

第 3 章

汇编程序设计与编译方法

CHAPTER

学习计算机课程的人都会感到汇编程序设计和编译方法课程比较难学,其实难学的原因是我们对计算机本身的结构,特别是 CPU 的结构和信息运行的方式理解不透所致。再者,实用的计算机过分注重应用,因而指令系统庞杂,反而干扰了我们对基本理论和方法的掌握。其实学习计算机汇编程序设计,不宜将问题搞得过于复杂,应尽可能地以简练的形式表述相关的内容。PMC110 计算机包括系统管理使用的指令在内,也不超过 52 条,这样用户容易记住,容易使用。

3.1 PMC110 计算机的输入输出管理

一台计算机确定之后,如果需要让它能够执行用户的程序,那么最活跃的部分恐怕就是它的输入输出设备了。运行用户程序,首先要将该程序设法送到存储器当中,然后才能够将执行的权力交给用户程序,这样用户程序才能够运行。

将用户程序输入计算机进行执行的过程,需要根据该计算机的输入输出设备,解决程序如何进入内存,放在内存的什么位置,怎样启动执行用户程序的一系列问题。PMC110 计算机的输入设备是二进制的开关组,输出设备是发光二极管和数码管,所以输入输出的操作设计相对比较简单。对于输出,只要在输出的前端安排保持数据的寄存器,就能够完成持续显示数据的任务。程序执行中数码管 HEX1 和 HEX0 一般直接显示输出寄存器的内容,而数码管 HEX3 和 HEX2 一般直接显示输出 x 寄存器的内容,在数据输入时,它们的功能会有不同的变化。

PMC110 计算机的输入设备可以将二进制数直接送到输入总线,数据能不能正确地进入存储器,关键的问题是人工操作频率和计算机频率的匹配问题。一般情况,每一个数据在存储器中只能占据一个位置,然而由于频率的差异,在人工输入数据的极短时间内,也会有若干个存储单元被输入这个数据,这当然是不能容忍的。

解决这个问题的办法是使用缓冲区技术,所谓的输入缓冲区就是一个能以一种较慢的频率接收数据,而又能够以较快的频率输出数据的存储器。PMC110 计算机解决频率问题的办法是用按钮瞬间控制脉冲,让每次单击按钮的动作只能产生一个有效的且与计算机主频一致的周期,从而保证了准确地输入数据到缓冲区。

前面已经提到过,PMC110 计算机输入缓冲区有 128B,每单击一次输入控制按钮,就向缓冲区指针指示的位置写入 1B 的数据,同时缓冲区指针加 1,指向下一个要写入数据的位置。

输入到缓冲区的数据可以通过另外一个按钮操作将数据读入内存。PMC110 计算机采用输入中断的方式,保证用户操作的过程和计算机运行的一致性,当用户的输入需要读入内存的时候,用按钮操作来恢复系统程序的运行,将缓冲区的内容传输到内存的指定位置。与此同时,系统会检查用户程序是否完成输入,如果读入的是用户程序输入完成的信号,那么就转到用户程序执行,而这一切都是由操作系统的管理程序来控制完成的。

3.2 PMC110 计算机的管理程序

PMC110 计算机的存储器有两个:程序存储器和数据存储器。程序存储器 ROM/RAM 有 4KB 个存储单元,数据存储器 RAM 有 4KB 个存储单元,它们各自单独编址。程序存储器由只读存储器 ROM 和另一个随机存储器 RAM 联合而成,其中的只读存储器只有 32 个存储单元,放置了 PMC110 计算机系统管理程序。从 32 号开始的随机存储单元起放置用户程序,就是说有 4064 个存储单元可以放置用户程序。

PMC110 计算机的管理程序是用本机的汇编指令编写的,该管理程序简单明了,语义清楚,很容易掌握它的基本含义和方法。PMC 计算机的系统管理程序主要是一个能够人机交互的缓冲区监控程序。这个管理程序的长度不超过 32B,然而对整机的运行起着十分重大的作用。

PMC110 计算机的管理程序如下:

```
main      stpk          ;缓冲区空中断
          sda    32      ;将 32 送累加器
          dalp         ;累加器内容送指针 ptr 的低 8 位
          zero        ;累加器置 0
          dahp         ;累加器内容送指针 ptr 的高 8 位
m1         call    p1     ;调用缓冲区输入驱动子程序
          inc         ;指针 ptr 加 1
          jend    exec    ;程序数据输入结束就执行用户程序
          jmp     m1      ;转 m1 执行
exec       call    32     ;调用用户子程序
          jmp     main    ;返回管理程序开始
p1         stpk          ;缓冲区空中断
          inp         ;输入一个数据到 ptr 指示的程序存储单元
          ret         ;返回调用点
```

相信用户能够读懂这个程序,短短的程序中包含了中断管理、存储分配和子程序调用等一系列内容。

这段程序的意思是,若缓冲区空就中断等待,不然将内存放置用户程序的首地址 `exec` 送到指针 `ptr`,然后调用“将缓冲区数据送到指针 `ptr` 所指示的内存单元”的简单键盘驱动子程序 `p1`,再将 `ptr` 的值加 1。然后再看从缓冲区读出的数据是否是程序输入结束信号,如果是,那么将执行 `exec` 开始的用户程序,执行完成,就停止程序执行;如果不是用户程序输入结束信号,那么继续调用 `p1` 读下一个数据。

这个程序可以作为一般输入带缓冲区的管理程序。它管理着程序和数据从键盘装入内存,并在完成装填之后执行装入的程序。这个管理程序以“死循环”的方式运行。

这个管理程序看似简单,但意义重大,它正是操作系统的最基本形式,任何操作系统中,管理程序都处在核心的位置,没有管理程序的计算机不可能正常地运行。

这个简单的管理程序体现了操作系统的大部分工作,具体归纳如下。

1. 资源管理

这个管理程序不但管理着缓冲区的数据进出,而且也管理着内存的分配使用,同时也管理着处理机的执行和中断,担负着系统程序和用户程序协调一致的任务。

2. 方便人机交互使用计算机

这个管理程序用中断执行指令控制着人机交互,从而完成了计算机的正常运行,由于计算机结构十分简单,所以把一些控制任务交给使用计算机的人完成(例如停止中断和输入状态控制)。随着计算机的复杂化,多数人工操作的任务将逐渐转到计算机方面完成。

3. 提高效率

计算机的运行效率是相比较而言的,原计算机的管理程序将用户程序执行和数据输入转化成了一种较为灵活的方式,这是系统管理程序的作用。如果没有管理程序,只能将用户程序固定在存储器中,开机一次只能执行那个固定的程序,而这个管理程序的存在,使用户程序成为管理程序的子程序,故而一次开机就可以执行若干个用户程序,效率得到了相应的提高。

以上三点正是操作系统的工作和任务,操作系统是管理资源、提高效率、方便使用的程序集合。因此,原计算机的管理程序就是一个最简单的操作系统,详细分析它不难加深对操作系统的理解。

3.3 汇编语言程序设计

计算机汇编语言中的汇编指令一般是和机器指令一一对应的,因而汇编语言程序是最贴近计算机直接功能的程序,也是在这个计算机上运行速度最快的,效率最高的程序。但不同计算机的指令系统设计往往是不相同的,因而用汇编语言编写程序需要对具体的计算机有深刻的理解,对机器指令的功能和执行过程有清楚的了解,才能够编写出满意的程序。

3.3.1 汇编程序的基本结构

虽然各种计算机的指令系统不都相同,程序设计的基本结构和方法却区别不大,这种共性主要来自人类思维的一般形式,来自处理问题的基本方法,这些形式和方法是不随计算机的指令设计改变而改变的。所以学习汇编语言同学习其他程序设计语言一样,要从基本的程序结构开始。

用 PMC110 计算机指令系统描述的基本程序结构有 4 种,其中包括顺序结构、分支结构、循环结构和子程序调用结构。

1. 顺序结构

程序的顺序结构是任何计算机程序语言程序都不能没有的基本结构。它的基本规律顺序地,一个接一个地执行一个队列中的指令,除非遇到执行的指令是改变顺序的,不然就一直连续地进行下去。顺序程序结构的流程图如图 3-1(a)所示。

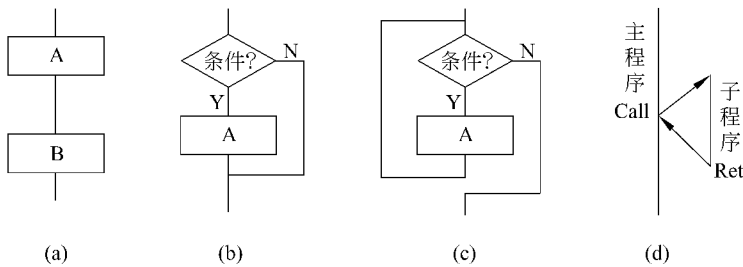


图 3-1 程序流程基本结构

用汇编指令描述顺序结构的形式,是将汇编指令从前到后一个接一个地写,按照这种先后关系得到的机器语言程序也不改变顺序,会按照汇编指令的顺序一个一个地执行下去。

2. 分支结构

分支结构的流程图如图 3-1(b)所示。用计算机的汇编语言描述分支结构,一般是对某个变量或运算的值进行条件判断,符合条件就进行某种操作,不然就进行另一操作过程或不进行操作。因为任何的运算条件都可以转化成 0 或负数的判断问题,所以对算术运算的结果判断,只考虑 0 或负数就可以。

PMC110 计算机将算术运算的结果放在累加器 DA 中,所以算术运行的条件判断利用 DA 中的值是正负或 0 来解决。分支结构由 x 判断的程序段一般如下:

```

...
LDA  X
JZ   P      ;也可以用是否为负数判断,使用 JN P 指令
ADD  Y
...
P:   ...

```

3. 循环结构

循环结构的流程图如图 3-1(c)所示,在汇编程序设计中可以用分支结构和无条件跳

转结构组合在一起描述。计算机用变量 N 来控制循环次数的程序段一般如下：

```

...
LOOP: LDA  N
      JZ   P      ;也可以用是否为负数判断,使用 JN P 指令
      SUB  ONE    ;N-1
      STR  N
      ...
      JMP  LOOP
P:    ...

```

4. 子程序调用结构

子程序调用结构的流程图如图 3-1(d)所示,这种结构在主程序要由子程序调用语句 Call,在子程序的出口要使用返回主程序语句 Ret。一般的程序段是：

```

...
CALL PROC      ;PROC 是子程序的首地址
...
PROC: ...
...
EXIT: RET      ;EXIT 是子程序的出口处,RET 是最后的指令

```

3.3.2 指令的分类

掌握了汇编语言程序的基本结构和指令的功能之后,剩下的就是训练自己的编程能力,而编程能力往往又取决于自己编写汇编语言程序的经验。本小节将用一些例题来说明某些常见的问题解决方法。这些题作者都在 PMC110 计算机上运行过,读者可以将它们进行适当的改编,这种做法对初学者很有好处。

学习汇编语言程序设计,首先要将每一个指令的意义搞清楚,特别是每条指令的动作要点,作用和最后的结果等都必须记牢。要能够快速牢固地记住指令,一般的方法要对指令进行分类。PMC110 计算机的指令可以分为算术运算类、数据传送类、逻辑运算类、跳转类、移位类和其他控制类。

1. 算术运算类指令

算术运算类指令包括 ADD、SUB、MULT、MULS、DIVI、ADDC、SUBC、MADD、INC 和 DEC 等,除了 INC 和 DEC 是一次指令之外,它们的共同之处都是三次指令,都是将累加器的内容与存储单元的内容进行运算,运算的结果都放在累加器 DA 中,只是乘法运算放的是积的低 8 位,除法运算放在累加器 DA 中的是商。还要记住,乘法运算将乘积的高 8 位放在了寄存器 X 当中,而除法运算却是将余数放在了寄存器 X 当中。MADD 是处理无符号超长数据乘法运算的专用指令,它可以与指针配合进行数组存放的超长数据处理。

INC 是让通用地址指针 ptr 加 1 的指令,DEC 是让 ptr 减 1 的指令。这两个指令的设置,主要是为了配合 ptr 进行连续的存储单元处理,INC 是解决增量处理使用的,而 DEC 是为减量处理准备的。

2. 数据传送类指令

数据传送类指令在程序设计中用得很多,也比较复杂。PMC110 计算机的数据传送类指令有 LDA、STR、SDA、IN、INP、INPD、OUT、DATX、XTDA、DATY、YTDA、DALP、PLDA、DAHP、PHDA、DAPD 和 DPDA 等,还有 PUSH 和 POP 指令。对于数据传送指令一定要清楚数据从什么地方获得,然后要传送到什么地方,中间通过什么设备,对数据传输有什么影响等。

这些数据传输指令中,往程序存储器写数据的只有 INP 指令,这个指令是通过通用指针 ptr 来隐含地确定程序存储器地址的,数据从缓冲区接收,然后写入程序存储器。往数据存储器写数据的指令有 IN、STR 和 INPD 三条指令,IN R 是从缓冲区将数据写入数据存储器的 R 存储单元;STR R 是将累加器 DA 的内容写入数据存储器的 R 存储单元;而 INPD 是将缓冲区的数据通过 ptr 指示写入数据存储器。

将一个常数用指令的方式直接放入寄存器的指令只有 SDA n,这一指令的结果会将常数送到累加器 DA 中。从数据存储器将数据输出到输出寄存器的指令只有 OUT R,它的执行会将数据存储器 R 存储单元的数据送到输出寄存器。

其余的数据传输类指令都是将数据从一个寄存器传递到另一个寄存器。值得一提的是,地址寄存器是 16 位的,数据寄存器是 8 位的,它们之间的数据传递要分地址寄存器的高 8 位和低 8 位处理,DALP、PLDA、DAHP 和 PHDA 就是这样的指令。

3. 逻辑运算类指令

基本的逻辑运算有“非(LNOT)”、“与(LAND)”、“或(LOR)”和“异或(LXOR)”,这几个基本的逻辑运算都是数据的对位运算。例如累加器 DA=01000110,存储单元 R 的内容是 00100111,那么 LNOT R 的结果是 10111001,LAND R 的结果是 00000110,LOR R 的结果是 01100111,LXOR R 的结果是 01100001。每一种逻辑运算的结果都保留在累加器 DA 中。

如果是考虑单值逻辑,那么只要判断结果为 0 和非 0 就可以了。

4. 跳转类指令

跳转类指令有 JMP、JZ、JN、JEND、JERR、JL、JR 和 JZX,其中 JMP 是无条件跳转指令,其余的都是有条件的跳转指令。条件跳转指令是程序中最具活力的元素,正是条件跳转的存在,才演绎出程序结果的千变万化。

在各种自动控制当中,经常将条件变化用某种传感信号来表达,当这种传感信号到达一定强度时,就可以认为是逻辑值 1,否则就是 0。将传感信号作成指令跳转的条件,就能够迅速地作出反应,在工业上就可以实现自动控制。

程序调用指令 CALL 和返回指令 RET 本质上也是转移类指令,只是这一对指令必须配合使用,它们进行程序转移的特点是从哪里转出去,就必须从那里转回来。

5. 数据移位指令

这类指令有 SHL 和 SHR 两个,SHL 是将累加器 DA 的值左移一位之后送到 X 寄存器,DA 的原值不变;SHR 是将累加器 DA 的值右移一位之后送到 X 寄存器,DA 的原值不变。要想进行多次的移位,只要将 X 寄存器的值不断地送回累加器 DA 就能够达到目的。

6. 其他控制类指令

PMC110 计算机除了上面的常用类型指令之外,还有一些控制类的指令。指令的执行只会产生计算机的动作,而不会进行数据处理。这类指令有 STPK、STP 和伪指令 END。ZERO 指令可以使累加器复位,因为这种复位可使累加器的值为 0,要强行将它归到运算类似乎也可以,但这种复位本质上是一种控制,故而还是划归到控制类指令比较好。NCRY 指令是调整加法,使带进位的加法运算的开始不具有进位的效果。NOP 指令是一个程序设计中占位的指令,不进行任何操作,也可以归纳到控制类指令当中。

3.3.3 分支循环结构

例 3-1 输入 10 个数,将其中最大的数输出。

这是一个有顺序、分支和循环的程序结构。可以将最大的数放在变量 MARX 当中,方法是将第一个输入的数放入 MARX,以后将每一个再输入的数都与 MARX 比较,比较之后将大数放在 MARX 中,最后将 MARX 输出即可。编写程序如下:

```
START      STPK
           SDA    10
           STR    N
           SDA    1
           STR    ONE
           IN     MARX
LOOP       LDA    N
           SUB    ONE
           JZ     EXIT
           STR    N
           IN     X
           LDA    MARX
           SUB    X
           JN     STOREX
           JMP    LOOP
STOREX     LDA    X
           STR    MARX
           JMP    LOOP
EXIT       OUT    MARX
           RET
           END
```

将这个程序写入程序设计器,输入不用区分大小写字母,然后单击两次“编译”按钮,可以得到如图 3-2 所示的结果。

将图 3-2 中“操作码”和“操作数”两栏的十六进制数顺序输入 PMC110 计算机执行,就能见到正确的结果。具体操作过程如下。

(1) 在 SW0 向下时,单击复位按钮,进行系统复位。

(2) SW0 向上,顺序输入操作码和操作数,有错误单击 KEY2 按钮后退修改,并可以

序号	标号	代码	代号	数据转化	操作码	操作数	数据量	变量位置安排	次数
1	start	stpk			52		0001		1
2		sda	5	5	8B	05	0003		2
3		str	n		8A	02 00	0006	n	3
4		sda	1	1	0B	01	0008	one	2
5		str	one		8A	03 00	000b	x	3
6		in	marx		89	05 00	000e	marx	3
7	loop	lda	n		81	02 00	0011		3
8		sub	one		83	03 00	0014		3
9		jz	exit		86	52 00	0017		3
10		str	n		8A	02 00	001a		3
11		in	x		89	04 00	001d		3
12		lda	marx		81	05 00	0020		3
13		sub	x		83	04 00	0023		3
14		jn	storex		87	49 00	0026		3
15		jmp	loop		85	2e 00	0029		3
16	storex	lda	x		81	04 00	002c		3
17		str	marx		8A	05 00	002f		3
18		jmp	loop		85	2e 00	0032		3
19	exit	out	marx		84	05 00	0035		3
20		ret			4E		0036		1
21		end			80		0037		1

图 3-2 用设计器编译的结果

将 SW1 开关向上检查进入缓冲区的数据量。

(3) 输入 80 之后,将 SW0 向下,然后单击 KEY3 按钮执行(此时右面数码管组闪烁)。

(4) SW0 向上,之后随便输入 5 个数(用 SW1 向上检查数量,如果多于 5,只要用 KEY2 按钮移回到 5 即可)。

(5) SW0 向下,然后单击 KEY3 按钮,就可以得到执行的结果。

如果出现错误,可以在 SW0 向下时复位,不按任何按钮,再将 SW0 向上进入输入修改状态。这时用 KEY1 按钮从前向后核对输入,对错误输入进行修改,直到程序末尾都正确后(出现 80),可以再次执行。此种操作方法可以多次执行程序,直到断电或输入其他程序为止。

3.3.4 多重循环

程序设计中多重循环的情况很多,掌握它的设计方法十分重要。程序设计中的有条件循环都要有控制循环的变量,这个变量叫控制循环变量。多重循环中,每层循环都要有控制循环变量管理循环次数,一般是外层控制变量改变一个数,内层变量要从头到尾变化一回。

例 3-2 按顺序将十六进制数 FF~00 以一定的速度输出。

用 PMC110 计算机运行下面测试时间的程序,完成 FF~00 的输出时间正好是 1 分钟,所以每输出一个数的时间是 60/256s。这个程序用 x 计数,用 x、m 和 n 作控制循环变量,循环分为三层。

START SDA 255

```

STR      X
STR      M
STR      N
SDA      1
STR      ONE
LOOP     LDA      X
        JZ      ROLL
LOOP0    LDA      M
        JZ      EXIT0
        SUB     ONE
        STR     M
LOOP1    LDA      N
        JZ      NEXT1
        SUB     ONE
        STR     N
        JMP     LOOP1
NEXT1    SDA      255
        STR     N
        JMP     LOOP0
EXIT0    SDA      255
        STR     M
        OUT     X
        LDA     X
        SUB     ONE
        STR     X
        JMP     LOOP
ROLL     SDA      100
        STR     X
        JMP     LOOP
```

编译之后得到的可执行程序为：

```

8B FF 8A 01 00 8A 02 00 8A 03 00 8B 01 8A 04 00 81 01 00 86 6D 00 81 02 00 86 59 00 83
04 00 8A 02 00 81 03 00 86 51 00 83 04 00 8A 03 00 85 42 00 8B FF 8A 03 00 85 36 00 8B FF 8A
02 00 84 01 00 81 01 00 83 04 00 8A 01 00 85 30 00 8B FF 8A 01 00 85 30 00 80
```

将这一串十六进制数输入 PMC110 计算机运行，就可以见到 FF~00 循环输出。

3.3.5 使用地址指针

进行数据块操作一般要使用指示操作地址的指针，通过指针的连续变化，可以方便地处理一块连续存储单元的内容。

PMC110 计算机有一个通用地址指针 ptr，其实它就是一个同步计数器，可以实现置数、初始化为 0、加 1 或减 1 操作。通过 ptr 进行操作的指令有 DALP、PLDA、DAHP、PHDA、INP、INPD、DAPD 和 DPDA，其中指令 INP 是通过 ptr 向程序存储器写数据的，

而指令 INPD 是通过 ptr 向数据存储器写数据的,使用中要区别清楚。

例 3-3 将一个数组输入到 PMC110 计算机,并将它们逆序显示出来。

由于 PMC110 计算机显示装置只有数码管或发光二极管,而且数量有限,故显示多个数据的办法,只能是分时进行,为此要用到延时子程序。

为使程序结构清晰,这里设计一个输入子程序,一个输出显示子程序和一个延时子程序。

输入子程序的设计如下:

```

INPUT      SDA      20          ;选择 20 号地址开始
           DALP          ;PTR 赋初值
           ZERO
           DAHP
INPUT1     STPK          ;等待输入
           INPD          ;输入数据到 PTR 所指单元
           LDA      I      ;计数
           SUB      ONE
           JZ       EXIT0
           STR      I
           INC          ;PTR+1
           JMP      INPUT1 ;继续输入
EXIT0      RET

```

逆序输出子程序设计的输出顺序就从当前位置开始,减序进行。具体设计如下:

```

DISPLY     DPDA          ;数据从 PTR 所指单元到输入 DA
           STR      X
           OUT      X
           CALL     ONESEC ;延时
           LDA      I      ;计数
           SUB      ONE
           JZ       EXIT1
           STR      I
           DEC          ;PTR-1
           JMP      DISPLY ;继续
EXIT1      RET

```

延时子程序采用双重循环,程序如下:

```

ONESEC     SDA      255
           STR      L
           STR      J
LOOP       LDA      L
           SUB      ONE
           JZ       EXIT10
           STR      L

```