

# 开放式模块化通用数控系统软件设计<sup>\*</sup>

谢 明 红

(华侨大学机电工程系, 泉州 362011)

**摘要** 由于数控系统软件设计中存在设计不规范、可靠性与可维护性差等缺点,出现了多品种、低档次的数控系统软件。为此,提出一种开放式、模块化软件设计方法。它基于DOS操作系统,采用C语言和汇编语言混合编程,调用08H号中断前后台结构,实现管理和控制同步进行,从而实现“宏观并行,微观分时处理”。同时,建立插补库以及将扩展内存(XMS)的使用引入数控系统软件设计。在实际软件开发过程中,取得良好的效果。

**关键词** 开放式, 模块化, 插补, 数控系统

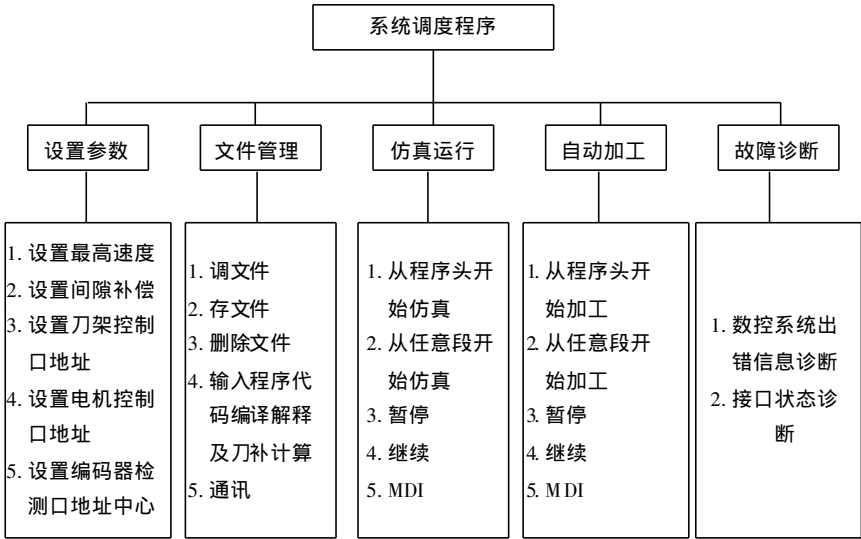
**分类号** TG 659

近几年来国外通用数控系统纷纷进入国内。它具有通用性好、可靠性高并便于与不同机床相接,而国产数控系统的市场占有率却呈下降趋势。一方面是国产硬件技术跟不上,如驱动电源、电子芯片等技术性能不过关,但另一方面更重要的原因是软件技术不过关。目前应用比较广泛的国产简易数控系统软件<sup>[1]</sup>仍以单片机为主、数码管显示,其功能单一、信息量小、通用性差,不便于修改和移植;增加功能、改变接口,甚至要修改整个软件,出现了多品种、低档次的数控系统;软件设计不规范,人力、物力大量重复使用;软件可靠性差、难以维护。所以现代数控系统(CNC)的好坏,关键取决于软件的结构及其设计方法。随着STD工控机、工业PC机性能的提高及价格的降低,使其在数控系统上的使用更加广泛。由于采用了DOS操作系统,可以使用高级语言(C语言)和8086汇编语言混合编程。其中C语言实现屏幕编程、图形和数据动态显示,汇编语言实现插补计算和实时控制。通用数控系统的发展迫切要求软件设计规范化,使软件更加模块化、开放式,便于修改和扩展功能。

## 1 软件总体结构<sup>[2]</sup>

通用数控系统是一种典型的实时控制系统。系统调度程序是整个系统的核心部分,它完成对整个系统多任务的控制和管理,同时对系统外设进行控制和管理。系统管理模块包括仿真运行、设置参数、文件管理、故障诊断等功能,采用C语言编程。系统控制模块即自动加工模块,包括为了节省内存而实行的动态译码、刀具半径补偿、速度处理、插补计算、位置控制等实时性要求比较高的部分,并采用8088/8086汇编语言编程。整个系统软件采用C语言和汇编语言混合编程,充分发挥两种语言各自的优点。软件要用08H号中断实现管理和控制同步进行,它

是一种前后台式结构. 后台程序(即实时中断服务程序)执行全部实时功能, 而准备工作及协调处理则在前台程序中完成. 前台程序是一个循环程序, 在其运行过程中实时中断服务程序不断插入, 共同完成零件加工任务, 从而实现“宏观并行, 微观分时处理”. 系统软件框图, 如附图所示.



附图 系统总调模块软件框图

2 插补库模块<sup>[6]</sup>

众所周知, 标准 FORTRON 语言、C 语言等都具有标准库, 软件开发者使用时只要给定函数名和入口参数就可调用标准库. 它既节省开发时间, 又使软件设计更好、更可靠. 为此, 我们可以根据数控系统特点建立一个插补库. 插补是数控系统的核心, 它的好坏直接关系到数控系统性能的好坏.

国内数控系统开发者都要开发自己的各自插补模块, 但不同的人具有不同的设计方法. 这不仅花费很多时间, 而且有时性能不一定好. 当然, 插补库不同于标准语言库, 它具有自己的特点. 如不同的系统控制电机接口不一样, 有的是软件环分, 有的是硬件环分; 机床电机不一样, 负载不一样, 升降速参数需要调节, 但插补计算过程是一样的. 因此, 可以将各插补计算过程编成独立模块, 而要调用的(上面提到的不同过程)也编成一个独立模块. 这一模块可以作为源程序提供, 使软件开发者可以根据自己需求进行修改. 将开放式和模块化有机结合起来. 为了便于以后开发, 有必要将插补算法程序标准化, 将各种算法放在一个插补库中, 开发人员就可直接调用. 这样, 可以加速程序开发过程, 使程序设计更加有效, 促进模块化设计, 提高软件设计的可靠性.

例如, 直线插补函数为

```
int G01 (long int X, long int Y, long int Z, int F);
```

顺圆插补函数为

```
int G02 (long int X, long int Y, long int Z,
```

long int  $I$ , long int  $J$ , long int  $K$ , long int  $F$ );

逆圆插补函数为

```
int G03(long int  $X$ , long int  $Y$ , long int  $Z$ ,
        long int  $I$ , long int  $J$ , long int  $K$ , long int  $F$ );
```

其中  $X, Y, Z$  分别为  $X$  轴、 $Y$  轴、 $Z$  轴方向要走距离值;  $I, J, K$  分别为圆弧插补时圆心相对于起点的坐标在  $X$  轴、 $Y$  轴、 $Z$  轴上分量值;  $F$  为进给速度。

由于在输入 G 代码时, 给定的  $X, Y, Z$  值一般为浮点数, 而插补程序用汇编语言编程. 为了便于编程和节省指令运行时间, 因此各插补函数入口参数距离值为长整型, 并在调用插补函数之前需要将浮点数转化为长整型的脉冲数. 所有插补函数返回值为整型, 表示插补程序退出时的状态标志. 如正常退出, 暂停退出, 异常退出等。

### 3 加工程序的解释模块

数控加工程序是由用户根据加工零件而编制的, 用户可以随时修改. 该模块的主要任务是负责对数控加工程序进行解释, 同时进行刀具半径补偿计算, 以使其转化成数控执行机构的相应的动作, 最后加工出合格的产品。

数控加工程序, 一般都是采用源代码形式, 即以 ASCII 码形式存储的, 其目的在于编辑方便. 解释程序首先对数控加工程序进行错误识别, 并能指出错误行号及错误类型, 只有在源程序准确无误的情况下才进行后面的转换工作。

此模块设计中的关键技术是构造一个最佳存储代码的数据结构, 以供插补程序循环调用, 一般可以采用结构数组和结构链表两种方式. 由于数控加工程序长短不一, 如果采用结构数组方式, 为了满足最大文件长度, 必然造成文件长度短时内存大量浪费. 因此, 最好采用结构链表方式. 它具有前后结构指针, 根据每个代码要求动态申请内存, 不需要时动态删除, 从而达到一种最优设计方法. 其结构定义为

```
struct Compile Table      /* 结构类型名 */
{
    unsigned char Code Type;
        /* 加工代码类型, 分别代表 Nxx, Gxx, Mxx, Txx, Sxx. */
    unsigned int value;      /* 加工代码值, 如 G01, 则 value= 1 */
    union ptrtosubtable      /* 各种加工方法指针联合类型名 */
    {
        struct G01type * G01_PTR;    /* 直线插补数据结构指针 */
        struct G02type * G02_PTR;    /* 圆弧插补数据结构指针 */
        .....                    /* 其它插补数据结构指针 */
    } table;
    unsigned int line_number;    /* 加工代码行号值 */
    struct Compile_Table * previous, * next; /* 分别为结构前后指针 */
} * table_ptr;                /* 一个加工代码结构指针变量名 */
```

## 4 扩展内存(XMS) 模块设计及使用

扩展内存(XMS)是指 1 MB 寻址范围以外的内存. 由于全屏幕编辑, 程序代码的编译解释会随着零件复杂程度增加, 循环指令增多, 程序所占内存也增多, 因而常规内存 640 KB 远远不够使用. 如果使用文件保存到硬盘, 再从硬盘调用, 则显然速度太慢. 本软件设计了扩展内存(XMS) 模块. 该模块使用 C 语言和嵌套汇编语言混合编程, 包括扩展内存(XMS) 的存在检测、扩展内存的大小检测、扩展内存的数据存取等几个子模块. 将其作为公用基本模块, 供其它模块(包括全屏幕编辑, 屏幕快速保存及弹出, 汉字库存取等) 使用, 使软件运行速度和功能大大增强. XMS 功能是由专门设备驱动程序(HIMEM.SYS) 提供.

### 4.1 扩展内存(XMS) 存在检测

```
MOV AX, 4300H
```

```
INT 2FH
```

返回值为 AX

上面两句为判断当前内存是否有扩展内存(XMS), 如果没有返回错误代码; 如果有, 该中斷返回 XMS 驱动子程序在内存中的段地址和偏移地址(设为 XMS \_CONROL).

### 4.2 扩展内存大小检测

```
MOV AX, 008H
```

```
CALL XMS _CONTROL
```

返回值为 AX (KB), 返回扩充内存大小. 如 AX 值为 7 000, 表示扩展内存有 7 000 KB, 即 7 MB 字节.

### 4.3 扩展内存数据存取

```
MOV AX, 008H
```

```
MOV DS, ...
```

```
MOV SI, ...
```

```
CALL XMS _CONTROL
```

DS: SI 为扩展内存块移动结构指针, DS 为段地址, SI 为偏移地址, 其结构形式为

```
struct EMB _Struc
```

```
{long int Count;          /* 要移动内存块字节大小 * /
  int Source _Handle;      /* 源块句柄, 如为常规内存, 则为零 * /
  long int Source _Ofs;    /* 源块偏移值 * /
  int Destin _Handle;      /* 目的块句柄, 如为常规内存, 则为零 * /
  long int Destin _Ofs; } /* 目的块偏移值 * /
```

具体设计过程见参考文献 [4].

## 5 结论

该软件全部采用 C 语言和 8088 汇编语言混合编程, 充分发挥各自的优点, 具有良好的人机交互界面. 由于采用了模块化结构, 使得软件使用和修改非常方便, 同时具有开放性, 可以根据自己的需要修改接口源程序. 该设计方法应用在我们开发的数控系统软件上, 取得良好的效

果. 各模块功能基本全部实现.

### 参 考 文 献

- 1 尤芳怡, 张奕鑫. 自动丝网印花机控制系统硬件的研究. 华侨大学学报(自然科学版), 1997, 18(2): 177~180
- 2 肖本贤, 谢明红. 数控蜗杆砂轮磨齿机的软件设计. 合肥工业大学学报(自然科学版), 1995, 18(4): 42~47
- 3 谢明红, 肖本贤. 非圆二次曲线插补算法. 制造技术与机床, 1996, (10): 18~20
- 4 求伯君. 新编深入 DOS 编程. 北京: 学苑出版社, 1994. 200~210

## An Open-Type Modular Software Design for General Numerical Control System

Xie Minghong

(Dept. of Mech. & Electr. Eng., Huaqiao Univ., 362011, Quanzhou)

**Abstract** In software design of numerical control system, there appear software of wide varieties and low grade due to the design not come up to standard and short of reliability and maintainability. In this connection the author puts forward an open-type modular method of software design based on DOS operating system. By adopting mixed programming with C language and assembler language and calling 08H interruption as a structure of foreground and background, the synchronization of management and control can be realized and then the combination of concurrent macro processing and time-sharing microprocessing can also be realized. Moreover, the author suggests to set up interpolation library and to extend the memory XMS into software design of numerical control system, of which good results have been achieved in actual software development.

**Keywords** open-type, modular, interpolation, numerical control system