

---

# Teil 7

---

# Hard- und Softwareergänzungen

# 7

## Hard- und Software- Ergänzungen

### Inhalt

<b>7/2</b>	<b>Bauanleitungen</b>
7/2.3	RESET-Taster
7/2.4	Druckerschnittstelle mit 8 Bit



## Teil 7: Hard- und Software-Ergänzungen

Teil 7 ist ganz den Zusatzgeräten zum Rechner gewidmet, sofern sie nicht unter die Kategorie Floppy, Kassettenrecorder oder Drucker fallen. Zumeist sind irgendwelche Hardware-Ergänzungen auch mit entsprechenden Software-Ergänzungen verbunden, wobei die Software als Treiber dann für die Hardware fungiert.

Kapitel 7/1 bezieht sich auf freikäufliche Handelsware und Kapitel 7/2 auf Bauanleitungen, die im Rahmen dieses Werkes vorgestellt werden. Bei der Handelsware werden wir neben Maussteuerungen auch verschiedene Joystickformen vorstellen. Vielleicht schreiben Sie dem Verlag auch einmal einen kleinen Brief, in dem Sie Ihre Wünsche an den vorzustellenden Geräten darlegen.

Vielleicht haben Sie auch irgendwo eine Bauanleitung oder eine Hardware-Erweiterung gebastelt, die wir im Rahmen dieses Buches besprechen können.

## 7/2.3

# RESET-Taster

Autor: Andreas Kißlinger

Nun soll die Beschreibung des Einbaues eines RESET-Tasters in den CPC beschrieben werden. Dazu werden sich gleich viele fragen, wozu man so etwas braucht, man könne doch mit dem Drei-Finger-Griff (CTRL-SHIFT-ESC) einen RESET auslösen. Diese Ansicht ist im allgemeinen richtig, sie ist jedoch falsch, wenn man diesen RESET näher betrachtet. Der RESET durch die drei Finger wird über die Tastatur ausgelöst. Die Tastatur wird jedoch nur softwaremäßig abgefragt. Diese Abfrage geschieht jede fünfzigstel Sekunde. Was geschieht aber, wenn diese Abfrage ausfällt? Richtig, dann kann man keinen RESET mehr auslösen. Dazu ein Beispiel: Geben Sie einmal folgende drei Befehle ein:

```
POKE &100,&F3  
POKE &101,&76  
CALL &100
```

Sie werden merken, daß nach der Eingabe des letzten Befehls der Rechner nicht mit ‚Ready‘ antwortet, keine wirren Muster auf den Bildschirm schreibt und auch keinen RESET auslöst. Versuchen Sie mal mit CTRL-SHIFT-ESC einen RESET auszulösen! Dabei werden Sie feststellen, daß dies nicht geht, das liegt daran, daß mit dieser Befehlsfolge ein Maschinenprogramm mit zwei Byte Länge angesprochen wird. Dieses schaltet zuerst einmal die Unterbrechungen ab (Maschinenbefehl DI, Kode &F3), um dann den Prozessor anzuhalten (Maschinenbefehl HALT, Kode &76). Dieser Haltebefehl kann nur aufgehoben werden, wenn eine Unterbrechungsanforderung an den Prozessor gestellt wird (INT- oder NMI-Signal), wenn ein RESET erfolgt oder natürlich, wenn man den Strom abstellt. Das NMI-Signal wird beim CPC nicht genutzt und die INT-Unterbrechungen wurden mit DI abgestellt. Fazit, der Prozessor wartet, bis der Strom abgestellt wird. Nicht so, wenn eine RESET-Taste vorhanden ist, die den Prozessor hardwaremäßig zurücksetzt (dabei werden auch die anderen Bausteine im CPC zurückgesetzt).

Jetzt werden sich viele von Ihnen fragen, wo denn der Vorteil gegenüber dem Ausschalten des Rechners liegt, wenn doch der Speicher beim Einschalten oder beim RESET immer gelöscht wird und dabei wichtige Daten verlorengehen können. Es gibt mehrere Vorteile, zum einen wird der Rechner nicht so rabiat abgestellt (die Schaltkreise lassen grüßen), außerdem wird beim RESET mit CTRL-SHIFT-ESC kein RESET der Peripherie ausgelöst. Dieses Fehlen eines Peripherie-RESET kann z. B. dazu führen, daß der

## 2.3 RESET-Taster

Teil 7: Hard- und Software-Ergänzungen

Floppycontroller, wenn der Rechner abgestürzt ist, ein weiteres Kommando vom Betriebssystem benötigt und sich das Betriebssystem daher bei der Initialisierungsroutine des Floppy-ROMs aufhängt. Dieser Zustand kann nur mittels des Hardware-RESET (sprich Ausschalten des Rechners) aufgehoben werden. Erst danach sind wieder sinnvolle Aktionen vom Rechner zu erwarten.

Ein weiterer Vorteil eines Hardware-RESETs gegenüber dem Ausschalten ist der Umstand, daß eine eventuell vorhandene Speichererweiterung nicht gelöscht wird. Für die Besitzer einer Speichererweiterung mit RAM-Disk ist dies besonders wichtig. Sollte sich ein Programm so aufhängen, daß man es mit einem Drei-Finger-Griff nicht mehr abrechnen und wertvollen Speicherinhalt der RAM-Disk retten kann, so kann man mit einer RESET-Taste immer abrechnen und nach dem erneuten Einbinden der RAM-Disk in das Betriebssystem den Inhalt wieder benutzen. Voraussetzung dabei ist natürlich, daß die RAM-Disk nicht automatisch beim Einbinden gelöscht wird.

Der Einbau eines RESET-Tasters in den CPC ist ziemlich einfach. Etwas verwirrend sind lediglich im Benutzerhandbuch die Bezeichnungen RESET und BUS RESET am Systembus. Denn nicht der RESET-Anschluß ist der benötigte Kontakt, sondern der BUS RESET. Der RESET-Anschluß ist nämlich ein Ausgang, der zum Zurücksetzen der angeschlossenen Peripherie (z. B. Floppy-Disc-Controller) vorhanden ist. Aus dem Handbuch erfährt der versierte Leser weiterhin, daß der Anschluß zum Aktivieren des RESETs auf logisch 0 (Masse) gelegt werden muß. Diese Aussage kann man an Hand des Querstrichs über der Signalbezeichnung herleiten. Die Masseleitungen am Systembus sind mit GND für Ground bezeichnet. Man muß für einen RESET also eine leitende Verbindung zwischen dem Anschluß 40 (BUS RESET) und 49 bzw. 2 (beide GND) herstellen. Nichts anderes soll unser RESET-Taster machen.

Zum Einbau des RESET-Tasters benötigt man:

<p>1 kleinen Ein-Taster 1 Stück isolierten Schaltdraht 1 LötKolben und Elektroniklot 1 Schraubenzieher Kreuzschlitz Größe 1 1 Messer zum Abschaben von Leiterbahnen 1 Seitenschneider zum Kürzen von Drähten und etwas Löterfahrung sowie die Akzeptanz, daß mit dem Eingriff die Garantie auf den Rechner erlischt.</p>
--

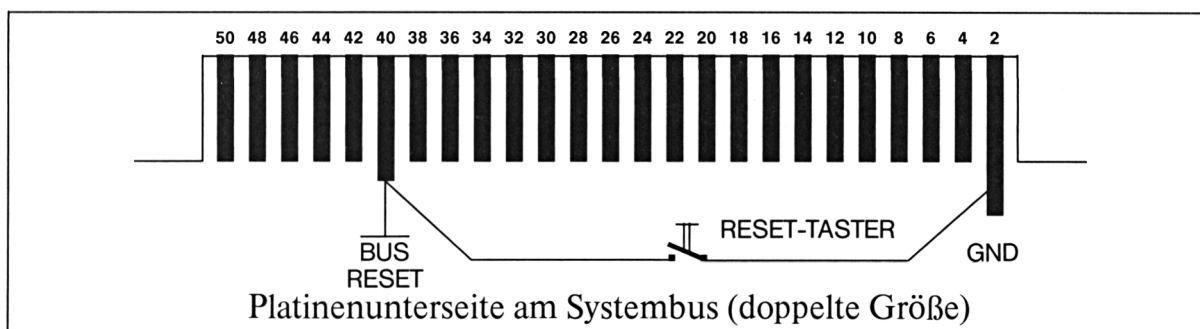
Wenn man selbst keinerlei Erfahrung im Umgang mit dem LötKolben hat, so sollte besser ein Freund mit entsprechenden Kenntnissen den Eingriff übernehmen. Auf keinen Fall sollte man normales Haushaltslot mit Lötfett als Flußmittel nehmen, da das Lötfett die Oberfläche der Bauelemente angreift und damit letztendlich nur für schlechtere Kontakteigenschaften sorgt, die jedoch erst nach längerer Zeit auftreten können.

### 2.3 RESET-Taster

Teil 7: Hard- und Software-Ergänzungen

Zum Eingriff selbst ist zu sagen, daß man zuerst den Rechner aufschrauben sollte. Dazu sind die sechs Kreuzschlitzschrauben an der Unterseite des Rechners (beim CPC 664 und 6128 auch noch zwei an der Unterseite des Floppylaufwerks) zu lösen und die Gehäuseoberseite (Tastatur) vom Rechner abzunehmen. Die Tastatur ist mit einem Kabel und einem Stecker mit der Rechnerplatine verbunden. Zum besseren Arbeiten am CPC 464 kann dieser Anschluß durch Abziehen des Steckers getrennt werden, bei den Versionen CPC 664 und CPC 6128 kann der Gehäusedeckel einfach weggeklappt werden, dabei darf man allerdings die Leiterfolie des Tastaturanschlusses nicht beschädigen. Beim CPC 464 ist noch ein weiterer Stecker vorhanden, der das Kassettenlaufwerk, die Leuchtdiode und den Ein-/Ausschalter mit der Platine verbindet. Auch diesen sollte man zum besseren Arbeiten entfernen. Außerdem sollte nun das Abschirmblech (nur in den neueren Computern vorhanden) entfernt werden, so daß man an die Hauptplatine herankommt. Die Hauptplatine muß nun ebenfalls gelöst werden, damit man an die Unterseite mit den Anschlüssen 2 bzw. 40 herankommt. Dazu sind ebenfalls sechs Kreuzschlitzschrauben zu lösen. Nach dem Herausnehmen der Hauptplatine kann man nun eine geeignete Einbaustelle für den RESET-Taster suchen, so daß der Taster geschützt liegt und den Zusammenbau bzw. den Betrieb des Rechners nicht stört. Von diesem Taster aus muß nun ein Draht an den Anschluß 40 des Systembus gelötet werden. Dazu ist entweder der Lötstoplack an diesem Anschluß vorsichtig etwas abzuschaben und der Draht dort anzulöten, oder aber die von dem Anschluß wegführende Leiterbahn ein Stück verfolgen und, sofern ein Lötauge zur Durchkontaktierung der Leiterbahn vorhanden ist, den Draht direkt durch ein Lötauge zu löten. Diese beiden Möglichkeiten hat man auch, wenn man den anderen Anschluß des Tasters mit einem der Systembusanschlüsse 2 oder 49 verbinden muß.

Als Anhaltspunkt soll die folgende schematische Zeichnung des Systembusses mit den anzuschließenden Kontakten und den wegführenden Leiterbahnen dienen (Bild 7/2.3-1):

**Bild 7/2.3-1**

Nach dem Einbau des Tasters und der Verbindung des Tasters mit dem BUS RESET und einem Masseanschluß kann der Einbau der Hauptplatine erfolgen. Dazu muß die

**2.3 RESET-Taster**

Teil 7: Hard- und Software-Ergänzungen

Abschirmung, sofern eine vorhanden war, wieder aufgesetzt werden. Dabei ist darauf zu achten, daß die Anschlußdrähte des RESET-Tasters nicht durch ein scharfkantiges Blech kurzgeschlossen werden. Nach dem Einbau der Abschirmung und der Hauptplatine können nun die Anschlüsse mit dem Gehäuseoberteil wieder hergestellt und damit die Tastatur und beim CPC 464 der Kassettenrecorder wieder angeschlossen werden. Danach wird das Gehäuse wieder zusammengesetzt und zugeschraubt.

Damit ist der Eingriff abgeschlossen, und Sie können die Funktionsweise des RESET-Tasters nach dem Einschalten des Rechners überprüfen.

Sollte Ihr Rechner nach dem Einschalten keinerlei Aktionen durchführen, so kann es sein, daß Sie einen Kurzschluß des BUS RESET mit dem Masseanschluß produziert haben oder aber die Stromversorgung nicht wieder richtig angesteckt ist; dies kann vor allem beim CPC 464 der Fall sein. Da der Eingriff aber relativ harmlos ist und kein Fall zu Ohren gekommen ist, daß der Rechner nach dem Eingriff nicht mehr funktioniert hat, sollte auch bei Ihnen der Einbau eines RESET-Tasters kein Problem sein. Das zu Beginn dieser Einbauanleitung gebrachte Beispiel der Blockade eines Rechners können Sie nun noch einmal ausprobieren. Nach dem CALL-Befehl muß der Rechner nun wieder hängen. Durch Betätigen der eingebauten RESET-Taste muß allerdings nun ein RESET ausgelöst werden und der CPC wieder die Einschaltmeldung ausgeben. Sollte dies nicht der Fall sein, so haben Sie einen Einbaufehler gemacht, der den BUS-RESET-Anschluß bei Betätigen des Tasters nicht auf Masse legt.

Mit dieser Bauanleitung für einen RESET-Taster können Sie nun auch z. B. aus Spielen aussteigen, welche den Drei-Finger-RESET blockieren. Diese Blockade kann übrigens sehr einfach durch folgende POKE erfolgen:

```
POKE &BDEE,&C9
```

Wenn Sie diesen Befehl z. B. in ein BASIC-Programm einfügen, so kann danach nur noch jemand mit einem RESET-Taster dieses Programm vollkommen abbrechen, ohne den Computer auszuschalten.

## 7/2.4

# Druckerschnittstelle mit 8 Bit

**Autor: Andreas Kißlinger**

Hier soll nun die Beschreibung vom Einbau und der softwaremäßigen Einbindung einer acht Bit breiten Druckerschnittstelle erfolgen. Manche von Ihnen werden sich nun fragen, wozu eine 8-Bit-Schnittstelle nötig sei, man könne doch alles mit dem bestehenden Schneider-Druckerport ausgeben. Für die meisten Anwendungsfälle ist hier beizupflichten, es gibt jedoch Anwendungen, die mit einer sieben Bit breiten Schnittstelle nur umständlich oder gar nicht zu realisieren sind. Solche Anwendungen sind z. B. die Ansteuerung eines zweiten Druckerzeichensatzes mit dem achten Bit, ein Bildschirmgrafikausdruck (Hardcopy), das Laden eines selbstdefinierten Zeichensatzes in den Drucker, die Ausgabe der Schneidergrafikzeichen 128 bis 255 und die Nutzung gewerblicher Programme für die Grafikausgabe von Zeichen und Entwürfen, welche, wie z. B. FANCY FONT (ein CP/M-Programm zur Ausgabe verschiedener Zeichensätze in verschiedenen Größen), eine acht Bit breite Druckerschnittstelle erfordern.

Zuerst soll nun der Einbau bzw. die nötige Hardwareänderung des Schneider-Computers beschrieben werden. Danach werden noch kurze Programme (teilweise Maschinenroutinen) vorgestellt, welche eine softwaremäßige Einbindung der Schnittstelle ins bestehende Betriebssystem darstellen.

Jetzt also zuerst die Beschreibung der Hardwareänderung:

Diese besteht im wesentlichen aus einem kurzen Draht, der das Datenbit des Kassettenrecorders auf das achte Bit für den Druckeranschluß umlenkt. Dabei treten keine gegenseitigen Beeinflussungen auf, da niemals Drucker und Kassettenlaufwerk zugleich bedient werden. Die Anzahl der benötigten Bauteile für diese Umrüstung ist sehr gering. Man braucht nämlich dazu nur:

- 1 Stück isolierten Schaltdraht
- 1 Lötkolben und Elektroniklot
- 1 Schraubenzieher Kreuzschlitz Größe 1
- 1 Messer zum Abschaben und Trennen von Leiterbahnen
- 1 Seitenschneider zum Kürzen von Drähten  
und etwas Löterfahrung

Man sollte sich auf jeden Fall klarmachen, daß mit einem Eingriff in den Computer dessen Garantie erlischt. Auf keinen Fall sollte man normales Haushaltslot mit Löt fett als

**2.4 Druckerschnittstelle mit 8 Bit**

Teil 7: Hard- und Software-Ergänzungen

Flußmittel nehmen, das das Lötfett die Metalloberfläche der Bauelemente angreift und damit letztendlich nur für schlechtere Kontakteigenschaften sorgt, die jedoch erst nach längerer Zeit auftreten können.

Bevor jedoch ein Eingriff in den (ausgeschalteten) Rechner gemacht werden kann, muß der Rechner geöffnet und die Platine freigelegt werden. Dazu müssen die sechs Kreuzschlitzschrauben an der Unterseite des Rechners gelöst werden. Gegebenenfalls sind beim CPC 664 und beim CPC 6128 auch noch die beiden Schrauben an der Unterseite des Floppylaufwerks zu entfernen. Danach kann man die Gehäuseoberseite, also die Tastatur, entfernen. Dieses Gehäuseteil ist, da es die Tastatur enthält, noch mit dem Unterteil, welches die Platine enthält, mit Kabeln verbunden. Diese müssen beim CPC 464, damit man besser arbeiten kann, mit den Steckverbindern abgezogen werden. Bei den anderen beiden CPC-Versionen kann das Gehäuseoberteil einfach zur Seite weggeklappt werden, wobei man aber auf die knickempfindliche Leiterfolie des Tastaturanschlusses achtgeben sollte. Danach kann man das Abschirmblech (welches nur bei den neueren Geräten vorhanden ist) entfernen. Die Platine liegt dann frei vor Ihnen. Jetzt kann mit der eigentlichen Arbeit begonnen werden.

Dazu muß zuerst der neunte Anschluß des Druckerports, der an die Systemmasse angeschlossen ist, von der Masse getrennt werden. Das kann mit einem scharfen Messer geschehen, indem die Masseleitung an Anschluß neun durchgeschabt wird. Der jetzt freie Anschluß zum Drucker muß mit dem zwölften Anschluß des Parallel-Ein-/Ausgabe-Bausteins 8255 verbunden werden. Dazu lötet man vorsichtig einen dünnen Draht an den zwölften Pin des 8255, man kann allerdings nachsehen, ob die Leiterbahn des 12. Anschlusses mit einem Lötauge von der Platinenvorder- zur Platinenrückseite wechselt. Ist dies der Fall, so kann man den Draht direkt durch das Lötauge löten und vermeidet so den direkten Lötkontakt mit dem Ein-/Ausgabe-Baustein.

Danach schabt man am neunten Anschluß des Druckerports den grünen Lötstoplack weg, so daß man die blanke Leiterbahn vor sich liegen hat. Ist das der Fall, so kann man den Draht dort anlöten. Die Unterbrechung des 9. Druckeranschlusses und die zwei Lötstellen sind also die einzigen echten hardwareseitigen Änderungen an Ihrem Schneider-CPC. Nach der Änderung des Druckeranschlusses muß die Abschirmung wieder eingebaut, die Tastatur wieder angesteckt und der Computer wieder zugeschraubt werden. Für die Angaben von Anschlüssen auf der Platine bzw. an dem Portbaustein 8255 kann die Skizze, Bild 7/2.4-1, hilfreich sein.

Sicher werden manche nun annehmen, daß mit der Hardwareänderung schon alles gelaufen ist und man nur den Rechner und den Drucker einzuschalten braucht, damit man eine acht Bit breite Schnittstelle für den Drucker hat. Rein prinzipiell stimmt das, jedoch hat die Geschichte einen Haken. Man muß dem Betriebssystem nämlich erst einmal klarmachen, daß nun eine acht Bit breite Schnittstelle vorhanden ist. Und weiterhin, da das achte Bit mit dem Kassettendatenbit simuliert wird, das Datenbit der zusätzlichen Leitung entsprechend aktivieren.



## 2.4 Druckerschnittstelle mit 8 Bit

Teil 7: Hard- und Software-Ergänzungen

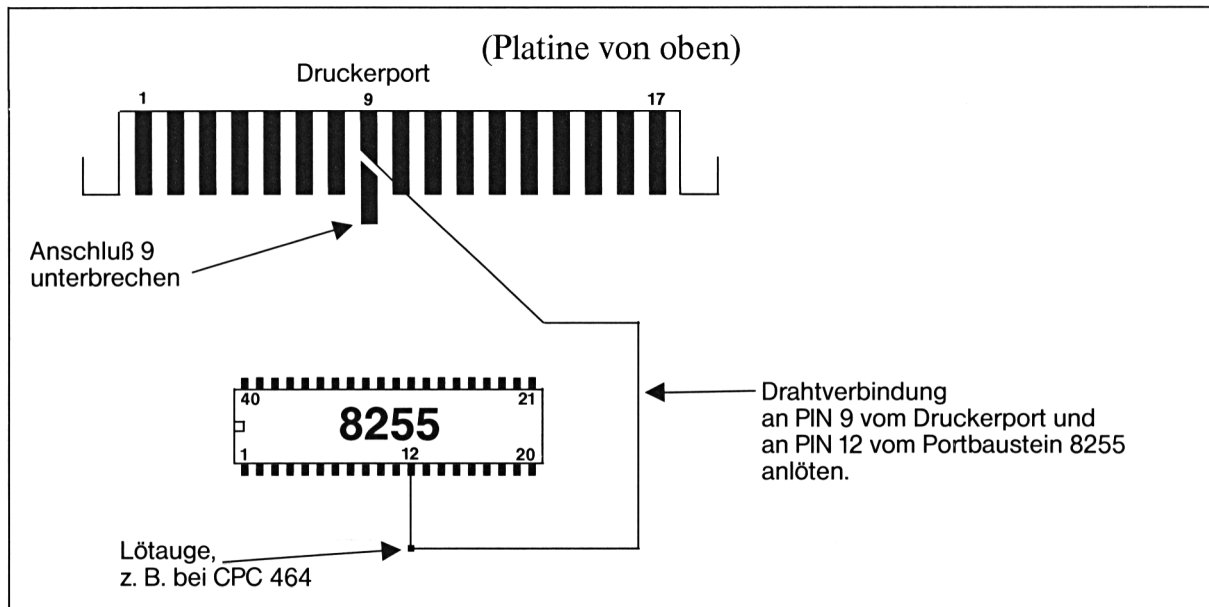


Bild 7/2.4-1

Dazu ist ein Treiberprogramm, ein sogenannter Patch, notwendig, das an geeigneter Stelle im Betriebssystem einen oder mehrere Vektoren auf eine kurze, selbstgeschriebene Routine umbiegt und damit das achte Bit in die normale Druckerausgabe einbindet. Die Initialisierung des Treibers, also der Aufruf des Patchprogramms, muß immer erfolgen, wenn das Betriebssystem seine Vektoren neu hergestellt hat. Die Einbindung muß daher einmal am Anfang nach der CP/M-Initialisierung und nach dem BASIC-Start erfolgen. Der Treiber bleibt bis zu einem RESET, dem Ausschalten, einem CP/M-Neustart oder einem BASIC-Neustart vorhanden und muß in der Zwischenzeit nicht neu eingebunden werden. Die Einbindung bleibt so lange vorhanden, weil erst dann die Betriebssystemroutine mit der Adresse &BD37 aufgerufen wird. Diese stellt nämlich fast die komplette Sprungtabelle neu her.

Eine mehrfache Einbindung kann man mit den hier vorgestellten Routinen zwar versuchen, die Programme erkennen allerdings selbständig, ob der Treiber bereits installiert ist und lassen eine mehrfache Einbindung nicht zu, da diese zu einer Endlosschleife bei der Druckerausgabe und damit zum Absturz des Rechners führen könnte.

Wenn sich jemand nicht um die Einbindung eines Treibers kümmern will, so ist es ihm freigestellt, eine Änderung im Betriebssystem durchzuführen. Dann muß allerdings diese Änderung mit dem restlichen Betriebssystem in ein EPROM oder PROM gebrannt werden, das ist jedoch nur für die Besitzer eines EPROM-Programmiergerätes interessant. Hier soll jedoch der für jeden machbare Weg gezeigt werden.



**2.4 Druckerschnittstelle mit 8 Bit**

Teil 7: Hard- und Software-Ergänzungen

Die Einbindung eines Treibers kann auf verschiedene Arten erfolgen. Zum einen kann man mit festen ROM-Einsprünge operieren, dadurch wird man aber in höchstem Maße inkompatibel zwischen den verschiedenen CPC-Versionen, da diese verschiedene Betriebssystem-ROMs enthalten. Dadurch benötigt man sieben verschiedene Treiber, je einen für jede CPC-Version unter BASIC und unter CP/M und einen für CP/M+. Andererseits kann man den alten ROM-Einsprung an den Schluß des Treibers kopieren und dann die Adresse der Treiberoutine als Zieladresse eines Sprungbefehls an dem alten Betriebssystemeinsprung eintragen. Außerdem sind die CPC-Geräte teilweise in der Hardware kompatibler als in der Software. Dadurch erreicht man eine Unabhängigkeit von der verwendeten Computerversion und benötigt daher nur zwei Treiberversionen, eine für BASIC (bzw. für die Maschinenprogrammierung unter dem CPC-Betriebssystem) und eine für die Installation des Treibers unter CP/M. Für unsere 8-Bit-Schnittstelle wurde logischerweise die zweite Möglichkeit gewählt. Die Treiberoutine gilt nicht für CP/M+. Andererseits erkennt das Maschinenprogramm, ob eine Speichererweiterung von Vortex vorhanden ist und schaltet die Speicherbänke entsprechend so um, daß der Treiber in das System eingebunden werden kann. Leider mußte dadurch der Treiber verlängert werden, da die Firma Vortex, um wenigstens etwas anders als AMSTRAD/Schneider zu arbeiten, einen anderen Betriebssystemeinsprung benutzt, welcher deswegen ebenfalls gepatcht werden muß.

Die eigentliche Treiberoutine kommt im Gegensatz zu den bisher in Zeitschriften veröffentlichten Routinen ganz ohne Sprünge und Unterprogramme aus, da der Ein-/Ausgabe-Baustein einen Einzelbitmodus für den Port C besitzt, welcher hervorragend für diesen Einsatzzweck geeignet ist, da das Kassettendatenbit ebenfalls über den Port C verwaltet wird. Das Bit, welches gesetzt bzw. zurückgesetzt werden muß, ist das Bit 5 des Portes C. Die Bits des Portes C können über den Steuerport des Bausteines gesteuert werden. Die Datenworte, die dazu an den Baustein zu senden sind, müssen im Bit 0 den zu setzenden Wert haben und in den Bits 1 bis 3 die Nummer des zu setzenden Bits. Die Bitnummer kann bei binärer Kodierung durch drei Bits zwischen 0 und 7 liegen.

Durch Verwendung des Einzelbitmodus von der PIO 8255 (Parallel Input Output) und der Tatsache, daß die Kassettensoftware das Datenbit immer in der benötigten Weise setzt (deswegen muß der vorherige Portzustand nicht wiederhergestellt werden) wird die Routine für das CPC-Betriebssystem (auch für BASIC) sehr kurz (10 Byte + 3 Byte für den alten Einsprung). Die bisherigen Routinen lagen in der Größenordnung von 30 bis 60 Byte und benötigten daher relativ viel Speicher im zur Verfügung stehenden Patchbereich. Für CP/M ist der Treiber etwas länger ( $13 + 2 \cdot 6 = 25$  Byte), da er zwei Systemeinsprünge versorgen muß. Auch sind unter CP/M zwei Unterprogrammaufrufe notwendig, da die Treiberoutine für zwei unterschiedliche Betriebssystemroutinen benötigt wird.

Der eigentliche Treiber geht zur korrekten Ausgabe den folgenden Weg:

**2.4 Druckerschnittstelle mit 8 Bit**

Teil 7: Hard- und Software-Ergänzungen

1. Retten des auszugebenden Zeichens auf den Stapel
2. Das achte Bit des Zeichens wird ins Carryflag rotiert
3. Das Register A wird mit 5, also der Bitnummer, geladen
4. Das Carryflag und der Akkuinhalt wird um eine Stelle nach links rotiert, wodurch die eingeschriebene 5 an die Stellen 1 bis 3 kommt.
5. Ausgabe des 8-Bit-Wortes an den Steuerport des Bausteines. Das achte Bit des Druckerports wird dadurch entsprechend gesetzt.
6. Das originale Zeichen wird wieder vom Stapel geholt und an die zuvor installierte (Betriebssystem-)Routine weitergegeben. Damit ist die Funktion der Treiberroutine beendet.

Wegen des zusätzlichen Patchens von einem zweiten Einsprung im CP/M (bedingt durch das Vortex eigene RAM-Bios \$OSC.SYS) muß im CP/M-Treiber am Anfang der Routine das BC-Registerpaar auf den Stapel gesichert und vor dem Einsprung ins Betriebssystem wieder vom Stapel geholt werden.

Der Treiber für das BASIC patcht nur eine Routine, für dessen Initialisierung genügt ein kurzes BASIC-Programm. Dieses überprüft ebenfalls, ob der Treiber bereits installiert ist und bricht ab, wenn der Treiber bereits im RAM an dieser Adresse vorhanden ist. Der BASIC-Treiber hat den Vorteil, daß er an eine gewünschte Adresse gelegt werden kann. Dazu muß nur die BASIC-Variable ‚startadresse‘ entsprechend belegt werden. Dies kann automatisch durch

MEMORY HIMEM-13 : startadresse = HIMEM+1

geschehen.

Die Grundroutine, auf die beide Programme aufbauen, wurde in ihrer Funktion schon zwei Absätze weiter oben beschrieben. Hier soll nun die zugehörige Maschinenroutine gezeigt werden:

```

PUSH AF      ; Retten des auszugebenden Zeichens
RLCA        ; Bit 7 vom Akku ins Carry-Flag
LD B,0F7H   ; Steuerport von PIO in B laden
LD A,5      ; Bit 5 soll beeinflusst werden
RLA         ; Akkuinhalt ein Bit nach links und
           ; Carry-Flag ins Bit 0
OUT (C),A   ; Bit 5 Port C := Bit 7 des Zeichens
POP AF      ; auszugebendes Zeichen holen
           ; Hier steht der alte Betriebssystemeinsprung

```

**2.4 Druckerschnittstelle mit 8 Bit**

Teil 7: Hard- und Software-Ergänzungen

Die Kodierung dieser Maschinenbefehle lautet (als Sedezimalzahlen):

0F5h,07h,06h,0F7h,3Eh,05h,17h,0EDh,079h,0F1h

Vor dem ersten Befehl (PUSH AF) und nach dem letzten Befehl (POP AF) kommen bei der CP/M-Version noch die Befehle PUSH BC (0C5h) bzw. POP BC (0C1h), da bei der Änderung des zweiten Betriebssystemeinsprungs der Inhalt des BC-Registerpaars konserviert werden muß.

Das folgende Assemblerlisting stellt die CP/M-Version des Treibers dar und kann abgetippt und mit fast jedem Z80-Assembler assembliert und in eine COM-Datei umgewandelt werden. Das Assemblerprogramm ist wie folgt aufgebaut:

Zuerst wird überprüft, ob das Programm unter CP/M 2.2 gestartet wurde. Wenn dies nicht der Fall war (es also unter CP/M Plus läuft), so wird eine Meldung ausgegeben, daß dieses Programm nicht unter CP/M Plus läuft. Danach wird zum Betriebssystem zurückgekehrt. Wenn das Programm unter CP/M 2.2 läuft, wird die Meldung ausgegeben, daß nun der Treiber installiert wird.

Danach wird, nach dem Sperren von Unterbrechungen, auf den zweiten Registersatz umgeschaltet und das DE-Register mit der Adresse des Warm-Boot Einsprungs geladen und der Zähler B auf 2 gesetzt.

Innerhalb einer Schleife wird der Wert im DE-Register daraufhin überprüft, ob er größer als der BIOS-Anfang im 62K CP/M ist. Wenn dies der Fall ist, dann wird die Schleife durch einen Sprung aus der Schleife heraus abgebrochen und das BC-Register mit der Speicherkonfiguration &7FEE geladen. Das bewirkt, daß eine Bankumschaltung ausgeführt werden kann und danach wieder die CP/M-Bank aktiviert werden kann.

Sollte der Wert in DE nicht größer als der BIOS-Anfang gewesen sein, so wird von der Folgeadresse dieses Wertes ein weiterer 16-Bit-Wert ins DE-Register geholt (nächste Sprungadresse in einer möglichen Kette) und mit diesem Wert an den Anfang der Schleife gesprungen. Wurde auch beim zweiten Durchlauf keine höhere Adresse als der BIOS-Anfang gefunden, so wird die Speicherkonfiguration im BC-Register auf &7F86 gesetzt, ein Wert, der für die Verwendung unter einem 44K CP/M ohne Speichererweiterung nötig ist. Der Test auf den BIOS-Anfang in den Sprungadressen kann erfolgen, weil das normale BIOS seine ganze Routine unterhalb der Adresse &E000 stehen hat, und auch das BIOS des Vortex-Laufwerks den Warm-Boot Einsprung zwischen den Adressen &C000 und &DFFF besitzt. Der Test erfolgt zweimal, da man auch ein 44K CP/M mit einer Speichererweiterung starten kann und dann allerdings die BIOS-Einsprünge an Adressen stehen, wie sie es normalerweise auch tun. Durch den Test der weiterleitenden Adressen kann so überprüft werden, ob der Einsprung nur umgeleitet wurde oder ob er tatsächlich nicht in das BIOS im RAM einspringt.

**2.4 Druckerschnittstelle mit 8 Bit**

Teil 7: Hard- und Software-Ergänzungen

Mit dem korrekt gesetzten BC-Register wird nun die aktuelle Speicherkonfiguration in Register E zwischengespeichert und eine eventuell vorhandene Speichererweiterung korrekt ausgeblendet. Als letzte Operation dieses Teils wird dann wieder auf den normalen Registersatz zurückgeschaltet.

Dann wird überprüft, ob an dem Druckereinsprung des Betriebssystems schon die Adresse des Treibers als Sprungadresse steht und, wenn das der Fall ist, zum Ende der Installationsroutine gesprungen. Sollte der Treiber noch nicht eingebunden sein, so werden nun die Maschinenbefehle und Adressen der beiden zu patchenden Routinen in die Treiberoutine eingetragen (jeweils 3 Byte) und der Treiber zu seiner Zieladresse &BE00 verschoben. Nun können die beiden Betriebssystemroutinen gepatcht werden, d. h. es wird als erstes von drei Byte der Kode &C3 (Maschinenbefehl JP) eingetragen und dann die jeweilige Treiberadresse dahinter geschrieben.

Zum Schluß (hier springt auch die Routine hin, wenn der Treiber schon installiert war) wird nochmals der zweite Registersatz gewählt, die originale Speicherkonfiguration wieder ins C-Register geladen und, falls nötig, eine eventuell vorhandene Speichererweiterung aktiviert. Danach werden die Unterbrechungen wieder zugelassen und mit einem RET-Befehl ins CP/M zurückgekehrt.

Nach dieser Routine kommt der eigentliche Treiber, welcher später an seine Zieladresse verschoben wird, und danach die zwei Meldungstexte, welche auf dem Bildschirm erscheinen können.

```
title    8-Bit am Drucker unter CP/M 2.2
;
;       Copyright 30.11.1986 by Andreas Kießlinger
;
; Zum Anwenden muß diese Datei assembliert und in eine
; COM-Datei umgewandelt werden. Diese darf nur aufgerufen
; werden, wenn das BC'-Register der Z80-CPU noch nicht
; durch ein Programm geändert wurde, also am besten nach
; dem Booten des CP/M. Beim Aufruf wird die Drucker-In-
; direction IND WAIT PRINTER und die Drucker-Ausgabe MC
; SEND PRINTER so gepatcht, daß sowohl mit der Vortex-
; Speichererweiterung unter 62K-CP/M, als auch unter AMSDOS
; eine 8 Bit breite Centronics-Schnittstelle aktiviert wird.
; Der Patch erfolgt nur, wenn bei MC SEND PRINTER noch kein
; entsprechender Eintrag vorhanden ist.

base     equ     0000h
wboot   equ     base
bdos     equ     base+5
tpa      equ     base+100h

strout   equ     9
cpmver   equ     12

cr       equ     0dh
lf       equ     0ah
strende  equ     '$'
```

## 2.4 Druckerschnittstelle mit 8 Bit

## Teil 7: Hard- und Software-Ergänzungen

```

mcsendprinter equ 0bd31h ; Adresse von MC SEND PRINTER
indwaitprinter equ 0bdf1h ; Adresse von IND WAIT PRINTER
bankbios equ 0f400h ; 62K-BIOS hat höhere Adresse
patchadr equ 0be00h ; Adresse zur Ablage der Routine
pioctrlport equ 0f7h ; Steuerport der PIO
cpm44k_bc2 equ 7f86h ; Speicherkonfiguration für 44K CP/M
cpm62k_bc2 equ 7feeh ; Speicherkonfiguration für 62K CP/M

        org tpa ; Das Programm fängt bei 100h an

start: ld c, cpmver ; zum Holen der CP/M Version
       call bdos ; CP/M Version erfragen
       ld a, 1 ; CP/M Version in Akku laden
       cp 22h ; ist es CP/M 2.2?
       jr z, cpm22ok ; wenn CP/M 2.2, dann weiter
       ld de, nocpm30 ; sonst String 'Nicht 3.0 CP/M'
       ld c, strout ; zur Strings Ausgabe vorbereiten
       call bdos ; Meldung ausgeben
       ret ; Rückkehr zum CCP

cpm22ok: ; CP/M Version war 2.2 => Treiber einbinden
       ld de, mesg1 ; Normale Meldung adressieren
       ld c, strout ; zum Meldungen ausgeben
       call bdos ; und ausgeben
       di ; wegen Bankumschaltung Interrupts sperren
       exx ; zweiten Registersatz wählen
       ld de, (wboot+1) ; Warmbootadresse in DE holen
       ld b, 2 ; maximal zweimal weitergeleitet
checklp: ; Schleife zur Prüfung ob 62K CP/M
       ld hl, bankbios-1 ; unter dem Anfang des Bank-BIOS
       or a ; Carry löschen
       sbc hl, de ; ist die Zieladresse größer als 62K?
       jr c, is62kcpm ; dann BC' für Bank-BIOS
       ex de, hl ; sonst wieder Adresse in HL
       inc hl ; Adresse eines Sprungs adressieren
       ld e, (hl) ; Lowbyte holen
       inc hl ; Highbyte adressieren
       ld d, (hl) ; Highbyte holen
       djnz checklp ; nächste Sprungadresse
       ld bc, cpm44k_bc2 ; 62K nicht gefunden => BC' für 44K CP/M
       jr dopatch ; zur Patchroutine
is62kcpm: ; 62K gefunden => BC' für 62K CP/M
       ld bc, cpm62k_bc2 ; BC' besetzen
dopatch: ; mit korrektem BC' weitermachen
       ld e, c ; ROM-Konfiguration in E' speichern
       res 5, c ; wenn nötig RAM-Erweiterung
       out (C), c ; mittels Portausgabe abschalten
       exx ; normalen Registersatz benutzen
       ld hl, (mcsendprinter+1) ; Adresse in Ausgaberroutine
       ld de, -patchadr ; negative Patchadresse
       add hl, de ; HL = -DE ==> HL = 0
       ld a, h ; HL=0 ==> Patch ist
       or l ; bereits installiert,
       jr z, ende ; dann nicht mehr patchen!

       ld hl, (mcsendprinter+1) ; MC SEND PRINTER
       ld (mcprtende+1), hl ; Adresse speichern;
       ld a, (mcsendprinter) ; ebenso den
       ld (mcprtende), a ; Opcode davor

       ld hl, (indwaitprinter+1) ; IND WAIT PRINTER
       ld (prtindende+1), hl ; Adresse speichern;
       ld a, (indwaitprinter) ; ebenso den

```

## 2.4 Druckerschnittstelle mit 8 Bit

## Teil 7: Hard- und Software-Ergänzungen

```

        ld      (prtindende),a      ; Opcode davor

        ld      hl,bit8_patch      ; sonst Patchroutine adressieren
        ld      de,patchadr        ; Zieladresse festlegen
        ld      bc,patchende-bit8_patch ; und Patchlänge holen
        ldir                      ; Routine installieren

        ld      a,0c3h             ; JP in Akku
        ld      (mcsendprinter),a  ; MC SEND PRINTER und
        ld      (indwaitprinter),a ; IND WAIT PRINTER patchen

        ld      hl,patchadr+0      ; Adresse des neuen
        ld      (mcsendprinter+1),hl ; MC SEND PRINTER

        ld      hl,patchadr+6      ; Adresse der neuen
        ld      (indwaitprinter+1),hl ; Indirection

ende:    exx                      ; zweiten Registersatz wählen
        ld      c,e                ; ROM-Konfiguration wieder in
        out     (c),c             ; C holen und Konfiguration
        exx                      ; selektieren, normalen Register-
        ret                       ; satz wählen und Rückkehr ins CP/M

bit8_patch: ; Beginn der Treiberroutine
sendprtpatch: ; Patch für MC SEND PRINTER
        call   patchadr+12        ; verschobene do_bit8 aufrufen
mcprtende:
        defs   3                  ; Platz für Opcode + Adresse

prtindpatch: ; Patch für IND WAIT PRINTER
        call   patchadr+12        ; verschobene do_bit8 aufrufen
prtindende:
        defs   3                  ; Platz für Opcode + Adresse

do_bit8: ; setzt Bit 7 vom Druckerport auf Bit 7 von Akku
        push  af                  ; Zeichen retten
        push  bc                  ; BC-Register nicht verändern!
        ld    b,pioctrlport      ; Steuerport der PIO in B laden
        rlca                       ; Bit 7 v. Zeichen ins Carry
        ld    a,5                ; Bit 5 vom Port C ansprechen
        rla                        ; Carry in Bit 0 vom Akku
        out   (c),a              ; Port-Bit entsprechend beeinflussen
        pop   bc                  ; BC-Register restaurieren
        pop   af                  ; Zeichen wieder holen
        ret                       ; zurück zum Aufrufer

patchende: ; Ende der Treiberroutine

nocpm30:
        db    'Dieser Druckertreiber kann nur im CP/M 2.2
        db    'installiert werden!',cr,lf,strende
mesg1:  db    'Der Treiber fuer das achte Druckerbit '
        db    'wird nun installiert!',cr,lf,strende

        end      start          ; Ende des Assemblerfiles

```

## 2.4 Druckerschnittstelle mit 8 Bit

## Teil 7: Hard- und Software-Ergänzungen

```
100 '*****
110 '*** (C)opyright 1986 by Andreas Kißlinger **
120 '*** Programm zur Ansteuerung einer 8 Bit ***
130 '*** Druckerschnittstelle. Der Patch wird ***
140 '*** unter der Adresse startadr abgelegt ***
150 '*** und patcht die Indirection ***
160 '*** MC WAIT PRINTER. Gute Adressen fuer ***
170 '*** den Patch sind z.B. &BE00 oder ***
180 '*** SYMBOL AFTER 256:MEMORY HIMEM-13 ***
190 '*** startadr=HIMEM+1:SYMBOL AFTER 240 ***
200 '*****
210 startadr=&BE00
220 IF PEEK(&BDF2)<>(UNT(startadr)AND &FF) THEN 250
230 IF PEEK(&BDF3)= (INT(startadr/256)AND &FF) THEN 380
240 ' mit den beiden IF's wird mehrfaches Initialisieren vermieden!
250 PRINT"Der Treiber fuer das achte Druckerbit wird installiert"
260 FOR n=startadr TO startadr+9
270     READ byte
280     POKE n,byte
290 NEXT
300 ' Jetzt noch die alte Indirection eintragen
310 POKE startadr+10,PEEK(&BDF1)
320 POKE startadr+11,PEEK(&BDF2)
330 POKE startadr+12,PEEK(&BDF3)
340 ' und die Indirection zur neuen Routine umbiegen
350 POKE &BDF1,&C3
360 POKE &BDF2,UNT(startadr)AND &FF
370 POKE &BDF3,INT(startadr/256)AND &FF
380 '*** Fertig ist der Patch!! ***
390 DATA &F5,&06,&F7,&07,&3E,&05,&17,&ED,&79,&F1
400 ' push af : ld b,f7 : rlca : ld a,5 : rla : out (c),a : pop af
```

## 7/2.5

# 256 kByte-Speicher für Schneider CPC

## Allgemeines

Vor Ihnen liegt eine Bauanleitung für eine 256 kB-Speichererweiterung, die intern in den CPC eingebaut wird. Die Bauanleitung wurde auf dem CPC 464 entwickelt, dürfte aber ohne Schwierigkeiten auch auf dem 664 und 6128 einsetzbar sein.

Sie besteht aus einer Ansteuerungsschaltung, die aus handelsüblichen IC's aufgebaut wird, und aus 8 x 256 kBit-dynamischen RAM-Bausteinen vom Typ 41256 oder 50256, die die 64 kBit-RAMs im CPC ersetzen.

Die zugehörige Treibersoftware ist hauptsächlich für den Einsatz von CP/M geschrieben:

'BankBIOS' heißt das Programm, das unter CP/M viele Verbesserungen bietet, z. B. eine RAM-Disk mit 127 KB Kapazität, 58.5 KB TPA und eine DIN-Tastatur. Mit 'Disk-Form.PT' lassen sich für den Betrieb mit CP/M fremde (und auch doppelseitige) Diskettenformate nutzen.

Daher wird auch der Anschluß eines Zweitlaufwerks in einem Abschnitt beschrieben.

Unter BASIC kann mittels eines Bankmanagers auf die zusätzlichen drei Speicherbänke à 64 KB zugegriffen werden.

Die Kompatibilität des CPC 464 wird durch den Einbau dieser Speichererweiterung nicht beeinträchtigt.

## Hard- und Softwarevoraussetzungen

**unter CP/M 2.2:**

- 58.5 KB TPA
- RAM-Floppy "C" mit 128 KB
- 8bit-Centronics-Schnittstelle
- zweiseitige Diskettenformate möglich (mit Formatierprogramm)
- DIN-Tastatur

**unter BASIC oder eigenen Z80-Programmen:**

- Bankmanager
- 8bit-Centronics-Schnittstelle





2.5 256 kByte-Speicher für Schneider CPC

Teil 7: Hard- und Software-Ergänzungen

Verdrahtungsplan

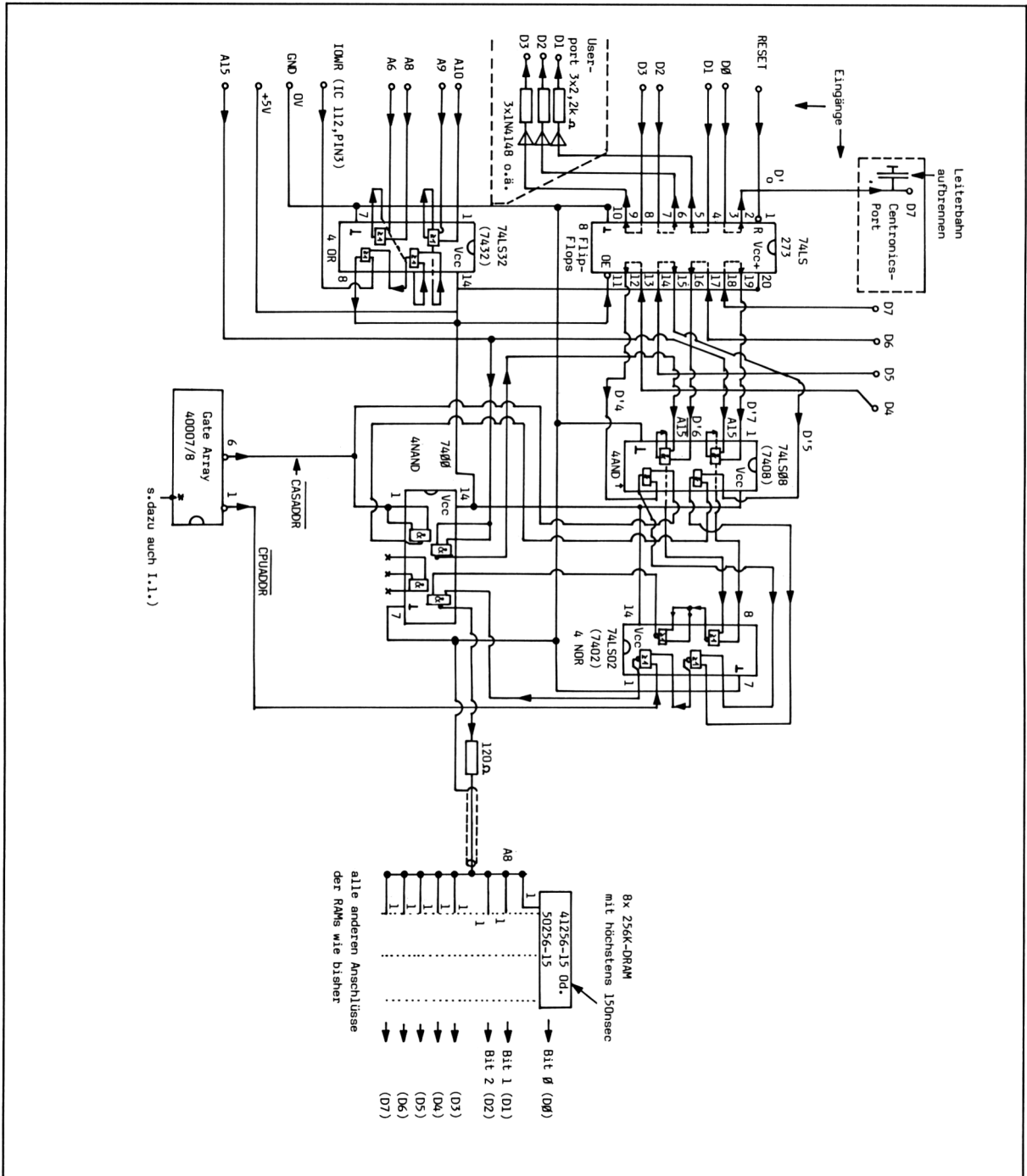


Bild 7/5.2.1-2

## 2.5 256 kByte-Speicher für Schneider CPC

Teil 7: Hard- und Software-Ergänzungen

## Bauteile-Liste

1x 7400	- IC	
1x 74LS02	(7402) - IC	
1x 74LS08	(7408) - IC	
1x 74LS32	(7432) - IC	
dafür 4x 14pol.	- IC Sockel	
1x 74LS273	- IC	
dafür 1x 20pol.	- IC Sockel	
1x 120 $\Omega$	- Widerstand	
3x 2.2 k $\Omega$	- Widerstand	
3x 1N4148 o. ä. Si-Dioden		
8x 41256-15 oder 50256-15 DRAMs (vor statischer Aufladung schützen!)		
8 cm x 5 cm	Lochrasterplatine	
2 cm x 10 cm	Antistatik-Schaumstoff	
2x	Distanzrollen (5mm lang, 3,5 mm $\phi$ )	
	20 cm einadriges abgeschirmtes Kabel	
Litze, Lötzinn (mind. 50% Sn), Isolierband		
Entlötlitze		> wenn die RAMs
8x 16pol.	- IC Sockel	> nicht gesockelt sind
In Klammern: Ersatztypen		

## Einbauanleitung

Zunächst lösen Sie die 6 eingelassenen Schrauben auf der Unterseite des CPC's, kippen das Oberteil (mit Tastatur und Recorder) nach vorn über, lösen seine Verbindungen mit der Hauptplatine <(1) u. (2)> und legen es beiseite.

Nun liegt die Hauptplatine offen vor Ihnen:

## 2.5 256 kByte-Speicher für Schneider CPC

Teil 7: Hard- und Software-Ergänzungen

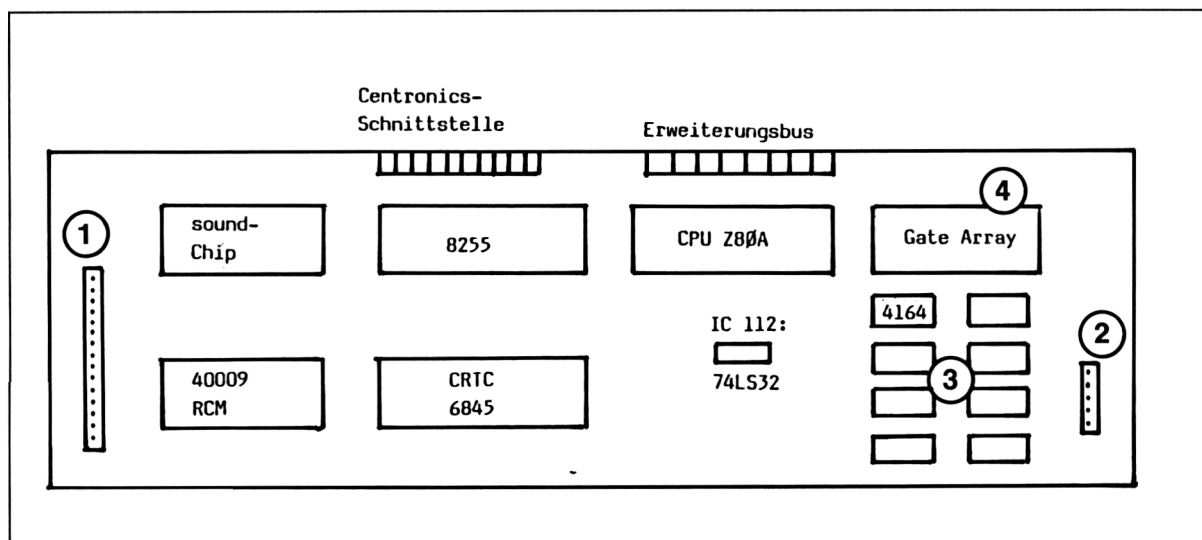


Bild 7/2.5.1-3

Prüfen Sie zuerst, ob die RAMs (3) gesockelt sind; normalerweise ist das der Fall, so daß dann Entlötlitze und die acht 16pol. IC-Sockel in der Bauteile-Liste entfallen.

Liegt das GA(Gatearray(4)) in der Kühlblech-Version vor, so markieren Sie die linke Seite, um beim späteren Einsetzen Verwechslungen (und böse Überraschungen) zu vermeiden. Bevor Sie das GA und die 64 K-RAMs mit Hilfe eines Schraubenziehers vorsichtig aus den Sockeln heben, sollten Sie an einer Wasserleitung o. ä. intensiv anfassen, um evtl. vorhandene statische Aufladung (die den RAMs leicht den Garaus machen kann) abzuleiten. Schrauben Sie nun auch die Hauptplatine ab. Wenn die RAMs nicht gesockelt sind, drehen Sie die Platine um. Die RAMs müssen dann vorsichtig (Hitze!) mit der Entlötlitze ausgelötet und die 16polige IC-Sockel eingelötet werden.

Die 64kBit-RAMs (4164) sind in den Antistatik-Schaumstoff einzustecken und bis zu weiterer Verwendung dort aufzubewahren (sie können dazu verwendet werden, einen PC von 512 auf 640 KB aufzurüsten).

Als nächstes löten Sie die Ansteuerungsschaltung gemäß Verdrahtungsplan Kapitel 7/2.5.1-2 zusammen. Es werden die Sockel eingelötet, ohne bereits die TTL-ICs einzusetzen. Danach sind die Anschlußkabel anzulöten, die folgende Längen haben sollen (um später nicht durcheinander zu kommen, sollten die einzelnen Kabel mit Signalfähnchen bezeichnet werden):

2.5 256 kByte-Speicher für Schneider CPC

Teil 7: Hard- und Software-Ergänzungen

Eingänge	
$\overline{\text{RESET}}$ -, Daten- und Adreßleitungen	: 7 cm (zum Erweiterungsbus)
$\overline{\text{CPUADDR}}$ -, $\overline{\text{CASADDR}}$ -Signal	: 15 cm (zum Gate-Array)
+5V, GND	: 15 cm (zur Stromversorgungsleiterbahn)
$\overline{\text{IOWR}}$	: 20 cm (zu IC 112; (74LS32), Pin 3)

Ausgänge	
A8	: 20 cm abgeschirmtes Kabel (zu IC 124, Pin 1; Alle Adreßleitungen aller RAMs sind parallel verbunden, auch Pin 1 (A8-Signal).)
D0'	: 25 cm (zu Pin 9 der Centronics-Parallelschnittstelle (D7), vorher Massenverbindung auftrennen)
ggf	:
D1'-D3'	: 20 cm (zum Userport; die Kabel können aus dem CPC herausgeführt oder es kann eine Buchse (z. B. 9pol D-Sub) auf der Rückseite angebracht werden.)

Die fertige Schaltung wird an der in Bild 7/2.5.1-3 sichtbaren Stelle montiert, im Querschnitt sieht das folgendermaßen aus:

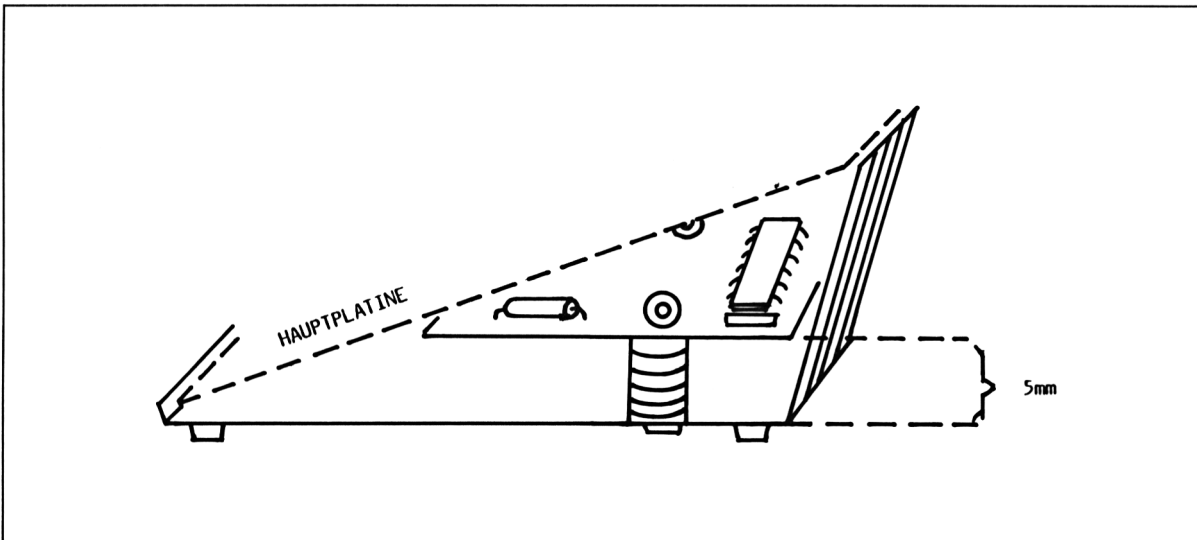


Bild 7/2.5.1-4

Die Pins der Signale, die am Erweiterungsbus abgegriffen werden, entnehmen Sie bitte dem Schneider Bedienungshandbuch Anhang V, S. 2.

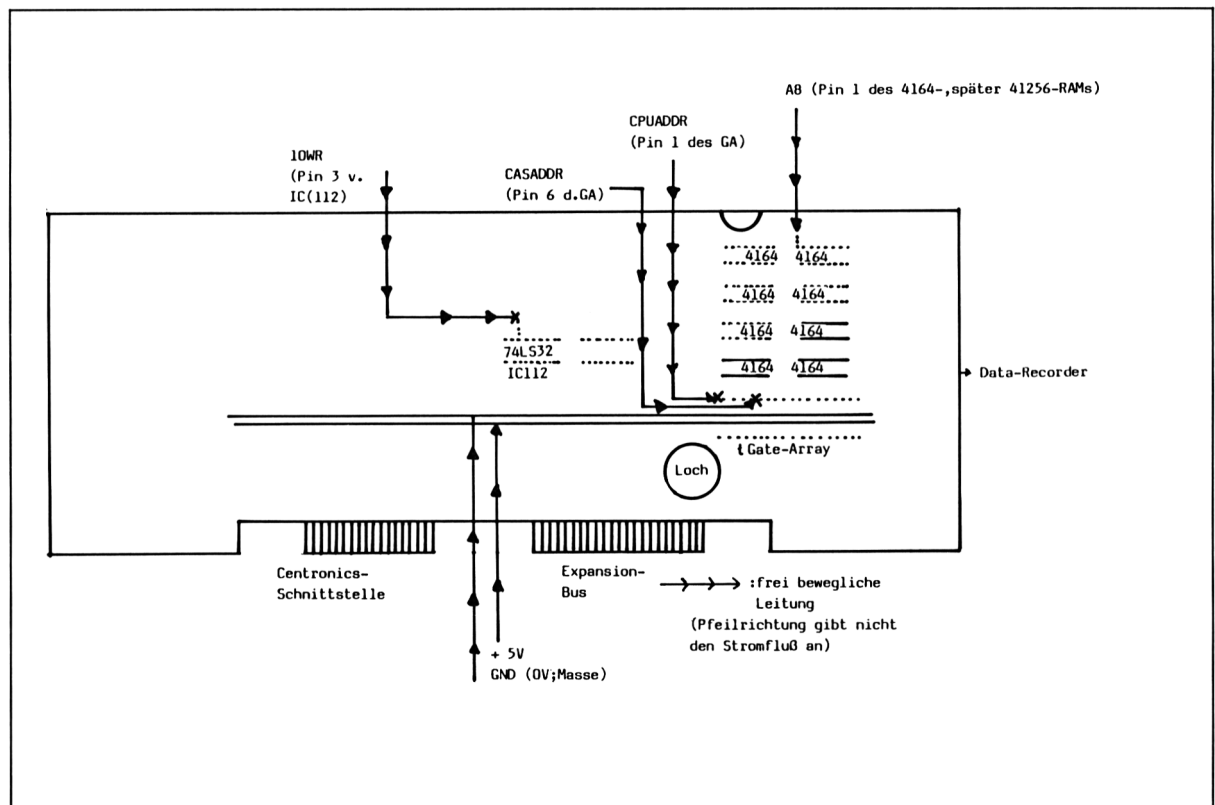
Alle anderen Signale werden an der Ober- bzw. Unterseite der Hauptplatine des Rechners angeschlossen. Die Stromversorgung (+ 5V, GND) kann wahlweise am Erweite-

## 2.5 256 kByte-Speicher für Schneider CPC

Teil 7: Hard- und Software-Ergänzungen

rungsbus oder auch an der Hauptplatine angeschlossen werden. Die beiden folgenden Zeichnungen zeigen, wie die Anschlüsse an der Hauptplatine herzustellen sind.

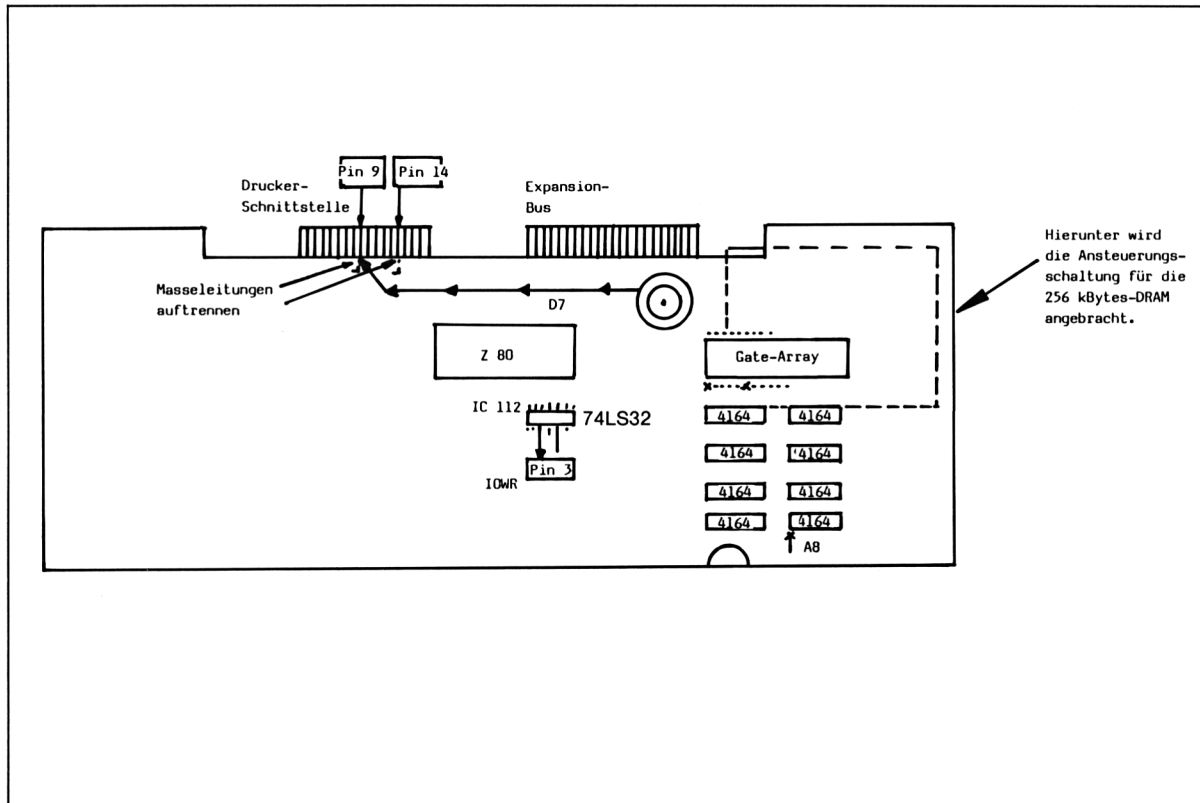
Abschließend setzen Sie die TTL-ICs ein, schrauben die Hauptplatine auf das Gehäuseunterteil, setzen GA (mit Markierung od. Nase nach links) und die 256kBit RAMs (vorher statische Aufladung eliminieren!) ein, dann schließen Sie den CPC wieder in umgekehrter Reihenfolge wie beim Öffnen (Recorder und Tastatur anschließen, zusammenschrauben).



**Bild 7/2.5.1-5**  
Hauptplatine von unten gesehen

## 2.5 256 kByte-Speicher für Schneider CPC

## Teil 7: Hard- und Software-Ergänzungen



**Bild 7/2.5.1-6**  
Hauptplatine von oben gesehen

## Funktionsweise der Schaltung

### Portadressierung

Die Signale  $\overline{\text{IORQ}}$  und  $\overline{\text{WR}}$  des Z80 werden in IC112 des CPC zu dem Signal  $\overline{\text{IOWR}}$  verknüpft (ODER). Dieses IOWR-Signal wird gemäß Schaltplan mit den Adreßleitungen A10, A9, A8 und A6 im 74LS32 verknüpft, so daß der OE-Eingang des 74LS273 nur dann low ist, wenn sowohl die Adreßleitungen A10, A9, A8 und A6, als auch IORQ- und WR-Signal des Z80 low sind.

Es ergibt sich somit eine mögliche Portadresse von F800H, um aber evtl. Kollisionen mit anderen Peripheriebausteinen zu vermeiden, werden alle möglichen Adreßbits gesetzt, so daß die effektive Portadresse **F8BFH** beträgt.

**2.5 256 kByte-Speicher für Schneider CPC**

Teil 7: Hard- und Software-Ergänzungen

Wird nun mittels eines OUT (C),A-Befehls mit BC=F8BFH dieser Port angesprochen, so wird der OE-Eingang des 74LS273 low. Das Datenbyte in A, das nun an den Datenleitungen anliegt, wird zu den Ausgängen des 74LS273 durchgeschaltet und liegt dort bis zu einer Neuprogrammierung oder einem Reset an.

Wozu nun diese Datenbits genutzt werden, soll im folgenden geklärt werden:

D0'	stellt das 8. Bit der Centronics-Parallelschnittstelle, das bisher auf Masse (0V) lag. Diese Masseverbindung (an der Druckerschnittstelle) muß vor dem Anschluß von D0' an Pin 9 aufgetrennt werden.  (Die Abtrennung des Pins 14 von Masse hat hiermit nichts zu tun, sie dient lediglich dazu, den doppelten Zeilenvorschub einiger Drucker zu unterbinden.)
D1'-D3'	stehen dem Anwender für eigene Erweiterungen zur Verfügung, sie sollten zur Sicherheit mit Schutzdioden und 2.2 K $\Omega$ -Abschlußwiderständen versehen werden.
D4'-D7'	werden für das Bank-Switching benutzt:

**Bank-Switching**

Zunächst eine Beschreibung der Signale  $\overline{\text{CPUADDR}}$ ,  $\overline{\text{CASADDR}}$  und A15:

Das Signal  $\overline{\text{CPUADDR}}$  (Pin 1 des GA) ist dann low, wenn die CPU die Adreß- und Datenleitungen für sich beansprucht. Das ist der Fall, wenn der Z80 auf RAM oder ROM zugreift. Dieses Signal ist notwendig, da auch der Videocontroller (6845) auf das RAM zugreift, um den Inhalt des VideoRAMs auf den Monitor zu bringen. Dann ist das Signal high. Für die Speichererweiterung ist das Signal insofern von Bedeutung, daß Speicherbänke nur für den Z80 zur Auswahl stehen, nicht aber für den Videocontroller, der immer auf den gleichen Bereich (Bank 0) zugreifen soll, unabhängig von der gewählten Speicherbank.

Das Signal  $\overline{\text{CASADDR}}$  (Pin 6 des GA) ist dann low, wenn die zweite Adreßhälfte zum RAM geschickt werden soll. Dieses Signal ist notwendig, da dynamische RAMs (wie es die 4164-, 41256-, 50256-Typen sind) ihre Adreßinformationen zum Auslesen oder Beschreiben einer Speicherzelle in eben zwei Hälften und nicht als „Ganzes“ (wie ROMs, EPROMs) bekommen. Für die Speichererweiterung ist dieses Signal dazu da, die Anwahl von 4 Speicherbänken zu je 64 KB zu ermöglichen.



## 2.5 256 kByte-Speicher für Schneider CPC

## Teil 7: Hard- und Software-Ergänzungen

Das Signal A15 (am Erweiterungsbus) dient dazu, Speicherbänke in zwei Hälften anzusprechen.

Für die Schaltung ergibt sich folgende Wertetabelle (Auszug):

Eingänge					Ausgänge			
Zeile	D7'	D6'	D5'	D4'	$\overline{\text{CASADDR}}$	$\overline{\text{CPUADDR}}$	A15	A8
1	0	0	0	0	0	0	0	1
2	0	0	0	0	1	0	0	1
3	0	0	0	0	0	1	1	1
4	0	1	0	1	0	0	1	1
5	0	1	0	1	1	0	1	1
6	0	1	0	1	0	0	0	0
7	0	1	0	1	1	0	0	1
8	0	1	1	1	0	0	0	0
9	0	1	1	1	1	0	0	0
10	1	1	1	1	x	0	1	0
11	1	1	1	1	x	1	x	1

Wie aus Zeile 1 zu ersehen ist, liegt der Ausgang A8 im Grundzustand (nach dem Einschalten) high, dies ändert sich auch nicht, wenn die erste Adreßhälfte zum RAM geschickt wird (Zeile 2), der Videocontroller auf das RAM zugreift oder sich A15 ändert (Zeile 3).

Wird nun auf Port F8BFH der Wert 0101xxxxB ausgegeben, so ändert sich bei gesetztem A15 nichts, egal, ob die erste oder die zweite Adreßhälfte zum RAM geschickt wird (Zeilen 4 und 5).

Bisjetzt blieb also immer die Standardbank (als Bank 0 definiert) selektiert. Wird nun A15 low, weil nun auf eine Adresse < \$8000H zugegriffen wird, so ändert sich das Signal A8 in Abhängigkeit von CASADDR (Zeilen 6 und 7). Für die 256K-DRAMs ist es ein Unterschied, ob A8 immer 1-, immer 0-, 0-1 oder 1-0-Pegel in Abhängigkeit vom CASADDR- (an den Speichern heißt es CAS-) Signal hat.

Auf diese Weise werden 4 Speicherbänke zu je 64 KB erreicht.

- Die Zeilen 8/9 zeigen Beispiele, in denen die Speicherbank 3 angewählt wird (Unabhängigkeit A8 vom CASADDR-Signal).
- Zeile 10 zeigt den Zugriff auf die gesamte Bank 3 (Unabhängigkeit A8 vom A15-Signal).
- Zeile 11 zeigt, daß bei Zugriff des Videocontrollers unabhängig von den anderen Signalen A8 High-Pegel annimmt (Bank 0).

**2.5 256 kByte-Speicher für Schneider CPC**

Teil 7: Hard- und Software-Ergänzungen

**Zusammenfassung**

Bankswitch Port F8BFH	Bit7 obere Bankhälfte	Bit6 untere Bankhälfte schalten	Bit5 Bit4 Bank Nr. . (binär)	Bit3 Bit2 Bit1 Anwender . . . . .	Bit0 D7 Centronics
--------------------------	-----------------------------	--	------------------------------------	--------------------------------------	-----------------------

**Beispiel:**

```

LD BC,F8BFH
LD A,10010001B
OUT (C),A ;Bank 1,2. Hälfte (b); Bank 0,1. Hälfte (a)
           D7 gesetzt
    
```

**Literaturhinweis Hardwareteil**

„CPC 464 INTERN mit komment. ROM-Listing“, R. Brückmann/L. Englisch/K. Gerits, Data Becker, Düsseldorf, Ausgabe 1985, ISBN 3-89011-080-0. Verwendet: Seite 22–25, 36–39, 72–74 sowie Schaltpläne.

**Anschluß eines Zweitlaufwerks**

Für den Anschluß als Zweitlaufwerk sind nur Laufwerke geeignet, deren Bus der Shugart-Norm entspricht und die in MFM (Modified Frequency Mode), d. h. mit einer Übertragungsrate von 250 KBit/s arbeiten (auch als Double-Density bezeichnet). Die Spurwechselzeit (Steprate) darf höchstens 12 ms betragen, durch softwaremäßige Änderung ist aber auch der Anschluß von Laufwerken mit größerer Steprate möglich (für CP/M mit SETUP.COM installieren).

Softwaremäßig lassen sich außergewöhnliche Laufwerkparameter so übergeben: AMS-DOS-Routine „Drive Parameter“ ab #C60D im ROM (7):

<b>Aufruf:</b>	LD HL,PARATAB	RST #18	DEFW ADR	RET
<b>ADR:</b>	DEFW #C60D	;Routine „Drive Parameter“		
	DEFW 7	;AMSDOS-ROM		

## 2.5 256 kByte-Speicher für Schneider CPC

Teil 7: Hard- und Software-Ergänzungen

PARATAB:	DEFW	MHWAIT	;Motor-Hochlaufzeit in 1/50 s (1s)
	DEFW	MNWAIT	;Motor-Nachlaufzeit (nach dem letzten Zugriff) in 1/50 s (5 s)
	DEFB	175	;Steprate beim Formatieren (mgl. nicht ändern)
	DEFB	LWAIT	;Wert für lange Verzögerungsschleife (15)
	DEFB	STEP	;Steprate in ms (12)
	DEFB	HDUNL	;Head Unload Time (FDC 765-Code: $1*32 = 32\text{ms}$ )
	DEFB	HDLD	;Head Load Time (FDC 765-Code: $(3*4)+4 = 16\text{ ms}$ )

8-Zoll-Laufwerke sind nicht nur wirtschaftlich uninteressant, sondern auch aus schaltungstechnischen Gründen nicht verwendbar.

3-Zoll-Laufwerke werden im allgemeinen nur in der 1x40-Spur-Version – die auch in der DDI-1-Floppy Verwendung findet – angeboten, so daß die Diskettenkapazität lediglich verdoppelt werden kann. Angesichts der hohen Preise für 3"-Disketten ein dünner Erfolg.

Sinnvoll ist dagegen der Anschluß eines 5.25"- oder 3.5"-Laufwerks, da diese Formate sehr weit verbreitet sind, mit dementsprechend niedrigen bzw. fallenden Diskettenpreisen. Die Diskettenkapazität läßt sich individuell steigern, da Laufwerkstypen nicht nur mit 1x40 Spuren, sondern auch mit 1x80, 2x40 und 2x80 Spuren angeboten werden. Ein weiteres Einsatzgebiet ist das Arbeiten mit CP/M-Fremdformaten, das im Softwareteil näher beschrieben wird. Die meisten Fremdformate haben folgende physikalische Daten: 5.25": 2x40 (u. a. CP/M-86 auf IBM-PC), 3.5": 2x80 Spuren.

## Allgemeine Bauanleitung (Kurzfassung)

Zum Laufwerk sind zwei Verbindungen herzustellen:

### 1. Datenkabel nach Shugart-Norm:

Flachbandkabel, 34adrig, mit 34pol. Stiftstecker für Anschluß an Controllerkabel;  
je nach Laufwerkstyp: 34pol. Direktstecker (für Platinenanschluß) oder  
34pol. Buchenstecker (für Anschluß an Pfostenstiftleiste)

Shugart-Belegung siehe Teil 7 Kapitel 2.5 Seite 14

### 2. Stromkabel:

4adrig mit 4pol. Floppystecker

## 2.5 256 kByte-Speicher für Schneider CPC

Teil 7: Hard- und Software-Ergänzungen

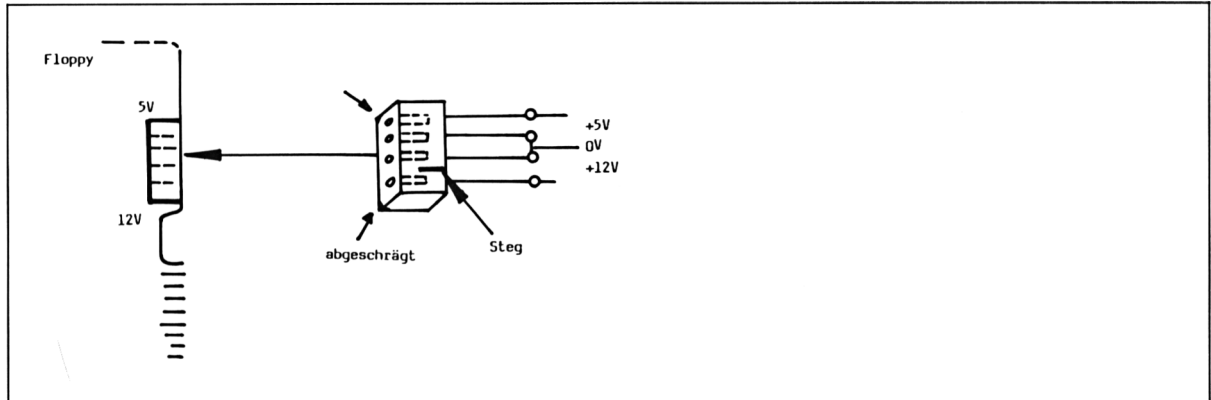


Bild 7/2.5.1-7

Normalerweise benötigen Laufwerke eine Spannung von +5V/+12V bei einem Stromverbrauch von jeweils bis zu 1A. Das benötigte Netzteil sollte möglichst etwas mehr leisten, um einen störungsfreien Betrieb zu gewährleisten. Fertige oder selbstgebaute Netzteile müssen auf jeden Fall den VDE-Vorschriften entsprechen. Insbesondere die Gehäuse, sowohl für Laufwerk als auch für das Netzteil, müssen stabil und sicher sein und dürfen keinen Hitzestau verursachen.

Beim Zweitlaufwerk sind die Abschlußwiderstände, die in der Form eines gesockelten ICs vorliegen, abzuziehen, um die LS-Treiber des Controllers nicht zu überlasten.

Um das Laufwerk als Drive B zu definieren, muß der Steckschuh folgende neue Position bekommen:

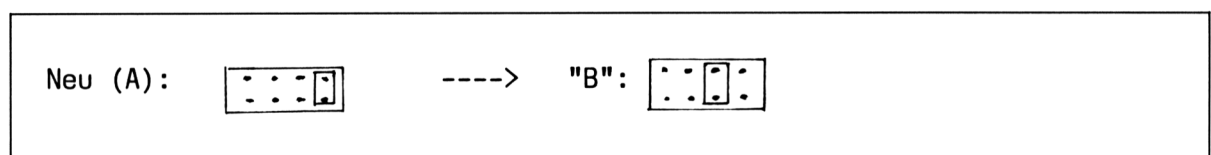


Bild 7/2.5.1-8

Bei einigen Laufwerksfabrikaten ist die  $\overline{\text{READY}}$ -Leitung (s. nächste Abbildung) nicht angeschlossen (Folge: Das Laufwerk wird vom Controller ignoriert). In diesem Fall ist das Ready-Signal laufwerksseitig gegen Masse (0V) zu legen. (Das QUME-Laufwerk '142' sendet beispielsweise kein Ready-Signal aus.)

Manche Laufwerke sollen vor Inbetriebnahme geölt werden, wofür nur harzfreies Nähmaschinenöl zu verwenden ist. Um an das Einfülloch für den Step-Motor oder an die Schienen für den/die Schreib-/Lesekopf/-köpfe heranzukommen, bedarf es jedoch oft einiger Fingerfertigkeit, so daß ggf. ein Fachmann zu befragen ist.

## 2.5 256 kByte-Speicher für Schneider CPC

## Teil 7: Hard- und Software-Ergänzungen

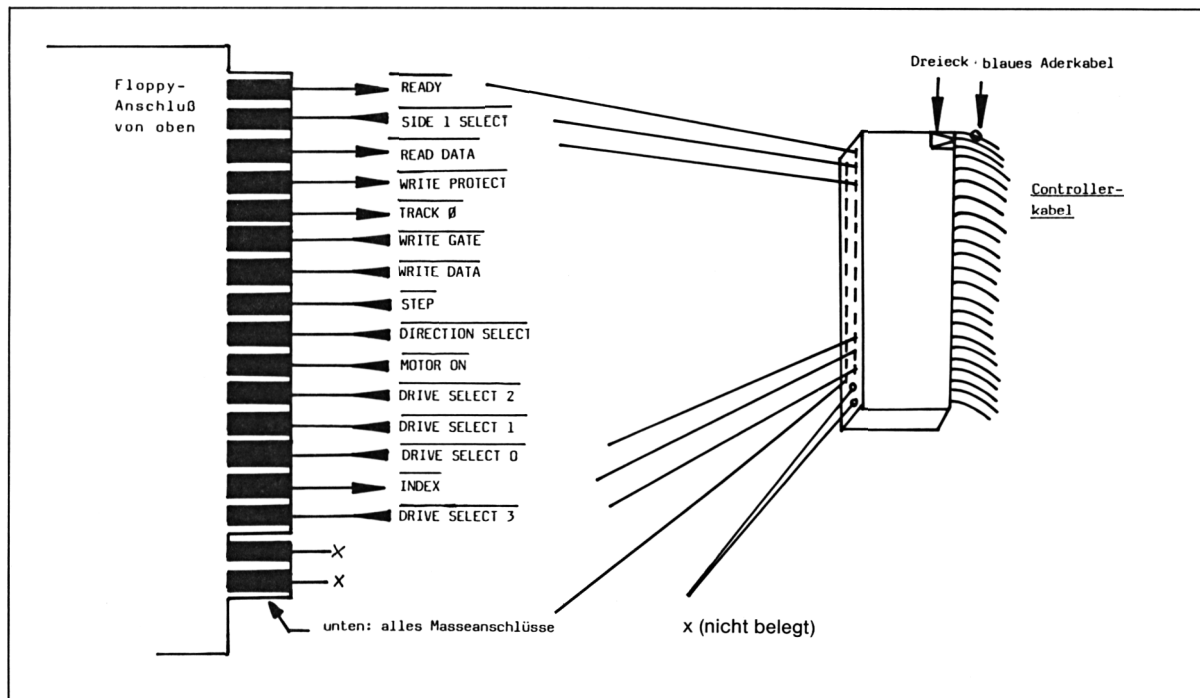


Bild 7/2.5.1-9  
Shugart-Norm-Belegung

## 7/2.5.2

## Software

### Das BankBIOS-Programm

Das Programm BankBIOS (Version 1.45) ermöglicht einen freien Speicherbereich von ca. 58.5 KB unter CP/M 2.2, indem es die Kontrolle über das BIOS im AMSDOS-ROM (als IBIOS bezeichnet) übernimmt und andererseits das System in die Speicherbank 1 hineinlädt und dort das Bindeglied zwischen BDOS und IBIOS per Sprungtabelle herstellt.

Ferner enthält es Routinen zum Betrieb der RAM-Disk C und der 8-Bit-Centronics-Schnittstelle.

**2.5 256 kByte-Speicher für Schneider CPC**

Teil 7: Hard- und Software-Ergänzungen

Unter BankBIOS sind alle Standard-CP/M-Programme lauffähig, z. B.:

- Dienstprogramme:  
ASM.COM, DDT.COM, DUMP.COM, ED.COM, LOAD.COM, MOVCPM.COM, PIP.COM, STAT.COM, SUBMIT.COM, XSUB.COM
- Compiler, Textverarbeitungsprogramme wie:  
Turbo-Pascal, Word-Star

Nicht lauffähig unter BankBIOS sind CPC-spezifische CP/M-Programme wie:

AMSDOS.COM, BOOTGEN.COM, CHKDISC.COM, CLOAD.COM, COPYDISC.COM, CSAVE.COM, DISCCHK.COM, DISCCOPY.COM, DISCKIT2.COM, FILECOPY.COM, FORMAT.COM, SETUP.COM, SYSGEN.COM, LOGO.COM (diese Programme können selbstverständlich unter dem herkömmlichen CPC-CP/M weiterhin benutzt werden).

Daß eine ganze Reihe von Programmen unter BankBIOS nicht laufen, sieht zunächst betrüblich aus, doch bei genauerem Betrachten fällt auf, daß

1. diese Programme von einem größeren, freien Speicherplatz nicht profitieren (sieht man einmal vom LOGO-Interpreter ab, der sich z. T. im AMSDOS-ROM befindet)
2. die vielen Kopier- und Verifizierprogramme durch PIP.COM zu ersetzen sind, da nun mit der RAM-Disk C auf jeden Fall ein Zweit- oder Dritt-„Laufwerk“ zum Kopieren zur Verfügung steht (Option für Verify <V>).

Die RAM-Disk, die wie jedes andere Laufwerk angewählt wird, besitzt eine Nettokapazität von 127 KB bei 32 Directory-Einträgen. (Das Programm AMSDOS.COM ist sowieso überflüssig, da mit <CTRL>+<SHIFT>+<ESC> der gleiche Reset-Erfolg erreicht wird.)

Hinlänglich bekannt, aber trotzdem wichtig, ist, daß beim Unterbrechen der Stromversorgung die Daten der RAM-Disk gelöscht werden. Unter gelöscht ist hier aber nicht ein Speicher, gefüllt mit lauter Nullen oder FFs, zu verstehen, sondern er kann beim Einschalten mit variablen Werten gefüllt sein. Daher erfolgt beim CP/M-Start eine Abfrage, ob das Directory der RAM-Disk gelöscht, d. h. mit dem Füllbyte der Diskette (E5H) versehen werden soll.

Wenn Sie den CPC einschalten, sollten Sie die Frage auf jeden Fall mit „Y“ (Ja) beantworten, damit für das BDOS klar ist, daß die RAM-Disk absolut frei ist (sonst könnten die zufälligen Werte als Einträge gedeutet werden, was zu Problemen beim Kopieren führen kann).

## 2.5 256 kByte-Speicher für Schneider CPC

Teil 7: Hard- und Software-Ergänzungen

Anders liegen die Dinge, wenn Sie bereits CP/M gebootet haben und dann mit CTRL/SHIFT/ESC aussteigen. Der Clou ist, daß bei einem solchen Reset der Inhalt der RAM-Disk NICHT gelöscht oder sonstwie beeinträchtigt wird, d. h. Sie können CP/M nun erneut booten, beantworten dann die Abfrage "C>ERA \*.\* All (Y/N)?" mit "N" und schon liegt der Inhalt der RAM-Disk vor Ihnen (und der befürchtete Datenverlust ist ausnahmsweise einmal ausgeblieben).

Begründung: Bei einem CTRL/SHIFT/ESC-Reset wird nur die Standardbank (als Bank 0 definiert) gelöscht, die Bank1 (CP/M-Bank) und die Bänke 2 und 3 (RAM-Disk) bleiben dem (unerwünschten) Zugriff des CPC-Betriebssystems entzogen.

Der Quelltext nach GENA3-Standard liegt im Listing-Teil als „BANKBIOS.145“ (Teil 1) und „DSBIOS.103“ (Teil 2: Treiberprogramm für Formate bis 2x80 Tracks, Installationsroutine) vor.

Neben dem CPC mit 256 KB-Speicher ist der AMSDOS-Controller für die Lauffähigkeit des BankBIOS-Programms Voraussetzung.

Abschließend ein paar allgemeine Tips zum effektiven Arbeiten mit der RAM-Disk:

Jede RAM-Disk zeichnet sich dadurch aus, daß sie im Gegensatz zum mechanischen Laufwerk verschleißfrei und sehr schnell über zig „Spuren“ hinweg auf einzelne Records zugreifen kann (kein Rattern, sondern echter Random Access!).

Dieser Vorteil wirkt sich daher besonders positiv bei relativen Dateien, Overlay-Dateien und sonstigen Programmen, die häufig aufgerufen werden, aus.

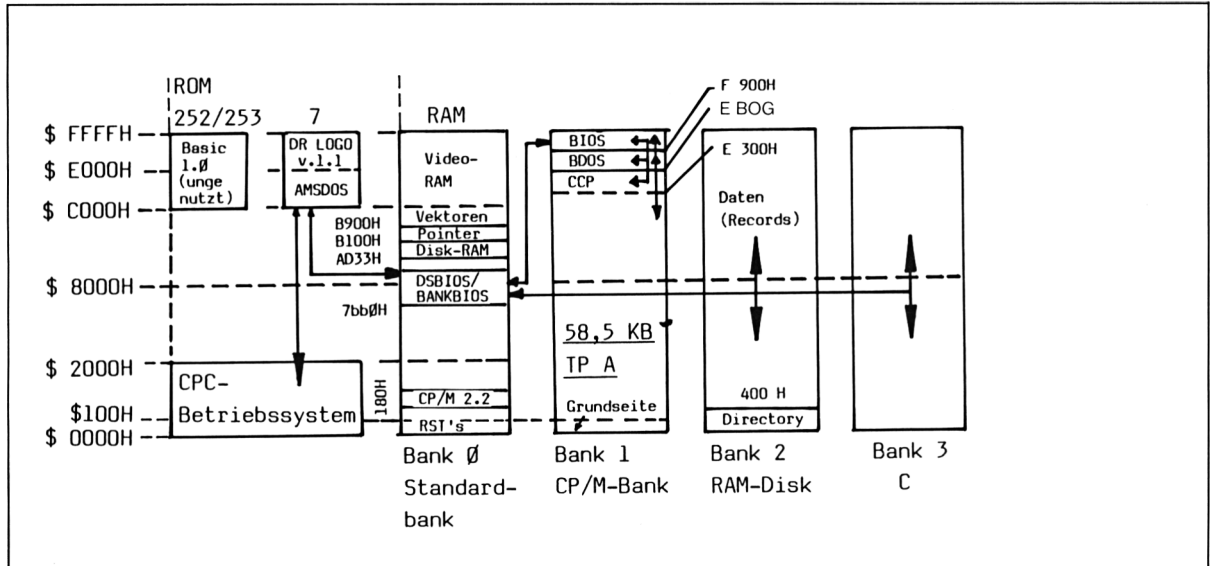
Es ist daher gute Praxis, alle Dateien und Programme, auf die häufige Lesezugriffe unternommen werden, mit PIP in die RAM-Disk C zu kopieren:

```
A>PIP
*c:=a:*OVR
COPYING -
MSG.S.OVR, ERRORS.OVR, DATA.OVR usw.
*c:=a:*COM
COPYING -
STAT.COM, PIP.COM, DDT.COM, SUBMIT.COM, sonstige Dienstprogramme
```

Für mehrfaches Kopieren gleicher Dateien ist die RAM-Disk – soweit die Kapazität reicht – ebenfalls geeignet. Fehl am Platz sind dort hingegen Arbeitsdateien (Pascal-Programme, Assemblerlistings z. B.), da die Gefahr besteht, daß ein Rückkopieren auf den magnetischen Datenträger Diskette vergessen wird. Nach Abschalten des Computers stellt man sonst schweißgebadet fest, daß alle Mühe für die Katz' war.

2.5 256 kByte-Speicher für Schneider CPC

Teil 7: Hard- und Software-Ergänzungen



**Bild 7/2.5.2-1**  
Schematische Funktionsweise des BankBIOS

**DiskForm.PT**

Das BASIC-Programm DiskForm.PT ist ein Diskettenverwaltungsprogramm, mit dem individuelle Diskettenformate für den Einsatz mit BankBIOS installiert werden können. Ferner ist es für die Installation des BankBIOS/DSBIOS-Komplexes selbst zuständig.

Erklärungen im einzelnen:

Menüpunkt 1 dient dazu, unbekannte Fremdformate im IBM34-Standard (definiertes Aufzeichnungsverfahren) zu analysieren. Es lassen sich feststellen:

Nr. des 1. Sektors, Größe eines Sektors (1->256, 2->512 Bytes pro Sektor), Format mit 40/80 Spuren und 1/2seitig:

eigenes Laufwerk	gewünschte Spur-Seite	Spur-Seite im ID	Ergebnis
2x40	z. B. 4-1	4-1	2x40-Format
2x80	4-0 u. 4-1	4-0 u. 4-0/NL	1x40
	4-1	2-1	2x40 ->Doppelschritt
	4-0 u. 4-1	2-0 u. 2-0/NL	1x40 "--
	4-1	4-1	2x80
		NL: Nicht lesbar	



2.5 256 kByte-Speicher für Schneider CPC

Teil 7: Hard- und Software-Ergänzungen

Auf Single-side-Laufwerken sind natürlich auch nur einseitige Formate einsetzbar, so daß nur Seite 0 zu analysieren ist.

Mit Menüpunkt 2 können Formate, die vordefiniert oder selbst programmiert wurden (s. 3), zum Formatieren ausgewählt werden.

Das Programmieren weiterer Formate ist mit dem 3. Menüpunkt möglich. Neben den mit dem ersten Menüpunkt ermittelten Werten sind noch folgende Formatangaben erforderlich:

- Spurwechsel im Wechselschritt (Spur 0, Seite 0 .. Spur 0 Seite 1 ....) ODER CP/M86-Verfahren (Spur 0, Seite 0 .. Spur 1, Seite 0 .. letzte Spur, nun auf Seite 1: Spur 39, 38, 37 ... 0)
- Anzahl der Systemspuren
- max. Anzahl der Directory-Einträge
- Blockgröße in KB

Durch den Menüpunkt 4 wird auf dem Bezugslaufwerk gemäß des ausgewählten Formats formatiert (Vendorformate).

Unter BankBIOS ist es möglich, jeweils ein Format mit 512 Bytes/Sektor und eines mit 256 Bytes/Sektor zu verwenden. Standardmäßig ist das 2x40 und das TA-alphatronic PC-Format eingestellt. Ändern läßt sich dies unter Menüpunkt 5.

Mit Punkt 6 wird BankBIOS/DSBIOS auf einer in Laufwerk A eingelegten Diskette installiert (siehe Kapitel 7/2.5.1-3).

Bereits definierte Formate:

Format	CPC- Sys	CPC- Data	CP/M86	1x80 2x40	2x80	TA alpha.
Sektoren/Spur :	9	9	8	10	10	16
Bytes/Sektor :	512	512	512	512	512	256
Systemspuren :	2	0	1	0	0	2 (p. S.)
max. Direintr. :	64	64	64	128	256	256
Blockgr. in KB :	1	1	1	2	2	2
Nr. 1. Sektor :	41H	C1H	1	10	10	1

Da unter BankBIOS das CP/M bei einem Warmstart (^C) aus dem RAM und nicht von Diskette nachgeladen wird, können auch auf Laufwerk A das Daten- und sonstige einseitige Formate benutzt werden.

**2.5 256 kByte-Speicher für Schneider CPC**

Teil 7: Hard- und Software-Ergänzungen

Bei jedem Formatwechsel, auch auf Laufwerk B, muß ein Login durchgeführt werden (im CCP: ^C).

**Installation des BankBIOS**

Bevor das BankBIOS mit DiskForm.PT installiert wird, sind folgende Schritte zu unternehmen:

Zunächst rufen Sie das Programm MOVCPM.COM (bei dem Sie sich bis jetzt vielleicht gefragt haben, wozu es eigentlich existiert) Ihrer Systemdiskette folgendermaßen auf:

```
A>MOVCPM 255 *
```

(Die Zahl 255 steht für einen Speicherplatz von  $255 * 256 = FF00H$  Bytes (inkl. üblicher BIOS-Bereich). Dies ist für MOVCPM der Maximalwert, obwohl man wegen des geringen Speicherplatzbedarfs des BIOS in der CP/M-Bank in diesem Fall noch höher gehen könnte.)

Darauf meldet sich CP/M zurück mit:

```
CONSTRUCTING 63k CP/M vers 2.2  
READY FOR "SYSGEN" OR  
"SAVE 34 CPM63.COM"
```

Wir tun letzteres und speichern das an einen Speicherplatz von 63k angepaßte CP/M als CPM63.COM ab:

```
A>SAVE 34 CPM63.COM
```

Eine Diskette im Systemformat, auf der sich SETUP.COM befindet, ist dann einzulegen. SETUP wird aufgerufen, gemäß beiliegendem Protokoll wird die DIN-Tastaturbelegung installiert. Mit BOOTGEN.COM kann diese Tastaturbelegung auf andere Disketten kopiert werden.

Nun rufen wir SYSGEN.COM folgendermaßen auf, mit dem Ziel, die Disketten mit dem 63K-CP/M zu versehen, die dann auch mit BankBIOS gebootet werden (auf keinen Fall die CP/M-Originaldiskette).

```
A>SYSGEN CPM63.COM*
```

## 2.5 256 kByte-Speicher für Schneider CPC

Teil 7: Hard- und Software-Ergänzungen

Folgen Sie dann den Anweisungen des Programms. Beachten Sie bitte ferner, daß SYSGEN.COM zu den CPC-spezifischen CP/M-Programmen gehört und damit unter BankBIOS (vers 1.45) nicht lauffähig ist, es muß also vom CPC-CP/M her aufgerufen werden.

Nun steigen wir aus dem CP/M mit CTRL/SHIFT/ESC aus und starten (im BASIC) das Programm BANKBIOS.BAS, das die Binärdatei BB145.BIN erzeugt.

Nach diesen Vorarbeiten kommt bei der nun folgenden BankBIOS/DSBIOS-Installation das BASIC-Programm Diskform.PT, das auch die zusätzlich einsetzbaren Diskettenformate verwaltet, zum Einsatz. Mit Menüpunkt 5 werden die einzusetzenden Diskettenformate ausgewählt, mit Menüpunkt 6 wird die Installation des BankBIOS/DSBIOS auf Laufwerk A ausgeführt.

Dabei wird der Bootsektor (&41) gepatcht und der BankBIOS/DSBIOS-Komplex in den Sektoren &43-&47 abgelegt, die sonst unbenutzt sind.

```
A>setup

SETUP V2.0
Initial command buffer empty

Is this correct (Y/N):_Y
                                > (CLS; Bildschirm gelöscht)
Sign-on string:

^Ü  ^Ü!09^Ü  ** 59.5 kB TPA-CP/M 2.2 - Installation - Disk ** ^J^M

Is this correct (Y/N):_Y
                                > (CLS)
** Printer power-up string empty

Is this correct (Y/N):_Y
                                > (CLS)
No keyboard translations set

Is this correct (Y/N):_N

Enter required command from:-
  A - Add key translation (...)
  D - (...)
  C - (...)
  F - Finish translation
```

Listing 7/2.5.2-1 (Teil 1)

## 2.5 256 kByte-Speicher für Schneider CPC

Teil 7: Hard- und Software-Ergänzungen

```
Command: _a57,51,64
          a41,55,47
          a32,48,61
          a25,126,63
          a24,94,96
          a26,125,93
          a17,43,42
          a70,254,254,253
          a29,124,92
          a28,123,91
          a19,35,39
          a39,44,59
          a31,46,58
          a30,45,95
          a22,60,62
          a43,122,90,26
          a71,121,89,25
          f

Is this correct (Y/N): _Y
.
.
.
(Alle weiteren Fragen ebenfalls mit "Y" beantworten.)
```

Listing 7/2.5.2-1 (Teil 2)

## Das BTRANS-Programm

---

Unter CP/M ist es wegen der fest definierten BIOS-Schnittstelle gut möglich, die zusätzlichen Speicherbänke als CP/M-Bank und für eine RAM-Disk zu nutzen. Vergleichbares ist jedoch im BASIC nur mit gigantischem Programmieraufwand zu erreichen, da BASIC ein Interpreter ist, der ständig auf das ganze Programm und die Variablen zugreifen muß. Meines Erachtens ist ein freier Speicher von ca. 42K aber auch völlig ausreichend, was im CP/M hingegen nicht der Fall ist. Auch der Einbau einer RAM-Disk würde sich ähnlich aufwendig gestalten, da Routinen wie Sektor - oder Record lesen/schreiben nicht per Vektoren, die man patchen könnte, angesprungen werden.

## 2.5 256 kByte-Speicher für Schneider CPC

Teil 7: Hard- und Software-Ergänzungen

Mir erscheint daher ein Programm, das unabhängig von BASIC oder AMSDOS auf die zusätzlichen Speicherbänke zugreift, am sinnvollsten. Die Routinen, die per RSX-Befehl (in BASIC) aufgerufen werden, kopieren frei definierbare Blöcke zwischen Standardbank (0) und den anderen Speicherbänken (1-3).

Es handelt sich um zwei Befehle:

IPUT,S,Z,L,B: kopiert einen Block der Länge L ab Adresse S aus Bank 0 nach Bank B(0-3), dort ab Adresse Z.

IGET,Start,Ziel,Länge,Speicherbank:  
wie oben, nur wird aus der Bank B zur Standardbank 0 kopiert.

Wenn B=0, wird bei beiden Befehlen ein gewöhnlicher LDIR-(Blockverschiebungs-) Befehl ohne Bankswitching ausgeführt. Ungültige Parameter bewirken die Nichtausführung des Befehls, aber keine Fehlermeldung.

Die Adressen und Längen sollten in Hexadezimal übergeben werden, da als Parameter (wie bei allen RSX-Befehlen) nur Integer-Werte akzeptiert werden.

Nun einige Anwendungsbeispiele:

IGET,&100,&C000,&4000,1/2/3

bringt nach dem Einschalten eine Graphik auf den Monitor (die den Einschaltzustand der RAMs wiedergibt).

IPUT,368,&100,x,1

kopiert ein BASIC-Programm der Länge x nach Bank 1. Nun können andere BASIC-Programme eingegeben oder geladen werden.

Mit GET,&100,368,x,1 holen (laden) Sie das alte BASIC-Programm zurück. Wenn x groß genug gewählt wurde, können auch die alten Variablen zurückgeholt werden.

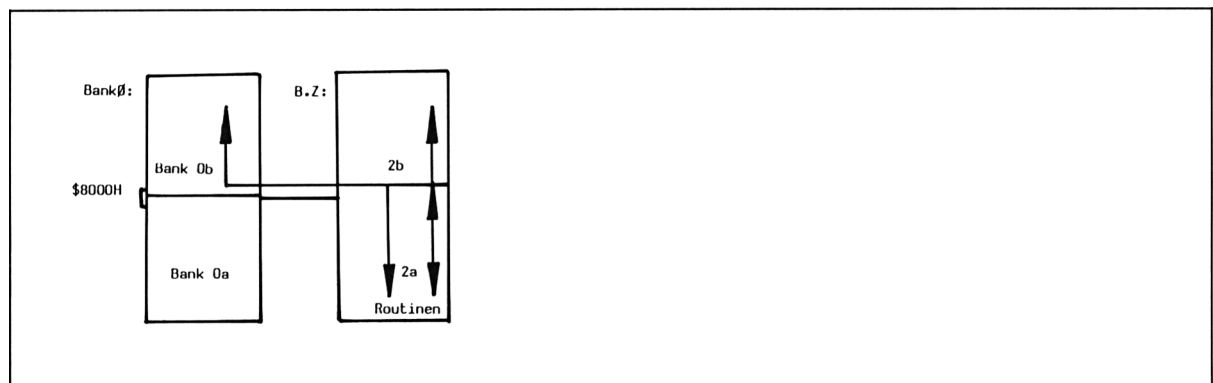
Bei großen Blöcken dauert es einige Sekunden, bis der Kopiervorgang beendet ist. Ein Absturz kann auftreten, wenn in Bereiche kopiert wird, die für BASIC, CPC-Betriebssystem, AMSDOS und BTRANS reserviert sind; diese Bereiche liegen in

- Bank 0: \$008H-\$16FH,\$A500H-\$A60FH,\$A700H-\$BFFF
- Bank 1/2/3: jeweils \$00H-\$12H

## Hinweise zur Erstellung eigener Bank-Switch-Programme

Beim Selektieren einer anderen Speicherbank darf nicht der „Ast abgesägt werden, auf dem man sitzt“, d. h. wenn beispielsweise ein Programm oberhalb \$8000H in Bank 0 auf Bank 1 zugreift (Daten lesen/schreiben oder eine Routine, die sich dort befindet, aufzurufen), so kann das zunächst nur dadurch geschehen, daß erst der untere Bereich bis \$8000H umgeschaltet wird. Gegebenenfalls muß dort hinein eine andere Routine kopiert werden, mit der man auf die gesamte Speicherbank zugreifen kann.

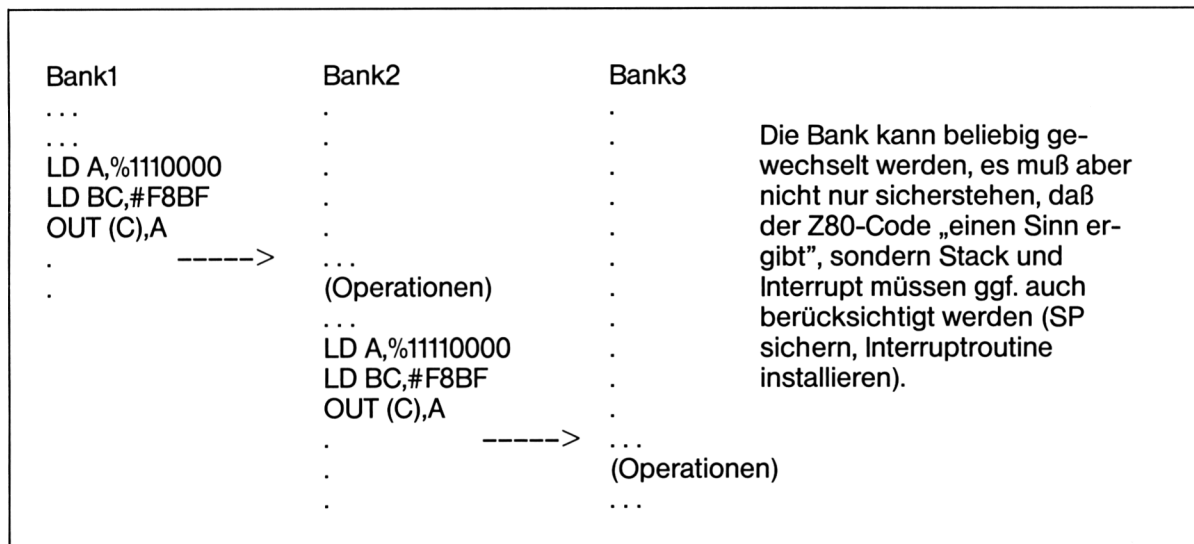
Prinzipiell sind Interrupts während solcher Operationen mit „DI“ zu verbieten, damit die IR-Routine nicht ins „Leere“ geht (was einem Absturz gleichkommt).



Obiges Verfahren wird sowohl bei BankBIOS als auch bei BTRANS verwendet. Eine weitere Möglichkeit besteht darin, parallel liegende Bereiche mit einem gleichen oder zusammenhängenden Bankswitch-Programm nach obigem Verfahren zu versehen. Innerhalb dieses Programms kann dann die gesamte Speicherbank gewechselt werden, da das Programm in gleicher Weise auf der neuen Speicherbank weiterläuft.

## 2.5 256 kByte-Speicher für Schneider CPC

## Teil 7: Hard- und Software-Ergänzungen



## Literaturhinweis Softwareteil

- „CPC 464 Intern mit komment. ROM-Listing“,  
Verwendet: S. 79–86, 130, 296/7 (RSX-Befehle)
- „Das große Floppy-Buch“, R. Brückmann/J. Schieb  
Ein Data Becker Buch, Düsseldorf, Ausgabe 1985 (mit DOS-Listing), ISBN 3-89011-093-2  
Verwendet: S. 28, S. 170, S. 175–177, S. 181–187, S. 191–194
- „CP/M 2.2 Anwender-Handbuch CPC 464/664/6128“, J. Hückstädt  
Markt & Technik Verlag, Haar bei München, Ausg. 1986, ISBN 3-89090-204-9  
Verwendet: S. 125–128, S. 135–137, S. 142–205 (CPC-spezifisch J/N)
- CP/M Operating System Manual, Digital Research, USA  
Verwendet: 6.10. Disk Parameter Tables

## 7/2.5.3 Listings

### „BANKBIOS.145“

```
=====
;=                                     =
;=  BANKBIOS  - Version 1.45A  -      =
;=                                     =
;=  auf Schneider CPC 464 mit DDI-1 =
;=  und 256KB-Speichererweiterung =
;=                                     =
;=  für Betriebssystem CP/M 2.2 mit =
;=  - nunmehr ++ 59.910 Bytes ++ TPA =
;=  - RAM-Disk "C" mit 127KB netto   =
;=  - DIN - Tastatur                =
;=  - RAM-Warmstart                 =
;=  - Doppelkopf-Formate mgl.      =
;=                                     =
:=  written Mai-Juli 1987           =
;=  by      Kai Meier              =
;=          Hauptstraße 25        =
;=          D-2361 Bebensee       =
=====
;
ORG #7B80
;
BANKPORT:EQU #F8BF ;Bankswitch-Portadresse
BUFFA: EQU #0100 ;Buffer in Bank 0a,
BUFFB: EQU #9000 ;      in Bank 0b
GATEARRAY:EQU #0040 ;Portadresse Gate Array u. ROM Konfig.
STACK: EQU #0042 ;Stack-Zwischenspeicher
START: EQU #FE00 ;Bank-BIOS-Handler
UROMON: EQU #BA5E ;KL U ROM (hier AMSDOS-ROM #7) an
UROMOFF: EQU #BA68 ;KL U ROM off
DRIVE: EQU #BE53 ;Drivenr.: A=0, B=1, C=2
RL: EQU #0080 ;Recordlänge: 128 Bytes
```

Listing 7/2.5.3-1 (Teil 1)



## 2.5 256 kByte-Speicher für Schneider CPC

## Teil 7: Hard- und Software-Ergänzungen

```
;
KALTSTART:LD HL,#C17F ;Standard BIOS Tabelle
           LD DE,#0500
           LD BC,#0033
           LDIR ;nach $500
           LD C,#42 ;Konfigurationssektor
           LD DE,#0000 ;Track 0, Drive A
           LD HL,#0300 ;nach $300
           JP #0112 ;Bootsektor gepatched ->

:-----
BOOTSRET:CALL #AD33 ;um Altern. Reg.Set
           DEFW KSINST ;zu bekommen, indirekter Aufruf

:-----
;- BIOS-JP-Block: 17 Jumps + 1 INT-Vektor -
;-----
BIOS: RST 1
      DEFW #C1B2 ;KALTSTART
      RST 1
      DEFW WARMSTART
      RST 1
      DEFW CONSTAT ;CONSOLE STATUS
      RST 1
      DEFW CONIN ;CONSOLE INPUT
      RST 1
      DEFW CONOUT ;CONSOLE OUTPUT
      RST 1
      DEFW PRINTOUT ;PRINTER OUTPUT
      RST 1
      DEFW PUNCHER ;(früher: Lochstreifenstanzer)
      RST 1
      DEFW READER ;(früher: Lochstreifenleser)
      RST 1
      DEFW HOME ;TRACK0
      RST 1
      DEFW SLDRV ;SELECT DRIVE
      RST 1
      DEFW #C524 ;SELECT TRACK
      RST 1
      DEFW #C529 ;SELECT RECORD
      RST 1
      DEFW INSTBUF ;INSTALL BUFFER
      RST 1
      DEFW READRCD ;READ RECORD
      RST 1
      DEFW WRITERCD ;WRITE RECORD
      RST 1
```

Listing 7/2.5.3-1 (Teil 2)

## 2.5 256 kByte-Speicher für Schneider CPC

## Teil 7: Hard- und Software-Ergänzungen

```

        DEFW PRNSTAT ;PRINTER STATUS
        JP  XLT      ;SKEWING TABELLE (nicht benutzt)
LBL1:   DI
        RST 1
        DEFW #0038 ;INTERRUPT
;----- DEVICE
LBL2:   DI          ;Interrupts verbieten
        LD  (ASAVE),A ;Akku retten
        EX  (SP),HL ;Rücksprungadresse holen, HL auf Stack
        LD  (BCSAVE),BC ;BC abspeichern
        LD  C,(HL)
        INC HL
        LD  B,(HL) ;BC:=RST1 Parameter
        LD  (JUMP),BC ;RST1-Parameter hinterlegen
        POP HL
        LD  A,(#0003) ;IO-Byte
        LD  (IOBYTE),A ;hinterlegen
        LD  A,%10010000 ;Bank 0a,1b selektieren
        LD  BC,BANKPORT
        OUT (C),A
        JP  IBIOS ;in Bank 0a
;----- BIOSRET
LBL3:   LD  A,%11010000 ;Bank 1a,1b
        OUT (C),A ;einschalten
        LD  BC,(BCSAVE)
        LD  A,(ASAVE)
        EI
        RET ;ins BDOS zurück
LBL4:   DEFW #0000 ;BCSAVE
LBL5:   DEFB #00 ;ASAVE
LBL6:   DEFW #0000 ;JUMP
LBL7:   DEFB #00 ;IOBYTE
;----- INSTALL
LBL8:   LD  A,%11010000 ;Bank 1a,1b selektieren
        LD  BC,BANKPORT
        OUT (C),A
        LD  A,#C3 ;JP-Opcode
        LD  HL,DEVICE
        LD  (#0008),A
        LD  (#0009),HL ;RST1 auf DEVICE-Routine setzen
        LD  HL,INTV
        LD  (#0038),A
        LD  (#0039),HL ;RST7 (Interrupt-Vektor) setzen
        POP HL ;BDOS-Entry holen
        LD  (#0005),A
        LD  (#0006),HL ;BDOS-Entry

```

Listing 7/2.5.3-1 (Teil 3)

## 2.5 256 kByte-Speicher für Schneider CPC

## Teil 7: Hard- und Software-Ergänzungen

```

LD (#0000),A
LD HL,#0003
ADD HL,DE
LD (#0001),HL ;BIOS-Entry
LD B,18 ;17 BIOS-Jumps + 1 INT-Vektor installieren
EX DE,HL ;Start der Tabelle nach HL
XOR A ;A=0
LD DE,START ;RST1-Tab
RLBL1: LD (HL),#C3 ;JP-Opcode
INC HL
LD (HL),A
INC HL
ADD A,3 ;Nr. des RST 1-Calls *3
LD (HL),D ;Highbyte der RST1-Tabelle
INC HL
DJNZ RLBL1
LD HL,#0004 ;User und Drive
LD A,(HL)
AND #0F ;Drive ausmaskieren
CP #03 ;A,B oder C ?
JR C,RLBL2 ;ist der Fall
LD (HL),0 ;sonst auf "A" setzen
RLBL2: LD C,(HL) ;Drive nach C
EI
RET ;CCP-Entry via RET anspringen
;----- XLT
LBL9: LD H,B ;Nur BC nach HL übertragen
LD L,C
RET
;--- IBIOS-Hilfsroutine: RECD11, Record inn. Bank 1 übertragen ---
LBL10: LD A,%11010000 ;Bank 1ab
LD BC,BANKPORT
OUT (C),A
LD BC,RL ;Record 128 Bytes
LDIR ;übertragen
LD BC,BANKPORT
LD A,%10010000 ;Bank 0a,1b
OUT (C),A
LD A,0
RET
;-----
;Die DPH's für Systemlaufwerk "A", Laufwerk "B" u. RAM-Disk "C":
LBL11: DEFW #0000 ;Skewing Tabelle (nicht genutzt, daher =0)
DEFS 6 ;Zwischenspeicher für BDOS: Spur-, Record- u. DIRnr.
DEFW #FD80 ;Directory-Buffer (128 Bytes)
DEFW DPBA ;Disk-Parameter Block (Formatangaben)
DEFW #FBAD ;DIR-Prüfsummentabelle (128 Bytes:bis zu 512 Dir-Ein.)

```

Listing 7/2.5.3-1 (Teil 4)

## 2.5 256 kByte-Speicher für Schneider CPC

Teil 7: Hard- und Software-Ergänzungen

```

        DEFW #FCC0      ;Blockbelegungstabelle (frei für 64*8 Blöcke)
;Drive B
        DEFW #0000      ;XLT
        DEFS 6          ;TRACK, RECORD, DIRNUM
        DEFW #FD80      ;DIRBUF (Bezeichnungen aus dem CP/M,
        DEFW DPBB       ;DPB      analog wie in "A")
        DEFW #FC20      ;CSV
        DEFW #FD00      ;ALV
;Drive C
        DEFW #0000
        DEFS 6
        DEFW #FD80
        DEFW DPBC
        DEFW #FCA0
        DEFW #FD40
; DPB RAM-Disk "C"
        DEFW #0080      ;SPT: Records pro Track
        DEFB #03        ;BSH: Block Shift > 1KB = 1Block
        DEFB #07        ;BLM: Block Maske >
        DEFB #00        ;EXM: Extent Maske
        DEFW #007F      ;DSM: Maximale Blocknummer (128 KB)
        DEFW #001F      ;DRM: 32 Directory Einträge
        DEFB #80        ;ALD> 1 Directory-Block (binär codiert)
        DEFB #00        ;AL1> "--
        DEFW #0000      ;CKS: kein "Diskettenwechsel" mgl., daher keine
        ;              ; Prüfung der DIR-Einträge nötig.
        DEFW #0000      ;OFF: kein (0) Systemspuren-Offset
;
LBL12: EQU $           ;128 Bytes für RECD11-Übertragungen
;-----
INTV:   EQU START+LBL1-BIOS ;Interrupt-Vektor in Bank 1b
DEVICE: EQU START+LBL2-BIOS ;springt die auszuführende IBIOS-Routine an
BIOSRET: EQU START+LBL3-BIOS ;springt ins BDOS zurück
BCSAVE: EQU START+LBL4-BIOS ;BC-Register Zwischenspeicher
ASAVE:   EQU START+LBL5-BIOS ;Akku-Zwischenspeicher
JUMP:    EQU START+LBL6-BIOS ;(Adresse) d. auszuführenden IBIOS-Routine
IOBYTE:  EQU START+LBL7-BIOS ;IO-Byte-Zwischenspeicher
INSTALL: EQU START+LBL8-BIOS ;Kaltstart-Installationen
XLT:     EQU START+LBL9-BIOS ;wg. Geschwindigkeit hier Skewing-Tab. übers.
RECD11:  EQU START+LBL10-BIOS;s.o.
DPHA:    EQU START+LBL11-BIOS;Disk-Parameter-Header in Bank1b
DPHB:    EQU DPHA+#10
DPHC:    EQU DPHB+#10
DPBA:    EQU START-#300      ;Disk-Parameter-Block Drive A in CP/M-Bank 1b
DPBB:    EQU START-#2E0      ;analog Drive B
DPBC:    EQU DPHC+#10       ;Disk-Parameter Block für RAM-Disk "C"
BUFF1:   EQU START+LBL12-BIOS;s.o.

```

Listing 7/2.5.3-1 (Teil 5)

## 2.5 256 kByte-Speicher für Schneider CPC

## Teil 7: Hard- und Software-Ergänzungen

```

;-----
;Schnittstelle zw. BIOS und IBIOS (s. auch $COFAH in ROM#7):
; leitet den Aufruf der eigtl. IBIOS-Routine ein (2. Registersatz sichern,
; RAM/ROM-Konfig. herstellen, Stack initialisieren).

IBIOS:  LD  A,(IOBYTE)  ;hinterlegtes IO-Byte
        LD  (#0003),A  ;eintragen
        LD  A,(ASAVE)
        LD  BC,(BCSAVE)
        EX  AF,AF'
        EXX
        LD  (STACK),SP ;Stack
        LD  SP,BIOS   ;initialisieren
        PUSH HL
        LD  HL,(JUMP)
        PUSH DE      ;alternativen Registersatz
        PUSH BC      ;retten
        PUSH AF
        PUSH IY
        CALL BANK0   ;Bank 0a,0b selektieren
        LD  BC,(GATEARRAY);Portadresse Gate Array u. ROM Konfiguration
        RES 3,C      ;U ROM (hier AMSDOS ROM #7)
        OUT (C),C    ;anschalten (s. KL U ROM Enable $4EF)
        OR  A
        LD  IY,#AC48 ;Basisadresse für Disk-RAM
        CALL JMP     ;die in (JUMP) eingetragene IBIOS-Routine ausf.
        DI
        EX  AF,AF'
        EXX
        SET 3,C      ;U ROM (hier AMSDOS ROM #7)
        OUT (C),C    ;abschalten (s. KL U ROM Disable $4F9)
        LD  (GATEARRAY),BC
        CALL BANK1B
        POP IY
        POP AF
        POP BC
        POP DE
        POP HL
        EXX
        EX  AF,AF'
        LD  SP,(STACK) ;alten Stackpointer wiederherstellen
        LD  (ASAVE),A  ;A und
        LD  (BCSAVE),BC ;BC zwischenspeichern (muß ohne Stack erfolgen)
        LD  BC,BANKPORT
        JP  BIOSRET   ;ins BIOS zurück.

```

Listing 7/2.5.3-1 (Teil 6)

## 2.5 256 kByte-Speicher für Schneider CPC

Teil 7: Hard- und Software-Ergänzungen

```

JMP:   PUSH HL           ;IBIOS-Routine in HL
      EXX
      EX  AF,AF'
      RET                ;via RET anspringen
;-----
WARMSTART:CALL SECTOUT   ;s. auch WARM BOOT-Routine in #7 ab $C22BH
      LD HL,BUFFA+RL     ;CP/M in Bank0a
      PUSH HL
      INC HL
      LD E,(HL)          ;JP-Adresse
      INC HL
      LD D,(HL)
      LD HL,#FCA4        ;Offset zum CCP-Start
      ADD HL,DE
      EX DE,HL           ;DE:= CCP-Entry
      POP HL
      CALL UROMOFF       ;AMSDOS-ROM#7 abschalten
      CALL BANK1B
      LD BC,#1600        ;11 Sektoren mit je 512 Bytes Länge
      LDIR               ;übertragen, DE:= BIOS-Entry
      LD SP,DPBA         ;Stack in Bank 1b
      PUSH DE
      LD HL,BIOS         ;BIOS-System
      LD DE,START        ;dorthin
      LD BC,IBIOS-BIOS   ;(Länge)
      LDIR               ;in Bank 1b übertragen
      POP DE
      LD HL,#EA00
      ADD HL,DE
      PUSH HL            ;HL:= CCP-Entry
      LD HL,#F206
      ADD HL,DE          ;HL:= BDOOS-Entry
      PUSH HL
      EXX
      LD (GATEARRAY),BC
      EXX
      JP INSTALL        ;Bank1b-Installation und Sprung in CCP
;-----
CONSTAT:LD HL,#C47D     ;(HL)=>CONSOLE STATUS Zuordnung
      JR LBL20
CONIN:  LD HL,#C486     ;analog s. oben
      JR LBL20
CONOUT: LD HL,#C48F
      JR LBL20
PRINTOUT:LD HL,#C4A1
      JR LBL20

```

Listing 7/2.5.3-1 (Teil 7)

## 2.5 256 kByte-Speicher für Schneider CPC

## Teil 7: Hard- und Software-Ergänzungen

```

PUNCHER: LD  HL,#C4AA
          JR  LBL20
READER:  LD  HL,#C4BC
          JR  LBL20
PRNSTAT: LD  HL,#C498
LBL20:   JP  #C46A      ;zu der in HL ausgewählten Tab. I/O Routine springen
;----- SELECT DRIVE (s. auch $C4F0H in #7)
SLDRV:   LD  A,C        ;Drivenr. in Akku
          LD  (DRIVE),A
          CP  2          ;RAM-Disk "C" ?
          LD  HL,DPHC
          RET Z          ;ja, ->
          LD  HL,#0000
          RET NC        ;gewünschtes Laufwerk nicht vorhanden
          LD  A,E        ;Purge-Flag (=0 bei Erstzugriff)
          RRA
          LD  E,C
          JR  C,NOLOGIN
          CALL IDREAD    ;ID lesen
          JR  C,DSFORMAT ;Doppelkopfformat ?
CPCFORM: CALL #C56C     ;CPC-Format ermitteln
          LD  HL,#0000
          RET NC        ;Drive nicht verfügbar!
          LD  A,(DRIVE)
          LD  E,A
          LD  A,#18      ;ehem. Login-Flag im DPB, spezifiziert DS-Formate
          CALL DPBHL
          LD  (HL),0     ;auf einseitige Formate zurücksetzen
NOLOGIN: BIT  0,E
          LD  HL,DPHA    ;DPH-Tabellenanfang "A"
          JR  Z,LBL21
          LD  HL,DPHB    ;DPH-Tabellenanfang "B"
LBL21:   PUSH HL
          LD  HL,DPBA    ;DPB in Bank1b
          JR  Z,LBL22
          LD  HL,DPBB
LBL22:   PUSH HL
          XOR  A
          CALL DPBHL     ;DPB in Bank0 des gew. Drives von HL
          LD  DE,BUFFA   ;nach BUFFA
          PUSH DE
          LD  BC,25
          PUSH BC
          LDIR           ;laden
          CALL BANK1B

```

Listing 7/2.5.3-1 (Teil 8)

## 2.5 256 kByte-Speicher für Schneider CPC

Teil 7: Hard- und Software-Ergänzungen

```

        POP BC          ;Länge holen
        POP HL          ;Start in Bank 0a holen
        POP DE          ;nach Bank 1b
        LDIR            ;übertragen
        CALL BANK0
        POP HL
        RET
;-- DPBs für Doppelkopfformate --
DSFORMAT:LD A,(#BE52) ;Sektorgröße beim ID-Lesen (1=256, 2=512 Bytes)
        DEC A
        LD HL,TADPB    ;DPB für TA-alphatronic PC-FORMAT
        JR Z,LBL23
        LD A,(#BE51)   ;Sektornr.
        CP 10          ;zweiseitiges IBM-Format (CP/M-86) ?
        JR C,CPCFORM   ;ja, (im CPC implementiert), ->
        LD HL,XDPB     ;DPB für XFORMAT
LBL23:  PUSH HL
        XOR A
        CALL DPBHL     ;Ziel nach HL
        LD A,E
        EX DE,HL
        POP HL
        LD BC,25
        LDIR
        LD E,A
        JR NOLOGIN
;400KB-Format auf 40Spur-DS-Laufwerk (396 KB netto)
XDPB:  DEFW 40         ;SPT: 40 Records/Track
        DEFB 04        ;BSH: Block: 2KB
        DEFB 15        ;BLM: "--
        DEFB 01        ;EXM: "--
        DEFW 199      ;DSM: maximale Blocknr.
        DEFW 127      ;DRM: 128 Directory-Einträge
        DEFW #00C0    ;AL01:2 Directory-Blöcke
        DEFW 32       ;CKS: Alle Directory-Einträge zu prüfen (32*4=128)
        DEFW 00       ;OFF: 0 reservierte Systemspuren
        DEFB 10       ;FSC: Nr. des 1.Sektors
        DEFB 10       ;PST: 10 Sektoren/Spur
        DEFB 25       ;GPS: Gap#3 bei I/O
        DEFB 48       ;GPT: Gap#3 beim Formatieren
        DEFB #E5      ;FLB: Füllbyte
        DEFB 02       ;BPS: 512 Bytes/Sektor (FDC-Code)
        DEFB 04       ;RPS: 4 Records/Sektor
        DEFS 2,0      ;Recalibrate / Spur 0
        DEFB %00000011 ;DS, Wechselschritt

```

Listing 7/2.5.3-1 (Teil 9)



## 2.5 256 kByte-Speicher für Schneider CPC

## Teil 7: Hard- und Software-Ergänzungen

```

;TA-alphatronic PC-Format: 296 KB netto, 256 DIR-Einträge
TADPB:  DEFB 32,00,04,15,01,152,000,255,000,#F0,#00,64,00,04,00
        DEFB 01,16,30,59,#E5,01,002,000,000,%00000011 ;Bedeutung s.o.
;-----
HOME:   CALL SECTOUT    ;ggf. eröffneten Sektor schreiben
        JP   #C522      ;Spur 0 eintragen
;-----
READRCD: LD  A,(DRIVE)
        CP  2           ;C ?
        JP  Z,RDRCD    ;Record aus RAM-Disk "C" holen
        CALL BUFFAB    ;IBIOS-Buffer
        CALL DSKRDRCD  ;DISK-READ RECORD
        PUSH AF
        CALL BUFFAB
        JR  Z,LBL25
        CALL UROMOFF
        CALL BANK1B    ;Sonderfall: Record im Bereich $7Fxx-$80xx
        PUSH DE        ;BUFFA in HL / Recordanfang in DE
        LD  DE,BUFF1   ;in den Zwischenspeicher in Bank 1b
        PUSH DE
        LD  BC,RL
        LDIR           ;übertragen
        POP HL         ;von dort nach
        POP DE        ;Recordanfang
        CALL RECD11    ;übertragen
        JR  LBL26
LBL25:  BIT  7,H
        CALL NZ,RECD01
        JR  NZ,LBL26
        CALL BANK1B
        LD  BC,RL      ;Record 128 Bytes lang
        LDIR           ;übertragen
LBL26:  CALL BANK0
        POP AF
        RET
;-----
WRITERCD:LD  A,(DRIVE)
        CP  2           ;C ?
        JP  Z,WRRCD    ;Record in RAM-Disk "C" schreiben
        PUSH BC
        CALL UROMOFF    ;der zu schreibende Record kann im RAM-Bereich
        CALL BUFFAB    ;$C000H - $FFFFH liegen
        EX  DE,HL
        JR  Z,LBL27
        CALL BANK1B    ;Sonderfall: Record im Bereich $7Fxx-$80xx
        PUSH DE        ;BUFFA in DE / Recordanfang in HL

```

Listing 7/2.5.3-1 (Teil 10)

## 2.5 256 kByte-Speicher für Schneider CPC

Teil 7: Hard- und Software-Ergänzungen

```

LD DE,BUFF1 ;in den Zwischenspeicher in Bank 1b
PUSH DE
CALL RECD11 ;übertragen
POP HL ;von dort nach
POP DE ;BUFFA
LD BC,RL
LDIR ;übertragen
CALL BANK0
JR LBL28
LBL27: BIT 7,H
CALL Z,RECD01
JR Z,LBL28
CALL BANK1B
LD BC,RL
LDIR
CALL BANK0
LBL28: POP BC
CALL UROMON
JF DSKWRRCD ;DISK-WRITE RECORD
;-----
BUFFAB: LD DE,(BUFF) ;angemeldeter Buffer nach DE
LD A,D
CP #7F
JR NZ,LBL29 ;Record liegt nicht im Bereich $7FxxH-$80xxH
LD A,E
CP #81
JR NC,LBL30 ;Record liegt ab $7F81+x (A=255)
LBL29: XOR A ;A=0
BIT 7,D ;Adressbit 15 = 0?
LD HL,BUFFB
JR Z,LBL31 ;Buffer so wählen, daß der Recd. einfacher übertragen
LBL30: LD HL,BUFFA ;werden kann (0a wenn Ziel/Quelle 1b, 0b <-> 1a)
LBL31: LD (#BE60),HL ;IBIOS-Recordbuffer
OR A ;HL=IBIOS-Recdbuffer, DE=CP/M-Record
RET ;Zeroflag gelöscht, wenn Spezialfall
;-----
RDRCDC: CALL BUFF34 ;Record aus RAM-Disk Bank 2 od. 3
JR LBL32 ;lesen (Carryflag gelöscht)
WRRCDC: CALL BUFF34 ;Record in RAM-Disk Bank 2 od. 3
EX DE,HL
SCF ;schreiben
LBL32: PUSH AF
CALL BANKXB
LD A,H
CP #7F
JR C,LBL35 ;Quellrecord im Bereich a (ohne $7F80+x - $80xx)

```

Listing 7/2.5.3-1 (Teil 11)

## 2.5 256 kByte-Speicher für Schneider CPC

## Teil 7: Hard- und Software-Ergänzungen

```
    JR  NZ,LBL33    ;Quellrecord im Bereich b
    LD  A,L
    CP  #81
    JR  C,LBL35    ;-> s.o.
    PUSH DE        ;Record aus Bank1 $7Fxx-$80xx nach BUFF1 bringen
    LD  DE,BUFF1   ;(kommt nur bei WRRDCDC vor, da RAM-Disk Records
    PUSH DE        ; $8000H nicht überqueren)
    CALL RECD11
    POP  HL
    POP  DE
LBL33: LD  BC,BUFFA
    CALL SUBLDIR   ;Record nach Bank 1a bringen
    LD  A,D
    CP  #7F
    JR  C,LBL34
    JR  NZ,LBL45
    LD  A,E
    CP  #81
    JR  C,LBL34
LBL44: CALL BANK1B   ;Ziel im Grenzbereich (kommt nur bei RORCD vor)
    LD  BC,BUFF1
    CALL SUBLDIR
    POP  AF
    JP  RECD11
LBL45: POP  AF        ;Ziel in Bank 2/3b
    CCF
    CALL BANKXB    ;Bank 1/2/3b auswählen
    LD  BC,RL
    LDIR
    XOR  A         ;kein I/O-Fehler
    RET
LBL34: CALL BANKO    ;Ziel in Bank 2/3a
    LD  BC,BUFFB
    CALL SUBLDIR
    POP  AF
    CCF
    JP  RECD123
LBL35: CALL BANKO    ;Quellrecord im Bereich a (ohne Grenzbereich)
    POP  AF
    PUSH AF
    PUSH DE
    LD  DE,BUFFB
    PUSH DE
    CALL RECD123   ;Nach BUFFB in Bank 0b holen
    POP  HL
```

Listing 7/2.5.3-1 (Teil 12)

## 2.5 256 kByte-Speicher für Schneider CPC

Teil 7: Hard- und Software-Ergänzungen

```

POP DE
LD A,D
CP #7F
JR C,LBL47 ;Ziel in Bereich a (ohne Grenzber.)
JR NZ,LBL46 ;Ziel in Bereich b
LD A,E
CP #81
JR C,LBL47
LD BC,BUFFA
CALL SUBDIR ;Zwischenübertragung nach BUFFA
JR LBL44
LBL46: LD BC,BUFFA
CALL SUBDIR ;nach Bank 0a bringen
POP AF
CCF
CALL BANKXB ;Bank 1/2/3b auswählen
LD BC,RL
LDIR
XOR A ;kein I/O Fehler (wenn die RAMs defekt sind,
RET ;läuft das ganze Programm nicht.)
LBL47: POP AF
CCF
JP RECD123
;-----
BUFF34: CALL UROMOFF
LD H,0 ;Recordadresse in RAM-Disk C errechnen
LD A,(#BE55) ;Recordnummer
LD L,A
LD B,7 ;2^7 Bytes/Record
LBL40: ADD HL,HL
DJNZ LBL40
PUSH HL
LD H,0
LD A,(#BE54) ;Tracknr.
LD L,A
XOR A
LD B,14 ;2^14 Bytes/Track
LBL41: ADD HL,HL
JR NC,LBL42 ;Bank 2 (kein Carry, A=0)
INC A ;Bank 3 (Carry, A=1)
LBL42: DJNZ LBL41
POP DE
ADD HL,DE ;Adresse des Records in d. RAM-Disk
LD DE,(BUFF) ;CP/M-Record
OR A ;Carry löschen, Zeroflag gem. Akku setzen
RET

```

Listing 7/2.5.3-1 (Teil 13)

## 2.5 256 kByte-Speicher für Schneider CPC

## Teil 7: Hard- und Software-Ergänzungen

```

;-----
SUBLDIR: PUSH DE      ;Record in Zwischenbuffer (in BC) übertragen
          LD  D,B
          LD  E,C
          PUSH DE
          LD  BC,RL
          LDIR
          POP HL
          POP DE
          RET

;-----
INSTBUF: LD  (BUFF),BC
          RET
BUFF:    DEFW #0000

;----- Hardware-spezifische Routinen
BANKXB:  JR  C,BANK1B  ;Carry -> Bank 1b
          OR  A
          LD  A,%10100000
          JR  Z,BANKX  ;A=0 -> Bank 2b
          SET 4,A
          JR  BANKX    ;A=1 -> Bank 3b
BANKO:   LD  A,0       ;Bank 0a,0b
          JR  BANKX
BANK1B:  LD  A,%10010000;Bank 0a,1b
BANKX:   DI          ;INTs verbieten
          LD  BC,BANKPORT;Bankswitch-Portadresse
          OUT (C),A   ;ausgeben
          RET

;-----
PRT8BIT: BIT 7,A
          JP  Z,#07F2  ;MC PRINT CHAR (im L ROM), da 8.Bit=0
          PUSH BC
          PUSH AF
          LD  BC,BANKPORT
          LD  A,1      ;D7 (8.Bit) des Druckerports
          OUT (C),A   ;über Bankswitch-Port ausgeben
          POP AF
          CALL #BDF1  ;MC WAIT PRINTER-Indirection
          POP BC
          RET          ;D7 wird beim Rücksprung ins CP/M gelöscht.

;-----
          DEFS #8000-$
;-----ab hier $8000H-----
RECD123: JR  C,RECD01
          OR  A        ;A=0 ->Bank 2a, A=1 -> Bank 3a, Carry -> Bank 1a
          LD  A,%01100000
          JR  Z,LBL50

```

Listing 7/2.5.3-1 (Teil 14)

## 2.5 256 kByte-Speicher für Schneider CPC

Teil 7: Hard- und Software-Ergänzungen

```

        SET 4,A
        JR  LBL50
RECD01: LD  A,%01010000;Bank 1a,0b selektieren
LBL50:  DI
        LD  BC,BANKPORT
        OUT (C),A
        LD  BC,RL
        LDIR                ;Record übertragen
        LD  BC,BANKPORT
        LD  A,0
        OUT (C),A
        RET

;-----Kaltstart-Installationen-----
KSINST: LD  SP,#C000    ;Stack nach Bank 0b legen
        LD  A,#EF      ;MC PRINT CHAR patchen (Opcode für RST 5)
        LD  (#BD2B),A
        LD  HL,PRT8BIT ;zeigt nun auf 8bit-Druckroutine
        LD  (#BD2C),HL
        LD  DE,#0040   ;Zeichensatztab. im RAM ab Zch. 64 (§)
        LD  HL,#9200   ;nach $9200H
        CALL #BBAB     ;TXT SET M TABLE
        LD  HL,SYMBTAB ;8 deutsche Sonderzeichen
        LD  A,(HL)     ;ASCII-Nr.
LBL51:  INC  HL
        CALL #BBAB     ;TXT SET MATRIX
        LD  A,(HL)
        OR  A
        JR  NZ,LBL51
        LD  HL,#0000
        LD  BC,#0100
        LD  DE,BUFFB
        PUSH BC
        PUSH HL
        PUSH DE
        LDIR                ;in Buffer 0b
        POP HL
        POP DE
        DI
        LD  BC,BANKPORT
        LD  A,%01010000;Bank 1a,0b
        OUT (C),A
        POP BC
        LDIR                ;Zeropage in Bank 1a übertragen
        LD  HL,MSG1     ;BankBIOS-Meldung / Directory Drive C löschen ?
        CALL MSG        ;ausgeben
        JR  NC,LBL53    ;nicht löschen

```

Listing 7/2.5.3-1 (Teil 15)

## 2.5 256 kByte-Speicher für Schneider CPC

## Teil 7: Hard- und Software-Ergänzungen

```

        DI          ;Directory "C" löschen
        LD BC,BANKPORT
        LD A,%01100000;Bank 2a,0b
        OUT (C),A
        LD HL,#0000
        LD (HL),#E5 ;Füllbyte
        LD DE,#0001
        LD BC,#400 ;bis $400H
        LDIR
LBL53:  LD BC,BANKPORT
        XOR A
        OUT (C),A ;Bank 0ab
        LD SP,BIOS
; CP/M 2.2 laden
BOOT:  LD HL,BUFFA+RL;ab #180 ablegen
        LD BC,#0B48 ;11 Sektoren ab Nr. #48
        LD DE,#0000 ;Track 0, Drive A
        PUSH HL
        CALL #C299 ;gewünschte Sektoren laden
        POP HL
        CALL C,#C2AC ;Sektor leer ?
        JR NC,ERROR ;Fehler
        JP WARMSTART ;->
;---CP/M nicht ladbar---
ERROR:  LD HL,MSG2 ;Fehlermeldung
        CALL MSG ;ausgeben und Wdh. abfragen
        JR C,BOOT ;"Y", -> Wiederholungsversuch
        RST 5
        DEFW #0000 ;CPC-Reset
;---- Meldung ausgeben und Y/N -Abfrage ----
MSG:   LD A,(HL) ;Meldung ausgeben
        INC HL
        CALL #BB5A ;TXT OUTPUT
        OR A
        JR NZ,MSG
LBL52: CALL #BB81 ;TXT CURSOR ON
        CALL #BB06 ;KM WAIT CHAR
        CALL #BB84 ;TXT CURSOR OFF
        AND %11011111 ;Groß-/Kleinschreibung ignorieren
        CP #4E ;"N" ?
        JP Z,#BB5A
        CP #59 ;"Y" ?
        JR NZ,LBL52 ;Weder "Y" noch "N", ->nochmal
        SCF
        JP #BB5A
;-----

```

Listing 7/2.5.3-1 (Teil 16)

## 2.5 256 kByte-Speicher für Schneider CPC

Teil 7: Hard- und Software-Ergänzungen

```
MSG1:  DEFM "^M^J Ready for CP/M 2.2 incl. ** 58.5 KB ** TPA^M^J^J"
        DEFM "C>ERA *.*^M^JALL (Y/N)?"
        DEFB 0
MSG2:  DEFM "^M^J CP/M 2.2 nicht ladbar.^M^JWiederholungversuch, evtl. mit"
        DEFM " anderer Diskette (Y/N) ?^M^J"
        DEFB 0

SYMBTAB:

DEFB 64      ;§
DEFB %00111100
DEFB %01100000
DEFB %00111100
DEFB %01100110
DEFB %00111100
DEFB %00000110
DEFB %00111100,0
DEFB 91      ;Ä
DEFB %01100110
DEFB %00011000
DEFB %00111100
DEFB %01100110
DEFB %01111110
DEFB %01100110
DEFB %01100110,0
DEFB 92      ;Û
DEFB %01100110
DEFB %00111000
DEFB %01101100
DEFB %11000110
DEFB %11000110
DEFB %01101100
DEFB %00111000,0
DEFB 93      ;Ü
DEFB %01101100
DEFB %11000110
DEFB %11000110
DEFB %11000110
DEFB %11000110
DEFB %11000110
DEFB %01111100,0
DEFB 94      ;^
DEFB %00111000
DEFB %01101100
DEFB %11000110,0,0,0,0,0
DEFB 123     ;ä
DEFB %01101100
```

Listing 7/2.5.3-1 (Teil 17)



## 2.5 256 kByte-Speicher für Schneider CPC

## Teil 7: Hard- und Software-Ergänzungen

```
DEFB %00000000
DEFB %01111000
DEFB %00001100
DEFB %01111100
DEFB %11001100
DEFB %01110110,0
DEFB 124      ;ö
DEFB %01101100
DEFB %00000000
DEFB %00111100
DEFB %01100110
DEFB %01100110
DEFB %01100110
DEFB %00111100,0
DEFB 125      ;ü
DEFB %01100110
DEFB %00000000
DEFB %01100110
DEFB %01100110
DEFB %01100110
DEFB %01100110
DEFB %00111110,0
DEFB 126      ;ß
DEFB %00111100
DEFB %01100110
DEFB %01100110
DEFB %01111100
DEFB %01100110
DEFB %01100110
DEFB %01111100
DEFB %01100000
DEFB 0        ;Tabellenende
;-----
CAT DSBIOS.103 ;BIOS für Doppelkopflaufwerke nachladen
```

Listing 7/2.5.3-1 (Teil 18)

## 2.5 256 kByte-Speicher für Schneider CPC

Teil 7: Hard- und Software-Ergänzungen

## „DSBIOS.103”

```

;*****
;* DSBIOS vers. 1.03          *
;*   written July 1987      *
;*   by Kai Meier, Hauptstr.25, *
;*     2361 Bebensee, W.-Germany *
;* performs                 *
;* I/O-Operations on Double-Head *
;* Drives (40/80 Tracks in MFM) *
;*****
;* Overlay Program on - BANKBIOS - *
;*   version 1.44 or better      *
;*****
;
FDCPORT: EQU #FB7E
FDCDRV:  DEFB 0
;
DSKRDRCD: XOR A          ;IBIOS-Routine: Read Record (#C54C, ROM#7)
          LD  (#BE59),A
          CALL SECTOFREC;Sektor aus Recordnr.bestimmen u. evtl. leser
          CALL #C8C7      ;Record in Buffer übertragen
          RET NC
          XOR A          ;kein Fehler
          RET
SECTOFREC:PUSH AF
          CALL #C854      ;Sektor lesen oder Record im Recd-Buffer ?
          JP  C,#C851     ;alles klar.
          CALL SECTOUT
          POP BC          ;Stack für Fehlerfall parat machen.
          RET NC          ;-> FEHLER
          PUSH BC
          CALL #C880      ;prüft auf Sektor Ende
          POP AF
          JR  C,RECDIMBUF;Record im Sektor-Buffer
          CALL #C8A2      ;Sektornr. errechnen (c=Nr.)
          CALL SECTRD     ;Sektor lesen
RECDIMBUF:JP #C84A      ;Abschluss od. Fehlerbehandlung
;-----
DSKWRRCD: PUSH BC          ;IBIOS-Routine: Write Record (#C52E, ROM#7)
          LD  A,C          ;Deblocking-Code

```

Listing 7/2.5.3-2 (Teil 1)

## 2.5 256 kByte-Speicher für Schneider CPC

## Teil 7: Hard- und Software-Ergänzungen

```

CP 2
CALL Z,#C7EB ;keine "preread Operation" nötig
CALL #C800 ;Test, ob Drive, Track u. Recd. unverändert
CALL C,#C81B ;sind gleich ->
CALL SECTOFREC ;Sektor aus Recordnr. errechnen
POP BC
RET NC
CALL #C8B6 ;überträgt Record in Sektorbuffer
DEC C ;Anzahl Records -1
SCF
CALL Z,SECTOUT ;Sektoroffset addieren und S. schreiben
RET NC ;Fehler
XOR A
RET
SECTOUT: LD HL,#BE5E ;Sektor lesen OK-Flag löschen
LD (HL),0
DEC HL
LD A,(HL) ;a = WRITE-Sektor-Flag
OR A
SCF
RET Z ;Flag ist 0, nicht schreiben ->
INC (HL)
CALL #C8A2 ;Sektornr. berechnen
SECTWR: CALL ALGO ;ggf. jubeln wir dem DOS einen anderen Sektor unter
LD A,#45 ;FDC-Opcode Sektor schreiben
CALL SECTIO ;jetzt gehts los
LD A,(#BE48)
LBL59: DEC A
INC BC
INC BC
INC BC
JR NZ,LBL59
RET
SECTRD: CALL ALGO ;Umrechnung der Sektor-/Spurnr. bei DS-Format
LD A,#66 ;FDC-Op-Code: Sektor lesen
SECTIO: LD (#BE62),HL ;Sektor I/O Buffer
LD H,A
LD L,C ;Sektornr.
LD (#BE74),HL ;sichern
CALL #C976 ;Motor an, inkl. Vor/Nachlaufzeit
LD C,10 ;10 Leseversuche
LD HL,FDCPROGR;FDC programmieren
LD A,E
LD (FDCDRV),A ;FDC-Code: Head/Unit Select
RES 2,E ;Seite 0 setzen (nötig für AMSDOS)
JP #C6FF ;Track in d suchen, CALL HL.

```

Listing 7/2.5.3-2 (Teil 2)

## 2.5 256 kByte-Speicher für Schneider CPC

Teil 7: Hard- und Software-Ergänzungen

```

FDCPROGR: LD HL, (#BE74) ;FDC-Op-Code und Sektornr. holen
          LD A, (FDCDRV)
          LD E, A ;HD/US-Signal zurückholen
          LD A, #18 ;LOGIN-Flag
          CALL DPBACC ;holen
          BIT 3, A ;auf Doppelschritt testen
          JR Z, STEP1 ;nein
          SRL D ;Doppelschritt, daher Track/2=Track im ID
STEP1: LD BC, FDCPORT ;FDC-Statusregister
       LD A, H ;Opcode
       CALL #C95C ;an FDC ausgeben
       LD A, E ;Head/Drive Select
       CALL #C95C ;an FDC ausgeben
       LD A, D ;Tracknr.
       CALL #C95C ;an FDC ausgeben
       XOR A ;A=0
       BIT 2, E ;Kopfnummer testen
       JR Z, LBL60 ;=0, dann ->
       INC A ;Kopfnr.=1
LBL60: CALL #C95C ;an FDC ausgeben
       LD A, L ;Sektornr.
       CALL #C95C ;an FDC ausgeben
       LD A, #14 ;Bytes/Sektor (DPB+#14)
       CALL DPBACC ;aus dem DPB in a
       OR A
       PUSH AF
BPS: CALL #C95C
     LD A, L ;Sektornr. als letzten Sektor
     CALL #C95C
     LD A, #11 ;GAP#3 bei RD/WR
     CALL DPBACC ;aus dem DPB in a
     CALL #C95C ;s.o.
     RES 2, E ;Seite 0 simulieren
     POP AF
     JP NZ, #C6BF ;Execution & Result Phase f. 256/512 Byte-S.
     LD A, #80
     JP #C6C1 ; --- für 128 Byte kleine Sektoren
;-----
DPBACC: PUSH HL ;DPB-Wert holen
       .CALL DPBHL
       LD A, (HL) ;Akku mit gewünschtem Byte laden
       POP HL
       RET
DPBHL: PUSH DE ;DPB-Adresse inkl. Offset ermitteln
       LD HL, (#BE42) ;Zeiger auf DPB A
       RR E ;Carry bei Drive 0

```

Listing 7/2.5.3-2 (Teil 3)

## 2.5 256 kByte-Speicher für Schneider CPC

## Teil 7: Hard- und Software-Ergänzungen

```

        LD  DE,#40      ;64
        JR  NC,LBL61
        ADD HL,DE       ;Offset für Drive B
LBL61:  LD  E,A         ;Akku zeigt auf gewünschtes Byte
        ADD HL,DE      ;d=0
        POP DE
        RET

:-----
ALGO:   LD  A,#18      ;ehem. LOGIN-Flag
        CALL DPBACC    ;in a
        RRA
        JR  NC,DSTEPCHK;einseitiges Format (evtl. Doppelschritt)
        RRA
        JR  NC,CPM86V  ;Spur in D nach dem CP/M86 Verfahren berechnen
        RRA           ;hier nach dem MS-DOS-Verfahren berechnen
        RRA           ;40/80 Spuren:hier nicht von Bedeutung
        PUSH AF       ;Doppelschritt-Flag sichern
        SRL D         ;Track/2
        JR  NC,LBL62   ;kein Carry, -> Seite 0
        SET 2,E       ;Seite 1
LBL62:  POP AF
        JR  DSTEPTTEST
DSTEPCHK: RRA
        RRA
DSTEPASK: RRA
DSTEPTTEST:RET NC     ;kein "double-step"
        RLC D         ;Track*2
        RET
CPM86V:  RRA
        LD  B,A       ;Bit0 gibt Einzel-/Doppelschritt an
        LD  A,40      ;40 Spuren
        JR  NC,LBL63
        ADD A,A       ;80 Spuren
LBL63:  DEC A         ;-1, da äußerste Spur mit 0 adressiert wird
        CP  D         ;gewünschte Spur
        JR  NC,LBL64  ;Track d noch auf Seite 0
        PUSH AF      ;max. Spurnr. sichern
        SET 2,E       ;Seite 1 selektieren
        ADD A,A       ;zu errechnende Spur:
        SUB A,D       ;a-(d-a) +1 = 2*a-d +1
        INC A
        LD  D,A       ;neue Tracknummer nach d
        POP AF       ;max. Spurnr. holen
        CP  D         ;Track in d > max. Spurnr.,
        LD  A,B
        JR  NC,DSTEPASK;nein (OK.) ->

```

Listing 7/2.5.3-2 (Teil 4)

## 2.5 256 kByte-Speicher für Schneider CPC

Teil 7: Hard- und Software-Ergänzungen

```

        POP HL          ;vorherige RET-Adresse freilegen
        CCF             ;nicht behebbaren Fehler
        RET            ;ans BDOS melden
LBL64:  LD  A,B
        JR  DSTEPASK.

;-----
IDREAD: CALL #C976     ;Motor an
        LD  A,#16
        CALL DPBACC
        SET 2,E        ;Seite 1
        LD  D,A        ;Tracknr.
        LD  C,#10     ;Leseversuche
        LD  HL,#C55D  ;ID lesen
        CALL #C6FF
        RES 2,E        ;Seite 0
        RET NC        ;Fehler
        LD  A,(#BE50) ;Kopfadr.
        DEC A
        RET Z        ;ok.
        CCF
        RET          ;falsche Kopfadresse

;----- obiges Programm auf Diskette (im CPC-System-Format) schreiben -----
SYSINIT: LD  C,#41     ;Bootsektor
        LD  DE,#0000  ;Track 0, Drive A
        LD  HL,BUFFB
        PUSH BC
        PUSH DE
        PUSH HL
        CALL RDSEC    ;lesen
        POP DE       ;BUFFA
        PUSH DE
        LD  HL,BOOTP  ;Bootsektor-Patch Routine
        LD  BC,RDSEC-BOOTP;Länge
        LDIR         ;in Bootsektor eintragen
        POP HL
        POP DE
        POP BC
        CALL WRSEC    ;u. schreiben

        LD  HL,KALTSTART
        LD  C,#43     ;erster leerer Sektor nach dem Konfig.-Sektor
        LD  B,5       ;5 Sektoren zu schreiben
LBL54:  PUSH BC
        CALL WRSEC    ;Das Programm ab KALTSTART speichern
        INC H         ;Buffer +200H
        INC H

```

Listing 7/2.5.3-2 (Teil 5)

## 2.5 256 kByte-Speicher für Schneider CPC

## Teil 7: Hard- und Software-Ergänzungen

```
POP BC
INC C           ;Sektornr. +1
DJNZ LBL54
RET

;-----
BOOTP: LD BC,#0543 ;die 5 (sonst leeren) S. nach Konfig.-Sektor
LD HL,KALTSTART ;nach KALTSTART
LD DE,#0000 ;auf Drive A, Spur 0
CALL #C299 ;laden
JP C,KALTSTART ;alles ok., ->
JP NC,#C224 ;Lesefehler ->
CALL #BE89 ;Konfigurations-Sektor lesen
JR NC,RDSEC ;K.-Sektor nicht lesbar, Fehler ->
CALL #150 ;Routinen im Bootsektor (Keys, IO-Byte usw.)
JP BOOTSRET

;-----
RDSEC: RST 3 ;Sektor lesen
DEFW LBL55
RET
LBL55: DEFW #C666
DEFB #07 ;ROM#7 (AMSDOS)

WRSEC: RST 3 ;Sektor schreiben
DEFW LBL56
RET
LBL56: DEFW #C64E
DEFB #07

;-----
END
```

Listing 7/2.5.3-2 (Teil 6)

## 2.5 256 kByte-Speicher für Schneider CPC

Teil 7: Hard- und Software-Ergänzungen

## Diskparameter bearbeiten „DISKFORM.PT“

```

100 '*****
105 '**
110 '** Multi-CP/M-Format Installation Tool: **
115 '** Individual Disk-Formats for **
120 '** BankBIOS (vers. 1.44 or better) **
125 '**
130 '** written July-September 1987 by **
135 '**
140 '** Kai Meier, Hauptstr. 25, **
145 '** 2361 Bebensee **
150 '*****
155 :
160 MEMORY &7BAF:DEFINT a-w,z:MODE ?
165 DIM FPM(25,30):' Parameter des Disk Parameter Blocks f. 30 Formate
170 DIM FORMAT$(30):' Namen der Formate
175 :
180 PRINT:PRINT
185 PRINT,CHR$(24)" * * CP/M - Diskformat - Programming Tool * * "CHR$(24)
190 LOCATE 2,10
195 INPUT"Mit welchem Laufwerk soll gearbeitet werden (A/B) ->",Drive$
200 Drive$=UPPER$(Drive$):IF Drive$<>"A"AND Drive$<>"B"THEN 190
205 PRINT
210 PRINT" Wieviele Seiten (1/2),":PRINT" wieviele Spuren (40/80)"
215 LOCATE 30,14:INPUT"->",Sides,Tracks
220 IF Sides<1 OR Sides>2 OR(Tracks<>40 AND Tracks<>80)THEN 215
225 PRINT:PRINT" Drive "CHR$(24)" "Drive$" "CHR$(24)" ":"Sides"Seite(n) *"Tracks"Spuren"
230 PRINT:INPUT" Ok. (J/N) ";yn$:IF UPPER$(yn$)<>"J"THEN 190
235 :
240 FOR z1=1 TO 2*Sides+1:READ format$(z1)
245 FOR z2=1 TO 24:READ v$:FPM(z2,z1)=VAL("&"+v$):NEXT z2
250 READ v$:FPM(z2,z1)=VAL("&"+v$)OR (Tracks-40)/5:NEXT z1
255 IF Tracks=40 THEN 355
260 RESTORE 330:FOR z3=z1 TO z1+Sides-1:READ format$(z3)
265 FOR z2=1 TO 25:READ v$:FPM(z2,z1)=VAL("&"+v$):NEXT z2:NEXT z3:z1=z3
270 :
275 :'DPB-Parameter (Bedeutungen s. Quelltext und Dokumentation)
280 DATA"CPC Sys", 24,00,03,07,00,AA,00,3F,00,C0,00,10,00,02,00
285 DATA 41,09,2A,52,E5,02,04,00,00,00
290 DATA"CPC Data", 24,00,03,07,00,B3,00,3F,00,C0,00,10,00,00,00

```

Listing 7/2.5.3-3 (Teil 1)



## 2.5 256 kByte-Speicher für Schneider CPC

## Teil 7: Hard- und Software-Ergänzungen

```

295 DATA          C1,09,2A,52,E5,02,04,00,00,00
300 DATA CP/M86-IBM,20,00,03,07,00,9B,00,3F,00,C0,00,10,00,01,00
305 DATA          01,08,2A,50,E5,02,04,00,00,00
310 DATA"TA alpha.",20,00,04,0F,01,97,00,FF,00,F0,00,40,00,04,00
315 DATA          01,10,1E,3B,E5,01,02,00,00,03
320 DATA 2X40     ,28,00,04,0F,01,C7,00,7F,00,C0,00,20,00,00,00
325 DATA          0A,0A,19,30,E5,02,04,00,00,03
330 DATA 1X80     ,28,00,04,0F,01,C7,00,7F,00,C0,00,20,00,00,00
335 DATA          0A,0A,19,30,E5,02,04,00,00,00
340 DATA 2X80     ,28,00,04,0F,01,8E,01,FF,00,F0,00,40,00,00,00
345 DATA          0A,0A,19,30,E5,02,04,00,00,03
350 :
355 XDPB=PEEK(&BE42)+PEEK(&BE43)*256+64*SGN(ASC(Drive$)-65)
360 Format=1:Formatprog=z1:Fremdfmt256=5:Fremdfmt512=z1-1
365 :'Formatier- und ID read- Routinen
370 RESTORE 380
375 FOR z=&9000 TO &9041:READ v$:POKE z,VAL("&"+v$):NEXT
380 DATA 00,00,00,00,00,00,00,00,00,00,DF,0D,90,C9,10,90,07
385 DATA ED,5B,01,90,CD,66,C7,01,7E,FB,3E,4A,CD,5C,C9,3A
390 DATA 01,90,F3,CD,5C,C9,CD,1C,C9,FB,C9,00,00,00,00,00
395 DATA 21,01,90,5E,23,56,23,4E,21,00,A0,DF,3F,90,C9,52
400 DATA C6,07
405 :'BankBIOS (V1.45) und DSBIOS 'reinladen
410 LOAD"BB145.BIN"
415 :
425 CLS:LOCATE 10,7:PRINT" * * * Hauptmenue * * *":PRINT
430 PRINT,Drive$"> "Format$(Format)"-Format":PRINT
435 PRINT,"1. Sektor-ID einer beliebigen Disk. nach IBM34-Standard // Status"
440 PRINT,"2. Format auswaehlen":PRINT,"3. Format programmieren (Fremdformate)"
445 PRINT,"4. Diskette formatieren"
450 PRINT,"5. BankBIOS/DSBIOS: Fremdformate installieren."
455 PRINT,"6. BankBIOS 1.45 & DSBIOS 1.03 installieren":PRINT,"7. Ende"
460 :
465 LOCATE 17,20:INPUT"Ihre Wahl (Nr.)   ->",>z:IF z<1 OR z>7 THEN 465
470 CLS:LOCATE 10,10:ON z GOSUB 510,625,675,985,1115,1300,480
475 GOTO 425
480 END
485 :
490 '-----
495 ' ID lesen/Status
500 '-----
505 :
510 PRINT" * * ID - Lesen // Statusinformation * *":PRINT
515 INPUT"Auf welcher Spur ID lesen ",track
520 IF track<0 OR track>=Tracks THEN 515
525 IF sides=2 THEN INPUT"Welche Seite (0/1)   ";side

```

Listing 7/2.5.3-3 (Teil 2)

## 2.5 256 kByte-Speicher für Schneider CPC

Teil 7: Hard- und Software-Ergänzungen

```

530 IF side<0 OR side>1 THEN 525
535 POKE &9002,track
540 PRINT:PRINT"Bitte Disk einlegen und eine Taste druecken ":CALL &8806
545 POKE &9001,side*4+ASC(Drive$)-65:OUT &FA7E,1:CALL &9009:OUT &FA7E,0
550 PRINT:PRINT
555 PRINT"Status 0/1/2: ";
560 PRINT BIN$(PEEK(&BE4C),8),BIN$(PEEK(&BE4D),8),BIN$(PEEK(&BE4E),8)
565 IF(PEEK(&BE4C)AND &X11000)>0 THEN PRINT"Drive nicht Ready.":GOTO 590
570 IF(PEEK(&BE4D)AND &X101)>0 THEN PRINT"ID nicht lesbar oder Disk unformatiert.":GOTO 590
575 PRINT:PRINT" Spur : ** "PEEK(&BE4F)" ***"
580 PRINT:PRINT" Kopf : ** "PEEK(&BE50)" ***"
585 PRINT:PRINT" Sektor - Nr. // Groesse : "PEEK(&BE51)" // "PEEK(&BE52)
590 PRINT:INPUT"Nach ein ID lesen (J/ )";yn$:IF UPPER$(yn$)="J"THEN 515
595 RETURN
600 :
605 '-----
610 ' Format auswaehlen
615 '-----
620 :
625 PRINT" * * * Format - Auswaehlen * * *":PRINT:PRINT
630 PRINT:PRINT"Installierte Formate":z=1
635 WHILE Format$(z)<>"":PRINT Z". "Format$(z)"-Format",:z=z+1:WEND
640 PRINT:PRINT:PRINT,:INPUT"Welches Format (Nr.) ->",Format
645 IF Format$(format)=" "THEN format=1
646 RETURN
650 :
655 '-----
660 ' Format programmieren
665 '-----
670 :
675 LOCATE 6,1:PRINT" * * * Format - Programmieren * * *":PRINT:PRINT
680 Z=Formatprog:Formatprog=Formatprog+1
685 DS=0:dstep=1:IF Sides=1 AND Tracks=80 THEN 715 ELSE IF Sides=1 THEN 730
690 PRINT "Ist das Format doppelseitig aufgebaut ?"
695 PRINT,"0. Nein.":PRINT,"1. JA, nach dem CP/M-86 Verfahren."
700 PRINT,"2. JA, nach dem Wechselschritt-Verfahren.":PRINT:PRINT
705 INPUT" (Nr.) ->",DS:IF DS<0 OR DS>2 THEN 705 ELSE PRINT:PRINT
710 :
715 IF TRACKS=80 THEN INPUT" Handelt es sich um ein 40-Spur Format (J/ ) ";A$
720 IF UPPER$(a$)="J"THEN dstep=2 ELSE dstep=1
725 :
730 PRINT:PRINT"Derzeit aktives Format: "Format$(Format)"-Format"
735 PRINT:PRINT
740 INPUT"Sektoren / Spur ->",PST
745 INPUT"Bytes / Sektor (128/256/512) ->",BPS
750 INPUT"Systemspuren (eine Seite) ->",TSYS

```

Listing 7/2.5.3-3 (Teil 3)

## 2.5 256 kByte-Speicher für Schneider CPC

## Teil 7: Hard- und Software-Ergänzungen

```

755 INPUT"Directory-Eintraege max.      ->",DIR
760 INPUT"Blockgroesse in KB          ->",BLK
765 INPUT"NR. des 1.SEKTORS           ->",FSC
780 :
785 ' Errechnung der CP/M-Parameter des DPB
790 FPM(1,Z)=BPS*PST/128:'           Records pro Spur (SPT)
795 FPM(2,Z)=0
800 FPM(3,Z)=LOG(BLK*8)/LOG(2):'     Block Shift (BSH)
805 FPM(4,Z)=2^FPM(3,Z)-1:'         Block Mask (BLM)
810 DSM=(Tracks/dstep-Tsys)*(SGN(DS)+1)*FPM(1,Z)/8/BLK-1
815 FPM(5,Z)=INT(INT(FPM(4,Z)/8)/INT(DSM/256+1))
820 'DSM: Kapazitaet in Bloecken -1 ^Extend Maske(EXM)
825 FPM(6,Z)=DSM MOD 256
830 FPM(7,Z)=INT(DSM/256):'         DSM
835 FPM(8,Z)=(DIR-1)MOD 256
840 FPM(9,Z)=INT((DIR-1)/256):'     Directory-Eintraege (DRM)
845 AL01=2^16-2^(16-(DIR/32/BLK))
850 FPM(10,Z)=INT(AL01/256):'       binaer codierte Directoryblock-
855 FPM(11,Z)=AL01-INT(AL01/256)*256:' belegungstabelle (ALO, AL1)
860 FPM(12,Z)=(DIR/4)MOD 256
865 FPM(13,Z)=INT(DIR/1024):'       Gr. der Pruefsummentabelle (CKS)
870 FPM(14,Z)=TSYS*(SGN(DS)+1)
875 FPM(15,Z)=0:'                 reservierte Systemspuren (OFF)
880 :
885 ' CPC-spezifische DPB-Parameter
886 GPT=INT((6000-(BPS+64)*PST)/PST)
887 GPS=GPT-25:IF GPS<0 THEN GPS=INT(GPT/3)+1
890 FPM(16,Z)=FSC:'
895 FPM(17,Z)=PST:'                 Physikalische Sektoren/Spur
900 FPM(18,Z)=GPS:'                 Gap#3 bei I/O
905 FPM(19,Z)=GPT:'                 Gap#3 beim Formatieren
910 FPM(20,Z)=&E5:'                 Fuellbyte
915 FPM(21,Z)=INT(BPS/256):'Bytes/Sektor in FDC-Code: 0/1/2<->128/256/512
920 FPM(22,Z)=BPS/128:'             Records/Sektor
925 FPM(23,Z)=0:'                 Recalibrate Flag 1x gesetzt
930 FPM(24,Z)=0:'                 akt. Spur-Speicher
935 FPM(25,Z)=SGN(DS)OR DS OR Tracks/dstep/40 OR (dstep-1)*8
940 :
945 PRINT:PRINT,:"INPUT"Alle Angaben richtig (J/ ) ";A$:A$=UPPER$(A$)
950 IF A$<>"J"THEN PRINT:PRINT:GOTO 685
955 PRINT:PRINT:INPUT"Wie heisst das Format ->",FORMAT$(Z)
960 IF Format$(Z)=""THEN 955 ELSE RETURN
965 :
970 '=====
975 ' DISKETTE FORMATIEREN
980 '=====
985 :

```

Listing 7/2.5.3-3 (Teil 4)

## 2.5 256 kByte-Speicher für Schneider CPC

Teil 7: Hard- und Software-Ergänzungen

```

990 PRINT" * * * Diskette formatieren * * *":PRINT:PRINT
995 FOR z=1 TO 21:POKE z+XDPB-1,fpm(z,Format):NEXT
1000 PRINT:PRINT"Eingeschaltetes Format : "FORMAT$(Format):PRINT
1005 INPUT"Anfangstrack  ->",ATR:IF ATR<0 OR ATR>Tracks-1 THEN 1005
1010 INPUT"Endtrack (ENTER bei vollst. Formatierung) ->",ETR
1015 IF ETR<0 OR ETR>Tracks THEN 1010 ELSE IF ETR<>0 AND ETR<ATR THEN 1005
1020 IF ETR=0 THEN ETR=Tracks-1
1025 PRINT:PRINT" Bitte die zu formatierende Diskette in Drive "CHR$(24)" ";
1030 PRINT Drive$" "CHR$(24)" einlegen":PRINT:PRINT:PRINT" und eine Taste druecken..."
1035 CALL &8B06:POKE &9003,FPM(16,Format)
1040 DS=FPM(25,Format)AND 1:dstep=(FPM(25,Format)AND &X1000)/8+1
1045 FOR track=ATR TO ETR:POKE &9002,track*dstep
1050 FOR side=0 TO SGN(DS):POKE &9001,4*side+ASC(Drive$)-65
1055 z=&A000:y=FPM(16,Format):FOR I=0 TO FPM(17,Format)-1
1060   POKE z,track:POKE z+1,side:POKE z+2,y:POKE z+3,FPM(21,Format)
1065   y=y+2:IF y>FPM(16,Format)+FPM(17,Format)-1 THEN y=FPM(16,Format)+1
1070   z=z+4
1075 NEXT
1080 CALL &9030:NEXT side:NEXT track
1085 PRINT:INPUT"Noch eine Diskette formatieren (J/ ) ";yn$
1090 IF UPPER$(yn$)="J"THEN 1000 ELSE RETURN
1095 :
1100 '=====
1105 ' BankBIOS/DSBIOS: Fremdformate installieren
1110 '=====
1115 PRINT"* * * BankBIOS/DSBIOS: Fremdformate installieren * * *":PRINT:PRINT
1120 :Z1=Format
1125 PRINT:PRINT:PRINT"Welches Fremdformat mit 512 Bytes/Sektor ?":GOSUB 625
1130 IF FPM(21,Format)<>2 THEN 1125
1135 FOR z=1 TO 25:POKE &7E16+z,FPM(z,Format):NEXT
1160 PRINT:PRINT:PRINT"Welches Fremdformat mit 256 Bytes/Sektor ?":GOSUB 625
1170 FOR z=1 TO 25:POKE &7E2F+z,FPM(z,Format):NEXT
1180 :Format=Z1:RETURN
1200 '=====
1210 ' BankBIOS/DSBIOS auf Disk`im System-Format
1220 '=====
1300 PRINT" * * * BankBIOS 1.45/DSBIOS 1.03 installieren * * *":PRINT:PRINT
1310 :
1320 PRINT" Bitte die betreffende Diskette in Drive "CHR$(24)" A "CHR$(24);
1330 PRINT" einlegen":PRINT:PRINT" und eine Taste druecken..."
1331 IF FPM(25,Format)AND 1=1 THEN 1334
1332 POKE &7E01,64:POKE &7E02,&30:POKE &82E2,0:POKE &82E3,0:POKE &82F4,&C9
1333 GOTO 1335
1334 POKE &7E01,10:POKE &7E02,&38:POKE &82E2,&CB:POKE &82E3,&D3:POKE &82F4,&CB
1335 CALL &8B06:CALL &82F7
1340 PRINT:PRINT:INPUT"Noch eine Diskette patchen (J/ ) ";yn$
1350 IF UPPER$(yn$)="J"THEN 1320 ELSE RETURN
1360 '15.9.87, 19.15

```

Listing 7/2.5.3-3 (Teil 5)

## 2.5 256 kByte-Speicher für Schneider CPC

## Teil 7: Hard- und Software-Ergänzungen

**Basic-Lader Bank BIOS „BANKBIOS.BAS“**

```

100 '*****
105 '* *
110 '* BankBIOS - Version 1.45 - *
115 '* *
120 '* geschrieben September 87 *
125 '* von Kai Meier *
130 '* Hauptstrasse 25 *
135 '* 2361 Bebensee *
140 '*****
145 :
150 MODE 2:LOCATE 10,10:PRINT"BankBIOS ** Version 1.45 **":PRINT:PRINT
152 PRINT,"Daten werden eingelesen":PRINT:PRINT,"- Bitte warten -":PRINT:PRINT
155 FOR z=&78B0 TO &83*256+&5F
160 READ v$:v=VAL("&"+v$):POKE z,v:s=s+v
165 NEXT
170 :
175 IF s<>222551 THEN PRINT"? DATA-Fehler !!!":STOP
180 :
185 PRINT"Bitte Diskette einlegen, auf der das BB145.BIN-File abgespeichert
190 PRINT"werden soll.":PRINT" >> Tastendruck <<":CALL &8B06
192 :
195 SAVE"BB145.BIN",8,&78B0,&780
200 :
205 INPUT"Noch einmal abspeichern (J/ )":yn$
210 IF UPPER$(yn$)="J"THEN 185 ELSE END
215 :
31664 DATA 21,7F,C1,11,00,05,01,33,00,ED,30,0E,42,11,00,00
31680 DATA 21,00,03,C3,12,01,CD,33,AD,20,80,CF,82,C1,CF,38
31696 DATA 7D,CF,77,7D,CF,7C,7D,CF,81,7D,CF,86,7D,CF,88,7D
31712 DATA CF,90,7D,CF,49,7E,CF,98,7D,CF,24,C5,CF,29,C5,CF
31728 DATA 87,7F,CF,4F,7E,CF,8E,7E,CF,95,7D,C3,88,FE,F3,CF
31744 DATA 38,00,F3,32,67,FE,E3,ED,43,65,FE,4E,23,46,ED,43
31760 DATA 68,FE,E1,3A,03,00,32,6A,FE,3E,90,01,8F,F8,ED,79
31776 DATA C3,DB,7C,3E,00,ED,79,ED,4B,65,FE,3A,67,FE,F8,C9
31792 DATA 00,00,00,00,00,00,3E,00,01,8F,F8,ED,79,3E,C3,21
31808 DATA 37,FE,32,08,00,22,09,00,21,33,FE,32,38,00,22,39
31824 DATA 00,E1,32,05,00,22,06,00,32,00,00,21,03,00,19,22
31840 DATA 01,00,06,12,EB,AF,11,00,FE,36,C3,23,77,23,C6,03
31856 DATA 72,23,10,F5,21,04,00,7E,E6,0F,FE,03,38,02,36,00
31872 DATA 4E,F8,C9,60,69,C9,3E,00,01,8F,F8,ED,79,01,80,00
31888 DATA ED,30,01,8F,F8,3E,00,ED,79,3E,00,C9,00,00,00,00

```

Listing 7/2.5.3-4 (Teil 1)

## 2.5 256 kByte-Speicher für Schneider CPC

Teil 7: Hard- und Software-Ergänzungen

```
31904 DATA 00,00,00,00,80,FD,00,FB,A0,FB,CD,FC,00,00,00,00
31920 DATA 00,00,00,00,80,FD,20,FB,20,FC,00,FD,00,00,00,00
31936 DATA 00,00,00,00,80,FD,01,FF,A0,FC,40,FD,80,00,03,07
31952 DATA 00,7F,00,1F,00,80,00,00,00,00,00,3A,6A,FE,32,03
31968 DATA 00,3A,67,FE,ED,4B,65,FE,08,D9,ED,73,42,00,31,CB
31984 DATA 7B,E5,2A,68,FE,D5,C5,F5,FD,E5,CD,C9,7F,ED,4B,40
32000 DATA 00,CB,99,ED,49,B7,FD,21,4B,AC,CD,34,7D,F3,08,D9
32016 DATA CB,D9,ED,49,ED,43,40,00,CD,CD,7F,FD,E1,F1,C1,D1
32032 DATA E1,D9,08,ED,7B,42,00,32,67,FE,ED,43,65,FE,01,BF
32048 DATA F8,C3,58,FE,E5,D9,08,C9,CD,FB,81,21,80,01,E5,23
32064 DATA 5E,23,56,21,A4,FC,19,EB,E1,CD,68,BA,CD,CD,7F,01
32080 DATA 00,16,ED,80,31,00,FB,D5,21,CB,7B,11,00,FE,01,10
32096 DATA 01,ED,80,D1,21,00,EA,19,E5,21,06,F2,19,E5,D9,ED
32112 DATA 43,40,00,D9,C3,68,FE,21,7D,C4,18,1C,21,86,C4,18
32128 DATA 17,21,8F,C4,18,12,21,A1,C4,18,0D,21,AA,C4,18,08
32144 DATA 21,BC,C4,18,03,21,98,C4,C3,6A,C4,79,32,53,BE,FE
32160 DATA 02,21,F1,FE,C8,21,00,00,0D,7B,1F,59,38,17,CD,DA
32176 DATA 82,38,41,CD,6C,C5,21,00,00,0D,3A,53,8E,5F,3E,18

32192 DATA CD,8F,82,36,00,CB,43,21,D1,FE,28,03,21,E1,FE,E5
32208 DATA 21,00,FB,28,03,21,20,FB,E5,AF,CD,8F,82,11,00,01
32224 DATA D5,01,19,00,C5,ED,80,CD,CD,7F,C1,E1,D1,ED,80,CD
32240 DATA C9,7F,E1,C9,3A,52,8E,3D,21,30,7E,28,0A,3A,51,8E
32256 DATA FE,0A,38,AF,21,17,7E,E5,AF,CD,8F,82,7B,EB,E1,01
32272 DATA 19,00,ED,80,5F,18,AE,28,00,04,0F,01,C7,00,7F,00
32288 DATA C0,00,20,00,00,00,0A,0A,19,30,E5,02,04,00,00,03
32304 DATA 20,00,04,0F,01,98,00,FF,00,FD,00,40,00,04,00,01
32320 DATA 10,1E,38,E5,01,02,00,00,03,CD,FB,81,C3,22,C5,3A
32336 DATA 53,8E,FE,02,CA,EE,7E,CD,0D,7E,CD,85,81,F5,CD,0D
32352 DATA 7E,28,17,CD,68,BA,CD,CD,7F,D5,11,10,FF,D5,01,80
32368 DATA 00,ED,80,E1,D1,CD,8B,FE,18,0F,CB,7C,C4,0B,80,20
32384 DATA 08,CD,CD,7F,01,80,00,ED,80,CD,C9,7F,F1,C9,3A,53
32400 DATA 8E,FE,02,CA,F3,7E,C5,CD,68,BA,CD,0D,7E,EB,28,17
32416 DATA CD,CD,7F,D5,11,10,FF,D5,CD,8B,FE,E1,D1,01,80,00
32432 DATA ED,80,CD,C9,7F,18,12,CB,7C,CC,0B,80,28,0B,CD,CD
32448 DATA 7F,01,80,00,ED,80,CD,C9,7F,C1,CD,5E,BA,C3,DE,81
32464 DATA ED,5B,BC,7F,7A,FE,7F,20,05,7B,FE,81,30,0B,AF,CB
32480 DATA 7A,21,00,90,28,03,21,00,01,22,60,8E,B7,C9,CD,85
32496 DATA 7F,18,05,CD,85,7F,EB,37,F5,CD,8E,7F,7C,FE,7F,38
32512 DATA 4A,20,0F,7D,FE,81,38,43,D5,11,10,FF,D5,CD,8B,FE
32528 DATA E1,D1,01,00,01,CD,AB,7F,7A,FE,7F,38,20,20,12,7B
32544 DATA FE,81,38,19,CD,CD,7F,01,10,FF,CD,AB,7F,F1,C3,8B
32560 DATA FE,F1,3F,CD,8E,7F,01,80,00,ED,80,AF,C9,CD,C9,7F
32576 DATA 01,00,90,CD,AB,7F,F1,3F,C3,00,80,CD,C9,7F,F1,F5
32592 DATA D5,11,00,90,D5,CD,00,80,E1,D1,7A,FE,7F,38,21,20
32608 DATA 0D,7B,FE,81,38,1A,01,00,01,CD,AB,7F,18,86,01,00
32624 DATA 01,CD,AB,7F,F1,3F,CD,8E,7F,01,80,00,ED,80,AF,C9
32640 DATA F1,3F,C3,00,80,CD,68,BA,26,00,3A,55,8E,6F,06,07
```

Listing 7/2.5.3-4 (Teil 2)

## 2.5 256 kByte-Speicher für Schneider CPC

## Teil 7: Hard- und Software-Ergänzungen

```
32656 DATA 29,10,FD,E5,26,00,3A,54,BE,6F,AF,06,0E,29,30,01
32672 DATA 3C,10,FA,D1,19,ED,5B,BC,7F,B7,C9,D5,50,59,D5,01
32688 DATA 80,00,ED,80,E1,D1,C9,ED,43,BC,7F,C9,00,00,38,00
32704 DATA 87,3E,A0,28,0A,CB,E7,18,06,3E,00,18,02,3E,90,F3
32720 DATA 01,BF,F8,ED,79,C9,CB,7F,CA,F2,07,C5,F5,01,BF,F8
32736 DATA 3E,01,ED,79,F1,CD,F1,8D,C1,C9,00,00,00,00,00,00
32752 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
32768 DATA 38,09,B7,3E,60,28,06,CB,E7,18,02,3E,50,F3,01,BF
32784 DATA F8,ED,79,01,80,00,ED,80,01,BF,F8,3E,00,ED,79,C9
32800 DATA 31,00,CD,3E,EF,32,2B,8D,21,D6,7F,22,2C,8D,11,40
32816 DATA 00,21,00,92,CD,AB,8B,21,62,81,7E,23,CD,AB,8B,7E
32832 DATA 87,20,F8,21,00,00,01,00,01,11,00,90,C5,E5,D5,ED
32848 DATA 80,E1,D1,F3,01,BF,F8,3E,50,ED,79,C1,ED,80,21,C5
32864 DATA 80,CD,A5,80,30,15,F3,01,BF,F8,3E,60,ED,79,21,00
32880 DATA 00,36,E5,11,01,00,01,00,04,ED,80,01,BF,F8,AF,ED
32896 DATA 79,31,CB,7B,21,80,01,01,48,0B,11,00,00,E5,CD,99
32912 DATA C2,E1,DC,AC,C2,30,03,C3,38,7D,21,0C,81,CD,A5,80
32928 DATA 38,E2,EF,00,00,7E,23,CD,5A,8B,87,20,F8,CD,81,8B
32944 DATA CD,06,8B,CD,84,8B,E6,DF,FE,4E,CA,5A,8B,FE,59,20
32960 DATA EC,37,C3,5A,8B,0D,0A,20,20,52,65,61,64,79,20,66
32976 DATA 6F,72,20,43,50,2F,4D,20,32,2E,32,20,69,6E,63,6C
32992 DATA 2E,20,2A,2A,20,35,38,2E,35,20,4B,42,20,2A,2A,20
33008 DATA 54,50,41,0D,0A,0A,43,3E,45,52,41,20,2A,2E,2A,0D
33024 DATA 0A,41,4C,4C,20,28,59,2F,4E,29,3F,00,0D,0A,20,20
33040 DATA 43,50,2F,4D,20,32,2E,32,20,6E,69,63,68,74,20,6C
33056 DATA 61,64,62,61,72,2E,0D,0A,57,69,65,64,65,72,68,6F
33072 DATA 6C,75,6E,67,76,65,72,73,75,63,68,2C,20,65,76,74
33088 DATA 6C,2E,20,6D,69,74,20,61,6E,64,65,72,65,72,20,44
33104 DATA 69,73,68,65,74,74,65,20,28,59,2F,4E,29,20,3F,0D
33120 DATA 0A,00,40,3C,60,3C,66,3C,06,3C,00,58,66,18,3C,66
33136 DATA 7E,66,66,00,5C,66,38,6C,C6,C6,6C,38,00,5D,6C,C6
33152 DATA C6,C6,C6,C6,7C,00,5E,38,6C,C6,00,00,00,00,00,7B
33168 DATA 6C,00,78,0C,7C,CC,76,00,7C,6C,00,3C,66,66,66,3C
33184 DATA 00,7D,66,00,66,66,66,66,3E,00,7E,3C,66,66,7C,66
33200 DATA 66,7C,60,00,00,AF,32,59,8E,CD,C2,81,CD,C7,C8,0D
33216 DATA AF,C9,F5,CD,54,C8,DA,51,C8,CD,FB,81,C1,0D,C5,CD
33232 DATA 80,C8,F1,38,06,CD,A2,C8,CD,1B,82,C3,4A,C8,C5,79
33248 DATA FE,02,CC,EB,C7,CD,00,C8,DC,1B,C8,CD,C2,81,C1,0D
33264 DATA CD,B6,C8,0D,37,CC,FB,81,DD,AF,C9,21,5E,BE,36,00
33280 DATA 2B,7E,B7,37,C8,34,CD,A2,C8,CD,9F,82,3E,45,CD,20
33296 DATA 82,3A,48,BE,3D,03,03,03,20,FA,C9,CD,9F,82,3E,66
33312 DATA 22,62,BE,67,69,22,74,BE,CD,76,C9,0E,0A,21,39,82
33328 DATA 7B,32,B4,81,CB,93,C3,FF,C6,2A,74,BE,3A,B4,81,5F
33344 DATA 3E,18,CD,88,82,CB,5F,28,02,CB,3A,01,7E,FB,7C,CD
33360 DATA 5C,C9,7B,CD,5C,C9,7A,CD,5C,C9,AF,CB,53,28,01,3C
33376 DATA CD,5C,C9,7D,CD,5C,C9,3E,14,CD,88,82,87,F5,CD,5C
```

Listing 7/2.5.3-4 (Teil 3)

## 2.5 256 kByte-Speicher für Schneider CPC

Teil 7: Hard- und Software-Ergänzungen

```
33392 DATA C9,7D,CD,5C,C9,3E,11,CD,88,82,CD,5C,C9,CB,93,F1
33408 DATA C2,8F,C6,3E,80,C3,C1,C6,E5,CD,9F,82,7E,E1,C9,D5
33424 DATA 2A,42,BE,CB,1B,11,40,00,30,01,19,5F,19,D1,C9,3E
33440 DATA 18,CD,88,82,1F,30,0F,1F,30,13,1F,1F,F5,CB,3A,30
33456 DATA 02,CB,D3,F1,18,03,1F,1F,1F,00,CB,02,C9,1F,47,3E
33472 DATA 28,30,01,87,3D,BA,30,0F,F5,CB,D3,87,92,3C,57,F1
33488 DATA BA,78,30,E4,E1,3F,C9,78,18,DE,CD,76,C9,3E,16,CD
33504 DATA 88,82,CB,D3,57,0E,10,21,5D,C5,CD,FF,C6,CB,93,D0
33520 DATA 3A,50,8E,3D,C8,3F,C9,0E,41,11,00,00,21,00,90,C5
33536 DATA D5,E5,CD,44,83,D1,D5,21,27,83,01,1D,00,ED,80,E1
33552 DATA D1,C1,CD,48,83,21,8D,7B,0E,43,06,05,C5,CD,48,83
33568 DATA 24,24,C1,0C,10,F6,C9,01,43,05,21,8D,7B,11,00,00
33584 DATA CD,99,C2,DA,8D,7B,D2,24,C2,CD,89,8E,30,06,CD,50
33600 DATA 01,C3,C6,7B,DF,48,83,C9,66,C6,07,DF,4F,83,C9,4E
33616 DATA C6,07,00,00,00,00,00,00,00,00,00,00,00,00,00
```

Listing 7/2.5.3-4 (Teil 4)



## 2.5 256 kByte-Speicher für Schneider CPC

## Teil 7: Hard- und Software-Ergänzungen

## „BTRANS.110“

```

;=====
;=                                     =
;=  BTRANS  - Version 1.10A -         =
;=                                     =
;=      auf Schneider CPC 464         =
;=  mit 256KB-Speichererweiterung    =
;=                                     =
;=  für BASIC 1.0 mit Zugriff auf    =
;=  3 zusätzl. Speicherbänke a 64K   =
;=                                     =
;=  geschrieben Juni 1987           =
;=  von                               =
;=      Kai Meier                    =
;=      Hauptstraße 25               =
;=      D-2361 Bebensee              =
;=====
;
ORG #A500
;
BANKPORT:EQU #F8BF ;Bankswitch-Portadresse
LIST: EQU #BD2B ;MC PRINT CHAR
EXT: EQU #BCD1 ;KL LOG EXT
;
    LD A,#EF      ;MC PRINT CHAR patchen (Opcode für RST 5)
    LD (LIST),A
    LD HL,PRT8BIT ;zeigt nun auf 8bit-Druckroutine
    LD (LIST+1),HL
    XOR A
    LD B,3        ;3 Speicherbänke
LBL1:  PUSH BC
      INC A
      PUSH AF
      CALL BANKSEL ;Speicherbank auswählen (untere Hälfte)
      LD DE,#0000 ;BTRANS-Routine an den Anfang
      LD HL,BTRANS
      LD BC,END-BTRANS
      LDIR        ;Übertragungsroutine einbauen
      POP AF
      POP BC
      DJNZ LBL1
      CALL BANK0  ;Bank 0 ab einschalten

```

Listing 7/2.5.3-5 (Teil 1)

## 2.5 256 kByte-Speicher für Schneider CPC

Teil 7: Hard- und Software-Ergänzungen

```

        LD  BC,RSX      ;RSX-Befehle
        LD  HL,KERNEL  ;4 Bytes für System
        JP  EXT        ;RSX-Befehle einhängen
;-----
RSX:    DEFW TAB
        JP  PUT        ;Syntax: IPUT,Start,Ziel,Länge,Speicherbank
        JP  GET        ;Syntax: IGET,... wie oben
TAB:    DEFB "P","U","T"+#80
        DEFB "G","E","T"+#80
        DEFB 0         ;Tabellenende
KERNEL: DEFS 4        ;4 Bytes für Kernel
;-----
PUT:    CALL PARAMS
        RET  NC        ;Fehler ->
        OR  A
        JR  Z,LDIR
        SCF            ;Schreibmodus
        JR  MOVE
GET:    CALL PARAMS
        RET  NC        ;Fehler ->
        OR  A         ;nc, Lesemodus
        JR  Z,LDIR
MOVE:   PUSH AF
        CALL BANKXA   ;Banknr. zu Bankswitch-Wert umrechnen
        LD  I,A       ;sichern
        POP AF
        DI
        JR  NC,READ   ;GET
WRITE:  PUSH BC       ;PUT
        LD  A,(HL)
        EX  AF,AF'
        INC HL
        LD  BC,BANKPORT;gewünschte Bank einschalten
        PUSH BC
        LD  A,I
        OUT (C),A
        SCF            ;Schreiben
        CALL #0000    ;BTRANS-Routine
        POP BC
        XOR A
        OUT (C),A    ;Bank 0ab
        INC DE
        POP BC       ;Zähler
        DEC BC
        LD  A,B
        OR  C
        JR  NZ,WRITE
        EI
        RET          ;fertig

```

Listing 7/2.5.3-5 (Teil 2)

## 2.5 256 kByte-Speicher für Schneider CPC

## Teil 7: Hard- und Software-Ergänzungen

```

READ:  PUSH BC
        LD  BC,BANKPORT;gewünschte Bank einschalten
        PUSH BC
        LD  A,I
        OUT (C),A
        OR  A          ;Lesen
        CALL #0000    ;BTRANS-Routine
        POP BC
        XOR A
        OUT (C),A     ;Bank 0ab
        INC HL
        EX  AF,AF'
        LD  (DE),A
        INC DE
        POP BC
        DEC BC
        LD  A,B
        OR  C
        JR  NZ,READ
        EI
        RET

;-----
LDIR:  LDIR          ;einfache Blockverschiebung in Bank 0
        RET

;-----
PARAMS: CP  4        ;4 Parameter ?
        SCF
        CCF
        RET  NZ      ;nein, -> Fehler
        LD  H,(IX+7)
        LD  L,(IX+6) ;HL: Start
        LD  D,(IX+5)
        LD  E,(IX+4) ;DE: Ziel
        LD  B,(IX+3)
        LD  C,(IX+2) ;BC: Länge
        LD  A,(IX+1)
        OR  A        ;muß 0 sein
        RET  NZ
        LD  A,(IX+0) ;Parameter Speicherbank
        CP  4
        RET          ;NC wenn Bank >3

;-----
PRT8BIT: BIT  7,A
        JP  Z,#07F2  ;MC PRINT CHAR (im L ROM), da 8.Bit=0
        PUSH BC
        PUSH AF
        LD  BC,BANKPORT

```

Listing 7/2.5.3-5 (Teil 3)

## 2.5 256 kByte-Speicher für Schneider CPC

Teil 7: Hard- und Software-Ergänzungen

```

LD A,1      ;D7 (8.Bit) des Druckerports
OUT (C),A   ;über Bankswitch-Port ausgeben
POP AF
CALL #BDF1  ;MC WAIT PRINTER-Indirection
PUSH AF
CALL BANK0  ;D7 rücksetzen
POP AF
POP BC
RET

:-----
BANKXA: OR A
RET Z      ;Bank 0
CP 2
JR NC,LBL10
LD A,%01010000;Bank 1a
RET
LBL10: LD A,%01100000;Bank 2a
RET Z
SET 4,A    ;Bank 3a
RET

;-----
BANKSEL: CALL BANKXA    ;A errechnen
DI
LD BC,BANKPORT
OUT (C),A    ;ausgeben
RET

;-----
BANK0: LD BC,BANKPORT
XOR A
OUT (C),A
EI
RET

;-----wird an den Anfang aller 3 zus. Speicherbänke kopiert-----
BTRANS: SET 7,A      ;obere Bankhälfte auch
OUT (C),A    ;selektieren
JR C,WR
EX AF,AF'
LD A,(HL)    ;Byte holen (GET-Befehl)
JR RETURN
WR: EX AF,AF'
LD (DE),A    ;Byte eintragen (PUT-Befehl)
RETURN: EX AF,AF'    ;alte Speicherkonfiguration holen/Byte retten
RES 7,A
OUT (C),A
RET

;-----
END: END

```

Listing 7/2.5.3-5 (Teil 4)

## Basic-Lader Verschieberoutine „BTRANS.BAS“

```
100 '*****
110 '* *
120 '* BTRANS - Version 1.10 - *
130 '* *
140 '* geschrieben Juni 1987 *
150 '* von Kai Meier *
160 '* Hauptstrasse 25 *
170 '* 2361 Babensee *
180 '*****
190 MODE 2:LOCATE 10,10:PRINT" - Bitte warten -":PRINT:PRINT
200 FOR z=&A500 TO &A60F
210 READ v$:v=VAL("&"+v$):POKE z,v:s=s+v
220 NEXT
230 :
240 IF s<>36230 THEN PRINT"? Data - Fehler !!":END
250 :
260 INPUT"Als BIN-File abspeichern (J/ )":a$
270 IF UPPER$(a$)<>"J"THEN 281
280 SAVE"btrans.bin",b,&A500,&110
281 CALL &A500 : 'RSX-Befehle und 8Bit Druckroutine einhängen
282 LOCATE 10,10:PRINT" Zusätzliche RSX-Befehle:"
283 PRINT:PRINT"Syntax:"
284 PRINT" öGET,Start-,Zieladresse,Länge,Speicherbank (0-3)"
285 PRINT" öPUT,Start-,Zieladresse,Länge,Speicherbank (0-3)"
286 PRINT:PRINT" Ferner ist nun der gesamte Zeichensatz des Druckers ausdrückbar (Bit7)"
287 PRINT:PRINT:END
290 DATA 3E,EF,32,2B,BD,21,BC,A5,22,2C,BD,AF,06,03,C5,3C
300 DATA F5,CD,E4,A5,11,00,00,21,F6,A5,01,12,00,ED,80,F1
310 DATA C1,10,EB,CD,EE,A5,01,2F,A5,21,3E,A5,C3,D1,BC,37
320 DATA A5,C3,42,A5,C3,4C,A5,50,55,D4,47,45,D4,00,FC,A6
330 DATA 2F,A5,CD,9A,A5,00,B7,28,4E,37,18,07,CD,9A,A5,00
340 DATA B7,28,44,F5,CD,D5,A5,ED,47,F1,F3,30,1D,C5,7E,08
350 DATA 23,01,BF,F8,C5,ED,57,ED,79,37,CD,00,00,C1,AF,ED
360 DATA 79,13,C1,08,78,B1,20,E5,FB,C9,C5,01,BF,F8,C5,ED
370 DATA 57,ED,79,B7,CD,00,00,C1,AF,ED,79,23,08,12,13,C1
380 DATA 08,78,B1,20,E5,FB,C9,ED,80,C9,FE,04,37,3F,CD,DD
390 DATA 66,07,DD,6E,06,DD,56,05,DD,5E,04,DD,46,03,DD,4E
400 DATA 02,DD,7E,01,B7,CD,DD,7E,00,FE,04,C9,CB,7F,CA,F2
410 DATA 07,C5,F5,01,BF,F8,3E,01,ED,79,F1,CD,F1,BD,F5,CD
420 DATA EE,A5,F1,C1,C9,B7,C8,FE,02,30,03,3E,50,C9,3E,60
430 DATA C8,CB,E7,C9,CD,D5,A5,F3,01,BF,F8,ED,79,C9,01,BF
440 DATA F8,AF,ED,79,FB,C9,CB,FF,ED,79,38,04,08,7E,18,02
450 DATA 08,12,08,CB,BF,ED,79,C9,00,00,00,00,00,00,00,00
```

Listing 7/2.5.3-6

## RSX-Befehle und 8-Bit-Druckerroutine einhängen „BTRANS.LDR“

```
100 '*****
110 '*
120 '* BTRANS (V.1.10) = Loader *
130 '*
140 '* geschrieben Mai/Juni 1987 *
150 '* von Kai Meier *
160 '* Hauptstrasse 25 *
170 '* 2361 Babensee *
180 '*****
190 :
200 MODE 2:MEMORY &A4FF:LOAD"BTRANS.BIN"
240 CALL &A500 :'RSX-Befehle und 8Bit Druckroutine einhängen
250 LOCATE 10,10:PRINT" Zusätzliche RSX-Befehle:"
260 PRINT:PRINT"Syntax:"
270 PRINT" öGET,Start-,Zieladresse,Länge,Speicherbank (0-3)"
280 PRINT" öPUT,Start-,Zieladresse,Länge,Speicherbank (0-3)"
290 PRINT:PRINT" Ferner ist nun der gesamte Zeichensatz des Druckers ausdrückbar (Bit7)"
300 PRINT:PRINT:END
```

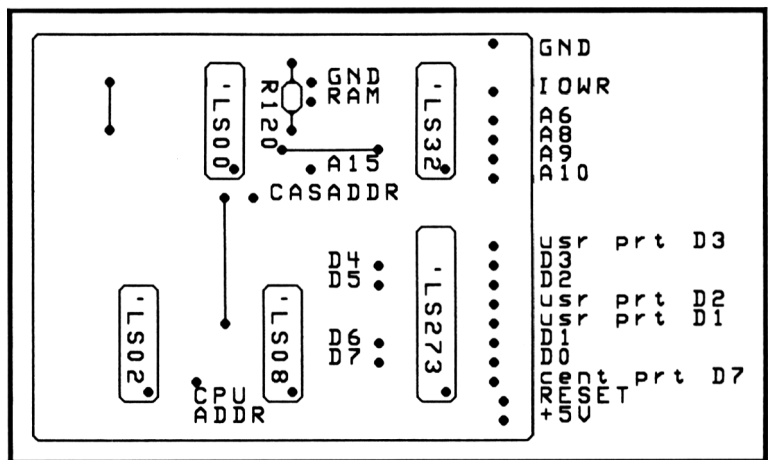
**Listing 7/2.5.3-7**

---

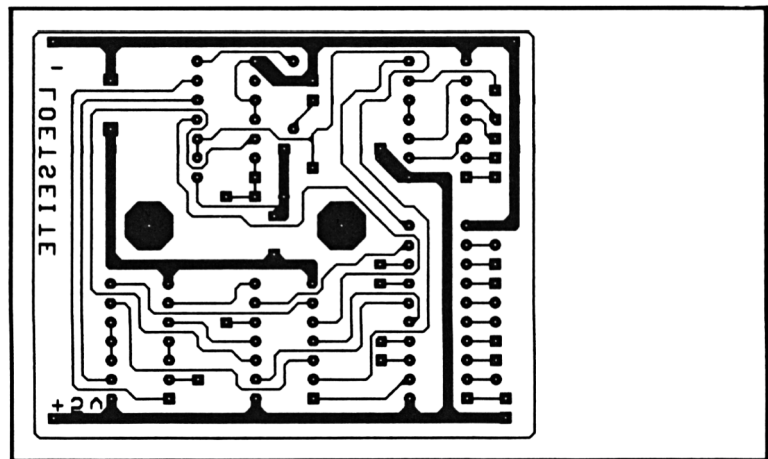
**2.5 256 kByte-Speicher für Schneider CPC**

Teil 7: Hard- und Software-Ergänzungen

Auf vielfachen Leserwunsch erhalten Sie heute noch das Platinenlayout und die Bestückungsseite zur CPC 256 kByte-Speichererweiterung.



Bestückungsseite



Platinenlayout



# Verdrahtungsplan

