW 0
SRM BD 3 DUP ( 05BH )
SP DW 0
ZIP DW 0
FILE DW ?
REF DB ?
CTRL DB ?
;
START ENDP
CSEG ENDS
DSEG SEGMENT
;
DSEG ENDS
END START

```

# 论MICRO VAX II 的编程方法

杭州链条研究所 吴锦前

## 一、EDT编辑方式和小键盘编辑方式

首先需要说明的是在以下的叙述中，都是就FORTRAN语言和Rdb数据库而言。如程序中有同Rdb数据库通讯的语句，则该文件名后缀必须为RFO，反之则其后缀为FOR。

### 1. EDT编辑方式

EDT是交互式的文本编辑程序，使用EDT即可以产生一个新文件，也可以编辑已有文件中的正文。

要引用EDT，需打入DCL命令ED，并

指出要编辑的文件名。例如：

\$ED 文件名.后缀

[EOB]

屏幕出现[EOB]，等待你输入新程序。

如该文件名原来就存在，则把该文件调到屏幕上，等待你的修改。当程序编辑完毕，可按CTRL+Z二键，使屏幕退至\*，即EDT状态。这时如需存盘，则键入EX，不存盘，则键入QUIT。

在使用EDT编辑程序中，有几条命令能给编程带来诸多方便，有必要一述。

- |                      |                                     |
|----------------------|-------------------------------------|
| • 初始行: 结束行           | 显示一段程序                              |
| • WR WJ.FOR 1:10     | 把1-10行写到本目录的WJ.FOR上                 |
| • D 初始行: 结束行/QUERY   | 删一段程序, 并提示每行验证                      |
| • T 40 THRU END      | 显示从第40行到结尾这段程序                      |
| • T WHOLE            | 显示从头至尾的当前缓冲区程序                      |
| • T 40; END ALL' IF' | 显示从40行至尾部所有IF所在行                    |
| • SH TERM            | 显示终端特性                              |
| • SH SCRE            | 显示屏幕允许输入宽度, 初始80                    |
| • SET SCRE 132       | 将显示长度设置为132                         |
| • REPL 3; 杭州         | 将第3行语句替代为‘杭州’                       |
| • M 1:8 TO 100       | 将1-8行移到第100行之前                      |
| • IN 25              | 在25行前插入                             |
| • IN XZ.FOR          | 在当前行前插入XZ.FOR程序                     |
| • C END              | 从EDT编辑方式改变为小键盘编辑方式, 将程序调到屏幕, 光标位于末尾 |
| • C                  | 进入小键盘编辑方式, 将程序调到屏幕, 光标在原初始行         |
| • COPY 1:2 TO 4:5    | 把1-2行复制到第4-5行上, 原4-5行后移, 原1-2行仍存在   |

## 2. 小键盘编辑方式

小键盘上部有4个键,分别从左至右命名为Pf1, Pf2, Pf3, Pf4。这几个键与数字键相互组合就能弥补 EDT编辑方式中的某些不足,在编辑中应灵活地加以引用。

小键盘 0键

键入Pf1+0, 插入一空行。

小键盘 ①键

键入①, 光标移到下一个WORD字的开头。

键入Pf1和①, 位于光标处的字符改变大小

写。

键入④, 再键入①, 光标下移一个字。

键入⑥, 再键入①, 光标上移一个字。

小键盘②键

键入②, 光标移动到当前行或前一行的末端。

端。

键入④, 再键入②, 光标下移到一行末端。

键入⑥, 再键入②光标上移到一行末端。

小键盘 ③键

键入③, 光标按原设置的方向移动一个字

符。

键入④, 再键入③, 光标右移一个字符。

键入⑥, 再入键③, 光标左移一个字符。

小键盘 ④键

键入④, 光标置成向右和向下方向。

键入Pf1, 再键入④, 光标移到程序底部。

小键盘 ⑤键

键入⑤, 光标置成向左和向下方向。

键入Pf1, 再键入⑤, 光标移到程序顶部。

小键盘 ⑥键

键入⊙, 再用↑↓←→键覆盖某段程序,

(取消覆盖, 用Pf1+⊙), 尔后用⑥键移走

所覆盖的序程到内存, 再把光标停在需插入处, 键入Pf1+⑥插入该程序。

小键盘 ⑦键

键入④, 光标置向下方向。再键入⑦, 光标移到程序底部。

键入⑥, 光标置向上方向, 再键入⑦, 光标移到程序顶部。

小键盘 ⑧键

键入④, 再键入⑧, 光标向下跑16行。

键入⑥, 再键入⑧, 光标向上跑16行。

小键盘 Pf2键

键入Pf2, 提示帮助。

小键盘 Pf3键

键入Pf1+Pf3, 寻找程序中某值, 当回答

所寻找值后, 以设置④键作为向下寻找, 设置⑥键作为向上寻找, 用‘ENTER’键作为回车键, 当需寻找下一个相同值时, 键入Pf3。

小键盘 Pf4键

键入Pf4, 删除光标所在行。

键入Pf1+Pf4, 在光标处插入一空行。

小键盘 ‘-’ 键

键入‘-’, 删除光标所在处一个WORD字。

键入Pf1+‘-’, 恢复已删除的一个WORD字。

## 二、编译、连接、运行

当一个程序编制完成后, 就得对其进行编译, 生成OBJ文件。对于后缀为FOR的程序, 编译语句为:

\$ FOR 文件名

对于后缀为RFO的程序, 编译语句为:

\$ RUN SYS \$SYSTEM.RDB  
PRE

INPUT FILE>1 FOR 文件名

编译完成后, 把OBJ文件连接成可执行文件, 其后缀为EXE。连接语句为:

\$ LINK 主文件名, 子文件名  
1, 子文件名2...

连接完成后, 就可执行文件, 执行语句为:

\$ RUN 主文件名

## 三、编程技巧

当你所编制的程序有与RDB数据库通讯时, 后缀应为RFO, 但这些都与数据库通讯的语句在程序中是如何反映呢? 在此有必要说明一下。当程序中按FORTRAN的书写格式其语句标号位出现 &RDB &, 这一行的语句即为RDB语句, 反之就为FORTRAN语句。现举一些实例加以说明。

### 1. 屏幕定位输入

```

S=LIB $ERASE-PAÇE (1, 1) [清屏
语句]
S=LIB$PUT_SCREEN (' 请 键 入 打
印 日,期,' , 10, 20)
31 S=LIB$PUT_SCREEN (' [月 日] ' ,
10, 42)
S=LIB$SET_CURSOR (10, 44)
READ ( *, 15, ERR=31) KS, NC
IF (NC.NE.0) THEN
NC=0
GOTO 31
END IF
IF ( ( KS.GT.12) .OR. (KS.LT.0) )
THEN
GOTO 31 [控制月份输入在1-12之间]
END IF
32 S=LIB$PUT_SCREEN (' 日' , 10,
48)
S=LIB$SET_CURSOR (10, 49)
READ ( *, 15, ERR=32) JS, NC
IF (NC.NE.0) THEN
NC=0
GOTO 32
END IF
IF ( ( JS.GT.31) .OR. (JS.LT.0) )
THEN
GOTO 32 [控制日期输入在1-31之间]
END IF
15 FORMAT (I2, Q)

```

### 2. 寻找某条记录

```

JNUB=1
&RDB & FOR K IN KJ WITH K.JNUB
=JNUB
&RDB & GET
&RDB & W=K.J1 [取出某个品种名到变量
W]
&RDB & END_GET
&RDB & FOR G IN GU WITH
&RDB & W EQ G.GU0 [在GU关系 中寻
找与W相同的记录]

```

```

&RDB & GET
&RDB & B1=G.GU2,
&RDB & B2=G.GU3
&RDB & END_GET
.
.
.

```

对取出的变量B1, B2进行处理

```

.
.
.
CONTINUE [再找有否满足
条件的记录]
&RDB & END_FOR[GU关系记录流封]
JNUB=JNUB+1
&RDB & END_FOR [KJ关系记录流封]
.

```

### 3. 取出最大记录号

```

&RDB & INVOKE DATABASE FILENA-
ME' HZLT1'
&RDB & START-TRANSACTION READ
-WRITE RESERVING
&RDB & KJ FOR SHARED READ
&RDB & GET MJ=COUNT OF K IN KJ
&RDB & END_GET
注:把KJ关系中最大记录号
送到变量MJ

```

### 4. 循环结束

```

JNUB=1
P1=' A'
DO WHILE (P1.NE.' N' )
&RDB & FOR K IN KJ WITH K.JNUB
=JNUB
&RDB & GET
.
.
.
取 数
.
&RDB & END_GET
&RDB & END_FOR [KJ关系记录流封]
&RDB & FOR G IN GU WITH
&RDB & W EQ G.GU0
.

```



## 运 算

```
&RDB&    END-FOR [GU关系记录流封]
          JNUB=JNUB+1
          IF (JNUB.GT.MJ) THEN
            P1=' N' [MJ为KJ关系中的最大
                      记录号]
          END IF
        END DO
```

## 5. 清除数据库

```
&RDB&    INVOKE DATABASE FILE-
          NAME' HZLT1'
&RDB&    START_TRANSACTION RE-
          AD_WRITE.RESERVING
&RDB&    F2 FOR SHARED WRITE
&RDB&    FOR F IN F2 ERASE F END
          _FOR
```

## 6. 修改一条记录

```
&RDB&    FOR D IN HZ1
&RDB&    WITH D.M1=' 08B-1'
&RDB&    MODIFY D USING
&RDB&    D.M2=L2;
&RDB&    D.M3=L3;
&RDB&    D.M4=L4;
&RDB&    D.M5=L5;
&RDB&    D.M6=L6;
&RDB&    END_MODIFY
&RDB&    END_FOR
```

注,D.M\*的为字段名,  
L\*的为变量名。

## 7. 追加一条记录

```
&RDB&    STORE D IN HZ1 USING
&RDB&    D.M2=L2;
&RDB&    D.M3=L3;
&RDB&    D.M4=L4;
&RDB&    D.M5=L5;
&RDB&    D.M6=L6*L7;
&RDB&    END_STORE
```

## 8. 屏幕锁定

```
PP2=' ' [定义变量]
DO WHILE ( (PP2.NE.' 1' ) .AND.
(PP2.NE.' 2' ) )
  PP2=' ' [清空语句]
```

S=LIB\$PUT\_SCREEN(' 1.输入合同  
2.修改合同', 23,6)

S=LIB\$PUT\_SCREEN(' ', 23,  
34, 2)

S=LIB\$SET\_CURSOR(23, 34)

READ(\*, 15) PP2

15 FORMAT(A1)

END DO

IF (PP2.EQ.' 1' ) THEN

CALL UR [转输入合同程序]

END IF

IF (PP2.EQ.' 2' ) THEN

CALL XG [转修改合同程序]

END IF

注,该程序不送1, 2情况下,永远不会继续运  
行下去.如免去屏幕1, 2显示,可作为一个  
密码口令。

## 9. 记录排序

```
&RDB&    FOR W IN WA
&RDB&    SORTED BY W.M1 WITH W.
          M18=' 机 械 维 修'
&RDB&    REDUCED TO W.M1
&RDB&    GET
&RDB&    H1=W.M1;
&RDB&    H2=W.M2;
&RDB&    H3=W.M3;
&RDB&    H4=W.M4;
&RDB&    H5=W.M5;
&RDB&    H6=W.M6;
&RDB&    H7=W.M7
&RDB&    END_GET
```

注,按字段M1进行排序,目前缺省  
(ASCENDING)即按升序  
排,如需降序,应设定为:  
(SORTED BY DESCEND-  
ING W.M1...)为使具有相同  
号码的M1字段避免重复,用R-  
EDUCED子句约束.被排序后  
的记录送到变量H\*,以H\*作  
为新的记录输出。

## 10. 菜单程序

C 这是一个主菜单程序

CHARACTER\*70 FOR1,FOR2,FOR3,

```

FOR4, FOR5, FOE6
CHARACTER FORM, OO
DATA FORM/' ' /
IMPLICIT INTEGER*4(A-Z)
DATA FOR1/' 销 售 科 管 理 系 统' /
DATA FOR2/' =====' /
DATA FOR3/' 1.--合同管理    2.--销
售管理' /
DATA FOR4/' 3.--计划管理    4.--仓
库管理' /
DATA FOR5/' 0.--退出系统' /
DATA FOR6/' 请 选 择[0-4]' /
10 S=LIB$ERASE-PAGE(1, 1)
S=LIB$PUT-SCREEN(FOR1, 5, 28)
S=LIB$PUT-SCREEN(FOR2, 7, 14)
S=LIB$PUT-SCREEN(FOR3, 9, 20)
S=LIB$PUT-SCREEN(FOR4, 12, 20)
S=LIB$PUT-SCREEN(FOR5, 15, 20)
S=LIB$PUT-SCREEN(FOR2, 17, 14)
S=LIB$PUT-SCREEN(FOR6, 19, 31)
S=LIB$PUT-SCREEN(FORM, 19, 50, 2)
50 S=LIB$ERASE-LINE(19, 50)
S=LIB$SET-CURSOR(19, 50)
READ(5, 100, ERR=50) OO, NCS
IF(NCS.NE.0) THEN
NCS=0
GOTO 50
END IF
IF(OO.GT.' 5' .OR.LT.' 0' ) THEN
GOTO 50
END IF
100 FORMAT(A1, Q)
IF(OO.EQ.' 1' ) THEN
CALL HT
GOTO 10
END IF
IF(OO.EQ.' 2' ) THEN
CALL XU
GOTO 10
END IF
IF(OO.EQ.' 3' ) THEN
CALL JH
GOTO 10

```

```

END IF
IF(OO.EQ.' 4' ) THEN
CALL CH
GOTO 10
END IF
IF(OO.EQ.' 0' ) THEN
S=LIB$ERASE-PAGE(1, 1)
END IF
END

```

### 11. 打印程序

```

C      这是一个RDB FORTRAN 打印子
      程序
      SUBROUTINE 子程序名
      INTEGER U, P, P1
      CHARACTER MS
      CHARACTER*2 T1,T4,T5,T6,
T7,T8,T9, T10, T11, T12, Q1, FORM1
      CHARACTER*4PP,PP1, UU,
UU1
      CHARACTER*6 T33, H17
      CHARACTER*8 T2,ST22,
ST23, ST24,ST2,ST3,ST4, ST6
      CHARACTER*8 ST7, ST8,
ST10, ST11, ST12, H1, H2, H16, H31
      CHARACTER*12 H30
      CHARACTER*14 H15
      CHARACTER*18 T3, H13
      CHARACTER*20 T61,T62,T63
      CHARACTER*25 KK
      CHARACTER*26 H14
      CHARACTER*36 T181, H3,
H12, MM3
      CHARACTER*58 SZ3
      CHARACTER*60 SZ1
      CHARACTER*70 FOR1
      CHARACTER*86 SZ2
      CHARACTER*130 W
      CHARACTER*204 SS, SS1,
SS2, SS3
&RDB &   INVOKE DATABASE FILEN-
AME' HZLT1'
&RDB &   START-TRANSACTION READ
_WRITE RESERVING

```

```

&RDB &    WANG FOR PROTECTED
WRITE
U=0    [库记录计数变量]
P=0    [页记录计数变量]
P1=1    [页数变量]
T1=' ['
T2=' '
T3=' '
T33=' '
T4=' | '
T5=' | '
T6=' |-'
T7=' -|'
T8=' T'
T9=' 上'
T10=' +'
T11=' L'
T12=' J'
T61=T2//T8//T2//T8[线段组合]
T181=T3//T3
SS1=T1//T61//T181//T8//
1 T181//T8//T3//T8//
1 T3//T2//T8//T2//T33//T8//T2
  //T8//
1 T33//T33//T8//T2//T8//T33//
  T5
T62=T2//T10//T2//T10
SS2=T6//T62//T181//T10//
1 T181//T10//T3//T10//
1 T3//T2//T10//T2//T33//T10//
  T2//T10//
1 T33//T33//T10//T2//T10//T33
  //T7
T63=T2//T9//T2//T9
SS3=T11//T63//T181//T9//
1 T181//T9//T3//T9//
1 T3//T2//T9//T2//T33//T9//T2
  //T9//
1 T33//T33//T9//T2//T9//T33//
  T12
S=LIB$CHAR(MS,14) , [设置
表头大写]

```

```

WRITE(1, 250), MS
KK=' '
SS=KK//' 索引表'
WRITE(1, 300), SS
SS=KK//' ====='
WRITE(1, 300), SS
S=LIB$CHAR(MS, 15) [设
置表格小写]
WRITE(1, 250), MS
250 FORMAT(A10)
PP=' '
S=FOR$CVT-D-TF(%REF(P
1), %DESCR(PP), %VAL(0))
S=STR$LEN-EXTR(PP1,PP,
1, 3)
W=' ' [即定义了130个空格]
SS=' 行业,五金经销' //
1 W//' 第' //PP1//' 页'
WRITE(1, 300), SS
WRITE(1, 300), SS1
WRITE(1, 350)
350 FORMAT(' | 卡片号码 | 合同号
      码 |
1 单 位 名 称 |
1 通 讯 地 址 |
1 开 户 银 行 | 账 号 |
1 电 话 | 电 报 | 到 站 |
1 邮政编码 | 代表人 | ')
&RDB &    FOR W IN WANG
&RDB &    SORTED BY W.M1 WITH W
M18=' 五金经销'
&RDB &    REDUCED TO W.M1
&RDB &    GET
&RDB &    H1=W.M1;
&RDB &    H2=W.M2;
&RDB &    H3=W.M3;
&RDB &    H12=W.M12;
&RDB &    H13=W.M13;
&RDB &    H14=W.M14;
&RDB &    H15=W.M15;
&RDB &    H16=W.M16;
&RDB &    H17=W.M17;
&RDB &    H30=W.M30;

```

```

&RDB &   H31=W.M31,
&RDB &   END_GET,
WRITE(1, 300), SS2
SZ1=T4//H1//T4//H2//T4//H3
//T4
SZ2=H12//T4//H13//T4//H14//
T4
SZ3=H15//T4//H16//T4//H30
//T4//H31//T4//H17//T4
SS=SZ1//SZ2//SZ3
WRITE(1, 300), SS
U=U+1    [库记录计数变量加1]
P=P+1    [页记录计数变量加1]
IF(P.EQ.23) THEN
WRITE(1, 300), SS3 [封底线]
WRITE(1, 300)
WRITE(1, 300)
WRITE(1, 300)
WRITE(1, 300) [换页跑空行]
WRITE(1, 300)
WRITE(1, 300)
WRITE(1, 300)
P=-3 [为分页后多打几条记录而
      设置]
P1=P1+1 [报表页数变量]
PP=' '
S=FOR$CVT_D_TF(%REF(P
1), %DESCR(PP), %VAL(0))
S=STR$LEN_EXTR(PP1, PP,
1, 3)
SS=' 行业:五金经销' //
1 W// '第' //PP1// '页'
WRITE(1, 300), SS
WRITE(1, 300), SS1

```

```

400 WRITE(1,400)
FORMAT(' | 卡片号码 | 合同号
      码 |
      | 单位名称 |
      | 通讯地址 |
      | 开户银行 | 账 号 | |
      | 电 话 | 电 报 | 到站 |
      | 邮政编码 | 代表人1' |
END IE [分页结束]
&RDB &   END_FOR [WAN G关系记录流
      封]
WRITE(1, 300), SS3
UU=' '
S=FOR$CVT_D_TF(%REF
(U), %DESCR(UU), %VAL(0))
S=STR$LEN_EXTR(UU1, UU,
1, 3)
SS=' 该行业用户数:' //UU1//
' 个'
WRITE(1, 300), SS
300 FORMAT(A204)
&RDB &   ROLLBACK [取消处理中所进行的
      全部更新]
RETURN [返主程序]
END
注, S=FOR$CVT_D_TF(%REF
(P1), %DESCR(PP), VAL(0))
把数字量P1转换成字符PP, 其中小
数位为0.
S=STR$LEN_EXTR(PP1, PP,
1, 3)
把字符串PP从第1位起取3位, 送到
PP1.
输入报表如下:

```

行业：五金经销 索引表

卡片号码	合同号码	单位名称	通讯地址	开户银行	账号	电话	电报	到站	邮政编码	代表人
0217000	5280	上海工农机械配件商店	上海市北京东路663号	工行黄办北分处	205-51198224	220553	0781	三门路 745号		林以诺
0217001	1083	上海第一五金店	上海市北京东路675号	工行黄办南分处	235-06190629	223679	2349	北郊		张传浩
0217002	5152	上海黄浦区五金交电公司经营部	上海市福建中路517号	黄办南分处	235-06101208	220091	3483	北郊		张仁祥
0217003	1080	上海勤俭五金机械商店	上海市北京东路765号	工行黄办南分处	235-50100056	222754	0264	北郊		张弘强
0217005	1077	上海第十八五金商店	上海市平凉路340号	杨办八埭头分理处	256-06107038	417014		北郊		孙永华
0217048	3063	上海五金机械公司机配分公司	天津市大沽路92号	市工行营业部	244-06122668	206910	0965	北郊		朱友根
0317006	3023	天津市五金交电公司工具批发部	唐山市复兴路32号	工行解放路分理处	107061045980	391730	1031	天津南		刘兵
0417008	1625	唐山链轮链条厂	无锡市人民西路21号	工行复兴路分理处	4004712291	61477	9353	唐山		许庆晖
1617009	1051	无锡五金站工具机配批发公司	体育场路159号	市支行	0610066	228891		无锡		刘骏发
1717011	5208	杭州市五交化公司	北京东路532号	市工行营业部	20106101604	555918		自提		马国林
1717012	1142	杭州立新五金店	杭州市庆春路405号	工行庆办	2020610072	22167				刘志鑫
1717017	1017	宁波建新五金商店	宁波市中山东路43号	市工行东门办	0618202	41153	7910	宁波北		胡荣康

该行业用户数：12个

# 适于并行处理的处理器—T414

天津市电子计算机研究所 陈荣华

**摘要** Transputer芯片是英国INMOS公司近几年开发的新产品,其T414、T800及Occam语言已组成各种应用系统并充分显示出并行处理的优越性。这些芯片从结构上看是功能很强的单片机,有很高的运算速度,极易连网组成阵列机。目前,用于提高IBM PC、NEC PC 9801机计算性能的加速板已问世,为促进这一技术的推广,本文简要介绍T414的结构及基本特点。有关T800的特点,与T414的差别也略加阐述。

## 一、T414的特点

Transputer是INMOS公司开发的微处理器。T414是32位微处理器。具有2K字节的静态RAM、定时器、4组高速串行链(标准传输速度10Mbps)、动态RAM刷新电路,时钟频率为20MHz时达到10MIPS指令执行能力。

就Transputer领域而言,并行处理语言的开发早于处理器。英国C.A.R. Hore发表的Communicating Sequential Processes论文就成了并行处理语言的源型。INMOS公司根据该论文的思想开发了Occam语言,Transputer正是为执行这一语言而设计的。

### 1. 处理器间通信

Transputer与一般微处理器最大的区别就在于其具有构成多处理器系统的串行链,联接串行链可容易且很自由地构成多处理器系统。通过多个处理器联接可大大提高处理能力。在应用开发方面,使用Occam可很容易记述并行处理。

并行工作的处理器间的数据传输必须取得同步。否则,收发数据时执行结果会与期待结果大不相同。同步方法大致可分为二种,多个处理器借助于“共有存储器”而取得同步法,或是通过处理器间直接联接回路传达信息的“通信”同步法。

Transputer采用了通信方式,使用称为串行链的4组双向高速串行回路,完成点到点联接,执行处理器间的信息及数据的传输。

其同步过程为:发送方发出信息。若接收方信息接收完成则返回允许信号。

### 2. 并行处理的记述

记述并行处理的Occam程序是类似于Pascal型式的高级语言。编译过的Occam程序在Transputer上以机器语言命令来执行,且命令小巧。

Occam语言在单处理机上也可记述并行处理,此时通信在存储器上进行。所以,无论是单处理器上执行的程序还是多处理器执行的程序,Occam程序几乎不必变动,仅仅在多处理器情况下稍加一点表示处理器间链的联接状态信息而已。

此外,Fortran、Pascal、c语言的编译器也发出来了,具有用于并行处理记述的函数模块,所以并行处理记述也很简单。

除上述特点外,它还可根据存储器种类、访问时间设定有关的定时;定时设定后刷新也自动进行,存储器接口电路非常简单。

## 二、T414内部结构

T414内部结构如图1所示,与80386,68020等32位微处理器相比结构相当简化,这主要是由于指令系统、寄存器的使用方法不同所致。地址和数据信号线复用,直接寻址存储器空间为4G字节。

### 1. 寄存器

6个寄存器均为32位宽,各寄存器工作如下。

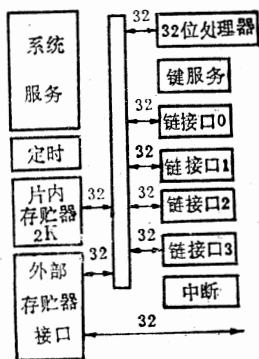


图1

(1) 工作空间指针: 指示局部变量的存贮区域。

(2) 指令指针: 指示下条应执行指令的地址 (或称为程序计数器)。

(3) 操作数寄存器: 用于形成指令操作数。

(4) A、B、C寄存器: 用于形成评价栈的通用寄存器。

A、B、C三个寄存器用于通常算术或逻辑运算的源和目的寄存器。所谓评价栈形成即用A、B、C三个寄存器组成与堆栈相同的结构, 譬如某一数值由存贮器传送到A寄存器时, 首先B寄存器内容传送到C寄存器, 其后A寄存器的内容传送到B寄存器, 最后将存贮器内容传送至A寄存器。

## 2. 并行进程

Transputer 在多处理器情况下可并行执行程序, 即使单处理器也可并行运行几个进程。

由于实时监控器或操作系统的支持, 一般微处理器中进程也可并行工作。而Transputer内藏有微码化的调度程序, 所以可通过将处理器时间分割成时间片的方式并行执行多个进程。

时间片周期是ClockIn输入的5120个周期, 约 $1\mu s$ 的间隔。要求进程切换时间在 $1\mu s$ 以下, 与一般微处理器通过软件切换相比是非常高速了。

(1) 进程的优先级。T414支持2级优先级。进程可分成高优先级和低优先级两种, 每种优先级下可有几种进程。优先级1 (低优先级) 进程只有在优先级0 (高优先级) 无效时可执行。

## (2) 定时

T414有2个32位定时器。其中之一只有高优先级进程可以访问, 以 $\mu s$ 为单位增值, 4295秒为周期反复操作。另一个只有低优先级进程可以访问, 以 $64\mu s$ 为增量, 76小时周期重复。

## 3. T414的输入、输出信号

图2为T414管脚配置, 采用84脚PGA封装。其主要管脚的信号功能如下。

	1	2	3	4	5	6	7	8	9	10
A		Link Special	Proc Clock out	Link123 Special	Link Ino	Link out1	Link In2	Event Ack	GND	Mem Wait
B	Hold To GND	Clock In		Link0 Special	Link out0	Link out2	Link out3	Event Res	Mem Res	not Mem WrB3
C	GND	Vcc	Cap Minus	Cap Plus	Vcc	Link In1	Link In3	Mem Config	Mem Granted	not Mem WrB1
D	Error	Hold To GND	Hold To GND						not Mem Rf	not Mem WrB0
E	Hold To GND	Boot From Rom	ReSet						not Mem Rd	not Mem S0
F	Hold To GND	Anal Se	Mem AD31						not Mem S3	not Mem S2
G	Mem AD30	GND	Mem AD27						Mem not WrD0	GND
H	Mem AD29	Mem AD25	Mem AD23	Vcc	Mem AD16	Mem AD12	Mem AD8	Mem AD4	Mem AD2	Mem not RfD1
J	Mem AD28	Mem AD24	Mem AD22	Mem AD19	Mem AD17	Mem AD13	GND	Mem AD6	Mem AD5	Mem AD2
K	Mem AD26	Mem AD21	Mem AD20	Mem AD18	Mem AD15	Mem AD14	Mem AD11	Mem AD10	Mem AD9	Mem AD7

图2

(1) 电源: 通过Vcc脚和GND脚供给T414, 所有的Vcc、GND脚均应使用。

(2) CapPlus, CapMinus: 连接PLL电路的退耦电容, 供T414产生内部工作时钟用, 一般接 $1\mu f$ 电容。电容有极性时正端接CapPlus端子。

(3) ClockIn: 就Transputer系列而言无论其种类和工作时钟频率如何均将5MHz标准时钟加至ClockIn。时钟的占空比并不重要, 必须注意其稳定性及上升沿 (10ns)



下降沿 (8ns) 的限制范围。

(4) Reset: 这是T414的系统复位, 复位后由内部配置 (用于存储器接口访问定时设定) 进行初始设定。而后执行存储器读周期用于可由用户任意设定的存储器访问定时设定, 即所谓的外部配置。T414系统复位分为Reset单独复位和为了错误分析与Analyse同时进行的复位。

(5) BootFromRom: 决定T414执行来自外部ROM或是链上的引导程序。该信号为“H”时, T414由7FFFFFFF的2字节写入的外部存储器的起始地址开始执行程序。该信号为“L”时, T414等待来自4组串行链中一组链的自举程序, T414存储器的内容可使用链进行读出、写入。

往存储器写入, 送出前导控制字节“ $\phi$ ”, 而后送入8字节数据。其中前4字节是存储器地址, 接下来的4字节为写入存储器的数据。

由存储器读出, 发出控制字节“1”, 而后将4字节的读出存储器地址送至T414, 最后由同一链输出从存储器读出的4字节数据。

控制字节为2~255值时, 解释为写入存储器数据的字节数。后续字节作为命令代码存在MemStart(\*80000048)开始的存储单元, 接收到最后的字节后T414作为低优先级的进程执行由MemStart开始的命令。

(6) Analyse: T414用于错误分析的输入信号, 该信号为“H”时T414保持各种状态信息后停止。而后加载错误分析程序, 进行错误分析。

(7) Error: 表示T414发生错误的输出信号, 为溢出、被0除、超越配置或由软件直接设定标志。

(8) ProcClock Out: 该时钟与T414内部工作时钟同频率, 由ClockIn 5MHz时钟作成。20MHz的T414该时钟周期为50ns。

T414访问存储器的定时标准采用Tm表示, 这一表示状态的Tm值为ProcClock Out周期的一半, 所以20MHz的T414此值为25

ns。

图3为读周期, 图4为写周期, 表1为外部存储器周期。通常T状态设定为1~4Tm, 该长度作为存储器配置在Reset时设置。

(9) MemAD<sub>2~31</sub>: T414外部存储器地址和数据在同一总线上切换实现。使用上位的30位MemAD<sub>2~31</sub>作为地址输出。地址信号在T<sub>1</sub>和T<sub>2</sub>间输出, 所以需要D锁存器锁存地址。

T414没有地址下位的2位信号。以字节为单位存取时读写情况如下: 读周期中Not MemRd输出给所有存储单元, 在T414内部只取入1个字节。写周期中只输出与字节地址相对应的写选通not MemWrB<sub>0~B<sub>3</sub></sub>之一。

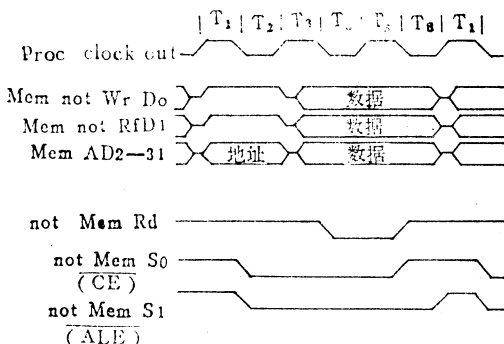


图3

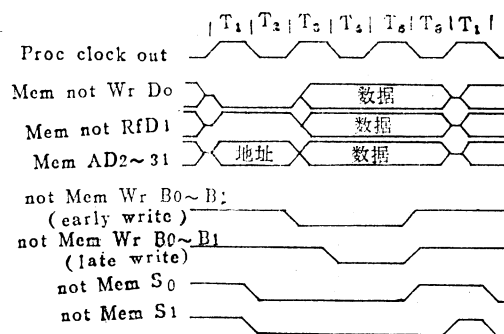


图4

表1 T状态

状态	功能
T <sub>1</sub>	地址有效选通前的地址建立时间
T <sub>2</sub>	地址有效选通后的地址保持时间
T <sub>3</sub>	读周期的3态或写周期的数据建立时间
T <sub>4,1</sub>	可扩充的数据建立时间
T <sub>5</sub>	数据的读或写时间
T <sub>6</sub>	数据保持时间

为实现32位宽数据总线,下两位使用MemnotWrD<sub>0</sub>和Memnot Rf D<sub>1</sub>。在读周期T<sub>3</sub>状态开始后将数据输出到总线上,写周期中由T<sub>3</sub>状态开始至T<sub>6</sub>状态结束输出数据。

(10) Mem not WrD<sub>0</sub>: 在写周期的T<sub>1</sub>和T<sub>2</sub>之间输出“L”,其它情况下为“H”。在T状态的T<sub>3</sub>~T<sub>6</sub>间作为数据总线的0位。

(11) Mem not RfD<sub>1</sub>: 在MemAD<sub>2</sub>~<sub>31</sub>上是刷新地址时,T<sub>1</sub>和T<sub>2</sub>之间其为“L”,其余场合为“H”。在T状态的T<sub>3</sub>~T<sub>6</sub>间作为数据总线的1位。

(12) not Mem Rd: 在读周期的T<sub>4</sub>和T<sub>5</sub>之间为“L”,其余情况下为“H”,数据在其上升沿时读入T414。

(13) not MemS<sub>0</sub>~<sub>4</sub>: 访问存储器的通用选通信号。这5个之中的4个信号可由用户改变定时。

S<sub>0</sub>是固定的选通信号,T<sub>2</sub>开始至T<sub>5</sub>结束期间有效。

S<sub>1</sub>开始与T<sub>2</sub>的开始一致,其结束可按0~31个T<sub>m</sub>的宽度设置。

S<sub>2</sub>~S<sub>4</sub>均在T<sub>5</sub>结束时同时结束。但其开始可比T<sub>2</sub>开始延迟1~31个T<sub>m</sub>宽度。

T414内部存储器访问周期中这些选通信号无效。

(14) not MemWrB<sub>0</sub>~<sub>3</sub>: 写选通信号。一般作为往存储器写入的选通信号,B<sub>0</sub>为地址最下位写选通,B<sub>3</sub>为地址最上位写选通信号。

T414具有高速和低速二种写入周期方式。迟的写周期中B<sub>0</sub>~<sub>3</sub>在T<sub>4</sub>和T<sub>5</sub>之间输出“L”。早的写周期中B<sub>0</sub>~<sub>3</sub>在T<sub>3</sub>输出“L”。数据输出一直到T<sub>6</sub>,在T<sub>5</sub>的上升沿锁存到存储器中。

(15) MemConfig: 用于设定外部存储器接口的访问定时。复位时读入该端子状态。首先读入备有的内部配置之一,而后无论是否找到内部配置就进行外部配置的读入。

一定时间间隔后开始刷新,等待程序的引导,若禁止刷新就不刷新而等待引导。

(16) notMemRf: 选择允许还是禁止动态存储器的刷新。该选择在内部或外部存储器配置周期里决定。

刷新周期定时与标准外部存储器周期相同,在T<sub>1</sub>开始之前插入两个T<sub>m1</sub>状态。所有地址线以相同的定时输出,MemAD<sub>2</sub>~<sub>11</sub>输出刷新地址,MemAD<sub>12</sub>~<sub>30</sub>在地址输出期间为“H”,而MemAD<sub>31</sub>仍为“L”。

选通信号notMemS<sub>0</sub>~<sub>4</sub>在刷新和标准的外部存储器周期中以相同的定时输出,只是notMemRd和notMemWrB<sub>0</sub>~<sub>3</sub>仍为“H”。

(17) MemWait: 用于延长存储器周期,输入为“H”延长T<sub>4</sub>周期。

(18) MemReq, MemGranted: MemReq输入为“H”时请求DMA,刷新及外部存储器周期均在T<sub>6</sub>状态采样MemReq。接受DMA请求后,在ProClockOut的上升沿之后经过2个T<sub>m</sub>状态地址总线呈三态。再经过一个T<sub>m</sub>状态建立MemGranted信号。

在ProClockOut“L”期间采样MemReq的解除,而后在ProClockOut下一个下降沿处解除MemGranted。刷新、外部周期等外部总线的工作在ProClockOut的下一个上升沿处开始。

(19) Event Req, EventAck: 用于外部事件与内部进程间的异步应答。外部事件(中断)与T414内部进程进行通讯时,外部电路使EvenReq为“H”提出通讯要求,T414的通道作与进程通讯的准备。通道和进程准备好后EvenAck为“H”调度进程。

外部电路使EvenReq为“L”时T414置EvenAck为“L”。

#### 4. 链

4组数字串行链用于Transputer之间或处理器与外部器件的同步通讯。每一链由输入通道和输出通道组成。

数据以字节为单位传输,数据与控制信号在同一链上传送。链输出通常为“L”。数据输出如图5所示,先是2个“H”开始位,再传输8个数据位,最后送出一个“L”停止



# 微小型计算机开发与应用

## 一九九〇年总目录

### 第一期

控制系统CAD软件系统的设计  
影子价格经营管理系统  
24×24点阵通用高级汉字打印驱动软件的设计与实现  
扩充C语言的矩阵运算功能  
提高VAX Rdb/VMS数据库存取速度的探讨与实现  
在APPLE II机上利用128K卡建立扩展数组区BASIC解释系统  
小球病毒的分析 and 防治  
关于在微机上将3维图形运用于工程CAD的几个问题的认识和处理  
微机三相全控整流可逆直流传动装置的研制  
单片机微机通信控制器  
开发内藏MMU的68030  
莫托洛拉的MC68040  
AST Premium/386C概述  
VAX-8350计算机的诊断系统  
排除接地干扰的几则实例  
用WIDTH命令定义屏幕显示的行宽

### 第二期

天津市计算机辅助教育决策咨询专家系统  
高考阅卷自动化系统  
软件加密技术的探讨  
C语言实用技巧几例  
DBASE与BASIC程序相互调用画面观动态菜单界面设计  
MCS-51单片机小型显示系统  
单片机在雷达自动录取设备中的应用  
一个油田微机网络管理信息系统的设计  
一种新颖的大规模报警系统  
Intel 80860微处理器的体系结构  
TMS320C25单片16/32位数字信号处理机  
IBM PC/XT系统板维修技术

### 第三期

GTY-10型弹簧实验机微机数据采集处理系统  
电机运行参数的微机监测系统  
单片机在水泥生料配料中的应用  
用TP801实现电视发射机的自动控制  
一种新型数据库—BC数据库  
施工图预算系统的设计·计算表  
VAX/VMS系统的软接口  
一个小型汉字显示程序

管理信息系统开发过程中的快速软件开发工具  
基于Turbo prolog的CAL软件开发技术  
微机在高校学生综合测评工作中的应用  
IBM PC机异步通信适配器(RS-232C)  
基本程序及其使用方法  
Apple机与TP801机之间的数据通讯方法

### 第四期

ADS—自动检测系统  
采用单片机控制的车速里程表  
计算机三维服装显示的逼真性探讨  
简而实用的单片机显示装置  
可编程温控仪的设计分析  
用微机作人机接口工作站  
单片机与测量仪器的接口技术  
基于单片机的局部网通信控制器  
一个快速处理多层嵌套的材料定额统计软件  
如何实现IBM-PC机和MICRO VAX系统的联网  
PC机全屏编辑软件NE的性能与使用  
代码页及其使用  
第二代RISC芯片—Am29000Nonstop II™  
系统结构

UNIX中Bsh程序与Csh程序兼容的方法  
超长超宽报表在屏幕上的平滑移动

### 第五期

冲压用微机控制自动送料系统  
计算机系统仿真及应用实例  
利用Apple-II开发MCS-51系列单片机  
8031单片机在智能微波癌症诊断仪上的应用  
简易微型打印机在单片机中的应用  
在不同的显示卡上建立通用的汉字系统的技术  
新颖统计图形的设计及实现方法  
采用编码技术压缩数据库文件存储结构  
财务记帐帐页的压缩打印技术  
AOS/VS系统记录的使用  
用24针式打印机打印屏幕图形  
IBM PC机除法溢出中断的修改  
VAX11-DMF32并行接口的开发应用  
如何将反汇编出的程序写入文件  
IBM PC/XT故障一例——ROM芯片接触不良  
高速宽行打印机不走纸的检修

### 第六期(见本期目录)