

2152118 史君宝 汇编语言 第二次作业

采用合适的程序完成问题的解答

一、给出下列 8 位二进制数相加后 CF、OF、SF、ZF 的值（用 0 或 1 表示）？如果把它们看作是无符号数相加则如何判断溢出（用 1 表示有，0 表示无）？

- | | CF | OF | SF | ZF | 看作无符号数时的溢出判断 |
|-----------------|------|------|------|------|---------------------|
| (1) 0FFH + 01H: | CF=1 | OF=0 | SF=0 | ZF=1 | CF=1 OF=1 SF=1 ZF=0 |
| (2) 80H + 81H: | CF=1 | OF=1 | SF=0 | ZF=0 | CF=1 OF=1 SF=1 ZF=0 |

(1) 解释：在汇编语言中，一般都是将数据看做有符号数的，所有 0FFH+01H 就看做是 $-1 + 1 = 0$

对应的 CF = CY = 1 发生进位

对应的 OF = NV = 0 未发生溢出

对应的 SF = PL = 0 是正数

对应的 ZF = ZR = 1 是 0

```
-t
AX=1CFF BX=0000 CX=0020 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=1CA5 ES=1C95 SS=1CA4 CS=1CA6 IP=0003  NU UP DI PL NZ NA PO NC
1CA6:0008 02060100  ADD     AL,[0001]
Trace Interrupt
-t
AX=1C00 BX=0000 CX=0020 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=1CA5 ES=1C95 SS=1CA4 CS=1CA6 IP=000C  NU UP DI PL ZR AC PE CY
1CA6:000C B44C     MOV     AH,4Ch
Trace Interrupt
```

(2) 解释：在汇编语言中，一般都是将数据看做有符号数的，所有 80H+81H 就看做是 $-128 + -127 = -255$

对应的 CF = CY = 1 发生进位

对应的 OF = NV = 1 发生溢出

对应的 SF = PL = 0 是正数

对应的 ZF = NZ = 0 不是 0

```
Trace Interrupt
-t
AX=1C00 BX=0000 CX=0020 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=1CA5 ES=1C95 SS=1CA4 CS=1CA6 IP=0003  NU UP DI PL NZ NA PO NC
1CA6:0008 02060100  ADD     AL,[0001]
Trace Interrupt
-t
AX=1C01 BX=0000 CX=0020 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=1CA5 ES=1C95 SS=1CA4 CS=1CA6 IP=000C  NU UP DI PL NZ NA PO CY
1CA6:000C B44C     MOV     AH,4Ch
Trace Interrupt
-t
```

二、给出下列 8 位二进制数相减法 CF、OF、SF、ZF 的值（用 0 或 1 表示）？如果把它们看作是无符号数相减则如何判断溢出（用 1 表示有，0 表示无）？

	CF	OF	SF	ZF	看作无符号数时的溢出判断
(1) 09H - 05H:	CF=1	OF=0	SF=0	ZF=0	CF=1 OF=0 SF=0 ZF=0;
(2) 05H - 09H:	CF=1	OF=1	SF=1	ZF=0	CF=1 OF=1 SF=1 ZF=0;

(1) 解释：在汇编语言中，一般都是将数据看做有符号数的，所有 09H+05H 就看做是 $9 - 5 = 4$

对应的 CF = NC = 1 未发生进位

对应的 OF = NV = 0 未发生溢出

对应的 SF = PL = 0 是正数

对应的 ZF = NZ = 0 不是 0

```
-t
AX=1C09 BX=0000 CX=0020 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=1CA5 ES=1C95 SS=1CA4 CS=1CA6 IP=0003  NU UP DI PL NZ NA PO NC
1CA6:0008 2A060100 SUB AL,[0001]
Trace Interrupt
-t
AX=1C04 BX=0000 CX=0020 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=1CA5 ES=1C95 SS=1CA4 CS=1CA6 IP=000C  NU UP DI PL NZ NA PO NC
1CA6:000C B44C MOV AH,4Ch
Trace Interrupt
```

(2) 解释：在汇编语言中，一般都是将数据看做有符号数的，所有 80H+81H 就看做是 $5 - 9 = -4$

对应的 CF = CY = 1 发生进位

对应的 OF = NV = 1 发生溢出

对应的 SF = NG = 1 是负数

对应的 ZF = NZ = 0 不是 0

```
Trace Interrupt
-t
AX=1C05 BX=0000 CX=0020 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=1CA5 ES=1C95 SS=1CA4 CS=1CA6 IP=0003  NU UP DI PL NZ NA PO NC
1CA6:0008 2A060100 SUB AL,[0001]
Trace Interrupt
-t
AX=1CFC BX=0000 CX=0020 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=1CA5 ES=1C95 SS=1CA4 CS=1CA6 IP=000C  NU UP DI NG NZ AC PE CY
1CA6:000C B44C MOV AH,4Ch
Trace Interrupt
```

三、采用程序的方式验证下面的问题的答案

1、已知 BX=028AH, SI=0294H, (DS: 051EH)=5432H。执行语句 LEA BX, [BX+SI]后, BX 中的内容是 (A)

- | | |
|----------|----------|
| A. 051EH | B. 5423H |
| C. 1E05H | D. 3254H |

我们构建下面的程序，并将最终的结果输入到 BX 这个寄存器中，我们在最后一次 -t 的调试中可以看到 BX 中的内容变成了 051EH

```

DATA    SEGMENT           ;数据段定义
DATA    ENDS
CODE    SEGMENT           ;代码段定义
ASSUME  CS:CODE,DS:DATA   ;ASSUME语句
START:  MOV AX,DATA        ;装填DS
        MOV DS,AX

        MOV BX, 028AH
        MOV SI, 0294H
        LEA BX, [BX+SI]

        MOV AH,4CH         ;返回DOS
        INT 21H
CODE    ENDS
END     START              ;结束语言

```

```

-t
AX=1CA5 BX=028A CX=0011 DX=0000 SP=0000 BP=0000 SI=0294 DI=0000
DS=1CA5 ES=1C95 SS=1CA4 CS=1CA5 IP=0000  NU UP DI PL NZ NA PO NC
1CA5:000B 8D1B          LEA    BX,[BX+SI]
Trace Interrupt
-t
AX=1CA5 BX=051E CX=0011 DX=0000 SP=0000 BP=0000 SI=0294 DI=0000
DS=1CA5 ES=1C95 SS=1CA4 CS=1CA5 IP=0000  NU UP DI PL NZ NA PO NC
1CA5:000D B44C          MOV    AH,4Ch
Trace Interrupt

```

2、下面是数据段的定义，用程序的方式解答下面的问题

.data

No1 dw 12

No2 db 20 dup(30)

No3 dd 34

No4 equ \$-No1

No4 的值是 C

A.56

B.78

C.1AH

D.27

我们构建下面的程序，并将最终的结果输入到 BX 这个寄存器中，我们在最后一次-t 的调试中可以看到 BX 中的内容变成了 001AH

```

6 ;当然,更好的是创建为一个工程,更方便及易于管理,使用万法:按Ctrl多远->创建
7
8 DATA    SEGMENT           ;数据段定义
9 No1 dw 12
10 No2 db 20 dup(30)
11 No3 dd 34
12 No4 equ $-No1
13 DATA    ENDS
14 CODE    SEGMENT           ;代码段定义
15 ASSUME  CS:CODE,DS:DATA   ;ASSUME语句
16 START:  MOV AX,DATA        ;装填DS
17          MOV DS,AX
18          MOV BX,No4
19
20          MOV AH,4CH         ;返回DOS
21          INT 21H
22 CODE    ENDS
23 END     START              ;结束语言

```

```

-t
AX=1CA5 BX=0000 CX=002C DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=1C95 ES=1C95 SS=1CA4 CS=1CA7 IP=0003  NU UP DI PL NZ NA PO NC
1CA7:0003 8ED8      MOV     DS,AX
Trace Interrupt
-t
AX=1CA5 BX=0000 CX=002C DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=1CA5 ES=1C95 SS=1CA4 CS=1CA7 IP=0005  NU UP DI PL NZ NA PO NC
1CA7:0005 BB1A00    MOV     BX,001Ah
Trace Interrupt
-t
AX=1CA5 BX=001A CX=002C DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=1CA5 ES=1C95 SS=1CA4 CS=1CA7 IP=0008  NU UP DI PL NZ NA PO NC
1CA7:0008 B44C      MOV     AH,4Ch
Trace Interrupt

```

3、有数据定义语句 BUF DW 0ABH, 1, 10 DUP(3 DUP(1, 0), 2)汇编后，为变量 BUF 分配的存储单元字节数是(B)。

- A.48H
- B.90H
- C.120
- D.60

我们构建下面的程序，并将最终的结果输入到 BX 这个寄存器中，我们在最后一次-t 的调试中可以看到 BX 中的内容变成了 0090H

```

8 DATA SEGMENT ;数据段定义
9 BUF DW 0ABH,1,10 DUP(3 DUP(1,0), 2)
0
1 DATA ENDS
2 CODE SEGMENT ;代码段定义
3 ASSUME CS:CODE,DS:DATA ;ASSUME语句
4 START: MOV AX,DATA ;装填DS
5        MOV DS,AX
6
7        MOV BX,sizeof BUF
8
9        MOV AH,4CH ;返回DOS
0 INT 21H
1 CODE ENDS
2 END START ;结束语言

```

```

-t
AX=1CA5 BX=0000 CX=009C DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=1C95 ES=1C95 SS=1CA4 CS=1CAE IP=0003  NU UP DI PL NZ NA PO NC
1CAE:0003 8ED8      MOV     DS,AX
Trace Interrupt
-t
AX=1CA5 BX=0000 CX=009C DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=1CA5 ES=1C95 SS=1CA4 CS=1CAE IP=0005  NU UP DI PL NZ NA PO NC
1CAE:0005 BB9000    MOV     BX,0090h
Trace Interrupt
-t
AX=1CA5 BX=0090 CX=009C DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=1CA5 ES=1C95 SS=1CA4 CS=1CAE IP=0008  NU UP DI PL NZ NA PO NC
1CAE:0008 B44C      MOV     AH,4Ch
Trace Interrupt

```

4、伪指令 DA1 DB 04H DUP(3 DUP(2), 2 DUP(3))在数据区中能构成 0203H 操作数共有 (C)

- A.07 个
- B.04 个
- C.03 个
- D.02 个

我们构建下面的程序，并打开-d ds:0000 观察数据的储存单元

```

DATA        SEGMENT           ;数据段定义
DA1 DB 04H DUP (3 DUP(2), 2 DUP(3))

DATA        ENDS
CODE        SEGMENT           ;代码段定义
ASSUME CS:CODE,DS:DATA       ;ASSUME语句
START:      MOV AX,DATA       ;装填DS
            MOV DS,AX

            MOV AH,4CH        ;返回DOS
            INT 21H
CODE        ENDS
END START           ;结束语言

```

```

-t
AX=1CA5 BX=0000 CX=0029 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=1C95 ES=1C95 SS=1CA4 CS=1CA7 IP=0003  NV UP DI PL NZ NA PO NC
1CA7:0003 8ED8      MOV     DS,AX
Trace Interrupt
-t
AX=1CA5 BX=0000 CX=0029 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=1C95 ES=1C95 SS=1CA4 CS=1CA7 IP=0005  NV UP DI PL NZ NA PO NC
1CA7:0005 B44C      MOV     AH,4Ch
Trace Interrupt
-d DS:0000
1CA5:0000 02 02 02 03 03 02 02 02-03 03 02 02 02 03 03 02 .....
1CA5:0010 02 02 03 03 00 00 00 00-00 00 00 00 00 00 00 .....
1CA5:0020 B8 A5 1C 8E DB B4 4C CD-21 00 00 00 02 00 01 00 8%..X4LM?.....
1CA5:0030 00 00 01 00 00 00 01 00-00 00 02 00 01 00 00 .....
1CA5:0040 01 00 00 00 01 00 00 00-02 00 01 00 00 00 01 .....
1CA5:0050 00 00 01 00 00 00 02 00-01 00 00 00 01 00 00 .....
1CA5:0060 01 00 00 00 02 00 01 00-00 00 01 00 00 00 01 .....
1CA5:0070 00 00 02 00 01 00 00 00-01 00 00 00 01 00 00 .....

```

5、有数据定义 LIST DB 45H, 36H, 18H, 76H, 执行 MOV AX, WORD PTR LIST+2, 则 AX=(D)。

- A.3600H
B.0036H
C.1876H
D.7618H

我们构建下面的程序，并将最终的结果输入到 BX 这个寄存器中，我们在最后一次-t 的调试中可以看到 BX 中的内容变成了 7618H

```

8 DATA        SEGMENT           ;数据段定义
9 LIST DB 45H, 36H, 18H, 76H
0
1 DATA        ENDS
2 CODE        SEGMENT           ;代码段定义
3 ASSUME CS:CODE,DS:DATA       ;ASSUME语句
4 START:      MOV AX,DATA       ;装填DS
5            MOV DS,AX
6
7            MOV AX, WORD PTR LIST+2
8
9            MOV AH,4CH        ;返回DOS
10           INT 21H
11           CODE        ENDS
12           END START           ;结束语言

```



```

-t
AX=1CA5 BX=0000 CX=001C DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=1C95 ES=1C95 SS=1CA4 CS=1CA6 IP=0003 NU UP DI PL NZ NA PO NC
1CA6:0003 8ED8 MOV DS,AX
Trace Interrupt
-t
AX=1CA5 BX=0000 CX=001C DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=1C95 ES=1C95 SS=1CA4 CS=1CA6 IP=0005 NU UP DI PL NZ NA PO NC
1CA6:0005 A10200 MOV AX,10002h
Trace Interrupt
-t
AX=7618 BX=0000 CX=001C DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=1CA5 ES=1C95 SS=1CA4 CS=1CA6 IP=0007 NU UP DI PL NZ NA PO NC
1CA6:0007 B44C MOV AH,4Ch
Trace Interrupt

```

四、对于下面的数据定义，各条 MOV 指令单独执行后，有关寄存器的内容是什么？

```

FLDB      DB      ?
TABLEA DW      20 DUP(?)
TABLEB DB      'ABCD'
(1)  MOV     AX,TYPE FLDB           _____AX = 0001_____
(2)  MOV     AX,TYPE TABLEA       _____AX = 0002_____
(3)  MOV     CX,LENGTH TABLEA     _____CX = 0014_____
(4)  MOV     DX,SIZE TABLEA       _____DX = 0028_____
(5)  MOV     CX,LENGTH TANLEB      _____CX = 0001_____

```

```

AX=0001 BX=0000 CX=0048 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=1CA5 ES=1C95 SS=1CA4 CS=1CA8 IP=0003 NU UP DI PL NZ NA PO NC
1CA8:0003 B80200 MOV AX,0002h
Trace Interrupt
-t
AX=0002 BX=0000 CX=0048 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=1CA5 ES=1C95 SS=1CA4 CS=1CA8 IP=0005 NU UP DI PL NZ NA PO NC
1CA8:0005 B91400 MOV CX,0014h
Trace Interrupt
-t
AX=0002 BX=0000 CX=0014 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=1CA5 ES=1C95 SS=1CA4 CS=1CA8 IP=0007 NU UP DI PL NZ NA PO NC
1CA8:0007 BA2800 MOV DX,0028h
Trace Interrupt
-t
AX=0002 BX=0000 CX=0014 DX=0028 SP=0000 BP=0000 SI=0000 DI=0000
DS=1CA5 ES=1C95 SS=1CA4 CS=1CA8 IP=0011 NU UP DI PL NZ NA PO NC
1CA8:0011 B90100 MOV CX,0001h
Trace Interrupt

```

再把里面的 size 和 length 替换成 sizeof，lengthof，通过程序方式来验证

```

(1)  MOV     CX,LENGTHOF TABLEA   _____CX = 0014_____
(2)  MOV     DX,SIZEOF TABLEA     _____DX = 0028_____
(3)  MOV     CX,LENGTHOF TANLEB    _____CX = 0004_____

```

```

-t
AX=0002 BX=0000 CX=0014 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=1CA5 ES=1C95 SS=1CA4 CS=1CA8 IP=0003 NU UP DI PL NZ NA PO NC
1CA8:0003 BA2800 MOV DX,0028h
Trace Interrupt
-t
AX=0002 BX=0000 CX=0014 DX=0028 SP=0000 BP=0000 SI=0000 DI=0000
DS=1CA5 ES=1C95 SS=1CA4 CS=1CA8 IP=0011 NU UP DI PL NZ NA PO NC
1CA8:0011 B90400 MOV CX,0004h
Trace Interrupt
-t
AX=0002 BX=0000 CX=0014 DX=0028 SP=0000 BP=0000 SI=0000 DI=0000
DS=1CA5 ES=1C95 SS=1CA4 CS=1CA8 IP=0014 NU UP DI PL NZ NA PO NC
1CA8:0014 B44C MOV AH,4Ch
Trace Interrupt

```