目 录

1 以字符形式给出的数据 2
1.1 例程 2
1.2 分析例程 2

2 [bx+idata] 2

3 SI 和 DI 2
3.1 [bx+si] 和 [bx+di] 3
3.2 [bx+si+idata] 和 [bx+di+idata] 3

1 以字符形式给出的数据

1.1 例程

首先看一段例程，如下所示：

```
assume ds:data

data segment
    db 'unIX'
    db 'foRX'
data ends

code segment
start:
    mov al, 'a'
    mov bl, 'b'
    mov ax, 4c00h
    int 21h
code ends

end
```

1.2 分析例程

在上述例子中数据以字符的形式给出，编译器会把它们转化为相对应的 ASCII 码。“db ‘unIX’”相当于“db 75H,6EH,49H,58H”，“mov al, ‘a’”相当于“mov al, 61H”。

2 [bx+idata]

我们已经知道可以用 [bx] 来指明一个内存单元，其实还可以用 [bx+idata] 这种更灵活的方式来指明内存单元，其中 idata 是常数。该指令还有等价的形式，如下所示：

```
mov ax, [idata+bx]
mov ax, idata[bx]
mov ax, [bx].idata
```

3 SI 和 DI

SI 和 DI 是 8086CPU 中和 bx 功能相近的寄存器。下面的三组指令实现了相同的功能：

```
mob bx, 0
mov ax, [bx]
```

```
3
4       mov si, 0
5       mov ax, [si]
6
7       mov di, 0
8       mov ax, [di]
```

下面三组指令也实现了相同的功能：

```
1       mov bx, 0
2       mov ax, [bx+123]
3
4       mov si, 0
5       mov ax, [si+123]
6
7       mov di, 0
8       mov ax, [di+123]
```

3.1 [bx+si] 和 [bx+di]

“mov ax, [bx+si]”的含义是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，段地址在 ds 中，偏移地址为 bx 中的数值加上 si 中的数值。“mov ax, [bx+di]”的功能类似。这两条指令也可以写成如下形式：

```
1       mov ax, [bx][si]
2       mov ax, [bx][di]
```

3.2 [bx+si+idata] 和 [bx+di+idata]

[bx+si+idata] 表示一个内存单元，它的偏移地址为 (bx)+(si)+idata。[bx+di+idata] 含义相似。这两条指令还可以写成如下形式：

```
1       [idata+bx+si]
2       [bx+idata+si]
3       idata[bx][si]
4       [bx].idata[si]
5       [bx][si].idata
6
7       [idata+bx+di]
8       [bx+idata+di]
9       idata[bx][di]
10      [bx].idata[di]
11      [bx][di].idata
```