

目 录

| | | |
|----------|-------------------|----------|
| 1 | 8086 汇编指令集 | 4 |
| 1.1 | aaa | 4 |
| 1.2 | adc | 4 |
| 1.3 | add | 4 |
| 1.4 | and | 4 |
| 1.5 | call | 4 |
| 1.6 | clc | 4 |
| 1.7 | cld | 4 |
| 1.8 | cli | 4 |
| 1.9 | cmc | 4 |
| 1.10 | cmp | 4 |
| 1.11 | cmps | 4 |
| 1.12 | dec | 4 |
| 1.13 | div | 4 |
| 1.14 | dup | 5 |
| 1.15 | esc | 6 |
| 1.16 | hlt | 6 |
| 1.17 | idiv | 6 |
| 1.18 | imul | 6 |
| 1.19 | inc | 6 |
| 1.20 | int | 6 |
| 1.21 | iret | 6 |
| 1.22 | ja | 6 |
| 1.23 | jb | 6 |
| 1.24 | jcxz | 6 |
| 1.25 | je | 6 |
| 1.26 | jmp | 6 |
| 1.27 | jna | 6 |
| 1.28 | jnb | 6 |
| 1.29 | jne | 6 |

| | |
|----------------------|----|
| 1.30 lock | 6 |
| 1.31 lods | 6 |
| 1.32 lodsb | 7 |
| 1.33 loop | 7 |
| 1.34 mov | 7 |
| 1.35 movsb | 7 |
| 1.36 movsw | 7 |
| 1.37 mul | 7 |
| 1.38 nop | 9 |
| 1.39 not | 9 |
| 1.40 or | 9 |
| 1.41 push | 9 |
| 1.42 pop | 9 |
| 1.43 pushf | 9 |
| 1.44 rcl | 9 |
| 1.45 rcr | 9 |
| 1.46 rep | 9 |
| 1.47 repe | 9 |
| 1.48 repne | 9 |
| 1.49 ret | 9 |
| 1.50 retf | 9 |
| 1.51 rol | 9 |
| 1.52 ror | 9 |
| 1.53 sal | 9 |
| 1.54 sar | 9 |
| 1.55 sbb | 9 |
| 1.56 scas | 9 |
| 1.57 shl | 9 |
| 1.58 shr | 9 |
| 1.59 stc | 9 |
| 1.60 std | 9 |
| 1.61 sti | 10 |
| 1.62 stos | 10 |
| 1.63 sub | 10 |

| | | |
|------|----------------|----|
| 1.64 | test | 10 |
| 1.65 | wait | 10 |
| 1.66 | xchg | 10 |
| 1.67 | xor | 10 |

1 8086 汇编指令集

1.1 aaa

1.2 adc

1.3 add

1.4 and

add 指令有如下几种形式：

```
1 add 寄存器, 数据
2 add 寄存器, 寄存器
3 add 寄存器, 内存单元
4 add 内存单元, 寄存器
```

1.5 call

1.6 cld

1.7 cld

cld 指令的功能是将 DF 标志置 0。

DF 标志是方向标志位。当 DF=0 时，串处理指令操作后 si, di 递增。当 DF=1 时，串处理指令操作后 si, di 递减。

1.8 cli

1.9 cmc

1.10 cmp

cmp 指令是比较指令，指令格式为：“cmp 操作对象 1, 操作对象 2”，cmp 的功能相当于减法指令，只是不保存结果，然后根据计算结果对标志寄存器进行设置。

1.11 cmps

1.12 dec

1.13 div

div 指令是除法指令，对它的介绍如下：

1. 除数：有 8 位和 16 位两种，在一个寄存器或内存单元中。
2. 被除数：如果除数为 8 位，那么被除数为 16 位，默认放在 AX 中。如果除数为 16 位，那么被除数为 32 位，默认在 DX 和 AX 中存放，DX 存放高 16 位，AX 存放低 16 位。
3. 结果：如果除数为 8 位，那么 AL 存储商，AH 存储余数。如果除数为 16 位，那么 AX 存储商，DX 存储余数。
4. div 指令的格式如下：

```
1  div reg
2  div 内存单元
```

可以看出，div 指令的操作对象不可以是段寄存器 sreg，只能是寄存器 reg。

1.14 dup

dup 是一个操作符，和 db、dw、dd 这些指令配合使用。dup 的使用格式如下：

- db 重复的次数 dup (重复的字节型数据)
- dw 重复的次数 dup (重复的字型数据)
- dd 重复的次数 dup (重复的双字型数据)

dup 的使用如下例所示：

```
1  db 3 dup (0)
2  db 3 dup (0, 1, 2)
3  db 3 dup ('abc', 'ABC')
```

1.15 esc

1.16 hlt

1.17 idiv

1.18 imul

1.19 inc

1.20 int

1.21 iret

1.22 ja

1.23 jb

1.24 jcxz

1.25 je

1.26 jmp

1.27 jna

1.28 jnb

1.29 jne

1.30 lock

1.31 lods

串传送指令 lods，功能为：

1 $(ax) = ((ds) * 16 + (si))$

1.32 lodsb

串传送指令 lodsb，功能为：

```
1  (al) = ((ds) * 16 + (si))
2  (si) = (si) + 1  如果 DF = 0
3  (si) = (si) - 1  如果 DF = 1
```

1.33 loop

1.34 mov

mov 指令有如下几种形式：

```
1  mov 寄存器, 数据
2  mov 寄存器, 寄存器
3  mov 寄存器, 内存单元
4  mov 内存单元, 寄存器
5  mov 段寄存器, 寄存器
6  mov 寄存器, 段寄存器
7  mov 段寄存器, 内存单元
8  mov 内存单元, 段寄存器
```

1.35 movsb

1.36 movsw

1.37 mul

mul 指令是乘法指令，乘法指令的介绍如下：

- 两个相乘的数：如果两个相乘的数都是 8 位，那么一个默认放在 AH 中，另一个放在 8 位寄存器或内存字节单元中。如果两个都是 16 位，那么一个默认放在 AX 中，另一个放在 16 位寄存器或内存字单元中。
- 结果：如果是 8 位乘法，结果默认放在 AX 中。如果是 16 位乘法，结果的高位默认放在 DX 中，低位放在 AX 中。

mul 指令的格式如下：

```
1  mul reg
2  mul 内存单元
```


1.38 nop

1.39 not

1.40 or

1.41 push

1.42 pop

1.43 pushf

1.44 rcl

1.45 rcr

1.46 rep

1.47 repe

1.48 repne

1.49 ret

1.50 retf

1.51 rol

1.52 ror

1.53 sal

1.54 sar

1.55 sbb

1.56 scas

1.57 shl

1.58 shr

1.59 stc

1.60 std

9

std 指令的功能是将 DF 标志置 1。

DF 标志是方向标志位。当 DF=0 时，串处理指令操作后 si, di 递增。当 DF=1 时，串处理指令操作后 si, di 递减。

1.61 sti

1.62 stos

1.63 sub

sub 指令有如下几种形式：

| | |
|---|---------------|
| 1 | sub 寄存器, 数据 |
| 2 | sub 寄存器, 寄存器 |
| 3 | sub 寄存器, 内存单元 |
| 4 | sub 内存单元, 寄存器 |

1.64 test

1.65 wait

1.66 xchg

1.67 xor