Command	Detail	Technical	Opcode	Clocks	Bytes	O D I T S Z A P C	Addressing modes
AAA	ASCII Adjust for Addition (For BCD)		37	4	1	U - - - U U X U X	
AAD	ASCII Adjust for Division		D5 0A	60	2	U - - - X X U X U	
AAM	ASCII Adjust for Multiply		D4 0A	83	1	U - - - X X U X U	
AAS	ASCII Adjust for Subtraction		3F	4	1	U - - - U U X U X	
ADC dest, src	Add with Carry	ADC AX,SI		3+	2-6	X - - - X X X X X	r,r r,m m,r r,i m,i a,i
ADD dest,src	The sum of the two operands	ADD CX,DX		3+	2-6	X - - - X X X X X	r,r r,m m,r r,i m,i ,a,i
AND dest,src	logical and	AND AL,BL		3+	2-6	0 - - - X X U X 0	r,r r,m m,r r,i m,i a,i
CALL procedure-name	call sub	CALL label… CALL AX		19+	2-5	- - - - - - - - -	II III memptr regptr memptr32
CBW	Convert Byte to Word... Sign Extend AL to AX	CBW	98	2	1	- - - - - - - - -	r,r r,m m,r r,i m,i a,i
CLC	Clear Carry flag	CF=0	F8	2	1	- - - - - - - - 0	
CLD	Clear direction flag	DF=0	FC	2	1	- - 0 - - - - - -	
CLI	Clear Interrupt-enable flag	IF=0	FA	2	1	- - 0 - - - - - -	
CMC	Complement carry flag	CF=!CF	F5	2	1	- - - - - - - - X	
CMP dest, src	Compare	CMP ah,40… CMP bx,cx		3+	2-6	X - - - X X X X X	r,r r,m m,r r,i m,i a,i
CMPS	Compare String... DS:SI to ES:DI… result in ZF	REPE CMPS 10,KEY	A6/A7	22+	1	X - - - X X X X X	ds,ss
CWD	Conv Word to DblW... sign extend AX to DX:AX	DX:AX=AX		5	1	- - - - - - - - -	
DAA	Decimal Adjust for Addition		27	4	1	X - - - X X X X X	
DAS	Decimal Adjust for Subtraction			4	1	U - - - X X X X X	
DEC dest	Decrement			2	1	X - - - X X X X -	rr r m
DIV source	divide			80+	2-5	U - - - U U U U U	r rr m mm
ESC externa/-opcode, src	Escape (for external cpu) (0-63)			2+	2-4	- - - - - - - - -	i,m i,r
HLT	Halt (stop CPU until reset / nmi)		F4	2	1	- - - - - - - - -	
IDIV source	Integer Divide	IDIV BL... IDIV CX		101+	2-4	U - - - U U U U U	r rr m mm
IMUL source	Integer Multiply			80+	2-4	X - - - U U U U U	r rr m mm
IN accumulator,port	Input byte or word (use DX for 16 bit port)	in ax,dx… in ax,1Fh		8+	1-2	- - - - - - - - -	a,i a,DX
INC dest	Increment			2+	1-4	X - - - X X X X -	rr r m
INT interrupt-type	Interrupt (0-255)			51/52	1-2	- - 0 0 - - - - -	i
INTO	Interrupt if overflow flag set			61	1	- - 0 0 - - - - -	
INTR*	External Maskable Interrupt			61	0?!	- - 0 0 - - - - -	
IRET	Interrupt Return		CF	24	1	R R R R R R R R R	
JA/JNBE short-label	above/not below nor equal	(CF OR ZF)=O		4/16	2	- - - - - - - - -	I
JAE/JNB short-label	above or equal/ not below	(CF OR ZF)=O		4/16	2	- - - - - - - - -	I
JB/JNAE short-label	below / not above nor equal	CF=1		4/16	2	- - - - - - - - -	I
JBE/JNA short-label	below or equal/ not above	(CF OR ZF)=1		4/16	2	- - - - - - - - -	I
JC short-label	carry	CF=1		4/16	2	- - - - - - - - -	I
JCXZ short-label	Jump If CX Zero (see loop)			4/16	2	- - - - - - - - -	I
JE/JZ short-label	equal/zero	ZF=1		4/16	2	- - - - - - - - -	I
JG/JNLE short-label	greater/ not less nor equal	((SF XOR OF) OR ZF)=O		4/16	2	- - - - - - - - -	I
JGE/JNL short-label	greater or equal/not less	(SF XOR OF)=O		4/16	2	- - - - - - - - -	I
JLE/JNG short-label	less or equal/ not greater	((SF XOR OF) OR ZF)=1		4/16	2	- - - - - - - - -	I
JLIJNGE short-label	less/not greater nor equal	(SF XOR OF)=1		4/16	2	- - - - - - - - -	I
JMP target	Jump to label (byte or word target)			11+	2	- - - - - - - - -	I II III memptr regptr memptr32
JNC short-label	not carry	CF=O		4/16	2	- - - - - - - - -	I
JNE/JNZ short-label	not equal/ not zero	ZF=O		4/16	2	- - - - - - - - -	I
JNO short-label	not overflow	OF=O		4/16	2	- - - - - - - - -	I
JNP/JPO short-label	not parity / parity odd	PF=O		4/16	2	- - - - - - - - -	I
JNS short-label	not sign	SF=O		4/16	2	- - - - - - - - -	I
JO short-label	overflow	OF=1		4/16	2	- - - - - - - - -	I
JP/JPE short-label	parity/ parity equal	PF=1		4/16	2	- - - - - - - - -	I
JS short-label	sign	SF=1		4/16	2	- - - - - - - - -	I
LAHF	Load AH from flags (see SAHF)	AH=F		4	1	- - - - - - - - -	
LDS dest,source	Load Pointer using DS(Set DS and Dest)	lds di,[es:si]		16+	2-4	- - - - - - - - -	rr,mmm
LEA dest, source	Load effective address		8D n	2+	2-4	- - - - - - - - -	rr,mm
LES dest, source	Load Pointer using ES			16+	2-4	- - - - - - - - -	rr,mmm
LOCK	Lock bus signal		F0	2	1	- - - - - - - - -	
LODSB/W	Load String Copy DS:SI to AL/AX		AC/AD	12+	1	- - - - - - - - -	ss
LOOP short-label	dec CX and loop if CX!=0 (See JCXZ)		E2	17/5	2	- - - - - - - - -	I
LOOPNZ short-label (or LOOPNE)	dec CX and loop if nz		E0	19/5	2	- - - - - - - - -	I
LOOPZ short-label (or LOOPE)	dec CX and loop if z		E1	18/6	2	- - - - - - - - -	I
MOV dest,source	Move			2+	2-6	- - - - - - - - -	m,a a,m r,r r,m m,r r,i m,i sr,rr sr,mm rr,sr mm,sr
MOVSB/W (like LDI)	Move String	Move DS:SI to ES:DI	A4/A5	18+	1	- - - - - - - - -	ds,ss
MUL source	Multiply AX= AL*byte / DX:AX = AX*word	MUL BL... MUL addr		70+	2-4	X - - - U U U U X	r rr m mm
NMI*				50	0?!	- - 0 0 - - - - -	
NEG dest	Negate	NEG AL.. NEG BX... NEG addr		3+	2-4	X - - - X X X 1*	r rr m
NOP	No Operation		90	3	1	- - - - - - - - -	
NOT dest	inverts the bits	NOT AL... NOT AX... NOT addr		3+	2-4	- - - - - - - - -	r rr m
OR dest,src	inclusive or			3+	2-6	0 - - - X X U X 0	r,r r,m m,r r,i m,i a,i
OUT port,accumulator	Output byte or word (use DX for 16 bit port)	out dx,ax… out 1Fh,ax		8-10	1-2	- - - - - - - - -	i,a DX,a
POP dest	Pop a word off the stack	POP DX... POP DX... POP WORD PTR [addr]		8+	1-4	- - - - - - - - -	rr sr mm
POPF	Pop flags off stack		9D	8-10	1	R R R R R R R R R	
PUSH source	Push word onto stack	PUSH DX... PUSH DX... PUSH WORD PTR [addr]		10+	1-4	- - - - - - - - -	rr sr mm
PUSHF	Push flags onto stack		9C	10	1	- - - - - - - - -	
RCL dest, count	Rotate through Carry Left	count must be 1 on 086		2+	2-4	X - - - - - - - X	r,1 r,CL m,1 m,CL
RCR dest, count	Rotate through Carry Right	count must be 1 on 086		2+	2-4	X - - - - - - - X	r,1 r,CL m,1 m,CL
REP cmd	Repeat cmd CX times	Cmd = INS/LODS/MOVS/OUTS/STOS	F3	2	1	- - - - - - - - -	
REPE / **REPZ** cmd	Repeat While Zero	cmd =CMPS/SCAS	F3	2	1	- - - - - - - - -	
REPNE / REPNZ cmd	Repeat While Not Zero	cmd =CMPS/SCAS	F2	2	1	- - - - - - - - -	
RET optional-pop-value	Return from sub (and pop n bytes off stack)	N= 0-64k		8-17	1-3	- - - - - - - - -	- i ii
ROL dest, count	Rotate Left (no carry)	count must be 1 on 086		2+	2-4	X - - - - - - - X	r,1 r,CL m,1 m,CL
ROR dest, count	Rotate Right (no carry)	count must be 1 on 086		2+	2-4	X - - - - - - - X	r,1 r,CL m,1 m,CL
SAHF	Store AH into flags (see LAHF)		9E	4	1	- - - - R R R R R	
SAL dest, count	Shift Arithmetic Left	count must be 1 on 086		2+	2-4	X - - - - - - - X	r,1 r,CL m,1 m,CL
SAR dest, count	Shift Arithmetic Right	count must be 1 on 086		2+	2-4	X - - - X X U X X	r,1 r,CL m,1 m,CL
SBB dest, src	Subtract with Borrow			3	2-6	X - - - X X X X X	r,r r,m m,r a,i r,i m,i
SCASB/W	Scan String... cmp AL/AX to ES:DI… result in ZF	SCAS	AE/AF	15+	1	X - - - X X X X X	ds
Segment*	Override to specific segment	MOV SS:parameter,AX		2	1	- - - - - - - - -	
SHL dest, count	Shift Logical Left	count must be 1 on 086		2+	2-4	X - - - - - - - X	r,1 r,CL m,1 m,CL
SHR dest, src	Shift Logical Right	count must be 1 on 086		2+	2-4	X - - - - - - - X	r,1 r,CL m,1 m,CL
SINGLE STEP	Trap flag interrupt	Interrupt if TF=1		2+	2-4	- - 0 0 - - - - -	
STC	Set Carry Flag	CF=1	F9	2	1	- - - - - - - - 1	
STD	Set direction flag	DF=0	FD	2	1	- - 1 - - - - - -	
STI	Set interrupt enable flag	IF=1	FB	2	1	- - 1 - - - - - -	
STOSB/W dest-string	Store String Byte/Word in ES:DI	STOS	AA/AB	11+	1	- - - - - - - - -	ds
SUB dest,src	Subtraction	SUB CX,BX		3+	2-6	X - - - X X X X X	r,r r,m m,r a,i r,i m,i
TEST dest, src	sets flags like AND without changing regs		9B	3+	2-6	0 - - - X X U X 0	r,r r,m m,r a,i r,i m,i
WAIT	Wait while test/busy not active (high)			3+5n	1	- - - - - - - - -	
XCHG dest,src	Exchange byte/word register	XCHG AX,BX... XCHG AL,BL... XCHG addr,AX		3+	1-4	- - - - - - - - -	a,r m,r r,r
XLAT	Translate (Read from LUT)... AL = [DS:BX+AL]	XLAT	D7	11	1	- - - - - - - - -	
XOR dest, src	Exclusive Or			3+	2-6	0 - - - X X U X 0	r,r r,m m,r a,i r,i m,i

186 Commands	Detail	Technical	Opcode	Clocks	Bytes	O D I T S Z A P C	Addressing modes
BOUND	Check Array Index Against Bounds		62	10		- - - - - Z - - -	
ENTER	Make Stack Frame for Procedure Parameters		C8	10/15+4		- - - - - - - - -	
INS	Input from Port to String		6C	9/29		- - - - - - - - -	
INSB	Input from Port to String		6D	9/29		- - - - - - - - -	
INSW	Input from Port to String		6D	9/29		- - - - - - - - -	
LEAVE	High Level Procedure Exit		C9			- - - - - - - - -	
OUTS	Output String to Port			8/28		- - - - - - - - -	
OUTSB	Output String to Port			8/28		- - - - - - - - -	
OUTSW	Output String to Port			8/28		- - - - - - - - -	
POPA	Pop all General Registers	DI,SI,BP,skipped,BX,DX,CX,AX		24		- - - - - - - - -	
PUSHA	Push all General Registers	AX,CX,DX,BX,SP,BP,SI,DI	60	18		- - - - - - - - -	
PUSHW							