

**Die vortex  
RAM-Erweiterungen  
SP 64 – SP 512**

**Benutzerhandbuch**



.....  
.  
.  
.  
.  
.....

Abschalten des BOS Eproms

Das HIMEM(ory=obere Speichergrenze unter BASIC) CPC 464 mit eingebauter vortex Speichererweiterung (HIMEM=42615) liegt 4 Bytes unter dem ohne Speichererweiterung (HIMEM=42619). Leider gibt es nun aber Software - vor allem Spiele - die starr von einer bestimmten Speicherkonfiguration ausgehen und gerade auch diese vier Bytes benötigen. Fazit: Die Spiele laufen nicht. Um hier Abhilfe zu schaffen, wurde in den BOS Eprom der zusätzliche Befehl DISBOS (DISable BOS) implementiert:

|DISBOS<ENTER> ; <ENTER>=Drücken der Enter Taste

schaltet den BOS Eprom ab und gibt dadurch die belegten 4 Bytes frei. Beachten Sie, daß hierbei ein Systemreset durchgeführt wird, und somit alle Daten im BASIC Ram verloren gehen.

Und so schalten Sie BOS wieder ein:

entweder

Rechner aus- und wieder einschalten

oder

Softreset durch CTRL+ESC+SHIFT Taste

oder

|BASIC<ENTER>

## Einleitung

\*\*\*\*\*

Lieber CPC 464 Besitzer,

endlich ist es soweit. Der CPC-Speicher kann erweitert werden. Bis auf 512KB. Sie haben sich für die vortex RAM-Erweiterung entschieden und damit ein hochwertiges und ausgereiftes Produkt erworben, das sicher allen Ihren Ansprüchen gerecht werden wird.

Modernste Elektronik und ausgeklügelte Software binden die RAM-Erweiterung nahtlos in die CPC 464 Arbeitsumgebung ein.

Ob Ihnen der BASIC-Programm/Datenspeicher nicht mehr ausreicht, oder Ihre CP/M-Programme mit dem zu kleinen CPC-Speicher nicht arbeiten, in jedem Fall erhalten Sie durch die vortex RAM-Erweiterung genügend Speicher... für jede Aufgabe.

### Leistungsmerkmale:

- durch perfekte Verbindungstechnik in jeden CPC 464 einbaubar. einfacher Einbau der Karte; kein Löten, nur Stecken.
- Karte bis 512KB RAM ausbaubar: kein Löten nur Zustecken entsprechender RAM-IC's.
- intelligente Software, die in jeder Ausbaustufe optimale Leistungen gewährleistet.
- sowohl mit Laufwerk (vortex 5.25" Floppy Disc Station F1 oder Schneider 3" Diskettenstation DDI-1), als auch ohne Laufwerk (hier natürlich nur unter BASIC) einsetzbar.
- voll einsetzbar unter CP/M: Sie erhalten ein 62K CP/M mit dem jedes CP/M Standard Programm läuft. Die Anzahl der möglichen Directory-Einträge vergrößert sich auf 128. Außerdem steht Ihnen immer ein 32KB großer Druckerpuffer zur Verfügung, so daß lästige Wartezeiten für Ausdrucke entfallen. Es kann simultan gedruckt und mit dem CPC gearbeitet werden. Je nach Ausbaustufe kommen Sie in den Genuß einer bis zu 448KB großen, superschnellen RAM-Disk.
- unter BASIC haben Sie je nach Ausbaustufe bis zu 288KB Programmspeicher und 256KB Datenspeicher. Neue BASIC-Befehle erlauben Ihnen den spielenden Umgang mit dem zusätzlichen Speicher.
- der von BASIC aus aufrufbare ROM-residente Z80-Monitor, läßt die Herzen aller Maschinenprogrammierer höher schlagen. Tracen, Dumpen, Listen, Assemblieren, Breakpoints,...
- zusätzliche BASIC Graphik-Befehle lassen für Bildschirnkünstler keine Wünsche mehr offen.
- gepufferter Expansions Bus. Dies ist die Voraussetzung für einen sicheren Betrieb weiterer Peripherie (z.B. RS 232 Schnittstelle) an einem CPC 464 mit Laufwerk.

Ein weiterer Pluspunkt ist der vortex Service Pass. Er liegt jeder vortex RAM-Erweiterung bei und sichert Ihnen nicht nur

Ihren selbstverständlichen Garantieanspruch, sondern gewährleistet auch, daß Sie ohne große Umstände oder Unkosten in den Genuß etwaiger Betriebssoftware-Erweiterungen kommen können.

Wir wünschen Ihnen nun viel Spaß und ein allzeit optimales und problemloses Arbeiten mit Ihrem CPC 464 und eingebauter vortex RAM-Erweiterung.

Ihr vortex Team

**WICHTIG WICHTIG WICHTIG WICHTIG WICHTIG WICHTIG WICHTIG**

Bitte beachten Sie die im Anhang beschriebenen Besonderheiten, bezüglich unsere Programme COPY, PARA, GRAPHIC MASTER 2.0 und SPTEST.COM.

Alle zum Lieferumfang der vortex RAM-Erweiterung gehörenden Programme sind urheberrechtlich geschützt (Copyright). Alle Rechte an diesen Programmen liegen bei der Firma vortex Computersysteme GmbH.

Vervielfältigung und Weitergabe - auch nur auszugsweise - dieses Handbuches bedürfen einer vorherigen schriftlichen Genehmigung der Firma vortex Computersysteme GmbH.

Alle Änderungen vorbehalten.

Z80 ist ein eingetragenes Warenzeichen der Firma Zilog.

vortex und vortex VDOS sind eingetragene Warenzeichen der Firma vortex Computersysteme GmbH.

CP/M ist ein eingetragenes Warenzeichen der Firma Digital Research Inc.

Dieses Handbuch wurde mit dem WordStar 3.0 (eingetragenes Warenzeichen der Firma MicroPro) von MicroPro auf dem CPC 464 mit eingebauter vortex RAM-Erweiterung geschrieben.

vortex Computersysteme Vertriebs GmbH  
Klingenberg 13  
7106 Neuenstadt

Neuenstadt im August 1985

**Einbau der vortex RAM-Erweiterung in Ihren CPC 464**

=====

Zum Einbau muß der CPC 464 geöffnet werden. Dazu brauchen wir einen mittleren Kreuzschraubenzieher und außerdem einen kleinen Geradschlitzschraubenzieher mit dessen Hilfe wir die Z80 CPU und das Gate Array aus ihren Fassungen heben können.

Das im Lieferumfang der RAM-Karte enthaltene Kühlblech wird nur bedingt, die Isolationsfolie auf jeden Fall benötigt. Für einen fehlerfreien Einbau muß die nachfolgende Reihenfolge unbedingt beachtet werden.

1. Wir schalten Monitor und CPC 464 sowie eine eventuell vorhandene Floppy aus und ziehen alle Netzstecker aus der Steckdose. Die beiden Kabel, die vom Monitor zum CPC 464 gehen, werden jetzt abgesteckt. Falls ein Floppy Controller angesteckt ist, so ziehen wir diesen ebenfalls vom CPC 464 ab.
2. CPC 464 mit der Unterseite nach oben auf eine weiche Unterlage legen und mit dem Kreuzschraubenzieher die sechs Kreuzschrauben, die das Bodenteil mit dem Oberteil verbinden lösen.
3. Den CPC 464 vorsichtig umdrehen. Dabei werden die sechs Schrauben herausfallen. Wir sammeln die Schrauben ein und legen sie beiseite. Der CPC 464 soll jetzt so vor uns liegen, wie er im Betrieb vor uns liegt.
4. Wir klappen jetzt das Oberteil vorsichtig nach vorne weg und stellen fest, daß dieses über zwei Kabelbäume mit dem Unterteil verbunden ist. Auf der linken Seite ziehen wir vorsichtig den einen der beiden Kabelbäume von der Unterseite der Tastatur und auf der rechten Seite den zweiten Kabelbaum von der Grundplatine des CPC 464 ab. Keine Angst, beide Kabelbäume stecken eindeutig und können also nicht "falsch herum" eingesteckt werden.
5. Jetzt können wir das Oberteil ganz vom Unterteil wegnehmen und legen es beiseite.
6. Wir sehen jetzt von oben auf die CPC 464 Platine. Von den Bauteilen (IC's, Widerstände, Kondensatoren...) interessieren uns nur sehr wenige. Genaugenommen nur zwei. Dies ist zum einen die Z80 und zum anderen das sogenannte Gate Array. Die Z80 ist ein integrierter Schaltkreis mit 40 "Beinchen" und befindet sich so ziemlich genau in der Mitte der Grundplatine auf einer Fassung. Das Gate Array, ebenfalls ein integrierter Schaltkreis mit 40 "Beinchen", befindet sich in der rechten oberen Ecke der Grundplatine, ganz in der Nähe der Steckbuchse für das Stromversorgungskabel. Das Besondere am Gate Array ist, daß es sich hierbei nicht um eine integrierte Schaltung handelt, die es in jedem besseren Elektronikladen zu kaufen gibt (wie dies z.B. bei der Z80 der Fall ist), vielmehr wurde dieser Baustein eigens für Amstrad konfiguriert und ist deshalb auch nur dort zu beziehen (oder bei der Firma Schneider).

**Achtung!** CPC 464 ist nicht gleich CPC 464. Es gibt hier bezüglich des Gate Arrays Unterschiede: Amstrad verwendet im CPC 464 zwei verschiedene Gate Arrays, wobei das eine ein Kühlblech braucht, wohingegen das andere ohne auskommt. Aber das Kühlblech ist nicht der einzige Unterschied, vielmehr sind beide Bausteine auch nicht pinkompatibel, d.h. man kann in eine Fassung in der bisher das eine Gate Array steckte nicht einfach das andere verwenden. Aus diesem Grund gibt es drei verschiedene Versionen der CPC 464 Grundplatine:

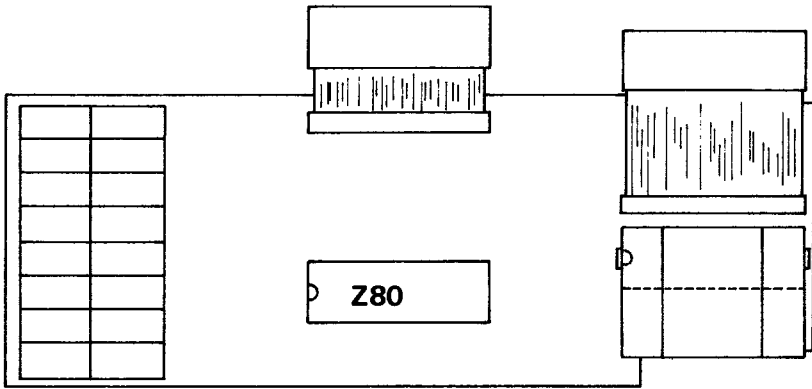
A. Grundplatine mit einer Fassung für ein Gate Array mit Kühlblech. (frühe Version des 464)

B. Grundplatine mit zwei optionalen Plätzen für eine Gate Array Fassung, wobei je nachdem, ob das Gate Array mit oder das ohne Kühlblech eingesetzt wurde, nur die obere oder die untere Fassung vorhanden ist. (Übergangsversion des 464)

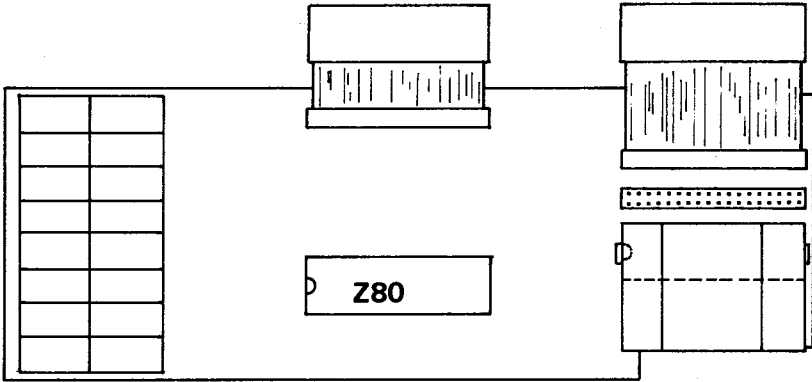
C. Grundplatine mit einer Fassung für ein Gate Array ohne Kühlblech.

D.h. die vortex Ram-Karte muß in der Lage sein, für drei verschiedene Lagen des Gate Arrays und zwei verschiedene Gate Arrays "einstellbar" zu sein. Dazu sind auf der RAM-Karte zwei 40 polige Fassungen und drei 40 polige Stiftleisten eingelötet.

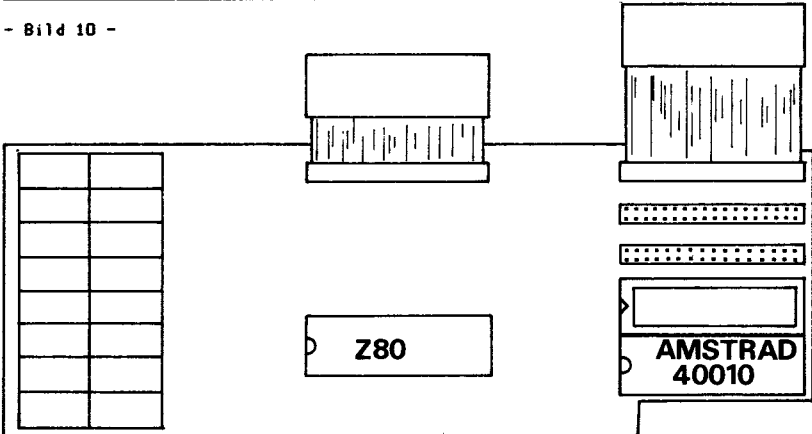
7. Wir sehen in der Mitte der Platine die Z80 CPU, ein 40 poliges IC, das in einer Fassung steckt. Wir hebeln es vorsichtig (keine "Hau-Ruck-Aktionen") mit dem Geradschlitzschraubenzieher aus seiner Fassung.
8. Befindet sich in Ihrem CPC das neue Gate Array (ohne Kühlblech, so heben Sie dieses genau so vorsichtig wie die Z80 aus seiner Fassung.  
Haben Sie ein Gate Array mit Kühlblech vor sich, so überprüfen Sie zunächst, ob sich an der Oberseite zwei halbkreisförmige Aussparungen befinden. Sollte dies nicht der Fall sein, so nehmen Sie einen Filzstift und bringen auf der linken Seite eine gut sichtbare Markierung auf dem Kühlblech an. Jetzt heben Sie das Gate Array samt Kühlblech vorsichtig aus der Fassung.
9. Jetzt entfernen wir den schwarzen Moosgummiblock, der sich an der Unterkante der CPC-Grundplatine befindet. Wird dieser nicht weggenommen, so lässt sich das Gehäuse nach dem Einbau RAM-Karte nicht mehr richtig schließen.
10. Nun legen wir unsere vortex RAM-Karte mit der Bauteilseite nach oben so vor uns auf die Arbeitsunterlage, daß die Verbindungskabel von uns weg weisen. Das Verbindungskabel auf der rechten Seite ist nicht eingelötet, sondern steckt auf einer der drei zweireihigen Stiftleisten. Dieses Verbindungskabel entfernen wir fürs erste von der Karte. In der Mitte der Karte sehen wir einen freien Steckplatz. Dort stecken wir, wie in den Bildern 10, 11 und 12 angedeutet die Z80 wieder ein. Dabei ist darauf zu achten, daß alle "Beinchen" der Z80 in der Fassung stecken und die Kerbe, so wie in den Bildern gezeigt, links zu liegen kommt.



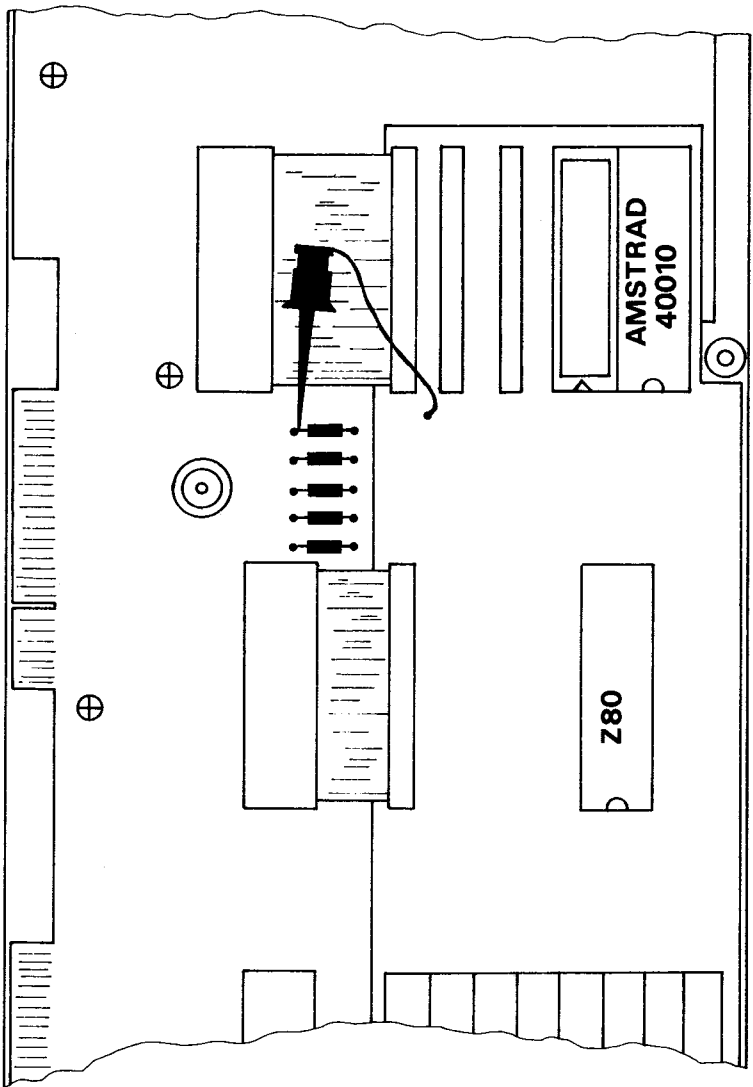
- Bild 11 -



- Bild 10 -



- Bild 12 -



- Bild 13 -



11. Auf der rechten Seite der RAM-Karte sehen wir zwei Fassungen und über diesen drei 40 polige Stiftleisten. Ein Gate Array ohne Kühlblech muß in die untere Fassung eingesteckt werden (Bild 12). Achten Sie darauf, daß alle "Beinchen" ordnungsgemäß in der Fassung stecken und die Kerbe am Gate Array nach links zeigt.

Ein Gate Array mit Kühlblech muß in die obere Fassung eingesteckt werden (Bilder 10 und 11). Achten Sie darauf, daß die halbkreisförmigen Aussparungen in dieselbe Richtung wie das Verbindungskabel weisen, also von Ihnen weg, bzw. die von Ihnen angebrachte Filzstiftmarkierung nach links weist. Dies ist deshalb wichtig, weil Sie ja wegen des Kühlblechs die Kerbe im Gate Array nicht sehen können, diese aber unbedingt auf der linken Seite sein muß.

Nachdem das Gate Array eingesteckt wurde, muß das Kühlblech gegen ein anderes (es liegt der RAM-Karten Packung bei) ausgetauscht werden. Dazu muß das alte Kühlblech entfernt werden, indem man den Bügel links und rechts des Gate Arrays etwas aufweitet, so daß das Blech herausgenommen werden kann. Die weiße Paste, die nun auf dem Gate Array sichtbar wird, darf auf keinen Fall abgewischt werden (es handelt sich dabei um sogenannte Wärmeleitpaste, die dafür sorgt, daß die Wärme, die im Gate Array erzeugt wird ohne nennenswerten Widerstand an das Kühlblech weitergegeben wird). Kontrollieren Sie nun auch noch einmal, ob die Kerbe nach links zeigt und ob alle "Beinchen" richtig in der Fassung stecken. Das neue Kühlblech wird wie aus Bild 10 oder 11 ersichtlich wieder aufgesteckt und der Bügel wieder etwas zusammengedrückt, so daß das Kühlblech sicher gehalten wird und richtig auf dem Gate Array aufsitzt.

**WICHTIG:** Z80 und Gate Array müssen sicher in ihren Fassungen sitzen und die Kerben im Baustein müssen in beiden Fällen nach links zeigen.

12. Der beiliegende 40 polige IC-Fassungsstecker mit dem Flachbandkabel, muß nun so auf eine der 40 poligen Stiftleisten gesteckt werden, daß die Karte so auf der CPC 464 Platine liegt, daß beide 40 poligen IC-Fassungsstecker in die freigewordenen Fassungen passen und die RAM-Karte parallel zur Grundplatine des CPC 464 liegt. Bild 11 zeigt die RAM-Karte mit Gate Array und Kühlblech für die CPC-Platine, die für beide Gate Arrays geeignet ist. Bild 10 zeigt die RAM-Karte für die CPC-Platine, die nur für das Gate Array mit Kühlblech ausgelegt ist.

**WICHTIG:** Wenn Sie beide Verbindungskabel richtig angebracht haben, so muß die Unterkante der RAM Karte parallel zur CPC Grundplatine verlaufen. Sollte dies nicht der Fall sein, so haben Sie das rechte Verbindungskabel (das über dem Gate Array) falsch eingesteckt.

13. In dem Bereich der CPC-Grundplatine, auf dem die RAM-Karte liegen wird, müssen alle Kondensatoren (neben jedem IC befindet sich einer) vorsichtig umgelegt werden. Keine Angst, ein liegender Kondensator ist genauso gut wie ein stehender.
14. Jetzt kann die RAM-Karte eingebaut werden. Dazu beachten wir Bild 13. RAM-Karte in Position auf CPC-Platine legen und die beiden 40 poligen Stecker genauso sorgfältig wie die Z80 und

das Gate Array in die Fassungen einstecken. Aus Bild 13 ist weiterhin zu entnehmen, wo der "Clip", welcher von der RAM-Karte weggeht "anzuclippen" ist.

**WICHTIG:** Sollten Sie den Clip aus Versehen falsch anbringen, so kann hierbei Ihr Gerät oder Ihre RAM-Karte nicht beschädigt werden.

15. Nun legen wir die mitgelieferte Isolationsfolie zwischen die RAM-Karte und die CPC-Platine, so daß nirgendwo Kontakte entstehen können.
16. Wir holen das CPC Oberteil und verbinden es über die zwei Kabelbäume wieder mit dem Unterteil.
17. Lesen Sie sich den nachfolgenden Abschnitt "WAS MACHEN, WENN DER COMPUTER NICHT MEHR LÄUFT?" gewissermaßen als Checkliste genau durch und prüfen Sie jeden Punkt noch einmal nach, falls Sie sich nicht sicher sein sollten. Überspringen Sie hierbei den ersten Punkt, da wir die Netzstecker selbstverständlich erst später einstecken.
18. Schrauben Sie jetzt Ober- und Unterteil wieder zusammen. Damit wäre der Einbau der vortex RAM-Karte erledigt.
19. Für das Arbeiten mit der Karte unter BASIC bedarf es keiner weiteren Vorbereitung. Für das Arbeiten unter CP/M müssen wir uns erst eine neue CP/M Systemdiskette herstellen. Dazu brauchen wir die mitgelieferte Datencassette.
20. Lesen Sie sich den nachfolgenden Abschnitt "WAS MACHEN, WENN DER COMPUTER NICHT MEHR LÄUFT?" genau durch und prüfen Sie jeden Punkt noch einmal nach, falls Sie sich nicht sicher sein sollten.
21. Schalten Sie nun Ihr Gerät ein. Wenn Sie alles richtig gemacht haben, so sehen Sie auf dem Bildschirm die gewohnte Einschaltmeldung und zusätzlich den ID (sozusagen den Steckbrief) Ihrer RAM Karte.

**WAS MACHEN, WENN DER COMPUTER NICHT MEHR LÄUFT?**  
-----

- Haben Sie alle Versorgungskabel wieder richtig eingesteckt? Sind die Netzstecker in der Steckdose? Leuchtet die rote Betriebsanzeige an der Konsole?
- Haben Sie beide Kabelbäume wieder richtig fest eingesteckt?
- Steckt die Z80 fest in der mittleren Fassung? Stecken alle "Beinchen" sauber in der Fassung? Wurde keines umgebogen? Zeigt die Kerbe nach links?
- Steckt das Gate Array fest in einer der beiden rechten Fassungen? (Mit Kühlblech in der oberen Fassung, ohne in der unteren.) Stecken alle "Beinchen" sauber in der Fassung? Wurde keines umgebogen? Zeigt die Kerbe nach links?
- Haben Sie das rechte Verbindungskabel ordnungsgemäß in die entsprechende 40 polige Stiftleiste eingesteckt? D. h. beide

Stiftreihen stecken sauber im Kabel, weder links noch rechts noch vorne noch hinten sind irgendwelche Stifte sichtbar.

- Stecken beide Verbindungskabelstecker sauber in den Fassungen der CPC Grundplatine? Haben Sie keine "Beinchen" umgebogen?
- Liegt die Isolationsfolie sauber zwischen RAM-Karte und Grundplatine?
- Haben Sie den (die) Clip(se) richtig angebracht?

Wenn Sie alle diese Punkte überprüft haben, dann müßte Ihr Rechner zusammen mit der RAM-Karte ordnungsgemäß arbeiten.

**Erstellen einer lauffähigen CP/M Arbeitsdiskette**  
=====

Voraussetzung, um überhaupt mit CP/M arbeiten zu können, ist das Vorhandensein einer Diskettenstation. Hier kommt die 5.25" vortex Floppy Disk Station F1, oder die 3" Schneider Diskettenstation DDI-1 in Frage. Haben wir weder die eine noch die andere, so brauchen wir den folgenden Abschnitt erst gar nicht zu lesen.

Sind wir aber stolzer Besitzer einer solchen Diskettenstation, so haben wir mit ihrem Kauf auch das CP/M Betriebssystem, sowie eine Reihe sogenannter Systemutilities (Dienstprogramme) miterhalten. Wie wir bald sehen, werden einige diese Utilities und natürlich auch das CP/M Betriebssystem im folgenden benötigt, um eine neue CP/M Systemdiskette zu erstellen. Das "Neue" an dieser Diskette sind Modifikationen an den CP/M-Spuren (Spur 0 und Spur 1), sowie zwei zusätzliche Dienstprogramme. Diese Zusätze werden benötigt, um die Speichererweiterung nahtlos in die bereits vorhandene Arbeitsumgebung einzubinden.

Das Programm, das Sie hierzu brauchen, befindet sich auf der mitgelieferten Datencassette und heißt PATCH.COM. Es ändert die Systemspuren, erzeugt die beiden neuen Dienstprogramme RAMDISK.COM und SPOOL.COM und legt automatisch einen sogenannten Systemfile (\$OSC.SYS) an, der Teile des Betriebssystems enthält. Mit dem Programm RAMDISC wird die RAM-Disc formatiert. Dieses Programm entspricht dem, bei der "normalen" Floppy benötigten Programm FORMAT. Das Programm SPOOL wirkt wie ein Schalter, mit ihm kann der Druckerpuffer ein- und ausgeschaltet werden.

Die Arbeitsgänge zur Erstellung dieser neuen Diskette sind im wesentlichen folgende:

1. Backup Kopie der Systemdiskette anfertigen (mit !FORMAT und COPY bei der vortex Station bzw. DISCCOPY bei der Schneider Station)
2. Das Programm PATCH von der Datencassette auf die Kopie der Systemdiskette übertragen (mit CASCOPY bei der vortex Station bzw. CLOAD bei der Schneider Station)
3. Starten des Programms PATCH zur Aktivierung der RAM-Karte unter CP/M
4. Erzeugung eines 62k CP/M's mit Hilfe der Programme MOVCPM und SYSGEN

Eine genaue Beschreibung der einzelnen Schritte finden Sie nachfolgend.

Nachdem diese Arbeiten erledigt sind, haben Sie eine neue CP/M Systemdiskette, die im Unterschied zur bisherigen statt einem 44k CP/M ein 62k CP/M enthält und außerdem einen 32k großen Druckerpuffer und je nach RAM-Karte eine bis zu 448k große RAM-Disc ermöglicht.

Es ist sinnvoll diese Diskette wie die bei der Floppy Station mitgelieferte Original Systemdiskette zu behandeln. D.h. sie soll von nun an als "Master" für alle weiteren von Ihnen benötigten Systemdisketten dienen und auch an einem sicheren Ort aufbewahrt werden.

Die nachfolgende Beschreibung zeigt die genaue Vorgehensweise zur Erstellung einer neuen CP/M Systemdiskette, auf der linken Seite für den CPC 464 mit vortex Floppy Disk Station F1-S oder F1-D und auf der rechten Seite für CPC 464 mit Schneider Diskettenstation.

=====

! CPC 464 + vortex 5.25" F1-D/S ! CPC 464 + Schneider 3" DDI-1 !

-----

! I. Erstellung einer Kopie der Original Systemdiskette !

! A. Formatieren einer neuen !	! A. Formatieren und gleich-	!
! leeren 5.25 " Diskette.	! zeitiges Kopieren der Original!	!
! Wir tun dies mit dem BASIC-	! Systemdiskette auf eine neue !	!
! Befehl  FORMAT,1 <CR> (Bild 1)!	! leere 3" Diskette. Dazu legen !	!
!	! Sie die Original Systemdisket-	!

! B. Original Systemdiskette !	! te in das Laufwerk A und laden!	!
! holen, in Laufwerk A einlegen !	! mit  CPM <CR> das CP/M !	!
! und mit dem Befehl  CPM <CR> !	! Betriebssystem. !	!
! das CP/M Betriebssystem laden !	! Besitzen wir nur ein Laufwerk, !	!
! (Bild 1) !	! dann starten wir jetzt das !	!
!	! Kopierprogramm mit der Eingabe!	!

! C. Kopieren der Original Sy-	! DISCCOPY <CR> (Bild 2). !	!
! stemdiskette auf die gerade !	!	!
! formatierte Diskette. !	!	!

! Dazu starten !	! Original und Kopie müssen so !	!
! wir mit COPY <CR> das Kopier-	! lange ausgetauscht werden, bis!	!
! programm (Bild 3). Wenn Sie !	! der Vorgang beendet ist. !	!
! die F1-S besitzen, müssen Sie !	! Haben Sie zwei Laufwerke, so !	!
! Original und Kopie solange !	! legen wir das Original in das !	!
! austauschen, bis der Vorgang !	! Laufwerk A und die zu erstel-	!
! beendet ist. Haben Sie die !	! lende Kopie in Laufwerk B und !	!
! die F1-D, so legen Sie das !	! starten durch die Eingabe !	!
! Original in Laufwerk A und die!	! COPYDISC <CR> das Kopierpro-	!
! zu erstellende Kopie in Lauf-	! gramm. !	!
! werk B. !	! WICHTIG: Die Kopie muß auf !	!
!	! jeden Fall das Schneider !	!
!	! Systemformat besitzen. !	!

! D. Original Systemdiskette !	! B. Original Systemdiskette !	!
! wieder an einen sicheren Ort !	! wieder an einen sicheren Ort !	!
! bringen. !	! bringen. !	!

-----

! II. Kopieren des Programms PATCH von Cassette auf Diskette !

! A. Mitgelieferte Datencasset-	! A. Mitgelieferte Datencasset-	!
! te in den Datenrekorder ein-	! te in den Datenrekorder ein-	!
! legen und eventuell zurück-	! legen und eventuell zurück-	!
! spulen. Kopie der Original !	! spulen. Kopie der Original !	!
! Systemdiskette in Laufwerk A !	! Systemdiskette in Laufwerk A !	!
! einlegen. !	! einlegen. !	!
!	!	!

! B. Durch die Eingabe von !	! B. Aus Platzgründen müssen !	!
------------------------------	--------------------------------	---

```
BASIC 1.0
Ready
IFORMAT,1
Laufwerk A: Diskette zum formatieren
eingelegt J/N ? J
Formatieren Seite 1 Spur 79
Ready
ICPM █
```

- Bild 1 -

```
CP/M 2.2 - Amstrad Consumer Electronics plc
A>DISCCOPY
DISCCOPY V2.0
Please insert source disc into drive A then press any key:_
Copying started
Please insert destination disc into drive A then press any key:█
```

- Bild 2 -

```
44K CP/M vers. 2.2-01/85 vortex GmbH
A>COPY
COPY 2.0 (C)1985 vortex GmbH
Quelldiskette in Laufwerk A oder B ? A
Zieldiskette in Laufwerk A oder B ? A
Quelldiskette in Laufwerk A einlegen und irgendeine Taste druecken
Lesen - Seite 0 Spur 07
Zieldiskette in Laufwerk A einlegen und irgendeine Taste druecken
Schreiben - Seite 0 Spur 07
Quelldiskette in Laufwerk A einlegen und irgendeine Taste druecken █
```

- Bild 3 -

```
A>CASCOPIY
CASCOPIY 2.0 (C)1985 vortex GmbH
Kopierrichtung (<,>) angeben:
Diskette < Cassette
Name der Cassettendatei:
Name der Diskettendatei:
Press PLAY then any key:
Loading PATCH.COM block 9
A>█
```

- Bild 4 -

```
A>CLOAD
CLOAD V2.0
Press PLAY then any key:
Loading PATCH.COM block 9
CLOAD V2.0 finished
A>█
```

- Bild 5 -

A>PATCH

**CP/M Initialisierung fuer vortex RAM Erweiterung (C)1985 vortex**

(Jede Eingabe kann mit ^C abgebrochen werden.)

Drucker-Spooler nach dem Booten (e)in oder (a)us ? E  
RAM-Disk beim Booten formatieren. (J)a oder (N)ein ? N  
Systemvektor aktiv. (J)a oder (N)ein ? J  
CP/M Neustart J/N ? N

- Bild 6 -

44K CP/M vers. 2.2-09/85 vortex GmbH  
(Spooler an - RAM Disk nicht formatiert)

A>MOVCPM 250 \*

CONSTRUCTING 62k CP/M vers 2.2  
READY FOR "SYSGEN" OR  
"SAVE 34 CPM62.COM"  
A>■

- Bild 7 -

A>SYSGEN

**SYSGEN 2.0 (C)1985 vortex GmbH**

Quelldiskette in Laufwerk A,B oder RETURN ?  
Zieldiskette in Laufwerk A,B oder RETURN ? A  
Zieldiskette in Laufwerk A einlegen und irgendeine Taste druecken

Noch eine Diskette J/N ? N  
Diskette mit Systemspur in Laufwerk A einlegen und irgendeine Taste druecken

A>■

- Bild 8 -

A>SYSGEN \*

SYSGEN V2.0

Please insert DESTINATION disc into drive A then press any key: \_

Do you wish to reconfigure another disc (Y/N) ? : N

Please insert a CP/M system disc into drive A then press any key: \_

SYSGEN V2.0 finished

A>■

- Bild 9 -

```
! CASCOPY <CR> starten wir das ! alle Programme ausschließlich !
! Cassetten Kopierprogramm ! MOVCPM.COM, SYSGEN.COM und !
! (Bild 4). ! CLOAD.COM mit Hilfe des ERA- !
! ! Kommandos gelöscht werden. !
! ! !
! ! C. Durch die Eingabe von !
! ! CLOAD <CR> starten wir das !
! ! Cassetten Kopierprogramm !
! ! (Bild 5). !
```

-----  
! III. Benutzung des Programms PATCH !  
-----

```
! Die Benutzung dieses Programms ist bei beiden Laufwerken !
! (vortex 5.25" F1-S/D, bzw. Schneider 3" DDI-1) identisch. !
! Wir befinden uns noch immer im CP/M und starten nun durch die !
! Eingabe PATCH <CR> das gerade zuvor überspielte Programm. !
! Dieses Programm stellt uns nun ein paar Fragen, die wir wie !
! folgt beantworten (Bild 6). Was sich hinter den einzelnen !
! Fragen verbirgt, werden wir später noch erklären. !
```

```
! Die erste Frage: !
! Drucker-Spooler nach dem Booten (e)in oder (a)us ? !
! beantworten wir mit einem 'E' für (e)in. !
```

```
! Die zweite Frage: !
! RAM-Disk beim Booten formatieren. (J)a oder (N)ein ? !
! beantworten wir mit einem 'N' für ein (N)ein. !
```

```
! Die dritte Frage: !
! Systemvektor aktiv. (J)a oder (N)ein ? !
! beantworten wir mit einem 'J' für ein (J)a. !
```

```
! Die vierte Frage: !
! CP/M Neustart J/N ? !
! beantworten wir mit einem 'J' für ein (J)a. !
```

```
! Nach der letzten Eingabe wird der Bildschirm dunkel und nach !
! kurzer Zeit kommt die neue Lademeldung, die uns u.a. mitteilt, !
! daß der Spooler an und RAM-Disk nicht formatiert ist (Bild 7). !
```

```
! Schauen wir uns jetzt mit DIR <CR> den Inhalt der Diskette !
! an, so stellen wir fest, daß drei neu Dateien vorhanden sind: !
! $OSC.SYS, RAMDISK.COM und SPOOL.COM. Dies sind die bereits !
! oben erwähnten neuen Dienstprogramme. !
! Mit den bisherigen Arbeitsschritten haben wir uns ein CP/M er- !
! zeugt, das in der Lage ist die RAM-Karte zu verwalten. !
```

-----  
! IV. Installation eines 62k CP/M 2.2 Betriebssystems !  
-----

```
! A. Wir starten durch die Ein-! A. Wir starten durch die Ein-!
! gabe von MOVCPM 250 * <CR> das! gabe von MOVCPM 250 * <CR> das!
! Programm, das uns ein 62K CP/M! Programm, das uns ein 62K CP/M!
! erzeugt (Bild 7). ! erzeugt (Bild 7). !
```

```
! B. Durch die Eingabe von ! B. Durch die Eingabe von !
! SYSGEN <CR> starten wir das ! SYSGEN * <CR> starten wir das !
! Programm, das das 62K CP/M auf! Programm, das das 62K CP/M auf!
! die Diskette wegschreibt (Bild! die Diskette wegschreibt (Bild!
! 8). Bei der Frage: ! 9). Bei der Bitte: !
! Quelldiskette in Laufwerk A, B ! Please insert DESTINATION disc!
! oder RETURN ? ! into drive A then press any !
```

```

! drücken wir die ENTER-Taste. ! key: !
! drücken wir die ENTER-Taste !
! Bei der Frage Zieldiskette in !
! Laufwerk A, B oder RETURN ? ! Bei der Frage:
! geben wir A ein. ! Do you wish to reconfigure an-
! ! other disc (Y/N) ?
! Bei der Bitte: ! antworten wir mit einem N
! Zieldiskette in Laufwerk A !
! einlegen und irgendeine Taste ! Bei der Bitte:
! drücken ! Please insert a CP/M system
! drücken wir die ENTER-Taste. ! disc into drive A then press
! ! any key:
! Die Frage: ! drücken wir die ENTER-Taste
! Noch eine Diskette J/N ? !
! beantworten wir mit einem N. !
! !
! Bei der Forderung: ! C. Nun führen wir einen
! Diskette mit Systemspur in ! System Reset aus, indem wir
! Laufwerk A einlegen und irgend! gleichzeitig die CTRL-, die
! eine Taste drücken ! SHIFT- und die ESC-Taste
! drücken wir die ENTER-Taste ! drücken.
! !
! C. Nun führen wir einen ! D. Durch die Eingabe von
! System Reset aus, indem wir ! | CPM <CR> starten wir erneut
! gleichzeitig die CTRL-, die ! das CP/M, das sich jetzt als
! SHIFT- und die ESC-Taste ! 62K großes CP/M meldet.
! drücken. !
! !
! D. Durch die Eingabe von !
! | CPM <CR> starten wir erneut !
! das CP/M, das sich jetzt als !
! 62K großes CP/M meldet. !
! !

```

-----

```

! Damit sind alle Arbeiten beendet und Sie haben eine neue CP/M !
! Systemdiskette, die jetzt voll mit der RAM-Karte arbeitet. !
! Alle weiteren Systemdisketten, die Sie benötigen, können Sie !
! als Kopie dieser Diskette erzeugen. !

```

-----

#### Drucker-Spooler, Booten, RAM-Disk, Systemvektor ??

-----

Um schnell zu einem Ergebnis zu kommen, haben wir vorhin keinen Wert auf Erklärungen gelegt, sondern nur einen vorgeschriebenen Weg benutzt, um eine neue CP/M Systemdiskette zu erhalten. Was allerdings hinter den einzelnen Fachbegriffen steckt, soll im folgenden erläutert werden.

Unter "Booten" versteht man das Laden und Starten des CP/M Betriebssystems. Dieser Vorgang wird beim CPC 464 mit Floppy Disk Station aus BASIC heraus durch die Eingabe von |CPM <CR> veranlasst. Dabei wird das CP/M von der Diskette - es befindet sich dort auf den ersten beiden Spuren - in den Speicher des CPC 464 geladen und gestartet. Nach dem Start übernimmt es die Kontrolle über den CPC 464.

Sie haben sich vielleicht auch schon darüber geärgert, daß Sie während des Ausdrucks eines Textes oder eines Listings, nicht mit Ihrem CPC 464 arbeiten können. Das liegt daran, daß der Drucker z.B. im Vergleich zum Monitor ein sehr langsames Ausgabegerät ist und die Daten vom CPC an den Drucker deshalb nur so



schnell übermittelt werden können, wie dieser sie verarbeiten kann d.h. der CPC 464 ist für die gesamte Zeit des Ausdrucks blockiert. Ein "Drucker-Puffer" oder "Drucker-Spooler" schafft hier Abhilfe. Der Puffer oder Spooler ist ein RAM-Speicher der sehr schnell mit den zu druckenden Daten "gefüllt" wird. Nur während dieser kurzen Zeit ist der CPC 464 "belegt". Er steht Ihnen danach wieder sofort zur Verfügung. Die "Leerung" des Puffers erfolgt dann mit der kleinen Geschwindigkeit des Druckers unabhängig davon, ob Sie mit dem CPC 464 arbeiten, also z.B. schon Schreibfehler in Ihrem Text korrigieren der gerade auch ausgedruckt wird. Der Drucker-Spooler auf der vortex RAM-Erweiterung ist 32KB groß. Das entspricht ca. 16 vollgeschriebenen DIN A4 Schreibmaschinen-Seiten; das ist sicher ausreichend.

Mit der Frage "Drucker-Spooler nach dem Booten (e)in oder (a)us?" kann entschieden werden, ob sofort nach dem Booten von CP/M jede Ausgabe, die auf den Drucker soll, über den Spooler gehen soll, oder nicht. Beachten Sie hierbei, daß der Drucker unter CP/M mit CTRL P ein- bzw. ausgeschaltet wird.

Mit dem neuen Dienstprogramm SPOOL kann der Spooler wie mit einem Schalter ein- und ausgeschaltet werden.

Eine RAM-Disk ist ein sogenanntes Pseudo-Laufwerk. Ein Laufwerk, wie z.B. das 80 Spuren BASF Laufwerk in Ihrer vortex Floppy Disk Station, besteht aus Mechanik (Motoren, Schreib/Leseköpfe) und Elektronik (Controller, Motorsteuerung). Das Speichermedium sind Magnetscheiben, auch Disketten genannt. Um z.B. Daten auf eine Diskette zu schreiben, muß zuerst der Antriebsmotor der Diskette anlaufen, muß der Schreib/Lesekopf auf die Diskette abgesenkt werden und die Stelle auf der Diskette angefahren werden, wo die Daten abgelegt werden sollen und dann müssen die Daten noch tatsächlich weggeschrieben bzw. gelesen werden. Das kostet alles Zeit, wobei der eigentliche Schreibvorgang nur einen Bruchteil dieser gesamten Zeit beansprucht. Bei häufigem Diskettenzugriff macht sich dieser Zeitaufwand bemerkbar und wirkt oftmals auf das eigentliche Arbeiten mit dem Computer hemmend.

Die Lösung bringt hier eine RAM-Disk. Sie besteht nur aus RAM-Speicher und beinhaltet somit also kein einziges Zeit beanspruchendes mechanisches Teil. Eine RAM-Disk ist im allgemeinen ca. 50 mal schneller als ein "normales" Laufwerk. Angesprochen wird die RAM-Disk wie ein mechanisches Laufwerk. Es bekommt unter CP/M die Laufwerkskennung "C" und muß wie die Disketten seiner "mechanischen Brüder" vor der ersten Benutzung formatiert werden. Es ist aber natürlich klar, daß der Inhalt der RAM-Disk beim dem Abschalten des Rechners verloren geht.

D.h. man muß die Daten zuvor auf Diskette sichern. Eine Anwendung der RAM-Disk im Zusammenhang mit einem diskettenzugriffintensiven Programm könnte wie folgt aussehen: wir haben ein Datenbankprogramm und möchten z.B. bereits vorhandene Datenbanken verändern oder ergänzen. Als aller erstes laden wir z.B. mit dem PIP Kommando alle hierzu erforderlichen Programme und Dateien in die RAM-Disk. Dort arbeiten wir mit dem Programm, ungestört von permanenten Unterbrechungen durch Laufwerkszugriff. Sind wir fertig, so kopieren wir einfach alles wieder zurück auf die Diskette, so daß wir ohne Datenverluste den CPC 464 auch ausschalten dürfen.

Mit der Frage "RAM-Disk beim Booten formatieren. (J)a oder (N)ein?" entscheiden Sie, ob bei jedem Neustart von CP/M die RAM-Disk formatiert wird, d.h. alle Daten in ihr gelöscht werden. Geben Sie hier "N" für nein ein, so geht der Inhalt der RAM-Disk

auch nicht dann verloren, wenn Sie z.B. mit VDOS <CR> CP/M verlassen haben und dann wieder mit |CPM <CR> aus BASIC heraus CP/M starten ("Booten") wollen. Es muß allerdings beachtet werden, daß in diesem Fall (RAM-Disk wird beim Booten nicht automatisch formatiert) beim ersten Starten von CP/M die RAM-Disk für die Benutzung vorbereitet, d.h. formatiert werden muß. Dies wird mit Hilfe des Dienstprogramms RAMDISK erledigt. Geben Sie bei obiger Frage ein "J" für ja ein, so wird bei jedem CP/M-Start mit |CPM <CR> aus BASIC heraus die RAM-Disk formatiert, d.h. der Inhalt gelöscht. Die RAM-Disk ist in der hier angegebenen Form nur unter dem CP/M Betriebssystem verwendbar. **WICHTIG:** in den Genuß der RAM-Disk kommen Sie ab der RAM-Karte SP128.

Der Systemvektor wird für die Funktionsfähigkeit der sogenannten Systemutilities oder Dienstprogramme (z.B. INSTALL, FORMAT...) benötigt. Die Dienstprogramme - bei vortex und Schneider - sind normalerweise nicht für ein 62K großes CP/M ausgelegt. Alle diese Programme stützen sich teilweise auf eine im RAM-Speicher liegende "Sprungleiste", auch "Sprungvektor" genannt. Ein Sprungvektor ist eine Reihe von Sprunganweisungen ("Jumps"), die zu bestimmten Unterprogrammen führen. Außerdem liegt dieser Sprungvektor an einer ganz bestimmten Stelle (ab einer bestimmten Adresse) im RAM, die von Benutzerprogrammen nicht belegt wird (d.h. außerhalb des TPA=Transient Program Area; für den Benutzer zur Verfügung stehender RAM-Bereich). Dadurch daß der Speicher erweitert und damit das CP/M vergrößert wird, "finden" die Dienstprogramme diesen Sprungvektor nicht mehr an dieser Stelle. Um diesen Notstand zu beseitigen wird an der "gewohnten" Stelle ein neuer Sprungvektor installiert, der "Systemvektor", der dann sozusagen als "Mittler" zwischen Dienstprogrammen und Sprungvektor dient. Mit der Frage "Systemvektor aktiv (J)a oder (N)ein?" können Sie entscheiden, ob Sie unter dem 62K CP/M die Dienstprogramme benutzen wollen, oder nicht. Sie fragen sich vielleicht, zu was es gut sein kann, unter einem 62K CP/M keine Dienstprogramme zu haben. Wie wir gerade erklärt haben, wird zum Betrieb dieser Programme der Systemvektor benötigt, der praktisch an der selben Adresse im RAM liegt, wie unter dem "normalen" 44K CP/M der eigentliche Sprungvektor. Diese Stelle liegt aber jetzt nicht mehr außerhalb des TPA, sondern mitten drin. Es kann hier beim Arbeiten mit dem Programm DDT, das zum CP/M gehört, und dem SAVE-Befehl unerwünschte Überlappungen geben. Wer also viel mit diesen Programmen arbeitet, sollte den Systemvektor nicht einschalten. Für die normale Benutzung des 62K CP/M ist es aber auf jeden Fall sinnvoll immer mit aktivem Systemvektor zu arbeiten.

**Die BASIC Befehlserweiterungen der vortex RAM-Erweiterung**  
-----

Die vortex RAM-Erweiterung bietet Ihnen in jeder Ausbaustufe unter BASIC zusätzlichen Programm- und Datenspeicher. Bei der kleinsten Karte sind es volle 64KB Programm- und 32KB Datenspeicher; bei der größten Karte erhalten Sie sage und schreibe 288KB Programm- und 256KB Datenspeicher! Das ist genug für jede Aufgabe.

Über dreißig neue BASIC-Befehle ermöglichen den optimalen Einsatz der RAM-Erweiterung unter BASIC.

Die Programmspeicher-Erweiterung wird dadurch erreicht, daß dem Programmierer bis zu 9, 32KB große, Programmspeicherbänke zur Verfügung stehen.

Belegt ein Programm mehrere Bänke, so finden sich unter Umständen in diesem Programm mehrmals gleiche Zeilennummern. Das ergibt aber keinen Konflikt, denn neben der Zeilennummer tritt zur eindeutigen Identifikation einer BASIC-Programmzeile noch die Banknummer, in der diese Zeile liegt hinzu. Eine Sprunganweisung könnte also wie folgt lauten: "springe von Zeile 25, Bank 2 zur Zeile 25, Bank 8".

**Leistungsmerkmale des erweiterten BASIC Betriebssystems:**  
-----

- (1) 64.....288KB Programmspeicher
- (2) 32.....256KB Datenspeicher
- (3) 32KB           Druckerpuffer

Der Druckerpuffer lässt sich ein- und ausschalten (SPOOL.ON, SPOOL.OFF). Ist er eingeschaltet, so werden in ihm alle Ausgaben an den LIST-Kanal mit hoher Übernahmegeschwindigkeit zwischengespeichert. Die tatsächliche Ausgabe auf den Drucker geschieht dann gewissermaßen im Hintergrund, d.h. während Sie mit den Rechner arbeiten. Die Arbeitsgeschwindigkeit des Rechners wird hierbei, solange der Druckerpuffer nicht voll ist, nur unwesentlich herabgesetzt.

Der Datenspeicher lässt sich in zweifacher Weise nutzen. Zum einen als Bildspeicher und zum anderen als Pseudo-Floppy.

Die Bildspeicher-Option ermöglicht es Ihnen, bis zu 16(!) komplette Bilder (jedes Bild ist 16KB groß!) abzuspeichern und in beliebiger Reihenfolge mit einer maximalen Wiederholrate von 3 Bildern pro Sekunde wieder darzustellen.

Über die Pseudo Floppy Option bekommen Sie die Möglichkeit an die Hand, eine bis zu 256KB große relative Datei anzulegen und frei zu verwalten. Natürlich können Sie auch durch eine geschickte Aufteilung der Datensatznummern mehrere kleine relative Dateien gleichzeitig benutzen.

**DIE ZUSÄTZLICHEN BASIC-BEFEHLE**

BANK  
BASIC  
BOS  
CALL  
COMMON  
DEV  
FAST  
FRAME  
GCHAR  
GOSUB  
GOTO  
GPAPER  
GPN  
ID  
LIST  
LOAD  
MASK  
MON  
NEW  
PEEK  
POKE  
RAMCLOSE  
RAMFIELD  
RAMOPEN  
RAMREAD  
RAMWRITE  
RECORDS  
RETURN  
RUN  
SAVE  
SCREEN.IN  
SCREEN.OUT  
SCREENS  
SLOW  
SPOOL.ON  
SPOOL.OFF  
UNMASK  
VIDEO.ON  
VIDEO.OFF

**WICHTIG:**

Unter BOS 1.0 sind die Zeichen ab dem ASCII-Wert 32 von vorne herein benutzerdefiniert (gleichbedeutend mit dem Befehl SYMBOL AFTER 32), d.h. der SYMBOL AFTER Befehl wird unter BOS 1.0 nicht mehr benötigt. Benutzen Sie ihn trotzdem, so erhalten Sie eine Fehlermeldung.

Bitte beachten Sie, daß alle bisher geschriebenen Programme ohne Änderungen nur unter Locomotive BASIC 1.0 einwandfrei arbeiten und nicht unter BOS 1.0. Daher sind diese Programme nur unter Locomotive BASIC 1.0 zu laden und zu starten. Dieses BASIC erhalten Sie nach jedem Reset (CTRL, SHIFT und ESC) oder nach der Eingabe von |BASIC <CR>.

Alle BASIC Befehle bis zum Kapitel "DER ROM-RESIDENTE Z80 - MONITOR" arbeiten nur unter BOS 1.0.

## DIE VORTEX BANK-BASIC ERWEITERUNGEN

=====

Die vortex BASIC Befehlsweiterungen beginnen alle mit einem senkrechten Strich "|", der durch gleichzeitiges Drücken der SHIFT-Taste und der Taste mit dem "Klammeraffen" ("@" ) erzeugt wird.

Folgende Abkürzungen werden bei den Befehlsbeschreibungen verwendet:

- l, m, n, o, ... Nummer einer Programmspeicherbank. Kann je nach Ausbaustufe zwischen 0 und 8 liegen und bedeutet jeweils eine 32KB große Programmspeicherbank.
- line Nummer einer BASIC Programmzeile.
- addr Adresse einer Speicherzelle.
- par1, ... Übergabe-Parameter bei einem CALL-Befehl
- chn Ausgabekanal-Nummer;  
chn=0 ist Bildschirm als Ausgabegerät (Defaultwert)  
chn=8 ist Drucker als Ausgabegerät  
chn=9 ist Floppy/Rekorder als Ausgabegerät
- i Bildspeicher-Nummer, kann je nach Ausbaustufe zwischen 0 und 15 liegen und bedeutet jeweils einen 16KB großen Bildspeicher
- string string steht für eine Stringvariable. Diese kann nicht direkt übergeben werden, sondern nur in der Form @var\$, wobei var\$ vorher ein Wert zugewiesen werden muß.
- reclng Datensatz- oder Recordlänge
- len1, ... reservierte Variablenlänge in einem RAMFIELD-Befehl
- recnum Datensatz- oder Recordnummer
- /.../ optionale Eingaben stehen in diesen "/.../"
- <CR> Drücken der ENTER-Taste

\*\*\*\*\*  
\* BANK \*  
\*\*\*\*\*

Format: |BANK, n

Funktion: Anwahl einer 32KB großen Programmspeicherbank. Die Zahl n läuft von 0...X-1, wobei X die Gesamtzahl, der vorhandenen Programmspeicherbanken ist.

Bemerkung: Das Kommando |BANK, n kann nur im Direkt-Mode verwendet werden und dient dazu, während des Programmierens zwischen verschiedenen Programmspeicherbanken hin- und herzuschalten. Die Eingabe von Programmzeilen bezieht sich immer auf die momentan aktivierte Programmspeicherbank.

Die Anzahl der verfügbaren Speicherbänke richtet sich nach dem Ausbaugrad der vortex RAM-Karte:

Typ	! Anzahl 32KB ! Programmspeicherbänke	! Summe ! Programmspeicher
SP64/M	2	64KB
SP64	2	64KB
SP128	3	96KB
SP256	5	160KB
SP320	6	192KB
SP512	9	288KB

Bank 0 bezeichnet bei allen Ausbaustufen den bereits auf der Grundplatine des CPC vorhandenen Speicher, während die Bänke mit Nummern zwischen 1 und 8 auf der Erweiterungskarte liegen.

**WICHTIG:** Die BASIC Variable HIMEM enthält auf Bank 0 den Wert 32616 und auf allen anderen Bänken 32767. Diese Werte dürfen mit dem MEMORY Befehl auf keinen Fall erhöht, aber durchaus nach Bedarf erniedrigt werden. Unter bestimmten Umständen kann HIMEM auf Bank 0 auch noch niedriger liegen (siehe Befehl |VIDEO.ON).

Beispiel:           \*\*\* Bank 0 aktiv \*\*\*

Ready  
|BANK,1 <CR>

\*\*\* Bank 1 aktiv \*\*\*

Ready

\*\*\*\*\*  
\*                   BASIC                   \*  
\*\*\*\*\*

Format:           | BASIC

Funktion:         Abschalten des BANK-BASIC.

Bemerkung:       Mit dem Befehl |BASIC setzen Sie den BASIC Interpreter in den Zustand nach dem Einschalten des Rechners zurück. Diesen Befehl können Sie dazu verwenden, das Kommando |BOS wieder rückgängig zu machen, d. h. das BANK-BASIC abzuschalten.

Beispiel:         | BASIC

\*\*\*\*\*  
\*          BOS          \*  
\*\*\*\*\*

Format:          | BOS

Bemerkung:      Mit dem Befehl |BOS (BASIC Operating System) wird das BANK-BASIC Betriebssystem eingeschaltet. Die Eingabe dieses Befehls ist Voraussetzung, um mit allen hier beschriebenen BASIC-Befehls-erweiterungen arbeiten zu können ! Mit der Eingabe des Befehls |BOS werden alle Daten im Speicher gelöscht. Es sollte also vor jedem Einschalten des BANK-BASIC darauf geachtet werden, daß sich keine relevanten Daten mehr im Speicher befinden, wenn ja dann müssen sie auf Diskette oder Cassette weggeschrieben werden. Nach der Eingabe dieses Befehls erscheint die Bereitschaftsmeldung des BANK-BASIC:

--- Bank-BASIC BOS 1.0      (C)1985 by vortex ---

anschließend wird die eingebaute RAM-Karte identifiziert. Dabei werden folgende Informationen ausgegeben:

-----  
! vortex RAM-Expansion Card ID!  
+-----+  
! w      xxxK      yyyK      zzK!  
! Bks    Prgm      Data      Spl!  
-----

- w ist die Anzahl der verfügbaren 32KB großen Programmspeicherbänke (Bks=Banks)
- xxx gesamter verfügbarer Programmspeicher in KB; w \* 32KB = xxxKB; (Prgm=Programmspeicher)
- yyy gesamter verfügbarer Datenspeicher in KB; (xxxKB + yyyKB) - 32KB = gesamter RAM-Speicher auf der RAM-Erweiterung
- zz Druckerpuffer in KB. Kann zwei Werte annehmen zz = 0KB oder zz = 32KB. Bedienung des Puffers erfolgt mit den Befehlen |SPOOL.ON und |SPOOL.OFF. Wird der Puffer eingeschaltet, so wird der Datenspeicher um 32KB kleiner.

Beispiel:      Ready  
              | BOS <CR>

\*\*\*\*\*  
 \* CALL \*  
 \*\*\*\*\*

Format: | CALL, n, addr/, par1, par2, ... /

Funktion: Aufruf eines Maschinen-Unterprogramms in Programmspeicherbank n, bei Adresse addr. Übergeben werden können die Parameter par1, par2, ... Es sind bis zu 30 Parameter zulässig.

Bemerkung: Dieser Befehl sollte nur von erfahrenen Programmieren eingesetzt werden, da er Kenntnisse in der Maschinenprogrammierung und Speicherstruktur des CPC voraussetzt.

Beispiel: 20 | CALL, 1, &6000

\*\*\*\*\*  
 \* COMMON \*  
 \*\*\*\*\*

Format: | COMMON/, string, l, m, n, ... /

Funktion: Mit dem Befehl |COMMON lassen sich Variablen vom Typ REAL, INTEGER oder STRING allen vorhandenen oder auch nur speziellen Programmspeicherbänken zur Verfügung stellen. (COMMON=gemeinsam).

**WICHTIG:** Felder sind immer nur lokal definierbar und können nicht direkt übertragen werden. Möchte man dennoch Feldelemente übertragen, so muß man diese vorher einer STRING-, REAL- oder INTEGER-Variable vom Typ COMMON zuweisen.

Bemerkung: Es lassen sich gezielt Variablengruppen auswählen, die in allen Programmspeicherbänken zugänglich sein sollen. Je nach Befehlsaufruf hat der Befehl |COMMON ganz verschiedene Wirkungen.

Wird nur |COMMON eingegeben, so haben alle Variablen nur in der Bank eine Bedeutung in der sie definiert wurden, d.h. alle Variablen sind nur lokal gültig. Es ist also durchaus möglich in jeder Bank eine Variable hugo\$ zu haben, die jedesmal einen anderen Wert besitzt.

|COMMON,@string\$ erklärt Variablengruppen als gültig in allen (!) Speicherbänken. Das Gruppenkriterium ist hierbei der Anfangsbuchstabe.

Die bei |COMMON zu übergebende Variable hat folgende Struktur:

string\$="b1, b2, b3-b4, b5, b6-b7... "

Hierbei stehen b1, b2, ... für die Anfangsbuchstaben der verschiedenen Variablengruppen.

|COMMON,@string\$, l, m, n, ... erklärt Variablengruppen



als gültig in den Bänken l,m,n,...

WICHTIG: Da beim Umschalten der Bänke alle als COMMON deklarierten Variablen direkt von Bank zu Bank transferiert werden, hängt die Geschwindigkeit des Umschaltvorgangs entscheidend von der Anzahl der COMMON-Variablen ab.

Beispiel:

a.) 10 ;COMMON

- alle Variablen nur noch lokal gültig

b.) 100 A\$="B,K-P,Z"  
110 ;COMMON,@A\$

- alle Variablen, die mit B,Z oder K-P beginnen gelten auf allen Bänken.

c.) 100 A\$="B,K-P,Z"  
110 ;COMMON,@A\$,0,2

- alle Variablen, die mit B,Z oder K-P beginnen gelten nur auf den Bänken 0 und 2.

Hat z.B. die Variable BETA auf der Bank 0 den Wert 123.456, so hat sie diesen Wert auch auf Bank 2, nicht aber auf allen anderen Bänken.

\*\*\*\*\*  
\* DEV \*  
\*\*\*\*\*

Format: ;DEV/,chn/

Funktion: Zuweisung eines Ausgabekanals für den erweiterten ;LIST Befehl.

Bemerkung: Das Ausgabegerät (DEvice) kann sein

- chn=0 Bildschirm
- chn=8 Drucker
- chn=9 Floppy/Rekorder

Weglassen der Kanalnummer selektiert automatisch Kanal 0, also den Bildschirm.  
Ein eingestellter Kanal bleibt bis zum nächsten ;DEV oder einem System Reset aktiv.

Beispiel: 100 ;DEV,8 <CR>

\*\*\*\*\*  
\*                  GOSUB                  \*  
\*\*\*\*\*

Format:          | GOSUB/, n/, line

Funktion:          Aufruf eines Unterprogramms in einer bestimmten  
                  Programmspeicherbank.

Bemerkung:          Mit dem Befehl |GOSUB kann in der Programmbank n,  
                  bei der BASIC-Zeile line ein Unterprogramm aufgeru-  
                  fen werden. Am Ende dieses Unterprogramms muß  
                  der Befehl |RETURN stehen, damit der Programmab-  
                  lauf in der Programmspeicherbank und nach der  
                  Zeile, von der aus das |GOSUB erfolgte, weiter-  
                  gehen kann.  
                  Da line variabel sein darf, ist es hiermit erst-  
                  mals auch möglich, eine direkte Programmbeeinflus-  
                  sung von BASIC aus vorzunehmen.  
                  Mit |GOSUB aufgerufene Programmsegmente können bis  
                  zu 30 Mal verschachtelt werden.

Beispiel:          a.) 20 |GOSUB, 1, 125  
                  b.) 10 INPUT A  
                      20 |GOSUB, 1, A

                  Abhängig von A verzweigt der Programmablauf  
                  in das Unterprogramm bei Zeile A, Bank 1.

\*\*\*\*\*  
\*                  GOTO                  \*  
\*\*\*\*\*

Format:          | GOTO/, n/, line

Funktion:          Umleitung des Programmablaufs in eine bestimmte  
                  Programmspeicherbank zur BASIC-Zeile line.

Bemerkung:          Mit dem Befehl |GOTO wird der Programmablauf in  
                  die Bank n zur Zeile line umgeleitet.  
                  Dort beginnt dann ein neues Segment des Programms.  
  
                  Da line variabel sein darf, ist es hiermit erst-  
                  mals auch möglich, eine direkte Programmbeeinflus-  
                  sung von BASIC aus vorzunehmen.

Beispiel:          10 REM  
                  20 |GOTO, 2, 1985  
  
                  10 line=1000  
                  20 |GOTO, 4, line

\*\*\*\*\*  
 \* ID \*  
 \*\*\*\*\*

Format: | ID

Funktion: Ausgabe des "Steckbriefs" der eingebauten vortex RAM-Erweiterung.

Bemerkung: siehe ID (Identifikation) beim Befehl |BOS.

Beispiel: 30 | ID

\*\*\*\*\*  
 \* LIST \*  
 \*\*\*\*\*

Format: | LIST/, l, m, n, o, ... /

Funktion: Listet ein Programm durch alle Bänke auf.

Bemerkung: Mit diesem erweiterten |LIST-Befehl können am Stück alle Programme, oder Programmteile, die in den einzelnen Programmspeicherbänken liegen aufgelistet werden. Mit dem erweiterten BASIC-Befehl |DEV kann bestimmt werden, auf welchem Ausgabegerät gelistet werden soll. Werden keine Programmbank-Nummern angegeben, so werden alle Programmbänke gelistet, in denen sich Programmzeilen befinden. Um innerhalb einer Programmbank ein Programm anzusehen, verwendet man den herkömmlichen LIST-Befehl.

Beispiel: \*\*\* Bank 0 aktiv \*\*\*

```
Ready
1 A$="A"
5 |COMMON,@A$
10 REM >> dieser Programmteil steht in Bank 0 <<
20 For A=1 TO 100
30 |GOSUB,1,20
40 NEXT A
|BANK,1
```

\*\*\* Bank 1 aktiv \*\*\*

```
Ready
10 REM >> dieser Programmteil steht in Bank 1 <<
20 PRINT "Das Quadrat von ";A;" ist ",A*A
30 |RETURN
```

| LIST

----- BANK 0 -----

```

1  A$="A"
5  |COMMON,@A$
10 REM >> dieser Programmteil steht in Bank 0 <<
20 For A=1 TO 100
30 |GOSUB,1,20
40 NEXT A

```

----- BANK 1 -----

```

10 REM >> dieser Programmteil steht in Bank 1 <<
20 PRINT "Das Quadrat von ";A;" ist ",A*A
30 |RETURN

```

\*\*\* Bank 1 aktiv \*\*\*

Ready

```

*****
*          LOAD          *
*****

```

Format: | LOAD, name

Funktion: Laden eines Programms, das zuvor mit dem erweiterten BASIC-Befehl |SAVE abgespeichert wurde. Das Programm besteht aus mehreren Teilen, die in verschiedenen Programmspeicherbänken liegen.

Bemerkung: Liegt ein Programm vor, daß z. B. aus drei Teilen, in drei verschiedenen Programmbänken besteht, so hat der |SAVE Befehl vier Dateien auf der Diskette angelegt (das Programm heißt z. B. TEST):

TEST.BAS TEST.BK0 TEST.BK1 TEST.BK2

Die Datei TEST.BAS ist der sogenannte "Lader". In ihr steht die Information in welche Programmbank welcher Programmteil gehört. Für den Benutzer ist diese Zuordnung durch die Extensionen ("BK0"=Bank 0, "BK1"=Bank 1, "BK2"=Bank 2) leicht erkennbar. Diese Dateien können selbstverständlich auch mit dem normalen LOAD-Befehl einzeln geladen werden. Dabei ist aber darauf zu achten, daß jeder Programmteil in die richtige Programmbank geladen wird.

Beispiel: 10 A\$="TEST"  
20 |LOAD,@A\$

\*\*\*\*\*  
\* NEW \*  
\*\*\*\*\*

Format: ;NEW/,1,m,n,o,.. /

Funktion: Löschen der Programmspeicherbänke.

Bemerkung: Mit dem erweiterten ;NEW-Befehl können die Inhalte der Programmspeicherbänke gelöscht werden. Optional lassen sich selektiv einzelne Bänke löschen.

WICHTIG: Wenn Sie in einem BASIC Programm den Befehl ;GOSUB verwendet haben und nun ein neues Programm schreiben möchten, so müssen (!) Sie den Speicher mit ;NEW löschen.

Beispiel: ;NEW,1,3

Löschen der Programmbänke 1 und 3, alle anderen bleiben unverseht.

\*\*\*\*\*  
\* PEEK \*  
\*\*\*\*\*

Format: ;PEEK,n,addr,@var

Funktion: weist der Integervariable var den Inhalt der Speicherzelle bei der Adresse addr in der Speicherbank n zu.

Bemerkung: Mit dem erweiterten ;PEEK-Befehl können Sie Speicherzelleninhalte von einer beliebigen Bank holen, egal ob Sie sich augenblicklich auf dieser Bank befinden oder nicht.

Beispiel: ;PEEK,2,&4000,@a

\*\*\*\*\*  
\* POKE \*  
\*\*\*\*\*

Format: ;POKE,n,addr,wert

Funktion: weist der Speicherzelle bei addr in der Bank n den Wert wert zu.

Bemerkung: Mit dem erweiterten ;POKE-Befehl können Sie Speicherzelleninhalte einer beliebigen Bank ändern, egal ob Sie sich augenblicklich auf dieser Bank befinden oder nicht.

Beispiel: ;POKE,2,&4000,&FF

\*\*\*\*\*  
\*                   RAMCLOSE                   \*  
\*\*\*\*\*

Format:            ; RAMCLOSE

Funktion:           Schließen der relativen RAM-Datei.

Bemerkung:         Der Datenspeicher auf der RAM-Erweiterung kann auch als relative Datei verwendet werden. Das ist eine Datei die aus Datensätzen bestimmter Länge besteht. Ein Datensatz, oft auch als "Record" bezeichnet, kann über eine Nummer ausgewählt werden. Diese Technik bringt im Vergleich zu sequentieller Dateiverwaltung eine enorme Geschwindigkeitssteigerung des Datenaustausches. Die zuvor mit ;RAMOPEN geöffnete Datei, sollte nach der Bearbeitung wieder mit ;RAMCLOSE geschlossen werden. Tut man dies nicht, so bleibt der Datenspeicher als RAM-Disk deklariert und die Gruppe der ;VIDEO-Befehle kann nicht angesprochen werden. Vergessen Sie nicht Ihre Daten aus der RAM-Disk auf Cassette oder Diskette zu sichern, bevor Sie die ;VIDEO-Befehle benutzen, da sie sonst zerstört werden.

Weitere Befehle sind ;RAMOPEN, ;RAMFIELD, ;RAM-READ, ;RAMWRITE und ;RECORDS.

Beispiel:         10 ;RAMOPEN, 128  
                  .  
                  .  
                  .  
                  90 ;RAMCLOSE

\*\*\*\*\*  
\*                   RAMFIELD                   \*  
\*\*\*\*\*

Format:            ; RAMFIELD, len1, len2, ...

Funktion:           Einteilung eines Datensatzes (records) in einzelne Felder.

Bemerkung:         Dieser Befehl dient zur Strukturierung der Datensätze der relativen RAM-Datei. Dieser Befehl darf erst nach dem ;RAMOPEN-Befehl im Programm stehen. Ein Datensatz mit einer Länge von z.B. 256Byte wird z.B. mit len1=30, len2=40 und len3=20 in drei unterschiedlich lange Felder zerlegt. Die Befehle ;RAMREAD und ;RAMWRITE benutzen in diesem Fall drei Variablen den drei Feldern eines Datensatzes entsprechend. Die Summe aller Feldlängen eines Datensatzes darf logischer Weise die gesamte Länge des Datensatzes nicht überschreiten.

Beispiel: 10 | RAMOPEN, 128  
20 | RAMFIELD, 32, 32, 64  
. . .  
90 | RAMCLOSE

\*\*\*\*\*  
\* RAMOPEN \*  
\*\*\*\*\*

Format: | RAMOPEN, reclng

Funktion: Öffnen der relativen RAM-Datei.

Bemerkung: Bevor mit der relativen RAM-Datei gearbeitet werden kann, muß diese geöffnet, d.h. für Datenaustausch vorbereitet werden. Mit reclng wird die Datensatzlänge angegeben, die dann durch den Befehl |RAMFIELD weiter strukturiert wird. reclng kann zwischen 8 und 1048 liegen.

WICHTIG: Es ist nicht möglich, die Gruppe der Pseudo-Disk und die der |VIDEO-Befehle gleichzeitig zu benutzen.

Beispiel: 10 | RAMOPEN, 512

\*\*\*\*\*  
\* RAMREAD \*  
\*\*\*\*\*

Format: | RAMREAD, recnum, @var1, @var2, ...

Funktion: Einlesen eines Datensatzes in die Feldvariablen var1, var2, ...

Bemerkung: Durch die Zahl recnum wird der Datensatz ausgewählt, der gelesen werden soll. Entsprechend den mit |RAMFIELD definierten Unterfeldern wird der Datensatz in die Feldvariablen var1, var2, ... eingelesen.

recnum darf zwischen 0 und REC-1 liegen. Die Systemvariable REC erhält man durch den Befehl |RECORDS, der erst nach dem |RAMOPEN kommen darf und die maximal verfügbare Anzahl von Datensätzen der angewählten Länge (reclng) angibt. Neben der Datensatzlänge, hängt diese Zahl natürlich auch von dem Bestückungsgrad der RAM-Erweiterung ab.

WICHTIG: var1, var2, ... müssen STRING-Variable sein und vor der ersten Benutzung des RAMREAD-Befehls definiert werden.

Beispiel: 100 | RAMREAD, 45, @A\$, @B\$, @V\$

\*\*\*\*\*  
\*           RAMWRITE           \*  
\*\*\*\*\*

Format:           ; RAMWRITE, recnum, @var1, @var2, ..

Funktion:         Schreiben eines Datensatzes in die relative RAM-Datei.

Bemerkung:        Der Inhalt der Feldvariablen var1, var2, .. wird in die entsprechenden Felder des Datensatzes mit der Nummer recnum geschrieben. recnum darf zwischen 0 und REC-1 liegen. Die Systemvariable REC erhält man durch den Befehl ;RECORDS, der nach einem ;RAMOPEN Befehl stehen muß und die maximal verfügbare Anzahl von Datensätzen der angewählten Länge (reclng) angibt. Neben der Datensatzlänge, hängt diese Zahl natürlich auch von dem Bestückungsgrad der RAM-Erweiterung ab.

WICHTIG: var1, var2, ... müssen STRING-Variable sein.

Beispiel:         100 ; RAMWRITE, 20, @A\$, @B\$, @EC\$

\*\*\*\*\*  
\*           RECORDS           \*  
\*\*\*\*\*

Format:           ; RECORDS

Funktion:         Errechnet nach einem ;RAMOPEN Befehl die mögliche Anzahl von Datensätzen.

Bemerkung:        Wird nach einem ;RAMOPEN, reclng Befehl der Befehl ;RECORDS eingegeben, so wird aus den beiden Parametern reclng und Datenspeicherkapazität auf der Karte, die Zahl der möglichen Datensätze errechnet und in der Systemvariablen REC abgelegt.

Beispiel:         10 ; RAMOPEN, 128  
                  40 ; RECORDS  
                  50 PRINT REC

\*\*\*\*\*  
\*           RETURN           \*  
\*\*\*\*\*

Format:           ; RETURN

Funktion:         Rücksprung aus einem Unterprogramm

Bemerkung:        Dieser Befehl muß am Ende eines Unterprogramms stehen, daß mit ;GOSUB, n, line aufgerufen wurde. Der Programmablauf kehrt dann an die BASIC-Zeile zurück, die nach der Zeile mit dem ;GOSUB steht.



**WICHTIG:** Der Basic Befehl RETURN (also ohne ein |) ist im Zusammenhang mit dem erweiterten Befehl |GOSUB (s. o.) unwirksam.

Beispiel:       45 |RETURN

\*\*\*\*\*  
\*                RUN                \*  
\*\*\*\*\*

Format:         | RUN, string/, n, line/

Funktion:       Laden und Starten eines BASIC-Programms.

Bemerkung:     Ein zuvor mit |SAVE gespeichertes Programm, wird mit dem Befehl |RUN geladen und gestartet. Lässt man die Bank- und Zeilennummer weg, so wird das Programm bei der ersten BASIC Zeile der momentan aktiven Bank gestartet. Gibt man nur eine Banknummer an, so wird das Programm in dieser Bank bei der ersten BASIC Zeile gestartet. Spezifiziert man auch noch die Zeilennummer, so kann ein Programm auf einer ganz bestimmten Bank in einer ganz bestimmten Zeile gestartet werden.

Beispiel:       A\$="TEST"  
                 | RUN, @A\$

\*\*\*\*\*  
\*                SAVE               \*  
\*\*\*\*\*

Format:         | SAVE, name/, l, m, n, o, . . . /

Funktion:       Abspeichern eines Programms, das auf mehrere Programmspeicherbänke verteilt ist.

Bemerkung:     Mit Hilfe des |SAVE-Befehls kann ein Programm, das mehrere Programmspeicherbänke belegt in einem Arbeitsgang auf Diskette/Cassette abgespeichert werden. Dabei wird für jede Programmspeicherbank, die von diesem Programm belegt wird, eine eigene Datei angelegt. Haben wir ein Programm - es heißt z. B. TEST -, das 3 Programmspeicherbänke belegt, so erzeugt der |SAVE-Befehl folgende Dateien:

TEST.BAS   TEST.BK0   TEST.BK1   TEST.BK2

Die Datei TEST.BAS ist der sogenannte "Lader". In dieser Datei steht, welche der Dateien TEST.BK0..TEST.BK2 in welche Programmspeicherbank gehört. Die Extensionen BK0...BK2 stehen für die Bänke 0..2.

Durch die Option l,m,n,o... kann festgelegt werden, daß nur bestimmte Programmspeicherbänke abgespeichert werden. Wird hier nichts angegeben, so werden alle Programmspeicherbänke abgespeichert, die von diesem Programm belegt sind.

Beispiel:        A\$="TEST"  
                   !SAVE,@A\$

\*\*\*\*\*  
 \*                SCREEN.IN        \*  
 \*\*\*\*\*

Format:         ! SCREEN.IN, i

Funktion:        Einladen eines kompletten 16KB großen Bildes aus einer Bildspeicherbank.

Bemerkung:      Bevor mit diesem Befehl gearbeitet werden kann, muß der Datenspeicher erst mit dem Befehl !VIDEO.ON zum Bildspeicher erklärt werden. Der Befehl liest in ca 0.30 sec ein komplettes 16KB großes Bild aus der durch i definierten Bildspeicherbank. Das Bild wurde zuvor mit dem Befehl !SCREEN.OUT in diese Bildspeicherbank geschrieben. Die Zahl i kann zwischen 0 und SCR liegen. SCR ist eine Systemvariable, die durch den Befehl !SCREENS angelegt wird und die höchst mögliche Bildnummer, abhängig von dem Bestückungsgrad der RAM-Karte, enthält.

Beispiel:        10 !VIDEO.ON  
                   20 FOR I=1 TO 6  
                   30 !SCREEN.IN,I  
                   40 NEXT I

\*\*\*\*\*  
 \*                SCREEN.OUT       \*  
 \*\*\*\*\*

FORMAT:         ! SCREEN.OUT, i

Funktion:        Speichern eines kompletten 16KB großen Bildes in einer Bildspeicherbank.

Bemerkung:      Bevor mit diesem Befehl gearbeitet werden kann, muß der Datenspeicher erst mit dem Befehl !VIDEO.ON zum Bildspeicher erklärt werden. Der Befehl schreibt ein komplettes 16KB großes Bild in die durch i angegebene Bildspeicherbank. Mit dem Befehl !SCREEN.IN,i kann dieses Bild in ca. 0.3 sec wieder ausgelesen werden. Die Zahl i kann zwischen 0 und SCR liegen. SCR ist eine Systemvariable, die durch den BEFEHL !SCREENS angelegt wird und die höchst mögliche Bildnummer, abhängig von dem Bestückungsgrad der RAM-Karte, enthält. Bei der Karte SP512 ist SCR=16. D.h. es können mit dieser Karte 17(!) komplette Bilder abgespeichert werden.

Beispiel:        10 !SCREEN.OUT,2

\*\*\*\*\*  
\* SCREENS \*  
\*\*\*\*\*

Format: | SCREENS

Funktion: Berechnen der höchst möglichen Bildnummer

Bemerkung: Um beim Arbeiten mit den Befehlen |SCREEN.IN und |SCREEN.OUT zu wissen, wieviele Bildspeicherbänke zur Verfügung stehen, kann mit dem Befehl |SCREENS die Systemvariable SCR angelegt werden. Es gilt: SCR+1=Anzahl der möglichen Bilder.

Beispiel: 10 |SCREENS  
20 PRINT SCR

\*\*\*\*\*  
\* SPOOL.ON \*  
\*\*\*\*\*

Format: | SPOOL.ON

Funktion: Schaltet den 32KB großen BASIC Druckerpuffer ein.

Bemerkung: Alle Ausgaben, die auf den Drucker gehen sollen, werden, nachdem der Befehl |SPOOL.ON gegeben wurde, in diesem 32KB großen Puffer zwischengespeichert. Dadurch kann gleichzeitig gedruckt und mit dem CPC 464 gearbeitet werden. Die 32KB werden dabei vom Datenspeicher "abgezackt". Man sieht das deutlich dann, wenn man den Spooler eingeschaltet hat und dann den Befehl |ID eingibt.

**WICHTIG:** Da der Spooler den Datenspeicher um 32KB verkleinert, sollte man sich vor der Benutzung dieses Befehls darüber im Klaren sein, ob man die Daten in dem vom Spooler belegten Bereich noch benötigt. Benutzen Sie den Datenspeicher gerade als Bildspeicher (|VIDEO.ON), so verkleinert sich die Anzahl der verfügbaren Bilder um zwei. Die Bilder mit den beiden höchsten Bildspeichernummern werden von den Druckerdaten überschrieben.

Sind Sie gerade im RAM-Floppy Mode (|RAMOPEN), so verringert sich die Anzahl der Datensätze entsprechend der von Ihnen definierten Datensatzlänge.

Die verfügbare Anzahl der Bilder bzw. Datensätze können Sie über die Befehle |SCREENS/|RECORDS erfahren.

Beispiel: |SPOOL.ON

\*\*\*\*\*  
\* SPOOL. OFF \*  
\*\*\*\*\*

Format: ; SPOOL. OFF

Funktion: Abschalten des 32KB großen Druckerpuffers.

Bemerkung: Der zuvor mit ; SPOOL. ON eingeschaltete Druckerpuffer kann mit ; SPOOL. OFF wieder abgeschaltet werden. Die 32KB werden dann wieder an den Datenspeicher zurückgegeben.

Beim Abschalten des Spoolers vergrößert sich automatisch wieder die Anzahl der verfügbaren Bilder bzw. Datensätze.

Beispiel: ; SPOLL. OFF

\*\*\*\*\*  
\* VIDEO. ON \*  
\*\*\*\*\*

Format: ; VIDEO. ON

Funktion: Benutzung des Datenspeichers als mehrfacher Bildspeicher.

Bemerkung: Mit diesem Befehl wird der gesamte Datenspeicher in 16KB große Bildspeicherbänke aufgeteilt, die dann mit den Befehlen ; SCREEN. IN und ; SCREEN. OUT, sowie mit dem Befehl ; SCREENS bearbeitet werden können.

WICHTIG: die Gruppe der ; VIDEO-Befehle kann nur von Bank 0 aus benutzt werden. ; VIDEO. ON setzt den Bildspeicher von &C000 nach &4000 herunter. Dadurch sinkt HIMEM auf 16360 und darf auch nicht über diesen Wert gesetzt werden.

Beispiel: 10 ; VIDEO. ON

\*\*\*\*\*  
\* VIDEO. OFF \*  
\*\*\*\*\*

Format: ; VIDEO. OFF

Funktion: Abschalten des zusätzlichen Bildspeichers.

Bemerkung: Der zuvor als mehrfacher Bildspeicher benutzte Datenspeicher, wird wieder als reiner Datenspeicher benutzt.

Beispiel: 10 ; VIDEO. ON

15 i=3  
20 ; SCREEN. IN, I

90 ; VIDEO. OFF

**DER ROM-RESIDENTE Z80 - MONITOR**

-----

Der eingebaute Monitor, mit dem Maschinensprachprogramme geladen, getestet und wieder abgespeichert werden können, ist ein leistungsfähiges Werkzeug für Programmierer, das sofort und ohne wesentlichen Speicherplatzverlust zur Verfügung steht. Er beinhaltet vielerlei Funktionen, wie z. B. Disassemblieren, Anzeigen und Ändern von Speicher- und Registerinhalten, Setzen von Breakpoints, Einzelschrittausführung, etc. Diese werden nachfolgend ausführlich erklärt. Der Monitor wird mit dem RSX-Befehl !MON gestartet ( ! erreichen Sie mit SHIFT und der Taste mit dem "Klammeraffen" ("@" ) und meldet sich mit dem Zeichen \* und einem blinkendem Cursor. Das \* - Zeichen wird im folgendem als Prompt bezeichnet. Sie können den Monitor durch Drücken der ESC-Taste immer dann verlassen, wenn dieser Prompt sichtbar ist und befinden sich danach an der Stelle nach dem Aufruf (d. h. der Monitor kann auch aus einem BASIC-Programm heraus aufgerufen werden und das Programm läuft nach der Rückkehr weiter).

**Die Befehle des Monitors**

-----

Nachfolgend werden in alphabetischer Reihenfolge die Befehle des Monitors erklärt. Zu beachten ist dabei, daß geforderte oder angezeigte Werte grundsätzlich in hexadezimaler Form dargestellt werden, wobei 8-Bit-Werte zweistellig und 16-Bit-Werte vierstellig sind, bzw. sein müssen.

**A adresse - Zeilenassembler**

Mit dem Zeilenassembler können Sie Z80-Befehle direkt in lesbaren Mnemonics eingeben, der Befehl wird dann sofort übersetzt und ab der angegebenen Adresse in den Speicher eingetragen. Sie können dann ab der nächsten Adresse nach dem letzten übersetzten Befehl den nächsten Z80-Befehl eingeben, können aber auch mit der ESC-Taste wieder zum Monitor-Prompt \* zurückkehren. Bei der Benutzung des Zeilenassemblers ist allerdings zu beachten, daß keine Labels (wie bei größeren Assemblern) verwendet werden können, sondern nur fixe Adressen und, daß Zahlenwerte immer in hexadezimaler Form mit führendem &-Zeichen eingegeben werden müssen.

**Beispiele:**

A2400 <ENTER>	Benutzung des Assemblers ab Adresse 2400H
2400 LD A, &65 <ENTER>	in Adressen 2400 u. 2401 steht 3E und 65
2402 LD H, A <ENTER>	in Adresse 2402 steht 67.
2403 EX DE, HL <ENTER>	in Adresse 2403 steht EB.
2404 JR &2400 <ENTER>	in Adressen 2404 u. 2405 steht 18 und FA.
2405 <ESC>	Abbruch des Zeilenassemblers und
*	Rückkehr zum Monitor-Prompt.

**B - Breakpoints ansehen/ändern**

Der Monitor bietet die Möglichkeit bis zu acht Breakpoints (=Unterbrechungspunkte) beliebig zu setzen. Trifft der Monitor beim Ablauf eines Programmes (siehe G-Befehl) auf einen dieser Breakpoints, so wird der Programmablauf unterbrochen und der Monitor-Prompt \* ausgegeben. Die Ausgabe der Breakpoints erfolgt in der Form:

-B1-	-B2-	-B3-	-B4-	-B5-	-B6-	-B7-	-B8-
0000	0000	0000	0000	0000	0000	0000	0000

Sie können dann den ersten Breakpoint ändern oder mittels <ENTER> zum nächsten Breakpoint gelangen, usw. bis nach dem achten Breakpoint oder durch Drücken von <ESC> der Monitor-Prompt \* wieder erscheint.

**D adresse1, adresse2 - Auflisten in hexadezimalen und ASCII-Format**

Der D-Befehl gibt Ihnen die Möglichkeit Speicherinhalte in folgendem Format anzuzeigen:

```
0100 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
```

Am Anfang der Zeile wird Ihnen die Adresse der momentanen Speicherzelle angezeigt und darauf die Inhalte dieser und der fünfzehn folgenden Speicherzellen in hexadezimaler Form. Danach erfolgt eine Interpretation der Werte als ASCII-Zeichen; ist der Wert kleiner als 32 oder grösser als 127, so wird ein Punkt, ansonsten aber das entsprechende ASCII-Zeichen ausgegeben. Geben Sie nach Drücken der Taste D nur eine Adresse ein, so werden die nachfolgenden 100H Speicherinhalte ausgegeben, drücken Sie allerdings nach der ersten Adresse nicht <ENTER>, sondern die Komma-Taste, so können Sie eine zweite Adresse eingeben. Es werden dann von der ersten Adresse bis zur zweiten Adresse die Speicherinhalte aufgelistet. Eine solche Auflistung können Sie durch Drücken einer beliebigen Taste kurzfristig anhalten. Drücken Sie nun <ESC>, so erscheint wieder der Monitor-Prompt \*, drücken Sie aber eine beliebige andere Taste, so läuft das Listing weiter.

**F adresse1, adresse2, wert - Auffüllen eines Speicherbereichs mit einem Wert**

Oft kommt es vor, daß bestimmte Speicherbereiche z.B. gelöscht werden sollen. Dies ist mit dem F-Befehl möglich, wobei von adresse1 bis adresse2 die Speicherinhalte mit der 8-Bit-Konstanten wert belegt werden.

Beispiel:

```
FC00,FFFF,FF <ENTER> füllt den Bildschirmspeicher (Adressen C000
* mit dem Wert FF.
```

### G adresse

Dieser Befehl führt ein ab der eingegebenen Adresse befindliches Maschinensprachprogramm aus. Wird hierbei ein Breakpoint (s.o.) erreicht, so geht die Kontrolle an den Monitor zurück.

Beispiel:

GBD19 <ENTER> führt das Maschinensprachprogramm ab der Adresse BD19 aus.

### I filename - Laden einer Binärdatei von Diskette/Cassette

Wurde der Monitor aus Basic heraus aufgerufen, so kann mit diesem Befehl vom selektierten Eingabegerät (Diskette oder Cassette) eine Binärdatei geladen werden. Sie wird dabei in den originalen Speicher, d.h. in den Speicherteil aus dem sie erstellt wurde, geladen. Es werden die Ladeadresse, die Länge, sowie eine eventuell vorhandene Startadresse angezeigt.

Beispiel:

ITEST.BIN <ENTER> lädt die Datei TEST.BIN in den Speicher.  
<2000,0380,2140> Die ermittelte Anfangsadresse ist 2000,  
\* die Länge 380 und die Startadresse 2140.

### L adresse1, adresse2 - Disassemblieren eines Speicherbereiches

Die Eingabe und Bedeutung der Adressen entspricht denen des D-Befehls (s.o.), wobei allerdings der Speicherinhalt in Form von Mnemonics dargestellt wird.

Beispiel:

```
L0100,0106 <ENTER>
0100 3E64 LD A,64H
0102 ED5B3204 LD BC,(0432H)
0106 E3 EX (SP),HL
*
```

Listet den Bereich von 100 bis 106. Am Anfang steht die Adresse, dann deren Inhalt in hexadezimaler Form und dann der lesbare Z80-Mnemonic.

### M adresse1, adresse2, adresse3

Der Inhalt des Speicherbereiches zwischen adresse1 und adresse2 wird in den Speicherbereich beginnend mit adresse3 kopiert.

Beispiel:

M0000,3FFF,C000 <ENTER> kopiert den Bereich von 0 - 3FFF in den Speicherbereich ab Adresse C000 (=Bildschirm).  
\*





Monitor zurückgegeben, d. h. Sie sehen die eventuell geänderten Register und können die Abarbeitung auch unterbrechen, wenn Sie eine optionelle Schrittzahl angegeben haben. Diese Unterbrechung erreichen Sie wie gewohnt mit ESC. Beachten Sie bitte, daß Programme im ROM oder Befehle die mit dem Zweitregistersatz des Z80 arbeiten nicht schrittweise abgearbeitet werden können. Desweiteren sollte der Stackpointer SP nicht unter einem ROM liegen, was zum "Absturz" des Systems führen kann.

Beispiel:

T#000,0007

führt sieben Schritte ab Adresse 4000 aus. Die Registerinhalte werden immer angezeigt und der Ablauf kann wie beim D-Befehl angehalten oder abgebrochen werden.

**DIE ZUSÄTZLICHEN BASIC-GRAPHIKBEFEHLE**  
\*\*\*\*\*

\*\*\*\*\*  
\*           **FAST**           \*  
\*\*\*\*\*

Format:       | FAST

Funktion:     verschnellerte Bildschirmausgabe im Mode 2

Bemerkung:    Wenn Sie sich in der 80-Zeichen-Bildschirmausgabe (Mode 2) befinden, haben Sie die Möglichkeit, die Ausgabe von Text um den Faktor 2 zu beschleunigen. Der einzige Nachteil, den Sie in Kauf nehmen müssen, ist eine nicht mehr einwandfrei arbeitende Window-Technik. Bei normaler Ausgabe von Text ohne definierte Windows arbeitet die verschnellerte Bildschirmausgabe vollgültig wie die standardmäßige Ausgabe. Die normale Ausgabe kann mit dem Befehl | SLOW wieder aktiviert werden.

Beispiel:     Als Demonstration für die Verschnellerung können Sie folgenden "Einzeiler" eingeben:

```
10 Z=Z+1: ?Z: GOTO 10
```

Starten Sie dieses Programm mit RUN, so sehen Sie die normale Ausgabe. Unterbrechen Sie das Programm nun, tippen |FAST <ENTER> und starten das Programm wieder mit RUN <ENTER>, so werden Sie die drastische Geschwindigkeitserhöhung sofort feststellen können.

\*\*\*\*\*  
\*           **FRAME**           \*  
\*\*\*\*\*

Format:       | FRAME

Funktion:     erzeugen von flackerfreier bewegter Graphik

Bemerkung:    Dadurch das die Ausgabe auf dem Bildschirm normalerweise nicht mit dem physikalischen Bildschirmaufbau des Monitors synchronisiert wird, kann es in bestimmten Fällen zu einem flackernden Bildschirmaufbau kommen.

Beispiel:     Als Beispiel können Sie folgendes kurzes Programm eingeben:

```
10 MODE 2
20 FOR I=1 TO 80
30 LOCATE I, 12: ?"o": LOCATE I, 12: ?" "
40 NEXT
50 GOTO 20
```

Wenn Sie das Programm starten, wird das Zeichen o relativ ruckhaft von links nach rechts bewegt. Fügen Sie aber die folgende Zeile ein, so werden Sie nach dem Neustart des Programmes eine vollkommen fließende Bewegung sehen. Die Zeile lautet:

```
35 |FRAME
```

```
*****
*          GCHAR          *
*****
```

Format: |GCHAR, x, y, @var

Funktion: lesen eines Zeichens vom Bildschirm

Bemerkung: Wenn Sie beispielsweise feststellen wollen, welches Zeichen sich an der Position 12,14 auf dem Bildschirm befindet, so können Sie dies mit diesem Befehl ermitteln. Sie müssen dazu zunächst eine Variable definieren, in die der ASCII-Wert des ermittelten Zeichens dann eingetragen wird. Diese Variable muß eine Integer-Variable sein, d.h. bevor Sie den |GCHAR-Befehl benutzen, müssen Sie beispielsweise die Variable A wie folgt definieren:

```
10 DEFINT A: A=0
```

Benutzen Sie danach den Befehl |GCHAR, X, Y, @A, dann wird der ASCII-Wert des Zeichens an den Koordinaten X/Y in die Variable A eingetragen. Diesen Wert können Sie dann mit PRINT A ansehen. Bitte beachten Sie, daß unzulässige Koordinaten (Y zwischen 1 und 25, X zwischen 1 und dem Maximalwert im jeweiligen Mode) oder durch das Anwenden von Graphikbefehlen nicht mehr identifizierbare Zeichen den Wert 0 zurückmelden.

```
*****
*          GPAPER         *
*****
```

Format: |GPAPER, wert

Funktion: Einstellen der Graphik-Hintergrundfarbe

Bemerkung: Mit dem Befehl |GPAPER, wert weisen Sie der Graphik-Hintergrundfarbe einen neuen Farbstift zu. Eine ausführliche Erklärung finden Sie beim |MASK-Befehl.

Beispiel: |GPAPER, 3

der neue Farbstift für die Graphik-Hintergrundfarbe ist der Farbstift Nummer 3.

\*\*\*\*\*  
 \*                    GPEN                    \*  
 \*\*\*\*\*

Format:            | GPEN, wert

Funktion:         Einstellen der Graphik-Vordergrundfarbe

Bemerkung:       Mit dem Befehl |GPEN, wert weisen Sie der Graphik-Vordergrundfarbe einen neuen Farbstift zu. Eine ausführliche Erklärung finden Sie beim |MASK-Befehl.

Beispiel:         | GPEN, 2

                  der neue Farbstift für die Graphik-Vordergrundfarbe ist der Farbstift Nummer 2.

\*\*\*\*\*  
 \*                    MASK                    \*  
 \*\*\*\*\*

Format:            | MASK, wert

Funktion:         definieren einer Maske für die Graphikausgabe

Bemerkung:       Punkte und Linien werden am CPC 464 normalerweise nur in einer Farbe ausgegeben. Mit Hilfe des |MASK-Befehls ist es nun möglich, auch zweifarbig gemusterte Graphikausgaben leicht zu erreichen. Wird mit |MASK eine Maske übergeben, so ist diese Maske des weiteren das verwendete Muster. Das Muster entspricht dem binären Wert der übergebenen Zahl, Nullen in dieser Maske ergeben einen Punkt in der Graphik-Hintergrundfarbe, Einsen einen Punkt in der Graphik-Vordergrundfarbe. Benutzen Sie beispielsweise den Befehl |MASK,&55, so wird der Wert &55 übergeben, der binär dargestellt so aussieht: 01010101. Das heißt, daß z.B. das Zeichnen einer Linie per DRAW X,Y eine abwechselnd zwischen Graphik-Vordergrund- und Hintergrundfarbe wechselnde Linie ergibt.

\*\*\*\*\*  
 \*                    UNMASK                    \*  
 \*\*\*\*\*

Format:            | UNMASK

Funktion:         Aufheben der Graphikmaskierung

Bemerkung:       Nach Eingabe des |UNMASK-Befehls ist die Graphikausgabe des CPC 464 wieder in ihrer ursprünglichen Form, d.h. jegliche Maskendefinition mittels des |MASK-Befehls wird aufgehoben.

\*\*\*\*\*  
\*                  SLOW                  \*  
\*\*\*\*\*

Format:          | SLOW

Funktion:        Aufheben der FAST-Routine

Bemerkung:      Mittels des Befehls |SLOW wird die Textausgabe auf  
                  die normale Geschwindigkeit zurückgesetzt.

**ANHANG**  
\*\*\*\*\*

**A. COPY-PATCH:**  
-----

Das Dienstprogramm COPY.COM auf der Systemdiskette der vortex Floppy Disk Station F1 läuft selbst dann nicht unter einem 62K CP/M, wenn Sie beim Patchen des CP/M's den Systemvektor aktiviert haben. Keine Angst, Sie können dieses Programm trotzdem benutzen, müssen es allerdings vorher etwas ändern, d.h. patchen. Hierzu benötigen wir das Programm DDT.COM, welches sich auch auf der Systemdiskette befindet.

Gehen Sie die folgenden Punkte einfach in der angegebenen Reihenfolge durch und Sie erhalten in einfachster Weise eine lauffähige COPY-Version.

<CR>=Drücken der Enter-Taste

(1) Legen Sie eine CP/M Systemdiskette, die die Programme

DDT.COM und COPY.COM

enthält ins Laufwerk A ein und starten Sie das CP/M Betriebssystem.

(2) Aufruf des Debuggers und gleichzeitiges Laden von COPY.COM

DDT COPY.COM<CR>

Haben Sie alles richtig gemacht, so erscheint folgende Meldung auf dem Bildschirm:

DDT VERS 2.2  
NEXT PC  
0780 0100  
-

Der Bindestrich ist hierbei das Bereitschaftszeichen (auch Prompt genannt) des Debuggers.

(3) Nun geben Sie

S1CA<CR>

ein. Mit dem 'S' am Anfang Ihrer Eingabe starten Sie den sogenannten "Substitute Mode" (substitute=ersetzen) des Debuggers. Dieser Mode erlaubt es den Inhalt von bestimmten Speicherzellen gezielt zu verändern. Hierzu muß der Debugger natürlich auch wissen, welche Zelle Sie verändern möchten. Dies sagen Sie ihm mit dem '1CA' hinter dem S in obiger Eingabe.

Jetzt sehen Sie auf dem Bildschirm:

01CA 2A x

und der Cursor steht an der Stelle, wo wir zur Markierung das x hingeschrieben haben. Geben Sie jetzt 21<CR> ein und Sie sehen:

01CA 2A 21  
01CB 06 x

auch hier steht x wieder für den Cursor. Nach der Eingabe von <CR> sehen Sie:

01CA 2A 21  
01CB 06  
01CC 00 x

jetzt geben Sie noch 9F<CR> ein:

01CA 2A 21  
01CB 06  
01CC 00 9F  
01CD B7 x

und der Patch ist so gut wie vollbracht. Den S-Mode verlassen wir durch die Eingabe von .<CR> und der Debugger meldet sich wieder mit seinem Prompt -.

Jetzt kontrollieren wir noch, ob wir auch alles richtig gemacht haben:

Hierzu geben wir D1CA<CR> ein und der Debugger liefert uns einen sogenannten Speicher-Dump, beginnend mit der Stelle, wo wir etwas geändert haben.

Wenn alles richtig sein soll, muß der Dump folgendermaßen aussehen:

```
01CA 21 06 9F B7 ED 52 !... R
01D0 D1 AF ED 52 3C 30 FB 3D FE 00 CA 15 04 32 B3 02 ... R<0.=.....2..
01E0 3A B4 02 FE 01 28 21 3A B2 02 3C 47 21 00 00 ED :...(!:..<G!...
01F0 5B B0 02 19 10 FD 3A B3 02 5F 16 00 AF ED 52 3C Å.....:.....R<
0200 30 FB 3D D6 1E D4 1A 04 3A B4 02 FE 00 28 06 CD 0.=.....(..
0210 38 01 CD 42 01 CD 99 03 AF 32 AA 02 3E 30 32 F8 8..B.....2..>02.
0220 06 32 15 07 16 00 06 00 AF 32 AB 02 CD 7E 02 3A .2.....2...ß.:
0230 B3 02 D5 CD B9 02 ED 53 AC 02 D1 CD 94 02 C5 CD .....S.....
0240 23 03 C1 ED 5B AC 02 3A AB 02 FE 01 20 DE 3A B2 #...Å.....:..
0250 02 FE 00 28 17 3A AA 02 3C 32 AA 02 FE 02 28 0C ..(.....<2.....(
0260 C6 30 32 F8 06 32 15 07 16 01 18 BA 11 5A 06 CD .02...2.....Z..
-
```

In der ersten Zeile müssen Sie die Zahlenfolge 21 06 9F sehen, wenn Ihr Patch richtig sein soll. Ist dies nicht der Fall, so geben Sie CTRL C (gleichzeitiges Drücken der CTRL-Taste und C) ein und machen alles nochmals.

Stimmt alles, so geben Sie ebenfalls CTRL C ein, um den Debugger zu verlassen. Wieder im CP/M angekommen, geben Sie ein:

SAVE 7 COPY62.COM<CR>

und sichern sich damit ein neues COPY-Programm names COPY62 auf die Diskette, welches nun auch unter dem 62K CP/M lauffähig ist (natürlich auch unter dem "alten" 44K CP/M).

#### B. PARA der vortex Disk-Manager:

-----

Das Programm PARA läuft weiterhin nur unter dem "normalen" 38K CP/M, mit dem Sie es bislang schon betrieben haben. Sollten Sie PARA nach dem Kauf der RAM-Erweiterung erworben haben, so beachten Sie bitte folgendes:

Dieses 38K CP/M erhalten Sie mit Hilfe der Programme MOVCPM und SYSGEN aus Ihrem 44K CP/M, das Sie beim Kauf der vortex Floppy Disk Station miterworben haben. Sie dürfen hier auf keinen Fall das durch das Programm "PATCH.COM" gepatchte 44K CP/M verwenden.

#### C. GRAPHIC MASTER 2.0:

-----

Sollte das Programm "GRAPHIC MASTER 2.0" nach dem Einbau der RAM-Karte nicht mehr laufen, so beheben Sie diesen Mißstand wie folgt: resetieren Sie Ihren Rechner und geben Sie die Befehle POKE &AC01,&AF <CR> POKE &AC02,&32 <CR> POKE &AC03,&45 <CR> POKE &AC04,&AE <CR> ein. Danach laden Sie von Ihrer Original GRAPHIC MASTER Diskette mit LOAD "GRAMA" <CR> das Programm GRAMA.BAS. In Zeile 30 müssen Sie nun den Teil mit dem MEMORY-Befehl eliminieren und das Programm mit SAVE "GRAMA",P <CR> wieder speichern.

#### D. SPTEST.COM:

-----

Auf der im Lieferumfang der RAM-Karte enthaltenen Datencassette befindet sich außer dem Programm PATCH.COM auch das Programm SPTEST.COM. Es ist ein menügesteuertes Speichertestprogramm, mit dem Sie nach Einbau oder Aufrüstung der RAM-Karte diese auf volle Funktion überprüfen können. Das Programm können Sie auf die gleiche Art, wie beim Programm PATCH.COM in diesem Handbuch beschrieben, auf Diskette kopieren und unter dem Original 44K CP/M (nicht (!) das gepatchte CP/M) mit SPTEST <CR> aufrufen. Beachten Sie bitte, daß das Programm endlos läuft, wenn keine Fehler gefunden werden.

R E F E R E N Z - K A R T E  
für die zusätzlichen BASIC-Befehle des vortex BOS 1.0

DIE NACHFOLGENDEN BEFEHLE, DIE MIT EINEM '\*' GEKENNZEICHNET SIND, FUNKTIONIEREN NUR DANN, WENN ZUVOR ;BOS <CR> EINGEGEBEN WURDE.

```

l,m,n,o      Nummer einer Programmspeicher-Bank 0....8
addr        Speicheradresse
./          optionale Angaben
par1...     Übergabeparameter bei einem CALL
chn         Kanalnummer des Ausgabegeräts
line       Programmzeilennummer
var        Integervariable
var$       Stringvariable
wert       Integerzahl
len1...    Feldlänge
reclng     Rekordlänge
recnum     Rekordnummer
i          Bildspeichernummer 0....16
    
```

Befehl	Syntax	Bemerkung
:BANK	,n	*
:BASIC		*
:BOS		*
:CALL	,n,addr/,par1,par2.../	*
:COMMON	/,@var\$,l,m,n.../	*
:DEV	/,chn/	chn=0,8,9 *
:FAST		
:FRAME		
:GCHAR	,x,y,@var	
:GOSUB	/,n/,line	*
:GOTO	/,n/,line	*
:GPAPER	,wert	
:GPEN	,wert	
:ID		*
:LIST	/,l,m,n,o/	*
:LOAD	,@var\$	*
:MASK	,wert	num=0..255
:MON		
:NEW	/,l,m,n,o/	*
:PEEK	,n,addr,@var	*
:POKE	,n,addr,wert	*
:RAMCLOSE		*
:RAMFIELD	,len1,len2,..	*
:RAMOPEN	,reclng	*
:RAMREAD	,recnum,@var1\$,@var2\$,..	*
:RAMWRITE	,recnum,@var1\$,@var2\$,..	*
:RECORDS		berechnet REC *
:RETURN		*
:RUN	,@var\$,/n,line/	*
:SAVE	,@var\$,/l,m,n,o/	*
:SCREEN.IN	,i	*
:SCREEN.OUT	,i	*
:SCREENS		berechnet SCR *
:SLOW		
:SPOOL.ON		*
:SPOOL.OFF		*
:UNMASK		
:VIDEO.ON		*
:VIDEO.OFF		*



**vortex**  
Computersysteme Vertriebs GmbH  
Klingenberg 13  
7106 Neuenstadt 5