

东南大学

《微机系统与接口实验》

实验报告

实验二 基本算术和逻辑运算

姓 名：薛宇飞

学 号：04020235

同 组：

学 号：

专 业：信息工程

实 验 室：金智楼硬件实验室

实验时间：2022 年 4 月 14 日

报告时间：2022 年 4 月 14 日

评定成绩：

评阅教师：裴文江

目录

1 实验目的与内容	3
2 实验任务	3
1 运行并检查标志位	3
2 求和与作积	5
3 编写功能程序段 1	6
4 编写功能程序段 2	7
5 清除程序段	7
6 尝试 BCD 码调整指令	8
1 BCD 码任务 1	8
2 BCD 码任务 2	8
3 实验总结	10
参考文献	10

一. 实验目的与内容

1. 熟悉算术和逻辑运算指令的功能；
2. 结合实验教材^[1-2]，进一步了解标志寄存器各标志位的意义和指令执行对它的影响。

二. 实验任务

实验全部资料及完整代码详见薛宇飞的 [GitHub 主页](#)^[3]。

(一) 运行并检查标志位

采用单步执行方式执行下列各程序段，检查各标志位的情况。

code1

```
1  MOV AX,0A0A0H
2  ADD AX,0FFFFH
3  MOV CX,0FF00H
4  ADD AX,CX
5  SUB AX,AX
6  INC AX
7  OR CX,00FFH
8  AND CX,0F0FH
9  MOV [0010],CX
```

分析 1: 指令对标志位的影响

第 2 行执行后，AX=0A09FH，结果溢出，CF=1，SF=1，PF=1；
第 4 行执行后，属于正常加法，AX=9F9FH，符号为未改变；
第 5 行执行后，AX=0000H，结果为正零，无借位，ZF=1，CF=0，SF=0；
第 6 行执行后，AX=0001H，PF=0，CF=0，ZF=0；
第 7 行执行后，CX 低 8 位置 1，PF=1，SF=1；
第 8 步执行后，CX 的 4-7 位和 12-15 位清零，SF=0，PF=1；
第 9 步执行后，DS:[0010]=0F，DS:[0011]=0F；
其余指令为简单的 MOV 操作。

code2

```
1  MOV BL,25H
2  MOV [0010],04H
3  MOV AL,[0010]
4  MUL BL
```

分析 2: 指令对标志位的影响

第 2 行指令输入后, 指令默认变成了 `MOV BYTE PTR [0100], 04H`, 执行后, `DS:[0100]=04`;
第 4 行, 属于无符号数的字节乘法运算, 源操作数为 `BL`, 目标操作数为 `AL`, 结果存放于 `AX=0094H`, 符号位 `OF=0`, `CF=0`。

code3

```
1  MOV BL, 04H
2  MOV WORD PTR [0010], 0080H
3  MOV AX, [0010]
4  DIV BL
```

分析 3: 指令对标志位的影响

第 2 行指令执行后, `DS:[0011]=00`, `DS:[0010]=80`, 这是因为指定了 `MOV` 指令移入一个字, 所以高位的 `00H` 被存放到了 `DS:[0011]` 中;
第 4 行, 属于无符号数的字节除法, 默认目标操作数为 `AX`, 商放在 `AL`, 余数放在 `AH`, 所以 `AX=0020H`, 对所有标志位均无影响。

code4

```
1  MOV AX, 00
2  DEC AX
3  ADC AX, 3FFFH
4  ADD AX, AX
5  NOT AX
6  SUB AX, 3
7  OR AX, 0FBFDH
8  AND AX, 0AFCFH
9  SHL AX, 1
10 RCL AX, 1
```

分析 4: 指令对标志位的影响

第 2 行指令执行后, `AX=FFFFH`, 最高位有借位, 看作有符号数为负数, 低 8 位的 1 个数为偶数, 所以 `AF=1`, `SF=1`, `PF=1`;
第 3 行指令执行后, `AX=3FFE`, 结果最高位有进位, `CF=1`, 正常加法完成后再加 1;
第 4 行指令执行后, `AX=7FFC`, 低半字节向高半字节有进位, 所以 `AF=1`;
第 5 行执行之前 `AX=0111 1111 1111 1100B`, 执行后所有位取反得 `AX=1000 0000 0000 0011B=8003H`; 第 6 行执行后, `AX=8000H`, 标志位 `SF=1`, `PF=1`;
第 7 行, `FBFDH=1111 1011 1111 1101B`, 指令表示将第 0、2、3、4、5、6、7、8、9、11、12、

13、14、15 位置 1，得到 $AX=FBFDH$ ；

第 8 行， $AFCFH=1010\ 1111\ 1100\ 1111B$ ，指令表示将第 4、5、12、14 位清零，得到 $AX=ABCDH$ ；
第 9 行指令执行后，将 AX 向左移位得到 $AX=579AH$ ，最高位为 1 挪进 CF 中，并且 $SF=0$ ， $ZF=0$ ， $PF=0$ 。

注意 1: 移位位数

位数若超过 1，需存放于 CL 中使用。

第 10 行指令执行后， $AX=AF35H$ ，并且原始最高位 0 置入 CF 中。

思考 1: 移位思想

右移位相当于将原数字（有符号数或无符号数）乘除 2。

注意 2: RCL/RCR 移位

左移为例，依次向左移位，将最高位放入 CF 中，并把最低位用 CF 原来的值补齐

(二) 求和与作积

将寄存器 BX 作地址指针，自 BX 所指的内存单元（0010H）开始连续存放着三个无符号数（10H、04H、30H）。试编写程序分别求它们的和与积，并将结果存放在这三个数之后的单元中。

code

```
1  MOV BYTE PTR [BX],10H ;初始化操作
2  MOV BYTE PTR [BX+1],04H
3  MOV BYTE PTR [BX+2],30H
4  MOV AX,0000H ;清空中转单元格
5  ADD AL,[BX]
6  ADD AL,[BX+1]
7  ADD AL,[BX+2]
8  MOV WORD PTR [BX+3],AX ;求和并存放结果
9  MOV AX,0000H ;清空中转单元格
10 MOV AL,[BX]
11 MOV CL,[BX+1]
12 MUL CL
13 MOV CL,[BX+2]
14 MUL CL
15 MOV WORD PTR [BX+5],AX ;求积并存放结果
```

实验结果见图 1

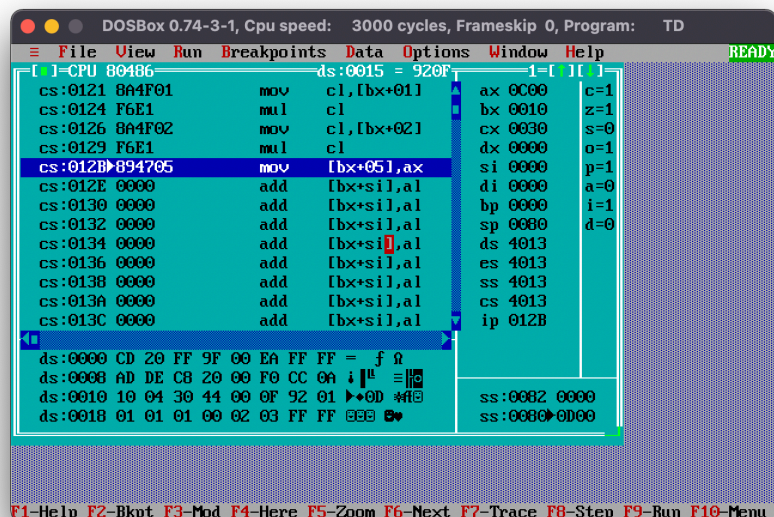


图 1: 实验二结果

注意 3: 实验中遇到的问题

1. MUL/IMUL 的目标操作数只能是 AL，且源操作数存储器寻址有 BX，BP，SI，DI 寻址；
2. 将数据移入存储器需要指明是 WORD PTR 或者 BYTE PTR，如果是双操作数，一个是寄存器，一个是存储器，则数据类型由寄存器确定；
3. 字节运算的结果可能是字，因此将结果移入存储器前需要事先规划好位置。

(三) 编写功能程序段 1

1. 传送 15H 到 AL 寄存器；
2. 将 AL 的内容乘以 2；
3. 传送 15H 到 BL 寄存器；
4. AL 的内容乘以 BL 的内容。
5. 求出最后结果 AX。

code

```

1  MOV AL,15H
2  SHL AL,1
3  MOV BL,15H
4  MUL BL ;最终得到结果: AL=0372H

```

(四) 编写功能程序段 2

1. 从地址 DS:0000H 单元中, 传送一个数据 58H 到 AL 寄存器;
2. 把 AL 寄存器的内容右移两位;
3. 再把 AL 寄存器的内容与字节单元 DS:0001H 中的数据 12H 相乘;
4. 将乘积存入字单元 DS:0002H 中。

code

```

1      ;首先修改DS:0000H的值为58H, DS:0001H为12H
2      MOV AL,[0000]
3      MOV CL,02H
4      SHR AL,CL
5      MOV BL,[0001]
6      MUL BL
7      MOV WORD PTR [0002],AX

```

实验结果见图 2

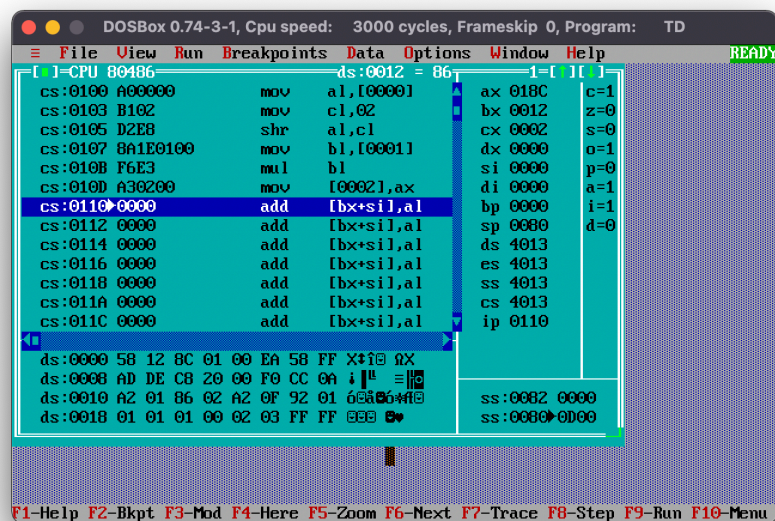


图 2: 实验四结果

(五) 清除程序段

假设下面的程序段用来清除数据段中相应字存储单元的内容 (即零送到这些存储单元中去), 其偏移地址从 0010H 到 001FH。

code

```
1      MOV SI,0010H
2  NEXT: MOV WORD PTR [SI],00
3      ADD SI,2
4      CMP SI,-----
5      JNE NEXT
```

空白处应补充完整的指令为 `CMP SI,001FH`。

假设要清除偏移地址从 `001FH` 到 `0010H` 字存储单元中的内容 (即由高地址到低地址清零), 试编写程序段。

code

```
1      MOV SI,001FH
2  NEXT: MOV WORD PTR [SI],00
3      SUB SI,2
4      CMP SI,0010H
5      JNE NEXT
```

需要调整上一个实验中的第 3 行为 `SUB SI,2`。

注意 4: TD 使用

在 TD 中, `NEXT` 指令无法输入, 需要换成相对应的偏移地址。

(六) 尝试 BCD 码调整指令

1. BCD 码任务 1

假设数据段: `[0000H]=18H`, `[0001H]=34H`, `[0010H]=98H`, `[0011H]=27H`。执行表 1。
最后得到两个压缩 BCD 码 3418 与 2798 的和为 6216。

分析 5: DAA 指令

DAA 操作逻辑:

AL 低 4 位大于 9 或 `AF=1`, `AL=AL+6` 且 `AF` 置 1;

AL 高 4 位大于 9 或 `CF=1`, `AL=AL+60` 且 `CF` 置 1。

2. BCD 码任务 2

假设数据段: `[0000H]=23H`, `[0001H]=43H`, `[0010H]=61H`, `[0011H]=25H`。执行表 2。
最后得到两个压缩 BCD 码 4323 减去 2561 为 1762。

表 1: 任务表格

指令	分析	AL
MOV AL, [0000H]	简单的 MOV 操作	-
ADD AL, [0010H]	简单的 ADD 操作, 其中 CF=0	B0
DAA	对 AX 进行 +6+60 操作, 注意此时 CF=1	16
MOV [0020H], AL	简单的 MOV 操作	-
MOV AL, [0001H]	简单的 MOV 操作	-
ADC AL, [0011H]	考虑 CF 进位的加法操作	5C
DAA	对 AX 进行 +6 操作	62
MOV [0021H], AL	简单的 MOV 操作	-

表 2: 任务表格

指令	分析	AL
MOV AL, [0000H]	简单的 MOV 操作	-
SUB AL, [0010H]	简单的 SUB 操作, 同时 CF=1	C2
DAS	具体操作为 $AL=AL-60$ 并且 CF=1	62
MOV [0020H], AL	简单的 MOV 操作	-
MOV AL, [0001H]	简单的 MOV 操作	-
SBB AL, [0011H]	考虑借位的减法	1D
DAS	具体操作为 $AL=AL-6$	17
MOV [0021H], AL	简单的 MOV 操作	-

分析 6: DAS 指令

DAS 操作逻辑:

AL 低 4 位大于 9 或 AF=1, AL=AL-6 且 AF 置 1;

AL 高 4 位大于 9 或 CF=1, AL=AL-60 且 CF 置 1。

三. 实验总结

实验总结已随文附在“注意”、“思考”、“分析”中。

参考文献

- [1] 李继灿. 新编 16/32 位微型计算机原理及应用 (第五版) [M]. 5 版. 北京: 清华大学出版社, 2013.
- [2] 微机教学组. 《微计算机实验讲义》[A]. 南京: 东南大学, 2015.
- [3] <https://github.com/xyfool-66/SEU-Microcomputer-Experiments/tree/master>.