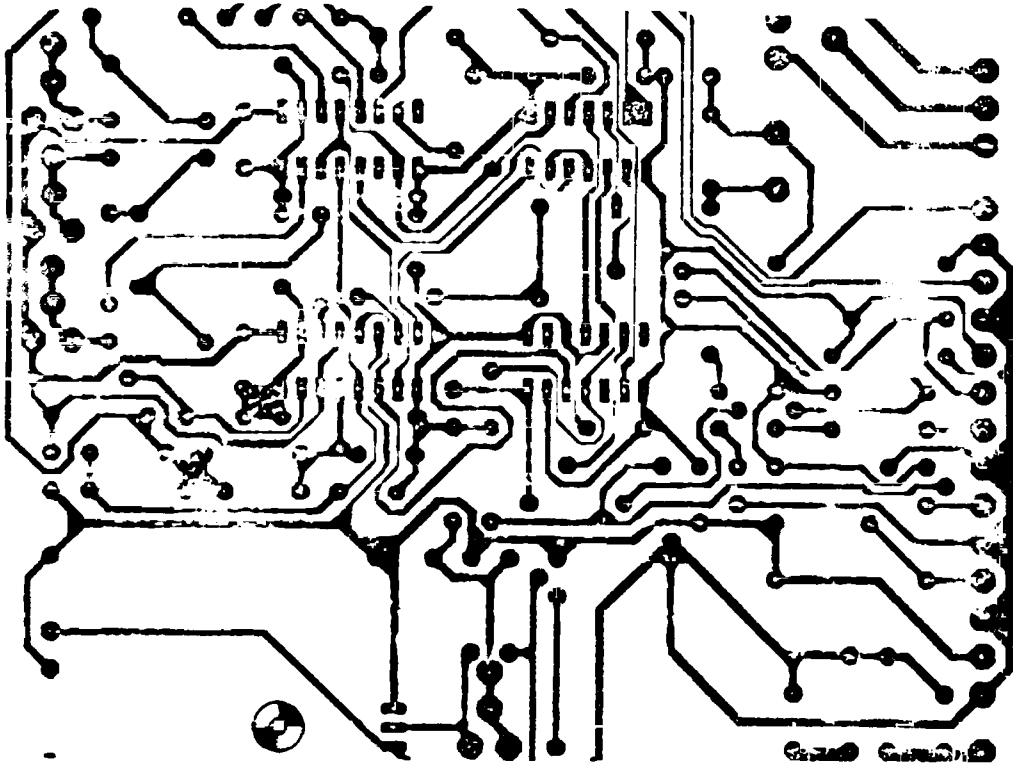


G E N T I E USER
CLUE
und Colour-Genie
USER
CLUE **T R E S I O**
B R E M E R H A V E N

HARDWARE

ANZEIGEN



3. JAHRGANG | 1985

Red.: Peter Spieß,
* Sortiert von: E

Inhalt

1-5	Wie aus dem Komtek ein Computer wird v. H. Bernhardt
6-10	Hardware-Umschaltung für CP/M v. H. Bernhardt
11-12	Bus-Terminierung v. P. Spieß
13-16	Ver-16-facher v. H. Bernhardt
17-22	Die Software dazu v. A. Sopp

Dieses Sonderheft ist unseren Mitgliedern Arnulf Sopp und Helmut Bernhardt gewidmet. Die selbstlose Mitarbeit im Club machte diesen Sonderdruck möglich.

Wie aus dem Komtek 1 ein Computer wird

Helmut Bernhardt (Tel. 0431/241907)

"Der Komtek 1 ist ein TRS 80-kompatibler Computer." Wie weit diese Aussage wirklich stimmt, soll im folgenden untersucht werden. Der dabei nötige Aufwand und die Notwendigkeit, dies alles hier wiederzugeben, wären garrnicht vorhanden, wenn der Hersteller eine Dokumentation, wie man sie bei TRS 80 und GENIE gewohnt ist, zur Verfügung stellen würde. Die hier zusammengetragenen Erkenntnisse mußten durch mühevolltes Verfolgen von Leiterbahnen erst herausgewickelt werden, um einige Hardware-Erweiterungen, die beim TRS 80 und GENIE nach dem Schaltplan angeschlossen werden können, auch hier zu nutzen.

Außerlich fällt an Komtek 1 zunächst auf, daß auf dem Bildschirm anstelle der deutschen Umlaute die entsprechenden amerikanischen Zeichen mit gleichem ASCII-Code erscheinen; ein Umstand, der bei der Textverarbeitung recht lästig ist. Dafür enthält die Tastatur aber ein paar Tasten mehr, so daß es möglich ist, durch Umstecken der Tastenkappen und Änderungen in der Matrix eine Schreibmaschinen-ähnliche Tastatur daraus zu machen.

Die Tasten BREAK und CLEAR müssen dabei aus dem Tastenfeld herausgenommen werden und durch Taster woanders ersetzt werden. Die ENTER-Taste ist eine Zeile höher angeordnet und die Links- und Rechts-Pfeil-Tasten (darunter) haben die Bezeichnungen BACKSPACE und RUBOUT. Eine RESET-Taste, die tatsächlich einen RESET am Z80 bewirkt, liegt in der Tastatur und führt gerne zu Katastrophen.

Auf der Rückseite sind 3 Cardedge-Stecker erreichbar, die den Anschluß von Drucker und Floppy ermöglichen. Der mittlere Stecker führt den Systembus nach außen. Weiter sind bei einer etwas teureren Version noch Anschlüsse für Sensoren und Schalter vorhanden.

Der Systembus enthält fast alle Signale des Z80. Etwas großzügig wurden außer den Signalen MREQ, IORQ, RD# und WR# auch noch die Signale MAR# und MRD# herausgeführt. Die Signale IN# und OUT# des TRS 80 waren sicher brauchbarer gewesen, zumal eine Speichererweiterung über den Systembus nicht vorgesehen ist.

Die Tacklen des Objektes zeigen sich aber erst auf der Platine. Man vermißt bei allen Z80-Signalen Treiber ICs. Der Z80 muß mit seinen Ausgängen den ganzen Computer mit allen eventuellen Erweiterungen treiben. Der Hardware-Freak muß sich hier also zurückhalten und wirklich nur das Allernötigste an Erweiterungen anschließen.

Erfreulich ist die Verwendung von 2732-EPROMs als ROM und eines 2716-EPROMs als Sonder-ROM und das Vorhandensein von 16 Steckplätzen für dynamische RAMs, so daß in früherer Denkweise auf dem CPU-Board 32K RAM verfügbar sind und nach heutigem Standard hier 128K RAM (8,5M Byte) untergebracht werden können (, wenn man sich eine Banking-Logik baut).

Der Stecker für den Drucker und die Sensoren und Schalter sind direkt an die Ports eines 8255-Chips angeschlossen. Es ist also zu erwarten, daß die Länge des Druckerkabels kritisch ist.

Neben dem Floppy-Anschluß befindet sich ein 40-poliger, 2-reihiger Pfostenstecker, der nach näherer Untersuchung offensichtlich den Anschluß einer Karte mit Floppy-Controller und weiteren 16K RAM ermöglicht.

Bei der Untersuchung der Adreßdecodierung auf dem Board wurden folgende Anomalitäten festgestellt:

1) Die Freigabe des 8255 (Drucker) wird anstelle von 4 I/O-Adressen durch die Speicheradressen 36X0H bis 36X3H erreicht (wobei das Digit X alle möglichen Werte haben darf). Der Druckertreiber im ROM muß also gegenüber dem TRS 80 (Adressen 37E8H bis 37EBH) oder dem GENIE (zusätzlich Port F0H) geändert sein. Programme mit eigenem Druckertreiber müssen versagen.

2) Die Anwahl des Disk-Laufwerks erfolgt über alle Adressen 37X0H bis 37X7H (Digit X = don't care). Der Floppy-Controller ist entsprechend unter den Adressen 37XCH bis 37XFH erreichbar. Wenn hier auch durch unvollständige Decodierung für Drucker und Floppy 1/2 K Byte Adreßraum verschenkt wurden, so ist immerhin beim Floppy-Betrieb Kompatibilität gewährleistet.

Da von einer Mainzer Firma auf die Bitte, Prospekte über Hardware-Erweiterungen zu übersenden, seit Monaten keine Reaktion erfolgte, mußte folgender Weg eingeschlagen werden:



Die 4 EPROMs wurden entfernt und gegen solche ausgetauscht, die das 12K-Level II-BASIC des TRS 80 und den Inhalt des Sonder-ROMs des GENIE enthielten.

In den Sockel des Z80 wurde eine Huckepackplatine gesteckt, die den Z80 und Treiber-Bausteine für alle Bussignale enthielt. Bei der Richtungssteuerung des Datenbustransceivers wurde JM2-Fähigkeit berücksichtigt.

Der 8255 wurde entfernt. (Es ist aber prinzipiell möglich, durch eine andere Decodierung der Signale CE*, WR* und RD* diesen Baustein anderweitig zu nutzen.)

Sämtliche ROM-ICs 4116 wurden entfernt. Stattdessen wurde eine Reihe (8 Sockel) mit 4164er ROMs bestückt, wobei entsprechende Änderungen (wie bei TRS 80 und GENIE) nötig waren.

Die Freigabe des Sonder-ROMs wurde auf den Adreßraum 3000H-37DFH erweitert.

Es wurde ein Expansion-Interface (EXP) von RB Elektronik ergebaut.

Die Tastatur wurde auf Umlaute erweitert und der deutschen Schreibmaschinentastatur angenähert.

Die Schaltung des CPU-Boards (Auszug mit den wesentlichen Bestandteilen für Umbauten und Erweiterungen)

Für die folgenden Schaltpläne und Erläuterungen gilt die Durchnummerierung der ICs, wie im beiliegenden Bestückungsplan, dessen Einschränkungen in der Genauigkeit der Lage der einzelnen Bauteile man bitte entschuldige (ein offizieller Plan liegt leider nicht vor).

Über die Umständlichkeit der Herleitung mancher Signale läßt sich streiten. Positiv zu werten ist aber die Tastaturfreigabe durch das voll ausdecodierte Signal 3800H-38FFH*. Hier läßt sich der Adreßraum 3900H - 38FFH ohne Änderung der Tastaturfreigabe für Anwenderschaltungen nutzen.

Nicht dokumentiert ist in den Schaltplänen, daß der RESET-Knopf tatsächlich einen RESET (und nicht wie bei TRS 80 und GENIE einen NMI) auslöst. Um einen NMI zum erneuten Booten des DOS ohne Löschen des Speichers zu bewirken, muß der Knopf an der linken Seite zweimal gedrückt werden. Dieser Knopf schaltet in der einen Stellung einen 10ms-Takt an den NMI-Eingang des Z80. In der anderen Stellung ist der Schalter offen. Das Booten besteht also in einer wilden Drückerei dieser beiden Knöpfe. Wenn man das erst einmal weiß, kann man zur Not damit leben. Es geht immerhin.

Wenn der Jumper J (in der Nähe des Z80) geändert wird, gelangt der 10ms-Takt an den INT*-Eingang des Z80. Es stehen dann regelmäßige Interrupts ohne Vorhandensein eines Expansion-Interface zur Verfügung und ein NMI ist nicht mehr möglich.

Treiber für die CPU-Signale

Da von vornherein der Einbau von etwas mehr Hardware vorgesehen war, wurden alle aktiven Z80-Signale gepuffert, ohne erst zu probieren, ob der Z80 eventuell auch selbst die nötige Treiberleistung aufbringt.

Die gangbarste Möglichkeit, solches nachträglich noch zu bewirken, besteht darin, daß in den Sockel des Z80 ein kleines Huckepack-Board gesteckt wird (wie die Doubler-Platine von Floppy-Controllern), auf dem sowohl der Z80 als auch die Treiber-ICs Platz finden. Auf diesem Board werden alle aktiven Signale des Z80 über Treiber an den Stecker geliefert. Durch die Richtungssteuerung des Datenbustransceivers 74LS245 per AND-Verknüpfung von RD* und M1* bleibt weiterhin der Interrupt-Mode 2 möglich. Layout und Bestückungsplan des Treiberboards sind in Abb.3 dargestellt.

Anschluß eines Expansion-Interface (EXP)

Wegen der schon erwähnten Vertriebsagilität einer Mainzer Lieferfirma mußte auf ein Floppy-Interface (EXP) von RB Elektronik, Eitorf zurückgegriffen werden. In der mitgelieferten Einbauanleitung fällt allerdings nirgendwo der Name "Kontek", so daß der Einbau nicht so ganz einfach ist. Nach einigen Änderungen am CPU-Board war das ganze Projekt dann aber noch eleganter, als es sich beim TRS 80 und GENIE gestaltet.

Für den Anschluß wurde die zweireihige, 40-polige Stiftleiste neben dem Floppy-Stecker auf der Rückseite benutzt, da dieser schon die meisten dafür nötigen Signale (Tabelle 1) liefert. Die nicht benötigten Signale wurden per Durchtrennen von Leiterbahnen abgetrennt und dafür die Stifte mit entsprechenden nicht vorhandenen aber notwendigen Signalen belegt. In der Tabelle 1 sind diese neuen Signale neben den abgetrennten (eingeklammerten) mit aufgeführt.

Der Umbau führte zu folgender Tastenanordnung:

```
..... 0 8 =/- */: Affe
... 0 P u ;/+ RETURN
..... L 0 4 BACK RUBOUT
..... >/. ?/ SHIFT
```

Die Tastenkappen für die Tasten (A), (B), (ü) und (R) mußten anderweitig aufgetrieben werden. Der Rest konnte durch Umstecken der Tastenkappen erreicht werden. Die Funktionen BREAK und CLEAR wurden auf zusätzliche Taster außerhalb des Tastenfeldes verlegt. Die Anschlüsse des Tasters (BREAK) wurden mit dem linken Lötunkt der Taste (Z) und dem linken Lötunkt der Taste (UP ARROW) verbunden. Der Taster (CLEAR) wurde mit dem rechten Punkt von Taste (J) und dem linken Punkt von (UP ARROW) verbunden.

(Gemäß der neuen Tastenanordnung) wurden folgende Leiterbahnen auf dem Tastaturboard durchtrennt:

- alle 3 Leiterbahnen an die Lötunkte der Taste (Affe) werden neben den Punkten durchtrennt
- die Leiterbahn zum linken Punkt der Taste (*/:) wird durchtrennt
- alle 3 Leiterbahnen zur Taste (ß) werden durchtrennt
- bei der Taste (</>) werden die zu (RETURN) und zu (*/:) führenden Leitungen am linken Punkt und beide Leitungen zum rechten Punkt durchtrennt
- alle 3 zur Taste (ü) führenden Leitungen werden durchtrennt
- die Leiterbahnen zu den Punkten der Taste (A) werden durchtrennt
- die Leiterbahnen zu den Punkten der Taste (B) (0_n, nicht Null) werden durchtrennt

Nun werden folgende Verbindungen hergestellt:

- linker Punkt von (Affe) mit linkem Punkt von (P) und rechtem Punkt von (N)
- rechter Punkt von (Affe) mit Durchkontaktierung zwischen den Punkten von (D) (wieder 0_n, nicht Null)
- linker Punkt von (R) mit rechtem Punkt von (J)
- rechter Punkt von (R) mit rechtem Punkt von (=/-) und rechtem Punkt von (9)
- linker Punkt von ß mit linkem Punkt von (RUBOUT)
- linker Punkt von (</>) mit linkem Punkt von ∞ und rechtem Punkt von (R)
- rechter Punkt von (</>) mit linkem Punkt von (B)
- linker Punkt von (ü) mit rechtem Punkt von (U)
- die rechten Punkte von (A), (S), (ü) und (ß) mit der Durchkontaktierung zwischen den Tasten (D) und (E)
- rechter Punkt von (RUBOUT) mit linkem Punkt von (BACKSPACE)

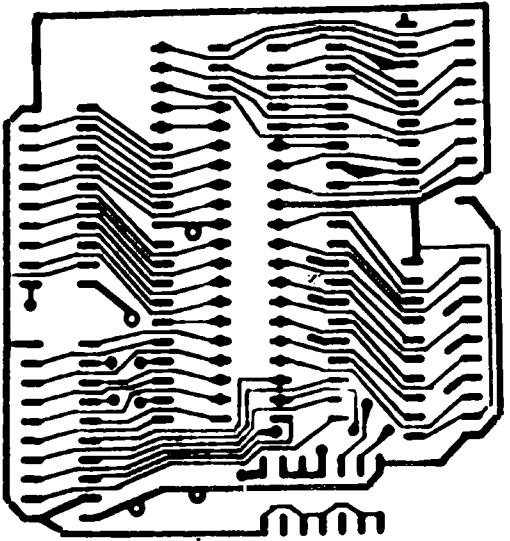
Zum Austausch der Positionen der Tasten (Y) und (Z) werden die Tastenkappen ausgewechselt und folgende Änderungen vorgenommen:

- die beiden zum rechten Punkt der (neuen) Taste (Y) führenden Leitungen werden von dem Punkt abgetrennt und die abgetrennten Enden miteinander verbunden
- der rechte Punkt von (Y) wird mit dem rechten Punkt von (A) verbunden
- die Leiterbahn zum linken Punkt der (neuen) Taste (Z) wird durchtrennt
- der linke Punkt von (Z) wird mit dem rechten Punkt von (R) verbunden

Diese uppigere Tastatur wird von den Tastatortreibern des Level II-BASIC und des DOS unterstützt. Auf dem Bildschirm erscheinen leider weiterhin amerikanische Sonderzeichen anstelle der deutschen Umlaute, über den Drucker erhält man (be: entsprechender Zeichensatzwahl) aber deutsche Umlaute.

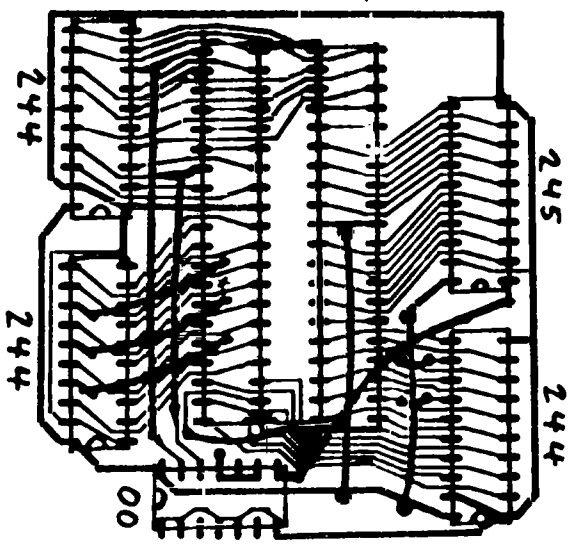
Einige Textverarbeitungsprogramme mit eigenem Tastatortreiber spielen nicht mit. Bei NEWSSCRIPT mußte z.E. an Ende des Files NS/CHD die Tastenbelegungstabelle entsprechend gepatched werden. Hier muß auch der Taste (UP ARROW) der Wert 7FH zugewiesen werden (bisher 'A' kurz hinter der Tabelle).





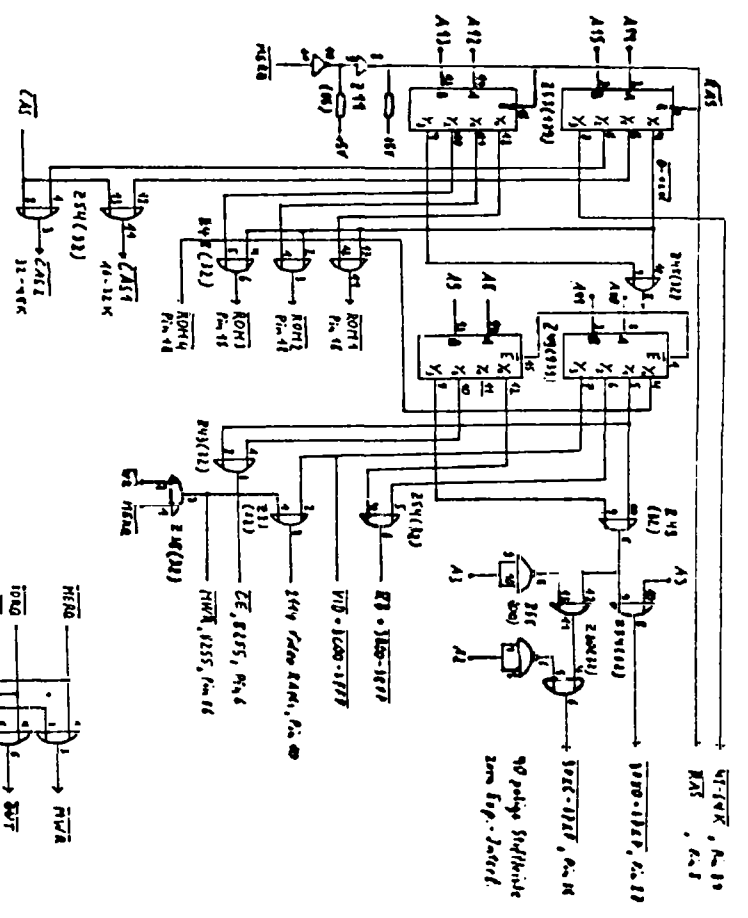
Stecker
(Lötseite)

Z 80

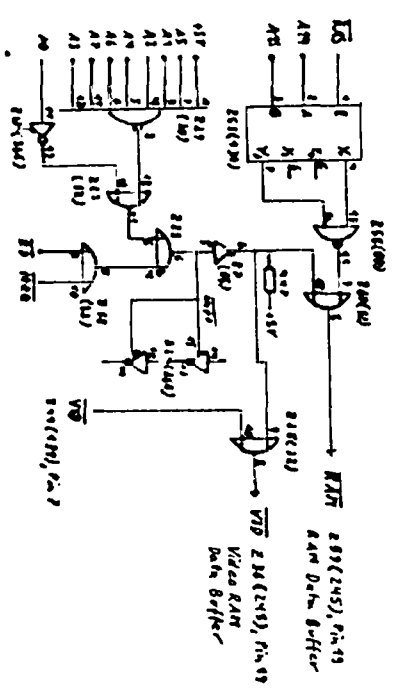
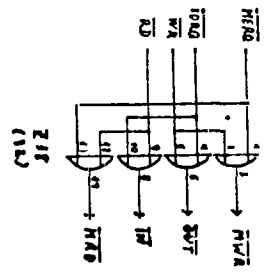


3) Das Treiberboard: Layout, Bestückung und Drahtbrücken

5



Address and Data Bus Connections



An E⁺ wurde das low aktive Freigabesignal OUT DFH⁺ gelegt, während an CLEAR ein aus RESET⁺, IN DFH⁺ und einem von Zustand des Ausgangs D7 gesteuertes NM1⁺ abgeleitetes Signal gelegt ist. Die Selektion des jeweiligen Latches wird durch Anlegen der Datenleitungen D0, D1 und D2 an die Eingänge A, B und C erreicht, während D3 an DATA gelegt den Zustand des durch E⁺ freigegebenen Latch-Ausgangs bestimmt.

So läßt sich durch Ausgabe eines Datenwortes, bei dem nur das untere Nibble signifikant ist, an den Port DFH ein durch die Pegel von D6 bis D2 adressiertes Latch mit dem Zustand von D3 laden.

Durch Lesen des Ports DFH, RESET⁺ oder einen nicht per OUT (DFH), DFH gesperrten NM1⁺ können alle Ausgänge auf low gesetzt werden.

Diese Ausgänge werden mit den jeweiligen high aktiven Speicherbereichs-Freigabesignalen und bei den beiden ROM-Bereichen zusätzlich noch jeweils einmal mit RD⁺ und mit WR⁺ NAND-verknüpft. Sie ergeben dann an Ausgang des jeweiligen NAND-Gatters ein low aktives Signal, wenn eine in diesem Bereich liegende Speicheradresse auf dem Bus liegt, wenn der steuernde Latchausgang high gesetzt ist und bei den ROM-Bereichen das Signal RD⁺ bzw. WR⁺ nicht aktiv ist.

über AND-Verknüpfung aller NAND-Ausgänge wird das Signal PHANTOM⁺ erhalten, das immer dann low wird, wenn ein NAND-Ausgang low wird.

Die zusätzliche BIGMEM-Funktion dieses Boards wird durch Invertieren der Adressen A14 und A15 zu A14' und A15' und der Herleitung eines 8-16K⁺ \bar{A} -Signals daraus erhalten. Ob die Adressen invertiert werden oder nicht, wird durch den Ausgang eines Flip Flops gesteuert, dessen Zustand wiederum durch die decodierten Portsignale IN DEN⁺, OUT DEN⁺, RESET⁺ und NM1⁺ bestimmt wird.

Durch RESET⁺, NM1⁺ und IN DEN⁺ wird das Flip Flop so eingestellt, daß die Adressen über die XOR-Gatter nicht invertiert werden. OUT DEN⁺ kippt das Flip Flop um, so daß die Adressen invertiert werden.

Um den beim TRS 80 und beim Komtek 1 gegebenen Voraussetzungen (kein PHANTOM⁺-Eingang auf dem Bus) Rechnung zu tragen, wurden nun auch beim GENIE nicht die Signale A14' und A15' an den Adreßdecoder geführt, der das Signal 0-16K⁺ herleitet. Es wurde auf dem Board ein Signal 0-16K⁺ aus A14', A15' und MERQ⁺ hergeleitet und mit dem Signal PHANTOM⁺ verknüpft. Dieses dabei erhaltene Signal 0-16K⁺ muß nun auf dem CPU-Board anstelle des ursprünglichen Signals 0-16K⁺ verwendet werden.

Auswahl der Funktionen des Boards über Drahtbrücken und Jumper

Grundversion, nur EG64MBA-Funktionen
(die ICs 13 bis 15 brauchen nicht bestückt zu werden)

Punkte neben IC 7 : M mit N verbinden
Punkte unterhalb IC 9 : T mit U und U mit W verbinden
N (über IC 7) mit M' (über IC 9) verbinden

Erweiterte Version, EG64MBA- und BIGMEM-Funktionen
(ICs 13 bis 15 müssen bestückt werden)

Punkte neben IC 7 : M mit B verbinden
Punkte unterhalb IC 9: folgende Verbindungen mit den Punkten unter IC 15 herstellen:
T mit T', U mit U', V mit V' und W mit W'
Anstelle von N (über IC 7) wird J (über IC 7) mit M' verbunden
Punkt X (neben IC 3) und Y (neben IC 15) miteinander verbinden

Allgemeine Drahtbrücken und Jumper für alle Rechner

Zwischen den unteren beiden IC-Reihen müssen folgende Punkte miteinander verbunden werden:

1 mit 1', 2 mit 2', 3 mit 3', 4 mit 4', 5 mit 5', 6 mit 6'

Zwischen IC 1 und IC 2 kann durch Wahl der Verbindungen A-C oder B-C die Einteilung der Grade der umschaltbaren Bereiche des Sonder-ROMs und des Drucker/Floppy-Bereichs eingestellt werden:

Verbdg. A-C: Sonder-ROM 3000H - 35FFH
Drucker/Floppy 3600H - 37FFH
Verbdg. B-C: Sonder-ROM 3000H - 370FH
Drucker/Floppy 37E0H - 37FFH

Die Variante B-C hat bei GENIE und Kontek nur dann Sinn, wenn der Sonder-ROM nachträglich voll decodiert wurde und beim Kontek die Druckerschnittstelle (B255 auf den Adressen 36X0H-36X3H) geändert wurde.
Anstelle einer festen Einstellung durch Jumper kann auch ein Umschalter eingesetzt werden.
Bei GENIE und Kontek werden außerdem noch die Verbindungen F - F', G - G', S - O, Q - R, P - P' hergestellt; die Punkte D und E werden nicht miteinander verbunden.

Änderungen für den Anschluß an den TRS 80

- 1) Die Drahtbrücken S-O und Q-R werden nicht gelegt.
- 2) Die Leiterbahn von Pin 5 von IC 9 zu Pin 4 von IC 4 wird durchtrennt.
- 3) Die Drahtbrücken F-F', G-G' und P-P' werden nicht gelegt.
- 4) Die Punkte D und E neben IC 1 werden miteinander verbunden.
- 5) Die Punkte S und R werden miteinander verbunden.
- 6) Anstelle von RD⁰ und WR⁰ werden IN⁰ und OUT⁰ an die Anschlußleiste des Boards geführt.
- 7) Das Signal MRD⁰ des Systembus wird an Pin 2 von IC 5 und Pin 4 von IC 10 gelegt.
- 8) Das Signal MWR⁰ wird an Pin 4 von IC 4 und Pin 5 von IC 10 gelegt.
- 9) Pin 6 von IC 10 wird mit Pin 9 von IC 9 verbunden.
- 10) Das Signal SYSRES⁰ des Systembus wird an die Anschlußpunkte RESET⁰ und NM1⁰ und Punkt P' oberhalb von IC 14 geführt. Damit besteht dann nicht die Möglichkeit, das Zurückschalten in den Ausgangszustand durch NM1⁰ per OUT DFH,DFH zu unterbinden.

Wenn dies gewünscht wird, müssen die Signale RESET⁰ und NM1⁰ anstelle von SYSRES⁰ über Drahtverbindungen von der Pias 26 und 17 des 280 an die entsprechenden Punkte auf der Anschlußleiste geführt werden. SYSRES⁰ muß weiterhin an P' geführt werden (das berührt nicht die witer den Punkten 3) und 7) bis 9) getroffenen Aussagen).

Berücksichtigung bestimmter Systemkomponenten

Bei Vorhandensein eines Original-Expansion-Interface bei TRS 80 und GENIE (EB 3014) müssen (in der voll ausgebauten Version dieses Switchboards) zum Expansion-Interface die Signale A14' und A15' anstelle von A14 und A15 geführt werden. Das Gleiche gilt auch für GENIE-Floppy-Laufwerke, die über den 50-poligen Systembus angeschlossen sind (Controller im Laufwerksgehäuse).

Außerdem hat das Signal PHMFTM⁰ bei der AdreBdecodierung des Drucker/Floppy-Bereichs bei diesen Interfaces keinen Einfluß. Um auch hier ein Ausblenden des I/O-Bereichs 37E0H-37FFH zu ermöglichen, müssen anstelle der Signale MERO⁰ beim GENIE bzw. MWR⁰ und MRD⁰ beim TRS 80 die mit Q7 (Pin 9 von IC11, 74LS259) OR-verknüpften Signale an das Expansion-Interface bzw. den Floppy-Controller im Laufwerksgehäuse weitergeleitet werden.

Dafür können die noch freien OR-Gatter in IC16, 74LS32, Pias 4, 5 und 6 sowie Pias 10, 9 und 8 verwendet werden. Die Änderung muß über freie Verdrahtung auf der Lötseite erfolgen.

Bei Verwendung eines EXP1-Interfaces von RB Elektronik in allen 3 Rechnern oder beim Kontek I-Floppy-Interface (dessen RAMs wegen der 4164-RAMs auf dem CPU-Board nicht bestückt sein dürfen) sind keine zusätzlichen Maßnahmen erforderlich.

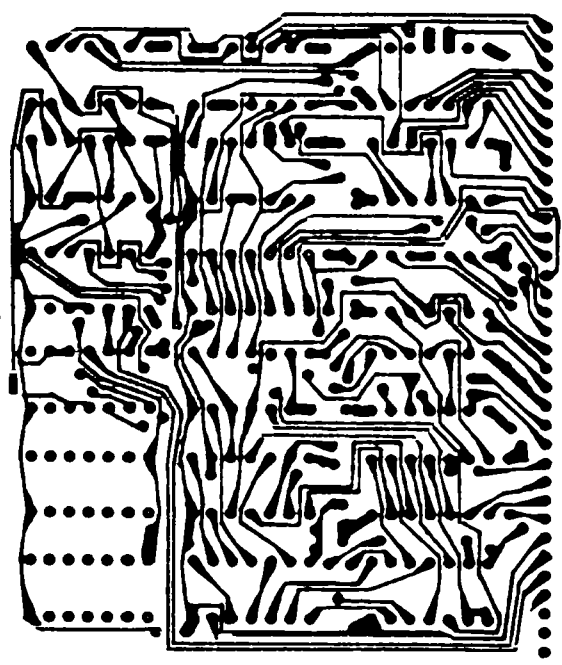
Änderungen auf dem CPU-Board

Um im Bedarfsfall auch ohne dieses Board arbeiten zu können, ist es sinnvoll, zwischen den Signalen 0-16K⁰ vom CPU-Board und PD-16K⁰ auf diesem Board umschalten zu können. Dafür wird an dem Mittelanschluß eines Umschalters α e abgetrennte Leitung gelegt, auf die vorher das Signal 0-16K⁰ gelangte, während an die beiden äußeren Anschlüsse die Signale 0-16K⁰ und PD-16⁰ gelegt werden.

Beim GENIE liegt 0-16K⁰ an Pin 12 von 2 25 (74LS139) und beim Kontek 1 an Pin 4 von IC 53 (74LS139).

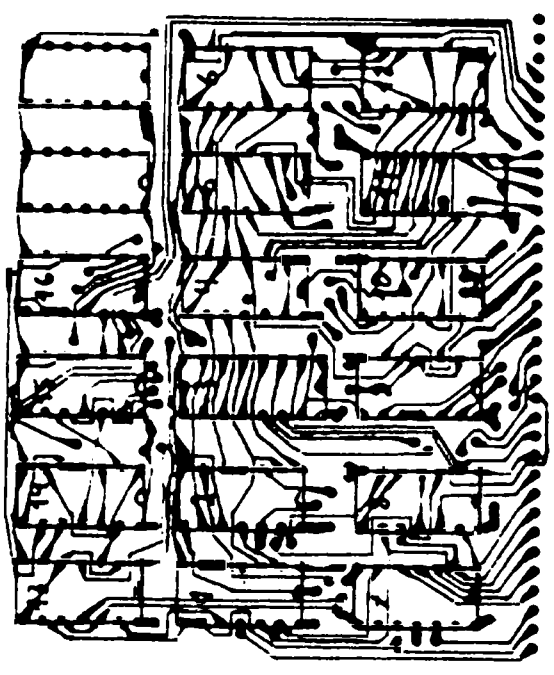
Der TRS 80 kommt dagegen ohne ein Signal 0-16K⁰ aus. Erst bei der Umrüstung auf 64K RAM⁰ auf dem CPU-Board wird ein solches Signal erzeugt. Damit TRS 80 Anwender dieses Board auch nutzen können, wird zum Schluß noch kurz der Umbau auf 64K auf dem CPU-Board beschrieben.

3



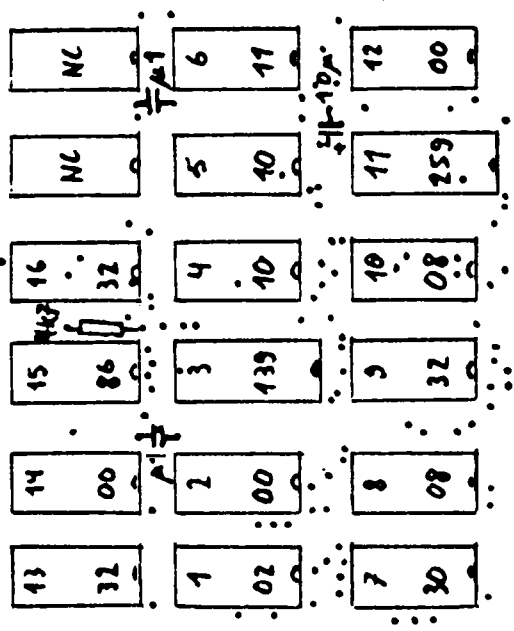
.....

4

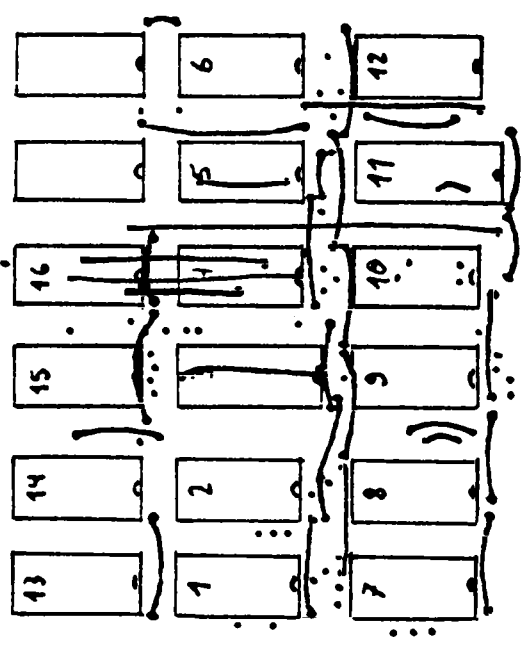


.....

5



6

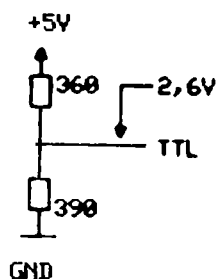


Bus - Terminierung

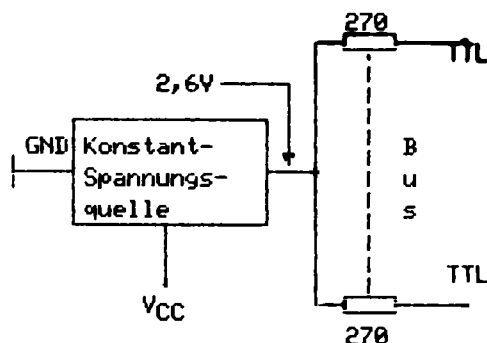
Bei langen Flachbandleitungen, die als Verlängerung des Bus-Anschlusses dienen, stellt sich oft ein Problem ein, dem vermeindlich nicht beizukommen ist; der Rechner spielt plötzlich verrückt. Ich stand vor den selben Ungereimtheiten, als ich mein Genie II in ein Genie III-Leergehäuse umgesiedelt hatte. Da ich Erweiterungen, wie den EG 64 MBA oder meine Grafikkarte (512 * 512 Bildpunkte) auch in Zukunft benutzen wollte, mußte ich den 50-poligen Systembus nach außen führen. Ein arbeiten mit den genannten Erweiterungen war aber nun nicht mehr möglich.

Terminierung ist das Stichwort. Durch lange Flachbandleitungen oder große Bussysteme können unerwünschte Übersprecher zwischen den einzelnen Busleitungen und lästige Leitungskapazitäten auftreten. Wenn dann der Systemtakt auch noch recht hoch gewählt wurde, ist das ohnehin schon kritische Timing fast nicht mehr beherrschbar.

Zur Auswahl stehen zwei unterschiedliche Verfahren, aktive und passive Terminierung. Das Endresultat ist aber das gleiche. Die einfachere Möglichkeit besteht aus zwei in Serie zwischen +5V und Masse geschalteten 360 Ohm und 390 Ohm Widerständen. Dies ergibt am Abgriff der TTL-Leitung 2,6V und erlaubt einem Standard-TTL-Ausgang definierte Logikpegel bei niedriger Leitungsimpedanz. Dadurch wird die Gefahr des Übersprechens und Signalverzerrungen über lange Leitungswege gemindert. Die Methode hat aber auch Nachteile: Jede TTL-Terminierung hat eine Stromaufnahme von 6,7mA. Demnach wird das Netzteil des Computers zusätzlich mit ca. einem viertel Ampere belastet, je nach Anzahl der terminierten Leitungen. Desweiteren ist auch das thermische Problem nicht ganz unbedeutend (Bild 1).



Ein besseres Verfahren stellt die aktive Terminierung dar. Hierbei werden die Busleitungen über je einen 270 Ohm Widerstand mit einer gemeinsamen 2,6V - Konstantspannungsquelle verbunden (Bild 2).



Ver-16-fachung des Sonder-ROM-Bereichs und Nutzung durch das Betriebssystem

Helmut Bernhardt (Hardware); Arnulf Sopp (Software)

Die übliche Technik, das Betriebssystem zu tunen, führt in der Regel dazu, daß durch Herunterrutschen des HIMEM der Anwender-RAM-Speicher immer kümmerlicher wird, wobei anspruchsvollere Programme zwar eine Fülle zusätzlicher Features des Betriebssystems zur Verfügung haben (ob sie wollen oder nicht) und dafür eventuell zu wenig Speicherplatz für sich und ihre Daten/ Variablen haben.

Zwei Möglichkeiten, wie zusätzlicher Speicherplatz geschaffen werden kann, der nicht auf Kosten des Anwenders geht, sind in 1) und 2) beschrieben. Während die in 1) beschriebene Methode (vollständige Decodierung der Tastatur 3800H- 38FFH und Bereitstellen von RAM im Bereich 3900H- 3BFFH) mit allen Hard- und Software-Produkten zusammen funktioniert, ist die in 2) beschriebene Erweiterung an das Vorhandensein eines herkömmlichen Expansion-Interface EG 3014 mit Sockeln für 2 x 8 dynamische RAMs gebunden und die Vervielfältigung des Sonder ROM Bereichs ist in das ganze Projekt nur als Abfallprodukt mit eingebunden.

Hier soll nun beschrieben werden, wie man die Vervielfältigung des Sonder-ROM-Bereichs und gleichzeitig dessen vollständige Decodierung, so daß auch der PUNCH Befehl des GENIE-Monitors (Maschinen Programme auf Cassette schreiben) aus 3) funktioniert, durchführen kann, ohne das aufwendige Banking-Board zur Verwaltung zusätzlicher dynamischer RAMs aufbauen zu müssen.

Zunächst grob qualitativ das Prinzip der Schaltung:

In ein portdecodiertes Latch (74116) wird ein 4-Bit-Wort als Nummer der einzuschaltenden Speicherbank eingetragen. Dieses 4-Bit-Wort liegt anschließend am Latch-Ausgang ständig an, bis ein anderes Wort eingetragen wird. Die Latchausgänge werden an einen 1-aus-16-Decoder (74154) gelegt, der dann je nach Bitmuster an seinen 4 Eingangs-Pins den entsprechenden Ausgangspin low aktiv werden läßt, wenn seine beiden E*-Eingänge low aktiv sind. Wenn man an diese E*-Eingänge das volldecodierte Freigabesignal 3000H-37DFH* für das Sonder-ROM legt, können die 16 Ausgänge des 74154 als Freigabesignal für insgesamt 16 Stück 2716-EPROMs benutzt werden.

Will man anstelle von EPROMs auch RAMs benutzen, kann man nicht einfach die Signale vom Sockel des Sonder-ROMs an die 16 Speicherbausteine (2716-EPROMs oder 6116-RAMs) führen.

1) werden die am Sonder-ROM-Sockel anliegenden Datenleitungen über unidirektionale Treiber nur in Leserichtung mit dem Systemdatenbus verbunden, weshalb ein Schreiben in RAMs so nicht möglich ist und

2) liegt an Pin 21 nicht das für RAMs nötige Signal WR* bzw. MWR* an. Dieser Pin ist einfach an +5V angeschlossen.

3) ist an den CE*-Pin 18 Masse angelegt und ein nicht vollständig decodiertes Freigabesignal 12-14K* (=3000H-37FFH*) ist an den Pin 20 OE* geführt, der eigentlich ein RD*- oder MRD*-Signal verdient hätte. Kollisionen mit den I/O-Adressen 37E0H-37FFH werden dadurch verhindert, daß durch das Signal 3600H-37FFH* an Pin 13 von 221 (74LS20) die Freigabe des Speicherlesetreibers gesperrt ist. In diesem großzügigen Abwürgen des Sonder-ROMs liegt auch die Ursache des nicht funktionierenden PUNCH-Befehls.

Um diesem Übel abzuhelpfen, sind folgende Eingriffe nötig:

1) Die Datenleitungen zum Sockel des Sonder-ROMs müssen durchtrennt werden.

2) Die Datenpins des Sonder-ROMs werden direkt mit den Ausgängen der Datenbustransceiver der CPU verbunden.



3) Die Datenleitung D7 die über den Sonder-ROM-Sockel an den RAM-Baustein 234 führt, muß zwischen RAM und Sonder-ROM durchtrennt werden. D7 für 234 muß stattdessen vom ROM 212 an das RAM geführt werden.

4) Die Leitungen zu den Pins 18 und 19 des Sonder-ROM-Sockels (GND und +5V) werden durchtrennt.

5) An Pin 18 des Sockels wird MRD* und an Pin 21 wird MWR* gelegt.

6) Das Signal 3600H-37FFH* wird von der Steuerung des Speicherlesetreibers abgetrennt. Stattdessen wird dort 12-14K* angelegt, so daß der Speicherlesetreiber für den gesamten Bereich 3000H- 37FFH gesperrt wird.

7) Durch ein Flachkabel und 2 24-polige Quetschstecker und ein in dieses Kabel eingefügtes Adapter-Board wird die 16-fach-Sonder-ROM/RAM-Karte mit dem wie oben beschriebenen geänderten Sonder-ROM-Sockel verbunden. Das Adapter-Board bringt nicht nur die Signale CE* und MRD* auf die richtigen Pins sondern leistet auch die volle Decodierung auf den Bereich 3000H- 37DFH und stellt einen bidirektionalen Datenbustransceiver bereit, so daß auch RAMs benutzt werden können.

Die Wahl eines 74116 Latches bietet den Vorteil, daß durch AND-Verknüpfung der Signale RESET* und NMI* (ergibt das beim TRS 80 bereits vorhandene Signal SYSRES*) und Anlegen an den CLEAR-Eingang des Latch nach dem Einschalten oder Drücken des "RESET"-Knopfes immer die Bank 0 eingestellt ist. Wenn der bisherige Sonder-ROM auf der Ver-16-fachungs-Karte durch den Ausgang 0 des 1-aus-16-Decoders 74154 freigegeben wird, verhält sich der Computer wie vorher auch. Wenn keine andere Bank eingeschaltet wird, liegt im Adreßbereich 3000H-37DFH immer der Sonder-ROM vor.

Arbeitsanleitung

1) Änderungen am Sonder-ROM-Sockel auf dem CPU-Board

Auf der Lötseite werden die Leitungen zu den Pins 18 und 21 des Sonder-ROMs durchtrennt. An Pin 18 wird das Signal MRD* von 215 (74LS32), Pin 11 und an Pin 21 das Signal MWR* von 215 (74LS32), Pin 3 gelegt.

Die Verbindung zwischen Pin 14 des RAMs 234 mit Pin 17 des Sonder-ROMs wird durchtrennt. Der Pin 14 des RAMs wird stattdessen mit dem Pin 17 des ROMs 3003 verbunden.

Auf der Bestückungsseite werden (von der Mitte des CPU-Boards aus gezählt) die ersten 7 Leitungen zwischen ROM 3 (3003) und dem Sonder-ROM-Sockel durchtrennt. Die 8. Leitung bleibt bestehen und die 9. Leitung wird ebenfalls durchtrennt. Dann werden mit isolierten Drähten auf der Lötseite folgende Verbindungen hergestellt:

Signal	Pin des Sockels	Verbindung zu	
		Pin von IC	Typ
D0	9	7	218 74LS367
D1	10	9	"
D2	11	5	"
D3	13	13	219
D4	14	11	"
D5	15	3	218
D6	16	13	"
D7	17	11	"

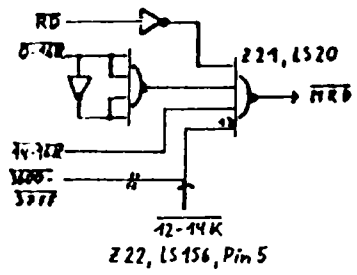


Abb. 1: Freigabe des Speicherlese-Treibers auf dem CPU-Board für den Betrieb der Ver-16-fachungs-Karte

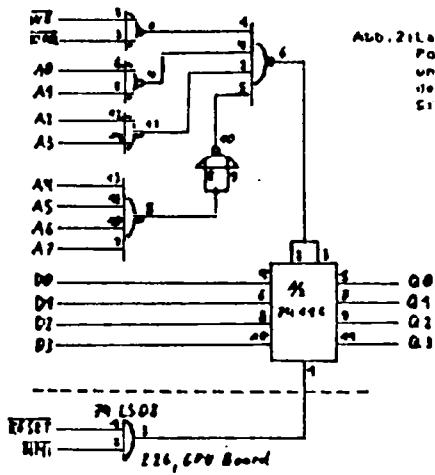


Abb. 2: Parity-Board mit Paritycodierung und Erzeugung des Signals SENSE

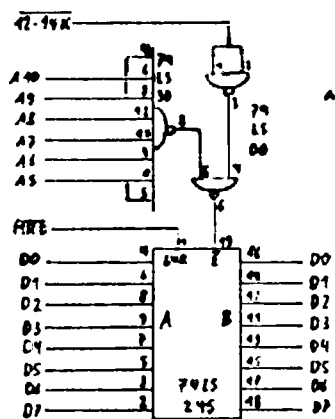


Abb. 3: Adapter-Board mit vollständiger Paritycodierung und Datenübertragung

15

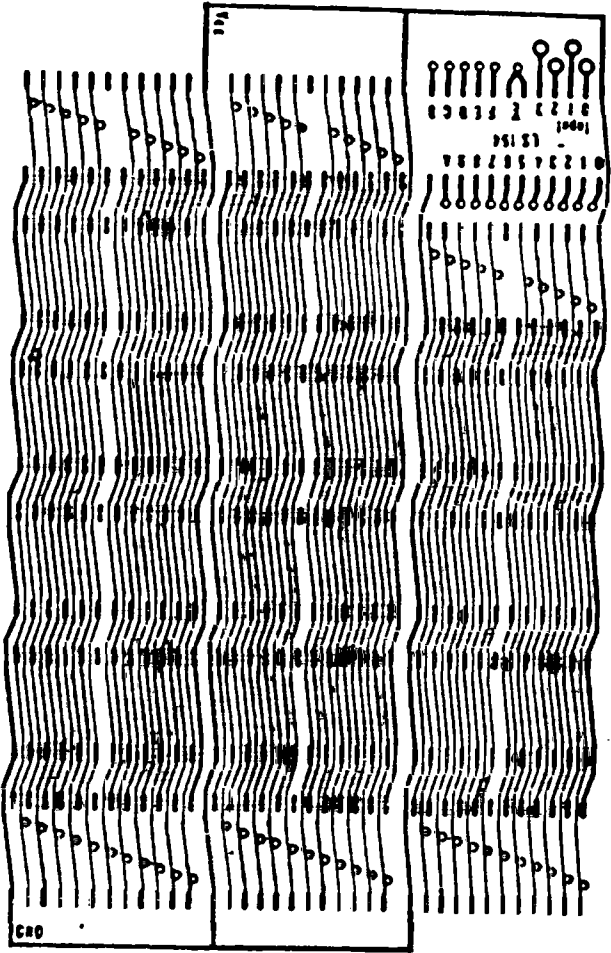


Abb. 4: Layout des Ver-16-fachungs-Boards
(Ansicht Lötseite)

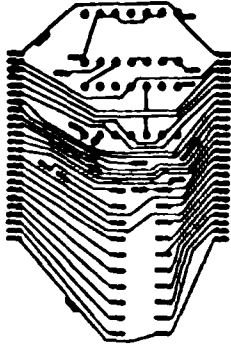


Abb.5:Layout des Adapter-Boards
(Ansicht Lötseite)

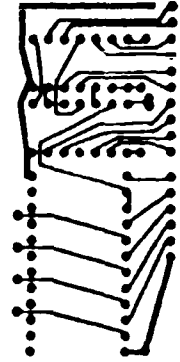


Abb.6:Layout des Latch-Boards
(Ansicht Lötseite)

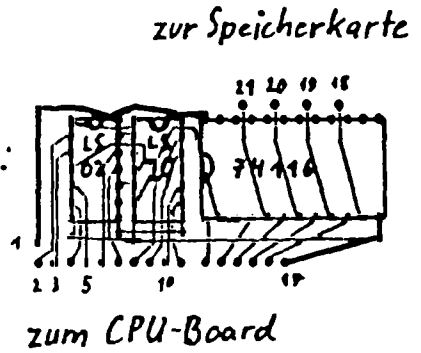
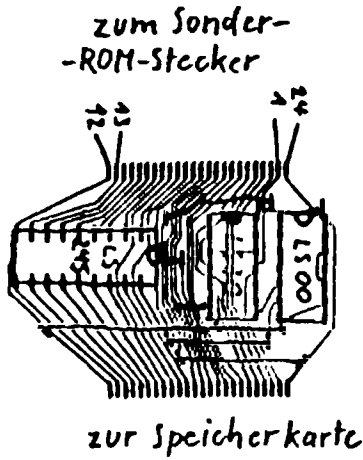


Abb.7:Bestückung und Anschlüsse des Adapter-Boards und
des Latch-Boards

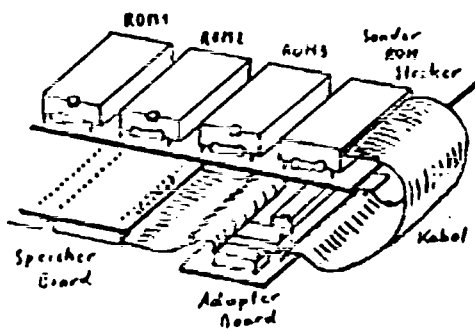


Abb. B10 Orientierung beim Zusammenbau von CPU-Board, Adapter-Board und Ver-16-fachungs-Board

Literatur:

- 1) H. Bernhardt: "Ein sicheres Plätzchen - Maschinenroutinen geschützt im Genie untergebracht", c't 5/1985, S. 126
- 2) H. Bernhardt, C. Ueberschaar: "192k RAM im Genie I und II", GENIE DATA 6/1984, S. 23-46
- 3) J. Tegeder: "Der Monitor im Video-Genie", mc 12/1983, S. 75

... und dazu etwas Software

Wie nützlich ein Umbau am Computer ist, zeigt sich natürlich erst, wenn er etwas zu tun bekommt. Das folgende Programm ist nur ein Beispiel, was man mit dem Versechzehnfachen anfangen kann. Jedes andere Maschinenprogramm, sogar BASIC-, Pascal- oder sonstige höhere Programmtexte können dort untergebracht werden (1,2), natürlich auch Datenbytes für Tabellen und dergleichen. Damit ist das leidige Himm-Theme von Tisch; nahezu jede popelige BASIC-Verschönerung (eingerrückte Zeilennummern und ähnlich überflüssiges) drängt sich dort oben und verschleißt den Platz für wirklich Wichtiges.

Dieses Programm ist ein Treiber für die HRG 1b von RB-Elektronik. Er lädt selbsttätig BASIC/CMD und arbeitet anschließend im Hintergrund mit BASIC zusammen. Sein Ladebereich geht von 3000-32E4. Dahinter sind noch ein paar Bytes als Datenpuffer reserviert. Um beim Laden des Treibers nicht den DOS-Fehler "Ladeversuch auf ROM-Speicherplatz" serviert zu bekommen, muß dort natürlich RAM vorhanden sein. Der User gibt deshalb zuvor auf den Port FO die Nummer eines Steckplatzes auf dem Versechzehnfachen aus, wo ein statisches RAM sitzt. Unter H-DOS geht das mit OUT,FO,xx (3). Ansonsten kann man zunächst BASIC starten und dann OUT240,xx, anschließend CMD"SaHRG" eingeben (sofern man dieses Programm unter dem Namen HRG/CMD auf Disk stehen hat).

A propos H-DOS: Da der Befehl CMD > (mehr darüber weiter unten) das Parallel-RAM auf der CPU-Platine benutzt, müssen Besitzer des EG 64 MRA mit diesem DOS zuvor INI.N eingeben. Damit werden zwar die interessantesten Features von H-DOS disabled, aber noch immer hat der User alle 6-DOS-Möglichkeiten und noch ein bißchen mehr. Auf allen Maschinen, mit denen Banking nicht möglich ist, funktioniert dieser Befehl ganz einfach nicht, ohne aber Schaden anzurichten oder zu einem Fehler zu führen.

Der Treiber stellt eine Reihe von Kommandos an die HRG zur Verfügung. Sie alle beginnen mit CMD (1). Um dem Interpreter anzuzeigen, daß es sich nicht um das gewohnte CMD handelt, folgt unmittelbar darauf nicht eine Variable oder ein Ausdruck zwischen Gänsefüßchen, sondern ein zweites BASIC-Token:

```
CMD +      : HRG einschalten
CMD -      : HRG ausschalten
CMD CLS    : HRG-Speicher löschen
CMD SET    : einen HRG-Punkt setzen
CMD RESET  : einen HRG-Punkt löschen
CMD POINT  : testen, ob ein HRG-Punkt gesetzt ist
CMD <      : HRG invertieren (weiß <-> schwarz)
CMD LPRINT : HRG-Inhalt auf Drucker ausgeben (Hardcopy)
CMD ASC    : ASCII-Bildschirm in Hex anzeigen, bis neue Taste gedrückt
CMD >      : Bildschirm gegen Pufferinhalt austauschen
CMD LINE   : noch nicht programmiert, ergibt aber keinen Fehler
```

Nach der üblichen BASIC-Syntax können CMD und das zweite Token direkt oder durch Blank getrennt hintereinander stehen. Diese Befehle können im Direktmodus und als Bestandteil eines Programms eingegeben werden.

Die drei erstgenannten HRG-Befehle bedürfen keiner näheren Erläuterung. CMD SET, CMD RESET und CMD POINT haben eine ganz ähnliche Syntax wie die gewohnten Befehle zur Ansteuerung der Genie-Pixelgraphik. Im Argument sind im Unterschied dazu jedoch Koordinaten im Bereich 0-3B3 für X und 0-191 für Y zulässig, entsprechend der feineren HRG-Matrix. Die Koordinaten können wie in Level 2 oder Disk-BASIC auch durch Variable dargestellt werden.

CMD < macht aus jedem Punkt im HRG-Speicher sein Komplement. Gesetzte Punkte werden dunkel, nicht gesetzte werden eingeschaltet. Es

entsteht eine negative Darstellung. Der normale ASCII-Bildschirm bleibt hiervon unberührt.

Bei einem Bildschirmausdruck der HRG mit CMD LPRINT wird der Drucker zunächst in den Einschaltzustand versetzt (Reset aller Druckparameter). Danach wird der linke Rand auf die 9. Stelle gesetzt, um das Bild zu zentrieren. Der Zeilenabstand wird mit 6/72" so eingestellt, daß die Druckzeilen lückenlos untereinander stehen. Nach getaner Arbeit werden diese Steuercodes alle wieder gelöscht; der Drucker ist wieder im Einschaltzustand. Diese Codes gelten für den Star Gemini-10X. Für andere Printer müssen sie entsprechend geändert werden. Von den vergleichbaren Epson-Typen ist mir bekannt, daß für die Randeinstellung lediglich in ESC M 09 aus dem "M" ein "I" gemacht zu werden braucht. Die übrigen Codes sind identisch.

Der Befehl CMD ASC ist für vielerlei Verwendungen interessant. Mit ihm hat der Programmierer sozusagen immer eine ASCII-Tabelle auf dem Bildschirm. Die ASCII-Zeichen werden als zweistellige Hexzahlen dargestellt (4). So werden gleiche Zeichen mit verschiedenen Codes unterscheidbar, um nur eine Anwendung zu nennen: Die beiden Blanks mit den Codes 32 und 128 sowie die Graphikblocks 128-191 und 192-255 sind als Hexzahlen eindeutig zu identifizieren. Bei der Ausführung von CMD ASC werden die 32er Blanks übrigens nicht mit umgewandelt, sondern als Leerzeichen belassen, um den Bildschirm übersichtlich zu halten. Dabei wird der normale ASCII-Bildschirm natürlich gelöscht, sonst würde er stören. Der Treiber rettet ihn in den HRG-Speicher, so daß er nach dem Druck auf irgendeine Taste wieder restauriert werden kann. Ein besonderer Puffer im RAM ist dafür nicht erforderlich.

Nach diesem Schema geht auch der Befehl CMD > vor. Zunächst wird der Bildschirm in die HRG gerettet. Von ihr sind nämlich nur 6 Bits pro Byte sichtbar, die beiden höchstwertigen Bits sind unbenutzt und können den Bildschirm aufnehmen. Dabei wird der sichtbare Inhalt des HRG-Speichers nicht beeinträchtigt. Alle Graphiken bleiben also erhalten. Im weiteren Verlauf der Ausführung dieses Befehls wird nun der Inhalt der HRG, also Graphik und ASCII, in einen Puffer gerettet, und der Inhalt dieses Puffers geht dafür in die HRG. Die beiden höchstwertigen Bits des neuen HRG-Inhalts werden in den ASCII-Bildschirm übertragen. Es ist demnach ein Austausch Bildschirm gegen Puffer. In diesem Treiber liegt der Puffer im Adreßbereich ab 0000. Das ist natürlich nur mit dem EG 64 MBA oder einem anderen Banker realisierbar. Entsprechende Änderungen, um stattdessen das gute alte HiMem dafür zu benutzen, sind kein Problem.

Der Befehl CMD LINE soll eines fernen Tages, wenn der Autor mal wieder Lust hat, an dem Treiber weiterzuarbeiten, Linien, Rechtecke usw. ziehen. Der Leser möge mir nachsehen, daß die Computerei mein Hobby ist, bei dem ich gnadenlos dem Lustprinzip folge (6).

An diesen fernen Tage werden wohl noch ein paar andere Modifikationen folgen, die es ermöglichen, den HRG-Treiber fest in ein EPROM zu brennen. Das ist mit dieser Version nicht möglich, denn das Programm vollzieht Schreibzugriffe in seinem eigenen Adreßraum. Dazu gehören z. B. die Datenpuffer an seinem Ende, aber auch variable Sprungdistanzen mitten im Programm (Selbstmodifikation). Der Vorteil dieser Variante ist freilich, daß der Treiber mit den Bedürfnissen jederzeit mitwachsen kann. Immerhin sind noch eineinviertel kB frei!

Auf die Programlogik möchte ich an dieser Stelle nicht eingehen. Die siebeneinhalb Seiten Listing wären halbwegs erschöpfend wohl nicht mit weniger als 15 Seiten Text erklärbar. Insbesondere diejenigen Features, die kein anderer mir bekannter HRG-Treiber zur Verfügung stellt, würden einige Seiten füllen: Z. B. die Umwandlung des ASCII-Bildschirms in die Hexdarstellung und das Puffern desselben im HRG-Speicher. Der Maschinensprache-Freak mag sich dabei amüsieren, sich in dem sehr ausführlich kommentierten Listing selber zurechtzufinden. Zu diesem Thema

3091	DB04	00067	IN	A,(4)	!HL	yte laden
3093	46	00068	LD	B,(HL)		!Pufferbyte holen
3094	77	00069	LD	(HL),A		!HRG-Byte puffern
3095	78	00070	LD	A,B		!Pufferbyte
3096	D305	00071	OUT	(5),A		!auf HRG ausgeben
3098	23	00072	INC	HL		!Pufferzeiger erhöhen
3099	7A	00073	LD	A,D		!überprüfen, ob
309A	B9	00074	CP	C		!HRG-Bereich überschr.?
309B	3BF1	00075	JR	C,swaplop		!falls nein
309D	DBDF	00076	IN	A,(0dfh)		!Banks rücksetzen
309F	FB	00077	EI			!INTs wieder zulassen
30A0	CD5532	00078	CALL	savrest		!Bildsch. restaurieren
30A3	E1	00079	POP	HL		!Befehlszeiger
30A4	1831	00080	JR	exit1		!fertig
		00081				
		00082				!CMD CLS: HRG-Speicher löschen
30A6	FEB4	00083	CLS	CP	B4h	!CLS?
30AB	2004	00084	JR	NZ,INV		!falls nein
30AA	1E00	00085	clear	LD	E,0	!Flag für HRG-CLS
30AC	1806	00086	JR	clsinv		!dort weiter
		00087				
		00088				!CMD <: HRG invertieren (positiv -> negativ)
30AE	FED6	00089	INV	CP	0d6h	!invertieren?
30B0	2027	00090	JR	NZ,LINE		!falls nein
30B2	1EFF	00091	LD	E,0ffh		!Flag für Inversion
30B4	010330	00092	clsinv	LD	BC,3003h	!B = MSB 12kB, C = Port 3
30B7	C5	00093	msloop	PUSH	BC	!retten
30BB	05	00094	DEC	B		!B um 1 zu hoch
30B9	ED41	00095	OUT	(C),B		!MSB HRG-Adresse
30BR	0600	00096	LD	B,0		!B <- 0 für HRG-LSB
30BD	0D	00097	DEC	C		!Port 2 für LSB
30BE	ED41	00098	lsloop	OUT	(C),B	!LSB HRG-Adresse
30C0	DB04	00099	IN	A,(4)		!HRG-Byte holen
30C2	F5	00100	PUSH	AF		!retten
30C3	E6C0	00101	AND	0c0h		!obere Bits isolieren
30C5	57	00102	LD	D,A		!und retten
30C6	F1	00103	POP	AF		!HRG-Byte
30C7	2F	00104	CPL			!invertieren
30CB	E63F	00105	AND	3fh		!obere Bits ausmaskieren
30CA	CB7B	00106	BIT	7,E		!CLS oder INV?
30CC	2001	00107	JR	NZ,outa		!falls INV
30CE	AF	00108	XOR	A		!sonst löschen
30CF	B2	00109	outa	DR	D	!obere Bits hinzufügen
30D0	D305	00110	OUT	(5),A		!Blank od. Kompl. ausgeb.
30D2	10EA	00111	DJNZ	lsloop		!LSB: 00, FF, FE ... 01
30D4	Cj	00112	POP	BC		!MSB und Port für MSB
30D5	10E0	00113	DJNZ	msloop		!MSB-1: 2F ... 00
		00114				
		00115				!Rückkehr für mehrere Unterprogramme
30D7	23	00116	exit1	INC	HL	!Befehlszeiger nachst.
30D8	C9	00117	RET			!erledigt
		00118				
		00119				!CMD LINE: (bisher nicht programmiert)
30D9	FE9C	00120	LINE	CP	9ch	!LINE?
30D8	2BFA	00121	JR	Z,exit1		!vorl. nicht implement.
		00122				
		00123				!CMD +: HRG-Speicher in den Bildschirm einblenden
30DD	FEC0	00124	CP	0cdh		!HRG einschalten?
30DF	2004	00125	JR	NZ,HRGoff		!falls nein
30E1	D301	00126	OUT	(1),A		!einschalten
30E3	1BF2	00127	JR	exit1		!fertig
		00128				
		00129				!CMD -: HRG-Speicher aus dem Bildschirm ausblenden
30E5	FECE	00130	HRGoff	CP	0ceh	!HRG ausschalten?
30E7	2004	00131	JR	NZ,LPRINT		!falls nein
30E9	D300	00132	OUT	(0),A		!ausschalten

```

30EB 18EA 00134 JR exit1 ;fertig
00135 ;CMD LPRINT: Hardcopy des HRG-Speichers
30ED FEAF 00136 LPRINT CP 0afh ;LPRINT (Hardcopy)?
30EF 2066 00137 JR NZ,ASC ;falls nein
00138
00139 ;LFRINT; Drucker initial.: Reset, 6/72" Zeilenabstand
30F1 E5 00140 PUSH HL ;Befehlszeiger retten
30F2 21D632 00141 LD HL,prestora+1 ;Druckerinitialisierung
30F5 0609 00142 LD B,9 ;mit 9 Codes
30F7 CDFC31 00143 CALL lprint ;Ausgabe auf Drucker
00144
00145 ;HRG-Zeiger laden, 16 (Doppel-) Zeilen vorbereiten
30FA 50 00146 LD D,B ;DE <- 00xx
30FB 58 00147 LD E,B ;DE <- 0000, Start HRG
30FC 010610 00148 LD BC,1006h ;16 Zeilen, Konst. 6
00149
00150 ;1 Doppelzeile zu je 6 Dotzeilen drucken
30FF C5 00151 scrnlop PUSH BC ;Zähler retten
3100 D5 00152 PUSH DE ;dto. HRG-Zeiger
3101 0602 00153 LD B,2 ;2 Halbzeilen/Zeile
3103 C5 00154 linelop PUSH BC ;wird verändert
3104 D5 00155 PUSH DE ;dto. HRG-Zeiger
3105 21DF32 00156 LD HL,lninit ;Druckerinit. für 1 Zeile
3108 0605 00157 LD B,5 ;mit 5 Codes
310A CDFC31 00158 CALL lprint ;Ausgabe
310D 21E432 00159 LD HL,buffer ;Puffer für Druckerodes
3110 0640 00160 LD B,40h ;64 Bytes/Zeile
00161
00162 ;1 Halbzeile drucken
3112 C5 00163 hlinlop PUSH BC ;Zähler retten
3113 D5 00164 PUSH DE ;dto. HRG-Zeiger
3114 7B 00165 LD A,E ;LSB des HRG-Zeigers
3115 D302 00166 OUT (2),A ;auf HRG ausgeben
3117 41 00167 LD B,C ;6 Bytes senkrecht
00168
00169 ;1 Byte drucken
311B C5 00170 bytelop PUSH BC ;wird verändert
3119 7A 00171 LD A,D ;MSB des HRG-Zeigers
311A D303 00172 OUT (3),A ;auf HRG ausgeben
311C DB04 00173 IN A,(4) ;Dotzeile aus HRG
311E 41 00174 LD B,C ;6 Dots/Stelle
311F E5 00175 PUSH HL ;retten
00176
00177 ;eal 1 Bit errechnen
3120 0F 00178 bitlop RRCA ;Cy <- HRG-Bit
3121 CB16 00179 RL (HL) ;nächstes Pufferbit <- Cy
3123 CBB6 00180 RES 6,(HL) ;nur untere Bits
3125 23 00181 INC HL ;nächstes Pufferbyte
3126 10FB 00182 DJNZ bitlop ;bis 6 Dots gepuffert
00183
00184 ;6 Bits fertig - 6 Bytes vervollständigen
312B 14 00185 INC D ;MSB auf nächste Dotzeile
3129 14 00186 INC D ;(= um 1 kB erhöhen)
312A 14 00187 INC D
312B 14 00188 INC D
312C E1 00189 POP HL ;Pufferzeiger
312D C1 00190 POP BC ;Zähler
312E 10EB 00191 DJNZ bytelop ;bis 1 Stelle in Puffer
00192
00193 ;6 Bytes ausdrucken und weiter mit Halbzeile
3130 E5 00194 PUSH HL ;wird verändert
3131 41 00195 LD B,C ;6 Dotspalten
3132 CDFC31 00196 CALL lprint ;ausgeben
3135 E1 00197 POP HL ;Pufferzeiger
3136 D1 00198 POP DE ;alter HRG-Zeiger

```

31E7	7C	00331	LD	A,H	;MSB
31E8	D303	00332	OUT	(03h),A	;dto.
31EA	C1	00333	POP	BC	;B ← Bit-Nr.
31EB	04	00334	INC	B	;B ← mindestens 1
31EC	3E80	00335	LD	A,80h	;Anfangswert für A
31EE	07	00336	doublop RLCA		;fortgesetzt A&2
31EF	10FD	00337	DJNZ	doublop	;bis richtiges Bit in A
31F1	E63F	00338	AND	3fh	;nur die ersten 6 Bits
31F3	4F	00339	LD	C,A	;Akku retten
31F4	DB04	00340	IN	A,(4)	;Inh. HRG-Speicherstelle
31F6	47	00341	LD	B,A	;HRG-Byte retten
31F7	F1	00342	POP	AF	;Token restaurieren
31F8	2AEA32	00343	LD	HL,(candbuf)	;Befehlszeiger restaur.
31FB	C9	00344	RET		;erledigt
		00345			
		00346			;Zeichenfolge ab (HL) auf Drucker ausgeben
31FC	CDD105	00347	lprint CALL	05dh	;Drucker bereit?
31FF	20FB	00348	JR	NZ,lprint	;falls nein
3201	7E	00349	LD	A,(HL)	;zu druckendes Zeichen
3202	D3FD	00350	OUT	(0fdh),A	;auf Drucker ausgeben
3204	23	00351	INC	HL	;nächste Stelle
3205	10F5	00352	DJNZ	lprint	;nächstes Zeichen
3207	C9	00353	RET		;zurück
		00354			
		00355			;UP Hexanzeige: 1 Byte ändern
3208	7C	00356	byte LD	A,H	;MSB der Videoadresse
3209	E603	00357	AND	03	;Adr. Vid. → Adr. HRG
320B	57	00358	LD	D,A	;neues MSB
320C	5D	00359	LD	E,L	;HRG-MSB wie Video-MSB
320D	79	00360	LD	A,C	;Videozeichen
320E	E&F0	00361	AND	0f0h	;oberes Nibble
3210	0F	00362	RRCA		;ins untere schieben
3211	0F	00363	RRCA		
3212	0F	00364	RRCA		
3213	0F	00365	RRCA		
3214	21B032	00366	LD	HL,chrTAB-5	;vor Tab. f. Hexzeich.
3217	E5	00367	PUSH	HL	;brauchen wir noch
3218	CD1F32	00368	CALL	nibble	;oberes Nibble anzeigen
321B	79	00369	LD	A,C	;alter Code
321C	E60F	00370	AND	0fh	;unteres Nibble
321E	E1	00371	POP	HL	;Tabellenzeiger
		00372			
		00373			;einzelnes Halbbyte in die HRG laden
321F	47	00374	nibble LD	B,A	;als Zähler i. d. Tabelle
3220	3A3932	00375	LD	A,(displc)	;Sprungdistanz
3223	EE02	00376	XOR	output-displc-1	;ueschalten
3225	323932	00377	LD	(displc),A	;neu laden
3228	04	00378	INC	B	;wegen DE = Tabelle -5
3229	23	00379	seekchr INC	HL	;Zeiger nachstellen
322A	23	00380	INC	HL	;über 5 Stellen, weil
322B	23	00381	INC	HL	;5 Codes pro Zeichen
322C	23	00382	INC	HL	
322D	23	00383	INC	HL	
322E	10F9	00384	DJNZ	seekchr	;bis Code gefunden
3230	6005	00385	LD	B,5	;5 Dotzeilen pro Zeichen
3232	05	00386	nibloop PUSH	BC	;Zähler retten
3233	CD4B32	00387	CALL	HRGadr	;HRG-Stelle adressieren
3236	1B	00388	DEC	DE	;HRG-Zeiger korrigieren
3237	7E	00389	LD	A,(HL)	;Dotzeile laden
3238	1B00	00390	JR	s+2	;variable Sprungdistanz
3239		00391	displc EQU	s-1	;hier Distanzbyte
323A	07	00392	RLCA		;lower Nibble verschieben
323B	07	00393	RLCA		;um 2 Dots
323C	4F	00394	output LD	C,A	;Dotzeile retten
323D	DB04	00395	IN	A,(4)	;HRG-Byte mit Videocode
323F	B1	00396	OR	C	;mit Dotzeile verknüpfen

```

3240 D305 0039 OUT (5),A ;Dotzeile in HRG laden
3242 14 0039B INC D ;im MSB um 1 kB erhöhen
3243 14 00399 INC D ;für nächste Dotzeile
3244 14 00400 INC D
3245 14 00401 INC D
3246 23 00402 INC HL ;nächster Code für Ziffer
3247 C1 00403 POP BC ;Zähler restaurieren
324B 10EB 00404 DJNZ nibloop ;bis Nibble angezeigt
324A C9 00405 RET ;zurück
00406
00407 ;UP, um die HRG-Adresse auszugeben
324B 7B 0040B HRGadr LD A,E ;LSB der HRG-Adresse
324C D302 00409 OUT (2),A ;auf Port ausgeben
324E 7A 00410 LD A,D ;MSB
324F D303 00411 OUT (3),A ;dto.
3251 13 00412 INC DE ;nächste HRG-Stelle
3252 C9 00413 RET
00414
00415 ;UP, um Bildschirm zu retten oder zu restaurieren
3253 0E00 00416 vidsav LD C,0c0h ;Konstante
3255 21003C 00417 savrest LD HL,3c00h ;Bildschirmadresse
325S 55 00418 LD D,L ;DE <- 00xx, HRG-Adresse
3259 5D 00419 LD E,L ;DE <- 0000
325A 0604 00420 vidsavl LD B,4 ;4 * 2 Bits/Byte
325C CD4B32 00421 vidsav2 CALL HRGadr ;HRG-Stelle adressieren
325F CB79 00422 BIT 7,C ;Bildschirm retten?
3261 2B12 00423 JR Z,restor ;falls nein
3263 7E 00424 LD A,(HL) ;Bildschirmzeichen
3264 A1 00425 AND C ;nur oberste 2 Bits
3265 C5 00426 PUSH BC ;C retten
3266 4F 00427 LD C,A ;Akku retten
3267 DB04 00428 IN A,(4) ;HRG-Byte holen
3269 E63F 00429 AND 3fh ;oberste Bits ausmaskieren.
326B B1 00430 OR C ;Videobits zufügen
326C C1 00431 POP BC ;C restaurieren
326D D305 00432 OUT (5),A ;auf HRG ausgeben
326F CB06 00433 RLC (HL) ;Zeichen 2 Bits aufrücken
3271 CB06 00434 RLC (HL)
3273 1808 00435 JR gosave ;dort weiter
3275 DB04 00436 restor IN A,(4) ;HRG-Byte holen
3277 07 00437 RLCA ;2 oberste Bits
327B CB16 00438 RL (HL) ;in den Bildsch. laden
327A 07 00439 RLCA
327B CB16 00440 RL (HL)
327D 10DD 00441 gosave DJNZ vidsav2 ;bis 1 Byte fertig
327F 23 00442 INC HL ;nächste Videostelle
3280 CB74 00443 BIT 6,H ;Bildsch. überschritten?
3282 C0 00444 RET NZ ;falls ja
3283 18D5 00445 JR vidsavl ;weiter, falls nein
00446
00447 ;Codetabelle für HRG-Zeichensatz
3285 02 00448 chrstab DB 2,5,5,5,2 ;0
328A 04 00449 DB 4,6,5,4,4 ;1
328F 03 00450 DB 3,4,2,1,7 ;2
3294 07 00451 DB 7,4,2,4,7 ;3
3299 04 00452 DB 4,5,7,4,4 ;4
329E 07 00453 DB 7,1,3,4,3 ;5
32A3 06 00454 DB 6,1,7,5,2 ;6
32A8 07 00455 DB 7,4,2,1,1 ;7
32AD 02 00456 DB 2,5,2,5,2 ;8
32B2 02 00457 DB 2,5,7,2,1 ;9
32B7 02 00458 DB 2,5,7,5,5 ;A
32BC 03 00459 DB 3,5,3,5,3 ;B
32C1 06 00460 DB 6,1,1,1,6 ;C
32C6 03 00461 DB 3,5,5,5,3 ;D
32CB 07 00462 DB 7,1,3,1,7 ;E

```

```

32D0 07      00463      DB      7,1,3,1,1      ;F
            00464
            00465 ;versch. Drucker codes, Puffer für Dotspalten usw.
32D5 0D      00466 prestor DB      0dh,0dh,1bh,'8' ;Drucker reinitialisieren
32D9 1B      00467      DB      1bh,'M',09h      ;li. Rand auf 9. Stelle
32DC 1B      00468      DB      1bh,'A',06h      ;Zeilenabstand 6/72"
32DF 0D      00469 lninit  DB      0dh,1bh,'K',80h,01h ;Dotgr. einf. D.
32E4 0000    00470 buffer  DW      0000h,0000h,0000h ;Puffer f. HRG
32EA        00471 cmdbuf  EQU      $              ;Puffer für Befehlszeiger
            00472
3000        00473      END      init

```

00000 Fehler

ASC	3157	CLS	30A6	HRGadr	324B	HRGoff	30E5	INV	30AE
LINE	30D9	LPRINT	30ED	PSR	3181	PSRexit	319A	argum	31A1
bitlop	3120	buffer	32E4	byte	320B	byteloop	311B	chrtab	32B5
clear	30AA	clsinv	30B4	cmd	3074	cmdbuf	32EA	command	302C
displc	3239	doublop	31EE	exit1	30D7	exit2	3155	fcterr	31AD
gosave	327D	hello	303B	hexdisp	3166	hlinlop	3112	init	3000
linelop	3103	lninit	32DF	lprint	31FC	lsloop	30BE	msloop	30B7
nibble	321F	nibloop	3232	out	319B	outa	30CF	output	323C
point	319D	prestor	32D5	reset	3192	restor	3275	savrest	3255
scrnlop	30FF	seekchr	3229	set	3196	swaplop	30BE	vidsav	3253
vidsav1	325A	vidsav2	325C						