Adresse:

http://www.configsystems.de/Hilfe-AssemblerBefehlsreferenz.html

8086/88 Assembler Befehlsreferenz

Der 8086/88 Mikroprozessor von Intel arbeitet mit 16 Bit. Er ist der Vorgänger der 80x86-Familie. Die Architektur entstand aus den Intel-CPUs 8080 und 8085, der Befehlssatz ist so gestaltet, dass Assembler-Quellcode für den 8080/8085 leicht in gültigen 8086/88-Quellcode umgesetzt werden kann. Dem 8086/88 fehlen einige wesentliche Bausteine, Interrupt- und DMA-Controller sind nicht vorhanden, diese kommen als externe Chips dazu. Die Speichersegmentierung ist eine Ausnahme: Die Segmente sind Adreßbereiche von je 64 kB Größe, die in 16-Byte-Schritten im Speicher verschoben werden können. Auf diese Weise ist es der CPU möglich, bis zu einem MB zu adressieren, wobei innerhalb jedes Segments mit 16-Bit-Adressen gearbeitet werden kann. Vorteil ist eine einfaches Portieren von 8-Bit-Programmen, Nachteil ist die umständliche Programmierung.

Abkürzung	Englisch	Deutsch
AAA	ASCII adjust after addition	ASCII-Anpassung nach einer Addition
AAD	ASCII adjust before divide	ASCII-Anpassung vor einer Division
AAM	ASCII adjust alter multiply	ASCII-Anpassung nach einer Multiplikation
AAS	ASCII adjust after subtract	ASCII-Anpassung nach einer Subtraktion
ADC	Add with carry	Addiere mit Übertrag-Statusbit
ADD	Add	Addiere
AND	And	Und
CALL	Call	Rufe auf
CBW	Convert byte to word	Wandle Byte in Wort um
CLC	Clear carry flag	Lösche Übertrag-Statusbit
CLD	Clear direction flag	Lösche Richtung-Statusbit
CLI	Clear interrupt flag	Lösche Unterbrechung-Statusbit
CMC	Complement carry flag	Invertiere Übertrag-Statusbit
CMP	Compare	Vergleiche
CMPS	Compare string	Vergleiche Zeichenkette
CMPSB	Compare string bytewise	Vergleiche Zeichenkette byteweise
CMPSW	Compare string wordwise	Vergleiche Zeichenkette wortweise
CWD	Convert word to double word	Wandle Wort in Doppelwort um
DAA	Decimal adjust after addition	Dezimale Anpassung nach einer Addition
DAS	Decimal adjust after subtract	Dezimale Anpassung nach einer Subtraktion
DEC	Decrement	Vermindere
DIV	Divide (unsigned)	Dividiere (vorzeichenlos)
ESC	Escape (to external device)	Entweiche (zu externer Einheit)
HLT	Halt	Halt
IDIV	Integer divide (signed)	Ganzzahlige Division (vorzeichenbehaftet)
IMUL	Integer multiply (signed)	Ganzzahlige Division Ganzzahlige Multiplikation (vorzeichenbehaftet)
IN	Input from	Lese ein von

INC	Increment	Erhöhe
INT	Interrupt	Unterbreche
INTO	Interrupt if overflow	Unterbreche, wenn Überlauf
IRET	Interrupt return	Kehre von Unterbrechungsprogramm zurück
JA	Jump if above	Springe, wenn darüber
JAE	Jump if above or equal	Springe, wenn darüber oder gleich
JB	Jump if below	Springe, wenn darunter
JBE	Jump if below or equal	Springe, wenn darunter oder gleich
JCXZ	Jump if CX equal zero	Springe, wenn CX gleich 0
JE	Jump if equal	Springe, wenn gleich
JG	Jump if greater	Springe, wenn größer
JGE	Jump if greater or equal	Springe, wenn größer oder gleich
JL	Jump if less	Springe, wenn kleiner
JLE	Jump if less or equal	Springe, wenn kleiner oder gleich
JMP	Jump	Springe
JNA	Jump if not above	Springe, wenn nicht darüber
JNAE	Jump if not above and not equal	Springe, wenn nicht darüber und nicht gleich
JNB	Jump if not below	Springe, wenn nicht darunter
JNBE	Jump if not below and not equal	Springe, wenn nicht darunter und nicht gleich
JNE	Jump if not equal	Springe, wenn nicht gleich
JNG	Jump if not greater	Springe, wenn nicht größer
JNGE	Jump if not greater and not equal	Springe, wenn nicht größer und nicht gleich
JNL	Jump if not less	Springe, wenn nicht kleiner
JNLE	Jump if not less and not equal	Springe, wenn nicht kleiner und nicht gleich
JNO	Jump if no overflow	Springe, wenn kein Überlauf
JNP	Jump if no parity	Springe, wenn keine Parität
JNS	Jump if no sign	Springe, wenn kein Vorzeichen (positiv)
JNZ	Jump if not zero	Springe, wenn nicht 0
JO	Jump if overflow	Springe, wenn Überlauf
JP	Jump if parity	Springe, wenn Parität
JPE	Jump if parity even	Springe, wenn Parität gerade
JPO	Jump if parity odd	Springe, wenn Parität ungerade
JS	Jump if sign	Springe, wenn Vorzeichen (negativ)
JZ	Jump if zero	Springe, wenn 0
LAHF	Load AH with flags	Lade AH mit den Statusbits
LDS	Load pointer using DS	Lade Zeiger unter Verwendung von DS
LEA	Load effective address to register	Lade effektive Adresse in Register
LES	Load pointer using ES	Lade Zeiger unter Verwendung von ES
LOCK	Lock Bus	Verriegle den Bus
LODS	Load string to AL/AX	Lade Zeichenkette in AL/AX
LODSB	Load string bytewise to AL	Lade Zeichenkette byteweise in AL

LODSW	LODSW Load string wordwise to AX	Lade Zeichenkette wonweise in AX
LOOP	Loop CX times	Führe Schleife CX-mal aus
LOOPE	Loop CX times while equal	Führe Schleife CX-mal aus solange gleich
LOOPNE	Loop CX times while not equal	Führe Schleife CX-mal aus solange ungleich
LOOPNZ	Loop CX times while not zero	Führe Schleife CX-mal aus solange nicht 0
LOOPZ	Loop CX times while zero	Führe Schleife CX-mal aus solange 0
MOV	Move	Verschiebe
MOVS	Move string	Verschiebe Zeichenkette
MOVSB	Move string bytewise	Verschiebe Zeichenkette byteweise
MOVSW	Move string wordwise	Verschiebe Zeichenkette wortweise
MUL	Muliply (unsigned)	Multipliziere (vorzeichenlos)
NEG	Negate	Negiere
NOP	No Operation	Keine Operation
NOT	Invert	Invertiere
OR	Or	Oder
OUT	Output to	Gebe aus nach
POP	Pop	Hole
POPF	Pop flags	Hole Statusbits
PUSH	Push	Schiebe
PUSHF	Push flags	Schiebe Statusbits
RCL	Rotate through carry left	Rotiere durch Übertrag-Statusbit nach links
RCR	Rotate through carry right	Rotiere durch Übertrag-Statusbit nach rechts
REP	Repeat CX times	Wiederhole CX-mal
REPE	Repeat CX times while equal	Wiederhole CX-mal solange gleich
REPNE	Repeat CX times while not equal	Wiederhole CX-mal solange nicht gleich
REPNZ	Repeat CX times while not zero	Wiederhole CX-mal solange nicht gleich 0
REPZ	Repeat CX times while zero	Wiederhole CX-mal solange 0
RET	Return from procedure	Kehre aus Prozedur zurück
ROL	Rotate left	Rotiere nach links
ROR	Rotate right	Rotiere nach rechts
SAHF	Store AH into flags	Speichere AH in die Statusbits
SAL	Shift arithmetic left	Schiebe arithmetisch nach links
SAR	Shift arithmetic right	Schiebe arithmetisch nach rechts
SBB	Subtract with borrow	Subtrahiere mit Ausborgen
SCAS	Scan string	Taste Zeichenkette ab
SCASB	Scan string bytewise	Taste Zeichenkette byteweise ab
SCASW	Scan string wordwise	Taste Zeichenkette wortweise ab
SHL	Shift logical left	Schiebe logisch nach links
SHR	Shift logical right	Schiebe logisch nach rechts
STC	Set carry flag	Setze Übertrag-Statusbit
STD	Set direction flag	Setze Richtung-Statusbit

STI	Set Interrupt flag	Setze Unterbrechung-Statusbit
STOS	Store string from AL/AX	Speichere Zeichenkette aus AL/AX
STOSB	Store string bytewise from AL	Speichere Zeichenkette byteweise aus AL
STOSW	Store string wordwise from AX	Speichere Zeichenkette wortweise aus AX
SUB	Subtract	Subtrahiere
TEST	Test	Teste
WAIT	Wait for test	Warte auf Test-Signal
XCHG	Exchange	Tausche aus
XLAT	Translate byte to AL	Setze Byte nach AL um
XOR	Exclusive or	Exklusiv-Oder