

目 录

1	[bx]	2
2	loop 指令	2
3	loop 和 [bx] 的联合应用	2
4	段前缀	3
5	段前缀的使用	3

1 [bx]

[bx] 类似于 [address], 表示一个内存单元, 它的偏移地址存放在 bx 中。[bx] 和 [address] 的用法一样, 如下所示:

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

2 loop 指令

loop 指令也就是循环指令, 当 CPU 执行 loop 指令的时候, 会执行以下两个步骤:

1. (cx)=(cx)-1
2. 判断 cx 中的值, 不为零则转至标号处执行程序, 如果为零则向下执行

使用 loop 指令的例子如下所示:

```
1  assume cs:code
2  code segment
3      mov ax, 2
4      mov cx, 11
5  s:
6      add ax, ax
7      loop s
8
9      mov ax, 4c00h
10     int 21h
11 code ends
```

3 loop 和 [bx] 的联合应用

计算 ffff:0 ~ ffff:b 单元中的数据的和, 结果存储在 dx 中, 实现代码如下:

```
1  assume cs:code
2  code segment
3      mov ax, 0ffffh
4      mov ds, ax
5      mov bx, 0
6
7      mov dx, 0
8
9      mov cx, 12
10
11 s:
```

```
12      mov al, [bx]
13      mov ah, 0
14      add dx, ax
15      inc bx
16      loop s
17
18      mov ax, 4c00h
19      int 21h
20  code ends
21  end
```

4 段前缀

访问内存单元时，段地址默认在 ds 中，不过我们也可以显式地给出内存单元的段地址所在的段寄存器。这种方式称为段前缀，如下例所示：

```
1      mov ax, ds:[bx]
2      mov ax, cs:[bx]
3      mov ax, ss:[bx]
4      mov ax, es:[bx]
5
6      mov ax, ds:[0]
7      mov ax, cs:[0]
8      mov ax, ss:[0]
9      mov ax, es:[0]
```

5 段前缀的使用

将内存 ffff:0 ~ ffff:b 单元中的数据拷贝到 0:200 ~ 0:20b 单元中，实现代码如下：

```
1      assume cs:code
2      code segment
3          mov bx, 0
4          mov cx, 12
5
6          mov ax, 0ffffh
7          mov ds, ax
8          mov ax, 0020h
9          mov es, ax
10
11     s:
12         mov al, ds:[bx]
13         mov es:[bx], al
14         inc bx
15         loop s
16
17     mov ax, 4c00h
18     int 21h
```

```
19  code ends
20  end
```