

微小型计算机



开发与应用

MICRO-MINICOMPUTER
DEVELOPMENT & APPLICATION



1990 2

微小型计算机开发与应用编辑部

中华学习机（苹果兼容机） 用户的佳音

为促进计算机事业的发展，本部在有关电脑软件工程师的协助下，特推出供学习机（苹果兼容机）用教育、系统工具，游戏及应用等各类软件几百种，为方便外地用户特办理邮购业务，售价每片13.00元，10片以上每片10.00元，邮费及包装费均免，详细目录来函免费索阅。

另本部还办理各类元器件的邮购业务。

热忱欢迎新老客户惠顾。

杭州华通电子元件经营部

地址：杭州天目山路53—8（杭州大学西侧）

邮政编码：310007

电话：577171

《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
付国明	许镇宇	朱植松
曲庭维	李凤祥	刘连棟
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

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2期目录

(总第46期)

计算机软件

天津市计算机辅助教育决策咨询专家系统

.....乐毓俊 王玉库 孙富元 (1)

高考阅卷自动化系统

.....黄效国 (5)

软件加密技术的探讨

.....陈致明 寇应展 (8)

C语言实用技巧几例.....开效军 (13)

DBASE与BASIC程序相互调用面

观.....陈建华 田燕波 (11)

动态菜单界面设计.....陆慰民 (20)

应用实例

MCS-51单片机小型显示系统

.....王攀峰 (22)

单片机在雷达自动录取设备中的应用

.....孙宇华 (25)

一个油田微机网络管理信息系统的设

计.....吴先旺 汤星辉 (28)

一种新颖的大规模报警系统

.....方沛力 (32)

新技术介绍

Intel 8086C微处理器的体系结构

.....林钟官 (35)

TMS320C25单片16/32位数字信号处

理机.....钱学文 (40)

经验点滴

IBM PC/XT系统板维修技术

.....陈文高 (46)

CONTENTS

SOFTWARE

- A Decision Expert System for Tianjin CAI
-An Expert System Using the Technique of Artificial
Intelligence and the Data Processing Development
.....Le Yujun Wang Yuku
Sun Fuyuan (1)
- The Automatic System of Going Over Examination Papers for
Matriculation Examination.....Huang Xiaoguo (5)
- An Approach to the Technique of Floppy Disk Encryption
.....Chen Zhiming
Kou Yingzhan (8)
- Examples of Active Skill on C.....Kai Xiao Jun (13)
- An Over-All View of the Interchangeability between DBASE III
and BASIC.....Chen Jianhua (14)
Tian Yanbo
- The Design for Dynamic Menu-Interface.....Lu Weimin (20)

APPLICATION EXAMPLE

- MCS-51 Microcontroller Small-Size Display System
.....Wang Panfeng (22)
- The Application of the Single-Chip Processor to the Radar
Automatic Record-Fetch Device.....Sun Yuhua (25)
- The Design and Implementation of an Oilfield Microcomputer
Network Information Management SystemWu Xianwang (28)
Tang Xinghui
- A New Type Alarm System on a Large Scale
.....Fang Peili (32)

NEW TECHNOLOGY

- The Architecture of the Intel 80386 Microcomputer
.....Lin Zhongguan (35)
- TMS 320 C25 13/32-Bit Digital Signal Processor on a Single
ChipQian Xuewen (40)

EXPERIENCE

- The Technique of Maintaining IBM PC/XT System Boards
.....Chen Wengao (46)

天津市计算机辅助教育决策咨询专家系统

—使用人工智能技术与数据处理开发的专家系统

天津师大 乐毓俊 王玉库
天津市计算中心 孙富元

摘要 天津市计算机辅助教育决策咨询专家系统旨在为市局领导提供有关计算机教育和计算机辅助教育的辅助决策和咨询服务。系统设计将人工智能技术和传统的数据处理有机地结合在一起,形成相互独立又密切相关的两个子系统。解决了用户与专家系统,专家系统与数据处理之间的两个接口。该系统使用中文专家系统开发工具CM.1和BASIC语言实现,在IBM PC/XT微机上运行。

一、概 述

天津市计算机辅助教育决策咨询专家系统旨在为市局领导提供有关计算机教育,计算机辅助教育的辅助决策和咨询服务。系统设计将人工智能技术和传统的数据处理有机地结合在一起,形成相互独立又密切相关的两个子系统——决策咨询专家系统(DES)和优化求解系统(PSS)。如图1所示。

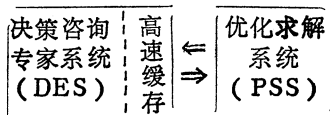


图1 系统结构图

系统根据已获取的专家知识或通过向用户提问所得到的数据进行综合性的逻辑推理与数据处理,为用户提供天津市计算机辅助教育规划方案。考虑到为市局领导提供有关文档材料和面向局级决策,系统按普教、二教、高教和市级主管部门划分层次。

为了实现用户——知识处理——数据处理——用户系统功能模块的自动链接,系统主要解决两个接口:用户与专家系统接口;专家系统与数据处理系统接口。

本系统主要有如下特点:

1. “问答”式中文会话(非自然语言)用户界面。
2. 可随时更新的知识库管理。
3. 以反向推理为主并辅以正向推理的推理机制。

4. 产生数据处理需要的模型系数。

5. 规划与最优化求解。

6. 综合分析与评价。

7. 咨询或多方案决策输出。

系统使用专家系统开发工具CM.1和BASIC语言实现,在IBM-PC/XT微机上运行。

二、决策咨询专家系统(DES)

1. 体系结构

DES的内容包括:本市教育系统开发辅助教育软件的投资规模、软件开发数量及其分配;投资及其效益分析;普教系统计算机硬件配置及预算咨询。系统设计成一个小专家系统,整个系统配置类似于MYCIN式的产生式系统,其主要体系结构如图2所示。

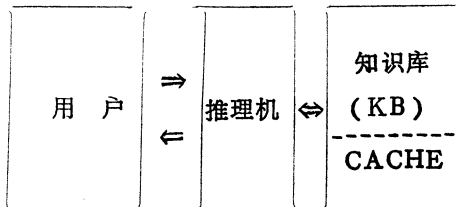


图2 DES体系结构简图

知识库含有计算机教育方面的知识(事实)和用产生式规则表示的有关规则。CACHE用于存放推理过程中产生的中间结果和用户随机装入的事实。同时,它也是DES系统和PSS系统进行数据通讯的重要数据缓冲区。

推理机是按EMYCIN风格设计的。它以反向推理为主,中间穿插用正向推理查询。同时,推理机还可通过自动计算可信度进行非精确推理。

2. 知识库设计

知识表示是知识工程学研究的重要领域,也是建造专家系统知识库的重要环节之一。知识库中的知识是用规则和事实的形式表示的。

〈事实〉 ::= 〈函数式〉 | 〈原子〉

〈规则〉 ::= IF〈前提〉 THEN 〈结论〉

〈前提〉 ::= 〈条件〉 { AND〈条件〉 }

〈条件〉 ::= 〈函数式〉 | 〈原子〉 |

NOT 〈条件〉 | 〈条件〉 { OR 〈条件〉 }

〈结论〉 ::= 〈函数式〉 | 〈原子〉

〈原子〉 ::= 〈逻辑变量〉 | 〈字母符号串〉

〈逻辑变量〉 ::= 〈大写的字母符号串〉

〈函数式〉 ::= 〈原子组成的表达式〉 |

〈原子与常量组成的表达式〉

〈常量〉 ::= 〈整数〉 | 〈实数〉 | 〈表〉

根据专家系统的内容,将知识库设计成按普教、二教、高教的软件咨询决策与投资分配、效益分析等六个子系统。

3. 推理机

针对知识库的设计,推理机采用反向推理作为自身的宏观推理控制策略。反向推理,又称目标驱动的推理方式。它首先根据用户或系统设置的假设逐一验证这些假设的真假性。其算法是:

- (1) 假设是否在数据库(CACHE)中,若存在,则假设成立。推理结束验证下一个假设,否则转下一步。

- (2) 判断这些假设是否是证据节点。若

是,系统向用户提问,否则,转下一步。

- (3) 找出结论部分包含此假设的那些规则。把这些规则的所有前提做为新的假设。

- (4) 重复(1)、(2)、(3)步。

为了使推理更加灵活,系统还设置:

Prosupportion(T) = 条件,其含义是:当满足条件时,才能对T求值。通过对T设置求值的前提条件来实现人们习惯的正向推理。这样把反向推理与正向推理很好地结合在一起,不仅增加了灵活性,同时也提高了系统的效率。

在推理机的设计中,允许系统使用非精确推理。系统采用类似EMYCIN中的可信度(CF)来处理不确定性。CF可通过下列公式计算:

$$CF_P = \begin{cases} \min(CF_1, CF_2, \dots, CF_m) & \text{AND情况} \\ \text{合成算法} & \text{OR情况} \end{cases}$$

其中, CF_P 表示前提部分的可信度, CF_i 表示用户回答的可信度。

合成算法: (同一事实可信度的变更)

$$CF_{\text{combined}}(CF_{\text{old}}, CF_{\text{new}}) = CF_{\text{old}} + (1 - CF_{\text{old}}) \times CF_{\text{new}}$$

4. 效益分析

系统提供普教、二教系统开发计算机辅助教育软件的投资分配及其效益分析。其中定性分析主要阐明软件开发的意义等。定量分析是在系统给定的效益函数基础上,通过动态规划方法计算出的投资分配额进行效益分析的。经反复推导与测试,确定的效益函数为:

$$F(X) = A_1 + A_2X + A_3X^2 \quad (X \text{ 为投资额})$$

其中, A_1 表示社会综合效益值。 A_2 表示计算机及其辅助教育软件的普及率。 A_3 表示计算机辅助教育的利用率。

三、优化求解系统(PSS)

1. 系统设计

该系统是天津市计算机辅助教育决策与咨询专家系统的一个子系统。主要用于求解计算机辅助教育软件的开发规模及其投资的最优化解。

优化求解系统通过与专家系统DES的部分数据交换,一方面完成对各种数学模型框架的量化处理及其求解;另一方面,将所求出的最优化结果以CACHE 所要求的格式形成专家系统的部分知识库,作为专家系统进行效益分析和其他推理的部分量化事实。其总体结构如图3所示。

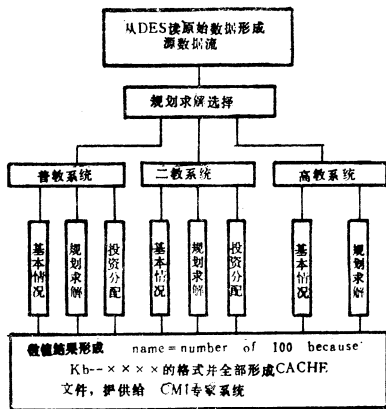


图3 最优化求解系统总体结构示意图

2. 算法描述

PSS对最优化求解问题采用线性规划中的改进单纯形算法;投资分配利用动态规划的投资问题算法,以下分别叙述这两个算法。

(1) 改进单纯形算法

第一步:确定初始可行基 B 。即利用 m 个单位列向量构成的单位阵作为初始基(矩阵)。

第二步:计算单纯形乘子:

$$Y = C_B B^{-1}$$

其中, C_B 表示基变量对应的价值系数。

(本系统价值系数为软件单位开发费用值)

第三步:计算检验数。非基变量的检验数行向量为: $\Gamma_N = C_N - YN$

如果其所有分量均小于等于零,则停止计算,这个解为最优解。否则,转下一步。(本系统设定所有分量均小于 $\epsilon = 10^{-6}$)。其中, N

表示非基子矩阵。

第四步:从 Γ_N 的正分量中选取最大的正分量 γ_k ,则让 A_k 进入基,并按下式计算 A'_k :

$$A'_k = B^{-1}A_k$$

若 A'_k 的所有元素 a'_{ik} , ($i = 1, 2, \dots, m$), 均小于等于零,则该线性规划问题无界。否则,转向下一步。

第五步:计算 θ :

$$\theta = \min \{ x_{B_i} / a'_{ik} \mid a'_{ik} > 0, i = 1, 2, \dots, m \}$$

其中, x_{B_i} 为对应于基 B 的第 i 个基变量之值; a'_{ik} 为 A'_k 的第 i 个元素的值,仅选用正元素。设 $i = 1$ 时上述比值最小,即 $\theta = x_{B_1} / a'_{1k}$,则以 A_1 为退出向量。

第六步:利用系数变换公式:

$$\begin{cases} a'_{ij} = a_{ij} - (a_{ik} / a'_{1k}) * a_{1j}, (i \neq 1) \\ a'_{1i} = a_{1i} / a'_{1k} \end{cases}$$

计算新基的逆阵 $(B')^{-1}$ 及相应的解向量 x'_B 。然后转向第二步。

(2) 投资问题算法

求一种分配方案,使得把一定数量的资源按这种方案分配给某些项目中能获得最大的总效益。本法适用于PSS的投资问题。

为阐明计算步骤,假设:

$G(I, x)$ 表示投资 $(x-1)$ 单位资源到项目 I 所得效益。

$F(I, x)$ 表示投资 $(x-1)$ 单位资源到项目 $I, I+1, \dots, N$ 所得最优效益。

$D(I, x)$ 表示当把 $(x-1)$ 单位资源按最优方案投资到 $I, I+1, \dots, N$ 项中, I 项应分到的资源数。

$XSTAB(I)$ 表示当把全部资源 K 投资到全部项目 N 时的各个项目的最优资源分配数。

具体算法如下:

1)设最后项目是唯一投资项目,则计算:

$$\begin{cases} F(N, X) = G(N, X) \\ D(N, X) = X-1, (X=1, 2, \dots, K+1) \end{cases}$$

2) 设最后两项是仅有项目, 令 $I = N - 1$

3) 令资源数为零, 并令 $X = 1$

4) 计算 $(X - 1)$ 单位资源数投资到项目 $I, I + 1, \dots, N$ 的最优效益及此时项目 I 的投资数。

$F(I, X) = \text{MAX}[G(I, Z) + F(I + 1, X - Z + 1)],$
 $Z = 1, 2, \dots, X$
 $D(I, X) = \text{产生 } F(I, X) \text{ 的 } Z \text{ 值。}$

5) 检查资源数是否已达最大数。

若 $X = K + 1$, 转 6); 否则令 $X = X + 1$,

返 4)。

6) 检查问题是否已解决:

若 $I = 1$, 转 7); 否则, 令 $I = I - 1$, 返

3)。

以下是确定投资到每个项目的单位数:

7) 令 $XSTAR(1) = D(1, K + 1)$

8) 令 $I = 2$

9) 令 $SUM = \sum XSTAR(J)$

10) 令 $XSTAR(I) = D(I, K + 1 - SUM)$

11) 若 $I = N$, 循环结束, 打印并停机; 否则,
 $I = I + 1$, 返 9)

3. 数学模型框架

结合本市的教育以及计算机应用与开发的实际情况, PSS在规划求解中建立了普教系统模型、二教系统模型以及高校模型。现以普教模型为例, 说明这三个模型的建立方法。

在实际优化求解之前, 先由专家系统提供主要的系数矩阵 A 的值、限定向量 B 的值与全部价值系数向量 C 的值。通过对以上原始数据的处理和转换, 得到一个有 24 个决策变量、18 个约束条件的基本模型:

投资目标函数: $S = \sum_{i=1}^{24} C_i X_i$

高校能力约束条件:

$$a_1 x_{13} + a_2 x_{16} + a_3 x_{19} + a_4 x_{22} \leq b_1 \quad (1)$$

$$a_1 x_{14} + a_2 x_{17} + a_3 x_{20} + a_4 x_{23} \leq b_2 \quad (2)$$

$$a_1 x_{15} + a_2 x_{18} + a_3 x_{21} + a_4 x_{24} \leq b_3 \quad (3)$$

各类软件开发量的约束条件:

$$x_1 + x_{13} \leq b_4 \quad (4)$$

$$x_3 + x_{15} \leq b_5 \quad (5)$$

$$x_1 + x_2 + x_3 \geq b_6 \quad (6)$$

$$x_1 + x_2 + x_3 + x_{13} + x_{14} + x_{15} = b_7 \quad (7)$$

$$x_4 + x_{16} \leq b_8 \quad (8)$$

$$x_6 + x_{18} \leq b_9 \quad (9)$$

$$x_4 + x_5 + x_6 \geq b_{10} \quad (10)$$

$$x_4 + x_5 + x_6 + x_{16} + x_{17} + x_{18} = b_{11} \quad (11)$$

$$x_7 + x_8 \leq b_{12} \quad (12)$$

$$x_9 + x_{21} \leq b_{13} \quad (13)$$

$$x_7 + x_8 + x_9 \geq b_{14} \quad (14)$$

$$x_7 + x_8 + x_9 + x_{19} + x_{20} + x_{21} = b_{15} \quad (15)$$

$$x_{10} + x_{22} \leq b_{16} \quad (16)$$

$$x_{12} + x_{24} \leq b_{17} \quad (17)$$

$$x_{10} + x_{11} + x_{12} + x_{22} + x_{23} + x_{24} \leq b_{18} \quad (18)$$

引进适当的松弛变量后, 将以上模型转化为标准形式:

$$\begin{cases} \text{Max} & S = CX \\ & AX = B \\ & X \geq 0 \end{cases}$$

其中, S 为目标函数, A 为系数矩阵, B 是限定向量, C 为价值系数向量, X 是决策向量。由于实际问题需要求投资的最小值, 故将目标函数乘以 “-1”, 然后按最大化求解。所求最优解和原问题的最优解相同, 但目标函数变号。

由于模型的全部系数由专家系统通过咨询给出, 从而保证了数据的可靠性和提高了可信程度。同时, 为进行多方案的比较选择提供了方便。节省了人工确定模型系数的工作量。其他两个模型与此类似, 这里不再赘述。

高考阅卷自动化系统

安徽省经济信息中心 黄效国

摘要 本文介绍利用微机进行高考标准化考试中客观题处理的系统——高考阅卷自动化系统。着重阐述设计思想、实践过程和实施效果。

一、引言

所谓阅卷自动化是指将光电阅读器和计算机联结在一起而形成的自动阅卷系统；在现阶段，它又是评阅传统考试中多重选择题的一种先进的非人工阅卷方法。几十年来，在我国高考及各类考试中，基本上采用的是传统的考试形式和人工阅卷的方法，不仅效率低，差错率高，而且需花费大量的人力、物力和财力。据有关资料统计，全国每年仅用于高考阅卷的费用就高达近一千万元人民币，动用数十万阅卷教师和工作人员。若加上其他类型的考试及阅卷，那花费的人力、物力和财力就更大。因此推广阅卷自动化技术，无论从考试制度与考试形式改革的角度，还是从经济节约的角度都是具有十分重要的意义。

二、系统的设计思想

1. 基本思想

系统模拟人工处理考试分数的方法，充分利用计算机处理速度快、存贮密度大、计算准确、打印清晰等特点，将考试的有关信息输入到计算机里，然后由计算机加工处理，并把结果反馈给我们。为提高处理速度，节省内存空间。我们对整个系统进行了编译处理。

2. 怎样用现代化手段进行自动评卷

众所周知，人工评卷是一个较为复杂的脑力劳动过程，在评阅过程中，评卷人员要用眼睛看考生的试卷内容，然后将看到的内

容传给大脑，大脑要进行一系列的分析判断，还要同标准答案进行比较，并根据比较的结果和评分标准得出考生的实际得分。根据这一特点，采用人工智能模拟的原理，将OMR光电阅读器与微机连机，形成阅卷自动化系统，达到了机械自动评阅的目的。

OMR是一种智能仪器，它通过阅读光学符号来识别数据信息。数据卡上的光学符号与所阅数据之间有着代码对应关系。数据卡实质上是一个光学符号位的图阵，它上面的光学符号位的编排是根据所要记录的数据内容来确定，不同的数据编码适应于不同的数据类型和不同的使用要求。根据OMR光电阅读器和数据卡的特点，我们认为采用光学符号识别方式能较好地适应大规模考试的阅卷需要，而且OMR设备的原理和操作方法都比较简单。

就全国高考处理而言，国产微机无论在信息储量和处理速度方面都完全可以胜任。并且具备较强的汉字功能，更为使用提供了方便。

由于以上两种设备在功能上具备了阅卷的能力，完全可以模拟人工的方法进行阅卷，这就从原理上保证了阅卷的可行性。我们通过RS232接口将两部分设备组合连接成一个系统。在该系统中，微机将OMR阅读后传来的信息进行处理，并根据标准答案和评分标准得出实际分数。然后按规定的格式存放在硬盘或软盘上，以备使用。

3. 阅卷过程中特殊情况的处理

在编制程序过程中，考虑到考生人数多，情况复杂，为了工作方便，在程序中对各种可能出现的情况设置了自动处理功能，

主要有以下几个方面:

(1) 状态初始化: 由于高考的试题答案和评分标准在考试前绝对保密, 我们在程序设计时安排了状态初始化功能, 即在拿到标准答案和评分标准后, 在很短的时间内就可输入计算机中, 进入正常阅卷。

(2) 在微机屏幕上随时显示目前的工作状态, 这些信息包括: 已阅考场数, 存盘记录条数, 以及等待阅卷、正在阅卷、正在统分等工作状态提示信息, 以便在工作过程中进行动态管理。

(3) 首考号的生成: 在阅卷过程中, 微机可根据任一张试卷考号自动生成该考场的首考号, 保证不出差错。

(4) 突然停电的处理: 由于某种原因突然停电会造成计算机来不及记录下目前状态的现象, 使操作者难以在供电后继续工作; 对此我们在程序设计中, 一是增加了“修改记录状态”功能块, 二是在阅卷时每阅一个考场存盘一次可保万无一失。

(5) 抽样处理: 计算机用于考试, 为考试的科学管理提供了条件。作为一次完整的考试, 用教育统计学的方法去分析试题和考试成绩情况是不可缺少的, 只有这样才能为改进教学和命题提供第一手资料。为此, 我们在机器评阅的同时, 按一定比例对试卷进行抽样, 以便对试卷进行定量, 定性分析。

(6) 异常情况的处理: 对非正常试卷, 包括未涂考号、涂错考号、涂重考号三种情况采用区间排除法, 并通过计算机准确预报, 提示出是哪张试卷异常。对在印刷和使用过程中产生的无法评阅的异常试卷, 程序允许键盘输入。另外, 程序还增加试题类型判断、科目判断、试卷边框判断、检查某考场未输入试卷号等一系列便于操作的功能。这些操作均为汉字提示, 为非专业人员的使用提供了方便。

(7) 日志文件: 这是为检查和跟踪阅卷

过程的操作是否正确, 同时自动记录下阅卷过程中所遇到一系列错误和处理办法。

由于比较圆满地解决了以上问题, 使阅卷管理的方法比较科学严谨, 工作的思路比较清晰。

三、系统的实现过程

今年安徽省参加高考的考生近十五万人, 有七门课的客观题, 近五十万张试卷, 需自动化评阅。在整个系统实现过程中, 根据计算机处理速度快的特点, 采取先建立数据文件, 再对该数据文件进行处理的原则; 也就是说先用光电阅读器将试卷上的信息读入计算机硬盘或软盘中, 然后再对这些信息处理。其操作过程, 如图1所示。整个评卷过程大致分为三个阶段。

1. 数据输入阶段

这一阶段的重点是建立考生试卷的原始数据文件, 它是下一步处理的基础和依据。

第一步: 由程序员对系统状态初始化; 通过这一功能, 计算机可掌握考试规模, 提示工作状态, 设置系统保密字, 确定考试有关参数等。

第二步: 通过系统检测和自检功能, 检测初始化结果是否正确, 自检计算机和光电阅读器能否在正常的状态下工作。

第三步: 调用数据读入程序; 数据读入的准确度如何, 直接关系到考生成绩, 也是计算机自动评阅成败的关键。为此, 本系统在输入方式上具有极大的灵活性。可根据需要随意地阅读任何一地区, 任何一科目的试卷, 而无须多虑它们的先后顺序; 另外, 对同一地区, 同一科目的试卷, 也可分批阅读。为了保证试卷评阅的准确性, 每个考生的试卷均阅读两次, 分别写到软盘和硬盘上, 以备后用。

2. 改卷阶段

改卷是计算机评阅的关键一环，它将读入考生试卷上的原始数据与标准答案比较，并按规定的评分标准，将该考生该科目的最后得分，分别存入软盘和硬盘，以备审核。

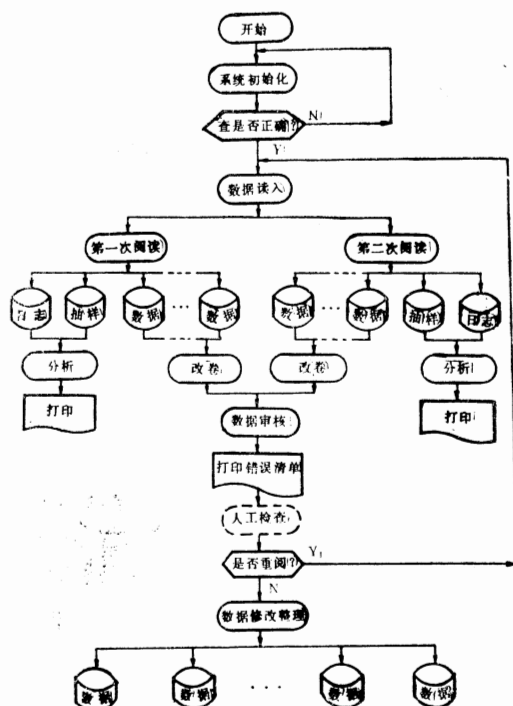


图1 系统流程简图

3. 数据审核阶段

这一阶段是为了准确得到考生的各单科成绩。

第一步：调用数据审核功能；该功能对两次读入的考生单科成绩进行比较，复核。打印机将记录下两次读入的不同的值，便于修改读入的错误。同时该功能还对每一个地区，每一个科目的考生人数、考场数、考生考号进行审核，并检查有无漏阅的考场和重复阅读的考场，打印机也做跟踪记录，以备检查。

第二步：调用数据修改功能。该功能是根据数据审核时，输出的错误清单，对数据

文件进行修改。它有二种方法：其一，是对某个考场整个的考生的成绩进行修改；其二是对某一考生的成绩进行修改。

第三步：操作人员针对数据文件审核、修改的情况，对数据文件进行查询、合并、排序、删除、打印和更名等操作。使数据文件便于今后使用和管理。

四、系统的实施效果

由于高考试卷评阅要求数据准确无误，又要求如期完成任务，本系统经多次模拟试验后，于89年7月中旬正式投入运行，收到了预期的效果。实践证明本系统具有如下特点：

1. 大大提高高考评阅工作的准确性，只要严格按工作规程做，可取消评阅过程中误差，提高考试的权威性。改变了以往人工评阅所发生的错误现象。

2. 大大提高了评阅工作的速度，由于计算机处理速度远远大于人工，就使得评阅效率明显提高。一般一人使用一台计算机评阅，可相当30至50人的工作量。

3. 可提供大量有用的信息，方便选拔人材，并可进行教学信息反馈，更好地发挥考试的作用。

4. 减少人为干扰，防止违纪现象出现，提高考试可信度。由于人工评卷用人的环节多，要做到各个环节上公正合理是困难的；尤其是通过考试进行选拔，社会影响面大，关系考生前途和千家万户的直接利益，容易在评阅上出现违纪，甚至舞弊现象；而使用计算机评阅，用人极少，有利于杜绝这些现象的发生。

5. 节省人力、物力、财力，提高经济效益。计算机评阅的开支一般比人工评阅节省开支百分之八十左右，而且输出资料正规、清晰。

软盘加密技术的探讨

石家庄军械工程学院 陈致明 寇应展

摘要 本文以IBM PC/XT为例,探讨了微型机软盘信息加密技术,并通过一些实例,介绍实现加密的方法并附有软盘加密的程序。

一、引言

随着计算机工业的迅猛发展,微型机的应用日益普及,各种系统软件、应用软件越来越丰富。软件是知识密集的产品,许多用于重要部门的数据和信息的保护,对于某些软件版权的保障,对于各种研制成功的软件,尤其是专用的系统软件和军用的软件,如何确保其安全,正在受到人们的重视。对软盘信息加密,采取防复制软件的保护技术,这是计算机安全保护技术的一个重要方面,是广大软件工作者研究的一个重要课题。

二、软盘组织格式及其读写操作

1. 软盘组织格式

在PC-DOS操作系统中,系统的FORMAT命令向用户提供了一套标准的软磁盘格式。若使用双面、双密度的5 $\frac{1}{4}$ 吋软盘,它可以格式化成双面9扇区。若以双面9扇区为例,软盘经格式化后,每面为40个磁道(0~39),每一磁道又分为9个扇区,每一扇区大小为512字节。这些扇区为方便系统管

理,要由DOS对扇区进行统一的逻辑编号,编号的规则如下:0道0面的第1扇区为0扇区,0道0面第9扇区为8扇区,接着道号不变,从1面对应磁道编,即0道1面第1扇区为9扇区,0道1面第2扇区为A(16进制)扇区……直至编完。

在720个众多扇区中,PC-DOS分成四类扇区,它们分别是:

(1) 引导程序区,它存放在0道0面0扇区,当系统启动时,由它来引导。该扇区用来存放:磁盘的OEM标识、BIOS参数表和引导程序。引导扇区第一字节是一转移指令,若JMP指令的操作码不存在,则表明磁盘未被格式化。引导扇区结构如下:

00H	转移指令(3字节)
03H	OEM标识及版本号(8字节)
0BH	每扇区字节数(2字节)
0DH	分配单元扇区数(1字节)
0EH	保留扇区(2字节)
10H	FAT数(1字节)
11H	根目录数(2字节)
13H	总扇区数(2字节)
15H	介质描述符字节(1字节)
16H	每FAT扇区数(2字节)
18H	每道扇区数(2字节)
1AH	磁头数(2字节)
1CH	隐藏扇区数(2字节)
1EH	引导例程

:

6. 在操作上,系统具有Write/Read/Change三种功能,可以方便地进行输入/输出/修改等工作。此外,非专业人员,稍加培训,便能熟练地使用本系统。

通过实践证明,本系统基本满足高考标

准化考试客观题评阅要求。实现了高考自动化阅卷的目的。本系统的使用成功有助于标准化考试推广以及考试制度和考试形式的改革。同时它也将在推进考试管理科学化中发挥重要作用。

(2) 文件分配表区FAT

它存放在0道0面的1—4扇区，它记录文件占用的空间。FAT 中存放着每个文件占有的簇的连续表。对于PC机，一簇为2个扇区，FAT 共有一式两份文件分配表，分别占用1—2和3—4扇区。

(3) 文件目录区

它存放在0道0面的5—8及0道1面的9~B扇区。它紧跟在文件分配表之后，它用来存放目录。每一目录项有32字节，用来说明文件、子目录和卷标，其中根目录的大小在引导程序的0011H中已指定。文件目录区共占用7个扇区，共有 7×512 个字节，即最多可存放 $7 \times 512 / 32 = 112$ 个文件名。

(4) 数据区

目录区之后的剩余部分扇区称为文件区或数据区，它用来真正存放文件或数据。

2. 软盘的读写操作

软盘的读/写操作依据软盘格式化时的格式参数进行操作。软盘的格式参数存放在0000:0078H为首址的连续四个单元中，即1EH号中断入口。软盘格式参数共 11 个字节，含义分别为：

字节1：高四位表示步进频率，低四位表示磁头去载时间；

字节2：表示磁头加载时间和DMA 操作；

字节3：表示操作后马达关闭时间；

字节4：表示每一扇区有多少字节，即0为128字节，1为256字节，2 为512 字节，3为1024字节。

字节5：表示每一磁道有多少扇区；

字节6：读/写时间间隙长度；

字节7：表示数据长度；

字节8：表示格式化时的间隙长度；

字节9：表示格式化时填充字符；

字节10：表示磁头稳定时间；

字节11：表示马达启动时间；

若这11个参数一经改变，则格式化的磁盘为非IBM—PC标准格式。

软盘的读写操作和格式化可调用 ROM—BIOS中的INT13H软中断。INT 13H 参数说明如下：

AH=2时，读指定扇区到内存；AH=3时，写内存指定扇区；AH=5 时，格式化指定磁道。在读/写、格式化时，所使用的寄存器为：DL 为驱动器号；DH 为磁头号；CL 为扇区号；CH 为磁道号；AL 为扇区数；ES:BX为缓冲区地址。在格式化时，ES:BX指向格式化标志字段，每个字段分别由四个字节组成，即为：磁道号，磁头号，扇区号，扇区字节数。

三、软盘加密的实现

防止复制软盘，进行软件加密，是与软盘的组织及读/写方式密切相关的。下面给出用软件易于实现的几个方法：

1. 简单加密

最简单的加密方法，一是使用半个汉字或不显示的区位码来组成的文件名，这样，在用DIR命令显示目录时，一般情况下，仅显示空格(20H)，但实际上并非为20H；二是用工具软件改变文件的属性，使其变为隐式文件，不能用DIR来显示。由于找不到正确的文件名，因而无法进行读/写操作，起到了加密的作用。但此类加密可用DEBUG.COM来快速解密。解密方法如下：

- (1) C>DEBUG ↵
- (2) -L 1000-0-5-7 ↵
- (3) -D 1000 ↵
- (4) -E x x x ↵
- (5) -W 1000-0-5-7 ↵

上述(1)为进入DEBUG状态(划横线表示从键盘键入的内容)；(2)为将A盘的目录区读到内存指定单元(即将从第5扇区开始，连续7个扇区的内容读到内存以1000H为首址的单元中)；(3)用D命令来显示A盘的目录，此时，系统文件，隐式文件均可看到，而且半个汉字，不显示的区域码一目了然；(4)找到要解密的文件名所在的地址，用E命令来

修改为可显示的字符；(5)写盘,用Q命令返回DOS,即可用DIR来显示了。

2. 查磁盘状态字

予先在磁盘某一区存放软盘的状态字,由系统执行时判定,查是否为原盘。若不是,可转出错处,或转走,或退出,使其不能正确执行,达到了加密的作用。下面所示的程序查0道0面1扇区0002H单元,判是否为0CDH,若不是,则转出错处。

程序如下:

```
BUFFER DB 512 DUP ( ? )
MOV BX, OFFSET BUFFER
MOV AX, 00H
MOV DX, 00H
MOV CX, 01H
INT 25H
ADD BX, 0002H
MOV AL, [BX]
CMP AL, 0CDH
JNZ ERR
```

这些状态字最好放在非用户使用区,这对于DOS提供的COPY命令来说,是不可拷贝的。

3. 变换文件目录区或FAT区

软盘上存放什么文件,是由目录区中的内容告诉用户的,如果能对目录区的内容加密,则即使全盘拷贝,也不能使用。

目录区中每一个目录项占用32个字节,但对于用户来说,前11个字节即文件名和扩展名,在DIR执行时,显示在屏幕上。若能将目录区每一目录项的前11个字节变换,使其发生混乱,然后再写回磁盘上。这样,非法用户在用DIR查看磁盘目录时,屏幕上会出现一片随机字符,使其不能真正理解磁盘目录内容。自己查看时,可用上述变换的逆变换来解密,从而起到了加密作用。

FAT为磁盘分配表,将其内容改变,则即使文件名正确,也无法进行读写操作,从而达到了加密的作用。

附录1所示的程序将A盘的目录区中每一目录项前11个字节的内容循环右移5位,再写

回磁盘。

若利用系统调用隐式键入要移位的次数,移位之前,再将文件名和扩展名与一密钥相异或,再移位,而且上述的操作只对目录区中有文件名和扩展名的目录项操作,同时并对所有文件的起始扇区号变换,使其即使解开文件名和扩展名,也不能正确读/写操作,进一步提高破译的难度。

4. 奇异格式化

对于上述的方法,都可用DEBUG来解密,即软盘的读写参数同于PC机的标准参数。若能使用一些奇异的格式化参数来格式化一磁盘。迫使标准DOS不能进行正常读写。以下的几组格式化参数,不难从中看出加密的意义。

(1)标准DOS 2.1的参数为:

DFH, 2, 25H, 2, 9, 2AH, FFH.
59H, F6H, 0, 2

即每道为9扇区,每扇区为512字节

(2)电子工业部六所的CCDOS 1.0汉字字库参数为:

DFH, 2, 25H, 3, 4, 35H, FFH,
74H, FBH, 0, 4

即每道为4扇区,每扇区为1024字节

(3)PC机奥林匹克十项全能游戏的参数为:

CFH, 2, 1EH, 1, EH, 2AH, 0,
50H, EAH, 19H, 4

即每道为14扇区,每一扇区为256字节

而且,用某组参数格式化出来的盘,用标准格式参数去读/写就会出错,因而无法读写。

改变磁盘的格式参数,即将新格式参数首地址送入0000:0078H单元中,亦即修改1EH入口地址。

或利用PC-DOS的FORMAT命令先格式化一磁盘,然后将某一磁道格式化成非标准格式。下面所示的程序就是实现非标准格式化,使DOS在读取该道信息时,发生错误,从而起到了加密的作用。


```
TABLE DB 0, 8, 01, 2, 0, 8, 03, 2, 0, 8,
      05, 2
      DB 0, 8, 07, 2, 0, 8, 09, 2, 0, 8,
      0A, 2
      DB 0, 8, 0C, 2, 0, 8, 0E, 2, 0,
      8, 11, 2
      MOV DH, 00H
      MOV DL, 00H
      MOV CH, 08H
      MOV CL, 01H
      MOV AH, 05H
      MOV AL, 09H
      INT 13H
```

5. 反动态跟踪

由于人们对计算机系统的内部结构、操作系统和汇编语言的了解，用户可以灵活地使用动态调试程序DEBUG.COM对各种各样的软盘进行破译分析，因此采用反动态跟踪技术来阻止技术人员破译软件已成为一种很重要的方法。反动态跟踪技术可以利用正常运行时间与调试运行时间不同来反跟踪；也可以用破坏DEBUG程序的T命令和G命令来做；还可以利用在DEBUG下存放键盘键入的文件名的位置与在COMMAND程序存放键盘键入的文件名的位置来判定，查看是否在DEBUG下调试执行。下面分别介绍后两种：

(1) 破坏0号中断和3号中断入口地址

ROM—BIOS的0号中断程序是完成单步中断处理的。3号中断是完成断点中断处理的。即常用的T命令和G命令，而这两条命令又是通过调用BIOS的0号和3号中断处理程序实现的，因此，可在程序中破坏0号和3号中断处理程序入口地址。使其使用T命令和G命令时发生混乱或死锁，致使其不能正常跟踪。下面所示的程序就是破坏0号和3号中断处理程序的入口地址。

```
XOR AX, AX
MOV ES, AX
MOV DI, AX
MOV AX, CS
MOV DS, AX
MOV SI, 0700H
MOV CX, 0010H
```

REPZ MOVSB

(2) 利用DEBUG和COMMAND存放键盘键入的文件名位置来判定

若用DEBUG命令时，其后面跟一个文件名和扩展名时，COMMAND.COM文件将上述的文件名和扩展名存入DS:0082H为首址的连续单元中，实际长度为实际字符个数加1；而DEBUG程序将键盘键入的文件名和扩展名存放在0000:2D6DH处，此两处存放文件名和扩展名的结构不同，若二处存放的为同一内容，则说明正在DEBUG下调试执行，可以破坏使其跟踪失败。附录2所示的程序则可检测出是正常执行还是调试执行。若正常执行，程序显示NEXT..；若调试执行，程序显示TRON..。可将其部分插入在要保密的程序中，防止跟踪。

四、结束语

采用加密技术，进行软件保护，是计算机软件安全保护技术的重要措施。由于它的实用性，正被越来越广泛地应用于各种各样的软件和程序上，并收到良好的效果。随着计算机的广泛应用，人们对计算机的各种技术将越来越熟悉，前述的单一方法，已很难低御解密者，所以人们常常采取多种方法的组合：如在程序中，多次加密，再加上一些允许次数；或配上反动态跟踪措施，以使非法者很难破译。本文所介绍的加密技术，是侧重于软加密技术。除此之外，对于硬加密技术目前也正在研究使用，由于篇幅所限。本文不再赘述了。

附录 1

```
DATA SEGMENT PUBLIC 'DATA'
DIRE DB 3584 DUP (0)
DATA ENDS
CODE SEGMENT PARA PUBLIC 'CODE'
START PROC FAR
      ASSUME CS: CODE, DS: DATA
      PUSH DS
      MOV AX, 0
```

```

        PUSH AX
        MOV AX, DATA
        MOV DS, AX
        MOV CX, 3
RETRY:  PUSH CX
        MOV BX, OFFSET DIRE
        MOV AX, 0
        MOV DX, 5
        MOV CX, 7
        INT 25H
        POP CX
        POP CX
        JNC READDP
        MOV AH, 0
        INT 13H
        LOOP RETRY
READDP: MOV CX, 112
        PUSH BX
LOOPPC: MOV DX, 11
LOOPR:  PUSH CX
        MOV CL, 5
        ROR BYTE PTR [BX], CL
        POP CX
        INC BX
        DEC DX
        JNZ LOOPR
        ADD BX, 21
        LOOP LOOPC
        POP BX
        MOV AX, 0
        MOV DX, 5
        MOV CX, 7
        INT 26H
DONE:   MOV AH, 4DH
        INT 21H
        MOV AH, 4CH
        INT 21H
START   ENDP
CODE    ENDS
        END START

```

附录 2

```

DATA    SEGMENT
TRON    DB 'TRON..', 0DH, 0AH, '$'
EXEC    DB 'NEXT..' , 0DH, 0AH, '$'
DATA    ENDS
CODE    SEGMENT

```

```

START   PROC FAR
        ASSUME CS: CODE, DS: DATA
        PUSH DS -
        PUSH DS
        POP AX
        MOV BX, AX
        MOV AX, 0
        PUSH AX
        MOV AX, DATA
        MOV DS, AX
        MOV ES, AX
        PUSH DS
        MOV CX, 11
        MOV AX, 0
        MOV DS, AX
        MOV ES, AX
        MOV SI, 2D6DH
        MOV DS, BX
        MOV DI, 0082H
SORT:   MOV AL, DS:[DI]
        CMP AL, ''
        JNZ SOR1
        INC DI
        JMP SORT
SOR1:   MOV DL, ES:[SI]
        CMP DL, ''
        JNZ SOR2
        INC SI
        LOOP SOR1
SOR2:   CMP AL, DL
        JNZ NEXT
        INC DI
        INC SI
        LOOP SORT
        POP DS
        MOV DX, OFFSET TRON
        MOV AH, 09H
        INT 21H
        JMP DONE
NEXT:   POP DS
        MOV DX, OFFSET EXEC
        MOV AH, 09H
        INT 21H
DONE:   MOV AH, 4DH
        INT 21H
        MOV AH, 4CH

```

C语言实用技巧几例

C语言是一种强有力的通用程序设计语言,许多编程人员为了提高自己源码的功效,常常加入一些自己的技巧,使开发出的C源码更为简洁而且颇具个性。

本文举出几例实用技巧并分析以供读者参考。

1. 对于多参量函数被调用情况,应先建立一个结构来存放这些变量,再将指向这一结构的指针做为参数传递给被调用函数,这样做有两点好处,(1)可以使复杂的参量传递变得非常清晰、简洁。

(2) 由于函数都要参数传递,长长的参数会很快占尽代码空间,而指针会大大减少占用的空间。这种技巧对于递归程序更有奇效。

2. C语言是一个高度可移植语言,所以我们使用C时要特别注意程序一定要高度可移植,可以使用编译命令来设计数据长度。

```
#if sizeof(int) == 2
#define INT long      /* 将十六位机整数
                        定为32位 */
#else
#define INT int
#endif                /* 三十二位机不变 */
```

这样设计出的C语言可以在16位微机和32位小型机上运行。另外条件编译在调试阶段也是非常有用的处理手段。

3. 巧用任意类型说明(void)来解决两个矛盾类型之间的调用。例如,要求结构a中有一个指针是指向B结构类型的,而结构B中又有一个指针指向a结构类型,正常的方法已经不可能安排这两个结构说明的先后次序,可以引入void,

```
struct a
{
    .....
    void *pa
};
struct b
{
    .....
    struct a *pb
};
```

在对pa指针赋值时,相当于对*pa进行强令指针变换/*struct b */pa = &.....。这种技巧为任意多结点连通图的模型的建立及遍历提供了一种方法。

天大冶金分校 开校军

```
INT 21H
START ENDP
CODE ENDS
END START
```

参 考 文 献

1. MS-DOS操作系统高等教程,李沐荪编译。北京:北京科海总公司培训中心,1987.9。
2. IBM-PC奥秘专辑, [美]皮特·诺尔顿著。北京:《计算机技术》副刊编辑部,1985
3. 磁盘机原理与软件加密解密实用技术,夏东海著。北京:北京科海总公司培训中心,1988
4. 密码学—计算机数据安全的一个新领域, [美]卡尔·梅尔等著,刘景伊艾译。北京:国防工业出版社,1988.7

DBASE III与BASIC程序相互调用面面观

天津棉四计算机室 陈建华 田燕波

摘要 本文比较详细地讨论了在IBM系列机支持下的DBASE III程序与BASIC程序相互联结,及BASIC程序直接调用·DBF文件的理论和应用实例。已建立的数据库文件经BASIC程序读取后直接应用及写入,并提供了一种主开发系统软件同时使用多种软件系统的连接方法。

DBASE III 具有很强的数据处理能力,且简单易学、操作方便、编程周期短。但对于大量复杂的计算,对于图象处理等方面的综合问题,就显得大为逊色。

当我们设计一个管理系统时,可能有大量的数据需要整理、加工并等待存入数据库中。有的数据在建库之前就已经以某种形式存在,我们就应在数据库中尽量利用这些数据建立数据库之间的通讯,充分发挥DBASE III的高效率、强数据处理的功能,充分发挥高级语言进行计算及其他处理的功能,建立一个功能完善的应用软件系统,成为亟待解决的问题。

一、DBASE III 程序调用高级语言程序

我们知道,在DBASE III 中设置了一个功能很强的语句RUN,其格式:RUN <命令>

这个语句可执行PC-DOS的内部命令,也可执行其外部命令。在执行完这些命令后,能把控制自动交还DBASE III。被调用的高级语言程序通常通过编译、连接形成.COM或.EXE文件,他们以外部文件的形式存储在磁盘上。当DBASE III 执行RUN语句时,会把这些文件调入内存,并以其入口开始运行。

只要高级语言程序已成为独立的可执行程序,DBASE III 总可用 RUN 语句来调用它。

二、DBASE III 与高级语言程序进行数据传递

1. 数据传递的实现:

由于DBASE III 没有直接提供与高级语言的接口,因而它与高级语言间数据的交换只能利用间接的方法进行。

这样,需要我们对DBASE III 文件类型进行分析:其文本文件都是 ASCII 文件,文件的数据都是以 ASCII 码的形式存储,而高级语言均有处理ASCII码文件的功能。因此,利用DBASE III 的文本文件可实现这种传递。

文本文件(.TXT)与数据库文件(.DBF)的区别在于:数据库文件保存数据库的结构和实际数据,而文本文件无结构,仅有数据。而DBASE III 中设置了两个功能极强的语句COPY和APPEND,可以把数据库文件中的数据送入文本文件或把文本文件中的数据追加到数据库文件中。这样,DBASE III 与高级语言程序进行数据传递变为可行。

(1) DBASE III 送数据给高级语言程序

先用DBASE III 的有关语句把数据库文件(.DBF),再通过COPY命令将数据库(.DBF)送入文本文件(.TXT),外部软件就可使用(.TXT)文件中的数据。

(2) 高级语言程序数据送DBASE III

先由高级语言程序建立.TXT文件,再用DBASE III 的APPEND语句把 .TXT文件中

的数据送入.DBF文件中,这样,DBASE III的程序便可使用这些数据了。其传递过程如图1所示:

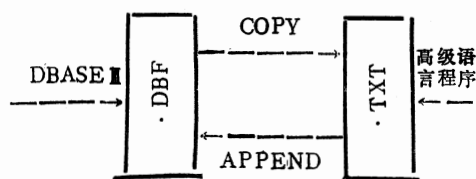


图1

2. “接口文件”的数据格式

在上面的分析中可以看到,在DBASE III程序与高级语言程序间进行数据传递时,文本文件.TXT起着关键的作用,它实际上是DBASE III与高级语言的数据接口,.TXT文件可以说是一种“接口文件”。一般数据文件有两种形式:标准格式、紧凑数据格式即非标准格式。在保证数据传递正确性的前提下,要求“接口文件”的数据格式能与双方匹配。

三、实现数据传递的DBASE III语句及其应用

由图1 DBASE III与高级语言间数据传递过程可见,COPY语句转换输出数据格式,使之成为外部程序能接收的数据文件;APPEND语句则转换外部程序产生的数据文件,使数据能传送到数据库文件中。

1. COPY语句

通过选择参数SDF、DELIMITED来改变其文件存储形式。SDF为标准格式,DE-LIMITED为紧凑数据格式。

2. APPEND语句

该语句的主要功能是将.DBF数据文件

或.TXT文件的数据输入到现行数据库文件上去。语句中<文件名>所指出的文件为提供数据文件;当前被打开的数据库文件是接收数据的文件。这些提供数据的文件可以用COPY命令建立的,也可以是用操作系统的行编辑程序。字处理程序或其他高级语言建立的。

由于提供数据的文本文件(.TXT)没有结构。因此,数据是按位置实现传递的。这时,必须注意提供数据的文件与接收数据文件在同一位置上的数据类型要一致。由于篇幅限制,不再举例,有兴趣的同行,不妨试试。

可见,用COPY语句和APPEND语句可较好实现DBASE III与高级语言程序的数据传递,为保证数据传递的准确,若特别注意有关数据格式的配合,这是DBASE III与高级语言进行数据传递成败的关键。DBASE III与高级语言的连接过程见图2。

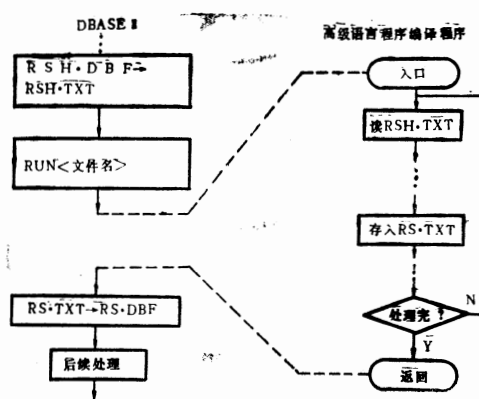


图2 DBASE III与高级语言的连接过程

高级语言处理完的RSH数据放在输出文件RS.TXT中。这样,就形成了高级语言接收DBASE III数据库文件中的数据,然后进行运算处理,最后将处理的数据又传送到

DBASE Ⅲ的数据库文件中,实现了DBASE Ⅲ与高级语言的连接。

四、DBASE Ⅲ与BASIC 程序的数据交换

1. BASIC数据文件存储格式分析

在BASIC语言中,其文件系统包含的数据文件有两种类型:顺序存取文件和随机存取文件。

我们选顺序文件形式完成传递是较为适宜的。

就BASIC的顺序文件而言,数据的存取也有多种不同的形式。为搞清这个问题,我们分析下列的示例。如:设BASIC变量AS、X、Y及Z的值分别为ABC、15、12和-56。在不同的格式控制下,这些数据的存放形式为:

- (1) ABC 15 12 -56
- (2) ABC 15 12 -56
- (3) ABC, 15, 12, -56
- (4) "ABC" 15 12 -56
- (5) "ABC", 15, 12, -56

我们知道,要想得到BASIC顺序文件的不同数据存储格式,关键在于灵活掌握写盘语句PRINT #1的控制格式。

比较上例中的几种存储格式后,我们看到,存储形式(1)与DBASE的文本文件的标准格式(.SDF)要求完全一致;存储形式(5)则与非标准格式(DELIMITED)要求完全一致。所以,利用(1)和(5)的数据存储形式,把BASIC的数据文件作为“接口文件”是切实可行的。

2. BASIC读取“接口文件”的数据

在前面分析BASIC语言数据存储格式的同时,我们已解决了如用BASIC程序建立数据的“接口文件”的问题。实现DBASE Ⅲ与BASIC程序数据交换还必须解决的另一个问题是,BASIC程序如何读取由DBASE Ⅲ生

成的“接口文件”的数据,这实际上仍然是一个数据格式的控制问题。

完成这个任务的是INPUT #1语句,它从磁盘文件中顺序地输入数据。对于前述的数据存储形式(1),可用下面的BASIC程序从“接口文件”中读取数据:

```
50 OPEN "1", #1, "A2.TXT"
60 KS=INPUT$(3, #1)
70 INPUT #1, C1
80 INPUT #1, C2
90 INPUT #1, C3
100 PRINT K$, C1, C2, C3
110 CLOSE
```

如果采用(5)所示的数据存储形式,则读取数据的程序可修改为:

```
50 OPEN "1", #1, "A5.TXT"
60 INPUT #1, K$, C1, C2, C3
70 PRINT K$, C1, C2, C3
80 CLOSE
```

由于BASIC语言中的字符串是用双引号括起来的,所以在接收DBASE Ⅲ数据库文件时,一般采用DELIMITED格式较好,这样就能保证字符段的信息由字符串变量接收,数字字段由数值变量接收。

在输出时,为了保证让数据项的宽度与DBASE Ⅲ数据库字段宽度匹配,一般可采用“PRINT #, USING”形式。

五、·TXT文件及·DBF 文件的结构及利弊

综合上面论述,使用.TXT文件的读写有以下三种缺点:

- a. 程序缺乏独立性,对.TXT文件的格式顺序、宽度等必须在编写DBASE Ⅲ程序和高级语言的程序时就约定好了。
- b. TXT文件是从.DBF文件中复制出来,内容重复,造成外存空间紧张,数据冗余。
- c. 交替两种工作状态频度较高,操作和等待

时间较多。

以上三种方法浪费了时间和空间,不能很好地利用系统的效率,这是我们所不希望的。较好的办法是,语言程序在运行中能够直接访问它所需要的库文件(.DBF),而不经任何中间变换。据此,在IBM-PC/XT数据库管理系统中常用的DBASE III文件结构进行分析。在搞清文件结构和数据之间的排列规律的基础上,就可以用BASIC语言直接处理其内部数据。

DBASE III的.DBF文件中数据的安排

用高级语言直接处理.DBF文件中的数据:首先必须搞清楚.DBF文件中的数据是如何安排的。在汉字操作系统CCDOS下,使用动态调试程序DEBUG来显示一个已经建立的.DBF文件的内容结构,找出其中规律。

.DBF文件结构显示和在DEBUG状态下显示结构如图3所示。

- A: 文件属性
- B: 最后修改日期
- C: 记录个数
- D: 结构长度差2
- E: 记录长度

经过对显示结果进行分析、整理,可以得出如下规律:

.DBF文件的构成可分为两大部分:

第一部分是文件结构信息及说明,位于.DBF文件之首。继续细分,还可分为两小部分,即:文件信息部分占用32个字节,文件结构记录部分,长度是字段数目的32倍。

第二部分是文件的实际数据部分:

首先,文件结构部分的前32个字节是文件的属性描述,说明DBASE III的文体标识,文件建立或修改日期、记录个数和记录长度。第一个字节为03H或83H,表示此文件为DBASE III数据库文件,如果是83H,还表示文件中含有备注型字。如果使用“USE”打开文件时,要检测该项,若不为03H(或83H)则给出错误信息。第2~4字节分别为最后修

改日期的年、月、日。第5~8字节表示记录数。高位在后,低位在前(下同),第9~10字节用来表示数据记录的开始字节数,在数值上差2。例如,A2H 03H(十进制数的930),表示从第932(即 $930+2$)字节数开始是数据记录。第11~12字节表示记录长度。

其余部分是记录中各个字段的描述。每个字段的描述占32个字节,说明每个字段的字段名、字段类型、字段宽度和小数位。具体分配如下:第1~10字节表示字段名;第12字节表示字段类型;第17字节表示字段宽度;第18字节表示小数位数。另外,在记录值开始的行中还有两个字节,一个是字段描述结束标志(0DH,回车键),另一个是空(00H, NULL),把这两个字节也算到文件的结构中去。到此为止,可以归纳出,一个关于.DBF文结构字段数目的表达式。

$$\text{STRUCTURE} = 32 + 32 \times \text{FIELD} - \text{NUMBERS} + 2$$
经过整理得:长度=给定的字段数 $\times 32 + 34$ 。该值是变化的。自变量FIELD- NUMBERS的取值范围是1至128,故因变量STRUCTURE的取值范围是66至4130个字节。

第二部分是已写入的.DBF文件的实际数据,其存放方式为各记录连续存放,每一个记录用一个空格(前导空格)开始,记录长度包括这个空格字节,记录中的各项数据项(包括未填满该字段增补上的空格,连续存放,数据项中的字符及数字均用ASCII码形式。

数据是以记录的形式存储的,每个记录的第一个字节是空格(20H),用于存储删除标志“*”,其余部分是记录值,在最后一个记录的尾部紧跟一个文件结束标志“→”(1AH)

在文件结构记录与文件数据记录之间还有一些字节:用DBASE III直接建立的.DBF文件有3个;由DBASE III的.DBF文件转换来的.DBF文件有更多的一些字节,这些与本文无关紧要,不再详述。

	A	B	C	D	E	
	低	高				
1C0B,0100	03	50 01 01	8D 02 00 00	A2 03	E6 00	00 00 00 00 .P.....*.f.....
1C0B,0110	00	00 00 00 00	00 00 00 00	00 00	00 00	00 00 00 00
1C0B,0120	D0 D5	C3 FB	00 00 00 00	00 00	43 0F	00 98 7B PUC{.....C..{
1C0B,0130	08	00 00 00 00	00 00 00 00	00 00	00 00	00 00 00 00

图 3

六、对DBASE III 数据库

的理论分析

假如A盘上有一个DBASE III 的.DBF文件:

A: LX.DBF, 其结构如下:

为了用BASIC语言程序直接处理LX.DBF 文件中的数据, 我们从两种不同的角度来描述同一个数据库文件。如图4、5所示。

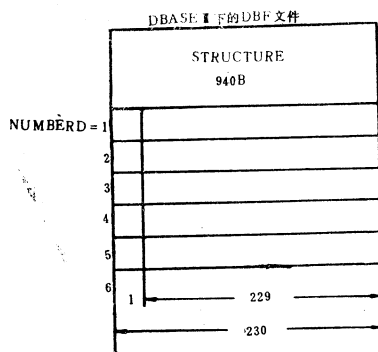


图 4

在图4、5中, 取BASIC下的随机文件的记录长度与DBASE III 下的.DBF文件的记录长度相等。

存取磁盘文件的操作通常以记录为单位, 所以对于.DBF文件的存取可以当作一般的ASCII文件处理, 只是要设法略去.DBF

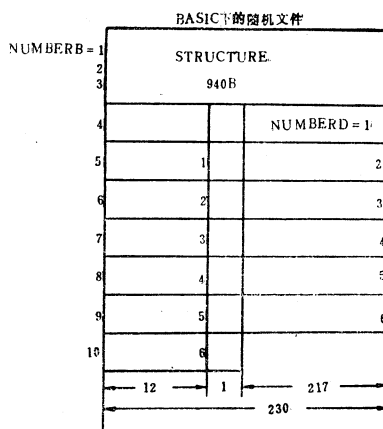


图 5

文件的第一部分。

设L为文件结构长度,W为记录长度, 为了有效地存取记录, 打开文件时应按二者中较小的打开, 并称此文件打开长度为单位的记为作业记录, 相应的序号叫作业记录号(用RN表示)。

令X为要访问的记录号, 来观察L与LN的关系, 得出X号记录来。

1. $L = LN$, 很显然, 作业记录号 $RN = X + 1$, 一次读出, 即是X号记录内容。
2. $L/LN, L \neq LN$, 即L为LN的约数。令 $f = LN/L$, 此时, $RN = (X - 1) \cdot f + 2$, 连续f次, 并依次将读出的数据串连接, 即是X号记录的内容。
3. 当 $L > LN$, 且 $LN \neq L$ 时, $f = [L/LN]$,

$r = L - f \cdot LN$, 可见, 真空记录与作业记录错位 r 个字节, 故 $RN = f + X$, 连续读两次, 第一次读出的数据右截 $LN - r$ 个字节, 第二次读出的数据串左截 r 个字节, 然后连接即是 X 号记录的内容。

4. 若 $L < LN$, 且 $L \nmid LN$, $f = [LN/L]$, $r = LN - f \cdot L$ 。另设 $L_0 = (X - 1) \cdot LN + L$, $K = [1/L]$; L_0 为 X 号记录之前的字节数, 这有四种可能:

(1) 如果 $K \cdot L = L_0$,

此时为左对齐, 即 X 号记录的首字符与某作业记录的首字符为同一字节, 故 $RN = K + 1$, 连续读出次数 $f + 1$, 前 f 次读出的数据串依次连续, 第 $f + 1$ 次读出的数据串左截 r 个字节, 然后进行连接, 所得整串即是 X 号记录的内容。

(2) 令 $L_1 = L_0 + LN$, 若 L/L_1 , 此时右对齐, 即 X 号记录的末字节与某一作业记录的末字节为同一字节, 所以应从 $RN = K + 1$ 开始, 连续读 $f + 1$ 次, 第一次的数据串右截 r 个字节, 然后读出的数据串跟它依次相连, 即是 X 号完整的记录。

(3) 令 $r_1 = L_0 - K \cdot L$, 若 $r_1 < L$, 则 $f + 1$ 个连续的作业记录将一个完整的被读记录全部包含起来。因而, $RN = f + 1$, 连续读 $f + 1$ 次, 第一次读出的数据串右截 $L + r_1$ 个字节, 依次跟后读出的数据串连接, 而末次读出的数据串左截 $r + r_1$ 个字节, 然后连成一个记录。

(4) 不属于上述三种情况时, 则 $RN = K + 1$, 连续读 $f + 2$ 次, 首次读出的数据串右截 $L - r_1$ 个字节, 最后一次读出的数据串左截 $r_1 - (L - r)$ 个字节, 然后, 按顺序连成完整的记录。

七、用BASIC语言程序直接读取.DBF文件的方法

至此, 我们讨论了语言程序直接读取数据库文件(.DBF), 可能出现的全部情况。

算法步骤作出如下整理:

为了在BASIC程序中处理.DBF文件的记录。可先打开文件, 取得记录长度 L , 记录开始字节数 M 及各字段的名称、类型、宽度、小数位等, 然后关闭文件, 再以 L 定义OPEN语句的缓冲区长度重新打开文件, 语句设计如下:

```
270 OPEN "R", #1, F$, L
280 FIELD #1, L AS W$
```

接下去的问题是如何从这个打开的.DBF文件的记录中取出数据记录。由 M 及 L , 容易算出第一个数据的开始位置在第 $K_0 + 1$ 个记录的第 M_0 个字节, 其中 $K_0 = M/L$, $M_0 = M - K_0 \cdot L$, 于是为了读取.DBF文件的第 N 个数据记录, 并送入变量 $R$$, 可由如下语句完成:

```
290 GET #1, Ko+N
300 A$=MID$(W$, Mo)
310 GET #1, Ko+N+1
320 A$=A$+MID$(W$, 1, Mo-1)
```

若需要进一步对记录加工处理, 可根据各字段的类型、宽度、小数点位置从 $R$$ 中取出并作适当的类型转换, 即可加工处理。处理后的第 N 个记录 $R_1$$ 若要送入数据库, 可由如下语句完成:

```
7190 R1$=MID$(D$, 1, L-Mo)
7200 GET #1, Ko+N
7210 LSET W$=MID$(W$, 1, Mo-1)
      +R1$
7220 PUT #1, Ko+N
7230 GET #1, Ko+N+1
7240 R2$=MID$(D$, L-Mo+1, Mo-1)
7250 LSET W$=R2$+MID$(W$, Mo)
7260 PUT #1, Ko+N+1
```

以上的记录只能在原记录的范围内进行修改和读写, 也可在指定的记录位置上写上新的内容, 也可方便地进行删除、插入和追加。但是, 必须注意以下几点:

1. 程序中所读出的记录内容剩余的字节数, 为数据库文件中指定记录的全部内容, 包括第一位的标志位。因此, 若非永久性地

动态菜单界面设计

上海城建学院 陆慰民

摘要 菜单技术是程序设计时常用技术之一,本文介绍动态菜单设计方法,在动态菜单控制下,应用系统就具有自扩展性,用户自编程序可以不经修改原有系统程序而参予运行,我们用dBASE III 论述问题。

一、问题的提出

在各种事务处理应用程序中,当具有若干项供用户选择的功能时,一般都使用交互技术进行分支处理。编程方法是在程序中利用显示控制语句,在屏幕上给出若干行提示信息,用于指示功能号及含义,再用交互命令进行人机对话,选定某一功能后转向执行该模块。此编程技术上的优点是简明方便,缺点是形式呆板,显示的功能有限。如果用户要扩展功能,将自编的某程序功能块加入到现有的应用系统中去,必须修改源程序,这

给程序维护带来许多困难。所以,我们研究了动态菜单界面的设计问题,使应用系统具备动态特性,用户通过编辑块可以随意地增、删、改菜单信息及转移控制,从而用户自编子模块可以作为新功能加入到原应用系统中,避免修改源程序所面临的各种困难。

二、动态菜单界面设计的基本思想

要实现菜单动态化,必须把菜单提示部分的内容从程序中分离出来,将提示信息及转向控制代码构成独立的菜单文件。再将原

~~~~~

删除一个记录需将剩余的字节数的第一个字节改成“\*”号,再将剩余的字节数写入相应的记录即可。

- 2.若要增加或减少记录,即插入、追加或永久性删除,则除了作相应的操作外,还必须修改数据库文件的表头中第5至第8个字节的数据。切记文件结束标志加在最后一记录之后。
- 3.若程序在记录长度超过255个字节时,应先编译后再运行。或在调入BASIC解释程序时键入:

A>BASIC/S:\*\*\*。(注:\*\*\*为实际记录长度的字节。)

### 参 考 文 献

- 1.《IBM个人计算机奥秘》美Peter—Norton著
- 2.《IBM—PC的原理与应用》张福英 蒋新儿 李滨宇 著
- 3.《关系型数据库应用》王秉湖 张喜英编
- 4.《新颖关系数据库管理系统》凌连生 李毓瑞
- 5.《dBASE III与汉字BASIC的信息共享》王大明 刘金玉等。

来程序中用于显示功能提示的语句更换成显示菜单文件内容及处理分支转向的命令行。这样,当菜单文件的内容变动时,所显示的功能也随之而变化。于是新系统就具有动态的特点,有自扩展功能。

为了便于说明问题,我们用dBASE III来说明编程思想。先建立菜单库CDK.DBF,它包含字段CDK NAME和CDK CODE。前者用于存放菜单功能含义说明,后者用于存贮分支转向程序模块名。CDK NAME 的长度应由说明内容而定,一般不要大于显示屏一行可显示的最大字符数80,CDK CODE字段的长度取为8,这是因为在DOS系统下,文件名的有效长度至多为8。

下面给出的程序框架,可以动态地显示菜单库文件 CDK.DBF 中 功能说明及分支转向处理。

```

clea
set talk off
sele 5
use cdk
do while, t,
go bott
pnu=recn ( )
pg=int ( (pnu-1)/5 )
go top
yl=0
do while, t,
clea
@ 1, 20 say —— 功能选择 ——
@ 2, 18 say '—————'
if pnu>4
y3='5'
else
y3=str (pnu, 1)
endi
z=1
do while z<6, and, z<=pnu, and, not, eof
( )
@ z+2, 24 say str (z, 1) +', '+cdkname
skip
z=z+1
endd

```

```

a5=chr (13)
@ z+3, 14 say '按数字键选择, 0返回, 键' 与
< 可换页'get a5
read
do case
case a5=' ',
if yl<pg
yl=yl+1
pnu=pnu-5
endi
go 1+5*y1
case a5=' ',
if yl>0
yl=yl-1
pnu=pnu+5
endi
go 1+5*y1
case a5)'0', and, a5<=y3
go 5*y1+&a5
hk=trim (cdkcode)
exit
case a5='0'
retu
othe
go 5*y1+1
endc
endd
do &hk
sele 5
endd
retu

```

程序第八行中参数5表明每屏可显示功能提示5条。编程时应根据系统所配置的显示器的特性,决定每屏可用于菜单功能提示的行数,对于640×200线的显示器,可选参数为5~8,若程序控制每行显示二条功能提示,则每屏功能说明可多达16项。如果你的微机系统配备的是高分辨显示器,在25行CCDOS管理下,每屏显示参数可定为20。

由于菜单库内一项功能占一条记录,故使用GO BOTTEM语句就能方便地得到功能总条数PNU,再根据显示参数可算出显示完全部功能说明需要的屏数PG。下转(第34页)

# MCS-51单片机小型显示系统

华北电力学院 王攀峰

**摘要** 本文介绍了利用8031单片机控制8行96列发光二极管显示系统的硬件和软件。该系统能显示字符、汉字,具有画面上、下、左、右移动、拉帘、闪动、12/24小时时钟等功能。利用键盘可进行光标做图和敲入数字及英文字母。此外,还能断电保存68幅画面达6个月以上。该显示系统可供车站预告车次、比赛中做计分牌和计时牌。

## 一、硬件的设计

显示屏由  $8 \times 96$  个发光二极管组成，单片机对显示屏进行分时扫描显示。为提高显示系统的抗干扰能力，CPU 以串行方式向显示屏传递信息。考虑到与 PC 机通讯要使用单片机的串行口，我们设计了高速八脉冲发射电路，一般发射 8 位数据共用 4 个机器周期，而单片机的串行口以最快的方式传递 8 位数据也要用 8 个机器周期。下面介绍单片机是怎样向显示屏传递信息的。

首先从显示缓冲区取出一个字节送到累加器A,然后把A的内容送到并入串出移位寄存器74LS165进行锁存,该移位寄存器的串出口与串入并出移位寄存器74LS164相联,然后再启动八脉冲发射器开始工作,大约经过4个机器周期后,8个脉冲发射完毕,A中的内容就送到显示屏的74LS164中,重复12次就发射完一行信息,然后在显示屏上进行显示。下面介绍脉冲发射电路。如图1所示

首先置P1.0为高电平，从而关断脉冲源ALE信号。复位计数器，复位后A<sub>0</sub>~A<sub>3</sub>输出为低电平，同时打开了三态门Ⅱ。接着置P1.0为低电平，脉冲信号经两个三态门输出，当输出8个脉冲后，计数器A<sub>3</sub>为高电平，关断三态门Ⅱ。这样仅发射了8个脉冲。由单片机的时序知，一般每个机器周期有2个ALE脉冲，大约经过4个周期就发射了8个脉冲。比单片机的串行口方式0的发射脉冲的速度快一倍。将三态门Ⅱ的输出端接到74LS165和其它74LS164的时钟端。因此发射8个脉冲后就把74LS165中的8位数传到74LS164中去。

在硬件设计中还扩展了键盘接口、显示屏行选信号以及RS232C接口。其硬件结构如图2所示。

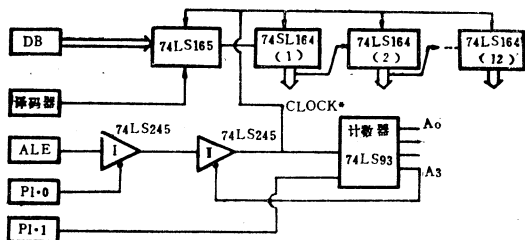


图 1 脉冲发射电路

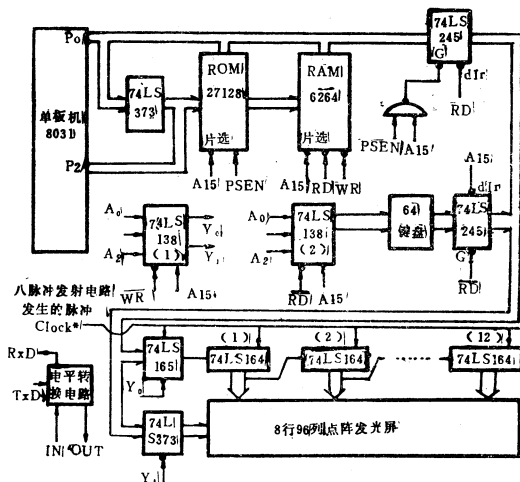


图 2 显示系统的硬件结构图



为了增加数据总线的驱动能力，我们使用了双向数据缓冲器74LS245。从硬件结构图可看出，第一和第二个译码器138 分别译出2个和8个地址。两个译码器用 $A_1$ ，做为控制信号，但由于一个是读操作，另一个是写操作，故不会发生读写错误。各部分地址分配如下：

8K RAM 6264的地址:0000H~1FFFH

写控制的第一片译码器138:8000H~8001H

读控制的第二片译码器138:8000H~8007H

考虑到显示系统要与PC 机进行串行通讯，因而增加了电平转换电路。

## 二、软件介绍

在上述硬件基础上，我们设计了一些子程序，在主程中反复调用这些子程序，就能实现画面上、下、左、右平动、拉帘、显示各种字母和数字等。下面对这些子程序做介绍。

**1. 字符发生器子程序。**给出要发生的字符的ASC II 码，以及该字符在显示屏中的位置(即字符左上角在显示屏中的行、列数)，然后调用字符发生器子程序，子程序根据ASC II 码找到字符字型值编码的首地址，从而把字符送到显示屏的指定位置，并进行显示。

**2. 中断显示子程序。**显示屏由8行96列发光二极管组成，与其对应的显示缓冲区是外部RAM的96个连续地址0000H~005FH。显示缓冲区中的每个字节对应8个发光二极管，从而构成8×96个发光二极管。利用单片机的定时器T<sub>0</sub>中断服务程序自动定时扫描，把显示缓冲区的信息传送到显示屏。由此可见，要想在显示屏上显示某个图案，只需把图案信息送到显示缓冲区即可。T<sub>0</sub>中断服务程序自动把显示缓冲区的信息传送到显示屏上。为了实现闪动效果，在中断显示子程序中安排了闪动标记位来实现这一功能。当该位置零时，定时关闭或开显示屏，从而形成闪动效

果。中断显示程序的框图如图3所示。

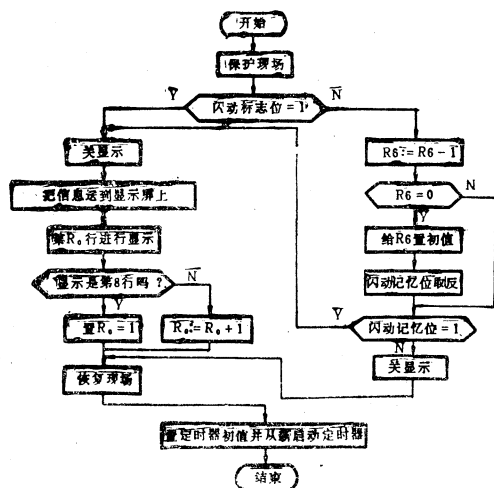


图 3

从显示子程序的框图知，通过对闪动标志位置零或置1 来控制闪动效果，而闪动记忆位只是定时关或开显示。

### 3. 画面移动子程序。

画面移动子程序是指显示缓冲区的信息向某个方向移动。每调用一次画面移动程序，显示缓冲区的信息整体移动一次，然后执行延时程序，再调用画面移动子程序，如此反复，就能在显示屏上看到移动的画面。

**4. 光点子程序。**把显示缓冲区分为8 行96列，给出行列所在位置和是否产生光点标志位，然后调用光点程序，就能在显示屏的某个指定位置产生和删除光点。利用光点子程序可方便地产生不断闪烁的光标。即定时改变光点标志位，就能出现闪动光标。另外，利用光点子程序和键盘可以产生各种图案。即用键盘中的“→”、“←”的“↑”，“↓”四个键移动光标所在位置，按“D”键后，就把光标所在位置的光点删掉，按“↵”键后，就在光标所在位置产生一个光点，从而可方便地实现光标做图功能。

**5. 拉帘子程序。**拉帘子程序有两种运行方式，一种是显示屏上的图案从左到右或从右到左逐列消失，另一种是原显示屏的图案从左到右或从右到左逐列恢复。拉帘速度可通过键盘改变。

**6. 时钟程序。**该子程序能产生年、月、日、时、分、秒。利用键盘可选择12小时/24小时运行方式。定时器T<sub>1</sub>中断服务程序产生的年、月、日、时、分、秒分别放在55H、54H、53H、52H、51H、50H单元中。在需要显示时间的时候,把55H~50H的内容通过字符发生器在显示屏上显示。

**7. 人机对话子程序。**当调用人机对话子程序后,显示屏上出现一个菜单。供用户选择。按数字键1,就进入光标做图模块,此时显示屏上出现闪动的光标。利用键“←”、“↑”、“↓”、“→”能移动光标,按“D”表示删除光标所在位置的光点,按回车键“↵”表示在光标所在位置产生光点。利用这种方式,可画出各种图案。

按数字键2就进入文本编辑模块。用户从键盘上敲入的画面可直接在显示屏上显示。由于键盘有双功能键,按字母的上档键显示屏上出现小写字母。

按数字键3就进入存入、取出、删除图形模块,用户在键盘上敲入的信息可存到外部RAM,在需要的时候取出来或把图形删掉,使用十分方便。

选择数字键4就能修改单片机内部的存储器。通过改变某些存储器的内容,可选择画面的移动方向和速度、闪动和拉帘的速度以及修改时间参数。

选择数字键5便进入运行模块。用户可选择画面移动、拉帘和时间等运行方式,当键入“CTL”后,程序可跳出人机对话子程序,并进入主程序。

最后再谈一谈主程序的设计。在主程序里主要是循环显示一些不同的画面,并基本上把显示屏的功能表演一次。当用户需要人机对话时,按下“CTL”键即可进入人机对话子程序,使用很方便。主程序框图如图4所示。

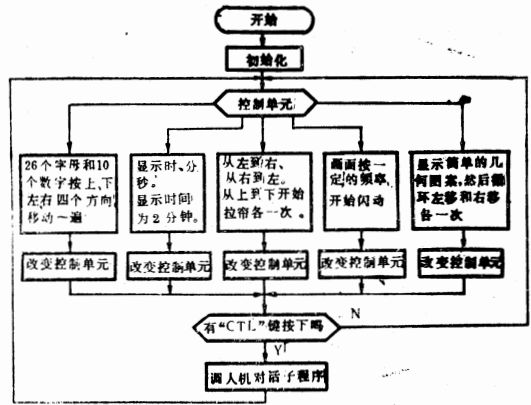


图 4

### 三、抗干扰技术

为了提高系统的抗干扰能力,我们设计了如下的自动复位电路如图5所示。

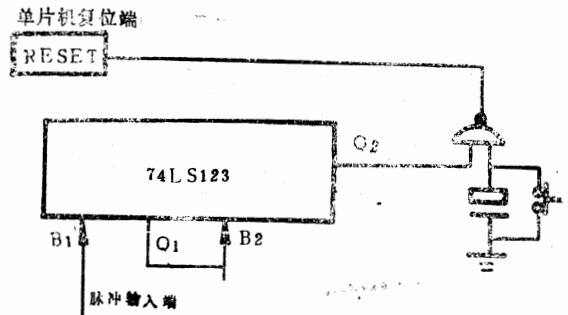


图 5 自动复位电路

当程序在正常工作时,P1.7口不断地向自动复位电路发脉冲,一旦程序出现死循或“飞”掉,P1.7口不再发出脉冲,自动复位电路经过一定时间的延时后,向单片机的复位脚发出一个复位脉冲,从而迫使单片机从头开始运行程序。使用这种方法提高了系统的抗干扰能力。

#### 参 考 文 献

1. 徐爱卿、孙涵芳、盛焕鸣、张继志、《单片微型计算机及其应用》,北京航空学院出版社,1986年。等

# 单片机在雷达自动录取设备中的应用

清华大学 孙宇华

**摘要** 本文介绍了一种军用雷达终端录取设备上所使用的单片机新技术。设计了 PC—PPI 单片机接口模块，介绍了软硬件设计技巧。它用八位单片机代替了过去传统使用的大量硬件，使其具有记忆和处理功能，能灵活而准确地完成上报目标的标志形成。做到了结构简单性能可靠，提高了实战能力。由于它具有可编程性，因此对其它雷达也有较好的实用价值。

## 一、引言

某警戒雷达自动录取设备是一个以微机为中心的系统工程，数据（检测部分用了单片信号处理器做 MTD）处理器使用 2 台 16 位单板机，由一台主控机（PC 机）管理调度，将录取后的目标上报显控台和 PPI，本文介绍 PC 机同 PPI 接口模块的设计，主要用单片机来控制，和整个系统协调一致。下面从软硬件两方面加以说明。

## 二、工作原理

在某雷达自动录取设备中，对重点上报的 12 批目标要在 PPI 上加以亮线标志。目标的预测方位、距离数据从终端处理机随时发来，未加排序。极端情况是在同方位上最多出现 6 批目标。所以，要获取正确的目标标记时刻，须对终端处理机（PC 机）的数据进行记忆和再处理。我们选用单片机做主控处理机，解决目标数据接收、排序及 PPI 亮线产生时刻问题。天线方位码 12 位，而 PPI 上像素较粗，对人眼来说，方位量化间隔取  $\frac{1}{2^8}$  足够。故采用 8 位的单片机 MCS—48 系列的 8039 做主控片。通过八位并口同 PC 机通讯，其框图见图 1。

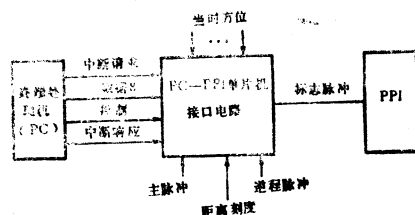


图1

PC 机将预测目标数据、方位、距离码以中断方式传给 8039。8039 对接收的数据按方位大小排队。若同方位有几批目标则按距离从小到大排好。CPU 不断地读取当时方位同目标方位比较。当两者相等时，就把距离码打入外部距离计数器，用距离刻度进行减计数。计为 0 时发生借位信号触发一个标记脉冲形成电路，产生该方位，该距离目标上的标记脉冲（30μs）送给 PPI 辉亮控制。

要解决的主要问题是控制标记辉亮的时间。如果仅在一个主脉冲周期内产生一次目标标记脉冲，其实是看不见的，必须要经纵向延时。如果连续 10 个主脉冲周期都产生标记脉冲，天线转过 14° 才改变方位码，10 个重复周期足可使人在 PPI 上看清亮线。

当同一方位上有 6 批目标（最多）时，一条计数器对一个目标就不可能象单目标时那样重复处理，只有在一个主脉冲周期内串行产生标记方波，并且同上道理也重复 10 次。

所以,要求目标数据排队时使头一个目标码为首,后边是它们之间的距离差。当该方位读到时,先给距离计数器予置目标距离码。然后查询借位脉冲。如查到即目标1结束,立即将距离差1打入计数器,由距离刻度继续减计数,计为0时,触发生成为第2个目标标记脉冲,以此类推,连续生成6个标记脉冲。其中置入差值时仅用2条指令5个微秒。其延时是固定的,可在排队处理时减掉,若粗略些,其延时无大影响(对PPI来说)。

对于最紧张的情况即2个目标几乎贴在一起,2个借位脉冲间隔小于单稳成形的30 $\mu$ s,则不能2次触发单稳态电路。故仍然看见一条亮线。这样2个目标用一个亮线。只要2者之间大于35微秒就可得到2个标记脉冲。总之考虑实际情况,利用软件灵活地去挖掘时间潜力,做到硬件最省而性能最佳。

### 三、PC—PPI单片机

#### 接口模块硬件

PC-PPI 接口电路分为四个部分。① 8039扩展电路;②距离计数器;③计数控制门;④脉冲产生电路。

MCS-48系列单片机是8位的CPU,单一的内部总线,1个8位的定时器/计数器,27根I/O口线。单级中断系统、两个中断源。4~11MHz外部晶振,内部有时钟电路,96条指令等。它集成了一台计算机的基本部件,但外部引线和普通CPU芯片一样多,只有40条。8039内部有128B的RAM,外部需扩展2KB的ROM。我们规定P1口接收来自PC机的12批目标数据,另用扩展I/O的一个8位口传送距离计数器的距离码,另外两个四位口接收天线当时方位码,原理框图如图2所示。

#### 1.8039扩展电路(如图3所示)

在本电路中,程序存储器为2KB,数据存储器为其内部128B。I/O线共31根。其中P1<sub>0</sub>~7为8位准双向口,8039复位后即处于

输入状态,用1NA,P1指令可读取外界即PC机所送数据。8039的P2<sub>4</sub>~7为8位准双向口的高4位。其中P2<sub>7</sub>和P2<sub>6</sub>做控制信号,P2<sub>5</sub>、P2<sub>4</sub>用来传送距离数据高两位。

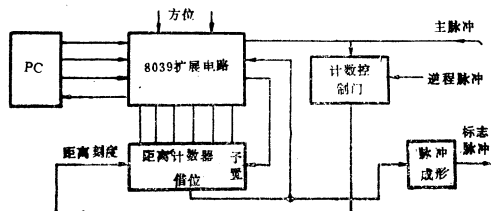


图2 PC—PPI单片机模块框图

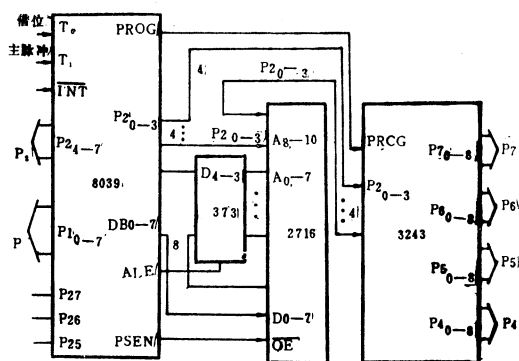


图3 8039扩展电路

8243为8039的专用I/O扩展接口片。它的4个4位口均有双向传送数据能力。P5、P6口用来接受当时方位码,前面已经交待,只取方位码高8位,故令P<sub>5</sub>口读低4位,P6口读高4位。P7和P4口用来传送数据即距离码给距离计数器。此仅8位,另2位由8039的P2<sub>5</sub>、P2<sub>4</sub>线传送,故距离码共10位。

本电路用INT为测试输入的中断申请信号同PC机连接。用T1做定时/计数器的外部计数端,接主脉冲信号,以便利用计数器借位中断产生主脉冲中断申请用T<sub>0</sub>做查询借位脉冲端。

#### 2.距离计数器

距离计数器由3片193构成,本电路用作减计数器,当计数为零时,其借位端输出为低电平,反之为高。其数据予置信号由P2<sub>6</sub>控制。当主程序检测到X距离上应有信号标志生成时,就由8039中的RAM表中取

出该距离码，送至计数器，由P2。发一个负跳信号将码打入计数器。

### 3. 计数控制门和脉冲形成

计数控制门由一个D触发器构成，由主脉冲和逆程脉冲信号使它翻转，形成了距离计数器的时钟（距离刻度）门。

标志脉冲形成主要由单稳电路构成。

## 四、PC—PPI软件

PC—PPI软件有三个功能：接受 PC 机送来的目标数据；对数据排队处理；判断时机向距离计数器予置距离码。以上功能由一个主程序调度。

### 1. 接收数据子程序

PC机通过中断向8039送数，其过程为：

a) 把数据传到8039的P1口后面8039发一中断请求信号。b) 8039将P1口数据读入累加器A，然后向PC机发一负脉冲做为中断响应信号。c) 将数据从累加器A存进数据存贮区。

### 2. 主脉冲中断处理程序

8039内部计数器予置为OFFH，一旦主脉冲到来就计满溢出，产生中断响应，进入主脉冲中断处理程序。天线一周15秒内可发生 $200 \times 15 = 3000$ 个中断。

该程序主要功能是检测当时方位码，给数据排队，适当时给距离计数器予置距离数据。当目标方位数据等于天线当时方位时，就要检查单批还是多批目标。如单批则将距离码打入计数器后返回，如多批则做串标记处理后返回。大量时间（主脉冲中断程序）做数据排队，随时查数据缓冲区有无新数据。排序是先按方位大小后按距离大小（同方位）故要有个冒泡处理。一般情况下 PC 机送数时刻不会落后到当时方位的  $180^\circ$  以后，故不论何时收到数据按方位大小排队，主脉冲中断只管查表输出。按方位排队仍有几个算法问题，暂略。

每一个目标标记都能保证复现10次，程

序流程如图4。

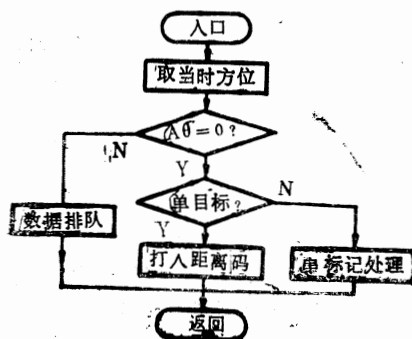


图4

### 3. 主程序

主程序主要是初始化各部分，并循环等中断，初始化时P1口处于输入状态；P2<sub>4~7</sub>处于高电平；内部计数器予置为FFH。置新数据存贮区为空；置处理数据存贮区为空；即置所有存放距离数据单元最高位为1。允许外部和溢出中断。启动内部计数器。以上结束后进入循环等中断。程序流程见图5。



图5

## 五、PC—PPI单片机接口性能

1. 该模块同PC 机的通信时间 $t_m$ 由两部分组成，PC送数时间 $t_{sm}$ 和PC等待时间 $t_{wm}$ 。送数时间由PC 机的程序和速度决定。等待时间由本接口中断处理时间决定。8039采用6MHz晶体，则机器周期为 $25\mu s$ 。则最大等待时间为1.6ms，PC机送数（12批36B）时间为 $t_{sm} = 36 \times 30\mu s = 1.1ms$ 。



# 一个油田微机网络管理 信息系统的设计

西北工业大学

吴先旺 汤星辉

**摘要** 本文介绍了新疆石油局采油二厂的微机网络管理信息系统。该系统采用了二级网络结构,多种通讯方式,实现了从采油队到上层管理科室的数据计算机传送。具有处理业务范围广,涉及层次多,使用方便,效益好等特点。

## 一、绪言

随着油田开采进入中后期,生产和管理难度日趋加大,生产经营任务十分艰巨。为使油田稳产高产,提高经济效益,我校计算机系与新疆石油局采油二厂联合,针对采油

二厂企业特点,和原手工生产管理系统情况,研制了“新疆石油局采油二厂微机网络管理信息系统”。该系统采用了二级网络拓扑结构,多种通讯方法,先进的管理方法和技术手段,促进了采油二厂各业务管理科室之间,科室与基层单位之间的有效信息交流,解决了原手工系统数据处理速度慢、效

则总通信时间为 $t_m = t_{sm} + t_{wm} = 27ms$

2.PPI在同一方位上能显示的目标数N

当方位码取8位时,一个方位量化单位为

$$\theta_1 = 360^\circ / 2^8 = 1.4^\circ$$

天线扫描周期为15s,则天线扫过一个方位量化间隔时间为

$$T_1 = 15s / 2^8 = 58.5ms$$

雷达主脉冲周期为 $T_1 = 4ms \sim 5ms$ 。按 $T_r = 5ms$ 算,则天线扫过一个方位量化间隔可出现的主脉冲周期数N:

$$N = \frac{T_1}{T_r} - 1 = \frac{58.5}{5} - 1 = 10$$

由于PPI显示器本身的局限,一般回波眉毛在余辉作用下宽不小于 $10\mu s$ ,  $30\mu s$ 长的标记亮线是可以看清的,即使两目标连在一起用一根亮线贯穿似略嫌短,但实际也是允许的。另外录取器发的方位码为回波中心予测值、如有误差可能使标记亮线偏在眉毛一边甚至不相交,打在附近也承认录取上。这是约好的。一般来说都会相交。最坏是机动时丢失目标(录取不上自然也打不上标记)尤其在近距离。因天线转速慢,数据率太低。

但警戒雷达对付远程目标是它长处。

在6MHz的晶振下8039的时钟为 $0.5\mu s$ 。一个主脉冲周期的处理或排序、或打标记都不超过2ms,故时间相当充裕。

主控机(PC机)送数通讯的中断和主脉冲中断是异步出现,因为8039是单级中断系统,故二者遭遇必居其一。先来居上。如果打标记的某一主脉冲周期被PC机通讯中断遮盖,仅少一次闪亮(至少十次重复)并不影响亮线的亮度。反之主脉冲中断遮盖了通讯中断,最多挂2ms。排序时被通讯遮盖一次,根本排序无影响。主脉冲中断频率3000次/天线周,而PC通讯中断最多12次/天线周,它们之间的遮盖对系统是允许的。若雷达数据率提高,8039还可加速;最高晶振源可达11MHz。故硬件可不改动。因此适应性很强。

由于该模块使用单片机,使用简单结构实现复杂的功能,结构简单符合实战要求。可靠性高。由于它有可编程性,可随现场情况而改动,故有通用性,利于标准化生产,符合现代化军备的要求。

率低,信息反馈不及时,重复严重等问题,并且使各部门之间的信息得到综合利用。

## 二、系统目标和主要设计思想

在系统的研制过程中,我们主要考虑到如下目标。

1.将原手工系统中需要人工处理的信息进行存贮、分类、检索、统计、分析和对比,生成各类管理用报表。充分发挥管理信息系统的信息加工、处理作用,提高管理工作质量及效率。

2.以生产系统为骨干,按信息的合理流程,理顺各类报表关系,推动机构的合理调整与改革,逐步由计算中心向信息中心过渡,形成一个全厂的信息中心。

3.通过管理信息系统的建立,理顺信息渠道,精简中间报表,建立各种标准化,以提高生产管理、科研管理、经营管理的现代化。

4.利用系统建立存贮的大量基础原始数据,逐步建立各种预决策支持系统,以增强企业的宏观控制能力,提高经济效益。

为了达到这些目标,在研制系统时,我们根据系统工程的开发原则,使用了自顶向下和自底向上相结合的设计路线,分层次分模块的结构化设计方法,同时强调各种文档、表格、资料整理的标准化。在把原系统的物理模型转化为新系统的物理模型时,分步进行。先把原系统的物理模型转化为新系统的逻辑模型,可以对原物理模型进行逻辑抽象,然后再把新系统的逻辑模型转化为物理模型,在此过程中,对新的物理系统进行扩充和完善。

## 三、系统设计与实现

### 1.系统结构

本系统使用二级网络拓扑结构,即一个二级总线网,把全厂4个管理站共9个子系统连为一体。

系统的第一级网络,采用3+以太网,用一根1000米的粗电缆,把厂14个科室以总线形式连成一个局部网,共享服务器资源。由于该网商品化程度高,采用CSMA/CD存取方式,传输速率达10Mb/s,便于以后扩充站点,软硬件升级。

但3+以太网系统成本较高,不可能把网延伸到全厂60多个二级基层单位,而这些单位中的采油队是全厂生产管理的主要原始数据来源。于是,在3+以太网的基础上,我们设计了面向采油队的第二级网络系统,即采油队联机系统。

采油队联机系统是面向终端的主从式总线型网络。该网的总线可以是双扭线或专用电话线,我们使用中华学习机作为该网的数据终端,网络的主机是3+以太网上的PC工作站,以查询(POLLING)方式工作,总线传输速率达4800b/s,基本上满足了从采油队收集数据的要求。

3+以太网和采油队联机系统这两个总线网,通过PC/XT巧妙地互相沟通。因PC/XT本身是连在3+以太网上的一个结点,又是联机系统的主机,与各采油队数据终端相连,起到了“门径”的作用。

第二级网络即采油队联机系统,总线长2—4公里,可以挂接多至25台数据终端。

系统网络结构如图1所示。

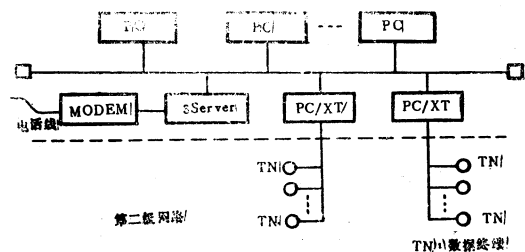


图1 采油二厂微机网络管理信息系统网络结构示意图

系统的第一级网即3+以太网,连接的工作站负责对厂9个管理子系统的业务处理和子系统间或子系统内部的通讯,分属于4个管理站,它们是:

**生产管理站:**有领导查询,生产管理子系统(包括抽油、采油、注水、修井、地下调度和计量等科室的业务),计划、机动、基建等子系统。

**经营管理站:**有领导查询,财务、人事劳资等子系统。

**油田开发管理站,**有领导查询,油田开发(包括静态、动态和油田开发指标技术预决策)等子系统。

**远程管理站:**使网上各工作站通过MODEM与远离网络中心的供应子系统,局有关单位和局内厂、处级单位进行通讯。

## 2. 通讯方式

厂内的各种信息流动,不仅存在于厂内科室内部和科室之间,也存在于厂、局有关单位之间,而全局各厂处二级单位应用计算机的水平不一致。为了满足各业务处的不同要求和厂内各业务系统的要求,在本系统中设计了四种通讯方式。

A.通过3+以太网在厂内联网各科室之间进行通讯。

B.通过采油队联机系统,采油队与相应的生产管理科室之间交换信息。

C.利用3+以太网提供的远程自动拨号服务,借助于现有电话,与厂供应科和局系统通讯。

D.最后一种是脱机方式的通讯。按照局定规范,以跑盘的方式,把厂内某些业务系统的有关数据拷贝在软磁盘上,送局机动处、地质处等单位。

## 3. 数据库、共享数据库的建立以及共享数据安全保密的设计

数据库的设计是系统设计的中心任务之一。系统数据库的设计运用了数据库的规范化理论。数据库的组织不仅考虑了其性质,使用频度,还考虑其简洁、清晰、操作处理

方便,冗余度低。本系统中,数据库分为系统控制数据库、子系统专用数据库和共享数据库三类。

在本系统中,以Dbase III Plus 为主语言,辅助以Basic和Pascal 语言对数据库进行建立、计算和处理。

共享数据库是由一个网络用户创建、多个网络用户使用。其设计根据数据需求方提出要求,供方统一考虑后完成。共享数据库的结构设计和数据更新由供方统一安排,这样可以避免共享数据的冗余,提供整个共享结构的安全性及可维护性。

数据的共享是通过3+以太网服务器来实现的。网络管理员在服务器上为实现所有网络用户能共享数据,建立一个共享用户Dbdata,在该共享用户下为每个网络用户分别开辟一个共享目录区和一定的保密字,共享数据的供方把产生的共享数据放在服务器上其相应的共享目录下,需方则在通过保密字检查后,访问供方共享目录区中的数据。

系统的共享关系示意图如图2所示。

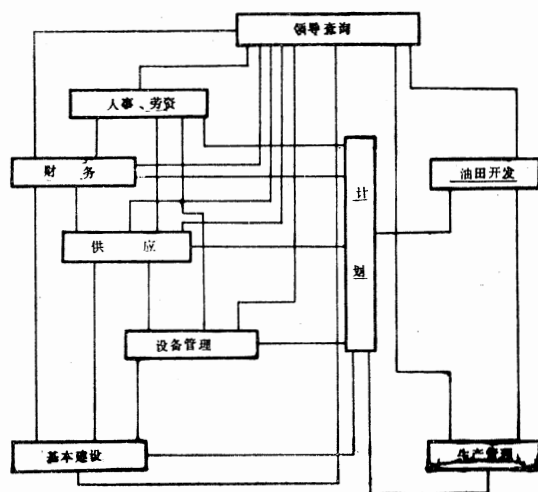


图2 采油二厂微机网络信息系统共享关系图

为了保护共享数据不受破坏,对于每个共享目录都赋予一定的访问权限,只允许数据的供方对自己目录下的数据具有创建、修改、更新数据的权力,而数据的需方只有权使用数据。

某些共享数据并非所有网络用户都可以访问，需采取保密措施，以防非授权用户偷看数据或破坏数据。本系统对DOS系统、3+以太网系统，Dbase III Plus系统提供的文件、数据保护功能，进行灵活的组合应用，实现了共享数据的安全性保护。如，利用DOS命令给文件赋予一定的属性；网络Dbase III Plus命令对数据库或数据库记录加锁；3+以太网系统进行通行字检查等。

#### 4. 系统功能设计

系统功能模块，包括生产管理等子系统。

(1) 网络系统为3+网上的子系统用户提供如下功能。

1) 硬件资源共享：联入3+以太网的各工作主机，可共享网络服务器3 Server上的硬件资源，如磁盘、磁带、打印机、绘图仪等。

2) 软件资源共享：网上的所有工作站主机都可共享服务器上系统用户下的系统软件、工具软件、管理软件和应用软件等。

3) 电子邮件服务：使用本系统提供的电子邮件系统、各工作站用户之间可以透明地进行信件传递，编辑、转发、归档、打印信件，还可以传送各种信件。

4) 脱机打印：各工作站可以把自己的打印任务，交给连在服务器上的CASIO液晶打印机脱机打印。

5) 远程拨号自动服务：远程工作站可以通过计算机键盘输入电话号码，由系统自动实现拨号服务，通过电话系统建立联接，为远离网络中心的工作站用户入网提供了便利。

6) 采油队联机系统：实现厂二级单位一采油队到网络的油水井原始日数据的发送。

(2) 各管理子系统为用户提供的功能如下：

##### 1) 生产管理子系统

该子系统包括地下调度、采油、注水、

修井、抽油等科室业务的处理。主要实现油田生产原始数据的采集、处理和网内传递。对日常生产动态进行分析、比较、实现生产(采油、抽油、注水、修井)的实时管理。建立车井生产台帐，供管理者查询、检索。对重点井进行措施跟踪和旬度水平分析，为油田的生产提供决策依据。

生产子系统工作站主机，作为采油队联机系统的主机与第二级网络的总线相连，负责接收来自十九个采油队的日生产数据，并进行分析，处理和存贮。

##### 2) 油田开发子系统

该子系统建立油水井井史库，对单井的数据进行非正常井分类，为单井的开发提供决策数据，完成油田的综合分析，提供油藏动态监测、岩心分析、油田开发等报表资料，并绘制有关曲线。

##### 3) 计划子系统

该系统完成全厂年、季、月、旬生产经营计划的制定，编制各种计划报表。另外，通过3+以太网及时了解和监督生产、经营管理等部门的计划执行情况，加强计划管理和统计监督，形成各类统计数据库；还对历史指标归类存贮，建立指标历史库，为计划决策提供历史数据。

##### 4) 财务管理子系统

完成对各类凭证的登、销、核对，自动生成各种明细帐、总帐。完成固定资产核算及工资计算，生成成本计划执行情况表，提供各类帐目、凭证的查询，制作各种会计报表。

##### 5) 人事劳资管理子系统

建立职工基础档案库，自动提供统计报表和全厂范围的采油效率定员情况，建立双定台帐，生产定员定额报表，提供职工档案和考工情况的查询、检索功能，实现职工人事管理及调配。

##### 6) 基建子系统

实现全厂管网流程的编辑和绘制，建立单井及计量站级管网流程档案，提供现场管

# 一种新颖的大规模报警系统

长春地质学院 方沛力

## 一、引言

目前我国公安和消防领域中所使用的报警系统，仍然普遍采用的是多线制式。这种类型的报警系统如果控制 $N$ 个点位的探测器，主机最少需要 $2N+1$ 根外接导线。这已无法适应目前兴起的高层建筑。以及防盗防火综合性区域报警的要求。因对这类设施至少需要有上百个点位的探测器，如果用多线制式主机来控制的话，是不可行的。因此如何提高主机控制点位的能力，尤其大幅度压缩主机的外接导线，是目前国内报警器厂家急待解决的问题。笔者通过几年的摸索，采

用地址编码的方式有效地解决了这个问题。地址编码或报警系统的主要特点就是在每个点位上分别配置一个地址编码电路，主机通过微机软件实现地址解码。使主机与点位之间只要几条共用线即可同时控制多达240个点位的探测器。经过在实践中的应用，证明该方法实用可靠。

## 二、编码电路的组成

编码式报警系统方框图如图1所示。主机采用MCS-51单片机，每个点位由编码电路和探测器所组成。编码电路如图2所示，

理单井及计量站段管网标准数据。

### 7) 机动力系统

对厂内各种机动车辆、施工机械、抽油机、泵、锅炉和机床等的运行、修保实施管理，建立历史台帐，及时提出设备的保养大修计划。

### 8) 供应子系统

实现材料、油料的需求、库存和帐目管理。

### 9) 领导查询子系统

该子系统软件可以在3+以太网上的任一工作站上运行。厂生产、经营领导通过该系统，可以查询到厂油田开发现状，生产情况、主要经济技术指标、设备状况，资金状况、人员状况、物资储存及利用等情况，而且还具有图形、表格、屏幕等多样输出形

式。

## 四、结束语

本系统使用了先进的技术和方法，业务覆盖面大，涉及层次多，信息量大，使用灵活方便等特点。在系统开发过程中，本着子系统先开发先使用的原则，各子系统相继投入使用，效果良好，整个系统的工作效率是原手工系统的3倍左右，各子系统及单项应用提供效率在5~100倍之间不等，增加年产值约270万元。更重要的是该系统的开发应用，为深化企业经营管理体制改革和发展油藏科学研究，开发企业决策支持系统打下了坚实的基础。



它是由两个计数器(CC4516)与一个四二输入端与非门(CC4011)等元件组成。CC4516是可预置4位二进制可逆计数器,PE为预置端由此决定了计数器的初始状态相同于 $DP_1 \sim DP_4$ 的电平值。CP为计数器计数脉冲输入端, $\overline{CO}$ 为进位端。两个计数器构成一个1/240的分频器。 $K_1 \sim K_8$ 为编码开关,在系统总体设计时,由此人为决定了该点位的地址。具体确定方法:地址编码开关 $K_1 \sim K_8$ 的权数分别为1,2,4,8,16,32,64,128,地址号计算方法:

地址号 = 240 - 开关置高电平权数之和  
或: 地址号 = 开关置低电平权数之和 - 15,  
(地址号  $\geq 1$ ) 由此,编码地址可以在1~240之间随意选定。

### 三、主机编码过程

报警系统主机外围设备很少,主要以软件为主。详细程序流程见图3,主机交替的工作在两种状态下:巡检各点位探测器,是否有灾情报警;巡检各点位探测器及线路,是否有故障。下面结合程序流程分析报警系统的工作过程。

#### 1. 主机巡检灾情报警

主机在开始巡检前,首先发预寻址脉冲到PE端,使所有编码电路按本身地址预置初值。随之主机向编码电路CP端发出寻址脉冲,各编码电路IC2的 $\overline{CO}$ 端都有一次输出低电平,发生在置位后的本地地址数个脉冲的时刻,如果某探测器遇到灾情,将使开关K变为高电平(平时低电平)从而使当寻址到该分机时, $G_2$ 输出低电平到I线,主机收到后根据发出的寻址脉冲数,可得到报警的点位号,随即通过声光方式报警。报警后继续进入巡检状态。

#### 2. 主机巡检线路及探测器是否存在故障

主机在每巡检一遍灾情报警之后,开始巡检系统中各部位是否工作正常,巡检方式是按点位号由小到大。巡检哪个点位,就在

置位脉冲后连续按该点位号数发固定的寻址脉冲,随即再次向PE端发置位脉冲,被检查点位的编码电路IC2的 $\overline{CO}$ 端为负,并经 $G_1$ 倒相为正之后与相继到达的PE正脉冲同时加到 $G_4$ 的输入端,使其输出为低电平,加到探测器的检测端,如果探测器工作正常将使I线得到低电平。主机收到I线低电平后,开始检查下一个探测器,否则声光故障报警。

编码电路中的 $G_3D_2$ 的作用是提高了地址分辨率。各编码电路中的 $D_1$ 构成一个N输入的与门。实践证明主机的寻址脉冲频率为10KHz,效果很好。

### 四、三级网报警系统的组成

区域性综合报警,是指把若干座楼房各类灾情报警的统一管理。这将涉及到成千上万个点位的探测器,实现这样的控制系统,可以采用集中,区域,点位三级网络系统。如图4所示。集中和区域控制器均采用MC S—51单片机,之间以串行通讯的方式联系。区域控制器即为前面所介绍的报警系统主机,工作方式也基本与前文所述相同,一旦出现灾情或故障,除自身报警外,马上报告集中控制器。集中控制器平时轮番巡检各区域控制器是否工作正常,如发现某区域控制器工作异常,将马上报警。当得到区域控制器的灾情或故障报警信息后,也将马上报警,并显示出具体地点,每台集中控制器管理区域控制器的最大容量为256台。每台区域控制器可控制探测器点位的最大容量为240个。由此构成的网络系统足够一般用户的需要。

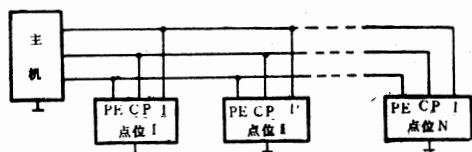


图1 编码式报警系统方框图

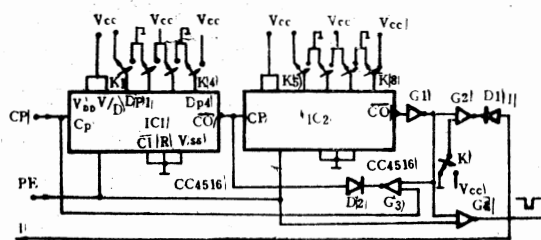


图2 编码电路

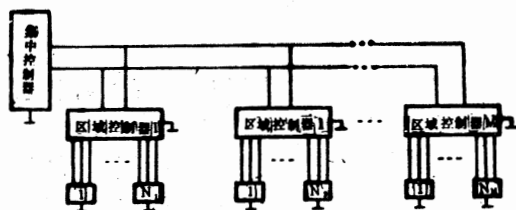


图4 三级网报警系统方框图

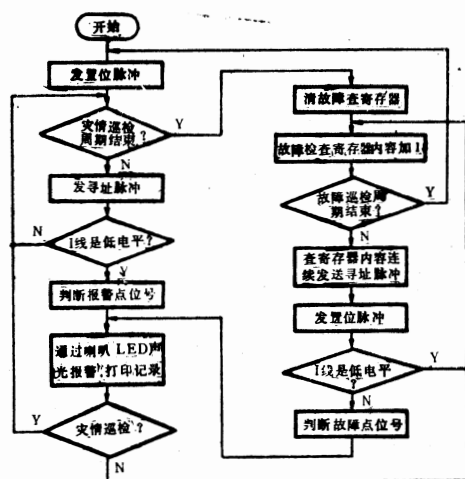


图3 主机程序流程图

上接（第21页）

多重分支DO CASE...ENDCASE语句用于菜单前后页换页控制、功能选定及容错处理。当选定某一有效功能后，通过定位命令把指针移到所需要的记录上，于是程序流就转入CDKCODE字段内所对应的子程序块中。

从上述程序可以看出，当CDKCODE字段内的程序名变更后，程序分支转向也跟着改变，当CDK.DBF增加一条记录后，系统就相应地增加了一个功能，所以，按此思想设计的菜单界面程序就具备了动态功能，它能方便地连接任何新功能模块。

关于CDK.DBF内容的维护，只需一小段简单的编辑修改程序就能办到，为节省篇幅，不再赘述。

### 三、优化处理

一般说来，一个应用系统中会有多个菜

单界面，若每一个界面都对应一个菜单文件，在程序执行时，文件的打开与关闭就非常频繁，执行效率不高，显然它不是一种好方案。我们可以将每一级菜单编上一个序号，在菜单库内增加一个字段，用于指示记录内所对应功能所属的菜单层次的级别。对菜单库CDK.DBF按序号这个关键字进行排序，当进入某级菜单时，先用搜索命令查出该级菜单在菜单库内的首尾地址，显示控制与分支转向就在该范围内进行。

为进一步提高执行速度，还可以在上述基础上再建立一个链表指针文件，该文件功能就是指示各级菜单在菜单库CDK.DBF内的位置及功能条数。如果要显示某一层次的菜单时，先从链表文件中直接读出有关的控制参数，然后在CDK.DBF文件内直接定位，这样可省去查找搜索所需花费的时间，提高执行效率。

# Intel 80860微处理器的体系结构

总参56所 林钟官

**摘要** Intel公司今年3月推出了不同于86系列的微处理器 80860。它采用  $1.0\mu\text{m}$  工艺的 CHMOS 制造技术,在单芯片上集成了 100 万支晶体管,使过去需要多块芯片才能实现的功能在单芯片上实现了。80860 采用了科学技术计算和图象处理的体系结构,具有速度快和处理三维图象功能等特点。芯片尺寸为  $10\times 15\text{mm}^2$ , 有33MHz与40MHz两种型号。

## 一、引言

80860 微处理器是以特定应用领域为设计目标,力求区别于 80386等86系列机。它采用了面向将有广阔市场前景的科学技术计算领域和图象处理的体系结构,构成32位 RISC 型CPU核心部件、采用流水线结构的64位浮点运算部件、三维图象处理部件和存储管理部件,以及配置了 8K字节的数据高速缓存和4K字节的指令高速缓存。

80860 微处理器在高速化方面采取了许多措施。如采用并行处理结构和总线带宽与运算部件处理性能充分匹配等技术措施,使 80860 的处理能力比32位微处理器提高了若干倍。其主要性能见表 1。

## 二、功能描述

80860微处理器的结构框图如图1所示,它由下列 9 个部件组成:

- 核心执行部件(整数运算部件)
- 浮点控制部件
- 浮点加法器部件
- 浮点乘法器部件
- 图象处理部件
- 分页部件
- 指令高速缓存

- 数据高速缓存
- 总线 and 高速缓存控制部件

表1. 80860的主要性能

|                 |                                  |
|-----------------|----------------------------------|
| 峰值速度 (40MHz)    |                                  |
| 整数运算部件          | 40MIPS                           |
| 浮点运算部件          | 80MFLOPS (单精度)<br>40MFLOPS (双精度) |
| 三维图象处理性能 (象素/秒) |                                  |
| 8位              | 3300万                            |
| 16位             | 1700万                            |
| 32位             | 120万                             |
| 数据高速缓存容量        | 8K字节                             |
| 指令高速缓存容量        | 4K字节                             |
| 片上存储管理部件 (MMU)  |                                  |
| 实地址             | 4G字节                             |
| 虚拟地址            | 4G字节                             |
| 虚拟存储方式          | 合页式 (4KB/页)                      |
| 封装              | 168管脚PGA                         |
| 功耗 (40MHz时)     | ~ 3.25W                          |

核心执行部件控制80860的总操作,它执行装入、存储、整数、位和控制转移操作以及为浮点部件取指令,为整数运算提供一组  $32\times 32$  位通用寄存器。装入和存储指令将数据送入这些寄存器或从中取出,其全部的整数、逻辑和控制转移指令为核心执行部件提供执行系统软件 and 应用程序的能力。陷阱

机制对异常和外部中断提供快速应答。通过有关数据或指令的访问陷阱功能支持调试。

浮点硬件连接到各组浮点寄存器，寄存器可以作为 $16 \times 64$ 位寄存器或 $32 \times 32$ 位寄存器存取。专用的装入和存储指令还可以按 $8 \times 128$ 位寄存器存取这些寄存器。所有的浮点指令使用这些寄存器寄存它们的源和目的操作数。

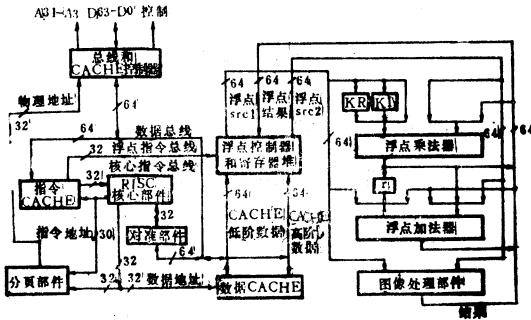


图1 80860微处理器结构框图

浮点控制部件控制浮点加法器、乘法器、发送指令，处理所有的源和结果异常以及修改浮点状态寄存器中的状态位。加法器和乘法器可并行操作，每个时钟产生2个结果。浮点数据类型、浮点指令和异常处理都支持IEEE二进制浮点运算标准。

浮点加法器执行64和32位浮点值的加法、减法、比较和转换。执行一条加法器指令时间为3~4个时钟，但以流水线方式工作时，每个时钟产生一个新的结果。

浮点乘法器执行64和32位浮点值的浮点和整数乘法以及倒数操作。执行一条乘法器指令时间为3~4个时钟，以流水线方式工作时，每个时钟可以产生一个单精度的新结果，每隔一个时钟产生一个倍精度的新结果。

图象处理部件在使用16位像素和16位Z缓存进行静面消除和成荫处理时，每秒可以处理17万个像素。它以三维物体为主要处理

对象。处理像素为组合式像素，像素字长可任选为8/16/32位。每个像素被分割为2到3个信息组，如果这8个信息组分别被冠以红、绿、兰三基色，则可进行彩色处理。

分页部件经一个64条目4路成组相联的存储器（TLB——转换后备缓冲器）实现保护的分页虚拟存储器。分页部件使用TLB将逻辑地址转换成物理地址并检查违法访问。访问保护形式使用两级特权：用户和管理程序。

指令高速缓存为4K字节的2路成组相联存储器。主频为33.3MHz时，传送速度为266M字节/秒。

数据高速缓存为8K字节的2路成组相联存储器，主频为33.3MHz时，传送速度为533M字节/秒。

总线和高速缓存控制器为核心部件实现数据和指令存取，它从核心部件接收周期请求和技术要求，执行数据高速缓存或指令高速缓存未命中处理，控制TLB转换和提供到外部总线的接口，其流水线结构支持3个未完成的总线周期。

### 三、体系结构

80860微处理器程序员可见的体系结构包括数据类型、寄存器、指令和陷阱等。

#### 1. 数据类型

80860提供整数和浮点数据操作。整数操作以32位操作数实现，有时也支持64位操作数。装入和存储指令可以访问8、16、32、64和128位操作数。浮点操作是以IEEE标准的32位和64位格式实现。图象处理指令以8、16或32位像素的数组操作。

##### （1）整数

8位和16位整数的算术操作可以通过符号扩展成32位来实现。

##### （2）序数

算术操作可供32位序数用，序数为不带符号的整数。加法和减法指令也可以按64位序数操作。

### (3) 单精度和倍精度实数

单精度实数为32位二进制浮点数，倍精度实数为64位二进制浮点数，它们均支持ANSI/IEEE标准754。

### (4) 像素

80860处理的像素为组合式像素，根据彩色和亮度分辨的要求，其字长可任选，以有效地处理彩色亮度图象。像素数据可由下面两类指令使用。

- 帮助实现隐面消除选择的像素存储指令。
- 帮助实现三维彩色亮度成荫像素的加法指令。

## 2. 寄存器

80860有下面4类共72个寄存器，见图2所示。

### (1) 整数寄存器

整数寄存器有32个32位寄存器，它由算术操作和装入与存储指令访问，用于地址计算和标量整数计算。读出时r0总是复原为0状态。

### (2) 浮点寄存器

浮点寄存器有32个32位寄存器，它同样由算术操作和装入与存储指令访问，用于浮点计算。读出时寄存器f0—f1总是复原为0。浮点寄存器还供一组整数运算用。

在存取64位浮点或整数值时，使用对准的一组8个寄存器。指令必须指明该组包含64位或128位值最低的寄存器编号，非对准的寄存器编号产生未定义的结果。

128位装入和存储指令以及在浮点寄存器和数据高速缓存之间的128位数据通路(见图2)有助于支持超高速计算。

### (3) 6个控制寄存器

控制寄存器只能由装入和存储控制寄存器指令访问。

- 处理器状态寄存器 (psr) — 它包含现行进程各种状态的信息。
- 扩展处理器状态寄存器 (epsr) — 它包含超出psr, 存储范围现行进程的附加状态信息。
- 数据断点寄存器 (db) — 它在80860中使数据操作数访问存储在这个寄存器中的地址时产生陷阱。这种陷阱由psr中的BR(数点读出)位和BW(断点写入)位赋能。在进行比较时, 根据操作数的大小忽略该地址的一些低阶位。保证与该寄存器中所含的地址重叠的任何存取将产生陷阱。
- 目录基地址寄存器 (dirbase) — 它用于控制地址转换、进行高速缓存和总线选择。
- 故障指令寄存器 (fir) — 出现陷阱时它包含陷阱指令的地址。
- 浮点状态寄存器 (fsr) — 它包含现行进程的浮点陷阱和舍入方式状态。

### (4) 4个专用寄存器

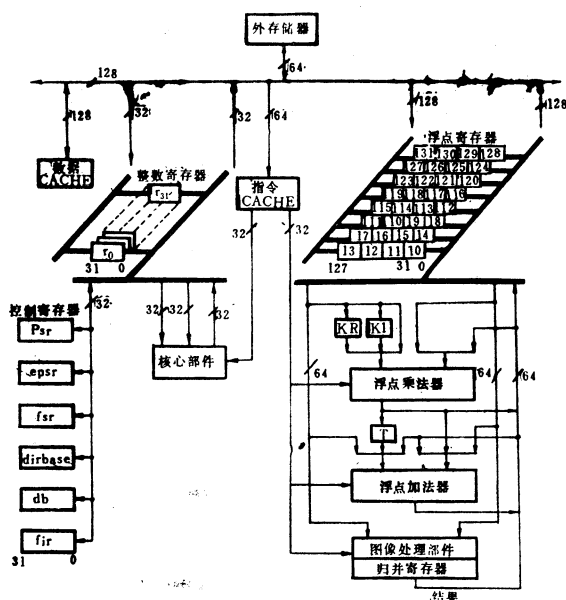


图2 寄存器和数据通路



专用寄存器KR、KI、T和MERGE供一些专用指令使用。KR、KI和T寄存器供双重操作浮点指令使用、起动加法和乘法器操作。KR、KI和T寄存器可以存储来自一条双重操作指令的值并将这些值作为随后双重操作指令的输入。

MERGE寄存器仅供向量整数指令累加重加操作的结果，累加后的结果存入操作寄存器。

### 3. 指令系统

80860微处理器采用了RISC结构的CPU核心部件，指令系统较为简单，仅用了82种指令：整数运算指令42种，浮点运算指令34种，汇编伪操作指令6种。

80860体系结构的突出特点是采用并行操作提高其运算速度，但其并行性是不透明的。

指令系统的特点：

#### (1) 流水线和标量操作

80860中的浮点部件除采用流水线方式执行指令外，还可按标量方式执行，其大多数浮点指令都含有流水线和标量方式操作的变种。

#### (2) 双重指令方式

双重指令方式，就是可以并行执行浮点运算指令和整数运算指令。此时指令由64位组成，低阶32位为浮点运算指令；高阶32位为整数运算指令。

#### (3) 双重操作指令（对偶运算指令）

为适应图象处理和科学计算领域中经常要用的积和运算，80860配备了将加法、减法和乘法组合执行的指令，即双重操作指令。它包括加、减法运算后再执行乘法运算的指令；乘法运算后再执行加、减法运算的指令。

### 4. 寻址方式

数据存取限于装入和存储指令，存储器地址由装入和存储指令的src1和src2两个字段计算。

(1) src1包含32位整数寄存器的识别符或者包含一个立即16位地址偏移值。

(2) src2始终指定一个寄存器。

由于src1或src2可能为空（零），所以各种有效寻址方式的结果为：

偏移值+寄存器：可用于存取记录内的字段，寄存器指向该记录的起点。可用于存取栈帧中的各项，寄存器用于指向栈帧的起点。

寄存器+寄存器：可用于栈帧内二维数组或数组存取。

寄存器：用作任何地址计算的结果。

偏移值：进入第一个64K逻辑地址空间的绝对地址。

另外，浮点装入和存储指令可选择自动增加寻址方式。这种方式中，在执行装入和存储以后，src2可用src1和src2的和代替，使通过数组的步进更有效。

### 5. 中断和陷阱

80860将陷阱类型分为6种。即指令故障、浮点故障、指令存取故障、数据存取故障、中断、复位。

### 6. 调试

80860微处理器用数据和指令断点支持调试，其特点包括：

(1) 数据断点寄存器(db)支持80860将要监控的数据地址的技术要求。

(2) 处理器状态寄存器(psr)的BR(断点读出)和BW(断点写入)位使读出或写入分别陷阱到db中的地址。

(3) psr的数据存取陷阱(DAT)位允许陷阱处理程序确定数据断点何时产生陷阱。

(4) 陷阱指令可以用来设置任意个数的代码断点，src1和src2字段的值促进识别已出现的那个断点。

(5) psr的指令陷阱(IT)位允许陷阱处理程序确定陷阱指令何时产生陷阱。

## 四、体系结构的 高速化措施

为了满足高速进行科学技术计算和三维图象处理的要求,80860采取了一些高速化的措施。诸如引用了巨型计算机的体系结构和确保与运算部件、处理部件相匹配的总线宽度等。

1. 浮点运算部件采用流水线并行操作  
流水线分级处理是巨型计算机体系结构常用的一种技术,当流水线达到高效率时,指令的实际执行时间为流水线一级运算的时间,所以它比非流水线处理性能更好。

80860采用了4条流水线:乘法器、加法器,向量整数运算器和浮点装入。浮点加法器、乘法器分别由3级流水线构成。浮点加法器每个时钟周期都可以输出单精度或倍精度运算结果。此外,运算结果向外部存储器传送可与浮点运算部件的处理并行执行。该数据传送仍由整数运算部件执行。

2. 配备了对偶运算指令和双重指令方式

80860配备了对偶运算指令和双重指令方式,而且对数据通路进行了改进,将加法器的结果输出端与乘法器的输入端相连,同时乘法器的结果输出端与加法器的输入端相连。这样,运算结果无需都要借助浮点运算寄存器就可以完成传送。此外,乘法器的输入端还配置了KI和KR两个常数专用寄存器,以便于完成带系数的乘法运算。

3. 片上配置了2个高速缓存和TLB

为了充分发挥运算部件的高速性,80860内藏8K字节的数据高速缓存、4K字节的指令高速缓存和64条目的TLB(转换后备缓存),以减少对存储器的平均访问次数和时间。此外,数据高速缓存的写入采用了回写方式。因此,每当对缓存写入时,无需一一写入片外存储器。

片内配置64条目的TLB也有利于减少外部访问。此外,由于采用了4路相连的缓存方案命中率得以提高,减少了对外部存储器中表的检索次数。

4. 提高总线宽度

80860提高了片内总线的宽度,以保证向运算部件供数的能力。在数据缓存与浮点寄存器之间采用128位宽的总线连接,其最大数据传送速度可达每秒640M字节。消除了运算部件在数据备齐之前必须等待的瓶颈问题。

同样,从指令高速缓存中引出了两条32位总线,分别与整数运算部件和浮点运算部件连接,使得它们可以并行处理。

80860适当提高了外部数据总线的宽度,即备有一条64位的外部数据总线连接到标准的DRAM,使得在发生未命中时能够高效地更新高速缓存的内容。

5. 采用流水线化的数据写入方式

CPU核心部件用于执行整数运算、进行系统控制及装入/存储操作。数据装入命令中包括对写入的数据进行缓存处理的命令和将数据缓存转入旁路的命令。其中后者称为流水线装入命令,该命令可将数据写入过程流水线化。由于采用了这种流水线装入命令,可在不更新数据高速缓存的情况下,从外部写入数据。

该命令在执行多元数组乘法运算时十分有效,尤其在执行 $40 \times 40$ 以上的较大数组运算时更加有效。

## 参 考 文 献

1. Intel i 860 Microprocessor, Intel Corporation, Feb. 1989.
2. Intel's ISSCC Bombshell, A Supercomputer on A Chip, Electronics/March 1989, pp. 24—30. 等

## TMS320C25单片16/32位数字信号处理机

**摘要** 本文介绍美国新近推出的TMS320C25单片16/32位数字信号处理机的芯片。它是一种经过优化的高度集成的数字信号单片处理机。其内部结构为32位，每秒可执行500万条指令。这样的功能水平已完全可以实现音频范围内的实时信号处理。

### 一、总体结构

TMS320C25的机器周期时间为100ns，有544字的片内RAM及4K字片内掩膜ROM，单周期乘法——累加指令，32位算术逻辑单元及累加器，16个I/O通道，片内定时器；其指令系统支持浮点操作及适应滤波，支持2进制FFT。TMS320C25的原理框图示于图1。

TMS320C25是一种Harvard结构的数字信号处理机。它是围绕两条主要总线构成的：程序总线 and 数据总线。程序总线传送来自程序存储器的指令码和立即操作数。数据总线将内部单元联到数据RAM，例如中央算术逻辑单元（CALU）、辅助寄存器组等就是通过数据总线接到数据RAM的。程序总线 and 数据总线把来自片内数据RAM和内外程序存储器的数据在一个周期内送到乘法器以进行乘法——累加操作。TMS320C25有一个高级的平行结构。例如当CALU正在处理数据时ARAU（辅助寄存器算术单元）可以同时运算。这种平行结构保证了功能极强的一组算术操作、逻辑操作及位控制操作能够在一个机器周期内完成。

TMS320C25为68脚PLCC封装。表1

出了它的信号、信号状态及其功能。

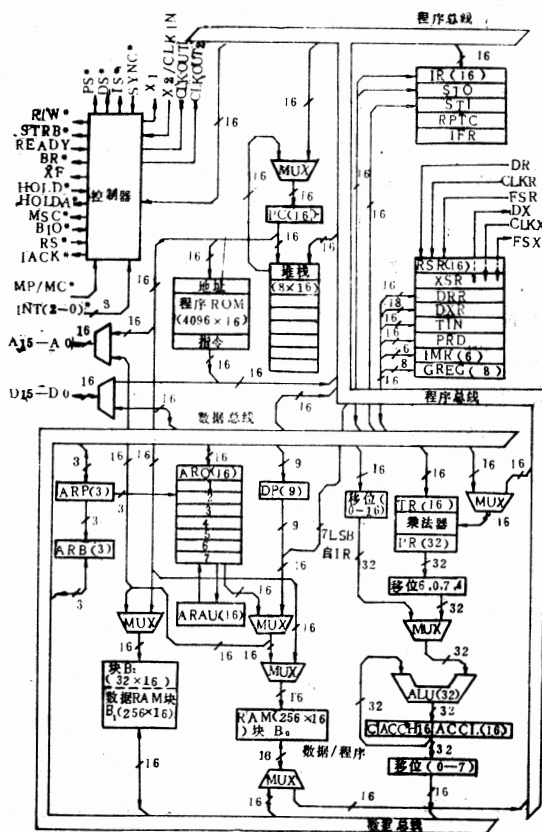


图1 TMS320C25框图

表1. TMS320C25信号说明

| 信号              | I/O/Z | 说 明                                              |
|-----------------|-------|--------------------------------------------------|
| V <sub>cc</sub> | I     | 5V电源                                             |
| V <sub>ss</sub> | I     | 接地                                               |
| X1              | O     | 内部振荡器输出, 晶体用                                     |
| X2/CLKIN        | I     | 对内部振荡器的输入, 来自晶体或外时钟                              |
| CLKOUT1         | O     | 主时钟输出信号(四分频晶体或CLKIN)                             |
| CLKOUT2         | O     | 次时钟输出信号                                          |
| D15—D0          | I/O/Z | 16位数据总线, D15 (MSB) 到D0 (LSB), 程序, 数据和 I/O空间均分路传送 |
| A15—A0          | O/Z   | 16位地址总线, A15 (MSB) 到A0 (LSB)                     |
| PS*, DS*, IS*   | O/Z   | 程序, 数据和I/O空间的选择信号                                |
| R/W*            | O/Z   | 读/写信号                                            |
| STRB*           | O/Z   | 选通信号                                             |
| RS*             | I     | 复位信号                                             |
| INT2*—INT0*     | I     | 外部用户的中断输入                                        |
| MP/MC*          | I     | 微处理器/微计算机方式选择                                    |
| MSC*            | O     | 微状态完成信号                                          |
| IACK            | O     | 中断应答信号                                           |
| READY           | I     | 数据准备输入, 由外部逻辑置位, 当用慢速设备时, 它表示当前总线传输已经完成          |
| BR*             | O     | 总线请求信号, 当TMS320C25需要对外部全局数据存储器空间进行访问时被置位         |
| XF              | O     | 外部标志输出                                           |
| HOLD*           | I     | 保持输入; 置位时, TMS320C25 进入空闲方式并将数据线, 地址线及控制线均置成高阻状态 |
| HOLDA*          | O     | 保持应答信号                                           |
| BIO*            | I     | 分支控制输入, 由BIOZ指令转态                                |
| DR              | I     | 串行数据接收输入                                         |

|      |       |                           |
|------|-------|---------------------------|
| CLKR | I     | 串行口接收输入用的时钟信号             |
| FSR  | I     | 接收输入用的帧同步脉冲               |
| DX   | O/Z   | 串行数据传送输出                  |
| FSX  | I/O/Z | 传输用的帧同步脉冲, 既可用于输入, 也可用于输出 |
| CLKX | I     | 串行口传送用的时钟信号               |

TMS320C25 数字信号处理机实现了单个累加器的Harvard结构, 程序存储器和数据存储器分配在不同的地址空间。这样, 取指令和执行指令可同时进行。在外部, 程序和数据存储器空间在同一总线上分路以使这两个空间的地址范围最大并使元件的引脚最少; 在内部, TMS320C25 拥有程序总线 and 数据总线两个独立的总线, 可使程序和数据全速执行, 继而得到最大的处理能力。

独立的流水线用来进行指令译码。取指令——指令译码——执行指令流水线对用户来说基本是不可见的。TMS320C25 大部分指令都在一个机器周期内完成, 不管用片内RAM或ROM程序存储器还是用外部程序存储器。利用READY信号可以同慢速的片外存储器及外围设备进行通讯。但这时指令就变成多周期的了。

## 二、存储器控制

TMS320C25提供了共544(16位)字的片内数据RAM。数据RAM分成独立的三块(B0, B1, B2), 其中, B0, 256字, 既可做数据存储器, 也可做程序存储器。具体做什么存储器, 由CNFD或CNFP指令来决定。利用RPTK及BLKP指令可将在廉价慢速的外部程序存储器中的指令装入B0, 然后用CNFP指令就可将B0变成程序存储器。用这种方法, 原程序存储中的时间苛刻的小程序块就可廉价地存储起来并全速执行。B1

和B2共288字，总是做为数据存储器。共544字的数据存储器可让TMS320C25 处理512字的数据，其余32个单元用于中间存贮。TMS320C25有一个4096字的片内ROM。用户程序可在工厂掩膜在ROM里。用微处理机/微计算机选择引脚MP/MC\*，TMS320C25可将ROM 分配到存储器空间以内或以外。这样，设计带TMS320C25的产品就节省了很多时间。由于片内已经有了4K的ROM，无需重新设计印刷电路板，所以产品的成本也大大下降。TMS320C25提供了直接编址64K字的片外存储器空间。在这个空间里，高速存储器程序可全速执行，慢速存储器程序需插入等待状态。

TMS 320C25 对程序存储器、数据存储器及I/O 分别提供了独立的地址空间。除了B0、B1及B2 以外，数据存储器的分配还包括分配给存储器的六个外围设备寄存器和备用单元。六个外围设备寄存器是二个串行口寄存器（DRR及DXR）、一个定时寄存器（TIM）、一个周期寄存器（PRD7）、一个中断屏蔽寄存器（IMR） 和一个全局存储器的分配寄存器（GREG）。备用单元不准存贮，读其内容是无意义的。

TMS320C25有一个寄存器组，它包括八个辅助寄存器（AR0—AR7）。它们可用来对数据存储器进行间接寻址，或做为暂时存贮。这些寄存器可用一条指令来直接寻址，也可用3位的辅助寄存器指示器（ARP）来间接寻址。这些寄存器和ARP 都可以由数据存储器装入，也可由指令中的立即操作数装入。这些寄存器的内容也可以存入数据存储器。这些辅助寄存器与辅助寄存器的算术单元（ARAU）相联，如图2所示。

当数据存储器单元寻址时，ARAU 可对当前辅助寄存器进行自动变址。变址可以加1，减1，或加减AR0 的内容。因此，对信息表的访问无需CALU进行地址控制。ARAU还可做为附加的通用算术单元使用，这是因为辅助寄存器可与数据存储器直接通

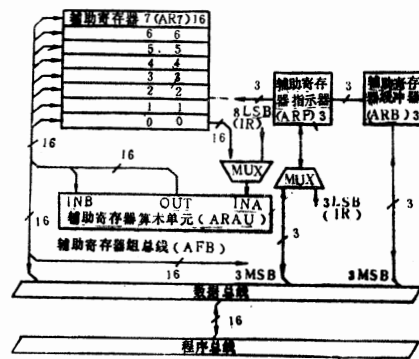


图2 辅助寄存器组

讯。ARAU 可进行10位无符号算术运算，所以 CALU 可完成32位 2 的补码的算术运算。TMS320C25还具有几种取决于ARP指定的辅助寄存器与AR0的比较的分支指令。

TMS320C25 有一个16位的程序计数器（PC）和一个用来存放 PC 的八级硬堆栈。堆栈用于中断或子程序。程序计数器通常总是访问程序存储器，不管程序存储器是片内的还是片外的。累加器的内容装入 PC 可完成计算转移操作。为处理子程序嵌套和八级中断，TMS320C25 具有压入和弹出指令，可在数据存储器里建立堆栈。

本地存储器接口由一个16位并行数据总线（D15~D0），一个 16 位程序地址总线（A15~A0），数据存储器，程序存储器或I/O空间的三只选择引脚（PS\*、DS\*及IS\*）和各种系统控制信号组成。R/W\* 信号控制数据传输的方向，STRB\* 提供控制传输的定时信号。

利用HOLD\*和 HOLDA\*信号，TMS320C25可支持其对外部程序存储器或数据存储器的DMA操作。HOLD\* 置低电平时，其它处理机可完全控制外部存储器。这时，TMS320C25 把地址线、数据线及控制线全部置成高阻状态。利用中断、外部处理机可与TMS320C25进行通讯。这里有两种通讯方式。一种，象TMS32020一样，在 HOLD\* 置位期间当前的执行暂时挂起。另一种，并行DMA 新方式，TMS320C25 继续执行程

序，其操作来自内部 RAM 或 ROM。对于数据量很大的场合，后一种方式可大大提高处理能力。

### 三、中央算术逻辑单元(CALU)

中央算术逻辑单元由一个16位的定标移位器、一个16×16位的平行乘法器，一个32位的算术逻辑单元（ALU）、一个32位的累加器（ACC）和一些累加器和乘法器输出的附加定标信号所组成。图3是CALU的方框图。典型的ALU指令的执行步骤是：（1）在数据总线上取数据（数据来自RAM），（2）数据通过定标移位器并在ALU中完成算术运算，（3）计算结果送到累加器。对ALU的输入总要经过累加器，其它输入可来自乘法器乘积寄存器（PR）或定标移位器。定标移位器由数据存储器装入。定标移位器有一个与数据总线相联的16位输入端和一个与ALU联接的32位输出端。定标移位器按指令将输入数据左移0—16位。移位时，输出数据的低位用0来补充，最高位可用0来补充也可用符号来扩展，视状态寄存器（ST1）中的符号扩展方式位的状态来决定。附加移位可使处理机进行数值定标和位的选取，从而扩展算术运算和防止溢出。32位的ALU和累加器可完成各种算术指令和逻辑指令，其中绝大多数指令可在一个时钟周期内完成。溢出方式由SOVM和ROVM（置位/复位溢出方式）指令编程决定。当累加器处于溢出状态，而且确实出现了溢出，则溢出标志被置位同时根据溢出方向累加器装入最大正数或最大负数。为便于数据存入数据存储器，32位的累加器分成两段：ACCH（累加器高段）和ACCL（累加器低段），每段16位。累加器高端的附加移位器进行由0—7位的左移。左移是在数据送到数据总线进行存贮时完成的。这时，累加器的内容保持不变。为了将累加器的内容进行移位，累加器内部也能左移或右移一位（SFL或SFR指令）并可将进位循环（ROL或

ROR指令）。TMS320C25设有进位位是为了完成多精度算术运算。加法和减法都产生进位。ADDC（带进位的加法）和SUBB（带借位减法）两条新指令在计算结果时都要用进位位。

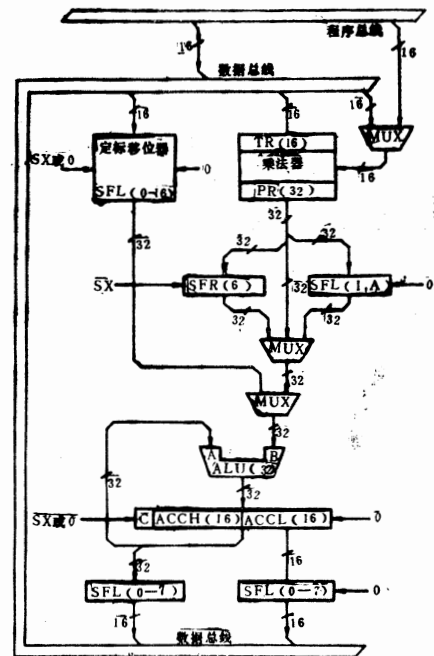


图3 中央算术逻辑单元

TMS320C25 用一个2的补码16×16位硬件乘法器在一个机器周期内计算出32位的乘积。有两个寄存器与乘法器相联，一个是16位的暂存寄存器（TR），它保存乘法器的一个操作数；另一个是32位的乘积寄存器（PR），它保存乘积。乘积寄存器的输出可左移1位或4位，主要用于分数的算术运算及验证分数乘积。乘积寄存器的输出还可以右移6位，从而可执行128次连续乘法——累加操作而不致溢出。无符号乘法指令（M-PYU）是一种扩展精度的乘法。TR中的无符号数与数据存储器寻址单元中的无符号数相乘，乘积放在PR中。MAC和MACD两条乘法——累加指令充分地利用乘法器的计算宽度，两个操作数可同时处理。对于MAC和MACD，两个操作数在每个周期里通过程序总线 and 数据总线送到乘法器。利用重复指



令 (RPT和RPTK) 可将上述操作变成单周期的乘法——累加。程序总线总是通过内部和外部ROM和RAM提供数据且总是单周期操作。平方加 (SQRA) 和平方减 (SQRS) 指令把同一个值送给乘法器的两个输入端。

为适应动态范围大的计算需要, TMS320C25支持浮点运算。规格化 (NORM) 指令用左移的办法使累加器中的定点数进行规格化。按TR移位装入累加器 (LACT) 指令通过输入定标移位器将规格化的浮点数变成定点数。ADDT和SUBT指令按TR指定的移位进行附加的算术操作, 可对16位尾数4位指数的浮点数进行运算。

TMS320C25有几条根据ALU状态转移的分支指令。还有位测试 (BIT和BITT) 指令。

## 四、系统控制

为便于系统控制, TMS320C25 设有一个片内定时器、一个重复计数器、三个用户外部可屏蔽中断和由串行口操作或定时器产生的内部中断。存贮器映象16位定时器 (TIM) 由被内部时钟连续钟控的逆计数器组成。定时器一旦减到零就出现定时器中断 (TINT)。在定时器到零后的下一个周期里, 周期 (PRD) 寄存器的内容重新装入定时器, 所以中断是可编程的, 而且中断的出现是以规则的  $(PRD + 1) \times CLKOUT$  1个周期为间隔。这对于控制操作、同步采样或写外围设备非常有用。TMS320C25设有重复指令, 它可使一条指令重复最多达256次。重复计数器 (RPTC) 在RPT指令时由数据存贮器的内容决定, 在RPTK指令时由立即数值决定。重复操作常用于乘法——累加指令、数据块移动指令, I/O传输指令以及表的读/写指令等。正常时是多周期的指令, 当使用重复指令时就成了流水作业了, 并且变成了高效率的单周期指令。譬如, 表读 (TBLR) 指令通常是三个周期, 当采用重复

指令时, 每一个周期都可读一个表单元。为适应外部设备对处理机进行中断的需要, TMS320C25设置了三个外部可屏蔽用户中断 (INT2\*—INT0\*)。内部中断由串行口、定时器或软件中断指令产生。中断是有优先级的。复位 (RESET) 的中断优先级最高, 串行口传送中断的优先级最低。ST0 和ST1两个状态寄存器存放TMS320C25的条件及方式。这些状态寄存器的内容可用指令存到数据存贮器或由数据存贮器装入。这些操作可在中断期间或子程序调用期间完成。

## 五、I/O接口

TMS320C25 能满足系统接口的广泛要求。三个独立的地址空间 (程序、数据和I/O) 提供了对存贮器及 I/O 的接口。其 I/O 处理与存贮器处理完全相同。处理机利用外部地址和数据总线将 I/O 设备分配到 I/O地址空间。存贮器与各种速度的I/O设备之间的联系是利用READY线完成的。片内串行口用来与串行装置及串行系统进行直接通讯。两只串行口寄存器 (数据传送/接收移位寄存器) 可以8位 (字节) 或16位 (字) 两种方式工作。帧传送同步脉冲可在内部产生也可在外部产生。串行口的最大速度为5 MHz。串行口设两个缓冲区, 可同时进行接收和传送, CLKR/CLKX 的最小频率为零, 设有帧同步方式位 (FSM), 无需帧同步脉冲即可连续操作。

## 六、指令系统

TMS320C25 指令系统可支持大量信号处理和一般应用, 例如多机处理和高速控制等。其目的码完全与TMS32020兼容, 所以TMS32020 程序不必修改就可在 TMS320C25上运行。为获得最大处理能力, 当前指令正在执行时就提前取下一条指令。

该指令系统有三种寻址方式: 直接寻

址、间接寻址和立即寻址。直接寻址和间接寻址可用来访问数据存储器。直接寻址时，7位的指令字与9位的数据存储器页指示器（DP）的内容联在一起构成16位的数据存储器地址。用128字的页长，DP寄存器指向数据存储器512页中的一页，从而得到64K的数据存储器空间。指令中的7位地址指向数据存储器页内的特定单元。直接寻址可用于除CALL、分支指令、立即操作数指令和无操作数指令以外的全部指令。八个辅助寄存器AR0—AR7提供了灵活的间接寻址方式。指令中所用的数据地址放在这八个辅助寄存器中的一个里。辅助寄存器指示器（ARP）里分别装以1, 2, 3, 4, 5, 6及7来对应AR0—AR7，由此可选择指定的辅助寄存器。

间接寻址有七种型式：增加或减少的变址、加上或减去AR0的内容的变址、加上或减去AR0内容且求反进位扩展的变址以及无变址（见表2）。全部变址操作均以原指令相同的周期在当前辅助寄存器里完成，但要给ARP一个新的值。位求反变址寻址方式对于2进制FFT程序中的再顺数据点可

高效地完成I/O运算。

表2 寻址方式

| 寻址方式            | 操 作             |
|-----------------|-----------------|
| OPA             | 直接寻址            |
| OP*(, NARP)     | 间接寻址, AR不变      |
| OP*+(, NARP)    | 间接寻址, 当前AR增加    |
| OP*-(, NARP)    | 间接寻址, 当前AR减少    |
| OP*O+(, NARP)   | 间接寻址, 当前AR加上AR0 |
| OP*O-(, NARP)   | 间接寻址, 当前AR减去AR0 |
| OP*BRO+(, NARP) | 间接寻址, 当前AR加上AR0 |
| OP*BRO-(, NARP) | 间接寻址, 当前AR减去AR0 |

注 表中NARP场可选择，用来指定ARP的新值。

立即寻址方式，其指令字中包括立即操作数。TMS320C25有单字（8位及13位常数）立即短指令和双字（16位常数）的立即长指令。在立即长指令中指令操作码后面的字就是立即操作数。TMS320C25指令系统有20条立即操作数指令。

哈尔滨电站设备成套设计研究所

钱学文 编译

## 小 消 息

Intel公司正在研究与高性能80×86兼容的新一代32位处理机80586系列。其研究目标是80586的速度将比80486快三到四倍，可能过40Mips。为此，Intel将在减少每条指令的时钟周期上下功夫。平均计算80386，每条指令需要5个周期，80486约需2个周期，而80586将减少到1个周期。预计5年后PC的速度将可达到每秒钟1亿条指令。

# IBM PC/XT系统板维修技术

华东工学院 陈文高

**摘要** 本文扼要介绍IBM PC/XT系统板维修技术,并提供该系统板上三片只读存储器和二片延时线的替换技术。

IBM PC/XT个人计算机在我国其数量不少,相应其维修量也就大一些。系统板是IBM PC/XT个人计算机的核心部件,在系统中起着很重要的作用。本文介绍本人维修IBM PC/XT系统板的一些做法。

## 一、系统板故障诊断技术

IBM PC/XT个人计算机系统板是由8088CPU子系统、ROM子系统、RAM子系统和I/O通道所组成,支持CPU工作的器件有:DMA控制器8237A—5提供4个DMA通道,中断控制器8259A提供8级中断源,定时器/计数器8253—5提供3个定时器/计数器通道,外围接口芯片8255A—5提供3个8位的并行接口,I/O通道由8个62线的插槽J<sub>1</sub>—J<sub>8</sub>组成,板上还有一个键盘接口电路和一个扬声器接口电路,CPU工作主频为4.77 MHz。

系统板的故障诊断,通常用开机自检(BIOS)和高级诊断程序诊断检查。

当系统板发生故障时,故障现象一般有以下几种:

- a. 开机无显示,系统死锁;
- b. 显示“101”错误信息;
- c. 显示“301”错误信息;
- d. 显示RAM错信息;
- e. 开机显示“PARITY CHECK 1”

错误信息。

要确定系统板发生故障的具体部位,主要应从检查以下三个部位的信息入手:

- a. 检查I/O插槽的各种信号是否正常?

在这里要特别注意数据总线(A2—A9)

和地址总线(A12—A31)的信号是否正常?

- b. 检查8088 CPU各引脚的信号是否正常?

在这里除了注意检查从8284定时器送来的CLK<sub>88</sub>、RESET和READY三个信号是否正常外,还要注意检查状态信号S<sub>0</sub>、S<sub>1</sub>和S<sub>2</sub>,这三个状态信号是表示总线周期状态的,从这里我们便可知道当前CPU正在执行何种操作。

总线周期状态标志如下:

| S <sub>2</sub> | S <sub>1</sub> | S <sub>0</sub> | 操作类型   |
|----------------|----------------|----------------|--------|
| 0              | 0              | 0              | 中断响应   |
| 0              | 0              | 1              | 读I/O口  |
| 0              | 1              | 0              | 写I/O口  |
| 0              | 1              | 1              | 暂停     |
| 1              | 0              | 0              | 取指令操作码 |
| 1              | 0              | 1              | 读存储器   |
| 1              | 1              | 0              | 写存储器   |
| 1              | 1              | 1              | 无操作    |

- c. 检查外围接口芯片8255A—5的PA口的输出电平是否正常?

在系统板加电自检时,CPU通过8255A—5接口芯片的PA口输出部件检测标志。如果系统检测到故障信息便停机,这时我们可通过测量PA口的输出电平,可以确定故障的部位,PA口输出的故障检测标志如下:

| PA <sub>2</sub> | PA <sub>1</sub> | PA <sub>0</sub> |            |
|-----------------|-----------------|-----------------|------------|
| 0               | 0               | 1               | BIOS累加和错   |
| 0               | 1               | 0               | 8253—5错    |
| 0               | 1               | 1               | 8237A—5错   |
| 1               | 0               | 0               | 前16KB RAM错 |
| 1               | 0               | 1               | 8259A错     |
| 1               | 1               | 0               | CRT适配器错    |

其中BIOS累加和错一般是PROM U<sub>18</sub>或U<sub>19</sub>损坏；8253—5错一般是定时器/计数器 8253—5 芯片损坏；前 16 KB RAM错可能是系统板上第一排RAM芯片中某一芯片损坏或是 RAM存储器读写控制电路有故障；8259A错一般是中断控制器8259A芯片损坏。

从上述三个部位的检测中，如发现有不正常的信息，便可从这个不正常的信息入手，进行追踪检查，就有可能查找到故障的具体芯片。如有条件的话，可用已知好的系统板进行对比检查。检测信号时，最好使用示波器。

## 二、系统板特殊芯片替换技术

IBM PC/XT个人计算机系统板上所用的芯片除RAM外绝大多数是74系列的 TTL 集成电路芯片，一般市场上可轻易地买到。但有以下五个芯片比较特殊，一般在市场上难以购到，这常常使维修人员感到头痛，它们是U<sub>18</sub>（PROM），U<sub>19</sub>（PROM），U<sub>44</sub>（PROM），TD1（延时线）和TD2（延时线），往往有许多机器因买不到上述器件，而被报废或闲置。本人通过实践提出以下器件替换方法：

1. U<sub>18</sub>、U<sub>19</sub>是IBM PC/XT 系统板上ROM子系统中一芯片，它是一个具有32KB容量的PROM芯片（如23256），内装固化BASIC解释程序的后24KB和8KB的基本输入输出系统BIOS，占用系统地址 F8000~FFFFFH。当发现U<sub>18</sub>损坏时，可用EPROM27256芯片替换。

EPROM27256写入方法如下：

可使用EPROM写入器（如 MEP256 写入卡）写入，具体操作方法是：可在一台已知好的IBM PC/XT个人计算机上操作，首先将内存地址F800：0~7FFFH中的内容搬入写入器的数据缓冲区，再将缓冲区里的数据写入EPROM27256芯片中。写好的EP-

ROM 27256 芯片便可替换 PROM U<sub>18</sub>使用，硬件电路无需作任何改动，便能正常工作。

2. U<sub>19</sub>、U<sub>19</sub>也是IBM PC/XT个人计算机系统板上 ROM子系统中一芯片，它是一个具有8KB 容量的 PROM 芯片（如MK 37050），内装固化BASIC解释程序的前8KB，占用系统地址F6000~F7FFFH。

当发现U<sub>19</sub>损坏时，可用 EPROM2764 芯片替换。

EPROM2764写入方法如下：

也可用 EPROM写入器写入，方法同U<sub>18</sub>，不同的只是将内存F600：0~1FFFH中的内容搬入写入器的数据缓冲区。硬件电路也无需作任何改动，便能正常工作。

3. U<sub>44</sub> IBM PC/XT系统板上RAM子系统的片选码电路由U<sub>44</sub>（PROM）、U<sub>42</sub>（74LS138）和 U<sub>50</sub>（74LS138）等芯片组成，用来产生读写RAM芯片所需要的行地址选通信号RAS和列地址选通信号 CAS以及控制地址转换的多路选择信号RAMADDRSEL。第一级译码电路 U<sub>44</sub>采用译码只读存储器芯片 TBP 24S10，它是一个三态输出标准可编程只读存储器，内含数据长度为4位的256个字。其内容如表1所示：

（1）当发现TBP24S10损坏时，可用 EPROM2716芯片替换。也可用EPROM写入器将表1中的数据写入到EPROM2716中，写入器数据缓冲区地址0~FFH写入表1中对应的内容，其余地址空间全写入0。

因TBP 24S10和EPROM2716的引脚数不同，故其硬件电路需转插如图1所示：

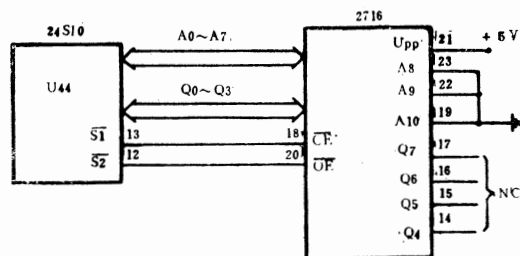


图1

(2) 另一种方法是：当发现U<sub>44</sub>TBP 24S10损坏后，不修理。可将内存扩充板开始地址拔为0，这时用内存扩充板上的RAM存储器作为系统板上的RAM存储器用，这样系统仍可正常工作，只是将系统板上的RAM存储器给闲置了。因为TBP24S10只影响系统板上RAM子系统工作，而不影响其它子系统工作。

#### 4. 延时线TD1

延时线TD1在IBM PC/XT系统板上RAM子系统中用来产生一个60ns和一个100ns时间延时作用。当发现延时线TD1损坏时，可用TTL反相器74LS14或74LS04芯片替换。

#### 5. 延时线TD2

延时线TD2是产生一个7ns的时间延

迟。DMA时钟产生电路如图2所示：

当发现TD2延时线损坏时，可用一个74LS14反相器同时替换U<sub>89</sub>（74LS04）反相器和延时线TD2，电阻R<sub>4</sub>和R<sub>5</sub>可焊掉不用。其替换电路如图3所示。

也可以采用一片74LS14芯片同时替换延时线TD1和TD2。具体做法：用一个14引脚的插座（只留引脚1，4，7，8，10，12，14）取代损坏的TD1，并取一片74LS14芯片按图4连接好各引脚，然后插在TD1插座上，并将引脚4连接到U<sub>74</sub>—1，引脚12连接到U<sub>89</sub>—9，并焊掉TD2、R<sub>4</sub>和R<sub>5</sub>不用。

解决了上述5个芯片替换问题之后，这样IBM PC/XT个人计算机的系统板就完全可以自己修理，再不用发愁买不到器件了。

表1

| 地 址                       | 内 容   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|---------------------------|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0000 09 09 0E 0E 0E 0E 0E | 0E—0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E |
| 0010 09 09 0B 0B 0E 0E 0E | 0E—0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E |
| 0020 09 09 0B 0B 0D 0D 0E | 0E—0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E |
| 0030 09 09 0B 0B 0D 0D 0F | 0F—0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E |
| 0040 0F 0E 0E 0E 0E 0E 0E | 0E—0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E |
| 0050 0F 0E 0E 0E 0E 0E 0E | 0E—0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E |
| 0060 0F 0E 0E 0E 0E 0E 0E | 0E—0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E |
| 0070 0F 0E 0E 0E 0E 0E 0E | 0E—0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E |
| 0080 09 09 09 09 0E 0E 0E | 0E—0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E |
| 0090 09 09 09 09 0B 0B 0B | 0B—0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E |
| 00A0 09 09 09 09 0B 0B 0B | 0B—0D | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E |
| 00B0 09 09 09 09 0B 0B 0B | 0B—0D | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E |
| 00C0 09 0E 0E 0E 0E 0E 0E | 0E—0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E |
| 00D0 09 0B 0E 0E 0E 0E 0E | 0E—0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E |
| 00E0 09 0B 0D 0E 0E 0E 0E | 0E—0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E |
| 00F0 09 0B 0D 0F 0E 0E 0E | 0E—0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E | 0E |

其中：09表示选中第一排动态RAM芯片；

0B表示选中第二排动态RAM芯片；

0D表示选中第三排动态RAM芯片；

0F表示选中第四排动态RAM芯片；

0E表示无效。

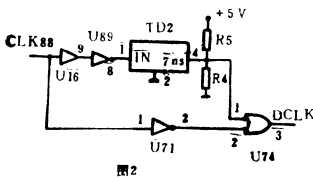


图2

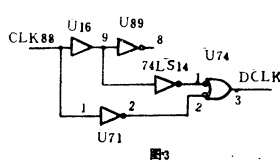


图3

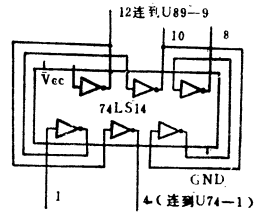


图4

图4