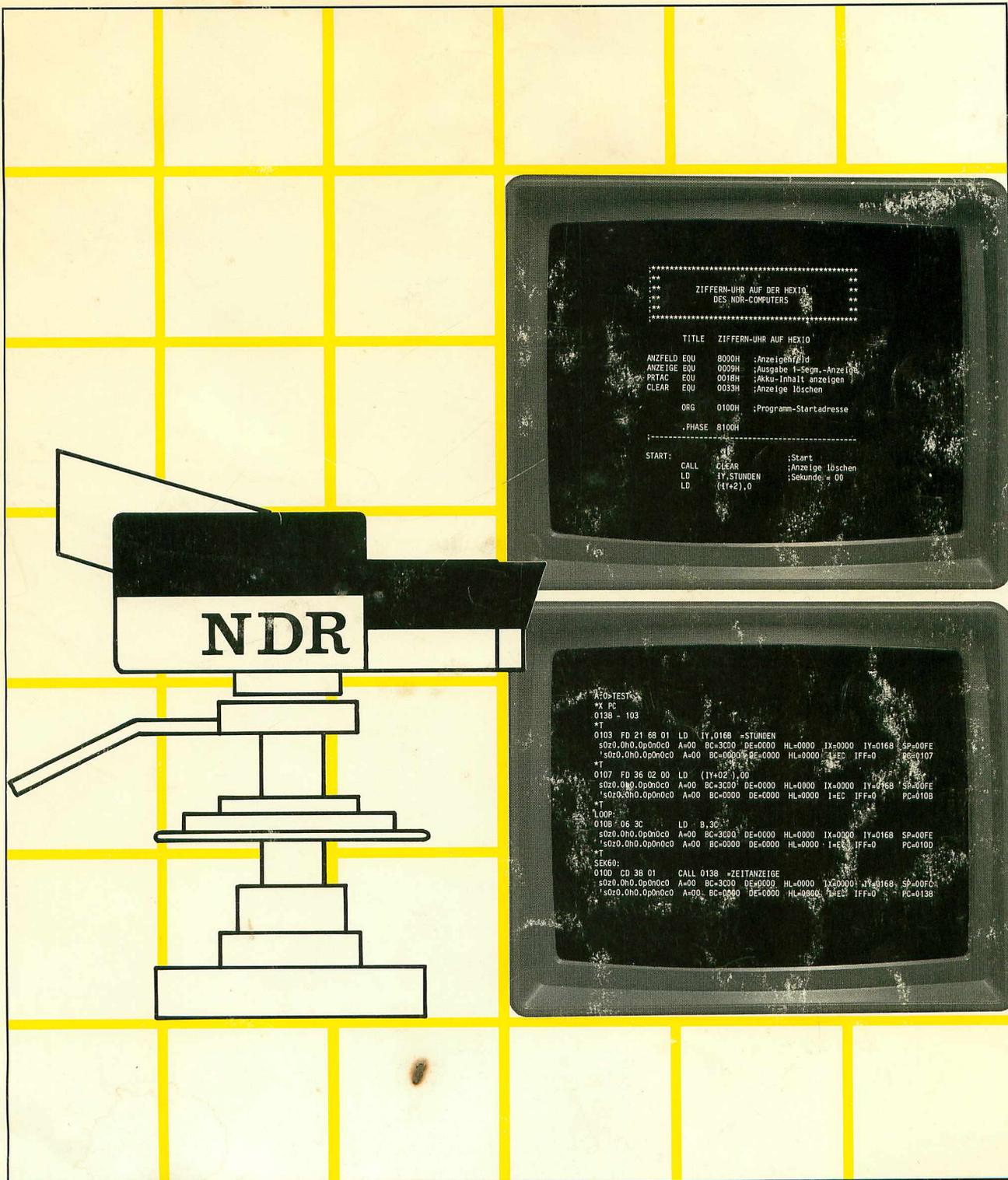


Mikroelektronik ZEAT-Betriebssystem



NDR

```
*****
***          ZIFFERN-UHR AUF DER HEXIO          ***
***          DES NDR-COMPUTERS                  ***
*****
```

```
TITLE ZIFFERN-UHR AUF HEXIO
ANZFELD EQU 8000H ;Anzeige feld
ANZEIGE EQU 0009H ;Ausgabe 1-Segm.-Anzeige
PRTAC EQU 0018H ;Akku-Inhalt anzeigen
CLEAR EQU 0033H ;Anzeige löschen
ORG 0100H ;Programm-Startadresse
```

```
.PHASE 8100H
START: CALL CLEAR ;Start
LD IV,STUNDEN ;Anzeige löschen
LD (IV+2),0 ;Sekunde = 00
```

```
;;TEST
*X PC
0138 - 103
**T
0103 FD 21 E8 01 LD IV,0168 ;STUNDEN
s0z0.0h0.0p0n0c0 A=00 BC=3C00 DE=0000 HL=0000 IX=0000 IY=0168 SP=00FE
s0z0.0h0.0p0n0c0 A=00 BC=0000 DE=0000 HL=0000 I=EC IFF=0 PC=0107
**T
0107 FD 36 02 00 LD (IV+02),00
s0z0.0h0.0p0n0c0 A=00 BC=3C00 DE=0000 HL=0000 IX=0000 IY=0168 SP=00FE
s0z0.0h0.0p0n0c0 A=00 BC=0000 DE=0000 HL=0000 I=EC IFF=0 PC=010B
**T
LOOP: 0108 06 3C LD B,3C
s0z0.0h0.0p0n0c0 A=00 BC=3C00 DE=0000 HL=0000 IX=0000 IY=0168 SP=00FE
s0z0.0h0.0p0n0c0 A=00 BC=0000 DE=0000 HL=0000 I=EC IFF=0 PC=010D
**T
SEK60: 0109 CD 38 01 CALL 0138 ;ZEITANZEIGE
s0z0.0h0.0p0n0c0 A=00 BC=3C00 DE=0000 HL=0000 IX=0000 IY=0168 SP=00FC
s0z0.0h0.0p0n0c0 A=00 BC=0000 DE=0000 HL=0000 I=EC IFF=0 PC=0138
```


Inhaltsverzeichnis

	Seite
Christiani-Betriebssystem	1
Kommando-Manager	1
Systemkommandos	1
Druckerprotokoll, Ausgabe-Start/Stop	3
Seitenformatierung Druckausgabe	3
Kommentare	3
Speicheraufteilung	4
Editor	5
Cursorbewegungen	6
Scrollfunktionen	6
Löschfunktionen	7
Blockoperationen	7
Einfügen	8
Randbegrenzung/word wrap	8
Druckersteuerung	8
Fehlermeldung	9
Abspeichern	9
Ende	9
Assembler	9
Z80-Befehle	10
Konstanten	10
Strings, Zeichenketten	11
Symbole	11
Rechenausdrücke	11
Kommentare	12
Pseudobefehle	12
Fehlermeldungen	15
Tester	
Kommandos	15
Telefonmodem-Betriebsprogramm	24
Kommandos	24
Modemparameter	25
Systemfunktionen	26
Standard-Systemfunktionen	26
Erweiterte Systemfunktionen	30
Z80-Befehle	35

© 1985 by Dr.-Ing. P.Christiani GmbH. Als Manuskript gedruckt.
Jedes Veräußern, Verleihen und sonstiges Vertreiben dieses Handbuchs, auch aus-
zugsweise, ist verboten.

Christiani-Betriebssystem

Das Christiiani-Betriebssystem besteht aus dem Editor, dem Assembler, dem Tester und dem Modemprogramm. Nach dem Erscheinen des Start-Menüs wird das Betriebssystem durch die Eingabe der Ziffer 3 aufgerufen. Bei diesem Kaltstart werden alle erforderlichen Speicherinitialisierungen vorgenommen, der Kommando-Manager aktiviert und die folgende Systemmeldung auf dem Bildschirm veranlaßt:

Christiani

Z80
Editor
Assembler
Tester

Nach dem Ende eines Benutzerprogramms oder nach der Ausführung eines Kommandos des Betriebssystems wird automatisch ein Warmstart durchgeführt. Dabei wird der Kommando-Manager reaktiviert. Es werden aber keine Speicherinitialisierungen vorgenommen, so daß die Speicherinhalte erhalten bleiben. Der Kommando-Manager ermöglicht den Aufruf der einzelnen Teile Des Betriebssystems über die Tastatur. Mit dem Kommando EXIT ist ein Rücksprung ins Grundsystem möglich.

Kommando-Manager

Der Kommando-Manager zeigt die Bereitschaft zur Annahme eines Systemkommandos mit einer eckigen Klammer am Anfang der Zeile an. Über die Tastatur kann dann eines der nachstehenden Kommandos eingegeben werden. Es wird ausgeführt, nachdem die Eingabe mit der Taste RETURN abgeschlossen wurde oder die Kommandozeile voll ist.

Eingabefehler können mit den Tasten DELETE oder BACKSPACE korrigiert werden. Die ganze Zeile läßt sich mit CTRL-X löschen. Die Ausführung der Kommandos ASM, TYPE und TYPE/L kann durch betätigen beliebiger Nicht-Buchstabentasten abgebrochen werden (wird eine Buchstabentaste gedrückt, so wird diese gespeichert, es können also Kommandos im voraus eingegeben werden. Der Kommando-Manager macht auf fehlerhafte Kommandos mit einem Piepsen und einem Fragezeichen auf dem Bildschirm aufmerksam.

Systemkommandos

Über den Kommando-Manager können folgende Systemkommandos aufgerufen werden:

MEMORY	Speicheraufteilung anzeigen
CLRSYM	Symboltabelle löschen
PACK	Größe des Textspeichers ändern
P	Seitenformatierung bei Druckausgabe
;	Kommentar
EDIT	Texteditor
TYPE	Ausgabe des gespeicherten Textes auf Bildschirm
TYPE/L	Ausgabe des gespeicherten Textes auf Drucker
ASM	Z80-Assembler
ASM/L	Z80-Assembler mit Druckausgabe
RUN	Anwenderprogramm ab Adresse 0100H starten
TEST	Z80-Tester
MODEM	Telefonmodem-Betriebsprogramm
PROFI	Freigabe geschützter Systemadressen
EXIT	Rücksprung ins Grundsystem
ZEAT	Verschiebung des Betriebssystems im Speicher (z.B. ZEAT 1000 = Anfang des Betriebssystems um 1000 Speicherstellen nach unten), um geschützten Speicherraum zu schaffen.

Drucker-Protokoll und Ausgabe- Start/Stop

Nach der Eingabe von CTRL-P werden alle Eingaben und Systemmeldungen auf einem angeschlossenen Drucker protokolliert. Das Druckerprotokoll läßt sich durch erneute Eingabe von CTRL-P abschalten.

Eine Ausgabe auf dem Bildschirm (z.B. durch TYPE) läßt sich durch CTRL-S anhalten. Die Ausgabe wird nach Betätigen einer beliebigen Taste fortgesetzt.

Druckerprotokollierung und Ausgabe-Start/Stop ist für den Dialog mit dem Kommando-Manager und mit allen Kommandos, außer dem bildschirmorientierten Texteditor, möglich. Benutzerprogramme können die automatische Druckerprotokollierung und die Ausgabe-Start/Stop-Funktion ebenfalls nutzen, wenn sie für Ein- und Ausgabe die Systemfunktionen verwenden.

Seitenformatierung Druckausgabe

Bei der mit CTRL-P zugeschalteten Druckausgabe oder der mit TYPE/L veranlaßten Druckausgabe des Textspeichers wird nach jeweils 66 Zeilen ein Seitenvorschub auf dem Drucker ausgelöst. Mit

P kann jederzeit ein Seitenvorschub ausgelöst, mit

P/data/ gleichzeitig die neue Seitengröße auf /data/ Zeilen pro Seite gesetzt werden.

P 0 schaltet die Seitenformatierung aus.

Die Funktion P kann nicht nur vom Kommando-Manager, sondern auch vom Tester und vom Modem-Betriebsprogramm aus benutzt werden.

Kommentare

Die am Zeilenanfang mit ";" markierten Kommandos werden als Kommentare interpretiert und bewirken nichts. Bei zugeschalteter Druckausgabe kann diese Funktion zur Kommentierung der Druckausgabe verwendet werden.

Die Kommentarfunktion kann nicht nur vom Kommando-Manager, sondern auch vom Tester und vom Modem-Betriebsprogramm aus benutzt werden.

Hinweis: Nach dem ";" muß ein Zwischenraum oder ein TAB eingegeben werden.

Speicheraufteilung

Das TLC90-Betriebssystem hat den Speicherbereich von 0000H bis 00FFH ähnlich belegt wie CP/M:

0000H bis 0002H Sprung zur Warmstartprozedur des Betriebssystems. Ermöglicht einen einfachen Warmstart von Anwenderprogrammen aus mit einem Sprung nach Adresse 0 oder dem RST0-Befehl.

0003H Intel standard IOBYTE

0005H bis 0007H Sprung ins Betriebssystem für Systemfunktionen. Adresse 0006H/0007H enthält die niedrigste vom Betriebssystem benutzte Adresse. Das Betriebssystem ändert diesen Wert je nach Größe von Textspeicher und Symboltabelle.

0008H bis 0037H Interrupt/Restart-Adressen 1 bis 6. Werden vom Betriebssystem (Assembler, Editor) teilweise benutzt, stehen dem Benutzer aber in Anwenderprogrammen zur Verfügung. Benutzt ein Anwenderprogramm die erweiterten Systemfunktionen, dann müssen die Restart-Adressen 1 und 2 (08H und 10H) unberührt bleiben (siehe auch unter Systemfunktionen).

0038H bis 003AH Restart-Adresse 7. Wird vom Tester für Breakpoints benutzt und sollte daher in Anwenderprogrammen nicht verwendet werden.

003BH bis 005BH Nicht benutzt, für CP/M reserviert.

005CH bis 00FFH Nicht benutzt, wird in CP/M für Filecontrol-Blocks und Disk-Buffer verwendet.

Danach folgen drei variable Speicherbereiche:

TPA Anwenderprogramm-Bereich von 0100H bis zur Symboltabelle.

SYM Symboltabelle.

TXT Text.

Die momentane Größe und Belegung dieser Bereiche kann mit

MEMORY oder einzeln mit
 MEMORY TPA
 MEMORY SYM
 MEMORY TXT abgefragt werden.

Die Symboltabelle kann mit CLRSYM gelöscht werden. Die TPA wird dadurch größer.

Der Textspeicher kann mit PACK vergrößert oder verkleinert werden. PACK alleine reduziert den Textspeicher auf die gerade benötigte Größe (Meldung auf dem Bildschirm: TXT FREE= 0). PACK mit nachfolgendem Ausdruck (Symbol oder Zahl) reserviert zusätzlich so viele Byte wie im Ausdruck angegeben. Beispiel: PACK 100H führt zur Meldung TXT FREE 0100.

Texteditor

Der Texteditor wird mit dem Kommando EDIT aufgerufen. Er schreibt zunächst den im Speicher befindlichen Text (so weit vorhanden) auf den Bildschirm und setzt den Cursor (Schreibmarke) in die linke obere Ecke des Bildschirms.

Danach kann sofort mit der Texteingabe begonnen werden. Die eingegebenen Zeichen werden anstelle des Cursors in den Text eingefügt. Jede mit Wagenrücklauf (Taste Return) abgeschlossene Zeile wird am rechten Rand mit einer eckigen Klammer markiert. Ist eine Zeile länger als der Bildschirm erlaubt, wird dies mit einem Pluszeichen (+) am rechten Rand angezeigt. Vorhandene Zeichen können wahlweise überschrieben oder nach rechts weitergeschoben werden (INSERT ON/OFF). Beim Überschreiten des programmierten rechten Randes kann das nicht mehr in die Zeile passende Wort wahlweise automatisch in die nächste Zeile gesetzt oder außerhalb des rechten Randes geschrieben werden (WORD-WRAP ON/OFF).

Mit den Steuertasten für den Cursor kann dieser innerhalb des Textes waagrecht und senkrecht bewegt werden. Mit den Scrollfunktionen kann der gesamte Text durchgeblättert werden.

Falsche Eingaben können zeichenweise, wortweise oder in markierten Textblöcken gelöscht werden. Außerdem können Blöcke innerhalb des Textes frei definiert und dann im Text verschoben, kopiert und, wie erwähnt, gelöscht werden.

Fertige Texte können auf Audiocassette (oder Diskette) gespeichert werden und später von diesen Speichern wieder in den Speicher des Rechners geladen werden.

Zusätzlich besteht die Möglichkeit, zwischen deutschem und amerikanischem Zeichensatz zu wählen sowie Kontrollzeichen für die Ausgabe über einen Drucker in den Text einzufügen.

Im Folgenden werden die Funktionen und Kommandos des Texteditors im einzelnen beschrieben.

Cursorbewegungen

Der Cursor läßt sich nicht nur mit den am rechten Rand der Tastatur befindlichen Steuertasten in der Richtung des jeweils aufgedruckten Pfeils bewegen. Vielmehr gibt es innerhalb der eigentlichen Schreibastatur vorprogrammierte Steuertasten, die durch gleichzeitiges Drücken der Controll-Taste folgende Bewegungen des Cursors ergeben:

CTRL-S	Zeichen links	CTRL-D	Zeichen rechts
CTRL-A	Wort links	CTRL-F	Wort rechts
CTRL-E	Zeile hoch	CTRL-X	Zeile runter

Abgesehen davon, daß mit diesen Steuertasten Bewegungen um Wortlängen möglich sind, ist ihr Gebrauch bei ausgesprochener Textverarbeitung nach kurzer Gewöhnung einfacher als derjenige der außerhalb der Schreibastatur liegenden Steuertasten. Dazu kommt, daß die Anordnung in der linken Tastaturhälfte leicht merkbar ist und der Richtung der Cursorbewegung entspricht:

		E		
A	S	D	F	
		X		

Wie die Cursorstueertasten am rechten Rand der Tastatur haben auch die Steuertasten innerhalb der Schreibastatur Repeatfunktion, d.h. bei längerem Drücken der Tasten wird die jeweilige Cursorbewegung wiederholt. Anders gesagt: Der Cursor wandert in der gewählten Richtung Schritt für Schritt weiter.

Scrollfunktionen

Mit den Scrollfunktionen kann im Textspeicher vor- und zurückgeblättert werden. Sie werden wie die Cursorstueertasten durch gleichzeitiges Betätigen der Controll-Taste aufgerufen:

CTRL-C	Seite rückwärts	CTRL-R	Seite vorwärts
CTRL-W	Zeile rückwärts	CTRL-Y	Zeile vorwärts

Die Anordnung der Steuertasten für die Scrollfunktionen fügt sich sinnfällig neben die Cursorstueertasten:

		W	E	R
A	S	D	F	
		Y	X	C

Auch die Steuertasten für die Scrollfunktionen haben Repeateigenschaft und wiederholen die jeweilige Bewegung bei längerem Drücken der Tasten. Beim Betätigen der Taste Tab (Tabulator) wird der Cursor um acht Zeichen nach rechts versetzt. Damit wird die Erstellung von Tabellen oder auch Programmen vereinfacht. Wenn der Cursor in einem geschriebenen Text auf einem Zeichen steht, wird dieses Zeichen gelöscht und der Rest der Zeile um den Raum von acht Zeichen nach rechts geschoben. Die gleichen Ergebnisse wie beim Betätigen der Taste Tab können mit der Steuertaste CTRL-I erzielt werden. Bei manchen Tastaturen liegt die Funktion Tab auch auf der Taste mit dem nach rechts weisenden Pfeil.

Löschfunktionen

Zum Löschen einzelner Zeichen, Worte oder ganzer Zeilen innerhalb eines Textes dienen folgende Funktionen:

CTRL-G	Zeichen unter Cursor	CTRL-T	Wort rechts
CTRL-Z	Zeile	Taste Del	Zeichen links

Die Delete-Funktion liegt bei manchen Tastaturen auch bei der Taste mit dem nach links weisenden Pfeil. Es können auch ganze Textabschnitte gelöscht werden, wenn sie vorher als Block markiert werden, wie anschließend bei den Blockoperationen gezeigt.

Blockoperationen

Die Steuerzeichen für die Blockoperationen werden durch die Taste Esc (ESCAPE) und eine nachfolgend (nicht gleichzeitig wie bei den mit CTRL- gewählten Funktionen) gedrückte Taste dargestellt. Mit den Blockoperationen können beliebig ausgesuchte Textabschnitte (Blöcke) markiert und manipuliert werden. Der Block wird auf dem Bildschirm mit (Anfang) und <K> (Ende) oder durch unterschiedliche Helligkeit markiert.

ESC B	Block Anfang markieren
ESC K	Block Ende markieren
ESC C	Block an Cursorposition kopieren
ESC V	Block an Cursorposition verschieben
ESC Z	Block löschen

Mit ESC C wird der markierte Block an der durch den Cursor bezeichneten Stelle kopiert, bleibt zugleich an der ursprünglichen Stelle erhalten. Mit ESC V hingegen wird er an der ursprünglichen Stelle gelöscht. Wenn der Cursor aus irgendwelchen Gründen innerhalb eines Textes plziert ist, wird dieser durch den kopierten oder verschobenen Block nicht überschrieben, der Block wird vielmehr eingefügt.

Einfügen

Einfügungen in vorhandenen Text sind nicht nur in Gestalt verschobener Blöcke möglich. Auch neu eingegebene Zeichen können in vorhandenen Text eingefügt werden, anstatt ihn zu überschreiben. Der Text wird dabei nach rechts weitergeschoben. Diese Funktion wird mit

ESC I INSERT ON/OFF

ein- und ausgeschaltet. Wenn dabei der vorhandene Text über den rechten Rand hinausgeschoben wird, zeigt dies ein Pluszeichen (+) am rechten Rand an. Um den verschwundenen Text wieder hereinzuholen, muß an der Stelle vor dem Pluszeichen (oder bei programmiertem rechten Rand an dessen Stelle) ein Wagenrücklauf (Taste Ret) eingefügt werden (mit ESC I).

Randbegrenzung und WORD-WRAP

Die Zeilenbreite des Textes kann schmaler als die Bildschirmbreite gemacht werden. Der gewünschte rechte Rand wird durch die Cursorstellung markiert und mit

ESC M MARGIN

programmiert. Über diesen programmierten rechten Rand hinausgehende Worte werden automatisch in die nächste Zeile gesetzt mit der Funktion WORD-WRAP. Diese wird mit

ESC W WORD-WRAP ON/OFF

ein- und ausgeschaltet. Normalerweise, d.h. wenn ESC M nicht benutzt wird, ist die Funktion WORD-WRAP automatisch eingeschaltet.

Druckersteuerung

Manche Drucker benötigen für interne Steuerfunktionen (z.B. Farbbandumschaltung usw.) Kontrollcodes, die normalerweise nicht in den Text eingefügt werden können (Ausnahmen CTRL-N für neue Zeile und Tab bzw. CTRL-I). Solche Kontrollcodes können mit

ESC P Buchstabe

eingegeben werden. Der gewünschte Kontrollcode wird direkt nach dem P eingegeben, also z.B. für CTRL-G steht dann ESC P G. Diese Funktion ist vorsichtig zu handhaben. Beispiel:

ESC P Z würde den gesamten Text ab Cursorposition löschen (CTRL-Z=End of Text). Die Textausgabe auf dem Drucker kann durch den Kommando-Manager mit dem Kommando TYPE/L durchgeführt und mit CTRL-C abgebrochen werden.

Fehlermeldung

Bei Überfüllung des Textspeichers erscheint *FULL*. Der Fehlerzustand wird aufgehoben durch Verkürzen des Textes oder Vergrößern des Textspeichers. Hierzu wird der Editor mit ESC E verlasen und mit dem Kommando PACK ein größerer Textspeicher bereitgestellt (z.B.PACK 1000 ergibt einen um 1000 Zeichen größeren Textspeicher).

Abspeichern

Ein Abspeichern auf Cassete oder Diskette ist mit ESC S (Save) möglich. Auf dem Bildschirm wird angezeigt:

*SAVE?

Das System möchte einen Namen in Gestalt eine vierstelligen Sedezimalzahl wissen, unter dem die Datei abgespeichert werden soll.

Das Einlesen einer Datei von der Cassete in den Textspeicher wird durch die Eingabe von ESC A (Append) veranlaßt. Mit der Anzeige

*APPEND?

wird nach dem Namen der Datei gefragt.

Ende

Nach Beendigung der Textverarbeitung kann ins Betriebssystem zurückgekehrt werden mit

ESC E EXIT

Verbunden mit nachfolgendem Neuaufruf des Texteditors kann diese Funktion auch dazu benutzt werden, um bei längeren Texten schnell vom Textende zum Textanfang zu gelangen.

Assembler

Der TLC90 Assembler wird mit dem Kommando ASM aufgerufen. Er assembliert dann (Taste Ret) das im Textspeicher stehende Quellprogramm in zwei Durchläufen. Im ersten Durchlauf wird eine Symboltabelle generiert, im zweiten

Durchlauf wird das übersetzte Programm in den Speicher geschrieben. Während der Assemblierung zeigt der Assembler auf dem Bildschirm mit pass 1 bzw. pass 2 an, in welchem Durchlauf er sich befindet.

Außerdem schreibt der Assembler nach jeweils 256 Bytes assemblierten Quelltext ein Pluszeichen auf den Bildschirm. Bei sehr langen Programmen kann der Benutzer dadurch verfolgen, daß der Assembler arbeitet und nicht etwa ausgestiegen ist (Im gleichen Abstand wird auch die Tastatur abgefragt, so daß die Assemblierung mit CTRL-C abgebrochen werden kann).

Nach der Assemblierung werden die aufgetretenen Syntaxfehler angezeigt, danach erfolgt der Rücksprung ins Betriebssystem. Wird an das Kommando ASM der Suffix /L angehängt, so erzeugt der Assembler im zweiten Durchlauf ein Druckprotokoll. Es enthält den Quelltext, den erzeugten Code in sedezimaler Darstellung sowie die erzeugte Symboltabelle.

Der Assembler erkennt weitgehend die Syntax des Microsoft M80-Assemblers und kann die meisten für M80 geschriebenen Programme ungeändert assemblieren. Abweichend von der M80 Syntax erlaubt der Assembler mehrere durch Ausrufezeichen getrennte Anweisungen in einer Zeile und beliebig lange Symbolnamen. Umgekehrt können für den Christiani-Assembler geschriebene Programme ungeändert mit M80 assembliert werden, wenn sie nur eine Anweisung pro Zeile und keine langen Symbolnamen aufweisen.

Z80-Befehle

Der Assembler erkennt alle Z80-Befehle in der Zilog-Notation. Diese Befehle sind in einer Tabelle ab Seite 35 zusammengestellt.

Konstanten

Eine Konstante ist ein Wert in Zahlendarstellung. Konstanten müssen mit einer Ziffer (0 bis 9) beginnen, um als solche erkannt zu werden. Normalerweise werden Konstanten zur Basis Zehn angegeben. Mit dem Suffix B können Binärzahlen, mit dem Suffix O Oktalzahlen und mit dem Suffix H Sedezimalzahlen markiert werden. Zum Beispiel sind also die Konstanten 255, 3770, OFFH und llllllllB identisch.

Außerdem kann mit .Radix die Standard-Zahlenbasis geändert werden. Der Pseudobefehl .Radix 16 beispielsweise bewirkt, daß der Assembler alle folgenden Konstanten als Zahlen in sedezimaler Darstellung bewertet.

Der ASCII-Wert eines Zeichens kann ebenfalls als Konstante angegeben werden, es muß zwischen zwei Apostrophe gesetzt werden. Die Konstanten 41H und 'A' sind also identisch.

Strings, Zeichenketten

Ein String ist eine zwischen zwei Apostrophe gesetzte Zeichenkette. Soll im String ein Apostroph selbst vorkommen, sind zwei aufeinanderfolgende Apostrophe zu schreiben. Beispiel: Der String Sommer'81 ist also zu schreiben als 'Sommer '81'.

Strings werden in der folgenden Beschreibung mit /string/ bezeichnet.

Symbole

Symbole sind beliebig lange Zeichenketten, denen während der Assemblierung ein Wert zugewiesen wird, der dann bei späteren Referenzen für das Symbol eingesetzt wird. Symbole dürfen nicht mit einer Ziffer beginnen.

Symbolzuweisungen können durch die Pseudobefehle EQU oder SET oder durch Labeldefinitionen erfolgen.

Labels werden definiert, indem sie an den Anfang der Zeile geschrieben werden. Auf das Label kann ein Doppelpunkt folgen. Der Rest der Zeile kann leer sein.

Rechenausdrücke

In Rechenausdrücken können Konstanten, Strings (maximale Länge 2 Zeichen), Symbolreferenzen, das Zeichen \$ für den Inhalt des Programmzählers sowie die folgenden Rechenzeichen vorkommen:

()
 LOW,HIGH
 *,/,MOD,SHR,SHL
 -
 +,-
 NOT
 AND
 OR,XOR

Die Rechenoperationen sind in der Reihenfolge fallender Priorität angeordnet.

Rechenausdrücke werden in der folgenden Beschreibung mit /ausdruck/ bezeichnet. Rechenausdrücke, die einen Wert

zwischen -128 und +255 ergeben müssen, also Integerzahlen mit 7 Bit und einem Vorzeichenbit, werden mit /offset/ bezeichnet. Rechenausdrücke, die einen Wert zwischen 0 und + 255 ergeben müssen, also Integerzahlen mit 8 Bit ohne Vorzeichen, werden als /data/ bezeichnet. Für /data/ werden auch Ergebnisse zwischen 0 und -256 (=0) akzeptiert, was einer Interpretation des Ergebnisses als Zahl aus 8 Bit und Vorzeichen entspricht. Diese Nomenklatur gilt auch bei der Beschreibung des Testers.

Kommentare

Kommentare können in jeder Zeile alleine , nach einem Befehl oder einem Pseudobefehl stehen. Der Beginn eines Kommentars wird mit einem Semikolon (;) markiert.

Pseudobefehle

Der Assembler erkennt folgende Pseudobefehle und bearbeitet sie gleich wie der M80-Assembler:

CSEG

Mit CSEG (und DSEG) kann zwischen zwei Programmzählern hin- und hergeschaltet werden. Damit können z.B. Daten (im RAM) vom Programm (im ROM) getrennt plaziert werden, aber im Programmtext dennoch gemischt vorkommen.

CSEG setzt den Programmzähler auf das Code-Segment (Programm-Segment). Dies ist der ursprüngliche Zustand des Programmzählers.

DSEG

DSEG setzt den Programmzähler auf das Datensegment. Nach DSEG muß in der Regel ein ORG-Befehl folgen, da sonst das Datensegment bei 100H beginnt und sich eventuell mit dem Code-Segment überlappt.

ORG

ORG /ausdruck/ setzt den Programmzähler im gerade aktivierten Segment auf den in /ausdruck/ errechneten Wert. Alle Referenzen in /ausdruck/ müssen vor der ORG-Zeile definiert sein. Ohne den ORG-Befehl setzt der Assembler die Programmzähler zum Programmanfang auf 0100H.

EQU

symbol EQU /ausdruck/ weist dem Symbol symbol den in /ausdruck/ errechneten Wert fest zu. Eine erneute Definition des Symbols symbol führt zu einer Fehlermeldung.

ASET

symbol ASET /ausdruck/ weist dem Symbol symbol den in /ausdruck/ errechneten Wert zu. Das Symbol symbol kann an anderer Stelle neu definiert werden.

.RADIX

.RADIX /ausdruck/ Die Standard-Basis für Konstanten ist 10. Mit .RADIX kann diese auf jeden Wert zwischen 2 und 16 geändert werden. Die Basis für den /ausdruck/ in dem .RADIX Befehl ist immer 10.

DB Definiere Byte

```
DB /data/
DB /string/
DB /data/,/data/,/data/...
DB /data/,/string/,/data/,/string/...
```

Die Argumente des Befehls DB sind Rechenausdrücke oder Strings. Die Ergebnisse der Rechenausdrücke und die Zeichen der Strings werden nacheinander byteweise im Speicher abgelegt.

DC

DC /string/ speichert die Zeichen der Zeichenkette /string/ in aufeinanderfolgenden Speicherstellen wie DB, setzt aber das höchste Bit des letzten Zeichens der Kette auf Eins.

DW Definiere Wort

```
DW /ausdruck/
DW /ausdruck/,/ausdruck/...
```

Die Argumente des Befehls DW sind Rechenausdrücke. Die Ergebnisse der Rechenausdrücke werden als 2-Byte-Worte in

aufeinanderfolgenden Speicherstellen gespeichert, zuerst das untere, dann das obere Halbbyte.

DS Definiere Speicherplatz

```
DS /ausdruck/
DS /ausdruck/,/data/
```

DS erhöht den Programmzähler um den in /ausdruck/ angegebenen Wert und reserviert so Speicherplatz für Variable. Wird /data/ mit angegeben, werden die reservierten Speicherstellen mit dem Wert von /data/ gefüllt.

DS /ausdruck/ hat die gleiche Wirkung wie
ORG \$+/ausdruck/.

REPT

```
REPT /data/.
.
.
ENDM
```

Der REPT-Befehl führt zur /data/-fachen Assemblierung des zwischen REPT und ENDM stehenden Programmtextes.

.PHASE und .DEPHASE

```
.PHASE /ausdruck/
.
.
.DEPHASE
```

Der zwischen .PHASE und .DEPHASE stehende Programmtext wird so assembliert, daß er ab der in /ausdruck/ angegebenen Speicherstelle lauffähig ist. Der Programmcode muß vor der Ausführung an die Stelle /ausdruck/ geschoben werden.

IF, ELSE und ENDIF Bedingte Assemblierung

```
IF /ausdruck/
.
.
ENDIF
```

```
IF /ausdruck/
.
.
```

```
ELSE
.
.
ENDIF
```

Der nach IF stehende Rechenausdruck /ausdruck/ wird ausgewertet. Hat er als Ergebnis Null, dann wird der folgende Programmtext bis zum Befehl ELSE oder ENDIF nicht assembliert, der eventuell zwischen ELSE und ENDIF stehende Programmtext jedoch assembliert. Ist der Wert von /ausdruck/ ungleich Null, dann wird der bis ELSE oder ENDIF stehende Programmtext assembliert, der eventuell zwischen ELSE und ENDIF stehende jedoch nicht assembliert.

TITLE

Mit TITLE titelstring kann ein String eingegeben werden, der dann bei Druckausgabe auf dem Kopf jeder Seite ausgedruckt wird. Dieser String wird nicht zwischen Apostrophen gesetzt.

PAGE

PAGE

PAGE /data/

PAGE löst bei Druckausgabe den Beginn einer neuen Zeile aus. Wird /data/ mit angegeben, dann wird nach /data/ Zeilen eine neue Seite begonnen. Ist /data/ gleich Null, dann wird die Seitenformatierung ausgeschaltet.

END

END

END /ausdruck/

Mit END wird das Ende des Programms markiert. Nach END stehende Anweisungen werden nicht mehr assembliert. Wird /ausdruck/ mit angegeben, dann wird der Programmzähler im Tester auf den Wert von /ausdruck/ gesetzt.

Fehlermeldungen

Fehlerhafte Programmzeilen werden während der Assemblierung auf dem Bildschirm ausgegeben und in der ersten

Stelle mit einem Fehlercode markiert. Bei Assemblierung mit Druckausgabe geschieht dies in der Druckausgabe ebenfalls. Die Fehlercodes haben folgende Bedeutungen:

- L Programmzeile falsch, z.B. falscher Opcode.
- M Multiple, mehrfach definiertes Symbol (führt zu P-Error in Durchlauf 2).
- P Phase, unterschiedlicher Wert für Symbol bei Durchlauf 1 und 2.
- U Undefined, nicht definiertes Symbol.
- A Abstand zu groß bei Offsetangaben, relativem Sprung, /data/ zu groß.
- N falsche Ziffer in Konstante.
- O Overflow, Überlauf beim Errechnen eines Ausdrucks.
- I Illegale Adresse, Programmzähler zeigt auf reservierte Systemadresse.

Anwenderprogramm starten

Mit dem Befehl RUN kann ein bei Adresse 0100H beginnendes Anwenderprogramm gestartet werden.

Das Anwenderprogramm kann über die Adresse 0005H Systemfunktionen aufrufen und über die Adresse 0000H ins Betriebssystem zurückspringen.

Z80-Tester

Der Tester wird mit dem Kommando TEST aufgerufen und meldet sich auf dem Bildschirm mit einem Stern als Prompt-Zeichen. Die einzelnen Tester-Kommandos bestehen jeweils aus einer Buchstabenkennung und eventuell einem oder mehreren Parametern.

Als Parameter können komplexe Ausdrücke angegeben werden, also Konstanten, Symbole oder Berechnungen. Es können alle in einem Assemblerlauf definierten Symbole verwendet werden.

Jede Konstante muß mit einer Ziffer beginnen, um sie von den Symbolen zu unterscheiden. Die Standardzahlenbasis für Konstanten ist 16 (default). Sie kann aber mit den Kommandos DEC und HEX umgeschaltet werden.

Im Gegensatz zu DDT (Digital Research) muß zwischen der Kommandozeichnung und den Parametern ein Zwischenraum eingegeben werden.

Auf falsche Kommandos reagiert der Tester mit der Ausgabe eines Fragezeichens und einem Piepston.

Der Tester kann wieder verlassen werden mit der Eingabe von CTRL-C am Anfang einer Zeile. Die jeweils aufgerufenen Tester-Funktionen können durch das Betätigen beliebiger Nicht-Buchstabentasten abgebrochen werden. Wird eine Buchstabentaste betätigt, so wird diese gespeichert, es können also Kommandos im Voraus eingegeben werden.

Im Tester sind die nachfolgend beschriebenen Kommandos implementiert.

Sedezimale Eingabe

HEX

Mit diesem Kommando wird die Standard-Zahlenbasis für Konstanteneingaben auf 16 festgesetzt. Konstanten mit der Darstellung im Oktalsystem können mit dem Suffix O eingegeben werden.

Dezimale Eingabe

DEC

Die Standard-Zahlenbasis für Konstanteneingaben wird auf 10 festgesetzt. Mit dem Suffix B können Konstanten in binärer Darstellung eingegeben werden, mit dem Suffix O solche in oktaler Darstellung und mit dem Suffix H solche in sedezimaler Darstellung.

Rechnen

? /ausdruck/

? /ausdruck/,/ausdruck/,...

Der Wert eines oder mehrerer durch jeweils ein Komma getrennter Ausdrücke wird berechnet und als sedezimal dargestellte Zahl ausgegeben, zusammen mit den zugehörigen Fehlermeldungen für Overflow oder unbekanntes Symbol.

Dump

D
 D /ausdruck/
 D /ausdruck1/,/ausdruck2/

Der Inhalt eines Speicherbereichs in sedezimaler Darstellung und die entsprechenden ASCII-Zeichen werden ausgegeben. Die Größe des ausgegebenen Bereichs kann durch die Angabe von Anfangs- und Endadresse bestimmt werden.

List

L
 L /ausdruck/
 L /ausdruck1/,/ausdruck2/

Der Inhalt eines Speicherbereichs wird in sedezimaler Darstellung und disassembliert ausgegeben. Adressen und Werte, die in der Symboltabelle definiert sind, werden zusätzlich als Symbol ausgegeben. Anfangs- und Endadresse der Ausgaben können in zwei Rechenausdrücken angegeben werden.

Setze Speicher

S /ausdruck/

Der Inhalt der Speicherstelle mit der Adresse /ausdruck/ wird in sedezimaler Darstellung ausgegeben. Der Inhalt kann durch Eingabe eines Rechenausdrucks (/data/) gefolgt von RETURN modifiziert werden. RETURN alleine läßt den Inhalt unverändert, der Inhalt der folgenden Speicherzelle wird dargestellt. Durch Eingabe eines Punktes (.) wird der Vorgang abgebrochen.

Der zur Modifizierung angegebene Rechenausdruck muß kleiner als 256 sein, sonst erfolgt eine Overflow-Meldung mit A:? . Wird versucht, in eine geschützte Systemadresse zu schreiben, erfolgt eine Meldung mit I:? . Nicht lösbare Symbolreferenzen werden mit U:? angezeigt.

Fülle Speicher (Fill)

F /ausdruck1/,/ausdruck2/,/data/

Der Inhalt des Speicherbereichs von der Adresse /ausdruck1/ bis zur Adresse /ausdruck2/ (einschließlich) wird mit dem Wert /data/ gefüllt.

Der angegebene Wert /data/ muß kleiner als 256 sein, sonst erfolgt eine Overflow-Meldung mit A:? . Wird versucht, in eine geschützte Systemadresse zu schreiben, erfolgt eine entsprechende Meldung mit I:? . Nicht lösbare Symbolreferenzen werden mit U:? angezeigt.

Verschiebe Speicher (Move)

M /ausdruck1/,/ausdruck2/,/ausdruck3/

Der Inhalt des Speicherbereichs von der Adresse /ausdruck1/ bis zur Adresse /ausdruck2/ (einschließlich) wird zur in /ausdruck3/ angegebenen Adresse verschoben. Die Speicherbereiche dürfen sich überlappen.

Wird versucht, in eine geschützte Systemadresse zu schreiben, erfolgt eine entsprechende Meldung mit I:? . Nicht lösbare Symbolreferenzen werden mit U:? angezeigt.

Vergleiche Speicher (Compare)

CMP /ausdruck1/,/ausdruck2/,/ausdruck3/

Der Inhalt des Speicherbereichs von /ausdruck1/ bis /ausdruck2/ einschließlich wird mit dem ab /ausdruck3/ beginnenden Speicherbereich verglichen. Die Speicherbereiche dürfen sich überlappen. Es werden alle Adressen ausgegeben, an denen sich die Inhalte der beiden Speicherbereiche unterscheiden.

Port lesen

IN /data/

Der Inhalt des Ports mit der Adresse /data/ wird gelesen und als zweistellige Sedezimalzahl angezeigt.

Port setzen

out /data1/,/data2/

Der Port mit der Adresse /data1/ wird mit dem Wert /data2/ gesetzt.

Register ausgeben, setzen

X

Alle Register der Z80-CPU werden ausgegeben und der zum momentanen Befehlszählerstand gehörende Befehl disassembliert.

Als Parameter kann eine Registerbezeichnung angegeben werden. Es wird dann nur das angegebene Register ausgegeben und kann durch Eingabe eines Rechenausdrucks geändert werden. Gültige Registerbezeichnungen sind:

PC, SP, IX, IY, AF, BC, DE, HL, A, F, B, C, D, E, H, L, AF', BC', DE', HL', A', F', B', C', D', E', H', L', I', IFF

Einzelschritt (Trace)

T

Der zum momentanen Befehlszählerstand gehörende Befehl wird disassembliert und dann ausgeführt. Anschließend werden die sich ergebenden Registerinhalte angezeigt.

Mit T /ausdruck/ werden /ausdruck/ Einzelschritte ausgeführt. Der Ablauf kann mit einer Tasteneingabe unterbrochen werden. Ist /ausdruck/ gleich Null, werden Einzelschritte ausgeführt, bis eine Tasteneingabe erfolgt.

Systemaufrufe über die Adresse 0005H werden in Echtzeit ausgeführt.

Einzelschritt (Untrace)

U

U /ausdruck/

Wie bei Kommando T werden ein oder mehrere Einzelschritte ausgeführt. Die sich ergebenden Registerinhalte werden jedoch nicht angezeigt.

Programmstart

G

Das im Speicher stehende Programm wird ab dem momentanen Befehlszählerstand in Echtzeit ohne Überwachung gestartet. Die Register werden vor dem Start geladen.

G /ausdruck/

Das im Speicher stehende Programm wird ab der Adresse /ausdruck/ in Echtzeit ohne Überwachung gestartet. Die Register werden vor dem Start geladen.

G /ausdruck/,/ausdruckl/
 G /ausdruck/,/ausdruckl;/ausdruck2/
 G ,/ausdruckl/
 G ,/ausdruckl;/ausdruck2/

Mit ,/ausdruckl/ können bis zu 8 Unterbrechungspunkte gesetzt werden. Mit ,/ausdruckl;/ausdruck2/ gibt /ausdruck2/ die Anzahl der gewünschten Durchläufe an der Unterbrechungsstelle /ausdruckl/ bis zum Abbruch an. Beim Auffinden einer Unterbrechungsstelle werden die Registerinhalte der CPU angegeben. Mit G 0 kann der Tester anstatt mit CTRL-C ebenfalls verlassen werden (Sprung zur Warmstartprozedur).

Programmstart ohne Registeranzeige

GU

Das Kommando GU hat die gleiche Funktion wie G , jedoch werden beim Auffinden einer Unterbrechungsstelle die Registerinhalte der CPU nicht ausgegeben.

Userprogramm definieren

DEFUSR /ausdruck/

Ein bei der Adresse /ausdruck/ beginnendes Programm wird als Userfunktion deklariert.

Userprogramm aufrufen

USR
 USR /ausdruck/
 USR /ausdruck/,/ausdruck/,...

Die mit DEFUSR definierte Adresse wird aufgerufen und eine beliebige Anzahl Parameter übergeben. Die Anzahl der Parameter steht im Register BC, die Parameter im Stack. Enthält BC z.B. 3, sind mit drei POP drei Parameter vom Stack zu holen, anschließend mit RET Rücksprung in den Tester. Bei Error mit gesetztem Carry-Flag gibt der Tester ein ? aus. Der zuletzt eingegebene Parameter wird zuerst vom Stack geholt.

File lesen

RF
RF /ausdruck/

Ein Datenfile wird ab der Adresse 0100H bzw. ab der Adresse /ausdruck/ von Cassette oder Diskette in den Speicher geladen. Das System fragt nach einem File-Namen. Nach der Ausführung wird die nächste freie Adresse angezeigt.

File schreiben

SF /ausdruck1/,/ausdruck2/

Der Speicherbereich von /ausdruck1/ bis /ausdruck2/ einschließlich wird in einem Datenfile gespeichert. Das System fragt nach einem File-Namen.

Daten empfangen

RX
RX /ausdruck/

Über Telefonmodem werden Daten empfangen und in den Speicherbereich ab der Adresse 0100H bzw. Adresse /ausdruck/ geschrieben. Das System fragt nach Modem-Parametern (siehe Abschnitt "Telefonmodem-Betriebsprogramm"). Nach der Ausführung wird die nächste freie Adresse angezeigt.

Daten senden

TX /ausdruck1/,/ausdruck2/

Der Inhalt des Speichers ab Adresse /ausdruck1/ bis Adresse /ausdruck2/ einschließlich wird über Telefonmodem gesendet. Das System fragt nach Modem-Parametern (siehe Abschnitt "Telefonmodem-Betriebsprogramm").

EPROM lesen

READ /ausdruck1/,/ausdruck2/,/ausdruck3/

Der Inhalt des EPROMs im Programmiergerät wird ab EPROM-Adresse /ausdruck1/ bis EPROM-Adresse /ausdruck2/ in den RAM-Speicherbereich ab Adresse /ausdruck3/ gelesen.

EPROM programmieren

PROG /ausdruck1/,/ausdruck2/,/ausdruck3/

Der Inhalt des RAM-Speicherbereichs von Adresse /ausdruck1/ bis Adresse /ausdruck2/ wird in das EPROM im Programmiergerät ab EPROM-Adresse /ausdruck3/ geschrieben. Dabei laufen auf dem Bildschirm nacheinander die 16 Buchstaben von A bis P durch und zeigen an, daß jeweils 16 Bytes "gebrannt" wurden. Danach erscheint ein Pluszeichen, womit dokumentiert wird, daß 16 mal 16 = 256 Bytes im EPROM geschrieben stehen. Auf diese Weise zeigt das System dem Benutzer ähnlich wie beim Assemblieren, daß "etwas geschieht".

EPROM prüfen

VER /ausdruck1/,/ausdruck2/,/ausdruck3/

Der Inhalt des EPROMS im Programmiergerät wird ab Eprom-Adresse /ausdruck1/ bis EPROM-Adresse /ausdruck2/ mit dem RAM-Speicherbereich ab Adresse /ausdruck3/ verglichen.

Seitenformatierung

P
P /data/

Auf dem Drucker wird ein Seitenvorschub ausgelöst. Wird /data/ mit angegeben, so wird die Seitenlänge auf /data/ gesetzt. Ist /data/ gleich Null, wird die Seitenformatierung abgeschaltet.

Kommentare

;

Mit einem Semicolon können Kommentarzeilen markiert werden, die keine weitere Funktion haben.

Freigabe der Systemadressen

PROFI
PROFI ON
PROFI OFF

Mit PROF I ON können die reservierten Systemadressen für den Assembler und Tester freigegeben werden. Es werden

dann keine I-Fehler mehr angezeigt. Mit RPOFI OFF kann dieser Zustand wieder aufgehoben werden. Mit PROFI allein kann der Zustand des Funktions-Flags abgefragt werden. Dieses Kommando kann im Tester und in der Kommando-Ebene gegeben werden.

Telefonmodem-Betriebsprogramm

Das Telefonmodem-Betriebsprogramm wird mit MODEM aufgerufen und meldet sich mit der Raute (#) als Prompt-Zeichen. Es kennt folgende Kommandos:

COM Terminal (Communication)-Mode
Zeichen von der Tastatur werden gesendet, empfangene Zeichen auf den Bildschirm geschrieben. Mit CTRL-T kann der Inhalt des Textspeichers gesendet werden, mit CTRL-R kann der empfangene Text im Textspeicher abgelegt werden. Eingegebene CR (Carriage Return, Wagenrücklauf, Taste Ret) werden automatisch in CR-LF umgewandelt. Abbruch ist mit CTRL-E (Exit) und CTRL-C (Warmstart) möglich.

- E** Terminal-Mode mit Echo
Zeichen von der Tastatur werden gesendet und auf den Bildschirm geschrieben, empfangene Zeichen werden auf den Bildschirm geschrieben und zurückgeschickt. Mit CTRL-T kann der Inhalt des Textspeichers gesendet werden, mit CTRL-R kann der empfangene Text im Textspeicher abgelegt werden. Eingegebene CR (Carriage Return, Wagenrücklauf, Taste Ret) werden automatisch in CR-LF umgewandelt. Abbruch ist mit CTRL-E (Exit) und CTRL-C (Warmstart) möglich.
- T** Terminal-Mode
Auf der Tastatur eingegebene Zeichen werden gesendet, empfangene Zeichen auf den Bildschirm geschrieben. Alle Control-Codes werden ungeändert gesendet, so daß mit dieser Funktion ein anderer Rechner bedient werden kann. Die im Terminal-Mode vorhandenen Befehle (CTRL-T, CTRL-R, CTRL-E, CTRL-C) sind durch vorherige Eingabe eines ESCAPE, Taste ESC, erreichbar. Das ESC-Zeichen selbst kann durch Eingabe von ESC, ESC gesendet werden.
- S** Inhalt des Textspeichers senden mit Protokoll.
- R** Inhalt des Textspeichers empfangen mit Protokoll.
- P** Seitenformatierung (siehe "Seitenformatierung/Druckausgabe" und "Tester").
- ;** Kommentierung (siehe "Systemkommandos" und "Tester".)

Das Telefonmodem-Betriebsprogramm reagiert auf fehlerhafte Kommandos mit Ausgabe eines Fragezeichens und eines Piepstons. Es kann mit Eingabe von CTRL-C zu Beginn einer Zeile wieder verlassen werden.

Modem-Parameter

Ergänzend zu den Kommandos können folgende Modem-Parameter angegeben werden (das gilt auch bei RX und TX vom Tester aus):

Bei den Befehlen S (TX) und R (RX) kann das Übertragungsprotokoll mit dem Parameter 0 abgeschaltet werden. Es wird dann einfach Byte für Byte gesendet bzw. empfangen ohne Prüfungen.

Bei allen Befehlen können die Baudraten für Sender und Empfänger als Parameter angegeben werden. Die Zahlenangaben für die Baudrate werden dezimal interpretiert. Beispiele:

T	Terminal-Mode, 300 Baud
E 1200	Echo-Mode, 1200 Baud
E 1200,75	Echo-Mode, 1200 Baud senden, 75 Baud empfangen.
S	Textspeicher senden
R 0 75 1200	Textspeicher ohne Protokoll mit 1200 Baud empfangen

Bei Angabe einer dritten Baudrate programmiert diese die zweite serielle Schnittstelle.

Ein Abbruch ist bei den Funktionen T und E mit CTRL-E und CTRL-C (Warmstart) möglich. Bei den Funktionen S und R kann mit CTRL-X oder CTRL-C abgebrochen werden. Bei Überfüllung des Textspeichers wird automatisch abgebrochen.

Das Übertragungsprotokoll des Telefonmodem-Betriebsprogramms ist voll kompatibel zu dem des Modemprogramms der CP/M-User-Group und kann daher mit diesem zusammenarbeiten.

Systemfunktionen

Das Betriebssystem simuliert über die Einsprungsadresse 0005H die BDOS-Funktionen 0 bis 12 von CP/M. Zusätzlich stellt es erweiterte Systemfunktionen mit Nummern ab 240 (OFOH) zur Verfügung.

Ein Warmstart des Betriebssystems kann mit JP 0 oder Systemfunktion 0 ausgelöst werden.

Der bei der Adresse 0005H programmierte Sprung geht wie bei CP/M zur ersten vom System belegten Adresse und wird je nach Text- und Symbolspeichergröße geändert.

Anwenderprogramme können also jeweils den Speicher bis zu der in Adresse 0006 und 0007 angegebenen Adresse benutzen oder z.B. ihren Stackpointer auf diese Adresse setzen.

Die Systemfunktionen sind identisch zu den BDOS-Funktionen von CP/M (siehe CP/M-Interface-Guide) mit folgender Ausnahme:

Das Löschen von einzelnen Zeichen auf dem Bildschirm bei Funktion 10 ist mit Backspace (Bs) und Delete (Del) möglich. Die Kommandos CTRL-R und CTRL-U bei Funktion 10 sind nicht vorgesehen.

Standard-Systemfunktionen

Funktion 0 - Warmstart

Parameter bei Aufruf:
Register C: 00H

Parameter beim Rücksprung:
keine, kein Rücksprung

Die Warmstartfunktion hat die gleiche Wirkung wie ein Sprung zur Adresse 0.

Funktion 1 - Konsoleingabe

Parameter beim Aufruf:
Register C: 01H

Parameter beim Rücksprung:
Register A: ASCII-Zeichen

Die Konsoleingabefunktion liest das nächste Zeichen von der Tastatur und zeigt es auf dem Bildschirm an. Tab's werden ausgeführt. Nicht darstellbare Control-Codes werden mit einem vorgestellten Pfeil angezeigt. Mit CTRL-S bzw. CTRL-P kann die Ausgabe angehalten bzw. fortgesetzt werden oder die Druckerprotokollierung an- bzw. abgeschaltet werden. Das Programm kehrt erst zurück, nachdem ein Zeichen eingegeben wurde.

Funktion 2 - Konsolausgabe

Parameter beim Aufruf:
Register C: 02H
Register E: ASCII-Zeichen

Parameter beim Rücksprung:
keine

Das ASCII-Zeichen im Register E wird auf dem Bildschirm angezeigt. Mit CTRL-S bzw. CTRL-P kann die Ausgabe angehalten bzw. fortgesetzt werden oder die Druckerprotokollierung an- bzw. abgeschaltet werden.

Funktion 3 - Lochstreifenleser, Eingabe

Parameter beim Aufruf:
Register C: 03H

Parameter beim Rücksprung:
 Register A: ASCII-Zeichen

Das nächste Zeichen vom Lochstreifenleser wird gelesen. Da kein LSL installiert ist, wird LAH (End of File) eingelesen.

Funktion 4 - Lochstreifenstanzer, Ausgabe

Parameter beim Aufruf:
 Register C: 04H
 Register E: ASCII-Zeichen

Parameter beim Rücksprung:
 keine

Das Zeichen im Register E wird zum Lochstreifenstanzer geschickt. Da kein LSS installiert ist, geschieht nichts.

Funktion 5 - Drucker, Ausgabe

Parameter beim Aufruf:
 Register C: 05H
 Register E: ASCII-Zeichen

Parameter beim Rücksprung:
 keine

Das Zeichen im Register E wird zum Drucker geschickt.

Funktion 6 - Direkte Konsol- Ein- und Ausgabe

Parameter beim Aufruf:
 Register C: 06H
 Register E: OFFH (Eingabe) oder
 ASCII-Zeichen (Ausgabe)

Parameter beim Rücksprung:
 Register A: ASCII-Zeichen oder Status

Die direkte Konsol- Ein- und Ausgabe unterscheidet sich von den Konsolfunktionen 1 und 2 dadurch, daß eingegebene Zeichen nicht auf dem Bildschirm angezeigt werden, Tab's nicht ausgeführt werden, und die Kommandos CTRL-S (Start/Stop) und CTRL-P (Druckerprotokoll) nicht beachtet werden.

Beim Aufruf der Funktion 6 enthält das Register E entweder OFFH oder ein ASCII-Zeichen, das dann (Ausgabe) auf dem Bildschirm angezeigt wird. Im ersten Fall wird im Re-

gister A ein eingegebenes ASCII-Zeichen übergeben (oder 00H, falls kein Zeichen eingegeben wurde).

Funktion 7 - Hole I/O-Byte

Parameter beim Aufruf:

Register C: 07H

Parameter beim Rücksprung:

Register A: I/O-Byte Wert

Funktion 7 holt den Wert des I/O-Byte in das Register A.

Funktion 8 - Setze I/O-Byte

Parameter beim Aufruf:

Register C: 08H

Register E: I/O-Byte Wert

Parameter beim Rücksprung:

keine

Funktion 8 setzt das I/O-Byte auf den im Register E übergebenen Wert.

Funktion 9 - Drucke String

Parameter beim Aufruf:

Register C: 09H

Registerpaar DE: String Anfangsadresse

Parameter beim Rücksprung:

keine

Funktion 9 gibt den ab der Adresse im Registerpaar DE beginnenden String auf dem Bildschirm aus, bis ein \$ im String vorkommt. Tab's und CTRL-S bzw. CTRL-P werden wie in Funktion 2 behandelt.

Funktion 10 - Lies Konsolpuffer

Parameter beim Aufruf:

Register C: 0AH

Registerpaar DE: Pufferadresse
Puffergröße (mx, siehe unten)

Parameter beim Rücksprung:

Zeichen im Puffer (siehe nächste Seite)

```
DE: +0 +1 +2 +3 +4 ... +n
     mx nc c1 c2 c3 ... cn
```

mx = maximale Pufferlänge
nc = Anzahl der eingegebenen Zeichen
c1 bis cn = ASCII-Zeichen

Funktion 10 liest einen String von der Tastatur und legt die eingegebenen Zeichen in einem Puffer ab. Dabei werden Tab's und CTRL-S bzw. CTRL-P wie in Funktion 2 behandelt. Fehlerhafte Zeichen können einzeln mit DELETE oder RUBOUT gelöscht werden. Die gesamte Eingabezeile kann mit CTRL-X gelöscht werden. Am Beginn der Zeile ist mit CTRL-C ein Warmstart möglich. Die Eingabe wird mit RETURN oder LINE-FEED oder bei vollem Puffer beendet.

Funktion 11 - Hole Konsolstatus

Parameter beim Aufruf:
Register C: OBH

Parameter beim Rücksprung:
Register A: Status

Funktion 11 prüft, ob an der Konsole (Tastatur) ein Zeichen eingegeben wurde. Ist dies der Fall, wird im Register A OFFH übergeben, andernfalls 00H.

Funktion 12 - Hole Versionsnummer

Parameter beim Aufruf:
Register C:0CH

Parameter beim Rücksprung:
Registerpaar HL: Versionsnummer

Funktion 12 übergibt im Registerpaar HL die Versionsnummer des eingesetzten Betriebssystems. Arbeitet TLC90 allein, wird 000H übergeben. Arbeitet TLC90 unter CP/M, wird die entsprechende CP/M-Versionsnummer (ab 2.0) im Register L übergeben (in Register H steht 00 für CP/M, 01 für MP/M).

Erweiterte Systemfunktionen

Das Christiani-Betriebssystem stellt über die Einsprungsadresse 0005H einige zusätzliche Funktionen zur Verfügung, mit deren Hilfe Anwenderprogramme auf die Rechen- und String-Verarbeitungsprogramme des Betriebssystems zurückgreifen und auf den Textspeicher zugreifen können.

Funktion 240 - Print Byte

Parameter beim Aufruf:

Register C: OF0H

Register E: auszugebendes Byte

Parameter beim Rücksprung:

keine

Funktion 240 gibt den Inhalt des Registers E als zwei-stellige Sedezimalzahl auf der Konsole (Bildschirm) aus. Mit CTRL-S bzw. CTRL-P kann die Ausgabe angehalten bzw. fortgesetzt werden oder die Druckerprotokollierung an-bzw. abgeschaltet werden.

Funktion 241 - Print Word

Parameter beim Aufruf:

Register C: OF1H

Registerpaar DE: auszugebendes Wort

Parameter beim Rücksprung:

keine

Funktion 241 gibt den Inhalt des Registerpaars DE als vierstellige Sedezimalzahl auf der Konsole (Bildschirm) aus. CTRL-S bzw. CTRL-P werden wie in Funktion 240 behan-delt.

Funktion 242 - Setze Zahlenbasis

Parameter beim Aufruf:

Register C: OF2H

Register E: Zahlenbasis

Parameter beim Rücksprung:

keine

Funktion 242 setzt die Standard-Zahlenbasis für Funktion 243 auf den im Register E angegebenen Wert (2 bis 16).

Funktion 243 - Rechne Ausdruck

Parameter beim Aufruf:

Register C: OF3H

Registerpaar DE: Anfangsadresse Ausdruck

Parameter beim Rücksprung:

Registerpaar HL: Ergebnis
Carry-Flag : Ausdruck vorhanden ja/nein

Funktion 243 errechnet den Wert eines im Speicher als ASCII-String abgelegten Rechenausdrucks, der den Konventionen für Rechenausdrücke im Assembler entspricht. Die Anfangsadresse des Ausdrucks wird im Registerpaar DE übergeben. Der Ausdrucks-String muß mit einem Komma oder RETURN (ODH) abgeschlossen sein. Das Ergebnis des ausgewerteten Ausdrucks wird im Registerpaar HL übergeben, zusammen mit dem gesetzten Carry-Flag. Das nicht gesetzte Carry-Flag bedeutet, daß kein Ausdruck vorgefunden wurde.

Funktion 244 - Print Error

Parameter beim Aufruf:

Register C: 0F4H

Parameter beim Rücksprung:

Carry-Flag : Error ja/nein

Funktion 244 gibt einen bei Funktion 243 aufgetretenen Fehler mit dem entsprechenden Fehlercode (Overflow, unbekanntes Symbol etc.) gefolgt von einem Doppelpunkt auf dem Bildschirm aus und setzt das Carry-Flag. Ist beim letzten Aufruf von Funktion 243 kein Fehler aufgetreten, erfolgt keine Ausgabe und das Carry-Flag wird nicht gesetzt.

Funktion 245 - Vergleiche String

Parameter beim Aufruf:

Register C: 0F5H

Registerpaar DE: Anfangsadresse String

Parameter beim Rücksprung:

Registerpaar DE: Anfangs/Endadresse String

Carry-Flag : Strings identisch ja/nein

Funktion 245 vergleicht einen im Speicher ab Adresse (DE) abgelegten String mit dem String, der direkt nach dem Call-Befehl für Funktion 245 im Speicher steht und dessen Ende mit Bit 7 =1 markiert ist. Das Programm wird an der darauffolgenden Speicherstelle fortgesetzt. Wenn beide Strings identisch sind, wird das Carry-Flag gesetzt und (DE) zeigt auf das Ende des im Speicher stehenden Strings. Andernfalls wird das Carry-Bit gelöscht und (DE) zeigt auf den Anfang des verglichenen Strings. Groß- und Kleinbuchstaben werden als identisch behandelt.

Funktion 246 - Vergleiche String mit Tabelle

Parameter beim Aufruf:

Register C: 0F6H
 Registerpaar DE: Anfangsadresse String
 Register IX: Anfangsadresse Tabelle

Parameter beim Rücksprung:

Register A: Stringnummer
 Registerpaar DE: Anfangs/Endadresse String
 Carry-Flag : String in Tabelle gefunden
 ja/nein

Funktion 246 vergleicht einen im Speicher ab Adresse (DE) abgelegten String mit einer Tabelle von Strings. Das Register IX zeigt auf den Anfang der Tabelle, die aus direkt aufeinanderfolgenden Strings besteht. Das Ende jedes Strings innerehalb der Tabelle ist mit Bit 7 =1 markiert, das Ende der Tabelle mit einem Stop-Byte (=0FFH). Falls der durch das Registerpaar DE adressierte String mit einem String innerhalb der Tabelle übereinstimmt, wird das Carry-Flag gesetzt, das Register A enthält die Nummer des Strings in der Tabelle und das Registerpaar DE zeigt auf das Ende des Strings. Stimmt der String mit keinem Element der Tabelle überein, wird das Carry-Flag gelöscht und das Registerpaar DE zeigt weiterhin auf den Anfang des untersuchten Strings. Groß- und Kleinbuchstaben werden als identisch behandelt. Die Tabelle muß jedoch in Großbuchstaben geschrieben sein.

Funktion 247 - Hole Textanfang

Parameter beim Aufruf:

Register C: 0F7H

Parameter beim Rücksprung:

Registerpaar HL: Textanfang

Funktion 247 lädt die momentane Anfangsadresse des Textspeichers ins Registerpaar HL.

Funktion 248 - Hole Textende

Parameter beim Aufruf:

Register C: 0F8H

Parameter beim Rücksprung:

Registerpaar HL: Textende

Funktion 248 lädt die momentane Endadresse des Textspeichers ins Registerpaar HL.

Funktion 249 - Hole Speicherende

Parameter beim Aufruf:

Register C: 0F9H

Parameter beim Rücksprung:

Registerpaar HL: max. Textspeicher

Funktion 249 lädt die größte für den Textspeicher zur Verfügung stehende Adresse ins Registerpaar HL. Das Ergebnis der Funktion 249 minus dem Ergebnis der Funktion 248 ergibt die Anzahl der noch freien Bytes für den Textspeicher.

Funktion 250 - Erzeuge File-Control-Block

Parameter beim Aufruf:

Register C: OFAH

Registerpaar DE: Pointer zur Filebezeichnung in Puffer

Registerpaar IX: Pointer zu File-Control-Block

Registerpaar IY: Pointer zu Anfangsadresse Puffer.

Parameter beim Rücksprung:

Carry-Flag: Filebezeichnung vorhanden ja/nein

FCB : File-Control-Block

Funktion 250 erzeugt aus einer in einem korrekt gesetzten Textpuffer (siehe Funktion 10) stehenden Filebezeichnung (z.B.: B:T*.COM) einen File-Control-Block für Diskettenoperationen unter CP/M. Hierfür wird die Adresse des Textpuffers im IY-Register übergeben. Die Stelle der Filebezeichnung im Puffer wird im DE-Registerpaar übergeben. Die Adresse, an der der FCB (File-Control-Block) erzeugt werden soll, wird im IX-Register übergeben. Wird eine gültige Filebezeichnung gefunden, so wird der entsprechende FCB erzeugt, das DE-Registerpaar zeigt auf das Ende der Filebezeichnung und das Carry-Flag ist bei der Rückkehr gesetzt. Andernfalls ist das Carry-Flag gelöscht (die Filebezeichnung wird von der Funktion unter Umständen im Textpuffer verschoben).

Die Funktion 250 erzeugt nur die ersten 12 Byte des FCB, also die Laufwerksbezeichnung und den Namen, ein Teil der restlichen Bytes müssen vor einigen Diskettenfunktionen gelöscht werden (siehe CP/M-Interface-Guide).

Z80-Befehle

Die nachfolgende Auflistung enthält alle Z80-Befehle in der Zilog-Notation. Diese Befehle erkennt der Christiani-Assembler.

Die Klammern bezeichnen keine fakultativen Parameter, sondern weisen in gewohnter Weise auf den Inhalt eines Registers oder einer Speicherzelle hin und sind in jedem Fall mit anzugeben.

ADC A,(HL)	AND (HL)
ADC A,(IX+/offset/)	AND (IX+/offset)
ADC A,(IY+/offset/)	AND (IY+/offset)
ADC A,A	AND A
ADC A,B	AND B
ADC A,C	AND C
ADC A,D	AND D
ADC A,E	AND E
ADC A,H	AND H
ADC A,L	AND L
ADC A,/data/	AND /data/
ADC HL,BC	BIT 0,(HL)
ADC HL,DE	BIT 0,(IX+/offset)
ADC HL,HL	BIT 0,(IY+/offset)
ADC HL,SP	BIT 0,A
	BIT 0,B
ADD A,(HL)	BIT 0,C
ADD A,(IX+/offset/)	BIT 0,D
ADD A,(IY+/offset/)	BIT 0,E
ADD A,A	BIT 0,H
ADD A,B	BIT 0,L
ADD A,C	
ADD A,D	BIT 1,(HL)
ADD A,E	BIT 1,(IX+/offset/)
ADD A,H	BIT 1,(IY+/offset)
ADD A,L	BIT 1,A
ADD A,/data/	BIT 1,B
	BIT 1,C
ADD HL,BC	BIT 1,D
ADD HL,DE	BIT 1,E
ADD HL,HL	BIT 1,H
ADD HL,SP	BIT 1,L
ADD IX,BC	
ADD IX,DE	
ADD IX,IX	
ADD IX,SP	
ADD IY,BC	
ADD IY,DE	
ADD IY,IY	
ADD IY,SP	

BIT 2,(HL)
 BIT 2,(IX+/offset/)
 BIT 2,(IY+/offset)
 BIT 2,A
 BIT 2,B
 BIT 2,C
 BIT 2,D
 BIT 2,E
 BIT 2,H
 BIT 2,L

BIT 3,(HL)
 BIT 3,(IX+/offset/)
 BIT 3,(IY+/offset)
 BIT 3,A
 BIT 3,B
 BIT 3,C
 BIT 3,D
 BIT 3,E
 BIT 3,H
 BIT 3,L

BIT 4,(HL)
 BIT 4,(IX+/offset/)
 BIT 4,(IY+/offset)
 BIT 4,A
 BIT 4,B
 BIT 4,C
 BIT 4,D
 BIT 4,E
 BIT 4,H
 BIT 4,L

BIT 5,(HL)
 BIT 5,(IX+/offset/)
 BIT 5,(IY+/offset)
 BIT 5,A
 BIT 5,B
 BIT 5,C
 BIT 5,D
 BIT 5,E
 BIT 5,H
 BIT 5,L

BIT 6,(HL)
 BIT 6,(IX+/offset/)
 BIT 6,(IY+/offset)
 BIT 6,A
 BIT 6,B
 BIT 6,C
 BIT 6,D
 BIT 6,E
 BIT 6,H
 BIT 6,L

BIT 7,(HL)
 BIT 7,(IX+/offset/)
 BIT 7,(IY+/offset)
 BIT 7,A
 BIT 7,B
 BIT 7,C
 BIT 7,E
 BIT 7,H
 BIT 7,L

CALL C, /ausdruck/
 CALL M, /ausdruck/
 CALL NC, /ausdruck/
 CALL /ausdruck/
 CALL NZ, /ausdruck/
 CALL P, /ausdruck/
 CALL PE, /ausdruck/
 CALL PO, /ausdruck/
 CALL Z, /ausdruck/

CCF

CP (HL)
 CP (IX+/offset/)
 CP (IY+/offset/)
 CP A
 CP B
 CP C
 CP D
 CP E
 CP H
 CP L
 CP /data/

CPD
 CPDR

CPI
 COIR

CPL

DAA

DEC (HL)	INC HL
DEC (IX+/offset/)	INC IX
DEC (IY+/offset/)	INC IY
DEC A	INC L
DEC B	INC SP
DEC BC	
DEC C	IND
DEC D	INDR
DEC DE	
DEC E	INI
DEC H	INIR
DEC HL	
DEC IX	JP (HL)
DEC IY	JP (IX)
DEC L	JP (IY)
DEC SP	JP M,/ausdruck/
	JP NC,/ausdruck/
DI	JP /ausdruck/
DJNZ /offset/	JP NZ,/ausdruck/
EI	JP P,/ausdruck/
	JP PE,/ausdruck/
EX (SP),HL	JP PO,/ausdruck/
EX (SP),IX	JP Z,/ausdruck/
EX (SP),IY	
EX AF,AF'	JR C,/offset/
EX DE,HL	JR /offset/
EXX	JR NC,/offset/
	JR NZ,/offset/
HALT	JR Z,/offset/
IM 0	LD (BC),A
IM 1	LD (DE),A
IM 2	LD (HL),A
	LD (HL),B
IN A,(C)	LD (HL),C
IN A,(/data/)	LD (HL),D
IN B,(C)	LD (HL),E
IN C,(C)	LD (HL),H
IN D,(C)	LD (HL),L
IN E,(C)	LD (HL),/data/
IN H,(C)	LD (IX+/offset/),A
IN L,(C)	LD (IX+/offset/),B
	LD (IX+/offset/),C
INC (HL)	LD (IX+/offset/),D
INC (IX+/offset/)	LD (IX+/offset/),E
INC (IY+/offset/)	LD (IX+/offset/),H
INC A	LD (IX+/offset/),L
INC B	LD (IX+/offset/),/data/
INC B	LD (IY+/offset/),A
INC C	LD (IY+/offset/),B
INC D	LD (IY+/offset/),C
INC DE	LD (IY+/offset/),D
INC E	LD (IY+/offset/),E
INC H	LD (IY+/offset/),H
	LD (IY+/offset/),L
	LD (IY+/offset/),/data/

LD	(/ausdruck/),A	LD	D,C
LD	(/ausdruck/),BC	LD	D,D
LD	(/ausdruck/),DE	LD	D,E
LD	(/ausdruck/),HL	LD	D,H
LD	(/ausdruck/),IX	LD	D,L
LD	(/ausdruck/),IY	LD	D,/data/
LD	(/ausdruck/),SP		
		LD	DE,(/ausdruck/)
LD	A,(BC)	LD	DE,/ausdruck/
LD	A,(DE)		
LD	A,(HL)	LD	E,(HL)
LD	A,(IX+/offset/)	LD	E,(IX+/offset/)
LD	A,(IY+/offset/)	LD	E,(IY+/offset/)
LD	A,(/ausdruck/)	LD	E,A
LD	A,A	LD	E,B
LD	A,B	LD	E,C
LD	A,C	LD	E,D
LD	A,D	LD	E,E
LD	A,E	LD	E,H
LD	A,H	LD	E,L
LD	A,I	LD	E,/data/
LD	A,L		
LD	A,/data/	LD	H,(HL)
		LD	H,(IX+/offset/)
LD	B,(HL)	LD	H,(IY+/offset/)
LD	B,(IX+/offset/)	LD	H,A
LD	B,(IY+/offset/)	LD	H,B
LD	B,A	LD	H,C
LD	B,B	LD	H,D
LD	B,C	LD	H,E
LD	B,D	LD	H,H
LD	B,E	LD	H,L
LD	B,H	LD	H,/data/
LD	B,L		
LD	B,/data/	LD	HL,(/ausdruck/)
		LD	HL,/ausdruck/
LD	BC,(/ausdruck/)		
LD	BC,/ausdruck/	LD	I,A
LD	C,(HL)	LD	IX,(/ausdruck/)
LD	C,(IX+/offset/)	LD	IX,/ausdruck/
LD	C,(IY+/offset/)		
LD	C,A	LD	IY,(/ausdruck/)
LD	C,B	LD	IY,/ausdruck/
LD	C,C		
LD	C,D	LD	L,(HL)
LD	C,E	LD	L,(IX+/offset/)
LD	C,H	LD	L,(IY+/offset/)
LD	C,L	LD	L,A
LD	C,/data/	LD	L,B
		LD	L,C
LD	D,(HL)	LD	L,D
LD	D,(IX+/offset/)	LD	L,E
LD	D,(IY+/offset/)	LD	L,H
LD	D,A	LD	L,L
LD	D,B	LD	L,/data/

LD SP, (/ausdruck/)	RES 0, (HL)
LD SP, HL	RES 0, (IX+/offset/)
LD SP, IX	RES 0, (IY+/offset/)
LD SP, IY	RES 0, A
LD SP, /ausdruck/	RES 0, B
	RES 0, C
LDD	RES 0, D
LDDR	RES 0, E
	RES 0, H
	RES 0, L
LDI	
LDIR	RES 1, (HL)
	RES 1, (IX+/offset/)
NEG	RES 1, (IY+/offset/)
NOP	RES 1, A
	RES 1, B
OR (HL)	RES 1, C
OR (IX+/offset/)	RES 1, D
OR (IY+/offset/)	RES 1, E
OR A	RES 1, H
OR B	RES 1, L
OR C	
OR D	RES 2, (HL)
OR E	RES 2, (IX+/offset/)
OR H	RES 2, (IY+/offset/)
OR L	RES 2, A
OR /data/	RES 2, B
	RES 2, C
OTDR	RES 2, D
OTIR	RES 2, E
	RES 2, H
	RES 2, L
OUT (C), A	RES 3, (HL)
OUT (C), B	RES 3, (IX+/offset/)
OUT (C), D	RES 3, (IY+/offset/)
OUT (C), E	RES 3, A
OUT (C), H	RES 3, B
OUT (C), L	RES 3, C
OUT (/data/), A	RES 3, D
	RES 3, E
OUTD	RES 3, H
OUTI	RES 3, L
POP AF	RES 4, (HL)
POP BC	RES 4, (IX+/offset/)
POP DE	RES 4, (IY+/offset/)
POP HL	RES 4, A
POP IX	RES 4, B
POP IY	RES 4, C
	RES 4, D
PUSH AF	RES 4, E
PUSH BC	RES 4, H
PUSH DE	RES 4, L
PUSH HL	
PUSH IX	
PUSH IY	

RES 5,(HL)	RLA
RES 5,(IX+/offset/)	RLC (HL)
RES 5,(IY+/offset/)	RLC (IX+/offset/)
RES 5,A	RLC (IY+/offset/)
RES 5,B	RLC A
RES 5,C	RLC B
RES 5,D	RLC D
RES 5,E	RLC E
RES 5,H	RLC H
RES 5,L	RLC L
RES 6,(HL)	RLCA
RES 6,(IX+/offset/)	RLD
RES 6,(IY+/offset/)	RR (HL)
RES 6,A	RR (IX+/offset/)
RES 6,B	RR (IY+/offset)
RES 6,C	RR A
RES 6,D	RR B
RES 6,E	RR C
RES 6,H	RR D
RES 6,L	RR E
RES 7,(HL)	RR H
RES 7,(IX+/offset/)	RR L
RES 7,(IY+/offset/)	RRA
RES 7,A	RRC (HL)
RES 7,B	RRC (IX+/offset/)
RES 7,C	RRC (IY+/offset)
RES 7,D	RRC A
RES 7,E	RRC B
RES 7,H	RRC C
RES 7,L	RRC D
RET	RRC E
RET C	RRC H
RET M	RRC L
RET NC	RRCA
RET NZ	RRD
RET P	RST 0
RET PE	RST 10H
RET PO	RST 18H
RET Z	RST 20H
RETI	RST 28H
RETN	RST 30H
RL (HL)	RST 38H
RL (IX+/offset/)	RST 8
RL (IY+/offset/)	
RL A	
RL B	
RL C	
RL D	
RL E	
RL H	
RL L	

SBC A,(HL)
 SBC A,(IX+/offset/)
 SBC A,(IY+/offset/)
 SBC A,A
 SBC A,B
 SBC A,C
 SBC A,D
 SBC A,E
 SBC A,H
 SBC A,L
 SBC HL,BC
 SBC HL,DE
 SBC HL,HL
 SBC HL,SP

SET 0,(HL)
 SET 0,(IX+/offset/)
 SET 0,(IY+/offset/)
 SET 0,A
 SET 0,B
 SET 0,C
 SET 0,D
 SET 0,E
 SET 0,H
 SET 0,L

SET 1,(HL)
 SET 1,(IX+/offset/)
 SET 1,(IY+/offset/)
 SET 1,A
 SET 1,B
 SET 1,C
 SET 1,D
 SET 1,E
 SET 1,H
 SET 1,L

SET 2,(HL)
 SET 2,(IX+/offset/)
 SET 2,(IY+/offset/)
 SET 2,A
 SET 2,B
 SET 2,C
 SET 2,D
 SET 2,E
 SET 2,H
 SET 2,L

SET 3,(HL)
 SET 3,(IX+/offset/)
 SET 3,(IY+/offset/)
 SET 3,A
 SET 3,B
 SET 3,C
 SET 3,D
 SET 3,E

SET 3,H
 SET 3,L

SET 4,(HL)
 SET 4,(IX+/offset/)
 SET 4,A
 SET 4,B
 SET 4,C
 SET 4,D
 SET 4,E
 SET 4,H
 SET 4,L

SET 5,(HL)
 SET 5,(IX+/offset/)
 SET 5,(IY+/offset/)
 SET 5,A
 SET 5,B
 SET 5,C
 SET 5,D
 SET 5,E
 SET 5,H
 SET 5,L

SET 6,(HL)
 SET 6,(IX+/offset/)
 SET 6,(IY+/offset/)
 SET 6,A
 SET 6,B
 SET 6,C
 SET 6,D
 SET 6,E
 SET 6,H
 SET 6,L

SET 7,(HL)
 SET 7,(IX+/offset/)
 SET 7,(IY+/offset/)
 SET 7,A
 SET 7,B
 SET 7,C
 SET 7,D
 SET 7,E
 SET 7,H
 SET 7,L

SLA (HL)
 SLA (IX+/offset/)
 SLA (IY+/offset/)
 SLA A
 SLA B
 SLA C
 SLA D
 SLA E
 SLA H
 SLA L

SRA (HL)	SUB (HL)
SRA (IY+/offset/)	SUB (IX+/offset/)
SRA A	SUB (IY+/offset/)
SRA B	SUB A
SRA C	SUB B
SRA D	SUB C
SRA E	SUB D
SRA H	SUB E
SRA L	SUB H
	SUB L
SRL (HL)	SUB /data/
SRL (IX+/offset/)	
SRL (IY+/offset/)	XOR (HL)
SRL A	XOR (IX+/offset/)
SRL B	XOR (IY+/offset/)
SRL C	XOR A
SRL D	XOR B
SRL E	XOR C
SRL H	XOR D
SRL L	XOR E
	XOR H
	XOR L
	XOR /data/

In den Bit-Set/Reset/Test Befehlen kann auch die Bitnummer als Ausdruck (Symbol) angegeben werden

In den Restart-Befehlen kann die Restart-Adresse als Ausdruck angegeben werden.

In den JR Befehlen und im DJNZ Befehl wird die Zieladresse angegeben. Der Assembler errechnet daraus das benötigte /offset/. In den Befehlen bezüglich der IX- und IY- Pointer ist dagegen das benötigte /offset/ direkt anzugeben.