

计算机工程学院

**汇编语言与接口技术**

**实验报告**

学年学期：  **2022 - 2023 学年 第1学期**

实验名称：  **实验二 子程序设计**

班级： 计算2114

学号： 202121331104

姓名： 庄佳强

日期： 2022.10.23

成绩：

## 一、 实验要求

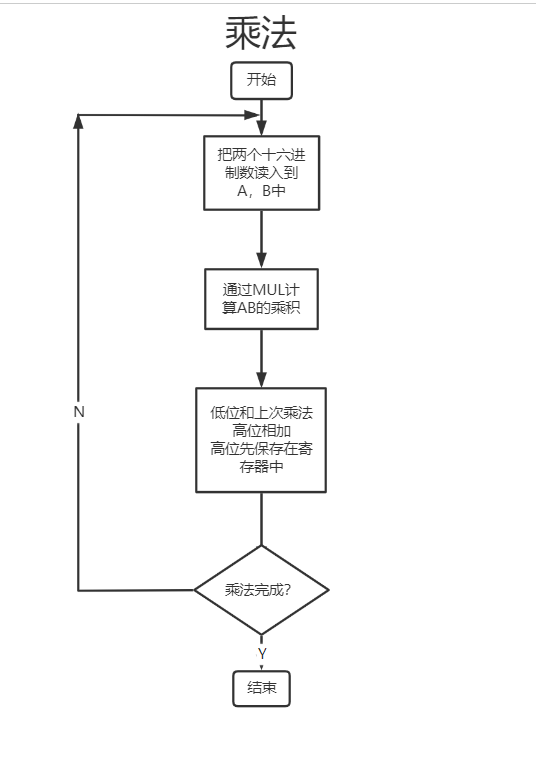
1. 熟练掌握算术运算汇编指令的使用
2. 熟练掌握子程序设计的基本方法
3. 熟练掌握程序的调试方法

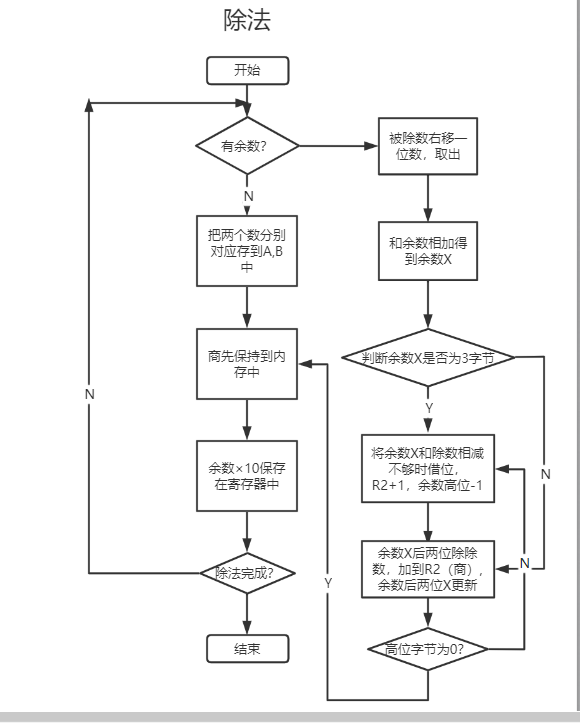
## 二、 实验设计

### 1.整体思路

在把数字保存到程序存储器中后，先把需要运算的十六进制数提取出来到累加器中，再主要通过MUL和DIV再结合一系列操作完成乘除法的运算，最后存储到内存中。

### 流程图





### 主要模块设计思路及分析

先把8888H,79H存储到程序存储器中。

**乘法**：取出88H和79H,到A,B中，79H存储到R7中，方便后续使用。使用MUL AB计算乘积，高位先存放在R1中，低位直接存储到内存22H中。

接着取出88H到A中，R7取到B中，MULAB计算乘积，低位和R1中的相加，有进位就和高位相加，结果低位存放到21H,高位存放到20H.

**除法**：取出88H和R7到A,B中，使用DIV AB 计算商余，商先直接存放到27H中，余数×10,先存放到寄存器R6（低位），R5（高位）中。

再取出 88H中的第一个8，加上余数，组成新的除数。若余数不是三字节，重复上面操作。若是三字节，把余数循环除除数再减除数，使得商R2不断累加，直到完成（余数除尽）。

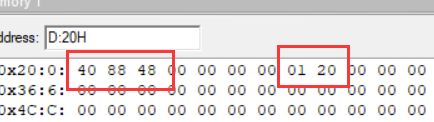
获得商和余数，商先存到28H,余数×10,先存放到寄存器R6（低位），R5（高位）中。

重复第二步操作，得到最后一位商，把28H的商取出×10,加上最后一位商，放到28H中。

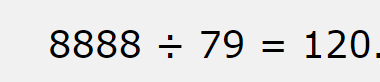
### 

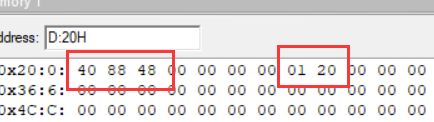
1. **实现效果**
2. 乘法

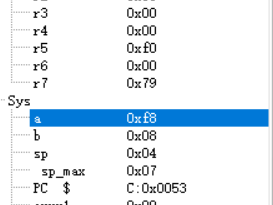




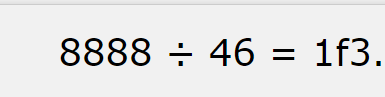
1. 除法

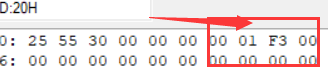






获得第一次除法余数和8组成新的除法 f8/79





可以实现所有除法

1. **总结**

1.很痛苦一开始，乘法很简单，但除法非常难，要考虑位的移动，写了几个小时才完成，然后第二天有想到更好的解法，又写了一遍.....

2.老师的解法一点也看不懂。

3.碰到了跳转bug.....NET直接跳转到0x00F0，一开始没问题，就是写报道再运行两次时候出现的bug,只要就一个地方跳转不了（未解决）。所有那处的跳转先弃用。11.49

4.就在刚刚，解决了bug。。。。12.03

**附录：**

ORG 0000H

NUM1: DW 8888H

NUM2: DB 46H

MAIN:

ACALL MULS

ACALL DIVS

nop

MULS:

PUSH PSW //现场保护

PUSH Acc

MOV DPTR,#NUM1+1 //取88H 到B

MOVC A,@A+DPTR

MOV B,A

CLR A

MOV DPTR,#NUM2 //取79H 到A

MOVC A,@A+DPTR

MUL AB //相乘

MOV 22H,A //低位

MOV R1,B //高位放R1

CLR A

CLR C

MOV DPTR,#NUM1

MOVC A,@A+DPTR

MOV B,A

CLR A

MOV DPTR,#NUM2

MOVC A,@A+DPTR

MUL AB

ADDC A,R1 //低位和上一次乘积高位相加

MOV 21H,A

MOV 20H,B

POP Acc

POP PSW

RET

DIVS:

PUSH PSW //一定要现场保护 ！！！！！

PUSH Acc

MOV DPTR,#NUM2

MOVC A,@A+DPTR

MOV R7,A //把79H放到R7中

CLR A

MOV DPTR,#NUM1

MOVC A,@A+DPTR

MOV B,R7

DIV AB //A/B

MOV 27H,A //低位放27H

LCALL LOOP3

MOV DPTR,#NUM1+1

MOVC A,@A+DPTR

MOV B,#16 //88H/10H =8 为一个位数

DIV AB

CLR C

ADDC A,R5 //低位 x0H +8 ==x8H

MOV B,R7 // x8H/79H

DIV AB

MOV R2,A

MOV R3,B

LCALL LOOP2

MOV 28H,R2

//同理

MOV B,R3

MOV A,B

MOV B,#16

MUL AB

MOV R6,B

MOV R5,A

CLR A

CLR A

MOV DPTR,#NUM1+1

MOVC A,@A+DPTR

SWAP A

MOV B,#16

DIV AB

CLR C

ADDC A,R5

MOV B,R7

DIV AB

MOV R2,A

MOV R3,B

LCALL LOOP2

MOV A,28H //把临时存储的商取出×10H加上之后的商， 存放回去

MOV B,#16

MUL AB

ADD A,R2

MOV 28H,A

POP Acc

POP PSW

AJMP LOOP5

LOOP2:

MOV A,R6

JZ LOOP4 //判断高位余数是否为0

MOV A,R3 //获得

SUBB A,R7 //获得借位

DEC R6 //高位余数-1

INC R2 //商++

MOV B,R7 //

DIV AB //除于获得商

ADD A,R2 //加上

MOV R2,A

MOV R3,B

CJNE R6,#0,LOOP2 //判断是否为0，不为0返回LOOP2

RET //返回

LOOP3:

MOV A,B

MOV B,#16 //高位×10

MUL AB

MOV R6,B //余数高位放R6

MOV R5,A //余数低位放R5

CLR A

//不知道为什么会直接跳转到0X00F0 ，第一次运行时没有错误，但第二次运行就出现了这个bug。。。。 ,只有这个跳转不行。。。。。

LOOP4: //返回跳转

RET

LOOP5: //结束跳转

END