

8086/80186/80286/80386/80486 Instruction Set

Instruction	Usage	Modifies flags	Operands
AAA - Ascii Adjust for Addition	AAA	AF CF (OF,PF,SF,ZF undefined)	none
AAD - Ascii Adjust for Division	AAD	SF ZF PF (AF,CF,OF undefined)	none
AAM - Ascii Adjust for Multiplication	AAM	PF SF ZF (AF,CF,OF undefined)	none
AAS - Ascii Adjust for Subtraction	AAS	AF CF (OF,PF,SF,ZF undefined)	none
ADC - Add With Carry	ADC d,s	AF CF OF SF PF ZF	r,r m,r r,m r,i m,i a,i
ADD - Arithmetic Addition	ADD d,s	AF CF OF PF SF ZF	r,r m,r r,m r,i m,i a,i
AND - Logical And	AND d,s	CF OF PF SF ZF (AF undefined)	r,r m,r r,m r,i m,i a,i
BSF - Bit Scan Forward	BSF d,s	ZF	r,r r,m r32,r32 r32,m32
BSR - Bit Scan Reverse	BSR d,s	ZF	r,r r,m r32,r32 r32,m32
BSWAP - Byte Swap	BSWAP r32	none	r32
BT - Bit Test	BT d,s	CF	r16,i8 m16,i8 r16,r16 m16,r16
BTC - Bit Test with Compliment	BTC d,s	CF	r16,i8 m16,i8 r16,r16 m16,r16
BTR - Bit Test with Reset	BTR d,s	CF	r16,i8 m16,i8 r16,r16 m16,r16
BTS - Bit Test and Set	BTS d,s	CF	r16,i8 m16,i8 r16,r16 m16,r16
CALL - Procedure Call	CALL d	None	address
CBW - Convert Byte to Word	CBW	None	none
CDQ - Convert Double to Quad	CDQ	None	none
CLC - Clear Carry	CLC	CF	none
CLD - Clear Direction Flag	CLD	DF	none
CLI - Clear Interrupt Flag (disable)	CLI	IF	none
CMC - Complement Carry Flag	CMC	CF	none
CMP - Compare	CMP d,s	AF CF OF PF SF ZF	r,r m,r r,m r,i m,i a,i
CMPS - Compare String (B, W or Dword)	CMPS d,s	AF CF OF PF SF ZF	ES:DI,DS:SI
CMPSCHG - Compare and Exchange	CMPSCHG d,s	AF CF OF PF SF ZF	r,r m,r
CWD - Convert Word to Doubleword	CWD	None	none (AX -> DX:AX)
CWDE - Convert Word to Ext.Dword	CWDE	None	none (AX->EAX)
DAA - Decimal Adjust for Addition	DAA	AF CF PF SF ZF (OF undefined)	none (AL)
DAS - Decimal Adjust for Subtraction	DAS	AF CF PF SF ZF (OF undefined)	none (AL)
DEC - Decrement	DEC d	AF OF PF SF ZF	r8 m r16/32
DIV - Divide	DIV s	AF,CF,OF,PF,SF,ZF undefined	r8 r16 r32 m8 m16 m32 (b:AX/src->q=AL,r=AH, w:DX:AX/src->q=AX,r=DX)
IDIV - Signed Integer Division	IDIV s	AF,CF,OF,PF,SF,ZF undefined	r8 r16 r32 m8 m16 m32
IMUL - Signed Multiply	IMUL s	CF OF (AF,PF,SF,ZF undefined)	r8 r16 r32 m8 m16 m32 (b:AL*src->AX, w:AX*src->DX:AX)
	IMUL d,s		r16,r16 r32,r32 r16,m16 r32,m32 r16,i r32,i
	IMUL d,s,i8		r16,r16,i r32,r32,i r16,m16,i r32,m32,i
IN - Input Byte or Word From Port	IN a,port	None	a,i8 a,DX
INC - Increment	INC d	AF OF PF SF ZF	r8 r16 r32 m
INS - Input String from Port	INS d,port	None	ES:DI,DX
INT - Interrupt	INT num	TF IF	i8
INTO - Interrupt on Overflow	INTO	IF TF	none
IRET/IRETD - Interrupt Return	IRET	AF CF DF IF PF SF TF ZF	none
JA Jump if Above		CF=0 and ZF=0	
JAE Jump if Above or Equal		CF=0	
JB Jump if Below		CF=1	
JBE Jump if Below or Equal		CF=1 or ZF=1	
JC Jump if Carry		CF=1	
JCXZ Jump if		CX Zero CX=0	
JE Jump if Equal		ZF=1	
JG Jump if Greater (signed)		ZF=0 and SF=OF	
JGE Jump if Greater or Equal (signed)		SF=OF	
JL Jump if Less (signed)		SF != OF	
JLE Jump if Less or Equal (signed)		ZF=1 or SF != OF	
JMP Unconditional Jump		unconditional	
JNA Jump if Not Above		CF=1 or ZF=1	
JNAE Jump if Not Above or Equal		CF=1	
JNB Jump if Not Below		CF=0	
JNBE Jump if Not Below or Equal		CF=0 and ZF=0	
JNC Jump if Not Carry		CF=0	
JNE Jump if Not Equal		ZF=0	
JNG Jump if Not Greater (signed)		ZF=1 or SF != OF	
JNGE Jump if Not Greater or Equal (signed)		SF != OF	
JNL Jump if Not Less (signed)		SF=OF	
JNLE Jump if Not Less or Equal (signed)		ZF=0 and SF=OF	
JNO Jump if Not Overflow (signed)		OF=0	
JNP Jump if No Parity		PF=0	
JNS Jump if Not Signed (signed)		SF=0	
JNZ Jump if Not Zero		ZF=0	
JO Jump if Overflow (signed)		OF=1	
JP Jump if Parity		PF=1	
JPE Jump if Parity Even		PF=1	
JPO Jump if Parity Odd		PF=0	
JS Jump if Signed (signed)		SF=1	
JZ Jump if Zero		ZF=1	
JCXZ/JECXZ - Jump if CX is Zero	JCXZ label	None	addr
JMP - Unconditional Jump	JMP target	None	addr
LAHF - Load Rister AH From Flags	LAHF	None	none

8086/80186/80286/80386/80486 Instruction Set

LDS - Load Pointer Using DS	LDS d,s	None	r16,m32 r,m
LEA - Load Effective Address	LEA d,s	None	r,m
LES - Load Pointer Using ES	LES d,s	None	
LFS - Load Pointer Using FS			
LGS - Load Pointer Using GS			
LODS - Load String (Byte, Word or Double)	LODS s	None	DS:SI->accumulator
LOOP - Decrement CX and Loop if CX Not Zero	LOOP label	None	
LOOPE - Loop While Equal	LOOPE label	None	
LOOPZ - Loop While Zero	LOOPZ label	None	
LOOPNZ - Loop While Not Zero	LOOPNZ label	None	
LOOPNE - Loop While Not Equal	LOOPNE label	None	
LSS - Load Pointer Using SS			
MOV - Move Byte or Word	MOV d,s	None	
MOVS - Move String (Byte or Word)	MOVS d,s	None	DS:SI->ES:DI
MOVSB - Move with Sign Extend	MOVSB d,s	None	
MOVZX - Move with Zero Extend	MOVZX d,s		
MUL - Unsigned Multiply	MUL s	CF OF (AF,PF,SF,ZF undefined)	
NEG - Two's Complement Negation	NEG d	AF CF OF PF SF ZF	
NOP - No Operation (90h)	NOP	None	
NOT - One's Complement Negation (Logical NOT)	NOT d	None	
OR - Inclusive Logical OR	OR d,s	CF OF PF SF ZF (AF undefined)	
OUT - Output Data to Port	OUT port,a	None	
OUTS - Output String to Port	OUTS port,s	None	
POP - Pop Word off Stack	POP d	None	
POPA/POPAD - Pop All Registers onto Stack	POPA	None	
POPF/POPFD - Pop Flags off Stack	POPF		
POPFD	all flags		
PUSH - Push Word onto Stack	PUSH s		
PUSHA/PUSHAD - Push All Registers onto Stack	PUSHA		
PUSHAD	None		
PUSHF/PUSHFD - Push Flags onto Stack	PUSHF		
PUSHFD	None		
RCL - Rotate Through Carry Left	RCL d,count	CF OF	
RCR - Rotate Through Carry Right	RCR d,count	CF OF	
REP - Repeat String Operation	REP	None	
REPE/REPZ - Repeat Equal / Repeat Zero	REPE	None	
REPNE/REPNZ - Repeat Not Equal / Repeat Not Zero	REPNE	None	
RET/RETF - Return From Procedure	RET nBytes		
ROL - Rotate Left	ROL d,count	CF OF	
ROR - Rotate Right	ROR d,count	CF OF	
SAHF - Store AH Register into FLAGS	SAHF	AF CF PF SF ZF	
SAL/SHL - Shift Arithmetic Left / Shift Logical Left	SAL d,count	CF OF PF SF ZF (AF undefined)	
	SHL d,count	CF OF PF SF ZF (AF undefined)	
	SAR d,count	CF OF PF SF ZF (AF undefined)	
SAR - Shift Arithmetic Right	SAR d,count	CF OF PF SF ZF (AF undefined)	
SBB - Subtract with Borrow/Carry	SBB d,s	AF CF OF PF SF ZF	
SCAS - Scan String (Byte, Word or Doubleword)	SCAS string	AF CF OF PF SF ZF	
SETAE/SETNB - Set if Above or Equal / Set if Not Below	SETAE d	none	r8 m8
SETB/SETNAE - Set if Below / Set if Not Above or Equal			
SETBE/SETNA - Set if Below or Equal / Set if Not Above			
SETE/SETZ - Set if Equal / Set if Zero			
SETNE/SETNZ - Set if Not Equal / Set if Not Zero			
SETL/SETNGE - Set if Less / Set if Not Greater or Equal			
SETGE/SETNL - Set if Greater or Equal / Set if Not Less			
SETLE/SETNG - Set if Less or Equal/Set if Not greater or Equal			
SETG/SETNLE - Set if Greater / Set if Not Less or Equal			
SETS - Set if Signed			
SETNS - Set if Not Signed			
SETC - Set if Carry			
SETNC - Set if Not Carry			
SETO - Set if Overflow			
SETNO - Set if Not Overflow			
SETP/SETPE - Set if Parity / Set if Parity Even			
SETNP/SETPO - Set if No Parity / Set if Parity Odd			
SHL - Shift Logical Left See: SAL			
SHR - Shift Logical Right	SHR d,count	CF OF PF SF ZF (AF undefined)	
SHLD/SHRD - Double Precision Shift	SHLD d,s,count	CF PF SF ZF (OF,AF undefined)	
	SHRD d,s,count	CF PF SF ZF (OF,AF undefined)	
STC - Set Carry	STC	CF	none
STD - Set Direction Flag	STD	DF	none
STI - Set Interrupt Flag	STI	IF	none
STOS - Store String (B,W,Dword)	STOS d	None	accumulator->ES:DI
SUB - Subtract	SUB d,s	AF CF OF PF SF ZF	r,r m,r r,i m,i a,i
TEST - Test For Bit Pattern	TEST d,s	CF OF PF SF ZF (AF undefined)	r,r m,m r,r i m,i a,i
XCHG - Exchange	XCHG d,s	None	r,r m,r r,m a,r r,a
XLAT/XLATB - Translate	XLAT X-table	None	[BX+AL]->AL
XOR - Exclusive OR	XOR d,s	CF OF PF SF ZF (AF undefined)	r,r m,r r,m r,i m,i a,i