

**NCR**

---

NCR DECISION MATE V

# Bedienungsanleitung

DRI PRODUCT	:	CP/M(R)-80
NCR VERSION NO.	:	2.2 (03.00)
PRODUCT NO.	:	D006-0224-0600
NCR PART NO.	:	D17-0033718
SERIAL NO.	:	CP2-761-51801

CP/M ist ein eingetragenes Warenzeichen von Digital Research, Inc.  
MS-DOS ist ein Warenzeichen der Microsoft Corporation.  
Z80A ist ein eingetragenes Warenzeichen von Zilog, Inc.  
UCSD p-System ist ein Warenzeichen der Regents of the University of California, San Diego.  
p-System ist ein Warenzeichen von SofTech Microsystems, Inc.

Bestellnummer 017-0032307

Copyright ©1983 by NCR Corporation  
Dayton, Ohio  
All Rights Reserved  
Printed in the Federal Republic of Germany

### **3. Auflage, Januar 1984**

NCR ist ständig bemüht, die Produkte im Zuge der Entwicklung von Technologie, Bauteilen, Soft- und Firmware dem neuesten Stand anzupassen. NCR behält sich deshalb das Recht vor, Spezifikationen ohne vorherige Ankündigung zu ändern.

Nicht alle hier beschriebenen Leistungen werden von NCR in allen Teilen der Welt vertrieben. Nähere Informationen bezüglich eventueller Einschränkungen oder auch Erweiterungen sowie den aktuellen Stand erfahren Sie von Ihrem Händler oder der nächstgelegenen NCR-Geschäftsstelle.

## VORWORT

Herzlichen Glückwunsch zu Ihrer Wahl des NCR DECISION MATE V. Decision Mate heißt soviel wie Entscheidungs-Helfer. Sie haben hiermit schon die erste wichtige Entscheidung für Ihren Einstieg in die Welt der Computer getroffen. Vielleicht sind Sie aber bereits erfahrener Anwender. In jedem Fall wird Sie Ihr NCR DECISION MATE V nun als professioneller Partner bei Ihrer täglichen Arbeit begleiten, Sie optimal unterstützen, und von zeitraubenden Nebentätigkeiten entlasten.

Die hohe Leistungsfähigkeit Ihres NCR DECISION MATE V wurde durch das modulare Systemkonzept und die konsequente Anwendung der neuesten Technologien erzielt. Eine große Anzahl von zusätzlichen Funktionen und ein breites Spektrum an einsetzbarer Software erlauben einen weiten Anwendungsbereich. Zudem ist Ihr NCR DECISION MATE V wegen seiner Einfachheit der Bedienung sehr benutzerfreundlich, wozu auch die ergonomisch richtige Gestaltung ganz wesentlich beiträgt.

Auch wenn Sie bereits Erfahrung im Umgang mit Computern haben, sollten Sie sich etwas Zeit nehmen, diese Anleitung zu lesen, da sie wichtige Informationen über Ihren Computer enthält.

Diese Bedienungsanleitung stellt Ihnen in fünf Kapiteln Ihren NCR DECISION MATE V vor. Die beiden ersten Kapitel machen Sie mit den Hauptelementen des Systems, dem Prozessor (Hardware) und dem Betriebssystem (Software) vertraut und beschreiben deren Installation. Kapitel 3 beschäftigt sich mit dem Betrieb, — also der Bedienung des Gerätes — sowie der Inbetriebnahme des Systems. Die letzten beiden Kapitel helfen Ihnen bei eventuell auftretenden Schwierigkeiten und geben Ihnen nützliche Empfehlungen, wie Sie den Computer pfleglich behandeln.

Nach dem Studium der Bedienungsanleitung und der Dokumentation des Betriebssystems sind Sie mit Ihrem Computer hinreichend vertraut und können bereits zahlreiche Operationen durchführen. Nun werden Sie sich für weitere Dokumentationen interessieren, die Sie in der Anwendung der Software weiterführen. Mit Hilfe dieser neu erworbenen Kenntnisse werden Sie in der Lage sein, Computersprachen und zusätzliche Betriebssysteme in Ihrem NCR DECISION MATE V einzusetzen. Hinweise dazu finden Sie im Kapitel *Systembeschreibung*.

Ihr NCR DECISION MATE V ist das Produkt eines erfahrenen Computerherstellers. Es wurde in Deutschland entwickelt und gefertigt.

Fortlaufende Forschung und Weiterentwicklungen erlauben es, Ihren Computer einfach und dem jeweiligen Einsatzzweck angepaßt auszubauen. Ihr Händler oder die nächstgelegene NCR Geschäftsstelle berät Sie hierzu gerne. Zum Abschluß möchten wir Ihnen nun viel Freude und Erfolg an Ihrem neuen Computer wünschen,

Ihre

NCR GmbH, Augsburg

# NCR DECISION MATE V BEDIENUNGSANLEITUNG

## INHALT

### INBETRIEBNAHME

Betriebsbereit in zwanzig Minuten .....	A
Drucker und Plotter .....	1-1

### SYSTEMBESCHREIBUNG

Hardware .....	2-1
Software .....	
Verbindungen innerhalb des Systems .....	

### BEDIENUNG DES COMPUTERS

Einführung .....	3-1
Disketten .....	
Festplatten .....	
Bedienungselemente .....	
Was wird beim Einschalten geprüft? .....	
Der Bildschirm .....	
Die Tastatur .....	
Wie kann ich Software laden? .....	
Einige praktische Übungen .....	

### WAS NUN?

Erkennen von Störungen .....	4-1
Service-Vereinbarungen .....	

### PRAKTISCHE HINWEISE

Aufstellen des Computers .....	5-1
Arbeiten mit dem Computer .....	
Behandlung der Disketten .....	
Ortswechsel des Computers .....	
Hinweise zur Pflege des Computers .....	

LEISTUNGSERWEITERUNGEN .....	6-1
------------------------------	-----

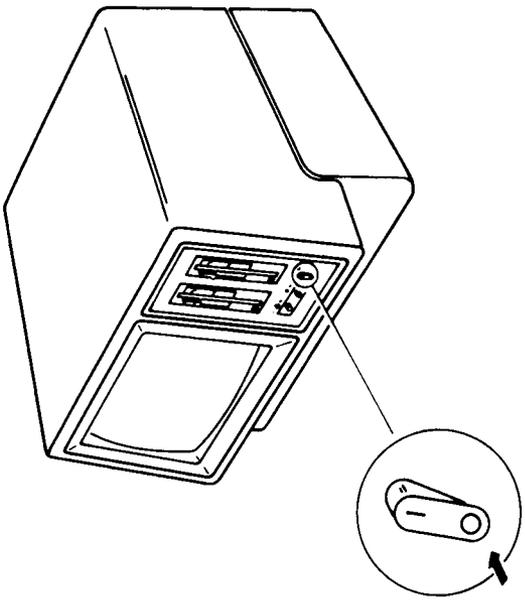
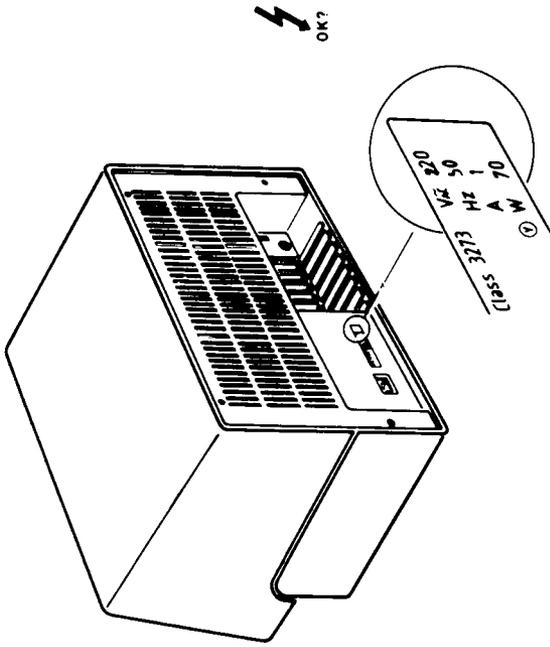
( )

)

( )

Die nachfolgenden Seiten zeigen Ihnen, wie Sie rasch und einfach Ihren NCR DECISION MATE V in Betrieb nehmen.

**BETRIEBSBEREIT IN ZWANZIG MINUTEN!**



Die auf dem Typenschild angegebene Spannung muß mit der an der Steckdose vorhandenen Spannung übereinstimmen.

Wenn dies nicht der Fall ist, sollten Sie sich mit Ihrem NCR-Partner in Verbindung setzen. Er wird Ihnen dann weiterhelfen.

Ihr NCR DECISION MATE V muß zunächst ausgeschaltet sein.

B

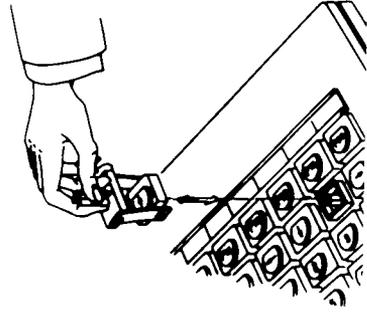
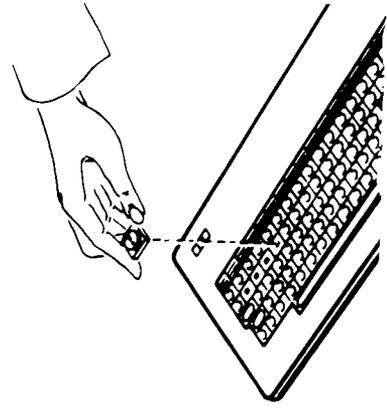
( ( (

Sie haben wahrscheinlich bereits bemerkt, daß die Tastatur nicht vollständig belegt ist. Dies erklärt sich dadurch, daß der NCR DECISION MATE V in vielen Ländern der Welt eingesetzt wird. Jede einzelne Sprache erfordert einige besondere Buchstaben oder andere sprachspezifische Zeichen. Auch die deutsche Sprache enthält besondere Buchstaben (ä, ö, ü, ß).

Die fehlenden Tasten befinden sich in einem durchsichtigen Kunststoffbehälter, der mit Ihrem NCR DECISION MATE V geliefert wurde.

Sie müssen lediglich die einzelnen Tasten in die Tastatur mit sanftem Druck einsetzen. Die Tasten rasten hörbar ein. Die Abbildungen "Tastatur-Belegung" (siehe nächste Seite) zeigen Ihnen, an welcher Stelle auf der Tastatur sich die jeweilige Taste befinden muß.

Im Fall, daß Sie versehentlich eine Taste an der falschen Stelle einsetzen, oder daß Sie später die Tastaturbelegung für eine andere Sprache auslegen möchten, sollten Sie das kleine, mitgelieferte Werkzeug verwenden (siehe Abbildung rechts).



## DIE TASTATUR-BELEGUNG

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20
ESC	1	2	3	4	5	6	7	8	9	0	?	;	←	TAB	↖	←	↑	↑	→
CONTROL	Q	W	E	R	T	Z	U	I	O	P	Ü	*	CONTROL		CLR	7	8	9	/
↵	A	S	D	F	G	H	J	K	L	Ö	Ä	↕			-	4	5	6	*
↵	≥	<	Y	X	C	V	B	N	M	;	=	↵			+	1	2	3	
															0	00	.	.	↵

## DEUTSCHSPRACHIGE TASTATUR

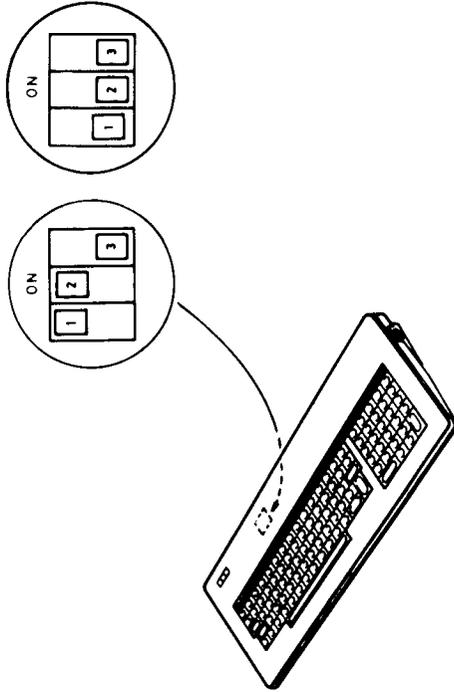
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20
ESC	1	2	3	4	5	6	7	8	9	0	?	;	←	TAB	↖	←	↑	↑	→
CONTROL	Q	W	E	R	T	Z	U	I	O	P	Ü	*	CONTROL		CLR	7	8	9	/
↵	A	S	D	F	G	H	J	K	L	Ö	Ä	↕			-	4	5	6	*
↵	≥	<	Y	X	C	V	B	N	M	;	=	↵			+	1	2	3	
															0	00	.	.	↵

## DEUTSCHSPRACHIGE TASTATUR (SCHWEIZ)

Sie müssen nun Ihrem NCR DECISION MATE V die Sprache mitteilen, die der von Ihnen gewählten Tastaturbelegung entspricht.

Die hierfür vorgesehenen Schalter befinden sich an der Bodenplatte der Tastatur. Diese drei roten Schalter sind mit den Nummern 1, 2 bzw. 3 gekennzeichnet. Neben den Schaltern befindet sich ein Aufkleber, der Ihnen Aufschluß über die Schalterstellungen für die verschiedenen Sprachen gibt.

Die zwei hier abgebildeten Schalterkombinationen entsprechen den zwei angebotenen deutschsprachigen Tastaturbelegungen.



Language Code

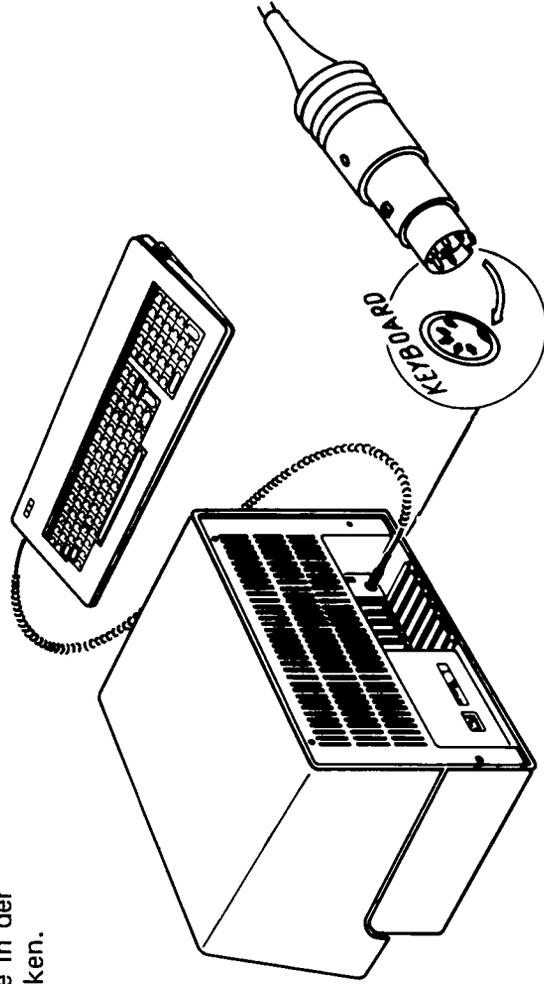
	1	2	3	
US English	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Swiss-German
UK/Int. English	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	Swiss-French
Danish	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	French
German	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	Canadian/Australian
Swedish/Finnish	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	Canadian(Bilingual)
Norwegian	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	South African
Spanish	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Portuguese
Italian	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Yugoslavian

= off       = on

Made in West Germany

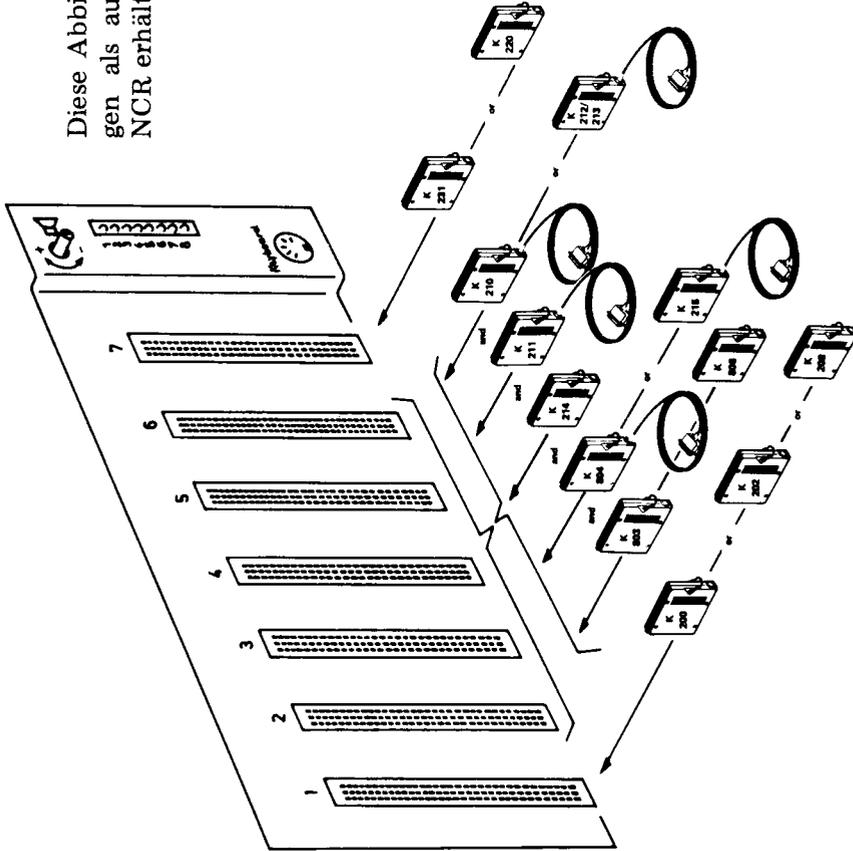
Sie können nun die Tastatur an Ihren Computer anschließen.

Dabei müssen Sie darauf achten, daß die Stellung der Einkerbung am Stecker mit derjenigen der Buchse übereinstimmt. Die fünf Stifte lassen sich dann mühelos in die in der Buchse befindlichen Fassungen einstecken.





Diese Abbildung zeigt Ihnen sowohl die Drucker-Verbindungen als auch die anderen Leistungserweiterungen, die von NCR erhältlich sind.



**ANMERKUNG:**

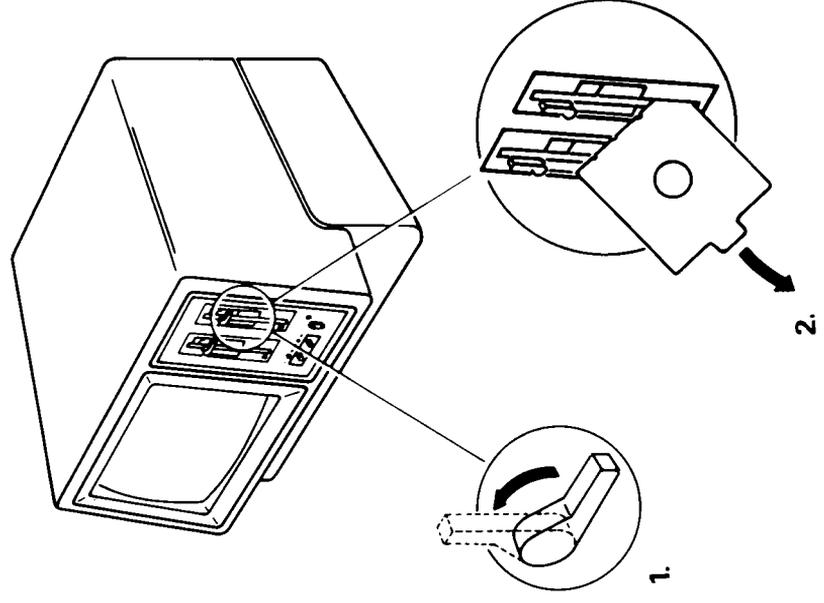
- K801 kann anstelle der Erweiterung K211, K212 oder K213 verwendet werden.
- Die Steckfassung 7 ist ausschließlich für das Diagnostik-Modul (K220) und die Prozessorerweiterung (K231) vorgesehen. Die Steckfassung 1 ist nur für eine Speichererweiterung zu verwenden.

Jedes Disketten-Laufwerk Ihres NCR DECISION MATE V enthält bei der Lieferung eine Scheibe aus Karton. Diese dient lediglich der Transportsicherung und sollte nun entfernt werden.

Bringen Sie den Sicherungshebel in die Senkrechtlage, indem Sie ihn nach links drehen (siehe Abbildung 1 rechts). Sie können nun die Scheibe entfernen (Abbildung 2).

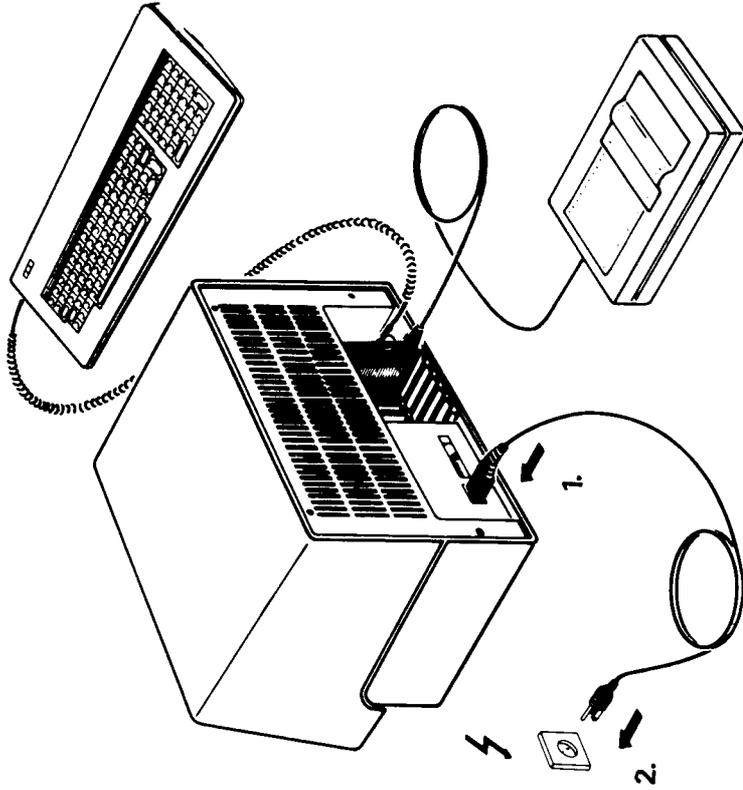
Bewahren Sie die Transportsicherungsscheiben gut auf. Bevor Sie Ihren Computer an einen anderen Ort verlegen, sollten Sie sie in die Disketten-Laufwerke wieder einlegen und die Sicherungshebel durch Rechtsdrehen in die Waagrechtstellung bringen.

Beim Entfernen und Einlegen der Transportsicherungsscheiben darf der Computer auf keinen Fall eingeschaltet sein.



Setzen Sie den Stecker des Netzkabels in die dafür vorgesehene Fassung an der Rückseite Ihres NCR DECISION MATE V ein.

Schließen Sie den Computer an die Wandsteckdose an.

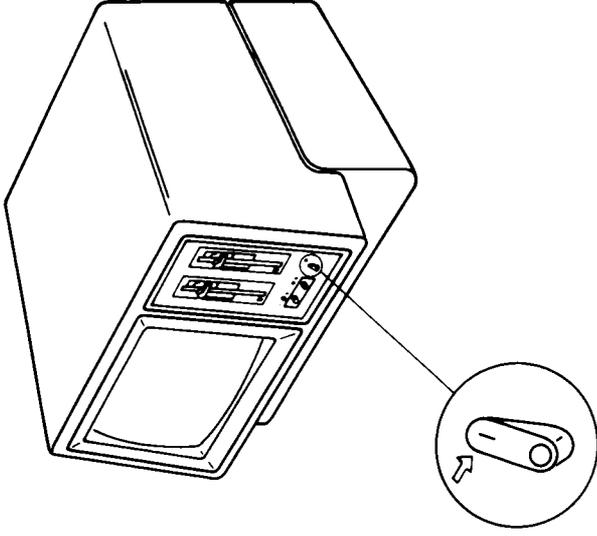


Ihr NCR DECISION MATE V ist nun betriebs-  
bereit.

Schalten Sie ein, indem Sie die orangefarbene  
Taste betätigen. Ihr Computer teilt Ihnen dann  
mit, daß er das Einlegen einer Diskette erwartet:

DISK A: NOT READY <CR>

Diese Mitteilung erscheint in der unteren linken  
Ecke des Bildschirms.



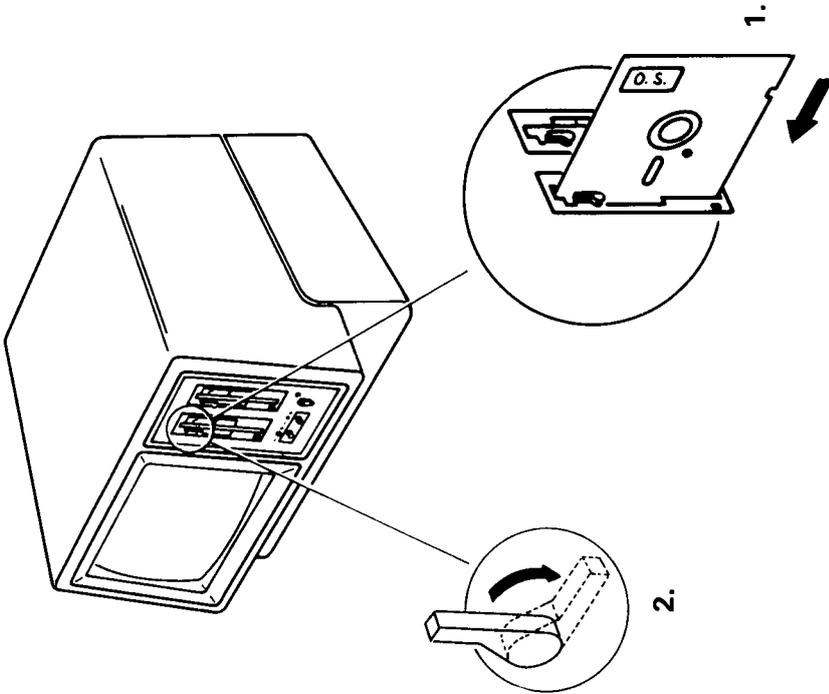
Ihr NCR DECISION MATE V kann nun das sogenannte Betriebssystem aufnehmen. Dieses befindet sich auf einer Diskette. Ein Betriebssystem besorgt die Verständigung zwischen den Bestandteilen Ihres Systems (Bildschirm, Tastatur, Speicher, Drucker usw.) und ermöglicht den Einsatz von Anwendungs-Software (z.B. ein Programm für die Erstellung einer Geschäftsbilanz).

Nehmen Sie die Betriebssystem-Diskette aus ihrer Schutzhülle heraus. Halten Sie die Diskette zwischen Daumen und Zeigefinger am Aufklebe-Etikett. Die Diskette läßt sich mühelos in das am weitesten links befindliche Disketten-Laufwerk einlegen. Dabei ist die links abgebildete Stellung der Diskette zu beachten (Abb. 1).

Nachdem Sie die Diskette eingelegt haben, müssen Sie den Sicherungshebel durch Rechtsdrehen in die Waagrechtstellung bringen (Abb. 2).

**WICHTIG: Vor dem Einlegen einer Diskette muß der Computer bereits eingeschaltet sein.**

Sie sollten eine Diskette nie knicken oder an der länglichen runden Öffnung anfassen. Sogar Fingerabdrücke, die vom menschlichen Auge nicht wahrgenommen werden, können eine Beschädigung der Speicherfläche verursachen. Es empfiehlt sich deshalb, jede Diskette in ihrer Schutzhülle aufzubewahren, wenn sie nicht gerade benötigt wird.



(

(

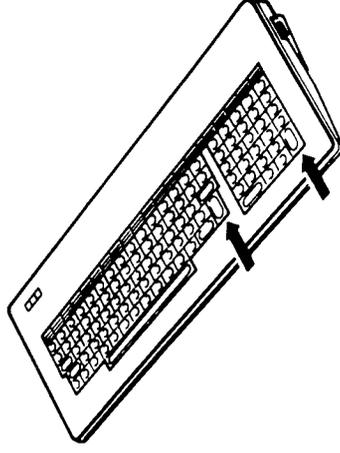
L

Die in der Abbildung mit Pfeilen gekennzeichneten Tasten erfüllen eine besondere Funktion. Je nach Ausführung der Tastatur Ihres NCR DECISION MATE V tragen diese Tasten die Bezeichnung **J** oder **NEW LINE**.

Mit dieser "Abschlußtaste" wird eine Aufforderung oder Mitteilung an den Computer Ihrerseits abgeschlossen und dem Computer übergeben.

Nachdem Sie die Betriebssystem-Diskette gemäß der auf Seite L enthaltenen Beschreibung in das Disketten-Laufwerk eingelegt haben, müssen Sie jetzt nur noch eine der zwei Abschlußtasten betätigen. Das Betriebssystem wird dann von der Diskette in den Computer geladen.

Sie können nun die Handhabung des von Ihnen gewählten Betriebssystems mit Hilfe des dazugehörigen Handbuchs kennenlernen. Dieses Handbuch enthält ein Kapitel ("Die ersten Schritte" oder "Starthilfe"), in dem die Erstellung der ersten, sehr wichtigen Kopie Ihrer Betriebssystem-Diskette beschrieben ist.



Die nachfolgenden Seiten (N...T) zeigen Ihnen die Tastaturbelegungen für die verschiedenen Landessprachen. Wenn Sie Ihre Tastatur für eine dieser Sprachen auslegen möchten, sollten Sie auch die Schalter an der Bodenplatte der Tastatur (siehe Seite E) entsprechend stellen.

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20
ESC	1	2	3	4	5	6	7	8	9	0	←	→	↵	TAB	DEL	←	↑	↓	→
CONTROL	Q	W	E	R	T	Y	U	I	O	P	{	}	CONTROL		DEL CLR	7	8	9	/
CAPS LOCK	A	S	D	F	G	H	J	K	L	;	"	~	NEW LINE		—	4	5	6	*
↵	Z	X	C	V	B	N	M	<	,	>	?	/			+	1	2	3	NEW LINE
															0	00	.		

ENGLISCH (USA)

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20
ESC	1	2	3	4	5	6	7	8	9	0	←	→	↵	TAB	CLR	←	↑	↓	→
CONTROL	Q	W	E	R	T	Y	U	I	O	P	{	}	CONTROL		CLR	7	8	9	/
↵	A	S	D	F	G	H	J	K	L	;	"	~			—	4	5	6	*
↵	Z	X	C	V	B	N	M	<	,	>	?	/			+	1	2	3	↵
															0	00	.		

ENGLISCH (GB)

( ) ( ) ( )

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
ESC	1	2	3	4	5	6	7	8	9	0	?	:	←	TAB
CONTROL	0	W	E	R	T	Y	U	I	O	P	A	..	CONTROL	
↑	A	S	O	F	G	H	J	K	L	O	A	⊗		
↓	>	<	Z	X	C	V	B	N	M	:	;	=	↑	↓

F16	F17	F18	F19	F20
↖	←	↑	↓	→
CLR	7	8	9	/
-	4	5	6	*
+	1	2	3	
0	00	.		

SCHWEDISH/FINNISCH

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
ESC	1	2	3	4	5	6	7	8	9	0	?	:	←	TAB
CONTROL	0	W	E	R	T	Y	U	I	O	P	A	..	CONTROL	
↑	A	S	O	F	G	H	J	K	L	O	A	⊗		
↓	>	<	Z	X	C	V	B	N	M	:	;	=	↑	↓

F16	F17	F18	F19	F20
↖	←	↑	↓	→
CLR	7	8	9	/
-	4	5	6	*
+	1	2	3	
0	00	.		

NORWEGISCH

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
ESC	1	2	3	4	5	6	7	8	9	0	?	!	~	TAB
CONTROL	Q	W	E	R	T	Y	U	I	O	P	~	*	CONTROL	
	A	S	D	F	G	H	J	K	L	N	;	{	}	
	>	<	Z	X	C	V	B	N	M	;	=	⇧	⇩	↵

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
ESC	1	2	3	4	5	6	7	8	9	0	?	!	~	TAB
CONTROL	Q	Z	E	R	T	Y	U	I	O	P	~	*	CONTROL	
	A	S	D	F	G	H	J	K	L	N	;	{	}	
	>	<	W	X	C	V	B	N	M	;	=	⇧	⇩	↵

SPANISCH

ITALIENISCH

( ) ( )

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
ESC	1	2	3	4	5	6	7	8	9	0	?	>	←→	TAB
CONTROL	Q	W	E	R	T	Y	U	I	O	P	A	..	CONTROL	
↓	A	S	D	F	G	H	J	K	L	Å	Ø	⊗		↵
↑	Z	X	C	V	B	N	M	:	=	⇧				↵

F16	F17	F18	F19	F20
↖	←	↓	↑	→
CLR	7	8	9	/
-	4	5	6	*
+	1	2	3	
0	00	.		↵

DÄNISCH

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
ESC	1	2	3	4	5	6	7	8	9	0	?	>	←→	TAB
CONTROL	Q	W	E	R	T	Z	U	I	O	P	Å	Ø	..	CONTROL
↓	A	S	D	F	G	H	J	K	L	Å	Ø	⊗		↵
↑	Z	X	C	V	B	N	M	:	=	⇧				↵

F16	F17	F18	F19	F20
↖	←	↓	↑	→
CLR	7	8	9	/
-	4	5	6	*
+	1	2	3	
0	00	.		↵

FRANZÖSISCH (SCHWEIZ)

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
ESC	1	2	3	4	5	6	7	8	9	0	;	'	→	TAB
CONTROL	A	Z	E	R	T	Y	U	I	O	P	~	*	CONTROL	
	↓	Q	S	D	F	G	H	J	K	L	M	%	^	↵
	↑	≥	W	X	C	V	B	N	?	:	;	'	=	↵

FRANZÖSISCH

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
ESC	!	@	2	3	4	5	6	7	8	9	0	;	'	→
CONTROL	Q	W	E	R	T	Y	U	I	O	P	[	~	CONTROL	
	↓	A	S	D	F	G	H	J	K	L	;	'	↵	
	↑	\	Z	X	C	V	B	N	<	>	?	/	↵	

AUSTRALIEN

R

(

(

(

,

)

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
ESC	1	2	3	4	5	6	7	8	9	0	→	←	→	TAB
CONTROL	Q	W	E	R	T	Y	U	I	O	P	^	⌘	CONTROL	
↓	A	S	D	F	G	H	J	K	L	;	;	;	↵	
↑	;	;	Z	X	C	V	B	N	M	<	>	<	↵	
F16	F17	F18	F19	F20										
/	←	↑	↑	→										
CLR	7	8	9	/										
-	4	5	6	*										
+	1	2	3											
0	00	00	.	↵										

KANADA (ZWEISPRACHIG)

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
ESC	↑	2	3	4	5	6	7	8	9	0	7	;	→	TAB
CONTROL	Q	W	E	R	T	Y	U	I	O	P	É	^	CONTROL	
↓	A	S	D	F	G	H	J	K	L	'	N	É	↵	
↑	;	;	Z	X	C	V	B	N	M	;	;	;	↵	
F16	F17	F18	F19	F20										
/	←	↑	↑	→										
CLR	7	8	9	/										
-	4	5	6	*										
+	1	2	3											
0	00	00	.	↵										

SÜDAFRIKA

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
ESC	1	2	3	4	5	6	7	8	9	0	?	~	←	TAB
CONTROL	A	Z	E	R	T	Y	U	I	O	P	Å	*	CONTROL	
↓	Q	S	D	F	G	H	J	K	L	Ç	Ö	;	↵	
↑	≥	W	X	C	V	B	N	M	:	=	↑			

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
ESC	1	2	3	4	5	6	7	8	9	0	?	⊖	←	TAB
CONTROL	Q	W	E	R	T	Z	U	I	O	P	Š	*	CONTROL	
↓	A	S	D	F	G	H	J	K	L	Č	Ž	;	↵	
↑	≥	Y	X	C	V	B	N	M	:	=	↑			

PORTUGIESISCH

JUGOSLAWISCH

## DRUCKER UND PLOTTER

Empfohlene Drucker und Plotter für den NCR DECISION MATE V.

- NCR 5403 (seriell) – Plotter
- NCR 6411 (seriell oder parallel) – Drucker
- NCR 6442 (parallel) – Drucker
- NCR 6455 (seriell) – Drucker

Die auf den folgenden Seiten beschriebenen Schalterstellungen der Drucker und Plotter gelten für den NCR DECISION MATE V. Weitere Informationen entnehmen Sie dem jeweiligen Drucker- bzw. Plotter-Handbuch:

- NCR 5403 Operator Instructions
- NCR 6411 Printer User Manual
- NCR 6442 Matrix Printer Operator Information
- NCR 6442 Matrix Printer Hardware Installation
- NCR 6455 Printer User Manual

Nur ein Drucker oder ein Plotter darf jeweils am System angeschlossen sein.

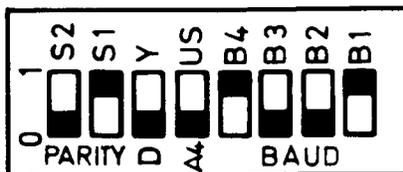
Ihre Betriebssystem-Software muß dem von Ihnen benutzten Drucker oder Plotter angepaßt werden. Der Abschnitt “Die ersten Schritte” oder “Starthilfe” Ihres Betriebssystem-Handbuchs zeigt Ihnen, wie Sie Ihre Betriebssystem-Software entsprechend anpassen können.

## NCR 5403

Die Datenübertragung vom NCR DECISION MATE V erfolgt seriell über den RS-232C-Plotter-Adapter (K213). Die Schalter befinden sich am Plotter (siehe "NCR 5403 Operator Instructions").

Die für den NCR DECISION MATE V erforderlichen Stellungen der Schalter:

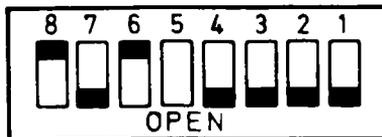
SCHALTER BEZEICHUNG	STELLUNG	FUNKTION
S2	0	} Keine Prüfung auf Parity
S1	1	
Y	0	Anschluß an Computer
US	0	Papierformat: A4
	1	Papierformat: US
B4	1	} Übertragungs- geschwindigkeit 4800 b/s
B3	0	
B2	0	
B1	1	



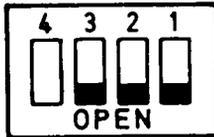
### NCR 6411

Die Datenübertragung vom NCR DECISION MATE V erfolgt seriell über den RS-232C-Peripherie-Adapter (K212). Hierzu müssen Sie Schalter an vier Schalteinheiten am Drucker stellen (siehe NCR 6411 Printer User Manual).

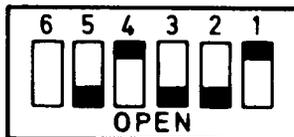
SCHALTER SW21		
SCHALTER-NUMMER	STELLUNG	FUNKTION
1	OPEN	Anzahl der Stop-Bits = 1
2	OPEN	SD
3	OPEN	} Parity-Prüfung: gerade
4	OPEN	
5		Nicht benutzt
6	nicht OPEN	Zeichenlänge: 7 bit
7	OPEN	} X-ON/X-OFF-Protokoll
8	nicht OPEN	



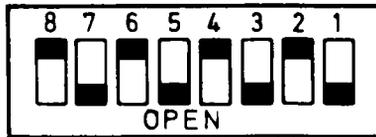
SCHALTER SW22		
SCHALTER-NUMMER	STELLUNG	FUNKTION
1	OPEN	} Datenübertragungs- geschwindigkeit: 9600 b/s
2	OPEN	
3	OPEN	
4		Nicht benutzt



SCHALTER SW23		
SCHALTER-NUMMER	STELLUNG	FUNKTION
1	nicht OPEN	} RS-232C
2	OPEN	
3	OPEN	} DSR, RS-232C
4	nicht OPEN	
5	OPEN	
6		Nicht benutzt



SCHALTER SW24		
SCHALTER-NUMMER	STELLUNG	FUNKTION
1 2	OPEN nicht OPEN	} DTR, X-ON/X-OFF
3 4	OPEN nicht OPEN	
5 6	OPEN nicht OPEN	} RTS
7 8	OPEN nicht OPEN	
		} CTS wird benutzt
		} CD wird nicht anerkannt



Der NCR 6411 kann mit dem Centronics-Peripherieadapter (K210) als Paralleldrucker betrieben werden. In diesem Fall ist das Stellen der Schalter nicht erforderlich.

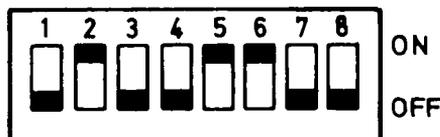
## NCR 6442

Die Datenübertragung vom NCR DECISION MATE V erfolgt parallel über den Centronics-Peripherieadapter (K210). Hierzu müssen die Schalter nicht gestellt werden.

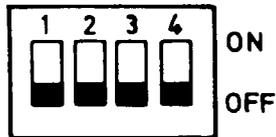
### NCR 6455

Die Datenübertragung vom NCR DECISION MATE V erfolgt seriell über den RS-232C-Peripherieadapter (K212). Hierzu müssen Sie Schalter an fünf Schalteinheiten am Drucker stellen (siehe NCR 6455 Printer User Manual).

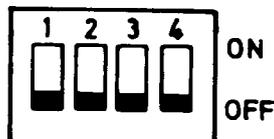
SCHALTER SW1		
SCHALTER-NUMMER	STELLUNG	FUNKTION
1	OFF	Einstellung auf externe Steuerung bereits beim Einschalten
2	ON	LF wird als Seitenvorschub anerkannt
3	OFF	einzelner horiz. Tab. zurückgesetzt
4	OFF	Auto-Return außer Kraft
5	ON	X-ON/X-OFF-Protokoll
6	ON	Interrupt-Signal wird beim Warnton gesetzt
7	OFF	Schalter für Seitenlänge außer Kraft
8	OFF	Normaler Betrieb (Fehlermonitor)



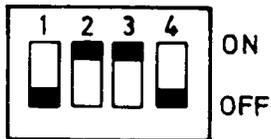
SCHALTER SW2		
SCHALTER-NUMMER	STELLUNG	FUNKTION
1	OFF	} Konstante Druckstellenbreite
2	OFF	
3	OFF	Normaler Betrieb (Papier- ende)
4	OFF	Normaler Betrieb (Test)



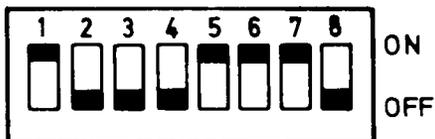
SCHALTER SW3		
SCHALTER-NUMMER	STELLUNG	FUNKTION
1	OFF	6 Zeilen/Zoll
2	OFF	} 10 Zeilen/Zoll
3	OFF	
4	OFF	



SCHALTER SW4		
SCHALTER-NUMMER	STELLUNG	FUNKTION
1	OFF	Schalter am Drucker für Zeilenvorschub außer Kraft
2	ON	} Prüfung auf Parity bei Senden/Empfangen: gerade
3	ON	
4	OFF	Halbduplex



SCHALTER SW1 AUF DER INTERFACE-PLATINE		
SCHALTER-NUMMER	STELLUNG	FUNKTION
1	ON	DSR eingeschaltet
2	OFF	CTS ausgeschaltet
3	OFF	Normale Prüfung v. CD (Modem)
4	OFF	Gegenrichtungskanal – active high
5	ON	Test
6	ON	2K-Puffer
7	ON	Nicht benutzt
8	OFF	Hammer aktiviert



**Ferner müssen Sie anhand der am Drucker befindlichen  
Bedienungselemente folgende Werte einstellen:**

**Geschwindigkeit: 8 (= 9600 baud)**

**Zeilen/Seite : 72**

**RS-232C X-ON/X-OFF**

TECHNISCHE DATEN				
Größe Computer Tastatur	Höhe 378 mm Höhe 37 mm	Breite 461 mm Breite 430 mm	Tiefe 370 mm Tiefe 216 mm	
Gewicht Computer Tastatur	22 kg 2 kg			
Netzspannung	Nennwert 100 Volt ~ 120 220 230 240	Bereich	90 bis 107 Volt ~ 104 bis 127 198 bis 235 207 bis 246 216 bis 257	
Netzfrequenz	50/60 Hz (49 – 61 Hz)			
Leistungsaufnahme	70 Watt (nur Computer, ohne Peripherie)			
Temperatur °C	Betrieb 10 bis 35, Lagerung -10 bis 50, Transport -40 bis 60			
Temperaturänderung	10° C per Stunde			
Relative Feuchtigkeit	20% bis 80%			
Kabellängen	Netzkabel 2,5 bis 3,5 Meter Tastatur 0,5 Meter, dehnbar Centronics Peripherie-Adapter 2,0 Meter RS-232C Peripherie-Adapter 2,0 Meter			
Schalldruckpegel	Bereitschaftsbetrieb 33dB (A), mit Diskette 41dB (A)			
Produktsicherheit	USA Canada Europa Deutschland	UL 478 UL CSA 22,2-154 IEC 380, werkseitig geprüft VDE 0806, GS-Zeichen		
Funkentstörung	USA Deutschland	FCC Docket No. 20780, Class B VDE 0871, Klasse A; FTZ-geprüft		
Strahlung	USA Deutschland	Public Law 90-602 DHEW Publication No. (FDA) 75-8003 Röntgenverordnung		

)

)

)

## SYSTEMBESCHREIBUNG

Ihr NCR DECISION MATE V besteht aus zwei zusammengehörenden Hauptelementen — der Hardware und der Software. Während die Hardware alle physikalischen Komponenten des Computers wie Prozessor und Tastatur umfaßt, besteht die Software aus verschiedenartigen Programmpaketen zum Betrieb des Computers. Dieser Abschnitt macht Sie mit der Grundausstattung der Hardware und Software bekannt und gibt Ihnen einen Überblick bezüglich zusätzlicher von NCR erhältlicher Leistungen und Erweiterungen.

### HARDWARE

Die Grundausstattung des NCR DECISION MATE V besteht aus dem Prozessor und der Tastatur. Die gesamte Hardware ist nach den neuesten Konstruktionsrichtlinien für Ergonomie sowie entsprechend den Sicherheitsvorschriften konzipiert.

### BASIS-KONFIGURATION

Das Kompakt-Gehäuse des Computers ist aufgrund seines Designs ideal für den Einsatz am Schreibtisch eingerichtet. Es enthält folgende wichtige Module:

- Prozessorplatine mit 64 KB Arbeitsspeicher
- 1 oder 2 Disketten-Laufwerke oder ein Disketten- und ein Festplattenlaufwerk
- Ein- oder mehrfarbiger Bildschirm, 305 mm (12 Zoll)
- Netzteil
- Steckerleisten für den Anschluß peripherer Einheiten und Leistungserweiterungen

Anmerkung: Ein KB (Kilobyte) = 1024 Zeichen (Bytes). Ein 64 KB-Speicher kann daher über 65000 Zeichen aufnehmen.

Der Prozessor ist in zwei Ausführungen lieferbar; als 8-Bit-Version für die Verwendung von 8-Bit-Software oder als 8/16-Bit-Version, um sowohl 8- als auch 16-Bit-Software verwenden zu können.

Die Eingabetastatur ist als freistehende Einheit über ein wendelförmiges, hochflexibles Kabel mit dem Prozessor verbunden. Die Tastatur verfügt über ein alphanumerisches Tastenfeld mit einigen Kontroll- und Funktionstasten und ein zusätzliches Ziffern-Tastenfeld für die schnelle Eingabe von ausschließlich numerischen Daten.

Ihr NCR DECISION MATE V verwendet Standard-5 1/4-Zoll-Disketten (130 mm) für zweiseitige Aufzeichnung mit doppelter Aufzeichnungsdichte. Dies ergibt eine Disketten-Speicherkapazität von bis zu 360 KB. Diese Disketten werden im Fachhandel oft als "double-sided, double-density"-Disketten bezeichnet.

Nur Disketten, die dieser Norm entsprechen, können eine einwandfrei Funktion gewährleisten. Sie können Ihre Disketten direkt von NCR oder einem anderen namhaften Hersteller beziehen.

Der interne Winchester-(Fest-)plattenspeicher bietet Ihnen eine noch erheblich größere Speicherkapazität von 10 MB.

Abbildung 2.1 zeigt die wichtigen Hardware-Bestandteile am Beispiel eines NCR DECISION MATE V mit einem Disketten- und einen Festplattenlaufwerk. Die Funktionsweise dieser Bestandteile wird im Kapitel 3 ("Bedienung des Computers") eingehend beschrieben.

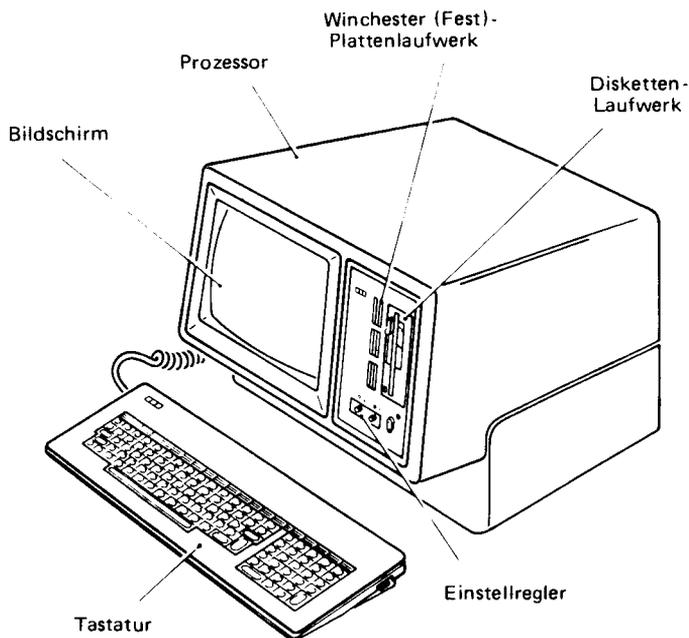


Abb. 2.1 NCR DECISION MATE V

Für technisch-wissenschaftliche Anwendungen ist eine besondere Hardware und Software-Ausstattung zu empfehlen. Diese besteht aus einem Hilfsprozessor und dem UCSD-p-Betriebssystem. Mittels dieser Ausstattung können technisch-wissenschaftliche Berechnungen besonders schnell ausgeführt werden. Dieser Einsatz Ihres NCR DECISION MATE V setzt folgende Hardware-Bestandteile voraus:

- 16-Bit-Prozessor
- Zwei Diskettenlaufwerke, oder ein Disketten- mit einem Festplattenlaufwerk

Mit Rücksicht auf dieses Einsatzgebiet des NCR DECISION MATE V werden folgende Hardware-Leistungserweiterungen angeboten:

- Echtzeit-Uhr (K803)
- IEEE-488-Interface (K804)

Ferner können bei Verwendung des UCSD-p-Betriebssystems ein RS-232C Drucker und ein RS-232C-Plotter gleichzeitig betrieben werden.

## **LEISTUNGSERWEITERUNGEN**

Der NCR DECISION MATE V wurde für einen weltweiten Einsatz konzipiert. Mit seinen verschiedenen Standardleistungen und Zusatzpaketen, die jetzt oder später verfügbar sind, läßt er sich jeder von Ihnen gewünschten Anforderung leicht anpassen. Die gegenwärtig erhältlichen Leistungserweiterungen werden nachstehend beschrieben. Sofern keine ausdrückliche Angabe einer zu benutzenden Steckfassung vorhanden ist, können Sie für ein Steckmodul jede beliebige Fassung 2 . . . 6 an der Rückseite des Computers verwenden.

### **8/16-Bit Prozessorerweiterung**

Diese Prozessorerweiterung ist in zwei Ausführungen erhältlich:

- als werkseitig eingebautes Modul (F230). Sämtliche Anschlüsse befinden sich innerhalb des Computer-Gehäuses. Infolgedessen stehen Ihnen alle sieben Steckfassung an der Rückwand des Computers für anderweitige Verwendungen zur Verfügung.
- als Steckmodul (K231) für die Fassung am weitesten rechts (7) an der Rückwand des Computers.

Die Funktionsweisen dieser zwei Ausführungen sind identisch: Ihr Computer kann nun sowohl mit 8-Bit- als auch mit 16-Bit-Betriebssystemen und den mit ihnen kompatiblen Anwenderprogrammen arbeiten.

### **Speichererweiterungsmodule**

Die von Ihnen gewünschte Speicherkapazität können Sie durch Auswahl des geeigneten Moduls bestimmen. Schieben Sie es in die Steckfassung am weitesten links (1) an der Rückwand des Computers ein. Derzeit sind drei Speichererweiterungen erhältlich:

- K200 — erweitert den Speicher von 64 KB auf 128 KB
- K202 — erweitert den Speicher von 64 KB auf 256 KB
- K208 — erweitert den Speicher von 64 KB auf 512 KB

### **Centronics-Peripherie-Adapter (K210)**

Mit diesem Adapter können Sie einen Centronics-kompatiblen Drucker (Parallel-Schnittstelle) anschließen. Hierzu müssen Sie den Adapter in eine der Steckfassungen (2 . . . 6) an der Rückseite Ihres NCR DECISION MATE V einschieben und das Kabel am Drucker anschließen. Der Adapter wird mit Kabel (2m) und einem geeigneten Stecker geliefert.

### **RS-232C-Communications-Peripherie-Adapter (K211)**

Mit diesem Adapter können Sie ein Gerät anschließen, das mit serieller Datenübertragung arbeitet. Anschluß und Ausstattung wie beim Adapter K210.

### **RS-232C-Drucker-Peripherie-Adapter (K212)**

Dieser Adapter ist dem Adapter K211 ähnlich. Der K212 ist für den Anschluß eines RS-232C-kompatiblen Druckers zu benutzen.

### **RS-232C-Plotter-Peripherie-Adapter (K213)**

Mit diesem Adapter können Sie einen kompatiblen Plotter (z.B. NCR 5403) an Ihren NCR DECISION MATE V anschließen.

### **RS-232C-Peripherie-Adapter (K801)**

Sie können diesen Adapter anstelle eines K211, K212 oder K213 benutzen. Die Umschaltung muß an zwei Stellen vorgenommen werden: Sie müssen einen mitgelieferten Stecker entsprechend der gewünschten Schnittstelle in die am Adapter dafür vorgesehene Fassung einsetzen. Der Stecker bietet Ihnen drei Möglichkeiten: K801-CC1 (für eine Communications-Schnittstelle), K801-CC2

(Drucker) und K801-CC3 (Plotter). Ferner müssen Sie die vier am Adapter befindlichen Schalter gemäß Abbildung 2.2 stellen.

DMV with Z80/8088	SWITCH				CABLE
CP/M MS-DOS UCSD-p	4	2	1	B	CONV.
PRINTER	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	2
COMMUNICATION	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	1
PLOTTER	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	3
PLOT.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	3
	○ = OFF ● = ON				

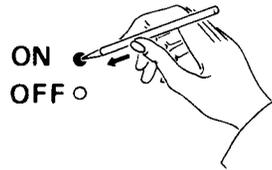


Abb. 2.2 Schalter (K801)

Anmerkungen:

1. Die Schalterstellungen für Communications und Drucker finden im Zusammenhang mit CP/M-, MS-DOS- und p-System-Software Anwendung.
2. Wenn Sie CP/M- oder MS-DOS-Software benutzen, müssen Sie beim Anschluß eines Plotters die Schalterstellungen für einen Drucker anwenden.
3. "Cable Conv." bezieht sich auf die Nummer des Steckers (CC1, CC2 oder CC3), den Sie mit der jeweiligen Stellung dieser Schalter benutzen sollten.

Falls Sie aus bestimmten Gründen einen von dieser Abbildung abweichenden Übertragungskanal (IFSEL = "Interface Select") einstellen wollen, sollten Sie Abbildung 2.3 heranziehen.

IFSEL	Schalter				Port-Adressen (hexadecimal)
	4	2	1	B	
0A	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	60-67
0B	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	68-6F
1A	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	70-77
1B	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	78-7F
2A	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	30-37
2B	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	38-3F
3A	<input type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	B0-B7
3B	<input type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	B8-BF
4A	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	C0-C7
4B	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	C8-CF
○ = OFF    ● = ON					

Abb. 2.3 IFSEL (K801, K803)

### Echtzeit-Uhr (K803)

Die Echtzeit-Uhr wird in eine der an der Rückwand Ihres NCR DECISION MATE V befindlichen Steckfassungen eingeschoben. Die Echtzeit-Uhr enthält auch einen Vierjahreskalender. Die Aufzeichnungseinheiten sind Monate, Tage, Stunden, Sekunden sowie Bruchteile einer Sekunde. Eine intern vorhandene Batterie sorgt dafür, daß die Uhr auch nach Einschalten des Computers den Zeitverlauf registriert.

Die Schalter, die sich am Gehäuse der Uhr befinden, sind gemäß nachstehender Tabelle zu stellen. Hierbei handelt es sich um die üblichen Stellungen für den Betrieb mit p-System-Software.

SCHALTER	STELLUNG
B	ON
1	OFF
2	OFF
4	ON

Abb. 2.4 Echtzeit-Uhr (Standard-Stellungen)

Sie können eine vom Standardwert (4B) abweichende IFSEL-Nummer bestimmen (siehe Abbildung 2.3).

### IEEE-488-Interface-Adapter (K804)

Dieser Adapter beinhaltet eine Schnittstelle zwischen Ihrem NCR DECISION MATE V und Meßgeräten, die der IEEE-488-Norm entsprechen. Sie können bis zu 15 Geräte in einer Ketten- (daisy chain) oder Stern-Konfiguration anschließen. Die Verkabelung darf nicht länger als 20m sein. Ferner darf die Kabellänge zwischen zwei Geräten nicht mehr als 2m betragen.

Sie können zusätzlich einen im Fachhandel erhältlichen Adapter und ein Kabel anschließen, damit Sie Ihr Interface-Modul mit Geräten benutzen können, die den IEC-625-Normen entsprechen.

Das Modul K804 unterstützt die Controller-, Listener- und Talker-Funktionen.

Die Schalter, die sich am Gehäuse des Interface-Adapters befinden, sind gemäß Abbildung 2.5 zu stellen. Hierbei handelt es sich um die üblichen Stellungen für den Betrieb mit p-System-Software.

SCHALTER	STELLUNG
A	OFF
1	OFF
2	ON
4	OFF

Abb. 2.5 IEEE-488-Interface (Standard-Stellungen)

Sie können eine vom Standardwert (2B) abweichende IFSEL-Nummer bestimmen (siehe Abbildung 2.6).

IFSEL	Schalter				Port-Adressen (hexadecimal)
	4	2	1	A	
0A	o	o	o	●	60-67
0B	o	o	o	o	68-6F
1A	o	o	●	●	70-77
1B	o	o	●	o	78-7F
2A	o	●	o	●	30-37
2B	o	●	o	o	38-3F
3A	o	●	●	●	B0-B7
3B	o	●	●	o	B8-BF
4A	●	o	o	●	C0-C7
4B	●	o	o	o	C8-CF

o = OFF    ● = ON

Abb. 2.6 IFSEL (K804)

**Maus-Adapter (K806)**

Dieser Adapter stellt die Verbindung zwischen Ihrem NCR DECISION MATE V und einer Maus her. Sie können mit diesem Adapter die folgenden Mäuse benutzen:

- Alps, Encoder-Mouse
- Hawley, Mark II
- Depraz, Souris P4
- Microsoft Mouse
- Logitech P4 oder LM-P-5
- Mouse Systems, Quad Mouse

Der Übertragungskanal (IFSEL-Nummer) läßt sich mit sieben Schaltbrücken innerhalb des Adapter-Gehäuses einstellen (siehe Abbildung 2.7). Mit Rücksicht auf die anderen Leistungserweiterungen Ihres NCR DECISION MATE V ist die IFSEL-Nummer 2A als Standardwert zu betrachten.

IFSEL	Schaltbrücken							Port-Adressen (hexadezimal)
	1	2	3	4	5	6	7	
0A	●	○	○	○	○	●	○	60-6F
0B	●	○	○	○	○	○	●	C8-6F
1A	○	●	○	○	○	○	●	70-7F
1B	○	●	○	○	○	○	●	78-7F
2A	○	○	●	○	○	○	○	30-3F
2B	○	○	●	○	○	○	○	38-3F
3A	○	○	○	●	○	○	○	B0-B7
3B	○	○	○	●	○	○	○	B8-BF
4A	○	○	○	○	●	○	○	C0-C7
4B	○	○	○	○	●	○	○	C8-CF

○ = offen (open)    ● = geschlossen (closed)

Abb. 2.7 IFSEL (Maus)

Drei weitere Schalter (J8, J9, J10) sind gemäß Abbildung 2.8 zu stellen:

Schalter	Maus
J8, J9, J10 – OFF	Hawley, Alps, Microsoft, Mouse Systems
J8, J9, J10 – ON	Depraz, Logitech

Abb. 2.8 Auswahlschalter (Maus)

### Adapter für synchrone/asynchrone Datenübertragung mit Zwischenspeicher (K215)

Dieser Adapter stellt die Verbindung zwischen Ihrem NCR DECISION MATE V und einem RS-232C-Modem her. Ein 2KB-Speicher (Buffer) ermöglicht eine hohe Übertragungsgeschwindigkeit.

**Numerischer Hilfsprozessor (K232)**

Sie können dieses Zusatzmodul in Verbindung mit dem 16-Bit-Prozessor (F230 oder K231) benutzen. Der Hilfsprozessor ermöglicht eine besonders hohe Rechengeschwindigkeit bei arithmetischen, trigonometrischen, logarithmischen und Exponentialbefehlen. Der Einbau des Hilfsprozessors setzt Erfahrung im Umgang mit modernen Halbleiter-Bauteilen voraus.

**Diagnostik-Modul (K220)**

Dieses Modul wurde für Techniker zwecks Diagnose etwaiger Fehlerzustände in der Hardware entwickelt. Nachdem Sie es in die Fassung 7 eingeschoben haben, können Sie zahlreiche Funktionsprüfungen durchführen. Der Spannungspegel der Logik-Signale (+5 und +12Vdc) wird ebenfalls geprüft; es wird angezeigt, ob diese Spannungen innerhalb des zulässigen Bereichs liegen. Die Serviceanleitung für den NCR DECISION MATE V enthält eine vollständige Beschreibung.

**Kippvorrichtung (K240)**

Sie können diese Vorrichtung an die Bodenplatte des Gehäuses befestigen. Sie haben dann die Möglichkeit, Ihren NCR DECISION MATE V in eine Schrägstellung (7°) zu bringen.

**Anschluß an Decision Net (K600)**

Mit diesem Adapter können Sie Ihren NCR DECISION MATE V an das Netzwerk Decision Net anschließen.

**Schutzsperre für Adapter (K880)**

Diese Sperre verhindert das Entfernen der Zusatzmodule, die Sie in die Steckfassungen an der Rückseite Ihres NDR DECISION MATE V eingesetzt haben.

**Ein zusätzliches integriertes Diskettenlaufwerk (K018, K019)**

Diese Zusatzmoduln erweitern die Plattenspeicherkapazität eines Systems mit einem Diskettenlaufwerk. Der einzige Unterschied zwischen K018 und K019 besteht darin, daß die Farbgestaltung des Gehäuses gemäß DIN (K018) und die Farbgestaltung gemäß dem Corporate Appearance Plan (CAP) nicht identisch sind.

**Festplatten (Winchester Disk)-Laufwerke (NCR 3282)**

Sie können bis zu drei externe Festplattenlaufwerke an Ihren NCR DECISION MATE V als Ketten-Konfiguration anschließen. Dies setzt voraus, daß Ihr NCR DECISION MATE V kein eingebautes

Festplattenlaufwerk beinhaltet. Die Speicherkapazität eines Festplattenlaufwerks nach Formatierung beträgt 10 MB.

Das Festplattenlaufwerk NCR 3282 ist in drei Ausführungen erhältlich:

- Als alleinstehendes Festplattenlaufwerk beinhaltet es eine eigene Plattensteuerung und einen Schnittstellen-Adapter für den Anschluß an den NCR DECISION MATE V. Bei der Benutzung dieser Ausführung können keine weiteren Festplattenlaufwerke angeschlossen werden.
- Als Hauptgerät beinhaltet es zusätzliche Hardware, die den Anschluß von bis zu zwei zusätzlichen Geräten in einer Ketten-Konfiguration ermöglicht.
- Als Zusatzgerät besitzt es weder eine eigene Plattensteuerung noch einen Schnittstellen-Adapter. Es kann in Verbindung mit einem weiteren Zusatzgerät mit einem Hauptgerät oder allein mit einem Hauptgerät betrieben werden.

Die hier beschriebenen Ausführungen werden mit Gehäuse, Netzteil, Kühlventilator, Netzkabel sowie den für die problemlose Aufstellung als Ketten-Konfiguration erforderlichen Datenkabeln geliefert.

Die Aufstellung erfordert keine umständliche Vorarbeit: Sie müssen lediglich die Laufwerke anschließen und (sofern Sie mehr als ein externes Laufwerk benutzen wollen) den mit einem Widerstand ausgerüsteten Endstecker in das Endgerät der Kette einsetzen. Sie können dann die Geräte einschalten.

### **Printer/Plotter**

Sie können beinahe jeden RS-232C oder Centronics-kompatiblen Drucker oder Plotter an Ihren NCR DECISION MATE V anschließen. Die folgenden Geräte sind besonders zu empfehlen:

- NCR 6411, Drucker (RS-232C oder Centronics)
- NCR 6442, Drucker (Centronics)
- NCR 6445, Drucker (RS-232C)
- NCR 5403, Plotter (RS-232C)

## **SOFTWARE**

Im allgemeinen arbeitet ein Computer mit drei Arten von Software: Betriebssystem, Programmiersprachen und Anwenderprogrammen. Die folgenden Definitionen sollen diese näher erläutern:

- Das Betriebssystem steuert den internen Ablauf im Computer, z.B. den Datenfluß zwischen den Funktionseinheiten untereinander, sowie die Organisation und Zuordnung des Speichers und der Plattenlaufwerke.
- Programmiersprachen enthalten eine Reihe von Programmbefehlen und deren Interpretation. Sie sind die Grundlage für Computerprogramme.
- Anwenderprogramme gibt es für eine Fülle von Applikationen. Der Benutzer kann sie fertig kaufen oder auch auf seinen speziellen Einsatzzweck hin selbst entwickeln.

## DAS BETRIEBSSYSTEM

Der wichtigste Teil der Software ist das Betriebssystem, denn ohne dieses kann der Computer nicht betrieben werden. Neben Kontrollprogrammen beinhaltet das Betriebssystem alle Informationen zur Verbindung der Hardware mit der Software. Das Betriebssystem befindet sich normalerweise auf einer Diskette. Derzeit sind viele Standardbetriebssysteme verfügbar. Diese Programmpakete sind jeweils für bestimmte Mikroprozessoren geschrieben und deshalb manchmal als 8-Bit oder als 16-Bit-Software bezeichnet. Für jedes dieser Betriebssysteme sind meist einige Anpassungen an die jeweilige Hardware erforderlich. Vier sehr weit verbreitete Betriebssysteme stehen Ihrem NCR DECISION MATE V zur Verfügung: CP/M-80, CP/M-86, MS-DOS, und das UCSD-p-System.

Wenn Ihr NCR DECISION MATE V einen 8-Bit-Prozessor verwendet, können Sie das CP/M-80-Betriebssystem einsetzen. Falls Sie sich für einen NCR DECISION MATE V mit einem 16-Bit-Prozessor entschieden haben oder eine Prozessorerweiterung benutzen (F230 oder K231), können Sie jedes beliebige der vier Betriebssysteme einsetzen.

CP/M ist eine Abkürzung für "Control Program for Microprocessors." CP/M-80 wurde von Digital Research, Inc. als Betriebs-

CP/M-86, ebenfalls von Digital Research entwickelt, ist ein Betriebssystem für 16-Bit-Prozessoren. Diese Software ist vollkommen kompatibel mit CP/M-80 und kann alle Dateien ohne Umwandlung weiterbenutzen. Die Verwendung von CP/M-80-kompatiblen 16-Bit-Anwenderprogrammen ist ebenfalls möglich.

MS-DOS ist ein Betriebssystem, das von der Microsoft Corporation für 16-Bit-Prozessoren entwickelt wurde. Dieses Programm arbeitet mit einer anderen Dateistruktur als CP/M, ist aber mit dem IBM Dateiformat kompatibel und erlaubt, wie CP/M-86, die Verwendung einer Reihe von 16-Bit-Anwenderprogrammen.

Bei Verwendung des UCSD-p-Systems können Sie UCSD-Pascal, Fortran-77 sowie BASIC in Ihrem NCR DECISION MATE

V einsetzen. Die vollständige Software-Kompatibilität mit dem IBM-PC UCSD-p-System ist gewährleistet. Das UCSD-p-System eignet sich besonders für CAD-Anwendungen, Grafik, Programm-entwicklung, technisch-wissenschaftliche Aufgaben sowie für den Unterricht.

Diese Betriebssysteme wurden für den NCR DECISION MATE V nicht nur wegen ihrer hervorragenden Leistungen ausgewählt, sondern auch wegen des sehr großen Angebots an geeigneten 8- und 16-Bit-Anwenderprogrammen.

### **ANWENDUNGS-PROGRAMME/PROGRAMMSPRACHEN**

Im Fachhandel ist bereits eine große Anzahl von Anwenderprogrammen erhältlich. Diese Pakete, die von verschiedenen Softwarehäusern angeboten werden, bieten Ihnen bestimmte Leistungen und Funktionen an, um Ihnen die Mühe und den Zeitaufwand für die Erstellung eigener Programme zu ersparen. Grundsätzlich müssen Sie nur noch die Programmdiskette in das Laufwerk legen und ein paar Hardwareparameter definieren. Sie können dann das Programm ausführen.

Aus der Tatsache, daß ein Programm in Ihrem Computer "läuft", kann man nicht folgern, daß dieses Programm eine optimale Ausnutzung der Leistungen Ihres NCR DECISION MATE V darstellt. Aus diesem Grunde hat NCR mehrere Anwenderprogramme getestet. Eine Auswahl der getesteten Programme wird von NCR empfohlen und auch angeboten.

Neben dieser Anwendersoftware stehen Ihnen auch Programm-Sprachen zur Verfügung, damit Sie eigene Programme entwickeln können.

### **ANPASSUNG DER SOFTWARE**

Die Leistungserweiterungen K215, K801, K803, K804 und K806 können mit jedem beliebigen Datenübertragungskanal (IFSEL 0A, 0B, 1A, 1B, 2A, 2B, 3A, 3B, 4A oder 4B) eingesetzt werden. Der in der jeweiligen Beschreibung angegebene Standardwert bezieht sich auf die Voreinstellung der dazugehörigen, von NCR angebotenen Software. Sie können die Schalter am Adapter und die Software auf andere IFSEL-Nummern einstellen. Damit sind auch zukünftige Leistungserweiterungen Ihres NCR DECISION MATE V berücksichtigt. Wenn Sie eine neue IFSEL-Nummer an einem Adapter einstellen, müssen Sie dafür sorgen, daß die von der Software benutzten Port-Adressen der neuen IFSEL-Nummer entsprechen.

Eine Neueinstellung der IFSEL-Nummer ist bei normaler Verwendung Ihres Computers und seiner Zusatzgeräte kaum erforderlich. Wenn Sie aber eigene Programme zur Steuerung dieser Geräte schreiben wollen, sollten Sie die Zuteilung der Port-Adressen flexibel gestalten, damit auch diese Software die Leistungsfähigkeit Ihres NCR DECISION MATE V voll ausnutzen kann.

Abbildung 2.9 ist besonders für Programmierer von Interesse. Sie beinhaltet eine Aufstellung der IFSEL-Nummern und der Port-Adressen, über die sie verfügen. Die für die IFSEL-Nummern vorgesehenen Leistungserweiterungen sind ebenfalls angegeben. Bei den obengenannten Leistungserweiterungen handelt es sich, wie bereits erwähnt, nicht um verbindliche Werte.

Leistungserweiterung	IFSEL												Steck-Fassung	DMA-KANAL	
	0A	0B	1A	1B	2A	2B	3A	3B	4A	4B					
DRUCKER Seriell/parallel K212/K213	■	■												2,6	
PLOTTER K213 (oder K801)	■	■												2,6	
COMMUNICATIONS K211			■	■										2,6	
RS-232C (umschaltbar) K801	■	■	■	■	■	■	■	■	■	■	■	■	■	2,6	
PLOTTER K801 (oder K213)	■				(mit CP/M oder MS-DOS)								2,6		
PLOTTER K801					(mit p-System)								2,6		
SYNCHR./ASYNCHR. UBERTRAGUNG K215			■											2,6	
ECHTZEIT-UHR K803												■		2,6	
IEEE-488 K804						■								2,6	
MAUS K806					■									2,6	
DECISION NET K600									■					2,6	1
FESTPLATTE (extern) NCR 3282										■				2,6	
FESTPLATTE (intern)										■					
PORT-ADRESSE von bis	60H 67H	68H 6FH	70H 77H	78H 7FH	30H 37H	38H 3FH	80H 87H	88H 8FH	C0H C7H	C8H CFH					
■ : Software und Hardware ■ : nur von der Hardware benötigt															

Abb. 2.9 IFSEL-Zusammenfassung

)

)

)

# BEDIENUNG DES COMPUTERS

## EINFÜHRUNG

Sie haben Ihren NCR DECISION MATE V bereits aufgestellt und ggf. die Leistungserweiterungen angeschlossen.

Dieses Kapitel beschreibt die Bedienung der Hauptbestandteile des Systems:

- Disketten und Diskettenlaufwerke
- Winchester (Fest-)plattenlaufwerk
- Bedienelemente am Computer
- Bildschirm
- Tastatur

Nachdem Sie diese Abschnitte gelesen haben, können Sie das Laden Ihrer Betriebssystem-Diskette vornehmen. Dieses Verfahren wird im Abschnitt "Wie kann ich Software laden?" beschrieben. Am Ende dieses Kapitels finden Sie einige praktische Übungen.

## DISKETTEN

Der Computer arbeitet mit zweiseitig beschreibbaren 5 1/4-Zoll-Disketten zur Speicherung des Betriebssystems und anderer Dateien. Zur näheren Betrachtung ziehen Sie die Diskette vorsichtig aus ihrer Schutzhülle heraus; die Abbildung 3.1 zeigt Ihnen die einzelnen Merkmale der Diskette.

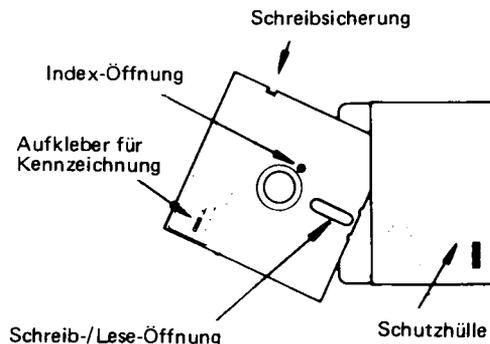


Abb. 3.1 Diskette

## HANDHABUNG DER DISKETTEN

Die Diskette sollte sich immer in ihrer Schutzhülle befinden, wenn sie nicht gerade im Laufwerk benutzt wird. Zur Aufbewahrung mehrerer Disketten empfehlen sich die hierfür erhältlichen speziellen Sammelboxen.

Halten Sie die Disketten von Magnetfeldern wie Motoren, Transformatoren oder sonstigen Magneten fern. Die Disketten sind ferner vor Staub- und Feuchtigkeit zu schützen.

Disketten sollen möglichst im gleichen Umgebungsklima wie der Computer selbst aufbewahrt werden. Bei extremen Unterschieden ist es zweckmäßig, die Disketten vor Betrieb etwa eine Stunde lang am Standort des Computers zu lagern. Disketten, die wichtige Daten wie Betriebssystem oder Anwenderprogramme beinhalten, sollten mit einer Schreibsicherung versehen werden. Diese Schreibsicherung wird dadurch erreicht, daß Sie einen kleinen (mit Disketten meistens mitgelieferten) Aufkleber über der Einkerbung für die Schreibsicherung anbringen. Auf diese Weise sorgen Sie dafür, daß die Diskette im Computer nicht versehentlich gelöscht wird. Die Betriebssystemdiskette, die Sie von NCR erhielten, ist mit einem permanenten Schreibschutz versehen.

## DISKETTENLAUFWERKE

Ihr NCR DECISION MATE V ist mit einem oder zwei Laufwerken für 5 1/4-Zoll-Disketten ausgestattet. Das linke Laufwerk wird mit "A", das rechte mit "B" bezeichnet. Bei den Ausführungen mit einem Diskettenlaufwerk erhält dieses die Bezeichnung "A". Hier handelt es sich um die Bezeichnungen, die die Betriebssysteme den Laufwerken zuteilen, wobei das UCSD-p-System eine Ausnahme bildet: Das linke Laufwerk wird als "4" und das rechte Laufwerk als "5" bezeichnet.

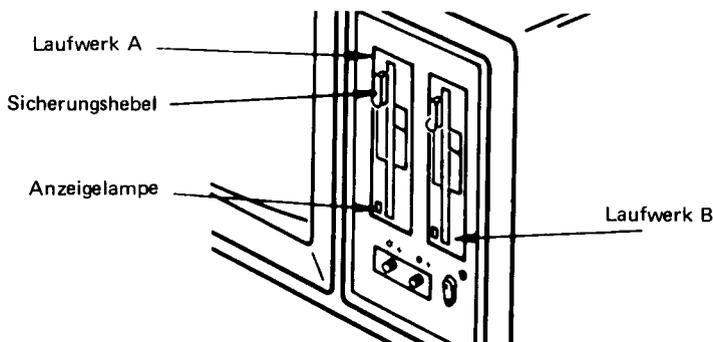


Abb. 3.2 Diskettenlaufwerke

Jedes Laufwerk hat eine Öffnung zur Aufnahme der Diskette, einen Sicherungshebel und eine rote Anzeigelampe (LED), siehe Abbildung 3.2. Die rote Anzeigelampe leuchtet immer dann auf, wenn auf die Diskette geschrieben oder von ihr gelesen wird. Es kann jeweils nur ein Laufwerk im Computer aktiv sein.

### DAS EINLEGEN VON DISKETTEN

Um eine Diskette in ein Diskettenlaufwerk einzulegen, verfahren Sie wie folgt:

- Öffnen Sie das Laufwerk, indem Sie den Verschußhebel nach links drehen. Er befindet sich dann in einer Senkrechtstellung (Abbildung 3.2).
- Entfernen Sie die Diskette aus der Schutzhülle.
- Schieben Sie die Diskette in das Laufwerk ein. Dabei zeigt die Schreib-/Lesekopfföffnung voraus und der Aufkleber in Richtung Bildschirm (siehe Abbildung 3.3).
- Drehen Sie den Verschußhebel nach rechts

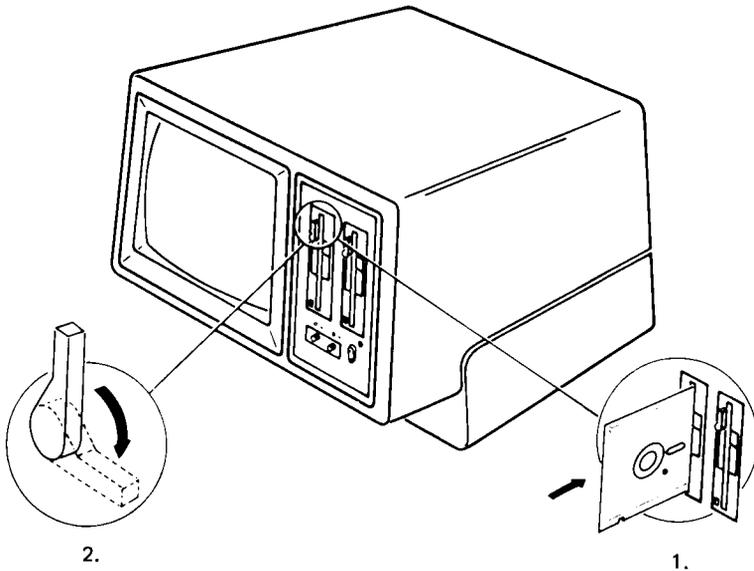


Abb. 3.3 Laden einer Diskette

Die Diskette ist nun im Laufwerk betriebsbereit. Die rote Anzeigelampe leuchtet allerdings erst dann auf, wenn das Betriebs-

system einen Schreib- oder Lesezugriff auf die Diskette vornimmt.

Wenn Sie die Diskette herausnehmen wollen, müssen Sie folgendes beachten:

- Warten Sie, bis die rote Lampe erlischt.
- Drehen Sie den Verschußhebel nach rechts.
- Entfernen Sie die Diskette aus dem Laufwerk und legen Sie sie in ihre Schutzhülle.

### WICHTIG

Nehmen Sie die Diskette nie vor Abschluß des Arbeitsvorgangs und dem Erscheinen des Bereitschaftszeichens des Betriebssystems (z.B. A> ) aus dem Laufwerk.

### FESTPLATTENSPEICHER

Im Prinzip arbeiten Systeme mit Festplatten (Winchester)-Laufwerken (siehe Abbildung 3.4) wie diejenigen mit zwei Disketten. Die Winchesterplatte ist ein Massenspeicher mit einer Kapazität von 10 MB und erfordert keine besondere Berücksichtigung in der Handhabung. Der Bereitschaftszustand des Laufwerks wird durch eine rote Leuchtdiode angezeigt. Die Geräten mit Festplatte ist das Diskettenlaufwerk rechtsseitig angeordnet und mit "A" bezeichnet.

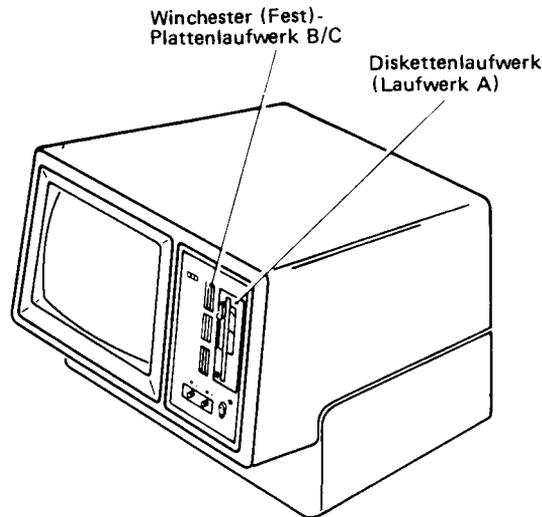
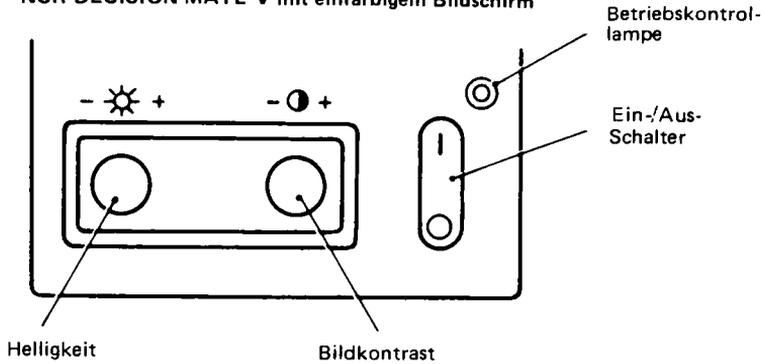


Abb. 3.4 System mit Festplatte

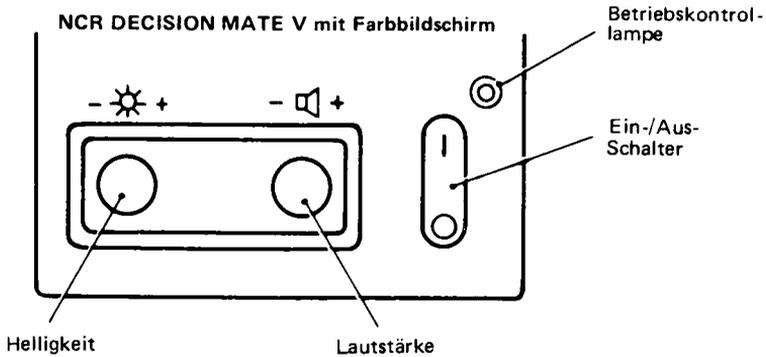
net. Das Winchesterlaufwerk enthält zwei Platten, die "B" und "C" genannt werden.

### BEDIENUNGSELEMENTE

NCR DECISION MATE V mit einfarbigem Bildschirm



NCR DECISION MATE V mit Farbbildschirm



Andere Ausführung

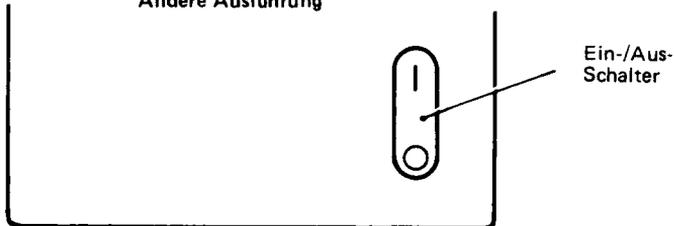


Abb. 3.5 Die Bedienelemente

Die wesentlichen Bedienungselemente des Computers befinden sich unterhalb der Diskettenlaufwerke (siehe Abbildung 3.5). Sie umfassen je nach Ausführung den Ein-/Aus-Schalter, die Betriebskontrolllampe und die Einstell-Drehknöpfe für den Bildschirm.

### **NCR DECISION MATE MIT EINFARBIGEM BILDSCHIRM**

Die Bedienungselemente umfassen einen Ein-/Aus-Schalter, eine Lampe, die bei eingeschaltetem Betriebszustand aufleuchtet, sowie zwei Drehknöpfe zur Steuerung des Bildschirms.

#### **Ein-/Aus-Schalter**

Drücken Sie den oberen Teil (1) des Schalters. Der Computer ist nun eingeschaltet, und die grüne Lampe leuchtet auf. Nachdem Sie Ihre Arbeiten mit dem Computer beendet haben, drücken Sie den unteren Teil (0) des Schalters, um den Computer auszuschalten.

#### **Helligkeit und Kontrast**

Stellen Sie mit den beiden Drehknöpfen den für Sie am angenehmsten erscheinenden Bildschirmzustand ein.

#### **Lautstärke**

Bei Systemen mit einfarbigem Bildschirm ist die Lautstärke des eingebauten Lautsprechers mit dem an der Rückseite angebrachten Regler einstellbar (siehe Abbildung 3.6).

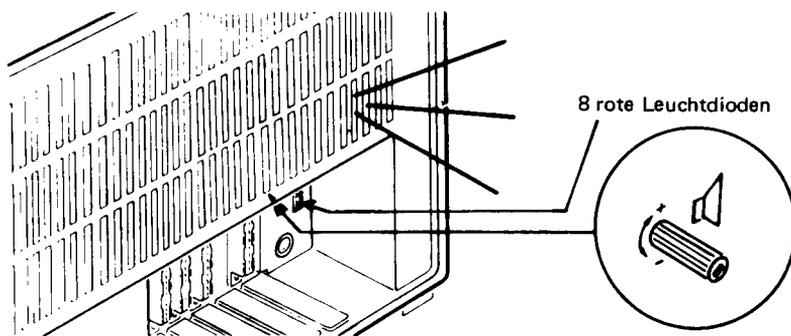


Abb 3.6 Regler für Lautstärke

### **NCR DECISION MATE V MIT FARBBILDSCHIRM**

Bei einem System mit einem Farbbildschirm sind der Ein-/Aus-Schalter, der Helligkeits-Drehknopf und die grüne Kontrolllampe wie bei einem System mit einfarbigem Bildschirm angebracht. Der

Drehknopf für Bildkontrast entfällt. An dieser Stelle befindet sich der Drehknopf für die Lautstärke des im NCR DECISION MATE V vorhandenen Lautsprechers. Dadurch entfällt der Regler für Lautstärke, die sich bei der vorher beschriebenen Ausführung an der Rückseite des Gehäuses befindet.

## ANDERE AUSFÜHRUNG

Das einzige Bedienungselement ist der Ein-/Aus-Schalter. Die "fehlenden" Funktionen — Bildkontrast und Lautstärke — werden von den Funktionstasten besorgt, die sich in der oberen Reihe der Tastatur befinden. Die Lautstärke kann mit acht Stufen, der Bildkontrast mit fünf Stufen eingestellt werden. Beim Einschalten des Computers werden für diese Funktionen normalen Einsatzbedingungen entsprechende Werte automatisch eingestellt, die Sie aber nach Wunsch gemäß folgender Beschreibung ändern können.

- Wenn Sie die Funktionstaste F1 bei gedrückter CONTROL-Taste betätigen, wird die Lautstärke vermindert.
- Wenn Sie die Funktionstaste F2 bei gedrückter CONTROL-Taste betätigen, wird die Lautstärke erhöht.
- Die Betätigung der Funktionstaste F3 bei gedrückter CONTROL-Taste setzt die Helligkeit des Bildschirms herab.
- Die Betätigung der Funktionstaste F4 bei gedrückter CONTROL-Taste setzt die Helligkeit des Bildschirms herauf.

## WAS WIRD BEIM EINSCHALTEN GEPRÜFT?

Sobald Sie Ihren NCR DECISION MATE V einschalten, führt er eine Prüfung seiner internen Hardware durch. Das Ergebnis dieser Überprüfung wird Ihnen mittels der acht an der Rückseite des Gehäuses befindlichen roten Leuchtdioden mitgeteilt (siehe Abbildung 3.6): Im Falle, daß nicht alle Lampen nach einigen Sekunden erlöschen, liegt ein Defekt vor.

## DER BILDSCHIRM

Der Bildschirm dient der visuellen Anzeige von Tastatur-Eingaben, Programmbefehlen, Systemmeldungen und anderen Informationen. Der Bildschirm ist für die alphanumerische Darstellung in 25 Zeilen zu je 80 Zeichen eingeteilt. Für graphische Darstellungen steht eine Auflösung von 640 horizontalen und 400 vertikalen Bildpunkten

zur Verfügung. Beide Darstellungsarten sind getrennt oder kombiniert anwendbar.

Die grünen Zeichen auf dunklem Hintergrund ergeben in Verbindung mit dem reflexionsfreien Bildschirm eine scharfe und flimmerfreie Anzeige. Mit dem Helligkeits- bzw. Kontrastregler läßt sich die Darstellung optimal an das jeweilige Bedienerblickfeld sowie die Lichtverhältnisse anpassen.

Systeme mit einem Farbbildschirm können folgende Farben darstellen: schwarz, weiß, rot, grün, blau, gelb, Magentarot und Zyan. Die Farbauswahl erfolgt durch das Programm. Bei diesen Systemen entfällt die Einstellung des Bildkontrastes.

## DIE TASTATUR

Die Tastatur wird zur Eingabe von Befehlen und Daten benutzt. Das sogenannte "Roll-over" ermöglicht eine sehr schnelle Arbeitsweise durch überlappende Eingabe. Das heißt, Sie können eine Taste drücken, noch bevor die vorher gedrückte Taste in seine Ausgangsstellung zurückgegangen ist. Ein 8-Zeichen-Eingabespeicher übernimmt die Zwischenspeicherung, bis das Programm die jeweils anliegenden Daten abrufen und verarbeitet. Wenn Sie eine Taste im gedrückten Zustand halten, wird das entsprechende Zeichen wiederholt erzeugt.

Akzeptiert der Computer Ihre Eingaben, so hören Sie einen kurzen Ton; fehlerhafte Eingaben quittiert er mit einem längeren Ton. In letzterem Fall ist eine erneute Eingabe notwendig.

Die Tastatur ist in zwei Felder aufgeteilt: einen alphanumerischen Teil, ähnlich dem einer Schreibmaschinentastatur, und ein rein numerisches Feld. Jedes Feld enthält eine Anzahl von Spezialtasten, deren programm- und anwenderabhängige Funktionen getrennt in der jeweiligen Softwaredokumentation aufgeführt sind. Die überwiegende Mehrzahl der Tasten erfüllen aber einheitliche, von der jeweils benutzten Software unabhängige Funktionen. Diese Tasten sind in den folgenden Abschnitten beschrieben.

## DIE ALPHANUMERISCHE TASTATUR

Das alphanumerische Feld besteht aus einer standardisierten Schreibmaschinenanordnung und einigen Spezialtasten zur Steuerung des Computers. Die Funktion der Spezialtasten ist unabhängig von der Landessprache. Für die übrigen Tasten bestehen nur geringe Unterschiede zwischen den Ländervarianten. Die nachfolgenden Abbildungen beziehen sich auf die deutsche Ausführung.

### Alphabetische Tasten

Bei Betätigung einer der in Abbildung 3.7 durch Umrandung hervorgehobenen Tasten erscheint das entsprechende Zeichen als Kleinbuchstabe auf dem Bildschirm und der Computer übernimmt den dazugehörigen Code. Sollen die Zeichen als Großbuchstaben eingegeben und angezeigt werden, so ist vorher die Umschalt-Feststelltaste zu betätigen.

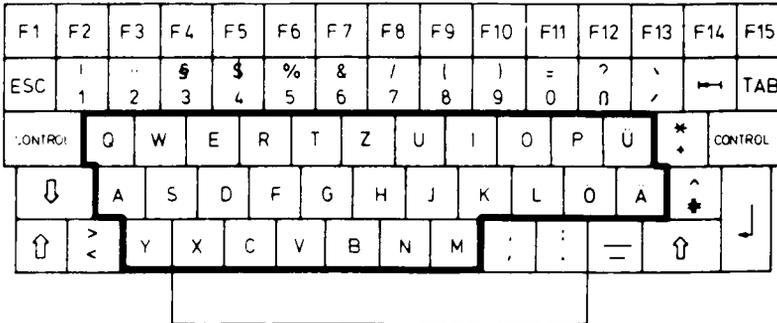


Abb 3.7 Alphabetische Tasten

### Die Umschalt-Feststelltaste

Nach dem Drücken bleibt diese Taste (siehe Abbildung 3.8) fest in aktivierter Position (Umschalt-Feststeller); alle alphabetischen Zeichen der Abbildung 3.7 erscheinen als Großbuchstaben. Die Funktion wird durch nochmaliges Drücken aufgehoben.

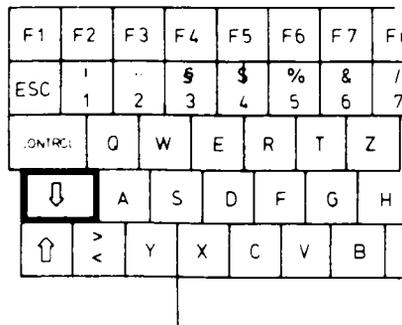


Abb. 3.8 Die Umschalt-Feststelltaste

### Numerische Tasten

Bei Betätigung einer der in Abbildung 3.9 durch Umrandung hervorgehobenen Tasten erfolgt die Eingabe und Anzeige des jeweils in der unteren Hälfte der Taste gezeigten Zeichens. Wenn Sie die Taste zusammen mit einer der Umschalttasten (Abbildung 3.10) drücken, wird das auf der oberen Hälfte der Taste abgebildete Zeichen eingegeben und angezeigt.

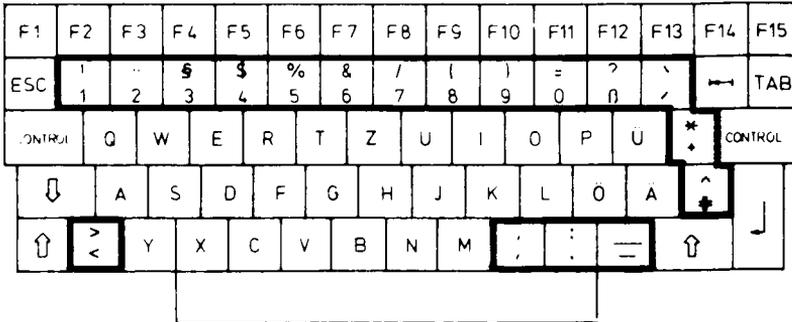


Abb. 3.9 Numerische und Symbol-Tasten

### Die Umschalt-Tasten

Zur leichteren Bedienung enthält die Tastatur rechts und links eine Umschalt-Taste (Abbildung 3.10). Diese Tasten sind nicht feststellbar und gehen daher nach Wegnahme des Fingers wieder in ihre Ausgangsstellung zurück. Solange diese Taste in gedrückter Stellung ist, wird das in der oberen Hälfte einer Taste abgebildete Zeichen bzw. ein Großbuchstabe erzeugt. Wenn gleichzeitig die Umschalt-Feststelltaste sich in gedrückter Stellung befindet, wird ein Kleinbuchstabe statt eines Großbuchstabens erzeugt.

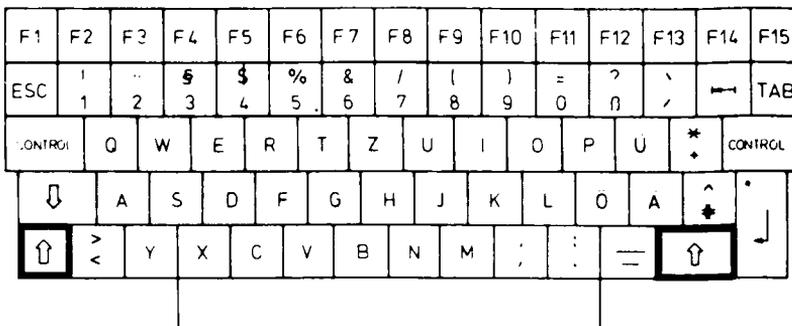


Abb. 3.10 Die Umschalt-Tasten

### Die Rücktaste

Die Betätigung der Rücktaste (Abbildung 3.11) löscht das zuletzt eingegebene Zeichen und setzt die Schreibmarke (Cursor) am Bildschirm um eine Stelle zurück.

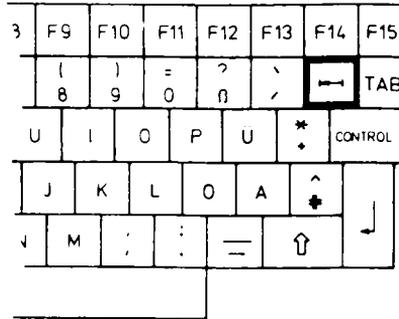


Abb. 3.11 Die Rücktaste

### Die CONTROL-Tasten

Diese Tasten (Abbildung 3.12) verändern die Funktion einiger alphabetischer Tasten zur Steuerung des Computers. Die speziellen Funktionen sind von der Software abhängig und werden an anderer Stelle beschrieben. Die CONTROL-Funktion ist durch Betätigung einer der beiden CONTROL-Tasten zu aktivieren. Der Finger muß auf der Taste bleiben, während die gewünschte alphabetische Taste gedrückt wird. Nach Loslassen der CONTROL-Taste nimmt die alphabetische Taste ihre normale Funktion wieder an.

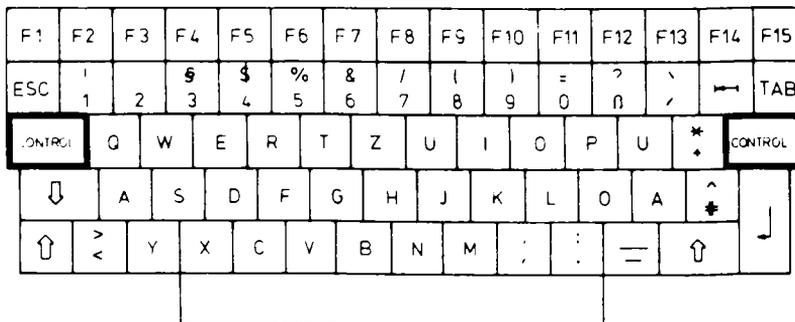


Abb. 3.12 Die CONTROL-Tasten

### Die Abschlußtaste

Diese ist die meistgebrauchte Taste (Abbildung 3.13); sie ist deshalb sowohl in der alphanumerischen Tastatur wie auch in der rechts befindlichen numerischen Tastatur vorhanden. Sie meldet dem Computer, daß eine Eingabe abgeschlossen ist.

Anmerkung: Diese Taste wird oft als Carriage Return (CR) oder NEW LINE bezeichnet.

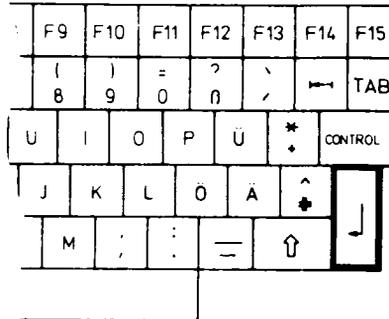


Abb. 3.13 Die Abschlußtaste

### Programmierbare Tasten

Die Funktionsweise der in der alphanumerischen Tastatur verbleibenden Tasten (siehe Abbildung 3.14) — ESC, TAB, F1 bis F15 — ist von der benutzten Software (Betriebssystem, Anwendungsprogramm) abhängig. Oberhalb der Tasten ist eine Schiene vorhanden, die einen Papierstreifen aufnehmen kann. Ein entsprechender Papierstreifen wird mit der Betriebssystem-Software von NCR mitgeliefert. Er enthält den Wortlaut der etwa vom Be-

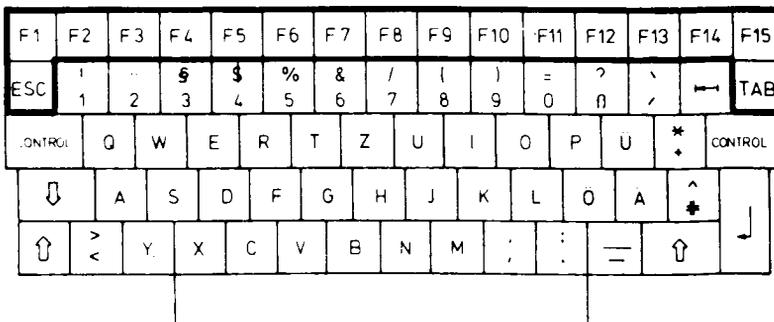


Abb. 3.14 Programmierbare Tasten

triebssystem bewirkten Vorprogrammierung der jeweiligen Taste. Sie können bei Neuprogrammierung der Funktionstasten diesen Streifen ergänzen oder ersetzen.

## DIE NUMERISCHE TASTATUR

Das numerische 10-Tasten-Feld ermöglicht eine zügige Eingabe numerischer Daten. Zusätzlich enthält die numerische Tastatur folgende Tasten:

- Fünf programmierbare Tasten (F16 bis F20)
- Fünf Tasten für die Schreibmarke (Cursor)
- Tasten für die arithmetischen Funktionen Addition, Subtraktion, Multiplikation, Division und Löschen
- Null-Taste
- Doppelnull-Taste
- Dezimal-Komma
- Abschlußtaste

### Programmierbare Tasten

Die fünf Tasten F16 bis F20 (Abbildung 3.15) sind genauso einsetzbar wie die Tasten F1 bis F15 der alphanumerischen Tastatur. Auch hier können Sie über der Tastenreihe eine Beschriftung anbringen.

F16	F17	F18	F19	F20
↖	←	↓	↑	→
CLR	7	8	9	/
-	4	5	6	*
+	1	2	3	⏏
0	00	.		

Abb. 3.15 Programmierbare Tasten

### Positionszeiger-Tasten

Bei Betätigung einer dieser Tasten (Abbildung 3.16) wird der Positionszeiger auf dem Bildschirm entsprechend der Pfeilrichtung der Taste bewegt. Bitte beachten Sie hierzu auch die detaillierten Hinweise in der jeweiligen Software-Dokumentation.

F16	F17	F18	F19	F20
↖	←	↓	↑	→
CLR	7	8	9	/
-	4	5	6	*
+	1	2	3	⏴
0	00	.		

Abb. 3.16 Bildschirm-Positionszeiger

### Arithmetische Funktionen

Diese Tasten (Abbildung 3.17) dienen den üblichen Grundrechnungsarten Addieren, Subtrahieren, Multiplizieren und Dividieren sowie der LösCHFunktion. Die Anordnung der Zifferntasten entspricht dem Tastenfeld vieler Taschenrechner.

F16	F17	F18	F19	F20
↖	←	↓	↑	→
CLR	7	8	9	/
-	4	5	6	*
+	1	2	3	⏴
0	00	.		

Abb. 3.17 Arithmetik-Tasten

### Die Doppelnull-Taste

F16	F17	F18	F19	F20
↖	←	↓	↑	→
CLR	7	8	9	/
-	4	5	6	*
+	1	2	3	⏴
0	00	.		

Abb. 3.18 Die Doppelnull-Taste

Zur schnelleren Eingabe ganzer Reihen von Nullen benutzen Sie die Doppelnul Taste (Abbildung 3.18). Gleichzeitiges Drücken von Null und Doppelnul ergibt 3 Nullen.

**Das Dezimal-Komma**

Diese Taste (Abbildung 3.19) verwenden Sie zum Setzen des Dezimal-Kommas in einer Ziffernfolge. Wenn Sie wollen, daß von Computer und Drucker statt des Kommas (,) ein Dezimalpunkt (.) angezeigt wird, müssen Sie Komma (,) bei gedrückter CONTROL-Taste betätigen. Nun wird bei Benutzung der Kommataste ein Dezimalpunkt angezeigt oder gedruckt. Die Rückstellung auf Normalbetrieb erfolgt durch erneutes Betätigen von CONTROL mit Komma oder durch Ausschalten des Systems.

F16	F17	F18	F19	F20
↖	←	↓	↑	→
CLR	7	8	9	/
-	4	5	6	*
+	1	2	3	⏏
0	00	,		

Abb. 3.19 Das Dezimal-Komma

**Die Abschluß-Taste**

Die Funktion dieser Taste (Abbildung 3.20) und die Funktion der Abschluß-taste in der alphanumerischen Tastatur sind identisch.

F16	F17	F18	F19	F20
↖	←	↓	↑	→
CLR	7	8	9	/
-	4	5	6	*
+	1	2	3	⏏
0	00	,		

Abb. 3.20 Die Abschluß-taste

## “KALTSTART”

Sie werden besonders bei der Entwicklung Ihrer eigenen Programme manchmal in einer Lage sein, in der das Aus- und ein erneutes Einschalten Ihres Computers zweckmäßig ist. In einem solchen Fall erübrigt sich das Ein- und Ausschalten mit dem dazugehörigen Neueinlegen Ihrer Diskette, wenn Sie die Funktionstaste F20 bei gedrückter CONTROL-Taste betätigen. Ihr NCR DECISION MATE V verhält sich dann, als ob Sie den Ein-/Aus-Schalter betätigt hätten.

## WIE KANN ICH SOFTWARE LADEN?

Sie haben nun gelernt, wie Sie das System einschalten und eine Diskette einlegen können. Die wichtigsten Schritte sind nachstehend noch einmal zusammengefaßt:

Schalten Sie Ihren Computer ein (Ein-/Aus-Schalter auf “1”) und warten Sie ein paar Sekunden. Der Computer meldet sich mit

DISK A: NOT READY <CR>

Schieben Sie eine Diskette, die ein Ihrem Computer entsprechendes Betriebssystem (8-Bit bzw. 16-Bit) enthält, mit der Schreib-/Lese-Öffnung voraus und dem Aufkleber in Richtung Bildschirm zeigend in das Laufwerk A ein. (Dasselbe Laufwerk wird im UCSD-p-System als “4” bezeichnet.) Danach drehen Sie den Verschußhebel nach rechts.

Betätigen Sie die Abschlußtaste ↓.

Der Computer meldet sich mit einer Software-Mitteilung und dem Bereitschaftszeichen >. (Beim UCSD-p-System erscheint eine Befehlsauswahl in der ersten Zeile des Bildschirms).

Nach Beendigung Ihrer Arbeit mit dem Computer, sollten Sie den Ein-/Aus-Schalter erst nach Erscheinen des Bereitschaftszeichens betätigen.

## EINIGE PRAKTISCHE ÜBUNGEN

Wenn Sie das MS-DOS-Betriebssystem oder eines der CP/M-Betriebssysteme verwenden, können Sie den folgenden Übungen nachgehen, damit Sie mit den Funktionen der Tastatur Ihres NCR DECISION MATE V vertraut werden. (Diese Übungen eignen sich

nicht bei Verwendung des UCSD-p-Systems, da dieses bei der Eingabe bestimmter Einzelbuchstaben mit der Verarbeitung sofort beginnt.) Beim MS-DOS-Betriebssystem müssen Sie zunächst die Betriebssystem-Diskette kopieren. Dieser Kopiervorgang wird automatisch eingeleitet (siehe "MS-DOS Benutzer-Handbuch").

Nachdem das System-Bereitschaftszeichen erschienen ist, geben Sie einfach irgendeinen belanglosen Text ein, z.B.

A> irgendwelche Informationen, die nicht von Bedeutung sind

Drücken Sie anschließend die Abschlußtaste. Durch diese Eingabe können Sie weder Ihrem Computer noch Ihrer Software einen Schaden zufügen — der Computer weiß nämlich zunächst gar nicht, worüber Sie reden. Er meldet sich daher durch Anzeige des ersten eingegebenen Wortes und eines Fragezeichens.

IRGENDWELCHE?

A>

Nun geben Sie unter Benutzung der in diesem Abschnitt beschriebenen Tasten weitere Daten ein. Schalten Sie auf Großbuchstaben um:

A> DATEN IN GROSSBUCHSTABEN EINGEBEN ↵  
DATEN?

A> und dann auf kleinbuchstaben zurückschalten ↵  
und?

A>

Falls Sie dabei Fehler machen, korrigieren Sie diese mit der Rücktaste; diese löscht ein einzelnes Zeichen oder auch alle Zeichen einer Zeile. So werden Sie schon bald mit der Tastatur vertraut sein. Bevor Sie mit der Arbeit am Computer beginnen, lesen Sie bitte sorgfältig die Einführung in dem zum Betriebssystem gehörenden Handbuch. Dort lernen Sie, wie Sie Disketten formatieren, Sicherungskopien von wichtigen Disketten anfertigen und das System für Ihre Anwendungen vorbereiten können. Lernen Sie, den Computer für Sie arbeiten zu lassen.

(

)

)

## WAS NUN?

### ERKENNEN VON STÖRUNGEN

Dieser Abschnitt erläutert Ihnen, wie Sie eventuell auftretende Störungen erkennen und beseitigen können. Schon mit ein paar Minuten Zeit, die Sie für eine selbständige Fehlersuche aufwenden, läßt sich oft das Herbeirufen des technischen Kundendienstes vermeiden.

Falls Sie doch einmal Hilfe in Anspruch nehmen müssen, so ist es von Vorteil, wenn Sie den Fehler genau beschreiben können. Der Kundendiensttechniker weiß dann, welche Prüfgeräte und Ersatzteile er mitbringen muß. Am besten, Sie notieren den aufgetretenen Fehler und die entsprechenden Fehleranzeigen in Stichworten nach folgendem Schema:

- Wann ist der Fehler aufgetreten?
- Bei welchem Betriebssystem und Anwenderprogramm?
- Genauer Wortlaut der Meldung am Bildschirm.
- Ob und ggf. welche Leuchtdioden an der Rückseite des Computers nicht ordnungsgemäß erlöschen.
- Leuchtanzeige des Diagnostik-Moduls (K220), falls vorhanden.

### SERVICE-VEREINBARUNG

Benötigen Sie die Hilfe eines gut geschulten Kundendienstes, so kann Ihnen NCR ein erfahrenes Team von Technikern anbieten. Etwa 1200 über die ganze Erde verteilte Geschäftsstellen stehen zur Verfügung, von denen sich sicher auch eine in Ihrer Nähe befindet. Sie können dabei unter folgenden verschiedenen Wartungsaufträgen auswählen. Ihre nächstgelegene NCR Geschäftsstelle berät Sie gerne.

- NCR Voll-Service — bietet den Vorteil einer alles umfassenden Systembetreuung für eine günstige Jahresgebühr. Dieser Service durch speziell geschultes Personal steht Ihnen während der üblichen Geschäftszeiten zur Verfügung und schließt gewöhnlich die Anfahrts-, Reparatur- und Ersatzteilkosten mit ein.

- Zeit- und Material Service — hier zahlen Sie bei einem Kundendienstbesuch die anfallende Zeit sowie die Ersatzteile.
- Reparatur im Service-Center — Sie bringen Ihren Computer zur nächstgelegenen NCR-Geschäftsstelle. Nach durchgeführter Reparatur wird Ihnen das Gerät von NCR angeliefert, oder Sie erhalten Nachricht, daß es zur Abholung bereit ist.

Die Wartung des Computers kann auch auf andere Weise abgewickelt werden

- Haben Sie Ihren Computer nicht direkt von NCR, sondern über einen Vertriebsbeauftragten erworben, so können Sie auch direkt mit diesem Händler eine entsprechende Servicevereinbarung treffen.
- Mit Hilfe des von NCR angebotenen Service-Handbuchs können technisch geschulte Personen in Ihrem Unternehmen den Computer selbst reparieren. Jedoch sollten solche Arbeiten nur von Technikern durchgeführt werden, die mit der Wartung von elektronischen Geräten sowie dem Umgang mit gedruckten Schaltungen und integrierten Schaltkreisen ausreichend vertraut sind.

Zur Unterstützung bietet Ihnen NCR Schulungskurse an. Näheres hierzu erfahren Sie bei Ihrer nächstgelegenen NCR-Geschäftsstelle.

## BESEITIGUNG VON FEHLERN

In den folgenden Tabellen finden Sie nun einige typische Beispiele von Fehlermöglichkeiten, ihre Ursachen sowie Hinweise zu ihrer Behebung.

Fehler	Mögliche Ursachen	Abhilfe
Betriebs-Kontrolllampe leuchtet nicht auf	<p>Netz-kabel nicht am Computer angeschlossen</p> <p>Netz-kabel nicht an Wandsteckdose angeschlossen</p> <p>Keine Spannung an der Wandsteckdose</p> <p>Computer nicht eingeschaltet</p> <p>Fehlerhaftes Netz-kabel</p> <p>Interner Fehler im Computer</p>	<p>Netz-kabel am Computer anschließen</p> <p>Netzstecker in Wandsteckdose stecken</p> <p>Benachrichtigen Sie den Haus-elektriker</p> <p>Ein-/Aus-Schalter betätigen</p> <p>Netz-kabel austauschen</p> <p>Benachrichtigen Sie den Service-Techniker</p>
Betriebs-system-meldung erscheint beim Laden des Betriebs-systems nicht am Bildschirm	<p>Kontrast-/Helligkeitsregler nicht richtig eingestellt</p> <p>Diskette mit dem Betriebs-system falsch eingelegt</p> <p>Falsche Diskette</p> <p>Systemdiskette in Laufwerk B (UCSD-p-System:5) bei Systemen mit zwei Disketten-Laufwerken. Richtig: Laufwerk A (UCSD-p-System:4)</p> <p>Systemdiskette beschädigt</p> <p>Interner Fehler im Computer</p>	<p>Regler-einstellung korrigieren</p> <p>Diskette im Laufwerk A richtig einlegen. Siehe Kapitel "Bedienung des Computers"</p> <p>Richtige Diskette auswählen</p> <p>Legen Sie die Systemdiskette in das richtige Laufwerk ein</p> <p>Verwenden Sie Ihre Sicherungskopie, nachdem Sie eine weitere Sicherungskopie angefertigt haben</p> <p>Benachrichtigen Sie den Service-Techniker</p>
Dateneingabe über Tastatur nicht möglich	<p>Umschalt-Feststelltaste ist falsch eingestellt</p> <p>Tastatur-Eingabespeicher voll</p> <p>Tastaturkabel nicht am Computer angeschlossen</p> <p>Interner Fehler in Tastatur oder Computer</p>	<p>Taste drücken</p> <p>Warten Sie, bis die laufende Verarbeitung abgeschlossen ist</p> <p>Kabel anschließen</p> <p>Benachrichtigen Sie den Service-Techniker</p>

Fehler	Mögliche Ursache	Abhilfe
Datenübertragung, auf Diskette wird nicht ausgeführt	<p>Diskette nicht im richtigen Laufwerk</p> <p>Diskette falsch eingelegt</p> <p>Schreibschutzaukleber nicht entfernt</p> <p>Diskette beschädigt</p> <p>Diskette nicht formatiert</p> <p>Interner Fehler im Computer bzw. Diskettenlaufwerk</p>	<p>Diskette in das richtige Laufwerk einlegen</p> <p>Diskette richtig einlegen (siehe Kapitel "Bedienung des Computers")</p> <p>Aufkleber entfernen, wenn der Schreibschutz aufgehoben werden soll</p> <p>Verwenden Sie Ihre Sicherungskopie</p> <p>Diskette formatieren</p> <p>Benachrichtigen Sie den Service-Techniker</p>
Diskette kann nicht gelesen werden	<p>Diskette nicht im richtigen Laufwerk</p> <p>Diskette falsch eingelegt</p> <p>Diskette beschädigt</p> <p>Interner Fehler im Computer oder Diskettenlaufwerk</p>	<p>Diskette in das richtige Laufwerk einlegen</p> <p>Diskette richtig einlegen (siehe Kapitel "Bedienung des Computers")</p> <p>Verwenden Sie Ihre Sicherungskopie</p> <p>Benachrichtigen Sie den Service-Techniker</p>
Undefinierter Fehler – keine weitere Verarbeitung möglich	<p>Probleme mit der Betriebssoftware und/oder dem Anwendungsprogramm</p> <p>Vorübergehende Schwierigkeiten mit der Netzspannungsversorgung</p> <p>Interner Fehler im Computer</p>	<p>Betriebssystem und Anwendungsprogramm neu laden</p> <p>Computer abschalten. Verwenden Sie die Sicherungskopie des Betriebssystems und laden Sie diese sowie das Anwendungsprogramm erneut</p> <p>Benachrichtigen Sie den Hauselektriker</p> <p>Benachrichtigen Sie den Service-Techniker</p>

( , ( )

## PRAKTISCHE HINWEISE

Mit "Praktische Hinweise" wollen wir Ihnen einige nützliche Empfehlungen für Ihre tägliche Arbeit mit dem Computer geben. Die Hinweise beziehen sich auf den Arbeitsplatz und dessen Umgebung sowie die Arbeitsweise mit dem Computer. So liegen z. B. die Ursachen für Schwierigkeiten mit Disketten meist in deren falscher Handhabung. Deshalb ist diesem Punkt besondere Aufmerksamkeit zu widmen. Auch wenn Sie einmal Ihren Computer an einen anderen Ort verlegen wollen, beachten Sie bitte die Hinweise am Ende dieses Kapitels.

### AUFSTELLEN DES COMPUTERS

Bei der Auswahl des Aufstellungsortes für Ihren Computer sollten Sie folgende Hinweise berücksichtigen:

- Wählen Sie einen Platz abseits von zu starkem Bürobetrieb, an dem Sie ungestört arbeiten können.
- Vermeiden Sie Räume mit extremer Temperatur und Feuchtigkeit sowie starker Staubentwicklung. Sie sollten Ihren Computer von unmittelbarer Sonneneinstrahlung und starken Heizungen fernhalten.
- Installieren Sie genügend Steckdosen für den Computer und die etwa vorhandenen Peripheriegeräte.
- Sorgen Sie für ausreichenden Ablageplatz für Disketten und ihre Hüllen, Druckpapier usw.
- Legen Sie Netz- und Verbindungskabel so, daß sie nicht behindern.
- Wenn Sie Ihren Computer in der Nähe von Geräten betreiben, die starke Magnetfelder erzeugen, kann sich die Bildqualität verschlechtern. Um dies zu vermeiden, sollte der Computer von der Störquelle entfernt werden.

## ARBEITEN MIT DEM COMPUTER

Die folgenden Hinweise helfen Ihnen, die tägliche Arbeit mit Ihrem NCR DECISION MATE V so effektiv wie möglich zu gestalten:

- Organisieren Sie die Arbeit in folgerichtigen Abläufen. Schaffen Sie genügend Platz zur Ablage der Arbeitsmittel, und trennen Sie erledigte von nichterledigter Arbeit.
- Schalten Sie den Computer nie aus, bevor das Bereitschaftszeichen des Systems am Bildschirm erscheint, sonst kann es vorkommen, daß Daten gelöscht werden.
- Legen Sie Ihre Disketten sofort nach Gebrauch wieder in das Ablagefach zurück.
- Erstellen Sie in regelmäßigen Zeitabständen Sicherungskopien und bewahren diese getrennt auf.
- Am Ende der Tagesarbeit ziehen Sie den Netzstecker aus der Wandsteckdose.

## BEHANDLUNG DER DISKETTEN

Sie können eine sehr hohe Lebensdauer von Ihren Disketten erwarten, vorausgesetzt, daß Sie sorgfältig mit ihnen umgehen:

- Für die Beschriftung der Diskettenaufkleber verwenden Sie Filzstifte. Abriebteilchen von Bleistiften, Radiergummis, Fettstiften oder Rückstände von Kugelschreibern könnten auf die Magnetschicht der Diskette gelangen und diese beschädigen. Am zweckmäßigsten beschriften Sie einen Aufkleber, bevor Sie ihn auf die Diskette kleben. Falls dies nicht mehr möglich ist, sollten Sie die Diskette wieder in ihre Schutzhülle stecken, bevor Sie den Aufkleber beschriften.
- Legen Sie die Diskette weder in grelles Sonnenlicht noch in die Nähe von magnetischen Gegenständen.
- Bringen Sie an der Hülle keine Büroklammern oder Gummiringe an.
- Berühren Sie nicht die Diskettenoberfläche. Versuchen Sie auch nicht, die Oberflächen zu reinigen. Dies ist nicht notwendig, weil sich die Diskette während des Laufes automatisch säubert.
- Nach Herausnahme aus dem Laufwerk stecken Sie die Diskette immer sofort in ihre Schutzhülle.

## ORTSWECHSEL DES COMPUTERS

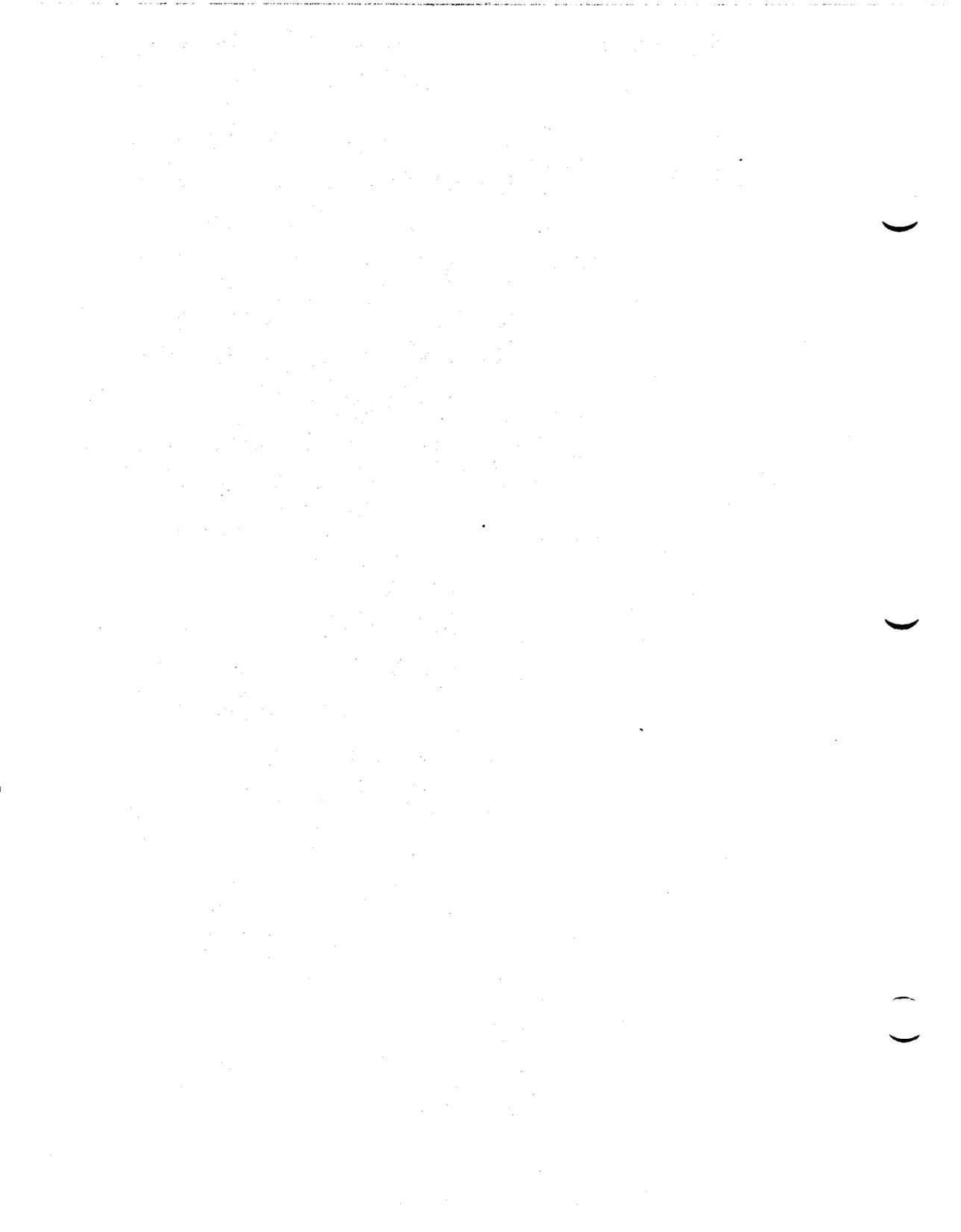
Ihr NCR DECISION MATE V ist so konzipiert, daß bei einem Standortwechsel keine außergewöhnlichen Vorkehrungen notwendig sind. Zur Vermeidung möglicher Beschädigungen sollten Sie jedoch folgende Empfehlungen beachten:

- Auch bei Transport über kurze Entfernungen sollten Sie Ihren Computer und die etwa angeschlossenen Zusatzgeräte vorher ausschalten und die Verbindungskabel lösen.
- Falls Sie den Computer öfters über größere Entfernungen transportieren oder gar mit der Bahn bzw. Post verschicken wollen, ist es zweckmäßig, die Originalverpackung aufzubewahren, um sie dann für den Transport zu benutzen. Diese Maßnahme gilt auch für die Einlegscheibe aus Karton, die sich bei der Lieferung in jedem Diskettenlaufwerk befand.

## REINIGUNG DES COMPUTERS

Ihr NCR DECISION MATE V erfordert keine besonderen Reinigungsmaßnahmen. Gelegentliches Abstauben mit einem weichen Tuch ist aber zu empfehlen. Achten Sie besonders auf den Bildschirm und die Tastatur. Hartnäckige Flecken sollen nur mit einem weichen, mit milder Seifenlösung angefeuchteten Lappen entfernt werden.

Während der Reinigung muß der Computer unbedingt ausgeschaltet sein.



ZU BEACHTEN BEI  
MASCHINEN MIT  
FARBILDSCHIRM !

ZULASSUNGSSCHEIN Nr. By 277/83/R6

Gemäß § 7 Abs. 2 der Röntgenverordnung (ROV) vom 01.03.1973 (BGBl. I S. 173), geändert durch Verordnung vom 13.10.1976 (BGBl. I S. 2905), wird die Bauart der nachstehend beschriebenen Röntgenstrahlröhre zugelassen.

Gegenstand der Zulassung: Störstrahlröhre  
 Hersteller und Zulassungsinhaber: NCR GmbH  
 8900 Augsburg 1  
 Firmenbezeichnung und Typ: NCR DDM V Personal Computer  
 C-3273-12 - (65-7)  
 Bildröhre: Typ AT 1298 ZAB 22 A-TC 72  
 der Firma Mitsubishi

Betriebsbedingungen:  
 Hochspannung max. 25 kV  
 Strahlstrom max. 0,3 mA

PTB-Prüfungsschein: Nr. 6 62-S 180 vom 07.12.1983  
 "Die freigelegten Buchstaben- und Zahlenkombinationen betreffen nicht den Strahlenschutz."

Die Zulassung ist befristet bis zum 16.01.1984

Die in diesem Zulassungsschein genannten Geräte können gemäß § 8 Abs. 2 ROV auch nach Ablauf dieser Frist weiterhin betrieben werden.

Wesentliche Merkmale für den Strahlenschutz sind:

a) die Bauart der Bildröhre und  
 b) die Hochspannungserzeugung und -stabilisierung dienenden Baugruppen des Chassis.

II.

Die Physikalisch-Technische Bundesanstalt in Braunschweig hat mit dem Prüfungschein Nr. 6 62-S 180 vom 07.12.1983 bestätigt, daß bei den hier beschriebenen Störstrahlröhren die Vorschriften über die Bauart von Störstrahlröhren nach Anlage II Abschnitt 4 ROV erfüllt sind

Der Computer NCR DECISION MATE V entspricht gemäß der nachstehend abgedruckten Zulassung den gesetzlichen Bestimmungen der Röntgen-Verordnung. Die gemessene Dosis-Leistung liegt weit unterhalb des gesetzlich zulässigen Grenzwerts. Daher bestehen keinerlei gesundheitliche Gefahren.

III.

Folgende Auflagen werden festgesetzt:

- Die Geräte müssen vom Zulassungsinhaber einer Stückprüfung daraufhin unterzogen werden sein, ob sie bezüglich der für den Strahlenschutz wesentlichen Merkmale der Bauartzulassung entsprechen.  
 Die Prüfung wird umlassen:  
 1. bei Geräten an jedem einzelnen Gerät und Messung der Dosisleistung nach näherer Angabe der Zulassungsbehörde.

Die Ergebnisse der Dosisleistungsmessung sind, nach Herstellungsnummern der Geräte zugeordnet, aufzuzeichnen und der Zulassungsbehörde auf Verlangen einzureichen.

Der Zulassungsbehörde ist berechtigt, einzelne Geräte nach eigener Auswahl anzufordern, um die für den Strahlenschutz wesentlichen Merkmale zu überprüfen oder überprüfen zu lassen. Die Geräte sind auf Verlangen im Originalzustand anzuliefern und angemessene Zeit zur Verfügung zu stellen.

- Die Herstellung und die Stückprüfung sind durch einen vom der Zulassungsbehörde bestimmten Sachverständigen überwachen zu lassen.

- Die Geräte sind auf der Rückwand mit dem Kennzeichen

By 277/83/R6

zu versehen sowie mit einem Hinweis folgenden Mindestinhalts

"Die in diesem Gerät entstehende Röntgenstrahlung ist durch Abschirmung geschützt. Betriebsspannung maximal 25 kV."

- Jedem Erwerber eines Gerätes ist ein Abdruck des Zulassungsscheins auszubehändigen, auf dem das Ergebnis der Stückprüfung/Auflage 1 bestätigt sein muß.

Jedem Gerät ist ferner eine Betriebsanleitung beizufügen, die ebenfalls den in Auflage 3 genannten Hinweis enthält.

Zur Vermeidung eines eventuell unzulässigen Anstiegs der Röntgenstrahlung infolge unsachgemäßer Eingriffe dürfen Justagen und Reparaturen nur von hierfür ausgebildetem Personal ausgeführt werden.

IV.

Darüber hinaus gilt für den Benutzer des Gerätes:

Unsachgemäße Eingriffe, insbesondere Veränderungen der Hochspannung oder Auswechseln der Bildröhre, können dazu führen, daß Röntgenstrahlung in erheblicher Stärke auftritt. Ein so verändertes Gerät entspricht nicht mehr dieser Zulassung und darf nicht betrieben werden.

München, 16.01.1984

Bayerisches Landesamt für Umweltschutz

I.A.



*Henckly*  
 Dr. Henseler  
 1. Lt. Reg. Direktor

# WICHTIGER HINWEIS

NCR 3273 (DECISION MATE V)

Bei Betrieb dieses Gerätes innerhalb der Bundesrepublik Deutschland sind die Funkschutzvorschriften zu beachten. Bitte füllen Sie den untenstehenden Antrag aus und senden ihn an das örtliche Fernmeldeamt/Funkamt.

➔ **Antrag**, beim örtlichen Fernmeldeamt/Funkamt einreichen.

➤ **Bescheinigung (s. Rückseite)**, verbleibt beim Anwender.

## Antrag auf Genehmigung zum Betrieb eines Hochfrequenzgerätes

Antragsteller (Vor- und Zuname, Firma)

Straße, Hausnummer, Postleitzahl, Ort

Hiermit beantrage ich die Genehmigung zum Betrieb des nachstehend gekennzeichneten Hochfrequenzgerätes, dem

die FTZ-Serienprüfnummer

C-004/83

vom Fernmeldetechnischen Zentralamt erteilt worden ist.

**Kennzeichnung des Hochfrequenzgerätes**

Gerätebezeichnung (Typangabe)

NCR 3273 (Decision Mate V)

Hersteller, Importfirma

NCR GmbH, Ulmer Straße 160, 8900 Augsburg

Betriebs-Nennfrequenz  
siehe C-004/83

Hochfrequenz-Nennleistung  
entfällt W/kW

**Verwendungszweck** (z. B. Kurzwellen-Therapie, dielektrische oder induktive Erwärmung von Werkstoffen, Ultraschall-Anwendung)  
Datenverarbeitung

**Betriebsort** (Aufstellungsort des Hochfrequenzgerätes, Straße, Hausnummer, Stockwerk, Postleitzahl, Ort)

(Unterschrift des Antragstellers)





# Allgemeine Lizenzbedingungen der NCR GmbH

1. Der Lizenznehmer verpflichtet sich, die Lizenzprogramme vor unbefugtem Gebrauch zu schützen. Das/Die Lizenzprogramm(e) dürfen, soweit dies technisch möglich ist, nur in maschinenlesbarer oder gedruckter Form kopiert werden, wenn die Kopie zur Modifizierung oder als Sicherheitskopie benötigt wird. Jede Kopie muß den Vermerk tragen, daß die NCR GmbH (nachfolgend kurz NCR genannt) das Urheberrecht am Lizenzprogramm besitzt.
2. Der Lizenznehmer darf das/die Lizenzprogramm(e) modifizieren oder zur Benutzung mit anderen Programmen verbinden, wenn deren Benutzung auf dem Computersystem NCR Decision Mate V erfolgt. Die modifizierten oder verbundenen Programme unterliegen diesen Allgemeinen Lizenzbedingungen der NCR. Die Modifikationen und Programmverbindungen müssen die Programm-Nummer des Ursprungsprogrammes und den Vermerk aufweisen, daß die NCR das Urheberrecht daran besitzt.
3. Die Benutzer-Lizenz sowie die sonstigen Rechte und Pflichten aus diesem Vertragsverhältnis dürfen vom Lizenznehmer nur mit vorheriger schriftlicher Zustimmung der NCR an Dritte übertragen werden.
4. Weist der Lizenznehmer innerhalb einer Frist von 6 Monaten, gerechnet nach dem Datum seiner Programmempfangsbestätigung nach, daß das (die) Lizenzprogramm(e) fehlerhaft ist (sind), hat die NCR wahlweise das Recht, ein anderes, gleiches Programm zu liefern, den Fehler nachzubessern oder dem Lizenznehmer zuzugestehen, daß dieser vom Vertrag über das fehlerhafte Programm zurücktritt. Im letzteren Fall werden dem Lizenznehmer die bereits bezahlten Lizenzgebühren gegen Rückgabe der (des) Lizenzprogramme(s) zurückerstattet. Soweit gesetzlich zulässig, sind damit alle weitergehenden Gewährleistungs- und Ersatzansprüche ausgeschlossen.
5. Die NCR übernimmt keine Garantie dafür, daß das (die) Lizenzprogramm(e) den Vorstellungen und Anforderungen des Lizenznehmers entsprechen.
6. Die Benutzer-Lizenz ist zeitlich beschränkt auf die Dauer der Benutzung des in Ziffer 2 genannten NCR-Computersystems. Das (Die) Lizenzprogramm(e) nebst Kopien, Modifikationen oder Programmverbindungen ist (sind) sofort der NCR zurückzugeben oder zu vernichten, sobald der Lizenznehmer dieses Computersystem ständig nicht mehr benutzt.  
Der Lizenznehmer ist verpflichtet, der NCR innerhalb von 30 Tagen nach endgültiger Einstellung der Benutzung schriftlich mitzuteilen, seit wann das NCR-Computersystem nicht mehr verwendet wird und daß er entweder das (die) Lizenzprogramme(e), Kopien, Modifikationen und/oder Programmverbindungen an die NCR zurückgeben wird oder vernichtet hat.
7. Verstößt der Lizenznehmer gegen die vorstehenden Bedingungen, hat die NCR das Recht, den Vertrag über die Benutzer-Lizenz ohne Einhaltung einer Frist zu kündigen und vom Lizenznehmer die sofortige Rückgabe der (des) Lizenzprogramme(s) nebst Kopien, Modifikationen oder Programmverbindungen zu verlangen, ohne zur Erstattung bereits bezahlter Lizenzgebühren verpflichtet zu sein.
8. Für evtl. Streitigkeiten aus diesem Vertragsverhältnis ist Augsburg Gerichtsstand, es sei denn, daß ein ausschließlicher Gerichtsstand besteht.

)

)

)

**NCR**

---

NCR DECISION MATE V

**CP/M<sup>®</sup>-80**

CP/M<sup>®</sup> ist ein eingetragenes Warenzeichen von Digital Research.  
Intel ist ein eingetragenes Warenzeichen der Intel Corporation.  
Z80 ist ein eingetragenes Warenzeichen von Zilog, Inc.  
ADM-3A und ADM31 sind Warenzeichen von Lear Siegler.  
Ladder und Catchum sind Warenzeichen von Yahoo Software.  
MS-BASIC ist ein eingetragenes Warenzeichen von Microsoft, Inc.

Copyright © 1984 by NCR Corporation  
Dayton, Ohio  
All Rights Reserved  
Printed in the Federal Republic of Germany

### **3. Auflage, Mai 1984**

NCR ist ständig bemüht, die Produkte im Zuge der Entwicklung von Technologie, Bauteilen, Soft- und Firmware dem neuesten Stand anzupassen. NCR behält sich deshalb das Recht vor, Spezifikationen ohne vorherige Ankündigung zu ändern.

Nicht alle hier beschriebenen Leistungen werden von NCR in allen Teilen der Welt vertrieben. Nähere Informationen bezüglich eventueller Einschränkungen oder auch Erweiterungen sowie den aktuellen Stand erfahren Sie von Ihrem Händler oder der nächstgelegenen NCR-Geschäftsstelle.

# BENUTZUNG DIESES HANDBUCHS

CP/M (Control-Programm für Mikrocomputer) ist ein vielfach eingesetztes Betriebssystem, entwickelt von der Digital Research, Inc., USA. Die Hauptursache seiner weltweiten Popularität liegt in seiner weitgehenden Unabhängigkeit von der Hardware. Das heißt, es ist mit nur kleinen, einfachen Änderungen für die verschiedensten Computersysteme verwendbar. Solche Änderungen betreffen die Schnittstellen, die für die Anpassung an die Hardware-Eigenschaften des Computers sorgen.

Dieses Handbuch beschreibt CP/M und enthält Änderungsinformationen für NCR DECISION MATE V. Gegliedert in 7 Kapiteln, führt es sowohl den Anfänger als auch den in Software schon Geübteren mit detaillierten Ausführungen in das Betriebssystem ein.

Das einleitende Kapitel, "Die ersten Schritte," erklärt, wie Sie eine Sicherungskopie der Software erstellen können. Sie werden gleichzeitig lernen, wie Sie die System-Konfiguration festlegen, d. h., Ihren Computer in seine Geräte-Umgebung einweisen können. Die Bedienung Ihres Computers wird vor allem durch die programmierbaren Funktionstasten erleichtert. Dieses Kapitel enthält auch nützliche Informationen zu der Anschaffung von Software und der Benutzung von EXCHANGE. (EXCHANGE ist ein "Dienstprogramm", das sich bereits auf Ihrer CP/M-Diskette befindet. Mit EXCHANGE können Sie beinahe jede Software in Ihrem Computer einsetzen.) Der letzte Abschnitt dieses einleitenden Kapitels zeigt Ihnen, wie Sie die Anwender-Software einsetzen können, die Ihre Systemdiskette enthält.

Kapitel 2 und 3 informieren Sie über das Betriebssystem, seine Befehle und Anwendungen, sowie die Handhabung des Editor-Programms. Falls Sie keine eigenen Programme für NCR DECISION MATE V schreiben wollen, geben Ihnen diese Kapitel alle notwendigen Informationen für den Einsatz von Software in Ihrem Computer. Bereits erfahrene CP/M-Anwender werden hier einiges bezüglich fünf neuer Dienstprogramme, FORMAT, CONFIG, DISCIT, BACKUP und EXCHANGE hinzulernen können.

In den weiteren Kapiteln können sich vor allem Programmierer über Schnittstellen, "Debugging" (Testhilfen) und andere programmierungs-abhängige Besonderheiten informieren.

Die verschiedenen Abschnitte im Anhang sollen allen Anwendern nützliche Referenzinformation für den Betrieb des Computers geben. Diese Abschnitte enthalten spezielle Angaben für die Anpassung und

Handhabung von Anwenderprogrammen, eine ASCII-Code-Tabelle, eine Übersetzungstabelle für die Tastatur der jeweiligen Landessprache sowie eine Auflistung von Systemmeldungen mit Erläuterungen und korrigierenden Maßnahmen.

Während des Lesens dieses Handbuchs werden Sie feststellen, daß CP/M einfach zu bedienen und Ihr Computer sehr "benutzerfreundlich" ist. Nach einigen praktischen Übungen sind Sie bereits bestens vertraut mit CP/M und Ihrem NCR DECISION MATE V.

## INHALTSVERZEICHNIS

### DIE ERSTEN SCHRITTE

1.1	Vorwort	1- 1
1.2	Grundbegriffe der Software	1- 4
1.2.1	Bereitschaftszeichen (Prompts)	1- 5
1.2.2	Laden des Betriebssystems	1- 5
1.2.3	CP/M-Kommandos	1- 6
1.2.4	Umschalten auf ein anderes Laufwerk	1- 8

### ÜBUNGEN

1.3	Laden des Betriebssystems	1- 9
1.3.1	Anzeigen des Platten-Dateiverzeichnisses	1-10
1.4	Die ersten Schritte - Nur für ein System mit zwei Disketten-Laufwerken	1-11
1.4.1	Formatieren einer neuen Diskette	1-11
1.4.2	Kopieren der Systemdiskette	1-13
1.4.3	Bestimmung der System-Konfiguration	1-14
1.5	Die ersten Schritte - Nur für ein System mit einem Disketten-Laufwerk und einem oder mehreren Festplatten-Laufwerken	1-16
1.5.1	Bestimmung der System-Konfiguration	1-17
1.5.2	Formatieren einer neuen Diskette	1-18
1.5.3	Formatieren einer Festplatte	1-19
1.5.4	Kopieren der Systemdiskette	1-21
1.5.5	Abschluß der Konfigurationsarbeiten	1-23

### ANWENDUNGS-PROGRAMME

LADDER	1-27
CATCHUM	1-27
DEMO5	1-28
MUSIC	1-28
VEGAS	1-29
VEGAS, Eingabe von Daten	1-30
VEGAS, Erstellen eines Diagramms	1-34
VEGAS, Änderung von Daten	1-39
VEGAS, Grafik-Beispiele	1-42
VEGAS, Fehlermeldungen	1-49

## WIE BEKOMME ICH SOFTWARE?

- 1.6 Verwendung des EXCHANGE-Dienstprogrammes . . . . 1-52
  - 1.6.1 Aufruf von EXCHANGE in einem System mit zwei Disketten-Laufwerken . . . . . 1-52
  - 1.6.2 Aufruf von EXCHANGE in einem System mit einem Disketten-Laufwerk und einem oder mehreren Festplatten-Laufwerken . . . . . 1-55

## KOMMANDOS

- 2.1 Datei-Bezeichnungen . . . . . 2- 1
- 2.2 Residente Kommandos . . . . . 2- 3
  - 2.2.1 ERA afn . . . . . 2- 4
  - 2.2.2 DIR afn . . . . . 2- 5
  - 2.2.3 REN ufn1 = ufn2 . . . . . 2- 6
  - 2.2.4 SAVE n ufn . . . . . 2- 7
  - 2.2.5 TYPE ufn . . . . . 2- 7
  - 2.2.6 USER n . . . . . 2- 8
- 2.3 Zeilenweises Editieren und Ausgabesteuerung . . . . . 2- 8
- 2.4 Transiente Kommandos . . . . . 2- 9
  - 2.4.1 STAT . . . . . 2-11
  - 2.4.2 ASM ufn . . . . . 2-19
  - 2.4.3 LOAD ufn . . . . . 2-20
  - 2.4.4 PIP . . . . . 2-22
  - 2.4.5 ED ufn . . . . . 2-33
  - 2.4.6 SYSGEN . . . . . 2-35
  - 2.4.7 SUBMIT ufn parm # 1 . . . parm # n . . . . . 2-37
  - 2.4.8 DUMP ufn . . . . . 2-40
  - 2.4.9 FORMAT . . . . . 2-40
  - 2.4.10 CONFIG . . . . . 2-41
  - 2.4.11 EXCHANGE . . . . . 2-43
  - 2.4.12 DISCIT . . . . . 2-44
  - 2.4.13 BACKUP . . . . . 2-45
- 2.5 BDOS-Fehlermeldungen . . . . . 2-52

## ED

- 3.1 Einführung in ED . . . . . 3- 1
  - 3.1.1 Arbeitsweise von ED . . . . . 3- 1
  - 3.1.2 Anweisungen zur Übertragung von Text . . . . . 3- 3
  - 3.1.3 Organisation des Pufferspeichers . . . . . 3- 5
  - 3.1.4 Zeilennummern und ED-Anlauf . . . . . 3- 6
  - 3.1.5 Operationen im Puffer . . . . . 3- 7

3.1.6	Ketten von Steuerzeichen . . . . .	3- 9
3.1.7	Suchen und Ändern von Text . . . . .	3-11
3.1.8	Bibliotheksdateien . . . . .	3-15
3.1.9	Mehrfache Abarbeitung eines Befehls bzw. einer Befehlsfolge . . . . .	3-16
3.2	ED-Fehlermeldungen . . . . .	3-17
3.3	Steuerzeichen und Befehle . . . . .	3-18

## ASSEMBLER

4.1	Einführung . . . . .	4- 1
4.2	Programm-Format . . . . .	4- 3
4.3	Angabe der Operanden . . . . .	4- 4
4.3.1	Marken . . . . .	4- 5
4.3.2	Numerische Konstante . . . . .	4- 5
4.3.3	Reservierte Wörter . . . . .	4- 6
4.3.4	Zeichenfolgen als Konstante . . . . .	4- 7
4.3.5	Arithmetische und logische Operatoren . . . . .	4- 8
4.3.6	Vorrangstufung der Operatoren . . . . .	4-10
4.4	Assembler-Direktiven . . . . .	4-11
4.4.1	Die ORG-Direktive . . . . .	4-12
4.4.2	Die END-Direktive . . . . .	4-12
4.4.3	Die EQU-Direktive . . . . .	4-13
4.4.4	Die SET-Direktive . . . . .	4-14
4.4.5	Die IF- und ENDIF-Direktiven . . . . .	4-14
4.4.6	Die DB-Direktive . . . . .	4-16
4.4.7	Die DW-Direktive . . . . .	4-16
4.4.8	Die DS-Direktive . . . . .	4-17
4.5	Operations-Codes . . . . .	4-17
4.5.1	Sprungbefehle, Unterprogrammaufrufe und -Rücksprünge . . . . .	4-18
4.5.2	Instruktionen, die den im Operandenfeld angegebenen Wert übertragen . . . . .	4-20
4.5.3	Inkrement- und Dekrement-Instruktionen . . . . .	4-21
4.5.4	Instruktionen zum Übertragen von Daten . . . . .	4-22
4.5.5	Arithmetische und logische Befehle . . . . .	4-24
4.5.6	Steuer-Anweisungen . . . . .	4-26
4.6	Fehlermeldungen . . . . .	4-27
4.7	Beispiele . . . . .	4-29

## DYNAMISCHE TESTHILFE

5.1	Einführungen . . . . .	5- 1
5.2	DDT-Befehle . . . . .	5- 4

5.2.1	Der A-(Assembly)Befehl . . . . .	5- 4
5.2.2	Der D-(Display)Befehl . . . . .	5- 5
5.2.3	Der F-(Fill)Befehl . . . . .	5- 6
5.2.4	Der G-(Go)Befehl . . . . .	5- 6
5.2.5	Der I-(Input)Befehl . . . . .	5- 8
5.2.6	Der L-(List)Befehl . . . . .	5- 8
5.2.7	Der M-(Move)Befehl . . . . .	5- 9
5.2.8	Der R-(Read)Befehl . . . . .	5- 9
5.2.9	Der S-(Set)Befehl . . . . .	5-10
5.2.10	Der T-(Trace)Befehl . . . . .	5-11
5.2.11	Der U-(Untrace)Befehl . . . . .	5-12
5.2.12	Der X-(Examine)Befehl . . . . .	5-12
5.3	Anmerkungen zur Implementierung . . . . .	5-14
5.4	Ein Beispiel . . . . .	5-14

## SYSTEMVERBINDUNG

6.1	Einführung . . . . .	6- 1
6.2	Konventionen der Betriebssystem-Aufrufe . . . . .	6- 3
6.3	Beispiel für ein Datei-Kopierprogramm . . . . .	6-31
6.4	Ein einfaches Dienstprogramm zur hexadezimalen Ausgabe einer Datei (Dump) . . . . .	6-34
6.5	Ein Programmbeispiel für Direktzugriff . . . . .	6-38
6.6	Übersicht über die Systemfunktionen . . . . .	6-47

## PROGRAMMIERUNG

7.1	Platten-Organisation . . . . .	7- 1
7.2	Die Eintrittspunkte zu BIOS . . . . .	7- 2
7.3	Reservierte Felder in Speicherseite Null . . . . .	7- 9
7.4	Plattenparameter – Tabellen . . . . .	7-11

## ANHANG

Software-Installation . . . . .	A-1
Übersetzungs-/Umsetzungs-Tabellen . . . . .	B-1
Meldungen . . . . .	C-1
Begriffe . . . . .	D-1

## WICHTIGER HINWEIS

Nur für Besitzer eines NCR DECISION MATE V mit einem 96TPI Diskettenlaufwerk (96 Spuren pro Zoll) und einem oder mehreren Festplattenlaufwerken.

Zur Verwendung von Hilfs- oder Anwenderprogrammen, die auf eine Diskette mit Hilfe eines 48TPI-Laufwerks aufgezeichnet wurden, beachten Sie bitte folgende Vorgehensweise:

1. Legen Sie die CP/M-80 Originaldiskette, die Ihrem Betriebssystemordner beigelegt war, in Laufwerk A ein. (Die Arbeitskopie dieser Diskette kann hier nicht verwendet werden, da die Kopie auf Ihrem NCR DECISION MATE V als 96TPI-Diskette markiert wurde.)
2. Führen Sie einen "Warmstart" durch (CONTROL-C).
3. Geben Sie PIP ein. Der Asterisk (\*) wird angezeigt.
4. Entfernen Sie nun die Originaldiskette des Betriebssystems und legen Sie die Anwender- oder Hilfsprogrammdiskette in Laufwerk A ein. Geben Sie dann B:=A:Dateiname[V] ein, um den Inhalt der Diskette auf die Festplatte zu übertragen. Nach der Übertragung erscheint wieder der Asterisk (\*). Das kopierte Programm bzw. die kopierten Programme sind nun abrufbereit.
5. Nachdem Sie die Diskette wieder aus dem Laufwerk A genommen haben, legen Sie die Arbeitskopie der Betriebssystemdiskette ein und betätigen CONTROL-C ("C" bei gleichzeitig gedrückter CONTROL-Taste)

Benötigen Sie das Anwender- oder Hilfsprogramm auch auf Diskette, dann entfernen Sie die Betriebssystemdiskette und legen Sie die zu bespielende Diskette in Laufwerk A ein. Danach geben Sie B:PIP A:=B:dateiname[V] ein und betätigen die Eingabetaste.

Anmerkung: Erscheint während den gerade vorgenommenen Arbeiten die Meldung "FATAL ERROR (R/O/X)", kann dies auf Unverträglichkeit der Diskettenaufzeichnungsarten hinweisen. Dies geschieht zum Beispiel, wenn Sie die Arbeitskopie der Betriebssystemdiskette mit 96TPI verwenden, während die Anwenderprogrammdiskette eine Aufzeichnungsdichte von 48TPI (Spuren pro Zoll) besitzt.

Mit freundlichen Grüßen  
Ihre NCR GmbH

)

)

)

# EINLEITUNG

## 1.1 Vorwort

CP/M ist ein Monitor-Steuerprogramm für Mikrocomputer-Systeme, die Disketten oder Winchester Festplatten als Hintergrundspeicher benutzen. CP/M sorgt für die geeignete Umgebung für Programmkonstruktion, Speicherung und Editieren. Ein Assembler und Testhilfen gehören ebenfalls zu CP/M.

Dank eines umfassenden Dateiverwaltungspaketes bietet der CP/M-Monitor einen schnellen Zugriff zu Programmen. Das Dateisubsystem unterstützt eine namensorientierte Dateistruktur und erlaubt sowohl dynamische Speicherplatzzuweisung als auch sequentiellen und wahlfreien Dateizugriff. Bei Verwendung dieses Dateisystems kann eine große Anzahl von Programmen sowohl im Quellformat als auch in maschinenausführbarer Form gespeichert werden.

CP/M 2 ist ein Einplatz-Betriebssystem hoher Leistung, das tabelengesteuerte Techniken benutzt, um eine Rekonfiguration der Geräte-Umgebung zu ermöglichen. Dadurch können z. B. verschiedene Plattenspeicherkapazitäten berücksichtigt werden. Aufwärtskompatibilität von früheren Versionen des Release 1 ist gegeben bei Vermeidung aller wesentlichen Einschränkungen im Dateibereich. Die Möglichkeiten von CP/M beinhalten auch die Spezifizierung von einem bis zu 16 logischen Laufwerken, jedes mit einer Speicherkapazität von bis zu acht Megabyte. Eine einzelne Datei darf die Speicherkapazität des Laufwerks voll beanspruchen. Die Möglichkeit der Erweiterung auf bis zu 32 Megabyte bei zukünftigen Versionen ist berücksichtigt worden.

Die Benutzer von CP/M 2 sind voneinander durch Benutzernummern physikalisch getrennt, mit Dateikopiermöglichkeiten von einem Benutzerbereich zum anderen. Sehr leistungsfähige Mechanismen sorgen in CP/M 2 für direkten Zugriff zu jedem der 65 536 Datensätze einer acht Megabyte großen Datei.

CP/M stellt auch einen leistungsfähigen Texteditor, einen Intelkompatiblen Assembler und ein Testhilfesystem zur Verfügung. Zusammen mit dem Console Command Processor übersteigt CP/M die Leistungsmöglichkeiten einiger Großrechner.

CP/M ist logisch in mehrere verschiedene Bestandteile untergliedert:

BIOS	Basic Input/Output (I/O) System (hardware-abhängig)
BDOS	Basic Disk Operating System
CCP	Console Command Processor
TPA	Transient Program Area

Das BIOS stellt die Primitivoperation zur Verfügung, die notwendig sind, um zu den Laufwerken zugreifen zu können und Standard-Peripheriegeräte anzuschließen (Bildschirm (CRT), Drucker und benutzerdefinierte Peripheriegeräte).

Das BDOS ist verantwortlich für die Plattenspeicherverwaltung. Es steuert ein oder mehrere Laufwerke, die unabhängige Dateiverzeichnisse enthalten. Das BDOS implementiert Plattenzuordnungsstrategien, die eine dynamische Dateikonstruktion ermöglichen, wobei die Kopfbewegung über die Platte während eines Zugriffs minimiert wird.

Das BDOS hat Eingangspunkte für folgende Primitivoperationen, zu denen programmgesteuerte Zugriffe möglich sind:

SEARCH	Suchen auf einer bestimmten Platte nach einer genannten Datei
OPEN	Datei für weitere Operationen öffnen
CLOSE	Datei nach Verarbeitung schließen
RENAME	Namen einer einzelnen Datei ändern
WRITE	Satz auf eine bestimmte Datei schreiben
SELECT	Bestimmtes Plattenlaufwerk für weitere Operationen auswählen

Der CCP sorgt für eine symbolische Schnittstelle zwischen der Benutzerkonsole und dem übrigen CP/M-System. Der CCP nimmt die Eingabe über das Konsolgerät auf und führt Befehle aus, wie z. B. Auflisten des Dateiverzeichnisses, Ausdrucken eines Dateiinhalts und Steuerung der Arbeit von nicht residenten (auf Platte gespeicherten) Programmen wie Assembler, Editor und Testhilfen.

Der letzte Bestandteil von CP/M ist der Bereich, der Transient Program Area (TPA) genannt wird. Die TPA beinhaltet Programme,

die von der Platte unter Steuerung des CCP geladen werden. Während des Programm-Editierens z. B. befinden sich in der TPA der Maschinencode des Texteditors und die Daten, die neu gestaltet werden. In ähnlicher Weise können Programme, die unter CP/M angelegt wurden, in die TPA geladen und ausgeführt werden.

Die CP/M-Komponenten können durch ein laufendes Programm überlagert werden: Sobald ein Programm in die TPA geladen ist, können die Speicherbereiche von CCP, BDOS und BIOS als Datenbereiche des Programms verwendet werden. Ein "bootstrap"-Lader ist verfügbar, wenn BIOS nicht überschrieben ist; das Benutzerprogramm braucht nur nach der Ausführung zum Bootstrap-Lader zu verzweigen, und dann wird der gesamte CP/M-Monitor von der Platte neu geladen.

Das CP/M Betriebssystem ist in verschiedene Module eingeteilt. Ein Teil, das BIOS, definiert die Hardware-Umgebung, in der CP/M läuft.

# DIE ERSTEN SCHRITTE

In diesem Kapitel wird Ihnen die CP/M-Bedienungs-Software vorgestellt. Sie werden hier erfahren, wie einfache aber gleichzeitig wichtige Funktionen ausgeführt werden. Eine schrittweise Beschreibung zeigt Ihnen, wie das Betriebssystem geladen wird, wie das Inhaltsverzeichnis einer Platte am Bildschirm angezeigt werden kann, wie Sie eine neue Festplatte oder Diskette benutzungsbereit machen können (Formatieren), und anschließend wie Sie die System-Diskette kopieren können. Eine weitere Beschreibung zeigt Ihnen, wie Sie Ihren Computer auf verschiedene Platten- oder Druckertypen abstimmen und die Funktionstasten programmieren können (Konfiguration). Sobald Sie mit Ihrem Computer vertraut sind, werden Sie mit Sicherheit in weiteren Abschnitten dieses Handbuchs lesen, damit Sie sich über sämtliche Einsatzmöglichkeiten der Software informieren können.

Nach umfangreichen Übungen, die in diesem Kapitel enthalten sind, können Sie lesen, wie Sie Nicht-NCR-Software beziehen und mit Hilfe des auf der System-Diskette befindlichen EXCHANGE-Programms in Ihrem Computer einsetzen können. Diese Information wird nützlich sein, wenn Sie bereits andere Programmier- oder Anwendungssoftware besitzen oder zu einem späteren Zeitpunkt kaufen.

**Hinweis:** Auch wenn Sie bereits Computer- und CP/M-Kenntnisse besitzen, wäre es trotzdem von Vorteil, dieses Kapitel zu lesen. So können Sie die neuen CP/M-Funktionen für NCR DECISION MATE V am schnellsten kennenlernen.

## 1.2 Grundbegriffe der Software

CP/M führt nicht nur Steuerungsfunktionen des Systems aus. Informationen, die Sie anhand der Tastatur eingeben, können verarbeitet und am Bildschirm ausgegeben werden. Das interaktive CP/M-Betriebssystem ermöglicht eine leichte, konversationsähnliche Verständigung zwischen Ihnen und Ihrem Computer: Der Computer stellt Ihnen eine Frage und wartet auf Ihre Antwort. Manchmal stellt Ihnen der Computer mehrere Entscheidungsmöglichkeiten zur Wahl. In diesem Fall erscheint eine Auswahltablelle, die meistens als

”Menu” bezeichnet wird. Der Computer gibt Ihnen dann genaue Auskunft, wie Sie ihm Ihre Entscheidung mitteilen können.

### 1.2.1 Bereitschaftszeichen (Prompts)

Die Bereitschaftszeichen (Prompts) des Systems sind A>, B>, C>, usw. Diese erinnern den Benutzer, mit welchem Laufwerk CP/M gerade arbeitet, und daß eine Anweisung des Benutzers erwartet wird. Andere auf Ihrer Diskette befindliche Programme verwenden andere Bereitschaftszeichen, z. B. ED und PIP geben ein Sternzeichen an (\*).

Um von einem Laufwerk auf ein anderes umzuschalten, müssen Sie lediglich die Bezeichnung für das gewünschte Laufwerk (z.B. A oder B) und einen Doppelpunkt (:) eingeben. Anschließend betätigen Sie die Abschlußtaste ↵ (oft als ”Carriage Return” oder einfach CR/cr bezeichnet). Die folgenden Schritte zeigen Ihnen, wie Sie von Laufwerk A auf Laufwerk B umschalten können, und umgekehrt.

A>B: ↵

und umgekehrt

B>A: ↵

A>

### 1.2.2 Laden des Betriebssystems

Das Bereitschaftszeichen des Systems kann erst erscheinen, nachdem Sie das Betriebssystem in den maschineninternen Speicher geladen haben. Hierzu müssen Sie den Computer einschalten und die CP/M-Diskette einlegen. Diese Startmöglichkeit wird als Kaltstart (auch ”cold boot”) bezeichnet. Danach erscheint am Bildschirm immer das Bereitschaftszeichen, wenn CP/M soeben eine Operation abgeschlossen hat und auf eine weitere Anweisung wartet. Sie sollten mit der nächsten Anweisung solange warten, bis das Bereitschaftszeichen wieder erscheint, da sonst die Anfangszeichen dieser Anweisung vom Computer möglicherweise nicht erkannt werden.

Manchmal werden Sie Anlaß haben, einen Warmstart (auch als ”warm boot“ oder ”system reboot“ bezeichnet) durchzuführen. Dadurch wird die laufende Operation abgebrochen und das Betriebssystem erneut geladen. Der Warmstart erfolgt, wenn Sie bei noch gedrückter CONTROL-Taste zusätzlich die C-Taste drücken. In

diesem Handbuch werden Sie für diese CONTROL-Taste das Symbol † (oder die Abkürzung ctl-)lesen.

Ein Warmstart ist immer bei einem Diskettenwechsel erforderlich, ebenfalls wenn Sie einen Programmablauf beenden und die Steuerung an CP/M zurückgeben wollen. Sie können einen Warmstart auch auslösen, wenn nach Beendigung einer Operation weder das Bereitschaftszeichen des Systems noch eine Fehlermeldung erscheint.

### **1.2.3 CP/M-Kommandos**

CP/M beinhaltet zwei Kommandoarten, nämlich residente (eingebaute) Kommandos und transiente (vorübergehend im Maschinenspeicher befindliche) Kommandos. Die residenten Kommandos werden mit dem Betriebssystem automatisch in den Speicher geladen. Sie können deshalb ohne weiteres aktiviert werden. Die transienten Kommandos befinden sich nur auf der Diskette, es sei denn, sie werden zwecks Ihrer Ausführung in den Speicher geladen. Die folgende Tabelle stellt eine Zusammenfassung dieser zwei Kommandoarten dar. Es ist nicht erforderlich, daß Sie bereits die Funktionsweise dieser Kommandos verstehen. Einige sind für erfahrene Computeranwender vorgesehen.

CP/M-Kommandos	
Resident	Funktion
DIR	Erstellt am Bildschirm eine Liste der Dateien im Platten-Inhaltsverzeichnis.
ERA	Löscht eine oder mehrere Plattendateien.
REN	Benennt eine Datei um.
SAVE	Sichert den Speicherinhalt als Plattendatei.
TYPE	Schreibt den Inhalt einer Datei am Bildschirm.
USER	Schaltet auf einen anderen Benutzer um.
Transient	Funktion
ASM	Konvertiert eine Datei, die Assemblersprache enthält, in eine Hex-Datei, die als Maschinencode im Speicher ausgeführt werden kann.
BACKUP+	Kopiert den Inhalt einer oder mehrerer Dateien von einer Platte auf eine andere. Dabei dürfen die Dateien beliebig groß sein. Mit BACKUP können Sie auch Dateien am Bildschirm oder am Drucker ausgeben.
CONFIG+	Wahlweise vorübergehende oder permanente Änderung von Konfigurations-Parametern.
DDT	Testhilfe: Laden, Änderung und Testablauf von Programmen.
DISCIT+	Richtet eine Festplatte für ihre Verwendung im Computer ein.
DUMP	Gibt den Inhalt einer Datei in hexadezimaler Form aus.
ED	Erstellen und Editieren von Textdateien.
EXCHANGE+	Ermöglicht die Benutzung von Nicht-NCR-Anwendungsdisketten im NCR DECISION MATE V.
FORMAT+	Richtet eine Diskette für ihre Verwendung im Computer ein.
LOAD	Laden eines Programms in den Speicher, wobei eine von ASM erstellte Hex-Datei in eine COM-Datei umgewandelt wird.
PIP	Peripheral Interchange Program: Ausführung von Kopiervorgängen mit Platten-Laufwerken und externen Geräten.
STAT	Anzeigen von Information zur Speicherplatzbelegung. Bestimmen der Gerätezuordnung.
SUBMIT	Ausführung einer Datei, die CP/M-Kommandos enthält.
SYSGEN	Legt eine Kopie der CP/M-Systemplatte an.
XSUB	Erweiterung von SUBMIT: Ermöglicht Eingabe im Rahmen eines SUBMIT-Ablaufs.

+ Neues Dienstprogramm für NCR DECISION MATE V

## 1.2.4 Umschalten auf ein anderes Laufwerk

Sie können von der gegenwärtig aktivierten Platte durch Eingabe einer Laufwerkbezeichnung (A bis P) gefolgt von einem Doppelpunkt (:) auf ein anderes Laufwerk umschalten, falls der CCP gerade auf eine Konsoleingabe wartet. Die nachstehend abgebildete Befehlsfolge kann ausgeführt werden, nachdem das CP/M-System von der Platte A geladen worden ist. Ihre CP/M-Diskette enthält allerdings andere als die hier abgebildeten Dateien. Diese Befehlsfolge erfordert ferner eine zweite, bereits benutzte Platte. Aus diesen Gründen sollten Sie diese Übung bis Kapitel 2 zurückstellen.

A>DIR                    Liste aller Dateien auf Platte A

A:SAMPLE ASM SAMPLE PRN

A>B:                    Schalte um auf Laufwerk B

B>DIR\*.ASM            Liste alle "ASM"-Dateien auf B

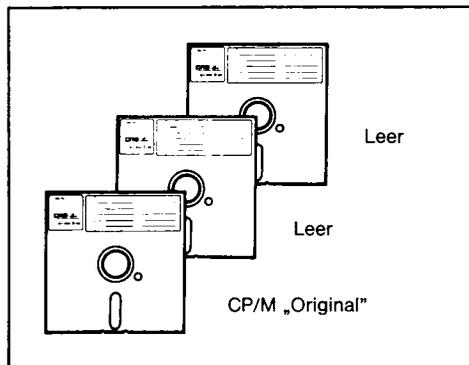
B:DUMP ASM FILES ASM

B>A:                    Schalte wieder auf A

# ÜBUNGEN

Die folgenden Beispiele zeigen wie einfach die Bedienung Ihres NCR DECISION MATE V sein kann. Für diese Übungen sind insgesamt drei Disketten zu empfehlen: Ihre CP/M-Systemdiskette und zwei leere Disketten. Anhand dieser Disketten werden Sie nun eine zusätzliche Kopie des Betriebssystems machen. (Auf diese Weise sorgen Sie dafür, daß Sie im Falle eines Diskettendefekts immer noch eine einwandfreie Kopie des Betriebssystems zur Verfügung haben.) Sie werden dann mit der Systemkopie und der dritten Diskette arbeiten. Die Original-Diskette muß gut aufbewahrt werden.

**Hinweis:** Wenn Sie keine zusätzlichen Leerdisketten haben, können Sie trotzdem ohne weiteres die ersten zwei Übungen ausführen.



Vergessen Sie nicht, daß die Tastatur-Eingaben, die im Rahmen dieser Übungen anfallen, jeweils durch Drücken der Carriage-Return-Taste (↵) abgeschlossen werden müssen.

## 1.3 Laden des Betriebssystems

Je nach Computermodell wird Ihnen ein System mit zwei Disketten-Laufwerken oder eine andere Zusammenstellung aus Festplatten-Laufwerken und Disketten-Laufwerken zur Verfügung stehen.

Unabhängig von Ihrem System werden Sie immer zunächst mit

dem Disketten-Laufwerk A arbeiten. Das wollen wir nun in die Praxis umsetzen, indem wir das Betriebssystem in den Speicher laden.

1. Schalten Sie den Computer ein.
2. Legen Sie die Systemdiskette in Laufwerk A ein.
3. Drücken Sie Carriage Return (auf der Tastatur als ↵ abgebildet).

Es erscheint am Bildschirm eine Copyright-Mitteilung, darunter das Bereitschaftszeichen.

```
CP/M (R) 2.2 for NCR DECISION MATE V
D006-0224-0000
Copyright (c) 198X, DIGITAL RESEARCH
Serial Number XXXXX
NCR-Software Version 2.2 (XX.XX)
Firmware Version: X.XX.XX
A>
```

Die programmierbaren Funktionstasten

In den folgenden Übungen werden Sie eine zeitsparende Bedienungshilfe Ihres NCR DECISION MATE V kennenlernen, nämlich die programmierbaren Funktionstasten in der oberen Reihe der Tastatur, die mit F1 bis F20 gekennzeichnet sind. Ein Papierstreifen mit den Bezeichnungen derjenigen Tasten, die bereits vorprogrammiert sind, befindet sich unmittelbar nach diesem Kapitel auf einer nicht nummerierten Seite. Sie können diesen Streifen ausschneiden und oberhalb der Funktionstasten anbringen.

**Hinweis:** Sollten Sie bereits eigene Verwendungen für diese Funktionstasten festgelegt haben, können Sie diese Tasten mit einem ähnlichen Papierstreifen entsprechend kennzeichnen. In den folgenden Übungen müssen Sie dann ggf. die Kommandos Zeichen für Zeichen eingeben, statt Funktionstasten zu benutzen. Falls Sie mit einem anderen Betriebssystem in Ihrem NCR DECISION MATE V arbeiten, könnte eine andere Belegung der Funktionstasten zu Dateiverlusten führen. Um das zu vermeiden, sollten Sie von der Benutzung dieser vorbelegten Funktionstasten absehen.

### 1.3.1 Anzeigen des Platten-Dateiverzeichnisses

Dieses Inhaltsverzeichnis enthält die Namen sämtlicher transienten Kommandos und Dienstprogramme, die Ihnen zur Verfügung stehen. Nachdem Sie Funktionstaste F1 gedrückt haben, können Sie

das Inhaltsverzeichnis Ihrer Diskette am Bildschirm lesen. Sie werden diese Funktion sehr oft benutzen.

Sie werden bemerkt haben, daß die CP/M-Kommandodateien den Namenszusatz COM aufweisen. Nachstehend finden Sie ein Beispiel eines solchen Inhaltsverzeichnisses. Ihre Diskette enthält die meisten der hier aufgelisteten Dateien:

```

A:STAT   COM:ASM       COM:LOAD   COM:PIP     COM
A:ED     COM:SYSGEN   COM:SUBMIT COM:DUMP    COM
A:FORMAT COM:CONFIG   COM:EXCHANGE COM:DISCIT  COM
A:BACKUP COM:DDT      COM:XSUB   COM:DEMO5   COM
A:LADDER COM:LADDER   DAT:CATCHUM COM:CATCHUM DAT
A:BRUN   COM:MUSIC   COM:VEGAS  COM:VELINE  COM
A:VEBAR  COM:VEPIE   COM:VECOLL COM:VECHNG  COM
A:VESCRIP COM:VESAFE  COM:VEPLOT  COM
A>

```

## 1.4 Die ersten Schritte

### Nur für ein System mit zwei Disketten-Laufwerken

**Wichtig:** Wenn Sie nur ein Disketten-Laufwerk haben, können Sie diesen Abschnitt überspringen. Lesen Sie in diesem Fall den Abschnitt "Die ersten Schritte - Für ein System mit einem Disketten-Laufwerk und einem oder mehreren Festplatten-Laufwerken" bzw. den Abschnitt "Die ersten Schritte - Für ein System mit einem Disketten-Laufwerk."

#### 1.4.1 Formatieren einer neuen Diskette

Es gibt keine einheitliche Methode, wie Computersysteme Daten auf Disketten lesen und schreiben. Deshalb muß eine Diskette dem vom jeweiligen Computer benutzten Format angepaßt werden. Mit dem Format-Kommando können Sie eine Diskette entsprechend einrichten.

1. Drücken Sie die Funktionstaste F2. Nach der Copyright-Mitteilung erscheint folgendes am Bildschirm:

```

          DISK TYPE MENU
1. . . . Format Double Density, Double Sided.
2. . . . Exit Program
Enter your selection:

```

2. Geben Sie 1 ein. Damit wird dem Computer mitgeteilt, daß eine Diskette für doppelseitige Beschriftung sowie verdoppelte Speicherdichte verwendet wird. Es wird dann folgendes am Bildschirm ausgegeben:

Enter drive name you wish to format

3. Geben Sie B ein. Damit haben Sie bestimmt, daß sich die zu formatierende Diskette im Laufwerk B befindet. Sie erhalten dann die folgende Aufforderung:

Insert diskette into Drive B:  
and hit (CR) to start formatting. ↵

4. Legen Sie eine zweite Diskette in Laufwerk B ein und betätigen Sie ↵.  
Das Format-Programm formatiert und prüft gleichzeitig jede Spur der Diskette. Währenddessen erscheint am Bildschirm die Nummer der Spur, die gerade geprüft wird. Nach der Formatierung haben Sie die folgende Wahl:

End Format  
Would you like to format any more (Y/N) ?

Wenn Sie mit Y antworten, können Sie weitere Disketten formatieren. Wenn Sie keine weiteren Disketten formatieren wollen, geben Sie N ein. Daraufhin wird A> wieder erscheinen.

## 1.4.2 Kopieren der Systemdiskette

Die Disketten, die Sie formatiert haben, können nun verwendet werden. Um die Software auf Ihrer Systemdiskette vor einem versehentlichen Löschen zu schützen, wollen wir nun eine sogenannte Sicherungskopie Ihrer Systemdiskette anfertigen. Dazu werden wir die SYSGEN- und PIP (Peripheral Interchange Program)-Kommandos benutzen. Das SYSGEN-Kommando (Funktionstaste F3) kopiert das Betriebssystem einschließlich aller permanent im Maschinenspeicher befindlichen (residenten) Kommandos. Das PIP-Kommando (F4) kopiert die übrigen Dateien. Die Einzelheiten der SYSGEN-Funktion werden im nächsten Kapitel eingehend erläutert. Zunächst wollen wir uns nur mit dem dazugehörigen Dialog zwischen Ihnen und Ihrem Computer befassen.

1. Vorausgesetzt, daß Sie die Systemdiskette im Laufwerk A und eine formatierte Diskette im Laufwerk B haben, können Sie nun F3 betätigen, um den nachstehend abgebildeten Dialog einzuleiten.

```
A>SYSGEN
  SYSGEN VER.2.X
  SOURCE DRIVE NAME (OR RETURN TO
  SKIP) (A)
  SOURCE ON A THEN TYPE RETURN ↵
  FUNCTION COMPLETE
  DESTINATION DRIVE NAME (OR RE-
  TURN TO REBOOT) (B)
  DESTINATION ON B THEN TYPE RETURN ↵
  FUNCTION COMPLETE
  DESTINATION DRIVE NAME (OR RETURN
  TO REBOOT) ↵
A>
```

Das Betriebssystem befindet sich nun auch auf der Diskette im Laufwerk B. Sie sind nun in der Lage, die transienten Kommando-dateien zu kopieren.

2. Drücken Sie F4. Der dadurch begonnene Kopiervorgang beschränkt sich nicht auf bestimmte Dateien. Die Sternzeichen \*.\* bewirken, daß sämtliche Dateien unabhängig vom Dateinamen und unabhängig vom Namenszusatz (z. B. COM) kopiert werden.

```
A>PIP B:=A: *.* [V]
```

Während des Kopiervorgangs zeigt das PIP-Programm den Namen der gerade kopierten Datei am Bildschirm an. Nach Beendigung von PIP erscheint das System-Bereitschaftszeichen A>.

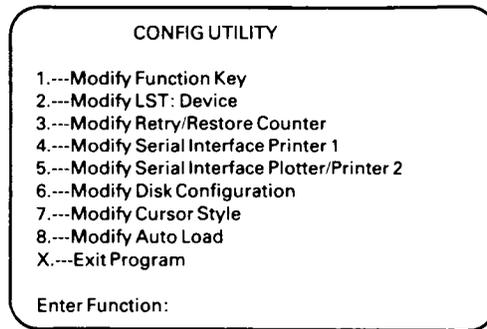
3. Um das Inhaltsverzeichnis der neu beschriebenen Diskette anzuschauen, betätigen Sie die Funktionstaste F5. Sie werden feststellen, daß die Inhaltsverzeichnisse der zwei Disketten gleich sind.
4. Sie können nun Ihre Systemdiskette aus Laufwerk A entfernen und an einem sicheren Ort aufbewahren. Nach Entfernung der Systemkopie aus Laufwerk B können Sie diese mit einem Aufkleber mit der Bezeichnung "CP/M-Systemdiskette" versehen. Legen Sie anschließend diese Diskette in Laufwerk A ein und drücken Sie ↑C (Drücken der C-Taste bei gedrückter CONTROL-Taste). Dadurch teilen Sie dem Computer mit, daß Sie in einem Laufwerk die Diskette gewechselt haben.

### **1.4.3 Bestimmung der System-Konfiguration**

Mit CONFIG können Sie Ihren Computer auf seine Umgebung sowohl auf der Hardware- als auch auf der Software-Ebene abstimmen. Dieses Dienstprogramm wird zu folgenden Zwecken eingesetzt:

1. Bestimmung der Funktionen von bis zu 20 programmierbaren Funktionstasten, die sich in der oberen Reihe der Tastatur befinden.
2. Einstellen des LST: Geräts
3. Änderung der Anzahl der Versuche, die der Computer macht, um auf Diskette zu lesen oder zu schreiben. Diese Zahl muß zwischen 0 und 9 einschließlich sein, wobei der Standardwert 4 ist.
4. Änderung der Eigenschaften (Parameter) der seriellen Schnittstelle zu Drucker 1. Diese Parameter enthalten Informationen zu der Anzahl der Stop-Bits, der Art der Parity-Festlegung, der Zeichenlänge sowie der Übertragungsgeschwindigkeit.
5. Änderung der Parameter der seriellen Schnittstelle zum Plotter bzw. zum Drucker 2. Diese Parameter enthalten Informationen zu der Anzahl der Stop-Bits, der Art der Parity-Festlegung, der Zeichenlänge und der Übertragungsgeschwindigkeit.
6. Änderung der Plattenkonfiguration.
7. Ändert die Darstellung des Cursors (Schreibmarke).
8. Stellt den Betrieb des Automatikstarts ein und ändert die zugehörige Befehlszeile.

1. Betätigen Sie die Funktionstaste F6. Nach der Copyright-Mitteilung erscheint das Konfiguration-Hauptmenu.



2. Sie können nun die Funktion auswählen, die Sie modifizieren wollen. Als Beispiel drücken Sie auf 1, um die bisherigen Funktionen der Funktionstasten einzusehen und eine dieser Funktionstasten neu zu definieren.

Beispiel:

Die Funktionstaste F13 in der oberen Reihe der Tastatur soll ab jetzt den Programmaufruf STAT A:\*. \* (CR) bewirken.

Vom neuen Menu aus geben Sie 2 ein, um die Neudefinition einzuleiten.

Ein weiteres Menu bietet Ihnen nun die Gelegenheit, genau eine Funktionstaste zu bestimmen. Betätigen Sie F13. Die bisherige Funktion von F13 wird im Format F13 = . . . angezeigt. Geben Sie nun Zeichen für Zeichen die neue Definition STAT A: \*. \* ein, und drücken Sie ↵

**Hinweis:** Sollten Sie ein falsches Zeichen eingeben, können Sie durch Drücken der F13-Taste zum Modify-Function-Key-Menu zurückkehren. Dann können Sie die Neudefinition von F13 erneut versuchen.

Abschließend betätigen Sie F13. Das Modify-Function-Key-Menu erscheint wieder.

Zur Bestätigung der Neudefinition geben Sie 1 ein. Sie können dem Bildschirm entnehmen, daß F13 die neue Funktion besitzt: FUNCTION 13 = STAT A: \*. \* (CR). Nach Drücken von ↵ können Sie mit folgenden Eingaben weiterverfahren:

X bewirkt die Rückkehr zum Hauptmenu.

X bewirkt die Beendigung des CONFIG-Programms.

1 schreibt die Änderung in den Speicher sowie auf Diskette. Damit erreichen Sie, daß F13 ab sofort die Funktion STAT A: \*.\* aufrufen wird.

3. Die anderen Konfigurations-Parameter können ebenfalls anhand der leicht verständlichen Menus geändert werden. Nachstehende Tabelle stellt eine Zusammenfassung dieser Funktionen dar.

Funktion	Standartwert	zulässige Änderungen
2. Änderung des LST:Geräts		Serieller Drucker 1 " " " 2 Bildschirm
3. Änderung der Rückschreib- und Wiederholungszähler	beide Fehlerausgleichszähler mit 4	Wert dieser Zähler darf 0 bis 9 sein
4. Änderung der Schnittstelle zum seriellen Drucker 1  a. Anzahl der Stop-Bits b. Parität  c. Zeichenlänge d. Baudrate e. Portadresse	1 gerade  7 9600 60	1 1/2,2 außer Kraft oder ungerade 5,6,8 50 - 19200 30 - C8
5. Änderung der Schnittstelle zum seriellen Drucker 2  a. Anzahl der Stop-Bits b. Parität  c. Zeichenlänge d. Baudrate e. Portadresse	1 gerade  7 9600 B0	1 1/2,2 außer Kraft oder ungerade 5,6,8 50 - 19200 30 - C8
6. Änderung der Platten-Konfiguration	zwei Disketten-Laufwerke	andere Platten-Konfiguration
7. Änderung der Cursor-darstellung	blinkende Unterstreichung	Unterstreichung blinkendes Rechteck Rechteck.
8. Änderung des Automatikstarts	Automatikstart bei Kalt- u. Warmstart. Ohne Funktion, da Befehlszeile leer.	kein Automatikstart Automatikstart bei Kaltstart, bei Warmstart
Standartwerte = diese Werte gelten automatisch, auch ohne Aufruf von CONFIG. Zulässige Änderungen = diese Werte können nur mit CONFIG festgelegt werden.		

4. Vergessen Sie nicht, eine Sicherungskopie Ihrer neuen Systemdiskette zu machen. Dies ist besonders sinnvoll, wenn Sie mit der geänderten Konfiguration später arbeiten wollen. Ein nochmaliger Aufruf von CONFIG ist dann nicht erforderlich.

Sie werden noch weitere Einzelheiten zu CONFIG in diesem Handbuch lesen.

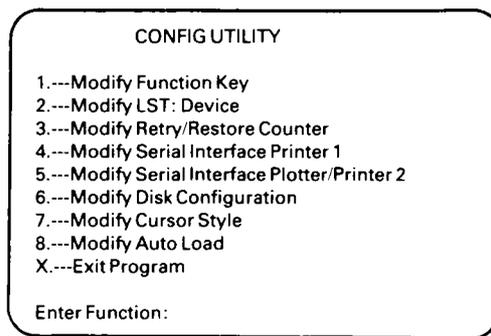
## 1.5 Die ersten Schritte

### Nur für ein System mit einem Disketten-Laufwerk und einem oder mehreren Festplatten-Laufwerken

#### 1.5.1 Bestimmung der System-Konfiguration

Sie müssen zunächst Ihrem Computer mit Hilfe des CONFIG-Dienstprogrammes die System-Konfiguration mitteilen, in der er arbeitet.

1. Betätigen Sie die Funktionstaste F6. Nach der Copyright-Mitteilung erscheint das Konfiguration-Hauptmenu.



2. Geben Sie 6 ein, um die Platten-Konfiguration zu ändern.
3. Geben Sie 1 ein. Sie können dann Ihrem Computer mitteilen, wieviele Disketten-Laufwerke Ihr System besitzt.
4. Nach entsprechender Aufforderung können Sie durch Betätigung der vom Menu geforderten Taste die Zahl eingeben.
5. Geben Sie 2 ein. Sie können dann Ihrem Computer mitteilen, wieviele Festplatten-Laufwerke Ihr System besitzt.
6. Geben Sie wie bei der Anzahl der Disketten-Laufwerke die Anzahl der Festplatten-Laufwerke ein.

**Wichtig:** Jedes Festplatten-Laufwerk wird vom System als zwei Laufwerke betrachtet. Bei einem System mit einem Disketten-Laufwerk und einem Festplatten-Laufwerk wird demzufolge das Disketten-Laufwerk als A und das Festplatten-Laufwerk als B und C bezeichnet.

7. Nach Drücken von 3 können Sie Ihre Eingaben auf Richtigkeit prüfen. Es erscheint am Bildschirm eine Abbildung, die diesen Eingaben entspricht. Nach Betätigung von ↵ können Sie weiterverfahren. Sollten Sie eine Fehleingabe bei der Anzahl der Disketten- oder Festplatten-Laufwerke festgestellt haben, können Sie die obigen Schritte wiederholen. Ansonsten können Sie bei 8 weiterfahren.
8. Geben Sie X ein, um zum Hauptmenu zurückzukehren.  
Geben Sie X ein, um das CONFIG-Programm zu verlassen.
10. Geben Sie 2 ein. Die vom Standardwert abweichende Konfiguration wird vom Betriebssystem im Speicher vermerkt. Anschließend erscheint das System-Bereitschaftszeichen.

### 1.5.2 Formatieren einer neuen Diskette

Es gibt keine einheitliche Methode, wie Computersysteme Daten auf Disketten lesen und schreiben. Deshalb muß eine Diskette dem vom jeweiligen Computer benutzten Format angepaßt werden. Mit dem Format-Kommando können Sie eine Diskette entsprechend einrichten. Bei der Benutzung dieses Kommandos ist Vorsicht geboten. Es werden nämlich sämtliche Daten vernichtet, die bisher auf der Diskette waren. Sie sollten deshalb nur eine unbenutzte Diskette und nicht etwa eine Systemdiskette für diese Übung verwenden.

1. Nach Betätigung der Funktionstaste F2 erscheinen die Copyright-Mitteilung

```
A>FORMAT ↵  
FORMAT for NCR DECISION MATE V  
Version X.Y  
Copyright (c), NCR CORPORATION YYYY
```

und anschließend das Disk-Type-Menu

DISK TYPE MENU

1. --- Format Double Density, Double Sided.  
2. --- Exit Program

Enter your selection:

2. Geben Sie 1 ein. Damit wird dem Computer mitgeteilt, daß eine Diskette für doppelseitige Beschriftung sowie verdoppelte Speicherdichte verwendet wird. Dann erscheint am Bildschirm die Aufforderung

Enter drive name you wish to format

3. Geben Sie A ein, um dem Computer mitzuteilen, daß Sie die Diskette im Laufwerk A formatieren wollen. Sie werden dann aufgefordert, die zu formatierende Diskette in Laufwerk A einzulegen:

Insert diskette into drive A:  
and hit (CR) to start formatting. ↵

4. Sie müssen allerdings zuerst die System-Diskette aus Laufwerk A entfernen. Legen Sie die unbenutzte Diskette ein und drücken Sie ↵.
- Das Format-Programm formatiert und prüft gleichzeitig jede Spur der Diskette. Währenddessen erscheint am Bildschirm die Nummer der Spur, die gerade geprüft wird. Nach der Formatierung haben Sie die folgende Wahl:

End Format

Would you like to format any more (Y/N)?

Wenn Sie mit Y antworten, können Sie weitere Disketten formatieren. Wenn Sie keine weiteren Disketten formatieren wollen, geben Sie N ein. Es erscheint die Meldung  
A:MOUNT O.S. Disk (CR).

5. Legen Sie Ihre Systemdiskette ein, und drücken Sie ↵ .

### 1.5.3 Formatieren einer Festplatte

Das DISCIT-Programm richtet eine Festplatte für Ihr System ein (Initialisierung). Jede Spur wird mit Daten belegt, und diese Daten werden dann vom System wieder gelesen und auf Genauigkeit geprüft. Hier gilt dasselbe Vorsichtsgebot wie bei FORMAT, da ein unüberlegter Aufruf von DISCIT zu der Vernichtung einer großen Menge wertvoller Daten führen könnte.

1. Drücken Sie die Funktionstaste F7. Nach der Copyright-Mitteilung erscheint

```
DISK TYPE MENU
1. --- Format Disk
2. --- Exit Program
Enter your selection:
```

2. Geben Sie 1 ein. Das DISCIT-Programm wird gestartet.

```
Enter drive name you wish to format
Winchester units are X,...,Y
```

3. Nun können Sie das Festplatten-Laufwerk B als die zu formatierende logische Einheit bestimmen. Dazu geben Sie B ein. Danach erscheint

```
Selected drive name is "B"
Enter number of certifications (1-99, CR=4):
```

Hier können Sie bestimmen, wie oft die Festplatte auf Speichertauglichkeit geprüft wird. Sie können jede beliebige Zahl von 1 bis einschließlich 99 eingeben. Falls Sie keine ausdrückliche Angaben machen, gilt der Standardwert 4.

4. Geben Sie 1 ein und danach ↵. Das eigentliche Formatieren kann nun beginnen.

```
Hit <CR> to start formatting
"Control" C to Abort
```

5. Dazu müssen Sie nur noch ↵ betätigen. DISCIT belegt jede Spur mit Daten und liest diese Daten wieder, wobei die Speichergenauigkeit jeder Spur überprüft wird. Gleichzeitig erscheint am Bildschirm (in aufsteigender Reihenfolge) die Nummer der Spur, die gerade überprüft wird. Nach Beendigung haben Sie die Wahl, ob Sie weitere Platteneinheiten formatieren wollen.

```
End Format
Would you like to format any more? (Y/N)
```

Wenn Sie weiter formatieren wollen, geben Sie Y ein. Andernfalls geben Sie N ein, um DISCIT zu verlassen.

### 1.5.4 Kopieren der Systemdiskette

Wenn Sie mit einem System mit nur einem Disketten-Laufwerk eine Sicherungskopie der System-Software anfertigen wollen, müssen Sie wie folgt vorgehen. Als Ausgangspunkt haben Sie bereits Ihre Systemdiskette in Laufwerk A eingelegt.

1. Drücken Sie die Funktionstaste F3, um das SYSGEN-Programm aufzurufen.

**Wichtig:** Nach Erscheinen der Frage "DESTINATION ON A THEN TYPE RETURN" (d. h. bevor Sie ↵ betätigen) müssen Sie unbedingt Ihre Systemdiskette aus Laufwerk A entfernen und die neu formatierte Diskette einlegen.

```

A > SYSGEN ↵
  SYSGEN VER 2.X
  SOURCE DRIVE NAME (OR RETURN
  TO SKIP) (A)
  SOURCE ON A THEN TYPE RETURN ↵
  FUNCTION COMPLETE
  DESTINATION DRIVE NAME (OR RE-
  TURN TO REBOOT) (A) _ _ _ _ _
  DESTINATION ON A THEN TYPE RETURN ↵
  FUNCTION COMPLETE
  DESTINATION DRIVE NAME (OR RE-
  TURN TO REBOOT) ↵
A >

```

erst Diskette  
wechseln, dann  
A eingeben!

Das Betriebssystem befindet sich nun auch auf der Diskette im Laufwerk A. Entfernen Sie nun diese Diskette und legen Sie die ursprüngliche Systemdiskette wieder ein. Anschließend müssen Sie einen "Warmstart" ausführen (1C).

2. Drücken Sie die Funktionstaste F4. Die Zeichen \*.\* bedeuten, daß alle Dateien unabhängig vom Namen und Namenszusatz (z. B. COM) kopiert werden. Die auf der Diskette befindlichen Dateien werden auf die Platteneinheit mit der logischen Bezeichnung B kopiert und am Bildschirm aufgelistet. Nach Beendigung des Kopiervorgangs erscheint das System-Bereitschaftszeichen.
3. Entfernen Sie die ursprüngliche Systemdiskette aus Laufwerk A, und legen Sie noch einmal die zweite Diskette, die jetzt eine Kopie des Betriebssystems enthält, in Laufwerk A ein. Führen Sie einen Warmstart (1C) aus.
4. Nun geben Sie ein

B:

und drücken Sie die Funktionstaste F8. Damit wird das im Laufwerk B befindliche PIP-Programm aufgerufen. PIP kopiert sämtliche transienten Kommandos auf Ihre neue Sicherungskopie der Systemdiskette im Laufwerk A.

```
B > PIP A:=B:*.*[V]
```

5. Sie können das Inhaltsverzeichnis der neu erstellten Systemdiskette anschauen, indem Sie die Funktionstaste F1 betätigen. Die neue Systemdiskette muß dieselben Dateinamen wie die ursprüngliche Systemdiskette enthalten.

### 1.5.5 Abschluß der Konfigurationsarbeiten

Mit CONFIG können Sie Ihren Computer auf seine Umgebung sowohl auf der Hardware- als auch auf der Software-Ebene abstimmen. Dieses Dienstprogramm wird zu folgenden Zwecken eingesetzt:

1. Bestimmung der Funktion von bis zu 20 programmierbaren Funktionstasten, die sich in der oberen Reihe der Tastatur befinden.
  2. Einstellen des LST:Geräts
  3. Änderung der Anzahl der Versuche, die der Computer macht, um auf Diskette zu lesen oder zu schreiben. Diese Zahl muß zwischen 0 und 9 einschließlich sein, wobei der Standardwert 4 ist.
  4. Änderung der Eigenschaften (Parameter) der seriellen Schnittstelle zu Drucker 1. Diese Parameter enthalten Informationen zu der Anzahl der Stop-Bits, der Art der Parity-Festlegung, der Zeichenlänge, der Übertragungsgeschwindigkeit, sowie der Portadresse.
  5. Änderung der Parameter der seriellen Schnittstelle für den Plotter/Drucker 2. Diese Parameter enthalten Informationen zu der Anzahl der Stop-Bits, der Art der Parity-Festlegung, der Zeichenlänge, der Übertragungsgeschwindigkeit, sowie der Portadresse.
  6. Änderung der Plattenkonfiguration.
  7. Änderung der Cursordarstellung.
  8. Einstellen der Betriebs mit Automatikstart und Ändern der zugehörigen Kommandozeile.
1. Betätigen Sie die Funktionstaste F6. Nach der Copyright-Mitteilung erscheint das Konfiguration-Hauptmenu.

```

CONFIG UTILITY

1---Modify Function Key
2---Modify LST: Device
3---Modify Retry/Restore Counter
4---Modify Serial Interface Printer 1
5---Modify Serial Interface Plotter/Printer 2
6---Modify Disk Configuration
7---Modify Cursor Style
8---Modify Auto Load
X---Exit Program

Enter Function:

```

2. Geben Sie 6 ein, um die Plattenkonfiguration zu ändern.
3. Vom neuen Menu geben Sie 1 ein. Sie können dann die Anzahl

- der Disketten-Laufwerke eingeben.
4. Diese Zahl muß 1 sein, da Ihr System nur ein Disketten-Laufwerk hat. Geben Sie also 1 ein.
  5. Geben sie 2 ein. Sie können dann Ihrem Computer mitteilen, wieviele Festplatten-Laufwerke Ihr System besitzt.
  6. Nach entsprechender Aufforderung können Sie die vom Menu geforderte Taste drücken.
  7. Nach Drücken von 3 können Sie Ihre Eingaben auf Richtigkeit prüfen. Es erscheint am Bildschirm eine Abbildung, die diesen Eingaben entspricht. Nach Betätigung von ↵ können Sie weiterverfahren. Es erscheint das Hauptmenu für die Diskonfigurierung. An diesem Punkt können Sie obige Schritte wiederholen, um Fehler zu korrigieren. Ansonsten können Sie bei 8. weiterverfahren.
  8. Geben Sie ein X ein, um zum Hauptmenu zurückzukehren.
  9. Geben Sie ein X ein, um das CONFIG-Programm zu verlassen.
  10. Geben Sie 1 ein. Die Konfiguration wird vom Betriebssystem sowohl auf der Diskette als auch im Speicher vermerkt.

**Hinweis:** Ein System-Reboot (Kalt- oder Warmstart) ist nicht erforderlich. Die Änderungen, die Sie hier ausgeführt haben, wurden auf der Diskette vermerkt. Damit gilt die soeben festgelegte Konfiguration auch bei späterer Verwendung dieser Diskette, ohne daß CONFIG noch einmal aufgerufen werden muß.

11. Vergessen Sie nicht, eine Sicherungskopie der Systemdiskette mit der neu festgelegten Konfiguration anzufertigen.

Um eine Funktionstaste neu zu definieren, müssen Sie die folgenden Arbeitsschritte befolgen:

1. Betätigen Sie noch einmal die Funktionstaste F6. Am Bildschirm erscheint das Konfiguration-Hauptmenu.

CONFIG UTILITY

- 1.---Modify Function Key
- 2.---Modify LST: Device
- 3.---Modify Retry/Restore Counter
- 4.---Modify Serial Interface Printer 1
- 5.---Modify Serial Interface Plotter/Printer 2
- 6.---Modify Disk Configuration
- 7.---Modify Cursor Style
- 8.---Modify Auto Load
- X.---Exit Program

Enter Function:

2. Drücken Sie auf 1, um die bisherigen Funktionen der Funktionstasten einzusehen und eine dieser Funktionstasten neu zu definieren.

Beispiel:

Die Funktionstaste F13 in der oberen Reihe der Tastatur soll ab jetzt den Programmaufruf STAT A:\*. \* (CR) bewirken.

Vom neuen Menu aus geben Sie 2 ein, um die Neudefinition einzuleiten.

Ein weiteres Menu gibt Ihnen nun die Gelegenheit, genau eine Funktionstaste zu bestimmen. Betätigen Sie F13. Die bisherige Funktion von F13 wird im Format F13 = . . . angezeigt. Geben Sie nun Zeichen für Zeichen die neue Definition STAT A:\*. \* ein, und drücken Sie ↵ .

**Hinweis:** Sollten Sie ein falsches Zeichen eingeben, können Sie durch Drücken der F13-Taste zum Modify-Function-Key-Menu zurückkehren. Dann können Sie die Neudefinition von F13 erneut versuchen.

Abschließend betätigen Sie F13. Das Modify-Function-Key-Menu erscheint wieder.

Zur Bestätigung der Neudefinition geben Sie 1 ein. Sie können dem Bildschirm entnehmen, daß F13 die neue Funktion besitzt: FUNCTION 13 = STAT A:\*. \* (CR). Nach Drücken von ↵ können Sie mit folgenden Eingaben weiterverfahren:

**X bewirkt die Rückkehr zum Hauptmenu.**

**X bewirkt die Beendigung des CONFIG-Programms.**

1 schreibt die Änderung in den Speicher sowie auf Diskette. Damit erreichen Sie, daß F13 ab sofort die Funktion STAT A:\*. \* aufrufen wird.

3. Die anderen Konfigurations-Parameter können ebenfalls anhand der leicht verständlichen Menus geändert werden. Die nachstehende Tabelle stellt eine Zusammenfassung dieser Funktionen dar.

Funktion	Standartwert	zulässige Änderungen
2. Änderung des LST:Geräts		Serieller Drucker 1 " " 2 Bildschirm
3. Änderung der Rückschreibe- und Wiederholungszähler	beide Fehlerausgleichszähler mit 4	Wert dieser Zähler darf 0 bis 9 sein
4. Änderung der Schnittstelle zum seriellen Drucker 1  a. Anzahl der Stop-Bits b. Parität  c. Zeichenlänge d. Baudrate e. Portadresse	1 gerade  7 9600 60	1 1/2,2 außer Kraft oder ungerade 5,6,8 50 - 19200 30 - C8
5. Änderung der Schnittstelle zum seriellen Drucker 2  a. Anzahl der Stop-Bits b. Parität  c. Zeichenlänge d. Baudrate e. Portadresse	1 gerade  7 9600 B0	1 1/2,2 außer Kraft oder ungerade 5,6,8 50 - 19200 30 - C8
6. Änderung der Platten-Konfiguration	zwei Disketten-Laufwerke	andere Platten-Konfiguration
7. Änderung der Cursor-darstellung	blinkende Unterstreichung	Unterstreichung blinkendes Rechteck Rechteck.
8. Änderung des Automatikstarts	Automatikstart bei Kalt- u. Warmstart. Ohne Funktion, da Befehlszeile leer.	kein Automatikstart Automatikstart bei Kaltstart, bei Warmstart
Standartwerte = diese Werte gelten automatisch, auch ohne Aufruf von CONFIG. Zulässige Änderungen = diese Werte können nur mit CONFIG festgelegt werden.		

4. Vergessen sie nicht, eine Sicherungskopie Ihrer neuen Systemdiskette zu machen. Dies ist besonders sinnvoll, wenn Sie mit der geänderten Konfiguration später arbeiten wollen. Ein nochmaliger Aufruf von CONFIG ist dann nicht mehr erforderlich.

Sie werden noch weitere Einzelheiten zu CONFIG in diesem Handbuch lesen.

# ANWENDUNGS-PROGRAMME

Nachdem Sie nun etwas Erfahrung im Umgang mit Ihrem Computer haben, wollen wir uns mit der Anwender-Software befassen, die sich bereits auf Ihrer Systemdiskette befindet.

## LADDER

LADDER ist ein Spielprogramm für Ihren NCR DECISION MATE V. Durch einfaches Drücken von **F16** können Sie das Programm aufrufen. Ein "Hauptmenu" erscheint am Bildschirm.

Wenn Sie "I" eingeben, können Sie eine Spielbeschreibung lesen. Nach Drücken von ↵ erscheint wieder das Hauptmenu. Von diesem aus können Sie:

- mit P das Spiel beginnen
- mit L die Schwierigkeitsstufe ändern
- mit I die Spielbeschreibung lesen
- mit E das Spiel beenden.

Geben Sie nun "P" ein, um das Spiel zu beginnen. Sie können zu jeder Zeit das Spiel beenden, indem sie ↑ C (Drücken der C-Taste bei gedrückter CONTROL-Taste) eingeben.

## CATCHUM

CATCHUM ist ein Spiel für einen oder zwei Spieler. Nach Drücken von **F17** erscheint das Hauptmenu.

Wenn Sie nun "I" eingeben, können Sie eine Spielbeschreibung lesen. Durch Drücken von ↵ kehren Sie dann zum Hauptmenu zurück. Jetzt können Sie:

- mit 1 bzw. 2 die Zahl der Spieler angeben
- mit L die Schwierigkeitsstufe ändern
- mit I die Spielbeschreibung lesen
- mit E das Spiel beenden.

Wenn Sie zu zweit spielen, werden Sie auch dem Computer mitteilen müssen, welcher Spieler gerade spielen will.

Sie können das Spiel zu jeder Zeit mit ↑ C (Drücken der C-Taste bei gedrückter CONTROL-Taste) beenden.

**Hinweis:** In dem Spielprogramm LADDER und CATCHUM können Sie das Spielobjekt mit folgenden Tasten in die in Klammern angegebene Richtungen bewegen: 8(↑), 2(↓), 4(←), 6(→).

## DEMO5

DEMO5 erzeugt eine sich bewegende grafische Darstellung am Bildschirm, damit sie die ausgezeichnete Bildschirm-Auflösung und Verarbeitungsgeschwindigkeit ihres NCR DECISION MATE V erleben können. Sie können dieses Programm durch Drücken von F18 aufrufen. Sie können zu jeder Zeit das Programm durch ↑ C (Drücken der C-Taste bei gedrückter CONTROL-Taste) beenden.

## MUSIC

MUSIC kann Ihnen 11 verschiedene Lieder in Ihrem NCR DECISION MATE V vorspielen. Drücken Sie F19. Die Liederauswahl erscheint am Bildschirm, und Sie können eine der elf Melodien (1-9, A, oder B) aussuchen. Drücken Sie einfach die entsprechende Ziffer bzw. den entsprechenden Buchstaben an der Tastatur (ohne ↵). Sie hören dann die gewählte Melodie. Anschließend erscheint wieder die Liederauswahl. Sie können eine neue Melodie aussuchen oder das Programm mit E (nicht ↑ C) beenden.

## VEGAS

VEGAS (Very Easy Graphic Application System) ist ein vielseitiges Programm, mit dem Sie einen tabellarischen Zusammenhang z. B. aus dem Wirtschafts- oder Geschäftsbereich veranschaulichen können. Mit Hilfe dieses Programms können Sie Linien-, Balken- und kreisförmige Diagramme erstellen und sie an einem Drucker ausgeben. Dieses Programm für CP/M, dessen eigentlicher Name "NCR Business Graphics" ist, wurde in MS-BASIC<sup>®</sup> erstellt. Hinzu kommt die grafische Erweiterung von NCR. Das Programm müssen Sie mit dem Namen VEGAS aufrufen. Dazu genügt das Drücken von F20, worauf das Hauptmenu erscheint.

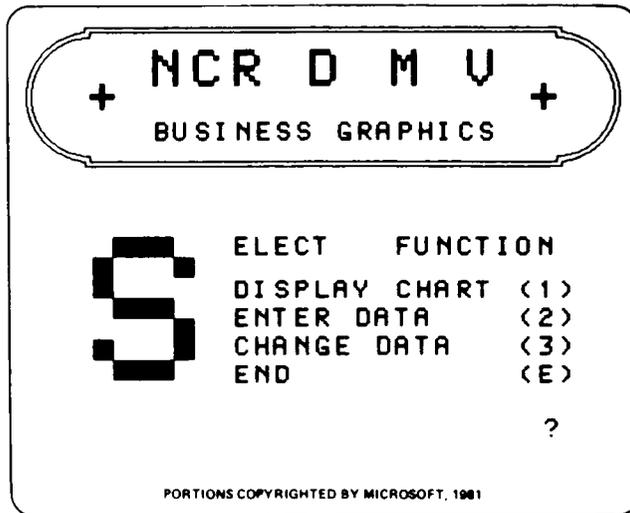


Abb. 1

Das Drücken von 1 zeigt ein bereits erstelltes Diagramm an.

2 gibt Ihnen die Gelegenheit, Daten zwecks Erstellens eines Diagramms einzugeben.

3 ermöglicht die Änderung bereits eingegebener Daten.

Jetzt können Sie dem Programm mitteilen, welcher Drucker angeschlossen ist.

```

***** NCR DECISION MATE V *****
BUSINESS GRAPHICS ***** BUSINESS GRAPHICS
                Select type of printer
                NCR 6411-8510 . . . . . (I)
                EPSON MX82 F/T . . . . . (E)
                EPSON FX80 . . . . . (F)
                OTHER OR NONE . . . . . (O)

```

Abb. 2

Geben Sie I ein, wenn Ihr Drucker ein NCR 6411-8510 (ITOH M8510A) ist. Ein EPSON MX82 erfordert die Eingabe E, ein EPSON FX80 die Eingabe F. Falls Sie einen anderen oder keinen Drucker haben, geben Sie O ein.

**Hinweis:** Bevor Sie Ihre Arbeit mit NCR Business Graphics fortsetzen, sei auf folgende Punkte hingewiesen:

- Wenn nur ein einziges Zeichen als Antwort angefordert wird, darf ↵ (CR) nicht gedrückt werden.
- Nachdem Sie eine Dateneingabe abgeschlossen haben, wird in der Regel "PRESS (CR) TO CONTINUE OR "R" TO REENTER?" am Bildschirm erscheinen. Die Eingabe von R setzt die Dateneingabe zum Anfang des Bildschirms zurück. Sie können dann die Daten für diese Bildschirmseite neu eingeben.
- Rows (Items) sind (waagrechte) Zeilen am Bildschirm.
- Current Chart ist das zuletzt von Ihnen benutzte Diagramm.

### Eingabe von Daten

Wenn Sie neue Daten zwecks Erstellens eines neuen Diagramms eingeben wollen, haben Sie im Hauptmenu 2 eingegeben. Am Bildschirm (Abb. 3) werden die Angaben angefordert, die für ein Diagramm erforderlich sind.

```

***** NCR DECISION MATE V *****
BUSINESS GRAPHICS ***** BUSINESS GRAPHICS

Enter two title lines for the chart, the value for units,
the name of the units and the number of columns and rows(items)
(Default = points for strings, one for numbers)

First title, uppercase only
.....
Second title, uppercase only
.....

Unit of measure ? Name of units, uppercase only .....

Number of columns ** Max length of column title ___
Number of rows(items) **
Press (CR) to continue or 'R' to reenter ?

```

Abb. 3

Diese Fragen werden einzeln gestellt und müssen von Ihnen beantwortet werden:

- First Title: Die erste Überschrift; bis zu 36 Großbuchstaben; kein Komma
- Second Title: Die zweite Überschrift; bis zu 72 Großbuchstaben; kein Komma
- Unit of Measure: Größenordnung der Zahlen; Eingabe 0-9; Standardwert = 0; (0 = \*1, 1 = \*10, 2 = \*100, 3 = \*1000 usw.)
- Name of Units: Bezeichnung der Einheiten; bis zu 12 Großbuchstaben (z. B. DOLLAR, STUECK, usw.)
- Number of Columns: Anzahl der (senkrechten) Spalten; bis zu 48; Standardwert = 1
- Number of Rows (Items): Anzahl der (waagrechten) Zeilen; bis zu 12; Standardwert = 1

Wenn Sie anstelle einer Zahleneingabe nur ↵ drücken, wird der angegebene Standardwert vom Computer verstanden. Wenn Sie durch Drücken von ↵ auf die Eingabe eines angeforderten Buchstabentextes verzichten, wird eine Reihe von Punkten (...) verstanden. Sobald Ihre Dateneingaben für diese Bildschirmseite vollständig sind, werden Sie aufgefordert, ↵ zu drücken. Aufgrund Ihrer bisherigen Eingaben wird eine zulässige Spaltenbreite in Zeichen vom Programm errechnet, die Ihnen dann mitgeteilt wird.

Als nächstes müssen Sie die Bezeichnungen für die verschiedenen Spalten (Abb. 4) eingeben (bis zu insgesamt 59 Zeichen, vorbehaltlich der oben erwähnten Beschränkung der Spaltenbreite). Nach jeder Bezeichnung drücken Sie ↵. Das Betätigen von ↵ ohne Zeicheneingabe erzeugt eine Reihe von Punkten.

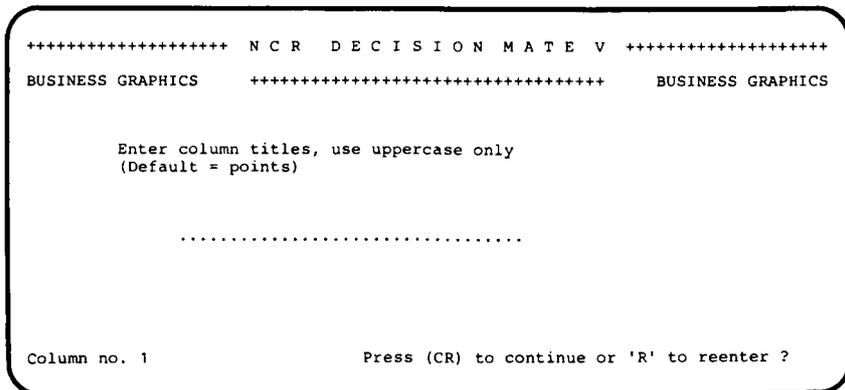


Abb. 4

Die nächste Eingabe betrifft die Zeilen (siehe Abb. 5).

```

+++++ N C R   D E C I S I O N   M A T E   V +++++
BUSINESS GRAPHICS   +++++ BUSINESS GRAPHICS

      Enter row(item) titles, use uppercase only
      (Default = points)

.....
Shade c.   **
Color     *

row(item) no.: 1          Press (CR) to continue or 'R' to reenter ?

```

Abb. 5

Row Titles:                    Bezeichnungen der Zeilen; bis zu 12 Großbuchstaben, u. U. weniger, je nach Anzahl der Zeilen

Shade:                        Druckmuster-Kennzahl (siehe nächsten Abschnitt "Grafik-Beispiele"); Eingabe 0-15 zulässig; Standardwert = 0

Hinweis: Die gelieferte VEGAS-Version kann keine Farben darstellen.  
Standardwert ist 7.

Anschließend müssen Sie den gewünschten Wert zu jeder (Tabel-)Position eingeben. Das Programm zeigt zu jeder Position Druckmuster und Farbe an. Ihre Angaben ersetzen die jeweils erschienenen Sternzeichen am Bildschirm (siehe Abb. 6).

```

                               ENTRY OF CHART VALUES

ITEM TITLE/CD/CO                COL 1
(item title, shade, color displayed)  *****

                               Press (CR) to continue or 'R' to reenter ?

```

Abb. 6

Values: Sie können zeilenweise die von Ihnen gewünschten Werte (nur Ganzzahlen  $H \leq 32760$ ) eingeben.

Ihre Dateneingabe für die Erstellung eines Diagramms ist somit vollständig. Das in Abb. 7 abgebildete Menu wird am Bildschirm ausgegeben. Sie können Ihre Daten auf Diskette sichern lassen (Eingabe 1), erneut vom Anfang an Daten eingeben (2), die Daten am Drucker ausgeben lassen (3), oder zum Hauptmenu zurückkehren (G).

```

+++++++ N C R   D E C I S I O N   M A T E V ++++++
BUSINESS GRAPHICS   ++++++ BUSINESS GRAPHICS

                                END   OF   DATA ENTRY
                                Save Date      (1)
                                Repeat Entry   (2)
                                List Data      (3)
                                Goto Main Menu  (G)      ?

```

Abb. 7

Am sinnvollsten ist zunächst die Ausgabe am Drucker, damit Sie die Gelegenheit haben, Ihre eingegebenen Daten zu prüfen. Dann erscheint folgende Meldung am Bildschirm:

```

+++++++ N C R   D E C I S I O N   M A T E V ++++++
BUSINESS GRAPHICS   ++++++ BUSINESS GRAPHICS

                                Listing data on printer

```

Abb. 8

Sobald der Drucker seine Arbeit beendet hat, wird das Auswahlmenu (Abb. 7) wiederholt. Wenn Ihre Daten keine Fehler enthalten, sollten Sie sie auf Diskette sichern. Falls Nachbesserungen erforderlich sind, bieten sich zwei Möglichkeiten an: Die Eingabe 2 bedeutet, daß Sie ganz von vorne anfangen wollen. Deshalb ist es bei nicht allzu umfangreichen Nachbesserungen sinnvoller, zuerst die Daten auf Diskette zu sichern (Eingabe 1). Sie können dann unter Verwendung

von "CHANGE DATA" vom Hauptmenu aus (siehe Abb. 1) die erforderlichen Änderungen vornehmen. Die genaue Funktionsweise wird später in diesem Abschnitt erläutert.

Bevor die Daten auf Diskette geschrieben werden, müssen Sie bestimmen, ob die zuletzt vom Programm angelegte Datei von den neuen Daten überschrieben werden, oder ob VEGAS eine neue, zusätzliche Datei für die Daten eröffnen soll. Nach der Frage am Bildschirm "REPLACE (R) LAST RECORD IN FILE OR WRITE (N) RECORDS?" müssen Sie also R (Überschreiben) bzw. N (Neu anlegen) eingeben. Die nächste Frage - "PRESS (CR) TO CONTINUE OR "R" TO REENTER?" - gibt Ihnen die Möglichkeit, diese Entscheidung durch Drücken von R zu revidieren. Wenn Sie mit Ihrer vorherigen Entscheidung einverstanden sind, drücken Sie einfach ↵. Das Programm teilt Ihrer Datei eine Nummer zu, die selbstverständlich auch Ihnen mitgeteilt wird (siehe Abb. 9).

```
+++++ N C R   D E C I S I O N   M A T E V +++++
BUSINESS GRAPHICS   +++++ BUSINESS GRAPHICS

                               Processing Data File

The TITLE: _____
is now stored unter TITLE NUMBER:  _

                               continue with (CR)
```

Abb. 9

Notieren Sie diese Nummer sowie den Verwendungszweck der soeben gesicherten Datei. Diese Information ist für eine zukünftige Benutzung dieser Datei unerlässlich. Von dem in Abb. 7 abgebildeten Menu aus können Sie durch Drücken von G zum Hauptmenu (Abb. 1) zurückkehren.

### Erstellen eines Diagramms

Im Hauptmenu wählen Sie die Funktion 1 "DISPLAY CHART" (s. Abb. 1). Die Aufgabe dieser Funktion besteht darin, nach Ihrem Wunsch ein Linien-, Balken- oder kreisförmiges, "kuchenähnliches" Diagramm am Bildschirm auszugeben. Wenn Daten, die zu einem

Diagramm gehören, sich noch im Maschinenspeicher befinden, werden Sie mit "USE CURRENT CHART? (Y/N)" gefragt, ob Sie diese Daten für das Diagramm benutzen wollen. Wenn ja, dann geben Sie Y ein. Das Menu für die Darstellungsformen (Abb. 11) wird dann ausgegeben. Wenn nein, oder im Falle, daß sich keine Daten im Maschinenspeicher befinden, werden Sie aufgefordert, die der gewünschten Datei von VEGAS zugeteilte Nummer einzugeben (Abb. 10).

```

+++++++ N C R   D E C I S I O N   M A T E V   ++++++
BUSINESS GRAPHICS   ++++++   BUSINESS GRAPHICS

Enter number of title in chart file
or enter 'R' to review all listed titles   ***

```

Abb. 10

Sollten Sie die Nummer der gewünschten Datei vergessen haben, können Sie durch Eingabe von R (mit anschließendem ↵) eine Liste sämtlicher von VEGAS gesicherten Dateien am Bildschirm ausgeben lassen. (Ihre Version von VEGAS enthält unter Nr. 2 Beispieldaten für ein Diagramm.) Nach Einsicht in diese Liste drücken Sie ↵. Es erscheint das Hauptmenu (Abb. 1), wo Sie noch einmal die Funktion 1 ("DISPLAY CHART") wählen können.

Sobald Sie dem Programm die Nummer der gewünschten Datei mitgeteilt haben, werden Sie gefragt, ob die Daten als Linien-(Eingabe 1), Balken- (2) oder kreisförmiges (3) Diagramm dargestellt werden sollen (Abb. 11).

```

+++++++ N C R   D E C I S I O N   M A T E V   ++++++
BUSINESS GRAPHICS   ++++++   BUSINESS GRAPHICS

          SELECT TYPE OF CHART
          LINE CHART   (1)
          BAR CHART   (2)
          PIE CHART   (3)
          Goto Main Menu (G)   ?

```

Abb. 11

Die Darstellung als Liniendiagramm erfordert zunächst die Eingabe einiger Parameter (Abb. 12).

```

+++++++ N C R   D E C I S I O N   M A T E V   ++++++
BUSINESS GRAPHICS   ++++++                               BUSINESS GRAPHICS

                Select Parameters for Display

Enter column no. for begin /end
Default = (CR) selects all          ** / **

Enter row no.(item) for begin / end
Default = (CR) allows mixed selection ** / **
Mixed selection `
Default = (CR) selects all          ** * * * * *
                                     ** * * * * *

Number of columns   ___
Number of rows(items) ___          Press (CR) to continue or 'R' to reenter ?

```

Abb. 12

Als erstes müssen Sie die erste und die letzte darzustellende Spalte (“column”) eingeben. Beispiel: Gesetzt den Fall, daß Ihre dem Diagramm zugrundeliegende Datei Daten für 4 Spalten enthält, daß Sie aber lediglich die Spalten 3 und 4 darstellen wollen. In diesem Fall müssen Sie 3 ↵ und dann 4 ↵ eingeben. Wenn Sie auf diese Eingaben durch Drücken von ↵ verzichten, werden alle Spalten im Diagramm erscheinen.

Der nächste Schritt ist die Eingabe der ersten und der letzten Zeile (“row”), die ausgegeben werden soll. Beispiel: Wenn Ihre Datei Daten für 3 Zeilen enthält, könne Sie wie bei den Spalten eine, zwei, oder alle drei ausgeben lassen. Das Drücken von ↵ bewirkt, daß alle Zeilen in die Darstellung einbezogen werden, es sei denn, Sie machen von dem darauffolgenden Angebot einer gemischten Zeilenauswahl (“Mixed selection”) Gebrauch. Hierdurch können Sie nach Belieben eine andere Reihenfolge der Zeilenausgabe, bestimmen. Beispiel: Sie könnten die Zeilen in der Reihenfolge 3-1-2 ausgeben lassen. Ein einfaches Drücken von ↵ ohne vorherige Bestimmung einer gemischten Zeilenauswahl sorgt endgültig dafür, daß alle Zeilen ausgegeben werden. Nach nochmaligem Drücken von ↵ erscheint das Diagramm.

Bevor ein Balkendiagramm ausgegeben werden kann, muß das Programm wissen, ob die Balken aufeinandergestapelt (“stacked” = Eingabe 1) oder nebeneinander (“side by side” = Eingabe 2) dargestellt werden sollen. Anschließend können Sie wie beim Liniendiagramm beschrieben die Parameter für Spalten und Zeilen eingeben (Ab. 13).

```

+++++ N C R   D E C I S I O N   M A T E   V   +++++
BUSINESS GRAPHICS  +++++ BUSINESS GRAPHICS

                Select Parameters for Display

                Rows(Item) stacked          (1)
                Rows(Item) side by side     (2)   ?

                Enter column no. for begin / end
                Default = (CR) selects all          ** / **

                Enter row no.(item) for begin / end
                Default = (CR) allows mixed selection ** / **
                Mixed Selection
                Default = (CR) selects all          ** * * * * * * * *
                                                         * * * * * * * *

Number of columns      ---
Number of rows(items) ---      Press (CR) to continue or 'R' to reenter ?

```

Abb. 13

Nach diesen Eingaben drücken Sie noch einmal ↵. Das gewünschte Diagramm erscheint am Bildschirm.

Die kreisförmige, "kuchenähnliche" Darstellung darf aus bis zu 12 "Schnitten" bestehen. Zunächst müssen Sie zwischen zwei verschiedenen Darstellungsarten wählen: Sie können bestimmen, ob eine vollständige Spalte ("Several rows ..." = Eingabe 1) oder eine Zeile quer durch mehrere Spalten ("Several columns ..." = Eingabe 2) Gegenstand der Darstellung sein soll. Wenn Sie sich für eine Spalte entscheiden (s. Abb. 14), geben Sie anschließend die Nummer dieser Spalte ein. Als nächstes erwartet das Programm wie bei der Linien- und Balkendarstellung Ihre Angaben zur ersten und letzten Zeile, die in die Darstellung einbezogen werden soll. Und wie bei der Linien- und Balkendarstellung besteht ebenfalls die Möglichkeit einer gemischten Zeilenauswahl.

```

+++++ N C R   D E C I S I O N   M A T E   V   +++++
BUSINESS GRAPHICS  +++++ BUSINESS GRAPHICS

                SELECT PARAMETERS FOR P I E   C H A R T   DISPLAY

                Several rows(items) for one column?  (1)
                Several columns for one row?          (2)

                Enter COLUMN NO. (Default = 1)

                Enter ROW NO.(item) for begin / end
                Default = (CR) allows mixed selection ** / **

                Mixed selection
                Default = (CR) selects all          ** * * * * * * * *
                                                         * * * * * * * *

Number of columns      ---
Number of rows(items) ---      Press (CR) to continue or 'R' to reenter ?

```

Abb. 14

Wenn Sie sich für eine Zeile quer durch mehrere Spalten entscheiden (Abb. 15), müssen Sie natürlich die Nummer der gewünschten Zeile eingeben. Danach machen Sie die übrigen Angaben zur Spaltenauswahl. Auch hier ist eine gemischte Auswahl möglich.

```

+++++++ N C R   D E C I S I O N   M A T E V   ++++++
BUSINESS GRAPHICS ***** BUSINESS GRAPHICS

SELECT PARAMETERS FOR P I E C H A R T   D I S P L A Y

Several rows(items) for one column?      (1)
Several columns for one row?             (2)

Enter ROW NO.(item)   (Default = 1)

Enter COLUMN NO. for begin / end
Default = (CR) allows mixed selection    ** / **

Mixed selection
Default = (CR) selects all                ** * * * * *
                                           ** * * * * *

Number of columns      —
Number of rows(items) —   Press (CR) to continue or 'R' to reenter ?

```

Abb. 15

Bei der "kuchenähnlichen" Darstellung wird rechts des kreisförmigen Diagramms eine Lesehilfe ausgegeben. Hier können Sie die Bezeichnung der gewählten Spalte bzw. Zeile, die dazugehörigen Werte sowie deren Anteil in % des Ganzen erfahren. Anteile, die einzeln weniger als 1,5 % des Ganzen ausmachen, werden in dem eigentlichen kreisförmigen Diagramm unter der Bezeichnung "MISC" (= sonstige) zusammengefaßt. Die Lesehilfe listet aber solche Werte einzeln auf.

Am Bildschirm unten links werden Sie bei jedem Diagramm einen Hinweis auf zusätzliche Ausgabemöglichkeiten bemerken. Durch die Eingabe von C können Sie eine sogenannte "Inversion" der Bildschirmausgabe bewirken: Zeichen und Zeichnungen die vorher hell auf dunklem Hintergrund abgebildet wurden, erscheinen nun dunkel auf hellem Hintergrund. Ein nochmaliges Eingeben von C macht diese Inversion rückgängig. Mit P oder D wird Ihr Diagramm am Drucker ausgegeben, wobei D einen besonders hervorstechenden Druck bewirkt.

Falls Sie einen Plotter angeschlossen haben, erscheinen die Zeichen und Zeichnungen schwarz auf weißem Hintergrund, wenn die Inversion der Bildschirmausgabe außer Kraft ist.

Bei der kreisförmigen Darstellung kann auf Wunsch ein bestimmter Schnitt des "Kuchens" herausgetrennt werden. Dazu muß die Bildschirmausgabe in invertiertem Zustand sein. Sie müssen dann lediglich die Nummer des herauszutrennenden Schnittes mit ent-

sprechendem Tastendruck angeben. Sie können beliebig viele Schnitte heraustrennen lassen. Diese können aber während der laufenden Diagrammausgabe nicht wieder eingesetzt werden.

## Änderung von Daten

Die dritte Auswahlmöglichkeit im Hauptmenu (Abb. 1) - "CHANGE DATE" - ermöglicht die Änderung von Daten, die bereits vom Programm als Datei auf Diskette geschrieben wurden. Geben Sie 3 ein, um diese Funktion aufzurufen. Wenn Daten, die zu einem Diagramm gehören, sich noch im Maschinenspeicher befinden, werden Sie mit "USE CURRENT CHART?(Y/N)" gefragt, ob Sie diese Daten ändern wollen. Wenn ja, dann geben Sie Y ein. Das Änderungsverfahren wird dann eingeleitet (Abb. 17). Wenn nein, oder im Falle, daß sich keine Daten im Maschinenspeicher befinden, werden Sie aufgefordert, die der gewünschten Datei von VEGAS zugeteilte Nummer einzugeben (Abb. 16).

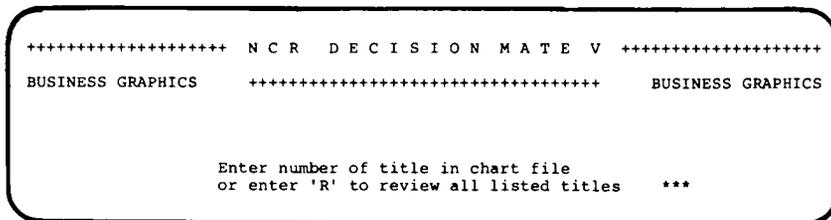


Abb. 16

Sollten Sie die Nummer der gesuchten Datei vergessen haben, können sie durch Eingabe von R (mit anschließendem ↵) eine Liste sämtlicher von VEGAS gesicherten Dateien am Bildschirm ausgeben lassen. Nach Einsicht in diese Liste drücken Sie ↵. Es erscheint das Hauptmenu (Abb. 1), wo Sie noch einmal die Funktion 3 ("CHANGE DATA") wählen können.

Die bisherigen Daten für Ihr Diagramm erscheinen einzeln am Bildschirm. Sie haben die Gelegenheit, an Stelle der Schreibmarke (oft als "Cursor" bezeichnet) eine Änderung einzugeben. An Stellen wo keine Änderung getätigt werden soll, drücken Sie einfach ↵. Die Schreibmarke springt dann zum nächsten Datenfeld. Es darf lediglich die Anzahl der Spalten sowie der Zeilen nicht geändert werden. Ferner sollten Sie beachten, daß eine pauschale Änderung der Größenordnung der Werte ("Unit of measure") eine Änderung sämtlicher Einzelwerte erforderlich macht.

Die folgenden Abbildungen geben Ihnen detaillierten Aufschluß über die Änderungsmöglichkeiten.

Sie haben die Möglichkeit, die übergeordnete sowie die zweite Überschriftszeile zu ändern (Abb. 17).

```
+++++++ N C R   D E C I S I O N   M A T E   V   ++++++
BUSINESS GRAPHICS      ++++++                BUSINESS GRAPHICS

The process for changing parameters and/or values is started.
It is impossible to change the number of columns and rows.
To change enter data / (CR) = no change

First Title
_____

Second Title
_____

Unit of measure  _
Name of unit    _____

Press (CR) to continue or 'R' to reenter ?
```

Abb. 17

Änderungen der Spaltenbezeichnungen (Abb. 18) sowie der Druckmuster und Farben der einzelnen Zeilen (Abb. 19) sind ebenfalls zulässig.

```
+++++++ N C R   D E C I S I O N   M A T E   V   ++++++
BUSINESS GRAPHICS      ++++++                BUSINESS GRAPHICS

To change column titles enter column no. / (CR) = no change

Column No.      Text
**
```

Abb. 18

```
+++++++ N C R   D E C I S I O N   M A T E   V   ++++++
BUSINESS GRAPHICS      ++++++                BUSINESS GRAPHICS

To change rows(items) enter row no. / (CR) = no change

ROW NO.      TEXT      SHADE CODE      COLOR
**
```

Abb. 19

Zum Schluß können sie die Werte der einzelnen (tabellarischen) Positionen ändern (Abb. 20). Geben Sie die Nummer für Spalte und

Zeile und den neuen Wert für diese Position ein. Nach der letzten Änderung geben Sie  $\leftarrow$  ohne eine Zahl ein.

```

+++++++ N C R   D E C I S I O N   M A T E   V ++++++
BUSINESS GRAPHICS  ++++++                BUSINESS GRAPHICS

      To change the value of a row(item) within a column enter
      the column no. and the row no. / (CR) = no change

              Column No. Row No.   Value
              **

```

Abb. 20

Nach Abschluß der oben beschriebenen Änderung erscheint folgendes Menu:

```

+++++++ N C R   D E C I S I O N   M A T E   V ++++++
BUSINESS GRAPHICS  ++++++                BUSINESS GRAPHICS

              End of Change

              Save Changes   (1)
              Repeat Changes (2)
              List Changes   (3)
              Goto Main Menu (G) ?

```

Abb. 21

Sie können durch Drücken von

- 1 die geänderte Datei auf Diskette sichern
- 2 das Änderungsverfahren erneut einleiten
- 3 die Änderungen am Drucker ausgeben lassen

G zum Hauptmenu (Abb. 1) zurückkehren.

Nachdem Sie die Änderungen mit Hilfe einer Druckausgabe noch einmal überprüft haben, können Sie Ihre Datei in der neuen geänderten Form auf Diskette sichern. Das Sichern wird dann am Bildschirm bestätigt (Abb. 22).

```

+++++++ N C R   D E C I S I O N   M A T E   V ++++++
BUSINESS GRAPHICS  ++++++                BUSINESS GRAPHICS

              Processing Data File

              Record is replaced

              continue with (CR)

```

Abb. 22

Nach abschließendem Drücken von ↵ erscheint das in Abb. 22 abgebildete Menu wieder. Die Weiterverarbeitung in VEGAS erfolgt nach der Rückkehr zum Hauptmenu.

### Grafik-Beispiele

#### LIST OF VALUES FOR GRAPHIC-CHART

Title: PERSONAL COMPUTER SALES

Second Title: FIRST QUARTER 1983

Value: 2

Name of Units: MACHINES

Number of COLUMNS / ROWS: 3 / 2

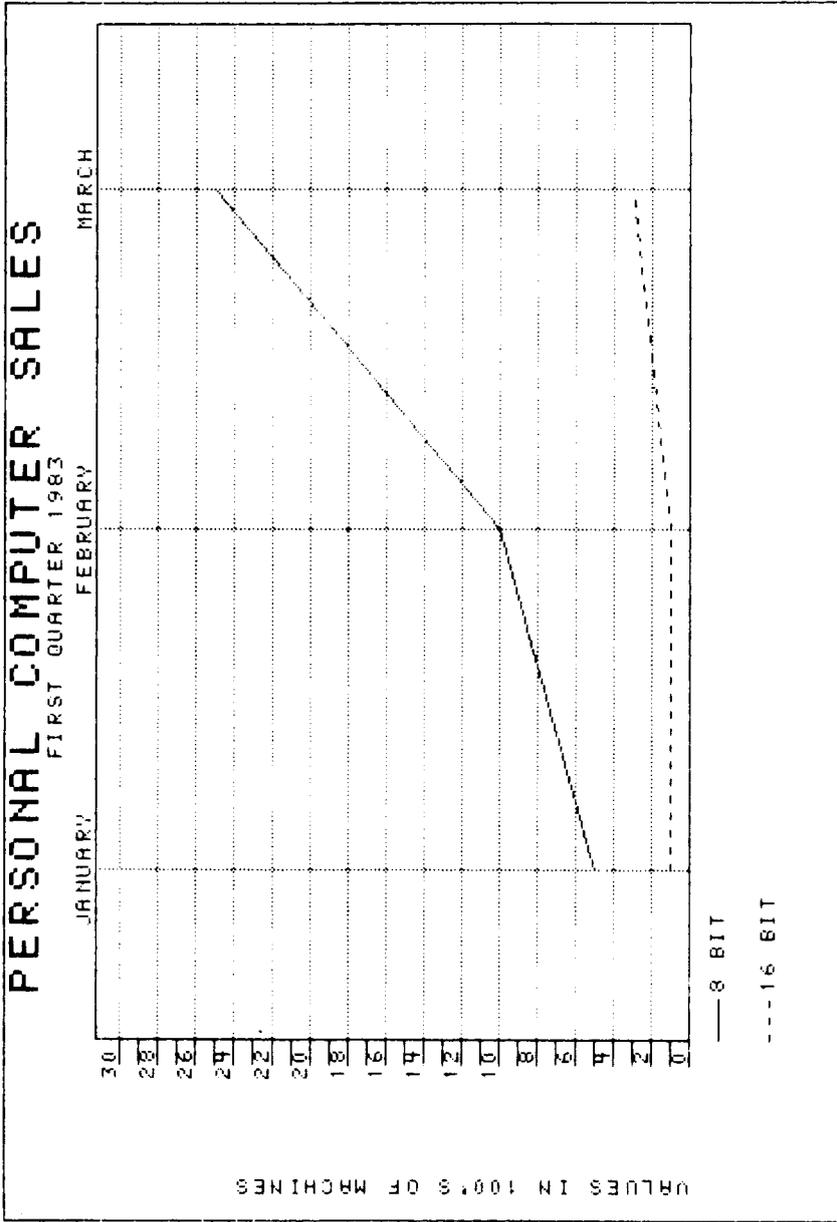
COLUMN / COLUMN-NO: 1 / 2

1 / 8 BIT

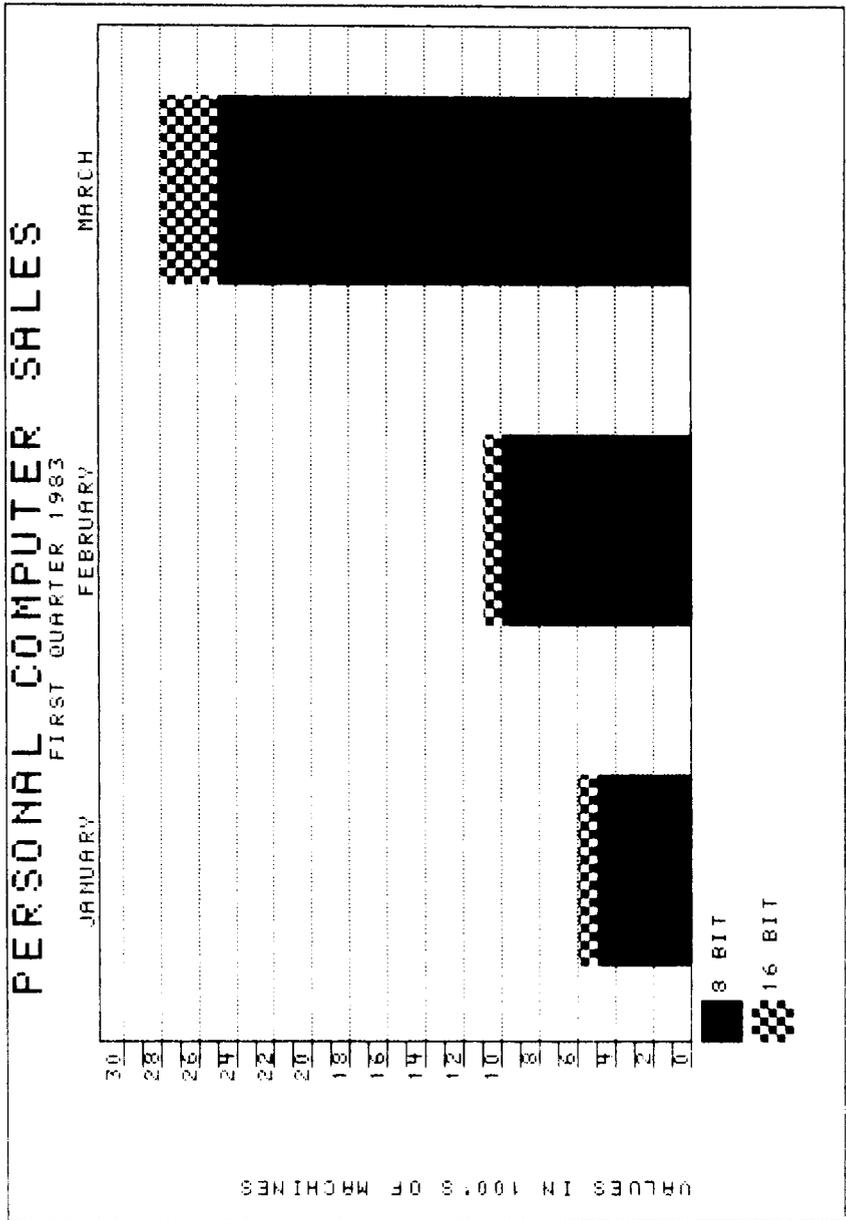
2 / 16 BIT

Month	Column 1	Column 2	Column 3
JANUARY	1	5	1
FEBRUARY	2	10	1
MARCH	3	25	3

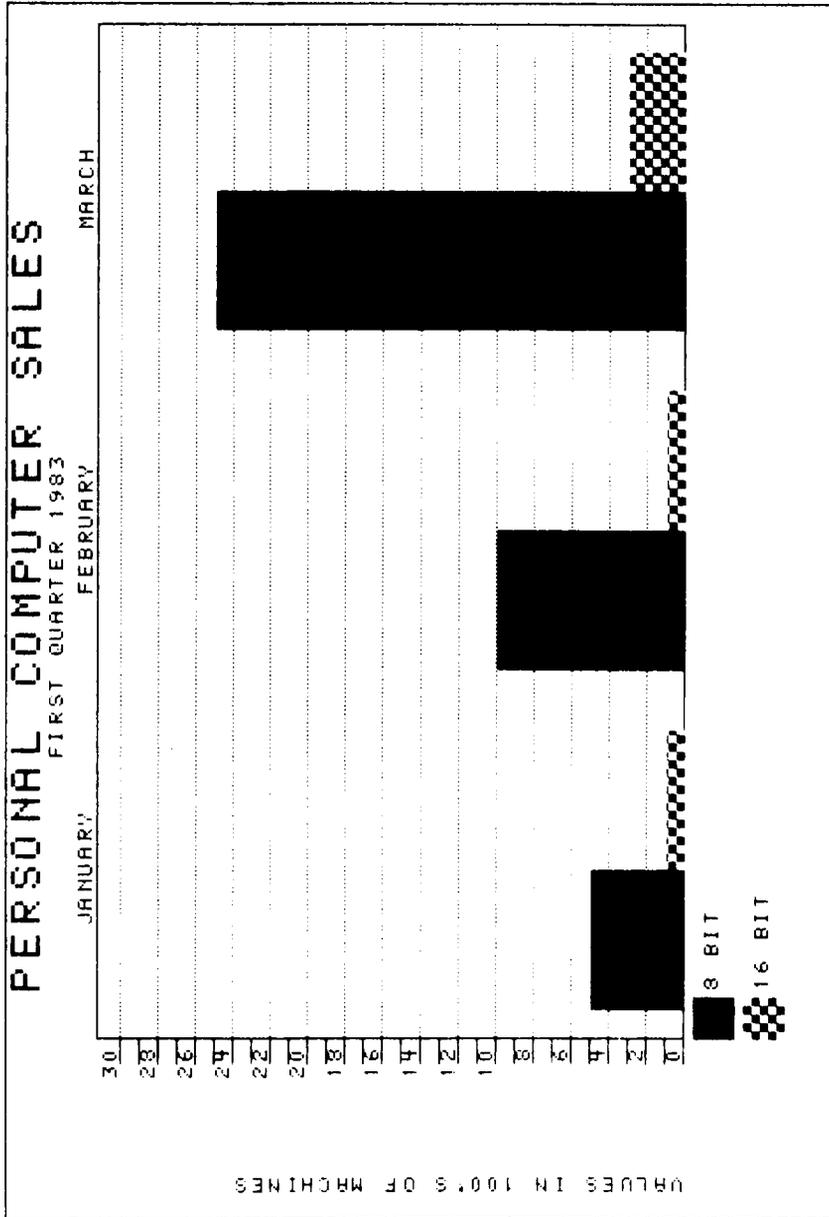
Datenausgabe



Liniendiagramm



Balkendiagramm (gestapelt)



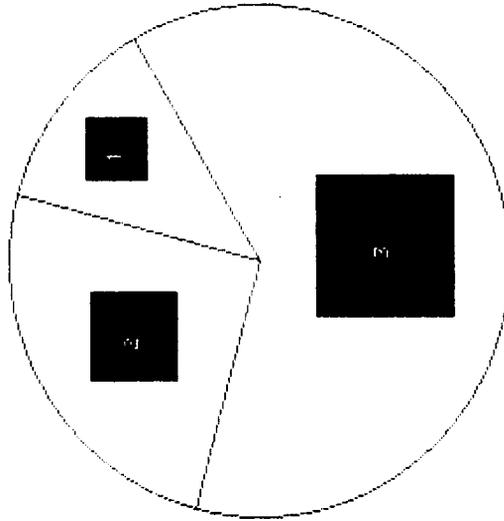
Balkendiagramm (nebeneinander)

# PERSONAL COMPUTER SALES

FIRST QUARTER 1983

8 BIT

LEGEND	
1	JANUARY 5
2	FEBRUARY 10
3	MARCH 25

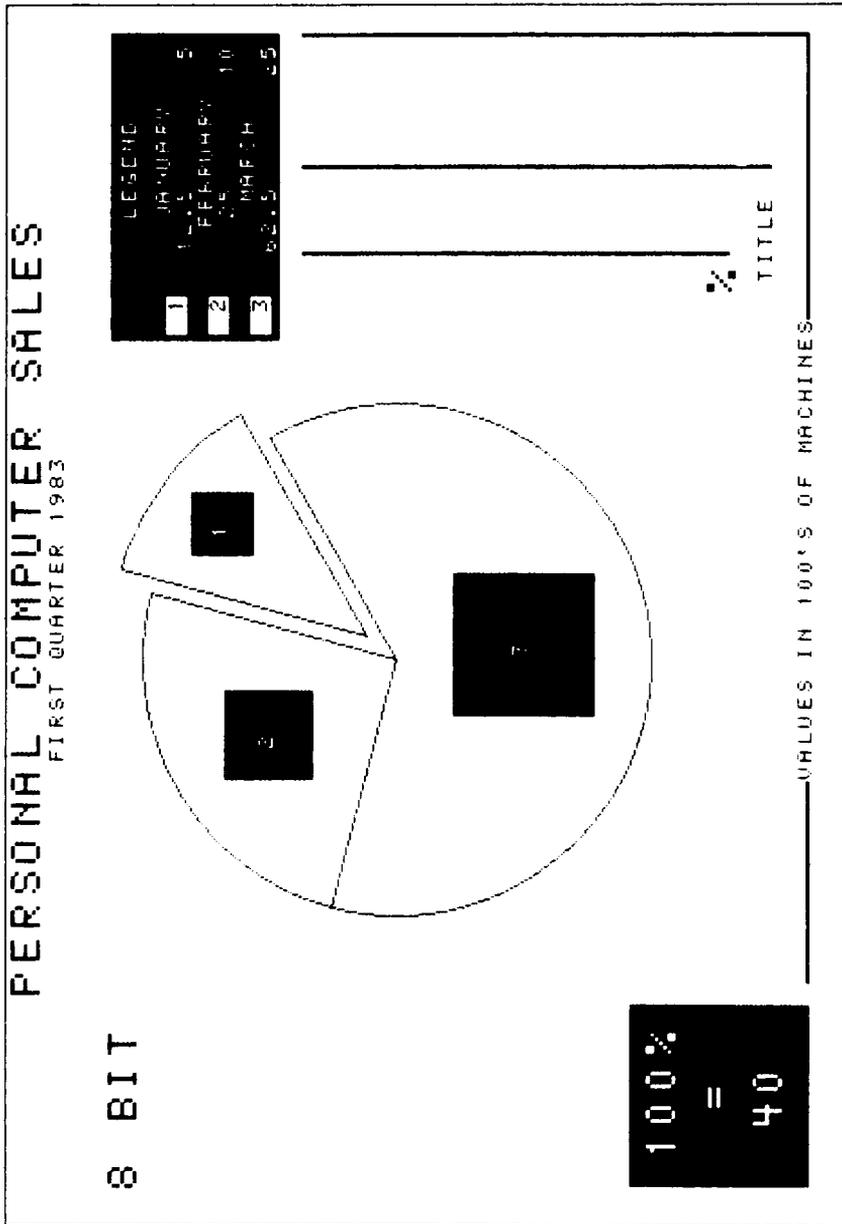


100%  
=  
40

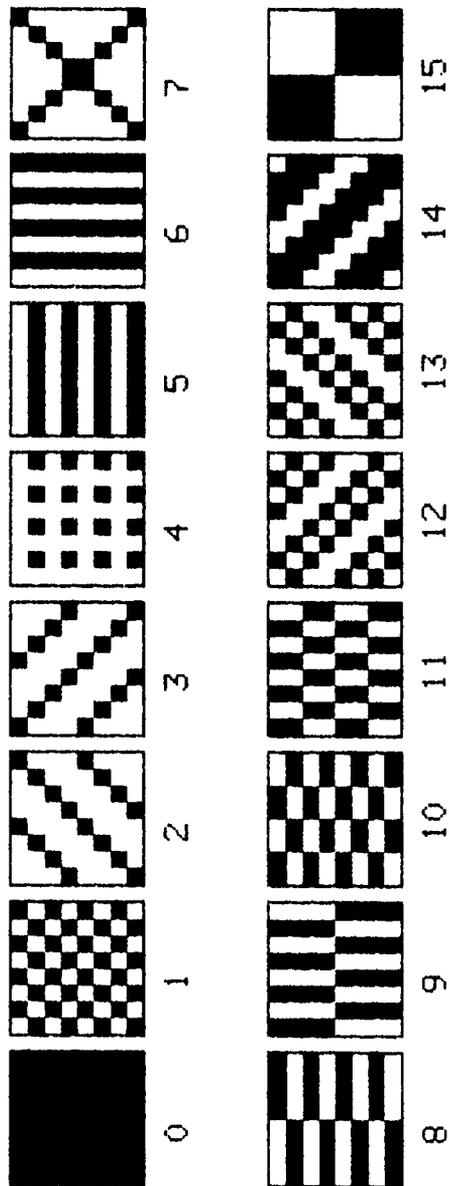
100%  
TITLE

VALUES IN 100'S OF MACHINES

Kreisförmiges Diagramm („Kuchendarstellung“)



Kreisförmiges Diagramm (mit "Kuchenschnitt")



Druckmuster mit Kennzahlen

( ( (

## FEHLERMELDUNGEN

Wenn Sie versuchen, eine ungültige Eingabe zu tätigen (z. B., Sie geben Buchstaben in ein Datenfeld ein, wofür eine Zahleneingabe erforderlich ist), wird diese Eingabe ohne ausdrückliche Fehlermeldung zurückgewiesen. VEGAS wartet dann auf eine gültige Eingabe.

Eine Fehlermeldung auf Systemebene kann vorkommen, während Sie versuchen, Daten für ein neues Diagramm auf Diskette zu sichern. VEGAS verfügt über zwei Dateien namens TITLE-F und VALUE-F, in denen sämtliche Daten für Ihre Diagramme gespeichert werden. Wenn kein Platz für eine Erweiterung dieser Dateien vorhanden ist, erscheint folgende Meldung:

Disk Full at address xxxx

Ihr NCR DECISION MATE V kehrt zur Systemebene zurück und zeigt das Bereitschaftszeichen an. Sie haben nun drei Möglichkeiten zur Wahl:

1. Löschen der Dateien TITLE-F und VALUE-F, um ihren Disketten-Speicherplatz freizugeben. (Das würde natürlich bedeuten, daß die Daten für sämtliche Diagramme vernichtet werden.)
2. Kopieren der Dateien TITLE-F und VALUE-F auf eine andere Diskette. (Die Daten werden aufbewahrt.)
3. Kopieren sämtlicher VEGAS-Dateien auf eine andere Diskette. Sie können dann diese zweite Diskette für zukünftige VEGAS-Anwendungen benutzen.

Die VEGAS-Dateien erkennen Sie daran, daß ihre ersten zwei Buchstaben jeweils VE sind, und daß sie immer den Namenszusatz COM aufweisen. Wenn Sie sich für die dritte Möglichkeit entscheiden, müssen Sie ebenfalls die Datei BRUN.COM auf die zweite Diskette kopieren. Das folgende Beispiel zeigt, wie die VEGAS-Dateien vom Laufwerk A auf die Diskette im Laufwerk B kopiert werden können:

```
PIP B: = A:VE*.COM  
PIP B: = A:BRUN.COM
```

**Wichtig:** Bei der Benutzung dieser Möglichkeit sollten Sie die Dateien

TITLE-F und VALUE-F nicht auf die zweite Diskette kopieren. Diese Dateien werden nämlich auf der zweiten Diskette von VEGAS automatisch neu erstellt.

## WIE BEKOMME ICH SOFTWARE?

### Die richtige Anschaffung

Ihr NCR DECISION MATE V ist in höchstem Grade anpassungsfähig. Sie können beinahe jedes Anwendungs-Programm sowie jede Programmiersprache in Ihrem Computer zur Ausführung bringen. Die dazugehörige Software muß lediglich CP/M-kompatibel sein. Es gibt drei verschiedene Kategorien von Software:

- Software, die von NCR angeboten wird
- Software, die von allen CP/M-Anwendern bezogen werden kann und auch im DECISION MATE V von NCR geprüft wurde
- Frei erhältliche Software, die von NCR nicht geprüft wurde.

Sie können Software der ersten Kategorie direkt über NCR oder Ihren NCR-Verkaufsbeauftragten oder -Einzelhändler beziehen. Software der anderen Kategorie, insbesondere Software "von der Stange" der dritten Kategorie, ist bei allen namhaften Software-Anbietern erhältlich. In letzterem Fall ist es aber wichtig zu wissen, ob diese Software für Ihren Computer geeignet ist. Bevor Sie sich für bestimmte Software entscheiden, sollten Sie die folgenden Erläuterungen zu Software-Kompatibilität sorgfältig lesen. Mit diesen Informationen und Fragen können Sie sich an jeden fachkundigen Software-Vertrieb wenden.

1. Erkundigen Sie sich, ob die Diskette im NCR-Format vorhanden ist. Wenn das der Fall ist, können Sie die Diskette ohne weiteres mit Ihrem NCR DECISION MATE V verwenden. Andernfalls fragen Sie, ob die Diskette eines der folgenden Formate hat:

	SS=single sided DS=double sided	DD = double density
● ITT 3030	DS	DD
● DEC VT 180	SS	DD
● ZENITH Z100 Type 1	DS	DD
● ZENITH Z100 Type 2	DS	DD
● TA ALPHATRONIC P2U	SS	DD
● OSBORNE 1	SS	DD
● KayPro	SS	DD
● IBM	SS	DD
● IBM	DS	DD
● Siemens PC-16	DS	DD
● Epson QX-10	DS	DD
● DEC /Rainbow (for 96 track/ inch drives only)	DS	DD

**Hinweis:** "Double sided" = doppelseitige Verwendung der Diskette. "Single sided" = einseitige Verwendung der Diskette. "Single density" = einfache Speicherdichte. "Double density" = verdoppelte Speicherdichte.

Nach Aufruf von EXCHANGE können Sie die oben genannten Disketten mit Ihrem DECISION MATE V verwenden.

2. Falls die Software auf 8-Zoll-Diskette vorhanden ist, könnten Sie fragen, ob diese Software auf eine 5 1/4-Zoll-Diskette im NCR-Format oder eines der obengenannten Formate übertragen werden kann.
3. Der Einsatz von Anwendungs-Software sowie Programmiersprachen-Software erfordert häufig ein zusätzliches Programm, um die Schnittstelle zu einer spezifischen Hardware-Konfiguration herzustellen. Fragen Sie deshalb, ob die Software diesbezüglich für Ihren DECISION MATE V geeignet ist. Wenn das nicht der Fall ist, wäre es trotzdem möglich, daß die angebotene Software eine Programmierhilfe enthält, mit der Sie diese Anpassung an die spezifische Konfiguration (Installation) selbst herstellen können.

Nach der Beschaffung der Software müssen Sie nur ggf. das EXCHANGE-Programm aufrufen. Ihre Software ist dann einsatzbereit. Die genaue Funktionsweise des EXCHANGE-Programms wird im nächsten Abschnitt erklärt. Sie sollten dann ebenfalls Teil A des Anhangs "Software Installation" lesen.

## **1.6 Verwendung des EXCHANGE-Dienstprogrammes**

EXCHANGE ermöglicht die Benutzung einer Nicht-NCR-Diskette in Ihrem DECISION MATE V. Unbedingte Voraussetzung ist, daß die Diskette eine 5 1/4-Zoll-Diskette ist. EXCHANGE bewirkt keine Änderung der Diskette.

### **1.6.1 Aufruf von EXCHANGE in einem System mit zwei Disketten-Laufwerken**

**Hinweis:** Wenn Ihr System nur ein Disketten-Laufwerk besitzt, sollten Sie diesen Abschnitt überspringen, Sie können dann im Abschnitt 1.6.2 "Aufruf von EXCHANGE in einem System mit einem Disketten-Laufwerk und einem oder mehreren Festplatten-Laufwerken" weiterlesen.

Das EXCHANGE-Programm wird gemäß folgender Beschreibung ausgeführt:

1. Legen Sie die NCR-CP/M-Systemdiskette in Disketten-Laufwerk A ein. (Die Dienstprogramme EXCHANGE, STAT, und PIP müssen auf der Systemdiskette vorhanden sein. Sie werden bei CP/M-Software von NCR immer mitgeliefert.)
2. Betätigen Sie die Funktionstaste F9, um EXCHANGE aufzurufen.

Nach der Überschrift "EXCHANGE" und einer Copyright-Mitteilung erscheint das EXCHANGE-UTILITY-Hauptmenu. Dieses nachstehend abgebildete Menu zeigt Ihnen, welche Fremdisketten-Formate Ihrem Computer angepaßt werden können.

```

EXCHANGE UTILITY

X---EXIT
A---ITT 3030 DD DS
B---DEC VT180 DD DS
C---ZENITH Z100 Type 1 DD DS
D---ZENITH Z100 Type 2 DD DS
E---TA ALPHATRONIC P2U DD SS
F---OSBORNE 1 DD SS
G---KayPro DD SS
H---IBM DD SS
I---IBM DD DS
J---Siemens PC-16 DD DS
K---Epson QX-10 DD DS
L---DEC / Rainbow DD DS (96 TPI)

Enter your selection:

```

Hinweis: Die Zeile: L---DEC/Rainbow DD DS (96TPI) erscheint nur, wenn Ihr System mit 96TPI Laufwerken arbeitet.

3. Geben Sie den dem Format der Fremdiskette entsprechenden Buchstaben ein. Zum Beispiel, F bedeutet, daß Sie eine OSBORNE 1 (double density, single sided)-Diskette haben. Nach Ausführung von EXCHANGE erscheint

EXCHANGE was successful - selected Typ = F

4. Mit Eingabe von X können Sie EXCHANGE verlassen. Das Bereitschaftszeichen A> erscheint dann am Bildschirm.
5. Legen Sie die Nicht-NCR-Diskette (in unserem Beispiel: OSBORNE 1-Diskette) in Laufwerk B ein.
6. Sie können nun die Dateien der Fremddiskette im Laufwerk B auf die NCR-Diskette im Laufwerk A kopieren. Während des Kopiervorgangs werden die Dateien der Fremddiskette in NCR-Format umgewandelt. Deshalb ist bei späterer Verwendung der

“ehemaligen” Fremddateien auf der NCR-Diskette ein nochmaliger Aufruf von EXCHANGE nicht erforderlich. Bevor Sie aber den Kopiervorgang ausführen können, müssen Sie prüfen, ob auf der NCR-Diskette im Laufwerk A genug freier Speicherplatz vorhanden ist, um die Dateien der Fremddiskette im Laufwerk B aufzunehmen. Dies können Sie erfahren, indem Sie die Funktionstaste F10 betätigen. Der noch zur Verfügung stehende Speicherplatz wird in der Form der Meldung “Bytes Remaining on A: x k” angegeben; “x” entspricht der Anzahl der Kilobyte. Sie sollten diese Nummer notieren.

7. Nach Drücken von F11 erfahren Sie den Bedarf an Speicherplatz der Nicht-NCR-Dateien auf der Diskette im Laufwerk B. Sie müssen nun die verschiedenen Dateigrößen (in Kilobyte angegeben) zusammenaddieren, wobei eine etwaige ungerade Zahl von KByte auf die nächsthöhere gerade Zahl aufzurunden ist. (Beispiel: Bei einer angegebenen Dateigröße von 3 müssen Sie 4 statt 3 berechnen.) Die so errechnete Endsumme muß mit der bei 6. notierten Zahl verglichen werden. Wenn die bei 6. notierte Zahl größer als die soeben errechnete Summe ist, können Sie bei 8. weiterverfahren. Ansonsten müssen Sie bei 9. weiterverfahren.
8. Nach Drücken der Funktionstaste F8 werden die Dateien der Diskette im Laufwerk B auf die Diskette im Laufwerk A kopiert. Die Namen der so kopierten Dateien erscheinen am Bildschirm. Nach Beendigung des Kopiervorgangs erscheint das Bereitschaftszeichen A>. Sie möchten nun wahrscheinlich die Inhaltsverzeichnisse der zwei Disketten vergleichen. Hierzu drücken Sie die Funktionstaste F1 und anschließend die Funktionstaste F5. Nachher können Sie bei 10. weiterverfahren.
9. Sie haben bei 7. festgestellt, daß der auf der Diskette im Laufwerk A noch zur Verfügung stehende Speicherplatz nicht ausreichend ist, um alle Dateien der Diskette im Laufwerk B aufzunehmen. Sie können aber Dateien der Fremddiskette einzeln auf die NCR-Diskette im Laufwerk A kopieren. Sie müssen lediglich für jede Datei ein PIP-Kommando in der Form PIP A:filename.typ =B:filename.typ eingeben. Wenn der Speicherplatz der Diskette im Laufwerk A vollständig belegt ist, und Sie weitere Dateien von der Nicht-NCR-Diskette im Laufwerk B kopieren möchten, müssen Sie eine zweite Systemdiskette, auf der noch freier Speicherplatz vorhanden ist, hinzuziehen. Diese zweite Systemdiskette muß natürlich das PIP-Programm enthalten. Sie können dann wie mit der ersten Systemdiskette die noch verbleibenden Dateien der Nicht-NCR-Diskette kopieren. Vergessen Sie nicht die Dis-

ketten mit einer Aufschrift auf Etikett zu kennzeichnen und Sicherungskopien anzufertigen.

10. Die Nicht-NCR-Diskette können Sie als Sicherungskopie betrachten. (Mit Hilfe von EXCHANGE können ihre Dateien jederzeit in NCR-Format umgewandelt werden.) Die Änderungen im Betriebssystem, die durch den Ablauf von EXCHANGE veranlaßt wurden, bleiben bis zum nächsten EXCHANGE-Aufruf oder Kaltstart wirksam. Ferner dürfen Sie nicht vergessen, daß bei jedem Diskettenwechsel ein Warmstart erforderlich ist.

### 1.6.2 Aufruf von EXCHANGE in einem System mit einem Disketten-Laufwerk und einem oder mehreren Festplatten-Laufwerken

Das EXCHANGE-Programm wird gemäß folgender Beschreibung ausgeführt.

1. Legen Sie die NCR-CP/M-Diskette in Disketten-Laufwerk A ein. Das EXCHANGE-Dienstprogramm muß auf dieser Systemdiskette vorhanden sein. Das PIP-Dienstprogramm muß auf der Festplatte (in dem folgenden Beispiel: Laufwerk B) vorhanden sein. Diese Dienstprogramme wurden mit Ihrer CP/M-Software mitgeliefert.

**Hinweis:** Sollte das PIP-Programm sich nicht auf der Festplatte befinden, geben Sie das Kommando `A> PIP B: = A: PIP.COM[V]` ein, und drücken Sie anschließend `↵`. PIP wird auf diese Weise auf die Festplatte kopiert.

2. Betätigen Sie die Funktionstaste F9, um EXCHANGE aufzurufen. Nach der Überschrift "EXCHANGE" und einer Copyright-Mitteilung erscheint das EXCHANGE-UTILITY-Hauptmenu. Dieses nachstehend abgebildete Menu zeigt Ihnen, welche Fremddisketten-Formate Ihrem Computer angepaßt werden können.

EXCHANGE UTILITY

X---EXIT  
A---ITT 3030 DD DS  
B---DEC VT180 DD DS  
C---ZENITH Z100 Type 1 DD DS  
D---ZENITH Z100 Type 2 DD DS  
E---TA ALPHATRONIC P2U DD SS  
F---OSBORNE 1 DD SS  
G---KayPro DD SS  
H---IBM DD SS  
I---IBM DD DS  
J---Siemens PC-16 DD DS  
K---Epson QX-10 DD DS  
L---DEC / Rainbow DD DS (96 TPI)

Enter your selection:

3. Geben Sie die dem Format der Fremddiskette entsprechenden Buchstaben ein. Zum Beispiel, F bedeutet, daß Sie eine OSBORNE 1 (double density, single sided) -Diskette haben. Nach Ausführung von EXCHANGE erscheint

EXCHANGE was successful - selected Typ = F

4. Geben Sie X ein, um EXCHANGE zu verlassen. Anschließend erscheint die Aufforderung

Mount "OSBORNE 1" Disk in Drive A (CR)

5. Legen Sie die Nicht-NCR-Diskette (in unserem Beispiel: OSBORNE 1-Diskette) in Laufwerk A ein. Drücken Sie ↵ .

**Wichtig:** Ab jetzt dürfen Sie keinen Warmstart ausführen.

6. Schalten Sie auf das Laufwerk B um, indem Sie B: eingeben und dann ↵ drücken.
7. Die Dateien der Nicht-NCR-Diskette im Laufwerk A können nun auf die Festplatte B kopiert werden. Während des Kopiervorgangs werden die Dateien der Fremddiskette in NCR-Format umgewandelt. Deshalb ist bei späterer Verwendung der "ehemaligen" Fremddateien auf der Festplatte ein nochmaliger Aufruf von EXCHANGE nicht erforderlich.

Nach Drücken der Funktionstaste F4 werden die Dateien der Diskette im Laufwerk A auf die Festplatte (Laufwerk B) kopiert. Die Namen der so kopierten Dateien werden am Bildschirm ausgegeben. Anschließend erscheint die Aufforderung

A: MOUNT O.S. DISK (CR)

8. Legen Sie die Systemdiskette in Laufwerk A ein, und drücken Sie ↵. Darauf erscheint das Bereitschaftszeichen B>.
9. Durch folgende Eingabe schalten Sie wieder auf Laufwerk A zurück:

A: ↵

10. Zur Bestätigung des Kopiervorgangs drücken Sie die Funktionstaste F5. Sämtliche Dateien der Festplatte, darunter auch die von der Fremddiskette aufgenommenen Dateien, werden aufgelistet.
11. Die Nicht-NCR-Diskette können Sie als Sicherungskopie betrachten. (Mit Hilfe von EXCHANGE können ihre Dateien jederzeit in NCR-Format umgewandelt werden.) Die Änderungen im Betriebssystem, die durch den Ablauf von EXCHANGE veranlaßt wurden, bleiben bis zum nächsten Warmstart wirksam. Für jede Nicht-NCR-Diskette ist ein nochmaliger Aufruf von EXCHANGE erforderlich.

)

)

)

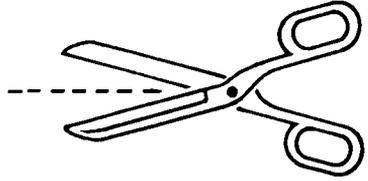
(

(

(

DIR A:	FORMAT	SYSGEN	PIP B:= A:.*.* [V]	DIR B:	CONFIG	DISCIT	PIP A:= B:.*.* [V]	EXCHANGE	STAT A:
--------	--------	--------	-----------------------	--------	--------	--------	-----------------------	----------	---------

STAT B:.*.*					LADDER	CATCHUM	DEMOS	MUSIC	VEGAS
-------------	--	--	--	--	--------	---------	-------	-------	-------



1

2

3

# KOMMANDOS

## 2.1 Datei-Bezeichnungen

Eine Datei-Bezeichnung identifiziert eine bestimmte Datei oder eine Gruppe von Dateien auf einer in das CP/M-System einbezogenen Platte. Diese Datei-Bezeichnungen sind entweder „eindeutig“ (ufn) oder „mehrdeutig“ (afn). Eine eindeutige Datei-Bezeichnung identifiziert genau eine Datei, während eine mehrdeutige Datei-Bezeichnung für eine Anzahl verschiedener Dateien zutrifft.

Datei-Bezeichnungen bestehen aus zwei Teilen, dem Primärdateinamen und dem Dateityp. Obwohl der Dateityp optional ist, dient er im allgemeinen als Gattungsname; der Dateityp „ASM“ z. B. wird benutzt, um die Datei als Quelldatei in Assemblersprache zu kennzeichnen, während der Primärdateiname eine bestimmte Quelldatei von einer anderen unterscheidet. Die beiden Namensarten werden durch einen Punkt („.“) voneinander getrennt, wie nachstehend gezeigt ist:

Dateiname.Typ

wobei Dateiname der aus acht oder weniger Zeichen bestehende Primärdateiname und Typ der Dateityp ist, der aus nicht mehr als drei Zeichen zusammengesetzt ist. Wie oben erwähnt, ist der Name

Dateiname

ebenfalls zulässig und entspricht einem Dateityp bestehend aus drei Leerzeichen. Die Zeichen, die zur Spezifizierung einer eindeutigen Datei-Bezeichnung verwendet werden, dürfen keines der Sonderzeichen

< > . , ; : = ? \* [ ] % | ( ) X \_

sein, während alle alphanumerischen Zeichen und die verbleibenden Spezialzeichen erlaubt sind.

Eine mehrdeutige Datei-Bezeichnung wird benutzt zum Durchsuchen des Directory und zum Zeichenmustervergleich. Die Form einer mehrdeutigen Datei-Bezeichnung ist ähnlich der einer eindeuti-

gen Bezeichnung, außer daß das „?“-Symbol an allen Stellen des Primär- und Sekundärnamens eingestreut werden kann. In verschiedenen Kommandos steht das „?“-Symbol für jedes Zeichen eines Dateinamens, das in der „?“-Position steht. So wird die mehrdeutige Bezeichnung

X?Z.C?M

erfüllt durch die eindeutigen Dateinamen

XYZ.COM

und

X3Z.CAM

Man beachte, daß die mehrdeutige Bezeichnung

\*.\*

gleichwertig ist der mehrdeutigen Bezeichnung

?????????.???

während

filename.\*

und

\*.typ

Abkürzungen sind für

filename.???

und

?????????.typ

Beispielsweise wird

A>DIR \*.\*

vom CCP als Befehl interpretiert, die Namen aller Plattendateien im Directory aufzulisten, während

A>DIR X.Y

nur nach einer Datei mit dem Namen X.Y. sucht. In ähnlicher Weise bewirkt das Kommando

A>DIR X?Y.C?M

ein Suchen nach allen (eindeutigen) Dateinamen auf der Platte, die dieser mehrdeutigen Bezeichnung genügen.

Die folgenden Dateinamen sind gültige eindeutige Dateibezeichnungen:

X	XYZ	GAMMA
X.Y	XYZ.COM	GAMMA.1

Ein zusätzlicher Komfort besteht darin, daß der Programmierer den Namen des Plattenlaufwerks zusammen mit dem Dateinamen spezifizieren kann. In diesem Fall erhält das Laufwerk als Name einen der Buchstaben von A bis P gefolgt von einem Doppelpunkt („:“). Das spezifizierte Laufwerk ist dann eingeloggt, bevor die Operation durchgeführt wird. So sind nachstehend gültige Dateinamen aufgeführt mit vorangestellten Plattennamen:

A:X.Y	B:XYZ	C:GAMMA
P:XYZ.COM	B:X.A?M	C:*.ASM

Alle Kleinbuchstaben in Datei- und Laufwerksnamen werden in Großschreibung übersetzt, wenn sie durch den CCP bearbeitet werden.

## 2.2 Residente Kommandos

Die Formen, Dateien und Laufwerke zu bezeichnen, können nun dazu benutzt werden, die Struktur der residenten Kommandos zu spezifizieren. Für die anschließenden Beschreibungen sollte sich der Benutzer die folgenden Abkürzungen einprägen:

ufn eindeutige Dateibezeichnung

afn mehrdeutige Dateibezeichnung

Es sei daran erinnert, daß der CCP immer Zeichen in Kleinschreibung intern in Großschreibung übersetzt. So werden Kleinbuchstaben behandelt, als wären sie Großbuchstaben, falls sie in Befehlsnamen und Dateibezeichnungen auftreten.

### 2.2.1 ERA afn

Das ERA (erase)-Kommando entfernt Dateien von der augenblicklich eingeloggten Platte (das ist der Plattenname, der derzeit in der Bereitschaftsmeldung von CP/M dem „>“-Zeichen vorangeht). Die Dateien, die gelöscht werden, sind diejenigen, die der mehrdeutigen Dateibezeichnung afn genügen. Die folgenden Beispiele sollen die Benutzung des ERA-Kommandos deutlich machen:

ERA X.Y Die Datei mit Namen X.Y auf der derzeit aktivierten Platte wird vom Platteninhaltsverzeichnis entfernt und der Speicherplatz steht wieder zur Verfügung.

ERA X.\* Alle Dateien mit dem Primärnamen X werden von der derzeit aktivierten Platte entfernt.

ERA \*.ASM Alle Dateien mit dem Sekundärnamen ASM werden von der derzeit aktivierten Platte entfernt.

ERA X?Y.C?M Alle Dateien auf der aktivierten Platte, die der mehrdeutigen Bezeichnung X?Y.C?M genügen, werden gelöscht.

ERA \*.\* Alle Dateien auf der aktivierten Platte sollen gelöscht werden. In diesem Fall reagiert der CCP mit der Meldung an der Konsole:

ALL FILES (Y/N)?

Es ist als Antwort ein „Y“ verlangt, bevor die Dateien wirklich gelöscht werden.

**ERA B:\*.PRN** Alle Dateien auf Laufwerk B, die der mehrdeutigen Bezeichnung ???????.PRN genügen, werden unabhängig vom gerade eingeloggt Laufwerk gelöscht.

### 2.2.2 DIR afn

Das DIR (Directory)-Kommando listet die Namen aller Dateien an der Konsole auf, die der mehrdeutigen Dateibezeichnung afn genügen. Als Sonderfall listet das Kommando

**DIR**

die Dateinamen der derzeit aktivierten Platte auf. (Das Kommando „DIR“ ist dem Kommando „DIR\*.\*“ gleichwertig. Gültige DIR-Kommandos sind:

**DIR X.Y**

**DIR X?Z.C?M**

**DIR???.Y**

Ähnlich wie bei anderen CCP-Kommandos kann dem Dateinamen afn ein Laufwerksname vorausgehen. Die folgenden DIR-Kommandos bewirken, daß zuerst das angewählte Laufwerk adressiert wird, bevor das Durchsuchen des Plattenverzeichnisses erfolgt.

**DIR B:**

**DIR B:X.Y**

**DIR B:\*.A?M**

Falls keine Datei auf der angewählten Platte die Directory-Anforderung erfüllt, wird die Meldung „NO FILE“ an der Konsole ausgegeben.

### 2.2.3 REN ufn1 = ufn2

Das REN (Rename)-Kommando erlaubt es dem Benutzer die Namen von Dateien auf der Platte zu ändern. Die Datei mit dem Dateinamen ufn2 wird in ufn1 umbenannt. Dabei wird unterstellt, daß die momentan eingeloggte Platte die umzubenennende Datei (ufn2) enthält. Der Benutzer kann auch einen nach links gerichteten Pfeil an Stelle des Gleichheitszeichens eintippen, falls die Konsole dieses graphische Zeichen bereitstellt. Beispiele für das REN-Kommando sind:

REN X.Y = Q.R                    Die Datei Q.R wird umbenannt  
in X.Y

REN XYZ.COM =XYZ.XXX       Die Datei XYZ.XXX wird um-  
benannt in XYZ.COM

Der Operator kann entweder ufn1 oder ufn2 (oder beide) mit einer vorangestellten Adresse versehen. Geht ufn1 ein Laufwerkname voraus, dann wird angenommen, daß ufn2 auf derselben Platte existiert. Ähnlich, ist ufn2 mit einem Laufwerknamen versehen, dann wird unterstellt, daß auch ufn1 auf diesem Laufwerk vorhanden ist. Dasselbe Laufwerk muß in beiden Fällen spezifiziert sein, wenn sowohl ufn1 als auch ufn2 ein Laufwerkname vorausgeht. Die REN-Kommandos unten zeigen dieses Format:

REN A:X.ASM=Y.ASM           Die Datei Y.ASM wird umbe-  
nannt in X.ASM auf Laufwerk  
A.

REN B:ZAP.BAS=ZOT.BAS      Die Datei ZOT.BAS wird um-  
benannt in ZAP.BAS auf Lauf-  
werk B.

REN B:A.ASM=B:A.BAK        Die Datei A.BAK wird umbe-  
nannt in A.ASM auf Laufwerk B

Falls ufn1 bereits vorhanden ist, antwortet das REN-Kommando mit der Fehlermeldung „FILE EXISTS“ und führt die Umbenennung nicht durch. Falls ufn2 auf der spezifizierten Platte nicht existiert, wird die Meldung „NO FILE“ auf der Konsole ausgegeben.

## 2.2.4 SAVE n ufn

Das SAVE-Kommando legt n Seiten (256-Byte-Blöcke) aus der TPA auf der Platte ab und bezeichnet diese Datei mit ufn. In dem CP/M-System beginnt die TPA bei 100H (hexadezimal), was die zweite Seite des Speichers ist. Das SAVE-Kommando muß 2 Speicherseiten spezifizieren, wenn das Benutzerprogramm den Bereich von 100H bis 2FFH belegt. Die Maschinencode-Datei kann danach geladen und ausgeführt werden. Beispiele sind:

SAVE 3 X.COM	Kopiert 100H bis 3FFH nach X.COM
SAVE 40 Q	Kopiert 100H bis 28FFH nach Q (Man erinnere sich, daß 28 die Anzahl der Seiten bis zur Speicherstelle 28FFH ist und $28H = 2 * 16 + 8 = 40$ dezimal)
SAVE 4 X.Y	Kopiert 100H bis 4FFH nach X.Y.

Das SAVE-Kommando kann im ufn-Teil des Kommandos auch ein Laufwerk spezifizieren, wie nachstehend gezeigt wird.

SAVE 10 B:ZOT.COM Kopiert 10 Seiten (100H bis 0AFFH) in die Datei ZOT.COM auf Laufwerk B.

## 2.2.5 TYPE ufn

Das TYPE-Kommando gibt den Inhalt der ASCII-Quelldatei ufn, die sich auf der gerade eingeloggten Platte befindet, an der Konsole aus. Gültige TYPE-Kommandos sind:

TYPE X.Y  
 TYPE X.PLM  
 TYPE XXX

Das TYPE-Kommando berücksichtigt Tabulationsbefehle (Zeichen ctl -I) im Text, wobei die Tabulatorposition an jeder achten Spalte angenommen wird. Der Dateiname ufn kann auch eine Referenz auf einen Laufwerksnamen enthalten.

TYPE B:X.PRN

Die Datei X.PRN auf Platte B wird ausgegeben.

## 2.2.6 USER n

Das USER-Kommando erlaubt die Haltung nach Benutzernummern getrennter Dateien in demselben Directory und hat die Form

USER n

wo n eine ganze Zahl im Bereich von 0 bis 15 ist. Beim Kaltstart wird der Operator automatisch in den Benutzerbereich 0 eingeloggt, der kompatibel ist mit Standard-CP/M1-Verzeichnissen. Der Operator kann ein USER-Kommando zu jeder Zeit absetzen, um in einen anderen logischen Bereich innerhalb des gleichen Directory zu gelangen. Laufwerke, die eingeloggt sind, während eine Benutzernummer adressiert wird, bleiben automatisch aktiv, wenn der Operator auf eine andere Benutzernummer umschaltet; eine Benutzernummer ist ein einfacher Vorspann, der den Zugriff zu bestimmten Verzeichniseinträgen auf dem aktivierten Laufwerk ermöglicht. Die aktive Benutzernummer bleibt erhalten, bis sie durch ein nachfolgendes USER-Kommando geändert wird oder bis zu einem Kaltstart, der wieder zum USER 0 führt.

## 2.3 Zeilenweises Editieren und Ausgabesteuerung

Der CCP erlaubt gewisse Editierfunktionen, während man Befehlszeilen schreibt.

- ctl-C CP/M-Warmstart, falls am Zeilenbeginn eingetippt
- ctl-E Physikalisches Zeilenende: Der Schreibkopf geht an Zeilenanfang, aber die Zeile wird nicht abgeschickt, bevor die Abschlußtaste (carriage return) gedrückt wird.
- ctl-H Rücksetzen um ein Zeichen
- ctl-J Beenden der laufenden Eingabe (line feed)
- ctl-M Beenden der laufenden Eingabe (carriage return)

ctl-R	Neueingabe der laufenden Kommandozeile nach Zeilenvorschub
ctl-U	Entfernen der laufenden Zeile nach der neuen Zeile
ctl-X	Rücksprung zu Zeilenanfang
ctl-Z	Eingabe - Ende an Konsole (benutzt in PIP und ED)
rub/del ( ← )	löscht das letzte Zeichen, das an der Konsole eingegeben wurde

Die Steuerfunktion ctl-P und ctl-S beeinflussen die Konsolenausgabe:

- ctl-P: Übertrage den gesamten nachfolgenden Konsol-Output an das gerade angeschlossene Druckgerät (siehe Abschnitt 2.4.1). Der Output wird an das Druckgerät und die Konsole geschickt, bis das nächste ctl-P eingegeben wird.
- ctl-S: Setzt die Konsolenausgabe vorübergehend aus. Programmausführung und Ausgabe werden fortgesetzt, sobald das nächste Zeichen an der Konsole eingegeben wird (z. B. ein erneutes ctl-S). Diese Einrichtung hält die Ausgabe an sehr schnellen Konsolen, wie Bildschirmen, an, damit Sie einen Ausschnitt des Listings ansehen können, bevor mit der Ausgabe fortgefahren wird.

Die ctl-Taste-Folgen realisiert man durch Drücken der Buchstabentaste bei gedrückter ctl-Taste. CCP-Kommandozeilen sind im allgemeinen 255 Zeichen lang. Sie kommen erst zur Wirkung, wenn die ↵-Taste gedrückt wird.

## 2.4 Transiente Kommandos

Transiente Kommandos werden von der momentan eingeloggtten Platte geladen und kommen in der TPA zur Ausführung. Diejenigen transienten Befehle, deren Ausführung vom CCP gesteuert wird, sind nachstehend beschrieben. Weitere Funktionen können in einfacher Weise durch den Benutzer definiert werden (siehe Abschnitt 2.4.3).

STAT	Gibt den auf der derzeit eingeloggten Platte noch verfügbaren Speicherplatz in Bytes aus, macht statistische Angaben über bestimmte Dateien und zeigt an oder ändert die Gerätezuordnung.
ASM	Lädt den CP/M-Assembler und holt das spezifizierte Quellprogramm zum Zweck der Assemblierung von der Platte.
LOAD	Lädt eine Datei in „HEX“-Maschinencode-Format und erzeugt eine Datei in einer von der Maschine ausführbaren Form, die in die TPA geladen werden kann (dieses geladene Programm wird ein neues Kommando zur Ausführung unter CCP).
DDT	Lädt die CP/M-Testhilfe in die TPA und setzt sie in Gang.
PIP	Lädt den Prozessor zum Informationsaustausch für nachfolgende Transferoperationen an Plattendateien und Peripheriegeräten.
ED	Lädt das CP/M-Texteditorprogramm und bringt es zur Ausführung.
SYSGEN	Legt eine neue CP/M-Systemplatte an.
SUBMIT	Übergibt eine Kommandodatei zur Abarbeitung im Batch-Betrieb.
DUMP	Gibt den Inhalt einer Datei in hexadezimaler Form aus.
CONFIG	Ändert (entweder vorübergehend oder für dauernd) die spezifischen Konfigurationsparameter.
FORMAT	Richtet eine Diskette für ihre Verwendung im Rechner ein.
DISCIT	Richtet die Festplatte für ihre Verwendung ein.
EXCHANGE	Ermöglicht die Benutzung von Nicht-NCR-Anwendungs-Disketten mit Ihrem NCR DECISION MATE V.

**BACKUP** Kopiert den Inhalt einer oder mehrerer Dateien von einer Platte auf eine andere. Die Dateien dürfen beliebig groß sein. Die Ausgabe am Bildschirm oder am Drucker ist ebenfalls möglich.

Transiente Kommandos werden in der gleichen Weise wie residente Kommandos spezifiziert, und zusätzliche Kommandos können in einfacher Weise durch den Benutzer definiert werden. Zur leichten Handhabung kann dem transienten Kommando ein Drive-Name vorangestellt werden, wodurch das Kommando vom spezifizierten Laufwerk in die TPA zur Ausführung geladen wird. So wird durch das Kommando

**B:STAT**

erreicht, daß CP/M vorübergehend Laufwerk B wählt („einloggt“), damit von dieser Einheit das nichtresidente STAT-Kommando beschafft werden kann für die anschließende Ausführung. Anschließend wird das ursprüngliche Laufwerk wieder eingeloggt.

Die grundlegenden transienten Kommandos werden nachstehend im Detail aufgeführt.

#### **2.4.1 STAT**

Das STAT-Kommando verschafft allgemeine statistische Informationen über Datei-Speicherung und Gerätezuordnung. Es wird durch Eingabe einer der folgenden Kommandoformen aufgerufen:

**STAT**

**STAT „Befehlszeile“**

Spezielle Ausprägungen der „Befehlszeile“ erlauben es, sich die derzeitige Gerätezuordnung zu vergegenwärtigen und sie zu ändern. Nachstehend werden die verschiedenen angebbaren Befehlszeilen vorgestellt und auf der rechten Hälfte jeweils erläutert.

**STAT** Gibt der Benutzer eine leere Befehlszeile ein, dann errechnet STAT den noch verfügbaren Speicherplatz auf allen aktivierten Laufwerken und gibt die Meldung

d:R/W,SPACE: nnnK

oder

d:R/O,SPACE: nnnK

für jedes aktivierte Laufwerk d: aus, wobei R/W anzeigt, daß die zum Laufwerk gehörige Platte gelesen und beschrieben werden kann, und R/O zum Ausdruck bringt, daß der Drive nur gelesen werden kann (ein Laufwerk bekommt die Eigenschaft R/O durch ausdrückliches Setzen auf diesen Zustand, so wie es unten gezeigt wird, oder durch unachtsamen Plattenwechsel ohne Durchführen eines Warmstarts). Der auf der Platte in Laufwerk d: verfügbare Speicherplatz wird durch nnn in Kilobytes angegeben.

STAT d: Wird der Name eines Laufwerkes angegeben, dann wird das betreffende Laufwerk vor der Speicherplatzberechnung ausgewählt. So könnte das Kommando „STAT B:“ abgesetzt werden, während man in A eingeloggt ist, was zur Systemmeldung

BYTES REMAINING ON B: nnnK

führt.

STAT afn: Die Befehlszeile kann auch eine Gruppe von Dateien spezifizieren, die durch das STAT-Kommando abgefragt wird. Die Dateien, deren Namen zur Maske afn passen, werden mit ihrem jeweiligen Speicherbedarf unter der Kopfzeile

RECS BYTES EXT ACC d:FILENAME.TYP

rrrr bbbK ee d:Dateiname.Typ

in alphabetischer Reihenfolge aufgelistet, wobei rrrr die Anzahl der 128-Byte Datensätze, die der Datei zugeordnet sind, bbb die Größe des für die Datei reservierten Speicherbereichs in Kilobyte ( $bbb = rrrr * 128 / 1024$ ) bedeuten, ee ist die Anzahl der Einträge zu je 16 KBytes ( $ee = bbb / 16$ ), (ACC = Zugriffsart), d: ist der Name des Laufwerks, das die betreffende Datei enthält (A...P), Dateiname steht für den (bis) acht Zeichen langen Namen der Primärdatei, und Typ ist der (bis) drei Zeichen lange Dateityp.

Nach der Auflistung der einzelnen Dateien wird der gesamte noch verfügbare Speicherplatz angegeben.

**STAT d:afn** Der Name des Laufwerks kann vor dem Dateinamen afn angegeben werden. Das spezifizierte Laufwerk wird dann angewählt und die Kommandoform „STAT afn“ ausgeführt.

**STAT d:R/O** Diese Form setzt bei dem durch d: vorgegebenen Laufwerk den Schreibschutz, der so lange wirksam bleibt, bis der nächste Warm- oder Kaltstart vorgenommen wird. Falls eine Platte im Zustand „schreibgeschützt“ ist, erscheint die Fehlermeldung

BDOS ERR ON d: READ ONLY

beim Versuch, diese schreibgeschützte Platte zu beschreiben. CP/M wartet bis eine Taste gedrückt und wieder losgelassen wird, bevor ein Warmstart erfolgt (wodurch die Platte in den Zustand R/W gelangt).

Das STAT-Kommando ermöglicht die Beeinflussung der Zuordnung von physikalischen zu logischen Gerätenamen (siehe die IOBYTE-Funktion, beschrieben in den Kapiteln 6 und 7). Es gibt vier logische Peripheriegeräte, von denen jedem zu einem bestimmten Zeitpunkt eines von mehreren physikalischen Peripheriegeräten zugeordnet ist. Die Namen der vier logischen Geräte sind:

**CON:** Das System-Konsolgerät (von CCP für den Verkehr mit dem Bediener benutzt)

**RDR:** Der Leser

**PUN:** Der Stanzer

**LST:** Systemausgabe-Einheit

Die an einem bestimmten Rechnersystem wirklich angeschlossenen Geräte werden durch Unterprogramme aus dem BIOS-Teil von CP/M gesteuert. So könnte z. B. das logische LST:-Gerät im konkreten Fall ein über eine Parallelschnittstelle angeschlossener Drucker

oder ein über eine Asynchronschnittstelle angeschlossener Drucker-  
teil eines Fernschreibers sein. Um etwas Flexibilität in der Bezeich-  
nung und Zuordnung der Geräte zu erlauben, sind eine Reihe von  
physikalischen Geräten nachstehend definiert:

TTY: Asynchrones Drucker-Interface (RS-232) mit X-ON/  
X-OFF Protokoll

CRT: DECISION MATE V-Bildschirm und -Eingabetastatur

BAT: Nicht implementiert

UC1: Nicht implementiert

PTR: Nicht implementiert

UR1: Nicht implementiert

UR2: Nicht implementiert

PTP: Nicht implementiert

UP1: Nicht implementiert

UP2: Nicht implementiert

LPT: Zeilendrucker (Centronics Parallel-Schnittstelle)

UL1: Nicht implementiert

Es sei betont, daß die Namen der physikalischen Geräte mit den  
Geräten, die diese Namen unterstellen, etwas zu tun haben können,  
aber nichts gemeinsam haben müssen. Der Befehl

**STAT VAL:**

liefert eine Zusammenstellung der verfügbaren Status-Kommandos  
und gibt diese wie folgt aus:

Temp R/O Disk d:\$R/O  
Set Indicator: filename.typ\$R/O \$R/W \$SYS \$DIR  
Disk Status: DSK: d:DSK  
User Status: USR:  
lobyte Assign:

Damit wird eine Zusammenstellung der derzeit verfügbaren STAT-Kommandos gegeben und die erlaubte Zuordnung der logischen zu den physikalischen Geräten gezeigt:

CON: = TTY: CRT: BAT: UC1:  
 RDR: = TTY: PTR: UR1: UR2:  
 PUN: = TTY: PTP: UP1: UP2:  
 LST: = TTY: CRT: LPT: UL1:

Auf der linken Seite des „=-Zeichens stehen die logischen Gerätenamen, auf der rechten die vier ihnen jeweils zuzuordnenden physikalischen Gerätenamen. Die aktuelle Zuordnung von logischen zu physikalischen Gerätenamen erfolgt durch Eingabe des Kommandos

STAT DEV:

wodurch eine Liste ausgegeben wird, in der jedes logische Gerät auf der linken Seite und das ihm aktuell entsprechende physikalische Gerät auf der rechten Seite aufgeführt ist. Die Originalzuordnung ist:

CON: IS CRT:  
 RDR: IS TTY:  
 PUN: IS TTY:  
 LST: IS LPT:

Die aktuelle Zuordnung der logischen zu den physikalischen Gerätenamen kann geändert werden durch Eingabe eines STAT-Befehls der Form

STAT l<sub>d1</sub> = p<sub>d1</sub>, l<sub>d2</sub> = p<sub>d2</sub>,..., l<sub>dn</sub> = p<sub>dn</sub>

wo l<sub>d1</sub> bis l<sub>dn</sub> logische Gerätenamen und p<sub>d1</sub> bis p<sub>dn</sub> dazu passende physikalische Gerätenamen sind (d. h. l<sub>di</sub> und p<sub>di</sub> erscheinen beim oben gezeigten „VAL:“-Kommando in derselben Zeile). Zulässige STAT-Befehle, welche die aktuelle Zuordnung der logischen zu den physikalischen Gerätenamen verändern, sind:

STAT CON: = CRT:

STAT PUN: = TTY:, LST: = LPT:;, RDR: = TTY:

## Die Befehlsform

STAT d:Dateiname.Typ \$\$

in der „d:“ ein optionaler Laufwerksname und „Dateiname.Typ“ ein eindeutiger oder mehrdeutiger Dateiname sind, liefert die Ausgabe im Format

<b>Size</b>	<b>Recs</b>	<b>Bytes</b>	<b>Ext Acc</b>
48	48	6K	1 R/O A:ED.COM
55	55	12K	1 R/O A:PIP.COM
65536	128	16K	2 R/W A:X.DAT

wobei der \$\$ Parameter bewirkt, daß das „Size“-Feld ausgegeben wird. (Ohne \$\$ wird das Size-Feld übergangen, die anderen Felder werden angezeigt.) Das Size-Feld listet die virtuelle Dateigröße in Aufzeichnungseinheiten auf, während das „Recs“-Feld die Anzahl der virtuellen Einträge in jedem logischen Verzeichniseintrag (extent) angibt. Für Dateien, die sequentiell angelegt wurden, ist „Size“ mit der „Recs“-Angabe identisch. Das „Bytes“-Feld gibt die Größe des auf der Diskette durch Aufzeichnungsblöcke reservierten Bereichs in Bytes für die betreffende Datei an. Die kleinste Zuordnungseinheit wird zur Konfigurationszeit festgelegt; die Größe einer Datei in Bytes läßt sich für eine sequentielle Datei aus der Anzahl der records und dem verbleibenden ungenutzten Platz im letzten zugeordneten Block ermitteln. Dateien mit Direktzugriff erhalten Datenbereiche nur beim Schreiben, damit gibt nur das Bytes-Feld eine genaue Angabe über die Zuordnung. Das Size-Feld gibt in diesem Fall die Position des Ende-der-Datei-Eintrags, und das Recs-Feld zählt die logischen Einträge von jedem Extent. (Jeder dieser Extents kann nicht zugeordnete Lücken enthalten, die bei den Records mitgezählt werden.) Das „Ext“-Feld zählt die Anzahl der physikalischen Extents, die der Datei zugeordnet sind. Die Anzahl der Extents gibt die Anzahl der logischen Verzeichniseinträge einer Datei an.

Das Acc-Feld gibt den R/O- oder R/W-Indikator an, der durch Anwendung nachstehend gezeigter Kommandos verändert werden kann.

Die vier Kommando-Formen

STAT d: Dateiname.Typ \$R/O

STAT d: Dateiname.Typ \$R/W

STAT d: Dateiname.Typ \$SYS

STAT d: Dateiname.Typ \$DIR

setzen verschiedene permanente Dateiindikatoren oder setzen sie zurück. Der R/O-Indikator versetzt eine Datei (oder eine Gruppe von Dateien) in den Zustand „schreibgeschützt“, in dem sie so lange verbleibt, bis durch ein nachfolgendes STAT-Kommando eine Änderung erzwungen wird. Der R/O Status wird mit der Datei in das Directory eingetragen, so daß sie über Kaltstart-Operationen hinweg schreibgeschützt bleibt. Der R/W-Indikator versetzt die Datei in einen permanenten Lese/Schreib-Zustand. Der SYS-Indikator versieht die Datei mit dem Kennzeichen „system“, während der DIR-Indikator dieses wieder entfernt. Der „Dateiname.Typ“ kann mehrdeutig oder eindeutig sein, aber Dateien, deren Attribute geändert werden, werden an der Konsole aufgelistet, sobald an ihnen die Änderung vorgenommen wird. Der Name des Laufwerks, angegeben durch d:, ist optional.

Trägt eine Datei das Kennzeichen R/O, so bewirken nachträgliche Versuche, die Datei zu löschen oder zu beschreiben, eine BDOS-Meldung am Terminal

BDOS Err on d: File R/O

BDOS erwartet dann eine Konsoleingabe, um anschließend einen Warmstart anzustoßen (ein ↵ ist hinreichend).

Die Befehlsform

STAT d:DSK:

listet die Charakteristika des Laufwerks der Platte „d:“ auf. d liegt im Bereich A, B, ..., P. Die Ausgabe erfolgt im Format:

d: Drive Charakteristika

2464: 128 Byte Gesamte Satzkapazität

308: Kilobyte Kapazität/Laufwerk

- 128: 32 Byte Verzeichnis Einträge
- 128: Geprüfte Directory (Verzeichnis)-Einträge
- 256: Sätze/Extent (Erweiterung)
- 16: Sätze/Block
- 32: Sektoren/Spur
- 3: Reservierte Spuren

„d.“ ist dabei das angewählte Laufwerk, gefolgt von der gesamten Satzkapazität, gefolgt von der Gesamtkapazität, ausgedrückt in Kilobytes. Als nächstes ist die Directorygröße aufgelistet, gefolgt von den „geprüften Einträgen“. Die Anzahl der geprüften Einträge ist gewöhnlich identisch mit der Directorygröße der auswechselbaren Datenträger, weil dieser Mechanismus dazu benutzt wird, ausgewechselte Datenträger festzustellen, wenn der Austausch ohne Warmstart erfolgt war. Die Anzahl der Sätze pro Extent bestimmt die Adressierungskapazität jedes Directory-Eintrags (256mal 128 Bytes, oder 32K im vorigen Beispiel). Die Anzahl der Sätze pro Block zeigt die kleinste Zuordnungsmenge (im Beispiel 16 Sätze/Block mal 128 Bytes pro Satz oder 2KBytes pro Block). In der Auflistung folgt dann die Anzahl der logischen Sektoren pro Spur und die Anzahl der für das Betriebssystem reservierten Spuren.

Die Befehlsform

STAT DSK:

liefert eine Tabelle der Laufwerk-Charakteristika für alle aktivierten Laufwerke.

Die letzte Form des STAT-Kommandos lautet:

STATUSR:

Dieses Kommando erzeugt eine Liste der Nummern der Benutzer, die Dateien auf der gerade adressierten Platte haben. Die Anzeige erfolgt in der Form:

Aktiver Benutzer: 0

Aktive Dateien: 0 1 3

Die erste Zeile listet die Nummern der gerade adressierten Benutzer auf, wie beim letzten CCP USER-Kommando festgelegt, gefolgt von einer Liste mit Benutzernummern, die durch Abfrage des laufenden Directory gewonnen wurden. In diesem Fall ist die Nummer des aktiven Benutzers 0 (beim Kaltstart automatisch zugeordnet), mit den Nummern von drei Benutzern, die aktive Dateien auf der momentan adressierten Platte haben. Der Operator kann nacheinander die Verzeichnisse der anderen Benutzernummern überprüfen durch Einloggen mit USER 1 - oder USER 3-Kommandos, gefolgt jeweils von einem DIR-Kommando auf CCP-Ebene.

## 2.4.2 ASM ufn

Laden und Ausführen des CP/M 8080-Assemblers werden mit dem ASM-Kommando bewerkstelligt. ufn spezifiziert dabei eine Quelldatei, die in Assemblersprache geschriebene Anweisungen enthält, wobei der Dateityp ASM unterstellt und nicht weiter spezifiziert wird. Die folgenden ASM-Kommandos sind gültig

ASM X

ASM GAMMA

Der Zwei-Pass-Assembler kommt automatisch zur Ausführung. Assemblerfehler, die während des zweiten Laufes zutage treten, werden an der Konsole ausgeschrieben.

Der Assembler erzeugt eine Datei

X.PRN

wo X der im ASM-Kommando spezifizierte Primärname ist. Die PRN-Datei enthält ein Listing des Quellprogramms (mit eingebetteten Tab.-Zeichen, falls im Quellprogramm vorhanden), mit dem für jede Anweisung generierten Maschinencode und gegebenenfalls Fehlermeldungen. Die PRN-Datei kann durch Verwendung des TYPE-Kommandos an der Konsole aufgelistet oder bei Anwendung des PIP (siehe Abschnitt 2.4.4) an ein Peripheriegerät geschickt

werden. Der Benutzer sollte beachten, daß die PRN-Datei das Originalquellprogramm enthält, ergänzt um diverse Assemblerinformationen in den ersten 16 Spalten (Programmadressen und Hexadezimal-Maschinencode, z. B.). Die PRN-Datei dient als Sicherung für die Originalquelldatei. Falls die Quelldatei unbeabsichtigt gelöscht oder zerstört wird, kann die PRN-Datei editiert (siehe Kapitel 3) werden, indem man die ersten 16 Spalten jeder Zeile entfernt. Dies kann durch Absetzen eines einzigen Editor-Makrobefehls erreicht werden. Die entstandene Datei ist identisch mit der Originalquelldatei und kann (mit REN) von PRN in ASM umbenannt werden für nachfolgendes Editieren und Assemblieren.

Die Datei

X.HEX

wird ebenfalls vom Assembler generiert. Diese Datei enthält 8080 Maschinensprache in Intel „HEX“-Format, geeignet für nachfolgendes Laden und Ausführen (siehe Abschnitt 2.4.3). Für detailliertere Behandlung der CP/M-Assemblersprache sei auf Kapitel 4 verwiesen.

Die Quelldatei, die assembliert werden soll, kann von einer anderen als der eingeloggten Platte durch Voranstellen des Laufwerknamens vor den Dateinamen geholt werden.

Das Kommando

ASM B:ALPHA

lädt den Assembler vom gerade eingeloggten Laufwerk und assembliert das auf Platte B sich befindliche Quellprogramm ALPHA.ASM. Die HEX- und PRN-Dateien werden in diesem Fall ebenfalls auf Platte B abgelegt.

### **2.4.3 LOAD ufn**

Der LOAD-Befehl liest die Datei ufn, von der unterstellt wird, daß sie HEX-Format Maschinencode enthält, und erstellt eine Speicher-Format-Datei, die anschließend zur Ausführung kommen kann. Eine Datei der Form

X.HEX

angenommen, und lediglich der Dateiname X muß im Kommando

spezifiziert werden. Das LOAD-Kommando legt eine Datei an mit dem Namen

### X.COM

was zum Ausdruck bringt, daß sie von der Maschine ausführbaren Code enthält. Die Datei wird in den Speicher geladen und ausgeführt, wenn der Benutzer den Dateinamen X unmittelbar hinter dem Prompt-Zeichen „>“ des CCP eingibt.

Der CCP liest im allgemeinen den dem Promptzeichen folgenden Dateinamen X und sucht nach einem residenten Funktionsnamen. Falls kein solcher Funktionsname gefunden wird, durchsucht der CCP das Verzeichnis der Systemplatte nach einer Datei mit dem Namen

### X.COM

Kann sie gefunden werden, wird der Maschinencode in die TPA geladen und das Programm wird ausgeführt. Der Benutzer braucht mit LOAD eine HEX-Datei also nur einmal zu laden; sie kann nachher durch Eingabe des Primärnamens beliebig oft zur Ausführung gebracht werden. Auf diese Weise kann der Benutzer neue Befehle in den CCP einbringen. (Initialisierte Platten enthalten die transienten Kommandos als COM-Dateien, die der Benutzer je nach Wunsch löschen kann.) Die ganze Operation kann auch mit einem anderen als dem eingeloggtten Laufwerk erfolgen, wenn dem Dateinamen der Name des Laufwerks vorangestellt wird. So bringt der Befehl

### LOAD B:BETA

das LOAD-Programm in die TPA von der gerade eingeloggtten Platte her und arbeitet nach Beginn der Ausführung mit Laufwerk B.

Man beachte, daß die Datei BETA.HEX Sätze enthalten muß, die in gültigem Intel-Hexadezimal-Maschinencode dargestellt sind (wie z. B. vom ASM-Programm erzeugt), und die an Speicherstelle 100 H der TPA beginnen. Die Adressen in den Hex-Sätzen müssen in aufsteigender Reihenfolge angeordnet sein. Lücken in Speicherbereichen werden durch das LOAD-Kommando beim Lesen der Sätze mit Nullen aufgefüllt. Das LOAD-Kommando sollte man nur verwenden, um CP/M-Standard „COM“-Dateien anzulegen, die in der TPA arbeiten. Programme, die andere Speicherbereiche belegen als die TPA, werden unter DDT geladen.

## 2.4.4 PIP

PIP ist das Programm zum Informationsaustausch zwischen Peripherieeinheiten. Es implementiert die zum Laden, Stanzen, Kopieren und Kombinieren von Plattendateien notwendigen Routinen zur Umsetzung auf andere Datenträger. Das PIP-Programm wird aufgerufen durch Eingabe des Kommandos in einer der beiden Formen

(1) PIP

(2) PIP 'Befehlszeile'

In beiden Fällen wird PIP in die TPA geladen und gelangt zur Ausführung. In Form (1) liest PIP Kommandozeilen direkt von der Konsole, wobei mit dem Bereitschaftszeichen „\*“ weitere Eingaben angefordert werden, bis ein leeres Kommando registriert wird (d. h. bis ein einfaches CR (↵) durch die Bedienungsperson abgesetzt wird). Jede Kommandozeile bewirkt irgendeine Datenträgerumsetzung gemäß den unten angegebenen Regeln. Form (2) des PIP-Kommandos ist äquivalent der Form (1), außer, daß hier die mit dem PIP-Kommando angegebene Befehlszeile automatisch ausgeführt wird und PIP ohne weitere Absetzung von Eingabeanforderungen an die Konsole seine Arbeit beendet. Die Form jeder Kommandozeile ist

Ziel = Quelle # 1, Quelle # 2, . . . Quelle # n

wobei „Ziel“ die Datei oder das Peripheriegerät bedeutet, die bzw. das die Daten empfangen soll, und „Quelle # 1“, . . . „Quelle # n“ stehen für eine Reihe von einem oder mehreren Dateien oder Geräten, die von links nach rechts in das Ziel kopiert werden.

Sind in der Befehlszeile mehrere Dateien (d. h.  $n > 1$ ) angegeben, so wird angenommen, daß die einzelnen Dateien ASCII-Zeichen enthalten und jeweils am Ende durch ein CP/M-EOF-Zeichen (ctl-Z) abgeschlossen sind (siehe O-Parameter zur Außerkraftsetzung dieser Annahme). ASCII-Kleinbuchstaben werden in Großschreibung umgesetzt, um mit den CP/M-Konventionen für die Namensgebung von Dateien und Geräten in Einklang zu sein. – Schließlich darf die Gesamtlänge einer Befehlszeile 255 Zeichen nicht überschreiten. Die Ziel- und Quellelemente sind eindeutige Bezeichnungen von CP/M-Quelldateien mit oder ohne vorangehende Laufwerkskennzeichnung. D. h. jede Datei kann mit einem voranstehenden Laufwerksnamen (A: bis P:) angegeben werden, der das Laufwerk definiert, von dem die Datei geholt wird oder auf das sie gespeichert wird. Ist der Lauf-

werkname nicht angegeben, wird die gerade eingeloggte Platte angenommen. Der Name der Zieldatei kann ebenfalls an Stelle einer oder mehrerer Quelldateien erscheinen. In diesem Fall wird die Quelldatei erst geändert, wenn die gesamte Verkettung abgeschlossen ist. Existiert die Zieldatei bereits, wird sie durch das PIP-Kommando überschrieben, falls die Befehlszeile richtig gebildet ist. Liegt eine Fehler-situation vor, so wird sie nicht gelöscht.

Die nachstehenden Befehlszeilen (mit Erklärungen auf der rechten Seite) sind zulässige Eingaben für PIP:

$X = Y$	Kopiert Datei Y nach Datei X, wobei X und Y eindeutige Dateinamen sind; die Datei Y wird nicht verändert.
$X = Y,Z$	Eine Verkettung der Dateien Y und Z wird nach Datei X übertragen, wobei Y und Z nicht verändert werden.
$X.ASM = Y.ASM,Z.ASM,FIN.ASM$	Eine Datei X.ASM wird angelegt und bekommt als Inhalt die Verkettung der Dateien Y,Z und FIN, jeweils mit dem Dateityp ASM.
$NEW.ZOT = B:OLD.ZAP$	Überträgt eine Kopie der Datei OLD.ZAP von Laufwerk B auf die derzeit eingeloggte Platte; Name der neuangelegten Datei NEW.ZOT
$B:A.U = B:B.V,A:C.W,D.X$	Verkettet Datei B.V auf Laufwerk B mit C.W auf Laufwerk A und D.X auf der gerade eingeloggten Platte; die Datei A.U wird auf der Platte in Laufwerk B angelegt.

In PIP sind der bequemerer Handhabung wegen Abkürzungen in den Kommandos zur Dateiübertragung zwischen verschiedenen Laufwerken zugelassen. Dabei sind folgende Formen möglich:

PIP d: = afn

PIP d1: = d2:afn

PIP ufn = d2:

PIP d1:ufn = d2:

Die erste Form kopiert alle Dateien, die der mehrdeutigen Bezeichnung afn genügen, von der gerade eingeloggten Platte in die gleichnamige Datei auf Laufwerk d (d = A . . . P). Die zweite Form ist gleichwertig der ersten, wobei die Quelldateien für den Kopiervorgang auf Laufwerk d2 (d2 = A . . . P) liegen. Die dritte Form ist gleichwertig dem Kommando

PIP d1:ufn = d2:ufn

das die Datei mit dem Namen ufn von Laufwerk d2 in die Datei ufn auf der gerade eingeloggten Platte kopiert.

Die vierte Form ist gleichwertig der dritten, wobei noch die Zielplatte explizit durch d1 angegeben wird.

Quell- und Ziellaufwerke müssen in allen genannten Fällen verschieden sein. Falls ein mehrdeutiger Name afn spezifiziert wird, listet PIP den Namen jeder einzelnen Datei ufn, der auf afn paßt, während des Kopierens auf. Falls eine Datei mit dem Namen der Zieldatei bereits existiert, wird sie nach erfolgreicher Durchführung des Kopiervorgangs gelöscht und durch die kopierte Datei ersetzt. Die nachstehenden PIP-Kommandos sind Beispiele zulässiger Kopieroperationen von Platte zu Platte:

B:=\*.COM            Kopiert alle Dateien, die den Dateityp (Namenszusatz) „COM“ haben, von dem gerade adressierten Laufwerk auf die Platte in Laufwerk B.

A:=B:ZAP.\*            Kopiert alle Dateien mit dem Primärnamen „ZAP“ auf Laufwerk B auf die Platte in LaufwerkA.

ZAP.ASM=B:            Äquivalent zu ZAP.ASM = B:ZAP.ASM

B:ZOT.COM=A:            Äquivalent zu B:ZOT.COM = A:ZOT.COM

B:=GAMMA.BAS    Äquivalent zu B:GAMMA.BAS =  
GAMMA.BAS

B:=A:GAMMA.BAS    Äquivalent zu B:GAMMA.BAS =  
A:GAMMA.BAS

PIP erlaubt den Bezug auf physikalische und logische Geräte, die in das CP/M-System eingebunden sind. Die Gerätenamen sind dieselben, wie beim STAT-Kommando angegeben. Dazu kommt noch eine Reihe von speziell bezeichneten Geräten. Die beim STAT-Befehl angegebenen logischen Geräte sind:

CON:(Konsole), RDR:(Leser), PUN:(Stanzer) und LST:(Drucker)

Die physikalischen Geräte sind demgegenüber:

TTY: (Konsole, Leser, Stanzer oder Drucker)

CRT: (Konsole oder Drucker), UC1: (Konsole)

PTR: (Leser), UR1: (Leser), UR2: (Leser)

PTP: (Stanzer), UP1: (Stanzer), UP2: (Stanzer)

LPT: (Drucker), UL1: (Drucker)

Die RDR, LST, PUN und CON-Einrichtungen sind innerhalb des BIOS-Teils von CP/M definiert. (Die gültige Abbildung der logischen auf die physikalischen Geräte ist durch das IOBYTE definiert; siehe Kapitel 7 bezüglich einer Erläuterung dieser Funktion.)

Das Zielgerät muß in der Lage sein, Daten zu empfangen und die Quellgeräte müssen in der Lage sein, Daten zu erzeugen (d. h. das LST: Gerät kann nicht gelesen werden).

Die in PIP-Befehlen zusätzlich verwendbaren Gerätenamen sind:

NUL:    Sendet 40 Nullbytes (ASCII 0) an das Zielgerät (dies kann am Ende eines Ausgabeblochstreifens abgesetzt werden).

EOF:    Sendet ein CP/M-Dateiendezeichen (ASCII ctl-Z) an die Zieldatei (wird durch PIP am Ende aller Datenübertragungen gesendet).

- INP: Spezielle PIP-Eingabeeinheit, die in das PIP-Programm eingefügt werden kann. PIP erhält die Eingabedaten Zeichen für Zeichen durch Aufruf der Adresse 103H, wobei die Daten in Adresse 109H (Paritätsbit muß 0 sein) abgelegt werden.
- OUT: Spezielle PIP-Ausgabe-Zieleinheit, die in das PIP-Programm eingefügt werden kann: PIP ruft Adresse 106H auf, wobei jedes auszugebende Zeichen aus Register C zu entnehmen ist. Der Benutzer sollte beachten, daß die Adressen 109H bis 1FFH des Speicherbereichs von PIP nicht benutzt werden und unter Verwendung von DDT durch besondere Treiberprogramme belegt werden können (siehe Kapitel 5).
- PRN: Wirkt wie LST; außerdem Tabulationsspalten von achten Zeichen Breite, Zeilennummerierung und Seitenvorschub nach jeweils 60 Zeilen mit einem Vorschub am Beginn des Ausgabevorgangs (entspricht der Verwendung der PIP-Optionen [t8np]).

Datei- und Gerätenamen können in die PIP-Kommandos eingestreut werden. In jedem Fall wird das spezifische Gerät (bzw. die spezifische Datei) bis zum Textendezeichen (ctl-Z für ASCII-Dateien und End of Data für Nicht-ASCII-Plattendateien) gelesen. Die Daten von jedem Gerät bzw. von jeder Datei werden von links nach rechts verkettet, bis die letzte Datenquelle gelesen ist. Die Zieldatei (das Zielgerät) wird mit den Daten der Quelldateien beschrieben und ein EOF-Zeichen (ctl-Z) bei ASCII-Dateien am Dateiende angefügt. Ist die Zieldatei eine Plattendatei, so wird zuerst eine Zwischendatei (Namenszusatz \$\$\$) angelegt, die in den aktuellen Dateinamen nur dann umbenannt wird, falls der Kopiervorgang erfolgreich abgeschlossen worden ist. Dateien mit dem Namenszusatz „COM“ werden als „Nicht-ASCII“ angenommen.

PIP führt eine besondere Funktion aus, falls die Zieldatei eine Plattendatei vom Typ „HEX“ (eine Intel hex-formatierte Maschinencoddatei) ist, und die Quelle ein externes Peripheriegerät, wie etwa ein Lochstreifenleser. In diesem Fall führt PIP eine Prüfung durch, um sicherzustellen, daß die Quelldatei eine richtig erstellte Hex-Datei ist mit gültigen Hexadezimalwerten und Kontrollsummen-einträgen. Wird ein ungültiger Eingabesatz entdeckt, so gibt PIP eine Fehlermeldung aus und wartet auf Korrekturmaßnahmen. Normalerweise genügt es, den Leser zu öffnen und ein Stück des Streifens von

neuem durchlaufen zu lassen (man setze dazu den Streifen etwa einen halben Meter zurück). Wenn die Vorbereitungen für das Neu-lesen des Streifens getroffen sind, wird  $\leftarrow$  an der Tastatur eingegeben und PIP startet dann einen neuen Leseversuch.

Falls diese Maßnahmen nicht in geeigneter Weise durchführbar sind, setzt der Benutzer den Lesevorgang fort (durch Eingabe von  $\leftarrow$  nach der Fehlermeldung) und fügt den Satz manuell mit Hilfe des ED-Programms ein, nachdem die Datei erst einmal konstruiert worden ist.

Als Erleichterung gestattet es PIP, das EOF-Zeichen manuell von der Tastatur her einzugeben, falls die Quelldatei ein Lesegerät ist. In diesem Fall liest das PIP-Programm die vom Gerät gelieferten Werte und überwacht die Eingabetastatur. Sobald  $\leftarrow$  an der Tastatur eingegeben wird, findet eine normale Beendigung des Lesevorgangs statt.

Beispiele gültiger PIP-Kommandos sind:

PIP LST:=X.PRN

Kopiert X.PRN in das LST-Gerät und terminiert das PIP-Programm.

PIP

Startet PIP für eine Folge von Kommandos (PIP meldet seine Bereitschaft durch Ausgabe von „\*“ am Bildschirm).

\*CON:=X.ASM,Y.ASM,Z.ASM

Verkettet drei ASM-Dateien und gibt das Ergebnis am Bildschirm aus.

\*X.HEX=CON:;Y.HEX,PTR:

Legt eine Hexdatei X.HEX an und beschreibt sie von der Tastatur aus. Der Abschluß der Tastatureingabe erfolgt durch Eintippen eines  $\leftarrow$ -Zeichens. Dann folgt der Inhalt der Datei Y.HEX und ein Lochstreifen, der mit einem  $\leftarrow$ -Zeichen abgeschlossen wird.

Die Arbeit von PIP wird durch einen einfachen CR beendet.

PIP PUN:=NUL:;X.ASM,EOF:;NUL: Bringt am Lochstreifen einen Vorspann von 40 Nullen an, dann wird eine Kopie von X.ASM in den Streifen gestanzt, gefolgt vom EOF-Zeichen (ctl-Z) und einem Nachspann von weiteren 40 Nullen.

Der Benutzer kann auch einen oder mehrere PIP-Parameter spezifizieren, die in eckige Klammern eingeschlossen werden und voneinander durch kein, ein oder mehrere Leerzeichen getrennt sind. Es sei an dieser Stelle an die Übersetzungs- und Konversionstabellen im Anhang verwiesen.

Jeder Parameter wirkt sich auf den Kopiervorgang aus. Die eingeschlossene Parameterliste muß unmittelbar auf die Datei, auf die eingewirkt werden soll, oder auf das Gerät, das beeinflußt werden soll, folgen. Jedem Parameter kann ein optionaler ganzzahliger Dezimalwert folgen (die S- und Q-Parameter bilden Ausnahmen). Nachstehend sind gültige PIP-Parameter aufgeführt.

- B Blockweise Übertragung: Die Daten werden von PIP gepuffert bis ein ASCII-x-off-Zeichen (ctl-S) von der Quelldatei (Quellgerät) her empfangen wird. Dies ermöglicht den Datentransfer zu einer Plattendatei von einem kontinuierlichen Lesegerät, wie einem Kassettenleser. Nach dem Empfang des x-off leert PIP die Plattenpuffer und fordert neue Eingaben an. Der Umfang an Daten, der gepuffert werden kann, hängt von der Speichergröße des Zentralsystems ab (PIP gibt eine Fehlermeldung aus, wenn die Puffer überlaufen).
- Dn Blendet die Zeichen aus der Übertragung von der Quelle zur Zieldatei bzw. zum Zielgerät aus, die über die Spalte n hinausreichen würden. Dieser Parameter wird hauptsächlich dazu benutzt, Zeilen, die zu einem Drucker oder Konsolgerät gesendet werden, in der Länge zu beschränken.

- E Wiederholt alle von einer Quelldatei gelesenen Zeichen auf dem Konsolenausgabegerät.
- F Entfernt alle Seitenvorschübe aus dem übertragenen Text. Der P-Parameter kann dann dazu benutzt werden, neue Seitenvorschübe einzufügen.
- Gn Kopiert eine Datei aus Benutzerbereich n ( $0 \leq n \leq 15$ ).
- H Es wird nachgeprüft, ob die gelesenen Daten dem Hexadezimalformat nach der Intel-Definition entsprechen. Nicht wesentliche Zeichen zwischen den Hex-Sätzen werden beim Kopiervorgang entfernt. Über die Konsole werden beim Auftreten von Fehlern Korrekturmaßnahmen gefordert.
- I (Ignore „:00“-Einträge): Läßt beim Übertragen von Intel-Hex-Dateien alle Aufzeichnungen unberücksichtigt, die mit „:00“ beginnen. Der I-Parameter aktiviert automatisch den H-Parameter.
- L Wandelt alle Großbuchstaben in Kleinschreibung um.
- N Setzt allen zur Zieldatei übertragenen Textzeilen Zeilennummern voran, wobei mit 1 begonnen und jeweils um 1 erhöht wird. Das Kommando unterdrückt führende Nullen und schließt die Zeilennummern mit einem Doppelpunkt ab.  
Falls N2 spezifiziert wird, werden alle führenden Nullen ausgedruckt und die Zeilennummern mit einem Tabulationsbefehl abgeschlossen. Der Tabulationsabstand wird vergrößert, falls T entsprechend gesetzt ist.
- O Objektdatei (Nicht-ASCII)-Übertragung: Ein normales CP/M-EOF wird nicht berücksichtigt.
- Pn Fügt nach jeweils n Zeilen einen Seitenvorschub ein (am Anfang der Zieldatei steht ein Seitenvorschub). Falls n=1 ist oder ganz weggelassen wird, erfolgt nach jeweils 60 Zeilen ein Seitenvorschub. Wird der F-Parameter benutzt, so erfolgt die Unterdrückung der Seitenvorschübe, bevor ihre Neudefinition eingefügt wird.

- Qs↑z Beendet den Kopiervorgang, nachdem die Zeichenfolge (String) s (abgeschlossen durch ctl-Z) übertragen ist.
- R Ermöglicht die Übertragung von Systemdateien.
- Ss↑z Beginnt den Kopiervorgang von der Quelldatei mit der Zeichenfolge s (abgeschlossen durch ein ctl-Z). Die S- und Q-Parameter können dazu benutzt werden, eine Datei teilweise zu kopieren. Die einleitende, sowie die abschließende Zeichenfolge werden ebenfalls mitübertragen.
- Wählt der Benutzer Form (2) des PIP-Befehls, so übersetzt der CCP eine den S- und Q-Parametern folgende Zeichenfolge s in Großschreibung. Form (1) des PIP-Aufrufs nimmt die automatische Übersetzung in Großschreibung nicht vor.
- (1) PIP
- (2) PIP 'Befehlszeile'
- Tn Setzt den Tabulationsbefehl (ctl-I-Zeichen) an jeder n-ten Spalte während der Übertragung von der Quell- zur Zieldatei.
- U Wandelt alle Kleinbuchstaben in Großschreibung um.
- V Überprüft nach Abschluß der Schreiboperation, ob die Daten richtig übertragen worden sind (Zieldatei muß Plattendatei sein).
- W Überschreibt schreibgeschützte Dateien ohne Rückfrage an der Konsole.
- Z Rücksetzen des Parity-Bits (Bit 7) in allen übertragenen ASCII-Zeichen.

Es folgen Beispiele gültiger PIP-Kommandos, in denen für die Dateiübertragung Parameter spezifiziert sind:

PIP X.ASM=B:[v]            Kopiert die Datei X.ASM von Laufwerk B auf das gerade eingeloggte Laufwerk und überprüft, ob die Übertragung korrekt erfolgt ist.

PIP LPT:=X.ASM[nt8u]      Kopiert X.ASM an die LPT:-Einheit; jede Zeile wird mit einer Nummer versehen, jede achte Spalte ist Tabulationsspalte und Kleinbuchstaben werden in Großschreibung umgewandelt.

PIP PUN:=X.HEX[i],Y.ZOT[h]      Zuerst wird X.HEX zur Stanzeinheit transferiert, der „:00“-Satz wird ignoriert; dann wird der Datentransfer durch Lesen der Datei Y.ZOT fortgesetzt, die HEX-Aufzeichnungen enthält, einschließlich aller „:00“-Sätze.

PIP X.LIB=Y.ASM[sSUBRI:†z qJMP L3†z]      Kopiert aus der Datei Y.ASM in die Datei X.LIB. Der Kopiervorgang beginnt mit der Zeichenfolge „SUBRI:“ und wird mit der Zeichenfolge „JMP L3“ abgeschlossen.

PIP PRN:=X.ASM[p50]      Überträgt die Datei X.ASM an die LST-Einheit, mit Zeilennummerierung, Tabulationsbefehlen an jeder achten Spalte und Seitenvorschub nach jeweils 50 Zeilen. Die angenommene Parameterliste der PRN-Datei ist nt8p60; p50 überschreibt den Standardwert.

Normalerweise überschreibt PIP eine permanent schreibgeschützte Datei nicht. Beim Versuch, dies zu tun, wird die Meldung

DESTINATION FILE IS R/O, DELETE (Y/N)

ausgegeben. Wenn Sie mit "Y" antworten, wird die Datei überschrieben. Andernfalls wird die Meldung

\*\* NOT DELETED \*\*

ausgeschrieben, der Datentransfer wird übersprungen, und PIP setzt die Arbeit mit der nächstfolgenden Operation fort.

Um diesen Dialog im Fall des Überschreibens von schreibgeschützten Dateien zu vermeiden, kann die Befehlszeile den W-Parameter enthalten.

PIP A:=B:\*COM[W]

Dieses Kommando kopiert alle Nicht-System-Dateien von Laufwerk B zu Laufwerk A und überschreibt bei diesem Vorgang alle schreibgeschützten Dateien. Falls in die Operation mehrere verkettete Dateien eingeschlossen sind, muß der W-Parameter nur bei der letzten Datei in der Liste angegeben werden, wie das nachstehende Beispiel zeigt:

PIP A.DAT=B.DAT,F:NEW.DAT,G:OLD.DAT[W]

Dateien mit dem System-Attribut können bei Dateitransfers mit dem PIP-Kommando erfaßt werden, falls der R-Parameter angegeben ist; andernfalls werden Systemdateien nicht erkannt. Die Befehlszeile

PIP ED.COM=B:ED.COM[R]

z. B. liest die Datei ED.COM von Platte B auch dann, wenn sie als „schreibgeschützt“ und „Systemdatei“ markiert ist. Das Attribut „Systemdatei“ wird – falls vorhanden – mitkopiert.

Anmerkung: Um Dateien in einen anderen Benutzerbereich kopieren zu können, muß zuerst PIP-COM in diesen Benutzerbereich gebracht werden. Man folge der nachstehend angegebenen Prozedur, um eine Kopie von PIP.COM in einen anderen Benutzerbereich zu machen:

USER 0	Einloggen von Benutzer 0
DDT PIP.COM	PIP wird in den Speicher geladen und die Dateigröße s festgestellt
G0	Rückkehr zum CCP
USER 3	Einloggen von Benutzer 3
SAVE s PIP.COM	

wobei s die Anzahl der „Speicherseiten“ (256-Byte-Segmente) angibt, die von PIP belegt werden. Die Zahl s kann beim Laden von PIP.-COM unter DDT ermittelt werden, wenn man den Wert, der unter

der Anzeige NEXT steht, heranzieht (siehe Abschnitt 5.2.8). Ist z. B. die nächste verfügbare Adresse 1D00, dann braucht PIP.COM 1C (Hexadezimal) Seiten (oder 1 mal  $16 + 12 = 28$  Seiten). Der Wert von s beim darauffolgenden SAVE-Kommando ist deshalb 28.

Ist PIP einmal auf diese Weise kopiert worden, so kann es unter Verwendung der soeben erstellten PIP-Kopie durch normale PIP-Transfers auf eine andere Platte kopiert werden.

## 2.4.5 ED ufn

Das ED-Programm ist der Kontext-Editor des CP/M-Systems. Es erlaubt die Neuanlage und Veränderung von ASCII-Textdateien im Umfeld von CP/M. Die genaueren Einzelheiten der Arbeitsweise werden in Kapitel 3 angegeben. ED ermöglicht es dem Benutzer, Quelldateien zu generieren und zu bearbeiten. Diese Dateien sind dabei als Folge von ASCII-Zeichen organisiert, als Trennzeichen gelten Zeilenendezeichen (Aufeinanderfolge von Carriage Return und Line Feed). Es gibt keine praktische Einschränkung für die Zeilenlänge (keine einzelne Zeile darf die Größe des verfügbaren Arbeitsspeichers überschreiten), d. h., für die Anzahl der Zeichen, die Sie bis zur nächsten Betätigung der Abschlußtaste ( $\leftarrow$ ) eingeben dürfen.

Das ED-Programm stellt eine Anzahl von Befehlen zum Durchsuchen einer Zeichenfolge, zum Ersetzen und Einfügen von Zeichen bereit, die bei der Neuanlage und der Korrektur von Programmen oder Textdateien unter CP/M von Nutzen sind. Obwohl CP/M nur über einen begrenzten Speicherbereich für eingegebene Daten (annähernd 5 000 Zeichen bei einem 20K CP/M-System) verfügt, ist die zu editierende Dateigröße praktisch nicht beschränkt, da die Dateien in einfacher Weise fragmentiert (in „Seiten“ aufgeteilt) werden können. Falls die spezifizierte Quelldatei (noch) nicht existiert, legt ED sie an und öffnet sie für den Zugriff. Falls die Quelldatei dagegen existiert, fügt der Programmierer Daten, die editiert werden sollen an (siehe das A-Kommando). Diese angefügten Daten können dann am Bildschirm dargestellt, geändert und vom Arbeitsspeicher zurück auf die Platte geschrieben werden (siehe das W-Kommando). Bestimmte Stellen im Programm können automatisch ausgelagert und mit Hilfe des Kontextes gesucht werden (siehe das N-Kommando), was einen einfachen Zugriff auf ausgewählte Teile einer großen Textdatei zuläßt. Wenn Sie die Befehlszeile

eingegeben haben, legt das ED-Programm eine Hilfsdatei mit dem Namen X.\$\$\$ an, in der während des Editiervorgangs die editierten Daten aufbewahrt werden. Sind die Editierarbeiten erledigt, so wird die Datei X.ASM (Originaldatei) umbenannt in X.BAK und die Hilfsdatei X.\$\$\$ erhält den Namen X.ASM. Damit ist die Datei X.BAK identisch mit der Original (nicht editierten)-Datei, und die Datei X.ASM enthält die neueditierte Datei. Sie können jederzeit zu der früheren Version einer Datei zurückkehren, indem Sie die neueste Version löschen und die frühere Version umbenennen. Ist die derzeit gültige Fassung der Datei X.ASM unsachgemäß editiert worden, so wird durch die nachstehende Kommandofolge die Sicherungsdatei zurückgeholt:

DIR X.*	Prüft, ob die Sicherungsdatei existiert.
ERA X.ASM	Löscht die neueste Version.
REN X.ASM=X.BAK	Benennt die Sicherungsdatei in X.ASM um.

Der Operator kann den Editiervorgang an jeder Stelle (Warmstart, Netzausfall, Eingabe von ctl-C oder Eingabe des Q-Befehls) abbrechen, ohne die Originaldatei zu zerstören. In diesem Fall wird die Sicherungsdatei nicht angelegt und die Originaldatei bleibt erhalten.

Das ED-Programm gestattet es dem Benutzer, die Originaldatei auf einer Platte zu editieren und die Sicherungsdatei auf einer anderen Platte anzulegen. Zu diesem Zweck ist ein ED-Kommando der Form

ED ufn d:

abzusetzen, wobei ufn der Name der Datei ist, die auf der eingeloggten Platte editiert werden soll und d der Name eines anderen Laufwerks. Das ED-Programm liest und bearbeitet die Quelldatei und schreibt die neue Datei auf die Platte in Laufwerk d unter dem Namen ufn. Nach der Verarbeitung wird die Originaldatei zur Sicherungsdatei. Falls der Operator die Platte A adressiert, ist das folgende Kommando zulässig

ED X.ASM B:

Die Datei X.ASM wird auf Laufwerk A editiert, während auf der Platte im Laufwerk B die neue Datei X.\$\$\$ angelegt wird. Nach erfolgreichem Editieren wird A:X.ASM umbenannt in A:X.BAK und

B:X.\$\$\$ wird umbenannt in B:X.ASM. Aus Zweckmäßigkeitsgründen wird Laufwerk B am Ende des Editiervorgangs zum aktuellen Laufwerk. Man beachte, daß eine Meldung

### FILE EXISTS

an der Konsole ausgedruckt wird, wenn eine Datei mit dem Namen B:X.ASM vor Beginn des Editiervorgangs existiert. Dies ist eine Vorsichtsmaßnahme, die ein unbeabsichtigtes Zerstören einer Quelldatei vermeiden soll. Löschen Sie dann gegebenenfalls zuerst die bestehende Datei und rufen Sie den Editor von neuem auf.

Ähnlich den anderen nichtresidenten Programmen kann der Editor auf einem vom gerade eingeloggten abweichenden Laufwerk ablaufen. Man stellt dazu dem Namen der Quelldatei einen Laufwerksnamen voran. Beispiele zulässiger Editieranforderungen sind:

ED A:X.ASM	Editiert die Datei X.ASM auf Laufwerk A; die neue Datei und die Sicherungsdatei werden auf Laufwerk A abgelegt.
ED B:X.ASM A:	Editiert die Datei X.ASM auf Laufwerk B in die Hilfsdatei X.\$\$\$ auf Laufwerk A. Nach dem Editieren wird auf Laufwerk B X.ASM in X.BAK umbenannt und auf Laufwerk A X.\$\$\$ in X.ASM.

### 2.4.6 SYSGEN

Das nichtresidente SYSGEN-Kommando erlaubt die Generierung einer initialisierten Systemplatte, die das CP/M-Betriebssystem enthält. Das SYSGEN-Programm fordert über die Konsole Eingaben gemäß dem nachstehend gezeigten Dialog.

SYSGEN <CR>	Einleiten des SYSGEN-Programms
SYSGEN VERSION m.m	SYSGEN-Bereitschaftsmeldung
SOURCE DRIVE NAME (OR RETURN TO SKIP)	Man gibt als Antwort den Namen des Laufwerks ein, das die CP/M-Platte enthält (A oder B), gewöhn-

lich A. Falls eine Kopie von CP/M bereits im Speicher vorliegt, gibt man nur ↵ ein. Eingabe eines Laufwerksnamens d veranlaßt folgende Antwort des Systems.

SOURCE ON d THEN  
TYPE RETURN

Man legt eine Diskette, die das CP/M-Betriebssystem enthält, in Laufwerk d (d ist A oder B). Sobald dies geschehen ist, drückt man die ↵ -Taste.

FUNCTION COMPLETE

Das Betriebssystem ist in den Speicher übertragen. SYSGEN meldet sich dann wieder mit der Anforderung:

DESTINATION DRIVE  
NAME (OR RETURN TO  
REBOOT)

Soll die Diskette das CP/M-System erhalten, so wird sie in ein Laufwerk eingelegt. Der Name dieses Laufwerks wird gegebenenfalls als Antwort angegeben. Andernfalls wird die ↵ -Taste gedrückt und das System führt von Laufwerk A aus einen Neustart durch. Die Eingabe eines Laufwerksnamens d veranlaßt SYSGEN zu folgender Meldung:

DESTINATION ON d  
THEN TYPE RETURN

Die neue Diskette liegt in Laufwerk d; es muß nur noch ↵ gedrückt werden. Antwort des Systems:

FUNCTION COMPLETE

Das Betriebssystem befindet sich nun auf den äußeren Spuren der Zielplatte in Laufwerk d.

Die „DESTINATION“-Anforderung wird wiederholt bis ein ↵ ohne Angabe eines Laufwerks an der Konsole eingegeben wird, so daß mehr als eine Diskette mit dem Betriebssystem versehen werden können.

Nach dem Abschluß einer erfolgreichen Systemgenerierung enthält die neue Platte das Betriebssystem. Dabei sind aber nur die

residenten Kommandos verfügbar. Eine formatierte Diskette erscheint CP/M als Platte mit einem leeren Inhaltsverzeichnis; daher müssen Sie die notwendigen COM-Dateien von einer vorhandenen CP/M-Diskette auf die neu generierte Diskette mittels des nichtresidenten PIP-Programms kopieren. Der Benutzer kann das Kopieren aller Dateien von einer bestehenden Diskette durch Eingabe des PIP-Kommandos

```
PIP B:=A:*. *[v]
```

bewerkstelligen. Dadurch werden alle Dateien der Platte in Laufwerk A auf die Platte in Laufwerk B kopiert, wobei geprüft wird, ob jede Datei korrekt übertragen worden ist. Der Name jeder Datei wird an der Konsole angezeigt.

Der Benutzer sollte beachten, daß eine Systemgenerierung mit SYSGEN die Dateien, die auf einer Diskette bereits existieren, nicht zerstört. SYSGEN konstruiert lediglich ein neues Betriebssystem (nur die beiden äußeren Spuren der Diskette sind betroffen!). Falls eine Platte (Diskette) nur in den Laufwerken B bis P benutzt werden soll und niemals als Herkunft einer Bootstrap-Operation in Laufwerk A, braucht SYSGEN mit dieser Platte nicht durchgeführt zu werden.

#### 2.4.7 SUBMIT ufn parm#1 . . . parm#n

Das SUBMIT-Kommando erlaubt die Zusammenfassung von CP/M-Befehlen zu einem Stapel zum Zweck automatischer Verarbeitung. Der im SUBMIT-Befehl angegebene Name ufn muß eine Datei sein, die auf der gerade eingeloggten Platte existiert und den Dateityp „SUB“ aufweist. Die SUB-Datei enthält CP/M-Prototyp-Befehle mit möglicher Parameterersetzung. Die aktuellen Parameter parm#1, . . . parm#n werden in die Prototyp-Kommandos eingesetzt und – falls keine Fehler auftreten – wird die Datei mit den aktuellen Kommandos anschließend von CP/M sequentiell abgearbeitet.

Die Prototyp-Befehlsdatei wird unter Benutzung des ED-Programms erstellt, wobei „\$“-Parameter der Form

```
$1$2$3 . . . $n
```

eingestreut werden, die Platzhalter für aktuelle Parameter sind. Diese werden angegeben, wenn die Datei zur Abarbeitung bereitgestellt wird: Wenn das nichtresidente SUBMIT-Programm zur Ausführung kommt, werden in den Prototyp-Befehlen die formalen Parameter \$1, . . . \$n entsprechend der Reihenfolge durch die aktuellen Parameter parm#1 . . . parm#n ersetzt.

Stimmt die Anzahl der formalen Parameter mit der der aktuellen Parameter nicht überein, so wird die Arbeit von SUBMIT abgebrochen und es erscheint eine Fehlermeldung an der Konsole.

Die SUBMIT-Funktion erstellt auf der eingeloggten Platte eine Datei \$\$\$SUB, die bereits die endgültigen Befehlszeilen enthält. Wenn das System zum Betriebssystem zurückkehrt (bei Abschluß von SUBMIT), wird diese Kommandodatei vom CCP als Herkunft der Eingabe an Stelle der Konsoleingabe gelesen. Wird die SUBMIT-Funktion auf irgendeiner anderen Diskette als der in Laufwerk A durchgeführt, so werden die einzelnen Befehle nicht abgearbeitet, bevor die Diskette in Laufwerk A eingelegt und das System neu gestartet wird. Der Benutzer kann die Befehlsabarbeitung zu jeder Zeit durch Drücken der RUB OUT-Taste (|←|) abbrechen, und zwar immer dann, wenn ein Kommando gelesen und als Quittung an der Konsole angezeigt wird. In diesem Fall wird die \$\$\$SUB-Datei gelöscht und die nachfolgenden Kommandos kommen von der Konsole. Die Befehlsabarbeitung wird auch abgebrochen, wenn der CCP einen Fehler in einem der Kommandos feststellt. Programme, die unter CP/M ablaufen, können die Abarbeitung einer Kommandodatei abbrechen, wenn Fehlersituationen durch Löschen einer bestehenden \$\$\$SUB-Datei auftreten.

Um das Dollarzeichen in eine SUBMIT-Datei einzubringen, kann der Benutzer ein „\$\$“ eingeben, das innerhalb der Kommandodatei zu einem einzigen „\$“ reduziert wird. Ein „↑“-Symbol kann dem Buchstaben x vorangehen und erzeugt dann ein ctl-x innerhalb der Datei.

Das letzte Kommando in einer SUB-Datei kann eine weitere SUB-Datei initiieren, wodurch eine Verkettung von Batch-Kommandos möglich wird.

Als Beispiel nehme man nun an, die Datei ASMBL.SUB existiere bereits auf der Platte A und sie enthalte die Prototyp-Befehle

ASM \$1

DIR \$1.\*

ERA\*.BAK

PIP \$2:=\$1.PRN

ERA \$1.PRN

und vom Operator werde das Kommando

## SUBMIT ASMBL X PRN

eingegeben. Das SUBMIT-Programm liest die Datei ASMBL.SUB, setzt „X“ an allen Stellen für \$1 ein und „PRN“ für \$2. Dadurch entsteht eine \$\$\$SUB-Datei, welche die Befehlszeilen

ASM X

DIR X.\*

ERA\*.BAK

PIP PRN:=X.PRN

ERA X.PRN

enthält, die nacheinander durch den CCP abgearbeitet werden.

Die SUBMIT-Funktion kann eine SUB-Datei die auf einer anderen Platte liegt, durch Voranstellung der Laufwerkbezeichnung beim Dateinamen aufrufen. Verarbeitet werden können SUBMIT-Dateien nur in Laufwerk A. Trotzdem kann man eine Kommandodatei in Laufwerk B anlegen; abgearbeitet werden kann sie zu einem späteren Zeitpunkt, wenn die entsprechende Diskette in Laufwerk A eingelegt wird.

Ein weiteres Dienstprogramm mit dem Namen XSUB erweitert die Leistungsfähigkeit der SUBMIT-Einrichtung durch die Möglichkeit, gepufferte Eingabe über die Konsole abzufangen und in die SUB-Datei einzufügen. Das XSUB-Kommando bildet immer die erste Zeile einer SUBMIT-Datei. Bei der Ausführung wird es direkt unter den CCP-Bereich geladen. Alle nachfolgenden SUBMIT-Befehlszeilen werden durch XSUB abgearbeitet, so daß Programme, die gepufferte Konsoleingaben anfordern, die entsprechenden Eingaben direkt aus der SUBMIT-Datei erhalten. Im folgenden Beispiel soll die Datei SAVER.SUB die Befehlszeilen

XSUB

DDT

I\$1.COM

R

GØ

## SAVE 1 \$2.COM

enthalten. Es wird ein SUBMIT-Kommando

```
A>SUBMIT SAVER PIP Y
```

eingegeben, das PIP an die Stelle von \$1 und Y an die Stelle von \$2 setzt.

Zunächst wird XSUB geladen, gefolgt von DDT, das für die Befehlszeilen PIP.COM, R und GØ aktiviert ist. Mit GØ erfolgt die Rückkehr zum CCP. Der Abschlußbefehl SAVE 1 Y.COM wird durch den CCP verarbeitet.

Das XSUB-Programm verbleibt im Arbeitsspeicher und gibt bei jeder Warmstartoperation die Meldung an der Konsole aus

```
(xsub active)
```

um seine Präsenz anzuzeigen. Nachfolgende SUBMIT-Befehlsströme erfordern kein XSUB, falls nicht aus irgendeinem Grund ein Kaltstart notwendig sein sollte.

Für den Fall, daß XSUB und das CP/M-DESPOOL-Dienstprogramm simultan ablaufen sollen, ist zu beachten, daß XSUB hinter DESPOOL geladen werden muß.

### 2.4.8 DUMP ufn

Das DUMP-Programm schreibt den Inhalt der Plattendatei (ufn) an der Konsole in hexadezimaler Form aus. Der Dateiinhalt wird dabei so ausgegeben, daß jeweils sechzehn Bytes in einer Zeile stehen. Die absolute Byteadresse ist am linken Rand jeder Zeile in hexadezimaler Form aufgelistet. Lange Auslistungen können durch Drücken der RUB OUT-Taste (|←|) abgebrochen werden.

### 2.4.9 FORMAT

Das FORMAT-Programm initialisiert eine Diskette für die Verwendung im Computer. Es werden Informationen auf die Platte geschrieben, anschließend erfolgt ein Prüfen jeder Spur. Leicht verständliche Anweisungen zur Handhabung der Disketten erscheinen am Bildschirm.

## 2.4.10 CONFIG

Das CONFIG-Programm erzeugt für den Computer bestimmte Information, wodurch die Verständigung zwischen dem Computer und den angeschlossenen Geräten erst möglich wird. Die Leistungen dieses Programmes sind sowohl auf der Hardware- als auch auf der Software-Ebene von Bedeutung.

Insbesondere erlaubt dieses Programm dem Programmierer die Definition (und Neudefinition) der programmierbaren Funktionstasten, den Austausch des LST:Geräts und die Änderung der vorgelegten Charakteristika der beiden seriellen Schnittstellen, eine Änderung der Plattenkonfiguration, die Vorgabe neuer Werte für die Rückschreibe- und Wiederholungszähler, die Änderung der Cursoranzeige und die Einstellung des Automatikstarts. Mit Ausnahme der Druckerfunktionen können die Änderungen dieser Konfigurationsparameter wahlweise als permanent (sowohl im Hauptspeicher als auch auf der Diskette) oder temporär (nur im Hauptspeicher) durchgeführt werden.

Anmerkung: Jeder System-Kaltstart macht temporäre Änderungen rückgängig, da CP/M von der Platte geholt wird.

Alle Funktionen von CONFIG werden durch leicht verständliche Hinweise am Bildschirm erklärt, wobei mit einem Haupt-Menü begonnen wird, das die änderbaren Parameter auflistet. Das Programm wird aufgerufen durch Eingabe von

### CONFIG

Am Bildschirm erscheint dann eine Copyright-Aussage und danach das Hauptmenü mit den Funktionen, die im folgenden beschrieben werden.

#### 1. Modifizieren der Funktionstasten

Die Funktion von jeder der 20 programmierbaren Funktionstasten kann durch den Programmierer festgelegt werden, unabhängig davon, ob sie bereits vom System vordefiniert wurde. Wird diese CONFIG-Funktion benutzt, dann kann sich der Programmierer alle Definitionen am Bildschirm anzeigen lassen, bevor er irgendwelche Änderungen vornimmt. Tastendefinitionen mit Hexadezimalwerten von 00 bis 1F (Steuerzeichen) werden im USASI-Code (siehe Anhang) angezeigt, jeweils zwischen den Symbolen <>; diese Symbole sind nicht Teil der Definition.

Die Definition, die keines der Hexadezimalzeichen aus dem Bereich von 80-FF enthalten darf, wird in einer Funktionstabelle niedergelegt. Diese Tabelle darf maximal 236 Zeichen enthalten. Gibt man mehr Zeichen ein, so wird die Meldung

#### FNCT TABLE FULL (CR)

ausgegeben. ↵ als Antwort löscht die Definition der Funktionstaste, die den Überlauf verursacht hat. Die Definition einer Taste kann jederzeit durch Anforderung der Funktion zur Neudefinition und darauffolgende (Neu-)Spezifizierung der Funktionstaste gelöscht werden. Zum Abschluß der Neudefinition wird die betreffende Funktionstaste gedrückt.

### 2. Modifizieren der LST:Gerät - Zuordnung

CP/M ist bereits auf den Anschluß eines Druckers mit einer Centronics-Schnittstelle vorbereitet. Falls ein serieller Drucker 1 oder 2 benutzt werden soll, muß das I/O Byte modifiziert werden, um das richtige Druckgerät wiederzugeben. Das Gleiche trifft zu, wenn Sie anstatt des Drucks eine Bildschirmanzeige wünschen.

Soll ein serieller Drucker benutzt werden, dann können auch neue Parameter für den Drucker definiert werden (siehe Funktion 4 oder 5).

**Wichtig:** Wenn Sie in CP/M durch ein Kommando bzw. Steuerzeichen einen Drucker ansprechen, muß ein Drucker tatsächlich bereits angeschlossen sein. Bei einem Druckeraufruf erwartet CP/M eine Bereitschaftsmeldung des Druckers. Falls er nicht angeschlossen ist, können Sie erst nach einem Kaltstart weiterverarbeiten.

### 3. Modifizieren der Rückschreibe- und Wiederholungszähler

Der Anfangswert dieser beiden Zähler zur Fehlerkorrektur ist 4. Die Anzahl der Platten Lese/und Schreibversuche kann durch Spezifizierung eines anderen Wertes zwischen 0 und 9 vergrößert oder vermindert werden.

#### 4. Modifizieren der seriellen Schnittstelle für Drucker 1

Die Parameter für den seriellen Drucker haben die Anfangswerte

Stop Bit = 1

Parity = aktiviert, gerade

Zeichenlänge = 7 Bits

Baud Rate = 9 600

niedrigste Portadresse = 60 Hex

Durch Auswahl aus der Funktionsliste kann jeder Parameter modifiziert werden. Den neuen Wert wählt man aus dem Angebot, das am Bildschirm dargestellt wird. Will man z. B. die Zeichenlänge ändern, erscheinen auf dem Bildschirm die Optionen von 5,6,7 oder 8 Bits.

#### 5. Modifizieren der seriellen Schnittstelle für den Plotter bzw. Drucker 2

Die Parameter für den seriellen Drucker haben die Anfangswerte.

Stop Bit = 1

Parity = aktiviert, gerade

Zeichenlänge = 7 Bits

Baud Rate = 9600

niedrigste Portadresse = B0

Durch Auswahl aus der Funktionsliste kann jeder Parameter modifiziert werden. Den neuen Wert wählt man aus dem Angebot, das am Bildschirm dargestellt wird.

## 6. Modifizieren der Platten-Konfiguration

Der anfängliche Zustand von CP/M geht von der Annahme aus, daß zwei Disketten-Laufwerke in Betrieb sind. Wenn Sie von einem Festplatten-Laufwerk Gebrauch machen wollen, müssen Sie die Platten-Konfiguration im Computer entsprechend ändern. Falls Sie ein System mit einem Disketten-Laufwerk und einem Festplatten-Laufwerk besitzen, muß der Wert jeweils in 1 geändert werden. (Das Festplatten-Laufwerk wird vom System als zwei logische Einheiten mit je 5 MB Speicherkapazität betrachtet.)

Die Platten-Konfiguration soll als permanente Änderung ausgeführt werden. Die Änderung wird allerdings auch ohne Kaltstart wirksam.

## 7. Modifizieren der Cursoranzeige

Diese Funktion ändert die Darstellungsart der Schreibmarke. Diese kann als Block oder als Unterstreichung dargestellt werden, ferner kann man beide Zeichen ständig oder blinkend anzeigen lassen. Die eingestellte Betriebsart ist "blinkende Unterstreichung".

## 8. Modifiziert den Automatikstartbetrieb

CP/M sieht die Möglichkeit vor, nach einem Kaltstart, einem Warmstart oder einem Kalt- und Warmstart einen Befehl automatisch auszuführen. Sie können eine dieser Möglichkeiten wählen und den auszuführenden Befehl eintragen bzw. ändern.

## X. Programmrückkehr

Diese Funktion wird benutzt, um anzuzeigen, ob die Änderung, der CP/M unterzogen werden soll, im Speicher und auf der Platte (permanente Änderung) oder nur im Speicher (temporäre Änderung) vorzunehmen ist. Falls sowohl permanente als auch temporäre Änderungen gewünscht sind, müssen zuerst die permanenten Änderungen ausgeführt werden.

### 2.4.11 EXCHANGE

Das EXCHANGE-Programm ermöglicht es, Nicht-NCR-Disketten mit CP/M-Anwendungsprogrammen auf NCR DECISION MATE V zu verwenden. (Die Diskette muß natürlich die Abmessung 5 1/4-inch aufweisen).

Dieses Programm informiert BIOS über die Formatierung der fremden Diskette. Es werden hierbei Teile des Betriebssystems ausgetauscht, damit das Lesen der fremden Diskette erfolgen kann.

Das EXCHANGE-Programm zeigt alle Nicht-NCR-Disketten am Bildschirm an, die mit NCR DECISION MATE V gelesen werden können. Falls die vorliegende Fremd-Diskette in der angezeigten Liste nicht aufgeführt ist, bedeutet dies, daß sie für die Verarbeitung nicht benutzt werden kann.

Zur Ausführung von EXCHANGE sind folgende Schritte erforderlich:

1. Einlegen der NCR CP/M Systemdiskette in Laufwerk A
2. Start des EXCHANGE-Programms

#### EXCHANGE

3. Auswählen des Formats der Diskette aus der Tabelle am Bildschirm
4. Ausführen der EXIT-Funktion
5. Bei Systemen mit nur einem Disketten-Laufwerk wird folgende Aufforderung angezeigt:

Mount "non-NCR" Disk in Drive A (CR).

Daraufhin müssen Sie die Nicht-NCR-Diskette einlegen und ↵ betätigen.

BIOS hat danach die Information übernommen, die das Format der fremden Diskette betrifft. Die Diskette steht jetzt zur Verarbeitung mit Standard-CP/M-Programmen zur Verfügung.

Wenn das System zwei Disketten-Laufwerke besitzt, dann bleibt diese Information im BIOS bis zum nächsten EXCHANGE oder Kaltstart wirksam. In einem System mit einem Disketten-Laufwerk wird diese Information bereits bei dem nächsten Warmstart vernichtet.

Es ist zu beachten, daß BIOS nur einen Eintrag über ein Fremdformat aufnehmen kann. Sollen Fremddisketten mit unterschiedlichem Format verwendet werden, ist es erforderlich, EXCHANGE vor der Benutzung jeder Fremddiskette einzusetzen.

## 2.4.12 DISCIT

Das DISCIT-Programm richtet die Festplatte für Ihr System ein (Initialisierung). Jede Spur wird mit Prüfdaten belegt. Diese Daten werden dann zwecks Fehlerprüfung von dem System gelesen.

Nach dem Aufruf von DISCIT erscheinen die Copyright-Information sowie das DISK TYPE MENU. Durch Eingabe von 1 wird der DISCIT-Vorgang eingeleitet. Nun werden Sie am Bildschirm aufgefordert, die Bezeichnung der Festplatten-Einheit, die initialisiert werden soll, einzugeben. Nach Eingabe des entsprechenden Buchstabens drücken Sie  $\leftarrow$  .

Die Auswahl der Festplatten-Einheit wird am Bildschirm bestätigt. Gleichzeitig werden Sie aufgefordert, die gewünschte Zahl der Prüfungsvorgänge einzugeben. Diese werden dann wie oben beschrieben erfolgen. Der Standardwert dieser Fehlerprüfung ist 5. Sie können aber eine beliebige Zahl zwischen 1 und 99 einschließlich eingeben. Zum Abschluß dieser Eingabe betätigen Sie  $\leftarrow$  .

Gemäß der Mitteilung, die nun am Bildschirm erfolgt, können Sie die Formatierung durch einfaches Drücken von  $\leftarrow$  starten. DISCIT belegt die Spuren der Festplatte nacheinander in aufsteigender Reihenfolge mit Prüfdaten. Diese Daten werden von der jeweiligen Spur von DISCIT wieder gelesen und auf ihre Richtigkeit geprüft. Die Nummer der gerade geprüften Spur wird jeweils am Bildschirm angezeigt. Sie können diesen Vorgang zu jeder Zeit durch Drücken von  $\text{ctl-C}$  abbrechen.

Sollte DISCIT während dieser Prüfung einen Fehler feststellen, so wird der dazugehörige Block als „untauglich“ zurückgewiesen (Bad Block). Vorausgesetzt, daß die Anzahl der Bad Blocks  $< 9$  ist, wird der Benutzer nicht über diese Fehler in Kenntnis gesetzt. (Die Fehler werden trotzdem zu einem Verlust an Speicherkapazität führen.) Falls die Anzahl solcher Fehler  $> 8$  ist, erscheint die Meldung "BAD MEDIA". Dies bedeutet, daß sie Festplatte nicht als zuverlässiges Speichermedium betrachtet werden kann.

Sobald DISCIT beendet ist, erscheinen die Meldungen "End Format" und "Would you like to format any more? (Y/N)". Durch Eingabe von "N" wird DISCIT verlassen, während "Y" einen nochmaligen Ablauf von DISCIT bewirkt.

## 2.4.13 BACKUP

Mit BACKUP können Sie beliebige Dateien von einer Platte auf eine andere kopieren. Sie können auch Dateien am Bildschirm oder am Drucker ausgeben.

Das Gruppensymbol \* ist erlaubt. Damit können Sie den Kopiervorgang auf eine bestimmte Gruppe von Dateien ausweiten.

BACKUP läßt sich auch zwecks Kopierens einer Datei unter einer anderen Benutzernummer einsetzen.

Wenn eine Datei mehr als den auf der Zielplatte vorhandenen Speicherplatz erfordert, wird sie in zwei oder mehrere Dateien aufgeteilt. Sie sollten die auf diese Weise aufgeteilten Dateien bei nächster Gelegenheit mit BACKUP wieder zusammenfügen, um die Gefahr einer Verwechslung der Reihenfolge der Teildateien zu vermeiden.

Eine etwaige in Ihrer Kommandozeile angegebene Dateibezeichnung verweist auf eine von Ihnen erstellte Datei, die Ihre Antworten auf die von BACKUP gestellten Fragen enthält.

Die Ausführung des BACKUP-Kommandos erfordert eine Maschinenspeicher-Kapazität von mindestens 128 KB.

Wichtigster Bestandteil des Dialogs mit BACKUP ist die am Bildschirm erscheinende Aufforderung "Enter command." Abbildung 2.1 enthält eine Beschreibung der Ihnen zur Verfügung stehenden Antwortmöglichkeiten.

Befehl	Wirkung
END	beendet die Ausführung von BACKUP. CP/M-80 wartet auf Ihre nächste Kommandozeile.
FILE oder FILES	Sie können eine Reihe von Dateinamen eingeben. Diese Eingabe wird mit einer Leerzeile abgeschlossen.
GO	Kopieren der gewünschten Datei(en).
INUSER:xx	<p>Nur die Dateien der Benutzernummer xx werden einbezogen. Ihre Auswahl einer Benutzernummer wird durch die Meldung "New input user number is: xx" bestätigt.</p> <p>Ihre Eingabe für xx darf nur Ziffern enthalten. Ein Verstoß führt zur Meldung "Bad user number – ignored."</p> <p>Wenn Sie eine Nummer größer als 15 eingeben, wird sie bei Anzeige der Meldung "Default user number selected" nicht berücksichtigt.</p>
OUTUSER:yy	Die Dateien sollen auf die Benutzernummer yy kopiert werden. Die bei INUSER :xx angegebenen Meldungen finden auch hier Anwendung.
NEW	bereitet BACKUP auf das Einlegen einer neuen Diskette vor.
REBUILD	stellt die Einheit einer von BACKUP aufgeteilten Datei wieder her.
RESET	<p>stellt die folgenden Standard-Werte für BACKUP ein:</p> <p>Kopieren mit Verifikation  Die Angabe einer zu kopierenden Datei liegt nicht vor  Unter Benutzernummern wird nicht unterschieden.</p>
VERON	Verifikation (nochmaliges Lesen der soeben übertragenen Daten zwecks Prüfung auf Fehlerfreiheit) findet während des Kopiervorgangs statt.
VEROFF	Verifikation außer Kraft.

Abb. 2.1 BACKUP-BEFEHLE

BACKUP fordert Sie zunächst auf, die Bezeichnungen der Laufwerke einzugeben, die die zu kopierenden Dateien ("Enter source drive") bzw. die Zielplatte des Kopiervorgangs ("Enter destination device") enthält. Ihre Antwort darf jeweils der Kennbuchstabe eines vorhandenen Laufwerks sein. Bei der Frage nach dem Zielgerät sind zusätzlich die Antworten LST: (für Drucker) und CON: (für Bildschirm) zulässig.

Die Aufforderung "Enter command" erfordert als Antwort einen der in Abbildung 2.1 enthaltenen Befehle. Wenn BACKUP eine Dateibezeichnung als Eingabe benötigt, erscheint die Aufforderung "Enter file name" am Bildschirm.

Sobald Sie den GO-Befehl eingegeben haben, werden Sie aufgefordert, eine bereits formatierte Platte in ein bestimmtes Laufwerk einzulegen: "Mount initialized media in drive . . . (Seq. . .).

Der REBUILD-Vorgang wird mit der Aufforderung "Enter file name ".REB":" eingeleitet. Sie müssen nun die Bezeichnung der Datei eingeben, die die Steuerungsanweisungen für die Wiederherstellung der aufgeteilten Datei beinhaltet. Diese Dateibezeichnung muß den Namenszusatz .REB aufweisen. Eine Reihe von Meldungen am Bildschirm hilft Ihnen, die Platten in der richtigen Reihenfolge einzulegen: "Mount disk sequence number x." REBUILD geht davon aus, daß die Teildateien der wiederherzustellenden Datei jeweils einen aus drei Zeichen bestehenden Namenszusatz besitzen, wobei das erste Zeichen der Buchstabe M sein muß.

Ihre Benutzung von BACKUP wird von einem ausführlichen Dialog unterstützt. Die von BACKUP ausgegebenen Meldungen sind im Anhang C dieses Handbuchs enthalten. Sollte bei der Benutzung von BACKUP ein Fehlerzustand eintreten, können Sie durch Betätigen der Taste X einen Warmstart des CP/M-86-Betriebssystems auslösen.

#### Beispiel 1:

Die auf der (Fest-) Platte des Laufwerks C befindliche Datei FILE.MVX (Größe 856KB) soll auf Disketten, die nacheinander in Laufwerk A eingelegt werden, übertragen werden. Das BACKUP-Kommando ist auf der Platte im Laufwerk B vorhanden.

```
B>BACKUP
BACKUP for NCR DECISION MATE V
Version x.x.
Copyright (c), NCR CORPORATION 1983
```

Enter source drive: C  
Enter destination device: A  
Reading directory .....

Total number of files: 279  
Files on user number 0: 279

Enter command: VEROFF ↓  
Enter command: FILE ↓  
Enter file name: FILE.MVX ↓  
1 File found  
Enter file name:

Enter command: GO ↓

Mount initialized media in drive A (Seq 1)

File FILE .REB written to drive A  
File FILE .MVX (Part 1) – 302KB copied

Mount initialized media in drive A (Seq 2)

File FILE .MVX (Part 2) – 304KB copied

Mount initialized media in drive A (Seq 3)

File FILE .MVX (Part 3) – 244KB copied

Enter command: END ↓

BACKUP complete

B>

Beispiel 2:

Die auf drei Disketten aufgeteilte Datei FILE.MVX soll auf der Festplatte im Laufwerk B als eine Datei wieder hergestellt werden. Die Disketten werden nacheinander in Laufwerk A eingelegt. Das BACKUP-Kommando befindet sich auf der Platte im Laufwerk B.

B>BACKUP ↓  
BACKUP for NCR DECISION MATE V

Version x.x  
 Copyright (c) NCR CORPORATION 1983

Enter source drive: A  
 Enter destination device: B  
 Reading directory .....

Total number of files: 2  
 Files on user number 0: 2

Enter command: VEROFF ↵  
 Enter command: REBUILD ↵

Enter file name "x.REB": FILE ↵  
 Mount disk sequence number 1 (Erste Diskette in A, dann ↵ )  
 Mount disk sequence number 2 (Zweite Diskette in A, dann ↵ )  
 Mount disk sequence number 3 (Dritte Diskette in A, dann ↵ )  
 File FILE .MVX rebuilt on drive B  
 Enter command: END ↵

BACKUP complete

B>

Beispiel 3:

Eine Datei BACKUPIN enthält eine Sequenz von BACKUP-Befehlen, die das Kopieren einer auf der (Fest-) Platte im Laufwerk C befindlichen Datei auf zwei Disketten im Laufwerk A bewirken soll. Der genaue Inhalt von BACKUPIN:

```
C
A
FILE
BACKUP.SRC
BACKUP.PRN
BACKUP.REL
BACKUP.COM
BDOSBKP.MAC
BDOSBKP.PRN
BDOSBKP.REL
```

GO  
NEW  
FILE  
CONFIG.SRC  
CONFIG.PRN  
CONFIG.REL  
CONFIG.COM  
BDOSCONF.MAC  
BDOSCONF.REL  
BDOSCONF.PRN

GO  
END

Nach Eingabe der BACKUP-Kommandozeile

B>BACKUP BACKUPIN ↵

werden diese Befehle gemäß folgender Darstellung ausgewertet:

BACKUP for NCR DECISION MATE V  
Version x.x  
Copyright (c), NCR CORPORATION 1983

Reading directives from file: BACKUPIN

Enter source drive: C  
Enter destination device: A  
Reading directory .....

Total number of files: 273  
Files on user number 0: 273

Enter command: FILE  
Enter file name: BACKUP.SRC  
1 File found  
Enter file name: BACKUP.PRN  
1 File found  
Enter file name: BACKUP.REL  
1 File found  
Enter file name: BACKUP.COM  
1 File found  
Enter file name: BDOSBKP.MAC  
1 File found

Enter file name: BDOSBKP.PRN

1 File found

Enter file name: BDOSBKP.REL

1 File found

Enter file name:

Enter command: GO

Mount initialized media in drive A (Seq 1)

File BACKUP	.REL	— 20k Bytes copied
File BACKUP	.COM	— 28k Bytes copied
File BACKUP	.PRN	— 86k Bytes copied
File BDOSBKP	.MAC	— 6k Bytes copied
File BDOSBKP	.PRN	— 10k Bytes copied
File BDOSBKP	.REL	— 2k Bytes copied
File BACKUP	.SRC	— 50k Bytes copied

Enter command: NEW

Enter command: FILE

Enter file name: CONFIG.SRC

1 File found

Enter file name: CONFIG.PRN

1 File found

Enter file name: CONFIG.REL

1 File found

Enter file name: CONFIG.COM

0 Files found

Enter file name: BDOSCONF.MAC

1 File found

Enter file name: BDOSCONF.REL

1 File found

Enter file name: BDOSCONF.PRN

1 File found

Enter file name:

Enter command: GO

Mount initialized media in drive A (Seq 2)

File CONFIG	.PRN	— 60k Bytes copied
File CONFIG	.REL	— 22k Bytes copied
File BDOSCONF	.MAC	— 6k Bytes copied
File BDOSCONF	.REL	— 2k Bytes copied
File BDOSCONF	.PRN	— 8k Bytes copied
File CONFIG	.SRC	— 32k Bytes copied

Enter command: END

BACKUP complete  
B>

## 2.5 BDOS-Fehlermeldungen

Es gibt drei Fehlersituationen, die das Basic Disc Operating System bei der Dateiverarbeitung abfängt. Falls eine dieser Bedingungen gegeben ist, druckt BDOS die Meldung aus:

BDOS ERR ON d: error

wod d der Name des Laufwerks und „error“ eine der drei Fehlermeldungen:

BAD SECTOR

SELECT

READ ONLY

ist.

Die „BAD SECTOR“-Meldung wird ausgegeben, falls eine BIOS Plattenfehlermeldung mit „O“ beantwortet wurde. Die „BAD SECTOR“-Meldung zeigt an, daß die Elektronik des Plattenkontrollers einen Fehler beim Lesen oder Beschreiben der Platte festgestellt hat.

Diese Situation wird im allgemeinen durch Fehler in der Plattensteuerung oder eine extrem abgenutzte Diskette verursacht. Stellt der Benutzer fest, daß CP/M diesen Fehler öfter als einmal im Monat meldet, sollten die Elektronik der Plattensteuerung und der Zustand des Datenträgers überprüft werden. Dem Benutzer kann diese Situation auch beim Lesen von Dateien begegnen, die von einer Plattensteuerung eines anderen Herstellers generiert wurden. Die Lösung aus dieser Fehlersituation wird durch Eingabe von `ctl-C` zum Zweck des Neustarts (der sicherste Weg) erreicht oder durch Drücken von `↵`, was den fehlerhaften Sektor in der Dateioperation einfach ignoriert. Sie sollten jedoch beachten, daß die Eingabe `↵` die Datenintegrität auf der Platte zerstören kann, insbesondere dann, wenn die Operation Schreiben in das Directory einschließt. Man sollte dies nur machen, falls man über geeignete Datensicherungen verfügt.

Der „SELECT“-Fehler tritt auf, wenn versucht wird, ein Laufwerk zu adressieren, dessen Bezeichnung außerhalb des Bereichs liegt, den BIOS unterstützt. In diesem Fall gibt der Wert von `d` in der Fehlermeldung die angewählte Laufwerksbezeichnung an.

Das System führt bei jeder Eingabe von der Konsole einen Neustart durch.

Die „READ ONLY“ Meldung erscheint beim Versuch, eine Diskette oder Datei zu beschreiben, die in einem STAT-Befehl als „schreibgeschützt“ bestimmt wurde oder einen Schreibschutz durch BDOS erhalten hat. Sie sollten CP/M durch die Warmstartprozedur (`ctl-C`) oder durch Ausführen eines Kaltstarts beim nächsten Plattenwechsel neu starten. Falls eine gewechselte Platte gelesen aber nicht beschrieben werden soll, läßt BDOS den Plattenwechsel ohne Warm- oder Kaltstart zu, markiert intern das Laufwerk aber als nur lesbar. Der Status des Laufwerks wird hernach durch einen Warm- oder Kaltstart in „lesbar/beschreibbar“ abgeändert.

Bei der Ausgabe obiger Meldung erwartet CP/M eine Eingabe von der Konsole. Ein automatischer Warmstart erfolgt nach jeder Eingabe.

)

)

)

# ED

## 3.1 Einführung in ED

ED ist der zeilenorientierte Editor von CP/M zur Anlage und Veränderung von CP/M Textdateien. Er wird aufgerufen durch Eingabe von

ED dn  
oder  
ED dn. typ

Dabei steht Dn abkürzend für „Dateiname“.

Im allgemeinen liest ED Segmente der Originaldatei Dn oder Dn.Typ in den Hauptspeicher, und zwar in die TPA. Ausschließlich dort erfolgt die Bearbeitung. Sind die Änderungen abgeschlossen, schreibt ED die Datei zurück auf die Platte. Existiert die beim Aufruf von ED angesprochene Datei noch nicht, wird sie durch ED angelegt und in angemessener Weise initialisiert (Zustand „leer“). Einen Überblick über die Arbeitsweise von ED gibt Fig. 3.1.

### 3.1.1 Arbeitsweise von ED

Mit ED werde eine Textdatei bearbeitet, die in Fig. 3.1 den Namen Dn. TXT hat. ED schleust den gesamten Text aus der Datei über einen Pufferspeicher in einen Zwischenspeicher. Während des Verweilens im Pufferspeicher kann der Text auf dem Bildschirm angezeigt oder verändert werden (die Anzahl der Textzeilen, die vom Pufferspeicher aufgenommen werden kann, ist abhängig von der Zeilenlänge; der Pufferspeicher hat bei einem 20K CP/M-System ein Gesamtfassungsvermögen von etwa 5 000 Zeichen!). Der bearbeitete Text wird dann gemäß geeigneter Benutzerbefehle in eine Zwischendatei geschrieben. Bei Beendigung des Editiervorgangs schreibt ED den gesamten Pufferspeicherinhalt in die Zwischendatei, gefolgt von dem evtl. vorhandenen (nicht bearbeiteten) Restteil der (Original) Textdatei. Der Name der Originaldatei wird in „Dn.BAK“ umgeändert. Damit kann man bei Bedarf immer auf die jeweils letzte Fassung

zurückgreifen (siehe die CP/M-Kommandos ERA und REN). Schließlich wird die Zwischendatei von „Dn.\$\$\$“ auf „Dn.TXT“ umbenannt. Unter diesem Namen ist damit die bearbeitete Originaldatei gespeichert.

Der Pufferspeicher liegt logisch zwischen der (Original)Textdatei und der Zwischendatei, wie in Fig. 3.2 gezeigt.

Abb. 3.1 Übersicht ED Operation

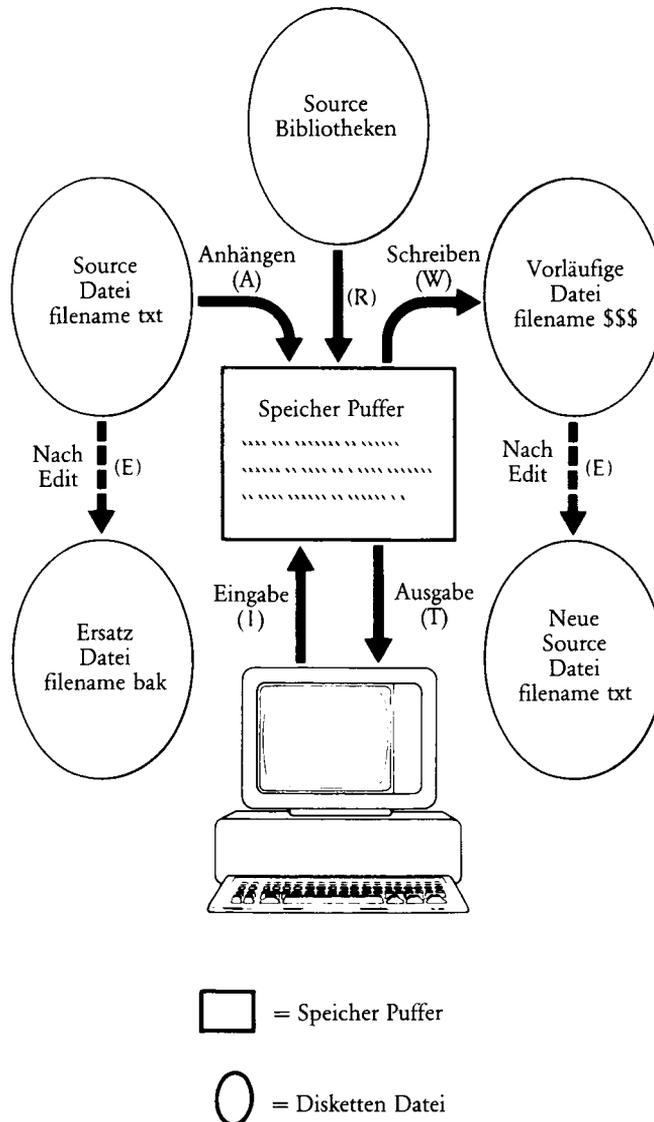
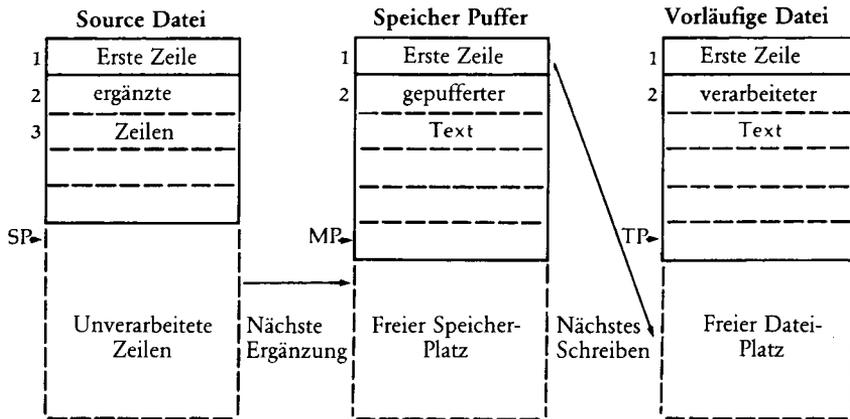


Abb. 3.2 Speicher-Puffer Organisation



SP = Source Zeiger  
 MP = Speicher Zeiger  
 TP = Vorläufiger Zeiger

### 3.1.2 Anweisungen zur Übertragung von Text

In ED gibt es einige Anweisungen, deren Namen aus nur einem Buchstaben bestehen, und die Textzeilen aus der Originaldatei über den Pufferspeicher in die Zwischen-(und evtl. End-)Datei übertragen. Anderen Kommandos kann eine Ganzzahl  $n$  von 0 bis 65535 vorangestellt werden. Die Kommandos werden am Bildschirm in Großbuchstaben angezeigt, können aber in Groß- oder Kleinschreibung eingegeben werden.

- nA Füge die nächsten  $n$  (noch nicht bearbeiteten) Textzeilen aus der Originaldatei ab der Position von SP an das Ende des Pufferspeichers in MP an. Erhöhe SP und MP um  $n!$  Falls auf Großschreibung geschaltet ist (siehe U-Kommando), und das A-Kommando in Großschreibung eingegeben wird, werden alle Eingabezeilen automatisch auf Großschreibung übersetzt.

- nW Schreibe die ersten n Zeilen des Pufferspeichers an das Ende der Zwischendatei. Schiebe den Block der restlichen Zeilen (von Zeilennummer n+1 bis Zeilennummer vor MP) an den Anfang des Pufferspeichers. Erhöhe TP um n!
- E Beende den Editiervorgang. Sowohl der gepufferte Text als auch die nichtbearbeiteten Textzeilen der Originaldatei werden in die Zwischendatei kopiert. Umbenennung erfolgt wie beschrieben.
- H Führe ein E-Kommando aus und setze die Arbeit beim Anfang der neuen Datei fort. Die bisherige Zwischendatei wird damit zur neuen Originaldatei, der Pufferspeicher wird geleert und eine neue Zwischendatei wird angelegt (das gleiche hätte erreicht werden können durch Absetzen eines E-Kommandos, gefolgt von einem Neuaufruf des Editors durch ED Dn.TXT).
- O Der Pufferspeicher wird geleert, die Zwischendatei gelöscht und der Zeiger SP wird auf Position 1 der Originaldatei zurückgesetzt. Die Auswirkungen der vorausgegangenen Editierkommandos werden so außer Kraft gesetzt.
- Q Verlasse ED ohne Dateiänderung.

Es sind einige Sonderfälle zu betrachten. Wird in einem ED-Kommando, bei dem eine ganze Zahl n als Angabe erlaubt ist, diese weggelassen, so wird  $n = 1$  angenommen. So liest das Kommando A aus der Originaldatei in den Puffer, das Kommando W schreibt genau eine (die erste) Zeile des Puffers in die Zwischendatei. Wird ferner das Doppelkreuz (#) an Stelle von n angegeben, so wird die ganze Zahl 65535 (der größte erlaubte Wert für n!) angenommen. Das Kommando # A liest alle noch vorhandenen Zeilen der Originaldatei in den Puffer bis entweder die Datei ausgeschöpft oder der Puffer voll ist. Da die meisten Textdateien üblicher Größe im Pufferspeicher vollständig untergebracht werden können, wird das Kommando # A häufig zu Beginn des Editierens benutzt, um die gesamte Textdatei in den Hauptspeicher zu lesen. In ähnlicher Weise schreibt das Kommando # W den gesamten Pufferinhalt in die Zwischendatei. Zwei Sonderformen der A- und W-Kommandos werden als Zusatzkomfort zur Verfügung gestellt: Das Kommando 0A liest so viele Zeilen aus der Originaldatei in den Pufferspeicher bis dieser mindestens zur

Hälfte gefüllt ist, während 0W so viele Zeilen in die Zwischendatei schreibt, bis der Puffer höchstens noch zur Hälfte gefüllt ist. Eine Fehlermeldung erfolgt, wenn der zur Verfügung stehende Pufferspeicherplatz überschritten wird. Der Benutzer kann dann jedes Kommando (z. B. W) eingeben, das keinen zusätzlichen Speicher benötigt. Der Rest der Zeile, bei deren Übertragung der Überlauf aufgetreten ist, wird bei der nächsten erfolgreichen Durchführung eines A-Kommandos in den Puffer gelesen.

### 3.1.3 Organisation des Pufferspeichers

Da bei den nachstehenden Ausführungen stets vom Inhalt des Pufferspeichers die Rede sein wird und nicht von seiner physikalischen Realisierung, der Inhalt aber ein Text ist, soll künftig die Bezeichnung „Textpuffer“ an Stelle von „Pufferspeicher“ verwendet werden.

Der Textpuffer kann dann als eine Folge von Textzeilen betrachtet werden, die nacheinander über ein A-Kommando aus der Originaldatei eingelesen worden sind. Eine bestimmte Stelle innerhalb des Textpuffers wird dadurch gekennzeichnet, daß man seinen Abstand vom Textanfang festhält. Man führt also einen „Textzeiger“ ein, der immer auf den als nächsten zu bearbeitenden Platz im Pufferspeicher zeigt. Die Bezeichnung CP (engl.: Character Pointer) wird als Abkürzung für Textzeiger verwendet.

Unter der Einwirkung bestimmter Benutzerkommandos wandert der Textzeiger durch den gesamten Textpuffer. Die Fig. 3.3 gibt das logische Erscheinungsbild des Textpuffers wieder. Die Bindestriche (-) sollen die Zeichen einer Textzeile nicht festgelegter Länge darstellen, abgeschlossen sei jede Zeile durch Carriage Return (<cr>) - und Zeilentransport (<lf>) - Zeichen. CP stellt den gedachten Textzeiger dar. Man beachte, daß der CP immer vor dem ersten Zeichen der ersten Zeile, hinter dem letzten Zeichen der letzten Zeile oder zwischen zwei Zeichen positioniert ist. Die „aktuelle“ Zeile CL (engl. current line) ist diejenige Textzeile, die den CP enthält.

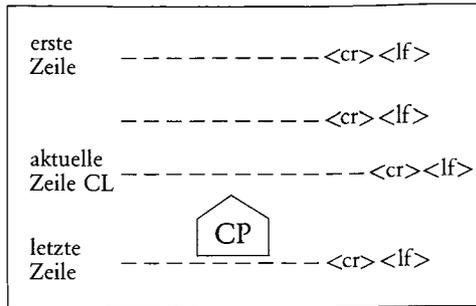


Abb. 3.3

### 3.1.4 Zeilennummern und ED-Anlauf

ED ordnet jeder Textzeile im Puffer eine Zeilennummer zu. Falls gewünscht, wird die fünfstellige Zahl am Beginn jeder Zeile an Ihrer Konsole angezeigt. Die Zeilennummer ist nicht Teil der Datei, d. h. nicht mit dem Text gespeichert. Zeilennummern können dazu benutzt werden, sich auf eine bestimmte Zeile oder eine Gruppe von Zeilen zu beziehen.

Die Anzeige der Zeilennummer erfolgt am Beginn einer jeden Zeile, wenn sich ED im „INSERT-Mode“ befindet (siehe I-Kommando im nächsten Abschnitt), wobei dann jede Zeilennummer die Form

nnnnn:

annimmt. nnnnn ist dabei eine Ganzzahl aus dem Bereich von 1 bis 65535. Ist der Pufferspeicher leer oder die aktuelle Zeile am Ende des Textpuffers, so werden fünf Leerzeichen ausgegeben.

Der Benutzer kann eine bestimmte Zeilennummer dadurch ansprechen, daß er der Befehlszeile eine Zahl voranstellt, gefolgt von einem Doppelpunkt. Die Zahl muß dabei in demselben Format angegeben werden, in dem das System die Zeilennummern ausgibt. Die durch diese Zahl angewählte Zeile muß im bestehenden Textpuffer vorhanden sein. In diesem Fall wird sie zur aktuellen Zeile. So wird z. B. das Kommando

345:T

vom System interpretiert als: Schiebe CP in die Zeile 345 und gib diese Zeile aus! Die Zeilennummern werden automatisch geändert,

falls Zeilen aus dem Text gelöscht oder irgendwo im Text eingefügt werden.

Der Benutzer kann durch Zeilennummern auch einen Abstand (vorwärts oder rückwärts) von der aktuellen Zeile angeben, indem er der Zeilennummer einen Doppelpunkt voranstellt. Das Kommando

:400 T

bedeutet: Schreibe von der aktuellen Zeile bis zur Zeile mit der Zeilennummer 400 alle Zeilen aus.

Die Kombination beider Angaben, also z. B. das Kommando

345::400 T

ist zu interpretieren als: Mache Zeile 345 zur aktuellen Zeile und schreibe alle Zeilen von CL bis zur Zeile mit der Nummer 400.

Zeilenangaben dieser Art können allen ED-Standardkommandos vorangestellt werden.

Die durch ED vorgenommene Zeilennummerierung kann durch das V-(Verify line numbers)Kommando eingeschaltet und durch das V-Kommando ausgeschaltet werden. Beim Aufruf von ED wird die Zeilennummerierung automatisch eingeschaltet.

Falls beim Aufruf von ED die angesprochene Datei noch nicht existiert, gibt ED die Meldung aus

NEW FILE

Der Benutzer muß dann ein „i“-Kommando absetzen, um in den „Eingabezustand“ von ED zu kommen. Nur in diesem Zustand kann er durch zeilenweises Eintippen Text in den Textpuffer schreiben. Die einzelnen Zeilen werden jeweils durch das <cr>-Zeichen abgeschlossen. Ein ctl-Z Zeichen ermöglicht das Verlassen des Eingabezustandes und führt in den Kommandozustand von ED zurück.

### 3.1.5 Operationen im Puffer

Beim Start von ED ist der Puffer leer. Der Benutzer kann entweder Zeilen aus einer bestehenden Textdatei einlesen (A-Kommando) oder Zeilen direkt an der Konsole vermöge des Eingabekommandos

eintippen. ED akzeptiert dann eine beliebige Anzahl von Eingabezeilen, wobei jede Zeile durch ein <cr>-Zeichen abgeschlossen wird. Das Zeichen <lf> wird automatisch bereitgestellt. Die Eingabe wird vom Benutzer durch Eintippen von ctl-z (↑z) beendet. Der Textzeiger CP ist dann hinter dem zuletzt eingegebenen Zeichen positioniert.

Die Eingabefolge

```
I <cr>  
NUN IST ES <cr>  
ZEIT FUER <cr>  
ALLE GUTEN MAENNER <cr>
```

↑ z

bewirkt folgenden Textpuffer

```
NUN IST ES <cr> <lf>  
ZEIT FUER <cr> <lf>  
ALLE GUTEN MAENNER <cr> <lf>
```

ED läßt grundsätzlich bei Einbuchstabenkommandos sowohl Groß- als auch Kleinschreibung zu. Die Wirkung der beiden Schreibweisen ist jedoch unterschiedlich:

Bei Großschreibung des Kommandos werden alle Eingabewerte in Großschreibung umgewandelt, bei Kleinschreibung des Kommandos bleiben Eingabetexte unverändert.

Wird insbesondere das „I“-Kommando eingegeben, dann werden alle Eingabezeilen intern automatisch in Großschreibung umgewandelt. Eingabe des „i“-Kommandos verändert den Eingabetext nicht.

Es gibt in ED verschiedene Kommandos, mit denen die Position des Textzeigers verändert oder Quelltext in der Umgebung des Textzeigers am Bildschirm angezeigt werden kann. Bei den unten gezeigten Befehlen geht dem Befehlsbuchstaben ein n voraus. Dies soll zum Ausdruck bringen, daß an Stelle von n eine positive ganze Zahl (ohne Pluszeichen) erwartet wird. Falls an Stelle von n eine negative ganze Zahl gewünscht wird, muß n mit dem Minuszeichen (–) angegeben werden. Wie oben wird intern das (#)-Zeichen durch 65535 ersetzt. Fehlt eine Angabe für n, so wird n = 1 angenommen.

- ±B Stellt den Textzeiger an den Anfang (+) oder an das Ende (-) des im Pufferspeicher festgehaltenen Textes.
- ±nC Verschiebt den Textzeiger um n Positionen in Richtung auf das Textende (bei positivem n) oder in Richtung Textanfang (bei negativem n). <cr> <lf> werden als zwei Zeichen gezählt.
- ±nD Löschen von n Zeichen ab dem Textzeiger in Richtung auf das Textende (+) oder auf den Textanfang (-).
- ±nK Löschen von n Zeilen ab dem Textzeiger. Falls dieser nicht am Anfang der gerade aktuellen Zeile positioniert ist, werden nur die auf den Textzeiger folgenden (oder bei negativem n die ihm vorangehenden) Zeichen gelöscht.
- ±nL Falls  $n = 0$  ist, wird der Zeichenzeiger an den Anfang der Zeile gestellt, in der er sich gerade befindet. Andernfalls wird der Zeichenzeiger zuerst an den Anfang der Zeile gestellt, in der er sich gerade befindet und dann um n Zeilen in Richtung Textende (+) oder Textanfang (-) verschoben. Der Zeichenzeiger (Textzeiger) kann nicht über das Textende oder den Textanfang hinaus verschoben werden.
- ±nT Für  $n = 0$  wird der Inhalt der Zeile, in der sich der Textzeiger befindet, bis zum letzten Zeichen vor dem Zeiger ausgeschrieben. Für  $n = 1$  wird der Inhalt der aktuellen Zeile ab dem Zeiger bis einschließlich zum Zeilenende ausgeschrieben. Für  $n > 1$  werden n Zeilen vom bzw. bis zum Textzeiger ausgegeben. Durch Betätigen irgendeiner Taste kann die Ausgabe abgebrochen werden.
- ±n Gleichwertig zu ±nLT. Es wird der Zeichenzeiger wie bei ±L verschoben und dann die erreichte Zeile ausgeschrieben.

### 3.1.6 Ketten von Steuerzeichen

Eine beliebige Anzahl von Kommandos kann unmittelbar aufeinanderfolgend eingetippt werden (bis der Konsolpuffer gefüllt ist). Zur Ausführung kommt eine solche Gruppe von Befehlen erst, nachdem

<cr> eingegeben wurde. Damit kann der Benutzer auch die Zeilen-Editiermöglichkeiten der Konsole zum Erstellen einer Befehlszeile heranziehen:

ctl-C	am Anfang einer Zeile eingegeben, wird CP/M neu gestartet.
ctl-E	Physikalisches Ende der Zeile: der Cursor geht an den Anfang der nächsten Zeile, der Befehl wird aber erst ausgeführt, sobald eingetippt wird.
ctl-H	Rücksetzen um eine Zeichenposition
ctl-J	Abschluß der Befehlseingabe und Ausführung, Sprung zur nächsten Zeile
ctl-M	Abschluß der Befehlseingabe und Ausführung, Sprung an Zeilenanfang
ctl-R	Nochmaliges Anzeigen der Befehlszeile: gibt den tatsächlichen Inhalt des Eingabepuffers auf der nächsten Zeile aus
ctl-U	Löschen der gesamten eingegebenen Zeile
ctl-X	Wie ctl-U
ctl-Z	Beendet Konsoleingabe (benötigt in PIP und ED)
rub/del (I←I)	Löschen des letzten Zeichens und Anzeige an der Konsole.

Es werde angenommen, daß der Textpuffer die Zeichen des Beispiels aus dem vorigen Abschnitt enthält, der Zeichenzeiger folge auf das letzte Zeichen im Puffer. Die nachstehend gezeigten Befehlsketten bewirken das rechts stehende Ergebnis. Es wird bei den Einbuchstabenbefehlen die Kleinschreibung benutzt, um eine automatische Umschaltung auf Großschreibung zu vermeiden.

Befehlskette	Wirkung	Entstandener Textpuffer
1. B2T<cr>	Textanzeiger an Pufferanfang stellen und 2 Zeilen ausgeben: 'NUN IST ES ZEIT FUER'	NUN IST ES<cr><lf> ZEIT FUER<cr><lf> ALLE GUTEN MAENNER<cr><lf>
2. 5C0T<cr>	Verschieben CP um 5 Positionen und Ausgeben der Zeichenfolge 'NUN I'	NUN I ST ES<cr><lf>

- |  |  |  |
|--|--|--|
| 3. 2L-T<cr>                                      | Textzeiger um 2 Zeilen nach unten schieben und vorausgehende Zeile ausgeben<br>'ZEIT FUER' | NUN IST ES<cr><lf><br>ZEIT FUER<cr><lf><br>ALLE GUTEN<br>MAENNER<cr><lf> |
| 4. -L#K<cr>                                      | Eine Zeile nach oben gehen und dann alle nachfolgenden Zeilen löschen                      | NUN IST ES<cr><lf>   |
| 5. I <cr><br>ZEIT ZUM<cr><br>EINFUEGEN<cr><br>↑Z | Zwei Textzeilen einfügen mit automatischer Umschaltung auf Großschreibung                  | NUN IST ES<cr><lf><br>ZEIT ZUM<cr><lf><br>EINFUEGEN<cr><lf>              |
| 6. -2L-#T<cr>                                    | Zwei Zeilen nach oben gehen und alle Zeilen vor CP ausgeben<br>'NUN IST ES'                | NUN IST ES<cr><lf><br>ZEIT ZUM<cr><lf><br>EINFUEGEN<cr><lf>              |
| 7. <cr>  | Textzeiger um eine Zeile nach unten schieben und erreichte Zeile ausgeben                  | NUN IST ES<cr><lf><br>ZEIT ZUM<cr><lf><br>EINFUEGEN<cr><lf>              |

### 3.1.7 Suchen und Ändern von Text

In ED gibt es auch ein Kommando, das die Position einer Zeichenkette im Textpuffer feststellt. Dieses Kommando hat die Form

nF s <cr>  
oder  
nF s ↑ Z

wobei s die aufzusuchende Zeichenkette darstellt, gefolgt von entweder einem <cr> oder ctl-Z (↑ z). ED beginnt mit dem Suchen der vorgegebenen Zeichenkette bei der momentanen Position des Textzeigers. Das System macht n Versuche, die Zeichenkette zu finden. Falls das Suchen erfolgreich war, steht der CP unmittelbar hinter dem zuletzt gefundenen Zeichen. Wird s im Pufferspeicher nicht gefunden, so bleibt der Textzeiger an seiner Ausgangsposition. Die zu suchenden Zeichenketten können ctl-L enthalten, was durch das Zeichenpaar <cr> <lf> ersetzt wird.

Die nachstehenden Befehlszeilen zeigen die Verwendung des F-Kommandos:

Befehlskette	Wirkung	Entstandener Textpuffer
1. B#T<cr>	Zeiger an Textanfang setzen und gesamten Text ausgeben	NUN IST ES<cr><lf> ZEIT FUER<cr><lf> ALLE GUTEN MAENNER<cr><lf>
2. FN I<cr>	das Ende der Kette 'N I' wird gesucht	NUN I ST ES<cr><lf>
3. FlzOTT<cr>	Finde das nächste 'I' und gib Zeile aus bis zum CP; dann ist der Rest der aktuellen Zeile auszugeben: 'ZEIT FUER'	NUN IST ES<cr><lf> ZEI T FUER<cr><lf>  ALLE GUTEN MAENNER<cr><lf>

Für die Einfüge-Anweisung (Insert-Kommando) ist auch eine verkürzte Form erlaubt, die häufig in Verbindung mit dem F-Kommando verwendet wird, um einfaches Ersetzen von Text zu ermöglichen. Das „I“-Kommando wird dann so geschrieben:

I s t z  
oder  
I s <cr> ,

wo s der einzufügende Text ist. Wird der einzufügende Text durch ein t z abgeschlossen, so wird der Text unmittelbar hinter dem CP eingefügt und der CP wird nach dem Einfügen unmittelbar hinter dem eingefügten Text positioniert. Dies wird auch erreicht, wenn dem Kommando ein <cr> folgt. Dann wird allerdings automatisch nach dem eingefügten Text noch ein <cr> <lf> eingefügt. Betrachten Sie die nachstehenden Befehlsfolgen als Beispiele für die Verwendung der F- und I-Kommandos:

Befehlskette	Wirkung	Entstandener Textpuffer
1. BISO, t z<cr>	Einfügen von 'SO', am Anfang des Textes	SO, NUN IST ES <cr><lf>  ZEIT FUER <cr><lf> ALLE GUTEN MAENNER<cr><lf>
2. FZEIT t z-4DI DIE GELEGENHEIT t z <cr>	Suchen von 'ZEIT' und löschen; dann ist 'DIE GELEGENHEIT' einzufügen	SO, NUN IST ES <cr><lf> DIE GELEGENHEIT FUER <cr><lf>  ALLE GUTEN MAENNER <cr><lf>



der F-Befehl nicht funktioniert, falls die vorgegebene Zeichenkette im aktuellen Textpuffer nicht gefunden werden kann). Die Wirkung des N-Kommandos ist genau die gleiche wie die des F-Kommandos ausgenommen den Fall, daß die vorgegebene Zeichenkette im Pufferspeicher nicht gefunden werden kann. In diesem Fall wird der gesamte Puffer in die Zwischendatei geschrieben (d. h. ein automatisches # W wird ausgelöst). Dann werden aus der Originaldatei Zeilen in den Puffer gelesen, bis dieser wenigstens halb voll ist oder die gesamte Originaldatei ausgeschöpft ist. Der Suchprozeß wird in dieser Weise fortgesetzt, bis die vorgegebene Zeichenkette n-mal gefunden worden ist, oder die Originaldatei vollständig in die Zwischendatei übertragen ist.

Das letzte Kommando zum Zeileneditieren, das ED zur Verfügung stellt, heißt Juxtaposition oder J-Befehl und lautet:

```
nJ s1↑z s2↑z s3 <cr>  
oder  
nJ s1↑z s2↑z s3↑z
```

Im Textpuffer läuft dabei n-mal folgender Vorgang ab: Der Editor beginnt, ausgehend von CP mit der Suche nach der Zeichenfolge s<sub>1</sub>. Ist diese gefunden, so wird s<sub>2</sub> direkt angeschlossen und der Textzeiger an das Ende von s<sub>2</sub> gesetzt. Dann werden alle Zeichen zwischen der Zeichenfolge s<sub>2</sub> und der Zeichenfolge s<sub>3</sub> (aber nicht diese Zeichenfolgen selbst) gelöscht. Der Textzeiger bleibt unverändert direkt hinter s<sub>2</sub>. Wird die Zeichenfolge s<sub>3</sub> nicht gefunden, so findet kein Löschvorgang statt. Ist z. B. die aktuelle Zeile

```
NUN IST ES <cr><lf>,  
so führt das Kommando  
JT↑zWAS LOS↑z↑l <cr>  
zum Ergebnis  
NUN IST WAS LOS <cr><lf>
```

(Der Benutzer sei daran erinnert, daß ↑l (ctl-L) in Such- und Tauschzeichenfolgen das Zeichenpaar <cr><lf> darstellen.)

Die Anzahl der Zeichen, die ED in den F-, S-, N- und J-Kommandos erlaubt, ist auf 100 Symbole begrenzt.

### 3.1.8 Bibliotheksdateien

ED ermöglicht vermöge des R-Kommandos das Einbinden von Bibliotheks-Quelldateien während des Editiervorgangs. Das R-Kommando ist von der Form

```
RDateiname tz
oder
RDateiname <cr> ,
```

wobei Dateiname der Name einer Quelldatei auf der Platte sein soll. Diese Datei sei vom Typ „LIB“. ED liest aus der angesprochenen Datei Zeichen für Zeichen in den Textpuffer hinter CP bis zum Erkennen eines EOF-Zeichens (tz) in ähnlicher Weise wie das I-Kommando. Setzt der Benutzer also das Kommando

```
RMACRO <cr>
```

ab, so liest ED Zeichen aus der Datei MACRO. LIB bis ein END-OF-FILE-Zeichen angetroffen wird und fügt sie in den Textpuffer ein.

Mit Hilfe des X(Transfer)-Kommandos gestattet ED die Übertragung eines ganzen Textausschnittes. In der Form

```
nX
```

werden n Zeilen, ausgehend von der aktuellen Zeile, in eine Hilfsdatei mit dem festen Namen

```
X$$$$$.LIB
```

übertragen, die nur während der Dauer des Editiervorgangs verwendet wird. Im allgemeinen können Sie durch Verschieben des Textzeigers jede Zeile der Originaldatei zur aktuellen Zeile machen und damit jeden Ausschnitt an die Hilfsdatei übertragen. Die nacheinander übertragenen Zeilen können durch Eingabe von

```
R
```

in den Puffer zurückübertragen werden. In dieser Form ist der einfachste Fall des R-Kommandos wiedergegeben. Sie sollten beachten, daß durch das X-Kommando die übertragenen Zeilen aus dem Textpuffer nicht entfernt werden – dazu müßte ein K-Kommando sich unmittelbar an das X-Kommando anschließen –, und daß das R-

Kommando beim Rückübertragen der Hilfsdatei diese nicht leer macht. Dies bedeutet, daß eine mittels des X-Kommandos übertragene vorgegebene Gruppe von Zeilen beliebig oft in die Originaldatei zurückgebracht werden kann.

Das Kommando 0X löscht die Hilfsdatei.

Zu beachten ist, daß bei normalem Verlassen des Editors durch das Q- oder das E-Kommando die Hilfsdatei gelöscht wird. Wird ED mit `ctl-C` abgebrochen, dann bleibt der Name dieser Bibliotheksdatei im Platteninhaltsverzeichnis zwar erhalten, doch kann zum Dateiinhalt nicht zugegriffen werden, da die Datei nicht ordnungsgemäß geschlossen wurde. Ein Neuaufruf von ED entfernt sie sogar aus dem Inhaltsverzeichnis.

### 3.1.9 Mehrfache Abarbeitung eines Befehls bzw. einer Befehlsfolge

Der Makrobefehl M erlaubt es dem ED-Benutzer mehrere ED-Befehle zu einer Gruppe zusammenzufassen mit dem Ziel der wiederholten Abarbeitung. Das M-Kommando hat die Form

`n M CS <cr>`  
oder  
`n M CS ↑z.`

Dabei stellt CS eine Folge von ED-Kommandos dar. Ein M-Kommando darf in der Folge nicht enthalten sein. Ist  $n > 1$ , dann arbeitet ED diese Kommandofolge n-mal ab. Wenn das Argument n fehlt oder wenn  $n = 0$  oder  $n = 1$  ist, werden die durch M zusammengefaßten Befehle so lange ausgeführt, bis eine Fehlersituation eintritt (z. B. Erreichen des Pufferendes bei einem F-Kommando).

M ist besonders nützlich beim wiederholten Ersetzen von Textteilen, da er eine Kontrolle über die Aktion erlaubt. Im nachstehenden Beispiel soll der angegebene Makrobefehl im gesamten Textpuffer das Wort GAMMA jeweils durch das Wort DELTA ersetzen und zur Kontrolle jede geänderte Zeile ausschreiben.

`MFGAMMA↑z5DIDELTA↑z0TT<cr>`

oder gleichwertig bei Verwendung des S-Kommandos

`MSGAMMA↑zDELTA↑z0TT<cr>`

### 3.2 ED-Fehlermeldungen

Bei Auftreten von Fehlern gibt ED die Meldung aus

„BREAK X AT C“

X stellt dabei einen der nachstehenden Fehlerindikatoren dar:

- ? Befehl nicht erkannt
- > Textpuffer voll  
Abhilfe: Man benutze eines der Kommandos D, K, N, S  
oder W um Zeichen aus dem Puffer zu entfernen;  
oder  
in einem Befehlstext (F-, N- oder S-Kommando) sind zu  
viele Zeichen angegeben.
- # Der Befehl kann nicht in dieser Form angewendet werden  
(z. B. in F-, S- und M-Kommandos)
- O Die durch einen R-Befehl zu lesende Bibliotheksdatei ist  
nicht vorhanden (kann nicht eröffnet werden)

Liegt ein Plattenfehler vor, so gibt CP/M die folgende Meldung aus:

BDOS ERR on d: BAD SECTOR

Der Benutzer hat zwei Möglichkeiten, darauf zu reagieren: Erstens, er kann die Meldung ignorieren, ↵ an der Konsole drücken und seine Arbeit normal fortsetzen. In diesem Falle sollte man allerdings den Pufferinhalt überprüfen, ob alle Daten ordnungsgemäß von der Platte gelesen worden sind. Zweitens kann er das Betriebssystem mit `ctl-C` neu starten und die Sicherungsdatei verwenden, falls eine solche existiert. Dies geschieht dadurch, daß man sich den Inhalt der Sicherungsdatei ausschreiben läßt, um sich von der Richtigkeit des Inhalts zu überzeugen

TYPE x.BAK

wobei x der Name der zu editierenden Datei sein soll. Kann die Sicherung verwendet werden, sollte die Primärdatei gelöscht werden:

ERA x.y

Dann folgt zweckmäßigerweise eine Umbenennung der Sicherungsdatei auf den Namen der Primärdatei

```
REN x.y = x.BAK
```

Die Datei kann dann neu editiert werden mit der zuletzt gültigen Fassung als Grundlage.

ED nimmt auch auf Dateiattribute Rücksicht. Versucht der Benutzer eine schreibgeschützte Datei zu editieren, wird an der Konsole die Meldung

```
**FILE IS READ/ONLY**
```

ausgegeben. Die Datei kann dann geladen und zum Zweck der Prüfung ausgeschrieben aber nicht geändert werden. Im Normalfall wird der Benutzer dann den Editiervorgang abbrechen und mit Hilfe von STAT das Dateiaattribut in R/W abändern. Hat die zu editierende Datei das Attribut „system“ (Systemdatei), so wird der Editiervorgang mit der Konsolmeldung

```
'SYSTEM' FILE NOT ACCESSIBLE
```

abgebrochen. Falls gewünscht, kann wieder das STAT-Programm verwendet werden, um das Dateiaattribut zu ändern.

### 3.3 Steuerzeichen und Befehle

Die nachstehende Tabelle faßt die in ED verfügbaren Steuerzeichen und Befehle zusammen:

<u>Steuerzeichen</u>	<u>Funktion</u>
ctl-C	System-Neustart
ctl-E	Physikalisch <cr> <lf> (kein Abschicken des Befehls)
ctl-H	Rücksetzen um ein Zeichen
ctl-J	Logischer Tabulator (Spalten 1, 9, 16, . . .)
ctl-L	Logisches <cr> <lf> beim Suchen und Ersetzen von Textteilen

ctl-R	Wiederholen der Zeilen-Anzeige
ctl-U	Löschen der gesamten Zeile
ctl-X	Löschen der gesamten Zeile
ctl-Z	Zeichenkettenbegrenzer
b/del	Lösche zuletzt eingegebenes Zeichen
<u>Befehl</u>	<u>Funktion</u>
nA	Füge Zeilen an
±B	Setze Textzeiger auf Anfang oder Ende des Puffers
±nC	Verschiebe den Textzeiger um n Positionen
±nD	Lösche n Zeichen
E	Beende das Editieren und schließe die Dateien (normales Ende)
nF	Finde Zeichenkette
H	Beende Editiervorgang, schließe und öffne Dateien neu
I	Füge Zeichen in die zu editierende Datei ein; man schreibe „i“ falls Groß- und Kleinbuchstaben eingegeben werden sollen.
nJ	Füge Text an bestimmte Stelle ein, ohne bestehenden Text zu verändern
±nK	Lösche n Zeilen
±nL	Bewege Textzeiger um n Zeichen nach unten/oben
nM	Makrodefinition
nN	Suche das n-te Auftreten des Textes

O	Kehre zur Originaldatei zurück
$\pm nP$	Verschiebe Textzeiger um n Textblöcke von einer Bildschirmlänge und schreibe den Textblock aus
Q	Verlasse ED ohne Dateiänderung
R	Lies Bibliotheksdatei in Textdatei ein
nS	Ersetze Textteil
$\pm U$	Schalte auf Großschreibung um, falls U eingegeben wird; schalte nicht um, falls -U eingegeben wird
$\pm V$	Schalte die Angabe von Zeilennummern ein (+) oder aus (-)
OV	Sonderfall des V-Befehls, bedeutet: Drucke Statistik über Pufferspeicher in der Form freier Bereich/Gesamtgröße, wobei der freie Bereich die Anzahl der noch verfügbaren Bytes im Pufferspeicher ausdrückt (dezimal) und Gesamtgröße den gesamten vorhandenen Umfang des Pufferspeichers angibt.
nW	Schreibe n Zeilen (in Zwischendatei)
nZ	Warte etwa n Sekunden
$\pm n$	Verschiebe Textzeiger um n Zeilen und schreibe die erreichte Zeile aus (gleichwertig mit $\pm nLT$ )

Um Datenverluste durch Schreibfehler zu verhindern, fordert ED, daß gewisse Befehle durch einzelne Buchstaben angegeben werden. So müssen die Befehle

E(end), H(head), O(original), Q(quit)

als Einbuchstabenbefehle eingegeben werden.

Die Kommandos I, J, M, N, R und S sollten als i, j, m, n, r und s geschrieben werden, falls sowohl Groß- als auch Kleinbuchstaben Verwendung finden sollen. Andernfalls werden alle Zeichen auf Großschreibung umgesetzt.

Wird ein Kommando in Großschreibung eingegeben, übersetzt ED die gesamte zugehörige Zeichenfolge in eine Großbuchstabenfolge.

1

2

3

# ASSEMBLER

## 4.1 Einführung

Der CP/M-Assembler liest Quelldateien von der Platte, die Programme in mnemonischer Form enthalten, und erzeugt 8080 Maschinensprache im Hexadezimalformat. Der CP/M-Assembler wird aufgerufen durch Eingabe von

ASM Dateiname  
oder  
ASM Dateiname.parms

In beiden Fällen nimmt der Assembler an, daß eine Datei auf der Platte existiert mit dem Namen

Dateiname.ASM

die eine Quelldatei in 8080-Assemblersprache darstellt. Die beiden Formen des Assembleraufrufs unterscheiden sich nur darin, daß die zweite Form eine Übergabe von Parametern an den Assembler erlaubt, mit dem Ziel, den Zugriff auf die Quelldatei sowie die Bestimmung der Hexadezimal- und Druckdatei zu steuern.

In jedem Fall druckt der CP/M Assembler die Meldung

CP/M Assembler Ver n.n

aus, wo n.n die laufende Versionsnummer ist. In der ersten Form des Assemblerkommandos liest der Assembler die Quelldatei mit dem unterstellten Datei-Typ ASM ein und erzeugt die beiden Ausgabedateien

Dateiname.HEX  
und  
Dateiname.PRN

Die HEX-Datei enthält den dem Originalprogramm entsprechenden Maschinencode im Intel-Hexadezimalformat, und die PRN-Datei enthält dazu noch eine Liste, welche den generierten Maschinen-

code, gegebenenfalls Fehleranzeigen sowie die Quellzeilen zeigt. Treten während der Übersetzung Fehler zutage, so gehen sie in die PRN-Datei ein und werden an der Konsole angezeigt.

Die Form `ASM Dateiname.p1p2p3` kann dazu benutzt werden, für die Eingabe- und Ausgabedateien gezielte Namen anzugeben. In diesem Fall besteht der Parameterteil des Kommandos aus einer Gruppe von drei Buchstaben, die die Herkunft der Quelldatei, die Bestimmung der HEX-Datei und der Druckdatei spezifizieren. Diese Form lautet

`Dateiname.p1p2p3`

wo p1, p2 und p3 für einzelne Buchstaben stehen.

p1: A,B,..., P bezeichnet den Namen der Platte, die die Quelldatei enthält

p2: A,B,..., P bezeichnet den Namen der Platte, welche die HEX-Datei aufnehmen soll

Z Generierung der HEX-Datei wird unterdrückt

p3: A,B,..., P bezeichnet den Namen der Platte, die die Druckdatei aufnehmen soll

X Listing wird an der Konsole ausgegeben

Z Generierung der Druckdatei wird unterdrückt

Das Kommando

`ASM X.AAA`

besagt somit, daß die Quelldatei (`X.ASM`) von der Platte A geholt werden soll, und daß die HEX (`X.HEX`)- und Druck (`X.PRN`)-Datei ebenfalls auf der Platte A angelegt werden sollen. Diese Form des Kommandos wird automatisch unterstellt, falls der Assembler von Platte A aus läuft. Das heißt, arbeiten Sie augenblicklich mit Platte A, so ist das obige Kommando gleichwertig mit

`ASM X`

Das Kommando

### ASM X.ABX

zeigt an, daß die Quelldatei von Platte A geholt werden, die Hexdatei auf Platte B abgelegt und das Listing an der Konsole ausgegeben werden soll. Das Kommando

### ASM X.BZZ

holt die Quelldatei von Platte B und unterdrückt die Generierung der Hex- und Druckdatei (dieses Kommando ist für Testzwecke nützlich).

Das Format des Quellprogramms ist kompatibel mit dem Intel 8080 Assembler (Makros sind in ASM nicht implementiert; siehe auf Wunsch verfügbarer Makroassembler MAC). Einige Erweiterungen im CP/M-Assembler erleichtern seine Benutzung etwas. Diese Erweiterungen sind im folgenden beschrieben.

## 4.2 Programm-Format

Ein in der Assemblersprache geschriebenes Programm, das als Eingabe für den Assembler akzeptabel sein soll, besteht aus einer Folge von Anweisungen der Form

Zeilennummer Marke Operation Operand; Kommentar

wo im Einzelfall einige oder alle der angegebenen Felder vorhanden sein können. Jede Anweisung der Assemblersprache wird abgeschlossen durch ein cr-lf-Zeichenpaar (lf wird automatisch vom ED-Programm hinzugefügt), oder durch das ! Zeichen, das vom Assembler als Zeilenendezeichen interpretiert wird. Auf diese Weise können mehrere Anweisungen des in Assemblersprache geschriebenen Programms in eine physikalische Zeile geschrieben werden, falls sie durch ein Ausrufezeichen voneinander getrennt sind.

Die Zeilennummer ist optional. Sie wird als ganzzahlige Dezimalzahl angegeben und stellt die Zeilennummer des Quellprogramms dar. Sie wird - falls angegeben - vom Assembler überlesen.

Das Markenfeld hat die Form

Bezeichner

oder

Bezeichner:

und ist ebenfalls optional außer bei einigen besonderen Typen von Anweisungen. Bezeichner ist eine Folge alphanumerischer Zeichen, wobei das erste Zeichen ein Buchstabe sein muß. Bezeichner dürfen vom Programmierer weitgehend freizügig verwendet werden, um Programmschritte und Assembler-Direktiven kennzeichnen zu können, sie dürfen eine Länge von 16 Zeichen indes nicht überschreiten. Innerhalb eines Bezeichners sind alle Zeichen signifikant ausgenommen ein eingebettetes Dollarzeichen (\$), welches dazu benutzt werden kann, die Lesbarkeit des Namens zu verbessern. Weiterhin werden alle Kleinbuchstaben behandelt als wären sie Großbuchstaben. Nachstehend sind einige zulässige Beispiele von Marken aufgeführt:

```
x      xy      langer$name
x:     Yxl:    weiter$bezeichnete$daten:
X1Y2   X1x2   x234$5678$9012$3456:
```

Das Operationsfeld enthält entweder eine Assemblerdirektive (Pseudo-Operation) oder einen 8080-Operationscode. Die Pseudo-Operationen und Maschinenoperationscodes sind unten beschrieben.

Das Kommentarfeld enthält beliebige Zeichen, die auf das ;-Symbol folgen bis zum nächsten physikalischen oder logischen EOL-Zeichen. Diese Zeichen werden gelesen, aufgelistet und ansonsten vom Assembler ignoriert. Der CP/M-Assembler behandelt im übrigen Anweisungen, die mit einem \* in Spalte 1 beginnen als Kommentarzeilen, die zwar aufgelistet, aber ansonsten während des Assemblierprozesses keine Berücksichtigung finden.

Das Assembler-Programm wird als eine Folge von Anweisungen der obigen Form geschrieben, eine abschließende END-Anweisung ist optional. Alle auf die END-Anweisung folgenden Anweisungen werden nicht zur Kenntnis genommen.

### 4.3 Angabe des Operanden

Um die Operationscodes und Pseudo-Operationen vollständig beschreiben zu können, ist es notwendig, zuerst die Gestaltung des Operandenfeldes zu beschreiben, da man sie in nahezu allen Anweisungen braucht. Ausdrücke im Operandenfeld bestehen aus einfachen Operanden (Marken, Konstanten und reservierten Wörtern), die in geeignet gebildeten Unterausdrücken mittels arithmetischer

und logischer Operatoren miteinander verknüpft sind. Die Ausdrücke werden vom Assembler im Zuge der Assemblierung ausgewertet. Jeder Ausdruck muß einen 16-Bit-Wert ergeben. Weiterhin muß die Anzahl der signifikanten Stellen im Ergebnis der beabsichtigten Verwendung angepaßt sein. Das heißt, falls z. B. ein Ausdruck in einer MVI-Instruktion verwendet wird, daß die 8 höherwertigen Bits verschwinden müssen. Diese Einschränkungen bezüglich der Signifikanz werden bei den einzelnen Instruktionen angegeben.

### 4.3.1 Marken

Wie oben erwähnt, ist eine Marke ein Bezeichner, der bei einer bestimmten Anweisung vorhanden sein kann. Im allgemeinen erhält eine Marke ihren Wert durch den Typ der Anweisung, der sie vorausgeht. Erscheint die Marke bei einer Anweisung, die Maschinencode generiert oder Speicherplatz reserviert (z. B. einer MOV-Instruktion oder einer DS Pseudo-Operation), erhält die Marke den Wert der Programmadresse, die sie bezeichnet. Geht die Marke einem EQU oder SET voraus, erhält sie den Wert, der das Ergebnis der Auswertung des Operandenfeldes ist. Ausgenommen die SET-Anweisung, kann ein Bezeichner nur für die Marke einer Anweisung verwendet werden.

Tritt der Name einer Marke in einem Operandenfeld auf, so wird er vom Assembler durch seinen Wert ersetzt. Dieser Wert kann dann mit anderen Operanden oder Operatoren zur Gestaltung des Operandenfeldes einer bestimmten Instruktion kombiniert werden.

### 4.3.2 Numerische Konstante

Eine Numerische Konstante ist ein 16-Bit-Wert in einer von mehreren Darstellungsformen. Die Basis des Zahlensystems, genannt im folgenden „Radix“ der Konstanten, wird durch einen angehängten Radixanzeiger kenntlich gemacht. Die verwendeten Radix-Anzeiger sind:

- B binäre Konstante (Basis 2)
- O oktale Konstante (Basis 8)
- Q oktale Konstante (Basis 8)
- D dezimale Konstante (Basis 10)

## H hexadezimale Konstante (Basis 16)

Q ist als Radixindikator für Oktalzahlen alternativ verwendbar, da der Buchstabe O leicht mit der Ziffer 0 verwechselt werden kann. Jede numerische Konstante, die nicht mit einem Radixindikator endet, wird als dezimale Konstante betrachtet.

Eine Konstante ist somit zusammengesetzt als Folge von Ziffern, gefolgt von einem optionalen Radixindikator, wobei die Ziffern in dem durch den Radix festgelegten Bereich liegen. Binärkonstante müssen somit aus den Binärziffern 0 und 1 zusammengesetzt sein, Oktalkonstante können die Ziffern 0 - 7 enthalten, während Dezimalkonstanten die Zehnerziffern enthalten.

Hexadezimalkonstante enthalten sowohl Dezimalziffern als auch Hexadezimalziffern A (10D), B (11D), C (12D), D (13D), E (14D) und F (15D). Man sollte beachten, daß die führende Ziffer einer Hexadezimalkonstanten eine Dezimalziffer sein muß, um die Verwechslung einer Hexadezimalkonstanten mit einem Bezeichner zu vermeiden. Eine führende Null genügt hierzu.

Eine in der genannten Weise zusammengesetzte Konstante muß bei der Umwandlung in eine Binärzahl mit 16 Bits dargestellt werden können, andernfalls wird die entstehende Binärzahl vom Assembler rechts abgeschnitten. Ähnlich wie bei Bezeichnern sind innerhalb von Konstanten eingebettete Dollarzeichen erlaubt, um die Lesbarkeit zu verbessern. Schließlich wird der Radixindikator im Falle einer Kleinschreibung durch den Benutzer vom Assembler in Großschreibung umgewandelt. Nachstehend einige Beispiele für gültige numerische Konstante

1234 1234D 1100B 1111\$0000\$1111\$0000B

1234H OFFEH 33770 33\$77\$22Q

33770 Of3h 1234d Offfh

### 4.3.3 Reservierte Wörter

Mehrere reservierte Zeichenfolgen haben eine vordefinierte Bedeutung im Operandenfeld einer Anweisung. Die Namen der 8080 Register sind unten aufgeführt. Sie ergeben bei der Auswertung den rechts stehenden Wert.

A	7
B	0
C	1
D	2
E	3
H	4
L	5
M	6
SP	6
PSW	6

(Auch hier haben Namen in Kleinschreibung den gleichen Wert wie bei Großschreibung). Maschineninstruktionen können ebenfalls im Operandenfeld benutzt und dann entsprechend ihrem internen Code ausgewertet werden. Im Falle von Instruktionen, die Operanden fordern und bei denen die spezifischen Operanden Teil des Bitmusters der Instruktion werden (z. B. MOV A,B), errechnet sich der Wert der Instruktion (in diesem Fall der MOV-Instruktion) aus dem Bitmuster der Instruktion mit Nullen in den optionalen Feldern (z. B. MOV wird zu 40H).

Tritt das Symbol \$ im Operandenfeld auf (nicht eingebettet in Bezeichner und numerische Konstante), wird sein Wert gleich der Adresse der nächsten zu generierenden Instruktion (nicht der Instruktion innerhalb der laufenden logischen Zeile).

#### 4.3.4 Zeichenfolgen als Konstante

Zeichenfolgen-Konstante stellen Folgen von ASCII-Zeichen dar und werden ausgedrückt durch Einschließen zwischen Hochkommas ('). Alle Zeichenfolgen müssen vollständig innerhalb der laufenden physikalischen Zeile enthalten sein (damit darf das Ausrufezeichen in der Zeichenkette enthalten sein) und dürfen eine Länge von 64 Zeichen nicht überschreiten. Das Hochkomma selbst kann in der Zei-

chenkette enthalten sein, muß dann aber als Apostroph (‘) dargestellt werden. Der Assembler macht daraus wieder ein normales Hochkomma. In den meisten Fällen ist die Länge der Zeichenfolge beschränkt auf ein oder zwei Zeichen (die DB-Pseudooperation ist eine Ausnahme), wodurch die Zeichenfolge bei der Umwandlung zu einer 8- bzw. 16-Bit-Zahl wird. Zweizeichen-Ketten werden zu einer 16-Bit-Konstanten mit der zweiten Zahl als niederwertigem Byte und der ersten Zahl als höherwertigem Byte.

Der Wert eines Zeichens ist der zugeordnete ASCII-Code. Innerhalb von Zeichenketten gibt es keine Umschaltung von Klein- auf Großschreibung, und damit können sowohl Zeichen in Groß- als auch in Kleinschreibung dargestellt werden. Es ist jedoch zu beachten, daß nur druckbare ASCII-Zeichen innerhalb von Zeichenketten verwendet werden dürfen.

Zulässige Zeichenketten	Es wird dargestellt
'A' 'B' 'ab' 'c'	A B ab c
'Walla Walla Wash.'	Walla Walla Wash.
'Sie sagte "Hallo" zu mir.'	Sie sagte 'Hallo' zu mir.
'Ich sagte "Hallo" zu ihr.'	Ich sagte 'Hallo' zu ihr.

#### 4.3.5 Arithmetische und logische Operatoren

Die oben beschriebenen Operanden können in normaler algebraischer Schreibweise miteinander verknüpft werden durch Verwendung irgendeiner Kombination geeignet gebildeter Operanden, Operatoren oder Klammersausdrücke. Im Operandenfeld werden folgende Operatoren als solche erkannt:

$a + b$	vorzeichenlose algebr. Summe aus a und b
$a - b$	vorzeichenlose algebr. Differenz zwischen a und b
+b	unäres Plus (b erzeugt)
-b	unäres Minus (identisch $0 - b$ )
$a * b$	vorzeichenlose Betragsmultiplikation von a und b
$a/b$	vorzeichenlose Betragsdivision von a durch b

a Mod b	Rest bei der Division a/b
NOT b	logische Inverse von b (alle 0 werden 1, 1 werden 0), wobei b als 16-Bit-Wert betrachtet wird.
a AND b	bitweises logisches „UND“ von a und b
a OR b	bitweises logisches „ODER“ von a und b
a XOR b	bitweises logisches „EXKLUSIV ODER“ von a und b
a SHL b	Wert, der bei Linksschiebung um b Bits entsteht mit Nullenauffüllung
a SHR b	Wert, der bei Rechtsschiebung um b Bits mit Nullenauffüllung entsteht.

In allen diesen Fällen sollen a und b einfache Operanden (Marken, numerische Konstante, reservierte Wörter und Zeichenketten von ein oder zwei Zeichen Länge) darstellen oder in geklammerten Unterausdrücken auftreten wie

10+20 10h+37Q LI/3 (L2+4) SHR 3

('a' and 5fh) + '0' ('B'+B) OR (PSW+M)

(1+(2+c)) shr (A-(B+1))

Man beachte, daß alle Berechnungen zur Assemblierzeit als 16-Bit-Berechnungen ohne Vorzeichenbeachtung durchgeführt werden. So wird z. B. -1 als 0 - 1 berechnet, was als Ergebnis 0ffffh (d. h. nur aus Ziffern 1 bestehend) liefert. Das Ergebnis muß dem Operationscode, in dem es verwendet wird, entsprechen. Wird der Ausdruck z. B. in einer ADI (addiere unmittelbar)-Instruktion verwendet, so müssen die Bits des höherwertigen Bytes verschwinden. Daraus folgt, daß die Operation ADI-1 eine Fehlermeldung verursacht (-1 wird 0ffffh, was nicht als 8-Bit-Wert dargestellt werden kann), während ADI(-1)AND 0FFH vom Assembler akzeptiert wird, weil die AND-Operation die Bits des höherwertigen Bytes zu Null macht.



$(a \text{ OR } b) \text{ AND } (\text{NOT } c) + d \text{ SHL } e$

wäre äquivalent dem geklammerten Ausdruck

$(a \text{ OR } b) \text{ AND } ((\text{NOT } c) + (d \text{ SHL } e))$

Ein nichtgeklammerter Ausdruck ist nur dann richtig aufgebaut, wenn der Ausdruck, der sich beim Einsetzen der wirklichen Klammern an die Stelle der gedachten ergibt, richtig aufgebaut ist.

#### 4.4 Assembler-Direktiven

Assembler-Direktiven dienen dazu, Marken während des Assemblier-Vorgangs Werte zuzuweisen, bedingtes Assemblieren zu ermöglichen, Speicherbereiche zu definieren und die Startadressen im Programm festzulegen. Jede dieser Direktiven wird als Pseudo-Operation geschrieben, die im Operationsfeld einer Zeile erscheint. Die vom Assembler akzeptierten Pseudo-Operationen sind

ORG	Programm-Anfangsadresse angeben
END	Arbeit am Programmtext wird abgebrochen
EQU	numerisches Gleichsetzen
SET	numerisches „Setzen auf Wert“
IF	Beginn des bedingten Assemblierens
ENDIF	Ende des bedingten Assemblierens
DB	Definieren Daten-Bytes
DW	Definieren Daten-Doppelbytes
DS	Definieren Speicherbereich

Die einzelnen Direktiven werden nachstehend genauer beschrieben.

#### 4.4.1 Die ORG-Direktive

Die ORG-Direktive hat die Form

Marke ORG Ausdruck

wobei Marke ein optionaler Programmbezeichner und Ausdruck ein 16-Bit Ausdruck ist, bestehend aus Operanden, die vor der ORG-Direktive definiert wurden. Der Assembler beginnt mit der Generierung des Maschinencodes an der Speicherstelle, die durch den Ausdruck spezifiziert wird. Es kann in einem bestimmten Programm eine beliebige Anzahl von ORG-Direktiven geben, und es wird nicht geprüft, ob der Programmierer sich überlappende Speicherbereiche definiert. Der Benutzer sollte ferner beachten, daß die meisten für das CPM-System geschriebenen Programme mit einer ORG-Direktive der Form

ORG 100H

beginnen, was bewirkt, daß die Maschinencodegenerierung an der Basis des transienten Programmereichs (TPA) von CPM beginnt. Wird in der ORG-Direktive eine Marke spezifiziert, so erhält sie den Wert des Ausdrucks. (Diese Marke kann dann im Operandenfeld anderer Anweisungen dazu benutzt werden, den Ausdruck zu repräsentieren.)

#### 4.4.2 Die END-Direktive

Die End-Direktive ist in einem in Assemblersprache geschriebenen Programm optional, aber wenn sie geschrieben wird, dann muß sie die letzte Anweisung sein (alle evtl. nachfolgenden Anweisungen werden beim Assemblieren nicht berücksichtigt). Die zwei Formen der END-Direktive sind

Marke END

Marke END Ausdruck

wo die Marke wieder optional ist. Wird die erste Form benutzt, so hält der Assembliervorgang an, und der Wert für die Startadresse des Programms wird als 0000 angenommen.

Andernfalls wird der Ausdruck ausgewertet und wird zur Startadresse des Programms (diese Startadresse ist im letzten Satz der im

Intel Hex-Format dargestellten Maschinencodeteil, die Ergebnis des Assemblierprozesses ist, enthalten). Daher enden die meisten CP/M-Assemblerprogramme mit der Anweisung

```
END 100H
```

um die Startadresse 100H (Beginn der TPA) zu bestimmen..

#### 4.4.3 Die EQU-Direktive

Die EQU-Direktive wird dazu benutzt, Synonyme für bestimmte numerische Werte einzuführen. Die Form dieser Anweisung lautet

```
Marke EQU Ausdruck
```

wobei die Marke angegeben werden muß und nicht für eine weitere Anweisung verwendet werden darf. Der Assembler wertet den Ausdruck aus und weist den Wert des Ergebnisses dem Bezeichner im Markenfeld zu. Dieser Bezeichner ist gewöhnlich ein Name, der den Wert in mehr auf einen menschlichen Leser hin orientierten Weise beschreibt. Weiter dient der Name dazu, das ganze Programm hindurch gewisse Funktionen zu parametrisieren. Man nehme an, die von einem Teletype empfangenen Daten erscheinen an einem bestimmten Eingabe-Port, und abgeschickt werden Daten an den Teletype über das nächste Ausgabe-Port. Die nachstehende Folge von EQU-Direktiven könnte dazu benutzt werden, diese Ports für eine vorgegebene Hardware-Umgebung zu definieren.

```
TTYBASE EQU 10H      ; Basis-Portadresse für TTY
TTYIN   EQU TTYBASE  ; TTY Daten ein
TTYOUT  EQU TTYBASE + 1 ; TTY Daten aus
```

An einer späteren Stelle im Programm könnten die Anweisungen, die zum Teletype zugreifen, so aussehen:

```
IN      TTYIN        ; Lies TTY-Daten nach Reg.-A
OUT     TTYOUT       ; Gib Daten aus zum Teletype aus Reg.-A
```

So ist das Programm leichter lesbar, als wenn nur die absoluten Adressen der I/O-Ports benutzt worden wären. Wenn darüber hinaus die Hardware-Umgebung neu definiert werden muß, so daß das

Teletype-Kommunikationsport bei 7FH an Stelle von 10H liegt, muß nur die erste Direktive geändert werden in

```
TTYBASE EQU 7FH ; Basis-Portadresse für TTY
```

und das Programm kann ohne Änderung der anderen Anweisungen neu assembliert werden.

#### 4.4.4 Die SET-Direktive

Die SET-Direktive ist ähnlich der EQU-Anweisung und hat die Form

```
Marke SET Ausdruck
```

Die Marke kann noch an anderen SET-Anweisungen innerhalb des Programms vorkommen. Der Ausdruck wird ausgewertet und wird zum aktuellen Wert, der der Marke zugewiesen wird. Damit definiert EQU eine Marke durch einen einzigen Wert, während SET einen Wert definiert, der Gültigkeit hat von der momentanen SET-Direktive bis zu der Stelle, wo die Marke an der nächsten SET-Direktive erscheint. Die Benutzung der SET-Direktive erfolgt ähnlich, wie die der EQU-Direktive, aber sie erfolgt meist zur Steuerung des bedingten Assemblierens.

#### 4.4.5 Die IF- und ENDIF-Direktiven

Die IF- und ENDIF-Direktiven definieren einen Bereich von Assembleranweisungen, die während des Assembliervorgangs eingeschlossen oder ausgeschlossen sein sollen. Ihre äußere Form lautet

```
IF Ausdruck  
Anweisung # 1  
Anweisung # 2  
...  
Anweisung # n  
ENDIF
```

Beim Antreffen der IF-Direktive wertet der Assembler den dem IF folgenden Ausdruck aus (alle Operanden im Ausdruck müssen vor der IF-Anweisung definiert sein). Falls der Ausdruck einen von Null verschiedenen Wert liefert, dann werden die Anweisungen von Anweisung # 1 bis Anweisung # n assembliert; ergibt der Ausdruck

den Wert Null, so werden die Anweisungen aufgelistet, aber nicht assembliert. Bedingtes Assemblieren wird oft dazu benutzt, ein sehr allgemeines Programm zu schreiben, das eine Anzahl möglicher Laufzeitumgebungen beinhaltet, wobei aber nur wenige Teile des Programms für einen speziellen Assemblerlauf ausgewählt werden. Der folgende Programmausschnitt z. B. könnte Teil eines Programms sein, das entweder mit einem Teletype oder einer Bildschirmkonsole (aber nicht mit beiden) kommuniziert und zwar durch Auswahl eines speziellen Wertes für TTY vor Beginn der Assemblierung.

```

TRUE     EQU     0FFFFH    ;DEFINE VALUE OF TRUE
FALSE    EQU     NOT TRUE  ;DEFINE VALUE OF FALSE
;
TTY      EQU     TRUE      ;TRUE IF TTY, FALSE IF CRT
;
TTYBASE  EQU     10H       ;BASE OF TTY I/O PORTS
CRTBASE  EQU     20H       ;BASE OF CRT I/O PORTS
IF       TTY             ;ASSEMBLE RELATIVE TO
                        ;TTYBASE
CONIN    EQU     TTYBASE   ;CONSOLE INPUT
CONOUT   EQU     TTYBASE+1 ;CONSOLE OUTPUT
ENDIF
;
IF       NOT TTY        ;ASSEMBLE RELATIVE TO
                        ;CRTBASE
CONIN    EQU     CRTBASE   ;CONSOLE INPUT
CONOUT   EQU     CRTBASE+1 ;CONSOLE OUTPUT
ENDIF
...
IN       CONIN          ;READ CONSOLE DATA
...
OUT     CONOUT          ;WRITE CONSOLE DATA

```

In diesem Fall würde das Programm für eine Umgebung mit einem angeschlossenen Teletype assembliert, die INPUT-Port-Adresse sei 10H. Die Definition von TTY könnte geändert werden in

```

TTY      EQU     FALSE

```

und in diesem Fall würde das Programm für einen Bildschirmanschluß mit der Basis-Port-Adresse 20H assemblieren.

#### 4.4.6 Die DB-Direktive

Die DB-Direktive ermöglicht es dem Programmierer, einen bestimmten Speicherbereich zu reservieren und in einfacher (ein Byte-) Genauigkeit zu initialisieren. Die Form dieser Direktive ist

Marke: DB e#1, e#2, ..., e#n,

wobei e#1 bis e#n entweder arithmetische Ausdrücke darstellen, deren Auswertung Zahlenwerte zwischen 0 und 255 ergeben oder ASCII Zeichenketten, deren Länge 64 Zeichen nicht übersteigen. Die Anzahl der Ausdrücke in einer Direktive ist praktisch nicht beschränkt. Die Ausdrücke werden ausgewertet und die Ergebnisse sequentiell in die Maschinencoddatei abgelegt, unmittelbar auf die letzte vom Assembler generierte Programmadresse folgend. Textzeichen werden im Speicher in ähnlicher Weise abgelegt; mit dem ersten Zeichen wird begonnen, mit dem letzten geendet. In komplizierteren Ausdrücken dürfen nur Zeichenfolgen mit bis zu zwei Zeichen als Operanden verwendet werden. Es ist zu beachten, daß ASCII-Zeichen im Speicher immer mit ausgeschaltetem Parity-Bit (Bit 7) abgelegt werden. Außerdem wird der Text in einer Zeichenkette nicht in Großschreibung übersetzt, sondern wie angegeben übernommen. Falls die Marke verwendet wird, kann durch sie im gesamten übrigen Programm auf den so definierten Speicherbereich zugegriffen werden.

Nachstehend einige Beispiele gültiger DB-Direktiven:

data: DB 0,1,2,3,4,5  
DB data and Offh,5,377Q,1+2+3+4

schalt: DB 'Bitte Namen eingeben',cr,lf,0  
DB 'AB' SHR 8, 'C', 'DE' AND 7FH

#### 4.4.7 Die DW-Direktive

Die DW-Direktive ist der DB-Direktive sehr ähnlich. Der wesentliche Unterschied liegt darin, daß der Wert von Zwei-Byte(= 16 Bit-) Worten festgelegt wird. Die DW-Vereinbarung hat die Form

Marke: DW e#1, e#2, ..., e#n,

wo e#1 bis e#n Ausdrücke darstellen, deren Auswertung 16 Bit-Ergebnisse liefern. Es ist zu beachten, daß in der Werteliste zwar Zeichenketten auftreten, diese aber zwei Zeichen nicht überschreiten dürfen. In jedem Fall ist die Datenspeicherung auf den 8080 Prozessor zugeschnitten: zuerst wird das niederwertige Byte gespeichert und dann das höherwertige Byte eines Ausdrucks.

Beispiele:

```
doppel: DW Offefh,doppel+4,schalt-$,255+255
        DW 'a', 5, 'a', 'CD', 6 shl 8 or 11b.
```

#### 4.4.8 Die DS-Direktive

Die DS-Direktive dient dem Zweck, einen Speicherbereich ohne Wertfestlegung zu reservieren. Die Form lautet

```
Marke:   DS   Ausdruck,
```

die Marke ist auch hier nicht vorgeschrieben. Der Assembler beginnt mit der nachfolgenden Code-Generierung hinter dem von DS reservierten Speicherbereich. Damit hat diese DS-Definition genau die gleiche Wirkung wie die Befehlsfolge:

```
Marke: EQU $ ;Festlegung der Anfangsadresse
        ORG $ + Ausdruck ;Überspringen des reservierten
        Bereichs
```

#### 4.5 Operations-Codes

Erinnern wir uns an die gebräuchliche Form einer Assembler-Instruktion:

```
[Marke:] Operation [Operanden] [;Kommentare]
```

Das Operationsfeld ist das einzige Feld, das immer vorhanden ist, alle anderen Felder sind optional. Daher sind die Operations-Codes, die den Inhalt des Operationsfeldes ausmachen, der wichtigste Teil eines Assemblersprache-Programms. Der CP/M-Assembler akzeptiert alle mnemonischen Standardbefehle des 8080-Mikrocomputers, die ausführlich in Intel's „8080 Assembly Language Programming Manual“ beschrieben sind. Marken sind in jeder Programmzeile zugelassen. Die einzelnen Operatoren sind der Vollständigkeit halber in den

folgenden Abschnitten aufgeführt; bezüglich tiefergehender Details muß selbstverständlich auf die Handbücher von Intel verwiesen werden. In den nachstehenden Tabellen stellt

- e3 einen 3-bit-Wert aus dem Bereich 0-7 dar, der eines der vordefinierten Register A, B, C, D, E, H, L, M, SP oder PSW sein kann.
- e8 einen 8-bit-Wert aus dem Bereich 0-255 dar.
- e16 einen 16-bit-Wert aus dem Bereich 0-65535 dar.

Diese Ausdrücke e3, e8, e16 können aus einer beliebigen Kombination von Operanden und Operatoren gebildet werden. In einigen Fällen sind die Operanden auf bestimmte Werte aus einem zulässigen Bereich beschränkt, wie z. B. im Falle der PUSH-Anweisung. Auf diese Fälle wird bei der Nennung der einzelnen Anweisungen hingewiesen.

In den nachfolgenden Abschnitten wird jeder Operationscode in seiner allgemeinsten Form aufgelistet zusammen mit einem spezifischen Beispiel, einer kurzen Erklärung und gegebenenfalls zu beachtenden Einschränkungen.

#### 4.5.1 Sprungbefehle, Unterprogrammaufrufe und Unterprogrammrücksprünge

Die Jump-, Call- und Return-Befehle sind in mehreren unterschiedlichen Formen möglich, welche die Zustandsanzeiger, die in der 8080 Mikrocomputer-CPU gesetzt sind, abfragen. Dies sind sie:

JMP	e16	JMP L1	Unbedingter Sprung zu Marke
JNZ	e16	JNZ L2	Sprung zu Marke, falls Nichtnull-Bedingung erfüllt ist.
JZ	e16	JZ 100H	Sprung zu Marke, falls Nullbedingung erfüllt ist.
JNC	e16	JNC L1+4	Sprung zu Marke, falls kein Übertrag
JC	e16	JC L3	Sprung zu Marke bei Übertrag

JPO	e16	JPO \$+8	Sprung zu Marke bei ungerader Parität
JPE	e16	JPE L4	Sprung zu Marke bei gerader Parität
JP	e16	JP GAMMA	Sprung zu Marke bei positivem Ergebnis
JM	e16	JM a1	Sprung zu Marke bei negativem Ergebnis
CALL	e16	CALL S1	Unbedingter Unterprogrammaufruf
CNZ	e16	CNZ S2	Unterprogrammaufruf, falls Nicht-nullbedingung erfüllt ist.
CZ	e16	CZ 100H	Unterprogrammaufruf, falls Nullbedingung erfüllt ist.
CNC	e16	CNC S1+4	Unterprogramm sprung, falls kein Übertrag
CC	e16	CC S3	Unterprogramm sprung, falls Übertrag
CPO	e16	CPO +8	Unterprogramm sprung bei ungerader Parität
CPE	e16	CPE S4	Unterprogramm bei gerader Parität
CP	e16	CP GAMMA	Unterprogramm sprung bei positivem Ergebnis
CM	e16	CM b1\$c2	Unterprogramm sprung bei ges. Minus-Anzeiger
RST	e3	RST 0	Programmgesteuerter Neustart, gleichwertig zu CALL 8 e3, ausgen. Ein-Byte-Call
RET			Rückprung aus Unterprogramm

RNZ	Rücksprung, falls Nullbedingung nicht erfüllt ist
RZ	Rücksprung, falls Nullbedingung nicht erfüllt ist
RNC	Rücksprung, falls kein Übertrag
RC	Rücksprung, falls Übertrag
RPO	Rücksprung bei ungerader Parität
RPR	Rücksprung bei gerader Parität
RP	Rücksprung bei positivem Ergebnis
RM	Rücksprung, falls Minusanzeiger gesetzt

#### 4.5.2 Instruktionen, die den im Operandenfeld angegebenen Wert übertragen

Es stehen mehrere Instruktionen zur Verfügung, die Ein-Byte- oder Zwei-Byte-Register oder Speicherplätze einfacher Länge mit konstanten Werten laden, sowie Instruktionen, die unmittelbare arithmetische Operationen mit dem Inhalt des Akkumulators (Register A) ausführen.

MVI e3, e8	MVI B,255	Unmittelbare Datenübertragung in die Register A, B, C, D, E, H, L oder M (Speicher)
ADI e8	ADI 1	Unmittelbares Addieren des Operanden zu A ohne Übertrag
ACI e8	ACI OFFH	Unmittelbares Addieren des Operanden zu A mit Übertrag

SUI e8	SUI L+3	Subtrahieren von A ohne Übertrag
SBI e8	SBI L AND 11B	Subtrahieren von A mit Übertrag
ANI e8	ANI \$ AND 7FH	Logisches „UND“ A mit unmittelbaren Daten
XRI e8	XRI 1111\$0000B	„Ausschließendes ODER“ A mit unmittelbaren Daten
ORI e8	ORI L AND 1+1	Logisches „ODER“ A mit unmittelbaren Daten
CPI e8	CPI 'a'	Vergleich von A mit unmittelb. im Operandenfeld geg. Daten (wie SUI, außer daß A nicht verändert wird)
LXI e3,e16	LXI B,100H	Laden des unmittelbar geg. 16-Bit-Wortes in das angegebene Registerpaar (e3 muß für B, D, H oder SP stehen).

### 4.5.3 Inkrement und Dekrement-Instruktionen

Der Mikrocomputer 8080 stellt Instruktionen zum Inkrementieren oder Dekrementieren von einzelnen Registern oder Registerpaaren bereit. Diese Instruktionen sind

INR e3	INR E	Zählt den Inhalt eines Einzelregisters (8-Bit-Registers) um Eins weiter. e3 steht für eines von A, B, C, D, E, H, L, M
DCR e3	DCR A	Zählt den Inhalt eines Einzelregisters um Eins herunter. e3 steht für eines der Register A, B, C, D, E, H, L, M

INX e3	INX SP	Zählt Inhalt eines Registerpaars um 1 weiter. e3 steht für B, D, H oder SP
DCX e3	DCX B	Zählt Inhalt eines Registerpaars um 1 herunter. (e3 steht für B, D, H oder SP)

#### 4.5.4 Instruktionen zum Übertragen von Daten

Instruktionen, die Daten vom Speicher zur Zentraleinheit und von der Zentraleinheit zum Speicher sowie von einem Register zu einem anderen Register übertragen, sind nachstehend aufgeführt.

MOV e3,e3	MOV A, B	Überträgt Daten vom rechts stehenden Element zum links stehenden Element. e3 steht für eines der Einzelregister A, B, C, D, E, H, L oder M. MOV mM ist nicht zulässig.
LDAX e3	LDAX B	Lädt Register A mit dem Wert, der unter der im Registerpaar angegebenen Adresse vorliegt. e3 steht für eines der Registerpaare B oder D.
STAX e3	STAX D	Speichert den Inhalt des Registers A auf der Speicherstelle, deren Adresse in dem angegebenen Registerpaar steht. e3 steht für eines der Registerpaare B oder D.
LHLD e16	LHLD L1	Lädt das Registerpaar HL mit dem unter der direkt angegebenen Adresse gefundenen 16-Bit-Wert.
SHLD e16	SHLD L5+x	Legt den Inhalt des Registerpaares HL in der 2 Bytes umfassenden direkt adressierten Speicherstelle ab.

LDA e16	LDA Gamma	Legt den unter der direkt angegebenen 16-Bit-Adresse vorliegenden 8-Bit-Wert in Register A ab.
STA e16	STA X3-5	Speichert den Inhalt von Register A an der Speicherstelle, deren Adresse durch e16 direkt gegeben ist.
POP e3	POP PSW	Lädt Registerpaar mit dem auf dem Stapelspeicher oben liegenden Wert, ändert den Wert des Stack-Pointers. e3 steht für B, D, H oder PSW.
PUSH e3	PUSH B	Legt den Inhalt des angegebenen Registerpaars auf den Stapelspeicher, verändert den Wert des Stack-Pointers. e3 steht für B, D, H oder PSW.
IN e8	IN 0	Lädt Register A mit Daten von Port e8
OUT e8	OUT 255	Sendet Daten von Register A zu Port e8
XTHL		Lädt das Registerpaar HL mit dem auf der Stapelspitze liegenden Zwei-Byte-Wert und legt den alten Inhalt von HL auf dem Stapel ab. Somit Datenaustausch zwischen Stapelspitze und HL-Register.
PCHL		Lädt Programmzähler mit Inhalt von HL. Ermöglicht indirekten Sprung.

SPHL	Lädt den Stack-Pointer mit dem Inhalt von HL.
XCHG	Vertauschen der Inhalte der beiden Registerpaare DE und HL.

#### 4.5.5 Arithmetische und logische Befehle

Es folgen Instruktionen, mittels derer arithmetische oder logische Operationen durchgeführt werden, wobei der 8-Bit-Akkumulator (A-Register) eines der beiden Operandenregister ist und das Ergebnis der Operation aufnimmt.

ADD e3	ADD B	Addiert den Inhalt des durch e3 gegebenen Registers zum Inhalt des Akkumulators ohne Übertrag. e3 steht für eines der Register A, B, C, D, E, H oder L.
ADC e3	ADC L	Addiert den Inhalt des durch e3 gegebenen Registers mit Übertrag zum Inhalt des Akkumulators. e3 wie oben.
SUB e3	SUB H	Subtrahiert den Inhalt von Register e3 ohne Übertrag vom Akkumulator. e3 ist wie oben definiert.
SBB e3	SBB 2	Subtrahiert Register e3 von A mit Übertrag. e3 ist wie oben definiert.
ANA e3	ANA 1+1	Verknüpft den Inhalt des durch e3 gegebenen Registers bitweise nach der logischen „UND“-Funktion mit dem Inhalt des Akkumulators.

XRA e3	XRA A	„Exklusiv ODER“-Verknüpfung von e3 mit A. e3 wie oben.
ORA e3	ORA B	Logisches „ODER“ mit A. e3 wie oben definiert.
CMP e3	CMP H	Vergleicht Registerinhalt mit Inhalt von A. e3 wie oben definiert.
DAA		Das Ergebnis der letzten Operation der ALU, der momentane Akkumulatorinhalt, wird durch Dezimaladjustierung in BCD-Darstellung gebracht.
CMA		Der Akkumulatorinhalt wird bitweise komplementiert.
STC		Setzt das Übertrags-Flag auf 1.
CMC		Komplementiere das Übertrag-Flag.
RLC		Der Übertrag wird gleich dem höchstwertigen Bit des alten Akkumulatorinhalts. Ansonsten werden alle 8 Bits des Akkumulators um eine Stelle zyklisch nach links verschoben, das höchstwertige Bit des alten Akku-Inhalts wird zum niederwertigsten des neuen Akku-Inhalts.
RRC		Der Übertrag wird gleich dem niedrigstwertigen Bit des alten Akkumulatorinhalts. Sonst geht man vor wie vorhin, allerdings in der anderen Richtung. Das

niedrigstwertige Bit des alten Akkuinhalts wird zum höchstwertigen Bit des neuen Akkuinhalts.

RAL

Das gedachte 9 Bit-Register Übertr./Akku wird um 1 Stelle nach links rotiert: Das bei der Linksverschiebung rechts freigewordene Bit wird mit dem (alten) Wert des Übertragsflag aufgefüllt.

RAR

Übertrag/Akku wird um 1 Stelle nach rechts rotiert. Das im Akkumulator links freigewordene Bit wird mit dem (alten) Wert des Übertragsflag aufgefüllt.

DAD e3

DAD B

Addiert den Inhalt des angegebenen Registerpaars e3 zum Inhalt des HL-Registers. e3 steht für eines der Registerpaare bzw. Doppelregister B, D, H oder SP.

#### 4.5.6 Steuer-Anweisungen

Die vier verbleibenden Instruktionen, die allgemein als „Steueranweisungen“ bezeichnet werden, sind:

HLT

Der 8080-Prozessor wird angehalten.

DI

Nach diesem Befehl akzeptiert das System keine Anforderung mehr auf Programmunterbrechung.

EI

Die Programmarbeit kann durch eine Anforderung von außen unterbrochen und

ein geeignetes „Unterbrechungsprogramm“ abgearbeitet werden.

NOP

Keine Operation.

## 4.6 Fehlermeldungen

Werden Fehler innerhalb des in Assembler-Sprache geschriebenen Quellprogramms entdeckt, dann werden sie als Ein-Zeichen -Anzeigen am äußersten linken Rand der Quellauflistung aufgeführt. Darüber hinaus wird die fehlerhafte Zeile an der Konsole nochmals ausgeschrieben, so daß die Auflistung des Quellprogramms nicht nach Fehlern durchsucht werden muß. Die folgenden Fehlercodes werden ausgegeben:

- C Kommandofehler: Das Kommando wird nicht erkannt. (Eine Art Syntax-Fehler.)
- D Datenfehler: Der Wert eines Operanden paßt nicht in den angegebenen Datenbereich.
- E Ausdruck falsch: Der Ausdruck ist fehlerhaft aufgebaut. Er kann nicht berechnet werden.
- L Markenfehler: Eine Marke kann in diesem Zusammenhang nicht vorkommen. (Häufige Ursache: Mehrere Anweisungen tragen Marke mit dem gleichen symbolischen Namen.)
- N Nicht implementiert: Der Befehl wird von der verwendeten Assemblerversion nicht akzeptiert.
- O Überlauf: Der Ausdruck ist zu kompliziert, d. h. er enthält zu viele Operatoren, um berechnet werden zu können; er sollte vereinfacht werden.
- P Phasenfehler: Der symbolische Name einer Marke hat bei zwei aufeinanderfolgenden Durchgängen durch das Programm nicht denselben Wert.
- R Registerfehler: Das angegebene Register kann bei diesem Befehl nicht auftreten.

- S        Syntax-Fehler: Die Anweisung ist nicht den vorgeschriebenen Regeln entsprechend formuliert.
- U        Nicht definierte Marke: Die verwendete Marke wurde nicht gefunden.
- V        Wert falsch: Ein im Ausdruck vorgefundener Operand ist unrichtig angegeben.

Mehrere Fehlermeldungen können ausgegeben werden aufgrund von Fehlersituationen, deren Ursachen nicht im eigentlichen Assemblerprogramm liegen.

**NO SOURCE FILE PRESENT** Die angegebene Assembler-Quelldatei existiert auf der Platte nicht.

**NO DIRECTORY SPACE** Das Plattenverzeichnis ist voll; löschen Sie nicht mehr gebrauchte Dateien und machen Sie einen neuen Versuch.

**SOURCE FILE NAME ERROR** Nicht korrekt gebildeter Dateiname

**SOURCE FILE READ ERROR** Die Quelldatei kann vom Assembler nicht ordnungsgemäß gelesen werden; geben Sie ein TYPE-Kommando ein, um die Fehlerstelle zu finden.

**OUTPUT FILE WRITE ERROR** Die Ausgabedateien können nicht sachgemäß geschrieben werden; häufigste Ursache ist eine volle Platte. Löschen Sie nicht (mehr) gebrauchte Dateien und versuchen Sie dann einen neuen Programmablauf.

**CANNOT CLOSE FILE** Ausgabedatei kann nicht geschlossen werden; bitte überprüfen Sie, ob die Platte schreibgeschützt ist.

## 4.7 Beispiele

In diesem Abschnitt wird das wechselseitige Zusammenwirken von Assembler und Testhilfen (debugger) bei dem Aufbau eines einfachen Programms in der Assemblersprache beschrieben. Der ↵ Pfeil bezeichnet das Drücken der Taste Wagenrücklauf.

```
A>ASM SORT ↵   Assembliere SORT.ASM
```

```
CP/M ASSEMBLER-VER 1.0
```

```
015C   Nächste freie Adresse
```

```
003H USE FACTOR  %-Satz der benutzten Tabelle 00 bis ff (hexadezimal)
```

```
END OF ASSEMBLY
```

```
A>DIR SORT.* ↵
```

```
SORT   ASM   Quelldatei
```

```
SORT   BAK   Sicherheitskopie aus der letzten Verwendung von EDITOR
```

```
SORT   PRN   Druckdatei
```

```
SORT   HEX   Maschinencode-Datei
```

```
A>TYPE SORT.PRN ↵
```

```
Quellezeile
```

```
      ;          SORTIER-PROGRAMM IM CP/M ASSEMBLER
      ;          START AM ANFANG DES TPA-BEREICHS
```

```
Maschinencode-Stelle
```

```
0100 ←          ORG
```

```
100H
```

Erzeugter Maschinencode

```

0100 214601 SORT: LXI H,SW ;ADRESS-KIPPSCHALTER
0103 3601 MVI M,1 ;SETZE 1 FÜR ERSTE ITERATION
0105 214701 LXI H,I ;ADR. INDEX
0108 3600 MVI M,0 ;I = 0
;
; VERGLEICH I MIT STRUKTURGRÖSSE
010A 7E COMPL: MOV A,M ;A REGISTER = I
010B FE09 CPI N-1 ;CY SET IF I < (N-1)
010D D21901 JNC CONT ;CONTINUE IF I <= (N-2)
;
; ENDE DES LAUFS
0110 214601 LXI H,SW ;PRÜFUNG DER NULL-SCHALTER
0113 7EB7C20001 MOV A, M! ORA A! JNZ SORT ;END OF SORT IF SW=0
;
0118 FF RST 7 ;GO TO DEBUGGER
; FORTSETZUNG DES LAUFS
Abgebrochen ; ADDRESSING I. SO LOAD AV(I) IN REGISTER
0119
5F16002148 CONT MOV E, A!MVI D, 0! LXI H, AV! DAD D! DAD D
0121 4E792346 MOV C, M! MOV A, C! INX H! MOV B, M
; LOW ORDER BYTE IN A AND C, HIGH ORDER BYTE IN B
;
; MOV H AND L TO ADRESSE AV(I+1)
0125 23 INX H
;
; WERTVERGLEICH MIT REGISTERINHALT AV (I)
0126 965778239E SUB M! MOV D, A! MOV A, B! INX H! SBB M ;SUBTRACT
;
; ÜBERLAUF SET IF AV(I+1) > AV(I)
012B DA3F01 JC INCI ;SKIP IF RICHTIGE FOLGE
;
; PRÜFUNG AUF GLEICHWERTIGKEIT
012E B2CA3F01 ORA D! JZ INCI ;SKIP IF AV(I) = AV(I+1)
0132 56702B5E MOV D, M! MOV M, B! DCX H! MOV E, M
0136 712B722B73 MOV M, C! DCX H! MOV M, D! DCX H! MOV M, E
;
; ERHÖHUNG SCHALT-ZÄHLER
013B 21460134 LXI H,SW! INR M
;
; INCREMENT I
013F 21470134C3INCI: LXI H,! INR M! JMP COMP
;
; DATEN-DEFINITIONEN
0146 00 SW: DB 0 ;PLATZRESERVIERUNG FÜR SCHALT-ZÄHLER
0147 I: DS 1 ;PLATZRESERVIERUNG FÜR INDEX
0148 050064001EAV: DW 5, 100, 30, 50, 20, 7, 1000, 300, 100, -32767
000A = N EQU ($-AV)/2 ;COMPUTE N
015C END
A>TYPE SORT.HEX Gleicher Wert

```

```

:10010000214601360121470136007EFE09D2190140 }
:100110002146017EB7C20001FF5F16002148011988 } Maschinencode
:10012000194E79234623965778239EDA3F01B2CAA7 } in HEX Format

```

```
:100130003F0156702B5E712B722B732146013421C7
:07014000470134C30A01006E
:10014800050064001E00320014000700E8032C01BB
:0401580064000180BE
:0000000000
A>DDT SORT. HEX/      Start Prüflauf
```

Maschinencode  
in HEX Format

```
16K DDT VER 1.0
NEXT PC
015C 0000      Default Adresse (keine Adresse beim END Statement)
-XP/
```

```
P=0000 100/      Ändere PC auf 100
```

```
-UFFFF/      Nichtverfolgung von 65535 Schritten
```

```
COZOM0E0I0 A=00 B=0000 D=0000 H=0000 S=0100 P=0100 LXI H,0146;0100
-T10/      Verfolgung von 10H Schritten
```

Abbruch

```
COZOM0E0I0 A=01 B=0000 D=0000 H=0146 S=0100 P=0100 LXI H, 0146
COZOM0E0I0 A=01 B=0000 D=0000 H=0146 S=0100 P=0103 MVI M, 01
COZOM0E0I0 A=01 B=0000 D=0000 H=0146 S=0100 P=0105 LXI H, 0147
COZOM0E0I0 A=01 B=0000 D=0000 H=0147 S=0100 P=0108 MVI M, 00
COZOM0E0I0 A=01 B=0000 D=0000 H=0147 S=0100 P=010A MOV A, M
COZOM0E0I0 A=00 B=0000 D=0000 H=0147 S=0100 P=010B CPI 09
C1ZOM1E0I0 A=00 B=0000 D=0000 H=0147 S=0100 P=010D JNC 0119
C1ZOM1E0I0 A=00 B=0000 D=0000 H=0147 S=0100 P=0110 LXI H, 0146
C1ZOM1E0I0 A=00 B=0000 D=0000 H=0146 S=0100 P=0113 MOV A, M
C1ZOM1E0I0 A=01 B=0000 D=0000 H=0146 S=0100 P=0114 ORA A
COZOM0E0I0 A=01 B=0000 D=0000 H=0146 S=0100 P=0115 JNZ 0100
COZOM0E0I0 A=01 B=0000 D=0000 H=0146 S=0100 P=0100 LXI H, 0146
COZOM0E0I0 A=01 B=0000 D=0000 H=0146 S=0100 P=0103 MVI M, 01
COZOM0E0I0 A=01 B=0000 D=0000 H=0146 S=0100 P=0105 LXI H, 0147
COZOM0E0I0 A=01 B=0000 D=0000 H=0147 S=0100 P=0108 MVI M, 00
COZOM0E0I0 A=01 B=0000 D=0000 H=0147 S=0100 P=010A MOV A, M*010B
-A10D      Beendet auf 10BH
```

```
010D JC 119/      Sprung bei Überlauf
0110/
```

```
-XP/
```

```
P=010B 100/      Rücksetzen des Programm-Zählers auf den Beginn des Programms
```

```
-T10/      Verfolgung der Ausführung von 10H Schritten
```

```
COZOM0E0I0 A=00 B=0000 D=0000 H=0147 S=0100 P=0100 LXI H,0146
COZOM0E0I0 A=00 B=0000 D=0000 H=0146 S=0100 P=0103 MVI M,01
COZOM0E0I0 A=00 B=0000 D=0000 H=0146 S=0100 P=0105 LXI H,0147
COZOM0E0I0 A=00 B=0000 D=0000 H=0147 S=0100 P=0108 MVI M,00
COZOM0E0I0 A=00 B=0000 D=0000 H=0147 S=0100 P=010A MOV A,M
COZOM0E0I0 A=00 B=0000 D=0000 H=0147 S=0100 P=010B CPI 09
C1ZOM1E0I0 A=00 B=0000 D=0000 H=0147 S=0100 P=010D JC 0119
C1ZOM1E0I0 A=00 B=0000 D=0000 H=0147 S=0100 P=0119 MOV E,A
C1ZOM1E0I0 A=00 B=0000 D=0000 H=0147 S=0100 P=011A MVI D,00
```

Geänderte Instruktion



-L10D/ Fehlerhaften OP-CODE auflisten

010D JNC 0119  
 0110 LXI H,0146

-Abbruch des Listens  
 -A10D/ Assemblieren des neuen OP-CODE

010D JC 119/

0110/

-L100/ Listen des ersten Programmabschnitts

0100 LXI H,0146  
 0103 MVI M,01  
 0105 LXI H,0147  
 0108 MVI M,00

-Abbruch des Listens  
 -A103/ Adreß-Kippschalter 00 setzen

0103 MVI M,0/

0105/

=C Rückkehr zu CP/M mit ctl-C (G0 ist ebenso verwendbar)

SAVE 1 SORT.COM/ Sichern 1 Speicherseite (250 Bytes, von 100H bis 1ffH) auf der Platte für  
 den Fall daß, ein Wiederladen erforderlich ist  
 A>DDT SORT.COM/ Wiederstart DDT mit dem gesicherten Speicherinhalt

16K DDT VER 1.0  
 NEXT PC  
 0200 0100 Die COM Datei beginnt immer auf Adresse 100H  
 -G/ Programmstart von PC=100H

\*0118 Programmierter Halt (RST 7) festgestellt  
 -D148

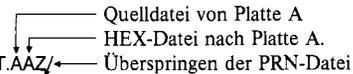
0148 05 00 07 00 14 00 1E 00 ..... Daten richtig sortiert  
 0150 32 00 64 00 64 00 2C 01 E8 03 01 80 00 00 00 00 2.D.D.....

0160 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....  
 0170 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....

-G0/ Rückkehr zu CP/M

A>ED SORT.ASM/ Durchführung von Änderungen des Originalprogramms

```
*N.0 ZOTT/ Finde „0“  
MVI M,0 ;I = 0  
  
*-/ Eine Zeile weiter im Text  
LXI H, I ;ADR. INDEX  
  
*-/ Und noch eine Zeile  
MVI M, 1 SETZE 1 FÜR ERSTE ITERATION  
  
*KT/ Die Zeile löschen und eine neue Zeile eingeben  
LXI H, I ;ADR. INDEX  
  
*/ Neue Zeile einfügen  
MVI M, 0 ;NULL  
  
*T/ LXI H, I ;ADR. INDEX  
  
*NJNC ZOT/  
JNC*T/  
CONT ;CONTINUE IF I <= (N-2)  
  
*-2DIC ZOLT/  
JC CONT ;CONTINUE IF I <= (N-2)
```

\*E/   
A>ASM SORT.AAZ/ ← Überspringen der PRN-Datei

CP/M ASSEMBLER - VER 1.0

015C Nächste zu assemblierende Adresse  
003H USE FACTOR  
END OF ASSEMBLY

A>DDT SORT.HEX/ Änderung des Prüfprogramms

16K DDT VER 1.0  
NEXT PC  
015C 0000  
-G100/

\*0118  
-D148/

  
0148 05 00 07 00 14 00 1E 00 .....  
0150 32 00 64 00 64 00 2C 01 E8 03 01 80 00 00 00 00 2.D.D.....  
0160 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00.....

-Abbruch

-GO/ Rückkehr zu CP/M-Programmprüfungen OK.

# DYNAMISCHE TESTHILFE

## 5.1 Einführung

Das DDT-Programm erlaubt dynamisches, interaktives Testen und Fehlerbeseitigen („DEBUG“) bei Programmen, die im Umfeld von CP/M erstellt wurden. Die Testhilfe kann mit einem Befehl einer der folgenden Formen aufgerufen werden:

DDT

DDT Dateiname.HEX

DDT Dateiname.COM

„Dateiname“ ist der Name des Programms, das geladen und getestet werden soll. In allen Fällen wird das DDT-Programm in den Hauptspeicher an die Stelle des CCP gebracht (der Benutzer sei auf Kapitel 6 bezüglich der Standard-Speicherorganisation verwiesen), und verbleibt dann direkt unterhalb des Basic Disk Operating System-Teils von CP/M. Die BDOS-Startadresse, festgehalten im Adreßfeld der JMP-Instruktion in Adresse 5H, wird geändert, um der verminderten Größe der TPA Rechnung zu tragen.

Die zweite und dritte Form des DDT-Befehls haben dieselbe Wirkung wie die erste, außer daß hier nachfolgend ein automatisches Laden der spezifizierten HEX- oder COM-Datei erfolgt. Der Ablauf ist identisch der Befehlsfolge

DDT

IDateiname.HEX oder IDateiname.COM

R

wobei der I-Befehl und der R-Befehl das zum Test vorgesehene Programm festhalten und einlesen (der Benutzer sei bezüglich genauer Details auf unten folgende Erklärungen zum I- und R-Befehl verwiesen).

Nach dem Aufruf gibt DDT die Bereitschaftsmeldung im Format

DDT VER m.m

aus, wobei m.m die Revisionsnummer angibt.

Dieser Nachricht folgt das Promptzeichen „-“ und damit wartet DDT auf Befehlseingaben von der Konsole.

Der Operator kann nun einen von mehreren Ein-Zeichen-Befehlen eingeben, abgeschlossen durch ein <CR>, um die Ausführung des Befehls einzuleiten. Jede Eingabezeile kann mit Hilfe von CP/M-Standardwerkzeugen aufbereitet bzw. korrigiert werden.

RUBOUT löscht das zuletzt eingegebene Zeichen

ctl-U löscht die gesamte Zeile und bereitet Neueingabe vor.

ctl-C System-Neustart

Jeder Befehl kann bis zu 32 Zeichen lang sein (ein automatischer <CR> wird als 33. Zeichen eingesetzt), wobei das erste Zeichen die Befehlsart festlegt.

- A Eingabe mnemonischer Assemblerbefehle mit Operanden
- D Anzeige der Speicherinhalte in Hex- und ASCII-Code
- F Einbringen von konstanten Daten in den Speicher
- G Start der Programmausführung mit optionalen Stops
- I Eröffnen eines Standard Eingabedateibescreibers
- L Auflisten der Speicherinhalte mittels des Disassemblers
- M Verschieben eines Speichersegmentes aus der ursprünglichen Lage in ein vorgegebenes Ziel
- R Liest ein Programm mit dem Ziel anschließenden Testens

- S      Verändern von Speicherinhalten
- T      Ausführung der Programmablaufverfolgung
- U      Ausführung der Programmablaufverfolgung ohne  
Ausgabe der Zwischenschritte
- X      Abfragen und gegebenenfalls Ändern des CPU-  
Status.

Dem Befehlsbuchstaben folgen in einigen Fällen kein, ein, zwei oder drei Hexadezimalwerte, die voneinander durch je ein Komma oder Leerzeichen getrennt sind. Alle zahlenmäßigen Ausgaben von DDT erfolgen in hexadezimaler Form. Die Befehle werden nicht ausgeführt, bevor am Ende des Befehls ein <CR> eingegeben wird.

An jeder Stelle des Testlaufs kann die Ausführung von DDT entweder durch `ctl-C` oder durch `G0` (Sprung zu Adresse `0000H`) gestoppt und der augenblickliche Speicherzustand durch ein `SAVE`-Kommando der Form

`SAVE n Dateiname.COM`

gesichert werden, wobei `n` die Anzahl der Seiten (Blöcke von 256 Bytes) angibt, die auf die Platte gespeichert werden sollen. Die Anzahl der Blöcke wird bestimmt durch das höherwertige Byte der Speicheradresse in der TPA und Umwandlung dieser Zahl in das Dezimalformat. Ist z. B. die höchste belegte Adresse in der TPA `1234H`, so beträgt die Seitenanzahl `12H` oder 18 in Dezimalangabe.

Der Operator könnte während des `DEBUG`-Laufes `ctl-C` eingeben, was eine Rückkehr zum `CCP`-Level zur Folge hätte. Nachfolgend wäre die Eingabe von

`SAVE 18 X.COM`

möglich. Der Speicherauszug wird als Datei `X.COM` auf die Platte gesichert und kann durch Eingabe von `X` direkt ausgeführt werden. Falls weiteres Testen erforderlich sein sollte, kann der Speicherauszug durch Eingabe von

`DDT X.COM`

abgerufen werden. Denn obiger Befehl lädt das vorher gesicherte Programm ab Adresse `100H` in einer Länge von 18 Seiten (bis `23FFH`). Der CPU-Status ist nicht Teil der `COM`-Datei; daher muß

das Programm von Anfang an neu gestartet werden, um einen angemessenen Test zu ermöglichen.

## 5.2 DDT-Befehle

Die einzelnen Befehle sind nachfolgend genauer beschrieben. In jedem Fall muß auf das Freigabezeichen („-“) gewartet werden, bevor ein Befehl eingegeben wird. Wenn die Kontrolle an ein zu testendes Programm übergeben wurde und dieses Programm einen Break-Punkt noch nicht erreicht hat, kann die Kontrolle wieder an DDT übertragen werden mittels Auslösung von RST 7 (Neustart) durch Aus- und Ein-Schalten des Computers.

Bei der Erklärung von jedem Befehl wird der Befehlsbuchstabe in einigen Fällen zusammen mit Zahlen dargestellt, die voneinander durch Kommas getrennt sind. Die Zahlen werden dabei durch Kleinbuchstaben dargestellt. Diese Zahlen werden stets als Hexadezimalzahlen von ein bis vier HEX-Ziffern Länge (längere Zahlen werden automatisch rechts abgeschnitten) angenommen.

Viele der Kommandos beziehen sich in ihrer Wirkung auf einen „CPU-Status“, der dem Ablauf des zu testenden Programms entspricht. Der CPU-Status meint in erster Linie den Inhalt der Register des Programms, das auf Fehler geprüft wird. Alle Register und Anzeigen enthalten im Grundzustand Nullen. Eine Ausnahme bilden nur der Programmzähler (PC) und der Stapelzeiger (S), die am Anfang den Wert 100H enthalten. Der Programmzähler wird anschließend auf den Wert der Startadresse gesetzt, die im letzten Eintrag einer HEX-Datei angegeben ist, falls eine solche Datei geladen wird (siehe die I- und R-Befehle).

### 5.2.1 Der A (Assembly)-Befehl

DDT ermöglicht die direkte Ablage von mnemonischen Befehlen in das Speicherbild mit Hilfe des A-Befehls. Dieser hat die Form

As

worin s die Startadresse in hexadezimaler Form für diese „innere“ Assemblierung darstellt. DDT antwortet darauf durch Konsolausgabe der Adresse für die nächste Instruktion, liest die darauffolgende Konsoleingabe, wobei ein mnemonischer Befehl erwartet wird (siehe Kapitel 4 die Intel 8080 Assembly Language Reference Card bezüglich einer Liste der mnemonischen Befehle), gefolgt von Registerbe-

zügen und Operanden in hexadezimaler Form. Jede nachfolgende Ladeadresse wird ausgedruckt, bevor die zugehörige Konsoleingabe gelesen wird. Das A-Kommando schließt ab, sobald eine leere Zeile (CR) am Konsolgerät abgesetzt wird.

Nach Abschluß der Eingabe von Assemblerbefehlen kann der Bediener sich einen Überblick über das ausgewählte Speichersegment verschaffen, indem er dieses disassembliert (siehe das L-Kommando).

Der Benutzer sollte beachten, daß der Speicherbereich des Assembler/Disassembler-Teils von DDT durch das zu testende Programm überschrieben werden kann. In diesem Fall meldet DDT beim Versuch, das A-Kommando oder das L-Kommando einzusetzen, einen Fehler.

### 5.2.2 Der D(Display)-Befehl

Der D-Befehl gibt dem Operator die Möglichkeit, den Inhalt des Speichers im Hexadezimal- und im ASCII-Format anzusehen. Der D-Befehl kann in folgenden Formen verwendet werden:

D

Ds

Ds,f

In der ersten Form werden sechzehn Zeilen Speicherinhalt, beginnend mit der momentanen Adresse (am Anfang 100H) ausgegeben. Jede Zeile der Anzeige sieht dabei so aus:

```
aaaa bb cccccccccccccccc
```

Dabei ist aaaa die Adresse des ersten Byte der sechzehn Byte langen Zeichenfolge und bb der Inhalt der sechzehn Bytes in hexadezimaler Form. Die ASCII-Zeichen dieser Zeichenfolge sind rechts aufgeführt (dargestellt durch die c's), wobei nichtdruckbare Zeichen als Punkt (.) ausgegeben werden. Man beachte, daß sowohl Groß- als auch Kleinbuchstaben angezeigt werden. Jede Zeile der Anzeige gibt 16 Datenbytes aus, wobei die erste Zeile so abgeschnitten wird, daß die folgenden Zeilen bei Adressen beginnen, die ein Vielfaches von 16 sind.

Die zweite Form des D-Befehls ist ähnlich der ersten, außer daß die Adresse, bei der die Ausgabe beginnen soll, mit s vorgegeben ist. Bei

der dritten Form soll die Ausgabe von Adresse s bis Adresse f erfolgen. In jedem Fall wird die „Anzeigeadresse“ auf den Wert der ersten Adresse gesetzt, deren Inhalt durch den eben abgearbeiteten Befehl nicht mehr ausgegeben wurde. Damit ist durch nachfolgende D-Befehle ohne explizite Adreßangabe eine kontinuierliche Ausgabe möglich.

Überlange Ausgaben können durch Drücken der Abschlußtaste abgebrochen werden.

### 5.2.3 Der F(Fill)-Befehl

Der F-Befehl nimmt die Form

Fs,f,c

an. Dabei ist s die Startadresse, f ist die Endadresse und c eine hexadezimale, ein Byte darstellende Konstante.

DDT legt die Konstante c bei Adresse s ab, dann wird s erhöht und die neue Adresse s mit der Endadresse f verglichen. Übersteigt der neue Wert von s den Wert von f, so endet die Operation, andernfalls wird der beschriebene Vorgang wiederholt. Der Füllbefehl kann somit dazu verwendet werden, jede Adresse des Speicherbereichs mit einer vorgegebenen Konstanten zu belegen.

### 5.2.4 Der G(Go)-Befehl

Bei Benutzung des G-Befehls läuft ein Programm mit bis zu zwei optionalen Unterbrechungsmarken. Der G-Befehl nimmt eine der Formen an:

G

Gs

Gs,b

Gs,b,c

G,b

G,b,c

In der ersten Form wird das Programm vom momentanen Wert des Programmzählers an ausgeführt, im momentanen Maschinenstatus. Break-Punkte sind nicht gesetzt. Die einzige Möglichkeit, die Kontrolle an DDT zurückzugeben, besteht darin, einen Hardware RST 7 auszulösen. Der derzeitige Wert des Programmzählers kann durch den X-Befehl oder den XP-Befehl ermittelt werden.

Die zweite Form ist ähnlich der ersten, jedoch wird vor der Programmausführung der PC auf den Wert s gesetzt.

Die dritte Form wirkt wie die zweite, außer daß die Programmausführung bei Erreichen der Adresse b (b muß innerhalb des zu testenden Programms liegen) terminiert. Die Instruktion bei Adresse b wird nicht mehr ausgeführt.

Die vierte Form ist weitgehend identisch der dritten, doch werden hier zwei Stopstellen spezifiziert, eine bei b und eine bei c. Bei Erreichen eines dieser Break-Punkte terminiert die Programmausführung und beide Stopstellen werden gelöscht.

Die beiden letzten Formen entnehmen den Wert des Programmzählers aus dem momentanen Maschinenstatus und setzen einen bzw. zwei Break-Punkte.

Die Programmausführung schreitet von der Startadresse in Echtzeit zum nächsten Break-Punkt fort. Es gibt in DDT keine Eingriffsmöglichkeit zwischen Startadresse und Break-Punkt. Falls das zu testende Programm den Break-Punkt aus irgendwelchem Grund nicht erreicht, kann DDT die Kontrolle nur durch Ausführung einer RST 7-Instruktion zurückerhalten. Wird der Break-Punkt dagegen erreicht, so bricht DDT die Programmausführung ab und gibt

\*d

aus, wobei d die Stop-Adresse ist. An dieser Stelle kann der Maschinenstatus durch Verwendung des X(Examine)-Befehls abgefragt werden.

Es ist sinnvoll, Break-Punkte zu spezifizieren, die vom Wert des Programmzählers am Anfang des G-Kommandos verschieden sind. Falls somit der momentane Wert des Programmzählers 1234H ist, dann generieren die Befehle

G,1234  
und  
G400,400

einen unmittelbaren Break-Punkt, ohne daß irgendwelche Instruktionen ausgeführt werden.

### 5.2.5 Der I(Input)-Befehl

Der I-Befehl ermöglicht es dem Benutzer, einen Dateinamen in den vorbesetzten Dateischreiber FCB (File Control Block) an der Stelle 5CH einzusetzen (der von CP/M für transiente Programme angelegte File Control Block liegt an dieser Stelle; siehe Kapitel 6). Der vorbesetzte FCB kann von dem zu testenden Programm verwendet werden, als stände es unter Kontrolle des CCP. Man beachte, daß DDT diesen Dateinamen auch zum Lesen zusätzlicher HEX- und COM-Dateien verwendet. Die Form des I-Kommandos lautet

IDateiname  
oder  
IDateiname.Typ

Wird die zweite Form des Befehls verwendet und der Dateityp ist entweder HEX oder COM, dann können nachfolgende R-Befehle dazu benutzt werden, den reinen Binärcode oder Hexadezimalmaschinencode zu lesen (Abschnitt 5.2.8 enthält weitere Details).

### 5.2.6 Der L(List)-Befehl

Der L-Befehl wird dazu benutzt, Segmente des Programms, das getestet werden soll, in mnemonischer Form auszulisten. Dies ist weitgehend identisch mit Disassemblierung. Der L-Befehl wird verwendet in den Formen:

L  
Ls  
Ls,f

In der ersten Form werden zwölf Zeilen des disassemblierten Maschinencodes vom momentanen Wert des Programmzählers an aufgelistet. In der zweiten Form wird der Wert der Ausgabeadresse auf den Wert s gesetzt, bevor ebenfalls zwölf Zeilen von eben dieser Ausgabeadresse an ausgegeben werden. In der letzten Form wird disassemblierter Code von s bis Adresse f ausgegeben. In allen drei Fällen wird die Ausgabeadresse auf den Wert der nächsten, noch nicht ausgegebenen Stelle gesetzt als Vorbereitung eines nachfolgenden L-Befehls.

Bei Feststellung eines Break-Punktes für die Programmausführung

wird die Ausgabeadresse auf den momentanen Wert des Programmzählers (G- und T-Befehle) gesetzt. Überlange Ausgaben können wieder durch Betätigen der Abschlußtaste abgebrochen werden.

### 5.2.7 Der M(Move)-Befehl

Der M-Befehl ermöglicht blockweisen Transport von Programm- oder Datenbereichen von einer Speicherstelle zu einer anderen. Der Befehl hat die Form

Ms,f,d

wobei s die Anfangsadresse der zu übertragenden Daten, f die Endadresse dieses Bereiches und d die Zieladresse angeben. Es wird der Inhalt der Adresse s nach Adresse d verschoben, dann werden beide Adressen inkrementiert. Übersteigt der neue Wert von s den Wert von f, so endet die Transportoperation; andernfalls wird der gesamte Vorgang wiederholt.

### 5.2.8 Der R(Read)-Befehl

Der R-Befehl wird in Verbindung mit dem I-Befehl dazu benutzt, COM- und HEX-Dateien von der Diskette in den Programmbereich der transienten Programme einzulesen als Vorarbeit für einen Test-(DEBUG-)Lauf. Der R-Befehl wird in den Formen

R

Rb

verwendet, wobei b eine optionale Verschiebung darstellt, die zu jeder Programm- oder Datenadresse hinzuaddiert wird. Die Ladeoperation darf keinen der Systemparameter in der ersten Seite des Speichers, d. h. im Bereich von 0000H bis 0FFH, überschreiben. Falls b weggelassen wird, wird b = 0000 angenommen.

Der R-Befehl erfordert einen vorausgehenden I-Befehl, der den Namen einer HEX-Datei oder einer COM-Datei angibt. Die Ladeadresse für jeden Eintrag erhält man aus dem entsprechenden HEX-Eintrag, während für COM-Dateien eine Ladeadresse von 100H unterstellt wird. Der Benutzer sollte beachten, daß das R-Kommando nach dem vorausgehenden I-Befehl beliebig oft abgesetzt werden kann, um das zu testende Programm erneut zu lesen (und evtl. zu korrigieren), wobei vorausgesetzt wird, daß dieses Programm den

Bereich um 5CH nicht zerstört. Jede Datei, die durch den Dateityp „COM“ gekennzeichnet ist, soll Maschinencode in rein binärer Form (angelegt mit dem LOAD- oder SAVE-Befehl) enthalten, und von allen anderen wird angenommen, daß sie Maschinencode in Intel Hexaformat (erzeugt z. B. mit dem ASM-Befehl) enthalten.

Es sei daran erinnert, daß der Befehl

DDT Dateiname.Dateityp,

der das DDT-Programm aufruft, gleichwertig ist der Befehlsfolge

DDT

-I Dateiname.Dateityp

-R

Wenn der R-Befehl abgesetzt wird, antwortet DDT entweder mit der Fehleranzeige „?“ (Datei kann nicht geöffnet werden oder Prüfsummenfehler in einer HEX-Datei) oder mit einer Ladenachricht der Form

NEXT PC

nnnn pppp

wobei nnnn die nächste, auf das geladene Programm folgende Adresse und pppp der angenommene Programmzähler (100H für COM-Dateien oder, falls eine HEX-Datei angegeben ist, aus dem letzten Eintrag entnommen) ist.

### 5.2.9 Der S(Set)-Befehl

Der S-Befehl ermöglicht die Abfrage bestimmter Speicherplätze und gegebenenfalls die Änderung ihres Inhalts. Der Befehl hat die Form

Ss

wobei s die Adresse in Hexadezimalform des Speicherplatzes ist, der abgefragt und geändert werden soll. DDT antwortet mit einem numerischen „Prompt“, indem es die Adresse des Speicherplatzes und seinen derzeitigen Inhalt ausgibt. Falls der Bediener ↵ eingibt, wer-

den die Daten nicht geändert. Wird ein Bytewert eingegeben, so wird dieser an der angezeigten Adresse gespeichert. In jedem Fall fährt DDT fort, durch die Ausgabe aufeinanderfolgender Adressen und ihrer Inhalte neue Eingaben anzufordern, bis entweder ein Punkt (.) eingegeben oder durch das System ein unzulässiger Eingabewert festgestellt wird.

### 5.2.10 Der T(Trace)-Befehl

Der T-Befehl ermöglicht es, den Ablauf der Programmausführung von 1 bis 65535 Programmschritten nachzuvollziehen. Die Formen des T-Befehls sind:

T  
Tn

Im ersten Fall wird der CPU-Status angezeigt und der nächste Programmschritt ausgeführt. Das Programm terminiert unmittelbar darauf, wobei die Adresse des Programmabbruchs wie folgt ausgegeben wird

\*hhhh

hhhh ist dabei die Adresse der nächsten auszuführenden Instruktion. Die Anzeigeadresse (verwendet im D-Befehl) erhält man aus den Werten der Register H und L und die Ausgabeadresse (verwendet im L-Befehl) wird auf den Wert hhhh gesetzt. Mittels des X-Befehls kann der CPU-Status beim Programmende abgefragt werden.

Die zweite Form des T-Befehls ist ähnlich der ersten, außer daß hier die Programmausführung über n Schritte (n ist ein hexadezimaler Wert) nachvollzogen wird, bevor ein Break-Punkt erscheint. Eine Unterbrechung kann im Trace Mode durch Eingabe eines RUBOUT-Zeichens erzwungen werden. Bevor jeder Programmschritt vollzogen wird, erfolgt im Trace Mode die Anzeige des CPU-Status. Das Format der Anzeige entspricht dem beim X-Befehl beschriebenen. Es sei darauf aufmerksam gemacht, daß die Programmablaufverfolgung über die Schnittstelle zum CP/M nicht fortgesetzt und wieder aufgenommen wird nach der Rückkehr vom CP/M zu dem Programm, das getestet werden soll. Somit laufen CP/M-Funktionen, die zu Ein-/Ausgabe-Geräten, wie Plattenlaufwerken, zugreifen, in Echtzeit ab, wodurch I/O-Zeitprobleme vermieden werden. Programme, die im Trace Mode ablaufen, haben eine etwa 500fache Ausführungszeit gegenüber denen, die in Echtzeit ablaufen, da DDT die Kontrolle

nach jeder ausgeführten Benutzerinstruktion erhält. Interrupt-Verarbeitungsroutinen können dem Trace-Prozeß unterzogen werden, aber Befehle, welche die Break-Punkt-Möglichkeit ausnutzen (G-, T- und U-Befehl) führen die Unterbrechung durch Verwendung einer RST 7 Instruktion aus, was zur Folge hat, daß das zu testende Programm eine solche Unterbrechungsstelle nicht wahrnehmen kann. Weiterhin bewirkt der Trace Mode, daß das Programm so abläuft, daß Interrupts grundsätzlich zugelassen sind. Dies kann zu Problemen führen, wenn asynchrone Interrupts während der Trace-Phase ankommen.

Sie sollten die CR-Taste verwenden, um DDT wieder die Kontrolle zu übertragen und nicht die RST 7 Operation erzwingen. Nur so ist sichergestellt, daß der Ablauf der laufenden Instruktion zu Ende verfolgt wird, bevor die Unterbrechung wirksam wird.

### 5.2.11 Der U(Untrace)-Befehl

Der U-Befehl ist identisch zum T-Befehl, außer daß Programmzweischritte nicht angezeigt werden. Der Untrace Mode erlaubt die Ausführung von 1 bis 65535 ( 0FFFFH) Schritten in überwachter Form und wird grundsätzlich verwendet, um die Kontrolle von einem laufenden Programm zurückzuerhalten, während es sich in einem stationären Zustand befindet. Alle Konditionen des T-Befehls gelten auch für den U-Befehl.

### 5.2.12 Der X(Examine)-Befehl

Der X-Befehl erlaubt die selektive Anzeige und Änderung des derzeitigen CPU-Zustandes für das zu tastende Programm. Er kommt in den Formen vor:

X

Xr

wobei r eines der 8080 CPU-Register oder Anzeiger (Flags) bedeutet.

C      Übertrags-Anzeiger (0/1)

Z      Null-Anzeiger      (0/1)

M	Minus-Anzeiger	(0/1)
E	Anzeiger Gerade Parität	(0/1)
I	Unterbrechung möglich/nicht möglich	(0/1)
A	Akkumulator	(0-FF)
B	BC Registerpaar	(0-FFFF)
D	DE Registerpaar	(0-FFFF)
H	HL Registerpaar	(0-FFFF)
S	Stapelzeiger	(0-FFFF)
PC	Programmzähler	(0-FFFF)

Im ersteren Fall wird der CPU-Register-Status im Format

`CfZfMfEfff A=bb B=dddd D=dddd H=dddd S=dddd P=dddd inst`

angezeigt, wobei f ein 0- oder 1-Anzeigerwert ist, bb ein Bytewert und dddd eine Zweibyte-Größe ist, die zu einem Registerpaar gehört.

Das „inst“-Feld enthält die disassemblierte Instruktion, die an der durch den Programmzähler des derzeitigen CPU-Status angegebenen Adresse erscheint. Die zweite Form erlaubt Anzeige und gegebenenfalls Änderung der Registerinhalte, wo r eines der oben angegebenen Register ist (C,Z,M,E,I,A,B,D,H,S oder P).

In jedem Fall wird der Wert des FLAG oder des Registerinhalts zuerst an der Konsole angezeigt. Das DDT-Programm akzeptiert dann Eingabe von der Konsole. Wird CR eingegeben, so wird der Wert eines Anzeigers oder der Registerinhalt nicht geändert. Wird dagegen ein Wert aus einem geeigneten Bereich eingegeben, so wird der Anzeiger oder der Registerinhalt geändert. Es ist zu beachten, daß BC, DE und HL als Registerpaare angezeigt werden. Sie müssen somit den neuen Inhalt des ganzen Registerpaares eingeben, auch wenn nur ein Einzelregister geändert werden soll.

### 5.3 Anmerkungen zur Implementierung

Die Organisation von DDT erlaubt die Überlagerung bestimmter, nicht wesentlicher Teile des CP/M-Systems, um die TPA zum Zweck des DEBUG großer Programme zu vergrößern. Das DDT-Programm besteht aus zwei Teilen: dem DDT-Kern und dem Assembler/Disassembler-Modul. Der DDT-Kern wird über den CCP geladen und der Assembler/Disassembler-Modul - wiewohl zugleich mit dem DDT-Kern geladen - kann überschrieben werden, falls er nicht zum Assemblieren/Disassemblieren benötigt wird.

Insbesondere wird die BDOS-Adresse 6H (Adreßfeld der JMP-Anweisung auf Adresse 5H im unteren reservierten CP/M-Bereich) von DDT umgewandelt in die niedrigste Adresse des DDT-Kerns, die ihrerseits eine JMP-Instruktion zum BDOS enthält. Somit erkennen Programme, die dieses Adreßfeld zur Größenbestimmung des verfügbaren Speichers benutzen, das logische Ende des Speichers am Beginn des DDT-Kerns und nicht am Beginn von BDOS.

Der Assembler/Disassembler-Modul liegt unmittelbar unterhalb dem DDT-Kern in der TPA. Werden A-, L-, T- oder X-Kommandos während des DEBUG-Vorgangs benutzt, so ändert das DDT-Programm erneut das Adreßfeld 6H, um diesen Modul einzubeziehen. Somit erfolgt eine weitere Herabsetzung des logischen Speicherendes. Falls ein Programm beim Laden Speicherbereiche über dem Beginn des Assembler/Disassembler-Module belegt, sind die A- und L-Befehle nicht mehr verfügbar, (der Versuch ihrer Verwendung erzeugt ein „?“ als Antwort) und die Trace- und Anzeige-Befehle listen das „inst“-Feld der Anzeige in hexadezimaler Form und nicht als dekorierte Instruktion aus.

### 5.4 Ein Beispiel

Das nachfolgende Beispiel zeigt den Editiervorgang, das Assemblieren und Fehlersuchen sowie Korrigieren für ein einfaches Programm, das einen Satz von Daten liest und den größten Wert der Menge bestimmt. Der größte Wert wird dem Datenvektor entnommen und unter der Bezeichnung „LARGE“ am Ende des Programms abgespeichert.

A&gt;ED SCAN.ASM

Erzeuge Quellen-Programm  
"/" bedeutend Abschlußtaste

```

*/
                                ORG      1-00H      ;START OF TRANSIENT
                                                ;AREA/
                                MVI      B, LEN      ;LENGTH OF VECTOR TO SCAN/
                                MVI      C, 0        ;LARGER-RST VALUE SO FAR/
LOOP:                            LXI      H, VECT     ;BASE OF VECTOR/
                                MOV      A, M        ;GET VALUE/
                                SUB      C          ;LARGER VALUE IN C?/
                                JNC      NFOUND       ;JUMP IF LARGER VALUE NOT
                                                ;FOUND/
;                                NEW LARGEST VALUE, STORE IT TO C/
                                MOV      C, A
NFOUND:                          INX      H          ;TO NEXT ELEMENT/
                                DCR      B          ;MORE TO SCAN?/
                                JNZ      LOOP        ;FOR ANOTHER/

                                END OF SCAN, STORE C/
                                MOV      A, C        ;GET LARGEST VALUE/
                                STA      LARGE/
                                JMP      0          ;REBOOT/

*/
*/                                TEST DATA
VECT:                            DB      2,0,4,3,5,6,1,5
LEN:                              EQU      $-VECT    ;LENGTH
LARGE:                            DS      1          ;LARGEST VALUE ON EXIT/
                                END/

↑-Z
*/                                ORG      100H      ;START OF TRANSIENT AREA
                                MVI      B, LEN      ;LENGTH OF VECTOR TO SCAN
                                MVI      C,0        ;LARGER VALUE SO FAR
                                LXI      H, VECT     ;BASE OF VECTOR
LOOP:                            MOV      A,M        ;GET VALUE
                                SUB      C          ;LARGER VALUE IN C?
                                JNC      NFOUND       ;JUMP IF LARGER VALUE NOT
                                                ;FOUND
;                                NEW LARGEST VALUE, STORE IT TO C
                                MOV      C,A

```

```

NFOUND:      INX      H      ;TO NEXT ELEMENT
             DCR      B      ;MORE TO SCAN?
             JNZ      LOOP    ;FOR ANOTHER
;
             END OF SCAN, STORE C
             MOV      A,C     ;GET LARGEST VALUE
             STA      LARGE
             JMP      0      ;REBOOT
;
;
;          TEST DATA

VECT:        DB      2,0,4,3,5,6,1,5
LEN          EQU      $-VECT ;LENGTH
LARGE:       DS      1      ;LARGEST VALUE ON EXIT
             END

```

```

*E/ ← End of edit
A>ASM SCAN/   Start Assembler
CP/M ASSEMBLER - VER 1.0
0122
002H USE FACTOR
END OF ASSEMBLY   Assemblierung beendet; siehe Programm-„Listing“

```

```

A>TYPE SCAN.PRN/
Code address      Source program
0100 ←            ORG 100H      ;START OF TRANSIENT AREA
0100 0608         MVI B,LEN    ;LENGTH OF VECTOR TO SCAN
0102 0E00 Machine code ↓      MVI C,0      ;LARGEST VALUE SO FAR
0104 211901      LXI H,VECT   ;BASE OF VECTOR
0107 7E         LOOP:  MOV A,M  ;GET VALUE
0108 91         SUB C        ;LARGER VALUE IN C?
0109 D20D01     JNC NFOUND   ;JUMP IF LARGER VALUE NOT
                                ;FOUND
                                NEW LARGEST VALUE, STORE IT TO C
010C 4F         MOV C, A

010D 23         NFOUND: INX H   ;TO NEXT ELEMENT
010E 05         DCR B        ;MORE TO SCAN?
010F C20701     JNZ LOOP    ;FOR ANOTHER
;
;          END OF SCAN, STORE C
0112 79         MOV A, C     ;GET LARGEST VALUE
0113 322101     STA LARGE

0116 C30000     JMP 0      ;REBOOT
;
;          TEST DATA
Code data listing ;
gekürzt           ;
0119 0200040305 VECT:  DB 2,0,4,3,5,6,1,5
0008 Ausgleichs-  LEN   EQU $-VECT ;LENGTH
0121 wert         LARGE: DS 1      ;LARGEST VALUE ON EXIT
0122             END

```

A>DDT SCAN.HEX/ Start Debugger verwendet Hex-Format Maschinencode

DDT VER 1,0

NEXT PC

0121 0000

-X/ Letzte Ladeadresse + 1

Nächste Instruktion zur  
Durchführung bei

PC = 0

C0Z0M0E010 A=00 B=0000 D=0000 H=0000 S=0100 P=0000 OUT 7F

-XP/

P=0000 100/ Ändere PC auf 100

-X/

C0Z0M0E010 A=00 B=0000 D=0000 H=0000 S=0100 P=0100 MVI B,08

-L100/

PC geändert

Nächste Instruktion zur  
Durchführung bei PC=100

```

0100 MVI      B,08
0102 MVI      C,00
0104 LXI      H,0119
0107 MOV      A,M
0108 SUB      C
0109 JNC      010D
010C MOV      C,A
010D INX      H
010E DCR      B
010F JNZ      0107
0112 MOV      A,C
-L/

```

Disassemblierter Maschinencode bei 100H  
(siehe Source Listing zum Vergleich)

```

0113 STA      0121
0116 JMP      0000
0119 STAX     B
011A NOP
011B INR      B
011C INX      B
011D DCR      B
011E MVI      B,01
0120 DCR      B
0121 LXI      D,2200
0124 LXI      H,0200

```

Etwas mehr Maschinen-  
code. Beachten Sie, daß  
das Programm bei Stelle  
116 endet, mit einem JMP  
zu 0000. Der Rest des  
Listing sind assemblierte  
Daten.

-A116/ In den „In-Line“-Assembler Modus gehen, um den JMP 0000 in ein RST 7. Dies veranlaßt das getestete Programm zur Rückkehr nach DDT, falls 116H stets durchgeführt ist.

0116 RST

0117/ Einmaliges Betätigen der Abschlußtaste beendet den Assembler Modus.

-L113/ Code bei 113 auflisten zur Prüfung, ob RST 7 richtig eingegeben wurde.

```

0113 STA      0121
0116 RST      07      an Stelle von IMP
0117 NOP
0118 NOP
0119 STAX     B
011A NOP
011B INR      B
011C INX      B

```

-X/ Sehen Sie nach den Registern

COZOM0E010 A=00 B=0000 D=0000 H=0000 S=0100 P=0100 MVI B,08

-T/ Programm für einen Halt Ursprünglicher CPU-Stand, vor Durchführung

COZOM0E010 A=00 B=0000 D=0000 H=0000 S=0100 P=0100 MVI B,08\*0102  
-T/ Automatischer „Break“

Einen Schritt weiter (beachte 08H in B)

COZOM0E010 A=00 B=0800 D=0000 H=0000 S=0100 P=0102 MVI C,00\*0104  
-T/

Weiter (Register C ist leer)

COZOM0E010 A=00 B=0800 D=0000 H=0000 S=0100 P=0104 LXI H,0119\*0107  
-T3/ 3 Schritte weiter

COZOM0E010 A=00 B=0800 D=0000 H=0119 S=0100 P=0107 MOV A,M

COZOM0E010 A=02 B=0800 D=0000 H=0119 S=0100 P=0108 SUB C

COZOM0E011 A=02 B=0800 D=0000 H=0119 S=0100 P=0109 JNC 010D\*010D  
-D119/ Anzeige des Speicherinhaltes, beginnt bei 119H Automatischer Break bei 10DH

0119 (02 00 04 03 05 06 01) Programmdaten  
0120 (05) 11 00 22 21 00 02 7E EB 77 13 23 EB 0B (78) B1 „kleinbuchstaben“ „W # . (X)  
0130 C2 27 01 C3 03 29 00 00 00 00 00 00 00 00 00 00 ..... ) .....  
0140 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....  
0150 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..... Daten werden in  
0160 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..... ASCII angezeigt  
0170 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..... mit einem „“ an  
0180 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..... der Stelle von  
0190 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..... nicht-graphischen  
01A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..... Zeichen.  
01B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....  
01C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....

-X/ Derzeitiger CPU-Stand

COZOM0E011 A=02 B=0800 D=0000 H=0119 S=0100 P=010D INX H  
-T5/

5 Schritte weiter

COZOM0E011 A=02 B=0800 D=0000 H=0119 S=0100 P=010D INX H

COZOM0E011 A=02 B=0800 D=0000 H=011A S=0100 P=010E DCR B

COZOM0E011 A=02 B=0700 D=0000 H=011A S=0100 P=010F JNZ 0107

COZOM0E011 A=02 B=0700 D=0000 H=011A S=0100 P=0107 MOV A,M

COZOM0E011 A=00 B=0700 D=0000 H=011A S=0100 P=0108 SUB C\*0109  
U5/ Automatischer Break

Weiter, ohne Zwischen-Stände  
COZ1M0E111 A=00 B=0700 D=0000 H=011A S=0100 P=0109 JNC 010D\*0108  
-X/

CPU-Stand am Ende von U5

COZOM0E111 A=04 B=0600 D=0000 H=011B S=0100 P=0108 SUB C  
-G/ Programm vom derzeitigen PC bis zum Abschluß führen (in Echt-Zeit)

\*0116 Break bei 116H, wegen Durchführung von RST 7 in Maschinencode.  
-X/

CPU-Stand am Ende des Programms

COZ1M0E111 A=00 B=0000 D=0000 H=0121 S=0100 P=0116 RST 07  
-XP/

Prüfe und ändere Programnzähler

P=0116 100/

-X/

C0Z1M0E111 A=00 B=0000 D=0000 H=0121 S=0100 P=0100 MVI B,08  
-T10/

Erstes Datenelement  
Derzeit größter Wert  
Zum Vergleich C abziehen  
10 Schritte weiter

C0Z1M0E111 A=00 B=0800 D=0000 H=0121 S=0100 P=0100 MVI B,08  
C0Z1M0E111 A=00 B=0000 D=0000 H=0121 S=0100 P=0102 MVI C,00  
C0Z1M0E111 A=00 B=0800 D=0000 H=0121 S=0100 P=0104 LXI H,0119  
C0Z1M0E111 A=00 B=0800 D=0000 H=0119 S=0100 P=0107 MOV A,M  
C0Z1M0E111 A=02 B=0800 D=0000 H=0119 S=0100 P=0108 SUB C  
C0Z0M0E011 A=02 B=0800 D=0000 H=0119 S=0100 P=0109 JNC 010D  
C0Z0M0E011 A=02 B=0800 D=0000 H=0119 S=0100 P=010D INX H  
C0Z0M0E011 A=02 B=0800 D=0000 H=011A S=0100 P=010E DCR B  
C0Z0M0E011 A=02 B=0700 D=0000 H=011A S=0100 P=010F JNZ 0107  
C0Z0M0E011 A=02 B=0700 D=0000 H=011A S=0100 P=0107 MOV A,M  
C0Z0M0E011 A=00 B=0700 D=0000 H=011A S=0100 P=0108 SUB C  
C0Z1M0E111 A=00 B=0700 D=0000 H=011A S=0100 P=0109 JNC 010D  
C0Z1M0E111 A=00 B=0700 D=0000 H=011A S=0100 P=010D INX H  
C0Z1M0E111 A=00 B=0700 D=0000 H=011B S=0100 P=010E DCR B  
C0Z0M0E111 A=00 B=0600 D=0000 H=011B S=0100 P=010F JNZ 0107  
C0Z0M0E111 A=00 B=0600 D=0000 H=011B S=0100 P=0107 MOV A,M\*0108  
-A109/

0109 JC 10D/ Einen „heißen Patch“ in den  
Maschinencode eingeben, um  
3NC in 3C zu ändern

010C/

Seit A > C sollte das Programm den Wert  
von A in C verändert haben. Falls dieser  
Code nicht durchgeführt wurde, kann es  
erforderlich sein, daß 3NC eine 3C Instruk-  
tion ist.

-GO/ DDT anhalten, um eine Version des geänderten Programms zu sichern.

A>SAVE 1 SCAN.COM/ Programm verharrt auf erster Seite;  
diese absichern.

A>DDT SCAN.COM/ DDT neu starten, unter Sicherung des Speicherinhalts,  
um den Test fortzuführen.

DDT VER 1.0

NEXT PC

0200 0100

-L100/

einige Codes auflisten

0100 MVI B,08  
0102 MVI C,00  
0104 LXI H,0119  
0107 MOV A,M  
0108 SUB C  
0109 JC 010D  
010C MOV C,A  
010D INX H  
010E DCR B  
010F JNZ 0107  
0112 MOV A,C  
-XP/

Vorherige Änderung ist in X.COM vorhanden.

P=0100/

-T10/ Weiter, um zu sehen, wie die  
geänderte Version funktioniert

Daten von A nach C verschoben

```

COZ0M0E0I0 A=00 B=0000 D=0000 H=0000 S=0100 P=0100 MVI B,08
COZ0M0E0I0 A=00 B=0800 D=0000 H=0000 S=0100 P=0102 MVI C,00
COZ0M0E0I0 A=00 B=0800 D=0000 H=0000 S=0100 P=0104 LXI H,0119
COZ0M0E0I0 A=00 B=0800 D=0000 H=0119 S=0100 P=0107 MOV A,M
COZ0M0E0I0 A=02 B=0800 D=0000 H=0119 S=0100 P=0108 SUB C
COZ0M0E0I1 A=02 B=0800 D=0000 H=0119 S=0100 P=0109 JC 010D
COZ0M0E0I1 A=02 B=0800 D=0000 H=0119 S=0100 P=010C MOV C,A
COZ0M0E0I1 A=02 B=0802 D=0000 H=0119 S=0100 P=010D INX H
COZ0M0E0I1 A=02 B=0802 D=0000 H=011A S=0100 P=010E DCR B
COZ0M0E0I1 A=02 B=0702 D=0000 H=011A S=0100 P=010F JNZ 0107
COZ0M0E0I1 A=02 B=0702 D=0000 H=011A S=0100 P=0107 MOV A,M
COZ0M0E0I1 A=00 B=0702 D=0000 H=011A S=0100 P=0108 SUB C
C1Z0M1E0I0 A=FE B=0702 D=0000 H=011A S=0100 P=0109 JC 010D
C1Z0M1E0I0 A=FE B=0702 D=0000 H=011A S=0100 P=010D INX H
C1Z0M1E0I0 A=FE B=0702 D=0000 H=011B S=0100 P=010E DCR B
C1Z0M0E1I1 A=FE B=0602 D=0000 H=011B S=0100 P=010F JNZ 0107*0107,
-X/ Break nach 16 Schritten

```

C1Z0M0E1I1 A=FE B=0602 D=0000 H=011B S=0100 P=0107 MOV A,M  
-G,108/ Lauf vom derzeitigen PC und Break bei 108H

\*0108  
-X/

Nächste Daten

C1Z0M0E1I1 A=04 B=0602 D=0000 H=011B S=0100 P=0108 SUB C  
-T/

Einzelschritt für einige Zyklen

C1Z0M0E1I1 A=04 B=0602 D=0000 H=011B S=0100 P=0108 SUB C\*0109  
-T/

COZ0M0E0I1 A=02 B=0602 D=0000 H=011B S=0100 P=0109 JC 010D\*010C  
-X/

COZ0M0E0I1 A=02 B=0602 D=0000 H=011B S=0100 P=010C MOV C,A  
-G/ Lauf zum Abschluß

\*0116  
-X/

COZ1M0E1I1 A=03 B=0003 D=0000 H=0121 S=0100 P=0116 RST 07  
-S121/ Sehe nach dem Wert von „Groß“

- 0121 03/ Falscher Wert!
- 0122 00/
- 0123 22/
- 0124 21/
- 0125 00/
- 0126 02/
- 0127 7E/ – Ende des S-Befehls

-L100/

```

0100 MVI      B,08
0102 MVI      C,00
0104 LXI      H,0119
0107 MOV      A,M
0108 SUB      C
0109 JC       010D
010C MOV      C,A
010D INX      H
010E DCR      B
010F JNZ      0107
0112 MOV      A,C
-L/

```

Code überprüfen

```

0113 STA      0121
0116 RST      07
0117 NOP
0118 NOP
0119 STAX     B
011A NOP
011B INR      B
011C INX      B
011D DCR      B
011E MVI      B,01
0120 DCR      B
-XP/

```

P=0116 100/ Programmzähler zurückstellen

-T/

Einzelschritt, und Datenwerte überwachen

COZ1M0E111 A=03 B=0003 D=0000 H=0121 S=0100 P=0100 MVI B,08\*0102

-T/

COZ1M0E111 A=03 B=0803 D=0000 H=0121 S=0100 P=0102 MVI C,00\*0104

-T/

Zähler setzen → „am größten“ gesetzt

COZ1M0E111 A=03 B=0800 D=0000 H=0121 S=0100 P=0104 LXI H,0119\*0107

-T/

Basisadresse der gesetzten Daten

COZ1M0E111 A=03 B=0800 D=0000 H=0119 S=0100 P=0107 MOV A,M\*0108

-T/

Erste Daten nach A gebracht

COZ1M0E111 A=02 B=0800 D=0000 H=0119 S=0100 P=0108 SUB C\*0109

-T/

COZ0M0E011 A=02 B=0800 D=0000 H=0119 S=0100 P=0109 JC 010D\*010C

-T/

COZ0M0E011 A=02 B=0800 D=0000 H=0119 S=0100 P=010C MOV C,A\*010D

-T/

Erste Daten korrekt nach C verschoben

COZ0M0E011 A=02 B=0802 D=0000 H=0119 S=0100 P=010D INX H\*010E

-T/

COZ0M0E011 A=02 B=0802 D=0000 H=011A S=0100 P=010E DCR B\*010F

-T/

COZ0M0E011 A=02 B=0702 D=0000 H=011A S=0100 P=010F JNZ 0107\*0107  
-T/

COZ0M0E011 A=02 B=0702 D=0000 H=011A S=0100 P=0107 MOV A,M\*0108  
-T/

COZ0M0E011 A=00 B=0702 D=0000 H=011A S=0100 P=0108 SUB C\*0109  
-T/

C1Z0M1E010 A=FE B=0702 D=0000 H=011A S=0100 P=0109 JC 010D\*010D  
-T/

C1Z0M1E010 A=FE B=0702 D=0000 H=011A S=0100 P=010D INX H\*010E  
-L100/

0100	MVI	B,08
0102	MVI	C,00
0104	LXI	H,0119
0107	MOV	A,M
0108	SUB	C ←Damit Register A nicht zerstört wird, sollte dies ein CMP sein.
0109	JC	010D
010C	MOV	C,A
010D	INX	H
010E	DCR	B
010F	JNZ	0107
0112	MOV	A,C

-A108/

0108 CMP C/ „Heißer Patch“ bei 108H, ändert SUB zu CMP

0109

-GQ/ DDT für SAVE anhalten

A>SAVE 1 SCAN.COM/ Sichert Speicher-Ansicht

A>DDT SCAN.COM/ Neustart von DDT

DDT VER1.0

NEXT PC

0200 0100

-XP/

P=0100

-L116/

0116	RST	07	} Code auf korrektes Laden überprüfen.
0117	NOP		
0118	NOP		
0119	STAX	B	
011A	NOP		

-G.116/ von 100H zum Abschluß weiterführen

\*0116

-XC/ siehe nach Führung

C1/  
 -X/  
 siehe nach CPU-Status

C1Z1M0E111 A=06 B=0006 D=0000 H=0121 S=0100 P=0116 RST 07  
 -S121/ Siehe bei „Groß“ - es scheint korrekt zu sein.

0121 06/

0122 00/

0123 22

-G0/ Stop DDT

A>ED SCAN.ASM/ Editieren Sie das Quellenprogramm neu, und machen Sie beide Änderungen.

```
*NSUB/
*OLT/
  ctI-Z SUB C ;LARGER VALUE IN C?
*SSUB↑ZCMP↑ZOLT/
  CMP C ;LARGER VALUE IN C?
  JNC NFOUND ;JUMP IF LARGER VALUE NOT FOUND
*SNC↑ZC↑ZOLT/
  JC NFOUND ;JUMP IF LARGER VALUE NOT FOUND
*/E/
```

A>ASM SCAN.AAZ/ ← Neu-Assemblierung, Quelle von Platte A wählen  
 Hex zu Platte A, zu Z zuordnen (selektiert keine „Print“-Datei)

CP/M ASSEMBLER VER 1.0

0122  
 002H USE FACTOR  
 END OF ASSEMBLY

A>DDT SCAN.HEX/ Debugger nochmals ablaufen lassen, um Änderungen zu überprüfen

DDT VER 1.0  
 NEXT PC  
 0121 0000  
 -L116/

0116 JMP 0000 Überprüfen zur Sicherstellung, daß das Ende noch bei 116H steht.

0119 STAX B

011A NOP  
 011B INR B  
 -(löschen)

-G100,116/ Lauf vom Beginn bis zum Ende mit einem Break

\*0116 Break erreicht  
 -D121/ Sehe nach bei „Groß“

Korrekter Wert

0121 06 00 22 21 00 02 7E EB 77 13 23 EB 0B 78 B1 ' !...W # X  
0130 C2 27 01 C3 03 29 00 00 00 00 00 00 00 00 00 00 ' ..).....  
0140 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....

- (löschen)

G0/ DDT anhalten, Debug-Beispiel ist abgeschlossen.

# SYSTEMVERBINDUNG

## 6.1 Einführung

Dieses Kapitel beschreibt die Systemorganisation von CP/M, Rel. 2 inklusive der Speicherstruktur und der Systemeingangspunkte. Es ist das Ziel, alle Informationen bereitzustellen, die notwendig sind, um Programme zu schreiben, die unter CP/M ablaufen und die E/A-Möglichkeiten des Systems bezüglich der Peripheriegeräte und Disketten ausnutzen.

CP/M ist logisch in vier Teile aufgeteilt, welche Basic I/O-System (BIOS), Basic Disk Operating System (BDOS), Console Command Processor (CCP) und Transient Program Area (TPA) genannt werden. Das BIOS ist ein hardwareabhängiger Modul, das die Schnittstelle zu einem bestimmten Computersystem auf unterer Ebene genau definiert, was für die periphere Ein/Ausgabe unerlässlich ist. BIOS und BDOS sind logisch zusammengefaßt zu einem einzigen Modul, das einen einzigen Eingangspunkt hat und FDOS genannt wird. Der CCP ist ein eigenes Programm, welches FDOS benutzt, um eine Verbindung herzustellen zwischen dem Bediener und der Information, die auf dem Hintergrundspeicher katalogisiert ist. Die TPA ist ein Speicherbereich (nicht belegt von FDOS und CCP), in dem Benutzerprogramme und diverse Betriebssystemkommandos ablaufen. Der untere Bereich des Speichers ist für bestimmte Systeminformationen reserviert und wird in späteren Abschnitten genauer behandelt. Die Speicherorganisation von CP/M ist nachstehend gezeigt.

FBASE: Basis von FDOS	FDOS (BDOS+BIOS)	Hohe Speicheradressen
CBASE: Basis von CCP	CCP	
TBASE: Basis von TPA	TPA	
BOOT: Boot	System-Parameter	Niedrige Speicheradressen

Alle Standard-Versionen von CP/M unterstellen  $BOOT = 0000H$ , was die Basis des RAM darstellt. Der Maschinencode, der bei Adresse  $BOOT$  gefunden wird, bewirkt einen „Warmstart“ des Systems. Dabei werden die für eine Rückgabe der Kontrolle an CCP notwendigen Programme und Variablen initialisiert. Damit brauchen nichtresidente Programme lediglich zu Adresse  $BOOT$  zu springen, um auf Befehlsebene die Kontrolle an CP/M zurückzugeben.

Weiterhin wird bei den Standard-Versionen davon ausgegangen, daß  $TBASE = BOOT + 0100H$ , was normalerweise zu Adresse  $0100H$  führt. Der eigentliche Eingangspunkt zum FDOS liegt bei  $BOOT + 0005H$  (normalerweise  $0005H$ ), dessen Inhalt ein Sprung zu FBASE ist. Das Adreßfeld an der Stelle  $BOOT + 0006H$  (normalerweise  $0006H$ ) enthält den Wert von FBASE und kann dazu benutzt werden, die Größe des verfügbaren Speichers zu bestimmen; wobei angenommen wird, daß der CCP von einem transienten Programm überlagert wird.

Transiente Programme werden in die TPA geladen und wie folgt ausgeführt. Der Operator verkehrt mit dem CCP durch Eingabe von Befehlszeilen nach jedem „Prompt“. Jede Befehlszeile nimmt eine der Formen an

Kommando

Kommando Datei 1

Kommando Datei 1 Datei 2

wobei „Kommando“ entweder für eine residente Funktion wie DIR oder TYPE steht oder der Name eines transienten Kommandos oder Programms ist. Falls das Kommando eine residente Funktion von

CP/M ist, wird es unmittelbar ausgeführt. Andernfalls sucht der CCP die momentan adressierte Platte nach dem Namen

### Kommando.COM

durch. Wird die Datei gefunden, so wird unterstellt, sie sei die Speicherabbildung eines Programms, das in der TPA abläuft und deshalb im Speicher bei TBASE beginnt. Der CCP lädt die COM-Datei von der Platte in den Speicher, beginnend bei TBASE und reicht gegebenenfalls bis CBASE.

Folgen dem Kommando ein oder zwei Dateispezifikationen, so sieht der CCP einen oder zwei Dateibesreiber (FCB=File Control Block)-Einträge im Systemparameterbereich vor. Diese optionalen FCB's sind formal notwendig, um über FDOS Zugriff zu den Dateien zu bekommen. Sie werden im nächsten Abschnitt beschrieben.

Das transiente Programm erhält die Kontrolle vom CCP und beginnt mit der Ausführung, wobei die I/O-Möglichkeiten von FDOS genutzt werden. Das transiente Programm wird vom CCP aufgerufen. Daher kann es nach Beendigung seiner Arbeit entweder einfach zum CCP zurückkehren oder zu BOOT springen, wodurch CP/M die Kontrolle erhält. Im ersteren Fall darf das transiente Programm keine Speicherstelle oberhalb von CBASE belegen, während im letzteren Fall der Speicher bis zur Adresse FBASE genutzt werden kann.

Das transiente Programm kann die I/O-Möglichkeiten von CP/M ausnutzen, um mit der Operatorkonsole und den Peripheriegeräten einschließlich des Platten-Subsystems Kommunikation zu treiben. Zum I/O-System erhält man Zugriff durch Übergabe einer Funktionsnummer und einer Informationsadresse an CP/M via FDOS-Eingangspunkt an der Stelle BOOT+0005H. Im Falle eines Lesens von der Platte z. B. sendet das transiente Programm die dem Lesen von der Platte entsprechende Funktionsnummer zusammen mit der Adresse eines FCB an den FDOS-Teil von CP/M. FDOS seinerseits führt die Operation durch und gibt die Kontrolle zurück, entweder mit der Nachricht, daß die Operation „Platte lesen“ beendet ist, oder mit einer Fehlernummer, die anzeigt, daß „Platte lesen“ nicht erfolgreich war.

## 6.2 Konventionen der Betriebssystem-Aufrufe

Dieser Abschnitt vermittelt detaillierte Informationen bezüglich der Durchführung von Betriebssystem-Aufrufen vom Benutzerprogramm aus.

CP/M-Möglichkeiten, die für den Zugriff durch transiente Programme verfügbar sind, lassen sich in zwei Gruppen einteilen: Einfache Geräte-I/O und Plattendatei-I/O. Die einfachen Geräteoperationen beinhalten:

- Lesen eines Zeichens von der Konsole
- Schreiben eines Zeichens an die Konsole
- Lesen eines Zeichens vom Lochstreifen
- Stanzen eines Zeichens in Lochstreifen
- Drucken eines Zeichens an Drucker
- Lesen oder Setzen des I/O-Status
- Konsolpuffer ausgeben
- Konsolpuffer einlesen
- Konsol-Status abfragen

Die FDOS-Operationen, die Platten I/O durchführen sind:

- Plattensystem zurücksetzen
- Laufwerk festlegen
- Datei anlegen
- Datei eröffnen
- Datei schließen
- Verzeichnis durchsuchen
- Datei löschen
- Datei umbenennen
- Unmittelbar oder sequentiell lesen
- Unmittelbar oder sequentiell schreiben

## Ermittlung der verfügbaren Laufwerke

## Ermitteln des Bezugslaufwerks

## DMA Adresse setzen

## Dateimerkmale setzen/zurücksetzen

Wie oben bemerkt, erfolgt der Zugriff zu den FDOS-Funktionen durch Übergabe einer Funktionsnummer und einer Informationsadresse über den Primärpunkt bei BOOT+0005H. Im allgemeinen wird die Funktionsnummer an Register C übergeben und die Informationsadresse in das Registerpaar DE gelegt. Ein-Byte-Werte (wie z. B. Zeichen) werden von CP/M an Register A zurückgegeben, Zwei-Byte-Werte werden an HL (0 wird zurückgegeben, falls die Funktionsnummer außerhalb des zulässigen Bereichs liegt) zurückgegeben. Aus Gründen der Kompatibilität werden nach der Rückgabe in allen Fällen die Register A und L, sowie B und H gleichgesetzt. Man beachte, daß die Konventionen für die Registerübergabe mit denen der Systemprogrammiersprache PL/M von Intel übereinstimmen.

CP/M-Funktionen und ihre Nummern sind in der folgenden Tabelle zusammengefaßt.

0 System Warmstart	19 Datei löschen
1 Konsoleingabe	20 Lesen sequentiell
2 Konsolausgabe	21 Satz schreiben, sequentiell
3 Lochstreifen lesen	22 Datei anlegen
4 Lochstreifen stanzen	23 Datei umbenennen
5 Zeichen an Drucker ausgeben	24 aktives Laufwerk ermitteln
6 Direkte Konsolin- und -ausgabe	25 Bezugslaufwerk ermitteln
7 IOBYTE abfragen	26 DMA-Adresse festlegen
8 IOBYTE setzen	27 Belegungstabelle ermitteln
9 Zeichenkette ausgeben	28 Platte vor Schreiben schützen
10 Konsolpuffer lesen	29 Geschütztes Laufwerk ermitteln
11 Konsolstatus abfragen	30 Dateimerkmale setzen
12 CP/M-Version ermitteln	31 Diskettenparameter ermitteln
13 Plattensystem zurücksetzen	32 Benutzernummer verwalten
14 Bezugslaufwerk festlegen	33 Wahlfrei lesen
15 Datei eröffnen	34 Wahlfrei schreiben
16 Datei schließen	35 Dateigröße ermitteln
17 Ersten Eintrag suchen	36 Random-Satz-Position setzen
18 Nächsten Eintrag suchen	37 Laufwerk zurücksetzen
40 Nullauffüllen beim wahlfreien Schreiben	

(Die Funktionen 28 und 32 sollten von Anwendungsprogrammen nicht benutzt werden, um Aufwärtskompatibilität mit CP/M zu behalten.)

Beim Eingang in ein transientes Programm sorgt CCP dafür, daß der Stapelzeiger auf einen Achtebenenstapel zeigt, mit der CCP-Rückkehradresse am Anfang. Damit verbleiben vor einem evtl. Überlauf sieben Ebenen zur Verfügung.

Obwohl dieser Stapel normalerweise von einem transienten Programm nicht verwendet wird (d. h. die meisten nichtresidenten Programme kehren zum CCP über einen Sprung zu Adresse 0000H zurück), reicht er für CP/M-Systemaufrufe aus, da zu einem lokalen Stapel am Systemeingang verzweigt wird. Der nachstehende Ausschnitt eines Programmbeispiels in Assemblersprache zeigt das kontinuierliche Einlesen von Zeichen, bis ein Stern festgestellt wird. In diesem Augenblick soll die Kontrolle an CCP zurückgegeben werden (es sei ein Standard CP/M-System mit BOOT=0000H vorausgesetzt).

```

BDOS    EQU    0005H    ;STANDARD CP/M-EINGANG
CONIN   EQU    1       ;KONSOLEINGABE-FUNKTION
;
NEXTC:  ORG    0100H    ;BASIS DER TPA
        MVI    C,CONIN ;NÄCHSTES ZEICHEN LESEN
        CALL  BDOS     ;ZEICHENRÜCKGABE NACH <A>
        CPI    '*'     ;ENDE DER VERARBEITUNG?
        JNZ   NEXTC   ;FALLS BED. NICHT ERFÜLLT, VERZWEIGUNG
        RET                ;RÜCKKEHR ZUM CCP
        END

```

CP/M implementiert eine namensorientierte Dateistruktur auf jeder Platte und stellt damit eine logische Organisation zur Verfügung, die jeder einzelnen Datei erlaubt, eine beliebige Anzahl von Eintragungen zu beinhalten, angefangen bei der völlig leeren Datei, bis zur vollen Plattenkapazität. Jedes Laufwerk ist von den anderen logisch zu unterscheiden und enthält ein eigenes Directory und einen eigenen Datei-Datenbereich. Die Namen der Plattendateien bestehen aus drei Teilen: Dem Code zur Auswahl des Laufwerks, dem aus bis zu acht Zeichen bestehenden Dateinamen (keines der Zeichen darf ein Leerzeichen sein) und dem Dateityp, der aus bis zu drei Zeichen (kein Leerzeichen) zusammengesetzt ist. Der Dateityp benennt die Entstehungskategorie einer bestimmten Datei, während der Dateiname die Dateien innerhalb einer Klasse voneinander unterscheidet. Die nachstehend aufgelisteten Dateitypen bezeichnen einige Kategorien, die sich eingebürgert haben, obgleich sie bis zu einem gewissen Grad willkürlich sind.

ASM	Assembler Quelldatei	PLI	PLI-Quellprogramm
PRN	Drucker Listing	HEX	HEX-Maschinencode
TEX	Quelldatei für Text-formatierung	BAK	Sicherung der ED-Originaldatei
BAS	Basic-Quellprogramm	REL	Verschiebbares Modul
INT	Zwischencode	SYM	SID Symboldatei
COM	Befehlsdatei	\$\$\$	Hilfsdatei

Quelldateien werden als Folge von ASCII-Zeichen behandelt, wobei jeder „Zeile“ der Quelldatei eine Zeichenfolge <CR> <LF> (0DH gefolgt von 0AH) nachgeht. Damit kann ein 128 Byte langer CP/M-Eintrag mehrere Quelltextzeilen enthalten. Das Ende einer ASCII-Datei wird mittels des ctl-Z-Zeichens (1AH) gekennzeichnet oder einem realen End-Of-File, geliefert von einer CP/M-Leseoperation. Innerhalb einer Maschinencoddatei bleiben jedoch ctl-Z-Zeichen ohne Wirkung und die Leseoperation wird in diesem Fall durch die von CP/M festgelegte Dateiende-Bedingung abgeschlossen.

Dateien in CP/M können gedacht werden als Folge von bis zu 65536 Einträgen zu je 128 Bytes Länge, die von 0 bis 65535 nummeriert sind und so 8 Megabytes pro Datei erlauben. Der Benutzer sollte jedoch beachten, daß die Einträge, logisch zusammenhängend gedacht, physikalisch im Plattenbereich nicht unbedingt zusammenhängend untergebracht sind. Intern sind alle Dateien in 16K Byte-Segmente aufgeteilt, die „logische Extents“ genannt werden. Damit können leicht 8-bit-Werte als Zähler eingeführt werden. Die Aufteilung in Extents wird in den folgenden Paragraphen erläutert; für den Programmierer ist sie nicht besonders bedeutsam, da zu jedem Extent sowohl sequentiell als auch unmittelbar zugegriffen werden kann.

Bei den Datei-Operationen, die mit Funktionsnummer 15 beginnen, adressiert DE gewöhnlich einen Dateibesreiber (FCB). Transiente Programme benutzen für einfache Dateioperationen häufig den von CP/M an der Stelle BOOT+005CH (normalerweise 005CH) reservierten FCB-Bereich mit vorbesetztem (Default) Inhalt. Die Grundeinheit der Dateiinformation ist ein 128 Byte langer Eintrag, wie er von allen Dateioperationen benutzt wird; damit wird bei Adresse BOOT+0080H (normalerweise 0080H) von CP/M eine Defaultadresse für Platten-I/O zur Verfügung gestellt, die zugleich die Anfangs-DMA-Adresse ist (siehe Funktion 26). Alle das Directory betreffenden Operationen finden in einem reservierten Bereich statt, der keinen Einfluß auf die Schreibpuffer hat, wie dies in Rel. 1 der Fall war, mit Ausnahme der Funktionen „Ersten Eintrag suchen“ und „Nächsten Eintrag suchen“, wo Kompatibilität gefordert ist.

Der FCB-Bereich besteht aus einer Folge von 33 Bytes für sequentiellen Zugriff und einer Reihe von 36 Bytes für den Fall, daß wahlfrei auf die Datei zugegriffen werden soll. Der vorbesetzte (Default) FCB, normalerweise bei 005CH beginnend, kann für Dateien mit wahlfreiem Zugriff benutzt werden, da die drei bei BOOT+ 007DH beginnenden Bytes für diesen Zweck zur Verfügung stehen. Das Format des Dateibesreibers (FCB) ist nachstehend gezeigt:

dr	f1	f2	/	/	f8	t1	t2	t3	ex	s1	s2	rc	d0	/	/	dn	cr	r0	r1	r2
00	01	02	...	08	09	10	11	12	13	14	15	16	...	31	32	33	34	35		

wobei

dr	Laufwerks-Code (0-16) 0 => das Default-Laufwerk soll f. Datei ben. werden 1 => Auswahl von Laufwerk A 2 => Auswahl von Laufwerk B ... 16 => Auswahl von Laufwerk P
f1...f8	enthalten Dateinamen in ASCII Großschreibung, höchstes Bit auf Null gesetzt
t1,t2,t3	enthalten den Dateityp in ASCII Großschreibung. Die höherwertigen Bits werden folgendermaßen benutzt: t1' = 1 => Schreibgeschützte Datei t2' = 1 => SYS Datei, keine Auflistung mit DIR
ex	enthält die aktuelle Extentnummer, vom Benutzer normalerweise auf den Wert 00 gesetzt, aber während Datei I/O im Bereich 0-31
s1	Reserviert für internen Systemgebrauch
s2	reserviert für internen Systemgebrauch, auf Null gesetzt nach Aufruf von OPEN, MAKE, SEARCH
rc	Satzzähler für Extent „ex“, nimmt Werte an von 0 - 127
d0...dn	von CP/M für internen Systemgebrauch belegt
cr	Laufende Nummer eines zu lesenden oder zu schreibenden Satzes in einer sequentiellen Dateioperation, normalerweise auf Null gesetzt
r0,r1,r2	optionale Satznummer bei Direktzugriff im Bereich 0 - 65535, Überlauf geht nach r2. r0, r1 bilden einen 16 Bit-Wert mit niederwertigem Byte r0, höherwertigem Byte r1

Jede Datei, zu der über CP/M zugegriffen werden soll, muß einen zugehörigen FCB haben, der Namen und Zuweisungsinformationen für alle nachfolgenden Dateioperationen enthält. Wenn zu Dateien zugegriffen wird, ist es Aufgabe des Programmierers, die unteren 16 Bytes des FCB zu füllen und das cr-Feld zu initialisieren. Normaler-

weise werden die Bytes 1 bis 11 auf den Wert der ASCII-Zeichen für Dateiname und Dateityp gesetzt, während die anderen Felder auf Null gesetzt werden.

Die Dateibesreiber sind im Directorybereich der Platte gespeichert und werden in den Zentralspeicher gebracht, bevor der Programmierer Dateioperationen durchführt (siehe die Funktionen 25 und 22). Die Speicherkopie des FCB wird aktualisiert, wenn die Dateioperation durchgeführt, und nach Beendigung wieder auf die Platte geschrieben wird (siehe Funktion 16).

Der CCP konstruiert die ersten 16 Bytes von zwei optionalen FCB's für ein transientes Kommando durch Abtasten des Restes der Kommandozeile, der auf den Befehlsnamen folgt und in der oben beschriebenen Prototypbefehlszeile mit Datei 1 und Datei 2 bezeichnet ist. Nicht spezifizierte Felder erhalten den ASCII-Wert " ". Der erste FCB wird bei der Adresse BOOT+005CH konstruiert und kann, wie er ist, für nachfolgende Dateioperationen benutzt werden. Der zweite FCB besetzt den d0 . . . dn-Teil des ersten FCB und muß in einen anderen Speicherbereich verschoben werden, bevor er benutzt werden kann. Gibt der Operator z. B. ein

#### PROGNAME B:X.ZOT Y.ZAP

so wird die Datei PROGNAME.COM in die TPA geladen und der Default-FCB an der Stelle BOOT+005CH wird initialisiert mit Laufwerk-Code 2, Dateiname X und Dateityp ZOT. Der zweite Laufwerk-Code nimmt den Defaultwert 0 an, der in Speicherstelle BOOT+006DH abgelegt wird und der Dateityp ZAP wird 8 Bytes später an der Stelle BOOT+0075H abgelegt. Alle verbleibenden Felder bis cr werden auf den Wert Null gesetzt. Der Benutzer sollte wieder beachten, daß es in der Verantwortlichkeit des Programmierers liegt, den zweiten Dateinamen und Dateityp in einen anderen Bereich zu verschieben, normalerweise in einen eigenen Dateibesreiber, bevor die Datei, die bei BOOT+005CH beginnt, eröffnet wird. Andernfalls würde die OPEN-Operation (Funktion 15) den zweiten Dateinamen und Dateityp überschreiben.

Falls im Originalbefehl keine Dateinamen angegeben sind, enthalten die Felder, die bei BOOT+005DH und BOOT+006DH beginnen, Leerzeichen. In allen Fällen übersetzt CCP Kleinbuchstaben in Großschreibung, um mit den CP/M-Konventionen für Dateinamen konsistent zu sein.

Als zusätzliche Unterstützung wird der automatisch zugeordnete (Default-)Pufferbereich beginnend an der Stelle BOOT+0080H mit dem Rest der Befehlszeile initialisiert, der vom Bediener, auf den

Programmnamen folgend, eingegeben wurde. Die erste Position enthält die Zeichenanzahl, daran schließen sich die Zeichen selbst an. Ist die obige Befehlszeile vorgegeben, so wird der Speicherbereich bei BOOT+0080H wie folgt initialisiert:

BOOT+0080H:

```
+00 +01 +02 +03 +04 +05 +06 +07 +08 +09 +A +B +C +D +E  
E   " 'B' ':' 'X' ':' 'Z' 'O' 'T' " 'Y' ':' 'Z' 'A' 'P'
```

Die Zeichen werden in ASCII Großschreibung übersetzt. Nach dem letzten gültigen Zeichen wird der Speicher nicht initialisiert. Es ist wieder Sache des Programmierers, die gebrauchten Informationen aus dem Puffer zu holen, bevor irgendwelche Dateioperationen durchgeführt werden, falls nicht die vorbesetzten DMA-Adressen explizit geändert werden.

Auf den folgenden Seiten werden die einzelnen Funktionen im Detail erläutert.

Funktion 0: Warmstart des Systems

Eingangsparameter:

Register C: 00H

Die Warmstart-Funktion gibt die Kontrolle an CP/M zurück auf CCP-Ebene. Der CCP initialisiert das Plattensubsystem neu, indem die Platte A ausgewählt und eingeloggt wird. Die Funktion hat genau die gleiche Wirkung wie ein Sprung zu BOOT.

Funktion 1: Konsoleingabe

Eingangsparameter:

— Register C: 01H

Zurückgelieferter Wert:

Register A: ASCII-Zeichen

Die Konsoleingabefunktion liest das nächste Zeichen von der Konsole nach Register A. Die Zeichen werden an der Konsole zusammen mit CR, LF und Back Space (ctl-H) an der Konsole wiederholt. Tabulationszeichen (ctl-I) verschieben den Cursor zum nächsten Tabulationshalt. Eine Prüfung auf ctl-S und ctl-P wird durchgeführt. Das FDOS gibt die Kontrolle nicht an das rufende Programm zurück,

solange ein Zeichen nicht eingegeben wird. Somit wird die Ausführung ausgesetzt, falls ein Zeichen nicht vorliegt.

Funktion 2: Konsolausgabe

Eingangsparameter:

Register C: 02H

Register E: ASCII-Zeichen

Das ASCII-Zeichen wird von Register E zum Konsolgerät gesendet. Wie in Funktion 1 werden Tabulatorspalten definiert und entsprechende Prüfungen für Drucker-Wiederholung usw. durchgeführt.

Funktion 3: Lochstreifen-Eingabe

Eingangsparameter:

Register C: 03H

Zurückgelieferter Wert:

Register A: ASCII-Zeichen

Die Funktion Lochstreifen-Eingabe liest das nächste Zeichen vom logischen Leser nach Register A. Die Kontrolle wird nicht zurückgegeben, solange ein Zeichen nicht vorliegt.

Funktion 4: Lochstreifen Stanzen

Eingangsparameter:

Register C: 04H

Register E: ASCII-Zeichen

Die Funktion Lochstreifen-Ausgabe sendet das Zeichen vom Register E zur logischen Stanz-Einheit.

Funktion 5: Zeichen an Drucker ausgeben

Eingangsparameter:

Register C: 05H

Register E: ASCII-Zeichen

Die Funktion „Zeichen an Drucker ausgeben“ sendet das ASCII-Zeichen in Register E zum Ausgabegerät.

**Funktion 6: Direkte Konsolin- u. -ausgabe**

**Eingangsparameter:**

Register C: 06H  
Register E: 0FFH (Eingabe) oder  
Zeichen (Ausgabe)

**Zurückgelieferter Wert:**

Register A: Zeichen oder Status

Direkte Konsolin- und -ausgabe wird unter CP/M für jene Spezialanwendungen unterstützt, bei denen zeichenweise Ein- und Ausgabe auf unterster Ebene verlangt wird. Die Verwendung dieser Funktion sollte im allgemeinen vermieden werden, da sie alle normalen Steuerzeichen-Funktionen von CP/M (z. B. ctl-S und ctl-P) umgeht. Programme, die unter früheren Versionen von CP/M direkte Ein- und Ausgabe mit Hilfe von BIOS-Programmteilen ausführen, sollten so geändert werden, daß sie auf direkte Ein- und Ausgabe unter BDOS zurückgreifen, damit sie auch unter künftigen Versionen von MP/M und CP/M voll unterstützt werden können.

Beim Eintritt in Funktion 6 enthält Register E entweder den Hexadezimalwert FF, womit Konsoleingabe angefordert wird, oder ein ASCII-Zeichen. Ist der Eingabewert FF, so liefert Funktion 6 den Wert 0 in Register A, falls kein Zeichen bereit ist, andernfalls enthält A das nächste Konsoleingabezeichen.

Ist der Eingabewert in E nicht FF, so nimmt Funktion 6 an, daß E ein gültiges ASCII-Zeichen enthält, das an der Konsole ausgegeben werden soll. Funktion 6 darf nicht in Verbindung mit anderen I/O-Funktionen verwendet werden.

**Funktion 7: IOBYTE abfragen**

**Eingangsparameter:**

Register C: 07H

**Zurückgelieferter Wert:**

Register A: Wert des IOBYTE

Die Funktion „IOBYTE abfragen“ liefert den derzeitigen Wert des IOBYTE nach Register A.

Funktion 8: IOBYTE setzen

Eingangsparameter:

Register C: 08H

Register E: Wert des IOBYTE

Die Funktion „IOBYTE setzen“ ändert den Wert des IOBYTE wie in Register E vorgegeben.

Funktion 9: Zeichenkette ausgeben

Eingangsparameter:

Register C: 09H

Register DE: Adresse der Zeichenkette

Die Funktion „Zeichenkette ausgeben“ sendet die Zeichenkette, die im Speicher an der Stelle abgelegt ist, die durch den Inhalt des Registerpaars DE adressiert ist, an das Konsolgerät. Die Ausgabe endet, sobald ein \$ in der Zeichenkette vorgefunden wird. Tabulatorspalten werden wie in Funktion 2 definiert, Start/Stop-Möglichkeit ist gegeben und die Anforderung einer Druckausgabe wird erkannt.

Funktion 10: Eingabe in Puffer übernehmen

Eingangsparameter:

Register C: 0AH

Register DE: Pufferadresse

Zurückgelieferter Wert:

Konsolzeichen im Puffer

Die Funktion „Eingabe in Puffer übernehmen“ liest eine Zeile aufbereiteter Konsoleingabe in einen Puffer, dessen Adresse in DE abgelegt ist. Die Konsoleingabe wird beendet entweder durch einen Pufferüberlauf oder wenn <CR> oder <LF> eingegeben wird. Der Lesepuffer ist von der Form

DE: +0 +1 +2 +3 +4 +5 +6 +7 +8 ... +n

mx	nc	c1	c2	c3	c4	c5	c6	c7	...	??
----	----	----	----	----	----	----	----	----	-----	----

Dabei ist mx die maximale Zeichenzahl, die der Puffer aufnehmen kann (1 bis 255) und nc ist die Anzahl der gelesenen Zeichen (von FDOS bei der Rückgabe gesetzt), gefolgt von den Zeichen, die an der Konsole eingegeben werden. Falls  $nc < mx$ , dann folgen auf das letzte Zeichen nicht definierte Stellen, in obiger Abbildung mit ?? bezeichnet. Einige Steuerzeichen werden während der Zeilenaufbereitung ggf. erkannt.

rub/del	löscht das letzte Zeichen und zeigt es an der Konsole an
ctl-C	Am Beginn einer Zeile eingegeben, löst es Warmstart aus
ctl-E	markiert physikalisches Zeilenende
ctl-H	Rücksetzung um eine Zeichenposition
ctl-J	(line feed) schließt Eingabezeile ab
ctl-M	(carriage return) schließt Eingabezeile ab
ctl-R	gibt laufende Zeile nach der neuen Zeile aus
ctl-U	löscht aktuelle Zeile nach Eingabe der neuen Zeile
ctl-X	Rücksprung an den Beginn der aktuellen Zeile

Sie sollten beachten, daß gewisse Funktionen, die den Cursor an die äußerste linke Position schicken (z. B. ctl-X), dies nur bis zu der Stelle, an der der Prompt endet, tun. Dies macht Dateneingabe und Zeilenkorrektur leichter lesbar.

#### Funktion 11: Abfragen des Konsolstatus

Eingangsparameter:

Register C: 0BH

Zurückgelieferter Wert:

Register A: Konsolstatus

Die Konsolstatusfunktion prüft, ob an der Konsole ein Zeichen eingegeben worden ist. Ist ein Zeichen bereit, so wird an Register A

ein Wert > 00H übergeben. Andernfalls wird der Wert 00H übergeben.

Funktion 12: Ermitteln der Versionsnummer von CP/M

Eingangsparameter:

Register C: 0CH

Zurückgelieferter Wert:

Register HL: Versionsnummer

Funktion 12 stellt Informationen bereit, die versionsunabhängiges Programmieren erlauben. Ein Zwei-Byte-Wert wird zurückgegeben, wobei mit H = 00 eine CP/M Version gekennzeichnet wird (H = 01 für MP/M), und zwar ist L = 00 für alle Vorgängerversionen von CP/M 2.0.

CP/M 2.0 gibt den Hexadezimalwert 20 nach Register L zurück, spätere Freigaben von Version 2 geben entsprechend 21, 22, ... 2F zurück.

Bei Verwendung von Funktion 12 kann der Benutzer z. B. Anwendungsprogramme schreiben, die Funktionen für sowohl sequentiellen als auch direkten Zugriff bereitstellen.

Funktion 13: Plattensystem zurücksetzen

Eingangsparameter:

Register C: 0DH

Diese Funktion dient dazu, vom Programm her das Dateisystem so in einen Grundzustand zurückzusetzen, daß der Schreibschutz bei allen Laufwerken aufgehoben ist (siehe Funktionen 28 und 29), daß nur Laufwerk A ausgewählt ist und die vordefinierte DMA-Adresse auf BOOT+0080H festliegt. Diese Funktion kann z. B. bei einem Anwenderprogramm verwendet werden, das einen Plattenwechsel ohne Warmstart fordert.

#### Funktion 14: Laufwerk festlegen

##### Eingangsparameter:

Register C: 0EH  
Register E: Laufwerknummer

Diese Funktion legt das Laufwerk, das in Register E angegeben ist, als Bezugslaufwerk für nachfolgende Dateioperationen fest, mit E = 0 für Laufwerk A, 1 für Laufwerk B, und so weiter bis 15, das dem Laufwerk P in einem vollen 16-Laufwerk-System entspricht. Das Laufwerk wird in den On-Line Status gesetzt, wodurch sein Directory aktiviert wird und aktiv bleibt, bis ein Kaltstart, ein Warmstart oder ein Rücksetzen des Plattensystems erfolgt.

Wird der Datenträger gewechselt, während das Laufwerk On-Line ist, dann geht dieses in einer Standard CP/M-Umgebung in den Zustand „schreibgeschützt“ über (siehe Funktion 28). Dateibesreiber, die den Laufwerkscode Null angeben (dr = 00H), beziehen sich automatisch auf das gegenwärtig gültige Bezugslaufwerk. Werte des Laufwerkscodes zwischen 1 und 16 ignorieren den festgelegten Bezug und beziehen sich auf Laufwerk A bis P.

#### Funktion 15: Datei eröffnen

##### Eingangsparameter:

Register C: 0FH  
Register DE: FCB Adresse

##### Zurückgelieferter Wert:

Register A: Directory Code

Diese Operation dient dazu, eine Datei, die derzeit im Platten-Directory existiert, für die momentan aktuelle Benutzernummer zu aktivieren. FDOS geht das angesprochene Directory durch und vergleicht mit den Positionen 1 bis 14 des FCB, der durch das Registerpaar DE festgelegt ist (Byte s1 ist automatisch auf Null gesetzt). Ein ASCII-Fragezeichen (3FH) paßt zu jedem Zeichen des Directory in einer dieser Positionen. Normalerweise sind keine Fragezeichen enthalten, und die Bytes ex und s2 des FCB sind Null. Wird Übereinstimmung mit einem Directory-Element angetroffen, wird die relevante Directory-Information in die Bytes d0 bis dn des FCB kopiert, wodurch der Zugriff für nachfolgende Lese- und Schreiboperationen

ermöglicht wird. Der Programmierer sollte beachten, daß zu einer bestehenden Datei nicht zugegriffen werden darf, bevor ein erfolgreiches Öffnen stattgefunden hat.

Nach Rückkehr liefert die Funktion einen Directory Code mit dem Wert 0 bis 3, falls das Eröffnen erfolgreich war, oder 0FFH (255 dezimal) falls die Datei nicht gefunden werden konnte. Treten im FCB Fragezeichen auf, wird der erste passende FCB aktiviert. Man beachte, daß der laufende Eintrag (current record = cr) durch das Programm auf Null gesetzt werden muß, falls der Zugriff zur Datei vom Eintrag an sequentiell erfolgen muß.

Funktion 16: Datei schließen

Eingangsparameter:

Register C: 10H

Register DE: FCB Adresse

Zurückgelieferter Wert:

Register A: Directory Code

Die Funktion „Datei schließen“ bewirkt das Gegenteil der Funktion „Datei eröffnen“. Geht man davon aus, daß der durch DE adressierte FCB früher durch Funktion 15 oder 22 aktiviert worden ist, so trägt Funktion 16 den neuen FCB in das angesprochene Directory ein. Der FCB-Vergleichsvorgang für das Schließen ist identisch dem der Funktion „Öffnen“. Der Directory Code, der bei erfolgreichem Schließen zurückgegeben wird, ist 0, 1, 2 oder 3, während ein 0FFH (255 Dezimal) zurückgegeben wird, falls die Datei in dem Directory nicht gefunden werden kann.

Eine Datei braucht nicht geschlossen zu werden, falls nur Leseoperationen vorgenommen worden sind. Falls jedoch Schreiboperationen vorgenommen worden sind, ist das Schließen notwendig, um die neue Directory-Information permanent einzutragen.

**Funktion 17: Den ersten Eintrag suchen**

**Eingangsparameter:**

Register C: 11H  
Register DE: FCB Adresse

**Zurückgelieferter Wert:**

Register A: Directory Code

Diese Funktion durchsucht das Directory nach Übereinstimmung mit der Datei, die durch den durch DE adressierten FCB gegeben ist. Der Wert 255 (hexadezimal FF) wird zurückgegeben, falls die Datei nicht gefunden wird. Andernfalls wird 0, 1, 2 oder 3 zurückgegeben, womit angezeigt wird, daß die Datei vorhanden ist. Falls die Datei gefunden wird, so wird die laufende DMA-Adresse mit dem Eintrag gefüllt, der die Directory-Aufzeichnung enthält, und die relative Startposition ergibt sich zu  $A * 32$  (d. h. Rotieren des A-Registers um 5 Positionen nach links, oder fünfmal ADD A).

Obwohl normalerweise für Anwendungsprogramme nicht benötigt, kann die Directory-Information aus dem Puffer an dieser Position entnommen werden.

Ein ASCII-Fragezeichen (63 Dezimal, 3F hexadezimal) paßt in jeder Position von fl bis ex zum entsprechenden Feld eines jeden Directory-Eintrags auf Bezugs- oder ausgewähltem Laufwerk. Enthält das dr-Feld ein ASCII-Fragezeichen, so wird die Auswahlfunktion außer Kraft gesetzt und das Bezugslaufwerk wird durchsucht, wobei die Suchfunktion jeden passenden Eintrag liefert, zugeordnet oder frei, der zu irgendeiner Benutzernummer gehört.

Diese letztere Funktion wird normalerweise von Anwenderprogrammen nicht genutzt, aber sie erlaubt völlige Freiheit, alle aktuellen Directory-Werte durchzugehen. Falls das dr-Feld kein Fragezeichen ist, wird das s2 Byte automatisch auf Null gesetzt.

**Funktion 18: Den nächsten Eintrag suchen**

**Eingangsparameter:**

Register C: 12H

**Zurückgelieferter Wert:**

Register A: Directory Code

Die Funktion 18 ist ähnlich der Funktion 17, außer daß das Durchsuchen des Directory vom letzten passenden Eintrag an fortgesetzt wird. Wie Funktion 17 gibt auch Funktion 18 den Dezimalwert 255 in A zurück, falls keine weiteren Directory-Elemente passen.

Funktion 19: Datei löschen

Eingangsparameter:

Register C: 13H  
Register DE: FCB Adresse

Zurückgelieferter Wert:

Register A: Directory Code

Die Funktion „Datei löschen“ entfernt Dateien, die zum FCB, der durch das Registerpaar DE adressiert ist, passen. Dateiname und Typ können mehrdeutige Bezüge (d. h. Fragezeichen in verschiedenen Positionen) enthalten, aber der Laufwerkscode kann nicht mehrdeutig sein, wie bei den „Suche-“ und „Suche nächsten Eintrag“-Funktionen.

Die Funktion 19 gibt einen Dezimalwert 255 zurück, falls die angesprochene Datei oder die Dateien nicht gefunden werden konnte(n); andernfalls wird ein Wert im Bereich 0 bis 3 zurückgegeben.

Funktion 20: Lesen sequentiell

Eingangsparameter:

Register C: 14H  
Register DE: FCB Adresse

Zurückgelieferter Wert:

Register A: Directory Code

Setzt man voraus, daß der durch DE adressierte FCB durch Funktion 15 oder 22 aktiviert worden ist, dann liest die Funktion 20 den nächsten 128 Byte langen Eintrag von der Datei in den Speicher bei der aktuellen DMA Adresse. Der Eintrag wird von Position cr des Extent an gelesen und das cr-Feld wird automatisch auf die nächste Eintragsposition erhöht. Falls das cr-Feld überläuft, wird der nächste logische Extent automatisch eröffnet und das cr-Feld wird als Vorbereitung für die nächste Lese-Operation auf Null zurückgesetzt. Der

Wert 00H wird nach Register A zurückgegeben, falls die Leseoperation erfolgreich war, während ein von Null abweichender Wert zurückgegeben wird, falls an der nächsten Eintragsposition keine Daten existieren (z. B. Dateieinde erscheint).

Funktion 21: Schreiben sequentiell

Eingangsparameter:

Register C: 15H  
Register DE: FCB Adresse

Zurückgelieferter Wert:

Register A: Directory Code

Setzt man voraus, daß der durch DE adressierte FCB durch Funktion 15 oder 22 aktiviert worden ist, dann schreibt die Funktion 21 den 128-Byte langen Datensatz an der aktuellen DMA-Adresse in die durch den FCB bezeichnete Datei. Der Satz wird bei Position cr der Datei abgelegt und das cr-Feld wird automatisch auf die nächste Satzposition erhöht. Falls das cr-Feld überläuft, wird automatisch der nächste logische Extent eröffnet und das cr-Feld wird als Vorbereitung für die nächste Schreiboperation auf Null zurückgesetzt. Schreiboperationen sind in existierende Dateien möglich. In diesem Fall überlagern die neu geschriebenen Sätze jene, die bereits in der Datei existieren. Register A hat den Inhalt 00H nach Rückkehr von einer erfolgreichen Schreiboperation, während ein von Null abweichender Wert anzeigt, daß eine volle Platte ein erfolgreiches Schreiben verhindert hat.

Funktion 22: Datei anlegen

Eingangsparameter:

Register C: 16H  
Register DE: FCB Adresse

Zurückgelieferter Wert:

Register A: Directory Code

Die Operation „Datei anlegen“ ist ähnlich der Operation „Datei eröffnen“, außer daß der FCB eine Datei benennen muß, die im momentan angesprochenen Plattenverzeichnis nicht existiert. Dieses

Plattenverzeichnis kann das Plattenverzeichnis des gegenwärtigen Bezugslaufwerks ( $dr = 0$ ) oder eines anderen Laufwerks ( $dr <> 0$ ) sein. FDOS legt die Datei an und initialisiert sowohl das Directory als auch den Hauptspeicherwert für eine leere Datei. Der Programmierer muß dafür sorgen, daß keine doppelten Dateinamen auftreten und eine vorausgehende Lösch-Operation genügt, falls irgendeine Möglichkeit eines doppelten Auftretens besteht. Bei der Rückgabe erhält Register A den Wert 0, 1, 2 oder 3 falls die Operation erfolgreich war und 0FFH (255 dezimal), falls in dem Directory kein Raum mehr verfügbar ist. Die Funktion „Datei anlegen“ hat den Nebeneffekt der Aktivierung des FCB und daher ist ein nachfolgendes Öffnen nicht notwendig.

Funktion 23: Datei umbenennen

Eingangsparameter:

Register C: 17H  
Register DE: FCB Adresse

Zurückgelieferter Wert:

Register A: Directory Code

Diese Funktion verwendet den durch DE adressierten FCB, um überall den Dateinamen, der in den ersten 16 Bytes abgelegt ist, durch den Dateinamen zu ersetzen, der in den zweiten 16 Bytes abgelegt ist. Der Laufwerks-Code  $dr$  an Position 0 dient dazu, das Laufwerk auszuwählen, während der Laufwerks-Code an Position 16 des FCB für die neue Datei als 0 angenommen wird. Bei der Rückkehr erhält Register A einen der Werte von 0 bis 3, falls die Umbenennung erfolgreich war und 0FFH (255 dezimal), falls der erste Dateiname beim Durchsuchen des Plattenverzeichnisses nicht gefunden werden konnte.

Funktion 24: Aktive Laufwerke ermitteln

Eingangsparameter:

Register C: 18H

Zurückgelieferter Wert:

Register HL: Log-in Vector

Der Wert des log-in Vektors, der von CP/M zurückgegeben wird, ist ein 16-Bit-Wert in HL, wo das niederwertige Bit von Register L zum ersten Laufwerk A gehört und das höherwertige Bit von H zum sechzehnten Laufwerk, genannt P. Ein 0-Bit zeigt an, daß das Laufwerk nicht aktiviert ist, während ein 1-Bit zum Ausdruck bringt, daß das zugehörige Laufwerk momentan aktiviert ist. Änderungen dieses Vektors werden entweder durch explizite Auswahl eines Laufwerks oder durch einen wegen einer Dateioperation von 0 abweichenden dr-Wert hervorgerufen. Der Benutzer sollte zur Kenntnis nehmen, daß Kompatibilität mit früheren Versionen erreicht wurde, da die Register A und L bei der Rückkehr die gleichen Werte erhalten.

Funktion 25: Bezugslaufwerk ermitteln

Eingangsparameter:

Register C: 19H

Zurückgelieferter Wert:

Register A: Nummern des  
Bezugslaufwerks

Funktion 25 gibt die Nummer der derzeit eingeloggtten Platte nach Register A zurück. Die Plattennummern reichen von 0 bis 15 und entsprechen den Laufwerken von A bis P.

Funktion 26: DMA-Adresse setzen

Eingangsparameter:

Register C: 1AH

Register DE: DMA-Adresse

DMA steht für Direct Memory Access (Direkter Speicherzugriff). DMA wird häufig in Verbindung mit Platten-Steuereinheiten benutzt, die direkt auf den Arbeitsspeicher des Computers zugreifen, um Daten zu oder von der Platte zu transferieren.

Bei CP/M bezeichnet die DMA-Adresse den Beginn eines Bereichs für einen 128-Byte Datensatz, der entweder auf die Platte geschrieben oder von dort gelesen wird. Beim Kalt- oder Warmstart und Zurücksetzen (reset) des Platten-Systems wird die DMA-Adresse automatisch auf BOOT+0080H gesetzt. Mit der DMA-

Funktion kann diese Standard-Adresse jedoch abgeändert werden, um einen anderen Bereich im Arbeitsspeicher für den oben genannten Zweck zu verwenden. Der DE-Wert wird danach solange als DMA-Adresse benutzt, bis eine erneute Änderung durch einen Warm- oder Kaltstart erfolgt oder das Plattensystem wieder zurückgesetzt wird.

Funktion 27: Belegungstabelle ermitteln

Eingangsparameter:

Register C: 1BH

Zurückgelieferter Wert:

Register HL: Zuordnungs-Adresse

Im Arbeitsspeicher wird ein Zuordnungsvektor (Tabelle) für jedes Plattenlaufwerk unterhalten. Verschiedene Systemprogramme benutzen die dort enthaltene Information, um den verfügbaren Speicherplatz zu ermitteln (siehe das STAT-Programm). Die Funktion 27 liefert die Basis-Adresse des Zuordnungsvektors für das gerade ausgewählte Laufwerk.

Die Information über die Zuordnung kann allerdings fehlerhaft sein, wenn die ausgewählte Platte nur als lesbar markiert wurde.

Diese Funktion wird normalerweise nicht bei Anwenderprogrammen benutzt. Kapitel 7 enthält weitere Details zum Zuordnungsvektor (ALV = Allocation Vector).

Funktion 28: Platte vor Schreiben schützen

Eingangsparameter:

Register C: 1CH

Diese Funktion bewirkt den zeitweiligen Schutz vor Beschreiben bei der gerade ausgewählten Platte. Jeder Versuch, die Platte vor dem nächsten Kalt- oder Warmstart doch zu beschreiben, führt zur Nachricht:

BDOS ERR on d: R/O

### Funktion 29: Geschütztes Laufwerk ermitteln

Eingangsparameter:

Register C: 1DH

Zurückgelieferter Wert:

Register HL: R/O Vektor Wert

Funktion 29 gibt den Bitvektor in Registerpaar HL wieder, der Laufwerke kennzeichnet, für die gerade nur das Lesen erlaubt ist (R/O d. h. „nur Lesen“-Bit ist gesetzt).

Wie bei Funktion 24 betrifft das niedrigstwertige Bit das Laufwerk A, während das höchstwertige Bit dem Laufwerk P zugeordnet ist.

Das R/O-Bit wird entweder über Funktion 28 oder durch die automatische Software-Operation von CP/M bei entdecktem Laufwerkswechsel gesetzt.

### Funktion 30: Dateimerkmale setzen

Eingangsparameter:

Register C: 1EH

Register DE: FCB-Adresse

Zurückgelieferter Wert:

Register A: Directory-Code

Die Funktion „Dateimerkmale setzen“ dient der Handhabung der ständigen Indikatoren, die den Dateien beigefügt sind. Insbesondere können die R/O- und System-Merkmale (t1' und t2') gesetzt oder gelöscht werden. Das DE-Paar adressiert einen eindeutigen Dateinamen mit geeigneten (gesetzten oder nicht-gesetzten) Merkmalen.

Funktion 30 prüft auf Übereinstimmung und ändert den betreffenden Directory-Eintrag entsprechend der gewählten Indikatoren ab.

Die Indikatoren f1' bis f4' werden zur Zeit nicht verwendet. Sie sind jedoch nützlich für Anwenderprogramme, da sie beim Öffnen und Schließen von Dateien nicht in den Abgleich mit einbezogen werden.

Die Indikatoren f5' bis f8' und t3' sind für zukünftige Systemerweiterungen reserviert.

**Funktion 31: Diskettenparameter ermitteln**

Eingangsparameter:

Register C: 1FH

Zurückgelieferter Wert:

Register HL: DPB-Adresse

Die Funktion 31 gibt die Adresse des in BIOS residenten Plattenparameter-Blocks in HL wieder. Diese Adresse kann für zwei Zwecke genutzt werden. So können die Werte der Parameter angezeigt oder für die Errechnung von Speicherplatz verwendet werden. Zum anderen können transiente Programme im Falle der Änderung der Plattenkonstellation die gegenwärtigen Plattenparameter abändern. Im Normalfall benötigen Anwenderprogramme diese Funktion nicht.

**Funktion 32: Benutzernummer verwalten**

Eingangsparameter:

Register C: 20H

Register E: 0FFH (ermitteln) oder  
Benutzernummer (setzen)

Zurückgelieferter Wert:

Register A: gegenwärtige Nummer  
oder (kein Wert)

Ein Anwenderprogramm kann mit dem Abruf der Funktion 32 die gegenwärtig aktive Benutzernummer ändern oder abfragen. Falls Register E = 0FFH, wird der Wert der gegenwärtigen Benutzernummer an Register A übermittelt. (Mögliche Werte sind 0 bis 15.) Falls Register E  $\neq$  0FFH, wird die gegenwärtige Benutzernummer geändert auf den Wert von E (Modulo 16).

### Funktion 33: Wahlfrei (random) lesen

Eingangsparameter:

Register C: 21H

Register DE: FCB-Adresse

Zurückgelieferter Wert:

Register A: Rückgabe-Code

Die Funktion 33 ist ähnlich aufgebaut wie die Funktion zum sequentiellen Lesen von Dateien bei früheren Versionen. Es besteht nur folgender Unterschied: Jetzt startet die Lese-Operation ab einer besonderen Satznummer, die bestimmt wird von einem 24-Bit Wert in einem 3-Byte Feld, das hinter dem FCB (file control block) angeordnet ist. (Byte-Positionen r0 bei 33, r1 bei 34 und r2 bei 35). Es ist dabei zu beachten, daß die Folge von 24 Bits wie folgt aufgebaut ist: das niedrigstwertige Byte an erster Stelle (r0), dann das mittlere Byte (r1) gefolgt von dem höchstwertigen Byte (r2).

CP/M macht keinen Gebrauch von Byte r2 mit Ausnahme der Funktion 35, die dieses Byte zur Errechnung der Kapazität einer Datei benutzt. Ein von 0 abweichender Wert in r2 würde den Überlauf einer Datei bedeuten.

Das Byte-Paar r0, r1 wird als Doppel-Byte oder „Wort“-Wert aufgefaßt. Es enthält die zum Lesen des Satzes erforderlichen Informationen. Dieser Wert deckt den Bereich von 0 bis 65535 ab und gestattet somit den Zugriff zu jedem Satz einer Datei mit einer Kapazität von 8 Mio Bytes.

Um eine Datei als Random-Datei zu benutzen, muß der Grund-Extent (Extent 0) zuerst eröffnet werden. Dabei ist es unerheblich, ob er irgendwelche Daten enthält oder nicht. Damit wird sichergestellt, daß die Datei ordnungsgemäß in dem Directory aufgeführt wird und beim Abruf von DIR angezeigt werden kann.

Die jeweilige Satznummer wird in dem für Sätze mit wahlfreiem Zugriff (random access) vorgesehenen Feld r0, r1 gespeichert und BDOS wird angesprochen, um den betreffenden Satz zu lesen.

Nach dem Abruf von BDOS enthält das Register A einen Fehler-Code, wie im folgenden dargestellt, oder den Wert 00, der anzeigt, daß die Operation erfolgreich abgeschlossen wurde. Im letzteren Fall enthält die letzte DMA-Adresse diejenige des Random-Satzes.

Es ist zu beachten, daß im Gegensatz zur sequentiellen Lese-Methode, die Satznummer nicht automatisch erhöht wird. Es ist deshalb

möglich, daß aufeinanderfolgende Lese-Operationen im Random-mode auf den gleichen Satz zugreifen können.

Bei jeder Lese-Operation mit wahlfreiem Zugriff werden der Extent und die Satzweite automatisch gesetzt. Daher kann die Datei ab der vorliegenden Position, die im wahlfreien Zugriff angesprochen wurde, sequentiell gelesen oder geschrieben werden.

Es sollte jedoch beachtet werden, daß in diesem Fall, der letzte im Random-Verfahren gelesene Satz beim Übergang auf das sequentielle Vorfahren erneut gelesen wird. Diese Wiederholung trifft ebenso beim Schreiben zu. Sequentielle Ein/Ausgabe-Operationen können natürlich dadurch erreicht werden, daß die Position der im Random-Verfahren gelesenen oder geschriebenen Sätze vom Programmierer jeweils erhöht wird.

Die Fehler-Codes, die beim wahlfreien Lesen im Register A zurückgegeben werden, sind nachstehend zusammengefaßt:

- 01 Lesen nicht geschriebener Daten
- 02 (keine Verwendung im Random-Betrieb)
- 03 der vorliegende Extent kann nicht geschlossen werden
- 04 Suchen eines ungeschriebenen Extents
- 05 (keine Verwendung im Lese-Betrieb)
- 06 Suchen über das physische Ende der Platte hinaus

Die Fehler-Codes 01 und 04 treten auf, wenn im wahlfreien Zugriff ein Block gelesen wird, der vorher nicht geschrieben wurde. Das gleiche trifft zu beim Zugriff auf einen nicht geschriebenen Extent.

Der Fehler-Code 03 kommt bei vorschriftsmäßigem Betrieb des Systems nicht vor. Wenn er trotzdem auftritt, ist der Fehler leicht durch erneutes Lesen oder durch die erneute Eröffnung des Extents 0 zu beseitigen. Dies gilt allerdings nur für eine Platte, die nicht schreibgeschützt ist.

In der CP/M Version 2.0 tritt der Fehler-Code 06 auf, falls der Inhalt von Byte r2 nicht Null ist.

Gewöhnlich basieren zurückgegebene Codes mit einem von Null abweichenden Wert auf fehlenden Daten. Der Wert 0 dagegen besagt in diesem Zusammenhang, daß die Operation erfolgreich abgeschlossen wurde.

Funktion 34: Wahlfrei (random) schreiben

Eingangsparameter:

Register C: 22H  
Register DE: FCB-Adresse

Zurückgelieferter Wert:

Register A: Rückgabe-Code

Das wahlfreie Schreiben wird ähnlich eingeleitet wie das entsprechende Lesen. Nur werden in diesem Fall die Daten ab der gegenwärtigen DMA-Adresse auf die Platte geschrieben.

Falls der Extent oder der Datenblock noch nicht zugeordnet wurde, erfolgt die erforderliche Zuordnung vor dem Schreiben.

Wie beim wahlfreien Lesen wird die Satznummer beim Schreiben nicht geändert. Die logische Extent-Nummer und die gegenwärtigen Satzpositionen des FCB werden entsprechend der Satznummer des zu schreibenden Satzes gesetzt. Sequentielle Lese- oder Schreib-Operationen können auf das wahlfreie Schreiben folgen. Auch hier ist zu bemerken, daß die beim wahlfreien Lesen beschriebene Wiederholung bei Betriebswechsel auftritt. Sequentielles Schreiben ist entsprechend durch einfache Erhöhung der Satz-Position möglich.

Zu beachten ist, daß beim wahlfreien Schreiben des letzten Satzes innerhalb eines Extents kein automatischer Extent-Übergang erfolgt wie das beim sequentiellen Betrieb der Fall ist.

Die Fehler-Codes die beim wahlfreien Schreiben verwendet werden, stimmen mit denen beim wahlfreien Lesen überein mit einer Ausnahme bei Code 05.

Dieser Code zeigt beim Schreiben an, daß ein neuer Extent wegen des Überlaufs des Directory nicht erstellt werden konnte.

Funktion 35: Dateigröße ermitteln

Eingangsparameter:

Register C: 23H  
Register DE: FCB-Adresse

Zurückgelieferter Wert: Feld für Random-Satz gesetzt

Zur Ermittlung der Größe einer Datei spricht das DE-Registerpaar einen FCB mit Random-Format (Bytes r0, r1 und r2 sind vorhanden) an. Der FCB enthält den eindeutigen Dateinamen, der bei der Suche im Directory benötigt wird.

Die Bytes für die Random-Sätze speichern danach die vom Directory übernommene „virtuelle“ Dateigröße, die der Adresse des Satzes nach dem Dateiende entspricht. Falls das höchstwertige Byte des FCB r2 den Wert 01 aufweist, besagt dies, daß die Datei die höchstmögliche Anzahl von 65536 Sätzen hat.

Sonst enthalten die Bytes r0 und r1 einen 16-Bit Wert, der die Dateigröße angibt. (r0 ist dabei das niedrigstwertige Byte wie zuvor.)

Mit Hilfe der Funktion 35 können Daten einer bestehenden Datei auf einfache Weise hinzugefügt werden. Die Position des Random-Satzes wird dabei auf das Dateiende gesetzt und es folgt eine Reihe von wahlfreien Schreib-Operationen beginnend mit der Adresse des vorgegebenen Satzes.

Bei einer sequentiell beschriebenen Datei stimmt ihre virtuelle Größe mit der physischen Größe überein. Im Fall einer Datei, die im wahlfreien Zugriff erstellt wurde, sind dagegen im allgemeinen Lücken in der Zuordnung vorhanden, was dazu führt, daß weniger Sätze in der Datei vorhanden sind als die nominelle Dateigröße besagt.

Wenn z. B. nur der letzte Satz einer 8 Mio-Byte Datei (Satznummer 65535) im Random-Verfahren geschrieben wird, wird als virtuelle Größe 65536 Sätze festgehalten, obwohl nur ein Satz zugeordnet wurde.

#### Funktion 36: Random-Satz setzen

Eingangsparameter:

Register C: 24H

Register DE: FCB-Adresse

Zurückgelieferter Wert: Feld für Random-Satz gesetzt

Diese Funktion veranlaßt BDOS, bei sequentiell gelesenen oder geschriebenen Dateien die gegenwärtige Satzposition zu errechnen. Die Funktion ist nützlich in den folgenden Fällen.

Es ist oft erforderlich, eine sequentielle Datei zu lesen und die Position von „Schlüssel“-Feldern zu ermitteln. Die Funktion 36 kann dazu verwendet werden, eine Tabelle für alle Schlüssel der Datei

aufzustellen. Mit Hilfe dieser Tabelle ist der wahlfreie Zugriff zu den jeden Schlüssel betreffenden Informationen möglich.

Dieses Verfahren kann auch für Sätze mit variabler Länge verwendet werden. Dafür muß lediglich die gewünschte Anfangsposition für jeden Satz zusätzlich gespeichert werden.

Die Funktion 36 kann auch beim Übergang vom sequentiellen zum Random-Lesen oder -Schreiben benutzt werden. Die Funktion hält dabei die Nummer des letzten Satzes fest, auf den sequentiell zugegriffen wurde.

Funktion 37: Laufwerk zurücksetzen

Eingangsparameter:

Register C: 25H

Register DE: Laufwerks-Tabelle

Zurückgelieferter Wert:

Register A: 00H

Diese Funktion erlaubt das Zurücksetzen einzelner Laufwerke. Der übermittelte Parameter ist ein 16-Bit Feld, das die Laufwerke bezeichnet, die zurückgesetzt werden sollen. Das geringstwertige Bit kennzeichnet Laufwerk A.

Um mit MP/M kompatibel zu sein, antwortet CP/M mit dem Wert 0.

Funktion 40: Nullauffüllen beim Schreiben

Eingangsparameter:

Register C: 28H

Register DE: FCB-Adresse

Zurückgelieferter Wert:

Register A: Rückgabe-Code

Diese Funktion arbeitet ähnlich wie Funktion 34 mit der Ausnahme, daß ein vorher nicht zugeordneter Block beim Random-Schreiben mit Nullen aufgefüllt wird, bevor die Daten geschrieben werden.

### 6.3 Beispiel für ein Datei-Kopierprogramm

Das unten gezeigte Programm stellt ein ziemlich einfaches Beispiel von Datei-Operationen dar. Das Quellprogramm wird unter der Bezeichnung COPY.ASM mit dem CP/M-Dienstprogramm ED erstellt und anschließend mit ASM oder MAC assembliert und in eine Hex-Datei umgewandelt. Mit LOAD wird die Datei in eine Datei COPY.COM umgesetzt. Diese kann dann unter CCP direkt aufgerufen und ausgeführt werden. Das Programm beginnt damit, daß der Stack-Pointer (Stapelzeiger) auf einen lokalen Bereich gesetzt wird und dann die zweite Bezeichnung vom automatisch zugeordneten (Default-)Bereich bei 006CH in einen 33-Byte Dateikontrollblock (DFCB) übertragen wird. DFCB wird dann durch Leeren des momentanen Satzfeldes für Datei-Operationen vorbereitet. Zu diesem Zeitpunkt sind die Dateikontrollblöcke (FCBs) von Ursprungs- und Zieldatei zur Ausführung bereit, da der SFCB auf 005CH vom CCP beim Eintritt in das Programm COPY richtig gesetzt wird. D. h. die erste Bezeichnung wird in den Default-FCB gesetzt, wobei die entsprechenden Felder auf Null gesetzt werden (inklusive das Feld des momentanen Satzes bei 007CH). Das Programm eröffnet dann die Ursprungsdatei, löscht, falls vorhanden, eine existierende Zieldatei und generiert die Zieldatei. Wenn dies alles erfolgreich durchgeführt ist, führt das Programm beim Bezeichner COPY solange eine Schleife aus, bis jeder Satz der Ursprungsdatei gelesen und in die Zieldatei geschrieben ist. Nach Beendigung der Datenübertragung wird die Zieldatei geschlossen und das Programm kehrt in den CCP Kommando-Modus zurück, indem es zu BOOT springt.

```

;          beispiel für ein dateikopierprogramm
;
;          das CCP-Kommando
;
;          copy a:x.y b:u.v
;
;          bewirkt, daß die datei x.y vom laufwerk a in
;          eine datei u.v. auf laufwerk b gebracht wird

0000 =      boot    equ 0000h    ;system neustart
0005 =      bdos    equ 0005h    ;bdos eintrittspunkt
005c =      fcb1    equ 005ch    ;erste dateibezeichnung
005c =      sfcbl   equ fcb1     ;ursprungs fcb
006c =      fcb2    equ 006ch    ;zweite dateibezeichnung
0080 =      dbuff   equ 0080h    ;vorgabepuffer
0100 =      tpa     equ 0100h    ;anfang des tpa
;
0009 =      printf  equ 9        ;drucker puffer funkt#
000f =      openf   equ 15       ;eröffne datei funkt#

```





Der Anwender muß beachten, daß dieses Programmbeispiel einige Vereinfachungen enthält. So wird zum Beispiel nicht geprüft, ob die Dateibezeichnungen eindeutig sind, d. h. keine Dateigruppensymbole enthalten. Dies könnte dadurch erreicht werden, daß der 32-Byte Default-Bereich ab 005CH nach „?“ und „\*“ abgefragt wird. Es sollte auch geprüft werden, ob tatsächlich Dateibezeichnungen eingegeben wurden (Prüfung der Positionen 005DH und 006DH auf Zeichen ungleich dem Leerzeichen). Schließlich wäre noch sicherzustellen, daß die Bezeichnungen von Ursprungs- und Zieldatei unterschiedlich sind. Eine Verbesserung der Geschwindigkeit des Programmes ist durch Pufferung von mehreren Daten bei jedem Lesevorgang zu erreichen. Man könnte z. B. die Speichergröße feststellen, indem man FBASE von Position 0006H ausliest und den ganzen Restspeicher als Datenpuffer benutzt. In diesem Falle bräucht der Programmierer nur die DMA-Adresse auf den nächstfolgenden 128-Byte-Bereich setzen, bevor jeweils gelesen wird. Beim Schreiben auf die Zieldatei wird die DMA-Adresse auf den Pufferanfang zurückgesetzt und nach jedem Schreibvorgang um 128 erhöht. Dies wird bis zum Ende des Puffers wiederholt.

#### 6.4 Ein einfaches Dienstprogramm zur hexadezimalen Ausgabe einer Datei (Dump)

Das Dateiausgabeprogramm, das unten gezeigt wird, ist etwas komplexer als das Kopierprogramm des vorangegangenen Kapitels. Es liest eine Eingabedatei, die im CCP-Kommando definiert ist und zeigt den Inhalt in hexadezimaler Form an der Konsole an. Beachten Sie, daß dieses Programm den CCP-Stack beim Eintritt speichert, den Stack auf einen lokalen Bereich setzt und den CCP-Stack zurückschreibt, bevor direkt in den CCP zurückgekehrt wird. Damit führt dieses Programm keinen Warmstart durch, wenn es die Ausführung beendet.

```

                                ;DUMP programm, das eine datei in hex-format anzeigt
                                ;
0100                                org 100h
0005 =                            bdos    equ 0005h    ;bdos eintrittspunkt
0001 =                            cons    equ 1        ;lies tastatur
0002 =                            typef   equ 2        ;zeige funktion an
0009 =                            printf  equ 9        ;puffere druckanfang
000b =                            brkf    equ 11       ;break-tasten funktion (wahr
                                ;                bei zeichen)
000f =                            openf   equ 15       ;datei offen
0014 =                            readf   equ 20       ;lese-funktion
                                ;

```

```

005c =      fcb      equ 5ch      ;dateikontrollblock-adresse
0080 =      buff     equ 80h      ;pufferadresse der eingabe-
                                   platte
                                   ;
                                   ;      nichtgrafische Zeichen
000d =      cr       equ 0dh      ;return
000a =      lf       equ 0ah      ;zeilenschaltung
                                   ;
                                   ;      definitionen des dateikontrollblocks
005c =      fcbdn    equ fcb+0    ;plattenbezeichnung
005d =      fcbfn    equ fcb+1    ;dateibezeichnung
0065 =      fcbft    equ fcb+9    ;dateityp (3 Zeichen)
0068 =      fcbfl    equ fcb+12   ;laufender extent der datei
006b =      fcbrc    equ fcb+15   ;satzzahl der datei (0 bis 128)
007c =      fcbcr    equ fcb+32   ;laufende (nächste) satznum-
                                   mer (0 bis 128)
007d =      fcblln   equ fcb+33   ;fcb-länge
                                   ;
                                   ;      stack setzen
0100 210000  lxi h,0
0103 39      dad sp
                                   ;      eingangs-stack pointer in hl vom ccp
0104 221502  shld oldsp
                                   ;      setze sp auf lokalen stackbereich (wird bei
                                   finis zurückgesetzt)
0107 315702  lxi sp,stktp
                                   ;      lese und zeige nachfolgende puffer an
010a cdc101  call setup      ;setze eingabedatei
010d feff    cpi 255      ;255 falls datei-offen nicht
                                   vorhanden
010f c21b01  jnz openok     ;überspringe falls dateieröff-
                                   nung o. k.
                                   ;datei nicht vorhanden, fehlernachricht und
                                   ;zurückkehren
0112 11f301  lxi d,opnmsg
0115 cd9c01  call err
0118 c35101  jmp finis      ;zurückkehren
                                   ;
openok:      ;eröffnung ok, setze puffer index auf ende
mvi a,80h
011b 3e80    sta ibp      ;setze pufferpointer auf 80
011d 321302  hl enthält die nächste anzuzeigende adresse
                                   ;
0120 210000  lxi h,0      ;starte mit 0000
                                   ;
gloop:      push h      ;speichere zeilennummer
0123 e5      call gnb
0124 cda201  pop h      ;rufe zeilennummer zurück
0127 e1      jc finis   ;bei dateiende setzt gnb
0128 da5101  carry
012b 47      mov b,a
                                   ;
                                   ;      zeige hex-werte an
012c 7d      mov a,l
012d e60f    ani 0fh     ;prüfe die unteren 4 bits
012f c24401  jnz nonum
                                   ;      zeige Zeilennummer an

```

```

0132 cd7201          call crlf
;
; prüfe, ob breaktaste
0135 cd5901          call break
;
0138 0f              rrc          ;in carry
0139 da5101          jc finis      ;nicht mehr anzeigen
;
013c 7c              mov a,h
013d cd8f01          call phex
0140 7d              mov a,l
0141 cd8f01          call phex
nonum:
0144 23              inx h          ;zur nächsten zeilennummer
0145 3e20            mvi a, ' '
0147 cd6501          call pchar
014a 78              mov a,b
014b cd8f01          call phex
014e c32301          jmp gloop
;
finis:
;
; ende von dump, gehe zurück zum ccp
; (beachte, daß ein sprung nach 0000h neu
; startet)
0151 cd7201          call crlf
0154 2a1502          lhld oldsp
0157 f9              sphl
;
; der stackpointer enthält die position des ccp-
; stack
0158 c9              ret          ;zum ccp
;
;
; unterroutinen
;
break: ;prüfe breaktaste (wirklich jede beliebige
; taste)
0159 e5d5c5          push h! push d! push b ;gespeichert
015c 0e0b            mvi c,brkf
015e cd0500          call bdos
0161 c1d1e1          pop b! pop d! pop h ;zurückgespeichert
0164 c9              ret
;
pchar: ;zeige ein zeichen an
;
; push h! push d! push b ;gespeichert
0165 e5d5c5          push h! push d! push b ;gespeichert
0168 0e02            mvi c,typef
016a 5f              mov e,a
016b cd0500          call bdos
016e c1d1e1          pop b! pop d! pop h;zurückspeichern
0171 c9              ret
;
crlf:
0172 3e0d            mvi a,cr
0174 cd6501          call pchar
0177 3e0a            mvi a,lf
0179 cd6501          call pchar
017c c9              ret
;

```

```

;
pnib: ;zeige halbbyte in register a an
017d e60f ani 0fh ;niederwertige 4 bits
017f fe0a cpi 10
0181 d28901 jnc p10
; weniger oder gleich 9
0184 c630 adi '0'
0186 c38b01 jmp prn
;
; größer oder gleich 10
0189 c637 p10: adi 'a' - 10
018b cd6501 prn: call pchar
018e c9 ret
;
phex: ;zeige hex-zeichen in register a an
018f f5 push psw
0190 0f rrc
0191 0f rrc
0192 0f rrc
0193 0f rrc
0194 cd7d01 call pnib ;zeige halbbyte an
0197 f1 pop psw
0198 cd7d01 call pnib
019b c9 ret
;
err: ;zeige fehlermeldung an
; die -adressen enden mit $
019c 0e09 mvi c,printf ;zeige pufferfunktion an
019e cd0500 call bdos
01a1 c9 ret
;
;
gnb: ;hole nächstes byte
01a2 3a1302 lda ibp
01a5 fe80 cpi 80h
01a7 c2b301 jnz g0
; lies einen anderen puffer
;
;
01aa cdce01 call diskr
01ad b7 ora a ;nullwert wenn lesen ok
01ae cab301 jz g0 ;für ein anderes byte
; ende der daten, zurück mit carry für eof
; gesetzt
01b1 37 stc
01b2 c9 ret
;
g0: ;lies das byte puffer + register a
01b3 5f mov e,a ;byte des pufferindex
01b4 1600 mvi d,0 ;doppelt genauer index auf
; de
01b6 3c inr a ;index = index+1
01b7 321302 sta ibp ;zurück zum speicher
; pointer wird erhöht
; speichere momentane dateiadresse
01ba 218000 lxi h,buff
01bd 19 dad d

```

```

01be 7e          ; absolute zeichenadresse in hl
                 mov a,m
                 ;
01bf b7          ; byte ist im akkumulator
01c0 c9          ora a           ;setze carrybit zurück
                 ret
                 ;
                 setup: ;setze datei auf
                 ;
01c1 af          ; öffne datei für eingabe
01c2 327c00      xra a           ;null an den akku
                 sta fcbcr      ;lösche laufenden satz
                 ;
01c5 115c00      lxi d,fcbr
01c8 0e0f        mvi c,openf
01ca cd0500      call bdos
                 ;
01cd c9          ; 255 in den akku bei eröffnungsfehler
                 ret
                 ;
                 disk:  ;lies dateisatz
01ce e5d5c5      push h! push d! push b
01d1 115c00      lxi d,fcbr
01d4 0e14        mvi c,readf
01d6 cd0500      call bdos
01d9 c1d1e1      pop b! pop d!
                 pop h
01dc c9          ret
                 ;
                 ;
01dd 64617465    ; signon: db 'datei dump version 2.0$'
01f3 0d0a6b65    opnmsg: db cr, lf, 'keine eingabedatei auf der platte $'
                 ;
0213             ; variabler bereich
0215             ibp: ds 2           ;pointer des eingabepuffers
                 oldsp: ds 2        ;eingangs sp-wert vom ccp
                 ;
                 ;
0217             ; stackbereich
                 ds 64           ;reserviere 32-stufigen stack
                 stktop:
                 ;
0257             end

```

## 6.5 Ein Programmbeispiel für Direktzugriff

Dieses Kapitel wird mit einem ausführlichen Beispiel für Direktzugriff abgeschlossen. Das unten gezeigte Programm führt auf Kommando von der Konsole Lesen und Schreiben von wahlfreien Dateisätzen aus. Vorausgesetzt, daß das Programm erstellt, assembliert und in eine Datei RANDOM.COM übertragen wurde, wird es mit dem CCP-Kommando

RANDOM X.DAT

aufgerufen. Das Programm sucht die Datei X.DAT (in diesem spe-

ziellen Falle) und fordert, sobald diese gefunden ist, den Bediener zu einer Konsoleingabe auf. Als Bedienerführung wird

Nächstes Kommando?

angezeigt. Daraufhin erfolgt eine Eingabe, welche mit CR abgeschlossen wird. Das eingegebene Kommando hat die Form:

nS nL E (nW nR Q)

wobei n ein ganzzahliger Wert zwischen 0 bis 65535 ist und S, L und E Kommandozeichen für Schreiben, Lesen und Ende sind. Wird S eingegeben, zeigt das Programm RANDOM an:

daten eingeben:

Der Bediener gibt dann bis zu 127 Zeichen ein, die mit Return abgeschlossen werden. RANDOM schreibt diese Zeichenkette in die Datei X.DAT auf Satz n. Wird L eingegeben, liest RANDOM den Satz n und zeigt die Zeichenkette am Bildschirm an. Wird E gewählt, so wird die Datei X.DAT geschlossen und das Programm kehrt zum CCP zurück. Im Interesse der Kürze ist die einzige Fehlernachricht:

fehler, bitte nochmals (error, try again)

Das Programm beginnt mit einer Initialisierungs-Sektion, in der die angegebene Datei eröffnet oder erstellt wird. Es folgt eine Schleife am Bezeichner „ready“, wo die Kommandos interpretiert werden. Der Default-Dateikontrollblock bei 005CH und der Default-Puffer auf 0080H werden bei allen Plattenoperationen benutzt. Dann folgen die Dienstroutinen, welche die hauptsächliche Zeilenbearbeitung („readc.“) enthalten. Dieses Beispiel zeigt die Elemente des direkten Dateizugriffes und kann als Basis für weitere Programmentwicklungen dienen.

**;Beispiel für Direktzugriff unter CP/M 2.0**

```

;
;
;
0100                org 100h        ;basis der tpa
;
0000 =             reboot equ 0000h  ;system neustart
0005 =             bdos  equ 0005h   ;bdos eintrittspunkt
;
0001 =             coninp equ 1       ;funktion tastatureingabe
0002 =             conout equ 2       ;funktion bildschirmausgabe
0009 =             pstring equ 9      ;zeige zeichenkette bis '$' an
000a =             rstring equ 10     ;lies konsolpuffer
000c =             version equ 12     ;gib versionsnummer zurück
000f =             openf equ 15       ;funktion dateieröffnung
0010 =             closef equ 16      ;funktion schließen
0016 =             makef equ 22       ;funktion datei erstellen
0021 =             readr equ 33       ;direkt lesen
0022 =             writr equ 34       ;direkt schreiben
;
005c =             fcb  equ 005ch     ;vorgegebener dateikontroll-
                                block
007d =             ranrec equ fcb+33   ;satzposition für direktzugriff
007f =             ranovf equ fcb+35   ;oberes byte (overflow)
0080 =             buff equ 0080h     ;pufferadresse
;
000d =             cr    equ 0dh       ;returntaste
000a =             lf    equ 0ah       ;Zeilenschaltung
;
;
;

```

**;Lade Stackpointer, Bereite Datei vor für Direktzugriff**

```

;
;
0100 31bc00        lxi sp,stack
;
;
;
0103 0e0c         mvi c,version
0105 cd0500       call bdos
0108 fe20         cpi 20h            ;version 2.0 oder spätere
010a d21600       jnc versok
;
;
;
010d 111b00       schlechte version, nachricht, systemstart
0110 cdda00       lxi d,badver
0113 c30000       call print
                                jmp reboot
;
;
;
versok:
;
0116 0e0f         richtige version für direktzugriff
0118 115c00       mvi c,openf        ;eröffne vorgegebenen fcb
011b cd0500       lxi d,fcb
011e 3c          call bdos
011f c23700       inr a              ;err 255 wird null
                                jnz ready
;
;
;

```







```

01e8 cdda00      call print      ;kommando?
01eb 0e0a       mvi c,rstring
01ed 117a00     lxi d,conbuf
01f0 cd0500     call bdos      ;lies kommandozeile
; kommandozeile da, durchlesen
01f3 210000     lxi h,0        ;beginne mit 0000
01f6 117c00     lxi d,conlin   ;kommandozeile
01f9 1a         readc: ldax d      ;nächstes kommandozei-
; chen
01fa 13         inx d          ;zur nächsten position
01fb b7         ora a          ;kann nicht das ende der
; zeile sein
01fc c8         rz
; nicht null numerisch?
01fd d630       sui '0'
01ff fe0a       cpi 10         ;carry, falls numerisch
0201 d21300     jnc endrd
;
0204 29         dad h          ;*2
0205 4d         mov c,l
0206 44         mov b,h        ;bc = wert *2
0207 29         dad h          ;*4
0208 29         dad h          ;*8
0209 09         dad b          ;*2 + *8 = *10
020a 85         add 1          ;+Ziffer
020b 6f         mov l,a
020c d2f900     jnc readc      ;für anderes zeichen
020f 24         inr h          ;überlauf
0210 c3f900     jmp readc      ;für anderes zeichen
;
endrd:
; ende des vorganges, speichere zeichen in a
0213 c630       adi '0'        ;kommando
0215 fe61       cpi 'a'        ;buchstabenübersetzung?
0217 d8         rc
; kleinbuchstaben, maske für kleinbuch-
; staben-bits
0218 e65f       ani 0101$1111b
021a c9         ret
;
;
; Zeichenketten für Bildschirmnachrichten
;
;
;
badver:
021b 73696520   db 'sie brauchen CP/M-version 2$'
nospace:
023a 6b65696e   db 'kein verzeichnisplatz $'
datmsg:
024 64617465    db 'daten eingeben: $'
errmsg:
0259 6665686c   db 'fehler, nochmals $'
prompt:
026b 6e616563   db 'nächstes kommando? $'
;
;
;

```

```

;Fester und Variabler Datenbereich
;
;
027a 21      conbuf:  db conlen      ;länge des konsolpuffers
027b        consiz:  ds 1          ;größe nach dem lesen
027c        conlin:  ds 32         ;32-zeichen puffer
0021 =      conlen   equ $-consiz
;
029c        ds 32          ;16-stufiger stack
stack:
02bc        end

```

Auch dieses Beispiel kann wesentlich verbessert werden. Mit etwas Aufwand kann es sogar in ein einfaches Datenbank-System umgewandelt werden. Man könnte z. B. eine Standard-Satzgröße von 128 Bytes mit willkürlichen Feldern annehmen. Ein Programm namens GETKEY könnte entwickelt werden, welches zuerst eine Datei sequentiell liest und ein vom Bediener spezifiziertes Feld extrahiert. Das Kommando

```
GETKEY NAMEN.DAT FAMILIENNAME 10 20
```

würde GETKEY veranlassen, die Datenbank NAMEN.DAT zu lesen und aus jedem Satz das Feld „FAMILIENNAME“ von Position 10 bis 20 extrahieren. GETKEY erstellt eine Tabelle im Speicher, die aus den FAMILIENNAME-Feldern und der 16-Bit Satznummer innerhalb der Datei besteht. Das Programm GETKEY sortiert dann diese Tabelle und schreibt eine neue Datei mit der Bezeichnung FAMILIEN.KEY, welches eine alphabetische Liste der Felder FAMILIENNAME mit der zugehörigen Satznummer darstellt. (In der Sprache der Informationsverarbeitung wird dies auch invertierter Index genannt.)

Wenn das oben genannte Programm in FINDE umbenannt und etwas aufpoliert wird, so daß es eine nach einem bestimmten Argument sortierte Datei in den Speicher liest, könnte die Kommandozeile so aussehen:

```
FINDE NAMEN.DAT FAMILIEN.KEY
```

anstatt eine Nummer zu lesen, liest das Programm FINDE eine alphabetische Zeichenkette, die einen Schlüssel in der Datei NAMEN.DAT darstellt. Da die Liste FAMILIEN.KEY sortiert ist, kann man einen bestimmten Eintrag durch eine Binärsuche schnell finden. Dies bedeutet, daß man, an beiden Enden der Liste angefangen, den mittleren Wert vergleicht. Stimmt er nicht, so wird entweder die

obere oder die untere Hälfte für die weitere Suche genommen. Auf diese Weise findet man schnell den gesuchten Begriff und die dazugehörige Satznummer. Der Bediener kann das Kommando über die Tastatur eingeben und den Satz über den Bildschirm anzeigen, wie in dem gezeigten Programmbeispiel.

Mit etwas mehr Aufwand kann auch eine feste Gruppenlänge, anders als 128-Bytes, genommen werden. Dies wird dadurch erreicht, daß man neben der Satznummer auch das Startbyte innerhalb des Satzes merkt. Wenn man die Gruppenlänge kennt, greift man direkt auf den Satz mit der entsprechenden Gruppe zu und liest bis zur gesuchten Gruppe sequentiell innerhalb des Satzes.

Schließlich kann man FINDE noch beträchtlich durch Boole'sche Ausdrücke verbessern, welche diejenigen Sätze errechnen, die mehreren Bedingungen entsprechen, wie z. B. FAMILIENNAME zwischen HARDY und LAUREL und ALTER unter 45. Es zeigt alle Sätze an, die diese Bedingungen erfüllen. Sind die Listen schließlich zu umfangreich, um im Speicher aufbewahrt zu werden, kann auch auf solche sortierte Dateien direkt auf der Platte zugegriffen werden.

## 6.6 Übersicht über die Systemfunktionen

FUNKTIONS- NUMMER	FUNKTIONS- BEZEICHNUNG	EINGABE	AUSGABE	
Dezimal Hex				
0	0	System Warmstart	C = 00H	keine
1	1	Konsoleingabe	C = 01H	A = ASCII-Zeichen
2	2	Konsolausgabe	E = Zeichen	keine
3	3	Lochstreifen lesen		A = ASCII-Zeichen
4	4	Lochstreifen stanzen	E = Zeichen	keine
5	5	Zeichen an Drucker ausgeben	E = Zeichen	keine
6	6	Direkte Konsolein- und -ausgabe	C = 06H E = 0FFH (Eingabe) o. 0FEH (Status) o. Zeichen (Ausgabe)	A=Zeichen o. Status (kein Wert)
7	7	IOByte abfragen	keine	A = IOBytewert
8	8	IOByte setzen	E = IOByte	keine
9	9	Zeichenkette ausgeben	DE = Pufferadresse	keine
10	A	Konsolpuffer lesen	DE = Puffer	Konsolzeichen im Puffer
11	B	Konsolstatus abfragen	keine	A = 00/ nicht Null
12	C	CP/M-Version ermitteln	keine	HL = Versionsnummer
13	D	Plattensystem zurücksetzen	keine	keine
14	E	Bezugslaufwerk festlegen	E = Laufwerksnr.	keine
15	F	Datei eröffnen	DE = FCB-Adresse	FF, falls nicht gefunden
16	10	Datei schließen	DE = FCB-Adresse	FF, falls nicht gefunden
17	11	Ersten Eintrag suchen	DE = FCB-Adresse	A = Verzeichniscode
18	12	Nächsten Eintrag suchen	keine	A = Verzeichniscode
19	13	Datei löschen	DE = FCB-Adresse	A = keine
20	14	Lesen sequentiell	DE = FCB-Adresse	A = Fehlercode
21	15	Satz schreiben, sequentiell	DE = FCB-Adresse	A = Fehlercode
22	16	Datei anlegen	DE = FCB-Adresse	A = FF, falls kein Platz im Verzeichnis
23	17	Datei umbenennen	DE = FCB-Adresse	A = FF, falls nicht gefunden
24	18	aktives Laufwerk ermitteln	keine	HL = Log-in Vektor*
25	19	Bezugslaufwerk ermitteln	keine	A = Bezugslaufwerk
26	1A	DMA-Adresse festlegen	DE = DMA-Adresse	keine
27	1B	Belegungstabelle ermitteln	keine	HL = ALLOC-Adresse*
28	1C	Platte vor Schreiben schützen	keine	keine
29	1D	Geschütztes Laufwerk ermitteln	keine	HL = R/O-Vektorwert*
30	1E	Dateimerkmale setzen	DE = FCB-Adresse	A = keine
31	1F	Diskettenparameter ermitteln	keine	HL = DPB-Adresse
32	20	Benutzernummer verwalten	E = 0FFH für GET E = 00 bis 0FH für SET	User-Nummer
33	21	Wahlfrei lesen	DE = FCB-Adresse	A = Fehlercode
34	22	Wahlfrei schreiben	DE = FCB-Adresse	A = Fehlercode
35	23	Dateigröße ermitteln	DE = FCB-Adresse	r0, r1, r2
36	24	Random-Satz setzen	DE = FCB-Adresse	r0, r1, r2

FUNKTIONS- NUMMER	FUNKTIONS- BEZEICHNUNG	EINGABE	AUSGABE
Dezimal Hex			
37	25	Laufwerk zurücksetzen	DE = Laufwerksvektor
38	26	Zugriff auf Laufwerk	nicht unterstützt
39	27	Laufwerk frei	nicht unterstützt
40	28	Nullauffüllen beim Schreiben	DE = FCB
			A = 0
			A = Fehlercode

\* Beachten: bei Rückkehr ist A = L und B = H.

# Programmierung

## 7.1 Platten-Organisation

Im ersten Sektor (siehe nachfolgende Tabelle) befindet sich das Ladeprogramm BOOT, das die Aufgabe hat, BIOS in den oberen Bereich des Arbeitsspeichers zu laden. Die Anfangsadresse von BIOS errechnet sich aus den zwei Bytes an den Maschinenadressen 001 und 002. Von diesem Ergebnis müssen Sie allerdings den Wert 3 subtrahieren.

### 7.1.1 48 TPI (Spuren/Zoll) Diskette

Spur #	Sektor # (512 Byte)	CP/M Modulbezeichnung
00	01	BOOT
00	02	BIOS
,	03	,
,	04	,
,	05	,
,	06	,
,	07	,
,	08	,
01	01	BIOS
,	02	,
,	03	,
,	04	,
,	05	BIOS
,	06	CCP
,	07	,
,	08	,
02	01	CCP
,	02	BDOS
,	03	,
,	04	,
,	05	,
,	06	,
,	07	,
,	08	BDOS
03-79	01-08	(Dateiverzeichnis und Daten)

## 7.1.2 96 TPI Diskette

Spur	Sektor (1024 Byte)	CP/M Modulbezeichnung
00	01	BOOT/BIOS
'	02	BIOS
'	03	'
'	04	'
'	05	nicht belegt
01	01	BIOS
'	02	BIOS
'	03	BIOS/CCP
'	04	CCP
'	05	nicht belegt
02	01	CCP/BDOS
'	02	BDOS
'	03	'
'	04	'
'	05	nicht belegt
03-159	01-05	(Dateiverzeichnis und Daten)

## 7.2 Die Eintrittspunkte zu BIOS

Die Eintrittspunkte zu BIOS vom Kaltstart-Lader und BDOS aus sind nachfolgend im einzelnen dargestellt. Der Sprungvektor besteht aus 17 Sprungkommandos, die die Programmsteuerung auf spezielle BIOS Unterroutinen übertragen.

Der Sprungvektor, der sich am Anfang von BIOS befindet sieht wie folgt aus. (Die einzelnen Sprungadressen sind dabei auf der linken Seite der Tabelle angegeben.)

### BIOS Start-Adresse

+0H	JMP BOOT	; EINTRITTS- PUNKT VON BOOT NACH DEM KALTSTART
+3H	JMP WBOOT	; EINTRITTS- PUNKT VON BOOT NACH DEM WARMSTART
+6H	JMP CONST	; PRÜFEN, OB EIN ZEICHEN VON DER KONSOLE VORLIEGT
+9H	JMP CONIN	; EINLESEN DES ZEICHENS VON DER KONSOLE
+CH	JMP CONOUT	; AUSGEBEN EINES ZEICHENS ZUR KONSOLE
+FH	JMP LIST	; AUSGEBEN EINES ZEICHENS ZUM DRUCKER
+12H	JMP PUNCH	; AUSGEBEN EINES ZEICHENS ZUM STANZER
+15H	JMP READER	; EINLESEN EINES ZEICHENS VOM LE- SER
+18H	JMP HOME	; POSITIONIEREN DES LESEKOPFES EINER PLATTENEINHEIT AUF SPUR 00
+1BH	JMP SELDSK	; AUSWÄHLEN DES PLATTENLAUF- WERKS
+1EH	JMP SETTRK	; SETZEN DER SPURNUMMER
+21H	JMP SETSEC	; SETZEN DER SEKTORNUMMER
+24H	JMP SETDMA	; SETZEN DER DMA-ADRESSE
+27H	JMP READ	; EINLESEN DES GEWÄHLTEN SEKTORS
+2AH	JMP WRITE	; SCHREIBEN DES GEWÄHLTEN SEK- TORS
+2DH	JMP LISTST	; RÜCKGEBEN DES DRUCKERSTATUS
+30H	JMP SECTTRAN	; REGISTER-TRANSFER DER SEKTOR- NUMMER

Zu jeder Sprungadresse gehört eine Unteroutine, die spezielle Funktionen übernimmt. Diese Funktionen werden nachfolgend beschrieben. Die Sprung-Tabelle enthält drei wesentliche Bestandteile: die (Wieder-) Initialisierung des Systems, die durch den Abruf von BOOT oder WBOOT ausgelöst wird; die Ein/Ausgabe eines Zeichens aufgrund von CONST, CONIN, CONOUT, LIST, PUNCH, READER und LISTST; und Ein/Ausgabeoperationen für die Platte, die durch HOME, SELDSK, SETTRK, SETSEC, SETDMA, READ, WRITE und SECTRAN eingeleitet bzw. bewirkt werden.

Alle Ein/Ausgabe-Operationen für ein Zeichen erwarten es als ASCII-Zeichen in Groß- oder Kleinschreibung mit dem höchstwertigen Bit (Parity-Bit) 0. Das Dateiende für ein Eingabegerät wird durch ASCII control-z (1AH) gekennzeichnet.

Periphere Geräte (Einheiten) werden von CP/M als logische Geräte betrachtet und werden physischen Geräten in BIOS zugeordnet.

Für die Benutzung von BDOS sind nur die Unterrouتين CONST, CONIN und CONOUT erforderlich. Der LISTST-Eintrag wird zur Zeit nur im Zusammenhang mit DESPOOL, dem Dienstprogramm für Drucker-Spooling, benutzt.

Für die Geräte werden die folgenden Kennzeichen verwendet:

CONSOLE	für die interaktive Konsole, die die Kommunikation mit dem Bediener gestattet. Der Zugriff zu ihr erfolgt mit Hilfe von CONST, CONIN und CONOUT. Der Bildschirm (CRT) ist das typische Beispiel für eine Konsole.
LIST	für den Drucker - soweit angeschlossen -. LIST neben Druckern auch für Teletype verwendet.
PUNCH	für den Stanzer, falls angeschlossen.
READER	für den Leser, falls angeschlossen.

Eine periphere Einheit kann gleichzeitig als LIST, PUNCH und READER bezeichnet werden.

Um die Flexibilität zu erhöhen, steht dem Benutzer zusätzlich die „IOBYTE“-Funktion zur Verfügung, die die Neuzuweisung von physischen und logischen Geräten gestattet. Diese Funktion erstellt eine Tabelle der Zuordnung der logischen zu den physischen Geräten, deren Inhalt bei Verwendung von CP/M geändert werden kann

(siehe z. B. das Kommando STAT). Die Definition der IOBYTE-Funktion entspricht dem Intel Standard: Ein 1-Byte Feld des Arbeitsspeichers genannt IOBYTE (zur Zeit mit der Adresse 0003H) dient der gerade bestehenden Zuordnung der logischen und physischen Geräte. Die Aufteilung von IOBYTE in 4 verschiedene Felder zu je 2 Bits bildet den Rahmen für die Tabelle. Die Felder tragen die Bezeichnungen CONSOLE, READER, PUNCH und LST. Der Aufbau der Tabelle sieht wie folgt aus:

	höchstwertige		niedrigstwertige	
IOBYTE IN 003H	LIST Bits 6, 7	PUNCH Bits 4, 5	READER Bits 2, 3	CONSOLE Bits 0, 1

Der Wert in jedem Feld dient der Zuordnung des jeweiligen peripheren Geräts. Er kann 0-3 betragen. Die Werte, die in den Feldern verwendet werden können, gehen aus der folgenden Übersicht hervor.

#### CONSOLE Feld (Bits 0, 1)

- 0 CONSOLE ist dem seriellen Drucker 1 (TTY1:) zugeordnet
- 1 CONSOLE ist dem Bildschirm (CRT:) zugeordnet
- 2 CONSOLE ist dem Bildschirm (CRT:) zugeordnet
- 3 CONSOLE ist dem seriellen Drucker 2 (TTY1:) zugeordnet

#### READER Feld (Bits 2, 3)

- 0 READER ist dem seriellen Drucker 2 (TTY2:) zugeordnet
- 1 READER ist dem Bildschirm (CRT:) zugeordnet
- 2 READER ist dem Bildschirm (CRT:) zugeordnet
- 3 READER ist dem seriellen Drucker 2 (TTY2:) zugeordnet

#### PUNCH Feld (Bits 4, 5)

- 0 PUNCH ist dem seriellen Drucker 2 (TTY2:) zugeordnet
- 1 PUNCH ist dem Bildschirm (CRT:) zugeordnet
- 2 PUNCH ist dem parallelen Drucker (LPT:) zugeordnet
- 3 PUNCH ist dem Bildschirm (CRT:) zugeordnet

#### LIST Feld (Bits 6, 7)

- 0 LIST ist dem seriellen Drucker 1 (TTY1:) zugeordnet
- 1 LIST ist dem Bildschirm (CRT:) zugeordnet
- 2 LIST ist dem parallelen Drucker (LPT:) zugeordnet
- 3 LIST ist dem seriellen Drucker 2 (TTY2:) zugeordnet

Die Ein/Ausgabe-Operationen für die Platte erfolgen durch eine Reihe von Abrufen von verschiedenen Zugriffs-Routinen, die die Nummer des betroffenen Laufwerks auswählen, die Spur- und Sektor-Nummer der Platte und die DMA-Adresse (DMA = Disk Memory Access) der jeweiligen Ein/Ausgabe-Operation setzen.

Es wird häufig eine Aufforderung an SELDSK gerichtet, ein Laufwerk zu aktivieren. Es folgt dann eine Reihe von Lese- oder Schreiboperationen, die die gewählte Platte betreffen. Danach kann der gleiche Vorgang für ein anderes Laufwerk gefordert werden.

Ebenso kann anfangs die DMA-Adresse gesetzt werden. Es folgen Lese/Schreiboperationen, mit dem gewählten DMA-Bereich, bevor die DMA-Adresse geändert wird.

Die Unterroutinen für die Spuren und Sektoren werden immer vor der Durchführung der Lese/Schreib-Operationen abgerufen.

Die Lese- und Schreibroutinen wiederholen die jeweilige Operation bei negativem Erfolg mehrfach (im Normalfall 10 mal), bevor sie eine Fehlermeldung ausgeben.

Es folgt nun eine Beschreibung der Funktion der Unterroutinen, die mit jedem Eintrittspunkt verbunden sind.

**BOOT** Dieser Eintrittspunkt wird vom Kaltstart-Lader aktiviert und ist für die wesentliche Initialisierung des Systems verantwortlich. Dazu gehört das Setzen von IOBYTE.

Die verschiedenen Parameter, die im Bereich des Punktes WBOOT gesetzt werden, müssen in die Initialisierung einbezogen werden. Danach wird die Steuerung für die nachfolgende Verarbeitung auf den CCP übertragen. Um Laufwerk A zu aktivieren, wird Register C entsprechend gesetzt.

**WBOOT** Bei einem Warmstart wird WBOOT aktiviert. Ein Warmstart erfolgt immer, wenn ein Anwenderprogramm auf Adresse 0000H verzweigt.

Das CP/M System wird vom Laufwerk A geladen (ohne BIOS). Systemparameter werden wie folgt eröffnet:

Speicher-      auf JMP WBOOT für Warmstart  
adresse 0,1,2    gesetzt

Speicher-      anfänglichen Wert von IOBYTE  
adresse 3        setzen

Speicher-      1. Halbbyte = Benutzer-Nummer  
adresse 4      2. Halbbyte = aktiviertes Laufwerk

Speicher-      auf JMP BDOS gesetzt. Das ist der  
adresse 5,6,7    primäre Eintrittspunkt zu CP/M  
für transiente Programme

In Abschnitt 7.3 wird die Benutzung der Speicherseite 0 eingehend beschrieben). Nach vollendeter Initialisierung verzweigt das WBOOT-Programm zu CCP, um den Start (Neustart) auszulösen. Beim Eintritt in CCP wird Register C mit der Laufwerksnummer geladen, die nach der Initialisierung angesprochen werden soll. Die WBOOT-Routine liest dazu den Inhalt von Byte 4 im Arbeitsspeicher und übergibt ihn an CCP in Register C.

CONST	CONST behandelt den Status der gegenwärtig zugeordneten Konsole. Wenn ein Zeichen zum Lesen von der Konsole bereitsteht, wird Register A mit einem Wert > 00H geladen. Falls dies nicht zutrifft, enthält Register A den Wert 00H.
CONIN	Das nächste Zeichen der Konsole wird in Register A eingelesen und das Parity-Bit (das höchstwertige Bit) wird auf 0 gestellt. Falls kein Zeichen verfügbar ist, verharrt Register A im Wartezustand bis ein Zeichen eingegeben wird.
CONOUT	Ein Zeichen wird von Register C im ASCII-Code an der Konsole ausgegeben. Das höchstwertige Bit weist dabei den Wert 0 auf.
LIST	Ein ASCII-Zeichen mit Parity-Bit 0 wird von Register C am gegenwärtig zugeordneten Drucker ausgegeben.
PUNCH	Ein ASCII-Zeichen mit Parity-Bit 0 wird von Register C am gegenwärtig zugeordneten Stanzer ausgegeben.
READER	Das nächste Zeichen wird von dem gegenwärtig zugeordneten Leser mit Parity-Bit 0 in Register A eingelesen.
HOME	Der Lese/Schreib-Kopf wird beim gegenwärtig angesprochenen Platten-Laufwerk zur Spur 00 bewegt.
SELDSK	Die Auswahl des Laufwerks erfolgt mit Hilfe des Inhalts von Register C. Der Inhalt 0 kennzeichnet Laufwerk A, 1 Laufwerk B und so weiter bis 15 für Laufwerk P (die Standard CP/M Version unterstützt 4 Laufwerke). Bei jeder Auswahl eines Laufwerks übermittelt SELDSK die Adresse eines 16-Byte Feldes für den Plattenparameter-Kopfsatz (siehe Abschnitt 7.4) an das Register HL. Falls der Versuch unternommen wird, ein nicht vorhandenes Laufwerk anzusprechen, antwortet SELDSK mit HL= 0000H als Fehleranzeige.

Obwohl SELDSK die Adresse des Kopfsatzes bei jedem Aufruf übermittelt, ist zu empfehlen, die Auswahl des Laufwerks erst zum Zeitpunkt der Platten-Operationen (Suchen, Lesen oder Schreiben) vorzunehmen.

- SETTRK** Register B enthält die Spurnummer für weitere Zugriffe auf die gerade ausgewählte Platte.
- SETSEC** Register BC enthält die Sektornummer für weitere Zugriffe auf die gerade ausgewählte Platte. Die Sektornummer in BC stimmt mit der vom Eintrittspunkt SECTRAN übermittelten Nummer überein.
- SETDMA** Register BC enthält die DMA-(Disk Memory Access) Adresse für weitere Lese-Schreiboperationen. Falls z. B. B = 00H und C = 80H wenn SETDMA angesprochen wird, lesen alle nachfolgenden Lese-Operationen die Daten in den Bereich von 80H bis 0FFH bis zum nächsten Abruf von SETDMA ein.
- READ** Nachdem das Laufwerk ausgewählt, die Spur- und Sektornummer und die DMA-Adresse gesetzt wurden, versucht die READ Unteroutine einen Sektor zu lesen. Je nach Erfolg wird dabei Code 0 oder 1 in Register A festgehalten.

Dabei besagt

0 es trat kein Fehler auf

1 es liegt eine nicht zu rekonstruierende Fehlerbedingung vor

Falls ein Lesefehler festgestellt wird, antwortet BIOS mit einer geeigneten Fehlernachricht. Der Benutzer kann in diesem Fall die Operation wiederholen oder abbrechen. Bei Benutzung der Taste O (Überschreiten) wird der Fehler an BDOS gemeldet. BDOS antwortet danach mit der Meldung „BDOS ERR ON x: BAD SECTOR“. Sie können dann mit CR reagieren, um den Fehler zu

ignorieren, oder mit `ctl-C`, um abzubrechen.

- WRITE** Die Daten werden von dem gerade gewählten DMA-Bereich auf die bezeichnete Spur und Sektor einer Platte geschrieben, die sich in dem gewählten Laufwerk befindet. Bei Disketten sollten die Daten als nicht zu löschende Daten markiert sein, um die Kompatibilität mit anderen CP/M Systemen zu gewährleisten. Die Fehler-Codes, die von dem `READ` Kommando ausgehen, werden in Register A festgehalten. Vorher wurde, wie oben beschrieben, durch die Wiederholung der Operation versucht, den Fehler auszuschalten.
- LISTST** Der Arbeitsstatus des Druckers, der von dem `DESPOOL` Programm benutzt wird, um Konsoloperationen zu beschleunigen, wird übermittelt. Der Wert `00` wird nach Register A gesendet, falls der Drucker nicht bereit ist, ein Zeichen zu empfangen und `0FFH` wird benutzt, um anzuzeigen, daß der Drucker bereit ist, ein Zeichen zu erhalten.
- SECTRAN** Diese Routine kopiert die Sektornummer von `BC` nach `HL`, ohne eine Übersetzung vorzunehmen (keine Übersetzung von logischen in physische Sektoren).

### 7.3 Reservierte Felder in Speicherseite Null

Die Speicherseite 0 im Arbeitsspeicher enthält im Bereich zwischen Adressen `00H` und `0FFH` mehrere Segmente mit Codes und Daten, die bei der Verarbeitung mit CP/M benutzt werden. Die Code- und Datenbereiche sind aus der folgenden Tabelle zu ersehen.

Felder von bis	Inhalt
0000H-0002H	Enthält eine Sprunginstruktion zum Eintrittspunkt für den Warmstart. Dies erlaubt einen einfach programmierten Neustart (JMP 0000H).
0003H-0003H	Enthält IOBYTE
0004H-0004H	Nummer des gegenwärtig automatisch zugeschalteten (Default-)Laufwerks (0=A, . . . , 15=P).
0005H-0007H	Enthält eine Sprunginstruktion zu BDOS und dient zwei Zwecken: JMP 0005H stellt den primären Eintrittspunkt zu BDOS zur Verfügung wie es in Kapitel 6 beschrieben wird und LHL 0006H übermittelt das Adressenfeld der Instruktion zu dem HL Registerpaar. Dieser Wert ist die niedrigste Adresse, die von CP/M im Arbeitsspeicher angesprochen wird (bei Unterstellung, daß CCP überlagert wird). Das DDT Programm ändert den Inhalt des Adressenfeldes, um die verringerte Speichergröße bei Testhilfe-Betrieb wiederzugeben.
0008H-0027H	Felder 1 bis 5 sind für Unterbrechungen vorgesehen. Sie werden zur Zeit nicht verwendet.
0030H-0037H	Feld 6 ist für Unterbrechungen reserviert. Zur Zeit wird es nicht verwendet.
0038H-003AH	Feld 7 für Neustart; enthält eine Sprunginstruktion zum DDT oder SID Programm für den Testhilfe-Betrieb bei Verwendung von programmierten Unterbrechungspunkten. CP/M verwendet dieses Feld lediglich in diesem Fall.
003BH-003FH	Gegenwärtig nicht benutzt; reserviert.

0040H-004FH	Ein reservierter 16-Byte Bereich, der von der angebotenen CP/M-Version nicht verwendet wird.
0050H-005BH	Gegenwärtig nicht benutzt, reserviert.
005CH-007CH	Von CCP vorgegebener (Default) Datei-Kontrollblock für ein transientes Programm.
007DH-007FH	Optionaler Speicherplatz für Random-Satz-Position im Default-FCB.
0080H-00FFH	Vorgegebener (Default) 128-Byte Puffer für die Platte (wird ebenso benutzt für die Kommandozeile, wenn ein transientes Kommando unter der Kontrolle von CCP geladen wird).

Diese Informationen werden für die normale Verarbeitung unter dem CP/M System bereitgestellt. Sie können jedoch von einem transienten Programm überschrieben werden, falls die BDOS-Leistungen dafür nicht benötigt werden.

Falls ein Programm z. B. nur der einfachen Ein / Ausgabe dient und zur Ausführung bei Adresse 0 starten soll, kann es mit einem kleinen MOVE-Programm (einem Bestandteil von CP/M) in die TPA geladen werden. Das MOVE-Programm übernimmt die Steuerung ab Adresse 0100H, dem Beginn aller transienten Programme.

Das MOVE-Programm kopiert dann das gesamte Spiegelbild des Speichers bis zur Adresse 0 und übergibt die Steuerung der Adresse 0 des neu geladenen Inhalts.

Falls BIOS oder das Feld mit Adresse 0 (es enthält den Eintrittspunkt für den Warmstart) überschrieben wird, muß der Bediener das CP/M System mit Kaltstart in den Speicher zurückholen.

## 7.4 Plattenparameter Tabellen

BIOS enthält Tabellen, die spezifische Kenndaten des Platten-Untersystems speichern, das von CP/M benutzt wird. Im folgenden werden die Elemente dieser Tabellen beschrieben.

Im allgemeinen hat jedes Platten-Laufwerk einen zugeordneten 16-Byte Plattenparameter-Kopfsatz, der Informationen über das Laufwerk enthält und einen Arbeitspuffer für gewisse BDOS-Operationen. Das Format des Plattenparameter-Kopfsatzes für jedes Laufwerk sieht wie folgt aus:

## Plattenparameter-Kopfsatz

XLT	0000	0000	0000	DIRBUF	DPB	CSV	ALV
16b	16b	16b	16b	16b	16b	16b	16b

Dabei ist jedes Feld ein 16 Bit-Wert. Die einzelnen Felder des Parameter-Kopfsatzes (DPH = Disk Parameter Header) bedeuten:

XLT	Immer 0000H, da keine Übersetzung der Sektornummer stattfindet, (d. h. die physische und die logische Sektornummern sind immer gleich).
0000	Werte für den Arbeitspuffer, der von BDOS verwendet wird (die anfänglichen Werte sind unwichtig).
DIRBUF	Adresse eines 128-Byte Pufferbereichs für Directory-Operationen innerhalb von BDOS. Alle DPH's adressieren den gleichen Arbeitspuffer-Bereich.
DPB	Adresse eines Plattenparameter-Blocks für dieses Laufwerk. Laufwerke mit identischen Kenndaten für Platten sprechen den gleichen Plattenparameter-Block an.
CSV	Adresse eines Arbeitspuffer-Bereichs, der benutzt wird, um softwaregesteuert Prüfungen bei einem Plattenwechsel vorzunehmen. Diese Adresse ist für jeden DPH verschieden.
ALV	Adresse eines Arbeitspuffer-Bereichs, der von BDOS benutzt wird, um Informationen zu speichern, die die Zuordnung von Speicherplatz auf Platten betreffen. Diese Adresse ist für jeden DPH verschieden.

Bei n Platten-Laufwerken sind die DPH's in einer Tabelle angeordnet, deren erste Reihe mit 16 Bytes dem Laufwerk 0 zugeordnet ist. Die letzte Reihe ist dem Laufwerk n-1 vorbehalten. Die Tabelle sieht also wie folgt aus:

DPBASE:

00	XLT 00	0000	0000	0000	DIRBUF	DBP 00	CSV 00	ALV 00
01	XLT 01	0000	0000	0000	DIRBUF	DBP 01	CSV 01	ALV 01
und so weiter bis								
n-1	XLTn-1	0000	0000	0000	DIRBUF	DBPn-1	CSVn-1	ALVn-1

Dabei kennzeichnet DPBASE die Basis-Adresse der DPH-Tabelle.

Es ist die Aufgabe der SELDSK-Unteroutine, die Basis-Adresse der DPH-Tabelle für das ausgewählte Laufwerk zu übermitteln. Der Plattenparameter-Block (DPB) für jedes Laufwerk ist komplexer.

Ein spezieller DPB, der von einem oder mehreren DPH's angesprochen wird, hat das folgende allgemeine Format:

SPT	BSH	BLM	EXM	DSM	DRM	AL0	AL1	CKS	OFF
16b	8b	8b	8b	16b	16b	8b	8b	16b	16b

Die Indikatoren 8 b oder 16 b geben an, ob das Feld aus einem oder zwei Bytes besteht.

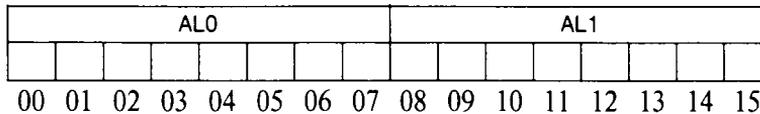
- SPT      gibt die gesamte Anzahl der Sektoren je Spur an.
- BSH      gibt den Verschiebefaktor für Blöcke mit zugeordneten Daten wieder, der von der Größe der zugeordneten Datenblöcke abhängt (BSH enthält bei Disketten den Wert 4, bei Festplatten den Wert 6.)
- BLM      ist die Maske des Blocks mit zugeordneten Daten (2\*\*BSH-1). (BLM enthält bei Disketten 0FH, bei Festplatten 3FH.)
- EXM      ist die Extent-Maske, die von der Blockgröße der zugeordneten Daten und der Anzahl der Blöcke abhängt.
- DSM      gibt die gesamte Speicherkapazität des Platten-Laufwerks an.
- DRM      gibt die Anzahl der Directory-Einträge an, die für dieses Laufwerk gespeichert werden können (AL0, AL1 bestimmen die reservierten Directory-Blöcke).

- CKS** ist die Größe des Directory Prüf-Vektors.
- OFF** gibt die Anzahl der reservierten Spuren am Anfang der (logischen) Platte an.

Der Wert von DSM stellt die größte Anzahl von Datenblöcken für das dazugehörige Laufwerk dar. Er wird in BLS-Einheiten angegeben (BLS = 2048 Bytes). Das Produkt BLS (DSM + 1) nennt die gesamte Anzahl der Bytes, die je Laufwerk zur Verfügung steht. Dabei werden die für das Betriebssystem vorgesehenen Spuren nicht berücksichtigt.

DRM ist um eins kleiner als die gesamte Anzahl von Directory-Einträgen, die ein 16-Bit Wert annehmen kann. Die Werte von AL0 und AL1 werden dagegen von DRM bestimmt.

Die Werte AL0 und AL1 können als eine Reihe von 16 Bits angesehen werden, wie nachfolgend gezeigt:



Dabei entspricht die Position 00 dem höchstwertigen Bit von Byte AL0 und 15 dem niedrigstwertigen Bit von Byte AL1. Jede Bit-Position reserviert einen Datenblock für die Nummern von Directory-Einträgen. Es können deshalb 16 Datenblöcke für Directory-Einträge zugeordnet werden. Die Bits werden beginnend mit 00 aufsteigend bis Position 15 zugeordnet. Jeder Directory-Eintrag belegt 32 Bytes.

BLS	Directory-Eintrag
2048	64 x Anzahl Bits

Wenn DRM = 127 (128 Directory-Einträge) und BLS = 2048 sind bei 64 Directory-Einträgen pro Block, 2 Blöcke zu reservieren. In diesem Fall werden die beiden höchstwertigen Bits von AL0 gesetzt. Dabei ergeben sich die Werte AL0 = 0C0H und AL1 = 00H.

Der CKS-Wert wird bei der Wechselplatte wie folgt bestimmt:  $CKS = (DRM+1):4$  wobei DRM die Nummer des letzten Directory-Eintrags ist.

Das OFF-Feld gibt die Anzahl der Spuren an, die zu Beginn der physischen Platte übersprungen werden (reservierte Spuren für das Betriebssystem). Dieser Wert wird automatisch hinzu addiert, sobald SETTRK abgerufen wird.

Bei der Rückkehr zum DPH für ein spezielles Laufwerk, bleiben

die beiden Adreßwerte CSV und ALV erhalten. Beide Adressen zielen auf einen nicht-initialisierten Bereich des Arbeitsspeichers, der auf BIOS folgt. Dieser Bereich wird laufwerk-spezifisch verwendet. Seine Größe wird jeweils durch die Werte im DPB festgelegt.

Die Größe des Bereichs, der von CSV adressiert wird, ist CKS Bytes. Dieser Bereich genügt, um die Prüfinformationen zum Directory für das betreffende Laufwerk aufzunehmen. Falls  $CKS = (DRM+1):4$ , müssen  $(DRM+1):4$  Bytes für Directory Prüfzwecke vorgesehen werden. Falls  $CKS = 0$ , muß kein Speicherplatz reserviert werden.

Die Größe des Bereichs, der von ALV adressiert wird, wird von der maximalen Anzahl von Datenblöcken bestimmt, die für die betreffende Platte zugelassen sind und wird nach der Formel  $(DSM:8)+1$  errechnet.

## SOFTWARE INSTALLATION

Dieser Abschnitt beschäftigt sich mit dem Einsatz von Anwenderprogrammen für Ihren Computer NCR DECISION MATE V.

Um alle Möglichkeiten des NCR DECISION MATE V voll auszuschöpfen, muß jede Anwendung in den Computer installiert oder speziell zugeschnitten werden. Dabei kann die Prozedur der Installation für jede Anwendung verschieden sein. So enthält jedes frei am Markt erhältliche Anwenderprogramm, wie z. B. „Wordstar<sup>®</sup>“ und ähnliche, eigene Programme zur Steuerung des Arbeitsablaufs. Gewöhnlich gehört zu diesen Softwarepaketen eine eigene Dokumentation, welche die Anwendung beschreibt. Wir empfehlen Ihnen, sich generell vor dem ersten Einsatz von Anwendungsprogrammen jeweils erst eine Kopie der Diskette anzufertigen und diese dann für die Arbeit zu benutzen. Die Originaldiskette sollten Sie sicher aufbewahren und nur zum Kopieren verwenden.

Am Anfang einer Installation melden sich die meisten Programme mit einer Auswahlliste verschiedener Computerfabrikate. Falls in der Liste NCR DECISION MATE V aufgeführt ist, verwenden Sie diesen Namen; andernfalls wählen Sie Lear Siegler, Typ ADM-31 oder Typ ADM-3A. (Tabelle 1 zeigt die aktivierbaren Funktionen bei Auswahl von Lear Siegler Terminals.)

Wenn keines der vorstehenden Terminals in der Liste erscheint (oder in Ausnahmefällen die Auswahl Lear Siegler nicht funktioniert), muß die entsprechende Terminal-Charakteristik des NCR DECISION MATE V eingegeben werden. Die Tabelle 1 zeigt hierzu den Dezimal- und Hexadezimal-Code sowie die zu betätigenden Tasten. (Diese Zuordnungen gelten nur für den Betrieb von Anwenderprogrammen und nicht für sonstige Eingaben von der Tastatur.)

Zwei weitere Tabellen zeigen für Anwendungen eventuell zusätzlich erforderliche Informationen. Tabelle 2 enthält sonstige allgemeine Informationen; Tabelle 3 gibt Hinweise und zeigt die Frequenzen und Zyklen für die Programmierung von Musik-Anwendungen.

TERMINAL FUNKTIONS CODES – Tabelle 1 (Teil 1)				
Funktion	Dezimal	Hexadezimal	Taste	LEAR SIEGLER ADM-31 LEAR SIEGLER ADM-3A
POSITION CURSOR (Erste Sequenz)	27 61	1B 3D	ESC =	X X
ROW + Offset	ROW + 32	ROW + 20		X X
Zweite Sequenz	keine erforderlich			
COL + Offset	COL + 32	COL + 20		X X
Letzte Sequenz	keine erforderlich			
CURSOR LEFT (nicht löschend)	08	08	CONTROL-H	X X
CURSOR DOWN (Zeilenaufschaltung)	10	0A	CONTROL-J	X X
CURSOR UP (Zeile zurück)	11	0B	CONTROL-K	X X
CURSOR RIGHT (nicht löschend)	12	0C	CONTROL-L	X X
CURSOR HOME (Bildschirm links oben)	30	1E	CONTROL-;	X X
BILDSCHIRM LÖSCHEN und CURSOR HOME	26	1A	CONTROL-Z	X X
CURSORPOSITION speichern	27,91,115	1B,5B,73	ESC,[,s	
CURSORPOSITION wiederherstellen	27,91,117	1B,5B,75	ESC,[,u	
LÖSCHEN ZUM ENDE DER ZEILE	23	17	CONTROL-W	
CARRIAGE RETURN	13	0D	CONTROL-M	X X
ABGABE	27	1B	ESC	X X
QUITTUNGSTON	07	07	CONTROL-G	X X

( ( (

### TERMINAL FUNKTIONS CODES – Tabelle 1 (Teil 2)

BEACHTTE: Nachstehenden Funktionen ist ein ESC-Zeichen voranzustellen, Hex 1B, Dezimal 27.

Funktion	Dezimal	Hexadezimal	Taste	LEAR SIEGLER ADM-31	LEAR SIEGLER ADM-3A
LÖSCHEN ZUM ENDE DER ZEILE	84	54	T		
	oder	74	t	X	
LÖSCHEN ZUM ENDE DES BILDES	89	59	Y		
	oder	79	y	X	
BILD LÖSCHEN und CURSOR HOME	58	3A	:		
	oder	2A	*		
HALBE LEUCHTDICHTE AUS	40	28	(	X	
HALBE LEUCHTDICHTE EIN	41	29	)	X	
MUSIK*	77	4D	M		
ZEILE EINFÜGEN	69	45	E	X	
ZEICHEN EINFÜGEN	81	51	Q	X	
ZEILE LÖSCHEN	82	52	R	X	
ZEICHEN LÖSCHEN	87	57	W	X	
RÜCKSETZEN VON BLINKENDER BILD- SCHIRM-AUSGABE UND -INVERSION**	71,48	47,30	G,0	X	
BLINKENDE BILDSCHIRM-AUSGABE EIN**	71,50	47,32	G,2	X	
BILDSCHIRM-INVERSION EIN**	71,52	47,34	G4	X	

\* Musikprogrammierung: Nach Empfangen des Funktionsaufrufs werden die nachfolgenden zwei Nummern als Frequenz und Tonlänge übernommen. Frequenz = 32 bis 74 (32 ist eine Pause, kein Ton). Tonlänge = 32 bis 255. Eine Tonlänge von 32 = 20 Millisekunden. Die Tonlänge nimmt in Stufen von jeweils 20 Millisekunden zu, bis 255 = 4,48 Sekunden. Siehe hierzu Tabelle 3 (Note – Frequenz – Zyklenzahl) am Ende dieses Abschnitts.

\*\* Erfordert die Betätigung von zwei Tasten.

BEACHTTE: In dieser Tabelle nicht aufgeführte Funktionen können im NCR DECISION MATE V nicht aufgerufen werden.

**TERMINAL FUNCTION CODES = Tabelle 1 (Teil 3)**

**HINWEIS:** Den folgenden Funktionen muß ein ESCAPE-Zeichen ("ESC", Hexadecimalwert 1B, Dezimalwert 27) und eine eckige Klammer ("["; Hexwert 5B; Dezimalwert 91) vorangestellt werden. Abgeschlossen werden sie durch ein "]" (Hexwert 6D, Dezimalwert 109).

<b>Funktion</b>	<b>Dezimal</b>	<b>Hexadezimal</b>	<b>Taste</b>	<b>LEAR SIEGLER ADM-31</b>	<b>LEAR SIEGLER ADM-3A</b>
<b>VORDERGRUND</b>					
Schwarz	51,48	33,30	3,0		
Rot	51,49	33,31	3,1		
Grün	51,50	33,32	3,2		
Gelb	51,51	33,33	3,3		
Blau	51,52	33,34	3,4		
Magenta	51,53	33,35	3,5		
Cyan	51,54	33,36	3,6		
Weiß	51,55	33,37	3,7		
<b>HINTERGRUND</b>					
Schwarz	52,48	34,30	4,0		
Rot	52,49	34,31	4,1		
Grün	52,50	34,32	4,2		
Gelb	52,51	34,33	4,3		
Blau	52,52	34,34	4,4		
Magenta	52,53	34,35	4,5		
Cyan	52,54	34,36	4,6		
Weiß	52,55	34,37	4,7		

**HINWEIS:** Die obigen Codes können Sie beliebig verketteten, ohne anfangs 1BH,5BH zu wiederholen. 3BH (;) muß als Trennzeichen zwischen jedem Element stehen, z.B. Roter Vordergrund und Schwarzer Hintergrund ESC[31;40m

Verschiedene Informationen - Tabelle 2	
Anzahl der Reihen	24 (0-23)
Anzahl der Spalten	80 (0-79)
Cursor Ursprung	0
Eingangs/Ausgangs-Technologie	BDOS calls
Cursor ein/aus	nicht aktiv
Tastatur-Ton Ein/Aus	nicht aktiv

Musik Codes - Tabelle 3 (Teil 1)		
Note	Frequenz	Zyklen
Pause	32	-
A	33	110
A#	34	116.5
H	35	123.5
C	36	131
C#	37	138.6
D	38	146.8
D#	39	155.8
E	40	164.8
F	41	174.6
F#	42	185
G	43	196
G#	44	208
A	45	220
A#	46	233
H	47	246.9
C (mittleres C)	48	261.6
C#	49	277.4
D	50	293.7
D#	51	311
E	52	329.6

**Musik Codes Tabelle 3 (Teil 2)**

<b>Note</b>	<b>Frequenz</b>	<b>Zyklen</b>
F	53	349.2
F#	54	370
G	55	392
G#	56	415
A	57	440
A#	58	465
H	59	493.9
C	60	523.2
C#	61	553
D	62	587.3
D#	63	622
E	64	659.3
F	65	698.5
F#	66	740
G	67	784
G#	68	830
A	69	880
A#	70	932
H	71	987.8
C	72	1046.5
C#	73	1108.7
D	74	1174.7

## ÜBERSETZUNGS-/UMSETZUNGS-TABELLEN

In den verschiedenen Ländern gibt es sprachabhängig einige Sonderzeichen. Daher stimmen bei manchen Beispielen in diesem Buch die Schriftzeichen (eingegebene oder angezeigte) teilweise nicht mit den Zeichen auf der Tastatur überein. Die untenstehende Tabelle zeigt Ersatzzeichen, welche bei gleichem Hexadezimal-Code die Anforderungen des Betriebssystems erfüllen. Ferner ist in der Tabelle „ASCII Code Chart“ eine Referenz zu den internationalen Zeichensätzen zusammengefaßt.

### Länderspezifische Zeichen

Landessprache	Hex Codes und äquivalente Zeichen											
	23	24	40	5B	5C	5D	5E	60	7B	7C	7D	7E
US-Englisch	#	\$	@	[	X	]	^	`	{		}	~
UK-Englisch	£	\$	@	[	X	]	†	`	{		}	~
Französisch	£	\$	à	°	ç	§	^	`	é	ù	è	..
Deutsch	#	\$	§	Ä	Ö	Ü		`	ä	ö	ü	ß
Schwedisch/Finnisch	#	◇	@	Ä	Ö	Å		`	ä	ö	å	..
Dänisch/Norwegisch	£	\$	@	Æ	Ø	Å		`	æ	ø	å	..
Spanisch	£	\$	@	í	Ñ	¿	^	`	{	ñ	}	~
Italienisch	£	\$	§	°	ç	é		ù	à	ò	è	ì

Länderspezifische Zeichen

Landessprache	Hex Codes und äquivalente Zeichen											
	23	24	40	5B	5C	5D	5E	60	7B	7C	7D	7E
Schweiz (französ.)	£	\$	ç	à	é	è	ˆ	˙	ä	ö	ü	˝
Kanada (engl.)	#	\$	@	[	\	]	ˆ	˙	£	l	q	˝
Kanada (französ.)	#	\$	@	[	ç	]	ˆ	˙	é	è	q	˝
Südafrika (afrikaans)	#	\$	@	Ê	'N	Ë	ˆ	˙	ë	'n	ë	˝
portugiesisch	£	\$	@	Ã	Õ	Ç	ˆ	˙	ã	õ	ç	˝
jugoslawisch	#	\$	Đ	Ć	Č	Š	Ž	đ	ć	č	š	ž

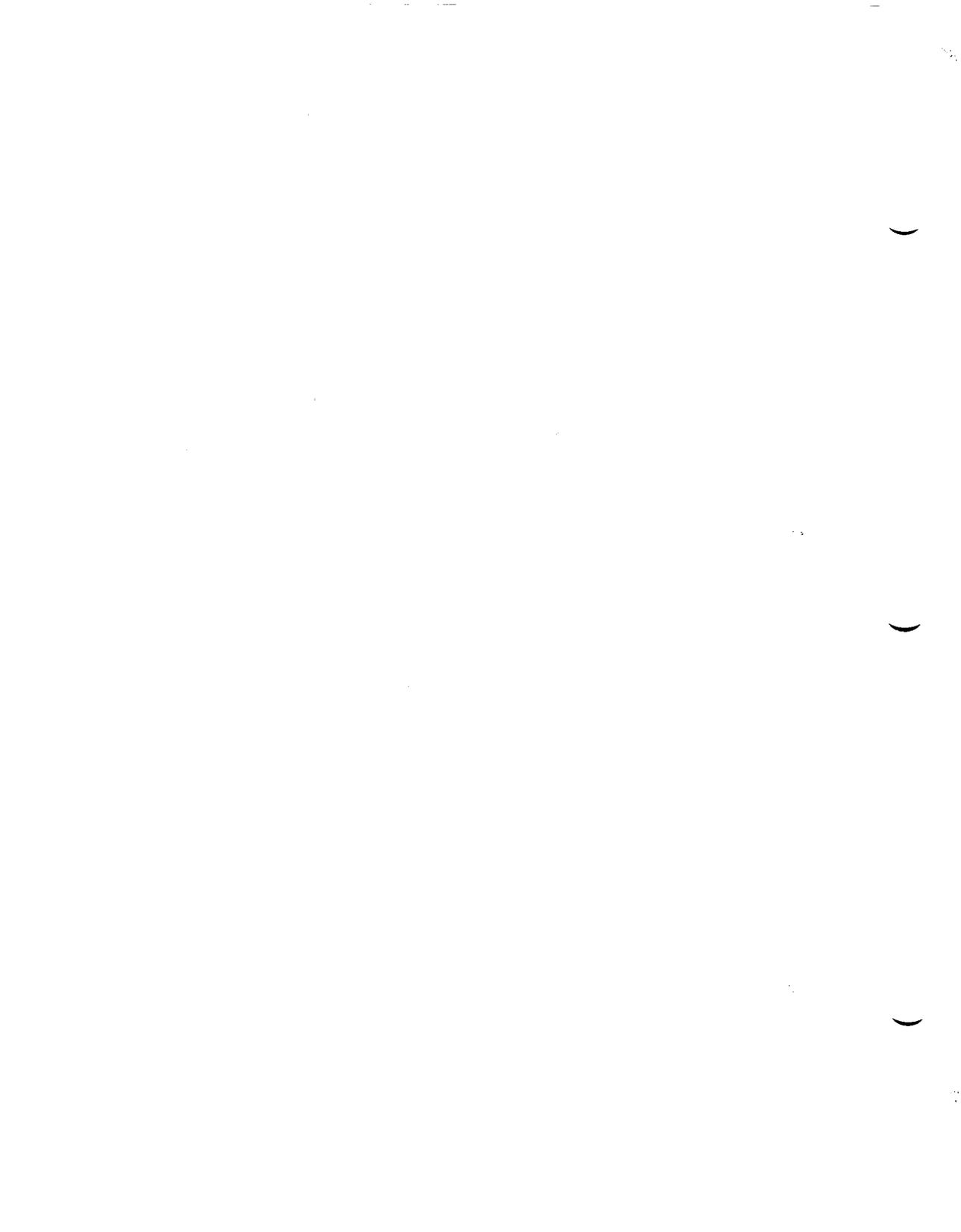
Länderspezifische Zeichen (Fortsetzung)

# ASCII Code Chart

LEGEND: (For Control Characters in USASI Code Set)

- NUL Null
- SOH Start of Heading
- STX Start of Text
- ETX End of Text
- EOT End of Transmission
- ENQ Enquiry
- ACK Acknowledge
- BEL Bell (Audible or Attention Signal)
- BS Backspace
- HT Horizontal Tabulation (Punched Card Strip)
- LF Line Feed
- VT Vertical Tabulation
- FF Form Feed
- CR Carriage Return
- SO Shift Out
- SI Shift In
- DLE Data Link Escape
- DC1 Device Control 1
- DC2 Device Control 2
- DC3 Device Control 3
- DC4 Device Control 4
- NAK Negative Acknowledge
- SYN Synchronous DLE (Sync Code)
- ETB End of Transmission Block
- CAN Cancel (Void Data)
- EM End of Media
- SUB Substitute
- ESC Escape
- FS File Separation (End of File)
- GS Group Separator
- RS Record Separator (End of Record)
- US Unit Separator (End of Field)
- DEL Delete

ASCII CODE CHART																
Binary b <sub>4</sub> -b <sub>1</sub>	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1111				
b <sub>8</sub> -b <sub>5</sub>	0	1	2	3	4	5	6	7	8	9	A	F				
0000	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
0001	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
0010	␣	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
0011	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0100	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0101	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
0110	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0111	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL



# Meldungen

Meldungen kommen von verschiedenen Quellen: CP/M Fehlermeldungen erfolgen bei fehlerhaften Aufrufen des „Basic Disk Operating System“ (BDOS) oder des „Basic Input/Output System“ (BIOS). Ferner meldet sich CP/M bei Fehlern in den Kommandozeilen (Befehlseingaben). Darüber hinaus verfügt jedes Dienstprogramm innerhalb von CP/M über einen eigenen Vorrat an Meldungen. Im folgenden sind die Meldungen des Betriebssystems CP/M und seiner Dienstprogramme zusammengefaßt. Bei Anwenderprogrammen können Meldungen erscheinen, die hier nicht aufgeführt sind. In solchen Fällen sehen Sie bitte in der zu den Anwenderprogrammen gehörenden Dokumentation nach.

?

DDT. Diese Meldung kann vier verschiedene Bedeutungen haben:

- 1) DDT versteht nicht die Instruktionen der Assemblersprache.
- 2) Die Datei läßt sich nicht eröffnen.
- 3) In einer HEX-Datei besteht ein Kontrollsummenfehler.
- 4) Der Assembler/Disassembler wurde überlagert.

## ABORTED

PIP. Sie haben eine PIP-Operation durch Betätigen einer Taste angehalten.

## ASM Fehler Meldungen

- C Kommandofehler: Das Kommando wird nicht erkannt. (Eine Art Syntax-Fehler.)
- D Datenfehler: Der Wert eines Operanden paßt nicht in den angegebenen Datenbereich.
- E Ausdruck falsch: Der Ausdruck ist fehlerhaft aufgebaut. Er kann nicht berechnet werden.
- L Markenfehler: Eine Marke kann in diesem Zusammenhang nicht vorkommen. (Häufige Ursache: Mehrere Anweisungen tragen Marke mit dem gleichen symbolischen Namen.)

- N Nicht implementiert: Der Befehl wird von der verwendeten Assemblerversion nicht akzeptiert.
- O Überlauf: Der Ausdruck ist zu kompliziert, d. h. er enthält zu viele Operatoren, um berechnet werden zu können; er sollte vereinfacht werden.
- P Phasenfehler: Der symbolische Name einer Marke hat bei zwei aufeinanderfolgenden Durchgängen durch das Programm nicht denselben Wert.
- R Registerfehler: Das angegebene Register kann bei diesem Befehl nicht auftreten.
- S Syntax-Fehler: Die Anweisung ist nicht den vorgeschriebenen Regeln entsprechend formuliert.
- U Nicht definierte Marke: Die verwendete Marke wurde nicht gefunden.
- V Wert falsch: Ein im Ausdruck vorgefundener Operand ist unrichtig angegeben.

#### **BAD DELIMITER**

STAT. Prüfe Befehlszeile auf Schreibfehler.

#### **Bad Load**

CCP Fehlermeldung, oder SAVE Fehlermeldung.

#### **Bad Media FORMAT.**

Die Platte ist defekt. Sie kann deshalb nicht formatiert werden.

#### **Bdos Err On d:**

BDOS-Fehler am zugeordneten Plattenlaufwerk: CP/M ersetzt d: mit der Bezeichnung des Laufwerkes, an welchem der Fehler auftrat. Auf diese Meldung erfolgt einer der vier Ausdrücke der nachstehend beschriebenen Situationen.

#### **Bdos Err On d: Bad Sector**

Diese Meldung tritt auf, wenn die Platte nicht ordnungsgemäß formatiert ist und als Reaktion auf die vorher angezeigte BIOS-Meldung „O“ eingegeben wurde. Drücken Sie ↑ C, um das Programm abzuschließen und zum Betriebssystem zurückzukehren. Sie können auch den Fehler ignorieren und CR (↵) drücken.

**Bdos Err On d: File R/O**

Sie versuchten, auf einer Read-Only (nur lesbaren) Datei Dateimerkmale zu löschen, umzubenennen oder neue zu setzen. Die Datei sollte erst mit dem Befehl: „STAT filespec \$R/W“ in den Zustand „Read-Write“ (RW = Lesen/Schreiben) gebracht werden.

**Bdos Er On d: R/O**

Das Laufwerk wurde mit dem STAT-Befehl in den Read Only (nur lesen) Zustand gebracht, oder die Platte im Laufwerk wurde ohne vorherige Initialisierung (mittels ↑ C) geändert. Das CP/M beendet das gegenwärtige Programm, sobald Sie irgendeine Taste drücken.

**Bdos Err On d: Select**

CP/M erhielt den Befehl für ein nichtexistierendes Laufwerk. Das CP/M beendet das gegenwärtige Programm, sobald Sie irgendeine Taste drücken. Drücken Sie zur Wiederholung die Taste CTRL-C oder CR (↵).

**BOOT ERROR (CR)**

BOOT oder WBOOT kann CP/M nicht laden. Drücken Sie ↵ , um neu zu laden, oder wechseln Sie die Platte mit dem Betriebssystem.

**Break „x“ at c**

ED. „x“ ist eines der nachstehend beschriebenen Symbole und c ist das beim Auftreten des Fehlers gerade benutzte Befehlszeichen.

- # Such-Fehler. ED kann keinen „String“ in einem F-, S-, oder N-Befehl finden.
- ? Nicht erkanntes Befehlszeichen c. ED erkennt nicht das Befehlszeichen, oder ein E, H, Q, oder O-Befehl steht nicht allein auf seiner Befehlszeile.
- O Die in einem R-Befehl angesprochene Datei ist nicht auffindbar.

- > Puffer ist voll. ED kann keine weiteren Zeichen in den Speicherpuffer einbringen; oder die Zeichenfolge in einem F-, N-, oder S-Befehl ist zu lang.
- E Befehl zurückgewiesen. Eine Tastenbetätigung hat den Befehl zurückgewiesen.
- F Platte oder Directory (Platteninhaltsverzeichnis) ist voll. Dieser Meldung folgt der Hinweis, ob die Platte oder das Directory voll ist. Beachten Sie hierzu die auf die Meldung folgenden Aufforderungen für die weitere Prozedur.

#### CANNOT CLOSE DESTINATION FILE - {filespec}

PIP. Eine Ausgangsdatei kann nicht abgeschlossen werden. Überzeugen Sie sich, ob sich die richtige Platte im Laufwerk befindet und die Platte nicht schreib-geschützt ist.

#### Cannot close, R/O CANNOT CLOSE FILES

CP/M kann nicht zur Datei schreiben. Dies tritt gewöhnlich auf, wenn die Platte schreib-geschützt ist und die Antwort auf die BIOS Meldung „d: WRITE PROTECTED (R/O/X) „O“ war.

ASM. Eine Ausgangsdatei kann nicht abgeschlossen werden. Dieser schwerwiegende Fehler führt zum Abbruch der ASM. Prüfen Sie, ob sich die Platte im Laufwerk befindet und nicht schreib-geschützt ist.

DDT. Eine mit W-Befehl geschriebene Platten-Datei läßt sich nicht abschließen. Dieser Fehler führt zum Abbruch von DDT. Prüfen Sie, ob sich die Platte im Laufwerk befindet und nicht schreib-geschützt ist.

SUBMIT. Dieser Fehler kann während der SUBMIT-Datei-Bearbeitung auftreten. Prüfen Sie, ob sich die richtige Systemplatte im Laufwerk A befindet und nicht schreib-geschützt ist. SUBMIT kann nach erneutem Laden von CP/M wieder gestartet werden.

#### CANNOT READ

PIP. PIP kann die angesprochene Quelle nicht lesen.

**CANNOT WRITE**

PIP. Das im PIP-Befehl spezifizierte Übertragungsziel ist unzulässig. Möglicherweise haben Sie ein Eingabegerät als Zielgerät angegeben.

**Checksum error**

PIP. Ein Kontrollsummenfehler in einer Hex-Datei wurde festgestellt. Der Fehler muß korrigiert werden; möglicherweise durch Neuerstellung der Hex-Datei.

**CHECKSUM ERROR**  
**LOAD ADDRESS hhhh**  
**ERROR ADDRESS hhhh**  
**BYTES READ:**  
**hhhh:**

LOAD. Datei enthält falsche Daten. Regenerieren Sie die Hex-Datei von ihrer Quelle.

**Command Buffer Overflow**

SUBMIT. Der SUBMIT Puffer erlaubt bis zu 2048 Zeichen in der Eingabe-Datei.

**Command too long**

SUBMIT. Ein Befehl in der SUBMIT-Datei kann nicht 125 Zeichen überschreiten.

**CORRECT ERROR, TYPE RETURN OR CTL-Z**

PIP. Ein Kontrollsummenfehler wurde während der Übertragung einer Hex-Datei festgestellt. Die Hex-Datei mit dem Kontrollsummenfehler sollte korrigiert werden, möglicherweise durch Neuerzeugung der Hex-Datei.

**DESTINATION IS R/O, DELETE (Y/N)?**

PIP. Die in einem PIP-Befehl genannte Ziel-Datei existiert bereits und ist Read Only (nur lesbar). Wenn sie Y eingeben, wird die Ziel-Datei vor Fertigstellung der Kopie gelöscht.

**Directory full**

ED. Auf der Zielplatte ist für die zu schreibende Datei nicht genügend Platz im Directory. Ohne den Editor zu verlassen, können Sie mit dem Befehl OXfilespec unnötige Dateien auf der Platte löschen.

SUBMIT. Für die Verarbeitung von SUBMITs ist zum Schreiben der \$\$\$SUB Datei nicht genügend Platz im Directory. Löschen Sie einige Dateien oder nehmen Sie eine neue Platte und beginnen Sie von Neuem.

DISK ERROR – das Betriebssystem kann nicht modifiziert werden.

CONFIG. CONFIG hat bei einer versuchten Änderung der Konfigurations-Parameter einen Fehler bemerkt. Diese Meldung erfolgt nur, wenn eine permanente Änderung versucht wird. Betätigen Sie ↵ , um CONFIG zu verlassen.

Disk full

ED. Auf der Platte ist nicht genügend Platz für die Ausgangsdatei. Dieser Fehler kann bei W-, E-, H-, oder X-Befehlen auftreten. Passiert dies bei einem X-Befehl, können Sie den Befehl mit vorangestelltem Dateinamen und einem anderen Laufwerk, wiederholen.

DISK READ ERROR – {filespec}

PIP. Die in einem PIP-Befehl genannte Eingangs-Platten-Datei ist nicht einwandfrei lesbar. Dies ist gewöhnlich die Folge eines unerwarteten Dateiendes. Die Datei ist zu korrigieren.

DISK WRITE ERROR – {filespec}

DDT. Während eines W-Befehls kann eine Platte nicht richtig beschrieben werden, weil die Platte möglicherweise voll ist. Löschen Sie einige unnötige Dateien oder verwenden Sie eine andere Platte mit mehr Platz.

PIP. Während eines PIP-Befehls kann eine Platte nicht richtig beschrieben werden, weil die Platte möglicherweise voll ist. Löschen Sie einige unnötige Dateien oder verwenden Sie eine andere Platte mit mehr Platz und beginnen erneut mit PIP.

SUBMIT. Das SUBMIT-Programm kann die \$\$\$SUB Datei nicht auf die Platte schreiben. Löschen Sie einige Dateien, oder verwenden Sie eine neue Platte und versuchen es nochmals.

ERROR: BAD PARAMETER

PIP. Sie haben einen unzulässigen Parameter im PIP-Befehl eingegeben. Machen Sie die richtige Eingabe.

ERROR: CANNOT OPEN SOURCE, LOAD ADDRESS hhhh

LOAD. Kommt zur Anzeige, falls LOAD nicht die angegebene Datei findet oder kein Dateiname angegeben wurde.

ERROR: CANNOT CLOSE FILE, LOAD ADDRESS hhhh

LOAD. Verursacht durch einen Fehlercode, der auf einen BDOS-Funktionsaufruf hin entstand.

ERROR: CANNOT OPEN SOURCE, LOAD ADDRESS hhhh

LOAD. Quell-Datei ist nicht auffindbar. Prüfen Sie das Platten-Directory.

ERROR: DISK READ, LOAD ADDRESS hhhh

LOAD. Verursacht durch einen Fehlercode, der auf einen BDOS-Funktionsaufruf hin entstand.

ERROR: DISK WRITE, LOAD ADDRESS hhhh

LOAD. Ziel-Platte ist voll.

ERROR: INVERTED LOAD ADDRESS, LOAD ADDRESS hhhh

LOAD. Die Adresse eines Records war zu weit von der Adresse eines vorher bearbeiteten Records entfernt. Dies ist eine interne Einschränkung von LOAD, welche sich aber umgehen läßt. Verwenden Sie DDT zum Einlesen der Hex-Datei in den Speicher, dann geben Sie den SAVE-Befehl, um eine Kopie des Speicherinhalts auf die Platte zu bringen.

ERROR: NO MORE DIRECTORY SPACE,  
LOAD ADDRESS hhhh

LOAD. Das Platten-Directory ist voll.

Error on line nnn message

SUBMIT. Das SUBMIT-Programm bringt seine Meldung in dem

vorstehend gezeigten Format, wobei nnn die Zeilennummer der SUBMIT-Datei bedeutet. Beachten Sie die der Zeilennummer folgende Meldung.

#### d: FATAL ERROR (R/O/X)

BIOS. Die Verbindung zum Ein- oder Ausgabegerät kann wegen mangelnder Kompatibilität nicht hergestellt werden, oder es liegt ein Hardware-Fehler vor. Zur Wiederholung drücken Sie R, oder O, um den Fehlerstatus zu BDOS zurückzuführen. Sie können auch das Programm mit X beenden.

#### FILE ERROR

ED. Die Platte oder das Directory ist voll; die Platte kann nicht mehr von ED aufnehmen. Es ist deshalb sinnvoll, sich vor jedem Editieren zu vergewissern, daß genügend Platz auf der Platte zur Verfügung steht.

#### FILE EXISTS

CP/M wurde angesprochen, um eine Datei zu eröffnen oder umzubenennen, unter Verwendung eines Dateinamens, der bereits für eine andere Datei belegt ist. Der bestehende Dateiname ist zu löschen oder ein neuer ist festzulegen.

REN. Der neue Dateiname ist identisch mit einer bereits existierenden Datei. Der Name einer Datei kann nicht in den Namen einer bereits existierenden Datei geändert werden. Falls Sie eine existierende Datei durch eine neuere Version ersetzen wollen, können Sie entweder die existierende Datei neu benennen oder löschen, oder die „PIP-Utility“ anwenden.

#### File exists, erase it

ED. Bei der Zuordnung der Zieldatei auf einer anderen als der Ursprungs-Platte wurde festgestellt, daß der Name der Bestimmungsdatei bereits existiert. Zum Erhalt der Ausgangsdatei ist der Dateiname zu löschen oder eine andere Platte zu verwenden.

**\*\*FILE IS READ/ONLY\*\***

ED. Die zum Aufruf von ED im Befehl genannte Datei ist als „nur lesbar“ gekennzeichnet. Mit ED können Sie sich die Datei nur ansehen, aber nicht verändern.

### File Not Found

CP/M findet die genannte Datei nicht. Prüfen Sie, ob Sie das richtige Laufwerk angesprochen haben, bzw. sich die richtige Platte im Laufwerk befindet.

ED. ED findet die genannte Datei nicht. Prüfen Sie, ob Sie das richtige Laufwerk angesprochen haben, bzw. sich die richtige Platte im Laufwerk befindet.

STAT. STAT findet die genannte Datei nicht. Diese Meldung kann auftreten, falls vergessen wurde, das Laufwerk zu nennen. Ebenso ist das Vorhandensein der richtigen Platte im Laufwerk zu prüfen.

### FILE NOT FOUND – {filespec}

PIP. Die ausgewählte Eingangsdatei existiert nicht.

### Filename required

ED. Ein ED-Befehl wurde ohne Dateiname eingegeben. Der ED-Befehl ist mit darauffolgendem Dateinamen erneut einzugeben.

### FNCT TABLE FULL (CR)

BIOS. Die Summe der Zeichen innerhalb der Funktionstastentabelle ist größer als 236. Zum Löschen der betroffenen Funktion ist die Abschlußtaste zu betätigen.

### hhhh??=dd

DDT. Das ?? zeigt, daß DDT nicht weiß, wie der Hexadezimalwert dd im Zusammentreffen mit der Adresse hhhh in der 8080 Assemblersprache darzustellen ist. dd entspricht nicht dem 8080 Maschinen-Code.

### Insert O. S. Disk in Drive A

CONFIG. Eine permanente Änderung der Konfiguration wird versucht, aber das Laufwerk A enthält keine System-Diskette. Legen Sie die System-Diskette ein. Betätigen Sie ↵, die Änderung wird dann ausgeführt.

### Insufficient memory

DDT. Für die in einem R- oder E-Befehl genannte Datei steht nicht genügend Speicherplatz zur Verfügung.

## Invalid Assignment

STAT. Eine falsche Dateizuordnung oder ein falsches Laufwerk wurde angegeben; oder ein falsch ausgedrückter Gerätename. Auf diese Fehlermeldung kann eine Liste gültiger Dateizuordnungen folgen. Im Falle des Versuches einer ungültigen Laufwerkszuordnung erscheint die Meldung „Use:d:=RO“, welche die richtige Angabe der Laufwerkszuordnung aufzeigt.

## Invalid control character

SUBMIT. Die einzigen gültigen Kontrollzeichen in SUBMIT-Dateien vom Typ SUB sind  $\hat{A}$  bis  $\hat{Z}$ . Beachten Sie, daß in einer SUBMIT-Datei das Kontrollzeichen mittels  $\hat{\quad}$ , ohne Betätigung der CONTROL-Taste, dargestellt ist.

## INVALID DIGIT - {filespec}

PIP. Ein ungültiges Hex-Digit ist während des Lesens einer Hex-Datei aufgetreten. Die Hex-Datei mit dem ungültigen Hex-Digit ist zu korrigieren, eventuell durch Neuerstellung der Hex-Datei.

## d:/O ERROR (R/O/X)

BIOS. Wird ein Datenfehler festgestellt und die Fehler-Wiederholung ist erfolglos (n\*xretry, m\*xrestore), geben Sie R zur Wiederholung der Operation ein; oder O, um den Fehler zu BDOS zurückzugeben, - oder Sie beenden das Programm mit X.

\* Beachte: n und m sind änderbare Werte der Rückschreibe- und Wiederholungszähler (CONFIG-Utility).

## Invalid Disk Assignment

STAT. Kann auftreten, wenn auf die Festlegung des Laufwerks irgendetwas außer =R/O folgt.

## Invalid Disk Select

CP/M empfing eine Befehlszeile mit der Angabe eines nicht existierenden Laufwerks; oder die eingelegte Platte ist nicht ordnungsgemäß formatiert. CP/M beendet das laufende Programm, sobald irgendeine Taste gedrückt wird.

## INVALID DRIVE NAME (Use A or B)

SYSGEN. SYSGEN erkennt nur die Laufwerke A und B als gültig für die Systemgenerierung an.

## Invalid File Indicator

STAT. Tritt auf, wenn nicht RO, RW, DIR oder SYS angegeben wird.

## INVALID FORMAT

PIP. Das Format des PIP-Befehls ist unzulässig. Siehe auch Beschreibung des PIP-Befehls.

## Invalid EXCHANGE-Version (CR)

EXCHANGE. EXCHANGE funktioniert nicht mit dem von Ihnen geladenen Betriebssystem. Betätigen Sie ↵, um EXCHANGE zu verlassen.

**Hinweis:** Ihre EXCHANGE-Version kann nur mit dem dafür vorgesehenen Betriebssystem verwendet werden.

## Invalid CONFIG-Version (CR)

CONFIG. CONFIG funktioniert nicht mit dem von Ihnen geladenen Betriebssystem. Betätigen Sie ↵, um CONFIG zu verlassen.

**Hinweis:** Ihre CONFIG-Version kann nur mit dem dafür vorgesehenen Betriebssystem verwendet werden.

## INVALID HEX DIGIT

LOAD ADDRESS hhhh

ERROR ADDRESS hhhh

BYTES READ:

hhhh

LOAD. Datei enthält falsches Hex-Digit.

## INVALID SEPARATOR

PIP. Ein ungültiges Zeichen wurde als Separator zwischen zwei Eingangsdatei-Namen eingefügt.

## INVALID USER NUMBER

PIP. Es wurde eine Teilnehmernummer größer als 15 angegeben.  
Zulässig ist der Bereich von 0 bis 15.

A:MOUNT O.S. DISK (CR)

BIOS. Geben Sie eine Platte mit dem Betriebssystem in das Laufwerk A und betätigen Sie ↵ , um weiter zu arbeiten.

n?

USER. Eine Teilnehmernummer größer als 15 wurde als Teilnehmerbereichs-Nummer angegeben. Z. B., wenn Sie USER 18 <cr> eingaben, meldet sich der Bildschirm mit 18?

NO DIRECTORY SPACE

ASM. Das Platten-Directory ist voll. Löschen Sie einige Dateien, um Platz für PRN- und HEX-Dateien zu gewinnen. Das Directory kann gewöhnlich 128 Dateinamen aufnehmen.

NO DIRECTORY SPACE – {filespec}

PIP. Für die Ausgangsdatei ist nicht genügend Directory-Platz vorhanden. Sie sollten entweder einige nichtbenötigte Dateien löschen oder eine andere Platte mit mehr Directory-Platz verwenden und PIP erneut durchführen.

NO FILE – {filespec}

DIR, ERA, REN, PIP. CP/M findet die angegebene Datei nicht, oder es existiert keine Datei.

ASM. Die angezeigte Quelle oder die beinhaltete Datei ist vom angezeigten Laufwerk nicht auffindbar.

DDT. Die in einem R- oder E-Befehl genannte Datei ist auf der Platte nicht auffindbar.

NO INPUT FILE PRESENT ON DISK

DUMP. Die angeforderte Datei existiert nicht.

No memory

Zum Laden des angegebenen Programms ist nicht genügend (Puffer?) Speicherplatz verfügbar.

#### NO NCR FORMAT

SYSGEN. SYSGEN hat eine im System nicht brauchbare Formatierung der Diskette im Laufwerk festgestellt. SYSGEN wurde zurückgewiesen.

#### NO SOURCE FILE ON DISK

SYSGEN. SYSGEN kann CP/M nicht finden, weder in Form von CPMxx.com, noch in den System-Spuren der Ursprungs-Platte.

#### NO SOURCE FILE PRESENT

ASM. Der Assembler kann die angegebene Datei nicht finden. Entweder wurde die Dateibezeichnung in der Befehlszeile falsch eingegeben, oder die Datei ist nicht vom Typ ASM.

#### NO SPACE

SAVE. Zu viele Dateien sind bereits auf der Platte. Oder es wurde nicht genügend Platz zum Sichern der Information auf der Platte reserviert.

#### No SUB file present

SUBMIT. Damit SUBMIT einwandfrei arbeitet, muß eine Datei vom Typ SUB erstellt werden. Die SUB-Datei enthält gewöhnliche CP/M-Befehle. Benützen Sie einen Befehl per Zeile.

#### NOT A CHARACTER SOURCE

PIP. Die im PIP-Befehl angegebene Quelle ist unzulässig. Möglicherweise wurde ein Ausgabe-Gerät als Quelle angegeben.

#### \*NOT DELETED\*

PIP. PIP entfernte nicht die Datei, welche das R/O-Kennzeichen haben könnte.

#### NOT FOUND

PIP. PIP findet die genannte Datei nicht.

d:NOT READY (R)  
d: NOT READY (R/X)

BIOS. Diese Meldung tritt auf, wenn CP/M das Fehlen einer Platte im Laufwerk feststellt. Legen Sie eine Platte ein und starten Sie erneut mit R. Andernfalls können Sie mit X Ihr Programm beenden.

#### OUTPUT FILE WRITE ERROR

ASM. Als Bestimmungsort für PRN- und HEX-Dateien haben Sie eine schreib-geschützte Platte angegeben; oder auf der Platte war nicht mehr genügend Platz. Der Fehler ist zu beheben, bevor Sie Ihr Programm assemblieren.

#### PARAMETER error

SUBMIT. Gültige Parameter innerhalb einer SUBMIT-Datei vom Typ Sub sind \$0 bis \$9.

#### PARAMETER ERROR, TYPE RETURN TO IGNORE

SYSGEN. Wenn Sie die Abschlußtaste betätigen, operiert SYS-GEN ohne Berücksichtigung des ungültigen Parameters.

#### QUIT NOT FOUND

PIP. Die angegebene Zeichenfolge zu einem Q-Parameter war in der Eingabe-Datei nicht vorhanden.

#### Read Error

TYPE. Ein Fehler trat beim Lesen der im Type-Befehl angegebenen Datei auf. Prüfen Sie die Platte und versuchen es nochmals. Der STAT Filespec-Befehl kann Fehler untersuchen.

#### Record Too Long

PIP. PIP kann einen Record mit mehr als 128 Bytes nicht verarbeiten.

#### START NOT FOUND

PIP. Die angegebene Zeichenfolge zu einem S-Parameter ist in der Quell-Datei nicht auffindbar.

#### SOURCE FILE INCOMPLETE

SYSGEN. Ihre Quell-Datei ist von SYSGEN nicht verwendbar.

#### SOURCE FILE NAME ERROR

ASM. Beim Assemblieren einer Datei dürfen Sie die für Substitutionszwecke reservierten Zeichen \* und ? nicht im Dateinamen verwenden. Gleichzeitig ist nur eine Datei assemblierbar.

#### SOURCE FILE READ ERROR

ASM. Der Assembler kann die Information der das Assemblersprachen-Programm beinhaltenden Datei nicht verstehen. Teile einer anderen Datei können auf Ihrer Assemblersprachen-Datei überschrieben sein, oder die Informationen wurden auf der Platte nicht richtig gesichert. Zur Lokalisierung des Fehlers benutzen Sie den TYPE-Befehl. Assemblersprachen-Dateien enthalten Buchstaben, Symbole und Ziffern entsprechend der Tastatur des Computers. Falls der Bildschirm einen undefinierbaren, fremdartigen Inhalt zeigt, können Sie sehen, wo sich Computerinstruktionen in die Datei eingeschlichen haben.

#### „SYSTEM“ FILE NOT ACCESSIBLE

Sie versuchten den Zugriff zu einer Datei, die sich durch den STAT-Befehl im SYS-Zustand befand.

#### \*\*TOO MANY FILES\*\*

STAT. Zum Sortieren der angegebenen Dateien steht für STAT nicht genügend Speicherplatz zur Verfügung, oder es wurden mehr als 512 Dateien angegeben.

#### UNEXPECTED END OF HEX FILE - {filespec}

PIP. Vor Abschluß eines Hex-Records ist ein Dateiende aufgetreten. Die Hex-Datei ohne Record-Abschluß ist zu korrigieren, evtl. durch Neuerstellen der Hex-Datei.

#### Unrecognized Destination

PIP. Überprüfen Sie die Befehlszeile auf eine gültige Zielangabe.

Use: STAT d:=RO

STAT. Ein ungültiger STAT-Befehl wurde gegeben. Die einzige gültige Laufwerkszuordnung in STAT ist STAT d:=RO.

VERIFY ERROR: - {filespec}

PIP. Beim Kopieren mit der V-Option fand PIP Unterschiede zwischen den gerade geschriebenen und danach gelesenen und mit dem Speicherpuffer-Inhalt verglichenen Daten. Gewöhnlich weist dies auf Fehler in der Zielplatte oder Laufwerk hin.

d:WRITE PROTECT (R/O/X)

BIOS. Die Platte ist schreib-geschützt. Drücken Sie R zur Wiederholung der Operation (vorher ist die Schreibsicherung zu entfernen), oder drücken Sie O zur Rückgabe des Fehlers an BDOS. Oder Sie beenden das Programm mit X.

Diese Meldung erscheint auch, wenn versucht wird, eine 48 TPI (Spuren pro Zoll) Diskette mit einem 96 TPI Laufwerk zu beschreiben.

XSUB ACTIVE

SUBMIT. XSUB wurde angerufen.

XSUB ALREADY PRESENT

SUBMIT. XSUB ist bereits im Speicher aktiv.

Your input?

Wenn CP/M den angegebenen Befehl nicht findet, kehrt es wieder zum eingegebenen Befehlsnamen zurück, - gefolgt von einem Fragezeichen. Prüfen Sie, ob die Befehlszeile korrekt eingegeben wurde, oder ob der fragliche Befehl als eine .COM-Datei auf der automatisch zugeordneten bzw. explizit angegebenen Platte existiert.

## BEGRIFFE

**Abgrenzungszeichen:** Spezialzeichen, die verschiedene Positionen in einer Kommandozeile gegeneinander abgrenzen. Der Doppelpunkt grenzt z. B. die Laufwerksdefinition von der Dateibezeichnung ab. CCP erkennt die folgenden Zeichen als Abgrenzungszeichen: . := ; < > - , Leerzeichen und CR (Return). Einige CP/M-Kommandos benutzen die Zeichen: , [ ] ( ) \$ als Abgrenzungszeichen. Es ist empfehlenswert, Abgrenzungszeichen sowie Kleinbuchstaben in Bezeichnungen für CP/M-Dateien zu vermeiden.

**Adresse:** Eine Zahl, die die Position einer bestimmten Speicherstelle (Byte) definiert. CP/M kennt zwei Arten von Adressen: logische und physische. Eine physische Adresse bezieht sich auf eine absolute, eindeutige Stelle des Systemspeichers. Eine logische Adresse gibt den Abstand in Relation zu einer Basis-Adresse an. Ein CP/M-Programm wird normalerweise bei der Adresse 0100H in den Systemspeicher geladen. Dies ist auch die Basis-Adresse. Die erste Programminstruktion hat die physische Adresse 0100H und die logische Adresse (= Offset) 0H.

**afn:** Siehe Mehrdeutige Dateibezeichnung

**Aktiviert:** Ein Laufwerk ist aktiviert, wenn es vom Benutzer oder einem Programm angewählt wird. Es bleibt solange aktiviert, bis beim Diskettenlaufwerk die Diskette gewechselt wird oder ctl-C als Kommandozeile eingegeben wird, oder bis eine BDOS-Funktion 0 ausgeführt wird.

**AL0, AL1:** Zwei Bytes im Platten-Parameterblock, die Platz für das Plattenverzeichnis (Directory) reservieren. Diese beiden Bytes werden in die beiden ersten Bytes des Allocation Vectors kopiert, wenn ein Laufwerk aktiviert wird (siehe Allocation Vector).

**Allocation Vector (ALV):** Ein Allocation Vector = Zuordnungsvektor wird im BIOS für jedes aktivierte Laufwerk bereitgestellt. Ein Vektor besteht aus einer Reihe von Bits (eines für jeden Block im Plattenlaufwerk). Das einem Block entsprechende Bit wird auf 1 gesetzt, wenn der Block zugeordnet wurde, andernfalls wird es auf 0 gesetzt. Die ersten beiden Bytes dieses Vektors werden mit den Bytes AL0 und

ALI initialisiert und ordnen so die Blöcke des Plattenverzeichnisses zu. Funktion 27 ergibt die Adresse des Allocation Vectors.

ALV: Siehe Allocation Vector

ambiguous File Name: Siehe mehrdeutige Dateibezeichnung.

American Standard Code for Information Interchange: Siehe ASCII.

Anwendungsprogramm: Ein Programm zur Lösung eines speziellen Problems. Typische Anwendungsprogramme sind Finanzbuchhaltungs-, Textverarbeitungs- und Adreßlistenprogramme.

Archiv-Attribut: Ein Datei-Attribut, das durch das höchstwertige Bit des Bytes t3 (FCB+11) in einem Verzeichniselement gesteuert wird. Dieses Attribut wird gesetzt, wenn die Datei archiviert wurde.

Argument: Ein Zeichen, üblicherweise ein Buchstabe, das eine Stelle anzeigt, die später durch eine Zahl, einen Buchstaben oder einen Namen ersetzt wird, um einer Formel eine entsprechende Bedeutung zu geben.

ASCII: American Standard Code for Information Interchange. ASCII ist ein Standardsatz von 7-Bit numerischen Zeichencodes, die benutzt werden, um Zeichen im Speicher darzustellen. Jedes Zeichen benötigt ein Byte im Speicher. Das höchstwertige Bit ist normalerweise auf Null gesetzt. Der Zeichensatz umfaßt Ziffern, Buchstaben und Symbole. Eine ASCII-Datei kann lesbar auf dem Bildschirm dargestellt oder gedruckt werden.

Assembler: Ein Programm das die Assemblersprache in den binären Maschinencode übersetzt. Die Assemblersprache besteht aus einer Reihe mnemonischer Begriffe, die den Befehlssatz des Prozessors ansprechen (siehe Abschnitt 4 dieses Handbuchs).

aufwärts kompatibel: Ausdruck, der aussagt, daß ein Programm, das für eine frühere Version eines Betriebssystems (oder Kompilierers, usw.) auch unter der neuesten Version des gleichen Betriebssystems läuft.

ausführbar: Ausführbarer Code besteht aus einer Reihe von Instruktionen, die vom Computer verarbeitet werden können. Ein Computer kann z. B. keine Namen und Adressen ausführen, kann jedoch ein

Programm ausführen, das all diese Namen und Adressen auf Aufkleber druckt.

**Ausführen eines Programmes:** Die Verarbeitung eines ausführbaren Programmes einleiten.

**Ausgabe:** Daten, die an die Konsole, Platte oder den Drucker gesandt werden.

**Automatische Zuordnung:** Siehe Vorgabe, Vorgabe-FCB, Vorgabepuffer.

**Back-up:** Die Sicherheitskopie einer Platte oder Datei bzw. der Kopiervorgang für eine Platte oder Datei.

**Basic Disk Operating System:** Siehe BDOS.

**Basic Input/Output System:** Siehe BIOS.

**BDOS:** Basic Disk Operating System. Das Modul BDOS des Betriebssystems CP/M bildet die Schnittstelle zwischen einem Anwendungsprogramm und dem Betriebssystem. Diese Schnittstelle besteht aus einem Satz von Funktionsaufrufen die ans BDOS durch Aufrufe an die Adresse 0005H in Seite Null (Page Zero) gehen. Anwendungsprogramme für CP/M sollten die BDOS-Funktionen für alle Ein-/Ausgabevorgänge benutzen um mit anderen CP/M-Systemen und zukünftigen CP/M-Versionen kompatibel zu sein. BDOS ist normalerweise im oberen Speicherbereich direkt unterhalb von BIOS angeordnet.

**Bedienerführung:** Jedes auf dem Bildschirm angezeigte Zeichen, das dem Bediener hilft, zu entscheiden, was als nächstes zu tun ist. Ein System-Bereitschaftszeichen ist eine spezielle Bedienerführung, die vom Betriebssystem angezeigt wird. Siehe CP/M-Bereitschaftszeichen. Der Buchstabe bezeichnet das vorgegebene Laufwerk. Einige Anwendungsprogramme haben eigene Bereitschaftszeichen.

**Benutzernummer:** (user Number). Nummer, die den Dateien eines Verzeichnisses zugeordnet ist, so daß jeder Benutzer nur mit seinen eigenen Dateien arbeitet und sein „eigenes“ Verzeichnis hat, obwohl alle mit derselben Platte arbeiten. In CP/M können Dateien auf 16 Benutzergruppen verteilt werden.

**Betriebssystem:** Eine Ansammlung von Programmen, welche die

Ausführung anderer Programme und die Verwaltungen der Systemelemente des Computers überwachen. Ein Betriebssystem schafft systematische Ein-/Ausgabebedingungen zwischen dem Computer und dessen Peripheriegeräten. Es ermöglicht die sichere Ausführung von Anwender-Programmen. Ein Betriebssystem standardisiert den Gebrauch der Computer-Systemelemente für die unter ihm laufenden Programme.

**Binär:** Ein Zahlensystem auf der Basis 2. Eine Binärziffer kann zwei Werte haben, 0 oder 1. Binärzahlen finden in Computern deshalb Verwendung, weil sich die beiden Werte durch die Hardware leicht als EIN und AUS darstellen lassen. Ein Bit im Speicher stellt eine Binärziffer dar.

**BIOS:** Basic Input/Output System. Das Modul BIOS ist das einzige hardware-abhängige Modul im Betriebssystem CP/M. Es unterstützt BDOS mit einer Reihe einfacher Ein-/Ausgabevorgängen. BIOS ist ein Assembler-Modul das normalerweise vom Systemhersteller, einem unabhängigen Softwarehaus oder dem Benutzer geschrieben wird. Es ist der Grund für die vielseitige Anwendbarkeit von CP/M. BIOS bietet die Schnittstelle von CP/M zur verwendeten Hardware durch eine standardisierte Sprungtabelle am Beginn von BIOS und eine Reihe von Plattenparameter-Tabellen, die die verwendeten Platten und Laufwerke definieren. Dadurch bietet BIOS für CP/M ein komplett tabellengesteuertes Ein-/Ausgabesystem.

**BIOS-Basis:** Niedrigste Speicheradresse des BIOS-Moduls, das auch gleichzeitig der Anfangspunkt der BIOS Sprungtabelle ist.

**BLM:** Siehe Blockmaske.

**Block:** Kleinste zuordenbare Plattenspeichereinheit. Jedes Laufwerk hat eine feste Blockgröße (BLS), die in seinem Plattenparameterblock in BIOS festgehalten ist. Ein Block kann aus 1K, 2K, 4K, 8K oder 16K zusammenhängender Bytes bestehen. Die Blöcke sind relativ zu Null nummeriert, so daß jeder Block eine eindeutige Nummer hat. Die Byte-Entfernung eines Blockes innerhalb einer Datei entspricht der Anzahl der Blöcke multipliziert mit der Blockgröße.

**Blockmaske (BLM):** Bytewert im Plattenparameterblock in DPB+3. Die Blockmaske ist immer Eins weniger als die Anzahl der 128-Byte Sektoren in einem Blocke. Beachten:  $BLM = (2 * BSH - 1)$ .

**Blockshift (BSH):** Byte-Parameter im Plattenparameterblock auf

DPB+2. Die Werte für Blockshift und Blockmaske (BLM) werden durch die Blockgröße (BLS) bestimmt. Beachten:  $BLM = (2 ** BSH - 1)$ .

**Blockungs- und Entblockungsalgorithmus:** Einige Plattenlaufwerke haben eine Sektorgröße von mehr als 128 Bytes. Üblich sind 256, 512, 1024 oder 2048 Bytes. Wenn das System eine Sektorgröße über 128 Bytes hat, müssen die Sektoren im Speicher gepuffert werden und die CP/M-Sektoren von 128 Bytes müssen durch ein zusätzliches Modul geblockt und entblockt werden. Das Modul 'Blockungs- und Entblockungsalgorithmus' ist zwischen den BIOS Ein-/Ausgaberroutinen und der wirklichen Plattenein-/ausgabe. Die Sektorgröße des Systems muß ein geradzahliges Mehrfaches von 128 Bytes sein, damit der Algorithmus richtig funktioniert. Dieses Modul ermöglicht es, daß BDOS und BIOS genauso arbeiten, als wenn die ganze Platte aus 128-Byte-Sektoren wie im Standard CP/M-System bestünde.

**BLS:** Blockgröße in Bytes. Siehe Block.

**Boot:** Siehe Start.

**BSH:** Siehe Blockshift.

**BTREE:** Mehrzweckmethode für Dateizugriffe, die zur Standardorganisation für Indizes in großen Datenbanksystemen wurde. BTREE bietet fast optimale Ausführung fast aller Dateivorgänge wie Einfügen, Löschen, Suchen, usw.

**Buffer:** Siehe Puffer.

**Built-in Commands:** Siehe Residente Kommandos.

**Byte:** Eine Speicher- oder Plattenspeichereinheit, die aus 8 Bits besteht. Ein Byte kann eine Binärzahl von 0 bis 255 darstellen und ist die kleinste adressierbare Speichereinheit bei 8-Bit-Prozessoren wie dem Intel 8080 oder dem Zilog Z-80.

**CCP:** Console Command Processor. Der CCP ist ein Modul des Betriebssystems CP/M. Es wird direkt unterhalb des Moduls BDOS geladen und interpretiert Befehle, die vom Anwender an der Konsole eingegeben werden, und führt sie aus. Normalerweise sind diese Befehle Programme, die CCP lädt und aufruft. Nach Beendigung eines Programmes kann dieses, falls CCP nicht überschrieben wurde, die Steuerung direkt an CCP zurückgeben. Andernfalls kann das Programm CCP wieder in den Speicher einlesen. Dies geschieht

entweder durch einen Warmstart, eingeleitet durch einen Sprung auf Null (BDOS System Reset, Funktion 0), oder durch einen Kaltstart. Außer seiner Anordnung im Speicher funktioniert CCP wie irgendein anderes CP/M-Programm; d. h. es erledigt alle Ein-/Ausgabevorgänge durch BDOS-Funktionsaufrufe.

CCP-Basis: Niedrigste Speicheradresse von CCP. Diese Bezeichnung gibt manchmal auch die Basis des CP/M-Systems im Speicher an, da CCP normalerweise das unterste CP/M-Modul im oberen Speicherbereich ist.

checksum Vector (CSV): Siehe Kontrollsummen-Vektor.

CKS: Anzahl der Verzeichniseinträge, die bei einem Verzeichniszugriff die Kontrollsumme bilden. Dies ist ein Parameter des Platten-Parameterblockes in BIOS. Wenn der Wert von CKS Null ist, werden keine Verzeichniseinträge kontrolliert. CKS ist auch ein Parameter in der „Diskdef Macro library“, wo es die aktuelle Anzahl der zu kontrollierenden Verzeichniseinträge anstatt der Anzahl der Verzeichnissätze angibt.

Cold Boot: Siehe Start und Kaltstart.

COM: Dateityp einer CP/M-Kommandodatei. Siehe Kommandodatei.

Command: Siehe Kommando.

Command File: Siehe Kommandodatei.

Command Key Word: Siehe Kommandowort.

Command Syntax: Siehe Kommando-Satzaufbau.

Command Tail: Siehe Kommando-Anhang.

CON: Mnemonischen Begriff für die CP/M-Konsole (siehe Konsole). Das CP/M-Kommando „PIP CON: = TEST.SUB“ zeigt z. B. die Datei TEST.SUB auf dem Bildschirm an. Die Erklärung zum Kommando STAT zeigt, wie das logische CON: verschiedenen physischen Geräten zugewiesen werden kann.

Concatenate: Siehe Verkettung

Concurrency: Siehe Parallele Operationen.

CONIN: Eintrittspunkt für eine BIOS-Routine, die ein Zeichen vom Konsolgerät einliest.

CONOUT: Eintrittspunkt für eine BIOS-Routine, die ein Zeichen ans Konsolgerät sendet.

Console: Siehe Konsole.

Console Command Processor: Siehe CCP.

CONST: Eintrittspunkt für eine BIOS-Routine, die den Konsolstatus ermittelt.

Control Character: Siehe Control-Zeichen.

Control Program for Microcomputers: Siehe CP/M.

Control-Zeichen: Nichtdruckende Zeichenkombination. CP/M nimmt einige Control-Zeichen als einfache Kommandos z. B. für Zeilenkorrektur-Funktionen. Um ein Control-Zeichen einzugeben, muß die Control-Taste niedergehalten werden, während die gewünschte Zeichentaste betätigt wird.

CP/M: Kontrollprogramm für Mikrocomputer. Ein Betriebssystem, das alle Systemelemente eines Computers verwaltet und eine Standard-Schnittstelle für Software-Produkte, die für eine Vielzahl von Mikrocomputersysteme geschrieben wurden, bietet.

CP/M-Bereitschaftszeichen: Zeichen, die anzeigen, daß CP/M bereit ist, das nächste Kommando anzunehmen. Das CP/M-Bereitschaftszeichen besteht aus einem Großbuchstaben (A-P) gefolgt vom Zeichen „>“; z. B.: A >. Der Buchstabe bezeichnet das augenblicklich angemeldete Laufwerk. CP/M sucht auf diesem Laufwerk nach der spezifischen Kommandodatei, außer, es ist ein residentes Kommando oder es wird durch eine Laufwerksangabe angeführt; z. B.: B:STAT.

CP/M Prompt: Siehe CP/M-Bereitschaftszeichen.

CP/NET: Ein Netzwerk-Betriebssystem von Digital Research, welches es Mikrocomputer ermöglicht, auf gemeinsame Ressourcen über ein Netzwerk zuzugreifen. CP/NET besteht aus MP/M-Primärgerä-

ten und CP/M-Sekundärgeräten mit einer Netzwerk-Schnittstelle dazwischen.

**CSV:** Siehe Kontrollsummen-Vektor.

**Cursor:** Ein-Zeichen-Symbol, welches an jeder beliebigen Bildschirmposition auftreten kann. Der Cursor zeigt die Position am Bildschirm an, an der die nächste Tastenbetätigung effektiv wird.

**Data File:** Siehe Datendatei.

**Datei:** Ansammlung von Zeichen, Instruktionen oder Daten, die mit einer gemeinsamen, eindeutigen Bezeichnung versehen werden können. Dateien werden gewöhnlich auf magnetischen Medien wie Platten, Disketten oder Magnetbändern gespeichert. Eine CP/M-Datei wird durch eine Dateispezifikation identifiziert und ist auf einer Platte als eine Ansammlung von Null bis zu 65.536 Sätzen. Jeder Satz ist 128 Bytes lang und kann entweder Binärdaten oder ASCII-Daten enthalten. Binärdateien enthalten Datenbytes mit der Wertigkeit von 0H bis 0FFH. ASCII-Dateien enthalten Folgen von Zeichencodes abgeschlossen durch eine Kombination von CR und Zeilenschaltung. Die Wertigkeit der Bytes liegt normalerweise zwischen 0H und 7FH. Das Verzeichnis ordnet die Dateien als eine Reihe von physischen Blöcken an. Obwohl Dateien als eine Reihe von zusammenhängenden logischen Sätzen definiert sind, müssen diese Sätze nicht in aufeinanderfolgenden Sektoren auf der Platte abgespeichert sein (siehe auch Block, Verzeichnis, Extent, Satz, Sektor).

**Dateibezeichnung:** Bezeichnung für eine Datei. Eine Dateibezeichnung kann eine Primärbezeichnung von 1 - 8 Zeichen und einen Dateityp von 0 - 3 Zeichen enthalten. Die Primärbezeichnung und der Dateityp werden durch einen Punkt voneinander getrennt.

**Dateigruppensymbole:** Spezielle Zeichen, die für bestimmte, spezifizierte Angaben stehen. In CP/M gibt es die beiden Dateigruppensymbole: ? und \*. Das ? kann für jedes beliebige Einzelzeichen eingesetzt werden, das \* kann statt der primären Dateibezeichnung, dem Dateityp oder beiden eingesetzt werden. Durch Einsetzen von Dateigruppensymbolen wird eine mehrdeutige Dateibezeichnung erzeugt und man kann einfach eine oder mehrere Dateien ansprechen.

**Dateikontrollblock (FCB):** Struktur für den Dateizugriff auf der Platte. Enthält Informationen über das Laufwerk, die Dateibezeichnung, den Dateityp sowie weitere Informationen die eine zu erstel-

lende oder zu verarbeitende Datei auf der Platte beschreiben. Ein Dateikontrollblock besteht aus 36 aneinanderhängenden Bytes, die durch den Benutzer für Dateiein-/ausgabefunktionen spezifiziert werden. FCB kann sich auch auf ein Verzeichniselement im Verzeichnis des zugeordneten Diskbereiches beziehen. Diese enthalten die ersten 32 Bytes des FCB, jedoch nicht die Bytes für den laufenden Satz und die wahlfreie Satznummer.

**Dateispezifikation:** Eindeutige Datei-Identifikation. Eine komplette Dateispezifikation in CP/M enthält eine Laufwerksspezifikation gefolgt von einem Doppelpunkt (:), eine primäre Dateibezeichnung von 1 - 8 Zeichen, einen Punkt und einen Dateityp von 0 - 3 Zeichen. Eine komplette CP/M-Dateispezifikation wäre z. B.: b:example.tex.

**Dateityp:** Anhang an eine Dateibezeichnung. Kann 0 - 3 Zeichen umfassen und wird von der primären Dateibezeichnung durch einen Punkt getrennt. Der Dateityp kann eine Aussage über die Datei machen. Einige Programme verlangen, daß die zu verarbeitenden Dateien bestimmte Dateitypen haben.

**Datendatei:** Eine Datei mit Informationen, die von einem Programm verarbeitet werden.

**Default:** Siehe Vorgabe.

**Default Buffer:** Siehe Vorgabepuffer.

**Default FCB:** Siehe Vorgabe-FCB.

**Delimiter:** Siehe Abgrenzungszeichen.

**Dienstprogramm:** (Utility) „Werkzeug“. Programm, das dem Benutzer die Ausführung bestimmter Aufgaben wie z. B. das Kopieren, das Löschen und das Editieren von Dateien ermöglicht. Dienstprogramme wurden zur Erleichterung für Programmierer und Anwender geschaffen.

**DIR-Attribut:** Dateiattribut. Eine Datei mit dem DIR-Attribut kann durch ein DIR-Kommando angezeigt werden. Die Datei kann nur von der vorgegebenen Benutzernummer und dem vorgegebenen Laufwerk zugegriffen werden.

**DIRBUF:** 128-Byte Arbeitspuffer für Verzeichnis-Operationen, normalerweise am Ende vom BIOS bereitgehalten. Wird von BDOS bei

Verzeichnis-Operationen benutzt. DIRBUF wird auch die Zwei-Byte-Adresse im Plattenparameter-Header auf DP-Base + 8 genannt.

Directory: Siehe Verzeichnis.

Directory Element: Siehe Verzeichniselement.

Directory Entry: Siehe Verzeichniseintrag.

Disk, Diskette, Platte: Magnetisches Medium das für Massenspeicherung in einem Computersystem dient. Programme und Daten werden ähnlich aufgezeichnet, wie Musik auf einer Kassette. Das Betriebssystem CP/M muß von der Platte gelesen werden, wenn der Computer eingeschaltet wird. Diskette bezeichnet die flexible Platte mit geringerer Kapazität, während mit Disk oder Platte sowohl eine Diskette, als auch eine Wechsel- oder Festplatte (Winchester Disk) bezeichnet wird. Festplatten haben eine Kapazität von fünf bis zu mehreren hundert Millionen Bytes (Megabytes).

Diskdef Macro Library: Eine „Bücherei“ von Codes, die zusammen mit MAC (dem Makroassembler von Digital Research) die Plattendefinitionstabellen wie DPB und DPH automatisch erstellt.

Disk Drive: Siehe Plattenlaufwerk.

Disk Parameter Block (DPB): Datenstruktur auf die einer oder mehrere Disk Parameter Header hinweisen. Der Disk Parameter Block definiert Plattencharakteristika in den folgenden Feldern:

SPT	Gesamtanzahl der Sektoren pro Spur
BSH	Der Blockshift-Faktor
BLM	Die Blockmaske
EXM	Die Extent-Maske, die von BLS und DRM
DSM	maximale Anzahl von Blöcken
DRM	maximale Anzahl von Verzeichniseinträgen - 1
AL0	reserviert Verzeichnisblöcke
AL1	reserviert Verzeichnisblöcke
CKS	Anzahl der kontrollierten Verzeichnissektoren
OFF	Anzahl der reservierten Systemspuren

Die Adresse des Disk Parameter Blocks ist auf der Position DP-Basis + 0AH im Disk Parameter Header zu finden. Laufwerke mit denselben Charakteristika können denselben Header und damit denselben DCB verwenden. Laufwerke mit anderen Charakteristika müssen

jeweils einen eigenen Header und Disk Parameter Block haben. Wenn BDOS den Eintrittspunkt von SELDSK in BIOS aufruft, muß SELDSK die Adresse des DCB für dieses Laufwerk im Register HL zurückgeben.

Disk Parameter Header (DPH): Datenstruktur, die Informationen über das Laufwerk enthält und einen Arbeitspuffer für bestimmte BDOS-Operationen bereitstellt. Der DPH enthält sechs Bytes Arbeitspuffer für BDOS und die folgenden fünf 2-Byte-Parameter:

XLT	Adresse der Sektor Übersetzungstabelle
DIRBUF	Adresse des Verzeichnispuffers
DPB	Adresse des Disk Parameter Blocks
CSV	Adresse des Kontrollsummen-Vektors
ALV	Adresse des Allocation Vectors.

Bei  $n$  Laufwerken sind die Disk Parameter Header in einer Tabelle angeordnet, deren erste 16-Byte-Zeile zu Laufwerk 0 und deren letzte Zeile zu Laufwerk  $n-1$  gehört.

DKS: Parameter in der „Diskdef Macro Library“, der die Anzahl der Datenblocks auf dem Laufwerk angibt.

DMA: Direkter Speicherzugriff (Direct Memory Access). DMA ist eine Methode der direkten Übertragung von Daten zwischen Platte und Speicher. In einem CP/M-System ruft BDOS den BIOS-Startpunkt READ um einen Plattensektor in die momentan angewählte DMA-Adresse zu lesen. Die DMA-Adresse muß auf einen 128-Byte-Sektor verweisen. Dies kann entweder der automatisch zugeordnete auf 0080H in Seite Null sein, oder ein vom Benutzer definierter Puffer im TPA. GleichermäÙe werden Daten auf die Platte geschrieben, in dem BDOS die Routine WRITE im BIOS anspringt.

DN: Parameter in der „Diskdef Macro Library“, der die logische Laufwerksnummer spezifiziert.

DPB: Siehe Disk Parameter Block.

DPH: Siehe Disk Parameter Header.

DRM: 2-Byte-Parameter im Disk Parameter Block auf DPB+7. DRM entspricht der maximalen Anzahl von Verzeichniseinträgen minus Eins. Dieser Wert ist bezogen auf die DBP-Bytes AL0 und AL1, welche bis zu 16 Blöcke für Verzeichniseinträge pro Laufwerk zur Verfügung stellen.

**DSM:** 2-Byte-Parameter des Disk Parameter Blocks bei DPB+5. DSM gibt die maximale Anzahl von Datenblocks im Laufwerk an. Das Produkt aus BLS \* (DSM+1) ergibt die Gesamtzahl von Bytes auf dem Laufwerk. Diese Zahl darf nicht die Kapazität der physischen Platte abzüglich der reservierten Systemspuren übersteigen.

**Editor:** Dienstprogramm zur Erstellung und Modifikation von Textdateien. Ein Editor kann sowohl zur Erstellung von Dokumenten, als auch zur Erstellung von Codes für Computerprogramme benutzt werden. Der CP/M-Editor wird durch das Kommandowort ED aufgerufen.

**Ein-/Ausgabe:** Bezieht sich auf Ein-/Ausgabevorgänge oder auf Routinen, welche die Eingabe und Ausgabe von Daten in einem Computersystem durchführen.

**Eingabe:** Daten, die in den Computer kommen. Normalerweise kommen sie von einer Bedieneingabe an der Tastatur oder werden von einem Programm von der Platte gelesen.

**Entblockung:** Siehe Blockungs- und Entblockungsalgorithmus.

**EX:** Extent Nummernfeld in einem FCB. Siehe Extent.

**Executable:** Siehe ausführbar.

**EXM:** Siehe Extent Maske.

**Extent:** Ein zusammenhängender Teil einer Datei in der Größe von 16K Bytes. Die Extents werden mit 0 bis 31 numeriert. Ein Extent kann aus 1, 2, 4, 8 oder 16 Blöcken bestehen. EX ist das Feld mit der Extentnummer im FCB. Es ist 1 Byte lang und an Position FCB+12, wobei FCB das erste Byte im FCB sei. Abhängig von der Blockgröße (BLS) und der maximalen Anzahl von Datenblocks (DSM) kann ein FCB 1, 2, 4, 8 oder 16 Extents enthalten. Das Feld EX wird normalerweise durch den Benutzer auf Null gesetzt, enthält jedoch die laufende Extentnummer während Ein-/Ausgabeoperationen auf der Datei. Mit 'FCB folding' werden FCB's mit mehr als einem Extent bezeichnet.

**Extent Maske (EXM):** Ein 1-Byte-Parameter im Disk Parameter Block auf DPB+3. Der Inhalt von EXM hängt von der Blockgröße (BLS) ab, und auch davon, ob die maximale Anzahl von Blöcken (DSM) 255 überschreitet. Pro Verzeichnis-FCB existieren EXM+1 Extents.

FCB: Siehe Dateikontrollblock.

**Festplatte:** Feste, magnetische Scheibe, die in einem Behälter versiegelt ist. Eine Festplatte kann mehr Information speichern als eine Diskette.

File: Siehe Datei.

File Control Block: Siehe Dateikontrollblock.

Filename: Siehe Dateibezeichnung.

File Specification: Siehe Dateispezifikation.

Filetype: Siehe Dateityp.

**Floppy Disk:** Flexible Magnetplatte zur Informationsspeicherung. Gängige Durchmesser sind 8 und 5 1/4 Zoll. Siehe Disk, Diskette, Platte.

**FSC:** Parameter in der „Diskdef macro library“, der die Nummer des ersten physischen Sektors angibt. Dieser Parameter wird zur Ermittlung von SPT und Erstellung von XLT benötigt.

Hard Disk: Siehe Festplatte.

Hardware: Die physischen Bestandteile eines Computers.

**Hexadezimale Aufzeichnung:** Aufzeichnung von Werten auf der Basis 16, wobei die Dezimalziffern und die Buchstaben A - F zur Darstellung der 16 Ziffern dienen. Die hexadezimale Aufzeichnung wird häufig zur Darstellung von Binärzahlen benutzt. Eine Binärzahl wird in Gruppen zu je 4 Bits beginnend beim niedrigstwertigen Bit aufgeteilt. Jede Gruppe wird als eine hexadezimale Ziffer dargestellt (0-F). So wird der Bitwert 1011 zu 0BH und 10110101 wird zu B5H.

**Hex-Datei:** Darstellung einer Kommandodatei (in Maschinensprache), die in ASCII-Format ausgegeben werden kann.

**Hex-Dateiformat:** Die absolute Ausgabe von ASM und MAC für den Intel 8080 ist eine Datei im Hex-Format, welche eine Folge von absoluten Sätzen enthält, die die Ladeadresse und die zu speichernden Bytewerte beginnend mit der Ladeadresse enthalten.

Hex File: Siehe Hex-Datei.

Hex File Format: Siehe Hex-Dateiformat.

Home: BIOS-Eintrittspunkt, der den Lese-/Schreibkopf des momentan selektierten Laufwerks in Ruheposition bringt (Spur 0).

Host: Physische Charakteristika eines Festplattenlaufwerkes in einem System, das den Blockungs- und Entblockungsalgorithmus benutzt. Die Bezeichnung „Host“ hilft, die physischen Hardware-Charakteristika von CP/M's logischen Charakteristika zu unterscheiden. So kennt z. B. CP/M nur 128-Byte-Sektoren, obwohl die Host-Sektorgröße ein Vielfaches von 128 Bytes sein kann.

Input: Siehe Eingabe.

Input/Output: Siehe Ein-/Ausgabe.

Interface: Siehe Schnittstelle.

IOBYTE: Ein 1-Byte-Feld in Seite Null (Page zero), momentan auf Position 0003H, das die Zuordnung von logischen zu physischen Geräten verzeichnet. Das IOBYTE wird gesetzt durch das Kommando:

STAT <logisches Gerät> = <physisches Gerät>

Die logischen Geräte von CP/M sind CON:, RDR:, PUN: und LST:. Jedes von ihnen kann einem von vier physischen Geräten zugeordnet werden. Das IOBYTE wird durch den Eintrittspunkt BOOT in BIOS initialisiert und durch die BIOS-Ein-/Ausgabe-Eintrittspunkte CONST, CONIN, CONOUT, LIST, PUNCH und READER interpretiert. Abhängig vom Inhalt des IOBYTE können verschiedene Ein-/Ausgabe-Treiber durchs BIOS selektiert werden. So könnte beispielsweise LST:=TTY: eine LIST-Ausgabe an einen seriellen Kanal schicken, während LST:=LPT: die LIST-Ausgabe an einen parallelen Kanal schicken würde.

I/O: Abkürzung für Input/Output. Siehe Ein-/Ausgabe.

Jederzeit verwendbare Prozedur (Reentrant Procedure): Code, der von einer Prozedur aufgerufen werden kann, während er von einer anderen Prozedur bereits ausgeführt wird. Solcher Code kann von mehreren Benutzern geteilt werden. Er darf sich nicht selbst modifi-

zieren; d. h. dieser Code darf nur Instruktionen, keine Daten enthalten. Daten für solche Prozeduren können in einem separaten Datenbereich oder im Rücksprungbereich (Stack) gespeichert werden.

K: Abkürzung für Kilobyte. Siehe Kilobyte.

Kaltstart: Siehe Start. Kaltstart (cold boot) kann auch auf den Startpunkt in der Sprungtabelle von BIOS verweisen.

Keyword: Siehe Kommandowort.

Kilobyte (K): 1024 bzw. 0400H Bytes Speicherplatz. Dies ist eine Standardeinheit für Speicherplatz. Der Intel 8080 unterstützt z. B. bis zu 64K Speicheradressen (= 65 536 Bytes). 1024 Kilobyte entsprechen einem Megabyte bzw. über eine Million Bytes.

Kommando: CP/M-Kommandozeile. Eine CP/M-Kommandozeile besteht im allgemeinen aus drei Teilen, dem Kommandowort, dem Kommando-Anhang und einer ↵ -Tastenbetätigung. Um ein Kommando auszuführen, muß eine CP/M-Kommandozeile unmittelbar nach dem Bereitschaftszeichen A> auf der Konsole eingegeben werden und mit der ↵ -Taste abgeschlossen werden.

Kommando-Anhang: Der Teil eines Kommandos, der dem Kommandowort in einer Kommandozeile folgt. Der Kommando-Anhang kann eine Laufwerksspezifikation, eine Dateibezeichnung und/oder einen Dateityp sowie Optionen oder Parameter enthalten. Einige Kommandos benötigen keinen Anhang.

Kommandodatei: Ausführbare Datei des Types COM. Eine Kommandodatei ist ein Modul in Maschinencode, das bereit ist, an die Speicheradresse 0100H geladen und ausgeführt zu werden. Um eine Kommandodatei auszuführen, genügt es, ihre primäre Bezeichnung als Kommandowort in einer CP/M-Kommandozeile einzugeben.

Kommando-Satzaufbau: Die Vorschrift, welche den richtigen Aufbau einer Kommandozeile festlegt. Die richtige Struktur besteht aus Kommandowort, dem Kommando-Anhang und der ↵ -Taste. Eine Satzaufbauzeile enthält normalerweise Symbole, die mit aktuellen Werten ersetzt werden muß, wenn ein Kommando eingegeben wird.

Kommandowort: Bezeichnung, die ein CP/M-Kommando identifiziert. Normalerweise die primäre Bezeichnung einer Datei des Types COM oder ein residentes Kommando. Das Kommandowort muß vor

dem Kommando-Anhang und der ↵-Taste in einer Kommandozeile erscheinen.

**Konsole:** Primäres Ein-/Ausgabegerät. Die Konsole besteht aus einem Listgerät, wie einem Bildschirm oder Konsoldrucker, sowie einer Tastatur, über die der Anwender mit dem Betriebssystem oder dem Anwendungsprogramm kommuniziert.

**Konsol-Kommando-Bearbeitung:** Siehe CCP.

**Kontrollsummen-Vektor (CSV):** Checksum Vector. Zusammenhängender Datenbereich in BIOS mit einem Byte für jeden zu kontrollierenden Verzeichnis-Sektor, d. h. CKS-Bytes (siehe CKS). Ein Kontrollsummen-Vektor wird für jedes angemeldete Laufwerk eingerichtet und verwaltet. Jeder Zugriff auf das Verzeichnis durch das System bewirkt die Berechnung einer Kontrollsumme, die mit derjenigen im Kontrollsummen-Vektor verglichen wird. Wenn eine Diskrepanz auftritt, wird dem Laufwerk der Status „Read only (nur lesen)“ gegeben. Diese Einrichtung bewahrt den Anwender davor, versehentlich Platten im angemeldeten Laufwerk auszuwechseln. Wird dies gemacht, so würde das System dies nicht erkennen und eventuell Daten einfach überschreiben.

**Lader:** Dienstprogramm, das ein absolutes Programmabbild in den Speicher bringt, so daß es vom CP/M-Betriebssystem ausgeführt werden kann, bzw. Dienstprogramm zur Erstellung eines derartigen Abbildes. LOAD erstellt beispielsweise eine absolute COM-Datei von der Hex-Datei, welche vom Assembler ausgegeben wurde. Diese COM-Datei ist fertig zur Ausführung unter CP/M.

**Linker:** Dienstprogramm zum Einfügen eines verschiebbaren Objektprogrammes in eine absolute Datei zur Ausführung derselben. LINK80 erstellt z. B. eine COM oder PRL-Datei von verschiebbaren RELDateien wie jenen, welche von PL/I-80 erzeugt wurden.

**LIST:** Ein BIOS-Eintrittspunkt zu einer Routine, welche Zeichen an ein Listgerät, normalerweise einen Drucker, sendet.

**List Device:** Siehe Listgerät.

**Listgerät:** Gerät wie z. B. ein Drucker, an welchem Daten gelistet oder gedruckt werden können.

**LISTST:** BIOS-Eintrittspunkt zu einer Routine, welche den Status (Ready Status) des Listgerätes ermittelt.

**Loader:** Siehe Lader.

**Logged In:** Siehe aktiviert.

**Logisch:** Darstellung von etwas, das sowohl gleich als auch unterschiedlich in seiner physischen Form sein kann. Z. B. kann eine Festplatte ein physisches Laufwerk sein und doch können Sie den Speicherplatz so unterteilen, als ob es sich um mehrere verschiedene Laufwerke handelte. Diese scheinbaren Laufwerke sind die logischen Laufwerke.

**Logischer Sektor:** Siehe Sektor.

**LSC:** Parameter in der „Diskdef Macro Library“, der die Nummer des letzten physischen Sektors angibt.

**LST:** Logisches Listgerät in CP/M (gewöhnlich ein Drucker). Das Listgerät in CP/M ist ein Gerät ausschließlich zur Ausgabe, das durch die BIOS-Eintrittspunkte LIST und LISTST angesprochen wird. Das Kommando STAT ermöglicht die Zuordnung von LST: zu einem der physischen Geräte: TTY:, CRT:, LPT: oder UL1:, vorausgesetzt, diese Geräte und das IOBYTE sind in den LIST und LISTST-Eintrittspunkten Ihres CP/M-BIOS-Moduls enthalten.

**Makro-Assembler:** Ein Übersetzer von Assemblercode, der Makroverarbeitungs-einrichtungen hat. Makrodefinitionen ermöglichen die Speicherung von Instruktionsgruppen und das Einsetzen derselben in das Quellprogramm, wenn immer die Makronamen auftauchen. Definitionen und Aufrufe können geschachtelt werden. Makroparameter können so gestaltet werden, daß sie beliebige Textgruppen an einen bestimmten Makrobefehl zum Einsetzen während der Erweiterung übergeben.

**Megabyte:** Über eine Million Bytes, 1024 Kilobytes. Siehe Byte und Kilobyte.

**Mehrdeutige Dateibezeichnung:** (Ambignons filename = afn). Eine Dateibezeichnung die eines der beiden Dateigruppensymbole „\*“ oder „?“ entweder in der primären Bezeichnung oder im Dateityp oder in beiden enthält. Wenn Sie Zeichen der Dateibezeichnung durch diese Dateigruppensymbole ersetzen, erzeugen Sie eine mehrdeutige Dateibezeichnung und können mit einer einzigen Kommandozeile mehrere CP/M-Dateien ansprechen.

**Mikroprozessor:** Silizium-Baustein, der die Zentralrecheneinheit (CPU) eines Mikrocomputers bildet. Der Intel 8080 und der Zilog Z-80 sind die gebräuchlichsten Mikroprozessoren in CP/M-Systemen.

**MP/M:** Multi-Programming Monitor control program. Ein Mikrocomputer-Betriebssystem, das mehrere Arbeitsplätze mit Multi-Programming an jedem Arbeitsplatz unterstützt.

**Multi-Programming:** Die Fähigkeit, mehr als ein Programm aufzurufen und gleichzeitig auszuführen. Diese Programme, auch Prozesse genannt, arbeiten im time-sharing Verfahren. Jeder Prozeß bekommt einen Anteil Prozessorzeit in einem Rundum-Verfahren. Siehe auch Parallele Operationen.

**Nibble:** Halbyte. Die Low-order oder High-order 4 Bits eines Bytes.

**OFF:** 2-Byte-Parameter im Disk Parameter Block auf DPB+13. Dieser Wert spezifiziert die Anzahl der reservierten Systemspuren. Das Plattenverzeichnis beginnt im ersten Sektor der Spur OFF.

**OFS:** Parameter der „Diskdef Macro Library“, der die Anzahl der reservierten Systemspuren angibt. Siehe OFF.

**Operating System:** Siehe Betriebssystem.

**Option:** Einer von mehreren Parametern, die Teil eines Kommando-Anhanges sein können. Optionen werden dazu benutzt, zusätzliche Bedingungen für die Ausführung eines Kommandos anzugeben.

**Output:** Siehe Ausgabe.

**Page:** Siehe Seite.

**Page Relocatable Program:** Siehe PRL.

**Page Zero:** Siehe Seite Null.

**Parallele Operationen:** Simultane Ausführung zweier Prozesse oder Operationen.

**Parameter:** Wert im Kommando-Anhang, der zusätzliche Informationen für das Kommando angibt. Technisch ist ein Parameter ein notwendiges Element eines Kommandos.

**Peripheriegeräte:** Geräte außerhalb der CPU. Z. B. Terminals, Drucker und Plattenlaufwerke sind gängige Peripheriegeräte, die nicht Teil des Prozessors sind, aber in Zusammenhang mit ihm benutzt werden.

**Physisch:** Charakteristika von Computerelementen, generell Hardware, die wirklich vorhanden sind. In Programmen werden physische Komponenten durch logische Komponenten ersetzt.

**Plattenlaufwerk:** Peripheriegerät, das Informationen auf bzw. von Platten oder Disketten schreibt und liest. CP/M vergibt für jedes Laufwerk einen Buchstaben. In einem System mit 4 Laufwerken werden diese z. B. mit A, B, C und D bezeichnet.

**Primäre Dateibezeichnung:** Die ersten 8 Zeichen einer Dateibezeichnung. Die primäre Dateibezeichnung ist ein eindeutiger Name, der dem Benutzer hilft, den Dateiinhalt zu identifizieren. Die primäre Dateibezeichnung enthält 1 - 8 Zeichen und kann jeden Buchstaben oder jede Ziffer und außerdem einige Spezialzeichen enthalten. Die primäre Dateibezeichnung folgt der Laufwerksspezifikation und wird vor dem Dateityp angegeben.

**PRL:** Page Relocatable Program = Programm verschiebbar in den Seiten. Ein solches Programm wird auf der Platte mit dem Dateityp PPL gespeichert. PRL-Programme können leicht an jede Seitengrenze verschoben werden und sind somit zur Ausführung in einem MP/M-System ohne Banking brauchbar.

**Programm:** Serie von kodierten Instruktionen, die spezielle Aufgaben erledigen, wenn sie von einem Computer ausgeführt werden. Ein Programm kann entweder in einer für den Processor spezifischen Sprache oder einer höheren Programmiersprache geschrieben sein. Im letzteren Falle kann das Programm sehr einfach auf einer Reihe verschiedenster Prozessoren eingesetzt werden.

**Prompt:** Siehe Bedienungsführung und CP/M-Bereitschaftszeichen.

**Puffer:** Speicherbereich, in dem während einer Informationsübertragung vorübergehend Daten gespeichert werden.

**PUN:** Logisches CP/M-Gerät (Punch = Stanzer). PUN ist ein Ausgabegerät, das durch den BIOS-Eintrittspunkt PUNCH angesprochen wird.

**PUNCH:** BIOS-Eintrittspunkt zu einer Routine, welche ein Zeichen an das Punch-Ausgabegerät sendet.

**Quelldatei:** Eine Textdatei im ASCII-Code. Sie wird üblicherweise mit einem Editor erstellt und ist eine Eingabedatei für ein Systemprogramm wie einen Sprachübersetzer oder Textformatierer.

**RDR:** Logisches CP/M-Gerät (Reader = Leser). RDR ist ein Eingabegerät, das durch den BIOS-Eintrittspunkt **READER** angesprochen wird.

**READ:** BIOS-Eintrittspunkt zu einer Routine, welche 128 Bytes vom momentan selektierten Laufwerk, der selektierten Spur und dem selektierten Sektor in die laufende DMA-Adresse liest.

**READER:** Eintrittspunkt einer BIOS-Routine, welche das nächste Zeichen vom gerade zugeordneten Reader-Gerät einliest.

**Read-only (RO):** Siehe Schreibschutz.

**Read-Write (RW):** Siehe Schreiberlaubnis.

**Record:** Siehe Satz.

**Reentrant Procedure:** Siehe Jederzeit verwendbare Prozedur.

**Rekursive Prozeduren:** Code, der sich während der Ausführung selbst aufrufen kann.

**Residente Kommandos:** Kommandos, die ständig im Speicher sind. Sie werden besonders schnell ausgeführt, da sie nicht von der Platte geladen werden müssen.

**Restart (RST):** Eine 1-Byte-Aufrufinstruktion, die normalerweise während Interrupt-Folgen und für Debug-Unterbrechungen benutzt wird. Es gibt 8 Restartpunkte, RST 0 bis RST 7, deren Adressen sich aus dem Produkt aus  $8 * \text{Restartnummer}$  ergeben.

**RO:** Read-only. Siehe Schreibschutz.

**ROM:** Read-only Memory. Dieser Speicher kann nur ausgelesen, jedoch nicht beschrieben werden und ist nur für Instruktionen und vorgegebene Festdaten verwendbar.

**RST:** Siehe Restart.

**Rücksprungbereich:** Stack. Reservierter Speicherbereich, in dem der Prozessor die Rücksprungadresse speichert, wenn eine Call-Instruktion auftritt. Bei einer Return-Instruktion gibt der Prozessor die laufende Rücksprungadresse an den Programmzähler zurück. In diesem Bereich können auch Daten wie z. B. die Registerinhalte abgespeichert werden. Die Instruktion Push übergibt die Daten an den Bereich und die Instruktion Pop holt sie wieder ab. Ein Wert wird in den Bereich gegeben, indem der Stack Pointer (SP) um 2 reduziert wird und der Wert dann in die SP-Adresse geschrieben wird. Der Stack wächst also nach unten im Speicher.

**RW:** Read-Write. Siehe Schreiberlaubnis.

**Satz:** Gruppe von Bytes in einer Datei. Ein physischer Satz besteht aus 128 Bytes und ist die Grundeinheit für die Datenübertragung zwischen dem Betriebssystem und dem Anwendungsprogramm. Ein logischer Satz kann in der Länge variieren. Er bezeichnet eine Informationseinheit. Zwei „Mitarbeiter-Sätze“ zu je 64 Bytes können in einem physischen Satz gespeichert werden. Sätze werden gruppiert und ergeben eine Datei.

**Schnittstelle:** Einrichtung, die es zwei unabhängigen Systemen ermöglicht, miteinander zu kommunizieren. Z. B. die Schnittstelle zwischen Hardware und Software in einem Mikrocomputer.

**Schreiberlaubnis:** Attribut, das einer Datei oder einem Laufwerk gegeben werden kann. Dieses Attribut erlaubt das Lesen und Schreiben auf eine bestimmte Datei oder auf jede Datei eines Laufwerks mit Schreiberlaubnis. Jede Datei und jedes Laufwerk hat entweder Schreiberlaubnis oder ist schreibgeschützt.

**Schreibschutz (RO):** Attribut, das einer Datei oder einem Laufwerk gegeben werden kann. Ist es einer Datei zugeordnet, es kann zwar von der Datei gelesen, jedoch nicht in sie geschrieben werden. Ist RO für ein Laufwerk spezifiziert, so kann jede Datei auf der Platte gelesen werden, jedoch kann weder eine neue Datei zugefügt werden, noch eine Datei gelöscht, geändert oder umbenannt werden, noch kann in irgendeine Datei geschrieben werden. Das Attribut kann vom STAT-Kommando für Datei und Laufwerk gesetzt werden. Jede Datei und jedes Laufwerk ist entweder schreibgeschützt oder nicht. Das automatisch zugeteilte Attribut für Laufwerk und Dateien ist Schreiberlaubnis. Jeder Fehler beim Rückstellen der Platte oder beim Platten-

austausch wird das Laufwerk automatisch mit einem Schreibschutz versehen, der bis zur Behebung des Fehlers in Kraft bleibt..

**Sectors per Track (SPT):** Siehe Sektoren pro Spur.

**SECTTRAN:** Eintrittspunkt in eine BIOS-Routine, welche die Umsetzung von logischen in physische Sektoren für BDOS durchführt.

**Seite:** 256 zusammenhängende Bytes im Speicher, die an einer Seitengrenze beginnen. Seitengrenzen haben Basisadressen, die ein Vielfaches von 256 (0100H) Bytes sind. In Hex-Aufzeichnung beginnen Seiten immer an einer Adresse deren niedrigstwertiges Byte den Wert Null hat.

**Seite Null:** Speicherbereich von 0000H bis 0100H, der dazu dient, wichtige Systemparameter zu speichern. Seite Null dient primär als Schnittstellenbereich zwischen Anwendungsprogrammen und dem CP/M-Modul BDOS. Zu beachten: In nicht-standard Systemen ist dieser Bereich die Basis-Seite des Systems und bezeichnet die ersten 256 Bytes des Speichers, die vom CP/M-System und den darunter laufenden Programmen benutzt werden.

**Sektor:** Im CP/M-System besteht ein Sektor immer aus 128 zusammenhängenden Bytes. Ein Sektor ist die Grundeinheit für das Lesen und Schreiben von Daten auf Platte durch das BIOS. Ein Sektor kann ein Satz in einer Datei oder ein Sektor im Verzeichnis sein. BDOS benötigt eine logische Sektornummer zwischen 0 und (SPT-1). Typischerweise wird diese Nummer in einen physischen Sektor durch die BIOS-Routine mit dem Eintrittspunkt SECTTRAN umgerechnet. Bei einigen Plattengeräten ist die physische Sektorgröße mehr als 128 Bytes, üblicherweise eine Zweierpotenz wie 256, 512, 1024 oder 2048 Bytes. Diese Sektoren werden in der CP/M-Dokumentation Host-Sektoren genannt und sollten nicht mit anderen Sektoren verwechselt werden. Sektor bezieht sich auf CP/M-Sektoren von 128 Bytes. Ist der Host-Sektor größer als 128 Bytes, so müssen diese Sektoren im Speicher gepuffert werden und die CP/M-Sektoren müssen aus ihnen durch Blocken und Entblocken geformt werden. Dies wird durch das Zusatzmodul „Blockungs- und Entblockungsalgorithmus“ zwischen den Ein-/Ausgaberoutinen für die Platte in BIOS und der wirklichen Platten-Ein-/Ausgabe durchgeführt.

**Sektoren pro Spur (SPT):** Ein 2-Byte-Parameter im Disk Parameter Block bei DPB+0. Das BDOS ruft den BIOS-Eintrittspunkt SECTTRAN mit logischen Sektornummern zwischen 0 und (SPT-1) im Register BC an.

**SELDSK:** Eintrittspunkt in eine BIOS-Routine, welche das momentan selektierte Laufwerk einträgt.

**SETDMA:** Eintrittspunkt einer BIOS-Routine, welche die momentan selektierte DMA-Adresse festhält. Die DMA-Adresse ist die Adresse eines 128-Byte-Puffers im Speicher, der zur Datenübertragung von und zu der Platte dient.

**SETSEC:** Eintrittspunkt für eine BIOS-Routine, die den zur Zeit selektierten Sektor einträgt.

**SETTRK:** Eintrittspunkt für eine BIOS-Routine, die die momentan selektierte Spur festhält.

**Skew Factor:** Siehe Versatzfaktor.

**SKF:** Ein Parameter der „Diskdef Macro Library“, welcher den Versatzfaktor (skew factor), der zum Aufbau von XLT verwendet wird, spezifiziert. Wenn SKF Null ist, wird keine Umsetzungstabelle generiert und das XLT-Byte im DPH ist 0000H.

**Software:** Programme, welche maschinenlesbare Instruktionen enthalten. Im Gegensatz dazu Hardware, die physischen Komponenten eines Computers.

**Source File:** Siehe Quelldatei.

**SP:** Stack pointer. Siehe Rücksprungbereich.

**Spooling:** Vorgang bei dem die an einen Drucker auszugebenden Daten in einer Datei gesammelt werden, während der Drucker anderweitig belegt ist. Die Datei wird gedruckt, sobald der Drucker frei ist. Ein Programm braucht nicht den langsamen Druckprozeß abwarten.

**SPT:** Siehe Sektoren pro Spur.

**Stack:** Siehe Rücksprungbereich.

**Stapel-Zeiger:** Siehe Rücksprungbereich.

**Start: (BOOT)** Das Laden des Betriebssystems in den Speicher. Ein Startprogramm (Boot program) ist ein kleines Stückchen Software, das automatisch ausgeführt wird, wenn das System eingeschaltet wird oder in Grundstellung (Reset) gebracht wird. Das Startprogramm lädt

den Rest des Betriebssystems in den Speicher. Dieser Vorgang wird auch Kaltstart (cold boot) genannt. Typischerweise ist das Startprogramm im ersten Sektor der Systemspur auf der Systemplatte.

**Syntax:** Satzaufbau für die Eingabe von Kommandos.

**SYS:** Siehe Systemattribut.

**SYSGEN Image:** Speicherabbild des CP/M-Systems das durch SYSGEN generiert wird, wenn kein Ziellaufwerk angegeben ist. Boot beginnt bei 900H, BIOS bei B00H und das CP/M-System bei 2300H.

**Systemattribut (SYS):** Dateiattribut. Kann einer Datei durch die Option SYS im STAT-Kommando oder durch die Dateiattribut-Funktion (BDOS-Funktion 12) gegeben werden. Eine Datei mit dem Systemattribut wird nicht durch ein DIR-Kommando angezeigt.

**Systembereitschaftsanzeige, Systembereitschaftszeichen:** Symbol, das vom Betriebssystem anzeigt, daß das System bereit ist für eine Eingabe. Siehe Bedienerführung und CP/M-Bereitschaftszeichen.

**System Prompt:** Siehe Systembereitschaftsanzeige.

**Systemspuren:** Spuren auf der Platte, die für das CP/M-System reserviert sind. Die Anzahl der Systemspuren wird durch den Parameter OFF im Disk Parameter Block (DPB) definiert. Die Systemspuren eines Laufwerks kommen immer vor den Datenspuren. Das Kommando SYSGEN kopiert das CP/M-System von den Systemspuren in den Speicher und umgekehrt.

**Terminal:** Siehe Konsole.

**TPA:** Transient Program Area. Flüchtiger Programmbereich. Speicherbereich in dem Anwenderprogramme ablaufen und Daten speichern. Dieser Bereich beginnt bei 0100H und geht bis zur Basis des CP/M-Systems im oberen Speicherbereich. Das erste Modul vom CP/M-System ist der CCP, der durch ein Anwenderprogramm überschrieben werden kann. In diesem Fall geht TPA bis zur Basisadresse vom BDOS-Modul. Wird CCP überschrieben, so muß das Programm entweder mit einem System-Reset (Funktion 0)-Aufruf oder einem Sprung auf Position Null in Seite Null (Page Zero) beendet werden. Die Basisadresse vom CP/M-BDOS ist in Adresse 0006H in Seite Null mit dem niedrigstwertigen Byte voran abgespeichert.

**TPI** Spuren pro Zoll (auf Disketten)

**Übersetzungstabelle** von logischen auf physische Sektoren: Siehe XLT.

**USER:** Bezeichnung zur Unterscheidung verschiedener Regionen im Verzeichnis bei CP/M und MP/M.

**User Number:** Siehe Benutzernummer.

**Utility:** Siehe Dienstprogramm.

**Vektor:** Speicherposition. Ein Eintrittspunkt in das Betriebssystem für Systemaufrufe oder Interrupt-Abhandlung.

**Verketten:** Bezeichnung des PIP-Vorganges, der zwei oder mehrere separate Dateien in eine neue Datei in der angegebenen Reihenfolge kopiert.

**Versatzfaktor:** Faktor, der die Umsetzung der logischen in physische Sektoren in XLT definiert. Logische Sektornummern werden vom BDOS benutzt und sind zwischen 0 und (SPT-1). Daten werden in fortlaufende 128-Byte-Sektoren geschrieben und in Datenblocks gruppiert. Die Anzahl der Sektoren pro Block ergibt sich aus BLS/128. Physische Sektoren auf der Platte werden ebenfalls fortlaufend nummeriert. Ist die Größe des physischen Sektors auch 128 Bytes, so ergibt sich ein Eins-zu-Eins-Verhältnis zwischen den logischen und den physischen Sektoren. Die Umsetzungstabelle XLT hält das Verhältnis von logischen zu physischen Sektoren fest. Der Versatzfaktor wird typischerweise zur Erstellung der Tabellenwerte verwendet. Ist der Versatzfaktor 6, so ist XLT:

Logisch:	0	1	2	3	4	5	6	...	25
Physisch:	1	7	13	19	25	5	11	...	22

Der Versatzfaktor ergibt zusätzliche Zeit für die Programmausführung, ohne daß der nächste Sektor übersprungen wird. Andernfalls muß das System jeweils eine ganze Plattenumdrehung abwarten, bevor der nächste Sektor gelesen wird. Der Versatzfaktor kann, abhängig von der Hardware-Geschwindigkeit und dem Programm, variiert werden. Beachten: Die Sektoren-Umsetzung wird nicht durchgeführt, wenn die physischen Sektoren größer als 128 Bytes sind, da in diesem Falle die Sektorenblockung/-entblockung stattfindet (siehe auch Sektor, SKF und XLT).

**Verzeichnis:** Teil einer Platte, der Einträge für alle Dateien auf der Platte enthält. Als Antwort auf das DIR-Kommando zeigt CP/M alle Dateibezeichnungen, die im Verzeichnis gespeichert sind, an. Das

Verzeichnis enthält auch die Positionen der jeder Datei zugewiesenen Blöcke. Jedes Verzeichniselement umfaßt einen 32 Byte-FCB. Eine Datei kann, abhängig von ihrer Größe, mehrere Verzeichniselemente benötigen. Die maximale Anzahl von Verzeichniselementen wird durch den Wert im Plattenparameterblock für DRM für das Laufwerk festgehalten.

**Verzeichniseintrag:** Dateieintrag, der durch das DIR-Kommando angezeigt wird. Manchmal wird mit Verzeichniseintrag auch das physische Verzeichniselement bezeichnet.

**Verzeichniselement:** Datenstruktur. Jede Plattendatei bekommt eines oder mehrere Verzeichniselemente zu je 32 Bytes zugeordnet. Jeder Verzeichnissektor enthält 4 Verzeichniselemente. Verzeichniselemente werden auch FCB (Dateikontrollblock) genannt.

**Vorgabe:** Augenblicklich selektiertes Plattenlaufwerk und Benutzer-Nummer (user number). Jedes Kommando, das kein Laufwerk oder keine Benutzer-Nummer spezifiziert, bezieht sich auf das vorgegebene Laufwerk und die vorgegebene Benutzer-Nummer. Nach dem ersten Laden von CP/M, ist das Laufwerk A und die Benutzer-Nummer 0 vorgegeben.

**Vorgabe-FCB:** Zwei FCB's werden von CCP auf den Adressen 005CH und 006CH in Seite Null (page zero) bereitgestellt. Der erste FCB wird vom ersten separaten Feld, der zweite FCB vom nächsten Feld im Kommando-Anhang initialisiert.

**Vorgabepuffer:** Vorgegebener Pufferbereich von 128 Bytes, der in Seite Null (Page zero) ab Adresse 0080H bereitgestellt wird. Wenn CCP eine Kommandodatei lädt, wird dieser Puffer für den Kommando-Anhang bereitgestellt; d. h. alle Zeichen, die nach der Bezeichnung der Kommandodatei eingegeben werden, werden in diesem Puffer gespeichert. Das erste Byte auf 0080H gibt die Länge des Kommando-Anhangs an, während der Kommando-Anhang selbst auf Position 0081H beginnt. Der Kommando-Anhang wird durch ein Byte mit dem Binärwert Null abgeschlossen. Das Kommando I in DDT bzw. SID initialisiert diesen Puffer genauso wie CCP.

**Warmstart:** Programmbeendigung durch: einen Sprung auf dem Warmstart-Vektor an Position 0000H, einen System-Reset (BDOS-Funktion 0) oder eine Eingabe von ctl-C an der Tastatur. Ein Warmstart reinitialisiert das Plattenuntersystem und gibt die Steuerung ans

Betriebssystem im CCP-Bereich zurück. Der Warmstart-Vektor ist einfach ein Sprung auf den WBOOT-Eintrittspunkt im BIOS.

WBOOT: Eintrittspunkt für eine BIOS-Routine, die für einen Warmstart benutzt wird. Ein Warmstart wird durchgeführt, wenn ein Anwendungsprogramm auf die Position 0000H verzweigt, wenn ein System-Reset erfolgt oder wenn ctl-C eingegeben wird. CCP und BDOS werden von den Systemspuren des Laufwerks A geladen.

Wildcard Characters: Siehe Dateigruppensymbole.

Winchester Disk: Siehe Festplatte.

Wort: 16-Bit bzw. 2-Byte-Wert wie z. B. eine Adresse. Obwohl der Intel 8080 ein 8-Bit-Prozessor ist, belegen Adressen 2 Bytes und werden Wortwerte genannt.

WRITE: Eintrittspunkt einer BIOS-Routine, welche den Satz an der selektierten DMA-Adresse auf das selektierte Laufwerk, die selektierte Spur und den selektierten Sektor schreibt.

XLT: Umsetzungstabelle von logischen auf physische Sektoren, die Teil vom BIOS ist. SECTTRAN benutzt XLT um logische Sektornummern in physische umzuwandeln. XLT bezieht sich auch auf die 2-Byte-Adresse im Disk Parameter Header bei DPBASE+0. Ist dieser Parameter 0, so wird keine Umsetzung vorgenommen. Andernfalls gibt dieser Parameter die Adresse der Umsetzungstabelle an.

Zero Page: Siehe Seite Null.

✓

✓

✓



# Lizenzprogramme für den Personalcomputer NCR - DM V

Bitte füllen Sie diese Karte aus und senden Sie an umstehende Adresse.

Ich bestätige hiermit, das (die) nachfolgende(n) aufgezählte(n) Programm(e) erhalten zu haben:

DRI PRODUCT : CP/M(R)-80  
NCR VERSION NO.: 2.2 (03.00)  
SERIAL NO. : CP2-761-51801

Für die Benutzung der vorstehenden Programme gelten die Allgemeinen Lizenzbedingungen der NCR GmbH, die ich erhalten habe und mir bekannt sind.

Name: \_\_\_\_\_

Vorname: \_\_\_\_\_

Anschrift: \_\_\_\_\_

Datum: \_\_\_\_\_

Unterschrift

Absender

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

(Postleitzahl) (Ort)

\_\_\_\_\_

Gebühr  
bezahlt  
Empfänger

Antwort-  
Postkarte

An die  
NCR GmbH  
Ulmer Straße 160 b  
8900 Augsburg