x86汇编

第一章

一个char是1字节，一个int是4字节

CPU基本功能：执行机器指令、暂存少量数据、访问存储器

机器指令：数据传送指令、算术逻辑运算指令、转移指令、处理器控制指令、其他指令等

存储器最基本的存储单元为一个字节

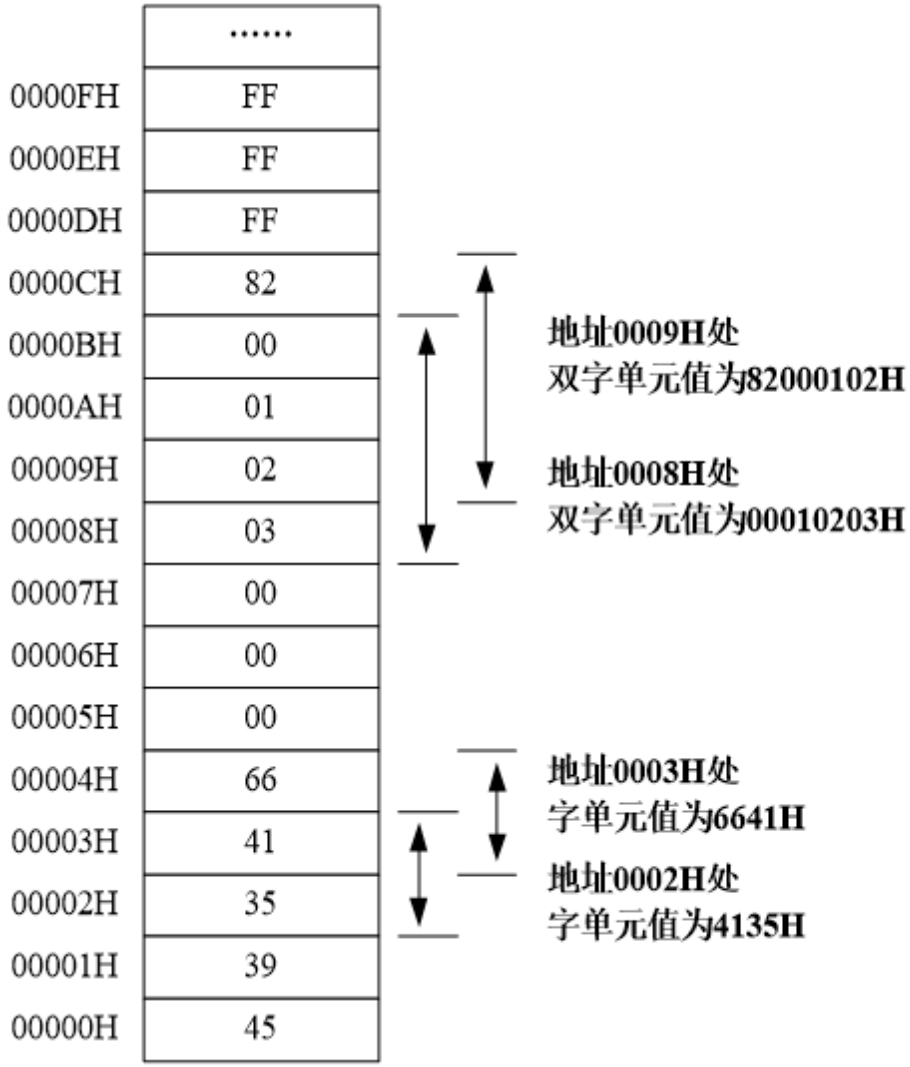
机器指令一般由操作码和操作数两部分构成，操作码指出要进行的操作或运算，操作数指出参与操作或运算的对象，也指出操作或运算结果存放的位置

汇编语言是低级语言

汇编语言编写的程序称为汇编语言源程序，把完成汇编工作的工具或程序叫做汇编程序（汇编器）

汇编语言优缺点：汇编语言与机器关系密切、汇编语言程序效率高 、编写汇编语言源程序繁琐 、汇编语言程序调试困难

一个字节存储单元的值是一个字节，两个连续的字节存储单元组成一个字存储单元，字存储单元的地址是其中较低的字节存储单元的地址，字存储单元的单元值从较高字节存储单元的值开始



第二章

一个双字等于32位，一个字等于16位，一个字节等于8位

8个32位的通用寄存器：EAX、EBX、ECX、EDX、ESI、EDI、EBP、ESP

8个16位的通用寄存器：AX、BX、CX、DX、SI、DI、BP、SP

8个8位的通用寄存器：AH、AL、BH、BL、CH、CL、DH、DL

传送指令MOV、交换指令XCHG,**源和目标的尺寸必须一致。不能同时是存储单元。（如不允许MOV [SI] [DI]）**

ADD、SUB、加1指令INC、减1指令DEC、NEG(**取得操作数的负数**)，**源和目标的尺寸必须一致，是否进位看是8位还是16位还是32位寄存器相加减**，例子看CH2B第34张PPT

LAHF(将 **标志寄存器（FLAGS）的低 8 位** 加载到 **AH 寄存器**)、SAHF(将 **AH 寄存器的值** 写回 **FLAGS 的低 8 位**)、ADC(目标操作数 = 目标操作数 + 源操作数 + CF)、SBB(目标操作数 = 目标操作数 - 源操作数 - CF)

立即寻址、寄存器寻址(直接寻址、寄存器间接寻址、寄存器相对寻址、基值+变址、通用)，ESP寄存器不可以作为变址寄存器,注意**变址的比例只能是1/2/4/8**

```
1. //显式指明存储器操作数的尺寸的特殊情况
```

```
2. MOV    DWORD PTR [EBX], 5      //双字存储单元
3. MOV    WORD  PTR [EBX+4], 5    //字存储单元
4. MOV    BYTE  PTR [EBX+8], 5    //字节存储单元
```

取有效地址指令：LEA REG OPRD 。源操作数OPRD必须是一个存储器操作数，目的操作数REG必须是一个16位或者32位的通用寄存器。

一个int是4字节、一个double是8字节，取int数组和double数组中特定某个元素的汇编看CH2C第45张幻灯片

条件转移指令：

指令格式	转移条件	转移说明	其他说明
JZ 标号 JE 标号	ZF=1 同上	等于0转移（Jump if zero） 相等转移（Jump if equal）	单个标志
JNZ 标号 JNE 标号	ZF=0 同上	不等于0转移（Jump if not zero） 不相等转移（Jump if not equal）	单个标志
JB 标号 JNAE 标号 JC 标号	CF=1 同上 同上	低于转移 不高于等于转移 进位位被置转移	单个标志 （无符号数）
JNBE 标号 JA 标号	(CF或ZF)=0 同上	不低于等于转移 高于转移	两个标志 （无符号数）
JLE 标号 JNG 标号	((SF异或OF)或ZF)=1 同上	小于等于转移 不大于转移	三个标志 （有符号数）
JNLE 标号 JG 标号	((SF异或OF)或ZF)=1 同上	不小于等于转移 大于转移	三个标志 （有符号数）

计算整型数组arri中10个元素值之和看CH2D第14张幻灯片

比较指令：CMP,用法：CMP DEST SRC，根据DEST-SRC的差影响标志寄存器的各个状态标志。两个操作数尺寸必须一致。一般配合JAE、JGE、JL等使用

有符号数间的次序关系称为大于(G)、等于(E)和小于(L)；

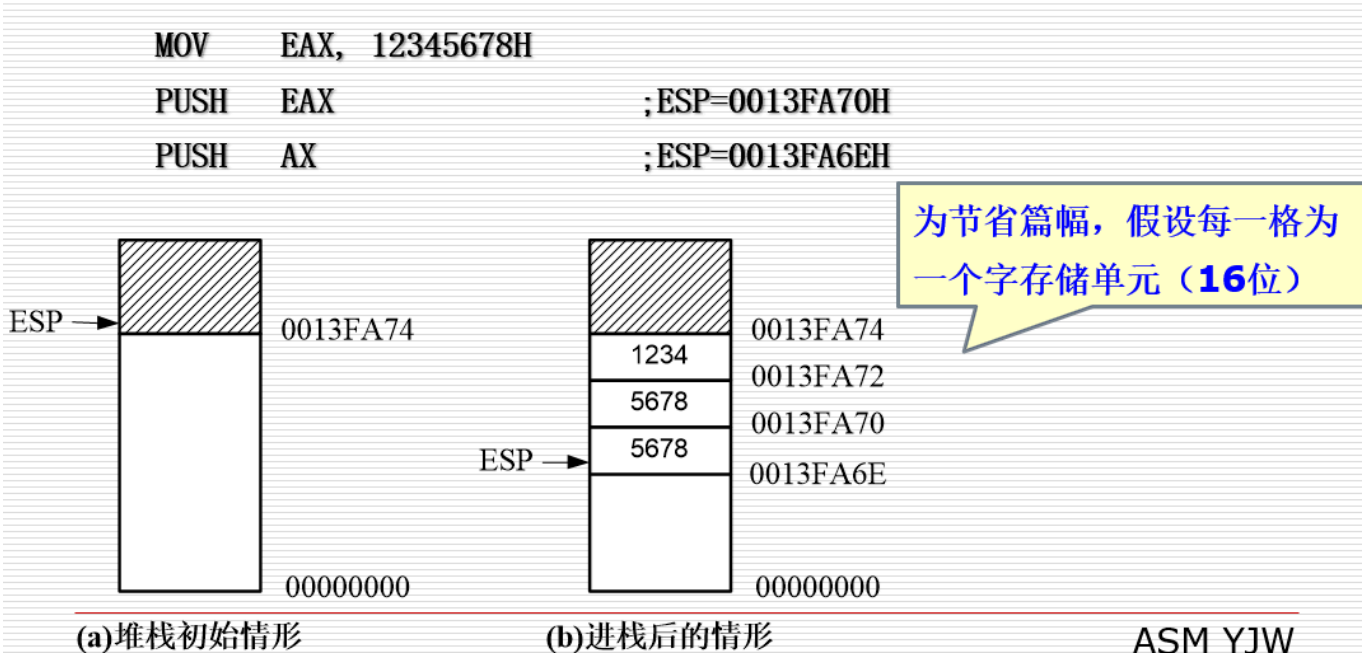
无符号数间的次序关系称为高于(A)、等于(E)和低于(B)。

判断数值大小看CH2D第18张幻灯片

无条件转移指令：JMP

堆栈：地址较大的一端被称为栈底，地址较小的一端被称为栈顶。主要用途：保护寄存器内容或者保护现场;保存返回地址;传递参数;安排局部变量或者临时变量

入栈指令：PUSH SRC，把一个双字数据压入堆栈时，先把ESP减4，然后再把双字数据送到ESP所指示的存储单元。把一个字数据压入堆栈时，先把ESP减2，再把字数据送到ESP所指示的存储单元。ESP总是指向栈顶。
操作数至少为一个字！所以PUSH AH/AL之类的八位寄存器操作数是错的



先插入高地址一个字的数据1234H，再插入低地址一个字的数据5678H，最后插入AX对应一个字的数据5678H

出栈指令：POP DEST，从栈顶弹出一个双字或者字数据到目的操作数DEST。出栈指令操作数不可以是立即数，也不能是代码段寄存器CS。从栈顶弹出一个双字数据时，先从ESP所指示的存储单元中取出一个双字送到目的操作数，然后把ESP加4。从栈顶弹出一个字数据时，先从ESP所指示的存储单元中取出一个字送到目的操作数，然后把ESP加2。至少出栈一个字！

PUSHA、POPA、PUSHAD、POPAD看CH2D第42-43张幻灯片

第三章

过程调用： CALL、RET，把’ abcde’ 转换成’ ABCED’ 看CH3E第十一张幻灯片，注意一个char为一个字节，所以MOV AX, [ESI]会把’ a’ ,’ b’ 加载到 AX（AL=’ a’ , AH=’ b’ ）

RET count:指令从堆栈弹出地址偏移，送到指令指针寄存器EIP。还额外把count 加到ESP,比如RET 4，ESP=ESP+count+4（+4 是因为 RET 本身会弹出返回地址）

无符号数乘法：MUL OPRD，OPRD不可以是立即数。乘数是OPRD，被乘数位于AL、AX或EAX中（由OPRD的尺寸决定，乘数和被乘数的尺寸一致）。

乘积尺寸翻倍：16位乘积送到AX；32位乘积送DX:AX；64位乘积送EDX:EAX。

有符号数乘法：

IMUL OPRD,与无符号数乘法类似

IMUL DEST, SRC。目的操作数DEST只能是16位或者32位通用寄存器。源操作数SRC可以是通用寄存器或存储单元（须与目的操作数尺寸一致），可以是一个立即数（尺寸不能超过目的操作数）

IMUL DEST, SRC1, SRC2。目的操作数DEST只能是16位或32位通用寄存器。源操作数SRC1可以是通用寄存器或存储单元（须与目的操作数尺寸一致），但不能是立即数。源操作数SRC2只能是一个立即数（尺寸不能超过目的操作数）。

无符号数除法:DIV OPRD，与无符号数乘法类似，被除数位于AX、DX:AX或EDX:EAX中。商在AL、AX或者EAX中；余数在AH、DX或者EDX中（商和余数的尺寸与oprd相同）。注意不能除溢出

有符号数除法:IDIV OPRD，与无符号数除法类似，如果不能整除，余数的符号与被除数一致，而且余数的绝对值小于除数的绝对值。

字节转换为字指令:CBW,把AL最高位符号拓展到AH。

字转换为双字指令:CWD,指令把AX中的符号扩展到DX。CWDE,把AX符号拓展到EAX高16位

双字转换为四字指令:CDQ,EAX符号拓展到EDX

符号拓展传送指令MOVSX, MOVSX DEST SRC

零拓展传送指令MOVZX，MOVZX DEST SRC，DEST只可以是16位或32位的通用寄存器

逻辑运算指令·：NOT、AND、OR、XOR、TEST

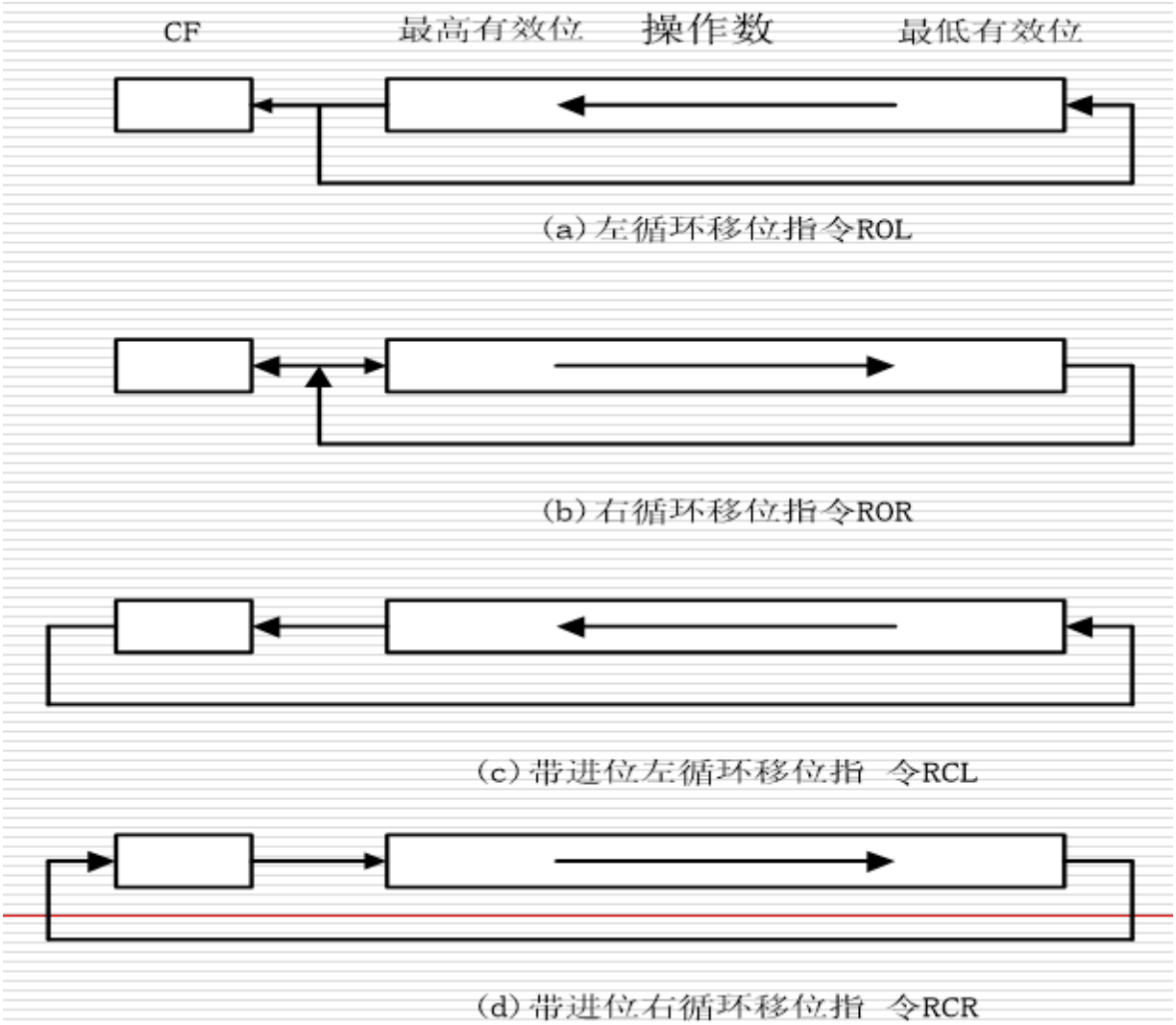
算术左移:SAL;逻辑左移:SHL;算术右移:SAR;逻辑右移:SHR;

左循环移位指令 ROL（ROtate Left）

右循环移位指令 ROR（ROtate Right）

带进位左循环移位指令 RCL（Rotate Left through CF）

带进位右循环移位指令 RCR（Rotate Right through CF）



双进度移位指令:

SHLD OPRD1，OPRD2，count

SHRD OPRD1，OPRD2，count

双精度左移指令SHLD，把目的操作数OPRD1左移指定的count位，在低端空出的位用操作数OPRD2高端的count位填补，但操作数OPRD2的内容保持不变。操作数OPRD1中最后移出的位保留在进位标志CF中。 双精度右移指令SHRD，把目的操作数OPRD1右移指定的count位，在高端空出的位用操作数OPRD2低端的count位填补，但操作数OPRD2的内容保持不变。操作数OPRD1最后移出的位保留在进位标志CF中。

循环指令:LOOP LAB。指令使寄存器ECX的值减1，如果结果不等于0，则转移到标号LAB处，否则顺序执行LOOP指令后的指令。类似于

```
1. DEC ECX
2. JNZ LAB
```

LOOPE/LOOPZ:指令使寄存器ECX的值减1，如果结果不等于0，并且零标志ZF等于1（表示上一条指令结果为0或相等），则转移到标号LAB处，否则顺序执行。

LOOPNE/LOOPNZ:指令使寄存器ECX的值减1，如果结果不等于0，并且零标志ZF等于0（表示不相等），则转移到标号LAB处，否则顺序执行。

JECXZ LAB;JCXZ LAB:指令实现当寄存器ECX的值等于0时转移到标号LAB处，否则顺序执行。或者，当寄存器CX的值等于0时转移到标号LAB处，否则顺序执行。

第四章

字符串装入指令:

LODSB:把寄存器ESI所指向的一个字节数据装入到累加器AL中，然后根据方向标志DF复位或置位使ESI之值增1或减1。

LODSW、LODSD则装入AX、EAX

字符串存储指令:

STOSB:把寄存器AL所指向的一个字节数据装入到寄存器EDI中，然后根据方向标志DF复位或置位使ESI之值增1或减1。

STOSW、STOSD类似

字符串传送指令:
MOVSB:不改变AL的值。把寄存器ESI所指向的一个字节数据传送到由寄存器EDI所指向的存储单元中，然后根据方向标志DF复位或置位使ESI和EDI之值分别增1或减1。

MOVSW、MOVSD类似

字符串扫描指令:

SCASB:把累加器AL的内容与由寄存器EDI所指向一个字节数据采用相减方式比较，相减结果反映到各状态标志（CF，ZF，OF，SF，PF和AF），但不影响两个操作数，然后根据方向标志DF复位或置位使EDI之值增1或减1。

SCASW、SCASD类似

字符串比较指令:
CMPSB:把寄存器ESI所指向的一个字节数据与由寄存器EDI所指向的一个字节数据采用相减方式比较，相减结果反映到各有关标志（CF，ZF，OF，SF，PF和AF），但不影响两个操作数，然后根据方向标志DF复位或置位使ESI和EDI之值分别增1或减1。

CMPSW、CPMSD类似

第五章

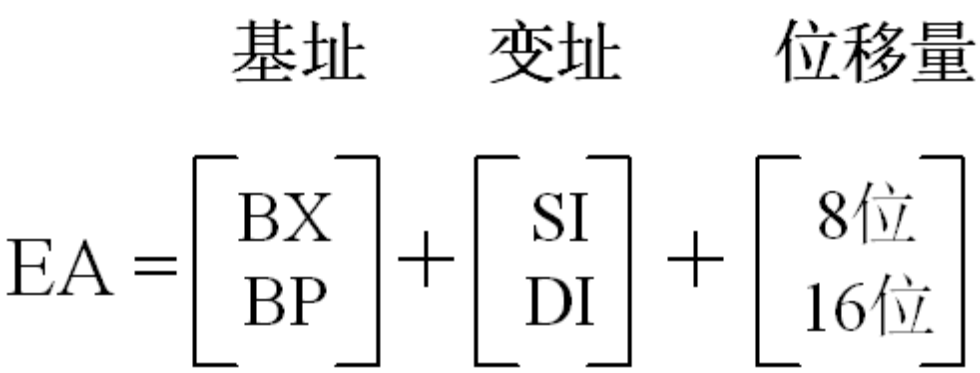
优化:用寄存器作为局部变量能大大提高效率;采用长度较短的指令或者指令片段

速度最大化方法：避免时钟数多的指令;减少转移指令;减少循环执行次数;存储器地址对齐

第六章

物理地址=段值*16+偏移，20位段起始地址的高16位XXXX称为段值。如十六进制表示的逻辑地址(段值:偏移)：1234:3456，1234H*10H+3456H=15796H

16位存储器寻址方式：



不允许两个变址寄存器同时出现在一个内存操作数中。如**MOV EAX, [SI+DI]**是非法的

段声明语句:SECTION xxx yyy，声明段xxx，段属性为yyy

第八章

CPU与外设之间交换的信息包括三类：数据;控制信息;状态信息

IN 累加器，端口地址,把端口地址的值送到累加器

OUT 端口地址，累加器；把累加器的值送到端口地址

中断指令:INT n;其中，n是一个0至0FFH的立即数。CPU在执行该中断指令后，便产生一个类型号为n的中断，从而转入对应的中断处理程序。