作者：Delort

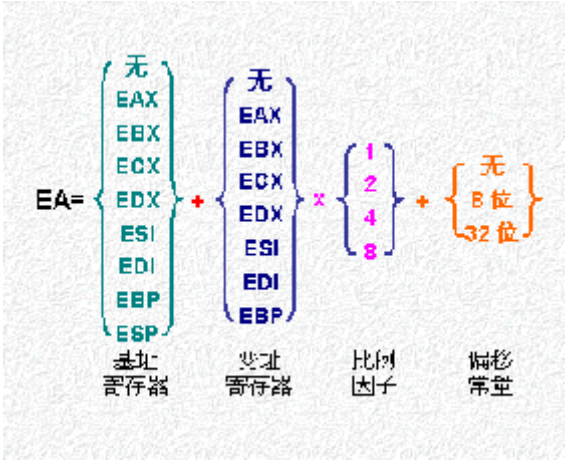
1.寻址方式

1. 立即寻址方式
2. 寄存器寻址方式
3. 直接寻址方式
4. 寄存器间接寻址方式
5. 寄存器相对寻址方式
6. 基址加变址寻址方式
7. 相对基址加变址寻址方式

   32位基址寄存器是：EAX、EBX、ECX、EDX、ESI、EDI、EBP和ESP；

   32位变址寄存器是：EAX、EBX、ECX、EDX、ESI、EDI和EBP(除ESP之外)。

   寄存器寻址规律图：



2.数据类型

名字	位数
byte	8
sbyte	8
word	16
sword	16
dword	32
sdword	32
fword	48
qword	64
tbyte	80
real4	32
real8	64
real10	80

3.常用伪指令

名称	含义
dup	使用一个整数表达式作为计数器，为多个数据项分配空间
db	定义有符号或者无符号8位变量
dw	定义有符号或者无符号16位变量
dd	定义有符号或者无符号32位变量，也可以定义实数
dq	定义有符号或者无符号64位变量，也可以定义实数
dt	定义有符号或者无符号80位变量，也可以定义实数
=	把符号与整数表达式联系起来
$	当前地址计数器
equ	把符号与整数表达式或任意文本联系起来
textequ	类似equ，分配文本，分配已有文本宏，分配整数常量表达式

4.与数据相关的指令和伪指令

指令

名字	含义	备注
mov	将源操作数复制到目的操作数	三个不能
movzx	进行全零扩展传输	
movsx	进行符号位扩展并传输	
lahf	将eflags寄存器的低字节复制到ah	
sahf	保存ah寄存器到eflags寄存器的低字节	
xchg	交换俩个操作数内容	
inc	自加一	
dec	自减一	
add	将长度相同的源操作数和目的操作数相加	
sub	从目的寄存器减去源操作数	
neg	将操作数转换为二进制补码，符号位取反	

伪指令和运算符

名字	含义	备注
offset	返回一个变量与其所在段起始地址之间的距离	
ptr	重写操作数默认的大小类型	
type	返回一个操作数或者数组中每个元素的大小	
lengthof	返回数组中元素的个数	
sizeof	返回数组初始化使用的字节数	
label	可以用不同的大小类型重新定义同一个变量	
typedef	创建用户自定义类型	pbyte typedef ptr byte

5.JMP和LOOP

5.1 jmp无条件跳转到目标地址

5.2 loop以cx为计数器进行循环，loopd是以ecx为计数器进行循环，loopw也是以cx寄存器位计数器

6.堆栈操作

名字	含义	备注
push	减少esp，再将源操作数复制到堆栈	16位esp减少2，32位esp减少4
pop	首先将esp指向的内容复制到目的操作数中，再增加esp值	
pushfd	把32位eflags寄存器压入堆栈	
popfd	把栈顶单元内容弹出到eflags寄存器	
pushad	将所有32位通用寄存器压入堆栈	顺序：eax,ecx,edx,ebx,esp,ebp,esi,edi
popad	与pushad相反	
pusha	与pushad类似，只不过变为16位寄存器	
popa	与pusha相反	

```
举例子：
;字符串翻转
.386
.model
.stack 4096
ExitProcess PROTO,dwExitCode:DWORD
.data
aName BYTE "Abraham Lincoln",0
nameSize = ($ - aName) - 1
.code
main PROC
;将名字压入堆栈
mov eax,nameSize
mov esi,0
L1:
   movzx eax,aName[esi] ;获取字符
   push eax
   inc esi
   loop L1
;将名字按逆序弹出堆栈
;并存入aName数组
   mov ecx,nameSize
   mov esi,0
L2:
   pop eax    ;获取字符
   mov aName[esi],al ;存入字符串
   inc esi
   loop L2
   INVOKE ExitProcess,0
main endp
```

```
END main
```

7.过程

7.1 PROC和ENDP伪指令，过程用PROC和ENDP来定义

```
;举例子：
main PROC
..
..
..
main ENDP
```

汇编可以从一个过程中goto到另外一个过程只需要在标号后面加::俩个:号定义为全局标号。

7.2 CALL和RET

名字	含义	备注
call	调用一个过程	将返回地址压入栈中，再把被调用过程复制到指令指针寄存器
ret	结束返回一个过程	类似 pop eip (不可以这样写: pop eip,但是ret含义与此相似)

7.3 USES

和PROC一起使用，让程序员列出在该过程中可能修改的寄存器，在过程开始处生成push，结尾处生成pop。

8.条件处理

名字	说明	备注
and	源操作数和目的操作数进行逻辑与操作	
or	源操作数和目的操作数进行逻辑或操作	
xor	源操作数和目的操作数进行逻辑异或操作	
not	对目标操作数进行逻辑非操作	
test	源操作数和目的操作数进行逻辑与操作，并设置CPU标志位	不修改操作数
cmp	从目的操作数中减去源操作数的隐含减法操作	不修改操作数

布尔指令影响零标志位，进位标志位，符号标志位，溢出标志位和奇偶标志位。

- 操作数结果等于0，零标志位置1。
- 操作数使得目标操作数的最高位有进位时，进位标志位置1。
- 符号标志位是目标操作数高位副本，1表示负数，0表示正数。
- 指令的结果超出了有符号目的操作数范围时，溢出标志置1。
- 指令使得目标操作数低字节中有偶数个1时，奇偶标志置1。

JCC	说明	备注
jz	为零跳转	zf=1
jnz		zf=0
jc	进位跳转	cf=1
jnc		cf=0
jo	溢出跳转	of=1
jno		of=0
js	有符号跳转	sf=1
jns		sf=0
jp	偶校验跳转	pf=1
jnp	奇校验跳转	pf=0
je	相等跳转	
jne	不相等跳转	
jcxz	cx=0跳转	
jecxz	ecx=0跳转	
jrcxz	rcx=0跳转	
ja	大于跳转	无符号（包括以下）
jb	小于跳转	
jnbe	不小于或等于跳转	
jnae	不大于或等于跳转	
jae	大于或等于跳转	
jbe	小于或等于跳转	
jnb	不小于跳转	
jna	不大于跳转	
jg	大于跳转	有符号数（包括以下）
jl	小于跳转	
jnle	不小于跳转或等于跳转	
jnge	不大于或等于跳转	
jge	大于或等于跳转	
jle	小于或等于跳转	
jnl	不小于跳转	

JCC	说明	备注
jng	不大于跳转	