

➤ (1) 程序转移指令

1、无条件转移指令（长转移）

JMP	无条件转移
CALL	过程调用
RET	过程返回

2、条件转移指令（短转移）

JA	不小于或不等于时转移.
JAE	大于或等于转移.
JB	小于转移.
JBE	小于或等于转移.

以上四条, 测试无符号整数运算的结果 (标志 C、Z)

JG	大于转移.
JGE	大于或等于转移.
JL	小于转移.
JLE	小于或等于转移.

以上四条, 测试有符号整数运算的结果 (标志 S、O、Z)

JZ	等于转移.
JNZ	不等于时转移.
JC	有进位时转移.
JNC	无进位时转移.
JNO	不溢出时转移.
JNP	奇数时转移.
JNS	符号位为“0”时转移.
JO	溢出转移.
JP	偶数时转移.
JS	符号位为“1”时转移.

3、循环控制指令 (短转移)

LOOP CX	不为零时循环.
LOOPZ	$CX \neq 0$ 且标志 $Z=1$ 时循环.
LOOPNZ	$CX \neq 0$ 且标志 $Z=0$ 时循环.
JCXZ	$CX=0$ 时转移.
JECXZ	$ECX=0$ 时转移.

4、中断指令

INT	中断指令
INTO	溢出中断
IRET	中断返回

5、处理器控制指令

HLT	停机
WAIT	等待
ESC	换码
LOCK	封锁
NOP	空操作.
STC	置 1 进位标志位.
CLC	清 0 进位标志位.
CMC	进位标志取反.
STD	置 1 方向标志位.
CLD	清 0 方向标志位.
STI	置 1 中断允许位.
CLI	清 0 中断允许位.

➤ (2) 串指令

DS:SI	源串段寄存器 : 源串变址.
ES:DI	目标串段寄存器: 目标串变址.
CX	重复次数计数器.
AL/AX	扫描值.

D 标志	0 表示重复操作中 SI 和 DI 自动增量; 1 表示重复操作中 SI 和 DI 自动减量;
Z 标志	用来控制扫描或比较操作的结束.
MOVS	串传送. (MOVSB 传送字符. MOVSW 传送字. MOVSD 传送双字.)
CMPS	串比较. (CMPSB 比较字符. CMPSW 比较字.)
SCAS	串扫描. (把 AL 或 AX 的内容与目标串比较, 结果反映在标志位.)
LODS	装入串. (LODSB 传送字符. LODSW 传送字. LODSD 传送双字.)
STOS	保存串. (是 LODS 的逆过程.)
REP	当 CX/ECX \neq 0 时重复.
REPZ	当 ZF=1 或比较结果相等, 且 CX/ECX \neq 0 时重复.
REPNZ	当 ZF=0 或比较结果不等, 且 CX/ECX \neq 0 时重复.
REPC	当 CF=1 且 CX/ECX \neq 0 时重复.
REPNC	当 CF=0 且 CX/ECX \neq 0 时重复.

➤ (3) 逻辑运算指令

AND	与运算.
OR	或运算.
XOR	异或运算.
NOT	取反运算.
TEST	测试. (两操作数作与运算, 仅修改标志位, 不回送结果).
SHL	逻辑左移.
SAL	算术左移.
SHR	逻辑右移.
SAR	算术右移.
ROL	循环左移.
ROR	循环右移.
RCL	通过进位的循环左移.
RCR	通过进位的循环右移.

移位=1 次时，可直接用操作码。如 SHL AX, 1.

移位>1 次时，可由寄存器 CL 给出移位次数。如 MOV CL, 04 SHL AX, CL

➤ （4）数据传输指令

1. 通用数据传送指令.

MOV	传送字或字节.
MOVSX	先符号扩展, 再传送.
MOVZX	先零扩展, 再传送.
PUSH	把字压入堆栈.
POP	把字弹出堆栈.
PUSHA	把 AX, CX, DX, BX, SP, BP, SI, DI 依次压入堆栈.
POPA	把 DI, SI, BP, SP, BX, DX, CX, AX 依次弹出堆栈.
PUSHAD	把 EAX, ECX, EDX, EBX, ESP, EBP, ESI, EDI 依次压入堆栈.
POPAD	把 EDI, ESI, EBP, ESP, EBX, EDX, ECX, EAX 依次弹出堆栈.
BSWAP	交换 32 位寄存器里字节的顺序
XCHG	交换字或字节. (至少有一个操作数为寄存器, 除了段寄存器)
CMPXCHG	比较并交换操作数. (第二个操作数必须为累加器 AL/AX/EAX)
XADD	先交换再累加. (结果在第一个操作数里)
XLAT	字节查表转换.

2. 输入输出端口传送指令.

IN	I/O 端口输入. (语法: IN 累加器, {端口号 DX})
OUT	I/O 端口输出. (语法: OUT {端口号 DX}, 累加器)

3. 目的地址传送指令.

LEA	装入有效地址.
例: LEA DX, string ;	把偏移地址存到 DX.
LDS	传送目标指针, 把指针内容装入 DS.
例: LDS SI, string ;	把段地址: 偏移地址存到 DS:SI.
LES	传送目标指针, 把指针内容装入 ES.

例: LES DI, string ;把段地址:偏移地址存到 ES:DI.

LFS 传送目标指针, 把指针内容装入 FS.

例: LFS DI, string ;把段地址:偏移地址存到 FS:DI.

LGS 传送目标指针, 把指针内容装入 GS.

例: LGS DI, string ;把段地址:偏移地址存到 GS:DI.

LSS 传送目标指针, 把指针内容装入 SS.

例: LSS DI, string ;把段地址:偏移地址存到 SS:DI.

4. 标志传送指令.

LAHF 标志寄存器传送, 把标志装入 AH.

SAHF 标志寄存器传送, 把 AH 内容装入标志寄存器.

PUSHF 标志入栈.

POPF 标志出栈.

PUSHD 32 位标志入栈.

POPD 32 位标志出栈.

➤ (5) 算术运算指令

ADD 加法.

ADC 带进位加法.

INC 加 1.

AAA 加法的 ASCII 码调整.

DAA 加法的十进制调整.

SUB 减法.

SBB 带借位减法.

DEC 减 1.

NEG 求反

CMP 比较. (两操作数作减法, 仅修改标志位, 不回送结果).

AAS 减法的 ASCII 码调整.

DAS 减法的十进制调整.

MUL	无符号乘法.
IMUL	整数乘法.
AAM	乘法的 ASCII 码调整.
DIV	无符号除法.
IDIV	整数除法.
AAD	除法的 ASCII 码调整.
CBW	字节转换为字. (把 AL 中字节的符号扩展到 AH 中去)
CWD	字转换为双字. (把 AX 中的字的符号扩展到 DX 中去)
CWDE	字转换为双字. (把 AX 中的字符符号扩展到 EAX 中去)
CDQ	双字扩展. (把 EAX 中的字的符号扩展到 EDX 中去)

➤ (6) 伪指令

DW	定义字(2 字节).
PROC	定义过程.
ENDP	过程结束.
SEGMENT	定义段.
ASSUME	建立段寄存器寻址.
ENDS	段结束.
END	程序结束.

➤ (7) 浮点运算指令集

1、控制指令

FINIT	初始化浮点部件
FCLEX	清除异常
FDISI/FENI	浮点检查禁止中断
WAIT/FWAIT	同步 CPU 和 FPU
FNOP	无操作
FXCH	交换 ST(0) 和 ST(1)
FXCH ST(i)	交换 ST(0) 和 ST(i)

FSTSW	ax	状态字到 ax
FSTSW	word ptr mem	状态字到 mem
FLDCW	word ptr mem	mem 到状态字
FSTCW	word ptr mem	控制字到 mem
FLDENV	word ptr mem	mem 到全环境
FSTENV	word ptr mem	全环境到 mem
FRSTOR	word ptr mem	mem 到 FPU 状态
FSAVE	word ptr mem	FPU 状态到 mem
FFREE	ST(i)	标志 ST(i) 未使用
FDECSTP		减少栈指针 1→0 2→1
FINCSTP		增加栈指针 0→1 1→2
FSETPM		浮点设置保护

2、数据传送指令

FLDZ		将 0.0 装入 ST(0)
FLD1		将 1.0 装入 ST(0)
FLDPI		将 π 装入 ST(0)
FLDL2T		将 $\ln 10 / \ln 2$ 装入 ST(0)
FLDL2E		将 $1 / \ln 2$ 装入 ST(0)
FLDLG2		将 $\ln 2 / \ln 10$ 装入 ST(0)
FLDLN2		将 $\ln 2$ 装入 ST(0)
FLD	real4 ptr mem	装入 mem 的单精度浮点数
FLD	real8 ptr mem	装入 mem 的双精度浮点数
FLD	real10 ptr mem	装入 mem 的十字节浮点数
FILD	word ptr mem	装入 mem 的二字节整数
FILD	dword ptr mem	装入 mem 的四字节整数
FILD	qword ptr mem	装入 mem 的八字节整数
FBLD	tbyte ptr mem	装入 mem 的十字节 BCD 数

FST	real4 ptr mem	保存单精度浮点数到 mem
FST	real8 ptr mem	保存双精度浮点数到 mem
FIST	word ptr mem	保存二字节整数到 mem
FIST	dword ptr mem	保存四字节整数到 mem
FSTP	real4 ptr mem	保存单精度浮点数到 mem 并出栈
FSTP	real8 ptr mem	保存双精度浮点数到 mem 并出栈
FSTP	real10 ptr mem	保存十字节浮点数到 mem 并出栈
FISTP	word ptr mem	保存二字节整数到 mem 并出栈
FISTP	dword ptr mem	保存四字节整数到 mem 并出栈
FISTP	qword ptr mem	保存八字节整数到 mem 并出栈
FBSTP	tbyte ptr mem	保存十字节 BCD 数到 mem 并出栈
FCMOVB		ST(0), ST(i) <时传送
FCMOVBE		ST(0), ST(i) <=时传送
FCMOVE		ST(0), ST(i) =时传送
FCMOVNB		ST(0), ST(i) >=时传送
FCMOVNBE		ST(0), ST(i) >时传送
FCMOVNE		ST(0), ST(i) !=时传送
FCMOVNU		ST(0), ST(i) 有序时传送
FCMOVU		ST(0), ST(i) 无序时传送

3、比较指令

FCOM		ST(0)–ST(1)
FCOMI		ST(0), ST(i) ST(0)–ST(1)
FCOMIP		ST(0), ST(i) ST(0)–ST(1) 并出栈
FCOM	real4 ptr mem	ST(0)–实数 mem
FCOM	real8 ptr mem	ST(0)–实数 mem
FICOM	word ptr mem	ST(0)–整数 mem
FICOM	dword ptr mem	ST(0)–整数 mem

FICOMP	word ptr mem	ST(0)-整数 mem 并出栈
FICOMP	dword ptr mem	ST(0)-整数 mem 并出栈
FTST		ST(0)-0
FUCOM	ST(i)	ST(0)-ST(i)
FUCOMP	ST(i)	ST(0)-ST(i) 并出栈
FUCOMPP		ST(0)-ST(1) 并二次出栈
FXAM		ST(0) 规格类型

4、运算指令

FADD		把目的操作数与来源操作数相加，并将结果存入目的操作数
FADDP	ST(i), ST	使目的操作数加上 ST 缓存器，并弹出 ST 缓存器
FIADD		把 ST 加上来源操作数，然后再存入 ST 缓存器
FSUB/FSUBP		减
FSUBR/FSUBRP/FISUB/FISUBR		减数与被减数互换
FMUL/FMULP/FIMUL		乘
FDIV/FDIVP/FDIVR/FDIVRP/FIDIV/FIDIVR		除
FCHS		改变 ST 的正负值
FABS		把 ST 值取出，取绝对值后再存回去。
FSQRT		将 ST 值取出，开根号后再存回去。
FSCALE		计算 $ST * 2^{ST(1)}$ 之值，再把结果存入 ST 里，而 ST(1) 之值不变
FRNDINT		把 ST 的数值舍入成整数，由 FPU 控制字组中的 RC 两个位决定
	RC	舍入控制
	00	四舍五入
	01	向负无限大舍入
	10	向正无限大舍入
	11	向零舍去