汇编语言知识点（一）

一、数制转换

1、常用进制：二进制、十进制、十六进制和八进制（转换方法略）

2、常用码制：

（1）原码 正数：符号为0，数值不变

负数：符号为1，数值不变

（2）反码 正数：符号为0，数值不变

负数：符号为1，数值各位取反（反码名称的由来）

（3）补码 正数：符号为0，数值不变

负数：符号为1，数值各位取反，末位加1（实际上是模运算）

3、补码加减运算：

（1）加法：两数连同符号一起加运算（符号在运算时当数值处理，这是补码普遍采用的原因之一）

（2）减法：被减数不变，减数各位取反，末位加1，然后两数连同符号一起加运算（因此计算机的减法是加实现的，这也是补码被普遍采用的另一个原因）

4、补码乘除运算：（具体规则见组成原理书，规则略）

（1）乘法：带符号数和无符号数运算结果不同，积为双倍字长

（2）除法：带符号数和无符号数运算结果不同，被除数为双倍字长，结果有商和余数两部分，根据运算规则，余数在高位，商在低位

二、8086CPU的基本组成

1、寄存器组织（所有寄存器均为16位）

（1）通用寄存器

AX ：即累加器，高8位称AH，低8位称AL，存放运算结果和I/O操作的数据

BX ：即基址寄存器，高8位称BH，低8位称BL，可用来存放基地址以及操作数据

CX ：即计数器，高8位称CH，低8位称CL，可用来存放计数值以及操作数据

DX ：即累加器，高8位称DH，低8位称DL，乘除法双倍字长的高半部分，也可存放操作数据，另外I/O操作时可存放I/O地址

（2）段基址寄存器

CS（Code Segment）：存放代码段基地址

DS（Data Segment）：存放数据段基地址

SS（Stack Segment）：存放堆栈段基地址

ES（Extra Segment）：存放附加段基地址

（3）指针寄存器

IP（Instruction Pointer）：存放指令指针，即代码段的偏移地址，与CS组合使用

SP（Stack Pointer）：存放栈顶指针，即堆栈段的栈顶地址，与SS组合使用

BP（Base Pointer）：存放基址指针，即堆栈段的偏移地址，与SS组合使用

**注意**：SP和BP的区别

（4）变址寄存器（数据段和附加段的偏移地址）

SI（Source Index）：存放源变址，与DS组合使用

DI（Destination Index）：存放目的变址，ES组合使用

**注意**：此二寄存器具有自动增量和自动减量功能，操作时亦可暂存数据

（5）程序状态字PSW

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | OF | DF | IF | TF | SF | ZF |  | AF |  | PF |  | CF |

**注意**：左为高位，右为低位，未标注的表示在8086CPU未使用

状态标志（6位）：

CF（Carry Flag）：进位或借位标志，CF=1，表示有进位或借位，初始状态CF=0

AF（Auxiliary Carry Flag）：半进位标志，AF=1，表示有半进位，初始状态AF=0，该标志位仅在BCD码运算时有意义

PF（Parity Flag）：奇偶标志，PF=1，表示结果1的个数为偶数，反之为奇数，初始状态PF=0

ZF（Zero Flag）：零标志，ZF=1，表示运算结果为0，初始状态ZF=0

SF（Sign Flag）：符号标志，SF即结果的最高位，初始状态SF=0

OF（Overflow Flag）：溢出标志，OF=1，表示运算结果溢出，即超出表示范围，初始状态OF=0；8位补码范围（-128—127），16位补码范围（-32768—32767）

控制标志（3位）：

IF（Interrupt Flag）：中断标志，IF=1，表示开放响应外部中断（即开中断状态），IF=0，表示屏蔽外部中断（即关中断状态），初始状态IF=0

DF（Direction Flag）：方向标志，DF=1，表示地址减量状态（即SI和DI减小），DF=0，表示地址增量状态（即SI和DI增大），初始状态DF=0

TF（Trap Flag）：陷阱标志，TF=1，表示CPU运行在调试时的单步方式，TF=0，表示CPU运行在正常方式，初始状态TF=0

2、物理地址的形成方式

**注意**：8086CPU的地址为20位，寻址空间为1M

（1）主存模型：

按字节（Byte）编址，即一个地址只对应8位二进制数

主存空间为一维线性空间

（2）物理地址形成方式

物理地址=（段基址寄存器）×16 + 偏移地址

即段基址寄存器的数值左移四位二进制数，再加上偏移地址，以此形成20位物理地址

由于段基址寄存器的值在某段操作时间内是不变的，所以每段最大空间64K

三、8086CPU的寻址方式

1、与数据有关的寻址方式（主要在数据段内找数据）

（1）立即寻址：数据在指令内以立即数形式存在

示例：MOV AX，05H （其中05H即为立即数）

操作限制：注意立即数的表数范围，不要超范围

立即数不可作为目的操作数

（2）寄存器寻址：数据存放在寄存器内

示例：MOV AX，BX （源操作数BX和目的操作数AX均为寄存器寻址）

操作限制：注意源操作数和目的操作数的格式匹配，即均为8位或均为16位

有些寄存器不允许作为目的操作数，例如：CS、DS、ES、SS、IP

（3）直接寻址：指令中给出数据的有效地址

示例：MOV AX，[2000H] （2000H即为偏移地址，加方括号和立即数区别）

MOV AX，TABLE （TABLE为符号地址名，对应相应的偏移地址值）

MOV AX，ES：[2000H] （允许段跨越形式，SS、CS亦可以，默认为DS）

操作限制：双操作数至少有一个为寄存器形式，不允许都为直接寻址

主存操作的高高低低原则

（4）寄存器间接寻址：数据的有效地址放在寄存器中

示例：MOV AX，[BX] （BX内为偏移地址，加方括号和寄存器寻址区别）

MOV [BP] ，AX （源和目的均可以）

操作限制：只有BX、BP、SI和DI可以使用，其它寄存器不允许

用BX、SI和DI时，是在数据段，以DS为基地址寄存器

用BP时，是在堆栈段，以SS为基地址寄存器

（5）寄存器相对寻址：在寄存器间址的基础上，再加上相对偏移量

示例：MOV AX，TABLE[BX] （符号地址TABLE即为相对偏移量）

MOV 50H[BP] ，AX （相对量也可以是立即数，立即数为补码）

注意：寄存器的值加上相对偏移量的值，二者之和为最后的偏移地址

**相对量可以是补码，位数不匹配，计算机自动完成位扩展，即正数前补0，负数前补1**

操作限制：只有BX、BP、SI和DI可以使用，其它寄存器不允许

用BX、SI和DI时，是在数据段，以DS为基地址寄存器

用BP时，是在堆栈段，以SS为基地址寄存器

**实质上可以构成一个一维数组**

（6）基址变址寻址：基址寄存器和变址寄存器的值相加，和为偏移地址

示例：MOV AX，[BX][SI] （两寄存器均加方括号）

MOV AX，[BX + SI] （也可这样书写）

**注意：基址寄存器只有BX和BP，变址寄存器只有SI和DI，二者排列组合只有四种形式**

操作限制：只有BX、BP、SI和DI可以使用，其SI和DI是配合使用

用BX时，是在数据段，以DS为基地址寄存器

用BP时，是在堆栈段，以SS为基地址寄存器

（7）相对基址变址寻址：上述（5）和（6）的组合使用

示例：MOV AX，TABLE[BX][SI] （基址、变址再加上相对量）

MOV AX，[BX + SI + TABLE] （也可这样书写）

操作限制：用BX时，是在数据段，以DS为基地址寄存器

用BP时，是在堆栈段，以SS为基地址寄存器

**实质上可以构成一个二维数组**

2、与转移地址有关的寻址方式（主要在代码段内找指令）

（1）段内直接寻址：段基址不变，指令直接给出**地址偏移量**（数值为补码）

**注意：不是偏移地址值**

**汇编程序计算当前指令与目的地址之间的相对偏移量，向后跳IP加正值，向前跳IP加负值（即减值）**

示例：JMP SHORT TABLE （TABLE为跳转的目的地址，SHORT标识为短跳，即8位偏移量）

JMP NEAR PTR TABLE （PTR为属性指针，NEAR标识16位偏移量）

**注意：**8位偏移量范围为-128—127，即IP最大加127，最小减128

16位偏移量范围为-32768—32767，即IP最大加32767，最小减32768

以上为IP的转移范围

操作限制：条件转移时，仅允许8位偏移量

只用符号地址形式，不允许直接给地址值

（2）段内间接寻址：段基址不变，用其它的寻址方式找出一个地址值（16位）放入IP

示例：JMP BX （把BX寄存器的值放入IP）

JMP WORD PTR [SI] （以寄存器SI的值为地址，取出一个字放入IP）

操作限制：除立即寻址外，其它寻址方式均可以（直接寻址方式即上述的段内直接寻址）

（3）段间直接寻址：给出的符号地址与跳转指令不在一个段内，段基址需要改变

示例：JMP FAR PTR TABLE （FAR标识TABLE在另一个段内）

操作时把TABLE所在段的段基址放入CS，把TABLE的段内偏移地址放入IP

（4）段间间接寻址：用其它的寻址方式找出一个双字，低位放入IP，高位放入CS

示例：JMP DWORD PTR [SI]

以寄存器SI的值为地址，取出一个双字，低字放入IP，高字放入CS

操作限制：立即寻址、寄存器寻址和直接寻址方式除外，其它寻址方式均可以

四、传送类指令

1、通用数据传送

（1）基本传送 MOV DST，SRC （SRC→DST）

﹡传送指令不影响标志位（后续传送类指令亦如此）

﹡有些寄存器不允许作为目的操作数，例如：CS、DS、ES、SS、IP，通常使用AX、BX、CX、DX、SI和DI

（2）交换指令 XCHG OPR1，OPR2 （OPR1←→OPR2）

﹡不允许使用立即数

（3）堆栈操作 PUSH SRC （SP—2→SP，SRC→[SP]）

POP DST （[SP]→DST，SP+2→SP）

﹡堆栈必须为16位操作

﹡出栈操作不要乱用寄存器

2、地址传送指令

（1）取有效地址 LEA REG，SRC

﹡目的操作数REG为16寄存器，段基址寄存器不允许使用

﹡源操作数必须有有效地址，立即寻址和寄存器寻址不允许使用

（2）地址指针传送（略）

3、标志位传送指令

（1）读写标志位 LAHF （程序状态字的低8位→AH）

SAHF （AH→程序状态字的低8位）

（2）标志位出入栈 PUSHF （SP—2→SP，程序状态字→[SP]）

POPF （[SP]→程序状态字，SP+2→SP）

4、I/O指令（亦称累加器专用指令）

IN DST，PORT （外设端口输入数据进累加器）

OUT PORT，SRC （累加器数据输出到外设端口）

﹡SRC或DST必须使用累加器，8位数据操作用AL，16位数据操作用AX

﹡PORT为端口地址，8位地址可直接书写（不加方括号），16位地址必须放入DX

五、算术运算指令

1、加法类指令 ADD DST，SRC （DST + SRC→DST）

ADC DST，SRC （DST + SRC + CF→DST）

INC OPR （OPR + 1 →OPR）

﹡双操作数至少有一个为寄存器形式

﹡通常使用AX、BX、CX、DX、SI和DI

﹡执行后影响相应的标志位，INC指令不影响CF

﹡**有符号数运算用OF判断是否溢出**

**无符号数运算用CF判断是否溢出**

2、减法类指令 SUB DST，SRC （DST—SRC→DST）

SBB DST，SRC （DST—SRC—CF→DST）

DEC OPR （OPR—1→OPR）

NEG OPR （—OPR→OPR，即OPR操作数各位取反，末位加1）

CMP OPR1，OPR2 （OPR1—OPR2，不存结果，置标志位）

﹡执行后影响相应的标志位，DEC指令不影响CF

3、乘法指令 MUL SRC （无符号数相乘）

IMUL SRC （有符号数相乘）

**注意：有符号数和无符号数乘法运算规则不同**

﹡源操作数SRC不允许使用立即寻址

﹡8位运算被乘数在AL寄存器中，乘积在AX寄存器中

﹡16位运算被乘数在AX寄存器中，乘积高半部分在DX寄存器中, 乘积低半部分在AX寄存器中

﹡对CF和OF有影响，其余标志位无意义（因为是多次加运算，标志位多次改变）

当乘积的高半部分均为0时，CF和OF均为0

当乘积的高半部分不为0时，CF和OF均为1

4、除法指令 DIV SRC （无符号数相除）

IDIV SRC （有符号数相除）

**注意：有符号数和无符号数除法运算规则也不同**

﹡源操作数SRC不允许使用立即寻址

﹡8位运算被乘数在AX寄存器中（双倍字长），商在AL寄存器中，余数在AH寄存器中

﹡16位运算被乘数在（DX，AX）双寄存器中，商在AX寄存器中, 余数在DX寄存器中

﹡因为是多次加法运算的累加，所以所有标志位无意义

5、位扩展指令 CBW （字节扩展为字，对AL处理，结果在AX中）

CWD （字扩展为双字，对AX处理，结果在（DX，AX）双寄存器中）

﹡**扩展规则 正数数值不变，前扩展位补0**

**负数数值不变，前扩展位补1**

6、十进制调整指令（略）

六、逻辑指令

1、逻辑运算 AND DST，SRC （DST∧SRC→DST 按位与，位与位之间无关联）

OR DST，SRC （DST∨SRC→DST 按位或，位与位之间无关联）

NOT OPR （OPR各位取反→OPR 按位取反，位与位之间无关联）

XOR DST，SRC （DST⊕SRC→DST 按位异或，位与位之间无关联）

TEST OPR1，OPR2 （OPR1∧OPR2，按位与不存结果，置标志位）

﹡NOT指令不影响标志位，其余指令置CF和OF为0

2、标准移位 SHL OPR，CNT （逻辑左移，最高位送CF，低位补0）

SHR OPR，CNT （逻辑右移，最低位送CF，高位补0）

SAL OPR，CNT （算术左移，最高位送CF，低位补0）

SAR OPR，CNT （算术右移，最低位送CF，高位保持不变）

﹡CNT为移位计数值，当CNT=1时，标志位有意义，

﹡一次移位后，最高位值改变，OF=1

﹡**CNT=1时，可直接书写；CNT>1时，必须放入CL寄存器中**

3、循环移位

ROL OPR，CNT（不带进位的循环左移，最高位送CF，同时最高位送最低位）

ROR OPR，CNT（不带进位的循环右移，最低位送CF，同时最低位送最高位）

RCL OPR，CNT（带进位的循环左移，最高位送CF，同时CF送最低位）

RCR OPR，CNT（带进位的循环右移，最低位送CF，同时CF送最高位）

﹡循环移位仅影响CF和OF，其它不影响

﹡当CNT=1时，OF有意义，最高位值改变，OF=1

七、串操作指令

﹡隐含寻址，源操作数在[DS：SI]，目的操作数在[ES：DI]

﹡SI和DI会自动递增或递减，当DF=0时，递增，当DF=1时，递减

﹡串的个数由CX计数

1、串传送（完成从源地址到目的地址的数据传送）

（1）标准格式 MOVS DST，SRC

示例：MOVS ES：BYTE PTR [DI]，DS：[SI] （执行完SI和DI自动加1或减1）

或MOVS ES：WORD PTR [DI]，DS：[SI] （执行完SI和DI自动加2或减2）

**注意：仅有此两种写法，唯一的区分是BYTE还是WORD**

（2）简便格式 MOVSB （字节传送，[SI]→[DI]，执行完SI和DI自动加1或减1）

MOVSW （字传送，[SI]→[DI]，执行完SI和DI自动加2或减2）

2、串取出（完成从源地址取出一个数据传送到累加器中）

（1）标准格式 LODS SRC （同上，需区分是BYTE还是WORD，示例略）

（2）简便格式 LODSB （字节传送，[SI]→AL，执行完SI自动加1或减1）

LODSW （字传送，[SI]→AX，执行完SI自动加2或减2）

3、串存入（完成从累加器中取出一个数据传送到目的地址中）

（1）标准格式 STOS DST （同上，需区分是BYTE还是WORD，示例略）

（2）简便格式 STOSB （字节传送，AL→[DI]，执行完DI自动加1或减1）

STOSW （字传送，AX→[DI]，执行完DI自动加2或减2）

﹡以上三个均不影响标志位

4、串比较（完成源地址和目的地址中的数据相减，不存结果，置标志位）

（1）标准格式 CMPS OPR1，OPR2 （同上，需区分是BYTE还是WORD，示例略）

（2）简便格式 CMPSB （字节比较，[SI]—[DI]，执行完SI和DI自动加1或减1）

CMPSW （字比较，[SI]—[DI]，执行完SI和DI自动加2或减2）

5、串比较（完成累加器中的数据和目的地址中的数据相减，不存结果，置标志位）

（1）标准格式 SCAS DST （同上，需区分是BYTE还是WORD，示例略）

（2）简便格式 SCASB （字节比较，AL—[DI]，执行完DI自动加1或减1）

SCASW （字比较，AX—[DI]，执行完DI自动加2或减2）

﹡以上两个影响标志位

6、重复前缀

**注意：上述串操作指令只完成一个数据的操作，要完成串内所有数据的操作，需加重复前缀**

（1）无条件前缀 REP 串指令

﹡执行过程 a、判断CX≠0，如不成立退出重复循环，否则b

b、CX—1→CX

c、执行其后的串指令，然后返回a

（2）有条件前缀 REPE/REPZ 串指令

﹡执行过程 a、判断CX≠0且ZF=1，如不成立退出重复循环，否则b

b、CX—1→CX

c、执行其后的串指令，然后返回a

**注意：仅判断条件不同，上述两个条件需同时成立，有一个不成立即退出循环**

REPNE/REPNZ 串指令

﹡执行过程 a、判断CX≠0且ZF=0，如不成立退出重复循环，否则b

b、CX—1→CX

c、执行其后的串指令，然后返回a

八、转移指令

1、无条件转移（即不判断条件码，执行至此就跳转）

JMP SHORT OPR （8位偏移量）

JMP NEAR PTR OPR （16位偏移量）

操作限制：OPR无允许为立即数，只能是符号地址

符号的取名原则：

（1）由小写字母a-z、大写字母A-Z、阿拉伯数字0-9、特殊符号@和\_组成

（2）第一个符号必须是字母或特殊符号\_

（3）有效长度31个字符

2、有条件转移（判断条件码或相应的标志位）

**注意：有条件转移指令均为8位偏移量**

（1）单个标志位

JZ/JE OPR （ZF=1时跳转，否则继续执行下一条指令）

JNZ/JNE OPR （ZF=0时跳转，否则继续执行下一条指令）

JO OPR （OF=1时跳转，否则继续执行下一条指令）

JNO OPR （OF=0时跳转，否则继续执行下一条指令）

JC OPR （CF=1时跳转，否则继续执行下一条指令）

JNC OPR （CF=0时跳转，否则继续执行下一条指令）

JP OPR （PF=1时跳转，否则继续执行下一条指令）

JNP OPR （PF=0时跳转，否则继续执行下一条指令）

JS OPR （SF=1时跳转，否则继续执行下一条指令）

JNS OPR （SF=0时跳转，否则继续执行下一条指令）

（2）比较两个无符号数 （之前执行X—Y操作）

JA/JNBE OPR （即X＞Y，CF=0且ZF=0时跳转）

JAE/JNB OPR （即X≥Y，CF=0时跳转，等同于JNC指令）

JB/JNAE OPR （即X＜Y，CF=1时跳转，等同于JC指令）

JBE/JNA OPR （即X≤Y，CF=1或ZF=1时跳转）

（3）比较两个有符号数 （之前执行X—Y操作）

JG/JNLE OPR （即X＞Y，SF⊕OF=0且ZF=0时跳转）

JGE/JNL OPR （即X≥Y，SF⊕OF=0时跳转）

JL/JNGE OPR （即X＜Y，SF⊕OF=1时跳转）

JLE/JNG OPR （即X≤Y，SF⊕OF=1或ZF=1时跳转）

（4） JCXZ OPR （CX=0时跳转）

3、循环指令

（1）无条件循环 LOOP OPR

﹡执行过程 a、CX—1→CX

b、判断CX≠0，如不成立执行下一条指令，如成立跳转

**注意：该指令先减1，再判断，因此LOOP指令需放在循环体的最后**

（2）有条件循环

LOOPE/LOOPZ OPR

﹡执行过程 a、CX—1→CX

b、判断CX≠0且ZF=1，如不成立执行下一条指令，如成立跳转

LOOPNE/LOOPNZ OPR

﹡执行过程 a、CX—1→CX

b、判断CX≠0且ZF=0，如不成立执行下一条指令，如成立跳转

注意：因为是两个条件同时成立，所以可能CX内数值不为0时，循环已经结束

4、子程序调用指令

CALL 子程序名 （子程序调用操作，即跳转到子程序入口处）

﹡因为需要返回，所以该指令把原断点入栈保护

RET < EXP > （子程序返回操作，出栈弹出原断点给IP）

\* EXP为数值表达式，可有可无，如有执行SP + EXP→SP

示例：

（1）段内直接调用 CALL OPR ； SP—2→SP

IP→[SP]

IP + 16位偏移量→IP

操作限制：OPR不允许为立即数，一般为符号地址

（2）段内间接调用 CALL BX ； SP—2→SP

IP→[SP]

BX→IP（即寻址找一个新地址值给IP）

操作限制：除立即寻址和直接寻址外，其它寻址方式皆可（上例为寄存器寻址）

（3）段间直接调用 CALL FAR PTR OPR ；SP—2→SP，CS→[SP]

SP—2→SP，IP→[SP]

OPR所在段的段基址→CS

OPR的偏移地址→IP

操作限制：OPR为符号地址，FAR PTR标示该符号地址与调用指令不在一个段内

（4）段间间接调用 CALL DWORD PTR [SI] ；SP—2→SP，CS→[SP]

SP—2→SP，IP→[SP]

所寻址到双字的高字→CS

所寻址到双字的低字→IP

操作限制：除立即寻址、直接寻址和寄存器寻址外，其它寻址方式皆可（必须要能寻址到双字，上例为寄存器间接寻址）

5、中断指令 （即特殊的段间间接子程序调用）

INT TYPE （TYPE为类型号，立即数形式存在）

INTO （溢出中断）

IRET （中断返回）

\*中断调用过程 SP—2→SP，PSW→[SP]

SP—2→SP，CS→[SP]

SP—2→SP，IP→[SP]

[TYPE\*4 + 2]→CS

[TYPE\*4]→IP

\*中断返回过程 [SP]→IP，SP + 2→SP

[SP]→CS，SP + 2→SP

[SP]→PSW，SP + 2→SP

九、其余相关指令

1、标志位处理指令

STI （IF设置为1）

CLI （IF设置为0）

STD （DF设置为1）

CLD （DF设置为0）

STC （CF设置为1）

CLC （CF设置为0）

CMC （CF取反）

2、其它控制指令

NOP （空操作指令，占一个字节）

HLT （停机等待，等待中断，中断返回后执行后续指令）

WAIT （等待指令，等待中断，中断返回后继续执行该指令）