

附录 B 8086/8088 汇编语言指令表

助记符	类别	指令格式	操 作	ODITSZAPC
MOV	数据 传送	MOV dst, src	从 src 确定的位置取出源操作数, 或把立即数形式的 src 作为源操作数, 送到目的操作数 dst 确定的位置	-----
PUSH		PUSH src	把字操作数 src 入栈	-----
POP		POP dst	出栈一个字数据, 送到操作数 dst 确定的位置	-----
XCHG		XCHG	把 dst、src 两个操作数中的内容互换	-----
XLAT		XLAT	以 BX+AL 的和作为偏移地址, 从数据段相应位置取出一个字节数据送 AL 寄存器	-----
LEA		LEA dst, src	取操作数 src 在内存的偏移地址, 送到 dst 确定的位置	-----
PUSHF		PUSHF	把标志寄存器 FLAGS 入栈	-----
POPF		POPF	出栈一个字数据, 送到标志寄存器 FLAGS 中	r r r r r r r r
IN		IN AL, src IN AX, src	从 src 指定的外设端口取出一个字节数据送到 AL; 或从 src 指定的外设端口及其下一端口取一个字数据送到 AX 中	-----
OUT		OUT dst, AL OUT dst, AX	把 AL 的值送到 dst 指定的外设端口; 或把 AX 的值送到 dst 指定的外设端口及其下一端口	-----
ADD	算术 运算	ADD dst, src	把两个操作数的值相加, 结果送到 dst 操作数确定的位置	x - - - x x x x x
ADC		ADC dst, src	把两个操作数及 CF 标志位的值三者相加, 结果送到 dst 操作数确定的位置	x - - - x x x x x
INC		INC opr	把 opr 的值加 1 后送回 opr 中	x - - - x x x x -
SUB		SUB dst, src	把 dst 减去 src 的差送回操作数 dst 中	x - - - x x x x x
SBB		SBB dst, src	把 dst 减去 src 的差再减去 CF 的值, 结果送回操作数 dst 中	x - - - x x x x x
DEC		DEC opr	把操作数 opr 的值减 1 送回 opr 中	x - - - x x x x -
NEG		NEG opr	对操作数 opr 的值取反加 1 后送回 opr 中	x - - - x x x x x
CMP		CMP dst, src	用 dst 减去 src, 根据相减情况设置各条件标志位	x - - - x x x x x
MUL		MUL src	无符号乘法: AL 乘以字节操作数 src, 乘积送到 AX 中; 或 AX 乘以字操作数 src, 乘积送到(DX,AX)中	x - - - u u u u x
IMUL		IMUL src	带符号乘法: AL 乘以字节型操作数 src, 乘积送到 AX 中; 或 AX 乘以字型操作数 src, 乘积送到(DX,AX)中	x - - - u u u u x

(续)				
助记符	类别	指令格式	操 作	ODITSZAPC
DIV	算术运算	DIV src	无符号除法: AX 除以 src, 商送 AL, 余数送 AH; 或(DX,AX)除以 src, 商送 AX, 余数送 DX	u - - - u u u u u
IDIV		IDIV src	带符号除法: AX 除以 src, 商送 AL, 余数送 AH; 或(DX,AX)除以 src, 商送 AX, 余数送 DX	u - - - u u u u u
DAA		DAA	把 AL 中的和调整到压缩的 BCD 格式	u - - - x x x x x
DAS		DAS	把 AL 中的差调整到压缩的 BCD 格式	u - - - x x x x x
AAA		AAA	把 AL 中的和调整到非压缩的 BCD 格式, AH 加调整产生的进位值	u - - - u u x u x
AAS		AAS	把 AL 中的差调整到非压缩的 BCD 格式, AH 减调整产生的借位值	u - - - u u x u x
AAM		AAM	把 AH 中的积调整到非压缩的 BCD 格式	u - - - x x u x u
AAD		AAD	实现除法的非压缩 BCD 码调整	u - - - x x u x u
AND	逻辑运算	AND dst, src	两个操作数按各个二进制位进行逻辑与运算, 结果送回 dst 中	0 - - - x x u x 0
OR		OR dst, src	两个操作数按各个二进制位进行逻辑或运算, 结果送回 dst 中	0 - - - x x u x 0
NOT		NOT opr	对 opr 的各个二进制位取反, 结果送回 opr	- - - - - - - -
XOR		XOR dst, src	两个操作数按各个二进制位进行逻辑异或运算, 结果送回 dst 中	0 - - - x x u x 0
TEST		TEST dst, src	两个操作数按各个二进制位进行逻辑与运算, 用计算结果设置标志位	0 - - - x x u x 0
SHL SAL	移位	SHL dst, src	把 dst 的各个二进制位向左移动 src 位, 右边空位填 0, 结果送回 dst, 最后移出的一位送 CF	0 - - - x x u x x
SHR		SHR dst, src	把 dst 的各个二进制位向右移动 src 位, 左边空位填 0, 结果送回 dst, 最后移出的一位送 CF	0 - - - x x u x x
SAR		SAR dst, src	把 dst 的各个二进制位向右移动 src 位, 左边空位填原数最高位的值, 结果送回 dst, 最后移出的一位送 CF	0 - - - x x u x x
ROL	循环移位	ROL dst, src	把 dst 的各个二进制位向左移动 src 位, 从左边移出的位再依次移到右边各空位上, 结果送回 dst, 最后移出的一位送 CF	x - - - - - x

(续)

助记符	类别	指令格式	操 作	ODIT SZAPC
ROR	循 环 移 位	ROR dst,src	把 dst 的各个二进制位向右移动 src 位, 从右边移出的位再依次移到左边各空位上, 结果送回 dst, 最后移出的一位送 CF	x - - - - - x
RCL		RCL dst, src	把 dst 的各个二进制位与 CF 联合在一起向左移动 src 位, 从左边移出的位再依次移到右边各空位上	x - - - - - x
RCR		RCR dst, src	把 dst 的各个二进制位与 CF 联合在一起向右移动 src 位, 从右边移出的位再依次移到左边各空位上	x - - - - - x
MOVS	串 操 作	MOVSB 或 MOVSW	DS 段 SI 所指的一个字节/字送到 ES 段 DI 所指处, 并根据 DF 标志位调整 SI 和 DI, 使其指向下一数据	- - - - -
STOS		STOSB 或 STOSW	把 AL/AX 的值送到 ES 段 DI 所指处, 并根据 DF 标志位调整 DI, 使其指向下一数据	- - - - -
LODS		LODSB 或 LODSW	从 DS 段 SI 所指处取一个字节/字送到 AL/AX 中, 并根据 DF 标志位调整 SI, 使其指向下一数据	- - - - -
CMPS		CMPSB 或 CMPSW	DS 段 SI 所指的一个字节/字与到 ES 段 DI 所指数据相减, 结果设置条件标志位, 并根据 DF 标志位调整 SI 和 DI, 使其指向下一数据	x - - - x x x x x
SCAS		SCASB 或 SCASW	AL/AX 减去 ES 段 DI 所指向的数据, 结果设置条件标志位, 并根据 DF 标志位调整 DI, 使其指向下一数据	x - - - - - x
REP			REP 与 CMPS、SCAS 配合使用, 当 CX 不为 0 重复执行串指令	
REPZ REPE			REPZ 与 CMPS、SCAS 配合使用, 当 CX 不为 0 且 ZF 为 1 时重复执行串指令	
REPNZ REPNE			REPNZ 与 CMPS、SCAS 配合使用, 当 CX 不为 0 且 ZF 为 0 时重复执行串指令	
JMP	跳 转	JMP label	转到 label 处继续执行	- - - - -
JZ JE		JZ label	若 ZF 为 1, 转到 label 处继续执行	- - - - -
JNZ JNE		JNZ label	若 ZF 为 0, 转到 label 处继续执行	- - - - -
JS		JS label	若 SF 为 1, 转到 label 处继续执行	- - - - -

(续)				
助记符	类别	指令格式	操 作	ODITSZAPC
JNS	跳 转	JNS label	若 SF 为 0, 转到 label 处继续执行	-----
JO		JO label	若 OF 为 1, 转到 label 处继续执行	-----
JNO		JNO label	若 OF 为 0, 转到 label 处继续执行	-----
JC JB JNAE		JC label	若 CF 为 1, 即无符号数比较的小于, 转到 label 处继续执行	-----
JNC JNB JAE		JNC label	若 CF 为 0, 即无符号数比较的不小于, 转到 label 处继续执行	-----
JBE JNA		JBE label	若 CF 为 1 或 ZF 为 1, 即无符号数比较的小于或等于, 转到 label 处继续执行	-----
JNBE JA		JNBE label	若 CF 为 0 且 ZF 为 0, 即无符号数比较的大于, 转到 label 处继续执行	-----
JL JNGE		JL label	若 SF 与 OF 不同, 即带符号数比较的小于, 转到 label 处继续执行	-----
JNL JGE		JNL label	若 SF 与 OF 相同, 即带符号数比较的大于或等于, 转到 label 处继续执行	-----
JLE JNG		JLE label	若 SF 与 OF 不同, 或者 ZF 为 1, 即带符号数比较的小于或等于, 转到 label 处继续执行	-----
JNLE JG		JNLE label	若 SF 与 OF 相同, 且 ZF 为 1, 即带符号数比较的大于, 转到 label 处继续执行	-----
JCXZ		JCXZ label	若 CX 为 0, 则转到 label 处继续执行	-----
LOOP		LOOP label	先把 CX 的值减 1 后回送 CX, 再判断当 CX 不为 0 时转到 label 处继续执行	-----
LOOPZ LOOPE		LOOPZ label	先把 CX 的值减 1 后回送 CX, 再判断当 CX 不为 0 且 ZF 为 1 时转到 label 处继续执行	-----
LOOPNZ LOOPNE		LOOPNZ label	先把 CX 的值减 1 后回送 CX, 再判断当 CX 不为 0 且 ZF 为 0 时转到 label 处继续执行	-----
CALL	子程序	CALL dst	根据子程序的类型是 NEAR 还是 FAR, 把 IP 或 CS 及 IP 入栈, 转到子程序继续执行	-----

(续)				
助记符	类别	指令格式	操 作	ODITSZAPC
RET	子程序	RET [n]	NEAR 型子程序中的 RET 将出栈一个字给 IP, FAR 型子程序中的 RET 将出栈两个字依次给 IP 和 CS; 操作数 n 指明出栈后再把 SP 的值加 n	-----
INT	中断	INT n	把标志寄存器 FLAGS、CS、IP 依次入栈, 清 IF 和 TF 标志位, 转 n 号中断服务程序执行	--00-----
INTO		INTO	把标志寄存器 FLAGS、CS、IP 依次入栈, 清 IF 和 TF 标志位, 转 4 号中断服务程序执行	--00-----
IRET		IRET	出栈 3 个字, 依次送 IP、CS、标志寄存器 FLAGS	rrrrrrrrrr
CBW	符号扩展	CBW	若 AL 最高位为 0, 则把 AH 清 0, 否则把 0FFH 送 AH	-----
CWD		CWD	若 AX 最高位为 0, 则把 DX 清 0, 否则把 0FFFFH 送 DX	-----
CLC	标志位控制	CLC	把 CF 标志位清 0	-----0
STC		STC	把 CF 标志位置 1	-----1
CMC		CMC	对 CF 标志位取反	-----x
CLD		CLD	把 DF 标志位清 0	-0-----
STD		STD	把 DF 标志位置 1	-1-----
CLI		CLI	把 IF 标志位清 0	--0-----
STI		STI	把 IF 标志位置 1	--1-----
NOP	处理器控制	NOP	空指令, 不做任何操作	-----
HLT		HLT	停机	-----
WAIT		WAIT	等待	-----
ESC		ESC	换码	-----
LOCK			封锁段前缀	-----

注: (1) 表中只列出的是本书涉及到的指令, 对于 8086/8088 而言, 除了表中列出的指令外, 还有一些不太常用的指令。

(2) 影响标志位的符号说明: 0 — 清 0; 1 — 置 1; x — 根据结果设置; u — 无定义; — — 不影响; r — 恢复原先保存的值。

(3) opr—操作数; src—源操作数; dst—目的操作数。