

标志寄存器

标志	真值为1	假值为0
OF	00	00
SF	NG	PL
ZF	ZR	NZ
PF	PE	PO
CF	CY	NC
DF	DN	UP

CF标志位Carry Flag

- CF CY NC CY = Carry Yes (进位) NC = not Carry
- 进位（最高位进位）add
- 和运算相关的指令会影响标志位 like add , sub
- 把操作数当作无符号数字

ZF标志位Zero Flag

- ZF ZR NZ ZR = Zero NZ = not ZERO
- 判断相等
- 最后结果是否为零

PF标志位pairty Flag

- 一的个数是否位偶数0 or 奇数1
- PF PE PO PE = EVEN 偶数 0 = ODD

SF标志位Sign Flag

- 正0负1
- SF NG PL NG = negative 负数 PL = Positive
- 计算的结果看陈整数和负数
- add sub 影响sf
- mul 不影响sf

OF标志位Overflow



-
- 运算过程中看是否溢出
- 两个操作数都当做有符号 运算过程中决定是否溢出

adc 带进位的加法寄存器

- 可以对更大的数字进行加法运算

sbb 带借位减法

- 实现对更大数的减法

cmp 比较指令

- 类似于减法指令 只是不保存结果，只是影响相关的标志位寄存器
- 可以判断两个操作数的大小 通过 **sf of** 标志位

<i>SF</i>	<i>OF</i>	大小
1	0	1<2
0	1	1<2
1	1	1>2
0	0	1>2

- 检测比较结果的转移指令
 - 和 **cmp**指令配合使用

•

指令	含义	检测相关的标志位
<i>je</i>	<i>equal</i>	<i>zf=1</i>
<i>jne</i>	<i>not equal</i>	<i>zf=0</i>
<i>jb</i>	<i>below</i>	<i>cf=0</i>
<i>jnb</i>	<i>not below</i>	<i>cf=1</i>
<i>ja</i>	<i>above</i>	<i>cf=0 && zf=1</i>
<i>jna</i>	<i>not above</i>	<i>cf=1 zf=1</i>

DF 标志和串传送指令

- **movsb**
- **movsw**
- 配合**rep**使用 **rep like loop** 由**cx**的大小决定 执行 上述 两条指令 的次数
 - **exp**

```
1 ;=====movsb=====
2 mov cx,16
3 rep movsb
4 ;循环16次 每次执行完后 si di ++
```

```
1 ;=====movsw
2 mov cx,16
3 rep movsw
4 ;循环16次每次 执行完 si,di --
```

- cld -> df==0++
- std -> df==1--

pushf && popf

- 使 标志位寄存器 压栈和出栈