# 第二章 计算机基本原理

2.1 简述计算机系统组成。

答：计算机由中央处理器CPU，存储器，输入系统，输出系统组成，由系统总线连接在一起。CPU包括运算器和控制器，运算器执行指令，控制器负责计算机的控制。存储器是计算机的记忆部件，以二进制形式存放程序和数据。输入输出系统包括大容量存储器，如硬盘，以及其他外设，如鼠标，键盘，显示器等。

2.2 简述16位机的各类寄存器的主要作用。

答：（1）通用数据寄存器：AX，BX，CX，DX；

AX：作为累加器，是算术运算的主要寄存器。在乘除等指令中存放操作数，在I/O指令中使用它与外部设备传送信息。

BX：当通用寄存器使用，在计算存储器地址时，作基址寄存器使用。

CX：当通用寄存器使用，此外常用来保存计数值，当计数器使用。

DX：当通用寄存器使用，一般在作双字长运算时把DX和AX组合使用，对某些I/O操作，DX用来存放I/O的端口地址。

（2）地址寄存器：SP，BP，SI，DI

（3）段寄存器：CS，DS，ES，SS；段寄存器的作用是专用于存储器寻址，用来直接或间接地存放段地址。

（4）专用寄存器：IP，FLAGS；IP寄存器专门存放下一条指令的地址，FLAGS标志寄存器，又称程序状态寄存器。它是存放条件码标志、控制标志和系统标志的寄存器。

2.3 写出每条汇编指令执行后相关寄存器的值。

mov ax, 1345H ax=1345H

mov ah, 24H ax=2445H

mov al, 45H ax=2445H

mov bx, 3412H bx=3412H

mov al, bh ax=2434H

mov ah,hl ax=1234

2.4实模式下，写出段地址和偏移地址为1234:2002、1430:0042、FF00:0FFF的物理地址。

答：

1234:2002=12340+2002=14342

1430:0042=14300+0042=14342

FF00:0FFF=FF000+0FFF=FFFFF

2.5下列各数均为十进制数，请采用8位二进制补码运算，并回答标志寄存器FLAGS中CF和OF的值，运算结果所代表的十进制数是多少？如果用16位二进制补码运算，其结果所代表的十进制数是多少？FLAGS中CF和OF的值呢？

(1)85+69 (2)85+ (-69) (3)85-(-69) (4)85-(69)

答：85=55H, 69=45H,-69=BBH

（1）85+69=55H+45H=01010101 + 01000101=9AH=-26D

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |

2.6 给定段地址为0001H，仅通过变化偏移地址寻址，CPU的寻址范围从 00010H到 10000FH 。

段地址×16=00010，偏移地址是从0000H～FFFFH，这样答案就是00010H，1000FH

2.7 2.7 有一数据存放在内存20000H单元中，现给定段地址为SA，若想用偏移

地址寻到此单元，则SA应满足的条件是：最小为\_\_\_\_\_\_\_，最大为\_\_\_\_\_\_\_\_。

最大地址为：SA\*16+0H=20000H—>SA=2000H

最小地址为：SA\*16+FFFFH=2000H—>SA\*10H=10001H

因为10001H / 10H= 1000H 余 1H，所以最小地址应该大于1000H，因为

1000H下面就是1001H，所以最小地址就是1001H了。

2.8 已知8086系统某存储单元物理地址为：52506H，你认为段基址的最大值、最小值分别是多少？8086微机最多可以有多少个不同的段基址

52506=5250：0006，段基址的最大值=5250

52506=4251：FFF6，段机制的小值=4251

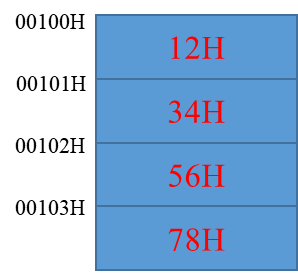
8086系统有20根地址线，寻址范围可达1M范围（00000H～0FFFFH）。因为16个单元为一个小段，每一个逻辑单元必须从一个小段开始，所以最多段的个数1M/16=64K=65536。因为每个逻辑段最大可以有64K个单元，所以最小可以有逻辑段1M/64K=16个。

**存储器地址分段**：从0地址开始，把1MB的存储器分成若干段，16字节为一

小段，段的起始地址必须从任意一小段的首地址开始，也就是说，当一个段开始的物理地址表示成20位的二进制地址码，最低位是0。

2.9从物理地址为00100H开始到00103H单元中顺序存放的数据为：12H，34H，56H，78H。请画出数据存放示意图，并回答以下问题：

(1)写出00101H字节单元的内容  
(2)写出00102H字单元的内容  
答案：  
(1)(00101)= 34H  
(2)(00102)= 7856H



# 第五章 常用指令系统

**5.1 溢出标志OF与进位标志CF有何作用和区别？**

答：处理器对两个操作数进行运算时，按照无符号数求得结果，并相应设置进位标志CF；同时，根据是否超出有符号数的范围设置溢出标志OF。应该利用哪个标志，则由程序员来决定。也就是说，如果将参加运算的操作数认为是无符号数，就应该关心进位；认为是有符号数，则要注意是否溢出。

5.3 CMP和TEST指令与其他指令的不同之处在于什么？它们通常都紧跟着跳转指令，用在什么场合？

答：CMP和TEST指令分别执行减法和逻辑与操作，但不回送结果，只影响标志位，通常用以判断两个数的大小或者用以判断操作数的某些位是1或0

**5.4 除了用4CH号系统功能调用结束程序的执行并退出，还有哪些办法？**

答：RET指令，INT 20H，但要正确使用。

**5.5 V是变量，指出下列错误的指令，说出错误原因，并给出合适的修改**

（1） MOV AX，[DX]

将DX改成BX或BP，SI，DI。因为DX不能储存偏移地址。只有BX或BP，SI，DI才能存放偏移地址。

（2） MOV DS，DATA

DATA不是指令，而是伪指令，实际上是一个动态的内存地址，要想运行，必须先放入到DS中，但是内存数不能放入段地址。MOV AX,DATA起到过渡作用。

（3） MOV CS，AX

因为CS是代码段寄存器，这个寄存器保存的是代码段的首地址，这个寄存器只能由系统管理，用户不能随便修改，MOV CS , AX 会修改CS，所以不允许这种指令执行。

（4）MOV AX，DL

长度不一致。

（5）PUSH AL

PUSH OPRD；OPRD为16位的寄存器或存储器操作数。

（6）ADD [BX]，[DI]

两个操作数不能同时为存储器。

（7）LEA [BX]，V

LEA为有效地址送寄存器指令。

（8）MOV [DX]，OFFSET V

[DX]，这是寄存器间接寻址。但是，寄存器间接寻址，只能用 BX、BP、SI、DI。

（9）MOV [SI]，2

在汇编程序无法判断数据类型时，需要专门指定。立即数 2 的数据类型不确定，可能是BYTE、WORD、DWORD等，所以应改为 MOV BYTE PTR [SI]，2 。

（10）MUL BX，CX

X86 CPU里MUL指令不支持两个操作数，IMUL才行。

（11）DIV 5

DIV SRC，SRC不能为立即数。

（12）MOV BYTE[SI]，AX

（13）MOV AX，[SI+DI]

SI和DI都是变址寄存器，没有变址+变址这样的寻址方式的，所以是错的，只有变址+基址寻址方式。

（14）SHR AX，4

SHR AX，1或者SHR AX，CL

（15）CMP 6，AX

CMP AX，6

（16）MOV [FFFF]，AX

[FFFF]是这个段的末尾，只有一个存储单元(8 byte) ，AX寄存器长度为16 byte，所以发生了源操作数和目的操作数长度不一致的错误。

（17）MOV AX，BX+4

MOV AX，[BX+4]  
（18）JMP FAR PRO

JMP FAR PTR PRO

1． MOV 1000H ， BX ；指令错误，原因:立即数不能作为目标操作数。

2． MOV BX ， CL ；指令错误，原因:字长不一致。

3． INC [BX] ；指令错误，原因:需指定操作数存储器操作字。

4． MOV [BX] ， [6014H] ；指令错误，原因:不能同时为存储器操作字。

5． ROL AL， 2 ；指令错误，原因:循环次数超过 1 用 CL 寄存器。

6． MOV CS， AX；指令错误，原因:CS是代码段寄存器，这个寄存器保存的是代码段的首地址，并且这个寄存器只能由系统管理，用户不可更改。原因也可参照下列要求第五条。

7． MOV AL， [SI+DI]；指令错误，原因:同时出现了两个变址寄存器。在基址变址相对寻址中，基址寄存器只能是BX/BP，变址寄存器只能是SI/DI。同理： MOV AL， [BX+SI]正确。

8． MOV DS,0100H;错误。源操作数是立即数时，目的操作数不能是段寄存器。

9． OUT 310H, AL;错误。端口直接寻址的范围应在0~ FFH之间。

10． MOV ES:[BX+ DI],AX ;正确。

11． MOV SS:[BX+SI+ 100H],BX ;正确。

12． ADD [SI], 20H ;错误,目的操作数类型不明确。

13． PUSH 2000H;错误。堆栈指令的操作数不能是立即数。

14． INC [DI] ;错误目的操作数类型不明确。

15． OUT BX, AL;错误。端口间接寻址的寄存器只能是DX寄存器。

16． XCHG CX, DS ;错误。交换指令中不能出现段寄存器。

17． POP AL;错误。堆栈指令的操作数只能是字操作数(即1 6位操作数)。

18． ADD BYTE PTR[BP],256 指令错。ADD指令要求两操作数等字长。

19． MOV DATA[SI],ES:AX 指令错。源操作数形式错，寄存器操作数不加段重设符。

20． JMP BYTE PTR[BX] 指令错。转移地址的字长至少应是1 6位的。

21． OUT 230H,AX 指令错。对输人输出指令，当端口地址超出8位二进制数的表达范围(即寻址的端口超出256个)时,必须采用间接寻址DX。

22． MOV DS,BP 指令正确 。

23． MUL 39H 指令错。MUL指令不允许操作数为立即数。

24． SBB DX , 10H 对。

25． INC [BP] 错 没有给出操作数的类型。

指令对操作数的要求。

MOV指令中两个操作数字长必须相同。

两个操作数不能同时为存储器操作数。若要在两个存储器单元之间进行数据传送，需要两条MOV指令实现。（即用一个寄存器做中间数传参）

不能用立即数直接给段寄存器赋值。

两个操作数不能同时为段寄存器。

一般情况下，指令指针IP及代码段寄存器CS的内容不通过MOV指令修改，即他们不能作为目标操作数，但可以作为源操作数。

虽然许多指令的执行对状态寄存器FLAGS的标志位产生影响，但通常情况下，FLAGS整体不能作为操作数。

**汇编语言中，MOV是最基本的一种操作。**

格式：MOV dest，src

操作：将src的内容移至dest，移动后src的内容仍在，dest的内容被src覆盖。

其规则如下：

1.CS、IP的值不可以作为目标操作数；

2.dest、src不可以同时作为存储器操作数出现；

3.段寄存器不能相互转送；

4.不能把立即数送人段寄存器。

常见错误写法：

1.

DATA DB 10H

MOV [SI],DATA

解释：[SI]指向的是内存内的数，违反规则2，所以此语句错误。

改正：MOV AX,DATA;

MOV [SI],AX

2.

MOV CS,AX

MOV DS,1000H

解释：见规则1、4。正确写法：MOV AX,CS(虽然这个没有什么实际意义，但是仅说明这种写法是可行的，即把段寄存器的内容传到寄存器里)；MOV BX,1000H;MOV DS,BX

3.

MOV AX,DL

解释：前后数据类型不匹配，所以不能传送。可以改成MOV AL,DL

4.

MOV DATA,AH

解释：DATA的类型没有说明，如果是DB类型，那么这条语句没错，否则前后类型不一致，错误。改正：DATA必须是DB类型。

5.

DATA DW 1234H

MOV DATA,WORD PTR AH

解释：PTR无法对寄存器进行操作。改正：MOV BYTE PTR DATA,AH。举例说AH的值是00H，操作完成后，内存中的数应该是00H 12H。

6.

MOV [SI],10H

解释：MOV不可以把立即数送入内存中。改正：MOV AH,10H MOV BYTE PTR[SI],AH。

总结一下：MOV命令的书写必须遵守开始列出的四条规则，除此之外注意操作类型需要一致。

**5.6 在数据段定义了ARRAY数组，其中依次存储了4个字数据，根据以下要求把第4个字送AX寄存器。**

(1)直接寻址

(2)使用BX的间接寻址

(3)使用BX和 ARRAY的寄存器相对寻址

(4)基址变址寻址

(5)MOV以外的其它指令

答：

(1)直接寻址

MOV AX，ARRAY+6 或MOV AX，[ARRAY+6]

(2)使用BX的间接寻址

LEA BX，ARRAY+6 ；

MOV AX，[BX]

(3)使用BX和 ARRAY的寄存器相对寻址

LEA BX，ARRAY ；

MOV AX，[BX+6]

(4)基址变址寻址

LEA BX，ARRAY ；

MOV SI，6 ；

MOV AX，[BX+SI]

(5)MOV以外的其它指令

SUB AX，AX；

ADD AX，[ARRAY+6]

**5.7 画出数据在数据段中的存放情况，程序执行后，BX、DI、CX、DX寄存器中的内容是什么? 程序如下：**

data segment

array dw 20，30，40，20h，30h，-6

buff db ‘abcd$’

data ends

code segment

assume cs:code,ds:data

start:

mov ax, data

mov ds, ax

mov bx, array+1

mov di, offset array

mov cx, [di+5]

mov dl, buff+3

mov ah, 4ch

int 21h

code ends

end start

答： 14 00 1E 00 28 00 20 00 30 00 FA FF 61 62 63 64 $

(BX)=1E00H，(DI)=0000H，(CX)=2000H，(DX)=0064H

**5.8 在DEBUG下设置(SP)=20H，设置AX、BX、CX、DX为不同值，把这四个寄存器内容依次压入堆栈，再从堆栈中依次弹出到SI、DI、BP、BX寄存器。写出一段程序实现上述操作，并画出每条入栈指令执行后SP和堆栈中数据的变化。**

答：

code segment

assume cs:code

start:

push AX

push BX

push CX

push DX

pop SI

pop DI

pop BP

pop BX

hlt

code ends

end start

**5.9 求出7450H与以下各十六进制数的和及差，并根据结果标出SF、ZF、CF、OF标志位的值。**

(1) 1234H (2)5678H (3)9804H (4)E0A0H

答： SF、ZF、CF、OF

(1)1234H+7450H， 1 0 0 1

(2)5678H+7450H， 1 0 0 1

(3)9804H+7450H， 0 0 1 0

(4)E0A0H+7450H， 0 0 1 0

**5.10 在数据段有32位的无符号数变量X,Y,按如下格式定义，其中‘？’请用数值代替，用16位指令按要求写出程序。**

X DW ？，？

Y DW ？，？

Z DW ？，？，？，？

(1)Z=X+Y。

(2)Z=X-Y。

(3)Z=|X-Y|。

(4)Z=X×Y。

答：(1)X和Y两个字数据相加，和存放在Z中。

MOV AX，X

MOV DX，X+2

ADD AX，Y

ADC DX，Y+2

MOV Z，AX

MOV Z+2，DX

(2)MOV AX，X

MOV DX，X+2

SUB AX，Y

SBB DX，Y+2

MOV Z，AX

MOV Z+2，DX

(3)MOV AX，X

MOV DX，X+2

SUB AX，Y

SBB DX，Y+2

TEST AX,8000H

JZ Exit

NOT AX

NOT DX

ADD DX,1

ADC AX,0

Exit: MOV Z，AX

MOV Z+2，DX

(4)MOV AX，X

MUL Y

MOV Z，AX

MOV Z+2，DX

MOV AX，X+2

MUL Y

ADD Z+2,AX

ADC DX,0

MOV Z+4,DX

MOV AX，X

MUL Y+2

ADD Z+4，AX

ADC DX,0

MOV Z+6,DX

MOV AX，X+2

MUL Y+2

ADD Z+6,AX

ADC DX,0

MOV Z+8,DX

**5.11用移位指令为主实现对AX中的无符号数乘以5，不考虑乘积可能超出16位。**

答： MOV DX，AX

MOV CL，2

SHL AX，CL

ADD AX，DX

**5.12 用移位指令为主实现对AX中的无符号数乘以5，考虑乘积可能超出16位的情况。**

答： MOV DX, 0

MOV BX，AX

SHL AX，1

ADC DX, 0 ;作用是将进位标志值1加入到DX。

SHL DX, 1

SHL AX，1

ADC DX, 0

ADD AX，BX

ADC DX, 0

5.13 把AX中的内容依次倒排序，即第0位移到第15位，第1位移到第14位，…。

答：先设AX值

mov di, 0

mov bx,0

Mov si,1

Mov cl,2

Ror ax,1

K: Rol ax,cl

Mov bx,ax

And bx,si

Add di,bx

Shl,si,1

Cmp si,0

Jnz k

Mov ax,di

5.14 在数据段有如下定义：

BUFF DB ‘ABCD$EFGHIJK$’

STR1 DB 12 DUP(?)

LEN DB ？

用串指令编写程序完成以下操作：

(1) 对字符串STR1全部置‘\*’符。

(2) 从左到右把BUFF中的字符串传送到STR1。

(3) 从右到左把BUFF中的字符串传送到STR1。

(4) 比较BUFF与STR1两个字符串是否相等，如相等则DX=1，否则DX=0。

(5) 查找BUFF中有无字符$，把字符$出现的次数计入BX寄存器。

答：

(1)

MOV AX，DATA

MOV DS，AX

MOV AL，‘\*’

LEA DI，STR1

MOV CX，STR1-BUFF

CLD

REP STOSB

(2)

MOV AX，DATA

MOV DS，AX

MOV ES，AX

CLD

LEA SI，BUFF

LEA DI，STR1

MOV CX，STR1-BUFF

REP MOVSB

(3)

MOV AX，DATA

MOV DS，AX

MOV ES，AX

STD

LEA SI，STR1-1

LEA DI，LEN-1

MOV CX，STR1-BUFF

REP MOVSB

(4)

MOV AX，DATA

MOV DS，AX

MOV ES，AX

CLD

LEA SI，BUFF

LEA DI，STR1

MOV CX，STR1-BUFF

REPE CMPSB

(5)

MOV AX，DATA

MOV ES，AX

MOV BX，0

CLD

MOV AL，‘$’

LEA SI，BUFF

MOV CX，STR1-BUFF

NEXT: REPNE SCASB

JCXZ NO-FOUND

INC BX

JMP NEXT

5.15 对于给定的AX和BX的值，执行下列程序段，程序将转向哪里？

ADD AX, BX

JNO L1

JNC L2

SUB AX，BX

JNC L3

JNO L4

JMP L5

(1) AX=1234H， BX=6789H

(2) AX=790EH， BX=8831H

(3) AX=E002H， BX=8086H

答：L1，L1，L5

**5.16 下面不完整的程序段是比较AX和BX的值，把其中大的数送MAX变量。如果是无符号数，应如何填写指令？如果是有符号数，应如何填写指令？**

CMP AX，BX

( )

MOV MAX，AX

( )

L1：MOV MAX，BX

L2：HLT

答： JB L1

JMP L2

5.17 在下列程序段的括号中分别填入如下指令，程序执行完后，AX、CX的内**容是什么？**

(1) LOOP L1

(2) LOOPE L1

(3) LOOPNZ L1

MOV AX， 6

MOV CX，3

L1：ROL AX，CL

TEST AL，3

( )

答：（1）AX=0C000H，CX=0000

（2）AX=0003H，CX=0002

（3）AX=0C000H，CX=0000

5.18 测试AL寄存器，如果最高位和最低位同时为0，则转L0，如果最高位和最低位只有一位为1，则转L1，如果最高位和最低位同时为1，则转L2。画出流程图，并编写程序段。

答：

Code segment

Assume cs:code

Start:

ROR AL,1

AND AL,3

JZ L0

SUB AL,3

JZ L2

JMP L1

L0：

JMP Exit

L1：

JMP Exit

L2：

Exit：hlt

code ends

end start

**5.19 从键盘输入一个英文字母，显示输出其大写。画出流程图，并编写程序段。**  
答：

Code segment

Assume cs:code

Start:

Mov ah,1

Int 21h

sub al,20h

mov dl,al

mov ah,2

int 21h

hlt

code ends

end start

**5.20 从键盘输入一位数字N，显示输出N个N。画出流程图，并编写程序段。**

答：

CODE SEGMENT

ASSUME CS:CODE

START:

MOV AH,1

INT 21H

MOV DL,AL

AND AL, 0FH

MOV CL,AL

MOV CH,0

NEXT: MOV AH,2

INT 21H

LOOP NEXT

MOV 4CH

INT 21H

CODE ENDS

END START

**5.21 在数据段有压缩的BCD码表示的十进制数，写出指令分别完成十进制加法A+B,K+J和减法A-B,K-J。结果放在AX,回答AX的内容。**

A DB 65H,

B DB 37H

K DB 98H

J DB 69H

答：(1) MOV Al,A

MOV Bl,B

ADD AL,BL

AAA

SUB AL,BL

DAS

(2) MOV Al,K

MOV Bl,J

ADD AL,BL

AAA

SUB AL,BL

DAS

(1)A+B=9CH,AX=0102H.A-B=2EH,AX=0028H

(2)K+J=01H,AX=0107H.K-J=2FH,AX=0029H

**5.22 用相对基址变址寻址方法求ARRAY行列式的值。**

ARRAY DW 2，3，5

DW 1，4，6

DW 2，7，9

答：

data segment

Array dw 2,3,5

dw 1,4,6

dw 2,7,9

data ends

code segment

assume cs:code,ds:data

start:

mov ax,data

mov ds,ax

mov ax,0

mov bx,0fffah

a: mov si,0

add bx,6

cmp bx,13

jg d

b: add ax,array[bx][si]

cmp si,3

jg a

add si,2

jmp b

d: hlt

code ends

end start