

Dein GIIIs, die 4 unbekanntes Wesen

A: Hardware-Erweiterung für 1 MB RAM

Die 128 Refreshzyklen des Z80 lassen sich auf 256 Zyklen erweitern, was die Verwendung von 41256-RAMs ermöglicht (Abb. 3). Dadurch kann jeder Z80-Computer auf mindestens das vierfache Speichervolumen aufgebohrt werden, indem man die vorhandenen 4164-RAMs durch 41256-Typen ersetzt. Die beiden zusätzlich nötigen Adressen kann ein I/O-adressiertes Latch liefern (Abb. 1). Damit kann man zwischen vier Blocks mit je 64 kB umschalten.

Dabei würden dann aber der Stack, Interrupt-Serviceroutinen und alles an Software abgekniffen werden, was für eine einwandfreie Funktion nötig ist. Der Z80 würde sich in der neuen Bank, wo noch kein Programm enthalten ist, verlaufen. Deshalb muß ein Teilbereich des Speichers auch beim Umschalten der Bank erhalten bleiben, wo der Z80 alles Lebenswichtige vorfindet.

Im Genie III s ist das mit vier Banks in Form von vier Reihen 4164-RAMs so gelöst, daß mit dem Zustand von Bit 0 des Ports F9h festgelegt wird, ob der Adreßbereich 0000h-3FFFh oder der Bereich E000h-FFFFh nicht mit umgeschaltet wird (Common-Bereich):

Port F9h, Bit 0: 0: Common = 0000h-3FFFh
 1: Common = E000h-FFFFh

Die Auswahl unter den vier Banks geschieht mit den Bits 6 und 7 desselben Ports:

Port F9h, Bits 6, 7: 00: Bank 0
 01: Bank 1
 10: Bank 2
 11: Bank 3

Unabhängig von der eingeschalteten Bank wird bei Zugriff auf den als Common definierten Bereich immer auf den RAM-Block 0 zurückgeschaltet. Das wird durch entsprechende Steuerung der CAS*-Freigabesignale für die vier RAM-Reihen erreicht (Abb. 4).

Zur Vorgabe der Adressen A16 und A17 für 41256-RAMs anstelle der ursprünglichen 4164-RAMs wurde ein unter der Portadresse F3h decodiertes Latch 74LS273 herangezogen, dessen Ausgänge Q0 und Q1 diese Adressen liefern. Damit läßt sich innerhalb eines 256-kB-Blocks (8 Stück 41256-RAMs) vorgeben, welches Viertel der RAM-Bausteine ansprechbar sein soll. Die RAM-Reihe selbst wird weiterhin über Port F9h, Bits 6 und 7 selektiert. Das Zusammenspiel beider Ports bei der Adressierung jeweils eines Blocks zu 64 kB veranschaulicht die Abb. 6.

Das Umschalten auf die RAM-Reihe 0 beim Zugriff auf den als Common definierten Adreßbereich (Port F9h, Bit 0) leistet eine entsprechende Schaltung im GIIIs. Die Common-Bedingung wird auf Wunsch (s. u.) auch bei der Umschaltung innerhalb der Viertel der 41256-Chips beachtet: Die Selektion des unteren Viertels einer Reihe 41256-RAMs wird über die Steuerung des Pins 1 (Select) des 74LS157-Multiplexers in der abgebildeten Schaltung erreicht. Das wäre prinzipiell durch die AND-Verknüpfung der beiden low aktiven Signale COML* und COMH* (U84, 74LS00, Pins 12 und 13) möglich. Wenn eines dieser Signale low ist, wird über low am SEL-Eingang des Multiplexers GND auf die Ausgänge durchgeschaltet und nicht die Bits Q0 und Q1 des Latch. COML* und COMH* werden low, also aktiv, wenn der unten bzw. oben im 64-kB-Adreßraum als Common definierte Speicherbereich angesprochen wird.

Hier wurde aber zusätzlich die Möglichkeit geschaffen, über Q7 des Latch (Port F3h, Bit 7) die Steuerung des SEL-Eingangs des Multiplexers

zu unterbinden. Wenn D7 des Ports F3h high ausgegeben ist (Bit im Output gesetzt), wird die automatische Umschaltung auf das untere Viertel einer 256K-RAM-Reihe unterbunden. Dann ist nur noch die ursprüngliche Steuerung der CAS*-Freigabesignale vom Zugriff auf den festgelegten Common-Bereich belangt (Selektion einer RAM-Reihe, aber nicht des unteren Viertels).

Das Verhalten des Computers läßt sich mit Q7=1 aus dem Normalverhalten (nur 1 aktiver Common-Bereich, 16 Banks) in einen zweiten Zustand umschalten: Es gibt dann 6 Blocks zu je 256 kB (Port F3h, Bits 0 und 1) mit je einem eigenen Common-Bereich an der durch Port F9h, Bit 0 definierten Adresse. Jeder dieser Blocks kann unabhängig hiervon je nach Port F9, Bits 6 und 7 eine der vier RAM-Reihen benutzen. Das entspricht der parallelen Existenz von vier Computern mit je vier Banks zu 64 kB.

In anderen Worten: Wahlweise stellt die hier vorgestellte Schaltung einem Computer 1 MB RAM zur Verfügung, oder vier voneinander unabhängige Computer verwalten je 256 kB RAM. Die zweite Variante entspricht dem Originalzustand des GIIIs - mal vier. Das Nebeneinander der nunmehr 16 Blocks zu je 64 kB ist in Abb. 5 b und c dargestellt.

Da das verwendete Latch 74LS273 aber 8 Bits verfügbar hat, sollte damit weiteres Tune-Up verwirklicht werden.

Der Bildwiederholtspeicher und der Zeichengenerator sind 6116LP15-RAMs. Diese Einheiten können unter Verwendung von 6264LP15-Speichern ebenfalls vervierfacht werden, wenn deren zusätzliche Adreßleitungen A11 und A12 mit den Ausgängen 02 und 03 des Latch beschaltet werden. Über D2 und D3 am Port F3h läßt sich einer von vier Bildwiederholtspeichern mit eigenem Zeichengenerator-RAM selektieren.

Und schließlich sollte auch noch der HRG-Speicher von 64 auf 256 kB durch Austausch der 4164-RAMs gegen 41256-Typen auf acht Pages aufgebohrt werden. Die Selektion eines Viertels des neuen Speichers sollte über D4 und D5 am Port F3h erfolgen. Innerhalb des adressierten Viertels könnte dann weiterhin wie bisher eine von zwei Pages zu 32 kB gewählt werden. Die alte Software würde dadurch in ihrer Funktionsfähigkeit nicht beeinträchtigt werden.

Aus nicht verständlichen Gründen funktionierte das Umschalten bei einem Versuch nicht, Q4 und Q5 als Adressen A16 und A17 über einen Multiplexer 74LS157 (auch 74HC157) auf die neuen 41256-RAMs des HRG-Speichers zu legen. Sein Pin 1 wurde durch das gleiche Signal gesteuert wie die beiden schon existierenden Multiplexer. Evtl, kann die Verwendung eines S- anstatt des LS- bzw. HC-Typs das Problem lösen. Vielleicht kann uns jemand mit einem Tip weiterhelfen?

Auch eine andere Funktion, die nach Schaltplan einwandfrei arbeiten sollte, versagte den Dienst. Durch NAND-Verknüpfung von Q6 des Latch und MWR* sollte erreicht werden, daß bei high gesetztem Bit 6 im Port F3h beim Lesen des Speichers auf Computer 0 geschaltet wird, und nur beim Schreiben der RAM-Speicher des mit D0 und D1 festgelegten Computers angesprochen wird. Damit wäre es besonders einfach geworden, den Speicherinhalt von Computer Nr. 0 auf seine drei Nachbarn zu übertragen, um sie mit Initial-Information zu laden. Das hätte der Fähigkeit des EG 64 MBA entsprochen, mit einem einfachen LDIR den ROM-Inhalt auf das parallele RAM zu kopieren. Auch dieses Feature funktioniert wider Erwarten nicht.

Im Layout wurden trotzdem die Schaltungen dieser nicht funktionierenden Elemente beibehalten. Evtl, gelingt es noch, diese Features später zum Laufen zu kriegen. Ansonsten steht einer anderweitigen Verwendung der Latch-Ausgänge Q4, Q5 und Q6 nichts im Wege. Die im folgenden beschriebene Software kommt ohne diese Funktionen aus. Die nicht funktionsfähigen Teile der Schaltung haben insb. auch keinen negativen Effekt auf sie, so daß sie im Hinblick auf späteres Gelingen durchaus bereits bestückt wer-

den können.

Die drei Gruppen zu je zwei Bits am Port F3h sind voneinander unabhängig ansprechbar. So kann z.B. über D2 und D3 zwischen mehreren Bildschirmen und Zeichensätzen gewählt werden, D4 und D5 können im Graphikspeicher einen Spooler mit luxuriöser Kapazität zur Verfügung stellen, während D0 und D1 ständig low bleiben, also gewissermaßen im Computer Nr. 0 gearbeitet wird. Und Computer Nr. 3 kann eine Graphik ändern, die Computer Nr. 1 erstellte usw..

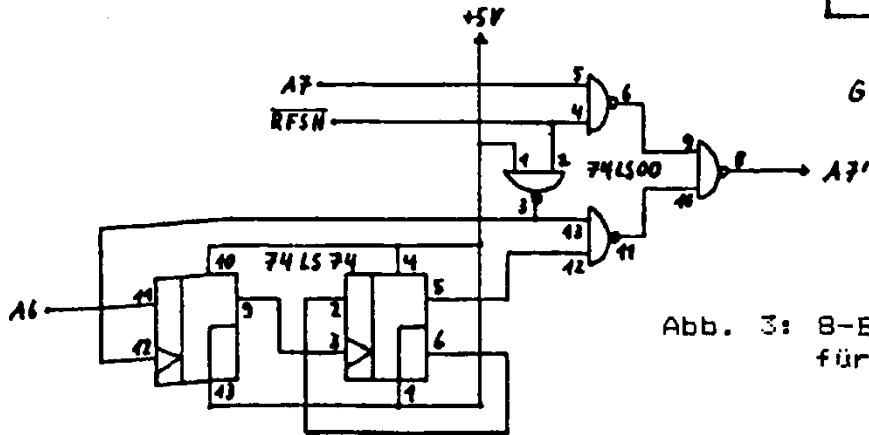
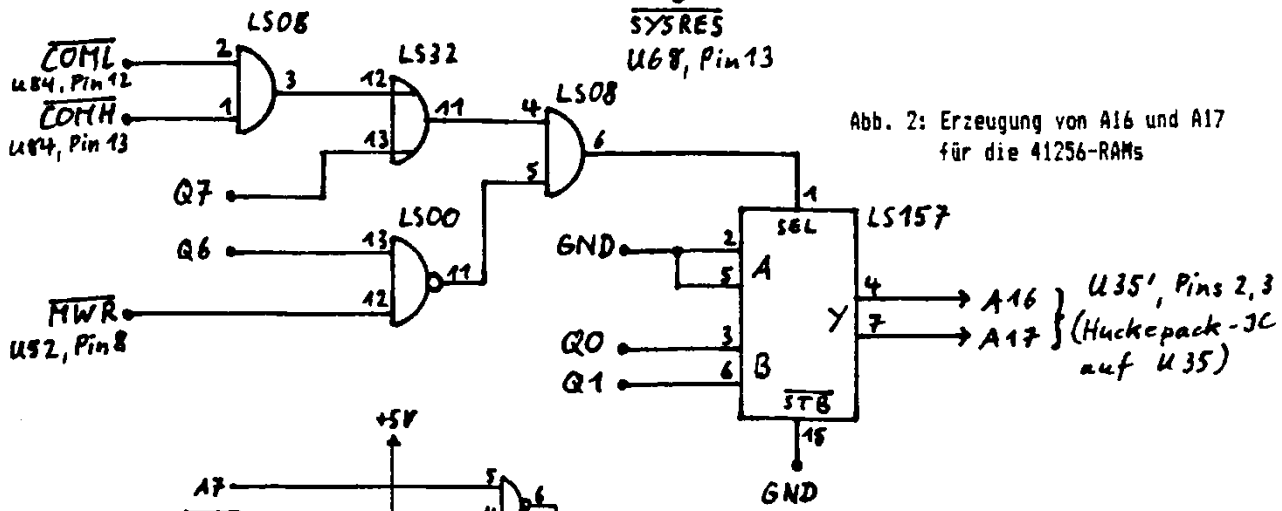
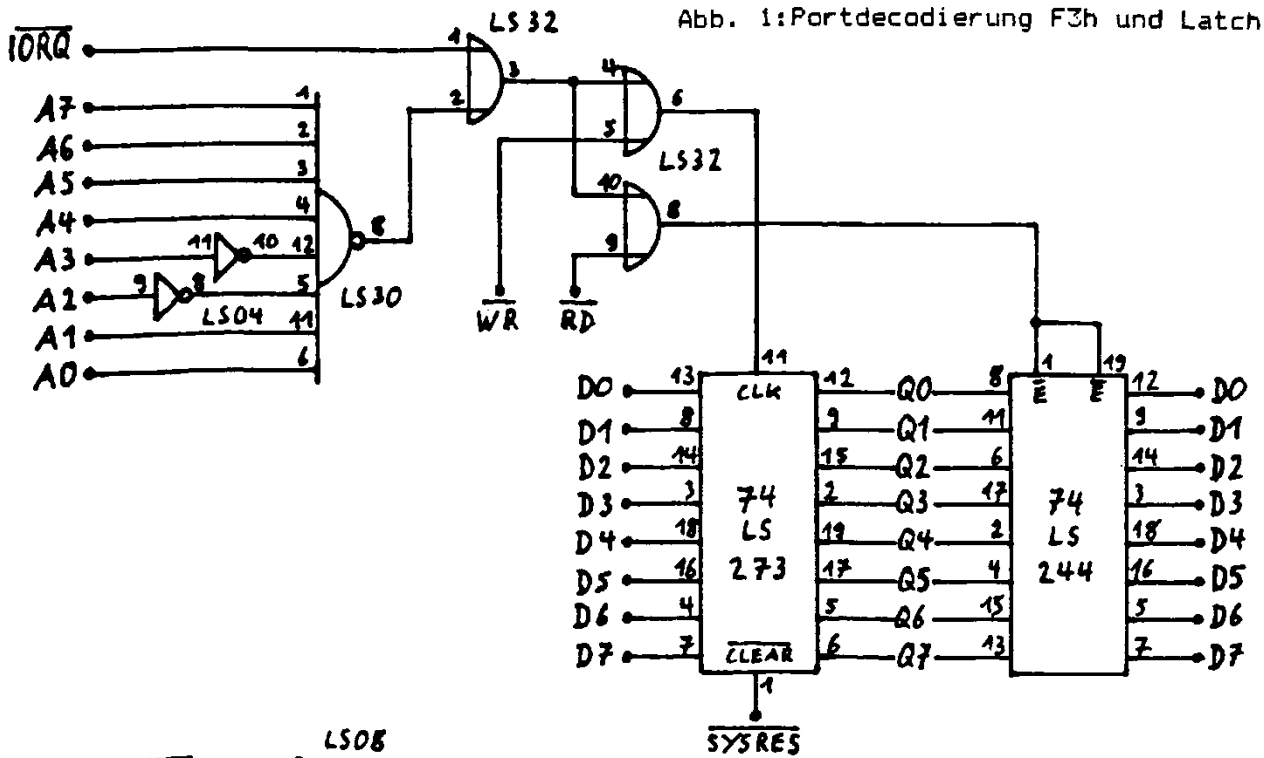


Abb. 4: CAS*-Freigabesignale der 4 64-kB-Blocks

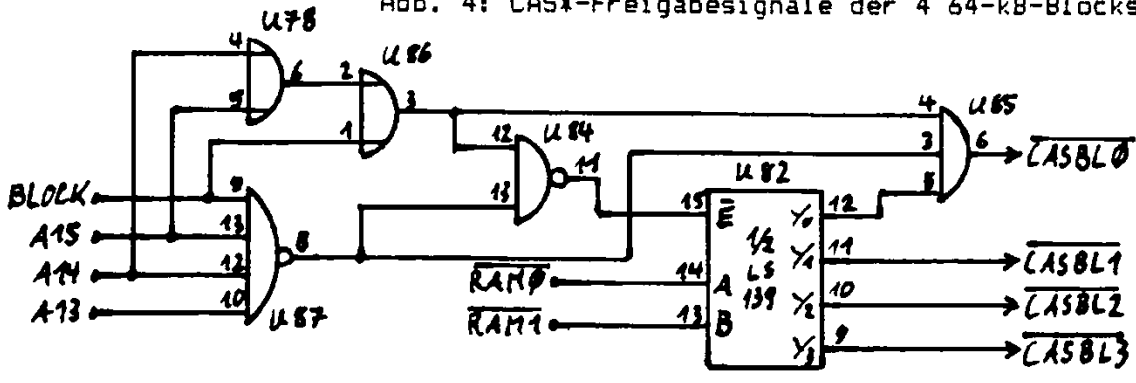
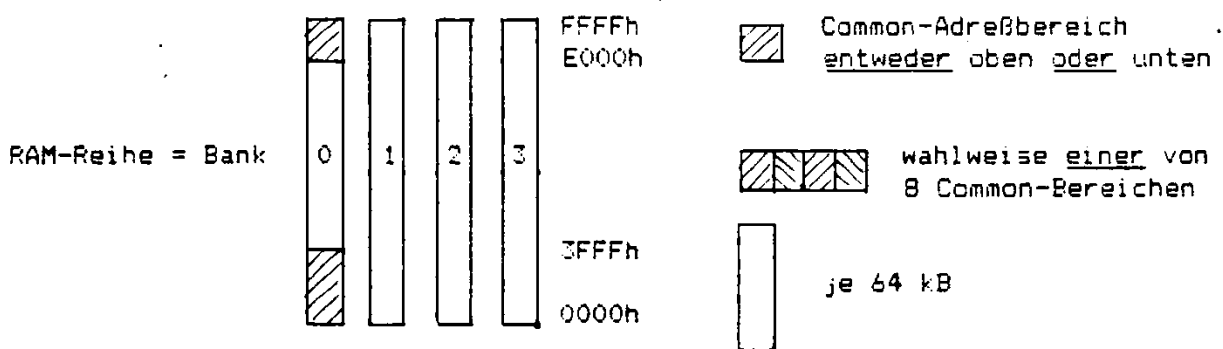
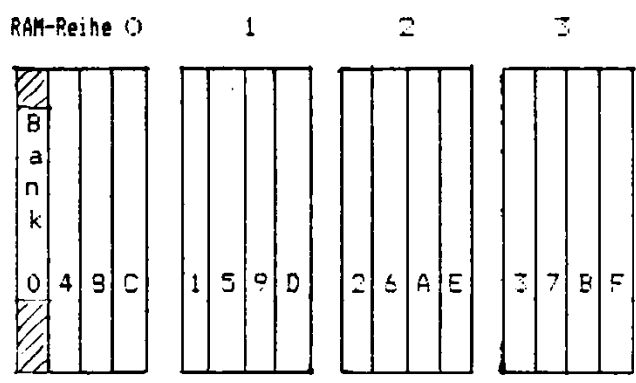


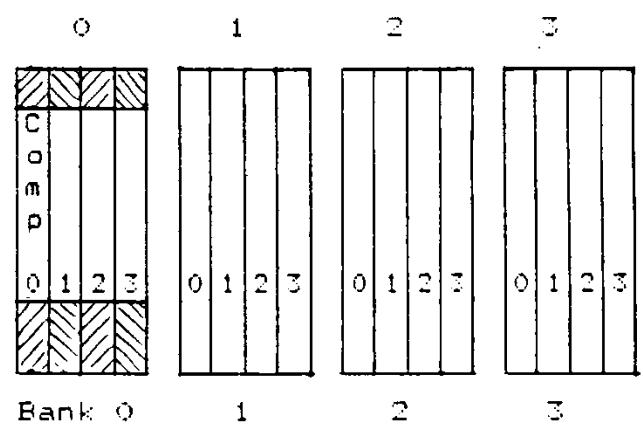
Abb. 5: Die Speicherkonfiguration des Genie III s



a) im Originalzustand



b) bei 1 Computer mit 1 MB RAM

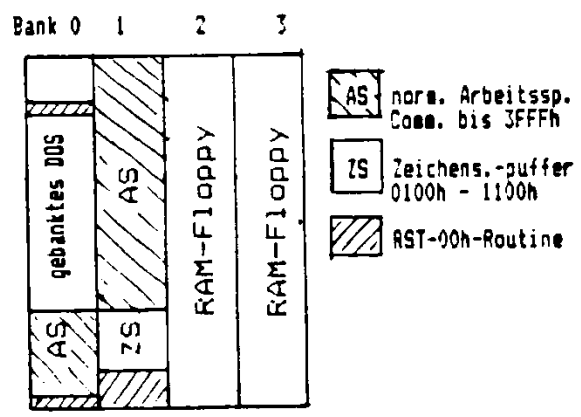


c) bei 4 Computern mit je 256 kB RAM

Abb. 6: Die Adressierung der 16 64-kB-Blocks

Part F9h	00xxxxxx	01xxxxxx	10xxxxxx	11xxxxxx
Part F3h	xxxxxx00	0	1	2	3
	xxxxxx01	4	5	6	7
	xxxxxx10	8	9	A	B
	xxxxxx11	C	D	E	F

Abb. 7: Die Speicherbelegung pro Computer (auch im Originalzustand)



B: Die Software zur Umschaltung zwischen vier Rechnern

Während dieser Text entsteht, lauert im selben Gehäuse, aber dennoch in einem zweiten Computer ZEUS mit dem Quelltext des Programms, das nun vorgestellt werden soll. So kann während der Arbeit mal eben hinübergeschaltet werden, um vergessene Einzelheiten ins Gedächtnis zurückzurufen. Und damit ist auch schon gesagt, was das Ganze soll. Wir sozial Schwachen, die sich nur einen Rechner leisten können, brauchen auf den Komfort mehrerer Maschinen nun nicht mehr zu verzichten.

Da gibt es Blocks, RAM-Reihen, Banks, Adreßräume usw.. Um die Verwirrung in Grenzen zu halten, schlagen wir folgende Sprachregelung vor: Ein Block ist eine bestimmte Datenmenge, die gleichzeitig bearbeitet wird. Eine RAM-Reihe sind acht Speicherchips 41256, die über Port F9h, Bits 6 und 7 gleichzeitig selektiert sind. Funktional gesehen ist es sinnvoll, sie als Banks zu bezeichnen. Die Speichervierviertel innerhalb einer Reihe von 41256-RAMs, um die es in diesem Beitrag geht, sollen hier Computer genannt werden, denn das Programm verwirklicht das Konzept von vier Rechnern, die einander auf Knopfdruck abwechseln.

Dieser Knopfdruck sollte nicht aus Versehen erfolgen können. Deshalb wird mit der Kombination von Shift-CTRL und einer Taste von 0-3 im Ziffernblock umgeschaltet. Dazwischen liegen rd. 35 cm, so daß Zufälle ausgeschlossen sind. Gleichwohl wäre ein solches Versehen nicht tragisch, denn eine Umschaltung ist sowohl mit, wie auch ohne Absicht erlaubt und ungefährlich. Alle Speicherinhalte bleiben bestehen, sogar die restliche Blinkzeit eines Cursors wird auf die Nanosekunde genau beim Zurückschalten nachgeholt. Der Tastendruck wird während der Interrupt-Bearbeitung erkannt. Hierzu wird ein CALL nach 45BEh (Erkennen der Dreitastenbefehle) an der Stelle 4601h umgeleitet (Z. 490 im Listing).

Das alte Lied: Wohin mit Erweiterungen des Betriebssystems? Diesmal mußte ein Speicherbereich dran glauben, der von TCS bei der Modifikation des Microsoft-Interpreters offenbar vergessen wurde. Ab 0000h wird weder nach dem Einschalten noch bei einem Reset gelesen, denn dann ist das Boot-EPROM aktiviert. In diesem Speicher-Niemandsland wird vom Programm erkannt, ob Shift-CTRL-Ziffer gedrückt wurde. Dazu stehen aber nur 8 Bytes bis zum Beginn der RST-08h-Routine zur Verfügung. Deshalb verabschiedet sich das neue RST 00h an der Stelle 0007h kurzerhand aus der Bank 0 und macht in Bank 1 weiter, wo das Betriebssystem erst wieder bei 0100h (Zeichensatz-Puffer) Platz beansprucht.

Diese Umschaltung geschieht mit dem Bit 0 von Port F9h. Da die normale Arbeitsbank Bank 1 ist, wobei Common unten in Bank 0 liegt, ist der Trick simpel: Das Common-Bit wird auf 1 gesetzt, so daß die CPU, ohne es zu merken, nun auch unten in Bank 1 liest, denn: Common, d. h. der Bereich, wo immer in Bank 0 gearbeitet wird, liegt jetzt ab E000h. Im Listing hat dieses Programmsegment das Label rst00 (Z. 664). Es liegt sowohl in Bank 0 als auch in Bank 1 bereit, so daß die Umschaltung aus beiden Banks auf die gleiche Weise geschieht. Es wird einfach D0 des Ports F9h invertiert.

Beim Umschalten aus Bank 0 (Common unten) wird die scheinbar nahtlose Fortsetzung in Bank 1 gefunden (Label bank1). Beim Zurückschalten aber muß rechtzeitig vor dem Erreichen der anschließenden RST-08h-Routine zur Fortsetzung der INT-Serviceroutine gesprungen werden. Deswegen wurde zuvor in Bank 1 HL mit 45BEh geladen, der Adresse, wo es weitergehen soll (s. o.). In Z. 495 wurde zu diesem Zweck der Befehl JP (HL) (Code E9h) in Bank 0 ans Ende dieses Segments gepatcht.

Nachdem HL die Adresse der Fortsetzung enthält, wird die Tastatur in 3880h und 38C0h nach den Tasten Shift und CTRL sowie nach den Ziffern 0-3 im Dezimalblock befragt. Wenn diese Tastenkombination nicht gedrückt wurde, will der User offenbar nicht den Computer wechseln. Es erfolgt ein

Rücksprung nach 0000h, wo wieder die alte Bank-Konfiguration eingestellt wird. Es ist, als wäre nichts geschehen.

Wurde aber Shift-CTRL-Ziffer eingegeben, wird zunächst der Ziffern-code in einen Zähler für die gewünschte Computer-Nr. umgerechnet und nach B geladen. Da sich die vier Computer außer der CPU natürlich auch alle Ports teilen müssen, werden die Zustände der beiden einzig wirklich lebenswichtigen, FAh und F9h, zwischengepuffert, jener sofort, dieser etwas später. Bei dieser Gelegenheit wird auf FAh das Bit 5 gesetzt, um einen Schreibschutz für den Adreßbereich des ehemaligen TRS-80-ROMs zu erreichen. Es könnte ja sein, daß ein Programm ihn als Sektorpuffer o. ä. benutzt.

In der Folge bekommt die Umschaltroutine auch ihren eigenen Stack. Das muß sein, weil er z. B. bei BASIC im Himem liegt, wo ja jetzt die Bank 0 selektiert ist; der Stack wäre unerreichbar! Der alte Stapelzeiger wird gepuffert. Je nach dem kurz zuvor aufgesetzten Computer-Zähler wird so oft zum Ausgangswert -15h (=+EBh) 15h addiert, bis die Nr. des zuständigen Computers im Akku steht.

Die Zahl 15h ergibt sich folgendermaßen: Beispielsweise bei der Selektion von RAM 1, Bildschirm und Zeichensatz Nr. 1 und Graphikspeicher Nr. 1 liegt am Port F3h der Wert xx01.0101b an (xx sind die beiden Steuerbits, die keinem eigenen Computer zugeordnet sind). Abgesehen von den Bits 6 und 7 ist das die Zahl 15h. Um einen Computer höher zu gehen, wird deshalb 15h addiert. Der so gewonnene Output für den Port F3h wird an die Adresse mit dem Label newcomp gepatcht.

Jetzt müssen alle Register gerettet werden, weil die benachbarten Computer nicht jeder eine eigene CPU haben. Die INT-Routine hat die Hälfte dieser Arbeit schon erledigt, so daß nur noch die Indexregister und der alternative Registersatz gepushed zu werden brauchen.

Nun kann umgeschaltet werden. Das Bit 7 wird zuvor gesetzt, damit bei Common-Einstellung nicht in die Bank 0 des Computers Nr. 0 zurückgeschaltet wird. Modifikationen, die der User oder das System dort vorgenommen hat, sollen ja nicht für alle vier Computer gelten.

Nach dem Umschalten auf einen Nachbarcomputer wird alles wieder restauriert. Es beginnt mit dem Output auf den Port F9h, der aber zunächst noch in E zwischengelagert wird. Zur Restauration des Bildschirmformats (auch den Video-Controller gibt es nur einmal) müssen nämlich noch im oberen Common-Bereich, der ja z. Zt. noch selektiert ist, die CRTC-Parameter aus 37F0h-37FFh ausgelesen und auf den Controller ausgegeben werden.

Diese Video-Codes liegen nun wiederum im unteren Common. Also hilft nur eine Umschaltung auf Bank 0, die beide Common-Bereiche gleichzeitig zugänglich macht. Daher der Umweg über das Unterprogramm vidpar, das die komplette Bank 0 verwalten kann. Zur Erinnerung: Der Arbeitsspeicher ist die Bank 1, daher diese Umstände, vidpar macht sich die Arbeit leicht, indem es ein fertiges Unterprogramm des Betriebssystems ab 3551h aufruft, das die Codes an den Video-Controller ausgibt. Zuletzt schaltet es wieder auf Bank 1, Common oben und kehrt zurück.

Jetzt werden auch der Port FAh, die CPU-Arbeitsregister und der Stackpointer restauriert. Wohlgermerkt sind es die Registerinhalte, die der Nachbarcomputer irgendwann zuvor für sich selber auf den eigenen Stack der Umschaltroutine gelegt hat. Was soeben gerettet wurde, liegt im alten Computer sicher. Mit dem abschließenden Sprung nach 0000h (Label rst00) wird Common wieder nach unten verlegt und die INT-Routine - diesmal diejenige des neuen Computers - zuende bearbeitet.

Außer der Umschaltung mit dem komplizierten Tastendruck geht es auch durch Aufruf des Programms mit seinem Namen. Um dabei nicht jedesmal die

Floppy rotieren zu lassen, ist es hier eine Systemdatei, so daß sie gleich beim Booten in den Speicher geladen wird. Sie erledigt verschiedene Aufgaben:

Bevor die Umschaltroutine arbeiten kann, sind etliche Initialisierungen notwendig. Hauptsächlich muß natürlich der Computer Nr. 0 auf die drei Nachbarn kopiert werden, damit die CPU nach dem Umschalten sofort sinnvollen Code lesen kann. Bei Testläufen hat es sich erwiesen, daß nur vom Computer 0 aus die Kopie stattfinden darf. Es könnte ja sein, daß der User ein wenig damit herumexperimentieren möchte oder sein digital bereits ebenso verseuchtes Kind nicht weiß, daß Mama oder Papa schon den Rechner wechselte. Deshalb wird bald nach dem Einsprung (Label test0) überprüft, ob die Bits 0 und 1 unseres neuen Systembytes 2 (zuständig für den Hauptspeicher) beide auf 0 stehen. Ist das nicht der Fall, wird einfach nur der derzeit selektierte Computer angezeigt. Diese Anzeige kann auch mit einem dem Aufruf folgenden Fragezeichen erhalten werden.

Ein weiteres Argument kann dem Befehl folgen: Wenn eine Ziffer von 0-4 eingegeben wurde, wird der so angewählte Computer selektiert. Allerdings nur dann, wenn zuvor die Initialisierung stattgefunden hat. Das merkt das Programm daran, ob an der Stelle 0000h in der Bank 1 bereits der Befehl DI steht. Ist das nicht der Fall, wird mit einer Fehlermeldung das Unternehmen abgebrochen. Wenn aber alles klar ist, ergibt die eingegebene Ziffer bereits die gewünschte Computer-Nr.. Es kann daher in Bank 1 an der Stelle switch mitten in die RST-00h-Routine gesprungen werden, wobei die Ermittlung des Computers und - sehr wichtig! - die Tastaturabfrage übersprungen werden.

Zusätzlich gibt es folgende Argumente: Wie in G-DOS üblich, kann das Ein- und Ausschalten des Features der vier Computer mit J (für "ja") und N (für "nein") erfolgen. J wird auch angenommen, wenn kein Argument folgt. N kann in jedem der vier Computer eingegeben werden. Dann bleibt immer dasselbe Viertel der 41256-RAMs selektiert, um in diesem Computer beispielsweise eine unverfälschte INT-Routine zu Testzwecken zu haben. Dennoch kann mit SYS,BK,n (n = Nr. des Computers) umgeschaltet werden; nur die Tastenkombination Shift-CTRL-n ist deaktiviert. Und dies gilt auch nur für den gerade selektierten Computer. Mit R (für "recover") werden die vier Computer reaktiviert, wenn ein zwischenzeitliches Reset oder ein SYS,BK,N es abschaltete. Auch hier muß zuvor bereits die Initialisierung erfolgt sein, sonst wird ein Fehler gemeldet.

Falls die Initialisierung der Nachbarcomputer gewünscht wird und erlaubt ist, wird's jetzt hektisch: Im Verlauf der nächsten Mikrosekunden werden der Stackpointer, das Systembyte 0 (Port F9h) und das Systembyte 1 (Port FAh) als Initialwerte in die RST-00h-Routine gepatcht. Deren kürzerer Teil, der in Bank 0 liegen soll, wird nach 0000h verschoben. Das bereits erwähnte Byte E9h für JP (HL) kommt dahinter. Durch verschiedene Manipulationen der beiden Systembytes 0 und 1 werden Bildschirme aus- und wieder angeknipst, expandiert und wieder komprimiert, Tastaturen abgewürgt, Zeichensätze freigegeben, Common-Bereiche umdefiniert und dergleichen mehr. Es würde zu weit führen, das hier im einzelnen zu erläutern. Die Kommentare im Listing protokollieren es jeweils. Eine Hilfe zum Verständnis kann auch die Tabelle am Ende dieses Abschnitts geben.

Die Kopie des Hauptspeichers kann nicht auf einen Schlag geschehen. Das liegt daran, daß jeweils ein Common-Bereich erforderlich ist, der immer in der Bank 0 arbeitet. Das Feature, das mit Q6 des Latches ermöglicht werden sollte, funktionierte ja leider nicht. Soll also beispielsweise im Common-Adreßraum gerade die Bank 2 kopiert werden, ist sie einfach nicht da. Daher wird das Unterprogramm COPY, das später erklärt werden soll, sowohl in das obere als auch in das untere Common kopiert. Die beiden Routinen bearbeiten dann jeweils nur eine Speicherhälfte von viermal 32 kB. Mit dem oben residierenden Kopierprogramm wird gleich VIDPAR mitverschoben.

Zuerst wird der Bildschirm vervierfacht. Dazu muß er zunächst in einen Puffer im unteren Adreßraum der Bank 1 kopiert werden. Er findet Platz hinter dem Zeichensatzpuffer ab 1100h. Beim Schreiben des gewählten Zeichensatzes in die drei benachbarten Zeichengeneratoren steht das Adreß-LSB des gerade bearbeiteten Zeichens im Bildschirm an der Stelle 3C00h. Um den dadurch veränderten Bildschirm nicht umständlich restaurieren zu müssen, erfolgt zuerst die Kopie des Zeichensatzes, erst anschließend wird der Bildschirminhalt aus dem Puffer in den Bildwiederholtspeicher geladen. Das Zeichen von 3C00h wurde im Puffer ja nicht verändert. Die Auswahl des gerade aktuellen Bildschirms bzw. Zeichensatzes geschieht über die Bits 2 und 3 des Ports F3h.

Um die beiden Kopier Routinen aufrufen zu können, ist ab Z. 576 wieder einiges Banking erforderlich. Auch hier mag der Verweis auf die Kommentare und die Tabelle ausreichen. Die Routinen selbst arbeiten folgendermaßen :

Am Label copy wird ein Zähler mit 32 aufgesetzt. Die Datenmenge, für die der erforderliche Puffer ausreicht, beträgt 1 kB. 32 kB pro Bank übernimmt eins der beiden Kopierprogramme. Gleich anschließend wird dieser Zähler gerettet und derjenige für die Banks geladen. Jetzt wird aus jeder Bank des Computers Nr. 0 ein Block zu 1 kB in den Puffer geholt. Schließlich wird in der inneren Schleife, welche die drei Nachbarcomputer abzählt, der Pufferinhalt auf die Nachbarn kopiert. Um die Adresse des zu bearbeitenden 1-kB-Blocks zu aktualisieren, genügt es, ihr MSB um 4 zu erhöhen (ab Z. 654), denn 0400h = 1 kB.

Um die Adressen beim Blocktransferbefehl LDIR nicht zu zerstören, müssen sie fortlaufend gerettet und wieder restauriert werden. Neben den LDIR-Befehlen selbst sind es diese PUSHes und POPs, die beim Kopieren die meiste Zeit verbrauchen. Insgesamt läuft das Programm deshalb etwa dreieinhalb Sekunden, bei immerhin einem Megabyte eine erträgliche Zeit. Um möglichst schnelle 8-Bit-Befehle anwenden zu können, werden nur die MSBs geladen. Die LSBs betragen immer 00h, so daß sie nicht berücksichtigt werden müssen. Diese - im Computer-Maßstab - extrem langwierige Prozedur läuft jedoch nur nach dem Aufruf des Programms ab. Nach der Initialisierung geschieht das Umschalten zwischen den Computern hundertmal schneller als ein Blinzeln.

Nach den Kopiervorgängen bleibt nur noch übrig, die Systembytes 0 und 1 sowie den Stackpointer zu restaurieren. Mit RET wird ins DOS zurückgekehrt.

Wie gesagt, müssen sich die vier Computer die gesamte Hardware teilen, lediglich die Speicherbausteine haben sie für sich. Der User muß deshalb wissen, daß gewisse Einschränkungen in Kauf zu nehmen sind. So ist es beispielsweise nicht möglich, mit der seriellen Schnittstelle auf Computer Nr. 0 eine Mailbox anzurufen und zwischendurch damit einen seriellen Drucker zu bedienen. Die Schnittstelle wird beim Umschalten nicht berücksichtigt. Dasselbe gilt für die Floppies, die PIO und evtl, angeschlossene Peripherie auf dem Systembus. Dennoch können alle Bauteile von allen Computern benutzt werden. Es darf eben nur nicht umgeschaltet werden, wenn sich z. B. gerade die Floppy dreht.

In der hier vorgestellten Form ist das Programm Teil der Systemdatei SYS25/SYS. Dies erschien sinnvoll, weil es ganz entschieden ins Betriebssystem gehört und notfalls auch unter Mini-DOS mitten in einem Programmablauf abrufbar sein sollte. Da die Library des Genie III schon voll ist, wird es mit dem SYS-Befehl (hier SYS,BK für "BANK") aktiviert, der an dieser Stelle bereits vorgestellt wurde. So erklärt sich das Vorplänkel am Beginn des Listings, wo zum alten Inhalt von SYS25/SYS (Bedienung der F-Tasten) der Test auf den neuen Requestcode DBh für SYS,BK eingefügt wurde. Was zwischen den Zeilen 27 und 405 steht, ist mit LIST OFF ausgespart, denn die F-Tasten sind hier nicht unser Thema. Dies

gilt auch für die Label-Definitionen bis Z. 22.

Es gibt keine Einschränkungen des Speicherplatzes. Nach dem Laden steht SYS25/SYS zunächst im DOS-Overlay-Bereich ab 4D00h, der für diese Zwecke vorgesehen und reserviert ist. Die Bereiche, die die RST-00h-Routine und das Unterprogramm vidpar belegen, sind völlig frei und werden weder vom Betriebssystem noch von den Autoren bekannten Anwenderprogrammen beansprucht. Für das RAM-Floppy-Programm MEMDISK/CMD gilt sogar, daß es vor dem Aufruf bereits aktiviert sein darf. Es wird nicht überschrieben, auch nicht die Dateien, die in der RAM-Floppy stehen. Alles wird einfach in die Nachbarcomputer hinüberkopiert und kann auch von dort aus benutzt werden.

Wie sich herausstellte, kann aber leider nicht unter BASIC CMD'SYS,BK<,J>' eingegeben werden, um BASIC/CMD solo oder gar mit bereits geladenem Programm in die drei Nachbarcomputer hinüberzukopieren. Vielleicht gelingt es in der Folge, das ebenfalls noch zu ermöglichen. Einstweilen muß es unter DOS geschehen, wonach dann in jedem der vier Computer BASIC/CMD gesondert gestartet werden muß. Auf jeden Fall aber kann nach erfolgter Initialisierung mitten im Lauf eines BASIC-Programms auf einen Nachbarcomputer umgeschaltet werden. Mit anderen Programmen sind bisher diese Schwierigkeiten nicht aufgetreten. Im Zweifelsfall hilft immer die Methode, die eben für BASIC beschrieben wurde.

Eine weitere Besonderheit muß beim Betrieb mit BASIC beachtet werden. In der RST-00h-Routine kehrt der Akku mit einem Wert >0 zurück (Output für den Port F9h). Das Z-Flag ist rückgesetzt, d. h. die NZ-Bedingung ist gegeben. BASIC versteht das als DOS-Fehler, gibt eine entsprechende Meldung aus und sagt 'READY'. Deshalb kann man zwar ohne weiteres den Befehl CMD"SYS,BK,n' (n ist die Computer-Nr.) über die Tastatur eingeben, wenn es aber innerhalb eines Programms geschieht, wird dieses wegen des angeblich auf getretenen Fehlers abgebrochen. Das wäre mit geringen Modifikationen zu beheben gewesen, die aber Speicherplatz im kostbareren oberen Adreßraum gekostet hätten. So wurde darauf verzichtet, denn eine Umschaltung, die ein BASIC-Programm selbsttätig steuert, ist praxisfern.

Das Verändern von SYS-Files und das Implementieren eines neuen Library-Befehls (auch in der Zweit-Lib mit SYS) wurde hier schon mehrfach gezeigt. Es soll nicht erneut erklärt werden. Wer sich scheut, auf diese Weise ins Betriebssystem einzugreifen, kann das Programm in der Zeile 421 mit ORG 5200h beginnen und ein CMD-File daraus machen. Dann muß das END-Statement die Startadresse sidekck haben. Ansonsten ändert sich dabei nichts.

Zuletzt sollen noch diejenigen Datenbits der hier erwähnten Ports erläutert werden, die für das Programm von Belang sind:

	Bit 7	6	5	4	3	2	1	0
F3h	Common-Defin.	.	=HRG-Nummer=	.	=Bildsch.- u. =Zeichens.-Nr.=	.	=Hauptsp.-Nr.=	.
F9h	=Orig.-Banks=	Zeichen-	Common-	Defin.
FAh	.	.	Schreib-	Video	.	.	mm-I/O	an/aus
	.	.	schutz	ab 3800h

Sollte es mit der Software zur Verwirklichung von 4 Computern in einem Schwierigkeiten geben, kann Arnulf Sopp weiterhelfen.

```

4000          00023      ORG      4d00h
4000  FEFB      00024  sys25  CP      0fbh      ;Lib-Befehl F# ?
4002  2822      00025      JR      Z,fkey      ;falls ja
4004  FEDB      00026      CP      0dbh      ;Lib-Befehls SYS,BK ?
4006  CA2A50    00027      JP      Z,sidekck      ;falls ja
00405
00406
00407 ;*****
00408 ;
00409 ;
00410 ;           Sidekick für das Genie IIIs
00411 ;           4 Computer in einem Gehäuse
00412 ;
00413 ;           Hardware: 1 MB RAM, 8 kB Bildschirm,
00414 ;           16 kB Zeichensatz, 256 kB Graphik
00415 ;
00416 ;           (C) 1986 by      Arnulf Sopp
00417 ;           starring:      Helmut Bernhardt
00418 ;*****
00419
00420 ;Test, ob vier Computer bereits initialisiert sind
5015  DBF9      00421  tstinit IN      A,(0f9h)      ;Systembyte 0
5017  57        00422      LD      D,A           ;retten
5018  F601      00423      OR      01h          ;unten Bank 1 selektieren
501A  F3        00424      DI           ;vorsichtshalber
501B  D3F9      00425      OUT     (0f9h),A
501D  3A0000    00426      LD      A,(0000h)    ;Anf. der RST-00-Routine
5020  FEF3      00427      CP      0f3h          ;steht dort bereits DI ?
5022  7A        00428      LD      A,D           ;Systembyte 0
5023  D3F9      00429      OUT     (0f9h),A    ;restaurieren
5025  C8        00430      RET     Z            ;o. k., falls schon DI
5026  E1        00431      POP     HL           ;RET-Adresse vom Stack
5027  C3224D    00432  jperr  JP      error      ;raus mit Fehlerbeding.
00433
00434 ;Tests, ob Befehl mit Argument eingegeben wurde
502A  DBF3      00435  sidekck IN      A,(0f3h)      ;Systembyte 2
502C  E603      00436      AND     03h          ;nur Hauptspeicher-Bits
502E  4F        00437      LD      C,A           ;retten
502F  CDD54C    00438      CALL   4cd5h          ;folgt etwas dem Befehl?
5032  2841      00439      JR      Z,test0       ;falls nein
5034  7E        00440      LD      A,(HL)        ;was folgt?
5035  FE52      00441      CP      'R'           ;Recover 4 Computer?
5037  200A      00442      JR      NZ,norecov    ;falls nein
5039  CD1550    00443      CALL   tstinit        ;schon initialisiert?
503C  3EC9      00444      LD      A,0c9h        ;RET-Befehl
503E  329150    00445      LD      (retrecv),A   ;dort patchen
5041  1838      00446      JR      init          ;dort weiter
5043  FE4A      00447  norecov CP      'J'           ;"ja" für initialisieren?
5045  282E      00448      JR      Z,test0       ;falls ja
5047  FE4E      00449      CP      'N'           ;"nein" für abschalten?
5049  2007      00450      JR      NZ,notoff     ;falls nicht
504B  21BE45    00451      LD      HL,45beh      ;alte Adr. in INT-Routine
504E  220146    00452      LD      (4601h),HL   ;dort restaurieren
5051  C9        00453      RET           ;dann Ende
5052  FE3F      00454  notoff CP      '?'          ;Status-Abfrage?
5054  2819      00455      JR      Z,dspcmp1     ;falls ja
5056  FE30      00456      CP      '0'           ;Computer-Nr. eingegeben?
5058  38CD      00457      JR      C,jperr       ;falls falsche Eingabe
505A  FE34      00458      CP      '4'           ;nur Comp. Nr. 0-3
505C  30C9      00459      JR      NC,jperr      ;falls falsche Eingabe
00460
00461 ;neuen Computer selektieren, falls schon initialisiert
505E  E603      00462      AND     03h          ;ASCII -> binär 0-3
5060  3C        00463      INC     A            ;für Zähler B: 1-4

```

```

5061 47      00464      LD      B,A          ;Computerzähler aufsetzen
5062 CD1550  00465      CALL   tstinit      ;schon initialisiert?
5065 E1       00466      POP    HL           ;RET-Adresse
5066 DBF9    00467      IN     A,(0f9h)     ;Systembyte 0
5068 F601    00468      OR     01h          ;Common oben, Bk. 1 unten
506A D3F9    00469      OUT   (0f9h),A     ;Systembyte neu ausgeben
506C C31E00  00470      JP     switch-rst00 ;dort Computer umschalten
                    00471
                    00472 ;aktuelle Computer-Nr. anzeigen
506F 79      00473 dspcmp1 LD      A,C          ;Computer-Nr.
5070 F630    00474 dspcmp2 OR     30h          ;binär -> ASCII
5072 C33300  00475      JP     0033h        ;anzeigen und raus
                    00476
                    00477 ;falls Comp. Nr. 0, dann initialisieren, sonst anzeigen
5075 79      00478 test0  LD      A,C          ;Computer-Nr.
5076 B7       00479      OR     A            ;Nr. 0 selektiert?
5077 20F7    00480      JR     NZ,dspcmp2   ;nur anzeigen, falls nein
5079 D3F3    00481      OUT   (0f3h),A     ;auch Video Nr. 0 usw.
                    00482
                    00483 ;INT-Patch in SYS0, RST-00-Routine für Bank 0 verschieben
507B ED73D751 00484 init  LD      (spbuf1),SP ;Initialstack
507F 216B51  00485      LD     HL,rst00     ;RST-00h-Routine
5082 010700  00486      LD     BC,bank1-rst00 ;Länge der Rout. in Bk. 0
5085 50       00487      LD     D,B          ;DE (- Zieladresse 0000
5086 58       00488      LD     E,B          ;
5087 F3       00489      DI                    ;keine Störungen
5088 ED530146 00490      LD     (4601h),DE   ;CALL 0000 statt 45BE
508C EDB0    00491      LDIR                    ;RST-00-Routine -> Bank 0
                    00492
                    00493 ;Befehl zum Weitersprung schließt RST-00-R. in Bank 0 ab
508E 3EE9    00494      LD     A,0e9h       ;Befehl JP (HL)
5090 12      00495      LD     (DE),A       ;dahinter patchen (0007h)
                    00496
                    00497 ;Rout. zur Kopie der ob. Speicherhälfte nach 3800 laden
5091 DBFA    00498 retrecv IN   A,(0fah)   ;Systembyte 1
5093 F5       00499      PUSH  AF            ;retten
5094 32C851  00500      LD     (fabuf),A    ;Patch in RST 00, Bank 1
5097 F611    00501      OR     11h          ;Großbildsch., mm-I/O aus
5099 D3FA    00502      OUT   (0fah),A     ;Systembyte 1 neu ausgeb.
509B 08      00503      EX   AF,AF'         ;nochmals retten
509C 213751  00504      LD     HL,copy       ;Kopierprg. f. ob. Hälfte
509F 110038  00505      LD     DE,combot     ;im unt. Common-Bereich
50A2 0E34    00506      LD     C,rst00-copy ;Länge der Routine
50A4 EDB0    00507      LDIR                    ;nach 3800 übertragen
                    00508
                    00509 ;Routine zur Restauration der Bildschirmparameter und
                    00510 ;zur Kopie der unteren Speicherhälfte nach F600 laden
50A6 DBF9    00511      IN     A,(0f9h)     ;Systembyte 0
50A8 F5       00512      PUSH  AF            ;retten
50A9 F601    00513      OR     01h          ;Common oben, Bank 1 unt.
50AB F5       00514      PUSH  AF            ;nochmals retten
50AC 32BD51  00515      LD     (f9buf),A    ;Patch in RST 00, Bank 1
50AF D3F9    00516      OUT   (0f9h),A     ;geänd. Systemb. ausgeben
50B1 21E51  00517      LD     HL,vidpar     ;CRIC-Restaur., Kopierr.
50B4 1100F6  00518      LD     DE,comtop     ;im oberen Common-Bereich
50B7 0E3D    00519      LD     C,rst00-vidpar ;Länge der Routine
50B9 EDB0    00520      LDIR                    ;nach F600 kopieren
                    00521
                    00522 ;RST-00-Routine für Bank 1 in diese Bank nach 0000 laden
50BB 50      00523      LD     D,B          ;DE (- 0000h, wie oben
50BC 58      00524      LD     E,B          ;(Zieladr. RST-00-Rout.)
50BD 0E70    00525      LD     C,endrst00-rst00 ;Länge RST-00-R.
50BF EDB0    00526      LDIR                    ;RST-00-Routine -> Bank 1
                    00527

```

```

00528 ;memory-mapped I/O einschalten, Zeichen-RAM freigeben
50C1 F602 00529 OR 02h ;Zeichensatz freigeben
50C3 D3F9 00530 OUT (0f9h),A
50C5 08 00531 EX AF,AF' ;geändertes Systembyte 1
50C6 F5 00532 PUSH AF ;wird noch gebraucht
50C7 E6FE 00533 AND Ofeh ;mm-I/O einblenden
50C9 D3FA 00534 OUT (0fah),A ;Systembyte 1 neu ausgeb.
00535
00536 ;Großbildschirm ab 3800 in Bank 1 ab 1100 zwischenspeich.
50CB 210038 00537 LD HL,3800h ;Anfang Bildschirm
50CE 110011 00538 LD DE,1100h ;Anfang Puffer
50D1 0608 00539 LD B,08h ;BC (- Länge 2 kB
50D3 EDB0 00540 LDIR ;Bildschirm puffern
00541
00542 ;Zeichensatz in 3 neue Zeichengeneratoren kopieren
00543 ;(3 obere Viertel der beiden 6264)
50D5 0603 00544 LD B,03h ;3 Bildsch. u. Zeichens.
50D7 C5 00545 vidloop PUSH BC ;Bildschirmzähler retten
50D8 78 00546 LD A,B ;Bildschirm-Nr.
50D9 07 00547 RLCA ;Bit 0-1 -> 2-3
50DA 07 00548 RLCA
50DB D3F3 00549 OUT (0f3h),A ;Bildschirmseite selekt.
50DD 2601 00550 LD H,01h ;HL (- Beginn Zeichensatz
50DF AF 00551 XOR A ;ab ASCII 00
50E0 32003C 00552 chrlop1 LD (3c00h),A ;LSB Zeichenadresse
50E3 08 00553 EX AF,AF' ;retten
50E4 1684 00554 LD D,84h ;MSB in D
50E6 0610 00555 LD B,10h ;16 Bytes pro Zeichen
50E8 7E 00556 chrlop2 LD A,(HL) ;1 Byte aus dem Puffer
50E9 12 00557 LD (DE),A ;ins Zeichensatz-RAM
50EA 23 00558 INC HL ;auf nächstes Byte
50EB 7A 00559 LD A,D ;Adreß-MSB
50EC C608 00560 ADD A,08h ;um 2 kB erhöhen
50EE 57 00561 LD D,A ;neues MSB
50EF 10F7 00562 DJNZ chrlop2 ;bis 1 Zeichen eingelesen
50F1 08 00563 EX AF,AF' ;LSB Zeichenadresse
50F2 3C 00564 INC A ;nächster ASCII-Wert
50F3 20EB 00565 JR NZ,chrlop1 ;bis 256 Zeichen eingel.
00566
00567 ;Bildschirm aus dem Puffer in 3 neue Video-Speicher kop.
00568 ;(3 obere Viertel des einzelnen 6264)
50F5 1638 00569 LD D,38h ;Bildschirmadresse
50F7 0608 00570 LD B,08h ;BC (- Bildschirmlänge
50F9 EDB0 00571 LDIR ;Bildschirmseite füllen
50FB C1 00572 POP BC ;Bildschirmzähler
50FC 10D9 00573 DJNZ vidloop ;bis 4 Bildschirme fertig
00574
00575 ;Vorber. d. Hauptspeicherkopien, Aufruf d. Kopier Routinen
50FE F1 00576 POP AF ;Systembyte 1
50FF D3FA 00577 OUT (0fah),A ;mm-I/O ausblenden
5101 F1 00578 POP AF ;Systembyte 0
5102 D3F9 00579 OUT (0f9h),A ;Common oben, Bank 1 unt.
5104 60 00580 LD H,B ;untere Speicherhälfte
5105 16F7 00581 LD D,comtop+256/256 ;DE (- Pufferadr.
5107 ED732551 00582 LD (spbuf2),SP ;Stackpointer retten
510B 3100F7 00583 LD SP,comtop+256 ;eig. Stack im ob. Common
510E CD09F6 00584 CALL copy+offset1 ;unteren Speicher kopier.
5111 212D40 00585 LD HL,402dh ;DOS-Ready als RET-Adr.
5114 E5 00586 PUSH HL ;auf den Initial-Stack
5115 E6FE 00587 AND Ofeh ;Common unten
5117 D3F9 00588 OUT (0f9h),A ;Systembyte 0 neu ausgeb.
5119 210080 00589 LD HL,8000h ;obere Speicherhälfte
511C 1630 00590 LD D,30h ;DE (- Pufferadresse
511E 310039 00591 LD SP,combot+256 ;Stack im unteren Common

```

```

5121 CD0038 00592 CALL copy-offset2 ;oberen Speicher kopieren
5124 310000 00593 LD SP,0000h ;Stackptr. restaurieren
5125 00594 sdbuf2 EQU $-2 ;(variabel)
5127 F1 00595 POP AF ;altes Systembyte 0
5128 D3F9 00596 OUT (0f9h),A ;restaurieren
512A F1 00597 POP AF ;dto. Systembyte 1
512B D3FA 00598 OUT (0fah),A
512D C9 00599 RET ;zurück i. Betriebssystem
00600
00601 ;Bildschirm-Parameter restaurieren (Arbeitsadresse F600)
F600 00602 comtop EQU 0f600h ;im oberen Common-Bereich
A4D2 00603 offset1 EQU comtop-$ ;für Kopie d. unt. Hälfte
512E D3F9 00604 vidpar OUT (0f9h),A ;Bank 0 selektieren
5130 CD5135 00605 CALL 3551h ;Videoparameter setzen
5133 7B 00606 LD A,E ;Systembyte 0
5134 D3F9 00607 OUT (0f9h),A ;restaurieren
5136 C9 00608 RET
00609
00610 ;Speicher von Computer Nr. 0 auf Nachbarcomputer kopieren
00611 ;(Arb.-adr. 3800 f. ob. Hälfte o. hinter vidpar f. unten)
3800 00612 combot EQU 3800h ;im unteren Common-Ber.
1937 00613 offset2 EQU $-combot ;für Kopie der ob. Hälfte
00614
00615 ;32 Blocks zu je 1 kB in 3 Computer mit je 4 Banks kop.
5137 0620 00616 copy LD B,32 ;32 Blocks zu 1 kB kop.
5139 C5 00617 coplop1 PUSH BC ;Blockzähler retten
00618
00619 ;1 Block zu je 1 kB vor Kopie im Puffer zwischenspeichern
513A 0604 00620 LD B,4 ;4 Banks
513C C5 00621 coplop2 PUSH BC ;Bankzähler retten
513D E5 00622 PUSH HL ;dto. Zeiger
513E D5 00623 PUSH DE
513F 0604 00624 LD B,04h ;0400 = 1 kB
5141 EDB0 00625 LDIR ;Block puffern
5143 E1 00626 POP HL ;Quelle und Ziel tauschen
5144 D1 00627 POP DE
00628
00629 ;1 Block zu je 1 kB in 1 Bank von 3 Computern kopieren
5145 0603 00630 LD B,3 ;Zähler 3 Nachbarcomputer
5147 C5 00631 coplop3 PUSH BC ;Computerzähler retten
5148 78 00632 LD A,B ;lfd. Computer-Nr.
5149 D3F3 00633 OUT (0f3h),A ;Computer selektieren
514B 0604 00634 LD B,04h ;1 kB
514D E5 00635 PUSH HL ;Zeiger retten
514E D5 00636 PUSH DE
514F EDB0 00637 LDIR ;1 Block in Nachbarcomp.
5151 D1 00638 POP DE ;Zeiger restaurieren
5152 E1 00639 POP HL
5153 C1 00640 POP BC ;dto. Computerzähler
5154 10F1 00641 DJNZ coplop3 ;bis 1 Bl. in 4 C., 1 Bk.
00642
00643 ;dasselbe in der nächsthöheren Bank
5156 AF 00644 XOR A ;Computer Nr. 0 selekt.
5157 D3F3 00645 OUT (0f3h),A
5159 DBF9 00646 IN A,(0f9h) ;Systembyte 0
515B C640 00647 ADD A,40h ;1 Bank höher
515D D3F9 00648 OUT (0f9h),A ;selektieren
515F EB 00649 EX DE,HL ;Zeiger rücktuschen
5160 C1 00650 POP BC ;Bankzähler
5161 10D9 00651 DJNZ coplop2 ;b. 1 Bl. in 3 C., 4 Bks.
00652
00653 ;dasselbe für den nächsten Block der Speicherhälfte
5163 24 00654 INC H ;auf nächsten Block
5164 24 00655 INC H ;(= 0400 addieren)

```

```

5165 24      00656      INC      H
5166 24      00657      INC      H
5167 C1      00658      POP      BC          ;Blockzähler
5168 10CF    00659      DJNZ    coplop1    ;bis 1 Memory-Hälfte kop.
516A C9      00660      RET          ;zum Aufrufer
00661
00662 ;RST-00h-Rout., Segment in Bank 0 u. 1 (Arbeitsadr. 0000)
00663 ;von Bank 0 auf Bank 1 und umgekehrt schalten
516B F3      00664 rst00  DI          ;keine Störungen
516C DBF9    00665      IN      A,(0f9h)    ;Systembyte 0
516E EE01    00666      XOR     01h        ;Bank 0 (->) 1
5170 D3F9    00667      OUT     (0f9h),A
00668 ;in Bank 0 an der Adresse 0007 folgt hier JP (HL)
00669
00670 ;Fortsetzung in Bank 1 ab 0007
00671 ;INT-Forts. sichern, prüfen, ob Sh.-CTRL-Ziffer gedrückt
5172 21BE45  00672 bank1 LD      HL,45beh    ;Forts. d. INT-Routine
5175 3A8038  00673      LD      A,(3880h)   ;Tastatur Steuerzeichen
5178 FE03    00674      CP      03h        ;Shift-CTRL gedrückt?
517A 20EF    00675      JR      NZ,rst00    ;falls nein
517C 3AC038  00676      LD      A,(38c0h)   ;Tastatur Ziffernblock
517F E60F    00677      AND    0fh        ;0 - 3 gedrückt?
5181 28E8    00678      JR      Z,rst00    ;falls nein
00679
00680 ;Tastendruck erfolgt, Zähler für Computer-Nr. aufsetzen
5183 0600    00681      LD      B,00h      ;Anfangsw. Computer-Nr.
5185 04      00682 getcomp INC     B          ;B hochzählen
5186 0F      00683      RRCA   ;Tastenbit ins Cy
5187 30FC    00684      JR      NC,getcomp ;falls noch nicht gefund.
00685
00686 ;Systembytes, Stack und Adresse der Programmforts. retten
00687 ;(ab hier auch Bearbeitung mit Befehl SYS,BK,n)
5189 DBFA    00688 switch IN     A,(0fah) ;Systembyte 1
518B E6DF    00689      AND    0dfh      ;Schreibschutz aufheben
518D D3FA    00690      OUT     (0fah),A   ;Systembyte neu ausgeben
518F F620    00691      OR      20h       ;Schreibschutz 0000-2FFF
5191 325D00  00692      LD      (fabuf-rst00),A ;Systembyte 1 retten
5194 ED736C00 00693      LD      (spbuf1-rst00),SP ;SP retten
5198 3100F7  00694      LD      SP,comtop+256 ;eigener Stack im Common
519B E5      00695      PUSH   HL         ;Fortsetzungsadresse
519C DBF9    00696      IN     A,(0f9h)    ;Systembyte 0
519E 325200  00697      LD      (f9buf-rst00),A ;retten
00698
00699 ;Nachbarcomp. errechnen (RAM, Video, Zeichens., HRG-RAM)
51A1 3EEB    00700      LD      A,0ebh     ;-15, Anfw. Comp. & Video
51A3 C615    00701 getail ADD     A,15h      ;Comp., Vid. u. Grph. ab 0
51A5 10FC    00702      DJNZ   getail     ;bis neuer Comp. errechn.
51A7 324A00  00703      LD      (newcomp-rst00),A ;dort patchen
00704
00705 ;alle Alternativregister des bisherigen Computers retten
51AA 08      00706      EX     AF,AF'
51AB 09      00707      EXX
51AC DDE5    00708      PUSH   IX
51AE FDE5    00709      PUSH   IY
51B0 F5      00710      PUSH   AF
51B1 E5      00711      PUSH   HL
51B2 D5      00712      PUSH   DE
51B3 C5      00713      PUSH   BC
00714
00715 ;neuen Computer selektieren, dessen Videoparameter setzen
51B4 3E00    00716      LD      A,00h      ;Output für Port F3
51B5      00717 newcomp EQU    $-1    ;je nach Tastendruck
51B6 E63F    00718      AND    3fh        ;Steuerbits ausmaskieren
51B8 F680    00719      OR     80h        ;Common nur comp.-intern

```

```

51BA D3F3      00720      OUT      (0f3h),A      ;Nachbarcomputer selekt.
51BC 1E00      00721      LD       E,00h      ;Systembyte 0 (kommt in
51BD          00722 f9buf    EQU      $-1         ;vidpar in den Akku)
51BE 21F037    00723      LD       HL,37f0h   ;dort CRTC-Parameter
51C1 0610      00724      LD       B,10h      ;16 Codes
51C3 AF         00725      XOR      A          ;ab CRTC-Register 0
51C4 CD00F6    00726      CALL    vidpar+offset1 ;Video-Attribute restaur.
                    00727
                    00728 ;Systembytes, Register usw. restaur. und Programm forts.
51C7 3E00      00729      LD       A,00h      ;Systembyte 1
51C8          00730 fabuf    EQU      $-1         ;je nach dem
51C9 D3FA      00731      OUT      (0fah),A   ;restaurieren
51CB C1         00732      POP     BC          ;alle Register restaur.
51CC D1         00733      POP     DE
51CD E1         00734      POP     HL
51CE F1         00735      POP     AF
51CF FDE1      00736      POP     IY
51D1 DDE1      00737      POP     IX
51D3 D9         00738      EXX                    ;in den Alternativsatz
51D4 08         00739      EX      AF,AF'
51D5 E1         00740      POP     HL          ;Fortsetzungsadresse
51D6 310000    00741      LD       SP,0000h   ;SP restaurieren
51D7          00742 spbuf1  EQU      $-2         ;(variabel)
51D9 1890      00743      JR      rst00       ;umschalten und zurück
                    00744 endrst00
                    00745
4D00          00746      END     sys25       ;dort Einsprung

```

00000 Fehler

```

10 REM: Testprogramm, beschränkt sich auf das Umschalten des Bildschirms.
20 REM: Die vier Bildschirme wechseln einander ab. Zuvor muß das Initiali-
30 REM: sierungsprogramm mit SYS,BK(,J) vom DOS aus aufgerufen worden sein.
40 REM:
50 A=INP(&HF3):REM      Zustand des neuen Systembytes 2
60 A=AAND&HF3:REM      Video-Bits ausmaskieren
70 B=B+&H04:REM      Video-Bits hochschalten
80 B=BAND&H0C:REM      höhere Bits nicht belangen
90 OUT&HF3,A+B:REM      auf nächsten Bildschirm wechseln
100 FORI=0TO100:NEXT:REM ein bißchen warten
110 GOT060:REM      dasselbe Spielchen von vorne

```

Dasselbe in Assembler:

```

5200          00001      ORG     5200h
5200 DBF3      00002 start  IN      A,(0f3h)      ;Systembyte 2
5202 C604      00003 loop   ADD     A,04h      ;nächste Bildschirmseite
5204 E60F      00004          AND     0fh      ;nur bis Bit 3
5206 D3F3      00005          OUT     (0f3h),A   ;neu ausgeben
5208 08        00006          EX     AF,AF'     ;retten
5209 010080    00007          LD     BC,8000h   ;Zähler für Verzögerung
520C 0B        00008 delay  DEC     BC          ;herunterzählen
520D 78        00009          LD     A,B        ;prüfen, ob Zähler
520E B1        00010          OR     C          ;schon abgelaufen
520F 20FB      00011          JR     NZ,delay   ;falls noch nicht
5211 08        00012          EX     AF,AF'     ;Systembyte 2
5212 18EE      00013          JR     loop       ;und von vorne
5200          00014          END     start     ;dort Einsprung

```

00000 Fehler

C: Einbau des Erweiterungsboards und Anschluß der damit erzeugten / dafür benötigten Signale

Da die meisten der für diese Schaltung benötigten Signale direkt vom Z80 geliefert werden, war es zweckmäßig, ein Board zu entwerfen, das anstelle des Z80 in dessen Sockel gesteckt wird und den Prozessor selbst aufnimmt. Damit sind dann nur noch zehn Signale über freie Verdrahtung mit dem CPU-Board zu verbinden. Diese Leitungen sind an einen zweireihigen Pfofenstecker geführt. Die Signale COML* und COMH* werden von den Pins 12 und 13 von U84, 74LS00 abgenommen und in beliebiger Anordnung an die Pins 3 und 4 des Steckers geführt. MWR* an Pin 5 des Steckers ist von Pin 8 von U52 (74LS32) zu beziehen, SYSRES* an Pin 6 des Steckers kommt von Pin 13 von U68, 74LS74.

Die Adressen A16 und A17 für die dynamischen RAMs (für den Arbeitsspeicher an den Pins 1 und 2 und für den HRG-Speicher an den Pins 9 und 10 des Steckers S2) können nicht direkt an die RAMs gelegt werden. Sie müssen gemultiplext an ihre jeweiligen Pins 1 geführt werden. Dafür ist zunächst die Computerplatine auszubauen und die Pins 1 aller 32 (Arbeitsspeicher) sowie 8 RAMs (Graphik) untereinander zu verbinden.

Dann werden auf U35 und U15 (beide 74S157) je ein 74S157-IC mit den Pins 1, 8, 15 und 16 huckepack aufgelötet. Dafür muß das untere IC aus der Fassung genommen werden, um es nicht versehentlich in derselben festzulöten. Selbstverständlich müssen beim Auf einander löten beide ICs die gleiche Orientierung haben (Kerbe auf der gleichen Seite). Die nicht verbundenen Pins des oberen ICs werden um ca. 45° hochgebogen, um keinen Kontakt mit dem unteren zu haben.

An die Pins 13 und 14 des auf U35 angebrachten 74S157 (hier genügt auch ein 74LS157) werden zwei isolierte Litzen gelötet, die an die Pins 1 und 2 des Steckers geführt werden. Zwei Litzen, die an die Pins 13 und 14 des auf U15 gesetzten ICs gelötet sind, werden an die Pins 9 und 10 des Steckers geführt. In beiden Fällen spielt die Anordnung der beiden Leitungen keine Rolle. Die Pins 12 der beiden Huckepack-ICs werden je über einen 33-Ω-Widerstand mit dem Pin 1 eines der benachbarten 41256-RAMs verbunden. Dafür muß das jeweilige RAM-IC aus der Fassung genommen und der Draht vorsichtig mit wenig Lötzinn an das obere dicke Ende des Pins 1 gelötet werden. Es ist ebenso möglich, bei herausgenommener Mutterplatine den Widerstand um deren Rand herum auf die Lötseite zu führen. Es kann zweckmäßig sein, den in die Leitung eingefügten 33-Ω-Widerstand mit einem Stück Schrumpfschlauch davor zu bewahren, Kontakt mit irgendwelchen IC-Pins zu bekommen.

Die zur Vervierfachung von Video-RAM und Zeichengenerator (BIG4, BIG5 und BIG6) benötigten 6264LP15-ICs passen nicht mehr in die Fassungen der 6116LP15-RAMs. Hier hilft folgende Vorgehensweise: Die 6116-RAMs werden aus den Fassungen genommen. Die drei Kondensatoren an den Stirnseiten der Sockel werden ausgelötet und auf dem Rücken der 6264-RAMs mit kurzen Drahtstücken zwischen die Pins 14 und 28 gelötet. Die Pins 26 und 28 jedes 6264-Bausteins werden miteinander verbunden. Das IC, das in den Sockel von BIG4 gesteckt werden soll, muß an der Stirnseite (mit Kerbe) so weit beschliffen werden, daß es neben U87 mit den überstehenden Pins 1, 2, 27 und 28 in den Sockel paßt. Die beiden anderen 6264-ICs passen ohne zusätzliche Maßnahmen mit den links überstehenden vier Pins in die Sockel. Vor den Einstecken werden aber die Pins 23 aller drei RAMs um 45° abgebogen.

Die Pins 2 aller drei Speicher werden miteinander und mit dem Pin 8 des Steckers S2 des 4XG3s-Boards verbunden. Auch die hochgebogenen Pins 23 werden untereinander und mit dem Pin 7 des Steckers S2 verbunden. An die Pins 27 der RAMs werden folgende Signale gelegt (die vorher an die Pins 21 der 6116-Bausteine gelangt waren):

Pin 27 von RAM-IC	Signal	von	IC Typ	Pin
BIG4	61WR *		U54	32 3
BIG5, BIG6	CHRWR*		U5	244 1

Sollten beim Einbau der Aus-1-mach-4-Erweiterung Fragen oder Probleme auftreten, kann sich der Bastler an Helmut Bernhardt wenden. Ungebohrte Leerplatten bietet er zum Selbstkostenpreis an.

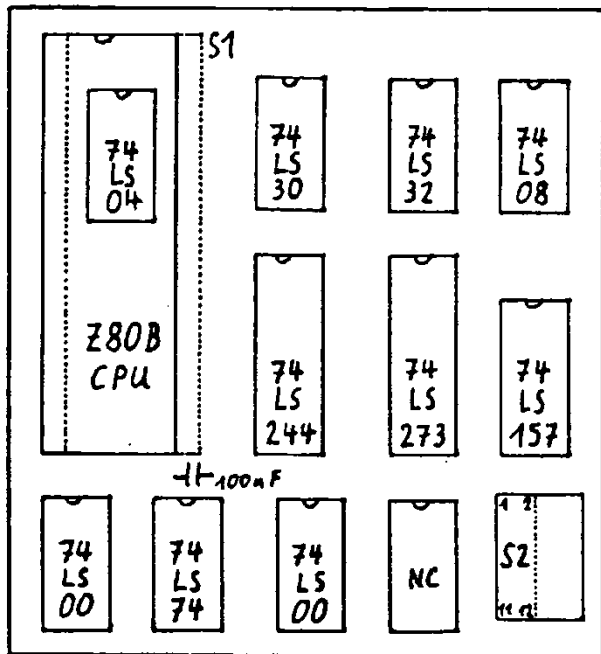


Abb. 8: Bestückungsplan der Platine (leicht vergrößert)

S1: 2 einreihige Steckerleisten für den 280-Sockel auf der Mutterplatine, je 20 Pins

S2: 1 zweireihiger geknickter Pfostenstecker, 12 Pins

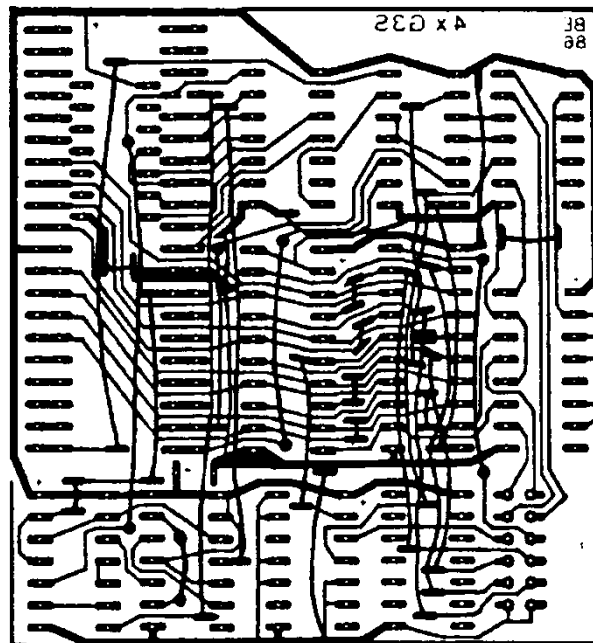
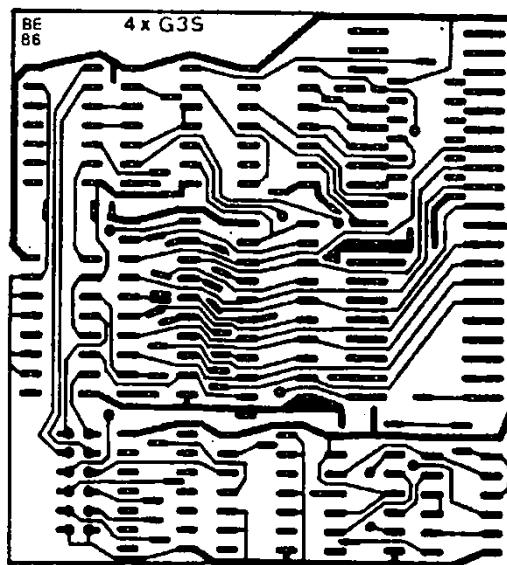
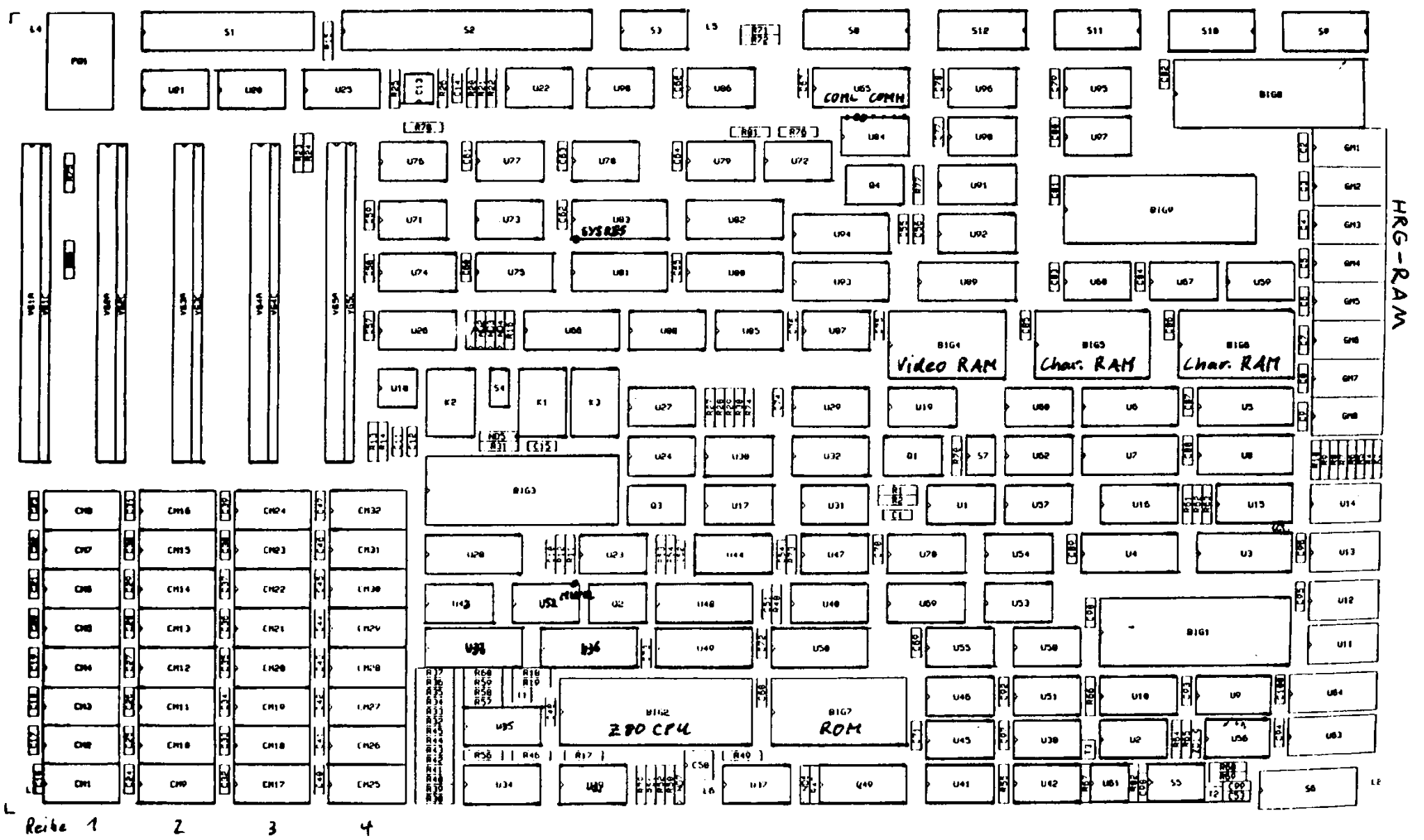


Abb. 9a: Platine des Bankers, Bestückungsseite mit Drahtbrücken (leicht vergrößert)

Abb. 9b: Platine des Bankers, Lötseite





WIKO NG 21-Mar-84 B302MU
 WIK068 [1-1]

Die Mutterplatine des Genie III s

D: Eine nützliche Anwendung für das Bit Q6

In Abschnitt A dieses Artikels müssen wir leider zugeben, daß die vereinfachte und beschleunigte Kopie des Computers Nr. 0 auf seine drei Nachbarn mit Hilfe eines gesetzten Bit 6 im Port F3h leider nicht funktionierte. Auf dieses Feature kann auch verzichtet werden. Es würde die Kopierzeit von ohnehin nur dreieinhalb Sekunden um vielleicht eine halbe Sekunde verringern können. Da man nur einmal, höchstens zweimal am Tag diese Initialisierung vornimmt, ist die Wartezeit nicht so wichtig. Um Q6 aber nicht zu verschenken, haben wir es bis zum Aufspüren des Fehlers einer anderen interessanten Verwendung zugeführt:

Das Boot-EPROM des Genie III s ist ohne Treiber direkt mit der CPU verbunden. Daher ist es grundsätzlich möglich, Daten nicht nur aus dem EPROM zu lesen, sondern es auch damit zu beschreiben. Es gibt dazwischen keinen unidirektionalen Treiber, der diese Datenrichtung verbieten würde. Nun kann freilich in einem ROM nicht schreibenderweise etwas geändert werden, denn es ist eben ein Read-Only-Memory. Es ist jedoch möglich, ein statisches RAM parallel dazu zu stecken. Dazu muß lediglich das WR*-Signal der CPU auf den WE*-Pin des RAMs gelegt werden. Eine einfache Logik trifft die Auswahl zwischen dem EPROM und dem statischen RAM. Die Abb. 10 zeigt die Beschaltung der nunmehr zwei Speicherbausteine anstelle des bisherigen EPROMs, nämlich dieses selbst und ein statisches RAM des Typs 6264.

Ein möglichst kurzes Flachbandkabel sitzt mit einem 28-poligen Quetschstecker im EPROM-Sockel. An seinem anderen Ende ist eine Lochraasterplatine entweder direkt aufgelötet oder mit einer Pfostensteckerverbindung befestigt. Fast alle Signale aus dem Sockel werden auf die korrespondierenden Pins der beiden Bausteine einfach durchgeschleift. Nur das Signal ROMEN* (ROM-enable) muß auf die in der Abbildung beschriebene Weise mit Bit 6 des Ports F3h verknüpft werden, um eine Auswahl zwischen dem EPROM und dem neuen statischen RAM treffen zu können. Außerdem wird MWR* von S2 des Bankers, Pin 5 dem Pin 27 (WE*) des RAMs zugeführt.

Q6 wird an Pin 5 des 74LS274 auf der Lötseite der Banker-Platine abgegriffen und auf einen freien Pin von S2 geführt, so daß sich die Verbindung mit der RAM/EPROM-Bank vereinfacht. Es ist nicht erforderlich, die weiterführende Leiterbahn zum 74LS00 zu durchtrennen und gegen GND zu legen, denn das Kopierfeature funktioniert vorerst sowieso nicht. Es kann also nichts passieren, und nach der Korrektur des Fehlers ist weniger Restaurationsarbeit erforderlich (falls man überhaupt noch restaurieren will, nachdem man die Vorzüge dieser ROM/RAM-Bank erkannt hat).

Der Platz zwischen dem EPROM-Sockel und der Zusatzplatine für den Banker ist so eng, daß zwar das EPROM noch eben dazwischen paßt, der neue Quetschstecker jedoch nicht mehr ohne weiteres. Um die Zusatzplatine dennoch fest und kontaktsicher in den Sockel der CPU stecken zu können, empfiehlt es sich, das Deckelteil des Steckers nach dem Anquetschen des Kabels wieder zu entfernen und die hochstehenden Spitzen der Kontaktklemmen abzuschleifen. Diese werden anschließend mit Isolierband beklebt, damit sie nicht die Lötseite der Zusatzplatine berühren können.

Wer den Banker nicht, wohl aber das parallele RAM neben dem EPROM einbauen möchte, kann sich diese Mühe sparen. Dann steht allerdings selbstredend nicht das Bit 6 des Ports F3h zur Verfügung. Ein Umschalter, der anstelle von Q6 wahlweise GND oder über einen Widerstand von 10 k Ω

+5V an die zugehörigen Eingänge des NAND-Gatters legt, erfüllt denselben Zweck.

Um beim Einschalten oder nach einem Reset auf jeden Fall das Boot-EPROM zu erreichen, bekommt es den Zustand 0 des Bits 6 zugeordnet. Wenn Q6 aber 1 lautet, ist das RAM selektiert.

Der zugängliche Adreßraum dieser neu geschaffenen EPROM/RAM-Bank ist 12 kB groß (0000-2FFFh). Das Original-EPROM des Genie III s (2732) belegt davon nur 4 kB (0000-0FFFh). Es kann einfach durch einen 2764 ersetzt werden, in dessen doppelt großem Speicherplatz interessante Features untergebracht werden können. Der 28-polige Sockel führt auch an den vier bisher nicht benutzten zusätzlichen Pins die dafür erforderlichen Signale. Das statische RAM 6264 belegt von vornherein den doppelten Adreßraum von 8 kB (0000-1FFFh).

Bei 1 MB Hauptspeicher grenzt die vergleichsweise bescheidene Zugabe von 8 kB an Memory-Raffgier, aber bei dem Konzept von vier Computern existiert ohne relativ aufwendiges Banking kein gemeinsamer Speicher, der zur Übergabe von Daten geeignet wäre. Dies ist in der EPROM/RAM-Bank der Fall. Wird nämlich Bit 2 des Ports FAh low gesetzt, dann ist ursprünglich auf jeden Fall im Adreßraum 0-12 kB das EPROM ansprechbar. Mit dieser Modifikation ist es nun, je nach Q6, wahlweise das EPROM oder das RAM. So können die vier Computer dort bei Selektion des RAMs auf bequeme Weise miteinander kommunizieren.

Eine andere Anwendung ist für Tüftler interessant, die das Boot-EPROM gerne nach ihrem eigenen Geschmack ändern möchten. Wenn in seinem Adreßraum auch ein RAM liegt, kann eine Versuchsversion des neuen EPROM-Inhalts dort probegefahren werden, ohne gleich ein EPROM dafür zu opfern. Hier empfiehlt es sich, das Flachkabel nach außen zu verlegen, um beim Einstecken eines neuen EPROMs nicht jedesmal den Computer zerlegen zu müssen. Nun ist es zwar nicht möglich, einfach durch Aufruf mit dem Programmnamen den EPROM-Code ins parallele RAM zu laden und dort zu starten. Das anschließend gelistete Programm besorgt das aber.

Die Datei muß zuvor zwischengepuffert werden, um sie anschließend nach dem Umschalten auf die EPROM/RAM-Bank und der Selektion des RAMs dort ablegen zu können. Die Programm-Laderoutine des G-DOS (Einsprung an 4419h) verfrachtet aber Programme immer gleich an ihre Bestimmungsadresse, was in diesem Falle mit der Fehlermeldung 'Ladeversuch auf ROM-Speicherplatz' quittiert würde. Deshalb muß diese Routine (in Grossers DOS-Buch LOADP genannt) zuvor durch einen Patch geändert werden:

An der Stelle 4C4Dh in SYS0 wird das MSB der Ladeadresse eines Records gelesen. In dem hier vorgestellten Programm wird dort eine Umleitung (Label devia) gezapt, die feststellt, wie dieses MSB lautet. Wenn die Adresse unterhalb 3000h liegt, muß es sich um den Code für das neue EPROM handeln, den wir zunächst puffern wollen. Deshalb wird dann in Z. 41 das Bit 7 des MSB gesetzt, so daß z. B. aus der Adresse 0500h die Adresse 8500h wird. So steht nach dem Laden das Programm ab 8000h im Speicher.

Im ersten Segment der Laderoutine muß nun noch der Patch in SYS0 rückgängig gemacht werden. Schließlich sorgt noch eine Sicherung dafür, daß ein fehlerhaft geladenes Programm nicht die Testbedingungen verfälscht: Bei einem aufgetretenen Fehler ist die NZ-Bedingung erfüllt. In diesem Fall erfolgt einfach unverrichteter Dinge eine Rückkehr ins DOS (Z. 18).

Wenn aber alles geklappt hat, wird zunächst in der EPROM/RAM-Bank das RAM selektiert, indem das Bit 6 des neuen Systembytes 2 (Port F3h) high gesetzt wird. Danach wird über das Systembyte 1 die Bank zugeschaltet. Jetzt ist von 0000-1FFFh das statische RAM zugänglich. Die Zeiger für die Quelle (Puffer ab 8000h), das Ziel (RAM ab 0000h) und der Zähler für die Programmlänge (2000h) werden aufgesetzt und der Code an seinen Bestimmungsort geladen.

Die DOS-Laderoutine legt die Einsprungsadresse eines Programms in 4403h ab. Von dort wird sie nach HL geladen, so daß mit JP (HL) das Programm angesprungen werden kann. Für ein Boot-EPROM wird diese Adresse immer 0000h sein, aber auf diese Weise kann auch ein beliebiges anderes

Programm in der neuen Bank abgearbeitet werden.

Beim Einsprung in das Boot-Programm wird geprüft, ob die F1-Taste gedrückt ist, um anstelle eines Boot im Monitor zu bleiben. Das Laden des Programms dauert je nach Größe lange genug, um nach dem Aufruf der Laderoutine rechtzeitig diese Taste zu erreichen. So können sowohl die Bootfähigkeit des neuen EPROM-Inhalts als auch der vielleicht modifizierte Monitor getestet werden.

Der Loader benutzt hier für den neuen EPROM-Inhalt den Filenamen BOOTEPRH/CMD. Das ist natürlich willkürlich. Der Name kann beliebig dem eigenen Geschmack angepaßt werden. Es muß auch nicht unbedingt ein Boot-Programm sein. Diese Routine lädt, was sie erwischt, und springt sie dann an.

Falls Schwierigkeiten beim Einbau oder Betrieb des Statt-ROM-RAMs auftreten (auch solo ohne den Banker), können beide Autoren weiterhelfen.

Abb. 10: Schaltung für die EPROM/RAM-Bank

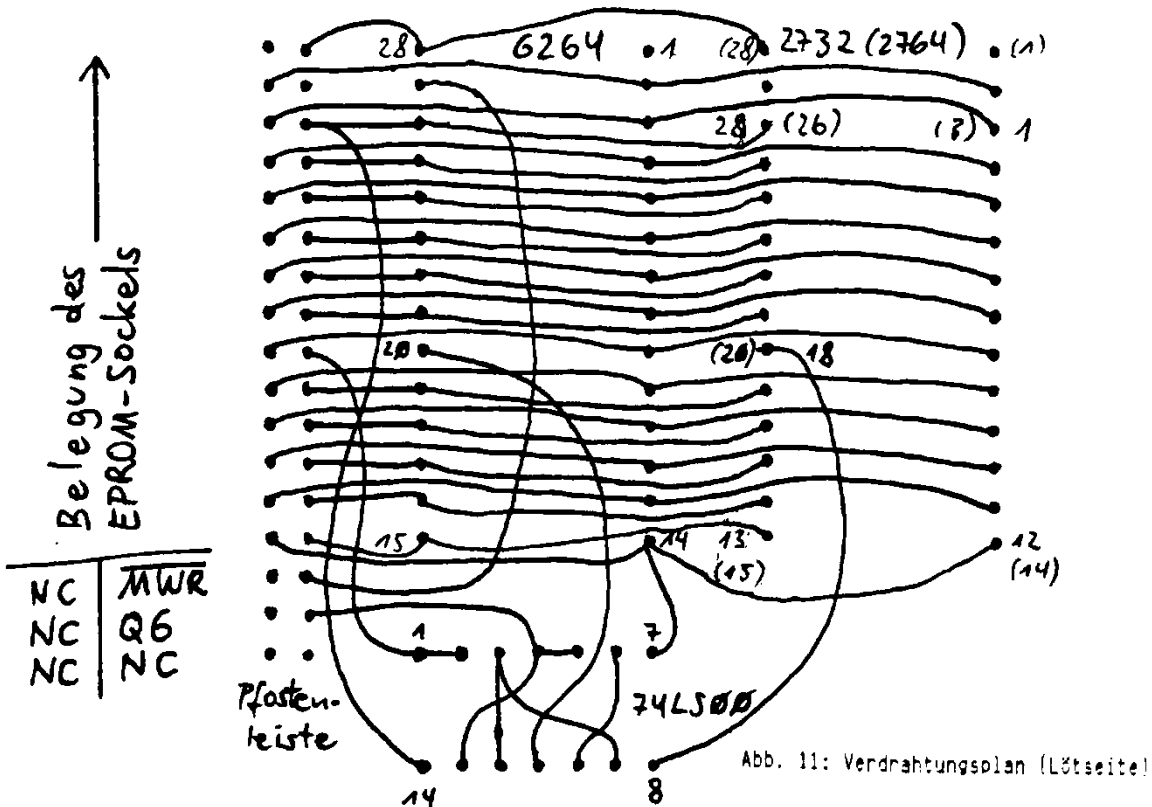
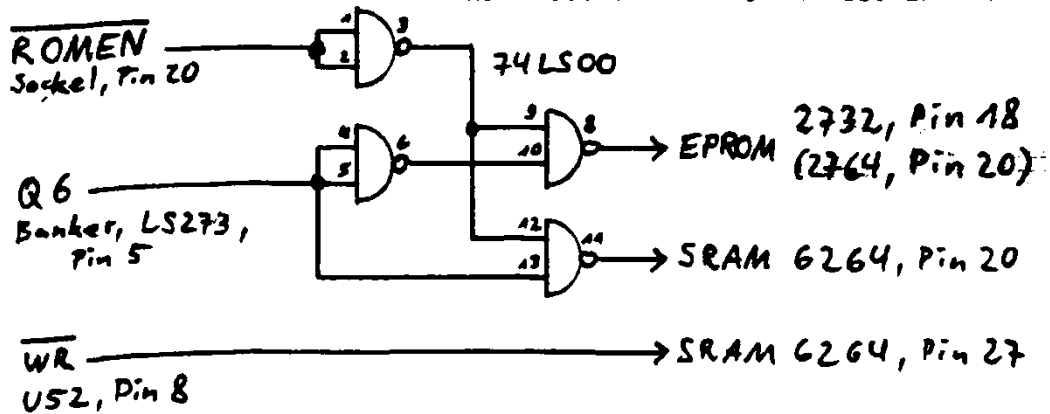


Abb. 11: Verdrahtungsplan (Lötseite)

```

00001 ;*****
00002 ; Routine zum Laden von Programmen, die im statischen RAM
00003 ;      neben dem Boot-EPROM (0000-1FFF) laufen sollen
00004 ;
00005 ;                      (C) 1986 by Arnulf Sopp
00006 ;*****
00007
3000      00008          ORG      3000h
00009
00010 ;Umleitung im UP LOADP (Programm laden) in SYS0 setzen,
00011 ;Programm BOOTEPRH/CMD nach 8000 verschoben laden
3000 212C30 00012 start  LD      HL,devia      ;Umleitung zur Modifikat.
3003 224E4C 00013          LD      (4c4eh),HL      ;in LOADP patchen
3006 213530 00014          LD      HL,doscall      ;Zeiger auf DOS-Befehl
3009 CD1944 00015          CALL     4419h      ;diesen ausführen
300C 21654C 00016          LD      HL,4c65h      ;alten Code in LOADP
300F 224E4C 00017          LD      (4c4eh),HL      ;restaurieren
3012 C0      00018          RET      NZ      ;Ende, falls Fehler
00019
00020 ;EPROM/RAM-Bank und dort wiederum das RAM selektieren
3013 F3      00021          DI      ;es wird brenzlich
3014 DBF3    00022          IN      A,(0f3h)      ;Systembyte 2
3016 F640    00023          OR      40h      ;RAM statt EPROM
3018 D3F3    00024          OUT     (0f3h),A      ;selektieren
301A 3E00    00025          LD      A,0c0h      ;auf RAM/EPROM-Bank
301C D3FA    00026          OUT     (0fah),A      ;Bank freigeben
00027
00028 ;Prg. von 8000 nach 0000 im Boot-RAM laden und anspringen
301E 210080 00029          LD      HL,8000h      ;Puffer des Boot-Progr.
3021 55      00030          LD      D,L      ;DE (- 0000
3022 5D      00031          LD      E,L      ;(= Zieladresse)
3023 010020 00032          LD      BC,2000h      ;Programmlänge
3026 ED80    00033          LDIR      ;Programm ins Boot-RAM
3028 2A0344 00034          LD      HL,(4403h)      ;Puffer der Startadresse
302B E9      00035          JP      (HL)      ;Programm anspringen
00036
00037 ;Patch in LOADP: Code bis 3000 soll nach 8000 ff. kommen
302C CD654C 00038 devia  CALL     4c65h      ;MSB Ladeadresse holen
302F FE30    00039          CP      30h      ;zum Boot-Progr. gehörig?
3031 D0      00040          RET      NC      ;norm. weiter, falls nein
3032 F680    00041          OR      80h      ;ab 8000 laden
3034 C9      00042          RET      ;normal weiter
00043
00044 ;DOS-Befehl zum Laden des neuen Boot-Programms
3035 4C      00045 doscall DM      'LOAD,BOOTEPRH/CMD',0dh ;DOS-Bef., ENTER
00046
3000      00047          END      start      ;dort Einsprung

00000 Fehler

devia 302C      doscall 3035      start 3000

```