

微机原理实验一 数据传送

2021/11/17

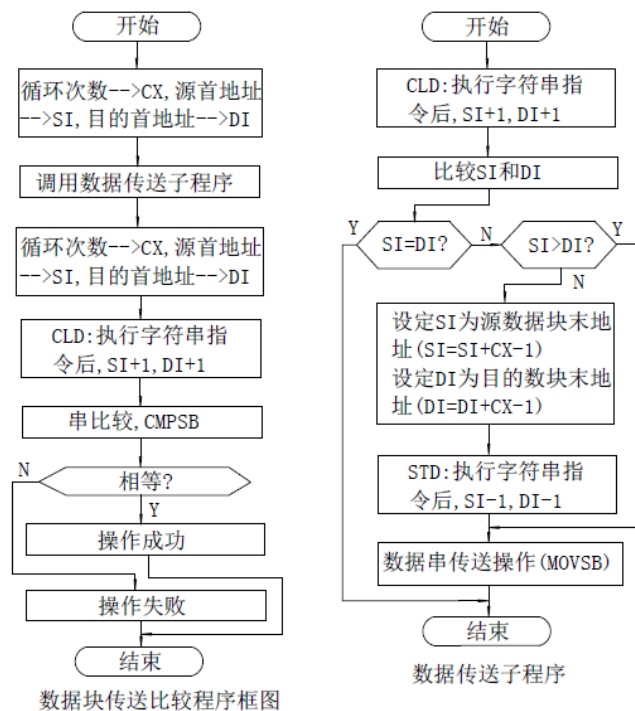
一、实验目的

熟悉星研集成环境软件的使用方法。熟悉 Borland 公司的 TASM 编译器
熟悉 8086 汇编指令，能自己编写简单的程序，掌握数据传输的方法。

二、实验内容

1、熟悉星研集成环境软件。2、编写程序，实现数据段的传送、校验。

三、程序框图



四、实验步骤

在 DS 段内 3000H~30FFH 中输入数据；使用单步、断点方式调试程序，检测 DS 段内 6000H ~ 60FFH 中的内容。熟悉查看特殊功能寄存器、CS 段、DS 段的各种方法。

五、思考题

1、子程序 Move 中为什么比较SI、DI？

通过比较SI，DI，如果SI>DI则从数据块首开始传送，反之从数据块尾传送，避免地址交叉导致丢失数据。

六、实验结果

```
_STACK      SEGMENT      STACK
              DW          100 DUP(?)

_STACK      ENDS
DATA        SEGMENT
DATA        ENDS
CODE        SEGMENT
START       PROC          NEAR
              ASSUME      CS:CODE, DS:DATA, SS:_STACK
              MOV         AX, DATA
              MOV         DS, AX
              MOV         ES, AX      ;初始化段寄存器
              NOP
              MOV         CX, 100H   ;将循环次数CX中存入100H
              MOV         SI, 3000H  ;将源首地址SI中存入3000H
              MOV         DI, 6000H  ;将目的首地址DI中存入6000H
              CALL        Move       ;执行MOVE函数
              MOV         CX, 100H   ;将循环次数CX中存入100H
              MOV         SI, 3000H  ;将源首地址SI中存入3000H
              MOV         DI, 6000H  ;将目的首地址DI中存入6000H
              CLD                  ;执行字符串指令后, SI+1, DI+1
              REPE        CMPSB      ;串比较
              JNE         ERROR      ;不等于就跳转
TRUE:        JMP $
ERROR:       JMP $
Move        PROC          NEAR
              CLD                  ;执行字符串指令后, SI+1, DI+1
              CMP         SI, DI     ;比较SI, DI
              JZ          Return     ;若相等则跳转Return
              JNB         Move1      ;若SI>DI跳转MOVE1
              ADD         SI, CX      ;设定SI为源数据块末地址(SI=SI+CX-1)
              DEC         SI
              ADD         DI, CX      ;设定DI为目的数据块末地址(DI=DI+CX-1)
              DEC         DI
              STD          ;STD:执行字符串指令后, SI-1, DI-1
Move1:       REP         MOVSB       ;传送数据串
Return:      RET
Move        ENDP          ;结束MOVE子程序
START       ENDP          ;结束START子程序
CODE        ENDS
```

● 寄存器数值变化

_STACK	SEGMENT	STACK
	DW	100 DUP(?)
_STACK	ENDS	
DATA	SEGMENT	
DATA	ENDS	
CODE	SEGMENT	;CF = ZF = SF = OF = PF = 0, IF = 1
START	PROC	NEAR
	ASSUME	CS:CODE, DS:DATA, SS:_STACK
	MOV	AX, DATA
	MOV	DS,AX
	MOV	ES,AX
	NOP	
	MOV	CX,100H
	MOV	SI,3000H
	MOV	DI,6000H
	CALL	Move
	MOV	CX,100H
	MOV	SI,3000H
	MOV	DI,6000H
	CLD	
	REPE	CMPSB ;ZF = 1
	JNE	ERROR
TRUE:	JMP \$	
ERROR:	JMP \$	
Move	PROC	NEAR
	CLD	
	CMP	SI,DI ;CF = 1, SF = 1, PF = 1
	JZ	Return
	JNB	Move1
	ADD	SI,CX ;CF = 0, SF = 0, PF = 1
	DEC	SI ;CF = 0, SF = 0, PF =1, AF = 1
	ADD	DI,CX ;CF = 0, SF = 0, PF = 1, AF = 0
	DEC	DI ;CF = 0, SF = 0, PF = 0, AF = 1