# 汇编语言基本概念知识汇总

1、8086/8088 CPU 的寄存器组中，8 位的寄存器有 8 个：
AL AH BL BH CL CH DL DH。 P21

2、8086/8088 CPU 的寄存器组中，16 位的寄存器有 14 个：
AX BX CX DX DS CS SS ES FS GS DI SI BP SP。P21

3、当使用 BP 作编程地址时，此时使用的是 SS 段。P21

4、影响标志位：
①、SAHF,POPF 根据计算情况影响标志位。P58
②、ADD,ADC,XADD,SUB,SBBCMP,NEG 等影响条件标志位。
ADD 指令对 OF 标志位影响：若两操作数相同且结果符号与之相反，则 OF=1，反之 OF=0。 P59
OF 表示有符号溢出，CF 表述无符号溢出。
③、乘法指令对除 CF,OF 以外的条件码位无定义。
④、AND,OR,XOR,TEST 将使 CF 和 OF 位为 0。
⑤、移位指令会影响 CF,OF，具体见书 73 页倒数第一段。

5、不影响标志位：
①、MOV 指令不影响标志位。
②、PUSH 和 POP 指令不影响标志位。
③、INC,DEC 不影响 CF 位。
④、除法指令对条件码位均无定义。
⑤、NOT 指令不影响标志位。
⑥、所有的跳转指令都不影响条件码。
⑦、循环指令不影响条件码。
⑧、CALL 和 RET 指令都不影响条件码。

6、双操作数指令的两个操作数中除源操作数为立即数的情况外，必须有一个操作数在寄存器中。单操作数指令不允许使用立即数方式。
我们也可以理解为双操作数不允许有两个[]。

7、CS 指令不允许使用 MOV 指令。

8、DX 不能作为间接寻址的寄存器。

9、有效地址=（BX/BP）+（SI/DI）。即二选一，不能同时存在。
基址寄存器 变址寄存器

10、如 JMP 指令采用段间间接寻址，那么由 4 个相邻字节单元中存放有转移地址，其中前两个字节存放的是 IP，后两个字节存放的是 CS。
段间转移时进行入栈操作时先存放 CS 再存放 IP，并且 SP 每次减 2。

11、Loop 指令执行原则为先执行循环体 CX 再减一。

12、SP=返回的字节+IP 的 2 个字节（由于是段内返回）。

13、寄存器操作数：存放在 4 个通用寄存器（AX、BX、CX、DX 8 位或 16 位）、 4 个专用寄存器（SI、DI、BP、SP 只能是 16 位）或 4 个段寄存器（CS、DS、ES、SS 只能是 16 位）中的操作数。

14、不允许将立即数传送到段寄存器。

15、一般不允许两个操作数同时为存储器操作数。

16、8086 的寻址方式
①、立即数寻址：操作数直接包含在指令中

例：MOV AX，3102H

立即数只能是 8 或 16 位整数，通常用于双操作数指令中，只能作为源操作数。

②、寄存器寻址：操作数包含在寄存器中

例：MOV DX，AX

对 16 位操作数，寄存器可以为 AX、BX、CX、DX、SI、DI、SP、BP、CS、DS、SS、ES

对 8 位操作数，寄存器可以为 AH、AL、BH、BL、CH、CL、DH、DL

可以对源操作数、目标操作数或两者都采用寄存器寻址方式

当指令中的源操作数和目标操作数均为寄存器时，必须采用同样长度的寄存器

两个操作数不能同时为段寄存器

目标操作数不能是代码段寄存器 CS

③、直接寻址：有效地址在指令中直接给出

例：MOV AX, [2000H]

EA = 基址 + 变址 + 偏移量

默认段寄存器为 DS，若操作数在其它段，则必须使用段超越前缀指明

④、寄存器间接寻址：有效地址 = 寄存器内容

例：MOV AX, [BX]

EA = 基址 + 变址 + 偏移量

只有 SI、DI、BX（段基址默认在 DS 中）和 BP（段基址默认在 SS 中）可做间址寄存器，段基址可使用段超越前缀更改

⑤、寄存器相对寻址（直接变址寻址）：有效地址 = 寄存器内容 + 偏移量

例：MOV AX,[BX+1000H]

⑥、基址变址寻址：有效地址 = 基址寄存器内容+变址寄存器内容

例：MOV AX, [BX+SI]

MOV AX, [BP][DI]

$$EA = \begin{Bmatrix} (BX) \\ (BP) \end{Bmatrix} + \begin{Bmatrix} (SI) \\ (DI) \end{Bmatrix}$$

**BX:操作数的段基址默认在DS中**
**BP:操作数的段基址默认在SS中**

⑦、相对基址变址寻址：有效地址 = 基址寄存器内容 + 变址寄存器内容 + 偏移量

例：MOV AX，BASE [SI] [BX]

MOV AX，[BX+BASE] [SI]

$$EA = \begin{Bmatrix} (BX) \\ (BP) \end{Bmatrix} + \begin{Bmatrix} (SI) \\ (DI) \end{Bmatrix} + \begin{Bmatrix} 8位 \\ 16位 \end{Bmatrix} 偏移量$$

**BX:操作数的段基址默认在DS中**
**BP:操作数的段基址默认在SS中**

存储器寻址小结：

① 直接寻址　　默认DS段

② 寄存器间接寻址
③ 寄存器相对寻址　　BX, SI, DI：默认DS段；BP：默认SS段

④ 基址变址寻址
⑤ 相对基址变址寻址　　BX：默认DS段；BP：默认SS段

基址由BX, BP提供，变址由SI, DI提供，偏移量为8位或16位常量

17、串操作的源操作数在 DS 段，目的操作数在 ES 段。

18、MOV 指令：

传送指令：　MOV　DST, SRC

执行操作：　(DST) ← (SRC)

注意:

* DST、SRC 不能同时为段寄存器　　MOV DS, ES ×
* 立即数不能直接送段寄存器　　MOV DS, 2000H ×
* DST 不能是立即数和CS
* DST、SRC 不能同时为存储器寻址
* 不影响标志位

```
MOV AX, DSEG
MOV DS, AX
```

19、PUSH、POP 指令：

* 堆栈操作必须以字为单位。
* 不影响标志位
* 不能用立即寻址方式　　PUSH 1234H　×
* DST不能是CS　　POP CS　×

堆栈：　'先进后出' 的存储区，段地址存放在SS中，
SP在任何时候都指向栈顶，进出栈后自动修改SP

20、XCHG 指令：

交换指令：　XCHG OPR1, OPR2

执行操作：　(OPR1) ↔ (OPR2)

注意:

* 不影响标志位
* 不允许使用段寄存器

21、有效地址送寄存器指令：　　LEA　REG,　SRC

不影响标志位

REG 不能是段寄存器

SRC 必须为存储器寻址方式

22、类型转换指令：

- 类型转换指令

**CBW** AL → AX

执行操作：若(AL)的最高有效位为0，则(AH)= 00H
若(AL)的最高有效位为1，则(AH)= FFH

**CWD** AX → (DX,AX)

执行操作：若(AX)的最高有效位为0，则(DX)= 0000H
若(AX)的最高有效位为1，则(DX)= FFFFH

例：(AX) = 0BA45H
CBW ; (AX)=0045H
CWD ; (DX)=0FFFFH (AX)=0BA45H

注意：
* 无操作数指令
* 隐含对AL 或AX 进行符号扩展
* 不影响条件标志位

22、ADD 指令：

加法指令对**条件标志位**的影响

SF= { 1 结果为负 ; 0 否则 }　　ZF= { 1 结果为0 ; 0 否则 }

CF= { 1 和的最高有效位 有 向高位的进位 ; 0 否则 }

OF= { 1 两个操作数符号相同，而结果符号与之相反，即溢出 ; 0 否则 }

CF 位表示 无符号数 相加的溢出。
OF 位表示 带符号数 相加的溢出。

23、SUB 指令：

减法指令对条件标志位（CF/OF/ZF/SF）的影响：

CF= { 1 被减数的最高有效位 有 向高位的借位 ; 0 否则 }
或
CF= { 1 减法转换为加法运算时 无 进位 ; 0 否则 }

OF= { 1 两个操作数符号相反，而结果的符号与减数相同 ; 0 否则 }

CF 位表示 无符号数 减法的溢出。
OF 位表示 带符号数 减法的溢出。

24、乘法指令：

* **AL (AX) 为隐含的乘数寄存器。**
* **AX (DX,AX) 为隐含的乘积寄存器。**
* **SRC不能为立即数。**
* **除CF和OF外，对条件标志位无定义。**

MUL指令: CF,OF = { 00 乘积的高一半为零；11 否则

IMUL指令: CF,OF = { 00 乘积的高一半是低一半的符号扩展；11 否则

25、除法指令：

* **AX (DX,AX) 为隐含的被除数寄存器。**
* **AL (AX) 为隐含的商寄存器。**
* **AH (DX) 为隐含的余数寄存器。**
* **SRC不能为立即数。**
* **对所有条件标志位均无定义。**

26、调整指令：

（1）压缩的BCD码调整指令

- **DAA** 加法的十进制调整指令
- **DAS** 减法的十进制调整指令

（2）非压缩的BCD码调整指令

- **AAA** 加法的ASCII码调整指令
- **AAS** 减法的ASCII码调整指令
- **AAM** 乘法的ASCII码调整指令
- **AAD** 除法的ASCII码调整指令

27、逻辑运算指令：

逻辑非指令：NOT OPR OPR 不能为立即数

28、移位指令：

* OPR可用除立即数以外的任何寻址方式
* CNT=1，SHL　OPR, 1
　CNT>1，MOV CL,　CNT
　　SHL　OPR, CL　　；以SHL为例
* 条件标志位：
　CF = 移入的数值
　$OF = \begin{cases} 1 & \text{CNT=1时，最高有效位的值发生变化} \\ 0 & \text{CNT=1时，最高有效位的值不变} \end{cases}$
　移位指令：
　　SF、ZF、PF 根据移位结果设置，AF无定义
　循环移位指令：
　　不影响 SF、ZF、PF、AF

29、串操作指令：

**源串首地址（末地址）→ SI**

**目的串首地址（末地址）→ DI**

**串长度 → CX**

**建立方向标志**

**（ CLD 使 DF=0，STD 使 DF=1 ）**

源串（数据段）→ 目的串（附加段）

30、伪操作：汇编程序对源程序进行汇编时处理的操作，完成处理器选择、存储模式定义、数据定义、存储器分配、指示程序开始结束等功能。

31、地址计数器 $ ：保存当前正在汇编的指令的地址。

32、子程序调用：隐含使用堆栈保存返回地址。

33、NEAR 属性：调用程序和子程序在同一代码段中（段内调用）；
　FAR 属性：调用程序和子程序不在同一代码段中（段间调用）。

34、宏：源程序中一段有独立功能的程序代码。
　宏指令：用户自定义的指令。在编程时，将多次使用的功能用一条宏指令来代替。

35、CPU 与外设交换的信息分为三种：控制、状态信息；数据信息；命令信息；控制与状态信息都是为了正确地获得数据信息。

36、接口一般就由控制逻辑部件和一系列的寄存器构成，每个接口设备内部的寄存器数量不一样。但都可以分为三类：状态寄存器，数据寄存器；命令寄存器。
　数据寄存器：存放外设和主机间传送的数据
　状态寄存器：保存外设或接口的状态信息
　命令寄存器：保存 CPU 发给外设或接口的控制命令

38、中断源：引起中断的事件

外中断（硬中断）：
　外设的 I/O 请求　　——　可屏蔽中断
　电源掉电 / 奇偶错　——　非屏蔽中断

内中断（软中断）：
　INT 指令 / CPU 错（除法错、溢出）/

为调试程序设置的中断

39、存储器：

存储器以字节（8 bit）为编程单位

每个字节单元都有唯一的地址编码

地址用无符号整数来表示（编程用十六进制表示）

一个字要占用相继的两个字节

低位字节存入低地址，高位字节存入高地址

字单元地址用它的低地址来表示

机器以偶地址访问（读 / 写）存储器

40、CPU 组成：算术逻辑部件 ALU、控制器、寄存器。

41、寄存器与存储器：

| 寄 存 器 | 存 储 器 |
| --- | --- |
| **在CPU内部** | **在CPU外部** |
| **访问速度快** | **访问速度慢** |
| **容量小，成本高** | **容量大，成本低** |
| **用名字表示** | **用地址表示** |
| **没有地址** | **地址可用各种方式形成** |

42、外部设备与主机（ CPU 和存储器）的通信是通过外设接口（Interface）进行的，每个接口包括一组寄存器。

43、外设中每个寄存器有一个端口（Port）地址，构成一个独立于内存的 I / O 地址空间：0000H ~ FFFFH。