

## Ausgewählte Prozessorbefehle der Intel-8086-Familie

Assemblerbefehl / Mnemonic	Ab 80x	M-Zyklen		Erklärung	O	D	I	T	S	Z	A	P	C
		486	Pro										
Arithmetische Befehle													
ADC Ziel, Quelle		1/3	2/4	Ziel+Quelle+Carry, Ergebnis in Ziel									
ADD Ziel, Quelle		1/3	1	Ziel+Quelle, Ergebnis in Ziel									
CBW		3	1	Das Byte in AL vorzeichenrichtig in ein Wort AX wandeln.									
CMP Op1, Op2		1/2	1/2	Op1 mit Op2 vergleichen: Op1-Op2, nur Flags setzen									
CMPXCHG Op1, Op2	486	6/10	cplx	Op1 mit Akkumulator vergleichen und Op1,Op2 tauschen									
DEC Operand		1/3	1/4	Operand - 1									
DIV Operand		16, 40	3/4	Passt Ergebnis nicht in das Register, dann Interrupt 0 (Überlauf) Dividend/Quotient/Rest (Division vorzeichenlos): AX/AL/AH, DX:AX/AX/DX, EDX:EAX/EAX/EDX	?					?	?	?	?
IDIV Operand		19 27 44	3/4	Passt Ergebnis nicht in das Register, dann Interrupt 0 (Überlauf) Dividend/Quotient/Rest (Integerdivision): AX/AL/AH, DX:AX/AX/DX, EDX:EAX/EAX/EDX	?					?	?	?	?
IMUL Faktor IMUL Ziel, Faktor	386	13-18	1/2	1 Operand: Operand*AL=AX / Operand*AX=DX:AX / Operand*EAX=EDX:EAX 2 Operanden: Ergebnis steht im ersten Operanden						?	?	?	?
INC Operand		1/3	1/4	Operand + 1									
MUL Operand		13-42	1-4	Multiplikation ohne Vorzeichen, Op*AL=AX / Op*AX=DX:AX / Op*EAX=EDX:EAX; CF und OF =0, wenn höherwertiges Byte, Word oder DW =0, sonst CF=OF=1						?	?	?	?
NEG Operand		1/3	1/4	Zweierkomplement des Operanden bilden (Negation) CF=0, wenn Operand 0, sonst CF=1									
SBB Ziel, Quelle		1	2	Ziel=Ziel-Quelle-CF									
SUB Ziel, Quelle		1/3	1/4	Ziel=Ziel-Quelle									
XADD Ziel, Quelle	486	3/4	colx	Ziel und Quelle werden vertauscht und dann addiert									
Boolesche Befehle													
AND Ziel, Quelle		1/3	1, 1/4	Ziel bitweise UND-verknüpfen mit Quelle									
NOT Operand		1/3	1/4	Einer-Komplement (stellenweises Invertieren)									
OR Ziel, Quelle		1/3	1/4	Bitweises Verknüpfen von Ziel und Quelle durch Inklusiv-ODER	0						?		0
XOR Ziel, Quelle		1-3	1-4	Ziel und Quelle werden bitweise Exklusiv-ODER verknüpft	0						?		0
Schiebebefehle													
RCL Operand, Anzahl		8-31	cplx	Linksrotieren mit Carry-Flag, CF geht ins LSB									
RCR Operand, Anzahl		8-31	cplx	Rechtsrotieren mit Carry-Flag, Bit 0 geht ins CF									
ROL Operand, Anzahl		3/4	1/4	Linksrotieren, MSB geht in CF und Bit 0 (LSB)									
ROR Operand, Anzahl		3/4	1/4	Rechtsrotieren, LSB geht in CF und MSB									
SAL Operand, Anzahl		3/4	1/4	Operand wird Anzahl * bitweise nach links, MSB zu CF, LSB=0									
SAR Operand, Anzahl		3/4	1/4	Operand wird Anzahl * bitweise nach rechts, LSB zu CF, MSB=CF								?	
Flag-Manipulation													
CLC		2	1	Carry-Flag löschen									0
CLI		5	cplx	Interrupt-Enable-Flag löschen		0							
CMC		2	1	Carry-Flag komplementieren									

Assemblerbefehl / Mnemonic	Ab 80x	M-Zyklen		Erklärung	O	D	I	T	S	Z	A	P	C
		486	PPro										
STC		2	1	Carry-Flag setzen									1
STI		5	cplx	Interrupt-Flag setzen			1						
Übertragungsbefehle													
LAHF		3	1	Lade AH mit dem niederwertigen Byte des Flagwortes									
LEA Ziel, Quelle		1	1	Lade effektive Adresse									
MOV Ziel, Quelle		1... 16	1 cplx	Inhalt von Quelle nach Ziel transportieren									
POP Ziel		3-6	Cplx	Ziel vom Stapel nehmen (CS nicht erlaubt), Stackpointer+2									
POPA (POPAD)	286	9	cplx	Registerinhalte vom Stapel nehmen (siehe PUSHA)									
POPF (POPCD)		9	cplx	Flags vom Stapel nehmen, Stackpointer+2 (+4)									
PUSH Quelle		4,1	4,3	Auf den Stapel legen, Stackpointer-2									
PUSHA (PUSHAD)		11	cplx	AX,CX,DX,BX,SP,BP,SI,DI in dieser Reihenfolge auf den Stack									
PUSHF (PUSHFD)		4	cplx	Stackpointer-2 (-4), Flags auf den Stapel bringen									
XCHG Op1, Op2		3/5	3	Inhalte von Op1 und Op2 tauschen									
Sprungbefehle													
CALL Name CALL Adresse		3,...	4, cplx	Unterprogrammaufruf, Rücksprungadresse auf Stapel sichern									
IRET		15- 32	cplx	Rückkehr von Unterbrechung, Zustand vom Stapel holen									
JA Ziel		3/1	1	CF=0 und ZF=0: Sprung zu Ziel, wenn vorzeichenlos größer									
JAE Ziel		3/1	1	CF=0 : Sprung zu Ziel, wenn vorzeichenlos größer oder gleich									
JB Ziel		3/1	1	CF=1 : Sprung zu Ziel, wenn vorzeichenlos kleiner									
JBE Ziel		3/1	1	CF=1 und ZF=1 : Sprung zu Ziel, wenn vorzeichenlos kleiner oder gleich									
JC Ziel		3/1	1	CF=1 : Sprung zu Ziel, wenn Carry-Flag gesetzt									
JCXZ Ziel		3/1	2	CX=0 : Sprung zu Ziel, wenn CX=0									
JE Ziel		3/1	1	ZF=1 : Sprung zu Ziel, wenn gleich									
JECXZ	386	3/1	2	ECX=0 : Sprung zu Ziel, wenn ECX=0									
JG Ziel		3/1	1	SF=OF und ZF=0 : Sprung zu Ziel, wenn vorzeichenrichtig größer									
JGE Ziel		3/1	1	SF=OF: Sprung zu Ziel, wenn vorzeichenrichtig größer oder gleich									
JL Ziel		3/1	1	SF<>OF: Sprung zu Ziel, wenn vorzeichenrichtig kleiner									
JLE Ziel		3/1	1	SF<>OF oder ZF=1: Sprung zu Ziel, wenn kleiner oder gleich									
JMP Ziel		3,...	1, cplx	Unbedingter Sprung									
JNA Ziel		3/1	1	CF=1 oder ZF=1: Sprung zu Ziel, wenn vorzeichenlos nicht größer									
JNAE Ziel		3/1	1	CF=1: Sprung zu Ziel, wenn vorzeichenlos nicht größer oder gleich									
JNB Ziel		3/1	1	CF=0: Sprung zu Ziel, wenn vorzeichenlos nicht kleiner									
JNBE Ziel		3/1	1	CF=0 oder ZF=0: Sprung zu Ziel, wenn vorzeichenlos nicht kleiner oder gleich									
JNC Ziel		3/1	1	CF=0: Sprung zu Ziel, wenn kein Carry									

## Ausgewählte Prozessorbefehle der Intel-8086-Familie

Assemblerbefehl / Mnemonic	Ab 80x	M-Zyklen		Erklärung	O	D	I	T	S	Z	A	P	C
		486	PPro										
JNE Ziel		3/1	1	ZF=0: Sprung zu Ziel, wenn nicht gleich									
JNG Ziel		3/1	1	SF<>OF oder ZF=1: Sprung zu Ziel, wenn vorzeichenrichtig nicht größer									
JNGE Ziel		3/1	1	SF<>OF: Sprung zu Ziel, wenn vorzeichenrichtig nicht größer oder gleich									
JNL Ziel		3/1	1	SF=OF: Sprung zu Ziel, wenn vorzeichenrichtig nicht kleiner									
JNLE Ziel		3/1	1	SF=OF und ZF=0: Sprung zu Ziel, wenn vorzeichenrichtig nicht kleiner oder gleich									
JNO Ziel		3/1	1	OF=0: Sprung zu Ziel, wenn kein Überlauf									
JNP Ziel		3/1	1	PF=0: Sprung zu Ziel, wenn Parity-Flag nicht gesetzt									
JNS Ziel		3/1	1	SF=0: Sprung zu Ziel, wenn Sign-Flag nicht gesetzt									
JNZ Ziel		3/1	1	ZF=0: Sprung zu Ziel, wenn nicht Null									
JO Ziel		3/1	1	OF=1: Sprung zu Ziel, wenn Überlauf									
JP Ziel		3/1	1	PF=1: Sprung zu Ziel, wenn Parity-Flag gesetzt									
JPE Ziel		3/1	1	PF=1: Sprung zu Ziel, wenn Parität gerade									
JPO Ziel		3/1	1	PF=0: Sprung zu Ziel, wenn Parität ungerade									
JS Ziel		3/1	1	SF=1: Sprung zu Ziel, wenn negativ (Sign-Flag gesetzt)									
JZ Ziel		3/1	1	ZF=1: Sprung zu Ziel, wenn Null									
RET (RET Wert)		5-13	cplx	Rücksprung aus Prozedur									
Bitoperationen													
BSF Ziel, Quelle	386	6-43	2/3	Bit-Suchen vorwärts (sucht erste 1 in Quelle, Position in Ziel)									
BSR Ziel, Quelle	386	6-104	2/3	Bit-Suchen rückwärts (sucht erste 1 in Quelle, Position in Ziel)									
BT Quelle, Position	386	3/8	1/2	Sucht in Quelle das mit Position angegebene Bit, Kopie in C-Flag									
Sonstige													
HLT		4	cplx	Prozessor-Halt, Verlassen durch Reset oder Interrupt									
IN Ziel, Kanal-Nr		14...28	cplx	Operanden: AL/AX/EAX/Konstante/DX Kanal-Nr: 1-Byte-Konstante (0...255)									
INT Nummer		26	cplx	Software-Interrupt auslösen, rettet CP, IP und Flags									
NOP		1	1	Keine Operation									
OUT Kanal, Quelle		16	cplx	Kanalnummer: 1-Byte-Konstante oder DX Quelle: AL, AX, EAX									

cplx= komplexer Befehl

Flag-Register 8086...80286:

15													0
	N	IOP	O	D	I	T	S	Z		A		P	C

## Ausgewählte Funktionen des DOS-Interrupts 21H:

Funktion (Register AH)	Eingabe	Wirkung
01H		Warten bis Tastatureingabe und Anzeige auf Bildschirm (Echo) Eingegebenes Zeichen in Register AL (ASCII-Code)
02H		Anzeige von (DL) auf Bildschirm
05H	Auszugebendes Zeichen in Register DL	Ausgabe in Drucker-Puffer
06H	(DL)=FFH (255)	Lesen des Tastaturpuffers Kein Zeichen im Tastaturpuffer: ZF=1 Zeichen im Tastaturpuffer. ZF=0 eingegebenes Zeichen in Register AL
08H		Warten auf Tastatureingabe eingegebenes Zeichen in Register AL
09H	DS:DX Zeichenkette	Zeichenketten-Ausgabe auf BS
0AH	DS:DX Eingabe-Puffer	Zeichenkette-Eingabe über Tastatur bis Ende-Zeichen in Form der Eingabe-Taste (CR) 1. Byte im Puffer: Maximalzahl der Bytes im Puffer einschl. CR (muss vorgegeben werden) 2. Byte im Puffer: Momentan eingegebene Anzahl Bytes ohne CR (wird automatisch hochgezählt)
Für alle Funktionen außer 09H gilt: Bei Eingabe von STRG+C oder STRG+Pause definierte Rückkehr nach DOS		
4CH	Definierte Rückkehr nach DOS (mit Übergabe der im Register AL stehenden Zahl an das aufrufende Programm)	