****

**实验一 汇编语言程序设计**

1. **实验目的**

1.掌握8088/8086汇编语言的基本语法结构；

2.熟悉8088/8086汇编语言程序设计基本方法。

**二、实验设备**

装有emu8086软件的PC机

**三、实验内容**

1.有一个10字节的数组，其值分别是80H，03H，5AH，FFH，97H，64H，BBH，7FH，0FH，D8H。编程并显示结果：

1)如果数组是无符号数，求出最大值，并显示；

2)如果数组是有符号数，求出最大值，并显示。

2.将二进制数500H转换成二－十进制（BCD）码，并显示：

“500H的BCD是：”

3.将二－十进制码（BCD）7693转换成ASCII码，并显示：

“BCD码7693的ASCII是：”

4.两个长度均为100的内存块，先将内存块1全部写上88H，再将内存块1的内容移至内存块2。在移动的过程中，以16进制－ASCII码显示移动次数1， 2 ，3…0AH…64H。

5.键盘输入一个小写字母(a~z)，转换成大写字母

显示：

请输入一个小写字母(a~z)：

转换后的大写字母是：

6.实现4字节无符号数加法程序，并显示计算结果，如：

99223344H + 99223344H = xxxxxxxxH

1. **实验代码及运行结果**

1.1、实验代码：

DATA SEGMENT

SZ DB 80H,03H,5AH,0FFH,97H,64H,0BBH,7FH,0FH,0D8H;存进数组

SHOW DB 'THE MAX IS: ','$'

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE,DS:DATA

START:

MOV AX,DATA ;把数据的基地址赋给DS

MOV DS,AX

MOV DX,OFFSET SHOW ;调用DOS显示字符串

MOV AH,09H

INT 21H

MOV SI ,OFFSET SZ ;数组的偏移地址赋给SI

MOV CX,10 ;存进数组的长度给CX

MOV DH,80H ;将数组的第一个数写进DH

NEXT:

MOV BL,[SI] ;将数组的第一个数写进BL

CMP DH,BL ;比较DH和BL中数的到校

JAE NEXT1 ;如果DH中的数大于BL中，将跳转到NEXT1

MOV DH,BL ;如果DH中的数小于BL中，将BL中的数赋给DH

NEXT1:

INC SI ;偏移地址加1

LOOP NEXT;循环，CX自减一直到0，DH中存数组的最大值

;接下来的程序是将将最大值DH在屏幕上显示输出

MOV BX,02H

NEXT2:

MOV CL,4

ROL DH,CL ;将DH循环右移四位

MOV AL,DH

AND AL,0FH ;将AL高位清零，一位一位进行ASCII码转换

CMP AL,09H ;与09H比较大小

JBE NEXT3 ;若小于，跳转至NEXT3，证明数值在0-9之间

ADD AL,07H ;若大于，则证明为字母，因为字母与数字的ASCII码相差7，所以要加上7

NEXT3:

ADD AL,30H ;因为转换为ASCII码要加30H

MOV DL,AL

MOV AH,2 ;DOS中断显示

INT 21H

DEC BX

JNZ NEXT2

MOV DL,'H'

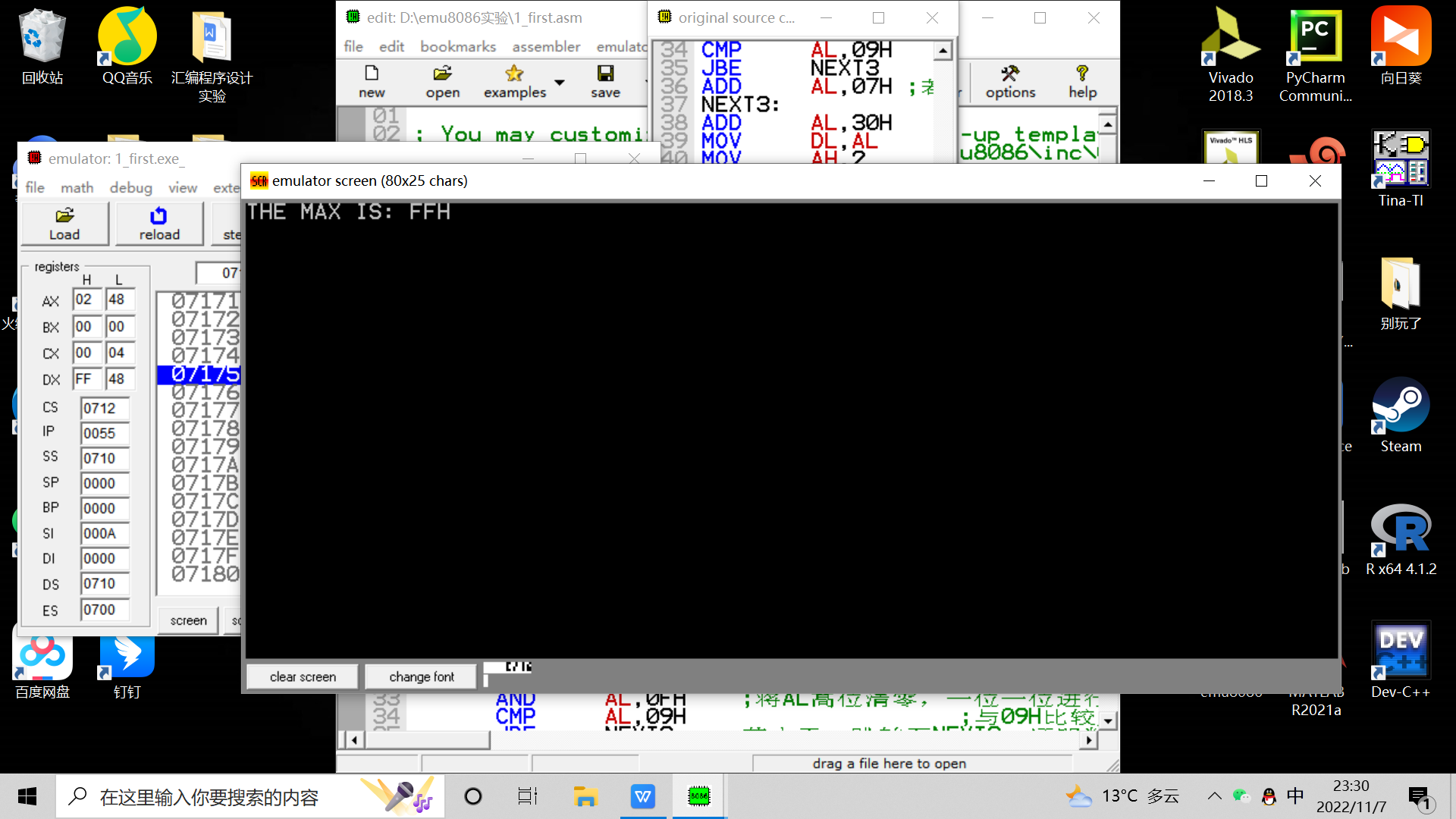
MOV AH,2

INT 21H

CODE ENDS

END START

实验结果：



1.2、实验代码

DATA SEGMENT

SZ DB 80H,03H,5AH,0FFH,97H,64H,0BBH,7FH,0FH,0D8H ;存进数组

SHOW DB 'THE MAX IS: ','$'

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE,DS:DATA

START:

MOV AX,DATA ;把数据的基地址赋给DS

MOV DS,AX

MOV DX,OFFSET SHOW ;调用DOS显示字符串

MOV AH,09H

INT 21H

MOV SI ,OFFSET SZ ;数组的偏移地址赋给SI

MOV CX,10 ;存进数组的长度给CX

MOV DH,80H ;将数组的第一个数写进DH

NEXT:

MOV BL,[SI] ;将数组的第一个数写进BL

CMP DH,BL ;比较DH和BL中数的到校

JGE NEXT1 ;如果DH中的数大于BL中，将跳转到NEXT1

;此为无符号数与有符号数的唯一差别，JGE是无符号数的跳转

MOV DH,BL ;如果DH中的数小于BL中，将BL中的数赋给DH

NEXT1:

INC SI ;偏移地址加1

LOOP NEXT ;循环，CX减1,一直到CX减为0，DH中存数组的最大值

;接下来的程序是将最大值DH在屏幕上显示输出

MOV BX,02H

NEXT2:

MOV CL,4

ROL DH,CL ;将DH循环右移四位

MOV AL,DH

AND AL,0FH ;将AL高位清零，一位一位进行ASCII码转换

CMP AL,09H ;与09H比较大小

JBE NEXT3 ;若小于，跳转至NEXT3，证明数值在0-9之间

ADD AL,07H;若大于，则证明为字母，因为字母与数字的ASCII码相差7，所以要加上7

NEXT3:

ADD AL,30H ;因为转换为ASCII码要加30H

MOV DL,AL

MOV AH,2 ;DOS中断显示

INT 21H

DEC BX

JNZ NEXT2

MOV DL,'H'

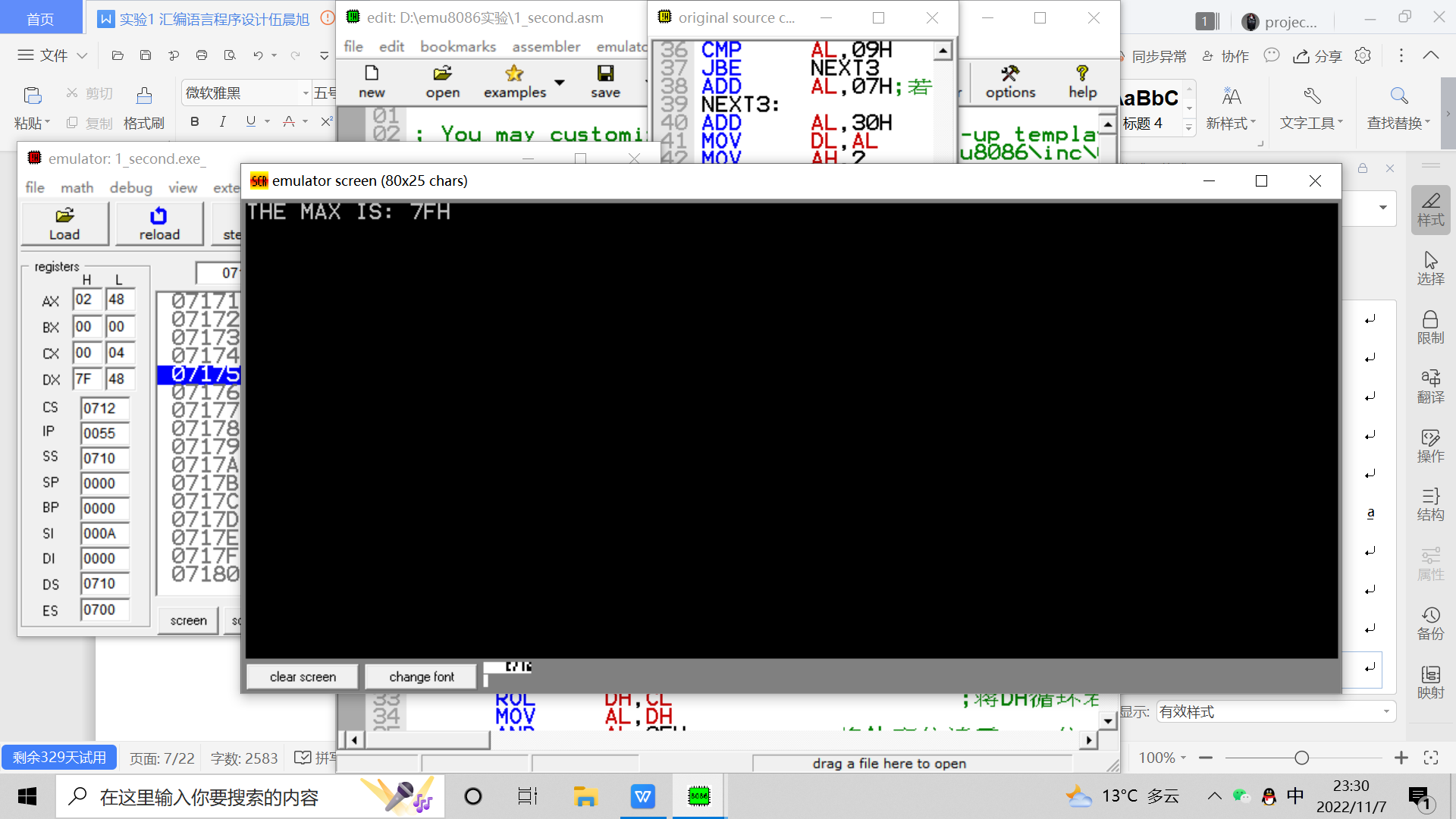
MOV AH,2

INT 21H

CODE ENDS

END START

实验结果：



2、实验代码

DATA SEGMENT

BNUM DW 0500H

DNUM DB 4 DUP()

MES DB 'THE BCD OF '500H' IS: ','$'

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE,DS:DATA

START:

MOV AX,DATA

MOV DS,AX

MOV DX,OFFSET MES ;将字符串偏移量写入DX

MOV AH,09H

INT 21H

MOV AX,BNUM ;将0500H存入AX中

LEA BX,DNUM ;取DNUM的偏移量

MOV DL,0

;计算1000的个数

COUNT1:

SUB AX,03E8H ;AX-1000

JC NEXT1 ;不够减，即若CF=1，有借位，转NEXT1

INC DL ;DL+1

JMP COUNT1 ;跳转AGAIN1

NEXT1:

ADD AX,03E8H ;AX+1000

MOV [BX],DL ;将DL(即千位数）存入DNUM中

INC BX ;BX+1

MOV DL,0 ;DL清零

;计算100个数

COUNT2:

SUB AX,0064H ;AX-100

JC NEXT2

INC DL

JMP COUNT2

NEXT2:

ADD AX,0064H

MOV [BX],DL ;将DL(即百位数）存入DNUM中

INC BX

MOV DL,0

;计算10个数

COUNT3:

SUB AX,000AH ;AX-10

JC NEXT3

INC DL

JMP COUNT3

NEXT3:

ADD AX,000AH

MOV [BX],DL ;将DL(十位数)存入DNUM中

INC BX

MOV [BX],AL ;计算个位数

;接下来的程序是将数值转化为ASCII并在屏幕上显示

MOV CX,4

LEA SI,DNUM ;取DNUM的地址偏移量

NEXT4:

MOV AL,[SI] ;将DNUM中所存的各个数值转换为ASCII码；

;在屏幕上输出显示

CMP AL,09H

JBE NEXT5

ADD AL,07H

NEXT5:

ADD AL,30H

MOV DL,AL

MOV AH,2

INT 21H

INC SI

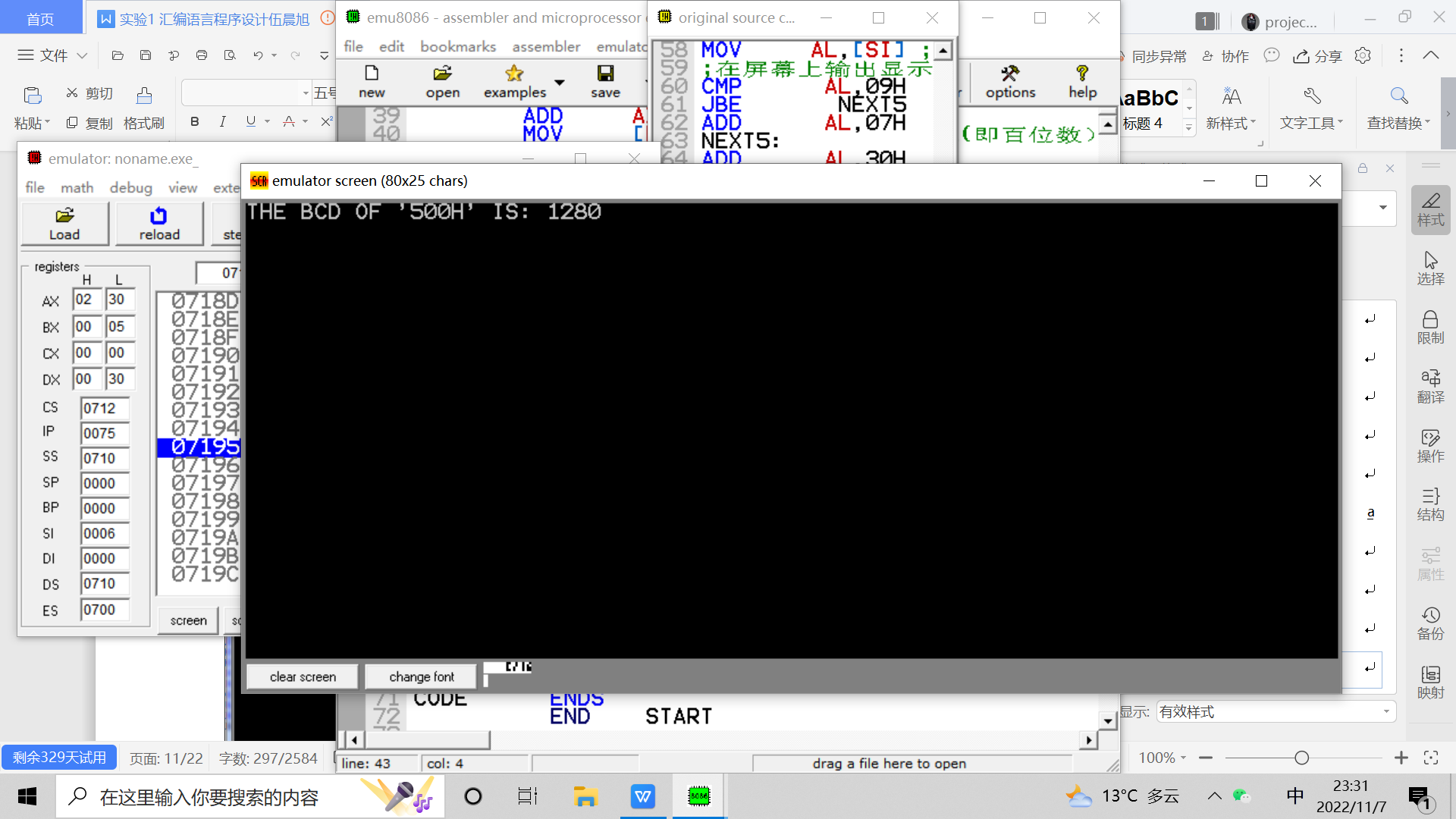
DEC CX

JNZ NEXT4

CODE ENDS

END START

实验结果



3、实验代码

DATA SEGMENT

OUP DB 'THE ASCII OF 7963 IS:','$'

A1 DB 07H,09H,06H,03H

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE,DS:DATA

START:

MOV AX,DATA

MOV DS,AX

MOV DX,OFFSET OUP

MOV AH,09H

INT 21H

MOV CL,4

MOV SI,0

NEXT:

MOV AL,A1[SI] ;将A1的第SI个数据传给AL

MOV BL,AL

OR AL,30H ;加30转换为ASCII码值

SHR AL,4

AND AL,0FH ;将右移后的高位清零

ADD AL,30H ;因为ASCII码与值之间相差30

MOV DL,AL

MOV AH,2 ;输出对应的ASCII码

INT 21H

MOV AL,BL

AND AL,0FH

ADD AL,30H

MOV DL,AL

MOV AH,02H

INT 21H

MOV DL,' '

MOV AH,02H

INT 21H

INC SI

LOOP NEXT

EXIT:

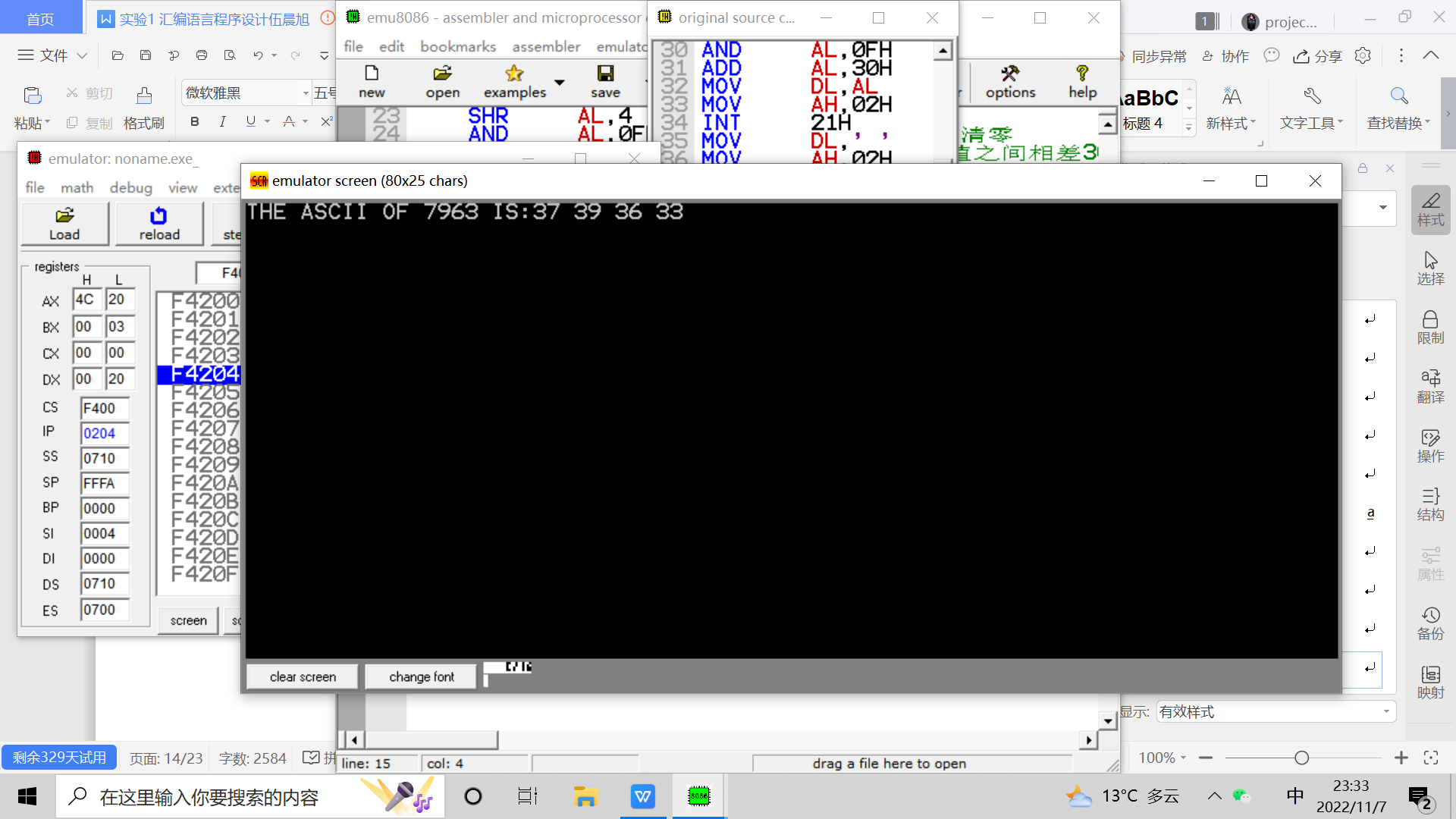
MOV AH,4CH

INT 21H

CODE ENDS

END START

实验结果



4、实验代码

DATA SEGMENT

YI DB 100 DUP(88H)

ER DB 100 DUP(00H) ;定义数组

MES DB 'THE NUMBER OF COUNTER IS: ','$'

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE,DS:DATA

MAIN PROC FAR

START: MOV AX,DATA

MOV DS,AX

MOV CL,100 ;计数器初值

MOV CH,00H ;转移次数统计，初值

LEA SI,YI ;将内存块1的偏移量给SI

LEA DI,ER ;将内存块2的偏移量给DI

NEXT: MOV AL,[SI]

MOV [DI],AL ;将YI中的数据复制到ER中

INC SI

INC DI

INC CH

MOV DX,OFFSET MES ;显示字符串

MOV AH,9

INT 21H

CALL COUNT ;调用子程序

DEC CL ;计数减1

JNZ NEXT ;计数不为零，跳转执行

HLT

MAIN ENDP ;主程序结束

;下面是子程序，实现屏幕显示转移次数功能

COUNT PROC NEAR ;定义子程序属性

PUSH CX ;CX进栈

MOV DH,CH

MOV BX,02H

NEXT2: MOV CL,4

ROL DH,CL ;将DH循环右移四位

MOV AL,DH

AND AL,0FH ;将AL高位清零

CMP AL,09H ;与09H比较大小

JBE NEXT3 ;若小于，跳转至NEXT3

ADD AL,07H

NEXT3: ADD AL,30H ;加30H

MOV DL,AL

MOV AH,2 ;DOS中断显示

INT 21H

DEC BX

JNZ NEXT2

MOV DL,'H' ;显示H

MOV AH,2

INT 21H

MOV DL,0AH ;回车

MOV AH,2

INT 21H

MOV DL,0DH ;换行

MOV AH,2

INT 21H

POP CX ;CX出栈

RET

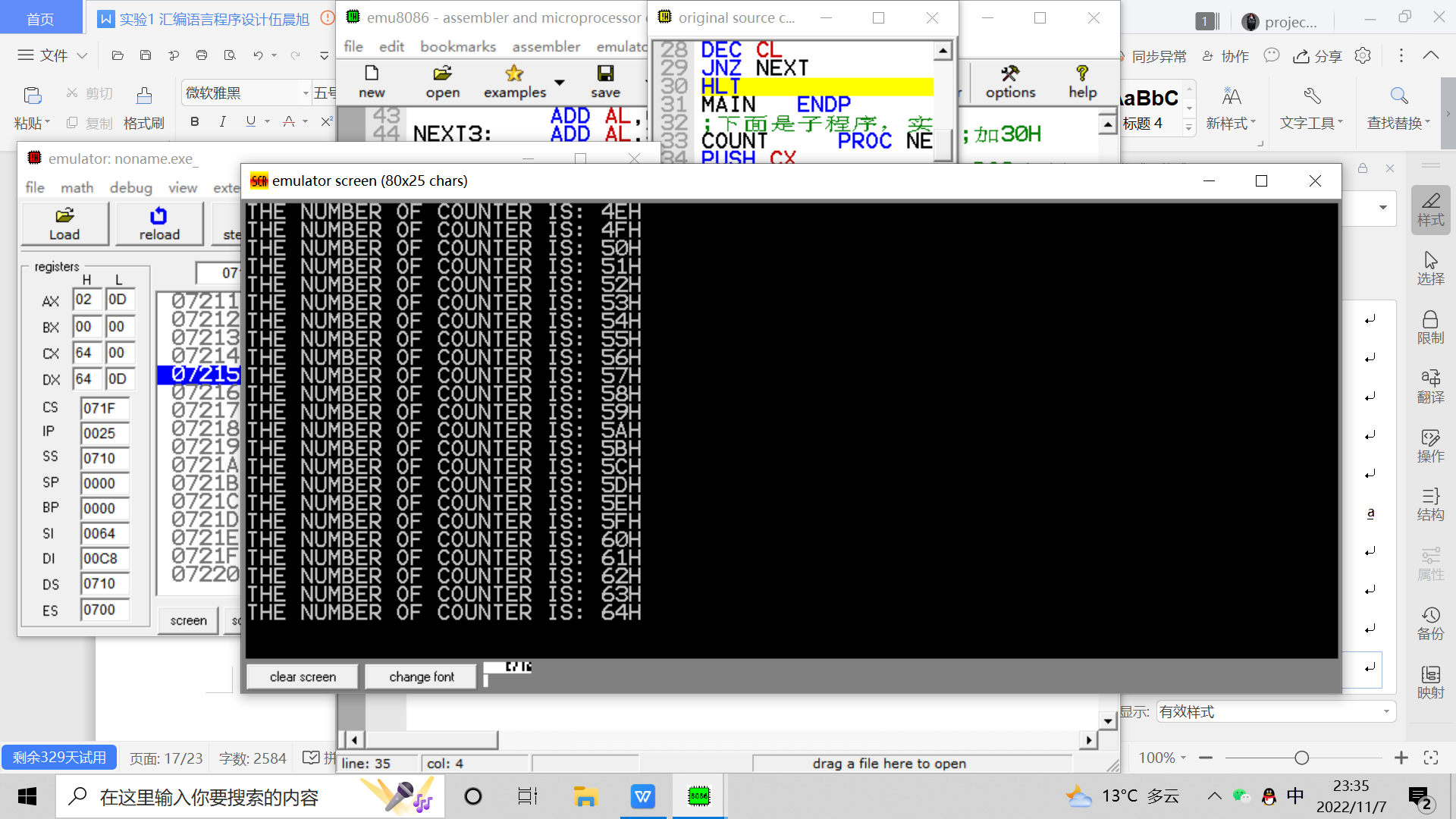
COUNT ENDP

;子程序结束

CODE ENDS

END START

实验结果



5、实验代码

DATA SEGMENT

INP DB 'PLEASE INPUT (a~z): ','$' ;设置字符串

OUP DB 'THE TRANSFORMATION(A~Z) IS:','$'

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE,DS:DATA

MAIN PROC FAR

START: MOV AX,DATA

MOV DS,AX

MOV DX,OFFSET INP ;显示字符串，提示输入

MOV AH,9

INT 21H

MOV AH,1 ;DOS中断调用，从键盘中读取一个字符

INT 21H

PUSH AX ;AX入栈

PUSH DX

MOV DL,0DH ;中断调用，回车

MOV AH,2

INT 21H

MOV DL,0AH ;中断调用，换行

MOV AH,2

INT 21H

MOV DX,OFFSET OUP ;显示字符串，输出提示

MOV AH,9

INT 21H

POP DX ;DX出栈

POP AX ;AX出栈

SUB AL,32 ;AL减去32，使大写转化为小写

;大小写之间的ASCII码值相差32

MOV DL,AL ;DOS中断显示结果

MOV AH,2

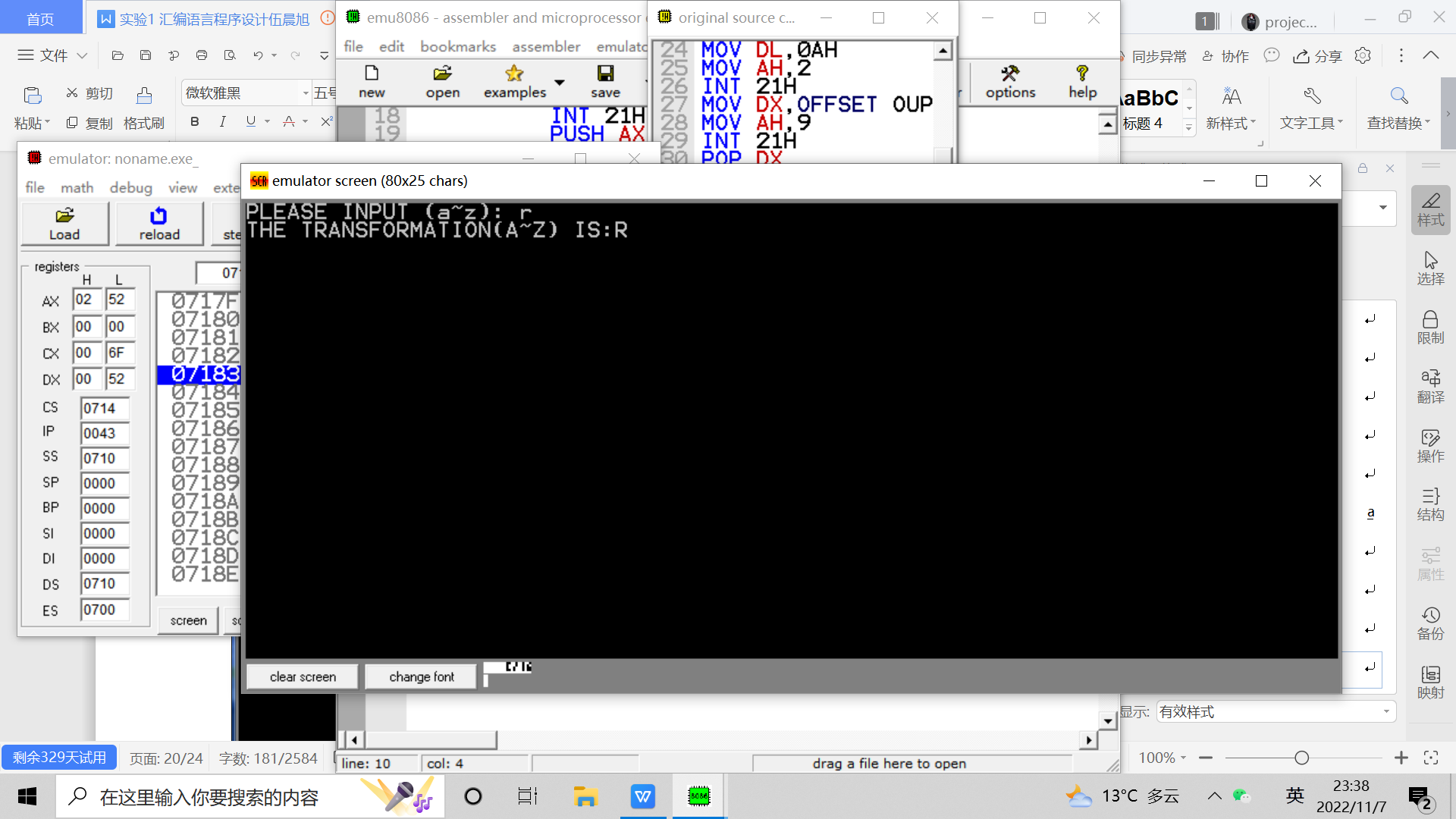
INT 21H

MAIN ENDP

CODE ENDS

END START

实验结果：



6、实验代码

DATA SEGMENT

YI DB 44H,33H,22H,99H,00H

ER DB 44H,33H,22H,99H,00H ;装载数据,必须有后面00H

SUM DB 5 DUP()

OUP DB '99223344H + 99223344H= ','$'

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE,DS:DATA

MAIN PROC FAR

START: MOV AX,DATA

MOV DS,AX

MOV DX,OFFSET OUP ;显示字符串

MOV AH,9

INT 21H

LEA BP,YI ;在BP中装入YI的地址偏移量

LEA SI,ER ;在SI中装入ER的地址偏移量

LEA DI,SUM ;在DI中装入SUM的地址偏移量

MOV CX,05 ;计数初值

CLC ;进位位清零

NEXT: MOV AL,[BP] ;将YI中的数据赋给AL

MOV AH,[SI] ;将ER中的数据赋给BL

ADC AL,AH ;YI和ER中的数据带进位的相加

MOV [DI],AL ;将相加的结果保存在SUM中

INC SI

INC DI

INC BP

DEC CX

JNZ NEXT ;结果不为零，跳转到NEXT

;上面的程序已经将相加的结果存在了SUM中，下面的程序是将SUM中的结果转化为ASCII显示

MOV CH,05 ;计数器初值

NEXT1: MOV DH,[DI-1] ;将所得结果最高位赋给DH

MOV BX,02H

NEXT2: MOV CL,4

ROL DH,CL ;将DH循环右移四位

MOV AL,DH

AND AL,0FH ;将AL高位清零

CMP AL,09H ;与09H比较大小

JBE NEXT3 ;若小于，跳转至NEXT3

ADD AL,07H

NEXT3: ADD AL,30H ;加30H

MOV DL,AL

MOV AH,2 ;DOS中断显示

INT 21H

DEC BX

JNZ NEXT2

DEC DI

DEC CH

JNZ NEXT1 ;循环显示SUM中的值

MOV DL,'H' ;显示H

MOV AH,2

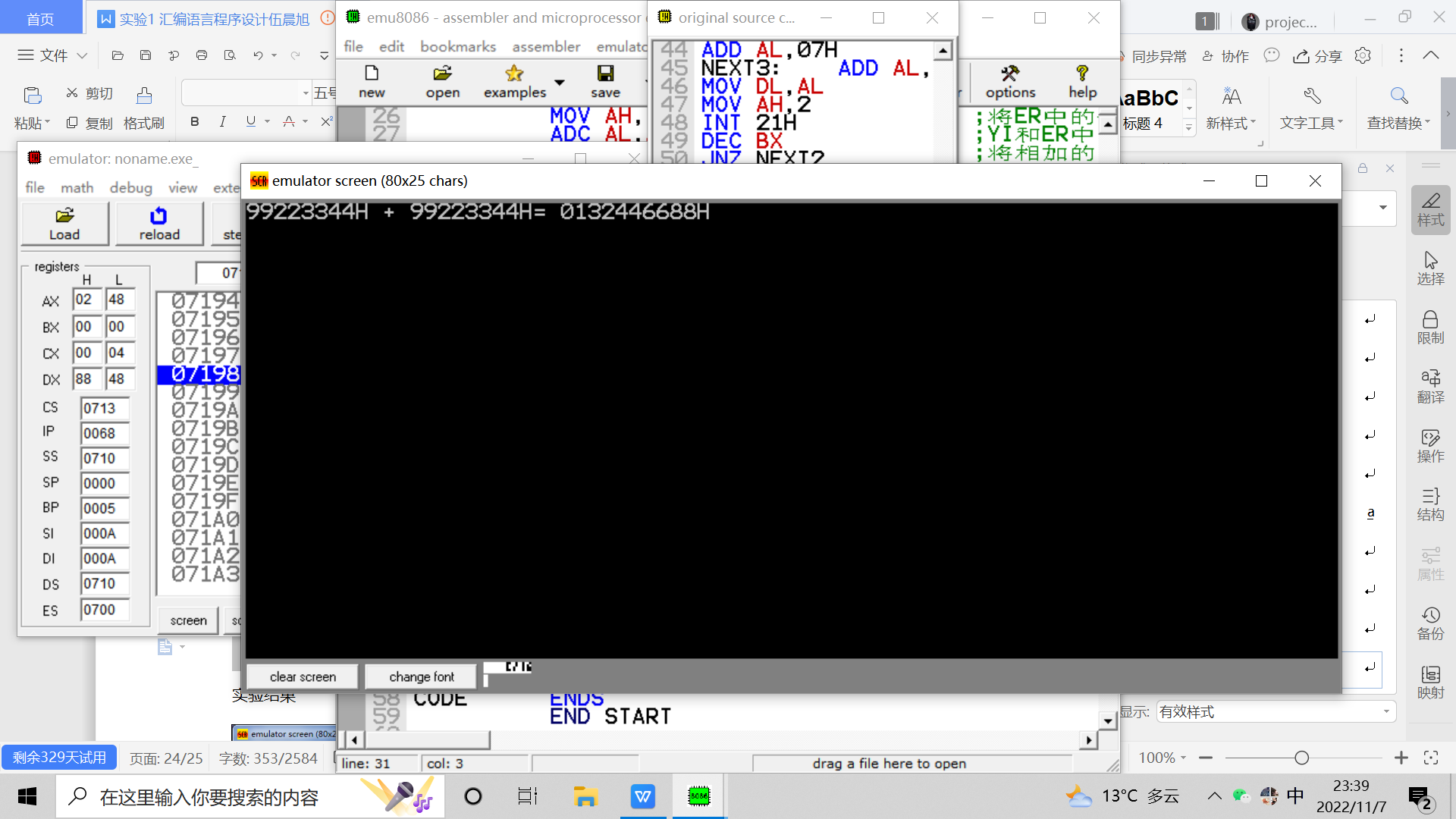
INT 21H

MAIN ENDP

CODE ENDS

END START

实验结果



五、分析与总结

通过实验熟练了8086的指令系统和基本的程序设计格式，实验中出现的各种问题通过一步步的分析自行解决的过程，提高了对于各种语句、指令的运用熟练程度。汇编语言程序运行速度快、占用资源少，可以充分利用硬件资源，设计完成任意功能的程序。但学习、编程较难，编程效率低，兼容性差。因此，主要用于系统软件和控制软件的设计。远不如高级语言应用广泛。

指令语句：包含在指令集中，每一个语句汇编后产生一组二进制数目标代码，可被CPU执行。伪指令语句：汇编后不产生目标代码，用于定义段、子程序、常量、变量，以及给变量分配存储单元。宏指令语句：用户定义的一个指令序列集合， 汇编后在引用的地方展开成相应的指令序列。用于提高编程效率

助教老师在验收时提出了问题，也就是第四题的“YI DB 100 DUP(88H）ER DB 100 DUP(00H) ”这两段代码的意思，是数据定义伪指令的一部分，DB为双字节，100为单元数量，DUP（88H)是重复单元，共100个重复单元。这些部分通过老师的ppt可以清晰地知道。

然后就是一段代码必须得有数据定义伪指令，不然这个代码是运行不出来结果的。