

Referenzkarte OMIKRON.Debugger

Die Tastaturbelegung des Debuggers

|                                |  |
|--------------------------------|--|
| SHIFT ↑                        | : Scrollt den Bildschirm (der Cursor bleibt stehen)  |
| SHIFT + CONTROL ↑              | : Scrollt den Bildschirm und bewegt den Cursor mit   |
| SHIFT ⇐ ⇒                      | : Bewegt den Cursor an den linken/rechten Rand   |
| Backspace                      | : Löscht das Zeichen vor dem Cursor  |
| TAB                            | : Der Cursor springt zur nächsten 8er Position   |
| CONTROL RETURN                 | : Setzt den Cursor in die linke untere Ecke  |
| SHIFT RETURN                   | : Schreibt CR  |
| Clr/Home                       | : Setzt den Cursor in die obere linke Ecke   |
| SHIFT Clr/Home                 | : Löscht den Bildschirm, .setzt.den Cursor nach oben links   |
| ESC                            | : Löscht den Bildschirm ab der Cursorposition  |
| Delete                         | : Löscht das Zeichen unterm Cursor, Rest rückt auf   |
| SHIFT Delete                   | : Löscht die Zeile, in der der Cursor steht, Rest rückt auf  |
| CONTROL Delete                 | : Löscht die Zelle ab der Cursorposition   |
| Insert                         | : Fügt ein Space an der Cursorposition ein   |
| SHIFT Insert                   | : Fügt eine Zeile ab der Cursorposition ein  |
| CONTROL Insert                 | : Schaltet den Insert-Modus an (Autoinsert)  |
| ALTERNATE Insert               | : Schaltet den Insert-Modus aus  |
| Undo                           | : Restauriert die Zeile (bis zum letzten RETURN)   |
| Help                           | : Schreibt eine Zelle zurück, die. mit   |
| SHIFT Help                     | : gespeichert wurde  |
| CONTROL 1-9                    | : Speichert den Bildschirminhalt in einen von zehn Buffer ab.<br>Mit:1- 9 sind die Tasten über der Alpha- Tastatur gemeint.<br>Es wird nur der Teil unter der Registeranzeige gespeichert!   |
| ALTERNATE 1 - 9                | : Holt. den Bildschirminhalt aus einem Buffer zurück   |
| ALTERNATE Zahl                 | : Wenn Sie eine Zahl über den Zehnerblock eingeben, wird das<br>entsprechende ASCII-Zeichen ausgegeben. Drücken Sie z. B.<br>ALTERNATE 138 (dann ALTERNATE loslassen), wird e<br>ausgegeben. |
| CONTROL ⇐ ⇒                    | : Damit können Sie durch den Cache blättern (entspricht den<br>Pfeilen links neben der Registeranzeige) (siehe UNTRACE)  |
| CONTROL Clr/Home               | : Zeigt die aktuellen Register an (entspricht dem “Fuller” links<br>von der Registeranzeige)   |
| Funktionstaste                 | : Wählt eine der oben angezeigten Befehle an (siehe 5.2.)  |
| CTRL + ALT + Delete            | : Warmstart des Debuggers  |
| CTRL + ALT + Rechtes SFT + Del | : Kaltstart des Debuggers<br>(ist das gleiche wie der RESET-Taster. Mit zwei Resets<br>schnell hintereinander wird der Debugger verlassen.)  |
| CONTROL + P                    | : Setzt den PC an die aktuelle Adresse   |
| CONTROL + B + 0-9 A-F          | : Setzt einen. Breakpoint an die aktuelle Adresse  |
| CONTROL + G                    | : Startet das Programm ab der aktuellen Adresse<br>(die aktuelle Adresse steht am Zeilenanfang)  |
| SHIFT-SHIFT (beide)            | : das laufende Programm wird unterbrochen und es<br>wird in den Debugger zurückgekehrt.  |
| CONTROL + Help                 | : Der Debugger wird beendet und springt in den<br>Assembler zurück. Das ist nur möglich, wenn der<br>Debugger vom Assembler gestartet wurde.   |
| CONTROL + SHIFT + Help         | : Der Bildschirminhalt wird im DEGAS-Format<br>abgespeichert.  |
| Caps Lock                      | : Umschaltung Zehnerblock  |

|                      |   |   |     |
|----------------------|---|---|-----|
| normal und mit SHIFT |   |   |     |
| (                    | ) | / | *   |
| 7                    | 8 | 9 | -   |
| 4                    | 5 | 6 | +   |
| 1                    | 2 | 3 | En- |
| 0                    |   | , | ter |

|                          |   |   |     |
|--------------------------|---|---|-----|
| Mit Caps Lock ohne SHIFT |   |   |     |
| A                        | B | C | D   |
| 7                        | 8 | 9 | E   |
| 4                        | 5 | 6 | F   |
| 1                        | 2 | 3 | En- |
| 0                        |   | , | ter |

Referenzkarte OMIKRON.Assembler

Die Tastaturbelegung des Assemblers

|                      |   |
|----------------------|---|
| Backspace            | : löscht das Zeichen links vom Cursor                     |
| Delete               | : löscht das Zeichen unter dem Cursor                     |
| Insert               | : fügt ein Leerzeichen an der Cursorposition ein          |
| SHIFT Insert         | : fügt, eine Leerzeile ein                                |
| CONTROL Delete       | : löscht die aktuelle’ Zeile                              |
| CONTROL Y            | : löscht die aktuelle Zeile                               |
| CONTROL Insert       | : schaltet die Editiermodi um (Überschr./Autoinsert I/II) |
| CLR/Home             | : zum Anfang der Seite                                    |
| SHIFT CLR/Home       | : zum Ende der Seite                                      |
| Tab                  | : zur nächsten Tabulatorposition                          |
| SHIFT Tab            | : zur nächsten Tabulatorposition ohne einzurücken.        |
| SHIFT Return         | : zur nächsten Zeile und fügt eine Leerzeile ein,         |
| CONTROL P            | : zum Ende der Zeile                                      |
| CONTROL Ü            | : zum Anfang der Zeile                                    |
| CONTROL Pfeil hoch   | : blättert eine Seite nach oben                           |
| CONTROL Pfeil runter | : blättert eine Seite nach unten                          |
| CONTROL CLR/Home     | : zum Anfang des Sourcetexts                              |
| CONTROL Z            | : zur Position der letzten Änderung                       |
| CONTROL J            | : zum nächsten Fehler                                     |
| CONTROL S            | : zur nächsten ";x"-Marke                                 |
| CONTROL SHIFT S      | : zur vorhergehenden ";x"-Marke                           |
| ALTERNATE E          | : zum Ende des Sourcetexts                                |
| ALTERNATE B          | : zum Blockanfang   |
| ALTERNATE K          | : zum Blockende   |
| ALTERNATE Z          | : zur Zelle (akt. Z.wird für CTRL Z gemerkt               |
| ALTERNATE SHIFT Z    | : zur Zeile o. erneute Abfrage des Label-namens           |
| ALTERNATE S          | : zur nächsten nicht-tokenisierbaren Z.(A-Load)           |
| ALTERNATE SHIFT S    | : zur letzten nicht-tokenisierbaren Zeile                 |
| ALTERNATE J          | : zum nächsten Fehler                                     |
| ALTERNATE SHIFT J    | : zum vorherigen Fehler                                   |
| ALTERNATE 1-0        | : Zu einer Positionsmarkierung                            |
| ALTERNATE 8          | : zum Anfang des TEXT-Segments                            |
| ALTERNATE *          | : zum Anfang des DATA-Segments                            |
| ALTERNATE =          | : zum Anfang des Block-Storage-Segments (BSS)             |
| CONTROL 1-0          | : setzt eine Positionsmarkierung                          |
| CONTROL SHIFT 1-0    | : löscht eine Positionsmarkierung                         |
| ALTERNATE 1- 0       | : zu einer Positionsmarkierung                            |
| CONTROL 8            | : markiert den Blockanfang (auch mit der Maus)            |
| CONTROL K            | : markiert das Blockende (auch mit der Maus)              |
| ALTERNATE B          | : zum Blockanfang   |
| ALTERNATE K          | : zum Blockende   |
| CONTROL H            | : löscht die Blockmarkierungen (Hide Block)               |
| ALTERNATE Y          | : löscht den Block  |
| CONTROL D            | : dupliziert die Zeile, in der der Cursor steht.          |
| CONTROL F            | : sucht ein Symbol (nur zu Symbolaufrufen)                |
| CONTROL L            | : sucht einen ASCII-Text                                  |
| CONTROL E            | : sucht und ersetzt einen ASCII-Text                      |
| ALTERNATE F          | ; setzt die Symbolsuche fort                              |
| ALTERNATE L          | : setzt die ASCII-Suche fort                              |
| CONTROL W            | : Zahlenbasis wandeln (Dez/Hex/Bin/“ASC”/‘ASC’)           |
| CONTROL +            | : Zeichensatz ausgeben (Auswahl mit Maus)                 |
| ALTERNATE SHIFT Help | : Speichert den Bildschirminhalt im DEGAS-Format          |

Pseudopcodes des Assemblers

|  |    |   |
|--|----|---|
| ^^DATE   | 27 | ergibt das Datum im TOS-Format                  |
| ^^TIME   | 27 | ergibt die Uhrzeit im TOS-Format                |
| ^^RSCOUNT  | 27 | ergibt den akt. Stand des RS-Counters           |
| BASE (DC.(B W)   (A0- A7)<br> TEXT DATA BSS Label  | 33 |   |
| LSECTION) BSS (BSS)  | 30 | Beginn des Block-Storage-Segments               |
| CNOP Offset, Justierung  | 34 | PC wird durch Justierung teilbar                |
| LSECTION) DATA (.DATA)   | 30 | Beginn des DATA-Segments                        |
| DC.(B W L) Wert,Wert (DC.(B W L))  | 25 | definiere konstanten                            |
| DS.(B W L) Wert[,Füllwert]<br>(.DS.(B W L)   BLK.(B W L)  <br>DABSS.(B W L)   DCB.(B W L)) | 28 | reserviert Speicher<br>(Größe in B W L = Wert)  |
| DX[.B] 'String'  | 28 | definiert Stringkonstante mit fester Länge      |
| DXSET Länge[,Füllwert]   | 28 | setzt Länge der Stringkonstante                 |
| ELSE   | 36 | wird übers e:zt, wenn Bedine Gung nicht erfüllt |
| END (.END)   | 31 | Ende des Sourcetextes                           |
| ENDC (ENDIF)   | 36 | Ende einer IF-Struktur (bedingte Assembl.}      |
| ENDR   | 35 | Ende einer REPEAT-Schleife                      |
| EVEN (.EVEN   ALIGN[.W])   | 28 | begradigt. die nächste Adresse                  |
| FAIL [Text]  | 36 | gibt Fehlertext aus (bedingte Assemblieren)     |
| IBYTES 'Filename'<br>[, [Anzahl][,Position]] (INCBIN)                                      | 32 | lädt ein File von Diskette nach                 |
| IF[NE] Bedingung   | 36 | Assemblieren, falls Bedingung erfüllt           |
| IFD Label  | 36 | Assemblieren, falls Label definiert             |
| IFEQ Bedingung   | 36 | Assemblieren, falls Bedingung nicht erfüllt     |
| IFGE Ausdruck  | 36 | Assemblieren, falls Ausdruck >= 0               |
| IFGT Ausdruck  | 36 | Assemblieren, falls Ausdruck >» G               |
| IFLE Ausdruck  | 36 | Assemblieren, falls Ausdruck <=: 0              |
| IFLT Ausdruck  | 36 | Assemblieren, falls Ausdruck << Q               |
| IFND Label   | 36 | Assemblieren, falls Label nicht definiert       |
| ILLEGAL  | 35 | fügt \$4AFC ein, löst illegal instruction aus   |
| OPT (D W S M C N)(+ -)(, ...)  | 31 | Stellt Assembleroptionen ein                    |
| ORG Adresse[,^]  | 35 | Stellt Adresse für absolute Assembl. ein        |
| OUTPUT   | 32 | Stellt Narren ein, unter dem gespeichert wird   |
| PATH   | 32 | Pfadname für IBYTES                             |
| REPT Anzahl  | 35 | Beginn einer REPEAT-Strucklur                   |
| RS.(B W L)   | 29 | reserviert Speicher                             |
| RSRESET  | 29 | Setzt RS-Counter zurück                         |
| RSET Wert  | 29 | Setzt den RS-Counter                            |
| RSEVEN   | 30 | Begradigt den RS-Counter                        |
| RSBSS  | 30 | Addiert RS-Counter zur BSS-Länge, RS=0          |
| Symbol EQU Wert (.EQU = = . =)   | 31 | Ordnet einem Label einen Wert zu                |
| Symbol REG Registerliste   | 31 | Definiert Registerliste für movem               |
| [SECTION] TEXT (.TEXT)   | 30 | Beginn des Textsegments                         |

Die Debugger-befehle

|  |     |  |
|--|-----|--|
| ) ASCII-Daten  | 85  | ASC-Präfix licht vor ASC-Dump            |
| ,Term{,Term}   | 64  | Schreibt Daten in Speicher (byteweise)   |
| .(W L)Term{,Term}  | 64  | Schreibt Daten wort- oder langwortweise  |
| / Hex-Daten  | 64  | Schreibt Hex-Daten (Disassemble)         |
| ? Term{,Term}  | 64  | Berechnet Term und gibt ihn aus          |
| ASCFIND [Von.Bis],String   | 61  | Sucht nach Strings in Disass.-Listings   |
| ASCII [Von][[. ]I#[Zeilen]][[,Bis]   | 67  | Zeigt Speicherbereich an (ASC)           |
| (BREAKPOINTSIB) [(Nummer[=Adresse[,(*]=[Wert]]Anzahl <br>?Bedingung)])(K[Nummer])] | 55  | Setzt/Löscht/Zeigt Breakpoints           |
| BSSCLEAR   | 75  | Löscht das BSS                           |
| CHECKSUMME [Adr [,Prüfsumme[,Anzahl]]]   | 75  | Berechnet eine Prüfsumme                 |
| CLR [Von][,Bis]  | 72  | Löscht einen Speicherbereich             |
| CLRCACHE   | 71  | Löscht Cache (Register/Befehls speicher) |
| CLS  | 172 | Löscht den Bildschirm                    |
| COMPARE Von,Bis,Adresse  | 72  | Vergleicht zwei Speicherbereiche         |
| CONTINUE   | 62  | Setzt Suche fort (ASCFIND, FIND, HUNT)   |
| (MOVE COPY) Von,Bis,Nach   | 77  | Kopiert einen Speicherbereich            |

|  |     |   |
|--|-----|---|
| CR [Anzahl]  | 65  | gibt (Anzahl) Leerzeilen aus                |
| CURSOR Nummer  | 77  | Stellt Aussehen des Cursors ein             |
| DIRECTORY [Pfad][Suchmaske]  | 156 | Inhaltsverzeichnis, stellt Pfadnamen ein    |
| DISASSEMBLE [Von][[,]#[Zeilen]][[,Bis]   | 57  | Disassembliert, mit Hex-Code ohne Labels    |
| DO (CALL [Adresse][,Breakpoint])<br>(DUMP MEMORY)[.(B W L)] [Von][[,]#[Zeilen]][[,Bis] | 67  | Führt ein Unterprogramm aus                 |
|  | 66  | Zeigt einen Speicherbereich an (Hex/ASC)    |
| (ERASE KILL) Filename  | 173 | Löscht ein File                             |
| EXIT QUIT SYSTEM   | 60  | Verläßt gen Debugger                        |
| FATTRIBUT [Name][,Attribut]  | 174 | Setzt fileattribute                         |
| FCLOSE   | 165 | Schließt eine Datei (für FILE}              |
| FILEbefehl   | 165 | Ausgabe des Folgebefehl auf File            |
| FILL Von,Bis,Wert{,Wert}   | 71  | Füllt einen Speicherbereich                 |
| FIND [Von,Bis],Term  | 61  | Durchsucht einen Speicherbereich            |
| FOPEN Filename   | 165 | Öffnet ein File (für FILE)                  |
| FORMAT [(DIS)S][,Laufwerk]   | 174 | Formatiert eine Diskette                    |
| FREE [Laufwerk]  | 173 | Erinnelt freien Speicher auf Laufwerk       |
| GETCACHE   | 71  | Holt Registerwerte aus Cache                |
| GETREGISTER [Adresse]  | 75  | Holt Registerwerte aus Speicher             |
| GO [Adresse][,Breakpoint]  | 60  | Startet ein Programm ab Adresse IPC         |
| HUNT [Von,Bis],Term  | 61  | Durchsucht einen Speicherbereich (.w)       |
| IF [Register Bedingung Term]   | 68  | Stellt Abbruchbedingung ein                 |
| INFO   | 62  | Gibt Start von TEXT-, DATA- und BSS aus     |
| INITREGISTER   | 76  | Setzt alle Register auf null                |
| KEY [([CIA]F Nummer=Text)]   | 72  | Belegt eine Funktionstaste                  |
| (ERASE KILL) Filename  | 173 | Löscht ein File                             |
| ((S L)ET) -(ALL Variable)=Term   | 66  | Setzt ein(alle) Register                    |
| LEXECUTE [Filename][,"Commandline"]  | 55  | Lädt ein Programm zum Debuggen (Revoziert)  |
| LINE   | 65  | gibt 79 mal "-" aus                         |
| LIST [Von][[,]#[Zeilen]][[,Bis]  | 58  | Disassembliert, mit Labels ohne Hex-Code    |
| LOAD Filename[,Adresse]  | 155 | Lädt ein File in den Speicher               |
| (DUMP MEMORY)[.(B W L)] [Von][[,]#[Zeilen]][[,Bis]                                     | 66  | Zeigt einen Speicherbereich an (Hex/ASC)    |
| M[OUSE]OFF   | 173 | Schaltet die Maus aus                       |
| M[OUSE]ON  | 173 | Schaltet die Maus ein                       |
| (MOVE COPY) Von,Bis,Nach   | 71  | Kopiert einen Speicherbereich               |
| MKDIRECTORY Filename   | 173 | Richtet ein Unterverzeichnis ein            |
| NAME Altname,Neuname   | 174 | Benennt ein File um                         |
| OBSERVE [Trap[, (? -1 Liste)]]   | 70  | Fängt bestimmte Traps ab                    |
| ! Opcode   | 64  | Präfix Opcode (Ausgabe von LIST)            |
| PRNbefehl  | 65  | Ausgabe des Folgebefehls auf Drucker        |
| (READTRACK RTRACK) [Track[,Seite[,Adresse[,Laufwerk]]]                                 | 79  | Liest einen kompletten, Track               |
| (RS READSECTOR) [Track[,Sektor[,Seite[,Adresse<br>[,Laufwerk]]]]]                      | 178 | Liest Sektoren (XBIOS 8)                    |
| RESET [A[LL][,Vektoren]]   | 75  | Die Systemvektoren gesetzt (auf Debugger)   |
| RESIDENT   | 82  | Verläßt den Debugger (bleibt resident)      |
| RMDIRECTORY Filename   | 173 | Löscht ein Unterverzeichnis (muß leer seln) |
| RWABS Rwfl.,Buf.,Anzahl,Recno,Laufwerk   | 178 | Schreibt(Liest) Sektoren (BIOS 4)           |
| SAVE [Date name][,Anfadr,Endadr]   | 163 | Speichert einen Speicherbereich auf Disk    |
| SAVE ![H][R]   | 177 | Speichert Einstellungen                     |
| SCROLLOFF/SCROLLON [Fangradius]  | 77  | Schaltet Auto-Scrolling ein(aus)            |
| SHOW (Adr[.(B W L)][,Anzahl]) (, .Num)   | 66  | Zeigt Speicherbereich ständig an.           |
| "String["]   | 66  | String ausgeben (PRN oder FILE)             |
| ((S L)ET) -(ALL Variable)=Term   | 66  | Setzt ein(alle) Register                    |
| SWITCH   | 76  | Schaltet Auflösung um                       |
| SYMBOLTABLE [Von][[,]#[Zeilen]][[,Bis]   | 58  | Gibt Symboltabelle aus                      |
| SYNC50 /SYNC60   | 77  | Stellt Bildfrequenz ein,                    |
| ]Term  | 64  | Schreibt Daten (ohne Neuausgabe)            |
| TRACE [Anzahl]   | 63  | Führt einen(mehrere) Befehle aus (Ft)       |
| TYPE [Date name]   | 178 | Anzeigen einer Datei (ASC)                  |
| UNTRACE [Adresse[,Anzahl]]   | 67  | Führt ein Programm bis zur Abbruchbed, aus  |
| (WS WRITESEKTOR) [Track[,Sektor[,Seite[,Adresse<br>[,Laufwerk]]]]]                     | 179 | Schreibt Sektoren (XBIOS 8)                 |
| Befehl   | 79  | Führt einen Befehl sofort aus               |

Reference Card OMIKRON.Debugger

The Keyboard Layout of the Debugger

|                              |   |
|------------------------------|---|
| SHIFT ↑                      | : Scrolls the screen (the cursor stops)   |
| SHIFT + CONTROL ↑            | : Scroll the screen and moves the cursor with   |
| SHIFT ⇐ ⇒                    | : Moves the cursor on the left / right edge   |
| Backspace                    | : Deletes the sign in front of the cursor   |
| TAB                          | : The cursor jumps to the next 8-time position  |
| CONTROL RETURN               | : Sets the cursor in the lower-left corner  |
| SHIFT RETURN                 | : Writes CR   |
| Clr/Home                     | : Sets the cursor into the upper-left corner  |
| SHIFT Clr/Home               | : Deletes the screen, set the cursor to the upper-left  |
| ESC                          | : Deletes the screen from the cursor position   |
| Delete                       | : Deletes the sign under the cursor, remainder moves on   |
| SHIFT Delete                 | : Deletes the line in which the cursor is resting on  |
| CONTROL Delete               | : Deletes the cell from the cursor position   |
| Insert                       | : Inserts a space at the cursor position  |
| SHIFT Insert                 | : Inserts a line from the cursor position   |
| CONTROL Insert               | : Cycle the INSERT mode (Auto-Insert)   |
| ALTERNATE Insert             | : Cycle the INSERT mode   |
| Undo                         | : Restores the line (until the last return)   |
| Help                         | : Write a cell back, with which   |
| SHIFT Help                   | : was saved   |
| CONTROL 1-9                  | : Stores the screen content in one of ten buffers.<br>With: 1-9, the keys are meant above the alpha keyboard.<br>Only the part is stored under the register indicator!                                    |
| ALTERNATE 1 - 9              | : Retrieve the screen content from a buffer   |
| ALTERNATE Number             | : If you enter a number on the numeric keypad, outputs the<br>the appropriate ASCII character. Press e.g.<br>Alternate 138 (then release alternate), E is output.<br>appropriate ASCII characters output. |
| CONTROL ⇐ ⇒                  | : This allows you to scroll through the cache (corresponds to the<br>arrows to the left of the register indicator) (see Untrace)  |
| CONTROL Clr/Home             | : Displays the current registers (corresponds to the "Fuller" links<br>from the register notice)  |
| Function key                 | : Selects one of the commands displayed above (see 5.2.)  |
| CTRL + ALT + Delete          | : Warm-start the Debugger   |
| CTRL + ALT + Right SFT + Del | : Cold-start the Debugger<br>(is the same as the Reset button. With two Resets<br>quickly in a row will leave the Debugger.)  |
| CONTROL + P                  | : Sets the PC to the current address  |
| CONTROL + B + 0-9 A-F        | : Sets Breakpoint to the current address  |
| CONTROL + G                  | : Start the program from the current address<br>(The current address is at the beginning of the line)   |
| SHIFT-SHIFT (both)           | : The current program is interrupted and it<br>is returned to the Debugger.   |
| CONTROL + Help               | : The Debugger is ended and jumps back into the<br>Assembler. That's only possible if the<br>Debugger was started by the Assembler.   |
| CONTROL + SHIFT + Help       | : The screen content is saved in Degas-format.  |
| Caps Lock                    | : Toggle keypad   |

| Normal and with SHIFT |   |   |     |
|-----------------------|---|---|-----|
| (                     | ) | / | *   |
| 7                     | 8 | 9 | -   |
| 4                     | 5 | 6 | +   |
| 1                     | 2 | 3 | En- |
| 0                     |   | , | ter |

| With Caps-Lock, without SHIFT |   |   |     |
|-------------------------------|---|---|-----|
| A                             | B | C | D   |
| 7                             | 8 | 9 | E   |
| 4                             | 5 | 6 | F   |
| 1                             | 2 | 3 | En- |
| 0                             |   | , | ter |

Reference Card OMIKRON.Assembler

The Keyboard Layout of the Assembler

|                      |  |
|----------------------|--|
| Backspace            | : Deletes the character to the left of the cursor    |
| Delete               | : Deletes the character under the cursor             |
| Insert               | : Inserts a space at the cursor position             |
| SHIFT Insert         | : adds a blank line                                  |
| CONTROL Delete       | : Deletes the current line                           |
| CONTROL Y            | : Deletes the current line                           |
| CONTROL Insert       | : Switches the editing modes (overwrite/insert I/II) |
| CLR/Home             | : Back to the top of the page                        |
| SHIFT CLR/Home       | : To the bottom of the page                          |
| Tab                  | : To the next tab position                           |
| SHIFT Tab            | : To the next tab position without mitigating        |
| SHIFT Return         | : To the next line and inserts a blank line          |
| CONTROL p            | : To the end of the line                             |
| CONTROL Ŭ            | : To the beginning of the line                       |
| CONTROL Arrow        | : Scrolls a page upwards                             |
| CONTROL Arrow down   | : Scroll down a page                                 |
| CONTROL CLR/Home     | : Back to the beginning of the source text           |
| CONTROL Z            | : To the position of the last change                 |
| CONTROL J            | : To the next mistake                                |
| CONTROL S            | : To the nights "; x" brand                          |
| CONTROL SHIFT S      | : To the previous "; X" brand                        |
| ALTERNATE E          | : To the end of the source text                      |
| ALTERNATE B          | : To the block start                                 |
| ALTERNATE K          | : To the block end                                   |
| ALTERNATE Z          | : To the cell (act. Z. is notified for Ctrl Z        |
| ALTERNATE SHIFT Z    | : To line o. Renewed query of the label name         |
| ALTERNATE S          | : To the next non-tokenizable Z. (A-Load)            |
| ALTERNATE SHIFT S    | : To the last non-tokenizable line                   |
| ALTERNATE J          | : To the next mistake                                |
| ALTERNATE SHIFT J    | : To the previous error                              |
| ALTERNATE 1-0        | : To a position mark                                 |
| ALTERNATE 8          | : To the beginning of the TEXT segment               |
| ALTERNATE *          | : Back to the top of the DATA Segment                |
| ALTERNATE =          | : Back to the top of the Block Storage Segment (BSS) |
| CONTROL 1-0          | : Sets a position marker                             |
| CONTROL SHIFT 1-0    | : Deletes a position mark                            |
| ALTERNATE 1-0        | : To a position marker                               |
| CONTROL 8            | : Marks the block start (also with the mouse)        |
| CONTROL K            | : Marks the blaze end (also with the mouse)          |
| ALTERNATE B          | : To the block start                                 |
| ALTERNATE K          | : To the block end                                   |
| CONTROL H            | : Deletes the block marks (hide block)               |
| ALTERNATE Y          | : View the block                                     |
| CONTROL D            | : Duplicate the line in which the cursor stands      |
| CONTROL F            | : Look for a symbol (only to call calls)             |
| CONTROL L            | : Look for an ASCII text                             |
| CONTROL E            | : Looks and replaces an ASCII text                   |
| ALTERNATE F          | ; Sets the symbol search                             |
| ALTERNATE L          | : Sets the ASCII search                              |
| CONTROL W            | : Convert numbers (DEC / HEX / BIN / "ASC" / 'ASC')  |
| CONTROL +            | : Output character set (selection with mouse)        |
| ALTERNATE SHIFT Help | : Saves the screen content in Degas-format           |

**Pseudo-opcodes of the Assembler**

|  |    |   |
|--|----|---|
| ^^DATE   | 27 | Gives the date in TOS-Format                    |
| ^^TIME   | 27 | Gives the time in the TOS-Format                |
| ^^RSCOUNT  | 27 | Gives the act. State of the RS-Counters         |
| BASE (DC.(B W)   (A0- A7)<br> TEXT DATA BSS Label  | 33 | Adjusts an addressing offset                    |
| LSECTION) BSS (BSS)  | 30 | Start of the Block Storage Segment              |
| CNOP Offset, adjustment  | 34 | PC is divisible by adjustment                   |
| LSECTION) DATA (.DATA)   | 30 | Beginning of the Data Segment                   |
| DC.(B W L) Wert,Wert (DC.(B W L))  | 25 | Define constants                                |
| DS.(B W L) value[,Fillvalue]<br>(.DS.(B W L)   BLK.(B W L)  <br>DABSS.(B W L)   DCB.(B W L)) | 28 | Define storage<br>(Size in B W L = Value)       |
| DX[.B] 'String'  | 28 | Defined string constant with fixed length       |
| DXSET Long[,Fill value]  | 28 | Set of the string constant                      |
| ELSE   | 36 | is over E: ZT if condition is not met           |
| END (.END)   | 31 | End of Source Text                              |
| ENDC (ENDIF)   | 36 | End of an IF structure (conditional assemblies) |
| ENDR   | 35 | End of a repeat loop                            |
| EVEN (.EVEN   ALIGN[.W])   | 28 | Straightened. the nearest address               |
| FAIL [Text]  | 36 | returns error text (conditional assemblies)     |
| IBYTES 'Filename'<br>[, [Number][,Position]] (INCBIN)  | 32 | Loads a file from floppy disk                   |
| IF[NE] condition   | 36 | Assemble if condition is met                    |
| IFD Label  | 36 | Assemble if label defined                       |
| IFEQ Condition   | 36 | Assemble if condition is not met                |
| IFGE Expression  | 36 | Assemble if expression >= 0                     |
| IFGT Expression  | 36 | Assemble if expression > 0                      |
| IFLE Expression  | 36 | Assemble if expression <= 0                     |
| IFLT Expression  | 36 | Assemble if expression < 0                      |
| IFND Label   | 36 | Assemble if label is not defined                |
| ILLEGAL  | 35 | adds \$4AFC, resolves illegal instruction       |
| OPT (D W S M C N)(+ -)(, ...)  | 31 | Sets assembler options                          |
| ORG Address[,^]  | 35 | Represents address for Absolute Assemblies      |
| OUTPUT   | 32 | Sets fools under which is stored                |
| PATH   | 32 | Path name for Ibytes                            |
| REPT Number  | 35 | Start of a repeat tower                         |
| RS.(B W L)   | 29 | Reserved memory                                 |
| RSRESET  | 29 | Return RS-Counter                               |
| RSSET value  | 29 | Sets the RS-Counter                             |
| RSEVEN   | 30 | Straightens the RS-Counter                      |
| RSBSS  | 30 | Add RS-Counter to the BSS length, RS = 0        |
| Symbol EQU Value (.EQU = = .=)   | 31 | Assigns a value to a label                      |
| Symbol REG Register  | 31 | Defines Movem tab                               |
| [SECTION] TEXT (.TEXT)   | 30 | Beginning of the TEXT segment                   |

**The Debugger Commands**

|  |     |   |
|--|-----|---|
| ) ASCII-Data   | 85  | ASC prefix light in front of ASC-dump     |
| ,Term{,Term}   | 64  | Writes data in memory (byte)              |
| .(W L)Term{,Term}  | 64  | Write data word or longword               |
| / Hex-Data   | 64  | Writes Hex Data (Disassemble)             |
| ? Term{,Term}  | 64  | Calculate term and gives it out           |
| ASCFIND [From.To],String   | 61  | Look for strings in disass. Listings      |
| ASCII [From][[.]I#[Lines]][[,To]   | 67  | Displays memory area (ASC)                |
| (BREAKPOINTS B) [(Number[=Address[,(* = Value] Number <br>?Condition)]])(K[Number])] | 55  | Sets / deletes / shows breakpoints        |
| BSSCLEAR   | 75  | Deletes the BSS                           |
| CHECKSUMME [Adr[,Check[,Number]]]  | 75  | Calculates a checksum                     |
| CLR [From][,To]  | 72  | Deletes a memory area                     |
| CLRCACHE   | 71  | Deletes cache (register / command memory) |
| CLS  | 172 | Deletes the screen                        |
| COMPARE From,To,Address  | 72  | Compares two storage areas                |
| CONTINUE   | 62  | Sets search (AscFind, Find, Hunt)         |
| (MOVE COPY) From,To,After  | 77  | Copies a memory area                      |

|  |     |  |
|--|-----|--|
| CR [Number]  | 65  | gives (number) blank lines                 |
| CURSOR Number  | 77  | Sets the look of the cursor                |
| DIRECTORY [Path][Mask]   | 156 | Table of contents, sets pathnames          |
| DISASSEMBLE [From][[,]#[Lines]][[,To]  | 57  | Disassembled, with hex code without labels |
| DO (CALL [Mail][,Breakpoint])<br>(DUMP MEMORY)[.(B W L)] [From][[,]#[Lines]][[,To] | 67  | Executes a sub-program                     |
|  | 66  | Displays a memory area (HEX / ASC)         |
| (ERASE KILL) Filename  | 173 | Deletes a file                             |
| EXIT QUIT SYSTEM   | 60  | Sticks gen debugger                        |
| FATTRIBUT [Name][,Attribute]   | 174 | Sets file attributes                       |
| FCLOSE   | 165 | Closes a file (for File)                   |
| FILEbefehl   | 165 | Output of the follow-up command to file    |
| FILL From,To,Value{,Value}   | 71  | Fills a storage area                       |
| FIND [From,To],Term  | 61  | Searches a memory area                     |
| FOPEN Filename   | 165 | Opens a file (for file)                    |
| FORMAT [(DIS)S][,Drive]  | 174 | Format a floppy disk                       |
| FREE [Drive]   | 173 | Erinitelt free memory on drive             |
| GETCACHE   | 71  | Bring tab values from cache                |
| GETREGISTER [Address]  | 75  | Brings registry values from memory         |
| GO [Address][,Breakpoint]  | 60  | Starts a Program from address IPC          |
| HUNT [From,To],Term  | 61  | Searches a memory area (.w)                |
| IF [Register Condition Term]   | 68  | Provides demolition condition              |
| INFO   | 62  | Returns Start of TEXT, DATA and BSS        |
| INITREGISTER   | 76  | Sets all registers to zero                 |
| KEY [([CIA]F Number=Text)]   | 72  | Occupies a function key                    |
| (ERASE KILL) Filename  | 173 | Deletes a file                             |
| ((S L)ET) -)(ALL Variable)=Term  | 66  | Sets a (all) registers                     |
| LEXECUTE [Filename][, "Commandline"]   | 55  | Loads a program for Debugger (relocate)    |
| LINE   | 65  | Is 79 Times "-"                            |
| LIST [From][[,]#[Lines]][[,To]   | 58  | Disassembled, with labels without hex code |
| LOAD Filename[,Address]  | 155 | Loads a file into the store                |
| (DUMP MEMORY)[.(B W L)] [From][[,]#[Lines]][[,To]                                  | 66  | Displays a memory area (HEX / ASC)         |
| M[OUSE]OFF   | 173 | Turn off the mouse                         |
| M[OUSE]ON  | 173 | Turns the mouse                            |
| (MOVE COPY) From,To,After  | 71  | Copies a memory area                       |
| MKDIRECTORY Filename   | 173 | A sub-directory sets up                    |
| NAME Altname,Newname   | 174 | Names a file                               |
| OBSERVE [Trap[, (? -1 Listing)]]   | 70  | Calls certain traps                        |
| ! Opcode   | 64  | Prefix opcode (issue of List)              |
| PRNcommand   | 65  | Output of the follow-up command to printer |
| (READTRACK RTRACK) [Track[,Side[,Address[,Drive]]]                                 | 79  | Reads a complete track                     |
| (RS READSECTOR) [Track[,Sector[,Side[,Address<br>[,Drive]]]]]                      | 178 | Reads sectors (XBIOS 8)                    |
| RESET [A[LL][,Vectors]]  | 75  | The system vectors set (on debugger)       |
| RESIDENT   | 82  | Leaves the debugger (remains resident)     |
| RMDIRECTORY Filename   | 173 | Delete a sub-directory (must be empty)     |
| RWABS Rwfl.,Buf.,Number,Recno,Drive  | 178 | Write (reads) sectors (BIOS 4)             |
| SAVE [Date name][,Startadr,Endadr]   | 163 | Saves a storage area to Disk               |
| SAVE ![H][R]   | 177 | Stores settings                            |
| SCROLLOFF/SCROLLON [Fangradius]  | 77  | Turns auto scrolling (OFF)                 |
| SHOW (Adr[.(B W L)][,Number]) (, .Num)   | 66  | Displays memory area constantly            |
| "String["]   | 66  | Output String (PRN or File)                |
| ((S L)ET) -)(ALL Variable)=Term  | 66  | Sets a (all) registers                     |
| SWITCH   | 76  | Turns Resolution                           |
| SYMBOLTABLE [From][[,]#[Lines]][[,To]  | 58  | Specifies symbol table                     |
| SYNC50 /SYNC60   | 77  | Sets frame frequency                       |
| ]Term  | 64  | Write data (without new edition)           |
| TRACE [Number]   | 63  | Performs a (multiple) commands (FT)        |
| TYPE [Date name]   | 178 | Viewing a File (ASC)                       |
| UNTRACE [Address[,Number]]   | 67  | Leads a program to the demolition          |
| (WS WRITESEKTOR) [Track[,Sector[,Side[,Address<br>[,Drive]]]]]                     | 179 | Writes Sectors (XBIOS 8)                   |
| command  | 79  | Executes a command immediately             |