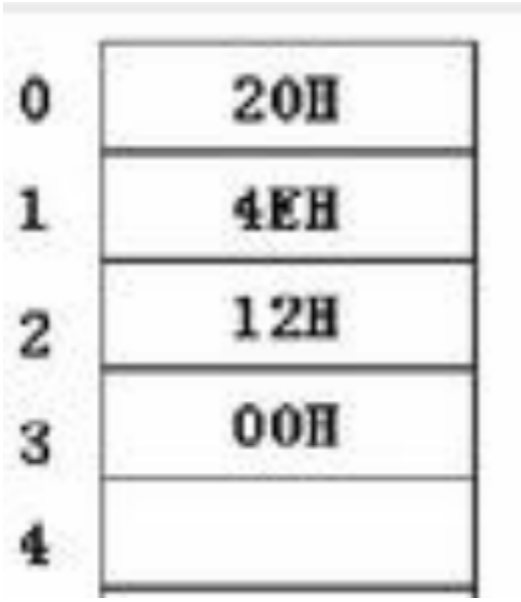
第三章 寄存器之内存访问

内存中字的储存

在CPU中，使用16位来储存一个字。其中**高8位存放高位字节，低8位存放低位字节。**

例如在0地址开始存放20000(4E20H)



0地址是低位地址，存放4E20H的低八位，也就是20H。

1地址是高位地址，存放4E@)H的高八位，也就是4EH。

0地址字单元存放的数据为 4E20H

1地址字单元存放的数据为 124EH

2地址字单元存放的数据为 0012H

DS 和 [address]

DS寄存器是一个**段寄存器**。

CPU在读写数据的时候必须要知道数据的地址，这个地址是通过**段地址：偏移地址**的形式给出的。8086CPU中的段寄存器`DS`一般用来**存放要访问数据的内存的段地址**。

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

这里的`mov al, [0]`中的0,表示的是偏移地址，其段地址在`dx`寄存器中存放。

注意：DS寄存器是段寄存器，我们不可以直接使用`mov ds, idata`（idata是一个常量，如1000H）这种形式为dx赋值，可以使用`mov dx, ax 这种形式。

上面的`mov al, [0]`是将'1000H:0H'字节单元中数据传送至al寄存器中，这种操作是可以反向的，也可以将寄存器中的内容传送至某一地址处，如。

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

mov， add， sub指令

mov指令

传送指令，可以有一下的几种形式。

形式	例子
mov reg, idata	mov ax, 8
mov reg, reg	mov ax, bx
mov reg, mem	mov ax, [0]
mov mem, reg	mov [0], ax
mov sreg, reg	mov ds, ax
mov reg, sreg	mov ax, ds
mov mem, sreg	mov [0], ds
mov sreg, mem	mov ds, [0]

其中，idata表示数据，reg表示寄存器，mem表示内存单元，sreg表示段寄存器。

add, sub指令

形式	例子
add reg, idata	add ax, 8
add reg, reg	add ax, bx
add reg, mem	add ax, [0]
add mem, reg	add [0], ax
sub reg, idata	sub ax, 8
sub reg, reg	sub ax, bx
sub reg, mem	sub ax, [0]
sub mem, reg	sub [0], ax

注意：段寄存器不可以参与算术运算。

数据段

将一组内存单元定义为数据段，一般使用`ds`做该数据段的段地址，跟据指令访问相应的单元。

```
mov ax, 123BH
mov ds, ax
mov al, 0
mov al, [0]
mov al, [1]
mov al, [2]
...
```

此时处理的是字节类型的数据，如果我们处理的是字型的数据，需要每次加2,如。

```
mov ax, 123BH
mov ds, ax
mov al, 0
mov al, [0]
mov al, [2]
mov al, [4]
```

栈

8086CPU提供入栈和出栈的指令，最基本的两个是`PUSH`和`POP`。

8086CPU的出栈和入栈操作都是以字为单位的，如push ax合法，push al非法。

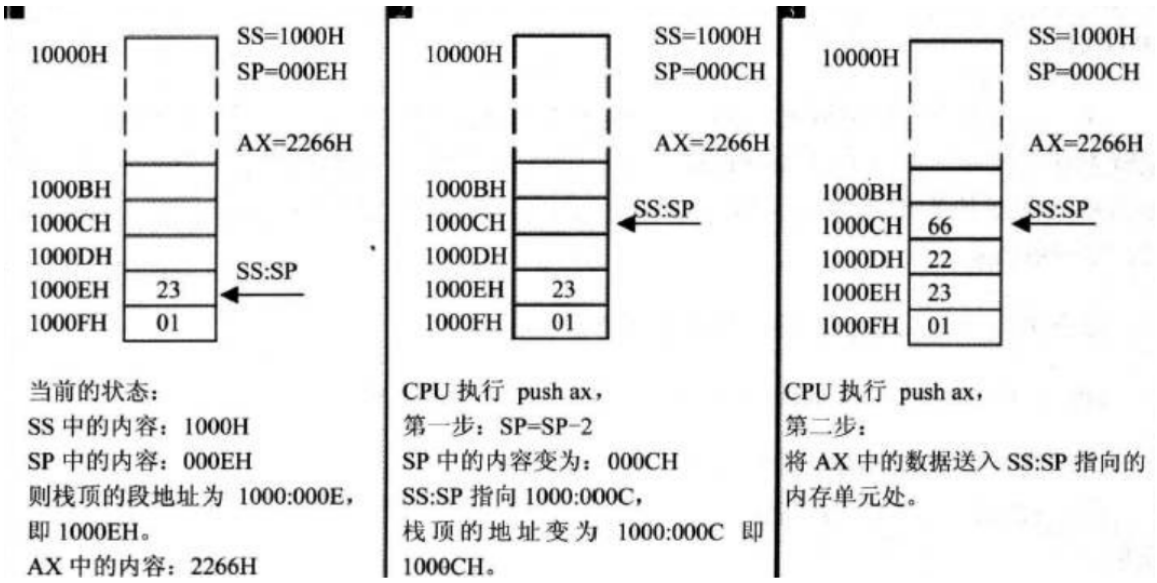
与栈有关的寄存器是**SS和SP**。

在任何时刻，8086CPU将`SS:SP`指向栈顶的元素。

PUSH的步骤

以`push ax`为例子。

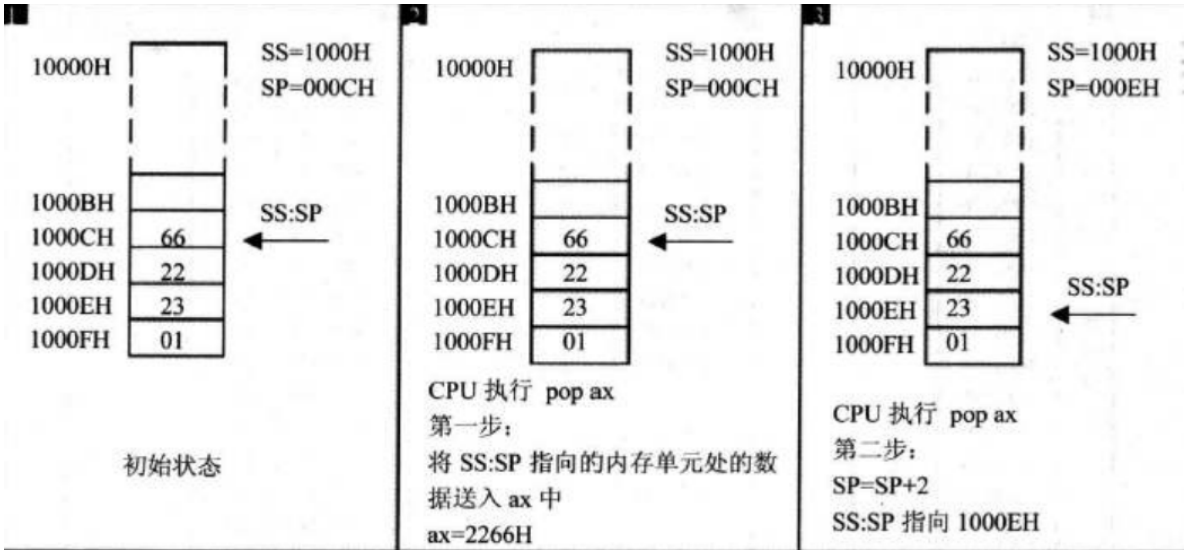
- `SP -= 2`，栈顶向低位移动一个字，以当前元素的前面的单元作为新的栈顶。
- 将ax中的内容送入到`SS:SP`指向的内容单元处，`SS:SP`此时指向新栈顶。



POP的步骤

以`pop ax`为例子

- 将`SS:SP`处的字单元数据送入到ax中
- `SP += 2`，栈顶向高位移动一个字，以当前栈顶的下面的单元为新的栈顶。



PUSH, POP指令的格式

push/pop指令有如下的格式。

形式	例子
push/pop reg	push/pop ax
push/pop sreg	push/pop ds
push/pop mem	push/pop [2]

借助栈对ax，bx寄存器中的内容进行交换。

```
mov ax, 1000H
mov ss, ax ;和ds寄存器一样，不能直接赋值，要借助其他寄存器
mov sp, 0010H ;设置好了栈顶的位置 1000H:0010H

mov ax, 000AH
mov bx, 000BH

push ax
push bx

pop ax ; bx -> ax
pop bx ; ax -> bx
```

栈段

和上面说的数据段相似，我们可以将一段内存单元作为栈来使用，使用`SS`作为段地址。

和数据段一样，栈段最大为64KB，此时`SP = 0H`