1 MByte Ramkarte

- für Apple II, II+, IIe, Basis 108 und andere kompatible Computer

- 64 kByte bis 1 MByte Speicherausbau

- Ramdisktreiber-Programme für die Betriebssysteme:

  > Apple DOS 3.3 (Master und Slave),

  > DiversiDos 2-C/4-C, mit- und ohne 'DDMOVER'

  > Apple ProDos 1.0.1, 1.1.1

  > Apple Pascal 1.1,
  Apple Pascal 1.2 / 64k, 1.2 / 128k, (auch 'runtime'-Versionen),
  Apple Pascal 1.3 / 64k, 1.3 / 128k

  > CP/M 2.2 für Microsoft Softcard, CP/M 2.20, CP/M 2.23,
  Microsoft premium card, CP/M 2.26,
  Microsoft Softcard II, CP/M 2.28

- umfangreiche und benutzerfreundliche Softwareunterstützung

  > Die Ramkarte ist unmittelbar nach Ablauf eines Initialisierungs-programmes betriebsbereit.
  Peripherieslot und Speicherkapazität werden automatisch erkannt, sodaß während der Initialisierung keinerlei Benutzereingaben erforderlich sind.

  > Der Dateninhalt der Ramkarte kann nur durch Stromausfall oder unsachgemäße Behandlung verlorengehen.
  Auch bei erneuter Initialisierung nach einem 'Systemabsturz' oder Betriebssystemwechsel bleiben alle Dateien erhalten.
  Dies gilt ebenso bei irrtümlich mehrfachem Aufruf des jeweiligen Initialisierungsprogrammes.
  Datenverlust durch Fehlbedienung ist damit so gut wie ausge-schlossen.

  > Je nach Betriebssystem und Initialisierungs-Software kann man von der Ramkarte aus booten (Warmstart bei CP/M 2.2).

  > Die Ramkarte ist mit dem erweiterten Floppy-Disk-System (AFDC2/AFDC3-Controller) der Firma erphi GmbH vollständig kompatibel.

Inhaltsverzeichnis Seite

Vorwort

Mit der vorliegenden Speichererweiterungskarte für Apple-II-Rechner und kompatible Geräte werden viele Arbeiten mit Ihrem Computersystem deutlich schneller und leiser ablaufen.
Ihre Floppy-Disk-Laufwerke und Disketten werden in der Regel weit weniger stark beansprucht und unterliegen somit einem entsprechend geringerem mechanischen Verschleiß.
Die Leistungsfähigkeit zahlreicher Programme tritt oftmals erst bei Verwendung einer Ramkarte voll in Erscheinung.

Voraussetzung für einen einwandfreien Betrieb der Ramkarte ist jedoch der korrekte Einbau der Karte in den Rechner und die Benutzung geeigneter Initialisierungsprogramme.

Aus diesem Grund empfehlen wir Ihnen dringend, vor Inbetriebnahme der Karte zumindest die Kapitel 1 und 2 dieser Beschreibung zu lesen.
Je nach Betriebssystem gilt dies auch für die wichtigsten Unterabschnitte der Kaptitel 3 bis 6.
Es ist dagegen nicht unbedingt erforderlich, die mit einem Stern (*) gekennzeichneten Abschnitte durchzuarbeiten.
Diese Textstellen enthalten Detailinformationen, die gewöhnlich nur fi Programmierer von Bedeutung sind.

1. Einführung

Die vorliegende Ramkarte ist speziell für den Einsatz als Ramdisk (= Pseudo-Disk) konzipiert.

Das jeweilige Betriebssystem verwaltet den zusätzlichen Speicherraum der Ramkarte in diesem Fall in gleicher Weise, wie bei einem Floppy-Disk Laufwerk.
Aus der Sicht eines Anwender-Programmes bedeutet das, daß der Zugriff auf die in der Ramkarte gespeicherten Daten über eine bereits bekannte Schnittstelle abläuft.
Infolgedessen kann man bei diesem Verfahren fast immer davon ausgehen, daß alle Programme, die Dateien von Floppy-Disk-Laufwerken bearbeiten, automatisch und ohne Änderung auch sämtliche Vorteile einer Ramdisk nutzen können:

- völlig lautloser und verschleißfreier Betrieb

- wesentlich kürzere Daten-Zugriffszeiten im Vergleich zu Floppy-Disk-Laufwerken und anderen Massenspeichern.
  Zahlreiche Programme können dadurch erheblich schneller abgearbeitet werden.

- Wenn Programme mit sehr intensivem Datenaustausch, wie z.B. Such- und Sortierprogramme (Datenbanken) oder auch Assembler, Compiler und Linker bei der Software-Entwicklung mit einer Ramkarte zusammen arbeiten, dann werden die mechanischen Floppy-Disk-Laufwerke und auch die Disketten selbst weitaus weniger stark beansprucht, sodaß deren Lebenserwartung deutlich ansteigt.

Man muß sich aber stets darüber im Klaren sein, daß diesen sehr günstigen Merkmalen leider auch ein schwerwiegender Nachteil gegenübersteht:

- Der gesamte Inhalt der Ramkarte geht beim Abschalten des Rechners verloren; ebenso natürlich auch bei einem Stromausfall.
  Wichtige Dateien, wie beispielsweise die aktuelle Arbeitsdatei bei der Textverarbeitung, müssen daher unbedingt von Zeit zu Zeit von der Ramkarte auf Floppy-Disk, oder ein anderes, nichtflüchtiges Speichermedium übertragen werden.

Ramkarten sollte man aus diesem Grund weniger als Ersatz, sondern vielmehr als eine sehr wertvolle Ergänzung der herkömmlichen Massenspeicher-Systeme betrachten.

2. Inbetriebnahme der Ramkarte

Die Inbetriebnahme der Ramkarte vollzieht sich in zwei Schritten:

1. Einsetzen der Karte in einen geeigneten Peripherieslot des Rechners (zuvor bitte ausschalten !!!)

2. Aufruf eines Initialisierungsprogrammes zum Einbinden der Ramdisk in das jeweilige Betriebssystem

Während Schritt 1 normalerweise nur einmal durchgeführt werden braucht ist der Aufruf des Initialisierungsprogramms bei jedem Kaltstart des Systems erforderlich.

2.1 Einsetzen der Ramkarte in den Computer

Die Ramkarte muß in einen der 7 oder 8 Erweiterungs-Steckplätze des Rechners eingesetzt werden.

Hinsichtlich der Hardware ist hierfür im Prinzip jeder der Slots 1 bis 7 geeignet.
Einschränkungen ergeben sich aber durch die Software der Betriebssysteme und der Initialisierungsprogramme und selbstverständlich auch dadurch, daß ein Teil der Erweiterungs-Slots für andere Peripheriegeräte benötigt wird.

Eine besonders günstige Konfiguration liegt vor, wenn man die Ramkarte in Slot 5 oder Slot 4 einsetzen kann.
(Beachten Sie bitte, daß die Floppy-Disk-Controller nach Möglichkeit in aufeinanderfolgenden Steckplätzen untergebracht sind, z.B Slot 6 bei einem- Slots 6 und 5 bei zwei- und Slots 6, 5 und 4 bei drei Floppy-Disk-Controllern.)

Sind die Erweiterungs-Slots 4 bis 6 bereits von Floppy-Disk-Controllern belegt, dann sollte man die Ramkarte in Slot 7 einsetzen.

Ist auch Slot 7 schon besetzt, dann kommt nur noch Slot 2 in Frage, sofern nicht mit dem Betriebssystem Apple Pascal gearbeitet werden soll. (Lediglich bei Apple ProDos kann man notfalls auf Slot 1 ausweichen. Dieser Steckplatz ist jedoch normalerweise für ein Drucker-Interface reserviert.)

Prüfen Sie bitte vor dem Einsetzen der Ramkarte, ob die Stromversorgung Ihres Rechners der zusätzlichen Belastung gewachsen ist.
Die Stromaufnahme der Ramkarte ist abhängig vom Speicherausbau und von den bei der Bestückung der Karte verwendeten Speicher-ICs, sodaß hier keine exakten Angaben möglich sind.
Bei vollständigem Speicherausbau sollte man bei der 5-V-Versorgung von einer zusätzlichen Stromaufnahme von etwa 0,5 A ausgehen.
Eine ungehinderte Abfuhr der Verlustwärme muß sichergestellt sein.

Beim Einsetzen der Ramkarte in den Rechner geht man am besten wie folgt vor:

> Rechner selbst, sowie alle anderen mit dem Computer verbundenen Geräte ausschalten !!

> Abdeckhaube des Rechners entfernen

> Karte bei vorsichtigem Hin- und Herbewegen in den ausgewählten Erweiterungs-Slot einstecken.
Jede übertriebene Gewaltanwendung ist zu vermeiden !

> Abdeckhaube des Rechners wieder aufsetzen.

Beachten Sie bitte darüberhinaus unbedingt noch die folgenden Hinweise

> Die Kontaktfinger der Ramkarte müssen frei sein von Verunreinigung jeder Art.

> Bedenken Sie, daß die auf der Ramkarte befindlichen integrierten Schaltkreise durch statischer Aufladung zerstört werden können

>>> Peripheriekarten dürfen in keinem Fall bei eingeschaltenem <<<
>>> Gerät ein- oder ausgesteckt werden. <<<
>>> Nichtbeachtung dieser Regel hat fast immer sehr schwere <<<
>>> Schäden zur Folge ! <<<
>>> Auch ein späterer, scheinbar nicht zu erklärender Ausfall <<<
>>> einzelner Systemkomponenten ist möglich. <<<

2.2 Die Initialisierungsprogramme

Bevor die in einem geeigneten Peripherieslot installierte Ramkarte als Ram-Disk einsatzbereit ist, müssen noch einige Modifikationen in der jeweiligen Betriebssystem-Software durchgeführt werden.

Dies geschieht mit Hilfe eines passenden Initialisierungsprogramms, welches am besten unmittelbar nach dem Booten gestartet wird.

Alle notwendigen Betriebssystem-Eingriffe betreffen ausschließlich die im Arbeitsspeicher des Rechners befindliche Kopie der jeweiligen Systemsoftware.
Die Daten auf den 'Systemspuren' der Bootdiskette (Apple DOS 3.3, CP/M bleiben in jedem Fall unverändert.

Neben der Betriebssystem-Anpassung besteht eine wesentliche Aufgabe des Initialisierungsprogrammes darin, auf der Ramkarte ein Inhalts-verzeichnis anzulegen, um die Karte so für zukünftige Lese- und Schreibvorgänge vorzubereiten.
Dabei ist sichergestellt, daß eventuell bereits in der Ramkarte abgelegte Dateien nicht zerstört werden.

Sollte die Ramkarte zum Zeitpunkt des Initialisierungsprogramm-Aufrufs Aufzeichnungen enthalten, die unter einem anderen Betriebssystem angelegt wonden sind, dann wird der Zugriff auf die Ramkarte gesperrt, um einen Verlust dieser Daten zu vermeiden.

Initialisierungsprogramme für die Betriebssysteme Apple DOS 3.3, Apple ProDos, Apple Pascal und CP/M 2.2 finden Sie auf der beiliegenden Multidiskette.

Bei der Multidiskette handelt es sich um eine Diskette mit beidseitige Datenaufzeichnung (2*35 Spuren), die in vier voneinander unabhängige Bereiche unterteilt ist.
Jeder Teilbereich enthält ein Inhaltsverzeichnis für eines der vier genannten Betriebssysteme, sowie ein oder mehere Ramkarten-Initialisierungsprogramme für das entsprechende Betriebssystem.

Während die Vorderseite sämtliche Dateien für Apple DOS 3.3, CP/M 2.2 und Apple Pascal enthält, findet man auf der Rückseite der Multidisket die entsprechenden Programme für Apple ProDos.

Es kommt nun darauf an, für jedes der in Frage kommenden Betriebssysteme das für Ihre Erfordernisse jeweils am besten geeignete Initialisierungsprogramm auszusuchen und auf Ihre Boot- und Arbeitsdisketten zu übertragen.

Um diese Auswahl zu erleichtern, werden in den folgenden Kapiteln die wichtigsten Merkmale der zur Verfügung stehenden Initialisierungsprogramme dargelegt.
Darüberhinaus finden Sie dort noch weitere wichtige Hinweise und Tips für den Betrieb der Ramkarte.

Die mit einem Stern (*) gekennzeichneten Unterabschnitte enthalten weiterführende Informationen über die Organisation der Daten in der Ramkarte, sowie über eventuelle Kompatibilitätsprobleme in Zusammenhang mit Betriebssystem-Modifikationen für andere Peripheriegeräte. Diese Teile der Beschreibung richten sich, - ebenso wie die Anhänge H und S - an fortgeschrittene Programmierer, die die Funktionsweise der Karte genauer kennenlernen möchten, oder selbst beabsichtigen, Initialisierungsprogramme bzw. Treibersoftware für andere Betriebssysteme oder für Sonderanwendungen zu erstellen.

3. Initialisierungsprogramme für Apple DOS 3.3

3.1 Voraussetzungen für den Betrieb der Ramkarte unter Apple DOS 3.3

Bei Verwendung von einem der im folgenden beschriebenen Initialisierungsprogramme können Sie Ihre Ramkarte unter Apple DOS 3.3 (Master- und Slave-Version) oder unter DiversiDos 2c/4c betreiben. Bei DiversiDos ist der Einsatz des Relocator-Programmes 'DDMOVER' zulässig, wobei es keine Rolle spielt, ob das Initialisierungsprogramm vor oder nach dem Ablauf des Relocators gestartet wird.

Als einzige zwingende Voraussetzung ist zu beachten, daß Ihr Rechner mit mindestens 48kByte Arbeitsspeicher ausgerüstet sein muß und daß sich die Ramkarte in einem der Peripherieslots 2, 4, 5, 6, oder 7 befindet (vgl. 2.1).

Der DOS 3.3-'INIT'-Befehl muß aus einer Reihe von Gründen bei Verwendu der Ramkarte leider außer Betrieb gesetzt werden.
Nach dem Ablauf der Initialisierungsprogramme können daher mit 'INIT' keine Disketten mehr formatiert werden.
(Versucht man dies trotzdem, dann erscheint die Fehlermeldung 'write protected'.)

3.2 Das Initialisierungsprogramm ERAM Version 2.3

3.2.1 Anwendungsbereich, Bedienung und Wirkungsweise

Bei Ablauf dieses Initialisierungsprogramms ('BRUN ERAM2') entstehen, je nach Speicherkapazität der Ramkarte ein einzelnes, oder auch mehre 'Pseudo-Laufwerke'.
Diese werden - wie unter DOS 3.3 üblich - mit Hilfe von Slot- und Drive-Nummer selektiert.

Eine solche Unterteilung in mehrere getrennte Ramdisks ist bei Karten mit hoher Speicherkapazität erforderlich, weil das DOS 3.3 - Betriebssystem nur (Disketten-) Speicher bis zu 400 kByte verwalten kann.

Im Falle einer 1-MByte-Ramkarte, die in Slot 5 installiert ist, erhält man 3 separate (logische) Drives, die mit 'S5, D1', 'S5, D2' und 'S5, D3' anzusprechen sind.
Drive 1 und Drive 2 stellen dabei Ramdisks maximaler Größe dar, währen die restliche Kapazität der Ramkarte für den entsprechend 'kleineren' Drive 3 genutzt wird.

Aus der folgenden Tabelle geht hervor, welche Kombinationen von Pseudo Laufwerken in Abhängigkeit von der Speicherkapazität der Karte vorgesehen sind.

Gesamt-kapazität der Ramkarte	Kapazität der Pseudolaufwerke					
	Drive 1		Drive 2		Drive 3	
kByte	kByte/Sektoren		kByte/Sektoren		kByte/Sektoren	
64	64	256	----------		----------	
128	128	512	----------		----------	
192	192	768	----------		----------	
256	256	1024	----------		----------	
512	400	1600	136	544	----------	
768	400	1600	392	1568	----------	
1024	400	1600	400	1600	272	1088

Bei den Angaben in dieser Tabelle ist zu beachten, daß jeweils die Gesamtkapazität bzw. die Gesamtzahl der Sektoren eines Pseudo-Laufwerk aufgeführt ist.
Zur letztlich nutzbaren Speicherkapazität gelangt man, indem man davon den Speicherplatzbedarf von Inhaltsverzeichnis und 'Systemspuren' abzieht (je nach Speicherkapazität 64 oder 128 Sektoren).

Die Tatsache, daß bei den Formaten mit mehr als einer Pseudo-Disk die Gesamtkapazität der logischen Drives größer ist, als die Gesamt-Speicherkapazität der Ramkarte selbst, mag zunächst widersinnig erscheinen.
Die Erklärung hierfür liegt einfach darin, daß das bei 'normalen' Floppy-Disk-Speichern auf jeder Diskette aufgezeichnete Betriebssystem bei einer Ramkarte mit mehreren logischen Drives natürlich nur einmal absgespeichert werden braucht.
Bei der 1-MByte-Ramkarte bringt dieser Trick einen scheinbaren Kapazitätszuwachs von 48 KByte (2 * 3 Spuren mit je 32 Sektoren).

Das Betriebssystem wird bei Aufruf von ERAM Version 2.3 von den Systemspuren der Diskette, von der auch das Initialisierungsprogramm gestartet wurde, eingelesen und anschließend in die Ramkarte übertragen.

Es ist aus diesem Grund möglich, das Betriebssystem von der Ramkarte aus zu booten, sofern es sich dabei um ein DOS 3.3 Slave- oder ein 48k DiversiDos 2c/4c System handelt.
Diese Zusatzfunktion ist recht nützlich, wenn z.B. bei der Softwareentwicklung nach einem Systemabsturz neu gebootet werden muß, oder wenn zwischenzeitlich mit einem anderen Betriebssystem gearbeitet wurd

Wenn in Ihrem Computersystem ein erphi AFDC2- oder ein AFDC3- Floppy-Disk-Controller installiert ist, dann ist es entscheidend, auf welche Weise Sie den Bootprozeß von der Ramkarte aus starten.

Um einen 'AUTOPATCH-Boot' (vgl. AFDC2/3-Handbuch der Fa. erphi GmbH) zu erreichen, ist die Eingabe von 'PR #s' (Basic) bzw. 's contr. P' (Monitor) erforderlich, wobei der Kleinbuchstabe 's' hier stellvertretend für die Slotnummer der Ramkarte steht.

Ein 'Originalsystem-Boot' wird sinngemäß mit den Befehlen 'IN #s' (Basic) bzw. 's contr. K' (Monitor) ausgelöst.
Bei AFDC2-Controllern mit einem älteren Firmware-ROM als Version 6.8 ist ein 'AUTOPATCH-Boot' von der Ramkarte aus nicht möglich.

Speziell für Besitzer eines Apple IIe Computers sei noch erwähnt, daß mit der üblichen Tastenkombination 'control - offener Apfel - Reset' nicht von der Ramkarte gebootet werden kann.
(Dies gilt auch dann, wenn sich die Ramkarte in Slot 7 befindet.)

3.2.2 * ERAM Version 2.3, RWTS-Schnittstelle

Da die Ramkarten-Treiber-Software auf RWTS-Ebene im Betriebssystem eingebunden ist, läßt sich die Ramkarte im Prinzip auch über die RWTS-Schnittstelle ansprechen.

Allerdings ist dabei zu beachten, daß sich in Abhängigkeit von der jeweiligen Ramkartenkapazität recht unterschiedliche Wertebereiche für die (logischen) Spur- und Sektornummern ergeben.
Bei den Ramkartenformaten 512 kByte bis 1024 kByte muß zudem noch der Drivenummern-Parameter berücksichtigt werden.
(Die Volume-Nummer wird dagegen stets ignoriert.)

Die folgende Tabelle zeigt den Zusammenhang der obengenannten RWTS-Parameter und dem Speicherausbau der Ramkarte:

Gesamtkapazität der Ramkarte in KByte	Drive 1		Drive 2		Drive 3	
	Spuren	Sektoren	Spuren	Sektoren	Spuren	Sektoren
64	0..15	0..15	----------		----------	
128	0..31	0..15	----------		----------	
192	0..47	0..15	----------		----------	
256	0..31	0..31	----------		----------	
512	0..49	0..31	0..16	0..31	----------	
768	0..49	0..31	0..48	0..31	----------	
1024	0..49	0..31	0..49	0..31	0..33	0..31

Die Spuren 0 bis 2 sind bei allen Formaten zum Abspeichern des Betriebssystems vorgesehen und werden von allen logischen Drives gemeinsam 'benutzt'.
Wegen dieser nicht eindeutigen Zuordnung, und um die 'Kaltstart'-Fähigkeit der Karte nicht zu gefährden, sind diese drei Spuren mit einem Schreibschutz versehen. (Der Schreibschutz läßt sich aufheben, indem man in den Speicherplatz mit der Adresse $CsFF ein Nullbyte einschreibt.)

Eine weitere Besonderheit ist bei logischen Drives zu beachten, deren Wertebereich für die Spurnummer die DOS 3.3 Katalog-Spurnummer 17 nicht mehr erfasst.
Die DOS 3.3-Filemanager-Software erwartet grundsätzlich bei jedem Speicherformat auf Spur 17 den sogenannten VTOC-Sektor, der den Ausgangspunkt für die gesamte Directory-Struktur bildet. (Nähere Informationen hierzu findet man in der einschlägigen DOS 3.3 Dokumentation.
Wenn man Laufwerke mit weniger als 18 Spuren realisieren möchte, dann ist es folglich zwingend notwendig, eine logische Spur mit der Nummer zuzulassen, die dann in eine geeignete physikalische Spur umzurechnen ist.

Im vorliegenden Fall ist die Angabe der Spurnummer 17 in der RWTS-Parameterliste bei den kleinformatigen Pseudo-Drives gleichbedeutend m der Vorgabe von Spur 3, sodaß hier die Catalog-Spur unmittelbar im Anschluß an die Systemspuren angeordnet ist.
(Alle Spuren von 0 bis einschließlich 3 sind daher bei diesen 'logisch Minidrives' als belegt gekennzeichnet.)

Die vorangegangenen Ausführungen haben sicherlich deutlich gezeigt, da in Verbindung mit dem Initialisierungsprogramm ERAM Version 2.3 beim Direktzugriff auf die Ramkarte über die RWTS-Schnittstelle mit großer Sorgfalt gearbeitet werden muß.
Von Anwenderprogrammen, die auf diese Weise mit der Ramkarte kommunizieren, wird zudem ein hohes Maß an Anpassungsfähigkeit gefordert, wen ein einwandfreier Betrieb mit Ramkarten unterschiedlicher Speicherkapazität möglich sein soll.

Etwas einfacher liegen die Verhältnisse bei Verwendung des Initialisierungsprogramms ERAM Version 1.3.
In Anwendungsfällen, wo auf einen RWTS-Direktzugriff nicht verzichtet werden kann, sollte man daher nach Möglichkeit diesem Initialisierungsprogramm den Vorzug geben.

3.2.3 * ERAM Version 2.3, Speicherorganisation

Die Zuordnung der auf RWTS-Ebene erscheinenden Wertebereiche von Spur- Sektor- und Drive-Nummer zu den Bestandteilen I/O-Startadresse, Spaltenadresse und Banknummer der RAM-Gesamtadresse (vgl. Anhang H und S) vollzieht sich in zwei Stufen.

In der ersten Stufe erfolgt eine Umsetzung der von der Speicherkapazität abhängigen Komponenten der Gesamtadresse in eine fortlaufende Spur- und Sektor-Nummer, wie in der folgenden Tabelle angedeutet:

Kapazität (kByte)	Sektoren	Spuren	Bänke
64	0..15	0..15	0
128	0..15	0..31	0..1
192	0..15	0..47	0..2
256	0..31	0..31	0..3
256	0..31	0..31	0
512	0..31	0..63	0..1
768	0..31	0..95	0..2
1024	0..31	0..127	0..3

Bestückung mit 64k DRAMs:

Spaltenadresse:
Spur(0..3) + 16 * Sektor(0..

Banknummer:
Spur(4..5) + 2 * Sektor(5)

Bestückung mit 256k DRAMs:

I/O-Startadresse:
256 * Sektor(0..1) + $C800

Spaltenadresse:
Spur(0..4) + 8 * Sektor(2..

Banknummer: Spur(5..6)

(die Zahlen in Klammern kennzeichnen Bitpositionen bei binärer Darstellung der Werte)

Man erkennt, daß damit für die Kapazitäts-Ausbaustufen bis einschließlich 256 kByte bereits alle erforderlichen Umrechnungen vorliegen. Bei höheren Kapazitäten ist dagegen noch eine Aufspaltung in mehrere logische Drives vorzunehmen, um so den für DOS 3.3 vorgeschriebenen Spurnummern-Wertebereich 0..49 einhalten zu können.

Zu diesem Zweck wird bei Drive-Nummern größer als 1 ein Spur-Offset zu eingangsseitig ankommenden RWTS-Spurnummer addiert.
Die Mehrfachzuordnung der Systemspuren 0..2 (vgl. 3.2.2) entsteht dadurch, daß die Addition nur bei Spurnummern größer als 2 stattfindet

Dieser Sachverhalt wird nachfolgend durch eine weitere Tabelle verdeutlicht:

Kapazität (kByte)	Spur	Drive 1	Drive 2	Drive 3
512	RWTS	0..2, 3..49	0..2, 3..16	----------
	intern	0..2, 3..49	0..2, 50..63	----------
768	RWTS	0..2, 3..49	0..2, 3..48	----------
	intern	0..2, 3..49	0..2, 50..95	----------
1024	RWTS	0..2, 3..49	0..2, 3..49	0..2, 3.. 33
	intern	0..2, 3..49	0..2, 50..96	0..2, 97..127

3.3 Das Initialisierungsprogramm ERAM Version 1.3

3.3.1 Anwendungsbereich, Bedienung und Wirkungsweise

Das Initialisierungsprogramm ERAM Version 1.3 ('BRUN ERAM1') teilt den Speicher der Ramkarte in Abhängigkeit von der Speicherkapazität in ein einzelnes- oder in mehrere gleichgroße Pseudo-Drives auf, wie in der folgenden Tabelle dargestellt.

Gesamt-Kapazität der Ramkarte	Kapazitat/Drive	Sektoren/Drive	Anzahl der Drives
64 kByte	64 kByte	256	1
128 kByte	128 kByte	512	1
192 kByte	192 kByte	768	1
256 kByte	256 kByte	1024	1
512 kByte	256 kByte	1024	2
768 kByte	256 kByte	1024	3
1024 kByte	256 kByte	1024	4

Die nutzbare Kapazität der logischen Drives ergibt sich aus den Tabellenangaben nach Abzug des für das Inhaltsverzeichnis benötigten Speicherplatzes (16 bzw. 32 Sektoren).

Bei den Ramkarten höherer Kapazität ist eine Aufspaltung in mehrere logische Drives unumgänglich, weil das DOS 3.3 Betriebssystem nur maximal 400 kByte große (Disketten-) Speicher verwalten kann.

Die jeweiligen Pseudo-Laufwerke werden, entsprechend der gängigen DOS 3.3 Konventionen, durch Angabe von Slot- und Drive-Nummer selektiert.

Ram-Disks, die mit dem Initialisierungsprogramm ERAM Version 1.3 generiert werden, enthalten keine Kopie des Betriebssystems. Ein 'Kaltstart' von der Ramkarte ist daher in diesem Fall nicht möglich.

Da die DOS 3.3 Software den normalerweise zur Unterbringung des Betriebssystems vorgesehenen Speicherraum nur noch teilweise zum Abspeichern von Benutzer-Daten nutzen kann, wird in den Inhaltsverzeichnissen der Pseudo-Disks ein entsprechend großer Speicherbereich als belegt gekennzeichnet.
Auf diese Weise erreicht man - gewissermaßen also durch 'Vortäuschen von Systemspuren, die es in Wahrheit gar nicht gibt - eine vollständi Ausnutzung des Ramkarten-Speichers.

Als Nebeneffekt dieser kleinen Manipulation ist zu beobachten, daß Programme, die die Gesamtspeicherkapazität einer Diskette anzeigen, im Falle eines mit ERAM Version 1.3 initialisierten Pseudo-Laufwerkes 48 bzw. 96 Sektoren mehr ausweisen, als tatsächlich an 'physikalischem Speicher vorhanden sind.

3.3.2 * ERAM Version 1.3, RWTS-Schnittstelle

Pseudolaufwerke, die mit dem Initialisierungsprogramm ERAM Version 1.3 installiert worden sind, lassen sich problemlos über die RWTS-Schnittstelle ansprechen.

Sieht man einmal davon ab, daß ab 512 kByte der Ramkartenspeicher in mehrere logische Drives unterteilt ist, dann ändert sich lediglich der Wertebereich für die Sektornummer in Abhängigkeit vom Speicherausbau d jeweiligen Karte.
Dies gilt jedoch nur für die in der Praxis relativ selten anzutreffenden Speicherkapazitäten bis 192 kByte.
Ab 256 kByte einschließlich liegen die Sektornummern konstant im Bereich 0..31.

Der zulässige Wertebereich für die Spurnummern beträgt bei allen Kapazitätsstufen 3..35.
Zu beachten ist, daß ein Zugriff auf die (System-) Spuren 0..2 nicht erlaubt ist und zur Ausgabe eines Fehlercodes führt ($40, Drive error)

In denjenigen Anwendungsfällen, wo ein linearer Spurbereich erwünscht ist, läßt sich dies durch Addieren eines Offsets von 3 leicht korrigieren.
Eine Auswertung der RWTS-Volume-Nummer findet nicht statt.

Kapazität der Ramkarte (kByte)	Spurnummer	Sektornummer	Drivenummer
64	3..34	0.. 7	1
128	3..34	0..15	1
192	3..34	0..23	1
256	3..34	0..31	1
512	3..34	0..31	2
768	3..34	0..31	3
1024	3..34	0..31	4

3.3.3 * ERAM Version 1.3, Speicherorganisation

Bei den mit ERAM Version 1.3 initialisierten Pseudo-Disks besteht eine relativ einfache Zuordnung der eingangs- (RWTS-) seitigen Spur-, Sekto und Drive-Parameter zur Ramkarten-Gesamtadresse (vgl. Anhang H und S).

In Ramkarten, die mit 64k-DRAMs bestückt sind, existiert nur ein einzelner logischer Drive, sodaß nur die Parameter Spur und Sektor zu verarbeiten sind:

Spaltenadresse: 8 * (Spur-3)(0..4) + Sektor(0..2)

Banknummer: Sektor(3..4)

Bei den Versionen mit 256k-DRAM-Bestückung ergibt sich die Banknummer unmittelbar aus der (um 1 reduzierten) Drivenummer und für die übrigen Komponenten gilt folgender Zusammenhang:

I/O-Startadresse: 256 * Sektor(0..1) + $C800

Spaltenadresse: 8 * (Spur-3)(0..4) + Sektor(2..4)

(Die eingeklammerten Bereichsangaben kennzeichnen Bitpositionen bei binärer Darstellung der entsprechenden Größen)

3.4 Das Kopieren von DOS 3.3 Dateien

Zum Kopieren einzelner Dateien verwendet man am besten die üblichen DOS 3.3-Speicherbefehle 'LOAD'/'SAVE' bzw. 'BLOAD' und 'BSAVE'. Für umfangreichere Kopierarbeiten wird man allerdings den Einsatz eines Kopier-Utility-Programms vorziehen.

Hierbei ist jedoch zu bedenken, daß sich nicht alle Kopierprogramme an die verschiedenen Speicherformate der Ramkarte anpassen können.

Generell ungeeignet sind aus diesem Grund sogenannte 'Spurkopierer' (Schnellkopier-Programme), wie z.B. 'COPYA', weil diese Programme praktisch immer speziell für das übliche 35-Spur-Disketten-Format ausgelegt sind.

Daten von der Ramkarte sollten daher stets File für File transferiert werden.

Dazu eignet sich beispielsweise das Kopierprogramm 'FID' der Firma Apple Computer.
Auch wenn vollständige Disketten umkopiert werden sollen, hält sich der zusätzliche Zeitaufwand gegenüber den Spurkopierern in akzeptablen Grenzen, weil ja zumindest auf der Seite der Ramkarte keinerlei Such- und Positionierzeiten anfallen. Die ständigen Directory-Zugriffe für die Dateiverwaltung verursachen daher, im Vergleich zu den mechanischen Laufwerken, keinen nennenswerten Zeitverlust.
Hinzu kommt, daß der Datenaustauch mit der Ramkarte normalerweise in einer Zeitspanne abläuft, die unter der Motor-Abschaltverzögerungszeit des beteiligten Floppy-Disk-Laufwerkes liegt.
Dadurch spielt bei umfangreichen Kopiervorgängen auch die Anlaufzeit der Laufwerksmechanik keine wesentliche Rolle mehr.

Beim Kopierprogramm 'FID' ist jedoch unbedingt zu beachten, daß auch dieses Programm bei Ramkarten-Kapazitäten ab 256 KByte ohne einige (geringfügige) Modifikationen nicht mehr ganz korrekt arbeitet.
(Bei den 32-Sektor-Formaten kann man den vorhandenen Speicherplatz nicht vollständig ausnutzen und Drive-Nummern größer als 2 sind generell nicht erlaubt.)

Mit dem kleinen Hilfsprogramm 'EFID', welches neben 'ERAM1' und 'ERAM2' in der DOS 3.3-Abteilung der Multidiskette zu finden ist, lassen sich diese Schwierigkeiten auf einfache Art beseitigen.

Übertragen Sie bitte zu diesem Zweck das Hilfsprogramm 'EFID' zusammen mit dem Apple-Utility 'FID' auf Ihre Boot- bzw. Arbeitsdisketten.
Beim Aufrufen von 'EFID' ('BRUN EFID') wird dann das eigenliche Kopierprogramm 'FID' in den Arbeitsspeicher geladen, alle erforderlichen Modifikationen eingefügt und anschließend gestartet.
Danach kann 'FID' ohne jede Einschränkung mit allen in Frage kommenden Pseudo-Disk-Formaten in gewohnter Weise benutzt werden.

3.5 Automatische Ramkarten-Initialisierung

In den vorangegangenen Abschnitten wude vorausgesetzt, daß das jeweilige Initialisierungsprogramm unmittelbar nach dem Booten mit einem 'BRUN'-Befehl gestartet wird.

Wenn man von der Ramkarte selbst booten kann, dann ist dieser Schritt natürlich überflüssig; man kann aber auch bei einem 'echten Kaltstart' eine automatische Initialisierung der Ramkarte erreichen.

Eine Möglichkeit wäre, das Initialisierungsprogramm als DOS 3.3-Start-file ('HELLO-File') zu spezifizieren.
Diese Methode ist jedoch ungeeignet, wenn nach dem Booten mehrere Programme nacheinander ablaufen sollen.

In einem solchen Fall ist es günstiger, das bisherige 'HELLO'-Programm selbst um einen Befehl wie zum Beispiel 'PRINT CHR$(4);"BRUN ERAM2"' zu erweitern.
Hierbei muß man aber unbedingt darauf achten, daß es zu keinen Überschneidungen des (BASIC-) 'HELLO'-Programmes mit dem Initialisierungsprogramm im Arbeitsspeicher des Rechners kommt.
(Die Initialisierungsprogramme liegen ab Adresse $1000 im Speicher.)

Besonders interessante Möglichkeiten ergeben sich, wenn man das jeweilige Initialisierungsprogramm von einem 'EXEC'-File aus aufruft.
Man kann auf diese Weise die Ramkarten-Initialisierung mit einem Kopierprogramm (z.B 'FID' bzw. 'EFID') kombinieren.
Beim EXEC-Aufruf wird dann die Ramkarte automatisch initialisiert und mit allen gewünschten Arbeitsdateien geladen.

Weiterführende Informationen über den Umgang mit EXEC-Files finden Sie in der einschlägigen DOS 3.3 -Dokumentation.

3.6 * Kompatibilität

Da die Treiberprogramme für die Ramkarte in einem statischen RAM auf der Ramkarte selbst untergebracht sind (vgl. Anhang H und S), sind in den Betriebssystemen selbst nur noch sehr geringfügige Eingriffe zum 'Einhängen' der Treibersoftware erforderlich.

Zu diesem Zweck wird bei Adresse $BD0B (RWTS) ein Unterprogrammaufruf zum Static-RAM der Ramkarte eingefügt.

Die Verwendung eines 'JSR'-Aufrufs anstelle eines absoluten Sprungbefehls hat an dieser Stelle den Vorteil, daß die Rücksprungadresse zu RWTS-Routine im Falle eines Diskettenspeicher-Zugriffs auf dem 6502-Stack liegt und so der Treibersoftware nicht explizit bekannt sein muß Man erreicht auf diese Weise, daß das Treiberprogramm den Rücksprung zur RWTS-Routine auch dann noch korrekt ausführen kann, wenn das Betriebssystem nach der Initialisierung der Ramkarte durch einen Relocat im Speicher verschoben wurde.

Bei einem Ramdisk-Zugriff wird die RWTS-Rücksprungadresse dagegen vom Stack entfernt und der RWTS-Aufruf endet im Static RAM.

Außer dem Einsprung in die Ramkarten-Treibersoftware sind zur Betriebssystemanpassung noch zwei weitere Maßnahmen erforderlich.
Es handelt sich dabei um die Stillegung des DOS 3.3 INIT-Befehls und u die Erweiterung des zulässigen Wertebereiches für die Drive-Nummer, sofern dies aufgrund der Speicherkapazität der verwendeten Ramkarte notwendig ist.

Die INIT-Funktion wird auf RWTS-Ebene dadurch außer Betrieb gesetzt, daß der entsprechende Sprungbefehl bei der RWTS-Kommando-Parameter-auswertung zum Fehlerausgang 'write protected' umgeleitet wird.

Der Wertebereich für die Drive-Nummern-Eingabe ist durch einen Tabelleneintrag (Adresse $A95B) im DOS 3.3 Kommando-Interpretierer festgelegt.
Diese Eintragung muß korrigiert werden, wenn bei der Ramkarten-Initialisierung mehr als zwei Pseudo-Laufwerke entstehen.

Alle Adreßangaben in den vorangegangenen Ausführungen beziehen sich auf die üblichen 48k-DOS 3.3-Systeme.
Die Initialisierungsprogramme können sich jedoch auch an relokalisiert DOS-Varianten anpassen.
Hierzu wird die aktuelle Position des Betriebssystems durch Auswerten des RWTS-Vektors (Adresse $3D9) ermittelt (und überprüft).

Es ist aus diesem Grund notwendig, daß sich die Vektortabelle zum Zeitpunkt der Ramkarten-Initialisierung noch im Originalzustand befindet.

4. Initialisierungsprogramme für Apple ProDos

4.1 Voraussetzungen für den Betrieb der Ramkarte unter ProDos

Die Initialisierungsprogramme 'ERAM' und 'ERAM.SYSTEM' eignen sich für die ProDos-(MLI)-Versionen 1.0.1 und 1.1.1.

Mit Ausnahme der Slots 0 und 3 kann die Ramkarte dabei in jedem beliebigen Peripherieslot des Rechners installiert sein. Wie schon in Abschnitt 2.1 erwähnt, sind allerdings die Slotnummern 4 bis 7 nach Möglichkeit vorzuziehen.

4.2 Die Initialisierungsprogramme 'ERAM' und 'ERAM.SYSTEM', Version 1

4.2.1 Anwendungsbereich, Bedienung und Wirkungsweise

Bei den beiden, auf der Rückseite der Multidiskette (vgl. 2.2) befindlichen Initialisierungsprogrammen 'ERAM' und 'ERAM.SYSTEM' handelt es sich im Prinzip um ein und dasselbe Programm. Der Unterschied liegt allein in der Art des Programmaufrufs.

'ERAM' ist ein 'binary-File' und wird mit dem 'BRUN'-Befehl, zum Beispiel vom BASIC-System ('BASIC.SYSTEM') aus gestartet. 'ERAM' eignet sich daher für diejenigen Anwendungsfälle, in denen sich zum Zeitpunkt der Ramkarten-Initialisierung bereits ein ProDos-Kommando-Interpretierer im Arbeitsspeicher des Rechners befindet.

Demgegenüber ist 'ERAM.SYSTEM' ein eigenständiges ProDos-'System-Programm'.
Hier erfolgt der Aufruf entweder automatisch unmittelbar nach dem Booten (vgl. 4.4), von BASIC aus mit dem '-'-Befehl, oder durch einen, normalerweise menügesteuerten Dialog vom jeweils vorausgegangenen ProDos-Systemprogramm.

In jedem Fall steht nach Ablauf des Initialisierungsprogramms ein Pseudo-Laufwerk mit dem Volume-Namen '/ERAM/' zur Verfügung.

Wegen des Speicherplatzbedarfes von Bootcode, Volume-Directory und Volume-Bitmap (vgl. ProDos Technical Reference Manual, Apple Computer) ist die nutzbare Speicherkapazität um 3,5 kByte (entsprechend 7 Blocks niedriger als die Gesamtkapazität der Ramkarte.
(121 Blocks für 64 kByte bis zu 2041 Blocks für 1 MByte, ein Block enthält 512 Bytes.)

Bootet man das ProDos-Betriebssystem bei bereits zuvor unter ProDos initialisierter Ramkarte, dann wird die Ramkarte vom ProDos-Kaltstartlader automatisch in das System eingebunden, sodaß ein erneuter Aufruf des Initialisierungsprogrammes eigentlich nicht erforderlich ist.
Nur das Initialisierungsprogramm trägt jedoch die Ramkarte in der sogenannten Device-Liste des ProDos-MLI-Systems (vgl. ProDos Technical Reference Manual, Fa. Apple Computer) derart ein, daß das Betriebssystem als Ausgangspunkt für gewisse Suchvorgänge nicht mehr den (mechanischen) Boot-Drive, sondern vielmehr die Ramdisk betrachtet. Dies führt in bestimmten Situationen zu einer höheren Arbeitsgeschwindigkeit.

Das ProDos-Betriebssystem kann auch von der Ramkarte aus gebootet werden, vorausgesetzt, daß die Ramkarte zumindest das File 'PRODOS' (MLI) enthält, sowie ein Systemprogramm mit dem Filenamenzusatz '.SYSTEM'.
Ein solcher 'Kaltstart' läuft selbstverständlich von der Ramdisk erheblich schneller ab, als von einem Floppy-Disk-Laufwerk.

Wenn in Ihrem Computersystem ein erphi AFDC2- oder ein AFDC3- Floppy-Disk-Controller installiert ist, dann ist es entscheidend, auf welche Weise Sie den Bootprozeß von der Ramkarte aus starten.

Um einen 'AUTOPATCH-Boot' (vgl. AFDC2/3-Handbuch der Fa. erphi GmbH) zu erreichen, ist die Eingabe von 'PR #s' (Basic) bzw. 's contr. P' (Monitor) erforderlich, wobei der Kleinbuchstabe 's' hier stellvertretend für die Slotnummer der Ramkarte steht.

Ein 'Originalsystem-Boot' wird sinngemäß mit den Befehlen 'IN #s' (Basic) bzw. 's contr. K' (Monitor) ausgelöst.
Bei AFDC2-Controllern mit einem älteren Firmware-ROM als Version 6.8 ist ein 'AUTOPATCH-Boot' von der Ramkarte aus nicht möglich.

Als Besitzer eines Apple IIe Computers können Sie mit der Tastenkombination 'control - offener Apfel - Reset' nur dann einen Ramkartenboot einleiten, wenn die Slotnumer der Ramkarte höher ist, als diejenige des im System installierten Floppy-Disk-Controller.
In Verbindung mit den erphi-Controllern AFDC2 und AFDC3 ergibt sich durch diese Methode ein 'AUTOPATCH-Boot'.

4.2.2 ERAM Version 1.2, MLI-Schnittstelle

Bei Verwendung der Initialisierungsprogramme 'ERAM' bzw. 'ERAM.SYSTEM' (Version 1.2) wird die Ramkarte entsprechend der im 'ProDOS Technical Reference Manual' dargelegten Richtlinien in das Betriebssystem eingebunden.

Die Befehlsparameter aus der Parameterliste des MLI-Aufrufs werden der Treibersoftware daher über die Zeropage-Speicherplätze $42 bis $47 zugeordnet.

Neben 'read' und 'write' (Befehlscode 1 bzw. 2) gibt es noch die Funktionen 'status request' (Befehlcode 0) und 'format' (Befehlscode 3

Während der Formatier-Befehl bei einer Ramkarte natürlich keine Au wir kungen hat, liefert die Status-Funktion die Gesamtzahl der Speiche blöcke in den Prozessor-Registern X und Y (höherwertiges Byte in Y).

Vor jedem Lese- oder Schreib-Vorgang wird die in der Parameterlist angegebene Blocknummer überprüft. Liegt sie außerhalb des durch d Ramkartenkapazität vorgegebenen Wertebereiches, dann endet der Auf auf der Treiber-Routine mit dem Fehlercode $27 (I/O-Error).
Das gleiche gilt bei unzulässigem Befehlsparameter.

Die Speicherplätze $CsO1, $CsO3, $CsO5 und $CsO7 im Static-RAM der Ram karte enthalten die übliche Kennung, die auch für Floppy-Disk-Spei her verwendet wird. (Der Kleinbuchstabe s steht hier stellvertretend für d Slotnummer der Ramkarte.)

Die Positionen $CsFC und $CsFD enthalten die Gesamtzahl der Speicher blöcke der Karte, die natürlich vom jeweiligen Speicherausbau abhängt. Damit läßt sich die Speicherkapazität der Ramkarte (- auch von einem Anwenderprogramm aus -) entweder mit Hilfe der 'status request'-Funk tion, oder aber durch Ablesen dieser Eintragung ermitteln.

Das letzte Byte der I/O-Page ($CsFF) gibt den Offset des Treiber programm-Einsprungpunktes an.
(Ein Sprung zur Adresse $CsOO bewirkt einen System-Boot !).

Der Speicherplatz mit der Adresse $CsFE beinhaltet ein Signaturbyte, aus dem die wichtigsten Merkmale des Ramkarten-Speichers hervorgehen.

Die einzelnen Bitpositionen in diesem Byte haben dabei folgende Bedeutung:

bit 0	:	'satus request'-Funktion vorhanden	(1)
bit 1	:	Lesen möglich	(1)
bit 2	:	Schreiben möglich	(1)
bit 3	:	Formatieren möglich	(1)
bit 4, 5	:	Anzahl der Volumes (nur ein Volume)	(0)
bit 6	:	Interrupt möglich	(1)
bit 7	:	Medium austauschbar (nein)	(0)

Das Signaturbyte hat somit den Wert $4F.

Man beachte, daß die obengenannten Eigenschaften im allgemeinen auch auf Hard-Disk Speicher zutreffen.
Dies ist auch die Erklärung dafür, daß die Ramkarte von einigen Start programmen, die den jeweiligen Ausbau der Rechner-Peripherie anzeigen, als 'Profile'-Harddisk ausgewiesen wird.

4.2.3 * ERAM Version 1.2, Speicherorganisation

Die Umrechnung der bei Lese- oder Schreib-Befehlen angeforderten Blocknummer in die Bestandteile der Ramkarten-Gesamtadresse (vgl. Anhang H und S) geht bei der vorliegenden Treibersoftware so vor sich, daß die Datenblöcke in linear geordneter Reihenfolge im I/O-Expansions-Adressbereich erscheinen.
Dies bedeutet mit anderen Worten, daß je nach Auswahl der Parameter Spaltenadresse und Bankummer im Falle einer mit 64k DRAMs bestückten Ramkarte jeweils auf einen halben Block (256 Bytes) im Adressbereich $C800..$C8FF direkt zugegriffen werden kann.
Bei einem Speicherausbau mit 256k DRAMs stehen jeweils ein geradzahliger und ein ungeradzahliger Block im Bereich $C800..CBFF zur Verfügung.
Es entsteht somit im Ramkartenspeicher ein Datenfenster von 256 bzw. 1024 Byte Größe, dessen Inhalt beliebig gelesen oder verändert werden kann.

Für viele Anwendungsfälle ist dieser Speicherausschnitt groß genug, um auf den sonst üblichen Transfer der Ramkarten-Daten in einen Pufferspeicher vollständig verzichten zu können.
Neben der Einsparung eines Datenpuffers läßt sich dadurch, - zum Beispiel bei Such- oder Sortier-Vorgängen - eine beträchtliche Zeiteinsparung erzielen.

Spaltenadresse und Banknummer erhält man bei Bestückung mit 64k DRAMs aus der Vorschrift:

```
                                Banknummer :  Blocknummer(7..8)
  Spaltenadresse für untere Blockhälfte :  Blocknummer(0..6) * 2
  Spaltenadresse für obere  Blockhälfte :  Blocknummer(0..6) * 2 + 1
```

Bei RamKarten, die mit 256k DRAMs bestückt sind, werden durch die Vorgabe von Spaltenadresse und Banknummer zwei Datenblöcke selektiert:

```
  Banknummer      : Blocknummer(9..10)
  Spaltenadresse  : Blocknummer(1.. 8)
```

Bei geradzahliger Nummer des angeforderten Datenblocks findet man hier die gewünschten Daten im Adressbereich $C800..$C9FF, bei ungerader Blocknummer dagegen im Bereich $CA00..$CBFF.

(Die in Klammer gesetzten Bereichsangaben kennzeichnen Bitpositionen bei binärer Darstellung der Blocknummer)

4.3 Das Kopieren von ProDos-Dateien

Für das Kopieren von Dateien gelten in Verbindung mit der Ramkarte dieselben Regeln wie bei Floppy-Disk-Laufwerken.

Sogenannte Spurkopierer oder Schnellkopierprogramme sind jedoch nicht zu gebrauchen; sie würden bei Beteiligung von Ramdisks auch nur einen deutlich geringeren Geschwindigkeitsvorteil bringen, als beim normalen Einsatz mit mechanischen Laufwerken.

Das ProDos-Systemprogramm 'FILER' ist ohne Änderung vollständig kompatibel.
Ein Kopieren oder Vergleichen vollständiger Volumes mit Hilfe der entsprechenden Funktionen aus den Volume-Kommandos des Filers ist allerdings grundsätzlich nur dann möglich, wenn Quellen- und Ziel-Volume die gleiche Speicherkapazität aufweisen.
Diese Voraussetzung dürfte aber beim Umkopieren von Ramkartenspeichern nur selten erfüllt sein.

4.4 Automatische Ramkarten-Initialisierung

Um einen automatischen Aufruf des Initialisierungsprogrammes unmittelb nach dem Booten zu erreichen, gibt es ein ganz einfaches Verfahren: Man sorgt dafür, daß 'ERAM.SYSTEM' im Volume-Directory der Boot-Disket als erstes File mit dem Filenamen-Zusatz '.SYSTEM' eingetragen ist.

Mit dieser Maßnahme allein ist allerdings noch nichts gewonnen, da unt diesen Umständen ja das eigentlich benötigte Anwenderprogramm nach der Ramkarten-Initialisierung 'von Hand' gestartet werden müsste.

Um auch diesen Schritt zu automatisieren, ist es lediglich erforderlic den Filenamen von 'ERAM.SYSTEM' in 'AUTOERAM.SYSTEM' abzuändern.

Eine spezielle Laderoutine des Initialisierungsprogramms sucht dann nach dem Initialisieren der Ramkarte im Volume-Directory der Boot-diskette das nächstfolgende Systemprogramm, dessen Filename auf '.SYSTEM' endet und führt anschließend den gewünschten Programmaufruf selbsttätig durch.

Die Binär-File-Variante von 'ERAM' Version 1.2 lässt sich auch von ein BASIC-Programm aus starten.
Verwendet man dazu unter 'BASIC.SYSTEM' das 'STARTUP'-File, dann er-öffnet sich ein weiterer Weg zu einer automatischen Ramkarten-Initiali sierung.

Das Aufrufen von Maschinensprache-Programmen von BASIC aus ist jedoch nicht ganz unproblematisch.
Man muß hierbei nämlich sicherstellen, daß es zwischen BASIC-Programm und eingeladenem Binärfile zu keinen Speicherplatzkonflikten kommt.
(Das Initialisierungsprogramm 'ERAM' Version 1.2 belegt den Arbeits-speicher ab Adresse $2000.)

Abschließend sei noch erwähnt, daß 'ERAM' auch von einem 'EXEC'-File aus gestartet werden kann.
Es besteht so die Möglichkeit, die Initialisierung der Ramkarte mit eventuell erforderlichen Kopierarbeiten zu kombinieren.

4.5 * Kompatibilität

Bei der Konzeption von ProDos wurde offenbar großer Wert darauf gelegt daß ein Betriebssystem entsteht, dessen Schnittstellen zur Peripherie flexibel genug gestaltet sind, daß bei einem Ausbau der Massenspeicher keine unüberwindbaren Software-Probleme entstehen.
Es ist aus diesem Grund nicht überraschend, daß zum 'Einbinden' einer Ramdisk keinerlei 'inoffizielle Betriebssystem-Modifikationen' erforderlich sind.

Das Initialisierungsprogramm muß hierzu lediglich in der 'System Globa Page' eine Vektoradresse eintragen und die 'Device-Liste' entsprechend ergänzen (vgl. ProDos Technical Reference Manual, Fa. Apple Computer).

Wesentlich schwieriger ist es normalerweise, einen geeigneten Speicherplatz für die benötigten Geräte-Treiberroutinen zu finden.
Diese Problematik ist im vorliegenden Fall jedoch dadurch vollkommen entschärft, weil die Treiber-Programme für die Ramkarte in einem spezi zu diesem Zweck vorgesehenen statischen RAM auf der Ramkarte selbst untergebracht sind (vgl. Anhang H und S).

Unverträglichkeiten mit anderen, nachträglich in das System eingefügte Peripherie-Geräten, sind daher kaum zu erwarten.

5. Initialisierungsprogramme für Apple Pascal

5.1 Voraussetzungen für den Betrieb der Ramkarte unter Apple Pasca

Mit Hilfe der in den folgenden Abschnitten detailliert beschriebenen Initialisierungsprogramme können Sie Ihre Ramkarte auch unter dem Apple-Pascal-Betriebssystem einsetzen.
Dabei werden folgende System-Varianten unterstützt:

Apple Pascal 1.1
Apple Pascal 1.2 / 64k, 1.2 / 128k
Apple Pascal 1.3 / 64k, 1.3 / 128k

Die inzwischen nur noch sehr selten anzutreffende, ältere System Vers 1.0 wurde bei der Konzeption der Ramkarten-Software nicht mehr berück sichtigt.

Bei Apple Pascal 1.2 sind auch runtime-Ausführungen zulässig (64k und 128k). Die Initialisierungs-Software liefert allerdings bei diesen Spezial-Systemen während der Programmausführung nur dann Informationen auf der Console, wenn bei der Initialisierung irgendwelche Fehler-situationen eintreten.

Die Ramkarte muß sich für den Betrieb mit Apple Pascal in einem der Erweiterungs-Steckplätze (Slots) 4 bis 7 befinden (vgl. 2.1).

5.2 Gemeinsame Eigenschaften der Initialisierungsprogramme

Die vorliegenden Initialisierungsprogramme 'ERAM.CODE', 'ERAMC.CODE', 'ERAMCS.CODE' und 'ERAMCS4.CODE' installieren in Ihrem Pascal-System jeweils ein einzelnes Pseudo-Laufwerk mit dem Volume-Namen 'ERAM:'.

Die nutzbare Speicherkapazität ergibt sich aus der Gesamtkapazität der Ramkarte unter Abzug von 3 kByte, die für Bootcode und Directory benö-tigt werden.

Von Pascal-Anwenderprogrammen aus lassen sich damit alle I/O-Funktione die normalerweise bei FLoppy-Disk-Laufwerken angewendet werden, auch i Verbindung mit der Ramkarte benutzen.
Hierzu zählen sowohl Prozeduren wie 'OPEN', 'CLOSE', 'READ', 'WRITE' usw., als auch die stärker hardware-orientierten Funktionen 'BLOCKREAD und 'BLOCKWRITE', sowie die Prozeduren 'UNITREAD', 'UNITWRITE', 'UNITCLEAR' und 'UNITSTATUS'.

Vor jeder Datenübertragung wird überprüft, ob die Parameter 'Start-Blocknummer' und 'Anzahl der zu übertragenden Bytes' noch in dem durch die Speicherkapazität der Ramkarte vorgegebenen Rahmen liegen.
Ist dies nicht der Fall, dann wird der Fehlercode 'IORESULT=17' ausge-geben.

Bei der 'UNITSTATUS'-Prozedur ist zu beachten, daß die Übergabe der Ergebnis-Parameter von den einzelnen Pascal-System-Varianten unterschiedlich gehandhabt wird.
Während bei den älteren Versionen 1.1 und 1.2 ein ganzer 'status-record' mit den Komponenten 'buffered bytes', 'bytes/sector' und 'Spurzahl' übertragen wird, besteht das Ergebnis ab System Version 1.3 aus einem einzelnen 16-Bit-Wort, das die Gesamtzahl der vorhandenen Speicherblöcke enthält (vgl. Barry Haynes, Attach Bios Document for Apple II Pascal 1.1, bzw. Apple Pascal 1.3, Language Manual).

Mit Ausnahme von 'ERAM.CODE' bieten alle Initialisierungsprogramme die Möglichkeit, entweder den gesamten Inhalt der Bootdiskette, oder nur einen Teilausschnitt davon bei der Initialisierung in die Ramkarte zu kopieren.
(Dies geschieht jedoch selbstverständlich nur dann, wenn die Ramkarte bei Ablauf des Initialisierungsprogramms noch keine Dateien enthält.)

Start und Ende dieses Kopiervorganges lassen sich durch spezielle Mark im Inhaltsverzeichnis der Bootdiskette festlegen.

Bei diesen Marken handelt es sich um Directory-Einträge (File-Länge ei Block) mit den Namen 'ERAM.STRCOPY' und 'ERAM.ENDCOPY'.
Diese Bezeichnungen setzen sich also aus dem jeweiligen Volume-Namen d Ramkarte und dem Filenamenzusatz '.STRCOPY' bzw. '.ENDCOPY' zusammen.
Zum Einfügen der Marken in das Directory der Bootdiskette verwendet ma am besten die Filer-Funktionen 'K(runch' und 'M(ake File' (vgl. Apple Pascal Operating System Manual).
Fehlt die Start-Markierung, dann beginnt das Kopieren mit dem ersten File der Bootdiskette. Dementspreched endet der Filetransfer ohne End-Markierung bei der letzten Datei der Bootdiskette.
Verzichtet man auf beide Markierungen, dann wird der gesamte Disketten inhalt in die Ramkarte übertragen.
Selbstverständlich wird der Kopiervorgang vorzeitig abgebrochen, wenn die Speicherkapazität der Ramkarte erschöpft ist.

5.2.1 Das Initialisierungsprogramm 'ERAM.CODE' Version 4.0

'ERAM.CODE' ist das einfachste der vier Initialisierungsprogramme.

Hier wird die Ramkarte lediglich in das Betriebssystem 'eingehängt' und ein neues Directory angelegt (sofern ein solches nicht schon in der Ramkarte vorhanden ist).

Die Unit-Nummer, die der Ramkarte bei der Initialisierung zugeordnet wird, richtet sich u.a. nach der Nummer des Peripherieslots, in dem si die Karte befindet.
Ist die Ramkarte in Slot 5 eingesetzt, dann ergibt sich normalerweise Unit #11, bei Slot 4 Unit #9.
Eine besondere Situation liegt vor, wenn die Karte in Slot 7 eingesetz ist. In diesem Fall können bis zu 3 Floppy-Disk-Controller im System installiert sein, sodaß die üblichen Unit-Nummern für 'geblockte Volumes' bereits belegt sind.

Bei den Systemen 1.2 und 1.3 wird unter diesen Umständen Unit #13 verwendet.
Dies ist bei den älteren 1.1-Systemen jedoch nicht möglich (- die höchste Unit-Nummer hat dort Unit #12 -), sodaß auf Unit #10 ausgewich werden muß, mit der Folge, daß kein Zugriff zu dem eventuell zuvor unter dieser Nummer erreichbaren Floppy-Disk-Laufwerk mehr möglich ist

Abweichungen von den obengenannten Regeln sind unter Apple Pascal 1.3 möglich, wenn das System nicht von Slot 6 gebootet wurde.
Die übliche Zuordnung der Floppy-Disk-Laufwerke zu den Unit-Nummern wird dann nämlich bereits vom Pascal-Kaltstartlader verändert, um zu erreichen, daß der Boot-Drive stets als Unit #4 installiert wird.

Das Ramkarten-Initialisierungsprogramm kann sich an jede dieser Situationen automatisch anpassen und zeigt die für die Ramkarte ausgewählte Unit-Nummer am Ende der Initialisierung auf der Console an. (Dies gilt nicht für 'runtime'-Systeme).

5.2.2 Das Initialisierungsprogramm 'ERAMC.CODE', Version 4.0

Verwendet man 'ERAMC.CODE' zur Initialisierung der Ramkarte, dann erhä man ein Pseudo-Laufwerk mit denselben Eigenschaften, wie beim Aufruf v 'ERAM.CODE'.

Der Unterschied zwischen diesen beiden Initialisierungsprogrammen besteht darin, daß 'ERAMC.CODE' Software zum Kopieren von Dateien der Boot-Diskette (Unit #4) in die Ramkarte enthält.

Alle wesentlichen Details über die Ausführung dieses Kopiervorganges wurden bereits in Abschnitt 5.2 beschrieben.

5.2.3 Das Initialisierungsprogramm 'ERAMCS.CODE', Version 4.0

Das Arbeiten mit dem Apple Pascal Betriebssystem wird bedeutend angene mer, wenn zumindest die am häufigsten benötigten Dienstprogramme wie Editor, Filer, Compiler usw. und natürlich auch die einzelnen Segmente von SYSTEM.PASCAL selbst nicht jedesmal von der Diskette nachgeladen werden müssen.

Mit dem Initialisierungsprogramm 'ERAMCS.CODE' läßt sich genau dieser Zustand erreichen.

Dies geht wie folgt vor sich:

Zunächst läuft der Initialisierungsvorgang genauso ab, wie bei Verwendung von 'ERAMC.CODE', d.h. es wird ein Pseudolaufwerk im System einge bunden und anschließend verschiedene Dateien von der Boot-Diskette in die Ramkarte übertragen (vgl. 5.2 und 5.2.2).

Sobald der Kopiervorgang beendet ist, wird das Inhaltsverzeichnis der Ramkarte daraufhin überprüft, ob die Files 'SYSTEM.PASCAL' und 'SYSTEM.MISCINFO' in der Ramkarte enthalten sind.
Ist dies der Fall, dann verändert die Initialisierungssoftware einige Variablen des Betriebssystems derart, daß die Ramkarte von diesem Zeitpunkt an bei allen Systemoperationen als 'System-Volume' betrachtet wird.

Dienstprogramme, die man von der Kommandozeile des Systems aus aufruft werden dann stets zuerst in der Ramkarte gesucht und - soweit dort vorhanden - auch von der Ramkarte geladen.

Die volle Leistungsfähigkeit einer solchen 'Systemübertragung' kommt daher erst dann voll zur Geltung, wenn man dafür sorgt, daß bei der Initialisierung außer den Dateien 'SYSTEM.PASCAL' und 'SYSTEM.MISCINFO nach Möglichkeit auch noch alle anderen System-Files in die Ramkarte kopiert werden.

Als 'Mindestausstattung' sollte man hier die folgende Zusammenstellung betrachten:

SYSTEM.APPLE (vgl. S.3)
SYSTEM.PASCAL
SYSTEM.MISCINFO
SYSTEM.LIBRARY
SYSTEM.CHARSET
SYSTEM.FILER
SYSTEM.EDITOR

Je nach Arbeitsgebiet kommen natürlich auch Compiler, Assembler, Linke usw. in Frage.

Arbeitsdateien wie 'SYSTEM.WRK.TEXT' und 'SYSTEM.WRK.CODE' wird man normalerweise natürlich auch in die Ramkarte verlegen.
Es versteht sich dabei von selbst, daß man diese Dateien von Zeit zu Zeit 'gegen Stromausfall' sichern muß.

Bei der Ramkarten-Initialisierung mit 'ERAMCS.CODE' behalten alle im System installierten 'geblockten Volumes' ihre ursprüngliche Unit-Nummer bei.
Nach einer 'Systemübertragung' in die Ramkarte ist folglich als System Unit-Nummer' nicht mehr die Unit Nummer 4 festgelegt, sondern vielmehr die der Ramkarte zugewiesene Nummer.
Dies hat zur Folge,daß einige Dateien (z.B 6500.ERRORS) nach wie vor in demjenigen Floppy-Disk-Laufwerk erwartet werden, dem die Unit-Numme #4 zugewiesen ist.
Obwohl diese Fälle relativ selten sind, ist daher die mit 'ERAMCS.CODE erreichte 'Systemübertragung' noch nicht ganz vollständig.

Ein Vorteil ist dabei aber, daß normalerweise keine Probleme mit kopie geschützter Software auftreten, deren Disketten nur in Laufwerken mit Unit Nummer #4 gelesen werden können.

5.2.4 Das Initialisierungsprogramm 'ERAMCS4.CODE', Version 4.0

Dieses Initialisierungsprogramm entspricht weitgehend der Ausführung 'ERAMCS.CODE' (vgl. 5.2.3).
Es läßt sich jedoch hier eine vollständige 'Systemübertragung' erreichen, d.h. es kommt zu einer Vertauschung der bisherigen Zuordnung der 'geblockten Volumes' zu den Unit-Nummern, sodaß der Ramkarte als neuem 'System-Volume' die Unit-Nummer #4 zugeteilt werden kann.

Dieser Vorgang verläuft nach denselben Richtlinien, die auch für die derzeit neueste Apple-Pascal-Variante, Version 1.3, festgelegt worden sind. Dort gibt es ein vergleichbares Problem, wenn der Kaltstart nich auf einem in Slot 6 installierten Floppy-Disk-Laufwerk stattfindet.

Unit-Nummern werden stets paarweise getauscht. Dabei bekommen diejenig Geräte (Floppy-Disk-Laufwerke), die ursprünglich als Units #4 und #5 installiert wurden, Unit-Nummern zugeteilt, die sonst der Slotnummer zugeordnet sind, in der sich die Ramkarte befindet.

Ein Sonderfall entsteht hierbei allerdings, wenn die Ramkarte den Peripherieslot 7 belegt, der bei Apple-Pascal offiziell nicht für 'Geräte mit Blockstruktur' vorgesehen ist.
Bei den Apple-Pascal-Varianten 1.2 und 1.3 werden dann die Unitnummern #13 und #14 herangezogen und mit den Nummern #4 und #5 getauscht.

Dies funktioniert bei dem älteren System Apple Pascal 1.1 leider nicht weil dort der Bereich der Unit-Nummern nur bis Unit #12 reicht.
Die Units #4 und #5 werden hier zunächst mit den Units #9 und #10 vertauscht. Das ursprünglich unter der Nummer #9 erreichbare Laufwerk bekommt dann die Unit-Nummer #5 zugewiesen; das 'alte' Unit #10 ist dagegen nicht mehr ansprechbar.

Um durch diese Umverteilung von Unit-Nummern keine allzugroße Verwirrung entstehen zu lassen, wird die neue Konfiguration beim Ablauf des Initialisierungsprogramms auf der Console angezeigt (dies gilt nicht bei 'runtime'-Systemen).

Man sollte sich in diesem Zusammenhang aber auch daran erinnern, daß die Adressierung von 'Volumes' beim Apple-Pascal-Betriebssystem im Normalfall nicht über Unit-Nummern vor sich geht, sondern daß vielmehr (symbolische) Volume-Namen verwendet werden, deren Zuordnung stets unverändert bleibt (z.B. APPLE1: oder ERAM:).
Im praktischen Betieb kommt daher der Vergabe von Unit-Nummern weit weniger Bedeutung zu, als dies hier aufgrund der umfangreichen Darstellung dieser Problematik den Anschein hat.

5.2.5 Pascal-Boot von der Ramkarte

Apple Pascal kann auch von der Ramkarte aus gebootet werden.
Dies setzt allerdings voraus, daß die Ramkarte zumindest die Files 'SYSTEM.APPLE', 'SYSTEM.PASCAL' und 'SYSTEM.MISCINFO' enthält.

Ein solcher 'Kaltstart' läuft selbstverständlich von der Ramdisk erheblich schneller ab, als von einem Floppy-Disk-Laufwerk.

Wenn in Ihrem Computersystem ein erphi AFDC2- oder ein AFDC3-Floppy-Disk-Controller installiert ist, dann ist es entscheidend, auf welche Weise Sie den Bootprozeß von der Ramkarte aus starten.

Um einen 'AUTOPATCH-Boot' (vgl. AFDC2/3-Handbuch der Fa. erphi GmbH) zu erreichen, ist die Eingabe von 'PR #s' (Basic) bzw. 's contr. P' (Monitor) erforderlich, wobei der Kleinbuchstabe 's' hier stellvertretend für die Slotnummer der Ramkarte steht.
Ein 'Originalsystem-Boot' wird sinngemäß mit den Befehlen 'IN #s' (Basic) bzw. 's contr. K' (Monitor) ausgelöst.
Bei AFDC2-Controllern mit einem älteren Firmware-ROM als Version 6.8 ist ein 'AUTOPATCH-Boot' von der Ramkarte aus nicht möglich.

Nach einem Boot von der Ramkarte weicht die Zuordnung der Unit-Nummern Ihrer Floppy-Disk-Laufwerke etwas von dem gewohnten Schema ab, das entsteht, wenn von einem in Slot 6 installierten Floppy-Disk-Drive gebootet wird.

Die Ursache hierfür liegt darin, daß das Boot-Volume beim Kaltstart vo Apple Pascal stets als Unit #4 im System verankert wird.
Die beiden Floppy-Disk-Laufwerke von Slot 6, die im Normalfall den Unit-Nummer #4 und #5 zugeordnet sind, bekommen daher neue Unit-Nummern zugeteilt.

Beim 'Kaltstart' von der Ramkarte entsteht dieselbe System-Konfiguration, die man auch bei Anwendung des Initialisierungsprogramms 'ERAMCS4.CODE' erhalten würde (vgl. 5.2.4).
Dies gilt unabhängig davon, mit welchem der 4 Initialisierungsprogramm die Ramkarte ursprünglich installiert wurde.

5.2.6 * ERAM, ERAMC, ERAMCS, ERAMCS4, Speicherorganisation

Die Umrechnung der bei Lese- oder Schreib-Befehlen (UNITREAD/UNITWRITE) angeforderten Blocknummern in die Bestandteile der Ramkarten-Gesamtadresse (vgl. Anhang H und S) geht bei der vorliegenden Treibersoftware so vor sich, daß die Datenblöcke in linear geordneter Reihenfolge im I/O-Expansions-Adreßbereich erscheinen.
Dies bedeutet mit anderen Worten, daß je nach Auswahl der Parameter Spaltenadresse und Banknummer im Falle einer mit 64k DRAMs bestückten Ramkarte jeweils auf einen halben Block (256 Bytes) im Adressbereich $C800..$C8FF direkt zugegriffen werden kann.
Bei einem Speicherausbau mit 256k DRAMs stehen jeweils ein geradzahliger und ein ungeradzahliger Block im Bereich $C800..CBFF zur Verfügung.
Es entsteht somit im Ramkartenspeicher ein Datenfenster von 256 bzw. 1024 Byte Größe, dessen Inhalt beliebig gelesen oder verändert werden kann.

Für viele Anwendungsfälle ist dieser Speicherausschnitt groß genug, um auf den sonst üblichen Transfer der Ramkarten-Daten in einen Pufferspeicher vollständig verzichten zu können.
Neben der Einsparung eines Datenpuffers läßt sich dadurch, - zum Beispiel bei Such- oder Sortier-Vorgängen - eine beträchtliche Zeiteinsparung erzielen.

Spaltenadresse und Banknummer erhält man bei Bestückung mit 64k DRAMs aus der Vorschrift:

	Banknummer :	Blocknummer(7..8)
Spaltenadresse für untere Blockhälfte	:	Blocknummer(0..6) * 2
Spaltenadresse für obere Blockhälfte	:	Blocknummer(0..6) * 2 + 1

Bei Ramkarten, die mit 256k DRAMs bestückt sind, werden durch die Vorgabe von Spaltenadresse und Banknummer zwei Datenblöcke selektiert:

Banknummer	: Blocknummer(9..10)
Spaltenadresse	: Blocknummer(1.. 8)

Bei geradzahliger Nummer des angeforderten Datenblocks findet man hier die gewünschten Daten im Adressbereich $C800..$C9FF, bei ungerader Blocknummer dagegen im Bereich $CA00..$CBFF.

(Die in Klammer gesetzten Bereichsangaben kennzeichnen Bitpositionen bei binärer Darstellung der Blocknummer)

5.3 Das Kopieren von Apple Pascal Dateien

Zur Durchführung von Kopierarbeiten wird beim Apple-Pascal-Betriebssystem fast ausschließlich das System-Programm 'SYSTEM.FILER' verwendet, welches unmittelbar von der 'Kommandozeile' des Systems aus aufgerufen wird.

Der 'Filer' ist auch in Verbindung mit der Ramkarte uneingeschränkt einsetzbar.

Nicht zu empfehlen ist allerdings das Kopieren vollständiger Volumes, denn dies ist nur dann zulässig, wenn Quellen- und Ziel-Volume die gleiche Speicherkapazität aufweisen.
Bei Beteiligung einer Ramkarte ist diese Voraussetzung aber fast nie erfüllt.
Leider geben die älteren Filer-Ausführungen keinerlei Warnmeldung aus, wenn man versucht, ungleich große Volumes zu kopieren.

5.4 Automatische Ramkarten-Initialisierung

Eine automatische Ramkarten-Initialisierung direkt nach dem Booten ist bei Apple Pascal am einfachsten dadurch zu erreichen, daß man dem jeweiligen Initialisierungsprogramm den Namen 'SYSTEM.STARTUP' gibt.

Etwas komplizierter ist die Situation allerdings dann, wenn der Filenam 'SYSTEM.STARTUP' bereits für den Start eines anderen Programms belegt ist.

Bei den Systemversionen 1.1 und 1.2 läßt sich unter diesen Umständen häufig auch der Filename 'SYSTEM.ATTACH' verwenden.
(Wenn auf der Bootdiskette ein File mit diesem Namen vorhanden ist, dann wird es noch vor 'SYSTEM.STARTUP' abgearbeitet).
Dabei ist jedoch zu beachten, daß das 'SYSTEM.ATTACH'-File ursprünglic zum Einbinden von Treibersoftware in das Apple-Pascal Betriebssystem vorgesehen ist (vgl. Barry Haynes, Attach Bios Document for Apple II-Pascal 1.1).

Die derzeit neueste Apple-Pascal-Variante (1.3) bietet in Verbindung m dem UNIT 'CHAINSTUFF' einige spezielle Spracherweiterungen, mit deren Hilfe von einem Pascal-Anwenderprogramm das jeweils nächste, abzu-arbeitende Programm spezifiziert werden kann.
(vgl. Apple Pascal 1.3 Language Manual, Chapter 16, Program Chaining)

5.5 * Kompatibilität

Bei dem Betriebssystem Apple Pascal gibt es gewisse Richtlinien, nach denen 'Geräte mit Blockstruktur' in das System eingebunden werden.

Eine zentrale Rolle spielt dabei die bei allen Systemvarianten ab Adresse $FEB0 im BIOS-Teil des Betriebssystems angeordnete Tabelle, in der für jedes aktive, 'geblockte Unit' entweder eine Disk-Nummer (0..5, für interne Floppy-Disk-Laufwerke), oder aber eine I/O-Handler-Startadresse (bei nachträglich hinzugefügten Geräten) eingetragen ist.

Im Falle der vorliegenden Ramkarte zeigt dieser Vektor zu dem auf der Ramkarte selbst untergebrachten statischen RAM, in dem alle benötigten Treiber-Routinen enthalten sind.

Auf BIOS-Ebene sind damit keinerlei 'inoffizielle' Eingriffe erforderlich, um die Ramkarte in das Betriebssystem einzugliedern.

Besonders vorteilhaft wirkt sich aus, daß auch kein besonderer Speicherplatz zur Unterbringung der Treibersoftware gefunden werden muß. Auf die in solchen Fällen sonst übliche Methode, bei der hierzu ein Te des System-Heaps verwendet wird, kann man daher verzichten.

Für das Umdefinieren des 'System-Volumes' gibt es demgegenüber keine offiziellen Empfehlungen.
Bei den Initialisierungsprogrammen 'ERAMCS.CODE' und 'ERAMCS4.CODE' wird zu diesem Zweck der Inhalt einiger globaler Variablen von 'SYSTEM.PASCAL' nachträglich verändert.

Wenn der Verdacht besteht, daß dadurch Kompatibilitätsprobleme mit bestimmten Anwenderprogrammen oder mit anderen, nachträglich in das System eingefügten Peripheriegeräte-Treibern entstehen, dann sollte ma überprüfen, ob diese Schwierigkeiten auch in Verbindung mit den Initialisierungsprogrammen 'ERAM.CODE' und 'ERAMC.CODE' zu beobachten sind.

Zu beachten ist, daß alle Initialisierungsprogramme die jeweilige Betriebssystem-Versions-Nummer durch Auswerten der entsprechenden Kennungs-Bytes mit den Adressen $BF21, $BF22 und $FFF6 (BIOS) ermittel Der Inhalt dieser Speicherplätze darf aus daher niemals zerstört werde

6. Initialisierungsprogramme für CP/M 2.2

6.1 Voraussetzungen für den Betrieb der Ramkarte unter CP/M 2.2

Bei Verwendung des im folgenden Abschnitt beschriebenen Initialisierungsprogramms 'ERAM.COM' (Version C.0) läßt sich die Ramkarte auch in Verbindung mit den Betriebssystemen 2.20, 2.23, 2.26 und 2.28 betreiben.

Diese von der Firma Microsoft vertriebenen Implementierungen von CP/M 2.2 (Digital Research) setzten allerdings den Einsatz spezieller Z80-Prozessorkarten im Apple II Computer voraus.

Während die beiden älteren Varianten 2.20 (56k) und 2.23 für die seit langem bekannte 'Z80-Softcard' (Microsoft) vorgesehen sind, handelt es sich bei den Systemen 2.26 und 2.28 um Neuentwicklungen für den Einsatz mit zwei wesentlich erweiterten Prozessorkarten, der 'Microsoft Premium Softcard IIe' (2.26) und der 'Microsoft Softcard II' (2.28).

Das Initialisierungsprogramm 'ERAM.COM' kann sich an jede der drei unterschiedlichen Hardware-Umgebungen anpassen und die jeweils geeignete Ramkarten-Betriebs-Software hinzufügen.
Bei den Systemen 2.26 und 2.28 wird hierzu allerdings auch ein Teil der sogenannten 'User Patch Area' benötigt (vgl. 6.5).

Die inzwischen nur noch sehr selten anzutreffenden 44k-CP/M 2.2-Systeme (Softcard) wurden bei der Entwicklung von 'ERAM.COM' nicht mehr berücksichtigt.

Für den Betrieb unter CP/M 2.2 muß sich die Ramkarte in einem der Peripherieslots 2, 4, 5 oder 7 befinden (vgl. 2.1).

Wenn Sie beabsichtigen, die Ramkarte als CP/M-Drive 'A:' zu installieren (vgl. 6.2, 'W'-Parameter) und in Ihrem Computer-System ein Floppy Disk-Controller vom Typ 'AFDC2' der Firma erphi GmbH eingesetzt ist, dann ist für diesen Controller ein Firmware ROM ab Version 6.8 erforderlich.

6.2 Das Initialisierungsprogramm 'ERAM.COM' Version C.0

6.2.1 Anwendungsbereich, Bedienung und Wirkungsweise

Das Ramkarten-Initialisierungsprogramm 'ERAM.COM' ruft man, - wie in CP/M-Systemen üblich - , durch Eingeben des Filenamens (ohne '.COM'-Erweiterung) in die Kommandozeile des CCP auf.

Die Wirkungsweise der Initialisierung wird dabei jedoch entscheidend davon beeinflußt, ob die Kommandozeile beim Programmaufruf den Parameter 'W' enthält oder nicht.

6. Initialisierungsprogramme für CP/M 2.2

6.1 Voraussetzungen für den Betrieb der Ramkarte unter CP/M 2.2

Bei Verwendung des im folgenden Abschnitt beschriebenen Initialisierungsprogramms 'ERAM.COM' (Version C.0) läßt sich die Ramkarte auch in Verbindung mit den Betriebssystemen 2.20, 2.23, 2.26 und 2.28 betreiben.

Diese von der Firma Microsoft vertriebenen Implementierungen von CP/M 2.2 (Digital Research) setzten allerdings den Einsatz spezieller Z80-Prozessorkarten im Apple II Computer voraus.

Während die beiden älteren Varianten 2.20 (56k) und 2.23 für die seit langem bekannte 'Z80-Softcard' (Microsoft) vorgesehen sind, handelt es sich bei den Systemen 2.26 und 2.28 um Neuentwicklungen für den Einsatz mit zwei wesentlich erweiterten Prozessorkarten, der 'Microsoft Premium Softcard IIe' (2.26) und der 'Microsoft Softcard II' (2.28).

Das Initialisierungsprogramm 'ERAM.COM' kann sich an jede der drei unterschiedlichen Hardware-Umgebungen anpassen und die jeweils geeignete Ramkarten-Betriebs-Software hinzufügen.
Bei den Systemen 2.26 und 2.28 wird hierzu allerdings auch ein Teil der sogenannten 'User Patch Area' benötigt (vgl. 6.5).

Die inzwischen nur noch sehr selten anzutreffenden 44k-CP/M 2.2-Systeme (Softcard) wurden bei der Entwicklung von 'ERAM.COM' nicht mehr berücksichtigt.

Für den Betrieb unter CP/M 2.2 muß sich die Ramkarte in einem der Peripherieslots 2, 4, 5 oder 7 befinden (vgl. 2.1).

Wenn Sie beabsichtigen, die Ramkarte als CP/M-Drive 'A:' zu installieren (vgl. 6.2, 'W'-Parameter) und in Ihrem Computer-System ein Floppy Disk-Controller vom Typ 'AFDC2' der Firma erphi GmbH eingesetzt ist, dann ist für diesen Controller ein Firmware ROM ab Version 6.8 erforderlich.

6.2 Das Initialisierungsprogramm 'ERAM.COM' Version C.0

6.2.1 Anwendungsbereich, Bedienung und Wirkungsweise

Das Ramkarten-Initialisierungsprogramm 'ERAM.COM' ruft man, - wie in CP/M-Systemen üblich - , durch Eingeben des Filenamens (ohne '.COM'-Erweiterung) in die Kommandozeile des CCP auf.

Die Wirkungsweise der Initialisierung wird dabei jedoch entscheidend davon beeinflußt, ob die Kommandozeile beim Programmaufruf den Parameter 'W' enthält oder nicht.

Bei Aufruf ohne 'W'-Parameter erhält man ein einzelnes Pseudolaufwerk, bei dem die gesamte Speicherkapazität der Ramkarte (nach Abzug des vom Inhaltsverzeichnis benötigten Speicherplatzes) für Dateien des Anwende zur Verfügung steht.

Dieser Pseudo-Drive ist anschließend, je nach Anzahl der im Sytem installierten Floppy-Disk-Cotroller (FDC), unter den Laufwerks-Buchstab 'C:' (ein FDC), 'E:' (zwei FDCs) oder 'G:' (3 FDCs, nur CP/M 2.20) zu erreichen.
Beim Betriebssystem 2.28 ist zu beachten, daß wegen der begrenzten Speicherplatzverhältnisse einschließlich Ramkarte nur maximal 4 Laufwerke möglich sind. Dies bedeutet, daß bei Verwendung von zwei Floppy-Disk-Controllern das zuvor als Drive 'D:' (Slot 5, Drive 2) ansprechba Laufwerk nicht mehr zur Verfügung steht. Der Laufwerksbuchstabe 'D:' wird in diesem Fall der Ramkarte zugeteilt.

Ruft man das Initialisierungsprogramm durch Eingabe der Kommandozeile 'ERAM W' auf, dann entsteht ein Ramdisk-Laufwerk, in dem auch eine Kopie des Betriebsystems vorhanden ist.

In diesem Fall bekommt die Ramkarte den Drive-Buchstaben 'A:' zugewiesen und alle übrigen Laufwerke sind unter dem jeweils nächstfolgenden Buchstaben zu erreichen ('B:' für alten Drive 'A:', 'C:' für für alten Drive 'B:' usw.).
In Verbindung mit der Betriebssystem-Variante 2.28 steht auch hier das 'alte' Laufwerk 'D:' nach Abschluß der Ramkarteninitialisierung nicht mehr zu Verfügung.

Bei jedem Warmstart (control C) wird jetzt das System von der Ramkarte aus neu geladen, was naturgemäß wesentlich schneller vor sich geht, al ein Warmstart von einem mechanischen Floppy-Disk-Laufwerk.

Nicht vorgesehen ist dagegen ein 'Kaltstart' von der Ramkarte.

Der zur Aufnahme von Anwender-Dateien verbleibende Speicherplatz auf der Ramkarte ist um 12 kByte niedriger, als bei Initialisierung ohne 'W'-Parameter; dieser Sachverhalt spiegelt sich auch in den Eintragungen des Disk-Parameter-Blocks (DPB) der Ramkarte wieder. (Beim DPB handelt es sich um eine spezielle Datenstruktur, aus der das Betriebssystem bei der Dateiverwaltung alle benötigten Informationen über die Größe und Organisation des betreffenden Speichers entnehmen kann.)

Aus diesem Grund müssen Ramkarten, die mit dem 'W'-Parameter initialisiert worden sind, nach einem Neustart des Systems auch wieder mit 'ERAM Version C.0' unter Verwendung des 'W'-Parameters in das System eingebunden werden, andernfalls erscheint beim Ablauf der Initialisierungs-Software die Fehlermeldung 'Ramcard already in use'. Dies gilt sinngemäß natürlich auch für Ramkarten, die ursprünglich ohne 'W'-Parameter installiert worden sind.

Die in der Ramdisk gespeicherten Dateien werden durch mehrfaches Initialisieren nicht zerstört.
Diese Schutzfunktion basiert auf einer speziellen Eintragung im Inhaltsverzeichnis der Ramkarte ('ERAM.LBL'), die daher niemals gelöscht werden darf.
Da bei diesem Directory-Eintrag neben dem 'read only'-Attribut auch das 'system'-Attribut gesetzt ist, ist er bei Verwendung der üblichen 'Catalog'-Funktionen 'DIR' und 'CAT' nicht sichtbar.

6.2.2 * 'ERAM.COM' Version C.0, BIOS-Schnittstelle und DPB

Der Speicherraum von Ramkarten, die mit 'ERAM.COM' Version C.0 initialisiert worden sind, ist wie bei Floppy-Disk-Laufwerken in (logische) Spuren und Sektoren unterteilt.
Jede Spur umfaßt dabei 32 Sektoren mit einer Größe von 128 Bytes.
Diese Auslegung vereinfacht u.a. die Realisierung eines leicht zu integrierenden Warmstart-Laders ('W'-Parameter, vgl. 6.2.1).

Der zulässige Wertebereich für die Spur-Nummern richtet sich nach der Gesamtkapazität der jeweiligen Ramkarte.

Gesamtkapazität in kByte	64	128	192	256	512	768	1024
Wertebereich der Spurnummern	0..15	0..31	0..47	0..63	0..127	0..191	0..255

Bei Pseudo-Laufwerken ohne 'Systemspuren' findet man im Disk-Parameter-Block folgende Eintragungen:

Kapazität (kByte)	SPT	BSH	BLM	EXM	DSM	DRM	ALV	CKS	OFF
64	20 00	03	07	00	3F 00	1F 00	80 00	00 00	00 00
128	20 00	03	07	00	7F 00	3F 00	C0 00	00 00	00 00
192	20 00	03	07	00	BF 00	3F 00	C0 00	00 00	00 00
256	20 00	03	07	00	FF 00	5F 00	E0 00	00 00	00 00
512	20 00	04	0F	01	FF 00	7F 00	C0 00	00 00	00 00
768	20 00	05	1F	03	BF 00	FF 00	C0 00	00 00	00 00
1024	20 00	05	1F	03	FF 00	FF 00	C0 00	00 00	00 00

Bei Ramkarten, deren erste drei Spuren für eine Kopie des Betriessyste gebraucht werden, erhält man in den Positionen DSM und OFF entsprechen modifizierte Werte:

Kapazität (kBYte)	SPT	BSH	BLM	EXM	DSM	DRM	ALV	CKS	OFF
64	20 00	03	07	00	33 00	1F 00	80 00	00 00	03 00
128	20 00	03	07	00	73 00	3F 00	C0 00	00 00	03 00
192	20 00	03	07	00	B3 00	3F 00	C0 00	00 00	03 00
256	20 00	03	07	00	F3 00	5F 00	E0 00	00 00	03 00
512	20 00	04	0F	01	F9 00	7F 00	C0 00	00 00	03 00
768	20 00	05	1F	03	BC 00	FF 00	C0 00	00 00	03 00
1024	20 00	05	1F	03	FC 00	FF 00	C0 00	00 00	03 00

(alle Eintragungen in hexadezimaler Form, 'low byte' zuerst)

Man sieht, daß die Blockgrößen in Abhängigkeit von der Speicher-Kapazi tät der Ramkarte von 1 kByte bis 4 kByte ansteigen, worunter bei den großen Formaten die Speicherausnutzung für sehr kleine Dateien leidet. Einer Verbesserung dieser Situation steht aber vor allem der Speicher- platzbedarf des ALV-Vektors entgegen, dessen Unterbringung ohnehin sch Schwierigkeiten bereitet.

Die DPH-Vektoren der im System aktiven Laufwerke liegen nach Initiali- sierung der Ramkarte nicht mehr in jedem Fall in geordneter Reihenfolg im Speicher.

Bei den Systemen 2.20 und 2.23, wo die Z80-CPU uneingeschränkten Zugri auf den I/O-Adreßraum besitzt, ist der DPH der Ramkarte, ebenso wie DPB und ALV-Vektor im statischen RAM auf der Karte selbst untergebrach

In 2.26- bzw. 2.28- Systemen ist dies leider nicht möglich.
Hier werden ausschließlich die bereits zur Systemerweiterung vor- handenen DPHs verwendet. (DPB und ALV-Vektor müssen dagegen in der Use Patch-Area angelegt werden (vgl. 6.5).)

Dieser Sachverhalt spielt allerdings aus der Sicht des Anwenders (und Programmierers) keine Rolle, solange die DPH-Startadresse 'vorschrifts mäßig' durch einen BIOS-Aufruf der 'Select-Disk'-Funktion ermittelt wird (HL-Register !).

Die übrigen BIOS-Funktionen 'read', 'write', 'set track', 'set sector' und 'set DMA' arbeiten in Verbindung mit der Ramkarte in gleicher Weis wie bei Floppy-Disk-Laufwerken.

Bei der Schreib-Funktion hat die Angabe eines Parameters zur Unterscheidung zwischen 'immediate mode' und 'deferred mode' keine Auswirkung. Dies hängt damit zusammen, daß bei der Ramkarte keinerlei 'blocking'/'deblocking' Operationen stattfinden, die sonst zur Anpassung des CP/M Sektorformats von 128 Bytes an die jeweilige, physikalische Sektorgröße erforderlich sind.

6.2.3 * 'ERAM.COM' Version C.0, Speicherorganisation

Die Umrechnung der bei den BIOS-Funktionen 'read' bzw 'write' angeforderten Sektoren in die Bestandteile der Ramkarten-Gesamtadresse (vgl. Anhang H und S) geht bei der vorliegenden Treibersoftware so vor sich, daß die Datensektoren in linear geordneter Reihenfolge im I/O-Expansions-Adreßbereich erscheinen.
Dies bedeutet mit anderen Worten, daß je nach Auswahl der Parameter Spaltenadresse und Banknummer im Falle einer mit 64k DRAMs bestückten Ramkarte jeweils auf zwei aufeinanderfolgende Sektoren (256 Bytes) im Adreßbereich $C800..$C8FF direkt zugegriffen werden kann.
Bei einem Speicherausbau mit 256k DRAMs stehen jeweils acht Sektoren im Bereich $C800..CBFF zur Verfügung.
Man erhält somit im Ramkartenspeicher ein Datenfenster von 256 bzw. 1024 Byte Größe, dessen Inhalt beliebig gelesen oder verändert werden kann.

Für viele Anwendungsfälle ist dieser Speicherausschnitt groß genug, um auf den sonst üblichen Transfer der Ramkarten-Daten in einen Pufferspeicher vollständig verzichten zu können.
Neben der Einsparung eines Datenpuffers läßt sich dadurch, - zum Beispiel bei Such- oder Sortier-Vorgängen - ein beträchtlicher Zeitgewir erzielen.

Spaltenadresse und Banknummer erhält man bei Bestückung mit 64k DRAMs aus der Vorschrift:

```
                    Spaltenadresse : Sektor(1..4) + 16 * Spur(0..2
                        Banknummer : Spur(4..5)
Startadresse im I/O-Expansionsbereich : $C800 + 128 * Sektor(0)
```

Bei Ramkarten, die 256k DRAMs bestückt sind, werden durch die Vorgabe von Spaltenadresse und Banknummer acht Sektoren angewählt:

```
                    Spaltenadresse : Sektor(3..4) + 4 * Spur(0..5)
                    Banknummer     : Spur(6..7)
Startadresse im I/O-Expansionsbereich : $C800 + 128 * Sektor(0..2)
```

(Die in Klammer gesetzten Bereichsangaben kennzeichnen Bitpositionen bei binärer Darstellung der Spur- und Sektornummern)

6.3 Das Kopieren von CP/M-Dateien

Zum Kopieren von Dateien dürfen in Verbindung mit der Ramkarte alle Kopierprogramme verwendet werden, die den Datenverkehr über die BDOS-Schnittstelle des CP/M 2.2 Systems abwickeln.

Das wohl am weitesten verbreitete Kopier-Utility 'PIP.COM' ist ohne jede Einschränkung anwendbar.

Der Einsatz von Spur-orientierten Schnell-Kopierprogrammen, wie beispielsweise 'COPY.COM', ist dagegen generell nicht zulässig.

Dies ist jedoch im allgemeinen keine schwerwiegende Einschränkung, denn bei Beteiligung einer Ramdisk laufen auch umfangreiche Kopierarbeiten in relativ kurzer Zeit ab, selbst wenn dabei alle Dateien einzeln transferiert werden.

6.4 Automatische Ramkarten-Initialisierung

Häufig besteht der Wunsch, unmittelbar nach dem Kaltstart des Betriebssystems die Ramkarte zu initialisieren und anschließend mit einer Reihe von Dateien zu laden, die für die dann folgenden Arbeiten benötigt werden.

Zur Automatisierung solcher Vorgänge bietet sich bei CP/M die Verwendung einer SUBMIT-Datei an.
Es handelt sich dabei im Wesentlichen um eine Textdatei mit dem Filenamen-Zusatz '.SUB', die alle erforderlichen Befehle enthält.
Mit einem 'SUBMIT-Aufruf', der als Parameter den Namen der genannten Befehlsdatei erhält, werden dann unter Zwischenschaltung einer temporären Hilfsdatei ($$$.SUB) alle im Textfile enthaltenen Befehle vom CCP abgearbeitet (vgl. einschlägige CP/M System-Dokumentation).

Der 'SUBMIT'-Aufruf arbeitet auch dann korrekt, wenn bei der Ramkarten-Initialisierung der 'W'-Parameter aktiviert wurde (vgl. 6.2.1).

Dies ist nicht ganz selbstverständlich, denn der CCP erwartet die temporäre Hilfsdatei '$$$.SUB', die die nächstfolgende Kommandozeile enthält, stets im System-Laufwerk 'A:'. Sobald 'ERAM.COM' abgelaufen ist, ist aber die zu diesem Zeitpunkt normalerweise noch leere Ramkarte als System-Laufwerk 'A:' im System installiert; das '$$$.SUB'-File befindet sich dann also im (neuen) Drive 'B:' !.

Um an dieser Stelle einen Abbruch der Submit-Bearbeitung zu verhindern, enthält das Ramkarten-Initialisierungsprogramm eine spezielle Unterroutine, die am Ende der Initilaisierung überprüft, ob das neu umbenannte Laufwerk 'B:' eine Datei mit Namen '$$$.SUB' enthält.
Dies ist dann als sicheres Zeichen dafür anzusehen, daß ein SUBMIT-Aufruf in Aktion ist.

Um dessen korrekte Fortführung zu ermöglichen, wird das '$$$.SUB'-File zunächst von Drive 'B:' in die Ramkarte (neuer Drive A:) transferiert und daran anschließend in Drive 'B:' gelöscht.

Die CCP-Software findet nun nach dem auf die Ramkarten-Initialisierung folgenden Warmstart in Laufwerk 'A:' (Ramkarte) eine '$$$.SUB'-Datei und entnimmt das nächstfolgende Kommando folglich nicht der System-Console, sondern eben dieser Datei.

Die Multidiskette enthält eine Submit-Datei mit Namen 'ERAML.SUB', die zur Demonstration dieser Möglichkeiten die Ramkarte mit Hilfe eines 'ERAM W'-Befehls als Drive 'A:' installiert und daran anschließend unter Verwendung von 'PIP' sämtliche Dateien der Bootdiskette in die Ramkarte überträgt.

Wenn auch der Submit-Aufruf selbst nach dem Booten automatisch generiert werden soll, dann kann man eventuell das CP/M Utility-Programm 'AUTORUN' einsetzen, welches allerdings das Betriebssystem auf den Systemspuren der Boot-Diskette verändert.
(vgl. CP/M-Dokumentation)

Für den Fall, daß auch diese Funktion in Anspruch genommen wird, enthält das Initialisierungsprogramm 'ERAM.COM' Version C.0 eine Sicherungs-Einrichtung, die bei mehrfachem Aufruf von 'ERAM' aus einer Submit-Datei die weitere Ausführung des Submit-Aufrufs unterbindet. Dadürch wird verhindert, daß eventuell bereits modifizierte Dateien de Ramkarte überschrieben und damit zerstört werden, wenn diese Dateien in ihrer ursprünglichen Form unter gleichem Namen auch auf der Bootdiskette zu finden sind.

Eine solche Sicherung existiert allerdings nur, wenn die Initialisieru der Ramkarte mit aktivem 'W'-Parameter durchgeführt wurde (vgl. 6.2).

6.5 * Kompatibilität

Das Einbinden von Pseudo-Laufwerken in die Betriebssysteme 2.20, 2.23, 2.26 und 2.28 ist mit einigem Aufwand verbunden.
Wegen der sehr unterschiedlichen I/O-Struktur der System 2.20 und 2.23 gegenüber den neueren Ausführungen 2.26 und 2.28 sind hier jeweils völlig verschiedene Lösungen erforderlich.

Während man bei 2.20 und 2.23 sowohl die Treiber-Routinen, als auch alle notwendigen Tabellen und Vektoren für den neuen Drive im static RAM auf der Ramkarte selbst unterbringen kann, trifft dies bei 2.26 und 2.28 nur für einen Teil der Treiber-Routinen zu.
Der Grund hierfür liegt darin, daß bei den neueren Systemen der direkt Zugriff der Z80-CPU, wie er für die DPH-, DPB- und ALV-Vektoren unbedingt erforderlich ist, auf den lokalen Speicher der Z80-CPU begrenzt ist. Der Speicherraum des 6502-Prozessors ist dagegen, ebenso wie der gesamte I/O-Adressbereich des Apple II Computers, nur über spezielle Transfer-Routinen zugänglich.

Diese arbeiten beim 2.26-System (Microsoft Premium Softcard IIe) nach dem Prinzip der Bankumschaltung; bei den 2.28-Systemen (Microsoft Softcard II) erfolgt die Datenübertragung zwischen Z80-Speicher und 6502-Speicherraum byteweise mit handshake-Synchronisation.

Das zuletzt geannte Verfahren bedeutet natürlich einen besonders hohen Zusatzaufwand, sobald ein umfangreicher Datenaustauch erforderlich ist

In jedem Fall benötigt man für die Betriebssysteme CP/M 2.26 und CP/M 2.28 außer den eigentlichen Ramkarten-Treiberprogrammen selbst (6502-Code) noch spezielle Interface-Routinen auf der Z80-Seite, um eine korrekte Parameter- und Datenübertragung zu ermöglichen.

Es bleibt daher nichts anderes übrig, als hierzu einen Teil der 'User-Patch-Area' (vgl. CP/M Dokumentation der Fa. Microsoft) zu verwenden, wobei natürlich die Gefahr besteht, daß Konflikte mit anderer, nachträ lich installierter Treiber-Software auftreten.

Leider ist der zusätzliche Speicherbedarf, vor allem im Fall 2.28, rec beträchtlich; es dürfen daher bei diesem Betriebssystem zusammen mit d Ramkarte nur noch drei weitere (Floppy-Disk-) Laufwerke im System installiert sein.
Diese Beschränkung wurde eingeführt, um zumindest mit dem erweiterten Floppy-Disk-System der Firma erphi GmbH vollständige Kompatibilität zu erreichen.

Bei CP/M 2.28 wird in der 'User-Patch-Area' für Interface-Routinen und ALV-Vektor zusammen der Speicherbereich von FE00 bis FE56 (einschließlich) belegt.
Der ursprünglich für den ALV-Vektor vorgesehene Speicherbereich ist mi 18 Byte Länge für die Ramkarte nicht ausreichend und wird daher für de Disk-Parameter-Block (DPB) verwendet.
Ohne Bedeutung für Ramkarten ist dagegen der 'Checkbyte-Vektor', desse Speicherplatz aus diesem Grund zur Unterbringung von zwei Patch-Routin genutzt wird.

Beim CP/M 2.26 -System läuft dies alles in etwa nach dem gleichen Sche ab, wobei allerdings der Platzbedarf für die Interface-Routinen glücklicherweise deutlich geringer ist.
Für Interface-Routinen und ALV-Vektor wird hier in der 'User-Patch-Area' der Speicherraum von FE00 bis FE3A (einschließlich) benötigt.
Weitere 27 Byte sind erforderlich, wenn die Ramkarte als Laufwerk 'E:' oder 'F:' in das System eingebunden wird.

Der Zugang zu den Interface-Routinen ist durch 'Umleiten' der BIOS-vektoren 'read' und 'write' realisiert. Für eine korrekte Weiterleitun der BIOS-Aufrufe ist dabei auch dann gesorgt, wenn diese Vektoren bereits vor Aufruf des Initialisierungsprogramms modifiziert worden sind Dies gilt auch für die Softcard Betriebssysteme 2.20 und 2.23, mit dem Unterschied, daß dort keine speziellen Interface-Routinen erforderlich sind, sodaß die Vektoren in diesem Fall direkt in den I/O-Bereich zum statischen RAM der Ramkarte zeigen können, wo auch die Treiber-Programme zu finden sind.

Bei den Systemen 2.20 und 2.23 befindet sich, wie schon erwähnt, auch der DPH-Vektor der Ramkarte im static RAM, sodaß hier in jedem Fall ei 'Patch' der BIOS-Select-Disk-Funktion notwendig ist.
Die ebenfalls im static RAM befindliche Patch-Routine wird durch einen Z80-CALL-Befehl aufgerufen und greift bei Adresse DD7C bzw. FEA6 an.

Weitere Modifikationen der BIOS-Software sind erforderlich, wenn die Romkarte als System-Laufwerk 'A:' in das Betriebssystem eingegliedert wird.

Stillegung und Ersatz des ursprünglichen Warmstart-Laders ist dabei noch relativ einfach (2.20: Z80-Adresse DAD6, 2.23: Z80-Adresse FAC2). Bei den Systemen 2.26 und 2.28 geschieht dies im 6502-BIOS-Teil, sodaß zumindest keine weitere Interface-Routine benötigt wird (2.26: 6502-Adresse $D940, 2.28: 6502-Adrese $D937).

Unangenehmer ist dagegen, daß die Zuordnung der (physikalischen) Laufwerke zu den (logischen) Drive-Buchstaben verändert werden muß. Dies ist nicht ganz einfach, weil man bei den Originalsystemen 2.20 bi 2.28 leider versäumt hat, zu diesem Zweck eine besondere Tabelle anzulegen.

Im Fall der 2.20/2.23 Systeme behalten alle Disketten-Laufwerke ihre ursprünglichen DPH-Vektoren bei, während die Ramkarte generell einen separaten DPH zugewiesen bekommt, der im static RAM der Ramkarte liegt

Um eine Verschiebung der Zuordnung der Laufwerksbuchstaben zu erreiche ist daher sowohl bei der Select-Disk-Routine eine Betriebssystem-Modifikation notwendig (- die BIOS-Select-Disk-Funktion berechnet die Startadresse des DPH-Vektors für den jeweils spezifizierten Drive -), als auch ein Eingriff an derjenigen Stelle der BIOS-Disketten-Treiber-Programme, wo die Umrechnung der jeweiligen Laufwerks-Nummer in Slot- und Drive-Nummer des zugehörigen (physikalischen) Laufwerks stattfinde (2.20: Adresse DE27, 2.23: Adresse B2D5).

Die Korrektur der Select-Disk-Routine bedeutet dabei keinen Mehraufwan weil an dieser Stelle ohnehin ein Betriebssystem-Patch erforderlich is damit ggf. die im static RAM gelegene Adresse des Ramkarten-DPH-Vektor als Ergebnis-Parameter des Select-Disk-Aufrufs bereitsteht.

Bei den CP/M-Varianten 2.26 und 2.28 wird die neue Zuordnung der Laufwerksbuchstaben dadurch herbeigeführt, daß die DPH-Vektoren der bisher im System aktiven Laufwerke, (- sie liegen hintereinander, in aufsteigender Reihenfolge im Speicher -), um jeweils 16 Byte nach oben geschoben werden, sodaß ausreichender Platz für den neuen DPH-'A:'-Vektor de Romkarte entsteht.
Der DPH-Vektor des Laufwerks mit der höchsten Drive-Nummer belegt dann den Speicherplatz, den der DPH-Vektor der Ramkarte eingenommen hätte, wenn die Karte ohne 'W'-Parameter initialisiert worden wäre.

Man erkennt, daß Änderungen der Select-Disk-Funktion bei diesem Verfahren eigentlich nicht erforderlich sind.
Im vorliegenden Fall wurde aber dennoch eine geringfügige Korrektur eingefügt (Adresse FC02 für 2.26 und 2.28), um die Kompatibilität mit den Floppy-Disk-Controllern der Firma erphi GmbH beizubehalten.

Eine Modifikation der Umrechnung von Laufwerksnummer in Slot- und Drive-Nummer bei den Disketten-Treiberroutinen ist natürlich auch hier unumgänglich. Die entsprechende Patchroutine greift sowohl beim System 2.26, als auch bei 2.28 bei Adresse FCBO an.

Zusammenfassend kann man sagen, daß Kompatibilitäts-Probleme nicht auszuschließen sind, wenn außer der Ramkarte selbst noch weitere Speichersysteme nachträglich in das jeweilige CP/M-System eingefügt werden. Dies gilt ganz besonders auch für Hard-Disk-Laufwerke.

Mit Hilfe der vorangegangenen Informationen sind sachkundige Ramkartenbesitzer im einen oder anderen Fall sicherlich in der Lage, solche Unverträglichkeiten zu beseitigen.

Weitere Schwierigkeiten können eventuell bei den System Versionen 2.26 und 2.28 durch den relativ hohen Speicherplatzbedarf in der 'User-Patc Area' auftreten.
Hierbei ist allerdings zu bemerken, daß lediglich auf der Z80-Seite ei gewisser Engpaß besteht, denn im Bereich der 6502-CPU bieten gerade diese Betriebssysteme noch beträchtliche Erweiterungsmöglichkeiten.

Anhang H * Hardware-Beschreibung und Speichererweiterung

1. Hardware-Beschreibung

Wie dem vereinfachten Blockschaltbild zu entnehmen ist, besteht die Ra karte im Wesentlichen aus den Schaltungsteilen Steuerlogik, Befehls- register, Register für Spaltenadresse, Refreshzähler, Multiplexer static RAM und natürlich der eigentlichen Speichermatrix.

Je nach Bestückung der Karte wird der Ramkarten-Inhalt in Form von 256 Byte (64k DRAMS) oder 1024 Byte (256k DRAMs) großen Segmenten im I/O-Expansionsbereich des Rechners abgebildet ($C800..$C8FF, bzw. $C800..$CBFF).
Das jeweils gewünschte Speichersegment wird dabei durch entsprechende Vorbereitung von Befehlsregister und Spaltenadressregister selektiert.

Beim Befehlsregister handelt es sich um ein 4 bit breites Steuer-Regis ter, dessen Inhalt die jeweiligen Betriebsbedingungen der Ramkarte festlegt.

Das Befehlsregister wird durch einen Schreib-Befehl auf eine der Devic Adressen angesprochen ($C08X+16*s, s=Slotnummer der Ramkarte), wobei der Zustand der unteren 4 Adressbits A0..A3 direkt in das Befehls- register übertragen wird.
Das bei diesem Schreibbefehl auf dem Datenbus erscheinende 8-Bit-Wort wird in das Register für die Spaltenadresse eingeschrieben.
Es ist also nur ein einziger Prozessorbefehl erforderlich, um den Inha von zwei Steuer-Registern festzulegen.

Nach einem System-Reset, oder auch nach Auftreten der Adresse $CFFF auf dem Adressbus des Rechners, sind alle Ausgänge des Befehlsregisters gelöscht.

Der niederwertigste Ausgang (Bit 0) des Befehlsregisters ist als eine Art Ein/Aus-Schalter zu betrachten.
Nur wenn dieser Ausgang aktiv ist (logisch 1), wird das aktuelle Segmer der Speichermatrix in den I/O Expansionsbereich eingeblendet und so ei Zugriff auf den Inhalt des Speichers ermöglicht.
Bevor man dieses Befehlsregrister-Bit setzt, ist daher unbedingt die I/O-Adresse $CFFF anzusprechen, damit es zu keiner Mehrfach-Belegung des I/O-Expansions-Adreßbereiches kommen kann
(vgl. Apple II Reference Manual).

Durch den Zustand der Befehlsregister-Bits 2 und 3 wird eine der maxim vier Speicherbänke der Matrix ausgewählt.
Je nach Bestückung der Karte umfaßt jede Speicherbank 64 kByte oder 256 kByte.

Auf der Ramkarte befindet sich außerdem ein separates statisches RAM, welches speziell zur Unterbringung von Treiber-Software vorgesehen wurde. Dieses RAM ist in zwei Speicherbänke unterteilt, die mit dem Befehlsregister-Ausgang Bit 1 umgeschaltet werden.

Jeweils eine der Speicherbänke erscheint im Adreßbereich $CsOO bis $CsFF (s=Slotnummer der Ramkarte).
Abgesehen von der Bankumschaltung ist der Zugriff auf das static RAM vom aktuellen Zustand des Befehlsregisters unabhängig.

Ein Schreib/Lese-Speicher hat an dieser Stelle gegenüber den sonst verwendeten Firmware-EPROMs eine Reihe von Vorteilen:

- Für jeden Anwendungsfall (jedes Betriebssystem) steht der gesamte static-RAM-Speicher zu Verfügung. Der RAM-Inhalt kann jederzeit veränderten Betriebsbedingungen angepaßt werden.
  Zur Einführung neuer Software-Versionen müssen keine Hardware-eingriffe vorgenommen werden (kein EPROM-Austausch).

- Im Gegensatz zu EPROM-Speichern können variable Größen der Treiber Software ebenfalls im static RAM untergebracht werden, sodaß kein zusätzlicher Speicherplatz gefunden werden muß
  (Keine Belegung von Zero-Page- oder Screenhole- Speicherplätzen)

- Im static RAM können auch selbstmodifizierende Treiber-Programme abgelegt werden. Die Software kann dadurch oftmals kürzer und effizienter gestaltet werden. Indirekte Adressierung beim 6502-Prozessor läßt sich beispielsweise damit auch ohne Verwendung von Zero-Page-Speicherplätzen realisieren.

Ungünstig ist dagegen, daß das statische RAM nach dem Einschalten des Rechners erst mit den benötigten Daten (Treiberprogrammen) geladen werden muß.
Dieser Nachteil ist jedoch bei Ramkarten normalerweise nicht so gravierend, denn im allgemeinen ist hier ohnehin ein besonderer Initialisierungsschritt erforderlich, in dem die Ramkarte selbst mit den benötigten Dateien geladen werden muß.

Man erkennt, daß nicht nur der gesamte Datenverkehr mit der Ramkarte, sondern auch der Zugriff auf die zugehörige Treiber-Software ausschließlich über den I/O-Adress-Bereich des Rechners abläuft.
Diese Auslegung hat gegenüber vielen anderen Konzepten den Vorteil, daß keine Probleme mit der Bankumschaltung einer Language-Card, oder der Bankumschaltung des Apple IIe Computers auftreten können.

2. Bestückung und Speichererweiterung

In der Speichermatrix kommen gewöhnliche, dynamic-RAM-Bausteine mit einer Organisation von entweder '64k x 1', oder '256k x 1' zum Einsatz. Die Refreshperiode kann 2 ms oder auch 4 ms betragen.

Zur Festlegung des jeweiligen IC-Typs gibt es auf der Ramkarte keinerl Schalter oder Drahtbrücken. Der jeweilige Speicherausbau wird vielmehr von der Software erkannt und entsprechend berücksichtigt.
(Eine gemischte Bestückung der Karte mit 64k- und 256k-Bausteinen ist zwar hinsichtlich der Hardware im Prinzip möglich, dürfte aber im Normalfall die Anpassungsfähigkeit der Initialisierungs- und Treiber-software weit überfordern und ist daher nicht zulässig.)

Es können grundsätzlich nur vollständige Speicherbänke bestückt oder nachgerüstet werden, wozu jeweils acht Speicher-ICs erforderlich sind.

Bei teilweiser Bestückung der Karte ist zu beachten, daß die einzelnen Speicherbänke in aufsteigender Reihenfolge zu besetzen sind.

Die exakte Lage der einzelnen Speicherbänke geht aus der beiliegenden Zeichnung hervor.

Das Einsetzen der Speicherbausteine sollte nur von sachkundigen Person unter Beachtung der folgenden Regeln vorgenommen werden:

- Intergrierte Schaltkreise dürfen niemals eingesetzt oder entfernt werden, wenn sich die Ramkarte im Periphierslot des Rechners befindet.
  Vor dem Herausnehmen und Wiedereinsetzen der Karte muß der Rechner unter allen Umständen ausgeschaltet werden.

- Beim Umgang mit den Speicher-ICs ist zu beachten, daß diese Bausteine empfindlich gegenüber statischer Aufladung sind.
  Sie sollten daher bis zu ihrem Gebrauch in leitendem Schaumstoff aufbewahrt werden.

- Beim Einsetzen der Speicher-Bausteine in die IC-Fassungen muß unbedingt die korrekte Orientierung der ICs eingehalten werden.
  (Markierung an IC-Gehäuse und -Fassung !)
  Falsch herum eingestzte ICs überleben den ersten Einschaltversuch normalerweise nicht und können weitere Schäden hervorrufen.

- Kontrollieren Sie bitte sorgfältig, ob wirklich alle IC-Anschlüsse einwandfreien Kontakt mit der jeweiligen Fassung haben

- Prüfen Sie bitte vor dem Wiedereinsetzen der Karte, ob während der Bestückungsarbeit irgendwelche Verunreinigungen an die Karte gekommen sind, die eventuell zu Kurzschlüssen führen könnten.
  Im Zweifelsfall sollte man die Ramkarte vor der Inbetriebnahme noch einmal reinigen.

- Vergessen Sie bitte nicht, daß die Karte einen höheren Stromverbrauch hat und mehr Verlustwärme erzeugt, wenn weitere Speicherbänke hinzugefügt worden sind.
  Nur wenn Stromversorgung und Ventilation Ihres Rechners den erhöht Anforderungen gewachsen sind, ist ein zuverlässiger Betrieb der Ramkarte möglich

Anhang S * Richtlinien für den Entwurf von Treibersoftware

Beim Entwurf von I/O-Treiber-Software für die Ramkarte ist es selbstverständlich von entscheidender Bedeutung, ob der Speicherraum der Karte in logische Teilbereiche wie Blöcke, Spuren und Sektoren unterteilt werden soll.
Dabei wird man von Fall zu Fall zu ganz unterschiedlichen Lösungen kommen.

Um den Software-Zugriff zur Ramkarte an einem möglichst allgemeingültigen Beispiel demonstrieren zu können, soll die Ramkarte in den folgenden Ausführungen als ein ungeteilter Speicher mit 8 Bit Wortbrei angesehen werden.
Um den Inhalt eines solchen Speichers (wahlfrei) ansprechen zu können, ist folglich, je nach Speicherkapazität der jeweiligen Karte, eine bis zu 20 Bit breite Adresse erforderlich, die hier als 'Gesamtadresse' bezeichnet wird.

Aufgabe der Treiber-Software ist es nun unter anderem, die Gesamtadresse in mehrere Teile zu zerlegen, sodaß ein Offset in den I/O-Expansions-Adreßbereich, ein 8-Bit-Wort für das Spalten-Adreßregister, sowie ein 2-Bit-Wort zu Auswahl der Speicherbank entstehen.
Wegen der unterschiedlichen Segment-Größe von 256 bzw. 1024 Byte (vgl. Anhang H, 1.) ist die Art der Aufteilung der Gesamtadresse davon abhängig, ob die Ramkarte mit 64k- oder mit 256k-Speicherbausteinen bestückt ist.

Bei Bestückung mit 64k-DRAMs erhält man z.B. folgende Zuordnungen:

```
Adresse im I/O-Expansionsbereich (IOA) :  Gesamtadresse( 0.. 7) + $C8
                  Spaltenadresse (SPA) :  Gesamtadresse( 8..15)
                      Banknummer (BNR) :  Gesamtadresse(16..17)
```

Für Ramkarten, die mit 256k DRAMs ausgerüstet sind erhält man:

```
Adresse im I/O-Expansionsbereich (IOA) :  Gesamtadresse( 0.. 9) + $C8
                  Spaltenadresse (IOA) :  Gesamtadresse(10..17)
                      Banknummer (BNR) :  Gesamtadresse(18..19)
```

(Die eingeklammerten Bereichsangaben kennzeichnen Bitpositionen bei binärer Darstellung der Gesamtadresse)

Geht man einmal davon aus, daß die Bestandteile der Gesamtadresse bereits durch geeignetes Verschieben und Maskieren errmittelt worden sind und in Variablen mit den Namen IOA, SPA und BNR abgespeichert sin dann verläuft der Zugriff auf die Ramkarte wie in den folgenden Programm-Beispielen dargestellt:

Datenbyte von der Ramkarte lesen:

```
        BIT   $CFFF     ; I/O-Expansionsbereich freischalten

        LDA   BNR       ; Banknummer zur
        ASL             ; Befehlsregisteradressierung
        ASL             ; in Bitpositionen 2 und 3 schieben
        TAY

        LDA   SPA       ; Spaltenadresse in Akku
        STA   $COD1,Y   ; Spaltenadresse in Device-Adresse
                        ; $C080 + 16 * Slotnummer, z.B. Slot 5
                        ;       +  4 * Banknummer
                        ;       +  1   (Einschalten der Ramkarte)
                        ; (Befehlsregisterbits 0, 2 und 3 setzen)
        LDY   #0
        LDA   (IOA),Y   ; Byte von der Ramkarte lesen

        STY   $CODO     ; Ramkarte ausschalten,
                        ; I/O-Expansionsbereich freigeben
                        ; (Schreibbefehl auf Adresse
                        ;  $C080 + 16 * Slotnummer + 0,
                        ;  alle Befehlsregister-Bits löschen)

        RTS             ; Ende, Ramkarten-Byte in Akku
```

schreiben, Datenbyte in Akku:

```
        BIT   $CFFF     ; I/O-Expansionsbereich freischalten

        PHA             ; Datenbyte vorübergehend auf Stack retten

        LDA   BNR       ; Banknummer zur
        ASL             ; Befehlsregisteradressierung
        ASL             ; in Bitpositionen 2 und 3 schieben
        TAY

        LDA   SPA       ; Spaltenadresse in Akku
        STA   $COD1,Y   ; Saltenadresse in Device-Adresse
                        ; $C080 + 16 * Slotnummer, z.B. Slot 5
                        ;       +  4 * Banknummer
                        ;       +  1   (Einschalten der Ramkarte)
                        ; (Befehlsregisterbits 0, 2 und 3 setzen)
        LDY   #0
        PLA             ; Datenbyte vom Stack holen
        STA   (IOA),Y   ; Datenbyte in die Ramkarte schreiben

        STY   $CODO     ; Ramkarte ausschalten,
                        ; I/O-Expansionsbereich freigeben
                        ; (Schreibbefehl auf Adresse
                        ;  $C080 + 16 * Slotnummer + 0,
                        ;  alle Befehlsregister-Bits löschen)

        RTS             ; Ende
```

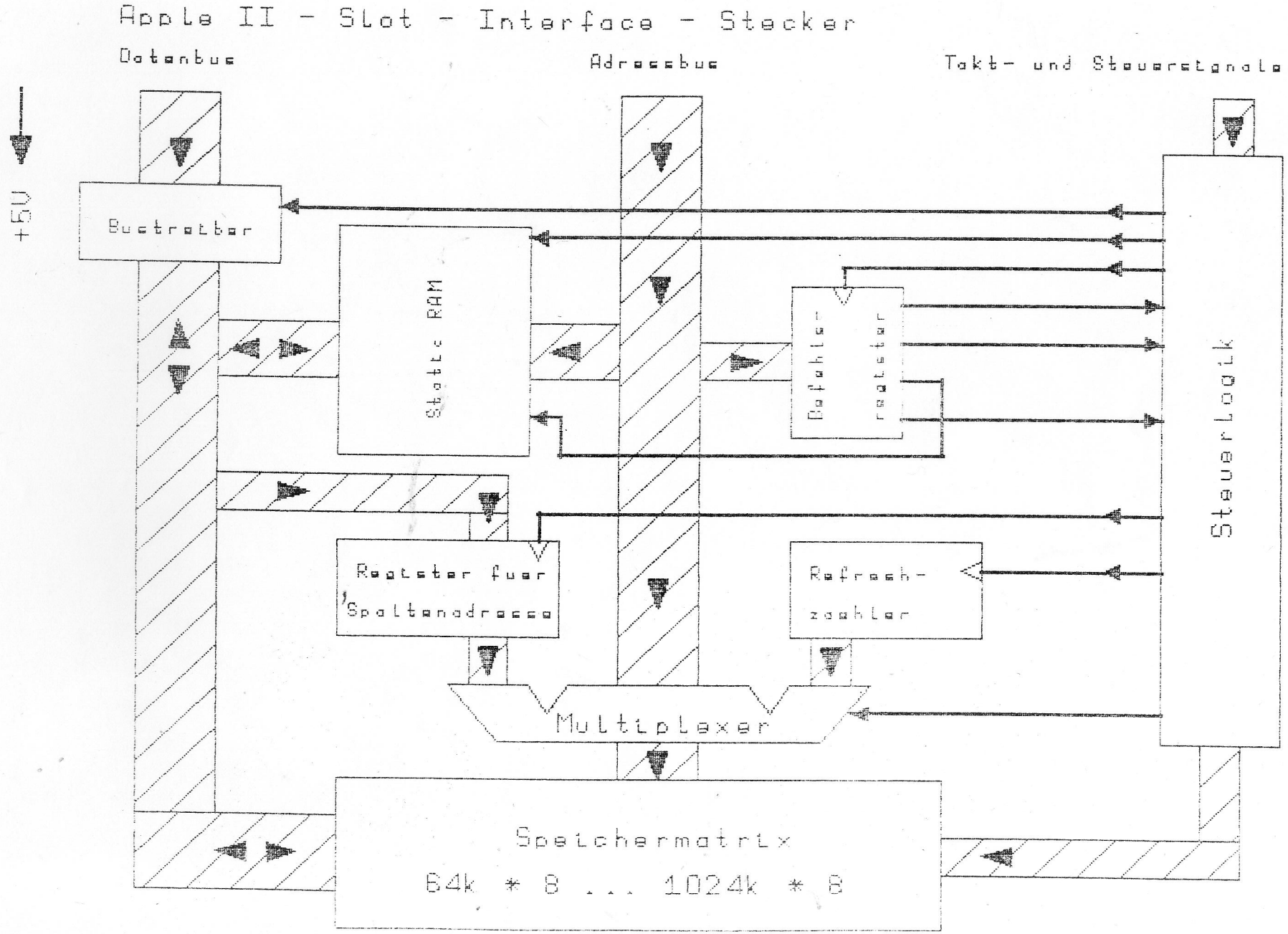
In den meisten Anwendungsfällen wird man die Datenbytes jedoch nicht einzeln übertragen, sondern vielmehr die Tatsache nutzen, daß ein uneingeschränkter Zugriff auf ein Speichersegment von 256 bzw. 1024 Byt Länge möglich ist, sobald Spaltenadresse und Banknummer selektiert worden sind.

Allerdings geht diese Einstellung verloren, sobald die Adresse $CFFF auf dem Adreßbus des Rechners erscheint (Befehlsregister wird gelöscht. vgl. Anhang H, 1.).

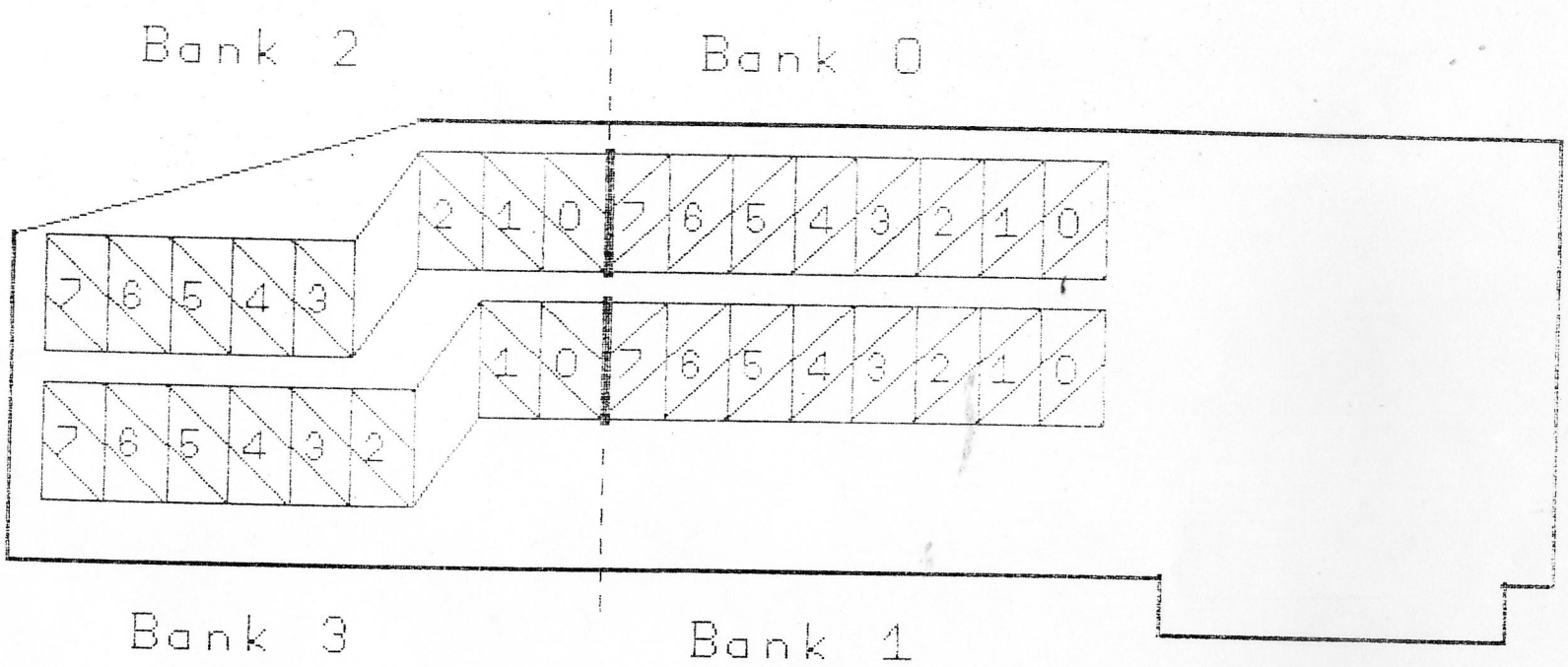
Diesem Sachverhalt muß unter Umständen besondere Aufmerksamkeit gewidmet werden, wenn im aktuellen Rechnersystem aktive Interruptquellen vorhanden sind.
In diesem Fall ist zu prüfen, ob die Interrupt-Handler selbst auch den I/O-Expansionsbereich benutzen, oder aber Peripheriegeräte ansprechen, die wiederum ihrerseits diesen Speicherbereich benötigen.
Wo dies nicht auszuschließen ist, muß man die Interruptverarbeitung für den Zeitraum der Datenübertragung aus der Ramkarte mit Hilfe eines 6502-'SEI'-Befehls sperren.

Bei den Initialisierungsprogrammen, die auf der Multidiskette zu finden sind, ist diese Maßnahme bereits vorsichtshalber getroffen worden.
Für manche Anwendungsfälle wird dadurch aber die Interrupt-Reaktionsze unnötig verschlechtert. Da die Ramkarten-Treiber-Software im static RAM untergebracht ist, kann man diese Situation durch Überschreiben des jeweiligen 'SEI'-Befehls mit einen 'NOP'-Opcode leicht bereinigen.

Werden beide Speicherbänke des statischen RAMs ausgenuzt, dann ist auch in diesem Zusammenhang zu beachten, daß nach jedem 'reset' oder '$CFFF-Zugriff' Bank 0 eingeblendet ist.
Man legt die Treibersoftware daher am besten so aus, daß sowohl sämtliche Einsprungstellen, als auch das Ende der Treiber-Routinen in der Bank 0 des statischen RAMs liegen.
Während des Ablaufs von Programmteilen, die sich in Bank 1 befinden, ist die Interruptverarbeitung gegebenenfalls vorübergehend zu sperren.



Vereinfachtes Blockschaltbild der Ramkarte


Bank 2
Bank 0
2 1 0 7 6 5 4 3 2 1 0
7 6 5 4 3
1 0 7 6 5 4 3 2 1 0
7 6 5 4 3 2
Bank 3
Bank 1