实验 2 串操作实验报告

1.实验目的

（1）掌握串操作指令的使用；

（2）理解算数运算指令、BCD 码调整指令；

（3）熟练应用 DEBUG 调试汇编程序；

2.实验内容

题目：使用串操作指令 MOVSB 对一段内存单元中的内容（1，2，3，……，100） 进行转移，再使用串操作指令 CMPS 对转移的内容进行比较来判断传输是否正 确，若不正确则进行重新传输；接着对已经正确传输的 100 个数据进行无符 号型的累加，最后使用 BCD 调整码，最终将答案放入内存，并将其显示在屏幕上。

3.实验要求

（1）上机实验前，仔细复习课本有关知识；

（2）独立完成实验，画出流程图并上交实验报告；

4.实验步骤

1．）算法分析 从实验的内容分析可知，要完成如下实验，可分为以下步骤：

（1）将 1，2，3，……，100 存入数据段相应内存中；

（2）转移字符串并比较；

（3）数据累加并调整。

2．）算法设计

（1）数据的存入、转移与比较

已知： a. MOVSB 指令的目标操作数与源操作数的逻辑地址由 ES:DI 和 DS:SI 指出；

b. 串传送指令常与无条件重复前缀连用；

c. 无条件重复 REP，仅仅判断 CX 是否为 0；

d. 串比较指令常与条件重复前缀连用，指令的执行不改变操作数， 仅影响标志位。

注意：

（1）在使用串操作指令时需要修改 flag 寄存器当中的 DF 位（方 向位），来确定串操作的进行方向，具体表现为：CLD 使 DF=0 增地址方 向；STD 使 DF=1 减地址方向；

（2）数据的累加与调整 BCD 码调整指令 AAM 用来调整寄存器 AX 当中的值，将 AL/10 的商放 在 AH 高位中，余数放在 AL 低位当中进行保存。 将结果答案显示到屏幕上时，需要的是数字的 ASCII 码，因此需要 ADD AX,3030H。

知识回顾：

8086 汇编语言指令系统中提供了 5 种串处理指令。分别是：

MOVS (move string)	串传送
CMPS (compare string)	串比较
SCAS (scan string)	串扫描
LODS (load string)	串获取
STOS (store string)	串存入

上述串指令应该和重复前缀 REP、REPZ/REPE、REPNZ/REPNE 结合.

1.串操作的指令默认目的串的段寄存器为 es 附加段，源串的端寄存器为数据段 ds（当然，可能出现附加段和数据段为同一段的情况，可以用 assume 进行设定）

2.目的串的偏移地址由 di 寄存器给出，源串的偏移地址由 si 寄存器给出。传送次数由 cx 给出。rep 前缀功能为：重复串操作直到(cx)=(cx)-1=0

3.分别给出两钟等效的说法

①rep movs byte ptr es:[di],ds:[si]

= rep movsb ;隐式地指出源串和目的串的地址和属性;以字节形式

②rep mov word ptr es:[di].ds:[si]

= rep movsw;隐式地指出源串和目的串的地址和属性;以字形式

关于串比较 cmps dest,source：

Repz/repe 前缀功能为:结果为 0 或相等就重复操作，若结果不为 0 或不相等提前推出重复操作，此时 cx 还没有减为 0，si 和 di 已经增量。（常用来检测某一字符串与另一字符串是否完全相同）

Repnz/repne 前缀功能为:结果不为 0 或不相等就重复操作，若结果为 0 或相等提前推出重复操作，此时 cx 还没有减为 0，si 和 di 已经增量(用来寻找字或字节，找到即停止)

程序说明：

1.两个 showmsg 和 enter 子程序分别用来显示提示字符串和显示回车符。

2.本题实际操作中并没有用到 aam 指令，（由于 aam 指令要求乘积不能超过 99，而本题答案是 5050）

3.过程

5050/10=01f9h(商 505)(ax=01f9h)-----余 0(dx=0) pop dx

505/10=0032h(商 50)(ax=0032h)-------余 5(dx=5) pop dx

50/10=0005h(商 5)(ax=0005h)----------余 0(dx=0) pop dx

5/10=0(商 0)(ax=0h)----------余 5(dx=0) pop dx

依次将 pop 出来依次得到 5,0,5,0;分别加上 30h 得到相应的 ASCII 码，用 2 号功能显示输出。

4.子程序格式

```
Zichenxu proc
.............
   ret
Zichenxu endp
```

5.本程序中由于被除数 ax 是 16 位寄存器，除数 bp 也是 16 位，故余数放 dx 中，每次除完的商放在 ax 中。

关于商和余数存放位置的总结：

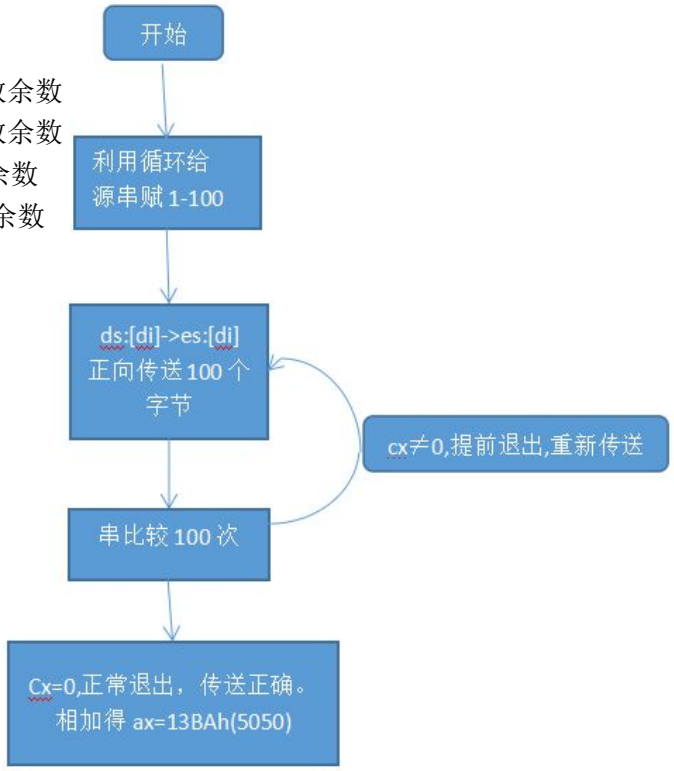
①div r8/r16 ;无符号字节除;al 放商，ah 放余数

idiv r8/r16 ;有符号字节除;al 放商，ah 放余数

②div r16/m16 ;无符号字除 ;ax 放商，dx 放余数

idiv r16/m16 ;有符号字除 ;ax 放商，dx 放余数

6.流程图:



7.源代码:

Twopothead 实验2串操作指令的使用

1 contributor

84 lines (81 sloc) | 2.36 KB

```
data segment
  msg  db '1+2+3+4+...+100=',0ah,0dh,'$'
  source db 100 dup(0)   ;源串
data ends
extra segment
  dest db 100 dup(0)     ;目的串
extra ends
stack segment
   db 30 dup(0)
stack ends
assume cs:code,ds:data,es:extra,ss:stack
code segment
showmsg proc           ;showmsg子程序调用dos9号中断，显示字符串
    mov ah,9
    mov dx,offset msg  ;AH=9,DS:DX=字符串地址
    int 21h
  ret
showmsg endp
enter proc            ;enter子程序显示回车字符
    mov ah,2          ;2号功能是单字符显示输出
    mov dl,0dh        ;回车ASCII码送dl
    int 21h
  ret
enter endp
start : mov ax,stack
        mov ss,ax
        mov sp,30     ;设栈顶指针
        mov ax,extra
        mov es,ax
        mov ax,data
        mov ds,ax
        call showmsg
        call enter
        mov bx,0            ;bx放源串的偏移地址
        mov al,1
        mov cx,100
    s : mov source[bx],al   ;给源串写入1—100的数据
         inc al
         inc bx
        loop s
  copy: lea si,source
        lea di,dest
        cld                 ;传送方向为正，传送100个字节
        mov cx,100
        rep movsb

        lea si,source
        lea di,dest
        cld                 ;串比较方向为正，比较100次
        mov cx,100
        repz cmpsb          ;结果为0或相等就重复操作
        cmp cx,0            ;若cx不为0，那一定是提前退出的
        jnz copy            ;即两个串并不完全相同，需要重新传送
        mov cx,100
        xor ax,ax           ;ax清零
        xor bx,bx           ;bx清零
        mov di,0
        plus:mov bl,byte ptr es:dest[di]
        add ax,bx
        inc di
        loop plus
        ;此时加完，ax里保存着结果的数值，接下来转十进制
        ;再用ASCII码输出显示

        mov bp,10
        xor bx,bx
        xor cx,cx
  bin2decimal  :mov dx,0
                inc cx
                ;push多少次，就要pop多少次（again循环）
                idiv bp     ;有符号字除 r16/m16 (ax)/(bp)
                push dx     ;idiv r16/m16 余数放在dx,商放在ax
                cmp ax,0
        jnz bin2decimal
        mov ah,2
  again:    pop dx
            add dl,30h  ;(dx)的数值加上30h,得到它的ASCII码
            int 21h     ;2号中断显示输出
          loop again

  mov ax,4c00h
  int 21h
code ends
end start
```