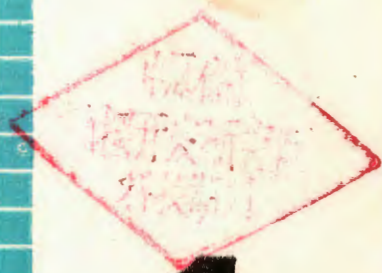


# 微小型计算机 开发与应用



1

1987

微小型计算机开发与应用编辑部

## 《DESKTOP系列计算机资料译丛》出版发行

DESKTOP GENERATION<sup>MT</sup>16位超级微型计算机是美国DG公司1983年7月发表的最新产品,该系列机具备4个机型:DESKTOP 10; 10/SP; 20; 30。具有下面几个显著特点:

1. 双CPU结构,可以和国内用户熟悉的两个系列机兼容,NOVA—ECLIPSE系列,即DJS 1000系列,IBM PC系列,包括PC和PC/XT。

2. 图形处理功能强,配备多种图形输入输出设备和GKS图形处理软件。

3. 通讯,网络软件丰富,可以同IBM大型机,ECLIPSE MV系列及IBM PC等十多种机器通信。

DESKTOP 系列机引入国内后,受到广大计算机用户的欢迎,为了满足国内广大用户需要,我们翻译了此套资料,取名为《DESKTOP系列计算机资料译丛》。

由于DG公司的DESKTOP机随机资料内容非常丰富,为用户使用、维护、开发提供了方便。我们现在发行的第一批软硬件资料48种,主要内容

有:

- (1) DESKTOP计算机性能介绍和使用。
- (2) 操作系统的各种使用手册和实用程序(包括AOS系统, RDOS系统, MS-DOS系统和CP/M—86系统)。
- (3) 各操作系统支持下的多种高级语言。
- (4) 通讯网络软件, INFOS文件管理系统和CEO系统。
- (5) 各种服务性程序用户手册等。

《DESKTOP系列计算机资料译丛》不仅是DESKTOP计算机的系统管理人员,操作人员,程序设计人员和维护人员的必读资料,而且是广大1000系列机用户和PC机用户的重要参考资料,同时也可作为从事计算机工作的技术人员和大专院校有关专业的参考资料。

这套译丛48种现已开始发行(全套价格180元)欢迎来函来人购买,批量购买、价格优惠。

天津电子计算机研究所

《DESKTOP系列计算机资料译丛》编辑部

## 欢迎订阅一九八七年《苹果园》杂志

《苹果园》杂志由Apple微机协会主办,系国内为Apple微机用户服务的专刊。本刊的宗旨是普及微机知识,推广微机应用,作好技术服务与交流,使用维修、信息反馈。《苹果园》主要栏目有:国内新闻、技术专文、应用篇、国外动态、使用与维修、娃娃园地、问与答、读者信息等。主要刊登国内外有Apple机的硬件系统、应用程序、应用系统、监控程序、操作系统、用户要求、使用维修和市场动态等方面文章。读者对象是从事

Apple机设计和应用的工程技术人员,科学研究人员,大专院校和中小学师生,以及有关Apple机用户的管理人员等。

《苹果园》杂志87年双月刊,双月25日出版,全年6期,每套3.00元。

订阅办法:单位请从银行汇至山东潍坊市《苹果园》杂志编辑部,开户银行:市支行东关办事处,帐号:896315。个人从邮局汇款至山东潍坊市东风大街60号《苹果园》邮购组。

编辑:《微小型计算机开发与应用》编辑部

出版:天津市电子计算机研究所

地址:天津市河西区友谊路爱民道5号

发行:天津市邮局

印刷:天津市静一胶印厂

定购处:全国各地邮局

定价:0.60



# 目 录

1987年

第1期 (总第27期)

## 计算机系统

- DGC32位超级小型机新产品——MV/7800..... (1)
- 32位机总线的实际性能比较..... (10)
- 微小型计算机的性能评价与选型.....毛根生等 (19)

## 计算机软件

- 一个软件工程开发估算系统的设计.....何建中 (23)
- ORACLE简介.....董保华 (26)
- 怎样定义好激光打印机与计算机的软件接口.....尹锋 (30)

## 微型计算机应用实例

- OAS/724办公室自动化系统的开发步骤与方法.....徐宝文 (31)
- 化工信息管理系统HUAGONG.....朱容村 张泽斌 (34)
- 在IBM PC/XT屏幕上绘制趋势图和相关图.....刘莉明 (37)
- 计算机服装辅助设计系统的应用研究.....章兵 (38)
- 一种控制器用的专用微机.....汪仁煌 (43)

## 程序介绍

- 通用性较强的数据检索程序 (四).....唐景岩 (45)

## 经验点滴

- 用APPLE II机把Z80汇编文件写入EPROM的一种简便方法.....张富有 马东升 (47)
- 用在Z80微处理机上的8位十六进制数字显示器..... (9)

## 征稿启事

(封三)

# 征 稿 启 事

一、《微小型计算机开发与应用》是经电子部计算机工业管理局批准编辑出版向全国发行的中等技术性刊物。重点介绍国内外微型、小型计算机总体结构, 系统配置, 软件和网络设计以及应用方面的专题, 还辟有“微小型计算机应用实例”, “计算机与网络”, “典型程序介绍”, “经济点滴”, “新机型”等栏目, 选登国内外优秀的应用程序, 介绍软、硬件设计心得和创见。向读者提供国内外计算机发展动态和最新的计算机科技知识, 适合计算机界广大科技人员, 管理人员、教学人员、大专院校学生以及各行各业计算机用户阅读。

欢迎计算机科技人员踊跃投稿。

## 二、征稿内容

1. 学术论文; 2. 科研成果; 3. 应用实例; 4. 经验体会与创见; 5. 专题综述, 动态分析等。

## 三、来稿要求

1. 来稿请一律用钢笔写在单面稿纸上, 文章力求主题突出, 内容充实, 数据可靠, 字迹工整。文字简练, 一般在1—2千字, 长篇技术性文章在5—6千字。

2. 文章前面必须有内容摘要, 插图需按制图要求(图幅最好不大于 $20 \times 14\text{cm}$ ), 插图在文中所占的位置用方框标出, 注明图号及图注。

3. 外文字需要正楷书写, 易混字母旁请用铅笔注加说明。外国人名、地名等除常见并有固定译法的可用中文外, 原则上均用原文。

4. 每一标点符号占一格, 文中需用黑体字时, 请在字下加波文线。

5. 参考文献均写于文章最后, 一般为书名、作者、页数、出版社与出版年月。

6. 来稿请写真实姓名(发表可用笔名), 工作单位, 职务, 详细通讯地址。

四、来稿如经采用, 本编辑部在收到稿件后三个月内通知作者。来稿不采用的除作者事先提出要求外一般不予退回。稿件发表后, 按规定付稿酬。

五、本刊愿就刊物内容和所辟栏目等问题倾听各位专家、作者及广大读者的意见与建议, 如蒙赐教, 不胜感激。

六、来稿请寄: 天津市河西区友谊路爱民道五号、天津电子计算机研究所《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》编辑部

## DGC 32位超级小型机新产品—MV/7800

ECLIPSE MV/7800 系统是美国 DATA GENERAL 公司最新发表的一种 32 位超级小型机新产品。它具有 32 位超级小型机 MV 系列的功能和特性，而且价格低。

MV/7800 系统采用超大规模集成电路技术，在 15 英寸见方的印制板上含有 CPU，FPU（浮点运算部件），远程诊断处理器，日历时钟，可写控存，2 或 4 MB 的主存和支持 DCH 和 BMC 的总线。主存可按 2，4，10MB 的增量增加到 14MB。

MV/7800 可支持 128 个终端/打印机。但对于 40 个用户的部门它有着最优的性能/价格比。而且由于 MV/7800 可以支持 DG 的 AOS/VS、AOS/DVS、AOS/RT32 和 DG/UX 操作系统，用户可以根据自己的应用需要加以选择，AOS/VS 用于分时系统，AOS/DVS 用于分布式系统，AOS/RT32 用于实时响应系统，DG/UX 适用于 UNIX 的用户。系统的构成如图 1 所示。

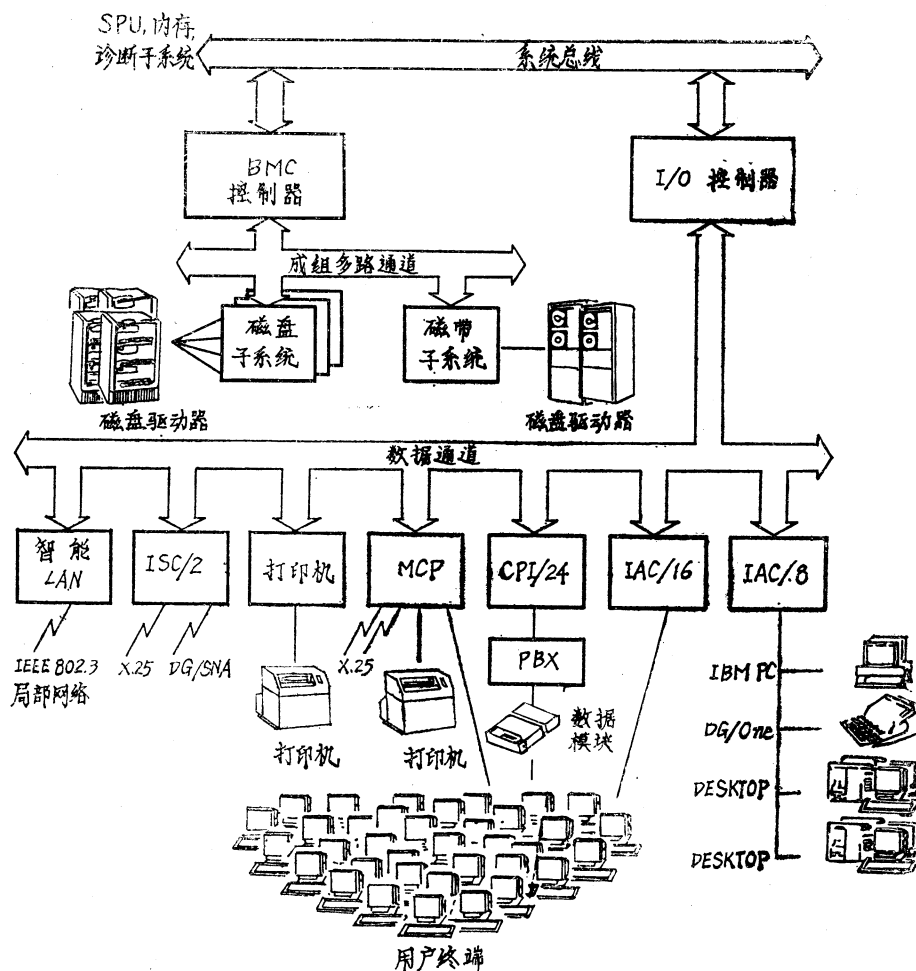


图1 MV/7800系统的构成



# 一、系统结构

MV/7800使用了二个主要的结构：ECLIPSE MV系列的结构和分布式处理器的结构。

MV/7800系统的主要特点如下：

- 宽带的系统总线。
- 分布式处理结构。
- 单块的系统印制板上包括以下的部件：  
ECLIPSE MV系列的体系结构；  
计算功能增强的32位系统处理单元；  
指令流的多级流水线；  
最佳的存贮管理设备；  
增加软件性能的硬件加速器；  
主存系统；  
I/O通道控制。
- 支持全部的通信和网络。
- 可靠性和方便的维护性。

MV/7800系统按其功能可分为以下几个主要的部分，如图2所示。

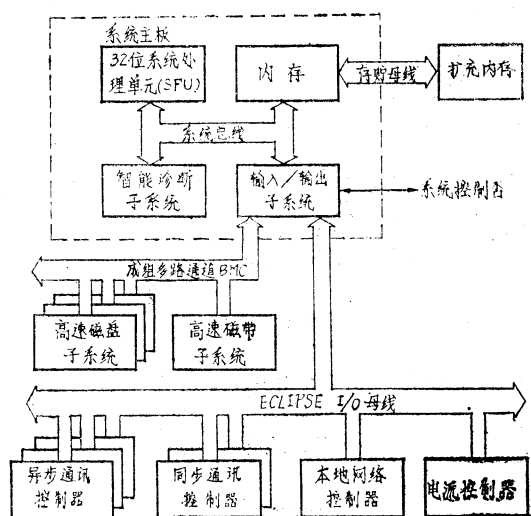


图2 典型的系统组成

## 1. 系统总线

系统总线是32位多路地址/数据总

线，它用于系统处理单元和存贮器、I/O及诊断子系统之间的通讯。总线工作的最大带宽为12.5MB/秒。系统总线支持多个具有最大地址空间为32MB的多路请求器，因此系统总线也起到存贮总线扩展的作用。总线的存取是用一个优先级表加以控制的。给予当前的请求器为最高优先级，存贮子系统为二级，I/O子系统为三级，系统处理单元(SPU)为最后一级。

系统总线的寻址方式是和32位ECLIPSE MV系列计算机系统物理寻址方式相对应的，数据一般为双字长格式(32位)而且，系统总线也适用于单字节(8位)和字(16位)的数据传送。

## 2. 存贮器

MV/7800系统支持4GB的逻辑地址空间，MV/7800系列支持最多14MB的物理主存和0.4GB的联机虚拟存贮器(高速的磁盘)。

有了这个存贮系统，系统处理器就能够有效的运行大的应用程序，而且同时支持在多道程序和多任务环境中的数据库。

为了便于管理大容量的存贮器，MV系列的系统把存贮系统逻辑地分为页和段，如图3所示。

段号	二级 页表指示器	一级 页表指示器	地址 位移
----	-------------	-------------	----------

图3 逻辑地址 = 段号 + 页表指示器 + 位移

## (1) 页管理方式

所有的ECLIPSE MV系列的机器都把存贮器分为页，每页为2KB，这个容量既能最大限度地发挥内存的性能，又使内存的空区变为最小。页的管理方式简化了内存的寻址，同时增强了内存的管理功能。从图2可以看出，它有二级页表指示器，用于计算有效的逻辑地址。系统如果只有一级页表

指示器, 而不用二级页表, 寻址的空间限制在1MB, 但可以得到更快的运行时间。页管理方式使得操作系统容易按2KB的较小增量控制其存取权限, 并根据运行程序的需要动态地进行分配。

## (2) 段

在页管理的基础上, 每页2KB的寻址空间又被组织为段, 每段512MB, 各段又组成一个分层的园环, 如图4所示, 这样可以方便内存的寻址, 但更重要的是保护系统软件。在逻辑层中的段编号从0-7。每个环包围了一个内存段。

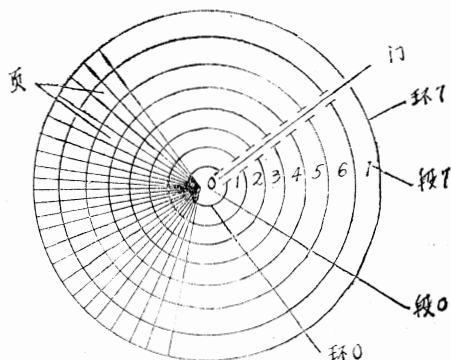


图4 内存层次结构

MV系统中, 操作系统核心是放在0段, 即环的最里层。在这个最优越的层次中, 核心系统可以自由的跨越环的边界去存取外面任一个环的内容。反之, 应用软件应放在第七层, 即最外面一层, 在这个最低层次运行的程序, 如要存取更高层次的信息; 需经过一组环的门径, 这些门径只能得到预先定义的软件的允许才能打开。这样就有效的保护了系统软件。MV系统提供给操作系统优先段管理指令。

## (3) 内存的组成

物理主存系统由内存贮控制单元和两个存贮阵列所组成(见图5)。

### 1) 内存控制单元

内存控制单元(MCU)采用ECL门阵列电路, 它管理系统内存并提供和系统总线

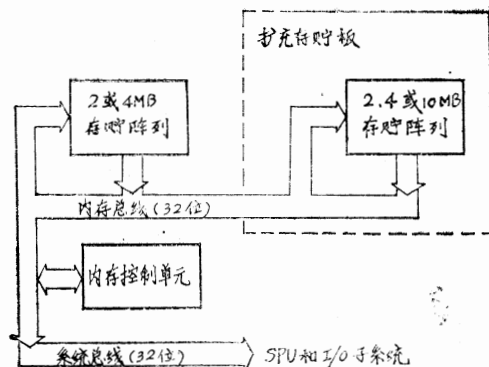


图5 物理主存的组成

的接口。MCU控制存贮的阵列的存取, 产生刷新地址, 检查写入读出信息的完整性和正确性。

为检查其完整性, MCU在写内存数据时附加了一位错误校正位, 在读内存数据时MCU能检查和校正一位数据错, 在内存刷新操作时MCU可有选择地进行完整性检查。它可防止一位错故障积累成破坏性的多位错故障。

### 2) 存贮阵列

系统处理器板上有2或4MB的存贮阵列。内存可最大扩充到14MB, 按2, 4或10MB的阵列增量扩充内存板。内存阵列采用100nS的RAM。这个内存速度和系统总线的速度完全相匹配, 因而消除了系统高速缓存的需要, 达到了系统性能的峰值。

## 3. 系统处理单元

带有硬浮点单元的SPU是MV/7800完成基本的处理功能的核心部件。它有三片NMOS的超大规模集成电路。SPU运行的主周期时间近似为320nS, 并且提供MV系列带有硬浮点部件的CPU的功能。

### (1) 指令组

SPU执行ECLIPSE MV系列32位指令组, 包括16位和32位的指令, 以及十进制指令组(DIS)。用蕴含指令组(IIS)的形式对一些通用的指令提供微码的支持。为了达到兼容的目的, 16位的指令作为它的指

令系统一个完整部分。

IIS加速了科学应用软件的性能,包括单双精度浮点代数和三角函数,如平方根,正弦和余弦。DIS加速了在COBOL应用中定义的十进制数据的比较,位移,增量和递减量。

32位的指令组还包括了增加操作系统效率的指令,如队列和特权的内存管理指令。同时有的一条指令就能执行高级语言的语句,如循环DO LOOPS。

## (2) SPU的构成

SPU执行多级的流水线,运行的主周期时间为320ns。它由下面的微码控制单元组成即:

中央处理部件(CPU);微顺序;浮点部件。

SPU是围绕着二条数据总线即系统总线和微控制总线构成的。这些总线连接了三个主要的SPU部件,每个部件集成在一片NMOS VLSI片中(见图6)。

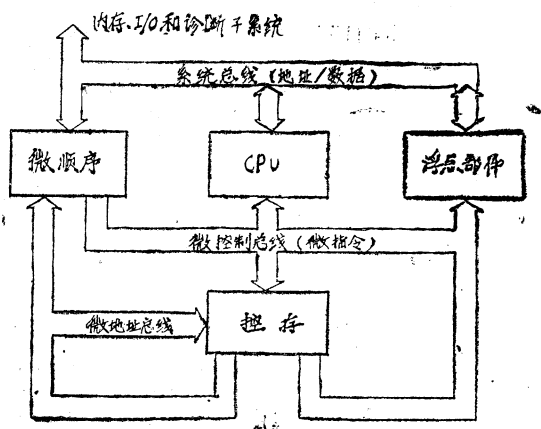


图6 系统处理单元

系统总线在三个部件之间传送地址和数据,同时也在SPU和内存,SPU和I/O,SPU和诊断子系统之间传送地址和数据。微控制总线把从微顺序器来的微指令送到控存,并从控存送往CPU和FPU。系统总线和微控制总线一起允许诊断子系统把

SPU初始化。

控制存储器是由RAM组成,有16K微字,40位宽的空间。在系统初始化期间,诊断子系统从外围设备中读入微码送到控存中。

CPU是32位的微程序处理器,执行地址转换和所有非浮点指令要求的数据操作。

FPU执行处理浮点数据指令的数据操作和执行处理定点乘除数据指令的数据操作。FPU包含有四个浮点累加器,每个累加器为64位以适应单双精度的操作。为了提高浮点运算的精度,FPU有两个6位备用数字位。根据程序控制标志的状态决定其结果的取舍。

## 4. I/O子系统

I/O子系统有MV/7800虚拟和附加的磁介质的高速存贮系统。同时,它提供MV/7800系统板的主要部件和各种通讯网络控制器之间的高速接口。提供各种外围设备和电源控制器。

### (1) I/O的构成

图7为I/O子系统的构成图。两个控制器为系统外围设备和存贮器,系统外围设备

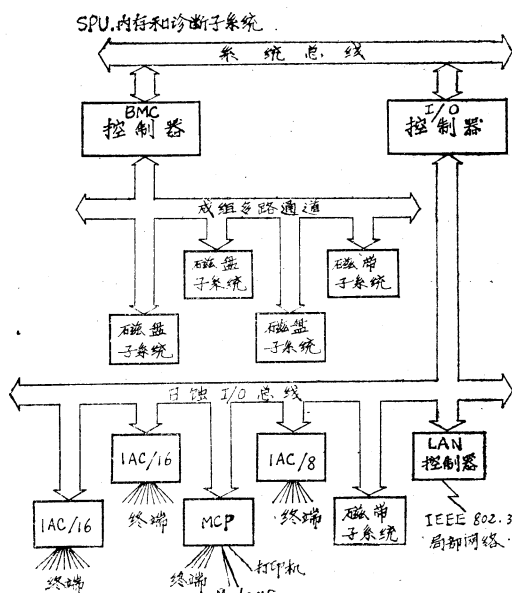


图7 I/O子系统的构成



和SPU之间提供通路。每个控制器支持着不同的总线。

### 1) 成组多路通道控制器:

控制器采用CMOS门阵列技术,支持一个成组多路通道(BMC),该通道用于高速的磁盘和磁带等设备,直接和内存进行数据块传送。BMC控制器支持符合总线传送约定的、并把有BMC的地址转换为内存的物理地址的变换图的设备。

为了有效的执行地址变换,BMC控制器有它自己的高速缓存和页表存取逻辑,这个高速缓存存有最近使用过的四页的传送地址。控制器允许智能设备控制器装入自己的传送地址(内存变换图)。

BMC控制器最多支持4个设备控制器,输入的传输率,最大为每秒10MB,输出近似为每秒8MB。

### 2) I/O控制器

I/O控制器支持数据通道和程序I/O设备,数据通道装置支持中速的直接内存存取的设备,存取的数据可以是单字或字节或是可变长度的数据块。传输率输入为每秒2.1MB,输出率为1.3MB/每秒。程序I/O支持低速的外部设备,在设备和SPU的累加器之间传送单字的信息。

和BMC控制器一样,I/O控制器也有自己的高速缓存和页表逻辑,高速缓存存有最近使用过的六页传送地址。它也允许智能设备控制器装入他们自己的地址变换图,且具有保护位。

和诊断系统相结合,I/O控制器有记录错误的装置,以报告硬件的失效。

I/O控制器还提供有控制台终端接口,可编程的时间间隔计时器和实时时钟。

### 3) 系统外围设备

MV/7800 I/O子系统的设备包括:9.4GB的高速磁盘存储器;8台磁带机;各种终端、打印机和图形设备。

### 5. 通信和网络控制器

MV/7800分布式处理结构可延伸到I/O子系统和各种智能的通信和网络控制器,包括:智能异步通信控制器;智能的同步通信控制器;多路通信处理机;智能的LAN控制器;智能计算机到专用交换机(PBX)接口。

智能的异步通信控制器有IAC/8,IAC/16符合EIA RS-232-C标准接口,同时还有计算机到PBX接口的异步通信设备,CPI/24,每个CPI/24最多支持24个设备,在计算机和PBX之间的吞吐量CPI/24比IAC高三倍。以上的异步通信控制器最多支持128个设备。

同步通信控制器有ISC/2。

多路通信处理器MCP,它提供8路全双工串行的异步智能通信控制器,和2路全双工串行的智能同步通信控制器,它支持位和字节的同步通信规约,传输率为56KB/秒。MCP有并行的行打接口。MCP支持大多数DGC的通信软件产品,如:XODIAC/X·25,X·25/SDLC,SNA/SDLC,RJE80,HASPI和RCX70,DG/GATE,DG/XAP和DG/BLAST。

智能的LAN控制器(ILC),是一个高性能的LAN设备,它遵守IEEE802.3局部网络标准。除支持TCP/IP通信软件外,MV/7800还支持XNS/ITP/VS,XNS/ITP/VS执行Xerox网络系统Internet的传送规约。

MV/7800除了支持上述的通信和网络产品外,它还能支持DG硬件和软件的卖主所开发的通信和网络产品。MV/7800也可以加入高速的局部网(称为HYPER通道),和其它的计算机系统如Cray的通讯。

### 6. 诊断子系统

诊断子系统是由微处理器组成的部件,功能如同MV/7800有一个前端诊断处理器。它支持系统的加电,系统初始化和系统故障的远程诊断,因为它的高度专门化功能,诊

断子系统可作为一个独立的计算机工作，用它自己的操作系统支持。

诊断子系统的组成（见图8）：围绕着远程诊断处理器（DRP）有8位的远程总线，和加电存贮。（即存贮了加电检查程序）

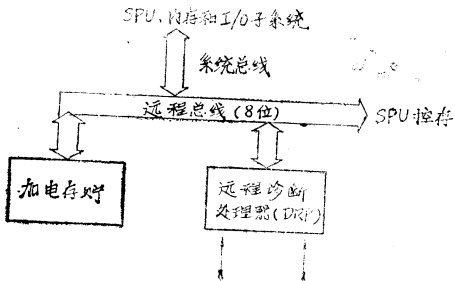


图8 诊断子系统的组成

## 7. 电源

在MV/7800的16槽机箱中包含有供给系统的直流稳压源，并且是一个智能化的电源控制器（UPSC）。它允许SPU通过I/O总线和电源系统进行通讯。工作时UPSC检查通风故障，电源过载和机箱温度，当出现故障时，中断CPU并报告故障的临界条件。

## 二、MV/7800的软件系统

### 1. MV/7800支持以下的操作系统：

- AOS/VS （虚拟操作系统）；
- AOS/RT32 （实时操作系统）；
- AOS/DVS （分布式操作系统）。

AOS/VS和AOS/DVS 包含有MV/UX，即UNIX操作系统的简化版本（Ported Version）。

MV/UX提供一个通用的分时的UNIX操作系统环境，它和 Berkeley，AT&T的UNIX操作系统的版本相兼容。

### 2. 通信与网络软件。

AOS/VS支持以下的通信网络软件：

- X·25 CCITT的包交换的协议。

- XODIAC 以X·25为基础的网络管理软件具有远程资源代理和远程INFOS I，DG/DBMS代理等功能。

- TCP/IP 在IEEE 802.3 LAN中，用于UNIX和AOS/VS系统之间的通信，是一种工业标准的通信协议。

- DG/SNA IBM系统和ECLIPSE计算机系统之间的通信软件。

- DG/SDLC 同步数据链控制协议，用于和有SNA的IBM系统的数据交换。

- XDLC X·25的数据链控制软件，通过X·25的网络和使用SNA的IBM系统交换数据。

- DG/BLAST 异步的文件传送程序，用于和其他的计算机如IBM系统的文件传送。

- DG/XAP 用于DG计算机系统之间的文件传送。

- CEO连接。

- CEO文件转换。

- RJE80 IBM2780或3780远程作业录入终端仿真程序。

- RCX70 IBM3271 远程群组通信，仿真IBM3271群控制器，使ECLIPSE系统能和IBM主机通信。

- HASPI IBM HASPI RJE工作站仿真程序，要求有一个同步通道。

- DG/GATE 一般的异步终端仿真程序。

### 3. 数据（资料）管理软件

DG/DBMS是和CODASYL兼容的数据库管理软件，有和COBOL, FORTRAN 77, PL/I 高级语言的接口。

INFOS I，功能强且为通用的软件包，用于文件的组织和管理。支持COBOL。商用BASIC FORTRAN V，FORTRAN 77, RPG11，PL/1 PASCAL, DGL，

APL,C等高级语言。

DG/SQL 是一个关系数据库管理系统, 主要用于工程, 科学和CAD/CAM等。高级语言COBOL, FORTRAN77, PL/1, C, Pascal, BASIC有接口。

CEO DG的CEO软件无论是商业(事

务), 工业和技术行业的办公室都将增加其办公室的出产率。CEO的字处理, 数据(信息)管理和决策支持,使其提供一个真正的综合事务自动化产品。同时, 由于和分布式数据处理功能相结合, 使得CEO发挥了 现存的决策支持的全部优点。图9所示为 CEO系统。

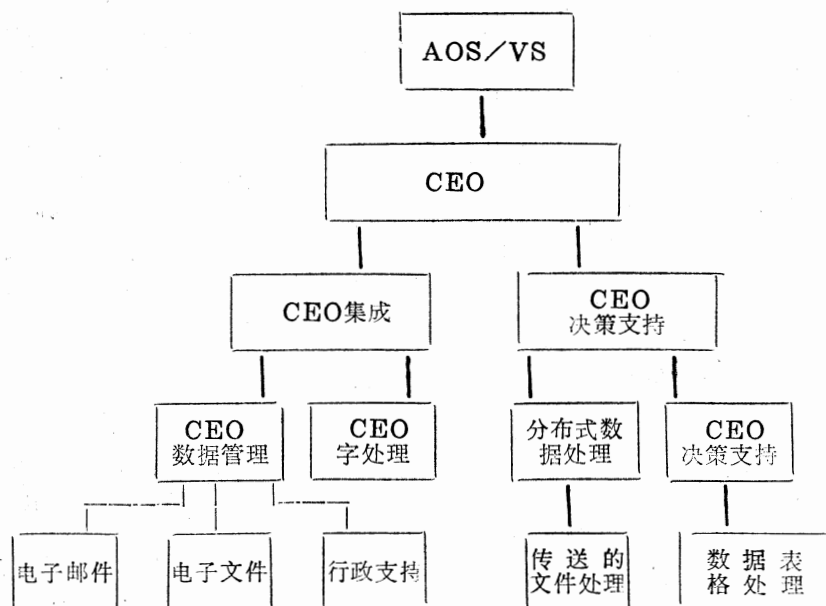


图9 CEO办公自动化系统

### 三、MV/7800的应用

MV/7800可广泛的用于办公自动化, 工业和技术的各个领域。

MV/7800通过采用14MB大容量存贮器和前后端处理机, 实现了128用户的支持, 且可配置大容量的磁盘, 可构成大规模的数据库, 同时有好的办公自动化软件CEO, 因此该系统在汉化后可在办公自动化中得到广泛的应用。

MV/7800和其他的DG产品一样, 有丰富的网络通信软件, 因此在各种事务处理中, 尤其在大范围内需要本地或远程网络支持的环境中应用, 将发挥其优点。MV/7800还具有和IBM中大型机以及其他机种通信的

能力, 因此在一个较大的系统中也应有它重要的地位。

MV/7800速度快, 且配有高速的浮点处理部件, 因此在各种结构分析, 模拟实验等需要高计算能力的科技领域, 也能以出色的性能价格比适应其要求。

MV/7800的基本系统只有一块板, 因此特别适合于作为OEM或系统内装的处理机, 它将在CAD、CAE、CAM等各方面得到广泛应用。

MV/7800有9.4GB的大容量外部存贮器和高速的输入输出通道, 因此在分布式环境下作为文件服务机。

配有AOS/RT32的MV/7800适用于实时处理的环境, 可以进行高性能和高速的计算, 以满足实时响应的要求。



#### 四、MV/7800 和MV/4000性能比较

##### 1. 系统的技术性能（见表1）。

表1

	MV/7800	MV/4000
字长	32位	32位
指令宽度	16~80位	16~80位
虚拟地址	4000MB	4000MB
实际地址	14MB	8MB
处理机周期	320ns	200ns
控存	16kw, 40B/w	选件
内存取数时间	100ns	200ns
WIPS	1050k	600k

##### 2. 系统的最大配置（见表2）。

表2

	MV/7800	MV/4000
主存	14MB	8MB
I/O控制器	14	
磁盘存贮器	9400MB	4820MB
磁带存贮器	8	8
异步通信	128线	64线
同步通信	16线	4
数据通道打印机	6	2
程序长度/每个用户	2000MB	2000MB

##### 3. 数据传输率（见表3）。

表3

	MV/7800	MV/4000
CPU - 内存	12.5MB/S	10MB/S
I/O - 内存	10MB/S	5MB/S
成组多路通道		
输入	10MB/S	5MB/S
输出	8MB/S	
数据通道		
输入	2.1MB/S	2.5MB/S
输出	1.3MB/S	1.25MB/S

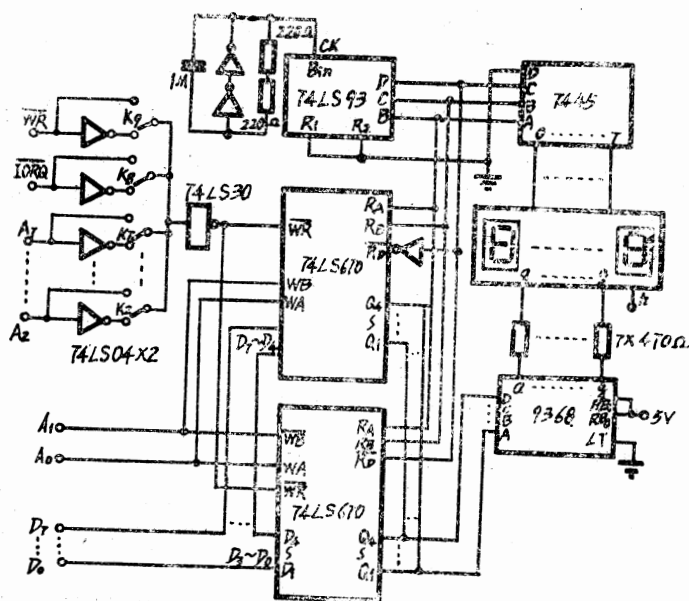
从上述几项性能的比较可看出MV/7800的性能优于MV/4000。MV/4000配置为CPU，2MB内存，AOS/VS的系统售价为3.9万美元；而MV/7800配置为CPU，2MB内存，AOS/VS的售价只有2.1万美元。当然MV/7800还得配上一米高的机柜等，但他们的差价在1万美元以上，MV/7800在系统板上还有FPU，而MV/4000以上的配置却没有FPU，要配上FPU，得另加5千美元以上。所以MV/7800有较好的性能价格比。

86年1月份，DEC发表了VAX8000系列的计算机。该系列机的性能从低档VAX8200到高档机VAX8800。VAX8200（1.1MIPS）性能优于MV/4000（0.6MIPS），接近于MV/7800（1.05MIPS），低于MV/8000。最高档机VAX8800和MV/20000—2性能相近。VAX8200，配置为CPU，4MB内存，操作系统VMS使用许可，基价为98000美元，比MV/7800价格要高，因此从性能价格比上看，MV/7800也优于VAX8200。

综上所述，MV/7800是一种32位的超级小型机，采用了VLSI技术，在单板上集成了SPU，MCU，FPU，4MB内存及I/O通道控制等功能，因此有较好的性能价值比。又由于MV/7800的设计上保持DGC系列计算机兼容的特性，使得MV系列计算机的系统软件和应用软件可以在MV/7800这种新型计算机上运行，因而它也和以前的计算机一样，可以在工业、商业、办公自动化、CAD等各种应用领域发挥其作用。

付国明 编译

### 用在Z80微处理机上的8位十六进制数字显示器



本文介绍的 8 位十六进制数字显示器仅占用四个 Z80 微处理机端口，它可直接与微计算机连接，用几条指令即可工作。用它来显示寄存器内容、地址或数据效果极好。它将给调试带来方便。

图中的 $K_2-K_7$ 是为设置显示器的地址所用。一个数据字节写入所使用的四个端口中的一个时，它可以被存贮起来，并作为两个十六进制数被显示。如只要显示十进制数，则可用74LS48来代替9368。74LS670是 $4 \times 4$ 位寄存器，本图中的接法特点是数据以四组8位从一侧输入时，在另一侧的输出端将得到八组4位数据。沙建军译

沙建军译

(上接第37页)

2. 例子：天津百货采购站要预测全国家用缝纫机的普及率。

序号	年份 (x)	普及率 (y)	序号	年份 (x)	普及率 (y)
1	1964	3.5	12	1975	14.5
2	1965	4.1	13	1976	15.6
3	1966	4.8	14	1977	17.0
4	1967	5.5	15	1978	18.6
5	1968	6.2	16	1979	20.9
6	1969	7.1	17	1980	23.5
7	1970	8.3	18	1981	27.8
8	1971	9.6	19	1982	32.7
9	1972	10.9	20	1983	36.8
10	1973	12.2	21	1984	40.9
11	1974	13.3	22		

要求：先根据缝纫机从1965—1984年的历史数据，绘制出普及率曲线，根据趋势再确定用何种定量预测方法。普及率的历史数据如上表。

运行程序号,把年份数和普及率两组数依次送进计算机,计算机根据这两组数可自动计算出坐标值并绘制出曲线。见图2。

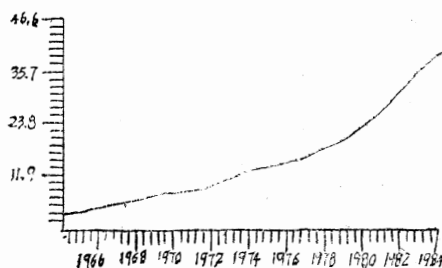


图2 缝纫机历史趋势图

根据曲线有趋向饱和的趋势, 则选择成长曲线法 (逻辑斯蒂克函数) 来预测普及率。

## 32位机总线的实际性能比较

提要：本文通过一张项目齐全的表格（见文后附表）对32位底板总线标准作了一客观实际的比较。该表的准确性目前是完全可以信赖的，随着时间的推移这类底板总线可能会因规范、环境和商业支持的变化而有所改变。为此，应定期地对该表作一些更新，并以最新的标准版本为准。表中项目统计的截止日期是1985年12月18日，版本3.0。表中“不支持”一词的含意是指不支持间接地按该总线规范操作的设备，这并不是说，这些性能不能通过一些非标准手段或将来规范的修改来实现。但是由于许多特性要求结构和协议应与规范相符，而这些特性是不太可能移植到现有规范中的。

### 一、性能

#### 1. 总线带宽

表中性能数字的算值尽可能地考虑到符合各总线实际应用惯例。计算中包含：总线驱动器和总线接收器传输延时；发送线路的驱动延时（指底板上的总线驱动问题和信号传输延时）；逻辑偏差和底板弯曲；锁存器的建立和保持时间；从设备访问时间。由于同步总线须有足够长的时钟周期以适应全负荷系统，为此对异步总线计算也假设一满负荷系统，即使这种假定可能永远不会出现：对尺寸较小的底板或在较轻巧的典型加载系统中，异步总线速度应快一些。

Nubus和Multi bus I 的性能有许多相似之处，因它们都使用一个以10MHz计时的同步协议。Multibus I 和 Nubus 在单传输方面的性能都具有某些并发特性，这些并发操作可能在地址译码，脉冲访问延时，地址和数据多路转接延时之间产生。

Nubus的仲裁时间比Multibus I 短，因为它的底板尺寸小，仲裁线少，因而逻辑操作更快。

通过简单的计算可知道，在底板上的任何二块通信插件板间的平均距离为底板长度的1/3。因此，在异步协议的底板延时计算中采用这一指标。同步总线的时钟周期固定，它必须能覆盖相距最远的两块板间通讯所用的时间。在任何时刻，两块板间的通信不影响它们的性能。

用于VMEbus, Futurebus和Fastbus的满负荷底板延迟时间大体相同，即使Future bus收发器加载总线少。Fastbus和Futurebus基本使用同样的协议，只是总线带宽不同，仲裁性能不同，这主要是由于采用的工艺不同（即Future bus采用先进的Schottky TTL，这与VMEbus一样，而Fast bus使用10KECL）。

在单传输方式的图中给出一个值得注意的结果：所有的总线性能表现很相似。其主要原因是，VMEbus（非多路复用）不象其他总线那样容许在一根线上复用地址和数据时有延时，因为在协议中Future bus和Fast bus除多路复用外还有一些处理开销：Future bus提供延时以克服在全广播方式地址转换中称为Wired-OR glitch的传输线效应。Fast bus在各种处理结束处有40ns的清除延时（为有效地解决同类问题）。

猝发性（burst）传输方式，也就是常说的在典型多处理机系统的处理中面向通讯的先决条件，在总线性能方面表现较大差别。然而，即使如此，在较长的访问时间处性能仍将迅速达到均等。

从这种研究得到的经验是：高性能总线



并不仅仅是高性能系统的保证。为充分利用总线可用带宽,一些结构式方法是必须的。例如,一种要么减少各板产生的总线通信量,要么减少“有效”总线访问时间的方法是必须的(这可通过使用字组经由总线通信以及使用“静态列地址”或半字节方式访问存贮器板上RAM来实现)。减少总线通信量的有效方法是:各处理机和总线间的高速缓存器;要么在各板上,要么通过一些局部扩充总线可访问的局部存贮器。前者是对程序透明的,后者一般不对程序透明。使系统在总线上传输字组而不是单个字节的有效传输方法是:高速缓存器。限制系统为功能性分布(报文通行)结构。高速缓存和报文通行在下面介绍。

附表对同步总线的性能作了大量有关处理机,存贮器和总线时钟相关性的假设。Multibus I 和 Nubus 上每一次传输,时钟延时问题将会付出半个时钟周期(50ns)的开销,从而在许多实际应用中降低了系统性能。

下面讨论时钟延时问题。如果处理机访问的所有指令和数据直接经由总线,那么所有基于总线的多处理机系统将由于瓶颈而受总线限制。这样,就必须寻求一种方法,尽可能使指令和数据位于各处理机附近,以减少使用总线的次数。为此,各处理机本地执行和操作数据的时间百分比应较大(90%),而它们访问总线的时间百分比应较少(10%),于是,优化处理机访问本地资源而不是更多地访问总线变得很实用。即选择一个处理机时钟速率使存贮器访问时间最经济实用。12.5MHz和16.7MHz处理机目前是流行的,预计1986年内会出现20MHz和25MHz的处理机。

这就是说,各局部处理机时钟将是不同的,且与总线时钟相互异步。结果,处理机需访问总线的几率为10%,它的接口必须以某种方式协同两种传送数据的方法。这种协

同引起两个问题:等待第一个有效时钟沿所需的延时问题;本质上为同步电路的暂态问题。

处理机通过总线完成的各种传输,在处理机将其请求信号送到接口后,接口必须等待第一有效时钟沿的到来。请求信号应刚好在总线时钟有效沿前到达,这样就不存在等待时间;如请求信号在总线时钟有效沿后到达,那就要等一个周期直到下一个时钟沿到达。为此,总的平均延时为半个时钟周期。

Multibus I 通过总线提供一种缓冲处理机定时的方法来解决这一问题,称为报文通行。即提供一种设备准备接受这种体系结构的约束条件。用这种技术基本上可解决延时问题。Nubus也有时钟延时问题,但并不明显地提供一种象报文通行那样的机构来解决延时问题。

## 2. 仲裁

估计总线仲裁时间开销和总线平均数据保持时间以代数形式和数字给定,这点是非常有用的。 $T_{arb}$ 为仲裁系统处理多个请求所花的时间。设这段时间以前不由某设备占有,则它等于该主设备获取总线而又尚未占用它所需的时间。所有总线都有一种有效的“停放”(Parking)方式,即如果总线主设备不变,总线获取基本不用延时。这在VMEbus中称作‘释放请求’(ROR)。除了在主从型多处理结构中‘执行释放’(R-WD)一般不使用。这种结构使用VMEbus效果很好。

$T_m$  是进行控制的时间,即任何主设备被允许在正常情况下保持总线的时间。对某些总线,并不总是明确规定它;而其他的则慎重地不去规定它。故设定这一参数完全是系统操作人员的特权和责任。

对Nubus,  $T_m$ 由数据块传输方式的协议建立。VMEbus为简化操作,要解决数据块传输中存贮器定界问题, $T_m$ 是任选的。在Multibus I 情况下,也不对 $T_m$ 限定,它与

报文通行方式中最大有用报文的尺寸有关。对Future bus和Fast bus,鼓励系统操作人员通过编程来适当限定其系统;16或32方阵(128字节或256字节)的选择为最大,插件板被编程将大数据块按这种最大尺度分为几个小数据块。但如要将虚拟存贮器的整个一页从磁盘控制器送到主存贮器时,有些系统中最大数据块尺寸要达1k字节(256次传输)或更多。

主设备获取各总线的时间( $T_{get}$ )也给出了。这里,设定数据块为16字,从设备访问时间为150ns,每个系统16个模块。有优先级时,给出 $T_{get}(best)$ ,设定系统中只有一个高优先级的主设备。在无优先级(Fairness)时,给出 $T_{get}(Worst)$ ,设定16个模块在公正方式下全部操作。

$T_{get}(best)$ 是满载系统中高优先级模块请求总线到获取总线的最大时间间隔。在一系统处理实时数据或必须对高速事件作出反应时这一参数最为重要。很清楚,如果总线当时在使用,那么即使仲裁发生,优胜者在以前主设备完成操作之前是不能控制总线的。这就出现一个复杂问题:即出现一排队模型,该模型中一块板正在使用总线,另一块板已裁决优先,等待使用总线(称‘主选’)。为此,一种新的高优先级主设备请求总线不仅要等到当前主设备完成操作,而且还要等到‘主选’完成,尽管主选可能是较低的优先级。 $T_{get}(best)$ 值由因子 $2T_m$ 反映,即两个主设备各自使用总线所占用的最大时间。

只有Future bus和Nubus不用 $2T_m$ 来表示。Future bus有一称为“预空”的内部机构,它允许在总线交出之前以高优先级主设备代替低优先级主设备选择。Nubus没有优先级机构用于仲裁,故以 $T_{get}(best)$ 和 $T_{get}(Worst)$ 各自反映最佳、最差访问时间。VMEbus有一种由BCLR线提供的与Futurebus的予空相类似的机构,

该机构被用来通知当前主设备,更高优先级主设备正等待使用总线,也可用于预空主选。然而,这在规范中没有详细规定,因而未包含在此表中。

$T_{get}(Worst)$ 是满载系统中从标准优先级模块请求总线到获取总线的最大时间间隔。该参数反映了无优先级仲裁的算法,确保所有请求设备得以操作。由于这种无优先级机制并不是一种完全先进先出算法,而仅仅是一种近似算法,故不能进一步细算。所有主设备在无优先级机制总线上都视为权利均等。下面让我们看看当打开这种机制的“闸门”,在请求器仲裁总线时会发生什么情况。如果最低全同模块以外的所有模块发出请求,那么该门关闭一个周期为这些请求服务。而后这些最低全同主设备立即准备请求门关闭,这可能不仅要等到当前排队处理结束( $(N-1)T_m$ 周期),而且在fairness门再次打开,所有主设备(最后一个除外)再一次准备请求总线时,同一队列要第二次排队(进而要 $(w-2)T_m$ 周期)。假设原先排队中排在最后的主设备由于它一直有效,错过了门第二次打开。对Nubus,正是考虑到 $T_{get}(best)$ 和 $T_{get}(Worst)$ 之间的差别,故使用无优先级机制实现仲裁。

## 二、价格和复杂性

表中的这一节指出了各总线的有关性能价格比。使用Multibus II可能代价很高,但这种状况似乎会改变: Intel声称,在不远的将来,Multibus II会比VMEbus总线便宜,因为其接口的集成度会更高,而硅材料成本会不断下降。

VMEbus由于传统的设计,以及与68000系列元件级信号相吻合,结构较简单。Future bus和Nubus总线必须以全32位宽度总线操作。这就是说,在使用16位处理器时其性能价格比低于VMEbus,但它却解

决了兼容性问题,消除了由于32位机中的混编带来的复杂性。Future bus的高速缓存能力,报文通行,广播功能和复杂的控制状态寄存器(CSR)功能都可消除或明显地减少操作的开销。特别是,它的插件板极易用低速,高集成度部件如PAL器件(可编程逻辑阵列)构成,它采用独立异步交换技术,可确保完全兼容性。因而具备很宽的性能价格范围。

由于Multibus II解决了时钟延迟问题,报文通行对获得实际的优越性来说都或多或少地超过Multibus I;报文通行在LSI中,实际等级和软件二次开发方面代价都很高。但Intel很巧妙地解决了这个问题:其iSSB串行总线使同一报文通行功能作为并行总线来实现。以致执行时带宽要求很低,仅一或两台处理机可使用iSSB总线,故很有效。

Nubus具有新的机制。由于它的性能不很复杂,所以比Multibus II便宜得多。Fast bus一般用于高性能系统,例如用在树形结构网络上连接大量数据采集子系统。

### 三、多处理机支持

#### 1. 超高速缓存器

总线对超高速缓存器的需要常常会引起误解。所有的总线都能实现某种超高速缓存系统。其基本策略是使用指令超高速缓存器和数据超高速缓存器。值得注意的是,任何总线都可实行某种指令超高速缓存(因为指令不大可能被修改)。尽管数据或数据与指令组合的超高速缓存器要比单纯的指令超高速缓存器有用,但实现起来困难较大。如果系统程序员至少能确定哪些数据是专用的,哪些是共享的,那末,他就能借助于存储器

管理单元使超高速缓存系统将所有共享数据看作是非高速缓存的。虽然这是一种低性能,混乱和非透明的方法。但所有总线都可按这种方式实现高速缓存。要求透明的和(或)高速缓存共享数据的较高性能系统为保持一致性,在总线中必须设置某种基本机构。这里有两种方案:写通过(Write through)即所有被写数据贯穿高速缓存器经由总线写入主存储器;写回(Write back),即仅把被写数据在送出之前简单地存入高速缓存器中。在用Write through方式时,总线上可以观察到处理过程,并且总线支持全交换广播机制(至少对地址),接着,具有失效数据拷贝的主设备可使该数据无效(或将更新的数据拷贝到总线,使其变为有用)。

对最高性能的Write back系统,附加的数据必须与各数据片一道存贮,以定义数据所有权。若该数据标为共享,那么转换到Write through方案;如果数据标为专用,那么写处理不必在总线上出现。数据是否变为共享或专用取决于数据首次读入时的总线状态线,或者,只要其他处理机发现在总线上数据写到同一单元时就可变其为无效。假如在超高速缓存器中有一数据片,标为专用,再假如另一块板希望读该数据所占据的主存储器单元,那么超高速缓存器必须在读周期时进行干预以禁止主存储器送出失效数据,并从自己内部取出最新数据提供给主设备。

只有Futurebus和Fastbus支持符合要求的广播机制,以支持Write through方案。只有Futurebus有必要的总线状态和协议干预装置,以支持高性能Write Back方案。

附表 32位总线的比较(附后)



附表 32 总线的比较

项 目	VMEbus	Futurebus	MultibusII	Nubus	Fastbus
<b>支 持</b>					
标准代号	IEEEP1014	IEEEP896.1	IEEEPxxx	IEEEPyyy	ANSI/IEEE960
现状	草案1.2, IEC821草案	草案7.2, 预计86年	Intel Rev.c	草案1.0	审定标准IEC45
最初研究者	Motorola	IEEEP896W.G.	Intel	Texas Instruments	USNIM委员会
最初支持者	Signetics, Mostek	尚无商用规范	-	Lisp Machine公司	Kinetic System, LeCroy Research
目前硅片支持	Signetics Motorola	国家半导体公司	Toshiba	没有	MarueiShoji公司提供几种门阵列
希望硅片支持	-	Ferranti, Monolithic Intel Memories Signetics, Texas Instruments	Intel	Texas Instruments	Valtronic公司Integrated Networkg.Phi-lips
性能	连续变化	连续变化	量化—100ns等待状态	量化—100ns等待状态	[连续变化
总线带宽	20~57兆字节/秒	117.6兆字节/秒 (相邻接的板)	40兆字节/秒 (设定10兆时钟)	37.5兆字节/秒 (16兆时钟)	160兆字节/秒
研究者宣布	Motorola	IEEE Micro, 84.8, P42	Intel	Texas Instrument	IEEE Trans.Nucl. Sci 85.2
资料来源	Motorola	IEEE Micro, 84.8, P42	Intel	Texas Instrument	IEEE Trans.Nucl. Sci 85.2
单向传输方式 (平均底板延时 = 底板长度的三分之一).					
Tacc = 0ns	25.0兆字节/秒	37.0兆字节/秒	20.0兆字节/秒	20.0兆字节/秒	37.0兆字节/秒
Tacc = 50ns	19.0兆字节/秒	25.3兆字节/秒	13.3兆字节/秒	13.3兆字节/秒	25.3兆字节/秒
Tacc = 100ns	15.4兆字节/秒	19.2兆字节/秒	13.3兆字节/秒	13.3兆字节/秒	19.2兆字节/秒
Tacc = 150ns	12.9兆字节/秒	15.5兆字节/秒	10.0兆字节/秒	10.0兆字节/秒	15.5兆字节/秒
Burst传输方式 (交换的字组长度不限, 底板平均延时 = 底板长度的三分之一)					
Tacc = 0ns	27.3兆字节/秒	95.2兆字节/秒	40.0兆字节/秒	40.0兆字节/秒	173.9兆字节/秒
Tacc = 50ns	20.7兆字节/秒	43.5兆字节/秒	20.0兆字节/秒	20.0兆字节/秒	54.8兆字节/秒
Tacc = 100ns	16.5兆字节/秒	28.2兆字节/秒	20.0兆字节/秒	20.0兆字节/秒	32.5兆字节/秒

续表

$T_{acc} = 150ns$	13.6兆字节/秒	20.8兆字节/秒	13.3兆字节/秒	13.3兆字节/秒	23.1兆字节/秒
流水线方式 (非交换的数据块长度不限)	不确定, $\sim 55$ 兆字节/秒	不确定, $\sim 280$ 兆字节/秒	不用, 40兆字节/秒	不用, 40兆字节/秒	不确定, $\sim 500$ 兆字节/秒
将来性能	未定义	按P896.2定义	30兆字节/秒	未定义	未定义
报文通行方式	过渡时间, 总线驱动问题, 逻辑延时, 不对称, 底板传输在规范内定时约束	逻辑延时, 不对称, 底板传输延时	信号采样间隔100ns, 不计时钟延时	信号采样间隔100ns, 不计时钟延时	逻辑延时, 不对称, 段传输延时
主要限制					
仲裁					
算法	RWD, ROR	Fair, 优先级	Fair, 优先级	Fair	Fair优先级
$T_{arb}$ (典型)	200~400ns	250ns	300ns	200ns	150ns
$T_{arb}$ (最佳)	150ns	150ns	300ns	200ns	90ns
$T_m$	266T <sub>t</sub>	未限定	32T <sub>t</sub>	16T <sub>t</sub>	未限定
$T_{g.o.t.}$ (优先)	$T_{arb} + 2T_m$	$T_{arb} + T_m$	$T_{arb} + 2T_m$	$T_{arb} + (N-1) + 2T_m$	$T_{arb} + 2T_m$
$T_{g.o.t.}$ (Fair)	$T_{arb} + T_m[2(N-2) + 1]$	$T_{arb} + T_m[2(N-2) + 1]$	$T_{arb} + T_m[2(N-2) + 1]$	$T_{arb} + T_m[2(N-2) + 1]$	$T_{arb} + T_m[2(N-2) + 1]$
用 $T_{arb}$ (典型)比较, $T_m$ 限16次传输, 且用字组传输方式(150ns访问时间)					
$T_{g.o.t.}$ (最佳)	9.8 $\mu s$	3.3 $\mu s$	9.9 $\mu s$	139.4 $\mu s$	5.8 $\mu s$
$T_{g.o.t.}$ (最佳)	136.9 $\mu s$	89.5 $\mu s$	139.5 $\mu s$	139.4 $\mu s$	80.5 $\mu s$
价格与复杂性					
地址空间	64	1	4	1	2
连接器	2	1	1	1	1
接脚	128	96	96	96	130
有效信号线	107	67	67	46	60
中断线	7	0	0	0	1
电源	+5V, +5VSBY, $\pm 12V$	只有+5V	+5V, +5V电池, $\pm 12V$	+5, -5.2V, $\pm 12V$	+28V, $\pm 15V$ , +5V -5.2V, -2.0V

续表

所公布的电气与可靠性				
总线接口	TTL混合(48mA, 64mA); 三态和集电极开路。	BTL(50mA底板收发器逻辑), 解决总线驱动问题。	TTL混合(48mA, 64mA); 三态与集电极开路。	ECL10k(几乎解决了总线驱动问题)
奇偶位	没有	任选	任选	任选
禁止控制	没有	通过CSR空间	没有	enable信号线
并行总线	没有	1位/字节	1位/字节	全部1位
控制线	没有	1位	2位	没有
标记	没有	1位	不用	不用
裁决	没有	1位	没有	没有
维护和配置能力				
可动插入	不存在	支持	不存在	部分支持(有一半线)
扩充卡	不允许	全部支持	不允许	部分支持
地区寻址	没有	5位槽ID	LACHn脚	5位ID
自动配置	不支持	任何时刻都支持(包括可动插入)	只在通电时支持	全部支持
多处理器支持				
虚拟中断	未定义	支持; 详情在P896.2中定义	支持	支持
裁决	4级链式	全分布; 异步	分布, 同步中央时钟	分布, 中央定时器件
确定性探测	支持; 四级深度循环系统	支持, (Fair)	支持 (Fair)	支持(保证访问方式, Fair)
优先级测试	支持, 仅除循环外四级	支持, 32级	支持, 可能32级	支持, 32级
报文通行	不支持	格式定义, 在P896.2内规定	支持	不直接支持
超高速缓存器				
写完成	能力限定	支持	能力有限	能力有限

续表

写回	不支持	支持	不支持	不支持	不支持	不支持
标记结构	不支持	用奇偶保护支持标记位	不支持	不支持	不支持	不支持
协议特性	不支持	支持	不支持	不支持	不支持	不支持
总线协议	异步	异步; 技术独立	同步; 10MHz时钟	同步; 10MHz时钟	同步; 10MHz时钟	异步和任选同步子操作
数据通路	非多路复用	多路复用	多路复用	多路复用	多路复用	多路复用
主通路	16位	32位	32位	32位	32位	仅32位
辅通路	32、24、16和8位	32、24、16和8位	32、24、16和8位	32、24、16和8位	32、16和8位	不支持
对齐	16位对齐	非对齐	16位对齐	16位对齐	非对齐	非对齐
非校准32/16位						
操作	仅对Rev.c	完全支持	完全支持	完全支持	完全支持	不支持
字节定位	大的端点	无约束	小的端点	小的端点	小的端点	不用
地址空间						
主	2 <sup>24</sup> 字节	2 <sup>32</sup> 字节	2 <sup>32</sup> 字节	2 <sup>32</sup> 字节	2 <sup>32</sup> 字节	2 <sup>32</sup> 四字节组
辅	2 <sup>32</sup> 字节	可扩展	无	无	无	可扩展
互连	在I/O空间	在CSR空间	2 <sup>14</sup> 字节	2 <sup>14</sup> 字节	在CSR空间	在CSR空间
I/O空间	2 <sup>10</sup> 字节	在CSR空间	2 <sup>10</sup> 字节	2 <sup>10</sup> 字节	在CSR空间	在CSR空间
广播(写至多个从设备)	不支持	在任何写操作广播	仅报文空间, 存储器和I/O空间不支持	仅报文空间, 存储器和I/O空间不支持	不支持	在任何写操作广播, 模块子集和系统子集选择装置
收集(读自多个从设备)	不支持	在任何读操作收集	不支持	不支持	不支持	少量数据通过T脚扫描
总线中继电器电路转换	不支持	总线中继电器死锁防止	总线中继电器死锁防止	总线中继电器死锁防止	总线中继电器死锁防止	完全支持
包转换	不支持	底板上扩充看作存储转发。	256节点用定义的报文通行机制。	256节点用定义的报文通行机制。	未定义	支持, 但ANSI/IEEE 960未定义
为将来扩充保留	1根保留线, 少量地址修正符。	协议中的扩充命令方式	2根保留线, 在SC线上1保留状态。	2根保留线, 在SC线上1保留状态。	未注明	包转换操作。5根保留线
锁定制优锁定	操作中用BBSY线, 并释放。	操作时释放	锁定制有效, 操作时释放	锁定制有效, 操作时释放	依靠Fair机制, 持续请求	操作时释放
地址锁定	所有传输假设定。	支持(锁定制)	支持(锁定制)	支持(锁定制)	不支持	不支持

单处理锁定	整个操作由AS插入	支持(锁定线)	支持(锁定线)	不支持	由AS/AK锁定
多处理锁	不支持	支持(控制变化不锁)	不支持	不支持	不支持
总线诊断特性					
调试	无	交换广播方式	无	无	禁止超时,等待线单步处理
监视/扫描	无	异步连接相允许检查地址	逻辑分析仪检查	逻辑分析仪检查	为检查,用等待线降低交换
		传输。			
总线效用	少量地址修改码; P <sub>2</sub> 连接器64脚空。	定义的装置用作辅助连接	附加功率总线	附加功率总线	( $\pm 12V, -5.2V$ );
物理特性		器,以保证插入所有的板	( $\pm 12V, +5V$ )	( $\pm 12V, -5.2V$ )	
板尺寸					
主板尺寸	233.35mm × 160mm	366.7mm × 280mm	233.35mm × 220mm	366.7mm × 280mm	366.7mm × 400mm
辅板尺寸	100mm × 160mm	233.35mm × 280mm	100mm × 220mm	无	无
板面积					
主板面积	373cm <sup>2</sup>	1027cm <sup>2</sup>	513cm <sup>2</sup>	1027cm <sup>2</sup>	1467cm <sup>2</sup>
辅板面积	160cm <sup>2</sup>	653cm <sup>2</sup>	220cm <sup>2</sup>	无	无
连接器					
专用型	2(IEC603-2)	1(IEC603-2)	1(IEC603-2)	1(IEC603-2)	AMP2-532956或Dupont 66527-565或SAERTP2525-1303
脚数	2 × 96	96	96	96	130
非专用型	0(64脚)(IEC603-2)	2(IEC603-2)	1(IEC603-2)	2(IEC603-2)	(如AMP2-532981-1)
脚数	在辅连接器上64	2 × 96	96	96	最大195
模块数					
逻辑模块	没有逻辑规定	每块底板32,65536个报文节点	每块底板32; 256个报文节点	每块底板16	每底板段16,所有接入的系统
专用物理槽	1	0	1	0	统中为16777216 × 255
非专用物理槽	20	21(48cm槽)	19	16	26(48cm槽)



# 微小型计算机的性能评价与选型

毛德操 毛根生 胡希明

(浙江大学计算机科学与工程系)

用户在购买计算机之前首先要考虑选型问题。

所谓选型,就是在若干种供选择的机型中就性能和价格两个方面进行权衡、挑选,这就要求对各种机型的性能有个恰当的评价。对计算机性能的评价表面看来很简单,实际上却很复杂。本文拟结合实例对此作一简要的介绍。

首先要明确:“性能”到底是指什么?专家们对这个问题的回答可谓意见纷纭、并无定论。首先我们应该排除主观因素,用客观的数值来反映机器的性能,这就是所谓“计算能力”。另一方面,由于主观因素取决于人们在使用计算机系统时的感受,因此也不应该完全加以排除;先将其排除是为了使不同系统的性能有一个统一的量度,排除主观因素以后,还是有问题。表面上看,似乎把计算能力定义成单位时间内完成的计算量就可以了,但是对“计算量”又该怎样定义呢?

计算能力的最粗略的定义是每秒钟百万条指令,即MIPS。其实MIPS是最不能说明问题的,在这里甚至连每条指令所处理的字长也没有考虑。因此,MIPS充其量也只能在字长相同的机器间作粗略比较时使用。与MIPS相似的MF-LOPS(每秒钟百万条浮点指令)也存在着类似的问题。

考虑到MIPS的主要缺陷,人们就用MIPS跟字长的乘积(每秒钟处理位数)来表征计算能力,常称为PDR。1PDR就是每秒钟一百万位的处理能力,PDR大致上能反映出计算机的计算能力。所以,巴黎统筹会和美国国防部在审批计算机出口时都以PDR作为计算能力的度量。例如,巴统会规定绝对禁止向苏联和东欧出口计算能力超过48PDR的计算机,相当于速度超过1.5MIPS的32位机。对规定完全解禁的是:5PDR以下可拆卸、搬动的计算机,28PDR以下不可拆卸、搬动的计算机。

对PDR以及MIPS中的指令条数并未说明是哪一类指令,而不同种类指令的执行速度一般都是不同的,所以实际上总是将执行各种指令的速度加

权平均。实际上PDR的计算式为:

$$PDR = (0.45A + 0.4B + 0.15C) / (0.85X + 0.09Y + 0.06Z)$$

这里: A = 平均指令长度

B = 定点操作数长度

C = 浮点操作数长度

X = 定点加法的平均执行时间

Y = 浮点加法的平均执行时间

Z = 浮点乘法的平均执行时间

除PDR外,人们另外提出过一些算式,如RP.Cervený和K.E.Knight:

$$P = 10^{12} \{ (L-7)(T)(WF) / [(32000)(36-7)] \}^{1/(t_{opr} + t_{i/o})}$$

式中: L = 字长(位)

T = 内存容量(字)

WF = 字长修正系数,对定字长内存为1、变字长内存为2。

$t_{opr}$  = CPU执行一百万次运算所需的平均时间

$t_{i/o}$  = 执行一百万次I/O操作期间

CPU处于空转等待状态的时间

其中 $t_{opr}$ 是对各种运算指令—包括定点加法、浮点加法、乘法、除法、及逻辑运算计权相加而得的,各类指令的权值又因科技计算或商业数据处理应用而不同。对 $t_{i/o}$ 也有一个复杂的计算式,此处从略。

这个计算式中的32000(36-7)反映了计算P时所考虑的标准内存大小,为32000×36位;(L-7)则使短字长(如8位)的机器处于很不利的地位。可见此式对于不同字长的计算机并不公正,其中仍然掺杂着相当大的主观因素。

实际上,迄今为止还找不到一个公式能全面而客观地反映各类计算机的计算能力。其原因在于:

首先,不同机型的指令集不一样。一台机器上的一条指令,其功能可能相当于另一台机器上好几条指令的总和。有些处理器的指令集可能特别适合面向商业的应用;而另一些则特别适宜于科学计

算,用于商业就差一些。

其次,计算机的计算能力并不单一地取决于CPU的速度和字长。内存的大小、系统结构的不同、输入/输出的方式和速度、外部设备的大小和速度等因素都影响着计算机的实际计算能力,但又很难用解析的方法加以描述。

更重要的是,现代的计算机绝不会以裸机的形式出现在用户面前,而总是与一定的软件结合在一起,这些软件掩盖了硬件的某些特性。从使用者的角度来说,所要求的也是整个虚拟系统的综合计算能力,而不是计算机硬件本身的计算能力。

既然如此,人们就只好另辟蹊径,设法以经验和统计的方法来解决或减轻这些问题,力求准确、客观地反映计算机系统的计算能力。其方法是以一组标准的“水平测试”程序(benchmark)来检测不同计算机系统完成特定计算量所需的时间,从而反映出这些系统的计算能力。最早进行这方面工作的是奥尔巴赫信息公司(Auerbach Information, Inc.),该公司从1962年起便采用一组水平测试程序来测试、比较各种不同计算机系统,提供一种称为“标准EDP报告”的测试报告。他们使用的程序共五个:一个一般的文件处理问题;一个随机访问的文件处理问题;一个排序问题;一个矩阵求逆问题;一个一般的科学计算问题。从那以后,又有一些公司或组织面向不同领域的应用而定义了一些不同的水平测试程序。

水平测试的结果反映了被测系统完成各种基本操作或提供各种基本功能所需的时间。用户根据各自的应用领域将这些数据加权相加,便可得出一个综合的数值,该数值即反映了被测系统在此领域中的计算能力,从而为选型提供了比较准确的客观依据。不过,水平测试数据也不能用作衡量计算能力的唯一依据,因为许多通用计算机系统要用来解算各方面的问题,往往很难准确地预计这些问题将各占多大比重。再说,有些因素,如内存、外存大小等,往往很难或者根本无法通过水平测试反映出来的。更何况还应考虑到一些与主观感受关系密切的因素(如响应时间、人机界面等)。因此,还是应该结合其它主观和客观的评价——包括可靠性、软件支持、可扩充性、可升级性等因素综合考虑。

作为实例,我们介绍一下美国商务部、大气海洋局、环境卫星数据中心、地球物理数据中心联合于1983年中至1984年底对超级微机市场所作的调

查。调查的结果发表于1985年初。

该项调查选择的准则是根据一系列先决条件确定一个范围,再在此范围内进行水平测试,选出其中的佼佼者。这些先决条件是:

- 1.支持多用户。
- 2.有一个功能较强的操作系统。
- 3.带大容量的(在线)磁盘。
- 4.可支持数个磁盘和/或磁带子系统。
- 5.采用适于速数据传送的总线结构。
- 6.支持Ethernet局部网络。
- 7.有常用的程序设计语言,包括

Fortran、Pascal、Cobol、和C。

对于符合这些条件的机型进行了水平测试,测试程序共十一个(算法相同而使用不同语言者按一个计算)。例如:

程序一,测试CPU进行整数加法运算的速度,循环地做1000000次 $K = K + 1$ ,测定其所需的时间。采用Fortran77语言。

```
k = 0
1000  print *, 'begin loop'
1001  k = k + 1
      if (k.gt.1000000) goto 1002
      goto 1001
1002  print *, 'end of loop'
      end
```

程序都是用Fortran77语言编写的,与此相当的还有用C语言编写的程序,此处不再列举。

水平测试程序一般都是很简单的,测试的都是计算机系统的基本功能,而且复盖了各个有代表性的方面。程序一测试的是CPU作整数运算及判定转移的速度。这项测试数据还在一定程度上间接地说明了CPU作非数值运算的速度。程序二、三、四(略)都是测浮点运算速度的,但侧重面不同。程序二、三测试的是浮点加法的速度,程序四测试综合浮点四则运算的速度。程序五(略)至程序十(略)都是测试计算机系统的文件操作速度。测试的结果主要说明计算机的I/O能力及磁盘速度。

选择这些程序作水平测试,其目的主要是用于数据库环境下的数值计算。测试程序中没有包括用于BCD码运算的程序,可见其目的不在于商业数据处理,也没有包括排序等常用的数据结构方面的处理,因而主要目的也不在于非数值运算。

值得一提的是, 这些测试程序是用高级语言编写的, 因而测试结果与编译的关系极大。此外, 与操作系统的关系也很大, 因为许多操作 (特别是文件操作) 是通过系统调用由操作系统内核完成的。所以, 测试结果实际上是CPU速度, 计算机系统结构、操作系统、编译及外围设备的综合效果, 而这正符合一般的要求。有时候, 为了排除软件的影响, 也有用汇编语言编写测试程序的。

对当时美国市场上的多种机型进行水平测试及试用后, 四个单位联合发表了调查报告, 认为在符合前述先决条件的机型中以采用超级微型机较为合算, 其中尤以Charles River公司的UNIVER-

SE68/137T更令人满意。报告中列举了水平测试的结果, 其测试数据摘录见表1, 其中以VAX11/780、VAX11/750 (超级小型机) 及IBM PC (微型机) 的相应数据作为对比。从中可以看出UV68/137确实是一种很不错的机型。表中所列的是UV68/37的测试数据, 比UV68/137稍差一些, 但相差不多。UV68/37采用12.5MHz的M68000处理器为CPU, 带32兆磁盘, (可扩至200兆), 在系统结构上采用32位的VERSAbus总线, 有4K字节高速缓存, 并且用另一片频率较低的M68000处理器来管理字符设备的I/O操作; 操作系统为经过改进的UNIX, 称为UNOS。

表1

运行时间(秒) 程序 机型	一	二	四	五	七	八	九	十
UV68/UNOS(带浮点选件)	3.9	39.9	34.5	13.7	17.7 (c0.4)	22.7	19.5	18.3
VAX11/780/VMS	5.9	7.1	11.7	4.8	3.9	4.5		
VAX11/750/UNIX	10.2	19.6	180.0*	15.6				
VAX11/750/VMS	6.4	6.5	15.2	4.6	3.3		4.4	26.7
IBM PC	51	224						

C表示用C语言改写测试程序后的结果

• 资料原文如此, 疑为18.0之误

从表1中可以看出:

第一, UV68的整数运算速度极快, 甚至超过了VAX11/780, 实际上已接近中等规模大型机的水平(程序一在UNIVAC-1162机上的测试结果是3.3秒)。

第二, UV68的浮点运算速度适中, 低于超级小型机的速度。这主要是因为M68000处理器中没有浮点指令, 而硬件浮点加速板(选件)与CPU又无法耦合得很紧的缘故。不过, 比之其它的超级微型机或小型机, UV68的浮点运算速度还是慢的。

下面列出几种机器的浮点运算速度:

W1CAT(软件浮点) 0.0022MFLOP  
 IBMPC(带8087协处理器)0.0027MFLOP  
 SUN(软件浮点) 0.010MFLOP  
 ApolloDN420(硬件浮点)0.037MFLOP  
 UV68/137(硬件浮点) 0.050MFLOP  
 DG M600(硬件浮点) 0.055MFLOP

第三, UV68顺序写记录时的速度比超级小型机略慢, 这主要与总线上的数据传输速率、磁盘速度、编译质量有关系。

第四, UV68须改进其Fortran77语言顺序读记录时的速度, 因为若用C语言实现同一目标竟比用Fortran77快十倍, 由此可见, 水平测试所反映的是硬件和软件的综合效果。一般来说, 超级微型机的编译质量不如超级小型机好, 所以其潜力并没有充分发掘出来。但是, 与W1CAT、SUN、MASSCOMP、Apollo等相比, UV68仍不失为一种出色的超级微机。

顺便指出, VAX机在采用UNIX操作系统时的速度显著低于采用VMS, 这可能是因为其目标代码的质量不够好。但仍有许多用户宁可采用UNIX, 其原因很多, 如软件的可移性、兼容性等, 但也不容否认, 主观(感受)因素在性能评价和选型中所起的重要作用。

从硬件组成来看, UV68能在水平测试中取得好成绩并非偶然。该机采用12.5MHz的M68000处理器为CPU, 并设计了高速缓存, CPU的速度与内存和总线匹配得很好, 从而消除了CPU在访问内存时的等待状态, 使CPU的潜力得以充分发挥。而有些设计得不太好的系统, 则在很大程度上压抑了

CPU的能力。例如，WICAT-150 采用8MHz 68000芯片，用程序一测试时，人们期望其运行时间与UV68相比大约应为12.5 : 8，可是实际上却是17.5 : 3.9，改用C语言编写程序一后的测试结果为3 : 1，仍不能接近12.5 : 8。这说明WICAT-150系统中CPU的潜力没有充分发挥，其原因是CPU与内存匹配得不好。如果采用说明书中的CPU时钟频率来推算运算速度作为性能评价依据的话，那就与实际情况相差很远了。此外，UV68具有比较好的I/O总线结构，经实测其传输速率可达每秒20兆字节，也有利于提高I/O操作（及文件操作）的速度，这在一般超级微型机中也是不多见的。

除计算能力以外，“性能”还应包括存储能力，即在磁盘和磁带的容量。至于磁盘的速度，在水平测试中已反映在计算能力里了。

除性能外，选型时考虑的第二个主要因素就是价格。它们之间保持什么样的关系才最合理呢？

早在40年代末期，Herbert R.J.Grosch就总结了当时计算机的性能和经济两方面的关系，得出一个描述计算能力和机器价格的经验公式：

$$P = KC^2$$

就是说，计算机的计算能力与价格的平方成正比。Grosch没有正式发表他得出的这个关系式，但后来常被用来衡量计算机的性能和价格是否有竞争力，并被称为“Grosch定则”（Grosch's Law）。

可是，这个公式至少是不确切的，因为如前所述，对计算机的计算能力其实并没有一致而科学的定义。从五十年代到现在，对计算能力的定义经常在变。拿Cerveng和Knight的公式来说，其中有一项（L-7），这显然只适用于微处理器问世之前，以4位微处理器为例，若以L=4代入公式，则显然会得出荒谬的结果。即然如此，Grosch的公式就不足为凭了。

从经济学的观点来看，计算机为用户提供的是使用价值（主要体现为计算能力和存储能力），理应与其价格成正比例，所以现在都采用“性能价格比”来指导选型，能较好地定量地客观反映计算能力的则是水平测试。因此，先剔除外存容量的因

素，然后以水平测试的数据按应用方向加权相加后作为性能的量度，在满足特定先决条件（如功能、软件资源、支付能力等）的前提下选择性能价格比较高的机型，并结合可靠性、可扩充性、可升级性、维修服务等条件综合考虑，就是合适的选型方法。

下面列出几个机型当时的市场（参考）价格：

厂商	型号	参考价(美元)
Charles	UV68/37	25000
River	(扩至2兆内存)	
DEC	VAX11/780(2兆内存)	145500
DG	MV8000 I (2兆内存)	130000
Wang	VS100 (2兆内存)	110000
IBM	PC	3000

可见，对于以整数运算应用（及非数值应用）为主的用户来说，UV68的性能价格比显著高于VAX11/780，与IBM PC相比也处于有利的地位。对于浮点运算为主的应用，UV68的性能价格比也能与VAX11/780相当，但比IBM PC略低，对于磁盘操作十分频繁的应用来说，UV68的性能价格比仍略高于VAX11/780，不过UV68实际可配备的磁盘容量（及磁带容量）远远小于VAX11/780

由此可得出这样的结论：如果仅从性能价格比考虑，则典型的超级微机优于超级小型微机，而个人计算机又可能略优于超级微型机。但用户在选择机型时往往附有一些先决条件，例如前述调查中因有七项条件而未将IBM PC列入选择范围；所以选中的未必就是性能价格比最高的机型。不过，随着时间的推移，情况也在发生变化，目前的超级微型机虽还不能像超级小型机那样配备大量在线磁盘和磁带，软件也不那么成熟、丰富，但这是可以很快改变的。即使在目前，调查报告也认为像UV68这样的超级微型机完全可以用于一般的数据中心及各种数据处理、图形、图像处理等应用环境。目前超级微型机CPU的发展方兴未艾，Charles River公司已经推出采用16MHz的68020处理器的新机型，旧的UV68也可以现场升级成采用68020，这样其计算能力就可以从1.2MIPS提高到2.7MIPS，而价格增加不到50%。估计在今后相当一段时期内，超级微型机将是用户的主要选择。

（上接第29页）

随着计算机的推广应用以及在异种机上解决数据共享的必要，ORACLE关系数据

库将在中国国内受到欢迎，并为实现分布式数据处理做好充分工作。

# 一个软件开发估算系统的设计

何建中

(广州市新技术应用研究所)

## 〔摘要〕

本文介绍了一个在APPLE II PASCAL上实现的软件开发估算系统。给出了进行软件开发估算的算法, 数据结构和程序实现的描述, 并说明了估算的基本功能和特点。

“估算系统”可供软件开发人员对规模在20万条语句以下的软件开发工程进行开发代价估算。

## 一、软件开发估算原理介绍

### 1. 标定软件开发局部环境

标定软件开发局部环境, 其意义是根据不同的开发项目的要求和特点, 对一系列影响软件开发生产率(代码行数/人月——LOCS/MM)的因素, 进行比较确定的估计, 从而为软件平均生产率以及开发难度的推断, 提供一个估算的基础。

令 $S = \{S_i \mid i = 0, 1, \dots, n\}$ 为软件开发环境状态变量集合。其中 $S_i = \{S_{ij} \mid j = 1, \dots, m\}$ 为有关子状态变量集合。从每个 $S_i$ 中, 选取一个 $S_{ij}'$ 子状态变量, 其所组成的集合即标定了—个软件开发局部环境:

$S' = \{S_{ij}' \mid i = 0, 1, \dots, n, j' \in \{1, \dots, m\}\} \subset S$ 因此, 对于不同的开发项目, 可以标定不同的软件开发局部环境 $S'$ 。

### 2. 软件平均生产率估算

对于标定的软件开发局部环境 $S'$ , 对应—有标准的软件生产率数据参考集合:

$N = \{N_{ij}' \mid i = 0, \dots, n, j' \in \{1, \dots, m\}\}$

软件平均生产率可由下式求出:

$$A_N = P_1 \frac{\sum_{i=1}^{d_1} N_{ij}'}{d_1} + P_2 \frac{\sum_{i=d_1}^{d_2} N_{ij}'}{d_2 - d_1}$$

$$+ P_3 \frac{\sum_{i=d_2}^{d_3} N_{ij}'}{d_3 - d_2} + P_4 \frac{\sum_{i=d_3}^n N_{ij}'}{n - d_3}$$

$A_N$ ——软件平均生产率

$d_1$ ——分析阶段状态变量数。

$d_2 - d_1$ ——设计阶段状态变量数。

$d_3 - d_2$ ——编码阶段状态变量数。

$n - d_3$ ——测试阶段状态变量数。

$p_1, p_2, p_3$ 和 $p_4$ 分别为对应的权因子  
( $p_1 + p_2 + p_3 + p_4 = 1$ )

### 3. 代码行工作量价格估算和任务工作量价格估算

代码行工作量价格估算是软件开发估算的一种基本技术的应用。其关系如下:

$$\text{PRICE}_1 = \frac{1}{6} \text{DG} \sum_{i=1}^N (\text{SL}_i + 4\text{PL}_i + \text{NL}_i) \quad (1)$$

$\text{PRICE}_1$ ——代码行价格估算值(¥)

$\text{DG}$ ——开发难度系数。

$\text{SL}_i$ ——开发项目功能 $i$ 最小代码行估算值(LOCS)

$\text{PL}_i$ ——开发项目功能 $i$ 最可能代码行估算值(LOCS)

$\text{NL}_i$ ——开发项目功能 $i$ 最大代码行估算值(LOCS)

$N$ ——开发项目划分功能数目

(1) 式为代码行工作量价格估算式。



$$WORKS_1 = \frac{1}{6} \sum_{i=1}^N (SL_i + 4PL_i + NL_i) / A_N \quad (2)$$

WORKS<sub>1</sub>——程序员人月工作量数

(2) 式为代码行工作量估算式。

任务工作量估算关系则如下：

$$PRICE_2 = A_N \times DG \left( \sum_{i=1}^N (C_1 AM_i + C_2 DM_i + C_3 CM_i + C_4 TM_i) \right) \quad (3)$$

A<sub>N</sub>——软件平均生产率。

DG——开发难度。

N——划分功能数。

AM<sub>i</sub>——开发项目功能 i 分析工作量 (人月)

DM<sub>i</sub>——开发项目功能 i 设计工作量 (人月)

CM<sub>i</sub>——开发项目功能 i 编码工作量 (人月)

TM<sub>i</sub>——开发项目功能 i 测试工作量 (人月)

C<sub>1</sub>、C<sub>2</sub>、C<sub>3</sub>、C<sub>4</sub>——对应任务阶段工作量价格因子。

(3) 式为任务工作量价格估算式

$$WORKS_2 = \sum_{i=1}^N (AM_i + DM_i + CM_i + TM_i)$$

——任务工作量估算式

#### 4. 关于代码行估算范围的冒险概率分析

代码行估算的冒险概率分析基于离散的正态分布原理。

设：SUB

$$= \sqrt{\sum_{i=1}^N \left[ (NL_i - SL_i) / 6 \right]^2}$$

为代码行估算标准差数。

在估算冒险概率为68.3%的情形下，代码行工作量变化范围为：

$$EL - SUB \sim EL + SUB$$

$$\left( EL = \frac{1}{6} \sum_{i=1}^N (SL_i + 4PL_i + NL_i) \right)$$

价格变化范围为：

$$PRICE_1 - PRI \sim PRICE_1 + PRI$$

$$(PRI = SUB \times DG)$$

工作量变化范围为：

$$WORKS_1 - W \sim WORKS_1 + W$$

$$(W = SUB / A_N)$$

还可以给出在不同概率下的估算范围分析，也可以进行类似的任务工作量估算的冒险概率分析。

#### 5. 开发人力需求的估算

开发人力需求的估算根据开发组织的通讯原理给出。

在开发人员之间通讯是必要的，通讯路径关系如下：

$$L = M(M-1) / 2$$

L——通讯路径

M——人数

开发组织内的通讯一般会降低软件的生产率，因此一个M人的开发组实际生产率为：

$$A'_N = (M - LQ) \cdot A_N (LOCS / \ln m)$$

A'<sub>N</sub>——开发组实际生产率。

M——开发人数。

L——通讯路径。

Q——通讯降低效率。

A<sub>N</sub>——软件平均生产率。

因此，由下式可以在开发周期MON已定的情况下求出所需人力数目。

$$MON = \frac{EL}{A'_N} \quad (1)$$

EL——代码行工作量估算期望值。

MON——开发周期(单位为月)。

由(1)式可以推算出开发组所需人员的数目。

在实际应用中可以知道过大的人员开发组和过短的开发时间, 均对生产率有较大的影响。

根据上述原理和方法, 我们就可以进行一般的软件工程开发估算的工作, 以评价有关项目的开发代价。

## 二、系统实现

按功能划分, 系统分为以下几个模块:

- 标定软件开发局部环境模块;
- 开发项目软件规模估算模块一; (建立代码行工作量估算文件模块)
- 开发项目软件规模估算模块二; (建立任务工作量估算文件模块)
- 软件生产率估算模块;
- 代码行估算技术应用模块;
- 任务估算技术应用模块。

前三个功能模块, 均面向具体的开发项目, 用来建立相应的磁盘估算数据文件; 后三个功能模块, 则对已建立的磁盘估算数据文件, 进行了一系列的估算处理和输出。在这个意义上, “估算系统” 实质上是一个磁盘文件操作的应用系统。

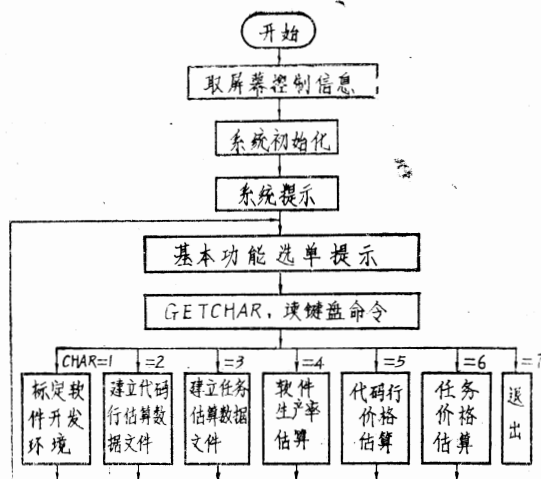
“估算系统” 的程序设计在使用形式上设计成面向用户操作的选单系统。功能提示选单为:

### 估算功能选单

- 1) 标定软件开发环境
- 2) 建立代码行估算数据文件
- 3) 建立任务估算数据文件
- 4) 软件生产率估算
- 5) 代码行价格估算
- 6) 任务价格估算
- 7) 退出

基本主控流程参见下图:

除7) 以外, 6个基本功能模块均可重入。因此, 对一个软件开发项目可进行动态的估算, 直至得出较为满意的估算结果为止。



## 三、结论

“估算系统” 的实现, 为从事软件和程序开发工作的单位和个人, 评价有关的软件开发代价, 估算软件的价值, 提供了一个实用的辅助工具。它既可以应用在软件开发工作的前端: 软件计划和分析阶段, 也可以用于软件开发工作的后端: 软件宣布阶段。这对于积累软件工程开发的历史数据和经验, 比较估算方案的效果, 都是极为有益的。

### 参考文献

- [1] APPLE I Apple PASCAL Language Reference Manual
- [2] APPLE I Apple PASCAL操作系统使用手册
- [3] Niklaus Wirth, “Algorithms + Data Structures = Programs”.
- [4] R.S.Pressman, “软件工程——实践者的研究途径和方法”。
- [5] J.L.Lawrence, “Why is software always Late? ”, ACM SIGSOFT, - Software Engineering Notes Vol 10, No 1.
- [6] GILL Ringland, “Software Engineering in a Development Group.” Software Practice & Experience, Vol 14, No.6, JUNE 1984.
- [7] 何克清, “计算机软件工程学”, 1982. 武汉大学出版社。

# ORACLE简介

董保华

(天津市电子计算机研究所)

## 一、Oracle的历史

关系数据库系统的概念是IBM公司的San Jose在1970年研究的基础上发展起来的。1976年11月IBM公司发表了关于SQL语言的完整的、规范化的说明。这时人们已认识到了关系数据库，无论从使用上还是向计算机提供技术能力上都是较为容易的，尽管它比占有优势的层次和网状数据库还慢一些。但遗憾的是关系DBMS的性能在一些大型数据库上仍然是令人失望的。

之后，Ellison和Bob Miner认为一个关系DBMS一定要表现出某些劣性并没有理论上的原因。他们预言经过七年SQL将成为关系DBMS的标准。

1977年他们建立了关系软件，Inc（后来改名为Oracle公司）确定了他们的目标：使用IBM公司的兰本和SQL的规范开发第一个关系DBMS，在速度和性能上，它将达到并超过传统的数据管管理系统。不仅经过专门训练的程序员需要它，并且普通的非程序员用户也需要它。

他们的目标在1979年实现了。当时第一版ORACLE装在顾客那里。直到1982年IBM公司推出SQL/DS之前也没有任何以SQL为基础的DBMS。

早在1980年Ellison和Miner就认识到了微机技术的开发将导致软件工业的革命。由于采用了价格不高的、分散的计算机资源，他们就预见到了各计算机公司将面临硬设备上的巴比尔之塔。在这种参差不齐的环境中，在正使用的大、中、小、微机上运

行标准的软件将是必要的。

为了满足这种市场需求，他们决定编制非常轻便的，可以跨越机种的第三版ORACLE。ORACLE和它的应用软件工具完全用C语言重新编写。这个目的是使第三版ORACLE独立于操作系统和计算机的体系结构。这项工作的结果可由以下事实证明：当ORACLE被移到一个新的环境中时，组成它的大约2000个程序仅仅需要修改50个。

到1983年，第三版ORACLE已装到若干种大、中、小和微型计算机上，此时ORACLE原封不动地进入了PC领域。IBM PC上的ORACLE与IBM公司大型机上的ORACLE功能完全相同，并不是一个子集。用户在不同的计算机系统之间仅做一点工作即可传输完整的数据库。

1984年该公司推出第四版ORACLE，提高了速度并采用了先进技术。在此之后，Oracle公司推出了ORACLE LINK。这是一个使运行ORACLE数据库的PC和小型机或大型机之间允许共享数据和应用程序的一个实用程序。通过ORACLE LINK出版说明可以看出该公司已朝着网络数据库管理系统前进了。

1986年2月第五版ORACLE出版了，介绍了ORACLE·net。它实际上给予了ORACLE分布式处理的能力，允许在各种不同的硬件上的ORACLE数据库和应用程序共享数据。第五版ORACLE还突出了一个新的特点，即ORACLE软件工具可以分开销售，允许用户决定选购他们仅仅需要的那部分ORACLE。这些工具无论是在性能方面，还是在使用方面都大大的改进了，

Oracle公司自宣布第一个产品到1985年雇员、收入及装机增长速度见图1。

## 二、产品介绍

### 1. 关系数据库DBMS

使用关系DBMS, 用户不需要知道他们的数据是如何存贮的。访问数据, 用户只要简单地指出他们所需要的数据, 而不必了解如何得到它们。DBMS通过计算机的内部存贮结构自动导航, 定位并将其取出。用户以非常简单而且熟悉的方式, 行和列组成的表去看他们的数据, 并且用SQL语言去操纵它们。SQL是一种类似英语的语言, 它容易满足非程序员使用, 而且还为数据处理专业人员提供了强有力的、完整的工具。

关系DBMS把所要看的数据从它们实际存贮的物理设备上隔离出来, 也就是说, 用户看到的和操纵的是被分隔存放于简单和独立的表格上的数据。一个关系DBMS并非必须分隔存放于各个表格, 这些表格可存放于计算机存贮设备中不同地方。多数关系DBMS分隔存放各个表格, 而ORACLE不这样。ORACLE能够将来自不同表格的数据存放于同一个磁盘页上。这种技术称为多表聚集。这使得ORACLE在多表处理性能上较之传统的DBMS是一个技术上的革新。

ORACLE提供了简易的使用方法以及现代关系数据库DBMS所不具有的兼顾系统性能的功能和可塑性。今天, ORACLE已用于存贮数千兆字节信息的大规模数据库。

### 2. SQL

所有ORACLE用户, 无论是在大型机、小型机、还是在微机上工作, 使用的是和在IBM上运行的关系数据库SQL/DS和DB2使用的同一个SQL, 而不是其子集。因此, SQL已成为终端用户和数据库处理专业人员之间的公共语言。用其即可进行应用程序之间通讯, 又可提供解决问题之办法。

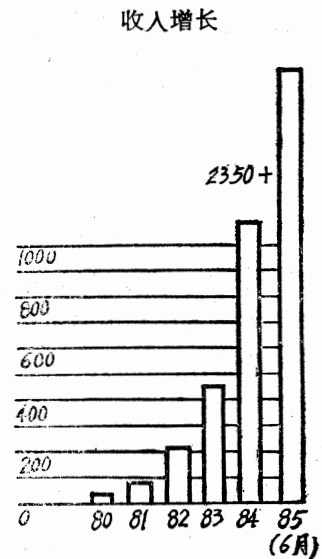
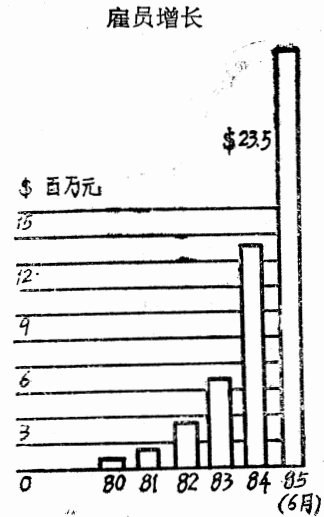
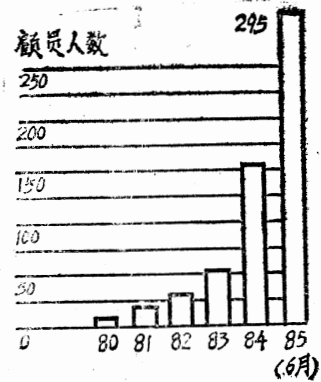


图 1

SQL仅仅是所需的语言，因为它包括查询命令，数据操纵、数据定义以及数据控制。SQL工具能够在终端上以交互方式使用，也可以嵌入标准程序设计语言，如COBOL，FORTRAN，C，Pascal，PL/I，BASIC和汇编语言等。

### 3. 软件开发工具

一个关系DBMS是简单的，它并不能代替应用程序。为了缩短开发应用所需要编写程序的时间，并便于用户建立与维护自己的应用程序而进行处理，那么增加软件开发工具是必要的。ORACLE包括下列各组开发工具。

#### (1) 交互应用生成工具——IAF

IAF (Interactive Application Facility) 是Oracle公司所提供的一种第四代软件开发工具。它包括两部分：生成部分 (IAG) 和执行部分 (IAP)。应用程序设计人员在终端上通过简单的问答对话将要求提交系统，IAG根据一系列回答进行编译，生成一个在IAP支持下运行的文件。用户可以根据设计出来的各种屏幕格式交互地对数据库进行查询、修改、删除和插入操作。不用编写程序就可以提交应用程序，而且可以对于所做设计进行编辑，生成新的应用程序。

#### (2) 报表生成软件 (REPORT WRITER)

由两部分组成：RPF和RPT。它们用于产生各种报表。RPT根据用户提供的SQL和过程控制语句提取数据库中的信息，产生一个临时文件，包含数据、表格描述文字和格式化命令。RPF根据该文件产生用户期望的报表。用几条简单的命令用户就能够增加页标题和总计，并且指定小计和其它运算等。由于这个软件工具仅有6个基本命令，所以不到一小时即可学会。

#### (3) 彩色绘图软件 (Color Graphics)

ORACLE图形软件允许用户以彩色图形，圆、曲线或直方图显示数据。用户指定ORACLE用SQL检索做图的数据。ORACLE支持各种图形终端，图形显示器以及硬拷贝设备。

所有的ORACLE软件工具可以配合起来工作，而且它们和ORACLE关系DBMS一样简便。可以一起形成一个综合的软件环境，开发应用程序而不需编写程序。

由于ORACLE和它的软件工具都是简便的，所以用它们开发的应用程序同样是简便的。一个应用程序能够在一个机器上开发，在任何运行ORACLE的机器上运行。

### 4. ORACLE Link

ORACLE Link是有关分布式数据库管理的第一个系列化产品。ORACLE Link允许一个ORACLE用户在PC上发SQL查询，通过普通的数据通讯将其传到一个正在运行ORACLE的小型或大型机上。于是SQL查询就面向这个目标机器(小型机或大型机)上的ORACLE运行，得到的结果再传回到PC，并且装入到它上面的ORACLE数据库。ORACLE允许个别的或临时的数据库用户访问公司的中央数据库，因此大大减少了长途通讯的费用以及与超载的数据处理部门之间的不必要的接触。ORACLE Link支持轻便的PC而不间断运行在公司的小型 and 大型机上的应用程序。

### 5. 新的方向

Oracle公司汇集了一流专家组成了小组，率先进行了分布式数据库管理开发工作。通过自己的开发工作以及与有关软件销售商的配合，该公司将继续提高ORACLE的性能并且增加新的产品，这些产品将使ORACLE构成一个完整的环境，在这个环境当中，全部应用程序，从设计到实现，正式的和非正式的使用都是自封闭的。Oracle将继续扩充能够在其上运行ORACLE软件的硬件设备。



附表 各种机器上面配置ORACLE情况

	厂 商	型 号	操 作 系 统
大型机	IBM	370,4300,30XX	VM/CMS,MVS,UNIX
	Amdahl	470,580	VM/CMS,MVS,UNIX
	Sperry	1100	UNIX
小型机	AT&T	3B5,3B15,3B20	UNIX
	Apollo	Domain	Aegis
	DEC	VAX-11,8600	VMS,UNIX,ULTRX
	DEC	PDP-11	RSX-11MX <sup>+</sup>
	DG	MV Series	AOS/VS,DG/UX
	HP	9000	HP/UX
	Harris	700,800,1000	VOS
	Honeywell	DPS-6	GCOS
	Prime	Series,50	Primos
	Pyramid	90X,98X	DS/X
	Sperry	5000,7000	UNIX
	Stratus	32	VOS
微型机	IBM	PC/XT,PC/AT	DOS,VENIX
	AT&T	6300,PC7300,3B2,3B1	DOS,UNIX
	CT	Mini,Megaframe	UNIX
	DEC	Rainbow,Micro VAX	DOS,VMS
	Motorola	6300,6600,2000	UNIX
	NCR	Tower	UNIX
	TI	Professional	DOS
	WANG	PC	DOS

### 三、ORACLE在中国

应中国有关部门的邀请Oracle公司于1986年到北京、天津、上海和沈阳等地进行技术交流。在京期间由北京科技交流中心主持了学术讨论会。政府部门,产业界和教育系统等许多人参加了会议。机械工业部提供了良好环境,包括IBM大型机,DEC公司的小型机VAX和微机IBM PC。这是一个ORACLE在异种机上运行的理想条件。

与会的专家中有的在关系理论方面有很深的造诣,他们对这次讨论会给予了很高评价。

国内自行研制的微机长城0520在用户中普遍使用。ORACLE关系DBMS在上面不做修改就可以运行并处理汉字信息。

目前国内机型,机种参差不齐。小型机和大型机以及微机之间共享数据越来越为用户所需求,开发应用程序的重复劳动也带来了不必要的浪费。上述工作正好解决了这方面的问题。

1986年9月应天津科技交流中心和天津市电子计算机研究所邀请,Oracle公司香港分公司赴津进行技术交流。在研究所有关人员配合下,在DG公司的MV系列机和IBM PC上分别安装表演ORACLE关系数据库并且利用Link在PC上向运行ORACLE的MV4000的数据库中存贮信息和取回信息。ORACLE关系数据库不经任何改动就可以在MV4000的汉字终端上处理汉字信息。这次技术交流引起了用户和数据库的同行们极大兴趣。(下转第22页)

# 怎样定义好激光打印机与计算机的软件接口

尹锋

(湖南省科技情报所)

现在,激光打印机的用户越来越多。激光打印机与计算机主机的接口有串行、并行和视频三种。对于串行接口,有一个软件定义接口的问题。这个问题解决不好,就会影响激光打印机的使用。定义软件接口的目的是使计算机可识别激光机,从而将输出改向至激光打印机,而不是传统的针打机。

本文以HP2686A型激光打印机与IBM/XT机为例,阐述激光打印机与计算机的软件接口的定义方法,供用户参考。

定义软件接口之前,必须将硬件接口联接好。首先检查接口电缆的型号是否与说明书一致。然后,将电缆与主机背板上的异步通讯适配器相连。如果主机上配有两个异步通讯适配器,一个是基本异步适配器,另一个是备用异步通讯适配器。联接时,要确保接口电缆插头插在正确的适配器上。因为这牵涉到软件接口参数的定义。一般都是与基本异步通讯适配器相连接。

定义软件接口的方法是:

①在PC-DOS(或与其兼容的汉字系统下)

C>copy con: AUTOEXEC.BAT✓

② mode com1: 9600,N,8,1,P✓

③ mode LPT1: =com1✓

④ 按 

c+rl
------

 + 

z
---

⑤ ✓

这样,通过建立批命令文件的方式就定义好了软件接口。通过行编辑程序EDLIN也可建立批命令文件。

对上述定义的解释如下。

“mode”是DOS中的一个外部型命令。其用途是建立在打印机上或者同彩色/图形监控连接器相连的显示器的操作方式

上,对异步通讯连接器设置任选项,或者使打印机输出改向到异步通讯连接器。

“COM1”指第一个异步通讯连接器,即基本异步通讯适配器。

“9600”指波特率。一般激光打印机的波特率有300,600,1200,2400,4800,9600,19200不等。波特率表示每秒钟传递的信息位的数量。它是所传送代码的最短码元占有时间的倒数。例如一个代码的最短信号码元宽度为20毫秒,则其波特率为每秒50波特。一般来说,波特率在出厂时就选定了。当然,用户也可根据需要改变波特率。

“N”指不需奇偶校验。如果写“O”,则是指奇校验。如果写“E”,则指偶校验,缺省时为E。

“8”指数字位为8,缺省时为7。

“1”指停止位为1,这个停止位只有两个状态,不是1就是2。如果波特率=110,缺省值为2;如果波特率=110,缺省值为1。

上述几个参数都是通讯协议的参数,它们被用来初始化异步通讯连接器。用户规定通讯协议时,必须规定波特率。其它参数可视情况而忽略或只输入逗号。

“P”指异步通讯连接器被用于串行接口的打印机。

批文件中的第二命令行:mode LPT1:=com1,其意义是把指向第1号打印设备的全部输出改向到异步通讯连接器1上。

在定义软件接口时,要注意,当使用mode方式把并行打印机输出改向到一个串行设备之前,必须用第一句命令语句初始化异步通讯连接器。串行的初始化命令必须

(下转第48页)

# OAS/724办公室自动化系统的开发步骤与方法

徐宝文

(中国舰船研究院南京船舶雷达研究所)

## 一、引言

OAS/724是我们正在为本所研制的一套办公室自动化系统。本系统由人事档案管理子系统、工资财务管理子系统、计划决策管理子系统、报表文件管理子系统、资产管理子系统等多个子系统组成。本文根据现有的工作,初步介绍OAS/724系统的性能要求与设计原则,开发步骤与开发方法。

## 二、系统性能要求 与设计原则

### 1. 可靠性

可靠性是设计本系统的首要指标。在确保系统可靠性方面,我们主要作了两方面的工作,一是强化系统本身的可靠性,体现在系统开发过程中就是要使用优良的开发方法与开发工具;二是使系统在硬件、软件两方面都具备一定的容错、纠错能力,如在程序员不慎发出错误的操作命令或输入错误的数据时,能部分或全部检测出来。

### 2. 通用性

本系统设计兼顾特殊性与一般性,尽量概括出各种用户的“共性”,把与具体用户有关的表格结构等放在系统设计的最后阶段考虑。这样,当系统服务于不同的用户时,只须在系统的最外层作些修改。

### 3. 实用性

考虑到用户文化程度不一,专业方向不一,对计算机的了解程度不一等各种实际情况,我们在设计系统时尽量兼顾各种用户的需要。对一般用户,要选择单位一步、一步地

进行引导;对有一定计算机知识的用户,可以使用各种命令(及其组合)对系统进行操作。

### 4. 灵活性、可变性与可扩充性

有两种情况要求系统具备这几个性能:一是系统服务于不同水平、不同职业、不同要求的用户时,用户接口要作些变化;二是由于情况改变,系统的服务项目需要作些增添修改,扩充的工作量应尽可能做到更少,不至于重新研制整个系统。

### 5. 集成性与可嵌入性

每个子系统都是整个系统的一部分,各子系统基本上是由不同的研制小组研制的。因此,既必须全面协调各子系统的用户接口(主要是统一好信息的存贮格式),同时兼顾与全院计算机网络的协调,使得本系统在基本上不作修改或不作大修改的情况下能嵌入到院计算机网络中。

## 三、开发步骤

由于人力、财力及其它各条件的限制,OAS/724系统的开发工作分作两步:

第一步,利用现有的微机系统(长城0520、TRS80、CROMEMCO机等)先行开发各个子系统。在功能、实现等方面各子系统之间又有某些内在联系,如人事档案子系统与工资财务子系统、工资财务子系统与资产子系统之间都有一些联系,因此,在各子系统的总体结构设计上又要兼顾这些联系,保证各子系统之间相同或相关信息项有相同或相容的字段格式,以利于将来的扩充,也有利于各子系统之间信息的共享。

第二步,在已开发的各个子系统的基础

上, 开发完整的OAS/724系统(同时可开发其它子系统)。OAS/724系统是把各个子系统有机地结合起来的产物。我们对各子系统的连接方式设想了两种方案。

第一种方案是把多台微型机用某种网络形式连接起来, 构成分布式或多机系统, 各个微机都处于完全平等的地位。

第二种方案是建立主计算机网络系统。我所有一台VAX11/780机, 可以把这台机器作为主计算机, 把长城1520等微机及其它智能终端与主机连接起来, 作为主机的智能终端设备。终端可以是专用的也可以是通用的。主计算机对终端微机发来的命令信息等进行收集、分析、分类、归档、存贮、检索、复制、删除。每个终端可以有自己常用的小型数据库, 以避免出现对主计算机的频繁访问, 当然, 这些小型库在主计算机中都必须有相应的副本。在OAS/724系统实现时, 我们初步考虑采用这种方案。好处是: 便于把它向更高一级的网络连接。

## 四、开发方法

### 1. 系统分析与软件计划、要求、分析

系统分析是OAS/724开发的第一步工作。在这一阶段, 首先对全所(抑或也包括对其它院、所、校、厂)机关、科室各部门的职能进行调研, 也要对全所将来各种机构的变化与职能的改变作出必要的、准确的预测。研究各部门对系统的功能与性能要求。在此基础上, 对系统进行初步设计分析。其后, 对初步系统进行可行性研究。从理论上把系统做得尽善尽美, 但我们又不能受人力、技术水平、物质条件、时间等的限制。必须根据这些条件对初步系统作出客观的分析、评价。有些难以办到的而又对整个系统影响不大的可以不实现, 有些难以办到的则可以分步实现。

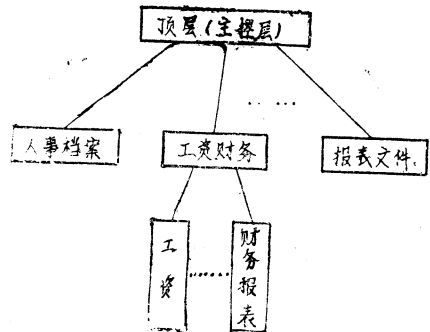
系统分析与软件要求分析均要牵涉到规

范技术、工具与方法论等多个方面。要求分析的结构必须文档化、规格化。文档应具备完整性、可测试性、灵活性(也称柔软性)等多种特性。目前, 有多种要求分析工具可以使用, 如PSL/PSA、HIPO、ISDOS等。要求分析可作为软件工程的第一步, 也可以把它贯穿于软件工程的全过程。在软件生命周期中, 每一阶段的工作都根据前一阶段的要求进行。当然第一种要求对系统的影响最大。我们考虑采用后一种技术要求。

### 2. 软件结构

软件结构反映要求定义分析所隐含定义的问题的软件解的各个元素(即模块)之间的联系。用于表示程序的体系结构的软件结构仅仅意味着各软件元素的层次控制关系, 并不表示各模块开发过程的先后, 侧重于功能而忽视实现。

对同一问题可以有多种软件结构。软件结构的形式影响到软件产品的质量、可测试性、可移植性、可维护性与可扩充性。我们用自顶向下的方法设计软件结构, 这样设计出的软件结构可满足很多性能要求。OAS/724的软件结构如下图所示



OAS/724软件结构

### 3. 设计方法与工具

软件要求定义了系统的基本功能, 软件结构确定了软件产品的基本构成元素。在设计阶段, 要对这些基本元素作出更进一步的设计(并非详细设计), 把“非过程性”的软件结构轻化成“过程性”的控制结构。

软件模块化对于大中型软件（的开发）是极为重要的。模块化软件有许多优点。从我们过去开发系统软件的经验看，划分模块有两种指导原则。一是以数据为中心来划分模块，这样划分的模块之间的接口关系简单，但功能分割不明显。另一种是以功能为中心划分模块，这种模块的优点是前一种的缺点。由于OAS/724的各个子系统是提供各种用户使用的，各子系统之间接口信息很少；对一具体的用户，他也不会同一时间作多种工作，因此可以把功能作为划分模块的准则。这样划分对OAS/724来讲也不会增加多少模块间的接口信息，因为不管如何划分，都要以一个（组）公共数据库（数据集）作为各模块的公共处理对象或公共数据接口。

在开发过程中，我们既进行自顶向下的开发，又进行由底向上的开发，对整个OAS/724系统，我们先开发各个子系统，然后再开发全系统，这是由底向上的设计；对特定子系统，我们先确定它的功能，进行总控设计，然后再逐步设计完成整个系统，这是由顶向下的设计。

我们选用框图与PDL作为由设计工具。设计的产品（中间产品，不是指最终软件产品）必须存档。

有两类编码语言可供使用，一种是普通程序设计语言（如Pascal、C、BASIC、Ada等），另一种是数据库查询语言（如dBASEⅡ、dBASEⅢ等）。在人事档案子系统的程序编制中，我们采用了dBASEⅢ数据库查询语言。从目前的使用情况看，该语言有很多缺点，如限制太多、速度慢。

#### 4. 人员组织

如何组织开发人员对软件的质量影响很大，根据软件工程原理，以往从事开发工作的经验以及参加开发工作的人员素质等情况，我们拟采用主程序组（chief programmer term）的人员组织形式。组中包含

四类开发人员：主程序员一人，负责全面设计，并分配、审核组中各程序员的程序；辅助设计员一人，协助主设计员作整体工作；程序资料管理员一人，把开发过程中的程序设计与测试状态记录在开发支撑库中；普通程序员数人，完成主（辅）程序员规定的各种程序段（模块）的设计与编码工作。这种组织形式不仅可以弥补开发人员间软件素质的差异，也与树型模块化的程序结构相协调，同时可以减少各个程序员之间的接口。

按功能分割的模块先分配给各个“逻辑程序员”，然后再视各“逻辑程序员”工作的轻重缓急与实际程序员（物理程序员）的实际水平，可以由一个实际程序员完成一个“逻辑程序员”的工作，也可以由几个实际程序员完成一个“逻辑程序员”的工作，也可以由一个实际程序员完成数个“逻辑程序员”的工作。这样可以避免“按人设事”方式中所出现的那些过多、过复杂的接口与扯皮现象。

#### 参考文献（略）

（上接第36页）

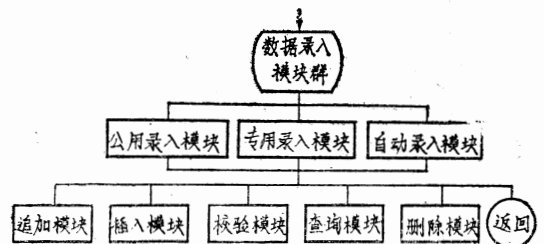


图2 数据录入模块群

良好，特别是在提高管理水平和效率方面取得了明显的效果。

1985年年底，用户使用该系统对全省化工产品信息进行了处理。烧碱、纯碱、橡胶、硫酸和轮胎等物资都分别、分批进行必要的预算、核算、平衡和统计及制表。总计输出表格近25余张，字符数达十二万之多。

由于本系统使用灵活、方便，它可供各级领导随时查询有关信息，以便对全省的化工产品的规划，平衡与分配作出科学性的决策。



# 化工信息管理系统HUAGONG

朱容村 张泽斌

(河北省计委计算中心)

## 一、前言

烧碱、纯碱、橡胶、硫酸和轮胎等,在整个国民经济计划中是不可缺少的化工产品,它们的需要、分配与平衡对国民经济的巩固和发展有着很大的影响。这些产品的供求状况如何,应当怎样分配,如何平衡安排,怎样科学管理,多年来,计统人员一直采用的是原始落后的管理方法,沿用的是表格算盘加经验的传统平衡表法。速度慢,效率低,科学性差,致使每年的产品需求、平衡与分配,至少需要两个月的时间才能编制完成。而且当手工计算出错或数字有所变更时,又需要重新分配与平衡,延误时间。特别是核算和预算,由于计算量大且关系错综复杂,计统人员常为此感到头疼。为了减轻计统人员的劳动强度,实现化工产品信息的科学管理,我们在最终用户的配合下,用了近三个月的时间,在微机上研制成功了化工信息管理系统——HUAGONG。

## 二、HUAGONG系统结构

本系统共管理五个数据库和十五个子数据库,即烧碱、纯碱、橡胶、硫酸和轮胎主库,每个主库包括需要、平衡、分配3个子库。

## 三、程序结构

HUAGONG系统共由二十四个功能

模块组成,它在软件设计上采取了层次化,结构化,模块化,屏幕格式等设计方法。模块与模块之间既界面清楚、又紧密相连。每个数据库子系统不仅具有修改、增加、删除、插入、录入、核算及信息查询、统计制表等若干功能,且其数据信息还能作为原始数据与预测方法,经济模型及其它算法程序相连接。程序结构见图1。

为了达到信息需求屏幕化,用户操作简单化,本系统在程序实现上采用了多级菜单控制的办法,即用1~5级菜单处理及多重菜单嵌套来选择各种功能以满足用户的需要。二级菜单控制的部分程序和三级菜单的实现程序如下:

二级菜单部分控制程序:

B>

ERASE

SET FORMAT TO SCREEN

□ 1,28 SAY " 产品数据库选择屏 "

□ 2,28 SAY " - - - - - "

□ 3,28 SAY "0. 烧碱数据库"

□ 4,28 SAY "1. 烧碱数据库"

□ 5,28 SAY "2. 纯碱数据库"

□ 6,28 SAY "3. 橡胶数据库"

□ 7,28 SAY "4. 硫酸数据库"

□ 8,28 SAY "5. 轮胎数据库"

□ 9, 0 SAY "请按所需代码(0—5)"

SET CONSOL OFF

WAIT TO ACTION

SET CONSOL ON

STORE! (ACTION) TO C



□ 03,29 SAY "1. 数据显示"

□ 4,29 SAY "2. 项目查询"

□ 5,29 SAY "3. 数据录入"

□ 6,29 SAY "4. 数据核算"

□ 7,29 SAY "5. 统计制表"

□ 8,29 SAY "6. 复制结构"

□ 9,29 SAY "7. 预测分析"

□ 10,0 SAY "请选择(0-7)"

SET CONSOL OFF

WAIT TO T1

SET CONSOL ON

STORE "PROG" + "&C" + "&T0"  
+ "&T1" TO PROG

DO &PROG

RETURN

## 四、系统的主要特点

### 1. 系统性

按照统计人员的要求分析和设计整个系统。步骤明确，结构紧凑，层次清楚，主控系统所包括的各种功能模块界线分明，接口清楚，既能独立地实现各自的功能，又通过共享的数据库，实现有机的联系，构成一个整体。

### 2. 先进性

- 在软件技术上，使用了分层结构设计方法，便于修改、维护或移植。

- 为了既加快查询速度又不致花费过多的内存空间，让主控模块，常用公共子模块和查询模块常驻内存，而对各独立的功能模块和其它辅助程序模块采用分块载入内存和程序复盖链接技术。

- 本系统完全模拟用户的思维过程，用主菜单，分菜单处理及多重菜单嵌套来选择所需要的各种功能以满足用户的不同需要。每级菜单的各功能选择均以0~9个数字键代表，通过按数字键来控制系统的整个工作流程，从而形成所谓“快餐式”的菜单操作方法。

### 3. 系统功能齐全

由于该系统在研制过程中，始终与统计人员一起进行系统分析和功能设计，所以功能齐全，除了用户所需要的各种功能外，还设有经济分析，预测方法及数学模型等其它功能，为统计人员进行各类分析提供方便。

### 4. 操作简单，易学易用

由于本系统采用了全过程的汉字菜单控制，用户在操作使用过程中，只须根据屏幕提示按相应数字键即可，不必经过特殊培训就能直接操作使用该系统。

### 5. 查询计算快速无误

本系统能快速查询或计算某一产品的各类数据，可根据需要进行汇总计算或分类排序。检索迅速，计算准确无误。

### 6. 安全保密

为了保证信息的保密性与安全性，本系统设置了“查询权限”和“修改权限”两个密级，只有持此权限的用户才能打开数据库进行相应的操作。

### 7. 通用性

系统研制过程中，考虑了同类部门的业务特点，系统着眼于通用性，易于在省、地、市、县各级计委部门中应用推广。另外还具有较强的可扩充性。

### 8. 数据维护方便

能方便地对各类数据进行录入、修改、插入、删除、校验等操作。其中，有多个录入模块供选择，如图2所示：数据录入公用模块不仅适合于本系统的数据录入，而且还能应用于许多类似的子系统；专用录入模块仅适合于本系统，而且在录入时，填参少，速度快；自动录入模块是根据需要从已有的其它数据中取出部分来，填充到当前的数据中。

## 五、系统应用效果

本系统于1985年下半年投入使用，效果

(下转33页)

# 在IBM PC/XT屏幕上绘制趋势图和相关图

刘 莉 明

(天津商学院)

在预测中,人们要根据历史数据序列或因果关系数据序列绘制出动态变化的趋势图或检验相关程度的相关图,从而确定采用什么样的定量预测方法,因此绘制图形在预测中占有相当重要的地位。笔者在编制“市场预测软件包”时,用BASIC A编制了一个在屏幕上绘制X-Y图形子程序。运用这程序,在IBMPC/XT 微机上,可以根据所给的任意两组数据,自动地按数据中的最大值和最小值计算出X、Y轴坐标值和间隔。然后画出格线,标上坐标值,自动在屏幕上绘制出趋势图和相关图。还可以在同一坐标区绘制出多条曲线,从而为预测提供了有力的工具。

## 一、程序设计

1.为了使图形取的点多,曲线平滑度好,则选用了高分辨率图形显示模式:SCREEN2,每屏200线,每线640点。

2.在图形中加有中文标题,系统采用CCDOS。

3.在SCREEN2及中文状态显示情况下:

(1) X轴最多包括500点,划分为50个小格,10个大格,一个大格包括5个小格。

(2) Y轴最多包括120点,划分为24个小格,4个大格,一个大格包括7个小格,一个小格包括5个点。

这样划分可以配合LOCATE 语句和PRINT语句,使标注的坐标值能对准格线。

4.轴坐标间隔值的计算:

X轴坐标间隔值:

$$X_D = \left( \frac{X_{大} - X_{小}}{5} \right) \times 5$$

Y轴坐标间隔值:

$$Y_D = \left( \frac{Y_{大} - Y_{小}}{24} \right) \times 7$$

5.在显示屏上画曲线:

先计算出各点的坐标值:

$A_i(C_i, D_i)$  (i从1—数据总个数)

$$C_i = 50 + (X_i - X_0) \div \left( \left( \frac{X_{大} - X_0}{50} \right) \div 10 \right)$$

$$D_i = 140 + (Y_i - Y_0) \div \left( \left( \frac{Y_{大} - Y_0}{24} \right) \div 5 \right)$$

其中 $X_0$ 和 $Y_0$ 为原点值(为最小值)。

计算出每点 $X_i$ 和 $Y_i$ 值,即可用LINE语句将点连起来,画出曲线。

## 二、程序框图和使用例子

### 1.程序框图见图1

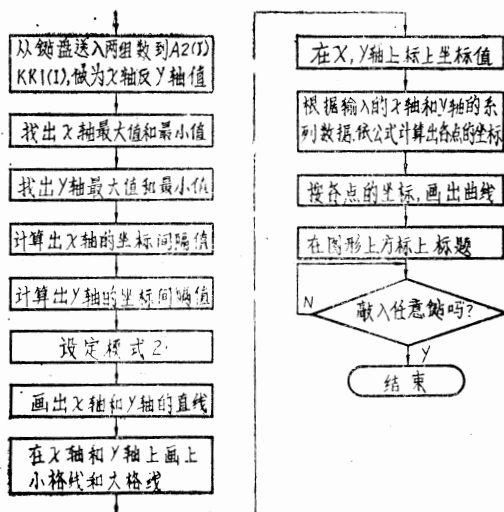


图1 程序框图

(下转第9页)

# 计算机服装辅助设计系统的应用研究

## 章 兵

(安徽省蚌埠市二轻局计算机室)

〔摘要〕计算机服装辅助设计——是当前计算机应用中发展十分迅速的一项新技术。从实用效果看,有较高的经济效益;可以改变服装工业落后的生产面貌,其社会效益也不可估量。

本文对系统的配置、功能特点、操作步骤作了简介;着重对系统的理论数据模型和技术处理方法进行了解剖和分析;另外,对系统硬件改造情况也作了介绍。

### 一、前言

一九八五年十月,我局从国外引进了一套计算机服装辅助设计系统:CA—30 系统。

我们对该系统进行了技术剖析,并对有关硬件进行了改造,使该系统在短时间内经过安装、调试便使用于出口服装的生产线上。从实用效果看:缩短了服装设计和生产周期,使产品更新投产快,生产效率,节约面料,减轻了设计人员的劳动强度,有较高的间接经济效益。

该系统硬件经改造后,使这台国外专机专用设备与通行的IBM PC 机兼容,实现了一机多用。目前,该系统除了用在服装CAD方面外,已经应用在现代化管理领域。

### 二、系统概述

#### 1. 服装CAD的一般处理方法

服装CAD处理方法的一般特点是:以某些经验公式为依据,在计算机内建立二维图形的数学模型,然后按用户的要求,进行服装设计、推档、排料等一系列操作。为了提高服装CAD系统的灵活性和适应性,通常要考虑以下几个方面:

(1) 建立一个能快速修改、用户控制方便的服装基样数学模型。

(2) 建立一套能适应多种处理、满足计算机时空要求的图形和数据结构。

(3) 由于服装CAD实质上是模拟人的设计活动,因而人机交互始终是设计核心。

#### 2. 系统配置

##### (1) 硬件

主机:EVERGO(爱美高)—PC88微型计算机。

内存:640k字节RAM

外存:一个软盘驱动器加一个硬盘驱动器。

CRT:高清晰度图形显示器(740×400线)。

数字化仪:MAX MD—A40绘图仪和鼠标器(定位器)。

选配件:3070宽行24针打印机和光笔。

系统配置如图1所示:

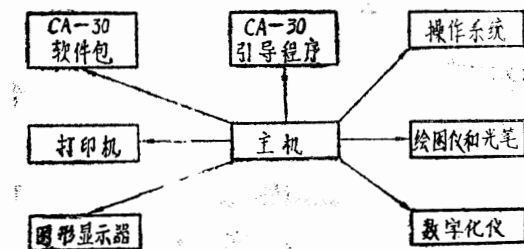


图1 系统配置

##### (2) 软件

CA—30软件包,用8088汇编语言编制,固化在ROM内,做成一块硬卡插在主机扩展槽上,使用过程中,由CA—30引导程序调用。

该软件包特点为:

1)为了在计算机内有效地描述平面几何图形,采用了分层环形的数据结构,将图形中逻辑上有关的元素分成若干层次。然后在横向和纵向分别把它们勾连起来。

2)该软件在逻辑上分三个层次:控制部分,基

本功能部分和服务程序部分。

3)程序采用了详尽的菜单提示供选择。

4)为了解决和外设异步通讯的不匹配问题,采用了陷井技术。

### 3.系统主要功能特点

主要包括以下两大部分:

(1)借助于交互式外部设备,通过人机对话的

方式,可以实现:

1)服装基样设计和结构造型。

2)服装款式的动态修改。

3)服装基样的推档(放码)。

4)服装用料面积计算。

绘出的衣片图形曲线光滑,规格准确。

(2)系统设有服装图形库,能贮存大量的服装和衣片图形,用户通过键盘和数字化仪能方便地进行嵌套组合及修改,从而开阔了设计人员的思路。

## 三、CA—30系统的数学模

## 型和技术处理方法

### 1.服装原型设计的概念

服装原型在服装设计中占有特别重要的位置。所谓服装原型,是指人体各部分形状所展示的平面图形。它准确地体现了人体各部分的比例关系。它是研究服装和人体关系的依据。

使用服装CAD系统设计时,服装原型采用三个参数控制,即胸围、背长、放松量。只要用户输入这三个数值,屏幕上就显示出所要的服装原型。用户可以参考服装原型和输入其他款式的有关信息,由数字化仪的定位器在屏幕上直接绘出服装基样,也可在原型的基础上进行省道转换和变化。

服装上的省道是实现服装款式变化的重要手段,如图2所示。省道转换时,人体的体型和设计的款式,收省的大小和位置是不相同的,基本要求是省道转换前后,服装曲线的形状和长度不发生变化。

### 2.服装基样的设计和造型

服装基样或衣片图形是由各种曲线和直线围成的,形状很不规则。用计算机构造这类形状,采用了两种形状模型化的方法:一种是“插值”的方法。它的目的是对形状进行精确地描述。当已知曲线或曲面上有多个型值点时,往往用这种方法;另

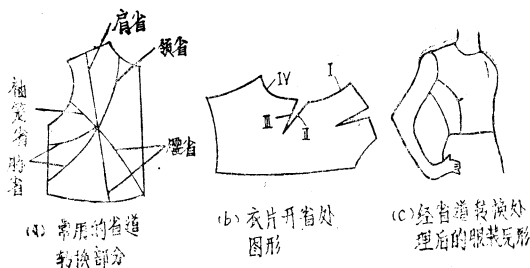


图2 省道设计示意图

一种是“逼近”的方法。它是以人机对话的方式不断地修改形状,使形状步步逼近设计者所给定的要求。

用计算机进行服装基样设计,是由设计人员向计算机输入胸围、衣长、袖长等参数,由计算机根据经验公式,算出服装基样上各部分的尺寸,找出基样上各条线段的型值点。(对曲线部分,则要算出若干个型值点)然后由计算机对这些点进行曲线拟合。

在设计一个新款式服装时,特别是对服装基样进行结构造型时,由于基样上型值点的自由度很大,这就要采用“逼近”的方法,即不断地修改(在绘图仪上或使用光笔)服装基样,以使曲线的形状和长度达到给定的要求。用计算机完成这项操作,涉及到描述形状的数学模型,达到的效果是:一方面计算机的几何模型化系统要向设计者提供准确逼真的形状信息;另一方面设计者只要通过修改少量几个参数,就能有效地直观地控制和修改服装形状。

### 3.CA—30系统的主要数学模型——B样条

系统采用B样条作为服装的衣片设计、推档、结构造型、动态修改、用料面积计算的主要数学模型。

如果给定了 $m+n+1$ 个控制点, $Q_k(k=0,1,\dots,m+n)$ 我们称: $P_{2,n}(t)=\sum_{i=0}^n Q_{i+1}N_{i,n}(t)$ , $0\leq t\leq 1\dots\dots$ (1式)为n次B样条曲线的第i段( $i=0,1,\dots,m$ ),其中, $N_{i,n}(t)$ 称为B样条基。

$$N_{i,n}(t)=\frac{1}{n!}\sum_{j=0}^{n-1}(-1)^j C_{n+1}^j (t+n-1-j)^n$$

$$(i=0,1,\dots,n) \quad 0\leq t\leq 1 \quad (2式)$$

如果用线段依次连接 $Q_{i+1}(i=0,1,\dots,n)$ 中

相邻二向量终点,得到的多边形称B样条的特征多边形。

在CA-30系统中,主要采用经过某些处理的二次和三次B样条,从(1)和(2)式可以看出:B样条上的点都与控制点Q<sub>i</sub>有关,即曲线上的点是特征多边形顶点的加权向量和,权因子就是B样条基。这说明特征多边形的形状决定了B样条的形状,通过控制B样条特征多边形顶点,可以达到控制曲线形状和长度的目的。其示意图如图3所示。

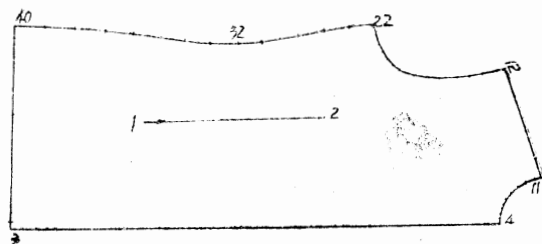


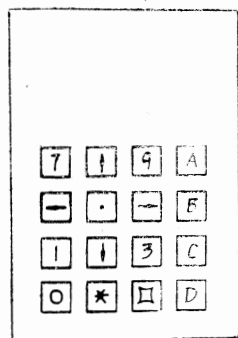
图3 衣片曲线上控制点示意图

#### 4. 动态修改服装曲线的方法

CA-30系统可以借助于键盘和数字化仪上的定位器任意选择修改点,直观地控制服装曲线的形状和长度变化。

(1) 用定位器动态修改的步骤如下:

1) 在数字化仪上拖动定位器,选取服装图形的控制点(即特征多边形的顶点)。定位器和其功能示意图如图4所示。



A—输入完毕  
B、C、D—控制点类别  
B—圆弧  
C—直线  
D—钮门  
0—9: 操作规格号码输入  
1—删除某控制点  
\*—输入某控制点

图4 定位器及功能示意图

2) 控制点选中后,系统发出选中信号(响声)。控制点的个数和位置确定后,屏幕上就显示出服装图形。

3) 动态修改时,可用定位器在曲线上确定新的控制点,或加插和删除有关控制点,随后修改后的图形便在屏幕上显示出来,用户可做进一步修改,直到形状和尺寸都满意为止。

(2) 使用键盘进行动态修改的方法如下:

用户首先在屏幕上查阅各控制点的位置(按序号排列);在各点之间通过键盘加插和删除新的控制点,其结果如同上述3)。

#### 5. 服装衣片的推档(放码)

一种款式的服装设计定型后,投入生产需要好多不同的规格,这是在服装基样的基础上,用推档的手段来实现的(见图5)。CA-30系统允许用户完成18种规格的推档,足以满足服装生产的需要。

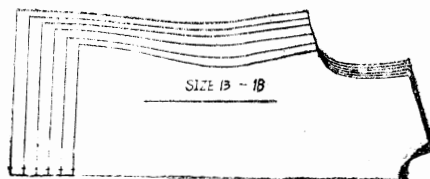


图5 衣片推档示意图

必须注意的是:服装基样已定,但经推档操作后,往往在袖笼和袖山处不能合适地配合,这是因为该处曲线弧度比较特殊,而选取控制点较少时,使该处推档后点距离扩大,产生了曲线弧度畸变的原因所至。这就需要用户在基样设计时,对于弧线处选取控制点的个数,摸索一个经验数据,这项工作也可以在推档操作中,通过动态修改的方法来完成。

具体推档操作过程如下:

用户选择进入推档状态后,屏幕上除显示服装图形和曲线上各控制点外,同时列出点距数目和X、Y方向推档参量供选择,如图6所示。

单位: 厘米		
**** 推档进幅 ****		
点距数目	X	Y
0-1	0.1	0.1
1-2	0.5	0.2
2-3	1.1	0.3
:	:	:

图6 推档操作示意图



说明: 0—1点曲线不推档(按原设计尺寸)。

1—2点曲线在X方向放码0.5厘米, Y方向放码0.2厘米;

2—3点曲线在X方向放码1厘米, Y方向放码0.3厘米。

因量度单位是预定的, 只要输入用点距表示的曲线各部位推档数值, 随后可由屏幕显示推档操作后的图形, 如图5, 或由绘图仪输出。

## 6. 用料面积计算和文件处理功能

CA—30系统的操作和服装图形库的管理都是以文件管理的方式进行的。文件名的取法是随意的, 但一件服装的各衣片取名应一致, 以后加特定的数字来区分。

这样在用户选择执行用料面积计算功能时, 系统自动把该类文件集中调用计算, 随后显示或打印输出经计算所得的各码用料面积和该款式服装用料的总面积。如果用户回答面料宽度后, 系统自动给出最佳用料长度及使用效率等结果。如图7所示。

单位: 厘米

面积计算

输入文件名: COAT PLO✓

尺码	13	14	15	16	17	18	数量
前幅 (COAT1)	252.0	276.0	370.0	334.0	359.0	366.0	1
后幅 (COAT2)	270.0	295.0	325.0	354.0	380.0	407.0	1
袖片 (COAT3)	152.0	167.0	175.0	191.0	197.0	211.0	1
领 (COAT4)	56.0	58.0	60.0	62.0	63.0	65.0	1
肩 (COAT5)	62.0	63.0	65.0	67.0	68.0	70.0	1
各码面积:	792.0	959.0	933.0	1008.0	1067.0	1139.0	1
用料总面积:	5798.0						
料宽 = 60	✓						
用料长度:	100.07						
用料效率:	85.4(%)						

图7 面积计算功能示意图

## 四、系统使用方法

系统操作, 采用了详尽的菜单选择执行方式, 使用户的操作都处在屏幕提示的引导之下, 这就保证了使用的方便性和灵活性。系统操作流程如图8所示。

## 五、系统硬件改造情况介绍

在设备验收过程中, 发现主机与 IBMPC 机型不兼容, 是属于服装CAD方面的专机专用设备。具体表现在PC—DOS或CC—DOS都引导不成功。根据生产实际情况, 由于提高了工作效率, 服装CAD使用占机时间并不太长。这样设备在其他时间闲置, 使用率不高。

我们在武汉外设所的协助下, 对系统硬件进行

了分析, 找出了不兼容的主要原因有二处:

1. 驻留在系统板上的基本输入/输出系统ROM—BIOS, 除了包括加电自控程序, 软盘自举装入程序, 字符点阵图形, I/O驱动程序外, 而主要是对系统中的I/O设备提供一级控制。从CA—30系统ROM—BIOS程序清单上分析, 主要区别是修改了有关系统调用指针; CA—30引导盘启动以后, 系统调用指针便指向扩展槽上特定的槽号, 把CA—30软件包自动调入内存执行。

2. 为了显示高质量的服装图形, CA—30系统配置了740×400线高分辨率图形板, 而没有设置图形向下兼容的软硬件接口, 这与IBM—PC标准的图形处理方式(640×200线)有所不同。

在硬件改造过程中采取的相应技术措施有:

• 针对图形方式不兼容的情况, 考虑到在保持图形板原功能状态下, 实现高分辨向下兼容, 涉及面比较广(要改写ROM—BIOS内容和图形板处

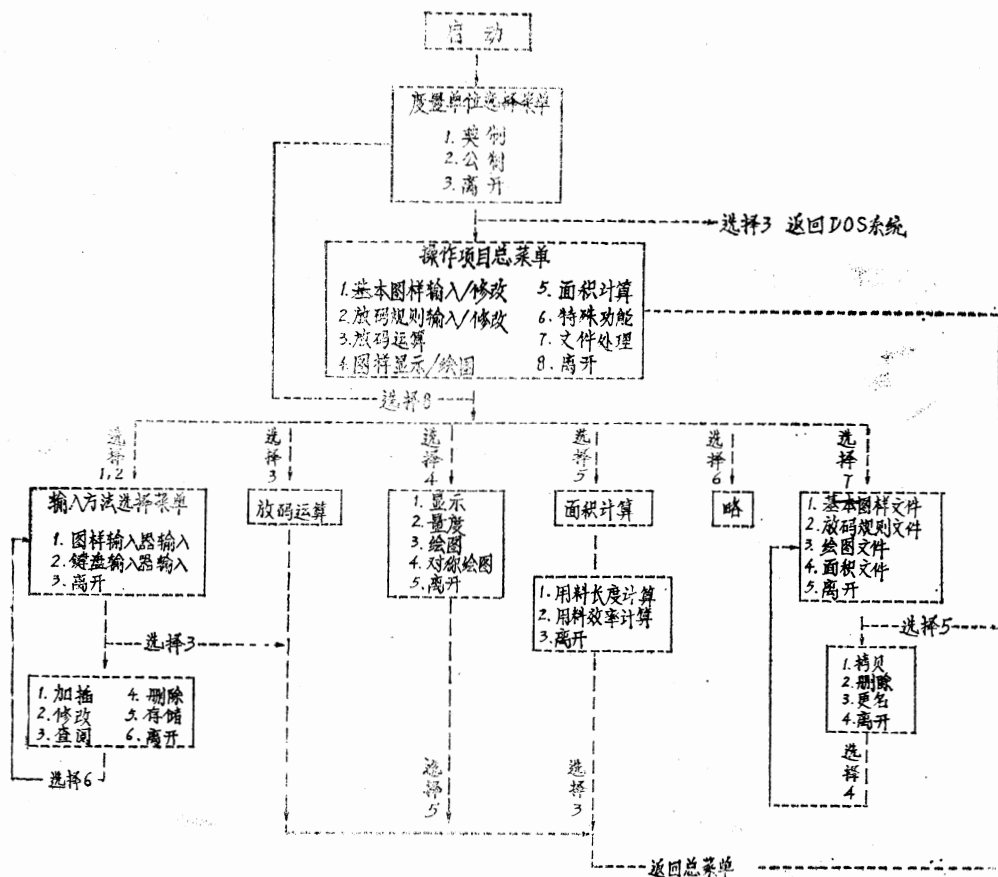


图8 CA—30系统操作流程框图

理方式)，搞不好会使CAD的图形方式受到破坏。于是我们在扩展槽上插上了一块IBM-PC标准图形板，使系统图形状态处于双通道工作方式，用于不同的场合只要改换一下图形板的显示器接口即可。

• CA—30专用机的ROM—BIOS内容有特殊的要求：既要保持原功能不变，又要实现与PC机兼容，两者往往很难兼顾。后经查询，了解到湖北电子开发公司研制的HED型中文卡，除了包括24×24点阵的国标字库外，还有改写后的ROM—CCBIOS，这样恰好满足了我们的要求。

改造后CC—DOS的引导过程如下：

系统加电自检后，首先寻找 AUTOEXEC. BAT 这个批文件，这个批文件要调用两个可执行的文件FILE1和CCCC。

FILE1.EXE文件的主要功能是检查驻留在盘上的CCLIB的完整性，并向系统为其申请内存空间。

CCCC.EXE程序主要完成将汉字库全部装入由FIFE1所申请的内存之中，并修改有关系统调用指针，使其进行I/O时，不再执行原ROM—BIOS中的中断管理程序，而去执行修改后的CC—BIOS中的有关处理程序。

参考文献(略)

# 一种控制器用的专用微机

汪仁煌

(湖南大学电气工程系)

## 一、概述

通过研制控制器的使用要求,采用参考设计法和负载计算法设计了一个控制器用的专用微机(见下图)。图中的专用微机系统具有2KROM、2个输入口(74LS244)和二个输出口(74LS273),具有加电和手动复位功能,无中断功能。

系统的设计是合理的,根据参考设计法,采用Z80单板机中所用的2716EPROM作专用微机的ROM,采用LSTTL系列的简单I/O口器件作输入口或输出口,故专用微机系统所用的器件满足Z80 CPU的时序要求;根据负载计算CPU可以驱动 EP-ROM 和I/O接口器件。

## 二、I/O特点

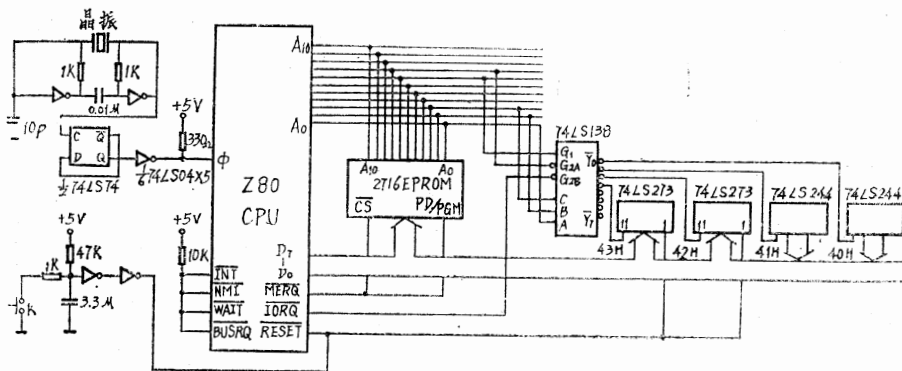
既然是专用微机,一般I/O口的定义很确切,不是用输入口,就是用输出口。由于不

要求中断功能,没有必要采用PIO口,可避免不必要的设置PIO工作状态的附加程序设计;在此采用LSTTL电路的简单I/O接口,为满足三态总线的要求,输入口采用8位3态门74LS244,输出口采用8位锁存器74LS273,此种器件成本低,驱动能力大。

在地址译码方法中,为简化电路结构,考虑到I/O扩展的方便性,采用不完全地址译码。图中I/O口译码器74LS138的使能端采用地址信号 $\overline{A}_7$ 、 $A_6$ 和控制信号 $\overline{IORQ}$ 作控制。由于采用不完全译码,在 $\overline{IORQ}$ 有效时,对应于译码器的输出 $\overline{Y}_0 \sim \overline{Y}_7$ 中每个片选信号有8个地址可使用。在此令 $A_5 = A_4 = A_3 = 0$ 时,则可得到表中所示的地址码。

表1 片选信号对应的地址码

片选信号	$\overline{Y}_0$	$\overline{Y}_1$	$\overline{Y}_2$	$\overline{Y}_3$	$\overline{Y}_4$	$\overline{Y}_5$	$\overline{Y}_6$	$\overline{Y}_7$
地址码(H)	40	41	42	43	44	45	46	47



专用微机系统接线图

一般情况,  $\overline{\text{IORQ}}$  信号有效只表示 CPU 执行 I/O 口操作, 它并不表示数据线中数据流的方向。这样, 片选信号是由不带方向的控制信号  $\overline{\text{IORQ}}$  与地址译码信号组合的信号, 这种片选信号既可用作输入口选通, 也可用作输出口选通。如  $\overline{\text{Y}}_0$  用作输入口的片选信号时, 执行  $\text{IN } 40\text{H}$ , 即可把数据从输入口送到 CPU 的累加器 A; 如  $\overline{\text{Y}}_0$  用作输出口的片选信号时, 执行  $\text{OUT } 40\text{H}$ , 即把累加器 A 的数据送到输出口。不管执行输入指令或输出指令, 由于  $\overline{\text{IORQ}}$  均有效, 都会在译码器 74LS138 的输出端 (地址码所对应的输出端) 中送出一个有效的片选信号。译码器输出  $\overline{\text{Y}}_0 \sim \overline{\text{Y}}_7$  中的任一个只可接一个输入口或一个输出口,  $\overline{\text{Y}}_0 \sim \overline{\text{Y}}_7$  共有 8 个片选信号, 这些片选信号既可用作 8 个输入口的选通; 也可作 8 个输出口选通; 或是用作既有输入口、又有输出口, 其总数不大于 8 个的 I/O 口选通。图中的片选信号有多余, 故扩充 I/O 很方便。考虑到 CPU 的数据总线的驱动能力, 则可接的最多 I/O 口见表 2 所示。由于译码器输出的片选信号中控制信号为  $\overline{\text{IORQ}}$ , 其片选信号控制灵活, 可根据控制器的需要按表 2 规定, 配置合乎要求的 I/O。

表2 可接的 I/O 口数

输出口 74LS273	4	3	2	1	0
输入口 74LS244	4	5	6	7	8

### 三、存贮器连接特点

图中所示的专用微机, 只用 1 片 2716 EP-

ROM, 片选信号中不使用地址信号, 直接采用  $\overline{\text{MERQ}}$  作片选信号, 以使电路简洁; 这种接法的另一个好处是存贮器地址空间中, 以 2K 为间隔, 地址重叠区有  $2^5 = 32$  个, 便于开发。

在使用一般的 Z80 单板机开发时, 单板机配上与专用微机 I/O 口地址码相同的 I/O 口电路及控制器的接口电路调试程序时, 若调试程序在 2000H 或 3000H (CMC-80) 开始的 RAM 区中存放, 在此调试程序运行正常后, 便可直接把此程序烧入 EPROM (不必把程序修改成从 0000H 开始执行的程序后再烧入 EPROM), 把此 EPROM 插入到如图示专用微机组成的样机中的存贮器插座, 整机即可正常工作。这种存贮器的连接法, 没有单板机上的调试程序与实际专器微机中运行程序地址不同所带来的麻烦, 单板机上调试好的程序不用修改地址即可使用。

### 四、用途

本文所述的专用微机具有最小微机系统的特点, 体积小、成本低, 工作可靠。使用 Z80 CPU 芯片具有很大的优点, Z80 指令功能强, 软件编制方便, 已成功地用在条件控制器中。

该机可以很方便地按实际要求扩充硬件, 采用 Z80 单板机也很容易开发其软件, 可用来代替单板机作顺序、条件控制器中的专用微机, 适用于微机化产品。

参考文献 (略)

# 通用性较强的数据检索程序 (四)

唐景岩

(天津市电子计算机研究所)

## 五、内部文件

在本文介绍的程序运行前,须用CREATE命令建立三个内部文件。其结构如下:

### 1.FAME文件

字段名 类 型	NUMB	NAME	TYPE	WIDTH	MEANS
	N2	C8	C1	N2	C20

### 2.IN文件

字段名 类 型	NUMB	OPC	VALUE	OPL
	N2	C2	C20	C1

### 3.OUT文件

字段名 类 型	NUMB
	N2

## 检索程序清单 (附一)

```

PARA NUM, COH0, SF
RET=' N'
SR=' 0'
SC=' 0'
DO WHILE RET=' N'
CLEAR
PP=' 0 '
@ 4, 20 SAY ' 输出方式 (0. 显示1. 打印) ?
      ' GET PP
READ
CLEAR

```

```

STRR=' -----'
STRR1=' -----'
SELE 5
COM0ID=COM0+' ID'
USE &COM0 INDEX &COM0ID
IF SR=' 0' .OR. SC=' 0'
GO TOP
@ 0, 0 SAY ' '
DO WHILE. NOT. EOF ( )
@ ROW ( ) , COL ( ) SAY SUBSTR
      (STR(NUMB, 2) + '=' + MEANS+' ' , 1, 16)
SKIP
ENDDO
ENDIF
@,9,0 SAY '退出检索(Y/N)? ' GET RET
READ
IF UPPE(RET)=' Y'
EXIT
ENDIF
SELE 3
USE IN
IF SR=' 0'
ZAP
ENDIF
SET RELA TO NUMB INTO E
SELE 4
USE OUT
IT SC=' 0'
ZAP
ENDIF
SET RELA TO NUMB INTO E
IF SC=' 0'
GO TOP
@3,0 CLEAR

```

```

@ 8, 2 SAY '确定输出格式: '
I = 1
DO WHILE 1 > 0
APPEND BLANK
DO WHILE 1 = 1
@ 9, 2 SAY '第' + SUBSTR (STR(I),
    9) + ' 列 = 项号' GET UNMB
READ
IF NUMB = 0
SELE 5
USE
SELE 4
DELETE
PACK
APPEND FROM &COM0
SELE 5
USE & COM0 INDEX &COM0ID
SELE 3
SET RELA TO NUMB INTO E
SELE 4
SET RELA TO NUMB INTO E
EXIT
ENDIF
GO RECNO ( )
@ 8, 40
IF NUMB <> E -> NUMB
@ 8, 40 SAY, 无此项号, 重新输入! '
?? CHR(7)
ELSE
EXIT
ENDIF
ENDDO
AA = ' '
@ 9, 30 'SAY结束(Y/N)? ' GET AA
READ
IF UPPER(AA) = ' N'
I = I + 1
LOOP
ENDIF
EXIT
ENDDO

```

```

@ 8, 52 SAY' 稍候.....'
K1 = ' ' | ' + '
K2 = ' '
K3 = ' '
K4 = ' '
K5 = ' '
K6 = ' '
K7 = ' '
I = 49
S = 0
TT = ' K1'
SELE 4
GO TOP
DO WHILE. NOT. EOF ( )
WW = E -> WIDTH
W = STR(WW, 2)
STRL = LEN(&TT)
IF STRL > 100
&TT = SUBSTR(&TT, 1, STRL - 1)
I = I + 1
TT = ' K ' + CHR (I)
ENDIF
TP1 = ' '
TP2 = ' '
IF E -> TYPE = ' N'
TP1 = ' STR ('
TP2 = ' , &W) '
ENDIF
&TT = &TT + ' SUBSTR ( ' + TP1 +
    TRIM (E -> NAME) + TP2 + ' +
    SPAC(10), 1, ' + W + ' ) + ' + " '
    | ' + "
STRL = LEN (&TT)
S = S + WW + 2
SKIP
ENDDO
&TT = SUBSTR (&TT, 1, STRL - 1)
KI = I - 43
ENDIF
IF SR = ' C'
@ 8, 6 CLEAR

```

( 待续 )

# 用APPLE II 机把Z80汇编文件 写入EPROM的一种简便方法

张富有 马东升

(北京工业学院自动控制系)

随着微机应用技术的推广普及,使用某些单板机来编程及调试的方法,已经不能满足实际需要。APPLE II plus 是目前应用比较广泛的一种微机系统。在配上 Z80 卡以后,可以运行CP/M操作系统支持的全部软件,它为单板机软件的开发调试提供了方便。但是APPLE II 所用的 EPROM 写入器是在 APPLE II DOS (磁盘操作系统)管理下工作的。不能直接在CP/M系统下将 Z80 汇编文件写入EPROM。

将Z80汇编文件写入EPROM的方法,一般是将要写入的Z80文件进行汇编、装配连接、存盘,得出可执行的目的文件。调试正确后,打印其全部十六进制机器码。然后在DOS下,将机器码一一键入,再写入EPROM。这样不但速度慢,而且容易出错。

下面介绍的方法是在应用目前现有硬件前提下,建立一个CPD.COM文件(建立方法在最后叙述),利用这个文件,完成从CP/M到DOS操作系统的转换,在转换过程中使用户在内存RAM区的内容不变。转换以后,即可在DOS系统下用现有的方法完成对EPROM的写入。

具体的操作步骤是:

1. 准备好要写入文件。假如要写入的文件名为ZFY.Z80,其起始地址为0800H,这个文件是输入的Z80汇编源程序,经编辑,汇编,装配连接后,得到可执行文件ZFY.COM。

把 EPROM 写入器插入外设槽 #1—#7中的任一空位置,例如#2。

将CP/M盘放入驱动器A,开启主机电源,这时计算机在CP/M操作系统管理下工作。

CRT 上显示:

APPLE II CP/M

56K Ver 2.20

(C) 1980 MICROSOFT

A>

2. 将要写入文件ZFY.COM调入内存:

A>DEBUG, ZFY.COM✓

DEBUG Version 00.17

NEXT=1000

NEXTM=1000

—A—C

A>

3. 运行CPD.COM文件,从CP/M转换到DOS。

A>CPD✓

□ (光标)

在运行CPD.COM文件后,进入等待状态,此时CRT上不出现任何提示符,只是光标□出现在下一行的开始,从驱动器A取下CP/M盘,换上DOS系统盘,按回车键(或任意键),CRT即显示:

]

这时计算机已转为DOS操作系统管理之下,可按DOS下对EPROM进行写入。

由于6502 CPU所用地址与 Z80 CPU



所用地址差1000H,此时要写入文件的起始地址应是1800H。

这种方法的突出特点是简单易行,只需调入和转换两步操作。

对于复杂的程序,上机调试需要反复进行多次,将重复的操作命令用编辑程序ED作成一个命令文件 ALL.SUB,文件名ALL是任意的,扩展名必须是SUB。建立的ALL.SUB根据需要可包括如下命令:

- ①ED    \$1.Z86
- ②ASMB    \$1
- ③L80    \$1/E
- ④SAVE    \$2    \$1.COM
- ⑤DEBUG    \$1.COM

其中\$1,\$2为形参,调用该命令文件时,用实在参数代入,设用户程序名为ZFY假定页数为14,通过调用批处理文件SUB-MIT使上述5个动作依次进行。

A> SUBMIT    ALL    ZFY    14

当执行到任一命令时,只要按下ESC键即可返回CP/M,而不会继续执行下去。

调试完毕,再调用CPD,进行转换,然后写入EPROM。用这种方法可以简单地完成上机调试,写入EPROM的全过程。

从CP/M向DOS转换的文件 CPD.COM 可按下述步骤建立(文件引自参考文献):

```
A> DEBUG ✓
DEBUG Version 00.17
-S100 ✓
0100 00 0E ✓
0101 00 01 ✓
0102 00 CD ✓
0103 00 05 ✓
0104 00 00 ✓
0105 00 21 ✓
0106 00 00 ✓
0107 00 C6 ✓
0108 00 22 ✓
0109 00 D0 ✓
010A 00 F3 ✓
010B 00 2A ✓
010C 00 DE ✓
010D 00 F3 ✓
010E 00 77 ✓
010F 00 . ✓
- ^ C
A> SAVE 1 CPD.COM ✓
A>
```

参考文献:

APPLE I CP/M 操作系统使用手册

(上接第30页)

包括“P”参数。这两条命令语句的次序不能颠倒。

定义软件接口的AUTOEXEC.BAT文件必须放在系统盘上。当启动主机时,如果硬件接口都联上的话,激光打印机与主机就会自动联接。需要指出,如果硬件接口用的是备用异步通讯适配器,批命令中的“com1”要改为“com2”。如果系统盘上原有的AUTOEXEC.BAT文件需要保留的话,定义软件接口时只要将2行命令语句添加到原有的批件中就行。

为了检查软件接口是否联通,可以编一小段BASIC程序,运行一下便知。如:

```
C> BASIC
10 LPRINT CHR$(27) + "E"
```

```
20 LPRINT "LASERJET OK!"
30 LPRINT CHR$(13);
40 LPRINT CHR$(12);
50 END
RUN
```

如果激光打印机打出字样“LASER-JET OK!”就说明软件接口定义好了。如果激光打印机不能打出字样,就得检查一下程序是否有误,批命令文件是否有误,接口电路是否畅通,异步通讯适配器是否松动,等等。查出原因,纠正错误后,使打印机自检,然后再运行BASIC程序。如果仍然不能打出字样,那就要参阅有关手册和资料,或向有经验的人请教。

激光打印机是一种年轻的很有发展前途的高级输出设备,希望更多的人来探讨激光打印机的开发和应用问题。