

$[bx+si+idata]$ 和  
 $[bx+di+idata]$

贺利坚 主讲



汇编语言程序设计  
Assembly Language

# $[bx+si+idata]$ 和 $[bx+di+idata]$ 方式指定地址

🖥️  $[bx+si+idata]$ 表示一个内存单元

📁 偏移地址为 $(bx)+(si)+idata$ ，即 $bx$ 中的数值加上 $si$ 中的数值再加上 $idata$

🖥️ 指令 $mov\ ax,[bx+si+idata]$ 的含义

📁 将一个内存单元的内容送入 $ax$

📁 这个内存单元的长度为2字节（字单元），存放一个字

📁 偏移地址为 $bx$ 中的数值加上 $si$ 中的数值再加上 $idata$ ，段地址在 $ds$ 中

🖥️ 数学化的描述

📁  $(ax) = (ds) * 16 + (bx) + (si) + idata$

🖥️ 指令 $mov\ ax,[bx+si+idata]$ 的其他写法

$mov\ ax,[bx+200+si]$

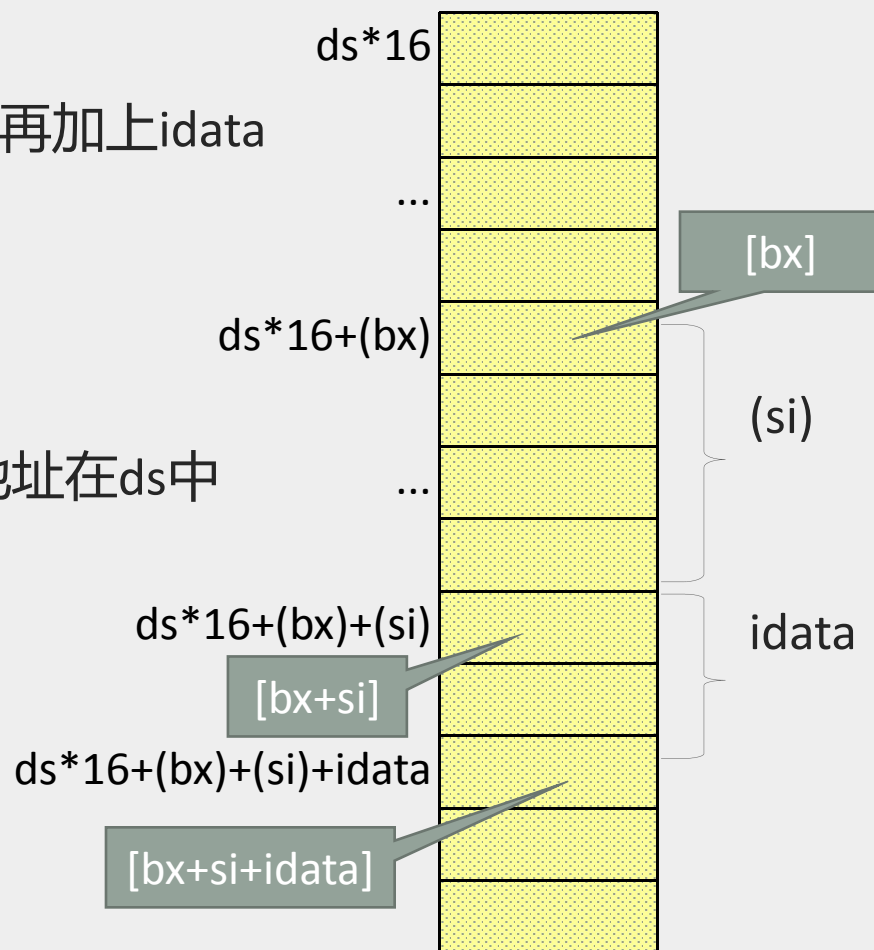
$mov\ ax,[bx].200[si]$

$mov\ ax,[200+bx+si]$

$mov\ ax,[bx][si].200$

$mov\ ax,200[bx][si]$

$mov\ ax,[bx][si]$



# 应用案例

💻 内存中数据：2000:1000 BE 00 06 00 6A 22 .....

💻 程序执行后，ax、bx、cx中的内容？

DS		SI		DI	
AX		BX		CX	

```
mov ax,2000H
mov ds,ax
mov bx,1000H
mov si,0
mov ax,[bx+2+si]
inc si
mov cx,[bx+2+si]
inc si
mov di,si
mov ax,[bx+2+di]
```

```
-a 073f:0100
073F:0100 mov ax, 2000
073F:0103 mov ds, ax
073F:0105 mov bx, 1000
073F:0108 mov si, 0
073F:010B mov ax, [bx+2+si]
073F:010E inc si
073F:010F mov cx, [bx+2+si]
073F:0112 inc si
073F:0113 mov di, si
073F:0115 mov ax, [bx+2+di]
073F:0118
-
-e 2000:1000 BE 00 06 00 6A 22
-g 010E
AX=0006 BX=1000 CX=0600 DX=0000 SP=00FD BP=0000 SI=0000 DI=0002
DS=2000 ES=073F SS=073F CS=073F IP=010E  NV UP EI PL NZ NA PO NC
073F:010E 46          INC     SI
-g 0112
AX=0006 BX=1000 CX=6A00 DX=0000 SP=00FD BP=0000 SI=0001 DI=0002
DS=2000 ES=073F SS=073F CS=073F IP=0112  NV UP EI PL NZ NA PO NC
073F:0112 46          INC     SI
-g 0118
AX=226A BX=1000 CX=6A00 DX=0000 SP=00FD BP=0000 SI=0002 DI=0002
DS=2000 ES=073F SS=073F CS=073F IP=0118  NV UP EI PL NZ NA PO NC
073F:0118 0000      ADD     [BX+SI],AL          DS:1002=06
-
```