可以修改IP，或同时修改CS和IP的指令统称为转移指令。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | AX  AH、AL | accumulator  累加寄存器 | | BX  BH、BL | base  基址寄存器 | | CX  CH、CL | count  计数寄存器 | | DX  DH、DL | data  数据寄存器 | | SP | Stack Pointer  堆栈指针寄存器  栈顶元素的偏移地址 | | BP | Base Pointer  基址指针寄存器 | | SI | Source Index  源变址寄存器 | | DI | Destination Index  目的变址寄存器 | | |  |  | | --- | --- | | IP | Instruction Pointer  指令指针寄存器  CPU当前读取指令的偏移地址 | | |  |  | | --- | --- | | CS | Code Segment  代码段寄存器  CPU当前读取指令的段地址 | | DS | Data Segment  数据段寄存器  要访问的数据的段地址 | | ES | Extra Segment  附加段寄存器 | | SS | Stack Segment  堆栈段寄存器  栈顶元素的段地址 | |

Flag

标志寄存器

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  | OF  Overflow Flag  溢出标志 | DF  Direction Flag  方向标志 | IF  Interrupt Flag  中断标志 | TF  Trap Flag  陷阱标志 | SF  Sign Flag  符号标志 | ZF  Zero Flag  零标志 |  | AF  Auxiliary carry Flag  辅助进位标志 |  | PF  Parity Flag  奇偶标志 |  | CF  Carry Flag  进位标志 |
| 0 |  |  |  |  | NV  No Overflow | UP  UP | DI  Disable Interrupt |  | PL  PLus | NZ  Not Zero |  | NA  No Auxiliary |  | PO  Parity Odd |  | NC  No Carry |
| 1 |  |  |  |  | OV  OVerflow | DN  DowN | EI  Enable  Interrupt |  | NG  NeGative | ZR  ZeRo |  | AC  Auxiliary Carry |  | PE  Parity Even |  | CY  CarrY |

立即数

|  |  |  |  |
| --- | --- | --- | --- |
|  | debug | 源文件 | |
| 二进制 | / | 后缀b或B | 1、可以加一层中括号。 |
| 八进制 | / | 后缀o或O |
| 十进制 | / | 无后缀或后缀d或D |
| 十六进制 |  | 不能以字母开头，后缀h或H |
| 字符 | / | '**字符**' |  |
| 段地址 | / | **段名** |  |
|  |  | offset **标号** | 1、相当于标号相对于当前段的偏移地址。例：   |  | | --- | | assume cs:codesg  codesg segment  start:  mov ax,offset codesg  mov ax,offset start  mov ax,offset s1  mov ax,offset s2  s1:  mov ax,offset codesg  mov ax,offset start  mov ax,offset s1  mov ax,offset s2  s2:  mov ax,offset codesg  mov ax,offset start  mov ax,offset s1  mov ax,offset s2  codesg ends  end start | |

|  |  |
| --- | --- |
| define byte  define word  define double word  (db|dw|dd) **立即数**(,**立即数**)\*  (db|dw|dd) **正整数** dup(**立即数**(,**立即数**)\*) |  |

内存单元

|  |  |
| --- | --- |
| ~~[~~**~~立即数~~**~~]~~  1、只能用于debug中输入指令。  **段寄存器**:**立即数**  **段寄存器**:[**立即数**] |  |
| (**段寄存器**:)?[bx(+**立即数**)?]  (**段寄存器**:)?[si(+**立即数**)?]  (**段寄存器**:)?[di(+**立即数**)?]  1、若不写立即数则默认0。  2、若不写段寄存器则默认ds。  (**段寄存器**:)?[bp(+**立即数**)?]  3、若不写段寄存器则默认ss。 | 1、寄存器提供结构体地址，立即数提供结构体的成员的“偏移”地址。立即数相当于c语言里的.**成员名**、->**成员名**运算符。 |
| (**段寄存器**:)?[bx+si]  (**段寄存器**:)?[bx+di]  1、若不写段寄存器则默认ds。  (**段寄存器**:)?[bp+si]  (**段寄存器**:)?[bp+di]  2、若不写段寄存器则默认ss。 | 1、第1个寄存器提供数组的地址，第2个寄存器提供数组的元素的“偏移”地址。 |
| (**段寄存器**:)?[bx+si+**立即数**]  (**段寄存器**:)?[bx+di+**立即数**]  1、若不写段寄存器则默认ds。  (**段寄存器**:)?[bp+si+**立即数**]  (**段寄存器**:)?[bp+di+**立即数**]  2、若不写段寄存器则默认ss。 |  |

|  |  |  |
| --- | --- | --- |
| 有寄存器 | | / |
| 没有寄存器 | 指令只进行字节操作或只进行字操作 |
| 字节操作 | 命令名后插入 byte ptr |
| 字操作 | 命令名后插入 word ptr |
| 双字操作 | 命令名后插入 dword ptr |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| inc  1、内容加1。  dec  2、内容减1。   |  |  | | --- | --- | | 立即数 | × | | 内存单元 | 需要单元长度 | | 寄存器 |  | | 段寄存器 | × | |  |
| add  sub  逻辑与指令and  逻辑或指令or   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  |  | 右 | | | | |  |  | 立即数 | 内存单元 | 寄存器 | 段寄存器 | | 左 | 立即数 | × | | | | | 内存单元 | 需要单元长度 | × |  | × | | 寄存器 |  |  |  | × | | 段寄存器 | × | | | | |  |
| adc  sbb   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  |  | 右 | | | | |  |  | 立即数 | 内存单元 | 寄存器 | 段寄存器 | | 左 | 立即数 | × | | | | | 内存单元 | 需要单元长度 | × |  | × | | 寄存器 |  |  |  | × | | 段寄存器 | × | | | | |  |
| mul  1、若两个乘数都是8位（AL存放），则AX存储结果；若两个乘数都是16位（AX存放），则DX存储结果高16位，AX存储结果低16位。  div  2、若除数为8位，则被除数为16位（AX存放），AL存储商，AH存储余数；若除数为16位，则被除数为32位（DX存放高16位，AX存放低16位），AX存储商，DX存储余数。   |  |  | | --- | --- | | 立即数 | × | | 内存单元 | 需要单元长度 | | 寄存器 |  | | 段寄存器 | × | |  |
| 传送指令mov   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  |  | 右 | | | | |  |  | 立即数 | 内存单元 | 寄存器 | 段寄存器 | | 左 | 立即数 | × | | | | | 内存单元 | 需要单元长度 | × |  |  | | CS | × | | | | | 除CS寄存器 |  |  |  |  | | 段寄存器 | × |  |  | × | |  |
| push  1、SP-=2，数据写入SS:SP。  pop  2、将SS:SP数据读出，SP+=2。   |  |  | | --- | --- | | 立即数 | × | | 内存单元 |  | | 寄存器 |  | | 段寄存器 |  | |  |
| 无条件转移指令jmp   |  |  | | --- | --- | | 段内短转移 | jmp short 标号  1、这里的标号不包括段名。  2、(标号的地址-jmp指令后字节的地址)在-128到127之间。 | | 段内近转移 | jmp near ptr 标号  1、这里的标号不包括段名。  2、(标号的地址-jmp指令后字节的地址)在-32768到32767之间。  jmp word ptr内存单元  1、内存单元存放目的偏移地址。 | | 段间转移 | ~~jmp 段地址:偏移地址~~  1、只能用于debug中输入指令  jmp far ptr 标号  2、这里的标号不包括段名。  jmp dword ptr 内存单元  3、高地址处的字存放目的段地址，低地址处的字存放目的偏移地址。 |   jmp 寄存器  1、修改IP的内容。 |  |
| 有条件转移指令jcxz  jcxz 标号  1、若CX==0，跳转。  2、这里的标号不包括段名。  3、(标号的地址-jmp指令后字节的地址)在-128到127之间。 |  |
| 循环指令loop  loop 标号  1、先CX-=1，后决定是否跳转。若CX!=0，跳转。  2、这里的标号不包括段名。  3、(标号的地址-jmp指令后字节的地址)在-128到127之间。 | 1、例：   |  |  | | --- | --- | | assume cs:codesg  codesg segment  start:  db 126 dup(0)  loop start  codesg ends  end start | assume cs:codesg  codesg segment  start:  db 127 dup(0)  loop start  codesg ends  end start | | assume cs:codesg  codesg segment  loop start  db 127 dup(0)  start:  codesg ends  end start | assume cs:codesg  codesg segment  loop start  db 128 dup(0)  start:  codesg ends  end start | | 正确 | 错误 | |
| 转移指令ret  ret  1、相当于pop IP。  retf  2、相当于pop IP pop CS。 |  |
| 转移指令call   |  | | --- | | call 标号  1、相当于push IP jmp near ptr 标号。  2、这里的标号不包括段名。  call 16位寄存器  3、相当于push IP jmp 16位寄存器。  call word ptr 内存单元  4、相当于push IP jmp word ptr内存单元。 | | call far ptr 标号  1、相当于push CS push IP jmp far ptr 标号。  2、这里的标号不包括段名。  call dword ptr 内存单元  3、相当于push CS push IP jmp dword ptr 内存单元。 | |  |
| cmp  1、对标志寄存器的影响与sub指令相同，但不保存减的结果。   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  |  | 右 | | | | |  |  | 立即数 | 内存单元 | 寄存器 | 段寄存器 | | 左 | 立即数 | × | | | | | 内存单元 | 需要单元长度 | × |  | × | | 寄存器 |  |  |  | × | | 段寄存器 | × | | | | | 1、   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 无符号数 | |  |  |  | | --- | --- | --- | |  | ZF | CF | | 0 | 左≠右 | 左≥右 | | 1 | 左＝右 | 左＜右 | | | 有符号数 | |  |  | | --- | --- | |  | ZF | | 0 | 左≠右 | | 1 | 左＝右 |  |  |  |  |  | | --- | --- | --- | --- | |  |  | SF | | |  |  | 0 | 1 | | OF | 0 | 左≥右 | 左＜右 | | 1 | 左＜右  左－右＜下界 | 左＞右  左－右＞上界 | | |
| 根据无符号数比较结果转移   |  |  | | --- | --- | | je  (jump equal) | 相等（ZF＝1） | | jne | 不相等（ZF＝0） | | jb  (jump below) | 小于（CF＝1） | | jnb | 大于等于（CF＝0） | | ja  (jump above) | 大于（CF＝0∧ZF＝0） | | jna | 小于等于（CF＝1∨ZF＝1） | |  |
| int 立即数 |  |