**微机原理与汇编语言**

**实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 姓 名 | 胡聿鑫 | 学 号 | 2220192813 | 专业班级 | 计算机科学与技术2019-3班 | |
| 课程名称 | 微机原理与汇编语言 | | | | 实验日期 | 2021年11月1日 |
| 实验名称 | 实验四 宏指令及子程序设计实验 | | | | 成 绩 |  |

1. 实验目的

①熟悉宏指令、宏定义、宏调用以及宏展开的概念，掌握宏指令的定义与调用的方法。

②掌握子程序的定义、调用以及调用程序与子程序之间参数的传递方法。

③了解宏指令与子程序的异同以及各自的适用场合。

1. 实验内容
2. 实验内容方案描述

从键盘输入 10 个无符号十进制数（小于 256），将其转换为二进制数并存放在 NUM 字节型变量中，

找出其中的最大数，并将找出的最大数在屏幕上显示出来。

要求：

1. 在屏幕上显示字符串提示信息的功能由宏指令 DSTRING 实现；
2. 将键盘输入的十进制数转换成二进制数由子程序 DTOB 实现；
3. 在 N 个无符号字节型二进制数中找出最大数的功能由子程序 FMAX 实现；

④ 将一个无符号字节型二进制数转换为十进制数并在屏幕上显示的功能由子程序 BTOAD 实现。

实验的关键在于如何利用堆栈传递参数，而对于具体的功能（十进制和二进制之间的转换，N个数中的最大值）前面的实验中均已经实现。

要想实现参数的传递，首先需要利用堆栈，将事先存入的数据入栈（此处为AX，CX）。子程序FMAX可以通过访问堆栈，从而获得主程序中的参数。由于指针SP始终指向栈顶，因此可以先使BP入栈，并将SP的值赋给BP，借助BP访问寄存器AX，CX。在结束调用FMAX后，需要使SP指向AX的上一单元，这相当于使得CX，AX出栈。



图1 BP访问堆栈示意图

程序实现的流程图如下：



图2 实验内容程序流程图

1. 实验习题1方案描述

从键盘输入 N 个十进制数，求它们的和（累加和要求不大于 65535），并将累加结果在屏幕上显示出来。要求给出必要的提示信息（用宏调用完成）；累加功能由子程序调用实现；二进制数形式的累加和转换为十进制数并显示由子程序调用实现。

实验习题1的程序编写过程与实验内容整体思路一致，只需要将子程序FMAX修改为具有累加功能的子程序FSUM即可，程序的其他部分无需修改。程序流程图如下：



图3 实验习题1程序流程图

1. 实验习题2方案描述

计算 1+2+3+……N。要求 N 由键盘输入；给出必要的提示信息（由宏调用实现）；累加功能由子程序调用实现；二进制数（累加和）转换为十进制数并显示由子程序调用实现。

实验习题2与实验习题1类似，只需在习题1 的基础上，取消子程序DTOB中的循环输入，只输入一次（即为N的值），同时对于具有累加功能的子程序稍作修改得到新的子程序NADD。针对子程序NADD，需要将N作为循环次数存入CX，用作判断条件，同时设计指令MOV BX,01H，从1开始累加，每次累加结束后利用LOOP指令使CX-1直至CX=0，最终的累加结果保存在AX中。

程序流程图如下：



图4 实验习题2程序流程图

（4）程序清单

① 实验内容完整程序代码

.model small

.data

COUNT EQU 10

NUM DB 10 DUP(0)

;NUM DW 10 DUP(0);可输入256-65535之间的数

IBUF DB 6,0,6 DUP(0)

OBUF DB 6 DUP(0)

INFOR1 DB "Please input 10 numbers:",0AH,0DH,'$'

INFOR2 DB "The max found in the 10 numbers is $"

INFOR3 DB 0AH,0DH,'$'

.stack

.code

DSTRING MACRO STRING

PUSH DX

PUSH AX

MOV DX, OFFSET STRING

MOV AH, 09H

INT 21H

POP AX

POP DX

ENDM

start:

MOV AX,@data

MOV DS,AX

DSTRING INFOR1

MOV BX,OFFSET NUM

MOV CX,COUNT

LOOP1:

CALL DTOB ;输入的数据在 AL 中

DSTRING INFOR3 ;回车换行

MOV [BX],AL

;MOV [BX],AX ;字节更改为字，需修改相应寄存器

INC BX

;ADD BX,2 ;一次移动两字节

LOOP LOOP1

MOV AX,OFFSET NUM

PUSH AX ;通过堆栈向子程序传递参数（数据个数及数据在数据段内的偏移地址）

MOV CX,COUNT

PUSH CX

CALL FMAX ;子程序，输入参数为数值地址和元素个数

ADD SP,4;堆栈指针返回到原来的位置

DSTRING INFOR2

CALL BTOAD ;数据以十进制形式显示的子程序

MOV AH,4CH

INT 21H

DTOB PROC

PUSH CX

PUSH DX

PUSH SI

MOV DX,OFFSET IBUF

MOV AH,0AH

INT 21H

MOV CL,IBUF+1;十进制位数送CX

MOV CH,0

MOV SI,OFFSET IBUF+2;指向输入字符的最高位

MOV AX,0

AGAIN1:

MOV DX,10

MUL DX

AND BYTE PTR [SI],0FH;将ASCII码转换成2进制数

ADD AL,[SI]

ADC AH,0;有进位则将向高位+1

INC SI;指向输入字符的次高位

LOOP AGAIN1

POP SI

POP DX

POP CX

RET

DTOB ENDP

FMAX PROC

PUSH BP

MOV BP,SP;将SP的值赋给BP，避免直接修改SP

MOV AL,[BP+6];指向主程序中存入的AX值

;MOV AX,[BP+6];字节更改为字，需修改相应寄存器

MOV CX,[BP+4];指向主程序中存入的CX值

AGAIN2:

INC SI

;ADD SI,2 ;一次移动两字节

CMP AL,[SI]

;CMP AX,[SI] ;字节更改为字，需修改相应寄存器

JAE AGAIN3

XCHG AL,[SI]

;XCHG AX,[SI] ;字节更改为字，需修改相应寄存器

AGAIN3:

DEC CX

JNZ AGAIN2

MOV AH,0 ; 若更改为字变量，则此处省略

POP BP

RET

FMAX ENDP

BTOAD PROC

PUSH BX

PUSH CX

PUSH DX

MOV BX,OFFSET OBUF+5 ;指向缓冲区末位

MOV BYTE PTR[BX],'$'

MOV CX,10

AGAIN4:

MOV DX,0;高16位置0

DIV CX ;(DX):(AX)/10

ADD DL,30H;将十进制数转换成ASCII码

DEC BX

MOV [BX],DL

OR AX,AX

JNZ AGAIN4;若商不为0则继续循环

MOV DX,BX

MOV AH,09H

INT 21H

POP BX

POP CX

POP DX

BTOAD ENDP

END start

* 1. 实验习题1完整程序代码

.model small

.data

COUNT EQU 10

NUM DB 10 DUP(0)

IBUF DB 6,0,6 DUP(0)

OBUF DB 6 DUP(0)

INFOR1 DB "Please input 10 numbers:",0AH,0DH,'$'

INFOR2 DB "The SUM of the 10 numbers is $"

INFOR3 DB 0AH,0DH,'$'

.stack

.code

DSTRING MACRO STRING

PUSH DX

PUSH AX

MOV DX, OFFSET STRING

MOV AH, 09H

INT 21H

POP AX

POP DX

ENDM

start:

MOV AX,@data

MOV DS,AX

DSTRING INFOR1

MOV BX,OFFSET NUM

MOV CX,COUNT

LOOP1:

CALL DTOB ;输入的数据在 AL 中

DSTRING INFOR3 ;回车换行

MOV [BX],AL

INC BX

LOOP LOOP1

MOV AX,OFFSET NUM

PUSH AX ;通过堆栈向子程序传递参数（数据个数及数据在数据段内的偏移地址）

MOV CX,COUNT

PUSH CX

CALL FSUM ;子程序，输入参数为数值地址和元素个数

ADD SP,4

DSTRING INFOR2

CALL BTOAD ;数据以十进制形式显示的子程序

MOV AH,4CH

INT 21H

DTOB PROC

PUSH CX

PUSH DX

PUSH SI

MOV DX,OFFSET IBUF

MOV AH,0AH

INT 21H

MOV CL,IBUF+1;十进制位数送CX

MOV CH,0

MOV SI,OFFSET IBUF+2;指向输入字符的最高位

MOV AX,0

AGAIN1:

MOV DX,10

MUL DX

AND BYTE PTR [SI],0FH;将ASCII码转换成2进制数

ADD AL,[SI]

ADC AH,0;有进位则将向高位+1

INC SI;指向输入字符的次高位

LOOP AGAIN1

POP SI

POP DX

POP CX

RET

DTOB ENDP

FSUM PROC

PUSH BP

MOV BP,SP

MOV AX,0

MOV SI,BP+6

MOV CX,[BP+4]

AGAIN2:

MOV BL,[SI];直接ADD AX,[SI]，会使AX一次存入两字节的数

MOV BH,0

ADD AX,BX

INC SI

DEC CX

JNZ AGAIN2

POP BP

RET

FSUM ENDP

BTOAD PROC

PUSH BX

PUSH CX

PUSH DX

MOV BX,OFFSET OBUF+5 ;指向缓冲区末位

MOV BYTE PTR[BX],'$'

MOV CX,10

AGAIN4:

MOV DX,0;高16位置0

DIV CX ;(DX):(AX)/10

ADD DL,30H;将十进制数转换成ASCII码

DEC BX

MOV [BX],DL

OR AX,AX

JNZ AGAIN4;若商不为0则继续循环

MOV DX,BX

MOV AH,09H

INT 21H

POP BX

POP CX

POP DX

BTOAD ENDP

END start

③ 实验习题2完整程序代码

.model small

.data

IBUF DB 6,0,6 DUP(0)

OBUF DB 6 DUP(0)

INFOR1 DB "Please input N:",0AH,0DH,'$'

INFOR2 DB "The SUM of 1+2+3+...+N is: $"

INFOR3 DB 0AH,0DH,'$'

.stack

.code

DSTRING MACRO STRING

PUSH DX

PUSH AX

MOV DX, OFFSET STRING

MOV AH, 09H

INT 21H

POP AX

POP DX

ENDM

start:

MOV AX,@data

MOV DS,AX

DSTRING INFOR1

CALL DTOB ;输入的数据在 AL 中

DSTRING INFOR3 ;回车换行

CALL NADD ;子程序，输入参数为数值地址和元素个数

DSTRING INFOR2

CALL BTOAD ;数据以十进制形式显示的子程序

MOV AH,4CH

INT 21H

DTOB PROC

PUSH CX

PUSH DX

PUSH SI

MOV DX,OFFSET IBUF

MOV AH,0AH

INT 21H

MOV CL,IBUF+1;十进制位数送CX

MOV CH,0

MOV SI,OFFSET IBUF+2;指向输入字符的最高位

MOV AX,0

AGAIN1:

MOV DX,10

MUL DX

AND BYTE PTR [SI],0FH;将ASCII码转换成2进制数

ADD AL,[SI]

ADC AH,0;有进位则将向高位+1

INC SI;指向输入字符的次高位

LOOP AGAIN1

POP SI

POP DX

POP CX

RET

DTOB ENDP

NADD PROC

PUSH CX

PUSH BX

MOV CX,AX

MOV AX,0

MOV BX,01H

LOOP1:

ADD AX,BX

INC BX

LOOP LOOP1

POP BX

POP CX

RET

NADD ENDP

BTOAD PROC

PUSH BX

PUSH CX

PUSH DX

MOV BX,OFFSET OBUF+5 ;指向缓冲区末位

MOV BYTE PTR[BX],'$'

MOV CX,10

AGAIN4:

MOV DX,0;高16位置0

DIV CX ;(DX):(AX)/10

ADD DL,30H;将十进制数转换成ASCII码

DEC BX

MOV [BX],DL

OR AX,AX

JNZ AGAIN4;若商不为0则继续循环

MOV DX,BX

MOV AH,09H

INT 21H

POP BX

POP CX

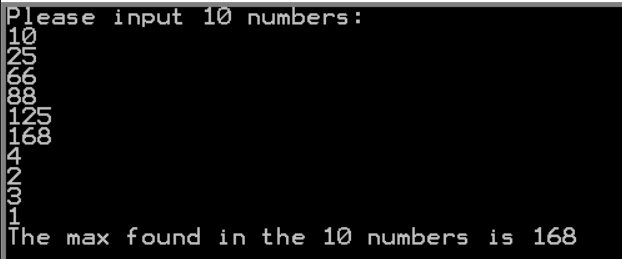
POP DX

BTOAD ENDP

END start

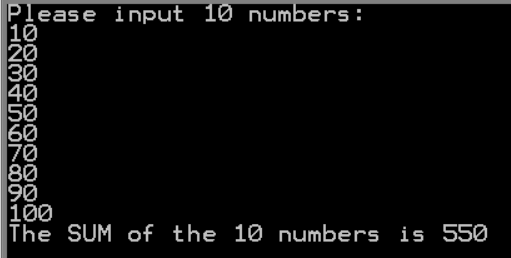
1. 实验结果
2. 实验内容结果

任意输入十个数，输出其中的最大值，得到的结果如下：



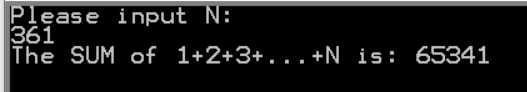
1. 实验习题1结果

令N=10，并任意输入十个数，求出累加和，得到的结果如下：



1. 实验习题2结果

输入一个N的值，并计算“1+2+3+……+N”的累加和，得到的结果如下：

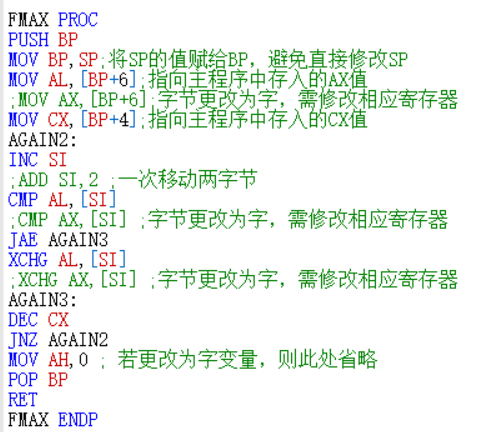


4. 实验总结

（1）实验中遇到的问题及解决措施

① **刚开始实验时没有利用堆栈向子程序传递参数**：借助BP实现对主程序中存入堆栈的值的访问，最后使SP指向栈底，实现出栈。

1. 实验内容中需要修改程序，使其允许输入255-65535之间的数：首先需要使存储输入数据的字节变量更改为字变量；在子程序DTOB中，输入的数据应当在AX中，同时以BX存储地址值后每次需要移动两字节；在子程序FMAX中，也需要对相应的寄存器进行更改，具体更改部分如下图所示（注释部分）：



③ **实验习题1中，直接使用指令ADD AX,[SI]所得结果不正确**：ADD AX,[SI]会使得AX一次读入两字节的数（即连续输入的两个数），从而无法得到正确的累加结果。因此需要借助BX寄存器，使用指令MOV BL,[SI]并使BX高8位置零，这样在累加时只需使用指令ADD AX,BX，即可在AX中求得正确的累加和。

（2）实验中学到的知识

此次实验使得我对于宏指令、宏定义、宏调用和宏展开的概念更加熟悉，也能够在程序中利用宏指令编写程序。同时，通过对多个不同子程序的编写，我掌握了子程序的定义和调用的基本方法，并且能够利用堆栈实现不同子程序之间的参数传递。

本次实验内容中，子程序从堆栈中读取参数并返回后，RET指令使得IP出栈，但AX和CX所存数据仍在栈中没有出栈，对此可以在主程序中结束调用子程序后，使用指令ADD SP,4使SP指向栈底，其效果相当于令AX和CX出栈。

（3）依然存在的问题

在输入多个数时，每输入一个数，需要等待其成功读入后才可以输入下一个，否则会使程序陷入死循环，若不输入直接回车同样也会导致死循环。