目 录

1 通用寄存器

8086CPU 中有 AX、BX、CX 和 DX 四个寄存器，用于存放一般性的数据。

通过 mov 指令可以修改 AX、BX、CX 和 DX 的值，例子如下：

```
1    mov ax,123
2    mov bx,123
3    mov cx,123
4    mov dx,123
```

2 8086CPU 给出物理地址的方法

8086CPU 有 20 位地址总线，可是 8086CPU 是 16 位结构，所以地址加法器采用物理地址 = 段地址 x16+ 偏移地址的方法将段地址和偏移地址合成物理地址。

需要注意的是，虽然“段地址”这个名称中包含段，但是内存没有分段，只是 CPU 用分段的方式来管理内存。

3 段寄存器

8086CPU 中有 CS、DS、SS 和 ES 四个寄存器，用于存放段地址。

不同于通用寄存器，8086CPU 不支持将数据直接送入段寄存器。可以将数据先送入通用寄存器，然后再将通用寄存器的内容送入段寄存器。也可以将内存单元中的数据送入段寄存器。

3.1 CS 和 IP

CS 是代码段寄存器，IP 为指令指针寄存器。当 CS 中的内容为 M，IP 中的内容为 N，8086CPU 将从内存 M×16+N 开始执行指令。

8086CPU 读取指令的步骤如下：

1. 从 CS:IP 指向的内存单元读取指令，读取的指令进入指令缓冲器。
2. IP=IP+ 所读取指令的长度，从而指向下一条指令。
3. 执行指令，转到步骤 1，重复这个过程。

3.2 修改 CS 和 IP 的值

mov 指令不能用于修改 CS、IP 的值，只有转移指令可以修改 CS、IP 的内容，比如说 jmp 指令。

如果想同时修改 CS、IP 的内容，可以用指令“jmp 段地址: 偏移地址”完成，如下例所示：

```
; 执行后，CS=2AE3H，IP=0003H，CPU从2AE33H处读取指令
jmp 2AE3:3
```

如果只想修改 IP 的内容，可以用指令“jmp 某一合法寄存器”，如下例所示：

```
; jmp指令用寄存器中的值修改IP，CS的内容不变
jmp ax
jmp bx
```

3.3 代码段

在编程时，可以根据需要将一组内存单元定义为一个段。代码段就是用于存放代码的一组内存单元。

需要知道的是，虽然我们编程时安排了代码段，但 CPU 不会由于这种安排自动地执行代码段中的指令，CPU 只认可 CS:IP 指向的内存单元中的内容为指令。所以要让 CPU 执行我们放在代码段中的指令，必须讲 CS:IP 指向所定义的代码段中的第一条指令的首地址。

4 DS 和 [address]

DS 寄存器用于访问数据的段地址，而 [address] 中的 address 是偏移地址，例子如下：

```
; 将1000:0处的数据读入al中
mov bx,1000H
mov ds,bx
mov al,[0]
```

mov 指令支持将一个内存单元中的内容送入一个寄存器中，指令格式为：

```
mov 寄存器名,内存单元地址
```

其中内存单元地址就是用 ds 和“[...]”表示，ds 表示内存单元的段地址，[...] 表示内存单元的偏移地址。

5 mov、add、sub 指令

mov 指令有如下几种形式：

```
mov 寄存器, 数据
mov 寄存器, 寄存器
mov 寄存器, 内存单元
mov 内存单元, 寄存器
mov 段寄存器, 寄存器
mov 寄存器, 段寄存器
mov 段寄存器, 内存单元
mov 内存单元, 段寄存器
```

add 指令有如下几种形式：

```
add 寄存器, 数据
add 寄存器, 寄存器
add 寄存器, 内存单元
add 内存单元, 寄存器
```

sub 指令有如下几种形式：

```
sub 寄存器, 数据
sub 寄存器, 寄存器
sub 寄存器, 内存单元
sub 内存单元, 寄存器
```

6 数据段

数据段就是用于存储数据的一组内存单元。在访问数据段中的数据时，只要用 ds 存放数据段的段地址，然后根据需要，用相关指令访问数据段中的具体单元。

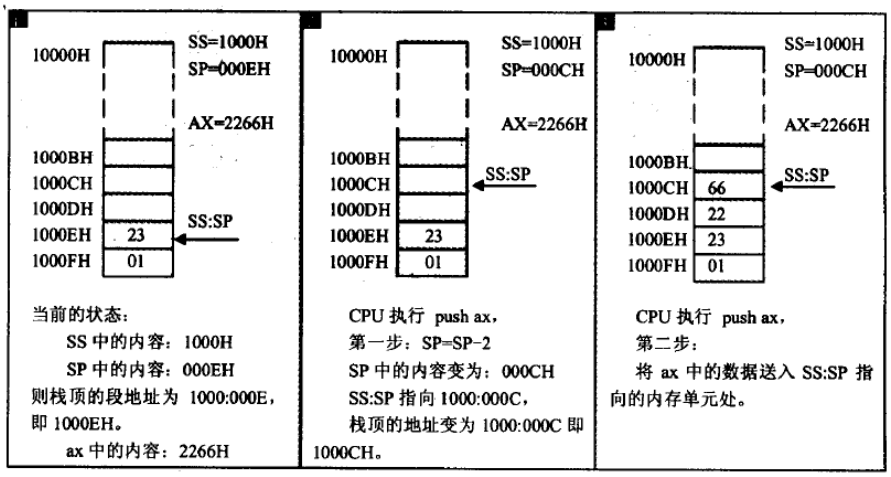
7 CPU 提供的栈机制

8086CPU 提供入栈和出栈指令，也就是 push 和 pop。

7.1 SS 和 SP 寄存器

8086CPU 提供了段寄存器 SS 和寄存器 SP，SS 用于存放栈顶的段地址，SP 用于存放偏移地址，SS:SP 指向栈顶元素。

8086CPU 对 push 指令的执行如下图所示：



可以看到，入栈时，栈顶从高地址向低地址增长，不过段基址是低地址，而 SP 初始值是栈的长度。

需要注意的是，8086CPU 不能保证我们对栈的操作不会越界，我们在编程的时候需要自己防止栈顶越界的问题。

7.2 push、pop 指令

push 的形式如下：

```
push 寄存器
push 段寄存器
push 内存单元
```

pop 的形式如下：

```
pop 寄存器
pop 段寄存器
pop 内存单元
```

7.3 栈段

我们可以将一组内存空间当作栈来使用，以栈的方式进行访问，那么这段空间就可以称为一个栈。

可以用 SS 存放栈段的基地址，将 SP 的初始值设为栈段的长度，就可以将 SS:SP 指向我们定义的栈段。