

Intel 系列CPU 指令速查手册

| 指令名称 | 指令形式 | 机器码 | 标志位 (设置/测试) | 说 明 | 应用举例 |
|----------|--------------------|---------------|-------------------------|---------------------|--|
| ES: | ES: | 26 | | ES 段跨越前缀 | |
| CS: | CS: | 2E | | CS段跨越前缀 | |
| SS: | SS: | 36 | | SS段跨越前缀 | |
| DS: | DS: | 3E | | DS段跨越前缀 | |
| FS: | FS: | 64 | | FS 段跨越前缀 | |
| GS: | GS: | 65 | | GS段跨越前缀 | |
| OpSize: | OpSize: | 66 | | 操作数类型跨越前缀 | |
| Address: | Address: | 67 | | 地址类型跨越前缀 | |
| | AAA | 37 | 设置 AF CF | 加法后的ASCII 码调整 AL | AAA |
| AAD | AAD | D5 0A 或 D5 ib | 设置 SF ZF PF | 除法前的ASCII 码调整 AX | AAD |
| AAM | AAM | D4 0A 或 D4 ib | 设置 PF SF ZF | 乘法后的ASCII 码调整 AX | AAM |
| AAS | AAS | 3F | 设置 AF CF | 减法后的ASCII 码调整 AL | AAS |
| ADC | ADC AL, imm8 | 14 ib | 设置 AF CF OF SF PF ZF | 带进位加法 | ADC AL, 1F |
| | ADC AX, imm16 | 15 iw | | | ADC AX, 4F80 |
| | ADC EAX, imm32 | 15 id | | | ADC EAX, 00004F80 |
| | ADC r/m8, imm8 | 80 /2 ib | | | ADC BYTE Ptr [006387EA], 39 |
| | ADC r/m16,imm16 | 81 /2 iw | | | ADC WORD Ptr [006387EA], 1039 |
| | ADC r/m32,imm32 | 81 /2 id | | | ADC DWORD Ptr [006387EA], 00001039 |
| | ADC r/m16,imm8 | 83 /2 ib | | | ADC WORD Ptr [006387EA], 39 |
| | ADC r/m32,imm8 | 83 /2 ib | | | ADC DWORD Ptr [006387EA], 39 |
| | ADC r/m8,r8 | 10 /r | | | ADC [006387EA], AL |
| | ADC r/m16,r16 | 11 /r | | | ADC [006387EA], |

| | | | | | |
|-----|--------------------|----------|-------------------------|-----|--|
| | | | | | AX |
| | ADC r/m32,r32 | 11 / r | | | ADC [006387EA], EAX |
| | ADC r8,r/m8 | 12 /r | | | ADC AL, [006387EA] |
| | ADC r16,r/m16 | 13 /r | | | ADC AX, [006387EA] |
| | ADC r32,r/m32 | 13 /r | | | ADC EAX, [006387EA] |
| ADD | ADD AL, imm8 | 04 ib | 设置 AF CF OF SF PF ZF | 加法 | ADD AL, 1F |
| | ADD AX, imm16 | 05 iw | | | ADD AX, 4F80 |
| | ADD EAX, imm32 | 05 id | | | ADD EAX, 00004F80 |
| | ADD r/m8, imm8 | 80 /0 ib | | | ADD BYTE Ptr [006387EA], 39 |
| | ADD r/m16,imm16 | 81 /0 iw | | | ADD WORD Ptr [006387EA], 1039 |
| | ADD r/m32,imm32 | 81 /0 id | | | ADD DWORD Ptr [006387EA], 00001039 |
| | ADD r/m16,imm8 | 83 /0 ib | | | ADD WORD Ptr [006387EA], 39 |
| | ADD r/m32,imm8 | 83 /0 ib | | | ADD DWORD Ptr [006387EA], 39 |
| | ADD r/m8,r8 | 00 /r | | | ADD [006387EA], AL |
| | ADD r/m16,r16 | 01 /r | | | ADD [006387EA], AX |
| | ADD r/m32,r32 | 01 / r | | | ADD [006387EA], EAX |
| | ADD r8,r/m8 | 02 /r | | | ADD AL, [006387EA] |
| | ADD r16,r/m16 | 03 /r | | | ADD AX, [006387EA] |
| | ADD r32,r/m32 | 03 /r | | | ADD EAX, [006387EA] |
| AND | AND AL, imm8 | 24 ib | 设置 CF OF PF SF ZF | 逻辑与 | AND AL, 1F |
| | AND AX, imm16 | 25 iw | | | AND AX, 4F80 |
| | AND EAX, imm | 25 id | | | AND EAX, 00004 |

| | | | | | |
|------|------------------|----------|--------|------------------|--|
| | 32 | | | | F80 |
| | AND r/m8, imm8 | 80 /4 ib | | | AND BYTE Ptr [006387EA], 39 |
| | AND r/m16,imm16 | 81 /4 iw | | | AND WORD Ptr [006387EA], 1039 |
| | AND r/m32,imm32 | 81 /4 id | | | AND DWORD Ptr [006387EA], 00001039 |
| | AND r/m16,imm8 | 83 /4 ib | | | AND WORD Ptr [006387EA], 39 |
| | AND r/m32,imm8 | 83 /4 ib | | | AND DWORD Ptr [006387EA], 39 |
| | AND r/m8,r8 | 20 /r | | | AND BYTE Ptr [006387EA], AL |
| | AND r/m16,r16 | 21 /r | | | AND WORD Ptr [006387EA], AX |
| | AND r/m32,r32 | 21 /r | | | AND DWORD Ptr [006387EA], EAX |
| | AND r8,r/m8 | 22 /r | | | AND AL, [006387EA] |
| | AND r16,r/m16 | 23 /r | | | AND AX, [006387EA] |
| | AND r32,r/m32 | 23 /r | | | AND EAX, [006387EA] |
| ARPL | ARPL r/m16,r16 | 63 /r | 设置 ZF | 调整请求特权级 (286+PM) | ARPL AX, BX ; 如果 AX 的 RPL 小于 BX 的 RPL, 则改为 BX 的 RPL, 且 ZF 置 1, 否则 ZF 清 0 |
| | BOUND r16,m16&16 | 62 /r | 不影响标志位 | 越界检查 (80188+) | BOUND AX, [006387EA] ; 如果 AX 不在 [006387EA]及 [006387EA+2]的值中间, 则产生异常 5 |
| | BOUND r32,m32&32 | | | | BOUND EAX, [006387EA] ; 如果 EAX 不在 [006387EA]及 [006387EA+4]的值中间, 则产生异常 |

| | | | | | |
|-------|---------------|----------|--------|----------------------------|---|
| | | | | | 5 |
| BSF | BSF r16,r/m16 | 0F BC | 设置 ZF | 从低到高扫描目标，查找对应为 1 的位 (386+) | BSF AX, BX ; 把 BX 由低到高第一个 1 的位置送 AX。如全 0 则 ZF 置 1, 否则 ZF 清 0 |
| | BSF r32,r/m32 | | | | BSF EAX, EBX ; 把 EBX 由低到高第一个 1 的位置送 EAX, 如全 0 则 ZF 置 1, 否则 ZF 清 0 |
| BSR | BSR r16,r/m16 | 0F BD | 设置 ZF | 从高到低扫描目标，查找对应为 1 的位 (386+) | BSR AX, BX ; 把 BX 由高到低第一个 1 的位置送 AX。如全 0 则 ZF 置 1, 否则 ZF 清 0 |
| | BSR r32,r/m32 | | | | BSR EAX, EBX ; 把 EBX 由高到低第一个 1 的位置送 EAX, 如全 0 则 ZF 置 1, 否则 ZF 清 0 |
| BSWAP | BSWAP reg32 | 0F C8+rd | 不影响标志位 | 32 位寄存器高低字节交换(486+) | BSWAP EAX |
| BT | BT r/m16,r16 | 0F A3 | 设置 CF | 测试目标的指定位 (386+) | BT AX, BX ; 如果 AX 中 BX 指定的位为 1, 则 CF 置 1, 否则 CF 清 0 |
| | BT r/m32,r32 | 0F A3 | | | BT EAX, EBX ; 如果 EAX 中 EBX 指定的位为 1, 则 CF 置 1, 否则 CF 清 0 |
| | BT r/m16,imm8 | 0F BA /4 | | | BT AX, 01 ; 如果 AX 中 01 位为 1, 则 CF 置 1, 否则 CF 清 0 |
| | BT r/m32,imm8 | 0F BA /4 | | | BT EAX, 01 ; 如果 EAX 中 01 位为 1, 则 CF 置 1, 否则 CF 清 0 |
| BTC | BTC r/m16,r16 | 0F BB | 设置 CF | 将目标指定位取反 (386+) | BTC AX, BX ; 将 AX 中 BX 指定的位取反, CF 存放指定位并取反位 |

| | | | | | |
|-----|----------------|----------|-------|------------------|---|
| | BTC r/m32,r32 | 0F BB | | | BTC EAX, EBX ; 将 EAX 中 EBX 指定的位取反, CF 存放指定位并取反 |
| | BTC r/m16,imm8 | 0F BA /7 | | | BTC AX, 01 ; 将 AX 中 01 位取反, CF 存放指定位并取反 |
| | BTC r/m32,imm8 | 0F BA /7 | | | BTC EAX, 01 ; 将 EAX 中 01 位取反, CF 存放指定位并取反 |
| Btr | Btr r/m16,r16 | 0F B3 | 设置 CF | 将目标指定位清 0 (386+) | Btr AX, BX ; 将 AX 中 BX 指定的位清 0, CF 存放指定位并清 0 |
| | Btr r/m32,r32 | 0F B3 | | | Btr EAX, EBX ; 将 EAX 中 EBX 指定的位清 0, CF 存放指定位并清 0 |
| | Btr r/m16,imm8 | 0F BA /6 | | | Btr AX, 01 ; 将 AX 中 01 位清 0, CF 存放指定位并清 0 |
| | Btr r/m32,imm8 | 0F BA /6 | | | Btr EAX, 01 ; 将 EAX 中 01 位清 0, CF 存放指定位并清 0 |
| BTS | BTS r/m16,r16 | 0F AB | 设置 CF | 将目标指定位置 1 (386+) | BTS AX, BX ; 将 AX 中 BX 指定的位置 1, CF 存放指定位并置 1 |
| | BTS r/m32,r32 | 0F AB | | | BTS EAX, EBX ; 将 EAX 中 EBX 指定的位置 1, CF 存放指定位并置 1 |
| | BTS r/m16,imm8 | 0F BA /5 | | | BTS AX, 01 ; 将 AX 中 01 位置 1, CF 存放指定位并置 1 |
| | BTS r/m32,imm8 | 0F BA /5 | | | BTS EAX, 01 ; 将 EAX 中 01 位置 1, CF 存放指定位并置 1 |

| | | | | | |
|--------|-------------------|----------|--------------------|------------------------|-----------------|
| | CALL rel16 | E8 cw | 不影响标志位 | 子程序调用(16 位相对寻址) | |
| | CALL rel32 | E8 cd | | 子程序调用(32 位相对寻址) | |
| | CALL r/m16 | FF /2 | | 子程序调用(16 位间接寻址) | |
| | CALL r/m32 | FF /2 | | 子程序调用(32 位间接寻址) | |
| | CALL ptr16:16 | 9A cd | | 子程序调用(直接绝对寻址) | |
| | CALL ptr16:32 | 9A cp | | 子程序调用(直接绝对寻址) | |
| | CALL m16:16 | FF /3 | | 子程序调用(间接绝对寻址) | |
| | CALL m16:32 | FF /3 | | | 子程序调用(间接绝对寻址) |
| CBW | CBW | 98 | 不影响标志位 | 将 AL 值带符号扩展到 AX | CBW |
| CDQ | CDQ | 99 | 不影响标志位 | 将 EAX 值带符号扩展到 EDX: EAX | CDQ |
| CLD | CLD | FC | 设置 DF | 清除方向位(DF)标志 | CLD |
| CLI | CLI | FA | 设置 IF | 清除中断允许位(IF)标志 | CLD |
| CLTS | CLTS | 0F 06 | 不影响标志位 | 清除任务开关标志(TSF) | CLTS |
| CMC | CMC | F5 | 设置 CF | 进位标志取反 | CMC |
| CMOVcc | CMOVA r16, r/m16 | 0F 47 /r | 高于(CF=0 and ZF=0) | 条件传送指令 | CMOVA AX, BX |
| | CMOVA r32, r/m32 | | | | CMOVA EAX, EBX |
| | CMOVAE r16, r/m16 | 0F 43 /r | 高于等于(CF=0) | | CMOVAE AX, BX |
| | CMOVAE r32, r/m32 | | | | CMOVAE EAX, EBX |
| | CMOVB r16, r/m16 | 0F 42 /r | 低于(CF=1) | | CMOVB AX, BX |
| | CMOVB r32, r/m32 | | | | CMOVB EAX, EBX |
| | CMOVBE r16, r/m16 | 0F 46 /r | 低于等于(CF=1 or ZF=1) | | CMOVBE AX, BX |
| | CMOVBE r32, r/m32 | | | | CMOVBE EAX, EBX |
| | CMOVC r16, r/m16 | 0F 42 /r | 有进位(CF=1) | | CMOVC AX, BX |
| | CMOVC r32, r/ | | | | CMOVC EAX, EBX |

| | | | |
|-----------------------|----------|-------------------------------|---------------------|
| m32 | | | |
| CMOVE r16, r/m16 | 0F 44 /r | 等于($ZF=1$) | CMOVE AX, BX |
| CMOVE r32, r/m32 | | | CMOVE EAX, EBX |
| CMOVG r16, r/m16 | 0F 4F /r | 大于($ZF=0$ and $SF=0F$) | CMOVG AX, BX |
| CMOVG r32, r/m32 | | | CMOVG EAX, EBX |
| CMOVGE r16, r/m16 | 0F 4D /r | 大于等于 ($SF=0F$) | CMOVGE AX, BX |
| CMOVGE r32, r/m32 | | | CMOVGE EAX, EBX |
| CMOVL r16, r/m16 | 0F 4C /r | 小于($SF<>0F$) | CMOVL AX, BX |
| CMOVL r32, r/m32 | | | CMOVL EAX, EBX |
| CMOVLE r16, r/m16 | 0F 4E /r | 小于等于($ZF=1$ or $SF<>0F$) | CMOVLE AX, BX |
| CMOVLE r32, r/m32 | | | CMOVLE EAX, EBX |
| CMOVNA r16, r/m16 | 0F 46 /r | 不高于($CF=1$ or $ZF=1$) | CMOVNA AX, BX |
| CMOVNA r32, r/m32 | | | CMOVNA EAX, EBX |
| CMOVNAE r16, r/m16 | 0F 42 /r | 不高等于($CF=1$) | CMOVNAE AX, BX |
| CMOVNAE r32, r/m32 | | | CMOVNAE EAX, EBX |
| CMOVNB r16, r/m16 | 0F 43 /r | 不低于($CF=0$) | CMOVNB AX, BX |
| CMOVNB r32, r/m32 | | | CMOVNB EAX, EBX |
| CMOVNBE r16, r/m16 | 0F 47 /r | 不低等于($CF=0$ and $ZF=0$) | CMOVNBE AX, BX |
| CMOVNBE r32, r/m32 | | | CMOVNBE EAX, EBX |
| CMOVNC r16, r/m16 | 0F 43 /r | 无进位($CF=0$) | CMOVNC AX, BX |
| CMOVNC r32, r/m32 | | | CMOVNC EAX, EBX |
| CMOVNE r16, r | 0F 45 /r | 不等于($ZF=0$) | CMOVNE AX, BX |

| | | | | |
|-----------------------|----------|-------------------------------|--|---------------------|
| /m16 | | | | |
| CMOVNE r32, r/m32 | | | | CMOVNE EAX, EBX |
| CMOVNG r16, r/m16 | 0F 4E /r | 不大于($ZF=1$ or $SF<>OF$) | | CMOVNG AX, BX |
| CMOVNG r32, r/m32 | | | | CMOVNG EAX, EBX |
| CMOVNGE r16, r/m16 | 0F 4C /r | 不大等于 ($SF<>OF$) | | CMOVNGE AX, BX |
| CMOVNGE r32, r/m32 | | | | CMOVNGE EAX, EBX |
| CMOVNL r16, r/m16 | 0F 4D /r | 不小于($SF=OF$) | | CMOVNL AX, BX |
| CMOVNL r32, r/m32 | | | | CMOVNL EAX, EBX |
| CMOVNLE r16, r/m16 | 0F 4F /r | 不小于于($ZF=0$ and $SF=OF$) | | CMOVNLE AX, BX |
| CMOVNLE r32, r/m32 | | | | CMOVNLE EAX, EBX |
| CMOVNO r16, r/m16 | 0F 41 /r | 无溢出($OF=0$) | | CMOVNO AX, BX |
| CMOVNO r32, r/m32 | | | | CMOVNO EAX, EBX |
| CMOVNP r16, r/m16 | 0F 4B /r | 非偶数($PF=0$) | | CMOVNP AX, BX |
| CMOVNP r32, r/m32 | | | | CMOVNP EAX, EBX |
| CMOVNS r16, r/m16 | 0F 49 /r | 非负数($SF=0$) | | CMOVNS AX, BX |
| CMOVNS r32, r/m32 | | | | CMOVNS EAX, EBX |
| CMOVNZr16, r/m16 | 0F 45 /r | 非零($ZF=0$) | | CMOVNZAX, BX |
| CMOVNZr32, r/m32 | | | | CMOVNZEAX, EBX |
| CMOVO r16, r/m16 | 0F 40 /r | 溢出($OF=1$) | | CMOVO AX, BX |
| CMOVO r32, r/m32 | | | | CMOVO EAX, EBX |
| CMOVPr16, r/m16 | 0F 4A /r | 偶数($PF=1$) | | CMOVPA, BX |
| CMOVPr32, r/ | | | | CMOVPEAX, EBX |

| | | | | | |
|-----|----------------------|----------|-------------------------|------------------|--|
| | m32 | | | | |
| | CMOVPE r16, r/m16 | 0F 4A /r | 偶数(PF=1) | | CMOVPE AX, BX |
| | CMOVPE r32, r/m32 | | | | CMOVPE EAX, EBX |
| | CMOVPO r16, r/m16 | 0F 4B /r | 奇数(PF=0) | | CMOVPO AX, BX |
| | CMOVPO r32, r/m32 | | | | CMOVPO EAX, EBX |
| | CMOVS r16, r/m16 | 0F 48 /r | 负数(SF=1) | | CMOVS AX, BX |
| | CMOVS r32, r/m32 | | | | CMOVS EAX, EBX |
| | CMOVZ r16, r/m16 | 0F 44 /r | 为零(ZF=1) | | CMOVZ AX, BX |
| | CMOVZ r32, r/m32 | | | | CMOVZ EAX, EBX |
| CMP | CMP AL, imm8 | 3C ib | 设置 AF CF OF PF SF ZF | 比较大小， 然后设置标志位 | CMP AL, 1F |
| | CMP AX, imm16 | 3D iw | | | CMP AX, 4F80 |
| | CMP EAX, imm32 | 3D id | | | CMP EAX, 00004F80 |
| | CMP r/m8, imm8 | 80 /7 ib | | | CMP BYTE Ptr [006387EA], 39 |
| | CMP r/m16,imm16 | 81 /7 iw | | | CMP WORD Ptr [006387EA], 1039 |
| | CMP r/m32,imm32 | 81 /7 id | | | CMP DWORD Ptr [006387EA], 00001039 |
| | CMP r/m16,imm8 | 83 /7 ib | | | CMP WORD Ptr [006387EA], 39 |
| | CMP r/m32,imm8 | 83 /7 ib | | | CMP DWORD Ptr [006387EA], 39 |
| | CMP r/m8,r8 | 38 /r | | | CMP BYTE Ptr [006387EA], AL |
| | CMP r/m16,r16 | 39 /r | | | CMP WORD Ptr [006387EA], AX |
| | CMP r/m32,r32 | 39 / r | | | CMP DWORD Ptr [006387EA], EAX |
| | CMP r8,r/m8 | 3A /r | | | CMP AL, [006387EA] |

| | | | | | |
|---------------|----------------------|--------------|-------------------------|--------------------|--|
| | CMP r16,r/m16 | 3B /r | | | CMP AX, [006387EA] |
| | CMP r32,r/m32 | 3B /r | | | CMP EAX, [006387EA] |
| CMP | CMPB m8, m8 | A6 | 设置 AF CF OF PF SF ZF | 比较字符串，每次比较1 个字节 | CMPS StrING1, StrING2 ; 源串 DS:(E)SI, 目的串: ES:(E)DI |
| | CMPB m16, m16 | A7 | | 比较字符串，每次比较1 个字 | CMPS StrING1, StrING2 ; 源串 DS:(E)SI, 目的串: ES:(E)DI |
| | CMPB m32, m32 | A7 | | 比较字符串，每次比较1 个双字 | CMPS StrING1, StrING2 ; 源串 DS:(E)SI, 目的串: ES:(E)DI (386+) |
| | CMPSB | A6 | | 比较字符串，每次比较1 个字节 | CMPSB ; 源串 DS:(E)SI, 目的串: ES:(E)DI |
| | CMPSW | A7 | | 比较字符串，每次比较1 个字 | CMPSW ; 源串 DS:(E)SI, 目的串: ES:(E)DI |
| | CMPSD | A7 | | 比较字符串，每次比较1 个双字 | CMPSD ; 源串 DS:(E)SI, 目的串: ES:(E)DI (386+) |
| CMPXCHG | CMPXCHG r/m8,r8 | 0F B0 /r | 设置 AF CF OF PF SF ZF | 比较交换 (486+) | CMPXCHG BL,CL ; 如果AL与BL相 等，则CL送BL且 ZF置1; 否则BL送 CL, 且ZF清0 |
| | CMPXCHG r/m16,r16 | 0F B1 /r | | | CMPXCHG BX,CX ; 如果AX与BX相 等，则CX送BX且 ZF置1; 否则BX 送CX, 且ZF清0 |
| | CMPXCHG r/m32,r32 | 0F B1 /r | | | CMPXCHG EBX,ECX ; 如果 EAX与EBX相等， 则ECX送EBX且 ZF置1; 否则EBX 送ECX, 且ZF清0 |
| CMPXCHG 8B | CMPXCHG8B m64 | 0F C7 /1 m64 | 设置 ZF | 比较交换 (486+) | CMPXCHG [ESI] ; 比较 EDX: EAX |

| | | | | | |
|-------|------------------|----------|-----------------------|--|---|
| | | | | | 与 64 位的目标，如果相等则 ECX: EBX 送往目标且 ZF 置 1，否则目标送 EDX: EAX 且 ZF 清 0 |
| CPUID | CPUID | 0F A2 | 不影响标志位 | CPU 标识送 EAX、EBX、ECX、EDX | CPUID |
| CWD | CWD | 99 | 不影响标志位 | 将 AX 带符号扩展到 DX: AX | CWD |
| CWDE | CWDE | 98 | 不影响标志位 | 将 AX 带符号扩展到 EAX | CWDE |
| | DAA | 27 | 设置 AF CF PF SF ZF | 加法后的 10 进制调整 AL | DAA |
| DAS | DAS | 2F | 设置 AF CF PF SF ZF | 减法后的 10 进制调整 AL | DAS |
| DEC | DEC r/m8 | FE /1 | 设置 AF OF PF SF ZF | 目标减 1 | DEC BYTE Ptr [00458A90] |
| | DEC r/m16 | FF /1 | | | DEC WORD Ptr [00458A90] |
| | DEC r/m32 | FF /1 | | | DEC DWORD Ptr [00458A90] |
| | DEC r16 | 48 +rw | | | DEC AX |
| | DEC r32 | 48 +rd | | | DEC EAX |
| DIV | DIV r/m8 | F6 /6 | AF CF OF PF SF ZF 未定义 | 无符号除法 | DIV BL ; AX 除以 BL，商在 AL 中，余数在 AH 中 |
| | DIV r/m16 | F6 /7 | | | DIV BX ; DX: AX 除以 BX，商在 AX 中，余数在 DX 中 |
| | DIV r/m32 | F6 /7 | | | DIV EBX; EDX: EAX 除以 BX，商在 EAX 中，余数在 EDX 中 |
| | EMMS | 0F 77 | 不影响标志位 | 清空 MMX 状态 | EMMS |
| ENTER | ENTER imm16,0 | C8 iw 00 | 不影响标志位 | 为子程序建立堆栈框架，imm16 指定要分配的堆栈大小，另外一个操作数指明子程序的等级 (80188+) | ENTER 12,0 |
| | ENTER imm16,1 | C8 iw 01 | | | ENTER 12,1 |
| | ENTER imm16,imm8 | C8 iw ib | | | DENTER 12,4 |
| ESC | ESC 0 | D8h xxh | 不影响标志位 | 处理器放弃总线 | ESC 0 |
| | ESC 1 | D9h xxh | | | ESC 1 |
| | ESC 2 | DAh xxh | | | ESC 2 |

| | | | | | |
|--------|--------------------|----------|---|--|----------------------------|
| | ESC 3 | DBh xxh | | | ESC 3 |
| | ESC 4 | DCh xxh | | | ESC 4 |
| | ESC 5 | DDh xxh | | | ESC 5 |
| | ESC 6 | DEh xxh | | | ESC 6 |
| | ESC 7 | DFh xxh | | | ESC 7 |
| | F2XM1 | D9 F0 | 设置 C1 (C0, C2, C3 未定义) | $ST(0) \leftarrow (2 * ST(0) - 1)$ | F2XM1 |
| FABS | FABS | D9 E1 | 设置 C1 (C0, C2, C3 未定义) | 求绝对值: $ST(0) \leftarrow ABS(ST(0))$ | FABS |
| FADD | FADD m32real | D8 /0 | 设置 C1 (C0, C2, C3 未定义) | 实数加法: $ST(0) \leftarrow ST(0) + m32real$ | FADD REAL4 Ptr [00459AF0] |
| | FADD m64real | DC /0 | | 实数加法: $ST(0) \leftarrow ST(0) + m64real$ | FADD REAL8 Ptr [00459AF0] |
| | FADD ST(0), ST(i) | D8 C0+i | | 实数加法: $ST(0) \leftarrow ST(0) + ST(i)$ | FADD ST(0), ST(1) |
| | FADD ST(i), ST(0) | DC C0+i | | 实数加法: $ST(i) \leftarrow ST(i) + ST(0)$ | FADD ST(1), ST(0) |
| FADDP | FADDP ST(i), ST(0) | DE C0+i | 设置 C1 (C0, C2, C3 未定义) | 先进行实数加法: $ST(i) \leftarrow ST(i) + ST(0)$, 然后进行一次出栈操作 | FADDP ST(1), ST(0) |
| | FADDP | DE C1 | | 先进行实数加法: $ST(0) \leftarrow ST(0) + ST(1)$, 然后进行一次出栈操作 | FADDP |
| FIADD | FIADD m32int | DA /0 | 设置 C1 (C0, C2, C3 未定义) | 加整数: $ST(0) \leftarrow ST(0) + m32int$ | FIADD DWORD Ptr [00812CD0] |
| | FIADD m16int | DE /0 | | 加整数: $ST(0) \leftarrow ST(0) + m16int$ | FIADD WORD Ptr [00812CD0] |
| FBLD | FBLD m80bcd | DF /4 | 设置 C1 (C0, C2, C3 未定义) | 将 BCD 数装入 ST(0), 然后压栈 push ST(0) | |
| FBSTP | FBSTP m80bcd | DF /6 | 设置 C1 (C0, C2, C3 未定义) | 将 ST(0) 以 BCD 数保存在 m80bcd, 然后出栈 pop ST(0) | |
| FCHS | FCHS | D9 E0 | 设置 C1 (C0, C2, C3 未定义) | 改变 ST(0) 的符号位, 即求负数 | FCHS |
| FCLEX | FCLEX | 9B DB E2 | PE, UE, OE, ZF, DE, IE, ES, SF, B 清 0, 设置 C0 (C1, C2, C3 未定义) | 清除浮点异常标志(检查非屏蔽浮点异常) | FCLEX |
| FNCLEX | FNCLEX | DB E2 | PE, UE, OE, ZF, DE, IE, ES, SF, B 清 0, 设 | 清除浮点异常标志(不检查非屏蔽浮点异常) | FNCLEX |

| | | | | | |
|---------|-----------------------|---------|---|--------------------------------------|----------------------------|
| | | | 置 C0 (C1, C2, C3 未定义) | | |
| FCMOVcc | FCMOVB ST(0), ST(i) | DA C0+i | 小于(CF=1), 设置 C1 (C0, C2, C3 未定义) | 条件浮点传送指令 | FCMOVB ST(0), ST(1) |
| | FCMOVE ST(0), ST(i) | DA C8+i | 等于(ZF=1), 设置 C1 (C0, C2, C3 未定义) | | FCMOVE ST(0), ST(1) |
| | FCMOVBE ST(0), ST(i) | DA D0+i | 小于等于(CF=1 or ZF=1), 设置 C1 (C0, C2, C3 未定义) | | FCMOVBE ST(0), ST(1) |
| | FCMOVU ST(0), ST(i) | DA D8+i | unordered(PF=1), 设置 C1 (C0, C2, C3 未定义) | | FCMOVU ST(0), ST(1) |
| | FCMOVNB ST(0), ST(i) | DB C0+i | 不小于(CF=0), 设置 C1 (C0, C2, C3 未定义) | | FCMOVNB ST(0), ST(1) |
| | FCMOVNE ST(0), ST(i) | DB C8+i | 不等于(ZF=0), 设置 C1 (C0, C2, C3 未定义) | | FCMOVNE ST(0), ST(1) |
| | FCMOVNBE ST(0), ST(i) | DB D0+i | 不小于等于 (CF=0 and ZF=0), 设置 C1 (C0, C2, C3 未定义) | | FCMOVNBE ST(0), ST(1) |
| | FCMOVNU ST(0), ST(i) | DB D8+i | not unordered(PF=1), 设置 C1 (C0, C2, C3 未定义) | | FCMOVNBE ST(0), ST(1) |
| FCOM | FCOM m32real | D8 /2 | 设置 C1, C0, C2, C3 | 实数比较: ST(0)-m32real, 设置标志位 | FCOM REAL4 Ptr [00812CD0] |
| | FCOM m64real | DC /2 | | 实数比较: ST(0)-m64real, 设置标志位 | FCOM REAL8 Ptr [00812CD0] |
| | FCOM ST(i) | D8 D0+i | | 实数比较: ST(0)-ST(i), 设置标志位 | FCOM ST(2) |
| | FCOM | D8 D1 | | 实数比较: ST(0)-ST(1), 设置标志位 | FCOM |
| FCOMP | FCOMP m32real | D8 /3 | 设置 C1, C0, C2, C3 | 实数比较: ST(0)-m32real, 设置标志位, 执行一次出栈操作 | FCOMP REAL4 Ptr [00812CD0] |

| | | | | | |
|---------|-----------------------|---------|---------------------------|---|-------------------------------|
| | FCOMP m64real | DC /3 | | 实数比较: ST(0)- m64real, 设置标志位, 执行一次出栈操作 | FCOMP REAL8 Ptr [00812CD0] |
| | FCOMP ST(i) | D8 D8+i | | 实数比较: ST(0)-ST(i), 设置标志位, 执行一次出 栈操作 | FCOMP ST(2) |
| | FCOMP | D8 D9 | | 实数比较: ST(0)-ST(1), 设置标志位, 执行一次出 栈操作 | FCOMP |
| FCOMI | FCOMI ST, ST(i) | DB F0+i | 设置 ZF, PF, CF, C1 | 实数比较: ST(0)-ST(i), 设置标志位 | FCOMI ST, ST(1) |
| FCOMIP | FCOMIP ST, ST(i) | DF F0+i | 设置 ZF, PF, CF, C1 | 实数比较: ST(0)-ST(i), 设置标志位, 执行一次出 栈操作 | FCOMIP ST, ST(1) |
| FUCOMI | FUCOMI ST, ST(i) | DB E8+i | 设置 ZF, PF, CF, C1 | 实数比较: ST(0)-ST(i), 检查 ordered 值, 设置标 志位 | FCOMIP ST, ST(1) |
| FUCOMIP | FUCOMIP ST, ST(i) | DF E8+i | 设置 ZF, PF, CF, C1 | 实数比较: ST(0)-ST(i), 检查 ordered 值, 设置标 志位, 执行一次出栈操作 | FUCOMIP ST, ST(1) |
| FCOS | FCOS | D9 FF | 设置 C1, C2 | 余弦函数 COS, ST(0)← cosine(ST(0)) | FCOS |
| FDECSTP | FDECSTP | D9 F6 | 设置 C1 (C0, C2, C3 未定义) | 将 FPU 的栈顶指针值减 小 1 | FDECSTP |
| FDIV | FDIV m32real | D8 /6 | 设置 C1 (C0, C2, C3 未定义) | 实数除法: ST(0)← ST(0)/m32real | FDIV REAL4 Ptr [00459AF0] |
| | FDIV m64real | DC /6 | | 实数除法: ST(0)← ST(0)/m64real | FDIV REAL8 Ptr [00459AF0] |
| | FDIV ST(0), ST(i) | D8 F0+i | | 实数除法: ST(0)← ST(0)/ST(i) | FDIV ST(0), ST(1) |
| | FDIV ST(i), ST(0) | DC F8+i | | 实数除法: ST(i)← ST(i)/ST(0) | FDIV ST(1), ST(0) |
| FDIVP | FDIVP ST(i), ST(0) | DE F8+i | 设置 C1 (C0, C2, C3 未定义) | 实数除法: ST(i)← ST(i)/ST(0), 执行一次 出栈操作 | FDIVP ST(1), ST(0) |
| | FDIVP | DE F9 | | 实数除法: ST(1)← ST(1)/ST(0), 执行一次 出栈操作 | FDIVP |
| FIDIV | FIDIV m32int | DA /6 | 设置 C1 (C0, C2, C3 未定义) | 除以整数: ST(0)← ST(0)/m32int | FIDIV DWORD Ptr [00459AF0] |
| | FIDIV m16int | DE /6 | | 除以整数: ST(0)← ST(0)/m16int | FIDIV WORD Ptr [00459AF0] |
| FDIVR | FDIVR m32real | D8 /7 | 设置 C1 (C0, C | 实数除法: ST(0)←m32r | FDIVR REAL4 Ptr |

| | | | | | |
|-------------|---------------------|-----------------------|------------------------|---|-----------------------------|
| | | | 2, C3 未定义) | real/ST(0) | [00459AF0] |
| | FDIVR m64real | DC /7 | | 实数除法: $ST(0) \leftarrow m64real/ST(0)$ | FDIVR REAL8 Ptr [00459AF0] |
| | FDIVR ST(0), ST(i) | D8 F8+i | | 实数除法: $ST(0) \leftarrow ST(i)/ST(0)$ | FDIVR ST(0), ST(1) |
| | FDIVR ST(i), ST(0) | DC F0+i | | 实数除法: $ST(i) \leftarrow ST(0)/ST(i)$ | FDIVR ST(1), ST(0) |
| FDIVRP | FDIVRP ST(i), ST(0) | DE F0+i | 设置 C1 (C0, C2, C3 未定义) | 实数除法: $ST(i) \leftarrow ST(0)/ST(i)$, 执行一次出栈操作 | FDIVRP ST(1), ST(0) |
| | FDIVRP | DE F1 | | 实数除法: $ST(1) \leftarrow ST(0)/ST(1)$, 执行一次出栈操作 | FDIVRP |
| FIDIVR | FIDIVR m32int | DA /7 | 设置 C1 (C0, C2, C3 未定义) | 被整数除: $ST(0) \leftarrow m32int/ST(0)$ | FIDIVR DWORD Ptr [00459AF0] |
| | FIDIVR m16int | DE /7 | | 被整数除: $ST(0) \leftarrow m16int/ST(0)$ | FIDIVR WORD Ptr [00459AF0] |
| FFREE | FFREE ST(i) | DD C0+i | (C0, C1, C2, C3 未定义) | 将与 ST(i) 相对应的标志位设置为空, 即 TAG(i) $\leftarrow 11B$ | FFREE ST(1) |
| FICOM | FICOM m16int | DE /2 | 设置 C1, C0, C2, C3 | 和整数比较: ST(0)-m16int, 设置标志 | FICOM WORD Ptr [00459AF0] |
| | FICOM m32int | DA /2 | | 和整数比较: ST(0)-m32int, 设置标志 | FICOM DWORD Ptr [00459AF0] |
| FICOMP | FICOMP m16int | DE /3 | 设置 C1, C0, C2, C3 | 和整数比较: ST(0)-m16int, 设置标志, 执行一次出栈操作 | FICOMP WORD Ptr [00459AF0] |
| | FICOMP m32int | DA /3 | | 和整数比较: ST(0)-m32int, 设置标志, 执行一次出栈操作 | FICOMP DWORD Ptr [00459AF0] |
| FILD | FILD m16int | DF /0 | 设置 C1 (C0, C2, C3 未定义) | 将 16 位整数压栈, 即装入 ST(0) | FILD WORD Ptr [00459AF0] |
| | FILD m32int | DB /0 | | 将 32 位整数压栈, 即装入 ST(0) | FILD DWORD Ptr [00459AF0] |
| | | | | | |
| FILD m64int | DF /5 | 将 64 位整数压栈, 即装入 ST(0) | | | |
| FINCSTP | FINCSTP | D9 F7 | 设置 C1 (C0, C2, C3 未定义) | 将 FPU 的栈顶指针值增大 1 | FINCSTP |
| FINIT | FINIT | 9B DB E3 | 设置 C1 (C0, C2, C3 未定义) | 初始化 FPU, 检查非屏蔽浮点异常 | FINIT |

| | | | | | |
|--------|----------------------|---------|------------------------|-------------------------------------|----------------------------|
| FNINIT | FNINIT | DB E3 | 设置 C1 (C0, C2, C3 未定义) | 初始化 FPU, 不检查非屏蔽浮点异常 | FNINIT |
| FIST | FIST m16int | DF /2 | 设置 C1 (C0, C2, C3 未定义) | 将 ST(0)以 16 位整数保存到 m16int | FIST WORD Ptr [00459AF0] |
| | FIST m32int | DB /2 | | 将 ST(0)以 32 位整数保存到 m32int | FIST DWORD Ptr [00459AF0] |
| FISTP | FISTP m16int | DF /3 | 设置 C1 (C0, C2, C3 未定义) | 将 ST(0)以 16 位整数保存到 m16int, 执行一次出栈操作 | FISTP WORD Ptr [00459AF0] |
| | FISTP m32int | DB /3 | | 将 ST(0)以 32 位整数保存到 m32int, 执行一次出栈操作 | FISTP DWORD Ptr [00459AF0] |
| | FISTP m64int | DF /7 | | 将 ST(0)以 64 位整数保存到 m64int, 执行一次出栈操作 | |
| FLD | FLD m32real | D9 /0 | 设置 C1 (C0, C2, C3 未定义) | 将 32 位实数压栈, 即装入 ST(0) | FLD REAL4 Ptr [00459AF0] |
| | FLD m64real | DD /0 | | 将 64 位实数压栈, 即装入 ST(0) | FLD REAL8 Ptr [00459AF0] |
| | FLD m80real | DB /5 | | 将 80 位实数压栈, 即装入 ST(0) | |
| | FLD ST(i) | D9 C0+i | | 将 ST(i)压栈, 即装入 ST(0) | FLD ST(7) |
| FLD1 | FLD1 | D9 E8 | 设置 C1 (C0, C2, C3 未定义) | 将+1.0 压栈, 即装入 ST(0) | FLD1 |
| FLDL2T | FLDL2T | D9 E9 | 设置 C1 (C0, C2, C3 未定义) | 将 $\log_2(10)$ 压栈, 即装入 ST(0) | FLDL2T |
| FLDL2E | FLDL2E | D9 EA | 设置 C1 (C0, C2, C3 未定义) | 将 $\log_2(e)$ 压栈, 即装入 ST(0) | FLDL2E |
| FLDPI | FLDPI | D9 EB | 设置 C1 (C0, C2, C3 未定义) | 将 π 压栈, 即装入 ST(0) | FLDPI |
| FLDLG2 | FLDLG2 | D9 EC | 设置 C1 (C0, C2, C3 未定义) | 将 $\log_{10}(2)$ 压栈, 即装入 ST(0) | FLDLG2 |
| FLDLN2 | FLDLN2 | D9 ED | 设置 C1 (C0, C2, C3 未定义) | 将 $\log_e(2)$ 压栈, 即装入 ST(0) | FLDLN2 |
| FLDZ | FLDZ | D9 EE | 设置 C1 (C0, C2, C3 未定义) | 将+0.0 压栈, 即装入 ST(0) | FLDZ |
| FLDCW | FLDCW m2byte | D9 /5 | C1, C0, C2, C3 未定义 | 从 m2byte 装入 FPU 控制字 | FLDCW BYTE Ptr [00459AF0] |
| FLDENV | FLDENV m14/28byte | D9 /4 | C1, C0, C2, C3 未定义 | 从 m14/28byte 装入 FPU 环境 | FLDENV BYTE Ptr [00459AF0] |
| FMUL | FMUL m32real | D8 /1 | 设置 C1 (C0, | 实数乘法: $ST(0) \leftarrow$ | FMUL REAL4 Ptr |

| | | | | | |
|---------|-----------------------|----------|---------------------------|--|-------------------------------|
| | | | C2, C3 未定义) | ST(0)*m32real | [00459AF0] |
| | FMUL m64real | DC /1 | | 实数乘法: ST(0)← ST(0)*m64real | FMUL REAL8 Ptr [00459AF0] |
| | FMUL ST(0), ST(i) | D8 C8+i | | 实数乘法: ST(0)← ST(0)*ST(i) | FMUL ST(0), ST(1) |
| | FMUL ST(i), ST(0) | DC C8+i | | 实数乘法: ST(i)← ST(i)*ST(0) | FMUL ST(1), ST(0) |
| FMULP | FMULP ST(i), ST(0) | DE C8+i | 设置 C1 (C0, C2, C3 未定义) | 实数乘法: ST(i)← ST(i)*ST(0), 执行一次 出栈操作 | FMULP ST(1), ST(0) |
| | FMULP | DE C9 | | 实数乘法: ST(1)← ST(1)*ST(0), 执行一次 出栈操作 | FMULP |
| FIMUL | FIMUL m32int | DA /1 | 设置 C1 (C0, C2, C3 未定义) | 乘以整数: ST(0)← m32int*ST(0) | FIMUL DWORD Ptr [00459AF0] |
| | FIMUL m16int | DE /1 | | 乘以整数: ST(0)← m16int*ST(0) | FIMUL WORD Ptr [00459AF0] |
| FNOP | FNOP | D9 D0 | C1, C0, C2, C3 未定义 | 空操作(等同于 NOP) | FNOP |
| FPATAN | FPATAN | D9 F3 | 设置 C1 (C0, C2, C3 未定义) | 反正切函数 arctan, ST(1)← arctan(ST(1)/ST(0)), 执行一次出栈操作 | FPATAN |
| FPREM | FPREM | D9 F8 | 设置 C1, C0, C2, C3 | 取余数, ST(0)←ST(0) MOD ST(1) | FPREM |
| FPREM1 | FPREM1 | D9 F5 | 设置 C1, C0, C2, C3 | 取余数(使用 IEEE 标准), ST(0)←ST(0) MOD ST(1) | FPREM1 |
| FATAN | FATAN | D9 F2 | 设置 C1, C2 (C0, C3 未定义) | 正切函数 Tan, ST(0)← tangent(ST(0)), 执行一 次压栈 1.0 的操作 | FATAN |
| FRNDINT | FRNDINT | D9 FC | 设置 C1 (C0, C2, C3 未定义) | 取整(四舍五入, 当小数部 分刚好等于 0.5 时: 如果 整数部分为偶数, 则“舍”; 如果整数部分为奇数, 则 “入”), ST(0)← Round(ST(0)) | FRNDINT |
| FRSTOR | FRSTOR m94/108byte | DD /4 | 设置 C0, C1, C2, C3 | 从 m94/108byte 装入 FPU 状态 | FRSTOR BYTE Ptr [00459AF0] |
| FSAVE | FSAVE m94/108byte | 9B DD /6 | 设置 C0, C1, C2, C3 | 将 FPU 状态保存在 m94/108byte 中, 检查 非屏蔽浮点异常, 然后初 始化 FPU | FSAVE BYTE Ptr [00459AF0] |

| | | | | | |
|---------|-----------------------|----------|------------------------|---|-----------------------------|
| FNSAVE | FNSAVE m94/108byte | 9B DD /6 | 设置 C0, C1, C2, C3 | 将 FPU 状态保存在 m94/108byte 中, 不检查非屏蔽浮点异常, 然后初始化 FPU | FNSAVE BYTE Ptr [00459AF0] |
| FSCALE | FSCALE | D9 FD | 设置 C1 (C0, C2, C3 未定义) | $ST(0) \leftarrow ST(0) * 2^{ST(1)}$ | FSCALE |
| FSIN | FSIN | D9 FE | 设置 C1, C2 (C0, C3 未定义) | 正弦函数 Sin, $ST(0) \leftarrow \text{sine}(ST(0))$ | FSIN |
| FSINCOS | FSINCOS | D9 FB | 设置 C1, C2 (C0, C3 未定义) | SinCos 函数: $ST(0) \leftarrow \text{sine}(ST(0))$, PUSH cosine(ST(0)) | FSINCOS |
| FSQRT | FSQRT | D9 FA | 设置 C1 (C0, C2, C3 未定义) | 平方根函数: $ST(0) \leftarrow \text{SQRT}(ST(0))$ | FSQRT |
| FST | FST m32real | D9 /2 | 设置 C1 (C0, C2, C3 未定义) | 将 ST(0) 复制到 m32real | FST REAL4 Ptr [00459AF0] |
| | FST m64real | DD /2 | | 将 ST(0) 复制到 m64real | FST REAL8 Ptr [00459AF0] |
| | FST ST(i) | DD D0+i | | 将 ST(0) 复制到 ST(i) | FST ST(3) |
| FSTP | FSTP m32real | D9 /3 | 设置 C1 (C0, C2, C3 未定义) | 将 ST(0) 复制到 m32real, 执行一次出栈操作 | FSTP REAL4 Ptr [00459AF0] |
| | FSTP m64real | DD /3 | | 将 ST(0) 复制到 m64real, 执行一次出栈操作 | FSTP REAL8 Ptr [00459AF0] |
| | FSTP m80real | DB /7 | | 将 ST(0) 复制到 m80real, 执行一次出栈操作 | |
| | FSTP ST(i) | DD D8+i | | 将 ST(0) 复制到 ST(i), 执行一次出栈操作 | FSTP ST(3) |
| FSTCW | FSTCW m2byte | 9B D9 /7 | C0, C1, C2, C3 未定义 | 将 FPU 控制字保存到 m2byte, 检查非屏蔽浮点异常 | FSTCW BYTE Ptr [00459AF0] |
| FNSTCW | FNSTCW m2byte | 9B D9 /7 | C0, C1, C2, C3 未定义 | 将 FPU 控制字保存到 m2byte, 不检查非屏蔽浮点异常 | FNSTCW BYTE Ptr [00459AF0] |
| FSTENV | FSTENV m14/28byte | 9B D9 /6 | C0, C1, C2, C3 未定义 | 将 FPU 环境保存到 m14/28byte, 检查非屏蔽浮点异常, 然后屏蔽所有浮点异常 | FSTENV BYTE Ptr [00459AF0] |
| FNSTENV | FNSTENV m14/28byte | D9 /6 | C0, C1, C2, C3 未定义 | 将 FPU 环境字保存到 m14/28byte, 不检查非屏蔽浮点异常, 然后屏蔽 | FNSTENV BYTE Ptr [00459AF0] |

| | | | | | |
|--------|---------------------|----------|------------------------|--|----------------------------|
| | | | | 所有浮点异常 | |
| FSTSW | FSTSW m2byte | 9B DD /7 | C0, C1, C2, C3 未定义 | 将 FPU 状态字保存到 m2byte, 检查非屏蔽浮点异常 | FSTSW BYTE Ptr [00459AF0] |
| | FSTSW AX | 9B DF E0 | | 将 FPU 状态字保存到 AX, 检查非屏蔽浮点异常 | FSTSW AX |
| FNSTSW | FNSTSW m2byte | DD /7 | C0, C1, C2, C3 未定义 | 将 FPU 状态字保存到 m2byte, 不检查非屏蔽浮点异常 | FNSTSW BYTE Ptr [00459AF0] |
| | FNSTSW AX | DF E0 | | 将 FPU 状态字保存到 AX, 不检查非屏蔽浮点异常 | FNSTSW AX |
| FSUB | FSUB m32real | D8 /4 | 设置 C1 (C0, C2, C3 未定义) | 实数减法: $ST(0) \leftarrow ST(0) - m32real$ | FSUB REAL4 Ptr [00459AF0] |
| | FSUB m64real | DC /4 | | 实数减法: $ST(0) \leftarrow ST(0) - m64real$ | FSUB REAL8 Ptr [00459AF0] |
| | FSUB ST(0), ST(i) | D8 E0-i | | 实数减法: $ST(0) \leftarrow ST(0) - ST(i)$ | FSUB ST(0), ST(1) |
| | FSUB ST(i), ST(0) | DC E8-i | | 实数减法: $ST(i) \leftarrow ST(i) - ST(0)$ | FSUB ST(1), ST(0) |
| FSUBP | FSUBP ST(i), ST(0) | DE E8-i | 设置 C1 (C0, C2, C3 未定义) | 先进行实数减法: $ST(i) \leftarrow ST(i) - ST(0)$, 然后进行一次出栈操作 | FSUBP ST(1), ST(0) |
| | FSUBP | DE E9 | | 先进行实数减法: $ST(1) \leftarrow ST(1) - ST(0)$, 然后进行一次出栈操作 | FSUBP |
| FISUB | FISUB m32int | DA /4 | 设置 C1 (C0, C2, C3 未定义) | 减整数: $ST(0) \leftarrow ST(0) - m32int$ | FISUB DWORD Ptr [00812CD0] |
| | FISUB m16int | DE /4 | | 减整数: $ST(0) \leftarrow ST(0) - m16int$ | FISUB WORD Ptr [00812CD0] |
| FSUbr | FSUbr m32real | D8 /5 | 设置 C1 (C0, C2, C3 未定义) | 实数减法: $ST(0) \leftarrow m32real - ST(0)$ | FSUbr REAL4 Ptr [00459AF0] |
| | FSUbr m64real | DC /5 | | 实数减法: $ST(0) \leftarrow m64real - ST(0)$ | FSUbr REAL8 Ptr [00459AF0] |
| | FSUbr ST(0), ST(i) | D8 E8+i | | 实数减法: $ST(0) \leftarrow ST(i) - ST(0)$ | FSUbr ST(0), ST(1) |
| | FSUbr ST(i), ST(0) | DC E0+i | | 实数减法: $ST(i) \leftarrow ST(0) - ST(i)$ | FSUbr ST(1), ST(0) |
| FSUbrP | FSUbrP ST(i), ST(0) | DE E0+i | 设置 C1 (C0, C2, C3 未定义) | 实数减法: $ST(i) \leftarrow ST(0) - ST(i)$, 执行一次出栈操作 | FSUbrP ST(1), ST(0) |
| | FSUbrP | DE E1 | | 实数减法: $ST(1) \leftarrow$ | FSUbrP |

| | | | | | |
|---------|------------------|----------|------------------------|---|-----------------------------|
| | | | | ST(0)-ST(1), 执行一次出栈操作 | |
| FISUbr | FISUbr m32int | DA /5 | 设置 C1 (C0, C2, C3 未定义) | 被整数减: ST(0)←m32int-ST(0) | FISUbr DWORD Ptr [00459AF0] |
| | FISUbr m16int | DE /5 | | 被整数减: ST(0)←m16int-ST(0) | FISUbr WORD Ptr [00459AF0] |
| FTST | FTST | D9 E4 | 设置 C0, C1, C2, C3 | 零检测, 将 ST(0)和 0.0 相比较 | FTST |
| FUCOM | FUCOM ST(i) | DD E0+i | 设置 C0, C1, C2, C3 | 比较 ST(0)和 ST(i) | FUCOM ST(4) |
| | FUCOM | DD E1 | | 比较 ST(0)和 ST(1) | FUCOM |
| FUCOMP | FUCOMP ST(i) | DD E8+i | 设置 C0, C1, C2, C3 | 比较 ST(0)和 ST(i), 执行一次出栈操作 | FUCOMP ST(4) |
| | FUCOMP | DD E9 | | 比较 ST(0)和 ST(1), 执行一次出栈操作 | FUCOMP |
| FUCOMPP | FUCOMPP | DD E8+i | 设置 C0, C1, C2, C3 | 比较 ST(0)和 ST(1), 执行两次出栈操作 | FUCOMPP |
| FWAIT | FWAIT | 9B | C0, C1, C2, C3 未定义 | 等待, 检查非屏蔽浮点异常 | FWAIT |
| FXAM | FXAM | D9 E5 | 设置 C0, C1, C2, C3 | 检查 ST(0)中的数值类型 | FXAM |
| FXCH | FXCH ST(i) | D9 C8+i | 设置 C1 (C0, C2, C3 未定义) | 交换 ST(0)和 ST(i)的内容 | FXCH ST(4) |
| | FXCH | D9 C9 | | 交换 ST(0)和 ST(1)的内容 | FXCH |
| FXRSTOR | FXRSTOR m512byte | 0F AE /1 | 恢复所有标志位 | 由 m512byte 恢复寄存器 x87 FPU, MMX, XMM, 和 MXCSR 的状态 | FXRSTOR BYTE Ptr [00459AF0] |
| FXSAVE | FXSAVE m512byte | 0F AE /0 | 不影响标志位 | 将寄存器 x87 FPU, MMX, XMM, 和 MXCSR 的状态保存到 m512byte | FXSAVE BYTE Ptr [00459AF0] |
| FXtrACT | FXtrACT | D9 F4 | 设置 C1 (C0, C2, C3 未定义) | 将 ST(0)中的数分成 exponent 和 significand 两部分, ST(0)←Exponent(ST(0)), PUSH Significand(ST(0)) | FXtrACT |
| FYL2X | FYL2X | D9 F1 | 设置 C1 (C0, C2, C3 未定义) | 计算: ST(1)←ST(1)*log2(ST(0)), 执行一次出栈操作 | FYL2X |
| FYL2XP1 | FYL2XP1 | D9 F9 | 设置 C1 (C0, | 计算: ST(1) ← | FYL2XP1 |

| | | | | | |
|------|------------------------|----------|-------------------------------|---|--|
| | | | C2, C3 未定义) | ST(1)*log2(ST(0) + 1.0), 执行一次出栈操作 | |
| | HLT | F4 | 不影响标志位 | 系统进入暂停状态 | HLT |
| | ID IV r/m8 | F6 /7 | AF CF OF PF SF ZF 未定义 | 有符号除法 | ID IV BL ; AX 除以 BL, 商在 AL 中, 余数在 AH 中 |
| | ID IV r/m16 | F7 /7 | | | ID IV BX ; DX: AX 除以 BX, 商在 AX 中, 余数在 DX 中 |
| | ID IV r/m32 | F7 /7 | | | ID IV EBX; EDX: EAX 除以 BX, 商在 EAX 中, 余数在 EDX 中 |
| IMUL | IMUL r/m8 | F6 /5 | 设置 CF OF (SF ZF AF PF 未定义) | 有符号乘法: $AX \leftarrow AL * r/m8$ | IMUL CL |
| | IMUL r/m16 | F7 /5 | | 有符号乘法: $DX:AX \leftarrow AX * r/m16$ | IMUL CX |
| | IMUL r/m32 | F7 /5 | | 有符号乘法: $EDX:EAX \leftarrow EAX * r/m32$ | IMUL ECX |
| | IMUL r16, r/m16 | 0F AF /r | | 有符号乘法: $r16 \leftarrow r16 * r/m16$ | IMUL AX, BX |
| | IMUL r32, r/m32 | 0F AF /r | | 有符号乘法: $r32 \leftarrow r32 * r/m32$ | IMUL EAX, EBX |
| | IMUL r16, r/m16, imm8 | 6B /r ib | | 有符号乘法: $r16 \leftarrow r/m16 * imm8$ | IMUL AX, BX, 39 |
| | IMUL r32, r/m32, imm8 | 6B /r ib | | 有符号乘法: $r32 \leftarrow r/m32 * imm8$ | IMUL EAX, EBX, 39 |
| | IMUL r16, imm8 | 6B /r ib | | 有符号乘法: $r16 \leftarrow r16 * imm8$ | IMUL AX, 37 |
| | IMUL r32, imm8 | 6B /r ib | | 有符号乘法: $r32 \leftarrow r32 * imm8$ | IMUL EAX, 37 |
| | IMUL r16, r/m16, imm16 | 69 /r iw | | 有符号乘法: $r16 \leftarrow r/m16 * imm16$ | IMUL AX, BX, 387E |
| | IMUL r32, r/m32, imm32 | 69 /r id | | 有符号乘法: $r32 \leftarrow r/m32 * imm32$ | IMUL EAX, EBX, 006387EA |
| | IMUL r16, imm16 | 69 /r iw | | 有符号乘法: $r16 \leftarrow r16 * imm16$ | IMUL AX, 387E |
| | IMUL r32, imm32 | 69 /r id | | 有符号乘法: $r32 \leftarrow r32 * imm32$ | IMUL EAX, 387E |

| | | | | | |
|----------|-------------|---------|----------------------|--------------------------------|-----------------------------|
| | imm32 | | | r32*imm32 | 006387EA |
| IN | IN AL,imm8 | E4 ib | 不影响标志位 | 从 imm8 指定的端口读字节(BYTE)到 AL | IN AL,E0 |
| | IN AX,imm8 | E5 ib | | 从 imm8 指定的端口读字节(BYTE)到 AX | IN AX,E0 |
| | IN EAX,imm8 | E5 ib | | 从 imm8 指定的端口读字节(BYTE)到 EAX | IN EAX,E0 |
| | IN AL,DX | EC | | 从 DX 指定的端口读字节(BYTE)到 AL | IN AL,DX |
| | IN AX,DX | ED | | 从 DX 指定的端口读字(WORD)到 AX | IN AX,DX |
| | IN EAX,DX | ED | | 从 DX 指定的端口读双字(DWORD)到 EAX | IN EAX,DX |
| INC | INC r/m8 | FE /0 | 设置 OF SF ZF AF PF | 目标加 1 | INC BYTE Ptr [00459AF0] |
| | INC r/m16 | FF /0 | | | INC WORD Ptr [00459AF0] |
| | INC r/m32 | FF /0 | | | INC DWORD Ptr [00459AF0] |
| | INC r16 | 40+rw | | | INC DX |
| | INC r32 | 40+rd | | | INC EDX |
| INS | INS m8, DX | 6C | 不影响标志位 | 从 DX 指定的端口读字节(BYTE)到 ES:(E)DI | |
| | INS m16, DX | 6D | | 从 DX 指定的端口读字(WORD)到 ES:(E)DI | |
| | INS m32, DX | 6D | | 从 DX 指定的端口读双字(DWORD)到 ES:(E)DI | |
| INSB | INSB | 6C | 不影响标志位 | 从 DX 指定的端口读字节(BYTE)到 ES:(E)DI | INSB |
| INSW | INSW | 6D | 不影响标志位 | 从 DX 指定的端口读字(WORD)到 ES:(E)DI | INSW |
| INSD | INSD | 6D | 不影响标志位 | 从 DX 指定的端口读双字(DWORD)到 ES:(E)DI | INSD |
| INT 3 | INT 3 | CC | 标志位被保存压栈 | 调试陷阱中断 | INT 3 |
| INT imm8 | INT imm8 | CD ib | 标志位被保存压栈 | 中断功能调用, imm8 是中断向量号 | INT 21 |
| INTO | INTO | CE | 标志位被保存压栈 | 溢出中断, 条件: OV=1, 4 号中断功能调用 | INTO |
| INVD | INVD | 0F 08 | 不影响标志位 | 使内部 CACHES 无效, 初始化外部 CACHES | INVD |
| INVLPG | INVLPG | 0F 01/7 | 不影响标志位 | 使 TLB 项无效 | INVLPG [EBX] |

| | | | | | |
|-------|------------|-------|-----------------------|------------|-------|
| IRET | IRET | CF | 恢复被压栈的标志位 | 中断返回(16 位) | IRET |
| IREtd | IREtd | CF | 恢复被压栈的标志位 | 中断返回(32 位) | IREtd |
| | JA rel8 | 77 cb | 高于(CF=0 and ZF=0) | 条件转移指令 | |
| | JAE rel8 | 73 cb | 高于等于(CF=0) | | |
| | JB rel8 | 72 cb | 低于(CF=1) | | |
| | JBE rel8 | 76 cb | 低于等于(CF=1 or ZF=1) | | |
| | JC rel8 | 72 cb | 有进位(CF=1) | | |
| | JCXZ rel8 | E3 cb | CX=0 则跳 | | |
| | JECXZ rel8 | E3 cb | ECX=0 则跳 | | |
| | JE rel8 | 74 cb | 等于(ZF=1) | | |
| | JG rel8 | 7F cb | 大于(ZF=0 and SF=OF) | | |
| | JGE rel8 | 7D cb | 大于等于(SF=OF) | | |
| | JL rel8 | 7C cb | 小于(SF<>OF) | | |
| | JLE rel8 | 7E cb | 小于等于(ZF=1 or SF<>OF) | | |
| | JNA rel8 | 76 cb | 不高于(CF=1 or ZF=1) | | |
| | JNAE rel8 | 72 cb | 不高等于(CF=1) | | |
| | JNB rel8 | 73 cb | 不低于(CF=0) | | |
| | JNBE rel8 | 77 cb | 不低等于(CF=0 and ZF=0) | | |
| | JNC rel8 | 73 cb | 无进位(CF=0) | | |
| | JNE rel8 | 75 cb | 不等于(ZF=0) | | |
| | JNG rel8 | 7E cb | 不大于(ZF=1 or SF<>OF) | | |
| | JNGE rel8 | 7C cb | 不大等于(SF<>OF) | | |
| | JNL rel8 | 7D cb | 不小于(SF=OF) | | |
| | JNLE rel8 | 7F cb | 不小于等于(ZF=0 and SF=OF) | | |
| | JNO rel8 | 71 cb | 无溢出(OF=0) | | |
| | JNP rel8 | 7B cb | 非偶数(PF=0) | | |
| | JNS rel8 | 79 cb | 非负数(SF=0) | | |
| | JNZ rel8 | 75 cb | 非零(ZF=0) | | |
| | JO rel8 | 70 cb | 溢出(OF=1) | | |

| | | | | | |
|-----|----------------|-------------|--------------------|--|--------------|
| | JP rel8 | 7A cb | 偶数(PF=1) | | |
| | JPE rel8 | 7A cb | 偶数(PF=1) | | |
| | JPO rel8 | 7B cb | 奇数(PF=0) | | |
| | JS rel8 | 78 cb | 负数(SF=1) | | |
| | JZ rel8 | 74 cb | 为零(ZF=1) | | |
| | JA rel16/32 | 0F 87 cw/cd | 高于(CF=0 and ZF=0) | | |
| | JAЕ rel16/32 | 0F 83 cw/cd | 高于等于(CF=0) | | |
| | JB rel16/32 | 0F 82 cw/cd | 低于(CF=1) | | |
| | JBE rel16/32 | 0F 86 cw/cd | 低于等于(CF=1 or ZF=1) | | |
| | JC rel16/32 | 0F 82 cw/cd | 有进位(CF=1) | | |
| | JE rel16/32 | 0F 84 cw/cd | 等于(ZF=1) | | |
| | JZ rel16/32 | 0F 84 cw/cd | 为零(ZF=1) | | |
| | JG rel16/32 | 0F 8F cw/cd | 大于(ZF=0 and SF=0F) | | |
| JMP | JMP rel8 | EB cb | 不影响标志位 | 无条件转移指令 | |
| | JMP rel16 | E9 cw | | | |
| | JMP rel32 | E9 cd | | | |
| | JMP r/m16 | FF /4 | | | |
| | JMP r/m32 | FF /4 | | | |
| | JMP ptr16:16 | EA cd | | | |
| | JMP ptr16:32 | EA cp | | | |
| | JMP m16:16 | FF /5 | | | |
| | JMP m16:32 | FF /5 | | | |
| | LAHF | 9F | 不影响标志位 | 将标志寄存器的低字节送入 AH, 格式为: AH=(SF:ZF:0:AF:0:PF:1:CF) | LAHF |
| LAR | LAR r16,r/m16 | 0F 02 /r | 设置 ZF | 将源操作数指定的选择子访问权送目标寄存器 | LAR AX, BX |
| | LAR r32,r/m32 | | | | LAR EAX, EBX |
| LDS | LDS r16,m16:16 | C5 /r | 不影响标志位 | 从内存位置 m16:16 装入地址值 DS:r16 | |
| | LDS r32,m16:32 | | | 从内存位置 m16:32 装入地址值 DS:r32 | |
| LSS | LSS r16,m16:16 | 0F B2 /r | 不影响标志位 | 从内存位置 m16:16 装入地址值 SS:r16 | |
| | LSS r32,m16:32 | | | 从内存位置 m16:32 装入地址值 SS:r32 | |
| LES | LES r16,m16:16 | C4 /r | 不影响标志位 | 从内存位置 m16:16 装入地址值 ES:r16 | |

| | | | | | |
|-------|-------------------|----------|--------|---------------------------------|------------|
| | LES r32,m16:32 | | | 从内存位置 m16:32 装入地址值 ES:r32 | |
| LFS | LFS r16,m16:16 | 0F B4 /r | 不影响标志位 | 从内存位置 m16:16 装入地址值 FS:r16 | |
| | LFS r32,m16:32 | | | 从内存位置 m16:32 装入地址值 FS:r32 | |
| LGS | LGS r16,m16:16 | 0F B5 /r | 不影响标志位 | 从内存位置 m16:16 装入地址值 GS:r16 | |
| | LGS r32,m16:32 | | | 从内存位置 m16:32 装入地址值 GS:r32 | |
| LEA | LEA r16,m | 8D /r | 不影响标志位 | 将源操作数的有效地址送 r16 | |
| | LEA r32,m | | | 将源操作数的有效地址送 r32 | |
| LEAVE | LEAVE | C9 | 不影响标志位 | 子过程返回释放堆栈框架 | LEAVE |
| LGDT | LGDT m16&32 | 0F 01 /2 | 不影响标志位 | 将内存 m16&32 处的值装入全局描述符表寄存器 | LGDT [EBP] |
| LIDT | LIDT m16&32 | 0F 01 /3 | 不影响标志位 | 将内存 m16&32 处的值装入中断描述符表寄存器 | LIDT [SI] |
| LLDT | LLDT r/m16 | 0F 00 /2 | 不影响标志位 | 将 r/m16 处的值装入局部描述符表寄存器 | LLDT BX |
| LMSW | LMSW r/m16 | 0F 01 /6 | 不影响标志位 | 将 r/m16 处的值装入机器状态字 | LMSW [BP] |
| LOCK | LOCK | F0 | 不影响标志位 | 总线锁定 | LOCK |
| LODS | LODS m8 | AC | 不影响标志位 | 将地址 DS:(E)SI 的值装入 AL | |
| | LODS m16 | AD | | 将地址 DS:(E)SI 的值装入 AX | |
| | LODS m32 | AD | | 将地址 DS:(E)SI 的值装入 EAX | |
| LODSB | LODSB | AC | 不影响标志位 | 将地址 DS:(E)SI 的值装入 AL | LODSB |
| LODSW | LODSW | AD | 不影响标志位 | 将地址 DS:(E)SI 的值装入 AX | LODSW |
| LODSD | LODSD | AD | 不影响标志位 | 将地址 DS:(E)SI 的值装入 EAX | LODSD |
| LOOP | LOOP rel8 | E2 cb | 不影响标志位 | 条件循环指令, 循环条件 CX/ECX != 0 | |
| LOOPE | LOOPE rel8 | E1 cb | 不影响标志位 | 条件循环指令, 循环条件 CX/ECX != 0 且 ZF=1 | |
| LOOPZ | LOOPZ rel8 | E1 cb | 不影响标志位 | 条件循环指令, 循环条件 CX/ECX != 0 且 ZF=1 | |

| | | | | | |
|--------|--------------------|----------|--------|-----------------------------------|------------------------------|
| LOOPNE | LOOPNE rel8 | E1 cb | 不影响标志位 | 条件循环指令，循环条件 CX/ECX != 0 且 ZF=0 | |
| LOOPNZ | LOOPNZ rel8 | E1 cb | 不影响标志位 | 条件循环指令，循环条件 CX/ECX != 0 且 ZF=0 | |
| LSL | LSL r16,r/m16 | 0F 03 /r | 设置 ZF | 装入选择子对应的描述符 的段界限 | LSL AX,BX |
| | LSL r32,r/m32 | | | | LSL EAX,EBX |
| Ltr | Ltr r/m16 | 0F 00 /3 | 不影响标志位 | 装入任务寄存器 | Ltr AX |
| | MOV r/m8,r8 | 88 /r | 不影响标志位 | 传送指令 | MOV [00459AF0],AL |
| | MOV r/m16,r16 | 89 /r | | | MOV [00459AF0],AX |
| | MOV r/m32,r32 | 89 /r | | | MOV [00459AF0],EAX |
| | MOV r8,r/m8 | 8A /r | | | MOV AL,[00459AF0] |
| | MOV r16,r/m16 | 8B /r | | | MOV AX,[00459AF0] |
| | MOV r32,r/m32 | 8B /r | | | MOV EAX,[00459AF0] |
| | MOV r/m16,Sreg | 8C /r | | | MOV AX,ES |
| | MOV Sreg,r/m16 | 8E /r | | | MOV ES,AX |
| | MOV AL,moffs8 | A0 | | | MOV AL,ES:[459A] |
| | MOV AX,moffs16 | A1 | | | MOV AX,ES:[459A] |
| | MOV EAX,moffs32 | A1 | | | MOV EAX,ES:[00459A F0] |
| | MOV moffs8,AL | A2 | | | MOV ES:[459A],AL |
| | MOV moffs16,AX | A3 | | | MOV ES:[459A],AX |
| | MOV moffs32,EAX | A3 | | | MOV ES:[00459AF0],E AX |
| | MOV r8,imm8 | B0+rb | | | MOV AL,F0 |
| | MOV r16,imm16 | B8+rw | | | MOV AX,9AF0 |
| | MOV r32,imm32 | B8+rd | | | MOV EAX,00459AF0 |

| | | | | | |
|------|---------------------------|-------------|--------|--------------------|--|
| | MOV r/m8,imm8 | C6 /0 | | | MOV BYTE Ptr [00459AF0],F0 |
| | MOV r/m16,imm16 | C7 /0 | | | MOV WORD Ptr [00459AF0],9AF0 |
| | MOV r/m32,imm32 | C7 /0 | | | MOV DWORD Ptr [00459AF0],00459AF0 |
| MOV | MOV CR0,r32 | 0F 22 /r | 不影响标志位 | 控制寄存器传送指令 | MOV CR0,EAX |
| | MOV CR2,r32 | 0F 22 /r | | | MOV CR2,EAX |
| | MOV CR3,r32 | 0F 22 /r | | | MOV CR3,EAX |
| | MOV CR4,r32 | 0F 22 /r | | | MOV CR4,EAX |
| | MOV r32, CR0 | 0F 20 /r | | | MOV EAX, CR0 |
| | MOV r32, CR2 | 0F 20 /r | | | MOV EAX, CR2 |
| | MOV r32, CR3 | 0F 20 /r | | | MOV EAX, CR3 |
| | MOV r32, CR4 | 0F 20 /r | | | MOV EAX, CR4 |
| MOV | MOV r32, DR0- DR7 | 0F 21 /r | 不影响标志位 | 调试寄存器传送指令 | MOV EAX, DR0 |
| | MOV DR0- DR7,r32 | 0F 23 /r | | | MOV DR0, EAX |
| MOVD | MOVD mm, r/m32 | 0F 6E /r | 不影响标志位 | 双字传送指令 | |
| | MOVD r/m32, mm | 0F 7E /r | | | |
| | MOVD xmm, r/m32 | 66 0F 6E /r | | | |
| | MOVD r/m32, xmm | 66 0F 7E /r | | | |
| MOVQ | MOVQ mm, r/m64 | 0F 6F /r | 不影响标志位 | 八字节传送指令 | |
| | MOVQ mm/m64, mm | 0F 7F /r | | | |
| | MOVQ xmm1, xmm2/m64 | F3 0F 7E | | | |
| | MOVQ xmm2/m64, xmm1 | 66 0F D6 | | | |
| MOVS | MOVS m8, m8 | A4 | 不影响标志位 | 字符串传送，每次传送1 个字节 | MOVS StrING1, StrING2；源串 DS:(E)SI，目的串： ES:(E)DI |
| | MOVS m16, m | A5 | | 字符串传送，每次传送1 | MOVS StrING1, |

| | | | | | |
|-------|--------------------|----------|----------------------------------|--|--|
| | 16 | | | 个字 | StrING2 ; 源串 DS:(E)SI, 目的串: ES:(E)DI |
| | MOVS m32, m32 | A5 | | 字符串传送, 每次传送1 个双字 | MOVS StrING1, StrING2 ; 源串 DS:(E)SI, 目的串: ES:(E)DI (386+) |
| | MOVSB | A4 | | 字符串传送, 每次传送1 个字节 | MOVSB ; 源串 DS:(E)SI, 目的串: ES:(E)DI |
| | MOVSW | A5 | | 字符串传送, 每次传送1 个字 | MOVSW ; 源串 DS:(E)SI, 目的串: ES:(E)DI |
| | MOVSD | A5 | | 字符串传送, 每次传送1 个双字 | MOVSD ; 源串 DS:(E)SI, 目的串: ES:(E)DI (386+) |
| MOVSX | MOVSX r16,r/m8 | 0F BE /r | 不影响标志位 | 带符号扩展传送指令 | MOVSX AX, BL |
| | MOVSX r32,r/m8 | 0F BE /r | | | MOVSX EAX, BL |
| | MOVSX r32,r/m16 | 0F BF /r | | | MOVSX EAX, BX |
| MOVZX | MOVZX r16,r/m8 | 0F B6 /r | 不影响标志位 | 零扩展传送指令 | MOVZX AX, BL |
| | MOVZX r32,r/m8 | 0F B6 /r | | | MOVZX EAX, BL |
| | MOVZX r32,r/m16 | 0F B7 /r | | | MOVZX EAX, BX |
| MUL | MUL r/m8 | F6 /4 | 设置 CF OF (SF ZF AF PF 未定义) | 无符号乘法: $AX \leftarrow$ $AL * r/m8$ | MUL CL |
| | MUL r/m16 | F7 /4 | | 无符号乘法: $DX:AX \leftarrow$ $AX * r/m16$ | MUL CX |
| | MUL r/m32 | F7 /4 | | 无符号乘法: $EDX:EAX$ $\leftarrow EAX * r/m32$ | MUL ECX |
| | NEG r/m8 | F6 /3 | 设置 CF OF SF ZF AF PF | 取负: $r/m8 = -r/m8$ | NEG CL |
| | NEG r/m16 | F7 /3 | | 取负: $r/m16 = -r/m16$ | NEG CX |
| | NEG r/m32 | F7 /3 | | 取负: $r/m32 = -r/m32$ | NEG ECX |
| NOP | NOP | 90 | 不影响标志位 | 空操作 | NOP |
| NOT | NOT r/m8 | F6 /2 | 设置 CF OF SF ZF AF PF | 按位取反: $r/m8 = \text{NOT}$ $r/m8$ | NOT CL |
| | NOT r/m16 | F7 /2 | | 按位取反: $r/m16 = \text{NOT}$ $r/m16$ | NOT CX |

| | | | | | |
|-----|-------------------|----------|----------------------|-----------------------|---|
| | NOT r/m32 | F7 /2 | | 按位取反: r/m32=NOT r/m32 | NOT ECX |
| | OR AL, imm8 | 0C ib | 设置 CF OF PF SF ZF | 逻辑或 | OR AL, 1F |
| | OR AX, imm16 | 0D iw | | | OR AX, 4F80 |
| | OR EAX, imm32 | 0D id | | | OR EAX, 00004F80 |
| | OR r/m8, imm8 | 80 /1 ib | | | OR BYTE Ptr [006387EA], 39 |
| | OR r/m16,imm16 | 81 /1 iw | | | OR WORD Ptr [006387EA], 1039 |
| | OR r/m32,imm32 | 81 /1 id | | | OR DWORD Ptr [006387EA], 00001039 |
| | OR r/m16,imm8 | 83 /1 ib | | | OR WORD Ptr [006387EA], 39 |
| | OR r/m32,imm8 | 83 /1 ib | | | OR DWORD Ptr [006387EA], 39 |
| | OR r/m8,r8 | 08 /r | | | OR [006387EA], AL |
| | OR r/m16,r16 | 09 /r | | | OR [006387EA], AX |
| | OR r/m32,r32 | 09 /r | | | OR [006387EA], EAX |
| | OR r8,r/m8 | 0A /r | | | OR AL, [006387EA] |
| | OR r16,r/m16 | 0B /r | | | OR AX, [006387EA] |
| | OR r32,r/m32 | 0B /r | | | OR EAX, [006387EA] |
| OUT | OUT imm8,AL | E6 ib | 不影响标志位 | 将 AL 输出到 imm8 指定的端口 | OUT E0,AL |
| | OUT imm8,AX | E7 ib | | 将 AX 输出到 imm8 指定的端口 | OUT E0,AX |
| | OUT imm8,EAX | E7 ib | | 将 EAX 输出到 imm8 指定的端口 | OUT E0,EAX |
| | OUT AL,DX | EE | | 将 AL 输出到 DX 指定的端口 | OUT DX,AL |
| | OUT AX,DX | EF | | 将 AX 输出到 DX 指定的端口 | OUT DX,AX |
| | OUT EAX,DX | EF | | 将 EAX 输出到 DX 指定的端口 | OUT DX,EAX |

| | | | | | |
|----------|--------------------------------|-------------|--------|-----------------------------|------------------------|
| OUTS | OUTS DX,m8 | 6E | 不影响标志位 | 将 DS:(E)SI 处的字节输出到 DX 指定的端口 | |
| | OUTS DX,m16 | 6F | | 将 DS:(E)SI 处的字输出到 DX 指定的端口 | |
| | OUTS DX,m32 | 6F | | 将 DS:(E)SI 处的双字输出到 DX 指定的端口 | |
| OUTSB | OUTSB | 6E | 不影响标志位 | 将 DS:(E)SI 处的字节输出到 DX 指定的端口 | OUTSB |
| OUTSW | OUTSW | 6F | 不影响标志位 | 将 DS:(E)SI 处的字输出到 DX 指定的端口 | OUTSW |
| OUTSD | OUTSD | 6F | 不影响标志位 | 将 DS:(E)SI 处的双字输出到 DX 指定的端口 | OUTSD |
| | PACKSSWB mm1, mm2/m64 | 0F 63 /r | 不影响标志位 | 紧缩带符号字到字节，溢出取饱和值 | |
| | PACKSSWB xmm1,xmm2/ m128 | 66 0F 63 /r | | | PACKSSWB XMM0, XMM1 |
| PACKSSDW | PACKSSDW mm1, mm2/m64 | 0F 6B /r | 不影响标志位 | 紧缩带符号双字到字，溢出取饱和值 | |
| | PACKSSDW xmm1,xmm2/ m128 | 66 0F 6B /r | | | PACKSSDW XMM0, XMM1 |
| PACKUSWB | PACKUSWB mm1, mm2/m64 | 0F 67 /r | 不影响标志位 | 紧缩无符号字到字节，溢出取饱和值 | |
| | PACKUSWB xmm1,xmm2/ m128 | 66 0F 67 /r | | | PACKUSWB XMM0, XMM1 |
| PADDB | PADDB mm, mm/m64 | 0F FC /r | 不影响标志位 | 紧缩字节相加，溢出则环绕 | |
| | PADDB xmm1,xmm2/ m128 | 66 0F FC /r | | | PADDB XMM6, XMM7 |
| PADDW | PADDW mm, mm/m64 | 0F FD /r | 不影响标志位 | 紧缩字相加，溢出则环绕 | |
| | PADDW xmm1,xmm2/ m128 | 66 0F FD /r | | | PADDW XMM6, XMM7 |
| PADDD | PADDD mm, mm/m64 | 0F FE /r | 不影响标志位 | 紧缩双字相加，溢出则环绕 | |
| | PADDD xmm1, | 66 0F FE /r | | | PADDD XMM6, X |

| | | | | | |
|---------|------------------------|-------------|--------|-----------------------------------|--------------------|
| | xmm2/m128 | | | | MM7 |
| PADDQ | PADDQ mm1, mm2/m64 | 0F D4 /r | 不影响标志位 | 紧缩四字相加，溢出则环绕 | |
| | PADDQ xmm1,xmm2/m128 | 66 0F D4 /r | | | PADDQ XMM6, XMM7 |
| PADDSB | PADDSB mm, mm/m64 | 0F EC /r | 不影响标志位 | 带符号紧缩字节相加，溢出取饱和值 | |
| | PADDSB xmm1,xmm2/m128 | 66 0F EC /r | | | PADDSB XMM6, XMM7 |
| PADDSW | PADDSW mm, mm/m64 | 0F ED /r | 不影响标志位 | 带符号紧缩字相加，溢出取饱和值 | |
| | PADDSW xmm1,xmm2/m128 | 66 0F ED /r | | | PADDSW XMM6, XMM7 |
| PADDUSB | PADDUSB mm, mm/m64 | 0F DC /r | 不影响标志位 | 无符号紧缩字节相加，溢出取饱和值 | |
| | PADDUSB xmm1,xmm2/m128 | 66 0F DC /r | | | PADDUSB XMM0, XMM1 |
| PADDUSW | PADDUSW mm, mm/m64 | 0F DD /r | 不影响标志位 | 无符号紧缩字相加，溢出取饱和值 | |
| | PADDUSW xmm1,xmm2/m128 | 66 0F DD /r | | | PADDUSW XMM1, XMM2 |
| PAND | PAND mm, mm/m64 | 0F DB /r | 不影响标志位 | 逻辑与操作 | |
| | PAND xmm1,xmm2/m128 | 66 0F DB /r | | | PAND XMM4, XMM1 |
| PANDN | PANDN mm, mm/m64 | 0F DF /r | 不影响标志位 | 求反后与操作: DEST ← (NOT DEST) AND SRC | |
| | PANDN xmm1,xmm2/m128 | 66 0F DF /r | | | PANDN XMM4, XMM1 |
| PAUSE | PAUSE | F3 90 | 不影响标志位 | 改善处理器的“spin-wait loop”性能 | PAUSE |
| PAVGB | PAVGB mm, mm/m64 | 0F E0 /r | 不影响标志位 | 取紧缩无符号字节的平均值(四舍五入) | |
| | PAVGB xmm1,xmm2/m128 | 0F E0 /r | | | PAVGB XMM4, XMM1 |

| | | | | | |
|---------|-------------------------------|-------------|--------|-----------------------|-----------------------|
| | m128 | | | | |
| PAVGW | PAVGW mm, mm/m64 | 0F E0 /r | 不影响标志位 | 取紧缩无符号字的平均值 (四舍五入) | |
| | PAVGW xmm1,xmm2/ m128 | 0F E0 /r | | | PAVGW XMM4, XMM1 |
| PCMPEQB | PCMPEQB mm, mm/m64 | 0F 74 /r | 不影响标志位 | 紧缩字节相等比较 | |
| | PCMPEQB xmm1,xmm2/ m128 | 66 0F 74 /r | | | PCMPEQB XMM4, XMM1 |
| PCMPEQW | PCMPEQW mm, mm/m64 | 0F 75 /r | 不影响标志位 | 紧缩字相等比较 | |
| | PCMPEQW xmm1,xmm2/ m128 | 66 0F 75 /r | | | PCMPEQW XMM4, XMM1 |
| PCMPEQD | PCMPEQD mm, mm/m64 | 0F 76 /r | 不影响标志位 | 紧缩双字相等比较 | |
| | PCMPEQD xmm1,xmm2/ m128 | 66 0F 76 /r | | | PCMPEQD XMM4, XMM1 |
| PCMPGTB | PCMPGTB mm, mm/m64 | 0F 64 /r | 不影响标志位 | 带符号紧缩字节大于比较 | |
| | PCMPGTB xmm1,xmm2/ m128 | 66 0F 64 /r | | | PCMPGTB XMM4, XMM1 |
| PCMPGTW | PCMPGTW mm, mm/m64 | 0F 65 /r | 不影响标志位 | 带符号紧缩字大于比较 | |
| | PCMPGTW xmm1,xmm2/ m128 | 66 0F 65 /r | | | PCMPGTW XMM4, XMM1 |
| PCMPGtd | PCMPGtd mm, mm/m64 | 0F 66 /r | 不影响标志位 | 带符号紧缩双字大于比较 | |
| | PCMPGtd xmm1,xmm2/ m128 | 66 0F 66 /r | | | PCMPGtd XMM4, XMM1 |
| PMADDWD | PMADDWD mm mm/m64 | 0F F5 /r | 不影响标志位 | 紧缩乘和加操作 | |
| | PMADDWD xmm1,xmm2/ m128 | 66 0F F5 /r | | | PMADDWD XMM4, XMM1 |
| PMAXSW | PMAXSW mm1, | 0F EE /r | 不影响标志位 | 有符号字比较, 返回最大 | |

| | | | | | |
|---------|-------------------------------|-------------|--------|--------------|-----------------------|
| | mm2/m64 | | | 值 | |
| | PMAXSW xmm1,xmm2/ m128 | 66 0F EE /r | | | |
| PMAXUB | PMAXUB mm1, mm2/m64 | 0F DE /r | 不影响标志位 | 无符号字比较，返回最大值 | PMAXUB XMM4, XMM1 |
| | PMAXUB xmm1,xmm2/ m128 | 66 0F DE /r | | | |
| PMINSW | PMINSW mm1, mm2/m64 | 0F EA /r | 不影响标志位 | 有符号字比较，返回最小值 | PMINSW XMM4, XMM1 |
| | PMINSW xmm1,xmm2/ m128 | 66 0F EA /r | | | |
| PMINUB | PMINUB mm1, mm2/m64 | 0F DA /r | 不影响标志位 | 无符号字比较，返回最小值 | PMINUB XMM4, XMM1 |
| | PMINUB xmm1,xmm2/ m128 | 66 0F DA /r | | | |
| PMULHUW | PMULHUW mm1, mm2/m64 | 0F E4 /r | 不影响标志位 | 紧缩无符号字相乘，存高位 | PMULHUW XMM4, XMM1 |
| | PMULHUW xmm1,xmm2/ m128 | 66 0F E4 /r | | | |
| PMULHW | PMULHW mm1, mm2/m64 | 0F E5 /r | 不影响标志位 | 紧缩有符号字相乘，存高位 | PMULHW XMM4, XMM1 |
| | PMULHW xmm1,xmm2/ m128 | 66 0F E5 /r | | | |
| PMULLW | PMULLW mm1, mm2/m64 | 0F D5 /r | 不影响标志位 | 紧缩有符号字相乘，存低位 | PMULLW XMM4, XMM1 |
| | PMULLW xmm1,xmm2/ m128 | 66 0F D5 /r | | | |
| PMULUDQ | PMULUDQ mm1, mm2/m64 | 0F F4 /r | 不影响标志位 | 无符号双字相乘，存四字 | PMULUDQ XMM4, XMM1 |
| | PMULUDQ xmm1,xmm2/ m128 | 66 0F F4 /r | | | |
| POP | POP r/m16 | 8F /0 | 不影响标志位 | 16 位数据出栈 | POP WORD Ptr |

| | | | | | |
|-------|----------------------|----------------|---------|---|-----------------------------|
| | | | | | [006387EA] |
| | POP r/m32 | 8F /0 | | 32 位数据出栈 | POP DWORD Ptr [006387EA] |
| | POP r16 | 58+rw | | 16 位数据出栈到寄存器 | POP AX |
| | POP r32 | 58+rd | | 32 位数据出栈到寄存器 | POP EAX |
| | POP DS | 1F | | 数据出栈到寄存器 DS | POP DS |
| | POP ES | 07 | | 数据出栈到寄存器 ES | POP ES |
| | POP SS | 17 | | 数据出栈到寄存器 SS | POP SS |
| | POP FS | 0F A1 | | 数据出栈到寄存器 FS | POP FS |
| | POP GS | 0F A9 | | 32 位数据出栈到寄存器 GS | POP GS |
| POPA | POPA | 61 | 不影响标志位 | 从堆栈中弹出全部 16 位通用寄存器: DI, SI, BP, BX, DX, CX, AX | POPA |
| POPAD | POPAD | 61 | 不影响标志位 | 从堆栈中弹出全部 32 位通用寄存器: EDI, ESI, EBP, EBX, EDX, ECX, EAX | POPAD |
| POPF | POPF | 9D | 设置所有标志位 | 从堆栈中弹出 16 位标志寄存器 | POPF |
| POPFD | POPFD | 9D | 设置所有标志位 | 从堆栈中弹出 32 位标志寄存器 | POPFD |
| POR | POR mm, mm/m64 | 0F EB /r | 不影响标志位 | 逻辑或操作 | |
| | POR xmm1,xmm2/m128 | 66 0F EB /r | | | POR XMM4, XMM1 |
| PSLLD | PSLLD mm, mm/m64 | 0F F2 /r | 不影响标志位 | 紧缩双字逻辑左移 | |
| | PSLLD xmm1,xmm2/m128 | 66 0F F2 /r | | | PSLLD XMM4, XMM1 |
| | PSLLD mm, imm8 | 0F 72 /6 ib | | | |
| | PSLLD xmm1, imm8 | 66 0F 72 /6 ib | | | |
| PSLLQ | PSLLQ mm, mm/m64 | 0F F3 /r | 不影响标志位 | 紧缩四字逻辑左移 | |
| | PSLLQ xmm1,xmm2/m128 | 66 0F F3 /r | | | PSLLQ XMM4, XMM1 |
| | PSLLQ mm, imm8 | 0F 73 /6 ib | | | |

| | | | | | |
|-------|-----------------------------|----------------|--------|----------|---------------------|
| | m8 | | | | |
| | PSLLQ xmm1, imm8 | 66 0F 73 /6 ib | | | |
| PSLLW | PSLLW mm, mm/m64 | 0F F1 /r | 不影响标志位 | 紧缩字逻辑左移 | |
| | PSLLW xmm1,xmm2/ m128 | 66 0F F1 /r | | | PSLLW XMM4, XMM1 |
| | PSLLW mm, imm8 | 0F 71 /6 ib | | | |
| | PSLLW xmm1, imm8 | 66 0F 71 /6 ib | | | |
| PSRAD | PSRAD mm, mm/m64 | 0F E2 /r | 不影响标志位 | 紧缩双字算术右移 | |
| | PSRAD xmm1,xmm2/ m128 | 66 0F E2 /r | | | PSRAD XMM4, XMM1 |
| | PSRAD mm, imm8 | 0F 72 /4 ib | | | |
| | PSRAD xmm1, imm8 | 66 0F 72 /4 ib | | | |
| PSRAW | PSRAW mm, mm/m64 | 0F E1 /r | 不影响标志位 | 紧缩字算术右移 | |
| | PSRAW xmm1,xmm2/ m128 | 66 0F E1 /r | | | PSRAW XMM4, XMM1 |
| | PSRAW mm, imm8 | 0F 71 /4 ib | | | |
| | PSRAW xmm1, imm8 | 66 0F 71 /4 ib | | | |
| PSRLD | PSRLD mm, mm/m64 | 0F D2 /r | 不影响标志位 | 紧缩双字逻辑右移 | |
| | PSRLD xmm1,xmm2/ m128 | 66 0F D2 /r | | | PSRLD XMM4, XMM1 |
| | PSRLD mm, imm8 | 0F 72 /2 ib | | | |
| | PSRLD xmm1, imm8 | 66 0F 72 /2 ib | | | |
| PSRLQ | PSRLQ mm, mm/m64 | 0F D3 /r | 不影响标志位 | 紧缩四字逻辑右移 | |
| | PSRLQ | 66 0F D3 /r | | | PSRLQ XMM4, |

| | | | | | |
|--------|------------------------------|----------------|--------|------------------|----------------------|
| | xmm1,xmm2/ m128 | | | | XMM1 |
| | PSRLQ mm, imm8 | 0F 73 /2 ib | | | |
| | PSRLQ xmm1, imm8 | 66 0F 73 /2 ib | | | |
| PSRLW | PSRLW mm, mm/m64 | 0F D1 /r | | | |
| | PSRLW xmm1,xmm2/ m128 | 66 0F D1 /r | 不影响标志位 | 紧缩字逻辑右移 | PSRLW XMM4, XMM1 |
| | PSRLW mm, imm8 | 0F 71 /2 ib | | | |
| | PSRLW xmm1, imm8 | 66 0F 71 /2 ib | | | |
| PSUBB | PSUBB mm, mm/m64 | 0F F8 /r | | | |
| | PSUBB xmm1,xmm2/ m128 | 66 0F F8 /r | 不影响标志位 | 紧缩字节减法，溢出则回绕 | PSUBB XMM6, XMM7 |
| PSUBW | PSUBW mm, mm/m64 | 0F F9 /r | | | |
| | PSUBW xmm1,xmm2/ m128 | 66 0F F9 /r | 不影响标志位 | 紧缩字减法，溢出则回绕 | PSUBW XMM6, XMM7 |
| PSUBD | PSUBD mm, mm/m64 | 0F FA /r | | | |
| | PSUBD xmm1,xmm2/ m128 | 66 0F FA /r | 不影响标志位 | 紧缩双字减法，溢出则回绕 | PSUBD XMM6, XMM7 |
| PSUBQ | PSUBQ mm1, mm2/m64 | 0F FB /r | | | |
| | PSUBQ xmm1,xmm2/ m128 | 66 0F FB /r | 不影响标志位 | 紧缩四字减法，溢出则回绕 | PSUBQ XMM6, XMM7 |
| PSUBSB | PSUBSB mm, mm/m64 | 0F E8 /r | | | |
| | PSUBSB xmm1,xmm2/ m128 | 66 0F E8 /r | 不影响标志位 | 带符号紧缩字节减法，溢出取饱和值 | PSUBSB XMM6, XMM7 |
| PSUBSW | PSUBSW mm, mm/m64 | 0F E9 /r | 不影响标志位 | 带符号紧缩字减法，溢出取饱和值 | |

| | | | | | |
|------------|----------------------------------|-------------|--------|------------------|-------------------------|
| | PSUBSW xmm1,xmm2/ m128 | 66 0F E9 /r | | | PSUBSW XMM6, XMM7 |
| PSUBUSB | PSUBUSB mm, mm/m64 | 0F D8 /r | 不影响标志位 | 无符号紧缩字节减法，溢出取饱和值 | |
| | PSUBUSB xmm1,xmm2/ m128 | 66 0F D8 /r | | | PSUBUSB XMM6, XMM7 |
| PSUBUSW | PSUBUSW mm, mm/m64 | 0F D9 /r | 不影响标志位 | 无符号紧缩字减法，溢出取饱和值 | |
| | PSUBUSW xmm1,xmm2/ m128 | 66 0F D9 /r | | | PSUBUSW XMM6, XMM7 |
| PUNPCKHBW | PUNPCKHBW mm, mm/m64 | 0F 68 /r | 不影响标志位 | 反紧缩高位，字节到字 | |
| | PUNPCKHBW xmm1,xmm2/ m128 | 66 0F 68 /r | | | PUNPCKHBW XMM6, XMM7 |
| PUNPCKHWD | PUNPCKHWD mm, mm/m64 | 0F 69 /r | 不影响标志位 | 反紧缩高位，字到双字 | |
| | PUNPCKHWD xmm1,xmm2/ m128 | 66 0F 69 /r | | | PUNPCKHWD XMM6, XMM7 |
| PUNPCKHDQ | PUNPCKHDQ mm1, mm2/m64 | 0F 6A /r | 不影响标志位 | 反紧缩高位，双字到四字 | |
| | PUNPCKHDQ xmm1,xmm2/ m128 | 66 0F 6A /r | | | PUNPCKHDQ XMM6, XMM7 |
| PUNPCKHQDQ | PUNPCKHQDQ xmm1,xmm2/ m128 | 66 0F 6D /r | 不影响标志位 | 反紧缩高位，四字到八字 | |
| PUNPCKLBW | PUNPCKLBW mm, mm/m64 | 0F 60 /r | 不影响标志位 | 反紧缩低位，字节到字 | |
| | PUNPCKLBW xmm1,xmm2/ m128 | 66 0F 60 /r | | | PUNPCKLBW XMM6, XMM7 |
| PUNPCKLWD | PUNPCKLWD mm, mm/m64 | 0F 61 /r | 不影响标志位 | 反紧缩低位，字到双字 | |
| | PUNPCKLWD xmm1,xmm2/ m128 | 66 0F 61 /r | | | PUNPCKLWD XMM6, XMM7 |
| PUNPCKLD | PUNPCKLDQ m | 0F 62 /r | 不影响标志位 | 反紧缩低位，双字到四字 | |

| | | | | | |
|-----------|---------------------------------|-------------|-------------|--|------------------------------|
| Q | m1, mm2/m64 | | | | |
| | PUNPCKLDQ xmm1,xmm2/ m128 | 66 0F 62 /r | | | PUNPCKLDQ XMM6, XMM7 |
| PUNPCKLDQ | PUNPCKLDQ xmm1, xmm2/m128 | 66 0F 6C /r | 不影响标志位 | 反紧缩低位，四字到八字 | |
| PUSH | PUSH r/m16 | FF /6 | 不影响标志位 | 16 位数据压栈 | PUSH WORD Ptr [006387EA] |
| | PUSH r/m32 | FF /6 | | 32 位数据压栈 | PUSH DWORD Ptr [006387EA] |
| | PUSH r16 | 50+rw | | 16 位寄存器数据压栈 | PUSH AX |
| | PUSH r32 | 50+rd | | 32 位寄存器数据压栈 | PUSH EAX |
| | PUSH imm8 | 6A | | 8 位立即数据压栈 | PUSH EA |
| | PUSH imm16 | 68 | | 16 位立即数据压栈 | PUSH 87EA |
| | PUSH imm32 | 58 | | 32 位立即数据压栈 | PUSH 006387EA |
| | PUSH CS | 0E | | 寄存器 CS 数据压栈 | PUSH CS |
| | PUSH SS | 16 | | 寄存器 SS 数据压栈 | PUSH SS |
| | PUSH DS | 1E | | 寄存器 DS 数据压栈 | PUSH DS |
| | PUSH ES | 06 | | 寄存器 ES 数据压栈 | PUSH ES |
| | PUSH FS | 0F A0 | | 寄存器 FS 数据压栈 | PUSH FS |
| | PUSH GS | 0F A8 | | 寄存器 GS 数据压栈 | PUSH GS |
| PUSHA | PUSHA | 60 | 不影响标志位 | 压栈全部 16 位通用寄存器: AX, CX, DX, BX, SP, BP, SI, DI | PUSHA |
| PUSHAD | PUSHAD | 60 | 不影响标志位 | 压栈全部 32 位通用寄存器: EAX, ECX, EDX, EBX, ESP, EBP, ESI, EDI | PUSHAD |
| PUSHF | PUSHF | 9C | 设置所有标志位 | 压栈 16 位标志寄存器 | PUSHF |
| PUSHFD | PUSHFD | 9C | 设置所有标志位 | 压栈 32 位标志寄存器 | PUSHFD |
| PXOR | PXOR mm, mm/m64 | 0F EF /r | 不影响标志位 | 逻辑异或操作 | |
| | PXOR xmm1,xmm2/ m128 | 66 0F EF /r | | | PXOR XMM6, XMM7 |
| | RCL r/m8, 1 | D0 /2 | CF(或 OF)被改变 | 带进位循环左移 1 次 | RCL AL, 1 |
| | RCL r/m8, CL | D2 /2 | | 带进位循环左移 CL 次 | RCL AL, CL |
| | RCL r/m8, imm8 | C0 /2 ib | | 带进位循环左移 imm8 次 | RCL AL, 03 |
| | RCL r/m16, 1 | D1 /2 | | 带进位循环左移 1 次 | RCL AX, 1 |
| | RCL r/m16, CL | D3 /2 | | 带进位循环左移 CL 次 | RCL AX, CL |

| | | | | | |
|-----------------|-----------------|----------------|-------------|----------------|------------|
| | RCL r/m16, imm8 | C1 /2 ib | | 带进位循环左移 imm8 次 | RCL AX, 03 |
| | | | | | |
| | | | | | |
| | RCL r/m32, 1 | D1 /2 | | 带进位循环左移 1 次 | RCL EAX, 1 |
| RCL r/m32 CL | D3 /2 | 带进位循环左移 CL 次 | RCL EAX, CL | | |
| RCL r/m32 imm8 | C1 /2 ib | 带进位循环左移 imm8 次 | RCL EAX, 03 | | |
| RCR | RCR r/m8, 1 | D0 /3 | CF(或 OF)被改变 | 带进位循环右移 1 次 | RCR AL, 1 |
| | RCR r/m8, CL | D2 /3 | | 带进位循环右移 CL 次 | RCR AL, CL |
| | RCR r/m8, imm8 | C0 /3 ib | | 带进位循环右移 imm8 次 | RCR AL, 03 |
| | RCR r/m16, 1 | D1 /3 | | 带进位循环右移 1 次 | RCR AX, 1 |
| | RCR r/m16, CL | D3 /3 | | 带进位循环右移 CL 次 | RCR AX, CL |
| | RCR r/m16, imm8 | C1 /3 ib | | 带进位循环右移 imm8 次 | RCR AX, 03 |
| | | | | | |
| | | | | | |
| | RCR r/m32, 1 | D1 /3 | | 带进位循环右移 1 次 | RCR EAX, 1 |
| RCR r/m32, CL | D3 /3 | 带进位循环右移 CL 次 | RCR EAX, CL | | |
| RCR r/m32, imm8 | C1 /3 ib | 带进位循环右移 imm8 次 | RCR EAX, 03 | | |
| ROL | ROL r/m8, 1 | D0 /0 | CF(或 OF)被改变 | 循环左移 1 次 | ROL AL, 1 |
| | ROL r/m8, CL | D2 /0 | | 循环左移 CL 次 | ROL AL, CL |
| | ROL r/m8, imm8 | C0 /0 ib | | 循环左移 imm8 次 | ROL AL, 03 |
| | ROL r/m16, 1 | D1 /0 | | 循环左移 1 次 | ROL AX, 1 |
| | ROL r/m16, CL | D3 /0 | | 循环左移 CL 次 | ROL AX, CL |
| | ROL r/m16, imm8 | C1 /0 ib | | 循环左移 imm8 次 | ROL AX, 03 |
| | | | | | |
| | | | | | |
| | ROL r/m32, 1 | D1 /0 | | 循环左移 1 次 | ROL EAX, 1 |
| ROL r/m32 CL | D3 /0 | 循环左移 CL 次 | ROL EAX, CL | | |
| ROL r/m32 imm8 | C1 /0 ib | 循环左移 imm8 次 | ROL EAX, 03 | | |
| ROR | ROR r/m8, 1 | D0 /1 | CF(或 OF)被改变 | 循环右移 1 次 | ROR AL, 1 |

| | | | | | |
|-----------------|-------------------|-------------|-------------|--|--|
| | ROR r/m8, CL | D2 /1 | 变 | 循环右移 CL次 | ROR AL, CL |
| | ROR r/m8, imm8 | C0 /1 ib | | 循环右移 imm8 次 | ROR AL, 03 |
| | ROR r/m16, 1 | D1 /1 | | 循环右移 1 次 | ROR AX, 1 |
| | ROR r/m16, CL | D3 /1 | | 循环右移 CL次 | ROR AX, CL |
| | ROR r/m16, imm8 | C1 /1 ib | | 循环右移 imm8 次 | ROR AX, 03 |
| | | | | | |
| | | | | | |
| | ROR r/m32, 1 | D1 /1 | | 循环右移 1 次 | ROR EAX, 1 |
| ROR r/m32, CL | D3 /1 | 循环右移 CL次 | ROR EAX, CL | | |
| ROR r/m32, imm8 | C1 /1 ib | 循环右移 imm8 次 | ROR EAX, 03 | | |
| RDMSR | RDMSR | 0F 32 | 不影响标志位 | 把 ECX 指定的模型专用寄存器内容送 EDX:EAX | RDMSR |
| RDPMC | RDPMC | 0F 33 | 不影响标志位 | 把 ECX 指定的性能监测计数器内容送 EDX:EAX | RDPMC |
| RDTSC | RDTSC | 0F 31 | 不影响标志位 | 读时间标记计数器到 EDX:EAX | RDTSC |
| REP | REP INS r/m8, DX | F3 6C | 不影响标志位 | 从 DX 指定的端口读 (E)CX 个字节(BYTE)到 ES:[(E)DI] | |
| | REP INS r/m16, DX | F3 6D | | 从 DX 指定的端口读 (E)CX 个字(WORD)到 ES:[(E)DI] | |
| | REP INS r/m32, DX | F3 6D | | 从 DX 指定的端口读 (E)CX 个双字(DWORD)到 ES:[(E)DI] | |
| | REP MOVS m8, m8 | F3 A4 | | 连续字符串传送, 每次传送 1 个字节, 传送(E)CX 次 | REP MOVS StrING1, StrING2 ; 源串 DS:[(E)SI], 目的串: ES:[(E)DI] |
| | REP MOVS m16, m16 | F3 A5 | | 连续字符串传送, 每次传送 1 个字, 传送(E)CX 次 | REP MOVS StrING1, StrING2 ; 源串 DS:[(E)SI], 目的串: ES:[(E)DI] |
| | REP MOVS m32, m32 | F3 A5 | | 连续字符串传送, 每次传送 1 个双字, 传送(E)CX 次 | REP MOVS StrING1, |

| | | | | | |
|------|-----------------------|-------|-------------------------|--|--|
| | | | | 次 | StrING2 ; 源串 DS:[(E)SI], 目的串: ES:[(E)DI] |
| | REP OUTS DX,r/m8 | F3 6E | | 将 DS:[(E)SI]处的 (E)CX 个字节输出到 DX 指定的端口 | |
| | REP OUTS DX,r/m16 | F3 6F | | 将 DS:[(E)SI]处的 (E)CX 个字输出到 DX 指 定的端口 | |
| | REP OUTS DX,r/m32 | F3 6F | | 将 DS:[(E)SI]处的 (E)CX 个双字输出到 DX 指定的端口 | |
| | REP LODS AL | F3 AC | | 将地址 DS:[(E)SI]处的 (E)CX 个字节装入 AL | |
| | REP LODS AX | F3 AD | | 将地址 DS:[(E)SI]处的 (E)CX 个字装入 AX | |
| | REP LODS EAX | F3 AD | | 将地址 DS:[(E)SI]处的 (E)CX 个双字装入 EAX | |
| | REP STOS m8 | F3 AA | | 用 AL 填充地址 ES:[(E)DI]处的(E)CX 个字节 | |
| | REP STOS m16 | F3 AB | | 用 AX 填充地址 ES:[(E)DI]处的(E)CX 个字 | |
| | REP STOS m32 | F3 AB | | 用 EAX 填充地址 ES:[(E)DI]处的(E)CX 个双字 | |
| REPE | REPE CMPS m8 m8 | F3 A6 | 设置 AF CF OF PF SF ZF | 比较字符串, 每次比较 1 个字节, 直到不相等字节 停止 | REPE CMPS StrING1, StrING2 ; 源串 DS:[(E)SI], 目的 串: ES:[(E)DI] |
| | REPE CMPS m16, m16 | F3 A7 | | 比较字符串, 每次比较 1 个字, 直到不相等字停止 | REPE CMPS StrING1, StrING2 ; 源串 DS:[(E)SI], 目的 串: ES:[(E)DI] |
| | REPE CMPS m32, m32 | F3 A7 | | 比较字符串, 每次比较 1 个双字, 直到不相等双字 停止 | REPE CMPS StrING1, StrING2 ; 源串 DS:[(E)SI], 目的 串: ES:[(E)DI] |
| | REPE SCAS m8 | F3 AE | | 扫描字符串 ES:[(E)DI] | REPE SCAS StrIN |

| | | | | | |
|-------|------------------------|-------|-------------------------|--|---|
| | | | | 中的 AL 字节值，遇到非 AL 值停止 | G1 |
| | REPE SCAS m16 | F3 AF | | 扫描字符串 ES:[(E)DI] 中的 AX 字值，遇到非 AX 值停止 | REPE CMPS StrING1 |
| | REPE SCAS m32 | F3 AF | | 扫描字符串 ES:[(E)DI] 中的 EAX 双字值，遇到非 EAX 值停止 | REPE CMPS StrING1 |
| REPNE | REPNE CMPS m8, m8 | F2 A6 | 设置 AF CF OF PF SF ZF | 比较字符串，每次比较 1 个字节，直到相等字节停止 | REPNE CMPS StrING1, StrING2 ; 源串 DS:[(E)SI], 目的 串: ES:[(E)DI] |
| | REPNE CMPS m16, m16 | F2 A7 | | 比较字符串，每次比较 1 个字，直到相等字停止 | REPNE CMPS StrING1, StrING2 ; 源串 DS:[(E)SI], 目的 串: ES:[(E)DI] |
| | REPNE CMPS m32, m32 | F2 A7 | | 比较字符串，每次比较 1 个双字，直到相等双字停止 | REPNE CMPS StrING1, StrING2 ; 源串 DS:[(E)SI], 目的 串: ES:[(E)DI] |
| | REPNE SCAS m8 | F2 AE | | 扫描字符串 ES:[(E)DI] 中的 AL 字节值，遇到 AL 值停止 | REPNE SCAS StrING1 |
| | REPNE SCAS m16 | F2 AF | | 扫描字符串 ES:[(E)DI] 中的 AX 字值，遇到 AX 值停止 | REPNE CMPS StrING1 |
| | REPNE SCAS m32 | F2 AF | | 扫描字符串 ES:[(E)DI] 中的 EAX 双字值，遇到 EAX 值停止 | REPNE CMPS StrING1 |
| | | | | | |
| REPZ | REPZ CMPS m8, m8 | F3 A6 | 设置 AF CF OF PF SF ZF | 比较字符串，每次比较 1 个字节，直到不相等字节停止 | REPZ CMPS StrING1, StrING2 ; 源串 DS:[(E)SI], 目的 串: ES:[(E)DI] |
| | REPZ CMPS m16, m16 | F3 A7 | | 比较字符串，每次比较 1 个字，直到不相等字停止 | REPZ CMPS StrING1, StrING2 ; 源串 DS:[(E)SI], 目的 串: ES:[(E)DI] |
| | REPZ CMPS m | F3 A7 | | 比较字符串，每次比较 1 | REPZ CMPS StrI |

| | | | | | |
|------|--------------------|-------|----------------------|---|---|
| | 32, m32 | | | 个双字，直到不相等双字停止 | 源串 DS:[(E)SI]，目的串：ES:[(E)DI] |
| | REPZ SCAS m8 | F3 AE | | 扫描字符串 ES:[(E)DI] 中的 AL 字节值，遇到非 AL 值停止 | REPZ SCAS StrING1 |
| | REPZ SCAS m16 | F3 AF | | 扫描字符串 ES:[(E)DI] 中的 AX 字值，遇到非 AX 值停止 | REPZ CMPS StrING1 |
| | REPZ SCAS m32 | F3 AF | | 扫描字符串 ES:[(E)DI] 中的 EAX 双字值，遇到非 EAX 值停止 | REPZ CMPS StrING1 |
| REPZ | REPZ CMPS m8, m8 | F2 A6 | 设置 AF CF OF PF SF ZF | 比较字符串，每次比较 1 个字节，直到相等字节停止 | REPZ CMPS StrING1, StrING2 ; 源串 DS:[(E)SI]，目的串：ES:[(E)DI] |
| | REPZ CMPS m16, m16 | F2 A7 | | 比较字符串，每次比较 1 个字，直到相等字停止 | REPZ CMPS StrING1, StrING2 ; 源串 DS:[(E)SI]，目的串：ES:[(E)DI] |
| | REPZ CMPS m32, m32 | F2 A7 | | 比较字符串，每次比较 1 个双字，直到相等双字停止 | REPZ CMPS StrING1, StrING2 ; 源串 DS:[(E)SI]，目的串：ES:[(E)DI] |
| | REPZ SCAS m8 | F2 AE | | 扫描字符串 ES:[(E)DI] 中的 AL 字节值，遇到 AL 值停止 | REPZ SCAS StrING1 |
| | REPZ SCAS m16 | F2 AF | | 扫描字符串 ES:[(E)DI] 中的 AX 字值，遇到 AX 值停止 | REPZ CMPS StrING1 |
| | REPZ SCAS m32 | F2 AF | | 扫描字符串 ES:[(E)DI] 中的 EAX 双字值，遇到 EAX 值停止 | REPZ CMPS StrING1 |
| RET | RET | C3 | 恢复压栈的标志位 | 子过程返回(Near) | RET |
| | RET | CB | | 子过程返回(Far) | RET |
| | RET imm16 | C2 iw | | 子过程返回(Near)，并从堆栈弹出 imm16 字节 | RET 08 |
| | RET imm16 | CA iw | | 子过程返回(Far)，并从堆栈弹出 imm16 字节 | RET 08 |

| | | | | | |
|----------------|-----------------|---|-------------------|---|------------|
| RSM | RSM | OF AA | 恢复所有标志位 | 从系统管理模式返回 | RSM |
| | | | | | |
| | SAHF | 9E | 设置 SF ZF AF PF CF | 装入 AH 到标志寄存器， 格式： (SF:ZF:0:AF:0:PF:1:CF)←AH | SAHF |
| | | | | | |
| SAL | SAL r/m8, 1 | D0 /4 | CF(或 OF)被改变 | 算术左移 1 次(乘法： $r/m8=r/m8*2$) | SAL AL, 1 |
| | SAL r/m8, CL | D2 /4 | | 算术左移 CL 次(乘法： $r/m8=r/m8*(2^{CL})$) | SAL AL, CL |
| | SAL r/m8, imm8 | C0 /4 ib | | 算术左移 imm8 次(乘法： $r/m8=r/m8*(2^{imm8})$) | SAL AL, 03 |
| | SAL r/m16, 1 | D1 /4 | | 算术左移 1 次(乘法： $r/m16=r/m16*2$) | SAL AX, 1 |
| | SAL r/m16, CL | D3 /4 | | 算术左移 CL 次(乘法： $r/m16=r/m16*(2^{CL})$) | SAL AX, CL |
| | SAL r/m16, imm8 | C1 /4 ib | | 算术左移 imm8 次(乘法： $r/m16=r/m16*(2^{imm8})$) | SAL AX, 03 |
| | | | | | |
| | | | | | |
| | SAL r/m32, 1 | D1 /4 | | 算术左移 1 次(乘法： $r/m32=r/m32*2$) | SAL EAX, 1 |
| SAL r/m32 CL | D3 /4 | 算术左移 CL 次(乘法： $r/m32=r/m32*(2^{CL})$) | SAL EAX, CL | | |
| SAL r/m32 imm8 | C1 /4 ib | 算术左移 imm8 次(乘法： $r/m32=r/m32*(2^{imm8})$) | SAL EAX, 03 | | |
| SAR | SAR r/m8, 1 | D0 /7 | CF(或 OF)被改变 | 算术右移 1 次(有符号除法： $r/m8=r/m8 / 2$) | SAR AL, 1 |
| | SAR r/m8, CL | D2 /7 | | 算术右移 CL 次(有符号除法： $r/m8=r/m8 / (2^{CL})$) | SAR AL, CL |
| | SAR r/m8, imm8 | C0 /7 ib | | 算术右移 imm8 次(有符号除法： $r/m8=r/m8 / (2^{imm8})$) | SAR AL, 03 |
| | SAR r/m16, 1 | D1 /7 | | 算术右移 1 次(有符号除 | SAR AX, 1 |

| | | | | | |
|-----------------|-----------------|---|-------------|---|------------|
| | | | | 法: $r/m16=r/m16 / 2$) | |
| | SAR r/m16, CL | D3 /7 | | 算术右移CL次(有符号除法: $r/m16=r/m16 / (2^{CL})$) | SAR AX, CL |
| | SAR r/m16, imm8 | C1 /7 ib | | 算术右移imm8次(有符号除法: $r/m16=r/m16 / (2^{imm8})$) | SAR AX, 03 |
| | | | | | |
| | SAR r/m32, 1 | D1 /7 | | 算术右移1次(有符号除法: $r/m32=r/m32 / 2$) | SAR EAX, 1 |
| SAR r/m32, CL | D3 /7 | 算术右移CL次(有符号除法: $r/m32=r/m32 / (2^{CL})$) | SAR EAX, CL | | |
| SAR r/m32, imm8 | C1 /7 ib | 算术右移imm8次(有符号除法: $r/m32=r/m32 / (2^{imm8})$) | SAR EAX, 03 | | |
| SHL | SHL r/m8, 1 | D0 /4 | CF(或 OF)被改变 | 逻辑左移1次(乘法: $r/m8=r/m8*2$) | SHL AL, 1 |
| | SHL r/m8, CL | D2 /4 | | 逻辑左移CL次(乘法: $r/m8=r/m8*(2^{CL})$) | SHL AL, CL |
| | SHL r/m8, imm8 | C0 /4 ib | | 逻辑左移imm8次(乘法: $r/m8=r/m8*(2^{imm8})$) | SHL AL, 03 |
| | SHL r/m16, 1 | D1 /4 | | 逻辑左移1次(乘法: $r/m16=r/m16*2$) | SHL AX, 1 |
| | SHL r/m16, CL | D3 /4 | | 逻辑左移CL次(乘法: $r/m16=r/m16*(2^{CL})$) | SHL AX, CL |
| | SHL r/m16, imm8 | C1 /4 ib | | 逻辑左移imm8次(乘法: $r/m16=r/m16*(2^{imm8})$) | SHL AX, 03 |
| | | | | | |
| | | | | | |
| | SHL r/m32, 1 | D1 /4 | | 逻辑左移1次(乘法: $r/m32=r/m32*2$) | SHL EAX, 1 |
| SHL r/m32 CL | D3 /4 | 逻辑左移CL次(乘法: $r/m32=r/m32*(2^{CL})$) | SHL EAX, CL | | |
| SHL r/m32 imm8 | C1 /4 ib | 逻辑左移imm8次(乘法: $r/m32=r/m32*(2^{imm8})$) | SHL EAX, 03 | | |

| | | | | | |
|-----------------|------------------|---|----------------------|---|------------------------------------|
| | | $r/m32 = r/m32 * (2^i \text{ mm}8))$ | | | |
| SHR | SHR r/m8, 1 | D0 /5 | CF(或 OF)被改变 | 逻辑右移1次(无符号除法: $r/m8 = r/m8 / 2$) | SHR AL, 1 |
| | SHR r/m8, CL | D2 /5 | | 逻辑右移CL次(无符号除法: $r/m8 = r/m8 / (2^{CL})$) | SHR AL, CL |
| | SHR r/m8, imm8 | C0 /5 ib | | 逻辑右移imm8次(无符号除法: $r/m8 = r/m8 / (2^{imm8})$) | SHR AL, 03 |
| | SHR r/m16, 1 | D1 /5 | | 逻辑右移1次(无符号除法: $r/m16 = r/m16 / 2$) | SHR AX, 1 |
| | SHR r/m16, CL | D3 /5 | | 逻辑右移CL次(无符号除法: $r/m16 = r/m16 / (2^{CL})$) | SHR AX, CL |
| | SHR r/m16, imm8 | C1 /5 ib | | 逻辑右移imm8次(无符号除法: $r/m16 = r/m16 / (2^{imm8})$) | SHR AX, 03 |
| | | | | | |
| | SHR r/m32, 1 | D1 /5 | | 逻辑右移1次(无符号除法: $r/m32 = r/m32 / 2$) | SHR EAX, 1 |
| SHR r/m32, CL | D3 /5 | 逻辑右移CL次(无符号除法: $r/m32 = r/m32 / (2^{CL})$) | SHR EAX, CL | | |
| SHR r/m32, imm8 | C1 /5 ib | 逻辑右移imm8次(无符号除法: $r/m32 = r/m32 / (2^{imm8})$) | SHR EAX, 03 | | |
| SBB | SBB AL, imm8 | 1C ib | 设置 AF CF OF SF PF ZF | 带借位减法 | SBB AL, 1F |
| | SBB AX, imm16 | 1D iw | | | SBB AX, 4F80 |
| | SBB EAX, imm32 | 1D id | | | SBB EAX, 00004F80 |
| | SBB r/m8, imm8 | 80 /3 ib | | | SBB BYTE Ptr [006387EA], 39 |
| | SBB r/m16, imm16 | 81 /3 iw | | | SBB WORD Ptr [006387EA], 1039 |
| | SBB r/m32, imm32 | 81 /3 id | | | SBB DWORD Ptr [006387EA], 00001039 |

| | | | | | |
|-------|----------------|----------|----------------------|------------------------------------|------------------------------|
| | SBB r/m16,imm8 | 83 /3 ib | | | SBB WORD Ptr [006387EA], 39 |
| | SBB r/m32,imm8 | 83 /3 ib | | | SBB DWORD Ptr [006387EA], 39 |
| | SBB r/m8,r8 | 18 /r | | | SBB [006387EA], AL |
| | SBB r/m16,r16 | 19 /r | | | SBB [006387EA], AX |
| | SBB r/m32,r32 | 19 / r | | | SBB [006387EA], EAX |
| | SBB r8,r/m8 | 1A /r | | | SBB AL, [006387EA] |
| | SBB r16,r/m16 | 1B /r | | | SBB AX, [006387EA] |
| | SBB r32,r/m32 | 1B /r | | | SBB EAX, [006387EA] |
| SCAS | SCAS m8 | AE | 设置 OF SF ZF AF PF CF | 用 AL 中的字节值扫描字符串 ES:(E)DI, 然后设置标志位 | |
| | SCAS m16 | AF | | 用 AX 中的字值扫描字符串 ES:(E)DI, 然后设置标志位 | |
| | SCAS m32 | AF | | 用 EAX 中的双字值扫描字符串 ES:(E)DI, 然后设置标志位 | |
| SCASB | SCASB | AE | 设置 OF SF ZF AF PF CF | 用 AL 中的字节值扫描字符串 ES:(E)DI, 然后设置标志位 | SCASB |
| SCASW | SCASW | AF | 设置 OF SF ZF AF PF CF | 用 AX 中的字值扫描字符串 ES:(E)DI, 然后设置标志位 | SCASW |
| SCASD | SCASD | AF | 设置 OF SF ZF AF PF CF | 用 EAX 中的双字值扫描字符串 ES:(E)DI, 然后设置标志位 | SCASD |
| SETcc | SETA r/m8 | 0F 97 | 高于(CF=0 and ZF=0) | 条件设置指令, 如果条件满足则 r/m8=1, 否则 r/m8=0 | SETA AL |
| | SETAE r/m8 | 0F 93 | 高于等于(CF=0) | | SETAE AL |
| | SETB r/m8 | 0F 92 | 低于(CF=1) | | SETB AL |
| | SETBE r/m8 | 0F 96 | 低于等于(CF=1 or ZF=1) | | SETBE AL |
| | SETC r/m8 | 0F 92 | 有进位(CF=1) | | SETC AL |
| | SETE r/m8 | 0F 94 | 等于(ZF=1) | | SETE AL |

| | | | | | |
|------|-------------|----------|----------------------------|--------------------|----------------|
| | SETG r/m8 | 0F 9F | 大于($ZF=0$ and $SF=OF$) | | SETG AL |
| | SETGE r/m8 | 0F 9D | 大于等于($SF=OF$) | | SETGE AL |
| | SETL r/m8 | 0F 9C | 小于($SF<>OF$) | | SETL AL |
| | SETLE r/m8 | 0F 9E | 小于等于($ZF=1$ or $SF<>OF$) | | SETLE AL |
| | SETNA r/m8 | 0F 96 | 不高于($CF=1$ or $ZF=1$) | | SETNA AL |
| | SETNAE r/m8 | 0F 92 | 不高等于($CF=1$) | | SETNAE AL |
| | SETNB r/m8 | 0F 93 | 不低于($CF=0$) | | SETNB AL |
| | SETNBE r/m8 | 0F 97 | 不低等于($CF=0$ and $ZF=0$) | | SETNBE AL |
| | SETNC r/m8 | 0F 93 | 无进位($CF=0$) | | SETNC AL |
| | SETNE r/m8 | 0F 95 | 不等于($ZF=0$) | | SETNE AL |
| | SETNG r/m8 | 0F 9E | 不大于($ZF=1$ or $SF<>OF$) | | SETNG AL |
| | SETNGE r/m8 | 0F 9C | 不大等于($SF<>OF$) | | SETNGE AL |
| | SETNL r/m8 | 0F 9D | 不小于($SF=OF$) | | SETNL AL |
| | SETNLE r/m8 | 0F 9F | 不小等于($ZF=0$ and $SF=OF$) | | SETNLE AL |
| | SETNO r/m8 | 0F 91 | 无溢出($OF=0$) | | SETNO AL |
| | SETNP r/m8 | 0F 9B | 非偶数($PF=0$) | | SETNP AL |
| | SETNS r/m8 | 0F 99 | 非负数($SF=0$) | | SETNS AL |
| | SETNZ r/m8 | 0F 95 | 非零($ZF=0$) | | SETNZ AL |
| | SETO r/m8 | 0F 90 | 溢出($OF=1$) | | SETO AL |
| | SETP r/m8 | 0F 9A | 偶数($PF=1$) | | SETP AL |
| | SETPE r/m8 | 0F 9A | 偶数($PF=1$) | | SETPE AL |
| | SETPO r/m8 | 0F 9B | 奇数($PF=0$) | | SETPO AL |
| | SETS r/m8 | 0F 98 | 负数($SF=1$) | | SETS AL |
| | SETZ r/m8 | 0F 94 | 为零($ZF=1$) | | SETZ AL |
| SGDT | SGDT m | 0F 01 /0 | 不影响标志位 | 保存全局描述符表寄存器到内存 m 处 | SGDT [EBP] |
| SIDT | SIDT m | 0F 01 /1 | 不影响标志位 | 保存中断描述符表寄存器到内存 m 处 | SIDT [ESI] |
| SLDT | SLDT r/m16 | 0F 00 /0 | 不影响标志位 | 保存 LDT 选择子到 r/m16 | SLDT SI |
| | SLDT r/m32 | | | 保存 LDT 选择子到 r/m32 | SLDT ESI |
| SHLD | SHLD r/m16, | 0F A4 | 设置 CF SF ZF | 双精度左移 | SHLD AX, BX, 3 |

| | | | | | |
|-------|-----------------------|----------|-------------------------|----------------------|-------------------|
| | r16, imm8 | | PF AF OF | | |
| | SHLD r/m16, r16, CL | 0F A5 | | | SHLD AX, BX, CL |
| | SHLD r/m32, r32, imm8 | 0F A4 | | | SHLD EAX, EBX, 3 |
| | SHLD r/m32, r32, CL | 0F A5 | | | SHLD EAX, EBX, CL |
| SHRD | SHRD r/m16, r16, imm8 | 0F AC | 设置 CF SF ZF PF AF OF | 双精度右移 | SHRD AX, BX, 3 |
| | SHRD r/m16, r16, CL | 0F AD | | | SHRD AX, BX, CL |
| | SHRD r/m32, r32, imm8 | 0F AC | | | SHRD EAX, EBX, 3 |
| | SHRD r/m32, r32, CL | 0F AD | | | SHRD EAX, EBX, CL |
| SMSW | SMSW r/m16 | 0F 01 /4 | 不影响标志位 | 保存机器状态字到 r/m16 | SMSW SI |
| | SMSW r32/m16 | | | 保存机器状态字到 r32/m16 | SMSW ESI |
| STC | STC | F9 | 设置 CF | 设置 CF=1 | STC |
| Std | Std | FD | 设置 DF | 设置 DF=1 | Std |
| STI | STI | FB | 设置 IF | 设置 IF=1, 开中断 | STI |
| STOS | STOS m8 | AA | 不影响标志位 | 将 AL 保存到地址 ES:(E)DI | |
| | STOS m16 | AB | | 将 AX 保存到地址 ES:(E)DI | |
| | STOS m32 | AB | | 将 EAX 保存到地址 ES:(E)DI | |
| STOSB | STOSB | AA | 不影响标志位 | 将 AL 保存到地址 ES:(E)DI | STOSB |
| STOSW | STOSW | AB | 不影响标志位 | 将 AX 保存到地址 ES:(E)DI | STOSW |
| STOSD | STOSD | AB | 不影响标志位 | 将 EAX 保存到地址 ES:(E)DI | STOSD |
| Str | Str r/m16 | 0F 00 /1 | 不影响标志位 | 保存任务寄存器到 r/m16 | Str AX |
| SUB | SUB AL, imm8 | 2C ib | 设置 AF CF OF SF PF ZF | 减法 | SUB AL, 1F |
| | SUB AX, imm16 | 2D iw | | | SUB AX, 4F80 |
| | SUB EAX, imm32 | 2D id | | | SUB EAX, 00004F80 |
| | SUB r/m8, im | 80 /5 ib | | | SUB BYTE Ptr [0 |

| | | | | | |
|----------|---------------------|----------|----------------------|---|---|
| | m8 | | | | 06387EA], 39 |
| | SUB r/m16,imm16 | 81 /5 iw | | | SUB WORD Ptr [006387EA], 1039 |
| | SUB r/m32,imm32 | 81 /5 id | | | SUB DWORD Ptr [006387EA], 00001039 |
| | SUB r/m16,imm8 | 83 /5 ib | | | SUB WORD Ptr [006387EA], 39 |
| | SUB r/m32,imm8 | 83 /5 ib | | | SUB DWORD Ptr [006387EA], 39 |
| | SUB r/m8,r8 | 28 /r | | | SUB [006387EA], AL |
| | SUB r/m16,r16 | 29 /r | | | SUB [006387EA], AX |
| | SUB r/m32,r32 | 29 / r | | | SUB [006387EA], EAX |
| | SUB r8,r/m8 | 2A /r | | | SUB AL, [006387EA] |
| | SUB r16,r/m16 | 2B /r | | | SUB AX, [006387EA] |
| | SUB r32,r/m32 | 2B /r | | | SUB EAX, [006387EA] |
| SYSENTER | SYSENTER | 0F 34 | 设置 VM IF RF | 快速调用系统 0 级过程 (用于系统 3 级应用程序), 配合 SYSEXIT 使用 | |
| SYSEXIT | SYSEXIT | 0F 35 | 不影响标志位 | 快速返回到系统 3 级用户 代码, 配合 SYSENTER 使用 | |
| | TEST AL, imm8 | A8 ib | 设置 CF OF PF SF ZF | 逻辑与测试, 但是不改变 目的操作数, 只设置相关 标志位 | TEST AL, 1F |
| | TEST AX, imm16 | A9 iw | | | TEST AX, 4F80 |
| | TEST EAX, imm32 | A9 id | | | TEST EAX, 00004F80 |
| | TEST r/m8, imm8 | F6 /0 ib | | | TEST BYTE Ptr [006387EA], 39 |
| | TEST r/m16,imm16 | F7 /0 ib | | | TEST WORD Ptr [006387EA], 1039 |
| | TEST r/m32,imm32 | F7 /0 ib | | | TEST DWORD Ptr [006387EA], 00001039 |

| | | | | | |
|--------|--------------------|----------|-------------------------|--------------------------------------|-----------------------------------|
| | TEST r/m8,r8 | 84 /r | | | TEST BYTE Ptr [006387EA], AL |
| | TEST r/m16,r16 | 85 /r | | | TEST WORD Ptr [006387EA], AX |
| | TEST r/m32,r32 | 85 /r | | | TEST DWORD Ptr [006387EA], EAX |
| | UD2 | 0F 0B | 不影响标志位 | 产生无效代码异常，用于 软件测试 | UD2 |
| | VERR | 0F 00 /4 | 设置 ZF | 如果目标选择子对应段可 读，则 ZF 置 1 | VERR |
| VERW | VERW | 0F 00 /5 | 设置 ZF | 如果目标选择子对应段可 写，则 ZF 置 1 | VERW |
| | WAIT | 9B | C0, C1, C2, C3 未定义 | 等待，检查非屏蔽浮点异 常 | WAIT |
| WBINVD | WBINVD | 0F 09 | 不影响标志位 | 写回内部 CACHE 并使之 失效 | WBINVD |
| WRMSR | WRMSR | 0F 30 | 不影响标志位 | 把 EDX:EAX 的内容写入 ECX指定的模型专用寄存 器 | WRMSR |
| | XADD r/m8, r8 | 0F C0 /r | 设置 CF PF AF SF ZF OF | 目的操作数和源操作数相 加，并使源操作数保存目 的操作数的值 | XADD AL, BL |
| | XADD r/m16, r16 | 0F C1 /r | | | XADD AX, BX |
| | XADD r/m32, r32 | 0F C1 /r | | | XADD EAX, EBX |
| XCHG | XCHG AX, r16 | 90+rw | 不影响标志位 | 目的操作数和源操作数的 值交换 | XCHG AX, BX |
| | XCHG r16, AX | 90+rw | | | XCHG CX, AX |
| | XCHG EAX, r32 | 90+rd | | | XCHG EAX, EBX |
| | XCHG r32, EAX | 90+rd | | | XCHG ECX, EAX |
| | XCHG r/m8, r8 | 86 /r | | | XCHG [006387EA], BL |
| | XCHG r8, r/m8 | 86 /r | | | XCHG AX, [006387EA] |
| | XCHG r/m16, r16 | 87 /r | | | XCHG [006387EA], BX |
| | XCHG r16, r/m16 | 87 /r | | | XCHG AX, [006387EA] |
| | XCHG r/m32, r32 | 87 /r | | | XCHG [006387EA], EBX |
| | XCHG r32, r/m32 | 87 /r | | | XCHG EAX, [006387EA] |