

第2章 作业

22920212204392 黄勛

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

(1) mov ax,1200h

(2) mov ax,bx

(3) mov ax,[1200h]

(4) mov ax,[bx]

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

(6) mov ax,[bx+si]

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

答：

(1) $AX = 1200H$

(2) $AX = 0100H$

(3) $AX = 4C2AH$ ；偏移地址= $bx=0100h$

(4) $AX = 3412H$ ；偏移地址= $bx=0100h$

(5) $AX = 4C2AH$ ；偏移地址= $bx+1100h=1200h$

(6) $AX = 7856H$ ；偏移地址= $bx+si=0100h+0002h=0102h$

(7) $AX = 65B7H$ ；偏移地址= $bx+si+1100h=0100h+0002h+1100h=1202h$

2.4 什么是堆栈，它的工作原理是什么，它的基本操作有哪两个，对应哪两种指令？

答：

堆栈是一个特定的存储区或寄存器，它的一端是固定的，另一端是浮动的。堆这个存储区存入的数据，是一种特殊的数据结构。所有的数据存入或取出，只能在浮动的一端（称栈顶）进行，严格按照“先进后出”的原则存取，位于其中的元素，必须在其栈上部（后进栈者）诸元素逐个移出后才能取出。在堆栈段中，SS 段寄存器记录其段地址。

工作原理：“先进后出”。

堆栈的两种基本操作是压栈和出栈，对应的指令是 PUSH 和 POP。

2.6 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

add al,al

add al,9dh

cmp al,0bch

sub al,al

dec al

inc al

答：

CF：进位标志位。一般情况下，在进行无符号运算的时候，它记录了运算结果的最高有效位向更高位的进位置，或从更高位借位值。（当作无符号数）

ZF：零标志位。它记录相关指令执行后，其结果是否为 0。如果结果为 0，那么 ZF=1；如果结果不为 0，那么 ZF=0。

SF: 符号标志位。它记录相关指令执行后,其结果是否为负。如果结果为负, sf=1; 如果非负, sf=0。(当作有符号数)

OF: 溢出标志位。在进行有符号数的运算的时候,如经过超过了机器所能表示的范围称为溢出。(当作有符号数)

PF: 奇偶标志位。它记录相关指令执行后,其结果的所有 bit 位中 1 的个数是否为偶数,如果 1 的个数为偶数, pf=1,如果为奇数,那么 pf=0。

(括号中为十进制补码——即实际有符号数)

	CF	ZF	SF	OF	PF
mov al,89h; AL=89H	不改变				
add al,al; AL=12H	1	0	0	1	1
1000 1001	(-119)				
+ 1000 1001	(-119)				
<hr/>					
1 0001 0010	(18)				

	CF	ZF	SF	OF	PF
add al,9dh; AL=0AFH	0	0	1	0	1
0001 0010	(18)				
+ 1001 1101	(-99)				
<hr/>					
1010 1111	(-81)				

	CF	ZF	SF	OF	PF
cmp al,0bch; AL=0AFH	1	0	1	0	1
1010 1111	(-81)				
- 1011 1100	(-68)				
<hr/>					
(-1)1111 0011	(-13)				

	CF	ZF	SF	OF	PF
sub al,al ; AL=00H	0	1	0	0	1
	CF	ZF	SF	OF	PF
dec al ; AL=0FFH	1	0	1	0	1
0000 0000	(0)				
- 0000 0001	(1)				
<hr/>					
(-1) 1111 1111	(-1)				

	CF	ZF	SF	OF	PF
inc al ; AL=00H	1	1	0	0	1
1111 1111	(-1)				
+ 0000 0001	(1)				
<hr/>					
(1) 00000000	(0)				

2.8 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。

(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。

(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，并把结果送回存储器中。

(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 3412H 相加，并把结果送回该存储单元中。

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

答：

- (1) ADD DX,BX
- (2) ADD AL,[BX+SI]
- (3) ADD [BX+0B2H],CX
- (4) ADD **WORD PTR** [0520H],3412H
- (5) ADD AL,0A0H

2.19 假设 DS=2000H，BX=1256H，SI=528FH，位移量 TABLE=20A1H，[232F7H]=3280H，[264E5H]=2450H。试问执行下列段内间接寻址的转移指令后，转移的有效地址是什么？

- (1) JMP BX
- (2) JMP TABLE[BX]
- (3) JMP [BX][SI]

答：

- (1) 转移的有效地址 EA= BX=1256H
- (2) 转移的有效地址 EA= [DS:20A1H+1256H]=[232F7H]=3280H
- (3) 转移的有效地址 EA= [DS:1256H+528FH]=[264E5H]=2450H

2.29 解释如下有关中断的概念：

- (1) 内部中断和外部中断
- (2) 单步中断和断点中断
- (3) 除法错中断和溢出中断
- (4) 中断向量号和中断向量表

答：

- (1) 内部中断是由于 8086CPU 内部执行程序引起的程序中断；
外部中断是来自 8086CPU 之外的原因引起的程序中断；
- (2) 单步中断是若单步标志 TF 为 1，则在每条指令执行结束后产生的中断；断点中断是供调试程序使用的，它的中断类型为 3。通常调试程序时，把程序按程序的任务分成几段，然后每段设一个断点来进行调试，在这时起作用的是断点中断；
- (3) 除法错中断是在执行除法指令时，若除数为 0 或商超过了寄存器所能表达的范围产生的中断；溢出中断是在执行溢出中断指令 INTO 时，溢出标志 OF 为 1 时产生的中断；
- (4) 中断向量号中断源的识别标志，可用来存放中断服务程序的入口地址或跳转到中断服务程序的入口地址；中断向量表就是中断向量号的列表，把系统中所有的中断类型码及其对应的中断向量按一定的规律存放在一个区域内，这个存储区域就叫中断向量表。