

**jk82 CP/M PLUS - Handbuch**

Ergänzungen zu den Digital Research Manuals

U P D A T E 2    zum jk82 CP/M PLUS

Der Update 2 zum jk82 CP/M PLUS hat auch einige Änderungen im Implementationshandbuch zur Folge!

Wir bitten Sie daher, die folgenden Seiten in Ihrem Handbuch auszutauschen bzw. zu ergänzen.

Alle Seiten des Handbuches, die im Zusammenhang mit dem Update 2 verändert wurden, sind in der Fußzeile mit dem Update-Level (UL: 02) gekennzeichnet worden.

Für das neue Programm 'DFE' existiert zur Zeit noch keine Beschreibung. Wir meinen jedoch, daß es wegen seiner Menüsteuerung trotzdem benutzbar ist.

**jk82 CP/M PLUS - Handbuch**

Ergänzungen zu den Digital Research Manuals

**U P D A T E 3    zum jk82 CP/M PLUS**

Der Update 3 zum jk82 CP/M PLUS hat auch einige Änderungen im Implementationshandbuch zur Folge!

Wir bitten Sie daher, die folgenden Seiten in Ihrem Handbuch auszutauschen bzw. zu ergänzen.

Alle Seiten des Handbuches, die im Zusammenhang mit dem Update 3 verändert wurden, sind in der Fußzeile mit dem Update-Level (UL: 03) gekennzeichnet worden.

Bevor Sie den Update 3 in Ihrem Handbuch vornehmen, sollten Sie erst den Update 2 in Ihrem Handbuch erledigt haben!

**Achtung: Bug Bug Bug Bug ....**

Bei Benutzung eines **HDC 5**-Controllers entsteht beim Partitionieren einer Winchester folgendes Problem:

Die 2., 3. und 4. Partition kann nur als DOS-Winchester benutzt werden, da die Winchester beim Formatieren nur am Anfang mit dem Code 0E5H beschrieben wird.

Abhilfe: Winchester nur für DOS PLUS partitionieren oder mit DM nach dem Formatieren 'E5-Sektoren' von Spur 0 und 1 zu den jeweiligen Startspuren der einzelnen Partitionen kopieren!

jjjjjjjjjjjj	kkkkkkkkkkkk	jk82 CP/M PLUS
jjjjjjjjjjjj	kkkkkkkkkkkk	jk82 CP/M PLUS
jjjjjjjjjjjj	kkkkkkkkkkkk	jk82 CP/M PLUS
jjjjjjjjjjjj	kkkkkkkkkkkk	jk82 CP/M PLUS
jjjjjjjjjjjj	kkkkkkkkkkkk	jk82 CP/M PLUS
jjjjjjjjjjjj	kkkkkkkkkkkk	jk82 CP/M PLUS
jjjjjjjjjjjj	kkkkkkkkkkkk	jk82 CP/M PLUS
jjjjjjjjjjjj	kkkkkkkkkkkk	jk82 CP/M PLUS
jjjjjjjjjjjj	kkkkkkkkkkkk	jk82 CP/M PLUS
jjjjjjjjjjjj	kkkkkkkkkkkk	jk82 CP/M PLUS
jjjjjjjjjjjj	kkkkkkkkkkkk	jk82 CP/M PLUS
jjjjjjjjjjjj	kkkkkkkkkkkk	jk82 CP/M PLUS
jjjjjjjjjjjj	kkkkkkkkkkkk	jk82 CP/M PLUS
jjjjjjjjjjjj	kkkkkkkkkkkk	jk82 CP/M PLUS
jjjjjjjjjjjj	kkkkkkkkkkkk	jk82 CP/M PLUS

jk82 CP/M PLUS-Handbuch

Ergänzungen zu den Digital Research Manuals

COPYRIGHT <C> 1984/85 by JANICH & KLASS Computersysteme und MICHELS & KLEBERHOFF Computer GmbH in Wuppertal.

Autoren: Dipl. Ing. Dietmar Janich und Dipl. Ing. Lutz Kleberhoff

Dieses Handbuch ist urheberrechtlich geschützt. Kein Teil dieses Handbuches darf ohne die Zustimmung der Autoren in irgendeiner Form verbreitet werden.

1.1 Vorwort

1.2 End-Benutzer Lizenz-Vereinbarung

1.3 Softwarepflege durch DRI/J&K/MKC

1.4 Mögliche Hardware-Zusammenstellungen

- Jumperpläne für Platinen
- Reservierte I/O-Portbelegung für CBIOS 1.6
- Speicherbelegung (Beispiel)
- Bemerkungen zu Disk-Laufwerken

1.5 Hinweise zur Inbetriebnahme des Betriebssystems  
(unbedingt vorher lesen!)

1.6 ASCII-Tabelle (deutsche Version)

- 2.1 CPMINST (Betriebssystem generieren)
- 2.2 Disk Manager
  - Übersicht
  - Kurzbeschreibung
  - Einführung
  - Disk-Monitor
  - Analyse von Disketten
  - Formate auswählen
  - Formate editieren
  - Formate ausdrucken
- 2.3 FORMAT (Formatierprogramm für alle Laufwerke)
- 2.4 DISKCOPY (schnelles 1:1 Kopierprogramm)
- 2.5 CONFIG (Schnittstellen konfigurieren)
- 2.6 TERM (Terminal-Programm zur Kommunikation mit fremden Rechnern)
- 2.7 SIMU (EPROM-Simulator 1 laden)
- 2.8 EprCopy (Erstellen einer EPROM-DISK)
- 2.9 Das Kopierprogramm MSN
- 2.10 DFE (Bearbeiten der Datei DISKFORM.DAT)

3.1 ZEIT.RSX (Datum/Zeit auf Programmlistings drucken)

3.2 DATUM.REL (Datum/Zeit in druckaufbereiteter Form aus dem System abrufen)



#### 4.1 Urlader LDR 1.600

#### 4.2 BIOSKERN

#### 4.3 Das BOOT-Modul

#### 4.4 Disk Routinen

- Übersicht
- DISKS.ASM
- DISKDEF.LIB
- FUTIL-Modul
- FDC2-Modul
- FDC85-Modul
- FDC5-Modul
- DDATA-Modul
- XEBEC-Modul
- HDC5-Modul
- EPC2FDC-Modul
- EPC1FDC-Modul
- Parameter-Tabellen

#### 4.5 Character I/O

- CHARIO.ASM
- MODEBAUD.LIB
- SYSDEF.LIB
- PORTS.LIB
- USER.LIB
- EPC7220.REL

#### 4.6 Move-Module

- RAE3MOVE: Move-Modul für RAM/Adreßerweiterungskarte bis Rev. 3
- RAE4MOVE: Move-Modul für RAM/Adreßerweiterungskarte ab Rev. 4
- CPU2MOVE: Move-Modul für CPU II
- CPU3MOVE: Move-Modul für CPU III mit DMA
- CP3NMOVE: Move-Modul für CPU III ohne DMA
- EPC1MOVE: Move-Modul für EPC 1.6
- EPC2MOVE: Move-Modul für EPC II

#### 4.7 Datum/Zeit-Module

- CPU3TIME
- FDC2TIME
- DATECOMP (Datumsumrechnung für CPU3TIME/FDC2TIME)
- TIME (Interruptuhr)

#### 4.8 CBIOS-Erweiterungen

- USER-Funktion
- Datei USER.ASM
- Datei PRINTER.ASM
- TRACE-Modul

## 1.1 Vorwort

Diese Implementation des CP/M PLUS Betriebssystems von Digital Research (DRI) weist einige Erweiterungen und Änderungen gegenüber den im Handbuch von DRI beschriebenen Eigenschaften auf. Dieses Handbuch soll diese Änderungen und Erweiterungen beschreiben.

Das Konzept des BIOS zu CP/M PLUS ist modular. Das vereinfacht eine Implementation zwar sehr, jedoch war es von DRI lediglich geplant, für jede konkrete Hardware ein CBIOS zu schreiben. Dies ist jedoch für unser modulares Platinen-System sehr ungünstig, da die Hardware nur bedingt konkretisiert werden kann (6 verschiedene CPU-Karten, 4 verschiedene Floppy-Controller usw.). Wir haben es uns deshalb mit der Implementation dieses Betriebssystems nicht leicht gemacht, damit die Anpassung für Sie einfacher wird. Das CBIOS dieses Betriebssystems ist für eine modulare Hardware ausgelegt und leicht an verschiedene Kombinationen anpaßbar. In den meisten Fällen reicht eine Anpassung mit Hilfe des Dienstprogrammes 'CPMINST'.

### **Änderungen gegenüber dem Konzept von DRI:**

- CP/M PLUS steht nur als gebankte Version mit mindestens 128KBytes RAM zur Verfügung. Eine nicht gebankte (d.h. weniger leistungsfähigere) Version wird nicht angeboten.
- Der Loader befindet sich vollständig im EPROM. Systemspuren existieren deshalb nicht. Lediglich bei Floppy-Laufwerken wird eine Spur reserviert um einen Disketten-Beschreibungsblock (beschreibt das jeweilige Diskettenformat) unterzubringen. Das Programm **COPYSYS** existiert deshalb in dieser Implementation nicht.

### **Erweiterungen und Besonderheiten dieser Implementation:**

- 'HELP' steht hier in **deutscher Sprache** zur Verfügung und ist um einige Erklärungen zu dieser Betriebssystem-Implementation erweitert worden.
- Wahlweise kann eine CPU 1, CPU 2, CPU 3, EPC 1.6, EPC 2 oder alte HKM-CPU mit dieser System-Implementation benutzt werden.
- Bei der CPU 1 bzw. HKM-CPU ist eine Adreßerweiterungsplatine Rev. 2/3 oder 4/5 möglich (und notwendig).
- FDC 5 (oder alternativ DPS-IO), FDC 8/5 und FDC 2 sind **gleichzeitig** möglich. Die FDC 8/5-Platine muß allerdings bei 8"-Betrieb auf die Adresse 04H gelegt werden. Bei 5,25"-Betrieb wird eine FDC 8/5-Platine wie eine FDC 5-Platine behandelt.

- VIDEO 1, SIO, P/SIO (oder alternativ DPS-IO) und die Centronics-Schnittstelle der GPIB-Platine werden zusätzlich zu den I/O-Ports der CPU-Platine wahlweise unterstützt.
- Es können zur Zeit bis zu 21 (!) Drives unterstützt werden:  
3 x 3,5/5,25" an der FDC 5 (bzw. DPS-IO oder FDC 8/5 auf Adresse 0), 3 x 8" an der FDC 8/5 (auf Adresse 4), 3 x 3,5/5,25" an der FDC 2, 4 x 8" an der FDC 2, 2 x Winchester an der FDC 2 (mit zusätzlichem XEBEC-Controller), 2 x Winchester am HDC 5, eine EPROM-DISK, eine CMOS-RAMDISK, eine SMB-RAMDISK (bis zu 3 SMB-Platinen mit je 1MBYTE RAM) und eine dyn. RAMDISK (und auf Wunsch alles gleichzeitig bis zu maximal 16 Drives!!). Dabei ist es auch möglich, verschiedene Floppy-Laufwerke zu kombinieren (z.B. 40 Track und 80 Track bei 5,25"). Bei Benutzung des EPC 1.6 oder des EPC 2 sind einige Einschränkungen zu den obigen Kombinationen zu beachten.
- Bei 5,25" 80 Track-Laufwerken ist ein beliebiger Wechsel zwischen 80 Track-Disketten und 40 Track-Disketten möglich (ohne daß ein besonderes Kommando notwendig wäre).
- Mit Hilfe des Dienstprogrammes 'DM' ist es außerdem möglich, zur Zeit ca. 55 verschiedene Fremdformate zu lesen und zu schreiben (wird ständig erweitert). Unbekannte Diskettenformate können mit DM analysiert werden. Mit dem Diskmonitor des DM ist es auch möglich, Disketten (und Winchester) direkt zu editieren.
- Einbinden von eigenem Character-I/O ist über vordefinierte MACROS (in USER.LIB) leicht möglich.
- Die Zuordnung verschiedener physikalischer Geräte zu den einzelnen logischen Geräten ist jederzeit vom Anwender durchführbar.
- Der Character-I/O ist erweitert um die Funktionen TIMEOUT, Einstellen von Hardware-Handshake, Datenwort-Format und Parity. Zum Einstellen dieser Funktionen existiert ein Dienstprogramm.
- Installieren des Betriebssystems erfolgt mit einem Install-Programm. Dabei werden alle systemrelevanten Informationen interaktiv abgefragt.
- CP/M PLUS läuft unter 4MHz; 4,9152MHz und 6MHz (ohne irgendwelche Anpassungen!).
- Zum schnellen Diskettenkopieren steht ein Dienstprogramm 'DISKCOPY' zur Verfügung.
- Mit Hilfe des Dienstprogrammes 'TERM' kann Ihr Computer zum Terminal für einen anderen Rechner gemacht werden.
- Programmlistings können mit Datum/Zeit-Stempeln versehen werden. Außerdem steht eine Datum/Zeit-Funktion als REL-File zur Verfügung.

## 1.2 End-Benutzer Lizenz-Vereinbarung

Diese Software und alle damit verbundenen Teile (insbesondere auch die Handbücher) sind urheberrechtlich geschützt. Das Urheberrecht (Copyright) an diesen Produkten liegt bei:

DIGITAL RESEARCH, INC., Pacific Grove, California, USA  
Dietmar Janich, J & K, Wuppertal  
Lutz Kleberhoff, MKC, Wuppertal

Die Benutzung dieser Software und der Handbücher ist nur gemäß folgender Regeln erlaubt:

### 1. Definitionen

- DRI ist eine Kurzbezeichnung für Digital Research. Digital Research ist der Autor und Eigentümer an dem Copyright des Betriebssystems CP/M PLUS ohne das BIOS und einigen Erweiterungen. J&K ist eine Kurzbezeichnung für Janich & Klass Computersysteme in Wuppertal. MKC ist eine Kurzbezeichnung für Michels und Kleberhoff Computer GmbH in Wuppertal. J&K/MKC (vertreten durch Dietmar Janich/Lutz Kleberhoff) sind die Autoren und Eigentümer an dem Copyright des CBIOS und der Erweiterungen zu diesem CP/M PLUS.
- Benutzer ist eine natürliche oder juristische Person, die das Recht zur Benutzung dieser Software erworben hat.
- Computer ist ein einzelner Microcomputer auf dem der Benutzer diese Software anwendet.
- Software ist die Menge der Computerprogramme, die zusammen mit dieser Lizenz ausgeliefert wird. Auch vom Benutzer modifizierte Programme, die ursprünglich zusammen mit dieser Lizenz ausgeliefert wurden, fallen darunter. Ebenso Programme, die auf Grund dieser Lizenz später an den Benutzer nachgeliefert werden.
- Lizenz meint diese Vereinbarung und die Rechte und Pflichten die sich auf Grund von bestehenden Gesetzen der Bundesrepublik Deutschland (z.B. Urheberrecht) ergeben.

### 2. Lizenz

DRI/J&K/MKC gestatten dem Benutzer diese serialisierte Kopie der Software auf einem einzigen Computer so lange zu benutzen, wie die Vereinbarungen dieser Lizenz eingehalten werden.

Der Benutzer darf diese Software nicht auf elektronischem Wege von einem Computer auf einen oder mehrere andere Computer übertragen (insbesondere auch nicht über ein öffentliches oder nicht öffentliches Netzwerk). Der Benutzer ist Eigentümer der Disketten und EPROMs, aber DRI/J&K/MKC sind die Eigentümer der

Software auf diesen Datenträgern und allen Kopien von diesen Datenträgern! Der Benutzer darf nur so viele Kopien von dieser Software anfertigen, wie für Backup-Zwecke notwendig sind (nicht mehr als 5). Auf diesen Datenträgern muß eine Copyright-notiz in der gleichen Weise wie auf den Original-Datenträgern angebracht werden. Falls diese Software auf mehr als einem Computer eingesetzt werden soll, muß die Lizenz für diese Software genau so oft erworben werden, wie Computer damit ausgestattet werden sollen.

Bei Verstoß gegen diese Vereinbarungen ist der Benutzer verpflichtet, die Software zu vernichten oder an die Eigentümer zurückzugeben. Eine Weiterbenutzung ist in diesem Falle nicht gestattet.

### 3. Weitergabe oder Reproduktion

Der Benutzer ist sich darüber im Klaren, daß eine unautorisierte Reproduktion und/oder unautorisierte Weitergabe dieser Software an Dritte einen Verstoß gegen das Urheberrecht darstellt und für DRI/J&K/MKC daraus ein wirtschaftlicher Schaden entsteht, der vom Benutzer zu ersetzen ist. Es ist nicht gestattet, irgend eine Kopie dieser Software an Dritte weiterzugeben, außer es werden alle Kopien (einschließlich der Originale und der Handbücher) weitergegeben. Der Benutzer muß den Händler, von dem er die Software erworben hat über diese Weitergabe informieren und Name und Adresse des Empfängers mitteilen, damit diesem eine Registrierungskarte übersandt werden kann. Die Registrierung des bisherigen Benutzers wird gleichzeitig gelöscht.

### 4. Adaptionen und Modifikationen

Der Benutzer ist Eigentümer aller Adaptionen und Modifikationen an dieser Software, sofern sie von Ihm selbst oder auf seinen Auftrag hin vorgenommen wurden. Falls jedoch diese Lizenz erlischt, darf kein Teil der Software, für die DRI/J&K/MKC das Copyright besitzen, mehr benutzt werden. Falls Adaptionen und Modifikationen des Benutzers ohne die Softwareteile von DRI/J&K/MKC unbenutzbar sind, dürfen auch diese Adaptionen und Modifikationen im Falle des Erlöschens der Lizenz nicht mehr benutzt werden.

### 5. Eingeschränkte Garantie

Die einzige Garantie, die DRI/J&K/MKC für diese Software geben, besteht darin, daß die Datenträger, auf denen die Software gespeichert ist, kostenlos ersetzt werden, falls sie bereits bei der Auslieferung defekt waren. Zerstörungen des Benutzers sind von dieser Garantie ausgeschlossen. Ein Garantieumtausch ist nur möglich, falls vom Benutzer bereits eine Registrierungskarte unterschrieben zurückgesandt wurde. DRI/J&K/MKC behalten sich das Recht vor, jederzeit ohne Benachrichtigung

Änderungen an der Spezifikation und an den Characteristica der Software vorzunehmen.

6. DRI/J&K/MKC GEBEN DARÜBER HINAUS KEINE WEITEREN GARANTIEN, WEDER EXPLIZIT NOCH IMPLIZIT. INSBESONDERE WIRD KEINE GEWÄHR FÜR DAS FEHLERFREIE FUNKTIONIEREN DER SOFTWARE ÜBERNOMMEN. EINE HAFTUNG FÜR SCHÄDEN, DIE IM ZUSAMMENHANG MIT DER BENUTZUNG DER SOFTWARE ENTSTEHEN, WIRD AUSDRÜCKLICH AUSGESCHLOSSEN. DABEI SPIELT ES KEINE ROLLE, VON WELCHER ART DIESER SCHADEN IST.
7. Die allgemeinen Geschäftsbedingungen des jeweiligen Verkäufers gelten für alle Punkte, die in dieser Vereinbarung nicht geregelt wurden. Die Unwirksamkeit einzelner Bestimmungen dieser Vereinbarung berühren die Gültigkeit der übrigen Bestimmungen nicht.
8. Mit Einsendung der Endbenutzer-Registrierungskarte werden die vorstehenden Bedingungen ohne Einschränkung akzeptiert.

1.3 Softwarepflege

Fehlerfreie Software ist sehr selten. Auch CP/M PLUS, die Systemprogramme und das CBIOS dürften an mehreren Stellen fehlerhaft sein. DRI/J&K/MKC sind bemüht, alle Fehler aufzuspüren und zu beseitigen. Wir sind dafür jedoch auf die Mithilfe der einzelnen Benutzer angewiesen.

Falls Sie einen Fehler im Betriebssystem oder den Systemprogrammen bemerken, bitten wir Sie, diesen Fehler genau zu beschreiben. Außerdem benötigen wir zur Reproduktion des Fehlers ein Beispiel mit dem der Fehler auftritt auf Diskette. Ohne genaue Fehlerbeschreibung und Beispiel auf Diskette kann ein Fehler durch uns nicht geortet werden.

Bevor Sie jedoch einen 'Fehler im Betriebssystem' melden, sollten Sie bedenken, daß die Wahrscheinlichkeit für einen Fehler in Ihrem eigenen Programm in der Regel deutlich höher ist, als die Wahrscheinlichkeit für einen Fehler im Betriebssystem. Außerdem ist das Verhalten eines Programmes bzw. des Betriebssystems gemäß den Beschreibungen und Definitionen der mitgelieferten Handbücher niemals ein Fehler.

Fehler werden von DRI durch 'Patch-Anweisungen' behoben, sobald sie bekannt und gefunden wurden. 'Patch-Anweisungen' sendet DRI an alle OEMs (also uns), die dann Ihre Masterdisketten nach diesen Anweisungen patchen. Diese Patches werden von DRI nummeriert. DRI nummeriert diese Patches auf 2 Arten:

- alle Patches werden durchnummeriert (Patch-Level)
- alle Patches für ein bestimmtes Programm werden durchnummeriert

Welche Patches in einem konkreten Programm aktiv sind, kann mit dem Dienstprogramm PATCH ermittelt werden:

PATCH programmname

gibt die installierten Patches in dem Programm mit dem Namen 'programmname' an, z.B.:

PATCH PATCH

```

CP/M 3 PATCH - Version 3.0
Current patches for PATCH.COM:
  1 2

```

Der Patch 2 für PATCH.COM war bereits der Patch 14 für CP/M PLUS. Die Durchnummerierung aller Patches bezeichnen wir mit Patch-Level des Betriebssystems.

Fehler werden von J&K/MKC durch UPDATE-Disketten behoben. Mit UPDATE-Disketten werden gleichzeitig Systemerweiterungen (neue Funktionen, neue Dienstprogramme, neue nützliche Module) zur Verfügung gestellt, wenn diese Systemerweiterungen zur neuen Standardversion gehören. Auf diesen UPDATE-Disketten befinden sich neue Dateien und Prozeduren zur Behebung von Fehlern und Erweiterungen des Systems. Gleichzeitig werden DRI-Patches als Prozeduren zur Verfügung gestellt.

Zusammen mit einer UPDATE-Diskette werden auch Änderungen des Handbuches versandt, falls solche vorhanden sind.

UPDATES werden ebenfalls durchnummeriert. Die Durchnummerierung der UPDATES bezeichnen wir als UPDATE-Level des Betriebssystems.

Auf einer der Systemdisketten befindet sich die Datei **LEVEL.CPM**.

LEVEL.CPM ist eine Textdatei, in der die beiden Levels beschrieben sind. Die Originalversion dieser Implementation hat den Patch-Level 14 und den Update-Level 0.

Updates für das Betriebssystem müssen lückenlos sein, d.h. es ist nicht möglich ein Betriebssystem mit dem Update-Level 0 auf den Update-Level 2 zu bringen, wenn nicht vorher der Update 1 durchgeführt wurde.

Update-Disketten erscheinen unregelmäßig. Sie werden grundsätzlich nur registrierten Benutzern angeboten. Der Preis für eine Update-Diskette beträgt DM 85,-- plus MWST in der jeweiligen gesetzlichen Höhe (zur Zeit der Drucklegung 14% ==> DM 96,90 incl. MWST). In diesem Preis sind die Versandkosten innerhalb der Bundesrepublik Deutschland enthalten.

Nach Eingang der Registrierungskarte wird jeder Benutzer automatisch für den nächsten Update vorgemerkt. Der Versand der ersten Update-Diskette erfolgt dann sofort nach Erscheinen kostenlos!

Weitere Update-Disketten müssen danach gekauft werden.

Alle registrierten Benutzer werden über das Erscheinen von neuen Update-Disketten kostenlos benachrichtigt.

Bitte geben Sie im Adressfeld Ihrer Registrierungskarte insbesondere bei großen Firmen Ihre Abteilung mit an, damit Benachrichtigungen Sie auch erreichen können!



**1.4 Mögliche Hardware-Zusammenstellungen**

Diese CP/M PLUS Implementation 1.6 unterstützt folgende Hardware-Komponenten:

CPU-Karten:	HKM-CPU, CPU I, CPU II, CPU III, EPC 1.6, EPC II
FDC-Karten:	FDC 5, FDC 8/5, FDC II, DPS-IO
HDC-Karten:	HDC 5
RAM-Karten:	bis 1MByte dyn. RAM, CMOS-RAM, EPROM-Karten, bis zu 3 SMB-Platinen mit je 1MByte RAM
RAE-Karten:	RAE-Karte Rev. 2/3, RAE-Karte Rev. 4/5
I/O-Karten:	VIDEO I, P/SIO, SIO, DPS-IO, GPIB (nur Centronics)
SASI-Controller:	XEBEC S1410 ab Rev. E, XEBEC S1410A

Außerdem werden noch eine Reihe von Floppy- und Winchester-Laufwerken unterstützt.

Bei Benutzung einer HKM-CPU oder CPU I ist eine RAE-Karte erforderlich. Bei Benutzung einer CPU II, CPU III, eines EPC 1.6 oder EPC II ist eine RAE-Karte im System nicht erlaubt.

Bei Benutzung einer CPU III ist eine P/SIO-Karte, DPS-IO-Platine oder eine VIDEO I erforderlich.

Bei Benutzung eines EPC II sind keine Speicherkarten erlaubt. Ferner können keine DMA-Karten (FDC 8/5) benutzt werden.

Die Verwendung eines EPC 1.6 schließt den FDC 8/5 und die FDC II (und somit auch den XEBEC) aus.

Bei Benutzung eines SASI-Controllers ist eine FDC II-Karte erforderlich.

Es muß genau eine der CPU-Karten im System vorhanden sein.  
FDC-Karten dürfen miteinander kombiniert werden.  
Es darf nur maximal eine RAE-Karte im System vorhanden sein.  
I/O-Karten dürfen miteinander kombiniert werden.  
Es darf nur maximal ein SASI-Controller im System vorhanden sein.  
Einige der obigen Platinen schließen sich gegenseitig aus!

Mindestens eine der obigen CPU-Karten, eine der obigen FDC-Karten, eine Adreßerweiterungslogik (CPU II, CPU III, EPC 1.6, EPC II oder RAE-Karte), 128KByte RAM und 2 Laufwerke, von denen eines ein Floppy-Laufwerk sein muß, sind in einem System erforderlich.

Fremdkarten können nur als zusätzliche I/O-Karten (nicht Boot-Konsole) eingebracht werden. Fremde CPU-Karten, Floppy-Controller oder Adreßerweiterungslogiken werden von diesem Betriebssystem nicht unterstützt!

Jumperpläne für Platinen:

- HKM-CPU:** Promsatz 2 (beide Steckplätze 4KByte EPROM) ist erforderlich.  
 Jumperfeld hinter der VG-Leiste: 2C-2B, 5C-5B,  
 2A-2B, 5A-5B  
 Jumper zwischen IC 7 und IC 10: Y-Z
- CPU I:** S2: offen ; S3: gesetzt  
 J1: S1 = b ; S2 = b ; S3 = a ; S4 bis S7 = a  
 J2: SA4 = a ; SA5 = b ; SA6 = b ; SA7 = b  
 J3: B, C, G  
 J4: alle offen  
 weitere Jumper sind abhängig von der Hardware
- CPU II:** J1: rechts  
 J2: links  
 J3: hardwareabhängig (BAI/BA0)  
 J4: nach oben  
 J5: links  
 Beim EPROM müssen die Pins 1 und 28 gebrückt sein
- Die Richtungsangaben gelten für eine Sicht auf die Bestückungsseite der Platine mit der VG-Leiste **links**.
- CPU III:** S1: hardwareabhängig (BAI/BA0)  
 S2: hardwareabhängig (RAM-Typ)  
 S3: a  
 SA4: geschlossen  
 SA5: offen  
 SA6: geschlossen  
 SA7: geschlossen
- EPC II:** J5: hardwareabhängig (Floppygröße)
- RAE3:** Portadresse: A2: geschlossen  
 A3: geschlossen  
 A4: offen  
 A5: geschlossen  
 A6: geschlossen  
 A7: geschlossen  
 S6: geschlossen  
 alle anderen Jumper sind hardwareabhängig
- RAE4:** J1: hardwareabhängig (BAI/BA0)  
 J2: S14, S12, S13 gesetzt  
 J3: S2, S3, S5, S6, S7 gesetzt ; S4 offen  
 J4: hardwareabhängig (RAM-Bestückung)
- FDC 5:** keine Jumper vorhanden  
 Bei TEAC oder kompatiblen Laufwerken ist eine Brücke von Pin 6 nach Pin 34 auf der Pfostenleiste zu löten. Die Anschlüsse an Pin 34 und Pin 4 sind abzutrennen (durchkratzen).

**FDC 8/5:** 8"-Betrieb: Adreßprom 1068.4 erforderlich (Adresse 4)  
 J1: hardwareabhängig (BUSAK)  
 J2: hardwareabhängig (Philips/andere)  
 J3: geschlossen  
 J4: hardwareabhängig (nur Philips)  
 J5: nach unten

5,25"-Betrieb: Adreßprom 1068.1 erforderlich (Adresse 0)  
 J1: hardwareabhängig (BUSAK)  
 J2: hardwareabhängig (Philips/andere)  
 J3: offen  
 J4: hardwareabhängig (nur Philips)  
 J5: nach oben

Die Richtungsangaben gelten für eine Sicht auf die Bestückungsseite der Platine mit der VG-Leiste links.

Bei TEAC oder kompatiblen Laufwerken ist eine Brücke von Pin 6 nach Pin 34 auf der Pfostenleiste zu löten.

**FDC II:** J1: falls vorhanden, immer offen (BAI/BA0)  
 J2: hardwareabhängig (Adreßeinstellung der RAMs)  
 J3: offen bei 4MHz; sonst geschlossen  
 J4: geschlossen bei BASF oder kompatibel  
 J5: geschlossen bei TEAC oder kompatibel

**VIDEO I:** alle Jumper hardwareabhängig

**P/SIO:** J0: Portadresse A4: offen  
 A5: geschlossen  
 A6: offen  
 A7: geschlossen  
 Alle weiteren Jumper sind hardwareabhängig.

**GPIB:** Adressen: S2: A4 = 1  
 A5 = 1  
 A6 = 0  
 A7 = 0  
 Alle weiteren Jumper sind hardwareabhängig.

**HDC 5:** keine Jumper vorhanden

**XEBEC:** SS = 5 (512Byte pro Sektor!)  
 Dieser Jumper steht bei der Auslieferung üblicherweise falsch!  
 SASI-Adresse ist mit 0 einzustellen. Normalerweise ist die SASI-Adresse bei Auslieferung richtig eingestellt.

**RAM:** Die RAM-Adressen müssen lückenlos ab Adresse 0 eingestellt werden. Auf der CPU II und der CPU III kann die Startadresse nicht eingestellt werden. Sie liegt immer auf der Adresse 0. Bei den weiteren RAM-Karten (auch FDC II) kann die RAM-Startadresse jeweils eingestellt werden. Beachten Sie dazu die Beschreibungen in den Hardware-Unterlagen zu den Platinen.

**SMB:** DIL-Schalter: 42H: DIL1 = 0 (on)  
 DIL2 = 1 (off)  
 DIL3 = 0 (on)  
 DIL4 = 0 (on)  
 DIL5 = 0 (on)  
 DIL6 = 0 (on)  
 DIL7 = 1 (off)  
 DIL8 = 0 (on)

S0, S1, S2, S3, S5 sind hardwareabhängig.  
 S4 nicht gesetzt ==> Benutzung wie unter RAM  
 beschrieben.  
 S4 gesetzt ==> Bestückung mit 1MByte erforderlichlich  
 und Benutzung als SMB-DISK.  
 S16, S17: Bei Benutzung als SMB-DISK immer 0.  
 S18, S19: Bei Benutzung als SMB-Disk nur folgende 3  
 Stellungen erlaubt: S18 = 1, S19 = 0 (Adr. 4)  
 S18 = 0, S19 = 1 (Adr. 8)  
 S18 = 1, S19 = 1 (Adr.12)

**DPS-IO:** alle Jumper hardwareabhängig.

**EPR0M-Simulator:**

J1: Adreßeinstellung = 43H ==> SA0 = 1 (offen)  
 SA1 = 1 (offen)  
 SA2 = 0 (gesetzt)  
 SA3 = 0 (gesetzt)  
 SA4 = 0 (gesetzt)  
 SA5 = 0 (gesetzt)  
 SA6 = 1 (offen)  
 SA7 = 0 (gesetzt)

J2: genau einer gesetzt (beliebig)  
 J3: alle 1 (kein RAM auf der Obergruppenadresse 0FH  
 erlaubt!)

J4: hardwareabhängig  
 Der Schalter zum Umschalten zwischen Master und Slave  
 darf nicht angeschlossen werden!

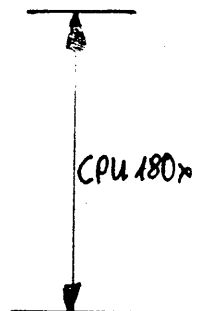
**jk82 GRAFIK II:**

DIL-Schalter: OBAH: DIL1 = 1 (off)  
 DIL2 = 0 (on)  
 DIL3 = 1 (off)  
 DIL4 = 1 (off)  
 DIL5 = 1 (off)  
 DIL6 = 0 (on)  
 DIL7 = 1 (off)  
 DIL8 = BAI/BA0

Reservierte I/O-Portbelegung für CBIOS 1.6

Die nachfolgende Tabelle gibt einen Überblick über die von CBIOS 1.6 und einigen unter CP/M PLUS ablauffähigen Dienstprogrammen benutzten I/O-Ports. Die mit '\*' gekennzeichneten I/O-Ports werden nur benutzt, wenn die betreffende Platine tatsächlich im System vorhanden ist. Alle anderen Adressen werden von CBIOS 1.6 bzw. dem Urlader immer benutzt und können vom Anwender nicht anderweitig belegt werden!

Adresse (Hex)	Adreßein- stellung	Platine
00-03	! Prom	! FDC 5 oder FDC 8/5, NON DMA, 5,25"-Laufwerke
00-03	! PAL	! DPS-IO
04-07	! Prom	! FDC 8/5, DMA, 8"-Laufwerke
08-0F	! Prom	! HDC 5
10-13	*! Schalter	! Adreßerweiterungsplatine RAE Rev. 2 bis 5
14-17	*! Schalter	! (E)PROM-Programmer
18-1B	*! Schalter	! Universal-Programmer
1C-1F	*!	! für zukünftige Benutzung reserviert (EPR. SIMU II)
20-2F	! Jumper	! CPU III / CPU 180x
30-3F	*! Jumper	! GPIB
40-41	!	! reserviert für Testzwecke
42	! Schalter	! SMB
43	*! Jumper	! EPROM-Simulator
44-47	*! Jumper	! Slave CPU II
48-4B	*! Jumper	! Slave 186
4C-4F	*!	! für zukünftige Benutzung reserviert
50-5F	! Schalter	! P/SIO
50-5F	! PAL	! DPS-IO
60-6F	*!	! SIO
70-7F	*!	! für zukünftige Benutzung reserviert
80-87	*!	! für zukünftige Benutzung reserviert
88-8F	*! Schalter	! BAW
90-9F	*!	! für den Anwender frei
A0-B7	*!	! für den Anwender frei
B8-B9	! Prom	! VIDEO I
BA-BB	! Schalter	! jk82 GRAFIK II
BC-BF	*!	! für zukünftige Benutzung reserviert
C0-CF	*!	! für den Anwender frei
D0-D1	*!	! für zukünftige Benutzung reserviert
D2-DF	*! Prom	! M.K.C. GRAFIK II
DC-EF	*! Prom	! EPC II
E0-EF	! Prom	! HKM-CPU oder CPU II
E0-EF	! Jumper	! CPU I
E0-F7	*! Prom	! EPC 1.6
F0-FE	! Prom	! FDC II
FF	! Prom	! HDC 5



Hier nicht aufgeführte Baugruppen werden vom CP/M PLUS nicht unterstützt. Software zur Benutzung der hier nicht aufgeführten Baugruppen zum Betrieb unter CP/M PLUS ist ebenfalls nicht vorgesehen. Diese zusätzlichen Baugruppen können deshalb beliebige Adressen belegen.

Speicherbelegung eines CP/M PLUS-Systems mit CBIOS 1.6

Die Speicheraufteilung eines jk82/MKC-CP/M PLUS ist sehr stark abhängig von der jeweiligen Installation des Betriebssystems. Nur sehr wenige Speicheradressen sind für alle Zusammenstellungen gleich:

Adresse 0000H bis 00FFH in Bank 1 (User-Bank):  
Page 0 des Betriebssystems wie in den CP/M PLUS Handbüchern beschrieben

Adresse OFF00H bis OFF2FH im Common-Memory:  
Interrupt-Vektoren des CBIOS 1.6, falls notwendig

Adresse OFF30H bis OFF6FH im Common-Memory:  
Interrupt-Vektoren des Anwenders, falls notwendig

Adresse OFF70H bis OFFBFH im Common-Memory:  
Stack 1 des CBIOS 1.6

Adresse OFFC0H bis OFFEFH im Common-Memory:  
Stack 2 des CBIOS 1.6

Adresse OFFF0H bis OFFFFH im Common-Memory:  
Hardware-Vektor des CBIOS 1.6 (CBIOS-Konstanten)

Der Hardware-Vektor enthält die vom Urlader gefundenen Systemdefinitionen. Er liegt auf Adresse OFFF0H im COMMON-Bereich. Die in ihm gespeicherten Werte dürfen von Anwendungsprogrammen gelesen werden! Sie werden im CBIOS beim Initialisieren des Systems ausgewertet.

**Definition des Hardware-Vektors: (in hexadezimaler Darstellung)**

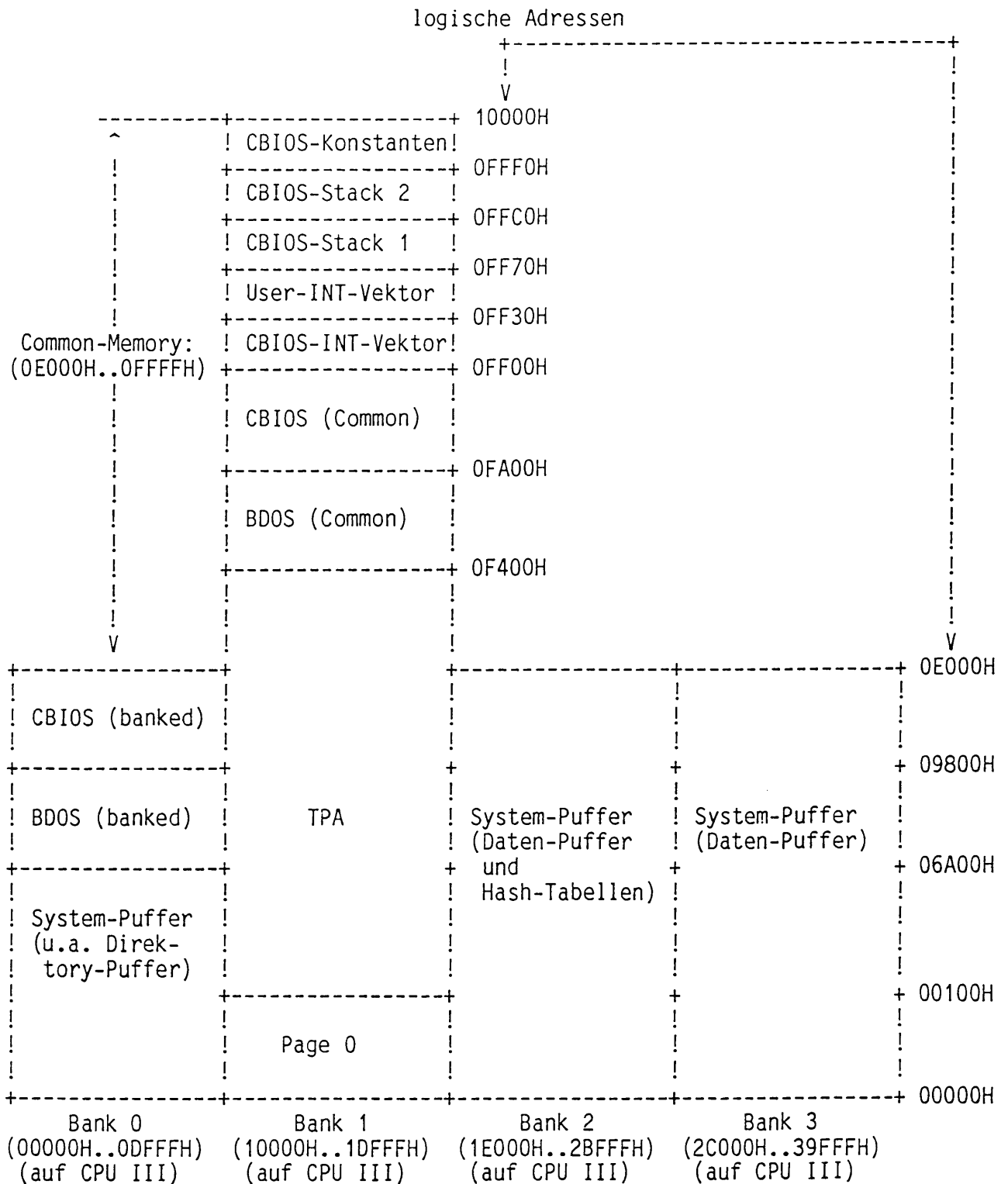
OFFF0H: Systemnummer	1 = CPU II	
	2 = CPU I oder HKM-CPU	
	3 = CPU III	
	4 = EPC II	
	5 = EPC 1.6	
	6 = CPU 180x	
OFFF1H: Boot-Disk	00H = FDC 5 rdrv 0 (5,25")	
	01H = FDC 5 rdrv 1 (5,25")	
	10H = FDC 8/5 rdrv 0 (8")	
	11H = FDC 8/5 rdrv 1 (8")	
	20H = FDC II rdrv 0 (5,25")	
	21H = FDC II rdrv 1 (5,25")	
	30H = FDC II rdrv 0 (8")	
	31H = FDC II rdrv 1 (8")	
	40H = SASI rdrv 0	
	60H = EPC II rdrv 0	
	61H = EPC II rdrv 1	
	70H = HDC 5 RDRV 0	
	90H = EPC 1.6 rdrv 0 (5,25")	
	91H = EPC 1.6 rdrv 1 (5,25")	
	A0H = CPU 180x rdrv 0 (5,25")	
	A1H = CPU 180x rdrv 1 (5,25")	
	B0H = CPU 180x winch rdrv 0	
56H = ext. CRAM-Disk geschaltet		
87H = " EPROM- " "		
53H = interne RAM-Disk (180x)		
	82H = EPROM - DISK	
	54H = RAM-Disk (kurz)	
		1.4-6
		geschaltet
		linear
		85 = interne EPROM-Disk 1802
		54 = interne CMOS-Ramdisk 180.
		51 = CMOS-Ramdisk 11i...

0FFF2H:	Disk-Controller	bit 0 = FDC 5 oder EPC 1.6 bit 1 = FDC 8/5 bit 2 = FDC II (5,25") bit 3 = FDC II (8") bit 4 = XEBEC bit 5 = HDC 5	bit 6 = FDC on CPU 180x bit 7 = HDC on CPU 180x
0FFF3H:	Boot-Konsole	0 = unbekannt 1 = VIDEO I 2 = SIOB (Adresse: 0E2H, 0E3H) 3 = SIO2 (Adresse: 052H, 053H)	
0FFF4H:	Baudrate	07H = 600 Baud 08H = 1200 Baud 0AH = 2400 Baud 0CH = 4800 Baud 0EH = 9600 Baud 0FH = 19K2 Baud	(kann gepatcht werden)
0FFF5H:	Systemtakt	0 = unbekannt 4 = 4 MHz 5 = 5 MHz 6 = 6 MHz 8 = 8 MHz	
0FFF6H:	SIO-Port-Adresse	0E3H oder 053H	
0FFF8H:	5" step-rate-time		kann gepatcht werden
0FFF9H:	8" step-rate-time		kann gepatcht werden
0FFFAH:	5" head-load-time		kann gepatcht werden
0FFFBH:	8" head-load-time		kann gepatcht werden (Adresse 70..7F im EPROM!)

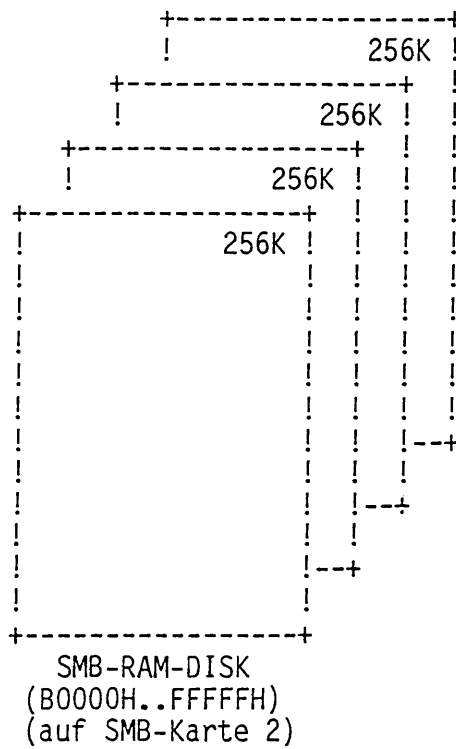
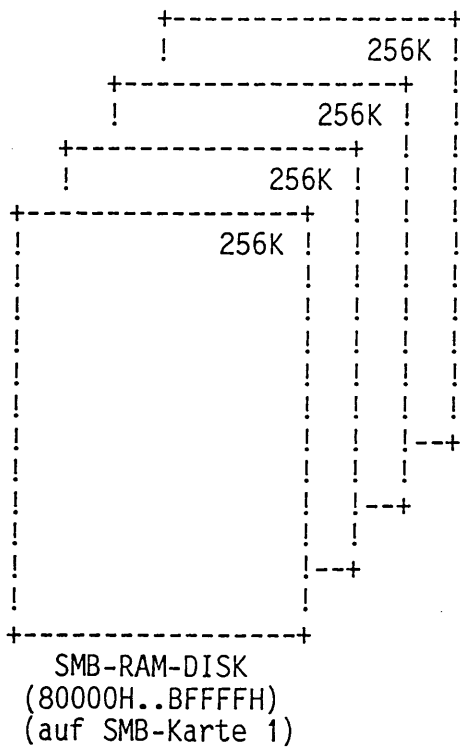
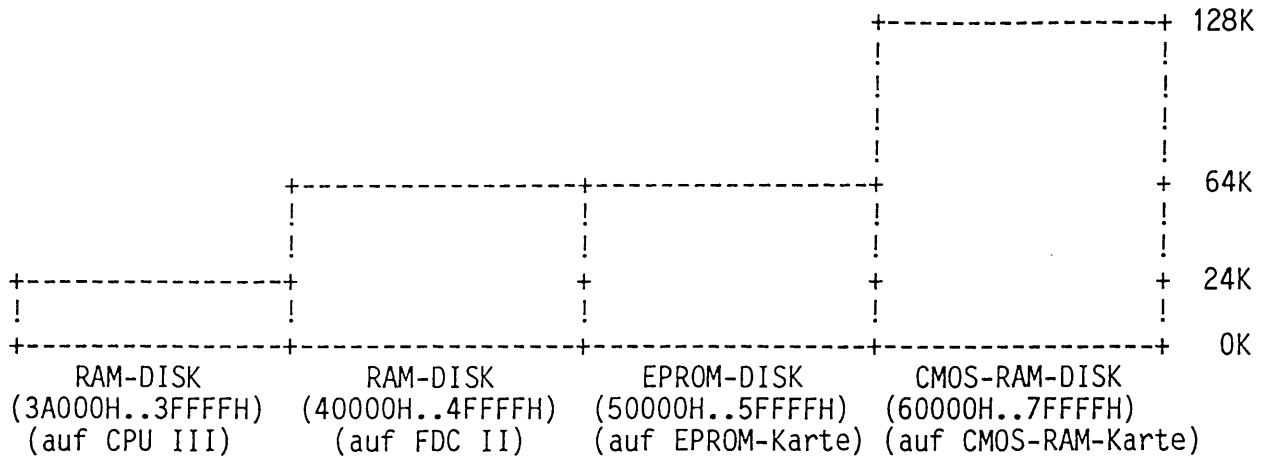
Alle anderen Adressen sind abhängig von der jeweiligen Installation! Die Startadresse des Common-Memories ist aus dem System-Control-Block (SCB) des Betriebssystems abzufragen (siehe BDOS-Funktion 49: Get/Set System-Control-Block).

**Achtung:** Der System-Control-Block ist in der Datei SCB.ASM fest auf Adresse 0FE00H definiert. Trotzdem liegt er in einem fertig installierten Betriebssystem nicht auf dieser Adresse! Das Programm 'GENCPM' verschiebt nach dem Linken das gesamte Betriebssystem und damit auch den SCB!

Zur Demonstration einer möglichen Speicheraufteilung sei hier die Aufteilung des in Kapitel 2.1 beispielhaft installierten Betriebssystems angegeben:







### Bemerkungen zu Disk-Laufwerken

Auf dem internationalen Markt wird eine unüberschaubare Anzahl von Floppy- und Winchester-Laufwerken angeboten. Es ist unmöglich, ein Betriebssystem zu schreiben, das für alle diese Laufwerke geeignet ist.

Viele Laufwerke sind zu irgend welchen anderen 'kompatibel'. Was unter 'kompatibel' verstanden wird, ist meist nicht zu erfahren. So gibt es z.B. 5,25"-Laufwerke, die zu 8" 'kompatibel' sind. Die Anschlußbelegung des Steckers ist jedoch meist ganz anders. Was bedeutet also 'kompatibel' ?

#### Definition:

Wir betrachten ein Laufwerk als kompatibel, wenn die Steckerbelegung identisch ist und alle Leitungen die gleiche Bedeutung und die gleichen Eigenschaften haben. Außerdem muß sich ein kompatibles Laufwerk in allen Einzelheiten vollkommen gleich verhalten wie das Laufwerk zu dem es kompatibel ist. Es ist nur dann kompatibel, wenn die Stecker von dem einen Laufwerk abgezogen und auf das andere Laufwerk aufgesteckt werden können, ohne das im Computer irgend eine Veränderung notwendig ist. Zwischen kompatiblen Laufwerken können Disketten problemlos ausgetauscht werden.

Ein BASF 6108 Laufwerk ist nicht kompatibel zu einem TEAC 55B Laufwerk, da das Signal 'READY' auf dem Stecker an einer anderen Stelle zur Verfügung steht und die Timings voneinander abweichen.

Diese Implementation des CP/M PLUS unterstützt folgende Laufwerke:

NEC 1165 (8"), TEAC FD55B oder NEC 1053 (5,25"/40Track), TEAC FD55F oder NEC 1055 (5,25"/80Track), TEAC FD35F oder NEC 1035 (3,5"/80Track), BASF 6108 (5,25"/40Track), RODIME R0204 oder NEC 5126 (Winchester) und alle dazu kompatiblen Laufwerke.

Welche Laufwerke dazu kompatibel sind, wissen wir nicht. Garantien für das Funktionieren anderer Laufwerke übernehmen wir nicht! Wenn noch andere Laufwerke in dem Install-Menue von CPMINST enthalten sind, so besagt das lediglich, daß es mindestens ein Laufwerk dieser Bauart gibt, das zusammen mit diesem Betriebssystem funktioniert hat. Ob es noch weitere Laufwerke dieser Bauart gibt, die mit diesem Betriebssystem funktionstüchtig sind, wissen wir nicht!

Eine Überprüfung von fremden Laufwerken auf Funktionstüchtigkeit mit diesem Betriebssystem ist nur gegen Berechnung möglich. Die Überprüfung dauert erfahrungsgemäß mehrere Tage. Pro Tag ist mit einem Tagessatz von ca. DM 800,- zu rechnen.

Falls Sie selbst ein fremdes Laufwerk in Betrieb nehmen wollen, so ist das dafür notwendige MACRO in **DISKDEF.LIB** und der Eintrag in **SYSDEF.DAT** zu schreiben (siehe 4.4 Disk Routinen).

### 1.5 Hinweise zur Inbetriebnahme des Betriebssystems

Dieses Betriebssystem muß für die jeweilige Hardware-Zusammenstellung installiert werden. Bevor Sie das tun, sollten Sie folgende Punkte unbedingt beachten:

Die Disketten sind schreibgeschützt. Eine 5,25"-Diskette ist schreibgeschützt, wenn die Schreibschutzkerbe überklebt ist, und eine 8"-Diskette ist schreibgeschützt, wenn die Schreibschutzkerbe nicht überklebt ist. Eine 3,5"-Diskette ist analog zu einer 8"-Diskette schreibgeschützt, wenn der Schreibschutzschieber offen ist.

Beseitigen Sie auf keinen Fall den Schreibschutz an den Original-Disketten! Es ist unter keinen Umständen gestattet auf die Original-Disketten zu schreiben. Wir lehnen grundsätzlich jegliche Unterstützung ab, wenn auf die Original-Disketten geschrieben wurde!

Falls Sie keine VIDEO I- oder EPC II-Platine in Ihrem System haben, benötigen Sie ein Terminal mit V.24(RS232C)-Schnittstelle. Dieses Terminal muß an dem Kanal B der SIO(DART) der jeweiligen CPU-Karte oder, falls Sie eine CPU III besitzen, an der P/SIO- oder DPS-IO-Karte (Kanal B der 1. SIO) angeschlossen werden. Das Terminal ist auf 19200Baud (nicht HKM-CPU und nicht CPU II bei 4MHz), 9600Baud, 4800Baud, 2400Baud oder 1200Baud mit 8 Datenbit, 1 Stopbit und kein Parity einzustellen. Bei einer CPU 180x-Platine sind nur die Baudraten 19200Baud, 9600Baud oder 4800Baud möglich, falls der Urlader noch ungepatcht ist (siehe Kapitel 4.1)!

Sie sollten nun zuerst folgende Kapitel in diesem Handbuch lesen:

- 1.2 End-Benutzer Lizenz-Vereinbarung
- 1.4 Mögliche Hardware-Zusammenstellungen
- 2.1 CPMINST (Betriebssystem generieren)
- 2.3 FORMAT (Disketten formatieren)
- 4.1 Urlader LDR 1.7

Erst danach sollten Sie das Betriebssystem booten, falls Sie sicher sind, daß Ihre Hardware in Ordnung ist. Das Betriebssystem befindet sich auf der Diskette #1.

Falls Sie noch nicht sicher sind, daß Ihre Hardware in Ordnung ist, empfehlen wir Ihnen folgende Vorgehensweise:

Versuchen Sie von einer leeren Diskette zu 'booten'. Sie können nun beobachten, ob der Bootstraploader korrekt arbeitet. Der vom Loader gemeldete Systemtakt (4, 5, 6MHz) muß stimmen, sonst arbeitet Ihr Floppy-Controller fehlerhaft! Außerdem muß ein Diskzugriff erfolgen. Da die Diskette nicht lesbar ist, wird der Loader nun auf einem weiteren Laufwerk nach einer gültigen Diskette suchen. Legen Sie deshalb auch dort eine 'leere' Diskette ein und beobachten Sie den Diskzugriff. Langes Herumfahren auf den Disketten darf nicht vorkommen (lediglich ein

erstes Positionieren auf den Track 0). Der Loader muß nun abwechselnd auf Ihren Laufwerken nach einer gültigen Diskette suchen.

Nach dem Booten meldet das Betriebssystem, welche Laufwerke es an welchen Controllern kennt. Beachten Sie, daß es vorkommen kann, daß Ihnen kein Laufwerk A zur Verfügung steht (controller- und laufwerksabhängig). Schreiben Sie sich nun auf, welche Laufwerke Sie von den dem System bekannten Laufwerken an Ihrem Computer angeschlossen haben.

Sie müssen nun eine Diskette formatieren. Das können Sie nur auf einem dem System bekannten Laufwerk tun! Nehmen Sie nicht das Laufwerk, von dem Sie gebootet haben!

Das FORMAT-Programm befindet sich auf der Diskette #1!

Das FORMAT-Programm fordert Sie auf, eine Diskette zum Formatieren einzulegen. Nehmen Sie an dieser Stelle zuerst die Diskette #1 wieder heraus, damit diese auch bei einer Fehleingabe und defekten Laufwerken nicht aus Versehen formatiert wird.

Bei 40 Track-Laufwerken sollten Sie das Format "CPM3 5/40 DD DS" wählen, da Sie sonst beim Installieren Probleme mit der Diskettenkapazität bekommen. Bei 80 Track-Laufwerken sollten Sie das Format "ZDOS 80 trk DS" benutzen.

Wenn Sie mit dem Formatieren fertig sind, legen Sie die Diskette #1 wieder ein. Tun Sie dies aber erst, wenn Sie bereits wieder im Betriebssystem sind (Laufwerksprompt!).

Rufen Sie nun PIP auf.

Nachdem PIP sich auf Ihrer Console gemeldet hat, nehmen Sie die Diskette #1 wieder aus Ihrem Laufwerk heraus und legen Sie stattdessen die Diskette #3 ein.

Kopieren Sie mit PIP nun die Diskette #3 auf die soeben formatierte neue Diskette. Geben Sie dazu z.B. folgenden Befehl ein:

**b:=a:\*. \*ÄvÜ**                      Ä,Ü sind eckige Klammern

**b** ist dabei das Ziel-Laufwerk und **a** das Quellen-Laufwerk mit der System-Diskette #3. Diese Bezeichnungen können bei Ihrem System anders sein! Kopieren Sie unbedingt nur mit diesem Befehl! Benutzen Sie an dieser Stelle auf keinen Fall das Dienstprogramm 'DISKCOPY'!

Nachdem Sie nun die Diskette #3 kopiert haben, können Sie mit der Installation beginnen. Erst ein fertig installiertes System hat alle angegebenen Eigenschaften.

Falls Sie nur 1 Diskettenlaufwerk und eine Winchester haben, müssen Sie die Systeminstallation auf der Winchester vornehmen. Sie dürfen dann aber nicht von dieser Winchester booten! Nach der Systeminstallation ist deshalb erst eine Diskette zu formatieren

und das System auf die Diskette zurückzukopieren! Diese Diskette muß dann gebootet und die Winchester unter dem installierten Betriebssystem neu formatiert werden.

Falls Sie überhaupt nur ein Laufwerk haben, müssen Sie zuerst ein 2. Laufwerk kaufen, da eine Systeminstallation mit nur einem Laufwerk unmöglich ist.

1.6 ASCII-Tabelle (deutsche Version)

Bit 3..0										Bit 6..4
↓		000	001	010	011	100	101	110	111	←---
V		0	1	2	3	4	5	6	7	
0000	0	NUL	DLE	SPACE	0	§	P	`	p	
0001	1	SOH	DC1	!	1	A	Q	a	q	
0010	2	STX	DC2	"	2	B	R	b	r	
0011	3	ETX	DC3	#	3	C	S	c	s	
0100	4	EOT	DC4	\$	4	D	T	d	t	
0101	5	ENQ	NAK	%	5	E	U	e	u	
0110	6	ACK	SYN	&	6	F	V	f	v	
0111	7	BEL	ETB	'	7	G	W	g	w	
1000	8	BS	CAN	(	8	H	X	h	x	
1001	9	HT	EM	)	9	I	Y	i	y	
1010	A	LF	SUB	*	:	J	Z	j	z	
1011	B	VT	ESC	+	;	K	Ä	k	ä	
1100	C	FF	FS	,	<	L	Ö	l	ö	
1101	D	CR	GS	-	=	M	Ü	m	ü	
1110	E	SO	RS	.	>	N	'	n	ß	
1111	F	SI	US	/	?	O	_	o	DEL	

Diese Tabelle zeigt die deutsche Version des ASCII-Codes!

Bitte beachten Sie, daß bei einigen Druckern '<' mit 'µ' und '>' mit '°' belegt ist!

**1. Generieren eines CP/M PLUS Betriebssystems mit CPMINST**

**CPMINST** ist ein Dienstprogramm zur Erleichterung der Installation des Betriebssystems CP/M PLUS auf eine vom Anwender bestimmte Hardware-Zusammenstellung. **CPMINST** ist menuegesteuert und fragt vom Anwender alle Baugruppen des jeweiligen Computers ab.

**CPMINST benötigt folgende Eingabe-Dateien:**

<b>CPMINST.MSG</b>	Datei mit Menuetexten für CPMINST.
<b>SYSDEF.DAT</b>	Datei mit den Beschreibungen der einzelnen in CBIOS 1.8 möglichen Systemkomponenten. Diese Datei sollte nicht modifiziert werden!
<b>CPMGEN.DAT</b>	Aus dieser Datei wird CPMGEN.SUB erzeugt. Alle in dieser Datei enthaltenen Befehle werden in die Datei CPMGEN.SUB übernommen. Der letzte Befehl muß der LINK-Befehl sein. Dieser Befehl muß mit einem '&' beendet werden, da CPMINST noch weitere Rel-Dateien hinzufügt!
<b>GENCPM.DAT</b>	Diese Datei wird für das DRI-Programm GENCPM von CPMINST modifiziert. CPMINST benötigt immer die ursprünglich mit diesem Betriebssystem ausgelieferte GENCPM.DAT oder eine von CPMINST aus dieser erzeugten GENCPM.DAT!

Außerdem wird überprüft, ob folgende Dateien vorhanden sind:

**RESBDOS3.SPR, BNKBDOS3.SPR, RMAC.COM, LINK.COM**

Diese Dateien müssen auf den Disketten 1 oder 3 vorhanden sein!

**CPMINST erzeugt folgende Ausgabedateien:**

<b>SYSDEF.LIB</b>	SYSDEF.LIB enthält die Systemdefinition in Form von 'EQU'-Anweisungen für bedingte Assemblierungen. SYSDEF.LIB wird für CHARIO.ASM und DISKS.ASM als 'MACLIB' benötigt.
<b>DISKS.ASM</b>	DISKS.ASM enthält die Definition aller aktiven Laufwerke. DISKS.ASM besteht im wesentlichen aus MACRO-Aufrufen für jedes in CPMINST definierte Laufwerk.
<b>VERSION.CPM</b>	VERSION.CPM ist eine Textdatei, in der die Hardware-Komponenten (incl. der Laufwerke) des gerade installierten Betriebssystems dokumentiert sind. VERSION.CPM kann mit 'TYPE' jederzeit auf den Bildschirm ausgegeben werden. Dadurch ist es möglich, später noch einmal

nachzuschauen, wie dieses Betriebssystem installiert wurde.

**GENCPM.DAT** GENCPM.DAT wird gemäß den Eingaben in CPMINST modifiziert. Dadurch sind in GENCPM nur noch 2 Eingaben von Hand notwendig.

**CPMGEN.SUB** CPMGEN.SUB ist eine SUBMIT-Datei zur automatischen Erzeugung eines Betriebssystems mit den in CPMINST eingegebenen Daten. CPMGEN.SUB wird von CPMINST automatisch gestartet.

Nach Ablauf der Prozedur ist die Datei CPM3.SYS neu erstellt worden. Eine eventuell vorher vorhandene Datei CPM3.SYS wird dabei gelöscht! Wir empfehlen deshalb dringend, vorher eine Kopie der Diskette 3 anzufertigen!

Die einzelnen Abfragen von CPMINST erklären sich im allgemeinen selbst. Lediglich bei folgenden Punkten scheint uns eine Anmerkung angebracht:

- **CPMINST** fragt nach dem im SYSTEM vorhandenen RAM in KByte. Hier ist das gesamte vorhandene RAM einzugeben. Diese Zahl muß ein ganzzahliges Vielfaches von 64 sein! Falls einige RAM-Banks in Ausnahmefällen nicht vom Betriebssystem verwaltet werden sollen, so sind diese hier abzuziehen! Nicht vom Betriebssystem verwaltete RAM-Banks können nur am Ende des Adreßraumes liegen. Bei der Speicheraufteilung wird zwischen EPROM, CMOS-RAM, ungeschaltetem dyn. RAM und auf der SMB-Platine geschaltetem RAM unterschieden! In der geschalteten Betriebsart muß die SMB-Platine mit 1MByte RAM bestückt sein! Bei Benutzung einer CPU 180x-Platine darf hier nur internes RAM angegeben werden.
- **CPMINST** fragt nach den zusätzlichen Banks, die das CP/M PLUS über 2 Banks hinaus für Puffer benutzen soll. Hier ist im allgemeinen eine 0 (bzw. einfach RETURN) einzugeben. Lediglich bei Installation von mindestens einem Winchesterlaufwerk empfiehlt sich hier eine Eingabe von 1 oder 2. Mehr als 2 zusätzliche Banks sind nicht sehr sinnvoll. Der restliche verfügbare Speicher wird für eine **RAMDISK** benutzt, falls sie selektiert wird.

Nach dem Start der von CPMINST ausgelösten Prozedur sind noch 2 Eingaben in GENCPM notwendig. Vorher erscheinen jedoch noch sehr viele Ausgaben auf dem Bildschirm. Unter anderem meldet GENCPM mehrfach 'ERROR: Memory conflict - segment trimmed'. Diese Fehlermeldung rührt daher, daß von CPMINST alle Speicherbereiche in GENCPM.DAT grundsätzlich zu groß eingetragen werden. GENCPM trimmt diese Speicherbereiche dann immer auf den größtmöglichen Wert. Genau das soll aber erreicht werden. Diese Fehlermeldungen sind deshalb bedeutungslos.

GENCPM fragt den Anwender nach den Directorypuffern und nach den Datenpuffern für einen Drive. Alle anderen Drives benutzen dann



dieselben Puffer. Leider gibt **GENCPM** den noch verfügbaren Platz (Available space) in Einheiten zu 256Byte an, während die Anzahl der Puffer in Einheiten zu 1024Byte abgefragt wird. Der verfügbare Platz muß deshalb zuerst in Einheiten zu 1024Byte umgerechnet werden (Division durch 4).

Zuerst erfolgt die Frage nach den Directorypuffern:

Setting up Blocking/Deblocking buffers:

The physical record size is 0400H:

Available space in 256 byte pages:

TPA = 00F5H, Bank 0 = 0055H, Other banks = 0000H

Number of directory buffers for drive A: (#1) ?

Directorypuffer werden immer in der Bank 0 angelegt. Der Platz in den anderen Banks ist hier nur insofern von Bedeutung, daß auch die Datenpuffer in die Bank 0 müssen, falls in den anderen Banks kein Platz mehr ist. Es sind deshalb hier 2 Fälle zu unterscheiden:

- In den anderen Banks ist noch Platz:

In diesem Fall sollte der gesamte in der Bank 0 zur Verfügung stehende Platz für Directorypuffer genutzt werden. Dabei ist jedoch zu beachten, daß das System noch Platz für einige Zeiger benötigt und deshalb weniger zur Verfügung steht, als angezeigt wird! Die hier einzugebene Zahl wird wie folgt berechnet:

$$(\text{verfügbarer Platz}) / 4 - 1$$

Es ist nur der ganzzahlige Teil einzugeben. Vor der einzugebenen Zahl ist ein '#' zu schreiben, falls die Zahl dezimal eingegeben werden soll!

Im obigen Beispiel ist der verfügbare Platz in der Bank 0 = 55H (in Einheiten zu 256Byte). Die nach der obigen Formel einzugebende Zahl ist gleich 14H (in Einheiten zu 1024Byte) ( $55H/4H=15H$  und davon noch 1 abziehen).

In einigen Fällen kann es vorkommen, daß mehr als 1 abgezogen werden muß (siehe komplettes Installationsbeispiel auf den folgenden Seiten).

- In den anderen Banks ist kein Platz mehr:

Es sollte so viel Pufferplatz vereinbart werden, daß mindestens eine Directory darin Platz findet. Eine Floppy-Directory benötigt ca. 4 Pufferplätze, wenn sie voll ist. Eine Winchester-Directory benötigt 128 Pufferplätze, wenn sie voll ist. Es ist deshalb nie genug Platz vorhanden, um eine Winchester-Directory komplett unterzubringen. Die einzugebene Zahl ist deshalb stark von dem insgesamt verfügbaren Platz in der Bank 0 abhängig. Für jeden Directorypuffer wird der verfügbare Platz (Available space) um 4 Einheiten zu 256Byte kleiner. Da noch Platz für Datenpuffer benötigt wird, sollte deshalb im allgemeinen hier

ein Wert zwischen 1 und 4 eingegeben werden. Dabei ist davon auszugehen, daß es günstiger ist, mehr Datenpuffer als Directorypuffer im System zu haben.

Anschließend erfolgt die Abfrage nach den Datenpuffern:

Available space in 256 byte pages:

TPA = 00F5H, Bank 0 = 0044H, Other banks = 0000H

Number of data buffers for drive A: (#1) ?

Hier ist analog zu den Directorypuffern vorzugehen. Falls in den anderen Banks (Other banks) noch Platz ist, sollte dieser Platz vollständig für Datenpuffer genutzt werden. Berechnungsformel:

$$(\text{verfügbarer Platz}) / 4$$

Falls in den anderen Banks kein Platz mehr ist, muß der noch verfügbare Platz in der Bank 0 genutzt werden. Dabei ist jedoch wieder zu beachten, daß eventuell vom Betriebssystem noch Platz für Zeiger benötigt wird. Berechnungsformel:

$$(\text{verfügbarer Platz}) / 4 - 1$$

Leider wird der verfügbare Platz von GENCPM nur in HEX angegeben. Wer nicht in HEX rechnen kann, sollte diese Werte erst in dezimal umrechnen. Berechnungsformel:

$$(\text{letzte Ziffer}) + (\text{vorletzte Ziffer} * 16) + (\text{Ziffer davor} * 256)$$

Dezimaleingaben müssen bei GENCPM mit einem führenden '#' versehen werden!

Sollte zum Schluß von GENCPM in irgendeiner Bank noch mehr als 3 Pages zu 256Bytes frei sein, sollte GENCPM noch einmal durch Eingabe von:

**GENCPM A**

neu gestartet werden. Die Anzahl der Puffer kann dann etwas größer gewählt werden.

Sollte in irgendeiner Bank ein Pufferüberlauf stattfinden (noch verfügbarer Platz: 00Fx statt 000x) muß GENCPM ebenfalls noch einmal neu gestartet werden (**GENCPM A**). Die Anzahl der Puffer in der Bank 0 muß dann um 1 reduziert werden.

Bei einem Eingabefehler (z.B. zu viele Puffer eingegeben) sollte **GENCPM** durch Eingabe von **ctrl C** abgebrochen und durch Eingabe von **'GENCPM A'** erneut gestartet werden. GENCPM fragt sonst auch das ab, was in den anderen Fällen automatisch beantwortet wird.

Falls nicht genug Speicherplatz vorhanden ist, um mindestens einen Datenpuffer und mindestens einen Directorypuffer zu selektieren, kann das so zu installierende System nicht generiert werden. Es ist dann notwendig, CPMINST erneut zu starten und entweder einige Laufwerke wegzulassen oder eine RAM-Bank mehr für das Betriebssystem zu vereinbaren. Alternativ ist es auch möglich 'GENCPM' ohne 'A'-Parameter zu starten und alle Fragen zu beantworten. Dabei können dann einige Hash-Tabellen weggelassen werden, welches dann zu mehr verfügbaren Datenpuffern führt (Siehe CP/M PLUS System Guide).

Es folgt nun ein Beispiel für eine etwas umfangreichere Systemgenerierung (Hardware: CPU III mit 256KByte RAM, FDC II mit 64KByte RAM, FDC 5, HDC 5, VIDEO I, P/SIO, 2x5,25"-Laufwerke, 2x3,5"-Laufwerke, 2x8"-Laufwerke, 1xWinchester, EPROM-Disk, CMOS-DISK, RAMDISK, 2MByte SMB-DISK):

Diskette #1 in Laufwerk A einlegen (das ist noch das Laufwerk 0 an der FDC II)!

Das System bootet über die VIDEO I automatisch. Falls Sie keine VIDEO I im System hätten, müssten Sie das System durch Eingabe eines Leerzeichens an der Tastatur des Terminals an SIO2 der P/SIO-Platine booten. Der Urlader meldet diverse gefundene Werte Ihres Systems. Nach vielen Zeilen Meldungen kommt:

A>

Rufen Sie dann FORMAT auf:

**Alle unterstrichenen Zeichen sind Eingaben durch den Anwender!**

A>FORMAT

Das Formatierprogramm meldet sich.

Geben Sie den Namen Ihres zweiten Laufwerks an, hier:

B

Wählen Sie ein zu Ihrem Laufwerk passendes Format aus, hier:

1

Stecken Sie eine neue Diskette in Ihr zweites Laufwerk, hier B. Geben Sie danach 'j' oder 'y' ein; die Diskette wird jetzt formatiert. Die Spuren werden dabei mitgezählt. Nach dem Ende geben Sie anstelle eines 'j' oder 'y' ein 'n' ein. Dann erscheint die Frage: Zurück zu CP/M. Hier geben Sie ein CARRIAGE RETURN ein. Sie sind dann wieder im CP/M. Rufen Sie nun PIP auf:

A>PIP

PIP meldet sich mit:

\*

Stecken Sie nun die Diskette #3 in Ihr erstes Laufwerk (hier A) und geben Sie folgendes Kommando ein:

\*b:=a:\*.\*[v]

Jetzt kopiert PIP ihre Diskette #3. Danach erscheint:

\*

Geben Sie jetzt CARRIAGE RETURN ein. Sie sind jetzt wieder im CP/M.

A>

Lassen Sie die soeben erstellte Kopie in ihrem 2. Laufwerk (hier B) und legen sie die Diskette 1 in Ihr 1. Laufwerk (hier A). Starten Sie die Installierungsprozedur durch Eingabe von:

A>CPMINST

Sie sind jetzt im INSTALL-Programm:

...

...

... CPMINST ist menuegesteuert. Auf den Abdruck der einzelnen  
 ... Menuepunkte wird hier verzichtet, da sich das Bild dieser  
 ... Menues in Abhängigkeit der unterstützten Hardware stark  
 ... abändern kann.

...

... Nach Auswahl aller Hardwarekomponenten startet CPMINST  
 ... automatisch die folgende Installationsprozedur:

...

START DER SYSTEMGENERIERUNG: SUBMIT CPMGEN

B>SUBMIT CPMGEN B A

B>; CPMGEN.DAT 1.600

B>rmac chario \$pz-s

CP/M RMAC ASSEM 1.1

0300

03BH USE FACTOR

END OF ASSEMBLY

B>rmac disks \$pz-s

CP/M RMAC ASSEM 1.1

0400

0B8H USE FACTOR

END OF ASSEMBLY

B>rmac user \$pz-s

CP/M RMAC ASSEM 1.1

0015

000H USE FACTOR

END OF ASSEMBLY

```
B>rmac printer $pz-s
CP/M RMAC ASSEM 1.1
0001
002H USE FACTOR
END OF ASSEMBLY
```

```
B>pip B:bnkbdos3.spr=A:bnkbdos3.spr [v
```

```
B>pip B:resbdos3.spr=A:resbdos3.spr [v
```

```
B>link bnkbios3[b,nr]=bioskern,scb,boot,chario,disks,sysex, futil,ddata,user,&
LINK 1.31
```

```
*printer,&
*cpu3move,&
*cpu3time,&
*fdc5card,&
*fdc2card,&
*hdc5card,&
*DATECOMP
```

```
XXLST 06D6 $ADRV 07FB $RDRV 07FC $TRK 07FD
$SECT 07FF $DMA 0801 $DBNK 0804 $CNT 0803
```

```
...
```

```
... (Im Original ist das Listing hier etwas länger.)
```

```
...
```

```
DAY 3179 MONTH 3178 YEAR 3177 YMD2D 3A07
HDC51D 35F3 SFLAG 3809
```

```
ABSOLUTE 0000
CODE SIZE 04C7 (0000-04C6)
DATA SIZE 3564 (0500-3A63)
COMMON SIZE 0000
USE FACTOR 61
```

```
B>;WARNING: DON'T CHANGE COMMON BASE AND NUMBER OF CP/M SEGMENTS
```

```
B>GENCPM A
```

```
CP/M 3.0 System Generation
Copyright (C) 1982, Digital Research
```

```
Create a new GENCPM.DAT file (N) ?
```

```
Display Load Map at Cold Boot (Y) ?
```

```
Number of console columns (#79) ?
```

```
Number of lines in console page (#24) ?
```

```
Backspace echoes erased character (N) ?
```

```
Rubout echoes erased character (N) ?
```

```
Initial default drive (A:) ?
```

```
Top page of memory (FE) ?
```

```
Bank switched memory (Y) ?
```

```
Common memory base page (E0) ?
```

Long error messages (Y) ?

Accept new system definition (Y) ?

Setting up Allocation vector for drive A:  
 Setting up Allocation vector for drive B:  
 Setting up Checksum vector for drive B:  
 Setting up Allocation vector for drive C:  
 Setting up Checksum vector for drive C:  
 Setting up Allocation vector for drive D:  
 Setting up Checksum vector for drive D:  
 Setting up Allocation vector for drive E:  
 Setting up Checksum vector for drive E:  
 Setting up Allocation vector for drive F:  
 Setting up Allocation vector for drive G:  
 Setting up Allocation vector for drive H:  
 Setting up Allocation vector for drive I:  
 Setting up Allocation vector for drive K:  
 Setting up Checksum vector for drive K:  
 Setting up Allocation vector for drive L:  
 Setting up Checksum vector for drive L:

\*\*\* Bank 1 and Common are not included \*\*\*  
 \*\*\* in the memory segment table. \*\*\*

Number of memory segments (#3) ?

CP/M 3 Base,size,bank (71,6F,00)

Enter memory segment table:

Base,size,bank (00,F0,00) ?

ERROR: Memory conflict - segment trimmed.

Base,size,bank (00,E0,00) ?

ERROR: Memory conflict - segment trimmed.

Base,size,bank (00,71,00) ?

Base,size,bank (00,F0,02) ?

ERROR: Memory conflict - segment trimmed.

Base,size,bank (00,E0,02) ?

Base,size,bank (00,F0,03) ?

ERROR: Memory conflict - segment trimmed.

Base,size,bank (00,E0,03) ?

CP/M 3 Sys 7100H 6F00H Bank 00

Memseg No. 00 0000H 7100H Bank 00

Memseg No. 01 0000H E000H Bank 02

Memseg No. 02 0000H E000H Bank 03

Accept new memory segment table entries (Y) ?

Setting up directory hash tables:

Enable hashing for drive B: (Y) ?

Enable hashing for drive C: (Y) ?

Enable hashing for drive D: (Y) ?  
 Enable hashing for drive E: (Y) ?  
 Enable hashing for drive H: (Y) ?  
 Enable hashing for drive K: (Y) ?  
 Enable hashing for drive L: (Y) ?

Setting up Blocking/Deblocking buffers:

The physical record size is 0400H:

Available space in 256 byte pages:  
 TPA = 00F4H, Bank 0 = 0071H, Other banks = 0170H

Number of directory buffers for drive B: (#1) ? 1b

Available space in 256 byte pages:  
 TPA = 00F4H, Bank 0 = 0003H, Other banks = 0170H

Number of data buffers for drive B: (#1) ? 5c  
 Allocate buffers outside of Common (Y) ?

...

... (Hier erfolgt im Original noch eine Liste von weiteren Meldungen.)

...

Available space in 256 byte pages:  
 TPA = 00F4H, Bank 0 = 00FEH, Other banks = 0000H

...

^

...

!

... An dieser Stelle darf zum Schluß nur 0000H, 0001H, 0002H  
 ... oder 0003H stehen! 00FEH bedeutet, daß von GENCPM nach der  
 ... Verteilung aller Puffer noch weitere interne Puffer ver-  
 ... geben wurden. Leider überprüft GENCPM an dieser Stelle nicht  
 ... mehr, ob überhaupt noch Platz vorhanden ist (Fehler in GENCPM).  
 ... Ein so installiertes Betriebssystem führt garantiert zu einem  
 ... Systemabsturz!  
 ... Ein erneuter Lauf von GENCPM ist hier notwendig! Dabei muß ein  
 ... Directory-Puffer weniger gewählt werden.

...

Accept new buffer definitions (Y) ?

BNKBIOS3 SPR FA00H 0500H  
 BNKBIOS3 SPR 9800H 4800H  
 RESBDOS3 SPR F400H 0600H  
 BNKBDOS3 SPR 6A00H 2E00H

\*\*\* CP/M 3.0 SYSTEM GENERATION DONE \*\*\*

B>

B>gencpm a <--- Erneuter Start von GENCPM

CP/M 3.0 System Generation  
 Copyright (C) 1982, Digital Research

```

...
... (An dieser Stelle erfolgen im Original die gleichen umfangreichen
... Meldungen wie oben.)
...

```

Setting up Blocking/Deblocking buffers:

The physical record size is 0400H:

```

Available space in 256 byte pages:
TPA = 00F4H, Bank 0 = 0071H, Other banks = 0170H

```

Number of directory buffers for drive B: (#1) ? 1A

```

Available space in 256 byte pages:
TPA = 00F4H, Bank 0 = 0007H, Other banks = 0170H

```

Number of data buffers for drive B: (#1) ? 5C  
Allocate buffers outside of Common (Y) ?

```

...
... (Hier erfolgt im Original noch eine Liste von weiteren Meldungen.)
...

```

```

Available space in 256 byte pages:
TPA = 00F4H, Bank 0 = 0002H, Other banks = 0000H
...
...
...
...
...

```

Accept new buffer definitions (Y) ?

```

BNKBIOS3 SPR FA00H 0500H
BNKBIOS3 SPR 9800H 4800H
RESBDOS3 SPR F400H 0600H
BNKBDOS3 SPR 6A00H 2E00H

```

\*\*\* CP/M 3.0 SYSTEM GENERATION DONE \*\*\*

B>

Jetzt ist Ihr angepaßtes CP/M PLUS fertig. Die auf der Diskette #3 befindliche Datei 'CPM3.SYS' enthält nun das Betriebssystem. Jetzt müssen Sie noch die Datei 'CCP.COM' und bei einem EPC II auch die Datei 'CHARGEN.COM' auf die Diskette 3 kopieren. Betätigen Sie danach RESET und starten Sie Ihr System neu. Jetzt muß sich Ihr System mit Ihrer Kennung melden, falls Sie nicht im Install-Programm die Kennungsmeldungen abgeschaltet haben. Ab jetzt ist die soeben erstellte Diskette Ihre Systemdiskette. Machen Sie eine Kopie! Arbeiten Sie nie auf Originaldisketten! Der Aufruf von GENCPM zerstört das bisher auf der Diskette befindliche CP/M-Systemfile (CPM3.SYS).

Als Erstellungsdatum hat CPMINST das augenblickliche Datum in das Betriebssystem eingetragen. In der Auslieferversion des Betriebs-



systems ist jedoch evtl. kein aktives Datum/Zeit-Modul vorhanden, so daß hier als Erstellungsdatum das ursprüngliche Erstellungsdatum dieser CP/M PLUS-Version oder ein Zufallswert eingetragen wurde. Bei zukünftigen Neuinstallationen durch den Anwender unter dem augenblicklich installierten Betriebssystem wird an dieser Stelle jedoch immer das aktuelle Datum eingetragen werden (falls es vorher gesetzt wurde!).

Falls Sie sich nun weitere bootfähige Disketten erstellen wollen, müssen Sie jeweils die Dateien 'CPM3.SYS', 'CCP.COM' und beim EPC II auch 'CHARGEN.COM' auf diese Systemdisketten kopieren.

Bei einer CPU 180x-Platine befindet sich der CCP im Boot-Eprom und braucht deshalb nicht auf der Systemdiskette vorhanden zu sein! Zur Erstellung weiterer bootfähiger Disketten genügt es hier deshalb nur die Datei 'CPM3.SYS' auf diese Systemdisketten zu kopieren.

2.2 Disk-Manager

DISK MANAGER (DM) ist ein Hilfsprogramm zum Arbeiten mit Disketten. DM erlaubt es, aktive Diskettenformate zu vereinbaren, neue Diskettenformate hinzuzufügen und fremde Disketten zu analysieren. Ferner ist in DM ein einfacher Disk-Monitor enthalten.

DM kann in 2 Betriebsarten benutzt werden:

1. interaktiv (dies ist die normale Anwendung)
2. im Batch (nur zur Auswahl der Fremdformate)  
 ==> unter 'Formate auswählen' (DM Funktion 3) beschrieben

In der interaktiven Betriebsart ist DM ein menuegesteuertes Programm. Es meldet sich mit der folgenden Funktionsauswahl:

DISK-MANAGER vers. 1.602

Copyright (c) 1984 by M.K.C

```

<1>   Disk-Monitor-Programm aufrufen
<2>   fremde Diskette analysieren
<3>   Auswahl von Formaten treffen
<3?>  Auswahl von Formaten treffen (mit Liste)
<4>   Editieren eines Fremdformates
<4>   Editieren eines Fremdformates (mit Liste)
      (Datei: DISKFORM.DAT)
<5>   Formate ausdrucken
<9>   zurueck zu CP/M

```

wahle Funktion: \_

Jetzt ist eine der im Menue angegebenen Zahlen und ein CR einzugeben. Dann erscheint die Meldung:

gewaehlte Funktion: hier steht die gewählte Funktion

Disk-Monitor (DM Funktion 1)

Dc Laufwerk "c" selektieren, löscht Spur und Sektor  
 Tn Spur "n" selektieren (für spätere Schreib/Leseoperationen)  
 Sn Sektor "n" anwählen (für spätere Schreib/Leseoperationen)  
 + nächsten Sektor selektieren (spurübergreifend)  
 - vorigen Sektor selektieren (spurübergreifend)  
 Gh Block "h" selektieren (ersetzt Spur und Sektor)

R selektierten Sektor in den Puffer lesen  
 W Puffer in den selektierten Sektor schreiben  
 Y selektierten Sektor mit Puffer vergleichen  
 Format: h h h CR Fehler quittieren, weiter  
           h h h . Fehler quittieren, abrechnen  
           (Adresse, Pufferinhalt, Sektorinhalt)

P Puffer drucken  
 Format: h:= hh hh hh hh hh hh hh hh cccccccccccccccc  
 im Page-Mode ist nach jeweils 8 Zeilen CR einzugeben

Fh gesamten Puffer mit "h" füllen  
 Ch Daten ab Adresse "h" im Puffer verändern  
 Format: h := h h ändern  
           h := h CR nicht ändern, nächster Wert  
           h := h . beenden  
           (Adresse, alter Inhalt, neuer Inhalt)

Mn Belegungsliste des Laufwerks ab Spur "n" ausgeben  
 Format: Spur n: cccc...cccc mit: - = leer, \* = be-  
           legt, ? = defekt, + = leer (invers),

L Translation Table ausgeben / ändern  
 Format: n:= n n ändern  
           n:= n CR nicht ändern, nächster Sektor  
           n:= n . abrechnen  
           (XLT-Adresse, alter Inhalt, neuer Inhalt)

V CP/M Disk Parameter ausgeben (siehe CP/M SYSTEM GUIDE)

. page mode einschalten (seitenweise Ausgabe)  
 , page mode ausschalten (kontinuierliche Ausgabe)  
 A gewählten Sektor sofort lesen und ausgeben  
 O Sektor nur selektieren (Lesen und Ausdrucken nur auf  
 Befehl)

I invertierendes Schreiben/Lesen selektieren  
 J nichtinvertierendes Lesen/Schreiben selektieren

X Disk-Monitor verlassen  
 ? Hilfstexte (Befehlsübersicht) ausgeben

Hier steht h für eine hexadezimale, n für eine dezimale Ein- oder Ausgabe und c für einen Buchstaben.

Ein . als Prompt zeigt die fehlerfreie, ein ? eine fehlerhafte Ausführung des letzten Kommandos an. Ein ? wird auch bei falschen (fehlerhaften) Kommandos benutzt.

Disketten analysieren (DM Funktion 2)

Nach der Eingabe des Laufwerks wird versucht, die fremde Diskette zu analysieren. Die Ergebnisse werden wie folgt dargestellt:

8"/5": 5" = ~~50~~, 8" = +8 High density = H  
 FM/MFM: FM = 0, MFM = 40, CO bedeutet: außen FM, innen MFM  
 N: 0 = 128, 1 = 256, 2 = 512, 3 = 1024 Byte / Sektor  
 Kopf1: head-id der Vorderseite  
 Kopf2: head-id der Rückseite  
 Spur0: track-id der äußersten Spur  
 Seiten: 1 = einseitig, 2 = zweiseitig  
 Spuren: 40, 77 oder 80 Spuren / Seite  
 Sektoren: Anzahl der Sektoren pro Spur  
 SKEW: Sektor-skew beim Formatieren (d.h. nicht fortlaufende Sektoren beim Formatieren)  
 FLAG0: 0: Startsektor = 1, 2: Startsektor = 0

Jetzt sind im Disk-Monitor die Werte: OFFSET, Anzahl der Directory-Sektoren, CP/M-Blockgröße, Übergang auf Rückseite und die XLT zu bestimmen.

Das so analysierte Format ist unter der Nummer 254 erreichbar, es sollte jetzt editiert (dabei überprüft) und dann unter einer anderen Nummer gespeichert werden. Nach der Aktivierung dieses neuen Formates kann mit dieser Diskette gearbeitet werden.

Formate auswählen (DM Funktion 3)

Im CBIOS v.1.6 ist Platz für bis zu 9 Fremdformate. Aus der Liste der bekannten Formate können also jeweils 9 aktiviert sein. Das System erkennt dann diese fremden Disketten automatisch. 3 übergeht die Liste der bekannten Formate und fragt gleich nach den Änderungen. Die Liste wird durch Eingabe von 3? vor der Frage nach den Veränderungen dargestellt.

Diese Funktion kann auch im Batch-Mode erreicht werden; Eingabe hierzu: DM (F+f-f+f....). Hierbei stehen die runden für eckige Klammern und f für die Nummer eines Formates, + für aktivieren und - für deaktivieren; -\* löscht alle aktiven Formate. Im Batch-Betrieb werden keinerlei Fehlermeldungen ausgegeben.

Beispiel: DM ÄF-1-2+26Ü löscht die Formate 1,2 und aktiviert 26.  
 (Ä und Ü sind eckige Klammern)

Formate editieren (DM Funktion 4)

Ein Punkt . beendet das Editieren, ein CR läßt den alten Wert unverändert. Vierstellige Zahlen sind hexadezimal, fünfstellige dezimal. Bei den Vergleichswerten Sektorgröße, Seiten, Spuren, und Sektoren bewirkt ein gesetztes höchstes Bit, daß dieser Wert für den Vergleich nicht berücksichtigt wird.

Seiten = 3 steht für eine zweiseitige Diskette, bei der Vorder- und Rückseite einer Spur als getrennte, nacheinanderliegende Spuren betrachtet werden. Diese Version ist immer dann erforderlich, wenn das Inhaltsverzeichnis auf der Rückseite beginnt, oder Vorder- und Rückseite zusammen mehr als 52 (physikalische) Sektoren enthalten.

Der Wert FLAG0 faßt folgende Bedingungen zusammen:

- Bit 0: normal (Bit 0 = 0) oder invers (Bit 0 = 1)
- Bit 1: Sektor 1 (Bit 1 = 0) oder Sektor 0 (Bit 1 = 1) als erster
- Bit 2: Kopf weiterzählen (Bit 2 = 0) oder gleichlassen (Bit 2 = 1) bei Übergang auf die Rückseite
- Bit 3: auf der Rückseite von außen nach innen (Bit 3 = 0) oder von innen nach außen (Bit 3 = 1) steppen
- Bit 4: für ALPHATRONIC P4 ist Bit 4 = 1

Der Wert FLAG1 faßt folgende Bedingungen zusammen:

- Bit 0: internes (1) oder normales (0) Format
- Bit 1..6: Nummer des zweiten Kopfes, falls FLAG0 bit 2 = 1 ist

### Formate ausdrucken (DM Funktion 5)

Hier wird eine Liste der in der DISKFORM.DAT enthaltenen Formate ausgegeben. Ein P veranlaßt die Ausgabe zum Drucker, ein X läßt die XLTs mitausgeben. Format der Ausgabe:

- Nr ist die Nummer des Formates (dezimal)
  - Name ist der Name in der DISKFORM.DAT
  - " ist die Größe der Diskette (5 ~~oder 8"~~ oder *High density*)
  - D ist die Dichte Double density, Single density oder Changing density
  - F ist die Anzahl der Seiten
  - Sp ist die Anzahl der Spuren (dezimal)
  - Sk ist die Anzahl der Sektoren pro Spur (dezimal)
  - IF ist das FLAG0 (vgl. letzte Seite) (hexadezimal)
  - SPT (SPT) ist sector per track (dezimal)
  - B (BSH) ist block-shift-factor (hexadezimal)
  - B (BLM) ist block-mask (hexadezimal)
  - E (EXM) ist extent-mask (hexadezimal)
  - DSM (DSM) ist Kapazität in Blocks - 1 (dezimal)
  - DRM (DRM) ist Anzahl der Directory-Einträge - 1 (dezimal)
  - AL01 (ALO, AL1) kennzeichnet die reservierten Blocks für das Directory (hexadezimal)
  - CS (CKS) ist checksumsize (dezimal)
  - O (OFF) ist offset (hexadezimal)
  - P (PSH) ist physical sector size (? = Fehler?) (hexadezimal)
  - P (PHM) ist physical sector mask (hexadezimal)
- anschließend folgt in den nächsten Zeilen die XLT (dezimal)

### Einführung

Das hier folgende Kapitel enthält allgemeine Erklärungen zu Disketten, Festplatten und den für das CBIOS notwendigen Parametertabellen. Es ist als kurze Einführung für das sinnvolle Arbeiten im DM gedacht.

Dieses Kapitel kann von erfahrenen CP/M-Anwendern übersprungen werden. Beim Lesen dieses Kapitels (und beim Analysieren von Disketten) sollten Sie den CP/M PLUS SYSTEM GUIDE zur Hand haben.

Jede Diskette muß vor der Benutzung in dem jeweiligen Computer formatiert werden. Danach hat sie eine systemabhängige Aufteilung. Diese ist teils durch die verwendete Hardware (Floppy-Controller und Laufwerk) und teils durch das verwendete Betriebssystem bedingt. Wir wollen hier nur CP/M-Disketten betrachten. Alle CP/M-Versionen (CP/M 1.4, CP/M 2.2, CP/M PLUS, CP/M 86, CP/M 68K, CCP/M und MP/M) haben im wesentlichen kompatible Filesysteme. Somit brauchen wir die unterschiedlichen CP/M-Versionen nicht zu betrachten.

Jetzt existieren noch die Unterschiede in der verwendeten Hardware:

- Disketten können 3", 3,5", 5,25" oder 8" groß sein
- Disketten können ein- oder zweiseitig oder als "Wendendisketten" benutzt sein
- Disketten können soft- oder hard-sektoriert sein. Eine softsektorierte Diskette hat ein Indexloch; eine hardsektorierte Diskette hat zusätzlich vor jedem Sektor noch ein Loch
- Disketten können unterschiedlich Spurdichten haben: 48 tpi, 96 tpi, 100 tpi und 135 tpi sind übliche Werte
- bei den Aufzeichnungsformaten wollen wir nur FM (single density) und MFM (double density) betrachten. Somit entfallen alle Disketten von APPLE II Rechnern.
- Disketten rotieren mit 300 oder 360 U/min (oder auch mit spurabhängiger variabler Geschwindigkeit)

Hieran kann man schon leicht sehen, daß es fast beliebig viele nicht verträgliche Diskettenversionen gibt. Wenn wir jetzt noch die im CP/M über Parameter steuerbaren Einteilungen (wie z.B. Sektorgröße, Anzahl der Sektoren pro Spur, Größe des Inhaltsverzeichnisses usw.) mitberücksichtigen, sind weit über 1000 unterschiedliche "Diskettenformate" möglich.

Wir setzen jetzt erst einmal voraus, daß alle zu betrachtenden Disketten kompatibel zu der verwendeten Hardware sind; d.h. es entfallen schon einmal eine ganze Menge an exotischen Formaten.

Somit sind jetzt "nur noch" die im CP/M möglichen Steuerparameter zu betrachten (Bezug: CBIOS 1.6).

Für jedes Diskettenformat existiert eine Tabelle der Steuerparameter; evtl. muß diese durch eine Analyse der Diskette selbst erstellt werden:

interne CBIOS 1.6 Werte: (diese Werte können automatisch erkannt werden, sie werden zur Formaterkennung im CBIOS 1.6 benutzt.)

Laufwerksgröße: 5,25" oder 8"  
 Aufzeichnungsdichte: FM, MFM oder außen FM und innen MFM  
 Sektorgröße: 128, 256, 512 oder 1024 Byte/Sektor  
 Nummer der äußersten Spur: 0...?  
 Nummer des Kopf "0": 0 oder ?  
 Anzahl der Spuren: 35, 39, 40, 77 oder 80 Spuren  
 Anzahl der Sektoren: 4 bis 26 Sektoren sind abhängig von der Sektorgröße und der Aufzeichnungsdichte möglich  
 erste Sektornummer: meist 1, aber auch 0 möglich

Jetzt kommen die Tabellen für CP/M:

Allgemeines:

CP/M teilt eine Diskette in zwei Bereiche, die Systemspuren (ganz außen) und den verwalteten Bereich (innen) auf. Am Anfang des verwalteten Bereichs liegt das Inhaltsverzeichnis. Im Inhaltsverzeichnis stehen Dateinamen und Zeiger (Bytes oder 16 bit Worte) auf die von der Datei belegten Blöcke. Diese Blöcke können im CP/M 1, 2, 4, 8 oder 16 KByte groß sein. Ein Block ist die kleinste Einheit einer Diskette, die im Inhaltsverzeichnis verwaltet werden kann. Die kleinste adressierbare Einheit einer Diskette ist ein physikalischer Sektor. Ferner werden von CP/M im Inhaltsverzeichnis Gruppen a 16KByte Länge verwendet. Eine oder mehrere dieser Gruppen werden in einem Eintrag im Inhaltsverzeichnis zusammengefaßt.

DISK PARAMETER BLOCK:

SPT logische (128-Byte) Sektoren pro Spur  
 hier kann entweder nur eine Seite oder Vorder- und Rückseite einer Spur gemeint sein)

BSH Blockgröße (block shift factor)

BLM Blockgröße (block mask)

BSH und BLM sind voneinander abhängig. Inkonsequente Werte führen zum Systemabsturz!

1 K Blockgröße:	BSH = 3	BLM = 07	EXM = 0	-
2 K Blockgröße:	BSH = 4	BLM = 0F	EXM = 1	0
4 K Blockgröße:	BSH = 5	BLM = 1F	EXM = 3	1
8 K Blockgröße:	BSH = 6	BLM = 3F	EXM = 7	3
16 K Blockgröße:	BSH = 7	BLM = 7F	EXM = F	7

EXM Anzahl der Gruppen pro Eintrag - 1  
 Dieser Wert ist von der Blockgröße und der Kapazität der Diskette (bzw. der Länge der Zeiger) abhängig. Der erste Wert gilt für 8 Bit lange Zeiger, der zweite für 16 Bit Zeiger. 8 Bit lange Zeiger werden immer dann verwendet, wenn die Kapazität der Diskette weniger als 256 Blocks beträgt. Bei größeren Disketten müssen 16 Bit lange Zeiger verwendet werden. In der Tabelle sind die Maximalwerte für EXM angegeben. Manche Hersteller verwenden jedoch kleinere Werte für EXM, obwohl dies keine Vorteile bringt.

DSM	Kapazität der Diskette in Blocks - 1 Hier wird nur die Kapazität des vom CP/M verwalteten Bereichs der Diskette angegeben. Die Systemspuren werden nicht berücksichtigt.	
DRM	Anzahl der Einträge im Inhaltsverzeichnis - 1 Jeder Eintrag im Inhaltsverzeichnis belegt 32 Byte	
ALO	reservierte Blocks für das Inhaltsverzeichnis ALO und AL1 stellen zusammengefaßt einen Bitvektor dar, in dem ein Bit für jeden reservierten Block an ist. Das höchste Bit in ALO steht für den Block 0, das niedrigste Bit in AL1 für den Block 15. Übliche Werte sind: 80h, C0h, E0h und F0h für 1, 2, 3 oder 4 reservierte Blocks.	
AL1		
CKS	Länge eines Prüfvektors für das Inhaltsverzeichnis. CP/M findet eine getauschte (neue) Diskette, in dem es bei jedem Zugriff auf das Inhaltsverzeichnis eine Prüfsumme des jeweiligen Sektors berechnet und diese mit einer gemerkten vergleicht. Sind die beiden Prüfsummen unterschiedlich, muß die Diskette gewechselt worden sein. Somit wird ein Byte pro logischen Sektor (also pro 4 Einträge) benötigt.	
OFF	ist die Anzahl der reservierten Systemspuren und gibt gleichzeitig die (Anfangs-) Spur des Inhaltsverzeichnisses an. Ein häufig verwendeter Wert ist 2.	
PSH	physikalische Sektorgröße (physical record shift factor)	
PHM	physikalische Sektorgröße (physical record mask)	
	PSH und PHM sind voneinander abhängig. Inkonsequente Werte führen zum Systemabsturz!	
	128 Byte Sektorgröße:	PSH = 0    PHM = 0
	256 Byte Sektorgröße:	PSH = 1    PHM = 1
	512 Byte Sektorgröße:	PSH = 2    PHM = 3
	1024 Byte Sektorgröße:	PSH = 3    PHM = 7

Viele der oben genannten Werte kann das Betriebssystem mit einer hohen Wahrscheinlichkeit bei der Analyse einer Diskette selbst berechnen. Hilfe vom Menschen wird nur benötigt, um die Lage des Inhaltsverzeichnisses, die verwendete Blockgröße und die Anzahl der Einträge zu bestimmen. Bei zweiseitigen Disketten ist zusätzlich Hilfe erforderlich, um den Zugriff auf die zweite Seite zu definieren. (Spurüberlauf, d.h. nach der innersten Spur der ersten Seite kommt die zweite Seite, oder Sektorüberlauf, d.h. nach dem letzten Sektor der Vorderseite kommt die Rückseite der gleichen Spur)

Zur Geschwindigkeitssteigerung verwendet CP/M eine Sektorverschränkung. Hierbei werden nicht zwei aufeinander folgende sondern z.B. jeder vierte Sektor gelesen. Dadurch hat die Software 3 Sektoren Zeit um den soeben gelesenen zu verarbeiten, bis der nächste Lesezugriff erfolgt. Diese Geschwindigkeitsoptimierung kann auch direkt beim Formatieren erfolgen, indem z.B. die Sektoren auf einer Spur wie folgt nummeriert werden:

1, 5, 2, 6, 3, 7, 4, 8



Hierbei wird die Sektorverschränkung im CP/M i.allg. nicht benutzt, d.h. die Sektorverschränkungstabelle des CBIOS sieht wie folgt aus:

1, 2, 3, 4, 5, 6, 7, 8

Die meisten Rechnersysteme formatieren aber ohne Verschrängung:

1, 2, 3, 4, 5, 6, 7, 8

Dann wird eine Sektorverschränkungstabelle im CBIOS benötigt, die für das obige Beispiel wie folgt aussieht:

1, 3, 5, 7, 2, 4, 6, 8

Natürlich können auch mehrere Sektoren auf einmal übersprungen werden, oder es können die beiden Verfahren gemeinsam benutzt werden.

Logischerweise hat das Betriebssystem keine Chance, eine Sektorverschränkungstabelle aufzubauen. Es nimmt daher vereinfachend an, daß die Tabelle: 1, 2, 3, 4,....., 50, 51, 52 gilt. Hier ist der Benutzer gefordert, die richtige Sektorverschränkungstabelle durch Probieren zu finden. Dies geht am einfachsten, wenn ein langer sinnvoller Text zur Verfügung steht.

**Funktion: 1 -- Disk Monitor**

Mit **Funktion 1** wird der eingebaute Disk-Monitor aufgerufen. Der Disk-Monitor ermöglicht es Ihnen, jedes einzelne Byte einer Diskette zu verändern. Somit ist er mit besonderer Vorsicht zu benutzen. Es gibt keine andere Möglichkeit, eine Diskette so schnell für CP/M unbrauchbar zu machen. Um Bytes sinnvoll zu verändern, ist eine genaue Kenntnis des CP/M File-Systems erforderlich. DM ist für jedes im CP/M vorhandene Laufwerk benutzbar, auch für Festplatten und RAM-Disks. Bei einer Festplatte ist zu beachten, daß der DISK-MONITOR mit den BIOS-Funktionen arbeitet. Somit stimmen die vom CP/M angelegten Puffer nach dem Schreiben im DISK-MONITOR leider nicht mehr. Zur Sicherheit sollte dann RESET betätigt werden.

Bei sinnvoller Nutzung stellt er jedoch ein wichtiges Hilfsmittel zu Analyse und Korrektur von Disketten dar.

Der Disk Monitor meldet sich mit einer Befehlsübersicht:

gewaehlte Funktion: DISK MONITOR

D I S K - M O N I T O R

Befehlsuebersicht:

Dc = Laufwerk "c" auswaehlen

Gh = Block "h" auswaehlen

Tn = Spur "n" auswaehlen

Sn = Sektor "n" auswaehlen

+ = Sektor + 1

- = Sektor - 1

R = Sektor lesen

W = Sektor schreiben

Mn = Belegung ab Spur "n" drucken

Y = Sektor vergleichen

P = Puffer drucken

Fh = Puffer mit "h" fuellen

Cn = Daten ab Adresse "n" aendern

( . zum Beenden)

X = zurueck zum Hauptmenue

L = Translation-Table eingeben

V = Disk Parameter ausgeben

. = page mode

, = no-page mode

I = invertierend

J = nicht invertierend

A = lesen und ausgeben

O = nur selektieren

? = dieses Menue anzeigen

Jetzt können Befehle laut obiger Übersicht eingegeben werden. Die eventuell erforderlichen Parameter sind ohne Leerzeichen direkt hinter dem Buchstaben für das Kommando einzugeben.

Ein . wird im Disk-Monitor als normaler Prompt verwendet. Ein ? als Prompt sagt aus, daß der letzte Befehl nicht in Ordnung war, es war entweder ein ungültiger Befehl oder er konnte nicht fehlerfrei ausgeführt werden.

Betriebsarten

Im Disk-Monitor sind mit den Steuerbefehlen . , A 0 verschiedene Betriebsarten wählbar:

**PAGE- oder NOPAGE-Mode**

Im PAGE-Mode wird die Ausgabe eines Sektors nach jeweils 8 Zeilen angehalten. Durch Eingabe von CR wird sie fortgesetzt. Diese Betriebsart kann auf drei Arten erreicht werden:

- im CP/M PLUS mit SETDEF (erste Voreinstellung)
- beim Aufruf durch Eingabe von **DM (PAGE)** eckige Klammern!  
(zweite Voreinstellung, übergeht die erste)
- im Disk-Monitor durch Eingabe von .  
(expliziter Befehl, Voreinstellungen werden nicht beachtet)

Im NOPAGE-Mode wird ein Sektor kontinuierlich ausgegeben. Die Ausgabe kann durch CNTL S angehalten, und kann danach durch Eingabe von CNTL Q fortgesetzt werden. Diese Betriebsart kann auf die folgenden drei Arten erreicht werden:

- im CP/M PLUS mit SETDEF (erste Voreinstellung)
- beim Aufruf durch Eingabe von **DM (NOPAGE)** eckige Klammern!  
(zweite Voreinstellung, übergeht die erste)
- im Disk-Monitor durch Eingabe von ,  
(expliziter Befehl, Voreinstellungen werden nicht beachtet)

**nur Anwahl oder auch Darstellung**

Es ist möglich, einen gewählten Sektor sofort ohne weitere Eingaben darstellen zu lassen. In dieser Betriebsart wird automatisch ein angewählter Sektor gelesen und ausgegeben. Sie ist sinnvoll, wenn eine Platte angeschaut werden soll. Falls aber Sektoren kopiert werden sollen, kann sie nicht verwendet werden. Voreinstellung beim Start ist **nur Anwahl**.

Die automatische Betriebsart wird durch Eingabe von A erreicht. Beispiel: einen Sektor lesen und darstellen:

```
.DA      Laufwerk auswählen
.TO      Spur setzen
.S1      Sektor setzen, lesen und darstellen
```

Die normale Betriebsart (Voreinstellung beim Start) wird durch Eingabe von 0 erreicht. Beispiel: einen Sektor lesen und darstellen:

```
.DA      Laufwerk auswählen
.TO      Spur setzen
.S1      Sektor setzen
.R       Sektor lesen
.P       darstellen
```

**invertierend oder nicht invertierend**

Mit den Kommandos **I** bzw. **J** wird zwischen der invertierenden und der normalen Betriebsart umgeschaltet.

In der invertierenden Betriebsart wird der Inhalt des Sektorpuffers nach dem Lesen sofort automatisch invertiert. Vor dem Schreiben wird der Puffer invertiert (also wieder normal) und danach nochmals invertiert (also wieder invers). Wenn zwischen einer Lese- und einer Schreiboperation invers selektiert wird, stehen anschließend auf der Platte inverse Daten.

Auch der beim Vergleichen benötigte zweite Puffer wird nach dem Lesen invertiert, wenn invers selektiert wurde.

**Anwahl von Laufwerk, Spur und Sektor**

Mit dem **D-Kommando** wird das Laufwerk für den Disk-Monitor gewählt. Es sind alle im CBIOS vereinbarten Laufwerke zulässig. Ein nicht vorhandenes Laufwerk (oder eine nicht verarbeitbare Diskette) bringt als Fehlermeldung ein ? anstelle des Punktes . als Prompt. Beispiel: . DA wählt Laufwerk "A" aus. Nach der Auswahl des Laufwerkes sind Spur und Sektor nicht mehr definiert; sie müssen anschließend neu gesetzt werden. Es gibt keine Vorbesetzung für das Laufwerk. Das D-Kommando benutzt die LOGIN-Funktion, daher muß die Diskette bekannt sein. Beispiel:

```
. dz          Laufwerk Z gibt es nicht
? da         Laufwerk A existiert
.
```

Mit dem **T-Kommando** wird eine Spur gewählt. (Voreinstellung: ungültige Spur!) Danach ist der Sektor undefiniert. Vorher muß ein Laufwerk ausgewählt worden sein. Durch Spurüberlauf kann die Rückseite einer Diskette erreicht werden.

```
. da
. t6
.
```

Achtung: Die RAMDISK beginnt i.allg. nicht bei Spur 0! Dort steht das Betriebssystem im Speicher. Mit dem V-Kommando kann der "OFFSET" der RAMDISK ("reservierte Spuren für das Betriebssystem im Speicher") gefunden werden.

Das **S-Kommando** wählt einen Sektor aus. Der Disk-Monitor arbeitet mit den physikalischen Sektoren der Platte (i.allg. nicht 128 Byte). Der Disk-Monitor holt sich bei der Laufwerksanwahl die Translation-Table des jeweiligen (Floppy-) Laufwerks. Zugriffe auf Sektoren gehen immer durch die interne XLT des Disk-Monitors. (Die XLT kann mit dem L-Kommando verändert werden.) Floppys haben Sektoren ab 1 während RAMDISK und Festplatte Sektoren ab 0 zählen. Vor Eingabe des Sektors müssen Laufwerk und Spur definiert worden sein. Es gibt keine Voreinstellung für den Sektor. Der Monitor arbeitet nicht eindeutig; d.h. eine zu hohe Sektornummer geht auf die Rückseite der Diskette. Beispiel: Bei einer zweiseitigen Diskette mit 80 Spuren pro Seite und 10 Sektoren pro Spur kann der erste Sektor der Spur 0 der Rückseite der Diskette mit 3 unterschiedlichen Eingaben erreicht werden:

- . DA
- . T0
- . S11            durch Sektorüberlauf
- . T80
- . S1            durch Spurüberlauf
- . T80
- . S11           durch Spur- und Sektorüberlauf
- . -

Das **+-Kommando** erhöht den (vorher gewählten noch nicht transformierten) Sektor um 1. Beispiel:

- . da            Laufwerk A
- . t0            Spur 0
- . s9            Sektor 9
- . +            Sektor 10
- . +            Spur 1, Sektor 1 (z.B.)

Das **--Kommando** erniedrigt den (vorher gewählten noch nicht transformierten) Sektor um 1. Beispiel:

- . da            Laufwerk A
- . t40           Spur 40
- . s2            Sektor 2
- . -            Sektor 1
- . -            Spur 39, Sektor 10 (z.B.)

Bei Unterschreiten der Spur 0 wird die Spur ungültig gesetzt. Danach muß mit dem T- oder G-Kommando eine neue Spur gewählt werden.

Das **G-Kommando** wählt Spur und Sektor über die im CP/M verwendeten Blocks an. Dieses Kommando ist sinnvoll, wenn Sie eine bestimmte Datei suchen. **G0** wählt das Inhaltsverzeichnis an. Jetzt können Sie die gewünschte Datei suchen. Die unteren 16 Bytes (10h bis 1fh) des Eintrags sind die zugehörigen Blocknummern. Ein Beispiel:

```
. da          Laufwerk A
. g0         Block 0
. r          lesen
. p          ausgeben

0000:=0044 4953 4B55 5449 4C43 4F4D 0000 0020 .DISKUTILCOM...
0010:=0108 0000 0000 0000 0000 0000 0000 0000 .....
```

. g1 Anfang der Datei (Blocknummer: 1)  
. r ersten Sektor  
. + zweiten Sektor  
bis  
. + achten Sektor (z.B. 1 Block = 8 Sektoren)  
jetzt kommt der zweite Block  
. g8 weiter mit Blocknummer: 8  
. r neunten Sektor

Bei langen Dateien werden mehrere Einträge im Inhaltsverzeichnis benötigt. Wir müssen also jetzt die Einträge genauer ansehen: Ein Eintrag kann mehrere Gruppen a 16 KByte umfassen; abhängig von der Zeigerlänge und der verwendeten Blockgröße. Als Beispiel nehmen wir 8 bit Zeigerlänge und 2 KByte Blockgröße an. (Diese diskettenabhängigen Werte sind mit dem V-Kommando zu sehen.) Wir haben also hier zwei Gruppen pro Eintrag:

```
0 1 2 3 4 5 6 7 8 9 A B C D E F

0000:=0044 4953 4B55 5449 4C43 4F4D 0500 0375 .DISKUTILCOM...
0010:=4142 4344 4546 4748 494A 4B4C 4D4E 4F50 .....
```

An der Adresse Eintrag + F steht der Record-Zähler (75h). Hierin wird von 0 bis 80h gezählt (128 Byte Sektoren). Bei einer längeren Datei werden in Eintrag + C (Inhalt: 05h) die Gruppen (a 16 KByte) weitergezählt (von 0 bis 1F). Reicht auch dies noch nicht aus, geht es in Eintrag + E (a 512 KByte) weiter (Inhalt: 03h).

```
0 1 2 3 4 5 6 7 8 9 A B C D E F

0000:=0044 4953 4B55 5449 4C43 4F4D 0500 0375 .DISKUTILCOM...
0010:=4142 4344 4546 4748 494A 4B4C 4D4E 4F50 .....
```

Obiger Eintrag reicht von  $3 * 512 \text{ K} + 4 * 32 \text{ K}$  bis  $3 * 512 \text{ K} + 5 * 32 \text{ K} + 75\text{h} * 128$ , d.h. von Byte 1703936 bis Byte 1751680 der Datei. Da in Eintrag + F nicht 80h steht ist bei einer sequentiellen Datei hiernach Ende der Datei.

Das **M-Kommando** gibt eine Übersicht der Sektoren einer Platte aus. Das **M-Kommando** gilt für die gesamte Diskette, es beginnt mit der nach **M** eingegebenen Spur und kann durch Eingabe von **cntl C** abgebrochen werden. Für das **M-Kommando** muß nur das Laufwerk definiert worden sein. Unterschieden wird zwischen leeren, leeren inversen, defekten und benutzten Sektoren. Achtung: Bei der RAMDISK und einer Winchester an der HDC-5-Controllerkarte ist nur das Inhaltsverzeichnis mit E5h formatiert. Der Rest ist i.allg. nicht sinnvoll vorbesetzt. Das Map-Kommando geht über das Ende der Platte hinaus. Es muß also von Hand abgebrochen werden (Eingabe: CNTL C). Auf dem Bildschirm erscheint z.B. ein Ausdruck in der Art:

```
. da
. m20
```

#### Belegung der Diskette

- = leer, + = leer (invers), \* = belegt, ? = defekt

```
Spur 00020:= *****          - bedeutet: Sektor enthält nur E5
Spur 00021:= *****_----- + bedeutet: Sektor enthält nur 1A
Spur 00022:= *****          * bedeutet: anderer Inhalt
Spur 00023:= *****          ? bedeutet: Lesefehler
Spur 00024:= *****
Spur 00025:= *****
Spur 00026:= ?***_-----
Spur 00027:= -----
Spur 00028:= *****
Spur 00029:= *****          cntl C eingegeben
***** abgebrochen mit 1 Fehlern *****
```

Angezeigt wird z.B. Spur 26 (dezimal!). 10 Sektoren der Vorderseite werden dargestellt: Sektor 1 ist defekt, die Sektoren 2 bis 4 wurden bereits benutzt, die Sektoren 5 bis 10 sind noch leer (d.h. formatiert). Mit etwas Übung ist es möglich, aus dieser Map die für CP/M notwendigen Parameter der Diskette zu bestimmen.

Pufferoperationen

Mit dem **F-Kommando** wird der Sektorpuffer mit einem festen Wert vorbesetzt. **. F34** füllt den gesamten Puffer mit dem Wert 34 (hexadezimal). Die Pufferlänge ist abhängig von der Sektorlänge des gewählten Laufwerkes (bzw. der Diskette in ihm).

```
. f33
. _
```

Das **R-Kommando** liest einen Sektor. Vorher muß Laufwerk, Spur und Sektor definiert worden sein. R liest den Sektor in den Puffer. Er wird hierbei noch nicht dargestellt. Ein Fragezeichen ? als Prompt gilt als Fehlermeldung; entweder waren die eingestellten Werte falsch, oder der Sektor ist beschädigt.

```
. r
. _
```

Das **W-Kommando** schreibt den Inhalt des Sektorpuffers auf die Diskette zurück. Ein ? anstelle des Punktes als Prompt gilt als Fehlermeldung. Folgende Fehler sind möglich: Die Diskette ist schreibgeschützt, die Werte für Laufwerk, Spur und Sektor sind nicht gesetzt oder falsch oder die Diskette ist nicht richtig eingelegt.

```
. w           ; z. B. Diskettentür war offen, schließen und:
? w          ; erneuter Versuch geht
. _
```

Das **P-Kommando** gibt den Inhalt des Sektorpuffers auf dem Bildschirm aus. Der Sektorpuffer sollte vorher gesetzt sein (i.allg. durch Lesen eines Sektors). Der Sektorpuffer hat immer die Länge eines Sektors der gewählten Diskette (d.h. i.allg. nicht 128 Byte). Der Ausdruck stoppt nach jeweils 128 Byte und wird durch Eingabe von CR fortgesetzt.

```
. p
0000:= 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A *****
0010:= 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A *****
0020:= 2A2A 2A2A 2A2A 2A2A 0D0A 093B 2A20 2063 *****...;* c
0030:= 616C 6375 6C61 7465 2062 6C6F 636B 206C alculate block l
0040:= 656E 6774 6820 616E 6420 6D65 6D6F 7279 ength and memory
0050:= 2073 6563 7469 6F6E 092A 0D0A 093B 2A2A section.*...; **
0060:= 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A *****
0070:= 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A 2A2A *****
```

CR eingeben



Das **C-Kommando** dient dazu, einzelne Bytes im Puffer zu verändern. Syntax ist: **Change Adresse CR** . Angezeigt wird daraufhin die Adresse und der alte Wert. Jetzt kann durch Eingabe eines neuen Wertes dieses Byte verändert werden. Der alte Wert kann beibehalten werden, indem nur ein CR eingegeben wird. Dieser Editiermodus wird durch Eingabe von einem Punkt (.) verlassen. C ändert nur Werte im Puffer, eine Änderung auf der Diskette erfolgt nur durch das W-Kommando. Beispiel:

```
. c0
0000 := 55 99
0001 := 83 .
. _
```

Das **Y-Kommando** vergleicht den Pufferinhalt mit dem (neu) gewählten Sektor. Das **Y-Kommando** kann nicht in der **A-Betriebsart** verwendet werden. Es kann nur sinnvoll bei gleichlangen zu vergleichenden Sektoren arbeiten. Bei gleichen Inhalten erscheint danach wieder der . als Prompt. Bei Unterschieden werden die relative Adresse des Bytes, der Inhalt des Puffers und der Inhalt des angewählten Sektors ausgegeben. Die Adresse und die beiden Inhalte sind hexadezimale Werte. Danach wird auf die Eingabe von CR (weitervergleichen) oder CNTL C (abbrechen) gewartet. Bei gefundenen Unterschieden erscheint danach ein ? als Prompt. Beispiel:

```
. da
. t0
. s0
. r           Spur 0, Sektor 0 lesen
. t2
. s0
. y           vergleichen mit Spur 2, Sektor 0
99:= 23 45   CNTL C eingegeben
?
```

### andere Befehle

Mit dem **?-Kommando** können Sie eine Liste der im Disk-Monitor zulässigen Befehle auf dem Bildschirm erhalten. (siehe Anfang der Disk-Monitor-Beschreibung)

Mit Eingabe des **X-Kommandos** wird der Disk-Monitor verlassen. Anschließend sind Sie wieder im DM.

Das **V-Kommando** gibt die zur gewählten Diskette gehörenden **CP/M-DPB** Parameter aus.

Werte mit 5 Stellen sind in dezimaler, Werte mit 2 oder 4 Stellen in hexadezimaler Darstellung.

Beispiel:

```
. v
      SPT  BSH BLM EXM  DSM    DRM  ALO/1  CKS   OFF  PSH PHM
00040  04  0F  00  00394  00127  C000  00032  00002  02  03
```

. \_

Interpretation der obigen Werte:

40 logische Sektoren (a 128 Byte) pro Spur (SPT = 40)

Blockgröße ist 2 KByte (BSH = 4, BLM = 0F)

16 bit Zeigerlänge, da DSM über 255 ist, somit ist EXM = 0  
(394+1)\*2 = 790 KByte Kapazität (DSM = 394)

Platz für 128 Einträge im Inhaltsverzeichnis (DRM = 127)

128 Einträge benötigen 4 KByte Platz, somit ist ALO = C0  
und AL1 = 0

128 Einträge belegen 32 Sektoren, somit ist CKS = 32

2 Systemspuren sind reserviert (OFF = 2)

512 Byte physikalische Sektorgröße (PSH = 2, PHM = 3)

Das **L-Kommando** erlaubt es, die (interne) Translation-Table des Disk-Monitors zu modifizieren. Als Grundeinstellung wird bei der Auswahl eines Laufwerks die Translation-Table dieses Laufwerks aus dem BIOS geholt. Diese Funktion ist nützlich, um seltsam verschränkte Diskettenformate auszuprobieren. Das L-Kommando kann nur bei Disketten verwendet werden. (RAMDISK und Festplatte haben keine Translation-Table!) Beispiel:

```
. dc
. l
Translation-Table eingeben, 1 Wert pro Zeile
00000:= 00000
00001:= 00001
00002:= 00003
00003:= 00005
00004:= 00007 8          tauschen 8 statt 7
00005:= 00009
00006:= 00002
00007:= 00004
00008:= 00006
00009:= 00008 7          tauschen 7 statt 8
00010:= 00010
00011:= 00000 .         10 Sektoren pro Spur, damit Schluß
```

. \_

Bei Ramdisk oder Festplatte wird das L-Kommando nicht beachtet, da keine Sektorverschränkungstabelle existiert. Es erscheint sofort wieder der . als Prompt.

**Funktion: 2 -- Analysieren von Fremdformaten**

Die Funktion 2 hilft beim Erkennen der für das BIOS notwendigen Parameter einer Diskette. Ein Beispiel:

DM aufrufen:

DISK-MANAGER vers. 1.300

Copyright (c) 1984 by M.K.C

- (1) Disk-Monitor-Programm aufrufen
- (2) fremde Diskette analysieren
- (3) Auswahl von Formaten treffen
- (4) Editieren eines Fremdformates  
(Datei: DISKFORM.DAT)
- (5) Formate ausdrucken
- (9) zurueck zu CP/M

waehle Funktion: 2

gewaehlte Funktion: Diskette analysieren

In welchem Laufwerk ist die unbekannte Diskette ? c

gefundene Werte der unbekanntten Diskette:

	8"	5"	FM	MFM	N	Kopf1	Kopf2	Spur0	Sei-	Spuren	Sek-	SKEW	FLAG
									ten		toren		
<del>501</del>	40	02	00000	00000	00000	01	00040	00009	01	00			

Sie muessen jetzt die folgenden Werte ermitteln:

Lage des Inhaltsverzeichnisses

Anzahl der fuer das Inhaltsverzeichnis benoetigten Sektoren

CP/M Blockgroesse

Lage von Vorder- / Rueckseite

Sektorverschraenkungstabelle

Sie sind jetzt im DISK-MONITOR, (? fuer Hilfe)

Die angezeigten gefundenen Werte sagen folgendes aus:

Wir haben eine einseitige (Seiten = 1) Diskette

mit 5,25" Größe (8"/5" = **5**)

mit double density (FM/MFM = 40)

die Sektorgröße beträgt 512 Byte (N = 2)

Sie hat 40 (oder auch 35, 39 o.ä.) Spuren

9 Sektoren

keine physikalische Verschränkung (SKEW = 1). Wir müssen also wahrscheinlich eine passende Sektorverschränkungstabelle suchen. Der erste Sektor ist 1 (FLAG = 0).

Zuerst suchen wir das Inhaltsverzeichnis:

```
. m                                MAP ausgeben lassen
```

### Belegung der Diskette

- = leer, + = leer (invers), \* = belegt, ? = defekt

```
Spur 00000:= *****
Spur 00001:= *****
Spur 00002:= ****_*_*_*
Spur 00003:= *****
Spur 00004:= *****
Spur 00005:= ***_**_*_*
***** abgebrochen mit 00000 Fehlern *****
```

Da CP/M das Inhaltsverzeichnis sequentiell aufbaut, ist i.allg. die Spur, die als erste Lücken hat, richtig. Das wäre hier die Spur 2. Wir betrachten also jetzt einen Sektor dieser Spur. Meistens ist der erste Sektor der richtige.

```
. t2                                Spur 2 enthält das Inhaltsverzeichnis
. s1                                Wir nehmen Sektor 1
. r                                  lesen ihn
. p                                  und drucken ihn
0000:=004C 4F41 4420 2020 2043 4F4D 0000 000E .LOAD    COM....
0010:=0203 0000 0000 0000 0000 0000 0000 0000 .....
und so weiter....
```

Dies ist ein Teil des Inhaltsverzeichnisses. Somit haben wir den ersten Wert, den wir nachher angeben müssen.

### Lage des Inhaltsverzeichnisses ist Spur 2

Das Byte 0Eh an der Adresse 000F besagt, daß die Datei LOAD.COM 0E (hex) = 14 Sektoren lang ist. Das sind 14 \* 128 Byte = 1,75 KByte. Die Adressen 0010 bis 001F enthalten die Allocation-Liste für diese Datei. Eine 0 bedeutet, daß dieser Block nicht benötigt wird. Somit belegt die Datei LOAD.COM also 2 Blocks (2 und 3). 1,75 KByte (also aufgerundet 2 KByte) durch 2 Blocks ergibt 1 KByte pro Block. (Dieser Wert ist immer auf die nächste Zweierpotenz aufzurunden). Somit ist die Blockgröße also 1 KByte. Wir haben also den nächsten Wert für nachher.

### Blockgröße ist 1 KByte

Jetzt versuchen wir, die Translation-Table aufzubauen. Aus der Map der Spur 2 sehen wir, daß zwischen 2 leeren Sektoren 1 belegter Sektor ist. Dies könnte folgende Sektorverschränkungstabelle sein:

```
1, 3, 5, 7, 9, 2, 4, 6, 8
```

Dies ist leicht auszuprobieren. Wir geben diese Tabelle ein.

```

. l
Translation-Table eingeben, 1 Wert pro Zeile
0000:= 0000          ohne Bedeutung
0001:= 0001          stimmt
0002:= 0002 3       ändern
0003:= 0003 5       ändern
0004:= 0004 7       ändern
0005:= 0005 9       ändern
0006:= 0006 2       ändern
0007:= 0007 4       ändern
0008:= 0008 6       ändern
0009:= 0009 8       ändern
0010:= 0010 .       ab jetzt ohne Bedeutung, also Schluß

```

Jetzt sehen wir uns nochmals die Map der Diskette an. Leere Sektoren können bei CP/M i.allg. nur am Schluß einer Datei oder des Inhaltsverzeichnisses sein. Dies überprüfen wir jetzt.

```

. m2                      MAP ausgeben lassen

                          Belegung der Diskette

- = leer, + = leer (invers), * = belegt, ? = defekt

Spur 00002:= ** -- ** ** *
Spur 00003:= * ** ** ** **
Spur 00004:= ** ** ** ** *
Spur 00005:= * *_ ** *_ **
Spur 00006:= ** ** ** ** *
Spur 00007:= - ** ** ** **
***** abgebrochen mit 0000 Fehlern *****

```

Die Leerzeichen in der obigen Map sind nur in dieser Dokumentation zum besseren Erkennen von jeweils einem Block eingefügt worden. Man sieht, daß die Translation-Table schon recht gut war. Die ersten 4 Sektoren ergeben ein sinnvolles Inhaltsverzeichnis, und alle leeren Sektoren liegen richtig am Ende eines Blocks. Somit nehmen wir an, daß wir die Translation-Table gefunden haben.

**Translation-Table ist: 1, 3, 5, 7, 9, 2, 4, 6, 8**

Ferner ist jetzt leicht zu sehen, daß 4 Sektoren für das Inhaltsverzeichnis benötigt werden. Dies kann nochmals geprüft werden: Wir sehen uns nochmal den ersten Eintrag an:

```

0000:=004C 4F41 4420 2020 2043 4F4D 0000 000E .LOAD COM....
0010:=0203 0000 0000 0000 0000 0000 0000 0000 .....

```

Der erste für Dateien benutzte Block ist 2, somit sind die Blocks 0 und 1 für das Inhaltsverzeichnis reserviert. 2 Blocks ergeben 2 KByte, 4 Sektoren a 512 Byte dasselbe. Damit haben wir:

**4 Sektoren für das Inhaltsverzeichnis**

Damit sind jetzt alle notwendigen Werte gefunden. Wir können also aus dem Monitor zurück zur Analyse.

. x                    jetzt geht es weiter mit der Analyse  
folgende Werte sind noch unbekannt:

```

Spur des Inhaltsverzeichnisses: 2
Blockgroesse in KByte: 1
Fuer das Inhaltsverzeichnis benoetigte Sektoren: 4
Sektorverschraenkungstabelle eingeben (1 Wert pro Zeile)
1
3
5
7
9
2
4
6
8

```

dann erscheint wieder das DM-Menue:

```

(1)   Disk-Monitor-Programm aufrufen
(2)   fremde Diskette analysieren
(3)   Auswahl von Formaten treffen
(4)   Editieren eines Fremdformates
      (Datei: DISKFORM.DAT)
(5)   Formate ausdrucken
(9)   zurueck zu CP/M

```

waehle Funktion: 4  
gewaehlte Funktion: Formate editieren

Zu lesen ist jetzt Format 254, dann muß die neue Nummer eingegeben werden und anschließend das so erstellte neue Format gespeichert werden (siehe Funktion 4: Editieren eines Fremdformates). Zum Arbeiten mit der Diskette muß danach noch das soeben erstellte neue Format aktiviert werden (siehe Funktion 3: Auswahl von Formaten treffen).

Nach dem Verlassen des DISK-MANAGER kann dann mit dieser Diskette gearbeitet werden.

Diese Diskette ist also jetzt vollständig vereinbart. Sie ist ab jetzt in dem Auswahlmenue enthalten. Jetzt ist ausführlich zu testen, ob alle Werte der Diskette richtig vereinbart sind.

**Schreiben Sie erst auf die Diskette, wenn Sie ganz sicher sind!**

Dies war ein relativ einfaches Beispiel, es gibt leider auch wesentlich schwierigere Formate.

**Einige Tips zur Analyse von fremden Disketten:**

Eine Diskette, bei der der Disk-Monitor (in der Analyse) das LOGIN nicht schafft, ist ziemlich zwecklos!  
Zu analysierende Disketten sollten fehlerfrei sein (d.h. keine BAD SECTORS enthalten).

Um Konflikte zu vermeiden, sollten diese Disketten immer nur auf dem Originalsystem formatiert worden sein. (Wenn das Originalsystem einseitige Disketten hat, die zweite Seite aber woanders schon mal formatiert wurde, wird die Analyse fehlerhaft arbeiten.)

Versuchen Sie nicht, kopiergeschützte Disketten zu analysieren! Manche Schutzverfahren bringen die Analyse durcheinander.

Das Inhaltsverzeichnis ist in der MAP leicht zu erkennen. Es liegt höchstwahrscheinlich auf der ersten Spur, die belegte und leere Sektoren enthält. Zur Kontrolle sollte einer der belegten Sektoren angeschaut werden. Er muß als Directory-Sektor Einträge in der folgenden Art enthalten:

Adresse	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1.Zeile	u	c	c	c	c	c	c	c	c	c	c	c	e	0	0	r
2.Zeile	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m

mit u = 0..F oder auch (im CP/M PLUS) 20 oder 21  
c = Großbuchstabe oder Zahlen, keinesfalls Kleinbuchstaben oder Sonderzeichen  
e = 0..1F  
r = 0..80  
m = Allocation Map, d.h. i.allg. fortlaufend steigende Zahlen (1 oder 2 Byte lang)

Spur- oder Sektorüberlauf erkennt man an der Map. Wenn die Vorderseite randvoll ist, und auf der Rückseite nur einige Spuren belegt sind, ist dies ein sicheres Zeichen für Spurüberlauf.

Wenn auf Vorder- und Rückseite etwa gleichviele Spuren belegt sind (z.B. beide bis Spur 43) ist Sektorüberlauf zu vermuten. Hier bleibt jetzt noch zu klären wo das Inhaltsverzeichnis liegt. Liegt es auf einer Vorderseite ist alles in Ordnung. Liegt es auf der Rückseite, muß Spurüberlauf und Seiten = 3 angegeben werden.

Die Blockgröße kann man aus dem recordcounter r in der ersten Zeile des Inhaltsverzeichnisses und den belegten Blocks berechnen. e sollte dabei 0 sein. Jeder record ist 128 Byte lang, in r steht die Anzahl der Records; somit ist diese Datei r \* 128 Byte lang (Achtung: r ist hexadezimal!). Jetzt sind die von 0 verschiedenen Blocknummern in der zweiten Zeile zu zählen. Die Länge der Datei durch die Anzahl der Blocks geteilt, ergibt aufgerundet die Blockgröße. Als Blockgrößen sind nur 1K, 2K, 4K, 8K oder 16K zulässig. Bei Floppies werden meist nur 1 bis 4 K benutzt.

Die Sektorverschränkungstabelle muß leider geraten werden. Einige Tips: CP/M vergibt Blocknummern sequentiell; wenn Sie also eine mit PIP kopierte Diskette haben, können Sie bei der Suche nach der

Verschränkung im Inhaltsverzeichnis Blocknummern ansehen. Ein Ausdruck des Inhaltsverzeichnisses vom Originalsystem hilft auch.

Für das Inhaltsverzeichnis können nur Blocks vergeben werden. Sie können daher aus der MAP der Directory-Spur und der Blockgröße die Anzahl der Sektoren raten. Leere Sektoren in dieser Spur sollten immer dem Directory zugeordnet werden. Eine andere Möglichkeit besteht darin, die niedrigste verwendete Blocknummer zu finden (hier zählen aber auch alle gelöschten Einträge mit). Alle Blocks darunter sollten zum Inhaltsverzeichnis gerechnet werden.

Es gibt leider auch exotische Versionen von Formaten:

Manche Systeme benutzen 16 Bit Blocknummern, obwohl sie weniger als 255 Blocks Kapazität haben. Ausweg: Geben Sie eine falsche Kapazität (mehr als 255 Blocks) an. Hierbei erscheint dann im Betrieb anstelle der Meldung: DISK FULL die Fehlermeldung BAD SECTOR bei Überschreiten der richtigen Kapazität.

CANON AS100 hat 9 Sektoren a 512 Byte, benutzt aber davon für CP/M nur 8. Somit gilt die Berechnung für SPT leider nicht mehr.

Die SORCERER haben eine Translation-Table, die nicht mit Sektor 1 anfängt: 6, 11, 16, 5, 10 .....

Manche Rechner beginnen nicht mit dem physikalischen Sektor 1, sondern mit dem Sektor 0. Dies wird in FLAG0 durch das Bit 1 angezeigt. FLAG = 2 weist also auf Anfangssektor 0 hin.

Dann hat beim ALPHATRONIC P4 jemand den verfügbaren Platz optimiert und ist zu folgender Spuraufteilung gekommen:

v0, v1, r0, r1, v2, r2, v3, r3 ... v79, r79

hier sind die beiden ersten Spuren reserviert, damit liegt das Inhaltsverzeichnis auf Spur 0 der Rückseite. Dies ist leider nicht zu lösen. Hierfür wurde das P4-Flag eingeführt. (Bit 4 in FLAG0 und Seiten = 3)

Jetzt hat noch jemand sich etwas ganz besonderes für zweiseitige Disketten einfallen lassen. Es gibt dann die folgende Spureinteilung: Auf der Vorderseite wird von außen nach innen gestept, auf der Rückseite aber von innen nach außen:

v0, v1 ..... v79, r79, r78 ..... r1, r0

Dies würde eine Geschwindigkeitssteigerung ergeben, wenn ohne Inhaltsverzeichnis gearbeitet würde. Unter CP/M bringt diese Lösung aber keine Vorteile. Hierfür ist Bit 3 in FLAG0 zuständig.

Weiterhin kann die Kopfnummer der Rückseite 0 oder 1 sein. 1 ist der Normalfall. Dies wird durch Kopf1 und Kopf2 in der Ergebniszeile angezeigt. Gesetzt werden muß bei gleichen Kopfnummern das Bit 2 in FLAG0.

Jetzt bleibt noch die Behandlung der Rückseite übrig. Sie kann auf 2 Versionen erreicht werden:

- Durch Spurüberlauf, d.h. nach der letzten Spur der Vorderseite geht es auf der Rückseite weiter. Lösung: Spurüberlauf



- Durch Sektorüberlauf, d.h. nach dem letzten Sektor auf der Vorderseite geht es auf der Rückseite weiter. Lösung: Sektorüberlauf
- Im Sektorüberlauf können aber auch Sektoren von der Vorder- und Rückseite wild gemischt benutzt werden. Lösung: Sektorüberlauf
- durch abwechselndes Benutzen von Vorder- und Rückseitenspuren. Lösung: Spurüberlauf und Seiten = 3 vereinbaren

Es gibt aber auch ein paar bisher ungelöste Versionen von fremden Formaten:

- Jemand hat das Inhaltsverzeichnis in die Mitte der Diskette verlegt, geht von dort aus erst nach innen, und dann anschließend von dort aus nach außen. Dies ließe sich nur mit einer vollständigen Spur-Verschrankungstabelle lösen. Dieser Aufwand lohnt aber für solche Exoten nicht.
- Jemand hat den Sinn der Sektorverschrankung nicht verstanden, und verschrankt logische 128 Byte Sektoren innerhalb eines physikalischen Sektors. Dies war unter CP/M 2.2 leider möglich, ist aber unter CP/M PLUS unmöglich, da die Translation-Table für physikalische Sektoren gilt.

Hier ist nochmals eine Zusammenfassung der FLAGS für Sonderfälle:

Dichte:     0       single density (FM)  
            40       double density (MFM)  
            C0       bis zum Directory FM und danach MFM. Die Systemspuren können dann nicht bearbeitet werden. Diese Disketten können auch nicht richtig formatiert werden.

Seiten:     1       einseitig  
            2       zweiseitig, übliche Version  
            3       zweiseitig, abwechselnd Vorder- und Rückseite

FLAG0:     Bit 0   normal (0) oder invers (1)  
            Bit 1   Sektor 1 (0) oder Sektor 0 (1) als erster  
            Bit 2   Kopf weiterzählen (0) oder aus FLAG1 (1) bei Übergang auf die Rückseite  
            Bit 3   auf der Rückseite von außen nach innen (0) oder von innen nach außen (1) steppen  
            Bit 4   (0) normal oder (1) für ALPHATRONIC P4

FLAG1:     Bit 0   internes (1) oder normales (0) Format  
            Bit 1-6 Nummer des zweiten Kopfes (vgl. FLAG0 bit 2)

Weiterhin hat bei allen beim LOGIN benutzten Vergleichswerten das höchste Bit eine besondere Bedeutung. Ein gesetztes 7tes Bit besagt, daß dieser Wert beim Vergleich nicht beachtet wird.

Jetzt sind zur Vereinbarung von Formaten nur noch die Gaplängen zu bestimmen. Die R/W-GAP kann bei einem vernünftigen Datenseparator i.allg. auf 2 gesetzt werden. Das geht (fast) immer. Die FORMAT-GAP muß etwas genauer betrachtet werden. Ihr Sinn besteht darin, hinter einem Sektor soviel Platz zu lassen, daß bei Geschwindigkeitsschwankungen der folgende Sektor nicht beeinträchtigt wird. Als Drehzahlschwankung sollte i.allg. mit +/- 2% gerechnet wer-

den. Somit können Werte für die FORMAT-GAP angegeben werden:

128 Byte	6 Byte min
256 Byte	11 Byte min
512 Byte	21 Byte min
1024 Byte	41 Byte min

Es ist aber besser, sich nach den Werten des anderen Computers zu richten. Diese sind leider nur durch Probieren oder Nachschauen im Quelltext des Formatierprogrammes zu bestimmen. Es ist zu beachten, daß durch zu lange FORMAT-GAPs die Kapazität einer Spur überschritten werden kann. Das Formatierprogramm beschwert sich dann. Werte für einige übliche Formate können in den Datenblättern der Floppy-Controller nachgesehen werden.

**Funktion: 3 -- Formate auswählen**

Mit der Funktion 3 Formate auswählen legen Sie die momentan aktiven Fremdformate fest. Dies sind die Formate, die Ihr System automatisch erkennt. Die Liste der bekannten Formate (Datei: DISKFORM.DAT) wird nur auf Wunsch ausgegeben. Sie müssen hierfür statt der 3 eine 3 und ein ? eingeben.

Die Datei DISKFORM.DAT muß auf dem Laufwerk vorhanden sein, von dem DM geladen wurde.

Diese Funktion des Disk-Managers ist auch im Batch-Betrieb erreichbar (siehe dort).

wähle Funktion: 3?  
gewählte Funktion: Formate auswählen

Liste der bereits bekannten Formate:

( 1) ZDOS 80 trk DS	( 2) ZDOS 40 trk DS	( 3) ZDOS 8" DD DS
( 4) CP/M A1 8" SD	( 5) ALPHATRONIC P2	( 6) SHARP EOS D&Z
( 7) DEC VT180	( 8) ALTOS 5 CP/M	( 9) TOSHIBA 5"
(10) WAVEMATE	(11) TELEVIDEO	(12) Alpatronic
(13) SORCERER 35 trk	(14) OSBORNE SD	(15) SORCERER 40 trk
(16) ZDOS 80 trk SS	(17) ZDOS 40 trk SS	(18) Philips P3800
(19) COSICOMP	(20) ZENITH Z89	(21) ZDOS 8" DD SS

Liste der aktiven Formate:

00001            00002    -       -       -       -       -       -       -       -

Welche Formate sollen deaktiviert werden?

. als Endezeichen

1

.

Welche Formate sollen aktiviert werden?

. als Endezeichen

3

.

Folgende Fehlermeldungen können hierbei auftreten:

\*\*\*\*\* dieses Format war nicht aktiv \*\*\*\*\*

Ein nicht aktiviertes Format kann nicht gelöscht werden, Sie können einfach weitermachen.

\*\*\*\*\* dieses Format ist nicht bekannt \*\*\*\*\*

Dieses Format ist nicht in der Datei DISKFORM.DAT enthalten, es ist dem System unbekannt und kann daher auch nicht aktiviert werden. Suchen Sie ein bekanntes Format aus.

\*\*\*\*\* kein Platz fuer weitere Formate \*\*\*\*\*

Im System ist kein Platz mehr für weitere Formate. Um das gewünschte Format zu aktivieren, müssen Sie vorher Platz schaffen, d.h. beenden Sie das Einfügen mit Eingabe eines Punktes, rufen Sie erneut 'Formate auswählen' auf, löschen Sie nicht mehr benötigte und erst jetzt können Sie das oben vergeblich versuchte Format aktivieren.

\*\*\*\*\* dieses Format ist unvertraeglich mit einem bereits aktiven \*\*\*\*\*

Sie haben bereits ein Format aktiv, das von dem jetzt gewünschten nicht unterscheidbar ist. Löschen Sie vorher das alte Format. Das Format wurde nicht in die Liste aufgenommen.

\*\*\*\*\* DISKFORM.DAT kann nicht gelesen werden \*\*\*\*\*

Die Datei DISKFORM.DAT fehlt oder kann nicht gelesen werden. DISKFORM.DAT wird auf dem Laufwerk gesucht, von dem DM geladen wurde! Verlassen Sie den DISK-MANAGER und kopieren Sie eine gültige Datei auf dieses Laufwerk. DISKFORM.DAT darf weder schreibgeschützt noch mit einem Password versehen sein. Jetzt können Sie es erneut probieren.

**Funktion: 4 -- Formate editieren (in DISKFORM.DAT)**

Mit der Funktion 4 **Formate editieren** können Sie Formate editieren. Durch Eingabe eines Punktes kann die jeweilige Funktion beendet werden. Bei Eingabe von nur CR wird der vorherige Wert nicht verändert. Ein Beispiel:

```

        waehle Funktion: 4
gewaehlte Funktion:  Format editieren

Welches Format soll gelesen werden          1
neue Nummer:                               00001 99
Name des Formats:                           ZDOS 80 trk DS
Diskgroesse (0=8", 1=5" } 2=high density) 0001
Dichte (0=FM, 40=MFM, C0=FM/MFM)           0040
Sektorgroesse (0=128, 1=256, 2=512, 3=1024) 0002
Nummer der Seite 0:                         00000
Nummer der Spur 0:                          00000
Seiten (1 oder 2):                          0002 1
Spuren / Seite:                             00080
Sektoren (einseitig):                       00010
read/write gap:                             00016
format gap:                                 00030 32
physikalische Verschraenkung:               00001 2
FLAG0 (inv,sec0,head,revstep,P3,...)       0001
FLAG1 (intern,Kopf2)                        0001

DPB -- SPT    00040
DPB -- BSH    00004
DPB -- BLM    00015
DPB -- EXM    00000
DPB -- DSM    00394
DPB -- DRM    00127
DPB -- ALO    00000
DPB -- AL1    00000
DPB -- CKS    00032
DPB -- OFF    00002
DPB -- PSH    00002
DPB -- PHM    00003
XLT ----- 00001
XLT ----- 00003
XLT ----- 00005
XLT ----- 00007
XLT ----- 00009
XLT ----- 00002
XLT ----- 00004
XLT ----- 00006
XLT ----- 00008
XLT ----- 00010
XLT ----- 00000 .

```

Speichern (S), Abbrechen (.) oder nochmal (N) : .

Die Datei DISKFORM.DAT darf nicht read-only gesetzt sein. Ferner darf kein Password vereinbart sein.

**Funktion: 5 -- Formate ausdrucken**

Mit der Funktion 5 **Formate ausdrucken** können Sie eine Liste der aktuell in der DISKFORM.DAT enthaltenen Formate erzeugen. Dieser Ausdruck kann wahlweise entweder auf dem Bildschirm oder Drucker erfolgen. Ferner kann angegeben werden, ob die Translation-Table mitausgegeben werden soll. Bei der Ausgabe zum Drucker wird nach jeweils ca. 60 Zeilen ein FORM-FEED ausgegeben. Bei Ausgabe auf dem Terminal wird nach je ca. 16 Zeilen auf ein CR gewartet.

DISK-MANAGER vers. 1.600

&lt;5&gt; Formate ausdrucken

wähle Funktion: 5

gewählte Funktion: Formate drucken

Ausgabe zum Drucker (P), mit XLT (X) : p

## Druckerausgabe ohne Translation Table:

Nr.	Name	"	D	F	Sp	Sk	IF	SPT	B	B	E	DSM	DRM	AL01	CS	0	P	P
001	ZDOS 80 trk DS	5	D	2	80	10	01	040	4	0F	0	394	127	C000	32	2	2	3
002	ZDOS 40 trk DS	5	D	2	40	10	01	040	4	0F	1	194	063	8000	16	2	2	3
003	ZDOS 8" DD DS	8	D	2	77	15	01	060	4	0F	0	569	191	E000	48	2	2	3
004	CP/M A1 8" SD	8	S	1	77	26	00	026	3	07	0	242	063	C000	16	2	0	0
005	ZDOS 80 trk SS	5	D	1	80	10	01	040	4	0F	0	194	127	C000	32	2	2	3
006	ZDOS 40 trk SS	5	D	1	40	10	01	040	4	0F	1	094	063	8000	16	2	2	3
007	ZDOS 8" DD SS	8	D	1	77	15	01	060	4	0F	0	280	191	E000	48	2	2	3
008	DEC VT180	5	D	1	40	09	00	036	3	07	0	170	063	C000	16	2	2	3
009	DEC RAINBOW 100	5	D	1	80	10	00	040	4	0F	1	194	127	C000	32	2	2	3 ^C

## Druckerausgabe mit Translation-Table:

Nr.	Name	"	D	F	Sp	Sk	IF	SPT	B	B	E	DSM	DRM	AL01	CS	0	P	P									
001	ZDOS 80 trk DS	5	D	2	80	10	01	040	4	0F	0	394	127	C000	32	2	2	3									
												01	03	05	07	09	02	04	06	08	10						
002	ZDOS 40 trk DS	5	D	2	40	10	01	040	4	0F	1	194	063	8000	16	2	2	3									
												01	03	05	07	09	02	04	06	08	10						
003	ZDOS 8" DD DS	8	D	2	77	15	01	060	4	0F	0	569	191	E000	48	2	2	3									
												01	05	09	13	02	06	10	14	03	07	11	15	04	08	12	
004	CP/M A1 8" SD	8	S	1	77	26	00	026	3	07	0	242	063	C000	16	2	0	0									
												01	07	13	19	25	05	11	17	23	03	09	15	21	02	08	14
												20	26	06	12	18	24	04	10	16	22						
005	ZDOS 80 trk SS	5	D	1	80	10	01	040	4	0F	0	194	127	C000	32	2	2	3									
												01	03	05	07	09	02	04	06	08	10						
006	ZDOS 40 trk SS	5	D	1	40	10	01	040	4	0F	1	094	063	8000	16	2	2	3									
												01	03	05	07	09	02	04	06	08	10						
007	ZDOS 8" DD SS	8	D	1	77	15	01	060	4	0F	0	280	191	E000	48	2	2	3									
												01	05	09	13	02	06	10	14	03	07	11	15	04	08	12	
008	DEC VT180	5	D	1	40	09	00	036	3	07	0	170	063	C000	16	2	2	3									
												01	03	05	07	09	02	04	06	08							

Die ausgegebene Liste der in der DISKFORM.DAT enthaltenen Formate enthält die folgenden Werte:

Nr ist die Nummer des Formates (dezimal)  
 Name ist der Name in der DISKFORM.DAT  
 " ist die Größe der Diskette (5,25" ~~oder 8"~~ *od. High density*)  
 D ist die Dichte Double density, Single density oder Changing density  
 F ist die Anzahl der Seiten  
 Sp ist die Anzahl der Spuren (dezimal)  
 Sk ist die Anzahl der Sektoren pro Spur (dezimal)  
 IF ist das FLAG0 (vgl. letzte Seite) (hexadezimal)  
     Bit 0 normal (0) oder invers (1)  
     Bit 1 Sektor 1 (0) oder Sektor 0 (1) als erster  
     Bit 2 Kopf weiterzählen (0) oder gleichlassen (1) bei Übergang auf die Rückseite  
     Bit 3 auf der Rückseite von außen nach innen (0) oder von innen nach außen (1) steppen  
     Bit 4 (0) normal oder (1) für ALPHATRONIC P4  
 SPT (SPT) ist sector per track (dezimal)  
 B (BSH) ist block-shift-factor (hexadezimal)  
 B (BLM) ist block-mask (hexadezimal)  
 E (EXM) ist extent-mask (hexadezimal)  
 DSM (DSM) ist Kapazität in Blocks - 1 (dezimal)  
 DRM (DRM) ist Anzahl der Directory-Einträge - 1 (dezimal)  
 AL01 (ALO, AL1) beschreibt die reservierten Blocks für das Directory (hexadezimal)  
 CS (CKS) ist checksumsize (dezimal)  
 O (OFF) ist offset (hexadezimal)  
 P (PSH) ist physical sector size (? = Fehler?) (hexadezimal)  
 P (PHM) ist physical sector mask (hexadezimal)

Anschließend folgt in den nächsten Zeilen die XLT (dezimal). Die Translation-Table darf bis zu 52 Werte enthalten (CBIOS 1.6). Ausgegeben werden nur die im CP/M benötigten Werte. Jede Zeile enthält 16 dezimale Werte.

Sowohl der DISK-MANAGER wie auch das FORMAT-Programm sind auf die Datei DISKFORM.DAT angewiesen. Diese enthält alle dem System bekannten Formate. Prinzipieller Aufbau der DISKFORM.DAT:

Jedes vereinbarte Format belegt genau einen (128 Byte langen) Sektor der Datei. Die Sektornummer (ab 0 gezählt) entspricht der Nummer des Formates. Als Formatnummern sind nur die Werte zwischen 1 und 253 erlaubt. Somit können maximal die Sektoren 1 bis 253 mit Formatbeschreibungen belegt sein. Der Sektor 0 enthält die Versionskennung der DISKFORM.DAT als ASCII-Text. Der Aufbau innerhalb eines Sektors ist im Kapitel 4.4 (Disketten-Steuer-Parameter) beschrieben.

Falls Sie eigene Formate vereinbart haben, müssen Sie jetzt zwei Dateien mischen. Sie benötigen hierzu die beiden Dateien, den DM und den SID. Zuerst suchen Sie im DM das Format mit der höchsten Nummer in der Originaldatei. Jetzt nennen Sie die Original-DISKFORM.DAT in DISKFORM.XXX um, und kopieren Ihre DISKFORM.DAT dazu. Jetzt müssen Sie alle die Formate in Ihrer DISKFORM.DAT auf höhere Nummern verschieben. Dies ist im DM mit der Funktion `Format editieren` leicht möglich. Sie editieren das Format "Nummer+1", lesen eins Ihrer Formate, und speichern es ab. Diese Prozedur wird solange wiederholt, bis Sie alle Ihre Formate auf höhere Nummern kopiert haben. Jetzt rufen Sie den SID auf, lesen zuerst Ihre DISKFORM.DAT mit dem Befehl `RDISKFORM.DAT` ein, dann überschreiben Sie den Anfang mit der Originaldatei mit `RDISKFORM.XXX`. Anschließend schreiben Sie die so überlagerten Formatbeschreibungen mit dem Befehl `WDISKFORM.DAT 100 XXXX` zurück. Hier steht XXXX für die höchste mit Ihrer DISKFORM.DAT belegte Adresse. (Diese Adresse erhält man im SID, indem man von MSZE 1 abzieht.)

Beim Übergang von Version 1.5 nach Version 1.6 müssen evtl. einige Formate geändert werden. Das BIOS Version 1.600 kann jetzt auch Disketten, die mit Sektor 0 beginnen, lesen. Alle diese Formate, die man früher hineinpfuschen (in der XLT Sektoren ab 0 statt ab 1) konnte, müssen jetzt korrigiert werden.



2.3 FORMAT

FORMAT ist ein Formatierprogramm für alle Laufwerke, d.h. es können sowohl Disketten wie auch die Festplatte und die RAMDISK formatiert werden. Ein Beispiel:

A>FORMAT

FORMAT (Formatierprogramm fuer CP/M PLUS) vers. 1.<sup>700</sup>~~600~~

Copyright (c) 1984, 85 by M.K.C

mit <ESC> kann jede Frage abgebrochen werden

In welchem Laufwerk soll formatiert werden? (A..P) A

Welches Diskettenformat soll benutzt werden? (? fuer Liste) ?

< 1> ZDOS 80 trk DS	< 2> ZDOS 40 trk DS	< 5> ALPHATRONIC P2
< 6> SHARP EOS D&Z	< 7> DEC VT180	< 8> ALTOS 5 CP/M
< 9> TOSHIBA 5"	< 10> WAVEMATE	< 11> TELEVIDEO
< 12> Alphatronic	< 13> SORCERER 35 trk	< 14> OSBORNE SD
< 15> SORCERER 40 trk	< 16> ZDOS 80 trk SS	< 17> ZDOS 40 trk SS
< 18> Philips P3800	< 19> COSICOMP	

Welches Diskettenformat soll benutzt werden? (? fuer Liste) ? 1

soll die Diskette in Laufwerk A formatiert werden ? <j/n> j  
Formatiert wird Laufwerk A, Spur 00000 bis 00159

soll die Diskette in Laufwerk A formatiert werden ? <j/n> j

A>

Das Formatierprogramm FORMAT v.1.6xx stellt die Liste der Formate nur noch auf Wunsch (Eingabe ?) auf dem Bildschirm dar. Im allgemeinen ist diese Liste nicht erforderlich, da Sie Ihre Formatnummer kennen. Ferner werden bei der Darstellung nur die in dem bereits ausgewählten Laufwerk möglichen Formate angezeigt. Somit entfallen bei einem 5,25" Laufwerk alle 8"-Formate.

FORMAT kennt die folgenden Fehlermeldungen:

```
*****
*****      falsches Betriebssystem      *****
*****      FORMAT kann nicht arbeiten    *****
*****
```

Das Formatprogramm und das CP/M PLUS CBIOS passen nicht zusammen. Diese Version des FORMAT-Programmes ist unter dieser Systemversion nicht lauffähig. Sie haben unterschiedliche Versionen gemischt!

### Schreibfehler DDB

Der Beschreibungsblock für eine interne Diskette kann nicht fehlerfrei geschrieben werden. Nehmen Sie eine andere Diskette. Diese arbeitet recht unzuverlässig.

### RAM-Disk formatieren => <j/n>

Kontrollabfrage, ob Sie wirklich die RAMDISK formatieren wollen. Dies ist i.allg. nicht erforderlich, da die RAMDISK nach jedem Einschalten (nicht RESET) automatisch formatiert wird. Formatiert wird aus Zeitgründen immer nur das Inhaltsverzeichnis der RAMDISK.

### Winchester-Laufwerk A: formatieren => <j/n>

Sind Sie wirklich ganz sicher, daß Sie Ihre Winchester formatieren wollen. Nach dem Formatieren sind alle Daten verloren. Geben Sie i.allg. kein j oder y ein, da Sie sich fast immer beim Laufwerk vertippt haben dürften. Die Winchester muß nur formatiert werden, wenn sie defekte Sektoren hat. Beim Formatieren wird der Eintrag .heads.. in das Inhaltsverzeichnis geschrieben; dieser Eintrag ist zum BOOTEN von der Winchester unbedingt notwendig. Da die Winchester nicht wechselbar ist, müssen Sie nach dem Formatieren RESET betätigen. Es gibt keine andere Lösung, die Hash-Tabelle und Buffer zu aktualisieren. Wenn Sie RESET vergessen, weiß CP/M nicht, daß Sie die Winchester formatiert haben, und geht weiterhin von dem alten Zustand aus. Dies kann zu schwerwiegenden Systemfehlern führen. Beim Formatieren der Winchester wird der optimale Skew-Faktor aus dem Systemtakt berechnet. Sie sollten also beim Umbau Ihres Rechners die Festplatte neu formatieren, um optimale Geschwindigkeit zu erreichen.

*Anzahl Spuren*  
 \*\*\*\*\* ~~Laufwerksdaten~~ nicht bekannt \*\*\*\*\*  
 sind die Spuren von Laufwerk und Diskette gleich? (ja: => <CR>)

Diese Meldung kann nur in unseren Auslieferversionen auftreten. Uns sind leider i.allg. die von Ihnen verwendeten Laufwerke nicht bekannt. Achten Sie bitte darauf, daß Sie jetzt nur Disketten mit der gleichen Spurdichte wie Ihr Laufwerk formatieren können. Wenn Sie einseitige Laufwerke (oder 8"-Disketten) verwenden, können Sie jetzt auch nur einseitige Formate auswählen. Das System verläßt sich darauf, daß Sie jetzt nur sinnvolle Formate auswählen; die sonst übliche Prüfung Ihrer Eingaben findet nicht statt. Bei der EPC-II Auslieferversion kann nicht zwischen 5,25" und 8" - Laufwerken unterschieden werden (Typ FL058); somit können Sie auch hier Fehler durch falsche Eingaben verursachen.

**Dieses Disketten-Format ist nicht vorhanden**

Sie haben ein Format ausgewählt, welches nicht in DISKFORM.DAT (dem oben angezeigten Menue) vorhanden ist. Versuchen Sie es mit einem vorhandenen Format, dann können Sie auch arbeiten!

**Dieses Laufwerk ist nicht ~~benutzbar~~ installiert**

*Das Laufwerk ist in Ihrem mit dem angegebenen Buchstaben ist in Ihrem Hardwarefehler, oder keine Diskette im Laufwerk system nicht installiert!*

**Disketten-Format und gewaehltes Laufwerk passen nicht  
 oder  
 falsches Disketten-Format**

Die Prüfung Ihrer Eingaben hat ergeben, daß dieses Format in dem von Ihnen selektierten Laufwerk nicht realisierbar ist. z.B. ein 5,25" Format in einem 8" Laufwerk ist unmöglich, außer in der Auslieferversion bei EPC II.

**Formatierfehler**

Die soeben formatierte Spur kann nicht fehlerfrei gelesen werden. Nehmen Sie besser eine andere Diskette (oder prüfen Sie Ihre Hardware).

**die Diskette ist schreibgeschuetzt**

Eine schreibgeschützte Diskette kann nicht beschrieben, also auch nicht formatiert werden. Entfernen Sie den Schreibschutz, d.h. bei 5,25"-Disketten kommt der Aufkleber ab, bei 8"-Disketten müssen Sie einen Aufkleber anbringen. (Eine im System auf R/O gesetzte Diskette ist trotzdem formatierbar.)

~~unmoeglicher Fehler~~ *diese Meldung dürfte nicht erscheinen! .....*

Diese Fehlermeldung dürfen Sie eigentlich nie sehen. Wenn sie trotzdem auftritt, ändert sich Ihre Konfiguration im laufenden System. Dies läßt auf Hardware- (oder bisher von uns nicht gefundene Software-) Fehler schließen. Schicken Sie in diesem Fall bitte Ihre momentan benutzten Disketten zur Kontrolle an uns ein.

2.4 DISKCOPY

DISKCOPY ist ein direktes Kopierprogramm für Disketten. Es kopiert physikalisch, geht also nicht über CP/M. Sie können mit DISKCOPY auch nicht-CP/M-kompatible Disketten kopieren. Voraussetzung zur Benutzung für DISKCOPY sind 2 gleiche Laufwerke. DISKCOPY formatiert zuerst die Zieldiskette, und kopiert und prüft danach spurweise. Fehlerhafte Sektoren der Quelldiskette führen nicht zum sofortigen Abbruch. Alle Fehler werden am Schluß aufgelistet. Nach jeweils 20 Fehlern wird der Ausdruck an der Konsole gestoppt.

Ein Beispiel:

```
A>DISKCOPY
DISKCOPY (Diskettenkopierprogramm)
```

Copyright (c) 1984 by M.K.C

DISKCOPY ist ein schnelles 1:1 Kopierprogramm fuer gleiche Disketten. Die Zieldiskette wird zuerst formatiert, danach wird kopiert und geprueft. fehlerhafte Sektoren der Quelldiskette werden durch Leerzeichen auf der Zieldiskette ersetzt. Nach Beendigung des Kopiervorganges werden die fehlerhaften Sektoren aufgelistet. Das Programm startet sofort nach Angabe des Quell- und Ziellaufwerkes.

```
Laufwerk mit Originaldiskette : A
Laufwerk mit neuer Diskette fuer Kopie : B
Formatiert wird Spur :      00000 bis 00079
Kopiert wird Spur :      00000 bis 00079
```

```
Lesefehler auf Spur: 00067 Sektor: 00004
Lesefehler auf Spur: 00068 Sektor: 00004
```

Ziel- und Quelldiskette stimmen nicht ueberein. Die Zieldiskette muss vor Benutzung unter CP/M erst mit dem -> DM <- restauriert werden. Entfaellt dieses kann ihre Benutzung zu Systemabstuerzen fuehren.

zurueck zu CP/M => <CR> carriage return  
A>

Fehlermeldungen von DISKCOPY:

\*\*\*\*\* Fehler, Kopieren wird abgebrochen \*\*\*\*\*

Schreib- oder Prüffehler, das Kopieren wird abgebrochen, versuchen Sie es erneut mit einer neuen Zieldiskette.

```
*****
***** falsches Betriebssystem *****
***** DISKCOPY kann nicht arbeiten *****
*****
```

Sie haben unterschiedliche Versionen gemischt, DISKCOPY weigert sich zu arbeiten, da es mit einem falschen CBIOS nicht richtig arbeiten kann.

Leider unmöglich!

Zum Kopieren sind 2 Laufwerke erforderlich. Ziel- und Quell-Laufwerk dürfen nicht identisch sein.

5,25" und 8" gemischt ist nicht möglich  
oder  
\*\*\*\*\* nur fuer Floppylaufwerke \*\*\*\*\*

Wie bereits oben gesagt: gleiche Floppy-Laufwerke!

\*\*\*\*\* raetselhafter Fehler \*\*\*\*\*

Kann eigentlich nie auftreten, schicken Sie uns bitte Ihre momentan benutzten Disketten zur Fehleranalyse.

\*\*\*\*\* Schreib- oder Lesefehler \*\*\*\*\*  
oder  
\*\*\*\*\* Schreibfehler \*\*\*\*\*

Fehler auf der Zieldiskette, nehmen Sie eine andere

\*\*\*\*\* Ziellaufwerk ist schreibgeschuetzt \*\*\*\*\*

andere Diskette nehmen, oder Schreibschutz (physikalisch) aufheben.

\*\*\*\*\* zu viele Lesefehler \*\*\*\*\*

ab ca. 100 defekten Sektoren der Quelldiskette ist ein Kopieren nicht mehr sinnvoll. DISKCOPY bricht ab.

Lesefehler auf Spur: \_\_\_\_\_ Sektor: \_\_\_\_\_  
Ziel- und Quelldiskette stimmen nicht ueberein. Die Zieldiskette muss vor Benutzung unter CP/M erst mit dem -> DM <- restauriert werden. Entfaellt dieses kann ihre Benutzung zu Systemabstuerzen fuehren.

Sie haben eine beschädigte Diskette kopiert. Lesefehler werden mit den physikalischen Spuren und Sektoren gemeldet. Gehen Sie in den DISK-MANAGER und reparieren Sie erst einmal die gerade erzeugte Kopie. Nehmen Sie diese Warnung bitte ernst. Es können wirklich krasse Fehler auftreten.

**2.5 CONFIG (Schnittstellen konfigurieren)**

Von DRI ist für den Character-I/O des CP/M PLUS lediglich die Einstellung der Baudraten und eines XON/XOFF-Protokolls vorgesehen worden. Das Dienstprogramm DEVICE von DRI ist deshalb auch nur in der Lage diese 2 Eigenschaften eines zeichenbezogenen physischen Gerätes einzustellen. In CBIOS 1.8 können jedoch folgende Eigenschaften von physischen Geräten einstellbar sein:

**- Baudraten von 50Baud bis 19200Baud**

Achtung: Bei der CPU II und der HKM-CPU ist 19200Baud bei 4MHz Systemtakt mit einem zu großen Fehler behaftet! Diese Baudrate sollte deshalb bei diesen CPU-Platinen unter 4MHz nicht eingestellt werden! Bei den Platinen der CPU 180x-Serie sind folgende Baudraten nicht einstellbar: 75Baud, 110Baud, 134Baud, 1800Baud, 3600Baud und 7200Baud. Falls versucht wird, eine dieser Baudraten einzuschalten, bleibt die vorher eingestellte Baudrate unverändert.

**- XON/XOFF-Protokoll**

Achtung: Das XON/XOFF-Protokoll benötigt einen seriellen Input ausschließlich für das Protokoll! Dieser Input darf deshalb keinem logischen Eingabegerät (CONIN:, AUXIN:) zugeordnet werden, d.h. ein Terminal mit XON/XOFF kann als Konsole nicht benutzt werden!

**- Hardware-Handshake**

Hardware-Handshake kann ein- oder ausgeschaltet werden. Bei eingeschaltetem Hardware-Handshake werden die Inputs CTS (Clear To Send) und DCD (Data Carrier Detect) als Handshake-Leitungen benutzt. Falls an einem der beiden Eingänge ein MARK-Signal (RS232 ==> negative Spannung) anliegt, wird dies als BUSY interpretiert. Umgekehrt ist eine serielle Schnittstelle bei eingeschaltetem Hardware-Handshake nur dann READY, wenn CTS und DCD auf SPACE (RS232 ==> positive Spannung) stehen. Ein offener Eingang wird in der Regel als SPACE interpretiert. Es besteht jedoch die Gefahr des Übersprechens, falls z.B. bei einem Flachbandkabel ein unbeschalteter Eingang im Kabel angeschlossen ist! Bei der CPU II, dem EPC 1.6 und der DPS-IO ist DCD nicht beschaltet. Für den Hardware-Handshake wird deshalb in diesen Fällen ausschließlich CTS benutzt. Die Ausgangssignale RTS und DTR (DTR ist bei der CPU II, dem EPC 1.6 und der DPS-IO nicht beschaltet) werden immer auf SPACE-Pegel belassen.

Diese Art des Hardware-Handshakes ist für die meisten Drucker mit serieller Schnittstelle geeignet.

Bei den beiden seriellen Schnittstellen auf den Platinen der CPU 180x-Serie wird der Hardware-Handshake verschieden gehandhabt:

**SIOB:**

Die SIOB verhält sich wie oben beschrieben (DTR und DCD sind nicht beschaltet).

SIOA:

Der Input CTS (Clear To Send) wird immer als Handshake-Leitung benutzt. Falls an diesem Eingang ein MARK-Signal (RS232 ==> negative Spannung) anliegt, wird dies als BUSY interpretiert. Umgekehrt ist diese serielle Schnittstelle nur dann READY, wenn CTS auf SPACE (RS232 ==> positive Spannung) steht.

Bei ausgeschaltetem Hardware-Handshake wird der Ausgang RTS immer auf SPACE-Pegel belassen. Bei eingeschaltetem Hardware-Handshake wird dieser Ausgang nach dem Einlesen eines Zeichens auf MARK-Pegel umgeschaltet. Bei der nächsten auf das Einlesen folgenden Status-Abfrage an dieser Schnittstelle wird der Ausgang RTS dann wieder auf SPACE-Pegel zurückgesetzt.

**- Parität**

Paritätsprüfung wird bei Eingängen nicht durchgeführt. Für Ausgänge besteht jedoch die Möglichkeit Paritätsbits zu generieren. Die Parität kann auf gerade oder ungerade eingestellt oder ausgeschaltet werden.

**- Zeichenformat**

Folgende Übertragungsformate sind einstellbar: 5, 6, 7 oder 8 Datenbits (CPU 180x: nur 7 oder 8 Datenbits) und 1 oder 2 Stopbits. Das 8. Bit wird bei einem Übertragungsformat mit 8 Datenbits nicht gelöscht!

**- Timeout**

Für jedes physische Gerät kann eine TIMEOUT-Zeit von 1 bis 255 Sekunden vereinbart werden. Nach einer Zeichenausgabe muß ein physisches Gerät innerhalb der TIMEOUT-Zeit wieder bereit für die Annahme eines weiteren Zeichens werden. Falls das Gerät nach Ablauf dieser Zeit noch immer nicht bereit ist, wird dies auf der CONSOLE gemeldet (TIME OUT name ?). 'name' ist dabei der Name des betreffenden physischen Gerätes. Die möglichen Reaktionen des Anwenders auf diese Fehlermeldung sind im Kapitel 'CBIOS 1.8 Modul: BIOSKERN' beschrieben. Die TIMEOUT-Prüfung kann auch ausgeschaltet werden.

CONFIG kann für jedes physische Gerät unter CBIOS 1.8 die oben beschriebenen Eigenschaften einstellen. Darüber hinaus ist es noch möglich den Namen eines physischen Gerätes zu verändern.

Die Zuordnung von logischen Geräten zu physischen Geräten ist mit CONFIG 1.8 ebenfalls möglich. Außerdem kann CONFIG 1.8 den Dateityp für ausführbare Programme abändern (normalerweise 'COM'), den Dateityp für Prozeduren abändern (normalerweise 'SUB') und den Namen des Programmes ändern, das zum Ausführen einer Prozedur aufgerufen wird (normalerweise 'SUBMIT').

CONFIG kann auf 2 verschiedene Arten aufgerufen werden:

1. CONFIG parameterliste
2. CONFIG

Im ersten Fall erfolgt direkt die Einstellung der Schnittstellen gemäß der Parameterliste. Im zweiten Fall wird erst der augenblickliche Zustand der physischen Geräte angezeigt und anschließend die Eingabe der Parameterliste verlangt. Durch Eingabe von nur CARRIAGE RETURN oder ctrl C kann CONFIG an dieser Stelle abgebrochen werden.

#### Syntax der Parameterliste:

```
l_geraeteliste=p_geraeteliste(eigenschaftsliste) kommentar
oder
l_geraeteliste=NULL ; kommentar
oder
CCP:(ccpliste) kommentar
```

Statt runden Klammern sind auch eckige oder geschweifte Klammern zulässig. Eigenschaftsliste und Kommentar können entfallen. Falls kein Kommentar vorhanden ist, kann auf eine schließende Klammer verzichtet werden. Statt einer schließenden Klammer ist auch ein ';' erlaubt.

#### **l\_geraeteliste:**

Aufzählung von logischen Geräten, denen physische Geräte gemäß p\_geraeteliste zugeordnet werden sollen. Diese Zuordnung hat einen ausschließlichen Character, d.h. falls vorher andere Zuordnungen bestanden, so werden diese getrennt. Die Anweisung l\_geraeteliste=NULL bewirkt nur das Auftrennen von bestehenden Zuweisungen, ohne neue Zuordnungen zu treffen. Falls mehrere Geräte aufgezählt werden sollen, so sind die einzelnen Namen durch ', ' oder tab oder ' ' voneinander zu trennen.

Die folgenden 5 logischen Geräte werden hier unterstützt:

```
CONIN:          alternativ: TASTATUR, KEYBOARD, KBD:
CONOUT:         alternativ: BILDSCHIRM, SCREEN:
AUXIN:
AUXOUT:
LST:           alternativ: LISTOUT:, PRINTER, DRUCKER
```

Für CONIN: und CONOUT: ist auch die Zusammenfassung möglich:  
CONSOLE: alternativ: CON:

Für AUXIN: und AUXOUT: ist auch die Zusammenfassung möglich:  
AUXILIARY: alternativ: AUX:

Die l\_geraeteliste kann vollständig entfallen. In diesem Fall entfällt auch das Zeichen '=' zwischen l\_geraeteliste und p\_geraeteliste.



Beispiel für eine logische Geräteliste:

```
PRINTER,TASTATUR,AUX:
```

Es werden die Geräte LST:, CONIN:, AUXIN: und AUXOUT: aufgezählt.

#### **p\_geraeteliste:**

Aufzählung von physischen Geräten, die den logischen Geräten gemäß l\_geraeteliste zugewiesen bzw. denen Eigenschaften gemäß der Eigenschaftsliste zugeordnet werden sollen. Allen Geräten werden dabei die gleichen Eigenschaften zugeordnet. Es ist auch zwingend notwendig, daß bei allen Geräten aus der Geräteliste alle Eigenschaften aus der Eigenschaftsliste einstellbar sind. Physische Geräte werden mit Ihrem Namen aus der CBIOS-Gerätetabelle angesprochen. Falls mehrere Geräte aufgezählt werden sollen, so sind die einzelnen Namen durch ', ' oder tab oder ' ' voneinander zu trennen. Falls hinter einem Gerätenamen ein '/' angegeben wird, erhält das Gerät einen neuen Namen (steht direkt hinter dem '/'). Der Name eines physischen Gerätes darf höchstens 6 Zeichen lang sein und kein ':' oder Trennzeichen (tab, ' ', ', ' usw.) enthalten.

Beispiel für eine physische Geräteliste:

```
SIO1 SIO2/TERM,CEN/EPSON
```

Es werden die Geräte SIO1, SIO2, CEN aufgezählt. SIO2 wird gleichzeitig in TERM und CEN wird gleichzeitig in EPSON umbenannt.

#### **eigenschaftsliste:**

Aufzählung von Eigenschaften, die den physischen Geräten aus der p\_geraeteliste zugeordnet werden sollen. Eigenschaften werden durch tab, ' ' oder ', ' voneinander getrennt. Sie werden syntaktisch durch 'schlüsselwort=zuordnung' oder 'befehlscode' beschrieben.

Liste der möglichen Schlüsselworte:

BAUDRATE	alternativ: BAUD, BD, B
PROTOKOLL	alternativ: PROT, PR, XONXOFF, XON, XX, X
HANDSHAKE	alternativ: HAND, HND, H, HANDGEBEN
PARITAET	alternativ: PARITY, PARIT, PAR, PA, PY
DATENBIT	alternativ: DATABIT, DATEN, DATA, DAT, DA, D
STOPBIT	alternativ: STOPBITS, STOP, ST, S
TIMEOUT	alternativ: TIME, ZEIT, ZEIT AUS, T
CONIN:	alternativ: TASTATUR, KEYBOARD, KBD:
CONOUT:	alternativ: BILDSCHIRM, SCREEN:
AUXIN:	
AUXOUT:	
LST:	alternativ: DRUCKER, PRINTER, LIST:
CONSOLE:	alternativ: CON: für CONIN: und CONOUT:
AUXILIARY:	alternativ: AUX: für AUXIN: und AUXOUT:

Darüber hinaus sind noch weitere Abkürzungen möglich (ausprobieren!).

Im folgenden Text steht 'logdev:' stellvertretend für 'CONIN:', 'CONOUT:', 'AUXIN:', 'AUXOUT:', 'LST:', 'CON:' oder 'AUX:'.

Mögliche Zuordnungen zu den einzelnen Schlüsselwörtern:

<b>BAUDRATE=</b>	50	alternativ:	50BD,	50BAUD		
	75	alternativ:	75BD,	75BAUD		
	110	alternativ:	110BD,	110BAUD		
	134	alternativ:	134BD,	134BAUD,	135BD,	135BAUD
	150	alternativ:	150BD,	150BAUD		
	300	alternativ:	300BD,	300BAUD		
	600	alternativ:	600BD,	600BAUD		
	1200	alternativ:	1200BD,	1200BAUD		
	1800	alternativ:	1800BD,	1800BAUD		
	2400	alternativ:	2400BD,	2400BAUD		
	3600	alternativ:	3600BD,	3600BAUD		
	4800	alternativ:	4800BD,	4800BAUD		
	7200	alternativ:	7200BD,	7200BAUD		
	9600	alternativ:	9600BD,	9600BAUD		
	19200	alternativ:	19K2BD,	19K2BAUD,	19200BD,	19200BAUD

Diese Zuordnungen können auch ohne Schlüsselwort als 'befehlscode' geschrieben werden. Sie werden dann ebenfalls als Baudrate interpretiert.

<b>PROTOKOLL=</b>	NOXON	alternativ: NOX, NX, XOFF, OFF, AUS, INAKTIV
	XON	alternativ: X, ON, EIN, E, AN, AKTIV

Die Zuordnungen 'NOXON', 'NOX', 'NX' und 'XOFF' werden auch ohne Schlüsselwort als 'XON/XOFF-Protokoll aus' interpretiert. Die Zuordnungen 'XON' und 'X' werden auch ohne Schlüsselwort als 'XON/XOFF-Protokoll ein' interpretiert.

**HANDSHAKE=** AUS alternativ: OFF, INAKTIV  
 EIN alternativ: ON, E, AN, AKTIV

**PARITAET=** AUS alternativ: OFF, INAKTIV  
 EVEN alternativ: EV, GERADE  
 ODD alternativ: UNGERADE

**DATENBIT=** 5, 6, 7 oder 8

**STOPBIT=** 1 oder 2

**TIMEOUT=** AUS alternativ: OFF, INAKTIV, 0  
 1 bis 255 alternativ: 1S bis 255S

**logdev:=** AUS alternativ: OFF, INAKTIV  
 EIN alternativ: ON, E, AN, AKTIV

Falls der Name eines logischen Gerätes hier ohne folgendes '=' als 'befehlscode' geschrieben wird, so entspricht das der Eigenschaft 'logdev:=EIN'.

Es ist zu beachten, daß bei der Zuweisung bzw. Trennung von Zuweisungen innerhalb einer Eigenschaftsliste nur die physischen Geräte beachtet werden, die in der p\_geraeteliste aufgezählt wurden. Insbesondere bleiben solche Zuweisungen unberührt, die mit physischen Geräten bestehen, die nicht in der Geräteliste aufgezählt wurden.

Darüber hinaus sind bei einigen Zuordnungen gegenüber den oben aufgezählten noch weitere Abkürzungen möglich (ausprobieren!).

#### Liste der möglichen Befehlscodes:

**NOLIST** NOLIST ist ein Pseudobefehlscode, der lediglich bewirkt, daß die CONFIG-Tabelle auf dem Bildschirm nicht angezeigt wird. Dies ist z.B. sinnvoll, wenn CONFIG innerhalb einer SUBMIT-Prozedur mehrfach aufgerufen wird. Dieser Pseudobefehlscode ist auch innerhalb einer ccpliste erlaubt.

**logdev:** Dieser Befehl ist identisch mit dem Befehl 'logdev:=ein'.

Als Befehlscode gilt auch die Eingabe einer gültigen Baudrate ohne das Schlüsselwort 'BAUDRATE=' oder die unter 'PROTOKOLL=' beschriebene Art 'XON/XOFF' ein- bzw. auszuschalten ohne das Schlüsselwort 'PROTOKOLL=' zu benutzen.

### **ccpliste:**

CONFIG 1.8 kann den Dateityp für ausführbare Programme abändern (normalerweise 'COM'), den Dateityp für Prozeduren abändern (normalerweise 'SUB') und den Namen des Programmes ändern, das zum Ausführen einer Prozedur aufgerufen wird (normalerweise 'SUBMIT').

Durch die Eingabe von 'CCP:' anstelle von 'l\_geraeteliste=p\_geraeteliste' wird diese Eigenschaft von CONFIG 1.8 selektiert.

Die 'ccpliste' ist eine Aufzählung von Umbenennungen für den Dateityp der ausführbaren Programme, den Dateityp der Prozeduren und des Programmes, das zur Ausführung von Prozeduren aufgerufen wird. Syntaktisch werden diese Umbenennungen durch 'schlüsselwort=neuename' beschrieben.

### Folgende Schlüsselworte sind hier möglich:

COMTYP	alternativ: COMSPEC, COM, KOMMANDO, COMMAND
SUBTYP	alternativ: SUBCOMMAND, SUBSPEC, SUB, SUBKOM
SUBPRG	alternativ: SUBMIT, SUBNAME, SUBPROGRAMM

Darüber hinaus sind noch weitere Abkürzungen möglich (ausprobieren!).

### Die Schlüsselworte in der 'ccpliste' haben folgende Bedeutung:

**COMTYP:** Schlüsselwort zur Umbenennung des Dateitypes für ausführbare Programme (normalerweise 'COM').

**SUBTYP:** Schlüsselwort zur Umbenennung des Dateitypes für Prozeduren (normalerweise 'SUB').

**SUBPRG:** Schlüsselwort zur Umbenennung des Programmes, das zur Ausführung von Prozeduren aufgerufen wird (normalerweise 'SUBMIT').

### **kommentar:**

Ein beliebiger Text, der von CONFIG nicht beachtet wird.

Beispiele:

```
sio1/modem(300bd,nx,time=45,py=even,data=7,stop=2) ==> Modem
```

SIO1 erhält den neuen Namen 'MODEM'. Als Baudrate wird 300Baud eingestellt; XON/XOFF wird ausgeschaltet; Parität wird auf 'gerade' eingestellt und außerdem werden noch 7 Datenbits und 2 Stopbits vereinbart. Der Text hinter der schließenden Klammer ist ein Kommentar und wird von 'CONFIG' nicht beachtet.

```
sio2,sio4(xx=an,baudrate=19K2,daten=8,zeit=aus,par=inaktiv)
```

SIO2 und SIO4 erhalten die folgenden Eigenschaften:

XON/XOFF-Protokoll wird eingeschaltet. Die Baudrate wird auf 19200Baud eingestellt. Die Datenwortlänge wird mit 8Bit vereinbart. Die TIMEOUT-Überprüfung wird abgeschaltet. Die Paritätsgenerierung wird ausgeschaltet. Alle anderen Eigenschaften bleiben unverändert.

```
CONFIG ccp: (com=cqm,nolist)
```

Der Dateityp für ausführbare Programme wird auf 'CQM' geändert und das Listing von CONFIG auf dem Bildschirm unterdrückt.

```
CONFIG aux:=cen1(lst:=off,nol,t=40)
```

AUXIN: und AUXOUT: werden ausschließlich dem physischen Gerät CEN1 zugewiesen. Gleichzeitig wird eine evtl. bestehende Zuweisung LST: mit CEN1 aufgehoben, das Listing von CONFIG auf dem Bildschirm unterdrückt und Timeout für CEN1 auf 40 Sekunden gesetzt.

Die nachfolgenden Beispiele zeigen Original-Bildschirm Ausgaben von CONFIG:

CONFIG 1.800 -- <C> 1984/87 by Janich & Klass

Physische Gerate und ihre Eigenschaften:

E: Eingabe / A:Ausgabe

Gerat	E/A	CON:	AUX:	LST:	Baudrate	Xon/ Koff	Hand- shake	Paritaet	Zeichenformat	Zeit
VIDEO1	E/A	----	----	----	----	----	----	----	----	----
SIO1	E/A	----	E/A	----	9600	aus	aus	aus	8 Data 1 Stop	30s
SIO2	E/A	E/A	----	----	9600	aus	aus	aus	8 Data 1 Stop	30s
SIO3	E/A	----	----	----	9600fest	aus	aus	aus	8 Data 1 Stop	30s
SIO4	E/A	----	----	----	9600	aus	aus	aus	8 Data 1 Stop	30s
SL186	E/A	----	----	----	----	----	----	----	----	----
COM1	E/A	----	----	----	9600	aus	----	aus	8 Data 1 Stop	----
PCCON	E/A	----	----	----	----	----	----	----	----	----
CEN1	A	----	----	A	----	----	----	----	----	30s

Kommandotyp: COM SUB-Kommandotyp: SUB SUB-Kommandoname: SUBMIT

Eingabe: sio1/modem(baud=300,hnd=ein,par=odd,data=7,time=80s)

E: Eingabe / A:Ausgabe

Gerat	E/A	CON:	AUX:	LST:	Baudrate	Xon/ Koff	Hand- shake	Paritaet	Zeichenformat	Zeit
VIDEO1	E/A	----	----	----	----	----	----	----	----	----
MODEM	E/A	----	E/A	----	300	aus	aktiv	ungerade	7 Data 1 Stop	80s
SIO2	E/A	E/A	----	----	9600	aus	aus	aus	8 Data 1 Stop	30s
SIO3	E/A	----	----	----	300fest	aus	aus	aus	8 Data 1 Stop	30s
SIO4	E/A	----	----	----	9600	aus	aus	aus	8 Data 1 Stop	30s
SL186	E/A	----	----	----	----	----	----	----	----	----
COM1	E/A	----	----	----	9600	aus	----	aus	8 Data 1 Stop	----
PCCON	E/A	----	----	----	----	----	----	----	----	----
CEN1	A	----	----	A	----	----	----	----	----	30s

Kommandotyp: COM SUB-Kommandotyp: SUB SUB-Kommandoname: SUBMIT

Die Baudrate von SIO3 ist nicht einstellbar ('fest'). Trotzdem wird diese Baudrate in dem obigen Beispiel verändert, da die Baudrateneinstellung von SIO1 und SIO3 hardwaremäßig miteinander gekoppelt ist. SIO1 und SIO3 haben deshalb immer dieselbe Baudrate!

A>config ccp:(subtyp=cmd,subprg=sl186) ; CP/M 86 Programme mit SLAVE 186

E: Eingabe / A:Ausgabe

Gerat	E/A	CON:	AUX:	LST:	Baudrate	Xon/ Xoff	Hand- shake	Paritaet	Zeichenformat	Zeit
VIDEO1	E/A	----	----	----	----	----	----	----	----	----
MODEM	E/A	----	E/A	----	300	aus	aktiv	ungerade	7 Data 1 Stop	80s
SIO2	E/A	E/A	----	----	9600	aus	aus	aus	8 Data 1 Stop	30s
SIO3	E/A	----	----	----	300fest	aus	aus	aus	8 Data 1 Stop	30s
SIO4	E/A	----	----	----	9600	aus	aus	aus	8 Data 1 Stop	30s
SL186	E/A	----	----	----	----	----	----	----	----	----
COM1	E/A	----	----	----	9600	aus	----	aus	8 Data 1 Stop	----
PCCON	E/A	----	----	----	----	----	----	----	----	----
CEN1	A	----	----	A	----	----	----	----	----	30s

Kommandotyp: COM SUB-Kommandotyp: CMD SUB-Kommandoname: SL186

Falls mit SETDEF die Suchreihenfolge auf 'COM,SUB' vereinbart ist, wird nun nicht mehr automatisch bei Eingabe von 'name' das Programm 'name.COM' und anschließend die Prozedur 'name.SUB' gesucht und ggf. durch 'SUBMIT name' gestartet, sondern hier 'name.CMD' gesucht und durch 'SL186 name' gestartet. Dies kann mit dem Slave 186 unter der CP/M 86-Emulation sehr nützlich sein.

A>config auxin:=sio4(lst:=on,xon)

E: Eingabe / A:Ausgabe

Gerat	E/A	CON:	AUX:	LST:	Baudrate	Xon/ Xoff	Hand- shake	Paritaet	Zeichenformat	Zeit
VIDEO1	E/A	----	----	----	----	----	----	----	----	----
MODEM	E/A	----	A	----	300	aus	aktiv	ungerade	7 Data 1 Stop	80s
SIO2	E/A	E/A	----	----	9600	aus	aus	aus	8 Data 1 Stop	30s
SIO3	E/A	----	----	----	300fest	aus	aus	aus	8 Data 1 Stop	30s
SIO4	E/A	----	E	A	9600	aktiv	aus	aus	8 Data 1 Stop	30s
SL186	E/A	----	----	----	----	----	----	----	----	----
COM1	E/A	----	----	----	9600	aus	----	aus	8 Data 1 Stop	----
PCCON	E/A	----	----	----	----	----	----	----	----	----
CEN1	A	----	----	A	----	----	----	----	----	30s

Kommandotyp: COM SUB-Kommandotyp: CMD SUB-Kommandoname: SL186

Das obige Kommando weist das logische Gerät AUXIN: dem physischen Gerät SIO4 zu und löscht es gleichzeitig bei allen anderen physischen Geräten. Das logische Gerät LST: wird ebenfalls dem physischen Gerät SIO4 zugewiesen, aber bei den anderen physischen Geräten nicht gelöscht. Außerdem wird bei dem physischen Gerät SIO4 das XON/XOFF-Protokoll aktiviert, was dazu führt, daß auf dem logischen Gerät AUXIN: Daten verloren gehen, wenn gleichzeitig Druckerausgaben nach LST: erfolgen!.

A>config sio3,sio4(conout:,printer=off,stop=2

E: Eingabe / A:Ausgabe

Geraet	E/A	CON:	AUX:	LST:	Baudrate	Xon/ Xoff	Hand- shake	Paritaet	Zeichenformat	Zeit
VIDEO1	E/A	----	----	----	----	----	----	----	----	----
MODEM	E/A	----	A	----	300	aus	aktiv	ungerade	7 Data 1 Stop	80s
SIO2	E/A	E/A	----	----	9600	aus	aus	aus	8 Data 1 Stop	30s
SIO3	E/A	A	----	----	300fest	aus	aus	aus	8 Data 2 Stop	30s
SIO4	E/A	A	E	----	9600	aktiv	aus	aus	8 Data 2 Stop	30s
SL186	E/A	----	----	----	----	----	----	----	----	----
COM1	E/A	----	----	----	9600	aus	----	aus	8 Data 1 Stop	----
PCCON	E/A	----	----	----	----	----	----	----	----	----
CEN1	A	----	----	A	----	----	----	----	----	30s

Kommandotyp: COM SUB-Kommandotyp: CMD SUB-Kommandoname: SL186

Den beiden physischen Geraten SIO3 und SIO4 wird die Console-Ausgabe zugewiesen. Die Druckerausgabe wird gleichzeitig bei diesen beiden physischen Geraten geloscht und zusatzlich 2 Stopbits vereinbart. Alle anderen physischen Gerate werden nicht beeinflusst.

A>config con:=sio2

E: Eingabe / A:Ausgabe

Geraet	E/A	CON:	AUX:	LST:	Baudrate	Xon/ Xoff	Hand- shake	Paritaet	Zeichenformat	Zeit
VIDEO1	E/A	----	----	----	----	----	----	----	----	----
MODEM	E/A	----	A	----	300	aus	aktiv	ungerade	7 Data 1 Stop	80s
SIO2	E/A	E/A	----	----	9600	aus	aus	aus	8 Data 1 Stop	30s
SIO3	E/A	----	----	----	300fest	aus	aus	aus	8 Data 2 Stop	30s
SIO4	E/A	----	E	----	9600	aktiv	aus	aus	8 Data 2 Stop	30s
SL186	E/A	----	----	----	----	----	----	----	----	----
COM1	E/A	----	----	----	9600	aus	----	aus	8 Data 1 Stop	----
PCCON	E/A	----	----	----	----	----	----	----	----	----
CEN1	A	----	----	A	----	----	----	----	----	30s

Kommandotyp: COM SUB-Kommandotyp: CMD SUB-Kommandoname: SL186

Die CONSOLE wird dem physischen Gerat SIO2 zugewiesen und gleichzeitig bei allen anderen physischen Geraten geloscht.



A>config aux:=null

E: Eingabe / A:Ausgabe

Geraet	E/A	CON:	AUX:	LST:	Baudrate	Xon/	Hand-	Paritaet	Zeichenformat	Zeit
VIDE01	E/A	----	----	----	----	----	----	----	----	----
MODEM	E/A	----	----	----	300	aus	aktiv	ungerade	7 Data 1 Stop	80s
SI02	E/A	E/A	----	----	9600	aus	aus	aus	8 Data 1 Stop	30s
SI03	E/A	----	----	----	300fest	aus	aus	aus	8 Data 2 Stop	30s
SI04	E/A	----	----	----	9600	aktiv	aus	aus	8 Data 2 Stop	30s
SL186	E/A	----	----	----	----	----	----	----	----	----
COM1	E/A	----	----	----	9600	aus	----	aus	8 Data 1 Stop	----
PCCOM	E/A	----	----	----	----	----	----	----	----	----
CEN1	A	----	----	A	----	----	----	----	----	30s

Kommandotyp: COM SUB-Kommandotyp: CMD SUB-Kommandoname: SL186

Das logische Gerät AUX: wird bei allen physischen Geräten gelöscht und keinem neuen physischen Gerät mehr zugewiesen.

### Mögliche Fehlermeldungen von CONFIG 1.8

Fehler: 00 -- unbekannter Fehler

Diese Fehlermeldung darf nicht auftreten. Falls dies doch der Fall sein sollte, bitten wir Sie mit ctrl P den Drucker einzuschalten und dann diese Fehlermeldung erneut zu provozieren. Das Listing bitten wir an Ihren Händler zwecks Fehlersuche durch uns zu übersenden.

Fehler: 01 -- logisches Geraet: Geraetenname unbekannt

In der Geräteliste ist ein logisches Gerät aufgezählt worden, das mit diesem Namen für CONFIG unbekannt ist.

Fehler: 02 -- physisches Geraet: Geraetenname unbekannt

In der Geräteliste ist ein physisches Gerät aufgezählt worden, das in der Gerätetabelle des CBIOS nicht enthalten ist.

Fehler: 03 -- physisches Geraet: ":" in Geraetenname unzulässig

Durch einen ':' in einem Gerätenamen wird ein logisches Gerät gekennzeichnet. Bei einem physischen Gerät ist ein ':' im Gerätenamen deshalb unzulässig.

Fehler: 04 -- mehr als 6 Zeichen im Geräetenamen unzulässig

In der Gerätetabelle des CBIOS sind nur 6 Zeichen für einen Gerätenamen zugelassen. Hier wurde jedoch versucht, einen Gerätenamen mit mehr als 6 Zeichen zu vereinbaren.

Fehler: 05 -- dieses Trennzeichen hier unzulässig

Hinter einem '=' darf kein erneutes Trennzeichen stehen. '/' ist innerhalb einer Eigenschaftsliste nicht erlaubt. '/' ist hinter einem Gerätenamen nur einmal erlaubt. Eine öffnende Klammer ist erst erlaubt, wenn bereits ein Gerät vereinbart wurde. '=' ist in einer Geräteliste nicht erlaubt. Hinter dem Wort 'CCP:' muß eine öffnende Klammer folgen. Hinter dem Wort 'NULL' ist eine öffnende Klammer unzulässig.

Fehler: 06 -- Schlüsselwort unbekannt

Das Wort vor dem '='-Zeichen ist kein gültiges Schlüsselwort.

Fehler: 07 -- unbekannter Befehlscode

Es ist ein Befehlscode ohne '=' eingegeben worden, der in der Befehlscode-tabelle nicht gefunden werden kann.

Fehler: 08 -- unbekannte Baudrate

Die eingegebene Baudrate kann in der Baudratentabelle nicht gefunden werden.

Fehler: 09 -- keine Baudraten einstellbar

Bei mindestens einem physischen Gerät aus der Geräteliste sind keine Baudraten einstellbar.

Fehler: 10 -- Zeitkonstante zu gross (oder fehlerhaft)

Als Zeitkonstante sind nur Werte zwischen 0 und 255 möglich.

Fehler: 11 -- Paritätswert unbekannt

Parität kann nur ausgeschaltet oder mit 'EVEN' oder 'ODD' vereinbart werden.

Fehler: 12 -- Parität nicht einstellbar

Bei mindestens einem physischen Gerät aus der Geräteliste ist Parität nicht einstellbar.

Fehler: 13 -- nur 1 oder 2 Stopbits erlaubt

Hinter dem Schlüsselwort 'STOPBIT' darf nur 1 oder 2 stehen.

Fehler: 14 -- Stopbits nur bei seriellen Schnittstellen einstellbar

Mindestens ein physisches Gerät aus der Geräteliste ist keine serielle Schnittstelle.

Fehler: 15 -- nur 5, 6, 7 oder 8 Datenbits erlaubt

Hinter dem Schlüsselwort 'DATABIT' darf nur 5, 6, 7 oder 8 stehen.

Fehler: 16 -- Datenbits nur bei seriellen Schnittstellen einstellbar

Mindestens ein physisches Gerät aus der Geräteliste ist keine serielle Schnittstelle.

Fehler: 17 -- Handshake kann nur ein- oder ausgeschaltet werden

Hinter dem Schlüsselwort 'HANDSHAKE' darf nur 'EIN' oder 'AUS' stehen.

Fehler: 18 -- Handshake nicht einstellbar

Bei mindestens einem physischen Gerät aus der Geräteliste ist Handshake nicht einstellbar.

Fehler: 19 -- XON/XOFF-Protokoll kann nur ein- oder ausgeschaltet werden

Hinter dem Schlüsselwort 'PROTOKOLL' darf nur 'EIN' oder 'AUS' stehen.

Fehler: 20 -- XON/XOFF nicht einstellbar

Bei mindestens einem physischen Gerät aus der Geräteliste ist XON/XOFF nicht einstellbar.

Fehler: 21 -- dieses Schlüsselwort hier unzulässig

Schlüsselworte aus der Gruppe 'eigenschaftsliste' sind in der Gruppe 'ccpliste' und umgekehrt nicht zulässig.

Fehler: 22 -- kein Befehlscode mehr erlaubt

Innerhalb einer 'ccpliste' sind keine Befehlscodes mehr erlaubt. Ausgenommen davon ist nur der Pseudobefehl 'NOLIST'.

Fehler: 23 -- kein logisches oder physisches Geräet mehr erlaubt

Nach Eingabe von 'CCP:' ist keine Eingabe von logischen oder physischen Geräten mehr zulässig.

Fehler: 24 -- ":" in Programm-Name bzw. Programm-Typ unzulässig

Ein ':' in einem Programm-Name oder Programm-Typ würde ggf. zu einem 'hängenden' System führen.

Fehler: 25 -- mehr als 3 Zeichen fuer Programm-Typ unzulässig

Gemäß CP/M PLUS-Syntax kann ein Programm-Typ (=Datei-Typ) nur aus maximal 3 Zeichen bestehen. Hier wurde jedoch versucht, eine Programm-Type mit mehr als 3 Zeichen zu vereinbaren.

Fehler: 26 -- mehr als 8 Zeichen fuer Programm-Name unzulässig

Gemäß CP/M PLUS-Syntax kann ein Programm-Name (=Datei-Name) nur aus maximal 8 Zeichen bestehen. Hier wurde jedoch versucht, einen Programm-Namen mit mehr als 8 Zeichen zu vereinbaren.

Fehler: 27 -- Zuweisung zu einem logischen Eingabegeräet nicht erlaubt

Bei mindestens einem physischen Gerät aus der Geräteliste ist die Zuweisung zu einem logischen Eingabegerät deshalb nicht möglich, da dieses physische Gerät keine Eingabefähigkeit besitzt.

Fehler: 28 -- Zuweisung zu einem logischen Ausgabegeräet nicht erlaubt

Bei mindestens einem physischen Gerät aus der Geräteliste ist die Zuweisung zu einem logischen Ausgabegerät deshalb nicht möglich, da dieses physische Gerät keine Ausgabefähigkeit besitzt.

Fehler: 29 -- logische Geräete koennen nur ein- oder ausgeschaltet werden

Hinter den Schlüsselwörtern 'CON:', 'CONIN:', 'CONOUT:', 'AUX:' 'AUXIN:', 'AUXOUT:' und 'LST:' darf nur 'EIN' oder 'AUS' stehen.

Fehler: 30 -- keine logische Eingabe-Konsole mehr vorhanden

Das logische Gerät 'CONIN:' muß immer einem physischen Gerät zugewiesen sein. Deshalb wird der Befehl 'CONIN:=NULL' und alle Befehle, die das gleiche bewirken, unterbunden.

### 2.6 TERM (Terminal-Programm)

TERM ist ein Terminal-Programm zur Kommunikation mit fremden Rechnern. Alle Eingaben von dem logischen Gerät 'CONSOLE' werden an das logische Gerät 'AUXILIARY' ausgegeben. Gleichzeitig werden alle Eingaben von dem logischen Gerät 'AUXILIARY' an das logische Gerät 'CONSOLE' ausgegeben. Der Rechner, auf dem das Programm TERM läuft, wird somit zu einem Terminal für einen anderen Rechner.

Alle Eingaben von 'AUXILIARY' werden zwischengepuffert, so daß es auch möglich ist, an 'AUXILIARY' mit einer höheren Baudrate als an der 'CONSOLE' zu arbeiten (allerdings müssen dann in der Datenübertragung gelegentlich Pausen eintreten, damit der Datenpuffer wieder geleert wird). Bei Pufferüberlauf werden Daten überschrieben.

TERM kann vom Benutzer beendet werden durch Eingabe von ESC und anschließendem ctrl C. Falls ein ESCAPE an den anderen Rechner übertragen werden soll, ist eine zweimalige Betätigung der ESC-Taste notwendig. Falls nach ESC weder ctrl C noch ESC eingegeben wird, bleibt die Betätigung von ESC ohne Wirkung.

2.7 SIMU (EPROM-Simulator laden)

SIMU ist ein Dienstprogramm zur Benutzung des jk82 EPROM-Simulators unter CP/M PLUS.

Syntax: SIMU dateiname adresse

"dateiname" ist der Name des Programmes, das in den EPROM-Simulator geladen werden soll. Dieses Programm soll im Zielsystem ablaufen und darf maximal 8KByte lang sein (abhängig von der Länge des zu simulierenden EPROMs).

"adresse" ist die auf dem EPROM-Simulator eingestellte RAM-Startadresse (nur 1. Ziffer der Startadresse eingeben!).

SIMU schaltet den EPROM-Simulator in das Master-System, lädt das Programm in den EPROM-Simulator und schaltet diesen dann auf Slave-Betrieb um.

Hardwareanforderungen zur Benutzung des Programmes 'SIMU':

- Der EPROM-Simulator muß Rev. 2 oder neuer sein.
- In dem verwendeten CP/M PLUS-System muß eine beliebige RAMDISK vereinbart sein!
- Die Umschaltung zwischen MASTER und SLAVE wird von SIMU per Software vorgenommen; ein Hardware-Umschalter darf nicht angeschlossen werden!
- Der EPROM-Simulator ist wie folgt zu jumpern:

IO-Adresse: 043H (falls mehrere EPROM-Simulatoren benutzt werden sollen, so sind alle auf die gleiche IO-Adresse einzustellen!)

BANK-Adresse: 0FH (bei mehreren EPROM-Simulatoren ist auch diese Adresse bei allen gleich einzustellen!)

RAM-Startadresse: eine von den 8 möglichen Startadressen (J2) (bei mehreren EPROM-Simulatoren im System muß diese Adresse unbedingt VERSCHIEDEN eingestellt werden!)

Auf der Adresse 0F000H darf keine RAM-Karte im System vorhanden sein!

Beispiele für die Benutzung des Programmes 'SIMU':

SIMU VIDE01.COM 0

In den EPROM-Simulator wird das Programm VIDE01.COM geladen; der EPROM-Simulator ist auf die RAM-Startadresse 0 eingestellt.

SIMU CHAR.DAT 2

In den 2. EPROM-Simulator wird die Datei CHAR.DAT geladen; der 2. EPROM-Simulator ist auf die RAM-Startadresse 2 eingestellt.

Auf die obige Art wurde die Firmware zur jk82 VIDEO I und der Zeichensatz zur VIDEO I getestet.

Mögliche Fehlermeldungen des Programmes 'SIMU':

**\*\*\*\*\* Ungeeignetes Betriebssystem \*\*\*\*\***

==> Kein CP/M PLUS oder fremdes CP/M PLUS.

**\*\*\*\*\* Keine RAMDISK im System implementiert \*\*\*\*\***  
**\*\*\*\*\* Ungeeignetes Betriebssystem \*\*\*\*\***

==> Zum Laden des EPROM-Simulators wird die RAMDISK-Write-Routine des CBIOS benutzt. Es ist deshalb notwendig, daß in dem benutzten Betriebssystem auch eine RAMDISK vereinbart wurde.

**\*\*\*\*\* Startadresse des EPROM-Simulators nicht angegeben \*\*\*\*\***

==> Syntaxfehler beim Aufruf des Programmes.

**\*\*\*\*\* kein Filename angegeben \*\*\*\*\***

==> Syntaxfehler beim Aufruf des Programmes.

**\*\*\*\*\* File nicht gefunden \*\*\*\*\*     Filename: name**

==> Die beim Aufruf angegebene Datei konnte auf dem zugehörigen Laufwerk nicht gefunden werden.

**\*\*\*\*\* Lesefehler \*\*\*\*\***

==> Beim Lesen der Datei wurde vom BDOS ein Fehler zurückgemeldet.

2.8 EprCopy (Erstellen einer EPROM-DISK)

Der Einsatz einer EPROM-Disk im CP/M PLUS fordert eine bestimmte Datenstruktur im EPROM. Es muß wie bei jedem anderen Laufwerk ein Inhaltsverzeichnis existieren und die Daten müssen in der CP/M-Blockstruktur verteilt sein.

Zum Programmieren der EPROMs ist es nützlich, wenn die Daten in mehreren Programmiermodulen vorliegen, deren Größe der Kapazität eines EPROMs entspricht.

Soll das Betriebssystem von einer EPROM-Disk geladen werden, so muß als erster Eintrag im Inhaltsverzeichnis ein bestimmtes Label existieren, ein DIR-Eintrag, bei dem die ersten 16Byte die Form

**20h, 'MEMDISKtgbd', 1, 0, 0, 0**

besitzen.

t:	Speicher-Disk-Typ	('0', '2', '5' oder '7')
g:	EPROM-Größe	('0', '1', '2' usw.)
		nur bei seitenadressierten EPROMs
b:	Blockgroesse	('0', '1', '4', '5')
d:	DIR-Einträge	('0', '1', '2', '3')

(vgl. Kapitel 4.1 über den Urlader 1.7)

EprCopy ist ein Programm, das die oben aufgeführten Arbeiten erledigt.

**Bedienung des Programmes**Programm aufrufen

Das Programm fragt nach der Anzahl der EPROMs, aus denen die EPROM-Disk aufgebaut werden soll.

Anzahl der EPROMs eingeben

Das Programm fragt nach der Größe eines EPROMs in KByte.

Größe eingeben

z.B. '64' für ein EPROM vom Typ 27512 bzw. 27513

Das Programm fragt nach der EPROM-Type.

Typ eingeben

z.B. 'S' für 27513.

Das Programm fragt nach der Zieldatei

Name der Zieldatei eingeben

(maximal 8 Buchstaben) ohne '.' und ohne Typ



Das Programm fragt nach den Dateinamen.

### Dateinamen eingeben

ein Dateiname pro Zeile

Dateien mit SYS-Attribut mit Parameter ÄRÜ angeben. Das Ende mit einer leeren Eingabe (nur <cr>) anzeigen.

EprCopy meldet fehlerhafte Eingaben sofort, nicht existierende Dateien erst, nachdem der letzte Dateiname eingegeben worden ist.

EprCopy zeigt nach der letzten Eingabe alle gültigen Dateinamen noch einmal an.

EprCopy zeigt ein Menue an, und verlangt vom Benutzer einen Befehl für die weiteren Arbeitsgänge.

Sollen noch weitere Dateien verarbeitet oder andere aus der Verarbeitung wieder herausgenommen werden, so ist das mit den Kommandos 'E' bzw. 'L' möglich. Das Kommando 'K' startet den Kopiervorgang, d.h. es werden jetzt die Programmiermodule erzeugt.

### Parameter bei der Eingabe der Dateinamen

Bei der Eingabe der Dateinamen können Parameter angegeben werden:

```
R  auch Dateien mit gesetztem Sys-Flag kopieren
A  nur Dateien mit nicht gesetztem Archiv-Flag kopieren
   Archiv-Flag setzen
Gn Datei von USER n holen (n= 0...15)
Fm Dateien mit File-Attribut kopieren (m=1..4)
   (vgl. CP/M-Programmers-Guide: 2.3.4 File Attributes)
```

Die Parameter müssen durch eine eckige Klammer bzw. einem 'Ä' vom Dateinamen abgetrennt werden.

### Erzeugte Dateien

EprCopy erzeugt Dateien mit dem eingegebenen Namen der Zieldatei und dem Datei-Typ 'Onm'. 'nm' sind zwei Ziffern, mit denen die Dateien durchnummeriert sind. Die erste Datei wäre z.B. 'EPROM.000', sie enthält u.a. das Inhaltsverzeichnis der EPROM-Disk.

Jede Datei stellt ein Programmiermodul für ein EPROM dar.

### Beispiel

Es soll eine EPROM-Disk mit 512KByte auf einer CMOS/EPROM-Karte erzeugt werden. Es werden EPROMs vom Typ 27512 verwendet. Insgesamt werden vier EPROMs eingesetzt.

Programmlauf:

(unterstrichene Textstellen sind Eingaben des Benutzers)

15H>eprcopy

Programm zum Erstellen von Programmiermodulen  
 fuer eine EPROM-Disk  
 Version 1.45 22.05.87  
 (c) 1987 by Janich & Klass

Anzahl der EPROMs? 4

Kapazitaet e i n e s EPROMs (in KByte)? 64

Lineares (z.B. 27512) EPROM (L)  
 seitenadressiertes (z.B. 27513) EPROM (S) 1

Name fuer die Zieldatei(en): eproms

Program EprCopy - User: 15 Disk: H:

Bitte die Dateinamen eingeben!

? cpm3.sysÄrg0

? ccp.comÄrg0

? eprcopy.com

?

Bitte warten!

Anzahl: 3

Name	Bytes	Attribute	Name	Bytes	Attribute
OH:CPM3	SYS	36K Sys	OH:CCP	COM	4K Dir
15H:EPRCOPY	COM	28K Dir			

Daten der Eprom-Disk

Dateien: 3 frei: 59 Eintraege  
 belegt: 68 KByte frei: 186 KByte

-----  
 Kopierlauf starten (K) Kopierlauf abbrechen (A)  
 Dateinamen anzeigen (D) Dateinamen loeschen (L)  
 zus. Dateinamen eintragen (E) ? k

kopiere Datei CPM3 SYS  
 kopiere Datei CCP COM  
 kopiere Datei EPRCOPY COM

Eine solche Platzverschwendung macht das Programm nicht mit. Statt der gewünschten Programmiermodule für vier EPROMs erzeugt es nur zwei:

EPROMS.000 und EPROMS.001

Für diese EPROM-Disk reichen also zwei EPROMs vom Typ 27512.

### 2.9 Das Kopierprogramm MSN

Unter dem Betriebssystem CP/M PLUS ist es möglich, eine Vielzahl verschiedener Diskettenformate zu verarbeiten. Voraussetzung ist lediglich, daß die Daten auf der Diskette entsprechend den CP/M-Konventionen abgespeichert sind.

Eine nicht CP/M-Struktur besitzen Disketten, die unter dem Betriebssystem MS-DOS der Firma Microsoft bzw. PC-DOS der Firma IBM beschrieben worden sind. Ähnliches gilt auch für das Betriebssystem TOS der Firma ATARI. Diese Betriebssysteme werden im folgenden - zur Unterscheidung vom CP/M - einheitlich mit DOS bezeichnet.

Mit dem Programm MSN ist es möglich, Daten zwischen den verschiedenen Rechner- bzw. Betriebssystemenarten auszutauschen.

- MSN kopiert Dateien von Disketten mit Fremdformaten (Nicht-CP/M-Formate) auf CP/M-Disketten und umgekehrt.

#### Aufruf von MSN

Das Programm MSN wird im Normalfall ohne Parameter aufgerufen.

```
A>msn
```

Es kann auch der Parameter 'F' angegeben werden, falls keine brauchbare Diskette mit DOS-Format zur Verfügung steht, und erst eine erzeugt werden soll:

```
A>msn f
```

In diesem Fall wird sofort der Befehl 'FORMAT' ausgeführt.

Wird der Parameter 'F' nicht angegeben, fragt MSN nach dem Aufruf nach dem Laufwerk, in dem sich die DOS-Diskette befindet.

Es ist nun der Kennbuchstabe (A ... P) für das Laufwerk einzugeben und die RETURN-Taste zu betätigen.

Das Programm meldet sich mit einem Prompt und wartet auf eine Eingabe. Das Prompt ähnelt dem CP/M-Prompt, in dem Laufwerksbezeichnung und USER-Nummer angegeben sind. Zur Unterscheidung ist die Laufwerksangabe mit einem Kleinbuchstaben gemacht worden.

**Befehlsübersicht:**

Innerhalb der Umgebung, die das Programm MSN schafft, sind die folgenden Befehle (und nur diese) möglich:

DIR	Inhaltsverzeichnis anzeigen
PIP	Datei(en) kopieren
CD	Inhaltsverzeichnis wechseln
RD	Inhaltsverzeichnis löschen
MD	Inhaltsverzeichnis erzeugen
USER	CP/M-User wechseln
ERA	Datei löschen
TYPE	Dateiinhalte auf dem Bildschirm darstellen
DRIVE	Laufwerk anwählen
FORMAT	Diskette formatieren
QUIT	Programm MSN verlassen

DIR: Das Kommando entspricht dem CP/M-Dir. Als Parameter sind Dateinamen (mit '?' bzw. '\*') und Laufwerksangaben (z.B. 'A:') erlaubt.

Bsp.: DIR \*.FOR

PIP: Das Kommando erwartet in derselben Zeile die Angabe von Quell- und Zieldatei. Die Angabe von Gerätenamen (CON:, LST:, AUX: usw.) ist nicht möglich. Ein Kopieren findet nur zwischen unterschiedlichen Disketten(-formaten) statt. Zum Kopieren von CP/M-Disketten auf CP/M-Disketten sollte die gleichnamige Funktion des CP/M PLUS verwendet werden.

Bsp.: PIP c:=\*.\*

CD: Auf der DOS-Diskette das Inhaltsverzeichnis wechseln (change directory).

Bsp.: CD ..

RD: Ein DOS-Untereinhaltsverzeichnis (subdirectory) löschen (remove directory).

Bsp.: RD xyz

MD: Ein DOS-Untereinhaltsverzeichnis erzeugen (make directory).

Bsp.: MD sdir

USER: Auf dem CP/M-Medium die USER-Nummer ändern. Dieser Befehl hat keine Auswirkung auf das Arbeiten mit dem DOS-Medium.

Bsp.: USER 5:

ERA: Datei auf dem DOS-Medium löschen. Für das Löschen von Dateien auf einem CP/M-Medium sollte der gleichnamige Befehl des Betriebssystems verwendet werden.

Bsp.: ERA \*.\*

TYPE: Den Inhalt einer Textdatei (vom DOS-Medium) auf den Bildschirm ausgeben. Soll der Inhalt einer CP/M-Datei angezeigt werden, so sollte der gleichnamige Befehl des Betriebssystems verwendet werden.

Bsp.: TYPE datei.txt

DRIVE: DOS-Diskette neu anwählen, oder anderes Laufwerk mit DOS-Diskette anwählen. Dem Befehl ist als Parameter die Laufwerkskennung (A:, B:, ... P:) anzugeben.

Bsp.: DRIVE c:

FORMAT FORMAT formatiert eine Diskette. FORMAT fragt zunächst nach dem Laufwerk in dem formatiert werden soll. Anschließend kann der Bediener eines von zur Zeit vier Formaten auswählen. Nach einer Kontrollabfrage, ob auch wirklich formatiert werden soll, beginnt das Programm mit der Formatierung der Diskette.

Das Verfahren setzt voraus, daß bei Programmaufruf (Aufruf von MSN) schon eine formatierte Diskette in einem Laufwerk bereitliegt (Frage nach Laufwerk). Diese Einschränkung läßt sich umgehen, wenn sofort nach dem Aufruf von MSN der Format-Befehl ausgeführt wird. Erreicht wird dies mit dem Parameter 'F', der beim Aufruf von MSN angegeben werden kann.

QUIT: Aufruf ohne Angabe eines Parameters. Beenden des Programmes MSN.

**2.10 DFE**

DFE ist ein Dienstprogramm zum Bearbeiten der Datei DISKFORM.DAT. Es ermöglicht das Entfernen, Verdecken (unsichtbarmachen), Einlesen aus anderen DISKFORM.DAT - Dateien und Umsortieren von Formaten.

Hat die Datei einen anderen Namen als DISKFORM.DAT, so ist er beim Programmstart als Parameter anzugeben. Nach dem Start wird die Datei in den Speicher geladen und jeder Formateintrag auf seine Gültigkeit überprüft. Nach der Ausgabe der Versionsnummer und dem Zeitpunkt der letzten Änderung können die Funktionen gewählt werden.

Änderungen an Formaten die dem Programm als 'gesperrt' bekannt sind, sind nicht möglich. Dies ist wichtig wenn Formate im Batch gewählt werden, da sich die Ordnungsnummern beim Umsortieren usw. sonst ändern. Durch Voreinstellung gesperrt sind die Formate 1..4 und 26. Zur Änderung der Sperrliste steht die Funktion (9) zur Verfügung.

Folgende Funktionen sind möglich:

- (?) **Liste**                    Formate listen.  
 Alle aufgedeckten Formate werden auf dem Bildschirm gelistet. Nach jeder Bildschirmseite stoppt die Ausgabe; die Eingabe von <RETURN> setzt die Liste fort, '.' bricht die Liste ab.
- (0) **loeschen**                Formate werden vollständig entfernt.  
 Ein existierendes, nicht gesperrtes Format wird durch Eingeben der Ordnungsnummer gelöscht. Bei '?' werden aufgedeckte, nicht gesperrte Formate zum Löschen angeboten.  
 '.' beendet die Funktion 'loeschen'.
- (2) **verdecken**              Formate erscheinen nicht mehr in der Liste; die Daten sind aber noch vorhanden.  
 Ein aufgedecktes, nicht gesperrtes Format wird durch Eingeben der Ordnungsnummer verdeckt. Bei '?' werden entsprechende Formate zum Verdecken angeboten.  
 '.' beendet die Funktion 'verdecken'.
- (3) **aufdecken**              Verdeckte Formate werden wieder in die Liste aufgenommen.  
 Ein verdecktes, nicht gesperrtes Format wird durch Eingeben der Ordnungsnummer aufgedeckt. Bei '?' werden entsprechende Formate zum Aufdecken angeboten.  
 '.' beendet die Funktion 'aufdecken'.

- (4) **verschieben**      Formate werden auf andere Ordnungsplätze verschoben.  
Zuerst wird durch Eingeben der Ordnungsnummer oder mit Hilfe von '?' das Format bestimmt, das verschoben werden soll. Dann muß die Ordnungsnummer eingegeben werden, auf der das Format abgelegt werden soll; Mit '?' kann man freie Plätze finden, mit <RETURN> das Format auf dem ersten freien Platz ablegen und mit '.' diesen Verschiebevorgang abbrechen.  
'.' beendet die Funktion 'verschieben'.
- (5) **einlesen**          Formate aus einer anderen Datei einlesen.  
Nach der Eingabe des Dateinamens wird die Versionsnummer und der letzte Änderungszeitpunkt dieser Datei ausgegeben. Die weitere Bedienung entspricht der Funktion (4), erweitert um den Befehl 'F'. Hier werden die Formate wie bei '?' zum Einlesen angeboten, jedoch nur solche, deren Datenkonfiguration in der aktuellen Datei noch nicht existieren.
- (6) **speichern**          Mit dieser Funktion kann eine modifizierte Kopie der Formate - Datei erzeugt werden, ohne die eigene Datei zu verändern.
- (7) **sortieren**          Die Formate werden nach ihrem Namen alphabetisch sortiert, wenn sie nicht gesperrt sind.
- (8) **Gleiche Einträge suchen**  
Formate deren Daten übereinstimmen werden auf den Bildschirm gelistet. Nach einer Bildschirmseite stoppt die Ausgabe und wird nach Eingabe von <RETURN> fortgesetzt. Die Funktion kann jederzeit durch Eingabe von '.' abgebrochen werden.
- (9) **sperrten/entsperren**  
Durch Eingabe der Ordnungsnummer eines existierenden Formates wird ein freigegebenes Format gesperrt bzw. ein gesperrtes Format entsperrt. Mit '?' werden die gesperrten Formate aufgelistet.  
'.' beendet die Funktion 'sperrten/entsperren'.
- (X) **beenden**            Die Bearbeitung beenden.  
Wurden Änderungen durchgeführt, kann die bearbeitete Datei abgespeichert werden. Mit '.' kann die Funktion verlassen werden, 'N' beendet das Programm ohne Abspeicherung. Bei 'J' wird die alte Datei mit den Extension '.BAK' gesichert. Die Versionsnummer der neuen Datei wird incrementiert, durch Systemdatum und -uhrzeit ergänzt und ausgegeben. Die Datei wird unter dem Bearbeitungsnamen abgespeichert.

Mögliche Fehlermeldungen des Programmes 'DFE':

**\*\*\*\*\* Ungeeignetes Betriebssystem \*\*\*\*\***

==> Kein jk82 CP/M PLUS.

**\*\*\*\*\* Kein Speicherplatz \*\*\*\*\***

==> Es ist nicht genügend Speicherplatz vorhanden.

**\*\*\*\*\* Datei Format-los \*\*\*\*\***

==> Es gibt kein gültiges Format in der Datei.



### 3.1 ZEIT.RSX (Datum/Zeit auf Programmlistings drucken)

ZEIT.RSX ist eine Systemerweiterung zum Drucken von Datum und Uhrzeit auf jeder Seite eines Programmlistings. Dadurch wird es sehr einfach möglich, mehrere Listings von einem Programm zu unterscheiden.

ZEIT.RSX kontrolliert alle Ausgaben an den Drucker (LST:). In Kopfzeilen (das sind solche Zeilen, die mit einem Formfeed beginnen) wird an das Ende der Zeile die Datum/Zeit-Information angefügt.

ZEIT.RSX ist für folgende Dienstprogramme geeignet:

MAC, RMAC, XREF, M80 (microsoft)

Falls andere Dienstprogramme ebenfalls Kopfzeilen (mit Formfeed) an den Drucker ausgeben, kann ZEIT.RSX dafür ebenfalls geeignet sein (ausprobieren).

ZEIT.RSX kann durch folgenden Befehl an das jeweilige Dienstprogramm angebunden werden:

GENCOM name ZEIT

'name' ist der Name des jeweiligen Dienstprogrammes.

Ausgaben, die nicht direkt auf den Drucker erfolgen, können von diesem RSX nicht mit Datum und Uhrzeit versehen werden.

Bitte beachten Sie, daß mit einem RSX nur BDOS-Aufrufe abgefangen werden können, nicht jedoch direkte BIOS-Aufrufe. Druckerausgaben, die über direkte BIOS-Aufrufe erfolgen, können deshalb von diesem RSX ebenfalls nicht mit Datum und Uhrzeit versehen werden.

3.2 DATUM.REL

DATUM.REL stellt Datum und Uhrzeit in druckaufbereiteter Form bereit. Es kann in Anwenderprogramme eingebunden werden.

**ENTRY-Name:** DATUM

**Eingabeparameter:** HL - Zeiger auf einen 8 Byte langen Datenbereich  
DE - Zeiger auf einen zweiten 8 Byte langen Datenbereich

**Ausgabe:** In den durch HL adressierten Datenbereich wird das Datum und in den durch DE adressierten Datenbereich die Uhrzeit in druckaufbereiteter Form eingetragen.

**Benutzte Register:** AF, BC, DE, HL

**Bemerkung:** Es wird eine Z80-CPU vorausgesetzt!  
Die Parameterübergabe ist kompatibel zu FORTRAN-80 von microsoft.

1. Beispiel -- Benutzung dieses Moduls in einem Assemblerprogramm:

```

        title    'Datum und Uhrzeit auf der Console ausgeben'
        extrn    datum            ; Datum/Zeit-Modul

prtstr  equ     9                ; BDOS-Funktion Print String
bdos    equ     5                ; BDOS-Funktionsaufruf

cr      equ     0dh              ; Carriage Return
lf      equ     0ah              ; Line Feed

        lxi     h,date           ; Datumsfeld
        lxi     d,time           ; Uhrzeitfeld
        call    datum            ; Datum und Uhrzeit eintragen

        lxi     d,msg            ; Ausgabezeile
        mvi     c,prtstr         ; Print String
        call    bdos             ; Datum/Uhrzeit auf der Console ausgeben
        jmp     0                ; Warmstart

msg:    db      cr,lf,'Datum: '
date:   ds      8
        db      '    Uhrzeit: '
time:   ds      8
        db      cr,lf,'$'

        end

```

Nach Aufruf des fertigen Programmes erfolgt z.B. folgende Ausgabe auf die Console:

Datum: 23.07.84    Uhrzeit: 15:26:21

2. Beispiel -- Benutzung dieses Moduls in einem FORTRAN-Programm:

Das gleiche Beispiel wie oben sieht in FORTRAN-80 so aus:

```
      program test
      double precision date,time
      call datum (date,time)
      write (1,100) date,time
100   format (1x,'Datum: ',a8,'   Uhrzeit: ',a8)
      end
```

Ausgabe auf die Console:

Datum: 23.07.84    Uhrzeit: 15:30:44

4.1 Urlader Version 1.700

Zu diesem CP/M PLUS existieren zur Zeit die folgenden Urlader-Versionen:

1. LDR1700.COM für CPU I, HKM-CPU, CPU II oder CPU III (8KByte).
2. LDR1703.COM wie 1. - kann nicht von einer XEBEC-Winchester booten, dafür aber von einer EPROM-Disk (8KByte).
3. LDR11700.COM für EPC 1.6 (8KByte).
4. LDRE1601.COM für EPC II (8KByte).
5. LDR1710.COM für CPU 1801 (32KByte).
6. LDR1720.COM für CPU 1802 (32KByte).

Der Urlader besteht im wesentlichen aus einem Laderprogramm, einem Lader-BDOS und einem Lader-BIOS. Das BIOS entspricht im wesentlichen dem CBIOS für das CP/M PLUS (wird aus den gleichen Modulen erstellt). Der Urlader ist komplett im EPROM untergebracht; somit existiert auf den Systemspuren einer Diskette nur der Diskettenbeschreibungsblock. Das System befindet sich in der Datei CPM3.SYS; die nichtinstallierten Systeme in den Dateien CPM31.SYS, CPM32.SYS, CPM33.SYS, CPM34.SYS, CPM35.SYS oder CPM36.SYS.

Funktionsweise:

Der Urlader läuft auf den folgenden Systemkonfigurationen:

1. CPU II, FDC 5, FDC 8/5, FDC II, VIDEO I, SLAVE 186/GRAFIK 187, XEBEC, HDC 5, EPROM-Disk, CRAM-Disk
2. HKM-CPU oder CPU I, FDC 5, FDC 8/5, FDC II, VIDEO I, SLAVE 186/GRAFIK 187, XEBEC, HDC 5, RAE3 oder RAE4, EPROM-Disk, CRAM-Disk
3. CPU III, FDC 5, FDC 8/5, FDC II, VIDEO I, P/SIO, SLAVE 186/GRAFIK 187, XEBEC, HDC 5, EPROM-Disk, CRAM-Disk
4. EPC II, HDC 5, (FDC II, XEBEC)
5. EPC 1.6, HDC 5, VIDEO I, SLAVE 186/GRAFIK 187, EPROM-Disk, CRAM-Disk
6. CPU 1801, FDC 5, FDC II, VIDEO I, SLAVE 186/GRAFIK 187, XEBEC, HDC 5, EPROM-Disk, CRAM-Disk
7. CPU 1802, FDC 5, FDC II, VIDEO I, SLAVE 186/GRAFIK 187, XEBEC, HDC 5, EPROM-Disk, CRAM-Disk

Statt FDC 5 und P/SIO ist auch eine DPS-IO-Platine möglich.  
Statt FDC II ist auch eine FDC III-Platine möglich.

Benötigt werden zum Betrieb immer mindestens 128KByte RAM, eine Konsole und wahlweise eine doppelseitige 5,25" oder 8" Floppy oder eine Winchester oder eine EPROM-/CRAM-Disk.

Bei Einsatz der VIDEO I oder des SLAVE 186/GRAFIK 187 startet das System ohne Eingriff von außen, bei Anschluß eines Terminals ist (bei einem nicht modifiziertem Lader) auf dem Terminal einmal die Leertaste zu betätigen.

Beim EPC II wird die Baudrate der seriellen Tastatur im Lader fest eingestellt.

Falls die Leertaste betätigt werden muß, berechnet der Urlader die Baudrate des Terminals (1200, 2400, 4800, 9600 oder 19200 Baud bzw. 4800, 9600 oder 19200 Baud für CPU 1801 / CPU 1802).

Er berechnet immer den Systemtakt (4, 5, 6 oder 8 MHz) und sucht ein Laufwerk, von dem er das System laden kann.

### Hochfahren von einer EPROM- oder CRAM-Disk

CP/M plus benötigt zum Hochfahren einen Massenspeicher, von dem das Betriebssystem (Datei CPM3.SYS) geladen wird. Ein solcher Massenspeicher wird in der CP/M-Terminologie als Laufwerk bezeichnet; es handelt sich gewöhnlich um einen Magnetplattenspeicher: eine Floppy-Disk oder Hard-Disk.

In manchen Anwendungen soll das Betriebssystem jedoch nicht von einem Magnetplatten-Laufwerk geladen werden, z.B. wenn eine kurze "Hochlaufzeit" gefordert ist oder das System ganz ohne mechanische Massenspeicher auskommen soll.

Zu diesem Zweck kann der Urlader das Betriebssystem auch von einer EPROM-Disk oder einer CMOS-RAM-Disk laden (Ausnahmen: LDR1700 und LDRE1700). Diese Memorydisks können z.B. durch eine Bytewide- oder eine ECRAM-Karte realisiert werden; auch von der EPROM- und CRAM-Disk der CPU 1802 kann gebootet werden.

Das Laden des Betriebssystems über FDC2/FDC3 und XEBEC-Controller von einer Hard-Disk ist mit dem Urlader LDR1703 nicht möglich.

### Reihenfolge beim Suchen:

1. Konsole: Der Urlader sucht zyklisch nach einer Konsole. Zuerst wird die VIDEO I abgefragt; falls nicht vorhanden, wird nach einem SLAVE 186/GRAFIK 187 gesucht. Ist dieser nicht vorhanden, wird noch die serielle I/O-Schnittstelle der CPU-Karte SIOB (oder falls CPU III: die SIO2 der P/SIO Karte) abgefragt. Wenn bis dahin die Leertaste noch nicht betätigt wurde, beginnt der Suchzyklus von vorne.  
Ausnahme: EPC II, hier kann nur die serielle Tastatur und der eigene Bildschirm als Boot-Konsole benutzt werden.
2. Laufwerke: Ist eine gültige EPROM-/CRAM-Disk im System vorhanden (s.u.), so wird von dieser gebootet; andernfalls wird nach einem Winchester-Controller gesucht. Falls kein Winchester-Controller existiert, werden die Floppycontroller (soweit vorhanden) zyklisch nach einem Laufwerk mit eingelegter Diskette abgesucht. Dabei gilt folgende Reihenfolge: FDC II (5,25"), FDC II (8"), FDC 5 (5,25"), FDC 8/5 (8") - jeweils Laufwerk 0; danach die gleichen Controller jeweils Laufwerk 1. Bei den CPU's 1801 und 1802 wird zuallererst nach einem Laufwerk am eigenen Controller gesucht; EPC II und EPC 1.6 können nur von der Winchester oder von Floppy-Laufwerken an ihrem eigenen Controller booten (EPC 1.6 zusätzlich von einer EPROM-/CRAM-Disk)! Vom ersten Laufwerk mit einer gültigen Diskette (d.h. einer Diskette mit einem

gültigen Beschreibungsblock) wird gebootet (notfalls Tür öffnen!). Dieser Suchvorgang kann einige Zeit dauern! Im Urlader können nur interne Disketten mit Beschreibungsblock verarbeitet werden.

Falls eine angeschlossene Winchester oder eine EPROM-/CRAM-Disk übergangen werden soll, ist sofort nach den ersten Meldungen des Urladers die ESCAPE-Taste zu betätigen.

Eine Winchester muß als ersten Directory-Eintrag `.heads..` enthalten; sonst kann sie nicht zum Laden benutzt werden.

Eine EPROM- oder CRAM-Disk muß mit dem Disk-Label `MEMDISK` versehen sein und die Datei `CPM3?.SYS` enthalten, damit von ihr gebootet werden kann. Der spezielle Aufbau des Disk-Labels ist weiter unten erklärt.

3. Betriebssystem: Zuerst wird ein `CPM3.SYS` gesucht, existiert dies nicht, wird die zur Hardware passende Auslieferungsversion (`CPM31.SYS`, `CPM32.SYS`, `CPM33.SYS`, `CPM34.SYS`, `CPM35.SYS` oder `CPM36.SYS`) geladen.

#### I/O-Adressen:

Die folgenden I/O-Adressen werden von den Urladern LDR1700 und LDR1703 benutzt:

00..01	FDC 5 oder FDC 8/5 mit 5,25"-Laufwerken
04..05	FDC 8/5 mit 8"-Laufwerken
08..0F	HDC 5 Winchestercontroller
10..13	RAE 2 ... RAE 5
20..2F	CPU III
40..41	reserviert für Testzwecke
48..4B	SLAVE 186
50..5F	P/SIO
B8..B9	VIDEO I
E0..EF	CPU II, CPU I, HKM-CPU
F0..FE	FDC II Karte (Floppy- / Winchester)
FF	HDC 5 Winchestercontroller

Die folgenden Adressen werden vom Urlader LDRE1601 in der EPC II-Version benutzt:

08..0F	HDC 5 Winchestercontroller
DC..EF	EPC II
F0..FE	FDC II (mit XEBEC Controller)
FF	HDC 5 Winchestercontroller

Die folgenden Adressen werden vom Urlader LDR11700 in der EPC 1.6-Version benutzt:

08..0F	HDC 5 Winchestercontroller
48..4B	SLAVE 186
B8..B9	VIDEO I
E0..F7	EPC 1.6
FF	HDC 5 Winchestercontroller

Für den I/O-Adressraum der CPU's 180x gelten besondere Konventionen! Bitte in den technischen Beschreibungen dieser CPU's nachschlagen. Vom externen Adressraum sind folgende I/O-Adressen benutzt:

00..01	FDC 5 oder FDC 8/5 mit 5,25"-Laufwerken
08..0F	HDC 5 Winchestercontroller
40..41	reserviert für Testzwecke
48..4B	SLAVE 186
B8..B9	VIDEO I
F0..FE	FDC II Karte (Floppy- / Winchester)
FF	HDC 5 Winchestercontroller

Diese Adressen dürfen keinesfalls von anderen Karten belegt sein. Die Suchalgorithmen des Urladers werden dann verwirrt. (Weitere Adressen können von dem geladenen CP/M PLUS belegt sein!) Ferner sind die Adreßbelegungen der Karten im System zu berücksichtigen. FDC 5, FDC 8/5 und CPU III belegen z.B. noch weitere Adressen, die aber im Urlader nicht benutzt werden.

### Meldungen des Urladers

```
Systemtakt = ..MHz, Baudrate (CP/M) = ..
existing controllers ..
<ESC> for floppy
from floppy ..

starting again
```

Als **Systemtakt** werden 4, 5, 6 oder 8MHz gemeldet. Eine 0 heißt, daß der Systemtakt nicht dem vorgesehenen Takt entspricht, d.h. kleiner als 4MHz oder größer als 8MHz ist.

Als **Baudrate** werden die CP/M Baudratencodes gemeldet.

0F = 19200, 0E = 9600, 0C = 4800, 0A = 2400, 08 = 1200 Baud und 0 = keine Baudrate, d.h. VIDEO I oder SLAVE 186/GRAFIK 187.

Nach **existing controllers** wird ein Bit-Vektor der im System erkannten Controller-Karten hexadezimal ausgegeben.

bit 0 = FDC 5, bit 1 = FDC 8/5, bit 2 und 3 = FDC II, bit 4 = XEBEC, bit 5 = HDC 5, bit 6 = CPU 180x (Floppy) und bit 7 = CPU 1801 (Winchester). Es können mehrere Bits gleichzeitig an sein, wenn mehrere Controller-Karten im System stecken.

**(ESC) for floppy** erscheint nur, wenn eine Winchester oder eine EPROM-Disk vorhanden ist. Sie haben dann ca. 1 Sekunde Zeit, die ESCAPE-Taste zu betätigen.

**from floppy** erscheint nur, wenn das CP/M nicht von der Winchester geladen wird. Danach wird der Laufwerkscode angegeben. (vgl. DISKS). Dieses Laufwerk ist anschließend das für CP/M aktuelle Laufwerk. Es muß nicht unbedingt das Laufwerk A sein.

Die Meldung **starting again** besagt, daß keine gültige Diskette gefunden wurde. Danach muß (falls erforderlich) wieder die Leertaste betätigt werden.

Im weiteren können die vom CBIOS her bekannten Fehlermel-

dungen auftreten.

Da die jk82-Implementation des CP/M-Plus ganz ohne mechanische Massenspeicher auskommen kann und es in solchen Fällen dann auch keine Floppycontroller gibt, kann der Urlader in solchen Systemen die Systemtaktfrequenz nicht bestimmen. Damit das geladene Betriebssystem trotzdem korrekt arbeitet, sollte die Taktfrequenz im Patchbereich des Laders definiert werden.

Patchadresse für den Systemtakt

75h im Lader.

Der Code an dieser Stelle kann folgende Werte annehmen:

4	für 4 MHz
5	für 5 MHz
6	für 6 MHz
8	für 8 MHz
Offh	falls ein Floppy-Controller im System - automatische Taktbestimmung

Andere Werte führen u. U. zu einem fehlerhaften Arbeiten des Betriebssystems.

### Kontroll-Eintrag bei EPROM- und CRAM-Disks

Soll das Betriebssystem von einer EPROM- oder CRAM-Disk geladen werden, so muß der erste Directory-Eintrag einer solchen Disk ("Disk-Label", vgl. CP/M 3 User's Guide, S. 5-88) die folgenden Form haben:

' MEMDISKtgbd'

Die Buchstaben t, g, b und d stehen dabei für die nachfolgend beschriebenen Größen.

t = Memory-Disk Typ:

'0'	= RAM-Disk auf externer Karte (z.B. SMB), geschaltet
'1'	= CRAM-Disk auf externer Karte (z.B. EPROM/CRAM), linear
'2'	= EPROM-Disk auf externer Karte (z.B. EPROM/CRAM), linear
'3'	= RAM-Disk auf CPU-Karte (z.B. CPU 1801), linear
'4'	= CRAM-Disk auf CPU-Karte (z.B. CPU 1802), linear
'5'	= EPROM-Disk auf CPU-Karte (z.B. CPU 1802), geschaltet
'6'	= CRAM-Disk auf externer Karte (z.B. S-CRAM), geschaltet
'7'	= EPROM-Disk auf externer Karte, geschaltet

g = EPROM-Größe:

Nur bei EPROM-Disks mit geschalteten EPROM's! Bei allen anderen Memory-Disks durch "Underline" ('\_') ersetzt.

'0'	= 64KByte-EPROM's
'1'	= 128KByte-EPROM's
'2'	= 256KByte-EPROM's usw.



b = Block-Größe:

DSM < 256	DSM >= 256	(DSM = Disk Storage Maximum in Blocks)
'0'	'4'	= 2KByte Blocks
'1'	'5'	= 4KByte Blocks
'2'	'6'	= 8KByte Blocks
'3'	'7'	= 16KByte Blocks

d = Anzahl Directory-Einträge:

'0'	= 64 Directory-Einträge	
'1'	= 128 Directory-Einträge	
'2'	= 192 Directory-Einträge	
'3'	= 256 Directory-Einträge	usw.

Bei EPROM-Disks wird dieser Eintrag vom Programm "EPRCOPY" automatisch erzeugt. Soll von einer CRAM-Disk gebootet werden, so muß der Anwender den Eintrag selbst generieren. Beispiel:

Drive C: ist eine externe 256KByte CMOS-Ramdisk mit 2KByte Blocks und 64 Directory-Einträgen:

SET C: ÄNAME=MEMDISK1\_00Ü

Außerdem müssen auf dieser Disk dann auch die Dateien "CPM3.SYS" und "CCP.COM" vorhanden sein (bei den CPU-Karten vom Typ CPU 180x ist die Datei "CCP.COM" überflüssig).

Für die jk82 CMOS-RAM-Karten können die Einträge aus folgender Tabelle entnommen werden.

jk82 EPROM/CRAM (bis 256KByte)	'MEMDISK1_00'
jk82 EPROM/CRAM (bis 512KByte)	'MEMDISK1_01'
jk82 EPROM/CRAM (bis 768KByte)	'MEMDISK1_42'
jk82 CPU1802	'MEMDISK4_00'

Generieren eines neuen Urladers

Sie benötigen hierzu die als Option erhältliche Diskette mit den Loader-Modulen. Ferner ist der Assembler MACRO-80 erforderlich. Die Diskette enthält alle Module für die 6 Urlader LDR1700, LDR1703, LDR11700, LDRE1601, LDR1710 und LDR1720.

Der Urlader LDR1700 besteht aus den Modulen:

LDRBOOT.MAC, LDRPGM.MAC, CPMLDR.REL, LDRKRNL.MAC,  
LDRCHAR.MAC, LFUTIL.REL (FUTIL), L2.REL (FDC2), L5.REL  
(FDC5), L85.REL (FDC85), LXEBEC.REL (XEBEC), LHDC5.REL  
(HDC5) und LDRDISKS.MAC.

Der Urlader LDR1703 besteht aus den Modulen:

LDRBOOT.MAC, LDRPGM.MAC, CPMLDR.REL, LDRKRNL.MAC,  
LDRCHAR.MAC, LFUTIL.REL (FUTIL), L2.REL (FDC2), L5.REL  
(FDC5), L85.REL (FDC85), LMVE.MAC, LHDC5.REL (HDC5) und  
LDRDISKS.MAC.

Der Urlader LDRE1601 besteht aus den Modulen:

LDREBOOT.MAC, LDRPGM.MAC, CPMLDR.REL, LDREKRNL.MAC,  
LDRECHAR.MAC, LFUTIL.MAC (FUTIL), LE2.REL (EPC2FDC),  
LXEBEC.REL (XEBEC), LHDC5.REL (HDC5), T.REL  
(EPC7220), LDRGEN.MAC und LDREDISK.MAC.

Der Urlader LDR11700 besteht aus den Modulen:

LDR1BOOT.MAC, LDRPGM.MAC, CPMLDR.REL, LDR1KRNL.MAC,  
LDR1CHAR.MAC, LMVE.MAC, LFUTIL.MAC (FUTIL), LE1.REL  
(EPC1FDC), LHDC5.REL (HDC5) und LDR1DISK.MAC.

Der Urlader LDR1710 besteht aus den Modulen:

L181BOOT.MAC, L180KRNL.MAC, L180CHAR.MAC, L180DISK.MAC,  
L181EPR.MAC, L180PGM.MAC, ROMBIOS.MAC, LFUTIL.REL,  
LHDC180.REL, LHDC5.REL, L18.REL (FDC180), L5.REL (FDC5),  
L2.REL (FDCII), LXEBEC.REL, CPMLDR.REL, CCP.COM.

Der Urlader LDR1720 besteht aus den Modulen:

L182BOOT.MAC, L180KRNL.MAC, L180CHAR.MAC, L180DISK.MAC,  
L182EPR.MAC, L180PGM.MAC, ROMBIOS.MAC, LFUTIL.REL,  
LHDC180.REL, LHDC5.REL, L18.REL (FDC180), L5.REL (FDC5),  
L2.REL (FDCII), LXEBEC.REL, CPMLDR.REL, CCP.COM.

Obige Module werden durch die Prozeduren LDR.SUB, LDREPR.SUB, LDR1.SUB, LDRE.SUB, LDR181.SUB bzw. LDR182.SUB zu einem ROM-fähigen Programm zusammengebunden. Für die ersten vier Loader reicht ein 8KByte EPROM aus, da die Datenbereiche nicht vorbesetzt sein müssen. Die letzten beiden Loader passen in ein 32KByte EPROM.

Achtung: Die 6 folgenden Prozeduren generieren je einen neuen LOADER für CP/M PLUS. Bei den ersten vier Prozeduren ist darauf zu achten, daß die "FF"-Bytes vor 2100H beginnen müssen; sonst ist der generierte LOADER zu lang und paßt nicht in ein 8 KByte EPROM!

```
;***** bootstraploader LDR 1.700 generieren -- LDR.SUB (08.04.87) --
pip
<eprldr.mac=con:
<epc1 equ 0
<^Jepr_ldr equ 0
<^J^Z
m80=ldrkrnl
m80=ldrchar
m80=ldrdisks
m80=ldrpgm
m80=ldrboot
era ldr1700.com
link lbiosÄlB00Ü=ldrkrnl,ldrchar,lfutil,lhdc5,lxebec,l5,l85,l2,ldrdisks
link ldrpgm
link cpmlldr
link ldrboot
sid
<rldrboot.com
<rcpmlldr.com C0
<rldrpgm.com 8000
<m8100 8280 1C0
<rlbios.com aC0
<d2080 217f
<wldr1700.com 100 20ff
<g0
;***** ende der prozedur LDR.SUB *****
```

```
;***** bootstraploader LDR 1.703 generieren -- LDREPR.SUB (08.04.87) --
;***** Version fuer cpu1/cpu2/cpu3 eprom/hdc5-loader
pip
<eprldr.mac=con:
<epc1    equ 0
<^Jepr_ldr equ -1
<^J^Z _
m80=ldrkrnl
m80=ldrchar
m80=ldrdisks
m80=ldrpgm
m80=ldrboot
m80=lmve
era ldr1703.com
link lbiosÄlB00Ü=ldrkrnl,lmve,ldrchar,lfutil,lhdc5,15,185,12,ldrdisks
link ldrpgm
link cpmlldr
link ldrboot
sid
<rldrboot.com
<rcpmlldr.com C0
<rldrpgm.com 8000
<m8100 8280 1C0
<rlbios.com aC0
<d2080 217f
<wldr1703.com 100 20ff
<g0
;**** ende der prozedur LDREPR.SUB ****
```

```

;***** bootstraploader LDR1 1.700 generieren -- LDR1.SUB (09.04.87) --
;***** Version fuer epc1 als eprom/hdc5-loader
pip
<eprldr.mac=con:
<epc1 equ -1
<^Jepr_ldr equ -1
<^J^Z -
m80=ldr1krnl
m80=ldr1char
m80=ldr1disk
m80=ldrpgm
m80=ldr1boot
m80=lmve
era ldr11700.com
link lbiosÄ18B00Ü=ldr1krnl,lmve,ldr1char,le1,lfutil,lhdc5,ldr1disk
link ldrpgmÄ18100Ü
link cpmlldrÄ18100Ü
link ldr1bootÄ10Ü
sid
<rldr1boot.com
<rcpmlldr.com 80
<rldrpgm.com 8000
<m8100 8280 180
<rlbios.com a80
<d2080 217f
<wldr11700.com 100 20ff
<g0
;**** ende der prozedur LDR1.SUB *****

```

```

;***** bootstraploader LDRE 1.601 generieren -- LDRE.SUB (26.04.85) --
m80 =ldrekrnl
m80 =ldrechar
m80 =ldredisk
m80 =ldrpgm
m80 =ldreboot
m80 =ldrgen
; ---- LOADER zusammenbauen
era ldre1601.com
link lbiosÄ1B00Ü=ldrekrnl,ldrechar,lfutil,le2,lxebec,lhdc5,t,ldrgen,ldredisk
link ldrpgm
link cpmlldr
link ldreboot
sid
<rlbios.com
<m100 2000 5000
<m5000 7000 0b00
<rcpmlldr.com
<rldrpgm.com
<m100 20ff 5000
<rldreboot.com
<m5000 6fff 180
<d2080 217f
<wldre1601.com 100 20ff
<g0
;**** ende der prozedur LDRE.SUB *****

```

```

;***** bootstraploader LDR 1.710 generieren -- LDR181.SUB (01.04.87) --
m80 =l180krnl
m80 =l180char
m80 =l180disk
m80 =l180pgm
m80 =l181boot
m80 =l181epr
m80 =rombios
era ldr1710.com
link lbiosÄlB00Ü=l180krnl,l180char,l181epr,lfulil,lhdc180,lhdc5,l18,l5,l2,
                lxebec,l180disk
link cpmlldr
link l181boot
link l180pgm
link rombios
sid
<rl181boot.com
<rcpmlldr.com 200
<rl180pgm.com B000
<mB100 B280 300
<rlbios.com C00
<rrombios.com 7000
<rccp.com 7380
<wldr1710.com 100 80ff
<g0
;**** ende der prozedur LDR181.SUB ****

```

```

;***** bootstraploader LDR 1.720 generieren -- LDR182.SUB (24.03.87) --
m80 =l180krnl
m80 =l180char
m80 =l180disk
m80 =l180pgm
m80 =l182boot
m80 =l182epr
m80 =rombios
era ldr1720.com
link lbiosÄlB00Ü=l180krnl,l180char,l182epr,lfulil,lhdc180,lhdc5,l18,l5,l2,
                lxebec,l180disk
link cpmlldr
link l182boot
link l180pgm
link rombios
sid
<rl182boot.com
<rcpmlldr.com 200
<rl180pgm.com B000
<mB100 B280 300
<rlbios.com C00
<rrombios.com 7000
<rccp.com 7380
<wldr1720.com 100 80ff
<g0
;**** ende der prozedur LDR182.SUB ****

```

In den Urlader-EPROMs vorhandene Patchbereiche:

```

0075  clk      Systemtakt (n u r bei Systemen ohne Floppy-
          Controller ändern)
0077  tc       CTC time constant
0078  st5      steprate + headunload-time für 5,25" Laufwerke
0079  st8      steprate + headunload-time für 8" Laufwerke
007A  hld5     headloadtime für 5,25" Laufwerke
007B  hld8     headloadtime für 8" Laufwerke

```

Ein Wert für hld5 bzw. hld8 wird immer genau dann akzeptiert, wenn st5 bzw. st8 gepatcht sind. D.h. wenn Sie st5 oder st8 patchen, darf der Wert FF keinesfalls in hld5 oder hld8 stehen bleiben; Sie müssen dann auch hld5 bzw. hld8 patchen (hld-Werte müssen geradzahlig sein!!).

Wenn für tc ein Wert ungleich FF gepatcht wird, so bootet der Urlader nur noch mit der seriellen Schnittstelle (SIO B bzw. SIO 2) und der angegebenen Baudrate! In diesem Fall muß also unbedingt ein Terminal mit der richtigen Baudrate angeschlossen sein!

Alle Patchbereiche sind mit OFFH vorbelegt, jeder andere Wert wird als feste Einstellung für den LOADER akzeptiert.

Werte für obige Patches:

CTC time constant ist der Wert, mit dem der CTC für die Konsole programmiert wird.

Beispiele:

	Baudrate	4MHz	5MHz	6MHz
CPU IIa	19K2	-	8	A
CPU IIa	9600	D	10	14
CPU IIa	4800	1A	20	27
CPU IIa	2400	34	40	4E
CPU IIa	1200	68	80	9C
CPU I	19K2	4	4	4
CPU I	9600	8	8	8
CPU I	4800	10	10	10
CPU I	2400	20	20	20
CPU I	1200	40	40	40
P/SIO	19K2	4	4	4
P/SIO	9600	8	8	8
P/SIO	4800	10	10	10
P/SIO	2400	20	20	20
P/SIO	1200	40	40	40
HKM-CPU	9600	0D		
HKM-CPU	4800	1A		
HKM-CPU	2400	34		
HKM-CPU	1200	68		

	Baudrate	4MHz	5MHz	6MHz
EPC 1.6	19K2	6	6	6
EPC 1.6	9600	C	C	C
EPC 1.6	4800	18	18	18
EPC 1.6	2400	30	30	30
EPC 1.6	1200	60	60	60
EPC 1.6	600	C0	C0	C0
EPC II	19K2			07
EPC II	9600			0E
EPC II	4800			1C
EPC II	2400			38
EPC II	1200			70
CPU 180x	19K2			01
CPU 180x	9600			08
CPU 180x	4800			09
CPU 180x	2400			0A
CPU 180x	1200			0B
CPU 180x	600			0C
CPU 180x	300			0D

Die time-constant-Werte sind hexadezimale Zahlen!

CPU IIa steht für CPU II und CPU II-256 ohne Baudraten-Quarz. Für eine CPU II-256 mit Baudratenquarz gilt die Tabelle von CPU I.

#### steprate- und headunload-time (vgl. $\mu$ PD765-Manual)

Wert	5,25"	8"	
0F	32	16	ms stepratetime
1F	30	15	ms stepratetime
2F	28	14	ms stepratetime
:	:	:	:
EF	4	2	ms stepratetime

Für 5,25" TEAC- oder NEC-Laufwerke beträgt die stepratetime 3ms. Somit ist 4ms, also st5 = EF einzustellen.

Für 8" NEC-Laufwerke beträgt die stepratetime 3ms. Somit ist 3ms also st8 = DF einzustellen

#### headloadtime (hld) (vgl. $\mu$ PD765-Manual)

5,25" Laufwerke: wert = hlt / 4 in ms  
8" Laufwerke: wert = hlt / 2 in ms

Achtung: wert muß geradzahlig sein!

Für 5,25" TEAC oder NEC-Laufwerke ist hlt = 50ms, also ist hld5 = 0E einzustellen

Für 8" NEC-Laufwerke ist hlt = 60ms, also ist hld8 = 1E einzustellen.



## 4.2 BIOSKERN

Das Modul BIOSKERN ist aus BIOSKRNL.ASM von DRI entstanden. Das Original von DRI ist im Anhang E des CP/M 3 System Guide abgedruckt. Das Original-Modul ist wie folgt erweitert worden:

### Initialisierung:

Von DRI war vorgesehen, alle zusätzlich notwendigen Initialisierungen in dem Modul BOOT vorzunehmen. Dies ist jedoch nicht möglich, da bereits vorher gebankt wird und der Systemtakt bekannt sein muß. Die Initialisierung der Banking-Hardware und des Character-I/O-Moduls (nicht der einzelnen Character-I/O-Devices) muß deshalb bereits vor allen anderen Initialisierungen geschehen!

Die weiteren Änderungen in diesem Bereich dienen der Vereinheitlichung der einzelnen Aufrufe (z.B. bei der Initialisierung der Disk-Routinen zeigt das Register DE auf den XDPH, da dies bei allen anderen Aufrufen von Disk-Routinen ebenfalls der Fall ist).

### Character-I/O:

Das gesamte Character-I/O-Modul ist in die Bank 0 verlegt worden. Der Aufruf der einzelnen Character-I/O-Routinen mußte deshalb geändert werden. Darüber hinaus wurde für Output-Devices ein 'TIME OUT' eingeführt. Falls ein Ausgabe-Gerät sich nicht innerhalb der vorgesehenen Zeit 'fertig' meldet, wird auf der Console die Meldung:

TIME OUT name ?

ausgegeben. 'name' ist der Geräte-Name des Gerätes, das nicht rechtzeitig 'fertig' geworden ist. Meist ist dieses Gerät dann nicht angeschlossen, nicht eingeschaltet oder befindet sich in einem Fehlerzustand (z.B. in dem Drucker ist kein Papier mehr). Nach Ausgabe der 'TIME OUT'-Meldung erwartet CBIOS 1.5 eine Eingabe von der Console.

Es sind 3 verschiedene Eingaben zulässig:

1. Return: Die Ausgabe wird erneut versucht. Diese Eingabe ist sinnvoll, wenn inzwischen der Grund für 'TIME-OUT' beseitigt wurde (z.B. Papier ist jetzt eingelegt).
2. ESC: Durch betätigen der ESC-Taste wird das TIME-OUT-Gerät deaktiviert, d.h. es werden nun keine Ausgaben mehr auf dieses Gerät versucht. Das Anwender-Programm läuft nun weiter. Diese Eingabe ist sinnvoll, wenn der Grund für 'TIME-OUT' nicht beseitigt werden kann, aber das laufende Programm nicht unterbrochen werden soll. Eine erneute Aktivierung des nun deaktivierten Gerätes kann mit DEVICE oder CONFIG vorgenommen werden.

### 3. ctrl C: Rücksprung in das Betriebssystem

Achtung: Wenn das TIME-OUT-Gerät der Drucker ist und ctrl P aktiv ist, ist diese Eingabe nicht sinnvoll, da ctrl P aktiv bleibt und das System immer noch hängt!

Die TIME-OUT-Länge wird in der erweiterten 'Character Device Table' festgelegt. Änderungen dieser Zeitkonstanten sind mit dem Dienstprogramm CONFIG möglich.

Das Modul BIOSKERN ist sowohl als Source- als auch als REL-File auf den Auslieferdisketten vorhanden. Von Änderungen im Source durch den Anwender wird jedoch dringend abgeraten, da diese Änderungen naturgemäß in zukünftigen Versionen dieses Moduls nicht berücksichtigt werden können!

Alle Änderungen gegenüber der Original-Version von DRI sind mit einem '\*' in der jeweiligen Zeile gekennzeichnet. Die Art der Änderungen kann durch Vergleich mit dem Original (Anhang E des System Guide) ermittelt werden.

#### WARNUNG:

Direkte BIOS-Aufrufe außer CONIN, CONOUT, CONSTA, LSTOUT, und LSTSTA sind nicht nur verboten, sondern führen in den meisten Fällen zu einem Systemabsturz! Bitte beachten Sie dazu die Definitionen von DRI!

### 4.3 Das Boot-Modul

Das Modul **BOOT.REL** initialisiert das soeben geladene CP/M PLUS Betriebssystem. Es definiert die Interrupt-Page und den Interrupt Mode, initialisiert die Uhr, und sendet die Versionsnummern der CBIOS-Module. Ferner enthält es eine innerhalb des CBIOS benutzbare Routine, die Vektoren in die Interrupt-Page einträgt.

#### Interrupt-Vereinbarungen

Interrupt Mode ist **IM 2**. Die Interrupt-Page liegt im **COMMON** auf Adresse FF00 bis FF7F. Das System belegt die Tabelle ab FF00 aufwärts. Oberhalb von FF80 befindet sich ein CBIOS-Stack.

#### Interrupt-Semaphor

Da beim Z80 nach einem DI-Befehl der vorherige Interrupt-Status nicht feststellbar ist, wurde im CBIOS eine Interrupt-Semaphore definiert. Programmbeispiel:

```

:
di                ; interrupt ausschalten
dec (intsem)      ; intsem runterzaehlen
:
inc (intsem)      ; wieder hochzählen
jr nz,notei      ; nur bei 0 darf EI sein
ei                ; interrupt wieder einschalten
notei:           :
```

#### Interrupthilfsroutine      **SETVEC**

Die Routine **SETVEC** trägt die für einen Baustein benötigten Adressen in die Interrupt-Tabelle ein. Zurückgegeben wird in L der in das Chip zu schreibende Vektor und in H ein FFH. Eine 0 in HL besagt, daß die Interrupt-Tabelle voll ist. SETVEC verwaltet die Interrupt-Tabelle. Alle direkten Zugriffe auf die Interrupt-Tabelle sind ausdrücklich verboten. Als Eingabeparameter erwartet SETVEC in DE die Adresse der Adressen, die in die Interrupt-Tabelle geschrieben werden sollen und in B die Anzahl der Worte.

Die nachfolgende Seite zeigt ein Beispiel dazu.

```

:                                ; CTC initialisieren
ld    de,table
ld    b,4                        ; CTC generiert 4 Interrupts
call  setvec                     ; in Tabelle eintragen lassen
ld    a,h
or    a                           ; Tabelle voll?
jr    z,error                    ; ja: Fehler, Programmabbruch
di
ld    a,l                        ; Vektor in A
out   (ctc),a
ld    a,(intsem)                ; nachsehen ob Interrupt enabled war
or    a                           ; 0 = enabled
jr    nz,notei                  ; nicht 0: disabled lassen
ei                                     ; Interrupt wieder einschalten
notei:                             ; jetzt wird der CTC programmiert

table:  defw  vector0            ; Vektor für Kanal 0
        defw  vector1            ; Vektor für Kanal 1
        defw  vector2            ; Vektor für Kanal 2
        defw  vector3            ; Vektor für Kanal 3

```

#### 4.4 Disk Routinen

Das CBIOS für CP/M PLUS ist ein modulares System. Es wird mit LINK zu der Datei BNKBIOS3.SPR gebunden. Für eventuelle Änderungen muß also nur das jeweilige Modul modifiziert werden. Der Diskteil besteht aus den folgenden Modulen:

- DISKS.ASM Das Modul DISKS.ASM wird von CPMINST erzeugt. Benötigt die Bibliotheken DISKDEF.LIB (Definition der Laufwerksdaten), RAMDISK (Definition einer Halbleiterfloppy) und DISKS.LIB (Definition der Controller und Voreinstellungen). In DISKS werden die vorhandenen Laufwerke und Floppy- bzw. Winchester-Controller vereinbart.
- DISKDEF.LIB Die Bibliothek DISKDEF.LIB enthält die Daten der uns bereits bekannten Laufwerke. Weitere Laufwerke sind als neue Makros hinzuzufügen.
- DISKS.LIB Die Bibliothek DISKS.LIB enthält Makros für unsere Controllerkarten. **Hier sollten vom Anwender keine Änderungen vorgenommen werden.**
- RAMDISK.LIB Die Bibliothek RAMDISK.LIB enthält die Makros zum Generieren einer RAMDISK, SMB-RAMDISK, EPROM-DISK und CMOS-RAMDISK. Die Treiber dafür sind im jeweiligen MOVE-Modul enthalten. **RAMDISK sollte nicht verändert werden.**
- FUTIL.REL In FUTIL.REL = Floppy-Utilities sind die kartenunabhängigen Teile der Floppy-Routinen zusammengefasst. FUTIL ruft über die Adressen im XDPH kartenabhängige Routinen auf. Als Floppy-Controller-Chip wird ein NEC µPD765 vorausgesetzt.
- FDC2.REL FDC2.REL ist der kartenabhängige Treiber für die FDC II Floppy-Controller-Karte. FDC2 umfaßt nur den Treiber für die Floppy-Schnittstelle der FDC II-Karte. FDC2 bedient max. 3 5,25"- und 4 8"-Floppy-Laufwerke.
- FDC85.REL FDC85.REL ist der kartenabhängige Treiber für die FDC8/5-Karte mit den I/O-Adressen 04H bis 07H. Die FDC8/5-Karte bedient ausschließlich 8" Floppylaufwerke im DMA-Mode.
- FDC5.REL FDC5.REL ist der Treiber für die FDC 5-, DPS-I0- oder FDC8/5-Platine bei Einsatz von 5,25" Floppylaufwerken. Dieser Treiber bedient 5,25" Laufwerke immer im NON-DMA-MODE.
- EPC2FDC.REL EPC2FDC.REL ist der kartenabhängige Treiber für den Floppy-Controller der EPC II-Karte. EPC2FDC umfaßt nur den Treiber für die Floppy-Schnittstelle. EPC2FDC bedient bis zu 3 Floppy-Laufwerke (5,25" oder 8")

- EPC1FDC.REL** EPC1FDC.REL ist der kartenabhängige Treiber für den Floppy-Controller der EPC 1.6-Karte. EPC1FDC umfaßt nur den Treiber für die Floppy-Schnittstelle. EPC1FDC bedient bis zu 3 Floppy-Laufwerke (3,5" oder 5,25")
- DDATA.ASM** DDATA.ASM enthält die beim Start des Systems aktiven Fremdformate. Hier sind 4 Formate vereinbart: ZDOS 5,25" 80 trk, ZDOS 5,25" 40 trk, ZDOS 8" und CP/M A1 (Single Density). Andere Formate können durch Editieren gewählt werden. Achtung: Ein fehlerhaftes Format in DDATA kann die Funktion des Systems verhindern. Die benötigten Werte für ein Format können Sie in der Datei DISKFORM.DAT finden! Nehmen Sie in DDATA nur dann Änderungen vor, wenn es anders nicht möglich ist. Fehler (inkonsequente Tabellen) sind i.a. katastrophal.
- DDATA.REL** Das Modul DDATA.REL enthält die Informationen zur Verarbeitung von fremden Disk-Formaten. Vereinbart sind 4 Formate: ZDOS 5,25" 80 trk, ZDOS 5,25" 40 trk, ZDOS 8" und CP/M A1 (Single Density). DDATA enthält Platz für 9 Formatbeschreibungen.
- XEBEC.REL** Das Modul XEBEC enthält die Treiberrountinen für bis zu zwei 5,25" Winchester-Laufwerke. Es setzt einen XEBEC S1410 oder S1410A Winchester-Controller an der SASI-Schnittstelle der FDC II-Karte voraus. Eine Spur für CP/M enthält 256 SASI-Blocks zu je 512 Byte (unabhängig von der realen Sektorzahl des Laufwerkes). In XEBEC werden die Optionen und das Timing der Firmware S1410 rev. E oder S1410A benutzt.
- HDC5CARD.REL** Das Modul HDC5 enthält die Treiberrountinen für bis zu zwei 5,25" (oder 3,5") Winchester-Laufwerke. Diese sind direkt mit der HDC 5-Karte zu verbinden. In HDC 5 wird mit den physikalischen Spuren des Laufwerks gearbeitet. Somit enthält eine Spur  $n * 9$  Sektoren a 1 KByte.  $n$  ist hier die Anzahl der Datenoberflächen und liegt zwischen 1 und 8. HDC 5 führt ein doppeltes Inhaltsverzeichnis. Somit ist eine Kopie der Dateien auch noch möglich, wenn einzelne Fehler im Inhaltsverzeichnis vorhanden sind. Bei Fehler wird allerdings nicht mehr geschrieben.

Das Modul DISKS:

Das Modul DISKS.ASM wird beim Generieren eines CP/M Betriebssystems von CPMINST erzeugt. Die Auslieferversionen unseres CP/M enthalten ein besonderes DISKS-Modul. Das System ist mit der Auslieferversion lauffähig, aber nicht optimal. Ein optimiertes System setzt Anpassungen in DISKS voraus. Achtung: Jedes Laufwerk hat eigene Datenfelder in DISKS. 2 gleiche Laufwerke sind auch zweimal in DISKS zu vereinbaren! In der drive-table (§DTBL) steht als erstes Laufwerk "A" und als 16tes Laufwerk "P". Die drive-table hat immer 16 Einträge.

Das Modul DISKS enthält die folgenden Tabellen:

- Die §DTBL (Drive-table) enthält die DPH-Adressen der 16 Laufwerke, ein nicht vorhandenes Laufwerk hat eine 0 als DPH-Adresse. Zwischen den Laufwerken darf eine Lücke sein. §DTBL ist immer 16 Worte lang! §DTBL ist von uns erweitert worden: Direkt vor §DTBL stehen Informationen über die Laufwerke und Controller. Das Informationsbyte enthält den Controllercode in den oberen 4 Bits (7..4) und die Laufwerksadresse (die am Laufwerk gejumperte Adresse) in den unteren 4 Bits (3..0).

Definition des Controllercodes:

0 = FDC5 (5,25" Laufwerk)	1 = FDC8/5 (8" Laufwerk)
2 = FDCII (5,25" Laufwerk)	3 = FDCII (8" Laufwerk)
4 = SASI am FDCII	5 = Ramdisk
6 = EPC II Floppy	7 = HDC 5 Winchester
8 = Epromdisk	9 = EPC I Floppy
10 = CPU 180x Floppy	11 = CPU 180x Winchester
15 = reserviert für optische Platten	

Genau dieser Code wird im Urlader für das BOOT-Floppy-Device gemeldet.

```

Beispiel:          db   drv_a   ; Laufwerk A
                   :
                   db   drv_p   ; Laufwerk P
§DTBL:            dw   dph_a   ; Parameterkopf A
                   :
                   dw   dph_p   ; Parameterkopf P

```

- DPH, XDPH und XXDPH (Disk-Parameter-Header, extended DPH, extended XDPH) sind die Tabellen der Laufwerksdaten. Die Adresse des DPH wird beim Aufruf einer Diskroutine im Register DE vom BIOSkernel übergeben. Die jeweilige Routine benutzt intern das Register IY als Pointer auf den XXDPH. Bezugspunkt für alle Daten ist immer die Adresse DPH. Damit ist die Reihenfolge XXDPH, XDPH und DPH festgelegt. Es dürfen nur Werte verändert werden, aber weder Bytes gelöscht noch hinzugefügt werden. Der XXDPH ist Controllerspezifisch definiert, d.h. es gibt 3 XXDPH-Typen für Floppy, Winchester und RAMDISK. Für jedes Laufwerk ist ein eigener XXDPH erforderlich!

Definition:

	XXDPH:	laufwerksabhängiger Teil (siehe unten)
	XDPH:	hier beginnt der XDPH
-10	word	Adresse der WRITE-Routine
-8	word	Adresse der READ-Routine
-6	word	Adresse der LOG-IN-Routine
-4	word	Adresse der Initialisierungsroutine
-2	byte	Controlleradresse des Laufwerks (0..3)
-1	byte	Laufwerkstyp (nicht benutzt)
	DPH:	hier beginnt der Disk-parameter-header
+0	word	Adresse der Sektorverschränkungsliste
+2	byte	-- reserviert --
3	byte	-- reserviert --
4	byte	-- reserviert --
5	byte	-- reserviert --
6	byte	-- reserviert --
7	byte	-- reserviert --
8	byte	-- reserviert --
9	byte	-- reserviert --
10	byte	-- reserviert --
11	byte	MEDIA-CHANGED-Flag (wird nicht benutzt)
12	word	Adresse des Disk-Parameter-Blocks = OFFFEH, reale Adresse wird von GENCPM eingetragen.
14	word	Adresse Checksum-Vector (OFFFEH)
16	word	Adresse Allocation-Vector (OFFFEH)
18	word	Adresse Directory-Buffer-Control-Block (OFFFEH)
20	word	Adresse Datenbuffer-Control-Block (OFFFEH)
22	word	Adresse der Hash-Tabelle (OFFFEH)
24	byte	Bank-nummer der Hash-Tabelle (0)

Der Wert OFFFEH veranlasst GENCPM den benötigten Platz bereit zu stellen, und die entsprechende Adresse einzutragen.



XXDPH: (für Floppylaufwerke)

laufwerksbezogene Daten:

- (A) byte *drvsid* Seiten des Laufwerks (1 oder 2)
- (H) byte *drvtrk* Spuren pro Seite (z.B. 40, 77 oder 80)
- byte *ptime* Größe und Precompensationtime für FDC II
  - bit 5 = 1 => 5,25" Laufwerk
  - bit 5 = 0 => 8" Laufwerk
  - bit 1, 2 und 3 stellen die Write-Precompensation-Zeit ein:

P2	P1	P0	5,25"	8"	
0	0	0	0	0	ns
0	0	1	125	62.5	ns
0	1	0	250	125	ns
0	1	1	375	187.5	ns
1	0	x	500	250	ns
1	1	x	625	312.5	ns

- byte *srt* step-rate und headunload-time (µPD765)
  - bit 7..4 = step-rate-time
  - 1111 = 1ms (8"), 2ms (5,25")
  - bis 0000 = 16ms (8"), 32ms (5,25")
  - bit 3..0 = head-unload-time
  - 1111 = 240ms (8"), 480ms (5,25")
- byte *hlt* headloadtime und mode (DMA, NON-DMA)
  - bit 7..1 = head load time von 2 bis 254ms in 2 ms Schritten (bei 8"); bei 5,25" verdoppeln sich die Zeiten.
  - bit 0 = mode (1 = NON-DMA, 0 = DMA-Mode)
  - \* Achtung: FDC II arbeitet im DMA-mode \*

berechnete Werte:

- (2) byte *trksft* trackshift (0 bei 1 step/track, 1 bei 2 steps/track)
- (2) word *sknl,h* Sektorlänge in byte (z.B. 128, 256..)
- (1) byte *size* Größe des Laufwerks (0 = 8", 1 = 5,25")

*Bit 1=1 => HD möglich  
bei Form. 1 wird nicht geprüft, ob größe pa*

erkannte Werte der Diskette:

- (1)(2) byte *fm* Aufzeichnungsdichte: 0 = FM, 40H = MFM, bei 0COH hat die MFM-Diskette Systemspuren in FM-Aufzeichnung; diese können nicht verarbeitet werden.
- (2) byte *ssize* Sektorgröße (N für den µPD765). 0 = 128 byte, 1 = 256 byte, 2 = 512 byte, 3 = 1024 byte. Dieses Byte entspricht dem PSH im Disk-Parameter-Block.
- (2) byte *diskid* Nummer der Diskette. Die Nummer steht in der ID eines jeden Sektors. Sie entspricht der HEAD-Information in der Sektor-ID. Normale Disketten verwenden hier nur die Nummern 0 und 1. Die weiteren Nummern können bei "internen" Disketten zur softwaremäßigen Erkennung eines Disketten-Wechsels benutzt werden. (geplant)
- (2) byte *number* Sektor-ID-Spurnummer der Spur 0. (normal: 0)
- (1) (3) byte *sides* ein- oder zweiseitige Diskette, wird erkannt durch versuchsweises Lesen der Rückseite).
- (1)(2)(3) byte *mxtrk* Spuren der Diskette auf einer Seite (z.B. 35, 40, 77 oder 80)

COMPARE

xxdget, xxdput

<p>(1)(2)(3) (2)(3) (3)</p>	<p>↑</p>	<p>byte <i>mxsec</i> Sektoren pro Spur (einseitig!)          byte <i>gapl</i> read/write-gap-length (μPD765)          byte <i>fgap</i> format-gap-length (μPD765)</p>	<p>8" FM: Byte/Sektor 128 256 512 1024 2048 4096          Sektoren 26 15 8 4 2 1          R/W-gaplength 7 14 27 71 200 200          Format-gap 27 42 58 138 255 255          8" MFM: Byte/Sektor 256 512 1024 2048 4096 8192          Sektoren 26 15 8 4 2 1          R/W-gaplength 14 27 69 153 200 200          Format-gap 54 84 116 255 255 255          5,25" FM: Byte/Sektor 128 128 256 512 1024 2048          Sektoren 18 16 8 4 2 1          R/W-gaplength 7 16 24 70 200 200          Format-gap 9 25 48 133 255 255          5,25" MFM: Byte/Sektor 256 256 512 1024 2048 4096          Sektoren 18 16 8 4 2 1          R/W-gaplength 10 32 42 128 200 200          Format-gap 12 50 80 240 255 255</p>
<p>(3)</p>	<p>↑</p>	<p>byte <i>skew</i> Format-Skew-Faktor der Diskette (ist meistens 1, da die Sektorverschränkung im CP/M gemacht werden kann)</p>	
<p>(1)(2)(3)</p>	<p>↑</p>	<p>byte <i>iflag</i> <math>\phi</math> Flag          Bit 0 normal (0) oder invers (1)          Bit 1 Sektor 1 (0) oder Sektor 0 (1) als erster          Bit 2 Kopf weiterzählen (0) oder gleichlassen (1) bei Übergang auf die Rückseite          Bit 3 auf der Rückseite von außen nach innen (0) oder von innen nach außen (1) steppen</p>	
<p>(3)</p>	<p>↑</p>	<p>byte <i>iflag</i> Bit 4 (0) normal oder (1) für ALPHATRONIC P4 fremde (0) oder Diskette mit Beschreibungsblock (1)</p>	
		<p>word <i>rdidl</i>,h Adresse der READ-ID-Routine          word <i>forml</i>,h Adresse der Formatier-Routine  <u>ab hier folgt der XDPH (für alle Laufwerke gleich)</u></p>	

- (1) = aus Description-Record in DISKFORM.DAT
- (2) = von LOGIN veränderte Werte
- (3) = aus Description-Block auf Spur  $\phi$  sector  $\phi$  [1]

## XYDPH:

DRV\_ART: byte 0: 1: Read (beliebig oft lesbar)  
0: not Read (nicht lesbar)  
1: 1: Write (beschreibbar)  
0: not Write (nicht beschreibbar)  
2: 1: Write once (falls beschreibbar)  
(nur einmal beschreibbar z.B. WORM)  
0: keine Begrenzung auf einmal beschreiben  
3: 1: nicht wechselbares Medium  
0: wechselbares Medium  
4: 1: Datenverlust nach Ausschalten  
0: kein Datenverlust nach Ausschalten  
5: 1: Drive enthält nur MS-DOS-Medien  
0: Drive enthält beliebige Medien  
6: reserviert (hier 0)  
7: reserviert (hier 0)

DRV\_STAT0: byte 0: 1: Drive-Treiber unterstützt Write-Protect  
0: Drive-Treiber unterstützt Write-Protect nicht  
1: 1: Drive-Treiber unterstützt Read-Protect  
0: Drive-Treiber unterstützt Read-Protect nicht  
2: 1: Drive-Treiber unterstützt Track-I/O  
0: Drive-Treiber unterstützt Track-I/O nicht  
3: 1: Drive-Treiber unterstützt Formatieren eines Track  
0: Drive-Treiber unterstützt Formatieren eines Track  
nicht  
4: 1: Medium ist partitioniert  
0: Medium ist nicht partitioniert  
5: reserviert (hier 0)  
6: reserviert (hier 0)  
7: reserviert (hier 0)

DRV\_STAT1: byte 0: 1: Drive ist Write-Protected (Softprotect im BIOS)  
0: Drive ist nicht Write-Protected (Softprotect only)  
1: 1: Drive ist Read-Protected (Softprotect im BIOS)  
0: Drive ist nicht Read-Protected (Softprotect only)  
2: reserviert (hier 0)  
3: reserviert (hier 0)  
4: reserviert (hier 0)  
5: reserviert (hier 0)  
6: reserviert (hier 0)  
7: reserviert (hier 0)

RES\_01: byte reserviert (hier 0)  
RES\_02: byte reserviert (hier 0)

DRV\_PART2: byte Drive-Nummer des 2. logischen Drives bei partitionierten  
Medien (§adrv) (255: 2. Drive existiert nicht)

DRV\_PART3: byte Drive-Nummer des 3. logischen Drives bei partitionierten  
Medien (§adrv) (255: 3. Drive existiert nicht)

DRV\_PART4: byte Drive-Nummer des 4. logischen Drives bei partitionierten

Medien (§adrv) (255: 4. Drive existiert nicht)

RES_03:	byte	reserviert (hier 0)
RES_04:	byte	reserviert (hier 0)
DRV_HEAD:	byte	Anzahl der Oberflächen - 1
DRV_HDIO:	byte	Anzahl der Oberflächen, die mit einer einzigen I/O-Operation bearbeitet werden können - 1
DRV_TRK:	word	Anzahl der Spuren pro Oberfläche - 1
TRK_BYTE:	word	Anzahl der Bytes pro Spur (nur eine Oberfläche)
TRK_READ:	word	Adresse der Funktion, die eine komplette Spur (evtl. über mehrere Oberflächen) liest. Sektorverschränkung (XLT) wird hier nicht beachtet. Die Routine startet im Common-Memory und kann auch vom Anwenderprogramm direkt angesprochen werden.
TRK_WRITE:	word	wie TRK_READ, jedoch schreiben
TRK_FMT:	word	Formatieren einer Spur unter Beachtung der angegebenen Oberflächen (DMA und DMA_BNK sind ohne Bedeutung)
RES_05:	byte	reserviert (hier 0)
RES_06:	byte	reserviert (hier 0)
RES_07:	byte	reserviert (hier 0)
RES_08:	byte	reserviert (hier 0)
RES_09:	byte	reserviert (hier 0)
RES_10:	byte	reserviert (hier 0)
RES_11:	byte	reserviert (hier 0)
RES_12:	byte	reserviert (hier 0)
RES_13:	byte	reserviert (hier 0)
RES_14:	byte	reserviert (hier 0)

hd 18t form

Parameterübergabe für TRK\_READ, TRK\_WRITE und TRK\_FMT:

Eingabe: de: .TRK\_DATA:

DRIVE:	byte	Drive-Nummer (§adrv)	→ (@adrv)
TRACK:	word	Track-Nummer (§trk)	→ (@trk)
DMA:	word	DMA-Adresse (§dma)	
DMA_BNK:	byte	DMA-Bank (§dbnk)	
HEAD0:	byte	Start-Oberfläche	→ (@ <del>head</del> head0)
HEAD1:	byte	End-Oberfläche	→ (@head1)

cpytrk de = trk\_data

de = dph

Rückgabe: A: Fehlercode (wie für BIOS READ/WRITE definiert) 3 = Hd defekt

- 0 = Kein Fehler
- 1 = Phys. Fehler
- 2 = R10
- 3 = Hd defekt
- FF = Media changed

TRK\_DATA befindet sich in der DMA-Bank, wenn diese die Bank 0 oder 1 ist. Ist die DMA-Bank von 0 oder 1 verschieden, so befindet sich TRK\_DATA in der Bank 0. Anders ausgedrückt: TRK\_DATA befindet sich in der Bank 1, wenn DMA\_BNK = 1 gilt, sonst in Bank 0.

offs. ≤ 10 f.  
1. Laufwerk

**CBIOS-Erweiterungen:**

**Unterfunktion 15**

**Funktion:** XYDPH vom CBIOS holen  
**Eingabe:** DE enthält die Laufwerksnummer  
HL zeigt auf ein Datenfeld mit 32 Byte Länge  
A enthält die 15 (Funktionscode)  
**Rückgabe:** Das Datenfeld enthält den XYDPH des Laufwerkes

**Unterfunktion 16**

**Funktion:** XYDPH.DRV\_STAT1 übergeben  
**Eingabe:** DE enthält die Laufwerksnummer  
C enthält den neuen Wert XYDPH.DRV\_STAT1  
A enthält die 15 (Funktionscode)  
**Rückgabe:** --- keine ---

XXDPH: (für Winchester-Laufwerke am XEBEC-Controller)

byte	Anzahl der Spuren (höheres Byte)
byte	Anzahl der Spuren (niedrigeres Byte)
byte	Anzahl der Köpfe (1 bis 8)
byte	Anfangsspur für reduzierten Schreibstrom (höheres Byte)
byte	Anfangsspur für reduzierten Schreibstrom (niedrigeres Byte)
byte	Anfangsspur Precompensation (höheres Byte)
byte	Anfangsspur Precompensation (niedrigeres Byte)
byte	ECC - Länge
byte	Optionsbyte für XEBEC Controller
byte	physikalischer Skew-Faktor (im Normalfall: 6)
word	Adresse der Formatieroutine

ab hier folgt der XDPH (für alle Laufwerke gleich)

Die oben anzugebenden Werte sind den Datenblättern der Winchester-Laufwerke zu entnehmen. (Teilweise befinden sich Angaben im Anhang der XEBEC-Beschreibung.)

XXDPH: (für RAMDISK, EPROMDISK, CMOS-RAMDISK und SMB-RAMDISK)

word	Adresse der Formatieroutine (formatiert wird nur das Inhaltsverzeichnis)
------	--

ab hier folgt der XDPH (für alle Laufwerke gleich)

XXDPH: (für Winchesterlaufwerke am HDC 5)

-19	byte	offset (Platz für Directory-Backup)
-18	byte	physikalischer Skew-Faktor (beim Formatieren)
-17	word	Anzahl der Spuren pro Oberfläche
-15	byte	Anzahl der Oberflächen
-14	byte	Stepratezeit
-13	byte	RWC (Start für Write-Precompensation und reduzierten Schreibstrom)
-12	word	Adresse der Formatieroutine

ab hier folgt der XDPH (für alle Laufwerke gleich)

## - DPB = Disk-Parameter-Block (einen pro Laufwerk)

Der Disk-Parameter-Block enthält die physikalischen Werte der Diskette für CP/M PLUS.

Der DPH für RAMDISK und Winchester ist statisch. Hier müssen alle Werte definiert sein. (Wird normalerweise durch CPMINST erledigt.)

Der DPH für Diskettenlaufwerke wird dynamisch der jeweiligen Diskette angepasst. Gesetzt sein müssen für GENCPM die Werte BSH, BLM, DSM, DRM, CKS, PSH und PHM. Hier müssen immer die größten im Betrieb möglichen Werte eingesetzt werden, sonst reichen die von GENCPM reservierten Felder im Speicher nicht aus und das System stürzt ab. Dies wird normalerweise durch CPMINST erledigt.

## DPB..:

word	SPT = Sektoren pro CP/M Spur (hier können auch Vorder- und Rückseite gemeint sein!). Ein CP/M-Sektor hat immer 128 Byte Länge!
byte	BSH = Block-Shift-Faktor (3 = 1K, 4 = 2K, 5 = 4K, 6 = 8K, 7 = 16K Blockgröße)
byte	BLM = Block-Maske (7 = 1K, 15 = 2K, 31 = 4K, 63 = 8K, 127 = 16K Blockgröße)
byte	EXM = Extent-Maske (muß aus Blockgröße und Kapazität bestimmt werden, siehe System Guide)
word	DSM = Kapazität in Blocks - 1
word	DRM = Maximalzahl der Directory-Einträge. Ein 128 Byte-Sektor enthält 4 Einträge.
byte	AL0
byte	AL1
	AL0 und AL1 ergeben einen Bit-Vektor, der die für das Directory benötigten Blocks schützt. Das MSB von AL0 entspricht dem Block 0, das LSB von AL1 entspricht dem Block 15.
word	CKS = Checksum-Vektor-länge muß berechnet werden: $CKS = ( DRM / 4 ) + 1$ Das höchste Bit definiert das jeweilige Laufwerk als pseudo Festplatte. D.h. die Prüfung, ob die Diskette gewechselt wurde, findet nicht mehr immer statt. Die Schreib- bzw. Leseroutinen erkennen bei bestimmten internen Formaten ein Wechseln der Diskette.
word	OFF = OFFSET = Lage der Directory-Spur
byte	PSH = Sektor-Shift-Faktor. 0 = 128 byte, 1 = 256 byte, 2 = 512 byte, 3 = 1024 byte pro phys. Sektor. (4 und 5 sind nicht zulässig)
byte	PHM = Sektor-Maske (0 = 128 byte, 1 = 256 byte, 3 = 512 byte, 7 = 1024 byte pro Sektor)

### - Sektorverschränkungsliste

Die XLT (sector-translation-table) enthält ein Byte für jeden physikalischen Sektor der CP/M-Spur (kann Vorder- und Rückseite umfassen). CP/M ruft die Verschränkroutine mit Sektornummern ab 0 auf, und erwartet eine Nummer ab 1 zurück. Ein Beispiel: (eine CP/M-Spur enthält 8 phys. Sektoren)

CP/M Aufruf	wirklicher Sektor
0	1
1	3
2	5
3	7
4	2
5	4
6	6
7	8

Damit ergibt sich die XLT zu:

```
byte          1,3,5,7,2,4,6,8,0....0 (44 mal 0)
```

Für die Sektorverschränkungsliste sind pro Laufwerk 52 byte reserviert. D.h. mehr als 52 Sektoren pro Spur sind nicht zulässig! Die Sektorverschränkungsliste wird dynamisch in Abhängigkeit der jeweiligen Diskette erstellt. Im Quelltext muß nur der entsprechende Platz reserviert werden! Die Sektorverschränkungsliste wird bei Winchester und RAMDISK nicht benutzt. Daher haben diese Sektornummern ab 0, während Floppys Sektornummern ab 1 haben.

### - HASH-Tabelle (eine pro Laufwerk)

Die Hash-Tabelle wird von CP/M PLUS benötigt, um schneller Directory-Einträge zu finden. Die HASH-Länge muß mindestens  $4 * (DRM + 1)$  betragen.

Der Wert OFFFEH als Adresse der HASH-Tabelle (Platzreservierung durch GENCPM) kann immer benutzt werden. GENCPM fragt dann, ob Hashing gewünscht wird.

Bei Winchesterlaufwerken (2048 Einträge im Inhaltsverzeichnis) sollte immer mit Hashing gearbeitet werden, da sonst die Suchzeiten im Inhaltsverzeichnis zu lang werden.



Das Modul DISKDEF.LIB:

In der Bibliothek **DISKDEF.LIB** sind die Makros enthalten, die die Parameter der implementierten Floppy- und Winchester-Laufwerke definieren. Der MACRO-name ist in SYSDEF.DAT eingetragen; somit kann CPINST nach diesen Laufwerken fragen.

Als Beispiel für ein Floppylaufwerk wird das Makro TEAC55F betrachtet:

```
TEAC55F macro
size set 5 ; 5,25" drive
sides set 2 ; double sided
tracks set 80 ; 80 track / 96 tpi
step set 3 ; 3ms step-time
hdload set 50 ; 50ms headload-time
drive set true
endm
hidden set true ; high density possible
```

Das Floppylaufwerk TEAC55F ist ein slimline 5,25" 96 tpi Typ mit 2 Seiten. Somit folgt: Größe = 5,25", also SIZE = 5; zweiseitig, also SIDES = 2; 80 Spuren pro Seite, also TRACKS = 80; 3ms Spur zu Spur Zeit, also STEP = 3; 50ms Kopfladezeit, also HDLOAD = 50; die Variable DRIVE muß immer auf TRUE gesetzt werden.

Als Beispiel für ein Winchesterlaufwerk wird das Makro für die RODIME R0204E betrachtet:

```
R0204E macro ; 54MByte Winchester
tracks set 640 ; 640 Spuren / Oberfläche
heads set 8 ; 8 Köpfe
redcur set 210 ; reduzierter Schreibstrom ab Spur 210 (XEBEC-Wert)
precom set 0 ; Schreibvorkompensation ab Spur 0 (XEBEC-Wert)
ecc set 11 ; ecc data burst length (XEBEC-Wert)
rwc set 0 ; RWC für WD1010 (redcur und precom)
option set 0000$0111b ; 15us buffered step (XEBEC-Wert)
srt set 0 ; 35us fast step (WD1010)
capac set 640*17*4 ; 640 Spuren, 8 Köpfe, 17 512Byte-Sektoren (XEBEC)
hcapac set 639*8*9 ; 640 Spuren, 8 Köpfe, 9 1KByte-Sektoren (WD1010)
trkoff set 1 ; reservierte Spuren für Directory-Backup (WD1010)
drive set true
endm
```

Die Kapazität eines Winchesterlaufwerks in KByte berechnet sich wie folgt:

```
XEBEC: tracks * (heads/2) * 15
WD1010: (tracks-trkoff) * heads * 9
```

Der XEBEC-Controller ist für 512Byte zu jumpern! (Jumper W2 / SS nach 5 statt nach 2). Die anderen Werte werden den technischen Unterlagen zum Controller und Laufwerk entnommen.

Beim HDC-5 (WD1010-Chip) wird ein zweites Directory angelegt. Hierfür sind Spuren zu reservieren (trkoff): bei bis zu 3 Köpfen 4 Spuren, bei zu zu 7 Köpfen 2 Spuren, bei 8 Köpfen 1 Spur.

Bei der HDC-5 (WD1010-Controller-Chip) ist für RWC die Anfangsspur durch 4 einzutragen. Das RWC-Register gilt sowohl für reduzierten Schreibstrom wie auch für Schreibvorkompensation. Falls diese beiden Werte nicht gleich sind, kann häufig in RWC der Wert für die Schreibvorkompensation gesetzt werden, da moderne Laufwerke einen Mikroprozessor enthalten, und dieser die Startspur für den reduzierten Schreibstrom intern berechnet. In diesem Fall ist im Laufwerksinterface der pin 2 (reduced current) nicht benutzt. Bei der HDC-5 können die folgenden step-rate-times eingestellt werden:

srt	step rate time
0	35 $\mu$ s
1	0,5ms
2	1,0ms
3	1,5ms
4	2,0ms
5	2,5ms
6	3,0ms
7	3,5ms
8	4,0ms
9	4,5ms
A	5,0ms
B	5,5ms
C	6,0ms
D	6,5ms
E	7,0ms
F	7,5ms

Das Modul FUTIL:

FUTIL ist ein universeller Treiber für alle Floppycontrollerkarten mit einem NEC  $\mu$ PD765 Controller- IC. FUTIL enthält nur allgemeine Routinen. Die kartenspezifischen Teile sind in anderen Modulen enthalten. Somit sind Änderungen in FUTIL nicht notwendig. Änderungen sind zur Anpassung von Laufwerken nur in DISKS erforderlich!

Übersicht der Routinen und Tabellen in FUTIL

SPEC	= specify timing parameter SPEC wird bei jedem SELECT eines Laufwerkes aufgerufen, es übergibt die Zeiten des gewählten Laufwerks an den $\mu$ PD765, somit sind auch Laufwerke mit unterschiedlichem Timing gleichzeitig verwendbar.
LOGIN	erkennt das Format der verwendeten Diskette durch Vergleich mit aktivierten Formatdaten. LOGIN wird beim ersten Zugriff auf ein Laufwerk oder beim nächsten Zugriff nach einem RESET DISK SYSTEM aufgerufen.
NOTLOG	meldet Fehler bei LOGIN. Als Fehlermeldung erscheint i.allg. "INVALID DRIVE". Dies kann folgende Ursachen haben: kein Laufwerk, keine oder falsch eingelegte Diskette oder Diskette mit nicht erkanntem Format.
INVERT	invertiert einen Sektor Daten. (wird für inverse Aufzeichnung z.B. ZDOS benötigt)
RWUTIL	sucht die entsprechende Spur und bereitet das Lese- oder Schreibkommando für den $\mu$ PD765 vor.
FRMCMD	sucht die entsprechende Spur und bereitet das Formatierkommando und die ID-Informationen für den $\mu$ PD765 vor.
FRMNXT	berechnet die nächste zu formatierende Spur. Formatiert wird immer zuerst die gesamte Vorderseite und erst danach (falls notwendig) die Rückseite.
RESULT	holt Resultate vom $\mu$ PD765 (nach jedem Kommando)
CMFD	übergibt ein Kommando an den $\mu$ PD765
ERROR	analysiert ein Resultat und wählt Fehlermeldungen aus.
SENDM	sendet Fehlermeldungen
W765	wartet 24 $\mu$ s vor jedem Zugriff auf das Statusregister des $\mu$ PD765.

Datenbereiche in FUTIL:

RETRY	ist der Wiederholungszähler bei Fehlern
COUNT	ist der voreingestellte Maximalwert für RETRY
SEKCMD	ist die Kommandotabelle für Suchoperationen
CMD	ist die Kommandotabelle für Schreib- oder Lese-Operationen
STO	ist die Resultatstabelle
COMPAR	enthält die Parameter der letzten Fremddiskette
FRDATA	ist der Bereich, in dem die Sektor-IDs für das Formatieren bereitgestellt werden.

Das Modul FDC2:

FDC2 ist der kartenabhängige Treiber für die Floppyschnittstelle der FDC II-Karte. FDC2 benutzt die Routinen aus FUTIL. Die Routinen in FDC2 werden vom System über die Adressen im DPH aufgerufen. Somit müssen also die auf FDC2 bezogenen Namen in den DPH eingetragen sein. (Wird von CPMINST erledigt!)

Hardware:

FDC2 bezieht sich auf die FDC II Floppy-/Winchestercontrollerkarte. Diese Karte kann gleichzeitig bis zu 3 5,25" und 4 8" Floppylaufwerke bedienen. Laufwerke werden ab Adresse 0 gejumpt, d.h. es gibt sowohl ein 5,25" Laufwerk mit der eingestellten Adresse 0 wie auch ein 8"-Laufwerk mit der Adresse 0. Bei 5,25" Laufwerken kann mittels eines Jumpers die Lage des READY-Signals gewählt werden. Die verwendeten Laufwerke müssen ein READY-Signal zur Verfügung stellen. Bei 5,25" Laufwerken wird der Motor gesteuert (der Motor wird auch bei Zugriff auf die 8" Laufwerke gestartet). Die Motornachlaufzeit ist programmierbar. Die FDC II Karte muß an der Stelle mit der höchsten Interrupt-Priorität stecken. Es ist nur eine FDC II-Karte pro System zulässig. Zum Betrieb der FDC II-Karte sind Interrupt zwingend erforderlich.

Hardwareeinstellungen und Änderungen:

Bei den FDC II-Karten bis rev.4 sind folgende Änderungen bzw. Jumpereinstellungen erforderlich:

1. IC 20 ist ein 74HC74 und nicht 74LS74.
2. pin 11 von IC 24 ist über einen Widerstand von 1K5 mit dem Ausgang des Oszillators zu verbinden.
3. J3 selektiert TEAC-Laufwerke (READY auf pin 34), J4 selektiert Laufwerke mit READY auf pin 6.
4. Bei 6MHz Systemtaktfrequenz ist der Jumper J3 zu setzen.
5. Das dynamische RAM auf der FDC II ist auf Page 1 zu jumpern, d.h. 3 Jumper sind zu setzen, der 4. Platz neben R1 bleibt offen.
6. Die FDC II-Karte muß auf dem Steckplatz mit der höchsten Priorität stecken. (Bei unseren Busplatten darf kein Steckplatz oberhalb der Bestückungsseite der FDC II-Karte mehr benutzt sein.

Routinen in FDC2:

FD2INI	initialisiert die FDC II-Karte beim ersten Aufruf. Weitere Aufrufe werden ignoriert.
FD2LOG	ist die LOG-In-Routine für eins der am FDC II angeschlossenen Laufwerke.
FD2ID	liest die ID-Information einer Diskette.
FD2WR	ist die Schreibroutine für einen Sektor.
FD2RD	ist die Leseroutine für einen Sektor.
FD2FRM	ist die Formatieroutine für eine komplette Diskette.

Diese Routinen sollen nur über ihre Adressen im DPH aufgerufen werden. Direkte Aufrufe sind nicht zulässig. Generell gilt, daß beim Aufruf das Registerpaar DE die DPH-Adresse des selektierten

Laufwerkes enthalten muß.

### Das Modul FDC85:

FDC85 ist der kartenabhängige Treiber für die FDC 8/5-Karte mit 8" Floppylaufwerken; für 5,25" Floppylaufwerke wird als Treiber das Modul FDC5 verwendet. FDC85 benutzt die Routinen aus FUTIL. Die Routinen in FDC85 werden vom System über die Adressen in DPH aufgerufen. Somit müssen also die auf FDC85 bezogenen Namen in den DPH eingetragen sein. (Wird von CPMINST erledigt!)

### Hardware:

FDC85 bezieht sich auf die FDC8/5 Floppycontrollerkarte. Diese Karte bedient mit dem FDC85-Modul als Treiber 3 8" Floppylaufwerke. Laufwerke werden ab Adresse 0 gejumpert. Die verwendeten Laufwerke müssen ein READY-Signal zur Verfügung stellen. Die Motornachlaufzeit beträgt ca. 5 bis 10 Sekunden. Die FDC8/5-Karte wird im **DMA-Mode** betrieben, es ist darauf zu achten, daß die BAI/BAO Daisy-Chain existiert. Als Adressprom (IC3) wird das 4er Prom benutzt (Adressen: 04h..07h)

### Hardwareeinstellungen:

1. Die Karte sollte immer mit einem Datenseparator (FDC9216) versehen sein. (kann bei älteren Revisionen nachgerüstet werden!)
2. 4er Adressprom verwenden! (Normalerweise werden die Karten mit dem 0er Adressprom ausgeliefert!)
3. Wenn die FDC8/5 auf der höchsten Priorität der BAI/BAO-Kette liegt (bzw. als einzige einen DMA hat), ist der Jumper J1 zu setzen.
4. J5 wählt die Write-Precompensation-Zeit aus. (i.allg. sollte J5 nach unten, d.h. auf 125ns gesteckt werden.)
5. Vor dem Anschluß von Laufwerken ohne READY-Signal wird gewarnt. (Es treten bei bestimmten Systemzuständen Fehler auf, Disketten können zerstört werden!)

### Routinen in FDC85:

FD8INI initialisiert die FDC8/5-Karte beim ersten Aufruf. Weitere Aufrufe werden ignoriert.  
 FD8LOG ist die LOG-In-Routine für eins der am FDC8/5 angeschlossenen Laufwerke.  
 FD8ID liest die ID-Information einer Diskette.  
 FD8WR ist die Schreibroutine für einen Sektor.  
 FD8RD ist die Leseroutine für einen Sektor.  
 FD8FRM ist die Formatieroutine für eine komplette Diskette.

Diese Routinen sollen nur über ihre Adressen im DPH aufgerufen werden. Direkte Aufrufe sind nicht zulässig. Generell gilt, daß beim Aufruf das Registerpaar DE die DPH-Adresse des selektierten Laufwerkes enthalten muß.

Das Modul FDC5:

FDC5 ist der kartenabhängige Treiber für 5,25" Laufwerke an einer FDC 5-, DPS-IO- oder FDC 8/5-Karte; für 8" Floppylaufwerke wird als Treiber das Modul FDC85 in Verbindung mit einer FDC 8/5-Karte verwendet. FDC5 benutzt die Routinen aus FUTIL. Die Routinen in FDC5 werden vom System über die Adressen im DPH aufgerufen. Somit müssen also die auf FDC5 bezogenen Namen in den DPH eingetragen sein. (Wird von CPMINST erledigt!)

Hardware:

FDC5 bezieht sich auf die FDC8/5-, DPS-IO- oder FDC5-Floppycontrollerkarte. Diese Karten bedienen mit dem FDC5-Modul als Treiber drei 5,25" Floppylaufwerke. Laufwerke werden ab Adresse 0 gejumpt. Die verwendeten Laufwerke müssen ein READY-Signal zur Verfügung stellen. Die Motornachlaufzeit beträgt ca. 5 bis 10 Sekunden. Die FDC 8/5-Karte wird im **NON-DMA-Mode** betrieben. Als Adreßprom (IC3) für die FDC8/5 wird das 0er Prom (1068.1) benutzt (Adressen: 00h..03h)

Hardwareeinstellungen:

1. FDC5 und FDC 8/5 sollten immer mit einem Datenseparator (FDC9216) versehen sein. (kann bei älteren Revisionen nachgerüstet werden!)
2. 0er Adressprom verwenden! (Normalerweise werden die Karten mit dem 0er Adressprom ausgeliefert!)
3. J1 ist auf der FDC8/5-Karte nicht zu setzen.
4. J5 wählt auf der FDC8/5-Karte die Write-Precompensation-Zeit aus, er sollte i.allg. nach oben, d.h. auf 250ns gesteckt werden.)
5. Vor dem Anschluß von Laufwerken ohne READY-Signal wird gewarnt. (Es treten bei bestimmten Systemzuständen Fehler auf, Disketten können zerstört werden!)
6. Bei Laufwerken mit **READY** auf **pin 34** ist ein Draht von Pin 34 nach Pin 6 zu ziehen (FDC5 bzw. FDC 8/5). Eine evtl. an Pin 34 führende Leitung ist vorher abzutrennen (FDC 5)! Bei der DPS-IO-Platine ist für READY ein Jumper vorhanden!

Routinen in FDC5:

FD5INI initialisiert die FDC8/5-, DPS-IO- oder FDC5-Karte beim ersten Aufruf. Weitere Aufrufe werden ignoriert.  
 FD5LOG ist die LOG-In-Routine für eins der am FDC8/5, DPS-IO oder FDC5 angeschlossenen 5,25" Laufwerke.  
 FD5ID liest die ID-Information einer Diskette.  
 FD5WR ist die Schreibroutine für einen Sektor.  
 FD5RD ist die Leseroutine für einen Sektor.  
 FD5FRM ist die Formatieroutine für eine komplette Diskette.

Diese Routinen sollen nur über ihre Adressen im DPH aufgerufen werden. Direkte Aufrufe sind nicht zulässig. Generell gilt, daß beim Aufruf das Registerpaar DE die DPH-Adresse des selektierten Laufwerkes enthalten muß.

### Das Modul EPC1FDC

EPC1FDC bedient bis zu 3 Floppylaufwerke an der EPC 1.6-Karte. 8"-Laufwerke werden nicht unterstützt. EPC1FDC benutzt die Routinen aus FUTIL. Alle Routinen in EPC1FDC werden über die Adressen im DPH aufgerufen.

### Hardware:

EPC1FDC bezieht sich auf den Floppyteil des Einplatinenrechners EPC 1.6. Es können bis zu 3 Laufwerke angeschlossen werden. (3,5" und/oder 5,25"). Floppylaufwerke werden ab Adresse 0 gejumpt. Die Adresse 3 darf nicht benutzt werden. Alle Laufwerke müssen ein READY-Signal zur Verfügung stellen. Bei Anschluß von unterschiedlichen Laufwerken ist sicherzustellen, daß diese untereinander verträglich sind.

### Hardwareeinstellungen:

Der EPC 1.6 hat einen 34pol. Pfostenstecker für 5,25"-Laufwerke. Mit Jumper J6 kann die READY-Leitung des Interfaces entweder auf Pin 6 (Stellung: 3-4) oder 34 (Stellung: 1-2) gelegt werden. Achtung: Das Interface ist nicht OPEN-COLLECTOR. Falscher Anschluß zerstört die Treiber.

### Routinen in EPC1FDC

EP1INI	initialisiert die Hardware beim ersten Aufruf, weitere Aufrufe werden ignoriert
EP1LOG	ist die LOGIN-Routine für die angeschlossenen Laufwerke
EP1ID	liest die ID-Information von der Diskette
EP1FRM	formatiert eine vollständige Diskette
EP1RD	liest einen Sektor von der Diskette
EP1WR	schreibt einen Sektor auf die Diskette

Das Modul DDATA:

Das Modul DDATA enthält die aktiven Diskformate, die bei einem LOGIN automatisch erkannt werden. DDATA enthält Platz für bis zu 9 unterschiedliche Fremdformate. Hiervon sind bereits 4 belegt (ZDOS 5,25" 80 und 40 Track, ZDOS 8" DD und CP/M A1). Änderungen der aktiven Formate sind dynamisch mit dem Dienstprogramm DM (Disk-Manager) möglich. Disk-Formate sind wie folgt definiert:

Formatnummer	0 = leer, 1...253 = gültig, 255 = Ende 254 = Zwischenformat beim Analysieren
***** XXDPH-Werte *****	
Laufwerksgröße	0 = 8", 1 = 5.25"
FM/MFM	0 = FM, 40H = MFM, COH = MFM mit FM-Systemspuren
Sektorgröße	0 = 128Byte, 1 = 256Byte,...
Disketten-Nummer	normal = 0
Spur 0	normal = 0
Seiten	1 oder 2
Spuren	35, 40, 77, 80 ... (für eine Seite!)
Sektoren	einer Spur (für einseitig!)
r/w-gap-Länge	z.B.: 10 (siehe Modul: DISKS)
format-gap-Länge:	z.B.: 30 (siehe Modul: DISKS)
format-skew	physikalische Sektorverschränkung 1 = kontinuierlich, 2 = jeder zweite,...
normal/invers	normale (0) oder inverse (1) Datenaufzeichnung
intern	fremde (0) oder interne Diskette mit Beschreibungsblock (1)
***** DPB-Werte *****	
SPT	128Byte Sektoren pro Spur (evtl. beidseitig)
BSH	data allocation block shift factor
BLM	data block mask
EXM	extent mask
DSM	Kapazität in Blocks
DRM	Anzahl der Einträge im Inhaltsverzeichnis
ALO, AL1	Vektor der reservierten Blocks
CKS	Prüfvektorlänge
OFF	Anfangsspur des Inhaltsverzeichnisses
PSH	physical record shift factor
PHM	physical record mask
***** XLT-Werte *****	
52 Byte für die Sektorverschränkungsliste (1 .. 52)	



Das Modul XEBEC:

XEBEC bedient bis zu 2 Winchesterlaufwerken. Benötigt werden die FDC II als SASI-Karte und ein XEBEC S1410 als SASI-Controller. Das Modul XEBEC benutzt einige Routinen aus FUTIL. FUTIL ist in jedem Fall im CBIOS enthalten.

**Hardware:**

1. Auf der FDC II Floppy- / Winchester-Controller-Karte sind keine SASI-spezifischen Jumper.
2. Der XEBEC Controller S1410 (rev. E) ist auf 512Byte Sektorgröße zu jumpern (W2 in Stellung 5, Auslieferungszustand ist leider 2).  
Der XEBEC Controller muß auf der SASI Adresse 0 liegen, d.h. die Brücke neben dem 50pin Verbinder muß in Stellung 0-S sein (Auslieferungszustand!).
3. XEBEC-kompatible Controller funktionieren in den meisten Fällen nicht zuverlässig, da die Software sich beim Datentransfer darauf verläßt, daß der XEBEC Controller schneller als das System ist. Dies ist bei kompatiblen Controllern nicht unbedingt zutreffend!

**Routinen in XEBEC:**

<b>HDFORM</b>	formatiert die gesamte Winchester. Ein Formatieren einzelner Spuren ist nicht vorgesehen. Das Formatieren einer 27MByte Winchester dauert ca. 5 Minuten!
<b>HDLOGI</b>	selektiert den XEBEC-Controller und überprüft, ob das angewählte Laufwerk READY ist. Falls kein Controller zu finden ist, oder das Laufwerk nicht innerhalb einer vorgegebenen Wartezeit READY wird, wird dieses Laufwerk aus der DRIVE-TABLE in DISKS ausgetragen. Es ist danach nicht mehr ansprechbar.
<b>HDINIT</b>	übergibt die im Winchester-XXDPH eingetragenen Parameter an den XEBEC-Controller.
<b>HDREAD</b>	liest einen oder mehrere Sektoren von der Festplatte. In XEBEC wird die von CP/M gebotene Möglichkeit des MULTI-I/O benutzt. Bei Ausnutzung von MULTI-I/O können 4MByte in ca. 2 Minuten transferiert werden.
<b>HDWRIT</b>	schreibt einen oder mehrere Sektoren auf die Festplatte. Auch hier ist MULTI-I/O implementiert.
<b>SELECT</b>	wird nur im BOOTSTRAPLOADER benutzt. SELECT selektiert den XEBEC-Controller.

Fehlermeldungen werden in hexadezimaler Schreibweise ausgegeben. Sie entsprechen der Fehlerliste in den Unterlagen des XEBEC-Controllers.

Das Modul HDC5:

HDC5 bedient bis zu 2 Winchesterlaufwerke. Benötigt wird nur die HDC 5-Karte als Winchester-Controller. Das Modul HDC5 benutzt Routinen aus FUTIL.

HDC5 führt aus Sicherheitsgründen ein zweites Directory mit. Falls bei Directory-Lesezugriffen ein Fehler auftritt, wird ersatzweise das zweite Directory benutzt. Schreibzugriffe verändern jeweils beide Directories. Bei Lesefehlern im Directory erscheint die Warnung: DIRECTORY READ ERROR auf dem Bildschirm. I.allg. kann hier trotzdem weitergearbeitet werden. Diese Warnung sollte aber der Anlaß für ein sofortiges Backup der gesamten Winchester sein. Schreibzugriffe werden ab jetzt nicht mehr zugelassen.

HDC5 arbeitet mit physikalischen Sektoren mit 1KByte Größe. Eine Spur besteht aus allen Oberflächen. Erst danach wird zur nächsten Spur gestept.

**Hardware:**

1. Auf der HDC 5-Karte sind keine Jumper!
2. Achtung: Das Laufwerk 0 ist mit dem Daten-0-Stecker und das Laufwerk 1 mit dem Daten-1-Stecker zu verbinden!  
Es sind nur gleiche Laufwerke benutzbar.

**Routinen in HDC5:**

HD5INI wird nicht benutzt  
 HD5LOG führt das LOGIN für eine Winchester durch. Beim ersten Mal wird eine RECALIBRATE-Operation ausgeführt. Falls kein Controller vorhanden oder das Laufwerk nicht READY ist, wird es aus der \$DTBL ausgetragen.  
 HD5FRM formatiert die gesamte Platte (oder wenn SFLAG gesetzt ist nur genau eine Spur)  
 HD5RD liest einen Sektor von der Platte  
 HD5WR schreibt einen Sektor auf die Platte

**globale Variable in HDC5**

SFLAG wird nur beim Formatieren benutzt (Option: nur eine Spur formatieren)

Das Modul EPC2FDC

EPC2FDC bedient bis zu 3 Floppylaufwerke an der EPC II-Karte. Die Größe der Laufwerke wird mittels eines Jumpers auf der Karte eingestellt. EPC2FDC benutzt die Routinen aus FUTIL. Alle Routinen in EPC2FDC werden über die Adressen im DPH aufgerufen.

Hardware:

EPC2FDC bezieht sich auf den Floppyteil des Einplatinenrechners EPC II. Es können bis zu 3 gemischte Laufwerke angeschlossen werden. (3,5", 5,25" und/oder 8"). Floppylaufwerke werden ab Adresse 0 gejumpert. Die Adresse 3 darf nicht benutzt werden. Alle Laufwerke müssen ein READY-Signal zur Verfügung stellen. Bei Anschluß von unterschiedlichen Laufwerken ist sicherzustellen, daß diese untereinander verträglich sind.

Hardwareeinstellungen:

Die EPC II hat einen kombinierten Stecker für 5,25" und 8" Laufwerke. 5,25"-Laufwerke werden so angeschlossen, daß pin 34 auf pin 50 des Platinensteckers kommt. Mit Jumper **J4** wird die Größe des Laufwerks eingestellt.

Es sind folgende Kombinationen möglich:

1	2	3	4	5				
0	0	0	0	0	A = 5"	B = 5"	C = 5"	
0	0	0	0	-0	A = 8"	B = 8"	C = 8"	
0	-0	0	0	0	A = 8"	B = 5"	C = 8"	
0	0	-0	0	0	A = 5"	B = 5"	C = 8"	

hierbei stellt - einen gesetzten Jumper dar

Routinen in EPC2FDC

EP2INI	initialisiert die Hardware beim ersten Aufruf, weitere Aufrufe werden ignoriert
EP2LOG	ist die LOGIN-Routine für die angeschlossenen Laufwerke
EP2ID	liest die ID-Information von der Diskette
EP2FRM	formatiert eine vollständige Diskette
EP2RD	liest einen Sektor von der Diskette
EP2WR	schreibt einen Sektor auf die Diskette

Steuerparameter für Disketten

Der Disketeil des CBIOS 1.6 ist ein parametergesteuertes System. Es gibt mehrere Definitionstabellen für die Floppyformate.

**1. XXDPH - Werte in CBIOS**

DPH-35	drvsid	Seiten des Laufwerks
DPH-34	drvtrk	Spuren des Laufwerks
DPH-33	ptime	write-precompensation-time
DPH-32	srt	steprate and headunload
DPH-31	hlt	headload and DMA-Mode
DPH-30	trksft	40 Spur emulieren falls 1
DPH-29	slenl	Sektorlänge (lower byte)
DPH-28	slenh	Sektorlänge (higher Byte)
DPH-27	size	5.25" Diskette wenn 1, 8" wenn 0
DPH-26	fm	FM falls 0, MFM falls 40H, bit 7 = außen FM, innen MFM
DPH-25	ssize	N (kodierte Sektorgröße)
DPH-24	diskid	Nummer der Spur 0
DPH-23	number	Nummer des Vorderseitenkopfes
DPH-22	sides	Seiten der Diskette (1 = einseitig, 2 = doppel- seitig, 3 = abwechselnd)
DPH-21	mxtrk	Spuren der Diskette
DPH-20	mxsec	Sektoren pro Spur
DPH-19	gapl	R/W-Gaplänge
DPH-18	fgap	Format-Gaplänge
DPH-17	skew	Formatierskewfaktor
DPH-16	iflag0	bit 0 = normal/invers bit 1 = Sektor 0 oder 1 als erster bit 2 = 1: gleiche Kopfnummer auf beiden Seiten bit 3 = 1: Rückseite rückwärts steppen bit 4 = 1: ALPHATRONIC P3
DPH-15	iflag1	interne oder normale Diskette

Beschreibungsblock

Interne Disketten haben auf Spur 0 Sektor 1 einen Beschreibungsblock. In diesem sind die Werte der Diskette enthalten:

0:	E5 normale, 1A inverse Aufzeichnung
1:	immer = 0
2:	Seiten
3:	Spuren pro Seite
4:	Sektoren pro Spur
5:	R/W-Gaplänge
6:	Formatier-Gaplänge
7:	Formatier-Skew-Faktor
8..11:	CP/M Nummer des Systems
12:	FLAG (iflag0: siehe oben)
13:	interne oder normale Diskette (immer 0)
14..31:	--- reserviert ---
32..48:	DPB
49..100:	XLT
100..101:	Prüfsumme

Vergleichswerte in DDATA

Das Modul DDATA enthält eine Tabelle mit Vergleichswerten für das LOGIN einer Fremddiskette (d.h. einer Diskette ohne Beschreibungsblock). Ein gesetztes msb bei einem Vergleichswert veranlaßt, daß dieser beim LOGIN nicht berücksichtigt wird. Als Vergleichswerte werden folgende Werte benutzt:

```

Größe
Dichte (FM/MFM)
N (Sektorgröße)
Kopf0
Spur0
Seiten
Spuren (40 oder 80 d.h. 48 oder 96 tpi)
Sektoren

```

Beim LOGIN wird ein Eintrag in DDATA durchsucht und mit den erkannten Werten der Diskette verglichen. Bei Gleichheit werden die DDATA-Tabellenwerte in XXDPH, DPH und XLT des Laufwerks übernommen. Es kann also passieren, daß ein falsches Format übernommen wird, wenn seine Vergleichswerte für die aktuelle Diskette passen.

Ein DDATA - Tabelleneintrag ist 83Byte lang und hat die Form:

0	Byte	gültig	0 = not defined, 1...253 = valid, 255 = end, 254 = intermediate format during analyse disk
1	Byte	Größe	0 = 8", 1 = 5.25"
2	Byte	FM/MFM	0 = FM, 40h = MFM, 0c0h = FM / MFM
3	Byte	N	0 = 128Byte, 1 = 256Byte, 2 = 512Byte, ...
4	Byte	Kopf0	normal = 0
5	Byte	Spur0	normal = 0
6	Byte	Seiten	1, 2 oder 3 (0 = login-error)
7	Byte	Spuren	35, 40, 77, 80 ... (for 1 side)
8	Byte	Sektoren	einseitig
9	Byte	r/w-gap	
10	Byte	format-gap	
11	Byte	PSKEW	1, 2, 3 ...
12	Byte	FLAG	iflag0: siehe oben
13	Byte	----	reserved for future use
14	Word	SPT	128byte records / track
16	Byte	BSH	data allocation block shift factor
17	Byte	BLM	block mask
18	Byte	EXM	extent mask
19	Word	DSM	storage capacity in blocks
21	Word	DRM	total number of directory entries - 1
23	Byte	ALO	reserved directory blocks
24	Byte	AL1	reserved directory blocks
25	Word	CKS	directory check vector size
27	Word	OFF	number of reserved tracks
29	Byte	PSH	physical record shift factor
30	Byte	PHM	physical record mask
31	Byte	XLT	52 Byte sector translation table

DISKFORM.DAT

In der Datei Diskform.dat sind alle bereits analysierten Fremdformate enthalten. Jedes Format belegt genau den (logischen) Sektor mit seiner Nummer. Der Sektor 0 enthält die Version der DISKFORM.DAT. Diese Datei wird von den Dienstprogrammen FORMAT und DM benutzt. Innerhalb eines Sektors der DISKFORM.DAT gilt der folgende Aufbau:

0..29	Nummer und Formatname
30	Größe
31	density FM, MFM oder FM/MFM
32	N = Sektorgröße (0 = 128, 1 = 256 ...)
33	Nummer der Spur 0
34	Nummer des Kopf 0
35	Seiten der Diskette
36	Spuren pro Seite
37	Sektoren pro Spur
38	R/W-Gaplänge
39	Formatiergaplänge
40	Formatierskew
41	FLAG (iflag0: siehe oben)
42	normale oder interne Diskette (hier nur für FORMAT)
43..59	DPB
60..111	XLT

### 4.5 Character I/O

Das Modul **CHARIO.ASM** realisiert die zeichenbezogene Ein- und Ausgabe der maximal 16 physikalischen 'Character-Devices'. CHARIO wird als Source mitgeliefert, um die Anpassung an eigene Eingabe/Ausgabe-Geräte zu erleichtern. Änderungen in CHARIO.ASM sind dafür jedoch nicht notwendig. Die Quelle sollte lediglich zur Information über die Struktur der zeichenbezogenen Ein- und Ausgabe herangezogen werden.

Außer CHARIO.ASM werden noch die folgenden Dateien benötigt:

'MODEBAUD.LIB' , 'SYSDEF.LIB' , 'PORTS.LIB' , 'EPC7220.REL' ,  
'PRINTER.ASM' und 'USER.LIB'

Übersicht der Routinen und Tabellen in CHARIO:

#### 1. Im 'Common-Memory' vereinbarte Tabellen:

**§ctbl:** Gerätetabelle für zeichenbezogene Ein- und Ausgabe

Eine genaue Beschreibung der §ctbl kann dem CP/M 3 System Guide Seite 32ff entnommen werden.

**§xctbl:** erweiterte Gerätetabelle für zeichenbezogene Ein- und Ausgabe

§xctbl ist eine CBIOS 1.6-Erweiterung von §ctbl und wie folgt definiert:

Für jedes in §ctbl eingetragene Gerät ist ein Eintrag in §xctbl notwendig. Die Reihenfolge der eingetragenen Geräte in §xctbl ist identisch mit der Reihenfolge der Geräte in §ctbl. Für jedes Gerät sind folgende 8 Bytes eingetragen:

Byte 1: Kontrollbyte für serielle Geräte.

Die 8 Bit des Kontrollbytes haben folgende Bedeutung:

Bit 0 - 1: Das Gerät kann Hardware-Handshake  
0: Das Gerät kann keinen Hardware-Handshake

Bit 1 - 1: Hardware-Handshake aktiv  
0: Hardware-Handshake nicht aktiv

Bit 2 - 1: Gerät kann Paritätskontrolle durchführen  
0: Gerät kann keine Paritätskontrolle durchführen

Bit 3 - 1: Paritätskontrolle aktiv  
0: Paritätskontrolle nicht aktiv

- Bit 4 - 1: bei aktiver Paritätskontrolle erfolgt Kontrolle auf gerade Parität (Parity even)  
 0: bei aktiver Paritätskontrolle erfolgt Kontrolle auf ungerade Parität (Parity odd)
- Bit 5 - 1: Es wird ein Stop-Bit verwendet  
 0: Es werden zwei Stop-Bit verwendet
- Bit 6+7 - 00: Übertragung von 5 Bit pro Zeichen  
 10: Übertragung von 6 Bit pro Zeichen  
 01: Übertragung von 7 Bit pro Zeichen  
 11: Übertragung von 8 Bit pro Zeichen

Bei einer Übertragung von 5 oder 6 Bit pro Zeichen wird sowohl bei der Eingabe als auch bei der Ausgabe das 7. und 8. Bit auf 0 gesetzt. Bei einer Übertragung von 7 Bit pro Zeichen wird lediglich das 8. Bit auf 0 gesetzt. Bei einer Übertragung von 8 Bit pro Zeichen, wird weder bei Ausgaben noch bei Eingaben ein 8. Bit gelöscht!

#### Byte 2: TIME-OUT-Konstante

Die TIME-OUT-Konstante gibt die maximale Zeit in Sekunden an, bis zu der das angeschlossene physikalische Gerät nach einer Zeichenausgabe garantiert wieder bereit für die Annahme eines weiteren Zeichens ist. Falls das Gerät nach Ablauf dieser Zeit noch immer nicht bereit ist, wird auf der Console die Fehlermeldung 'TIME OUT name ? ' ausgegeben. 'name' ist dabei der Name des betreffenden Gerätes aus \$ctbl. Die mögliche Reaktion auf diese Fehlermeldung ist in dem Kapitel über das Modul 'BIOSKERN' in diesem Handbuch beschrieben. Eine TIME-OUT-Konstante von 0 unterdrückt die zuvor beschriebene Fehlermeldung und veranlaßt CBIOS 1.7 beliebig lange auf das angeschlossene Gerät zu warten.

#### Byte 3: Kontrollbyte für Slave-CPU-Baugruppen

Die 8 Bit des Kontrollbytes für Slave-CPU-Baugruppen haben folgende Bedeutung:

- Bit 0 - 1: Schnittstelle befindet sich auf dem Slave 186  
 0: Schnittstelle befindet sich nicht auf dem Slave 186
- Bit 1 bis 7: Reserviert für zukünftige Erweiterungen (in CBIOS 1.7 müssen diese Bits 0 sein).
- Byte 4 bis 8: Reserviert für zukünftige Erweiterungen (in CBIOS 1.7 müssen diese Bytes 0 sein).

\$xctbl wird von dem Dienstprogramm CONFIG ausgewertet und ggf. modifiziert. ?dvini beachtet bei der Initialisierung eines Gerätes die Einträge in \$ctbl und in \$xctbl.



Die Adresse von \$xctbl ist unmittelbar vor \$ctbl eingetragen. Ein externes Programm kann die Adresse von \$xctbl deshalb auf folgende Weise ermitteln:

```
- Aufruf der BIOS-Funktion 20 'DEVTBL'
  HL zeigt nun auf $ctbl
- dec hl
  ld d,(hl)
  dec hl
  ld e,(hl)
  DE zeigt nun auf $xctbl
```

## 2. In der Bank 0 vereinbarte Entries und Variablen:

**cioini:** Initialisierung des Programm-Moduls und Zuweisung der Geräte nach dem Ladevorgang

cioini muß aufgerufen werden, bevor irgend eine andere Routine aus CHARIO aufgerufen wird. cioini wertet die Informationen vom Urlader aus, initialisiert die Gerätezuweisung, initialisiert die systemtaktabhängigen Baudratentabellen und trägt den Systemtakt für spätere Auswertungen in der globalen Variablen sys\$clk ein.

**?dvini:** Initialisierung eines Gerätes gemäß den Einträgen in \$ctbl und \$xctbl

Eingaberegister: B - Gerätenummer

**?chris:** Eingabestatus

Eingaberegister: B - Gerätenummer  
Ausgaberegister: A - Statusergebnis (FF = READY / 00 = BUSY)  
Z - Flag gemäß Ergebnis in A

**?chros:** Ausgabestatus

Eingaberegister: B - Gerätenummer  
Ausgaberegister: A - Statusergebnis (FF = READY / 00 = BUSY)  
Z - Flag gemäß Ergebnis in A

**?chrin:** Eingabe

Eingaberegister: B - Gerätenummer  
Ausgaberegister: A - eingegebenes Zeichen  
Bemerkung: Es ist keine Statusabfrage erforderlich, da dies bereits im BIOSKERN geschehen ist.

**?chrout:** Ausgabe

Eingaberegister: B - Gerätenummer

C - auszugebendes Zeichen

Bemerkung: Es ist keine Statusabfrage erforderlich, da dies bereits im BIOSKERN geschehen ist.

**sys\$clk:** Systemtakt des Computers

Der Systemtakt wird vom Urlader ermittelt und an das Modul CHARIO.ASM übergeben. CHARIO speichert den vom Urlader übergebenen Wert in der Variablen 'sys\$clk' ab. Dieser Wert steht BIOS-intern jeder anderen Routine zur Verfügung.

Die Anpassung des Character-I/O an die verschiedenen Hardware-Komponenten geschieht durch bedingte Assemblierung. Die Auswahl der unterstützten Komponenten wird durch die Datei SYSDEF.LIB gesteuert. SYSDEF.LIB wird von dem Dienstprogramm CPMINST erzeugt.

In MODEBAUD.LIB sind einige Konstanten für die Benutzung in \$ctbl festgelegt (siehe CP/M 3 System Guide Seite 34). Die hier vorliegende Version wurde um einige Konstanten für \$xctbl erweitert.

PORTS.LIB enthält alle in CHARIO.ASM benötigten I/O-Ports, sowie die für die jeweiligen Geräte relevanten Statusbits.

PRINTER.ASM enthält Initialisierungen für eine grafikfähigen Drucker (wird nur zusammen mit dem EPC II benötigt).

Nähere Einzelheiten zu diesen Files können dem kommentierten Source entnommen werden.

**Erweiterung des CBIOS 1.7 um eigene zeichenbezogene Eingabe/Ausgabe-Routinen:**

Zur Erweiterung des CBIOS 1.7 um eigenen Character-I/O sind keine Modifikationen in CHARIO.ASM erforderlich. Zur Modifikation des Character-I/O sind in USER.LIB einige MACROs vordefiniert, die wie folgt zu modifizieren sind:

**inctbl:** Dieses Macro fügt einen oder mehrere Einträge in die 'Device Table' \$ctbl ein. Für jedes zusätzlich zu definierende Gerät sind diese Einträge hier hintereinander in einem Macro erforderlich.

**inxctbl:** Dieses Macro fügt einen oder mehrere Einträge in die erweiterte Geräte-Tabelle \$xctbl ein. Für jedes zusätzlich zu definierende Gerät sind diese Einträge hier hintereinander in einem Macro erforderlich.

**inchtbl:** Dieses Macro fügt einen oder mehrere Einträge in die CHARIO-Treibertabelle ein. Für jedes zu definierende

#### 4. CBIOS 1.7 4.5 Erweiterung des Character I/O durch den Anwender

Gerät ist ein Satz von 5 Adreßeinträgen hier hintereinander in einem Macro erforderlich.

**usersetio:** In diesem Macro kann eine abweichende Standard-I/O-Zuordnung nach dem Ladevorgang vereinbart werden. Zu diesem Zweck sind die Systemvariablen `%civec`, `%covec`, `%aivec`, `%aovec`, und `%lovec` aus dem System Control Block entsprechend zu setzen. Normalerweise wird die I/O-Zuordnung nach dem Ladevorgang aus den vom Urlader übergebenen Informationen ermittelt. In **CHARIO.ASM** ist die Konstante `maxdev` definiert worden. Diese Konstante gibt die maximal vorhandenen Devices an (siehe **CHARIO.ASM**). Eine abweichende Gerätevereinbarung kann deshalb wie folgt realisiert werden:

```
GERK    equ    (8000H) shr (maxdev-1)
lxi     h,GERK                ; vom Anwender definiertes Gerät
shld   vector                ; %civec, %covec ...
```

Alle Macros werden in **CHARIO.ASM** an der richtigen Stelle bereits aufgerufen und sind deshalb in **USER.LIB** bereits als 'leere' Macros definiert.

Zusätzlich zu den oben beschriebenen Macros sind für jedes zusätzliche Gerät noch folgende 5 Routinen zu schreiben:

<b>ini:</b>	Initialisierung der Device	siehe ?dvini
<b>instat:</b>	Input-Status der Device	siehe ?chris
<b>otstat:</b>	Output-Status der Device	siehe ?chros
<b>input:</b>	Input eines Zeichens	siehe ?chrin
<b>output:</b>	Output eines Zeichens	siehe ?chrout

Die Übergabekonventionen stimmen mit denen der zugehörigen **CHARIO**-Entries überein.

Die oben genannten Routinen müssen in einem eigenen Modul zur Verfügung gestellt werden. Aus diesem Grunde ist es auch notwendig, diese Routinen in **USER.LIB** als 'extrn' zu vereinbaren. Als externes Modul eignet sich **USER.ASM** besonders gut, da es bei jeder durch **CPMINST** generierten Systeminstallation mit assembliert und gelinkt wird. Falls nicht **USER.ASM** benutzt werden soll, muß in **CPMGEN.DAT** die **LINK**-Anweisung modifiziert werden.

Die Erweiterung des Character-I/Os mit Hilfe der in **USER.LIB** definierten Macros hat den Vorteil, daß bei späteren Änderungen in **CHARIO.ASM** durch den Hersteller die Erweiterungen des Anwenders davon nicht betroffen werden.

Es wird deshalb dringend empfohlen, Erweiterungen nur auf diese Art vorzunehmen.

### Der Terminaltreiber EPC7220

Das Modul EPC7220 gehört mit zum Character-I/O. Es emuliert ein Terminal auf dem Grafikbildschirm des EPC II. Zur Darstellung von Zeichen wird der vom Urlader LDRE1600 in den "Shadow-Common" geladene Zeichengenerator CHARGEN.COM benutzt. CHARGEN.COM ist ein komprimierter Zeichengenerator mit einer Zeigerstruktur am Anfang. Sie können sich einen eigenen CHARGEN und somit eigene Zeichen erzeugen, indem Sie CHARGEN.ASM editieren und dann die Prozedur SUBMIT CHARGEN starten. (Achtung: Assemblierzeit ca. 10 Minuten!)

Im Modul EPC7220 wird ein "intelligenter" Cursor benutzt. Der Cursor blinkt immer dann, wenn vom Rechner her eine Eingabe gefordert wird. Seine Blinkgeschwindigkeit ist einstellbar.

Das Bildschirmformat beträgt 800 Punkte in 264 Zeilen, bzw. 24 Zeilen mit je 88, 113 oder 132 Zeichen.

Die hier beschriebene Terminalsoftware emuliert im Alpha-Mode ein Lear Siegler ADM 3A. Im Grafik-Mode wird entweder ein Tektronix 4010 (ESC 6, default) emuliert oder direkte Grafik (ESC 5), d.h. Tektronix ohne Koordinatentransformation benutzt.

Ein CNTL § (hex-code: 00) von der Tastatur bewirkt eine Hardcopy des Bildschirms auf den (im Modul PRINTER.ASM angepaßten) grafikfähigen Drucker.

Im Alpha-Mode werden die folgenden Attribute unterstützt:

- invers
- schrägstellen
- unterstreichen
- fett
- hochstellen
- tiefstellen
- 88 Zeichen / Zeile
- 113 Zeichen / Zeile
- 132 Zeichen / Zeile
- deutscher oder US-ASCII Zeichensatz
- weitere Zeichensätze

Es stehen insgesamt ca. 12KByte für die Zeichensätze zur Verfügung. Da hier eine komprimierte Speicherung benutzt wird, reicht der Platz für insgesamt 8 Zeichensätze. Damit ergibt sich ein normaler und ein alternativer (momentan mit Spiegelschrift belegter) Zeichensatz:

- |                           |    |                                     |
|---------------------------|----|-------------------------------------|
| normaler Zeichensatz:     | 1. | 88 Zeichen / Zeile normal           |
|                           | 2. | 113 bzw. 132 Zeichen / Zeile normal |
|                           | 3. | Hoch- bzw. tiefgestellte Zeichen    |
|                           | 4. | 4010-Zeichensatz                    |
| alternativer Zeichensatz: | 5. |                                     |
|                           | 6. |                                     |
|                           | 7. |                                     |
|                           | 8. |                                     |

direkte Steuersequenzen: (Control-Codes)  
im Terminalmode (ADM 3A kompatibel)

01	CTRL	A	Markierungspunkt setzen
08	CTRL	H Backspace	Cursor eine Position nach links, nicht in die vorherige Zeile.
0A	CTRL	J Linefeed	Cursor um eine Position nach unten, in der letzten Zeile wird der Bildschirm hochgescrollt.
0B	CTRL	K Upline	Cursor um eine Position nach oben, in der obersten Zeile wird Upline ignoriert.
0C	CTRL	L Forespace	Cursor eine Position weiterbewegen, am Ende einer Zeile geht der Cursor in die nächste, falls wraparound an ist.
0D	CTRL	M Carriage Return	Cursor zurück an den Anfang der Zeile.
1A	CTRL	Z Clear Screen	Bildschirm löschen und Cursor nach links oben.
1B	CTRL	Ä Escape	Start einer Steuersequenz (ESCAPE-Sequenz)
1C	CTRL	Ö Point-Mode	4010 Point-Mode einschalten Siehe Grafik-Mode
1D	CTRL	Ü Vector-Mode	4010 Vector-Mode einschalten Siehe Grafik-Mode
1E	CTRL	^ Home	Cursor nach links oben
1F	CTRL	_ New Line	Cursor in die erste Position der nächsten Zeile (Scroll falls erforderlich)

ESCAPE-Sequenzen im Terminalmode (ADM 3A kompatibel)

1B 28	ESC (	normale Intensität
1B 29	ESC )	Fettschrift (reduced Intensity)
1B 2B	ESC +	Clear Screen, Attribute löschen
1B 30	ESC 0	88 Zeichen / Zeile
1B 31	ESC 1	113 Zeichen / Zeile
1B 32	ESC 2	132 Zeichen / Zeile
1B 33	ESC 3	wrap around einschalten
1B 34	ESC 4	wrap around ausschalten
1B 35	ESC 5	direkte Grafik (ohne Umrechnung)
1B 36	ESC 6	Tektronix Grafik (mit Umrechnung)
1B 38	ESC 8	US ASCII Zeichensatz
1B 39	ESC 9	deutschen Zeichensatz

ESCAPE-Sequenzen im Terminalmode (Fortsetzung)

1B 3D	y x	ESC =	y x	Direkte Cursorpositionierung. y ist die ROW, x ist die COLUMN. Offset ist 20 (hex). ESC = blank blank entspricht dem Home-Befehl.
1B 3F		ESC ?		Cursorposition lesen (y x CR)
1B 41		ESC A		hochstellen einschalten
1B 42		ESC B		tiefstellen einschalten
1B 43		ESC C		normalstellen einschalten
1B 44		ESC D		default Attribute setzen. d.h: US-ASCII, 88 Zeichen, normale Schrift, ZOOM = 0.
1B 45		ESC E		insert line
1B 47		ESC G		normal / invers toggeln
1B 48		ESC H		Testbild darstellen (alles H)
1B 49		ESC I		invers einschalten
1B 4A		ESC J		invers ausschalten
1B 4B		ESC K		schräg einschalten
1B 4C		ESC L		schräg ausschalten
1B 4D		ESC M		schräg toggeln
1B 4E		ESC N		alternativen Zeichensatz selektieren
1B 4F		ESC O		normalen Zeichensatz selektieren
1B 50		ESC P		Hardcopy zum Drucker
1B 52		ESC R		delete line
1B 54		ESC T		Erase to end of line. Von Cursor bis Zeilenende löschen.
1B 55		ESC U		unterstreichen einschalten
1B 56		ESC V		unterstreichen ausschalten
1B 57		ESC W		unterstreichen toggeln
1B 58		ESC X		Cursor blinken toggeln (blinken)
1B 59		ESC Y		Erase to end of screen. Von Cursor bis Schirmende löschen.
1B 5A	x	ESC Z	x	ZOOM-Faktor wählen, x = 0 (30 hex) bis 9 (39 hex). Der Cursor ist die linke untere Ecke des Zeichens!
1B 6F	21	ESC o	!	Reset Terminal: Clear Screen, Home, normal, US-ASCII, 88 Zeichen / Zeile, ZOOM 0.
1B 7E	30	ESC B	0	Bild ausschalten
1B 7E	31	ESC B	1	Bild ein-, Cursor ausschalten
1B 7E	32	ESC B	2	Bild ein-, Cursor einschalten
1B 78		ESC x		schnell blinkender Cursor
1B 79		ESC y		langsam blinkender Cursor

Grafik-Mode (Tektronix 4010 Emulation)

Es gibt 2 Versionen des Grafikmodes. Die erste (ESC 6) ist Tektronix-kompatibel (780 \* 1024 Punkte durch Koordinatentransformation); die zweite (ESC 5) entspricht der Tektronix-Version, enthält aber die Umrechnungen nicht, d.h. die Auflösung beträgt 269 \* 780 Punkte.

Bildschirm-Koordinaten: (Tektronix)

```

+-----+
Y  ! 0, 779                                     1023, 779 !
-  !                                           !
A  !                                           !
c  !                                           !
h  !                                           !
s  !                                           !
e  ! 0, 0                                     1023, 0 !
+-----+
                                X - Achse

```

Diese Koordinaten sind durch die Emulation des 4010 zwingend vorgeschrieben. Es findet eine automatische Umrechnung von den logischen 4010-Koordinaten in die physikalischen Bildschirm-Koordinaten statt. Im Treiberprogramm werden die 4010-Koordinaten mitgeführt, um Rundungsfehler zu minimieren.

Vektor-Mode:

Der GS-Controlcode selektiert den Vektor-Mode. Hierbei werden automatisch Vektoren aneinandergehängt. Die erste übertragene Koordinate ist der Startpunkt des ersten Vektors, die nächste Koordinate der Endpunkt. Dieser Endpunkt wird Startpunkt für den nächsten Vektor.  
Übertragungsformat für eine Koordinate:

```

High Y:    0  1  Y9 Y8 Y7 Y6 Y5
Low Y:     1  1  Y4 Y3 Y2 Y1 Y0
High X:    0  1  X9 X8 X7 X6 X5
Low X:     1  0  X4 X3 X2 X1 X0

```

Punkt-Mode:

Der FS-Controlcode selektiert den Punkt-Mode. Hier wird der Punkt, dessen Koordinaten übertragen wurden gesetzt (bzw. gelöscht). Das Übertragungsformat für die Koordinaten ist wie im Vektor-Mode.

4010 Alpha-Mode:

Der Alpha-Mode wird im 4010-Betrieb durch die Zeichen: CR, ESC, FF oder US erreicht. Der Cursor bleibt beim Umschalten an der im Vektor- oder Punkt-Mode definierten Stelle. Im 4010 Alpha-Mode werden keine Kleinbuchstaben dargestellt. Die Matrix ist 14 Punkte in X- und 22 Punkte in Y-Richtung. Es gibt kein Bildschirmscrolling. Es werden 35 Zeilen zu je 73 Zeichen dargestellt. Anstelle eines Scrolls bei Erreichen der 36-ten Zeile wird an der Koordinate 512, 779 weitergearbeitet. Der Alpha-Mode setzt den Mode immer auf Zeichnen (nicht Löschen).

4010 Control-codes:

08	CTRL H	cursor left	Cursor X = Cursor X - 14
09	CTRL I	cursor right	Cursor X = Cursor X + 14
0A	CTRL J	cursor down	Cursor Y = Cursor Y - 22
0B	CTRL K	cursor up	Cursor Y = Cursor Y + 22
0D	CTRL M	carriage return	selektiert 4010 Alpha-Mode, Cursor an den Anfang der Zeile, Zeichnen an.
18	CTRL X	cancel	nur gültig im 4010 Alpha-Mode: selektiert wieder den normalen (ADM 3-kompatiblen) Alpha-Mode.
19	CTRL Y	clear mem	löscht den Bildschirm, Zeichnen an
1C	CTRL Ö	Punkt-Mode	selektiert den Punkt-Mode
1D	CTRL Ü	Vektor-Mode	selektiert den Vektor-Mode
1E	CTRL _	Alpha-Mode	selektiert den 4010 Alpha-Mode

4010 Escape-Sequenzen:

1B 61	ESC a	zeichnen einschalten
1B 7F	ESC delete	löschen einschalten
1B 49	ESC I	invertieren einschalten (XOR)
1B 0C	ESC CTRL L	selektiert den 4010 Alpha-Mode, Bildschirm löschen, Cursor nach links oben, Zeichnen ein.
1B 5A x	ESC Z x	ZOOM-Faktor auswählen. x = A bis P für ZOOM 0 bis 15.

Alpha-Mode

Im Alpha-Mode wird der Cursor nach der Ausgabe eines Zeichens automatisch um ein Zeichen (d.h. 14 logische Punkte) weitergesetzt. Am Ende einer Zeile wird in der nächsten weitergearbeitet (22 Punkte). Bei gezoomten Zeichen steht der Cursor richtig hinter dem Zeichen, nur der Zeilenvorschub bleibt wie in der 1:1 Darstellung (mehrere LINE-FEEDs ausgeben!).

Der Cursor ist an der linken unteren Ecke des Zeichens.



\*\*\*\* TERMINAL MODE \*\*\*\*

CTRL A Markierungspunkt  
 CTRL H Backspace  
 CTRL J Linefeed  
 CTRL K Upline  
 CTRL L Forespace  
 CTRL M Carriage Return  
 CTRL Z Clear Screen  
 CTRL Ä Escape  
 CTRL Ö Point-Mode  
 CTRL Ü Vector-Mode  
 CTRL ^ Home  
 CTRL \_ New Line

\*\*\*\* GRAFIK MODE \*\*\*\*

CTRL H cursor left  
 CTRL I cursor right  
 CTRL J cursor down  
 CTRL K cursor up  
 CTRL M carriage ret  
 CTRL X cancel  
 CTRL Y clear mem  
 CTRL Ö Punkt-Mode  
 CTRL Ü Vektor-Mode  
 CTRL \_ Alpha-Mode

\*\*\*\* GRAFIK MODE \*\*\*\*

ESC a zeichnen ein  
 ESC del löschen ein  
 ESC I invertieren ein  
 ESC CTL L 4010 Alpha-Mode  
 ESC Z x ZOOM-Faktor

\*\*\* GRAFIK Koordinaten \*\*\*

High Y: 0 1 Y9 Y8 Y7 Y6 Y5  
 Low Y: 1 1 Y4 Y3 Y2 Y1 Y0  
 High X: 0 1 X9 X8 X7 X6 X5  
 Low X: 1 0 X4 X3 X2 X1 X0

\*\*\*\* TERMINAL MODE \*\*\*\*

ESC ( normale Intensität  
 ESC ) Fettschrift  
 ESC + Clear Screen  
 ESC 0 88 Zeichen  
 ESC 1 113 Zeichen  
 ESC 2 132 Zeichen  
 ESC 3 wrap around ein  
 ESC 4 wrap around aus  
 ESC 5 direkte Grafik  
 ESC 6 Tektronix Grafik  
 ESC 8 US ASCII  
 ESC 9 deutsch  
 ESC =yx Cursorpositionierung  
 ESC ? Cursorposition lesen  
 ESC A hoch ein  
 ESC B tief ein  
 ESC C normal ein  
 ESC D def. Attr.  
 ESC E insert line  
 ESC G normal/invers  
 ESC H Testbild  
 ESC I invers ein  
 ESC J invers aus  
 ESC K schräg ein  
 ESC L schräg aus  
 ESC M schräg toggeln  
 ESC N altern. Z.satz  
 ESC O norm. Z.satz  
 ESC P Hardcopy  
 ESC R delete line  
 ESC T Erase to end of line  
 ESC U underline ein  
 ESC V underline aus  
 ESC W underline an/aus  
 ESC X Cursor blinken an/aus  
 ESC Y Erase to end of screen  
 ESC Z x ZOOM-Faktor  
 ESC o ! Reset Terminal  
 ESC x schelles Blinken  
 ESC y langsames Blinken  
 ESC B 0 Bild aus  
 ESC B 1 Bild ein, Cursor aus  
 ESC B 2 Bild ein, Cursor ein

```

Y +-----+
- ! 0, 779                                     1023, 779 !
A !
c !
h !
s ! 0, 0                                     1023, 0 !
e +-----+

```

X - Achse

4.6 Move-Module

Auf den Disketten zum CBIOS 1.6 befinden sich 7 verschiedene Move-Module:

RAE3MOVE: Move-Modul für RAM/Adreß-Erweiterungskarte bis Rev. 3  
 RAE4MOVE: Move-Modul für RAM/Adreß-Erweiterungskarte ab Rev. 4  
 CPU2MOVE: Move-Modul für CPU II  
 CPU3MOVE: Move-Modul für CPU III mit DMA  
 CP3NMOVE: Move-Modul für CPU III ohne DMA  
 EPC1MOVE: Move-Modul für EPC 1.6  
 EPC2MOVE: Move-Modul für EPC II

Die 7 Move-Module schließen sich gegenseitig aus. Ebenso schließen sich die 7 verschiedenen Platinen gegenseitig aus, d.h. sie können niemals gleichzeitig in einem System vorhanden sein. Bei einer Betriebssystem-Installation mit Hilfe von CPMINST wird dies bereits bei der Abfrage der einzelnen Module berücksichtigt.

Das Move-Modul enthält folgende Entries und globale Variable:

1. Entries und Variable im Common-Memory:

**bnkini:** Initialisierung der Banking-Hardware  
 bnkini muß aufgerufen werden, bevor irgend eine andere Routine aus dem Move-Modul aufgerufen wird.

**?xmove:** Interbank-Move vorbereiten  
 (siehe CP/M 3 System Guide Seite 66)

**?move:** Memory-Move-Routine  
 (siehe CP/M 3 System Guide Seite 65)

**?bank:** Bank-Selekt-Routine  
 Diese Routine selektiert die im Register A übergebene Bank für CPU-Operationen.

**call0:** Aufruf einer Routine in Bank 0  
 Die Startadresse dieser Routine muß im Register HL übergeben werden. Falls die Bank 0 nicht aktiv ist, wird sie vor Aufruf dieser Routine selektiert. call0 sichert dann vorher den Anwenderstack und benutzt für den Aufruf dieser Routine einen eigenen Stack im Common-Memory. Nach Return von der gerufenen Routine wird die vor dem Aufruf aktive Bank wieder hergestellt. call0 zerstört keine Registerinhalte!

**cpubnk:** Augenblicklich aktive CPU-Bank  
 Diese Variable ist nicht immer identisch mit \$cbnk aus dem BIOSKERN-Modul, da gelegentlich die CPU auch in der \$dbnk arbeiten muß (z.B. wenn Diskoperationen ohne DMA stattfinden, oder der DMA nicht in die \$dbnk transferieren kann).  
 Diese Variable ist ab CBIOS 1.6 nicht mehr im Move-

Modul, sondern im BIOSKERN-Modul definiert!

?ldccp: CCP.COM von der Diskette laden und in den Hintergrund-Speicher verschieben.

## 2. Entries und Variable in der Bank 0:

- dskbnk:** Erlaubte Disk-Banks für den FDC 8/5 DMA  
Bei Verwendung einer CPU II als Move-Modul kann der FDC 8/5-DMA nur in die Bank 0 transferieren. Disk-I/O mit anderen Banks ist direkt nicht möglich. In diesen Fällen erfolgt immer eine Zwischenspeicherung in der Bank 0 (betrifft nur den FDC 8/5-DMA!).  
Bei Verwendung einer CPU III als Move-Modul kann der FDC 8/5-DMA außer in die Bank 0 noch zusätzlich in die Bank 1 transferieren.  
Bei Verwendung einer RAM/Adreß-Erweiterungsplatine gibt es für den FDC 8/5-DMA keine Einschränkungen.
- ?rlccp:** CCP.COM aus dem Hintergrund-Speicher nach Adresse 100H in der Anwenderbank laden. Dieser Transfer wird immer mit einem schnellen Interbank-Move (optimiert auf die verwendete Hardware) durchgeführt.
- ramfrm:** Formatieren der RAM-DISK  
Es wird nur das Inhaltsverzeichnis formatiert.
- ramwr:** Schreiben eines Sektors in die RAM-DISK
- ramrd:** Lesen eines Sektors aus der RAM- oder EPROM-DISK  
Beim Lesen und Schreiben von Sektoren in der RAM-DISK erfolgt eine Zeitoptimierung durch Auswertung des Multi-I/O-Flags im BIOSKERN. Bei aktivem Multi-I/O wird der zu übertragende Block aus mehreren Sektoren in einem einzigen Interbank-Move übertragen. Der verwendete Interbank-Move ist optimiert auf die verwendete Hardware.
- ramlog:** Dummy; ohne Funktion
- ramini:** Initialisierung der RAM-DISK  
ramini prüft ob im Inhaltsverzeichnis der RAM-DISK ein fehlerfreier Eintrag 'ramdisk.ok' mit SYS- und RO-Attribut vorhanden ist. Falls nein, wird die RAM-DISK formatiert und dieser Eintrag vorgenommen. Falls ja, wird die RAM-DISK nicht neu initialisiert. Dadurch ist es möglich, nach 'RESET' den Inhalt der RAM-DISK zu erhalten.
- romfrm:** Formatieren der EPROM-DISK  
Die EPROM-DISK ist 'READ ONLY'. Es erfolgt eine entsprechende Fehlermeldung.

**romwr:** Schreiben eines Sektors in die EPROM-DISK  
Die EPROM-DISK ist 'READ ONLY'. Es erfolgt eine entsprechende Fehlermeldung.

**romini:** Dummy; ohne Funktion

Der physikalische Common-Memory ist bei den einzelnen Move-Modulen verschieden:

CPU II: Der Common-Memory hat eine Größe von 16KByte.  
CPU III: Der Common-Memory hat eine Größe von 8KByte.  
RAE3: Der Common-Memory hat eine Größe von 16KByte.  
RAE4: Der Common-Memory hat eine Größe von 4KByte.  
EPC 1.6: Der Common-Memory hat eine Größe von 32KByte  
EPC II: Der Common-Memory hat eine Größe von 16KByte

Bei CPU II, EPC 1.6 und CPU III wird der gesamte im System zur Verfügung stehende Speicher benutzt, ohne daß etwas davon verloren geht.

Die Ram/Adreß-Erweiterungsplatinen müssen für den Betrieb unter CP/M PLUS auf 'Banking' gejumpert werden (siehe 'Technische Beschreibung' zur jeweiligen RAE-Platine).

Die RAE-Platine bis Rev. 1 kann nicht auf 'Banking' umgestellt werden und ist deshalb nicht unter CP/M PLUS verwendbar.

Die RAE-Platine Rev. 2 + 3 kann nur auf '48K-Banking' eingestellt werden. Bei dieser Betriebsart gehen jedoch in jeder Bank 16KByte Speicher für die Adressierung verloren. Ausgenommen ist davon lediglich die Bank 0, in der zusätzlich der Common-Memory zu finden ist. Diese Eigenschaft der RAE-Platine wird in dem zugehörigen Move-Modul berücksichtigt. Bei einer Installation des Betriebssystems mit CPMINST ist deshalb immer der tatsächlich vorhandene und nicht der tatsächlich benutzte Speicher anzugeben.

Bei der RAE-Platine ab Rev. 4 kann die Größe des Common-Memories durch Jumper eingestellt werden. Für einen Betrieb unter CP/M PLUS mit CBIOS 1.6 ist hier eine Größe von 4K einzustellen (siehe 'Technische Beschreibung' zu dieser Platine). Der eingestellte Common-Memory geht mit Ausnahme der Bank 0 für die Adressierung durch die CPU verloren, nicht jedoch für den DMA! Bei bis zu 3 Banks für CP/M PLUS (incl. USER-BANK) und den REST als RAM-DISK geht kein Speicherplatz verloren (In den oberen 4K von Bank 0 ist der Common-Memory zu finden; in den oberen 4K der Bank 1 ist der CCP untergebracht; in den ersten 4K hinter den CP/M-Banks wird die Directory der RAM-DISK positioniert). Bei mehr CP/M-Banks geht in jeder weiteren Bank ein Bereich von 4KByte verloren.

Bei der CPU III-Platine wird die Größe des Common-Memories per Software in der bnkini-Routine eingestellt.

Der EPC 1.6 kann nur maximal 512KByte adressieren! Falls bei der Installation mit CPMINST ein größerer Speicherbereich selektiert wurde, erfolgt automatisch eine Begrenzung auf 512KByte!

Der EPC II kann nur die auf der Platine befindlichen 128KByte RAM adressieren. Eine Speichererweiterung ist nicht möglich. Aus diesem Grunde können RAM-DISK, EPROM-DISK, CMOS-RAM-DISK und SMB-RAM-DISK mit dem EPC II nicht benutzt werden. Die entsprechenden Entries im Move-Modul sind dementsprechend auch nicht vorhanden.

Bitte beachten Sie, daß alle Routinen des MOVE-Modules ausschließlich innerhalb des BIOS benutzbar sind und auf keinen Fall vom Anwender aufgerufen werden können (auch nicht über die BDOS-Funktion 50!).

4.7 Datum/Zeit-Module

Der Datum/Zeit-Teil von CBIOS 1.5 besteht aus den folgenden Modulen:

- CPU3TIME.REL** Das Modul CPU3TIME.REL enthält alle hardwareabhängigen Teile zur Ansteuerung des Uhren-Moduls auf der CPU III-Platine. Dieses Modul wird nur eingebunden, wenn in dem betreffenden Computer eine CPU III-Platine vorhanden ist.
- FDC2TIME.REL** Das Modul FDC2TIME.REL enthält alle hardwareabhängigen Teile zur Ansteuerung des Uhren-Moduls auf der FDC II-Platine. Dieses Modul wird nur eingebunden, wenn in dem betreffenden Computer zwar eine FDC II-Platine, aber keine CPU III-Platine vorhanden ist.
- DATECOMP.REL** Das Modul DATECOMP.REL enthält eine allgemeine Umrechnung des CP/M-Datums in ein BCD-Datum und umgekehrt. Dieses Modul wird sowohl von FDC2TIME als auch von CPU3TIME benötigt.
- TIME.ASM** Dieses Modul enthält eine Interrupt-Uhr für die HKM-CPU, CPU I, CPU II und den EPC I. Beim EPC II wird mit diesem Modul nur eine Pseudo-Uhr realisiert. TIME.ASM wird nur dann assembliert und dazugelinkt, wenn weder eine CPU III-, noch eine FDC II-Platine im System vorhanden ist.

TIME.ASM mit der HKM-CPU:

Es wird der Kanal 3 des auf der CPU-Platine vorhandenen CTC's benutzt. Mit diesem CTC-Kanal wird ca. alle 16ms ein Interrupt ausgelöst.

TIME.ASM mit dem EPC I:

Es wird der Kanal 3 des auf dem EPC I vorhandenen CTC's benutzt. Mit diesem CTC-Kanal wird ca. alle 16ms ein Interrupt ausgelöst.

TIME.ASM mit der CPU II:

Es werden die Kanäle 1 und 3 des auf der CPU II-Platine vorhandenen CTC's benutzt. Diese beiden Kanäle sind auf der Platine hintereinander geschaltet. Durch die Kaskadierung dieser beiden Kanäle wird erreicht, daß nur zu jeder vollen Sekunde ein Interrupt ausgelöst werden muß.

TIME.ASM mit der CPU I:

Es werden die Kanäle 1 und 3 des auf der CPU I-Platine vorhandenen CTC's benutzt. Diese beiden Kanäle müssen auf

der Platine hintereinander geschaltet werden. Zu diesem Zweck ist auf der CPU I-Platine eine Verbindung von Pin 24 nach Pin 26 auf dem Stecker 1 zu jumpern. Diese beiden Anschlüsse liegen auf dem Stecker 1 direkt nebeneinander. Durch die Kaskadierung dieser beiden Kanäle wird erreicht, daß nur zu jeder vollen Sekunde ein Interrupt ausgelöst werden muß.

Für die Anpassung einer eigenen Hardware-Uhr kann das Modul DATECOMP.REL sehr nützlich sein. Es folgt deshalb hier die Beschreibung der Entries und Externalis:

Extern sind die Variablen 'year', 'month' und 'day' zur Verfügung zu stellen (bei CBIOS 1.5 geschieht dies in den Modulen CPU3TIME bzw. FDC2TIME). Diese Variablen sind in DATECOMP durch die Anweisung:

```
extrn year,month,day ; Jahr, Monat, Tag
```

vereinbart worden. Diese Variablen müssen in dem rufenden Programm mit einer Länge von einem Byte vereinbart sein. Über diese 3 Variable erfolgt der Datenaustausch zwischen den Modulen.

```
year:  packed BCD; die letzten 2 Ziffern einer Jahreszahl
month: packed BCD; Monatszahl (01..12)
day:   packed BCD; Tageszahl (01..31)
```

DATECOMP hat folgende 2 Entries:

**ymd2d:** Eingabewerte: year, month, day  
 Jahr, Monat, Tag in BCD (wie es von Uhren-Chips meist geliefert wird)  
 Ausgabe: \$date im System-Control-Block wird auf den laufenden Tag seit dem 1.1.78 gesetzt.

**d2ymd:** Eingabewert: \$date aus dem System-Control-Block  
 Ausgabe: year, month, day  
 Jahr, Monat, Tag in BCD (wie es von Uhren-Chips meist zum Setzen erwartet wird)

4.8 CBIOS-Erweiterungen

Wir benutzen in unserer Implementation des CBIOS die USER-Funktion. Sie ist zur Erweiterung wie folgt definiert:

Das Register A enthält beim Aufruf die Nummer der gewünschten Unterfunktion. Die Unterfunktionen 0 bis 31 (dezimal) sind implementiert oder für Erweiterungen durch uns reserviert. Die Unterfunktionen 32 bis 255 sind für anwenderspezifische CBIOS-Erweiterungen vorgesehen. Hierfür wird das Modul USER.ASM mitgeliefert.

Unterfunktionsübersicht:

- 0. Versionskennung zurückgeben
- 1. Translation-Table zurückgeben
- 2. BIOS-TRACE ein- oder ausschalten (falls implementiert)
- 3. RETRY ein- bzw. ausschalten
- 4. DPB zurückgeben
- 5. --- reserviert ---
- 6. Laufwerkscode zurückgeben
- 7. Formattabelle von DDATA zum Benutzer
- 8. modifizierte Formattabelle zurück zum CBIOS
- 9. erkannte Werte der letzten Diskette übergeben
- 10. Laufwerk formatieren
- 11. Floppy-Parameter (XXDPH) zum CBIOS
- 12. Floppy-Parameter (XXDPH) vom CBIOS zum Benutzer
- 13. Bildpunkte zurückgeben
- 14 bis 19 --- reserviert ---
- 20. Systemtakt zurückgeben
- 21 bis 31 --- reserviert ---

Diese Funktionen werden von den von uns erstellten System-Utilities benutzt. Sie sollten vom Anwender nicht verwendet werden. Für den Anwender erlaubt sind die Unterfunktionen 0, 2 und alle Funktionen oberhalb von 31 (selbst schreiben!).

Unterfunktion 0

Funktion:	Versionskennung übergeben
Eingabe:	DE zeigt auf ein Datenfeld mit 6Byte Länge A enthält die 0 (Funktionscode)
Rückgabe:	Das Datenfeld ist gefüllt mit:
	word: OEM-code
	word: CBIOS-Versionskennung
	word: Anwender-Kennung (eintragen in USER)



Unterfunktion 1

Funktion: Translationtable übergeben  
 Eingabe: DE enthält die Laufwerksnummer (0 bis 15)  
 HL zeigt auf ein Datenfeld mit 52 Byte Länge  
 A enthält die 1 (Funktionscode)  
 Rückgabe: A enthält die Sektorgröße (PSH)  
 Das Datenfeld enthält die XLT des Laufwerks  
 (falls es für dieses Laufwerk eine gibt!)

Unterfunktion 2

Funktion: alle Aufrufe an das CBIOS protokollieren, Voraussetzung hierfür ist, daß das Modul TRACE ins CBIOS eingelinkt wurde. Die TRACE-Identifikation wird beim Hochfahren des Systems gemeldet, falls TRACE im CBIOS enthalten ist.  
 Eingabe: C = 0 TRACE ausschalten  
 C = 1 TRACE einschalten (Ausgabe: CON)  
 C = 2 TRACE einschalten (Ausgabe: AUX)  
 C = 3 TRACE einschalten (Ausgabe: LST)  
 A enthält eine 2 (Funktionscode)  
 Rückgabe: --- keine ---

Unterfunktion 3

Funktion: RETRY ein- bzw. ausschalten (Testlesen von Laufwerken)  
 Eingabe: C enthält eine  $\uparrow$  zum Ausschalten bzw. eine  $\times \phi$  zum Einschalten des RETRY  
 A enthält die 3 (Funktionscode)  
 Rückgabe: --- keine ---

Unterfunktion 4

Funktion: DPB-Werte übergeben  
 Eingabe: DE enthält die Laufwerksnummer  
 HL zeigt auf ein Datenfeld mit 17Byte Länge  
 A enthält die 4 (Funktionscode)  
 Rückgabe: Das Datenfeld enthält die DPB des Laufwerks

Unterfunktion 6

Funktion: Laufwerkstyp übergeben  
 Eingabe: DE enthält die Laufwerksnummer (0 bis 15)  
 A enthält die 6 (Funktionscode)  
 Rückgabe: A enthält den Laufwerkstyp aus der erweiterten \$DTBL (siehe: Kapitel 4.4 DISKS)

Unterfunktion 7

Funktion: Tabelle der aktiven Fremdformate zum Benutzer  
 Eingabe: DE zeigt auf ein Datenfeld mit entsprechender Länge (i.a. 2EC hex lang). Diese Länge ist in DDATA definiert und abhängig von der Anzahl der gleichzeitig aktiven Fremdformate.  
 Rückgabe: A enthält die 7 (Funktionscode)  
 Das Datenfeld enthält die Tabelle der Fremdformate

Unterfunktion 8

Funktion: Tabelle der aktiven Fremdformate zum System  
 Eingabe: DE zeigt auf die vom Benutzer erstellte Tabelle  
 Rückgabe: A enthält die 8 (Funktionscode)  
 --- keine ---

Unterfunktion 9

Funktion: erkannte Werte des letzten LOGINS übergeben  
 Eingabe: DE zeigt auf ein Datenfeld mit 15Byte Länge  
 Rückgabe: A enthält die 9 (Funktionscode)  
 Das Datenfeld enthält die Vergleichswerte:  
 - Diskgröße  
 - density  
 - Sektorgröße  
 - Nummer des vorderseitigen Kopfes  
 - Nummer der Spur 0  
 - Seiten  
 - Spuren pro Seite  
 - Sektoren pro Spur  
 - SKEW vom Formatieren  
 - FLAG  
 - normal / intern  
 - Nummer der Spur 8  
 - Nummer des rückseitigen Kopfes  
 - reserviert  
 - reserviert

Unterfunktion 10

Funktion: Laufwerk formatieren  
 Eingabe: DE enthält die Laufwerksnummer (0 bis 15)  
 HL enthält die Anfangsspur  
 Rückgabe: A enthält die 10 (Funktionscode)  
 A enthält einen Fehlercode  
 A = 0h: richtig formatiert  
 A = 1h: Fehler (BAD SECTOR)  
 A = 2h: schreibgeschützt  
 A = 10h: kein Laufwerk  
 A = FFh: falsches Format

Unterfunktion 11

Funktion: XXDPH-Werte ans CBIOS übergeben (LOGIN emulieren)  
 Eingabe: DE enthält die Laufwerksnummer  
 HL zeigt auf die bereitgestellten Werte; 13 Byte von XXDPH-Laufwerksgröße bis XXDPH-intern)  
 A enthält die 11 (Funktionscode)  
 Rückgabe: --- keine ---

Unterfunktion 12

Funktion: XXDPH-Werte vom CBIOS holen  
 Eingabe: DE enthält die Laufwerksnummer (0 bis 15)  
 HL zeigt auf ein Datenfeld von 13Byte Länge  
 A enthält die 12 (Funktionscode)  
 Rückgabe: Das Datenfeld enthält die XXDPH-Werte von Laufwerksgröße bis intern  
 H enthält die Anzahl der Spuren des Laufwerks  
 A enthält die Anzahl der Seiten des Laufwerks

Unterfunktion 13

Funktion: EPC II: Bildpunkte an Benutzer übergeben.  
 Format: 800Byte mit den unteren 4Bit. Übergeben werden vier aufeinanderfolgende horizontale Zeilen. (800 Punkte Zeilenlänge)  
 Eingabe: BC enthält eine Zeilennummer; 0 für die ersten 4 Zeilen, 1 für die nächsten 4 usw. bis 67 für die letzten 4 Zeilen  
 DE zeigt auf ein Datenfeld mit 800Byte Länge  
 A enthält die 13 (Funktionscode)  
 Rückgabe: Das Datenfeld enthält die Bildzeile

Unterfunktion 20

Funktion: Systemtakt an Benutzer übergeben  
 Eingabe: A enthält die 20 (Funktionscode)  
 Rückgabe: A = 4: 4MHz Systemtakt  
 A = 5: 5MHz Systemtakt  
 A = 6: 6MHz Systemtakt  
 A = 8: 8MHz Systemtakt  
 A = 0: falscher Sytemtakt

**Achtung:** Die Benutzung der USER-Funktionen in Anwenderprogrammen führt zur Inkompatibilität gegenüber beliebigen anderen CP/M PLUS-Implementationen. Das Verhalten anderer Implementationen kann nicht vorhergesagt werden. In schlimmsten Fällen ist mit Systemabstürzen und Datenverlusten zu rechnen!

Unterfunktion 27 Funktionen: Absolute CPU-Bank an Benutzer

Rückgabe: HL = CPU Bank  
 A = ...

Die Datei USER.ASM

USER.ASM wird beim Generieren eines Betriebssystems assembliert und ins CBIOS eingefügt. Zur Erweiterung des CBIOS sind Routinen in USER.ASM zu schreiben. Es gibt folgende ENTRYPOINTS:

USERF wird aufgerufen bei Unterfunktionsnummern von 32 bis 255. In der Auslieferungsversion steht hier nur ein RET-Befehl.

UVERS wird beim Aufruf der Versionskennung gerufen. Hier kann, falls gewünscht, eine anwenderspezifische Kennung in das Register HL geschrieben werden.

USERID ist die Meldung, die beim Hochfahren des Systems vom BOOT-Modul an die Konsole geschickt wird. USERID muß wie folgt gestaltet sein:

```
userid:    db      'ID',0dh,0ah,'irgendein Text',0
```

Nur der unterstrichene Teil darf modifiziert werden. Er darf kein NUL-Zeichen (hex-code: 00h) enthalten. Das NUL-Zeichen wird zur Endeerkennung benutzt. TABs funktionieren im CBIOS nicht. Sie werden ohne Expansion einfach gesendet.

Die Datei PRINTER.ASM

PRINTER.ASM wird beim Generieren eines Betriebssystems assembliert und ins CBIOS eingefügt. PRINTER.ASM definiert einige Strings zur Initialisierung eines grafikfähigen Druckers. Es gibt folgende ENTRYPOINTS:

**DOTRET** implementiert die Unterfunktion 13 (nur EPC II)

---- die folgenden Strings werden nur für den EPC II benötigt ----

**PRNTID** ist die Meldung, die beim Hochfahren des Systems vom BOOT-Modul an die Konsole geschickt wird (nur wenn grafikfähiger Drucker implementiert ist). Der Aufbau muß analog zu USERID erfolgen.

**GRINIT** ist der Initialisierungsstring, um auf dem angeschlossenen Drucker 800 Punkte pro Zeile ausgeben zu können

**GREXIT** initialisiert den Drucker wieder für den normalen Betrieb im Sytem.

**GRLEAD** ist die Zeichenfolge, nach der der Drucker 800 Dotinformationen akzeptiert.

**GRCRLF** veranlaßt den Drucker, ein CR und ein LF für 4 Punkte durchzuführen.

**GRDOTS** ist ein FLAG, das die Zuordnung zwischen den Nadeln des Druckers und der Bitwertigkeit herstellt. Bei 0 steht das msb, bei 1 das lsb für die oberste Nadel.

### Das Modul: TRACE.REL

TRACE stellt ein Hilfsmittel dar, mit dem CBIOS-Aufrufe protokolliert werden können. Das Ausgabegerät ist hierbei einstellbar. Ein Protokoll sieht wie folgt aus:

```
dsk 0002 EF67
home
trk 0045 xlt 0000 5467 0001 sec 0001 dma 3078 rd 00
trk 0045 xlt 0001 5467 0003 sec 0003 dma 30F8 rd 00
trk 0045 xlt 0001 5467 0003 sec 0003 dma 30F8 wr 00
```

Alle Ausgaben erfolgen in hexadezimaler Schreibweise. Folgende Aufrufe werden protokolliert:

SELECT DISK	dsk, Laufwerksnummer, DPH-Adresse
HOME	home
SELECT TRACK	trk, Spurnummer
SELECT SECTOR	sec, Sektornummer
SET DMA	dma, DMA-Adresse
READ	rd, Fehlercode
WRITE	wr, Fehlercode
SECTTRANSLATION	xlt, log. Sektor, XLT-Adresse, phys. Sektor
SET MULTI-I/O	mio, Sektorzähler

TRACE ist gedacht für den erfahrenen Programmierer mit guten Kenntnissen über die internen Strukturen und Arbeitsweisen im CP/M PLUS. TRACE hilft festzustellen, warum ein Programm nicht unter CP/M läuft. (Alle direkten BIOS Disk-Aufrufe sind unter CP/M PLUS verboten, die BIOS-Schnittstelle ist gegenüber CP/M 2.2 undefiniert worden.)

Um TRACE zu selektieren ist die LINK-Anweisung in **CPMGEN.DAT** zu modifizieren! TRACE muß vor BIOSKERN eingebunden werden!