

基于单片机汇编语言的通用多字节无符号数除法的改进

严克剑, 张森, 王丽琼

(广东工业大学 自动化学院, 广东广州 510090)

摘要:论述一种在8051系列单片机上实现的高可靠多字节除法运算的改进型算法,该算法通过改进传统的“移位-相减”算法,克服了常用的多字节无符号除法的局限性,可以针对任意长度的除数且不会发生商溢出。经实验得到计算结果准确无误,该算法具有算法简单、适应性强、高度可靠等优点,具有很强的实用性。

关键词: 单片机; 汇编指令; 多字节除法

中图分类号: TP301.4

文献标识码: A

文章编号: 1007-7162(2007)01-0022-04

单片微型计算机的应用越来越广泛,它不仅应用在各种控制上,有时也用来做数据处理运算^[1]。现在应用广泛的51系列单片机具有强大的运算能力,可以直接选用算术运算指令完成单字节(8位)数据运算。同时,它的进位(借位)标志为无符号整数算术运算提供了方便。目前有关数据处理运算的资料比较多,然而笔者在参考关于多字节无符号数据整数除法的资料时发现,目前资料所介绍的有关程序应用范围非常小,应用时受除数长度限制比较大。而在实际应用中,除法运算非常重要,甚至有时候要求必须快速地进行除法运算,这就要求编写的除法程序力求高效、快速^[2]。本文巧妙地利用“移位-相减”法^[3,5],以51系列单片机为例,设计了除数不受字节长度限制的单片机除法运算算法。该算法还可以根据实际需要扩充为位数更高的多字节除数算法,也可以移植到其它的单片机平台上。

1 目前常用的单片机多字节除法运算方法及其局限性

目前,单片机多字节除法运算要求被除数为除数的双倍字长,否则除法运算就不能执行。当被除数是除数的双倍字长时,首先比较被除数的高位字与除数的大小,如果出现被除数的高位字大于、等于除数,则认为是商溢出,置位溢出标志不执行除法操作。如果这些条件都成立,再采用一系列相减和移位操作来执行除法运算,每“移位-相减”一次求得一位商,当余数够减时得商1,不够减时得商0,循环此

步骤,直到被除数的所有位都处理完为止。

根据上述算法,设被除数为(R2R3R4R5),除数为(R6R7),商存于R4R5中,余数存于R2R3中。相应的程序流程图如图1所示。被除数(R2R3R4R5)的长

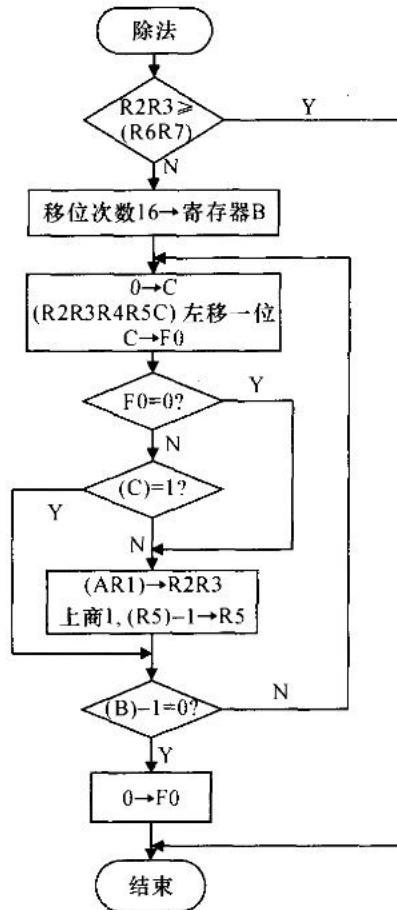


图1 传统多字节除法流程图

收稿日期: 2005-05-10

基金项目: 广东省科技计划项目(2003C102022)

作者简介: 严克剑(1982-),男,硕士研究生,主要研究方向为控制网络集成。

度是4个字节时,如果除数的长度是2位的,而且被除数高位(R2R3)与除数(R6R7)比较,如果前者小于后者,这个程序可以得到正确的答案。如果除数是3位或者被除数的高位(R2R3)大于除数(R6R7)这个程序没有办法得到运算所得的商。因此,如果被除数是4位字节长,而除数是3位或者4位时,就无法应用这个程序来进行除法操作。

2 除数为小于被除数的任意长度除法的单片机算法原理

在实际运用中,参与运算的数是任意的,有时需要求出余数,有时商数要求有4个字节,从上面的叙述可以发现,以前的算法在实际应用中受到了很大的限制。因此,笔者针对以上情况,在传统的算法基础上,设计了一套新的算法,克服了上述算法中的缺点,可以适合广泛的实际需要。下面以MCS-51汇编语言进行4字节长的除法为例进行说明:

1) 准备一个字节(8位)的空单元,这个空单元是最后用来存储余数最高8位的对应单元。

2) 计算被除数字长和除数字长之差,得到的值加1然后把得到的和乘以8,保存得到的积作为移位次数。例如,如果被除数是4个字节,除数是2个字节的, $((4-2)+1) \times 8 = 18H$,需要移位24次。

3) 判断除数最高字节是否为0,如果为0,则除数作为整体左移1个字节(8位),直到除数最高位不为0,把除数变成与被除数一样长度。如果除数为0,置位溢出,不执行除法运算。

4) 把被除数从高位到低位一位一位地移入之前准备的空单元,以这个空单元为最高位与被除数组成一个被减数,然后采用“移位-相减”法^[3],完成除法运算。

5) 所得结果存储在连续的5个存储单元里面,高位是余数的低位,低位是商的低位,余数的长度与除数的长度是一样的,如果除数的长度是4个字节,则这5个存储单元中的高4位是S余数,最低位是商,如果除数的长度是3个字节,则这5个存储单元中的高3位是余数,低2位是商,如此类推。

3 程序流程图、汇编语言的实现及测试结果

3.1 程序流程图及汇编语言的实现

在研制单片机应用系统时大多采用汇编语言作为软件工具^[4],本文所用的例子均为汇编语言编写。

被除数是4字节长的除法算法流程图如图2,程序如下:

```

MOV B,#08H
CJNE R0,#00H,DVDL
MOV B,#10H
CLR A
XCH A,R3
XCH A,R2
XCH A,R1
XCH A,R0
CJNE R0,#00H,DVDL
CLR A
XCH A,R2
XCH A,R1
XCH A,R0
MOV B,#18H
CJNE R0,#00H,DVDL
CLR A
XCH A,R1
XCH A,R0
MOV B,#20H
CJNE R0,#00H,DVDL
SETB OV
LJMP END_DIV
DVDL:
MOV DPL,#00H
DVDL2:
CLR C
MOV A,R7
RLC A
MOV R7,A
MOV A,R6
RLC A
MOV R6,A
XCH A,R5
RLC A
XCH A,R5
XCH A,R4
RLC A
XCH A,R4
XCH A,DPL
RLC A
XCH A,DPL
JC DVDLL1
DVDLL4:
SUBB A,R3
MOV A,R5
SUBB A,R2
MOV A,R4
SUBB A,R1
MOV A,DPL
SUBB A,R0
JC DVDLL3

```

DVDL1:

```

CLR C
MOV A,R6
SUBB A,R3
MOV R6,A
MOV A,R5
SUBB A,R2
MOV R5,A
MOV A,R4
SUBB A,R1
MOV R4,A
MOV A,DPL
SUBB A,R0
MOV DPL,A
INC R7;商的低位置1

```

DVDL3:

```

DJNZ B,DVDL2
END_DIV:
MOV R3,DPL
RET

```

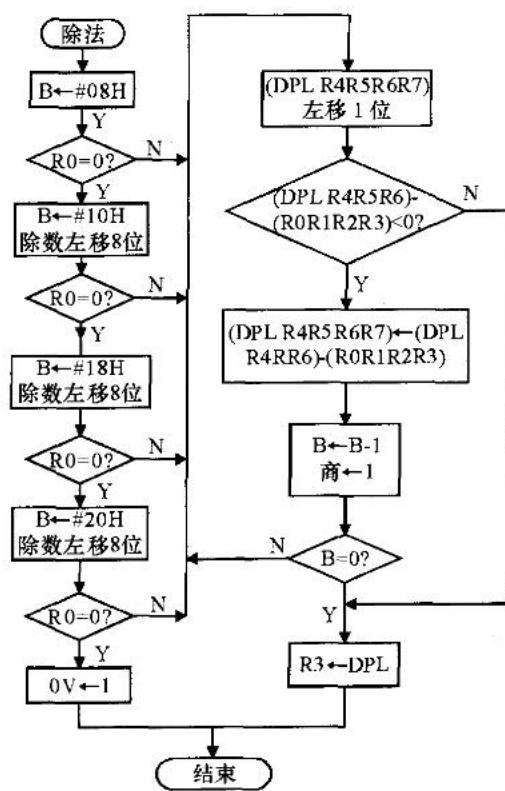


图2 通用多字节除法流程图

程序说明:本程序中(R4R5R6R7)存放被除数,(R0R1R2R3)存放除数,结果存放在(R3R4R5R6R7):当除数的长度为4字节时,(R3R4R5R6)是余数,(R7)是商;当除数的长度是3字节时,(R3R4R5)是余数,(R6R7)是商;当除数的长度是2字节时,(R3R4)是余数,(R5R6R7)是商;当除数的长度是1字节时,(R3)是余数,(R4R5R6R7)是商.限于篇

幅,程序里没有做把余数和商放到特定的RAM的步骤,但是相信这并不妨碍读者成功地运用本程序.

3.2 测试结果

图3、图4、图5、图6、图7分别是对除数是0字节(除数为0)、1字节、2字节、3字节、4字节时测试所得的结果.这里被除数都是0FFFFFFFH,4个字节长.

p _{sw}	0x04
p	0
f1	0
ov	1
r _s	0
f0	0
ac	0
cy	0

图3 除数为0字节

r ₃	0x00
r ₄	0x01
r ₅	0x01
r ₆	0x01
r ₇	0x01

图4 除数是单字节

r ₃	0x00
r ₄	0x00
r ₅	0x01
r ₆	0x00
r ₇	0x01

图5 除数是双字节

r ₃	0x00
r ₄	0x00
r ₅	0xff
r ₆	0x01
r ₇	0x00

图6 除数是3字节

r ₃	0x00
r ₄	0x00
r ₅	0x00
r ₆	0x0f
r ₇	0x10

图7 除数是4字节

图3中,除数为0,运算结果为溢出OV为1,结果正确.

图4中,除数为0FFH,运算结果是1010101H,余数为0,结果正确.

图5中,除数为0FFFFH,运算结果是10001H,余数为0,结果正确.

图6中,除数为0FFFFFFH,运算结果是100H,余数为0FFH,结果正确.

图7中,除数为0FFFFFFFFFFH,运算结果是10H,余数为0FH,结果正确.

综上所述,测试结果与实际结果完全一致.

4 结论

试验结果表明,本算法通过巧妙地利用并改进了“移位-相减”法,实现单片机多字节除法的运算,使程序具有较高的可靠性,克服了商溢出及除数长度只能是被除数一半的缺点,并且程序的代码也较为简单,可以移植到其他单片机平台上,具有很高的使用价值.该算法已经应用于笔者设计的项目当中,运行1年余并没有出现过问题.

参考文献:

- [1] 赵伟,王宏志,陈兴良.单片机多字节浮点除法快速扫描运算的实现[J].吉林工学院学报,2000,21(2):8-9.
- [2] 赵慧,张森,冯垛生.TMS320C2XX(DSP)中无符号多字除法程序的实现[J].河南科学,2005,23(1):124-126.
- [3] 孙育才.MCS-51系列单片微型计算机及其应用[M].南京:东南大学出版社,2000.
- [4] 彭秀华,徐爱钧.C语言编程在智能化重力加速度测试仪中的应用[J].江汉石油学院学报,1998,20(1):81-86.
- [5] 周航慈.单片机应用程序设计(修订版)[M].北京:北京航空航天大学出版社,2002.
- [6] 周航慈.单片机程序设计基础(修订版)[M].北京:北京航空航天大学出版社,2003.

General Unsigned Multi-bytes Division Based on Single-chip Assembly Language

YAN Ke-jian, ZHANG Miao, WANG Li-qiong

(Faculty of Automation, Guangdong University of Technology, Guangzhou 510090, China)

Abstract: This paper presents a series of improved algorithm in 8051 to achieve the high reliability computing multi-byte division. The algorithm updates the traditional “shifting-subtracting” algorithm while owns the limitations of the unsigned multi-byte division, and it solves the overflow problem. And via experiments, the accuracy of results proves correct. The algorithm is simple, and with strong adaptability, high reliability and very good practicality.

Key words: SCM; assembler instruction; multi-bytes’ division