

文章编号: 1006-2939(2003)03-0065-03

## Keil C51 与 A51 的混合编程

杜波, 李丽娜, 吴川

(长春工业大学 电气与电子工程学院, 吉林 长春 130012)

**摘要:** 讨论 C 语言程序与汇编程序混合编程时需注意的几个问题, 并给出实例进行说明。

**关键词:** 函数名转换; 参数传递; 段; 内联汇编

**中图分类号:** TP274.2      **文献标识码:** A

### 0 引言

以前单片机应用系统主要用汇编语言编写, 由于汇编语言程序的可读性和可移植性都较差, 用汇编语言编写的单片机应用程序周期长、调试和排错也比较困难, 而且对于很多通晓高级语言而不熟悉硬件的人来说, 很难掌握。C 语言既具有一般高级语言的特点, 又能直接对计算机的硬件进行操作, 表达和运算能力也较强, 许多以前只能用汇编语言来解决的问题现在都可以改用 C 语言来解决。Keil C51 是一种专为 8051 单片机设计的高效 C 语言编译器, 符合 ANSI 标准, 与其它高级语言比较, 生成的程序代码运行速度快, 所需要的存储器空间小, 完全可以和汇编语言相媲美<sup>[1]</sup>。但有时为了编程的直观或某些特殊地址的处理, 如对于某些特殊 I/O 地址的处理或控制时序要求较严格的场合, 还须采用一定的汇编语言编程。在以汇编语言为主体的程序中, 如果涉及复杂的数学运算, 又往往需要在汇编中调用 C 程序。这就涉及到汇编语言与 C 语言之间的混合编程和相互调用的问题。文中主要采用 Keil C51 编译器实现 C 程序与汇编程序的相互调用以及在 C 程序中内联汇编。

### 1 C51 与 A51 混合编程的几点说明

C 语言程序与汇编程序的相互调用可视为函数的调用, 要把汇编程序加入到 C 程序中, 必须使汇编程序像 C 程序, 这就是说要和 C 程序一样有明确的边界参数、返回值和局部变量。C 程序

与汇编程序间的相互调用一般需注意以下几点。

#### 1.1 函数名的转换

C51 编译器在对源程序编译时, 函数名依据其定义的性质不同会转换为不同的函数名, 因此要求 C 语言程序和汇编程序的相互调用时, 汇编程序必须服从这种函数名的转换规则, 其函数名的转换见文献[2]。

#### 1.2 参数传递

在混合语言编程中, 关键是传递参数和函数的返回值。它们必须有完整的约定, 否则传递的参数在程序中取不到。通常情况下汇编模块服从高级语言。8051 的工作寄存器最多传递三个参数, 参数传递的工作寄存器选择见文献[2]。

如果在调用时, 参数无寄存器可用, 或是采用了编译控制指令“NOREGPARMS”, 则参数的传递将发生在固定的存储器区域, 该存储器区域称为参数传递段, 其地址空间取决于编译时所选择的存储器模式。

#### 1.3 段

汇编程序中, 每一个功能函数都有自己的代码段。每个具有局部变量的汇编语言程序函数都必须指定自己的局部数据段, 这个局部数据段可以用来为其它函数访问作参数传递用。如果用汇编语言编写的函数和 C 编译器编译出来的代码风格一样, 连接器将能够对数据段进行覆盖分析。如果有局部变量, 它们也都有相应的存储空间(DATA, XDATA 等)。依赖于所使用的存储器模式, 在相互调用时, 汇编语言必须服从 C51 有关段名的命名规则, 段名的命名规则见表 1。

收稿日期: 2002-09-04

作者简介: 杜波(1976—), 女, 吉林长春人, 长春工业大学硕士研究生, 主要从事单片机应用研究。

表1 段名的命名规则

存储空间	CODE	XDATA	DATA	BIT	PDATA
段名开头	? PR? CO	? XD	? DT	? BI	? PD

参数传递段给出汇编子程序使用的固定存储区,就象参数传递给C函数一样,参数传递段的首址通过名为“? 函数名? BYTE”的PUBLIC符号确定。当传递位值时,使用名为“? 函数名? BIT”的PUBLIC符号。对于char, int, long和float类型的数据,使用“? 函数名? BYTE”的PUBLIC符号。所有传递的参数放在以首址开始递增的存储区内,并在数据段中先按被传递参数的顺序定义若干字节,然后再定义其它局部变量数据字节。

#### 1.4 返回值

当函数具有返回值时,也需传递参数,函数返回值放入CPU寄存器中,函数返回值参数的传递见文献[2]。

#### 1.5 内联汇编

可以通过预编译指令“# pragma asm”和“# pragma endasm”在C代码中插入汇编代码。这种方法的实质是通过asm与endasm告诉C51编译器中间行不用编译为汇编行,因而在编译控制指令中有SRC以控制将这些不用编译的行存入其中。如果你不指定“src”选项,编译器将忽略在“asm”和“endasm”中的代码,很重要的一点是编译器不会编译你的代码,并把它放入它所产生的目标文件中。

## 2 应用实例

限于篇幅,仅采用三个例子,旨在说明问题。

### 2.1 在C51中调用汇编程序

由8051控制指示灯,从与P0口相连的8个开关读入控制量信息,将最近读入的十次信息存储在字节数组array[10]中,并将每次读入的信息显示在与P2相连的8位LED指示灯上。每次读入信息间隔1s, 0.9s, ..., 0.1s,由延时程序msec(100-j)完成。程序反复执行,无限循环。8051单片机外接12MHz晶振。

C程序代码如下,其中msec函数的参数传递在R7中进行。

```
NAME DELAY.C
#include <reg51.h>
void msec(unsigned char);/*对被调用函
```

数进行说明\*/

```
void main()
{unsigned char array[10];
unsigned char i,j;
while(1)
{for(i=0;i<=9;i++)
{j=10*i;
array[i]=P2=P0;
msec(100-j);}
i=0;} }
```

C程序所调用的汇编代码如下:

```
NAME MSEC.ASM
? PR? _MSEC? DELAY SEGMENT
CODE;定义程序代码段
PUBLIC _MSEC;公共符号定义
RSEG ? PR? _MSEC? DELAY;程序代码段
_MSEC: (延时程序略);起始地址
```

### 2.2 汇编中调用C函数

当用汇编调用C函数时,汇编程序相当于C程序中的主函数。输入三角形三边的边长,其数值类型为浮点数,利用数学公式计算出三角形面积。汇编代码如下:

```
NAME AFUNC.ASM
? PR? funca51? AFUNC SEGMENT
CODE;定义程序代码段
? DT? VAR? AFUNC SEGMENT DATA
OVERLAYABLE;定义可覆盖局部数据段
STACK SEGMENT IDATA
EXTRN CODE(cfnc);外部函数声明
EXTRN DATA(? cfnc? BYTE)
RSEG ? DT? VAR? AFUNC;可覆盖局部
数据段
v_a: DS 4;定义传递参数字节
v_b: DS 4
v_c: DS 4
result: DS 4;存放 cfnc 函数结果的变量
RSEG STACK
DS 20H;为堆栈保留 32 个字节
CSEG AT 0;定义绝对代码段
LJMP start
RSEG ? PR? funca51? AFUNC; funca51
代码段起始
start:
USING 0
```

```

MOV SP, #STACK-1 ;初始化堆栈
MOV ? cfunc? BYTE+0, v_a+0 ;取三个
float 参数
:
MOV ? cfunc? BYTE+11, v_c+3
LCALL cfunc;调用 C 函数 cfunc
MOV result, R4;存取结果
MOV result+01H, R5
MOV result+02H, R6
MOV result+03H, R7
RET
END
MOV R2, #0
MOV R3, #0
DELAY: DJNZ R3, DELAY
        DJNZ R2, DELAY
# pragma endasm
j--;}
i--,;} } }

```

C51 源文件中有一个名为 cfunc 的函数,它完成求取三角形面积的算术运算功能,其中传递的三个参数为三角形的三个边长。该 C 程序略。

### 2.3 在 C 中内联汇编

P1.0~P1.6 接有 7 个发光二极管 LED0~LED6。当并行口线输出为低电平,LED 亮,反之 LED 灭。P1.7 接按钮 K。当 K 按下闭合时, P1.7=0,反之 P1.7=1。执行此程序,开始 7 个发光二极管全灭。按下 K 后,则 LED6~LED0 依次变亮,且每个灯亮的时间依次递增。其中的延时程序是用汇编语言编写内联在 C 程序中的。

```

#include<reg51.h>
sbit P_0=P1^0 ;设置位
:
sbit P_7=P1^7;
main()
{ int i,j;
while(P_7==0)
{i=6; /* i 通过 R6, R7 进行参数传递 */
while(i>=0)
P1=255;
switch (i)
{case 0 :P_0=0;break;

```

以上程序均通过 KEIL C51 V7.0 调试成功。

### 3 调用汇编与内联汇编

建议在 C 程序中调用汇编程序。这样可以保持 C 语言的完整性。C 语言可移植性较强而汇编程序较差。当需要移植时,汇编程序需要大量更改。为了使程序具有较好的可移植性,建议在 C 程序中少使用汇编程序。

### 4 结论

虽然 C 语言越来越广泛地应用于开发单片机程序,但汇编语言在系统开发中仍然有不可替代的作用。用高级语言可使产品的开发更加快速而稳定,而汇编语言能够完成一些高级语言不能做到的事情,将汇编语言和 C 语言结合使用,就可扬长避短,开发出更好的应用程序。

#### 参考文献:

- [1] 徐爱钧,彭秀华. 单片机高级语言 C51 Windows 环境编程与应用[M]. 北京:电子工业出版社,2001.
- [2] 马忠梅,马岩. 单片机的 C 语言应用程序设计[M]. 北京:北京航空航天大学出版社,1999.

## Mixed Programming of Keil C51 and A51

DU Bo, LI Li-na, WU Chuan

(School of Electrical & Electronic Engineering, Changchun University of Technology, Changchun 130012, China)

**Abstract:** A few problems that must be noticed when programming with mixed C51 and A51 are discussed and illustrated with examples.

**Keywords:** function name conversion; transfer parameter; segment; embed assembler.