

汇编语言作业答案

1. 3 什么是汇编语言源程序、汇编程序、目标程序？

见教材 P4

1. 10 数码 0~9、大写字母 A~Z、小写字母 a~z 对应的 ASCII 码分别是多少？ASCII 码为 0Dh、0aH 对应的是
什么字符？

见教材 P8

1. 11 计算机中有一个“01100001”编码，如果把它认为是无符号数，它是十进制什么数？如果认为它是 BCD
码，则表示什么数？又如果它是某个 ASCII 码，则代表哪个字符？

97,61h,a

1.13 什么是 DOS 和 POM-BIOS？

DOS:磁盘操作系统；POM-BIOS:可编程的基本输入输出系统

2.3 举例说明 CF 和 OF 标志的差异。

见教材 P20-21

2. 5 什么是 8086 中的逻辑地址和物理地址？逻辑地址如何转换成物理地址？请将如下逻辑地址用物理地址表
达：

(1) FFFFh:0 (2)40h:17h (3)2000h:4500h (4)B821h:4567h

见教材 P22-23 FFFF0h 417h 24500h BC777h

2.6 8086 有哪 4 种逻辑段，各种逻辑段分别是什么用途？

见教材 P23

2.10 说明下列指令中源操作数的寻址方式？如果 BX=2000H，DI=40H，给出 DX 的值或有效地址 EA 的值。

(1) mov dx,[1234h] 直接寻址

(2) mov dx,1234h 立即寻址

(3) mov dx,bx 寄存器寻址

(4) mov dx,[bx] 寄存器间接寻址

(5) mov dx,[bx+123h] 寄存器相对寻址

(6) mov dx,[dx+di] 基址变址寻址

(7) mov dx,[bx+di+1234h] 相对基址变址寻址

2.11 已知 DS=2000H、BX=0100H、SI=0002H，存储单元[20100H]~[20103H]依次存放 12 34 56 78H，
[21200H]~[21203H]依次存放 2A 4C B7 65H，说明下列每条指令执行后 AX 寄存器的内容。

(1) mov ax,1200h 1200h

(2) mov ax,bx 0100h

(3) mov ax,[1200h] 2a4ch

(4) mov ax,[bx] 1234h

(5) mov ax,[bx+1100h] 2a4ch

(6) mov ax,[bx+si] 5678h

(7) mov ax,[bx][si+1100h] b765h

2.15 已知 SS=FFA0H、SP=00B0H，画图说明执行下列指令序列时，堆栈区和 SP 的内容如何变化？

mov ax,8057h

push ax SP=00B2 [00B1]=80h

mov ax,0f79h

push ax SP=00B2 [00B2]=0f79h

pop bx BX=0f79h SP=00B1

pop [bx] SP=00B0h

2. 16 给出下列各条指令执行后 AL 值，以及 CF、ZF、SF、OF 和 PF 的状态：

mov al,89h 不影响标志
add al, al 见教材 P38
add al, 9dh 见教材 P38
cmp al, 0bch 见教材 P40
sub al, al 见教材 P39
dec al 见教材 P39
inc al 见教材 P38

2. 17 设均为双字数据，分别存放在地址为的存储单元中，他们的运算结果存入单元。阅读如下程序段，给出运算公式。

```
mov ax,X
mov dx,X+2
add an,Y
adc dx,Y+2
add ax,24
adc dx,0
sub ax,Z
sbb dx,Z+2
mov W,ax
mov W+2,dx      Y+X+24-Z
```

2.18 请分别用一条汇编语言指令完成如下功能：

(1) 把 BX 寄存器和 DX 寄存器的内容相加，结果存入 DX 寄存器。ADD DX,BX

(2) 用寄存器 BX 和 SI 的基址变址寻址方式把存储器的一个字节与 AL 寄存器的内容相加，并把结果送到 AL 中。ADD AL,[BX+SI]

(3) 用 BX 和位移量 0B2H 的寄存器相对寻址方式把存储器中的一个字和 CX 寄存器的内容相加，并把结果送回寄存器中。ADD CX,[BX+0B2H]

(4) 用位移量为 0520H 的直接寻址方式把存储器中的一个字与数 4312H 相加，并把结果送回该存储单元中。ADD [0520H],4321H

(5) 把数 0A0H 与 AL 寄存器的内容相加，并把结果送回 AL 中。ADD AL,0A0H

2.20 指出下列指令的错误

(1) xchg[si],30h 不能用立即数	(2)pop cs 没有错误
(3) sub[si],[di] 不能全为 mem	(4)push ah ah 应为字数据
(5) adc ax,ds 不能用段寄存器	(6)add [si],80h [si]应为 reg 或 mem
(7)in al ,3fch 位数不对	(8)out dx,ah ah 只能用 al 或 ax

2.21 给出下列各条指令执行后的结果，以及状态标志 CF、OF、SF、ZF、PF 的状态。

```
mov ax,1470h
and ax,ax
or ax,ax
xor ax,ax
not ax
```

test ax,0f0f0h 参考教材 P20

2.23 编写程序段完成如下要求：

(1) 用位操作指令实现 AL（无符号数）乘以 10 参考教材 P49 例 2.32

(2) 用逻辑运算指令实现数字 0~9 的 ASCII 码与非压缩 BCD 码的相互转换

参考教材 P47 例 2.28

(3) 把 DX.AX 中的双字右移 4 位参考教材 P49 例 2.32

2. 25 已知数据段 500h~600h 处存放了一个字符串，说明下列程序段执行后的结果：

```
mov si,600h
```

```
mov di,601h
```

```
mov ax,ds
```

```
mov s,ax
```

```
mov cx,256
```

```
std      数据段向后移动一个字，即数据段首地址为 501h 和—601h
```

```
rep movsb
```

2.26 说明下列程序段的功能

```
cld
```

```
mov ax,0fefh
```

```
mov cx,5
```

```
move s,bx
```

```
mov di,2000h
```

```
rep stows  将 0fefh 开始的五个字的内容，送致 32000H 开始的单元中
```

2.32 判断下列程序段跳转的条件

(1) xor ax,leleh

```
je equal    ax=leleh
```

(2)test al,10000001b

```
jnz there  al=10000000 或 0000001 或 10000001
```

(3)cmp cx,64h

```
jb there   cx<64h
```

2.34 假设 AX 和 SI 存放的是有符号数，DX 和 DI 存放的是无符号数，请用比较指令和条件转移指令实现以下判断：

(1) 若 DX>DI,转到 above 执行；CMP DX,DI JA above

(2) 若 AX>SI,转到 greater 执行；CMP AX,SI JA greater

(3) 若 CX=0，转到 zero 执行；JNC zero

(4) 若 AX-SI 产生溢出，转到 overflow 执行；CMP AX,SI JO overflow

(5) 若 SI<=AX,转到 less_eq 执行；CMP SI,AX JBE less_eq

(6) 若 DI<=DX,转到 below_eq 执行。CMP DI,DX JBE below_eq

2.35 有一个首地址为 array 的 20 个字的数值，说明下列程序的功能。

```
mov ex,20
```

```
mov ax,0
```

```
mov si,ax
```

```
sun_loop: add ax,array[si]
```

```
add si,2
```

```
loop sum_loop
```

```
mov total,ax 把 array 中的 20 个字的内容相加，并把结果送至 total 中。
```

2.36 按照下列要求，编写相应的程序段：

(3) 编写一个程序段，在 DX 高 4 位全为 0 时，使 AX=0；否则使 AX=-1。

参考教材 P48 例 2.30

(6) 以知字符串 string 包含有 32KB 内容，将其中的 ‘\$’符号替换成空格。

参考教材 P52 例 2.37

(7) 有一个 100 个字节元素的数组，其首地为 array，将每个元素减 1（不考虑溢出）存于原处。参考教材 P52 例 2.36

2.39 以知 AX、BX 存放的是 4 位压缩 BCD 表示的十进制数，请说明如下子程序的功能和出口参数。

```
add al,bl
daa
xchg al,ah
adc al,bh
daa
xchg al,ah
```

实现 AX、BX 中的 4 位 BCD 数相加，结果仍为 BCD 表示的十进制数，存于 AX 中，出口参数为 ax.

2.44 补充例 2.42,当有溢出时显示 “Error!Overflow!”,无溢出时显示 “OK”。

参看教材 P59 例 2.42 和 P66 例 2.49

3.17 修改例题 3.3，现在用字定义伪指令 dw、字串传送指令 movsw 和字符传显示 9 号功能调用实现。参看教材 P79 例 3.1a 和 P93 例 3.3

3.22 在 SMALL 存储模式下，简化段定义格式的代码段、数据段和堆栈段和却省段名、定位、组合以及类别属性分别是什么？参看教材 P99 (2) SMALL

3.25 按下面要求写一个简化段定义格式的源程序

(1) 定义常量 num，其值为 5；数据段中定义字数组变量 datalist，它的头 5 个字单元中依次存放-1、0、2、5 和 4，最后 1 个单元初值不定；参考教材 P93 例 3.2

(2) 代码段中的程序将 datalist 中头 num 个数的累加和存入 datalist 的最后 1 个字单元中。

```
Num      equ 5
Datalist  dw -1,0,2,5,4,?
          Mov  cx,num
          Mov  bx,offset datalist
          Mov  bx,0
Again:    add  ax,[bx]
          Add  bx,2
          Loop again
          Mov  [bx],ax
```

3.26 按下面要求写一个完整段定义格式的源程序

- (1) 数据段从双字边界开始，其中定义一个 100 字节的数组，同时该段还作为附加段；
- (2) 堆栈段从节边界开始，组合类型为 stack；
- (3) 代码段的类别是'coda',指定段寄存器对应的逻辑段；主程序指定从 100h 开始，给有关段寄存器赋初值；将数组元素全部设置为 64h.

参考教材 P79 例 3.1b

4.11 编制一个程序，把变量 bufX 和 bufY 中较大者存入 bufZ；若两者相等，则把其中之一存入 bufZ 中。假设变量存放的是 8 位无符号数。

参考教材 P115 例 4.3a

4.12 设变量 bufX 为有符号 16 位数，请将它的符号状态保存在 signX，即：如果 X 大于等于 0，保存 0；如果 X 小于 0，保存—1 (ffh)。编写该程序。

参考教材 P121 例 4.6

4.13 bufX、bufY 和 bufZ 是 3 个有符号 16 进制数，编写一个比较相等关系的程序；

- (1) 如果这 3 个数都不相等，则显示 0；
- (2) 如果这 3 个数中有两个数相等，则显示 1；
- (3) 如果这 3 个数都相等，则显示 2。

参考教材 P119 例 4.3b,子程序中加一段显示程序即可。

4.23 编写程序,将一个包含有 20 个有符号数据的数组 arrayM 分成两个数组:正数数组 arrayP 和负数数组 arrayN, 并分别把这两个数组中的数据个数显示出来。

参考教材 P121 例 4.6,

4.24 编写计算 100 个正整数之和的程序。如果和不超过 16 位字的范围 (65535), 则保存其和到 wordsum, 如超过则显示 'overflow'。

```
Parry      dw  23,32,56,...; 假设有 100 个数据
Wordsum    dw  0
Msg        db  'overflow','$'

          Mov  cx,100
          Mov  ax,0
          Mov  bx,offset parry

Again:     add  ax,[bx]
          Jnc  next
          Mov  dx,offset msg
          Mov  ah,9
          Int  21h
          Jmp  done

Next:      add  bx,2
          Loop again
          Mov  wordsum,ax

Done:      ...
```

4.34 编制一个子程序把一个 16 位二进制数用十六进制形式在屏幕上显示出来，分别运用如下 3 种参数传递方法，并用一个主程序验证它。

- (1) 采用 AX 寄存器传递这个 16 位二进制数。
- (2) 采用 wordTEMP 变量传递这个 16 位二进制
- (3) 采用堆栈方法传递这个 16 位二进制数

参考教材 P126 例 4.10,

4.35 设有一个数组存放学生的成绩 (0—100), 编制一个子程序统计 0—59 分, 60—69 分, 70—79 分, 80—89 分, 90—100 分的人数, 并分别存放到 scoreE、scoreD、scoreC、scoreB 及 scoreA 单元中。编写一个主程序与之配合使用。

```
Sarray     db  90,85,70,57,95,...
Count      db  $-sarray
ScoreE     db  0; 记录 0—59 分之间的人数
ScoreD     db  0; 记录 60—69 分之间的人数
ScoreC     db  0; 记录 70—79 分之间的人数
ScoreB     db  0; 记录 80—89 分之间的人数
ScoreA     db  0; 记录 90—100 分之间的人数
Snum       proc
```

```

                                Push  ax
                                Push  dx
                                Push  si
Again:                          mov  al,[bx]
                                Cmp   al,60
                                Jb    next0
                                Cmp   al,90
                                Jae   next1
                                Mov   dl,10
                                Mov   ah,0
                                Div   dl
                                Sub   al,5
                                Mov   ah,0
                                Mov   si,ax
                                Jump  next2
Next0:                          mov   si,0
                                Jump  next2
Next1:                          mov   si,4
Next2:                          inc   scoreE[si]
                                Inc   bx
                                Loop  again
                                Pop   si
                                Pop   dx
                                Pop   ax
                                Ret
Snum                            endp
                                Mov   bx,offset sarray
                                Mov   cx,count
                                Call snum

```

4.51 将例题 4.7 的大写转换为小写字母写成宏完成。

参考教材 P122 例 4.7,和 P115 例 4.13b,

4.52 请用宏结构实现例题 4.4 数据段中 8 个标号组成的地址表,以及代码段的 8 个处理程序段;然后运行通过。自己上机实现

4.57 从键盘输入一个字符串,先将它原样显示一遍;然后将其中的小写字母转换为大写,再将其中的大写字母转换为小写显示,最后将其中的大小写字母互换显示。显示字符串的功能调用采用宏,大写转换、小写转换和大小写互换写成子程序。

- (1) 编写一个完整的源程序完成题目要求
- (2) 把子程序单独写在一个文件中,用源程序包含的方法完成题目要求
- (3) 把子程序单独汇编,用目标文件连接的方法完成题目要求
- (4) 把子程序加入到一个子程序库中,用子程序库调入的方法完成题目要求

以下的解答请参考教材。

5.5 为什么说 32 位通用寄存器比 16 位通用寄存器更通用? 参考教材 P170

5.14 回答下列问题:

- (1) ADD ECX, AX 指令错在哪里?

- (2) INC[BX]指令错在哪里?
- (3) 说明 IMUL BX, DX, 100H 指令的操作
- (4) JECXZ 指令什么条件下转移?
- (5) MOV AX, [EBX+ECX]指令正确吗?
- (6) 32 位 x86 CPU 的 Jcc 指令的转移范围可有多大?
- (7) 如何让汇编程序识别 80386 指令?
- (8) 如何让汇编程序形成 16 位段和 32 位段?

5.20 填空如下程序段的运行结果:

```

mov eax,01234567h      ;EAX=_____
mov edx,5ABCDEF9h      ;EDX=_____
shrd ead,edx,4          ;EAX=_____,EDX=_____,CF=_____
shrd eax,edx,8          ;EAX=_____,EDX=_____,CF=_____
shld dx,ax,1           ;EAX=_____,EDX=_____,CF=_____,OF=_____

```

上机调试例题 2.31: 移位指令的功能

```

mov cl,4
mov al,0f0h            ;al=f0h
shl al,1               ;al=e0h,CF=1, SF=1、ZF=0、PF=0, OF=0
shr al,1               ;al=70h,CF=0, SF=0、ZF=0、PF=0, OF=1
sar al,1               ;al=38h,CF=0, SF=0、ZF=0、PF=0, OF=0
sar al,cl               ;al=03h,CF=1, SF=0、ZF=0、PF=0, OF=1

```

上机调试例题 2.36: 将数据段 source 指示的 100 个字符数传送到附加段 destination 指令的主存区

```

mov si,offset source    ;汇编伪指令 source 的偏移地址
mov di,offset destination
mov cx,100              ;cx←———传送次数
cld                     ;设置 DF=0, 实现地址增加
again:movsb             ; 传送一个字节
    dec cx              ; 传送次数减 1
    jnz again           ; 判断传送次数 cx 是否为 0。不为 0 (ZF=0),
                        ; 则转移到 again 位置继续执行; 否则, 结束

```

上机调试例 2.47: 利用子程序完成将 AL 低 4 位中的一位 16 进制数转换成对应的 ASCII 码

```

; 主程序
mov al,0fh              ;提供参数 al
call htoasc              ;调用子程序
...
;子程序
htoasc: and al,0fh        ; 只取 al 的低 4 位
        or  al,30h        ; al 高 4 位变成 3
        cmp al,39h        ; 是 0~9, 还是 0ah~0fh
        jbe htoend
        add al,7           ;是 0ah~0fh,其 ASCII 还要加上 7 (见表 1.1)

```

htoend: ret ;子程序返回

上机调试例 2.49：用 09 号 DOS 系统功能调用在显示器输出一个字符串

string db'hello,Everybody!','0dh,0ah','\$' ;在数据段定义要显示的字符串

...

mov ah,09h ;设置功能号：ah← 09h

mov dx,offset string ;提供入口参数：dx← 字符串的偏移地址

int 21h ; DOS 功能调用：显示