

oettle & reichler  
datentechnik

Schießgrabenstr. 28 a  
8900 Augsburg 1

Tel.: (0821) 15 46 32

---

C B I O S - 3

Customer Bios für CP/M Plus

H A N D B U C H

Copyright (C) by DATENTECHNIK OETTLE + REICHLER, Augsburg  
September 1984

## Inhaltsverzeichnis

<b>Die CP/M-Plus Philosophie .....</b>	<b>4</b>
Dateiverwaltung .....	4
Directory Hashing und Caching .....	5
LRU Record Pufferung .....	5
Multi-Sektor-I/O .....	6
Datums- und Zeitmarkierung .....	6
Password-Schutz .....	6
Automatisches Disk-Login .....	6
Banked CP/M-Plus .....	7
<b>Speicherverwaltung unter CBIOS-3 .....</b>	<b>8</b>
Common Memory .....	8
System-Bank 0 .....	9
TPA-Bank 1 .....	9
Ram-Floppy-Bank 2 - 15 .....	9
<b>Die CBIOS-3 Funktionen .....</b>	<b>11</b>
System Initialisierung .....	11
Zeichen Ein/Ausgabe .....	12
Disk Ein/Ausgabe .....	12
Speicher Selektion/Transfer .....	14
Uhrzeit- und Datumsunterstützung .....	15
Residente Biosfunktionen .....	15
Interrupts .....	15
<b>I/O-Geräte .....</b>	<b>16</b>
Parallele Schnittstellen .....	16
Serielle Schnittstellen .....	16
RGB-Terminal .....	17
I/O-Geräte Voreinstellung .....	17
Baudraten .....	18
<b>Disk-Ein/Ausgabe .....</b>	<b>19</b>
Disketten Wechsel .....	19
Disk Select .....	20
Disk Login .....	20
Fehlermeldungen bei Disk I/O .....	20
<b>Standard Disk-Formate .....</b>	<b>23</b>
Mini Formate 80 Tracks .....	23
Mini Formate 8 Zoll kompatibel .....	23

8 Zoll Formate .....	23
Mini Formate 40 Tracks .....	24
Directoryeinträge und Blockgrößen .....	24
Übersicht Standard-Formate .....	25
<b>Ram-Floppy Erweiterung .....</b>	<b>26</b>
<b>Systemstruktur .....</b>	<b>27</b>
Der CPM3.SYS-File .....	27
Der CCP.COM-File .....	27
Der Cold-Boot-Ladevorgang .....	28
Der Boot-Eprom .....	28
Der CP/M-Loader .....	29
<b>Der BNKBIOS3.SPR-File .....</b>	<b>30</b>
Erstellen des BNKBIOS3-Files .....	30
Das Binden der Bios-Module .....	31
Gebanktes und resistentes BIOS .....	32
<b>Systemgenerierung .....</b>	<b>33</b>
So arbeitet GENCPM .....	33
Generierung eines neuen CPM-Loaders .....	38
Struktur der Bios-Module .....	39
<b>Der AUTODPB0-File .....</b>	<b>40</b>
Disk-Timing .....	40
Das Type-Flag .....	41
Die Disk-Parameter-Header .....	41
Die Drive-Tabelle .....	42
Die Disk-Parameter-Blöcke .....	42
Format-Erkennung .....	43
Automatische Type-Byte Modifikation .....	44
<b>Der USRROUT-File .....</b>	<b>46</b>

---

Kein Teil dieser Veröffentlichung darf reproduziert, vervielfältigt, gespeichert oder übersetzt werden, ohne die ausdrückliche schriftliche Zustimmung von DATENTECHNIK oettle & reichler. Wir behalten uns das Recht vor, Änderungen, die einer Verbesserung einer Schaltung oder unserer Produkte dienen, ohne besondere Hinweise vorzunehmen. Für die Richtigkeit der hier gegebenen Daten, Schaltpläne, Programme und Beschreibungen wird keine Haftung übernommen.

### **Die CP/M-Plus Philosophie:**

Durch die neuartige Banking-Technik wird das Problem des begrenzten Adressumfangs eines 8-Bit-Prozessor auf sehr elegante Weise gelöst. CP/M Plus ist in der Lage, komplexe Speicherbanks zu adressieren und zu verwalten, so daß auch ein Z-80 System bis zu 1 MByte RAM-Speicher verwalten kann. Außerdem hat CP/M Plus eine Zeit- und Datumsmarkierung, Fehlersuche und Fehlerkorrektur, automatisches Disk-Login, 'hashed' Directory-Zugriff, Record Pufferung und Multi-Sektor Ein-Ausgabe. CP/M Plus kann bis zu 16 einzelne RAM-Arbeitsspeicherbereiche (banks) und bis zu 16 DiskettenPlattenspeichereinheiten von jeweils 512 MByte Kapazität verwalten. Als logische Ein/Ausgabe-Einheiten werden bis zu 16 verschiedene Devices verwaltet.

### **1 MByte Systemspeicher durch banking**

CP/M Plus schlägt eine völlig neue Technik der Speicherverwaltung von der Betriebssystemebene aus ein, dem sogenannten 'banking'. Der Adressraum des Prozessors wird durch entsprechende Hardware-Unterstützung in bis zu 16 verschiedene 64 k Banks unterteilt. Das BDOS schaltet je nach Bedarf die richtige Bank in den Adressraum des Prozessors. Wichtig dabei ist, daß ein Teil des Adressraums von jeder Bank aus erreichbar ist. Dieser sog. Common Memory, der in jede Speicherbank eingeblendet ist, belegt je nach Hardware die obersten 4 bis 16 kByte jeder Bank. Von diesem Common-Bereich aus geschieht die Adressumschaltung der einzelnen Banks und die Betriebssystem-Aufrufe.

### **Leistungsfähige Dateiverwaltung**

Zahlreiche Disk-Zugriffe setzen die Geschwindigkeit von Microcomputer-Systemen stark herab, da mechanische Laufwerke leider sehr langsam arbeiten. Um einen File zu transferieren müssen zeitraubende Seek-Operationen, Kopf-Lade-Zeiten und nicht zu unterschätzende Suchoperationen nach dem richtigen File durchgeführt werden. Moderne Laufwerke weisen zwar immer kürzere Zugriffszeiten auf, im Vergleich zu Prozessor-Operationen dauert ein Disk-Transfer immer noch sehr lang. Ein Ausweg aus dieser Situation wäre eine Geschwindigkeitssteigerung der mechanischen Laufwerke, wie es z.B. bei Festplatten-Laufwerken mit Übertragungsraten von 10 MBit/sec der Fall ist. Leider ist diese Lösung auch nur unbefriedigend und mit hohen Kosten verbunden. Eine zweite viel elegantere Lösung zeigt das neue Betriebssystem:

Files werden im RAM-Bereich zwischengespeichert - Diskoperationen auf ein absolutes Minimum reduziert und erheblich beschleunigt. Unter CP/M Plus werden einige völlig neue Techniken der Dateiverwaltung eingeschlagen:

### **Directory Hashing und Caching**

Hash-Tabellen verkürzen die Suche nach bestimmten Directory-Einträgen (das Inhaltsverzeichnis einer Diskette) erheblich. Mit Hilfe der Hash-Tabelle kann das BDOS die Stelle (Sektor, Track) des gewünschten Directory-Eintrags durch Schaffung eines numerischen Index sofort bestimmen. Dadurch entfällt das Durchsuchen des oft mehrere kByte großen Directory's nach bestimmten Einträgen. Je nach verfügbarem Systemspeicher werden große Teile des Directorys eines jeden Laufwerks in den Systemspeicher geladen (Directory Caching), von wo aus sie sehr schnell verarbeitet werden können. Hash-Tabellen und Directory-Puffer vermeiden daher viele der Directory-Zugriffe, besonders wenn auf große Files zugegriffen wird. Langwierige Seek-Operationen des Laufwerks zwischen den Daten- und Directory-Spuren der Diskette entfallen.

### **LRU - Record Pufferung**

Neben den Directory-Puffern unterhält das BDOS getrennt für jedes Laufwerk eine bestimmbare Anzahl von Daten-Puffer zur Zwischenspeicherung von Files und zum Blocking/Deblocking. Soll ein File bearbeitet werden, so findet dies ohne jeden Disk-Zugriff in den sog. Record Puffern statt. Der File wird dort manipuliert und erst nach Beenden der File-Operationen (Close File, Disk Reset, Warm-Boot) wieder auf die Diskette zurückgeschrieben. Wird von einem Anwenderprogramm auf einen File zugegriffen, so untersucht das BDOS zunächst, ob sich der gewünschte Record nicht schon in einem der Daten-Puffer befindet um einen Disk-Zugriff zu vermeiden. Reichen die verfügbaren Zwischenpuffer zur Bearbeitung nicht mehr aus, so sichert das BDOS denjenigen Puffer auf Diskette, der die längste Zeit nicht mehr beschrieben wurde. Damit wird Platz für öfter angesprochene Teile des File frei (LRU Least Recently Used Record Pufferung).

### **Multi-Sektor Ein-Ausgabe**

Oft kommt es vor, daß große Teile eines Files nicht verstreut auf der Diskette liegen, sondern sequentiell aufgezeichnet sind, also so wie sie später in den RAM geladen werden. Das BDOS erkennt dies und teilt dem BIOS mit, daß in einem Laufwerkszugriff und in einer Umdrehung der Diskette mehrere Sektoren sequentiell gelesen oder geschrieben werden sollen. In einem einzigen Zugriff lassen sich somit bis zu 16 kByte lange Programmteile sehr schnell transferieren. Unter CP/M 2.2 sind dazu immerhin 128 einzelne Diskzugriffe nötig. Dabei versucht das BDOS beim Schreiben von Files nicht wie unter CP/M 2.2 die 'Lücken' einer Diskette von vorne her aufzufüllen, sondern möglichst große zusammenhängende Teile des Files zu schreiben, um die Multi-Sektor-Fähigkeit des BIOS voll zu unterstützen.

### **Datums- und Zeitmarkierung von Dateien**

Im Directory einer Diskette können jeder Datei zwei Datums- und Zeitmarkierungen zugeordnet werden. Der Zeitpunkt der Erstellung, der letzten Modifikation und des letzten Zugriffs auf eine Datei kann mit Hilfe des Dienstprogrammes DIR wieder angezeigt werden.

### **Password Protection**

Neben der Datums- und Zeitmarkierung von Dateien besteht unter CP/M Plus die Möglichkeit des wirkungsvollen Schutzes einzelner Files oder einer ganzen Diskette vor fremden Zugriff mit Hilfe von 'Passwords'. Jedem File kann ein bestimmtes Password zugeordnet werden, ohne dessen Eingabe bestimmte Zugriffsrechte auf den File verwehrt bleiben. Der File kann vor unbefugtem Lesen, Schreiben oder Löschen geschützt werden. Die selben Schutzmöglichkeiten sind auch für eine gesamte Diskette möglich.

### **Automatisches Disk-Login**

Das automatische Disk-Login macht es überflüssig jedesmal mit CTRL-C zu 'reseten' wenn man die Disk wechselt. Das BDOS erkennt den Disk-Wechsel automatisch. Nach einem Disk-Login wird die Diskette automatisch in das System eingebunden.

### User 0 directory

Eine weitere Verbesserung ist, daß häufig benutzte Programme unter der 'User 0 directory' abgelegt werden. Dadurch kann von jedem beliebigen der insgesamt 16 User Bereiche auf dieses Programm zugegriffen werden. Befindet sich dieses Programm in Laufwerk A, User 0 directory, so ist sogar ein Zugriff von jedem anderen Laufwerk aus unter einer beliebigen User Nummer möglich.

### Ausführliche Fehlermeldung

Treten Fehler im System auf, so gibt das CP/M Plus System ausführliche Meldungen über Art und Behebbarkeit der Fehler aus. Das Bild zeigt ein Beispiel für einen Schreibversuch auf eine schreibgeschützte Diskette. Das BIOS erkennt den Fehler, fragt den Benutzer ob ein neuer Schreibversuch gestartet werden soll. Wird dies verneint so gibt das BIOS den Fehler an das die Funktion aufrufende BDOS weiter, das eine weitere Fehlermeldung an den Benutzer abgibt.

#### BIOS Error on D:

```
Track - 00126, Sector - 00015, Write  
Not Writable !  
Retry <Y> ? N
```

#### CP/M Error on D: Disk I/O

```
BDOS Funktion = 19 File = TEST .COM
```

#### ERROR: Bad close

### Banked CP/M Plus

In der gebankten Version gliedert sich das CP/M-Betriebssystem in zwei Teile: Dem residenten Teil und dem gebankten Teil. Der residente Teil befindet sich im Common Memory und kann von jeder Bank aus angesprochen werden. Er dient zum Aufruf des Betriebssystems, zum Umschalten zwischen den verschiedenen Banks und zur Abwicklung einfacher Operationen wie Console In, Console Out ... Der gebankte Teil des CP/M Plus Systems ist unterhalb des Common Memorys in Bank 0 eingelagert. Für Anwenderprogramme steht die gesamte Bank 1 und der noch freie Teil des Common Memory zur Verfügung.

**Das CBIOS-3:**

Das CBIOS-3 stellt die hardwareabhängige Schnittstelle zwischen CP/M-System und dem System 8000 dar. Das Bios ist in Z-80 Code geschrieben und belegt ca. 10 kByte Speicherplatz. Das CBIOS-3 verwaltet zwei RS232 / V24 Schnittstellen mit XON/XOFF und ETX/ACK Protokoll, eine serielle Centronics Schnittstelle, einen bidirektionalen seriellen Tastaturkanal, das RGB-Graphik-Terminal und eine parallele Ein/-Ausgabe Schnittstelle. Es können bis zu vier gemischte Laufwerke angeschlossen werden. Die Auto-Disk-Select Funktion erkennt standardmäßig pro Laufwerk 40 vordefinierte Diskformate. Weitere Formate können durch ändern des AUTODPBO-Files erkannt werden. Die vollständige Implementierung der Funktionen XMOVE (Interbank-Move) und MULTIO (Lesen mehrerer Sektoren in einem Laufwerkszugriff) und ein spezieller Bios-Track-Buffer zum Transferieren eines gesamten Disktracks in einem Zugriff tragen erheblich zur hohen Leistungsfähigkeit des Systems bei. Eine ausführliche Fehlermeldung beim Disk-Read, -Write und -Login hilft dem Anwender bei der Fehlersuche und ihrer Behebung.

**Die Speicheraufteilung unter CBIOS-3**

Ein gebanktes CP/M Plus System benötigt mindestens 128 kByte RAM-Speicher, also zwei 64 kByte Bank's. Im folgendem wird die Speicheraufteilung eines Systems 8023 unter dem EPC-Computer dargestellt ( Die Werte in Klammern gelten für das System 8043 unter der Rechnerkarte CPU):

**Common Memory:**

Der Common-Memory-Bereich beträgt 16 (4) kByte. D.h. der Bereich von 0C000h (0F000h) bis 0FFFFh ist von jeder Bank aus erreichbar. Der residente Teil von CP/M Plus beträgt 4 kByte und erstreckt sich somit von 0F000h bis 0FFFFh. Das residente BDOS startet bei 0F000h und ist 1.5 kByte lang. Bei 0F600h startet das residente BIOS-Teil mit einer Länge von 2.5 kByte. Der Rest des Common Memorys von 0C000h bis 0EFFFh (0) zählt zur TPA und ist somit frei für Anwenderprogramme.



**System-Bank 0:**

Der gebankte Teil des CP/M Systems liegt in Bank 0 und startet mit dem BDOS bei 07900h (0AA00h). Das gebankte BDOS belegt 11 k Speicherplatz, somit startet der gebankte Teil des BIOS bei 0A700h (0D800h) mit einer Länge von 6.25 kByte bis 0BFFFh (0EFFFh), dem Ende der Bank 0. Der Rest der Bank 0 von ca. 31 kByte (43 k)Länge wird zum Anlegen der Cache-Buffer für Daten-und Directorys, Hash-Tabellen, dem Bios-Trackpuffer und zum Zwischenspeichern des CCP's (Console Command Prozessor) verwendet.

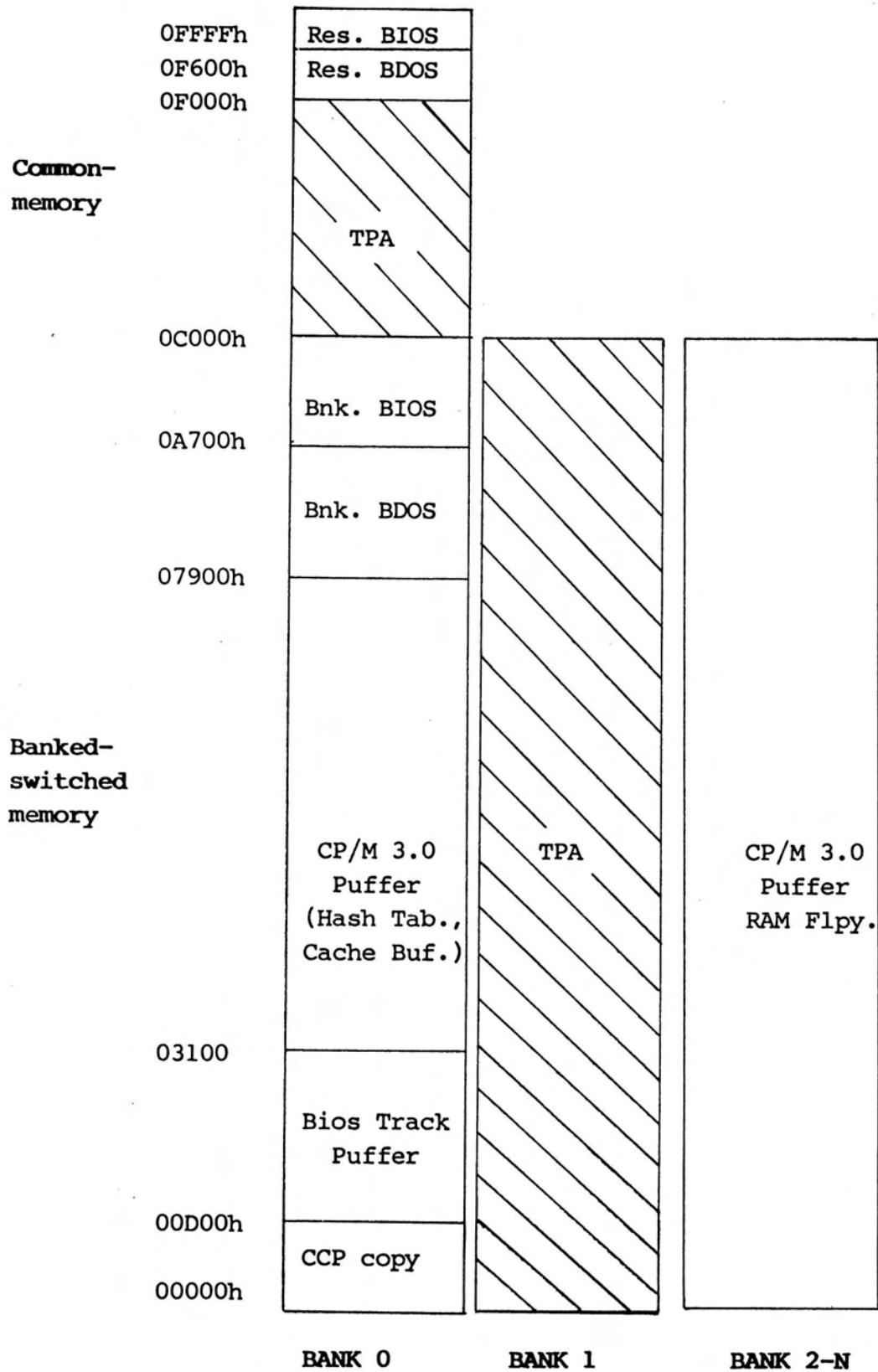
**TPA-Bank 1:**

Die gesammte Bank 1 mit 48 kByte (60 kByte) von 00000h bis 0BFFFh (0EFFFh) ist frei für Anwenderprogramme. Unter dem System 8023 zählt der noch verbleibende Teil des Common Memorys mit 12 kByte zur TPA. Somit steht dem Benutzer ein TPA-Bereich von 60 kByte zur Verfügung trotz einer Betriebssystemlänge von 21 kByte, die zahlreichen Puffer noch nicht mitgerechnet !

**RAM-Floppy Bank 2 - 15:**

Die restlichen Banks 2-15 werden zum Betrieb des Systems nicht unbedingt benötigt. Dieser Bereich kann nach Speicherausbau des Systems zum Anlegen weitere Puffer zugeteilt werden, oder er dient als Speicher einer sehr schnell arbeitenden RAM-Floppy. Dazu wird RAM-Speicher wie ein normales physikalisches Laufwerk angesprochen, ohne zeitraubende Seek-, Transfer- und Kopfladezeiten herkömmlicher Laufwerke abwarten zu müssen.

Diagramm der Speicheraufteilung System 8023:



**Die CBIOS-3-Funktionen:**

Das BIOS des CP/M-Plus Systems besteht wie unter CP/M 2.2 aus einer Sprungleiste am Anfang des BIOS die zu den einzelnen Funktionen führt. Im Unterschied zu CP/M 2.2 müssen jetzt 33 Funktionen bedient werden, deren ersten 17 Funktionen sich weitgehend mit denen von CP/M 2.2 bekannten Funktionen decken. Die BIOS-Funktionen lassen sich grob in fünf Gruppen einteilen, die an Hand der EPC-BIOS-Implementation (8023) besprochen werden:

**System-Initialisierung:**

- 0 BOOT: Boot initialisiert alle Hardware-Komponenten des Systems neu (STI, DART, Baudraten einstellen ...) und gibt eine Bereitschaftsanzeige aus, die Daten wie verfügbarer Speicherplatz, Uhrzeit und Datum enthält. Danach wird eine Kopie des CCP's in Bank 0 angelegt, indem der File CCP.COM von Diskette gelesen wird. Nun wird die Kontrolle an die Warm-Boot-Funktion weitergeleitet.
- 1 WBOOT: Warm-Boot wird immer nach Verlassen eines Anwenderprogramms aufgerufen. Diese Funktion selektiert die TPA-Bank 1, setzt wie unter CP/M 2.2 zwei Sprünge zum BIOS (0000h) und BDOS (0005h) in Bank 1 und lädt schließlich die Kopie des CCP's in Bank 1 ab 100h. Danach erfolgt ein Sprung nach 100h auf den CCP.
- 20 DEVTBL: Sinn der Funktion 'DEVTBL' ist es, dem BDOS die Adresse der sog. Charakter-Tabelle zu übergeben. Diese Tabelle enthält alle Informationen der angeschlossenen I/O-Einheiten wie Baud-Rate, Protokoll, Name usw.
- 21 DEVINI: DEVINI initialisiert eine bestimmte I/O-Einheiten neu. Das aufrufende Programm setzt in der zugehörigen Charakter-Tabelle bestimmte Parameter neu, die DEVINI entsprechend verarbeitet. So definiert z.B. das BDOS die Baud-Rate für eine serielle Schnittstelle neu.
- 22 DRVTBL: DRVTBL ('Drive-Table') übergibt ähnlich wie DEVTBL die Adresse einer Tabelle die Informationen über die Anzahl und Art der angeschlossenen Laufwerke beinhaltet. Der erste Tabelleneintrag gehört zu Laufwerk A, der zweite zu Laufwerk B usw. Die Einträge zeigen auf den zugehörigen Disk-Parameter-Header (DPH) des Laufwerks.

**Zeichen Ein/Ausgabe:**

CP/M Plus 'kennt' drei logische Ein/Ausgabeeinheiten: Die Konsole (3: CONIN, 4: CONOUT, 2: CONST, 17: CONOST), den Drucker (5: LIST, 15: LISTST) und einen Hilfskanal (7: AUXIN, 6: AUXOUT, 18: AUXIST, 19: AUXOST), der die Reader und Punch Einheiten unter CP/M 2.2 ersetzt. Zu jeder der fünf logischen Ein/Ausgabeeinheit (CON-IN/OUT, AUX-IN/OUT, LIST) existiert ein 16 Bit breiter Umleitungs-Vektor (I/O Redirection Vektor), der das I/O-Byte unter CP/M 2.2 ablöst. Dieser Vektor bildet sozusagen die fünf logischen Ein/Ausgabeeinheiten auf maximal 16 physikalische Ein/Ausgabe-Geräte ab. Jedem der 16 Bits ist eine physikalische I/O-Einheit zugeordnet. Jedes gesetzte Bit des Vektors besagt, daß Zeichen von und zu der zugehörige Einheit transferiert werden sollen. Es auch mehrere Bits gesetzt werden, um gleichzeitig mehrere I/O-Einheiten anzusprechen. Sind z.B. meherer Konsolen an das System angeschlossen, so kann durch Setzen der entsprechenden Bits im Vektor die Ein- und Ausgabe von einer beliebigen Konsole aus erfolgen und Zeichen werden automatisch auf alle Einheiten ausgegeben. Dabei ist es gleichgültig ob die Ausgabeeinheit z.B. ein Drucker, ein Bildschirm oder ein Kassettenlaufwerk ist. Da für Ein- und Ausgabe getrennte Umleitungs-Vektoren existieren, kann z.B. die Zeicheneingabe von einer ganz anderen Einheit aus erfolgen als für die Ausgabe. In Verbindung mit der DEVINI-Funktion, mit der sich sogar Protokolle und Baudraten im Betrieb ändern lassen, bedeuten diese neuen BIOS-Funktion eine gewaltige Flexibilität-Steigerung gegenüber dem alten CP/M 2.2 System, deren Wert jeder erfahrene Anwender zu schätzen weiß.

**Disk Ein/Ausgabe:**

Die Disk Ein/Ausgabe unterscheidet sich bei einfachen Implementationen nur unwesentlich von den bekannten Funktionen des 2.2 - Systems, da es nicht unbedingt notwendig ist spezielle Funktionen wie Multi-I/O, Flush usw. zu verwirklichen. Doch gerade sie steigern die Leistung des neuen Betriebssystems erheblich. Daher wurden sie im CBIOS-3 sehr sorgfältig verwirklicht. Doch dazu kommt noch eine komplizierte Auto-Disk-Select Funktion, die 40 verschiedene Diskformate automatisch erkennt. Es besteht auch die Möglichkeit spezielle Disk-Formate mit all den zugehörigen Parametern selbst zu definieren und in das CBIOS-3 einzubinden. Ein zusätzliche Geschwindigkeitssteigerung bringt neben dem Deblocking-Algorithmus, der es erlaubt beliebig lange Sektorgrößen von 128 Byte aufwärts zu verarbeiten, der sog. BIOS-

Track-Puffer, der einen ganzen Track einer Diskette zwischenspeichern kann, um beim Schreiben oder Lesen die Daten in einem Durchgang sehr schnell transferieren zu können.

8 HOME: Die Home Funktion ist identisch mit der Anwahl von Track 0 über die SETTRK-Funktion.

9 SELDSK: Wie unter dem 2.2-System dient SELDSK der Anwahl des gewünschten Laufwerks. Das aufrufende BDOS gibt hier jedoch noch einen Hinweis ob es sich um einen Erst-Aufruf des Laufwerks handelt, d.h. ob das Format der eingelegten Diskette ermittelt werden soll. Ist dies der Fall, so wird die LOGIN-Prozedur aufgerufen die das Format bestimmt.

10 SETTRK: 'Set-Track' übernimmt einfach die gewünschte Spur und legt den Wert ab.

11 SETSEC: Ähnlich wie SETTRK übernimmt 'Set-Sector' den gewünschten Sektor und speichert ihn ab.

12 SETDMA: 'Set-DMA' speichert nur die DMA-Adresse ab.

13 READ: Hier beginnt das BIOS den gewünschten Sektor zu übertragen. Zuerst wird geprüft, ob der Sektor sich nicht schon im Track-Buffer befindet. Ist dies der Fall, so findet eine Übertragung ohne jeden Disk-Zugriff statt. Andernfalls wird geprüft, ob sich der Laufwerks-Kopf nicht schon auf der gewünschten Spur befindet, um unnötige zeitraubende Seek-Operationen zu vermeiden.

14 WRITE: Die Write-Funktion durchläuft die gleichen Routinen wie READ unter Vertauschung des Datenrichtung.

16 SECTRN: Diese Funktion ist schon vom CP/M 2.2 System her bekannt und übernimmt die Umrechnung der logischen in die physikalische Sektornummer mit Hilfe einer Übersetzungstabelle.

23 MULTIO: MULTIO gibt an, daß die nachfolgenden Read/Write Befehle mehrere Sektoren umfassen, die sequentiell zwischen Speicher und Diskette transferiert werden.

24 FLUSH: Flush dient zum Schreiben noch nicht gesicherten Zwi-

schenspeicher auf Diskette. Diese Puffer dienen in der Regel dem Blocking/Deblocking. Doch auch im Track-Puffer können sich noch nicht gesicherte Daten befinden, die von vorhergegangenen Write-Befehlen zwischengespeichert wurden. Flush bewirkt unter CBIOS-3, daß der Track-Puffer - falls beschrieben - auf Diskette gesichert und sein Inhalt für ungültig erklärt wird.

### **Speicherselektion und Transfers:**

Folgende vier BIOS-Funktionen dienen der Anwahl einer Speicherbank oder dem schnellen Datentransfer zwischen verschiedenen Speicherblöcken.

- 25 MOVE: MOVE wird aufgerufen um größere Datenmengen im Speicher zu transferieren. Der Transfer kann dabei innerhalb der gleichen Speicherbank oder zwischen verschiedenen Speicherbanks stattfinden. Unter dem System 8023 wird dazu der Z80 Block-Move-Befehl LDIR eingesetzt. Muß zwischen verschiedenen Banks transferiert werden, so werden die Daten 128-Byte Weise über einen Zwischenpuffer kopiert, indem zwischen Start- und Zielbank umgeschaltet wird. Besonders wirkungsvoll ist dieser Befehl unter dem System 8043, da die Z80-DMA zum Transfer eingesetzt wird, die in der Lage ist direkt zwischen verschiedenen Banks zu transferieren.
- 27 SELMEM: Diese Funktion dient zum Umschalten zwischen verschiedenen Speicherbanks um dort Programme auszuführen.
- 28 SETBNK: Beim Aufruf von 'Set Bank' wird eine Speicherbank definiert in die Daten beim Lesen vom Diskettenlaufwerk geschrieben werden oder umgekehrt beim Schreiben gelesen werden sollen. Die durch SELMEM spezifizierte Speicherbank für die Programmausführung wird von diesem Befehl nicht beeinflußt.
- 29 XMOVE: XMOVE wird aufgerufen falls die MOVE-Funktion zwischen verschiedenen Speicherbanks transferieren soll (Interbank Move). Es werden dabei Start- und Zielbank spezifiziert. Wird die MOVE-Funktion ohne vorhergehendes XMOVE aufgerufen, so sollen die Daten innerhalb der

derzeitige selektierten Bank transferiert werden.

#### **Uhrzeit- und Datumsunterstützung:**

Die BIOS-Funktion 26 dient zum Setzen und Lesen von Uhrzeit und Datum. Das Datum wird dabei in Tagen seit dem 1. Januar 1978 gezählt. Da der auf dem System eingesetzte Uhrenbaustein auch das Datum getrennt nach Jahr, Monat und Tag verarbeitet, müssen diese Angaben noch entsprechend umgerechnet werden.

#### **Residente BIOS-Funktionen:**

Auf Grund der Aufteilung in 'banked' und 'non-banked' Bereiche ist es Anwenderprogrammen nicht mehr ohne weiteres möglich die BIOS-Funktionen direkt aufzurufen. Der BIOS-Sprung-Vektor startet bei 0F600h im Common-Bereich und ist von jeder Bank aus erreichbar, doch die Routinen selbst liegen nur teilweise im Common-Bereich. Folgende BIOS-Funktionen können jederzeit von Anwenderprogrammen aufgerufen werden, da sie im Common-Bereich liegen bzw. selbstständig auf die Systembank 0 umschalten:

WBOOT, DEVTBL, DEVINI, CONST, CONIN, CONOUT, LIST, LISTST, CONOST, AUXOUT, AUXIN, AUXIST, AUXOST, MOVE, SELMEM, TIME.

Alle restlichen BIOS-Funktion dürfen von der TPA-Bank 1 aus nicht aufgerufen werden. Ist dies dennoch notwendig, so steht dem Programmierer dafür die BDOS-Funktion 50 'Direct BIOS-Call' zur Verfügung.

#### **Interrupts:**

Um mit Interrupts in einem gebankten System zu arbeiten gibt es folgende Regeln. Die Z80 CPU arbeitet unter dem System in der Vektorinterrupt-Betriebsart IM 2. Der Interruptvektor der CPU zeigt auf die oberste Bank 0FFh. Wichtig ist, daß die Interruptvektortabelle und die Interrupt-Service-Routine im Common-Bereich des Speichers stehen, d.h. sie müssen von jeder Bank aus erreichbar sein. Handelt es sich um eine längere Interrupt-Routine, so kann deren Hauptteil auch im gebankten Speicherbereich stehen. Der Einsprungpunkt des Interrupts jedoch muß im Common-Bereich stehen und auf die Interrupt-Bank umschalten. Nach dem Verlassen der Routine muß wieder auf die ursprüngliche Bank zurückgeschaltet werden. Der ursprüngliche Wert des Stackpointers muß beim Verlassen der Routine erhalten bleiben.

**Die I/O-Geräte:**

Das CBIOS-3 verwaltet standardmäßig 6 Ein/Ausgabeeinheiten. Beim Kalt-Start des Systems sind natürlich bestimmte Voreinstellungen für Zuordnung zu den logischen CP/M-Einheiten, Baudraten und Protokolle getroffen. Im Betrieb lassen sich diese Einstellungen mit dem DEVICEE-Kommando jederzeit ändern. Sollen jedoch schon beim Kaltstart andere Einstellungen wirksam sein, so muß der USRDEF0-File des BIOS entsprechend modifiziert und ein neuer CPM3.SYS File geschaffen werden.

Device Name	Gerät	Protokoll	Baud	Bit	Stop	Parity	Dir	
0	CRT	RGB-Term.	rev. XON	9600	8	1	none	I/O
1	V24-A	RS 232 A	XON/XOFF	9600	8	1	none	I/O
2	V24-B	RS 232 B	XON/XOFF	1200	7	2	even	I/O
3	CENTR	Centronic	none	none	8	0	none	O
4	PARA	parallel	none	none	8	-	none	I
5	TTLSER	STI-ser.	XON/XOFF	9600	8	1	none	I/O
6	DIABLO	RS 232 B	ETX/ACK	1200	7	2	even	I/O

**Parallele Schnittstellen:**

Neben einer nach der Centronics-Norm arbeitenden Druckerschnittstelle wird beim System 8030 - 8043 noch eine zweite Parallel-Schnittstelle als Eingang unterstützt. Die Centronics-Schnittstelle arbeitete mit den Steuersignalen 'BUSY' und 'Paper-Empty' um den Status des Druckers abzufragen.

**Serielle Schnittstellen:**

Das CBIOS-3 stellt drei bidirektionale serielle Schnittstellen zur Verfügung: Zwei serielle, nach RS 232/ V24 gepufferte Schnittstellen, dienen zum Anschluß eines Terminals, Druckers oder anderer externer I/O-Einheiten.

Die **RS-232 A** Schnittstelle arbeitet standardmäßig mit einer Baudrate von 9600 Baud. Es werden 8 gültige Datenbits, 1 Stoppbit ohne Parität übertragen. Es findet das XON/XOFF-Protokoll Verwendung. Damit eignet sich dieser Kanal zum Anschluß der meisten modernen RS 232 - Geräte.

Die **RS-232 B** Schnittstelle ist für den Anschluß langsamer RS-232



Einheiten gedacht. Die Baudrate ist auf 1200 Baud reduziert, es werden 7 Datenbits, 2 Stoppbits mit gerader Parität übertragen. Als Protokoll wird XON/XOFF (V24-B) oder ETX/ACK (DIABLO) verwendet. Beim Lesen von der Schnittstelle wird das oberste Datenbit 7 auf Null zurückgesetzt.

Die dritte serielle Schnittstelle dient z.B. dem Anschluß einer seriellen Tastatur mit TTL-Pegel. Dabei werden 8 Datenbits und 1 Stoppbit ohne Parität übertragen. Die Übertragungsfrequenz beträgt 9600 Baud. Es findet eine spezielle Protokollart Verwendung: Wird ein Zeichen von der Schnittstelle empfangen, so wird ein XON-Zeichen zurückgesendet, um die Empfangsbereitschaft für ein neues Zeichen zu signalisieren. Wird dieses Protokoll nicht benutzt, so ist die Leitung TxD (Transmitted Data) zum I/O-Gerät nicht anzuschließen. Ein Protokoll ist nur bei sehr schnell arbeitenden Tastaturen erforderlich, die nicht nur einzelne Codes, sondern ganze Zeichenketten auf einmal ausgeben können (z.B. KeyStar).

#### **RGB-Terminal:**

Wird das RGB-Grafik-Terminal an das System angeschlossen, so wird dies als Console über den Namen 'CRT' angesprochen. Die Terminal-Software ist dabei schon in das BIOS eingebunden. Auszugebende Zeichen werden an das RGB-Modul über den ECB-Bus übergeben, über den seriellen TTL-Kanal werden Zeichen empfangen.

#### **I/O-Geräte-Voreinstellung:**

Beim Kaltstart des Systems prüft das CBIOS-3, ob das RGB-Terminal an das System angeschlossen ist. Verläuft der Test positiv so erfolgt automatisch folgende Voreinstellung:

```
CON    = CRT    -> RGB-Grafik Terminal
AUX    = V24-A  -> RS 232 A über ext. Terminal
LST    = CENTR  -> Paralleler Centronics Drucker
```

Erkennt das BIOS das RGB-Grafik-Terminal nicht, so wird folgende Voreinstellung getroffen:

```
CON    = V24-A  -> RS 232 A über ext. Terminal
AUX    = V24-B  -> RS 232 B zweites ext. Terminal oder Drucker
LST    = CENTR  -> Paralleler Centronics Drucker
```

**Abschalten der automatischen I/O-Voreinstellung:**

Ist diese automatische Voreinstellung des BIOS unerwünscht, so kann diese durch Modifikation des `USRDEF0`-Files abgeschaltet und undefiniert werden (s. `USRDEF0.ASM`-File). Dazu dient das Flag bei der Marke 'Bintab' im `USRDEF0`-File. Ist dieses Flag auf einen Wert ungleich Null gesetzt, so findet keine automatische Voreinstellung statt, sondern die Werte aus der dem Flag folgenden Tabelle werden direkt übernommen.

**Baudraten:**

Den drei seriellen Schnittstellen können jeweils getrennte Baudraten zwischen 110 und 19,6 kBaud zugeordnet werden. 1800, 3600 und 7200 Baud sind dabei Näherungswerte. Device 0 und 5 und Device 2 und 6 unterscheiden sich nur in der Art des Protokolls. Ihre Baudrate muß immer identisch gewählt werden, da es sich um die gleiche physikalische I/O-Einheiten handelt. Beim Device 6 'DIABLO' wird von einer Puffergröße des Empfangkanals von 159 Zeichen ausgegangen.

**Disk I/O:**

Bei der Auslegung der Disk I/O-Routinen wurde besonders auf eine rasche Abwicklung des Datentransfers von und zum Laufwerk gelegt. Dazu sind die Kopfladezeiten, die Motor on/off Steuerung und die Seek-Kommandos genau aufeinander abgestimmt, so daß unnötiges Warten auf die Bereitschaft der Laufwerke vermieden wird. Das BIOS 'kennt' von jedem der vier Laufwerke die Position der Schreib-Lese-Köpfe und ihren Ladezustand, so daß unnötige Seek-Kommandos mit entsprechend notwendigen Zeitverzögerungen zum Absenken und zur Beruhigung des Schreib-Lese-Kopfes vermieden werden. Das BIOS unterstützt die Funktion MULTIO vollständig. Auch Disketten mit einem Interleaving-Faktor größer als 1 sind davon nicht ausgenommen. Nach dem Aufruf der Funktion und einem entsprechenden Read/Write-Kommando werden in einem Diskettenzugriff und einer Laufwerksumdrehungen mehrere Sektoren (maximal ein Track) transferiert. Dies beschleunigt die Diskettenoperationen erheblich. Sehr viel Wert wurde beim Erstellen des CBIOS-3 auf eine komfortable Auto-Disk-Selekt-Funktion gelegt. Man kann an jede der vier Laufwerksnummern verschiedene Laufwerke anschließen und verschieden formatierte Disketten einlegen. Parameter wie Aufzeichnungsdichte, Spurdichte, Anzahl der Seiten und Sektoren werden für jedes Laufwerk getrennt ermittelt und verarbeitet. Zudem besteht die Möglichkeit auf einem Laufwerkstyp Formate anderer Laufwerke zu Lesen oder zu Schreiben und zusätzliche Formate in das CBIOS-3 einzubinden.

**Disketten Wechsel:**

Wird auf ein Laufwerk das erste Mal zugegriffen, so führt das BIOS für dieses Laufwerk eine Login-Prozedur durch, in der das Format der eingelegten Diskette ermittelt wird. Ein Diskettenwechsel ist unter CP/M-Plus jederzeit auch ohne '^C' zu Drücken möglich, soweit das gleiche Format wie zuvor verwendet wird. Wird ein anderes Format eingelegt, so kann es zu Schreib-Lese-Fehlern kommen. Erkennt das BIOS einen Fehler, so untersucht es ob der Fehler auf das Einlegen eines neuen Formats zurückzuführen ist. Ist dies der Fall, so wird automatisch ein neuer Login für dieses Laufwerk durchgeführt. Um jedoch diesen unsicheren Betriebszustand zu vermeiden, sollte nach dem Wechsel auf ein neues Diskettenformat '^C' betätigt werden. Dies führt dazu, daß für alle Laufwerke ein 'Logout' durchgeführt wird, d.h. bei Erst-Zugriff auf das Laufwerk wird sein Format neu ermittelt.

**Disk Selekt:**

Die BIOS-Funktion 9 SELDSK ist für das Login der Laufwerke verantwortlich. Zuerst wird geprüft, ob das angewählte Laufwerk existiert. Laufwerksnummern zwischen A und D werden akzeptiert. Ist ein Ram-Floppy implementiert, so kann noch ein fünftes logisches Drive E angesprochen werden. Danach wird geprüft, ob für das gewählte Laufwerk bereits ein Login durchgeführt wurde. Ist dies der Fall, so werden die bereits bekannten Werte für die Kopfposition als aktuelle Positionsparameter verwendet. Ist das Laufwerk im Logout-State, so wird ein Login für diese Laufwerksnummer durchgeführt.

**Disk Login:**

Die Login-Routine verwendet die im AUTODPBO-File vordefinierte Disk-Formate, die nach entsprechender Modifikation auf 4 mal so viele Formate ausgeweitet werden. Eines dieser Formate muß dem aktuellen Format entsprechen. Die vordefinierten Formate können in einem speziellen BIOS-Modul geändert werden. Entspricht keines dieser Formate dem aktuellem Format so erzeugt dies folgende Fehlermeldung:

```
BIOS Error on X , Login  
Track-0000, Sector-0000  
Retry <Y> ?
```

Die Login-Routine erwartet nun die Eingabe von 'Y' um einen neuen Login-Versuch zu starten. Alle anderen Eingaben brechen das Login ab und geben den Fehler an das die SELDSK-Funktion aufrufende Programm weiter. Im Login haben die Angaben 'Track und Sector' keine Bedeutung und stehen immer auf 0. Findet die Login-Routine das entsprechende Format, so werden die zugehörigen Parameter als aktuell übernommen, entsprechende Werte für die Disk-Parameter-Base des CP/M-Systems werden angeglichen.

**Fehlermeldungen beim Disk I/O:**

Beim Lesen oder Schreiben von und zum Laufwerk kann es zu Übertragungsfehlern kommen, die behebbar sind. Daher startet das BIOS automatisch bei Fehlern zehn weitere Schreib/Lese Versuche. Tritt ein Fehler zehn mal auf, so erscheint z.B. folgende Meldung.

```
BIOS Error on D , Write  
Track-0053, Sector-0014  
Not Writable,  
Retry <Y> ?
```

Wird nach dieser Meldung Y eingegeben, so werden 10 weitere Versuche gestartet. Andere Eingaben führen zum Abbruch des Datentransfers und einer Fehlermitteilung an das aufrufende Programm. Im Beispiel könnte der Benutzer den Fehler beheben, indem er den Schreibschutz ('Not Writable') von der Diskette entfernt.

#### **Fehler-Track/Sektor:**

Die Angaben dienen der Lokalisierung des Fehlers. 'Track' gibt die logische Position des Schreib/Lese-Kopfes an. Um die physikalische Position des Kopfes zu ermitteln, ist die Track-Angabe bei zweiseitigen Laufwerken durch zwei zu dividieren. 'Sector' entspricht dem physikalischen Sektor, der gelesen oder beschrieben werden sollte. Beide Angabe erfolgen dezimal.

#### **Fehlerart:**

Folgende Status-Meldung, die auch gemischt auftreten können, geben Hinweise auf die vorliegende Fehlerart.

'End of Cylinder': Es wurde eine Sektornummer angewählt, die größer als die tatsächlich vorhandenen Sektoren war.

'Data Error': Ein Übertragungsfehler im Daten- oder ID-Field ist aufgetreten, da die gelesenen CRC-Bytes nicht mit den neu errechneten übereinstimmen. Die CRC-Bytes (Cyclic redundancy check) werden in einer Schreib-Operation aufgezeichnet und beim Lesen zum Überprüfen der korrekten Übertragung benutzt. Ursache für diesen Fehler ist meist ein schlechtes Diskettenmaterial, zu lange Übertragungswege, unsauber positionierter Kopf oder eine von fremden Laufwerken beschriebene Diskette.

'Over Run': Die zu übertragenden Bytes vom oder zum Laufwerk können nicht rechtzeitig vom CBIOS-3 verarbeitet werden. Ursachen sind zu niedrige

Taktfrequenzen des Systems oder gesperrte Interrupts durch Anwenderprogramme.

- 'No Data': Dieser schwere Fehler tritt auf, falls die gewünschte Sektornummer auf der Diskette nicht gefunden werden kann.
- 'Not Writable': Es wurde versucht, auf eine mit Schreibschutz versehene Diskette zu Schreiben.
- 'No Adress Mark': Der Anfang eines Sektors wird nicht gefunden, da eine fremde oder defekte Diskette gelesen oder beschrieben werden soll.
- 'Control Mark': Eine Sektor mit einer sog. 'deleted data adress mark' sollte gelesen werden. Diese Adress-Mark dient zum Markieren defekter Sektoren.
- 'in Data Field': Der Fehler trat im Daten-Feld des Sektors auf. Fehlt diese Meldung so trat der Fehler im Id-Field des Sektors auf.
- 'Wrong Cylinder': Dieser Fehler tritt im Zusammenhang mit 'No Data' auf. Die gewünschte Tracknummer stimmt nicht mit der tatsächlichen Kopfposition überein. Dieser sog. Seek-Fehler ist auf eine falsch eingestellte Step-Rate oder dejustierte Laufwerke zurückzuführen.
- 'Bad Cylinder': Es wurde versucht eine nicht existierende Spur anzuwählen. Diese Meldung tritt zusammen mit dem 'No Data'-Fehler auf.

**Standard-Formate:**

Im CBIOS-3 sind zehn gebräuchliche Standard-Formate definiert, die nach selbstständiger Modifikation durch das Bios auf 40 Formate ausgeweitet werden. Das Bild zeigt eine Übersicht dieser Standard-Formate des Systems. All diese Formate können gleichzeitig auf beliebigen Laufwerken (A-D) benutzt werden. Erkennt das BIOS keines dieser Formate so leitet es automatisch 40 weitere Versuche ein, mit dem Unterschied, daß die Standardformate automatisch abgeändert werden: Alle Formate die als zweiseitig definiert wurden (Double Sided) werden nur einseitig gelesen. Somit entfällt das Definieren neuer Tabellen sonst gleicher Formate für einseitige Laufwerkstypen. Die zweite automatische Formatänderung des BIOS ist das sogenannte 'double-stepping'. Dazu wird einfach die doppelte Spur der gewünschten Spurnummer angewählt. Dies ist sehr praktisch um z.B. 40-Track Disketten (z.B. ECMAC-70, NCR DM-V ...) dennoch auf den modernen 80-Track Laufwerken bearbeiten zu können.

**Mini-Formate 80-Tracks:**

Format 1 ist das **Standard-Format** für moderne 5.25 und 3.5-Zoll Laufwerke mit 80 Spuren (beim erstellen einer Systemkopie soll dieses Format verwendet werden). Wie bei allen 80-Track Laufwerken ist ein Diskettenmaterial mit doppelter Spurdichte (96 tpi) zu verwenden. Besonders interessant ist Format 3, das auf den gleichen Laufwerkstypen zu verwenden ist, jedoch einen schnelleren Zugriff durch die großen Sektoren (1024 Bytes) zeigt und die Diskette optimal ausnutzt.

**8" kompatible Mini-Formate:**

Diese Formate sind für den Einsatz auf 8 Zoll kompatiblen Mini-Laufwerken gedacht. Allerdings sind dabei sehr hochwertige Disketten einzusetzen, da eine erhöhte Spurdichte (96 tpi) und eine größere Aufzeichnungsdichte (9646 bpi) vorliegt. Geeignet sind z.B. Maxell-MD2-HD (Double Sided, Double Track, High Density). Format 4 zeigt die höchste Diskettenkapazität mit 1.4 MByte.

**Maxi-Formate:**

Dies sind beides Formate für echte 8-Zoller, die sehr häufig anzutreffen sind. Format 8 stellt das bekannte IBM 3740 Format dar, das das einzige wirkliche genormte Diskettenformat ist. Es

gewährleistet eine sehr sichere Aufzeichnung, da es große Toleranzen im Kopfpositionier-System zulässt. Daher wird es besonders gern beim Kauf neuer Software eingesetzt. Format 5 ist auch sehr häufig auf 8 Zoll-Maschinen anzutreffen, zeigt dabei eine vernünftige Kapazität von ca. 1,1 MByte. Beide Formate sind natürlich auch auf den 8-Zoll kompatiblen Mini-Laufwerken einzusetzen.

#### **Mini-Formate 40 Tracks:**

Diese Mini-Formate sind für 40-Track Laufwerke geeignet. Format 6 besitzt eine hohe Kapazität von 400 kByte. Format 7 ist das sogenannte ECMA-70 Standard-Format und ist sehr häufig anzutreffen, doch es besitzt nur eine Kapazität von 160 kByte. Format 9 ist z.B. auf den Rechnern von NCR DM-V anzutreffen. All diese Formate können Dank des automatischen 'double-Stepping' des BIOS ohne jede Änderung auch auf modernen 80-Spur Laufwerken gelesen und geschrieben (!) werden.

#### **Directory-Einträge und Blockgröße:**

Gerade bei Disketten hoher Kapazität unter CP/M-Plus ist die Anzahl der Directory-Einträge sehr entscheidend, da die Zeit- und Datumsmarkierungen zusätzliche Directory-Einträge erzeugen. Dabei kann es leicht passieren, daß obwohl die Diskette noch gar nicht 'voll' ist kein Platz mehr für neue Directory-Einträge vorhanden ist. Daher sollte die Anzahl dieser Einträge nicht zu knapp bemessen sein (minimal 128). Doch das BDOS muß oft das Directory nach bestimmten Einträgen durchsuchen, was mehr Zeit bei vielen Einträgen in Anspruch nimmt. Ein dritte wichtige Größe stellt die Blockgröße dar. Die Blockgröße gibt an wieviel Bytes der Diskette pro Eintrag reserviert werden. Erhöht man die Blockgröße z.B. auf 4 kByte, so kann mehr Speicher auf der Diskette adressiert werden und die Anzahl der Directory-Einträge kann gesenkt werden. Doch bearbeitet man nur sehr kleine Dateien dann wird auf Grund zu großer Blockgrößen sehr viel Platz auf der Diskette verschwendet, da ein Eintrag mehr Platz reserviert als für eine kleine Datei notwendig wäre. Zum Bearbeiten großer Dateien (z.B. WordStar, DBASE, PASCAL, lange Texte ...) ist eine große Blockgröße zu wählen, die Anzahl der Directory-Einträge kann gesenkt werden und ein schneller Zugriff auf diese Dateien ist gewährleistet. Arbeitet man mit kleinen Dateien so ist die Blockgröße zu senken und dafür die Anzahl der Directory-Einträge zu erhöhen. Man kann unter CBIOS-3 natürlich gemischt verschiedene Formate einsetzen, die auf die jeweilige Dateigrößen optimal abgestimmt sind.



## Übersicht der Standard Formate des CBIOS-3:

Format Nummer	1	2	3	4	5	6	7	8	9	10
Mini = 0										
Maxi = 1	0	1	0	1	1	0	0	1	0	1
Double/Single Sided	DS	DS	DS	DS	DS	DS	SS	SS	DS	DS
Double-Single Density	DD	DD	DD	DD	DD	DD	DD	SD	DD	DD
Sektorgröße in Bytes	512	512	1024	1024	512	512	256	128	512	512
Sektoren/Spur	9	16	5	9	15	10	16	26	8	14
Anzahl der Spuren	80	80	80	80	77	40	40	77	40	80
Anzahl der Systemspuren	4	4	4	4	4	4	4	2	3	4
Skew-Faktor	3	3	6	5	5	3	1	6	1	3
Blockgröße in kByte	2	2	4	2	2	2	1	1	2	4
Directory Einträge	128	256	128	256	128	64	64	64	128	128
Speicherkapaz. formatiert kB.	720	1280	800	1440	1155	400	160	250	320	1120

**RAM-Floppy Erweiterung:**

Nach einer Ram-Floppy Erweiterung steht dem Anwender ein fünftes logisches Drive zur Verfügung. Voraussetzung dazu ist eine entsprechende Speichererweiterung des Systems über 128 kByte hinaus. Bis zu 1 MByte Ram-Bereich werden zur Abwicklung aller herkömmlichen Disk-Operationen unter enormen Geschwindigkeitssteigerungen verwendet. Die Leistungssteigerung kommt durch die hohe Daten-Transferrate zustande, ohne daß zeitraubende Seek-Operationen, Kopfladezeiten, Beruhigungszeiten usw. abgewartet werden müssen.

Beim Kaltstart des Systems führt das BIOS einen Speichertest durch, der alle Pages umfaßt. Der vorhandene Speicherplatz wird ermittelt und alle dafür notwendigen Parameter automatisch angeglichen. Am Anschluß an den RAM-Test wird der Directory-Bereich initialisiert, d.h. alle Directory-Einträge werden gelöscht. Der RAM-Bereich muß dabei kontinuierlich ab Bank 2 angeordnet sein. Es ist nicht zulässig eine Bank auszulassen, und eine nachfolgende Bank wieder zu bestücken.

Alle Diskoperationen, so z.B. auch das Kopieren von Files zwischen Ram-Floppy und physikalischen Laufwerken, sind möglich. Die Ram-Floppy wird unter der Laufwerksnummer E angesprochen. Übertragen werden jeweils 128 Bytes pro Sektor. Die CP/M Data-Allocation-Size beträgt 2048 Byte Blöcke. Dabei sind bis zu 64 Directory-Einträge möglich.

**Systemstruktur:****Der CPM3.SYS File:**

Der größte Teil des CP/M-Plus-Betriebssystems befindet sich in der Systembank 0 und ist für Anwenderprogramme transparent. Unter CBIOS-3 steht das gesamte CP/M-System nicht auf den Systemspuren, sondern auf den Datenspuren im File CPM3.SYS. Man unterscheidet vier Betriebssystem-Teile: Gebanktes BDOS, BIOS und residentes BDOS, BIOS. Beim Start des Systems werden diese vier Teile vom CPM3.SYS-File an ihre entsprechenden Positionen in den Systemspeicher geladen. Danach wird dieser File nicht mehr angesprochen, es kann z.B. die Systemdiskette in Laufwerk A durch eine andere Datendiskette ausgetauscht werden.

**Der CCP.COM File:**

Eine Kopie des Consol-Command-Prozessor's (CCP) befindet sich im untersten Teil der Systembank 0. Da der CCP unter CP/M-Plus sehr groß ist, bildet der CCP anders wie unter CP/M 2.2 ein ganz normales lauffähiges Programm (CCP.COM-File), das in der TPA ab 100h abläuft. Wird ein Programm aufgerufen so überschreibt dies den ursprünglichen CCP, d.h. dieser muß nach Beendigung des Programms wieder neu in die TPA nachgeladen werden. Diese Aufgabe übernimmt das BIOS (Warmboot), das ohne jeden Diskzugriff die Kopie des CCP's von Bank 0 in die TPA Bank 1 transferiert und ausführt. Ähnlich wie beim CPM3.SYS-File wird daher unter CBIOS-3 der CCP.COM-File nach dem System-Startup nicht mehr angesprochen.

**Der Coldboot-Ladevorgang:**

Das Laden des CP/M-Systems unter CBIOS-3 in den Speicher vollzieht sich in folgenden vier Schritten:

- (1) Der Boot-Loader, ein im Boot-Eprom befindliches Programm, sorgt nach dem Einschalten des Systems dafür, daß der CP/M-Loader von den Systemspuren in den RAM geladen und ausgeführt wird.
- (2) Die Aufgabe des CP/M-Loaders besteht darin, das eigentliche CP/M-Plus System von den Datenspuren der Systemdiskette richtig in die entsprechenden Banks zu plazieren und einen BIOS-Coldboot auszuführen.

- (3) Die Coldboot-Routine lädt den CCP von den Datenspuren in Bank 0 (0000h - 0D00h), nimmt die restlichen Initialisierungen vor und springt die BIOS-Warmboot Routine an. Diese Funktion wird immer nach dem Verlassen eines Anwenderprogramms aufgerufen und dient hauptsächlich dem Laden des CCP's in den TPA-Bereich bei 100h.
- (4) Warmboot transferiert dazu einfach die Kopie des CCP's aus Bank 0, die ja die Coldboot-Routine angelegt hat, in den TPA-Bereich und führt den CCP aus. Nun erscheint die CCP-Message 'A> ' auf dem Bildschirm, d.h. das System ist bereit.

Der Grund für diesen vielschichtigen Ladevorgang und das Auslagern des Betriebssystems auf die Datenspuren der Systemdiskette liegt in seinem Platzbedarf. Auf kleinen Diskettenformaten ist nicht genug Platz um das BDOS, CCP und BIOS auf den Systemspuren abzuspeichern. Daher mußten diese Teile auf die normalen Datenspuren der Systemdiskette ausgelagert werden. Dies nimmt zwar Speicherplatz der Diskette in Anspruch, hat aber einen entscheidenden Vorteil: Das gesamte CP/M-System (CPM3.SYS) wird wie jeder andere normale File behandelt, kann also beliebig modifiziert und kopiert werden ohne auf seine Länge achten zu müssen. Änderungen im BIOS können jederzeit vorgenommen werden ohne z.B. einen Boot-Eprom auswechseln zu müssen.

#### **Der Boot-Eprom:**

Nach dem Einschalten der Versorgungsspannung sorgt die Power-On Reset Schaltung für ein ordnungsgemäßes Rücksetzen des Systems (Reset). Der Memory-Prom des Systems sorgt dafür, daß der BOOT-Eprom des Computers im Bereich von 0 - 3FFFh (0 - 0FFFh) in den Adressraum der CPU geschaltet wird und daß sich die entsprechenden Adressen des dynamischen RAM's nicht mit denen des Boot-Eprom's überlagern, d.h. die untersten 16k (4k) des RAM werden ausgeblendet (Deselect). Die CPU arbeitet nach dem Reset den Boot-Eprom beginnend bei 0000h ab. Die Aufgabe des Boot-Eproms besteht darin das System zu initialisieren (STI, DART, RGB-Farbgrafik ... ), einen RAM-Test durchzuführen um evt. eine RAM-Floppy einrichten zu können und initialisieren der RAM-Floppy falls mehr als 128 kByte Speicher vorhanden sind. Der Boot-Loader untersucht, ob z.B. das RGB-Grafik-Terminal oder ein anderes externes Terminal über die RS-232-A Schnittstelle als Console angeschlossen ist. Je nachdem lenkt er die entsprechenden Ein-

und Ausgaben um. Der Boot-Eprom ermittelt das Format der in Laufwerk A befindlichen Systemdiskette. Es werden die gleichen 40 Formate erkannt wie beim eigentlichen CBIOS-3.

Nach der Formaterkennung lädt der Boot-Eprom die Systemspuren (Track 0,1 / bei double sided Laufwerken beide Seiten) in den RAM-Speicher ab 4000h (2000h) aufwärts. Auf den Systemspuren befindet sich normalerweise der CP/M-Loader. Nach dem Laden führt der Boot-Eprom einen Sprung nach 4000h (2000h) aus und übergibt die Kontrolle des Systems an den dort befindlichen CP/M-Loader.

#### **Der CP/M-Loader:**

Die Aufgabe des CP/M-Loaders ist das CP/M-Plus System vom CPM3.SYS File zu lesen und die entsprechenden Teile in den residenten und gebankten Teil des Speichers zu laden. Der CPM3.SYS File enthält dazu auch die Informationen wie groß die vier Teile des Systems sind und wohin sie geladen werden müssen. Da der CPM3.SYS File wie jeder andere normale File auf den Datenspuren der Diskette steht, muß seine Position auf der Diskette über die Directory-Einträge errechnet werden. Dazu bedarf es einer genauen Kenntniss der File-Struktur des Systems, so wie das eigentliche BDOS Daten von der Diskette liebt. Daher stellt der CP/M-Loader ein einfaches nichtgebanktes CP/M-System dar das aus zwei Teilen, dem CPMLDR und dem LDRBIOS (vgl. BDOS und BIOS), besteht. Das Loader-BIOS behandelt alle hardwareabhängigen Funktionen zum Laden des CPM3.SYS Files und besteht wie das normale BIOS aus einer Sprungleiste zu den einzelnen Funktionen am Anfang und den dazugehörigen Routinen. Genauso wie das eigentliche CBIOS-3 müssen auch hier wieder alle der 40 möglichen Diskettenformate erkannt werden und genaue Kenntnisse über deren Datenstruktur vorliegen. Nach dem plazieren des CPM3.SYS-Files in den System-speicher geschieht der Sprung auf die Bios-BOOT-Funktion zum eigentlichen Start des Systems.

Wird das BIOS des Systems verändert oder ist eine andere Speicheraufteilung erwünscht, so muß ein neuer CPM3.SYS File erstellt werden. Das Programm GENCPM dient der Erstellung eines CPM3.SYS-Files, das diesen durch Verbinden der Files BNKBDOS3.SPR (gebankter Teil des BDOS), RESBDOS3.SPR (resistenter Teil des BDOS) und dem BNKBIOS3.SPR (dem CBIOS-3) erstellt.

**Der BNKBIOS3.SPR-File:**

Wie der Name andeutet handelt es sich hier um den BIOS-File der mit dem BDOS gebunden zu einem CPM3.SYS-File wird. Die File-Extension 'SPR' besagt, daß dies ein 'System Page Relocatable'-File ist, der also noch nicht auf eine bestimmte Speicheradresse fixiert ist, sondern zu 256-Byte (eine Page) verschiebbar ist. Zudem besteht dieser File intern aus zwei Teilen, dem gebankten und dem residenten BIOS-Teil.

**Das Erstellen des BNKBIOS3-Files:**

Das CBIOS-3 aus zwölf verschiedenen Modulen, die völlig unabhängig voneinander bearbeitet werden können:

- BIOSKRNO.ASM:** Dies stellt das noch weitgehend hardwarunabhängige Grundgerüst des BIOS dar ('BIOS Kernel'), und enthält die BIOS-Jump-Vektoren. Von ihm aus geschehen die einzelnen Aufrufe an die anderen BIOS-Module.
- XMOVE0.ASM:** Dieses Modul enthält alle Funktionen zur Speicherselektion und zum Speichertransfer.
- TIME0.ASM:** Time stellt die BIOS-Funktion 26 Time zur Verfügung und steuert den Uhrenbaustein des Systems an.
- BOOT0.ASM:** Das Boot-Modul wickelt die Boot-Funktion des BIOS zum Erstellen und Nachladen des CCP's ab.
- AUTODPB0.ASM:** Dieses Modul enthält alle Informationen die zum Verarbeiten der verschiedenen Disk-Formate notwendig sind. Es enthält die Drive-Tabelle, die Disk-Parameter-Header, die Disk-Parameter-Bases (vgl. AUTODPB) und Umrechnungstabellen der logischen in die physikalischen Sektornummern (Skew-Tabellen).
- AUTODSK0.ASM:** Dieses Modul enthält die wichtigen Funktionen Disk-Read/Write und Login. Alle Ansteuerungsfunktionen zu den Laufwerken werden hier getätigt.
- TVI950.ASM:** Dieses Modul enthält die Definition der Steuerzeichen und Escape-Sequenzen des RGB-Terminals.

- CHARIOO.ASM:** CHARIOO ('Character I/O') enthält alle Sequenzen für die Zeichen Ein/Ausgabe. Es wickelt die verschiedenen Protokollarten ab und setzt die Baud-Raten fest.
- SCB.ASM:** Dieser sogenannte 'System Control Block' stellt eine Tabelle dar, über die Daten zwischen dem BDOS und dem BIOS ausgetauscht werden. Der SCB dient zum Beispiel zum Übertragen von Datum und Uhrzeit, er enthält die fünf I/O-Redirection Vektoren, gibt Informationen über die Art der Fehleranzeige des BIOS an und enthält die Eintrittsadresse in das BDOS.
- USRDEFO.ASM:** Die Device-Tabelle, die Aufschluß über die möglichen Zeichen-Ein/Ausgabe-Einheiten gibt, befindet sich in diesem Modul.
- RGBTERMO.ASM:** Ist das System mit dem RGB-Grafik Terminal ausgerüstet, so dient dieses Modul der alphanumerischen Zeichen-Ausgabe über die Grafik-Karte. Es werden 85x24 Zeichen in einer 8x12 Punkte Matrix dargestellt. Zahlreiche Attribute wie unterstreichen, invers, deutsch/internationaler Zeichensatz, Cursivschrift usw. sind in diesem Modul verwirklicht. Die Funktionen sind weitgehend dem TVI950-Terminal kompatibel. Die Definition der Control- und Escapesequenzen geschieht im Modul TVI950.
- USRROUT.ASM:** Dieses Modul dient der einfachen Erweiterung des BIOS. Jede BIOS-Funktion ruft vor Ihrer normalen Abarbeitung eine zugehörige Routine in diesem Modul auf. Soll das BIOS erweitert werden, so kann man auf einfache Weise durch Hinzufügen eigener Routinen in dieses Modul die jeweilige BIOS-Funktion erweitern oder abändern.

#### **Das Binden der Bios-Module:**

Zu dem Betriebssystem CP/M-Plus gehört unter anderem auch der relocative Assembler RMAC und der Linker LINK-80. Sie dienen dem Assemblieren und Binden des CBIOS-3. RMAC erzeugt aus dem Bios-Teil ein beliebig verschiebbares Modul mit der File-Extension 'REL'. Liegen alle BIOS-Module fehlerfrei als REL-Files vor, so

werden diese durch den Linker zu einem einzigen File BNKBIOS3.SPR verschmolzen. Die Reihenfolge des Bindens ist nach folgendem Beispiel zwingend vorgeschrieben:

```
A>link bnkbios3[b] =bioskrn0,xmove0,time0,boot0,autodpb0,autodsk0,
      tvi950,chario0,scb,usrdef0,rgbterm0,usrROUT,
      ... (yourfile)
```

Das Zeichen 'b' in eckigen Klammern veranlasst den Linker den 'System-Page-Relocatable'-File BNKBIOS3.SPR zu erzeugen.

### Zwei Teile des BIOS:

Wie schon erwähnt besteht das CP/M-System aus vier Teilen. Genau so wie das BDOS existieren 'gebankte' und 'resistente' Teile des BIOS. Durch Kennzeichnung mit dem Assembler-Pseudo-Op-Code CSEG ('Code Section') in der Quelldatei veranlasst man den Assembler Programmcode im resistenten Systemspeicher zu erzeugen, d.h. so werden die nicht-gebankten Teile des BIOS erzeugt. Das Zeichen DSEG ('Data Section') schaltet vom nicht-gebankten zum gebankten Teil um. Die Länge des resistenten BIOS-Teils jedoch ist begrenzt, da die Gesamt-Länge des resistenten Teils des Betriebssystems auf 4 kByte begrenzt ist, um einen möglichst großen TPA-Bereich von 60 kByte zu erhalten. Da die Länge des resistenten BDOS mit 1.5 kByte unveränderlich ist, bleibt für das resistente BIOS ein Maximal-Länge von 2.5 kByte (09FFh), die nicht überschritten werden darf. Der Linker gibt nach seinem Lauf unter anderem eine Meldung aus, wie groß der gebankte und resistente Teil des BIOS ist. Die Linker-Meldung muß auf jeden Fall folgendermaßen erscheinen:

```
ABSOLUTE          0000
CODE SIZE          09XX (0000 - 09XX)
DATA SIZE          ...  (0A00 - XXXX)
COMMON SIZE        0000
```

Der 'CSEG'-Bereich darf 2.5 kByte nicht überschreiten (0000-09XX), die Länge des 'DSEG'-Bereichs ist fast beliebig und nur durch die Größe der Systembank 0 begrenzt.

Liegt ein fehlerfreier BNKBIOS3.SPR-File vor, so muß dieser mit dem BDOS gebunden werden. Diese Aufgabe übernimmt das Programm GENCPM.



**Neue Systemgenerierung:**

Natürlich läuft das System mit der entsprechenden Systemdiskette auch ohne die unten aufgeführte Systemgenerierung. Sollte es jedoch nötig sein das BIOS zu ändern, so muß eine neue Systemgenerierung durchgeführt werden. Folgende Möglichkeiten das CP/M-System zu ändern gibt es:

- Nach Hinzufügen neuer Speicherbaugruppen muß dies durch GENCPM angepasst werden.
- Man ordnet den fünf Laufwerken anderen Puffer-Größen zu. Dies geschieht ebenfalls über GENCPM.
- Man fügt neue BIOS-Routinen dem System an. Dies geschieht über den USSRROUT-File.
- Will man Disk-Formate bearbeiten, die in der Standard-Version nicht bearbeitet werden können, so muß man den AUTODPBO-File ändern.
- Sollen die Standard-Parameter für die Ein/Ausgabe-Geräte wie Baudrate, Protokolle und Zuordnung zu den logischen I/O-Einheiten geändert werden, so muß dies im USRDEF0-File angepasst werden.

**So arbeitet GENCPM:**

GENCPM benötigt die vier Teile des CP/M-Plus Betriebssystems um den CPM3.SYS File zu erzeugen. Die beiden BIOS-Teile liegen im BNKBIOS3.SPR-File vor, die beiden BDOS-Teile werden mit dem Betriebssystem von Digital Research geliefert und tragen die Namen RESBDOS3.SPR und BNKBDOS3.SPR. Bevor GENCPM den CPM3.SYS-File aus diesen drei Eingangs-Files erzeugt, hat der Anwender noch eine Reihe von Möglichkeiten das Betriebssystem nach seinen Wünschen anzupassen. Die Anzahl der verfügbaren Speicherbanks, deren Nutzung, die Art der Fehlermeldungen usw. kann hier noch festgelegt werden. Im folgendem Beispiel wird gezeigt, wie die Fragen von GENCPM unter dem System 8023 beantwortet werden sollten:

A>GENCPM

CP/M 3.0 System Generation

Copyright (C) 1982, Digital Research

Default entries are shown in (parens).

Default base is Hex, precede entry with £ for decimal

Use GENCPM.DAT for defaults (Y) ? Y

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

Das angepasste CP/M-Plus für das System enthält im GENCPM.DAT-File Default-Werte, die alle richtigen System-Variablen beinhalten.

Display Load Map at Cold Boot (Y) ? Y

Beim Coldboot des Systems listet der CPM-Loader eine sog. 'Load-Map' aus, die Auskunft über die Speicherbelegung des Systems gibt. Ist dies erwünscht so muß mit <Y> geantwortet werden.

Number of console columns (£85) ? £85

Number of lines in console page (£24) ? £24

Backspace echoes erased character (N) ? N

Rubout echoes erased character (N) ? N

Das Betriebssystem benötigt Daten über die angeschlossenen Console-Einheit. Wird die RGB-Graphik-Karte als Terminal verwendet, so sind obige Werte zu übernehmen.

Initial default drive (A:) ? A

Diese Eingabe legt die Laufwerksnummer nach dem Coldboot des Systems fest.

Top page of memory (FF) ? FF

Bank switched memory (Y) ? Y

Common memory base page (C0) ? C0

Die Common-Memory-Base, also die Adresse wo der nicht gebankte Speicherbereich anfängt, liegt beim System 8023 (EPC) bei 48 kByte, also in der Page 0C0h. Unter dem System 8043 (CPU) liegt diese Adresse bei 0F0h.

Long error messages (Y) ? Y

Wird diese Frage mit <Y> beantwortet, so erzeugt das System sog. lange, d.h. sehr detaillierte und aufschlußreiche Meldungen falls ein Fehler auftritt. Allerdings benötigen diese Meldungen mehr Speicherplatz.

Accept new system definition (Y) ? Y

Setting up Allocation vector for drive A:,B:,C:,D:,E:  
Setting up Checksum vector for drive A:,B:,C:,D:,E:

GENCPM deutet mit dieser Meldung (verkürzt!) an, daß es für die fünf Laufwerke entsprechende Tabellenplätze reserviert.

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

Number of memory segments (l1) ? 1

Hier fragt GENCPM nach der Anzahl der Speichersegmente. Die Standard-Systemkonfiguration ohne Speichererweiterung hat 128 kByte Speicher, entsprechend 2 Speicherbanks. GENCPM rechnet die TPA-Bank 1 nicht in die freien Systemspeicherbanks mit ein, daher muß die Frage normalerweise mit <1> beantwortet werden. Steht nach einem Ausbau mehr Speicher zur Verfügung, so können diese weiteren Banks entweder der RAM-Floppy oder dem Betriebssystem zugeteilt werden. Wird eine RAM-Floppy eingesetzt, so muß die obige Frage auch mit <1> beantwortet werden, da die RAM-Floppy den Speicherplatz ab Bank 2 aufwärts benutzt.

CP/M 3 Base,size,bank (7B,45,00)  
Enter memory segment table:

Base,size,bank (31,4A,00) ? 31,4A,00

GENCPM zeigt in der ersten Meldung die Belegung der Systembank 0 mit dem Betriebssystem an. Das gebankte Betriebssystem (System 8023) hat eine Länge von knapp 32 kByte und erstreckt sich daher von 7B00h bis 0BFFFh. Der Bereich unterhalb von 7B00h ist frei zum Anlegen

der verschiedenen Puffer-Bereiche für das Betriebssystem. Nun muß eingegeben werden welcher Bereich des noch freien Teils der Bank 0 dem CP/M-Plus zugeordnet werden soll. Wie schon erwähnt befindet sich in Bank 0 eine Copy des CCP's und der BIOS-Track-Puffer, die standardmäßig den Bereich von 0-30FFh belegen. Die Länge des CCP's liegt fest und erstreckt sich von 0 - 0CFFh. Bei 0D00h startet der BIOS-Track-Puffer, dessen Länge sich nach der größten Track-Länge der verwendeten Formate richtet. Die Standard-CBIOS-Version hat als größte Tracklänge das 8 Zoll Format mit 9x1024 Byte pro Track, also reserviert der Puffer den 9 kByte langen Bereich von 0D00h - 30FFh. Für die CP/M-Puffer verbleibt der Bereich von 3100h bis 7AFFh mit einer Länge von 4Ah Seiten.

```
CP/M 3 Sys      7B00H 4500H  Bank 00
Memseg No. 00  3100H 4A00H  Bank 00
```

Accept new memory segment table entries (Y) ? Y

Setting up directory hash tables:

```
Enable hashing for drive A: (Y) ? Y
Enable hashing for drive B: (Y) ? Y
Enable hashing for drive C: (N) ? N
Enable hashing for drive D: (N) ? N
Enable hashing for drive E: (N) ? N
```

GENCPM fragt, für welches der fünf Laufwerke Hash-Tabellen angelegt werden sollen. Das 'Directory-Hashing' verkürzt den Disk-Zugriff, benötigt jedoch Speicherplatz. Daher sollte nur für diejenigen Laufwerke mit <Y> geantwortet werden, mit denen man hauptsächlich arbeitet.

Setting up Blocking/Deblocking buffers:  
The physical record size is 0400H:

Nun legt GENCPM Daten- und Directory Puffer für die fünf Laufwerke an. Es erkennt aus den Disk-Parameter-Header-Einträgen, daß die maximale Sektorgröße 1024 Bytes (0400h) beträgt. Nun läßt sich für jedes Laufwerk getrennt angeben, wieviele Daten- und Directory Puf-

fer ihm zugeordnet werden sollen. Da jeder Puffer Speicherplatz in Anspruch nimmt, steht natürlich nur eine begrenzte Anzahl an Puffern zur Verfügung, die nun optimal aufgeteilt werden sollten. Als Regel gilt: Die Laufwerke mit denen man am meisten arbeitet sollten auch die meisten Puffer bekommen. Für Laufwerk A sieht das folgendermaßen aus:

Available space in 256 byte pages:

TPA = 00F0H, Bank 0 = 0042H, Other banks = 0000H

Number of directory buffers for drive A: (£4) ? 4

Available space in 256 byte pages:

TPA = 00F0H, Bank 0 = 0031H, Other banks = 0000H

Number of data buffers for drive A: (£4) ? £4

Allocate buffers outside of Common (Y) ? Y

GENCPM fragt, ob es die Daten-Puffer außerhalb des Common-Memory-Bereichs installieren soll. Diese Frage sollte mit <Y> beantwortet werden, um den TPA-Bereich nicht unnötig herabzusetzen. Alle Puffer werden unter CBIOS-3 außerhalb des Common-Bereichs installiert, da das BIOS über die XMOVE-Funktion verfügt. Die folgenden GENCPM-Fragen werden aus Platzgründen etwas verkürzt dargestellt:

Number of directory buffers for drive B: (£4) ? 4

Number of data buffers for drive B: (£4) ? 4

Number of directory buffers for drive C: (£0) ? 0

Number of data buffers for drive C: (£0) ? 0

Share buffer(s) with which drive (B): ? B

Number of directory buffers for drive D: (£0) ? 0

Number of data buffers for drive D: (£0) ? 0

Share buffer(s) with which drive (B): ? B

Number of directory buffers for drive E: (2) ? 2

Da das RAM-Floppy (Laufwerk E) mit 128 Byte Sektoren arbeitet, werden für Drive E keine Datenpuffer angelegt. Die letzte GENCPM-Speicher-Meldung muß folgendermaßen aussehen:

Available space in 256 byte pages:

TPA = 00F0H, Bank 0 = 0000H, Other banks = 0000H

Die TPA muß auf Page 0F0h liegen, die Speicherbanks sollten keinen freien Platz mehr aufweisen.

Accept new buffer definitions (Y) ? Y

```
BNKBIOS3 SPR F600H 0A00H
BNKBIOS3 SPR A700H 1900H
RESBDOS3 SPR F000H 0600H
BNKBDOS3 SPR 7900H 2E00H
```

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

GENCPM hat seinen Lauf beendet und einen neuen CPM3.SYS-File auf Diskette erstellt. Um eine neue Systemdiskette zu erhalten, muß noch der File CCP.COM kopiert und mit COPYSYSN der CPM-Loader auf die Systemspuren gebracht werden.

#### Generierung des CPM-Loaders:

Der CPM-Loader stellt ein einfaches nicht-gebanktes CP/M-System dar, das auf den Systemspuren der Diskette steht. Der CP/M-Loader besteht beim System 8000 aus zwei Teilen: Dem von Digital Research gelieferten CPMLDR-BDOS (CPMLDR.REL) und dem System-spezifischen CPMLDR-BIOS. Das Loader-BIOS wiederum besteht aus dem Hauptteil LDRBIOS0 und dem Tabellenteil LDRDPB0, der formatspezifische Informationen ähnlich wie der AUTODPB0-File enthält. Ändert man das Loader-BIOS, so assembliert man die geänderten Teile neu und bindet diese nach folgendem Schema mit den übrigen zu einem neuem CPM-Loader zusammen:

```
A>RMAC LDRDPB0
A>LINK CPMLDR0.400 [L4000] =CPMLDR,LDRBIOS0,LDRDPB0
```

Die Anweisung 'L4000' (L2000) in eckigen Klammern veranlaßt den Linker, ein lauffähiges Programm, den CPM-Loader mit Namen CPMLDR0.400 (CPMLDR0.200), mit einer Startadresse von 4000h (2000h) zu erzeugen. Diese Adresse deckt sich genau mit der Adress des Boot-Laders, wohin dieser den CPM-Loader lädt und startet. Unter dem System 8043 (CPU-Modul) beträgt diese Adresse nicht 4000h sondern 2000h! Nachdem ein neuer CPM-Loader-File vorliegt, muß dieser mit Hilfe von COPYSYSN auf die Systemspuren kopiert werden: A>COPYSYSN CPMLDR0.400

**Die Struktur der Bios-Module:**

Auf der System-Diskette steht z.B. das Bios-Modul `USRDEFO.ASM`. In ihm wird die Device-Tabelle aufgestellt. Man sieht dort sehr gut, wie ein BIOS-Modul aufgebaut ist. Die erste Assembleranweisung `'cseg'` bedeutet, daß nachfolgender Code im Common-Memory-Bereich stehen soll. Die folgende Anweisung `'maclib modebaud'` weist den Assembler an, die Macro-Library `'Modebaud'` einzulesen. Diese Library enthält die Definitionen aller gültigen Werte, die in die Device-Tabelle eingesetzt werden dürfen. Die nächste Anweisung dient dem Binden der BIOS-Module. Die beiden Marken `'Bintab'` und `'Bctbl'` sind als 'öffentliche' (public) Marken definiert, auf die sich andere BIOS-Module beziehen können. Soll in einem anderen BIOS-Modul auf die Marke `'Bctbl'` zugegriffen werden, so muß in diesem Modul die Marke über `'extrn Bctbl'` als extern definiert werden. Eine der Aufgaben des Linkers ist es nun, all diese Marken, die in den verschiedenen Modulen als `'extern'` oder `'public'` definiert werden, durch absolute Adressen zu ersetzen.

**Der AUTODPBO-File:**

Der Inhalt des AUTODPBO-Files bestimmt die Disk-Formate, die bearbeitet werden können. Hier erfolgt eine Anpassung an das Laufwerks-Timing (Steprate, Headload-Time ...) und die exakte logische und physikalische Definition der Formate.

**Disktiming:**

Laufwerke verschiedener Hersteller weisen oft unterschiedliche Zeiten auf, mit denen sie den Schreiblesekopf bewegen. Im AUTODPBO-File können diese Zeiten exakt, getrennt für jedes Format bestimmt werden. Die Angabe 'hltg equ 32' im AUTODPBO-File besagt z.B., daß die Kopflade-Zeit in diesem Fall 32 ms beträgt. Die drei Angaben 'hltx' (Head-Load-Time), 'hutx' (Head-Unload-Time) und 'srtx' (Steprate-Time) werden nach entsprechender Umrechnung zu dem sog. Specify-Parameter 'specx' umgerechnet, der zur Programmierung des FDC-Controllers uPD 765 dient.

**Das Type-Flag:**

Das Type-Flag gibt dem BIOS einige physikalische Angaben über das betreffende Disk-Format:

- Bit 0 = 1 : Es findet die sog. FM-Aufzeichnung Verwendung. Ist dieses Bit auf 0 gesetzt, so werden die Daten mit doppelter Aufzeichnungsdichte übertragen.
- Bit 1 = 1 : Ist dieses Bit gesetzt, so liegt ein 5 1/4 Zoll kompatibles Format vor, andernfalls ein 8 Zoll Format mit doppelter Schreibtaktfrequenz (500/250 kHz).
- Bit 2 = 1 : Bit 2 dient zur Kennzeichnung älterer zweiseitiger Diskettenformate. Bei diesen Formaten wird von Spur 0 bis 39 der Kopf 0 des Laufwerks adressiert, bei Spur 40 bis 79 wird Kopf 1 Track 0 bis 39 adressiert. Ab welcher Spurzahl auf Spur 0, Kopf 1 zurückgeschaltet werden soll bestimmt die Marke 'REVTRK'. Standardmäßig geschieht dies ab Track 40 (z.B. NCR DM-V).

Ist dieses Bit auf Null gesetzt, so wird bei doppelseitigen Formaten das weitaus gebräuchlich-



ere Verfahren der Spur-Anwahl eingesetzt: Bei geraden Spurnummern wird Kopf 0, bei ungeraden Spurnummern Kopf 1 adressiert, d.h. bei steigenden Spurnummern wird immer zwischen Kopf 0 und Kopf 1 abgewechselt. Vorteil dieser Methode sind um die Hälfte reduzierte 'Step-Befehle'.

Bit 5 = 1: Ist diese Bit gesetzt, so liegt nur eine einseitig formatierte Diskette vor, d.h. es wird nur Kopf 0 des Laufwerks angesprochen. Ist dieses Bit rückgesetzt, so werden beide Köpfe des Laufwerks angesprochen.

Bit 6 = 1: Dieses sog. 'Skip'-Bit veranlasst das BIOS beim Anwählen einer Spur die Spurnummer zu verdoppeln, d.h. jede zweite Spur zu überspringen. Dies ist bei Einsatz von 80-Track-Laufwerken interessant. Mit Hilfe der Skip-Funktion können auf 80-Track Laufwerken auch ältere 40-Track Disketten verarbeitet werden.

#### **Die 'Disk Parameter Header':**

Für jedes Laufwerk existiert unter CP/M-Plus ein 'Disk Parameter Header' (DPH), der die aktuellen logischen Informationen über das Laufwerk enthält. Diese Tabellen wurden unter CBIOS-3 etwas erweitert und enthalten zusätzliche physikalische Informationen die das BIOS benötigt. Das Anlegen der Tabelle übernimmt die Macro-Library 'dph', die unter anderem über die Assembleranweisung 'maclib cpm3' eingelesen wird. Als Eingangsparameter benötigt dieses 'Macro' zwei Werte: Die Adresse eines Disk-Parameter-Blocks und die zugehörige Sektor-Übersetzungstabelle. Die Tabelle enthält unter anderem auch Informationen darüber, wie groß der 'allocation'- und 'checksum'-Vektor für dieses Laufwerk ist, die an Hand des Disk-Parameter-Blocks ermittelt werden. Da das CBIOS-3 über eine Auto-Disk-Selekt-Funktion verfügt, kann jedem Laufwerk ein beliebiges Format zugeordnet werden. Daher kann die Größe dieser Vektoren unterschiedlich sein. Um auf jeden Fall genügend Platz für diese Vektoren zu reservieren, sollte nach dem Macro-Aufruf 'dph' derjenige Disk-Parameter-Block eingesetzt werden, der den meisten Platz für diese Vektoren in Anspruch nimmt. Im Standard-BIOS ist dies das Format 4 mit 256 möglichen Directory-Einträgen und einer Kapazität von ca. 1.4 MByte.

### Die Drive-Tabelle:

Nach dem Disk-Parameter-Header im AUTODPB0-File steht die Laufwerks-Tabelle 'BDTBL', die wiederum ein Macro 'dtbl' anlegt. Diese Tabelle enthält die Adressen der Disk-Parameter-Header und bestimmt, wieviele Laufwerke vom BIOS verwaltet werden. Im Standard-BIOS werden vier Laufwerke (dphA - dphD) und eine Ram-Floppy (dphR) angesprochen. Sind weniger Laufwerke angeschlossen, so sind diese entsprechend aus der Drive-Table zu streichen.

### Die Disk-Parameter-Blöcke:

Ein Disk-Parameter-Block (DPB) gibt genaue Auskunft über die Art des verwendeten Disketten-Formats. Er enthält einerseits logischen Daten wie z.B. Anzahl der Directory-Einträge, Blockgröße usw., andererseits alle notwendigen Informationen über den physikalischen Aufbau des Formats z.B. Anzahl der Spuren, Größe der Sektoren, Aufzeichnungsdichte usw. Jedes Format, das erkannt werden soll, benötigt seinen eigenen Disk-Parameter-Block. Standardmäßig sind im CBIOS-3 zehn Blöcke vorhanden. Diese können natürlich modifiziert werden. Man kann eigene neue Blöcke hinzufügen, die bestehenden modifizieren oder nicht benötigte Formate streichen. Wieviele DPB's existieren muß bei der Marke 'DPBNR' angegeben werden.

Unter CBIOS-3 wurden die normalen DPB's etwas erweitert und werden mit XDPB (eXtended DPB) gekennzeichnet. Daher bestehen die Disk-Parameter-Blöcke aus zwei Teilen, dem BIOS- und dem BDOS-spezifischen Teil.

Der erste BIOS-spezifische Teil setzt sich aus folgenden Werten zusammen, die teilweise direkt mit der Programmierung des FDC-Controllers uPD 765 (s. Datenblatt) in Verbindung stehen:

1. NUM      0 falls 128 Bytes Sektorgröße, 1 bei 256 Byte, 2 bei 512 Bytes pro Sektor ...
2. MXSC     Sektoren pro Spur
3. GAP3     Länge der GAP3
4. DTL      -1 falls NUM > 0, sonst Bytes pro Sektor
5. SPEC     'Specify Parameter' für das Disk-Timing
6. TYPE     Type Flag
7. XLT      Adresse der Sektor-Übersetzungstabelle

Der zweite Teil ist der eigentliche Disk-Parameter-Block. Das

Macro 'dpb' generiert diesen automatisch aus den angegebenen Parametern.

Verändert man einen Disk-Parameter-Block, so muß man darauf achten, daß die Länge der zehn Blöcke nicht verändert wird. Am einfachsten ist man verändert nur die 'Equate'-Anweisungen am Kopf jedes Blocks.

#### **Format-Erkennung:**

Beim Login, d.h. beim Identifizieren des eingelegten Formats stehen dem BIOS die definierten (standard zehn) Disk-Parameter-Blöcke zur Verfügung. Das BIOS startet seinen Login mit der Marke DPB1. Daher muß diese Marke die erste der zehn DPB-Marken sein. Nun übernimmt das BIOS die Daten aus dem jeweiligen Block, programmiert den FDC-Controller entsprechend und startet auf bestimmten Positionen der Diskette Lese-Versuche. Sind diese fehlerfrei, so wird untersucht ob die Sektorzahl pro Track, die Anzahl der Seiten usw. mit den Angaben des jeweiligen Disk-Parameter-Blocks übereinstimmen.

Folgende Parameter dienen zum Unterscheiden der Disk-Formate:

- \* 8 Zoll / 5.25 Zoll
- \* Skip oder 'double stepping'
- \* Single sided / double sided
- \* FM oder MFM Aufzeichnung
- \* Reverse Seek
- \* Sektorgröße 128 - 1024 Bytes/Sektor 'SECT'
- \* Sektoren / Track 'MXSC'

Stimmen zwei Formate in diesen Größen überein, so können sie nicht unterschieden werden. Daher ist beim Anlegen neuer Formate darauf zu achten, daß nur ein Format mit genau diesen Parametern existiert. Logische Parameter wie z.B. Größe eines Blocks können natürlich nicht zur Unterscheidung der Formate dienen.

Verlaufen bei der Format-Ermittlung alle Tests positiv, so übernimmt das BIOS die Daten aus dem Disk-Parameter-Block und setzt sie als aktuelle Laufwerksparameter ein und gleicht den Disk-Parameter-Header für dieses Laufwerk entsprechend an. Diese Werte haben nun Gültigkeit, bis der Anwender durch Betätigen von '^C' einen Disk-Reset durchführt oder ein Schreib-Lese-Fehler auf der Diskette erkannt wird, der auf das Einlegen einer anders formatierten Diskette zurückzuführen ist.

Verläuft einer der oben genannten Tests negativ, so überspringt das BIOS den derzeitigen Disk-Parameter-Block und startet einen Login-Versuch beim nächsten Block. Verlaufen alle Format-Tests negativ, so leitet das BIOS automatisch eine Modifizierung der Formate ein:

#### **Automatische Type-Byte-Modifikation:**

Das BIOS verändert dazu in vier Durchgängen die ursprünglichen Werte des Type-Bytes im Disk-Parameter-Block, um dennoch das eingelegte Format zu erkennen. Die zwei Bits 5 (single/double-sided) und Bit 6 (skip/no-skip).

Schritt 1: Rücksetzen von Bit 5 und 6. Ausschalten der Skip-Option und zweiseitiger Diskettenzugriff.  
TYPE I: X00X.XXXX

Nun startet das BIOS wieder Login-Versuche entsprechend der Anzahl der DPB's, bis es das entsprechende Format findet. Falls erfolglos wird das Byte ein zweitesmal geändert.

Schritt 2: Einschalten der Skip-Option und zweiseitige Aufzeichnung.  
TYPE II: X10X.XXXX

Nun werden z.B. doppelseitige Formate von 40 Track-Laufwerken erkannt, die in 80 Track-Laufwerken eingelegt wurden, ohne dafür einen eigenen Disk-Parameter-Block definieren zu müssen.

Schritt 3: Ausschalten der Skip-Option und einseitige Aufzeichnung.  
TYPE III: X01X.XXXX

Hier können die ursprünglich für zweiseitige Aufzeichnung vorgesehenen Formate auch auf einseitig formatierte Disketten ange-

wendet werden, d.h. es werden jetzt alle einseitigen Diskettenformate erkannt.

Schritt 4:       Einschalten der Skip-Option und einseitige Aufzeichnung.  
                  TYPE IV: X11X.XXXX

Schritt 1 bis 4 wird natürlich für alle Disk-Parameter-Blöcke durchgeführt. Dies ergibt maximal 40 verschiedene Formate (Standard CBIOS-3 ohne Hinzufügen eigener DPB's) die das Standard-BIOS über diesen einfachen Trick erkennt. Verlaufen alle Schritte erfolglos, so setzt das BIOS eine Fehlermeldung ab und startet, falls der Bediener es wünscht, einen neue Login-Prozedur oder gibt den Fehler an das aufrufende Programm weiter.

#### **Die Sektor-Übersetzungs-Tabellen:**

Jedem Format kann eine eigene Übersetzungstabelle zugeordnet werden, die sich alle am Schluß des AUTODPBO-Files befinden. Das Aufstellen der Tabelle übernimmt das Macro 'skew' nach Angabe der Sektorzahl und dem Skew-Faktor ('interleaving faktor').

**Der USRROUT.ASM-File:**

Ist es notwendig bestimmte Routinen dem Bios zuzufügen oder abzuändern, so kann dies über diesen File geschehen. Die dortigen Einsprünge werden bei jedem Bios-Aufruf als erstes über einen 'CALL' angesprungen. Zu jeder Bios-Routine existiert ein eigener Sprung. Im Standard-Bios kehren dies Z-Aufrufe durch einen 'RET'-Befehl sofort wieder in die aufrufende Routine zurück, d.h. sie sind noch funktionslos. Näheres s. im USRROUT-File.

Im folgendem eine Liste von Marken, auf die sich Anwender-Routinen im USRROUT-File beziehen können:

βdtbl: Adresse der Bios-Drive-Tabelle, Wort  
 βdbnk: Derzeitige DMA-Bank, Byte  
 βadrv: Derzeitige Laufwerksnummer, Byte  
 βtrk: Derzeitige Spurnummer, Wort  
 βsect: Derzeitige Sektornummer, Wort  
 βdma: Derzeitige DMA-Adresse, Wort  
 βcbnk: Derzeitige Programm-Bank, Byte

vatab: Initilisierungstabelle für DART Teil A  
 vbtab: Initilisierungstabelle für DART Teil B

Folgende 8 Adressen sind Routinen die direkt auf die beiden RS-232 Schnittstellen der DART zugreifen:

vainst: Test Input-Status Dart A. Z = 0 falls bereit  
 vaoust: Test Output-Status Dart A. Z = 0 falls bereit  
 vain: Zeichen von Dart A in Register A holen  
 vaout: Zeichen in Register C an Dart A senden

vbinst,vboust,vbin,vbout: Wie oben für Kanal B

Folgende Adressen sind die Einsprungpunkte für die Standard-Bios-Funktionen:

?boot, ?wboot, ?const, ?conin, ?cono, ?list, ?auxo, ?auxi,  
 ?home, ?sldsk, ?sttrk, ?stsec, ?stdma, ?read, ?write,  
 ?lists, ?sctrn, ?conos, ?auxis, ?auxos, ?dvtbl, ?devin,  
 ?drtbl, ?mltio, ?flush, ?mov, ?tim, ?bnksl, ?stbnk, ?xmov