

学号： WA2224013      专业： 机器人工程      姓名： 郭义月

实验日期： 2024. 11. 30    指导教师： 鲍华      实验成绩：

课程目标 1 (权重____)	课程目标 2 (权重____)	课程目标 3 (权重____)	课程目标 4 (权重____)	课程目标 5 (权重____)	课程目标 6 (权重____)	综合成绩 (目标数可增删)

## 安徽大学人工智能学院本科实验报告

【课程名称】 微型计算机原理及接口技术

【课程目标】

掌握汇编语言中的数据拆分与合并、成绩分布统计及屏幕输出操作，提升对内存操作、条件判断和循环控制的理解与应用能力。

【实验名称】 实验一

【实验目的】

- 1. 掌握汇编语言设计和调试方法。
- 2. 掌握汇编语言多分支结构设计方法和dos功能调用显示功能

【实验原理及方法】

本实验通过汇编语言实现数据拆分、合并和统计功能。通过访问内存地址，将4000H中的16位数据拆开，分别存到4001H和4002H，常用于数据处理或显示缓冲区的操作。同时，提取4000H和4001H中的低位数据，通过按位操作将它们合并为一个字节存储到4002H，用于拼装显示缓冲区数据。实验还要求统计十个学生的成绩分布，通过条件判断和循环控制，统计各个分数段的人数，并将结果输出到屏幕。通过这几个任务，我们熟悉了内存操作、条件跳转、循环控制及屏幕输出的基本方法，掌握了汇编语言中的数据处理技巧。

【实验内容及过程】

## 任务1：拆字

把 4000H 单元的内容拆开， 高位送 4001H 低位， 低位送4002H 低位， 4001H、4002H 的高位清零， 一般本程序用于把数据送显示缓冲区时用。

汇编程序：

```
DATAS SEGMENT
    ;此处输入数据段代码
DATAS ENDS

STACKS SEGMENT
    ;此处输入堆栈段代码
STACKS ENDS

CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
    MOV AX,DATAS
    MOV DS,AX
    ;此处输入代码段代码
    MOV DI, 12H
    MOV SI, 4000H
    MOV BYTE PTR [SI], DI
    MOV AL, BYTE PTR [SI]
    MOV BL, AL
    ; 清除高位并将结果存入 4001H 和 4002H
    AND BL, 0F0H
    MOV CL, 4
    SHR BL, CL
    MOV [SI+1], BL
    MOV BL, AL
    AND BL, 0FH
    MOV [SI+2], BL
    ; 程序结束后，4001H 存储高位字节 01H，4002H 存储低位字节 02H，任务完成
    MOV AH,4CH
    INT 21H
CODES ENDS
    END START
```

## 任务2：拼字

把 4000H、 4001H 低位分别送入 4002H 高低位， 一般本程序用于把显示缓冲区的数据取出拼装成一个字节。

汇编程序：

```
MOV BX, 4000H
MOV DL, 34H
MOV BYTE PTR [BX], DL
```

```

MOV BYTE PTR [BX+1], 78H
MOV AL, [BX]
AND AL, 0FH
MOV CL, 4
SHL AL, CL
MOV AH, [BX+1]
AND AH, 0FH
ADD AH, AL
MOV [BX+2], AH

```

**任务3：**数据区中一变量score存有十个同学的考试成56, 69, 84, 82, 73, 88, 99, 63, 100, 80。要求统计十个成绩的分布情况。即100分有几个，90到100分（包括90）有几个。80到90（包括80）有几个，70到80（包括70）有几个，60到70（包括60）有几个，60分以下有几个。将统计结果保存在数据段中，并将结果在屏幕上显示输出。

```

DATAS SEGMENT
    SCORE DB 56, 69, 84, 82, 73, 88, 99, 63, 100, 80 ; 10个学生的成绩

    ; 统计结果变量：每个变量表示一个成绩段的人数
    COUNT100 DB 0
    COUNT90T0100 DB 0
    COUNT80T090 DB 0
    COUNT70T080 DB 0
    COUNT60T070 DB 0
    COUNTBELOW60 DB 0

    MSGRESULT DB 'Results:', 0Dh, 0Ah, '$'
DATAS ENDS

STACKS SEGMENT
    ; 堆栈段代码可以根据需要设置
STACKS ENDS

CODES SEGMENT
    ASSUME CS:CODES, DS:DATAS, SS:STACKS
START:
    MOV AX, DATAS
    MOV DS, AX
    ; 遍历成绩数据
    LEA SI, SCORE
    MOV CX, 10
CHECK_SCORES:
    MOV AL, [SI]
    INC SI

    CMP AL, 100
    JE SCORE100

```

```

    CMP AL, 90
    JAE SCORE90T0100
    CMP AL, 80
    JAE SCORE80T090
    CMP AL, 70
    JAE SCORE70T080
    CMP AL, 60
    JAE SCORE60T070

    JMP SCOREBELOW60
SCORE100:
    INC BYTE PTR [COUNT100]
    JMP NEXT_SCORE
SCORE90T0100:
    INC BYTE PTR [COUNT90T0100]
    JMP NEXT_SCORE
SCORE80T090:
    INC BYTE PTR [COUNT80T090]
    JMP NEXT_SCORE
SCORE70T080:
    INC BYTE PTR [COUNT70T080]
    JMP NEXT_SCORE
SCORE60T070:
    INC BYTE PTR [COUNT60T070]
    JMP NEXT_SCORE
SCOREBELOW60:
    INC BYTE PTR [COUNTBELOW60]
NEXT_SCORE:
    LOOP CHECK_SCORES
    ; 输出统计结果
    LEA DX, MSGRESULT
    MOV AH, 09H
    INT 21H
    LEA SI, COUNT100
    CALL PRINTCOUNT
    LEA SI, COUNT90T0100
    CALL PRINTCOUNT
    LEA SI, COUNT80T090
    CALL PRINTCOUNT
    LEA SI, COUNT70T080
    CALL PRINTCOUNT
    LEA SI, COUNT60T070
    CALL PRINTCOUNT
    LEA SI, COUNTBELOW60
    CALL PRINTCOUNT
    MOV AH, 4CH
    INT 21H

PRINTCOUNT:
    MOV AL, [SI]

```

```

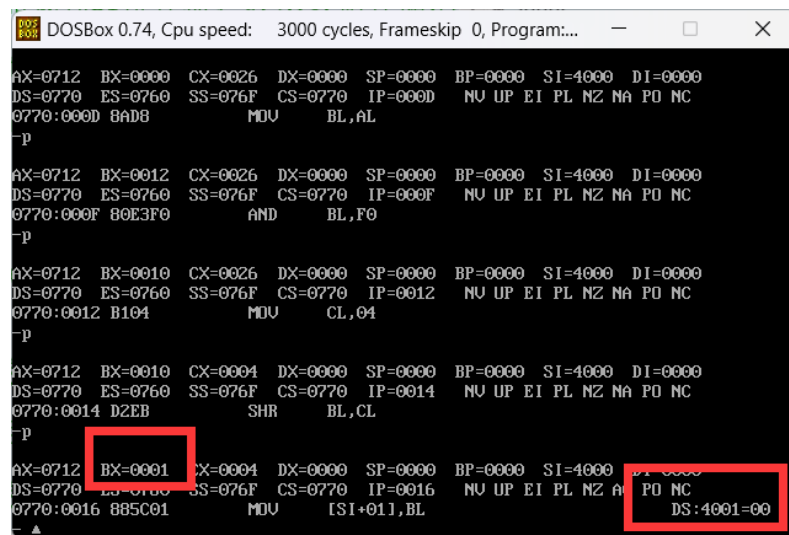
ADD AL, '0'
MOV DL, AL
MOV AH, 02H
INT 21H
RET
MOV AH, 4CH
INT 21H
CODES ENDS
END START

```

## 【实验结果】

### 任务1：拆字

将高位（01H）存在4001中

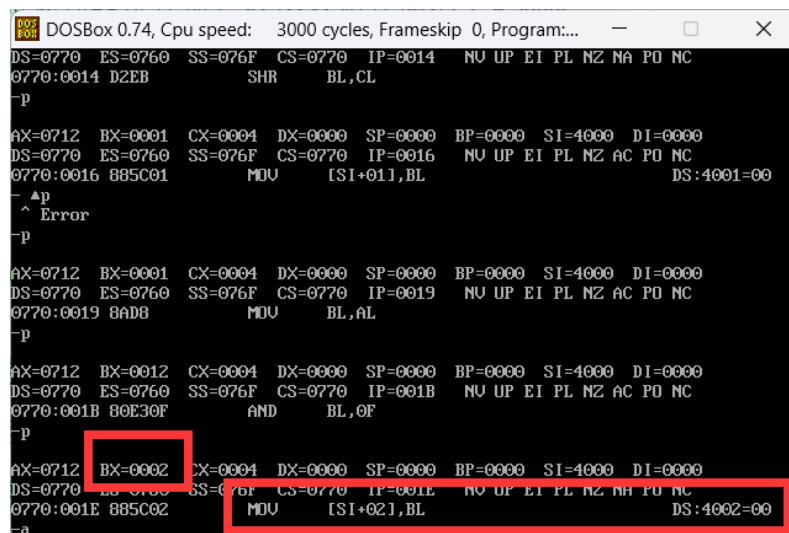


```

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program:...
AX=0712 BX=0000 CX=0026 DX=0000 SP=0000 BP=0000 SI=4000 DI=0000
DS=0770 ES=0760 SS=076F CS=0770 IP=000D NU UP EI PL NZ NA PO NC
0770:000D 8AD8 MOV BL,AL
-p
AX=0712 BX=0012 CX=0026 DX=0000 SP=0000 BP=0000 SI=4000 DI=0000
DS=0770 ES=0760 SS=076F CS=0770 IP=000F NU UP EI PL NZ NA PO NC
0770:000F 80E3F0 AND BL,F0
-p
AX=0712 BX=0010 CX=0026 DX=0000 SP=0000 BP=0000 SI=4000 DI=0000
DS=0770 ES=0760 SS=076F CS=0770 IP=0012 NU UP EI PL NZ NA PO NC
0770:0012 B104 MOV CL,04
-p
AX=0712 BX=0010 CX=0004 DX=0000 SP=0000 BP=0000 SI=4000 DI=0000
DS=0770 ES=0760 SS=076F CS=0770 IP=0014 NU UP EI PL NZ NA PO NC
0770:0014 D2EB SHR BL,CL
-p
AX=0712 BX=0001 CX=0004 DX=0000 SP=0000 BP=0000 SI=4000 DI=0000
DS=0770 ES=0760 SS=076F CS=0770 IP=0016 NU UP EI PL NZ NA PO NC
0770:0016 8B5C01 MOV SI,011,BL
DS:4001=00

```

将低位（02H）存在4002中



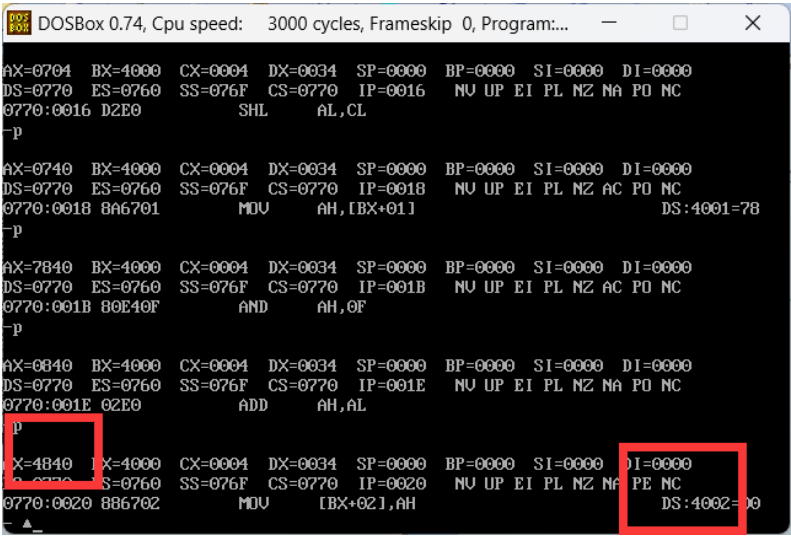
```

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program:...
DS=0770 ES=0760 SS=076F CS=0770 IP=0014 NU UP EI PL NZ NA PO NC
0770:0014 D2EB SHR BL,CL
-p
AX=0712 BX=0001 CX=0004 DX=0000 SP=0000 BP=0000 SI=4000 DI=0000
DS=0770 ES=0760 SS=076F CS=0770 IP=0016 NU UP EI PL NZ AC PO NC
0770:0016 8B5C01 MOV SI,011,BL
DS:4001=00
Error
-p
AX=0712 BX=0001 CX=0004 DX=0000 SP=0000 BP=0000 SI=4000 DI=0000
DS=0770 ES=0760 SS=076F CS=0770 IP=0019 NU UP EI PL NZ AC PO NC
0770:0019 8AD8 MOV BL,AL
-p
AX=0712 BX=0012 CX=0004 DX=0000 SP=0000 BP=0000 SI=4000 DI=0000
DS=0770 ES=0760 SS=076F CS=0770 IP=001B NU UP EI PL NZ AC PO NC
0770:001B 80E30F AND BL,0F
-p
AX=0712 BX=0002 CX=0004 DX=0000 SP=0000 BP=0000 SI=4000 DI=0000
DS=0770 ES=0760 SS=076F CS=0770 IP=001E NU UP EI PL NZ NA PO NC
0770:001E 8B5C02 MOV SI,021,BL
DS:4002=00

```

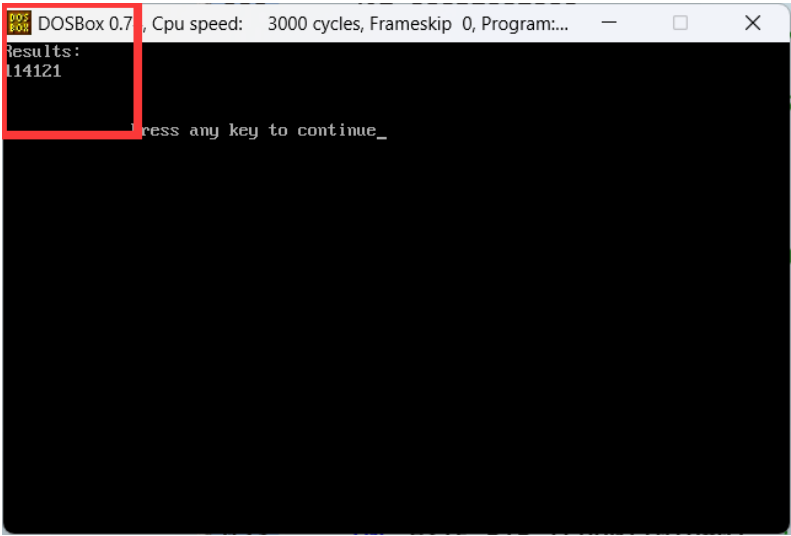
### 任务2：拼字

当4000H与4001H中存储的数分别为34H，78H时，拆字的结果应为48H，调试结果可知，AH=48H，且DS=4002H，说明已经顺利把48H存入4002H中



```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program:...\nAX=0704 BX=4000 CX=0004 DX=0034 SP=0000 BP=0000 SI=0000 DI=0000\nDS=0770 ES=0760 SS=076F CS=0770 IP=0016 NU UP EI PL NZ NA PO NC\n0770:0016 D2E0 SHL AL,CL\n-p\nAX=0740 BX=4000 CX=0004 DX=0034 SP=0000 BP=0000 SI=0000 DI=0000\nDS=0770 ES=0760 SS=076F CS=0770 IP=0018 NU UP EI PL NZ AC PO NC\n0770:0018 8A6701 MOV AH,[BX*01] DS:4001=78\n-p\nAX=7B40 BX=4000 CX=0004 DX=0034 SP=0000 BP=0000 SI=0000 DI=0000\nDS=0770 ES=0760 SS=076F CS=0770 IP=001B NU UP EI PL NZ AC PO NC\n0770:001B 80E40F AND AH,0F\n-p\nAX=0B40 BX=4000 CX=0004 DX=0034 SP=0000 BP=0000 SI=0000 DI=0000\nDS=0770 ES=0760 SS=076F CS=0770 IP=001E NU UP EI PL NZ NA PO NC\n0770:001E 02E0 ADD AH,AL\n-p\nAX=4B40 BX=4000 CX=0004 DX=0034 SP=0000 BP=0000 SI=0000 DI=0000\nDS=0770 ES=0760 SS=076F CS=0770 IP=0020 NU UP EI PL NZ NA PO NC\n0770:0020 8B6702 MOV EBX,[BX*02],AH DS:4002=00
```

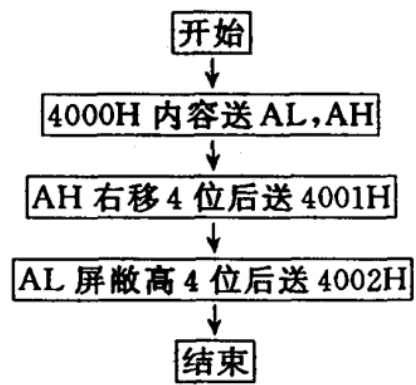
任务3：统计学生成绩



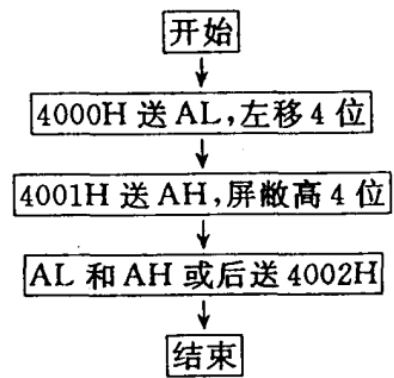
```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program:...\nResults:\n114121\nPress any key to continue_
```

【数据分析及处理】

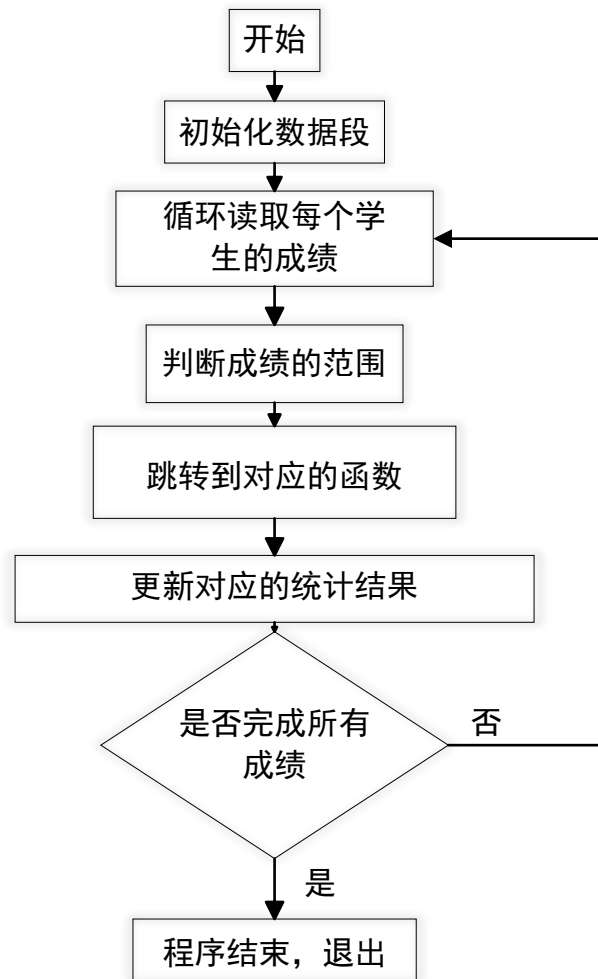
任务1流程图：



任务2流程图:



任务3流程图:



## 【总结或讨论】

当需要进行与，或相关操作，需要涉及立即数时，必须要在前面补零，否则就会报错：无法识别。

正确指令：

```
AND BX,0FF00H
```

错误指令示例：



```
all.asm
TEST.ASM
软件应用问题解答
简单的入门程序实例
数据传送指令(免费)
算术运算指令
逻辑运算指令
串操作指令
程序转移指令
汇编伪指令
DOS汇编源程序
WINDOWS汇编实例源程序
汇编语言错误信息表
DOS 功能调用表(免费)
BIOS 功能调用表(免费)
DEBUG 命令(免费)
ASCII码字符表(免费)

14 ;此处输入汇编代码
15 MOV DI, 1234H ; 给AX赋值1234H
16 MOV SI, 4000H
17 MOV WORD PTR [SI], DI ; 将1234H存入4000H
18 MOV AX, word ptr [SI]
19 ;MOV AX, [4000H] ; 从4000H加载内容到AX寄存器
20 MOV BX, AX
21 AND BX, FF00H
22 MOV CL, 8
23 SHR BX, CL
24 MOV [SI+1], BX
25
26 HLT ; 程序结束
27
28 MOV AH, 4CH
29 INT 21H
30 :ODES ENDS
31 END START
32

兼容WinXP模式】编译源程序 D:\Desktop\最近\汇编语言程序实验\第一次\all.asm
:\Desktop\最近\汇编语言程序实验\第一次\all.asm(21): error A006: undefined symbol : FF00H
```

通过本次实验，我掌握汇编语言中的数据拆分与合并、成绩分布统计及屏幕输出操作，提升对内存操作、条件判断和循环控制的理解与应用能力。