

DS段地址

将al中的数据送入内存单元10000H中

```
mov bx,1000H
mov ds,bx
mov [0],al
```

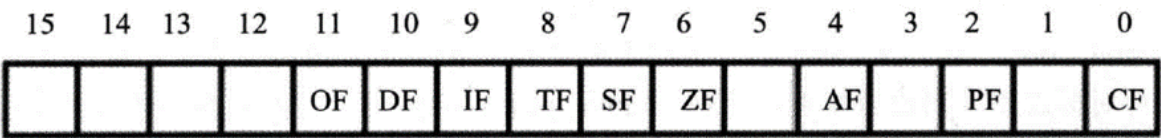
SUB指令

sub 操作数A,操作数B
影响ZF标志位

标志寄存器

8086 CPU 标志寄存器有16位，其中存储的信息通常被称为程序状态字PSW（Program State Word）。

- 用来存储相关指令的某些执行结果
- 用来为CPU执行相关指令提供行为依据
- 用来控制CPU的相关工作方式



ZF标志位

flag的第6位是ZF，零标志位，记录相关指令执行后，其结果是否为0。如果结果为0，ZR（Zero），ZF=1，如果结果不为0，NZ（Not Zero），ZF=0。

```
C:\Users\Administrator>debug
-r
AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0B20 ES=0B20 SS=0B20 CS=0B20 IP=0100 NV UP EI PL NZ NA PO NC
0B20:0100 0E          PUSH    CS
-a
0B20:0100 mov ax,1
0B20:0103 sub ax,1
0B20:0106
-t
AX=0001 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0B20 ES=0B20 SS=0B20 CS=0B20 IP=0103 NV UP EI PL NZ NA PO NC
0B20:0103 2D0100      SUB      AX,0001
-t
AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0B20 ES=0B20 SS=0B20 CS=0B20 IP=0106 NV UP EI PL ZR NA PE NC
0B20:0106 0E          PUSH    ES
```

```

-a
0B20:0106 mov ax,1
0B20:0109 and ax,0
0B20:010C
-
-t
AX=0001 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0B20 ES=0B20 SS=0B20 CS=0B20 IP=0109 NV UP EI PL ZR NA PE NC
0B20:0109 250000 AND AX,0000
-t
AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0B20 ES=0B20 SS=0B20 CS=0B20 IP=010C NV UP EI PL ZR NA PE NC
0B20:010C F2 REPNZ
0B20:010D AE SCASB
-a
0B20:010C mov ax,1
0B20:010F or ax,0
0B20:0112
-t
AX=0001 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0B20 ES=0B20 SS=0B20 CS=0B20 IP=010F NV UP EI PL ZR NA PE NC
0B20:010F 0D0000 OR AX,0000
-t
AX=0001 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0B20 ES=0B20 SS=0B20 CS=0B20 IP=0112 NV UP EI PL NZ NA PO NC
0B20:0112 FFBFB801 ??? [BX+01B8] DS:01B8=90BE
-

```

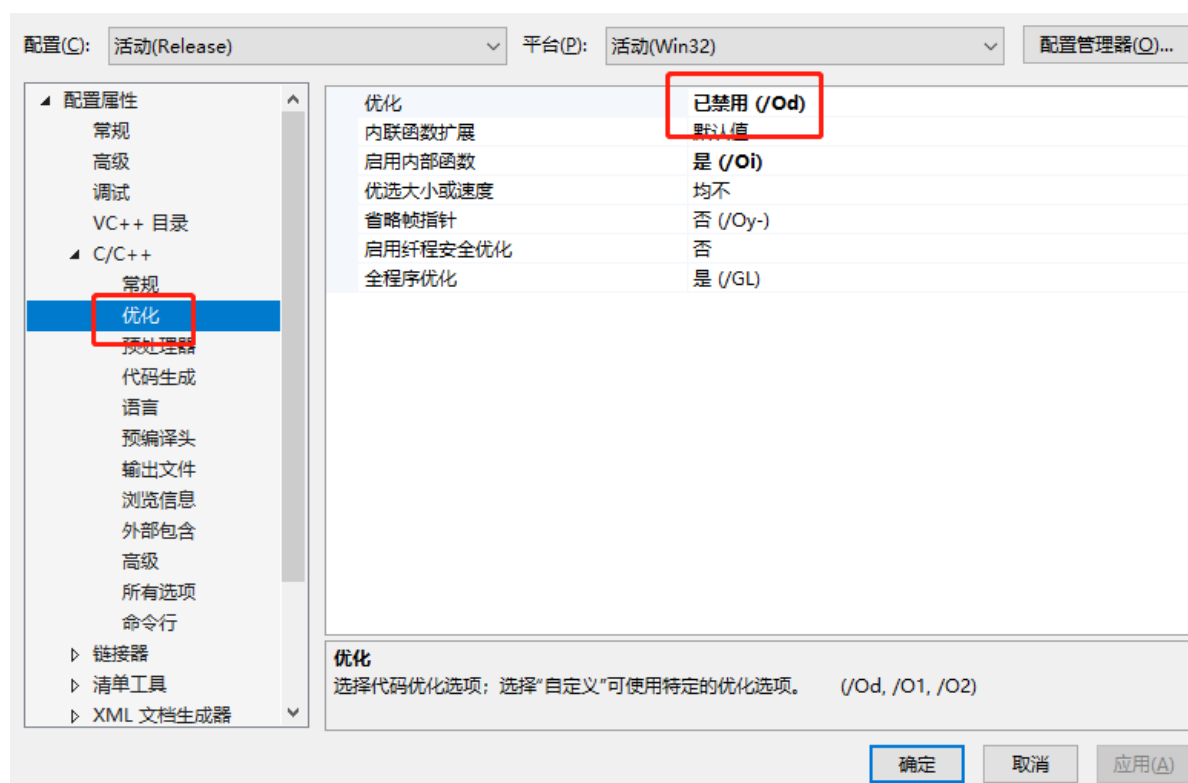
OD命令栏指令

- bp 下断点
- bc 清除断点
- dd 以双字显示数据
- dw 以字显示数据
- db 以字节显示数据
- dc 以字符显示数据
- ? 计算表达式的值

PDB文件

全称为 程序数据库 文件。我们使用它（更确切的说是看到它被应用）大多数场景是调试应用程序。目前我们对.PDB文件的普遍认知是它存储了被编译文件的调试信息，作为符号文件存在。

禁用VS优化



练习

将123B0H~123B9H的内存单元定义为数据段，累加这个数据段中的前3个单元中的数据。

```
mov ax,123BH
mov ds,ax          ;将 123BH 送入 ds 中，作为数据段的段地址
mov al,0           ;用 al 存放累加结果
add al,[0]         ;将数据段第一个单元(偏移地址为 0)中的数值加到 al 中
add al,[1]         ;将数据段第二个单元(偏移地址为 1)中的数值加到 al 中
add al,[2]         ;将数据段第三个单元(偏移地址为 2)中的数值加到 al 中
```

累加数据段中前3个字型数据。

```
mov ax,123BH
mov ds,ax          ;将 123BH 送入 ds 中，作为数据段的段地址
mov ax,0           ;用 ax 存放累加结果
add ax,[0]         ;将数据段第一个字(偏移地址为 0)加到 ax 中
add ax,[2]         ;将数据段第二个字(偏移地址为 2)加到 ax 中
add ax,[4]         ;将数据段第三个字(偏移地址为 4)加到 ax 中
```

