链滴

# 汇编语言 学习笔记（一）

作者：houjie
原文链接：https://ld246.com/article/1510123669944
来源网站：链滴
许可协议：署名-相同方式共享 4.0 国际 (CC BY-SA 4.0)

# 汇编语言学习笔记

# 一、概论

8086,8088 构架

## 1. 寄存器

### 8个通用寄存器

- AX [AH, AL] - 累加器
- BX [BH, BL] - 基址寄存器
- CX [CH, CL] - 计数器
- DX [DH, DL] - 数据寄存器
- SP - 堆栈指针
- BP - 基址指针
- SI - 源地址寄存器
- DI - 目的地址寄存器

### 2个控制寄存器

- IP - 指令指针 【 **用户不能直接访问IP**】
- F - 状态标志寄存器 (溢出、方向、符号、进位、零、奇偶...)

### 4个段寄存器

- CS - 代码段寄存器
- DS - 数据段寄存器
- SS - 堆栈段寄存器
- ES - 附加段寄存器

## 2. 存储器

数据长度

| 类型名称 | 定义符号 | 类型值 |
| --- | --- | --- |
| 字节变量 | DB | 1 |
| 字变量 | DW | 2 |
| 双字变量 | DD | 4 |
| 长节变量 | DQ | 8 |
| 10字节变量 | DT | 10 |
| 近标号 | NEAR | -1(FFH) |

| | | |
|---|---|---|
| 远标号 | FAR | -2(FEH) |

far是远调用；near是近调用。

near只能在段内被调用，所以调用是只需将ip进栈；而far则可以在段间被调用，所以需要cs与ip都进。

jmp short 标号（转移标号处执行指令） / jmp near ptr 标号 功能：(IP) = (IP)+16位位移。 实现是段内近转移。

jmp far ptr 标号 实现的是段间转移（远转移）

## 数制

- 2进制 - B
- 10进制 - D (BCD码)
- 16进制 - H

## 字符

- ASC II 码

# 二、寻址方式

() - 表示地址内的内容

[] - 直接地址

## 与数据有关的寻址方式

### 1. 立即数寻址

```
MOV ax, 13          ; (ax) ← 13
```

a. 立即数可以使八位或者十六位

b. 目的寄存器可以是8或者16位的 **通用寄存器** 或者 **段寄存器**

### 2. 寄存器寻址

```
MOV ax, bx          ; (ax) ← (bx)
```

### 3. 存储器寻址

#### 1. 直接寻址 （有效地址 = 基址 + 偏移地址）

```
MOV  ax, [2000H]       ; 将存储器 [ds:2000H] 的内容赋值到 ax
MOV  ax, es:[2000H]    ; 将存储器 [es:2000H] 的内容赋值到 ax
```

*a. * **DS** 隐含指定

### 2. 寄存器间接寻址 （有效地址 = 段寄存器地址x16 + 寄存器内容）

```
MOV ax, [bx]          ; 将存储器 [(ds)*16 + (bx)] 对应的存储单元的内容送到 ax中
MOV ax, [ss]          ; 将存储器 [(bp)*16 + (ss)] 对应的存储单元的内容送到 ax中
```

a. bx,si,di 所对应的段寄存器为 ds **（数据段 ）**

b. bp 所对应的段寄存器为 ss **（堆栈段 ss - 寄存器 bp）**

c. 物理地址 = (段寄存器) x 16 + (寄存器) **即段寄存器后面加一个0,再加上制定寄存器的值为地址**

### 3. 寄存器相对寻址 （有效地址 = 段寄存器地址x16 + 寄存器内容 + 8/16 位移量）

```
MOV ax, count[si] ;将存储器[(ds)*16 + count + (si)]对应的存储单元的内容送到 ax 中
MOV ax, count[bp] ;将存储器[(ss)*16 + count + (si)]对应的存储单元的内容送到 ax 中
```

a. bx,si,di 所对应的段寄存器为 ds **（数据段 ）**

b. bp 所对应的段寄存器为 ss **（堆栈段 ss - 寄存器 bp）**

### 4. 基址变址寻址 （有效地址 = 段寄存器地址x16 + 基址寄存器 + 变址寄存器）

```
MOV  ax, [bx + di]   ; (ax) ← mem[(ds)*16 + (bx) + (di)]
MOV  ax, [bp][si]    ; (ax) ← mem[(ds)*16 + (bp) + (si)]
```

a. 基址寄存器 **bx，bp**

b. 变址寄存器 **si，di**

### 5. 相对基址变址寻址 （有效地址 = 段寄存器*16 + 基址寄存器 + 编制寄存器 + 位量）

```
; (ax) ← mem[mk + (bx) + (si)]
MOV AX, MK[BX][SI]
MOV AX, MK[BX+SI]
MOV AX, [MK+BX+SI]
```

*a. * **DS** 隐含指定

## 转移地址寻址

### 1. 段内直接转移

#### 1. JMP　SHORT OPR

```
JMP   SHORT OPR     ; (IP) ← (IP) + 8位位移量
```

a. 8位和16位位移量由 **目标地址OPR** *确定 *

### 2. JMP　NEAR PTR OPR

JMP NEAR PTR OPR ; (IP) ← (IP) + 16位位移量

## 2. 段内间接转移

### 1. JMP WORD PTR OPR

JMP ax ; (IP) ← (ax)
JMP WORD PTR TABLE[bx] ; (IP) ← (mem[(ds)*16 + TABLE + (bx)])

a. 有效地址EA值由OPR的寻址方式确定，可以使用mem或reg

## 3. 段间直接转移

### 1. JMP FAR PTR OPR

; (IP) ← OPR的段内偏移地址
; (CS) ← OPR所在段的段地址
JMP FAR PTR NEXT ; NEXT 与 当前指令不在一个段内

## 4. 段内间接转移

### 1. JMP DWORD PTR OPR

;(IP) ← (EA)
;(CS) ← (EA+2)
JMP DWORD PTR OPR

a. EA由OPR的寻址方式为存储器寻址方式