TRACE80

SYMBOLISCHER DEBUGGER FÜR Z80-MIKROCOMPUTER

VERSION 2.0

VERTRIEB: LOTHAR LAUTERBACH
ELEKTRONIK-ENTWICKLUNG
DEFREGGERSTR. 1
D-8012 OTTOBRUNN
TEL. 089/606173

INHALT

EINLEITUNG

TRACE80 ist ein symbolischer Debugger für Mikrocomputer mit Z80-CPU und dem Betriebssystem CP/M. Das Programm benötigt eine Symboltabelle, die von den meisten für CP/M verfügbaren Linkern erzeugt wird. Somit können nicht nur Assembler-, sondern auch von Compilern erzeugte Objektprogramme getestet werden.

ENTWURFSGESICHTSPUNKTE

CP/M-Systeme verfügen heute in aller Regel über einen RAM-Speicher von 64 KByte. Anwenderprogramme, die in Assembler entwickelt werden, überschreiten selten die Grenze von 20 KBytes. Der Datenbereich läßt sich in der Entwicklungsphase meist einschränken. Daher wurde mehr Wert auf einen leistungsfähigen Befehlssatz als auf Minimierung der Programmlänge gelegt.

SYSTEMANFORDERUNGEN

TRACE80 Version 2.0 benötigt 18 KByte und zusätzlich für jedes verwendete Symbol 8 bzw. 10 Byte. Das Betriebssystem belegt in der Regel 5 KByte, so daß bei einem 32-KByte-System noch maximal 9 KByte zur Verfügung stehen.

Das CRT-Terminal sollte mindestens 4800 Baud verarbeiten.

SYSTEMANPASSUNG

Die Anpassung des Programms an verschiedene Computer-Systeme erfolgt über eine Tabelle (siehe Anhang A). Um Fehlfunktionen zu vermeiden, müssen zumindest die Funktionen 'Löschen Bildschirm' und 'Löschen Zeile' adaptiert, sowie die Parameter für Bildschirmformat und Druckerbreite eingestellt werden.

LADEN TRACE80

Programme werden in CP/M-Systemen durch Angabe ihres Namens aufgerufen. In der gleichen Zeile kann durch Angabe eines weiteren Programmnamens ein File in den Arbeitsbereich des Tracers geladen werden. Ohne Angabe einer Startadresse wird ab 100H geladen, vorhandene Adressparameter geben Ladeadresse, Limit und Offset an.

Existiert ein Symbolfile gleichen Namens, prüft der Relokator die Länge der Datei und bemißt die Länge des Symbolpuffers entsprechend. Wird kein Symbolfile gefunden, wird ein Puffer mit fester Länge erzeugt.

Das Steuerzeichen '/' übergibt alle folgenden Zeichen an den Befehlsdecoder des TRACE80.

Beispiele

A>T80 TEST	File TEST.COM wird ab 100H geladen.
A>T80 TEST.COD	File TEST.COD wird ab 100H geladen
A>T80 TEST. 100 200	File TEST wird von 100H bis maximal einschließlich 200H geladen.
A>T80	Die Länge des Symbolpuffers beträgt 512 Symbole
A>T80 :200 TEST	Die Länge des Symbolpuffers beträgt 200 Symbole, File TEST.COM wird ab 100H geladen.
A>T80 /BAT X 2	Der Befehl 'BAT X 2' wird nach Laden des TRACE80 sofort ausgeführt.

Beim Programmaufruf lokalisiert sich TRACE80 unterhalb des BDOS (Disk Operating System des CP/M). Der Symbol-Puffer befindet sich zwischen TRACE80 und BDOS.

TRACE80 stellt für Anwenderprogramme ein verkleinertes System dar. Der BDOS-Einsprungvektor wird auf den Anfang von TRACE80 umgeschaltet. TRACE80 greift zur Kommunikation mit Terminal und Drucker direkt auf das BIOS zu. Die Funktion 'WARMBOOT' initialisiert das BDOS. Anwenderprogramme, die unter TRACE80 laufen, können mit BDOS-Funktion 0 oder Sprung auf Adresse 0 beendet werden. Dabei wird TRACE80 aufgerufen. Der Rücksprung ins Betriebssystem erfolgt durch den Befehl 'BYE' des TRACE80.

Anwenderprogramme unter CP/M laufen ab 100H und benötigen den Bereich ab 0 bis 0FFH zum Datenaustausch mit dem Betriebssystem. Maschinenprogramme, die keine Unterstützung durch das Betriebssystem benötigen, können ab Adresse 0 geladen werden. TRACE80 belegt einen RESTART-Adresse bei Ausführung von Programmen in Echtzeit und die Adressen von 66H bis 68H bei Verwendung des Befehls 'NS' (NMI).

TRACE80 kann nur geladen werden, solange der Befehl XSUB des Betriebssystems nicht aktiv ist. XSUB kann deaktiviert werden, indem ein Warmstart ausgeführt wird. Dies kann durch einen Sprung auf die Adresse BIOS+3* erreicht werden. Ein Sprung auf Adresse 0 belässt das Programm XSUB im Speicher.

*WARMBOOT

BEFEHLSEINGABE

Hinweis

Im Folgenden steht

<CR>	(Car.Return)	für Wagenrücklauf
<LF>	(Line Feed)	für Zeilenvorschub
<SP>	(Space)	für Leerzeichen
<BS>	(Back Step)	für Rückwärtsschritt
<HT>	(Hor. Tab.)	Für Tabulator
<VT>	(Vert. Tab.)	für Vert. Tabulator ↑K
<ESC>	(Escape)	für Escape
<FF>	(Form Feed>	für Seitenvorschub ↑L
<ETX>	(End of Text)	für ^C
<DEL>	(Delete)	für Löschtaste

TRACE80 meldet sich nach dem Laden mit '*' und zeigt damit die Bereitschaft zur Befehlseingabe.

Befehle bestehen aus einem Befehlswort und bis zu 20 Parametern. Die Eingabe erfolgt gepuffert und wird mit <CR> oder <LF> abgeschlossen.

Folgende Steuerzeichen dienen zum Editieren der Eingabezeile:

<ESC>	Löschen der Zeile und Setzen des Cursors zum Zeilenanfang.
<DEL>	Löschen eines Zeichens.
<VT>	Einfügen eines Zeichens. ↑K
<HT>	Cursor nach rechts.
<BS>	Cursor nach links.

Befindet sich der Cursor in vorderster Position und ist der Eingabepuffer leer, akzeptiert TRACE80 Kommandos, die sofort ausgeführt werden:

,	Einzelschritt
.	Einzelschritt mit Protokoll
;	Befehl überspringen
:	Einzelschritt mit Kurzanzeige
/	Wiederholen des letzten Befehls
_	Echtzeitunterprogramm
<LF>	Register-Darstellung
<HT>	Übernahme des letzen Eingabe
<ETX>	Initialisierung des BDOS
<FF>	Löschen Bildschirm
<ESC>	Löschen BATCH-Mode

Die Steuerzeichen für die Direkt-Kommandos können zur Anpassung an das Terminal geändert werden. (Anpassung siehe Anhang A). Die Einzelschrittausführung und der Unterprogrammaufruf sollen ohne Betätigen der Shift-Taste ausführbar sein. Bei Terminals mit getrennter Zifferntastatur können diese Tasten mit den Direkt-Kommandos belegt werden.

UNTERBRECHUNGEN

In Ausführung befindliche Kommandos können durch Drücken der Leertaste unterbrochen oder durch <CR>, <LF>, <ESC>, <FF> oder <ETX> abgebrochen werden. Die Abbruchzeichen werden dabei wie Direkt-Kommandos verarbeitet.

Bei Unterbrechung erscheint die Programmadresse und '-' als Prompt. Mit <SP> kann der Befehl fortgesetzt werden. Alle anderen Zeichen werden sofort in den Eingabepuffer übernommen, bzw. als Direkt-Kommandos sofort ausgeführt.

EINGABEFORMAT

Befehle bestehen aus Kommandowort und Parametern. Befehle und Parameter sind in Großbuchstaben einzugeben. Sollte das Terminal keine Alpha-Lock-Taste besitzen, kann durch Setzen eines Flags die Wirkung der Shift-Taste umgekehrt werden. (Änderung siehe Anhang A).

Zwischen Befehlswort und Parametern muß mindestens ein Leerzeichen stehen. Parameter müssen kompakt, d. h. ohne Leerzeichen eingeben werden. Die Trennung zwischen den Parametern erfolgt entweder mit Leerzeichen oder mit Komma. Bei strukturierten Parametern oder zur Eingabe eines leeren Parameters ist das Komma unbedingt zu verwenden.

Beispiele

3+3	Ein Parameter mit Wert 6
3 +3	Zwei Parameter mit Wert 3 und 3
3,,3	Drei Paramter mit Wert 3,leer, und 3

FEHLERMELDUNGEN

TRACE80 kennt folgende Fehlermeldungen:

COMMAND ERROR	Befehlswort ungültig oder Befehl nicht zulässig.
ARGUMENT ERROR	Fehler in den Argumenten, zu große Klammerungstiefe, falsche Trennzeichen.
INVALID REGISTER NAME	Unzulässiger Name für CPU-Register
INVALID SYMBOL	Symbol nicht vorhanden oder kein Symbol-File geladen.
MEMORY ERROR	Fehlerhafte Speicherzelle oder Versuch, in ROM zu schreiben.
CANNOT ASSEMBLE	Assemblierungs-Fehler
CANNOT TRACE	Unzulässiger OP-Code
STACK UNDERFLOW	Wert des Stapelzeigers ist größer als Limit
FILE NOT FOUND	Datei nicht vorhanden.
FILE TOO LONG	Datei überschreibt TRACE80
DISK FULL	Diskette voll
DIRECTORY FULL	Directory voll
SYMBOL BUFFER OVERFLOW	Symbolpuffer zu klein
INVALID SYMBOL FILE	Symbol-File fehlerhaft
ADRESS TABLE OVERFLOW	Adresstabelle z.B. für Haltepunkte voll

ARITHMETISCHE AUSDRÜCKE

Arithmetische Ausdrücke bestehen aus Konstanten, Variablen und Operationssymbolen. Variable und Konstanten werden intern durch 16-Bit Zahlen dargestellt. Überläufe bei Operationen werden ignoriert. Negative Zahlen werden durch ihr Zweierkomplement dargestellt. Ausdrücke sind ohne Zwischenraum einzugeben, da das Leerzeichen zur Trennung der Ausdrücke verwendet wird.

KONSTANTEN

HEX-KONSTANTEN

Konstanten werden, soweit sie nicht durch Vorschalten eines Sonderzeichens kenntlich gemacht sind, grundsätzlich als Hex-Zahlen interpretiert. Daher dürfen in TRACE80 Zahlen mit den Buchstaben 0-9 und A-F beginnen.

Beispiele	1000	1000H
	-1000	0F000H

DEZIMALZAHLEN

Dezimalzahlen werden durch das Prozentzeichen gekennzeichnet. Zulässig sind Werte von 0..65535.

Beispiele	%1024	0400H
	-%1024	FC00H

BINÄRKONSTANTEN

Binärkonstanten werden mit dem Ausrufezeichen markiert. Der Wertebereich erstreckt sich von 0..1111_1111_1111_1111. Zur Strukturierung kann der Unterstrich benutzt werden.

Beispiel	!100_0000_0000	0400H

SYMBOLE

Symbolnamen beginnen mit '.' . Das erste Zeichen muß ein Buchstabe oder ein Sonderzeichen sein. Zulässig sind die Symbole 0-9, A-Z,$,_,.,%,?,@.

Beispiel .STACK

ASCII-KONSTANTEN

ASCII-Zeichen, mit Ausnahme der Steuerzeichen und des Leerzeichens, können durch das einfache Stringzeichen gekennzeichnet werden.

Beispiel 'A 41H

ARITHMETISCHE VARIABLEN

Arithmetische Variablen werden benutzt, um auf häufig verwendete Adressen schnell zugreifen zu können.

PROGRAMMZÄHLER

Die Variable '$' beinhaltet den augenblicklichen Stand des Programmzählers. Die Variable wird nur bei der Emulation aktualisiert. Bei Echtzeitausführung enthält sie die Adresse des letzten emulierten Befehls, bzw. bei Echtzeitausführung mit Haltepunkten die Adresse des zuletzt durchlaufenen Haltepunktes.

VARIABLE X

X enthält den (aktuellen!) Wert desjenigen arithmetischen Ausdrucks, der beim Befehl HF gespeichert wird.

VARIABLE Y

Y enthält den Wert des letzten Parameters. Die Variable ist dann hilfreich, wenn bei Befehlen, die eine Bereichsangabe erfordern, statt der Grenzen Anfangsadresse und Distanz eingegeben werden sollen

Beispiel D .RDHL-10 Y+%20 Es werden 20 Bytes ab Adresse RDHL-10 dargestellt.

VARIABLE Z

Die Variable Z wird in den Befehlen F und CP aktualisiert. Sie zeigt somit immer auf die zuletzt gefundene Adresse vor Abbruch der Befehle F oder CP.

VARIABLE M (MORE)

M wird durch die Befehle D,L,LP und LD aktualisiert.

Beispiel	D 100	Dump ab 100H
	<CR>	Abbruch des Befehls
	D M	Fortsetzung des Speicher-Dumps

CPU-REGISTER

Die Inhalte der Register der CPU können durch Voranstellung von 'R' vor den Registernamen verwendet werden.

Beispiele	RHL	Inhalt von HL
	RC'	Inhalt von C'
	RCA	Inhalt des Carry-Flags

Gültige Registernamen siehe Anhang B.

ARITHMETISCHE OPERATIONEN

NEGATIVE ZAHLEN

Das Minuszeichen bildet immer das Zweier-komplement der zugehörigen Hex-Zahl

Beispiel	-%1024	FC00H

EINER-KOMPLEMENT

Das Einer-Komplement ist die bitweise Invertierung der Hex-Zahl und errechnet sich aus dem Zweierkomplement durch Subtraktion von 1 . TRACE80 wandelt Zahlen ins Einer-Komplement durch Voranstellen von 'U' (Unary Minus).

Beispiele	U0000	FFFFH
	-U0000	1H
	U-0000	FFFFH

ZWEIER-POTENZ

'^' bildet die Zweierpotenz einer Zahl zwischen 0 und 15.

Beispiele	^0	0H
	^F	8000H

ADRESSBEZÜGE

Die Funktion @ ergibt als Resultat den Inhalt der Speicherzelle, die durch den Operanden adressiert wird.

Beispiele	@1000	Inhalt der Zellen 1000H und 1001H
	@RHL	Inhalt der Speicherzellen, die durch HL adressiert sind.

SPEICHERBYTES

Die Funktion '[' hat als Ergebnis den Wert des Speicherbytes, das durch das Argument adressiert wird.

Beispiele	[1000	Inhalt der Zelle 1000H
	[(RIX+3)	Inhalt der Zelle (RIX+3)
	[RIX+3	3 + Inhalt von (IX)

FUNKTION H (HIGH)

Wert des High-Bytes des Operanden

Beispiel	H1000	0010H

FUNKTION L (LOW)

Wert des Low-Bytes des Operanden

Beispiel	L1003	0003H

FUNKTION I (IF)

Syntax: I(<Bedingung>,<Wert1>[,<Wert2>])

I hat als Ergebnis Wert1, wenn die Bedingung wahr ist, ansonst Wert2 oder 0.

Beispiel I($>1000,2000,100)

Ergebnis ist 2000H, falls Programmzähler größer 1000H, sonst 100H.

FUNKTION P (PORT)

P liest einen 8-Bit-Wert vom I/O-Port.

Beispiel P10 Inhalt von Port 10H

MULTIPLIKATION

Die Multiplikation erfolgt zwischen 16-Bit-Werten, ohne Überläufe zu berücksichtigen.

Beispiele 1000*3 3000H

1000*10 0H

DIVISION

Die Division erfolgt ganzzahlig ohne Rest.
Bei Division durch 0 erscheint die Meldung

/0 ARGUMENT ERROR

Beispiel %10/3 3H

MODULO

Die Operation MODULO wird durch '\' dargestellt.

Beispiel 8\3 2H

BITWEISE VERKNÜPFUNG

Um eine schnelle Interpretation und übersichtliches Aussehen in der Kompakt-Schreibweise zu gewährleisten, werden Bit-Operationen durch das Nummernzeichen eingeschlossen.

Im Gegensatz zu booleschen Operationen, die als Ergebnis die Aussage 'Wahr' oder 'Falsch' liefern, stellen die Operationen '#' eine bitweise logische Verknüpfung dar. Das Ergebnis ist eine Zahl.

Funktionen:

N#	NOT	Bitweise Invertierung
#A#	AND	Bitweise UND-Verknüpfung
#O#	OR	Bitweise ODER-Verknüpfung
#X#	XOR	Bitweise Antivalenz-Verknüpfung

Beispiele	1000#O#2000	3000H
	RHL#A#1	Maskierung von Bit O in Register HL

HIERARCHIE

Arithmetische Operationen werden entsprechend der nachfolgenden Reihenfolge ausgeführt:

1. (.....)
2. U,+,-,@,[,I,H,L,P (+/- als Vorzeichen)
3. *,/,\
4. +,- (+,- als Operation)
5. N#
6. #A#
7. #O#,#X#

Innerhalb gleicher Ebenen wird der Ausdruck von links nach rechts abgearbeitet.

BYTE-AUSDRÜCKE

Ausdrücke, deren Ergebnis einer Byte-Variablen zugewiesen wird, dürfen als Resultat nur Werte zwischen -256 und +255 besitzen.

BIT-AUSDRÜCKE

Eingaben zum Löschen oder Setzen von FLAGS dürfen nur die Werte 0 oder 1 besitzen.

BOOLESCHE AUSDRÜCKE (BEDINGUNGEN)

Boolesche Ausdrücke dienen als Steuerkriterien bei Trace-Befehlen. Ihr Wert ist entweder 'Wahr' oder 'Falsch'.

BOOLESCHE KONSTANTE T (TRUE)

Die Konstante T besitzt den Wert 'True'.

BOOLESCHE VARIABLE Q (QUERY)

Q besitzt als Wert das aktuelle Ergebnis des im Befehl BF eingegebenen Booleschen Ausdrucks

BOOLESCHE VARIABLE S (SYMBOL)

S ist 'Wahr', wenn der Programmzähler eine symbolische Adresse enthält.

VERGLEICHE

Vergleiche sind Operationen zwischen zwei arithmetischen Ausdrücken. Als Operationszeichen sind zugelassen:

=	Gleich
>	Größer
<	Kleiner
>=	Größer gleich
=>	Größer gleich
<=	Kleiner gleich
=<	Kleiner gleich
<>	Ungleich
><	Ungleich

Beispiel $>1000 Wahr wenn Programmzähler größer 1000H

BOOLESCHE OPERATIONEN

Boolesche Operationen sind von Bit-Operationen zu unterscheiden:

N:	NOT
:A:	AND
:O:	OR
:X:	XOR

Beispiele	$>100:A:RHL=0	'Wahr' wenn Programmzähler größer 100H und HL gleich 0
	N:T	'Falsch'

PRIORITÄTEN

Boolesche Ausdrücke werden entsprechend der folgenden Reihenfolge abgearbeitet:

1. (.....)
2. T,Q,S,Vergleiche
3. N:
4. :A:
5. :O:, :X:

ANZEIGEFORMATE

BEFEHLSZEILEN

Befehlszeilen werden wie folgt dargestellt:

```
G/B/C/J/ PC 1000  CD 00 20    MARKE   CALL 2000  ;OUT
  (1)    (2)  3      (4)        5)       (6)       7)
```

(1) Ablaufsteuermarken

/B für Break
/C für Call
/G für Go
/J für Jump

(2) Markierung für Programmzähler

3) Adresse

4) OP-Code (1 - 4 Bytes)

(5) Symbolische Marke

6) Mnemonic

(7) Symbolische Marke der Operandenadressee oder ASCII-Zeichen

CPU-REGISTER

CPU-Register'erscheinen wie folgt:

```
84218421  A   B  C   D  E   H  L     SP   IX   IY   I  IFF
........  00  00 00  00 00  00 00    9000 0000 0000 00 0

84218421  A'  B' C'  D' E'  H' L'    -4   -2
........  00  00 00  00 00  00 00    1000 2000
```

Flags werden, wenn nicht gesetzt, durch '.', ansonsten durch

C	CARRY
N	Hilfsflag für Addition
P	PARITY
H	HALF-CARRY
Z	ZERO
S	SIGN

angezeigt. Unten rechts wird ein Teil des Stacks abgebildet. I ist das Interruptregister, IFF das Interrupt-Flip-Flop der CPU.

DATEN

Datenbereiche werden immer in Hex- und ASCII am Bildschirm dargestellt. Hinter jedem Hex-Byte mit Adressen XXXFH wird statt des Leerzeichens ein Punkt gedruckt.

DRUCKER

Am Drucker kann in zwei Betriebsarten protokolliert werden:

VOLLAUSDRUCK

Einschalten/Ausschalten durch ^P

Sämtliche Ausgaben am Terminal werden mitgedruckt.

TEILAUSDRUCK

Einschalten/Ausschalten durch ^T

Der Systemdrucker druckt nur Kurzzeilen, wenn am Terminal die Registerdarstellung erfolgt. Außerdem werden TRACE80-Befehle und Fehlermeldungen erfaßt.

BEFEHLE

Alle TRACE80 Befehle sind in folgendem Format dargestellt:

FORMAT:	Vollständige Darstellung der Syntax
VERSION:	Liste der Versionen, in denen der Befehl verfügbar ist
FUNKTION:	Kurze Beschreibung der Funktion
BEMERKUNGEN:	Beschreibung der Besonderheiten
BEISPIELE:	Typische Beispiele

NOTATION

Die Syntax wird gemäß folgenden Regeln dargestellt:

1. Namen in Großbuchstaben müssen wie dargestellt eingegeben werden.
2. Namen in Kleinbuchstaben in Winkel-Klammern stehen für einen Ausdruck.
3. Ausdrücke in rechteckigen Klammern können weggelassen werden.
4. Alle Interpunktionen müssen wie dargestellt eingegeben werden.
5. Endet die Syntaxdarstellung mit '...' so kann der vorhergehende Ausdruck wiederholt werden.
6. Zeilen, die unterstrichen sind, werden vom TRACE80 ausgegeben.

D (DUMP)

Format: D [<Startadresse> [<Endadresse>]]

Version: Disk, ROM

Funktion: Darstellung des Speicherinhalts in HEX und ASCII

Bemerkungen: Fehlt die Adressangabe, werden die Grenzen des letzten D-Befehls benutzt.

Die Variable M wird aktualisiert.

Beispiele:

D 1000	Speicher-Dump ab 1000H
D 1000 1100	Speicher-Dump bis 1100H
D	Speicher-Dump mit alten Parametern
D M	Fortsetzung (Dump More)

L (LIST)

Format: L [<Startadresse> [<Endadresse>]]

Version: Disk, ROM

Funktion: Darstellung des Programms in HEX und symbolisch, Darstellung der symbolischen Adressen und der Ablaufsteuermarken.

Bemerkungen: Fehlen die Adressgrenzen, werden die Grenzen des letzten L-Befehls benutzt.

Die Variable M wird aktualisiert

Beispiele:

L 1000	Listing ab 1000H
L	Listing mit alten Parametern
L M	Fortsetzung des Listings
L .OUT Y+20	Listung ab 'OUT' mit mindestens 32 Bytes

LP (LIST PROGRAM)

Format: LP

Version: Disk

Funktion: Darstellung des Programms ab augenblicklichem Programmzählerstand bis zur nächsten absoluten Verzweigung.

Bemerkungen: Die Variable M wird aktualisiert.

LD (LIST DATA)

Format: LD [<Startadresse> [<Endadresse>]]

Version: Disk, ROM

Funktion: Listing in Hex und ASCII, byteweise und mit symbolischen Adressen versehen.

Bemerkungen: Die Variable M wird aktualisiert.

Beispiele:

LD 1000	Listing ab 1000H
LD 1000 1100	Listing von 1000H bis 1100H
LD M	Fortsetzung des Listings

V (VIEW)

Format: V [<Adresse> ...]

Version: Disk, ROM

Funktion: Eingabe der Maske für Daten.

Anwendung: Gleichzeitige Darstellung von Datenbereichen in Verbindung mit der Register-Darstellung am Terminal.

Bemerkungen: V mit Argumenten übernimmt die Daten direkt. V ohne Argumente öffnet den Puffer zum Editieren.

Beispiele:

V 1000 Y+10	Direkte Eingabe von 2 Adressen in den Puffer.
V 1000 Y+10 10	öffnen des Puffers. Eingabe einer zusätzlichen Adresse.
V 1000 Y+10 10 <ESC>	Löschen des Puffers.

VP (VIEW PROGRAM)

Format: VP [<Zeilenzahl>]

Version: Disk, ROM

Funktion: Eingabe der Bildschirmmaske für Programm

Bemerkungen: Ist die Zeilenzahl zu groß, wird nur ein Teil dargestellt.

Beispiele:

VP 5	Es werden zusätzlich 5 Befehle dargestellt.
VP	Löschen der zusätzlichen Befehlszeilen.

P (PRINT)

Format: P [<Adresse>] [, <Datum> ...]

Version: Disk, ROM

Funktion: Eingabe von Daten in den Arbeitsspeicher

Bemerkungen: P mit Angabe von Daten schreibt die Werte direkt in den Arbeitsspeicher.

P mit oder ohne Adressangabe geht in den interaktiven Eingabemodus über.

P ohne Adressangabe setzt die Eingabe bei der nächsten Adresse fort.

Daten: Folgende Eingabedaten werden akzeptiert:

Byteausdrücke	Es wird jeweils ein Byte übernommen. Der Wert des arithmetischen Ausdrucks muß zwischen -256 und +255 liegen.
Adressen	Durch Voranstellen eines '#'-Zeichens werden arithm. Ausdrücke als Adressen interpretiert und als Doppelbyte übernommen.
ASCII-Strings	Der doppelte Anführungsstrich schließt Zeichenketten ein, die im ASCII-Format ohne Paritätsbit übernommen werden.

Die Übernahme in den Speicher erfolgt erst nach erfolgreicher Syntaxkontrolle.

Steuerzeichen: Im interaktiven Modus werden in erster Cursorposition und bei leerem Puffer folgende Steuerzeichen akzeptiert:

<CR>	Ende der Eingabe
<LF>	Listing des vorangehenden Blockes
<BS>	Adresse um 1 erniedrigen
<SP>	Adresse um 1 erhöhen
<ESC>	Löschen BATCH-Mode

Beispiele:

P 1000 23 -3	Eingabe von 2 Bytes ab Adresse 1000H
P ,45 'A	Fortsetzung der Eingabe mit 45H und ASCII 'A'
P	Interaktive Eingabe
P 1000 "ABC" 0D	Eingabe des Strings 'ABC' und 0DH ab 1000H
P , #1000	Eingabe des Wertes 1000H als Adresse

A (ASSEMBLE)

Format: A [<Adresse>] [<Symbol>:][<Mnemonic>[<Argumente>]]

Version: Disk, ROM

Funktion: Eingabe von Programmen in den Arbeitsspeicher

Bemerkungen: A mit Angabe von Daten schreibt den Befehl direkt in den Arbeitsspeicher.

A mit oder ohne Adressangabe geht in den interaktiven Mode über.

A ohne Adressangabe setzt die Eingabe mit der nächsten Adresse fort.

Marken: Marken dürfen aus maximal 6, bzw. 8 Zeichen bestehen und müssen mit ':' enden.

Mnemonics: Befehlsmnemonics sind im Zilog-Format anzugeben.

Argumente: Der Assembler akzeptiert alle numerischen Argumente. Hex-Konstanten müssen mit führender Ziffer beginnen. Die Variable $ beinhaltet den Wert des TRACE80-Programmzählers! Alle Argumente müssen kompakt eingegeben werden.

Steuerzeichen: Im interaktiven Modus werden in erster Cursor-Position und bei leerem Puffer folgende Steuerzeichen akzeptiert:

<CR>	Ende der Eingabe
<LF>	Listing des vorangehenden Blocks
<BS>	Adresse um 1 erniedrigen
<SP>	Adresse um Befehlslänge erhöhen
<ESC>	Löschen BATCH-Modus

Beispiele:

```
A 1000  LD A,B      Assemblierung ab 1000H

EEE: OR  23         Eingabe der Marke EEE und
                    des Befehls OR 23H

CALL NZ,.EEE        Eingabe von CALL NZ,EEE

DEFB "ABC"          Eingabe von ASCII 'ABC'
```

FILL

Format: FILL <Startadresse> <Endadresse> [<Datum>]

Version: Disk, ROM

Funktion: Laden einer Konstanten in den Speicher

Bemerkungen: FILL besetzt den Arbeitsspeicher mit einer Konstanten, bzw. mit 0

Gleichzeitig wird ein einfacher Speichertest durchgeführt.

Beispiele:

FILL 1000 2000 33	Laden von 1000H bis 2000H mit 23H
FILL 1000 2000	Laden von 1000H bis 2000H mit 0

MOVE

Format: MOVE <Startadresse> <Endadresse> <Ziel>

Version: Disk, ROM

Funktion: Kopieren eines Speicherbereichs

Bemerkungen: MOVE kopiert den Inhalt von Startadresse bis Endadresse nach Ziel. Die Bereiche dürfen auch überlappen. Gleichzeitig wird ein einfacher Speichertest durchgeführt.

Beispiele:

MOVE 1000 2000 3000	Kopieren von 1000H bis 2000H nach 3000H
MOVE 1000 2000 1000	Speichertest von 1000H bis 2000H

FCB (FILE CONTROL BLOCK)

Format: FCB <Dateiname> [<Adresse>]

Version: Disk

Funktion: Besetzen eines FCB's

Bemerkungen: Der Befehl FCB schreibt einen Dateinamen in einen File-Control-Block. Die Bytes 12 bis 15 werden mit 0 initialisiert.

Wird keine Adresse angegeben, wird der FCB bei 5CH besetzt.

Beispiel: FCB XXX.REL 1200 Der FCB auf Adresse 1000H wird besetzt.

BUF (BUFFER)

Format: BUF <Text>

Version: Disk

Funktion: Laden des Puffers bei 80H und besetzen der FCB's bei 5CH und 6CH.

Bemerkungen: Der Befehl BUF besetzt FCB's und den 'Default Buffer' in der gleichen Weise wie das Betriebssystem bei Aufruf eines Programmes.

Beispiele: BUF XXX YYY FCB 1 wird mit XXX besetzt, FCB 2 mit YYY. Zelle 80H enthält die Länge von <Text>, der Text steht ab 81H und endet mit 0H .

I (INPORT)

Format: I [<Portadresse> ...]

Version: Disk, ROM

Funktion: Lesen der Zustände der I/O-Ports

Bemerkungen: Der Befehl I liest I/O-Ports entsprechend den angegebenen Adressen. Es können bis zu 20 Adressen angegeben werden. Die Adressen werden gespeichert und können mit dem Befehl I ohne Angabe von Adressen erneut gelesen werden.

Beispiele:

I 10 11	Lesen der Ports 10H und 11 H
I	Wiederholtes Lesen der gleichen Ports

O (OUTPORT)

Format: O <Portadresse> <Datum>

Version: Disk, ROM

Funktion: Ausgabe eines Bytes über I/O-Port.

Bemerkungen: <Portadresse> muß ein arithmetischer Ausdruck mit Wert zwischen 0 und 255 sein. <Datum> ist ein Byte-Ausdruck.

Beispiele: D 10 'A Ausgabe von ASCII 'A' an Port 10H

R (REGISTER)

Format: R <Registername> [<Wert>] [, ...]

Version: Disk, ROM

Funktion: Setzen einzelner CPU-Register.

Bemerkungen: Beim Start des TRACE80 zeigt der Stackpointer auf das Ende des freien Speichers. Das Interrupt-Register bleibt auf dem Wert, den es bei Aufruf des Debuggers besitzt. Die restlichen Register werden mit 0 vorbesetzt.

Fehlt die Angabe eines Wertes, wird das betreffende Register gelöscht.

Das Komma ist bei Verwendung mehrerer Registernamen unbedingt zu verwenden.

Gültige Registernamen siehe Anhang B.

Beispiele:

R HL 1000,C 'A	Register HL wird mit 1000H, Register C mit 'A' besetzt.
R HL,C' A	Register HL wird mit 0, Register C' mit 0AH besetzt.

Besonderheiten: TRACE80 besitzt zwei zusätzliche Register, die nicht in der CPU implementiert sind.

Das Register SL (Subroutine Level) stellt die augenblickliche Unterprogrammebene dar. Bei jedem emulierten Z80-Return-Befehl wird der Inhalt dekrementiert, bei CALL- und RST-Befehlen um 1 erhöht.

Das Register SB (Stack Basis) zeigt auf den Anfang des Stacks. Sein Inhalt bestimmt die Anzeige der Stapeltiefe. Ist der Inhalt des Stackpointers größer SB, erscheint die Meldung STACK UNDERFLOW .

Beide Register können wie normale CPU-Register gesetzt und mit dem Befehl H gelesen werden.

K (KILL)

Format: K

Version: Disk, ROM

Funktion: Löschen sämtlicher Register

Bemerkungen: Es werden alle Register außer Programmzähler und Interrupt-Register gelöscht.Der Stackpointer wird initialisiert.

STO (STORE)

Format: STO

Version: Disk

Funktion: Der augenblickliche Registerzustand wird gespeichert.

Bemerkungen: TRACE80 besitzt einen zweiten Schatten-Register-Satz. Dieser wird mit STO geladen und kann mit REC wieder aufgerufen werden.

REC (RECALL)

Format: REC

Version: Disk, ROM

Funktion: Lesen des 2. Registersatzes

Bemerkungen: -

PC (PROGRAM COUNTER)

Format: PC [<Adresse>]

Version: Disk, ROM

Funktion: Setzen des Programmzählers

Bemerkungen: PC ohne Adresse setzt den Programmzähler auf die beim letzten PC-Befehl benutzte Adresse.

Beispiele:

PC 1000	Setzen des Programmzählers auf 1000H
PC $+3	Erhöhen des Programmzählers um 3

; (SKIP)

FORMAT: --Direktkommando--

Version: Disk

Funktion: Der Befehl springt über Befehle, ohne sie auszuführen.

Bemerkungen: Der jeweils nächste Befehl wird angezeigt.

H (HEX)
HF (HEX FUNKTION)

Format: H <Ausdruck>
Format: HF [<Ausdruck>]

Version: Disk

Funktion: Berechnen eines arithmetischen Ausdrucks

Bemerkungen: HF mit Argument übernimmt den arithmetischen Ausdruck direkt in den Funktions-Puffer. HF ohne Argument öffnet den Puffer zum Editieren.

Der aktuelle Wert des arithmetischen Ausdrucks kann unter der Funktionsvariablen X in anderen Ausdrücken verwendet werden.

Die Variable X darf im Befehl HF nicht verwendet werden (rekursiver Aufruf).

Der Befehl H berechnet den Wert eines arithmetischen Ausdrucks, ohne den Ausdruck in den Funktions-Puffer zu übernehmen.

Beispiele:

H .STACK	Der Wert des Symbols STACK wird angezeigt.
H @1000	Inhalt der Zellen 1000H und 1001H
HF X	nicht erlaubt !

B (BOOLE)
BF (BOOLE FUNKTION)

Format: B <Bedingung>
Format: BF [<Bedingung>]

Version: Disk

Funktion: Berechnung eines Booleschen Ausdrucks

Bemerkungen: BF mit Argument übergibt die Bedingung direkt zum Funktionspuffer.

BF ohne Argument öffnet den Puffer zum Editieren.

Der Wert der Funktion kann in anderen booleschen Ausdrücken unter der Variablen Q verwendet werden.

B berechnet den Wert einer Bedingung, ohne die Bedingung in den Funktionspuffer zu übernehmen.

Die Variable Q darf im Befehl BF nicht verwendet werden.

Beispiele:

B RHL>1000:Q:$>1000	Wahr wenn HL>1000H oder Programmzähler größer 1000H
BF Q	Nicht erlaubt !

CP (COMPARE)

Format: CP [<Startadr.> <Endadr.> <Vergleichsadr.>]

Version: Disk

Funktion: Vergleich zweier Speicherbereiche und Ausgabe der Unterschiede

Bemerkungen: Wird ein Unterschied gefunden, erscheint der Bindestrich als Bereitschaftszeichen. Der Vergleich kann durch Drücken der Leertaste fortgesetzt werden.

Die Variable Z enthält die Adresse der jeweils zuletzt gefundenen Differenz.

Der Befehl CP ohne Argumente benutzt die Parameter des letzten CP-Befehls.

Beispiele: CP 1000 1100 2000 vergleicht den Inhalt von 1000H bis 1100H mit dem Speicherinhalt ab 2000H

F (FIND)

Format: F [<Startadr.> <Endadr.>] [<Datum> ...]

Version: Disk, ROM

Funktion: Suchen nach HEX- und ASCII-Strings.

Bemerkungen: Bei Fehlen der Bereichsgrenzen wird innerhalb der zuletzt benutzten Grenzen gesucht.

Die Variable Z wird aktualisiert.

Mit <SP> kann der Suchvorgang fortgesetzt werden.

Die Daten müssen im selben Format erscheinen wie im Befehl P . Die Länge des Strings ist auf 40 Bytes begrenzt.

P 4-8

Beispiele:

F 1000 2000 "ABC"	Suchen im Bereich von 1000H bis 2000H nach dem ASCII-String 'ABC'.
F ,,"ERROR" 0D	Suchen mit alten Bereichsgrenzen nach dem String 'ERROR' und 0DH.
F 1000 3000	Suchen mit neuen Grenzen und dem zuletzt benutztem String.
F	Suchen mit alten Adressgrenzen und alten String

#hhhh 16 Bit Adresse

HELP

Format: HELP

Version: Disk

Funktion: Darstellung der Befehle des TRACE80

WAIT

Format: WAIT [<Wartezeit>]

Version: Disk, ROM

Funktion: Einstellung einer Wartezeit

Bemerkungen: Bei den Befehlen T, UT und PASS kann eine zusätzliche Wartezeit vor dem Bildschirmlöschen erzeugt werden.

Der Parameter ist vom Rechnertyp und der Prozessor-Geschwindigkeit abhängig.

In der Anpaßtabelle des TRACE80 kann ein Standardwert eingetragen werden.

Ohne Angabe eines Parameters ist die Wartezeit 0.

Beispiele: WAIT 1000

SS (SYMBOL SET)

Format: SS <Symbol> <Adresse> [, ...]

Version: Disk

Funktion: Erzeugen eines Symbols und Zuweisung eines Wertes.

Bemerkungen: Symbolnamen bestehen aus maximal 6, bzw. 8 Zeichen und müssen mit einem Buchstaben oder einem Sonderzeichen beginnen.

Neu generierte Symbole befinden sich immer am Anfang des Symbol-Puffers, so daß der Zugriff auf diese Namen recht schnell erfolgt.

Symbol-Puffer und TRACE80-Stack benutzen den gleichen Speicher. Es ist möglich, daß bei starker Verschachtelung von Ausdrücken die Meldung ARGUMENT ERROR erscheint, obwohl keine formalen Fehler vorliegen. In diesem Fall kann durch Löschen einiger Symbole die Fehlfunktion beseitigt werden.

Beispiele: SS NAME 1000 Das Symbol 'NAME' wird auf die Adresse 1000H gesetzt.

SC (SYMBOL CLEAR)

Format: SC [<Startadresse> [<Endadresse>]]

Version: Disk

Funktion: Löschen aller Symbole innerhalb eines bestimmten Bereichs.

bemerkungen: SC ohne Adressangabe löscht sämtliche Symbole.

Der Befehl LOAD löscht ebenfalls alle Symbole im Ladeadressbereich.

Beispiele:

SC 1000 2000	Löschen sämtlicher Symbole zwischen 1000H und 2000H.
SC 2000	Löschen sämtlicher Symbole über 2000H.

SD (SYMBOL DELETE)

Format: SD <Symbol> [...]

Version: Disk

Funktion: Löschen einzelner Symbole

Bemerkungen: -

Beispiele: SD XXX YYY Die Symbole XXX und YYY werden gelöscht.

SL (SYMBOL LIST)

Format: SL [<Startadresse> [<Endadresse>]]

Version: Disk

Funktion: Listing aller verwendeten Symbole in alphabetischer Reihenfolge.

Bemerkungen: Bei Angabe von Bereichsgrenzen werden nur Symbole dargestellt, die innerhalb dieser Grenzen liegen.

Beispiele:

SL 1000 2000	Listing aller Symbole zwischen 1000H und 2000H
SL	Listing aller Symbole

PROGRAMMTEST

Zu testende Programmteile können mit TRACE80 auf zwei Arten ausgeführt werden:

durch EMULATION oder in ECHTZEIT.

Echtzeittest erlauben Eingriffe in den Programmablauf nur an Haltepunkten. Die Ausführung des Programms erfolgt mit der maximalen Prozessorgeschwindigkeit.

Bei der Emulation wird jeder Befehl interpretiert. Der Zustand der simulierten CPU ist immer bekannt. Das Programm kann nach jedem Befehl angehalten werden. Durch die Emulation ist die Geschwindigkeit um den Faktor 100 bis 500 langsamer. Beim Befehl UT oder bei Aktivierung des Befehls W verlängert sich die Ausführungszeit zusätzlich um den Zeitbedarf zur Interpretation der Bedingung. Sind im Anwenderprogramm zeitaufwendige Programmteile vorhanden, sollten diese zuerst getestet und dann als Echtzeitbereich markiert werden.

Alle Emulationsbefehle testen die Abbruchbedingung erst nach dem ersten Emulationsschritt. Es wird daher immer mindestens ein Befehl emuliert.

, (SINGLE STEP)
.
:

Format: --Direktkommando--

Version: Disk, ROM

Funktion: Emulation eines Z80-Befehls

Bemerkungen: ',' emuliert einen Befehl, '.' druckt zusätzlich eine Kurzzeile am Drucker. Der Befehl ':' emuliert einen Z80-Befehl, ohne die Registerzustände anzuzeigen.

T (TRACE)

Format: T [<Anzahl>]

Version: Disk, ROM

Funktion: Emulation mehrerer Befehle mit Darstellung am Bildschirm

Bemerkungen: T ohne Argumente emuliert soviele Befehle, wie beim letzten T-Befehl.

<Anzahl> ist ein arithmetischer Ausdruck.

Beispiele:

T 10	Es werden 16 Befehle (10H) emuliert.
T %10	Es werden 10 Befehle emuliert.

U (UNTRACE)

Format: U [<Anzahl>]

Version: Disk, ROM

Funktion: Emulation mehrerer Befehle ohne Registerzustände und Befehlszeile nach jeder Emulation am Bildschirm oder Drucker auszugeben.

Bemerkungen: <Anzahl> ist ein arithmetischer Ausdruck.

Beispiele: U 10 16 Befehle emulieren.

UU (UNTRACE UNTIL)

Format: UU [<Adresse> [<Durchlaufzähler>]]

Funktion: Die Emulation wird fortgesetzt, bis die angegeben Adresse erreicht ist.

Bemerkungen: Ist der Durchlaufzähler größer 0, wird die Adresse entsprechend oft passiert.

Beispiele:

UU 1000	TRACE bis Programmzähler gleich 1000H
UU 1000,10	TRACE bis Adresse 1000H 16-mal durchlaufen.

UR (UNTRACE REGISTER)

Format: UR <Registername> [<Wert>]

Version: Disk

Funktion: Der Befehl steuert den Programmab-
in Abhängigkeit von CPU-Register-
inhalten.

Bemerkungen: Ohne Angabe eines Wertes wird bei
Registerinhalt 0 abgebrochen.

Der Befehl kann in Zählschleifen
angewandt werden.

Beispiele:

UR HL	TRACE bis HL gleich 0
UR CA 1	TRACE bis Carryflag gesetzt.

UL (UNTRACE LEVEL)

Format: UL [<Wert>]

Version: Disk, ROM

Funktion: Emulation, bis bestimmte Unterprogramm-ebene erreicht.

Bemerkungen: Der Befehl UL emuliert solange Befehle, bis die Anzahl der Unterprogrammaufrufe (CALL und RST) abzüglich der Zahl der Unterprogrammrücksprunge (RET) gleich dem Ausdruck <Wert> ist. Das Argument stellt also die relative Unterprogramm-ebene dar, bei der abgebrochen wird.

Soll der TRACE fortgesetzt werden, bis ein Unterprogramm beendet ist, muß mit negativen Argumenten gearbeitet werden.

Der Befehl UL ohne Argumente wirkt wie UL -1 (Rückkehr aus Unterprogramm).

Die Ausführung von Unterprogrammen in Echtzeit (Befehl C und J) ist dann nicht möglich, wenn z.B. die Parameterübergabe an das Unterprogramm hinter dem CALL-Befehl erfolgt. Hier kann mit dem Befhel UL (UL = J, UL 0 = C) fehlerfrei emuliert werden.

Beispiele:

UL 1	TRACE bis Unterprogramm
UL	TRACE bis aufrufendes Programm erreicht.
UL -2	TRACE bis Unterprogramm-ebene um 2 kleiner

UB (UNTRACE BRANCH)

Format: UB

Version: Disk, ROM

Funktion: Der Programmablauf wird solange fortgesetzt, bis das Programm verzweigt.

Bemerkungen: Der Befehl ist nur ausführbar, wenn der Programmzähler auf einem bedingtem Sprung steht (z.B. JR NZ, RET C, JP (HL)).

TRACE80 prüft die Richtung der Verzweigung und setzt die Emulation solange fort, bis an der gleichen Programmstelle in die andere Richtung verzweigt wird.

Der Befehl UB dient zum Beenden von Schleifen. Dabei muß bei Befehlsausführung der Programmzähler auf den Sprungbefehl zeigen.

Vorsicht bei Schleifen mit mehreren Ausgängen!

Beispiele: Programm

```
        LD      B,10
LOOP:   LD      A,(DE)
        INC     HL
        INC     DE
LP1:    DJNZ    LOOP
        LD      A,C
```

Das Programm kann bis LP1 mit Einzelschrittbefehlen emuliert werden. Dann wird mit UB die Schleife bis zum Absprung durchlaufen.

UT (UNTRACE TRUE)

Format: UT [<Bedingung1>[<Zähler>][<Bedingung2>]]

Version: Disk, ROM

Funktion: Emulation bis Bedingung1 erfüllt.

Bemerkungen: Der Befehl UT ist der langsamste und vielseitigste Emulationsbefehl. Im Gegensatz zu allen anderen Emulationsbefehlen wird der Wert der Argumente nach jedem Schritt neu berechnet.

Ist Bedingung1 erfüllt, wird der Stand des Bedingungszählers mit dem Wert von <Zähler> verglichen und bei Gleichstand die Emulation abgebrochen; sonst wird der Bedingungszähler un 1 erhöht. Bei Erfüllung der Bedingung2 wird am Terminal der CPU-Register-Zustand dargestellt.

Der Befehl UT ohne Argumente öffnet den Puffer zum Editieren.

Die Programmausführung kann beschleunigt werden, wenn statt symbolischer Adressen HEX-Adressen verwendet werden und Klammern nach Möglichkeit vermieden werden.

Beispiele:	UT $=1000,3,T	Adresse 1000H wird bis zum Abbruch 3-mal durchlaufen, die Registerzustände werden nach jedem Emulationsschritt dargestellt.
	UT N:T,,$=1000	TRACE ohne Abbruch, Anzeige der Registerzustände bei 1000H
	UT (RIX+2)=0	TRACE bis Inhalt der durch IX+2 adressierten Speicherzelle Null ist.
	UT S	TRACE bis symbolische Adresse.

HL	(HISTORY LIST)
HC	(HISTORY CLEAR)

Format: HL [<Anzahl der Befehle>]
Format: HC

Version: Disk, ROM

Funktion: HL listet die zuletzt emulierten Befehle.
HC löscht den Puffer.

Bemerkungen: TRACE80 speichert die Adressen der letzen 255 Befehle, die emuliert wurden. Mit HL kann dieser Puffer am Terminal dargestellt werden.

Beispiele:

HL %10	Listing der letzten 10 Befehle
HC	Löschen des Puffers

W (WATCH)

Format: W [<Bedingung>]

Version: Disk, ROM

Funktion: Setzen einer Abbruchbedingung

Bemerkungen: Für sämtliche Emulationsbefehle kann eine Abbruchbedingung gesetzt werden. Die Bedingung wird nach jedem Z80-Befehl geprüft. Somit wird immer mindestens ein Befehl ausgeführt.

Wird der Befehl W ohne Argument benutzt, kann die alte Bedingung editiert werden.

Ist die Bedingung erfüllt, erscheint vor dem Prompt ein Pluszeichen, ansonst ein Minuszeichen.

Die Geschwindigkeit der Emulation wird verlangsamt.

Beispiele:

W $>1000:0:$<100	Einschränkung des zulässigen Bereichs für den Programmzähler.
W RSP<RSB-200	Abbruch wenn Stack-Länge größer 200H

ECHTZEITAUSFÜHRUNG

Bei der Echtzeitausführung von Programmen hat der Bediener keinen Einfluß auf den Ablauf des Programms. Der Abbruch erfolgt durch einen Haltepunkt.

Haltepunkte werden realisiert, indem das erste Byte eines Z80-Befehls durch einen RESTART-Befehl ersetzt wird (Nur Disk-Version). Auf die zum RESTART-Befehl gehörende Unterprogrammadresse wird ein Sprung in den TRACE80 eingetragen.

Anwenderprogramme können durch eine Sprung auf Adresse 0 abgebrochen werden (CP/M Warmstart).

Steht bei Ausführung eines Echtzeitbefehls oder einer Ablaufsteuermarke ein Haltepunkt auf der Startadresse, wird zunächst ein Befehl emuliert und dann das Programm in Echtzeit fortgesetzt. Somit können die Echtzeitbefehle auch mit Adressen benutzt werden, die einen Haltepunkt beinhalten.

Haltepunkte dürfen nur auf die Adresse des ersten Bytes eines Z80-Befehls gesetzt werden. Sie sind nur im RAM zulässig (Disk-Version).

TRACE80 setzt beim Programmstart den Interrupt-Flip-Flop der CPU. Damit können Interrupts verarbeitet werden, solange keine Anwenderprogramme in Echtzeit ausgeführt werden. Das Register IFF des TRACE80 hat nur Gültigkeit für Echtzeitprogrammteile. Der Z80-Befehl EI gibt den Interrupt erst beim übernächsten Befehl frei. TRACE80 setzt schon bei nächsten Befehl das Register IFF !

G (GO)

Format: G [<Startadresse>] [<Haltepunkt> ...]

Version: Disk, ROM

Funktion: Starten eines Programms in Echtzeit

Bemerkungen: Der Befehl G setzt das Programm bei <Startadresse> in Echtzeit fort. Ohne Angabe eine Startadresse wird das Programm beim augenblicklichen Programmzählerstand fortgesetzt.

Zusätzlich können für die Dauer des Befehls 10 Haltepukte gesetzt werden. Bei Verwendung dieser Möglichkeit werden die zusätzlichen Haltepunkte gelistet. Mit <SP> kann das Programmm gestartet werden.

Beispiele:

G 1000	Echtzeitausführung ab 1000H
G	Echtzeitausführung ab $
G ,2000	Echtzeitausführung mit Haltepunkt auf 2000H

GB (GO BRANCH)

Format: GB

Version: Disk

Funktion: Beenden einer Schleife in Echtzeit

Bemerkungen: Der Befehl GB ist nur erlaubt, wenn der Programmzähler auf einer bedingten Verzweigung steht.

TRACE80 setzt einen Haltpunkt auf diese Adresse und setzt das Programm in Echtzeit fort. Der Haltpunkt wird solange ignoriert, wie die Verzweigungsrichtung derjenigen beim Befehlsaufruf entspricht.

Der Befehl ist das Echtzeit-Äquivalent zum Befehl-UB. Er dient zum schnellen Beenden von Schleifen.

Beispiele: Programm

```
WAIT:   LD      B,3
WAIT1:  PUSH    BC
        LD      BC,0
WAIT2:  DEC     BC
        LD      A,B
        OR      C
        JR      NZ,WAIT2
        POP     BC
WAIT3:  DJNZ    WAIT1
        RET
```

Das Programm kann, wenn der Programmzähler auf WAIT3 zeigt, mit dem Befehl GB beendet werden.

GN (GO NO BREAKPOINTS)

Format: GN [<Startadresse>] [<Haltepunkt> ...]

Version: Disk

Funktion: Starten eines Programmes in Echtzeit.

Bemerkungen: Der Befehl GN wirkt genau wie der Befehl G, mit dem Unterschied, daß die Haltepunkte nicht sofort gesetzt werden, sondern erst, wenn am Terminal ein bestimmtes Zeichen eingegeben wird (Control ' ').

Das Anwenderprogramm muß also auf die Betriebssystem-Funktion CONSOLE IN zugreifen.

Damit ist ein gezielter Abbruch des Anwenderprogramms möglich.

Beispiele:

GN ,1000,1200	Starten des Programms mit Breakpoint bei 1000H und 1200H

J (JUMP)

Format: J [<Startadresse>]

Version: Disk, ROM

Funktion: Beenden von Unterprogrammen in Echtzeit

Bemerkungen: Der Befehl setzt einen nichtresistenden Haltepunkt auf diejenige Adresse, die zuoberst im Stack steht.

Das Unterprogramm kann daher nur mit J in Echtzeit ausgeführt werden, wenn nicht noch weitere Werte im Stapel gesichert wurden. Der Befehl ist an allen Unterprogramm-Einsprungstellen erlaubt.

Beispiele: Unterprogramm

```
SUBR:   LD      A,C
SUBR1:  PUSH    HL
        ..
        ..
        POP     HL
SUBR2:  ..
        ..
SUBR3:  RET
```

Der Befehl J kann nur an den durch Marken gekennzeichneten Stellen benutzt werden.

C (CALL)

Format: C

Version: Disk, ROM

Funktion: Ausführung eines Unterprogramms in Echtzeit.

Bermerkungen: Steht der Programmzähler auf einem CALL- oder RST-Befehl, wird das Unterprogramm in Echtzeit ausgeführt.

Dazu wird ein nichtresistender Haltpunkt hinter den CALL- oder RST-Befehl eingetragen.

Die Taste '_' führt den Befehl C im Direktmodus aus. Wird auf der Adresse kein Unterprogramm-Aufruf gefunden, wird ein Emulationsschritt ausgeführt.

E (EXECUTE)

Format: E [<Startadresse>]

Version: Disk, ROM

Funktion: Ausführung eines Unterprogramms in Echtzeit.

Bemerkungen: Der Befehl dient zum Testen einzelner Module. In Gegensatz zum Befehl J wird der Stackpointer nicht inkrementiert. Der Programmzähler zeigt nach der Unterprogrammausführung auf den Anfang des Unterprogramms.

Wird keine Startadresse angegeben, wird angenommen, daß der Programmzähler bereits auf das Unterprogramm zeigt.

Beispiele:

E 1000	Ausführung des Unterprogramms bei 1000H

BS	(BREAK SET)
GS	(GO SET)
JS	(JUMP SET)
CS	(CALL SET)

Format1: ?S <Adresse> ...
Format2: ?S /<Maske> [<Startadresse>[<Endadresse>]]

Version: Disk, ROM

Funktion: Setzen von Ablaufsteuermarken

Bemerkungen: Mit dem Befehl BS können Haltepunkte gesetzt werden. Die Befehle GS, JS und CS markieren Programmstellen, an denen statt der Emulation der entsprechende Echtzeit-Befehl ausgeführt wird.

Die Zahl der Haltepunkte ist auf 40 begrenzt, die der Ablaufsteuermarken für J auf 40, für C und G auf jeweils 20.

Format1 gestattet, einzelne Marken durch Angabe der Adresse zu setzen.

Format2 generiert die Marken aus den verwendeten Symbolen innerhalb der angegebenen Adressgrenzen.

Durch sinnvolle Benennung der Marken im Quelltext können die Ablaufsteuermarken bereits dort definiert werden.

<Maske> ist ein String mit maximal 8 Zeichen. Ein Fragezeichen steht für einen beliebigen Buchstaben.

Beispiele:

BS 1000 1200	Setzen von Haltepunkten auf Adresse 1000H und 1200H
BS /L???????	Setzen von Haltepunkten auf alle Adressen, deren Symbol mit 'L' beginnt.

BC	(BREAK CLEAR)
GC	(GO CLEAR)
JC	(JUMP CLEAR)
CC	(CALL CLEAR)
MC	(MARK CLEAR)

Format: ?C [<Startadresse> [<Endadresse>]]

Version: Disk, ROM

Funktion: Die Befehle löschen alle Marken eines Typs, MC löscht alle Marken.

Bemerkungen: Wird zusätzlich ein Bereich angegeben, werden nur Marken innerhalb der Adressgrenzen gelöscht.

Beispiele:

BC	Löschen aller Haltpunkte
CC 1000 2000	Löschen aller Steuermarken für C im Bereich zwischen 1000H und 2000H

BD	(BREAK DELETE)
GD	(GO DELETE)
JD	(JUMP DELETE)
CD	(CALL DELETE)

Format1: ?D <Adresse> ...
Format2: ?D /<Maske> [<Startadresse>[<Endadresse>]]

Version: Disk, ROM

Funktion: Löschen einzelner Ablaufsteuermarken

Bemerkungen: Die Befehle BD, GD, JD und CD löschen die Ablaufsteuermarken, bzw. Haltepunkte an den entsprechenden Adressen.

Format1 gestattet, einzelne Marken durch Angabe der Adresse zu löschen.

Format2 löscht die Marken mit Hilfe einer Maske innerhalb der Adressgrenzen.

<Maske> ist ein String mit maximal 8 Zeichen. Ein Fragezeichen steht für einen beliebigen Buchstaben.

Beispiele:

BD 1000 1200	Löschen der Haltepunkte auf Adresse 1000H und 1200H
BD /L???????	Löschen aller Haltepunkte, deren Symbole mit L beginnen.

BL	(BREAK LIST)
GL	(GO LIST)
JL	(JUMP LIST)
CL	(CALL LIST)
ML	(MARK LIST)

Format: ?L [<Startadresse> [<Endadresse>]]

Version: Disk, ROM

Funktion: Listing der Ablaufsteuermarken eines Typs; ML listet alle Marken.

Bemerkungen: Durch eine Bereichsangabe werden nur Marken gelistet, deren Adresse innerhalb des angegebenen Bereichs liegt.

Beispiele:

BL 1000 2000	Listing aller Haltepunkte im Bereich zwischen 1000H und 2000H

PASS

Format: PASS [<Bedingung1> [<Zähler>] [<Bedingung2>]]

Version: Disk, ROM

Funktion: Eingabe einer Bedingung für Haltepunkte

Bemerkungen: PASS ohne Argumente öffnet den Puffer zum Editieren. PASS mit Argumenten übernimmt den Ausdruck direkt in den Puffer.

Der Befehl PASS erlaubt, Haltepunkte mit Bedingungen zu verknüpfen. Ist Bedingung1 wahr, wird der Haltpunkt ignoriert und das Programm in Echtzeit fortgesetzt. Der Programmablauf kann am Haltepunkt mit der Leertaste unterbrochen, bzw. mit <CR> unterbrochen werden. Die Größe <Zähler> gibt an, wie oft die Bedingung1 am Haltepunkt erfüllt sein muß, bis der Programmabbruch erfolgt (Durchlaufzähler für Haltepunkte).

Ist Bedingung2 wahr, wird der Registerzustand am Terminal angezeigt.

Beispiel 1:

```
BC                  Löschen sämtlicher Halte-
                    punkte
BS 1000             Setzen eines Haltepunkts
                    auf Adresse 1000H

PASS T %1000 RC=0   Eingabe einer Bedingung

G                   Starten des Programms in
                    Echtzeit
```

Der Haltepunkt auf 1000H wird 1000-mal durchlaufen, bis der Programm-Abbruch erfolgt. Am Terminal wird der Registerzustand dargestellt, sofern das Register C den Wert 0 besitzt.

Beispiel 2:

```
BC
BS 1000 1200
PASS T
G
```

Das Programm wird ohne Abbruch ausgeführt, läßt sich jedoch an den Stellen 1000H und 1200H unterbrechen.

NS (NMI SET)
NC (NMI CLEAR)

Format: NS
Format: NC

Version: Disk

Funktion: Setzen/Löschen der Option NMI

Bemerkungen: Der NON-MASKABLE-INTERRUPT der Z80-CPU kann benutzt werden, um Programme abzubrechen. Dazu muß ein Sprungbefehl auf die Adresse 66H eingetragen werden. Auf den gleichen Adressen steht ein FCB des CP/M.

Deshalb wird der Sprungbefehl nur bei gesetzter Option und bei Echtzeit-Befehlen eingetragen.

Der alte Zustand der Speicherzellen wird gerettet und nach Beendigung des Programmteils zurückgespeichert.

Zur Auslösung des NMI ist ein prellfreier Schalter oder eine Logikschaltung erforderlich.

SPL (SPOOLER)

Format: SPL <Dateiname>

Version: Disk

Funktion: Ausdruck einer Datei am Drucker

Bemerkungen: Mit SPL wird eine Datei zum Drucker übertragen. Der Zugriff durch andere Programme wird vorübergehend blockiert.

Da das Betriebssystem nicht reentrant ist, wird die Funktion solange suspendiert, wie sich der Programmzähler oberhalb des freien Speichers befindet.

Ohne Angabe eines Datei-Typs wird der Typ .PRN ausgedruckt.

Beispiele:

SPL XXX	Ausdruck von XXX.PRN
SPL XXX.	Ausdruck von XXX
SPL XXX.MAC	Ausdruck von XXX.MAC

SPE (SPOOLER END)

Format: SPE

Version: Disk

Funktion: Abbruch der Übertragung zum Drucker.

Bemerkungen: SPE erzeugt in jedem Fall einen Seiten-Vorschub.

LOAD

Format: LOAD <Datei> [<Ladeadr.> | <Limit> [<Offset>]]

Version: Disk

Funktion: Laden einer Datei

Bemerkungen: LOAD ließt eine Datei von der Diskette in den Speicher.

Ohne Adressangabe wird bei 100H geladen.

Offset bezieht sich auf die Diskette, d.h. der Wert Offset ist die Anzahl der Bytes, die von der Diskette gelesen werden, bis der Ladevorgang bei der Ladeadresse beginnt.

Bei Dateinamen ohne Angabe eines Typs wird der Typ .COM geladen.

Alle Symbole innerhalb des Ladeadressbereichs werden gelöscht. Befindet sich ein Symbolfile auf Diskette, werden alle Symbole innerhalb der Adressgrenzen geladen.

Beispiele:

LOAD XXX	Laden des Files XXX.COM und der Datei XXX.SYM
LOAD XXX 100 200	Laden der Datei XXX.COM ab 100H bis maximal 200H, einschließlich der zugehörigen Symbole

---

SAVE

Format: SAVE <Dateiname> <Startadresse> <Endadresse>

Version: Disk

Funktion: Übertragen des Arbeitsspeicher-Inhalts zur Diskette

Bemerkungen: SAVE kopiert den Speicherbereich auf ein Disketten-Datei. Ohne Angabe eines Datei-Typs wird ein .COM-File erzeugt.

Die freien Bytes des letzten Sektors werden mit 0 gefüllt.

Gleichzeitig wird ein Symbol-File generiert, das alle Symbole innerhalb der Adressgrenzen enthält.

Beispiele:

SAVE XXX 100 200	Kopiert den Bereich von 100H bis 200H auf die Datei XXX.COM

RUN

Format: RUN <Dateiname> [<Text>]

Version: Disk

Funktion: Laden und Starten eines Programmes

Bemerkungen: RUN lädt eine Datei bei 100H, besetzt die FCB's bei 5CH und 6CH und den Puffer ab 80H mit <Text> und startet das Programm bei 100H .

Die Rückkehr zum TRACE80 erfolgt entweder mit RET oder durch WARMBOOT.

Beispiele: RUN M80 Ausführung des Programmes M80.COM.

BOOT

Format: BOOT

Version: Disk

Funktion: Kaltstart des Betriebssystems

Bemerkungen: In der Anpaßtabelle des TRACE80 kann ein Sprung zur Kaltstartroutine, bzw. einige Befehle, die einen Systemreset auslösen, eingetragen werden.

BYE

Format: BYE

Version: Disk

Funktion: Übergabe der Ablaufsteuerung an das Betriebssystem

Bemerkungen: BYE übergibt die Kontrolle an das Betriebssystem (CCP). Vorher werden die Modifikationen innerhalb der BIOS-Sprungleiste beseitigt.

BAT (BATCH)

Format: BAT <Dateiname> [<Seite>]

Version: Disk

Funktion: Aufruf einer BATCH-Datei

Bemerkungen: Befehle des TRACE80 lassen sich in einer Textdatei speichern. Mit dem Befehl BAT wird die Datei gelesen und die Befehle ausgeführt. Für jeden Befehl muß eine neue Zeile verwendet werden. In einer Textdatei lassen sich mehrere Befehlsgruppen unterbringen, die durch <FF> getrennt sein müssen.

BAT ohne Angabe einer Seitenzahl greift immer auf die erste Seite (Seite 0) zu.

Vor Ausführung eines BAT-Befehls wird geprüft, ob am Terminal ein Zeichen vorhanden ist. Ist dies der Fall, wird die Kontrolle bis zum <CR> an das Terminal übergeben.

<ESC> in erster Cursor-Position und bei leerem Eingabepuffer löscht den Befehl BAT.

Solange der Befehl BAT oder BAR aktiv ist, wird vor dem Prompt ein 2. Stern gedruckt.

Der Dateityp für die Batch-Datei ist .T80

Beispiele:

BAT ZZZ 2	Seite 2 der Datei ZZZ.T80 wird als Batch-Datei aktiviert.
BAT ZZZ.ZZZ	Seite 0 der Datei ZZZ.ZZZ wird aktiviert.

BAR (BATCH REQUEST)

Format: BAR <Dateiname> [<Seite>]

Version: Disk

Funktion: Aufruf eines Batch-Datei

Bemerkungen: Befehle des TRACE80 lassen sich in einer Textdatei speichern. Mit dem Befehl BAR wird die Datei gelesen und der Befehl ausgeführt. Für jeden Befehl muß eine neue Zeile verwendet werden. In einer Textdatei lassen sich mehrere Befehlsgruppen unterbringen, die durch <FF> getrennt sein müssen.

BAR ohne Angabe einer Seitenzahl greift immer auf die erste Seite (Seite 0) zu.

Der BAR-Befehl liest einen Befehl in den Befehlspuffer des TRACE80 und übergibt die Kontrolle bis zum <CR> an das Terminal <ESC> in vorderster Cursor-Position und bei leerem Puffer löscht den Befehl BAR.

Solange der Befehl BAT oder BAR aktiv ist, werden als Prompt zwei Sterne ausgegeben.

Der Dateityp für die Batch-Datei ist .T80

Beispiele:

BAR ZZZ 2	Seite 2 der Datei ZZZ.T80 wird als Batch-Befehl aktiviert.
BAR ZZZ.ZZZ	Seite 0 der Datei ZZZ.ZZZ werden aktiviert.

SYSTEMANPASSUNG

Zur Anpassung des TRACE80 an das System sind ab Adresse 0A010H Tabellen vorhanden. Bei grossen Systemen koennen die Anpassungen mit TRACE80 selbst vorgenommen werden. Hierzu ist ein Symbol-File auf der Orginal-Diskette. Nach dem Aendern der Tabelle ist das geaenderte Programm mit SAVE 81 T80.COM des geaenderte Programm abzuspeichern. Vorher sollte zur Datensicherung eine Kopie des Orginals angelegt werden.

Auf der Orginaldiskette befinden sich zwei Versionen des Programms TRACE80.

T80.S06 ist eine Version, die Symbole bis zu 6 Zeichen verarbeitet.

T80.S08 verarbeitet Symbole bis zu einer Laenge von 8. Die Version ist nur auf Terminals mit einer Zeilenlaenge von mindestens 68 Zeichen lauffaehig.

DDT T80.S08

SAVE 80 T80.COM

100 5DFF

50H = 80D

```
                        ;-------------------------------------------------

                        ;       Dieser Bereich ist reserviert fuer den Befehl
                        ;       BOOT des TRACE80. Hier kann ein Sprung auf den
                        ;       Urlader des Rechners eingetragen werden, bzw.
                        ;       ein Programm, das den Urlader aktiviert und das
                        ;       Betriebssystem initialisiert. Innerhalb dieses
                        ;       Programmteils duerfen keine absoluten Spruenge
                        ;       verwendet werden, da diese Programmteil auf
                        ;       verschiedene Adressen relokalisert werden kann.

0A09    C3 0000         USEBOT::JP      0               ;Ohne Aenderung wird ein
                                                        ;Warmstart erzeugt.

0A0C    00 00 00 00             DEFB    0,0,0,0,0,0,0
0A10    00 00 00
0A13    00 00 00 00             DEFB    0,0,0,0,0,0,0
0A17    00 00 00
0A1A    00 00 00 00             DEFB    0,0,0,0,0,0,0
0A1E    00 00 00
0A21    00 00 00 00             DEFB    0,0,0,0,0,0,0
```

```
0A25    00 00 00

                        ;-------------------------------------------------

                        ;       Tabellen zur Anpassung von TRACE80 an das CRT
                        ;       und den Drucker

0A28    50              CONLEN::DEFB    80          ;Breite des Terminals

0A29    18              CONLIN::DEFB    24          ;Zeilenzahl des Terminals

0A2A    84              LSTLEN::DEFB    132         ;Breite des System-
                                                    ;druckers

0A2B    3E              MAXLIN::DEFB    62          ;Anzahl der Zeilen, bis
                                                    ;Seitenvorschub am
                                                    ;Drucker erfolgt; wird ein
                                                    ;Wert von 255 eingetragen,
                                                    ;erfolgt kein automatischer
                                                    ;Seitenvorschub

0A2C    FF              LINCTR::DEFB    255         ;Zeilenzaehler fuer Drucker;
                                                    ;ist der Wert 255, wird
                                                    ;immer erst ein Seitenvor-
                                                    ;schub erzeugt. Zulaessige
                                                    ;Werte sind 0 und 255.

0A2D    00              USEALF::DEFB    0           ;Alpha-Lock-Flag
                                                    ; 0 keine Wirkung
                                                    ; 255 die Wirkung der
                                                    ; Shift-Taste wird umgedreht.

                        ;-------------------------------------------------

                        ;       FLAGS

0A2E    00              BDOSRT::DEFB    0           ;       Echtzeitflag fuer Betriebs-
                                                    ;       systemaufrufe

                                                    ; 0     Betriebssystemauf-
                                                    ;       rufe werden immer in
                                                    ;       Echtzeit ausgefuehrt.

                                                    ; 255   Das Betriebssystem
                                                    ;       kann emuliert aus-
                                                    ;       gefuehrt werden.
```

```
0A2F    FF              STTSTF::DEFB    0FFH        ; Stack-Test-Flag

                                                    ; 0   Stack-Unterlauf wird
                                                    ;     nicht geprueft.

                                                    ; 255 Stack-Unterlauf wird
                                                    ;     geprueft.

                        ;---------------------------------------------------

                        ;       Symbol-Puffer

0A30    0200            DEFSYM::DEFW    512         ;Groesse des Symbol-Puffers,
                                                    ;falls beim Laden des
                                                    ;TRACE80 kein File geladen
                                                    ;wird oder nicht explizit
                                                    ;die Puffergroesse definiert
                                                    ;wird.

                        ;---------------------------------------------------

                        ;       Standard-Datei-Typen

0A32    43 4F 4D        USECOM::DEFM    'COM'       ;Befehle LOAD, SAVE, RUN
0A35    53 59 4D        USESYM::DEFM    'SYM'       ;Symbol-File
0A38    50 52 4E        USEPRN::DEFM    'PRN'       ;Befehl SPL
0A3B    54 38 30        USEBAT::DEFM    'T80'       ;Befehle BAT, BAR
0A3E    20 20 20        USEFCB::DEFM    '   '       ;Befehl FCB, BUF

                        ;---------------------------------------------------

                        ;       Steuerzeichen

0A41    10              CONTRP::DEFB    'P'-40H     ;Steuerzeichen fuer Voll-
                                                    ;ausdruck

0A42    14              CONTRT::DEFB    'T'-40H     ;Steuerzeichen fuer Teil-
                                                    ;ausdruck

0A43    1C              CONTRB::DEFB    '\'-40H     ;Steuerzeichen fuer
                                                    ;Befehl GN

                        ;---------------------------------------------------

                        ;       Direkt-Kommandos

                        ;       Die Kommandos bestehen aus genau einem Zeichen.
                        ;       Es sind alle Zeichen ausser Buchstaben zulaessig.
```

```
0A44                    DIRTAB::
0A44    20              BLANK:: DEFB    ' '             ;Keine Operation
0A45    0D              CR::    DEFB    0DH             ;Neue Zeile am Drucker
0A46    0A              LF::    DEFB    0AH             ;Register-Darstellung
0A47    0C              FF::    DEFB    0CH             ;Bildschirm-Loeschen
0A48    2C              EMU::   DEFB    ','             ;Einzelschritt
0A49    2E              EMUP::  DEFB    '.'             ;Einzelschritt mit Protokoll
0A4A    3B              SKIP::  DEFB    ';'             ;Ueberspringen eines Befehls
0A4B    3A              EMUS::  DEFB    ':'             ;Einzelschritt ohne Register-
                                                        ;darstellung
0A4C    5F              CAL::   DEFB    '_'             ;Echtzeit-Unterprogramm
0A4D    2F              REPEAT::DEFB    '/'             ;Wiederholen des letzten
0A4E    03              WBOOT:: DEFB    03H             ;Initialisierung des BDOS
0A4F    08              BS::    DEFB    08H             ;Keine Operation
0A50    1B              ESC::   DEFB    1BH             ;Loeschen BAT oder BAR
0A51    09              TAB::   DEFB    09H             ;Aufruf letzte Eingabe

                        ;---------------------------------------------------

                        ;       Steuerzeichen fuer Zeilen-Editierung

0A52                    EDICHA::
0A52    0D              ECR::   DEFB    0DH             ;Abschluss der Zeile
0A53    0A              ELF::   DEFB    0AH             ;Abschluss der Zeile
0A54    03              EETX::  DEFB    03H             ;Abschluss der Zeile
0A55    08              EBS::   DEFB    08H             ;Cursor zurueck
0A56    09              ETAB::  DEFB    09H             ;Cursor nach rechts
0A57    1B              EESC::  DEFB    1BH             ;Loeschen der Zeile und
                                                        ;Cursor zum Zeilenanfang.
0A58    0B              EINS::  DEFB    'K'-40H         ;Einfuegen eines Zwischen-
                                                        ;raums
0A59    7F              EDEL::  DEFB    7FH             ;Loeschen eines Zeichens

                        ;---------------------------------------------------

                        ;       Steuerzeichen zum Abbruch eines TRACE80-Befehls

0A5A                    USEBRK::
0A5A    1B              BESC::  DEFB    1BH             ;ESC
0A5B    0C              BFF::   DEFB    0CH             ;FF
0A5C    0A              BLF::   DEFB    0AH             ;LF
0A5D    0D              BCR::   DEFB    0DH             ;CR
```

```
0A5E  00                    DEFB   0            ;NUL
0A5F  00                    DEFB   0            ;NUL
0A60  00                    DEFB   0            ;NUL
0A61  03              BETX:: DEFB   03H          ;ETX

                      ;-----------------------------------------------------

                      ;     Steuerzeichen zur Unterbrechung eines TRACE80-Befehls

0A62  20              BRKWAI::DEFB  ' '          ;BLANK

                      ;-----------------------------------------------------

                      ;     Loeschen des Bildschirms

                      ;     Hier steht eine Steuersequenz, die den Bildschirm
                      ;     loescht und den Cursor zum Seitenanfang bewegt.

                      ;     Die Sequenz muss mit 0 enden !

0A63  ~~1E~~ 1B 4D 00 USECLR::DEFB  ''-40H       ;HOME  1   <ESC>,'M'  Master Reset
0A64  ~~1B 59~~ 48          DEFB   1BH,'Y'       ;ESC Y     <ESC>,'H'  Home
0A66  00                    DEFB   0
0A67  00                    DEFB   0             ;Ende

                      ;-----------------------------------------------------
ERASE EOS  <ESC> 'J' 4A
HOME       <ESC> 'H' 48
                      ;     Steuersequenz fuer Zeilen-Loeschen

                      ;     Die Steuerzeichen muessen die Zeile Loeschen
                      ;     und den Cursor zum Zeilenanfang bewegen.

                      ;     Die Sequenz muss mit 0 enden !

0A68  0D              CLRSTR::DEFB  0DH          ;Return
0A69  1B ~~54~~ 4B          DEFB   1BH,'T'       ;ESC 'T'   <ESC>,'K'  Erase EOL
0A6B  00                    DEFB   0
0A6C  00                    DEFB   0             ;Ende

                      ;-----------------------------------------------------

                      ;     Steuersequenzen fuer Drucker

                      ;     Diese Sequenz wird beim Aufruf des TRACE80 an
                      ;     Drucker uebertragen.

0A6D  00 00 00 00     INITP:: DEFB  0,0,0,0,0,0
```

```
0A71    00 00

                        ;       Steuersequenz fuer Drucker

                        ;       Diese Sequenz wird beim Befehl BYE an den
                        ;       Drucker uebertragen.

0A73    00 00 00 00     RETP::  DEFB    0,0,0,0,0,0
0A77    00 00

                        ;-------------------------------------------------

                        ;       Wartezeit fuer Druckerstatus

                        ;       Manche Drucker liefern nach dem Uebertragen eines
                        ;       Zeichens automatisch ein NOT-READY-Signal, obwohl
                        ;       der Drucker-Puffer nicht voll ist. Daher muss die
                        ;       Abfrage des Drucker-Status etwas verzoegert werden.
                        ;       Die Verzoegerung betraegt 23 Taktzyklen pro Einheit.

0A79    64              WAITST::DEFB    100

                        ;-------------------------------------------------

                        ;       OP-Code des verwendeten RST-Befehls

                        ;       Zulaessig sind  RST 8
0A7A    D7                                      RST 10H
                        ;                       RST 18H
                        ;                       RST 20H
                        ;                       RST 28H
                        ;                       RST 30H
                        ;                       RST 38H

                        ;               soweit das Betriebssytem keinen der
                        ;               Vektoren belegt.

0A7B    FF              BRKRST::RST     38H

                        ;-------------------------------------------------

                        ;       Standard-Wert fuer Befehl VP

0A7C    04              ARGVP:: DEFB    4
```

```
                        ;-------------------------------------------------

                        :       Standardwert fuer Befehl WAIT

 0A7D    01F4           ARGWAI::DEFW    500

                        ;-------------------------------------------------
```

ANHANG B

Zulässige Registernamen

Einfachregister

A	Akkumulator
F	Flag-Register
B	Register B
C	Register C
D	Register D
E	Register E
H	Register H
L	Register L
HX	High-Byte Indexregister IX
LX	Low-Byte Indexregister IX
HY	High-Byte Indexregister IY
LX	Low-Byte Indexregister IY
A'	Register A'
F'	Register F'
B'	Register B'
C'	Register C'
D'	Register D'
E'	Register E'
H'	Register H'
L'	Register L'
I	Interrupt-Register
R	Refresh-Register

Doppelregister

AF	Register AF
BC	Register BC
DE	Register DE
HL	Register HL
AF'	Register A'F'
BC'	Register B'C'
DE'	Register D'E'
HL'	Register H'L'
IX	Indexregister IX
IY	Indexregister IY
SP	Stapelzeiger
PC	Programm-Zähler
SL	Unterprogramm-Ebene
SB	Stapel-Anfang

Flag-Register

CA	Carry
N	Additions-/Subtraktions-Flag
P	Paritäts-Flag
HC	Halbübertrags-Byte
Z	Zero-Flag
S	Vorzeichen-Flag
CA', P'	Flags des 2. Registersatzes
N', HC'	
Z', S'	
IFF	Interrupt-Flip-Flop